Model Test Results Report

Automated Test Documentation for Model Governance

MKM Research Labs

Version 1.0 — 06-April-2026

David K Kelly

CONFIDENTIAL — PROPRIETARY

Legal Notice

PROPRIETARY AND CONFIDENTIAL

This document, including all algorithms, models, methodology, formulas, calculations, and intellectual concepts contained herein, constitutes the exclusive intellectual property and confidential trade secrets of MKM Research Labs ("MKM").

Any usage, reproduction, distribution, modification, or disclosure of this document or any portion thereof, without the express prior written authorisation from MKM Research Labs is strictly prohibited and constitutes an infringement of intellectual property rights.

The document is provided on an "as is" basis. MKM Research Labs makes no representations or warranties of any kind, express or implied, regarding its accuracy, reliability, or suitability for any particular purpose.

All rights reserved. © 2021–2026 MKM Research Labs.

Governed by the laws of the United Kingdom.

Contents

1 Executive Summary **3**

1.1 Test Environment 3

1.2 Results by Model 3

2 Test Traceability Matrix **4**

3 Flood Risk (Depth-Damage) (MKM-DD-001) **170**

4 Flood Polynomial Model (MKM-FPO-001) **173**

5 GEV Hazard Model (MKM-GH-001) **175**

6 GaugeHD Synthetic (MKM-GHD-001) **177**

7 Insurance Premium (MKM-IP-001) **181**

8 Mortgage Pricer (MKM-MP-001) **183**

9 PRS Pricing Model (MKM-PR-001) **187**

10 Property Valuation (MKM-PV-001) **191**

11 Storm-Gauge Response Model (MKM-SG-001) **194**

12 Storm Intensity Distribution (MKM-SI-001) **197**

13 Spatial Interpolation (MKM-SP-001) **201**

14 Storm Sequence Generator (MKM-SS-001) **203**

15 Stress Test Pipeline (MKM-ST-001) **205**

16 Trading Desk (MKM-TD-001) **211**

17 CDM Schema Validation (CDM-ALL) **222**

18 End-to-End Browser Tests (E2E-ALL) **225**

A Platform Infrastructure Tests **236**

1 Executive Summary

This report documents the results of automated model tests run on **06 April 2026 at 07:49:50**.

Table 1: Overall Test Results

Total Tests	7255
Passed	7205
Failed	12
Skipped	38
Pass Rate	99.3%
Total Duration	445.13s

1.1 Test Environment

Platform	macOS-26.3.1-arm64-arm-64bit-Mach-O
Python	3.13.1
pytest	8.3.4
Coverage	98.5%
Git Commit	`ce8cb657dcc2`

1.2 Results by Model

Table 2: Test Results by Model

Model ID	Model Name	Total	Pass	Fail	Skip	Time (s)
MKM-DD-001	Flood Risk (Depth-Damage)	32	32	0	0	0.00
MKM-FPO-001	Flood Polynomial Model	33	33	0	0	0.00
MKM-GH-001	GEV Hazard Model	16	16	0	0	0.26
MKM-GHD-001	GaugeHD Synthetic	62	62	0	0	27.81
MKM-IP-001	Insurance Premium	36	36	0	0	0.00
MKM-MP-001	Mortgage Pricer	67	67	0	0	0.05
MKM-PR-001	PRS Pricing Model	57	57	0	0	7.70
MKM-PV-001	Property Valuation	55	55	0	0	16.55
MKM-SG-001	Storm-Gauge Response Model	41	41	0	0	0.04
MKM-SI-001	Storm Intensity Distribution	63	63	0	0	0.02
MKM-SP-001	Spatial Interpolation	41	41	0	0	0.01
MKM-SS-001	Storm Sequence Generator	17	17	0	0	0.00
MKM-ST-001	Stress Test Pipeline	107	107	0	0	20.75
MKM-TD-001	Trading Desk	212	212	0	0	1.31
CDM-ALL	CDM Schema Validation	33	33	0	0	0.00
E2E-ALL	End-to-End Browser Tests	229	197	12	20	1814.40
PLATFORM	Platform Infrastructure	6154	6136	0	18	256.92

2 Test Traceability Matrix

The following table maps every executed test to its assigned model component, providing evidence of coverage required under SR 11-7 §4.2 and SS1/23.

Model	Module	Test Name	Status
MKM-DD-001	velocity.py::TestBuildPropertyHydrograph	test_all_hours_present	Pass
	velocity.py::TestBuildPropertyHydrograph	test_depth_never_negative	Pass
	velocity.py::TestBuildPropertyHydrograph	test_each_entry_has_required_keys	Pass
	velocity.py::TestBuildPropertyHydrograph	test_empty_readings	Pass
	velocity.py::TestBuildPropertyHydrograph	test_high_floor_prevents_flooding	Pass
	velocity.py::TestBuildPropertyHydrograph	test_no_flooding_before_arrival	Pass
	velocity.py::TestBuildPropertyHydrograph	test_output_length_matches_input	Pass
	velocity.py::TestBuildPropertyHydrograph	test_peak_produces_flooding	Pass
	velocity.py::TestComputeSlope	test_basic_slope	Pass
	velocity.py::TestComputeSlope	test_min_slope_clamp	Pass
	velocity.py::TestComputeSlope	test_slope_is_absolute	Pass
	velocity.py::TestComputeSlope	test_zero_distance	Pass
	velocity.py::TestManningVelocity	test_deeper_water_faster	Pass
	velocity.py::TestManningVelocity	test_negative_depth_returns_zero	Pass
	velocity.py::TestManningVelocity	test_negative_slope_uses_abs	Pass
	velocity.py::TestManningVelocity	test_positive_depth_and_slope	Pass
	velocity.py::TestManningVelocity	test_slope_clamped_to_min	Pass
	velocity.py::TestManningVelocity	test_steeper_slope_faster	Pass
	velocity.py::TestManningVelocity	test_zero_depth_returns_zero	Pass
	velocity.py::TestManningVelocity	test_zero_roughness_returns_zero	Pass
	velocity.py::TestRetention	test_at_length_scale	Pass
	velocity.py::TestRetention	test_backwards_compat_alias	Pass
	velocity.py::TestRetention	test_far_distance_near_zero	Pass
	velocity.py::TestRetention	test_near_river_high_retention	Pass
	velocity.py::TestRetention	test_negative_distance	Pass
	velocity.py::TestRetention	test_one_km_retention	Pass
	velocity.py::TestRetention	test_zero_distance_full_strength	Pass
	velocity.py::TestRetention	test_zero_length	Pass
	velocity.py::TestTravelTime	test_basic_travel_time	Pass
	velocity.py::TestTravelTime	test_farther_takes_longer	Pass
	velocity.py::TestTravelTime	test_zero_depth_infinite	Pass
	velocity.py::TestTravelTime	test_zero_distance	Pass
MKM-FPO-001	flood_poly.py::TestConfigIntegration	test_coefficients_from_config	Pass
	flood_poly.py::TestConfigIntegration	test_fit_metadata_from_config	Pass
	flood_poly.py::TestConfigIntegration	test_log_eps_from_config	Pass
	flood_poly.py::TestConfigIntegration	test_storm_hours_from_config	Pass
	flood_poly.py::TestModelCard	test_coefficients_count	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	flood_poly.py::TestModelCard	test_fit_quality_has_metrics	Pass
	flood_poly.py::TestModelCard	test_has_required_fields	Pass
	flood_poly.py::TestModelCard	test_model_id_matches_config	Pass
	flood_poly.py::TestModelCard	test_returns_dict	Pass
	flood_poly.py::TestPFloodPoly	test_extreme_water_near_one	Pass
	flood_poly.py::TestPFloodPoly	test_high_water_high_probability	Pass
	flood_poly.py::TestPFloodPoly	test_hour_boundary_167	Pass
	flood_poly.py::TestPFloodPoly	test_hour_boundary_zero	Pass
	flood_poly.py::TestPFloodPoly	test_hour_zero_lower_than_mid_storm	Pass
	flood_poly.py::TestPFloodPoly	test_in_range	Pass
	flood_poly.py::TestPFloodPoly	test_low_water_low_probability	Pass
	flood_poly.py::TestPFloodPoly	test_monotonic_in_water_level	Pass
	flood_poly.py::TestPFloodPoly	test_negative_severe_returns_zero	Pass
	flood_poly.py::TestPFloodPoly	test_returns_float	Pass
	flood_poly.py::TestPFloodPoly	test_very_low_water_near_zero	Pass
	flood_poly.py::TestPFloodPoly	test_zero_severe_returns_zero	Pass
	flood_poly.py::TestPFloodSeries	test_all_in_range	Pass
	flood_poly.py::TestPFloodSeries	test_correct_length	Pass
	flood_poly.py::TestPFloodSeries	test_empty_list	Pass
	flood_poly.py::TestPFloodSeries	test_returns_list	Pass
	flood_poly.py::TestPFloodSimple	test_above_severe_capped_at_one	Pass
	flood_poly.py::TestPFloodSimple	test_at_severe_equals_one	Pass
	flood_poly.py::TestPFloodSimple	test_in_range	Pass
	flood_poly.py::TestPFloodSimple	test_negative_severe_returns_zero	Pass
	flood_poly.py::TestPFloodSimple	test_ratio_calculation	Pass
	flood_poly.py::TestPFloodSimple	test_returns_float	Pass
	flood_poly.py::TestPFloodSimple	test_zero_severe_returns_zero	Pass
	flood_poly.py::TestPFloodSimple	test_zero_water	Pass
MKM-GH-001	hazard_builder.py::TestGEVFitter	test_exceedance_probability_in_range	Pass
	hazard_builder.py::TestGEVFitter	test_fit_returns_three_floats	Pass
	hazard_builder.py::TestGEVFitter	test_force_gumbel_sets_shape_zero	Pass
	hazard_builder.py::TestGEVFitter	test_higher_threshold_lower_probability	Pass
	hazard_builder.py::TestGEVFitter	test_return_level_increases_with_period	Pass
	hazard_builder.py::TestGaugeResponseModel	test_compute_all_responses_returns_dict	Pass
	hazard_builder.py::TestGaugeResponseModel	test_compute_response_returns_gauge_response	Pass
	hazard_builder.py::TestGaugeResponseModel	test_response_threshold_flags	Pass
	hazard_builder.py::TestHazardCurveBuilder	test_annual_flood_probabilities_in_range	Pass
	hazard_builder.py::TestHazardCurveBuilder	test_build_returns_dict_of_curves	Pass
	hazard_builder.py::TestHazardCurveBuilder	test_each_curve_is_gauge_hazard_curve	Pass
	hazard_builder.py::TestTermStructure	test_cumulative_prob_increasing	Pass
	hazard_builder.py::TestTermStructure	test_probabilities_sum_to_one	Pass
	hazard_builder.py::TestTermStructure	test_returns_correct_length	Pass
	hazard_builder.py::TestTermStructure	test_returns_term_structure_points	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	hazard_builder.py::TestTermStructure	test_survival_prob_decreasing	Pass
MKM-GHD-001	test_synthetic_coverage.py::TestHistoricalHighInjected	test_historical_high_injected_post_loop	Pass
	test_synthetic_coverage.py::TestHistoricalHighInjected	test_historical_high_not_in_range_skipped	Pass
	test_synthetic_coverage.py::TestHistoricalHighInjected	test_historical_high_date	Pass
	synthetic.py::TestGenerateSynthetic	test_all_levels_positive	Pass
	synthetic.py::TestGenerateSynthetic	test_correct_length_approximately	Pass
	synthetic.py::TestGenerateSynthetic	test_dates_are_sequential	Pass
	synthetic.py::TestGenerateSynthetic	test_different_seeds_different_output	Pass
	synthetic.py::TestGenerateSynthetic	test_each_obs_has_date_and_level	Pass
	synthetic.py::TestGenerateSynthetic	test_flat_gauge_data_uses_defaults	Pass
	synthetic.py::TestGenerateSynthetic	test_historical_high_date_in_range	Pass
	synthetic.py::TestGenerateSynthetic	test_historical_high_date_invalid_format	Pass
	synthetic.py::TestGenerateSynthetic	test_historical_high_date_out_of_range	Pass
	synthetic.py::TestGenerateSynthetic	test_historical_high_injected_when_not_placed	Pass
	synthetic.py::TestGenerateSynthetic	test_missing_flood_stages_uses_defaults	Pass
	synthetic.py::TestGenerateSynthetic	test_partial_tidal_gauge_produces_output	Pass
	synthetic.py::TestGenerateSynthetic	test_returns_list	Pass
	synthetic.py::TestGenerateSynthetic	test_seed_reproducible	Pass
	synthetic.py::TestGenerateSynthetic	test_tidal_gauge_produces_output	Pass
	synthetic.py::TestGenerateSynthetic	test_tidal_spring_neap_modulation	Pass
	test_gaugehd_module.py::TestGenerateAllGaugeHistories	test_generate_all_gauge_histories	Pass
	test_gaugehd_module.py::TestGenerateAllGaugeHistories	test_generate_all_gauge_histories_error_continues	Pass
	test_gaugehd_module.py::TestLoadGaugePortfolio	test_load_gauge_portfolio_file_not_found	Pass
	test_gaugehd_module.py::TestLoadGaugePortfolio	test_load_gauge_portfolio_success	Pass
	test_gaugehd_module.py::TestProcessNrfaDirectory	test_process_nrfa_directory_bad_csv_continues	Pass
	test_gaugehd_module.py::TestProcessNrfaDirectory	test_process_nrfa_directory_basic	Pass
	test_gaugehd_module.py::TestProcessNrfaDirectory	test_process_nrfa_directory_default_output	Pass
	test_gaugehd_module.py::TestProcessNrfaDirectory	test_process_nrfa_directory_empty_dir	Pass
	test_synthetic_gauge.py::TestSyntheticGauge	test_appends_to_gauge_json	Pass
	test_synthetic_gauge.py::TestSyntheticGauge	test_creates_synthetic_gauges	Pass
	test_synthetic_gauge.py::TestSyntheticGauge	test_deterministic	Pass
	test_synthetic_gauge.py::TestSyntheticGauge	test_generated_elevation	Pass
	test_synthetic_gauge.py::TestSyntheticGauge	test_generated_flood_stages	Pass
	test_synthetic_gauge.py::TestSyntheticGauge	test_gauge_json_returns_zero	Pass
	test_synthetic_gauge.py::TestSyntheticGauge	test_property_json_still_generates	Pass
	test_synthetic_gauge.py::TestSyntheticGauge	test_small_count_request	Pass
	test_synthetic_gauge.py::TestSyntheticGauge	test_synthetic_gauge_has_cdm_structure	Pass
	test_synthetic_gauge.py::TestSyntheticGauge	test_synthetic_gauge_on_river_centreline	Pass
	test_synthetic_gauge_coverage.py::TestDegenerateRiverSegment	test_degenerate_segment_skipped	Pass
	test_synthetic_gauge_coverage.py::TestFlaskGaugeMissing	test_flask_gauge_missing_gauge_returns_none	Pass
	test_synthetic_gauge_coverage.py::TestInsufficientGaugePoints	test_insufficient_gauge_points_returns_zero	Pass
	test_synthetic_gauge_coverage.py::TestInsufficientGaugePoints	test_insufficient_gauge_points_returns_zero	Pass
	test_synthetic_gauge_coverage.py::TestNoRiverPolyline	test_no_river_polyline_returns_none_when_no_cache	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_synthetic_gauge_coverage.py	TestNearestPolylineCoords::test_returns_original_when_no_polyline	Pass
	test_synthetic_gauge_coverage.py	TestPolylineEndpoints::test_includes_exact_endpoints	Pass
	test_synthetic_gauge_coverage.py	TestGenerateSyntheticGauges::test_generates_with_valid_property_json	Pass
	test_synthetic_gauge_coverage.py	TestRespectsCountParameter::test_respects_count_parameter	Pass
	test_synthetic_gauge_coverage.py	TestSyntheticGaugeSkipsUnmatched::test_skips_unmatched_to_gauge_point	Pass
	test_synthetic_gauge_lineage.py	TestPropertyHCWithSyntheticGauges::test_property_HCW_uses_synthetic_nearest_gauges	Pass
	test_synthetic_gauge_lineage.py	TestPropertyHCWithSyntheticGauges::test_property_HCW_synthetic_gauge_has_basis	Pass
	test_synthetic_gauge_lineage.py	TestPropertyTSCarryGauges::test_every_property_TSCarry_has_positive_synthetic_gauge	Pass
	test_synthetic_gauge_lineage.py	TestPropertyTSCarryGauges::test_every_property_TSCarry_has_positive_gauges_total	Pass
	test_synthetic_gauge_lineage.py	TestPropertyTSCarryGauges::test_every_property_TSCarry_has_positive_real_gauges	Pass
	test_synthetic_gauge_lineage.py	TestGaugeEstimates::test_gauge_estimates_reach_property_events	Pass
	test_synthetic_gauge_lineage.py	TestSomePropertiesFlood::test_some_properties_actually_flood	Pass
	test_synthetic_gauge_lineage.py	TestGaugeJsonHasSyntheticGauges::test_gauge_json_has_synthetic_gauges	Pass
	test_synthetic_gauge_lineage.py	TestSyntheticGaugeHasRequiredFields::test_synthetic_gauge_has_required_cdm_fields	Pass
	test_synthetic_gauge_lineage.py	TestGaugeHasSyntheticCurves::test_gauge_has_synthetic_curves	Pass
	test_synthetic_gauge_lineage.py	TestSyntheticGaugeInstructure::test_synthetic_gauge_in_structure	Pass
	test_synthetic_gauge_lineage.py	TestSyntheticGaugeHazardInCurves::test_synthetic_gauge_has_positive	Pass
	test_synthetic_gauge_lineage.py	TestSyntheticGaugesInClaims::test_synthetic_gauges_in_claims_exist	Pass
	test_synthetic_gauge_lineage.py	TestSyntheticGaugesInGauges::test_synthetic_gauges_have_exceeded_alerts	Pass
	test_synthetic_gauge_lineage.py	TestSyntheticGaugesInGauges::test_synthetic_gauges_have_storm_responses	Pass
MKM-IP-001	insurance.py::TestApplyInsurance	test_all_factors_applied	Pass
	insurance.py::TestApplyInsurance	test_basic_calculation	Pass
	insurance.py::TestApplyInsurance	test_claims_factor_increases_premium	Pass
	insurance.py::TestApplyInsurance	test_clamped_to_max	Pass
	insurance.py::TestApplyInsurance	test_clamped_to_min	Pass
	insurance.py::TestApplyInsurance	test_contents_premium_added	Pass
	insurance.py::TestApplyInsurance	test_high_flood_risk_increases_premium	Pass
	insurance.py::TestApplyInsurance	test_returns_float	Pass
	insurance.py::TestApplyInsurance	test_security_factor_reduces_premium	Pass
	insurance.py::TestGetAgePremium	test_100_years_is_heritage	Pass
	insurance.py::TestGetAgePremium	test_10_years_is_modern	Pass
	insurance.py::TestGetAgePremium	test_30_years_is_established	Pass
	insurance.py::TestGetAgePremium	test_default_current_year	Pass
	insurance.py::TestGetAgePremium	test_established	Pass
	insurance.py::TestGetAgePremium	test_exactly_29_years_modern	Pass
	insurance.py::TestGetAgePremium	test_exactly_99_years_established	Pass
	insurance.py::TestGetAgePremium	test_exactly_9_years_new_build	Pass
	insurance.py::TestGetAgePremium	test_heritage_old	Pass
	insurance.py::TestGetAgePremium	test_modern	Pass
	insurance.py::TestGetAgePremium	test_new_build	Pass
	insurance.py::TestGetAreaPremium	test_exactly_100_third_band	Pass
	insurance.py::TestGetAreaPremium	test_exactly_50_second_band	Pass
	insurance.py::TestGetAreaPremium	test_just_below_100_second_band	Pass
	insurance.py::TestGetAreaPremium	test_just_below_50_first_band	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	insurance.py::TestGetAreaPremiumFactor	test_large_area_fourth_band	Pass
	insurance.py::TestGetAreaPremiumFactor	test_medium_area_second_band	Pass
	insurance.py::TestGetAreaPremiumFactor	test_returns_tuple	Pass
	insurance.py::TestGetAreaPremiumFactor	test_small_area_first_band	Pass
	insurance.py::TestInsuranceFactorsTables	test_area_premium_factors_all_positive	Pass
	insurance.py::TestInsuranceFactorsTables	test_area_bands_are_sorted	Pass
	insurance.py::TestInsuranceFactorsTables	test_base_rate_range	Pass
	insurance.py::TestInsuranceFactorsTables	test_construction_type_factors_present	Pass
	insurance.py::TestInsuranceFactorsTables	test_flood_risk_factors_increase_with_risk	Pass
	insurance.py::TestInsuranceFactorsTables	test_flood_risk_factors_present	Pass
	insurance.py::TestInsuranceFactorsTables	test_min_max_premium_bounds	Pass
	insurance.py::TestInsuranceFactorsTables	test_property_type_factors_present	Pass
MKM-MP-001	credit.py::TestCreditSpread	test_custom_tax_rate	Pass
	credit.py::TestCreditSpread	test_floor_at_10bps	Pass
	credit.py::TestCreditSpread	test_higher_affordability_higher_spread	Pass
	credit.py::TestCreditSpread	test_negative_income_fallback	Pass
	credit.py::TestCreditSpread	test_spread_bounded_by_schedule	Pass
	credit.py::TestCreditSpread	test_zero_income_fallback	Pass
	credit.py::TestFloodRiskFactor	test_batch_passes_flood_risk	Pass
	credit.py::TestFloodRiskFactor	test_case_insensitive	Pass
	credit.py::TestFloodRiskFactor	test_flood_factor_in_result	Pass
	credit.py::TestFloodRiskFactor	test_flood_risk_increases_spread	Pass
	credit.py::TestFloodRiskFactor	test_flood_risk_multipliers[High-1.4]	Pass
	credit.py::TestFloodRiskFactor	test_flood_risk_multipliers[Low-1.05]	Pass
	credit.py::TestFloodRiskFactor	test_flood_risk_multipliers[Medium-1.2]	Pass
	credit.py::TestFloodRiskFactor	test_flood_risk_multipliers[Very High-1.75]	Pass
	credit.py::TestFloodRiskFactor	test_flood_risk_multipliers[Very Low-1.0]	Pass
	credit.py::TestFloodRiskFactor	test_no_flood_risk_defaults_to_unity	Pass
	credit.py::TestFloodRiskFactor	test_none_returns_unity	Pass
	credit.py::TestFloodRiskFactor	test_unknown_category_returns_unity	Pass
	credit.py::TestLTVRiskFactor	test_ltv_thresholds[0.5-1.0]	Pass
	credit.py::TestLTVRiskFactor	test_ltv_thresholds[0.8-1.0]	Pass
	credit.py::TestLTVRiskFactor	test_ltv_thresholds[0.85-1.1]	Pass
	credit.py::TestLTVRiskFactor	test_ltv_thresholds[0.9-1.1]	Pass
	credit.py::TestLTVRiskFactor	test_ltv_thresholds[0.91-1.3]	Pass
	credit.py::TestLTVRiskFactor	test_ltv_thresholds[0.95-1.3]	Pass
	credit.py::TestLTVRiskFactor	test_ltv_thresholds[0.96-1.5]	Pass
	credit.py::TestLTVRiskFactor	test_ltv_thresholds[1.0-1.5]	Pass
	credit.py::TestLTVRiskFactor	test_zero_property_value	Pass
	payment.py::TestMonthlyPayment	test_higher_rate_higher_payment	Pass
	payment.py::TestMonthlyPayment	test_shorter_term_higher_payment	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	payment.py::TestMonthlyPayment	test_standard_mortgage	Pass
	payment.py::TestMonthlyPayment	test_zero_rate	Pass
	payment.py::TestTotalCost	test_positive_rate_exceeds_principal	Pass
	payment.py::TestTotalCost	test_zero_rate_equals_principal	Pass
	pricing.py::TestBatchPricing	test_default_ids_assigned	Pass
	pricing.py::TestBatchPricing	test_error_isolation	Pass
	pricing.py::TestBatchPricing	test_prices_multiple_mortgages	Pass
	pricing.py::TestConstructor	test_custom_tax_rate	Pass
	pricing.py::TestConstructor	test_default_tax_rate	Pass
	pricing.py::TestDebugAndEdgeCases	test_debug_also_in_credit_spread	Pass
	pricing.py::TestDebugAndEdgeCases	test_debug_true_does_not_crash	Pass
	pricing.py::TestDebugAndEdgeCases	test_debug_with_zero_income	Pass
	pricing.py::TestDebugAndEdgeCases	test_flood_risk_none_with_debug	Pass
	pricing.py::TestDebugAndEdgeCases	test_zero_interest_rate	Pass
	pricing.py::TestPortfolioMetrics	test_all_errors_returns_error	Pass
	pricing.py::TestPortfolioMetrics	test_high_risk_count	Pass
	pricing.py::TestPortfolioMetrics	test_mixed_results_exclude_errors	Pass
	pricing.py::TestPortfolioMetrics	test_valid_results	Pass
	pricing.py::TestPriceMortgage	test_balance_converges_to_zero	Pass
	pricing.py::TestPriceMortgage	test_expected_losses_non_negative	Pass
	pricing.py::TestPriceMortgage	test_fair_value_less_than_par	Pass
	pricing.py::TestPriceMortgage	test_hazard_rates_positive	Pass
	pricing.py::TestPriceMortgage	test_high_ltv_increases_spread	Pass
	pricing.py::TestPriceMortgage	test_higher_haircut_more_loss	Pass
	pricing.py::TestPriceMortgage	test_input_validation_clamps	Pass
	pricing.py::TestPriceMortgage	test_ltv_factor_in_result	Pass
	pricing.py::TestPriceMortgage	test_raises_on_invalid_types	Pass
	pricing.py::TestPriceMortgage	test_recovery_haircut_zero_low_ltv	Pass
	pricing.py::TestPriceMortgage	test_survival_first_period_less_than_one	Pass
	pricing.py::TestPriceMortgage	test_survival_monotonically_decreasing	Pass
	mortgage_generator.py::TestMortgageGenerator	test_each_mortgage_has_property_id	Pass
	mortgage_generator.py::TestMortgageGenerator	test_processing_status	Pass
	mortgage_generator.py::TestMortgageGenerator	test_output_file_exists	Pass
	mortgage_generator.py::TestMortgageGenerator	test_output_file_is_valid_json	Pass
	mortgage_generator.py::TestMortgageGenerator	test_generate_returns_expected_keys	Pass
	mortgage_generator.py::TestMortgageGenerator	test_mortgage_count_matches_properties	Pass
	mortgage_generator.py::TestMortgageGenerator	test_mortgage_id_format	Pass
	mortgage_generator.py::TestMortgageGenerator	test_mortgage_ids_unique	Pass
MKM-PR-001	maturity.py::TestComputeMaturityDate	test_1y_from_feb_2026	Pass
	maturity.py::TestComputeMaturityDate	test_3y_from_feb_2026	Pass
	maturity.py::TestComputeMaturityDate	test_3y_from_jan_2027	Pass
	maturity.py::TestComputeMaturityDate	test_3y_from_sep_2026	Pass
	maturity.py::TestComputeMaturityDate	test_5y_from_feb_2026	Pass
	maturity.py::TestComputeMaturityDate	test_5y_from_sep_2026	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	maturity.py::TestComputeMaturityDate	test_maturity_is_friday	Pass
	maturity.py::TestComputeMaturityDate	test_roll_jump_at_august	Pass
	maturity.py::TestCurrentRollDate	test_aug_period	Pass
	maturity.py::TestCurrentRollDate	test_dec_uses_aug	Pass
	maturity.py::TestCurrentRollDate	test_feb_period	Pass
	maturity.py::TestCurrentRollDate	test_jan_uses_prev_aug	Pass
	maturity.py::TestCurrentRollDate	test_jul_still_feb_roll	Pass
	maturity.py::TestCurrentRollDate	test_mar_still_feb_roll	Pass
	maturity.py::TestDefaultDates	test_compute_maturity_default	Pass
	maturity.py::TestDefaultDates	test_current_roll_date_default	Pass
	maturity.py::TestDefaultDates	test_maturity_table_default	Pass
	maturity.py::TestLastFridayOfMonth	test_aug_2026	Pass
	maturity.py::TestLastFridayOfMonth	test_feb_2026	Pass
	maturity.py::TestLastFridayOfMonth	test_may_2029	Pass
	maturity.py::TestLastFridayOfMonth	test_nov_2029	Pass
	maturity.py::TestLastFridayOfMonth	test_result_is_friday	Pass
	maturity.py::TestMaturityTable	test_actual_years_increasing	Pass
	maturity.py::TestMaturityTable	test_all_may_for_feb_roll	Pass
	maturity.py::TestMaturityTable	test_all_nov_for_aug_roll	Pass
	maturity.py::TestMaturityTable	test_maturity_str_format	Pass
	maturity.py::TestMaturityTable	test_table_has_5_entries	Pass
	maturity.py::TestMaturityTable	test_tenors_sequential	Pass
	maturity.py::TestTenorFromMaturity	test_all_tenors_round_trip	Pass
	maturity.py::TestTenorFromMaturity	test_closest_match_fallback	Pass
	maturity.py::TestTenorFromMaturity	test_exact_match	Pass
	basis.py::TestEdgeCases	test_empty_propertyts_dir	Pass
	basis.py::TestEdgeCases	test_no_gauge_hazard_curves	Pass
	basis.py::TestEdgeCases	test_no_propertyts_dir	Pass
	basis.py::TestSpreadDecomposition	test_decomposition_attached_after_post_processing	Pass
	basis.py::TestSpreadDecomposition	test_decomposition_effects_not_all_zero	Pass
	basis.py::TestSpreadDecomposition	test_decomposition_paths_sum_correctly	Pass
	basis.py::TestSpreadDecomposition	test_decomposition_values_nonzero_with_synthetic_files	Pass
	basis.py::TestSpreadDecomposition	test_decomposition_without_synthetic_files	Pass
	basis.py::TestSpreadDecomposition	test_get_5yr_spread_returns_correct_value	Pass
	basis.py::TestSpreadDecomposition	test_get_5yr_spread_returns_zero_for_empty	Pass
	basis.py::TestSpreadDecomposition	test_get_5yr_spread_with_threshold	Pass
	generator.py::TestPropertyHazardCurveGenerator	test_basis_calculation	Pass
	generator.py::TestPropertyHazardCurveGenerator	test_depth_thresholds	Pass
	generator.py::TestPropertyHazardCurveGenerator	test_generate_produces_output	Pass
	generator.py::TestPropertyHazardCurveGenerator	test_get_params_always_none	Pass
	generator.py::TestPropertyHazardCurveGenerator	test_high_transmission_rate	Pass
	generator.py::TestPropertyHazardCurveGenerator	test_output_structure	Pass
	generator.py::TestPropertyHazardCurveGenerator	test_properties_processed	Pass
	generator.py::TestPropertyHazardCurveGenerator	test_spread_is_flat_across_tenors	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	generator.py::TestPropertyHazardCurveGenerator	test_curve_structure	Pass
	pricing.py::TestEventCountSpread	test_spread_is_flat_term_structure	Pass
	pricing.py::TestEventCountSpread	test_spread_proportional_to_floods	Pass
	pricing.py::TestEventCountSpread	test_zero_floods_zero_spread	Pass
	pricing.py::TestGetGaugeSeverecount	test_returns_severe_event_count	Pass
	pricing.py::TestGetGaugeSeverecount	test_returns_zero_for_empty_dict	Pass
	pricing.py::TestGetGaugeSeverecount	test_returns_zero_when_missing	Pass
MKM-PV-001	property_generator.py::TestBuildSection	test_non_dict_schema_returns_empty	Pass
	property_generator.py::TestBuildSection	test_skips_type_field	Pass
	property_generator.py::TestDateTimeEncoder	test_datetime_to_isoformat	Pass
	property_generator.py::TestDateTimeEncoder	test_numpy_floating	Pass
	property_generator.py::TestDateTimeEncoder	test_numpy_integer	Pass
	property_generator.py::TestDateTimeEncoder	test_numpy_ndarray	Pass
	property_generator.py::TestDateTimeEncoder	test_unknown_type_raises	Pass
	property_generator.py::TestGenerateFallbacks	test_fallback_allocation_has_required_keys	Pass
	property_generator.py::TestGenerateFallbacks	test_fallback_returns_correct_count	Pass
	property_generator.py::TestGenerateFallbacks	test_fallback_with_target_elevation	Pass
	property_generator.py::TestGenerateProperties	test_properties_basic	Pass
	property_generator.py::TestGenerateProperties	test_properties_output_file_created	Pass
	property_generator.py::TestPropertyCoordinates	test_coordinates_within_bounds	Pass
	property_generator.py::TestPropertyDataQuality	test_quality_flags_range	Pass
	property_generator.py::TestPropertyDataQuality	test_processing_stats	Pass
	property_generator.py::TestPropertyGenerator	test_output_file_exists	Pass
	property_generator.py::TestPropertyGenerator	test_output_file_is_valid_json	Pass
	property_generator.py::TestPropertyGenerator	test_generate_correct_count	Pass
	property_generator.py::TestPropertyGenerator	test_generate_returns_expected_keys	Pass
	property_generator.py::TestPropertyGenerator	test_property_id_format	Pass
	property_generator.py::TestPropertyGenerator	test_property_ids_unique	Pass
	property_generator.py::TestPropertyUtils	test_generate_grid_reference_format	Pass
	property_generator.py::TestPropertyUtils	test_generate_owner_name_two_words	Pass
	property_generator.py::TestPropertyUtils	test_generate_past_date_format	Pass
	property_generator.py::TestPropertyUtils	test_generate_past_date_range	Pass
	rand_property_mortgage.py::TestCalculatePropertyInsurance	test_calculate_insurance_premium_flood_risk_high	Pass
	rand_property_mortgage.py::TestCalculatePropertyInsurance	test_calculate_insurance_premium_no_area	Pass
	rand_property_mortgage.py::TestCalculatePropertyInsurance	test_calculate_insurance_premium_no_value	Pass
	rand_property_mortgage.py::TestCalculateMonthlyPayment	test_calculate_monthly_payment_all_types	Pass
	rand_property_mortgage.py::TestCalculateMonthlyPayment	test_calculate_monthly_payment_no_area	Pass
	rand_property_mortgage.py::TestCalculateMonthlyPayment	test_calculate_monthly_payment_no_value	Pass
	rand_property_mortgage.py::TestCalculateMonthlyPayment	test_calculate_monthly_payment_positive	Pass
	rand_property_mortgage.py::TestCalculatePropertyArea	test_calculate_property_area_all_types	Pass
	rand_property_mortgage.py::TestCalculateSalePrice	test_calculate_sale_price_invalid_date_falls_back	Pass
	rand_property_mortgage.py::TestCalculateSalePrice	test_calculate_sale_price_no_current_value	Pass
	rand_property_mortgage.py::TestCalculateSalePrice	test_calculate_sale_price_positive	Pass
	rand_property_mortgage.py::TestCalculateSalePrice	test_calculate_sale_price_with_sale_date_1_to_2_years	Pass
	rand_property_mortgage.py::TestCalculateSalePrice	test_calculate_sale_price_with_sale_date_over_2_years	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	rand_property_mortgage.py::TestCalculatePropertyValue	test_calculate_property_value_with_sale_date_recent	Pass
	rand_property_mortgage.py::TestGenerateMortgage	test_generate_mortgage_financials_returns_dict	Pass
	rand_property_mortgage.py::TestMortgageFinancials	test_mortgage_financial_data_keys	Pass
	rand_property_mortgage.py::TestMortgageFinancials	test_mortgage_first_date_range	Pass
	rand_property_mortgage.py::TestMortgageFinancials	test_mortgage_amount_less_than_value	Pass
	rand_property_mortgage.py::TestMortgageFinancials	test_mortgage_id_format	Pass
	rand_property_mortgage.py::TestMortgageFinancials	test_mortgage_financial_consistency_check	Pass
	rand_property_mortgage.py::TestPropertyCalculations	test_calculate_insurance_premium_positive	Pass
	rand_property_mortgage.py::TestPropertyCalculations	test_calculate_property_area_positive	Pass
	rand_property_mortgage.py::TestPropertyCalculations	test_calculate_property_value_in_range	Pass
	rand_property_mortgage.py::TestPropertyGeneration	test_generate_bedrooms_detached	Pass
	rand_property_mortgage.py::TestPropertyGeneration	test_generate_bedrooms_flat	Pass
	rand_property_mortgage.py::TestPropertyGeneration	test_generate_postcode_format	Pass
	rand_property_mortgage.py::TestPropertyGeneration	test_generate_field_value_decimal	Pass
	rand_property_mortgage.py::TestPropertyGeneration	test_generate_field_value_text	Pass
	rand_property_mortgage.py::TestPropertyGeneration	test_construction_year_in_range	Pass
	rand_property_mortgage.py::TestPropertyGeneration	test_property_id_format	Pass
	rand_property_mortgage.py::TestPropertyGeneration	test_returns_dict_with_required_keys	Pass
MKM-SG-001	gauge_generator.py::TestDateTimeEncoder	test_encodes_datetime	Pass
	gauge_generator.py::TestDateTimeEncoder	test_encodes_numpy_float32	Pass
	gauge_generator.py::TestDateTimeEncoder	test_encodes_numpy_floating	Pass
	gauge_generator.py::TestDateTimeEncoder	test_encodes_numpy_integer	Pass
	gauge_generator.py::TestDateTimeEncoder	test_encodes_numpy_ndarray	Pass
	gauge_generator.py::TestDateTimeEncoder	test_fallback_raises_for_unknown	Pass
	gauge_generator.py::TestGaugeGenerator	test_formatDataQuality_ordering	Pass
	gauge_generator.py::TestGaugeGenerator	test_processQuality_stats	Pass
	gauge_generator.py::TestGaugeGenerator	test_save_exception_propagates	Pass
	gauge_generator.py::TestGaugeGenerator	test_secondaryGenerate_hits_end_time_branch	Pass
	gauge_generator.py::TestGaugeGenerator	test_buildSection_non_dict_schema_returns_empty	Pass
	gauge_generator.py::TestGaugeGenerator	test_buildSection_skips_metadata_keys	Pass
	gauge_generator.py::TestGaugeGenerator	test_computeEdges_exceeds_available_points_capped	Pass
	gauge_generator.py::TestGaugeGenerator	test_computeEdges_zero_raises_value_error	Pass
	gauge_generator.py::TestGaugeGenerator	test_generateEdges_gauges_convenience_function	Pass
	gauge_generator.py::TestGaugeGenerator	test_generateEdges_single_gauge_failure_counted	Pass
	gauge_generator.py::TestGaugeGenerator	test_logEdgeMethod_works	Pass
	gauge_generator.py::TestGaugeGenerator	test_setSpecifics_values_creates_missing_sections	Pass
	gauge_generator.py::TestGaugeGenerator	test_verbosityEdgeCase_sets_warning_level	Pass
	gauge_generator.py::TestGaugeGenerator	test_output_file_exists	Pass
	gauge_generator.py::TestGaugeGenerator	test_output_file_is_valid_json	Pass
	gauge_generator.py::TestGaugeGenerator	test_output_json_has_correct_count	Pass
	gauge_generator.py::TestGaugeGenerator	test_main_block	Pass
	gauge_generator.py::TestGaugeGenerator	test_gauge_id_format	Pass
	gauge_generator.py::TestGaugeGenerator	test_gauge_ids_unique	Pass
	gauge_generator.py::TestGaugeGenerator	test_output_correct_count	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	gauge_generator.py::TestGaugeGenerator	test_generate_returns_expected_keys	Pass
	gauge_generator.py::TestGaugeGenerator	test_locations_have_coordinates	Pass
	rand_gauge.py::TestFloodRiskAssessment	test_close_to_thames_high_risk	Pass
	rand_gauge.py::TestFloodRiskAssessment	test_far_from_thames_lower_risk	Pass
	rand_gauge.py::TestGenerateFieldValue	test_date_field_returns_date_string	Pass
	rand_gauge.py::TestGenerateFieldValue	test_decimal_field_returns_float	Pass
	rand_gauge.py::TestGenerateFieldValue	test_gauge_name_includes_area	Pass
	rand_gauge.py::TestGenerateFieldValue	test_menu_field_returns_valid_option	Pass
	rand_gauge.py::TestGenerateFieldValue	test_text_field_returns_string	Pass
	rand_gauge.py::TestGenerateGaugeVector	test_attachment_name_stored	Pass
	rand_gauge.py::TestGenerateGaugeVector	test_flood_stage_ordering	Pass
	rand_gauge.py::TestGenerateGaugeVector	test_gauge_id_format	Pass
	rand_gauge.py::TestGenerateGaugeVector	test_historical_high_level_range	Pass
	rand_gauge.py::TestGenerateGaugeVector	test_install_date_format	Pass
	rand_gauge.py::TestGenerateGaugeVector	test_returns_dict_with_required_keys	Pass
MKM-SI-001	storm_duration.py::TestDurationParams	test_all_categories_defined	Pass
	storm_duration.py::TestDurationParams	test_catastrophic_extended	Pass
	storm_duration.py::TestDurationParams	test_each_tier_doubles_approx	Pass
	storm_duration.py::TestDurationParams	test_extreme_extended	Pass
	storm_duration.py::TestDurationParams	test_moderate_extended	Pass
	storm_duration.py::TestDurationParams	test_severe_extended	Pass
	storm_duration.py::TestHelpers	test_get_duration_range	Pass
	storm_duration.py::TestHelpers	test_get_max_duration	Pass
	storm_duration.py::TestSampleDuration	test_invalid_category	Pass
	storm_duration.py::TestSampleDuration	test_mode_is_most_common	Pass
	storm_duration.py::TestSampleDuration	test_no_rng_uses_default	Pass
	storm_duration.py::TestSampleDuration	test_reproducibility	Pass
	storm_duration.py::TestSampleDuration	test_within_bounds[baseline]	Pass
	storm_duration.py::TestSampleDuration	test_within_bounds[catastrophic]	Pass
	storm_duration.py::TestSampleDuration	test_within_bounds[extreme]	Pass
	storm_duration.py::TestSampleDuration	test_within_bounds[minimal]	Pass
	storm_duration.py::TestSampleDuration	test_within_bounds[moderate]	Pass
	storm_duration.py::TestSampleDuration	test_within_bounds[severe]	Pass
	storm_gap.py::TestFitsInWindow	test_constants	Pass
	storm_gap.py::TestFitsInWindow	test_exact_boundary	Pass
	storm_gap.py::TestFitsInWindow	test_exceeds_window	Pass
	storm_gap.py::TestFitsInWindow	test_simple_doublet_fits	Pass
	storm_gap.py::TestFitsInWindow	test_single_storm	Pass
	storm_gap.py::TestGapParams	test_long_gaps	Pass
	storm_gap.py::TestGapParams	test_medium_gaps	Pass
	storm_gap.py::TestGapParams	test_short_gaps	Pass
	storm_gap.py::TestGetGapRange	test_doublet_medium	Pass
	storm_gap.py::TestGetGapRange	test_isolated_zero	Pass
	storm_gap.py::TestSampleGap	test_doublet_uses_medium	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	storm_gap.py::TestSampleGap	test_isolated_returns_zero	Pass
	storm_gap.py::TestSampleGap	test_no_rng_uses_default	Pass
	storm_gap.py::TestSampleGap	test_persistent_uses_short	Pass
	storm_gap.py::TestSampleGap	test_reproducibility	Pass
	storm_gap.py::TestSampleGap	test_within_bounds[SequenceType.CLUSTER]	Pass
	storm_gap.py::TestSampleGap	test_within_bounds[SequenceType.DOUBLET]	Pass
	storm_gap.py::TestSampleGap	test_within_bounds[SequenceType.PERSISTENT]	Pass
	storm_gap.py::TestValidation	test_duration_mismatch	Pass
	storm_gap.py::TestValidation	test_exceeds_precip_window	Pass
	storm_gap.py::TestValidation	test_invalid_intensity_category	Pass
	storm_gap.py::TestValidation	test_invalid_sequence_type	Pass
	storm_gap.py::TestValidation	test_overlapping_storms	Pass
	storm_gap.py::TestValidation	test_storm_count_mismatch	Pass
	storm_gap.py::TestValidation	test_too_many_storms	Pass
	storm_gap.py::TestValidation	test_valid_doublet	Pass
	storm_sequence.py::TestEnums	test_gap_types	Pass
	storm_sequence.py::TestEnums	test_sequence_gap_mapping	Pass
	storm_sequence.py::TestEnums	test_sequence_types	Pass
	storm_sequence.py::TestEnums	test_storm_count_ranges	Pass
	storm_sequence.py::TestIDGeneration	test_storm_ids_unique	Pass
	storm_sequence.py::TestIDGeneration	test_sequence_id_format	Pass
	storm_sequence.py::TestIDGeneration	test_sequence_ids_unique	Pass
	storm_sequence.py::TestIDGeneration	test_storm_id_format	Pass
	storm_sequence.py::TestSequenceStorm	test_create_basic	Pass
	storm_sequence.py::TestSequenceStorm	test_from_dict_round_trip	Pass
	storm_sequence.py::TestSequenceStorm	test_precipitation_rounding	Pass
	storm_sequence.py::TestSequenceStorm	test_to_dict	Pass
	storm_sequence.py::TestStormSequence	test_antecedent_defaults	Pass
	storm_sequence.py::TestStormSequence	test_create_doublet	Pass
	storm_sequence.py::TestStormSequence	test_from_dict_round_trip	Pass
	storm_sequence.py::TestStormSequence	test_isolated_no_gaps	Pass
	storm_sequence.py::TestStormSequence	test_spatial_correlation_defaults	Pass
	storm_sequence.py::TestStormSequence	test_spatial_correlation_enabled	Pass
	storm_sequence.py::TestStormSequence	test_to_dict	Pass
MKM-SP-001	spatial.py::TestBuildCorrelationMatrix	test_diagonal_is_one	Pass
	spatial.py::TestBuildCorrelationMatrix	test_off_diagonal_between_zero_and_one	Pass
	spatial.py::TestBuildCorrelationMatrix	test_shape_is_n_by_n	Pass
	spatial.py::TestBuildCorrelationMatrix	test_symmetric	Pass
	spatial.py::TestBuildSpatialIndex	test_empty_dataframe_returns_none	Pass
	spatial.py::TestBuildSpatialIndex	test_none_gauge_data_returns_none	Pass
	spatial.py::TestBuildSpatialIndex	test_with_gauges_returns_kdtree	Pass
	spatial.py::TestHaversineDistance	test_equatorial_degree_approx_111km	Pass
	spatial.py::TestHaversineDistance	test_london_paris_approx_340km	Pass
	spatial.py::TestHaversineDistance	test_north_south_degree_approx_111km	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	spatial.py::TestHaversineDistance	test_returns_float	Pass
	spatial.py::TestHaversineDistance	test_same_point_is_zero	Pass
	spatial.py::TestHaversineDistance	test_symmetry	Pass
	spatial.py::TestIDWInterpolate	test_all_zero_weights_returns_zero	Pass
	spatial.py::TestIDWInterpolate	test_closer_gauge_has_higher_weight	Pass
	spatial.py::TestIDWInterpolate	test_empty_list_returns_zero	Pass
	spatial.py::TestIDWInterpolate	test_equal_distance_gauges_averages_wse	Pass
	spatial.py::TestIDWInterpolate	test_near_gauge_returns_wse_directly	Pass
	spatial.py::TestIDWInterpolate	test_power_one_changes_result	Pass
	spatial.py::TestIDWInterpolate	test_single_far_gauge	Pass
	spatial.py::TestSpatialInterpolateWSE	test_at_gauge_location_returns_wse_directly	Pass
	spatial.py::TestSpatialInterpolateWSE	test_empty_gauge_data_returns_zero	Pass
	spatial.py::TestSpatialInterpolateWSE	test_idw_returns_weighted_value	Pass
	spatial.py::TestSpatialInterpolateWSE	test_single_gauge	Pass
	test_spatial_coverage.py::TestBuildCorrelationMatrixCoverage	test_diagonal_is_one	Pass
	test_spatial_coverage.py::TestBuildCorrelationMatrixCoverage	test_higher_base_correlation_increases_off_diagonal	Pass
	test_spatial_coverage.py::TestBuildCorrelationMatrixCoverage	test_larger_distance_increases_correlation	Pass
	test_spatial_coverage.py::TestBuildCorrelationMatrixCoverage	test_off_diagonal_positive	Pass
	test_spatial_coverage.py::TestBuildCorrelationMatrixCoverage	test_returns_square_matrix	Pass
	test_spatial_coverage.py::TestBuildCorrelationMatrixCoverage	test_symmetric	Pass
	test_spatial_coverage.py::TestBuildSpatialIndexCoverage	test_kdtree_coords_are_radians	Pass
	test_spatial_coverage.py::TestBuildSpatialIndexCoverage	test_single_gauge_builds_tree	Pass
	test_spatial_coverage.py::TestBuildSpatialIndexCoverage	test_valid_dataframe_returns_kdtree	Pass
	test_spatial_coverage.py::TestIDWInterpolateEdgeCases	test_custom_power	Pass
	test_spatial_coverage.py::TestIDWInterpolateEdgeCases	test_empty_returns_zero	Pass
	test_spatial_coverage.py::TestIDWInterpolateEdgeCases	test_single_gauge_at_exact_min_distance	Pass
	test_spatial_coverage.py::TestIDWInterpolateEdgeCases	test_three_gauges_weighted	Pass
	test_spatial_coverage.py::TestIDWInterpolateEdgeCases	test_very_close_gauge_returns_directly	Pass
	test_spatial_coverage.py::TestSpatialInterpolateWSEEdgeCases	test_at_exact_gauge_location_returns_wse	Pass
	test_spatial_coverage.py::TestSpatialInterpolateWSEEdgeCases	test_empty_dfs_returns_zero	Pass
	test_spatial_coverage.py::TestSpatialInterpolateWSEEdgeCases	test_two_gauges_weighted_average	Pass
MKM-SS-001	sequence_gen.py::TestBuildIsolated	test_isolated_duration_clamped	Pass
	sequence_gen.py::TestBuildIsolated	test_isolated_has_one_storm	Pass
	sequence_gen.py::TestSequenceGenerator	test_aggregate_stats	Pass
	sequence_gen.py::TestSequenceGenerator	test_catchment_id_in_storms	Pass
	sequence_gen.py::TestSequenceGenerator	test_cluster_sequence	Pass
	sequence_gen.py::TestSequenceGenerator	test_combined_force	Pass
	sequence_gen.py::TestSequenceGenerator	test_isolated_force	Pass
	sequence_gen.py::TestSequenceGenerator	test_minimum_category_often_isolated	Pass
	sequence_gen.py::TestSequenceGenerator	test_multiple_calls	Pass
	sequence_gen.py::TestSequenceGenerator	test_persistent_sequence	Pass
	sequence_gen.py::TestSequenceGenerator	test_returns_storm_sequence	Pass
	sequence_gen.py::TestSequenceGenerator	test_sequence_fits_in_window	Pass
	sequence_gen.py::TestSequenceGenerator	test_storms_have_correct_fields	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	sequence_gen.py::TestSequenceGenerator	test_unknown_catchment	Pass
	sequence_gen.py::TestSequenceGenerator	test_default_catchment	Pass
	sequence_gen.py::TestSequenceGenerator	test_seed_reproducibility	Pass
	sequence_gen.py::TestSequenceGenerator	test_unknown_catchment_defaults_to_35	Pass
MKM-ST-001	test_batch_train.py::TestBatchTrainCli	test_classifier_already_trained	Pass
	test_batch_train.py::TestBatchTrainCli	test_classifier_prints_after_first_gauge	Pass
	test_batch_train.py::TestBatchTrainCli	test_missing_gauge_json	Pass
	test_batch_train.py::TestBatchTrainCli	test_missing_sequences	Pass
	test_batch_train.py::TestBatchTrainCli	test_no_valid_gauges	Pass
	test_batch_train.py::TestBatchTrainCli	test_partial_training	Pass
	test_batch_train.py::TestBatchTrainCli	test_successful_training	Pass
	test_batch_train.py::TestBatchTrainCli	test_training_failure_is_caught	Pass
	test_batch_train.py::TestUpdateTrainingSummary	test_creates_new_summary	Pass
	test_batch_train.py::TestUpdateTrainingSummary	test_merges_into_existing	Pass
	test_batch_train.py::TestUpdateTrainingSummary	test_replaces_existing_gauge	Pass
	test_batch_train.py::TestUpdateTrainingSummary	test_skipped_gauge_not_counted_as_trained	Pass
	test_gauge_parser.py::TestLoadGaugeBaselines	test_corrupt_file_skipped	Pass
	test_gauge_parser.py::TestLoadGaugeBaselines	test_empty_directory	Pass
	test_gauge_parser.py::TestLoadGaugeBaselines	test_missing_directory	Pass
	test_gauge_parser.py::TestLoadGaugeBaselines	test_single_mean_level_skipped	Pass
	test_gauge_parser.py::TestLoadGaugeBaselines	test_single_gauge	Pass
	test_gauge_parser.py::TestLoadGaugeBaselines	test_valid_gauge_files	Pass
	test_gauge_parser.py::TestParseGaugeWithBaseline	test_without_monthly_means	Pass
	test_gauge_parser.py::TestParseGaugeWithBaseline	test_with_baseline	Pass
	test_gauge_parser.py::TestParseGaugeWithBaseline	test_without_baseline_uses_heuristic	Pass
	test_gauge_parser.py::TestSeasonalBaseline	test_empty_monthly_means	Pass
	test_gauge_parser.py::TestSeasonalBaseline	test_missing_month_falls_back	Pass
	test_gauge_parser.py::TestSeasonalBaseline	test_no_monthly_means	Pass
	test_gauge_parser.py::TestSeasonalBaseline	test_with_monthly_means	Pass
	test_generate.py::TestGenerateStreamflowFiles	test_gauge_summary_dir_written	Pass
	test_generate.py::TestGenerateStreamflowFiles	test_gauge_summary_gauge_ids	Pass
	test_generate.py::TestGenerateStreamflowFiles	test_gauge_summary_num_gauges	Pass
	test_generate.py::TestGenerateStreamflowFiles	test_gauge_summary_schema_version	Pass
	test_generate.py::TestGenerateStreamflowFiles	test_gauge_summary_sequence_count	Pass
	test_generate.py::TestGenerateStreamflowFiles	test_sequences_file_loadable	Pass
	test_generate.py::TestGenerateStreamflowFiles	test_sequences_json_written	Pass
	test_generate.py::TestGenerateStreamflowFiles	test_summary_json_written	Pass
	test_generate.py::TestGenerateStreamflowFiles	test_sent_count_non_negative	Pass
	test_generate.py::TestGenerateStreamflowFiles	test_skipped_positive	Pass
	test_generate.py::TestGenerateStreamflowFiles	test_summary_gauges	Pass
	test_generate.py::TestGenerateStreamflowFiles	test_summary_sequences	Pass
	test_generate.py::TestGenerateStreamflowFiles	test_summary_dict	Pass
	test_generate.py::TestGenerateStreamflowFiles	test_severe_alt_warning	Pass
	test_generate.py::TestGenerateStreamflowFiles	test_type_counts_present	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_generate.py::TestGenerateStreamingSummary	test_warning_alert	Pass
	test_generate.py::TestMissingGaugeJson	test_no_crash_when_gauge_json_absent	Pass
	test_generate.py::TestMissingGaugeJson	test_num_gauges_zero_without_gauge_json	Pass
	test_generate.py::TestMissingGaugeJson	test_sequences_still_written_without_gauge_json	Pass
	test_generate.py::TestNoValidGauges	test_returns_gracefully_when_all_gauges_invalid	Pass
	test_generate.py::TestNoValidGauges	test_sequences_still_written_when_all_gauges_invalid	Pass
	test_generate.py::TestVerboseProgress	test_verbose_false_no_mid_loop_output_at_small_count	Pass
	test_generate.py::TestVerboseProgress	test_verbose_run_completes	Pass
	test_helpers.py::TestBuildSummary	test_counts_default_to_zero	Pass
	test_helpers.py::TestBuildSummary	test_num_gauges	Pass
	test_helpers.py::TestBuildSummary	test_num_sequences	Pass
	test_helpers.py::TestBuildSummary	test_returns_required_keys	Pass
	test_helpers.py::TestExtractGauges	test_dict_format	Pass
	test_helpers.py::TestExtractGauges	test_empty_list	Pass
	test_helpers.py::TestExtractGauges	test_list_format	Pass
	test_helpers.py::TestExtractGauges	test_missing_key_returns_empty	Pass
	test_helpers.py::TestParseGauge	test_base_level_derived_from_alert	Pass
	test_helpers.py::TestParseGauge	test_flood_alert	Pass
	test_helpers.py::TestParseGauge	test_flood_warning	Pass
	test_helpers.py::TestParseGauge	test_gauge_id	Pass
	test_helpers.py::TestParseGauge	test_lat_lon	Pass
	test_helpers.py::TestParseGauge	test_missing_alert_returns_none	Pass
	test_helpers.py::TestParseGauge	test_missing_gauge_id_returns_none	Pass
	test_helpers.py::TestParseGauge	test_missing_lat_returns_none	Pass
	test_helpers.py::TestParseGauge	test_severe_warning	Pass
	test_helpers.py::TestParseGauge	test_valid_record_returns_dict	Pass
	test_helpers.py::TestParseGauge	test_warning_defaults_when_absent	Pass
	test_helpers.py::TestStormResponseFields	test_base_level_matches_gauge_params	Pass
	test_helpers.py::TestStormResponseFields	test_gauge_a_values	Pass
	test_helpers.py::TestStormResponseFields	test_gauge_b_values	Pass
	test_helpers.py::TestStormResponseFields	test_level_change_is_peak_minus_base	Pass
	test_helpers.py::TestStormResponseFields	test_response_has_base_level_m	Pass
	test_helpers.py::TestStormResponseFields	test_response_has_level_change_m	Pass
	test_pipeline_coverage.py::TestClassifiedTasksPerFailureMode	test_failure_does_not_crash_pipeline	Pass
	test_pipeline_coverage.py::TestGaugeBaselines	test_baselines_loaded_from_gaugehd	Pass
	test_pipeline_coverage.py::TestLegacyFileCleanup	test_legacy_summary_file_removed	Pass
	test_pipeline_coverage.py::TestMalformedSequenceStart	test_defaults_to_month_1	Pass
	test_pipeline_coverage.py::TestMalformedSequenceStart	test_month_sequence_start_handled	Pass
	test_pipeline_coverage.py::TestNonVerboseProgress	test_no_progress_eta_line_at_2000	Pass
	test_records.py::TestIntensityCategory	test_batch_generator_populates_category	Pass
	test_records.py::TestIntensityCategory	test_from_dict_missing_category_defaults_empty	Pass
	test_records.py::TestIntensityCategory	test_from_dict_round_trip	Pass
	test_records.py::TestIntensityCategory	test_serialized_sequences_have_category	Pass
	test_records.py::TestIntensityCategory	test_to_dict_includes_intensity_category	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_records.py::TestSequenceRecordsSchema	test_all_have_intensity_category	Pass
	test_records.py::TestSequenceRecordsSchema	test_all_have_sequence_id	Pass
	test_records.py::TestSequenceRecordsSchema	test_all_have_sequence_type	Pass
	test_records.py::TestSequenceRecordsSchema	test_all_have_total_precip	Pass
	test_records.py::TestSequenceRecordsSchema	test_gauge_file_count_matches_num_gauges	Pass
	test_records.py::TestSequenceRecordsSchema	test_peak_hours_in_valid_range	Pass
	test_records.py::TestSequenceRecordsSchema	test_peaks_positive	Pass
	test_records.py::TestSequenceRecordsSchema	test_per_gauge_records_have_alert_flags	Pass
	test_records.py::TestSequenceRecordsSchema	test_per_gauge_records_have_peak	Pass
	test_records.py::TestSequenceRecordsSchema	test_per_gauge_records_have_peak_hour	Pass
	test_records.py::TestSequenceRecordsSchema	test_per_gauge_sequence_count	Pass
	test_records.py::TestSequenceRecordsSchema	test_severe_implies_warning	Pass
	test_records.py::TestSequenceRecordsSchema	test_warning_implies_alert	Pass
	test_single_gauge.py::TestSingleGaugeMode	test_error_message_lists_available_ids	Pass
	test_single_gauge.py::TestSingleGaugeMode	test_invalid_gauge_id_raises	Pass
	test_single_gauge.py::TestSingleGaugeMode	test_named_file_gauge_ids	Pass
	test_single_gauge.py::TestSingleGaugeMode	test_named_file_peaks_length_one	Pass
	test_single_gauge.py::TestSingleGaugeMode	test_named_file_schema_version	Pass
	test_single_gauge.py::TestSingleGaugeMode	test_named_file_sequence_count	Pass
	test_single_gauge.py::TestSingleGaugeMode	test_named_output_file_written	Pass
	test_single_gauge.py::TestSingleGaugeMode	test_summary_num_gauges_is_one	Pass
	test_single_gauge.py::TestSingleGaugeWithClassifier	test_single_gauge_classifier_result_dict	Pass
	test_single_gauge.py::TestSingleGaugeWithClassifier	test_single_gauge_train_classifier	Pass
MKM-TD-001	leg_pvs.py::TestComputeLegPVs	test_higher_spread_means_higher_premium	Pass
	leg_pvs.py::TestComputeLegPVs	test_positive_hazard_rate	Pass
	leg_pvs.py::TestComputeLegPVs	test_zero_hazard_rate	Pass
	market_making.py::TestGenerateBook	test_balanced_book	Pass
	market_making.py::TestGenerateBook	test_balanced_notional	Pass
	market_making.py::TestGenerateBook	test_counterparties_distributed	Pass
	market_making.py::TestGenerateBook	test_files_saved	Pass
	market_making.py::TestGenerateBook	test_payer_trades_below_fair	Pass
	market_making.py::TestGenerateBook	test_receiver_trades_above_fair	Pass
	market_making.py::TestGenerateBook	test_reproducible_with_seed	Pass
	market_making.py::TestGenerateBook	test_small_mtm	Pass
	market_making.py::TestGenerateBook	test_valid_cdm_structure	Pass
	market_making.py::TestPrintBookSummary	test_summary_runs	Pass
	market_making.py::TestTradeSpreadsRange	test_market_making_spreads_are_integers	Pass
	market_making.py::TestTradeSpreadsRange	test_thames_central_spreads_are_integers	Pass
	thames_central.py::TestThamesCentralBook	test_files_saved	Pass
	thames_central.py::TestThamesCentralBook	test_calendar_spreads	Pass
	thames_central.py::TestThamesCentralBook	test_reproducible_with_seed	Pass
	thames_central.py::TestThamesCentralBook	test_trade_count	Pass
	thames_central.py::TestThamesCentralBook	test_central_gauges	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	thames_central.py::TestThamesCentralBuild	test_build_cdm_structure	Pass
	thames_central.py::TestThamesCentralBuild	test_westminster_is_net_short	Pass
	engine.py::TestDeltaEngine	test_build_risk_grid	Pass
	engine.py::TestDeltaEngine	test_enrich_receiver_trade_negative_dv01	Pass
	engine.py::TestDeltaEngine	test_enrich_trade	Pass
	engine.py::TestDeltaEngine	test_payer_receiver_dv01_opposite	Pass
	engine.py::TestHazardCurveRevaluation	test_curve_change_moves_fair_spread	Pass
	engine.py::TestHazardCurveRevaluation	test_curve_change_moves_mtm	Pass
	engine.py::TestHazardCurveRevaluation	test_curve_change_pnl_consistent_with_fd01	Pass
	engine.py::TestHazardCurveRevaluation	test_curve_change_produces_daily_pnl	Pass
	formulas.py::TestBasisDelta	test_basis_delta_positive	Pass
	formulas.py::TestBasisDelta	test_basis_delta_similar_to_gauge_delta	Pass
	formulas.py::TestGaugeDelta	test_delta_spread_approximately_one	Pass
	formulas.py::TestGaugeDelta	test_higher_notional_means_higher_dv01	Pass
	formulas.py::TestGaugeDelta	test_positive_dv01_for_long_position	Pass
	formulas.py::TestGaugeDelta	test_zero_hazard_rate_gives_zero_delta	Pass
	formulas.py::TestMarkToMarket	test_mtm_calculation_correct	Pass
	formulas.py::TestMarkToMarket	test_mtm_negative_when_spread_widens_for_seller	Pass
	formulas.py::TestMarkToMarket	test_mtm_positive_when_spread_widens_for_buyer	Pass
	formulas.py::TestMarkToMarket	test_mtm_zero_when_spreads_equal	Pass
	formulas.py::TestRiskyAnnuity	test_longer_tenor_means_higher_annuity	Pass
	formulas.py::TestRiskyAnnuity	test_positive_hazard_rate_reduces_annuity	Pass
	formulas.py::TestRiskyAnnuity	test_reasonable_range	Pass
	formulas.py::TestRiskyAnnuity	test_zero_hazard_rate	Pass
	market_state.py::TestEffectiveRates	test_effective_rates_include_all_gauges	Pass
	market_state.py::TestEffectiveRates	test_effective_rates_mark_adjusted	Pass
	market_state.py::TestEffectiveRates	test_effective_rates_merge_correctly	Pass
	market_state.py::TestMarketStateLoad	test_base_rates_match_gaugehc	Pass
	market_state.py::TestMarketStateLoad	test_initial_load_creates_state	Pass
	market_state.py::TestMarketStateLoad	test_load_persists	Pass
	market_state.py::TestMarketStateReset	test_reset_all	Pass
	market_state.py::TestMarketStateReset	test_reset_single_gauge	Pass
	market_state.py::TestMarketStateUpdate	test_get_effective_rate_with_adjustment	Pass
	market_state.py::TestMarketStateUpdate	test_unadjusted_gauge_returns_base	Pass
	market_state.py::TestMarketStateUpdate	test_update_gauge_rate	Pass
	market_state.py::TestMissingCoverage	test_commit_yield_curve	Pass
	market_state.py::TestMissingCoverage	test_get_yield_rate_above_max_tenor	Pass
	market_state.py::TestMissingCoverage	test_get_yield_rate_below_min_tenor	Pass
	market_state.py::TestMissingCoverage	test_get_yield_rate_exact_tenor	Pass
	market_state.py::TestMissingCoverage	test_get_yield_rate_interpolated	Pass
	market_state.py::TestMissingCoverage	test_load_base_curves_gauge_with_no_id_skipped	Pass
	market_state.py::TestMissingCoverage	test_load_base_curves_gauges_key	Pass
	market_state.py::TestMissingCoverage	test_load_base_curves_no_file	Pass
	market_state.py::TestMissingCoverage	test_load_base_curves_unknown_format	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	market_state.py::TestMissingCoverage	test_reset_yield_curve	Pass
	market_state.py::TestMissingCoverage	test_update_gauge_rate_with_notes	Pass
	daily_pnl.py::TestDailyPnL	test_closed_trades_excluded	Pass
	daily_pnl.py::TestDailyPnL	test_daily_pnl_without_previous_eod	Pass
	daily_pnl.py::TestDailyPnL	test_pnl_decomposition_sums	Pass
	daily_pnl.py::TestDailyPnL	test_running_pnl_calculation	Pass
	daily_pnl.py::TestPhantomPnLFix	test_market_pnl_nonzero_with_eod_mark	Pass
	daily_pnl.py::TestPhantomPnLFix	test_market_pnl_zero_for_missing_eod_trade	Pass
	daily_pnl.py::TestPhantomPnLFix	test_new_trade_has_inception_pnl	Pass
	daily_pnl.py::TestPhantomPnLFix	test_pnl_decomposition_with_prev_eod	Pass
	eod.py::TestEodSnapshot	test_eod_history	Pass
	eod.py::TestEodSnapshot	test_generate_snapshot	Pass
	eod.py::TestEodSnapshot	test_idempotent_snapshot	Pass
	eod.py::TestEodSnapshot	test_pnl_series	Pass
	eod.py::TestEodSnapshot	test_snapshot_persists	Pass
	marks.py::TestTradeMarks	test_close_trade	Pass
	marks.py::TestTradeMarks	test_load_empty_marks	Pass
	marks.py::TestTradeMarks	test_update_and_load_mark	Pass
	stress_hydrograph.py::TestStressHydrograph	test_KnockOutLogic_price_stays_at_notional_after_ko	Pass
	stress_hydrograph.py::TestStressHydrograph	test_KnockOutLogic_triggers_ko_independently	Pass
	stress_hydrograph.py::TestStressHydrograph	test_KnockOutLogic_when_below_trigger	Pass
	stress_hydrograph.py::TestStressHydrograph	test_KnockOutLogic_ko_cash_is_negative	Pass
	stress_hydrograph.py::TestStressHydrograph	test_KnockOutLogic_stocks_out_at_trigger	Pass
	stress_hydrograph.py::TestSyntheticHydrograph	test_all_above_base	Pass
	stress_hydrograph.py::TestSyntheticHydrograph	test_custom_hours	Pass
	stress_hydrograph.py::TestSyntheticHydrograph	test_monotonic_rise	Pass
	stress_hydrograph.py::TestSyntheticHydrograph	test_peak_level	Pass
	stress_hydrograph.py::TestSyntheticHydrograph	test_recession_after_peak	Pass
	stress_hydrograph.py::TestSyntheticHydrograph	test_returns_168_hours	Pass
	stress_hydrograph.py::TestSyntheticHydrograph	test_short_storm	Pass
	stress_hydrograph.py::TestSyntheticHydrograph	test_starts_at_base_level	Pass
	stress_hydrograph.py::TestSyntheticHydrograph	test_zero_duration	Pass
	eod_routes.py::TestEODErrorHandlers	test_eod_history_engine_error_returns_500	Pass
	eod_routes.py::TestEODErrorHandlers	test_get_eod_pdf_error_returns_500	Pass
	eod_routes.py::TestEODErrorHandlers	test_submit_eod_engine_error_returns_500	Pass
	eod_routes.py::TestEODHistory	test_eod_history_after_snap	Pass
	eod_routes.py::TestEODHistory	test_eod_history_empty	Pass
	eod_routes.py::TestEODHistory	test_multiple_eod_snaps	Pass
	eod_routes.py::TestEODNoDateBody	test_submit_eod_no_json_uses_today	Pass
	eod_routes.py::TestEODPdf	test_eod_pdf_found_serves_file	Pass
	eod_routes.py::TestEODPdf	test_eod_pdf_not_found	Pass
	eod_routes.py::TestEODPdfGeneration	test_submit_eod_import_error_still_succeeds	Pass
	eod_routes.py::TestEODPdfGeneration	test_submit_eod_pdf_exception_still_succeeds	Pass
	eod_routes.py::TestPnLResetAfterEOD	test_pnl_uses_eod_as_baseline	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	eod_routes.py::TestSubmitEOD	test_eod_idempotent_same_day	Pass
	eod_routes.py::TestSubmitEOD	test_submit_eod	Pass
	eod_routes.py::TestSubmitEOD	test_submit_eod_returns_summary	Pass
	eod_routes.py::TestSubmitEOD	test_submit_eod_saves_snapshot_file	Pass
	eod_routes.py::TestSubmitEOD	test_submit_eod_snapshot_has_market_state	Pass
	eod_routes.py::TestSubmitEOD	test_submit_eod_snapshot_has_positions	Pass
	market_state_routes.py::TestGetMarketState	test_get_market_state	Pass
	market_state_routes.py::TestGetMarketState	test_get_market_state_has_effective_rates	Pass
	market_state_routes.py::TestGetMarketState	test_get_market_state_initial_load_creates_defaults	Pass
	market_state_routes.py::TestMarketStateErrorHandlers	test_get_market_state_engine_error_returns_500	Pass
	market_state_routes.py::TestMarketStateErrorHandlers	test_reset_market_state_engine_error_returns_500	Pass
	market_state_routes.py::TestMarketStateErrorHandlers	test_update_market_state_null_body_returns_400	Pass
	market_state_routes.py::TestResetMarketState	test_reset_market_state_all	Pass
	market_state_routes.py::TestResetMarketState	test_reset_market_state_single_gauge	Pass
	market_state_routes.py::TestUpdateMarketState	test_update_market_state_persists	Pass
	market_state_routes.py::TestUpdateMarketState	test_update_market_state	Pass
	market_state_routes.py::TestUpdateMarketState	test_update_market_state_missing_gauge	Pass
	market_state_routes.py::TestUpdateMarketState	test_update_market_state_no_body	Pass
	market_state_routes.py::TestUpdateMarketState	test_update_market_state_returns_pnl_impact	Pass
	risk_routes.py::TestRiskErrorHandlers	test_risk_grid_engine_error_returns_500	Pass
	risk_routes.py::TestRiskErrorHandlers	test_trade_map_engine_error_returns_500	Pass
	risk_routes.py::TestRiskGrid	test_risk_grid_has_gauges	Pass
	risk_routes.py::TestRiskGrid	test_risk_grid_returns_grid	Pass
	risk_routes.py::TestRiskGrid	test_risk_grid_with_no_trades	Pass
	risk_routes.py::TestTradeMap	test_trade_map_fs01_by_tenor	Pass
	risk_routes.py::TestTradeMap	test_trade_map_gauge_has_coordinates	Pass
	risk_routes.py::TestTradeMap	test_trade_map_includes_fs01	Pass
	risk_routes.py::TestTradeMap	test_trade_map_includes_notional	Pass
	risk_routes.py::TestTradeMap	test_trade_map_includes_pnl	Pass
	risk_routes.py::TestTradeMap	test_trade_map_num_trades	Pass
	risk_routes.py::TestTradeMap	test_trade_map_returns_positions	Pass
	risk_routes.py::TestTradeMap	test_trade_map_with_no_trades	Pass
	risk_routes.py::TestTradeMapWithProperties	test_trade_map_gauge_not_in_gauge_hc_loaded	Pass
	risk_routes.py::TestTradeMapWithProperties	test_trade_map_loads_property_locations	Pass
	risk_routes.py::TestTradeMapWithProperties	test_trade_map_skips_closed_trades	Pass
	blotter_detail.py::TestBlotterEndpoints	test_blotter_fetches_correct_endpoint	Pass
	blotter_detail.py::TestBlotterEndpoints	test_close_endpoint	Pass
	blotter_detail.py::TestFilterBar	test_clear_filters	Pass
	blotter_detail.py::TestFilterBar	test_external_filter_api	Pass
	blotter_detail.py::TestFilterBar	test_filter_function_exists	Pass
	blotter_detail.py::TestFilterBar	test_get_filtered_trades	Pass
	blotter_detail.py::TestFilterBar	test_remove_single_filter	Pass
	blotter_detail.py::TestHazardRateArrows	test_down_arrow	Pass
	blotter_detail.py::TestHazardRateArrows	test_prev_fair_spread_comparison	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	blotter_detail.py::TestHazardRateArrows	test_up_arrow	Pass
	blotter_detail.py::TestMarketPnlCol	test_market_pnl_column	Pass
	blotter_detail.py::TestMarketPnlCol	test_mkt_pnl_label	Pass
	blotter_detail.py::TestPnLBar	test_pnl_bar_daily	Pass
	blotter_detail.py::TestPnLBar	test_pnl_bar_from_market	Pass
	blotter_detail.py::TestPnLBar	test_pnl_bar_net_notional	Pass
	blotter_detail.py::TestPnLBar	test_pnl_bar_running	Pass
	blotter_detail.py::TestSignedNotional	test_is_payer_referenced	Pass
	blotter_detail.py::TestSignedNotional	test_signed_notional_direction	Pass
	blotter_detail.py::TestSortAndNavigation	test_sort_toggle	Pass
	blotter_detail.py::TestSortAndNavigation	test_trade_view_click	Pass
	blotter_detail.py::TestTableStructure	test_gauge_fs01_column	Pass
	blotter_detail.py::TestTableStructure	test_maturity_formatting	Pass
	eod_detail.py::TestEODEndpoints	test_eod_blotter_fetch	Pass
	eod_detail.py::TestEODEndpoints	test_eod_history_fetch	Pass
	eod_detail.py::TestEODEndpoints	test_eod_pnl_series_fetch	Pass
	eod_detail.py::TestEODEndpoints	test_eod_submit_endpoint	Pass
	eod_detail.py::TestEODFunctions	test_attribution_section	Pass
	eod_detail.py::TestEODFunctions	test_chart_rendering	Pass
	eod_detail.py::TestEODFunctions	test_cleanup_function	Pass
	eod_detail.py::TestEODFunctions	test_download_pdf_function	Pass
	eod_detail.py::TestEODFunctions	test_submit_function	Pass
	eod_detail.py::TestEODUIStructure	test_attribution_wrap	Pass
	eod_detail.py::TestEODUIStructure	test_cards_container	Pass
	eod_detail.py::TestEODUIStructure	test_date_input	Pass
	eod_detail.py::TestEODUIStructure	test_history_wrap	Pass
	eod_detail.py::TestEODUIStructure	test_submit_button	Pass
	market_detail.py::TestActionButtonFormat	test_button_font_matches_tabs	Pass
	market_detail.py::TestActionButtonFormat	test_button_padding_matches_tabs	Pass
	market_detail.py::TestActionButtonFormat	test_buttons_in_shared_container	Pass
	market_detail.py::TestActionButtonFormat	test_commit_button_blue	Pass
	market_detail.py::TestActionButtonFormat	test_history_button_grey_not_blue	Pass
	market_detail.py::TestActionButtonFormat	test_individual_border_radius_on_buttons	Pass
	market_detail.py::TestActionButtonFormat	test_plot_button_grey_not_blue	Pass
	market_detail.py::TestActionButtonFormat	test_reset_button_grey	Pass
	market_detail.py::TestCommitButton	test_commit_button_calls_function	Pass
	market_detail.py::TestCommitButton	test_commit_button_id_present	Pass
	market_detail.py::TestCommitButton	test_commit_button_not_hidden	Pass
	market_detail.py::TestCommitButton	test_commit_function_defined	Pass
	market_detail.py::TestCommitEndpoint	test_commit_posts_hazard_ts	Pass
	market_detail.py::TestCommitEndpoint	test_commit_posts_yield_curve	Pass
	market_detail.py::TestCommitEndpoint	test_commit_shows_pnl_impact	Pass
	market_detail.py::TestCommitEndpoint	test_commit_switches_to_blotter	Pass
	market_detail.py::TestDirtyStateTracking	test_hazard_dirty_keys	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	market_detail.py::TestDirtyStateTracking	test_committed_label	Pass
	market_detail.py::TestDirtyStateTracking	test_yield_dirty_flag	Pass
	market_detail.py::TestMarketUIStructure	test_cleanup_function	Pass
	market_detail.py::TestMarketUIStructure	test_gauge_list_function	Pass
	market_detail.py::TestMarketUIStructure	test_market_canvas	Pass
	market_detail.py::TestMarketUIStructure	test_mode_selector	Pass
	market_detail.py::TestNewTradeButton	test_new_trade_button_grey	Pass
	market_detail.py::TestNewTradeButton	test_new_trade_button_label	Pass
	market_detail.py::TestNewTradeButton	test_new_trade_button_present	Pass
	market_detail.py::TestNewTradeButton	test_new_trade_calls_function	Pass
	market_detail.py::TestNewTradeButton	test_new_trade_function_opens_prs	Pass
	market_detail.py::TestNewTradeButton	test_new_trade_left_of_history	Pass
	market_detail.py::TestNewTradeButton	test_no_border_radius_on_new_trade_btn	Pass
	market_detail.py::TestResetEndpoints	test_reset_function_exists	Pass
	market_detail.py::TestResetEndpoints	test_reset_hazard_ts_endpoint	Pass
	market_detail.py::TestResetEndpoints	test_reset_yield_curve_endpoint	Pass
CDM-ALL	gauge_mapping.py	test_all_cdm_fields_present	Pass
	gauge_mapping.py	test_all_types_valid	Pass
	gauge_mapping.py	test_all_values_valid	Pass
	mortgage_mapping.py	test_all_cdm_fields_present	Pass
	mortgage_mapping.py	test_all_types_valid	Pass
	mortgage_mapping.py	test_all_values_valid	Pass
	property_mapping.py	test_all_cdm_fields_present	Pass
	property_mapping.py	test_all_types_valid	Pass
	property_mapping.py	test_all_values_valid	Pass
	schema_basics.py::TestCDMSchemaBasics	test_get_required_fields_returns_list[gauge_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_get_required_fields_returns_list[mortgage_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_get_required_fields_returns_list[property_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_get_required_fields_returns_list[prs_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_get_required_fields_returns_list[storm_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_get_required_fields_returns_list[stormts_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_list_all_fields_returns_paths[gauge_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_list_all_fields_returns_paths[mortgage_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_list_all_fields_returns_paths[property_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_list_all_fields_returns_paths[prs_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_list_all_fields_returns_paths[storm_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_list_all_fields_returns_paths[stormts_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_repr_returns_string[gauge_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_repr_returns_string[mortgage_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_repr_returns_string[property_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_repr_returns_string[prs_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_repr_returns_string[storm_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_repr_returns_string[stormts_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_schema_returns_non_empty_dict[gauge_cdm]	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	schema_basics.py::TestCDMSchemaBasics	test_schema_returns_non_empty_dict[mortgage_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_schema_returns_non_empty_dict[property_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_schema_returns_non_empty_dict[prs_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_schema_returns_non_empty_dict[storm_cdm]	Pass
	schema_basics.py::TestCDMSchemaBasics	test_schema_returns_non_empty_dict[stormus_cdm]	Pass
E2E-ALL	TestContextMenuNavigation.py	test_hazard_curve_item_opens_gauge_panel[chromium]	Pass
	TestGaugeContextMenu.py	test_click_away_closes_menu[chromium]	Pass
	TestGaugeContextMenu.py	test_context_menu_has_header[chromium]	Pass
	TestGaugeContextMenu.py	test_context_menu_has_items[chromium]	Pass
	TestGaugeContextMenu.py	test_right_click_gauge_shows_menu[chromium]	Pass
	TestNavMenuDropdowns.py	test_gauge_select_exists[chromium]	Pass
	TestNavMenuDropdowns.py	test_gauge_select_has_options[chromium]	Pass
	TestNavMenuDropdowns.py	test_nav_container_exists[chromium]	Pass
	TestNavMenuDropdowns.py	test_property_select_exists[chromium]	Pass
	TestNavMenuDropdowns.py	test_property_select_has_options[chromium]	Pass
	TestNavMenuDropdowns.py	test_selecting_gauge_shows_action_buttons[chromium]	Pass
	TestCounterpartyAPI.py	test_counterparty_summary_fields[chromium]	Pass
	TestCounterpartyAPI.py	test_counterparty_types_diverse[chromium]	Pass
	TestCounterpartyAPI.py	test_get_counterparty_detail[chromium]	Pass
	TestCounterpartyAPI.py	test_list_counterparties_returns_data[chromium]	Pass
	TestCounterpartyAPI.py	test_nonexistent_counterparty_returns_404[chromium]	Pass
	TestCounterpartyInPRS.py	test_counterparty_dropdown_has_options[chromium]	Pass
	TestCounterpartyInPRS.py	test_counterparty_options_show_rating[chromium]	Pass
	TestCounterpartyInPRS.py	test_prs_has_counterparty_dropdown[chromium]	Pass
	TestCounterpartyInPRS.py	test_selecting_counterparty_updates_pricing[chromium]	Pass
	TestFS01ToBlotter.py	test_fs01_cell_click_filters_blotter[chromium]	Skip
	TestFS01ToBlotter.py	test_fs01_tab_shows_risk_grid[chromium]	Pass
	TestGaugeBlotterFlow.py	test_gauge_panel_blotter_link_opens_trading_desk[chromium]	Pass
	TestGaugeBlotterFlow.py	test_trading_desk_blotter_visible_after_gauge_link[chromium]	Pass
	TestContextMenuToTradingDesk.py	test_gauge_blotter_menu_item_opens_trading_desk[chromium]	Pass
	TestContextMenuToTradingDesk.py	test_right_click_gauge_shows_context_menu[chromium]	Pass
	TestHistoricalToStress.py	test_historical_tab_has_content[chromium]	Pass
	TestHistoricalToStress.py	test_storm_scenario_click_activates_stress_tab[chromium]	Skip
	TestGaugePRSToBlotterRoundTrip.py	test_01_open_gauge_prs_shows_hazard_curve[chromium]	Pass
	TestGaugePRSToBlotterRoundTrip.py	test_02_gauge_with_trade_shows_active_blotter_button[chromium]	Pass
	TestGaugePRSToBlotterRoundTrip.py	test_03_blotter_button_opens_td_filtered_to_gauge[chromium]	Pass
	TestGaugePRSToBlotterRoundTrip.py	test_04_blotter_button_muted_for_gauge_without_trades[chromium]	Skip
	TestNavMenuToPanel.py	test_nav_gauge_select_opens_panel[chromium]	FAIL
	TestNavMenuToPanel.py	test_nav_property_select_opens_panel[chromium]	FAIL
	TestPropertyStormToPRS.py	test_01_storm_panel_has_prs_tab[chromium]	Pass
	TestPropertyStormToPRS.py	test_02_prs_tab_opens_property_hazard_panel[chromium]	Pass
	TestPropertyStormToPRS.py	test_03_storm_flood_history_click_opens_timeline[chromium]	Skip
	TestStormScenarioTabConsistency.py	test_distribution_and_history_show_same_flood_count[chromium]	Skip
	TestStormScenarioTabConsistency.py	test_header_flood_count_matches_history[chromium]	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	TestStormScenarioTabConsistency.py	test_worst_storms_are_subset_of_history[chromium]	Skip
	TestHazardCurveDataLoads.py	test_gauge_panel_loads_data_not_error[chromium]	Pass
	TestHazardCurveDataLoads.py	test_hazard_api_returns_success[chromium]	Pass
	TestHazardCurveDataLoads.py	test_hazard_curve_tab_has_real_data[chromium]	Pass
	TestNoErrorNotifications.py	test_no_error_notifications_after_clearing[chromium]	Pass
	TestStressStormDataLoads.py	test_portfolio_storms_api_returns_data[chromium]	Pass
	TestStressStormDataLoads.py	test_stress_storms_api_returns_data[chromium]	Pass
	TestPropertyDataLoads.py	test_property_panel_loads_without_errors[chromium]	Pass
	TestPropertyDataLoads.py	test_property_storm_api_returns_data[chromium]	Pass
	TestPropertyDataLoads.py	test_propertyts_summary_api_returns_data[chromium]	Pass
	TestTradingDataLoads.py	test_blotter_api_returns_trades[chromium]	Pass
	TestTradingDataLoads.py	test_market_state_api_returns_data[chromium]	Pass
	TestTradingDataLoads.py	test_risk_grid_api_returns_data[chromium]	Pass
	TestInvalidGaugeId.py	test_nonexistent_gauge_does_not_crash_page[chromium]	Pass
	TestInvalidGaugeId.py	test_nonexistent_gauge_shows_error[chromium]	Pass
	TestInvalidPropertyId.py	test_nonexistent_property_does_not_crash_page[chromium]	Pass
	TestInvalidPropertyId.py	test_nonexistent_property_shows_error[chromium]	Pass
	TestEscapeKeyClosesPanels.py	test_escape_closes_gauge_panel[chromium]	Pass
	TestEscapeKeyClosesPanels.py	test_escape_closes_governance_panel[chromium]	Pass
	TestEscapeKeyClosesPanels.py	test_escape_closes_trading_desk[chromium]	Pass
	TestMapLayerControls.py	test_layer_control_exists[chromium]	Pass
	TestMapLayerControls.py	test_layer_toggle_changes_markers[chromium]	Pass
	TestGaugeBlotterButtonState.py	test_blotter_button_click_opens_trading_desk[chromium]	Pass
	TestGaugeBlotterButtonState.py	test_blotter_button_enabled_for_gauge_with_trades[chromium]	Pass
	TestGaugeBlotterButtonState.py	test_blotter_button_exists_in_tab_bar[chromium]	Pass
	TestGaugeBlotterButtonState.py	test_blotter_button_muted_for_gauge_without_trades[chromium]	Pass
	TestFloodProbabilityTab.py	test_chart_rendered[chromium]	Pass
	TestFloodProbabilityTab.py	test_content_shows_probability_data[chromium]	Pass
	TestHazardCurveTab.py	test_chart_canvas_rendered[chromium]	Pass
	TestHazardCurveTab.py	test_stats_bar_shows_data[chromium]	Pass
	TestHazardCurveTab.py	test_tab_title_updates[chromium]	Pass
	TestReturnPeriodTab.py	test_chart_rendered[chromium]	Pass
	TestReturnPeriodTab.py	test_content_shows_return_period_data[chromium]	Pass
	TestHistoricalTab.py	test_content_loads_with_timeseries_data[chromium]	Pass
	TestHistoricalTab.py	test_content_mentions_historical_context[chromium]	Pass
	TestHistoricalTab.py	test_storm_scenarios_list_exists[chromium]	Pass
	TestStressTestTab.py	test_auto_selects_worst_case_storm[chromium]	Pass
	TestStressTestTab.py	test_charts_render[chromium]	Skip
	TestStressTestTab.py	test_content_mentions_stress_concepts[chromium]	Pass
	TestStressTestTab.py	test_storm_dropdown_exists[chromium]	Pass
	TestPRSPricingTab.py	test_chart_canvas_exists[chromium]	Pass
	TestPRSPricingTab.py	test_commit_button_exists[chromium]	Pass
	TestPRSPricingTab.py	test_controls_exist[chromium]	Pass
	TestPRSPricingTab.py	test_direction_dropdown_options[chromium]	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	TestPRSPricingTab.py	test_trigger_change_updates_display[chromium]	FAIL
	TestPRSPricingTab.py	test_trigger_dropdown_options[chromium]	FAIL
	TestGaugePanelBlotterLink.py	test_blotter_link_exists[chromium]	Pass
	TestGaugePanelBlotterLink.py	test_blotter_link_opens_trading_desk[chromium]	Pass
	TestGaugePanelOpens.py	test_open_gauge_panel_via_js[chromium]	Pass
	TestGaugePanelOpens.py	test_panel_has_tab_bar[chromium]	Pass
	TestGaugePanelOpens.py	test_panel_title_shows_gauge_name[chromium]	Pass
	TestGaugePanelTabs.py	test_hazard_curve_tab_renders[chromium]	Pass
	TestGaugePanelTabs.py	test_historical_tab_renders[chromium]	Pass
	TestGaugePanelTabs.py	test_prs_tab_renders[chromium]	Pass
	TestGaugePanelTabs.py	test_stress_tab_renders[chromium]	Pass
	TestGaugePRSMarketCurve.py	test_hazard_term_structure_api_reachable[chromium]	FAIL
	TestGaugePRSMarketCurve.py	test_mkt_hazard_ts_has_current_gauge[chromium]	FAIL
	TestGaugePRSMarketCurve.py	test_mkt_hazard_ts_populated[chromium]	FAIL
	TestGaugePRSMarketCurve.py	test_prs_hazard_chart_has_data[chromium]	FAIL
	TestGaugePRSMarketCurve.py	test_prs_hazard_chart_values_are_reasonable[chromium]	FAIL
	TestGaugePRSMarketCurve.py	test_pv_premium_chart_has_data_on_open[chromium]	FAIL
	TestBCBS239PrincipleEdit.py	test_edit_buttons_on_principles[chromium]	Pass
	TestBCBS239PrincipleEdit.py	test_edit_opens_modal[chromium]	Skip
	TestBCBS239PrincipleEdit.py	test_modal_has_confirm_cancel[chromium]	Skip
	TestBibliographyExport.py	test_bibtex_export_button[chromium]	Pass
	TestBibliographyExport.py	test_reference_table_sortable[chromium]	Pass
	TestDocumentUploadDownload.py	test_document_list_renders[chromium]	Pass
	TestDocumentUploadDownload.py	test_download_buttons_clickable[chromium]	Skip
	TestDocumentUploadDownload.py	test_upload_area_exists[chromium]	Pass
	TestDataLineageAPI.py	test_lineage_api_returns_data[chromium]	Pass
	TestDataLineageAPI.py	test_staleness_api_returns_data[chromium]	Pass
	TestDataLineageTab.py	test_content_mentions_lineage_keywords[chromium]	Pass
	TestDataLineageTab.py	test_has_pipeline_steps[chromium]	Pass
	TestDataLineageTab.py	test_has_staleness_info[chromium]	Pass
	TestDataLineageTab.py	test_tab_content_loads[chromium]	Pass
	TestProvenanceTraceAPI.py	test_trace_gauge_returns_steps[chromium]	Pass
	TestProvenanceTraceAPI.py	test_trace_missing_params_returns_400[chromium]	Pass
	TestProvenanceTraceAPI.py	test_trace_nonexistent_returns_empty[chromium]	Pass
	TestProvenanceTraceAPI.py	test_trace_property_returns_steps[chromium]	Pass
	TestAuditReportsAPI.py	test_audit_reports_api_returns_files[chromium]	Pass
	TestAuditReportsAPI.py	test_test_summary_api[chromium]	Pass
	TestAuditReportsTab.py	test_content_mentions_audit_keywords[chromium]	Pass
	TestAuditReportsTab.py	test_has_artefact_files[chromium]	Pass
	TestAuditReportsTab.py	test_has_test_summary_or_file_cards[chromium]	Pass
	TestAuditReportsTab.py	test_tab_content_loads[chromium]	Pass
	TestFieldLineageAPI.py	test_field_lineage_api_returns_data[chromium]	Pass
	TestFieldLineageAPI.py	test_field_search_api[chromium]	Pass
	TestFieldLineageTab.py	test_content_mentions_field_keywords[chromium]	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	TestFieldLineageTab.py	test_has_report_cards_or_search[chromium]	Pass
	TestFieldLineageTab.py	test_shows_field_counts[chromium]	Pass
	TestFieldLineageTab.py	test_tab_content_loads[chromium]	Pass
	TestModelWorkstreamDetail.py	test_back_button_returns_to_list[chromium]	Pass
	TestModelWorkstreamDetail.py	test_model_row_opens_detail[chromium]	Pass
	TestMRCAgendaCRUD.py	test_add_agenda_item[chromium]	Pass
	TestMRCAgendaCRUD.py	test_delete_agenda_item[chromium]	Pass
	TestMRCMeetingsAPI.py	test_list_meetings_returns_data[chromium]	Pass
	TestMRCMeetingsAPI.py	test_meeting_detail_returns_full_object[chromium]	Pass
	TestMRCMeetingsAPI.py	test_meeting_summary_has_fields[chromium]	Pass
	TestMRCActionsCRUD.py	test_add_action[chromium]	Pass
	TestMRCActionsCRUD.py	test_delete_action[chromium]	Pass
	TestMRCDecisionsCRUD.py	test_add_decision[chromium]	Pass
	TestMRCDecisionsCRUD.py	test_delete_decision[chromium]	Pass
	TestMRCMeetingDetailUI.py	test_meeting_detail_has_tabs[chromium]	Pass
	TestMRCMeetingDetailUI.py	test_meeting_detail_shows_metadata[chromium]	Pass
	TestMRCMeetingDetailUI.py	test_meeting_list_has_rows[chromium]	Pass
	TestMRCMeetingDetailUI.py	test_meeting_row_clickable[chromium]	Pass
	TestMRCParticipantsCRUD.py	test_add_participant[chromium]	Pass
	TestMRCParticipantsCRUD.py	test_delete_participant[chromium]	Pass
	TestMRCMeetingCRUD.py	test_add_button_opens_form[chromium]	Pass
	TestMRCMeetingCRUD.py	test_add_meeting_button_exists[chromium]	Skip
	TestMRCMeetingCRUD.py	test_meeting_list_renders[chromium]	Pass
	TestRACIMatrixInteraction.py	test_cells_interactive[chromium]	Pass
	TestRACIMatrixInteraction.py	test_matrix_grid_renders[chromium]	Pass
	TestAuditTab.py	test_content_mentions_audit[chromium]	Pass
	TestAuditTab.py	test_tab_content_loads[chromium]	Pass
	TestBCBS239Tab.py	test_content_mentions_bcbs[chromium]	Pass
	TestBCBS239Tab.py	test_tab_content_loads[chromium]	Pass
	TestBibliographyTab.py	test_content_mentions_bibliography[chromium]	Pass
	TestBibliographyTab.py	test_tab_content_loads[chromium]	Pass
	TestDocumentsTab.py	test_content_area_exists[chromium]	Pass
	TestDocumentsTab.py	test_tab_content_loads[chromium]	Pass
	TestGovernancePanelOpens.py	test_has_content_area[chromium]	Pass
	TestGovernancePanelOpens.py	test_has_tab_buttons[chromium]	Pass
	TestGovernancePanelOpens.py	test_panel_is_visible[chromium]	Pass
	TestMRCTab.py	test_content_mentions_mrc[chromium]	Pass
	TestMRCTab.py	test_tab_content_loads[chromium]	Pass
	TestModelInventoryTab.py	test_content_mentions_model[chromium]	Pass
	TestModelInventoryTab.py	test_tab_content_loads[chromium]	Pass
	TestRACITab.py	test_content_mentions_raci[chromium]	Pass
	TestRACITab.py	test_tab_content_loads[chromium]	Pass
	TestCloseOutTradePL.py	test_01_open_blotter_and_count_trades[chromium]	Pass
	TestCloseOutTradePL.py	test_02_click_close_button_opens_modal[chromium]	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	TestCloseOutTradePL.py	test_03_enter_closeout_spread_and_confirm[chromium]	Pass
	TestCloseOutTradePL.py	test_04_verify_closed_trade_in_blotter[chromium]	Pass
	TestEODSnapAndHistory.py	test_01_open_eod_tab_count_history[chromium]	Pass
	TestEODSnapAndHistory.py	test_02_submit_eod[chromium]	Pass
	TestEODSnapAndHistory.py	test_03_verify_new_snapshot_in_history[chromium]	Pass
	TestGaugePRSBookTrade.py	test_01_open_prs_and_verify_controls[chromium]	Pass
	TestGaugePRSBookTrade.py	test_02_select_counterparty[chromium]	Pass
	TestGaugePRSBookTrade.py	test_03_set_trade_parameters[chromium]	**FAIL**
	TestGaugePRSBookTrade.py	test_04_commit_trade[chromium]	Pass
	TestGaugePRSBookTrade.py	test_05_new_trade_appears_in_blotter[chromium]	Pass
	TestMarketUpdatePL.py	test_01_open_market_tab_record_yield[chromium]	Pass
	TestMarketUpdatePL.py	test_02_change_yield_curve_value[chromium]	Pass
	TestMarketUpdatePL.py	test_03_commit_market_changes[chromium]	Pass
	TestMarketUpdatePL.py	test_04_verify_pnl_after_curve_change[chromium]	Pass
	TestAggregateTab.py	test_has_leaflet_or_svg[chromium]	Pass
	TestAggregateTab.py	test_map_view_visible[chromium]	Pass
	TestFS01Tab.py	test_content_mentions_risk_keywords[chromium]	Pass
	TestFS01Tab.py	test_has_risk_grid[chromium]	Pass
	TestFS01Tab.py	test_risk_view_visible[chromium]	Pass
	TestMarketTab.py	test_has_charts_or_inputs[chromium]	Pass
	TestMarketTab.py	test_has_hazard_term_structure[chromium]	Pass
	TestMarketTab.py	test_has_yield_curve_content[chromium]	Pass
	TestMarketTab.py	test_market_view_visible[chromium]	Pass
	TestCurvesTab.py	test_curves_view_visible[chromium]	Pass
	TestCurvesTab.py	test_has_chart_or_curve_content[chromium]	Pass
	TestEODTab.py	test_content_mentions_eod_keywords[chromium]	Pass
	TestEODTab.py	test_eod_view_visible[chromium]	Pass
	TestEODTab.py	test_has_history_list_or_table[chromium]	Pass
	TestStressTab.py	test_has_gauge_dropdown[chromium]	Pass
	TestStressTab.py	test_has_storm_dropdown[chromium]	Pass
	TestStressTab.py	test_stress_view_visible[chromium]	Pass
	TestCloseoutFormValidation.py	test_close_button_opens_modal[chromium]	Skip
	TestCloseoutFormValidation.py	test_negative_closeout_spread_handling[chromium]	Skip
	TestMarketFormValidation.py	test_negative_rate_handling[chromium]	Pass
	TestMarketFormValidation.py	test_non_numeric_tenor_input[chromium]	Skip
	TestPRSFormValidation.py	test_empty_counterparty_disables_commit[chromium]	Skip
	TestPRSFormValidation.py	test_negative_spread_handling[chromium]	Pass
	TestPRSFormValidation.py	test_zero_notional_disables_commit[chromium]	Skip
	TestTradeCloseOut.py	test_blotter_has_close_button[chromium]	Pass
	TestTradeCloseOut.py	test_close_button_opens_dialog[chromium]	Pass
	TestTradeCloseOut.py	test_confirm_closeout_posts[chromium]	Skip
	TestTradeCloseOut.py	test_spread_input_updates_settlement[chromium]	Skip
	TestEODSubmit.py	test_eod_history_has_pdf_link[chromium]	Pass
	TestEODSubmit.py	test_eod_submit_button_exists[chromium]	Skip

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	TestEODSubmit.py	test_eod_submit_creates_snapshot[chromium]	Pass
	TestGaugePRSCommit.py	test_commit_triggers_response[chromium]	FAIL
	TestGaugePRSCommit.py	test_counterparty_dropdown_has_options[chromium]	Pass
	TestGaugePRSCommit.py	test_fill_notional_and_spread[chromium]	Pass
	TestGaugePRSCommit.py	test_selecting_counterparty_enables_commit[chromium]	Pass
	TestMarketStateCommit.py	test_dirty_state_indicator[chromium]	Pass
	TestMarketStateCommit.py	test_save_button_commits_changes[chromium]	Pass
	TestMarketStateCommit.py	test_tenor_inputs_accept_values[chromium]	Pass
	TestMarketStateCommit.py	test_yield_curve_inputs_exist[chromium]	Pass
	TestMarketStateReset.py	test_reset_button_exists[chromium]	Pass
	TestMarketStateReset.py	test_reset_restores_defaults[chromium]	Pass
	TestPropertyPRSCommit.py	test_commit_button_state[chromium]	Skip
	TestPropertyPRSCommit.py	test_fill_prs_values[chromium]	Pass
	TestPropertyPRSCommit.py	test_prs_controls_exist[chromium]	Pass
PLATFORM	catchment_integration.py	test_config	Pass
	catchment_integration.py	test_config_endpoints_include_health_check	Pass
	catchment_integration.py	test_html_file	Pass
	catchment_integration.py	test_routes	Pass
	test_data_lineage_report_collect.py::TestCollectAll	test_collect_result_structure	Pass
	test_data_lineage_report_collect.py::TestCollectAll	test_graph_matches_step_details	Pass
	test_data_lineage_report_collect.py::TestCollectAll	test_counts_positive	Pass
	test_data_lineage_report_collect.py::TestCollectAll	test_returns_dict	Pass
	test_data_lineage_report_collect.py::TestCollectAll	test_step_details_list	Pass
	test_data_lineage_report_collect.py::TestPdfReport	test_pdf_created	Pass
	test_data_lineage_report_collect.py::TestPdfReport	test_pdf_has_header	Pass
	test_data_lineage_report_collect.py::TestPdfReport	test_pdf_non_empty	Pass
	test_data_lineage_report_collect.py::TestPdfReport	test_pdf_with_empty_step_details	Pass
	test_data_lineage_report_collect.py::TestPdfReport	test_pdf_with_issues	Pass
	test_data_lineage_report_collect.py::TestPdfReport	test_pdf_with_lineage_results	Pass
	test_data_lineage_report_collect.py::TestPdfReport	test_returns_path	Pass
	test_dl_report_builders.py::TestHelpers	test_status_colour_fresh	Pass
	test_dl_report_builders.py::TestHelpers	test_status_colour_missing	Pass
	test_dl_report_builders.py::TestHelpers	test_status_colour_stale	Pass
	test_dl_report_builders.py::TestHelpers	test_status_colour_unknown	Pass
	test_dl_report_builders.py::TestHelpers	test_status_label_unknown_uppercased	Pass
	test_dl_report_builders.py::TestHelpers	test_status_label_values	Pass
	test_dl_report_builders.py::TestStationBuilder	test_build_healthy	Pass
	test_dl_report_builders.py::TestStationBuilder	test_build_issues	Pass
	test_dl_report_builders.py::TestStationBuilder	test_exec_summary_healthy	Pass
	test_dl_report_builders.py::TestStationBuilder	test_exec_summary_issues	Pass
	test_dl_report_builders.py::TestStationBuilder	test_quality_metrics_healthy	Pass
	test_dl_report_builders.py::TestStationBuilder	test_quality_metrics_with_stale_details	Pass
	test_dl_report_builders.py::TestStationBuilder	test_reconciliation_no_results	Pass
	test_dl_report_builders.py::TestStationBuilder	test_reconciliation_with_failures	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_dl_report_builders.py::TestSectionBuilders	test_reconciliation_with_results	Pass
	test_dl_report_builders.py::TestSectionBuilders	test_remediation_compliant	Pass
	test_dl_report_builders.py::TestSectionBuilders	test_remediation_with_issues	Pass
	test_dl_report_builders.py::TestSectionBuilders	test_section_policy	Pass
	test_dl_report_builders.py::TestSectionBuilders	test_section_documentation	Pass
	test_dl_report_builders.py::TestSectionBuilders	test_section_documentation_empty_params	Pass
	test_dl_report_builders.py::TestSectionBuilders	test_topology	Pass
	test_dl_report_remediation_part1.py::TestCoverageBadge	test_coverage_badge_health	Pass
	test_dl_report_remediation_part1.py::TestCoverageBadge	test_degraded_on_failures	Pass
	test_dl_report_remediation_part1.py::TestCoverageBadge	test_inconsistent_on_broken_chain	Pass
	test_dl_report_remediation_part1.py::TestFailureBreakdown	test_failure_breakdown_renderable	Pass
	test_dl_report_remediation_part1.py::TestFailureBreakdown	test_failure_breakdown_bullet	Pass
	test_dl_report_remediation_part1.py::TestRemediationVerdict	test_compliant_verdict	Pass
	test_dl_report_remediation_part1.py::TestRemediationVerdict	test_non_compliant	Pass
	test_dl_report_remediation_part1.py::TestRemediationVerdict	test_stale_steps	Pass
	test_dl_report_remediation_part1.py::TestRemediationVerdict	test_stale_failure_metadata	Pass
	test_dl_report_remediation_part1.py::TestRemediationVerdict	test_shows_strengths_and_weaknesses	Pass
	test_dl_report_remediation_part2.py::TestExecSummaryVerdict	test_compliant	Pass
	test_dl_report_remediation_part2.py::TestExecSummaryVerdict	test_non_compliant	Pass
	test_dl_report_remediation_part2.py::TestExecSummaryVerdict	test_shows_verdict	Pass
	test_dl_report_remediation_part2.py::TestLoadLineageResults	test_loads_existing	Pass
	test_dl_report_remediation_part2.py::TestLoadLineageResults	test_returns_dict_when_missing	Pass
	test_dl_report_remediation_part2.py::TestLoadManifest	test_loads_valid_manifest_json	Pass
	test_dl_report_remediation_part2.py::TestLoadManifest	test_loads_existing_file	Pass
	test_dl_report_remediation_part2.py::TestLoadManifest	test_returns_dict	Pass
	test_dl_report_remediation_part2.py::TestMain	test_main_creates_audit_dir	Pass
	test_dl_report_remediation_part2.py::TestMain	test_main_prints_summary	Pass
	test_dl_report_remediation_part2.py::TestMain	test_main_produces_pdf	Pass
	test_dl_report_remediation_part2.py::TestRetentionPrinciples	test_retention_principle_lineage_reproducibility	Pass
	test_dl_report_verdicts.py::TestCoverageVerdict	test_healthy	Pass
	test_dl_report_verdicts.py::TestCoverageVerdict	test_consistent_when_all_clear	Pass
	test_dl_report_verdicts.py::TestCoverageVerdict	test_degraded_on_stale_steps	Pass
	test_dl_report_verdicts.py::TestCoverageVerdict	test_degraded_on_test_failures	Pass
	test_dl_report_verdicts.py::TestCoverageVerdict	test_health_colour_mapping	Pass
	test_dl_report_verdicts.py::TestCoverageVerdict	test_inconsistent_on_broken_chain	Pass
	test_dl_report_verdicts.py::TestCoverageVerdict	test_compliant_when_all_pass	Pass
	test_dl_report_verdicts.py::TestCoverageVerdict	test_empty_lineage_results	Pass
	test_dl_report_verdicts.py::TestCoverageVerdict	test_non_compliant_on_broken_chain	Pass
	test_dl_report_verdicts.py::TestCoverageVerdict	test_non_compliant_on_majority_failures	Pass
	test_dl_report_verdicts.py::TestCoverageVerdict	test_non_compliant_on_missing_steps	Pass
	test_dl_report_verdicts.py::TestCoverageVerdict	test_partially_compliant_on_test_failures	Pass
	test_dl_report_verdicts.py::TestCoverageVerdict	test_partially_compliant_when_no_tests_run	Pass
	test_dl_report_verdicts.py::TestRemediationVerdict	test_failure_metadata_covers_key_tests	Pass
	test_dl_report_verdicts.py::TestRemediationVerdict	test_retention_principles_principle_column	Pass
	test_dl_report_verdicts.py::TestTopologyOwners	test_owners_covers_all_graph_steps	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_dl_report_verdicts.py::TestTopologyOverlay	test_topology_overlay_renders_owner	Pass
	test_duplication_report.py::TestAnalyse	test_analyse_empty_report	Pass
	test_duplication_report.py::TestAnalyse	test_analyse_fmt_stats_has_python	Pass
	test_duplication_report.py::TestAnalyse	test_analyse_largest_line_numbers_are_integers	Pass
	test_duplication_report.py::TestAnalyse	test_analyse_largest_line_numbers_not_question_mark	Pass
	test_duplication_report.py::TestAnalyse	test_analyse_largest_sorted_descending	Pass
	test_duplication_report.py::TestAnalyse	test_analyse_num_clones_correct	Pass
	test_duplication_report.py::TestAnalyse	test_analyse_returns_dict	Pass
	test_duplication_report.py::TestAnalyse	test_analyse_total_present	Pass
	test_duplication_report.py::TestAnalyse	test_analyse_worst_populated	Pass
	test_duplication_report.py::TestMain	test_main_handles_jscpd_error	Pass
	test_duplication_report.py::TestMain	test_main_prints_stats	Pass
	test_duplication_report.py::TestMain	test_main_produces_pdf	Pass
	test_duplication_report.py::TestMakePdf	test_makepdf_good_rating_for_low_pct	Pass
	test_duplication_report.py::TestMakePdf	test_makepdf_high_rating_report	Pass
	test_duplication_report.py::TestMakePdf	test_makepdf_header	Pass
	test_duplication_report.py::TestMakePdf	test_makepdf_nonempty	Pass
	test_duplication_report.py::TestMakePdf	test_makepdf_returns_bytes	Pass
	test_model_risk_report_cover.py::TestBuildCover	test_build_cover_basic	Pass
	test_model_risk_report_cover.py::TestBuildCover	test_build_cover_no_bcbs	Pass
	test_model_risk_report_cover.py::TestBuildCover	test_build_cover_no_tests	Pass
	test_model_risk_report_cover.py::TestBuildCover	test_build_cover_with_bcbs	Pass
	test_model_risk_report_cover.py::TestBuildCover	test_build_cover_with_tests	Pass
	test_model_risk_report_cover.py::TestBuildExecSummary	test_build_exec_summary_all_pass	Pass
	test_model_risk_report_cover.py::TestBuildExecSummary	test_build_exec_summary_failures	Pass
	test_model_risk_report_cover.pyTestBuildExecSummary	test_build_exec_summary_no_coverage	Pass
	test_model_risk_report_cover.pyTestBuildExecSummary	test_build_exec_summary_no_tests	Pass
	test_model_risk_report_cover.pyTestBuildExecSummary	test_build_exec_summary_renders	Pass
	test_model_risk_report_cover.pyTestBuildInventory	test_build_inventory_empty	Pass
	test_model_risk_report_cover.pyTestBuildInventory	test_build_inventory_no_risk_ratings	Pass
	test_model_risk_report_cover.pyTestBuildInventory	test_build_inventory_rag_colours	Pass
	test_model_risk_report_cover.pyTestBuildInventory	test_build_inventory_renders	Pass
	test_model_risk_report_cover.pyTestBuildInventory	test_build_inventory_risk_ratings	Pass
	test_model_risk_report_cover.pyTestBuildValidation	test_build_validation_all_reviewers_assigned	Pass
	test_model_risk_report_cover.pyTestBuildValidation	test_build_validation_high_assumptions	Pass
	test_model_risk_report_cover.pyTestBuildValidation	test_build_validation_no_assumptions	Pass
	test_model_risk_report_cover.pyTestBuildValidation	test_build_validation_no_questions	Pass
	test_model_risk_report_cover.pyTestBuildValidation	test_build_validation_peer_reviewer_gaps	Pass
	test_model_risk_report_cover.pyTestBuildValidation	test_build_validation_renders	Pass
	test_model_risk_report_evidence.pyTestBuildAuditTrail	test_build_audit_trail_date_range	Pass
	test_model_risk_report_evidence.pyTestBuildAuditTrail	test_build_audit_trail_empty	Pass
	test_model_risk_report_evidence.pyTestBuildAuditTrail	test_build_audit_trail_timestamps	Pass
	test_model_risk_report_evidence.pyTestBuildAuditTrail	test_build_audit_trail_renders	Pass
	test_model_risk_report_evidence.pyTestBuildDocs	test_build_docs_empty	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_model_risk_report_evidence.py	TestBuildDocumentlist	Pass
	test_model_risk_report_evidence.py	TestBuildDocrenders	Pass
	test_model_risk_report_evidence.py	TestBuildRecommendations_allclean	Pass
	test_model_risk_report_evidence.py	TestBuildRecommendations_behind_at_risk	Pass
	test_model_risk_report_evidence.py	TestBuildRecommendations_overdue_remediation	Pass
	test_model_risk_report_evidence.py	TestBuildRecommendations_peer_reviewer_gap	Pass
	test_model_risk_report_evidence.py	TestBuildRecommendations_backup_gaps	Pass
	test_model_risk_report_evidence.py	TestBuildRecommendations_renders	Pass
	test_model_risk_report_evidence.py	TestBuildRecommendations_test_failures	Pass
	test_model_risk_report_evidence.py	TestBuildRecommendations_validation_gap	Pass
	test_model_risk_report_evidence.py	TestBuildTestEvidence	Pass
	test_model_risk_report_evidence.py	TestBuildTestEvidence_sensitivity_generators	Pass
	test_model_risk_report_evidence.py	TestBuildTestEvidence_per_model_coverage	Pass
	test_model_risk_report_evidence.py	TestBuildTestEvidence_renders	Pass
	test_model_risk_report_evidence.py	TestBuildTestEvidence_with_coverage	Pass
	test_model_risk_report_evidence.py	TestBuildTestEvidence_with_sensitivity_generators	Pass
	test_model_risk_report_mrc.py	TestBuildBCBS239_compliant	Pass
	test_model_risk_report_mrc.py	TestBuildBCBS239_at_risk	Pass
	test_model_risk_report_mrc.py	TestBuildBCBS239_principles	Pass
	test_model_risk_report_mrc.py	TestBuildBCBS239_renders	Pass
	test_model_risk_report_mrc.py	TestBuildMRC_completed_actions_excluded	Pass
	test_model_risk_report_mrc.py	TestBuildMRC_decisions	Pass
	test_model_risk_report_mrc.py	TestBuildMRC_meetings	Pass
	test_model_risk_report_mrc.py	TestBuildMRC_open_actions	Pass
	test_model_risk_report_mrc.py	TestBuildMRC_renders	Pass
	test_model_risk_report_mrc.py	TestBuildMRC_with_decisions	Pass
	test_model_risk_report_mrc.py	TestBuildRAGI_escalation_triggers_dict	Pass
	test_model_risk_report_mrc.py	TestBuildRAGI_escalation_triggers_string	Pass
	test_model_risk_report_mrc.py	TestBuildRAGI_missing_backup	Pass
	test_model_risk_report_mrc.py	TestBuildRAGI_roles	Pass
	test_model_risk_report_mrc.py	TestBuildRAGI_renders	Pass
	test_model_risk_report_mrc.py	TestBuildRAGI_with_activities	Pass
	test_model_risk_report_mrc.py	TestBuildRemediation_all_completed	Pass
	test_model_risk_report_mrc.py	TestBuildRemediation_empty	Pass
	test_model_risk_report_mrc.py	TestBuildRemediation_overdue_highlighted	Pass
	test_model_risk_report_mrc.py	TestBuildRemediation_priority_sorting	Pass
	test_model_risk_report_mrc.py	TestBuildRemediation_renders	Pass
	test_model_risk_report_pdf.py	TestCreatePdfReport_fallback	Pass
	test_model_risk_report_pdf.py	TestCreatePdfReport_audit_dir	Pass
	test_model_risk_report_pdf.py	TestCreatePdfReport_pdf	Pass
	test_model_risk_report_pdf.py	TestCreatePdfReport_empty_data	Pass
	test_model_risk_report_pdf.py	TestCreatePdfReport_summary	Pass
	test_model_risk_report_pdf.py	TestCreatePdfReport_with_real_collect	Pass
	test_model_risk_report_pdf.py	TestFlagCritical_log_single_event	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_model_risk_report_pdf.py::TestRiskCases	test_all_zero_max_score	Pass
	test_model_risk_report_pdf.py::TestRiskCases	test_no_coverage	Pass
	test_model_risk_report_pdf.py::TestRiskCases	test_model_with_long_name	Pass
	test_model_risk_report_pdf.py::TestRiskCases	test_model_with_no_name	Pass
	test_model_risk_report_pdf.py::TestRiskCases	test_mitigations_with_missing_fields	Pass
	test_model_risk_report_pdf.py::TestRiskCases	test_multiple_models_risk_ratings	Pass
	test_model_risk_report_pdf.py::TestRiskCases	test_no_activities_no_triggers	Pass
	test_model_risk_report_pdf.py::TestRiskCases	test_mitigation_null_due_date	Pass
	test_model_risk_report_pdf.py::TestRiskCases	test_validation_question_no_short_label	Pass
	test_model_risk_report_pdf.py::TestStationBuilders	test_all_builders_exported	Pass
	test_model_risk_report_pdf.py::TestStationBuilders	test_station_list	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_all_keys	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_audits_files	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_audits_files_missing_dir	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_sensitivity_generators	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_sensitivity_generators_no_dir	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_audits_log	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_audits_log_dict_fallback	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_codes	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_coverage	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_coverage_corrupt	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_coverage_missing	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_inventory	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_inventory_bad_json	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_inventory_missing	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_judges	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_judges_corrupt	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_judges_missing	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_meetings	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_meetings_dict_fallback	Pass
	test_model_risk_report_styles.py::TestDataLoaders	test_load_raci	Pass
	test_model_risk_report_styles.py::TestStyles	test_colour_constants	Pass
	test_model_risk_report_styles.py::TestStyles	test_get_styles_keys	Pass
	test_model_risk_report_styles.py::TestStyles	test_risk_colour_maps	Pass
	test_model_risk_report_styles.py::TestStyles	test_section_rule_appends_flowables	Pass
	test_model_risk_report_styles.py::TestStyles	test_tblstyle_returns_tablestyle	Pass
	test_blotter_data_part1.py::TestGaugeHubDate	test_curve_points_present	Pass
	test_blotter_data_part1.py::TestGaugeHubDate	test_hub_gauge_has_required_fields	Pass
	test_blotter_data_part1.py::TestGaugeHubDate	test_file_exists	Pass
	test_blotter_data_part1.py::TestGaugeHubDate	test_hub_trigger_ordering	Pass
	test_blotter_data_part1.py::TestGaugeHubDate	test_hub_count_at_least_40	Pass
	test_blotter_data_part1.py::TestGaugeHubDate	test_hub_ids_are_well_formed	Pass
	test_blotter_data_part1.py::TestGaugeHubDate	test_hazard_curves_key	Pass
	test_blotter_data_part1.py::TestGaugeHubDate	test_hazard_rate_ordering	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_blotter_data_part1.py::TestGaugeHazard	test_hazard_rates_positive	Pass
	test_blotter_data_part1.py::TestGaugeHazard	test_hazard_structure_present	Pass
	test_blotter_data_part1.py::TestMarketStateDates	test_state_dates_not_empty	Pass
	test_blotter_data_part1.py::TestMarketStateFile	test_file_exists	Pass
	test_blotter_data_part1.py::TestMarketStateFile	test_has_required_keys	Pass
	test_blotter_data_part1.py::TestMarketStateFile	test_hazard_term_structure_has_triggers	Pass
	test_blotter_data_part1.py::TestMarketStateFile	test_hazard_term_structure_not_empty	Pass
	test_blotter_data_part1.py::TestMarketStateFile	test_last_updated_present	Pass
	test_blotter_data_part1.py::TestMarketStateFile	test_yield_curve_has_all_tenors	Pass
	test_blotter_data_part1.py::TestMarketStateFile	test_yield_curve_rates_in_range	Pass
	test_blotter_data_part2.py::TestEODSnapshotData	test_snapshot_dates_are_business_days	Pass
	test_blotter_data_part2.py::TestEODSnapshotData	test_snapshot_directory_exists	Pass
	test_blotter_data_part2.py::TestEODSnapshotData	test_snapshot_files_are_valid_json	Pass
	test_blotter_data_part2.py::TestEODSnapshotData	test_snapshots_have_required_keys	Pass
	test_blotter_data_part2.py::TestEODSnapshotData	test_snapshot_trade_match_json	Pass
	test_blotter_data_part2.py::TestEODSnapshotData	test_snapshot_count	Pass
	test_blotter_data_part2.py::TestPastTradeDeltas	test_trade_deltas_file_exists	Pass
	test_blotter_data_part2.py::TestPastTradeData	test_each_trade_has_physical_swap_root	Pass
	test_blotter_data_part2.py::TestPastTradeData	test_each_trade_has_swap_id	Pass
	test_blotter_data_part2.py::TestPastTradeDeltas	test_trade_directory_exists	Pass
	test_blotter_data_part2.py::TestPastTradeDeltas	test_swap_id_matches_filename	Pass
	test_blotter_data_part2.py::TestPastTradeDeltas	test_trade_count_thames_central	Pass
	test_blotter_data_part2.py::TestPastTradeDeltas	test_trade_statuses_are_valid	Pass
	test_blotter_data_part2.py::TestPastTradeDeltas	test_trade_deltas_are_present	Pass
	test_blotter_data_part2.py::TestPastTradeDeltas	test_trade_deltas_have_gauge_set	Pass
	test_id_consistency_gauges.py::TestClassifierGauge	test_classifier_gauge_ids_match	Skip
	test_id_consistency_gauges.py::TestClassifierGauge	test_classifier_summary_exists	Skip
	test_id_consistency_gauges.py::TestGaugeDCs	test_gauge_dcs_in_has_gauges	Pass
	test_id_consistency_gauges.py::TestGaugeDCs	test_gauge_dcs_match_gauges	Pass
	test_id_consistency_gauges.py::TestHazardCurves	test_hazard_curves_match_gauges	Pass
	test_id_consistency_gauges.py::TestHazardCurves	test_hazard_curves_subset_of_gauges	Pass
	test_id_consistency_gauges.py::TestTradeGauges	test_trade_gauges_exist_in_gauge_json	Pass
	test_id_consistency_gauges.py::TestTradeGauges	test_trade_gauges_have_hazard_curves	Pass
	test_id_consistency_gauges.py::TestPropertyDCs	test_property_dcs_in_header	Pass
	test_id_consistency_gauges.py::TestPropertyDCs	test_property_dcs_has_properties	Pass
	test_id_consistency_gauges.py::TestPropertyDCs	test_property_dcs_files_match_properties	Pass
	test_id_consistency_pipeline_part2.py::TestPipelineOutputs	test_all_pipeline_outputs_exist	Pass
	test_id_consistency_pipeline_part2.py::TestPipelineOutputs	test_pipeline_outputs_match_stress_data	Skip
	test_id_consistency_pipeline_part2.py::TestPipelineOutputs	test_no_empty_output_directories	Pass
	test_id_consistency_pipeline_part2.py::TestStressDCs	test_stress_dcs_from_sequences	Pass
	test_id_consistency_pipeline_part2.py::TestPortfolioDCs	test_portfolio_dcs_from_sequences	Pass
	test_id_consistency_pipeline_part2.py::TestDataLineage	test_dependency_graph_complete	Pass
	test_id_consistency_pipeline_part2.py::TestDataLineage	test_lineage_files_exist	Pass
	test_id_consistency_pipeline_part2.py::TestDataLineage	test_data_lineage_exists	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_id_consistency_pipeline_part2.py::TestDatabaseIntegrity	test_output_hashes_current	Pass
	test_id_consistency_pipeline_part2.py::TestGenerationIsDeterministic	test_generation_is_deterministic	Skip
	test_lineage_manifest.py::TestGetCurrentRunId	test_format	Pass
	test_lineage_manifest.py::TestHashArtifact	test_hash_directory	Pass
	test_lineage_manifest.py::TestHashArtifact	test_hash_file	Pass
	test_lineage_manifest.py::TestHashArtifact	test_hash_missing	Pass
	test_lineage_manifest.py::TestHashDirectory	test_hash_empty_directory	Pass
	test_lineage_manifest.py::TestHashDirectory	test_hash_multiple_files_deterministic	Pass
	test_lineage_manifest.py::TestHashDirectory	test_hash_respects_pattern	Pass
	test_lineage_manifest.py::TestHashDirectory	test_hash_single_file	Pass
	test_lineage_manifest.py::TestHashFile	test_hash_binary_file	Pass
	test_lineage_manifest.py::TestHashFile	test_hash_empty_file	Pass
	test_lineage_manifest.py::TestHashFile	test_hash_known_content	Pass
	test_lineage_manifest.py::TestLoadManifest	test_load_corrupt_json_returns_skeleton	Pass
	test_lineage_manifest.py::TestLoadManifest	test_load_missing_returns_skeleton	Pass
	test_lineage_manifest.py::TestLoadManifest	test_load_valid_manifest	Pass
	test_lineage_manifest.py::TestRecordStep	test_record_and_reload	Pass
	test_lineage_manifest.py::TestRecordStep	test_record_with_auto_run_id	Pass
	test_lineage_manifest.py::TestSaveManifest	test_save_cleans_up_on_failure	Pass
	test_lineage_manifest.py::TestSaveManifest	test_save_creates_file	Pass
	test_lineage_manifest.py::TestSaveManifest	test_save_overwrites_existing	Pass
	test_lineage_prerequisites_part1.py::TestOutputsExist	test_all_outputs_exist_empty	Pass
	test_lineage_prerequisites_part1.py::TestOutputsExist	test_all_outputs_exist_missing	Pass
	test_lineage_prerequisites_part1.py::TestOutputsExist	test_all_outputs_exist_present_non_empty	Pass
	test_lineage_prerequisites_part1.py::TestOutputsExist	test_some_outputs_exist_missing	Pass
	test_lineage_prerequisites_part1.py::TestOutputsExist	test_some_outputs_exist_present	Pass
	test_lineage_prerequisites_part1.py::TestOutputsExist	test_partial_outputs_missing	Pass
	test_lineage_prerequisites_part1.py::TestUnknownStep	test_unknown_step	Pass
	test_lineage_prerequisites_part1.py::TestUpstreamFresh	test_all_fresh_no_prerequisites	Pass
	test_lineage_prerequisites_part1.py::TestUpstreamFresh	test_missing_upstream_files_exist	Pass
	test_lineage_prerequisites_part1.py::TestUpstreamFresh	test_missing_upstream_files	Pass
	test_lineage_prerequisites_part1.py::TestUtilityFunctions	test_resolve_target	Pass
	test_lineage_prerequisites_part1.py::TestUtilityFunctions	test_hash_dir_fallback	Pass
	test_lineage_prerequisites_part1.py::TestUtilityFunctions	test_deps_returns_empty	Pass
	test_lineage_prerequisites_part1.py::TestWalkDownstream	test_walk_downstream	Pass
	test_lineage_prerequisites_part1.py::TestWalkDownstream	test_target_not_in_prerequisites	Pass
	test_lineage_prerequisites_part1.py::TestWalkDownstream	test_topological_order	Pass
	test_lineage_prerequisites_part1.py::TestWalkDownstream	test_transitive_chain	Pass
	test_lineage_prerequisites_part1.py::TestWalkDownstream	test_unknown_prerequisites_empty	Pass
	test_lineage_prerequisites_part2.py::TestFullMode	test_full_mode_skips_prereq_resolution	Pass
	test_lineage_prerequisites_part2.py::TestPostIntegration	test_single_step_integration	Pass
	test_lineage_prerequisites_part2.py::TestPrintPrereqs	test_print_prereq_flags	Pass
	test_lineage_prerequisites_part2.py::TestPrintPrereqs	test_print_model_skipping_prereq_resolution	Pass
	test_lineage_query.py::TestGetFileInfo	test_file_with_no_producer_entry	Pass
	test_lineage_query.py::TestGetFileInfo	test_unknown_file	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_lineage_query.py::TestGetFileLineage	test_unknown_file	Pass
	test_lineage_query.py::TestGetStepLineage	test_mid_step	Pass
	test_lineage_query.py::TestGetStepLineage	test_root_step	Pass
	test_lineage_query.py::TestGetStepLineage	test_step_not_in_manifest	Pass
	test_lineage_query.py::TestInputsOfStep	test_known_inputs	Pass
	test_lineage_query.py::TestOutputsOfStep	test_known_outputs	Pass
	test_lineage_query.py::TestTraceDataPoint	test_gauge_id_chain	Pass
	test_lineage_query.py::TestTraceDataPoint	test_property_id_chain	Pass
	test_lineage_query.py::TestTraceDataPoint	test_unknown_data_type	Pass
	test_lineage_query.py::TestWalkUpstream	test_cycle_safety	Pass
	test_lineage_query.py::TestWalkUpstream	test_deep_chain	Pass
	test_lineage_query.py::TestWalkUpstream	test_mid_chain	Pass
	test_lineage_query.py::TestWalkUpstream	test_root_has_no_upstream	Pass
	test_lineage_repair.py::TestRepairManifest	test_repair_empty_dir_skipped	Pass
	test_lineage_repair.py::TestRepairManifest	test_repair_from_scratch	Pass
	test_lineage_repair.py::TestRepairManifest	test_repair_hashes_are_current	Pass
	test_lineage_repair.py::TestRepairManifest	test_repair_idempotent	Pass
	test_lineage_repair.py::TestRepairManifest	test_repair_inputs_hashed_correctly	Pass
	test_lineage_repair.py::TestRepairManifest	test_repair_preserves_existing_metadata	Pass
	test_lineage_repair.py::TestRepairManifest	test_repair_run_id_format	Pass
	test_lineage_repair.py::TestRepairManifest	test_repair_skips_missing_outputs	Pass
	test_lineage_repair.py::TestRepairManifest	test_repair_updates_runs_index	Pass
	test_lineage_validation_part1.py::TestCheckInputs	test_no_hash_for_input	Pass
	test_lineage_validation_part1.py::TestCheckInputs	test_never_run	Pass
	test_lineage_validation_part1.py::TestCheckInputs	test_when_hashes_match	Pass
	test_lineage_validation_part1.py::TestCheckInputs	test_no_hash_recorded_for_producer_output	Pass
	test_lineage_validation_part1.py::TestCheckInputs	test_never_run	Pass
	test_lineage_validation_part1.py::TestCheckStep	test_always_fresh	Pass
	test_lineage_validation_part1.py::TestCheckStep	test_when_hashes_differ	Pass
	test_lineage_validation_part1.py::TestCheckStep	test_unknown_step	Pass
	test_lineage_validation_part1.py::TestCheckPrerequisites	test_all_present	Pass
	test_lineage_validation_part1.py::TestCheckPrerequisites	test_missing_upstream	Pass
	test_lineage_validation_part1.py::TestGetStaleDownstream	test_gauges_affects_many	Pass
	test_lineage_validation_part1.py::TestGetStaleDownstream	test_leaf_has_no_downstream	Pass
	test_lineage_validation_part1.py::TestGetStaleDownstream	test_one_stale	Pass
	test_lineage_validation_part2.py::TestCompletePipeline	test_complete_pipeline	Pass
	test_lineage_validation_part2.py::TestCompletePipeline	test_empty_pipeline_complete	Pass
	test_lineage_validation_part2.py::TestCompletePipeline	test_missing_file	Pass
	test_lineage_validation_part2.py::TestCompletePipeline	test_partial_pipeline	Pass
	test_lineage_validation_part2.py::TestDependencyGraph	test_every_input_has_producer_in_transitive_deps	Pass
	test_lineage_validation_part2.py::TestDependencyGraph	test_dependency_graph_integrity	Pass
	test_lineage_validation_part2.py::TestValidateFullChain	test_consistent_chain	Pass
	test_lineage_validation_part2.py::TestValidateFullChain	test_empty_manifest	Pass
	test_lineage_validation_part2.py::TestValidateFullChain	test_stale_step_detected	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_prs_pricer_data_lineage.py::	TestCDMSourceFields::test_has_flood_zone	Pass
	test_prs_pricer_data_lineage.py::	TestCDMSourceFields::test_has_ground_level	Pass
	test_prs_pricer_data_lineage.py::	TestCDMSourceFields::test_has_location	Pass
	test_prs_pricer_data_lineage.py::	TestCDMSourceFields::test_has_properties	Pass
	test_prs_pricer_data_lineage.py::	TestCDMSourceFields::test_has_reference_gauges	Pass
	test_prs_pricer_data_lineage.py::	TestCalibrationConsistency::test_all_properties_have_complete_phcdata	Pass
	test_prs_pricer_data_lineage.py::	TestCalibrationConsistency::test_calibration_propertyts_to_propertyhc	Pass
	test_prs_pricer_data_lineage.py::	TestCalibrationConsistency::test_flood_counts_matches_propertyts_events	Pass
	test_prs_pricer_data_lineage.py::	TestCalibrationConsistency::test_flood_zone_propertyts_to_propertyhc	Pass
	test_prs_pricer_data_lineage.py::	TestCalibrationConsistency::test_gauge_id_propertyts_to_propertyhc	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_depth_thresholds	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_elevation	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_flood_count	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_flood_zone	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_flood_zone_matches_propertyts	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_floor_level	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_geo_params_is_none	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_has_geo_is_false	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_location	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_nearest_gauges_has_basis_bps	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_nearest_gauges_has_flood_counts	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_nearest_gauges	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_only_severe_threshold	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_pricing_methods_is_event_count	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_property_id	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_spread_equals_event_count_formula	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_spread_not_negative	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_summary	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_term_structure_exists	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_term_structure_has_severe	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyHCFields::test_term_structure_is_flat	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyTSFields::test_flood_zone_derived_from_synthetic	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyTSFields::test_has_elevation	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyTSFields::test_has_flood_events	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyTSFields::test_has_flood_zone	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyTSFields::test_has_floor_level	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyTSFields::test_has_location	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyTSFields::test_has_nearest_gauges	Pass
	test_prs_pricer_data_lineage.py::	TestPropertyTSFields::test_has_property_id	Pass
	test_prs_pricer_data_lineage.py::	TestSpreadDecomposition::test_lineage	Pass
	test_prs_pricer_data_lineage.py::	TestSpreadDecomposition::test_lineage_basis	Pass
	test_prs_pricer_data_lineage.py::	TestSpreadDecomposition::test_lineage	Pass
	test_prs_pricer_data_lineage.py::	TestSpreadDecomposition::test_lineage_basis	Pass
	test_prs_pricer_data_lineage.py::	TestGaugeSpreadPresent::test_lineage	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_prs_pricer_data_lineage.py::	TestSpreadDrivesPricing::test_spread_matches_term_structure	Pass
	test_prs_pricer_data_lineage.py::	TestSpreadDrivesPricing::test_spread_present	Pass
	test_prs_pricer_data_lineage.py::	TestSpreadDrivesPricing::test_source_files_exist	Pass
	test_prs_pricer_data_lineage.py::	TestSpreadDrivesPricing::test_spread_present_in_lineage	Pass
	test_prs_pricer_data_lineage.py::	TestSpreadDrivesPricing::test_spread_present_in_lineage	Pass
	test_prs_pricer_data_lineage.py::	TestSyntheticGaugeCorrectness::test_nearest_gauge	Pass
	test_prs_pricer_data_lineage.py::	TestSyntheticGaugeCorrectness::test_consistent_with_synthetic_offset	Pass
	test_prs_pricer_data_lineage.py::	TestTerrainGridMetadata::test_has_terrain_grid	Pass
	test_prs_pricer_data_lineage.py::	TestTerrainGridMetadata::test_distances	Pass
	test_prs_pricer_data_lineage.py::	TestTerrainGridMetadata::test_elevations	Pass
	test_prs_pricer_data_lineage.py::	TestTerrainGridMetadata::test_has_all_zones	Pass
	test_prs_pricer_data_lineage.py::	TestTerrainGridMetadata::test_spread_decreases_with_distance	Pass
	test_prs_pricer_data_lineage.py::	TestTerrainGridMetadata::test_zone3_gt_zone1_at_origin	Pass
	test_prs_pricer_data_lineage.py::	TestTerrainGridMetadata::test_zone_dimensions	Pass
	test_prs_pricer_data_lineage.py::	TestTerrainGridMetadata::test_zone_ordering_at_origin	Pass
	test_prs_pricer_data_lineage.py::	TestZoneSpreadConsistency::test_all_four_zones_represented	Pass
	test_prs_pricer_data_lineage.py::	TestZoneSpreadConsistency::test_spread_ordering_by_zone	Pass
	test_prs_pricer_data_lineage.py::	TestZoneSpreadConsistency::test_zone3_has_higher_avg_flood_count_than_zone_1	Pass
	test_prs_pricer_data_lineage.py::	TestZoneSpreadConsistency::test_zone3_has_higher_avg_spread_than_zone_1	Pass
	test_prs_pricer_data_lineage.py::	TestZoneSpreadConsistency::test_zone3_lower_avg_offset_than_zone_1	Pass
	test_prs_pricer_data_lineage.py::	TestZoneSpreadConsistency::test_zone_offset_ordering	Pass
	test_storm_data_classifiers.py::	TestGaugeDataFile::test_each_has_required_keys	Pass
	test_storm_data_classifiers.py::	TestGaugeDataFile::test_each_simulation_has_readings	Pass
	test_storm_data_classifiers.py::	TestGaugeDataFile::test_gauge_id_field_matches_filename	Pass
	test_storm_data_classifiers.py::	TestGaugeDataFile::test_gauge_data_dir_exists	Pass
	test_storm_data_classifiers.py::	TestGaugeDataFile::test_gauge_id_format	Pass
	test_storm_data_classifiers.py::	TestGaugeDataFile::test_gauge_data_not_empty	Pass
	test_storm_data_classifiers.py::	TestGaugeDataFile::test_readings_count_matches_simulation_hours	Pass
	test_storm_data_consistency.py::	TestStormDataConsistency::test_classifier_covers_gaugehc_gauges	Skip
	test_storm_data_consistency.py::	TestStormDataConsistency::test_classifier_consistent_for_stress_gauges	Skip
	test_storm_data_consistency.py::	TestStormDataConsistency::test_stress_data_covers_gauges_in_gaugehc	Skip
	test_storm_data_consistency.py::	TestStormDataConsistency::test_stress_data_consistent_overlap_propertyts	Skip
	test_storm_data_consistency.py::	TestStormDataConsistency::test_stress_data_consistent_subset_of_gaugets	Skip
	test_storm_data_storms.py::	TestStormFile::test_all_unique_gauge_ids_in_responses	Pass
	test_storm_data_storms.py::	TestStormFile::test_all_gauge_ids_use_gauge_prefix	Pass
	test_storm_data_storms.py::	TestStormFile::test_all_storm_ids_use_valid_prefix	Pass
	test_storm_data_storms.py::	TestStormFile::test_at_least_one_storm_exceeds_alert	Pass
	test_storm_data_storms.py::	TestStormFile::test_each_storm_has_gauge_responses	Pass
	test_storm_data_storms.py::	TestStormFile::test_each_storm_has_required_fields	Pass
	test_storm_data_storms.py::	TestStormFile::test_file_exists	Pass
	test_storm_data_storms.py::	TestStormFile::test_gauge_file_well_formed	Pass
	test_storm_data_storms.py::	TestStormFile::test_gauge_responses_have_required_fields	Pass
	test_storm_data_storms.py::	TestStormFile::test_storms_nonempty	Pass
	test_storm_data_storms.py::	TestStormFile::test_intensity_categories_present	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_storm_data_storms.py::TestStormFile	test_storm_count	Pass
	test_storm_data_storms.py::TestStormFile	test_peak_level_exceeds_base	Pass
	test_storm_data_storms.py::TestStormFile	test_peak_levels_positive	Pass
	test_storm_data_storms.py::TestStormFile	test_storm_ids_unique	Pass
	test_storm_data_storms.py::TestStormFile	test_storm_ids_well_formed	Pass
	test_storm_data_storms.py::TestStormFile	test_storms_cover_multiple_gauges	Pass
	test_storm_data_storms.py::TestStormFile	test_storms_sorted_by_severity_descending	Pass
	test_storm_data_storms.py::TestStormFile	test_storms_key_matches_list_length	Pass
	test_storm_data_storms.py::TestTriggerFile	test_trigger_summary_max_trigger_valid	Pass
	test_storm_data_storms.py::TestTriggerFile	test_trigger_summary_populated_for_all_storms	Pass
	test_flood_propagation_diagnostics.py::TestGaugeTimeseries	test_gauges_have_flood_simulation	Pass
	test_flood_propagation_diagnostics.py::TestGaugeTimeseries	test_gauges_have_storm_responses	Pass
	test_flood_propagation_diagnostics.py::TestGaugeTimeseries	test_some_storms_exceed_alert	Pass
	test_flood_propagation_diagnostics.py::TestGaugeTimeseries	test_synth_gauges_have_storm_responses	Pass
	test_flood_propagation_diagnostics.py::TestNearestGauge	test_all_properties_have_nearest_gauges	Pass
	test_flood_propagation_diagnostics.py::TestNearestGauge	test_most_properties_near_gauge	Pass
	test_flood_propagation_diagnostics.py::TestNearestGauge	test_nearest_gauges_have_elevations	Pass
	test_flood_propagation_diagnostics.py::TestPropertyPropagation	test_build_property_propagation_produces_depth	Skip
	test_flood_propagation_diagnostics.py::TestFloodPropagation	test_flood_events_drop_expectation	Pass
	test_flood_propagation_diagnostics.py::TestFloodPropagation	test_depths_produce_flooding	Pass
	test_flood_propagation_diagnostics.py::TestFloodPropagation	test_property_receives_gauge_alert_storms	Pass
	test_flood_propagation_diagnostics.py::TestPortfolioFloodStats	test_portfolio_summary	Pass
	test_flood_propagation_diagnostics.py::TestPortfolioFloodStats	test_storm_count_vs_property_event_count	Pass
	test_gauge_loader.py::TestGaugeLoader	test_basic	Pass
	test_gauge_loader.py::TestGaugeLoader	test_bad_gauges_key_also_works	Pass
	test_gauge_loader.py::TestGaugeLoader	test_load_all_returns_list	Pass
	test_gauge_loader.py::TestGaugeLoader	test_load_missing_file_returns_empty	Pass
	test_gauge_loader.py::TestGaugeLoader	test_caching_works	Pass
	test_gauge_loader.py::TestGaugeLoader	test_clear_cache	Pass
	test_gauge_loader.py::TestGaugeLoader	test_force_reload	Pass
	test_gauge_loader.py::TestGaugeLoader	test_lookup_false	Pass
	test_gauge_loader.py::TestGaugeLoader	test_lookup_true	Pass
	test_gauge_loader.py::TestGaugeLoader	test_find_by_id_missing_returns_none	Pass
	test_gauge_loader.py::TestGaugeLoader	test_find_by_id_returns_entity	Pass
	test_gauge_loader.py::TestGaugeLoader	test_list_all_returns_summaries	Pass
	test_gauge_loader.py::TestGaugeLoader	test_get_status_returns_dict	Pass
	test_mortgage_loader.py::TestMortgageLoaderBase	test_load_all_returns_summaries	Pass
	test_mortgage_loader.py::TestMortgageLoaderBase	test_load_all_returns_list	Pass
	test_mortgage_loader.py::TestMortgageLoaderBase	test_find_by_id_lookup	Pass
	test_mortgage_loader.py::TestMortgageLoaderBase	test_find_by_id_property_nested_structure	Pass
	test_mortgage_loader.py::TestMortgageLoaderBase	test_find_by_id_missing_returns_none	Pass
	test_property_county.py::TestFindByCounty	test_find_by_county_case_insensitive	Pass
	test_property_county.py::TestFindByCounty	test_find_by_county_empty_portfolio	Pass
	test_property_county.py::TestFindByCounty	test_find_by_county_exact_match	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_property_county.py::TestFindByCounty	test_find_by_county_missing_county_field	Pass
	test_property_county.py::TestFindByCounty	test_find_by_county_no_match	Pass
	test_property_county.py::TestFindByCounty	test_find_by_county_partial_match	Pass
	test_property_loader.py::TestPropertyLoaderBasic	test_custom_filename	Pass
	test_property_loader.py::TestPropertyLoaderBasic	test_load_all_returns_list	Pass
	test_property_loader.py::TestPropertyLoaderBasic	test_missing_file_empty	Pass
	test_property_loader.py::TestPropertyLoaderBasic	test_find_by_id	Pass
	test_property_loader.py::TestPropertyLoaderBasic	test_load_all_has_property_id	Pass
	test_registry.py::TestCreateRegistry	test_creates_registry	Pass
	test_registry.py::TestCreateRegistry	test_with_file_overrides	Pass
	test_registry.py::TestLoaderRegistryCache	test_clear_all_caches	Pass
	test_registry.py::TestLoaderRegistryCache	test_cache_all_status	Pass
	test_registry.py::TestLoaderRegistryCache	test_invalidate_all_caches	Pass
	test_registry.py::TestLoaderRegistryDataFiles	test_check_data_files_all_missing	Pass
	test_registry.py::TestLoaderRegistryDataFiles	test_check_data_files_present	Pass
	test_registry.py::TestLoaderRegistryDataFiles	test_data_file_overrides	Pass
	test_registry.py::TestLoaderRegistryDataFiles	test_get_data_dir	Pass
	test_registry.py::TestLoaderRegistryLoaders	test_get_available_loaders	Pass
	test_registry.py::TestLoaderRegistryLoaders	test_get_gauge_loader	Pass
	test_registry.py::TestLoaderRegistryLoaders	test_get_loader_by_name	Pass
	test_registry.py::TestLoaderRegistryLoaders	test_get_loader_unknown_name_returns_none	Pass
	test_registry.py::TestLoaderRegistryLoaders	test_get_mortgage_loader	Pass
	test_registry.py::TestLoaderRegistryLoaders	test_get_property_loader	Pass
	test_registry.py::TestLoaderRegistryLoaders	test_get_storm_loader	Pass
	test_registry.py::TestLoaderRegistryLoaders	test_get_timeseries_loader	Pass
	test_registry.py::TestLoaderRegistryLoaders	test_loader_cached_on_repeated_access	Pass
	storm_loader_part1.py::TestFindByBasin	test_basin	Pass
	storm_loader_part1.py::TestFindByBasin	test_case_insensitive	Pass
	storm_loader_part1.py::TestFindByBasin	test_unknown_basin_returns_empty	Pass
	storm_loader_part1.py::TestFindByBasin	test_wpac_basin	Pass
	storm_loader_part1.py::TestFindByCategory	test_category_4	Pass
	storm_loader_part1.py::TestFindByCategory	test_category_3	Pass
	storm_loader_part1.py::TestFindByCategory	test_category_5	Pass
	storm_loader_part1.py::TestFindByCategory	test_no_matches_returns_empty	Pass
	storm_loader_part1.py::TestFindByName	test_case_insensitive	Pass
	storm_loader_part1.py::TestFindByName	test_exact_match	Pass
	storm_loader_part1.py::TestFindByName	test_no_match_returns_none	Pass
	storm_loader_part1.py::TestFindByName	test_partial_match	Pass
	storm_loader_part1.py::TestFindBySeason	test_missing_season_returns_empty	Pass
	storm_loader_part1.py::TestFindBySeason	test_season_2017	Pass
	storm_loader_part1.py::TestFindBySeason	test_season_with_multiple	Pass
	storm_loader_part1.py::TestGetEntityId	test_empty_entity_returns_none	Pass
	storm_loader_part1.py::TestGetEntityId	test_extracts_event_id	Pass
	storm_loader_part1.py::TestGetEntityId	test_missing_id_returns_none	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	storm_loader_part1.py::TestGetEntity	test_missing_fields_return_none	Pass
	storm_loader_part1.py::TestGetEntity	test_returns_summary_dict	Pass
	storm_loader_part2.py::TestGetImpact	test_missing_id_returns_none	Pass
	storm_loader_part2.py::TestGetImpact	test_returns_impact_dict	Pass
	storm_loader_part2.py::TestGetMajorHurricanes	test_includes_category_1_and_2	Pass
	storm_loader_part2.py::TestGetMajorHurricanes	test_includes_cat_3_4_5	Pass
	storm_loader_part2.py::TestGetMajorHurricanes	test_returns_category_3_plus	Pass
	storm_loader_part2.py::TestGetStormAffectingLocation	test_each_storm_returned_at_most_once	Pass
	storm_loader_part2.py::TestGetStormAffectingLocation	test_empty_for_remote_location	Pass
	storm_loader_part2.py::TestGetStormAffectingLocation	test_finds_nearby_storm	Pass
	storm_loader_part2.py::TestGetStormAffectingLocation	test_larger_radius_finds_more	Pass
	storm_loader_part2.py::TestGetTrack	test_missing_id_returns_none	Pass
	storm_loader_part2.py::TestGetTrack	test_returns_track_for_existing_storm	Pass
	storm_loader_part2.py::TestGetTrack	test_track_has_lat_lon	Pass
	test_base_loader_coverage.py::TestAssetGetEntity	test_get_entity_id_called	Pass
	test_base_loader_coverage.py::TestAssetGetEntity	test_get_entity_summary_called	Pass
	test_base_loader_coverage.py::TestCapabilityAliases	test_capability_summary_delegates_to_get_entity_summary	Pass
	test_base_loader_coverage.py::TestCapabilityAliases	test_capability_id_delegates_to_get_entity_id	Pass
	test_base_loader_coverage.py::TestCapabilityAliases	test_capability_class_keys_returns_class_attr	Pass
	test_base_loader_coverage.py::TestCapabilityAliases	test_capability_status_delegates_to_get_status	Pass
	test_base_loader_coverage.py::TestNormalizeNullNaN	test_normalize_null_nan_returns_empty	Pass
	test_base_loader_coverage.py::TestNormalizeNullNaN	test_normalize_list_negative_returns_empty	Pass
	test_base_loader_coverage.py::TestNormalizeNullNaN	test_normalize_list_string_returns_empty	Pass
	test_base_loader_coverage.py::TestNormalizeNullNaN	test_normalize_list_none_returns_empty	Pass
	test_lookups_gauge_property.py::TestBuildGaugeFloodInfo	test_empty_returns_empty	Pass
	test_lookups_gauge_property.py::TestBuildGaugeFloodInfo	test_gauge_data_supplements_missing_fields	Pass
	test_lookups_gauge_property.py::TestBuildGaugeFloodInfo	test_gauge_without_flood_info_skipped	Pass
	test_lookups_gauge_property.py::TestBuildGaugeFloodInfo	test_hazard_curves_only	Pass
	test_lookups_gauge_property.py::TestBuildGaugeFloodInfo	test_gauge_data_takes_precedence	Pass
	test_lookups_gauge_property.py::TestBuildPropertyFloodInfo	test_empty_returns_empty	Pass
	test_lookups_gauge_property.py::TestBuildPropertyFloodInfo	test_property_data_when_not_in_phc	Pass
	test_lookups_gauge_property.py::TestBuildPropertyFloodInfo	test_property_flood_hazard_data	Pass
	test_lookups_gauge_property.py::TestBuildPropertyFloodInfo	test_gauge_data_precedence_over_property_data	Pass
	test_lookups_gauge_property.py::TestClassifyPropertyRisk	test_classify_high_threshold_is_high	Pass
	test_lookups_gauge_property.py::TestClassifyPropertyRisk	test_classify_medium_risk	Pass
	test_lookups_gauge_property.py::TestClassifyPropertyRisk	test_classify_low_risk_is_low	Pass
	test_lookups_gauge_property.py::TestClassifyPropertyRisk	test_classify_threshold	Pass
	test_lookups_gauge_property.py::TestClassifyPropertyRisk	test_classify_property_risk	Pass
	test_lookups_gauge_property.py::TestClassifyPropertyRisk	test_classify_negative_risk	Pass
	test_lookups_ids.py::TestAnalyzeIdRelationships	test_all_nulls_returns_zeros	Pass
	test_lookups_ids.py::TestAnalyzeIdRelationships	test_counts_match_data	Pass
	test_lookups_ids.py::TestAnalyzeIdRelationships	test_flood_mortgages_counted	Pass
	test_lookups_ids.py::TestAnalyzeIdRelationships	test_flood_properties_overlap	Pass
	test_lookups_ids.py::TestAnalyzeIdRelationships	test_returns_required_keys	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_lookups_ids.py::TestExtractGaugeIds	test_extracts_ids	Pass
	test_lookups_ids.py::TestExtractGaugeIds	test_missing_gauge_id_skipped	Pass
	test_lookups_ids.py::TestExtractGaugeIds	test_none_returns_empty_set	Pass
	test_lookups_ids.py::TestExtractMortgageIds	test_extracts_ids	Pass
	test_lookups_ids.py::TestExtractMortgageIds	test_none_returns_empty_set	Pass
	test_lookups_ids.py::TestExtractMortgagePropertyIds	test_extracts_property_ids_from_mortgages	Pass
	test_lookups_ids.py::TestExtractMortgagePropertyIds	test_none_returns_empty_set	Pass
	test_lookups_ids.py::TestExtractPropertyIds	test_empty_items_returns_empty_set	Pass
	test_lookups_ids.py::TestExtractPropertyIds	test_extracts_ids	Pass
	test_lookups_ids.py::TestExtractPropertyIds	test_missing_property_id_skipped	Pass
	test_lookups_ids.py::TestExtractPropertyIds	test_none_returns_empty_set	Pass
	test_lookups_mortgage.py::TestBuildAllLookups	test_all_lookups_returns_empty_dicts	Pass
	test_lookups_mortgage.py::TestBuildAllLookups	test_all_lookups_returns_four_keys	Pass
	test_lookups_mortgage.py::TestBuildAllLookups	test_with_data_populates_lookups	Pass
	test_lookups_mortgage.py::TestBuildMortgageLookup	test_empty_dict_returns_empty	Pass
	test_lookups_mortgage.py::TestBuildMortgageLookup	test_mortgage_from_loan_details_fallback	Pass
	test_lookups_mortgage.py::TestBuildMortgageLookup	test_mortgage_without_property_id_skipped	Pass
	test_lookups_mortgage.py::TestBuildMortgageLookup	test_multiple_mortgages	Pass
	test_lookups_mortgage.py::TestBuildMortgagesByProperty	test_none_returns_empty	Pass
	test_lookups_mortgage.py::TestBuildMortgageLookup	test_none_returns_empty	Pass
	test_lookups_mortgage.py::TestBuildMortgageLookup	test_single_mortgage	Pass
	test_lookups_mortgage.py::TestBuildMortgageRiskInfo	test_basic_risk_info	Pass
	test_lookups_mortgage.py::TestBuildMortgageRiskInfo	test_mortgage_risk_info_mortgage_or_property_id_skipped	Pass
	test_lookups_mortgage.py::TestBuildMortgageRiskInfo	test_no_property_flood_info_uses_unknown	Pass
	test_lookups_mortgage.py::TestBuildMortgageRiskInfo	test_none_mortgage_returns_empty_structure	Pass
	test_lookups_mortgage.py::TestBuildMortgageRiskInfo	test_value_risk_calculation	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderBasic	test_loads_number_of_gauge_files	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderBasic	test_get_available_gauge_ids	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderBasic	test_get_available_gauge_ids_missing_dir	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderBasic	test_list_all_returns_summaries	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderBasic	test_load_all_from_directory	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderBasic	test_missing_dir_returns_empty	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderCache	test_clear_cache	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderCache	test_force_reload	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderCache	test_invalidate_cache	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderCache	test_load_timestamp_skipped_all_gauges_path	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderCache	test_load_timestamp_skipped_gauge_path	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderQueries	test_get_readings_in_range_all_gauges	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderQueries	test_get_readings_in_range_for_gauge	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderQueries	test_get_latest_reading	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderQueries	test_get_latest_reading_missing_returns_none	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderQueries	test_get_peak_reading	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderQueries	test_get_peak_reading_missing_returns_none	Pass
	test_timeseries_loader.py::TestTimeseriesLoaderQueries	test_get_readings_for_gauge	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_timeseries_loader.py::TestTimeseriesLoader	test_get_readings_for_gauge_by_name_fallback	Pass
	test_timeseries_loader.py::TestTimeseriesLoader	test_get_readings_missing_gauge_returns_empty	Pass
	test_timeseries_loader.py::TestTimeseriesLoader	test_get_storm_responses	Pass
	test_timeseries_loader.py::TestTimeseriesLoader	test_get_storm_responses_missing_gauge	Pass
	test_timeseries_loader.py::TestTimeseriesLoader	test_get_gauge_statistics_missing_gauge	Pass
	test_timeseries_loader.py::TestTimeseriesLoader	test_get_gauge_statistics_no_numeric_levels	Pass
	test_timeseries_loader.py::TestTimeseriesLoader	test_get_gauge_statistics_with_data	Pass
	test_timeseries_loader.py::TestTimeseriesLoader	test_exists_by_timestamp	Pass
	test_timeseries_loader.py::TestTimeseriesLoader	test_find_by_id_not_found_returns_none	Pass
	test_timeseries_loader.py::TestTimeseriesLoader	test_get_file_status_alias	Pass
	test_timeseries_loader.py::TestTimeseriesLoader	test_get_status_returns_dict	Pass
	audit.py::TestLogModelUsage	test_appends_entry	Pass
	audit.py::TestLogModelUsage	test_bad_audit_path_does_not_raise	Pass
	audit.py::TestLogModelUsage	test_corrupted_file_restarted	Pass
	audit.py::TestLogModelUsage	test_creates_audit_file	Pass
	audit.py::TestLogModelUsage	test_entry_has_required_keys	Pass
	audit.py::TestLogModelUsage	test_existing_file_appended	Pass
	audit.py::TestLogModelUsage	test_multiple_calls_accumulate	Pass
	audit.py::TestResolveModelId	test_case_insensitive	Pass
	audit.py::TestResolveModelId	test_depth_damage_maps_to_dd	Pass
	audit.py::TestResolveModelId	test_distribution_maps_to_si	Pass
	audit.py::TestResolveModelId	test_mkm_prefix_returned_as_is	Pass
	audit.py::TestResolveModelId	test_shorthand_gev_resolved	Pass
	audit.py::TestResolveModelId	test_shorthand_intensity_resolved	Pass
	audit.py::TestResolveModelId	test_shorthand_mortgage_resolved	Pass
	audit.py::TestResolveModelId	test_shorthand_prs_resolved	Pass
	audit.py::TestResolveModelId	test_spatial_maps_to_sp	Pass
	audit.py::TestResolveModelId	test_unknown_shorthand_returned_as_is	Pass
	audit.py::TestSanitiseParams	test_bool_passed_through	Pass
	audit.py::TestSanitiseParams	test_empty_params	Pass
	audit.py::TestSanitiseParams	test_float_passed_through	Pass
	audit.py::TestSanitiseParams	test_int_passed_through	Pass
	audit.py::TestSanitiseParams	test_long_list_summarised	Pass
	audit.py::TestSanitiseParams	test_none_passed_through	Pass
	audit.py::TestSanitiseParams	test_object_converted_to_string	Pass
	audit.py::TestSanitiseParams	test_short_list_converted_to_str	Pass
	audit.py::TestSanitiseParams	test_string_passed_through	Pass
	audit.py::TestSanitiseParams	test_tuple_treated_like_list	Pass
	depth_damage.py::TestCalculateFloodDepths	test_depths_severe_level_some_positive	Pass
	depth_damage.py::TestCalculateFloodDepths	test_depths_severe_level_returns_zeros	Pass
	depth_damage.py::TestCalculateFloodDepths	test_depths_capped_at_5m	Pass
	depth_damage.py::TestCalculateFloodDepths	test_empty_gauge_data_returns_zeros	Pass
	depth_damage.py::TestCalculateFloodDepths	test_no_gauge_data_returns_zeros	Pass
	depth_damage.py::TestCalculateFloodDepths	test_returns_correct_length	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	depth_damage.py::TestDepthDamageFunction	test_commercial_higher_damage_than_residential	Pass
	depth_damage.py::TestDepthDamageFunction	test_floor_level_reduces_damage	Pass
	depth_damage.py::TestDepthDamageFunction	test_large_depth_all_near_one	Pass
	depth_damage.py::TestDepthDamageFunction	test_output_clipped_to_0_1	Pass
	depth_damage.py::TestDepthDamageFunction	test_returns_array_same_length	Pass
	depth_damage.py::TestDepthDamageFunction	test_zero_depths_all_zero_damage	Pass
	depth_damage.py::TestScalarDepthDamage	test_damage_max_depth_returns_one	Pass
	depth_damage.py::TestScalarDepthDamage	test_between_control_points_interpolated	Pass
	depth_damage.py::TestScalarDepthDamage	test_damage_between_0_and_1	Pass
	depth_damage.py::TestScalarDepthDamage	test_half_meter_damage	Pass
	depth_damage.py::TestScalarDepthDamage	test_max_depth_returns_one	Pass
	depth_damage.py::TestScalarDepthDamage	test_monotonically_increasing	Pass
	depth_damage.py::TestScalarDepthDamage	test_negative_depth_returns_zero	Pass
	depth_damage.py::TestScalarDepthDamage	test_one_meter_damage	Pass
	depth_damage.py::TestScalarDepthDamage	test_returns_float	Pass
	depth_damage.py::TestScalarDepthDamage	test_small_depth_returns_small_damage	Pass
	depth_damage.py::TestScalarDepthDamage	test_zero_depth_returns_zero	Pass
	flood_risk_model.py::TestFloodRiskModel	test_basic_construction	Pass
	flood_risk_model.py::TestFloodRiskModel	test_correlation_matrix_built	Pass
	flood_risk_model.py::TestFloodRiskModel	test_custom_correlation_params	Pass
	flood_risk_model.py::TestFloodRiskModel	test_missing_elevation_warns_and_defaults	Pass
	flood_risk_model.py::TestFloodRiskModel	test_missing_floor_level_defaults_to_zero	Pass
	flood_risk_model.py::TestFloodRiskModel	test_missing_geometry_raises	Pass
	flood_risk_model.py::TestFloodRiskModel	test_missing_property_id_auto_assigned	Pass
	flood_risk_model.py::TestFloodRiskModel	test_missing_property_type_warns_and_defaults	Pass
	flood_risk_model.py::TestFloodRiskModel	test_missing_value_raises	Pass
	flood_risk_model.py::TestFloodRiskModel	test_properties_copied	Pass
	flood_risk_model.py::TestFloodRiskModel	test_with_gauge_data	Pass
	flood_risk_model.py::TestFloodRiskModelMethods	test_analyze_risk_clusters_returns_df	Pass
	flood_risk_model.py::TestFloodRiskModelMethods	test_analyze_spatial_concentration_returns_df	Pass
	flood_risk_model.py::TestFloodRiskModelMethods	test_calculate_advanced_metrics_returns_dict	Pass
	flood_risk_model.py::TestFloodRiskModelMethods	test_calculate_direct_impacts_returns_array	Pass
	flood_risk_model.py::TestFloodRiskModelMethods	test_calculate_flood_depths_returns_array	Pass
	flood_risk_model.py::TestFloodRiskModelMethods	test_haversine_distance_delegate	Pass
	flood_risk_model.py::TestFloodRiskModelMethods	test_simulate_portfolio_impact_returns_dict	Pass
	test_depth_damage_coverage.py::TestCalculateFloodDepthNeg	test_with_gauge_data_delegates_to_synthetic	Pass
	test_depth_damage_coverage.py::TestDepthDamageFunctionCoverage	test_commercial_gets_10%_factor	Pass
	test_depth_damage_coverage.py::TestDepthDamageFunctionCoverage	test_floor_level_adjustment	Pass
	test_depth_damage_coverage.py::TestDepthDamageFunctionCoverage	test_full_function_returns_correct_length	Pass
	test_depth_damage_coverage.py::TestDepthDamageFunctionCoverage	test_industrial_gets_0.9_factor	Pass
	test_depth_damage_coverage.py::TestDepthDamageFunctionCoverage	test_output_clipped_to_unit_interval	Pass
	test_depth_damage_coverage.py::TestDepthDamageFunctionCoverage	test_unknown_type_defaults_to_1.0	Pass
	test_depth_damage_coverage.py::TestDepthDamageFunctionCoverage	test_zero_depths_all_zero	Pass
	test_depth_damage_coverage.py::TestScalarDepthDamageCoverage	test_scalar_fallback	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_depth_damage_coverage.py::TestSingleDepthDamageFunction	test_just_below_first_positive_depth	Pass
	test_depth_damage_coverage.py::TestSyntheticFloodDepths	test_below_second_returns_zeros	Pass
	test_depth_damage_coverage.py::TestSyntheticFloodDepths	test_capped_at_5m	Pass
	test_depth_damage_coverage.py::TestSyntheticFloodDepths	test_distance_factor_applied	Pass
	test_depth_damage_coverage.py::TestSyntheticFloodDepths	test_negative_returns_zeros	Pass
	test_depth_damage_coverage.py::TestSyntheticFloodDepths	test_far_away_property_gets_zero	Pass
	test_depth_damage_coverage.py::TestSyntheticFloodDepths	test_finding_path_produces_positive_depths	Pass
	test_depth_damage_coverage.py::TestSyntheticFloodDepths	test_some_negative_returns_zeros	Pass
	test_hydrograph_part1.py::TestBuildPulseGaugeHydrograph	test_base_level_when_no_exceedance	Pass
	test_hydrograph_part1.py::TestBuildPulseGaugeHydrograph	test_peak_near_expected_hour	Pass
	test_hydrograph_part1.py::TestBuildPulseGaugeHydrograph	test_peak_value	Pass
	test_hydrograph_part1.py::TestGammaPdf	test_symmetric_for_small_alpha	Pass
	test_hydrograph_part1.py::TestGammaPdf	test_valid_alpha	Pass
	test_hydrograph_part1.py::TestGammaPdf	test_max_is_one[0.3]	Pass
	test_hydrograph_part1.py::TestGammaPdf	test_max_is_one[0.7]	Pass
	test_hydrograph_part1.py::TestGammaPdf	test_max_is_one[1.0]	Pass
	test_hydrograph_part1.py::TestGammaPdf	test_negative_u	Pass
	test_hydrograph_part1.py::TestGammaPdf	test_peak_at_alpha_squared[0.3]	Pass
	test_hydrograph_part1.py::TestGammaPdf	test_peak_at_alpha_squared[0.5]	Pass
	test_hydrograph_part1.py::TestGammaPdf	test_peak_at_alpha_squared[0.7]	Pass
	test_hydrograph_part1.py::TestGammaPdf	test_peak_at_alpha_squared[1.0]	Pass
	test_hydrograph_part1.py::TestGammaPdf	test_zero_at_zero	Pass
	test_hydrograph_part1.py::TestGammaPdf	test_zero_beyond_one	Pass
	test_hydrograph_part1.py::TestGammaPdf	test_matches_scipy	Pass
	test_hydrograph_part1.py::TestGammaPdf	test_zero_outside_window	Pass
	test_hydrograph_part1.py::TestInfiltration	test_fully_impervious	Pass
	test_hydrograph_part1.py::TestInfiltration	test_reduces_depth	Pass
	test_hydrograph_part1.py::TestInfiltration	test_saturates	Pass
	test_hydrograph_part1.py::TestInfiltration	test_zero_depth	Pass
	test_hydrograph_part1.py::TestStationFactor	test_identity_no_antecedent	Pass
	test_hydrograph_part1.py::TestStationFactor	test_known_value	Pass
	test_hydrograph_part1.py::TestStationFactor	test_monotonic	Pass
	test_hydrograph_part1.py::TestStationFactor	test_negative_precip	Pass
	test_hydrograph_part1.py::TestSumPulses	test_composePulses	Pass
	test_hydrograph_part1.py::TestSumPulses	test_empty_pulses	Pass
	test_hydrograph_part1.py::TestSumPulses	test_non_overlapping_sequential	Pass
	test_hydrograph_part1.py::TestSumPulses	test_overlapping_exceeds_single	Pass
	test_hydrograph_part1.py::TestSumPulses	test_single_pulse	Pass
	test_hydrograph_part2.py::TestBuildCumulativeHydrograph	test_cumulative_monotonic	Pass
	test_hydrograph_part2.py::TestBuildCumulativeHydrograph	test_single_worst_pulse	Pass
	test_hydrograph_part2.py::TestBuildIsolatedPropertyHydrograph	test_output_shape	Pass
	test_hydrograph_part2.py::TestBuildIsolatedPropertyHydrograph	test_positive_flow	Pass
	test_hydrograph_part2.py::TestBuildIsolatedPropertyHydrograph	test_no_flood_when_below_threshold	Pass
	test_hydrograph_part2.py::TestBuildIsolatedPropertyHydrograph	test_output_shape	Pass
	test_builder_coverage.py::TestBuildExceedanceCurve	test_probs_capped_at_001	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_builder_coverage.py::TestBuilderLog	test_verbose_false_no_log	Pass
	test_builder_coverage.py::TestBuilderLog	test_verbose_true_emits_log	Pass
	test_builder_coverage.py::TestBuilderProgressLog	test_progress_log_every_10_gauges	Pass
	test_distribution_coverage.py::TestInverseNormalCdf	test_one_returns_pos_inf	Pass
	test_distribution_coverage.py::TestInverseNormalCdf	test_one_returns_neg_inf	Pass
	test_distribution_coverage.py::TestInverseNormalCdf	test_zero_returns_pos_inf	Pass
	test_distribution_coverage.py::TestInverseNormalCdf	test_zero_returns_neg_inf	Pass
	test_distribution_coverage.py::TestInverseNormalCdf	test_high_p_branch	Pass
	test_distribution_coverage.py::TestInverseNormalCdf	test_high_p_branch_0_98	Pass
	test_distribution_coverage.py::TestInverseNormalCdf	test_high_p_branch_positive	Pass
	test_distribution_coverage.py::TestInverseNormalCdf	test_low_p_branch_high_p	Pass
	test_distribution_coverage.py::TestNormalPdf	test_normal_pdf_at_zero	Pass
	test_distribution_coverage.py::TestNormalPdf	test_normal_pdf_decreases_from_center	Pass
	test_distribution_coverage.py::TestNormalPdf	test_normal_pdf_positive	Pass
	test_distribution_coverage.py::TestNormalPdf	test_normal_pdf_symmetric	Pass
	test_ea_flood_zone_rates.py::TestEAFloodZoneRates	test_all_rates_below_one	Pass
	test_ea_flood_zone_rates.py::TestEAFloodZoneRates	test_all_rates_positive	Pass
	test_ea_flood_zone_rates.py::TestEAFloodZoneRates	test_has_all_five_zones	Pass
	test_ea_flood_zone_rates.py::TestEAFloodZoneRates	test_zone_rate_float[Zone 1]	Pass
	test_ea_flood_zone_rates.py::TestEAFloodZoneRates	test_zone_rate_float[Zone 2]	Pass
	test_ea_flood_zone_rates.py::TestEAFloodZoneRates	test_zone_rate_float[Zone 3]	Pass
	test_ea_flood_zone_rates.py::TestEAFloodZoneRates	test_zone_rate_float[Zone 3a]	Pass
	test_ea_flood_zone_rates.py::TestEAFloodZoneRates	test_zone_rate_float[Zone 3b]	Pass
	test_ea_flood_zone_rates.py::TestEAFloodZoneRates	test_zone3_equals_3a	Pass
	test_ea_flood_zone_rates.py::TestEAFloodZoneRates	test_zone_ordering_2_gt_1	Pass
	test_ea_flood_zone_rates.py::TestEAFloodZoneRates	test_zone_ordering_3a_gt_2	Pass
	test_ea_flood_zone_rates.py::TestEAFloodZoneRates	test_zone_ordering_3b_gt_3a	Pass
	test_io_coverage.py::TestBuildHazardGrid	test_missing_single_file_raises	Pass
	test_io_coverage.py::TestBuildHazardGrid	test_missing_multiple_sequences_raises	Pass
	test_io_coverage.py::TestBuildHazardGrid	test_verbose_false_no_log	Pass
	test_io_coverage.py::TestBuildHazardGrid	test_verbose_true_logs_messages	Pass
	test_prs_analytical_coverage.py::TestComputeRiskSpreadEdges	test_empty_zero_spread_fallback	Pass
	test_prs_analytical_coverage.py::TestComputeRiskSpreadEdges	test_negative_spread_returns_zero	Pass
	test_prs_analytical_coverage.py::TestComputeRiskSpreadEdges	test_with_risk_curve	Pass
	test_prs_analytical_coverage.py::TestComputeRiskSpreadEdges	test_zero_hazard_spread_zero	Pass
	test_prs_analytical_coverage.py::TestInterpolateYieldRateEdges	test_empty_direct_yield_rate_fallback	Pass
	test_prs_analytical_coverage.py::TestInterpolateYieldRateEdges	test_empty_direct_returns_fallback	Pass
	test_prs_analytical_coverage.py::TestInterpolateYieldRateEdges	test_interpolate_through_two_points	Pass
	test_prs_analytical_coverage.py::TestInterpolateYieldRateEdges	test_interpolate_returns_last	Pass
	test_prs_analytical_coverage.py::TestInterpolateYieldRateEdges	test_interpolate_returns_last	Pass
	test_terrain_grid.py::TestGridMotion	test_spread_decreases_with_distance	Pass
	test_terrain_grid.py::TestGridMotion	test_spread_decreases_with_elevation	Pass
	test_terrain_grid.py::TestGridMotion	test_zone2_gt_zone1_everywhere	Pass
	test_terrain_grid.py::TestGridMotion	test_zone3a_gt_zone2_everywhere	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_terrain_grid.py::TestGridMonotonicity	test_zone3b_gt_zone3a_everywhere	Pass
	test_terrain_grid.py::TestGridSpotChecks	test_zone1_origin	Pass
	test_terrain_grid.py::TestGridSpotChecks	test_zone2_mid_grid	Pass
	test_terrain_grid.py::TestGridSpotChecks	test_zone3a_far_high	Pass
	test_terrain_grid.py::TestGridSpotChecks	test_zone3b_origin	Pass
	test_terrain_grid.py::TestGridStructure	test_distance_count	Pass
	test_terrain_grid.py::TestGridStructure	test_elevation_count	Pass
	test_terrain_grid.py::TestGridStructure	test_grid_dimensions_per_zone	Pass
	test_terrain_grid.py::TestGridStructure	test_has_all_zones	Pass
	test_terrain_grid.py::TestGridStructure	test_has_distances	Pass
	test_terrain_grid.py::TestGridStructure	test_has_elevations	Pass
	test_terrain_grid.py::TestGridStructure	test_params_present	Pass
	test_terrain_grid.py::TestGridStructure	test_zone_rates_match_config	Pass
	test_terrain_grid.py::TestInterpolation	test_at_grid_point_exact	Pass
	test_terrain_grid.py::TestInterpolation	test_at_origin	Pass
	test_terrain_grid.py::TestInterpolation	test_clamp_beyond_max_distance	Pass
	test_terrain_grid.py::TestInterpolation	test_clamp_beyond_max_elevation	Pass
	test_terrain_grid.py::TestInterpolation	test_clamp_negative_distance	Pass
	test_terrain_grid.py::TestInterpolation	test_clamp_negative_elevation	Pass
	test_terrain_grid.py::TestInterpolation	test_delta_higher_zone_positive	Pass
	test_terrain_grid.py::TestInterpolation	test_delta_same_zone_is_zero	Pass
	test_terrain_grid.py::TestInterpolation	test_interpolation_matches_analytical_at_grid_corners[Zone 1]	Pass
	test_terrain_grid.py::TestInterpolation	test_interpolation_matches_analytical_at_grid_corners[Zone 2]	Pass
	test_terrain_grid.py::TestInterpolation	test_interpolation_matches_analytical_at_grid_corners[Zone 3]	Pass
	test_terrain_grid.py::TestInterpolation	test_interpolation_matches_analytical_at_grid_corners[Zone 3a]	Pass
	test_terrain_grid.py::TestInterpolation	test_interpolation_matches_analytical_at_grid_corners[Zone 3b]	Pass
	test_terrain_grid.py::TestInterpolation	test_midpoint_between_grid_values	Pass
	cli.py::TestCompareScenarios	test_contains_header	Pass
	cli.py::TestCompareScenarios	test_contains_scenario_names	Pass
	cli.py::TestCompareScenarios	test_none_uses_all_scenarios	Pass
	cli.py::TestCompareScenarios	test_returns_string	Pass
	cli.py::TestCompareScenarios	test_single_scenario	Pass
	cli.py::TestCompareScenarios	test_two_scenarios	Pass
	cli.py::TestMain	test_compare_mode	Pass
	cli.py::TestMain	test_default_describe_mode	Pass
	cli.py::TestMain	test_exceedance_json_mode	Pass
	cli.py::TestMain	test_exceedance_mode	Pass
	cli.py::TestMain	test_json_default_mode	Pass
	cli.py::TestMain	test_return_period_json_mode	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	cli.py::TestMain	test_return_period_mode	Pass
	cli.py::TestMain	test_samples_json_mode	Pass
	cli.py::TestMain	test_samples_mode	Pass
	cli.py::TestMain	test_scenario_mode	Pass
	cli.py::TestMain	test_seed_reproducible	Pass
	distribution_part1.py::TestConstruction	test_custom_parameters	Pass
	distribution_part1.py::TestConstruction	test_default_parameters	Pass
	distribution_part1.py::TestConstruction	test_from_params_object	Pass
	distribution_part1.py::TestConstruction	test_from_scenario_baseline	Pass
	distribution_part1.py::TestConstruction	test_from_scenario_unknown_raises	Pass
	distribution_part1.py::TestConstruction	test_seed_gives_deterministic_output	Pass
	distribution_part1.py::TestConstruction	test_tail_prob_between_0_and_1	Pass
	distribution_part1.py::TestExceedanceProbability	test_above_max_is_zero	Pass
	distribution_part1.py::TestExceedanceProbability	test_above_threshold_uses_pareto	Pass
	distribution_part1.py::TestExceedanceProbability	test_at_max_intensity_is_zero	Pass
	distribution_part1.py::TestExceedanceProbability	test_at_min_intensity_is_one	Pass
	distribution_part1.py::TestExceedanceProbability	test_at_threshold_continuous	Pass
	distribution_part1.py::TestExceedanceProbability	test_below_min_is_one	Pass
	distribution_part1.py::TestExceedanceProbability	test_below_threshold_uses_normal	Pass
	distribution_part1.py::TestExceedanceProbability	test_monotonically_decreasing	Pass
	distribution_part1.py::TestInverseExceedanceProb	test_one_prob_returns_min	Pass
	distribution_part1.py::TestInverseExceedanceProb	test_result_in_bounds	Pass
	distribution_part1.py::TestInverseExceedanceProb	test_roundtrip_above_threshold	Pass
	distribution_part1.py::TestInverseExceedanceProb	test_roundtrip_below_threshold	Pass
	distribution_part1.py::TestInverseExceedanceProb	test_zero_prob_returns_max	Pass
	distribution_part1.py::TestParetoUnstable	test_fat_tail_higher_exceedance	Pass
	distribution_part1.py::TestParetoUnstable	test_pareto_above_threshold_between_0_and_1	Pass
	distribution_part1.py::TestParetoUnstable	test_pareto_at_threshold_is_one	Pass
	distribution_part1.py::TestParetoUnstable	test_pareto_below_threshold_is_one	Pass
	distribution_part1.py::TestSampling	test_different_seeds_different_output	Pass
	distribution_part1.py::TestSampling	test_sample_n_returns_correct_length	Pass
	distribution_part1.py::TestSampling	test_sample_single	Pass
	distribution_part1.py::TestSampling	test_samples_within_bounds	Pass
	distribution_part1.py::TestSampling	test_seeded_reproducibility	Pass
	distribution_part2.py::TestReturnPeriod	test_100yr_greater_than_10yr	Pass
	distribution_part2.py::TestReturnPeriod	test_ordering_10_50_100	Pass
	distribution_part2.py::TestReturnPeriod	test_result_in_bounds	Pass
	distribution_part2.py::TestReturnPeriod	test_storms_per_year_affects_result	Pass
	distribution_part2.py::TestSerialisation	test_describe_returns_string	Pass
	distribution_part2.py::TestSerialisation	test_params_roundtrip	Pass
	distribution_part2.py::TestSerialisation	test_tail_probability_matches_internal	Pass
	distribution_part2.py::TestSerialisation	test_to_dict_keys	Pass
	distribution_part2.py::TestSummaryStatistics	test_exceedance_probs_decreasing	Pass
	distribution_part2.py::TestSummaryStatistics	test_fat_tail_higher_p99_than_thin	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	distribution_part2.py::TestSummaryStatistics	test_mean_in_reasonable_range	Pass
	distribution_part2.py::TestSummaryStatistics	test_percentiles_ordered	Pass
	distribution_part2.py::TestSummaryStatistics	test_required_keys_present	Pass
	prs_hazard_curve.py::TestCreateFlatHazardCurve	test_day_counter	Pass
	prs_hazard_curve.py::TestCreateFlatHazardCurve	test_returns_curve_and_quote	Pass
	prs_hazard_curve.py::TestCreateSurvivalCurve	test_custom_day_counter	Pass
	prs_hazard_curve.py::TestCreateSurvivalCurve	test_default_day_counter	Pass
	prs_hazard_curve.py::TestCreateSurvivalCurve	test_full_survival_at_start	Pass
	prs_hazard_curve.py::TestCreateSurvivalCurve	test_survival_curve_from_hazard	Pass
	prs_hazard_curve.py::TestCreateSurvivalCurve	test_survival_equal_1_uses_fallback_hazard_rate	Pass
	prs_hazard_curve.py::TestCreateSurvivalCurve	test_survival_equal_0_uses_fallback_hazard_rate	Pass
	prs_hazard_curve.py::TestLoadHazardCurves	test_loaded_data_has_gauge	Pass
	prs_hazard_curve.py::TestLoadHazardCurves	test_loads_from_file	Pass
	prs_hazard_curve.py::TestPricePrs	test_alert_trigger	Pass
	prs_hazard_curve.py::TestPricePrs	test_custom_notional	Pass
	prs_hazard_curve.py::TestPricePrs	test_flat_hazard_curve	Pass
	prs_hazard_curve.py::TestPricePrs	test_result_has_fair_spread_bps	Pass
	prs_hazard_curve.py::TestPricePrs	test_result_has_npv	Pass
	prs_hazard_curve.py::TestPricePrs	test_result_has_survival_probabilities	Pass
	prs_hazard_curve.py::TestPricePrs	test_returns_dict	Pass
	prs_hazard_curve.py::TestPricePrs	test_severe_trigger	Pass
	prs_hazard_curve.py::TestPricePrs	test_term_structure_true	Pass
	prs_hazard_curve.py::TestPrintPrsLogs	test_logs_gauge_name	Pass
	prs_hazard_curve.py::TestPrintPrsLogs	test_logs_without_error	Pass
	test_load_and_curves.py::TestCreateFlatHazardCurve	test_actual360_day_counter_accepted	Pass
	test_load_and_curves.py::TestCreateFlatHazardCurve	test_quote_is_simple_decimal	Pass
	test_load_and_curves.py::TestCreateFlatHazardCurve	test_survival_curve_matches_input	Pass
	test_load_and_curves.py::TestCreateFlatHazardCurve	test_survival_probability_formula	Pass
	test_load_and_curves.py::TestCreateFlatHazardCurve	test_very_high_hazard_rate_near_zero_survival	Pass
	test_load_and_curves.py::TestCreateSurvivalCurve	test_actual360_day_counter_produces_curve	Pass
	test_load_and_curves.py::TestCreateSurvivalCurve	test_simple_hazard_rate_formula	Pass
	test_load_and_curves.py::TestCreateSurvivalCurve	test_near_zero_survival_positive_hazard	Pass
	test_load_and_curves.py::TestCreateSurvivalCurve	test_single_point_high_risk_ts	Pass
	test_load_and_curves.py::TestCreateSurvivalCurve	test_single_point_term_structure	Pass
	test_load_and_curves.py::TestCreateSurvivalCurve	test_survival_curve_strictly_positive	Pass
	test_load_and_curves.py::TestLoadHazardCurves	test_empty_curves_object_returns_empty_dict	Pass
	test_load_and_curves.py::TestLoadHazardCurves	test_invalid_json_raises_json_decode_error	Pass
	test_load_and_curves.py::TestLoadHazardCurves	test_missing_file_raises_file_not_found	Pass
	test_load_and_curves.py::TestLoadHazardCurves	test_term_structure_flag	Pass
	test_price_prs.py::TestPricePrsGauge	test_annual_hazard_rate_matches_input	Pass
	test_price_prs.py::TestPricePrsGauge	test_gauge_origin_in_result	Pass
	test_price_prs.py::TestPricePrsGauge	test_gauge_name_in_result	Pass
	test_price_prs.py::TestPricePrsGauge	test_seven_annual_hazard_rate_matches_input	Pass
	test_price_prs.py::TestPricePrsLegs	test_npv_is_sum_of_legs	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_price_prs.py::TestPricePrsLegSign	test_premium_leg_negative	Pass
	test_price_prs.py::TestPricePrsLegSign	test_protection_leg_positive	Pass
	test_price_prs.py::TestPricePrsRecovery	test_nonzero_recovery_rate_stored	Pass
	test_price_prs.py::TestPricePrsRecovery	test_recovery_rate_affects_fair_spread	Pass
	test_price_prs.py::TestPricePrsRiskFree	test_higher_risk_free_rate_changes_npv	Pass
	test_price_prs.py::TestPricePrsRiskFree	test_risk_free_rate_not_in_output	Pass
	test_price_prs.py::TestPricePrsTypes	test_fair_spread_is_float	Pass
	test_price_prs.py::TestPricePrsTypes	test_fair_upfront_is_float	Pass
	test_price_prs.py::TestPricePrsTypes	test_npv_is_float	Pass
	test_price_prs.py::TestPricePrsTypes	test_running_spread_bps_identity	Pass
	test_price_prs.py::TestPricePrsTypes	test_survival_probs_all_in_zero_one	Pass
	test_price_prs.py::TestPricePrsTypes	test_tenor_1yr_gives_one_survival_entry	Pass
	test_price_prs.py::TestPricePrsTypes	test_use_term_structure_stored_false	Pass
	test_price_prs.py::TestPricePrsTypes	test_use_term_structure_stored_true	Pass
	test_print_and_cli.py::TestMain	test_custom_notional_and_spread	Pass
	test_print_and_cli.py::TestMain	test_default_gauge_selection_returns_zero	Pass
	test_print_and_cli.py::TestMain	test_explicit_gauge_returns_zero	Pass
	test_print_and_cli.py::TestMain	test_flat_flag_dispatches_flat_mode	Pass
	test_print_and_cli.py::TestMain	test_list_flag_logs_gauge_names	Pass
	test_print_and_cli.py::TestMain	test_list_flag_returns_zero	Pass
	test_print_and_cli.py::TestMain	test_missing_gauge_logs_available_gauges	Pass
	test_print_and_cli.py::TestMain	test_missing_gauge_logs_error	Pass
	test_print_and_cli.py::TestMain	test_missing_gauge_returns_one	Pass
	test_print_and_cli.py::TestMain	test_trigger_alert_explicit	Pass
	test_print_and_cli.py::TestMain	test_trigger_severe_explicit	Pass
	test_print_and_cli.py::TestMain	test_trigger_warning_explicit	Pass
	test_print_and_cli.py::TestPrintPricing	test_alert_trigger_in_log	Pass
	test_print_and_cli.py::TestPrintPricing	test_survival_years_in_log	Pass
	test_print_and_cli.py::TestPrintPricing	test_contract_terms_header_in_log	Pass
	test_print_and_cli.py::TestPrintPricing	test_fair_spread_in_log	Pass
	test_print_and_cli.py::TestPrintPricing	test_gauge_id_in_log	Pass
	test_print_and_cli.py::TestPrintPricing	test_gauge_name_in_log	Pass
	test_print_and_cli.py::TestPrintPricing	test_npv_value_in_log	Pass
	test_print_and_cli.py::TestPrintPricing	test_premium_leg_in_log	Pass
	test_print_and_cli.py::TestPrintPricing	test_protection_leg_in_log	Pass
	test_print_and_cli.py::TestPrintPricing	test_recovery_rate_in_log	Pass
	test_print_and_cli.py::TestPrintPricing	test_severe_trigger_in_log	Pass
	test_print_and_cli.py::TestPrintPricing	test_survival_curve_label_in_log	Pass
	test_print_and_cli.py::TestPrintPricing	test_trigger_level_in_log	Pass
	test_risk_assessor_core.py::TestCombinedRisk	test_combined_50_in_100	Pass
	test_risk_assessor_core.py::TestCombinedRisk	test_combined_zero_in_100	Pass
	test_risk_assessor_core.py::TestCombinedRisk	test_combined_returns_float	Pass
	test_risk_assessor_core.py::TestCombinedRisk	test_combined_risk_in_concrete_lower	Pass
	test_risk_assessor_core.py::TestCombinedRisk	test_combined_risk_in_steel_lowest	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_risk_assessor_core.py::TestCombinedRiskScore	test_combined_construction_timber_higher	Pass
	test_risk_assessor_core.py::TestCombinedRiskScore	test_combined_construction_wood_equals_timber	Pass
	test_risk_assessor_core.py::TestCombinedRiskScore	test_combined_high_1_multiplier	Pass
	test_risk_assessor_core.py::TestCombinedRiskScore	test_combined_high_60_to_80_multiplier	Pass
	test_risk_assessor_core.py::TestCombinedRiskScore	test_combined_high_80_to_95_multiplier	Pass
	test_risk_assessor_core.py::TestCombinedRiskScore	test_combined_medium_1_multiplier	Pass
	test_risk_assessor_core.py::TestCombinedRiskScore	test_combined_unknown_construction_type	Pass
	test_risk_assessor_core.py::TestCombinedRiskScore	test_combined_unknown_flood_level	Pass
	test_risk_assessor_core.py::TestCombinedRiskScore	test_combined_very_high_flood_capped	Pass
	test_risk_assessor_core.py::TestCombinedRiskScore	test_combined_very_low_flood_lowest_score	Pass
	test_risk_assessor_core.py::TestFloodRiskLevel	test_risk_high_threshold	Pass
	test_risk_assessor_core.py::TestFloodRiskLevel	test_risk_low_threshold	Pass
	test_risk_assessor_core.py::TestFloodRiskLevel	test_risk_medium_threshold	Pass
	test_risk_assessor_core.py::TestFloodRiskLevel	test_risk_very_low_threshold	Pass
	test_risk_assessor_core.py::TestFloodRiskLevel	test_risk_high_range	Pass
	test_risk_assessor_core.py::TestFloodRiskLevel	test_risk_medium_range	Pass
	test_risk_assessor_core.py::TestFloodRiskLevel	test_risk_level_one_with_depth	Pass
	test_risk_assessor_core.py::TestFloodRiskLevel	test_risk_negative_returns_unknown	Pass
	test_risk_assessor_core.py::TestFloodRiskLevel	test_risk_none_returns_unknown	Pass
	test_risk_assessor_core.py::TestFloodRiskLevel	test_risk_very_high	Pass
	test_risk_assessor_core.py::TestFloodRiskLevel	test_risk_zero_is_very_low	Pass
	test_risk_assessor_core.py::TestLTVRiskLevel	test_ltv_high_boundary	Pass
	test_risk_assessor_core.py::TestLTVRiskLevel	test_ltv_low_boundary	Pass
	test_risk_assessor_core.py::TestLTVRiskLevel	test_ltv_moderate_boundary	Pass
	test_risk_assessor_core.py::TestLTVRiskLevel	test_critical_ltv	Pass
	test_risk_assessor_core.py::TestLTVRiskLevel	test_high_ltv	Pass
	test_risk_assessor_core.py::TestLTVRiskLevel	test_low_ltv	Pass
	test_risk_assessor_core.py::TestLTVRiskLevel	test_moderate_ltv	Pass
	test_risk_assessor_core.py::TestLTVRiskLevel	test_none_returns_unknown	Pass
	test_risk_assessor_core.py::TestLTVRiskLevel	test_normalises_95_pct	Pass
	test_risk_assessor_core.py::TestLTVRiskLevel	test_normalises_percentage_input	Pass
	test_risk_assessor_core.py::TestMortgageRisk	test_mortgage_risk_ltv_medium_flood	Pass
	test_risk_assessor_core.py::TestMortgageRisk	test_mortgage_risk_high_flood_overrides	Pass
	test_risk_assessor_core.py::TestMortgageRisk	test_mortgage_risk_high_ltv_medium_flood	Pass
	test_risk_assessor_core.py::TestMortgageRisk	test_mortgage_risk_low_flood_risk	Pass
	test_risk_assessor_core.py::TestMortgageRisk	test_mortgage_risk_medium_flood_default	Pass
	test_risk_assessor_core.py::TestMortgageRisk	test_mortgage_risk_minimal_risk	Pass
	test_risk_assessor_core.py::TestMortgageRisk	test_mortgage_risk_negative_value_critical	Pass
	test_risk_assessor_core.py::TestMortgageRisk	test_mortgage_risk_negative_value_high	Pass
	test_risk_assessor_core.py::TestMortgageRisk	test_mortgage_risk_negative_value_moderate	Pass
	test_risk_assessor_core.py::TestMortgageRisk	test_mortgage_risk_ltv_no_crash	Pass
	test_risk_assessor_core.py::TestMortgageRisk	test_mortgage_risk_very_high_flood_overrides	Pass
	test_risk_assessor_core.py::TestPropertyVulnerability	test_vulnerability_water_increases_score	Pass
	test_risk_assessor_core.py::TestPropertyVulnerability	test_flood_depth_computed	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_risk_assessor_core.py::TestPropertyVulnerability	test_high_risk_recommendations_present	Pass
	test_risk_assessor_core.py::TestPropertyVulnerability	test_no_flood	Pass
	test_risk_assessor_core.py::TestPropertyVulnerability	test_none_elevation_returns_unknown	Pass
	test_risk_assessor_core.py::TestPropertyVulnerability	test_none_floodlevel_returns_unknown	Pass
	test_risk_assessor_core.py::TestPropertyVulnerability	test_result_dict_keys	Pass
	test_risk_assessor_core.py::TestPropertyVulnerability	test_score_capped_at_100	Pass
	test_risk_assessor_factors.py::TestDistanceRiskFactor	test_distance_zero_is_max	Pass
	test_risk_assessor_factors.py::TestDistanceRiskFactor	test_close_is_high	Pass
	test_risk_assessor_factors.py::TestDistanceRiskFactor	test_far_is_low	Pass
	test_risk_assessor_factors.py::TestDistanceRiskFactor	test_medium_is_medium	Pass
	test_risk_assessor_factors.py::TestDistanceRiskFactor	test_monotonic_decrease	Pass
	test_risk_assessor_factors.py::TestDistanceRiskFactor	test_near_is_high	Pass
	test_risk_assessor_factors.py::TestDistanceRiskFactor	test_negative_returns_default	Pass
	test_risk_assessor_factors.py::TestDistanceRiskFactor	test_none_returns_default	Pass
	test_risk_assessor_factors.py::TestDistanceRiskFactor	test_very_close	Pass
	test_risk_assessor_factors.py::TestDistanceRiskFactor	test_very_far	Pass
	test_risk_assessor_factors.py::TestCategoryReliability	test_daily_lt_15	Pass
	test_risk_assessor_factors.py::TestCategoryReliability	test_decommissioned	Pass
	test_risk_assessor_factors.py::TestCategoryReliability	test_fully_operational	Pass
	test_risk_assessor_factors.py::TestCategoryReliability	test_monthly_lt_5	Pass
	test_risk_assessor_factors.py::TestCategoryReliability	test_maintenance_required	Pass
	test_risk_assessor_factors.py::TestCategoryReliability	test_monthly_lt_50	Pass
	test_risk_assessor_factors.pyTestCategoryReliability	test_real_time_no_adjustment	Pass
	test_risk_assessor_factors.pyTestCategoryReliability	test_result_keys	Pass
	test_risk_assessor_factors.pyTestCategoryReliability	test_score_clamped_at_zero	Pass
	test_risk_assessor_factors.pyTestCategoryReliability	test_temporarily_offline	Pass
	test_risk_assessor_factors.pyTestCategoryReliability	test_unknown_status	Pass
	test_risk_assessor_factors.pyTestCategoryReliability	test_weekly_lt_30	Pass
	test_risk_assessor_factors.pyTestGenerateRecommendations	test_all_items_are_strings	Pass
	test_risk_assessor_factors.pyTestGenerateRecommendations	test_date_flood_extra_items	Pass
	test_risk_assessor_factors.pyTestGenerateRecommendations	test_high_risk_foundation_more	Pass
	test_risk_assessor_factors.pyTestGenerateRecommendations	test_high_risk_mentions_flood_insurance	Pass
	test_risk_assessor_factors.pyTestGenerateRecommendations	test_low_risk	Pass
	test_risk_assessor_factors.pyTestGenerateRecommendations	test_medium_risk	Pass
	test_risk_assessor_factors.pyTestGenerateRecommendations	test_very_low_risk	Pass
	test_risk_assessor_factors.pyTestInsuranceFactor	test_base_premium_level	Pass
	test_risk_assessor_factors.pyTestInsuranceFactor	test_capped_at_10_pct	Pass
	test_risk_assessor_factors.pyTestInsuranceFactor	test_depth_increases_premium	Pass
	test_risk_assessor_factors.pyTestInsuranceFactor	test_depth_multiplier_formula	Pass
	test_risk_assessor_factors.pyTestInsuranceFactor	test_unknown_level_uses_default	Pass
	test_risk_assessor_factors.pyTestInsuranceFactor	test_zero_depth_no_effect	Pass
	test_risk_assessor_factors.pyTestValueAtRiskFactor	test_value_at_risk_wins_when_higher_than_depth	Pass
	test_risk_assessor_factors.pyTestValueAtRiskFactor	test_value_1_to_2m	Pass
	test_risk_assessor_factors.pyTestValueAtRiskFactor	test_value_half_to_1m	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_risk_assessor_factors.py::TestValueAtRisk	test_depth_over_2m	Pass
	test_risk_assessor_factors.py::TestValueAtRisk	test_depth_under_half_m	Pass
	test_risk_assessor_factors.py::TestValueAtRisk	test_negative_property_value	Pass
	test_risk_assessor_factors.pyTestValueAtRisk	test_risk_level_percentages	Pass
	test_risk_assessor_factors.pyTestValueAtRisk	test_zero_depth_uses_base_pct	Pass
	test_risk_assessor_factors.pyTestValueAtRisk	test_zero_property_value	Pass
	statistics.py::TestCalculateFlowStatistics	test_has_mean_flow_not_flow_cumecs	Pass
	statistics.py::TestCalculateFlowStatistics	test_has_q_flows	Pass
	statistics.py::TestCalculateFlowStatistics	test_returns_dict	Pass
	statistics.py::TestCalculateLevelStatistics	test_has_mean_level_not_mean_level_meters	Pass
	statistics.py::TestCalculateLevelStatistics	test_has_min_max	Pass
	statistics.py::TestCalculateLevelStatistics	test_returns_dict	Pass
	statistics.py::TestCalculateTimeseriesStatistics	test_all_same_values	Pass
	statistics.py::TestCalculateTimeseriesStatistics	test_empty_returns_empty_dict	Pass
	statistics.py::TestCalculateTimeseriesStatistics	test_flood_exceedances_zero_when_all_below	Pass
	statistics.py::TestCalculateTimeseriesStatistics	test_flood_stages_adds_exceedances	Pass
	statistics.py::TestCalculateTimeseriesStatistics	test_flow_cumecs_adds_q_flows	Pass
	statistics.py::TestCalculateTimeseriesStatistics	test_has_annual_maxima	Pass
	statistics.py::TestCalculateTimeseriesStatistics	test_has_basic_keys	Pass
	statistics.py::TestCalculateTimeseriesStatistics	test_has_extreme_events	Pass
	statistics.py::TestCalculateTimeseriesStatistics	test_has_monthly_means	Pass
	statistics.py::TestCalculateTimeseriesStatistics	test_has_percentiles	Pass
	statistics.py::TestCalculateTimeseriesStatistics	test_max_date_correct	Pass
	statistics.py::TestCalculateTimeseriesStatistics	test_no_flood_stages_no_exceedances_key	Pass
	statistics.py::TestCalculateTimeseriesStatistics	test_record_start_end	Pass
	statistics.py::TestCalculateTimeseriesStatistics	test_returns_dict	Pass
	statistics.py::TestCalculateTimeseriesStatistics	test_single_observation_no_stdev_error	Pass
	statistics.py::TestCalculateTimeseriesStatistics	test_total_days_correct	Pass
	data_structures.py::TestDecayKernel	test_exponential_value	Pass
	data_structures.py::TestDecayKernel	test_gaussian_value	Pass
	data_structures.py::TestDecayKernel	test_linear_value	Pass
	data_structures.py::TestGaugeConfigFromPortfolio	test_has_level_calculated	Pass
	data_structures.py::TestGaugeConfigFromPortfolio	test_creates_gauge_config	Pass
	data_structures.py::TestGaugeConfigFromPortfolio	test_defaults_when_no_flood_stages	Pass
	data_structures.py::TestGaugeConfigFromPortfolio	test_flood_stages_data	Pass
	data_structures.py::TestGaugeConfigFromPortfolio	test_flood_warn_set	Pass
	data_structures.py::TestGaugeConfigFromPortfolio	test_gauge_id_set	Pass
	data_structures.py::TestGaugeConfigFromPortfolio	test_historical_high_set	Pass
	data_structures.py::TestGaugeResponseToDict	test_excludes_timeseries_when_requested	Pass
	data_structures.py::TestGaugeResponseToDict	test_flooded_is_bool	Pass
	data_structures.py::TestGaugeResponseToDict	test_has_required_keys	Pass
	data_structures.py::TestGaugeResponseToDict	test_includes_timeseries_by_default	Pass
	data_structures.py::TestGaugeResponseToDict	test_returns_dict	Pass
	data_structures.py::TestGaugeResponseToDict	test_values_rounded	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	data_structures.py::TestIntensityProfile	test_beta_value	Pass
	data_structures.py::TestIntensityProfile	test_gaussian_value	Pass
	data_structures.py::TestIntensityProfile	test_triangular_value	Pass
	data_structures.py::TestStormTotalDist	test_custom_kernel	Pass
	data_structures.py::TestStormTotalDist	test_decay_kernel_serialised_as_string	Pass
	data_structures.py::TestStormTotalDist	test_has_storm_id	Pass
	data_structures.py::TestStormTotalDist	test_has_track	Pass
	data_structures.py::TestStormTotalDist	test_intensity_profile_serialised_as_string	Pass
	data_structures.py::TestStormTotalDist	test_returns_dict	Pass
	data_structures.py::TestStormTotalDist	test_track_has_coords	Pass
	forward_model_part1.py::TestComputeDecay	test_all_kernels_decrease_with_distance	Pass
	forward_model_part1.py::TestComputeDecay	test_all_kernels_in_0_1_range	Pass
	forward_model_part1.py::TestComputeDecay	test_exponential_at_half	Pass
	forward_model_part1.py::TestComputeDecay	test_exponential_at_one_lambda	Pass
	forward_model_part1.py::TestComputeDecay	test_gaussian_at_one_sigma	Pass
	forward_model_part1.py::TestComputeDecay	test_gaussian_at_two_sigma	Pass
	forward_model_part1.py::TestComputeDecay	test_linear_beyond_cutoff_is_zero	Pass
	forward_model_part1.py::TestComputeDecay	test_linear_midpoint	Pass
	forward_model_part1.py::TestComputeDecay	test_zero_distance_returns_one_all_kernels	Pass
	forward_model_part1.py::TestFindNearestTrackPoint	test_distance_is_positive	Pass
	forward_model_part1.py::TestFindNearestTrackPoint	test_returns_nearest_of_three	Pass
	forward_model_part1.py::TestFindNearestTrackPoint	test_single_track_point	Pass
	forward_model_part1.py::TestHaversine	test_equatorial_degree_approx_111km	Pass
	forward_model_part1.py::TestHaversine	test_london_to_paris_approx_340km	Pass
	forward_model_part1.py::TestHaversine	test_same_point_is_zero	Pass
	forward_model_part1.py::TestHaversine	test_symmetric	Pass
	forward_model_part2.py::TestComputeAllResponses	test_gauge_ids_match	Pass
	forward_model_part2.py::TestComputeAllResponses	test_near_gauge_higher_response	Pass
	forward_model_part2.py::TestComputeAllResponses	test_one_response_per_gauge	Pass
	forward_model_part2.py::TestComputeIntensityAtGauge	test_after_last_point	Pass
	forward_model_part2.py::TestComputeIntensityAtGauge	test_at_peak_time_is_nonzero	Pass
	forward_model_part2.py::TestComputeIntensityAtGauge	test_before_first_point	Pass
	forward_model_part2.py::TestComputeIntensityAtGauge	test_empty_track_returns_zero	Pass
	forward_model_part2.py::TestComputeIntensityAtGauge	test_far_from_track_is_lower	Pass
	forward_model_part2.py::TestComputeResponse	test_accumulation_nonnegative	Pass
	forward_model_part2.py::TestComputeResponse	test_duration_nonnegative	Pass
	forward_model_part2.py::TestComputeResponse	test_flooded_flag_correct	Pass
	forward_model_part2.py::TestComputeResponse	test_high_intensity_storm_floods	Pass
	forward_model_part2.py::TestComputeResponse	test_peak_level_at_least_base	Pass
	forward_model_part2.py::TestComputeResponse	test_peak_level_is_max_of_series	Pass
	forward_model_part2.py::TestComputeResponse	test_response_ids_match	Pass
	forward_model_part2.py::TestComputeResponse	test_timeseries_length	Pass
	forward_model_part2.py::TestComputeResponse	test_to_dict_roundtrip	Pass
	forward_model_part2.py::TestComputeResponse	test_to_dict_without_timeseries	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	forward_model_part2.py::TestIntensityToLevel	test_monotone_increasing	Pass
	forward_model_part2.py::TestIntensityToLevel	test_negative_intensity_returns_zero	Pass
	forward_model_part2.py::TestIntensityToLevel	test_positive_intensity_positive_level	Pass
	forward_model_part2.py::TestIntensityToLevel	test_sensitivity_scales_output	Pass
	forward_model_part2.py::TestIntensityToLevel	test_uses_historical_high_when_set	Pass
	forward_model_part2.py::TestIntensityToLevel	test_zero_intensity_returns_zero	Pass
	storm_factory.py::TestCreateStorm	test_auto_generated_id_starts_with_storm	Pass
	storm_factory.py::TestCreateStorm	test_decay_kernel_stored	Pass
	storm_factory.py::TestCreateStorm	test_decay_parameter_stored	Pass
	storm_factory.py::TestCreateStorm	test_default_num_track_points	Pass
	storm_factory.py::TestCreateStorm	test_default_start_time_is_recent	Pass
	storm_factory.py::TestCreateStorm	test_duration_hours_stored	Pass
	storm_factory.py::TestCreateStorm	test_explicit_id_preserved	Pass
	storm_factory.py::TestCreateStorm	test_explicit_name	Pass
	storm_factory.py::TestCreateStorm	test_explicit_start_time	Pass
	storm_factory.py::TestCreateStorm	test_footprint_km_stored	Pass
	storm_factory.py::TestCreateStorm	test_num_track_points	Pass
	storm_factory.py::TestCreateStorm	test_peak_intensity_stored	Pass
	storm_factory.py::TestCreateStorm	test_returns_storm_object	Pass
	storm_factory.py::TestCreateStorm	test_speed_is_positive	Pass
	storm_factory.py::TestCreateStorm	test_track_end_time_equals_duration	Pass
	storm_factory.py::TestCreateStorm	test_track_longitude_moves_from_start_to_end	Pass
	storm_factory.py::TestCreateStorm	test_track_points_are_trackpoint_instances	Pass
	storm_factory.py::TestCreateStorm	test_track_start_time_is_zero	Pass
	storm_factory.py::TestCreateStorm	test_track_time_increases	Pass
	storm_factory.py::TestIntensityProfiles	test_all_profiles_produce_valid_storm	Pass
	storm_factory.py::TestIntensityProfiles	test_beta_profile_creates_storm	Pass
	storm_factory.py::TestIntensityProfiles	test_gaussian_profile_peak_near_midpoint	Pass
	storm_factory.py::TestIntensityProfiles	test_triangular_max_equals_peak_intensity	Pass
	storm_factory.py::TestIntensityProfiles	test_triangular_profile_peak_near_midpoint	Pass
	storm_factory.py::TestLoadGaugesFromPortfolio	test_empty_portfolio_returns_empty_list	Pass
	storm_factory.py::TestLoadGaugesFromPortfolio	test_loads_gauge_from_portfolio	Pass
	flood_classifier.py::TestFloodClassifierTrainer	test_trainer_creates_joblib_file	Pass
	flood_classifier.py::TestFloodClassifierTrainer	test_trainer_has_metrics	Pass
	flood_classifier.py::TestFloodClassifierTrainer	test_trainer_returns_dict	Pass
	flood_classifier.py::TestFloodClassifierTrainer	test_trainer_single_class_skipped	Pass
	flood_classifier.py::TestFloodClassifierTrainer	test_trainer_status_trained	Pass
	flood_classifier.py::TestFloodPredictor	test_available_gauges	Pass
	flood_classifier.py::TestFloodPredictor	test_available_gauges_empty_dir	Pass
	flood_classifier.py::TestFloodPredictor	test_load_summary	Pass
	flood_classifier.py::TestFloodPredictor	test_load_summary_cached	Pass
	flood_classifier.py::TestFloodPredictor	test_predict_caches_model	Pass
	flood_classifier.py::TestFloodPredictor	test_predict_high_level_higher_probability	Pass
	flood_classifier.py::TestFloodPredictor	test_predict_in_range	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	flood_classifier.py::TestFloodPredictor	test_predict_missing_model_raises	Pass
	flood_classifier.py::TestFloodPredictor	test_predict_returns_float	Pass
	flood_classifier.py::TestFloodPredictor	test_predict_series_all_in_range	Pass
	flood_classifier.py::TestFloodPredictor	test_predict_series_default_derivatives	Pass
	flood_classifier.py::TestFloodPredictor	test_predict_series_returns_list	Pass
	flood_classifier.py::TestFloodPredictor	test_predict_series_with_derivatives	Pass
	flood_classifier.py::TestFloodPredictor	test_predict_surface_returns_grid	Pass
	test_flood_propagation_sensitivity.py::TestDistanceSensitivity	test_close_points_flood	Pass
	test_flood_propagation_sensitivity.py::TestDistanceSensitivity	test_distant_points_do_not_flood	Pass
	test_flood_propagation_sensitivity.py::TestDistanceSensitivity	test_flood_depth_decreases_with_distance	Pass
	test_flood_propagation_sensitivity.py::TestDistanceSensitivity	test_flood_values_at_key_distances	Pass
	test_flood_propagation_sensitivity.py::TestDistanceSensitivity	test_transition_is_realistic	Pass
	test_flood_propagation_sensitivity.py::TestElevationSensitivity	test_elevation_threshold_boundary	Pass
	test_flood_propagation_sensitivity.py::TestElevationSensitivity	test_flood_depth_decreases_with_elevation	Pass
	test_flood_propagation_sensitivity.py::TestElevationSensitivity	test_flood_level_raises_with_threshold	Pass
	test_flood_propagation_sensitivity.py::TestElevationSensitivity	test_high_elevation_does_not_flood	Pass
	test_flood_propagation_sensitivity.py::TestElevationSensitivity	test_low_elevation_floods	Pass
	test_flood_propagation_travel_time.py::TestEstimateAgreesWithHydrography	test_estimate_agrees_with_hydrography[100-6.0-0.3-8.0]	Pass
	test_flood_propagation_travel_time.py::TestEstimateAgreesWithHydrography	test_estimate_agrees_with_hydrography[1000-6.5-0.5-9.0]	Pass
	test_flood_propagation_travel_time.py::TestEstimateAgreesWithHydrography	test_estimate_agrees_with_hydrography[200-8.0-0.3-10.0]	Pass
	test_flood_propagation_travel_time.py::TestEstimateAgreesWithHydrography	test_estimate_agrees_with_hydrography[2000-6.0-0.3-7.0]	Pass
	test_flood_propagation_travel_time.py::TestEstimateAgreesWithHydrography	test_estimate_agrees_with_hydrography[3000-7.5-0.0-10.0]	Pass
	test_flood_propagation_travel_time.py::TestEstimateAgreesWithHydrography	test_estimate_agrees_with_hydrography[500-5.5-0.8-6.0]	Pass
	test_flood_propagation_travel_time.py::TestEstimateAgreesWithHydrography	test_estimate_agrees_with_hydrography[500-7.0-0.3-10.0]	Pass
	test_flood_propagation_travel_time.py::TestEstimateAgreesWithHydrography	test_estimate_agrees_with_hydrography[5000-6.0-0.0-12.0]	Pass
	test_flood_propagation_travel_time.py::TestTravelTimeSensitivity	test_arrival_time_increases_with_distance	Pass
	test_flood_propagation_travel_time.py::TestTravelTimeSensitivity	test_flood_duration_decreases_with_distance	Pass
	test_flood_propagation_travel_time.py::TestTravelTimeSensitivity	test_travel_time_decreases_with_depth	Pass
	test_flood_propagation_travel_time.py::TestTravelTimeSensitivity	test_travel_time_increases_with_distance	Pass
	test_hazard_pricing.py::TestCreateCustomPricingEngine	test_engine_is_none	Pass
	test_hazard_pricing.py::TestCreateCustomPricingEngine	test_returns_dict_with_engines	Pass
	test_hazard_pricing.py::TestCreateHazardous	test_hazard_raises_when_approved	Pass
	test_hazard_pricing.py::TestCreateHazardous	test_survival_probability_decreases	Pass
	test_hazard_pricing.py::TestCreateHazardous	test_zero_hazard_approx_full_survival	Pass
	test_hazard_pricing.py::TestHazardPointsToCurve	test_points_to_curve_count	Pass
	test_hazard_pricing.py::TestHazardPointsToCurve	test_high_hazard_gives_low_survival	Pass
	test_hazard_pricing.py::TestHazardPointsToCurve	test_points_to_curve_returns_curve	Pass
	test_hazard_pricing.py::TestHazardPointsToCurve	test_survival_probs_decreases	Pass
	test_hazard_pricing.py::TestHazardPointsToCurve	test_zero_hazard_gives_survival_near_one	Pass
	test_engine_coverage.py::TestPropertyBasedTrade	test_trade_has_basis_info	Pass
	test_engine_coverage.py::TestRemainingEdgeCases	test_empty_edge_case_returns_5	Pass
	test_engine_coverage.py::TestRemainingEdgeCases	test_invalid_edge_case_format_returns_5	Pass
	test_engine_coverage.py::TestRemainingEdgeCases	test_missing_edge_case_returns_5	Pass
	test_engine_coverage.py::TestRemainingEdgeCases	test_one_edge_case_returns_5	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_engine_coverage.py::TestRemainingTerm	test_end_date_returns_years	Pass
	test_engine_coverage.py::TestRevaluationDate	test_default_market_state	Pass
	test_pnl_coverage.py::TestGenerateEodSnapshot	test_snapshot_default_date	Pass
	test_pnl_coverage.py::TestGetEodSnapshot	test_existing_snapshot_returns_data	Pass
	test_pnl_coverage.py::TestGetEodSnapshot	test_nonexistent_snapshot_returns_none	Pass
	test_pnl_coverage.py::TestGetTradeMark	test_existing_trade_mark	Pass
	test_pnl_coverage.py::TestGetTradeMark	test_nonexistent_trade_mark	Pass
	test_pnl_coverage.py::TestGetTradeMark	test_trade_mark_after_multiple_updates	Pass
	test_market_state_coverage.py::TestCommitYieldCurve	test_creates_yield_curve_if_missing	Pass
	test_market_state_coverage.py::TestGetHazardRate	test_missing_term_falls_back_to_flat_rate	Pass
	test_market_state_coverage.py::TestGetHazardRate	test_no_hazard_rate_falls_back_to_zero	Pass
	test_market_state_coverage.py::TestGetYieldCurve	test_empty_curve_returns_risk_free_rate	Pass
	test_market_state_coverage.py::TestGetYieldCurve	test_unknown_curve_key_returns_risk_free_rate	Pass
	test_market_state_coverage.py::TestInterpolateYield	test_interpolation_falls_back	Pass
	test_market_state_coverage.py::TestUpdateYieldCurve	test_creates_yield_curve_if_missing	Pass
	test_market_state_reconciliation.py::TestLoadReconciled	test_gauge_reconciled_on_load	Pass
	test_market_state_reconciliation.py::TestLoadReconciled	test_no_reconciliation_when_no_new_gauges	Pass
	test_market_state_reconciliation.py::TestLoadReconciled	test_reconciled_gauge_has_term_structure	Pass
	test_market_state_reconciliation.py::TestLoadReconciled	test_reconciliation_persisted_to_disk	Pass
	property_value.py::TestApplyValueFactors	test_all_factors_applied	Pass
	property_value.py::TestApplyValueFactors	test_base_calculation	Pass
	property_value.py::TestApplyValueFactors	test_clamped_to_max	Pass
	property_value.py::TestApplyValueFactors	test_clamped_to_min	Pass
	property_value.py::TestApplyValueFactors	test_high_risk_discount_reduces_value	Pass
	property_value.py::TestApplyValueFactors	test_no_base_factor_applied	Pass
	property_value.py::TestApplyValueFactors	test_returns_float	Pass
	property_value.py::TestFactorTables	test_age_band_factors_within_reasonable_bounds	Pass
	property_value.py::TestFactorTables	test_base_area_ranges_all_positive	Pass
	property_value.py::TestFactorTables	test_base_price_per_sqm_all_positive	Pass
	property_value.py::TestFactorTables	test_condition_factors_ordered_by_severity	Pass
	property_value.py::TestFactorTables	test_epc_factors_a_highest	Pass
	property_value.py::TestFactorTables	test_flood_risk_factors_decrease_with_risk	Pass
	property_value.py::TestFactorTables	test_min_max_value_bounds	Pass
	property_value.py::TestFactorTables	test_proximity_zones_have_range_key	Pass
	property_value.py::TestFactorTables	test_rent_per_sqm_all_present	Pass
	property_value.py::TestFactorTables	test_rental_yield_rates_all_present	Pass
	property_value.py::TestGetAgeBand	test_100_years_is_heritage	Pass
	property_value.py::TestGetAgeBand	test_10_years_is_modern	Pass
	property_value.py::TestGetAgeBand	test_25_years_is_established	Pass
	property_value.py::TestGetAgeBand	test_50_years_is_period	Pass
	property_value.py::TestGetAgeBand	test_default_current_year	Pass
	property_value.py::TestGetAgeBand	test_established	Pass
	property_value.py::TestGetAgeBand	test_exactly_24_years_is_modern	Pass
	property_value.py::TestGetAgeBand	test_exactly_49_years_is_established	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	property_value.py::TestGetAgeBand	test_exactly_99_years_is_period	Pass
	property_value.py::TestGetAgeBand	test_exactly_9_years_is_new_build	Pass
	property_value.py::TestGetAgeBand	test_heritage_old	Pass
	property_value.py::TestGetAgeBand	test_modern	Pass
	property_value.py::TestGetAgeBand	test_new_build	Pass
	property_value.py::TestGetAgeBand	test_period	Pass
	property_value.py::TestGetProximityZone	test_close_high_risk_high	Pass
	property_value.py::TestGetProximityZone	test_close_high_risk_medium	Pass
	property_value.py::TestGetProximityZone	test_close_high_risk_very_high	Pass
	property_value.py::TestGetProximityZone	test_close_low_risk_low	Pass
	property_value.py::TestGetProximityZone	test_close_low_risk_very_low	Pass
	property_value.py::TestGetProximityZone	test_exactly_200m_is_medium_zone	Pass
	property_value.py::TestGetProximityZone	test_exactly_500m_is_far	Pass
	property_value.py::TestGetProximityZone	test_far_zone	Pass
	property_value.py::TestGetProximityZone	test_medium_zone	Pass
	property_value.py::TestGetProximityZone	test_medium_zone_upper_bound	Pass
	display.py::TestBlotterTradeExistence	test_blotter_glob_pattern_matches_prs_id_protocol	Pass
	display.py::TestBlotterTradeExistence	test_blotter_returns_trades_with_fixture	Pass
	display.py::TestStormDisplayFormat	test_all_files_include_storm_id_in_label	Pass
	display.py::TestStormDisplayFormat	test_all_files_use_sentinel_or_pipe_separator	Pass
	display.py::TestStormDisplayFormat	test_canonical_format_defined_in_config_for_format	Pass
	display.py::TestStormDisplayFormat	test_no_file_uses_old_precipitation_label	Pass
	display.py::TestStormDisplayFormat	test_no_file_uses_old_severe_levels_format	Pass
	display.py::TestStormDisplayFormat	test_port_stress_uses_sentinel_with_warning	Pass
	display.py::TestStormDisplayFormat	test_sp_table_uses_sentinel_pattern	Pass
	display.py::TestStormDisplayFormat	test_storm_option_js_custom_var	Pass
	display.py::TestStormDisplayFormat	test_storm_option_js_peak_variant	Pass
	display.py::TestStormDisplayFormat	test_storm_option_js_warning_variant	Pass
	display.py::TestStormDisplayFormat	test_trading_stress_uses_sentinel_with_peak	Pass
	display.py::TestStormDisplayFormat	test_visual_files_use_sentinel_not_hardcoded_format	Pass
	fixtures.py::TestProtocolUniformity	test_all_entity_suffixes_have_same_length	Pass
	fixtures.py::TestProtocolUniformity	test_all_uuid4_suffixes_produce_lowercase_hex	Pass
	fixtures.py::TestProtocolUniformity	test_gauge_prop_mort_storm_share_same_suffix_derivation	Pass
	fixtures.py::TestTestFixtureGaugeIDs	test_backward_compat_gauge_ids_are_documented	Pass
	fixtures.py::TestTestFixtureGaugeIDs	test_gauge_json_and_gaugehc_ids_are_consistent	Pass
	fixtures.py::TestTestFixtureGaugeIDs	test_gaugehc_new_entries_follow_protocol	Pass
	fixtures.py::TestTestFixtureGaugeIDs	test_new_conftest_gauge_ids_match_protocol	Pass
	fixtures.py::TestTestFixtureGaugeIDs	test_stress_storm_gauge_ids_are_in_known_set	Pass
	fixtures.py::TestTestFixtureStormIDs	test_port_stress_routes_storm_constants_match_protocol	Pass
	fixtures.py::TestTestFixtureStormIDs	test_sample_port_stress_storms_ids_match_protocol	Pass
	fixtures.py::TestTestFixtureStormIDs	test_sample_stress_storms_storm_ids_match_protocol	Pass
	fixtures.py::TestTestFixtureStormIDs	test_stress_routes_storm_constants_match_protocol	Pass
	generators.py::TestGaugeIDGenerator	test_generator_does_not_use_sequential_format	Pass
	generators.py::TestGaugeIDGenerator	test_generator_source_uses_deterministic_ids	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	generators.py::TestGaugeIDGenerator	test_generator_source_uses_gauge_prefix	Pass
	generators.py::TestGaugeIDGenerator	test_uuid4_produces_protocol_matching_ids	Pass
	generators.py::TestMortgageIDGenerator	test_generator_does_not_use_sequential_format	Pass
	generators.py::TestMortgageIDGenerator	test_generator_source_uses_mort_prefix	Pass
	generators.py::TestMortgageIDGenerator	test_generator_source_uses_uuid	Pass
	generators.py::TestMortgageIDGenerator	test_uuid4_produces_protocol_matching_ids	Pass
	generators.py::TestPRSTradeIDGenerator	test_book_thames_uses_prs_prefix	Pass
	generators.py::TestPRSTradeIDGenerator	test_book_thames_uses_uuid	Pass
	generators.py::TestPRSTradeIDGenerator	test_uuid4_hex_upper_produces_protocol_matching_ids	Pass
	generators.py::TestPropertyIDGenerator	test_generator_does_not_use_sequential_format	Pass
	generators.py::TestPropertyIDGenerator	test_generator_source_uses_prop_prefix	Pass
	generators.py::TestPropertyIDGenerator	test_generator_source_uses_uuid	Pass
	generators.py::TestPropertyIDGenerator	test_uuid4_produces_protocol_matching_ids	Pass
	generators.py::TestStormIDGenerator	test_multi_storm_make_storm_id_follows_protocol	Pass
	generators.py::TestStormIDGenerator	test_single_storm_generator_does_not_use_sequential_format	Pass
	generators.py::TestStormIDGenerator	test_single_storm_generator_uses_storm_prefix	Pass
	generators.py::TestStormIDGenerator	test_single_storm_generator_uses_uuid	Pass
	generators.py::TestStormIDGenerator	test_uuid4_produces_protocol_matching_ids	Pass
	patterns.py::TestGaugeIDPattern	test_accepts_valid_uuid_derived_ids	Pass
	patterns.py::TestGaugeIDPattern	test_rejects_non_hex_chars	Pass
	patterns.py::TestGaugeIDPattern	test_rejects_sequential_integers	Pass
	patterns.py::TestGaugeIDPattern	test_rejects_uppercase_hex	Pass
	patterns.py::TestGaugeIDPattern	test_rejects_wrong_length	Pass
	patterns.py::TestMortgageIDPattern	test_accepts_valid	Pass
	patterns.py::TestMortgageIDPattern	test_rejects_sequential	Pass
	patterns.py::TestPRSTradeIDPattern	test_accepts_lowercase	Pass
	patterns.py::TestPRSTradeIDPattern	test_accepts_uppercase	Pass
	patterns.py::TestPRSTradeIDPattern	test_rejects_non_hex_suffix	Pass
	patterns.py::TestPropertyIDPattern	test_accepts_valid	Pass
	patterns.py::TestPropertyIDPattern	test_rejects_sequential	Pass
	patterns.py::TestPropertyIDPattern	test_rejects_uppercase	Pass
	patterns.py::TestStormIDPattern	test_accepts_valid	Pass
	patterns.py::TestStormIDPattern	test_rejects_descriptive_labels	Pass
	patterns.py::TestStormIDPattern	test_rejects_sequential_integers	Pass
	patterns.py::TestStormIDPattern	test_rejects_uppercase	Pass
	patterns.py::TestStormIDPattern	test_rejects_wrong_prefix_format	Pass
	test_book_common_coverage_b6.py::TestBuildGlobalRecordValidationWarning	test_build_global_record_validation_logged	Pass
	test_book_common_coverage_b6.py::TestLoadCounterpartyFallback	test_empty_counterparty_file_uses_fallback	Pass
	test_book_common_coverage_b6.py::TestLoadCounterpartyFallback	test_header_only_counterparty_file_uses_fallback	Pass
	test_book_common_coverage_b6.py::TestLoadCounterpartyFallback	test_populated_counterparty_file_uses_loaded_data	Pass
	test_book_coverage.py::TestEmptyCurves	test_curves_raises_value_error	Pass
	test_book_coverage.py::TestFallbackCounterparty	test_empty_counterparty_file_uses_fallback	Pass
	test_book_coverage.py::TestFallbackCounterparty	test_missing_counterparty_file_uses_fallback	Pass
	test_book_coverage.py::TestGenerateTradeFailure	test_returns_empty_list	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_book_coverage.py::TestGenerateTodayPartial	test_partial_failure_returns_successful_only	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_contact_info_fields_extracted	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_counterparty_present	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_empty_data_returns_dict	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_date_returns_flat_dict	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_invalid_data_raises_value_error	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_none_values_excluded	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_get_required_fields_returns_list	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_schema_is_dict	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_empty_counterparties_adds_error	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_empty_data_returns_errors	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_invalid_counterparty_role_adds_error	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_missing_counterparty_role_adds_error	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_missing_party_id_adds_error	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_missing_party_name_adds_error	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_missing_party_ref_adds_error	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_two_capital_same_role_adds_error	Pass
	ctpy_mapping.py::TestCounterpartyDMCMapping	test_valid_data_no_errors	Pass
	class_attrs.py::TestClassAttributes	test_metadata_categories_contains_expected_keys	Pass
	class_attrs.py::TestClassAttributes	test_metadata_categories_station_has_required_subkeys	Pass
	class_attrs.py::TestClassAttributes	test_schema_version_is_string	Pass
	class_attrs.py::TestGaugeHistoricalDailyUnit	test_available_flag_is_bool	Pass
	directory_main.py::TestMain	test_main_skips_bad_file_without_raising	Pass
	directory_main.py::TestMain	test_main_with_csv_files_runs_without_error	Pass
	directory_main.py::TestMain	test_main_with_no_csv_files_runs_without_error	Pass
	filter.py::TestFilterByYears	test_empty_flows_returns_empty	Pass
	filter.py::TestFilterByYears	test_filter_1_year_keeps_only_recent	Pass
	filter.py::TestFilterByYears	test_filter_keeps_recent	Pass
	filter.py::TestFilterByYears	test_none_years_returns_all	Pass
	filter.py::TestFilterByYearsBoundary	test_all_dates_within_range_kept	Pass
	filter.py::TestFilterByYearsBoundary	test_zero_years_returns_only_most_recent_year	Pass
	generate.py::TestGenerate	test_daily_flows_filtered_by_years	Pass
	generate.py::TestGenerate	test_explicit_gauge_id_overrides_metadata	Pass
	generate.py::TestGenerate	test_output_file_written	Pass
	generate.py::TestGenerate	test_returns_dict_with_required_keys	Pass
	generate.py::TestGenerateUsesInstance	test_explicit_years_overrides_instance_default	Pass
	generate.py::TestGenerateUsesInstance	test_instance_default_years_used_when_no_years_arg	Pass
	generate.py::TestGenerateWithNoOutputPath	test_generate_output_path_none_uses_config_input_dir	Pass
	metadata.py::TestGetNrfaMetadataForGauge	test_returns_expected_keys	Pass
	metadata.py::TestGetNrfaMetadataForGauge	test_truncates_data_quality_notes	Pass
	metadata.py::TestGetNrfaMetadataForGauge	test_all_expected_keys_present	Pass
	metadata.py::TestGetNrfaMetadataForGauge	test_comments_key_always_none	Pass
	metadata.py::TestGetNrfaMetadataForGauge	test_custom_missing_keys	Pass
	metadata.py::TestGetNrfaMetadataForGauge	test_database_missing_keys_from_station_metadata	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	metadata.py::TestGetNrfaMetadataFromMissingKeys	test_defaults_flow_units_m3_per_s	Pass
	metadata.py::TestGetNrfaMetadataFromMissingKeys	test_empty_data_quality_notes_when_no_description	Pass
	metadata.py::TestGetNrfaMetadataFromMissingKeys	test_missing_data_key_source_returns_none	Pass
	metadata.py::TestGetNrfaMetadataFromMissingKeys	test_missing_stage_keys_return_none	Pass
	parse_csv.py::TestParseNrfaCsv	test_flow_values_are_floats	Pass
	parse_csv.py::TestParseNrfaCsv	test_flows_have_date_and_flow_cumecs	Pass
	parse_csv.py::TestParseNrfaCsv	test_invalid_rows_skipped	Pass
	parse_csv.py::TestParseNrfaCsv	test_metadata_contains_station_name	Pass
	parse_csv.py::TestParseNrfaCsv	test_metadata_station_id	Pass
	parse_csv.py::TestParseNrfaCsv	test_returns_metadata_and_flows	Pass
	parse_csv.py::TestParseNrfaCsv	test_short_row_skipped	Pass
	parse_csv.py::TestParseNrfaCsvMetadataEdgeCases	test_all_metadata_categories_parsed	Pass
	parse_csv.py::TestParseNrfaCsvMetadataEdgeCases	test_flow_row_with_non_numeric_value_skipped	Pass
	parse_csv.py::TestParseNrfaCsvMetadataEdgeCases	test_metadata_row_with_no_value_uses_empty_string	Pass
	parse_csv.py::TestParseNrfaCsvMetadataEdgeCases	test_metadata_row_with_three_columns_stores_value	Pass
	process_directory.py::TestProcessDirectory	test_empty_directory_returns_empty_list	Pass
	process_directory.py::TestProcessDirectory	test_non_matching_files_skipped	Pass
	process_directory.py::TestProcessDirectory	test_output_dir_none_uses_input_dir	Pass
	process_directory.py::TestProcessDirectory	test_output_files_created	Pass
	process_directory.py::TestProcessDirectory	test_process_directory_error_continues	Pass
	process_directory.py::TestProcessDirectory	test_processes_matching_files	Pass
	statistics.py::TestCalculateStatistics	test_annual_maxima_present	Pass
	statistics.py::TestCalculateStatistics	test_basic_stats_keys	Pass
	statistics.py::TestCalculateStatistics	test_empty_returns_empty_dict	Pass
	statistics.py::TestCalculateStatistics	test_extreme_events_above_p99	Pass
	statistics.py::TestCalculateStatistics	test_mean_correct	Pass
	statistics.py::TestCalculateStatistics	test_min_max_correct	Pass
	statistics.py::TestCalculateStatistics	test_monthly_means_present	Pass
	statistics.py::TestCalculateStatistics	test_percentiles_present	Pass
	statistics.py::TestCalculateStatistics	test_single_flow_no_std_dev_error	Pass
	statistics.py::TestCalculateStatistics	test_all_percentile_keys_present	Pass
	statistics.py::TestCalculateStatistics	test_annual_maxima_events_capped_at_ten	Pass
	statistics.py::TestCalculateStatistics	test_extreme_events_top_10_capped	Pass
	statistics.py::TestCalculateStatistics	test_max_flow_date_correctly_identified	Pass
	statistics.py::TestCalculateStatistics	test_p5_flow_equals_p95	Pass
	statistics.py::TestCalculateStatistics	test_p95_flow_equals_p5	Pass
	statistics.py::TestCalculateStatistics	test_records_start_and_end_match_data	Pass
	statistics.py::TestCalculateStatistics	test_single_year_annual_maxima_std_dev_zero	Pass
	statistics.py::TestCalculateStatistics	test_two_years_approximate	Pass
	statistics.py::TestCalculateStatistics	test_zero_flow_std_dev_non_zero	Pass
	schemas_part1.py::TestFloodGaugeGDM	test_create_mapping_returns_flat_dict	Pass
	schemas_part1.py::TestFloodGaugeGDM	test_create_mapping_with_empty_input	Pass
	schemas_part1.py::TestFloodGaugeGDM	test_get_field_info_for_gauge_id	Pass
	schemas_part1.py::TestFloodGaugeGDM	test_required_fields_include_gauge_id	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	schemas_part1.py::TestFloodGaugeCDM	test_schema_has_flood_gauge_root	Pass
	schemas_part1.py::TestFloodGaugeCDM	test_schema_has_header_section	Pass
	schemas_part1.py::TestFloodGaugeCDM	test_validate_empty_data_returns_errors	Pass
	schemas_part1.py::TestFloodGaugeCDM	test_validate_valid_gauge_data	Pass
	schemas_part1.py::TestMortgageCDM	test_create_mapping_returns_flat_dict	Pass
	schemas_part1.py::TestMortgageCDM	test_required_fields_include_mortgage_id	Pass
	schemas_part1.py::TestMortgageCDM	test_schema_has_mortgage_root	Pass
	schemas_part1.py::TestMortgageCDM	test_validate_valid_mortgage	Pass
	schemas_part1.py::TestPropertyCDM	test_create_mapping_returns_flat_dict	Pass
	schemas_part1.py::TestPropertyCDM	test_required_fields_not_empty	Pass
	schemas_part1.py::TestPropertyCDM	test_schema_has_property_root	Pass
	schemas_part1.py::TestPropertyCDM	test_validate_valid_property	Pass
	schemas_part1.py::TestStormEventsCDM	test_create_mapping_exception_raises	Pass
	schemas_part1.py::TestStormEventsCDM	test_create_mapping_full_data	Pass
	schemas_part1.py::TestStormEventsCDM	test_create_mapping_returns_flat_dict	Pass
	schemas_part1.py::TestStormEventsCDM	test_get_required_fields_returns_list	Pass
	schemas_part1.py::TestStormEventsCDM	test_required_fields_include_storm_id	Pass
	schemas_part1.py::TestStormEventsCDM	test_schema_has_storm_root	Pass
	schemas_part1.py::TestStormEventsCDM	test_validate_empty_data	Pass
	schemas_part1.py::TestStormEventsCDM	test_validate_exception_returns_error_dict	Pass
	schemas_part1.py::TestStormEventsCDM	test_validate_missing_catchment_id	Pass
	schemas_part1.py::TestStormEventsCDM	test_validate_missing_storm_id	Pass
	schemas_part1.py::TestStormEventsCDM	test_validate_valid_storm	Pass
	schemas_part2.py::TestPhysicalRiskSwapCDM	test_create_mapping_returns_flat_dict	Pass
	schemas_part2.py::TestPhysicalRiskSwapCDM	test_required_fields_has_gauge_set	Pass
	schemas_part2.py::TestPhysicalRiskSwapCDM	test_required_fields_include_swap_id	Pass
	schemas_part2.py::TestPhysicalRiskSwapCDM	test_schema_has_root	Pass
	schemas_part2.py::TestStormTimeSeriesCDM	test_create_mapping_exception_raises_value_error	Pass
	schemas_part2.py::TestStormTimeSeriesCDM	test_create_mapping_full_data	Pass
	schemas_part2.py::TestStormTimeSeriesCDM	test_create_mapping_returns_flat_dict	Pass
	schemas_part2.py::TestStormTimeSeriesCDM	test_create_reading_mapping	Pass
	schemas_part2.py::TestStormTimeSeriesCDM	test_get_required_fields_returns_list	Pass
	schemas_part2.py::TestStormTimeSeriesCDM	test_required_fields_include_ts_id	Pass
	schemas_part2.py::TestStormTimeSeriesCDM	test_schema_has_root	Pass
	schemas_part2.py::TestStormTimeSeriesCDM	test_validate_empty_data	Pass
	schemas_part2.py::TestStormTimeSeriesCDM	test_validate_exception_returns_error_dict	Pass
	schemas_part2.py::TestStormTimeSeriesCDM	test_validate_missing_catchment_id	Pass
	schemas_part2.py::TestStormTimeSeriesCDM	test_validate_missing_ts_id	Pass
	schemas_part2.py::TestStormTimeSeriesCDM	test_validate_valid_data	Pass
	test_block1_coverage_base.py::TestBaseCDMCoverage	test_required_fields_default	Pass
	test_block1_coverage_base.py::TestBaseCDMCoverage	test_field_info_leaf_returns_dict	Pass
	test_block1_coverage_base.py::TestBaseCDMCoverage	test_field_info_not_found_returns_none	Pass
	test_block1_coverage_base.py::TestBaseCDMCoverage	test_field_info_partial_path_returns_none	Pass
	test_block1_coverage_base.py::TestBaseCDMCoverage	test_field_info_returns_none_for_non_dict_leaf	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_block1_coverage_base.py::TestBaseCDMCoverage	test_list_all_fields_nested_metadata_keys	Pass
	test_block1_coverage_base.py::TestBaseCDMCoverage	test_list_all_fields_prs	Pass
	test_block1_coverage_base.py::TestBaseCDMCoverage	test_list_all_fields_returns_nested_paths	Pass
	test_block1_coverage_base.py::TestBaseCDMCoverage	test_list_all_fields_skips_metadata_keys	Pass
	test_block1_coverage_base.py::TestBaseCDMCoverage	test_repr	Pass
	test_block1_coverage_base.py::TestGaugeValidator	test_exception_returns_validation_error	Pass
	test_block1_coverage_base.py::TestGaugeValidator	test_invalid_menu_value_produces_sensor_error	Pass
	test_block1_coverage_base.py::TestGaugeValidator	test_multiple_invalid_menu_values	Pass
	test_block1_coverage_base.py::TestGaugeValidator	test_valid_menu_values_no_sensor_errors	Pass
	test_block1_coverage_base.py::TestPRSCDMMapping	test_custom_gauge_basket_size	Pass
	test_block1_coverage_base.py::TestPRSCDMMapping	test_default_gauge_basket_size	Pass
	test_block1_coverage_base.py::TestPropertyValidator	test_exception_returns_validation_error	Pass
	test_block1_coverage_base.py::TestPropertyValidator	test_get_required_fields	Pass
	test_block1_coverage_base.py::TestPropertyValidator	test_property_errors_collected	Pass
	test_block1_coverage_base.py::TestPropertyValidator	test_missing_latitude	Pass
	test_block1_coverage_base.py::TestPropertyValidator	test_missing_longitude	Pass
	test_block1_coverage_base.py::TestPropertyValidator	test_valid_returns_empty	Pass
	test_block1_coverage_validation.py::TestCreateMappingCDM	test_exception_raises_value_error	Pass
	test_block1_coverage_validation.py::TestValidateCDMValid	test_valid_returns_empty	Pass
	test_block1_coverage_validation.py::TestValidateCDMException	test_exception_returns_error_dict	Pass
	test_block1_coverage_validation.py::TestValidateCDMValid	test_header_errors_collected	Pass
	test_block1_coverage_validation.py::TestValidateCDMMissing	test_missing_attachment_id	Pass
	test_block1_coverage_validation.py::TestValidateCDMMissing	test_missing_mortgage_id	Pass
	test_block1_coverage_validation.py::TestValidateCDMMissing	test_missing_original_loan	Pass
	test_block1_coverage_validation.py::TestValidateCDMMissing	test_missing_property_id	Pass
	test_block1_coverage_validation.py::TestValidateCDMValid	test_terms_errors_collected	Pass
	test_block1_coverage_validation.py::TestCreateMappingPRSCDM	test_exception_raises_value_error	Pass
	test_block1_coverage_validation.py::TestCreateMappingPRSCDM	test_full_data	Pass
	test_block1_coverage_validation.py::TestValidatePRSCDM	test_empty_data	Pass
	test_block1_coverage_validation.py::TestValidatePRSCDM	test_empty_gauge_basket	Pass
	test_block1_coverage_validation.py::TestValidatePRSCDM	test_exception_returns_error_dict	Pass
	test_block1_coverage_validation.py::TestValidatePRSCDM	test_gauge_errors_appended	Pass
	test_block1_coverage_validation.py::TestValidatePRSCDM	test_header_errors_appended	Pass
	test_block1_coverage_validation.py::TestValidatePRSCDM	test_missing_attachment_id	Pass
	test_block1_coverage_validation.py::TestValidatePRSCDM	test_missing_gauge_set_id	Pass
	test_block1_coverage_validation.py::TestValidatePRSCDM	test_missing_swap_id	Pass
	test_block1_coverage_validation.py::TestValidatePRSCDM	test_valid_returns_empty	Pass
	test_block2_cdm_coverage.py::TestGaugeMappingCoverage	test_create_mapping_exception_raises_value_error	Pass
	test_block2_cdm_coverage.py::TestGaugeMappingCoverage	test_get_mapping_fields_returns_expected_list	Pass
	test_block2_cdm_coverage.py::TestGaugeBulkDirectoryCoverage	test_bulk_dir_csv_files	Pass
	test_block2_cdm_coverage.py::TestGaugeBulkDirectoryCoverage	test_bulk_dir_with_csv_files	Pass
	test_block2_cdm_coverage.py::TestGaugeProcessDirectoryCoverage	test_process_dir_default_output	Pass
	test_block2_cdm_coverage.py::TestGaugeProcessDirectoryCoverage	test_process_dir_file_error_logged	Pass
	test_block2_cdm_coverage.py::TestPropertyMappingCoverage	test_property_mapping_default_elevation	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_block2_cdm_coverage.py::TestToCreateMappingCoverage	test_create_mapping_exception_raises_value_error	Pass
	test_ctpy_coverage_b6.py::TestValidateHexadecimal	test_validate_exception_message_is_string	Pass
	test_ctpy_coverage_b6.py::TestValidateHexadecimal	test_validate_with_integer_returns_validation_error	Pass
	test_ctpy_coverage_b6.py::TestValidateHexadecimal	test_validate_with_non_dict_returns_validation_error	Pass
	counterparty_generator.py::TestCounterpartyGenerate	Counterparty_output_file	Pass
	counterparty_generator.py::TestCounterpartyGenerate	Counterparty_list	Pass
	counterparty_generator.py::TestCounterpartyGenerate	default_count_uses_all_counterparties	Pass
	counterparty_generator.py::TestCounterpartyGenerate	output_path_is_pathlib_path	Pass
	counterparty_generator.py::TestCounterpartyGenerate	metadata_has_total_generated	Pass
	counterparty_generator.py::TestCounterpartyGenerate	output_file_is_valid_json	Pass
	counterparty_generator.py::TestCounterpartyGenerate	counterparty_dict_with_required_keys	Pass
	counterparty_generator.py::TestCounterpartyGenerate	zero_count_returns_empty	Pass
	counterparty_generator.py::TestGenerateCounterpartiesFunction	generate_counterparties	Pass
	counterparty_generator.py::TestGenerateCounterpartiesFunction	returns_counterparties	Pass
	counterparty_generator.py::TestCreditSupport	credit_support_has_csa	Pass
	counterparty_generator.py::TestCreditSupport	credit_support_parties_list_has_two_parties	Pass
	counterparty_generator.py::TestCreditSupport	credit_support_has_counterparty_set	Pass
	counterparty_generator.py::TestCreditSupport	credit_support_master_agreement_is_true	Pass
	counterparty_generator.py::TestCounterparty	counterparty_has_accounts	Pass
	counterparty_generator.py::TestCounterparty	counterparty_has_contact_info	Pass
	counterparty_generator.py::TestCounterparty	counterparty_has_id	Pass
	counterparty_generator.py::TestCounterparty	counterparty_has_name	Pass
	counterparty_generator.py::TestCounterparty	counterparty_has_natural_person	Pass
	counterparty_generator.py::TestCreditPlatform	credit_platform_has_credit_rating	Pass
	counterparty_generator.py::TestCreditSupport	credit_support_party_ids	Pass
	test_counterparty_coverage_b6.py::TestVerboseLogging	test_verbose_logging_no_per_counterparty_logs	Pass
	test_counterparty_coverage_b6.py::TestVerboseLogging	test_verbose_logging_logs_per_counterparty	Pass
	test_counterparty_coverage_b6.py::TestVerboseLogging	test_verbose_logging_logs_summary	Pass
	business_days.py::TestBusinessDays	test_chronological_order	Pass
	business_days.py::TestBusinessDays	test_ends_before_start	Pass
	business_days.py::TestBusinessDays	test_large_count	Pass
	business_days.py::TestBusinessDays	test_last_day_is_day_before_start	Pass
	business_days.py::TestBusinessDays	test_no_weekends	Pass
	business_days.py::TestBusinessDays	test_returns_correct_count	Pass
	business_days.py::TestBusinessDays	test_single_day	Pass
	business_days.py::TestBusinessDays	test_start_on_monday_skips_weekend	Pass
	business_days.py::TestBusinessDays	test_start_on_wednesday	Pass
	business_days.py::TestSimulationParameters	test_business_days_is_three_months	Pass
	business_days.py::TestSimulationParameters	test_daily_vol_is_small	Pass
	business_days.py::TestSimulationParameters	test_floors_is_positive	Pass
	business_days.py::TestSimulationParameters	test_max_move_at_least_ten_daily_vols	Pass
	business_days.py::TestSimulationParameters	test_max_move_bounds_vol	Pass
	history_files_part1.py::TestGenerateHistoryFiles	test_empty_dir_produces_no_file	Pass
	history_files_part1.py::TestGenerateHistoryFiles	test_last_day_entries_have_date	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	history_files_part1.py::TestGenerateHistoryFile	test_history_entries_have_tenors	Pass
	history_files_part1.py::TestGenerateHistoryFile	test_history_keyed_by_gauge	Pass
	history_files_part1.py::TestGenerateHistoryFile	test_history_keyed_by_trigger	Pass
	history_files_part1.py::TestGenerateHistoryFile	test_metadata_has_generated_at	Pass
	history_files_part1.py::TestGenerateHistoryFile	test_metadata_has_snapshots	Pass
	history_files_part1.py::TestGenerateHistoryFile	test_multiple_gauges_in_metadata	Pass
	history_files_part1.py::TestGenerateHistoryFile	test_multiple_snapshots_produce_multiple_entries	Pass
	history_files_part1.py::TestGenerateHistoryFile	test_output_has_metadata_and_history	Pass
	history_files_part1.py::TestGenerateHistoryFile	test_single_snapshot_creates_file	Pass
	history_files_part2.py::TestGenerateTradePlHistoryFile	test_empty_history_produces_no_file	Pass
	history_files_part2.py::TestGenerateTradePlHistoryFile	test_history_entry_structure	Pass
	history_files_part2.py::TestGenerateTradePlHistoryFile	test_metadata_has_generated_at	Pass
	history_files_part2.py::TestGenerateTradePlHistoryFile	test_metadata_has_snapshots	Pass
	history_files_part2.py::TestGenerateTradePlHistoryFile	test_multiple_days_builds_history_vector	Pass
	history_files_part2.py::TestGenerateTradePlHistoryFile	test_multiple_trades_in_snapshot	Pass
	history_files_part2.py::TestGenerateTradePlHistoryFile	test_output_has_metadata_and_trades	Pass
	history_files_part2.py::TestGenerateTradePlHistoryFile	test_pnl_history_missing_swap_id_is_skipped	Pass
	history_files_part2.py::TestGenerateTradePlHistoryFile	test_single_snapshot_creates_file	Pass
	history_files_part2.py::TestGenerateTradePlHistoryFile	test_trades_keyed_by_swap_id	Pass
	history_files_part2.py::TestGenerateTradePlHistoryFile	test_trade_metadata_fields	Pass
	gaugets_part1.py::TestGaugeTimeSeriesDefaults	test_default_simulation_hours_is_168	Pass
	gaugets_part1.py::TestGaugeTimeSeriesDefaults	test_generate_default_argument_is_168	Pass
	gaugets_part1.py::TestGaugeTimeSeriesDefaults	test_gaugets_function_default_is_168	Pass
	gaugets_part1.py::TestGaugeTimeSeriesDefaults	test_peak_hour_max_within_simulation_window	Pass
	gaugets_part1.py::TestGaugeTimeSeriesDefaults	test_peak_hour_min_less_than_max	Pass
	gaugets_part1.py::TestGaugeTimeSeriesDefaults	test_168_produces_more_readings_than_60	Pass
	gaugets_part1.py::TestGaugeTimeSeriesConfigure	test_configure_known_param	Pass
	gaugets_part1.py::TestGaugeTimeSeriesConfigure	test_configure_unknown_param_ignored	Pass
	gaugets_part1.py::TestGaugeTimeSeriesGenerate	test_generate_168_hours_produces_168_readings	Pass
	gaugets_part1.py::TestGaugeTimeSeriesGenerate	test_generate_60_hours_stores_60_in_metadata	Pass
	gaugets_part1.py::TestGaugeTimeSeriesMissing	test_missing_gauge_file_raises	Pass
	gaugets_part1.py::TestGaugeTimeSeriesSimulation	test_simulation_hours_stored_in_output	Pass
	gaugets_part1.py::TestGaugeTimeSeriesTmpGauge	test_tmp_gauge_files_removed	Pass
	gaugets_part1.py::TestGaugetsRandomDefaults	test_default_peak_hour_max_within_window	Pass
	gaugets_part1.py::TestGaugetsRandomDefaults	test_default_simulation_hours_is_168	Pass
	gaugets_part2.py::TestGaugetsDateTimeEncoder	test_encodes_datetime	Pass
	gaugets_part2.py::TestGaugetsDateTimeEncoder	test_encodes_numpy_float32	Pass
	gaugets_part2.py::TestGaugetsDateTimeEncoder	test_encodes_numpy_integer	Pass
	gaugets_part2.py::TestGaugetsDateTimeEncoder	test_encodes_numpy_ndarray	Pass
	gaugets_part2.py::TestGaugetsDateTimeEncoder	test_fallback_raises_for_unknown	Pass
	gaugets_part2.py::TestGaugetsMain	test_main_block	Pass
	gaugets_part2.py::TestGenerateGaugetsCalls	test_generate_gaugets_calls_generator	Pass
	test_gauge_random_coverage.py::TestFindValueNoDefault	test_find_value_no_default_returns_empty_string	Pass
	test_gauge_random_coverage.py::TestFindValueNoDefault	test_find_value_no_default_returns_empty_string	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_gauge_random_coverage.py::	TestFieldValueNoDef::test_string_field_returned_as_is	Pass
	test_gauge_random_coverage.py::	TestFieldTypeDefaultsToText::test_unknown_type_defaults_to_text	Pass
	test_gauge_random_coverage.py::	TestFieldValueUnknownTypeReturns::test_unknown_type_returns_empty	Pass
	test_gauge_random_coverage.py::	TestGroupNameFallback::test_empty_name_falls_back_back_to_area	Pass
	test_gauge_random_coverage.py::	TestGroupNameFallback::test_group_name_key_falls_back_back_to_area	Pass
	test_gauge_random_coverage.py::	TestIntegerDefault::test_integer_field_returns_int	Pass
	test_gauge_random_coverage.py::	TestMenuOptionsDefault::test_menu_options_returns_empty	Pass
	test_gauge_random_coverage.py::	TestMenuOptionsDefault::test_menu_options_cycles	Pass
	test_gauge_random_coverage.py::	TestMenuOptionsDefault::test_menu_options_wraps	Pass
	test_gaugehd_runner_part1.py::	TestGenerateAllErrorPath::test_empty_portfolio_returns_empty	Pass
	test_gaugehd_runner_part1.py::	TestGenerateAllErrorPath::test_error_extracts_gauge_id_from_nested_header	Pass
	test_gaugehd_runner_part1.py::	TestGenerateAllErrorPath::test_error_with_no_header_uses_unknown	Pass
	test_gaugehd_runner_part1.py::	TestGenerateAllErrorPath::test_mixed_good_and_bad_gauges	Pass
	test_gaugehd_runner_part1.py::	TestStaleFileClean::test_does_not_remove_non_matching_files	Pass
	test_gaugehd_runner_part1.py::	TestStaleFileClean::test_removes_multiple_stale_files	Pass
	test_gaugehd_runner_part1.py::	TestStaleFileClean::test_removes_stale_gauge_files	Pass
	test_gaugehd_runner_part2.py::	TestMain::test_main_default_synthetic	Pass
	test_gaugehd_runner_part2.py::	TestMain::test_main_no_catchment_leaves_config_unchanged	Pass
	test_gaugehd_runner_part2.py::	TestMain::test_main_short_catchment_flag	Pass
	test_gaugehd_runner_part2.py::	TestMain::test_main_short_years_flag	Pass
	test_gaugehd_runner_part2.py::	TestMain::test_main_with_catchment	Pass
	test_gaugehd_runner_part2.py::	TestMain::test_main_with_nrfa_dir	Pass
	test_gaugehd_runner_part2.py::	TestMain::test_main_with_years	Pass
	test_gaugehd_runner_part2.py::	TestProcessNrfaDirectory::test_processes_nrfa_directory_pattern	Pass
	test_gaugehd_runner_part2.py::	TestProcessNrfaDirectory::test_processes_nrfa_directory_continues	Pass
	test_gaugehd_runner_part2.py::	TestProcessNrfaDirectory::test_processes_nrfa_directory_runner	Pass
	test_gaugehd_runner_part2.py::	TestProcessNrfaDirectory::test_processes_nrfa_directory_filename	Pass
	test_gaugehd_synthetic.py::	TestGaugeHistoricalDaily::test_calculate_statistics_from_levels	Pass
	test_gaugehd_synthetic.py::	TestGaugeHistoricalDaily::test_calculate_statistics_via_class	Pass
	test_gaugehd_synthetic.py::	TestGaugeHistoricalDaily::test_historical_years	Pass
	test_gaugehd_synthetic.py::	TestGaugeHistoricalDaily::test_default_years	Pass
	test_gaugehd_synthetic.py::	TestGaugeHistoricalDaily::test_historical_years_explicit	Pass
	test_gaugehd_synthetic.py::	TestGaugeHistoricalDaily::test_historical_years_none	Pass
	test_gaugehd_synthetic.py::	TestGaugeHistoricalDaily::test_generate_from_gauge_portfolio_explicit_years	Pass
	test_gaugehd_synthetic.py::	TestGaugeHistoricalDaily::test_generate_from_gauge_portfolio_via_class	Pass
	test_gaugehd_synthetic.py::	TestGaugeHistoricalDaily::test_generate_from_nrfa_uses_default_years	Pass
	test_gaugehd_synthetic.py::	TestGaugeHistoricalDaily::test_generate_from_nrfa_via_class	Pass
	test_gaugehd_synthetic.py::	TestGaugeHistoricalDaily::test_generate_synthetic_timeseries_via_class	Pass
	test_gaugehd_synthetic.py::	TestGaugeHistoricalDaily::test_historical_csv_via_class	Pass
	test_gaugehd_synthetic.py::	TestGaugeFromGaugePortfolio::test_from_gauge_portfolio	Pass
	test_gaugehd_synthetic.py::	TestGaugeFromGaugePortfolio::test_daily_observations_generated	Pass
	test_gaugehd_synthetic.py::	TestGaugeFromGaugePortfolio::test_gauge_metadata_extracted	Pass
	test_gaugehd_synthetic.py::	TestGaugeFromGaugePortfolio::test_from_nested_gauge_data	Pass
	test_gaugehd_synthetic.py::	TestGaugeFromGaugePortfolio::test_from_flat_gauge_data	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_gaugehd_synthetic.py::TestGenerateFromGaugeHd	test_returns_dict	Pass
	test_gaugehd_synthetic.py::TestGenerateFromGaugeHd	test_schema_version_in_portfolio	Pass
	test_gaugets_random_coverage.py::TestCalculateWaterLevel	test_explicit_water_level_overrides_defaultParams	Pass
	test_gaugets_random_coverage.py::TestCalculateWaterLevel	test_calculates_water_level_defaultsParams	Pass
	test_gaugets_random_coverage.py::TestDefaultSimparams	test_defaultSimparams_produces_data	Pass
	test_gaugets_random_coverage.py::TestFloodstatUtilities	test_floodstatutilities_parse_time	Pass
	test_gaugets_random_coverage.py::TestGaugereadingReading	test_gaugereadingReading_defaultfields	Pass
	test_gaugets_random_coverage.py::TestGaugereadingReading	test_gaugereadingReading_with_explicitfutimestamp	Pass
	test_gaugets_random_coverage.py::TestGenerateTimeSeries	test_generateTimeSeries_returns_series_structure	Pass
	test_gaugets_random_coverage.py::TestRemovingZeroes	test_removingZeroes_preserves_nonzero_amplitude	Pass
	test_gaugets_random_coverage.py::TestTimegaugeReading	test_timegaugeReading_skipped	Pass
	test_nrfa_parsing.py::TestFilterByYear	test_5_year_filter	Pass
	test_nrfa_parsing.py::TestFilterByYear	test_empty_list_returns_empty	Pass
	test_nrfa_parsing.py::TestFilterByYear	test_large_years_returns_all	Pass
	test_nrfa_parsing.py::TestFilterByYear	test_none_years_returns_all	Pass
	test_nrfa_parsing.py::TestGenerateFromNrfa	test_output_path	Pass
	test_nrfa_parsing.py::TestGenerateFromNrfa	test_creates_json_file	Pass
	test_nrfa_parsing.py::TestGenerateFromNrfa	test_gauge_id_override	Pass
	test_nrfa_parsing.py::TestGenerateFromNrfa	test_returns_dict	Pass
	test_nrfa_parsing.py::TestGenerateFromNrfa	test_schema_version_in_output	Pass
	test_nrfa_parsing.py::TestGenerateFromNrfa	test_station_metadata_extracted	Pass
	test_nrfa_parsing.py::TestParseNrfaCsv	test_empty_file	Pass
	test_nrfa_parsing.py::TestParseNrfaCsv	test_flows_have_date_and_flow	Pass
	test_nrfa_parsing.py::TestParseNrfaCsv	test_metadata_contains_station_id	Pass
	test_nrfa_parsing.py::TestParseNrfaCsv	test_metadata_contains_station_name	Pass
	test_nrfa_parsing.py::TestParseNrfaCsv	test_returns_tuple	Pass
	test_nrfa_parsing.py::TestParseNrfaCsv	test_skips_short_rows	Pass
	test_stress_storms_core.py::TestGenerateHugeIndex	test_gauge_id_present_in_all_responses	Pass
	test_stress_storms_core.py::TestGenerateHugeIndex	test_multiple_gauges_have_correct_ids	Pass
	test_stress_storms_core.py::TestGenerateStressStorms	test_raises_when_gaugets_empty	Pass
	test_stress_storms_core.py::TestGenerateStressStorms	test_index_has_gauge_ids_alert	Pass
	test_stress_storms_core.py::TestGenerateStressStorms	test_index_has_required_top_level_keys	Pass
	test_stress_storms_core.py::TestGenerateStressStorms	test_per_storm_file_has_gauge_responses	Pass
	test_stress_storms_core.py::TestGenerateStressStorms	test_storm_id_prefix_uses_config_constant	Pass
	test_stress_storms_core.py::TestGenerateStressStorms	test_total_storms_excluded	Pass
	test_stress_storms_core.py::TestGenerateStressStorms	test_total_storms_matches_list_length	Pass
	test_stress_storms_core.py::TestGenerateStressStorms	test_writes_index_and_storm_files	Pass
	test_stress_storms_core.py::TestRetrieve	test_returns_dict_with_total_storms	Pass
	test_stress_storms_core.py::TestRetrieve	test_returns_elapsed_s	Pass
	test_stress_storms_coverage.py::TestCallCorruptGauge	test_all_corrupt_gauge_files_no_storms	Pass
	test_stress_storms_coverage.py::TestCorruptGaugeFile	test_corrupt_gauge_file_skipped	Pass
	test_stress_storms_coverage.py::TestCorruptStormsJson	test_corrupt_storms_json_does_not_crash	Pass
	test_stress_storms_coverage.py::TestDerivedSeverityCategory	test_derived_severity_category_warning_returns_baseline	Pass
	test_stress_storms_coverage.py::TestDerivedSeverityCategory	test_derived_severity_category_with_warnings_returns_moderate	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_stress_storms_coverage.py::	TestDeriveIntensityCategory::test_negative_severe_with_warnings_returns_moderate	Pass
	test_stress_storms_coverage.py::	TestDeriveIntensityCategory::test_negative_severe_zero_warnings_returns_baseline	Pass
	test_stress_storms_coverage.py::	TestDeriveIntensityCategory::test_severe_overrides_warning	Pass
	test_stress_storms_coverage.py::	TestDeriveIntensityCategory::test_zero_severe_zero_warnings_baseline	Pass
	test_stress_storms_coverage.py::	TestEmptyStormIdSkipped::test_empty_storm_id_skipped	Pass
	test_stress_storms_coverage.py::	TestEmptyStormIdSkipped::test_missing_storm_id_key_skipped	Pass
	test_stress_storms_metadata.py::	TestMetadataEnrichment::test_duration_from_storms_json	Pass
	test_stress_storms_metadata.py::	TestMetadataEnrichment::test_fallback_to_config_defaults	Pass
	test_stress_storms_metadata.py::	TestMetadataEnrichment::test_missing_storms_json_does_not_crash	Pass
	test_stress_storms_metadata.py::	TestNaming::test_first_storm_gets_greek_name	Pass
	test_stress_storms_metadata.py::	TestNaming::test_name_from_storms_json_takes_priority	Pass
	test_stress_storms_metadata.py::	TestSorting::test_alert_only_is_last	Pass
	test_stress_storms_metadata.py::	TestSorting::test_severe_is_first	Pass
	test_stress_storms_metadata.py::	TestSorting::test_is_descending_by_severe	Pass
	test_stress_storms_triggers.py::	TestIntensityCategoryDerivation::test_baseline_category_derivation	Pass
	test_stress_storms_triggers.py::	TestIntensityCategoryDerivation::test_severe_category_derivation	Pass
	test_stress_storms_triggers.py::	TestIntensityCategoryDerivation::test_extreme_category_derivation	Pass
	test_stress_storms_triggers.py::	TestIntensityCategoryDerivation::test_moderate_category_derivation	Pass
	test_stress_storms_triggers.py::	TestIntensityCategoryDerivation::test_severe_category_derivation	Pass
	test_stress_storms_triggers.py::	TestTriggerSummary::test_alert_count	Pass
	test_stress_storms_triggers.py::	TestTriggerSummary::test_impacted_matches_alert	Pass
	test_stress_storms_triggers.py::	TestTriggerSummary::test_severe_count	Pass
	test_stress_storms_triggers.py::	TestTriggerSummary::test_warning_count	Pass
	test_stress_storms_triggers.py::	TestTriggerSummary::test_is_alert_when_alert_only	Pass
	test_stress_storms_triggers.py::	TestTriggerSummary::test_is_severe	Pass
	test_stress_storms_triggers.py::	TestTriggerSummary::test_is_warning_when_no_severe	Pass
	test_stress_storms_triggers.py::	TestTriggerSummary::test_summary_present	Pass
	test_synthetic_standalone.py::	TestSynthetic::test_at_least_180_synthetics	Pass
	test_synthetic_standalone.py::	TestSynthetic::test_elevation_between_flanking_gauges	Pass
	test_synthetic_standalone.py::	TestSynthetic::test_no_duplicate_synthetics	Pass
	test_synthetic_standalone.py::	TestSynthetic::test_synthetic_has_elevation	Pass
	test_synthetic_standalone.py::	TestSynthetic::test_synthetic_has_flood_stages	Pass
	test_synthetic_standalone.py::	TestSynthetic::test_synthetic_has_name	Pass
	test_synthetic_standalone.py::	TestSynthetic::test_synthetic_exists	Pass
	hazard_io_part1.py::TestLoadGauges	test_empty_gauge_list_raises	Pass
	hazard_io_part1.py::TestLoadGauges	test_file_not_found_raises	Pass
	hazard_io_part1.py::TestLoadGauges	test_gauge_has_id	Pass
	hazard_io_part1.py::TestLoadGauges	test_gauge_without_gauge_id_raises	Pass
	hazard_io_part1.py::TestLoadGauges	test_loads_gauges	Pass
	hazard_io_part1.py::TestLoadStorms	test_file_not_found_raises	Pass
	hazard_io_part1.py::TestLoadStorms	test_loads_list	Pass
	hazard_io_part1.py::TestLoadStorms	test_storm_has_id	Pass
	hazard_io_part2.py::TestSaveGauges	test_creates_gauge_file	Pass
	hazard_io_part2.py::TestSaveGauges	test_overwrites_existing_file	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	hazard_io_part2.py::TestSaveGaugesToCurves	test_multiple_gauges	Pass
	hazard_io_part2.py::TestSaveGaugesToCurves	test_saved_file_has_responses	Pass
	hazard_io_part2.py::TestSaveHazardCurves	test_creates_file	Pass
	hazard_io_part2.py::TestSaveHazardCurves	test_creates_parent_dirs	Pass
	hazard_io_part2.py::TestSaveHazardCurves	test_file_has_hazard_curves	Pass
	hazard_io_part2.py::TestSaveHazardCurves	test_file_has_metadata	Pass
	hazard_io_part2.py::TestSaveHazardCurves	test_metadata_merged	Pass
	business_days.py::TestBusinessDays	test_all_days_before_start	Pass
	business_days.py::TestBusinessDays	test_returns_ascending_order	Pass
	business_days.py::TestBusinessDays	test_returns_business_days_only	Pass
	business_days.py::TestBusinessDays	test_returns_correct_count	Pass
	business_days.py::TestBusinessDays	test_single_day	Pass
	business_days.py::TestBusinessDays	test_skips_weekends	Pass
	business_days.py::TestBusinessDays	test_zero_days	Pass
	eod_series.py::TestGenerateHistoricalEODSeries	test_business_days_are_weekdays	Pass
	eod_series.py::TestGenerateHistoricalEODSeries	test_cleans_existing_eod_files	Pass
	eod_series.pyTestGenerateHistoricalEODSeries	test_creates_eod_dir	Pass
	eod_series.pyTestGenerateHistoricalEODSeries	test_empty_trades_returns_zero	Pass
	eod_series.pyTestGenerateHistoricalEODSeries	test_enrichment_failure_is_skipped	Pass
	eod_series.pyTestGenerateHistoricalEODSeries	test_multiple_trades_adds_more_snapshots	Pass
	eod_series.pyTestGenerateHistoricalEODSeries	test_returns_integer	Pass
	eod_series.pyTestGenerateHistoricalEODSeries	test_seed_is_reproducible	Pass
	eod_series.pyTestGenerateHistoricalEODSeries	test_with_one_trade_generates_snapshots	Pass
	eod_series.pyTestGenerateHistoricalEODSeries	test_with_one_trade_writes_eod_files	Pass
	eod_series.pyTestGenerateHistoricalEODSeries	test_with_one_trade_writes_history_files	Pass
	hazard_history.pyTestGenerateHistoricalHazard	test_creates_output_file	Pass
	hazard_history.pyTestGenerateHistoricalHazard	test_history_entries_have_date	Pass
	hazard_history.pyTestGenerateHistoricalHazard	test_metadata_lists_gauges	Pass
	hazard_history.pyTestGenerateHistoricalHazard	test_multiple_gauges	Pass
	hazard_history.pyTestGenerateHistoricalHazard	test_no_curve_files_returns_early	Pass
	hazard_history.pyTestGenerateHistoricalHazard	test_output_has_history	Pass
	hazard_history.pyTestGenerateHistoricalHazard	test_output_has_metadata	Pass
	hazard_history.pyTestGenerateHistoricalHazard	test_curves_stored_numerically	Pass
	hazard_history.pyTestGenerateHistoricalHazard	test_two_curve_files_two_history_entries	Pass
	pnl_history.pyTestGenerateTradeHistoricalPnl	test_creates_output_file	Pass
	pnl_history.pyTestGenerateTradeHistoricalPnl	test_history_entry_has_required_keys	Pass
	pnl_history.pyTestGenerateTradeHistoricalPnl	test_multiple_eods_accumulate	Pass
	pnl_history.pyTestGenerateTradeHistoricalPnl	test_multiple_trades_in_same_eod	Pass
	pnl_history.pyTestGenerateTradeHistoricalPnl	test_no_eod_files_returns_early	Pass
	pnl_history.pyTestGenerateTradeHistoricalPnl	test_snapshots_in_metadata	Pass
	pnl_history.pyTestGenerateTradeHistoricalPnl	test_output_has_metadata	Pass
	pnl_history.pyTestGenerateTradeHistoricalPnl	test_output_has_trades	Pass
	pnl_history.pyTestGenerateTradeHistoricalPnl	test_position_without_swap_id_skipped	Pass
	pnl_history.pyTestGenerateTradeHistoricalPnl	test_trade_has_history	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	pnl_history.py::TestGenerateTradeline	PnlHistory_metadata_preserved	Pass
	test_mortgage_builder.py::TestBuildSection	test_empty_schema_returns_empty_dict	Pass
	test_mortgage_builder.py::TestBuildSection	test_nested_dict_without_type_recurses	Pass
	test_mortgage_builder.py::TestBuildSection	test_non_dict_schema_returns_empty_dict	Pass
	test_mortgage_builder.py::TestBuildSection	test_none_schema_returns_empty_dict	Pass
	test_mortgage_builder.py::TestBuildSection	test_none_value_from_generate_field_value	excluded
	test_mortgage_builder.py::TestBuildSection	test_reserved_key_description_skipped	Pass
	test_mortgage_builder.py::TestBuildSection	test_reserved_key_options_skipped	Pass
	test_mortgage_builder.py::TestBuildSection	test_reserved_key_type_skipped	Pass
	test_mortgage_builder.py::TestBuildSection	test_reserved_key_values_skipped	Pass
	test_mortgage_builder.py::TestQualityCheck	test_current_exceeds_original_is_capped_to_95pct	Pass
	test_mortgage_builder.py::TestQualityCheck	test_current_within_original_unchanged	Pass
	test_mortgage_builder.py::TestQualityCheck	test_delegates_to_random_module_when_method_present	Pass
	test_mortgage_builder.py::TestQualityCheck	test_over_100_set_to_95	Pass
	test_mortgage_builder.py::TestQualityCheck	test_under_10_set_to_60	Pass
	test_mortgage_builder.py::TestQualityCheck	test_valid_fraction_unchanged	Pass
	test_mortgage_builder.py::TestSetSpecificMortgageValue	test_existing_mortgage_dict_updated	Pass
	test_mortgage_builder.py::TestSetSpecificMortgageValue	test_initialises_header_key_when_absent	Pass
	test_mortgage_builder.py::TestSetSpecificMortgageValue	test_initialises_mortgage_key_when_absent	Pass
	test_mortgage_builder.py::TestSetSpecificMortgageValue	test_missing_property_id_key_defaults_empty_string	Pass
	test_mortgage_builder.py::TestSetSpecificMortgageValue	test_set_mortgage_id	Pass
	test_mortgage_builder.py::TestSetSpecificMortgageValue	test_set_mortgage_valid_fields	Pass
	test_mortgage_builder.py::TestSetSpecificMortgageValue	test_set_mortgage_valid	Pass
	test_mortgage_builder.py::TestSetSpecificMortgageValue	test_set_property_id_in_header	Pass
	test_mortgage_coverage.py::TestSaveFailure	test_save_error_is_reraised	Pass
	test_mortgage_coverage.py::TestSingleMortgageFailure	test_all_mortgages_fail_returns_empty_list	Pass
	test_mortgage_coverage.py::TestSingleMortgageFailure	test_failed_mortgage_is_counted_and_skipped	Pass
	test_mortgage_encoder_extractor.py::TestDateTimeEncoder	test_datetime_encoded_as_isoformat	Pass
	test_mortgage_encoder_extractor.py::TestDecimalEncoder	test_decimal_encoded	Pass
	test_mortgage_encoder_extractor.py::TestNumpyFloat32Encoder	test_numpy_float32_encoded	Pass
	test_mortgage_encoder_extractor.py::TestNumpyFloat64Encoder	test_numpy_float64_encoded_as_float	Pass
	test_mortgage_encoder_extractor.py::TestNumpyInt32Encoder	test_numpy_int32_encoded	Pass
	test_mortgage_encoder_extractor.py::TestNumpyInt64Encoder	test_numpy_int64_encoded_as_int	Pass
	test_mortgage_encoder_extractor.py::TestNumpyNdarrayEncoder	test_numpy_ndarray_encoded_as_list	Pass
	test_mortgage_encoder_extractor.py::TestUnknownTypeEncoder	test_unknown_type_raises_type_error	Pass
	test_mortgage_encoder_extractor.py::TestExtractPropertyInfo	test_extract_property_info	Pass
	test_mortgage_encoder_extractor.py::TestExtractPropertyInfo	test_flood_risk_extracted	Pass
	test_mortgage_encoder_extractor.py::TestExtractPropertyInfo	test_extract_property_info	Pass
	test_mortgage_encoder_extractor.py::TestMissingPropertyInfo	test_missing_property_defaults_construction_year	Pass
	test_mortgage_encoder_extractor.py::TestMissingPropertyInfo	test_missing_header_defaults_property_id	Pass
	test_mortgage_encoder_extractor.py::TestMissingPropertyInfo	test_missing_property_defaults_postcode_and_coords	Pass
	test_mortgage_encoder_extractor.py::TestMissingPropertyInfo	test_missing_property_header_returns_defaults	Pass
	test_mortgage_encoder_extractor.py::TestMissingPropertyInfo	test_missing_property_defaults_flood_risk	Pass
	test_mortgage_encoder_extractor.py::TestMissingPropertyInfo	test_missing_property_defaults_property_value	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_mortgage_encoder_extractor.py	TestExtractPropertyInfo	Pass
	test_mortgage_encoder_extractor.py	TestPropertyID::test_property_id_extracted_from_header	Pass
	test_mortgage_encoder_extractor.py	TestPropertyType::test_property_type_extracted	Pass
	test_mortgage_encoder_extractor.py	TestPropertyValue::test_property_value_extracted	Pass
	test_mortgage_generate.py	TestGenerateEmpty::test_empty_property_list_returns_zero_mortgages	Pass
	test_mortgage_generate.py	TestGenerateMissingProp::test_missing_property_file_raises_file_not_found	Pass
	test_mortgage_generate.py	TestGenerateOutputBalance::test_current_balance_does_not_exceed_original_loan	Pass
	test_mortgage_generate.py	TestGenerateOutputLoan::test_loan_details_values_are_positive	Pass
	test_mortgage_generate.py	TestGenerateOutputLtv::test_ltv_within_valid_range	Pass
	test_mortgage_generate.py	TestGenerateOutputMortgage::test_mortgage_data_key_is_list	Pass
	test_mortgage_generate.py	TestGenerateOutputMortgage::test_mortgage_data_structure_has_mortgage_key	Pass
	test_mortgage_generate.py	TestGenerateOutputMortgage::test_mortgage_header_has_required_fields	Pass
	test_mortgage_generate.py	TestGenerateOutputMortgage::test_mortgage_ids_key_is_list	Pass
	test_mortgage_generate.py	TestGenerateOutputFile::test_output_file_is_valid_json	Pass
	test_mortgage_generate.py	TestGenerateOutputJson::test_output_json_contains_generation_metadata	Pass
	test_mortgage_generate.py	TestGenerateOutputJson::test_output_json_contains_mortgages_key	Pass
	test_mortgage_generate.py	TestGenerateOutputStats::test_processing_stats_contain_timing	Pass
	test_mortgage_generate.py	TestGenerateNoOutputFile::test_generate_without_output_property_file	Pass
	test_mortgage_generate.py	TestGenerateMortgagesFromProperties::test_mortgages_from_property_file_exists	Pass
	test_mortgage_generate.py	TestLogLevel::test_debug_level_does_not_raise	Pass
	test_mortgage_generate.py	TestLogLevel::test_error_level_does_not_raise	Pass
	test_mortgage_generate.py	TestLogLevel::test_info_level_does_not_raise	Pass
	test_mortgage_generate.py	TestLogLevel::test_success_level_falls_back_gracefully	Pass
	test_mortgage_generate.py	TestLogLevel::test_unknown_level_falls_back_to_info	Pass
	test_mortgage_generate.py	TestLogLevel::test_warning_level_does_not_raise	Pass
	test_mortgage_random_fields_partial.py	TestGenerateDateValue::test_application_date_is_string	Pass
	test_mortgage_random_fields_partial.py	TestGenerateDateValue::test_default_date_not_defaulted	Pass
	test_mortgage_random_fields_partial.py	TestGenerateDateValue::test_default_date_when_defaulted	Pass
	test_mortgage_random_fields_partial.py	TestGenerateDateValue::test_good_date	Pass
	test_mortgage_random_fields_partial.py	TestGenerateDateValue::test_missing_date	Pass
	test_mortgage_random_fields_partial.py	TestGenerateDateValue::test_unknown_field	Pass
	test_mortgage_random_fields_partial.py	TestGenerateIntegerValue::test_arrears_months_converted	Pass
	test_mortgage_random_fields_partial.py	TestGenerateIntegerValue::test_arrears_months_in_years	Pass
	test_mortgage_random_fields_partial.py	TestGenerateIntegerValue::test_borrower_age_extreme	Pass
	test_mortgage_random_fields_partial.py	TestGenerateIntegerValue::test_borrower_age_int	Pass
	test_mortgage_random_fields_partial.py	TestGenerateIntegerValue::test_borrower_age_lexpensive	Pass
	test_mortgage_random_fields_partial.py	TestGenerateIntegerValue::test_borrower_age_normal	Pass
	test_mortgage_random_fields_partial.py	TestGenerateIntegerValue::test_credit_score_expensive_property	Pass
	test_mortgage_random_fields_partial.py	TestGenerateIntegerValue::test_credit_score_low_risk	Pass
	test_mortgage_random_fields_partial.py	TestGenerateIntegerValue::test_credit_score_normal	Pass
	test_mortgage_random_fields_partial.py	TestGenerateIntegerValue::test_days_in_arrears_converted	Pass
	test_mortgage_random_fields_partial.py	TestGenerateIntegerValue::test_days_in_arrears_default	Pass
	test_mortgage_random_fields_partial.py	TestGenerateIntegerValue::test_days_in_arrears_in_years	Pass
	test_mortgage_random_fields_partial.py	TestGenerateIntegerValue::test_family_members	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_mortgage_random_fields_part1.py	TestGenerateIntegerValue	Pass
	test_mortgage_random_fields_part1.py	TestGenerateIntegerValue	Pass
	test_mortgage_random_fields_part1.py	TestGenerateIntegerValue	Pass
	test_mortgage_random_fields_part1.py	TestGenerateIntegerValue	Pass
	test_mortgage_random_fields_part1.py	TestGenerateIntegerValue	Pass
	test_mortgage_random_fields_part1.py	TestGenerateIntegerValue	Pass
	test_mortgage_random_fields_part1.py	TestGenerateIntegerValue	Pass
	test_mortgage_random_fields_part1.py	TestGenerateIntegerValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateFieldValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateFieldValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateFieldValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateFieldValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateFieldValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateFieldValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateFieldValue_returns_string	Pass
	test_mortgage_random_fields_part2.py	TestGenerateFieldValue_returns_empty	Pass
	test_mortgage_random_fields_part2.py	TestGenerateFieldValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateFieldValue_returns_empty	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestGenerateTextValue	Pass
	test_mortgage_random_fields_part2.py	TestQualityConsistencyCheck_capped	Pass
	test_mortgage_random_fields_part2.py	TestQualityConsistencyCheck	Pass
	test_mortgage_random_fields_part2.py	TestQualityConsistencyCheck	Pass
	test_mortgage_random_fields_part2.py	TestQualityConsistencyCheck	Pass
	test_mortgage_random_fields_part2.py	TestQualityConsistencyCheck	Pass
	test_mortgage_random_fields_part2.py	TestQualityConsistencyCheck	Pass
	test_mortgage_random_financials.py	TestCalculateMortgageFinancials	Pass
	test_mortgage_random_financials.py	TestGenerateMortgageFinancials	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_mortgage_random_financials.py	TestLoanAmountPositive	Pass
	test_mortgage_random_financials.py	TestMonthlyPaymentPositive	Pass
	test_mortgage_random_financials.py	TestRequiredKeysPresent	Pass
	test_mortgage_random_financials.py	TestReturnsDict	Pass
	test_mortgage_random_financials.py	TestValidMortgageType	Pass
	test_mortgage_random_financials.py	TestEmptyPropertyInfoReturns_valid	Pass
	test_mortgage_random_financials.py	TestBuildingFamilyMortgageType	Partial
	test_mortgage_random_financials.py	TestDelinquencyHistoryMortgageType	Partial
	test_mortgage_random_financials.py	TestDelinquencyMortgageType	Partial
	test_mortgage_random_financials.py	TestVacantPropertyMortgageType	Pass
	test_mortgage_random_financials.py	TestDetermineOccupancyTypeDevelopment_or_primary	Pass
	test_mortgage_random_financials.py	TestHolidayHomeOccupancyTypeSecond_residence	Pass
	test_mortgage_random_financials.py	TestDetermineOccupancyTypeprimary_or_second	Pass
	test_mortgage_random_financials.py	TestSecondHomeOccupancyTypeSecond_residence	Pass
	test_mortgage_random_financials.py	TestEstimatedPropertyValue	Pass
	test_mortgage_random_financials.py	TestEstimatedPropertyValue	Pass
	test_mortgage_random_financials.py	TestEstimatedPropertyValue	Pass
	test_mortgage_random_financials.py	TestEstimatedPropertyValue	Pass
	test_mortgage_random_financials.py	TestEstimatedPropertyValue	Pass
	test_mortgage_random_financials.py	TestEstimatedPropertyValue	Pass
	test_mortgage_random_financials.py	TestWithSingleEstimatePropertyValue	Pass
	test_mortgage_random_financials.py	TestWithSingleEstimatePropertyValue	Pass
	test_mortgage_random_financials.py	TestZeroPropertyValueFallsThrough	Pass
	test_mortgage_random_financials.py	TestGeneratedValueWhenZero	Pass
	test_mortgage_random_financials.py	TestGeneratedRiskPassedThrough	Pass
	test_mortgage_random_financials.py	TestGeneratedWithMortgage_id	Pass
	test_mortgage_random_financials.py	TestGeneratedPropertyValue	Pass
	test_mortgage_random_values.py	TestAdvisedBoolean	Pass
	test_mortgage_random_values.py	TestCommercialPurpose	Pass
	test_mortgage_random_values.py	TestDefaultBoolean	Pass
	test_mortgage_random_values.py	TestExecutionOnly	Pass
	test_mortgage_random_values.py	TestFirstTimeBuyerAlwaysFalse	Pass
	test_mortgage_random_values.py	TestFirstTimeBuyerExpensive	Pass
	test_mortgage_random_values.py	TestFirstTimeBuyerNormal	Pass
	test_mortgage_random_values.py	TestGenerateBool	Pass
	test_mortgage_random_values.py	TestGenerateBool	Pass
	test_mortgage_random_values.py	TestGenerateBool	Pass
	test_mortgage_random_values.py	TestGenerateBoolReturns_bool	Pass
	test_mortgage_random_values.py	TestIsPropertyValuation	Pass
	test_mortgage_random_values.py	TestGenerateDecimal	Pass
	test_mortgage_random_values.py	TestGenerateDecimal	Pass
	test_mortgage_random_values.py	TestGenerateDecimal	Pass
	test_mortgage_random_values.py	TestGenerateDecimal	Pass
	test_mortgage_random_values.py	TestGenerateDecimal	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_mortgage_random_values.py	TestCurrentDtvDecimalValue	Pass
	test_mortgage_random_values.py	TestCurrentPaymentDecimalValue	Pass
	test_mortgage_random_values.py	TestDebtToIncomeRatioDecimalValue	Pass
	test_mortgage_random_values.py	TestDebtToIncomeRatioZeroIncome	Pass
	test_mortgage_random_values.py	TestDefaultFieldDecimalValue	Pass
	test_mortgage_random_values.py	TestInterestRateDecimalValue	Pass
	test_mortgage_random_values.py	TestLoanToValueRatioDecimalValue	Pass
	test_mortgage_random_values.py	TestOrigFieldDecimalValue	Pass
	test_mortgage_random_values.py	TestOriginalLendingRate	Pass
	test_mortgage_random_values.py	TestOriginalLoanDecimalValue	Pass
	test_mortgage_random_values.py	TestOriginalLoanAmount	Pass
	test_mortgage_random_values.py	TestOriginalLtvDecimalValue	Pass
	test_mortgage_random_values.py	TestOutstandingBalance	Pass
	test_mortgage_random_values.py	TestPurchaseValueDecimalValue	Pass
	test_mortgage_random_values.py	TestApplicationChannel	Pass
	test_mortgage_random_values.py	TestLoanPurposeVal	Pass
	test_mortgage_random_values.py	TestLoanPurposeResidential	Pass
	test_mortgage_random_values.py	TestMortgageTypeValue	Pass
	test_mortgage_random_values.py	TestOccupancyTypeValue	Pass
	test_mortgage_random_values.py	TestOriginalMrateType	Pass
	test_mortgage_random_values.py	TestPaymentFrequency	Pass
	test_mortgage_random_values.py	TestPropertyTypeVal	Pass
	test_mortgage_random_values.py	TestPropertyTypeResidential	Pass
	test_mortgage_random_values.py	TestUnknownFieldNoOptions	Pass
	test_mortgage_random_values.py	TestUnknownFieldWithOptions	Pass
	generators.py::TestCounterparty	test_all_counterparties_have_party_id	Pass
	generators.py::TestCounterparty	test_cdm_structure	Pass
	generators.py::TestCounterparty	test_custom_count	Pass
	generators.py::TestCounterparty	test_generates_counterparty_file	Pass
	generators.py::TestGaugeHistoricalDaily	test_generates_per_gauge_files	Pass
	generators.py::TestGaugeHistoricalDaily	test_output_has_daily_observations	Pass
	generators.py::TestGaugeHistoricalDaily	test_statistics_computed	Pass
	generators.py::TestGaugeTimeSeries	test_generates_per_gauge_files	Pass
	generators.py::TestGaugeTimeSeries	test_output_file_has_readings	Pass
	generators.py::TestGaugeTimeSeries	test_requires_gauge_portfolio	Pass
	generators.py::TestNostressFlag	test_explicit_stressm_flag_runs_despite_nostress	Pass
	generators.py::TestNostressFlag	test_nostress_flag_exists_in_parser	Pass
	generators.py::TestNostressFlag	test_nostress_skips_stressm_when_all	Pass
	generators.py::TestPropertyTimeSeries	test_generates_per_property_files	Pass
	generators.py::TestPropertyTimeSeries	test_per_property_files_have_flood_data	Pass
	generators.py::TestPropertyTimeSeries	test_portfolio_summary_has_statistics	Pass
	integration.py::TestDependencyChain	test_gaugehd_references_gauges	Pass
	integration.py::TestDependencyChain	test_gaugets_reads_gauge	Pass
	integration.py::TestDependencyChain	test_hazard_reads_gauge_and_storms	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	integration.py::TestDependencyChain	test_mortgage_reads_property	Pass
	integration.py::TestDependencyChain	test_propertyhc_references_properties	Pass
	integration.py::TestDependencyChain	test_propertyts_uses_nearest_gauges	Pass
	integration.py::TestEndToEndPipeline	test_full_pipeline_generates_all_outputs	Pass
	integration.py::TestEndToEndPipeline	test_gauge_count_propagates	Pass
	integration.py::TestEndToEndPipeline	test_pipeline_output_file_integrity	Pass
	integration.py::TestEndToEndPipeline	test_property_count_propagates	Pass
	property_energy.py::TestCalculateAnnualEnergy	test_larger_area_gives_larger_energy	Pass
	property_energy.py::TestCalculateAnnualEnergy	test_returns_positive_value	Pass
	property_energy.py::TestCalculateAnnualEnergy	test_uses_default_area_when_missing	Pass
	property_energy.py::TestCalculateCarbonEmissions	test_is_float	Pass
	property_energy.py::TestCalculateCarbonEmissions	test_larger_area_tends_to_give_larger_emissions	Pass
	property_energy.py::TestCalculateCarbonEmissions	test_returns_positive_value	Pass
	property_energy.py::TestCalculateCarbonEmissions	test_uses_default_area_when_missing	Pass
	property_energy.py::TestCalculateEnergyBill	test_energybill_calculation	Pass
	property_energy.py::TestCalculateEnergyBill	test_is_float	Pass
	property_energy.py::TestCalculateEnergyBill	test_returns_positive_value	Pass
	property_energy.py::TestCalculateEnergyBill	test_uses_defaults_when_missing	Pass
	property_energy.py::TestCalculateEnergyBill	test_zero_usage_returns_zero	Pass
	property_energy.py::TestCalculateGasUsage	test_less_than_remaining_energy	Pass
	property_energy.py::TestCalculateGasUsage	test_is_negative_when_electricity_exceeds_total	Pass
	property_energy.py::TestCalculateGasUsage	test_returns_non_negative	Pass
	property_energy.py::TestCalculateGasUsage	test_uses_defaults_when_missing	Pass
	property_energy.py::TestCalculateGridElectricity	test_fraction_of_total	Pass
	property_energy.py::TestCalculateGridElectricity	test_returns_positive_value	Pass
	property_energy.py::TestCalculateGridElectricity	test_uses_default_energy_when_missing	Pass
	property_energy.py::TestCalculateSolarCapacity	test_heat_pump_returns_zero	Pass
	property_energy.py::TestCalculateSolarCapacity	test_multiple_returns_positive	Pass
	property_energy.py::TestCalculateSolarCapacity	test_no_solar_returns_zero	Pass
	property_energy.py::TestCalculateSolarCapacity	test_no_system_returns_zero	Pass
	property_energy.py::TestCalculateSolarCapacity	test_solar_panel_returns_positive	Pass
	property_energy.py::TestCalculateSolarCapacity	test_solar_panels_uses_default_area_when_missing	Pass
	test_geometry.py::TestEnsureOffset	test_close_point_pushed_away	Pass
	test_geometry.py::TestEnsureOffset	test_far_point_unchanged	Pass
	test_geometry.py::TestEnsureOffset	test_returns_tuple_of_two_floats	Pass
	test_geometry.py::TestHaversine	test_approximate_one_degree_lat	Pass
	test_geometry.py::TestHaversine	test_approximate_one_degree_lon_at_equator	Pass
	test_geometry.py::TestHaversine	test_same_point_returns_zero	Pass
	test_geometry.py::TestHaversine	test_symmetry	Pass
	test_geometry.py::TestMinRiverDistance	test_distant_point_returns_larger_distance	Pass
	test_geometry.py::TestMinRiverDistance	test_point_on_gauge_returns_small_distance	Pass
	test_geometry.py::TestMinRiverDistance	test_returns_float	Pass
	test_geometry.py::TestNearestOnSegment	test_degenerate_segment_returns_start	Pass
	test_geometry.py::TestNearestOnSegment	test_distance_is_non_negative	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_geometry.py::TestNearestOnSegment	test_midpoint_projects_to_midpoint	Pass
	test_geometry.py::TestNearestOnSegment	test_point_before_start_clamps_to_start	Pass
	test_geometry.py::TestNearestOnSegment	test_point_past_end_clamps_to_end	Pass
	test_geometry.py::TestPerpendicularOffset	test_degenerate_segment_falls_back_to_pure_lat_offset	Pass
	test_geometry.py::TestPerpendicularOffset	test_offset_moves_point	Pass
	test_geometry.py::TestPerpendicularOffset	test_returns_two_floats	Pass
	test_location_coverage_b6.py::TestCumbriaBoundary	test_at_exactly_boundary_values	Pass
	test_location_coverage_b6.py::TestCumbriaBoundary	test_just_above_lower_value	Pass
	test_location_coverage_b6.py::TestCumbriaBoundary	test_just_below_lower_boundary	Pass
	test_location_coverage_b6.py::TestCumbriaBoundary	test_just_below_upper_boundary	Pass
	test_location_coverage_b6.py::TestCumbriaBoundary	test_just_below_upper_boundary	Pass
	test_location_coverage_b6.py::TestCottageB3bedroomHouse	test_cottage_3_bedroom_house_paths_1_or_2	Pass
	test_location_coverage_b6.py::TestCottageB4bedroomHouse	test_cottage_4_bedroom_house_paths_2_or_3	Pass
	test_location_coverage_b6.py::TestCottageB5bedroomHouse	test_cottage_5_bedroom_house_paths_2_or_3	Pass
	test_locations.py::TestGenerateLocations	test_each_location_has_required_keys	Pass
	test_locations.py::TestGenerateLocations	test_elevation_is_non_negative	Pass
	test_locations.py::TestGenerateLocations	test_fallback_used_when_fewer_than_3_gauge_points	Pass
	test_locations.py::TestGenerateLocations	test_fallback_used_when_no_gauge_points	Pass
	test_locations.py::TestGenerateLocations	test_primary_idx_zero_uses_gauges_012	Pass
	test_locations.py::TestGenerateLocations	test_reference_gauge_indices_sorted	Pass
	test_locations.py::TestGenerateLocations	test_reference_gauge_indices_three_elements	Pass
	test_locations.py::TestGenerateLocations	test_returns_correct_count	Pass
	test_locations.py::TestGenerateLocations	test_fallback_without_elevation_uses_random	Pass
	test_multi_storm_cluster.py::TestClusterSuper	test_super_max_depth_is_from_highest_storm	Pass
	test_multi_storm_cluster.py::TestClusterSuper	test_super_produces_three_flood_events	Pass
	test_multi_storm_cluster.py::TestClusterSuper	test_three_distinct_storm_ids_in_events	Pass
	test_multi_storm_cluster.py::TestMixedSequence	test_storms_collected_severe_counted	Pass
	test_multi_storm_cluster.py::TestMixedSequence	test_storms_not_in_flood_events	Pass
	test_multi_storm_cluster.py::TestPersistentSequence	test_all_events_have_readings	Pass
	test_multi_storm_cluster.py::TestPersistentSequence	test_four_events_recorded	Pass
	test_multi_storm_doublet.py::TestDoubletSequence	test_each_event_has_independent_damage	Pass
	test_multi_storm_doublet.py::TestDoubletSequence	test_events_have_distinct_storm_ids	Pass
	test_multi_storm_doublet.py::TestDoubletSequence	test_produces_two_flood_events	Pass
	test_multi_storm_doublet.py::TestDoubletSequence	test_summary_counts_all_doublet_floods	Pass
	test_multi_storm_portfolio.py::TestGenerateWithMultiStormSequence	test_generate_zero_floods	Pass
	test_multi_storm_portfolio.py::TestGenerateWithMultiStormSequence	test_summary_counts_all_sequence_storms	Pass
	test_multi_storm_portfolio.py::TestGenerateWithMultiStormSequence	test_summary_counts_for_clusters	Pass
	test_nearest_synthetic_priority.py::TestSyntheticAlwaysFirst	test_synthetic_falls_back_to_distance_order	Pass
	test_nearest_synthetic_priority.py::TestSyntheticAlwaysFirst	test_gauges_sorted_by_distance_after_synthetic	Pass
	test_nearest_synthetic_priority.py::TestSyntheticAlwaysFirst	test_synthetic_at_position_zero	Pass
	test_nearest_synthetic_priority.py::TestSyntheticAlwaysFirst	test_synthetic_always_first_even_when_far	Pass
	test_nearest_synthetic_priority.py::TestSyntheticAlwaysFirst	test_synthetic_gauge_info_preserved	Pass
	test_nearest_synthetic_priority.py::TestSyntheticAlwaysFirst	test_synthetic_always_returned	Pass
	test_property_random_coverage.py::TestGeneratorExceptionFallback	test_exception_falls_through_to_type	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_property_random_coverage.py::TestIntegerFallback	test_integer_fallback_returns_int	Pass
	test_property_random_coverage.py::TestNumberFallback	test_number_fallback_returns_float	Pass
	test_property_random_coverage.py::TestStringWithOptions	test_string_with_options_picks_from_list	Pass
	test_property_random_coverage.py::TestStringWithOptions	test_string_with_options_returns_empty	Pass
	test_property_values.py::TestGaugeMap	test_corrupted_gauge_json_does_not_crash	Pass
	test_property_values.py::TestGaugeMap	test_gauge_map_empty_when_no_gauge_json	Pass
	test_property_values.py::TestGaugeMap	test_gauge_map_loaded_when_gauge_json_present	Pass
	test_property_values.py::TestGaugeMap	test_gauge_map_works_with_gauges_key	Pass
	test_property_values.py::TestLogMethod	test_debug_does_not_raise	Pass
	test_property_values.py::TestLogMethod	test_error_does_not_raise	Pass
	test_property_values.py::TestLogMethod	test_info_does_not_raise	Pass
	test_property_values.py::TestLogMethod	test_success_falls_back_gracefully	Pass
	test_property_values.py::TestLogMethod	test_warning_does_not_raise	Pass
	test_property_values.py::TestProcessingStatsFromPropertiesJson	test_failed_properties_tracked	Pass
	test_property_values.py::TestPropertyOutputFileStructure	test_output_file_contains_generation_metadata	Pass
	test_property_values.py::TestPropertyOutputFileStructure	test_output_file_contains_properties_key	Pass
	test_property_values.py::TestPropertyOutputFileStructure	test_property_header_has_property_id	Pass
	test_property_values.py::TestPropertyOutputFileStructure	test_output_file_contains_locations	Pass
	test_property_values.py::TestSetSpecificPropertyValues	test_property_value_set_in_attrs	Pass
	test_property_values.py::TestSetSpecificPropertyValues	test_relative_to_ground_level	Pass
	test_property_values.py::TestSetSpecificPropertyValues	test_empty_reference_gauge_indices_no_ref_key	Pass
	test_property_values.py::TestSetSpecificPropertyValues	test_property_value_in_location	Pass
	test_property_values.py::TestSetSpecificPropertyValues	test_relative_value_in_location_does_not_set_ground_level	Pass
	test_property_values.py::TestSetSpecificPropertyValues	test_reference_gauge_indices_no_ref_key	Pass
	test_property_values.py::TestSetSpecificPropertyValues	test_property_value_set_in_attrs	Pass
	test_property_values.py::TestSetSpecificPropertyValues	test_reference_gauges_fallback_when_not_in_map	Pass
	test_property_values.py::TestSetSpecificPropertyValues	test_reference_gauges_set_from_gauge_id_map	Pass
	test_property_zone_consistency.py::TestBoundaryParametrized	test_EAZone_parametrized[0.0-Zone 3b]	Pass
	test_property_zone_consistency.py::TestBoundaryParametrized	test_EAZone_parametrized[0.25-Zone 3b]	Pass
	test_property_zone_consistency.py::TestBoundaryParametrized	test_EAZone_parametrized[0.5-Zone 3a]	Pass
	test_property_zone_consistency.py::TestBoundaryParametrized	test_EAZone_parametrized[1.0-Zone 3a]	Pass
	test_property_zone_consistency.py::TestBoundaryParametrized	test_EAZone_parametrized[1.5-Zone 2]	Pass
	test_property_zone_consistency.py::TestBoundaryParametrized	test_EAZone_parametrized[10.0-Zone 1]	Pass
	test_property_zone_consistency.py::TestBoundaryParametrized	test_EAZone_parametrized[2.5-Zone 2]	Pass
	test_property_zone_consistency.py::TestBoundaryParametrized	test_EAZone_parametrized[3.0-Zone 1]	Pass
	test_property_zone_consistency.py::TestFeetToZoneFromElevation	test_feet_to_zone_from_elevation_function	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_property_zone_consistency.py::TestEAZoneOffsetElevation	test_unknown_defaults_to_zone_1	Pass
	test_property_zone_consistency.py::TestZoneAt30mElevation	test_zone_1_at_30	Pass
	test_property_zone_consistency.py::TestZoneAt1mElevation	test_zone_1_at_large_offset	Pass
	test_property_zone_consistency.py::TestZoneAt15mElevation	test_zone_2_at_15	Pass
	test_property_zone_consistency.py::TestZoneAt29mElevation	test_zone_2_at_29	Pass
	test_property_zone_consistency.py::TestZoneAt05mElevation	test_zone_3a_at_05	Pass
	test_property_zone_consistency.py::TestZoneAt14mElevation	test_zone_3a_at_14	Pass
	test_property_zone_consistency.py::TestZoneAt04mElevation	test_zone_3b_at_04	Pass
	test_property_zone_consistency.py::TestZoneAt0mElevation	test_zone_3b_at_zero_offset	Pass
	test_property_zone_consistency.py::TestVertical	test_vertical_offset_location_dict	Pass
	test_property_zone_consistency.py::TestZoneElevationBoundsConfig	test_zone_elevations_have_finite_upper[Zone 2]	Pass
	test_property_zone_consistency.py::TestZoneElevationBoundsConfig	test_zone_elevations_have_finite_upper[Zone 3a]	Pass
	test_property_zone_consistency.py::TestZoneElevationBoundsConfig	test_zone_elevations_have_finite_upper[Zone 3b]	Pass
	test_property_zone_consistency.py::TestZoneElevationBoundsConfig	test_zone_elevation_bounds_consistency	Pass
	test_property_zone_consistency.py::TestZoneElevationBoundsConfig	test_zone_1_has_no_lower_bound	Pass
	test_property_zone_consistency.py::TestZoneElevationBoundsConfig	test_zone_1_lower_bound_non_negative[Zone 1]	Pass
	test_property_zone_consistency.py::TestZoneElevationBoundsConfig	test_zone_1_lower_bound_non_negative[Zone 2]	Pass
	test_property_zone_consistency.py::TestZoneElevationBoundsConfig	test_zone_1_lower_bound_non_negative[Zone 3a]	Pass
	test_property_zone_consistency.py::TestZoneElevationBoundsConfig	test_zone_1_lower_bound_non_negative[Zone 3b]	Pass
	test_property_zone_consistency.py::TestZoneElevationBoundsConfig	test_zone_1_lower_bounds_monotonically_increasing	Pass
	test_property_zone_consistency.py::TestZoneElevationBoundsConfig	test_zone_1_has_no_upper_bound	Pass
	test_property_zone_consistency.py::TestZoneElevationBoundsConfig	test_zone_3b_starts_at_zero	Pass
	test_propertyhc_generate.py::TestGetAvgAgreement	test_avg_agreement_is_flat	Pass
	test_propertyhc_generate.py::TestGetAvgAgreement	test_avg_agreement_rate_between_0_and_1	Pass
	test_propertyhc_generate.py::TestGetAggregateStats	test_aggregate_stats	Pass
	test_propertyhc_generate.py::TestGetAggregateStats	test_aggregate_events_counted	Pass
	test_propertyhc_generate.py::TestLogMethod	test_log_methods_not_raise	Pass
	test_propertyhc_generate.py::TestLogMethod	test_log_with_empty_string	Pass
	test_propertyhc_generate.py::TestPrintSummary	test_prints_severe_threshold	Pass
	test_propertyhc_generate.py::TestPrintSummary	test_prints_spread_non_negative	Pass
	test_propertyhc_generate.py::TestPrintSummary	test_structure_is_flat	Pass
	test_propertyhc_helpers.py::TestGetCategoryCounts	test_ignores_other_keys	Pass
	test_propertyhc_helpers.py::TestGetCategoryCounts	test_returns_event_counts	Pass
	test_propertyhc_helpers.py::TestGetCategoryCounts	test_missing_keys_defaults_to_zero	Pass
	test_propertyhc_helpers.py::TestGetCategoryCounts	test_negative_severe_event_count	Pass
	test_propertyhc_helpers.py::TestGetCategoryCounts	test_zero_severe_returns_zero	Pass
	test_propertyhc_helpers.py::TestGetPrior	test_prior_none_when_quantlib_unavailable	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_propertyhc_helpers.py::TestLoadDefaults	test_datetime_encoded_as_isoformat	Pass
	test_propertyhc_helpers.py::TestLoadDefaults	test_empty_float32_encoded	Pass
	test_propertyhc_helpers.py::TestLoadDefaults	test_empty_float_encoded	Pass
	test_propertyhc_helpers.py::TestLoadDefaults	test_empty_int32_encoded	Pass
	test_propertyhc_helpers.py::TestLoadDefaults	test_empty_int_encoded	Pass
	test_propertyhc_helpers.py::TestLoadDefaults	test_empty_ndarray_encoded_as_list	Pass
	test_propertyhc_helpers.py::TestLoadDefaults	test_unknown_type_raises_type_error	Pass
	test_propertyhc_helpers.py::TestLoadGaugeHazardCurve	test_missing_gauge_returns_empty_and_default_storms	Pass
	test_propertyhc_helpers.py::TestLoadGaugeHazardCurve	test_present_gauge_returns_curves	Pass
	test_propertyhc_helpers.py::TestLoadGaugeHazardCurve	test_present_gauge_returns_num_storms	Pass
	test_propertyhc_process.py::TestDepthThresholds	test_annual_probability_equals_count_over_n_storms	Pass
	test_propertyhc_process.py::TestDepthThresholds	test_only_severe_key_in_depth_thresholds	Pass
	test_propertyhc_process.py::TestDepthThresholds	test_return_period_none_when_zero_floods	Pass
	test_propertyhc_process.py::TestDepthThresholds	test_return_period_positive_when_floods_present	Pass
	test_propertyhc_process.py::TestIdwGaugeSpread	test_single_gauge_spreads_computed_with_two_gauges	Pass
	test_propertyhc_process.py::TestIdwGaugeSpread	test_two_gauges_idw_returns_dict	Pass
	test_propertyhc_process.py::TestProcessPropertyEvents	test_flood_events_counts_event_count	Pass
	test_propertyhc_process.py::TestProcessPropertyEvents	test_many_events_uses_event_count	Pass
	test_propertyhc_process.py::TestProcessPropertyEvents	test_spread_equals_flood_count_over_storms	Pass
	test_propertyhc_process.py::TestProcessPropertyEvents	test_non_flooded_events_excluded_from_depth_count	Pass
	test_propertyhc_process.py::TestProcessPropertyEvents	test_zero_flooded_events_uses_event_count_pricing	Pass
	test_propertyhc_process.py::TestProcessPropertyEvents	test_max_depth_positive_for_flooded_property	Pass
	test_propertyhc_process.py::TestProcessPropertyEvents	test_property_summary_when_no_gauge_floods	Pass
	test_propertyhc_process.py::TestProcessPropertyEvents	test_zero_depths_for_non_flooded	Pass
	test_propertyhc_storm_details.py::TestGaugeThresholds	test_empty_when_no_gauge_info	Pass
	test_propertyhc_storm_details.py::TestGaugeThresholds	test_has_alert_level	Pass
	test_propertyhc_storm_details.py::TestGaugeThresholds	test_has_severe_level	Pass
	test_propertyhc_storm_details.py::TestGaugeThresholds	test_has_warning_level	Pass
	test_propertyhc_storm_details.py::TestGaugeThresholds	test_key_exists	Pass
	test_propertyhc_storm_details.py::TestStormDetailsFromEvents	test_consistent_flood_events	Pass
	test_propertyhc_storm_details.py::TestEachStormDetailsField	test_has_damage_ratio	Pass
	test_propertyhc_storm_details.py::TestEachStormDetailsField	test_has_exceeded_severe	Pass
	test_propertyhc_storm_details.py::TestEachStormDetailsField	test_has_flood_depth_m	Pass
	test_propertyhc_storm_details.py::TestEachStormDetailsField	test_has_flooded_flag	Pass
	test_propertyhc_storm_details.py::TestEachStormDetailsField	test_has_gauge_peak_m	Pass
	test_propertyhc_storm_details.py::TestEachStormDetailsField	test_has_retention_factor	Pass
	test_propertyhc_storm_details.py::TestEachStormDetailsField	test_has_storm_id	Pass
	test_propertyhc_storm_details.py::TestFloodedStormDetailsFields	test_have_positive_depth	Pass
	test_propertyhc_storm_details.py::TestNonFloodedStormDetailsFields	test_have_zero_depth	Pass
	test_propertyhc_storm_details.py::TestStormDetailsPresent	test_empty_when_no_floods	Pass
	test_propertyhc_storm_details.py::TestStormDetailsPresent	test_includes_non_flooded	Pass
	test_propertyhc_storm_details.py::TestStormDetailsPresent	test_key_exists	Pass
	test_propertyhc_storm_details.py::TestStormDetailsPresent	test_length_matches_flood_events	Pass
	test_propertyshd_generate.py::TestPropertyShdGenerate	test_shd_has_structure	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_propertyshd_generate.py::TestPropertyShdGenerate	test_shd_output_file_is_propertyshd_json	Pass
	test_propertyshd_generate.py::TestPropertyShdGenerate	test_shd_reads_from_propertytsd	Pass
	test_propertyshd_generate.py::TestPropertyShdGenerate	test_shd_has_correct_structure	Pass
	test_propertyshd_generate.py::TestPropertyShdGenerate	test_shd_output_file_is_propertyshe_json	Pass
	test_propertyshd_generate.py::TestPropertyShdGenerate	test_shd_reads_from_propertytse	Pass
	test_propertyts_compound.py::TestBuildCompound	test_compound_flooded_event_has_readings	Pass
	test_propertyts_compound.py::TestBuildCompound	test_compound_not_flooded_no_readings	Pass
	test_propertyts_compound.py::TestBuildCompound	test_compound_sequence_type	Pass
	test_propertyts_compound.py::TestBuildCompound	test_compound_event_keys	Pass
	test_propertyts_compound.py::TestBuildCompound	test_component_flood_cap_applied	Pass
	test_propertyts_compound.py::TestBuildCompound	test_zero_distance_travel_time	Pass
	test_propertyts_compound.py::TestComputePeakFlowDispatches	test_compute_peak_flow_dispatches_to_simple	Pass
	test_propertyts_compound.py::TestComputePeakFlowDispatches	test_compute_peak_flow_dispatches_to_compound	Pass
	test_propertyts_compound.py::TestDepthDamageDelegate	test_delegates_to_delegate_depth_damage	Pass
	test_propertyts_compound.py::TestZeroDepthDamageDelegate	test_zero_depth	Pass
	test_propertyts_encoder.py::TestDateTimeEncoderPropertyts	test_datetime_encoding	Pass
	test_propertyts_encoder.py::TestTimedeltaEncoderPropertyts	test_timedelta_encoding	Pass
	test_propertyts_encoder.py::TestTimedeltaEncoderPropertyts	test_timedelta_integer	Pass
	test_propertyts_encoder.py::TestTimedeltaEncoderPropertyts	test_timedelta_array	Pass
	test_propertyts_encoder.py::TestTimedeltaEncoderPropertyts	test_unknown_raises	Pass
	test_propertyts_encoder.py::TestDepthDamage	test_depth_flood_near_max_damage	Pass
	test_propertyts_encoder.py::TestDepthDamage	test_positive_depth_returns_positive	Pass
	test_propertyts_encoder.py::TestDepthDamage	test_zero_depth_returns_zero	Pass
	test_propertyts_encoder.py::TestProcessPropertyEarlyExits	test_empty_gauge_returns_none	Pass
	test_propertyts_encoder.py::TestProcessPropertyEarlyExits	test_missing_prop_returns_none	Pass
	test_propertyts_encoder.py::TestProcessPropertyEarlyExits	test_zero_peak_returns_none	Pass
	test_propertyts_flood.py::TestBuildFlood	test_arrival_and_peak_times_set_when_flooded	Pass
	test_propertyts_flood.py::TestBuildFlood	test_flooded_when_wse_exceeds_threshold	Pass
	test_propertyts_flood.py::TestBuildFlood	test_no_gauge_readings_empty_hydrograph	Pass
	test_propertyts_flood.py::TestBuildFlood	test_not_flooded_when_wse_below_threshold	Pass
	test_propertyts_flood.py::TestBuildFlood	test_returns_expected_keys	Pass
	test_propertyts_flood.py::TestBuildFlood	test_zero_distance_travel_time	Pass
	test_propertyts_flood.py::TestComputeDistance	test_distances_less_than_1m_short_circuits	Pass
	test_propertyts_flood.py::TestComputeDistance	test_multiple_gauges_uses_idw	Pass
	test_propertyts_flood.py::TestComputeDistance	test_no_matching_gauge_returns_none	Pass
	test_propertyts_flood.py::TestComputeDistance	test_returns_distance_with_storm_id	Pass
	test_propertyts_flood.py::TestProcessPropertyWithFlood	test_high_elevation_property_does_not_flood	Pass
	test_propertyts_flood.py::TestProcessPropertyWithFlood	test_no_storms_zero_floods	Pass
	test_propertyts_flood.py::TestProcessPropertyWithFlood	test_summary_has_required_keys	Pass
	test_propertyts_flood.py::TestProcessPropertyWithFlood	test_writes_propertyts_file	Pass
	test_propertyts_generate.py::TestGenerateRatio	test_logging_includes_one_at_50	Pass
	test_propertyts_generate.py::TestGenerateRatio	test_ratio_computed_when_storms_present	Pass
	test_propertyts_generate.py::TestGenerateRatio	test_ratio_zero_when_no_gauge_storms	Pass
	test_propertytsd.py::TestShdFloodMix	test_shd_sets_effective_elevation_to_gauge	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_propertytsd.py::TestShdFloodMixin	test_sine_sets_distance_to_zero	Pass
	test_propertytsd.py::TestShdOutputDirectory	test_dir_writes_to_propertytsd_dir	Pass
	test_propertytsd.py::TestShdOutputDirectory	test_dir_writes_to_propertytse_dir	Pass
	test_zone_floor_pricing.py::TestEventCountPricing	test_100_floods_gives_50bp	Pass
	test_zone_floor_pricing.py::TestEventCountPricing	test_664_floods_gives_332bp	Pass
	test_zone_floor_pricing.py::TestEventCountPricing	test_spread_scales_with_num_storms	Pass
	test_zone_floor_pricing.py::TestEventCountPricing	test_zero_floods_gives_zero_spread	Pass
	test_zone_floor_pricing.py::TestEventCountPricing	test_zone_floor_still_valid	Pass
	test_zone_floor_pricing.py::TestSyntheticBasisPricing	test_no_basis_returns_none	Pass
	test_zone_floor_pricing.py::TestSyntheticBasisPricing	test_synthetic_prefix_detection	Pass
	test_zone_from_synthetic.py::TestZoneFromSyntheticGauge	test_inside_zone_3b_with_elevation	Pass
	test_zone_from_synthetic.py::TestZoneFromSyntheticGauge	test_inside_zone_3b_preserves_real_offset	Pass
	test_zone_from_synthetic.py::TestZoneFromSyntheticGauge	test_zone_uses_synthetic_gauge_elevation	Pass
	test_zone_from_synthetic.py::TestZoneFromOffset	test_boundary_offset[0.0-Zone 3b]	Pass
	test_zone_from_synthetic.py::TestZoneFromOffset	test_boundary_offset[0.25-Zone 3b]	Pass
	test_zone_from_synthetic.py::TestZoneFromOffset	test_boundary_offset[0.49-Zone 3b]	Pass
	test_zone_from_synthetic.py::TestZoneFromOffset	test_boundary_offset[0.5-Zone 3a]	Pass
	test_zone_from_synthetic.py::TestZoneFromOffset	test_boundary_offset[1.0-Zone 3a]	Pass
	test_zone_from_synthetic.py::TestZoneFromOffset	test_boundary_offset[1.49-Zone 3a]	Pass
	test_zone_from_synthetic.py::TestZoneFromOffset	test_boundary_offset[1.5-Zone 2]	Pass
	test_zone_from_synthetic.py::TestZoneFromOffset	test_boundary_offset[2.5-Zone 2]	Pass
	test_zone_from_synthetic.py::TestZoneFromOffset	test_boundary_offset[2.99-Zone 2]	Pass
	test_zone_from_synthetic.py::TestZoneFromOffset	test_boundary_offset[3.0-Zone 1]	Pass
	test_zone_from_synthetic.py::TestZoneFromOffset	test_boundary_offset[5.0-Zone 1]	Pass
	test_zone_from_synthetic.py::TestZoneFromOffset	test_boundary_offset[50.0-Zone 1]	Pass
	test_zone_from_synthetic.py::TestZoneFromOffset	test_negative_offset_gives_zone_3b	Pass
	terrain.py::TestFloodZoneInOutput	test_flood_zone_default_when_missing	Pass
	terrain.py::TestFloodZoneInOutput	test_flood_zone_is_valid_zone_name	Pass
	terrain.py::TestFloodZoneInOutput	test_flood_zone_present_in_result	Pass
	terrain.py::TestFloodZoneInOutput	test_flood_zone_values_correct	Pass
	terrain.py::TestTerrainGridInMetadata	test_terrain_grid_has_all_zones	Pass
	terrain.py::TestTerrainGridInMetadata	test_terrain_grid_has_distances	Pass
	terrain.py::TestTerrainGridInMetadata	test_terrain_grid_has_elevations	Pass
	terrain.py::TestTerrainGridInMetadata	test_terrain_grid_present	Pass
	terrain.py::TestTerrainGridInMetadata	test_terrain_grid_zone_has_grid_array	Pass
	batch_generator_part1.py::TestGenerateEventSet	test_all_event_sequences	Pass
	batch_generator_part1.py::TestGenerateEventSet	test_event_set_propagated	Pass
	batch_generator_part1.py::TestGenerateEventSet	test_correct_count	Pass
	batch_generator_part1.py::TestGenerateEventSet	test_different_seeds_differ	Pass
	batch_generator_part1.py::TestGenerateEventSet	test_event_sequence_type_doublet	Pass
	batch_generator_part1.py::TestGenerateEventSet	test_event_sequence_type_isolated	Pass
	batch_generator_part1.py::TestGenerateEventSet	test_reproducible_with_same_seed	Pass
	batch_generator_part1.py::TestGenerateEventSet	test_return_list	Pass
	batch_generator_part1.py::TestGenerateEventSet	test_weights_change_distribution	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	batch_generator_part1.py::TestGetEventStyleDistribution	test_event_style_distributions_present	Pass
	batch_generator_part1.py::TestGetSequenceTypeDistribution	test_sequence_type_distribution	Pass
	batch_generator_part1.py::TestValidation	test_all_sequences_valid	Pass
	batch_generator_part1.py::TestValidation	test_medium_batch_all_valid	Pass
	batch_generator_part1.py::TestValidation	test_no_negative_durations	Pass
	batch_generator_part1.py::TestValidation	test_no_negative_precipitation	Pass
	batch_generator_part1.py::TestValidation	test_timing_constraint	Pass
	batch_generator_part2.py::TestSaveAndLoadRoundTrip	test_creates_parent_dirs	Pass
	batch_generator_part2.py::TestSaveAndLoadRoundTrip	test_load_missing_file_raises	Pass
	batch_generator_part2.py::TestSaveAndLoadRoundTrip	test_load_returns_correct_count	Pass
	batch_generator_part2.py::TestSaveAndLoadRoundTrip	test_load_wrong_schema_raises	Pass
	batch_generator_part2.py::TestSaveAndLoadRoundTrip	test_loaded_sequences_still_valid	Pass
	batch_generator_part2.py::TestSaveAndLoadRoundTrip	test_round_trip_num_storms	Pass
	batch_generator_part2.py::TestSaveAndLoadRoundTrip	test_round_trip_precipitation	Pass
	batch_generator_part2.py::TestSaveAndLoadRoundTrip	test_round_trip_sequence_type	Pass
	batch_generator_part2.py::TestSaveAndLoadRoundTrip	test_round_trip_storm_fields	Pass
	batch_generator_part2.py::TestSaveAndLoadRoundTrip	test_round_trip_total_duration	Pass
	batch_generator_part2.py::TestSaveAndLoadRoundTrip	test_save_creates_file	Pass
	batch_generator_part2.py::TestSaveAndLoadRoundTrip	test_schema_version_in_file	Pass
	batch_generator_part2.py::TestSummary	test_summary_creates_file	Pass
	batch_generator_part2.py::TestSummary	test_summary_duration_stats_present	Pass
	batch_generator_part2.py::TestSummary	test_summary_num_sequences_matches	Pass
	batch_generator_part2.py::TestSummary	test_summary_precipitation_stats_present	Pass
	batch_generator_part2.py::TestSummary	test_summary_schema_version	Pass
	batch_generator_part2.py::TestSummary	test_summary_type_counts_sum_to_total	Pass
	batch_generator_part2.py::TestSummary	test_summary_type_fractions_sum_to_one	Pass
	storm_intensity.py::TestGetStormIntensity	test_cluster_returns_3_or_4	Pass
	storm_intensity.py::TestGetStormIntensity	test_doublet_returns_2	Pass
	storm_intensity.py::TestGetStormIntensity	test_isolated_returns_1	Pass
	storm_intensity.py::TestGetStormIntensity	test_persistent_returns_4_or_5	Pass
	storm_intensity.py::TestSampleSequenceType	test_catastrophic_tends_toward_cluster_persistent	Pass
	storm_intensity.py::TestSampleSequenceType	test_defined_categories[catastrophic]	Pass
	storm_intensity.py::TestSampleSequenceType	test_defined_categories[extreme]	Pass
	storm_intensity.py::TestSampleSequenceType	test_defined_categories[moderate]	Pass
	storm_intensity.py::TestSampleSequenceType	test_defined_categories[severe]	Pass
	storm_intensity.py::TestSampleSequenceType	test_returns_sequence_type	Pass
	storm_intensity.py::TestSampleSequenceType	test_unknown_category_uses_default_weights	Pass
	storm_intensity.py::TestSampleStormIntensity	test_decreasing_path_subsequent_storm	Pass
	storm_intensity.py::TestSampleStormIntensity	test_first_storm_downward_variation	Pass
	storm_intensity.py::TestSampleStormIntensity	test_first_storm_returns_positive	Pass
	storm_intensity.py::TestSampleStormIntensity	test_intensity_within_20_percent_range	Pass
	storm_intensity.py::TestSampleStormIntensity	test_subsequent_storm_mostly_up	Pass
	storm_intensity.py::TestSampleStormIntensity	test_subsequent_storm_positive	Pass
	storm_intensity.py::TestSampleStormIntensity	test_very_low_base_clamped_to_01	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	storm_intensity.py::TestShouldGenerate	test_defined_categories	Pass
	storm_intensity.py::TestShouldGenerate	test_catastrophic_high_probability	Pass
	storm_intensity.py::TestShouldGenerate	test_minor_low_probability	Pass
	storm_intensity.py::TestShouldGenerate	test_returns_bool	Pass
	storm_intensity.py::TestShouldGenerate	test_unknown_category_defaults_to_015	Pass
	test_classifier_part1.py::TestPrintClassifierResult	test_no_metrics_printed	Pass
	test_classifier_part1.py::TestPrintClassifierResult	test_status	Pass
	test_classifier_part1.py::TestPrintClassifierResult	test_shows_feature_importance	Pass
	test_classifier_part1.py::TestPrintClassifierResult	test_shows_label_info	Pass
	test_classifier_part1.py::TestPrintClassifierResult	test_status_prints_metrics	Pass
	test_classifier_part1.py::TestPrintGaugeProgress	test_gauge_marker_in_output	Pass
	test_classifier_part1.py::TestPrintGaugeProgress	test_prints_one_types_reported	Pass
	test_classifier_part1.py::TestPrintGaugeProgress	test_empty_category_skipped	Pass
	test_classifier_part1.py::TestPrintGaugeProgress	test_output_contains_gauge_id	Pass
	test_classifier_part1.py::TestPrintGaugeProgress	test_severe_marker_in_output	Pass
	test_classifier_part1.py::TestPrintGaugeProgress	test_warning_marker_in_output	Pass
	test_classifier_part2.py::TestPortfolioClassifier	test_portfolio_classifier_summary_unchanged	Pass
	test_classifier_part2.py::TestPortfolioClassifier	test_portfolio_classifier_trains_all_gauges	Pass
	test_classifier_part2.py::TestTrainGaugeFallbackClassifier	test_fallback_classifier_when_no_severe_events	Pass
	test_classifier_part2.py::TestTrainGaugeStatsClassifier	test_model_file_written	Pass
	test_classifier_part2.py::TestTrainGaugeStatsClassifier	test_stats_has_gauge_id	Pass
	test_classifier_part2.py::TestTrainGaugeStatsClassifier	test_stats_has_label_threshold	Pass
	test_classifier_part2.py::TestTrainGaugeStatsClassifier	test_returns_classifier_dict	Pass
	test_classifier_part2.py::TestTrainGaugeStatsClassifier	test_severe_zero_search	Pass
	test_sequence_response_compound.py::TestCompoundStrategy	test_compounding_stronger_for_short_gaps	Pass
	test_sequence_response_compound.py::TestCompoundStrategy	test_gap_monotonically_increasing	Pass
	test_sequence_response_compound.py::TestCompoundStrategy	test_short_gap_matches_baseline	Pass
	test_sequence_response_compound.py::TestCompoundStrategy	test_storm2_peak_exceeds_storm1_majority	Pass
	test_sequence_response_compound.py::TestPeakCompositeStormPeaks	test_above_baseline	Pass
	test_sequence_response_compound.py::TestPeakCompositeStormPeaks	test_indices_match_storm_indices	Pass
	test_sequence_response_compound.py::TestPeakCompositeStormPeaks	test_one_peak_per_storm	Pass
	test_sequence_response_hydro_pattern.py::TestHydrologyModelRunSeries	test_consistent_with_model	Pass
	test_sequence_response_hydro_pattern.py::TestHydrologyModelRunSeries	test_initial_state_affects_output	Pass
	test_sequence_response_hydro_pattern.py::TestHydrologyModelRunSeries	test_non_negative_quickflow	Pass
	test_sequence_response_hydro_pattern.py::TestHydrologyModelRunSeries	test_quickflow_positive	Pass
	test_sequence_response_hydro_pattern.py::TestHydrologyModelRunSeries	test_shape	Pass
	test_sequence_response_hydro_pattern.py::TestHydrologyModelRunSeries	test_state_decays_without_rain	Pass
	test_sequence_response_hydro_pattern.py::TestHydrologyModelStep	test_dry_start_no_quickflow	Pass
	test_sequence_response_hydro_pattern.py::TestHydrologyModelStep	test_groundwater_accumulates_from_soil	Pass
	test_sequence_response_hydro_pattern.py::TestHydrologyModelStep	test_negative_treated_as_zero	Pass
	test_sequence_response_hydro_pattern.py::TestHydrologyModelStep	test_precipitation_increases_soil_moisture	Pass
	test_sequence_response_hydro_pattern.py::TestHydrologyModelStep	test_produces_quickflow	Pass
	test_sequence_response_hydro_pattern.py::TestHydrologyModelStep	test_soil_moisture_decays_without_rain	Pass
	test_sequence_response_hydro_pattern.py::TestHydrologyModelStep	test_wet_start_produces_quickflow	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_sequence_response_hydro_pattern.py::	TestMakePrecipSeries::test_single_storm_peak	Pass
	test_sequence_response_hydro_pattern.py::	TestMakePrecipSeries::test_two_storms	Pass
	test_sequence_response_hydro_pattern.py::	TestMakePrecipSeries::test_negative	Pass
	test_sequence_response_hydro_pattern.py::	TestMakePrecipSeries::test_output_shape	Pass
	test_sequence_response_hydro_pattern.py::	TestMakePrecipSeries::test_total_precip_conserved	Pass
	test_sequence_response_hydro_pattern.py::	TestComputeSequenceGaugeResponse::test_peak	Pass
	test_sequence_response_hydro_pattern.py::	TestComputeSequenceGaugeResponse::test_drainage_window_lower_gauge_response	Pass
	test_sequence_response_hydro_pattern.py::	TestComputeSequenceGaugeResponse::test_higher_intensity_higher_gauge_response	Pass
	test_sequence_response_hydro_pattern.py::	TestComputeSequenceGaugeResponse::test_doubling_same_first_storm	Pass
	test_sequence_response_hydro_pattern.py::	TestComputeSequenceGaugeResponse::test_above_base_during_storm	Pass
	test_sequence_response_hydro_pattern.py::	TestComputeSequenceGaugeResponse::test_negative_values	Pass
	test_sequence_response_hydro_pattern.py::	TestComputeSequenceGaugeResponse::test_precip_in_drainage_window	Pass
	test_sequence_response_hydro_pattern.py::	TestComputeSequenceGaugeResponse::test_output_shape	Pass
	test_sequence_response_hydro_pattern.py::	TestComputeSequenceGaugeResponse::test_return_to_baseline_by_hour_168	Pass
	test_spatial_correlation_matrix.py::	TestCholesky::test_cache_key_rounding	Pass
	test_spatial_correlation_matrix.py::	TestCholesky::test_cached	Pass
	test_spatial_correlation_matrix.py::	TestCholesky::test_lower_triangular	Pass
	test_spatial_correlation_matrix.py::	TestCholesky::test_reconstruction	Pass
	test_spatial_correlation_matrix.py::	TestCholesky::test_cholesky_reconstruction	Pass
	test_spatial_correlation_matrix.py::	TestDiagonalIsOne::test_diagonal_is_one	Pass
	test_spatial_correlation_matrix.py::	TestIncreasingDistance::test_higher_correlation	Pass
	test_spatial_correlation_matrix.py::	TestNoneCorrelation::test_off_diagonal	Pass
	test_spatial_correlation_matrix.py::	TestOffDiagonal::test_open_interval	Pass
	test_spatial_correlation_matrix.py::	TestPositiveDefinite::test_positive_definite	Pass
	test_spatial_correlation_matrix.py::	TestSymmetric::test_symmetric	Pass
	test_spatial_correlation_matrix.py::	TestCorrelationMatrix::test_positive_definite	Pass
	test_spatial_correlation_matrix.py::	TestHigherIntensity::test_longer_range	Pass
	test_spatial_correlation_matrix.py::	TestMinimalIntensity::test_no_reduction	Pass
	test_spatial_correlation_matrix.py::	TestModerateIntensity::test_equals_base	Pass
	test_spatial_correlation_matrix.py::	TestRangeScales::test_with_rho	Pass
	test_spatial_coverage_b6.py::	TestCholeskyFallback::test_fallback_adds_jitter	Pass
	test_spatial_coverage_b6.py::	TestFromCoverageGroup::test_portfolio_file	Pass
	test_spatial_coverage_b6.py::	TestSampleMultipliers::test_without_rng	Pass
	test_spatial_multipliers.py::	TestApplyToSequence::test_non_negative	Pass
	test_spatial_multipliers.py::	TestApplyToSequence::test_mean_preserved_per_hour	Pass
	test_spatial_multipliers.py::	TestApplyToSequence::test_output_shape	Pass
	test_spatial_multipliers.py::	TestApplyToSequence::test_spatial_variation_present	Pass
	test_spatial_multipliers.py::	TestApplyToSequence::test_total_precipitation_conserved	Pass
	test_spatial_multipliers.py::	TestApplyToSequence::test_zero_precip_hours_remain_zero	Pass
	test_spatial_multipliers.py::	TestSampleMultipliers::test_positive	Pass
	test_spatial_multipliers.py::	TestSampleMultipliers::test_empirical_correlation_consistent_with_model	Pass
	test_spatial_multipliers.py::	TestSampleMultipliers::test_higher_intensity_lower_variance	Pass
	test_spatial_multipliers.py::	TestSampleMultipliers::test_mean_is_one	Pass
	test_spatial_multipliers.py::	TestSampleMultipliers::test_normalisation_lognormal_mean	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_spatial_multipliers.py::TestSampleMultipliers	test_reproducible_with_seed	Pass
	test_spatial_multipliers.py::TestSampleMultipliers	test_shape	Pass
	test_spatial_params_distance.py::TestConfigFile	test_constants_valid	Pass
	test_spatial_params_distance.py::TestConfigFile	test_structure	Pass
	test_spatial_params_distance.py::TestConfigFile	test_trip_params	Pass
	test_spatial_params_distance.py::TestSaveFile	test_creates_file	Pass
	test_spatial_params_distance.py::TestDiagonalZero	test_diagonal_zero	Pass
	test_spatial_params_distance.py::TestDistanceMatrix	test_known_distance	Pass
	test_spatial_params_distance.py::TestDistanceMatrix	test_off_diagonal	Pass
	test_spatial_params_distance.py::TestDistanceMatrix	test_shape	Pass
	test_spatial_params_distance.py::TestDistanceMatrix	test_symmetric	Pass
	test_spatial_params_distance.py::TestDistanceMatrix	test_distance_plausible	Pass
	test_spatial_params_distance.py::TestDistanceMatrix	test_gauges	Pass
	test_spatial_params_distance.py::TestFromGaugeMatrix	test_portfolio	Pass
	test_spatial_params_distance.py::TestFromGaugePortfolioFile	test_shape	Pass
	test_spatial_params_distance.py::TestFromGaugePortfolioFile	test_52_gauges	Pass
	test_spatial_params_distance.py::TestSpatialCorrelationParams	test_shape	Pass
	test_spatial_params_distance.py::TestSpatialCorrelationParams	test_from_file	Pass
	test_spatial_params_distance.py::TestSpatialCorrelationParams	test_load_from_file	Pass
	test_import_isolation_part1.py::TestBookExportsQuantlib	test_exports_when_quantlib_present	Pass
	test_import_isolation_part1.py::TestBookRequiresQuantlib	test_importable_without_quantlib	Pass
	test_import_isolation_part1.py::TestBookRequiresQuantlib	test_thames_central_areas_count	Pass
	test_import_isolation_part1.py::TestBookRequiresQuantlib	test_pricing_requires_quantlib	Pass
	test_import_isolation_part1.py::TestGaugeIsImportable	test_importable_without_quantlib	Pass
	test_import_isolation_part1.py::TestGenerateHazardImportable	test_importable_without_quantlib	Pass
	test_import_isolation_part1.py::TestGenerateLossImportable	test_importable_without_quantlib	Pass
	test_import_isolation_part1.py::TestGenerateReportsImportable	test_importable_without_quantlib	Pass
	test_import_isolation_part1.py::TestGenerateStormsImportable	test_importable_without_quantlib	Pass
	test_import_isolation_part1.py::TestPackageDocNoEagerImports	test_importing_book_symbols	Pass
	test_import_isolation_part1.py::TestPackageDocNoEagerImports	test_import_book	Pass
	test_import_isolation_part1.py::TestPackageDocNoEagerImports	test_generate_thames_central	Pass
	test_import_isolation_part2.py::TestCmdPlotImportable	test_importable_without_matplotlib	Pass
	test_import_isolation_part2.py::TestCmdPlotImportable	test_importable_without_quantlib	Pass
	test_import_isolation_part2.py::TestGeoDamageImportable	test_importable_without_geopandas	Pass
	test_import_isolation_part2.py::TestGeoFloodRiskImportable	test_does_not_import_flood_risk_model	Pass
	test_import_isolation_part2.py::TestGeoPropertyImportable	test_importable_without_geopandas	Pass
	test_import_isolation_part2.py::TestGeoscalarDepthDamage	test_works_without_geopandas	Pass
	test_import_isolation_part2.py::TestGeoSpatialImportable	test_importable_without_geopandas	Pass
	analysis.py::TestMultiGaugeComparison	test_count_gauges_above_threshold	Pass
	analysis.py::TestMultiGaugeComparison	test_find_highest_level_gauge	Pass
	analysis.py::TestSingleGaugeAnalysis	test_full_gauge_analysis	Pass
	analysis.py::TestSingleGaugeAnalysis	test_gauge_not_found	Pass
	analysis.py::TestSingleGaugeAnalysis	test_gauge_status_in_analysis	Pass
	analysis.py::TestSingleGaugeAnalysis	test_load_gauge_metadata	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	analysis.py::TestSingleGaugeAnalysis	test_load_gauge_readings	Pass
	classification.py::TestExceedanceProbability	test_high_level_low_probability	Pass
	classification.py::TestExceedanceProbability	test_low_level_high_probability	Pass
	classification.py::TestExceedanceProbability	test_return_period_calculation	Pass
	classification.py::TestFloodDuration	test_duration_above_threshold	Pass
	classification.py::TestFloodDuration	test_duration_hours_calculation	Pass
	classification.py::TestFloodDuration	test_no_duration_normal_readings	Pass
	classification.py::TestGaugeLevelAnalysis	test_current_level	Pass
	classification.py::TestGaugeLevelAnalysis	test_level_trend_falling	Pass
	classification.py::TestGaugeLevelAnalysis	test_level_trend_rising	Pass
	classification.py::TestGaugeLevelAnalysis	test_level_trend_stable	Pass
	classification.py::TestGaugeLevelAnalysis	test_peak_level_detection	Pass
	classification.py::TestGaugeLevelAnalysis	test_peak_timing	Pass
	classification.py::TestGaugeStatusClassification	test_classification_status	Pass
	classification.py::TestGaugeStatusClassification	test_between_thresholds	Pass
	classification.py::TestGaugeStatusClassification	test_normal_status	Pass
	classification.py::TestGaugeStatusClassification	test_severe_status	Pass
	classification.py::TestGaugeStatusClassification	test_warning_status	Pass
	cli.py::TestErrorHandling	test_bad_gauge_file_exits_with_1	Pass
	cli.py::TestErrorHandling	test_invalid_json_exits_with_1	Pass
	cli.py::TestListCategories	test_list_categories_does_not_open_gauge_file	Pass
	cli.py::TestListCategories	test_list_categories_runs_without_error	Pass
	cli.py::TestListPages	test_list_pages_does_not_open_gauge_file	Pass
	cli.py::TestListPages	test_list_pages_runs_without_error	Pass
	cli.py::TestOptionalLoading	test_gauge_id_filters_data	Pass
	cli.py::TestOptionalLoading	test_timeseries_file_is_loaded	Pass
	cli.py::TestReportGeneration	test_monitoring_without_timeseries_hits_else	Pass
	cli.py::TestReportGeneration	test_report_type_analysis	Pass
	cli.py::TestReportGeneration	test_report_type_basic	Pass
	cli.py::TestReportGeneration	test_report_type_full_hits_else_branch	Pass
	cli.py::TestReportGeneration	test_report_type_monitoring_with_timeseries	Pass
	cli.py::TestReportGeneration	test_success_logging_executes	Pass
	convenience.py::TestEdgeCases	test_completely_empty_gauge_data	Pass
	convenience.py::TestEdgeCases	test_gauge_data_with_hazard_curve_and_timeseries	Pass
	convenience.py::TestEdgeCases	test_none_flood_gauge_header	Pass
	convenience.py::TestEdgeCases	test_output_is_non_empty_bytes	Pass
	convenience.py::TestEdgeCases	test_report_with_all_pages	Pass
	convenience.py::TestEdgeCases	test_single_page_report	Pass
	convenience.py::TestFindGaugeById	test_dict_without_known_key_raises	Pass
	convenience.py::TestFindGaugeById	test_filtered_gauges_key_first_gauge	Pass
	convenience.py::TestFindGaugeById	test_filtered_gauges_key_last_gauge	Pass
	convenience.py::TestFindGaugeById	test_gauge_not_found_raises	Pass
	convenience.py::TestFindGaugeById	test_integer_input_raises	Pass
	convenience.py::TestFindGaugeById	test_list_input_used_directly	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	convenience.py::TestFindGaugeBytes	test_malformed_entry_missing_flood_gauge_key	Pass
	convenience.py::TestFindGaugeBytes	test_malformed_entry_with_none_skipped	Pass
	convenience.py::TestFindGaugeBytes	test_single_flood_gauge_dict_returns_itself	Pass
	convenience.py::TestFindGaugeBytes	test_string_input_raises	Pass
	convenience.py::TestGenerateGaugeReport	test_analysis_type_calls_generate_analysis	Pass
	convenience.py::TestGenerateGaugeReport	test_open_false_returns_path	Pass
	convenience.py::TestGenerateGaugeReport	test_open_true_handles_exception	Pass
	convenience.py::TestGenerateGaugeReport	test_open_true_open_succeeds	Pass
	convenience.py::TestGenerateGaugeReport	test_basic_type_calls_generate_basic	Pass
	convenience.py::TestGenerateGaugeReport	test_monitoring_type_with_timeseries_calls_monitoring	Pass
	convenience.py::TestGenerateGaugeReport	test_monitoring_without_timeseries_falls_back	Pass
	convenience.py::TestGenerateGaugeReport	test_returns_path	Pass
	convenience.py::TestGenerateGaugeReport	test_unknown_type_calls_generate_report	Pass
	elements.py::TestGenerateElements	test_empty_page_list_returns_empty	Pass
	elements.py::TestGenerateElements	test_exception_in_page_continues	Pass
	elements.py::TestGenerateElements	test_first_page_no_page_break	Pass
	elements.py::TestGenerateElements	test_mixed_valid_and_unknown_pages	Pass
	elements.py::TestGenerateElements	test_multiple_pages_have_page_break	Pass
	elements.py::TestGenerateElements	test_none_timeseries_passed_to_page	Pass
	elements.py::TestGenerateElements	test_returns_list	Pass
	elements.py::TestGenerateElements	test_timeseries_passed_to_page	Pass
	elements.py::TestGenerateElements	test_unknown_page_skipped	Pass
	elements.py::TestGenerateReport	test_all_unknown_pages_still_creates_pdf	Pass
	elements.py::TestGenerateReport	test_auto_select_called_when_pages_none	Pass
	elements.py::TestGenerateReport	test_auto_select_not_called_when_pages_provided	Pass
	elements.py::TestGenerateReport	test_custom_output_filename	Pass
	elements.py::TestGenerateReport	test_custom_pages_to_include	Pass
	elements.py::TestGenerateReport	test_empty_gauge_data	Pass
	elements.py::TestGenerateReport	test_pdf_exists	Pass
	elements.py::TestGenerateReport	test_pdf_is_bytes	Pass
	elements.py::TestGenerateReport	test_pdf_non_empty	Pass
	elements.py::TestGenerateReport	test_pdf_suffix	Pass
	elements.py::TestGenerateReport	test_returns_path	Pass
	elements.py::TestGenerateReport	test_with_timeseries_data	Pass
	init.py::TestAutoSelectPages	test_all_pages_exist	Pass
	init.py::TestAutoSelectPages	test_always_includes_gauge_info_pages	Pass
	init.py::TestAutoSelectPages	test_always_includes_operational_pages	Pass
	init.py::TestAutoSelectPages	test_always_includes_summary_pages	Pass
	init.py::TestAutoSelectPages	test_returns_list	Pass
	init.py::TestAutoSelectPages	test_with_hazard_curve_includes_analysis_pages	Pass
	init.py::TestAutoSelectPages	test_with_timeseries_includes_analysis_pages	Pass
	init.py::TestAutoSelectPages	test_without_timeseries_or_hazard_no_analysis_pages	Pass
	init.py::TestGaugeReportGenerator	test_categories_dict_initialized	Pass
	init.py::TestGaugeReportGenerator	test_explicit_output_dir_path	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	init.py::TestGaugeReportGenerator	test_explicit_output_dir_string	Pass
	init.py::TestGaugeReportGenerator	test_no_output_dir_uses_config	Pass
	init.py::TestGaugeReportGenerator	test_output_dir_created	Pass
	init.py::TestGaugeReportGenerator	test_pages_dict_initialized	Pass
	init.py::TestGenerateFilename	test_contains_gauge_id	Pass
	init.py::TestGenerateFilename	test_ends_with_pdf	Pass
	init.py::TestGenerateFilename	test_missing_gauge_id_uses_unknown	Pass
	init.py::TestGenerateFilename	test_partial_path_missing_gauge_id	Pass
	init.py::TestGenerateFilename	test_partial_path_missing_header	Pass
	init.py::TestGenerateFilename	test_returns_string	Pass
	init.py::TestGenerateFilename	test_timestamp_in_filename	Pass
	specialised.py::TestHeaderFooter	test_header_footer_callable	Pass
	specialised.py::TestHeaderFooter	test_header_footer_draws_confidential_footer	Pass
	specialised.py::TestHeaderFooter	test_header_footer_draws_header_strings	Pass
	specialised.py::TestSpecialisedReport	test_generate_analysis_report_custom_filename	Pass
	specialised.py::TestSpecialisedReport	test_generate_analysis_report_exists	Pass
	specialised.py::TestSpecialisedReport	test_generate_analysis_report_with_timeseries	Pass
	specialised.py::TestSpecialisedReport	test_generate_basic_report_custom_filename	Pass
	specialised.py::TestSpecialisedReport	test_generate_basic_report_exists	Pass
	specialised.py::TestSpecialisedReport	test_generate_basic_report_non_empty	Pass
	specialised.py::TestSpecialisedReport	test_generate_basic_with_timeseries	Pass
	specialised.py::TestSpecialisedReport	test_generate_monitoring_report_custom_filename	Pass
	specialised.py::TestSpecialisedReport	test_generate_monitoring_report_exists	Pass
	specialised.py::TestSpecialisedReport	test_generate_monitoring_report_no_timeseries	Pass
	specialised.py::TestUtilityMethods	test_get_page_categories_contains_expected_keys	Pass
	specialised.py::TestUtilityMethods	test_get_page_categories_returns_copy	Pass
	specialised.py::TestUtilityMethods	test_get_page_categories_returns_dict	Pass
	specialised.py::TestUtilityMethods	test_list_available_pages_contains_expected	Pass
	specialised.py::TestUtilityMethods	test_list_available_pages_non_empty	Pass
	specialised.py::TestUtilityMethods	test_list_available_pages_returns_list	Pass
	specialised.py::TestUtilityMethods	test_validate_pages_all_invalid	Pass
	specialised.py::TestUtilityMethods	test_validate_pages_all_valid	Pass
	specialised.py::TestUtilityMethods	test_validate_pages_empty_list	Pass
	specialised.py::TestUtilityMethods	test_validate_pages_mixed	Pass
	metrics.py::TestGaugeMetrics	test_level_range	Pass
	metrics.py::TestGaugeMetrics	test_max_level	Pass
	metrics.py::TestGaugeMetrics	test_mean_level	Pass
	metrics.py::TestGaugeMetrics	test_min_level	Pass
	metrics.py::TestGaugeMetrics	test_rate_of_rise	Pass
	page_01_title_overview.py::TestGaugeTitleOverviewPage	test_build_overview_page_does_not_crash	Pass
	page_01_title_overview.py::TestGaugeTitleOverviewPage	test_empty_data_does_not_crash	Pass
	page_01_title_overview.py::TestGaugeTitleOverviewPage	test_gauge_id_in_output	Pass
	page_01_title_overview.py::TestGaugeTitleOverviewPage	test_overview_section_header	Pass
	page_01_title_overview.py::TestGaugeTitleOverviewPage	test_summary_section_header	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	page_01_title_overview.py::TestGaugeIdPassing	test_get_gauge_id_passing	Pass
	page_01_title_overview.py::TestGaugeIdPresent	test_get_gauge_id_present	Pass
	page_01_title_overview.py::TestHasTables	test_has_tables	Pass
	page_01_title_overview.py::TestInstallationPage	test_installation_page_calculated	Pass
	page_01_title_overview.py::TestInstallationPage	test_installation_page_does_not_crash	Pass
	page_01_title_overview.py::TestReturnsList	test_returns_list	Pass
	page_01_title_overview.py::TestTitleOverviewPage	test_title_overview_page_present	Pass
	page_01_title_overview.py::TestTitleOverviewPage	test_title_overview_page_fallback	Pass
	page_01_title_overview.py::TestWithOtherPages	test_with_other_pages_does_not_crash	Pass
	page_02_sensor_details.py::TestEmptyDataDoesNotCrash	test_empty_data_does_not_crash	Pass
	page_02_sensor_details.py::TestGaugeDetailsPage	test_gauge_details_subtitle	Pass
	page_02_sensor_details.py::TestGaugeDetailsPage	test_gauge_details_location_section	Pass
	page_02_sensor_details.py::TestGetGaugeIdPassing	test_get_gauge_id_passing	Pass
	page_02_sensor_details.py::TestGetGaugeIdPresent	test_get_gauge_id_present	Pass
	page_02_sensor_details.py::TestHasTables	test_has_tables	Pass
	page_02_sensor_details.py::TestHistoricalPage	test_historical_page_sensor_statistics_section	Pass
	page_02_sensor_details.py::TestLocationPage	test_location_page_information_section	Pass
	page_02_sensor_details.py::TestMeasurementsPage	test_measurements_page_configuration_section	Pass
	page_02_sensor_details.py::TestNoSensorDetails	test_no_sensor_details_does_not_crash	Pass
	page_02_sensor_details.py::TestReturnsList	test_returns_list	Pass
	page_02_sensor_details.py::TestTimeDetailsPage	test_time_details_page	Pass
	page_02_sensor_details.py::TestWithOtherPages	test_with_other_pages_does_not_crash	Pass
	page_03_location.py::TestGaugeLocationPage	test_in_page_distance_close_to_thames	Pass
	page_03_location.py::TestGaugeLocationPage	test_in_page_distance_zero_on_thames	Pass
	page_03_location.py::TestGaugeLocationPage	test_empty_data_does_not_crash	Pass
	page_03_location.py::TestGaugeLocationPage	test_flood_risk_section	Pass
	page_03_location.py::TestGaugeLocationPage	test_geographic_coordinates_section	Pass
	page_03_location.py::TestGaugeLocationPage	test_has_tables	Pass
	page_03_location.py::TestGaugeLocationPage	test_in_page_central_london	Pass
	page_03_location.py::TestGaugeLocationPage	test_in_page_north_london	Pass
	page_03_location.py::TestGaugeLocationPage	test_in_page_south_london	Pass
	page_03_location.py::TestGaugeLocationPage	test_on_the_thames_info_does_not_crash	Pass
	page_03_location.py::TestGaugeLocationPage	test_partitioning_summary_section	Pass
	page_03_location.py::TestGaugeLocationPage	test_returns_list	Pass
	page_03_location.py::TestGaugeLocationPage	test_risk_category_in_location_profile	Pass
	page_03_location.py::TestGaugeLocationPage	test_risk_score_formatted_as_out_of_10	Pass
	page_03_location.py::TestGaugeLocationPage	test_thames_context_section	Pass
	page_03_location.py::TestGaugeLocationPage	test_title_present	Pass
	page_04_measurements.py::TestEmptyDataDoesNotCrash	test_empty_data_does_not_crash	Pass
	page_04_measurements.py::TestEmptyReadingsDoesNotCrash	test_empty_readings_list_does_not_crash	Pass
	page_04_measurements.py::TestGetGaugeIdPassing	test_get_gauge_id_passing	Pass
	page_04_measurements.py::TestGetGaugeIdPresent	test_get_gauge_id_present	Pass
	page_04_measurements.py::TestHasTables	test_has_tables	Pass
	page_04_measurements.py::TestHistoricalStatsPage	test_historical_stats_page_statistics_section	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	page_04_measurements.py::TestGaugeMeasurementsPage	test_measurements_page_configuration_section	Pass
	page_04_measurements.py::TestGaugeMeasurementsPage	test_no_timeseries_shows_message	Pass
	page_04_measurements.py::TestGaugeMeasurementsPage	test_operational_status_section	Pass
	page_04_measurements.py::TestGaugeMeasurementsPage	test_returns_200	Pass
	page_04_measurements.py::TestGaugeMeasurementsPage	test_title_present	Pass
	page_04_measurements.py::TestGaugeMeasurementsPage	test_with_timeseries_has_extra_table	Pass
	page_04_measurements.py::TestGaugeMeasurementsPage	test_with_timeseries_shows_current_readings	Pass
	page_05_flood_stages.py::TestGaugeFloodStagesPage	test_current_level_at_alert	Pass
	page_05_flood_stages.py::TestGaugeFloodStagesPage	test_current_level_at_severe	Pass
	page_05_flood_stages.py::TestGaugeFloodStagesPage	test_current_level_at_warning	Pass
	page_05_flood_stages.py::TestGaugeFloodStagesPage	test_current_level_below_alert_is_normal	Pass
	page_05_flood_stages.py::TestGaugeFloodStagesPage	test_empty_data_does_not_crash	Pass
	page_05_flood_stages.py::TestGaugeFloodStagesPage	test_gauge_id_missing	Pass
	page_05_flood_stages.py::TestGaugeFloodStagesPage	test_gauge_id_present	Pass
	page_05_flood_stages.py::TestGaugeFloodStagesPage	test_has_stat_cards	Pass
	page_05_flood_stages.py::TestGaugeFloodStagesPage	test_historical_exceedance_section	Pass
	page_05_flood_stages.py::TestGaugeFloodStagesPage	test_no_timeseries_shows_message	Pass
	page_05_flood_stages.py::TestGaugeFloodStagesPage	test_returns_200	Pass
	page_05_flood_stages.py::TestGaugeFloodStagesPage	test_threshold_analysis_section	Pass
	page_05_flood_stages.py::TestGaugeFloodStagesPage	test_threshold_differences_computed	Pass
	page_05_flood_stages.py::TestGaugeFloodStagesPage	test_title_present	Pass
	page_05_flood_stages.py::TestGaugeFloodStagesPage	test_thresholds_section	Pass
	page_05_flood_stages.py::TestGaugeFloodStagesPage	test_with_timeseries_shows_current_status_section	Pass
	page_06_risk_assessment_part1.py::TestDistanceToFlood	test_distance_label	Pass
	page_06_risk_assessment_part1.py::TestDistanceToFlood	test_zero_distance_highest_risk	Pass
	page_06_risk_assessment_part1.py::TestFrequencyExceedances	test_four_exceedances_high	Pass
	page_06_risk_assessment_part1.py::TestFrequencyExceedances	test_one_exceedance	Pass
	page_06_risk_assessment_part1.py::TestFrequencyExceedances	test_two_exceedances_moderate	Pass
	page_06_risk_assessment_part1.py::TestFrequencyExceedances	test_zero_exceedances	Pass
	page_06_risk_assessment_part1.py::TestGaugeRiskAssessmentPageBasic	test_generate_assessment_page	Pass
	page_06_risk_assessment_part1.py::TestGaugeRiskAssessmentPageBasic	test_gauge_risk_assessment_shows_page_header	Pass
	page_06_risk_assessment_part1.py::TestGaugeRiskAssessmentPageBasic	test_with_timeseries_data	Pass
	page_06_risk_assessment_part1.py::TestHistoricalHighWater	test_exceedance_high_water_label	Pass
	page_06_risk_assessment_part1.py::TestHistoricalHighWater	test_historical_high_water_level	Pass
	page_06_risk_assessment_part1.py::TestRiskScoreBranches	test_high_risk_label	Pass
	page_06_risk_assessment_part1.py::TestRiskScoreBranches	test_low_risk_label	Pass
	page_06_risk_assessment_part1.py::TestRiskScoreBranches	test_moderate_risk_label	Pass
	page_06_risk_assessment_part1.py::TestRiskScoreBranches	test_risk_score_formatted	Pass
	page_06_risk_assessment_part1.py::TestRiskScoreBranches	test_very_high_risk_label	Pass
	page_06_risk_assessment_part2.py::TestExceptionHandling	test_get_gauge_missing_key	Pass
	page_06_risk_assessment_part2.py::TestExceptionHandling	test_malformed_gauge_data_returns_error_element	Pass
	page_06_risk_assessment_part2.py::TestLastExceedDate	test_last_exceedance_gracefully_skipped	Pass
	page_06_risk_assessment_part2.py::TestLastExceedDate	test_last_exceedance_recent_activity	Pass
	page_06_risk_assessment_part2.py::TestLastExceedDate	test_last_exceedance_old_activity	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	page_06_risk_assessment_part2.py::TestRecentActivity	test_recent_activity_within_1_year	Pass
	page_06_risk_assessment_part2.py::TestThresholdAnalysis	test_threshold_analysis_exceeds_severe_by_less_than_1m	Pass
	page_06_risk_assessment_part2.py::TestThresholdAnalysis	test_threshold_analysis_exceeds_severe_by_more_than_1m	Pass
	page_06_risk_assessment_part2.py::TestThresholdAnalysis	test_threshold_analysis_within_severe	Pass
	page_06_risk_assessment_part2.py::TestThresholdAnalysis	test_threshold_analysis_moderate_range	Pass
	page_06_risk_assessment_part2.py::TestThresholdAnalysis	test_threshold_analysis_alert_range	Pass
	page_06_risk_assessment_part2.py::TestThresholdAnalysis	test_threshold_analysis_wide_alert_range	Pass
	page_06_risk_assessment_part2.py::TestTimeseriesAlertStatus	test_timeseries_alert_status	Pass
	page_06_risk_assessment_part2.py::TestTimeseriesAlertStatus	test_timeseries_alert_status	Pass
	page_06_risk_assessment_part2.py::TestTimeseriesAlertStatus	test_timeseries_alert_status	Pass
	page_06_risk_assessment_part2.py::TestTimeseriesAlertStatus	test_timeseries_alert_no_level	Pass
	page_06_risk_assessment_part2.py::TestTimeseriesAlertStatus	test_timeseries_alert_no_readings	Pass
	page_06_risk_assessment_part2.py::TestTimeseriesAlertStatus	test_timeseries_alert_status	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_data_summary_page	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_empty_data_does_not_crash	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_freq_high_significance	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_freq_low_significance	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_freq_zero_exceedances_conclusion	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_freq_zero_significance	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_gauge_overview_summary_section	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_get_gauge_id_missing	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_get_gauge_id_present	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_get_gauge_name_missing	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_get_gauge_name_present	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_has_tables	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_high_risk_conclusion	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_high_exceeds_severe_significance	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_high_within_range_significance	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_key_metrics_summary_section	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_low_risk_conclusion	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_moderate_risk_conclusion	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_operational_status_conclusion	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_operational_status_summary_section	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_report_conclusions_section	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_terms_history	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_timeseries_alert_conclusion	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_timeseries_normal_conclusion	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_timeseries_severe_conclusion	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_timeseries_warning_conclusion	Pass
	page_07_data_summary.py::TestGaugeDataSummaryPage	test_title_present	Pass
	page_08_flood_history.py::TestGaugeFloodHistoryPage	test_flood_history_graph_no_dates_fallback	Pass
	page_08_flood_history.py::TestGaugeFloodHistoryPage	test_flood_history_graph_with_thresholds	Pass
	page_08_flood_history.py::TestGaugeFloodHistoryPage	test_flood_history_responses_shows_message	Pass
	page_08_flood_history.py::TestGaugeFloodHistoryPage	test_flood_history_data_shows_no_data_message	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	page_08_flood_history.py::TestGauge	test_realised_floods_no_exceedances	Pass
	page_08_flood_history.py::TestGauge	test_realised_floods_with_exceedances	Pass
	page_08_flood_history.py::TestGauge	test_flood_history_page	Pass
	page_08_flood_history.py::TestGauge	test_flood_history_page_with_daily_returns_more_elements	Pass
	page_08_flood_history.py::TestGauge	test_flood_history_page_with_responses_includes_table	Pass
	page_08_flood_history.py::TestGauge	test_flood_history_page_with_severity_classification	Pass
	page_08_flood_history.py::TestGauge	test_flood_history_page_with_top_10_table	Pass
	page_09_hazard_curves.py::TestGauge	test_hazard_curves_page_empty_return_period_levels_no_extra_table	Pass
	page_09_hazard_curves.py::TestGauge	test_hazard_curves_page_shows_fallback	Pass
	page_09_hazard_curves.py::TestGauge	test_hazard_curves_page	Pass
	page_09_hazard_curves.py::TestGauge	test_hazard_curves_page_basic	Pass
	page_09_hazard_curves.py::TestGauge	test_hazard_curves_page_return_period_levels_renders_table	Pass
	page_10_prs_pricing.py::TestGauge	test_prs_pricing_page_shows_fallback	Pass
	page_10_prs_pricing.py::TestGauge	test_prs_pricing_page	Pass
	page_10_prs_pricing.py::TestGauge	test_prs_pricing_page_with_zero_rates_renders_spreads	Pass
	page_10_prs_pricing.py::TestGauge	test_prs_pricing_page_with_rates_renders_na	Pass
	page_11_current_risk.py::TestGauge	test_current_risk_page_empty_historical_daily_shows_fallback	Pass
	page_11_current_risk.py::TestGauge	test_current_risk_page_no_observations_data_shows_fallback	Pass
	page_11_current_risk.py::TestGauge	test_current_risk_page	Pass
	page_11_current_risk.py::TestGauge	test_current_risk_page_with_observations_includes_content	Skip
	page_11_current_risk.py::TestGauge	test_current_risk_page_with_observations_returns_elements	Skip
	page_12_trading_part1.py::TestBuild	test_build_trades_section_returns_message	Pass
	page_12_trading_part1.py::TestBuild	test_build_trades_section_trade_direction	Pass
	page_12_trading_part1.py::TestBuild	test_build_trades_section_trigger_capitalized	Pass
	page_12_trading_part1.py::TestBuild	test_build_trades_section_with_trades_returns_table	Pass
	page_12_trading_part1.py::TestBuild	test_build_trades_section_with_trades_shows_count	Pass
	page_12_trading_part1.py::TestBuild	test_build_market_curves_section_no_market_curves_returns_message	Pass
	page_12_trading_part1.py::TestBuild	test_build_market_curves_section_partial_curves_shows_none_rates	Pass
	page_12_trading_part1.py::TestBuild	test_build_market_adjustments_section_with_adjustments_returns_extra_table	Pass
	page_12_trading_part1.py::TestBuild	test_build_market_curves_section_returns_table	Pass
	page_12_trading_part1.py::TestDrift	test_both_missing_returns_na	Pass
	page_12_trading_part1.py::TestDrift	test_five_year_tenor	Pass
	page_12_trading_part1.py::TestDrift	test_invalid_date_format_returns_na	Pass
	page_12_trading_part1.py::TestDrift	test_missing_end_returns_na	Pass
	page_12_trading_part1.py::TestDrift	test_missing_start_returns_na	Pass
	page_12_trading_part1.py::TestDrift	test_valid_dates_returns_years	Pass
	page_12_trading_part1.py::TestGauge	test_gauge_unknown_key_returns_message	Pass
	page_12_trading_part1.py::TestGauge	test_gauge_returns_list	Pass
	page_12_trading_part1.py::TestGauge	test_with_gauge_id_returns_trading_header	Pass
	page_12_trading_part2.py::TestLoad	test_load_gauge_reports_dir_raises_fallback	Pass
	page_12_trading_part2.py::TestLoad	test_load_gauge_trades_dir_not_exist_returns_empty	Pass
	page_12_trading_part2.py::TestLoad	test_load_gauge_trades_matching_gauge_id_appended	Pass
	page_12_trading_part2.py::TestLoad	test_load_gauge_trades_non_matching_gauge_skipped	Pass
	page_12_trading_part2.py::TestLoad	test_load_market_state_exception_returns_empty	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_base_page_format.py::TestFormatCoordinate	test_coordinate_six_decimals	Pass
	test_base_page_format.py::TestFormatCoordinate	test_negative_coordinate	Pass
	test_base_page_format.py::TestFormatCoordinate	test_non_numeric_delegates_to_format_value	Pass
	test_base_page_format.py::TestFormatFieldVariation	test_abbreviation_replacements	Pass
	test_base_page_format.py::TestFormatFieldVariation	test_camelCase_split	Pass
	test_base_page_format.pyTestFormatFrequency	test_non_numeric_delegates	Pass
	test_base_page_format.pyTestFormatFrequency	test_positive_returns_times	Pass
	test_base_page_format.pyTestFormatFrequency	test_zero_returns_never	Pass
	test_base_page_format.pyTestFormatIntegerValue	test_integer_value	Pass
	test_base_page_format.pyTestFormatNonNumeric	test_non_numeric_delegates_to_format_value	Pass
	test_base_page_format.pyTestFormatNumericValue	test_numeric_value_with_custom_unit	Pass
	test_base_page_format.pyTestFormatNumericValue	test_numeric_value_with_default_unit	Pass
	test_base_page_format.pyTestFormatNumericValue	test_empty_string_returns_not_specified	Pass
	test_base_page_format.pyTestFormatValue	test_False_returns_no	Pass
	test_base_page_format.pyTestFormatValue	test_float_whole_number_strips_decimal	Pass
	test_base_page_format.pyTestFormatValue	test_float_with_decimals_formatted	Pass
	test_base_page_format.pyTestFormatValue	test_integer_returns_str	Pass
	test_base_page_format.pyTestFormatValue	test_None_returns_not_specified	Pass
	test_base_page_format.pyTestFormatValue	test_normal_string_returned	Pass
	test_base_page_format.pyTestFormatValue	test_True_returns_yes	Pass
	test_current_risk.pyTestCurrentRiskGenerateFlowEdgeCases	test_empty_daily_observations_key	Pass
	test_current_risk.pyTestCurrentRiskGenerateFlowEdgeCases	test_empty_obs_list	Pass
	test_current_risk.pyTestCurrentRiskGenerateFlowEdgeCases	test_full_path_returns_flow_events	Pass
	test_current_risk.pyTestCurrentRiskGenerateFlowEdgeCases	test_invalid_date_in_chronological_loop	Pass
	test_current_risk.pyTestCurrentRiskGenerateFlowEdgeCases	test_invalid_date_in_flow_obs	Pass
	test_current_risk.pyTestCurrentRiskGenerateFlowEdgeCases	test_non_numeric_month_levels_all_levels	Pass
	test_current_risk.pyTestCurrentRiskGenerateFlowEdgeCases	test_non_time_series_data	Pass
	test_current_risk.pyTestCurrentRiskGenerateFlowEdgeCases	test_non_dict_in_obs_loop_skipped	Pass
	test_flood_history.pyTestBuildRecordsFromDb	test_alert_classification	Pass
	test_flood_history.pyTestBuildRecordsFromDb	test_invalid_date_in_exceedance_skipped	Pass
	test_flood_history.pyTestBuildRecordsFromDb	test_multi_day_event_updates_peak	Pass
	test_flood_history.pyTestBuildRecordsFromDb	test_multi_day_range_date_format	Pass
	test_flood_history.pyTestBuildRecordsFromDb	test_no_flood_events_returns_message	Pass
	test_flood_history.pyTestBuildRecordsFromDb	test_severe_classification	Pass
	test_flood_history.pyTestBuildRecordsFromDb	test_single_day_event_date_format	Pass
	test_flood_history.pyTestBuildRecordsFromDb	test_two_separate_events	Pass
	test_flood_history.pyTestBuildRecordsFromDb	test_warning_classification	Pass
	test_gauge_generator_extras.pyTestAutoOpen	test_auto_open_called	Pass
	test_gauge_generator_extras.pyTestAutoOpen	test_auto_open_false_does_not_raise	Pass
	test_gauge_generator_extras.pyTestAutoOpen	test_auto_open_success	Pass
	test_gauge_generator_extras.pyTestFindGaugeById	test_finds_first_in_flood_gauges_list	Pass
	test_gauge_generator_extras.pyTestFindGaugeById	test_finds_in_flood_gauges_key	Pass
	test_gauge_generator_extras.pyTestFindGaugeById	test_finds_in_list	Pass
	test_gauge_generator_extras.pyTestFindGaugeById	test_finds_single_gauge_object	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_gauge_generator_extras.py::TestFindGaugeById	test_invalid_id_raises_value_error	Pass
	test_gauge_generator_extras.py::TestFindGaugeById	test_invalid_type_raises_value_error	Pass
	test_gauge_generator_extras.py::TestFindGaugeById	test_not_found_in_list_raises_value_error	Pass
	test_gauge_generator_extras.py::TestFindGaugeById	test_not_found_raises_value_error	Pass
	test_generator_direct.py::TestFindGaugeById	test_find_gauges_key_found	Pass
	test_generator_direct.py::TestFindGaugeById	test_invalid_dict_structure_raises	Pass
	test_generator_direct.py::TestFindGaugeById	test_invalid_type_raises	Pass
	test_generator_direct.py::TestFindGaugeById	test_list_input_found	Pass
	test_generator_direct.py::TestFindGaugeById	test_malformed_gauge_entry_skipped	Pass
	test_generator_direct.py::TestFindGaugeById	test_not_found_raises_value_error	Pass
	test_generator_direct.py::TestFindGaugeById	test_single_gauge_dict_returns_self	Pass
	test_generator_direct.py::TestGaugeGenerateDirectCall	test_generate_basic_report_direct	Pass
	test_generator_direct.py::TestGaugeGenerateDirectCall	test_generate_fn_analysis	Pass
	test_generator_direct.py::TestGaugeGenerateDirectCall	test_generate_fn_auto_open_true	Pass
	test_generator_direct.py::TestGaugeGenerateDirectCall	test_generate_fn_basic	Pass
	test_generator_direct.py::TestGaugeGenerateDirectCall	test_generate_fn_fallback	Pass
	test_generator_direct.py::TestGaugeGenerateDirectCall	test_generate_fn_monitoring	Pass
	test_generator_direct.py::TestGaugeGenerateDirectCall	test_generate_no_pages_uses_auto_select	Pass
	test_generator_direct.py::TestGaugeGenerateDirectCall	test_generate_no_open_returns_false	Pass
	test_generator_direct.py::TestGaugeReportGenerateDirect	test_auto_generate_pages_no_analysis_data	Pass
	test_generator_direct.py::TestGaugeReportGenerateDirect	test_auto_generate_pages_with_hazard	Pass
	test_generator_direct.py::TestGaugeReportGenerateDirect	test_generate_analysis_report	Pass
	test_generator_direct.py::TestGaugeReportGenerateDirect	test_generate_elements_exception_continues	Pass
	test_generator_direct.py::TestGaugeReportGenerateDirect	test_generate_elements_skips_unknown_page	Pass
	test_generator_direct.py::TestGaugeReportGenerateDirect	test_generate_filename_no_gauge_id	Pass
	test_generator_direct.py::TestGaugeReportGenerateDirect	test_generate_filename_with_gauge_id	Pass
	test_generator_direct.py::TestGaugeReportGenerateDirect	test_generate_monitoring_report	Pass
	test_generator_direct.py::TestGaugeReportGenerateDirect	test_get_page_categories	Pass
	test_generator_direct.py::TestGaugeReportGenerateDirect	test_list_available_pages	Pass
	test_generator_direct.py::TestGaugeReportGenerateDirect	test_output_dir_uses_config	Pass
	test_generator_direct.py::TestGaugeReportGenerateDirect	test_validate_pages_valid	Pass
	test_generator_direct.py::TestGaugeReportGenerateDirect	test_validate_pages_with_invalid	Pass
	test_integrator.py::TestGenerateReportForKey	test_all_for_key_structure_works	Pass
	test_integrator.py::TestGenerateReportForGauge	test_for_gauge_returns_path	Pass
	test_integrator.py::TestGenerateReportForGauge	test_gauge_not_found_returns_none	Pass
	test_integrator.py::TestGenerateReportForGauge	test_no_matching_file_returns_none	Pass
	test_integrator.py::TestGenerateReportForGauge	test_output_contains_gauge_id	Pass
	test_integrator.py::TestGenerateReportForGauge	test_output_dir_created	Pass
	test_integrator.py::TestGenerateReportForGauge	test_output_file_created	Pass
	test_integrator.py::TestGetAvailableGauges	test_missing_file_returns_empty_list	Pass
	test_integrator.py::TestGetAvailableGauges	test_returns_all_ids	Pass
	test_integrator.py::TestGetAvailableGauges	test_returns_correct_id	Pass
	test_integrator.py::TestGetAvailableGauges	test_returns_list	Pass
	test_integrator.py::TestValidateGaugeExists	test_exists_key_structure	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_integrator.py::TestValidateGauge	test_existing_gauge_returns_true	Pass
	test_integrator.py::TestValidateGauge	test_existing_file_returns_false	Pass
	test_integrator.py::TestValidateGauge	test_nonexistent_gauge_returns_false	Pass
	test_integrator_exception.py::TestCorrupt	test_corrupt_jsonl_raises_exception	Pass
	test_integrator_exception.py::TestCorrupt	test_corrupt_profile_raises_exception	Pass
	test_integrator_exception.py::TestAvailable	test_available_gauges_skips_empty	Pass
	test_integrator_exception.py::TestAvailable	test_available_gauges_skips_corrupt	Pass
	test_integrator_exception.py::TestValidate	test_validate_gauge_exists_raises_false	Pass
	test_integrator_exception.py::TestValidate	test_validate_gauge_exists_raises_false	Pass
	test_percentile_gauge.py::TestBuildPercentile	test_derived_gauge_RiskLabels	Pass
	test_percentile_gauge.py::TestBuildPercentile	test_fixed_thresholds_RiskLabels_shown_when_present	Pass
	test_percentile_gauge.py::TestBuildPercentile	test_high_label_RiskLabels	Pass
	test_percentile_gauge.py::TestBuildPercentile	test_low_label_RiskLabels	Pass
	test_percentile_gauge.py::TestBuildPercentile	test_normal_label_RiskLabels	Pass
	test_percentile_gauge.py::TestBuildPercentile	test_single_gauge_RiskLabels_guard	Pass
	test_percentile_gauge.py::TestBuildPercentile	test_static_gauge_RiskLabels	Pass
	test_percentile_gauge.py::TestBuildPercentile	test_very_high_label_RiskLabels	Pass
	test_report_generation.py::TestGenerateGaugeReport	test_gauge_report_type	Pass
	test_report_generation.py::TestGenerateGaugeReport	test_gauge_open_failure_does_not_raise	Pass
	test_report_generation.py::TestGenerateGaugeReport	test_gauge_open_true_calls_open_pdf	Pass
	test_report_generation.py::TestGenerateGaugeReport	test_basic_report_type	Pass
	test_report_generation.py::TestGenerateGaugeReport	test_gauge_report_type_with_timeseries	Pass
	test_report_generation.py::TestGenerateGaugeReport	test_returns_path	Pass
	test_report_generation.py::TestGenerateGaugeReport	test_unknown_report_type_falls_back	Pass
	mortgage_pages.py::TestGenerateMortgageReport	test_mortgage_function_creates_pdf	Pass
	mortgage_pages.py::TestGenerateMortgageReport	test_mortgage_function_returns_path	Pass
	mortgage_pages.py::TestMortgageReportGenerator	test_generate_elements_returns_list	Pass
	mortgage_pages.py::TestMortgageReportGenerator	test_generate_filename_fallback	Pass
	mortgage_pages.py::TestMortgageReportGenerator	test_generate_filename_with_id	Pass
	mortgage_pages.py::TestMortgageReportGenerator	test_generate_report_creates_pdf	Pass
	mortgage_pages.py::TestMortgageReportGenerator	test_generate_report_returns_path	Pass
	mortgage_pages.py::TestMortgageReportGenerator	test_init_creates_output_dir	Pass
	mortgage_pages.py::TestMortgageReportGenerator	test_page_order_set	Pass
	mortgage_pages.py::TestMortgageReportGenerator	test_pages_initialised	Pass
	mortgage_pages.py::TestMortgageTitlePage	test_elements_non_empty	Pass
	mortgage_pages.py::TestMortgageTitlePage	test_empty_property_data_does_not_crash	Pass
	mortgage_pages.py::TestMortgageTitlePage	test_extracts_mortgage_id	Pass
	mortgage_pages.py::TestMortgageTitlePage	test_mortgage_data_returns_elements	Pass
	mortgage_pages.py::TestMortgageTitlePage	test_returns_list	Pass
	mortgage_pages.py::TestMortgageTitlePage	test_title_elements_included	Pass
	test_mortgage_generator_coverage.py::TestMortgageGenerator	test_init_no_output_dir_uses_default_OutputDir	Pass
	test_mortgage_generator_coverage.py::TestMortgageGenerator	test_generate_filename_non_dict_raises_attribute_error	Pass
	test_mortgage_generator_coverage.py::TestMortgageGenerator	test_generate_filename_no_mortgage_data	Pass
	test_mortgage_generator_coverage.py::TestMortgageGenerator	test_page_exception_is_caught	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_mortgage_generator_coverage.py::	test_unknown_page_in_order_is_skipped	Pass
	gauge_pdf.py::TestGaugePdfGenerator	test_basic_gauge_report_generates_pdf	Pass
	gauge_pdf.py::TestGaugePdfGenerator	test_full_gauge_report_generates_pdf	Pass
	gauge_pdf.py::TestGaugePdfGenerator	test_gauge_output_dir_created_if_missing	Pass
	gauge_pdf.py::TestGaugePdfGenerator	test_gauge_pdf_filename_contains_gauge_id	Pass
	gauge_pdf.py::TestGaugePdfGenerator	test_gauge_pdf_has_reasonable_size	Pass
	gauge_pdf.py::TestGaugePdfGenerator	test_gauge_pdf_starts_with_pdf_header	Pass
	gauge_pdf.py::TestGaugePdfGenerator	test_gauge_report_with_minimal_data	Pass
	gauge_pdf.py::TestGaugeReportGenerator	test_generator_initialises	Pass
	gauge_pdf.py::TestGaugeReportGenerator	test_list_available_pages	Pass
	gauge_pdf.py::TestGaugeReportGenerator	test_validate_pages	Pass
	property_pdf.py::TestPropertyPdfGenerator	test_full_property_report_generates_pdf	Pass
	property_pdf.py::TestPropertyPdfGenerator	test_mortgage_focused_report_generates_pdf	Pass
	property_pdf.py::TestPropertyPdfGenerator	test_output_dir_created_if_missing	Pass
	property_pdf.py::TestPropertyPdfGenerator	test_pdf_filename_contains_property_id	Pass
	property_pdf.py::TestPropertyPdfGenerator	test_pdf_has_reasonable_size	Pass
	property_pdf.py::TestPropertyPdfGenerator	test_pdf_starts_with_pdf_header	Pass
	property_pdf.py::TestPropertyPdfGenerator	test_property_only_report_generates_pdf	Pass
	property_pdf.py::TestPropertyPdfGenerator	test_report_with_minimal_property_data	Pass
	property_pdf.py::TestPropertyPdfGenerator	test_risk_focused_report_generates_pdf	Pass
	property_pdf.py::TestPropertyReportGenerator	test_auto_select_pages_with_mortgage	Pass
	property_pdf.py::TestPropertyReportGenerator	test_auto_select_pages_without_mortgage	Pass
	property_pdf.py::TestPropertyReportGenerator	test_generator_initialises	Pass
	property_pdf.py::TestPropertyReportGenerator	test_list_available_pages	Pass
	property_pdf.py::TestPropertyReportGenerator	test_validate_pages_valid	Pass
	server_flow.py::TestServerReportFlow	test_gauge_report_via_route_function	Pass
	server_flow.py::TestServerReportFlow	test_gauge_report_with_real_data	Pass
	server_flow.py::TestServerReportFlow	test_property_report_via_route_function	Pass
	server_flow.py::TestServerReportFlow	test_property_report_with_real_data	Pass
	test_data_loader_mixin.py::TestLoadAll	test_classifier_count	Pass
	test_data_loader_mixin.py::TestLoadAll	test_hod_count_defaults_zero_on_import_error	Error
	test_data_loader_mixin.py::TestLoadAll	test_gaugehd_baselines_loaded	Pass
	test_data_loader_mixin.py::TestLoadAll	test_gaugehd_count	Pass
	test_data_loader_mixin.py::TestLoadAll	test_gaugets_count	Pass
	test_data_loader_mixin.py::TestLoadAll	test_loads_counterparties	Pass
	test_data_loader_mixin.py::TestLoadAll	test_loads_gaugehc	Pass
	test_data_loader_mixin.py::TestLoadAll	test_loads_gauges	Pass
	test_data_loader_mixin.py::TestLoadAll	test_loads_mortgages	Pass
	test_data_loader_mixin.py::TestLoadAll	test_loads_properties	Pass
	test_data_loader_mixin.py::TestLoadAll	test_loads_propertyhc	Pass
	test_data_loader_mixin.py::TestLoadAll	test_loads_seq_summary	Pass
	test_data_loader_mixin.py::TestLoadAll	test_loads_stress_storms_index	Pass
	test_data_loader_mixin.py::TestLoadAll	test_loads_stress_storms_legacy_fallback	Pass
	test_data_loader_mixin.py::TestLoadAll	test_missing_files_return_empty	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_data_loader_mixin.py::TestLoadAll	test_propertyts_count	Pass
	test_data_loader_mixin.py::TestLoadAll	test_stress_storms_empty_when_neither_exists	Pass
	test_data_loader_mixin.py::TestLoadAll	test_trade_count_defaults_zero_on_import_error	Pass
	test_data_loader_mixin.py::TestLoadAll	test_training_summary	Pass
	test_data_loader_mixin.py::TestLoadCampbellBaselines	test_computes_summer_mean	Pass
	test_data_loader_mixin.py::TestLoadCampbellBaselines	test_computes_winter_mean	Pass
	test_data_loader_mixin.py::TestLoadCampbellBaselines	test_handles_missing_monthly_means	Pass
	test_data_loader_mixin.py::TestLoadCampbellBaselines	test_loads_baselines	Pass
	test_data_loader_mixin.py::TestLoadCampbellBaselines	test_returns_empty_for_nonexistent_dir	Pass
	test_data_loader_mixin.py::TestLoadCampbellBaselines	test_skips_files_without_mean_level	Pass
	test_data_loader_mixin.py::TestLoadCampbellBaselines	test_skips_invalid_files	Pass
	test_data_loader_primitives.py::TestCountDirs	test_counts_matching_files	Pass
	test_data_loader_primitives.py::TestCountDirs	test_empty_directory	Pass
	test_data_loader_primitives.py::TestCountDirs	test_returns_zero_for_no_matches	Pass
	test_data_loader_primitives.py::TestCountDirs	test_returns_zero_for_none_dir	Pass
	test_data_loader_primitives.py::TestCountDirs	test_returns_zero_for_nonexistent_dir	Pass
	test_data_loader_primitives.py::TestLoadJson	test_loads_list_json	Pass
	test_data_loader_primitives.py::TestLoadJson	test_loads_nested_json	Pass
	test_data_loader_primitives.py::TestLoadJson	test_loads_valid_json	Pass
	test_data_loader_primitives.py::TestLoadJson	test_returns_empty_dict_for_directory	Pass
	test_data_loader_primitives.py::TestLoadJson	test_returns_empty_dict_for_invalid_json	Pass
	test_data_loader_primitives.py::TestLoadJson	test_returns_empty_dict_for_missing_file	Pass
	test_generator_part1.py::TestGenerate	test_catchment_name_from_input_dir	Pass
	test_generator_part1.py::TestGenerate	test_generate_with_empty_data	Pass
	test_generator_part1.py::TestGenerate	test_generate_with_minimal_data	Pass
	test_generator_part1.py::TestGenerate	test_output_path_creates_parent_dirs	Pass
	test_generator_part1.py::TestGenerate	test_pdf_has_multiple_pages	Pass
	test_generator_part1.py::TestGenerate	test_pdf_starts_with_magic_bytes	Pass
	test_generator_part1.py::TestGenerate	test_produces_pdf_file	Pass
	test_generator_part1.py::TestGenerate	test_returns_path	Pass
	test_generator_part1.py::TestInit	test_input_dir_accepts_string	Pass
	test_generator_part1.py::TestInit	test_input_dir_set	Pass
	test_generator_part1.py::TestInit	test_output_path_explicit	Pass
	test_generator_part1.py::TestInit	test_styles_initialised	Pass
	test_generator_part2.py::TestEdgeCases	test_gauge_with_dash_values	Pass
	test_generator_part2.py::TestEdgeCases	test_mortgage_with_none_fields	Pass
	test_generator_part2.py::TestEdgeCases	test_property_with_none_value	Pass
	test_generator_part2.py::TestMixinIntegration	test_make_kv_table	Pass
	test_generator_part2.py::TestMixinIntegration	test_load_all	Pass
	test_generator_part2.pyTestMixinIntegration	test_make_table	Pass
	test_generator_part2.py::TestMixinIntegration	test_section_methods	Pass
	test_generator_part2.py::TestMixinIntegration	test_load_all_and_sections_compatible	Pass
	test_sections_gauges.py::TestSectionsGauges	test_heading	Pass
	test_sections_gauges.py::TestSectionsGauges	test_no_baselines_message	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_sections_gauges.py::TestSections	test_gauge_computed	Pass
	test_sections_gauges.py::TestSections	test_returns_list	Pass
	test_sections_gauges.py::TestSections	test_seasonal_averages	Pass
	test_sections_gauges.py::TestSections	test_table_when_baselines	Pass
	test_sections_gauges.py::TestSections	test_empty_gauges_message	Pass
	test_sections_gauges.py::TestSections	test_multiple_gauges	Pass
	test_sections_gauges.py::TestSections	test_returns_list	Pass
	test_sections_gauges.py::TestSections	test_section_heading	Pass
	test_sections_gauges.py::TestSections	test_summary_line_counts	Pass
	test_sections_gauges.py::TestSections	test_table_present	Pass
	test_sections_gauges.py::TestSections	test_alternative_key_format	Pass
	test_sections_gauges.py::TestSections	test_dict_gauge_curve	Pass
	test_sections_gauges.py::TestSections	test_heading	Pass
	test_sections_gauges.py::TestSections	test_no_curves_message	Pass
	test_sections_gauges.py::TestSections	test_returns_list	Pass
	test_sections_gauges.py::TestSections	test_table_with_curves	Pass
	test_sections_gauges.py::TestSections	test_mortgage_in_summary	Pass
	test_sections_gauges.py::TestSections	test_empty_mortgages_message	Pass
	test_sections_gauges.py::TestSections	test_heading	Pass
	test_sections_gauges.py::TestSections	test_returns_list	Pass
	test_sections_gauges.py::TestSections	test_table_present	Pass
	test_sections_gauges.py::TestSections	test_empty_properties_message	Pass
	test_sections_gauges.py::TestSections	test_heading_present	Pass
	test_sections_gauges.py::TestSections	test_property_count_in_summary	Pass
	test_sections_gauges.py::TestSections	test_returns_list	Pass
	test_sections_gauges.py::TestSections	test_table_with_properties	Pass
	test_sections_gauges.py::TestSections	test_heading	Pass
	test_sections_gauges.py::TestSections	test_table_present	Pass
	test_sections_gauges.py::TestSections	test_returns_list	Pass
	test_sections_gauges.py::TestSections	test_with_full_summary	Pass
	test_sections_portfolio.py::TestSections	test_heading	Pass
	test_sections_portfolio.py::TestSections	test_table	Pass
	test_sections_portfolio.py::TestSections	test_returns_list	Pass
	test_sections_portfolio.py::TestSections	test_alternative_cdm_format	Pass
	test_sections_portfolio.py::TestSections	test_empty_message	Pass
	test_sections_portfolio.py::TestSections	test_heading	Pass
	test_sections_portfolio.py::TestSections	test_returns_list	Pass
	test_sections_portfolio.py::TestSections	test_table_present	Pass
	test_sections_portfolio.py::TestSections	test_heading	Pass
	test_sections_portfolio.py::TestSections	test_no_section_when_empty	Pass
	test_sections_portfolio.py::TestSections	test_section_added	Pass
	test_sections_portfolio.py::TestSections	test_returns_list	Pass
	test_sections_portfolio.py::TestSections	test_data_size_computed	Pass
	test_sections_portfolio.py::TestSections	test_heading	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_sections_portfolio.py::TestSectionSummary	test_summary_present	Pass
	test_sections_portfolio.py::TestSectionSummary	test_returns_list	Pass
	test_styles.py::TestAvailWidth	test_avail_width_value	Pass
	test_styles.py::TestColourConstants	test_blue_is_hex_colour	Pass
	test_styles.py::TestColourConstants	test_dark_blue	Pass
	test_styles.py::TestColourConstants	test_grey	Pass
	test_styles.py::TestColourConstants	test_light_amber	Pass
	test_styles.py::TestColourConstants	test_light_blue	Pass
	test_styles.py::TestColourConstants	test_light_green	Pass
	test_styles.py::TestColourConstants	test_light_red	Pass
	test_styles.py::TestKvTable	test_custom_label_width	Pass
	test_styles.py::TestKvTable	test_default_label_width	Pass
	test_styles.py::TestKvTable	test_multiple_pairs	Pass
	test_styles.py::TestKvTable	test_returns_table	Pass
	test_styles.py::TestKvTable	test_value_converted_to_string	Pass
	test_styles.py::TestMakeTable	test_col_widths_propagated	Pass
	test_styles.py::TestMakeTable	test_custom_alt_color	Pass
	test_styles.py::TestMakeTable	test_default_alt_color_is_blue	Pass
	test_styles.py::TestMakeTable	test_empty_rows	Pass
	test_styles.py::TestMakeTable	test_repeat_rows_set	Pass
	test_styles.py::TestMakeTable	test_returns_table	Pass
	test_styles.py::TestMakeTable	test_table_has_correct_row_count	Pass
	test_styles.py::TestSetupStyles	test_body_style_font_size	Pass
	test_styles.py::TestSetupStyles	test_kv_style_font_size	Pass
	test_styles.py::TestSetupStyles	test_kv_style_left_indent	Pass
	test_styles.py::TestSetupStyles	test_section_style_font_size	Pass
	test_styles.py::TestSetupStyles	test_section_style_text_color	Pass
	test_styles.py::TestSetupStyles	test_subsection_style_font_size	Pass
	test_styles.py::TestSetupStyles	test_subsection_style_text_color	Pass
	test_styles.py::TestSetupStyles	test_subtitle_style_font_size	Pass
	test_styles.py::TestSetupStyles	test_subtitle_style_text_color	Pass
	test_styles.py::TestSetupStyles	test_title_style_font_name	Pass
	test_styles.py::TestSetupStyles	test_title_style_font_size	Pass
	test_styles.py::TestSetupStyles	test_title_style_text_color	Pass
	test_styles.py::TestTableStyleCommands	test_alt_rows_amber	Pass
	test_styles.py::TestTableStyleCommands	test_alt_rows_blue	Pass
	test_styles.py::TestTableStyleCommands	test_alt_rows_green	Pass
	test_styles.py::TestTableStyleCommands	test_hdr_style_first_command_is_background	Pass
	test_styles.py::TestTableStyleCommands	test_hdr_style_has_grid	Pass
	test_styles.py::TestTableStyleCommands	test_hdr_style_is_list	Pass
	analysis.py::TestSinglePropertyAnalysis	test_find_nearest_gauge	Pass
	analysis.py::TestSinglePropertyAnalysis	test_full_analysis_high_risk_property	Pass
	analysis.py::TestSinglePropertyAnalysis	test_full_analysis_low_risk_property	Pass
	analysis.py::TestSinglePropertyAnalysis	test_load_gauge_data	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	analysis.py::TestSinglePropertyAnalysis	test_load_property_data	Pass
	calculations.py::TestDamageRatio	test_commercial_higher_damage	Pass
	calculations.py::TestDamageRatio	test_damage_capped	Pass
	calculations.py::TestDamageRatio	test_damage_increases_with_depth	Pass
	calculations.py::TestDamageRatio	test_no_damage_no_flood	Pass
	calculations.py::TestDistanceCalculation	test_haversine_known_distance	Pass
	calculations.py::TestDistanceCalculation	test_haversine_same_point	Pass
	calculations.py::TestDistanceCalculation	test_haversine_teddington_to_richmond	Pass
	calculations.py::TestFloodDepthCalculation	test_flood_above_trigger	Pass
	calculations.py::TestFloodDepthCalculation	test_flood_attenuates_with_distance	Pass
	calculations.py::TestFloodDepthCalculation	test_high_elevation_no_flood	Pass
	calculations.py::TestFloodDepthCalculation	test_no_flood_below_trigger	Pass
	calculations.py::TestRiskClassification	test_high_risk	Pass
	calculations.py::TestRiskClassification	test_low_risk	Pass
	calculations.py::TestRiskClassification	test_medium_risk	Pass
	calculations.py::TestRiskClassification	test_minimal_risk	Pass
	formatters.py::TestConstants	test_margin_positive	Pass
	formatters.py::TestConstants	test_page_dimensions	Pass
	formatters.py::TestFmtGbp	test_large_value	Pass
	formatters.py::TestFmtGbp	test_negative	Pass
	formatters.py::TestFmtGbp	test_non_numeric_fallback	Pass
	formatters.py::TestFmtGbp	test_none_fallback	Pass
	formatters.py::TestFmtGbp	test_positive_float	Pass
	formatters.py::TestFmtGbp	test_zero	Pass
	formatters.py::TestLayouts	test_body_style_alignment	Pass
	formatters.py::TestLayouts	test_body_style_bold	Pass
	formatters.py::TestLayouts	test_body_style_default	Pass
	formatters.py::TestLayouts	test_build_header_footer_callable	Pass
	formatters.py::TestLayouts	test_white_hdr_style_returns_paragraph_style	Pass
	formatters.py::TestSeqTypeColor	test_case_insensitive	Pass
	formatters.py::TestSeqTypeColor	test_cluster	Pass
	formatters.py::TestSeqTypeColor	test_doublet	Pass
	formatters.py::TestSeqTypeColor	test_isolated	Pass
	formatters.py::TestSeqTypeColor	test_none_returns_white	Pass
	formatters.py::TestSeqTypeColor	test_persistent	Pass
	formatters.py::TestSeqTypeColor	test_unknown_returns_white	Pass
	formatters.py::TestSetupStyles	test_required_keys	Pass
	formatters.py::TestSetupStyles	test_returns_stylesheet	Pass
	formatters.py::TestSetupStyles	test_styles_are_paragraph_styles	Pass
	generator.py::TestClaimReportGenerator	test_claim_ref_contains_prop_id_suffix	Pass
	generator.py::TestClaimReportGenerator	test_generate_empty_sequence_lookup	Pass
	generator.py::TestClaimReportGenerator	test_generate_returns_bytes	Pass
	generator.py::TestClaimReportGenerator	test_generate_without_mortgage	Pass
	generator.py::TestClaimReportGenerator	test_minimal_prop_data	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	generator.py::TestClaimReportGenerator	test_output_is_valid_pdf	Pass
	generator.py::TestClaimReportGenerator	test_pdf_size_reasonable	Pass
	generator.py::TestCompatShim	test_shim_class_is_same	Pass
	generator.py::TestCompatShim	test_shim_imports_generator	Pass
	generator.py::TestPublicAPI	test_all_exports	Pass
	generator.py::TestPublicAPI	test_import_from_package	Pass
	pages1_3.py::TestPage1Cover	test_contains_claim_ref_paragraph	Pass
	pages1_3.py::TestPage1Cover	test_no_flood_events_still_renders	Pass
	pages1_3.py::TestPage1Cover	test_non_float_coordinates	Pass
	pages1_3.py::TestPage1Cover	test_returns_list	Pass
	pages1_3.py::TestPage1Cover	test_stats_reflect_events	Pass
	pages1_3.py::TestPage1Cover	test_without_mortgage	Pass
	pages1_3.py::TestPage2Chronology	test_events_with_no_sequence_id	Pass
	pages1_3.py::TestPage2Chronology	test_no_flood_events	Pass
	pages1_3.py::TestPage2Chronology	test_returns_list	Pass
	pages1_3.py::TestPage2Chronology	test_stats_bar_present	Pass
	pages1_3.py::TestPage2Chronology	test_table_included	Pass
	pages1_3.py::TestPage2Chronology	test_unknown_sequence_id_handled	Pass
	pages1_3.py::TestPage3Damage	test_contains_tables	Pass
	pages1_3.py::TestPage3Damage	test_high_damage_rows_highlighted	Pass
	pages1_3.py::TestPage3Damage	test_multiple_sequences_grouped	Pass
	pages1_3.py::TestPage3Damage	test_no_flood_events	Pass
	pages1_3.py::TestPage3Damage	test_returns_list	Pass
	pages1_3.py::TestPage3Damage	test_zero_property_value	Pass
	pages4_5.py::TestPage4Mortgage	test_contains_tables	Pass
	pages4_5.py::TestPage4Mortgage	test_negative_equity_highlights_row	Pass
	pages4_5.py::TestPage4Mortgage	test_no_flood_events_with_mortgage	Pass
	pages4_5.py::TestPage4Mortgage	test_with_mortgage	Pass
	pages4_5.py::TestPage4Mortgage	test_without_mortgage	Pass
	pages4_5.py::TestPage5Determination	test_contains_reference_table	Pass
	pages4_5.py::TestPage5Determination	test_damage_calculation	Pass
	pages4_5.py::TestPage5Determination	test_negative_equity_flag	Pass
	pages4_5.py::TestPage5Determination	test_returns_list	Pass
	pages4_5.py::TestPage5Determination	test_signature_block_present	Pass
	pages4_5.py::TestPage5Determination	test_without_mortgage	Pass
	page_01_title_overview.py::TestTitleOverviewPage	test_building_age_calculated	Pass
	page_01_title_overview.py::TestTitleOverviewPage	test_empty_property_does_not_crash	Pass
	page_01_title_overview.py::TestTitleOverviewPage	test_get_property_id_missing_nested_key	Pass
	page_01_title_overview.py::TestTitleOverviewPage	test_get_property_id_present	Pass
	page_01_title_overview.py::TestTitleOverviewPage	test_has_tables	Pass
	page_01_title_overview.py::TestTitleOverviewPage	test_has_title_paragraph	Pass
	page_01_title_overview.py::TestTitleOverviewPage	test_property_id_in_output	Pass
	page_01_title_overview.py::TestTitleOverviewPage	test_property_overview_section_header	Pass
	page_01_title_overview.py::TestTitleOverviewPage	test_property_summary_section_header	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	page_01_title_overview.py::TestTitleOverviewPage	test_returns_list	Pass
	page_01_title_overview.py::TestTitleOverviewPage	test_unknown_property_id_fallback	Pass
	page_01_title_overview.py::TestTitleOverviewPage	test_with_mortgage_data_does_not_crash	Pass
	page_02_location.py::TestLocationPage	test_address_subsection_header	Pass
	page_02_location.py::TestLocationPage	test_administrative_subsection_header	Pass
	page_02_location.py::TestLocationPage	test_area_classification_subsection_header	Pass
	page_02_location.py::TestLocationPage	test_density_high	Pass
	page_02_location.py::TestLocationPage	test_density_low	Pass
	page_02_location.py::TestLocationPage	test_density_low_medium	Pass
	page_02_location.py::TestLocationPage	test_density_medium	Pass
	page_02_location.py::TestLocationPage	test_density_very_high	Pass
	page_02_location.py::TestLocationPage	test_empty_property_does_not_crash	Pass
	page_02_location.py::TestLocationPage	test_geographic_subsection_header	Pass
	page_02_location.py::TestLocationPage	test_has_section_header	Pass
	page_02_location.py::TestLocationPage	test_has_tables	Pass
	page_02_location.py::TestLocationPage	test_no_coordinates_skips_geo_rows	Pass
	page_02_location.py::TestLocationPage	test_returns_list	Pass
	page_02_location.py::TestLocationPage	test_with_mortgage_data_does_not_crash	Pass
	page_03_attributes.py::TestAttributesPage	test_contemporary	Pass
	page_03_attributes.py::TestAttributesPage	test_historic	Pass
	page_03_attributes.py::TestAttributesPage	test_mature	Pass
	page_03_attributes.py::TestAttributesPage	test_modern	Pass
	page_03_attributes.py::TestAttributesPage	test_basic_info_subsection	Pass
	page_03_attributes.py::TestAttributesPage	test_building_age_subsection	Pass
	page_03_attributes.py::TestAttributesPage	test_condition_fair_recommendation	Pass
	page_03_attributes.py::TestAttributesPage	test_condition_good_recommendation	Pass
	page_03_attributes.py::TestAttributesPage	test_condition_poor_recommendation	Pass
	page_03_attributes.py::TestAttributesPage	test_condition_unknown_falls_back	Pass
	page_03_attributes.py::TestAttributesPage	test_empty_property_does_not_crash	Pass
	page_03_attributes.py::TestAttributesPage	test_external_features_subsection	Pass
	page_03_attributes.py::TestAttributesPage	test_has_section_header	Pass
	page_03_attributes.py::TestAttributesPage	test_has_tables	Pass
	page_03_attributes.py::TestAttributesPage	test_no_construction_year_skips_age	Pass
	page_03_attributes.py::TestAttributesPage	test_physical_dimensions_subsection	Pass
	page_03_attributes.py::TestAttributesPage	test_returns_list	Pass
	page_03_attributes.py::TestAttributesPage	test_room_composition_subsection	Pass
	page_03_attributes.py::TestAttributesPage	test_with_mortgage_data_does_not_crash	Pass
	page_03_gauge_time_series.py::TestGaugeTimeSeries	test_empty_timeseries_returns_short_list	Pass
	page_03_gauge_time_series.py::TestGaugeTimeSeries	test_gauge_name_in_title	Pass
	page_03_gauge_time_series.py::TestGaugeTimeSeries	test_missing_gauge_name_does_not_crash	Pass
	page_03_gauge_time_series.py::TestGaugeTimeSeries	test_no_timeseries_message_included	Pass
	page_03_gauge_time_series.py::TestGaugeTimeSeries	test_with_timeseries_returns_short_list	Pass
	page_03_gauge_time_series.py::TestGaugeTimeSeries	test_returns_list	Pass
	page_03_gauge_time_series.py::TestGaugeTimeSeries	test_mixed_statuses	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	page_03_gauge_time_series.py::TestGaugeTimeSeries	test_gauge_time_series_with_thresholds	Pass
	page_03_gauge_time_series.py::TestGaugeTimeSeries	test_gauge_time_series_returns_more_elements	Pass
	page_03_gauge_time_series.py::TestGaugeTimeSeries	test_gauge_time_series_table_included	Pass
	page_04_construction.py::TestConstructionPageBasics	test_construction_year_shown	Pass
	page_04_construction.py::TestConstructionPageBasics	test_empty_property_does_not_crash	Pass
	page_04_construction.py::TestConstructionPageBasics	test_floor_levels_shown	Pass
	page_04_construction.py::TestConstructionPageBasics	test_has_section_header	Pass
	page_04_construction.py::TestConstructionPageBasics	test_has_tables	Pass
	page_04_construction.py::TestConstructionPageBasics	test_construction_data_returns_message	Pass
	page_04_construction.py::TestConstructionPageBasics	test_construction_year	Pass
	page_04_construction.py::TestConstructionPageBasics	test_tables_with_data	Pass
	page_04_construction.py::TestConstructionPageBasics	test_site_height_shown	Pass
	page_04_construction.py::TestConstructionTypeAnalysis	test_construction_type	Pass
	page_04_construction.py::TestConstructionTypeAnalysis	test_construction_type_shown	Pass
	page_04_construction.py::TestConstructionTypeAnalysis	test_empty_construction_type	Pass
	page_04_construction.py::TestConstructionTypeAnalysis	test_has_section_header	Pass
	page_04_construction.py::TestConstructionTypeAnalysis	test_section_type	Pass
	page_04_construction.py::TestConstructionTypeAnalysis	test_timber_construction	Pass
	page_04_construction.py::TestConstructionTypeAnalysis	test_unknown_construction_type	Pass
	page_04_construction.py::TestFoundationTypeAnalysis	test_pad_foundation	Pass
	page_04_construction.py::TestFoundationTypeAnalysis	test_pile_foundation	Pass
	page_04_construction.py::TestFoundationTypeAnalysis	test_raft_foundation	Pass
	page_04_construction.py::TestFoundationTypeAnalysis	test_strip_foundation	Pass
	page_04_construction.py::TestFoundationTypeAnalysis	test_unknown_foundation	Pass
	page_04_construction.py::TestHeightClassification	test_medium_height_10_to_15m	Pass
	page_04_construction.py::TestHeightClassification	test_missing_height_skips_classification	Pass
	page_04_construction.py::TestHeightClassification	test_standard_height_under_10m	Pass
	page_04_construction.py::TestHeightClassification	test_tall_structure_over_15m	Pass
	page_05_risk_assessment.py::TestElevationRiskBasics	test_good_elevation_over_20m	Pass
	page_05_risk_assessment.py::TestElevationRiskBasics	test_low_elevation_5_to_10m	Pass
	page_05_risk_assessment.py::TestElevationRiskBasics	test_moderate_elevation_10_to_20m	Pass
	page_05_risk_assessment.py::TestElevationRiskBasics	test_negative_risk_level	Pass
	page_05_risk_assessment.py::TestElevationRiskBasics	test_very_low_elevation_under_5m	Pass
	page_05_risk_assessment.py::TestFloodZoneInformation	test_unknown_zone	Pass
	page_05_risk_assessment.py::TestFloodZoneInformation	test_zone_1_low_risk	Pass
	page_05_risk_assessment.py::TestFloodZoneInformation	test_zone_2_medium_risk	Pass
	page_05_risk_assessment.py::TestFloodZoneInformation	test_zone_3a_high_risk	Pass
	page_05_risk_assessment.py::TestFloodZoneInformation	test_zone_3b_floodplain	Pass
	page_05_risk_assessment.py::TestFloodZoneInformation	test_zone_in_page_output	Pass
	page_05_risk_assessment.py::TestRiskAssessmentPageBasics	test_assessment_shown	Pass
	page_05_risk_assessment.py::TestRiskAssessmentPageBasics	test_empty_property_does_not_crash	Pass
	page_05_risk_assessment.py::TestRiskAssessmentPageBasics	test_flood_risk_type_shown	Pass
	page_05_risk_assessment.py::TestRiskAssessmentPageBasics	test_has_section_header	Pass
	page_05_risk_assessment.py::TestRiskAssessmentPageBasics	test_has_tables	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	page_05_risk_assessment.py::	TestRiskAssessmentPageBasics::test_lake_distance_shown	Pass
	page_05_risk_assessment.py::	TestRiskAssessmentPageBasics::test_last_flood_date_shown	Pass
	page_05_risk_assessment.py::	TestRiskAssessmentPageBasics::test_page_loads	Pass
	page_05_risk_assessment.py::	TestRiskAssessmentPageBasics::test_seismic_zone	Pass
	page_05_risk_assessment.py::	TestRiskAssessmentPageBasics::test_last_flood_date	Pass
	page_05_risk_assessment.py::	TestRiskAssessmentPageBasics::test_summary_tiles	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_clay_soil_adds_1	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_close_river_adds_1	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_high_flood_risk_adds_3	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_high_score_level	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_low_elevation_adds_2	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_low_flood_risk_adds_1	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_medium_score_level	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_low_risk_level	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_medium_flood_risk_adds_2	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_medium_score_level	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_moderate_elevation_adds_1	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_risk_factors_message	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_sand_soil_adds_2	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_return_all_keys	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_very_close_river_adds_2	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_very_high_flood_risk_adds_4	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_very_high_score_level	Pass
	page_05_risk_assessment.py::	TestRiskScoreCalculation::test_very_low_elevation_adds_3	Pass
	page_05_risk_assessment.py::	TestRiverProximity::test_river_close_100_to_500m	Pass
	page_05_risk_assessment.py::	TestRiverProximity::test_river_distance_over_1000m	Pass
	page_05_risk_assessment.py::	TestRiverProximity::test_river_nearby_500_to_1000m	Pass
	page_05_risk_assessment.py::	TestRiverProximity::test_very_close_under_100m	Pass
	page_05_risk_assessment.py::	TestSoilRiskAssessment::test_medium_high	Pass
	page_05_risk_assessment.py::	TestSoilRiskAssessment::test_high_risk	Pass
	page_05_risk_assessment.py::	TestSoilRiskAssessment::test_keyword	Pass
	page_05_risk_assessment.py::	TestSoilRiskAssessment::test_low_risk	Pass
	page_05_risk_assessment.py::	TestSoilRiskAssessment::test_medium_risk	Pass
	page_05_risk_assessment.py::	TestSoilRiskAssessment::test_unknown_soil	Pass
	page_06_financial.py::	TestFinancialPageBasics::test_empty_property_does_not_crash	Pass
	page_06_financial.py::	TestFinancialPageBasics::test_financial_header	Pass
	page_06_financial.py::	TestFinancialPageBasics::test_table	Pass
	page_06_financial.py::	TestFinancialPageBasics::test_area_skips_value_per_sqm	Pass
	page_06_financial.py::	TestFinancialPageBasics::test_property_value	Pass
	page_06_financial.py::	TestFinancialPageBasics::test_valuation_data	Pass
	page_06_financial.py::	TestFinancialPageBasics::test_valuation_field_with_value_in_name	Pass
	page_06_financial.py::	TestFinancialPageBasics::test_valuation_field_without_value_in_name	Pass
	page_06_financial.py::	TestFinancialPageBasics::test_rains_list	Pass
	page_06_financial.py::	TestFinancialPageBasics::test_rains_non_empty_list	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	page_06_financial.py::TestFinancialPage	test_price_per_sqm_shown_when_area_present	Pass
	page_06_financial.py::TestInvestment	test_below_average_2_to_4pct	Pass
	page_06_financial.py::TestInvestment	test_excellent_over_8pct	Pass
	page_06_financial.py::TestInvestment	test_fair_4_to_6pct	Pass
	page_06_financial.py::TestInvestment	test_good_6_to_8pct	Pass
	page_06_financial.py::TestInvestment	test_poor_under_2pct	Pass
	page_06_financial.py::TestMarketSegment	test_entry_level_under_250k	Pass
	page_06_financial.py::TestMarketSegment	test_mid_250k_to_500k	Pass
	page_06_financial.py::TestMarketSegment	test_premium_over_1m	Pass
	page_06_financial.py::TestMarketSegment	test_upper_mid_500k_to_1m	Pass
	page_06_financial.py::TestTransaction	test_monthly_rent_and_annual_rent	Pass
	page_06_financial.py::TestTransaction	test_no_transaction_data	Pass
	page_06_financial.py::TestTransaction	test_rental_yield_computed	Pass
	page_06_financial.py::TestTransaction	test_sale_date_shown	Pass
	page_06_financial.py::TestTransaction	test_sale_price_and_appreciation	Pass
	page_06_financial.py::TestTransaction	test_section_rendered	Pass
	page_07_protection.py::TestProtection	test_all_natural_home_recommendation	Pass
	page_07_protection.py::TestProtection	test_natural_measures_section_rendered	Pass
	page_07_protection.py::TestProtection	test_empty_property_returns_list	Pass
	page_07_protection.py::TestProtection	test_empty_protection_measures_dict_shows_fallback	Pass
	page_07_protection.py::TestProtection	test_protection_measures_shows_fallback	Pass
	page_07_protection.py::TestProtection	test_all_protection_fewer_recommendations	Pass
	page_07_protection.py::TestProtection	test_generate_protection_recommendations_returns_dict	Pass
	page_07_protection.py::TestProtection	test_missing_critical_measures_in_recommendations	Pass
	page_07_protection.py::TestProtection	test_recommendations_all_natural_false	Pass
	page_07_protection.py::TestProtection	test_recommendations_flood_re_eligible	Pass
	page_07_protection.py::TestProtection	test_recommendations_missing_critical	Pass
	page_07_protection.py::TestProtection	test_recommendations_section_always_rendered	Pass
	page_07_protection.py::TestProtection	test_all_installed_excellent_rating	Pass
	page_07_protection.py::TestProtection	test_building_measures_listed	Pass
	page_07_protection.py::TestProtection	test_average_excellent_80pct	Pass
	page_07_protection.py::TestProtection	test_average_fair_40pct	Pass
	page_07_protection.py::TestProtection	test_average_good_60pct	Pass
	page_07_protection.py::TestProtection	test_average_limited_20pct	Pass
	page_07_protection.py::TestProtection	test_average_poor_0pct	Pass
	page_07_protection.py::TestProtection	test_critical_measures_listed	Pass
	page_07_protection.py::TestProtection	test_emergency_measures_listed	Pass
	page_07_protection.py::TestProtection	test_flood_re_measure_still_shown	Pass
	page_07_protection.py::TestProtection	test_excess_amounts_shown	Pass
	page_07_protection.py::TestProtection	test_insurance_premium_shown	Pass
	page_07_protection.py::TestProtection	test_other_risk_fields_shown	Pass
	page_07_protection.py::TestProtection	test_recommendations_include_flood_re	Pass
	page_07_protection.py::TestProtection	test_risk_assessment_section_rendered	Pass
	page_08_energy.py::TestEnergyPage	test_analyze_epc_rating_all_bands	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	page_08_energy.py::TestEnergyPage	test_analyze_epc_rating_case_insensitive	Pass
	page_08_energy.py::TestEnergyPage	test_analyze_epc_rating_unknown	Pass
	page_08_energy.py::TestEnergyPage	test_energy_intensity_calculation	Pass
	page_08_energy.py::TestEnergyPage	test_full_data_no_crash	Pass
	page_08_energy.py::TestEnergyPage	test_no_energy_data_fallback_message	Pass
	page_08_energy.py::TestEnergyPage	test_returns_list	Pass
	page_08_energy.py::TestEnergyPage	test_with_building_fabric	Pass
	page_08_energy.py::TestEnergyPage	test_with_energy_usage	Pass
	page_08_energy.py::TestEnergyPage	test_with_ratings_data	Pass
	page_08_energy.py::TestEnergyPage	test_with_ratings_includes_table	Pass
	page_09_history_part1.py::TestHistoryPage	test_assess_historical_risks_active_subsidence	Pass
	page_09_history_part1.py::TestHistoryPage	test_assess_historical_risks_high_env	Pass
	page_09_history_part1.py::TestHistoryPage	test_assess_historical_risks_low_env	Pass
	page_09_history_part1.py::TestHistoryPage	test_assess_historical_risks_minor_subsidence	Pass
	page_09_history_part1.py::TestHistoryPage	test_assess_historical_risks_moderate_env	Pass
	page_09_history_part1.py::TestHistoryPage	test_assess_historical_risks_moderate_flood	Pass
	page_09_history_part1.py::TestHistoryPage	test_assess_historical_risks_no_flood_damage	Pass
	page_09_history_part1.py::TestHistoryPage	test_assess_historical_risks_severe_flood	Pass
	page_09_history_part1.py::TestHistoryPage	test_assess_historical_risks_stable_ground	Pass
	page_09_history_part1.py::TestHistoryPage	test_build_flood_history_bad_json	Pass
	page_09_history_part1.py::TestHistoryPage	test_build_flood_history_no_flood_days	Pass
	page_09_history_part1.py::TestHistoryPage	test_build_flood_history_no_hd_file	Pass
	page_09_history_part1.py::TestHistoryPage	test_build_flood_history_no_reference_gauges	Pass
	page_09_history_part1.py::TestHistoryPage	test_build_flood_history_with_flood_data	Pass
	page_09_history_part1.py::TestHistoryPage	test_empty_flood_events	Pass
	page_09_history_part1.py::TestHistoryPage	test_empty_property_does_not_crash	Pass
	page_09_history_part1.py::TestHistoryPage	test_has_history_header	Pass
	page_09_history_part1.py::TestHistoryPage	test_no_flood_history	Pass
	page_09_history_part1.py::TestHistoryPage	test_returns_list	Pass
	page_09_history_part1.py::TestHistoryPage	test_with_environmental_issues	Pass
	page_09_history_part1.py::TestHistoryPage	test_with_fire_incidents	Pass
	page_09_history_part1.py::TestHistoryPage	test_with_ground_conditions	Pass
	page_09_history_part1.py::TestHistoryPage	test_with_mortgage_data	Pass
	page_10_transactions.py::TestTransactionsPage	test_assess_yield_average	Pass
	page_10_transactions.py::TestTransactionsPage	test_assess_yield_below_average	Pass
	page_10_transactions.py::TestTransactionsPage	test_assess_yield_excellent	Pass
	page_10_transactions.py::TestTransactionsPage	test_assess_yield_good	Pass
	page_10_transactions.py::TestTransactionsPage	test_assess_yield_poor	Pass
	page_10_transactions.py::TestTransactionsPage	test_empty_property_does_not_crash	Pass
	page_10_transactions.py::TestTransactionsPage	test_empty_transactions	Pass
	page_10_transactions.py::TestTransactionsPage	test_has_transactions_header	Pass
	page_10_transactions.py::TestTransactionsPage	test_market_analysis_appreciation	Pass
	page_10_transactions.py::TestTransactionsPage	test_market_analysis_few_vacancies	Pass
	page_10_transactions.py::TestTransactionsPage	test_market_analysis_many_vacancies	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	page_10_transactions.py::TestTransactionsPage	test_market_analysis_marketing_above_30	Pass
	page_10_transactions.py::TestTransactionsPage	test_market_analysis_marketing_above_90	Pass
	page_10_transactions.py::TestTransactionsPage	test_market_analysis_never_rented	Pass
	page_10_transactions.py::TestTransactionsPage	test_market_analysis_no_vacancy	Pass
	page_10_transactions.py::TestTransactionsPage	test_market_analysis_yield	Pass
	page_10_transactions.py::TestTransactionsPage	test_transaction_history	Pass
	page_10_transactions.py::TestTransactionsPage	test_transactions_list	Pass
	page_10_transactions.py::TestTransactionsPage	test_sales_extended_marketing	Pass
	page_10_transactions.py::TestTransactionsPage	test_sales_quick_sale	Pass
	page_10_transactions.py::TestTransactionsPage	test_sales_very_long_marketing	Pass
	page_10_transactions.py::TestTransactionsPage	test_with_mortgage_data	Pass
	page_10_transactions.py::TestTransactionsPage	test_with_rental_data	Pass
	page_10_transactions.py::TestTransactionsPage	test_with_sales_data	Pass
	page_11_mortgage_overview.py::TestMortgageOverviewPage	test_application_subsection	Pass
	page_11_mortgage_overview.py::TestMortgageOverviewPage	test_mortgage_overview_hidden_when_default	Pass
	page_11_mortgage_overview.py::TestMortgageOverviewPage	test_mortgage_overview_shown_when_not_default	Pass
	page_11_mortgage_overview.py::TestMortgageOverviewPage	test_empty_mortgage_data_does_not_crash	Pass
	page_11_mortgage_overview.py::TestMortgageOverviewPage	test_flag_mortgage_data_accepted	Pass
	page_11_mortgage_overview.py::TestMortgageOverviewPage	test_has_tabs	Pass
	page_11_mortgage_overview.py::TestMortgageOverviewPage	test_key_financial_terms_subsection	Pass
	page_11_mortgage_overview.py::TestMortgageOverviewPage	test_mortgage_overview_from_loan_and_purchase	Pass
	page_11_mortgage_overview.py::TestMortgageOverviewPage	test_mortgage_features_subsection	Pass
	page_11_mortgage_overview.py::TestMortgageOverviewPage	test_mortgage_refinancing_subsection	Pass
	page_11_mortgage_overview.py::TestMortgageOverviewPage	test_no_mortgage_data_returns_message	Pass
	page_11_mortgage_overview.py::TestMortgageOverviewPage	test_no_mortgage_data_returns_message	Pass
	page_11_mortgage_overview.py::TestMortgageOverviewPage	test_returns_overview_with_data	Pass
	page_11_mortgage_overview.py::TestMortgageOverviewPage	test_section_header_present	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_active_tab_field	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_borrower_terms_subsection	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_high_risk	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_low_risk	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_medium_high_risk	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_medium_risk	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_subsection	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_tabs_present	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_interest_rates_subsection	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_loan_percentage_of_purchase_calculated	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_high_risk	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_low_medium_risk	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_low_risk	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_medium_risk	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_subsection	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_very_high_risk	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_no_financial_terms_returns_message	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_mortgage_data_returns_message	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_mortgage_details_list_with_data	Pass
	page_11a_mortgage_details.py::TestMortgageDetailsPage	test_section_header_present	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_cost_analysis_subsection	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_cost_high_burden	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_cost_low_burden	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_cost_low_medium_burden	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_cost_medium_burden	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_cost_medium_high_burden	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_discount_points_analysis	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_empty_mortgage_dict_does_not_crash	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_has_tables	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_lender_credits_reduces_net_cost	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_loan_terms_returns_message	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_mortgage_data_returns_message	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_origination_setup_subsection	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_penalties_subsection	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_terms_list_with_data	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_section_header_present	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_total_costs_as_percentage_of_loan	Pass
	page_11b_mortgage_costs.py::TestMortgageCostsPage	test_total_costs_overview_subsection	Pass
	page_11c_regulatory.py::TestCombinedSections	test_all_sections_together	Pass
	page_11c_regulatory.py::TestCombinedSections	test_exception_handled_gracefully	Pass
	page_11c_regulatory.py::TestCombinedSections	test_flat_mortgage_structure	Pass
	page_11c_regulatory.py::TestCommonAdvised	test_advised_flag	Pass
	page_11c_regulatory.py::TestCommonBusinessPurpose	test_business_purpose_false	Pass
	page_11c_regulatory.py::TestCommonBusinessPurpose	test_business_purpose_true	Pass
	page_11c_regulatory.py::TestCommonCombustion	test_combustion_section_returns_elements	Pass
	page_11c_regulatory.py::TestCommonReference	test_reference_number	Pass
	page_11c_regulatory.py::TestCommonMultiple	test_multiple_regulatory_flags	Pass
	page_11c_regulatory.py::TestHmdaSection	test_hmda_minimal_fields	Pass
	page_11c_regulatory.py::TestHmdaSection	test_hmda_rate_spread_formatted	Pass
	page_11c_regulatory.py::TestHmdaSection	test_hmda_section_creates_table	Pass
	page_11c_regulatory.py::TestMcobSection	test_mcob_advice_rejection_reason	Pass
	page_11c_regulatory.py::TestMcobSection	test_mcob_all_fields_returns_list	Pass
	page_11c_regulatory.py::TestMcobSection	test_mcob_section_creates_tables	Pass
	page_11c_regulatory.py::TestMcobSection	test_mcob_with_minimal_fields	Pass
	page_11c_regulatory.py::TestNoMortgageData	test_no_mortgage_message	Pass
	page_11c_regulatory.py::TestNoMortgageData	test_no_mortgage_returns_list	Pass
	page_11c_regulatory.py::TestNoRegulatoryData	test_no_regulatory_returns_message	Pass
	page_11c_regulatory.py::TestNoRegulatoryData	test_returns_list	Pass
	page_12_current_status.py::TestCurrentStatusPage	test_base_page_basics	Pass
	page_12_current_status.py::TestCurrentStatusPage	test_current_payment_shown	Pass
	page_12_current_status.py::TestCurrentStatusPage	test_has_status_header	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	page_12_current_status.py::TestCurrentStatusPageBasics	test_has_table	Pass
	page_12_current_status.py::TestCurrentStatusPageBasics	test_interest_paid_shown	Pass
	page_12_current_status.py::TestCurrentStatusPageBasics	test_last_payment_date_shown	Pass
	page_12_current_status.py::TestCurrentStatusPageBasics	test_loan_status_shown	Pass
	page_12_current_status.py::TestCurrentStatusPageBasics	test_loan_type_shown	Pass
	page_12_current_status.py::TestCurrentStatusPageBasics	test_mortgage_returns_message	Pass
	page_12_current_status.py::TestCurrentStatusPageBasics	test_mortgage_returns_message	Pass
	page_12_current_status.py::TestCurrentStatusPageBasics	test_outstanding_balance_shown	Pass
	page_12_current_status.py::TestCurrentStatusPageBasics	test_principal_paid_shown	Pass
	page_12_current_status.py::TestCurrentStatusPageBasics	test_total_payments_shown	Pass
	page_12_current_status.py::TestCurrentStatusPageBasics	test_total_mortgage_returns_list	Pass
	page_12_current_status.py::TestLTVBranches	test_ltv_as_percentage_not_decimal	Pass
	page_12_current_status.py::TestLTVBranches	test_ltv_over_60_low_medium	Pass
	page_12_current_status.py::TestLTVBranches	test_ltv_over_70_medium	Pass
	page_12_current_status.py::TestLTVBranches	test_ltv_over_80_medium_high	Pass
	page_12_current_status.py::TestLTVBranches	test_ltv_over_90_high_risk	Pass
	page_12_current_status.py::TestLTVBranches	test_ltv_under_60_low_risk	Pass
	page_12_current_status.py::TestPaymentHolidays	test_all_holiday_fields_shown	Pass
	page_12_current_status.py::TestPaymentHolidays	test_holidays_taken_shown	Pass
	page_12_current_status.py::TestPaymentHolidays	test_last_holiday_date_shown	Pass
	page_12_current_status.py::TestPaymentHolidays	test_no_holiday_data_no_section	Pass
	page_12_current_status.py::TestPaymentHolidays	test_total_holidays_shown	Pass
	page_12_current_status.py::TestPaymentPerformance	test_arrears_balance_shown	Pass
	page_12_current_status.py::TestPaymentPerformance	test_five_missed_payments_poor	Pass
	page_12_current_status.py::TestPaymentPerformance	test_four_missed_payments_concerning	Pass
	page_12_current_status.py::TestPaymentPerformance	test_highest_arrears_shown	Pass
	page_12_current_status.py::TestPaymentPerformance	test_missed_payments_fields_present	Pass
	page_12_current_status.py::TestPaymentPerformance	test_one_missed_payment_good	Pass
	page_12_current_status.py::TestPaymentPerformance	test_three_missed_payments_concerning	Pass
	page_12_current_status.py::TestPaymentPerformance	test_two_missed_payments_good	Pass
	page_12_current_status.py::TestPaymentPerformance	test_zero_missed_payments_excellent	Pass
	page_13_risk_analysis_part1.py::TestCreditRiskBranches	test_excellent_credit_800_plus	Pass
	page_13_risk_analysis_part1.py::TestCreditRiskBranches	test_fair_credit_580_to_669	Pass
	page_13_risk_analysis_part1.py::TestCreditRiskBranches	test_good_credit_670_to_739	Pass
	page_13_risk_analysis_part1.py::TestCreditRiskBranches	test_poor_credit_under_580	Pass
	page_13_risk_analysis_part1.py::TestCreditRiskBranches	test_very_good_credit_740_to_799	Pass
	page_13_risk_analysis_part1.py::TestLTVRiskBranches	test_high_ltv_90_to_95	Pass
	page_13_risk_analysis_part1.py::TestLTVRiskBranches	test_low_ltv_under_70	Pass
	page_13_risk_analysis_part1.py::TestLTVRiskBranches	test_low_ltv_70_to_80	Pass
	page_13_risk_analysis_part1.py::TestLTVRiskBranches	test_medium_ltv_80_to_90	Pass
	page_13_risk_analysis_part1.py::TestLTVRiskBranches	test_very_high_ltv_over_95	Pass
	page_13_risk_analysis_part1.py::TestPaymentPerformanceBranches	test_payment_perf_score	Pass
	page_13_risk_analysis_part1.py::TestPaymentPerformanceBranches	test_many_missed_payments_over_3	Pass
	page_13_risk_analysis_part1.py::TestPaymentPerformanceBranches	test_no_missed_payments_good	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	page_13_risk_analysis_part1.py::	TestPaymentPerformance_no_payments_to_3	Pass
	page_13_risk_analysis_part1.py::	TestRiskAnalysisPageBasics_with_mortgage	Pass
	page_13_risk_analysis_part1.py::	TestRiskAnalysisPageBasics_empty_property_does_not_crash	Pass
	page_13_risk_analysis_part1.py::	TestRiskAnalysisPageBasics	Pass
	page_13_risk_analysis_part1.py::	TestRiskAnalysisPageBasics	Pass
	page_13_risk_analysis_part1.py::	TestRiskAnalysisPageBasics	Pass
	page_13_risk_analysis_part1.py::	TestRiskAnalysisPageBasics	Pass
	page_13_risk_analysis_part2.py::	TestDTIBrackets_under_28	Pass
	page_13_risk_analysis_part2.py::	TestDTIBrackets_36_to_43	Pass
	page_13_risk_analysis_part2.py::	TestDTIBrackets_28_to_36	Pass
	page_13_risk_analysis_part2.py::	TestDTIBrackets_payment_dti_unknown	Pass
	page_13_risk_analysis_part2.py::	TestDTIBrackets_very_high_dti_over_43	Pass
	page_13_risk_analysis_part2.py::	TestGenerateMethods_monitoring_high_score	Pass
	page_13_risk_analysis_part2.py::	TestGenerateMethods_monitoring_low_score	Pass
	page_13_risk_analysis_part2.py::	TestGenerateMethods_monitoring_medium_score	Pass
	page_13_risk_analysis_part2.py::	TestGenerateMethods_recommendations_high_score	Pass
	page_13_risk_analysis_part2.py::	TestGenerateMethods_recommendations_low_score	Pass
	page_13_risk_analysis_part2.py::	TestGenerateMethods_recommendations_medium_score	Pass
	page_13_risk_analysis_part2.py::	TestIdentifyKeyFactors_high_risk	Pass
	page_13_risk_analysis_part2.py::	TestIdentifyKeyFactors_medium_risk	Pass
	page_13_risk_analysis_part2.py::	TestInterestRateBrackets_5_to_7	Pass
	page_13_risk_analysis_part2.py::	TestInterestRateBrackets_over_7	Pass
	page_13_risk_analysis_part2.py::	TestInterestRateBrackets_under_3	Pass
	page_13_risk_analysis_part2.py::	TestInterestRateBrackets_3_to_5	Pass
	page_13_risk_analysis_part2.py::	TestInterestRateBrackets_unknown	Pass
	page_13_risk_analysis_part2.py::	TestOverallRiskLevels_critical_risk_scenario	Pass
	page_13_risk_analysis_part2.py::	TestOverallRiskLevels_low_risk_scenario	Pass
	page_13_risk_analysis_part2.py::	TestOverallRiskLevels_property_only_high_flood_risk	Pass
	page_13_risk_analysis_part2.py::	TestOverallRiskLevels_property_only_very_low_flood_risk	Pass
	page_14_borrower_profile.py::	TestAgeRiskBrackets_approaching_retirement_over_55	Pass
	page_14_borrower_profile.py::	TestAgeRiskBrackets_early_career_under_25	Pass
	page_14_borrower_profile.py::	TestAgeRiskBrackets_prime_years_25_to_55	Pass
	page_14_borrower_profile.py::	TestAgeRiskBrackets_approaching_retirement	Pass
	page_14_borrower_profile.py::	TestAgeRiskBrackets_early_career	Pass
	page_14_borrower_profile.py::	TestAgeRiskBrackets_prime_years	Pass
	page_14_borrower_profile.py::	TestCreditRiskBrackets_high_for_poor_score	Pass
	page_14_borrower_profile.py::	TestCreditRiskBrackets_low_for_high_score	Pass
	page_14_borrower_profile.py::	TestCreditRiskBrackets_low_medium_for_good_score	Pass
	page_14_borrower_profile.py::	TestCreditRiskBrackets_medium_for_fair_score	Pass
	page_14_borrower_profile.py::	TestEmploymentRiskBrackets_established_2_to_5	Pass
	page_14_borrower_profile.py::	TestEmploymentRiskBrackets_recent_under_2	Pass
	page_14_borrower_profile.py::	TestEmploymentRiskBrackets_stable_5_plus	Pass
	page_14_borrower_profile.py::	TestIncomeRiskBrackets_low_for_high_income	Pass
	page_14_borrower_profile.py::	TestIncomeRiskBrackets_low_medium_adequate	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	page_14_borrower_profile.py::TestAssessBorrowerRisk	test_medium_limited	Pass
	page_14_borrower_profile.py::TestAssessBorrowerRisk	test_returns_dict	Pass
	page_14_borrower_profile.py::TestBorrowerProfilePageBasics	test_family_members_shown	Pass
	page_14_borrower_profile.py::TestBorrowerProfilePageBasics	test_header_breadcrumb	Pass
	page_14_borrower_profile.py::TestBorrowerProfilePageBasics	test_income_profile	Pass
	page_14_borrower_profile.py::TestBorrowerProfilePageBasics	test_income_verification_shown	Pass
	page_14_borrower_profile.py::TestBorrowerProfilePageBasics	test_marital_status_shown	Pass
	page_14_borrower_profile.py::TestBorrowerProfilePageBasics	test_mortgage_shown	Pass
	page_14_borrower_profile.py::TestBorrowerProfilePageBasics	test_not_found_returns_early	Pass
	page_14_borrower_profile.py::TestBorrowerProfilePageBasics	test_not_found_shows_message	Pass
	page_14_borrower_profile.py::TestBorrowerProfilePageBasics	test_residency_status_shown	Pass
	page_14_borrower_profile.py::TestBorrowerProfilePageBasics	test_secondary_income_shown	Pass
	page_14_borrower_profile.py::TestBorrowerProfilePageBasics	test_with_mortgage_returns_list	Pass
	page_14_borrower_profile.py::TestCreditScoreBrackets	test_excellent_800_plus	Pass
	page_14_borrower_profile.py::TestCreditScoreBrackets	test_fair_580_to_669	Pass
	page_14_borrower_profile.py::TestCreditScoreBrackets	test_good_670_to_739	Pass
	page_14_borrower_profile.py::TestCreditScoreBrackets	test_poor_under_580	Pass
	page_14_borrower_profile.py::TestCreditScoreBrackets	test_very_good_740_to_799	Pass
	page_14_borrower_profile.py::TestEmploymentStability	test_employment_stability_1_year	Pass
	page_14_borrower_profile.py::TestEmploymentStability	test_employment_stability_5_plus_years	Pass
	page_14_borrower_profile.py::TestEmploymentStability	test_fair_1_to_2_years	Pass
	page_14_borrower_profile.py::TestEmploymentStability	test_good_2_to_5_years	Pass
	page_14_borrower_profile.py::TestIncomeAnalysis	test_fair_income_30k_to_50k	Pass
	page_14_borrower_profile.py::TestIncomeAnalysis	test_good_income_50k_to_100k	Pass
	page_14_borrower_profile.py::TestIncomeAnalysis	test_high_income_over_100k	Pass
	page_14_borrower_profile.py::TestIncomeAnalysis	test_limited_income_under_30k	Pass
	page_14_borrower_profile.py::TestIncomeAnalysis	test_total_income_with_secondary	Pass
	page_15_data_summary.py::TestAnalyzeDataCompleteness	test_completeness_percentage_computed	Pass
	page_15_data_summary.py::TestAnalyzeDataCompleteness	test_empty_section_excluded	Pass
	page_15_data_summary.py::TestAnalyzeDataCompleteness	test_mortgage_data_included	Pass
	page_15_data_summary.py::TestAnalyzeDataCompleteness	test_property_header_section_present	Pass
	page_15_data_summary.py::TestAnalyzeDataCompleteness	test_returns_dict	Pass
	page_15_data_summary.py::TestAnalyzeDataCompleteness	test_section_not_present_excluded	Pass
	page_15_data_summary.py::TestAssessDataQuality	test_best_and_worst_sections_shown	Pass
	page_15_data_summary.py::TestAssessDataQuality	test_completeness_high	Pass
	page_15_data_summary.py::TestAssessDataQuality	test_completeness_low	Pass
	page_15_data_summary.py::TestAssessDataQuality	test_completeness_medium	Pass
	page_15_data_summary.py::TestAssessDataQuality	test_excellent_over_90	Pass
	page_15_data_summary.py::TestAssessDataQuality	test_fair_60_to_75	Pass
	page_15_data_summary.py::TestAssessDataQuality	test_good_75_to_90	Pass
	page_15_data_summary.py::TestAssessDataQuality	test_limited_40_to_60	Pass
	page_15_data_summary.py::TestAssessDataQuality	test_poor_under_40	Pass
	page_15_data_summary.py::TestDataSummaryPageBasics	test_summary_page	Pass
	page_15_data_summary.py::TestDataSummaryPageBasics	test_summary_page_does_not_crash	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	page_15_data_summary.py::TestDataSummaryPageBasics	test_has_data_key	Pass
	page_15_data_summary.py::TestDataSummaryPageBasics	test_has_total	Pass
	page_15_data_summary.py::TestDataSummaryPageBasics	test_no_extra_top_level_property	Pass
	page_15_data_summary.py::TestDataSummaryPageBasics	test_returns_200	Pass
	page_15_data_summary.py::TestDataSummaryPageBasics	test_returns_non_empty	Pass
	page_15_data_summary.py::TestDataSummaryPageBasics	test_with_mortgage_data	Pass
	page_15_data_summary.py::TestGenerateDatasetRecommendations	test_fail_dataset_common_50_to_75_gets_enhancement_rec	Pass
	page_15_data_summary.py::TestGenerateDatasetRecommendations	test_good_dataset_consumer_loans_recommendations	Pass
	page_15_data_summary.py::TestGenerateDatasetRecommendations	test_max_5_recommendations	Pass
	page_15_data_summary.py::TestGenerateDatasetRecommendations	test_pool_dataset_common_under_50_gets_recommendation	Pass
	page_15_data_summary.py::TestGenerateDatasetRecommendations	test_returns_list	Pass
	page_15_data_summary.py::TestGenerateReportMetadata	test_has_report_generated_timestamp	Pass
	page_15_data_summary.py::TestGenerateReportMetadata	test_last_updated_extracted	Pass
	page_15_data_summary.py::TestGenerateReportMetadata	test_mortgage_id_extracted	Pass
	page_15_data_summary.py::TestGenerateReportMetadata	test_property_id_extracted	Pass
	page_15_data_summary.py::TestGenerateReportMetadata	test_returns_dict	Pass
	page_15_data_summary.py::TestGenerateReportMetadata	test_scope_combined_with_mortgage	Pass
	page_15_data_summary.py::TestGenerateReportMetadata	test_scope_property_only	Pass
	page_15_data_summary.py::TestGenerateReportMetadata	test_unknown_property_id_when_missing	Pass
	page_15_data_summary.py::TestGenerateReportMetadata	test_uppercasing_extracted	Pass
	cli.py::TestErrorHandling	test_bad_property_file_exits_with_1	Pass
	cli.py::TestErrorHandling	test_invalid_json_exits_with_1	Pass
	cli.py::TestListCategories	test_list_categories_does_not_open_property_file	Pass
	cli.py::TestListCategories	test_list_categories_runs_without_error	Pass
	cli.py::TestListPages	test_list_pages_does_not_open_property_file	Pass
	cli.py::TestListPages	test_list_pages_runs_without_error	Pass
	cli.py::TestOptionalLoading	test_mortgage_file_is_loaded	Pass
	cli.py::TestOptionalLoading	test_property_id_filters_without_mortgage	Pass
	cli.py::TestOptionalLoading	test_property_id_with_mortgage_file	Pass
	cli.py::TestReportGeneration	test_else_branch_generates_report	Pass
	cli.py::TestReportGeneration	test_report_type_mortgage_focused_with_mortgage	Pass
	cli.py::TestReportGeneration	test_report_type_mortgage_focused_without_mortgage_falls_throu	Pass
	cli.py::TestReportGeneration	test_report_type_property_only	Pass
	cli.py::TestReportGeneration	test_report_type_risk_focused	Pass
	cli.py::TestReportGeneration	test_success_logging_executes	Pass
	convenience.py::TestGenerateProperty	test_auto_open_exception_does_not_raise	Pass
	convenience.py::TestGenerateProperty	test_auto_open_false_skips_open	Pass
	convenience.py::TestGenerateProperty	test_auto_open_true_calls_open	Pass
	convenience.py::TestGenerateProperty	test_bad_report_type_falls_back_to_generate_report	Pass
	convenience.py::TestGenerateProperty	test_mortgage_focused_type_with_mortgage	Pass
	convenience.py::TestGenerateProperty	test_mortgage_focused_type_without_mortgage_falls_back	Pass
	convenience.py::TestGenerateProperty	test_property_only_type	Pass
	convenience.py::TestGenerateProperty	test_reports_path_convenience_function	Pass
	convenience.py::TestGenerateProperty	test_risk_focused_type	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	convenience.py::TestGenerateProperty	test_unknown_type_falls_back_to_generate_report	Pass
	elements.py::TestGenerateElements	test_empty_page_list	Pass
	elements.py::TestGenerateElements	test_exception_in_page_module_continues	Pass
	elements.py::TestGenerateElements	test_first_page_no_page_break	Pass
	elements.py::TestGenerateElements	test_mixed_valid_and_unknown_pages	Pass
	elements.py::TestGenerateElements	test_mortgage_data_passed_to_page	Pass
	elements.py::TestGenerateElements	test_multiple_pages_have_page_breaks	Pass
	elements.py::TestGenerateElements	test_returns_list	Pass
	elements.py::TestGenerateElements	test_unknown_page_skipped	Pass
	elements.py::TestGenerateReport	test_all_unknown_pages_still_creates_pdf	Pass
	elements.py::TestGenerateReport	test_auto_select_called_when_pages_none	Pass
	elements.py::TestGenerateReport	test_auto_select_not_called_when_pages_provided	Pass
	elements.py::TestGenerateReport	test_custom_output_filename	Pass
	elements.py::TestGenerateReport	test_custom_pages_to_include	Pass
	elements.py::TestGenerateReport	test_empty_property_data	Pass
	elements.py::TestGenerateReport	test_full_data_report	Pass
	elements.py::TestGenerateReport	test_pdf_exists	Pass
	elements.py::TestGenerateReport	test_pdf_non_empty	Pass
	elements.py::TestGenerateReport	test_pdf_suffix	Pass
	elements.py::TestGenerateReport	test_returns_path	Pass
	elements.py::TestGenerateReport	test_with_mortgage_data	Pass
	finders.py::TestEdgeCases	test_completely_empty_property_data	Pass
	finders.py::TestEdgeCases	test_negative_value_property	Pass
	finders.py::TestEdgeCases	test_pdf_content_is_bytes	Pass
	finders.py::TestEdgeCases	test_property_data_with_none_header	Pass
	finders.py::TestEdgeCases	test_report_with_all_pages	Pass
	finders.py::TestEdgeCases	test_zero_value_property	Pass
	finders.py::TestFindMortgageByPropertyId	test_dict_with_mortgages_found	Pass
	finders.py::TestFindMortgageByPropertyId	test_dict_with_mortgages_not_found_returns_none	Pass
	finders.py::TestFindMortgageByPropertyId	test_dict_without_mortgages_key_returns_data_itself	Pass
	finders.py::TestFindMortgageByPropertyId	test_empty_mortgages_list_returns_none	Pass
	finders.py::TestFindMortgageByPropertyId	test_invalid_type_returns_data_as_is	Pass
	finders.py::TestFindMortgageByPropertyId	test_list_input_found	Pass
	finders.py::TestFindMortgageByPropertyId	test_list_input_not_found_returns_none	Pass
	finders.py::TestFindPropertyById	test_dict_without_list_key_wraps_itself	Pass
	finders.py::TestFindPropertyById	test_empty_properties_list_raises	Pass
	finders.py::TestFindPropertyById	test_invalid_type_raises_value_error	Pass
	finders.py::TestFindPropertyById	test_list_input_found	Pass
	finders.py::TestFindPropertyById	test_list_not_found_raises	Pass
	finders.py::TestFindPropertyById	test_none_raises_value_error	Pass
	finders.py::TestFindPropertyById	test_not_found_raises_value_error	Pass
	finders.py::TestFindPropertyById	test_portfolio_key_found	Pass
	finders.py::TestFindPropertyById	test_properties_key_found	Pass
	finders.py::TestOpenPdfFile	test_darwin_success	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	finders.py::TestOpenPdfFile	test_generic_exception_returns_false	Pass
	finders.py::TestOpenPdfFile	test_linux_success	Pass
	finders.py::TestOpenPdfFile	test_subprocess_called_process_error_returns_false	Pass
	finders.py::TestOpenPdfFile	test_unknown_platform_uses_webbrowser	Pass
	finders.py::TestOpenPdfFile	test_windows_case_insensitive	Pass
	finders.py::TestOpenPdfFile	test_windows_success	Pass
	init.py::TestAutoSelectPages	test_always_includes_analysis_pages	Pass
	init.py::TestAutoSelectPages	test_always_includes_property_pages	Pass
	init.py::TestAutoSelectPages	test_empty_property_data	Pass
	init.py::TestAutoSelectPages	test_with_mortgage_includes_mortgage_pages	Pass
	init.py::TestAutoSelectPages	test_without_mortgage_excludes_mortgage_pages	Pass
	init.py::TestGenerateFilename	test_contains_property_id	Pass
	init.py::TestGenerateFilename	test_ends_with_pdf	Pass
	init.py::TestGenerateFilename	test_missing_property_id_uses_unknown	Pass
	init.py::TestGenerateFilename	test_partial_path_missing_header	Pass
	init.py::TestGenerateFilename	test_partial_path_missing_property_id	Pass
	init.py::TestGenerateFilename	test_returns_string	Pass
	init.py::TestGenerateFilename	test_timestamp_in_filename	Pass
	init.py::TestPropertyReportGenerator	test_categories_dict_initialized	Pass
	init.py::TestPropertyReportGenerator	test_explicit_output_dir_path	Pass
	init.py::TestPropertyReportGenerator	test_explicit_output_dir_string	Pass
	init.py::TestPropertyReportGenerator	test_output_dir_uses_config	Pass
	init.py::TestPropertyReportGenerator	test_output_dir_created	Pass
	init.py::TestPropertyReportGenerator	test_property_pages_dict_initialized	Pass
	specialised.py::TestHeaderFooter	test_header_footer_callable	Pass
	specialised.py::TestHeaderFooter	test_header_footer_draws_confidential	Pass
	specialised.py::TestSpecialisedReportMethods	test_generate_mortgage_focused_report_custom_filename	Pass
	specialised.py::TestSpecialisedReportMethods	test_generate_mortgage_focused_report_exists	Pass
	specialised.py::TestSpecialisedReportMethods	test_generate_property_only_report_custom_filename	Pass
	specialised.py::TestSpecialisedReportMethods	test_generate_property_only_report_exists	Pass
	specialised.py::TestSpecialisedReportMethods	test_generate_risk_focused_report_custom_filename	Pass
	specialised.py::TestSpecialisedReportMethods	test_generate_risk_focused_report_with_mortgage	Pass
	specialised.py::TestSpecialisedReportMethods	test_generate_risk_focused_report_without_mortgage	Pass
	specialised.py::TestSpecialisedReportMethods	test_mortgage_focused_includes_mortgage_context_pages	Pass
	specialised.py::TestSpecialisedReportMethods	test_risk_focused_with_mortgage_includes_current_status	Pass
	specialised.py::TestSpecialisedReportMethods	test_risk_focused_without_mortgage_excludes_current_status	Pass
	specialised.py::TestUtilityMethods	test_get_page_categories_returns_copy	Pass
	specialised.py::TestUtilityMethods	test_get_page_categories_returns_dict	Pass
	specialised.py::TestUtilityMethods	test_list_available_pages_contains_all_expected	Pass
	specialised.py::TestUtilityMethods	test_list_available_pages_returns_list	Pass
	specialised.py::TestUtilityMethods	test_validate_pages_all_invalid	Pass
	specialised.py::TestUtilityMethods	test_validate_pages_all_valid	Pass
	specialised.py::TestUtilityMethods	test_validate_pages_empty_list	Pass
	specialised.py::TestUtilityMethods	test_validate_pages_mixed	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_base_page_coverage.py::TestAbstractGenerateSubclass	AbstractGenerateSubclass_returns_empty_list	Pass
	test_base_page_coverage.py::TestFormatCurrency	FormatCurrency_None_returns_not_specified	Pass
	test_base_page_coverage.py::TestFormatCurrency	FormatCurrency_String_delegates	Pass
	test_base_page_coverage.py::TestFormatPercentage	FormatPercentage_None_returns_not_specified	Pass
	test_base_page_coverage.py::TestFormatPercentage	FormatPercentage_String_delegates	Pass
	test_base_page_coverage.py::TestFormatValueEmpty	FormatValueEmpty_None_returns_not_specified	Pass
	test_base_page_coverage.py::TestFormatValueEmpty	FormatValueEmpty_EmptyString_returns_not_specified	Pass
	test_base_page_coverage.py::TestFormatValue	FormatValue_None_returns_not_specified	Pass
	test_base_page_coverage.py::TestPropertyBasePyleTitleStyle	PropertyBasePyleTitleStyle_Exists	Pass
	test_generator_direct.py::TestGenerateAutoOpenRecent	test_AutoOpenRecent_exception_does_not_raise	Pass
	test_generator_direct.py::TestGenerateAutoOpenRecent	test_AutoOpenRecent_false_skips	Pass
	test_generator_direct.py::TestGenerateAutoOpenRecent	test_AutoOpenRecent_true_tries	Pass
	test_generator_direct.py::TestGenerateMortgageFocused	test_MortgageFocused_type	Pass
	test_generator_direct.py::TestGeneratePropertyOnly	test_PropertyOnly_type	Pass
	test_generator_direct.py::TestGenerateRiskFocused	test_RiskFocused_type	Pass
	test_generator_direct.py::TestGenerateUnknownType	test_UnknownType_falls_back	Pass
	test_generator_direct.py::TestProcessAutoSelectPages	test_AutoSelectPages_with_mortgage	Pass
	test_generator_direct.py::TestProcessAutoSelectPages	test_AutoSelectPages_without_mortgage	Pass
	test_generator_direct.py::TestProcessGenerateElementDirect	test_GenerateElementDirect_exception_continues	Pass
	test_generator_direct.py::TestProcessGenerateElementDirect	test_GenerateElementDirect_skips_unknown	Pass
	test_generator_direct.py::TestProcessGenerateFilenameDirect	test_GenerateFilenameDirect_unknown_id	Pass
	test_generator_direct.py::TestProcessGenerateMortgageFocused	test_GenerateMortgageFocused_report	Pass
	test_generator_direct.py::TestProcessGeneratePropertyOnly	test_GeneratePropertyOnly_report	Pass
	test_generator_direct.py::TestProcessGenerateRiskFocused	test_GenerateRiskFocused_report_no_mortgage	Pass
	test_generator_direct.py::TestProcessGenerateRiskFocused	test_GenerateRiskFocused_report_with_mortgage	Pass
	test_generator_direct.py::TestProcessGetPageCategories	test_GetPageCategories_Direct	Pass
	test_generator_direct.py::TestProcessListAvailablePages	test_ListAvailablePages_Direct	Pass
	test_generator_direct.py::TestProcessMainOutputDirectory	test_MainOutputDirectory_uses_config	Pass
	test_generator_direct.py::TestProcessValidatePages	test_ValidatePages_Direct	Pass
	test_helpers.py::TestFindMortgagesByProperty	testByProperty_with_mortgages_found	Pass
	test_helpers.py::TestFindMortgagesByProperty	testByProperty_with_mortgages_not_found_returns_none	Pass
	test_helpers.py::TestFindMortgagesByProperty	testByProperty_invalid_type_returns_data	Pass
	test_helpers.py::TestFindMortgagesByProperty	testByProperty_list_input_found	Pass
	test_helpers.py::TestFindPropertiesById	testById_dict_without_properties_wraps_in_list	Pass
	test_helpers.py::TestFindPropertiesById	testById_invalid_type_raises	Pass
	test_helpers.py::TestFindPropertiesById	testById_list_input	Pass
	test_helpers.py::TestFindPropertiesById	testById_not_found_raises	Pass
	test_helpers.py::TestFindPropertiesById	testById_portfolio_key	Pass
	test_helpers.py::TestFindPropertiesById	testById_properties_key	Pass
	test_helpers.py::TestOpenPdfFile	test_darwin_success	Pass
	test_helpers.py::TestOpenPdfFile	test_generic_exception_returns_false	Pass
	test_helpers.py::TestOpenPdfFile	test_linux_success	Pass
	test_helpers.py::TestOpenPdfFile	test_subprocess_failure_returns_false	Pass
	test_helpers.py::TestOpenPdfFile	test_unknown_platform_uses_webbrowser	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_helpers.py::TestOpenPdfFile	test_windows_success	Pass
	test_integrator.py::TestGenerateRiskReport	test_found_property_returns_path	Pass
	test_integrator.py::TestGenerateRiskReport	test_missing_property_file_returns_none	Pass
	test_integrator.py::TestGenerateRiskReport	test_no_mortgage_file_still_generates	Pass
	test_integrator.py::TestGenerateRiskReport	test_output_contains_property_id	Pass
	test_integrator.py::TestGenerateRiskReport	test_output_dir_created	Pass
	test_integrator.py::TestGenerateRiskReport	test_output_file_created	Pass
	test_integrator.py::TestGenerateRiskReport	test_property_not_found_returns_none	Pass
	test_integrator_exception.py::TestGenerateRiskReportException	test_corrupt_property_file_returns_none	Pass
	test_integrator_exception.py::TestGenerateRiskReportException	test_permission_error_returns_none	Pass
	generation.py::TestGenerateRiskReport	test_analysis_type	Pass
	generation.py::TestGenerateRiskReport	test_basic_type	Pass
	generation.py::TestGenerateRiskReport	test_detailed_type	Pass
	generation.py::TestGenerateRiskReport	test_summary_type	Pass
	generation.py::TestGenerateRiskReport	test_unknown_type_fallback	Pass
	generation.py::TestRiskReportEdgeCases	test_auto_select_pages_returns_list	Pass
	generation.py::TestRiskReportEdgeCases	test_empty_gauge_data	Pass
	generation.py::TestRiskReportEdgeCases	test_generate_elements_exception_continues	Pass
	generation.py::TestRiskReportEdgeCases	test_generate_elements_skips_unknown	Pass
	generation.py::TestRiskReportEdgeCases	test_get_page_categories	Pass
	generation.py::TestRiskReportEdgeCases	test_high_risk_portfolio_report	Pass
	generation.py::TestRiskReportEdgeCases	test_list_available_pages	Pass
	generation.py::TestRiskReportEdgeCases	test_minimal_data_report	Pass
	generation.py::TestRiskReportEdgeCases	test_missing_mortgage_data	Pass
	generation.py::TestRiskReportEdgeCases	test_output_dir_uses_config	Pass
	generation.py::TestRiskReportEdgeCases	test_validate_pages_valid_and_invalid	Pass
	generation.py::TestRiskReportGeneration	test_analysis_report_generation	Pass
	generation.py::TestRiskReportGeneration	test_basic_report_generation	Pass
	generation.py::TestRiskReportGeneration	test_custom_filename	Pass
	generation.py::TestRiskReportGeneration	test_custom_page_selection	Pass
	generation.py::TestRiskReportGeneration	test_detailed_report_generation	Pass
	generation.py::TestRiskReportGeneration	test_summary_report_generation	Pass
	cli.py::TestErrorHandling	test_bad_flood_file_exits_with_1	Pass
	cli.py::TestErrorHandling	test_invalid_json_exits_with_1	Pass
	cli.py::TestListCategories	test_list_categories_iterates_all_categories	Pass
	cli.py::TestListCategories	test_list_categories_runs_without_error	Pass
	cli.py::TestListPages	test_list_pages_does_not_open_flood_file	Pass
	cli.py::TestListPages	test_list_pages_runs_without_error	Pass
	cli.py::TestReportGeneration	test_logging_after_generation	Pass
	cli.py::TestReportGeneration	test_report_type_analysis	Pass
	cli.py::TestReportGeneration	test_report_type_basic	Pass
	cli.py::TestReportGeneration	test_report_type_detailed	Pass
	cli.py::TestReportGeneration	test_report_type_else_branch	Pass
	cli.py::TestReportGeneration	test_report_type_summary	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	convenience.py::TestEdgeCases	test_all_unknown_pages_still_creates_pdf	Pass
	convenience.py::TestEdgeCases	test_completely_empty_flood_data	Pass
	convenience.py::TestEdgeCases	test_large_property_risk_dataset	Pass
	convenience.py::TestEdgeCases	test_missing_summary_key	Pass
	convenience.py::TestEdgeCases	test_negative_values_in_summary	Pass
	convenience.py::TestEdgeCases	test_none_gauge_data_key	Pass
	convenience.py::TestEdgeCases	test_report_with_all_pages	Pass
	convenience.py::TestEdgeCases	test_single_page_report	Pass
	convenience.py::TestEdgeCases	test_zero_properties_at_risk	Pass
	convenience.py::TestGenerateRisk	test_analysis_type_returns_path	Pass
	convenience.py::TestGenerateRisk	test_basic_type_returns_path	Pass
	convenience.py::TestGenerateRisk	test_default_report_type_is_basic	Pass
	convenience.py::TestGenerateRisk	test_detailed_type_returns_path	Pass
	convenience.py::TestGenerateRisk	test_output_dir_as_string	Pass
	convenience.py::TestGenerateRisk	test_pdf_is_non_empty_bytes_on_disk	Pass
	convenience.py::TestGenerateRisk	test_summary_type_returns_path	Pass
	convenience.py::TestGenerateRisk	test_unknown_type_falls_back_to_generate_report	Pass
	elements.py::TestGenerateElement	test_empty_page_list	Pass
	elements.py::TestGenerateElement	test_exception_in_page_continues	Pass
	elements.py::TestGenerateElement	test_first_page_no_page_break	Pass
	elements.py::TestGenerateElement	test_mixed_valid_and_unknown_pages	Pass
	elements.py::TestGenerateElement	test_multiple_pages_inserts_page_breaks	Pass
	elements.py::TestGenerateElement	test_returns_list	Pass
	elements.py::TestGenerateElement	test_unknown_page_skipped	Pass
	elements.py::TestGenerateReport	test_auto_select_pages_when_none	Pass
	elements.py::TestGenerateReport	test_auto_select_skipped_when_provided	Pass
	elements.py::TestGenerateReport	test_custom_output_filename	Pass
	elements.py::TestGenerateReport	test_custom_pages_to_include	Pass
	elements.py::TestGenerateReport	test_empty_flood_data	Pass
	elements.py::TestGenerateReport	test_full_data_report	Pass
	elements.py::TestGenerateReport	test_pdf_exists	Pass
	elements.py::TestGenerateReport	test_pdf_non_empty	Pass
	elements.py::TestGenerateReport	test_pdf_suffix	Pass
	elements.py::TestGenerateReport	test_returns_path	Pass
	init.py::TestAutoSelectPages	test_all_pages_exist	Pass
	init.py::TestAutoSelectPages	test_always_includes_appendix	Pass
	init.py::TestAutoSelectPages	test_empty_flood_data	Pass
	init.py::TestAutoSelectPages	test_includes_analysis_pages	Pass
	init.py::TestAutoSelectPages	test_includes_overview_pages	Pass
	init.py::TestAutoSelectPages	test_returns_list	Pass
	init.py::TestGenerateFilename	test_contains_flood_risk_report	Pass
	init.py::TestGenerateFilename	test_contains_timestamp	Pass
	init.py::TestGenerateFilename	test_ends_with_pdf	Pass
	init.py::TestGenerateFilename	test_returns_string	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	init.py::TestGenerateFilename	test_two_calls_produce_string_names	Pass
	init.py::TestRiskReportGenerator	test_init_categories_dict_initialized	Pass
	init.py::TestRiskReportGenerator	test_init_explicit_output_dir_path	Pass
	init.py::TestRiskReportGenerator	test_init_explicit_output_dir_string	Pass
	init.py::TestRiskReportGenerator	test_init_no_output_dir_uses_config	Pass
	init.py::TestRiskReportGenerator	test_init_output_dir_created	Pass
	init.py::TestRiskReportGenerator	test_init_pages_dict_initialized	Pass
	specialised.py::TestHeaderFooter	test_header_footer_callable	Pass
	specialised.py::TestHeaderFooter	test_header_footer_draws_strings	Pass
	specialised.py::TestSpecialisedReportMethods	test_basic_vs_detailed_page_counts	Pass
	specialised.py::TestSpecialisedReportMethods	test_generate_analysis_report_custom_filename	Pass
	specialised.py::TestSpecialisedReportMethods	test_generate_analysis_report_exists	Pass
	specialised.py::TestSpecialisedReportMethods	test_generate_basic_report_exists	Pass
	specialised.py::TestSpecialisedReportMethods	test_generate_basic_report_non_empty	Pass
	specialised.py::TestSpecialisedReportMethods	test_generate_detailed_report_exists	Pass
	specialised.py::TestSpecialisedReportMethods	test_generate_detailed_report_includes_mortgage_page	Pass
	specialised.py::TestSpecialisedReportMethods	test_generate_summary_report_custom_filename	Pass
	specialised.py::TestSpecialisedReportMethods	test_generate_summary_report_exists	Pass
	specialised.py::TestUtilityMethods	test_get_page_categories_contains_expected_keys	Pass
	specialised.py::TestUtilityMethods	test_get_page_categories_returns_copy	Pass
	specialised.py::TestUtilityMethods	test_get_page_categories_returns_dict	Pass
	specialised.py::TestUtilityMethods	test_list_available_pages_contains_all_expected	Pass
	specialised.py::TestUtilityMethods	test_list_available_pages_non_empty	Pass
	specialised.py::TestUtilityMethods	test_list_available_pages_returns_list	Pass
	specialised.py::TestUtilityMethods	test_validate_pages_all_invalid	Pass
	specialised.py::TestUtilityMethods	test_validate_pages_all_valid	Pass
	specialised.py::TestUtilityMethods	test_validate_pages_empty_list	Pass
	specialised.py::TestUtilityMethods	test_validate_pages_mixed	Pass
	utilities.py::TestConvenienceFunctions	test_generate_risk_report_basic	Pass
	utilities.py::TestConvenienceFunctions	test_generate_risk_report_default	Pass
	utilities.py::TestConvenienceFunctions	test_generate_risk_report_detailed	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_count_non_numeric	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_count_numeric	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_currency_non_numeric	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_currency_numeric	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_depth_non_numeric	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_depth_numeric	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_field_name_camel_case	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_field_name_replacements	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_percentage_non_numeric	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_percentage_numeric	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_value_bool_false	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_value_bool_true	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_value_empty_string	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	utilities.py::TestRiskBasePageFormatters	test_format_value_float_decimal	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_value_float_integer	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_value_int	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_value_non_empty_string	Pass
	utilities.py::TestRiskBasePageFormatters	test_format_value_none	Pass
	utilities.py::TestRiskReportUtilities	test_get_page_categories	Pass
	utilities.py::TestRiskReportUtilities	test_list_available_pages	Pass
	utilities.py::TestRiskReportUtilities	test_validate_pages_invalid	Pass
	utilities.py::TestRiskReportUtilities	test_validate_pages_valid	Pass
	utilities.py::TestRiskReportWorkflow	test_complete_workflow_all_types	Pass
	utilities.py::TestRiskReportWorkflow	test_complete_workflow_basic	Pass
	eod_pages_part1.py::TestEODPositionsPage	test_single_gauge_id_truncated	Pass
	eod_pages_part1.py::TestEODPositionsPage	test_multiple_gauges_grouped	Pass
	eod_pages_part1.py::TestEODPositionsPage	test_no_positions_shows_message	Pass
	eod_pages_part1.py::TestEODPositionsPage	test_returns_list	Pass
	eod_pages_part1.py::TestEODPositionsPage	test_single_position	Pass
	eod_pages_part1.py::TestEODPositionsPage	test_with_positions_creates_table	Pass
	eod_pages_part1.py::TestEODPositionsPage	test_zero_pnl_renders_dash	Pass
	eod_pages_part1.py::TestEODSummaryPage	test_date_appears_in_paragraph	Pass
	eod_pages_part1.py::TestEODSummaryPage	test_empty_summary	Pass
	eod_pages_part1.py::TestEODSummaryPage	test_has_paragraph_and_table	Pass
	eod_pages_part1.py::TestEODSummaryPage	test_negative_pnl_snapshot	Pass
	eod_pages_part1.py::TestEODSummaryPage	test_returns_list	Pass
	eod_pages_part2.py::TestEODChartsPage	test_insufficient_data_message	Pass
	eod_pages_part2.py::TestEODChartsPage	test_returns_list	Pass
	eod_pages_part2.py::TestEODChartsPage	test_single_entry_insufficient	Pass
	eod_pages_part2.py::TestEODChartsPage	test_sufficient_data_creates_content	Pass
	eod_pages_part2.py::TestEODPnlDetailPage	test_market_movers_top_20	Pass
	eod_pages_part2.py::TestEODPnlDetailPage	test_market_move_pnl_creates_table	Pass
	eod_pages_part2.py::TestEODPnlDetailPage	test_new_trade_pnl_creates_table	Pass
	eod_pages_part2.py::TestEODPnlDetailPage	test_no_positions_shows_messages	Pass
	eod_pages_part2.py::TestEODPnlDetailPage	test_returns_list	Pass
	eod_pages_part2.py::TestEODReportGenerator	test_eod_in_filename	Pass
	eod_pages_part2.py::TestEODReportGenerator	test_generate_empty_snapshot	Pass
	eod_pages_part2.py::TestEODReportGenerator	test_generate_report_returns_path	Pass
	eod_pages_part2.py::TestEODReportGenerator	test_generate_report_without_pnl_series	Pass
	eod_pages_part2.py::TestEODReportGenerator	test_output_dir_uses_config	Pass
	eod_pages_part2.py::TestEODReportGenerator	test_output_dir_created	Pass
	eod_pages_part2.py::TestEODReportGenerator	test_pdf_not_empty	Pass
	eod_pages_part2.py::TestPnLDetailPage	test_zero_pnl_pnl_returns_dash	Pass
	test_eod_charts_coverage.py::TestDailyPnlChart	test_empty_daily_pnl_returns_drawing	Pass
	test_eod_charts_coverage.py::TestDailyPnlChart	test_mixed_positive_negative_renders	Pass
	test_eod_charts_coverage.py::TestDailyPnlChart	test_negative_only_renders	Pass
	test_eod_charts_coverage.py::TestBuildCharts	test_unknown_chart_key_returns_drawing	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_eod_charts_coverage.py::TestByGaugeChartEmpty	test_by_gauge_chart_gauge_renders_correctly	Pass
	test_eod_charts_coverage.py::TestByGaugeChartEmpty	test_by_gauge_chart_empty_returns_drawing	Pass
	test_eod_charts_coverage.py::TestByGaugeChartEmpty	test_by_gauge_chart_with_zero_pnl_returns_drawing	Pass
	test_eod_charts_coverage.py::TestRunningPnlChartEmpty	test_running_pnl_chart_empty_returns_drawing	Pass
	open_pdf.py::TestOpenPdfFile	test_called_process_error_returns_false	Pass
	open_pdf.py::TestOpenPdfFile	test_darwin_calls_open	Pass
	open_pdf.py::TestOpenPdfFile	test_generic_exception_returns_false	Pass
	open_pdf.py::TestOpenPdfFile	test_linux_calls_xdg_open	Pass
	open_pdf.py::TestOpenPdfFile	test_unknown_platform_uses_webbrowser	Pass
	open_pdf.py::TestOpenPdfFile	test_windows_calls_start	Pass
	catchment.py::TestCatchmentExceptionHandlers	test_selected_catchment_selector_exception_returns_500	Pass
	catchment.py::TestCatchmentExceptionHandlers	test_selected_catchment_exception_returns_500	Pass
	catchment.py::TestServeCatchmentSelector	test_returns_404_when_file_missing	Pass
	catchment.py::TestServeCatchmentSelector	test_returns_html_when_file_exists	Pass
	catchment.py::TestServeCatchmentSelector	test_returns_json_error_or_html	Pass
	catchment.py::TestSetCatchment	test_case_insensitive_thames	Pass
	catchment.py::TestSetCatchment	test_missing_catchment_id_returns_400	Pass
	catchment.py::TestSetCatchment	test_no_json_returns_400	Pass
	catchment.py::TestSetCatchment	test_null_catchment_id_returns_400	Pass
	catchment.py::TestSetCatchment	test_options_returns_200	Pass
	catchment.py::TestSetCatchment	test_unknown_catchment_returns_400	Pass
	catchment.py::TestSetCatchment	test_valid_thames_returns_success	Pass
	counterparty.py::TestCounterpartyErrors	test_get_counterparty_error_returns_500	Pass
	counterparty.py::TestCounterpartyErrors	test_list_counterparties_error_returns_500	Pass
	counterparty.py::TestGetCounterparty	test_found_returns_200	Pass
	counterparty.py::TestGetCounterparty	test_found_second_counterparty	Pass
	counterparty.py::TestGetCounterparty	test_not_found_returns_404	Pass
	counterparty.py::TestListCounterparties	test_no_file_returns_empty_list	Pass
	counterparty.py::TestListCounterparties	test_summary_fields	Pass
	counterparty.py::TestListCounterparties	test_with_data_returns_summary	Pass
	cross_entity.py::TestCrossEntityConsistencyAPI	test_all_gauges_have_timeseries	Pass
	cross_entity.py::TestCrossEntityConsistencyAPI	test_all_gauges_hazard_no_500	Pass
	cross_entity.py::TestCrossEntityConsistencyAPI	test_all_properties_have_flood_data	Pass
	cross_entity.py::TestInteractivityJSRendering	test_context_menus_renders	Pass
	cross_entity.py::TestInteractivityJSRendering	test_gauge_hazard_curve_renders	Pass
	cross_entity.py::TestInteractivityJSRendering	test_gauge_pdf_panel_renders	Pass
	cross_entity.py::TestInteractivityJSRendering	test_gauge_storm_analysis_renders	Pass
	cross_entity.py::TestInteractivityJSRendering	test_notifications_renders	Pass
	cross_entity.py::TestInteractivityJSRendering	test_property_hazard_curve_renders	Pass
	cross_entity.py::TestInteractivityJSRendering	test_property_pdf_panel_renders	Pass
	cross_entity.py::TestInteractivityJSRendering	test_property_storm_analysis_renders	Pass
	cross_entity.py::TestInteractivityJSRendering	test_property_storm_js_guards_empty_events	Pass
	gauge.py::TestDataFormatValidation	test_gaugehc_file_structure	Pass
	gauge.py::TestDataFormatValidation	test_gaugets_file_structure	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	gauge.py::TestDataFormatValidation	test_propertyhc_file_structure	Pass
	gauge.py::TestDataFormatValidation	test_propertyts_file_structure	Pass
	gauge.py::TestGaugeHazardRoutes	test_gauge_hazard_existing	Pass
	gauge.py::TestGaugeHazardRoutes	test_gauge_hazard_gev_params	Pass
	gauge.py::TestGaugeHazardRoutes	test_gauge_hazard_nonexistent_gauge	Pass
	gauge.py::TestGaugeHazardRoutes	test_gauge_hazard_not_found	Pass
	gauge.py::TestGaugeHazardRoutes	test_gauge_storms_existing	Pass
	gauge.py::TestGaugeIDConsistency	test_gaugehc_ids_match_gauge_portfolio	Pass
	gauge.py::TestGaugeIDConsistency	test_gaugets_ids_match_gauge_portfolio	Pass
	gauge.py::TestGaugeIDConsistency	test_gaugets_internal_id_matches_filename	Pass
	property.py::TestMissingDataHandling	test_property_hazard_no_file	Pass
	property.py::TestMissingDataHandling	test_propertyhc_summary_no_file	Pass
	property.py::TestMissingDataHandling	test_propertyts_summary_no_dir	Pass
	property.py::TestPropertyHCRoutes	test_property_hazard_existing	Pass
	property.py::TestPropertyHCRoutes	test_property_hazard_nonexistent	Pass
	property.py::TestPropertyHCRoutes	test_property_hazard_not_in_curves	Pass
	property.py::TestPropertyHCRoutes	test_propertyhc_basis	Pass
	property.py::TestPropertyHCRoutes	test_propertyhc_summary	Pass
	property.py::TestPropertyIDConsistency	test_propertyhc_ids_match_property_portfolio	Pass
	property.py::TestPropertyIDConsistency	test_propertyhc_ids_subset_of_propertyts	Pass
	property.py::TestPropertyIDConsistency	test_propertyts_ids_match_property_portfolio	Pass
	property.py::TestPropertyTSRoutes	test_list_flood_storms	Pass
	property.py::TestPropertyTSRoutes	test_list_flood_storms_only_returns_damaging	Pass
	property.py::TestPropertyTSRoutes	test_property_floods_existing	Pass
	property.py::TestPropertyTSRoutes	test_property_floods_few_events	Pass
	property.py::TestPropertyTSRoutes	test_property_floods_not_found	Pass
	property.py::TestPropertyTSRoutes	test_property_storms_existing	Pass
	property.py::TestPropertyTSRoutes	test_property_storms_not_found	Pass
	property.py::TestPropertyTSRoutes	test_property_storms_zero_floods	Pass
	property.py::TestPropertyTSRoutes	test_propertyts_summary	Pass
	financial.py::TestPortfolioImpactNoData	test_no_pts_dir_returns_404	Pass
	financial.py::TestPortfolioImpactOptions	test_options_returns_200	Pass
	financial.py::TestPortfolioImpactStormNotFound	test_unknown_storm_returns_404	Pass
	financial.py::TestPortfolioImpactSuccess	test_mortgage_enrichment_present	Pass
	financial.py::TestPortfolioImpactSuccess	test_property_entry_has_damage_fields	Pass
	financial.py::TestPortfolioImpactSuccess	test_response_has_portfolio_key	Pass
	financial.py::TestPortfolioImpactSuccess	test_response_has_properties_list	Pass
	financial.py::TestPortfolioImpactSuccess	test_storm_id_in_response	Pass
	financial.py::TestPortfolioImpactSuccess	test_valid_storm_returns_200	Pass
	test_crud.py::TestCrudGetGauge	test_get_gauge_not_found	Pass
	test_crud.py::TestCrudGetGauge	test_get_gauge_success	Pass
	test_crud.py::TestCrudListGauges	test_list_gauges_error_returns_500	Pass
	test_crud.py::TestCrudListGauges	test_list_gauges_legacy_route	Pass
	test_crud.py::TestCrudListGauges	test_list_gauges_options	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_crud.py::TestCrudListGauges	test_list_gauges_returns_200	Pass
	test_hazard.py::TestHazardErrorPaths	test_get_gauge_hazard_corrupted_file_returns_500	Pass
	test_hazard.py::TestPricePrsHappyPath	test_price_prs_exception_returns_500	Pass
	test_hazard.py::TestPricePrsHappyPath	test_price_prs_gauge_not_in_hazard	Pass
	test_hazard.py::TestPricePrsHappyPath	test_price_prs_no_hazard_file	Pass
	test_hazard.py::TestPricePrsHappyPath	test_price_prs_no_json_body	Pass
	test_hazard.py::TestPricePrsHappyPath	test_price_prs_null_json_body	Pass
	test_hazard.py::TestPricePrsHappyPath	test_price_prs_severe_trigger	Pass
	test_hazard.py::TestPricePrsHappyPath	test_price_prs_success	Pass
	test_hazard.py::TestPricePrsHappyPath	test_price_prs_with_defaults	Pass
	test_history.py::TestHistoryErrorPaths	test_get_gauge_history_corrupted_file_returns_500	Pass
	test_history.py::TestHistoryErrorPaths	test_get_gauge_statistics_exception_returns_500	Pass
	test_history.py::TestHistoryErrorPaths	test_get_gaugets_exception_returns_500	Pass
	test_reports.py::TestReportsErrorPaths	test_gaugehd_load_error_warning	Pass
	test_reports.py::TestReportsErrorPaths	test_general_exception_returns_500	Pass
	test_reports.py::TestReportsErrorPaths	test_hazard_load_error_warning	Pass
	test_reports.py::TestReportsErrorPaths	test_import_error_returns_500	Pass
	test_reports.py::TestReportsErrorPaths	test_timeseries_load_success_with_storm_responses	Pass
	test_reports.py::TestReportsErrorPaths	test_timeseries_load_warning	Pass
	test_storms.py::TestStormsErrorPaths	test_gaugets_dir_corrupted_file_continues	Pass
	test_storms.py::TestStormsErrorPaths	test_no_gaugets_dir	Pass
	test_storms.py::TestStormsErrorPaths	test_no_storm_sequences_file	Pass
	test_storms.py::TestStormsErrorPaths	test_storm_sequences_corrupted_continues	Pass
	test_storms.py::TestStormsErrorPaths	test_storms_exception_returns_500	Pass
	gauges_hazard.py::TestGetGaugeHazard	test_gauge_not_found_returns_404	Pass
	gauges_hazard.py::TestGetGaugeHazard	test_no_curve_for_gauge_returns_404	Pass
	gauges_hazard.py::TestGetGaugeHazard	test_no_hazard_file_returns_404	Pass
	gauges_hazard.py::TestGetGaugeHazard	test_response_has_annual_flood_probs	Pass
	gauges_hazard.py::TestGetGaugeHazard	test_response_has_curve_points	Pass
	gauges_hazard.py::TestGetGaugeHazard	test_response_has_flood_stages	Pass
	gauges_hazard.py::TestGetGaugeHazard	test_response_has_gauge_id	Pass
	gauges_hazard.py::TestGetGaugeHazard	test_response_has_gev_parameters	Pass
	gauges_hazard.py::TestGetGaugeHazard	test_response_has_term_structures	Pass
	gauges_hazard.py::TestGetGaugeHazard	test_valid_hazard_returns_200	Pass
	gauges_hazard.py::TestPricePrs	test_gauge_not_in_hazard_returns_404	Pass
	gauges_hazard.py::TestPricePrs	test_invalid_trigger_level_returns_400	Pass
	gauges_hazard.py::TestPricePrs	test_missing_gauge_id_returns_400	Pass
	gauges_hazard.py::TestPricePrs	test_no_hazard_file_returns_404	Pass
	gauges_hazard.py::TestPricePrs	test_no_json_returns_400	Pass
	gauges_hazard.py::TestPricePrs	test_options_returns_ok	Pass
	gauges_history.py::TestGaugeHistoryStatus	test_response_has_count_and_gauges_keys	Pass
	gauges_history.py::TestGaugeHistoryStatus	test_returns_200_with_no_gaugehd_files	Pass
	gauges_history.py::TestGaugeHistoryStatus	test_returns_correct_count_when_files_present	Pass
	gauges_history.py::TestGetGaugeHistory	test_days_larger_than_available_returns_all	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	gauges_history.py::TestGetGaugeHistory	test_days_param_limits_observations	Pass
	gauges_history.py::TestGetGaugeHistory	test_days_param_zero_returns_all	Pass
	gauges_history.py::TestGetGaugeHistory	test_no_history_file_returns_404	Pass
	gauges_history.py::TestGetGaugeHistory	test_response_has_daily_observations	Pass
	gauges_history.py::TestGetGaugeHistory	test_response_has_flood_stages	Pass
	gauges_history.py::TestGetGaugeHistory	test_response_has_gauge_metadata	Pass
	gauges_history.py::TestGetGaugeHistory	test_response_has_statistics	Pass
	gauges_history.py::TestGetGaugeHistory	test_valid_history_returns_200	Pass
	gauges_history.py::TestGetGaugeStatistics	test_gauge_not_found_returns_404	Pass
	gauges_history.py::TestGetGaugeStatistics	test_valid_gauge_returns_success	Pass
	gauges_history.py::TestGetGauge	test_gauge_not_found_returns_404	Pass
	gauges_history.py::TestGetGauge	test_response_has_count_and_readings	Pass
	gauges_history.py::TestGetGauge	test_valid_gauge_returns_success	Pass
	gauges_reports.py::TestGenerateCustomReport	test_gauge_not_found_returns_404	Pass
	gauges_reports.py::TestGenerateCustomReport	test_legacy_route_alias	Pass
	gauges_reports.py::TestGenerateCustomReport	test_missing_gauge_id_returns_400	Pass
	gauges_reports.py::TestGenerateCustomReport	test_no_body_returns_400	Pass
	gauges_reports.py::TestGenerateCustomReport	test_non_dict_json_body_returns_400	Pass
	gauges_reports.py::TestGenerateCustomReport	test_options_returns_ok	Pass
	gauges_reports.py::TestGenerateCustomReport	test_response_has_file_path	Pass
	gauges_reports.py::TestGenerateCustomReport	test_response_has_pdf_base64	Pass
	gauges_reports.py::TestGenerateCustomReport	test_valid_gauge_generates_report	Pass
	gauges_reports.py::TestGenerateCustomReport	test_with_gaugehd_file_present	Pass
	gauges_reports.py::TestGenerateCustomReport	test_with_hazard_file_present	Pass
	gauges_storms.py::TestGetGaugeStorms	test_gauge_not_found_returns_404	Pass
	gauges_storms.py::TestGetGaugeStorms	test_response_has_flood_simulation	Pass
	gauges_storms.py::TestGetGaugeStorms	test_response_has_flood_stages	Pass
	gauges_storms.py::TestGetGaugeStorms	test_response_has_gauge_id	Pass
	gauges_storms.py::TestGetGaugeStorms	test_response_has_storm_responses	Pass
	gauges_storms.py::TestGetGaugeStorms	test_storm_enrichment_from_storms_json	Pass
	gauges_storms.py::TestGetGaugeStorms	test_valid_gauge_returns_200	Pass
	gauges_storms.py::TestGetGaugeStorms	test_with_severe_exceeded_counts_gauges	Pass
	audit_part1.py::TestGetAuditTrail	test_after_log_entry_appears	Pass
	audit_part1.py::TestGetAuditTrail	test_empty_trail_returns_200	Pass
	audit_part1.py::TestGetAuditTrail	test_filter_by_event_type	Pass
	audit_part1.py::TestGetAuditTrail	test_filter_by_model_id	Pass
	audit_part1.py::TestGetAuditTrail	test_filter_by_model_id_no_match	Pass
	audit_part1.py::TestGetAuditTrail	test_has_returned_field	Pass
	audit_part1.py::TestGetAuditTrail	test_limit_parameter	Pass
	audit_part1.py::TestLogModelUsage	test_defaults_applied_when_no_json	Pass
	audit_part1.py::TestLogModelUsage	test_entry_has_expected_fields	Pass
	audit_part1.py::TestLogModelUsage	test_model_not_found_returns_404	Pass
	audit_part1.py::TestLogModelUsage	test_parameters_stored	Pass
	audit_part1.py::TestLogModelUsage	test_valid_model_returns_200	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	audit_part1.py::TestUpdateValidation	test_all_valid_statuses	Pass
	audit_part1.py::TestUpdateValidation	test_invalid_status_returns_400	Pass
	audit_part1.py::TestUpdateValidation	test_model_not_found_returns_404	Pass
	audit_part1.py::TestUpdateValidation	test_question_not_found_returns_404	Pass
	audit_part1.py::TestUpdateValidation	test_risk_rating_recalculated	Pass
	audit_part1.py::TestUpdateValidation	test_valid_update_has_model_and_audit_entry	Pass
	audit_part1.py::TestUpdateValidation	test_valid_update_returns_success	Pass
	audit_part2.py::TestGetRiskRating	test_model_not_found_returns_404	Pass
	audit_part2.py::TestGetRiskRating	test_response_has_risk_rating	Pass
	audit_part2.py::TestGetRiskRating	test_risk_rating_has_component_scores	Pass
	audit_part2.py::TestGetRiskRating	test_valid_model_returns_success	Pass
	audit_part2.py::TestOverrideRiskRating	test_clear_override_with_null	Pass
	audit_part2.py::TestOverrideRiskRating	test_invalid_rating_returns_400	Pass
	audit_part2.py::TestOverrideRiskRating	test_model_not_found_returns_404	Pass
	audit_part2.py::TestOverrideRiskRating	test_no_reason_field_returns_400	Pass
	audit_part2.py::TestOverrideRiskRating	test_no_reason_returns_400	Pass
	audit_part2.py::TestOverrideRiskRating	test_override_stored_in_model	Pass
	audit_part2.py::TestOverrideRiskRating	test_response_has_audit_entry	Pass
	audit_part2.py::TestOverrideRiskRating	test_valid_override_returns_success	Pass
	audit_part2.py::TestPdfEndpoints	test_documentation_pdf_known_model_no_file_404	Pass
	audit_part2.py::TestPdfEndpoints	test_documentation_pdf_unknown_model_404	Pass
	audit_part2.py::TestPdfEndpoints	test_mrc_tor_pdf_responds	Pass
	audit_part2.py::TestPdfEndpoints	test_parameter_inventory_pdf_responds	Pass
	audit_part2.py::TestPdfEndpoints	test_test_results_pdf_known_model_no_file_404	Pass
	audit_part2.py::TestPdfEndpoints	test_test_results_pdf_unknown_model_404	Pass
	audit_reports.py::TestDownloadAuditReport	test_report_not_found_returns_404	Pass
	audit_reports.py::TestDownloadAuditReport	test_report_pdf_served_as_attachment	Pass
	audit_reports.py::TestDownloadAuditReport	test_path_traversal_returns_403	Pass
	audit_reports.py::TestDownloadAuditReport	test_valid_pdf_served_inline	Pass
	audit_reports.py::TestListAuditReports	test_coverage_available_false_when_no_index	Pass
	audit_reports.py::TestListAuditReports	test_coverage_available_true_when_index_present	Pass
	audit_reports.py::TestListAuditReports	test_empty_dir_returns_empty_list	Pass
	audit_reports.py::TestListAuditReports	test_file_has_size_and_modified	Pass
	audit_reports.py::TestListAuditReports	test_file_type_label_json	Pass
	audit_reports.py::TestListAuditReports	test_file_type_label_pdf	Pass
	audit_reports.py::TestListAuditReports	test_file_type_label_unknown	Pass
	audit_reports.py::TestListAuditReports	test_no_dir_returns_empty_list	Pass
	audit_reports.py::TestListAuditReports	test_pdf_file_listed	Pass
	audit_reports.py::TestListAuditReports	test_subdirectories_not_listed	Pass
	audit_reports.py::TestServeCoverageIndex	test_no_coverage_dir_returns_404	Pass
	audit_reports.py::TestServeCoverageIndex	test_with_coverage_index_returns_200	Pass
	bibliography.py::TestAddBibliography	test_add_appears_in_list	Pass
	bibliography.py::TestAddBibliography	test_add_assigns_id	Pass
	bibliography.py::TestAddBibliography	test_add_no_json_body_uses_defaults	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	bibliography.py::TestAddBibliography	test_add_returns_200	Pass
	bibliography.py::TestAddBibliography	test_add_returns_reference	Pass
	bibliography.py::TestAddBibliography	test_add_same_category_twice_no_duplicate	Pass
	bibliography.py::TestAddBibliography	test_add_stores_all_fields	Pass
	bibliography.py::TestAddBibliography	test_add_with_category_updates_categories	Pass
	bibliography.py::TestDeleteBibliography	test_delete_nonexistent_still_returns_success	Pass
	bibliography.py::TestDeleteBibliography	test_delete_returns_success	Pass
	bibliography.py::TestDeleteBibliography	test_deleted_ref_not_in_list	Pass
	bibliography.py::TestGetBibliography	test_empty_references	Pass
	bibliography.py::TestGetBibliography	test_has_categories_key	Pass
	bibliography.py::TestGetBibliography	test_returns_200	Pass
	bibliography.py::TestGetBibliography	test_returns_success_status	Pass
	bibliography.py::TestUpdateBibliography	test_update_ref_not_found_returns_404	Pass
	bibliography.py::TestUpdateBibliography	test_update_ref_changes_field	Pass
	bibliography.py::TestUpdateBibliography	test_update_ref_returns_success	Pass
	bibliography.py::TestUpdateBibliography	test_update_ref_with_new_category_adds_to_list	Pass
	compliance_part1.py::TestGetBcbs239	test_no_data_returns_404	Pass
	compliance_part1.py::TestGetBcbs239	test_principles_count	Pass
	compliance_part1.py::TestGetBcbs239	test_response_has_assessment	Pass
	compliance_part1.py::TestGetBcbs239	test_with_data_returns_200	Pass
	compliance_part1.py::TestGetBcbs239Pdf	test_bcbs239_pdf_responds	Pass
	compliance_part1.py::TestUpdateBcbs239Principle	test_assessment_date_updated	Pass
	compliance_part1.py::TestUpdateBcbs239Principle	test_no_data_file_returns_404	Pass
	compliance_part1.py::TestUpdateBcbs239Principle	test_principle_not_found_returns_404	Pass
	compliance_part1.py::TestUpdateBcbs239Principle	test_score_1_derives_non_compliant_status	Pass
	compliance_part1.py::TestUpdateBcbs239Principle	test_score_4_derives_fully_compliant_status	Pass
	compliance_part1.py::TestUpdateBcbs239Principle	test_update_evidence_field	Pass
	compliance_part1.py::TestUpdateBcbs239Principle	test_update_gaps_and_remediation	Pass
	compliance_part1.py::TestUpdateBcbs239Principle	test_valid_update_returns_200	Pass
	compliance_part2.py::TestUpdateActivityExtra	test_activity_not_found_returns_404	Pass
	compliance_part2.py::TestUpdateActivityExtra	test_invalid_role_id_returns_400	Pass
	compliance_part2.py::TestUpdateActivityExtra	test_activity_updated	Pass
	compliance_part2.py::TestUpdateActivityExtra	test_allowed_role_allowed	Pass
	compliance_part2.py::TestUpdateActivityExtra	test_valid_activity_update_returns_success	Pass
	compliance_part2.py::TestUpdateEscalationExtra	test_escalation_threshold_updated	Pass
	compliance_part2.py::TestUpdateEscalationExtra	test_response_has_audit_entry	Pass
	compliance_part2.py::TestUpdateEscalationExtra	test_trigger_not_found_returns_404	Pass
	compliance_part2.py::TestUpdateEscalationExtra	test_valid_escalation_update_returns_success	Pass
	compliance_part2.py::TestUpdateRoleExtra	test_backup_field_updated	Pass
	compliance_part2.py::TestUpdateRoleExtra	test_no_role_extra_file_returns_404	Pass
	compliance_part2.py::TestUpdateRoleExtra	test_response_has_audit_entry	Pass
	compliance_part2.py::TestUpdateRoleExtra	test_role_not_found_returns_404	Pass
	compliance_part2.py::TestUpdateRoleExtra	test_valid_role_update_returns_success	Pass
	documents_part1.py::TestDeleteDocument	test_deleted_doc_not_in_list	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	documents_part1.py::TestDeleteDocument	test_not_found_returns_404	Pass
	documents_part1.py::TestDeleteDocument	test_valid_delete_returns_success	Pass
	documents_part1.py::TestDownloadDocument	test_not_found_returns_404	Pass
	documents_part1.py::TestDownloadDocument	test_valid_download_returns_200	Pass
	documents_part1.py::TestGetDocuments	test_empty_list_returns_success	Pass
	documents_part1.py::TestUploadDocument	test_empty_filename_returns_400	Pass
	documents_part1.py::TestUploadDocument	test_no_file_returns_400	Pass
	documents_part1.py::TestUploadDocument	test_upload_appears_in_list	Pass
	documents_part1.py::TestUploadDocument	test_upload_response_has_id	Pass
	documents_part1.py::TestUploadDocument	test_upload_stores_description	Pass
	documents_part1.py::TestUploadDocument	test_valid_upload_returns_200	Pass
	documents_part2.py::TestDeleteSavedMetadata	test_delete_save_failure_returns_500	Pass
	documents_part2.py::TestDownloadFileMissingOnDisk	test_missing_on_disk_returns_404	Pass
	documents_part2.py::TestUploadSaveMetadataFails	test_upload_save_failure_returns_500	Pass
	inventory.py::TestInventorySummary	test_get_models_includes_risk_fields	Pass
	inventory.py::TestRiskRating	test_get_risk_rating_calculation	Pass
	inventory.py::TestRiskRating	test_risk_rating_clear_override	Pass
	inventory.py::TestRiskRating	test_risk_rating_override	Pass
	inventory.py::TestRiskRating	test_risk_rating_override_invalid_rating	Pass
	inventory.py::TestRiskRating	test_risk_rating_override_requires_reason	Pass
	inventory.py::TestValidationQuestions	test_update_triggers_risk_recalculation	Pass
	inventory.py::TestValidationQuestions	test_update_validation_question	Pass
	inventory.py::TestValidationQuestions	test_update_validation_question_invalid_status	Pass
	inventory.py::TestValidationQuestions	test_update_validation_question_not_found	Pass
	models.py::TestGetEditableFields	test_has_fields	Pass
	models.py::TestGetEditableFields	test_returns_200	Pass
	models.py::TestGetEditableFields	test_returns_success_status	Pass
	models.py::TestGetModelDetail	test_model_has_validation_questions	Pass
	models.py::TestGetModelDetail	test_model_not_found_returns_404	Pass
	models.py::TestGetModelDetail	test_response_has_audit_entries	Pass
	models.py::TestGetModelDetail	test_valid_model_returns_success	Pass
	models.py::TestGetModels	test_category_distribution	Pass
	models.py::TestGetModels	test_model_has_review_status	Pass
	models.py::TestGetModels	test_model_has_risk_rating	Pass
	models.py::TestGetModels	test_model_has_validation_coverage	Pass
	models.py::TestGetModels	test_response_has_metadata	Pass
	models.py::TestGetModels	test_returns_200	Pass
	models.py::TestGetModels	test_returns_models_list	Pass
	models.py::TestGetModels	test_returns_success_status	Pass
	models.py::TestGetModels	test_tier_distribution	Pass
	models.py::TestUpdateModelField	test_empty_value_returns_400	Pass
	models.py::TestUpdateModelField	test_invalid_choice_value_returns_400	Pass
	models.py::TestUpdateModelField	test_invalid_field_returns_400	Pass
	models.py::TestUpdateModelField	test_model_not_found_returns_404	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	models.py::TestUpdateModelField	test_no_reason_returns_400	Pass
	models.py::TestUpdateModelField	test_response_has_audit_entry	Pass
	models.py::TestUpdateModelField	test_update_lifecycle_stage	Pass
	models.py::TestUpdateModelField	test_update_validation_status	Pass
	models.py::TestUpdateModelField	test_valid_choice_field_update_returns_success	Pass
	models.py::TestUpdateModelField	test_valid_text_field_update_returns_success	Pass
	mrc_crud_actions.py::TestActionsCrud	test_add_action_auto_id	Pass
	mrc_crud_actions.py::TestActionsCrud	test_add_action_missing_meeting_returns_404	Pass
	mrc_crud_actions.py::TestActionsCrud	test_add_action_returns_200	Pass
	mrc_crud_actions.py::TestActionsCrud	test_delete_action_missing_meeting_returns_404	Pass
	mrc_crud_actions.py::TestActionsCrud	test_delete_action_returns_200	Pass
	mrc_crud_actions.py::TestActionsCrud	test_update_action_not_found_returns_404	Pass
	mrc_crud_actions.py::TestActionsCrud	test_update_action_returns_200	Pass
	mrc_crud_actions.py::TestParticipantsCrud	test_add_participant_auto_id	Pass
	mrc_crud_actions.py::TestParticipantsCrud	test_add_participant_missing_meeting_returns_404	Pass
	mrc_crud_actions.py::TestParticipantsCrud	test_add_participant_returns_200	Pass
	mrc_crud_actions.py::TestParticipantsCrud	test_delete_participant_missing_meeting_returns_404	Pass
	mrc_crud_actions.py::TestParticipantsCrud	test_delete_participant_returns_200	Pass
	mrc_crud_actions.py::TestParticipantsCrud	test_update_participant_not_found_returns_404	Pass
	mrc_crud_actions.py::TestParticipantsCrud	test_update_participant_returns_200	Pass
	mrc_crud_agenda.py::TestAgendaCrud	test_add_agenda_item_increments_count	Pass
	mrc_crud_agenda.py::TestAgendaCrud	test_add_agenda_item_returns_200	Pass
	mrc_crud_agenda.py::TestAgendaCrud	test_add_agenda_missing_meeting_returns_404	Pass
	mrc_crud_agenda.py::TestAgendaCrud	test_delete_agenda_item_returns_200	Pass
	mrc_crud_agenda.py::TestAgendaCrud	test_delete_agenda_missing_meeting_returns_404	Pass
	mrc_crud_agenda.py::TestAgendaCrud	test_update_agenda_item_not_found_returns_404	Pass
	mrc_crud_agenda.py::TestAgendaCrud	test_update_agenda_item_returns_200	Pass
	mrc_crud_agenda.py::TestDecisionsCrud	test_add_decision_auto_ids	Pass
	mrc_crud_agenda.py::TestDecisionsCrud	test_add_decision_missing_meeting_returns_404	Pass
	mrc_crud_agenda.py::TestDecisionsCrud	test_add_decision_returns_200	Pass
	mrc_crud_agenda.py::TestDecisionsCrud	test_delete_decision_missing_meeting_returns_404	Pass
	mrc_crud_agenda.py::TestDecisionsCrud	test_delete_decision_returns_200	Pass
	mrc_crud_agenda.py::TestDecisionsCrud	test_update_decision_not_found_returns_404	Pass
	mrc_crud_agenda.py::TestDecisionsCrud	test_update_decision_returns_200	Pass
	mrc_crud_agenda.py::TestMinutesCrud	test_update_minutes_missing_meeting_returns_404	Pass
	mrc_crud_agenda.py::TestMinutesCrud	test_update_minutes_returns_200	Pass
	mrc_crud_edge_part1.py::TestMinutesItemCrud	test_add_minute_item_missing_meeting_returns_404	Pass
	mrc_crud_edge_part1.py::TestMinutesItemCrud	test_add_minute_item_returns_200	Pass
	mrc_crud_edge_part1.py::TestMinutesItemCrud	test_delete_minute_item_legacy_format_returns_400	Pass
	mrc_crud_edge_part1.py::TestMinutesItemCrud	test_delete_minute_item_missing_meeting_returns_404	Pass
	mrc_crud_edge_part1.py::TestMinutesItemCrud	test_delete_minute_item_returns_200	Pass
	mrc_crud_edge_part1.py::TestMinutesItemCrud	test_update_minute_item_legacy_format_returns_400	Pass
	mrc_crud_edge_part1.py::TestMinutesItemCrud	test_update_minute_item_not_found_returns_404	Pass
	mrc_crud_edge_part1.py::TestMinutesItemCrud	test_update_minute_item_returns_200	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	mrc_crud_edge_part2.py::TestMatterAddActionI	test_add_action_with_malformed_existing_id	Pass
	mrc_crud_edge_part2.py::TestMatterAddDecisionI	test_add_decision_with_malformed_existing_id	Pass
	mrc_crud_edge_part2.py::TestMatterAddParticipantI	test_add_participant_with_malformed_existing_id	Pass
	mrc_crud_edge_part2.py::TestMinutesAddingItemsToMeeting	test_add_minutes_item_when_minutes_is_string	Pass
	mrc_crud_edge_part2.py::TestUpdateMissingMeeting	test_update_action_missing_meeting_returns_404	Pass
	mrc_crud_edge_part2.py::TestUpdateMissingMeeting	test_update_decision_missing_meeting_returns_404	Pass
	mrc_crud_edge_part2.py::TestUpdateMissingMeeting	test_update_minutes_item_missing_meeting_returns_404	Pass
	mrc_crud_edge_part2.py::TestUpdateMissingMeeting	test_update_participant_missing_meeting_returns_404	Pass
	mrc_pdf.py::TestBuildMeetingPdf	test_all_sections	Pass
	mrc_pdf.py::TestBuildMeetingPdf	test_buffer_not_empty	Pass
	mrc_pdf.py::TestBuildMeetingPdf	test_meeting_with_actions	Pass
	mrc_pdf.py::TestBuildMeetingPdf	test_meeting_with_agenda	Pass
	mrc_pdf.py::TestBuildMeetingPdf	test_meeting_with_decisions	Pass
	mrc_pdf.py::TestBuildMeetingPdf	test_meeting_with_minutes	Pass
	mrc_pdf.py::TestBuildMeetingPdf	test_meeting_with_no_optional_fields	Pass
	mrc_pdf.py::TestBuildMeetingPdf	test_meeting_with_participants	Pass
	mrc_pdf.py::TestBuildMeetingPdf	test_minimal_meeting	Pass
	mrc_pdf.py::TestBuildMeetingPdf	test_returns_bytesio	Pass
	mrc_pdf.py::TestGetMeetingPdf	test_meeting_not_found_returns_404	Pass
	mrc_pdf.py::TestGetMeetingPdf	test_valid_meeting_content_type_pdf	Pass
	mrc_pdf.py::TestGetMeetingPdf	test_valid_meeting_returns_bytes	Pass
	mrc_pdf.py::TestGetMeetingPdf	test_valid_meeting_returns_pdf	Pass
	raci.py::TestRACIMatrix	test_get_raci_matrix	Pass
	raci.py::TestRACIMatrix	test_get_raci_matrix_file_missing	Pass
	raci.py::TestRACIMatrix	test_update_raci_activity	Pass
	raci.py::TestRACIMatrix	test_update_raci_activity_invalid_role	Pass
	raci.py::TestRACIMatrix	test_update_raci_escalation	Pass
	raci.py::TestRACIMatrix	test_update_raci_role	Pass
	raci.py::TestRACIMatrix	test_update_raci_role_not_found	Pass
	test_bibliography_coverage.py::TestBibliographyCreateSaveFailure	test_bibliography_save_failure_returns_500	Pass
	test_bibliography_coverage.py::TestBibliographyDeleteSaveFailure	test_bibliography_save_failure_returns_500	Pass
	test_bibliography_coverage.py::TestBibliographyUpdateSaveFailure	test_bibliography_save_failure_returns_500	Pass
	test_block2_coverage_part1.py::TestMrcCreateMeetingSaveFail	test_mrc_create_meeting_save_failure	Pass
	test_block2_coverage_part1.py::TestMrcGetMeetingDocumentSend	test_mrc_get_meeting_document_not_found	Pass
	test_block2_coverage_part1.py::TestMrcGetMeetingDocumentSend	test_mrc_get_meeting_document_success	Pass
	test_block2_coverage_part1.py::TestMrcUploadDocumentUpload	test_mrc_upload_document_empty_filename	Pass
	test_block2_coverage_part1.py::TestMrcUploadDocumentUpload	test_mrc_upload_document_meeting_not_found	Pass
	test_block2_coverage_part1.py::TestMrcUploadDocumentUpload	test_mrc_upload_document_no_file	Pass
	test_block2_coverage_part1.py::TestMrcUploadDocumentUpload	test_mrc_upload_document_save_failure	Pass
	test_block2_coverage_part1.py::TestMrcUploadDocumentUpload	test_mrc_upload_document_success	Pass
	test_block2_coverage_part1.py::TestMrcUploadDocumentUpload	test_mrc_upload_creates_documents_dir	Pass
	test_block2_coverage_part1.py::TestMrcUpdateMeetingSaveFail	test_mrc_update_meeting_save_failure	Pass
	test_block2_coverage_part2.py::TestAuditLogHelpersUsage	test_audit_log_overflow_trim	Pass
	test_block2_coverage_part2.py::TestAuditLogHelpersUsage	test_audit_log_inventory_not_found	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_block2_coverage_part2.py::	TestAuditLogEdgeCases::test_log_model_usage_save_failure	Pass
	test_block2_coverage_part2.py::	TestOverrideRiskRatingEdgeCases::test_overrideRiskRating_overflow	Pass
	test_block2_coverage_part2.py::	TestOverrideRiskRatingEdgeCases::test_overrideRiskRating_not_found	Pass
	test_block2_coverage_part2.py::	TestOverrideRiskRatingEdgeCases::test_overrideRiskRating_failure	Pass
	test_block2_coverage_part2.py::	TestPdfServing::test_model_doc_pdf_not_found	Pass
	test_block2_coverage_part2.py::	TestPdfServing::test_model_doc_pdf_served	Pass
	test_block2_coverage_part2.py::	TestPdfServing::test_mrm_rating_pdf_not_found	Pass
	test_block2_coverage_part2.pyy::	TestPdfServing::test_mrm_rating_pdf_served	Pass
	test_block2_coverage_part2.py::	TestPdfServing::test_results_pdf_not_found	Pass
	test_block2_coverage_part2.py::	TestPdfServing::test_results_pdf_served	Pass
	test_block2_coverage_part2.py::	TestRiskRatingEdgeCases::test_get_inventory_not_found	Pass
	test_block2_coverage_part2.py::	TestValidateQuestionEdgeCases::test_validate_question_overflow	Pass
	test_block2_coverage_part2.py::	TestValidateQuestionEdgeCases::test_validate_question_not_found	Pass
	test_block2_coverage_part2.py::	TestValidateQuestionEdgeCases::test_validate_question_failure	Pass
	test_compliance_coverage.py::	TestBt239PdfNotFound::test_bt239_pdf_not_generated_returns_404	Pass
	test_compliance_coverage.py::	TestBt239SaveFailure::test_bt239_save_failure_returns_500	Pass
	test_compliance_coverage.py::	TestActivityNoData::test_activity_no_data_returns_404	Pass
	test_compliance_coverage.py::	TestActivitySaveFailure::test_activity_save_failure_returns_500	Pass
	test_compliance_coverage.py::	TestActivityThisEmpty::test_activity_this_empty_posted	Pass
	test_compliance_coverage.py::	TestEscalationNoData::test_escalation_no_data_returns_404	Pass
	test_compliance_coverage.py::	TestEscalationSaveFailure::test_escalation_save_failure_returns_500	Pass
	test_compliance_coverage.py::	TestRoleSaveFailure::test_role_save_failure_returns_500	Pass
	test_helpers_data.py::	TestLoadBt239::test_loads_valid_data	Pass
	test_helpers_data.py::	TestLoadBt239::test_returns_none_when_missing	Pass
	test_helpers_data.py::	TestLoadBibliography::test_loads_bibliography	Pass
	test_helpers_data.py::	TestLoadBibliography::test_returns_default_when_missing	Pass
	test_helpers_data.py::	TestLoadGovDocs::test_loads_documents	Pass
	test_helpers_data.py::	TestLoadGovDocs::test_returns_empty_list_when_missing	Pass
	test_helpers_data.py::	TestLoadRaci::test_loads_valid_data	Pass
	test_helpers_data.py::	TestLoadRaci::test_returns_none_when_missing	Pass
	test_helpers_data.py::	TestSaveBt239::test_returns_false_on_oserror	Pass
	test_helpers_data.py::	TestSaveBt239::test_saves_data	Pass
	test_helpers_data.py::	TestSaveBibliography::test_returns_false_on_oserror	Pass
	test_helpers_data.py::	TestSaveBibliography::test_saves_bibliography	Pass
	test_helpers_data.py::	TestSaveGovDocs::test_returns_false_on_oserror	Pass
	test_helpers_data.py::	TestSaveGovDocs::test_saves_documents	Pass
	test_helpers_data.py::	TestSaveRaci::test_returns_false_on_oserror	Pass
	test_helpers_data.py::	TestSaveRaci::test_saves_data	Pass
	test_helpers_io.py::	TestFindMeeting::test_finds_existing_meeting	Pass
	test_helpers_io.py::	TestFindMeeting::test_returns_error_for_missing_meeting	Pass
	test_helpers_io.py::	TestLoadAuditLog::test_loads_entries	Pass
	test_helpers_io.py::	TestLoadAuditLog::test_returns_empty_list_on_bad_json	Pass
	test_helpers_io.py::	TestLoadAuditLog::test_returns_empty_list_when_missing	Pass
	test_helpers_io.py::	TestLoadInventory::test_loads_valid_file	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_helpers_io.py::TestLoadInventory	test_returns_none_on_bad_json	Pass
	test_helpers_io.py::TestLoadInventory	test_returns_none_when_missing	Pass
	test_helpers_io.py::TestLoadMeetings	test_loads_list	Pass
	test_helpers_io.py::TestLoadMeetings	test_returns_empty_list_when_missing	Pass
	test_helpers_io.py::TestSaveAndRespond	test_saves_and_returns_response	Pass
	test_helpers_io.py::TestSaveAuditLog	test_returns_false_on_oserror	Pass
	test_helpers_io.py::TestSaveAuditLog	test_saves_entries	Pass
	test_helpers_io.py::TestSaveInventory	test_returns_false_on_oserror	Pass
	test_helpers_io.py::TestSaveInventory	test_saves_data	Pass
	test_helpers_io.py::TestSaveMeetings	test_returns_false_on_oserror	Pass
	test_helpers_io.py::TestSaveMeetings	test_saves_meetings	Pass
	test_helpers_risk.py::TestCalculateRiskRating	test_conditional_rating_middle_range	Pass
	test_helpers_risk.py::TestCalculateRiskRating	test_decision_gives_half_score	Pass
	test_helpers_risk.py::TestCalculateRiskRating	test_fairness_review_gives_full_score	Pass
	test_helpers_risk.py::TestCalculateRiskRating	test_high_assumptions_penalised	Pass
	test_helpers_risk.py::TestCalculateRiskRating	test_high_coverage_acceptable	Pass
	test_helpers_risk.py::TestCalculateRiskRating	test_high_limitations_penalised	Pass
	test_helpers_risk.py::TestCalculateRiskRating	test_low_coverage_unacceptable	Pass
	test_helpers_risk.py::TestCalculateRiskRating	test_no_review_date_gives_zero	Pass
	test_helpers_risk.py::TestCalculateRiskRating	test_overdue_review_gives_zero_score	Pass
	test_helpers_risk.py::TestCalculateRiskRating	test_returns_dict_with_required_keys	Pass
	test_helpers_risk.py::TestCalculateRiskRating	test_score_is_float	Pass
	test_helpers_risk.py::TestVqSummary	test_all_addressed	Pass
	test_helpers_risk.py::TestVqSummary	test_empty_questions	Pass
	test_helpers_risk.py::TestVqSummary	test_none_addressed	Pass
	test_helpers_risk.py::TestVqSummary	test_some_not_applicable	Pass
	test_inventory_integrity_part1.py::TestInventoryFile	test_inventory_file_exists	Pass
	test_inventory_integrity_part1.py::TestInventoryFile	test_inventory_valid_json	Pass
	test_inventory_integrity_part1.py::TestInventoryFile	test_inventory_lists_at_least_one_model	Pass
	test_inventory_integrity_part1.py::TestInventoryFile	test_inventory_has_metadata	Pass
	test_inventory_integrity_part1.py::TestInventoryFile	test_inventory_model_dirs_exists	Pass
	test_inventory_integrity_part1.py::TestInventoryFile	test_inventory_models_key_exists	Pass
	test_inventory_integrity_part1.py::TestAllModelDirs	test_all_models_have_model_id	Pass
	test_inventory_integrity_part1.py::TestModelIDs	test_model_ids_are_unique	Pass
	test_inventory_integrity_part1.py::TestModelIDs	test_model_ids_follow_naming_convention	Pass
	test_inventory_integrity_part1.py::TestOverallRiskRatingStructure	test_overall_risk_rating_is_valid	Pass
	test_inventory_integrity_part1.py::TestOverallRiskRatingStructure	test_overall_risk_rating_has_component_scores	Pass
	test_inventory_integrity_part1.py::TestOverallRiskRatingStructure	test_overall_risk_rating_has_required_keys	Pass
	test_inventory_integrity_part1.py::TestRequiredDirectoryFiles	test_required_dirs_exist[overall_risk_rating]	Pass
	test_inventory_integrity_part1.py::TestRequiredDirectoryFiles	test_required_dirs_exist[test_coverage]	Pass
	test_inventory_integrity_part1.py::TestTestCoverageStructure	test_test_coverage_booleans_are_bool	Pass
	test_inventory_integrity_part1.py::TestTestCoverageStructure	test_test_coverage_has_required_keys	Pass
	test_inventory_integrity_part1.py::TestTestCoverageStructure	test_test_coverage_has_test_file	Pass
	test_inventory_integrity_part2.py::TestAPIEndpointAuthentication	test_models_endpoint_returns_200	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_inventory_integrity_part2.py::	TestAllInventoryEndpoint::test_all_inventory_endpoint_succeeds	Pass
	test_inventory_integrity_part2.py::	TestAllInventoryEndpoint::test_all_inventory_model_count_matches_file	Pass
	test_inventory_integrity_part2.py::	TestFieldsIsList::test_fields_is_list[assumptions]	Pass
	test_inventory_integrity_part2.py::	TestFieldsIsList::test_fields_is_list[downstream_models]	Pass
	test_inventory_integrity_part2.py::	TestFieldsIsList::test_fields_is_list[limitations]	Pass
	test_inventory_integrity_part2.py::	TestFieldsIsList::test_fields_is_list[upstream_models]	Pass
	test_inventory_integrity_part2.py::	TestFieldsIsStringDict::test_fields_is_string_dict[category]	Pass
	test_inventory_integrity_part2.py::	TestFieldsIsStringDict::test_fields_is_string_dict[description]	Pass
	test_inventory_integrity_part2.py::	TestFieldsIsStringDict::test_fields_is_string_dict[name]	Pass
	test_inventory_integrity_part2.py::	TestFieldsIsStringDict::test_fields_is_string_dict[owner]	Pass
	test_inventory_integrity_part2.py::	TestFieldsIsStringDict::test_fields_is_string_dict[status]	Pass
	test_inventory_integrity_part2.py::	TestFieldsIsStringDict::test_fields_is_string_dict[version]	Pass
	test_lineage_coverage_part1.py::	TestCounterpartyTrace::test_counterparty_trace_not_found_returns_empty	Pass
	test_lineage_coverage_part1.py::	TestCounterpartyTrace::test_counterparty_trace_finds_master_record	Pass
	test_lineage_coverage_part1.py::	TestCounterpartyTrace::test_counterparty_trace_finds_prs_trades	Pass
	test_lineage_coverage_part1.py::	TestGaugeTraceIncludesClassifier::test_gauge_trace_includes_classifier_summary	Pass
	test_lineage_coverage_part1.py::	TestGaugeTraceIncludesGaugehd::test_gauge_trace_includes_gaugehd	Pass
	test_lineage_coverage_part1.py::	TestGaugeTraceIncludesMarketState::test_gauge_trace_includes_market_state	Pass
	test_lineage_coverage_part1.py::	TestGaugeTraceIncludesSequenceGauge::test_gauge_trace_includes_sequence_gauge	Pass
	test_lineage_coverage_part1.py::	TestGaugeTraceSequenceGaugeCleanFallback::test_gauge_trace_sequence_gauge_clean_fallback	Pass
	test_lineage_coverage_part1.py::	TestGaugeTraceSequenceGaugeLegacyFallback::test_gauge_trace_sequence_gauge_legacy_fallback	Pass
	test_lineage_coverage_part2.py::	TestPropertySubLoanIncludes::test_property_sub_loan_includes_mortgage	Pass
	test_lineage_coverage_part2.py::	TestPropertySubLoanIncludes::test_property_sub_loan_includes_propertyhd	Pass
	test_lineage_coverage_part2.py::	TestRouteErrorHandling::test_route_staleness_error_returns_empty	Pass
	test_lineage_coverage_part2.py::	TestRouteTamperHandling::test_route_tamper_error_returns_500	Pass
	test_lineage_coverage_part2.py::	TestRouteTraceErrorHandling::test_route_trace_error_returns_500	Pass
	test_lineage_coverage_part2.py::	TestTradeSubTradeIncludes::test_trade_sub_trade_includes_eod	Pass
	test_lineage_coverage_part2.py::	TestTradeSubTradeIncludes::test_trade_sub_trade_includes_marks	Pass
	test_lineage_coverage_part2.py::	TestUtilityValueCaseError::test_utility_value_case_error_falls_back_to_absolute	Pass
	test_lineage_coverage_part2.py::	TestUtilityFindCaseFiles::test_utility_find_case_files_filename_stem_match	Pass
	test_lineage_coverage_part2.py::	TestUtilityFindCaseFiles::test_utility_find_case_files_missing_dir	Pass
	test_lineage_coverage_part2.py::	TestUtilityLoadJsonSafe::test_utility_load_json_safe_exception_returns_false	Pass
	test_lineage_data_routes.py::	TestDataLineage::test_data_lineage	Pass
	test_lineage_data_routes.py::	TestDataLineage::test_data_lineage_health	Pass
	test_lineage_data_routes.py::	TestDataLineage::test_data_lineage_list	Pass
	test_lineage_data_routes.py::	TestDataLineage::test_data_lineage_statuses	Pass
	test_lineage_data_routes.py::	TestDataLineage::test_data_lineage_with_manifest	Pass
	test_lineage_data_routes.py::	TestDataLineage::test_data_lineage_{data_id}	Pass
	test_lineage_data_routes.py::	TestDataLineage::test_data_lineage_{data_type}	Pass
	test_lineage_data_routes.py::	TestDataLineage::test_data_lineage_{path:path}	Pass
	test_lineage_data_routes.py::	TestDataLineage::test_data_lineage_rebuilt_scan	Pass
	test_lineage_data_routes.py::	TestDataLineage::test_data_lineage_{id}/manifest	Pass
	test_lineage_data_routes.py::	TestDataLineage::test_data_lineage_not_found	Pass
	test_lineage_data_routes.py::	TestDataLineage::test_data_lineage_Route	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_lineage_data_routes.py::TestFieldLineageRoute	test_field_lineage_only_params	Pass
	test_lineage_field.py::TestFieldLineageLookup	test_lookup_found	Pass
	test_lineage_field.py::TestFieldLineageLookup	test_lookup_not_found	Pass
	test_lineage_field.py::TestFieldLineageLookup	test_lookup_wrong_report	Pass
	test_lineage_field.py::TestFieldLineageLookup	test_lookup_wrong_section	Pass
	test_lineage_field.py::TestFieldLineageLookup	test_no_params_400	Pass
	test_lineage_field.py::TestFieldLineageLookup	test_no_registry_404	Pass
	test_lineage_field.py::TestFieldLineageLookup	test_lookup_by_cdm_path	Pass
	test_lineage_field.py::TestFieldLineageLookup	test_lookup_by_source_field	Pass
	test_lineage_field.py::TestFieldLineageLookup	test_lookup_case_insensitive	Pass
	test_lineage_field.py::TestFieldLineageLookup	test_lookup_finds_matches	Pass
	test_lineage_field.py::TestFieldLineageLookup	test_lookup_no_results	Pass
	test_lineage_field.py::TestFieldLineageLookup	test_lookup_result_structure	Pass
	test_lineage_field.py::TestFieldLineageLookup	test_lookup_takes_priority_over_partial_exact	Pass
	test_lineage_field.py::TestGetFieldLineage	test_filter_by_report	Pass
	test_lineage_field.py::TestGetFieldLineage	test_filter_unknown_report_404	Pass
	test_lineage_field.py::TestGetFieldLineage	test_full_registry	Pass
	test_lineage_field.py::TestGetFieldLineage	test_no_registry_404	Pass
	test_lineage_field.py::TestGetFieldLineage	test_summary_contents	Pass
	test_lineage_helpers.py::TestCheckStale	test_all_missing	Pass
	test_lineage_helpers.py::TestCheckStale	test_directory_step_fresh	Pass
	test_lineage_helpers.py::TestCheckStale	test_directory_step_stale	Pass
	test_lineage_helpers.py::TestCheckStale	test_empty_directory_is_missing	Pass
	test_lineage_helpers.py::TestCheckStale	test_fresh_file	Pass
	test_lineage_helpers.py::TestCheckStale	test_hidden_files_ignored_in_directory	Pass
	test_lineage_helpers.py::TestCheckStale	test_manifest_last_run_not_overwritten_by_mtime	Pass
	test_lineage_helpers.py::TestCheckStale	test_manifest_last_run_used	Pass
	test_lineage_helpers.py::TestCheckStale	test_stale_file	Pass
	test_lineage_helpers.py::TestTraceDataGauge	test_gauge_trace_finds_multiple_steps	Pass
	test_lineage_helpers.py::TestTraceDataGauge	test_gauge_trace_has_context	Pass
	test_lineage_helpers.py::TestTraceDataGauge	test_gauge_trace_includes_classifier	Pass
	test_lineage_helpers.py::TestTraceDataGauge	test_gauge_trace_includes_prs_trades	Pass
	test_lineage_helpers.py::TestTraceDataGauge	test_none_lineage_returns_empty	Pass
	test_lineage_helpers.py::TestTraceDataGauge	test_nonexistent_gauge_returns_empty	Pass
	test_lineage_helpers.py::TestTraceDataGauge	test_property_trace_finds_steps	Pass
	test_lineage_helpers.py::TestTraceDataGauge	test_trade_trace_finds_origin	Pass
	test_lineage_staleness.py::TestStalenessRoute	test_all_fresh_full_health	Pass
	test_lineage_staleness.py::TestStalenessRoute	test_all_missing_zero_health	Pass
	test_lineage_staleness.py::TestStalenessRoute	test_has_as_of	Pass
	test_lineage_staleness.py::TestStalenessRoute	test_partial_health	Pass
	test_lineage_staleness.py::TestStalenessRoute	test_stale_does_not_count_as_fresh	Pass
	test_models_coverage.py::TestAuditLogTail	test_tail_log_truncated_to_10000	Pass
	test_models_coverage.py::TestGetModelDriftNotInventory	test_model_drift_not_inventory_returns_404	Pass
	test_models_coverage.py::TestGetModelNotInventory	test_model_not_inventory_returns_404	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_models_coverage.py::TestGetModelsNotScheduled	test_models_not_scheduled	Pass
	test_models_coverage.py::TestGetModelsOverdueReview	test_models_overdue_review	Pass
	test_models_coverage.py::TestUpdateModelInventory	test_model_inventory_404	Pass
	test_models_coverage.py::TestUpdateModelSafetyFile	test_model_safety_file_500	Pass
	test_mrc_agenda_minutes.py::TestAgendaCRUD	test_add_agenda_item	Pass
	test_mrc_agenda_minutes.py::TestAgendaCRUD	test_add_agenda_unknown_meeting_returns_404	Pass
	test_mrc_agenda_minutes.py::TestAgendaCRUD	test_add_multiple_items_increments_number	Pass
	test_mrc_agenda_minutes.py::TestAgendaCRUD	test_delete_agenda_item_renumbers	Pass
	test_mrc_agenda_minutes.py::TestAgendaCRUD	test_delete_agenda_unknown_meeting_returns_404	Pass
	test_mrc_agenda_minutes.py::TestAgendaCRUD	test_update_agenda_item	Pass
	test_mrc_agenda_minutes.py::TestAgendaCRUD	test_update_agenda_unknown_meeting_returns_404	Pass
	test_mrc_agenda_minutes.py::TestAgendaCRUD	test_update_missing_item_returns_404	Pass
	test_mrc_agenda_minutes.py::TestMinutesCRUD	test_add_minute_item	Pass
	test_mrc_agenda_minutes.py::TestMinutesCRUD	test_delete_minute_item_legacy_format_returns_400	Pass
	test_mrc_agenda_minutes.py::TestMinutesCRUD	test_update_minute_item_legacy_format_returns_400	Pass
	test_mrc_agenda_minutes.py::TestMinutesCRUD	test_update_minute_item_not_found_returns_404	Pass
	test_mrc_agenda_minutes.py::TestMinutesCRUD	test_update_minutes_text	Pass
	test_mrc_agenda_minutes.py::TestMinutesCRUD	test_update_minutes_unknown_meeting_returns_404	Pass
	test_mrc_decisions_actions.py::TestActionsCRUD	test_add_action_auto_generates_id	Pass
	test_mrc_decisions_actions.py::TestActionsCRUD	test_add_action_unknown_meeting_returns_404	Pass
	test_mrc_decisions_actions.py::TestActionsCRUD	test_delete_action	Pass
	test_mrc_decisions_actions.py::TestActionsCRUD	test_delete_action_unknown_meeting_returns_404	Pass
	test_mrc_decisions_actions.py::TestActionsCRUD	test_update_action_status	Pass
	test_mrc_decisions_actions.py::TestActionsCRUD	test_update_missing_action_returns_404	Pass
	test_mrc_decisions_actions.py::TestDecisionsCRUD	test_add_decision_auto_generates_id	Pass
	test_mrc_decisions_actions.py::TestDecisionsCRUD	test_add_decision_unknown_meeting_returns_404	Pass
	test_mrc_decisions_actions.py::TestDecisionsCRUD	test_delete_decision	Pass
	test_mrc_decisions_actions.py::TestDecisionsCRUD	test_delete_decision_unknown_meeting_returns_404	Pass
	test_mrc_decisions_actions.py::TestDecisionsCRUD	test_decisions_increments_id	Pass
	test_mrc_decisions_actions.py::TestDecisionsCRUD	test_update_decision	Pass
	test_mrc_decisions_actions.py::TestDecisionsCRUD	test_update_missing_decision_returns_404	Pass
	test_mrc_decisions_actions.py::TestParticipantsCRUD	test_add_participant_auto_generates_id	Pass
	test_mrc_decisions_actions.py::TestParticipantsCRUD	test_add_participant_unknown_meeting_returns_404	Pass
	test_mrc_decisions_actions.py::TestParticipantsCRUD	test_delete_participant	Pass
	test_mrc_decisions_actions.py::TestParticipantsCRUD	test_delete_participant_unknown_meeting_returns_404	Pass
	test_mrc_decisions_actions.py::TestParticipantsCRUD	test_update_missing_participant_returns_404	Pass
	test_mrc_decisions_actions.py::TestParticipantsCRUD	test_update_participant	Pass
	test_mrc_meetings.py::TestCreateMeeting	test_auto_generates_id	Pass
	test_mrc_meetings.py::TestCreateMeeting	test_create_returns_meeting	Pass
	test_mrc_meetings.py::TestCreateMeeting	test_custom_attendees_become_participants	Pass
	test_mrc_meetings.py::TestCreateMeeting	test_custom_title_used	Pass
	test_mrc_meetings.py::TestCreateMeeting	test_default_chair_populated	Pass
	test_mrc_meetings.py::TestCreateMeeting	test_default_participants_created	Pass
	test_mrc_meetings.py::TestCreateMeeting	test_minimal_body_uses_defaults	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_mrc_meetings.py::TestGetMeeting	test_get_created_meeting	Pass
	test_mrc_meetings.py::TestGetMeeting	test_unknown_id_returns_404	Pass
	test_mrc_meetings.py::TestListMeetings	test_created_meeting_appears_in_list	Pass
	test_mrc_meetings.py::TestListMeetings	test_empty_list_returns_success	Pass
	test_mrc_meetings.py::TestListMeetings	test_list_summary_has_required_keys	Pass
	test_mrc_meetings.py::TestListMeetings	test_sorted_by_date_desc	Pass
	test_mrc_meetings.py::TestMeetingDocs	test_serve_missing_document_returns_404	Pass
	test_mrc_meetings.py::TestMeetingDocs	test_upload_empty_filename_returns_400	Pass
	test_mrc_meetings.py::TestMeetingDocs	test_upload_no_file_returns_400	Pass
	test_mrc_meetings.py::TestMeetingDocs	test_upload_not_found_meeting_returns_404	Pass
	test_mrc_meetings.py::TestUpdateMeeting	test_update_sets_updated_at	Pass
	test_mrc_meetings.py::TestUpdateMeeting	test_update_title	Pass
	test_mrc_meetings.py::TestUpdateMeeting	test_update_unknown_id_returns_404	Pass
	test_report_routes_part1.py::TestAppendOutputLine	test_appends_line	Pass
	test_report_routes_part1.py::TestAppendOutputLine	test_clear_then_append	Pass
	test_report_routes_part1.py::TestAppendOutputLine	test_trims_buffer_at_2000	Pass
	test_report_routes_part1.py::TestReportOutputs	test_has_lines_field	Pass
	test_report_routes_part1.py::TestReportOutputs	test_has_running_field	Pass
	test_report_routes_part1.py::TestReportOutputs	test_lines_includes_appended	Pass
	test_report_routes_part1.py::TestReportOutputs	test_running_initially	Pass
	test_report_routes_part1.py::TestReportOutputs	test_returns_200	Pass
	test_report_routes_part1.py::TestReportSummary	test_not_found_when_no_file	Pass
	test_report_routes_part1.py::TestReportSummary	test_ok_when_file_exists	Pass
	test_report_routes_part1.py::TestReportSummary	test_summary_fields	Pass
	test_report_routes_part1.py::TestReportSummary	test_running_field_is_bool	Pass
	test_report_routes_part2.py::TestAppendAndSave	test_appends_started_header	Pass
	test_report_routes_part2.py::TestAppendAndSave	test_appends_line_appended_on_exception	Pass
	test_report_routes_part2.py::TestAppendAndSave	test_handles_exception_gracefully	Pass
	test_report_routes_part2.py::TestAppendAndSave	test_sets_running_false_on_completion	Pass
	test_report_routes_part2.py::TestTaskReadyRunning	test_already_running_message	Pass
	test_report_routes_part2.py::TestTaskReadyRunning	test_already_running_returns_202	Pass
	test_report_routes_part2.py::TestTestReport	test_response_has_started_status	Pass
	test_report_routes_part2.py::TestTestReport	test_start_report_run_returns_202	Pass
	test_report_routes_part2.py::TestStreamPipe	test_calls_proc_wait	Pass
	test_report_routes_part2.py::TestStreamPipe	test_reads_stdout_into_buffer	Pass
	test_report_routes_part2.py::TestStreamPipe	test_strips_newlines	Pass
	test_hazard_to_pricer.py::TestHazardCurveLine	test_curve_has_long_term_structure_returns_data	Pass
	test_hazard_to_pricer.py::TestHazardCurveLine	test_curve_line_matches_market_state_file	Pass
	test_hazard_to_pricer.py::TestHazardCurveLine	test_gaugehc_exists_with_curves	Pass
	test_hazard_to_pricer.py::TestHazardCurveLine	test_gaugehc_has_annual_hazard_rates	Pass
	test_hazard_to_pricer.py::TestHazardCurveLine	test_market_state_exists_with_hts	Pass
	test_hazard_to_pricer.py::TestHazardCurveLine	test_market_state_hts_has_all_gaugehc_gauges	Pass
	test_hazard_to_pricer.py::TestHazardCurveLine	test_market_state_hts_has_all_tenors	Pass
	test_hazard_to_pricer.py::TestHazardCurveLine	test_sequence_count_matches_storm_sequences	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_trade_to_blotter.py::TestActiveTrade	test_trade_appears_in_active_trade	Pass
	test_trade_to_blotter.py::TestBlotterSnapshot	test_new_trade_in_blotter	Pass
	test_trade_to_blotter.py::TestTradeCommitCreates	test_commit_creates_swap_id	Pass
	test_trade_to_blotter.py::TestTradeCommitWrites	test_commit_writes_to_temp_dir	Pass
	test_cluster_and_isolation.py::TestClusterIndividualStorm	test_cluster_individual_storm_ltv_computed_per_storm	Pass
	test_cluster_and_isolation.py::TestClusterIndividualStorm	test_each_individual_storm_individually_queryable	Pass
	test_cluster_and_isolation.py::TestClusterIndividualStorm	test_largest_individual_storm_has_most_damage	Pass
	test_cluster_and_isolation.py::TestClusterIndividualStorm	test_unknown_storm_from_sequence_returns_404	Pass
	test_cluster_and_isolation.py::TestNegativeEquityFromSequence	test_damage_and_ltv_computed_correctly	Pass
	test_cluster_and_isolation.py::TestNegativeEquityFromSequence	test_negative_equity_flag_set_on_property	Pass
	test_cluster_and_isolation.py::TestNegativeEquityFromSequence	test_post_damage_ltv_exceeds_100_pct	Pass
	test_cluster_and_isolation.py::TestNegativeEquityFromSequence	test_severe_storm_triggers_negative_equity	Pass
	test_cluster_and_isolation.py::TestSequenceIsolation	test_cross_sequence_no_contamination	Pass
	test_cluster_and_isolation.py::TestSequenceIsolation	test_seqA_storm_returns_only_its_damage	Pass
	test_cluster_and_isolation.py::TestSequenceIsolation	test_seqB_storm_returns_only_seqB_damage	Pass
	test_doublet_storms.py::TestDoubletFirstStorm	test_first_storm_damage_is_independent_of_second	Pass
	test_doublet_storms.py::TestDoubletFirstStorm	test_first_storm_returns_only_flooded_properties	Pass
	test_doublet_storms.py::TestDoubletPortfolioTotals	test_portfolio_totals_differ_between_storms	Pass
	test_doublet_storms.py::TestDoubletSecondStorm	test_second_storm_larger_damage_for_prop_a	Pass
	test_doublet_storms.py::TestDoubletSecondStorm	test_second_storm_returns_both_properties	Pass
	test_doublet_storms.py::TestDoubletSecondStorm	test_second_storm_response_matches_query	Pass
	test_sequence_endpoint.py::TestSequenceEndpoint	test_damage_model_is_max_depth	Pass
	test_sequence_endpoint.py::TestSequenceEndpoint	test_multi_points_per_property_reported	Pass
	test_sequence_endpoint.py::TestSequenceEndpoint	test_multi_storm_in_sequence_reported	Pass
	test_sequence_endpoint.py::TestSequenceEndpoint	test_prop_a_point_double_counted	Pass
	test_sequence_endpoint.py::TestSequenceEndpoint	test_prop_a_uses_max_depth_event	Pass
	test_sequence_endpoint.py::TestSequenceEndpoint	test_prop_b_included_via_second_storm	Pass
	test_sequence_endpoint.py::TestSequenceEndpoint	test_prop_c_excluded_from_different_sequence	Pass
	test_sequence_endpoint.py::TestSequenceEndpoint	test_sequence_damage_less_than_sum_of_storms	Pass
	test_sequence_endpoint.py::TestSequenceEndpoint	test_sequence_endpoint_options	Pass
	test_sequence_endpoint.py::TestSequenceEndpoint	test_sequence_endpoint_returns_200	Pass
	test_sequence_endpoint.py::TestSequenceEndpoint	test_sequence_id_in_response	Pass
	test_sequence_endpoint.py::TestSequenceEndpoint	test_unknown_sequence_returns_404	Pass
	test_sequence_endpoint.py::TestSequenceEndpoint	test_sequence_negative_equity_from_worst_storm	Pass
	test_sequence_endpoint.py::TestSequenceEndpoint	test_sequence_num_flooded_is_four	Pass
	test_sequence_endpoint.py::TestSequenceEndpoint	test_sequence_negative_equity_first_storm	Pass
	properties_core.py::TestGenerateReport	test_generate_report_exception_returns_500	Pass
	properties_core.py::TestGenerateReport	test_generate_report_success	Pass
	properties_core.py::TestGenerateReport	test_missing_property_id_returns_400	Pass
	properties_core.py::TestGenerateReport	test_no_body_returns_400	Pass
	properties_core.py::TestGenerateReport	test_no_report_returns_error	Pass
	properties_core.py::TestGenerateReport	test_no_reports_returns_ok	Pass
	properties_core.py::TestGenerateReport	test_property_not_found_returns_404	Pass
	properties_core.py::TestGenerateReport	test_unknown_error_returns_500	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	properties_core.py::TestGetProperty	test_found_returns_property	Pass
	properties_core.py::TestGetProperty	test_not_found_returns_404	Pass
	properties_core.py::TestListProperties	test_loader_exception_returns_500	Pass
	properties_core.py::TestListProperties	test_success_empty_list	Pass
	properties_core.py::TestListProperties	test_success_with_properties	Pass
	properties_core.py::TestNoJsonBodyEdge	test_property_report_null_json_returns_400	Pass
	properties_mortgage.py::TestGenerateMortgage	test_generate_mortgage_report_exception_returns_500	Pass
	properties_mortgage.py::TestGenerateMortgage	test_generate_mortgage_report_success	Pass
	properties_mortgage.py::TestGenerateMortgage	test_missing_property_id_returns_400	Pass
	properties_mortgage.py::TestGenerateMortgage	test_mortgage_not_found_returns_404	Pass
	properties_mortgage.py::TestGenerateMortgage	test_no_json_body_returns_400	Pass
	properties_mortgage.py::TestGenerateMortgage	test_no_mortgage_returns_error	Pass
	properties_mortgage.py::TestGenerateMortgage	test_mortgage_returns_ok	Pass
	properties_mortgage.py::TestGenerateMortgage	test_property_not_found_returns_404	Pass
	properties_mortgage.py::TestGenerateMortgage	test_mortgage_import_error_returns_500	Pass
	properties_mortgage.py::TestListMortgages	test_lists_mortgages_empty	Pass
	properties_mortgage.py::TestListMortgages	test_lists_mortgages_exception_returns_500	Pass
	properties_mortgage.py::TestListMortgages	test_lists_mortgages_success	Pass
	properties_mortgage.py::TestNoJsonBodyEdge	test_mortgage_report_null_json_returns_400	Pass
	properties_mortgage.py::TestPropertyMortgage	test_mortgage_returns_mortgage	Pass
	properties_mortgage.py::TestPropertyMortgage	test_loader_exception_returns_500	Pass
	properties_mortgage.py::TestPropertyMortgage	test_not_found_returns_404	Pass
	properties_mortgage.py::TestPropertyMortgage	test_mortgage_returns_ok	Pass
	propertyhc_part1.py::TestPropertyHazard	test_data_contains_hazard_curve	Pass
	propertyhc_part1.py::TestPropertyHazard	test_data_contains_nearest_gauges	Pass
	propertyhc_part1.py::TestPropertyHazard	test_data_contains_property_id	Pass
	propertyhc_part1.py::TestPropertyHazard	test_data_contains_summary	Pass
	propertyhc_part1.py::TestPropertyHazard	test_known_property_returns_200	Pass
	propertyhc_part1.py::TestPropertyHazard	test_known_property_status_success	Pass
	propertyhc_part1.py::TestPropertyHazard	test_no_data_returns_404	Pass
	propertyhc_part1.py::TestPropertyHazard	test_no_data_status_error	Pass
	propertyhc_part1.py::TestPropertyHazard	test_options_returns_200	Pass
	propertyhc_part1.py::TestPropertyHazard	test_options_status_ok	Pass
	propertyhc_part1.py::TestPropertyHazard	test_response_has_data_key	Pass
	propertyhc_part1.py::TestPropertyHazard	test_second_property_also_works	Pass
	propertyhc_part1.py::TestPropertyHazard	test_unknown_property_message_contains_id	Pass
	propertyhc_part1.py::TestPropertyHazard	test_unknown_property_returns_404	Pass
	propertyhc_part1.py::TestPropertyHazard	test_unknown_property_status_error	Pass
	propertyhc_part1.py::TestPropertyHazard	test_distribution_avg_basis_bps	Pass
	propertyhc_part1.py::TestPropertyHazard	test_distribution_avg_flood_count	Pass
	propertyhc_part1.py::TestPropertyHazard	test_distribution_max_depth_m	Pass
	propertyhc_part1.py::TestPropertyHazard	test_distribution_num_properties	Pass
	propertyhc_part1.py::TestPropertyHazard	test_summary_message_mentions_command	Pass
	propertyhc_part1.py::TestPropertyHazard	test_summary_returns_404	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	propertyhc_part1.py::TestPropertyEmptyCurves	test_no_data_status_is_error	Pass
	propertyhc_part1.py::TestPropertyEmptyCurves	test_options_returns_200	Pass
	propertyhc_part1.py::TestPropertyEmptyCurves	test_options_status_ok	Pass
	propertyhc_part1.py::TestPropertyEmptyCurves	test_response_has_data_key	Pass
	propertyhc_part1.py::TestPropertyEmptyCurves	test_response_has_distribution	Pass
	propertyhc_part1.py::TestPropertyEmptyCurves	test_response_has_metadata	Pass
	propertyhc_part1.py::TestPropertyEmptyCurves	test_response_has_summary	Pass
	propertyhc_part1.py::TestPropertyEmptyCurves	test_returns_200_with_data	Pass
	propertyhc_part1.py::TestPropertyEmptyCurves	test_status_is_success	Pass
	propertyhc_part1.py::TestPropertyEmptyCurves	test_summary_avg_flood_count_is_zero	Pass
	propertyhc_part1.py::TestPropertyEmptyCurves	test_summary_num_properties_is_zero	Pass
	propertyhc_part1.py::TestPropertyEmptyCurves	test_summary_returns_200	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_basis_entry_has_avg_basis_bps	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_basis_entry_has_flood_count	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_basis_entry_has_nearest_gauges	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_basis_entry_has_property_id	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_basis_table_count_matches	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_basis_table_length	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_basis_table_sorted_desc_by_basis	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_nearest_gauge_has_distance_km	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_nearest_gauge_has_gauge_id	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_no_data_returns_404	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_no_data_status_error	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_options_returns_200	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_options_status_ok	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_response_has_basis_table	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_response_has_count	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_returns_200_with_data	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_status_is_success	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_summary_basis_table_is_list	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_summary_count_is_zero	Pass
	propertyhc_part2.py::TestPropertyEmptyBasis	test_summary_returns_200	Pass
	propertyhc_shd_she.py::TestPropertyShd	test_no_data_returns_404	Pass
	propertyhc_shd_she.py::TestPropertyShd	test_options_returns_200	Pass
	propertyhc_shd_she.py::TestPropertyShd	test_unknown_property_returns_404	Pass
	propertyhc_shd_she.py::TestPropertyShd	test_unknown_property_status_error	Pass
	propertyhc_shd_she.py::TestPropertyShd	test_valid_property_has_data	Pass
	propertyhc_shd_she.py::TestPropertyShd	test_valid_property_returns_200	Pass
	propertyhc_shd_she.py::TestPropertyShd	test_valid_property_status_success	Pass
	propertyhc_shd_she.py::TestPropertyShe	test_no_data_returns_404	Pass
	propertyhc_shd_she.py::TestPropertyShe	test_options_returns_200	Pass
	propertyhc_shd_she.py::TestPropertyShe	test_unknown_property_returns_404	Pass
	propertyhc_shd_she.py::TestPropertyShe	test_unknown_property_status_error	Pass
	propertyhc_shd_she.py::TestPropertyShe	test_valid_property_has_data	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	propertyhc_shd_she.py::TestProperty	test_valid_property_returns_200	Pass
	propertyhc_shd_she.py::TestProperty	test_valid_property_status_success	Pass
	propertyhc_shd_she.py::TestProperty	test_no_data_returns_404	Pass
	propertyhc_shd_she.py::TestProperty	test_no_data_status_error	Pass
	propertyhc_shd_she.py::TestProperty	test_options_returns_200	Pass
	propertyhc_shd_she.py::TestProperty	test_options_status_ok	Pass
	propertyhc_shd_she.py::TestProperty	test_response_has_metadata	Pass
	propertyhc_shd_she.py::TestProperty	test_response_has_summary	Pass
	propertyhc_shd_she.py::TestProperty	test_returns_200_with_data	Pass
	propertyhc_shd_she.py::TestProperty	test_status_is_success	Pass
	propertyhc_shd_she.py::TestProperty	test_no_data_returns_404	Pass
	propertyhc_shd_she.py::TestProperty	test_no_data_status_error	Pass
	propertyhc_shd_she.py::TestProperty	test_options_returns_200	Pass
	propertyhc_shd_she.py::TestProperty	test_options_status_ok	Pass
	propertyhc_shd_she.py::TestProperty	test_response_has_metadata	Pass
	propertyhc_shd_she.py::TestProperty	test_response_has_summary	Pass
	propertyhc_shd_she.py::TestProperty	test_returns_200_with_data	Pass
	propertyhc_shd_she.py::TestProperty	test_status_is_success	Pass
	claim_part1.py::TestClaimReportGenerator	test_generator_error_has_message	Pass
	claim_part1.py::TestClaimReportGenerator	test_generator_error_returns_500	Pass
	claim_part1.py::TestClaimReportGenerator	test_generator_error_status_is_error	Pass
	claim_part1.py::TestClaimReportHappyPath	test_content_length_header_set	Pass
	claim_part1.py::TestClaimReportHappyPath	test_filename_contains_prop_id	Pass
	claim_part1.py::TestClaimReportHappyPath	test_response_body_is_pdf_bytes	Pass
	claim_part1.py::TestClaimReportHappyPath	test_response_has_content_disposition	Pass
	claim_part1.py::TestClaimReportHappyPath	test_response_is_pdf_content_type	Pass
	claim_part1.py::TestClaimReportHappyPath	test_success_returns_200	Pass
	claim_part1.py::TestClaimReportNotDir	test_no_dir_message_mentions_propertyts	Pass
	claim_part1.py::TestClaimReportNotDir	test_no_dir_returns_404	Pass
	claim_part1.py::TestClaimReportNotDir	test_no_dir_status_error	Pass
	claim_part1.py::TestClaimReportNotFloodEvents	test_no_flood_events_message_mentions_events	Pass
	claim_part1.py::TestClaimReportNotFloodEvents	test_no_flood_events_returns_404	Pass
	claim_part1.py::TestClaimReportNotFloodEvents	test_no_flood_events_status_error	Pass
	claim_part1.py::TestClaimReportNotFound	test_unknown_property_message_contains_id	Pass
	claim_part1.py::TestClaimReportNotFound	test_unknown_property_returns_404	Pass
	claim_part1.py::TestClaimReportNotFound	test_unknown_property_status_error	Pass
	claim_part2.py::TestClaimReportHappyPath	test_error_sequences_file_handled	Pass
	claim_part2.py::TestClaimReportHappyPath	test_missing_mortgage_json_still_generates	Pass
	claim_part2.py::TestClaimReportHappyPath	test_missing_property_json_still_generates	Pass
	claim_part2.py::TestClaimReportHappyPath	test_missing_sequences_file_still_generates	Pass
	core_enrichment_filtering.py::TestFloodStandardEnrichmentFiltering	test_standard_enrichment_filtering_continues	Pass
	core_enrichment_filtering.py::TestFloodStandardEnrichmentFiltering	test_property_enrichment_filtering_continues	Pass
	core_enrichment_filtering.py::TestFloodStandardEnrichmentFiltering	test_property_enrichment_filtering_skipped	Pass
	core_enrichment_filtering.py::TestFloodStandardEnrichmentFiltering	test_storm_flood_depth_filtering	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	core_enrichment_storms.py::TestFloodStormsEnrichment	test_gauges_enrichment_continues	Pass
	core_enrichment_storms.py::TestFloodStormsEnrichment	test_properties_flooding_enrichment	Pass
	core_enrichment_storms.py::TestFloodStormsEnrichment	test_storms_with_empty_storm_id_skipped	Pass
	core_enrichment_storms.py::TestFloodStormsEnrichment	test_storms_metadata_from_storms_json	Pass
	core_enrichment_storms.py::TestFloodStormsEnrichment	test_storms_severity_classification_all_levels	Pass
	core_storms.py::TestListFloodStorms	test_no_storms_file_returns_404	Pass
	core_storms.py::TestListFloodStorms	test_options_returns_ok	Pass
	core_storms.py::TestListFloodStorms	test_storm_has_expected_fields	Pass
	core_storms.py::TestListFloodStorms	test_storm_name_from_intensity_category	Pass
	core_storms.py::TestListFloodStorms	test_storms_list_count	Pass
	core_storms.py::TestListFloodStorms	test_storms_sorted_by_gauges_severe	Pass
	core_storms.py::TestListFloodStorms	test_with_data_returns_success	Pass
	core_storms.py::TestPropertyStormsEnrich	test_storms_file_without_gauge_json	Pass
	core_storms.py::TestPropertyStormsEnrich	test_storms_file_without_storm_sequences_json	Pass
	core_summary.py::TestPropertyFloods	test_existing_property_returns_success	Pass
	core_summary.py::TestPropertyFloods	test_include_readings_false_strips_readings	Pass
	core_summary.py::TestPropertyFloods	test_include_readings_true_keeps_readings	Pass
	core_summary.py::TestPropertyFloods	test_missing_property_returns_404	Pass
	core_summary.py::TestPropertyFloods	test_options_returns_ok	Pass
	core_summary.py::TestPropertyStorms	test_existing_property_returns_success	Pass
	core_summary.py::TestPropertyStorms	test_flood_events_tagged_with_sequence_type	Pass
	core_summary.py::TestPropertyStorms	test_missing_property_returns_404	Pass
	core_summary.py::TestPropertyStorms	test_nearest_gauges_enriched_with_flood_stages	Pass
	core_summary.py::TestPropertyStorms	test_options_returns_ok	Pass
	core_summary.py::TestPropertyStorms	test_response_has_flood_events	Pass
	core_summary.py::TestPropertyStorms	test_response_has_nearest_gauges	Pass
	core_summary.py::TestPropertyStorms	test_response_has_property_info	Pass
	core_summary.py::TestSummary	test_no_data_returns_404	Pass
	core_summary.py::TestSummary	test_options_returns_ok	Pass
	core_summary.py::TestSummary	test_with_data_returns_success	Pass
	financial.py::TestPortfolioImpact	test_known_storm_returns_success	Pass
	financial.py::TestPortfolioImpact	test_mortgaged_property_has_ltv_fields	Pass
	financial.py::TestPortfolioImpact	test_no_pts_dir_returns_404	Pass
	financial.py::TestPortfolioImpact	test_options_returns_ok	Pass
	financial.py::TestPortfolioImpact	test_portfolio_damage_pct_non_negative	Pass
	financial.py::TestPortfolioImpact	test_portfolio_totals_count	Pass
	financial.py::TestPortfolioImpact	test_properties_sorted_by_damage_descending	Pass
	financial.py::TestPortfolioImpact	test_property_entry_has_financial_fields	Pass
	financial.py::TestPortfolioImpact	test_response_has_portfolio	Pass
	financial.py::TestPortfolioImpact	test_response_has_properties_list	Pass
	financial.py::TestPortfolioImpact	test_response_has_storm_id	Pass
	financial.py::TestPortfolioImpact	test_unknown_storm_returns_404	Pass
	financial.py::TestSequencePortfolioImpact	test_known_sequence_returns_success	Pass
	financial.py::TestSequencePortfolioImpact	test_no_pts_dir_returns_404	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	financial.py::TestSequencePortfolio	test_options_returns_ok	Pass
	financial.py::TestSequencePortfolio	test_properties_sorted_by_damage_descending	Pass
	financial.py::TestSequencePortfolio	test_property_entry_has_financial_fields	Pass
	financial.py::TestSequencePortfolio	test_property_entry_has_num_sequence_floods	Pass
	financial.py::TestSequencePortfolio	test_property_entry_has_worst_storm_id	Pass
	financial.py::TestSequencePortfolio	test_response_has_damage_model	Pass
	financial.py::TestSequencePortfolio	test_response_has_num_storms	Pass
	financial.py::TestSequencePortfolio	test_response_has_portfolio	Pass
	financial.py::TestSequencePortfolio	test_response_has_properties_list	Pass
	financial.py::TestSequencePortfolio	test_response_has_sequence_id	Pass
	financial.py::TestSequencePortfolio	test_unknown_sequence_returns_404	Pass
	risk_part1.py::TestPortfolioVarEtest	test_missing_sequences_message_mentions_doublet	Pass
	risk_part1.py::TestPortfolioVarEtest	test_missing_sequences_returns_404	Pass
	risk_part1.py::TestPortfolioVarEtest	test_no_pts_dir_returns_404	Pass
	risk_part1.py::TestPortfolioVarEtest	test_no_pts_dir_status_error	Pass
	risk_part1.py::TestPortfolioVarEtest	test_options_returns_ok	Pass
	risk_part1.py::TestPortfolioVarNtest	test_no_mortgage_impairment_when_above_water	Pass
	risk_part1.py::TestPortfolioVarNtest	test_prob_loss_pct_is_100	Pass
	risk_part1.py::TestPortfolioVarNtest	test_property_damage_cond_mean_positive	Pass
	risk_part1.py::TestPortfolioVarNtest	test_property_damage_max_is_positive	Pass
	risk_part1.py::TestPortfolioVarNtest	test_storm_count_is_one	Pass
	risk_part1.py::TestPortfolioVarNtest	test_storms_with_damage_is_one	Pass
	risk_part1.py::TestPortfolioVarNtest	test_storms_zero_plus_with_damage_equals_total	Pass
	risk_part1.py::TestPortfolioVarNtest	test_total_portfolio_mortgages	Pass
	risk_part1.py::TestPortfolioVarNtest	test_total_portfolio_value	Pass
	risk_part1.py::TestPortfolioVarStest	test_histogram_bin_has_lo_hi_count	Pass
	risk_part1.py::TestPortfolioVarStest	test_mort_histogram_has_50_bins	Pass
	risk_part1.py::TestPortfolioVarStest	test_mortgage_impairment_sub_keys	Pass
	risk_part1.py::TestPortfolioVarStest	test_prop_histogram_has_50_bins	Pass
	risk_part1.py::TestPortfolioVarStest	test_property_damage_sub_keys	Pass
	risk_part1.py::TestPortfolioVarStest	test_returns_200	Pass
	risk_part1.py::TestPortfolioVarStest	test_status_success	Pass
	risk_part1.py::TestPortfolioVarStest	test_tail_storm_has_required_fields	Pass
	risk_part1.py::TestPortfolioVarStest	test_tail_storms_is_list	Pass
	risk_part1.py::TestPortfolioVarStest	test_tail_storms_sorted_descending	Pass
	risk_part1.py::TestPortfolioVarStest	test_top_level_keys_present	Pass
	risk_part2.py::TestPortfolioVarMissing	test_missing_mortgage_json_returns_200	Pass
	risk_part2.py::TestPortfolioVarMissing	test_missing_mortgage_json_zero_portfolio_mortgages	Pass
	risk_part2.py::TestPortfolioVarMissing	test_missing_property_json_returns_200	Pass
	risk_part2.py::TestPortfolioVarMissing	test_missing_property_json_status_success	Pass
	risk_part2.py::TestPortfolioVarMissing	test_missing_property_json_zero_portfolio_value	Pass
	risk_part2.py::TestPortfolioVarMortgage	test_mortgage_impairment_max_is_positive	Pass
	risk_part2.py::TestPortfolioVarMortgage	test_mortgage_impairment_value	Pass
	risk_part2.py::TestPortfolioVarMortgage	test_tail_storm_has_mortgage_impairment	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	risk_part2.py::TestPortfolioVarMultiple	test_prob_loss_pct_is_50	Pass
	risk_part2.py::TestPortfolioVarMultiple	test_storm_count_is_two	Pass
	risk_part2.py::TestPortfolioVarMultiple	test_storms_with_damage_is_one	Pass
	risk_part2.py::TestPortfolioVarMultiple	test_storms_zero_damage_is_one	Pass
	risk_part2.py::TestPortfolioVarMultiple	test_storms_contains_the_damaged_storm	Pass
	risk_part2.py::TestPortfolioVarMultiple	test_storms_limited_to_50	Pass
	risk_part2.py::TestPortfolioVarUtils	test_prop_value_from_known_property	Pass
	risk_part2.py::TestPortfolioVarUtils	test_unknown_prop_id_is_skipped	Pass
	risk_part2.py::TestPortfolioVarZero	test_zero_depth_cond_mean_is_zero	Pass
	risk_part2.py::TestPortfolioVarZero	test_zero_depth_prob_loss_is_zero	Pass
	risk_part2.py::TestPortfolioVarZero	test_zero_depth_property_damage_max_is_zero	Pass
	risk_part2.py::TestPortfolioVarZero	test_zero_depth_storms_with_damage_is_zero	Pass
	test_animate_composite.py::TestAnimateCompositeEvents	test_flood_events_at_all_returns_404	Pass
	test_animate_composite.py::TestAnimateCompositeEvents	test_flood_events_returns_404	Pass
	test_animate_composite.py::TestAnimateCompositeEvents	test_composite_returns_404	Pass
	test_animate_composite.py::TestAnimateCompositeEvents	test_composite_status_error	Pass
	test_animate_composite.py::TestAnimateCompositeEvents	test_composite_returns_ok	Pass
	test_animate_composite.py::TestAnimateCompositeFrame	test_has_ranges_and_edges	Pass
	test_animate_composite.py::TestAnimateCompositeFrame	test_has_hour_field	Pass
	test_animate_composite.py::TestAnimateCompositeFrame	test_has_properties_structure	Pass
	test_animate_composite.py::TestAnimateCompositeFrame	test_has_stats_structure	Pass
	test_animate_composite.py::TestAnimateCompositeProperty	test_composite_property_stale_true_at_peak_time	Pass
	test_animate_composite.py::TestAnimateCompositeProperty	test_composite_property_stale_false_after_peak_time	Pass
	test_animate_composite.py::TestAnimateCompositeProperty	test_property_beyond_requested_hours_defaults	Pass
	test_animate_composite.py::TestAnimateCompositeProperty	test_property_does_not_have_wse_m	Pass
	test_animate_composite.py::TestAnimateCompositeProperty	test_property_has_peak_fields	Pass
	test_animate_composite.py::TestAnimateCompositeProperty	test_property_has_required_fields	Pass
	test_animate_composite.py::TestAnimateCompositeSeries	test_series_length	Pass
	test_animate_composite.py::TestAnimateCompositeSeries	test_series_equals_storm_hours	Pass
	test_animate_composite.py::TestAnimateCompositeSeries	test_properties_affected_is_one	Pass
	test_animate_composite.py::TestAnimateCompositeSuccess	test_returns_200	Pass
	test_animate_composite.py::TestAnimateCompositeSuccess	test_status_success	Pass
	test_animate_composite.py::TestAnimateCompositeSuccess	test_storm_id_is_composite	Pass
	test_animate_composite.py::TestAnimateCompositeWithHigher	test_composite_wss_higher_depth_storm	Pass
	test_animate_storm_basic.py::TestAnimateStormError	test_storm_returns_404	Pass
	test_animate_storm_basic.py::TestAnimateStormError	test_storm_status_error	Pass
	test_animate_storm_basic.py::TestAnimateStormSuccess	test_storm_returns_ok	Pass
	test_animate_storm_basic.py::TestAnimateUnknownError	test_unknown_error_message_mentions_storm_id	Pass
	test_animate_storm_basic.py::TestAnimateUnknownError	test_unknown_error_returns_404	Pass
	test_animate_storm_basic.py::TestAnimateUnknownError	test_unknown_error_status_error	Pass
	test_animate_storm_basic.py::TestAnimateStormFrame	test_frame_has_ranges_and_edges	Pass
	test_animate_storm_basic.py::TestAnimateStormFrame	test_frame_has_hour_field	Pass
	test_animate_storm_basic.py::TestAnimateStormFrame	test_frame_has_properties_structure	Pass
	test_animate_storm_basic.py::TestAnimateStormFrame	test_frame_has_stats_structure	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_animate_storm_basic.py::TestAnimateStorm	test_frame_hour_zero_is_first	Pass
	test_animate_storm_basic.py::TestAnimateStorm	test_hour_values_sequential	Pass
	test_animate_storm_basic.py::TestAnimateStorm	test_frames_list_length_equals_storm_hours	Pass
	test_animate_storm_basic.py::TestAnimateStorm	test_frames_equals_storm_hours	Pass
	test_animate_storm_basic.py::TestAnimateStorm	test_properties_affected_is_one	Pass
	test_animate_storm_basic.py::TestAnimateStorm	test_metric_250	Pass
	test_animate_storm_basic.py::TestAnimateStorm	test_status_success	Pass
	test_animate_storm_basic.py::TestAnimateStorm	test_storm_id_in_response	Pass
	test_animate_storm_state.py::TestAnimateGauge	test_gauge_has_required_fields	Pass
	test_animate_storm_state.py::TestAnimateGauge	test_gauge_id_correct	Pass
	test_animate_storm_state.py::TestAnimateGauge	test_status_critical_when_above_alert	Pass
	test_animate_storm_state.py::TestAnimateGauge	test_status_normal_when_below_alert	Pass
	test_animate_storm_state.py::TestAnimateGauge	test_status_severe	Pass
	test_animate_storm_state.py::TestAnimateGauge	test_status_warning	Pass
	test_animate_storm_state.py::TestAnimateGauge	test_water_level_key_fallback	Pass
	test_animate_storm_state.py::TestAnimateGauge	test_with_no_readings_defaults_to_zero	Pass
	test_animate_storm_state.py::TestAnimateGauge	test_water_level_from_readings	Pass
	test_animate_storm_state.py::TestAnimateProperty	test_affected_flag_before_arrival_time	Pass
	test_animate_storm_state.py::TestAnimateProperty	test_affected_true_after_arrival_time	Pass
	test_animate_storm_state.py::TestAnimateProperty	test_property_beyond_readings_defaults_to_zero	Pass
	test_animate_storm_state.py::TestAnimateProperty	test_property_has_required_fields	Pass
	test_animate_storm_state.py::TestAnimateProperty	test_property_id_correct	Pass
	test_animate_storm_state.py::TestAnimateStorm	test_gauges_flooded_counts_non_normal	Pass
	test_animate_storm_state.py::TestAnimateStorm	test_gauges_flooded_zero_when_normal	Pass
	test_animate_storm_state.py::TestAnimateStorm	test_properties_flooded_is_positive_early_on	Pass
	test_animate_storm_state.py::TestAnimateStorm	test_water_depth_is_non_negative	Pass
	test_financial_coverage.py::TestLoadMortgageLoan	test_mortgage_loan_exception_still_returns_data	Pass
	test_financial_coverage.py::TestLoadPropertyMissing	test_property_missing_property_json_returns_404	Pass
	test_financial_coverage.py::TestLoadPropertyMissing	test_property_missing_property_sequence_returns_404	Pass
	test_financial_coverage.py::TestPropertyNotInPortfolio	test_property_not_in_portfolio_returns_404	Pass
	test_financial_coverage.py::TestPropertyNotInPortfolio	test_property_not_in_portfolio_returns_404	Pass
	test_financial_coverage.py::TestPropertyWithoutMortgage	test_property_without_mortgage_returns_zero_fill	Pass
	test_financial_coverage.py::TestPropertyWithoutMortgage	test_property_without_mortgage_sequence	Pass
	routes_part1.py::TestCORS	test_cors_headers_on_get	Pass
	routes_part1.py::TestCORS	test_options_returns_ok	Pass
	routes_part1.py::TestGaugeRoutes	test_generate_report_missing_gauge_id	Pass
	routes_part1.py::TestGaugeRoutes	test_get_gauge	Pass
	routes_part1.py::TestGaugeRoutes	test_get_gauge_not_found	Pass
	routes_part1.py::TestGaugeRoutes	test_get_gauge_statistics	Pass
	routes_part1.py::TestGaugeRoutes	test_get_gaugets	Pass
	routes_part1.py::TestGaugeRoutes	test_legacy_list_gauges	Pass
	routes_part1.py::TestGaugeRoutes	test_list_gauges	Pass
	routes_part1.py::TestHealthRoutes	test_detailed_health_check	Pass
	routes_part1.py::TestHealthRoutes	test_health_check	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	routes_part1.py::TestHealthRoutes	test_root_redirects	Pass
	routes_part1.py::TestPropertyRoutes	test_generate_report_missing_body	Pass
	routes_part1.py::TestPropertyRoutes	test_generate_report_missing_property_id	Pass
	routes_part1.py::TestPropertyRoutes	test_generate_report_property_not_found	Pass
	routes_part1.py::TestPropertyRoutes	test_get_property	Pass
	routes_part1.py::TestPropertyRoutes	test_get_property_not_found	Pass
	routes_part1.py::TestPropertyRoutes	test_legacy_generate_property_report	Pass
	routes_part1.py::TestPropertyRoutes	test_list_properties	Pass
	routes_part1.py::TestPropertyRoutes	test_options_preflight	Pass
	routes_part2.py::TestErrorHandling	test_invalid_json	Pass
	routes_part2.py::TestErrorHandling	test_missing_content_type	Pass
	routes_part2.py::TestGaugeCrudRoutes	test_list_gauges_error_returns_500	Pass
	routes_part2.py::TestGaugeCrudRoutes	test_list_gauges_options	Pass
	routes_part2.py::TestServerMain	test_main_calls_app_run	Pass
	consistency.py::TestStormIdConsistency	test_storm_from_list_is_runnable	Pass
	consistency.py::TestStormIdConsistency	test_worst_storm_auto_selection_is_runnable	Pass
	consistency.py::TestStressRunInputsFields	test_storm_id_exact_match_required	Pass
	consistency.py::TestStressRunInputsFields	test_storm_list_has_all_run_inputs	Pass
	hydrograph.py::TestActualHydrographFunction	test_function_decays_after_peak	Pass
	hydrograph.py::TestActualHydrographFunction	test_function_reaches_peak	Pass
	hydrograph.py::TestActualHydrographFunction	test_function_returns_168_hours	Pass
	hydrograph.py::TestActualHydrographFunction	test_function_starts_at_base	Pass
	hydrograph.py::TestActualHydrographFunction	test_peak_hour_matches_peak_position	Pass
	hydrograph.py::TestGaugetsHydrographFunction	test_gauges_level_field_fallback	Pass
	hydrograph.py::TestGaugetsHydrographFunction	test_gauges_waterlevel_field	Pass
	hydrograph.py::TestGaugetsHydrographFunction	test_missing_gaugets_returns_404	Pass
	ordering.py::TestPeakOrdering	test_severe_storm_alpha_is_first	Pass
	ordering.py::TestPeakOrdering	test_storms_sorted_worst_first	Pass
	ordering.py::TestPeakOrdering	test_worst_storm_is_first	Pass
	presentation.py::TestPresentationConsistency	test_all_three_use_same_storms_endpoint	Pass
	presentation.py::TestPresentationConsistency	test_gauge_stress_auto_selects_worst_case	Pass
	presentation.py::TestPresentationConsistency	test_gauge_stress_tab_uses_all_required_fields	Pass
	presentation.py::TestPresentationConsistency	test_historical_tab_uses_all_required_fields	Pass
	presentation.py::TestPresentationConsistency	test_stand_level_m_in_trading_stress_label	Pass
	presentation.py::TestPresentationConsistency	test_trading_stress_auto_selects_worst_case	Pass
	presentation.py::TestPresentationConsistency	test_trading_stress_tab_uses_all_required_fields	Pass
	response_fields.py::TestStressRunResponseFields	test_response_has_alert_warning_severe_levels	Pass
	response_fields.py::TestStressRunResponseFields	test_response_has_hydrograph_source_info	Pass
	response_fields.py::TestStressRunResponseFields	test_response_has_storm_name_and_category	Pass
	response_fields.py::TestStressRunResponseFields	test_response_hourly_count_is_168	Pass
	response_fields.py::TestStressRunResponseFields	test_response_probability_surface_has_full_168h	Pass
	response_fields.py::TestStressStorms	test_no_gauge_id_response_has_count	Pass
	response_fields.py::TestStressStorms	test_no_gauge_id_returns_200_not_400	Pass
	response_fields.py::TestStressStorms	test_no_gauge_id_returns_all_storms	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	response_fields.py::TestStressStorm	test_no_gauge_id_returns_success	Pass
	response_fields.py::TestStressStorm	test_with_gauge_id_still_filters	Pass
	scenario_fields.py::TestScenarioVisitedFields	test_field_list_has_base_level_m	Pass
	scenario_fields.py::TestScenarioVisitedFields	test_field_list_has_duration_hours	Pass
	scenario_fields.py::TestScenarioVisitedFields	test_field_list_has_effective_precipitation_mm	Pass
	scenario_fields.py::TestScenarioVisitedFields	test_field_list_has_gauges_severe	Pass
	scenario_fields.py::TestScenarioVisitedFields	test_field_list_has_intensity_category	Pass
	scenario_fields.py::TestScenarioVisitedFields	test_field_list_has_level_change_m	Pass
	scenario_fields.py::TestScenarioVisitedFields	test_field_list_has_max_trigger	Pass
	scenario_fields.py::TestScenarioVisitedFields	test_field_list_has_name	Pass
	scenario_fields.py::TestScenarioVisitedFields	test_field_list_has_peak_level_m	Pass
	scenario_fields.py::TestScenarioVisitedFields	test_field_list_has_peak_position	Pass
	test_health_coverage.py::TestStaticAssets	test_apple_touch_icon	Pass
	test_health_coverage.py::TestStaticAssets	test_apple_touch_icon_precomposed	Pass
	test_health_coverage.py::TestStaticAssets	test_favicon_ico	Pass
	test_propertyhc_storm_details.py::TestGaugeThresholdsInHazardResponse	test_gauge_thresholds_in_hazard_response	Pass
	test_propertyhc_storm_details.py::TestGaugeThresholdsInHazardResponse	test_gauge_thresholds_in_hazard_response	Pass
	test_propertyhc_storm_details.py::TestGaugeThresholdsInHazardResponse	test_gauge_thresholds_in_hazard_response	Pass
	test_propertyhc_storm_details.py::TestGaugeThresholdsInHazardResponse	test_gauge_thresholds_in_hazard_response	Pass
	test_propertyhc_storm_details.py::TestStormDetailsInHazardResponse	test_storm_details_in_hazard_response	Pass
	test_propertyhc_storm_details.py::TestStormDetailsInHazardResponse	test_storm_details_in_hazard_response	Pass
	test_propertyhc_storm_details.py::TestStormDetailsInHazardResponse	test_storm_details_in_hazard_response	Pass
	test_propertyhc_storm_details.py::TestStormDetailsInHazardResponse	test_storm_details_in_hazard_response	Pass
	test_propertyhc_storm_details.py::TestStormDetailsInHazardResponse	test_storm_details_in_hazard_response	Pass
	test_prs_basic.py::TestCommitTrade	test_commit_with_cashflows	Pass
	test_prs_basic.py::TestCommitTrade	test_commit_with_explicit_maturity	Pass
	test_prs_basic.py::TestCommitTrade	test_commit_with_terrain_fields	Pass
	test_prs_basic.py::TestCommitTrade	test_commit_without_terrain_fields_backward_compat	Pass
	test_prs_basic.py::TestCommitTrade	test_empty_json_returns_400	Pass
	test_prs_basic.py::TestCommitTrade	test_minimal_valid_commit	Pass
	test_prs_basic.py::TestCommitTrade	test_no_json_returns_error	Pass
	test_prs_basic.py::TestGetTradePDF	test_existing_pdf_served	Pass
	test_prs_basic.py::TestGetTradePDF	test_missing_pdf_returns_404	Pass
	test_prs_basic.py::TestListTrades	test_empty_dir_returns_zero_count	Pass
	test_prs_basic.py::TestListTrades	test_returns_trades_list_key	Pass
	test_prs_basic.py::TestListTradesWithFiles	test_returns_trades_from_json_files	Pass
	test_prs_generate_part1.py::TestGenerateTradePDF	test_close_out_payable_direction	Pass
	test_prs_generate_part1.py::TestGenerateTradePDF	test_generate_pdf_no_cashflows	Pass
	test_prs_generate_part1.py::TestGenerateTradePDF	test_generate_pdf_with_cashflows	Pass
	test_prs_generate_part1.py::TestGenerateTradePDF	test_generate_pdf_with_close_out	Pass
	test_prs_generate_part1.py::TestGenerateTradePDF	test_generate_pdf_with_close_out_date	Pass
	test_prs_generate_part2.py::TestCommitCloseOut	test_close_out_path_exercised	Pass
	test_prs_generate_part2.py::TestCommitValidation	test_invalid_fails_returns_400	Pass
	test_prs_generate_part2.py::TestListTradesEmptyDir	test_trade_file_returns_500	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_prs_terrain_payload.py::TestCDMRecordTerrainFields	test_has_ea_flood_zone	Pass
	test_prs_terrain_payload.py::TestCDMRecordTerrainFields	test_has_ea_flood_zone_actual	Pass
	test_prs_terrain_payload.py::TestCDMRecordTerrainFields	test_has_terrain_delta_bps	Pass
	test_prs_terrain_payload.py::TestCDMRecordTerrainFields	test_data_get	Pass
	test_prs_terrain_payload.py::TestCamelPayloadJS	test_backward_compat_fallback	Pass
	test_prs_terrain_payload.py::TestCamelPayloadJS	test_ea_flood_zone_actual_in_payload	Pass
	test_prs_terrain_payload.py::TestCamelPayloadJS	test_ea_flood_zone_in_payload	Pass
	test_prs_terrain_payload.py::TestCamelPayloadJS	test_terrain_delta_bps_in_payload	Pass
	blotter_routes_part1.py::TestBlotterFilters	test_closed_trades_by_default	Pass
	blotter_routes_part1.py::TestBlotterFilters	test_include_closed	Pass
	blotter_routes_part1.py::TestCloseOut	test_closeout_trade_has_final_pnl	Pass
	blotter_routes_part1.py::TestCloseOut	test_closeout_trade_marks_closed	Pass
	blotter_routes_part1.py::TestCloseOut	test_closeout_trade_not_found	Pass
	blotter_routes_part1.py::TestCloseOut	test_closeout_trade_requires_closeout_spread	Pass
	blotter_routes_part1.py::TestCloseOut	test_closeout_trade_settlement_uses_closeout_spread	Pass
	blotter_routes_part1.py::TestCloseOut	test_closeout_trade_success	Pass
	blotter_routes_part1.py::TestGetBlotter	test_blotter_empty_when_no_trades	Pass
	blotter_routes_part1.py::TestGetBlotter	test_blotter_fs01_sign_payer_positive	Pass
	blotter_routes_part1.py::TestGetBlotter	test_blotter_fs01_sign_receiver_negative	Pass
	blotter_routes_part1.py::TestGetBlotter	test_blotter_gauge_name_injected	Pass
	blotter_routes_part1.py::TestGetBlotter	test_blotter_has_pnl_fields	Pass
	blotter_routes_part1.py::TestGetBlotter	test_blotter_has_prev_fair_spread	Pass
	blotter_routes_part1.py::TestGetBlotter	test_blotter_has_required_fields	Pass
	blotter_routes_part1.py::TestGetBlotter	test_blotter_summary_fields	Pass
	blotter_routes_part1.py::TestGetBlotter	test_get_blotter_returns_trades	Pass
	blotter_routes_part2.py::TestActiveGauges	test_all_gauges_with_trades_are_returned	Pass
	blotter_routes_part2.py::TestActiveGauges	test_empty_blotter_returns_empty_list	Pass
	blotter_routes_part2.py::TestActiveGauges	test_gauge_without_trade_is_not_returned	Pass
	blotter_routes_part2.py::TestActiveGauges	test_lambeth_appears_when_it_has_an_open_trade	Pass
	blotter_routes_part2.py::TestActiveGauges	test_lambeth_excluded_after_its_trade_is_closed	Pass
	blotter_routes_part2.py::TestActiveGauges	test_other_gauges_unaffected_by_lambeth_closeout	Pass
	blotter_routes_part2.py::TestActiveGauges	test_response_is_sorted	Pass
	blotter_routes_part2.py::TestBlotterErrorHandling	test_active_gauges_corrupt_file_skipped	Pass
	blotter_routes_part2.py::TestBlotterErrorHandling	test_blotter_engine_error_returns_500	Pass
	blotter_routes_part2.py::TestBlotterErrorHandling	test_closeout_engine_error_returns_500	Pass
	blotter_routes_part2.py::TestBlotterErrorHandling	test_closeout_pdf_regen_exception_caught	Pass
	blotter_routes_part2.py::TestBlotterGaugeNameInjection	test_gauge_name_injected_when_missing	Pass
	curves_routes_part1.py::TestHazardTSCommit	test_commit_hazard_ts	Pass
	curves_routes_part1.py::TestHazardTSCommit	test_commit_hazard_ts_missing_gauge	Pass
	curves_routes_part1.py::TestHazardTSCommit	test_commit_hazard_ts_missing_rates	Pass
	curves_routes_part1.py::TestHazardTSCommit	test_commit_hazard_ts_only_affects_target_gauge	Pass
	curves_routes_part1.py::TestHazardTSReset	test_reset_hazard_ts_all	Pass
	curves_routes_part1.py::TestHazardTSReset	test_reset_hazard_ts_single_gauge	Pass
	curves_routes_part1.py::TestHazardTSReset	test_reset_hazard_term_structure	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	curves_routes_part1.py::TestHazardCurve	test_put_hazard_ts_invalid_tenor	Pass
	curves_routes_part1.py::TestHazardCurve	test_put_hazard_ts_missing_gauge	Pass
	curves_routes_part1.py::TestHazardCurve	test_put_hazard_ts_single_point	Pass
	curves_routes_part1.py::TestYieldCurve	test_get_yield_curve	Pass
	curves_routes_part1.py::TestYieldCurve	test_get_yield_curve_default_shape	Pass
	curves_routes_part1.py::TestYieldCurve	test_put_yield_curve_invalid_tenor	Pass
	curves_routes_part1.py::TestYieldCurve	test_put_yield_curve_no_body	Pass
	curves_routes_part1.py::TestYieldCurve	test_put_yield_curve_single_tenor	Pass
	curves_routes_part1.py::TestYieldCurve	test_commit_yield_curve	Pass
	curves_routes_part1.py::TestYieldCurve	test_commit_yield_curve_affected_count	Pass
	curves_routes_part1.py::TestYieldCurve	test_commit_yield_curve_empty_rates	Pass
	curves_routes_part1.py::TestYieldCurve	test_commit_yield_curve_no_body	Pass
	curves_routes_part1.py::TestYieldCurve	test_reset_yield_curve	Pass
	curves_routes_part2.py::TestCurveHistory	test_curve_history_empty_when_no_eods	Pass
	curves_routes_part2.py::TestCurveHistory	test_curve_history_requires_gauge_id	Pass
	curves_routes_part2.py::TestCurveHistory	test_curve_history_with_eod	Pass
	curves_routes_part2.py::TestCurveHandler	test_commit_hazard_ts_error_returns_500	Pass
	curves_routes_part2.py::TestCurveHandler	test_curve_history_error_returns_500	Pass
	curves_routes_part2.py::TestCurveHandler	test_get_hazard_ts_error_returns_500	Pass
	curves_routes_part2.py::TestCurveHandler	test_get_yield_curve_error_returns_500	Pass
	curves_routes_part2.py::TestCurveHandler	test_pnl_series_error_returns_500	Pass
	curves_routes_part2.py::TestCurveHandler	test_reset_hazard_ts_error_returns_500	Pass
	curves_routes_part2.py::TestCurveHandler	test_reset_yield_curve_error_returns_500	Pass
	curves_routes_part2.py::TestCurveHandler	test_update_yield_curve_error_returns_500	Pass
	curves_routes_part2.py::TestCurveNullBody	test_commit_hazard_ts_null_body	Pass
	curves_routes_part2.py::TestCurveNullBody	test_commit_yield_curve_null_body	Pass
	curves_routes_part2.py::TestCurveNullBody	test_update_hazard_ts_null_body	Pass
	curves_routes_part2.py::TestCurveNullBody	test_update_yield_curve_null_body	Pass
	curves_routes_part2.py::TestPnLSeries	test_pnl_series_after_eod	Pass
	curves_routes_part2.py::TestPnLSeries	test_pnl_series_empty	Pass
	test_classifiers_batch.py::TestBatchTraining	test_status_eta_fallback_from_elapsed	Pass
	test_classifiers_batch.py::TestBatchTraining	test_status_eta_none_when_zero_completed	Pass
	test_classifiers_batch.py::TestBatchTraining	test_train_all_all_already_trained	Pass
	test_classifiers_batch.py::TestBatchTraining	test_train_all_already_running	Pass
	test_classifiers_batch.py::TestBatchTraining	test_train_all_complete_includes_results	Pass
	test_classifiers_batch.py::TestBatchTraining	test_train_all_starts	Pass
	test_classifiers_batch.py::TestBatchTraining	test_train_all_status_fields	Pass
	test_classifiers_batch.py::TestBatchTraining	test_train_all_status_idle	Pass
	test_classifiers_batch.py::TestClassifierReadiness	test_readiness_all_trained	Pass
	test_classifiers_batch.py::TestClassifierReadiness	test_readiness_error	Pass
	test_classifiers_batch.py::TestClassifierReadiness	test_readiness_no_models	Pass
	test_classifiers_batch.py::TestClassifierReadiness	test_readiness_partial	Pass
	test_classifiers_batch.py::TestClearAllClassifiers	test_clear_all_classifiers_empty	Pass
	test_classifiers_batch.py::TestClearAllClassifiers	test_clear_all_classifiers_error	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_classifiers_batch.py::TestClearAllClassifiers	test_clear_removes_models	Pass
	test_classifiers_coverage.py::TestBatchCancelAfterIteration	test_cancel_after_iteration_returns_early	Pass
	test_classifiers_coverage.py::TestBatchCancelBeforeFirstChunk	test_cancel_before_first_chunk_when_cancelled_before_start	Pass
	test_classifiers_coverage.py::TestBatchReadsMetrics	test_batch_reads_metrics_from_summary	Pass
	test_classifiers_coverage.py::TestClearAllEmpty	test_clear_all_succeeds_with_import_error	Pass
	test_classifiers_coverage.py::TestStatusAvgAuc	test_status_avg_auc_when_no_metrics	Pass
	test_classifiers_coverage.py::TestStatusCorrectness	test_status_computes_avg_auc_and_accuracy	Pass
	test_classifiers_coverage.py::TestTrainAllSimulator	test_train_all_returns_500_on_setup_error	Pass
	test_classifiers_helpers.py::TestSummaryEdgeCases	test_avg_auc_null_when_no_trained	Pass
	test_classifiers_helpers.py::TestSummaryEdgeCases	test_gauge_classifieds_present	Pass
	test_classifiers_helpers.py::TestSummaryEdgeCases	test_mixed_trained_avg_auc	Pass
	test_classifiers_helpers.py::TestSummaryEdgeCases	test_summary_error	Pass
	test_classifiers_helpers.py::TestSummaryEdgeCases	test_trained_without_auc	Pass
	test_classifiers_helpers.py::TestTimingHelpers	test_save_gauge_seconds	Pass
	test_classifiers_helpers.py::TestTimingHelpers	test_save_gauge_seconds_empty	Pass
	test_classifiers_helpers.py::TestTimingHelpers	test_load_timings_corrupt_json	Pass
	test_classifiers_helpers.py::TestTimingHelpers	test_load_timings_missing_file	Pass
	test_classifiers_helpers.py::TestTimingHelpers	test_load_timings_valid	Pass
	test_classifiers_helpers.py::TestTimingHelpers	test_save_timings_truncates_to_20	Pass
	test_classifiers_summary.py::TestClassifiersSummary	test_avg_auc_computed	Pass
	test_classifiers_summary.py::TestClassifiersSummary	test_gauge_fields	Pass
	test_classifiers_summary.py::TestClassifiersSummary	test_gauges_sorted_by_longitude	Pass
	test_classifiers_summary.py::TestClassifiersSummary	test_no_models_initially	Pass
	test_classifiers_summary.py::TestClassifiersSummary	test_returns_all_gauges	Pass
	test_classifiers_summary.py::TestClassifiersSummary	test_returns_success	Pass
	test_classifiers_summary.py::TestClassifiersSummary	test_trained_gauge_shows_metrics	Pass
	test_classifiers_summary.py::TestSetModelClear	test_clear_all_removes_all_metrics	Pass
	test_classifiers_summary.py::TestSetModelPresent	test_model_present_shows_metrics	Pass
	test_classifiers_summary.py::TestSetModelSupersedes	test_no_model_supersedes_stale_metrics	Pass
	test_classifiers_training.py::TestRunBatchTraining	test_batch_training_cancelled_mid_run	Pass
	test_classifiers_training.py::TestRunBatchTraining	test_batch_training_with_failure	Pass
	test_classifiers_training.py::TestRunBatchTraining	test_successful_training_batch	Pass
	test_port_stress_coverage.py::TestAlertThreshold	test_alert_threshold_classification	Pass
	test_port_stress_coverage.py::TestClosedTrapezoid	test_closed_trapezoid_excluded_from_stress	Pass
	test_port_stress_coverage.py::TestEmptyStormId	test_empty_storm_id_returns_400	Pass
	test_port_stress_coverage.py::TestHistogramFromGauge	test_histogram_from_gauge_falls_back	Pass
	test_port_stress_coverage.py::TestHistogramFromGauge	test_histogram_from_gauge_enough_readings	Pass
	test_port_stress_coverage.py::TestPortfolioSingleUse	test_portfolio_single_use_ratio	Pass
	test_port_stress_coverage.py::TestPortfolioUnexpectedException	test_portfolio_unexpected_exception_returns_500	Pass
	test_port_stress_coverage.py::TestStormNameFallback	test_storm_name_fallback_name_uses_storm_id	Pass
	test_port_stress_infra.py::TestPortfolioEngineError	test_portfolio_engine_error_returns_500	Pass
	test_port_stress_infra.py::TestPortfolioMissingStorms	test_portfolio_missing_storms_file_returns_404	Pass
	test_port_stress_infra.py::TestPortfolioJsonBody	test_portfolio_json_body_returns_error	Pass
	test_port_stress_infra.py::TestStormCacheReload	test_storm_cache_not_reloaded_when_mtime_unchanged	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_port_stress_infra.py::TestStressCache	test_cache_reloads_on_mtime_change	Pass
	test_port_stress_run.py::TestPortfolioStress	test_stress_pnl_positive_notional	Pass
	test_port_stress_run.py::TestPortfolioStress	test_stress_pnl_total_is_sum_of_gauges	Pass
	test_port_stress_run.py::TestPortfolioStress	test_stress_pnl_negative_notional	Pass
	test_port_stress_run.py::TestPortfolioStress	test_stress_pnl_breach_inches_p_flood_to_100pct	Pass
	test_port_stress_run.py::TestPortfolioStress	test_stress_pnl_formula	Pass
	test_port_stress_run.py::TestPortfolioStress	test_unaffected_gauge_has_zero_p_flood	Pass
	test_port_stress_run.py::TestRunPortfolio	test_portfolio_gauge_results_fields	Pass
	test_port_stress_run.py::TestRunPortfolio	test_portfolio_run_gauges_severe_list	Pass
	test_port_stress_run.py::TestRunPortfolio	test_portfolio_run_gauges_warning_list	Pass
	test_port_stress_run.py::TestRunPortfolio	test_portfolio_run_has_gauge_results	Pass
	test_port_stress_run.py::TestRunPortfolio	test_portfolio_run_has_portfolio_totals	Pass
	test_port_stress_run.py::TestRunPortfolio	test_portfolio_invalid_storm	Pass
	test_port_stress_run.py::TestRunPortfolio	test_portfolio_missing_storm_id	Pass
	test_port_stress_run.py::TestRunPortfolio	test_portfolio_returns_success	Pass
	test_port_stress_storms.py::TestGetPortfolioStorms	test_portfolio_storms_has_storms	Pass
	test_port_stress_storms.py::TestGetPortfolioStorms	test_portfolio_storms_success	Pass
	test_port_stress_storms.py::TestGetPortfolioStorms	test_portfolio_storms_has_required_fields	Pass
	test_port_stress_storms.py::TestGetPortfolioStorms	test_portfolio_storms_no_data	Pass
	test_port_stress_storms.py::TestGetPortfolioStorms	test_portfolio_storms_secondary_sort	Pass
	test_port_stress_storms.py::TestGetPortfolioStorms	test_portfolio_storms_sorted_by_severity	Pass
	test_port_stress_storms.py::TestPortfolioStorms	test_name_never_equals_storm_id	Pass
	test_port_stress_storms.py::TestPortfolioStorms	test_storm_name_is_display_label_not_id	Pass
	test_port_stress_storms.py::TestPortfolioStorms	test_gauge_results_severe_before_warning	Pass
	test_port_stress_storms.py::TestPortfolioStorms	test_gauge_results_warning_before_alert	Pass
	test_port_stress_storms.py::TestPortfolioStorms	test_gauges_with_trades_have_trade_details	Pass
	test_port_stress_storms.py::TestPortfolioStorms	test_trade_detail_fields	Pass
	test_port_stress_storms.py::TestPortfolioStorms	test_storm_sorts_severe_before_moderate_with_equal_severe_count	Pass
	test_stress_infra_part1.py::TestHydrographRemainingZeroEdge	test_remaining_zero	Pass
	test_stress_infra_part1.py::TestLoadStressCache	test_stress_cache_returns_cached	Pass
	test_stress_infra_part1.py::TestLoadStressCache	test_stress_cache_returns_none_when_no_cache	Pass
	test_stress_infra_part1.py::TestRunStressEdgeCases	test_stress_empty_gauge_id_returns_400	Pass
	test_stress_infra_part1.py::TestRunStressEdgeCases	test_stress_no_open_trades_returns_404	Pass
	test_stress_infra_part1.py::TestRunStressEdgeCases	test_stress_null_body_returns_400	Pass
	test_stress_infra_part1.py::TestRunStressEdgeCases	test_stress_uses_gaugets_hydrograph	Pass
	test_stress_infra_part1.py::TestStressErrorHandling	test_stress_engine_error_returns_500	Pass
	test_stress_infra_part1.py::TestStressErrorHandling	test_stress_gauges_engine_error_returns_500	Pass
	test_stress_infra_part1.py::TestStressErrorHandling	test_stress_storms_engine_error_returns_500	Pass
	test_stress_infra_part1.py::TestSynthesizeHydrograph	test_synthesize_hydrograph_peak_at_zero	Pass
	test_stress_infra_part1.py::TestSynthesizeHydrograph	test_synthesize_hydrograph_remaining_zero	Pass
	test_stress_infra_part2.py::TestLegacyFallbackPaths	test_legacy_cache_reuse	Pass
	test_stress_infra_part2.py::TestLegacyFallbackPaths	test_legacy_file_loaded_when_no_index	Pass
	test_stress_infra_part2.py::TestLegacyFallbackPaths	test_stress_cache_returns_cache	Pass
	test_stress_infra_part2.py::TestLoadSingleStormFromLegacy	test_single_storm_from_legacy_file	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_stress_run.py::TestRunStress	test_run_stress_hourly_fields	Pass
	test_stress_run.py::TestRunStress	test_run_stress_invalid_storm	Pass
	test_stress_run.py::TestRunStress	test_run_stress_knock_out	Pass
	test_stress_run.py::TestRunStress	test_run_stress_missing_params	Pass
	test_stress_run.py::TestRunStress	test_run_stress_no_body	Pass
	test_stress_run.py::TestRunStress	test_run_stress_portfolio_summary	Pass
	test_stress_run.py::TestRunStress	test_run_stress_probability_surface	Pass
	test_stress_run.py::TestRunStress	test_run_stress_returns_hourly	Pass
	test_stress_run.py::TestRunStress	test_run_stress_trade_summary	Pass
	test_stress_run.py::TestRunStress	test_surface_caps_at_severe	Pass
	test_stress_run.py::TestRunStress	test_surface_shows_full_168h	Pass
	test_stress_scenario_coverage.py::TestHydrographPadding	test_hydrograph_bad_gauge_returns_404	Pass
	test_stress_scenario_coverage.py::TestHydrographPadding	test_hydrograph_pads_to_storm_hours	Pass
	test_stress_scenario_coverage.py::TestModelAucNone	test_model_auc_is_none	Pass
	test_stress_scenario_coverage.py::TestNoOpenTradesAtGauge	test_gauge_with_no_trades_returns_404	Pass
	test_stress_scenario_coverage.py::TestPFloodProbUsed	test_pflood_values_in_range	Pass
	test_stress_scenario_coverage.py::TestPFloodProbUsed	test_severe_probability_latches_to_one	Pass
	test_stress_scenario_coverage.py::TestStressStormNotFound	test_stress_storms_not_found	Pass
	test_stress_storms.py::TestGetStorms	test_get_storms_has_peak_level	Pass
	test_stress_storms.py::TestGetStorms	test_get_storms_has_storm_fields	Pass
	test_stress_storms.py::TestGetStorms	test_get_storms_no_data	Pass
	test_stress_storms.py::TestGetStorms	test_get_storms_no_gauge_id_returns_all_storms	Pass
	test_stress_storms.py::TestGetStorms	test_get_storms_no_matching_gauge	Pass
	test_stress_storms.py::TestGetStorms	test_get_storms_returns_list	Pass
	test_stress_storms.py::TestGetStorms	test_get_storms_sorted_by_peak	Pass
	test_stress_storms.py::TestGetStreamgauges	test_get_gauges_has_required_fields	Pass
	test_stress_storms.py::TestGetStreamgauges	test_get_gauges_has_trade_counts	Pass
	test_stress_storms.py::TestGetStreamgauges	test_get_gauges_returns_list	Pass
	test_stress_storms.py::TestGetStreamgauges	test_get_gauges_sorted_by_longitude	Pass
	test_training_status.py::TestClassifyStatus	test_status_failed_from_job_dict	Pass
	test_training_status.py::TestClassifyStatus	test_status_failed_no_error_key	Pass
	test_training_status.py::TestClassifyStatus	test_status_joblib_takes_precedence_over_job_dict	Pass
	test_training_status.py::TestClassifyStatus	test_status_not_trained_when_no_joblib_no_job	Pass
	test_training_status.py::TestClassifyStatus	test_status_ready_from_job_dict	Pass
	test_training_status.py::TestClassifyStatus	test_status_ready_when_joblib_exists	Pass
	test_training_status.py::TestClassifyStatus	test_status_training_in_progress	Pass
	test_training_status.py::TestClassifyStatus	test_status_unknown_gauge_not_trained	Pass
	test_training_status.py::TestEdgeCases	test_gauge_json_dict_format	Pass
	test_training_status.py::TestEdgeCases	test_train_does_not_start_if_joblib_created_between_check_and_	Pass
	test_training_status.py::TestEdgeCases	test_training_jobs_isolated_between_gauges	Pass
	test_training_summary.py::TestUpdateTrainingSummary	test_all_skipped_avg_auc_is_zero	Pass
	test_training_summary.py::TestUpdateTrainingSummary	test_creates_new_summary	Pass
	test_training_summary.py::TestUpdateTrainingSummary	test_merges_into_existing_summary	Pass
	test_training_summary.py::TestUpdateTrainingSummary	test_replaces_existing_gauge_entry	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_training_summary.py::TestUpdateTrainingSummary	test_skipped_gauge_not_counted_in_auc	Pass
	test_training_train_part1.py::TestTrainFor	test_already_in_progress	Pass
	test_training_train_part1.py::TestTrainFor	test_already_trained_returns_ready	Pass
	test_training_train_part1.py::TestTrainFor	test_corrupt_gauge_json_500	Pass
	test_training_train_part1.py::TestTrainFor	test_gauge_not_found_404	Pass
	test_training_train_part1.py::TestTrainFor	test_no_gauge_json_404	Pass
	test_training_train_part1.py::TestTrainFor	test_no_storm_sequences_404	Pass
	test_training_train_part1.py::TestTrainFor	test_previously_failed_can_restart	Pass
	test_training_train_part1.py::TestTrainFor	test_starts_background_thread	Pass
	test_training_train_part2.py::TestTrainSingle	test_failure_updates_job_to_failed	Pass
	test_training_train_part2.py::TestTrainSingle	test_success_updates_job_to_ready	Pass
	visualization.py::TestServeVisualization	test_get_returns_json_error	Pass
	visualization.py::TestServeVisualization	test_no_gauge_data_returns_error	Pass
	visualization.py::TestServeVisualization	test_no_gauge_data_returns_error_status	Pass
	visualization.py::TestServeVisualization	test_catchment_exception_returns_500	Pass
	visualization.py::TestServeVisualization	test_name_case_insensitive	Pass
	visualization.py::TestServeVisualization	test_name_redirects	Pass
	visualization.py::TestServeVisualization	test_unknown_catchment_returns_400	Pass
	visualization.py::TestServeVisualization	test_unsupported_catchment_message	Pass
	visualization.py::TestServeVisualization	test_viz_import_error_returns_503	Pass
	visualization.py::TestServeVisualization	test_viz_success_returns_200	Pass
	base_loader.py::TestBaseLoaderCaching	test_caching_prevents_reload	Pass
	base_loader.py::TestBaseLoaderCaching	test_clear_cache	Pass
	base_loader.py::TestBaseLoaderCaching	test_force_reload_bypasses_cache	Pass
	base_loader.py::TestBaseLoaderCaching	test_invalidate_cache	Pass
	base_loader.py::TestBaseLoaderInterface	test_abstract_methods_enforced	Pass
	base_loader.py::TestBaseLoaderInterface	test_subclass_must_implement_get_entity_id	Pass
	base_loader.py::TestBaseLoaderInterface	test_subclass_must_implement_get_entity_summary	Pass
	base_loader.py::TestBaseLoaderJsonLoading	test_file_exists_check	Skip
	base_loader.py::TestBaseLoaderJsonLoading	test_load_alternative_container_key	Pass
	base_loader.py::TestBaseLoaderJsonLoading	test_load_dict_with_container_key	Pass
	base_loader.py::TestBaseLoaderJsonLoading	test_load_list_format	Pass
	base_loader.py::TestBaseLoaderJsonLoading	test_load_missing_file	Pass
	base_loader.py::TestBaseLoaderJsonLoading	test_load_single_object	Pass
	base_loader.py::TestBaseLoaderQueries	test_count	Pass
	base_loader.py::TestBaseLoaderQueries	test_exists	Pass
	base_loader.py::TestBaseLoaderQueries	test_find_by_id	Pass
	base_loader.py::TestBaseLoaderQueries	test_find_by_id_not_found	Pass
	base_loader.py::TestBaseLoaderQueries	test_get_ids	Skip
	base_loader.py::TestBaseLoaderQueries	test_get_status	Skip
	base_loader.py::TestBaseLoaderQueries	test_list_all	Pass
	gauge_loader.py::TestGaugeLoaderBasics	test_entity_id_extraction	Pass
	gauge_loader.py::TestGaugeLoaderBasics	test_entity_summary	Pass
	gauge_loader.py::TestGaugeLoaderBasics	test_load_gauges	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	gauge_loader.py::TestGaugeLoaderHelpers	test_get_active_gauges	Pass
	gauge_loader.py::TestGaugeLoaderHelpers	test_get_coordinates	Pass
	gauge_loader.py::TestGaugeLoaderHelpers	test_get_flood_stages	Pass
	gauge_loader.py::TestGaugeLoaderHelpers	test_get_gauge_name	Pass
	gauge_loader.py::TestGaugeLoaderHelpers	test_get_gauge_name_not_found	Pass
	gauge_loader.py::TestGaugeLoaderQuery	test_find_by_id	Pass
	gauge_loader.py::TestGaugeLoaderQuery	test_find_by_name	Pass
	gauge_loader.py::TestGaugeLoaderQuery	test_find_by_name_case_insensitive	Pass
	gauge_loader.py::TestGaugeLoaderQuery	test_find_by_owner	Pass
	gauge_loader.py::TestGaugeLoaderQuery	test_find_by_status_active	Pass
	gauge_loader.py::TestGaugeLoaderQuery	test_find_by_status_inactive	Pass
	gauge_loader.py::TestGaugeLoaderQuery	test_find_by_type_river	Pass
	gauge_loader.py::TestGaugeLoaderQuery	test_find_by_type_tidal	Pass
	gauge_loader.py::TestGaugeLoaderSpatial	test_get_gauges_in_radius	Pass
	gauge_loader.py::TestGaugeLoaderSpatial	test_get_gauges_in_radius_none_found	Pass
	loader_registry.py::TestCreateRegistryHelper	test_create_registry	Pass
	loader_registry.py::TestCreateRegistryHelper	test_create_registry_with_overrides	Pass
	loader_registry.py::TestLoaderRegistryBasic	test_basic_registry	Pass
	loader_registry.py::TestLoaderRegistryBasic	test_get_available_loaders	Pass
	loader_registry.py::TestLoaderRegistryCaching	test_clear_all_caches	Pass
	loader_registry.py::TestLoaderRegistryCaching	test_different_loaders_separate_instances	Pass
	loader_registry.py::TestLoaderRegistryCaching	test_loader_instances_cached	Pass
	loader_registry.py::TestLoaderRegistryFileOverrides	test_file_overrides	Pass
	loader_registry.py::TestLoaderRegistryGetLoader	test_get_loader_by_name	Pass
	loader_registry.py::TestLoaderRegistryGetLoader	test_get_loader_unknown_type	Pass
	loader_registry.py::TestLoaderRegistryStatus	test_check_data_files	Pass
	loader_registry.py::TestLoaderRegistryStatus	test_get_all_status	Skip
	loader_registry.py::TestLoaderRegistryTypedAccessors	test_get_gauge_loader	Pass
	loader_registry.py::TestLoaderRegistryTypedAccessors	test_get_mortgage_loader	Pass
	loader_registry.py::TestLoaderRegistryTypedAccessors	test_get_property_loader	Pass
	loader_registry.py::TestLoaderRegistryTypedAccessors	test_get_school_loader	Pass
	loader_registry.py::TestLoaderRegistryTypedAccessors	test_get_tidal_loader	Pass
	mortgage_loader.py::TestMortgageLoaderAggregations	test_get_delinquent_mortgages	Pass
	mortgage_loader.py::TestMortgageLoaderAggregations	test_get_exposure_by_lender	Pass
	mortgage_loader.py::TestMortgageLoaderAggregations	test_get_property_ids_with_mortgages	Pass
	mortgage_loader.py::TestMortgageLoaderAggregations	test_get_total_exposure	Pass
	mortgage_loader.py::TestMortgageLoaderBasic	test_by_id_extraction	Pass
	mortgage_loader.py::TestMortgageLoaderBasic	test_by_summary	Pass
	mortgage_loader.py::TestMortgageLoaderBasic	test_load_mortgages	Pass
	mortgage_loader.py::TestMortgageLoaderQuery	test_find_by_id	Pass
	mortgage_loader.py::TestMortgageLoaderQuery	test_find_by_lender	Pass
	mortgage_loader.py::TestMortgageLoaderQuery	test_find_by_lender_case_insensitive	Pass
	mortgage_loader.py::TestMortgageLoaderQuery	test_find_by_property_id	Pass
	mortgage_loader.py::TestMortgageLoaderQuery	test_find_by_property_id_not_found	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	mortgage_loader.py::TestMortgageLoaderQuery	test_find_by_status_active	Pass
	mortgage_loader.py::TestMortgageLoaderQuery	test_find_by_status_delinquent	Pass
	mortgage_loader.py::TestMortgageLoaderPropertyLinks	test_get_mortgages_for_property	Pass
	mortgage_loader.py::TestMortgageLoaderIntegration	test_property_with_mortgage	Pass
	property_loader.py::TestPropertyLoaderAggregations	test_get_total_value	Pass
	property_loader.py::TestPropertyLoaderAggregations	test_get_value_by_state	Pass
	property_loader.py::TestPropertyLoaderBasic	test_entity_id_extraction	Pass
	property_loader.py::TestPropertyLoaderBasic	test_entity_summary	Pass
	property_loader.py::TestPropertyLoaderBasic	test_load_properties	Pass
	property_loader.py::TestPropertyLoaderGeo	test_get_coordinates	Pass
	property_loader.py::TestPropertyLoaderGeo	test_get_coordinates_not_found	Pass
	property_loader.py::TestPropertyLoaderGeo	test_get_location	Pass
	property_loader.py::TestPropertyLoaderGeo	test_get_location_not_found	Pass
	property_loader.py::TestPropertyLoaderGeo	test_get_properties_in_radius	Pass
	property_loader.py::TestPropertyLoaderQuery	test_find_by_address	Pass
	property_loader.py::TestPropertyLoaderQuery	test_find_by_address_case_insensitive	Pass
	property_loader.py::TestPropertyLoaderQuery	test_find_by_address_not_found	Pass
	property_loader.py::TestPropertyLoaderQuery	test_find_by_id	Pass
	property_loader.py::TestPropertyLoaderQuery	test_find_by_state	Pass
	property_loader.py::TestPropertyLoaderQuery	test_find_by_state_case_insensitive	Pass
	property_loader.py::TestPropertyLoaderQuery	test_find_by_zip	Pass
	property_loader.py::TestPropertyLoaderQuery	test_find_by_zip_multiple	Pass
	timeseries_loader.py::TestTimeseriesLoaderGauge	test_get_timeseries_for_portfolio_gauge	Pass
	timeseries_loader.py::TestTimeseriesLoaderGauge	test_timeseries_gauge_ids_match_portfolio	Pass
	timeseries_loader.py::TestTimeseriesLoaderBasic	test_entity_basis_timestamp	Pass
	timeseries_loader.py::TestTimeseriesLoaderBasic	test_entity_summary	Pass
	timeseries_loader.py::TestTimeseriesLoaderBasic	test_load_timeseries	Pass
	timeseries_loader.py::TestTimeseriesLoaderGauge	test_get_latest_reading	Pass
	timeseries_loader.py::TestTimeseriesLoaderGauge	test_get_peak_reading	Pass
	timeseries_loader.py::TestTimeseriesLoaderGauge	test_get_readings_for_gauge	Pass
	timeseries_loader.py::TestTimeseriesLoaderGauge	test_get_readings_for_gauge_by_name	Pass
	timeseries_loader.py::TestTimeseriesLoaderGauge	test_get_readings_for_gauge_not_found	Pass
	timeseries_loader.py::TestTimeseriesLoaderGauge	test_get_available_gauge_ids	Pass
	timeseries_loader.py::TestTimeseriesLoaderGauge	test_get_gauge_statistics	Pass
	timeseries_loader.py::TestTimeseriesLoaderGauge	test_get_gauge_statistics_not_found	Pass
	timeseries_loader.py::TestTimeseriesLoaderGauge	test_get_readings_in_range	Skip
	timeseries_loader.py::TestTimeseriesLoaderGauge	test_get_readings_in_range_for_gauge	Skip
	test_b10_quick_wins.py::TestAuditEntries	test_entries_truncated_above_10000	Pass
	test_b10_quick_wins.py::TestEODPostDerivZero	test_zero_notional_new_trade	Pass
	test_b10_quick_wins.py::TestGaugeLoaderFactory	test_directory_mode_exists	Pass
	test_b10_quick_wins.py::TestInsuranceAreaFallback	test_area_fallback_falls_through	Pass
	test_b10_quick_wins.py::TestIntensityMainMsity	test_main_msity_module_runs	Pass
	test_b10_quick_wins.py::TestIntensityGetStormFallback	test_get_storm_fallback_unknown_type_returns_1	Pass
	test_b10_quick_wins.py::TestJSONFileConfigSet	test_get_raises_on_unknown_key	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_b10_quick_wins.py::TestJSONFileConfig	test_get_returns_known_key	Pass
	test_b10_quick_wins.py::TestLtvRetentionCurve	test_ltv_is_percentage	Pass
	test_b10_quick_wins.py::TestPropertyMapping	test_unknown_type_falls_back	Pass
	test_b10_quick_wins.py::TestPropertyMapping	test_unknown_type_raises_typeerror	Pass
	test_b10_quick_wins.py::TestPropertyMapping	test_unknown_type_stringified	Pass
	test_b10_quick_wins.py::TestServerMain	test_main_block_encoded_as_float	Pass
	test_b10_quick_wins.py::TestStorageFactory	test_create_file_backend_main_callable	Pass
	test_b10_quick_wins.py::TestTrackingDashboard	test_create_file_backend_nonlinear_fallback	Pass
	test_b10_quick_wins.py::TestTrackingDashboard	test_get_statistics_returns_dimensions	Pass
	test_b10_quick_wins.py::TestVisualizationFactory	test_create_visualization_returns_instance	Pass
	data_structures.py::TestDataValidationResult	test_invalid_result_fields	Pass
	data_structures.py::TestDataValidationResult	test_is_frozen_tuple	Pass
	data_structures.py::TestDataValidationResult	test_tuple_unpacking	Pass
	data_structures.py::TestDataValidationResult	test_valid_result_fields	Pass
	data_structures.py::TestLoadedData	test_can_set_gauge_data	Pass
	data_structures.py::TestLoadedData	test_can_set_multiple_fields	Pass
	data_structures.py::TestLoadedData	test_counts_default_to_zero	Pass
	data_structures.py::TestLoadedData	test_defaults_are_none	Pass
	data_structures.py::TestLoadedData	test_is_dataclass	Pass
	data_structures.py::TestLoadedData	test_lookup_defaults_are_none	Pass
	helpers.py::TestDataLoaderHelpers	test_data_summary_contains_errors_count	Pass
	helpers.py::TestDataLoaderHelpers	test_data_summary_contains_warnings_count	Pass
	helpers.py::TestDataLoaderHelpers	test_data_summary_loaded_flag_false_when_no_file	Pass
	helpers.py::TestDataLoaderHelpers	test_data_summary_loaded_flag_true_for_gauge	Pass
	helpers.py::TestDataLoaderHelpers	test_get_data_summary_empty_before_load	Pass
	helpers.py::TestDataLoaderHelpers	test_get_data_summary_returns_dict	Pass
	helpers.py::TestDataLoaderHelpers	test_get_validation_summary_empty_before_load	Pass
	helpers.py::TestDataLoaderHelpers	test_get_validation_summary_is_copy	Pass
	helpers.py::TestDataLoaderHelpers	test_get_validation_summary_returns_dict	Pass
	helpers.py::TestDataLoaderHelpers	test_is_data_complete_depends_only_on_gauge	Pass
	helpers.py::TestDataLoaderHelpers	test_is_data_complete_false_before_load	Pass
	helpers.py::TestDataLoaderHelpers	test_is_data_complete_false_when_no_gauge	Pass
	helpers.py::TestDataLoaderHelpers	test_is_data_complete_true_with_gauge	Pass
	helpers.py::TestDataLoaderHelpers	test_validation_summary_has_gauge_key	Pass
	helpers.py::TestPrintLoadingSummary	test_summary_with_full_data_does_not_crash	Pass
	helpers.py::TestPrintLoadingSummary	test_summary_with_no_data_does_not_crash	Pass
	helpers.py::TestPrintLoadingSummary	test_summary_with_partial_data_does_not_crash	Pass
	init.py::TestDataLoaderInit	test_init_with_input_dir	Pass
	init.py::TestDataLoaderInit	test_init_with_string_path_converts_to_path	Pass
	init.py::TestDataLoaderInit	test_loaded_data_initialised	Pass
	init.py::TestDataLoaderInit	test_private_loaders_created	Pass
	init.py::TestDataLoaderInit	test_validation_results_empty_at_init	Pass
	load_all.py::TestDataLoaderLoadAll	test_counterparty_data_loaded	Pass
	load_all.py::TestDataLoaderLoadAll	test_empty_dir_returns_loaded_data	Pass
	load_all.py::TestDataLoaderLoadAll	test_gauge_data_count_matches_records	Pass
	load_all.py::TestDataLoaderLoadAll	test_gauge_data_has_items_key	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	load_all.py::TestDataLoaderLoadAll	test_gaugehd_dir_counted	Pass
	load_all.py::TestDataLoaderLoadAll	test_gaugets_dir_counted	Pass
	load_all.py::TestDataLoaderLoadAll	test_gaugets_dir_multiple_files_counted	Pass
	load_all.py::TestDataLoaderLoadAll	test_hazard_curves_loaded	Pass
	load_all.py::TestDataLoaderLoadAll	test_lookups_built_after_load	Pass
	load_all.py::TestDataLoaderLoadAll	test_no_gaugehd_dir_count_stays_zero	Pass
	load_all.py::TestDataLoaderLoadAll	test_no_gaugets_dir_count_stays_zero	Pass
	load_all.py::TestDataLoaderLoadAll	test_no_propertyts_dir_count_stays_zero	Pass
	load_all.py::TestDataLoaderLoadAll	test_property_hazard_curves_loaded	Pass
	load_all.py::TestDataLoaderLoadAll	test_property_hazard_summary_fields_accessible	Pass
	load_all.py::TestDataLoaderLoadAll	test_propertyts_dir_counted	Pass
	load_all.py::TestDataLoaderLoadAll	test_storm_data_list_format	Pass
	load_all.py::TestDataLoaderLoadAll	test_storm_data_loaded_dict_format	Pass
	load_all.py::TestDataLoaderLoadAll	test_storm_data_unknown_format	Pass
	load_all.py::TestDataLoaderLoadAll	test_storm_data_with_items_key	Pass
	load_all.py::TestDataLoaderLoadAll	test_with_gauge_data_loaded	Pass
	load_all.py::TestDataLoaderLoadAll	test_with_mortgage_data_loaded	Pass
	load_all.py::TestDataLoaderLoadAll	test_with_property_data_loaded	Pass
	loaders.py::TestLoadCounterpartyData	test_corrupt_counterparty_file_leaves_data_none	Pass
	loaders.py::TestLoadCounterpartyData	test_missing_counterparty_file_leaves_data_none	Pass
	loaders.py::TestLoadCounterpartyData	test_valid_counterparty_file_loads	Pass
	loaders.py::TestLoadHazardCurves	test_corrupt_hazard_file_leaves_hazard_data_none	Pass
	loaders.py::TestLoadHazardCurves	test_corrupt_property_hazard_file_leaves_phc_none	Pass
	loaders.py::TestLoadHazardCurves	test_missing_hazard_file_leaves_hazard_data_none	Pass
	loaders.py::TestLoadHazardCurves	test_missing_property_hazard_file_leaves_phc_none	Pass
	loaders.py::TestLoadHazardCurves	test_valid_hazard_file_populates_hazard_data	Pass
	loaders.py::TestLoadStormData	test_corrupt_storm_file_leaves_storm_data_none	Pass
	loaders.py::TestLoadStormData	test_missing_storm_file_leaves_storm_data_none	Pass
	loaders.py::TestLoadStormData	test_storms_dict_with_storms_key	Pass
	loaders.py::TestLoadStormData	test_storms_list_format	Pass
	loaders.py::TestLoadStormData	test_storms_other_type	Pass
	validation.py::TestLoadWithValidation	test_empty_data_list_records_invalid_result	Pass
	validation.py::TestLoadWithValidation	test_empty_data_list_returns_none	Pass
	validation.py::TestLoadWithValidation	test_exception_in_loader_returns_none	Pass
	validation.py::TestLoadWithValidation	test_exception_records_error_in_validation	Pass
	validation.py::TestLoadWithValidation	test_none_return_from_loader_returns_none	Pass
	validation.py::TestLoadWithValidation	test_successful_load_records_valid_result	Pass
	validation.py::TestLoadWithValidation	test_successful_load_returns_data_dict	Pass
	validation.py::TestLoadWithValidation	test_validation_summary_count_matches_items	Pass
	validation.py::TestLoadWithValidation	test_validation_summary_has_type_field	Pass
	gauge_panel.py::TestGaugeGraphInteractiveConfigure	test_configure_panel_height	Pass
	gauge_panel.py::TestGaugeGraphInteractiveConfigure	test_configure_panel_width	Pass
	gauge_panel.py::TestGaugeGraphInteractiveConfigure	test_get_statistics	Pass
	gauge_panel.py::TestGaugeHazardCurveConfigure	test_configure_panel_height	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	gauge_panel.py::TestGaugeHazardCurve	test_configure_panel_width	Pass
	gauge_panel.py::TestGaugeHazardCurve	test_get_statistics	Pass
	gauge_panel.py::TestGaugeStormAnalysis	test_configure_panel_height	Pass
	gauge_panel.py::TestGaugeStormAnalysis	test_configure_panel_width	Pass
	gauge_panel.py::TestGaugeStormAnalysis	test_get_statistics	Pass
	gauge_panel.py::TestHistoricalTab	test_historical_fetches_gauge_history	Pass
	gauge_panel.py::TestHistoricalTab	test_historical_fetches_stress_storms	Pass
	gauge_panel.py::TestHistoricalTab	test_historical_has_storm_scenarios	Pass
	gauge_panel.py::TestHistoricalTab	test_historical_has_time_series_chart	Pass
	gauge_panel.py::TestHistoricalTab	test_historical_renders	Pass
	gauge_panel.py::TestHistoricalTab	test_historical_renders_storm_list	Pass
	gauge_panel.py::TestHistoricalTab	test_historical_shows_flood_stages	Pass
	gauge_panel.py::TestHistoricalTab	test_historical_shows_trigger_colors	Pass
	gauge_panel.py::TestHistoricalTab	test_historical_storm_click_navigates_to_stress	Pass
	gauge_panel.py::TestHistoricalTab	test_historical_uses_event_listeners	Pass
	gauge_panel.py::TestStressTestTab	test_stress_fetches_storms	Pass
	gauge_panel.py::TestStressTestTab	test_stress_has_cash_price_concept	Pass
	gauge_panel.py::TestStressTestTab	test_stress_has_chart_tabs	Pass
	gauge_panel.py::TestStressTestTab	test_stress_has_storm_selector	Pass
	gauge_panel.py::TestStressTestTab	test_stress_renders	Pass
	gauge_panel.py::TestStressTestTab	test_stress_runs_scenario	Pass
	gauge_panel.py::TestStressTestTab	test_stress_shows_p_flood	Pass
	gauge_panel.py::TestStressTestTab	test_stress_storm_hint_pickup	Pass
	gauge_panel.py::TestStressTestTab	test_stress_trade_table	Pass
	gauge_panel.py::TestStressTestTab	test_stress_uses_event_listeners_for_tabs	Pass
	historical_detail_part1.py::TestHistoricalTab	test_historical_tab_renders	Pass
	historical_detail_part1.py::TestHistoryFetch	test_fetches_gauge_history	Pass
	historical_detail_part1.py::TestHistoryFetch	test_fetches_gauge_id_variable	Pass
	historical_detail_part1.py::TestStormList	test_shows_overflow_count	Pass
	historical_detail_part1.py::TestStormList	test_shows_top_30_storms	Pass
	historical_detail_part1.py::TestStormList	test_sorted_by_peak_level	Pass
	historical_detail_part1.py::TestStormPresentation	test_uses_effective_precipitation	Pass
	historical_detail_part1.py::TestStormPresentation	test_uses_gauge_severe	Pass
	historical_detail_part1.py::TestStormPresentation	test_uses_intensity_category	Pass
	historical_detail_part1.py::TestStormPresentation	test_uses_max_trigger_for_colour	Pass
	historical_detail_part1.py::TestStormPresentation	test_uses_peak_level_m	Pass
	historical_detail_part1.py::TestStormPresentation	test_uses_storm_id	Pass
	historical_detail_part1.py::TestStormPresentation	test_uses_storm_name_field	Pass
	historical_detail_part1.py::TestStormsEndpoint	test_uses_gauge_storms_endpoint	Pass
	historical_detail_part1.py::TestStormsEndpoint	test_uses_storm_scenarios	Pass
	historical_detail_part1.py::TestStormsEndpoint	test_uses_trading_stress_storms_endpoint	Pass
	historical_detail_part2.py::TestHistoricalChart	test_colour_amber	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	historical_detail_part2.py::TestHistoricalChart	test_daily_observations_field	Pass
	historical_detail_part2.py::TestHistoricalChart	test_renders_historical_chart	Pass
	historical_detail_part2.py::TestHistoricalChart	test_severe_colour_red	Pass
	historical_detail_part2.py::TestHistoricalChart	test_shows_flood_alert_threshold	Pass
	historical_detail_part2.py::TestHistoricalChart	test_shows_flood_warning_threshold	Pass
	historical_detail_part2.py::TestHistoricalChart	test_shows_severe_flood_threshold	Pass
	historical_detail_part2.py::TestHistoricalChart	test_subsamples_observations	Pass
	historical_detail_part2.py::TestHistoricalChart	test_warning_colour_orange	Pass
	historical_detail_part2.py::TestHistoricalDataCache	test_cache_keyed_by_gauge_id	Pass
	historical_detail_part2.py::TestHistoricalDataCache	test_caches_history_data	Pass
	historical_detail_part2.py::TestHistoricalDataCache	test_uses_cached_data_on_revisit	Pass
	historical_detail_part2.py::TestHistoricalStatsBar	test_stats_bar_function	Pass
	historical_detail_part2.py::TestHistoricalStatsBar	test_stats_bar_flood_exceedances	Pass
	historical_detail_part2.py::TestHistoricalStatsBar	test_stats_bar_mean_level	Pass
	historical_detail_part2.py::TestHistoricalStatsBar	test_stats_bar_record_years	Pass
	historical_detail_part2.py::TestStormClickNavigation	test_click_handler_defined	Pass
	historical_detail_part2.py::TestStormClickNavigation	test_click_handler_iife_safe	Pass
	historical_detail_part2.py::TestStormClickNavigation	test_clicking_navigates_to_tab_5	Pass
	historical_detail_part2.py::TestStormClickNavigation	test_sets_storm_hint	Pass
	historical_detail_part2.py::TestStormClickNavigation	test_storm_hint_set_before_switch_tab	Pass
	historical_detail_part2.py::TestStormClickNavigation	test_storm_id_passed_to_navigate	Pass
	historical_detail_part2.py::TestStormFieldConsistency	test_field_consistency_with_stress_tab	Pass
	historical_detail_part2.py::TestStormFieldConsistency	test_gauges_severe_consistent	Pass
	historical_detail_part2.py::TestStormFieldConsistency	test_intensity_category_consistent	Pass
	historical_detail_part2.py::TestStormFieldConsistency	test_peak_level_consistent	Pass
	historical_detail_part2.py::TestStormFieldConsistency	test_storm_id_consistent	Pass
	prs_detail.py::TestAnalyticalPrice	test_cashflow_computation	Pass
	prs_detail.py::TestAnalyticalPrice	test_survival_interpolation	Pass
	prs_detail.py::TestAnalyticalPrice	test_yield_rate_interpolation	Pass
	prs_detail.py::TestMaturityConversion	test_compute_maturity_date	Pass
	prs_detail.py::TestMaturityConversion	test_current_roll_date	Pass
	prs_detail.py::TestMaturityConversion	test_last_friday_of_month	Pass
	prs_detail.py::TestMaturityConversion	test_maturity_popup	Pass
	prs_detail.py::TestPRSControls	test_controls_function	Pass
	prs_detail.py::TestPRSControls	test_counterparty_dropdown	Pass
	prs_detail.py::TestPRSControls	test_direction_toggle	Pass
	prs_detail.py::TestPRSControls	test_maturity_selector	Pass
	prs_detail.py::TestPRSControls	test_notional_input	Pass
	prs_detail.py::TestPRSControls	test_trigger_selector	Pass
	prs_detail.py::TestPRSTabRenders	test_prs_tab_renders	Pass
	prs_detail.py::TestPRSTabRenders	test_prs_yield_curve_fetch	Pass
	prs_detail.py::TestTradeCommit	test_commit_endpoint	Pass
	prs_detail.py::TestTradeCommit	test_payer_flag_in_commit	Pass
	stress_detail.py::TestChartSubTabs	test_event_listeners_not_inline	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	stress_detail.py::TestChartSubTabs	test_flood_probability_tab	Pass
	stress_detail.py::TestChartSubTabs	test_stress_pnl_tab	Pass
	stress_detail.py::TestChartSubTabs	test_surface_tab	Pass
	stress_detail.py::TestKnockOutDisplay	test_first_trigger_hour	Pass
	stress_detail.py::TestKnockOutDisplay	test_ko_annotation_line	Pass
	stress_detail.py::TestKnockOutDisplay	test_ko_label	Pass
	stress_detail.py::TestKnockOutDisplay	test_triggered_hour_reference	Pass
	stress_detail.py::TestStormSelector	test_auto_selects_worst_storm	Pass
	stress_detail.py::TestStormSelector	test_storm_dropdown	Pass
	stress_detail.py::TestStressEndpoints	test_run_endpoint	Pass
	stress_detail.py::TestStressEndpoints	test_storms_endpoint	Pass
	stress_detail.py::TestStressTabRenders	test_stress_tab_renders	Pass
	stress_detail.py::TestSurfaceTable	test_surface_ko_trim	Pass
	stress_detail.py::TestSurfaceTable	test_surface_reads_probability_surface	Pass
	stress_detail.py::TestSurfaceTable	test_surface_renders_table	Pass
	stress_detail.py::TestSurfaceTable	test_surface_trigger_band_shading	Pass
	stress_detail.py::TestSurfaceTable	test_surface_wrap_element	Pass
	js_safety_part1.py::TestFullCompilation	test_full_js_contains_iife	Pass
	js_safety_part1.py::TestFullCompilation	test_full_js_contains_regulatory_button	Pass
	js_safety_part1.py::TestFullCompilation	test_full_js_contains_render_audit_reports	Pass
	js_safety_part1.py::TestFullCompilation	test_full_js_contains_script_tags	Pass
	js_safety_part1.py::TestFullCompilation	test_full_js_contains_show_mg_panel	Pass
	js_safety_part1.py::TestFullCompilation	test_full_js_contains_terminal_popup	Pass
	js_safety_part1.py::TestFullCompilation	test_full_js_is_str	Pass
	js_safety_part1.py::TestFullCompilation	test_full_js_non_empty	Pass
	js_safety_part1.py::TestFullCompilation	test_full_js_passes_node_syntax_check	Pass
	js_safety_part1.py::TestGetJsRetrieval	test_all_modules_non_empty	Pass
	js_safety_part1.py::TestGetJsRetrieval	test_all_modules_return_str	Pass
	js_safety_part1.py::TestNoLiteralNewlinesInJsStrings	test_all_modules_no_newline_in_js_strings	Pass
	js_safety_part1.py::TestNoLiteralNewlinesInJsStrings	test_audit_reports_no_newline_in_js_strings	Pass
	js_safety_part1.py::TestNoLiteralNewlinesInJsStrings	test_full_compiled_js_no_newline_in_js_strings	Pass
	js_safety_part2.py::TestAuditReportsCount	test_audit_terminal_refresh_exposed_on_window	Pass
	js_safety_part2.py::TestAuditReportsCount	test_download_button	Pass
	js_safety_part2.py::TestAuditReportsCount	test_poll_output_endpoint	Pass
	js_safety_part2.py::TestAuditReportsCount	test_refresh_audit_tab_button	Pass
	js_safety_part2.py::TestAuditReportsCount	test_run_test_suite_exposed_on_window	Pass
	js_safety_part2.py::TestAuditReportsCount	test_run_test_suite_function	Pass
	js_safety_part2.py::TestAuditReportsCount	test_terminal_overlay_id	Pass
	js_safety_part2.py::TestAuditReportsCount	test_toggle_failure_detail_exposed_on_window	Pass
	test_backend_handler.py::TestAddToMap	test_add_to_map_calls_add_child	Pass
	test_backend_handler.py::TestBackendHandlerInit	test_handler_init	Pass
	test_backend_handler.py::TestBackendHandlerInit	test_handler_init_uses_config	Pass
	test_backend_handler.py::TestBackendHandlerInit	test_handler_lists_are_copied	Pass
	test_backend_handler.py::TestConfigure	test_configure_no_args_no_change	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_backend_handler.py::TestConfigure	test_configure_updates_endpoints	Pass
	test_backend_handler.py::TestConfigure	test_configure_updates_server_url	Pass
	test_backend_handler.py::TestGetJs	test_contains_timeout	Pass
	test_backend_handler.py::TestGetJs	test_returns_script_tag	Pass
	test_backend_handler.py::TestGetStatistics	test_statistics_keys	Pass
	test_backend_handler.py::TestGetStatistics	test_statistics_single_endpoint	Pass
	test_context_menus.py::TestAddToMap	test_add_to_map_calls_add_child	Pass
	test_context_menus.py::TestConfigure	test_configure_both_menus	Pass
	test_context_menus.py::TestConfigure	test_configure_gauge_menu	Pass
	test_context_menus.py::TestConfigure	test_configure_no_args_no_change	Pass
	test_context_menus.py::TestConfigure	test_configure_property_menu	Pass
	test_context_menus.py::TestContextMenus	test_custom_menus	Pass
	test_context_menus.py::TestContextMenus	test_default_init	Pass
	test_context_menus.py::TestGetStatistics	test_custom_statistics	Pass
	test_context_menus.py::TestGetStatistics	test_default_statistics	Pass
	test_context_menus.py::TestNavMenusJS	test_nav_menus_included_in_get_js	Pass
	test_context_menus.py::TestNavMenusJS	test_nav_menus_js_contains_gauge_select	Pass
	test_context_menus.py::TestNavMenusJS	test_nav_menus_js_contains_property_select	Pass
	test_context_menus.py::TestNavMenusJS	test_nav_menus_js_filters_synthetic_gauges	Pass
	test_context_menus.py::TestNavMenusJS	test_nav_menus_js_has_action_bar	Pass
	test_context_menus.py::TestNavMenusJS	test_nav_menus_js_includes_menu_actions	Pass
	test_manager.py::TestConfigure	test_configure_menus	Pass
	test_manager.py::TestConfigure	test_configure_notification_position	Pass
	test_manager.py::TestConfigure	test_configure_notification_timeout	Pass
	test_manager.py::TestConfigure	test_configure_returns_self	Pass
	test_manager.py::TestConfigure	test_configure_server_url	Pass
	test_manager.py::TestGetStatistics	test_returns_dict_with_expected_keys	Pass
	test_manager.py::TestGetStatistics	test_statistics_are_dicts	Pass
	test_manager.py::TestSetupMap	test_adds_all_components	Pass
	test_manager.py::TestSetupMap	test_copyright_element_added	Pass
	test_manager.py::TestSetupMap	test_returns_self	Pass
	test_nav_menus.py::TestNavMenuActionLine	test_nav_actions_match_context_menu	Pass
	test_nav_menus.py::TestNavMenuActionLine	test_nav_menu_items_serialized_from_python	Pass
	test_nav_menus.py::TestNavMenuActionLine	test_nav_actions_match_context_menu	Pass
	test_nav_menus.py::TestNavMenuGaugeDataLoader	test_gauge_data_loader_has_synth_gauges	Pass
	test_nav_menus.py::TestNavMenuGaugeDataLoader	test_gauge_data_loader_returns_gaugeId_field	Pass
	test_nav_menus.py::TestNavMenuGaugeDataLoader	test_gauge_data_loader_uses_gaugeId_not_gauge_id	Pass
	test_nav_menus.py::TestNavMenuGaugeDataLoader	test_gauge_data_loader_uses_gauge_name_field	Pass
	test_nav_menus.py::TestNavMenuPropertyDataLoader	test_property_data_loader_correct_preloader_path	Pass
	test_nav_menus.py::TestNavMenuPropertyDataLoader	test_property_data_loader_uses_property_id_field	Pass
	test_nav_menus.py::TestNavMenuPropertyDataLoader	test_property_data_loader_has_property_id	Pass
	test_nav_menus.py::TestNavMenuSyntheticFilter	test_synthetic_filter_uses_correct_prefix	Pass
	test_notifications.py::TestAddNotificationsToMap	test_adds_to_map	Pass
	test_notifications.py::TestAddNotificationsToMap	test_creates_notification_system	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_notifications.py::TestAddToMap	test_add_to_map_calls_add_child	Pass
	test_notifications.py::TestBuildPosition	test_returns_position_function	Pass
	test_notifications.py::TestBuildTemplate	test_contains_all_types	Pass
	test_notifications.py::TestBuildTemplate	test_includes_template_values	Pass
	test_notifications.py::TestBuildTemplate	test_returns_valid_json	Pass
	test_notifications.py::TestConfigure	test_configure_max_visible	Pass
	test_notifications.py::TestConfigure	test_configure_no_args_no_change	Pass
	test_notifications.py::TestConfigure	test_configure_position	Pass
	test_notifications.py::TestConfigure	test_configure_timeout	Pass
	test_notifications.py::TestCreateNotification	test_custom_max_visible	Pass
	test_notifications.py::TestCreateNotification	test_custom_timeout	Pass
	test_notifications.py::TestCreateNotification	test_invalid_position_defaults_to_top_right	Pass
	test_notifications.py::TestCreateNotification	test_valid_position	Pass
	test_notifications.py::TestGetStatistics	test_returns_expected_keys	Pass
	test_notifications.py::TestGetStatistics	test_statistics_reflect_config	Pass
	test_startup.py::TestGetPreloaderJs	test_get_preloader_js_matches_get_js	Pass
	test_startup.py::TestGetPreloaderJs	test_get_preloader_js_returns_string	Pass
	test_startup.py::TestStartupPreloaderAddToMap	test_add_to_map_calls_add_child	Pass
	test_startup.py::TestStartupPreloaderAddToMap	test_add_to_map_output_contains_script	Pass
	test_td_main_map_coverage.py::TestMainMapFS01GetStatic	test_add_to_map_calls_add_child	Pass
	test_td_main_map_coverage.py::TestMainMapFS01GetStatic	test_get_js_resources_returns_script	Pass
	test_td_main_map_coverage.py::TestMainMapFS01GetStatic	test_get_statistics_returns_dict	Pass
	property_panel.py::TestPropertyHazardConfigure	test_configure_height	Pass
	property_panel.py::TestPropertyHazardConfigure	test_configure_does_not_change	Pass
	property_panel.py::TestPropertyHazardConfigure	test_configure_panel_height	Pass
	property_panel.py::TestPropertyHazardConfigure	test_configure_panel_width	Pass
	property_panel.py::TestPropertyHazardCustomConfigure	test_customconfigure	Pass
	property_panel.py::TestPropertyHazardCustomConfigure	test_customconfigure_returns_dict	Pass
	test_phc_basis_explorer.py::TestBasisExplorerPanelConfig	test_explorer_panel_dimensions	Pass
	test_phc_basis_explorer.py::TestBasisExplorerPanelConfig	test_default_panel_dimensions	Pass
	test_phc_basis_explorer.py::TestBasisExplorerPanelConfig	test_explorer_panel_has_top_level_tabs	Pass
	test_phc_basis_explorer.py::TestBasisExplorerSubModule	test_explorer_sub_module_available	Pass
	test_phc_basis_explorer.py::TestBasisExplorerSubModule	test_explorer_sub_module_is_a_function	Pass
	test_phc_basis_explorer.py::TestBasisExplorerSubModule	test_explorer_sub_module_imports	Pass
	test_phc_basis_explorer.py::TestBasisExplorerSubModule	test_explorer_sub_module_injected	Pass
	test_phc_basis_explorer.py::TestBasisExplorerSubModule	test_explorer_sub_module_injected	Pass
	test_phc_basis_explorer.py::TestBasisExplorerSubModule	test_explorer_sub_module_injected	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_active_tab_state	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_selected_storm_state	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_wiring_on_sub_tab_switch	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_wiring_column_for_basis	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_wiring_reset_for_non_basis	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_wiring_up_basis_state	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_wiring_explorer_function	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_sub_tab_function	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_wrapped	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_removed_on_tab_switch	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_stylingl	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_dispatches_gauge	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_dispatches_property	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_dispatches_shd	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_dispatches_she	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_names_array	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_names_defined	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_calls_basis_explorer	Pass
	test_phc_basis_explorer.py::TestBasisExplorerTabWiring	test_explorer_tab_renamed	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_chart_destroyed_before_render	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_chart_variable_declared	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_click_triggers_rerender	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_colours_by_severity	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_reads_gauge_thresholds	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_reads_nearest_gauges	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_reads_storm_details	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_render_function_defined	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_scatter_chart_type	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_severe_count_in_summary	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_sorts_by_peak_level	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_stats_bar_updated	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_storm_click_sets_selected	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_threshold_lines_for_alert	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_threshold_lines_for_severe	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_threshold_lines_for_warning	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_tooltip_shows_storm_id	Pass
	test_phc_basis_gauge.py::TestBasisGaugeJS	test_updates_exceeded_severe_flag	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_basis_computed	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_chart_variable_declared	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_click_sets_selected_storm	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_colours_by_damage_ratio	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_gauge_spread_in_stats	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_property_spread_in_stats	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_quadrant_basis_risk	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_quadrant_hedged	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_quadrant_no_event	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_quadrant_unhedged	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_reads_spread_decomposition	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_reads_storm_details	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_render_function_defined	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_scatter_chart_type	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_phc_basis_property.py::TestBasisPropertyJS	test_property_vertical_annotation	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_summary_cards_rendered	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_tooltip_shows_quadrant_info	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_uses_exceeded_severe_flag	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_x_axis_gauge_peak	Pass
	test_phc_basis_property.py::TestBasisPropertyJS	test_y_axis_property_depth	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_axes_drawn	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_chart_variable_declared	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_click_sets_selected_storm	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_colour_gradient_for_depth	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_distance_decay_curve_drawn	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_distance_decay_function_defined	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_gauge_marker_at_origin	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_property_marker	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_reads_distance_km	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_reads_retention_factor	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_reads_storm_details	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_render_function_defined	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_scatter_chart_type	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_selected_storm_highlight	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_shd_spread_in_stats	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_sorts_by_flood_depth	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_summary_shows_avg_retention	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_summary_shows_with_depth_count	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_tooltip_shows_flood_depth	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_tooltip_shows_retention	Pass
	test_phc_basis_shd.py::TestBasisSHDJS	test_two_canvas_layout	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_bankfull_offset	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_chart_variable_declared	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_click_sets_selected_storm	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_colours_green_for_reaches	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_colours_grey_for_no_impact	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_colours_orange_for_severe_but_misses	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_cross_section_draws_ground	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_cross_section_draws_water	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_cross_section_severe_line	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_elevation_section_function_defined	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_flood_threshold_calculated	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_floor_level_drawn	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_height_diff_calculated	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_property_threshold_annotation	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_reaches_property_classification	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_reads_floor_level	Pass
	test_phc_basis_she.py::TestBasisSHEUSJS	test_reads_gauge_elevation	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_phc_basis_she.py::TestBasisSHEUS	test_reads_property_elevation	Pass
	test_phc_basis_she.py::TestBasisSHEUS	test_reads_storm_details	Pass
	test_phc_basis_she.py::TestBasisSHEUS	test_render_function_defined	Pass
	test_phc_basis_she.py::TestBasisSHEUS	test_scatter_chart_type	Pass
	test_phc_basis_she.py::TestBasisSHEUS	test_she_spread_in_stats	Pass
	test_phc_basis_she.py::TestBasisSHEUS	test_summary_shows_counts	Pass
	test_phc_basis_she.py::TestBasisSHEUS	test_two_canvas_layout	Pass
	test_phc_prs_render_terrain.py::TestAdjustedZone	test_adjusted_spread_in_stats_bar	Pass
	test_phc_prs_render_terrain.py::TestAdjustedZone	test_adjusted_zone_conditional	Pass
	test_phc_prs_render_terrain.py::TestAdjustedZone	test_adjusted_zone_present	Pass
	test_phc_prs_render_terrain.py::TestAdjustedZone	test_adjusted_zone_shows_actual	Pass
	test_phc_prs_render_terrain.py::TestTerrainEffect	test_adjusted_prop_spread_in_total	Pass
	test_phc_prs_render_terrain.py::TestTerrainEffect	test_selected_zone_shown_in_detail	Pass
	test_phc_prs_render_terrain.py::TestTerrainEffect	test_terrain_effect_colour_green_negative	Pass
	test_phc_prs_render_terrain.py::TestTerrainEffect	test_terrain_effect_colour_orange_positive	Pass
	test_phc_prs_render_terrain.py::TestTerrainEffect	test_terrain_effect_delta_variable	Pass
	test_phc_prs_render_terrain.py::TestTerrainEffect	test_terrain_effect_effect_label	Pass
	test_phc_prs_render_terrain.py::TestTerrainEffect	test_terrain_effect_row_background	Pass
	test_phc_prs_render_terrain.py::TestTerrainEffect	test_terrain_effect_row_conditional	Pass
	test_phc_prs_render_terrain.py::TestTerrainEffect	test_terrain_effect_row_in_path1	Pass
	test_phc_prs_render_terrain.py::TestTerrainEffect	test_terrain_effect_row_in_path2	Pass
	test_phc_prs_terrain_pricer.py::TestTerrainPricerCrossValidation	test_delta_interpolation_cross_validating_zone	Pass
	test_phc_prs_terrain_pricer.py::TestTerrainPricerCrossValidation	test_delta_interpolation_cross_validating_zone	Pass
	test_phc_prs_terrain_pricer.py::TestTerrainPricerCrossValidation	test_delta_zero_same_zone	Pass
	test_phc_prs_terrain_pricer.py::TestTerrainPricerCrossValidation	test_python_interpolation_cross_validation	Pass
	test_phc_prs_terrain_pricer.py::TestTerrainPricerJS	test_terrain_zone_from_phcdata	Pass
	test_phc_prs_terrain_pricer.py::TestTerrainPricerJS	test_avg_distance_computed_from_gauges	Pass
	test_phc_prs_terrain_pricer.py::TestTerrainPricerJS	test_bilinear_interpolation_logic	Pass
	test_phc_prs_terrain_pricer.py::TestTerrainPricerJS	test_delta_zero_when_same_zone	Pass
	test_phc_prs_terrain_pricer.py::TestTerrainPricerJS	test_elevation_diff_computed	Pass
	test_phc_prs_terrain_pricer.pyTestTerrainPricerJS	test_grid_clamping	Pass
	test_phc_prs_terrain_pricer.pyTestTerrainPricerJS	test_interpolate_accepts_terrain_grid	Pass
	test_phc_prs_terrain_pricer.pyTestTerrainPricerJS	test_interpolate_function_present	Pass
	test_phc_prs_terrain_pricer.pyTestTerrainPricerJS	test_terrain_includes_terrain_fields	Pass
	test_phc_prs_terrain_pricer.pyTestTerrainPricerJS	test_selected_zone_read_from_dropdown	Pass
	test_phc_prs_terrain_pricer.pyTestTerrainPricerJS	test_terrain_delta_computed	Pass
	test_phc_prs_zone_dropdown.pyTestZoneDropdownJS	test_zone_options_present	Pass
	test_phc_prs_zone_dropdown.pyTestZoneDropdownJS	test_change_listener_includes_zone	Pass
	test_phc_prs_zone_dropdown.pyTestZoneDropdownJS	test_default_from_phcdata	Pass
	test_phc_prs_zone_dropdown.pyTestZoneDropdownJS	test_label_present	Pass
	test_phc_prs_zone_dropdown.pyTestZoneDropdownJS	test_select_element_present	Pass
	test_phc_prs_zone_dropdown.pyTestZoneDropdownJS	test_zone_order	Pass
	sp_sim_detail.pyTestScrubberRange	test_hour_label_present	Pass
	sp_sim_detail.pyTestScrubberRange	test_scrubber_id_present	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	sp_sim_detail.py::TestScrubberRange	test_scrubber_max_from_config	Pass
	sp_sim_detail.py::TestScrubberRange	test_scrubber_max_not_59	Pass
	sp_sim_detail.py::TestScrubberRange	test_scrubber_min_is_zero	Pass
	sp_sim_detail.py::TestScrubberRange	test_scrubber_type_is_range	Pass
	sp_sim_detail.py::TestSimFunctions	test_get_js_returns_nonempty	Pass
	sp_sim_detail.py::TestSimFunctions	test_spSeekTo_defined	Pass
	sp_sim_detail.py::TestSimFunctions	test_spSetSpeed_defined	Pass
	sp_sim_detail.py::TestSimFunctions	test_spStopAnim_defined	Pass
	sp_table_detail.py::TestPanelTitle	test_panel_title_id_present	Pass
	sp_table_detail.py::TestPanelTitle	test_panel_title_prefix	Pass
	sp_table_detail.py::TestPanelTitle	test_title_reads_selected_option_text	Pass
	sp_table_detail.py::TestPanelTitle	test_title_uses_label_variable_not_bare_storm_id_concat	Pass
	sp_table_detail.py::TestStormList	test_loads_from_propertyts_storms	Pass
	sp_table_detail.py::TestStormList	test_logging_storms_message	Pass
	sp_table_detail.py::TestStormList	test_uses_preloaded_cache	Pass
	sp_table_detail.py::TestTableStructure	test_preload_portfolio_impact_defined	Pass
	sp_table_detail.py::TestTableStructure	test_portfolio_impact_endpoint	Pass
	sp_table_detail.py::TestTableStructure	test_sort_columns_defined	Pass
	sp_table_detail.py::TestTableStructure	test_summary_cards_present	Pass
	test_cross_iife_exports.py::TestCrossPanelExports	test_bfCrossPanelExports_refresh_main_map	Pass
	test_cross_iife_exports.py::TestCrossPanelExports	test_bfCrossPanelExports_apply_filter	Pass
	test_cross_iife_exports.py::TestCrossPanelExports	test_f01CrossPanelExports_cell_click	Pass
	test_cross_iife_exports.py::TestCrossPanelExports	test_pendingFilterCrossPanelExports_window	Pass
	test_cross_iife_exports.py::TestCrossPanelExports	test_refreshMainMapCrossPanelExports_f01_exported	Pass
	test_cross_iife_exports.py::TestCrossPanelExports	test_refreshMainMapCrossPanelExports_window	Pass
	test_cross_iife_exports.py::TestCrossPanelExports	test_setupCrossPanelExports_window_hint	Pass
	test_cross_iife_exports.py::TestCrossPanelExports	test_toggleDeskCrossPanelExports_show_hide	Pass
	test_cross_iife_exports.py::TestCrossPanelOpen	test_createPanelOpenCreates_blotter_view	Pass
	test_cross_iife_exports.py::TestCrossPanelOpen	test_openPanelOpenFunction_exists	Pass
	test_cross_iife_exports.py::TestCrossPanelOpen	test_openPanelOpenSetsDisplay_flex	Pass
	test_cross_iife_exports.py::TestCrossPanelOpen	test_pickPanelOpenCallsShowPanel	Pass
	test_cross_iife_exports.py::TestCrossPanelOpen	test_runPreloadOpenFunction_exists	Pass
	test_cross_iife_exports.py::TestCrossPanelOpen	test_showPanelOpenFunction_exists	Pass
	test_cross_iife_exports.py::TestCrossPanelOpen	test_showPanelOpenReadsWindow_preload_done	Pass
	test_cross_iife_exports.py::TestCrossPanelOpen	test_switchPanelOpenSteps_open	Pass
	test_cross_iife_ownership.py::TestCentralized	test_cache_starts_initialized_on_window[_preGovAudit]	Pass
	test_cross_iife_ownership.py::TestCentralized	test_cache_starts_initialized_on_window[_preGovBib]	Pass
	test_cross_iife_ownership.py::TestCentralized	test_cache_starts_initialized_on_window[_preStorms]	Pass
	test_cross_iife_ownership.py::TestCentralized	test_cache_starts_initialized_on_window[_tdPreBlotter]	Pass
	test_cross_iife_ownership.py::TestCentralized	test_cache_starts_initialized_on_window[_tdPreEodHistory]	Pass
	test_cross_iife_ownership.py::TestCentralized	test_cache_starts_initialized_on_window[_tdPreGauges]	Pass
	test_cross_iife_ownership.py::TestCentralized	test_cache_starts_initialized_on_window[_tdPreGovDocs]	Pass
	test_cross_iife_ownership.py::TestCentralized	test_cache_starts_initialized_on_window[_tdPreMarket]	Pass
	test_cross_iife_ownership.py::TestCentralized	test_cache_starts_initialized_on_window[_tdPrePortStorms]	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_cache_starting_initialised_on_window[_td_PreStressStorms]	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_cache_starting_initialised_on_window[_td_PreYieldCurve]	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_no_bare_var_declarations_for_cache_vars	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_no_var_declarations_for_td_preload_done	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_window_td_preload_done_initialised	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_window_td_preload_done_set_after_setup	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_blotter_cache_consume_after_consume	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_blotter_cache_reads_window_pre_blotter	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_gov_docs_cache_consume_after_consume	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_gov_docs_cache_reads_window_pre_gov_docs	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_port_stress_cache_consume_after_consume	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_port_stress_cache_reads_window_pre_port_storms	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_preloader_does_not_use_bare_td_preload_done_assignment	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_preloader_sets_window_preload_done	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_storm_portfolio_cache_consume_after_consume	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_storm_portfolio_cache_reads_window_pre_storms	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_trading_desk_cache_reads_window_preload_done	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_blotter_NoBareReads_bare_pre_blotter	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_gov_docs_NoBareReads_bare_pre_gov_docs	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_port_stress_NoBareReads_bare_pre_port_storms	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_preloader_NoBareReads_bare_preload_done_read	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_storm_NoBareReads_bare_pre_storms	Pass
	test_cross_iife_ownership.py::TestCrossIifeOwnership	test_trading_desk_NoBareReads_bare_preload_done	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_governance_submodules_syntax	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_preloader_syntax	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_portfolio_syntax	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_trading_aggregate_submodule_syntax	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_trading_blotter_submodule_syntax	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_trading_curves_submodule_syntax	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_trading_desk_js_syntax	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_trading_ecod_submodule_syntax	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_trading_fs01_submodule_syntax	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_trading_market_submodule_syntax	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_trading_port_stress_submodule_syntax	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_trading_stress_submodule_syntax	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_aggregate_generates	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_blotter_generates	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_curves_generates	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_ecod_generates	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_fs01_generates	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_governance_bibliography_generates	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_governance_data_generates	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_market_generates	Pass
	test_cross_iife_syntax.py::TestCompileJS	test_port_stress_generates	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_cross_iife_syntax.py::TestCo	test_preloader_generates	Pass
	test_cross_iife_syntax.py::TestCo	test_startup_generates	Pass
	test_cross_iife_syntax.py::TestCo	test_transformer_generates	Pass
	test_cross_iife_syntax.py::TestCo	test_trees_generates	Pass
	test_cross_iife_syntax.py::TestCo	test_trading_desk_generates	Pass
	test_startup_preloader_part1.py::Te	test_startup_preloader_defined	Pass
	test_startup_preloader_part1.py::Te	test_all_four_endpoints_present	Pass
	test_startup_preloader_part1.py::Te	test_first_preloader_block_loaded	Pass
	test_startup_preloader_part1.py::Te	test_market_state_behaviour	Pass
	test_startup_preloader_part1.py::Te	test_hazard_curves_behaviour	Pass
	test_startup_preloader_part1.py::Te	test_status_preload_done_behaviour	Pass
	test_startup_preloader_part1.py::Te	test_show_progress_popup_behaviour	Pass
	test_startup_preloader_part1.py::Te	test_startup_preloader_class_defined	Pass
	test_startup_preloader_part1.py::Te	test_declares_pre_g6_gauge_vars	Pass
	test_startup_preloader_part1.py::Te	test_declares_pre_g6_counter_vars	Pass
	test_startup_preloader_part1.py::Te	test_declares_pre_g6_hist_vars	Pass
	test_startup_preloader_part1.py::Te	test_declares_pre_g6_property_vars	Pass
	test_startup_preloader_part1.py::Te	test_declares_pre_g6_cache_vars	Pass
	test_startup_preloader_part1.py::Te	test_declares_pre_g6_cache_counter_vars	Pass
	test_startup_preloader_part1.py::Te	test_declares_pre_g6_cache_history_vars	Pass
	test_startup_preloader_part1.py::Te	test_declares_pre_g6_gauge_vars	Pass
	test_startup_preloader_part1.py::Te	test_declares_pre_g6_counter_vars	Pass
	test_startup_preloader_part1.py::Te	test_declares_pre_g6_market_vars	Pass
	test_startup_preloader_part1.py::Te	test_declares_pre_g6_pnl_vars_storms	Pass
	test_startup_preloader_part1.py::Te	test_declares_pre_g6_stress_vars_storms	Pass
	test_startup_preloader_part1.py::Te	test_declares_pre_g6_yield_curve_vars	Pass
	test_startup_preloader_part1.py::Te	test_TradingPreloader_does_not_redeclare_td_pre_blotter	Pass
	test_startup_preloader_part1.py::Te	test_TradingPreloader_does_not_redeclare_td_pre_market	Pass
	test_startup_preloader_part1.py::Te	test_TradingPreloader_does_not_redeclare_td_preload_done	Pass
	test_startup_preloader_part2.py::Te	test_applyDocumentsFromCache_exists	Pass
	test_startup_preloader_part2.py::Te	test_checkAndGetDocumentsFromCache	Pass
	test_startup_preloader_part2.py::Te	test_clearAndCacheAfterUse	Pass
	test_startup_preloader_part2.py::Te	test_HealthRoute_200_at_api_v1	Pass
	test_startup_preloader_part2.py::Te	test_HealthRoute_200_at_root	Pass
	test_startup_preloader_part2.py::Te	test_HealthRoute_check_function_serves_both	Pass
	test_startup_preloader_part2.py::Te	test_HealthRoute_has_api_v1_health_route	Pass
	test_startup_preloader_part2.py::Te	test_HealthRoute_has_plain_health_route	Pass
	test_startup_preloader_part2.py::Te	test_HealthRoute_returns_ok_status	Pass
	test_startup_preloader_part2.py::Te	test_initializeAllManagers_before_notifications	Pass
	test_startup_preloader_part2.py::Te	test_startupPreloaderWindow_imported_in_init	Pass
	test_startup_preloader_part2.py::Te	test_startupPreloaderWindow_instantiated_in_init	Pass
	test_startup_preloader_part2.py::Te	test_all_DetailStates_keys_have_stat_handler	Pass
	test_startup_preloader_part2.py::Te	test_StartupDetailStat_uses_trades_length	Pass
	test_startup_preloader_part2.py::Te	test_StartupDetailStat_uses_history_length	Pass
	test_startup_preloader_part2.py::Te	test_GaugeDetailStat_uses_count	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	test_startup_preloader_part2.py::TestGovernorDetailStatus	test_gauges_uses_gauges_length	Pass
	test_startup_preloader_part2.py::TestGovernorDetailStatus	test_gov_uses_events_length	Pass
	test_startup_preloader_part2.py::TestGovernorDetailStatus	test_gov_uses_references_length	Pass
	test_startup_preloader_part2.py::TestGovernorDetailStatus	test_gov_uses_documents_length	Pass
	test_startup_preloader_part2.py::TestMarketDetailStatus	test_market_uses_object_keys_gauges	Pass
	test_startup_preloader_part2.py::TestMortgagesDetailStatus	test_mortgages_uses_mortgages_length	Pass
	test_startup_preloader_part2.py::TestPortfolioDetailStatus	test_portfolio_uses_storms_length	Pass
	test_startup_preloader_part2.py::TestPropertyDetailStatus	test_property_uses_storms_length	Pass
	test_startup_preloader_part2.py::TestPropertyDetailStatus	test_property_uses_total_properties	Pass
	test_startup_preloader_part2.py::TestStormsDetailStatus	test_storms_uses_storms_length	Pass
	test_startup_preloader_part2.py::TestYieldCurveDetailStatus	test_yield_curve_uses_object_keys	Pass
	test_startup_preloader_part2.py::TestApplyPortfolioCache	test_apply_portfolio_cache_function_exists	Pass
	test_startup_preloader_part2.py::TestCheckPortfolioCache	test_check_portfolio_cache	Pass
	test_startup_preloader_part2.py::TestClearPortfolioCache	test_clear_portfolio_cache_after_use	Pass
	blotter_js.py::TestBlotterSignedNotional	test_met_notional_in_pnl_bar	Pass
	blotter_js.py::TestBlotterSignedNotional	test_signed_notional_payer_negative	Pass
	blotter_js.py::TestMarketPnLDisplay	test_blotter_has_from_market_in_pnl_bar	Pass
	blotter_js.py::TestMarketPnLDisplay	test_blotter_has_market_pnl_column	Pass
	blotter_js.py::TestMarketPnLDisplay	test_commit_shows_pnl_impact	Pass
	blotter_js.py::TestMarketPnLDisplay	test_hazard_column_up_down_arrows	Pass
	classifiers_js.py::TestClassifiersEndpoints	test_status_endpoint	Pass
	classifiers_js.py::TestClassifiersEndpoints	test_summary_endpoint	Pass
	classifiers_js.py::TestClassifiersEndpoints	test_train_all_endpoint	Pass
	classifiers_js.py::TestClassifiersEndpoints	test_train_endpoint	Pass
	classifiers_js.py::TestClassifiersInPanel	test_classifiers_cleanup	Pass
	classifiers_js.py::TestClassifiersInPanel	test_classifiers_in_views_array	Pass
	classifiers_js.py::TestClassifiersInPanel	test_classifiers_switch_handler	Pass
	classifiers_js.py::TestClassifiersInPanel	test_classifiers_tab_in_array	Pass
	classifiers_js.py::TestClassifiersInPanel	test_classifiers_view_created	Pass
	classifiers_js.py::TestClassifiersStudioHas	test_classifiers_renders	Pass
	classifiers_js.py::TestClassifiersStudioHas	test_has_batch_training	Pass
	classifiers_js.py::TestClassifiersStudioHas	test_has_cleanup	Pass
	classifiers_js.py::TestClassifiersStudioHas	test_has_create_view	Pass
	classifiers_js.py::TestClassifiersStudioHas	test_has_feature_chart	Pass
	classifiers_js.py::TestClassifiersStudioHas	test_has_load_data	Pass
	classifiers_js.py::TestClassifiersStudioHas	test_has_metric_card	Pass
	classifiers_js.py::TestClassifiersStudioHas	test_has_progress_bar	Pass
	classifiers_js.py::TestClassifiersStudioHas	test_has_render_detail	Pass
	classifiers_js.py::TestClassifiersStudioHas	test_has_render_summary_table	Pass
	classifiers_js.py::TestClassifiersStudioHas	test_has_single_training	Pass
	classifiers_js.py::TestClassifiersStudioHas	test_has_state_vars	Pass
	market_js.py::TestCommitButtonPresent	test_commit_button_calls_commit_function	Pass
	market_js.py::TestCommitButtonPresent	test_commit_button_in_html	Pass
	market_js.py::TestCommitButtonPresent	test_commit_button_not_hidden	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	market_js.py::TestCommitButtonJS	test_commit_function_exists	Pass
	market_js.py::TestCommitButtonJS	test_commit_posts_to_endpoint	Pass
	market_js.py::TestCommitButtonJS	test_commit_refreshes_blotter	Pass
	market_js.py::TestCurvesTabRedesign	test_curves_all_gauges_chart	Pass
	market_js.py::TestCurvesTabRedesign	test_curves_has_trigger_selector	Pass
	market_js.py::TestCurvesTabRedesign	test_curves_hsl_colours	Pass
	market_js.py::TestCurvesTabRedesign	test_curves_renders	Pass
	market_js.py::TestCurvesTabRedesign	test_curves_strips_thames_prefix	Pass
	market_js.py::TestCurvesTabRedesign	test_curves_uses_market_state	Pass
	market_js.py::TestMapClickThrough	test_map_hides_trading_desk	Pass
	market_js.py::TestMapClickThrough	test_map_new_trade_link	Pass
	market_js.py::TestMapClickThrough	test_map_view_hazard_curve	Pass
	panel.py::TestInteractivityManager	test_manager_exports_trading_desk	Pass
	panel.py::TestInteractivityManager	test_manager_has_trading_desk	Pass
	panel.py::TestPreloadDoneScoping	test_preloader_sets_window_preload_done	Pass
	panel.py::TestPreloadDoneScoping	test_startup_uses_window_preload_done_init	Pass
	panel.py::TestPreloadDoneScoping	test_startup_uses_window_preload_done_set	Pass
	panel.py::TestPreloadDoneScoping	test_tradingdesk_reads_window_preload_done	Pass
	panel.py::TestTradingDeskPanelJS	test_add_to_map	Pass
	panel.py::TestTradingDeskPanelJS	test_blotter_submodule	Pass
	panel.py::TestTradingDeskPanelJS	test_eod_submodule	Pass
	panel.py::TestTradingDeskPanelJS	test_map_submodule	Pass
	panel.py::TestTradingDeskPanelJS	test_market_submodule	Pass
	panel.py::TestTradingDeskPanelJS	test_no_unescaped_braces	Pass
	panel.py::TestTradingDeskPanelJS	test_panel_contains_all_tabs	Pass
	panel.py::TestTradingDeskPanelJS	test_panel_contains_global_export	Pass
	panel.py::TestTradingDeskPanelJS	test_panel_contains_pi_symbol	Pass
	panel.py::TestTradingDeskPanelJS	test_panel_renders	Pass
	panel.py::TestTradingDeskPanelJS	test_risk_submodule	Pass
	port_stress_js_part1.py::TestPortfolioJS	test_portfolio_run_endpoint	Pass
	port_stress_js_part1.py::TestPortfolioJS	test_portfolio_storms_endpoint	Pass
	port_stress_js_part1.py::TestPortfolioJS	test_post_method_used_for_run	Pass
	port_stress_js_part1.py::TestPortfolioJS	test_id_conflicts_with_stress_tab	Pass
	port_stress_js_part1.py::TestPortfolioJS	test_id_prefixes_content	Pass
	port_stress_js_part1.py::TestPortfolioJS	test_id_prefixes_storm_sel	Pass
	port_stress_js_part1.py::TestPortfolioJS	test_id_prefixes_tabs	Pass
	port_stress_js_part1.py::TestPortfolioJS	test_flood_tab_canvas_id	Pass
	port_stress_js_part1.py::TestPortfolioJS	test_flood_tab_render_function	Pass
	port_stress_js_part1.py::TestPortfolioJS	test_flood_tab_threshold_colours	Pass
	port_stress_js_part1.py::TestPortfolioJS	test_flood_tab_uses_chartjs	Pass
	port_stress_js_part1.py::TestPortfolioJS	test_portfolio_tab_canvas_id	Pass
	port_stress_js_part1.py::TestPortfolioJS	test_portfolio_tab_fmt_gbp	Pass
	port_stress_js_part1.py::TestPortfolioJS	test_portfolio_tab_render_function	Pass
	port_stress_js_part1.py::TestPortfolioJS	test_portfolio_tab_uses_horizontal_bars	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	port_stress_js_part1.py::TestPortStressJs	test_port_stress_has_all_sub_tabs	Pass
	port_stress_js_part1.py::TestPortStressJs	test_port_stress_has_cleanup	Pass
	port_stress_js_part1.py::TestPortStressJs	test_port_stress_has_create_view	Pass
	port_stress_js_part1.py::TestPortStressJs	test_port_stress_has_load_data	Pass
	port_stress_js_part1.py::TestPortStressJs	test_port_stress_has_storm_changed	Pass
	port_stress_js_part1.py::TestPortStressJs	test_port_stress_has_switch_subtab	Pass
	port_stress_js_part1.py::TestPortStressJs	test_port_stress_renders	Pass
	port_stress_js_part1.py::TestPortStressJs	test_port_stress_state_vars	Pass
	port_stress_js_part2.py::TestPortStressJs	test_gauge_pnl_tabs_switch_tab_stress	Pass
	port_stress_js_part2.py::TestPortStressJs	test_gauge_pnl_full_detail_button	Pass
	port_stress_js_part2.py::TestPortStressJs	test_gauge_pnl_has_gauge_table	Pass
	port_stress_js_part2.py::TestPortStressJs	test_gauge_pnl_render_function	Pass
	port_stress_js_part2.py::TestPortStressJs	test_gauge_pnl_tabs_gauge_hint	Pass
	port_stress_js_part2.py::TestPortStressJs	test_gauge_pnl_tabs_storm_hint	Pass
	port_stress_js_part2.py::TestPortStressJs	test_full_trading_desk_valid_with_port_stress	Pass
	port_stress_js_part2.py::TestPortStressJs	test_load_port_stress_data_called	Pass
	port_stress_js_part2.py::TestPortStressJs	test_no_trade_desk	Pass
	port_stress_js_part2.py::TestPortStressJs	test_port_stress_tab_in_full_js	Pass
	port_stress_js_part2.py::TestPortStressJs	test_port_stress_view_appended	Pass
	port_stress_js_part2.py::TestPortStressJs	test_port_stress_view_created	Pass
	port_stress_js_part2.py::TestPortStressJs	test_port_stress_in_switch_tab	Pass
	port_stress_js_part2.py::TestPortStressJs	test_port_stress_in_tabs_array	Pass
	port_stress_js_part2.py::TestPortStressJs	test_port_stress_in_views_array	Pass
	port_stress_js_part2.py::TestPortStressJs	test_port_stress_in_hide_panel	Pass
	port_stress_js_part2.py::TestPortStressJs	test_port_stress_iife	Pass
	port_stress_js_part2.py::TestPortStressJs	test_severity_tab_gauge_pnl_link	Pass
	port_stress_js_part2.py::TestPortStressJs	test_severity_tab_has_four_groups	Pass
	port_stress_js_part2.py::TestPortStressJs	test_severity_tab_render_function	Pass
	port_stress_js_part2.py::TestPortStressJs	test_severity_tab_sets_selected_gauge	Pass
	port_stress_js_part2.py::TestPortStressJs	test_severity_tab_severe_colour	Pass
	stress_js_part1.py::TestStressGaugeFunctions	test_functions_contains_populate_gauge_dropdown	Pass
	stress_js_part1.py::TestStressGaugeFunctions	test_functions_contains_stress_gauge_changed	Pass
	stress_js_part1.py::TestStressGaugeFunctions	test_functions_no_unclosed_template_literal	Pass
	stress_js_part1.py::TestStressGaugeFunctions	test_functions_selector_element_id_present	Pass
	stress_js_part1.py::TestStressGaugeFunctions	test_populate_gauge_dropdown_function_defined	Pass
	stress_js_part1.py::TestStressGaugeFunctions	test_populate_storm_dropdown_function_defined	Pass
	stress_js_part1.py::TestStressGaugeFunctions	test_storm_selector_element_id_present	Pass
	stress_js_part1.py::TestStressGaugeFunctions	test_functions_functions_not_before_iife_close	Pass
	stress_js_part1.py::TestStressGaugeFunctions	test_functions_gauge_changed_binds_to_dropdown	Pass
	stress_js_part1.py::TestStressGaugeFunctions	test_functions_gauge_changed_function_defined	Pass
	stress_js_part1.py::TestStressGaugeFunctions	test_functions_storm_changed_binds_to_dropdown	Pass
	stress_js_part1.py::TestStressGaugeFunctions	test_functions_storm_changed_function_defined	Pass
	stress_js_part1.py::TestStressKnockOut	test_stress_ko_annotation_on_charts	Pass
	stress_js_part1.py::TestStressKnockOut	test_stress_ko_status_in_table	Pass

(continued on next page)

(continued from previous page)

Model	Module	Test Name	Status
	stress_js_part1.py::TestStressKnockout	test_stress_stats_bar_ko_info	Pass
	stress_js_part2.py::TestStressGaugeHint	test_fallback_only_when_hint_not_matched	Pass
	stress_js_part2.py::TestStressGaugeHint	test_hint_applied_before_fallback	Pass
	stress_js_part2.py::TestStressGaugeHint	test_hint_consumed_after_use	Pass
	stress_js_part2.py::TestStressGaugeHint	test_hint_fallback_to_blotter_filter	Pass
	stress_js_part2.py::TestStressGaugeHint	test_load_stress_data_accepts_gauge_hint	Pass
	stress_js_part2.py::TestStressGaugeHint	test_load_stress_data_stores_hint	Pass
	stress_js_part2.py::TestStressGaugeHint	test_populate_gauge_dropdown_reads_hint	Pass
	stress_js_part2.py::TestStressGaugeHint	test_stress_gauge_hint_state_variable	Pass
	stress_js_part2.py::TestStressGaugeHint	test_switch_tab_stress_blotter_filter_first	Pass
	stress_js_part2.py::TestStressGaugeHint	test_switch_tab_stress_passes_gauge_hint	Pass
	stress_js_part2.py::TestStressGaugeHint	test_switch_tab_stress_uses_blotter_data_fallback	Pass
	stress_js_part2.py::TestStressTabCascades	test_stress_chart_canvas	Pass
	stress_js_part2.py::TestStressTabCascades	test_stress_chart_tabs	Pass
	stress_js_part2.py::TestStressTabCascades	test_stress_charts	Pass
	stress_js_part2.py::TestStressTabCascades	test_stress_cleanup	Pass
	stress_js_part2.py::TestStressTabCascades	test_stress_gauge_dropdown	Pass
	stress_js_part2.py::TestStressTabCascades	test_stress_gauges_endpoint	Pass
	stress_js_part2.py::TestStressTabCascades	test_stress_has_create_and_load	Pass
	stress_js_part2.py::TestStressTabCascades	test_stress_ko_annotation	Pass
	stress_js_part2.py::TestStressTabCascades	test_stress_link_property_damage	Pass
	stress_js_part2.py::TestStressTabCascades	test_stress_link_storm_simulation	Pass
	stress_js_part2.py::TestStressTabCascades	test_stress_renders	Pass
	stress_js_part2.py::TestStressTabCascades	test_stress_stats_bar	Pass
	stress_js_part2.py::TestStressTabCascades	test_stress_storm_dropdown	Pass
	stress_js_part2.py::TestStressTabCascades	test_stress_surface_table	Pass
	stress_js_part2.py::TestStressTabCascades	test_stress_trade_table	Pass
	stress_js_part2.py::TestStressTabCascades	test_surface_table_scrollable	Pass

3 Flood Risk (Depth-Damage) (MKM-DD-001)

Table 4: Test Results Summary — Flood Risk (Depth-Damage) (MKM-DD-001)

Total Tests	32
Passed	32
Failed	0
Skipped	0
Duration	0.00s

Table 5: Detailed Test Results — Flood Risk (Depth-Damage)

Test Description	Acceptance Criteria	Result	Time
Every hour from 0..n must be present in output, in order.	`hours == list(range(len(gauge_readings)))`	Pass	0.000s
Flood depth must be non-negative at every time step.	`r['depth_m'] >= 0.0`	Pass	0.000s
Each hydrograph entry must contain hour, wse_m, depth_m, and flooded.	`'hour' in r`; `'wse_m' in r` (+2 more)	Pass	0.000s
Empty gauge readings must return empty property hydrograph.	`build_property_hydrograph([], 8.0, 1.0, 0.9, 6.0, 0.3) == []`	Pass	0.000s
When floor level exceeds peak WSE, no time step should be flooded.	`r['flooded'] is False`	Pass	0.000s
With travel time of 5 hours, first 5 hours must show zero flood depth.	`r['depth_m'] == 0.0`; `r['flooded'] is False`	Pass	0.000s
Property hydrograph must have same number of time steps as gauge input.	`len(result) == len(gauge_readings)`	Pass	0.000s
Low-elevation property at gauge location must flood at peak water level.	`max(peak_depths) > 0`	Pass	0.000s
Slope from 10m to 8m over 1000m should be 0.002 (2m/1000m).	`abs(s - 0.002) < 1e-6`	Pass	0.000s
Flat terrain (equal elevations) must return MIN_SLOPE, not zero.	`s == MIN_SLOPE`	Pass	0.000s
Slope must be the same regardless of flow direction (uses absolute value).	`s1 == s2`	Pass	0.000s
Zero horizontal distance must return MIN_SLOPE to avoid division by zero.	`t == 0.0`; `s == MIN_SLOPE`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Monotonicity: deeper water must produce higher velocity at constant slope.	`v2 > v1`	Pass	0.000s
Negative depth is physically invalid; must return zero velocity.	`compute_manning_velocity(-1.0, 0.005) == 0.0`	Pass	0.000s
Negative slope (downhill in opposite direction) must use absolute value.	`v_pos == v_neg`	Pass	0.000s
Manning velocity at d=1.0m, S=0.005, n=0.04 should be ˜1.77 m/s.	`1.5 < v < 2.0`	Pass	0.000s
Slopes below MIN_SLOPE must be clamped to MIN_SLOPE to avoid	`v == v_min`	Pass	0.000s
Monotonicity: steeper slope must produce higher velocity at constant depth.	`v2 > v1`	Pass	0.000s
Zero depth must produce zero velocity (no flow without water).	`compute_manning_velocity(0.0, 0.005) == 0.0`	Pass	0.000s
Zero roughness coefficient causes division by zero; must return 0.	`compute_manning_velocity(1.0, 0.005, roughness=0.0) == 0.0`	Pass	0.000s
At distance = characteristic length, factor must equal 1/e 0.368.	`abs(f - 1.0 / math.e) < 0.001`	Pass	0.000s
compute_attenuation is a backwards-compatible alias.	`compute_attenuation(800) == compute_retention(800)`	Pass	0.000s
At 5x the decay length, retention should be ¡ 1%.	`f < 0.01`	Pass	0.000s
At 800m with default 10km length scale, retention 0.923.	`0.90 <= f <= 0.95`	Pass	0.000s
Negative distance is clamped to zero; must return 1.0.	`compute_retention(-100) == 1.0`	Pass	0.000s
At 1km with default 10km length scale, retention 0.905.	`0.88 <= f <= 0.93`	Pass	0.000s
At zero distance, retention must be 1.0 (full signal).	`compute_retention(0) == 1.0`	Pass	0.000s
Zero decay length means instant decay; must return 0.0.	`compute_retention(100, 0) == 0.0`	Pass	0.000s
Travel time for 500m at d=1.0m, S=0.005 should be well under 1 hour.	`0 < t < 1.0`	Pass	0.000s
Monotonicity: greater distance must produce longer travel time.	`t2 > t1`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Zero depth produces zero velocity, so travel time must be infinite.	`t == float('inf')`	Pass	0.000s
Zero distance must produce zero travel time.	`t == 0.0; s == MIN_SLOPE`	Pass	0.000s

4 Flood Polynomial Model (MKM-FPO-001)

Table 6: Test Results Summary — Flood Polynomial Model (MKM-FPO-001)

Total Tests	33
Passed	33
Failed	0
Skipped	0
Duration	0.00s

Table 7: Detailed Test Results — Flood Polynomial Model

Test Description	Acceptance Criteria	Result	Time
Coefficients used by model match config/models.py.	`hasattr(flood_poly_coeffs, 'a')`; `hasattr(flood_poly_coeffs, 'f')` (+1 more)	Pass	0.000s
Fit metadata from config	`flood_poly_fit["model_id"] == "flood-poly-v1"`; `flood_poly_fit["r_squared"] > 0.9`	Pass	0.000s
Log eps from config	`flood_poly_log_eps == 1e-8`	Pass	0.000s
Storm hours from config	`flood_poly_storm_hours == 168`	Pass	0.000s
Coefficients count	`card["num_parameters"] == 6`	Pass	0.000s
Fit quality has metrics	`"r_squared" in fq`; `"mae" in fq` (+1 more)	Pass	0.000s
Has required fields	`"model_id" in card`; `"version" in card` (+3 more)	Pass	0.000s
Model id matches config	`card["model_id"] == flood_poly_fit["model_id"]`	Pass	0.000s
Returns dict	`isinstance(card, dict)`	Pass	0.000s
Water far above severe at late hour should be near 1.0.	`p > 0.9`	Pass	0.000s
Water well above severe threshold should give high P(flood).	`p > 0.8`	Pass	0.000s
hour=167 (last storm hour) should not raise.	`0.0 <= p <= 1.0`	Pass	0.000s
hour=0 should not raise.	`0.0 <= p <= 1.0`	Pass	0.000s
P(flood) at hour 0 should be lower than mid-storm for moderate water.	`p_early < p_mid`	Pass	0.000s
In range	`0.0 <= result <= 1.0`; `0.0 <= result <= 1.0`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Water well below severe threshold should give low P(flood).	`p < 0.1`	Pass	0.000s
Higher water level should give higher P(flood) at same hour.	`p_low < p_mid < p_high`	Pass	0.000s
Negative severe returns zero	`p_flood_poly(water_level=5.0, hour=48, severe_level=-1.0)...; p_flood_simple(5.0, -1.0) == 0.0`	Pass	0.000s
Returns float	`isinstance(result, float); isinstance(result, float)`	Pass	0.000s
Near-zero water level should give near-zero P(flood).	`p < 0.01`	Pass	0.000s
Zero severe returns zero	`p_flood_poly(water_level=5.0, hour=48, severe_level=0.0) ...; p_flood_simple(5.0, 0.0) == 0.0`	Pass	0.000s
All in range	`all(0.0 <= p <= 1.0 for p in result)`	Pass	0.000s
Correct length	`len(result) == 5`	Pass	0.000s
Empty list	`p_flood_series([], severe_level=6.58) == []`	Pass	0.000s
Returns list	`isinstance(result, list)`	Pass	0.000s
Above severe capped at one	`p_flood_simple(10.0, 6.58) == 1.0`	Pass	0.000s
At severe equals one	`p_flood_simple(6.58, 6.58) == 1.0`	Pass	0.000s
In range	`0.0 <= result <= 1.0;  0.0 <= result <= 1.0`	Pass	0.000s
Negative severe returns zero	`p_flood_poly(water_level=5.0, hour=48, severe_level=-1.0)...; p_flood_simple(5.0, -1.0) == 0.0`	Pass	0.000s
P = min(1, w/s).	`abs(p_flood_simple(3.29, 6.58) - 0.5) < 0.01`	Pass	0.000s
Returns float	`isinstance(result, float); isinstance(result, float)`	Pass	0.000s
Zero severe returns zero	`p_flood_poly(water_level=5.0, hour=48, severe_level=0.0) ...; p_flood_simple(5.0, 0.0) == 0.0`	Pass	0.000s
Zero water	`p_flood_simple(0.0, 6.58) == 0.0`	Pass	0.000s

5 GEV Hazard Model (MKM-GH-001)

Table 8: Test Results Summary — GEV Hazard Model (MKM-GH-001)

Total Tests	16
Passed	16
Failed	0
Skipped	0
Duration	0.26s

Table 9: Detailed Test Results — GEV Hazard Model

Test Description	Acceptance Criteria	Result	Time
Exceedance probability in range	`0.0 <= prob <= 1.0`	Pass	0.001s
Fit returns three floats	`isinstance(shape, float)`; `isinstance(loc, float)` (+1 more)	Pass	0.013s
Force gumbel sets shape zero	`shape == 0.0`	Pass	0.000s
Higher threshold lower probability	`p_high < p_low`	Pass	0.000s
Return level increases with period	`rl_100 > rl_10`	Pass	0.000s
Compute all responses returns dict	`isinstance(responses, dict)`; `len(responses) == len(synthetic_flat_gauges)` (+1 more)	Pass	0.000s
Compute response returns gauge response	`isinstance(response, GaugeResponse)`; `response.gauge_id == gauge_id` (+1 more)	Pass	0.000s
Response threshold flags	`response.exceeded_alert`; `response.exceeded_warning`	Pass	0.000s
Annual flood probabilities in range	`0.0 <= curve.annual_flood_prob_alert <= 1.0`; `0.0 <= curve.annual_flood_prob_warning <= 1.0` (+1 more)	Pass	0.081s
Build returns dict of curves	`isinstance(curves, dict)`; `len(curves) == len(synthetic_flat_gauges)` (+2 more)	Pass	0.078s
Each curve is gauge hazard curve	`isinstance(curve, GaugeHazardCurve)`; `curve.gauge_id == gauge_id`	Pass	0.086s
Cumulative prob increasing	`probs[i] >= probs[i - 1]`	Pass	0.000s
Probabilities sum to one	`abs(point.survival_prob + point.prob_at_least_one - 1.0) ...`	Pass	0.000s
Returns correct length	`len(ts) == 5`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Returns term structure points	`isinstance(point, TermStructurePoint)`	Pass	0.000s
Survival prob decreasing	`survivals[i] <= survivals[i - 1]`	Pass	0.000s

6 GaugeHD Synthetic (MKM-GHD-001)

Table 10: Test Results Summary — GaugeHD Synthetic (MKM-GHD-001)

Total Tests	62
Passed	62
Failed	0
Skipped	0
Duration	27.81s

Table 11: Detailed Test Results — GaugeHD Synthetic

Test Description	Acceptance Criteria	Result	Time
_snap_to_river skips zero-length segments without error.	`isinstance(lat, float); isinstance(lon, float)`	Pass	0.000s
_create_synthetic_at_position returns None when flanking gauge not in lookup.	`result is None`	Pass	0.003s
Lines 83-110: generate_all_gauge_histories processes all gauges.	`isinstance(result, list); len(result) == 1`	Pass	0.197s
Lines 105-107: error on one gauge -¿ logged, continues.	`isinstance(result, list)`	Pass	0.379s
All levels positive	`all(obs["level_meters"] > 0 for obs in result)`	Pass	0.006s
Correct length approximately	`728 <= len(result) <= 732`	Pass	0.003s
Dates are sequential	`(d_curr - d_prev).days == 1`	Pass	0.003s
Different seeds different output	`levels1 != levels2`	Pass	0.012s
Each obs has date and level	`"date" in obs; "level_meters" in obs` (+1 more)	Pass	0.001s
Gauge data without FloodGauge wrapper uses defaults.	`isinstance(result, list)`	Pass	0.001s
When historical high date is within range, that day gets historical_high level.	`match[0]["level_meters"] == 7.5`	Pass	0.001s
Invalid date format doesn't crash.	`isinstance(result, list)`	Pass	0.001s
Historical high date before start_date is ignored.	`isinstance(result, list)`	Pass	0.001s
Lines 176-179: Historical high injected via fallback when date not matched in loop.	`len(match) == 1; match[0]["level_meters"] == 8.0`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Gauge data with no FloodStages uses sensible defaults.	`isinstance(result, list); len(result) > 0`	Pass	0.001s
Lines 97-100: Partially tidal gauge uses dampened tidal signal.	`len(result) > 700; all(lev > 0 for lev in levels)`	Pass	0.003s
Returns list	`isinstance(result, list)`	Pass	0.001s
Seed reproducible	`r1 == r2`	Pass	0.003s
Lines 92-95: Tidal gauge uses wider daily_std, tidal_amplitude ¿ 0.	`len(result) > 700; max(levels) > min(levels)`	Pass	0.003s
Lines 147-149: Tidal component with spring-neap cycle.	`l1 != l2`	Pass	0.003s
Lines 138-141: historical high injected after main loop.	`len(match) == 1; match[0]["level_meters"] == 9.99`	Pass	0.004s
When historical_high_dt is None (out of range), post-loop block skipped.	`all(o["level_meters"] != 8.0 for o in result)`	Pass	0.001s
When historical_high_date is None, both branches skipped.	`isinstance(result, list); len(result) > 0`	Pass	0.001s
generate() returns count=0 when GAUGE_POINTS has ¡ 2 entries.	`result["count"] == 0`	Pass	0.002s
generate() returns count=0 when GAUGE_POINTS is None.	`result["count"] == 0`	Pass	0.001s
Lines 74-75: missing file raises FileNotFoundError.	—	Pass	0.000s
Lines 72-80: load_gauge_portfolio reads gauge.json.	`isinstance(result, list); len(result) == 1`	Pass	0.001s
_load_river_polyline returns None when cache file doesn't exist.	`result is None or isinstance(result, list)`	Pass	0.001s
_snap_to_river returns original coords when river is None.	`lat == 51.46; lon == -0.30`	Pass	0.000s
Synthetic gauges should not be placed at exact polyline endpoints.	`not (loc["GaugeLatitude"] == gp_lat and ...`	Pass	0.365s
Lines 143-144: bad CSV -¿ error logged, continues.	`isinstance(result, list)`	Pass	0.001s
Lines 113-147: process_nrfa_directory scans CSV files.	`isinstance(result, list); len(result) == 1`	Pass	0.002s
Line 134: output_dir=None uses config.get_gaugehd_dir().	`isinstance(result, list)`	Pass	0.002s
Lines 131-147: no matching files -¿ empty list.	`result == []`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Propertyhc has synth in nearest gauges	`found_synth`	Pass	0.005s
Synthetic gauge in propertyhc should have basis_bps computed.	`"basis_bps" in ng`	Pass	0.004s
Every property has one synthetic gauge	`not failures`	Pass	1.844s
Every property has three gauges total	`len(ngs) == 3`	Pass	0.040s
Every property has two real gauges	`not failures`	Pass	1.454s
Generator no longer needs property.json — uses random positions on river.	`result["count"] > 0`	Pass	0.026s
Generator should create up to the requested count of synthetic gauges.	`0 < result["count"] <= 3`	Pass	0.009s
Properties should have flood_events sourced from gauge storm responses.	`properties_with_events > 0`	Pass	1.532s
At least some properties should have flooded=True events.	`flooded_count > 0`	Pass	1.462s
Appends to gauge json	`len(data["flood_gauges"]) > 3`; `len(synth) > 0`	Pass	2.244s
Creates synthetic gauges	`result["count"] > 0`; `len(result["gauge_ids"]) == result["count"]`	Pass	2.225s
Synthetic gauges placed too close together should be deduplicated.	`result["count"] < 200`; `result["count"] > 0`	Pass	2.182s
Interpolated elevation	`3.0 <= elev <= 5.0`	Pass	2.224s
Interpolated flood stages	`3.5 <= stages["FloodAlert"] <= 4.5`; `4.5 <= stages["FloodWarning"] <= 5.5` (+1 more)	Pass	2.220s
No gauge json returns zero	`result["count"] == 0`	Pass	0.000s
Generator no longer needs property.json — generates random positions on river.	`result["count"] > 0`	Pass	2.179s
Requesting a small number of synthetic gauges should work.	`0 < result["count"] <= 5`	Pass	0.013s
Synthetic gauge has cdm structure	`"GaugeID" in fg["Header"]`; `"CatchmentID" in fg["Header"]` (+8 more)	Pass	2.202s
Synthetic gauge lat/lon should be on the polyline (very close to it).	`dist_m < 50.0`; `dist_m < 1.0`	Pass	2.415s

Test Description	Acceptance Criteria	Result	Time
A pre-existing SYNTH- gauge in gauge.json is skipped in polyline matching.	`result["count"] >= 0`	Pass	2.197s
Gauge json has synth gauges	`len(synth) > 0`	Pass	0.002s
Synth gauge has required cdm fields	`fg["Header"]["GaugeID"].startswith("SYNTH")`, `fg["Location"]["GaugeLatitude"] != 0` (+7 more)	Pass	0.002s
Gaugehc has synth curves	`len(synth) > 0`	Pass	0.013s
Synth has term structure	`len(ts) > 0`	Pass	0.010s
Synth hazard rates positive	`hc.get("annual_hazard_rate_alert", 0) > 0`	Pass	0.010s
Synth gaugets files exist	`len(synth_files) > 0`	Pass	0.001s
At least some storms should exceed alert at synthetic gauges.	`len(exceeded) > 0`	Pass	0.138s
Synth gaugets have storm responses	`len(resps) > 0`	Pass	0.152s

7 Insurance Premium (MKM-IP-001)

Table 12: Test Results Summary — Insurance Premium (MKM-IP-001)

Total Tests	36
Passed	36
Failed	0
Skipped	0
Duration	0.00s

Table 13: Detailed Test Results — Insurance Premium

Test Description	Acceptance Criteria	Result	Time
All factors applied	`abs(result - round(expected, 2)) < 0.01`	Pass	0.000s
Basic calculation	`result == 1250.0`	Pass	0.000s
Claims factor increases premium	`with_claims > base`	Pass	0.000s
Clamped to max	`result == MAX_PREMIUM`	Pass	0.000s
Clamped to min	`result == MIN_PREMIUM`	Pass	0.000s
Contents premium added	`with_contents == base + 500.0`	Pass	0.000s
High flood risk increases premium	`high_flood > low_flood`	Pass	0.000s
Returns float	`isinstance(result, float)`	Pass	0.000s
Security factor reduces premium	`with_security < without`	Pass	0.000s
100 years is heritage	`get_age_premium_band(1925, current_year=2025) == 'heritage'`	Pass	0.000s
10 years is modern	`get_age_premium_band(2015, current_year=2025) == 'modern'`	Pass	0.000s
30 years is established	`get_age_premium_band(1995, current_year=2025) == 'establi...`	Pass	0.000s
Default current year	`result in AGE_PREMIUM_FACTORS`	Pass	0.000s
Established	`get_age_premium_band(1960, current_year=2025) == 'establi...`	Pass	0.000s
Exactly 29 years modern	`get_age_premium_band(1996, current_year=2025) == 'modern'`	Pass	0.000s
Exactly 99 years established	`get_age_premium_band(1926, current_year=2025) == 'establi...`	Pass	0.000s
Exactly 9 years new build	`get_age_premium_band(2016, current_year=2025) == 'new_build'`	Pass	0.000s
Heritage old	`get_age_premium_band(1850, current_year=2025) == 'heritage'`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Modern	`get_age_premium_band(2000, current_year=2025) == 'modern'`	Pass	0.000s
New build	`get_age_premium_band(2020, current_year=2025) == 'new_build'`	Pass	0.000s
Exactly 100 third band	`(lo, hi) == AREA_PREMIUM_BANDS[2][1]`	Pass	0.000s
Exactly 50 second band	`(lo, hi) == AREA_PREMIUM_BANDS[1][1]`	Pass	0.000s
Just below 100 second band	`(lo, hi) == AREA_PREMIUM_BANDS[1][1]`	Pass	0.000s
Just below 50 first band	`(lo, hi) == AREA_PREMIUM_BANDS[0][1]`	Pass	0.000s
Large area fourth band	`(lo, hi) == AREA_PREMIUM_BANDS[3][1]`	Pass	0.000s
Medium area second band	`(lo, hi) == AREA_PREMIUM_BANDS[1][1]`	Pass	0.000s
Returns tuple	`isinstance(result, tuple); len(result) == 2`	Pass	0.000s
Small area first band	`(lo, hi) == AREA_PREMIUM_BANDS[0][1]`	Pass	0.000s
Age premium factors all positive	`lo > 0; hi >= lo`	Pass	0.000s
Area bands are sorted	`thresholds == sorted(thresholds)`	Pass	0.000s
Base rate range	`lo > 0; hi > lo`	Pass	0.000s
Construction type factors present	`'Brick' in CONSTRUCTION_TYPE_FACTORS; 'Timber Frame' in CONSTRUCTION_TYPE_FACTORS`	Pass	0.000s
Flood risk factors increase with risk	`very_high_hi > very_low_lo`	Pass	0.000s
Flood risk factors present	`set(FLOOD_RISK_PREMIUM_FACTORS.keys()) == expected`	Pass	0.000s
Min max premium bounds	`MIN_PREMIUM == 200; MAX_PREMIUM == 20_000`	Pass	0.000s
Property type factors present	`set(PROPERTY_TYPE_PREMIUM_FACTORS.keys()) == expected`	Pass	0.000s

8 Mortgage Pricer (MKM-MP-001)

Table 14: Test Results Summary — Mortgage Pricer (MKM-MP-001)

Total Tests	67
Passed	67
Failed	0
Skipped	0
Duration	0.05s

Table 15: Detailed Test Results — Mortgage Pricer

Test Description	Acceptance Criteria	Result	Time
Default ids assigned	`results[0]["mortgage_id"] == "MORTGAGE_0"`; `results[0]["property_id"] == "PROPERTY_0"`	Pass	0.001s
Error isolation	`len(results) == 3`; `"error" not in results[0]` (+2 more)	Pass	0.001s
Prices multiple mortgages	`len(results) == 2`; `results[0]["mortgage_id"] == "M1"` (+1 more)	Pass	0.001s
Custom tax rate	`MortgagePricer(tax_rate=0.40).tax_rate == 0.40`	Pass	0.000s
Default tax rate	`MortgagePricer().tax_rate == 0.20`	Pass	0.000s
Custom tax rate	`spread_high_tax > spread_low_tax`	Pass	0.000s
Floor at 10bps	`spread >= 0.001`	Pass	0.000s
Higher affordability higher spread	`spread_low_income > spread_high_income`	Pass	0.000s
Negative income fallback	`spread == 0.15`	Pass	0.000s
Spread bounded by schedule	`spread >= 0.001`; `spread < 0.05`	Pass	0.000s
Zero income fallback	`spread == 0.15`	Pass	0.000s
debug=True in the credit spread calculation.	`"credit_spread" in result`	Pass	0.001s
debug=True exercises logging branches without error.	`result["mortgage_value"] > 0`	Pass	0.001s
Zero income with debug=True exercises zero-income debug branch.	`result["credit_spread"] > 0`	Pass	0.001s
flood_risk_category=None with debug=True.	`result["mortgage_value"] > 0`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
When interest_rate=0, monthly_rate==0 → loan_amount/n_periods formula.	`result["mortgage_value"] > 0`	Pass	0.001s
Batch passes flood risk	`results[0]["flood_risk_factor"] == 1.40; results[1]["flood_risk_factor"] == 1.00` (+1 more)	Pass	0.001s
Case insensitive	`MortgagePricer.calculate_flood_risk_impact("very high") =...; MortgagePricer.calculate_flood_risk_impact("VERY LOW") ==...` (+1 more)	Pass	0.000s
Flood factor in result	`result["flood_risk_factor"] == 1.20`	Pass	0.001s
Flood risk increases spread	`high["credit_spread"] > low["credit_spread"]; high["flood_risk_factor"] > low["flood_risk_factor"]` (+1 more)	Pass	0.001s
Flood risk multipliers[high-1.4] [category='High', expected=1.4]	`MortgagePricer.calculate_flood_risk_impact(category) == e...`	Pass	0.000s
Flood risk multipliers[low-1.05] [category='Low', expected=1.05]	`MortgagePricer.calculate_flood_risk_impact(category) == e...`	Pass	0.000s
Flood risk multipliers[medium-1.2] [category='Medium', expected=1.2]	`MortgagePricer.calculate_flood_risk_impact(category) == e...`	Pass	0.000s
Flood risk multipliers[very high-1.75] [category='Very High', expected=1.75]	`MortgagePricer.calculate_flood_risk_impact(category) == e...`	Pass	0.000s
Flood risk multipliers[very low-1.0] [category='Very Low', expected=1.0]	`MortgagePricer.calculate_flood_risk_impact(category) == e...`	Pass	0.000s
No flood risk defaults to unity	`result["flood_risk_factor"] == 1.0`	Pass	0.001s
None returns unity	`MortgagePricer.calculate_flood_risk_impact(None) == 1.0`	Pass	0.000s
Unknown category returns unity	`MortgagePricer.calculate_flood_risk_impact("Unknown") == 1.0; MortgagePricer.calculate_flood_risk_impact("Extreme") == 1.0`	Pass	0.000s
Ltv thresholds[0.5-1.0] [ltv=0.5, expected=1.0]	`pricer.calculate_loan_to_value_impact(pv * ltv, pv) == ex...`	Pass	0.000s
Ltv thresholds[0.8-1.0] [ltv=0.8, expected=1.0]	`pricer.calculate_loan_to_value_impact(pv * ltv, pv) == ex...`	Pass	0.000s
Ltv thresholds[0.85-1.1] [ltv=0.85, expected=1.1]	`pricer.calculate_loan_to_value_impact(pv * ltv, pv) == ex...`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Ltv thresholds[0.9-1.1] [ltv=0.9, expected=1.1]	`pricer.calculate_loan_to_value_impact(pv * ltv, pv) == ex...`	Pass	0.000s
Ltv thresholds[0.91-1.3] [ltv=0.91, expected=1.3]	`pricer.calculate_loan_to_value_impact(pv * ltv, pv) == ex...`	Pass	0.000s
Ltv thresholds[0.95-1.3] [ltv=0.95, expected=1.3]	`pricer.calculate_loan_to_value_impact(pv * ltv, pv) == ex...`	Pass	0.000s
Ltv thresholds[0.96-1.5] [ltv=0.96, expected=1.5]	`pricer.calculate_loan_to_value_impact(pv * ltv, pv) == ex...`	Pass	0.000s
Ltv thresholds[1.0-1.5] [ltv=1.0, expected=1.5]	`pricer.calculate_loan_to_value_impact(pv * ltv, pv) == ex...`	Pass	0.000s
Zero property value	`pricer.calculate_loan_to_value_impact(100_000, 0) == 1.5`	Pass	0.000s
Higher rate higher payment	`high > low`	Pass	0.000s
Shorter term higher payment	`short > long`	Pass	0.000s
£200k at 3.5% over 25yr £1,001.25/month.	`995 < payment < 1010`	Pass	0.000s
Zero-rate degenerate case: M = P / n.	`payment == pytest.approx(1000.0)`	Pass	0.000s
Each mortgage has property id	`mapping.get("property_id") is not None`	Pass	0.004s
Processing stats	`stats["successful_mortgages"] == 5; stats["failed_mortgages"] == 0`	Pass	0.005s
Output file exists	`result["file_path"].exists()`	Pass	0.004s
Output file is valid json	`isinstance(data, dict)`	Pass	0.006s
Generate returns expected keys	`"data" in result; "file_path" in result` (+1 more)	Pass	0.003s
Mortgage count matches properties	`len(result["data"]["mortgages"]) == 5`	Pass	0.004s
Mortgage id format	`re.match(r"^MORT-[a-f0-9]{8}$", mid)`	Pass	0.003s
Mortgage ids unique	`len(ids) == len(set(ids))`	Pass	0.004s
All errors returns error	`"error" in calculate_portfolio_metrics(results)`	Pass	0.000s
High risk count	`calculate_portfolio_metrics(results)["high_risk_mortgages"]`	Pass	0.000s
Mixed results exclude errors	`metrics["total_mortgages"] == 1; metrics["error_count"] == 1`	Pass	0.001s
Valid results	`metrics["total_mortgages"] == 2; metrics["error_count"] == 0` (+3 more)	Pass	0.002s
Balance converges to zero	`result["outstanding_balance"][-1] < 1.0`	Pass	0.001s
Expected losses non negative	`all(el >= 0 for el in result["expected_losses"])`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Fair value less than par	`result["mortgage_value"] < base_mortgage_params["loan_amo...; result["discount_to_par"] > 0` (+1 more)	Pass	0.001s
Hazard rates positive	`result["hazard_rates"][0] == 0.0; all(result["hazard_rates"][i] > 0 for i in range(1, len(r...`	Pass	0.001s
High ltv increases spread	`high_ltv["credit_spread"] > low_ltv["credit_spread"]; high_ltv["ltv_factor"] > low_ltv["ltv_factor"]`	Pass	0.001s
Higher haircut more loss	`high["pv_losses"] > low["pv_losses"]`	Pass	0.001s
Input validation clamps	`result["mortgage_value"] > 0`	Pass	0.000s
Ltv factor in result	`"ltv_factor" in result; result["ltv_factor"] == 1.0`	Pass	0.001s
Raises on invalid types	—	Pass	0.000s
Recovery haircut zero low ltv	`all(lgd == 0 for lgd in result["lgds"])`	Pass	0.001s
Survival first period less than one	`result["survival_probs"][1] < 1.0`	Pass	0.001s
Survival monotonically decreasing	`surv[i] <= surv[i - 1]`	Pass	0.001s
Positive rate exceeds principal	`cost > 200_000`	Pass	0.000s
Zero rate equals principal	`cost == pytest.approx(100_000.0)`	Pass	0.000s

9 PRS Pricing Model (MKM-PR-001)

Table 16: Test Results Summary — PRS Pricing Model (MKM-PR-001)

Total Tests	57
Passed	57
Failed	0
Skipped	0
Duration	7.70s

Table 17: Detailed Test Results — PRS Pricing Model

Test Description	Acceptance Criteria	Result	Time
1Y from Feb 2026 = last Friday May 2027.	`mat == last_friday_of_month(2027, 5)`	Pass	0.000s
3Y from Feb 2026 = last Friday May 2029.	`mat == last_friday_of_month(2029, 5)`; `mat.month == 5` (+1 more)	Pass	0.000s
Jan 2027 uses Aug 2026 roll → Nov 2029 maturity.	`mat == last_friday_of_month(2029, 11)`	Pass	0.000s
3Y from Sep 2026 = last Friday Nov 2029 (Aug roll).	`mat == last_friday_of_month(2029, 11)`; `mat.month == 11`	Pass	0.000s
5Y from Feb 2026 = last Friday May 2031.	`mat == last_friday_of_month(2031, 5)`	Pass	0.000s
5Y from Sep 2026 = last Friday Nov 2031.	`mat == last_friday_of_month(2031, 11)`	Pass	0.000s
All maturities should fall on a Friday.	`mat.weekday() == 4`	Pass	0.000s
3Y maturity jumps from May to Nov at the Aug roll.	`may_mat.month == 5`; `nov_mat.month == 11` (+1 more)	Pass	0.000s
Aug onwards uses Aug roll.	`roll == last_friday_of_month(2026, 8)`; `roll.month == 8`	Pass	0.000s
December uses the same year's Aug roll.	`roll.month == 8`; `roll.year == 2026`	Pass	0.000s
Feb-Jul uses Feb roll.	`roll == last_friday_of_month(2026, 2)`; `roll.month == 2`	Pass	0.000s
January uses the previous year's Aug roll.	`roll == last_friday_of_month(2026, 8)`; `roll.month == 8` (+1 more)	Pass	0.000s
July is the last month of the Feb roll period.	`roll.month == 2`; `roll.year == 2026`	Pass	0.000s
March is still in the Feb roll period.	`roll.month == 2`; `roll.year == 2026`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
compute_maturity_date with no ref_date uses today.	`isinstance(mat, date); mat.weekday() == 4`	Pass	0.000s
current_roll_date(None) uses today and returns a valid date.	`isinstance(roll, date); roll.weekday() == 4`	Pass	0.000s
maturity_table with no ref_date uses today.	`len(table) == 5; all(isinstance(e['maturity_date'], date) for e in table)`	Pass	0.000s
Empty propertyts dir	`stats["total_properties"] == 0; stats["properties_processed"] == 0`	Pass	0.011s
Generator should still work without gauge hazard curves (empty basis).	`stats["properties_processed"] == 3; stats["properties_skipped"] == 0` (+1 more)	Pass	0.272s
No propertyts dir	—	Pass	0.000s
Spread is flat term structure	`all(s == spreads[0] for s in spreads)`	Pass	0.273s
Spread proportional to floods	`spread3 > spread1 > spread2`	Pass	0.278s
Zero floods zero spread	`spreads[0] == expected`	Pass	0.318s
Returns severe event count	`gen._get_gauge_severe_count(gauge_hc) == 7`	Pass	0.000s
Returns zero for empty dict	`gen._get_gauge_severe_count({}) == 0`	Pass	0.000s
Returns zero when missing	`gen._get_gauge_severe_count(gauge_hc) == 0`	Pass	0.000s
Aug 2026: 31st is Monday, last Friday = 28th.	`last_friday_of_month(2026, 8) == date(2026, 8, 28)`	Pass	0.000s
Feb 2026: 28th is Saturday, last Friday = 27th.	`last_friday_of_month(2026, 2) == date(2026, 2, 27)`	Pass	0.000s
May 2029: 31st is Thursday, last Friday = 25th.	`last_friday_of_month(2029, 5) == date(2029, 5, 25)`	Pass	0.000s
Nov 2029: 30th is Friday.	`last_friday_of_month(2029, 11) == date(2029, 11, 30)`	Pass	0.000s
Result should always be a Friday (weekday=4).	`d.weekday() == 4`	Pass	0.000s
Actual years should be strictly increasing.	`years[i] > years[i - 1]`	Pass	0.000s
All maturities should be in May for the Feb roll.	`e['maturity_date'].month == 5`	Pass	0.000s
All maturities should be in November for the Aug roll.	`e['maturity_date'].month == 11`	Pass	0.000s
Maturity string should be formatted as 'DD Mon YYYY'.	`len(e['maturity_str']) > 8; 'May' in e['maturity_str'] or 'Nov' in e['maturity_str']`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Default table has 5 entries (1Y through 5Y).	`len(table) == 5`	Pass	0.000s
Tenors should be 1, 2, 3, 4, 5.	`tenors == [1, 2, 3, 4, 5]`	Pass	0.000s
Basis calculation	`len(nearest) > 0`; `ng0["gauge_id"] == "GAUGE-001"` (+5 more)	Pass	0.315s
Depth thresholds	`"severe" in thresholds`; `"annual_probability" in thresholds["severe"]` (+1 more)	Pass	0.296s
Generate produces output	`output_path.exists()`; `"metadata" in data` (+2 more)	Pass	0.278s
Gev params always none	`pc["gev_params"] is None`	Pass	0.296s
PROP-003 has 10 floods, nearest gauge has 8 severe events.	`prop3["summary"]["flood_transmission_rate"] > 0`	Pass	0.305s
Output structure	`"PROP-001" in curves`; `"PROP-002" in curves` (+13 more)	Pass	0.308s
Properties processed	`stats["properties_processed"] == 3`; `stats["properties_skipped"] == 0`	Pass	0.279s
With event count pricing, term structure spread is flat (storms independent).	`s == spreads[0]`	Pass	0.313s
Term structure	`ts["tenors"] == TENORS`; `"severe" in ts` (+2 more)	Pass	0.290s
After attach_spread_decomposition, propertyhc.json has decomposition.	`"spread_decomposition" in pc`; `"gauge_spread_bps" in sd` (+5 more)	Pass	0.825s
At least one effect per path must be non-zero when gauge != property.	`df["distance_effect_bps"] != 0.0 or df["elevation_effect_...`; `ef["elevation_effect_bps"] != 0.0 or ef["distance_effect_...`	Pass	0.870s
Both decomposition paths should arrive at the property spread.	`abs(path1 - prop) < 0.1`; `abs(path2 - prop) < 0.1`	Pass	1.057s
With shd/she files present, decomposition values must be non-zero.	`sd["property_spread_bps"] != 0.0`; `sd["shd_spread_bps"] != 0.0` (+1 more)	Pass	0.830s
Without shd/she files, decomposition still works (zeros).	`n > 0`; `sd["shd_spread_bps"] == 0.0` (+1 more)	Pass	0.282s
_get_5yr_spread must return index 4 when available.	`PropertyHazardCurveGenerator._get_5yr_spread(ps) == 50.0`	Pass	0.000s
_get_5yr_spread must return 0.0 for empty/missing data.	`PropertyHazardCurveGenerator._get_5yr_spread(ps) == 0.0`; `PropertyHazardCurveGenerator._get_5yr_spread( ...` (+1 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
_get_5yr_spread must work with specified threshold.	`PropertyHazardCurveGenerator._get_5yr_spread(pc, 'severe'...; PropertyHazardCurveGenerator._get_5yr_spread(pc, 'any_flo...`	Pass	0.000s
All tenors should round-trip through maturity and back.	`tenor_from_maturity(mat, ref) == t`	Pass	0.000s
A date not in the table returns the closest tenor.	`result in [1, 2, 3, 4, 5]`	Pass	0.000s
Exact maturity date returns correct tenor.	`tenor_from_maturity(mat, date(2026, 2, 24)) == 3`	Pass	0.000s

10 Property Valuation (MKM-PV-001)

Table 18: Test Results Summary — Property Valuation (MKM-PV-001)

Total Tests	55
Passed	55
Failed	0
Skipped	0
Duration	16.55s

Table 19: Detailed Test Results — Property Valuation

Test Description	Acceptance Criteria	Result	Time
Line 541: non-dict schema → returns {}.	`result == {}`	Pass	0.000s
Line 544-545: 'type'/'options'/'description' fields are skipped.	`"type" not in result; "options" not in result`	Pass	0.000s
Calculate insurance premium flood risk high	`premium > 0`	Pass	0.099s
Line 214: area=None → computes it first.	`isinstance(premium, float); premium > 0`	Pass	0.086s
Line 211: property_value=None → computes it first.	`isinstance(premium, float); premium > 0`	Pass	0.171s
Calculate monthly rent all types	`800 <= rent <= 15_000`	Pass	0.000s
Lines 165-166: area=None → computes it first.	`isinstance(rent, float); rent > 0`	Pass	0.000s
Lines 162-163: property_value=None → computes it first.	`isinstance(rent, float); rent > 0`	Pass	0.086s
Calculate monthly rent positive	`isinstance(rent, float); 800 <= rent <= 15_000`	Pass	0.000s
Calculate property area all types	`area > 0`	Pass	0.000s
Lines 128-129: invalid date format → random years_ago.	`isinstance(price, float)`	Pass	0.000s
Lines 120-121: property_value=None → computes it first.	`isinstance(price, float); price > 0`	Pass	0.086s
Calculate sale price positive	`isinstance(price, float); 50_000 <= price <= 8_000_000`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Lines 135-137: sale_date 1-2 years ago.	`isinstance(price, float)`	Pass	0.000s
Lines 139-140: sale_date ¿ 2 years ago.	`isinstance(price, float)`	Pass	0.000s
Lines 124-128: sale_date within 1 year.	`isinstance(price, float)`	Pass	0.000s
Line 60-61: datetime → isoformat string.	`"2024-01-01" in result`	Pass	0.000s
Lines 64-65: np.floating → float.	`"3.14" in result`	Pass	0.000s
Lines 62-63: np.integer → int.	`result == "42"`	Pass	0.000s
Lines 66-67: np.ndarray → list.	`result == [1, 2, 3]`	Pass	0.000s
Line 68: super().default() raises TypeError for unknown type.	—	Pass	0.000s
Lines 510-517: each fallback location has lat/lon/name/elevation.	`"lat" in loc`; `"lon" in loc` (+2 more)	Pass	0.000s
Lines 499-518: fallback generates right number of locations.	`len(locs) == 5`	Pass	0.000s
Lines 505-506: uses params.get_elevation if available.	`all(loc["elevation"] == 10.0 for loc in locs)`	Pass	0.000s
Lines 608-612: generate_properties returns result dict.	`isinstance(result, dict)`; `"data" in result` (+1 more)	Pass	0.783s
Lines 610-612: generate_properties creates output file.	`result["file_path"].exists()`	Pass	0.512s
Calculate mortgage financials returns dict	`isinstance(result, dict)`; `"loan_amount" in result` (+2 more)	Pass	0.000s
Generate financial data keys	`"mortgage_id" in data`; `"loan_amount" in data` (+1 more)	Pass	0.000s
Interest rate range	`0.01 <= data["interest_rate"] <= 0.15`	Pass	0.000s
Loan amount less than value	`data["loan_amount"] <= info["property_value"] * 1.05`	Pass	0.000s
Mortgage id format	`re.match(r"^MORT-[a-f0-9]{8}$", data["mortgage_id"])`	Pass	0.000s
Quality consistency check	`isinstance(corrected, dict)`	Pass	0.000s
Calculate insurance premium positive	`isinstance(premium, (int, float))`; `premium > 0`	Pass	0.086s
Calculate property area positive	`isinstance(area, (int, float))`; `area > 0`	Pass	0.000s
Calculate property value in range	`50_000 <= value <= 10_000_000`	Pass	0.955s
Generate bedrooms detached	`isinstance(beds, int)`; `1 <= beds <= 7`	Pass	0.000s
Generate bedrooms flat	`isinstance(beds, int)`; `1 <= beds <= 4`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Generate postcode format	`isinstance(postcode, str)`; `len(postcode) >= 5`	Pass	0.000s
Generate field value decimal	`isinstance(val, (int, float))`	Pass	0.172s
Generate field value text	`isinstance(val, str)`	Pass	0.086s
Coordinates in thames range	`51.0 <= loc["lat"] <= 52.0`; `-1.0 <= loc["lon"] <= 1.0`	Pass	2.657s
Processing stats	`stats["successful_properties"] == 5`; `stats["failed_properties"] == 0`	Pass	1.291s
Output file exists	`result["file_path"].exists()`	Pass	1.308s
Output file is valid json	`isinstance(data, dict)`	Pass	1.299s
Generate correct count	`len(result["data"]["properties"]) == 5`	Pass	1.390s
Generate returns expected keys	`"data" in result`; `"file_path" in result` (+1 more)	Pass	1.329s
Property id format	`re.match(r"^PROP-[a-f0-9]{8}$", pid)`	Pass	1.392s
Property ids unique	`len(ids) == len(set(ids))`	Pass	2.563s
Construction year in range	`1600 <= year <= 2026`	Pass	0.000s
Property id format	`re.match(r"^PROP-[a-f0-9]{8}$", meta["property_id"])`	Pass	0.095s
Returns dict with required keys	`"property_id" in meta`; `"property_type" in meta`	Pass	0.095s
Lines 73-75: generate_grid_reference returns a non-empty string.	`isinstance(ref, str)`; `len(ref) > 4`	Pass	0.000s
Lines 50-52: generate_owner_name returns 'Firstname Lastname'.	`len(parts) == 2`; `all(p[0].isupper() for p in parts)`	Pass	0.000s
Lines 43-45: generate_past_date returns YYYY-MM-DD string.	`isinstance(result, str)`; `len(parts) == 3`	Pass	0.000s
Custom days_range is respected (result is a past date).	`dt < datetime.now()`	Pass	0.000s

11 Storm-Gauge Response Model (MKM-SG-001)

Table 20: Test Results Summary — Storm-Gauge Response Model (MKM-SG-001)

Total Tests	41
Passed	41
Failed	0
Skipped	0
Duration	0.04s

Table 21: Detailed Test Results — Storm-Gauge Response Model

Test Description	Acceptance Criteria	Result	Time
Encodes datetime	`"2026-01-01" in result`	Pass	0.000s
Line 80: np.float32 is np.floating but NOT a Python float subclass in NumPy 2.x,	`"2.5" in result`	Pass	0.000s
Encodes numpy floating	`"3.14" in result`	Pass	0.000s
Encodes numpy integer	`"42" in result`	Pass	0.000s
Encodes numpy ndarray	`"1.0" in result`	Pass	0.000s
Fallback raises for unknown	—	Pass	0.000s
Close to thames high risk	`result["FloodRiskScore"] >= 7; result["FloodRiskCategory"] in ["High", "Very High"]`	Pass	0.000s
Far from thames lower risk	`result["FloodRiskScore"] <= 6; result["FloodRiskCategory"] in ["Low", "Medium"]`	Pass	0.000s
Flood stages ordering	`alert < warning < severe`	Pass	0.002s
Processing stats	`stats["successful_gauges"] == 5; stats["failed_gauges"] == 0`	Pass	0.001s
Lines 264-266: exception during json.dump is logged and re-raised.	—	Pass	0.000s
Line 245: end_time is set by the first generate(); the second generate() call	—	Pass	0.001s
Line 290: non-dict section_schema returns {}.	`result == {}`	Pass	0.000s
Line 294: 'type', 'options', 'description', 'values' keys are skipped.	`skip not in result`	Pass	0.000s
Lines 159-162: count ¿ max_gauges → capped to max_gauges.	`len(result["data"]["flood_gauges"]) == max_available`	Pass	0.006s

Test Description	Acceptance Criteria	Result	Time
Line 155-156: count ¡ 1 raises ValueError.	—	Pass	0.000s
Lines 354-355: generate_gauges() convenience function.	"data" in result; len(result["data"]["flood_gauges"]) == 3	Pass	0.001s
Lines 212-215: exception in _generate_single_gauge increments failed_gauges.	result["processing_stats"]["failed_gauges"] == 1	Pass	0.000s
Lines 122-125: log() dispatches to correct logger method.	—	Pass	0.000s
Lines 309-327: missing FloodGauge/Header/Location/FloodStages are created.	gauge_data["FloodGauge"]["Header"]["GaugeID"] == "GAUGE-t...; gauge_data["FloodGauge"]["Location"]["GaugeLatitude"] == ...	Pass	0.000s
Lines 99-100: verbose=False sets logger to WARNING.	logging.getLogger("port.src.gauge.gauge").level == loggin...	Pass	0.000s
Output file exists	result["file_path"].exists()	Pass	0.001s
Output file is valid json	"flood_gauges" in data	Pass	0.008s
Output json has correct count	len(data["flood_gauges"]) == 5	Pass	0.004s
Lines 378-381: __main__ block calls generate_gauges and logs results.	(tmp_path / 'gauge.json').exists()	Pass	0.005s
Gauge id format	re.match(r"^GAUGE-[a-f0-9]{8}$", gid)	Pass	0.001s
Gauge ids unique	len(ids) == len(set(ids))	Pass	0.002s
Generate correct count	len(result["data"]["flood_gauges"]) == 5	Pass	0.001s
Generate returns expected keys	"data" in result; "file_path" in result (+1 more)	Pass	0.003s
Locations have coordinates	"lat" in loc; "lon" in loc (+1 more)	Pass	0.001s
Date field returns date string	isinstance(val, str); re.match(r"^\{}d{4}-\{}d{2}-\{}d{2}$", val)	Pass	0.000s
Decimal field returns float	isinstance(val, (int, float))	Pass	0.000s
Gauge name includes area	"TestArea" in val	Pass	0.000s
Menu field returns valid option	isinstance(val, str)	Pass	0.000s
Text field returns string	isinstance(val, str); len(val) > 0	Pass	0.000s
Catchment name stored	meta["catchment_name"] == "Thames"	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Flood stage ordering	`meta["flood_alert"] < meta["flood_warning"]; meta["flood_warning"] < meta["severe_flood_warning"]` (+1 more)	Pass	0.000s
Gauge id format	`re.match(r"^GAUGE-[a-f0-9]{8}$", meta["gauge_id"])`	Pass	0.000s
Historical high level range	`5.0 <= meta["historical_high_level"] <= 8.0`	Pass	0.000s
Install date format	`re.match(r"^\{}d{4}-\{}d{2}-\{}d{2}$", meta["install_date"])`	Pass	0.000s
Returns dict with required keys	`key in meta`	Pass	0.000s

12 Storm Intensity Distribution (MKM-SI-001)

Table 22: Test Results Summary — Storm Intensity Distribution (MKM-SI-001)

Total Tests	63
Passed	63
Failed	0
Skipped	0
Duration	0.02s

Table 23: Detailed Test Results — Storm Intensity Distribution

Test Description	Acceptance Criteria	Result	Time
All categories defined	`set(DURATION_PARAMS.keys()) == expected`	Pass	0.000s
Catastrophic extended	`high == 240`	Pass	0.000s
Each tier's max roughly doubles the previous.	`1.5 <= ratio <= 3.0`	Pass	0.000s
Extreme extended	`DURATION_PARAMS["extreme"] == (36, 72, 144)`	Pass	0.000s
Moderate extended	`low == 12`; `mode == 30` (+1 more)	Pass	0.000s
Severe extended	`DURATION_PARAMS["severe"] == (24, 48, 96)`	Pass	0.000s
Gap types	`GapType.SHORT.value == "short"`; `GapType.MEDIUM.value == "medium"` (+1 more)	Pass	0.000s
Sequence gap mapping	`SEQUENCE_GAP_TYPE[SequenceType.DOUBLET] == GapType.MEDIUM`; `SEQUENCE_GAP_TYPE[SequenceType.CLUSTER] == GapType.MEDIUM` (+2 more)	Pass	0.000s
Sequence types	`SequenceType.ISOLATED.value == "isolated"`; `SequenceType.DOUBLET.value == "doublet"` (+2 more)	Pass	0.000s
Storm count ranges	`SEQUENCE_TYPE_STORM_COUNTS[SequenceType.ISOLATED] == (1, 1)`; `SEQUENCE_TYPE_STORM_COUNTS[SequenceType.DOUBLET] == (2, 2)` (+2 more)	Pass	0.000s
Constants	`EVENT_WINDOW_HOURS == 168`; `MIN_DRAINAGE_WINDOW_HOURS == 12` (+1 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Exact boundary	`fits_in_window([78.0, 78.0], [0.0])`	Pass	0.000s
Exceeds window	`not fits_in_window([72.0, 72.0], [48.0])`	Pass	0.000s
Simple doublet fits	`fits_in_window([24.0, 24.0], [48.0])`	Pass	0.000s
Single storm	`fits_in_window([100.0], [])`	Pass	0.000s
Long gaps	`GAP_PARAMS[GapType.LONG] == (48, 96, 144)`	Pass	0.000s
Medium gaps	`GAP_PARAMS[GapType.MEDIUM] == (24, 48, 72)`	Pass	0.000s
Short gaps	`GAP_PARAMS[GapType.SHORT] == (6, 18, 36)`	Pass	0.000s
Doublet medium	`get_gap_range(SequenceType.DOUBLET) == (24, 48, 72)`	Pass	0.000s
Isolated zero	`get_gap_range(SequenceType.ISOLATED) == (0, 0, 0)`	Pass	0.000s
Get duration range	`get_duration_range("moderate") == (12, 30, 60)`	Pass	0.000s
Get max duration	`get_max_duration("catastrophic") == 240; get_max_duration("minimal") == 12`	Pass	0.000s
Ids unique	`len(ids) == 100`	Pass	0.000s
Sequence id format	`sid.startswith("STORM-"); len(sid) == 14`	Pass	0.000s
Sequence ids unique	`len(ids) == 100`	Pass	0.000s
Storm id format	`sid.startswith("STORM-"); len(sid) == 14`	Pass	0.000s
Invalid category	—	Pass	0.000s
Mode should be near the most frequent value in a large sample.	`abs(median - mode) < 15`	Pass	0.005s
Line 50: rng=None → creates np.random.RandomState() internally.	`low <= d <= high`	Pass	0.000s
Reproducibility	`d1 == d2`	Pass	0.000s
All sampled durations fall within [min, max] for the category. [category='baseline']	`low <= d <= high`	Pass	0.000s
All sampled durations fall within [min, max] for the category. [category='catastrophic']	`low <= d <= high`	Pass	0.000s
All sampled durations fall within [min, max] for the category. [category='extreme']	`low <= d <= high`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
All sampled durations fall within [min, max] for the category. [category='minimal']	`low <= d <= high`	Pass	0.000s
All sampled durations fall within [min, max] for the category. [category='moderate']	`low <= d <= high`	Pass	0.000s
All sampled durations fall within [min, max] for the category. [category='severe']	`low <= d <= high`	Pass	0.000s
Doublet gaps should fall in medium range (24-72h).	`all(24 <= g <= 72 for g in gaps)`	Pass	0.000s
Isolated returns zero	`sample_gap(SequenceType.ISOLATED) == 0.0`	Pass	0.000s
Line 49: rng=None → creates np.random.RandomState() internally.	`low <= g <= high`	Pass	0.000s
Persistent gaps should fall in short range (6-36h).	`all(6 <= g <= 36 for g in gaps)`	Pass	0.000s
Reproducibility	`g1 == g2`	Pass	0.000s
Within bounds[sequencetype.cluster] [seq_type=¡SequenceType.CLUSTER: 'cluster'¿]	`low <= g <= high`	Pass	0.000s
Within bounds[sequencetype.doublet] [seq_type=¡SequenceType.DOUBLET: 'doublet'¿]	`low <= g <= high`	Pass	0.000s
Within bounds[sequencetype.persistent] [seq_type=¡SequenceType.PERSISTENT: 'persistent'¿]	`low <= g <= high`	Pass	0.000s
Create basic	`storm.storm_id == "STORM-abc12345"`; `storm.duration_hours == 24.0`	Pass	0.000s
From dict round trip	`restored.storm_id == original.storm_id`; `restored.duration_hours == original.duration_hours` (+6 more)	Pass	0.000s
Precipitation rounding	`d["precipitation_mm"] == 35.12`; `d["intensity_factor"] == 1.234568` (+1 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
To dict	d["storm_id"] == "STORM-test1234"; d["start_time_hours"] == 48.0 (+6 more)	Pass	0.000s
Antecedent defaults	seq.antecedent_soil_moisture == "normal"; seq.antecedent_groundwater == "normal"	Pass	0.000s
Create doublet	seq.num_storms == 2; len(seq.storms) == 2 (+2 more)	Pass	0.000s
From dict round trip	restored.storm_id == original.storm_id; restored.duration_hours == original.duration_hours (+6 more)	Pass	0.000s
Isolated no gaps	len(seq.inter_storm_gaps_hours) == 0	Pass	0.000s
Spatial correlation defaults	seq.spatial_correlation_enabled is False; seq.spatial_correlation_range_km is None	Pass	0.000s
Spatial correlation enabled	d["spatial_correlation_enabled"] is True; d["spatial_correlation_range_km"] == 43.2	Pass	0.000s
To dict	d["storm_id"] == "STORM-test1234"; d["start_time_hours"] == 48.0 (+6 more)	Pass	0.000s
Duration mismatch	any("total_duration_hours" in e for e in errors)	Pass	0.000s
Exceeds precip window	any("MAX_PRECIP_END_HOUR" in e for e in errors)	Pass	0.000s
Invalid intensity category	any("invalid intensity" in e for e in errors)	Pass	0.000s
Invalid sequence type	any("Invalid sequence_type" in e for e in errors)	Pass	0.000s
Overlapping storms	any("starts at" in e and "before storm" in e for e in err...	Pass	0.000s
Storm count mismatch	any("num_storms" in e for e in errors)	Pass	0.000s
Too many storms	any("not in [1, 5]" in e for e in errors)	Pass	0.000s
Valid doublet	errors == []	Pass	0.000s

13 Spatial Interpolation (MKM-SP-001)

Table 24: Test Results Summary — Spatial Interpolation (MKM-SP-001)

Total Tests	41
Passed	41
Failed	0
Skipped	0
Duration	0.01s

Table 25: Detailed Test Results — Spatial Interpolation

Test Description	Acceptance Criteria	Result	Time
Diagonal is one	—	Pass	0.000s
Off diagonal between zero and one	`np.all(off_diag >= 0) and np.all(off_diag <= 1)`	Pass	0.000s
Shape is n by n	`mat.shape == (n, n)`	Pass	0.000s
Symmetric	—	Pass	0.000s
Diagonal is one	—	Pass	0.000s
Higher base correlation increases off diagonal	`np.all(off_high > off_low)`	Pass	0.000s
Larger correlation_distance → slower exponential decay → higher off-diag.	`np.all(off_far >= off_near)`	Pass	0.000s
Off diagonal positive	`np.all(off > 0)`	Pass	0.000s
Returns square matrix	`mat.shape == (3, 3)`	Pass	0.000s
Symmetric	—	Pass	0.000s
Empty dataframe returns none	`build_spatial_index(df) is None`	Pass	0.000s
None gauge data returns none	`build_spatial_index(None) is None`	Pass	0.000s
With gauges returns kdtree	`isinstance(result, cKDTree)`	Pass	0.000s
Lines 43-45: coordinates are converted to radians.	`tree.data[0][1] == pytest.approx(np.pi, rel=1e-6)`	Pass	0.000s
Single gauge should still produce a valid KDTree.	`isinstance(tree, cKDTree); tree.n == 1`	Pass	0.000s
Lines 42-46: non-empty DataFrame builds a cKDTree.	`isinstance(tree, cKDTree); tree.n == 3`	Pass	0.000s
Equatorial degree approx 111km	`111_000 < d < 112_000`	Pass	0.000s
London paris approx 340km	`330_000 < d < 350_000`	Pass	0.000s
North south degree approx 111km	`110_000 < d < 112_000`	Pass	0.000s
Returns float	`isinstance(haversine_distance(51.0, -0.1, 52.0, 0.0), float)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Same point is zero	`haversine_distance(51.5, -0.1, 51.5, -0.1) == pytest.appr...`	Pass	0.001s
Symmetry	`d1 == pytest.approx(d2, rel=1e-9)`	Pass	0.000s
All zero weights returns zero	`result == pytest.approx(7.5)`	Pass	0.000s
Closer gauge has higher weight	`result < 5.0`	Pass	0.000s
Empty list returns zero	`idw_interpolate_from_gauges([]) == 0.0`	Pass	0.000s
Equal distance gauges averages wse	`result == pytest.approx(3.0)`	Pass	0.000s
Near gauge returns wse directly	`result == pytest.approx(5.0)`	Pass	0.000s
Power one changes result	`r2 != pytest.approx(r1, rel=0.01)`	Pass	0.000s
Single far gauge	`result == pytest.approx(3.0)`	Pass	0.000s
Line 158: power parameter affects weights.	`r3 < r1`	Pass	0.000s
Line 149-150: empty input → 0.0.	`idw_interpolate_from_gauges([]) == 0.0`	Pass	0.000s
Gauge at exactly min_distance=1.0 is NOT less than, so IDW used.	`result == pytest.approx(5.0)`	Pass	0.000s
Multiple gauges produce weighted result.	`2.0 < result < 6.0`; `result < 4.0`	Pass	0.000s
Line 156-157: dist ¡ min_distance → return wse.	`result == pytest.approx(42.0)`	Pass	0.000s
At gauge location returns wse directly	`result == pytest.approx(3.0)`	Pass	0.000s
Empty gauge data returns zero	`result == 0.0`	Pass	0.000s
Idw returns weighted value	`2.5 <= result <= 4.5`	Pass	0.000s
Single gauge	`result > 0`	Pass	0.000s
Lines 119-120: distance ¡ 1m → direct return.	`result == pytest.approx(7.0)`	Pass	0.000s
Line 97-98: no gauge data → 0.0.	`spatial_interpolate_wse(51.5, -0.1, {}, {}) == 0.0`	Pass	0.000s
Lines 113-124: IDW with two equidistant gauges.	`result == pytest.approx(2.0, rel=0.01)`	Pass	0.000s

14 Storm Sequence Generator (MKM-SS-001)

Table 26: Test Results Summary — Storm Sequence Generator (MKM-SS-001)

Total Tests	17
Passed	17
Failed	0
Skipped	0
Duration	0.00s

Table 27: Detailed Test Results — Storm Sequence Generator

Test Description	Acceptance Criteria	Result	Time
Duration must not exceed MAX_PRECIP_END_HOUR (156).	`result.storms[0].duration_hours <= MAX_PRECIP_END_HOUR`	Pass	0.000s
Isolated has one storm	`isinstance(result, StormSequence)`; `result.num_storms == 1` (+2 more)	Pass	0.000s
Aggregate stats	`result.max_intensity_factor >= result.avg_intensity_factor`; `result.total_precipitation_mm > 0` (+1 more)	Pass	0.000s
Catchment id in storms	`result.storms[0].catchment_id == "rhine"`	Pass	0.000s
Cluster sequence	`result.sequence_type in (SequenceType.CLUSTER.value, Sequ...`	Pass	0.000s
force_sequence_type='doublet' → always doublet.	`result.sequence_type == SequenceType.DOUBLET.value`; `result.num_storms == 2`	Pass	0.000s
force_sequence_type='isolated' → always isolated.	`result.sequence_type == SequenceType.ISOLATED.value`; `result.num_storms == 1`	Pass	0.000s
Minimal intensity (5% sequence probability) mostly produces isolated.	`isolated >= 10`	Pass	0.001s
Generate multiple sequences without error.	`isinstance(result, StormSequence)`; `result.num_storms >= 1`	Pass	0.000s
Persistent sequence	`result.sequence_type in (SequenceType.PERSISTENT.value, S...`	Pass	0.000s
Returns storm sequence	`isinstance(result, StormSequence)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
All sequences must have total_duration ¡= 168h.	`result.total_duration_hours <= 168.0`	Pass	0.000s
Storms have correct fields	`storm.duration_hours > 0`; `storm.intensity_factor > 0` (+2 more)	Pass	0.000s
Custom catchment	`gen.catchment_id == "rhine"`; `gen.base_precipitation == CATCHMENT_BASE_PRECIP["rhine"]`	Pass	0.000s
Default catchment	`gen.catchment_id == "thames"`; `gen.base_precipitation == CATCHMENT_BASE_PRECIP["thames"]`	Pass	0.000s
Same seed → same sequence.	`seq1.num_storms == seq2.num_storms`; `seq1.sequence_type == seq2.sequence_type`	Pass	0.000s
Unknown catchment defaults to 35	`gen.base_precipitation == 35.0`	Pass	0.000s

15 Stress Test Pipeline (MKM-ST-001)

Table 28: Test Results Summary — Stress Test Pipeline (MKM-ST-001)

Total Tests	107
Passed	107
Failed	0
Skipped	0
Duration	20.75s

Table 29: Detailed Test Results — Stress Test Pipeline

Test Description	Acceptance Criteria	Result	Time
All already trained	`result["trained"] == 0`; `result["skipped"] == 3`	Pass	0.002s
After the first gauge trains, ETA should appear in output.	`"ETA" in captured`	Pass	0.220s
Missing gauge json	—	Pass	0.000s
Missing sequences	—	Pass	0.001s
No valid gauges	`result["trained"] == 0`	Pass	0.001s
One gauge already trained, one succeeds, one fails.	`result["trained"] == 1`; `result["failed"] == 1` (+1 more)	Pass	0.156s
Successful training	`result["trained"] == 3`; `result["failed"] == 0` (+2 more)	Pass	0.229s
Training failure is caught	`result["trained"] == 0`; `result["failed"] == 3` (+1 more)	Pass	0.243s
Counts default to zero	`s["alert_sequences"] == 0`; `s["warning_sequences"] == 0` (+1 more)	Pass	0.001s
Num gauges	`s["num_gauges"] == 4`	Pass	0.001s
Num sequences	`s["num_sequences"] == 7`	Pass	0.001s
Returns required keys	`key in s`	Pass	0.001s
When one gauge's classifier fails, pipeline continues.	`"classifier failed" in out`; `result["num_gauges"] == 2`	Pass	0.467s
Dict format	`len(result) == 2`	Pass	0.000s
Empty list	`_extract_gauges({"flood_gauges": []}) == []`	Pass	0.000s
List format	`result == [_G1, _G2]`	Pass	0.000s
Missing key returns empty	`_extract_gauges({}) == []`	Pass	0.000s
When gaugehd/ has baseline data, the 'Loaded' message appears.	`"gauge baselines from gaugehd" in out or "heuristic base ...`	Pass	0.479s

Test Description	Acceptance Criteria	Result	Time
Gauge summary dir written	`sg_dir.is_dir()`	Pass	0.000s
Gauge summary gauge ids	`len(d["gauge_ids"]) == 3`	Pass	0.000s
Gauge summary num gauges	`d["num_gauges"] == 3`	Pass	0.000s
Gauge summary schema version	`meta.exists(); d["schema_version"] == SCHEMA_VERSION_SPATIAL`	Pass	0.001s
Gauge summary sequence count	`len(gauge_files) == 3`	Pass	0.000s
Sequences file loadable	`len(seqs) == 40`	Pass	0.001s
Sequences json written	`(gauge_dir / SEQUENCES_FILENAME).exists()`	Pass	0.000s
Summary json written	`(gauge_dir / SUMMARY_FILENAME).exists()`	Pass	0.000s
Alert count non negative	`full_run["alert_sequences"] >= 0`	Pass	0.000s
Elapsed positive	`full_run["elapsed_seconds"] > 0`	Pass	0.000s
Num gauges	`full_run["num_gauges"] == 3`	Pass	0.000s
Num sequences	`full_run["num_sequences"] == 40`	Pass	0.000s
Returns dict	`isinstance(full_run, dict)`	Pass	0.000s
Severe lte warning	`full_run["severe_sequences"] <= full_run["warning_sequenc...`	Pass	0.000s
Type counts present	`"type_counts" in full_run; sum(full_run["type_counts"].values()) == 40`	Pass	0.000s
Warning lte alert	`full_run["warning_sequences"] <= full_run["alert_sequences"]`	Pass	0.000s
Batch generator populates category	`all(s.intensity_category in valid for s in seqs)`	Pass	0.001s
From dict missing category defaults empty	`reconstructed.intensity_category == ""`	Pass	0.000s
From dict round trip	`reconstructed.intensity_category == s.intensity_category`	Pass	0.001s
Serialized sequences have category	`all(s.intensity_category in valid for s in seqs)`	Pass	0.001s
To dict includes intensity category	`"intensity_category" in d; d["intensity_category"] == s.intensity_category`	Pass	0.001s
Legacy sequence_gauge_summary.json is deleted after split-mode write.	`not legacy.exists()`	Pass	0.469s
Corrupt file skipped	`len(baselines) == 1; "GAUGE-OK" in baselines`	Pass	0.002s
Empty directory	`_load_gaugehd_baselines(d) == {}`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Missing directory	`_load_gaugehd_baselines(tmp_path / "nonexistent") == {}`	Pass	0.000s
Missing mean level skipped	`len(baselines) == 0`	Pass	0.002s
Multiple gauges	`len(baselines) == 3`	Pass	0.003s
Valid gaugehd files	`"GAUGE-001" in baselines; baselines["GAUGE-001"]["mean_level"] == 1.25` (+1 more)	Pass	0.001s
Sequence with malformed sequence_start doesn't crash.	`result["num_sequences"] == 5`	Pass	0.464s
Explicitly test malformed date by patching sequences post-generation.	`result["num_sequences"] == 3`	Pass	0.468s
No crash when gauge json absent	`isinstance(result, dict)`	Pass	0.002s
Num gauges zero without gauge json	`result["num_gauges"] == 0`	Pass	0.001s
Sequences still written without gauge json	`(tmp_path / SEQUENCES_FILENAME).exists()`	Pass	0.002s
gauge.json present but every record missing gauge_id → n_all == 0.	`isinstance(result, dict); result["num_gauges"] == 0` (+1 more)	Pass	0.002s
Sequences still written when all gauges invalid	`(tmp_path / SEQUENCES_FILENAME).exists()`	Pass	0.002s
count=2001 with verbose=False prints ETA line at i=2000.	`result["num_sequences"] == 2001; "2,000" in out or "ETA" in out`	Pass	2.596s
Base level derived from alert	`r["base_level"] == pytest.approx(3.5 * 0.35)`	Pass	0.000s
Flood alert	`_parse_gauge(_G1)["flood_alert"] == pytest.approx(3.5)`	Pass	0.000s
Flood warning	`_parse_gauge(_G1)["flood_warning"] == pytest.approx(4.6)`	Pass	0.000s
Gauge id	`_parse_gauge(_G1)["gauge_id"] == "GAUGE-test001"`	Pass	0.000s
Lat lon	`r["lat"] == pytest.approx(51.46); r["lon"] == pytest.approx(-0.30)`	Pass	0.000s
Missing alert returns none	`_parse_gauge(bad) is None`	Pass	0.000s
Missing gauge id returns none	`_parse_gauge(bad) is None`	Pass	0.000s
Missing lat returns none	`_parse_gauge(bad) is None`	Pass	0.000s
Severe warning	`_parse_gauge(_G1)["severe_warning"] == pytest.approx(5.5)`	Pass	0.000s
Valid record returns dict	`result is not None`	Pass	0.000s
Warning defaults when absent	`r is not None; r["flood_warning"] == pytest.approx(4.0 * 1.33)` (+1 more)	Pass	0.000s
Baseline without monthly means	`result["base_level"] == 1.5; result["monthly_means"] is None`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
With baseline	`result["base_level"] == 1.25; result["monthly_means"] == {"01": 1.4}`	Pass	0.000s
Without baseline uses heuristic	`result["base_level"] == pytest.approx(3.5 * 0.35); result["monthly_means"] is None`	Pass	0.000s
Empty monthly means	`_seasonal_base_level({}, mean_level=1.0, month=6) == 1.0`	Pass	0.000s
Missing month falls back	`_seasonal_base_level(monthly, mean_level=1.0, month=3) ==...`	Pass	0.000s
No monthly means	`_seasonal_base_level(None, mean_level=1.0, month=6) == 1.0`	Pass	0.000s
With monthly means	`_seasonal_base_level(monthly, mean_level=1.0, month=1) ==...; _seasonal_base_level(monthly, mean_level=1.0, month=7) ==...`	Pass	0.000s
All have intensity category	`all(r["intensity_category"] in valid for r in records)`	Pass	0.000s
All have sequence id	`all("sequence_id" in r for r in records)`	Pass	0.000s
All have sequence type	`all(r["sequence_type"] in valid for r in records)`	Pass	0.000s
All have total precip	`all("total_precip_mm" in r for r in records); all(r["total_precip_mm"] > 0 for r in records)`	Pass	0.000s
Gauge file count matches num gauges	`len(gauge_records) == 3`	Pass	0.000s
Peak hours in valid range	`0 <= s["peak_hour"] <= 167`	Pass	0.000s
Peaks positive	`all(s["peak_m"] > 0 for s in seqs)`	Pass	0.000s
Per gauge records have alert flags	`all("alert" in s for s in seqs)`	Pass	0.000s
Per gauge records have peak	`all("peak_m" in s for s in seqs)`	Pass	0.000s
Per gauge records have peak hour	`all("peak_hour" in s for s in seqs)`	Pass	0.000s
Per gauge sequence count	`len(seqs) == len(records)`	Pass	0.000s
Severe implies warning	`s["warning"]`	Pass	0.000s
Every sequence that breached warning must also have breached alert.	`s["alert"]`	Pass	0.000s
Error message lists available ids	`"GAUGE-test001" in str(exc_info.value)`	Pass	0.001s
Invalid gauge id raises	—	Pass	0.001s
Named file gauge ids	`d["gauge_ids"] == ["GAUGE-test002"]`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Named file peaks length one	`all(len(r["peaks_m"]) == 1 for r in d["sequences"])`	Pass	0.000s
Named file schema version	`d["schema_version"] == SCHEMA_VERSION_SPATIAL`	Pass	0.003s
Named file sequence count	`len(d["sequences"]) == 40`	Pass	0.001s
Named output file written	`(gauge_dir / "sequence_gauge_GAUGE-test002.json").exists()`	Pass	0.000s
Summary num gauges is one	`single_run["num_gauges"] == 1`	Pass	0.000s
Single gauge classifier result dict	`isinstance(result, dict)`	Pass	6.168s
Single gauge + train_classifier=True covers lines 353-361.	`result["num_gauges"] == 1; stressm_dir.exists()`	Pass	6.086s
Base level matches gauge params	`responses["GAUGE-A"][0]["base_level_m"] == pytest.approx(...; responses["GAUGE-B"][0]["base_level_m"] == pytest.approx(...`	Pass	0.000s
Gauge a values	`r["base_level_m"] == pytest.approx(1.5); r["peak_level_m"] == pytest.approx(5.0)` (+1 more)	Pass	0.000s
Gauge b values	`r["base_level_m"] == pytest.approx(2.0); r["peak_level_m"] == pytest.approx(7.0)` (+1 more)	Pass	0.000s
Level change is peak minus base	`r["level_change_m"] == pytest.approx(expected)`	Pass	0.000s
Response has base level m	`"base_level_m" in r`	Pass	0.000s
Response has level change m	`"level_change_m" in r`	Pass	0.000s
Creates new summary	`summary["num_gauges"] == 1; summary["num_trained"] == 1` (+1 more)	Pass	0.003s
Merges into existing	`summary["num_gauges"] == 2; summary["num_trained"] == 2` (+1 more)	Pass	0.002s
Replaces existing gauge	`summary["num_gauges"] == 1; summary["gauges"][0]["metrics"]["auc_roc"] == 0.95`	Pass	0.002s
Skipped gauge not counted as trained	`summary["num_trained"] == 0; summary["num_skipped"] == 1`	Pass	0.001s
With count ¡ 2000 and verbose=False, no mid-loop progress line.	`"ETA" not in out or "[" not in out`	Pass	1.076s

count=501 with verbose=True triggers the progress line at i=500.	`result["num_sequences"] == 501; "[500" in out or "500/" in out`	Pass	1.577s

16 Trading Desk (MKM-TD-001)

Table 30: Test Results Summary — Trading Desk (MKM-TD-001)

Total Tests	212
Passed	212
Failed	0
Skipped	0
Duration	1.31s

Table 31: Detailed Test Results — Trading Desk

Test Description	Acceptance Criteria	Result	Time
All buttons must use font-size:11px;font-weight:600 matching header tabs.	`count >= 4`	Pass	0.000s
All buttons must use padding:4px 14px matching header tab buttons.	`count >= 4`	Pass	0.000s
All four buttons must sit inside a single flex container (group).	`'display:flex;border:1px solid #ddd;border-radius:4px;ove...`	Pass	0.000s
Commit button must be blue (#1976d2) — primary action like active header tab.	`'#1976d2' in snippet; 'color:white' in snippet`	Pass	0.000s
History button must be grey (#f5f5f5) — inactive like unselected header tabs.	`'#f5f5f5' in snippet; '#1976d2' not in snippet`	Pass	0.000s
Individual buttons must NOT have border-radius — container provides the rounding.	`'border-radius' not in snippet`	Pass	0.000s
PL Hist button must be grey (#f5f5f5) — inactive like unselected header tabs.	`'#f5f5f5' in snippet; '#1976d2' not in snippet`	Pass	0.000s
Reset button must be grey (#f5f5f5) — inactive.	`'#f5f5f5' in snippet`	Pass	0.000s
Basis delta should be positive.	`result['basis_delta_bps'] > 0; result['basis_dv01_gbp'] > 0`	Pass	0.000s
With constant basis, basis delta ~ gauge delta.	`abs(basis_delta['basis_delta_bps'] - ...`	Pass	0.000s
Hits /api/v1/trading/blotter.	`'/api/v1/trading/blotter' in blotter_js`	Pass	0.000s
Close-out hits /trading/close/.	`'/trading/close/' in blotter_js`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Button must call tdCommitMarket().	'tdCommitMarket()' in market_js	Pass	0.000s
td-commit-btn ID present in output.	'td-commit-btn' in market_js	Pass	0.000s
Commit button must NOT have display:none in its own style.	'display:none' not in own_style	Pass	0.000s
tdCommitMarket function must exist.	'tdCommitMarket' in market_js	Pass	0.000s
Hits /trading/hazard-term-structure/commit.	'/trading/hazard-term-structure/commit' in market_js	Pass	0.000s
Hits /trading/yield-curve/commit.	'/trading/yield-curve/commit' in market_js	Pass	0.000s
Commit notification shows P&L impact.	'total_pnl_impact' in market_js; 'gross_pnl_impact' in market_js	Pass	0.000s
After commit, switches to blotter tab.	"switchTab('blotter')" in market_js	Pass	0.000s
Higher spread means higher premium	pvs_high["premium_leg_pv"] > pvs_low["premium_leg_pv"]	Pass	0.000s
Positive hazard rate	pvs["premium_leg_pv"] > 0; pvs["protection_leg_pv"] > 0 (+2 more)	Pass	0.000s
Zero hazard rate	pvs["premium_leg_pv"] == 0; pvs["protection_leg_pv"] == 0	Pass	0.000s
Closed trades excluded	pnl_engine.compute_daily_pnl(trades)["num_open_trades"] == 0	Pass	0.000s
Daily pnl without previous eod	result["num_open_trades"] == 2; result["total_notional"] == 15_000_000 (+1 more)	Pass	0.000s
Pnl decomposition sums	abs(pos["daily_pnl"] - (pos["new_trade_pnl"] + pos["marke...	Pass	0.000s
Running pnl calculation	abs(pos1["running_pnl"] - 43_500) < 1; abs(pos2["running_pnl"] - (-7_125)) < 1	Pass	0.000s
Test risk grid construction.	len(grid['buckets']) == 3; len(grid['gauges']) == 2 (+3 more)	Pass	0.000s
Receiver trade should have negative gauge DV01.	enriched['gauge_fs01'] < 0; enriched['is_payer'] is False	Pass	0.002s
Test trade enrichment with sample data.	enriched['swap_id'] == 'PRS-TEST001'; enriched['gauge_id'] == 'GAUGE-TEST-001' (+4 more)	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Payer and receiver DV01 should be equal magnitude, opposite sign.	`abs(payer_enriched['gauge_fs01'] + rcv_enriched['gauge_fs...`	Pass	0.005s
tdHazardDirtyKeys tracks per-gauge hazard changes.	`'tdHazardDirtyKeys' in market_js`	Pass	0.000s
Shows UNCOMMITTED label when dirty.	`'UNCOMMITTED' in market_js`	Pass	0.000s
tdYieldDirty variable tracks yield curve changes.	`'tdYieldDirty' in market_js`	Pass	0.000s
Also fetches blotter for live summary.	`'/api/v1/trading/blotter' in eod_js`	Pass	0.000s
Fetches /api/v1/trading/eod/history.	`'/api/v1/trading/eod/history' in eod_js`	Pass	0.000s
Fetches /api/v1/trading/pnl-series.	`'/api/v1/trading/pnl-series' in eod_js`	Pass	0.000s
Hits /api/v1/trading/eod for submit.	`'/api/v1/trading/eod' in eod_js`	Pass	0.000s
get_eod_history returns 500 when engine raises (lines 88-90).	`resp.status_code == 500`	Pass	0.001s
get_eod_pdf returns 500 on unexpected error (lines 107-109).	`resp.status_code == 500`	Pass	0.001s
submit_eod returns 500 when engine raises (lines 70-72).	`resp.status_code == 500; data['status'] == 'error'`	Pass	0.002s
renderEodAttribution function exists.	`'renderEodAttribution' in eod_js`	Pass	0.000s
renderEodChart function exists.	`'renderEodChart' in eod_js`	Pass	0.000s
tdCleanupEodCharts exists for chart cleanup.	`'tdCleanupEodCharts' in eod_js`	Pass	0.000s
tdDownloadEodPdf function exists.	`'tdDownloadEodPdf' in eod_js`	Pass	0.000s
tdSubmitEod function exists.	`'tdSubmitEod' in eod_js`	Pass	0.000s
Returns list of snapshots after submission.	`len(data['history']) == 1; entry['eod_id'] == 'EOD-20260228'` (+3 more)	Pass	0.040s
Returns empty list when no snapshots exist.	`resp.status_code == 200; data['status'] == 'success'` (+2 more)	Pass	0.001s
Can submit multiple EODs on different dates.	`data['count'] == 2`	Pass	0.066s
POST with empty JSON body uses today's date (no date key).	`resp.status_code == 200; data['status'] == 'success'`	Pass	0.040s

Test Description	Acceptance Criteria	Result	Time
GET for existing PDF date serves the file.	resp.status_code == 200; resp.content_type == 'application/pdf'	Pass	0.002s
GET for non-existent date returns 404.	resp.status_code == 404	Pass	0.000s
ImportError in PDF gen is caught; EOD snapshot still saved (lines 56-57).	resp.status_code == 200; data['status'] == 'success'	Pass	0.029s
Exception in PDF gen is caught; EOD snapshot still saved (lines 58-59).	resp.status_code == 200; data['status'] == 'success'	Pass	0.042s
Attribution section container present.	'td-eod-attribution' in eod_js	Pass	0.000s
P&L summary cards container present.	'td-eod-cards' in eod_js	Pass	0.000s
Date picker for EOD date.	'td-eod-date' in eod_js	Pass	0.000s
EOD History section present.	'td-eod-history-wrap' in eod_js; 'EOD History' in eod_js	Pass	0.000s
Submit EOD button present.	'td-eod-submit-btn' in eod_js; 'EOD Submit' in eod_js	Pass	0.000s
All gauges from base should appear.	'GAUGE-001' in rates; 'GAUGE-002' in rates	Pass	0.001s
Adjusted gauges should be flagged.	rates['GAUGE-001']['is_adjusted'] is True; rates['GAUGE-002']['is_adjusted'] is False	Pass	0.004s
Adjusted rates should override base, non-adjusted should use base.	rates['GAUGE-001']['annual_hazard_rate_warnings'] == 0.04; rates['GAUGE-001']['annual_hazard_rate_alert'] == 0.04 (+1 more)	Pass	0.002s
Eod history	len(history) == 2; history[0]["date"] == "2026-02-21"	Pass	0.003s
Generate snapshot	snapshot["eod_id"] == "EOD-20260224"; snapshot["date"] == "2026-02-24" (+3 more)	Pass	0.000s
Idempotent snapshot	len(pnl_engine.get_eod_history()) == 1	Pass	0.003s
Pnl series	len(series) == 2; series[0]["date"] == "2026-02-20" (+1 more)	Pass	0.003s
Snapshot persists	snapshot is not None; snapshot["eod_id"] == "EOD-20260224"	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
tdClearFilters function exists.	'tdClearFilters' in blotter_js	Pass	0.000s
tdApplyFilter exposed on window for external callers.	'tdApplyFilter' in blotter_js	Pass	0.000s
tdFilterChanged function exists.	'tdFilterChanged' in blotter_js	Pass	0.000s
getFilteredTrades function exists.	'getFilteredTrades' in blotter_js	Pass	0.000s
tdRemoveFilter function exists.	'tdRemoveFilter' in blotter_js	Pass	0.000s
For small rates, 1bp bump in rate ˜= 1bp bump in spread.	0.5 < result['delta_spread_bps'] < 2.0	Pass	0.000s
DV01 scales with notional.	abs(r2['dv01_gbp'] / r1['dv01_gbp'] - 2.0) < 0.01	Pass	0.000s
DV01 should be positive for a buyer (spread increases with hazard).	result['dv01_gbp'] > 0; result['delta_spread_bps'] > 0	Pass	0.000s
At zero hazard rate, spread is at floor so delta may be minimal.	result['delta_spread_bps'] >= 0	Pass	0.000s
Balanced book	payer == rcv; len(trades) == 6	Pass	0.005s
Balanced notional	payer_not == rcv_not	Pass	0.003s
Counterparties distributed	len(ctpys) == 2	Pass	0.003s
Files saved	len(list(output_dir.glob("PRS-*.json"))) == 6	Pass	0.004s
Payer trades below fair	ps["Pricing"]["SpreadBps"] < ps["Pricing"]["FairSpreadBps"]	Pass	0.003s
Receiver trades above fair	ps["Pricing"]["SpreadBps"] > ps["Pricing"]["FairSpreadBps"]	Pass	0.004s
Reproducible with seed	t1["PhysicalSwap"]["Pricing"]["SpreadBps"] ...	Pass	0.004s
Small mtm	abs(ps["Pricing"]["NPV"]) < ps["LegData"]["Notional"] * 0...	Pass	0.003s
Valid cdm structure	ps["Header"]["SwapID"].startswith("PRS-"); isinstance(ps["LegData"]["Payer"], bool) (+2 more)	Pass	0.003s
GET returns market_state with yield_curve and hazard_term_structure.	resp.status_code == 200; data['status'] == 'success' (+2 more)	Pass	0.002s
Response includes per-gauge effective rates.	'gauges' in data; len(gauge_ids) >= 1	Pass	0.003s
First load creates default yield curve.	resp.status_code == 200; len(yc) > 0	Pass	0.003s
Bumping hazard curve higher should increase fair PRS spread.	spread_after > spread_before; spread_after - spread_before > 1.0	Pass	0.004s

Test Description	Acceptance Criteria	Result	Time
Bumping hazard curve should change trade MTM.	`mtm_after > mtm_before`	Pass	0.004s
P&L from a 1bp parallel shift should approximately equal FS01.	`abs(mtm_change - fs01) < abs(fs01) * 0.20`	Pass	0.003s
Full cycle: enrich → EOD snap → bump curve → re-enrich → daily P&L non-zero.	`len(eod_snap['positions']) == 1`; `new_fair != prev_fair` (+5 more)	Pass	0.005s
Shows down arrow for rate decrease.	`'\{}\{}u25bc' in blotter_js`	Pass	0.000s
Blotter compares to prev_fair_spread_bps.	`'prev_fair_spread_bps' in blotter_js`	Pass	0.000s
Shows up arrow for rate increase.	`'\{}\{}u25b2' in blotter_js`	Pass	0.000s
Verify the MTM formula.	`abs(mtm - 80_000) < 1`	Pass	0.000s
Protection seller loses when spread widens.	`mtm < 0`	Pass	0.000s
Protection buyer profits when spread widens.	`mtm > 0`	Pass	0.000s
MTM is zero when fair = trade spread.	`mtm == 0`	Pass	0.000s
Blotter has market_pnl column.	`'market_pnl' in blotter_js`	Pass	0.000s
Blotter shows Mkt P&L label.	`'Mkt P' in blotter_js`	Pass	0.000s
GET market-state returns 500 when engine raises (lines 44-46).	`resp.status_code == 500`; `data['status'] == 'error'`	Pass	0.001s
POST market-state/reset returns 500 when engine raises (lines 119-121).	`resp.status_code == 500`	Pass	0.001s
POST market-state with null JSON body → 400 (line 55).	`resp.status_code == 400`	Pass	0.000s
Base rates should match what's in gaugehc.json.	`g1['annual_hazard_rate_warning'] == 0.025`; `g1['annual_hazard_rate_severe'] == 0.01`	Pass	0.001s
First load should initialise from base curves.	`'base_rates' in state`; `'gauge_adjustments' in state` (+2 more)	Pass	0.001s
Subsequent loads should return the saved state.	`state1['base_rates'] == state2['base_rates']`	Pass	0.003s
Reset all adjustments.	`len(state['gauge_adjustments']) == 0`	Pass	0.005s
Reset a single gauge's adjustments.	`'GAUGE-001' not in state['gauge_adjustments']`; `'GAUGE-002' in state['gauge_adjustments']`	Pass	0.006s

Test Description	Acceptance Criteria	Result	Time
Effective rate should use adjustment if present.	`rate == 0.025`; `rate == 0.04`	Pass	0.004s
Gauge without adjustment should return base rate.	`rate == 0.03`	Pass	0.003s
Updating a rate should persist.	`adj['annual_hazard_rate_warning'] == 0.035`	Pass	0.003s
tdCleanupMarketCharts exists.	`'tdCleanupMarketCharts' in market_js`	Pass	0.000s
renderGaugeList function exists.	`'renderGaugeList' in market_js`	Pass	0.000s
Chart canvas element present.	`'td-market-canvas' in market_js`	Pass	0.000s
Curve mode selector with yield/hazard options.	`'td-curve-mode' in market_js`; `'Yield Curve' in market_js` (+1 more)	Pass	0.000s
commit_yield_curve saves all tenors at once.	`state['yield_curve']['1'] == 0.04`; `state['yield_curve']['3'] == 0.05`	Pass	0.001s
Tenor above maximum returns last key rate.	`rate == r6`	Pass	0.001s
Tenor below minimum returns first key rate.	`rate == r1`	Pass	0.001s
Exact tenor match from yield curve.	`isinstance(rate, float)`; `rate > 0`	Pass	0.002s
Non-integer tenor triggers interpolation.	`min(r1, r2) <= rate <= max(r1, r2)`	Pass	0.001s
Gauge entries with no gauge_id are skipped.	`'GAUGE-B' in base`; `'' not in base`	Pass	0.001s
gaugehc.json with 'gauges' key is handled.	`'GAUGE-A' in state.get('base_rates', {})`	Pass	0.001s
When gaugehc.json is missing, base curves returns empty dict.	`isinstance(state, dict)`	Pass	0.000s
gaugehc.json with unknown format returns empty base_rates.	`state.get('base_rates', {}) == {}`	Pass	0.001s
reset_yield_curve restores defaults.	`state['yield_curve']['1'] == MarketStateManager.DEFAULT_Y...`	Pass	0.004s
Notes parameter is stored in gauge adjustments.	`'adjustment_notes' in adj`; `'override' in adj['adjustment_notes']`	Pass	0.002s
New Trade button must be grey (#f5f5f5) — non-primary action.	`'#f5f5f5' in snippet`; `'#1976d2' not in snippet`	Pass	0.000s
Button text must be 'New Trade'.	`'New Trade' in market_js`	Pass	0.000s
td-newtrade-btn must exist in Market tab.	`'td-newtrade-btn' in market_js`	Pass	0.000s
Button must call tdMarketNewTrade().	`'tdMarketNewTrade' in market_js`	Pass	0.000s
tdMarketNewTrade must hide trading desk and call viewHazardCurve.	`'viewHazardCurve' in market_js`; `'tdSelectedGauge' in market_js`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
New Trade button must appear to the LEFT of History in the DOM.	`nt_idx < hist_idx`	Pass	0.000s
New Trade button must not have its own border-radius (container clips it).	`'border-radius' not in snippet`	Pass	0.000s
Market pnl nonzero with eod mark	`abs(result["positions"][0]["market_pnl"] - 80_000.0) < 1`	Pass	0.000s
Market pnl zero for missing eod trade	`abs(pos["market_pnl"]) < 0.01; abs(pos["new_trade_pnl"]) < 0.01` (+1 more)	Pass	0.000s
New trade has inception pnl	`abs(pos["new_trade_pnl"] - expected) < 1; abs(pos["market_pnl"]) < 0.01`	Pass	0.000s
Pnl decomposition with prev eod	`abs(pos["daily_pnl"] - (pos["new_trade_pnl"] + pos["marke...`	Pass	0.000s
Shows daily P&L metric.	`'dailyPnl' in blotter_js`	Pass	0.000s
Shows From Market metric.	`'From Market' in blotter_js; 'fromMarket' in blotter_js`	Pass	0.000s
Shows signed net notional.	`'netNotional' in blotter_js; 'fmtGBP(netNotional)' in blotter_js`	Pass	0.000s
Shows Running P&L.	`'Running' in blotter_js`	Pass	0.000s
After EOD snap, next blotter P&L uses snap as baseline.	`resp.status_code == 200; 'daily_pnl' in t` (+1 more)	Pass	0.045s
Summary runs	`"Market-Making Book Summary" in out; "Payer (short):" in out` (+1 more)	Pass	0.003s
tdResetCurve function exists.	`'tdResetCurve' in market_js`	Pass	0.000s
Hits /trading/hazard-term-structure/reset.	`'/trading/hazard-term-structure/reset' in market_js`	Pass	0.000s
Hits /trading/yield-curve/reset.	`'/trading/yield-curve/reset' in market_js`	Pass	0.000s
Reset clears all adjustments.	`resp.status_code == 200; data['status'] == 'success'` (+1 more)	Pass	0.029s
Reset clears one gauge only.	`resp.status_code == 200; 'GAUGE-001' in data['message']`	Pass	0.002s
risk-grid returns 500 when engine raises (lines 44-46).	`resp.status_code == 500; data['status'] == 'error'`	Pass	0.001s
trade-map returns 500 when engine raises (lines 197-199).	`resp.status_code == 500`	Pass	0.001s
Grid should contain entries for traded gauges.	`len(grid) > 0`	Pass	0.025s

Test Description	Acceptance Criteria	Result	Time
GET returns a risk grid structure.	resp.status_code == 200; data['status'] == 'success' (+1 more)	Pass	0.025s
Returns empty grid when no trades exist.	resp.status_code == 200; data['status'] == 'success'	Pass	0.003s
Longer tenor means more payments.	a5 > a1	Pass	0.000s
Higher hazard rate means lower annuity (higher default prob).	annuity_low > annuity_high	Pass	0.001s
Annuity should be between 0 and tenor.	0 < annuity < 5	Pass	0.000s
With no default risk, annuity equals risk-free annuity.	abs(annuity - expected) < 1e-6	Pass	0.000s
is_payer field used to determine sign.	'is_payer' in blotter_js	Pass	0.000s
Payer notional should be negated.	'is_payer' in blotter_js; 'Math.abs' in blotter_js	Pass	0.000s
tdToggleSort function exists.	'tdToggleSort' in blotter_js	Pass	0.000s
tdViewTrade function for click-through to PRS pricing.	'tdViewTrade' in blotter_js	Pass	0.000s
After knock-out, cash price must stay at full notional.	hourly[2]['per_trade'][0]['triggered'] is True; hourly[2]['per_trade'][0]['cash_price'] == 1_000_000 (+2 more)	Pass	0.000s
Alert trade KOs before severe trade.	'ALERT' in triggered; 'SEVERE' in triggered (+1 more)	Pass	0.000s
Trade must NOT knock out if water stays below trigger.	'T1' not in triggered	Pass	0.000s
Receiver (short protection) gets negative cash on KO.	hourly[1]['per_trade'][0]['cash_price'] == -1_000_000	Pass	0.000s
Trade must knock out when water ¿= trigger level.	'T1' in triggered; triggered['T1'] == 3	Pass	0.000s
Same-day snap overwrites previous.	len(eod_files) == 1	Pass	0.059s
POST creates EOD snapshot.	resp.status_code == 200; data['status'] == 'success' (+1 more)	Pass	0.043s
Response includes trade count and P&L summary.	'num_open_trades' in summary; 'total_notional' in summary (+2 more)	Pass	0.043s
EOD-¡date¿.json created in eod/ directory.	snapshot_path.exists(); snapshot['eod_id'] == 'EOD-20260228' (+1 more)	Pass	0.040s

Test Description	Acceptance Criteria	Result	Time
Snapshot includes yield_curve and hazard_term_structure.	'yield_curve' in ms; 'hazard_term_structure' in ms	Pass	0.042s
Snapshot includes per-trade positions.	len(positions) == 16; 'swap_id' in p (+2 more)	Pass	0.038s
All levels should be ¿= base level (positive level_change).	lvl >= base - 0.01	Pass	0.000s
Custom num_hours should produce the right length.	len(levels) == 100	Pass	0.000s
Water levels should be non-decreasing before peak.	levels[i] >= levels[i - 1] - 0.01	Pass	0.000s
Peak should reach base + level_change.	peak >= base + change * 0.9; peak <= base + change + 0.1	Pass	0.000s
Water level should decrease after peak.	levels[-1] < levels[peak_idx]	Pass	0.000s
Default hydrograph must return 168 hourly values (STORM_HOURS = 7-day window).	len(levels) == 168	Pass	0.000s
Very short storm peaks early.	peak_idx <= 10	Pass	0.000s
Hour 0 should be at base level.	abs(levels[0] - 2.0) < 0.01	Pass	0.000s
Zero duration should still produce valid levels.	len(levels) == 168; levels[0] > 2.0	Pass	0.000s
gauge_fs01 column present.	'gauge_fs01' in blotter_js	Pass	0.000s
fmtMaturity function exists.	'fmtMaturity' in blotter_js	Pass	0.000s
Files saved	len(list(output_dir.glob("PRS-*.json"))) == 50	Pass	0.016s
Has calendar spreads	calendar_spread_gauges >= 5	Pass	0.015s
Reproducible with seed	t1["PhysicalSwap"]["Pricing"]["SpreadBps"] ...	Pass	0.025s
Trade count	len(trades) == 50	Pass	0.014s
Uses central gauges	gauge_ids.issubset(expected_ids)	Pass	0.016s
Valid cdm structure	ps["Header"]["SwapID"].startswith("PRS-"); ps["Pricing"]["RiskyAnnuity"] > 0 (+1 more)	Pass	0.013s
Westminster is net short	westminster_notional < 0	Pass	0.013s
Gauge positions include FS01 broken down by tenor.	'fs01_by_tenor' in g; isinstance(g['fs01_by_tenor'], dict)	Pass	0.025s
Each gauge position has lat, lon, gauge_name.	'lat' in g; 'lon' in g (+2 more)	Pass	0.025s
Each gauge has net_fs01.	'net_fs01' in g	Pass	0.026s

Test Description	Acceptance Criteria	Result	Time
Each gauge has total and net notional.	'total_notional' in g; 'net_notional' in g (+1 more)	Pass	0.024s
Gauge positions include daily P&L.	'daily_pnl' in g; 'running_pnl' in g	Pass	0.026s
Gauge positions show number of trades.	gauges_by_id['GAUGE-001']['num_trades'] == 2; gauges_by_id['GAUGE-002']['num_trades'] == 1	Pass	0.024s
GET returns gauge positions.	resp.status_code == 200; data['status'] == 'success' (+1 more)	Pass	0.025s
Returns empty positions when no trades exist.	resp.status_code == 200; len(data['gauges']) == 0	Pass	0.006s
Gauge in gauge.json but not gaugehc.json gets coordinates (lines 83-85).	resp.status_code == 200	Pass	0.055s
Property locations loaded from property.json (lines 97-101).	resp.status_code == 200; 'gauges' in data	Pass	0.057s
Closed trades are excluded from trade map gauge aggregation (line 115).	resp.status_code == 200; 'GAUGE-9042bd95' not in gauge_ids	Pass	0.040s
Close trade	mark["trade_status"] == "Closed"; mark["close_spread_bps"] == 260.0 (+1 more)	Pass	0.003s
Load empty marks	pnl_engine.load_trade_marks() == {}	Pass	0.000s
Update and load mark	pnl_engine.load_trade_marks()["PRS-001"]["trade_status"] ...	Pass	0.001s
Market making spreads are integers	spread == round(spread)	Pass	0.003s
Thames central spreads are integers	spread == round(spread)	Pass	0.013s
Updated state persists across requests.	resp.status_code == 200; data['market_state']['num_adjusted'] >= 1	Pass	0.026s
POST updates gauge rate successfully.	resp.status_code == 200; data['status'] == 'success' (+1 more)	Pass	0.025s
Missing gauge returns 400.	resp.status_code == 400	Pass	0.000s
No JSON body returns error (500 — Flask BadRequest caught).	resp.status_code in (400, 500)	Pass	0.001s
Update returns affected_trades and total_pnl_impact.	'affected_trades' in data; 'total_pnl_impact' in data	Pass	0.026s

17 CDM Schema Validation (CDM-ALL)

Table 32: Test Results Summary — CDM Schema Validation (CDM-ALL)

Total Tests	33
Passed	33
Failed	0
Skipped	0
Duration	0.00s

Table 33: Detailed Test Results — CDM Schema Validation

Test Description	Acceptance Criteria	Result	Time
All cdm fields present	`gauge_mapping_summary.fields_missing == 0`	Pass	0.000s
All cdm fields present	`mortgage_mapping_summary.fields_missing == 0`	Pass	0.000s
All cdm fields present	`property_mapping_summary.fields_missing == 0`	Pass	0.000s
All types valid	`gauge_mapping_summary.fields_type_invalid == 0`	Pass	0.000s
All types valid	`mortgage_mapping_summary.fields_type_invalid == 0`	Pass	0.000s
All types valid	`property_mapping_summary.fields_type_invalid == 0`	Pass	0.000s
All values valid	`gauge_mapping_summary.fields_value_invalid == 0`	Pass	0.000s
All values valid	`mortgage_mapping_summary.fields_value_invalid == 0`	Pass	0.000s
All values valid	`property_mapping_summary.fields_value_invalid == 0`	Pass	0.000s
Get required fields returns list[gauge cdm] [cdm_name='gauge_cdm']	`isinstance(cdm.get_required_fields(), list)`	Pass	0.000s
Get required fields returns list[mortgage cdm] [cdm_name='mortgage_cdm']	`isinstance(cdm.get_required_fields(), list)`	Pass	0.000s
Get required fields returns list[property cdm] [cdm_name='property_cdm']	`isinstance(cdm.get_required_fields(), list)`	Pass	0.000s
Get required fields returns list[prs cdm] [cdm_name='prs_cdm']	`isinstance(cdm.get_required_fields(), list)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Get required fields returns list[storm cdm] [cdm_name='storm_cdm']	`isinstance(cdm.get_required_fields(), list)`	Pass	0.000s
Get required fields returns list[stormts cdm] [cdm_name='stormts_cdm']	`isinstance(cdm.get_required_fields(), list)`	Pass	0.000s
List all fields returns paths[gauge cdm] [cdm_name='gauge_cdm']	`isinstance(fields, list); len(fields) > 0` (+2 more)	Pass	0.000s
List all fields returns paths[mortgage cdm] [cdm_name='mortgage_cdm']	`isinstance(fields, list); len(fields) > 0` (+2 more)	Pass	0.000s
List all fields returns paths[property cdm] [cdm_name='property_cdm']	`isinstance(fields, list); len(fields) > 0` (+2 more)	Pass	0.000s
List all fields returns paths[prs cdm] [cdm_name='prs_cdm']	`isinstance(fields, list); len(fields) > 0` (+2 more)	Pass	0.000s
List all fields returns paths[storm cdm] [cdm_name='storm_cdm']	`isinstance(fields, list); len(fields) > 0` (+2 more)	Pass	0.000s
List all fields returns paths[stormts cdm] [cdm_name='stormts_cdm']	`isinstance(fields, list); len(fields) > 0` (+2 more)	Pass	0.000s
Repr returns string[gauge cdm] [cdm_name='gauge_cdm']	`isinstance(repr(cdm), str)`	Pass	0.000s
Repr returns string[mortgage cdm] [cdm_name='mortgage_cdm']	`isinstance(repr(cdm), str)`	Pass	0.000s
Repr returns string[property cdm] [cdm_name='property_cdm']	`isinstance(repr(cdm), str)`	Pass	0.000s
Repr returns string[prs cdm] [cdm_name='prs_cdm']	`isinstance(repr(cdm), str)`	Pass	0.000s
Repr returns string[storm cdm] [cdm_name='storm_cdm']	`isinstance(repr(cdm), str)`	Pass	0.000s
Repr returns string[stormts cdm] [cdm_name='stormts_cdm']	`isinstance(repr(cdm), str)`	Pass	0.000s
Schema returns non empty dict[gauge cdm] [cdm_name='gauge_cdm']	`isinstance(schema, dict); len(schema) > 0`	Pass	0.000s
Schema returns non empty dict[mortgage cdm] [cdm_name='mortgage_cdm']	`isinstance(schema, dict); len(schema) > 0`	Pass	0.000s
Schema returns non empty dict[property cdm] [cdm_name='property_cdm']	`isinstance(schema, dict); len(schema) > 0`	Pass	0.000s
Schema returns non empty dict[prs cdm] [cdm_name='prs_cdm']	`isinstance(schema, dict); len(schema) > 0`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Schema returns non empty dict[storm cdm] [cdm_name='storm_cdm']	`isinstance(schema, dict); len(schema) > 0`	Pass	0.000s
Schema returns non empty dict[stormts cdm] [cdm_name='stormts_cdm']	`isinstance(schema, dict); len(schema) > 0`	Pass	0.000s

18 End-to-End Browser Tests (E2E-ALL)

Table 34: Test Results Summary — End-to-End Browser Tests (E2E-ALL)

Total Tests	229
Passed	197
Failed	12
Skipped	20
Duration	1814.40s

Table 35: Detailed Test Results — End-to-End Browser Tests

Test Description	Acceptance Criteria	Result	Time
01 open blotter and count trades[chromium]	—	Pass	9.093s
01 open eod tab count history[chromium]	—	Pass	9.100s
01 open gauge prs shows hazard curve[chromium]	—	Pass	11.038s
01 open market tab record yield[chromium]	—	Pass	9.096s
01 open prs and verify controls[chromium]	—	Pass	9.150s
01 storm panel has prs tab[chromium]	—	Pass	4.029s
02 change yield curve value[chromium]	—	Pass	13.628s
02 click close button opens modal[chromium]	—	Pass	12.110s
02 gauge with trade shows active blotter button[chromium]	—	Pass	11.049s
02 prs tab opens property hazard panel[chromium]	—	Pass	9.076s
02 select counterparty[chromium]	—	Pass	10.647s
02 submit eod[chromium]	—	Pass	10.620s
03 blotter button opens td filtered to gauge[chromium]	—	Pass	14.102s
03 commit market changes[chromium]	—	Pass	22.669s
03 enter closeout spread and confirm[chromium]	—	Pass	28.680s

Test Description	Acceptance Criteria	Result	Time
03 set trade parameters[chromium]	—	FAIL	309.251s
03 storm flood history click opens timeline[chromium]	—	Skip	7.064s
03 verify new snapshot in history[chromium]	—	Pass	9.106s
04 blotter button muted for gauge without trades[chromium]	—	Skip	0.011s
04 commit trade[chromium]	—	Pass	19.924s
04 verify closed trade in blotter[chromium]	—	Pass	9.086s
04 verify pnl after curve change[chromium]	—	Pass	9.191s
05 new trade appears in blotter[chromium]	—	Pass	9.074s
Add action[chromium]	—	Pass	0.007s
Add agenda item[chromium]	—	Pass	0.010s
Add button opens form[chromium]	—	Pass	3.127s
Add decision[chromium]	—	Pass	0.009s
Add meeting button exists[chromium]	—	Skip	1.553s
Add participant[chromium]	—	Pass	0.007s
Audit reports api returns files[chromium]	—	Pass	0.012s
Auto selects worst case storm[chromium]	—	Pass	15.085s
Back button returns to list[chromium]	—	Pass	4.688s
Bibtex export button[chromium]	—	Pass	3.103s
Blotter api returns trades[chromium]	—	Pass	0.092s
Blotter button click opens trading desk[chromium]	—	Pass	14.127s
Blotter button enabled for gauge with trades[chromium]	—	Pass	11.077s
Blotter button exists in tab bar[chromium]	—	Pass	11.072s
Blotter button muted for gauge without trades[chromium]	—	Pass	11.091s
Blotter has close button[chromium]	—	Pass	9.108s
Blotter link exists[chromium]	—	Pass	0.019s

Test Description	Acceptance Criteria	Result	Time
Blotter link opens trading desk[chromium]	—	Pass	3.072s
Cells interactive[chromium]	—	Pass	2.511s
Chart canvas exists[chromium]	—	Pass	9.101s
Chart canvas rendered[chromium]	—	Pass	9.105s
Chart rendered[chromium]	—	Pass	9.111s
Chart rendered[chromium]	—	Pass	9.103s
Charts render[chromium]	—	Skip	15.071s
Click away closes menu[chromium]	—	Pass	3.096s
Close button opens dialog[chromium]	—	Pass	12.128s
Close button opens modal[chromium]	—	Skip	9.252s
Commit button exists[chromium]	—	Pass	9.103s
Commit button state[chromium]	—	Skip	3.289s
Commit triggers response[chromium]	—	**FAIL**	25.856s
Confirm closeout posts[chromium]	—	Skip	12.114s
Content area exists[chromium]	—	Pass	3.086s
Content loads with timeseries data[chromium]	—	Pass	12.238s
Content mentions audit[chromium]	—	Pass	3.074s
Content mentions audit keywords[chromium]	—	Pass	7.563s
Content mentions bcbs[chromium]	—	Pass	3.076s
Content mentions bibliography[chromium]	—	Pass	3.078s
Content mentions eod keywords[chromium]	—	Pass	4.610s
Content mentions field keywords[chromium]	—	Pass	7.569s
Content mentions historical context[chromium]	—	Pass	12.189s
Content mentions lineage keywords[chromium]	—	Pass	7.575s
Content mentions model[chromium]	—	Pass	3.092s
Content mentions mrc[chromium]	—	Pass	3.080s
Content mentions raci[chromium]	—	Pass	3.086s
Content mentions risk keywords[chromium]	—	Pass	4.612s

Test Description	Acceptance Criteria	Result	Time
Content mentions stress concepts[chromium]	—	Pass	15.087s
Content shows probability data[chromium]	—	Pass	9.093s
Content shows return period data[chromium]	—	Pass	9.097s
Context menu has header[chromium]	—	Pass	1.574s
Context menu has items[chromium]	—	Pass	1.560s
Controls exist[chromium]	—	Pass	9.087s
Counterparty dropdown has options[chromium]	—	Pass	9.104s
Counterparty dropdown has options[chromium]	—	Pass	9.168s
Counterparty options show rating[chromium]	—	Pass	9.102s
Counterparty summary fields[chromium]	—	Pass	0.008s
Counterparty types diverse[chromium]	—	Pass	0.006s
Curves view visible[chromium]	—	Pass	4.619s
Delete action[chromium]	—	Pass	0.008s
Delete agenda item[chromium]	—	Pass	0.009s
Delete decision[chromium]	—	Pass	0.008s
Delete participant[chromium]	—	Pass	0.008s
Direction dropdown options[chromium]	—	Pass	9.098s
Dirty state indicator[chromium]	—	Pass	9.167s
Distribution and history show same flood count[chromium]	—	Skip	8.130s
Document list renders[chromium]	—	Pass	7.575s
Download buttons clickable[chromium]	—	Skip	7.574s
Edit buttons on principles[chromium]	—	Pass	14.136s
Edit opens modal[chromium]	—	Skip	3.120s
Empty counterparty disables commit[chromium]	—	Skip	15.160s
Eod history has pdf link[chromium]	—	Pass	6.138s
Eod submit button exists[chromium]	—	Skip	6.076s

Test Description	Acceptance Criteria	Result	Time
Eod submit creates snapshot[chromium]	—	Pass	15.116s
Eod view visible[chromium]	—	Pass	4.631s
Escape closes gauge panel[chromium]	—	Pass	3.101s
Escape closes governance panel[chromium]	—	Pass	4.577s
Escape closes trading desk[chromium]	—	Pass	3.061s
Field lineage api returns data[chromium]	—	Pass	0.012s
Field search api[chromium]	—	Pass	0.007s
Fill notional and spread[chromium]	—	Pass	12.316s
Fill prs values[chromium]	—	Pass	6.176s
Fs01 cell click filters blotter[chromium]	—	Skip	3.083s
Fs01 tab shows risk grid[chromium]	—	Pass	3.130s
Gauge blotter menu item opens trading desk[chromium]	—	Pass	20.613s
Gauge panel blotter link opens trading desk[chromium]	—	Pass	12.128s
Gauge panel loads data not error[chromium]	—	Pass	9.034s
Gauge select exists[chromium]	—	Pass	5.024s
Gauge select has options[chromium]	—	Pass	10.029s
Get counterparty detail[chromium]	—	Pass	0.010s
Has artefact files[chromium]	—	Pass	7.584s
Has chart or curve content[chromium]	—	Pass	4.613s
Has charts or inputs[chromium]	—	Pass	4.599s
Has content area[chromium]	—	Pass	1.542s
Has gauge dropdown[chromium]	—	Pass	4.626s
Has hazard term structure[chromium]	—	Pass	4.626s
Has history list or table[chromium]	—	Pass	4.610s
Has leaflet or svg[chromium]	—	Pass	4.620s
Has pipeline steps[chromium]	—	Pass	7.588s
Has report cards or search[chromium]	—	Pass	7.651s
Has risk grid[chromium]	—	Pass	4.601s
Has staleness info[chromium]	—	Pass	7.584s

Test Description	Acceptance Criteria	Result	Time
Has storm dropdown[chromium]	—	Pass	4.623s
Has tab buttons[chromium]	—	Pass	1.577s
Has summary or file cards[chromium]	—	Pass	7.581s
Has yield curve content[chromium]	—	Pass	4.641s
Hazard api returns success[chromium]	—	Pass	0.048s
Hazard curve item opens gauge panel[chromium]	—	Pass	10.676s
Hazard curve tab has real data[chromium]	—	Pass	12.099s
Hazard curve tab renders[chromium]	—	Pass	9.115s
Hazard term structure api reachable[chromium]	—	**FAIL**	6.056s
Header flood count matches history[chromium]	—	Pass	6.084s
Historical tab has content[chromium]	—	Pass	12.053s
Historical tab renders[chromium]	—	Pass	12.275s
Layer control exists[chromium]	—	Pass	0.014s
Layer toggle changes markers[chromium]	—	Pass	7.565s
Lineage api returns data[chromium]	—	Pass	0.020s
List counterparties returns data[chromium]	—	Pass	0.032s
List meetings returns data[chromium]	—	Pass	0.010s
Map view visible[chromium]	—	Pass	4.613s
Market state api returns data[chromium]	—	Pass	0.026s
Market view visible[chromium]	—	Pass	17.265s
Matrix grid renders[chromium]	—	Pass	1.579s
Meeting detail has tabs[chromium]	—	Pass	5.602s
Meeting detail returns full object[chromium]	—	Pass	0.009s
Meeting detail shows metadata[chromium]	—	Pass	5.606s
Meeting list has rows[chromium]	—	Pass	3.561s
Meeting list renders[chromium]	—	Pass	1.583s
Meeting row clickable[chromium]	—	Pass	5.613s

Test Description	Acceptance Criteria	Result	Time
Meeting summary has fields[chromium]	—	Pass	0.006s
Mkt hazard ts has current gauge[chromium]	—	**FAIL**	6.040s
Mkt hazard ts populated[chromium]	—	**FAIL**	6.037s
Modal has confirm cancel[chromium]	—	Skip	3.099s
Model row opens detail[chromium]	—	Pass	3.174s
Nav container exists[chromium]	—	Pass	5.023s
Nav gauge select opens panel[chromium]	—	**FAIL**	13.603s
Nav property select opens panel[chromium]	—	**FAIL**	13.622s
Negative closeout spread handling[chromium]	—	Skip	9.125s
Negative rate handling[chromium]	—	Pass	10.681s
Negative spread handling[chromium]	—	Pass	15.157s
No error notifications after clearing[chromium]	—	Pass	9.020s
Non numeric tenor input[chromium]	—	Skip	6.094s
Nonexistent counterparty returns 404[chromium]	—	Pass	0.006s
Nonexistent gauge does not crash page[chromium]	—	Pass	3.034s
Nonexistent gauge shows error[chromium]	—	Pass	6.045s
Nonexistent property does not crash page[chromium]	—	Pass	3.052s
Nonexistent property shows error[chromium]	—	Pass	6.044s
Open gauge panel via js[chromium]	—	Pass	0.041s
Panel has tab bar[chromium]	—	Pass	0.016s
Panel is visible[chromium]	—	Pass	1.553s
Panel title shows gauge name[chromium]	—	Pass	0.022s
Portfolio storms api returns data[chromium]	—	Pass	0.037s
Property panel loads without errors[chromium]	—	Pass	9.044s
Property select exists[chromium]	—	Pass	5.024s

Test Description	Acceptance Criteria	Result	Time
Property select has options[chromium]	—	Pass	10.027s
Property storm api returns data[chromium]	—	Pass	0.024s
Propertyts summary api returns data[chromium]	—	Pass	0.015s
Prs controls exist[chromium]	—	Pass	3.110s
Prs has counterparty dropdown[chromium]	—	Pass	9.087s
Prs hazard chart has data[chromium]	—	**FAIL**	6.041s
Prs hazard chart values are reasonable[chromium]	—	**FAIL**	6.176s
Prs tab renders[chromium]	—	Pass	9.084s
Pv premium chart has data on open[chromium]	—	**FAIL**	6.044s
Reference table sortable[chromium]	—	Pass	1.583s
Reset button exists[chromium]	—	Pass	6.064s
Reset restores defaults[chromium]	—	Pass	9.189s
Right click gauge shows context menu[chromium]	—	Pass	3.052s
Right click gauge shows menu[chromium]	—	Pass	15.640s
Risk grid api returns data[chromium]	—	Pass	0.045s
Risk view visible[chromium]	—	Pass	4.609s
Save button commits changes[chromium]	—	Pass	10.681s
Selecting counterparty enables commit[chromium]	—	Pass	12.251s
Selecting counterparty updates pricing[chromium]	—	Pass	10.629s
Selecting gauge shows action buttons[chromium]	—	Pass	10.553s
Shows field counts[chromium]	—	Pass	7.582s
Spread input updates settlement[chromium]	—	Skip	12.122s
Staleness api returns data[chromium]	—	Pass	0.012s
Stats bar shows data[chromium]	—	Pass	9.105s
Storm dropdown exists[chromium]	—	Pass	15.063s

Test Description	Acceptance Criteria	Result	Time
Storm scenario click activates stress tab[chromium]	—	Skip	12.168s
Storm scenarios list exists[chromium]	—	Pass	12.202s
Stress storms api returns data[chromium]	—	Pass	0.124s
Stress tab renders[chromium]	—	Pass	12.115s
Stress view visible[chromium]	—	Pass	4.620s
Tab content loads[chromium]	—	Pass	7.578s
Tab content loads[chromium]	—	Pass	7.563s
Tab content loads[chromium]	—	Pass	7.571s
Tab content loads[chromium]	—	Pass	3.089s
Tab content loads[chromium]	—	Pass	3.078s
Tab content loads[chromium]	—	Pass	3.086s
Tab content loads[chromium]	—	Pass	3.075s
Tab content loads[chromium]	—	Pass	3.086s
Tab content loads[chromium]	—	Pass	3.074s
Tab content loads[chromium]	—	Pass	3.086s
Tab title updates[chromium]	—	Pass	9.100s
Tenor inputs accept values[chromium]	—	Pass	7.761s
Summary api[chromium]	—	Pass	0.007s
Trace gauge returns steps[chromium]	—	Pass	0.023s
Trace missing params returns 400[chromium]	—	Pass	0.004s
Trace nonexistent returns empty[chromium]	—	Pass	0.017s
Trace property returns steps[chromium]	—	Pass	0.011s
Trading desk blotter visible after gauge link[chromium]	—	Pass	13.605s
Trigger change updates display[chromium]	—	**FAIL**	9.100s
Trigger dropdown options[chromium]	—	**FAIL**	9.107s
Upload area exists[chromium]	—	Pass	7.569s
Worst storms are subset of history[chromium]	—	Skip	8.109s
Yield curve inputs exist[chromium]	—	Pass	6.178s

Test Description	Acceptance Criteria	Result	Time
Zero notional disables commit[chromium]	—	Skip	15.161s

Failed Test Details

test_nav_gauge_select_opens_panel[chromium] File: tests/e2e/test_cross_panel_prs_blotter/TestNav

```
AssertionError: No panel opened after nav menu action
assert False
```

test_nav_property_select_opens_panel[chromium] File: tests/e2e/test_cross_panel_prs_blotter/Test

```
AssertionError: No panel opened after nav menu action
assert False
```

test_trigger_dropdown_options[chromium] File: tests/e2e/test_gauge_detail_prs/TestPRSPricingTab

```
playwright._impl._errors.Error: Page.evaluate: TypeError: undefined is not iterable (cannot
    read property Symbol(Symbol.iterator))
    at Array.from (<anonymous>)
    at eval (eval at evaluate (:290:30), <anonymous>:4:26)
    at UtilityScript.evaluate (<anonymous>:297:18)
    at UtilityScript.<anonymous> (<anonymous>:1:44)
```

test_trigger_change_updates_display[chromium] File: tests/e2e/test_gauge_detail_prs/TestPRSPrici

```
playwright._impl._errors.Error: Page.evaluate: TypeError: undefined is not iterable (cannot
    read property Symbol(Symbol.iterator))
    at Array.from (<anonymous>)
    at eval (eval at evaluate (:290:30), <anonymous>:4:26)
    at UtilityScript.evaluate (<anonymous>:297:18)
    at UtilityScript.<anonymous> (<anonymous>:1:44)
```

test_hazard_term_structure_api_reachable[chromium] File: tests/e2e/test_gauge_prs_market_curve/T

```
failed on setup with "AssertionError: Data load errors after opening gauge panel: ['Failed to
    load hazard curve data', 'Failed to load hazard curve data']"
```

test_mkt_hazard_ts_populated[chromium] File: tests/e2e/test_gauge_prs_market_curve/TestGaugePRS

```
failed on setup with "AssertionError: Data load errors after opening gauge panel: ['Failed to
    load hazard curve data', 'Failed to load hazard curve data']"
```

test_mkt_hazard_ts_has_current_gauge[chromium] File: tests/e2e/test_gauge_prs_market_curve/Test

```
failed on setup with "AssertionError: Data load errors after opening gauge panel: ['Failed to
    load hazard curve data', 'Failed to load hazard curve data']"
```

test_prs_hazard_chart_has_data[chromium] File: tests/e2e/test_gauge_prs_market_curve/TestGaugePR

```
failed on setup with "AssertionError: Data load errors after opening gauge panel: ['Failed to
    load hazard curve data', 'Failed to load hazard curve data']"
```

test_pv_premium_chart_has_data_on_open[chromium] File: `tests/e2e/test_gauge_prs_market_curve/T`

```
failed on setup with "AssertionError: Data load errors after opening gauge panel: ['Failed to
    load hazard curve data', 'Failed to load hazard curve data']"
```

test_prs_hazard_chart_values_are_reasonable[chromium] File: `tests/e2e/test_gauge_prs_market_curve`

```
failed on setup with "AssertionError: Data load errors after opening gauge panel: ['Failed to
    load hazard curve data', 'Failed to load hazard curve data']"
```

test_03_set_trade_parameters[chromium] File: `tests/e2e/test_lifecycle_gauge_prs/TestGaugePRSBool`

```
playwright._impl._errors.TimeoutError: Locator.select_option: Timeout 300000ms exceeded.
Call log:
  - waiting for locator("#prs-trigger")
    - locator resolved to <input type="hidden" value="severe" id="prs-trigger"/>
  - attempting select option action
    2    waiting for element to be visible and enabled
      - element is not visible
    - retrying select option action
    - waiting 20ms
    2    waiting for element to be visible and enabled
      - element is not visible
    - retrying select option action
      - waiting 100ms
    595    waiting for element to be visible and enabled
        - element is not visible
      - retrying select option action
        - waiting 500ms
```

test_commit_triggers_response[chromium] File: `tests/e2e/test_workflow_gauge_prs/TestGaugePRSCom`

```
AssertionError: No feedback after commit     button text: 'commit', page excerpt: 'gauges (52)
  thames barking creek
  thames barnes bridge
  thames battersea bridge
  thames blackfriars bridge
  thames blackwall
  thames blackwall tunnel
  thames cannon street
  thames caversham bridge
  thames chelsea bridge
  thames chertsey bridge
  thames chiswick bridge
  thames cookham bridge
  thames dagenham dock'
assert (False or 0 > 0 or False or False)
 +  where 0 = count()
 +    where count = <Locator frame=<Frame name= url='http://127.0.0.1:52163/visualization'>
    selector="[class*='notification'], [class*='toast'], [class*='alert'], .notif-message, [id
    *='notif']">.count
```

A Platform Infrastructure Tests

Table 36: Test Results Summary — Platform Infrastructure (PLATFORM)

Total Tests	6154
Passed	6136
Failed	0
Skipped	18
Duration	256.92s

Table 37: Detailed Test Results — Platform Infrastructure

Test Description	Acceptance Criteria	Result	Time
Test that config has CATCHMENT attribute.	—	Pass	0.000s
Test that config ENDPOINTS includes health_check for backend handler.	`cfg.ENDPOINTS['health_check'] == '/api/v1/health'`; `ep in cfg.ENDPOINTS`	Pass	0.000s
Test that select_catchment.html exists.	—	Pass	0.000s
Test that routes register correctly.	—	Pass	0.044s
Existing synthetic fixture – confirms endpoint logic works.	`response.status_code == 200`	Pass	0.002s
Mounts the real model_inventory.json and confirms the endpoint does not 500.	`response.status_code == 200`; `data["status"] == "success"` (+1 more)	Pass	0.061s
All models in the file are served by the endpoint.	`response.status_code == 200`; `len(data["models"]) == expected_count`	Pass	0.046s
Concrete subclass returns empty list	`result == []`	Pass	0.000s
Get entity id called	`loader.find_by_id('A') is not None`	Pass	0.001s
Get entity summary called	`summaries[0]['id'] == 'B'`	Pass	0.002s
Add action auto generates id	`r.status_code == 200`; `actions[0]["id"] == "A-001"` (+1 more)	Pass	0.001s
Add action unknown meeting returns 404	`r.status_code == 404`	Pass	0.001s
Delete action	`r.status_code == 200`; `r.get_json()["meeting"]["actions"] == []`	Pass	0.003s
Delete action unknown meeting returns 404	`r.status_code == 404`	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Update action status	`r.status_code == 200; r.get_json()["meeting"]["actions"][0]["status"] == "Closed"`	Pass	0.003s
Update missing action returns 404	`r.status_code == 404`	Pass	0.002s
Add action auto id	`"A-001" in ids; "A-002" in ids`	Pass	0.002s
Add action missing meeting returns 404	`r.status_code == 404`	Pass	0.002s
Add action returns 200	`r.status_code == 200; len(meeting["actions"]) == 2`	Pass	0.002s
Delete action missing meeting returns 404	`r.status_code == 404`	Pass	0.002s
Delete action returns 200	`r.status_code == 200; len(r.get_json()["meeting"]["actions"]) == 0`	Pass	0.002s
Update action not found returns 404	`r.status_code == 404`	Pass	0.002s
Update action returns 200	`r.status_code == 200; action["status"] == "Closed"`	Pass	0.002s
Westminster, Chelsea and Lambeth all have trades – all must appear.	`self.WESTMINSTER_ID in active; self.CHELSEA_ID in active` (+1 more)	Pass	0.021s
No trades at all -¿ empty gauge_ids list -¿ all right-click	`resp.status_code == 200; data['gauge_ids'] == []`	Pass	0.001s
A gauge that has no PRS trades must not appear in active-gauges,	`self.NO_TRADE_ID not in data['gauge_ids']`	Pass	0.021s
Lambeth Bridge (GAUGE-9042bd95) must be in active-gauges	`resp.status_code == 200; data['status'] == 'success'` (+1 more)	Pass	0.021s
After closing PRS-TEST-LAMBETH, Lambeth must leave active-gauges	`self.LAMBETH_ID in before['gauge_ids']; self.LAMBETH_ID not in after['gauge_ids']`	Pass	0.036s
Closing Lambeth's trade must not affect Westminster or Chelsea.	`self.WESTMINSTER_ID in after['gauge_ids']; self.CHELSEA_ID in after['gauge_ids']`	Pass	0.030s
gauge_ids are returned in sorted order for stable JS dict lookup.	`ids == sorted(ids)`	Pass	0.022s
Gauge in active list	`active['status'] == 'success'; gauge_id in active.get('gauge_ids', [])`	Pass	0.008s
Production function: levels decay back toward base after peak.	`levels[-1] < max(levels)`	Pass	0.000s
Production function: peak ¿= base + level_change.	`max(levels) >= base + change * 0.9`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
_synthesize_hydrograph default returns 168 hours (STORM_HOURS constant).	`STORM_HOURS == 168`; `len(levels) == 168`	Pass	0.000s
Production function: hour 0 = base_level.	`abs(levels[0] - 3.0) < 0.01`	Pass	0.000s
Peak occurs near duration_hours * peak_position.	`abs(actual_peak_hour - expected_peak_hour) <= 2`	Pass	0.000s
Add appears in list	`len(r.get_json()["references"]) == 1`	Pass	0.004s
Add assigns id	`"id" in ref`; `len(ref["id"]) > 0`	Pass	0.002s
Add no json body uses defaults	`r.status_code in (200, 400, 500)`	Pass	0.002s
Add returns 200	`r.status_code == 200`	Pass	0.002s
Add returns reference	`data["status"] == "success"`; `"reference" in data`	Pass	0.002s
Add same category twice no duplicate	`r.get_json()["categories"].count("Risk") == 1`	Pass	0.004s
Add stores all fields	`ref["author"] == "Jones"`; `ref["journal"] == "Nature"` (+1 more)	Pass	0.002s
Add with category updates categories	`"Flood Risk" in r.get_json()["categories"]`	Pass	0.003s
Adds to map	—	Pass	0.002s
Returns notification system	`isinstance(result, NotificationSystem)`	Pass	0.001s
Add to map calls add child	—	Pass	0.002s
Add to map calls add child	—	Pass	0.003s
Add to map calls add child	—	Pass	0.001s
Approaching retirement over 55	`isinstance(result, list)`	Pass	0.000s
Early career under 25	`isinstance(result, list)`	Pass	0.000s
Prime years 25 to 55	`isinstance(result, list)`	Pass	0.000s
Add agenda item	`r.status_code == 200`; `len(agenda) == 1` (+2 more)	Pass	0.002s
Add agenda unknown meeting returns 404	`r.status_code == 404`	Pass	0.001s
Add multiple items increments number	`[a["item"] for a in agenda] == [1, 2, 3]`	Pass	0.005s
Delete agenda item renumbers	`len(agenda) == 2`; `[a["item"] for a in agenda] == [1, 2]` (+1 more)	Pass	0.006s
Delete agenda unknown meeting returns 404	`r.status_code == 404`	Pass	0.002s
Update agenda item	`r.status_code == 200`; `item["title"] == "Updated"` (+1 more)	Pass	0.003s

Test Description	Acceptance Criteria	Result	Time
Update agenda unknown meeting returns 404	`r.status_code == 404`	Pass	0.001s
Update missing item returns 404	`r.status_code == 404`	Pass	0.001s
Add agenda item increments count	`len(meeting["agenda"]) == 3`	Pass	0.004s
Add agenda item returns 200	`r.status_code == 200`; `r.get_json()["status"] == "success"`	Pass	0.002s
Add agenda missing meeting returns 404	`r.status_code == 404`	Pass	0.002s
Delete agenda item returns 200	`r.status_code == 200`; `len(meeting["agenda"]) == 1` (+1 more)	Pass	0.003s
Delete agenda missing meeting returns 404	`r.status_code == 404`	Pass	0.002s
Update agenda item not found returns 404	`r.status_code == 404`	Pass	0.002s
Update agenda item returns 200	`r.status_code == 200`; `meeting["agenda"][0]["title"] == "Updated Title"`	Pass	0.002s
When peak_wl ¿= alert but ¡ warning, threshold is 'alert'.	`resp.status_code == 200`; `g001 is not None` (+1 more)	Pass	0.055s
Empty report	`result['num_clones'] == 0`; `result['worst'] == []` (+1 more)	Pass	0.000s
Fmt stats has python	`'python' in result['fmt_stats']`	Pass	0.000s
Line numbers extracted from 'start' field must be integers.	`isinstance(entry[1], int)`; `isinstance(entry[3], int)`	Pass	0.000s
Regression: jscpd uses 'start' not 'startLine' — must not produce '?' in output.	`l1 != '?'`; `l2 != '?'`	Pass	0.000s
Largest sorted descending	`lines == sorted(lines, reverse=True)`	Pass	0.000s
Num clones correct	`result['num_clones'] == 2`	Pass	0.000s
Returns dict	`isinstance(result, dict)`	Pass	0.000s
Total present	`'total' in result`; `result['total']['lines'] == 10000`	Pass	0.000s
Worst populated	`len(result['worst']) > 0`; `isinstance(fname, str)` (+2 more)	Pass	0.000s
computePRSCashflows function exists.	`'computePRSCashflows' in prs_js`	Pass	0.000s
interpolateSurvival function exists.	`'interpolateSurvival' in prs_js`	Pass	0.000s
interpolateYieldRate function exists.	`'interpolateYieldRate' in prs_js`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Completeness percentage computed	`r["PropertyHeader"]["percentage"] == pytest.approx(66.67,...`	Pass	0.000s
Empty section excluded	`"PropertyHeader" not in r`	Pass	0.000s
Mortgage data included	`"Mortgage Data" in r`	Pass	0.000s
Property header section present	`"PropertyHeader" in r`	Pass	0.000s
Returns dict	`isinstance(r, dict); isinstance(r, dict)`	Pass	0.000s
Section not present excluded	`len(r) == 0`	Pass	0.000s
All none returns zeros	`result["counts"]["properties"] == 0; result["counts"]["mortgages"] == 0` (+1 more)	Pass	0.000s
Counts match data	`result["counts"]["properties"] == 2; result["counts"]["mortgages"] == 1` (+1 more)	Pass	0.000s
Flood mortgages counted	`result["counts"]["flood_mortgages"] == 1`	Pass	0.000s
Flood properties overlap	`result["counts"]["flood_properties"] == 2; result["overlaps"]["properties_with_flood_risk"] == 1`	Pass	0.000s
Returns required keys	`"counts" in result; "overlaps" in result`	Pass	0.000s
pts dir exists but property has no flood_events → 404.	`client.get(_COMPOSITE_URL).status_code == 404`	Pass	0.031s
pts dir exists but all flood_depths are zero → 404.	`client.get(_COMPOSITE_URL).status_code == 404`	Pass	0.034s
No pts dir returns 404	`pts_client_no_data.get(_COMPOSITE_URL).status_code == 404`	Pass	0.000s
No pts dir status error	`pts_client_no_data.get(_COMPOSITE_URL).get_json()["status"] ==`	Pass	0.000s
Options returns ok	`pts_client_no_data.options(_COMPOSITE_URL).status_code ==...`	Pass	0.000s
Frame has gauges	`"gauges" in frame0`	Pass	0.000s
Frame has hour	`frame0["hour"] == 0`	Pass	0.000s
Frame has properties	`"properties" in frame0`	Pass	0.000s
Frame has stats	`"gauges_flooded" in stats; "properties_flooded" in stats` (+1 more)	Pass	0.000s
Arrived true at arrival time	`frames[4]["properties"][0]["arrived"] is False; frames[5]["properties"][0]["arrived"] is True`	Pass	0.005s

Test Description	Acceptance Criteria	Result	Time
Peak true at and after peak time	`frames[11]["properties"][0]["peak"] is False; frames[12]["properties"][0]["peak"] is True`	Pass	0.006s
hour ¿= len(readings) → flooded=False, depth_m=0, arrived=False, peak=False.	`p["flooded"] is False; p["depth_m"] == 0` (+2 more)	Pass	0.034s
Property does not have wse m	`"wse_m" not in frames[0]["properties"][0]`	Pass	0.000s
Property has peak field	`"peak" in frames[0]["properties"][0]`	Pass	0.000s
Property has required fields	`f in p`	Pass	0.000s
Frames list length	`len(pts_env["client"].get(_COMPOSITE_URL).get_json()["frames...`	Pass	0.006s
N frames equals storm hours	`pts_env["client"].get(_COMPOSITE_URL).get_json()["n_frames...`	Pass	0.006s
N properties affected is one	`pts_env["client"].get(_COMPOSITE_URL).get_json()["n_properties...`	Pass	0.006s
Returns 200	`pts_env["client"].get(_COMPOSITE_URL).status_code == 200`	Pass	0.006s
Status success	`pts_env["client"].get(_COMPOSITE_URL).get_json()["status"]...`	Pass	0.006s
Storm id is composite	`pts_env["client"].get(_COMPOSITE_URL).get_json()["storm_id"]...`	Pass	0.006s
Property with two storms: composite uses the one with higher flood_depth.	`data["status"] == "success"; data["frames"][0]["properties"][0]["depth_m"] == pytest.a...`	Pass	0.034s
No pts dir returns 404	`r.status_code == 404`	Pass	0.000s
No pts dir status error	`pts_client_no_data.get( f"/api/v1/propertyts/...`	Pass	0.000s
Options returns ok	`r.status_code == 200`	Pass	0.000s
Unknown storm message mentions storm id	`"STORM-GHOST" in msg`	Pass	0.031s
Unknown storm returns 404	`client.get("/api/v1/propertyts/animate/STORM-GHOST").status...`	Pass	0.033s
Unknown storm status error	`client.get( "/api/v1/propertyts/animate/STORM...`	Pass	0.031s
Frame has gauges	`"gauges" in frame0; len(frame0["gauges"]) == 1`	Pass	0.000s
Frame has hour field	`"hour" in frame0`	Pass	0.000s
Frame has properties	`"properties" in frame0; len(frame0["properties"]) == 1`	Pass	0.000s
Frame has stats	`"gauges_flooded" in stats; "properties_flooded" in stats` (+1 more)	Pass	0.000s
Frame hour zero is first	`frame0["hour"] == 0`	Pass	0.000s
Hour values sequential	`[f["hour"] for f in frames] == list(range(STORM_HOURS))`	Pass	0.038s

Test Description	Acceptance Criteria	Result	Time
Gauge has required fields	`f in gauge0`	Pass	0.000s
Gauge id correct	`gauge0["gauge_id"] == "GAUGE-001"`	Pass	0.000s
Gauge status alert when above alert	`gauge0["status"] == "alert"`	Pass	0.000s
Gauge status normal when below alert	`g["status"] == "normal"`	Pass	0.035s
Gauge status severe	`g["status"] == "severe"`	Pass	0.035s
Gauge status warning	`g["status"] == "warning"`	Pass	0.038s
Gauge readings using water_level_m key (not waterLevel) still work.	`g["water_level_m"] == 4.3`	Pass	0.037s
Gauge with no readings defaults to zero	`frames[10]["gauges"][0]["water_level_m"] == 0`	Pass	0.035s
Water level from readings	`gauge0["water_level_m"] == 4.2`	Pass	0.000s
Arrived false before arrival time	`frames[0]["properties"][0]["arrived"] is False; frames[4]["properties"][0]["arrived"] is False`	Pass	0.000s
Arrived true at and after arrival time	`frames[5]["properties"][0]["arrived"] is True; frames[10]["properties"][0]["arrived"] is True`	Pass	0.000s
When hour ¿= len(readings), flooded=False, depth_m=0, wse_m=0.	`p["flooded"] is False; p["depth_m"] == 0` (+1 more)	Pass	0.040s
Property has required fields	`f in p`	Pass	0.000s
Property id correct	`frames[0]["properties"][0]["property_id"] == "PROP-001"`	Pass	0.000s
Gauges flooded counts non normal	`stats["gauges_flooded"] == 1`	Pass	0.035s
Gauges flooded zero when normal	`stats["gauges_flooded"] == 0`	Pass	0.035s
Properties flooded is positive early on	`stats["properties_flooded"] >= 1`	Pass	0.221s
Total depth m non negative	`all(f["stats"]["total_depth_m"] >= 0 for f in frames[:5])`	Pass	0.033s
Frames list length equals storm hours	`len(data["frames"]) == STORM_HOURS`	Pass	0.004s
N frames equals storm hours	`data["n_frames"] == STORM_HOURS`	Pass	0.007s
N properties affected is one	`data["n_properties_affected"] == 1`	Pass	0.006s
Returns 200	`anim_client.get(f"/api/v1/propertyts/animate/{STORM_ID}/").`	Pass	0.006s
Status success	`data["status"] == "success"`	Pass	0.005s
Storm id in response	`data["storm_id"] == STORM_ID`	Pass	0.005s
Appends line	`"hello world" in lines`	Pass	0.000s
Clear then append	`lines == ["fresh line"]`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Trims buffer at 2000	`length <= 2000`	Pass	0.001s
All non negative	`np.all(result >= 0.0)`	Pass	0.001s
For each active hour, mean(gauge_precip) approximately equals catchment_precip.	`result[:, h].mean() == pytest.approx(precip[h], rel=1e-6)`	Pass	0.001s
Output shape	`result.shape == (thames_model.n_gauges, 168)`	Pass	0.001s
Different gauges get different precipitation amounts.	`col.std() > 0.0`	Pass	0.001s
Mean across gauges of total precip approximately equals original catchment total.	`per_gauge_total.mean() == pytest.approx(catchment_total, ...`	Pass	0.001s
Zero precip hours remain zero	`np.all(result[:, zero_hours] == 0.0)`	Pass	0.001s
All factors applied	`abs(result - round(expected, 2)) < 0.01`	Pass	0.000s
Basic calculation	`result == 800_000.0`	Pass	0.000s
Clamped to max	`result == MAX_PROPERTY_VALUE`	Pass	0.000s
Clamped to min	`result == MIN_PROPERTY_VALUE`	Pass	0.000s
Flood risk discount reduces value	`with_risk < no_risk`	Pass	0.000s
Noise factor applied	`result2 > result1`	Pass	0.000s
Returns float	`isinstance(result, float)`	Pass	0.000s
Age risk approaching retirement	`"Medium" in r["Age Risk"]`	Pass	0.000s
Age risk early career	`"Medium" in r["Age Risk"]`	Pass	0.000s
Age risk prime years	`"Age Risk" in r; "Low" in r["Age Risk"]`	Pass	0.000s
Credit risk high for poor score	`"High" in r["Credit Risk"]`	Pass	0.000s
Credit risk low for high score	`"Credit Risk" in r; "Low" in r["Credit Risk"]`	Pass	0.000s
Credit risk low medium for good score	`"Low-Medium" in r["Credit Risk"]`	Pass	0.000s
Credit risk medium for fair score	`"Medium" in r["Credit Risk"]`	Pass	0.000s
Employment risk established 2 to 5	`"Low-Medium" in r["Employment Risk"]`	Pass	0.000s
Employment risk recent under 2	`"Medium-High" in r["Employment Risk"]`	Pass	0.000s
Employment risk stable 5 plus	`"Employment Risk" in r; "Low" in r["Employment Risk"]`	Pass	0.000s
Income risk low for high income	`"Income Risk" in r; "Low" in r["Income Risk"]`	Pass	0.000s
Income risk low medium adequate	`"Low-Medium" in r["Income Risk"]`	Pass	0.000s
Income risk medium limited	`"Medium" in r["Income Risk"]`	Pass	0.000s
Returns dict	`isinstance(result, dict)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Best and worst sections shown	`"Best Data Section" in r; "Weakest Data Section" in r`	Pass	0.000s
Critical completeness high	`"Critical Data Reliability" in r; "High" in r["Critical Data Reliability"]`	Pass	0.000s
Critical completeness low	`"Low" in r["Critical Data Reliability"]`	Pass	0.000s
Critical completeness medium	`"Medium" in r["Critical Data Reliability"]`	Pass	0.000s
Excellent over 90	`"Excellent" in r["Overall Completeness"]`	Pass	0.000s
Fair 60 to 75	`"Fair" in r["Overall Completeness"]`	Pass	0.000s
Good 75 to 90	`"Good" in r["Overall Completeness"]`	Pass	0.000s
Limited 40 to 60	`"Limited" in r["Overall Completeness"]`	Pass	0.000s
Poor under 40	`"Poor" in r["Overall Completeness"]`	Pass	0.000s
Building 25 years → Contemporary category.	`isinstance(result, list)`	Pass	0.001s
Building older than 100 years → Historic category.	`isinstance(result, list)`	Pass	0.001s
Building 50–100 years → Mature category.	`isinstance(result, list)`	Pass	0.001s
Building 25–50 years → Modern category.	`isinstance(result, list)`	Pass	0.001s
Basic info subsection	`any("Basic Property Information" in t for t in texts)`	Pass	0.001s
Building age subsection	`any("Building Age" in t for t in texts)`	Pass	0.001s
Condition fair recommendation	`isinstance(result, list)`	Pass	0.001s
Condition good recommendation	`isinstance(result, list)`	Pass	0.001s
Condition poor recommendation	`isinstance(result, list)`	Pass	0.001s
Condition unknown falls back	`isinstance(result, list)`	Pass	0.001s
Empty property does not crash	`isinstance(result, list)`	Pass	0.000s
External features subsection	`any("External Features" in t for t in texts)`	Pass	0.001s
Has section header	`any("Property Attributes" in t for t in texts)`	Pass	0.001s
Has tables	`any(isinstance(e, Table) for e in result)`	Pass	0.001s
No construction year skips age	`isinstance(result, list)`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Physical dimensions subsection	`any("Physical Dimensions" in t for t in texts)`	Pass	0.001s
Returns list	`isinstance(result, list); len(result) > 0`	Pass	0.001s
Room composition subsection	`any("Room Composition" in t for t in texts)`	Pass	0.001s
With mortgage data does not crash	`isinstance(result, list)`	Pass	0.001s
Line 82: audit log ¿ 10000 entries gets trimmed.	`r.status_code == 200; len(log) <= 10000`	Pass	0.062s
Line 57: inventory file empty/missing → 404.	`r.status_code == 404`	Pass	0.002s
Line 85: save failure → 500.	`r.status_code == 500`	Pass	0.003s
Audit log truncated to 10000	`r.status_code == 200; len(entries) == 10000` (+1 more)	Pass	0.057s
_auditTerminalRefreshNow must be on window – called from onclick in innerHTML.	`'window._auditTerminalRefreshNow' in audit_reports_js`	Pass	0.000s
JSON button was deliberately removed – must not reappear.	`'test_failures_report.json' not in audit_reports_js`	Pass	0.000s
Poll output endpoint	`'test-report/output' in audit_reports_js`	Pass	0.000s
Refresh audit tab button	`'_auditTerminalRefreshNow' in audit_reports_js`	Pass	0.000s
_runTestSuite must be on window – onclick attributes execute in global scope.	`'window._runTestSuite' in audit_reports_js`	Pass	0.000s
Run suite function	`'_runTestSuite' in audit_reports_js`	Pass	0.000s
Terminal overlay id	`'_audit-terminal-overlay' in audit_reports_js`	Pass	0.000s
_toggleFailureDetail must be on window – called from onclick in innerHTML.	`'window._toggleFailureDetail' in audit_reports_js`	Pass	0.000s
When audit log exceeds 10 000 entries, it keeps only the last 10 000.	`len(entries) == 10_000; entries[0].get("seq") == 1 or entries[0].get("model_id") ...`	Pass	0.056s
auto_open=True enters the auto-open block (lines 324-325).	`report_path is not None; Path(report_path).exists()`	Pass	0.026s
auto_open=False (default) — no open attempt, returns PDF path normally.	`report_path is not None; Path(report_path).exists()`	Pass	0.024s

Test Description	Acceptance Criteria	Result	Time
auto_open=True with a successful open_pdf_file call (lines 324-325).	`report_path is not None; Path(report_path).exists()`	Pass	0.026s
All pages exist	`p in gen.pages`	Pass	0.001s
All pages exist	`p in gen.pages`	Pass	0.001s
Always includes analysis pages	`ap in pages`	Pass	0.002s
Always includes appendix	`"appendix" in pages`	Pass	0.001s
Always includes gauge info pages	`gp in pages`	Pass	0.001s
Always includes operational pages	`op in pages`	Pass	0.001s
Always includes property pages	`pp in pages`	Pass	0.002s
Always includes summary pages	`sp in pages`	Pass	0.001s
Empty flood data	`isinstance(pages, list)`	Pass	0.001s
Empty property data	`isinstance(pages, list)`	Pass	0.002s
Includes analysis pages	`p in pages`	Pass	0.001s
Includes overview pages	`p in pages`	Pass	0.001s
Returns list	`isinstance(pages, list); len(pages) > 0`	Pass	0.001s
Returns list	`isinstance(pages, list); len(pages) > 0`	Pass	0.001s
With hazard curve includes analysis pages	`any(p in pages for p in analysis_pages)`	Pass	0.001s
With mortgage includes mortgage pages	`mp in pages`	Pass	0.002s
With timeseries includes analysis pages	`any(p in pages for p in analysis_pages)`	Pass	0.001s
Without mortgage excludes mortgage pages	`mp not in pages`	Pass	0.002s
Without timeseries or hazard no analysis pages	`not any(p in pages for p in analysis_pages)`	Pass	0.001s
Avail width value	`AVAIL_WIDTH == pytest.approx(7.3 * inch)`	Pass	0.000s
Lines 60-61: custom server_url and endpoints override config.	`handler.server_url == 'http://test:9999'; handler.endpoints == {'a': '/api/a'}`	Pass	0.000s
Lines 60-61: default server_url and endpoints from config.	`handler.server_url == config.SERVER_URL; handler.endpoints == config.ENDPOINTS`	Pass	0.000s
Endpoints dict is a copy, not a reference.	`'z' not in handler.endpoints`	Pass	0.000s
Line 154: BaseCDM.get_required_fields returns empty list by default.	`cdm.get_required_fields() == []`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
get_field_info on a leaf field returns its definition.	`info is not None; info["type"] == "text"`	Pass	0.000s
Line 116: field path not found returns None.	`mortgage_cdm.get_field_info("Mortgage.Nonexistent.Field")`	Pass	0.000s
Line 116: partial path that resolves to non-dict returns None.	`mortgage_cdm.get_field_info("Nonexistent") is None`	Pass	0.000s
Line 118: get_field_info returns None when leaf is not a dict.	`cdm.get_field_info("Root.Field") is None`	Pass	0.000s
Line 133: metadata keys at nested level are also skipped.	`"Root.Sub.Name" in fields; 'options' not in path_parts` (+1 more)	Pass	0.000s
list_all_fields on PRS CDM covers nested GaugeSet section.	`any("GaugeSet" in f for f in fields); any("Triggers" in f for f in fields)`	Pass	0.000s
Line 133: list_all_fields recurses nested sections.	`isinstance(fields, list); len(fields) > 10` (+2 more)	Pass	0.000s
Line 133: 'type', 'options', 'values', 'description', 'items' are skipped.	`len(fields) == 1; fields[0] == "Section.RealField"`	Pass	0.000s
Line 157: __repr__ returns class name.	`"MortgageCDM" in repr(mortgage_cdm)`	Pass	0.000s
Verify data is cached after first load.	`len(result1) == len(result2) == 3`	Pass	0.001s
Test cache clearing.	`loader._cache is not None; loader._cache is None` (+1 more)	Pass	0.001s
Verify force_reload bypasses cache.	`len(result1) == 3; len(result2) == 0`	Pass	0.002s
Test cache invalidation.	`loader._cache_valid is True; loader._cache_valid is False`	Pass	0.001s
Verify abstract methods must be implemented.	—	Pass	0.000s
Subclass must implement get_entity_id.	—	Pass	0.000s
Subclass must implement get_entity_summary.	—	Pass	0.000s
Test file_exists method.	`loader.file_exists() is True; loader2.file_exists() is False`	Skip	0.000s
Load data with alternative container key (portfolio).	`len(result) == 1`	Pass	0.001s
Load data when JSON has container key.	`len(result) == 2`	Pass	0.001s
Load data when JSON is a direct list.	`len(result) == 1; result[0]['PropertyHeader']['Header']['PropertyID'] == 'P1'`	Pass	0.001s
Handle missing file gracefully.	`result == []`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Load data when JSON is single object.	`len(result) == 1`	Pass	0.001s
Test count method.	`property_loader.count() == 3`	Pass	0.001s
Test exists method.	`property_loader.exists("PROP-001") is True; property_loader.exists("NONEXISTENT") is False`	Pass	0.001s
Test finding entity by ID.	`result is not None; result['PropertyHeader']['Header']['PropertyID'] == "PROP...`	Pass	0.001s
Test finding non-existent entity.	`result is None`	Pass	0.001s
Test getting all IDs.	`len(ids) == 3; "PROP-001" in ids` (+2 more)	Skip	0.000s
Test get_status method.	`status['entity_type'] == 'property'; status['file_exists'] is True` (+1 more)	Skip	0.000s
Test listing all entities.	`len(result) == 3; all('propertyId' in item for item in result)`	Pass	0.001s
Custom panel dimensions	`"1400px" in js; "900px" in js`	Pass	0.000s
Default panel dimensions	`panel.panel_width == "1100px"; panel.panel_height == "750px"`	Pass	0.000s
Panel has four top level tabs	`js.count("'Hazard Curve'") >= 1; js.count("'Term Structure'") >= 1` (+2 more)	Pass	0.000s
Distance decay available	`"_drawDistanceDecay" in js`	Pass	0.000s
Elevation section available	`"_drawElevationSection" in js`	Pass	0.000s
Gauge module injected	`"var basisGaugeChart" in js`	Pass	0.000s
Property module injected	`"var basisPropertyChart" in js`	Pass	0.000s
Shd module injected	`"var basisSHDChart" in js`	Pass	0.000s
She module injected	`"var basisSHEChart" in js`	Pass	0.000s
Basis active sub tab state	`"var basisActiveSubTab" in js`	Pass	0.000s
Basis selected storm state	`"var basisSelectedStorm" in js`	Pass	0.000s
All basis charts destroyed before rendering new sub-tab.	`"basisGaugeChart.destroy()" in js; "basisSHEChart.destroy()" in js` (+2 more)	Pass	0.000s
Chart container set to flex-column for basis layouts.	`"flexDirection" in js`	Pass	0.000s
Chart container style reset when switching to non-basis tabs.	`"container.style.display = ''" in js`	Pass	0.000s
hidePanel must reset basis state and destroy charts.	`"basisSelectedStorm = null" in js; "basisActiveSubTab = 0" in js`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Render basis explorer function	"function renderBasisExplorer()" in js	Pass	0.000s
Render basis sub tab function	"function renderBasisSubTab(idx)" in js	Pass	0.000s
Sub tab bar created	"phc-basis-subtab-bar" in js	Pass	0.000s
Sub-tab bar should be removed when switching to non-basis tabs.	"existingSubBar" in js or "subBar.remove()" in js	Pass	0.000s
Sub tab bar styling	"phc-basis-subtab" in js	Pass	0.000s
Sub tab dispatches gauge	"renderBasisGauge()" in js	Pass	0.000s
Sub tab dispatches property	"renderBasisProperty()" in js	Pass	0.000s
Sub tab dispatches shd	"renderBasisSHD()" in js	Pass	0.000s
Sub tab dispatches she	"renderBasisSHE()" in js	Pass	0.000s
Sub tab names array	"basisSubTabNames" in js	Pass	0.000s
Sub tab names defined	"'Gauge'" in js; "'SHE'" in js (+2 more)	Pass	0.000s
Switch tab calls basis explorer	"renderBasisExplorer()" in js	Pass	0.000s
Tab bar label is Basis Explorer, not Basis Analysis.	"'Basis Explorer'" in js; "'Basis Analysis'" not in js	Pass	0.000s
Chart destroyed before render	"basisGaugeChart.destroy()" in js	Pass	0.000s
Chart variable declared	"var basisGaugeChart" in js	Pass	0.000s
Click triggers rerender	"renderBasisSubTab(basisActiveSubTab)" in js	Pass	0.000s
Severe storms red, others grey.	"'#F44336'" in js; "'#BDBDBD'" in js	Pass	0.000s
Reads gauge thresholds	"gauge_thresholds" in js	Pass	0.000s
Reads nearest gauges	"phcData.nearest_gauges" in js	Pass	0.000s
Reads storm details	"phcData.storm_details" in js	Pass	0.000s
Render function defined	"function renderBasisGauge()" in js	Pass	0.000s
Scatter chart type	"type: 'scatter'" in js	Pass	0.000s
Severe count in summary	"severeCount" in js	Pass	0.000s
Sorts by peak level	"gauge_peak_m" in js; "sort(" in js	Pass	0.000s
Stats bar updated	"phc-stats-bar" in js	Pass	0.000s
Storm click sets selected	"basisSelectedStorm" in js	Pass	0.000s
Threshold lines for alert	"alertLevel" in js; "alertLine" in js	Pass	0.000s
Threshold lines for severe	"severeLevel" in js; "severeLine" in js	Pass	0.000s
Threshold lines for warning	"warningLevel" in js; "warningLine" in js	Pass	0.000s
Tooltip shows storm id	"Storm " in js; "stormId" in js	Pass	0.000s
Uses exceeded_severe from storm_details, not threshold comparison.	"exceeded_severe" in js	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Basis computed	"propSpread - gaugeSpread" in js	Pass	0.000s
Chart variable declared	"var basisPropertyChart" in js	Pass	0.000s
Click sets selected storm	"basisSelectedStorm" in js	Pass	0.000s
Colour by damage ratio	"damageRatio" in js; "'#FFB74D'" in js (+3 more)	Pass	0.000s
Gauge spread in stats	"gauge_spread_bps" in js	Pass	0.000s
Property spread in stats	"property_spread_bps" in js	Pass	0.000s
Quadrant basis risk	"qBottomRight" in js; "Basis Risk" in js	Pass	0.000s
Quadrant hedged	"qTopRight" in js; "Hedged" in js	Pass	0.000s
Quadrant no event	"qBottomLeft" in js; "No Event" in js	Pass	0.000s
Quadrant unhedged	"qTopLeft" in js; "Unhedged" in js	Pass	0.000s
Reads spread decomposition	"phcData.spread_decomposition" in js	Pass	0.000s
Reads storm details	"phcData.storm_details" in js	Pass	0.000s
Render function defined	"function renderBasisProperty()" in js	Pass	0.000s
Scatter chart type	"type: 'scatter'" in js	Pass	0.000s
Severe vertical annotation	"severeLine" in js; "Gauge Severe Trigger" in js	Pass	0.000s
Summary cards rendered	"#E8F5E9" in js; "#FFF3E0" in js (+1 more)	Pass	0.000s
Tooltip shows quadrant info	"Basis risk:" in js or "basis risk" in js; "Hedged" in js	Pass	0.000s
Uses exceeded severe flag	"exceeded_severe" in js	Pass	0.000s
X axis is gauge peak	"Gauge Peak Water Level" in js	Pass	0.000s
Y axis is property depth	"Property Flood Depth" in js	Pass	0.000s
Axes drawn	"'Distance (km)'" in js; "'Retention Factor'" in js	Pass	0.000s
Chart variable declared	"var basisSHDChart" in js	Pass	0.000s
Click sets selected storm	"basisSelectedStorm" in js	Pass	0.000s
Blue gradient for depth, grey for zero.	"'#BDBDBD'" in js	Pass	0.000s
Decay curve: retention = 1 - d/25.	"1 - d / 25" in js	Pass	0.000s
Distance decay function defined	"function _drawDistanceDecay(" in js	Pass	0.000s
Gauge marker at origin	"'#4CAF50'" in js	Pass	0.000s
Property marker	"'#F44336'" in js	Pass	0.000s
Reads distance km	"distance_km" in js	Pass	0.000s
Reads retention factor	"retention_factor" in js	Pass	0.000s
Reads storm details	"phcData.storm_details" in js	Pass	0.000s
Render function defined	"function renderBasisSHD()" in js	Pass	0.000s
Scatter chart type	"type: 'scatter'" in js	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Selected storm gets a highlight circle.	"'#FF9800'" in js	Pass	0.000s
Shd spread in stats	"shd_spread_bps" in js	Pass	0.000s
Sorts by flood depth	"flood_depth_m" in js; "sort(" in js	Pass	0.000s
Summary shows avg retention	"avgRetention" in js	Pass	0.000s
Summary shows with depth count	"withDepthCount" in js	Pass	0.000s
Tooltip shows flood depth	"Flood depth:" in js	Pass	0.000s
Tooltip shows retention	"Retention:" in js	Pass	0.000s
Two canvas layout	"basis-shd-decay" in js; "basis-shd-scatter" in js	Pass	0.000s
Bankfull level uses 0.5m offset from severe.	"severeLevel - 0.5" in js	Pass	0.000s
Chart variable declared	"var basisSHEChart" in js	Pass	0.000s
Click sets selected storm	"basisSelectedStorm" in js	Pass	0.000s
Colours green for reaches	"'#4CAF50'" in js	Pass	0.000s
Colours grey for no impact	"'#BDBDBD'" in js	Pass	0.000s
Colours orange for severe but misses	"'#FF9800'" in js	Pass	0.000s
Cross section draws ground	"gaugeElev" in js; "propElev" in js (+1 more)	Pass	0.000s
Cross section draws water	"rgba(33, 150, 243" in js	Pass	0.000s
Cross section severe line	"'Severe'" in js	Pass	0.000s
Elevation section function defined	"function _drawElevationSection(" in js	Pass	0.000s
Flood threshold calculated	"floodThreshold" in js	Pass	0.000s
Floor level drawn	"'Floor +" in js	Pass	0.000s
Height diff calculated	"heightDiff" in js	Pass	0.000s
Property threshold annotation	"propThreshold" in js	Pass	0.000s
Reaches property classification	"reachesProperty" in js	Pass	0.000s
Reads floor level	"phcData.floor_level_m" in js	Pass	0.000s
Reads gauge elevation	"gauge_elevation_m" in js	Pass	0.000s
Reads property elevation	"phcData.elevation_m" in js	Pass	0.000s
Reads storm details	"phcData.storm_details" in js	Pass	0.000s
Render function defined	"function renderBasisSHE()" in js	Pass	0.000s
Scatter chart type	"type: 'scatter'" in js	Pass	0.000s
She spread in stats	"she_spread_bps" in js	Pass	0.000s
Summary shows counts	"reachCount" in js; "severeCount" in js	Pass	0.000s
Two canvas layout	"basis-she-section" in js; "basis-she-scatter" in js	Pass	0.000s
Post loop cancellation returns early	cl_mod._batch_job is None	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Batch returns immediately when cancelled before start	`call_count == 0`; `cl_mod._batch_job is None`	Pass	0.000s
Successful batch reads metrics from summary	`cl_mod._batch_job["status"] == "complete"`; `result["auc_roc"] == 0.95` (+1 more)	Pass	0.001s
ETA computed from elapsed/completed when avg_per_gauge is 0.	`data["eta_seconds"] is not None`; `data["eta_seconds"] >= 40`	Pass	0.001s
ETA is None when no gauges completed yet and no historical avg.	`data["eta_seconds"] is None`	Pass	0.001s
POST when every gauge has a .joblib returns 'complete'.	`data["status"] == "complete"`; `data["message"] == "All gauges already trained"`	Pass	0.008s
POST while batch is running returns current progress without starting new job.	`data["status"] == "running"`; `data["completed"] == 2` (+1 more)	Pass	0.001s
Complete status includes results list.	`data["status"] == "complete"`; `len(data["results"]) == 2`	Pass	0.001s
POST returns running or complete.	`data["status"] in ("running", "complete", "error")`	Pass	0.003s
Status endpoint returns expected fields when batch is running.	`data["status"] == "running"`; `data["total"] == 10` (+4 more)	Pass	0.001s
Status endpoint returns idle when no batch is running.	`data["status"] == "idle"`	Pass	0.001s
Pdf not generated returns 404	`r.status_code == 404`; `"not yet generated" in r.get_json()["message"]`	Pass	0.001s
Save failure returns 500	`r.status_code == 500`; `r.get_json()["status"] == "error"`	Pass	0.001s
Line 74: POST add when save fails.	`r.status_code == 500`; `r.get_json()["status"] == "error"`	Pass	0.002s
Line 112: DELETE when save fails.	`r.status_code == 500`; `r.get_json()["status"] == "error"`	Pass	0.002s
Line 100: PUT update when save fails.	`r.status_code == 500`; `r.get_json()["status"] == "error"`	Pass	0.002s
Closed trades excluded from blotter by default.	`'PRS-TEST-001' not in swap_ids`	Pass	0.038s
include_closed=true returns closed trades too.	`'PRS-TEST-001' in swap_ids`	Pass	0.038s
Corrupt JSON in prs dir is skipped via inner exception (lines 134-135).	`resp.status_code == 200`; `data['status'] == 'success'`	Pass	0.052s
Blotter returns 500 when engine raises (lines 94-96).	`resp.status_code == 500`; `data['status'] == 'error'`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Close trade returns 500 when engine raises (lines 251-253).	`resp.status_code == 500`; `data['status'] == 'error'`	Pass	0.001s
PDF regen exception is caught; trade close still succeeds (lines 236-237).	`resp.status_code == 200`; `data['swap_id'] == 'PRS-TEST-001'` (+1 more)	Pass	0.026s
Trade with empty GaugeName gets name injected from gauge locations.	`resp.status_code == 200`; `noname[0].get('gauge_name', '') != ''`	Pass	0.056s
New trade in blotter	`blotter['status'] == 'success'`; `swap_id in trade_ids`	Pass	0.051s
P&L bar should display net notional (signed sum).	`'netNotional' in js`; `'fmtGBP(netNotional)' in js`	Pass	0.000s
Payer (Pay) should show negative notional.	`'is_payer' in js`; `'netNotional' in js`	Pass	0.000s
PRS-*.json glob must match PRS-{8hex} filenames (the production format).	`fnmatch.fnmatch(name, 'PRS-*.json')`; `fnmatch.fnmatch(name, 'PRS-*.json')`	Pass	0.000s
When PRS trade files are present, GET /trading/blotter returns them.	`len(found) == 1`	Pass	0.000s
When QuantLib IS available, book must export its public API.	`isinstance(THAMES_CENTRAL_AREAS, list)`; `len(THAMES_CENTRAL_AREAS) > 0`	Pass	0.000s
port.src.book itself can be imported without QuantLib because	`hasattr(mod, "THAMES_CENTRAL_AREAS")`	Pass	0.002s
Thames Central book should cover exactly 10 gauge areas.	`len(THAMES_CENTRAL_AREAS) == 10`	Pass	0.000s
models.hazard.pricing has 'import QuantLib as ql' at module level.	—	Pass	0.001s
Family members shown	`isinstance(result, list)`	Pass	0.000s
Has borrower header	`any("Borrower" in t for t in texts)`	Pass	0.000s
Has table	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
Income verification shown	`isinstance(result, list)`	Pass	0.000s
Marital status shown	`isinstance(result, list)`	Pass	0.000s
Nationality shown	`isinstance(result, list)`	Pass	0.000s
No borrower details returns early	`any("No borrower" in t for t in texts)`	Pass	0.000s
No mortgage returns message	`any("No mortgage" in t for t in texts)`	Pass	0.000s
Residency status shown	`isinstance(result, list)`	Pass	0.000s
Secondary income shown	`isinstance(result, list)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
With mortgage returns list	`isinstance(result, list) and len(result) > 0`	Pass	0.001s
All none returns empty dicts	`result["mortgage_lookup"] == {}; result["gauge_flood_info"] == {}` (+1 more)	Pass	0.000s
All none returns four keys	`set(result.keys()) == { "mortgage_lookup", "g...`	Pass	0.000s
With data populates lookups	`"P1" in result["mortgage_lookup"]`	Pass	0.000s
Audit trail date range	`len(story) >= 6`	Pass	0.000s
Audit trail empty	`len(story) >= 2`	Pass	0.000s
Audit trail no timestamps	`len(story) >= 4`	Pass	0.000s
Audit trail renders	`len(story) > 0`	Pass	0.000s
Bcbs239 all compliant	`len(story) >= 4`	Pass	0.000s
Bcbs239 at risk	`len(story) >= 5`	Pass	0.000s
Bcbs239 no principles	`len(story) >= 2`	Pass	0.000s
Bcbs239 renders	`len(story) > 0`	Pass	0.000s
No trades returns message	`any("No open PRS trades" in t for t in texts)`	Pass	0.000s
Receiver trade direction	`isinstance(result, list)`	Pass	0.000s
Trigger capitalized	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
With trades returns table	`len(tables) >= 1`	Pass	0.000s
With trades shows count	`any("Total trades:  2" in t for t in texts)`	Pass	0.000s
Line 186: when CDM validation returns errors, a warning is logged.	`record is not None; "PhysicalSwap" in record` (+1 more)	Pass	0.001s
When compound event causes flooding, readings are present.	`"readings" in result; result["flood_depth_m"] > 0` (+2 more)	Pass	0.164s
Property too high for flood -¿ no readings key.	`result["flooded"] is False; "readings" not in result` (+1 more)	Pass	0.164s
Isolated storm type passed through to event.	`result["sequence_type"] == "isolated"; result["num_pulses"] == 1`	Pass	0.164s
Compound event has all expected keys including compound-specific ones.	`result["compound"] is True; result["num_pulses"] == 2` (+2 more)	Pass	0.164s
When gauge severe ¿ base_level, superposition cap is computed.	`call_kwargs is not None; cap_arg is not None`	Pass	0.164s
Distance=0 -¿ travel_time_hrs=0.	`result["travel_time_hrs"] == 0.0`	Pass	0.164s

Test Description	Acceptance Criteria	Result	Time
Compound cluster should produce more flooding than its best pulse alone.	`max_cluster > max_single`	Pass	0.001s
Cluster should have longer flood duration than isolated.	`dur_cluster > dur_iso`	Pass	0.001s
A big isolated pulse should flood the property.	`max_depth > 2.0`	Pass	0.000s
Property too high above gauge -¿ no flooding.	`max_depth == 0.0`	Pass	0.000s
Output schema	`len(result) == 168; all('hour' in r for r in result)` (+3 more)	Pass	0.000s
Cover basic	`len(story) > 0`	Pass	0.001s
Cover no bcbs	`len(story) >= 3`	Pass	0.000s
Cover no tests	`len(story) >= 3`	Pass	0.000s
Cover with bcbs	`'BCBS' in combined or len(story) >= 5`	Pass	0.001s
Cover with tests	`len(story) >= 5`	Pass	0.000s
Documents empty	`len(story) >= 2`	Pass	0.000s
Documents file list	`len(story) >= 4`	Pass	0.000s
Documents renders	`len(story) > 0`	Pass	0.000s
Exec summary all pass	`len(story) > 0`	Pass	0.000s
Exec summary failures	`len(story) > 0`	Pass	0.000s
Exec summary no coverage	`len(story) > 0`	Pass	0.000s
Exec summary no tests	`len(story) > 0`	Pass	0.001s
Exec summary renders	`len(story) > 0`	Pass	0.000s
When flooded, arrival_time and peak_time are set.	`result["arrival_time_hrs"] is not None; result["peak_time_hrs"] is not None`	Pass	0.164s
High WSE above property elevation -¿ flooded=True.	`result["flooded"] is True; result["flood_depth_m"] > 0`	Pass	0.164s
Gaugets with empty readings -¿ no readings key (not flooded).	`"readings" not in result`	Pass	0.164s
Low WSE -¿ flooded=False.	`result["flooded"] is False`	Pass	0.164s
_build_flood_event returns dict with required keys.	`key in result; "readings" in result`	Pass	0.164s
Distance=0 -¿ travel_time=0 path.	`result["travel_time_hrs"] == 0.0`	Pass	0.164s
Both none returns empty	`build_gauge_flood_info(None, None) == {}`	Pass	0.000s
Gauge data supplements missing fields	`"GAUGE-002" in result; result["GAUGE-002"]["gauge_name"] == "My Gauge"` (+2 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Gauge without id skipped	`result == {}`	Pass	0.000s
Hazard curves only	`"GAUGE-001" in result; result["GAUGE-001"]["gauge_name"] == "Thames at Chelsea"` (+2 more)	Pass	0.000s
Hazard data takes precedence	`result["GAUGE-003"]["gauge_name"] == "From HC"`	Pass	0.000s
Line 247: FileNotFoundError when gauge.json missing.	—	Pass	0.004s
Line 235: FileNotFoundError when storm_sequences.json missing.	—	Pass	0.000s
Verbose=False should skip log messages.	`result['summary']['num_gauges'] == 1; len(info_msgs) == 0`	Pass	0.023s
Lines 242, 254, 273, 280-281, 288: verbose logging.	`result['summary']['num_gauges'] == 1; result['summary']['num_storms'] == 20` (+3 more)	Pass	0.025s
Inventory empty	`len(story) >= 2`	Pass	0.000s
Inventory no risk ratings	`len(story) >= 3`	Pass	0.000s
Inventory rag colours	`len(story) >= 3`	Pass	0.000s
Inventory renders	`len(story) > 0`	Pass	0.000s
Inventory risk ratings	`len(story) >= 4`	Pass	0.000s
Mrc completed actions excluded	`len(story) >= 3`	Pass	0.000s
Mrc no decisions	`len(story) >= 3`	Pass	0.000s
Mrc no meetings	`len(story) >= 2`	Pass	0.000s
Mrc open actions	`len(story) >= 4`	Pass	0.000s
Mrc renders	`len(story) > 0`	Pass	0.000s
Mrc with decisions	`len(story) >= 5`	Pass	0.000s
No market state returns message	`any("No market hazard term structure" in t for t in texts)`	Pass	0.000s
Some tenor slots missing -¿ show 'N/A'.	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
With adjustments returns extra table	`len(tables) >= 2`	Pass	0.000s
With hts returns table	`len(tables) >= 1`	Pass	0.000s
All sections	`len(content) > 1000`	Pass	0.003s
Buffer not empty	`len(content) > 100`	Pass	0.003s
Meeting with actions	`isinstance(result, io.BytesIO)`	Pass	0.001s
Meeting with agenda	`isinstance(result, io.BytesIO)`	Pass	0.001s
Meeting with decisions	`isinstance(result, io.BytesIO)`	Pass	0.001s
Meeting with minutes	`isinstance(result, io.BytesIO)`	Pass	0.001s
Meeting with no date/time/location/chair still works.	`isinstance(result, io.BytesIO)`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Meeting with participants	isinstance(result, io.BytesIO)	Pass	0.003s
Minimal meeting	isinstance(result, io.BytesIO)	Pass	0.001s
Returns bytesio	isinstance(result, io.BytesIO)	Pass	0.003s
Empty dict returns empty	build_mortgage_lookup({}) == {}	Pass	0.000s
Loan from loan details fallback	result["PROP-002"]["OriginalLoan"] == 300_000	Pass	0.000s
Mortgage without property id skipped	build_mortgage_lookup(data) == {}	Pass	0.000s
Multiple mortgages	set(result.keys()) == {"P1", "P2"}	Pass	0.000s
No items key returns empty	build_mortgage_lookup({"other": []}) == {}	Pass	0.000s
None returns empty	build_mortgage_lookup(None) == {}	Pass	0.000s
Single mortgage	"PROP-001" in result; result["PROP-001"]["mortgage_id"] == "MORT-001" (+3 more)	Pass	0.000s
Basic risk info	"M1" in result["by_mortgage_id"]; "P1" in result["by_property_id"] (+3 more)	Pass	0.000s
Missing mortgage or property id skipped	result == {"by_mortgage_id": {}, "by_property_id": {}}	Pass	0.000s
No property flood info uses unknown	result["by_mortgage_id"]["M2"]["flood_risk_level"] == "Un...	Pass	0.000s
None mortgage returns empty structure	result == {"by_mortgage_id": {}, "by_property_id": {}}	Pass	0.000s
Value at risk calculation	result["by_mortgage_id"]["M4"]["mortgage_value_at_risk"] ...	Pass	0.001s
60 ¡= percentile ¡ 80 -¿ 'Elevated'.	any('Elevated' in t for t in texts)	Pass	0.088s
FloodAlert and FloodWarning in metadata -¿ no crash.	len(elements) > 0	Pass	0.095s
80 ¡= percentile ¡ 95 -¿ 'High'.	any('High' in t for t in texts)	Pass	0.090s
percentile ¡ 40 -¿ 'Low'.	any('Low' in t for t in texts)	Pass	0.089s
40 ¡= percentile ¡ 60 -¿ 'Normal'.	any('Normal' in t for t in texts)	Pass	0.091s
Single same_month_levels entry -¿ stdev guard -¿ no crash.	len(elements) > 0	Pass	0.087s
Stats table and distribution statistics header present.	any('Distribution Statistics' in t for t in texts)	Pass	0.088s
percentile ¿= 95 -¿ 'Very High'.	any('Very High' in t for t in texts)	Pass	0.091s
Returns position function	'getPositionStyles' in css; 'top-right' in css (+1 more)	Pass	0.000s
All none returns empty	build_property_flood_info(None, None, None) == {}	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
From property data when not in phc	`"PROP-002" in result; result["PROP-002"]["flood_zone"] == "Zone 2"` (+1 more)	Pass	0.000s
From property hazard data	`"PROP-001" in result; result["PROP-001"]["annual_flood_prob"] == pytest.approx(...` (+2 more)	Pass	0.000s
Phc takes precedence over property data	`result["PROP-003"]["annual_flood_prob"] == pytest.approx(...`	Pass	0.000s
Base level when no exceedance	—	Pass	0.000s
Peak near expected hour	`abs(peak_hour - expected) <= 2`	Pass	0.000s
Peak value	`max(arr) == pytest.approx(8.0, abs=0.5)`	Pass	0.000s
Raci escalation triggers dict	`len(story) >= 5`	Pass	0.000s
Raci escalation triggers string	`len(story) >= 5`	Pass	0.000s
Raci missing backup	`len(story) >= 3`	Pass	0.000s
Raci no roles	`len(story) >= 2`	Pass	0.000s
Raci renders	`len(story) > 0`	Pass	0.000s
Raci with activities	`len(story) >= 5`	Pass	0.000s
alert ¡= peak ¡ warning -¿ 'Alert'.	`len(elements) > 1`	Pass	0.000s
ValueError/TypeError in date parsing (lines 179-180) -¿ continue.	`len(elements) > 0`	Pass	0.000s
Days within 3 days grouped; higher level updates peak.	`not any('No flood events' in t for t in texts)`	Pass	0.000s
Consecutive days -¿ single event with range format.	`len(elements) > 1`	Pass	0.000s
No days exceeding alert -¿ 'No flood events recorded'.	`any('No flood events' in t for t in texts)`	Pass	0.000s
Peak ¿= severe_level -¿ 'Severe' row in table.	`len(elements) > 1`	Pass	0.000s
Single-day flood: date_str = start.strftime(...) only.	`len(elements) > 1`	Pass	0.000s
Days ¿ 3 days apart -¿ two separate flood events.	`len(elements) > 1`	Pass	0.000s
warning ¡= peak ¡ severe -¿ 'Warning'.	`len(elements) > 1`	Pass	0.000s
Recommendations all clean	`len(story) >= 3`	Pass	0.000s
Recommendations bcbs at risk	`len(story) >= 4`	Pass	0.000s
Recommendations overdue remediation	`len(story) >= 4`	Pass	0.000s
Recommendations peer reviewer gap	`len(story) >= 4`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Recommendations raci backup gaps	`len(story) >= 4`	Pass	0.000s
Recommendations renders	`len(story) > 0`	Pass	0.001s
Recommendations failures	`len(story) >= 4`	Pass	0.000s
Recommendations validation gap	`len(story) >= 4`	Pass	0.000s
Remediation all completed	`len(story) >= 4`	Pass	0.000s
Remediation empty	`len(story) >= 2`	Pass	0.000s
Remediation overdue highlighted	`len(story) >= 4`	Pass	0.000s
Remediation priority sorting	`len(story) >= 4`	Pass	0.000s
Remediation renders	`len(story) > 0`	Pass	0.000s
Empty schema returns empty dict	`make_generator(tmp_path)._build_section({}, 0, {}) == {}`	Pass	0.000s
Nested dict without type recurses	`"SubSection" in result; isinstance(result["SubSection"], dict)`	Pass	0.000s
Non dict schema returns empty dict	`make_generator(tmp_path)._build_section("not a dict", 0, ...`	Pass	0.000s
None schema returns empty dict	`make_generator(tmp_path)._build_section(None, 0, {}) == {}`	Pass	0.000s
None value from generate field value excluded	`"AField" not in result`	Pass	0.000s
Reserved key description skipped	`"description" not in make_generator(tmp_path)._build_sect...`	Pass	0.000s
Reserved key options skipped	`"options" not in make_generator(tmp_path)._build_section(...`	Pass	0.000s
Reserved key type skipped	`"type" not in result`	Pass	0.000s
Reserved key values skipped	`"values" not in make_generator(tmp_path)._build_section({...`	Pass	0.000s
Contains all types	`t.key in parsed; 'icon' in parsed[t.key]` (+2 more)	Pass	0.000s
Info template values	`parsed['info']['color'] == '#2196F3'`	Pass	0.000s
Returns valid json	`isinstance(parsed, dict)`	Pass	0.000s
Testing no junit	`len(story) >= 3`	Pass	0.000s
Testing no sensitivity generators	`len(story) >= 4`	Pass	0.000s
Testing per model coverage	`len(story) >= 6`	Pass	0.000s
Testing renders	`len(story) > 0`	Pass	0.000s
Testing with coverage	`len(story) >= 5`	Pass	0.000s
Testing with sensitivity generators	`len(story) >= 6`	Pass	0.000s
Validation all reviewers assigned	`len(story) >= 3`	Pass	0.000s
Validation high assumptions	`len(story) > 3`	Pass	0.000s
Validation no assumptions	`len(story) >= 3`	Pass	0.000s
Validation no questions	`len(story) >= 1`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Validation peer reviewer gaps	`len(story) >= 4`	Pass	0.000s
Validation renders	`len(story) > 0`	Pass	0.000s
Line 88: exceedance probs between 0 and 0.01 are capped to 0.01.	`pt.annual_exceedance_prob >= 0.01 or pt.annual_exceedance...`	Pass	0.026s
Line 61: log() does nothing when verbose=False.	`"should not appear" not in caplog.text`	Pass	0.000s
Line 61: log() calls logger.info when verbose=True.	`"test message" in caplog.text`	Pass	0.001s
Line 152: progress logged every 10 gauges.	`len(curves) == 11; "10/11" in caplog.text or "Processed" in caplog.text`	Pass	0.187s
All days before start	`d < today`	Pass	0.000s
Chronological order	`days[i] > days[i - 1]`	Pass	0.000s
Ends before start	`d < start`	Pass	0.000s
Large count	`len(days) == 252; d.weekday() < 5`	Pass	0.000s
Last element should be the business day immediately before start.	`last < start; last.weekday() < 5`	Pass	0.000s
No weekends	`d.weekday() < 5`	Pass	0.000s
Returns ascending order	`result == sorted(result)`	Pass	0.000s
Returns business days only	`d.weekday() < 5`	Pass	0.000s
Returns correct count	`len(days) == 63`	Pass	0.000s
Returns correct count	`len(result) == 5`	Pass	0.000s
Single day	`len(days) == 1; days[0].weekday() < 5`	Pass	0.000s
Single day	`len(result) == 1; result[0].weekday() < 5`	Pass	0.000s
Skips weekends	`wd not in returned_set`	Pass	0.000s
The day before Monday is Sunday — should skip back to Friday.	`days[0].weekday() == 4`	Pass	0.000s
5 days back from Wednesday should contain no Saturday/Sunday.	`len(days) == 5; d.weekday() < 5`	Pass	0.000s
Zero days	`result == []`	Pass	0.000s
The blueprint source should include EAFloodZone in Pricing section.	`'"EAFloodZone"' in source`	Pass	0.001s
Blueprint source has ea flood zone actual	`'"EAFloodZoneActual"' in source`	Pass	0.000s
Blueprint source has terrain delta bps	`'"TerrainDeltaBps"' in source`	Pass	0.000s
Terrain fields should use data.get() with defaults for backward compat.	`'data.get("ea_flood_zone", "")' in source; 'data.get("ea_flood_zone_actual", "")' in source` (+1 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Cdm has ea flood zone	`zone is not None; zone in EA_FLOOD_ZONE_RATES`	Pass	0.003s
Cdm has ground level	`ra.get("GroundLevelMeters") is not None`	Pass	0.003s
Cdm has location	`loc.get("LatitudeDegrees") is not None; loc.get("LongitudeDegrees") is not None`	Pass	0.003s
Cdm has properties	`len(data.get("properties", [])) > 0`	Pass	0.005s
Cdm has reference gauges	`len(refs) > 0`	Pass	0.003s
Test CORS headers on GET request.	`response.status_code == 200`	Pass	0.000s
Test OPTIONS requests return 200.	`response.status_code == 200`	Pass	0.001s
Larger area gives larger energy	`sum(large) > sum(small)`	Pass	0.000s
Returns positive value	`result > 0`; `result > 0` (+2 more)	Pass	0.000s
Uses default area when missing	`result > 0`; `result > 0`	Pass	0.000s
Is float	`isinstance(result, float); isinstance(result, float)`	Pass	0.000s
Larger area tends to give larger emissions	`sum(large) > sum(small)`	Pass	0.000s
Returns positive value	`result > 0`; `result > 0` (+2 more)	Pass	0.000s
Uses default area when missing	`result > 0`; `result > 0`	Pass	0.000s
Correct calculation	`abs(result - 2430.0) < 0.01`	Pass	0.000s
Is float	`isinstance(result, float); isinstance(result, float)`	Pass	0.000s
Returns positive value	`result > 0`; `result > 0` (+2 more)	Pass	0.000s
Uses defaults when missing	`result >= 0; abs(result - 2430.0) < 0.01`	Pass	0.000s
Zero usage returns zero	`result == 0.0`	Pass	0.000s
Above severe level some positive	`np.any(depths > 0)`	Pass	0.000s
Below severe level returns zeros	—	Pass	0.001s
Depths capped at 5m	`np.all(depths <= 5.0)`	Pass	0.000s
Empty gauge data returns zeros	—	Pass	0.000s
No gauge data returns zeros	—	Pass	0.000s
Returns correct length	`len(depths) == len(properties)`	Pass	0.000s
Lines 50-57: non-None gauge_data calls calculate_synthetic_flood_depths.	`len(depths) == 3; isinstance(depths, np.ndarray)`	Pass	0.000s
Key remapping: 'flow_cumecs' → 'flow'.	`"mean_flow" in result; "mean_flow_cumecs" not in result`	Pass	0.000s
Flow stats should include q5_flow and q95_flow.	`"q5_flow" in result; "q95_flow" in result`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Returns dict	`isinstance(result, dict)`; `isinstance(result, dict)` (+1 more)	Pass	0.000s
Less than remaining energy	`result <= 9001`	Pass	0.000s
No negative when electricity exceeds total	`result == 0`	Pass	0.000s
Returns non negative	`result >= 0`	Pass	0.000s
Uses defaults when missing	`result >= 0`; `abs(result - 2430.0) < 0.01`	Pass	0.000s
Fraction of total	`all(4500 <= r <= 12001 for r in results)`	Pass	0.000s
Returns positive value	`result > 0`; `result > 0` (+2 more)	Pass	0.000s
Uses default energy when missing	`result > 0`	Pass	0.000s
Key remapping: 'level_meters' → 'level'.	`"mean_level" in result`; `"mean_level_meters" not in result`	Pass	0.001s
Has min max	`"min_level" in result`; `"max_level" in result`	Pass	0.001s
Returns dict	`isinstance(result, dict)`; `isinstance(result, dict)` (+1 more)	Pass	0.001s
Buy to let type	`result["mortgage_type"] == "Buy-to-Let"`	Pass	0.000s
High flood risk reduces ltv	`high_risk["ltv_ratio"] <= normal["ltv_ratio"] + 0.01`	Pass	0.000s
Loan amount positive	`result["loan_amount"] > 0`	Pass	0.000s
Monthly payment positive	`result["monthly_payment"] > 0`	Pass	0.000s
Required keys present	`key in result`	Pass	0.000s
Returns dict	`isinstance(result, dict)`	Pass	0.000s
Second home type	`result["mortgage_type"] == "Second Home"`	Pass	0.000s
Conditional rating middle range	`result["calculated_rating"] in ("Conditional", "Unaccepta...`	Pass	0.000s
Due soon gives half score	`result["component_scores"]["review_currency"] == 0.5`	Pass	0.000s
Far review gives full score	`result["component_scores"]["review_currency"] == 1.0`	Pass	0.000s
High assumptions penalised	`result["component_scores"]["assumption_risk"] == 0.0`	Pass	0.000s
High coverage acceptable	`result["calculated_rating"] == "Acceptable"`	Pass	0.000s
High limitations penalised	`result["component_scores"]["limitation_risk"] == 0.0`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
No coverage unacceptable	`result["calculated_rating"] == "Unacceptable"`	Pass	0.000s
No review date gives zero	`result["component_scores"]["review_currency"] == 0.0`	Pass	0.000s
Overdue review gives zero score	`result["component_scores"]["review_currency"] == 0.0`	Pass	0.000s
Returns dict with required keys	`"calculated_rating" in result; "calculated_score" in result` (+1 more)	Pass	0.000s
Score is float	`isinstance(result["calculated_score"], float)`	Pass	0.000s
Heat pump returns zero	`result == 0`	Pass	0.000s
Multiple returns positive	`result > 0`	Pass	0.000s
No solar returns zero	`result == 0`	Pass	0.000s
No system returns zero	`result == 0`	Pass	0.000s
Solar pv returns positive	`result > 0`	Pass	0.000s
Solar uses default area when missing	`result > 0`	Pass	0.000s
Annual maxima present	`"annual_maxima" in stats; stats["annual_maxima"]["years_count"] == 2`	Pass	0.000s
Basic stats keys	`key in stats`	Pass	0.000s
Empty returns empty dict	`GaugeHistoricalDaily().calculate_statistics([]) == {}`	Pass	0.000s
Extreme events above p99	`"extreme_events" in stats; stats["extreme_events"]["count"] >= 1`	Pass	0.000s
Mean correct	`gen.calculate_statistics(flows)["mean_flow"] == pytest.ap...`	Pass	0.000s
Min max correct	`stats["min_flow"] == pytest.approx(5.0); stats["max_flow"] == pytest.approx(100.0)`	Pass	0.000s
Monthly means present	`"01" in stats["monthly_means"]; "02" in stats["monthly_means"]`	Pass	0.000s
Percentiles present	`"percentiles" in stats; p in stats["percentiles"]`	Pass	0.000s
Single flow no std dev error	`gen.calculate_statistics(flows)["std_dev"] == 0`	Pass	0.000s
All percentile keys present	`p in stats["percentiles"]`	Pass	0.000s
Annual maxima events capped at ten	`len(stats["annual_maxima"]["events"]) <= 10`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Extreme events top 10 capped	`len(stats["extreme_events"]["top_10"]) <= 10`	Pass	0.000s
Max flow date correctly identified	`stats["max_flow_date"] == "2020-01-02"; stats["max_flow"] == pytest.approx(200.0)`	Pass	0.000s
Q5 flow equals p95	`stats["q5_flow"] == stats["percentiles"]["p95"]`	Pass	0.000s
Q95 flow equals p5	`stats["q95_flow"] == stats["percentiles"]["p5"]`	Pass	0.000s
Record start and end match data	`stats["record_start"] == "2018-05-01"; stats["record_end"] == "2021-08-15"`	Pass	0.000s
Single year annual maxima std dev zero	`stats["annual_maxima"]["years_count"] == 1; stats["annual_maxima"]["std_dev"] == 0`	Pass	0.000s
Total years approximate	`stats["total_days"] == 20; stats["total_years"] == round(20 / 365.25, 1)`	Pass	0.000s
Two flow std dev non zero	`gen.calculate_statistics(flows)["std_dev"] > 0`	Pass	0.000s
All same values	`result["mean_level_meters"] == 3.0; result["std_dev"] == 0.0`	Pass	0.000s
Empty returns empty dict	`calculate_timeseries_statistics([], "level_meters") == {}`	Pass	0.000s
Flood exceedances zero when all below	`result["flood_exceedances"]["flood_alert"]["count"] == 0`	Pass	0.000s
Flood stages adds exceedances	`"flood_exceedances" in result; result["flood_exceedances"]["severe_warning"]["count"] >= 1`	Pass	0.001s
Flow cumecs adds q flows	`"q95_flow" in result; "q5_flow" in result`	Pass	0.000s
Has annual maxima	`"annual_maxima" in result; "years_count" in result["annual_maxima"]`	Pass	0.001s
Has basic keys	`"total_days" in result; "mean_level_meters" in result` (+2 more)	Pass	0.001s
Has extreme events	`"extreme_events" in result`	Pass	0.001s
Has monthly means	`"monthly_means" in result`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Has percentiles	`"percentiles" in result; "p5" in result["percentiles"]` (+1 more)	Pass	0.001s
Max date correct	`result["max_level_meters_date"] == "2026-01-02"`	Pass	0.000s
No flood stages no exceedances key	`"flood_exceedances" not in result`	Pass	0.001s
Record start end	`result["record_start"] == "2020-01-01"; result["record_end"] == "2025-12-31"`	Pass	0.000s
Returns dict	`isinstance(result, dict); isinstance(result, dict)` (+1 more)	Pass	0.001s
Single observation no stdev error	`result["std_dev"] == 0`	Pass	0.000s
Total days correct	`result["total_days"] == 100`	Pass	0.000s
Explicit none params	`isinstance(level, float)`	Pass	0.000s
None params uses defaults	`isinstance(level, float); level > 0`	Pass	0.000s
Lines 54-56: internal error returns 500 JSON error.	`r.status_code in (404, 500); data["status"] == "error"`	Pass	0.001s
Lines 108-113: exception in set_catchment returns 500.	`r.status_code in (200, 500)`	Pass	0.001s
Uses addEventListener, not inline onclick for sub-tabs.	`'addEventListener' in stress_js; 'stress-ctab-0' in stress_js` (+2 more)	Pass	0.000s
Flood Probability sub-tab present.	`'Flood Probability' in stress_js`	Pass	0.000s
Stress P&L sub-tab present.	`'Stress P' in stress_js`	Pass	0.000s
Surface sub-tab present.	`'Surface' in stress_js`	Pass	0.000s
Consumer has no hash for input	`not ok; "no recorded hash" in issues[0].lower()`	Pass	0.000s
Consumer never run	`not ok; "never run" in issues[0].lower()`	Pass	0.000s
Fresh when hashes match	`ok; issues == []`	Pass	0.000s
No hash recorded for producer output	`not ok; "No hash recorded" in issues[0]`	Pass	0.000s
Producer never run	`not ok; "never run" in issues[0].lower()`	Pass	0.000s
Root steps (gauges, counterparties) have no inputs → always fresh.	`ok; issues == []`	Pass	0.000s
Stale when hashes differ	`not ok; "stale" in issues[0].lower()`	Pass	0.000s
Unknown step	`not ok; "Unknown step" in issues[0]`	Pass	0.000s
All outputs present → complete.	`result["complete"]; result["missing"] == []` (+1 more)	Pass	0.005s
Empty directory is as broken as missing.	`not result["complete"]; len(empty) == 1` (+2 more)	Pass	0.005s

Test Description	Acceptance Criteria	Result	Time
Missing gauge.json → incomplete.	`not result["complete"]`; `"gauge.json" in missing_outputs` (+2 more)	Pass	0.000s
Only root steps present → reports downstream as missing.	`not result["complete"]`; `result["present"] == 3` (+3 more)	Pass	0.001s
All pipeline steps missing produces 'missing' statuses.	`len(results) == 14`; `r["status"] == "missing"` (+2 more)	Pass	0.000s
A directory step checks most recent file mtime.	`ghd_step["status"] == "fresh"`	Pass	0.001s
A directory with only old files is stale.	`ghd_step["status"] == "stale"`	Pass	0.001s
An empty directory counts as missing.	`ghd_step["status"] == "missing"`	Pass	0.001s
A recently modified file is reported as 'fresh'.	`gauge_step["status"] == "fresh"`; `gauge_step["last_run"] is not None` (+1 more)	Pass	0.000s
Dotfiles in directories are ignored.	`ghd_step["status"] == "missing"`	Pass	0.001s
If manifest has last_run AND file exists, manifest last_run is kept.	`gauge_step["last_run"] == "2026-01-01T00:00:00"`	Pass	0.000s
If lineage manifest has last_run, it is carried through.	`gauge_step["last_run"] == "2026-03-01T12:00:00"`	Pass	0.000s
A file older than 72 hours is reported as 'stale'.	`prop_step["status"] == "stale"`; `len(prop_step["issues"]) == 1` (+1 more)	Pass	0.000s
All present	`ok`; `missing == []`	Pass	0.002s
Missing upstream	`not ok`; `"gauges" in missing`	Pass	0.000s
Nearby ranges round to same cache key.	`L1 is L2`	Pass	0.000s
Second call at same range returns cached object.	`L1 is L2`	Pass	0.000s
Cholesky lower triangular	`np.allclose(np.triu(L, k=1), 0.0)`	Pass	0.000s
L @ L.T == C within numerical tolerance.	`np.allclose(L @ L.T, C, atol=1e-10)`	Pass	0.002s
Thames cholesky reconstruction	`np.allclose(L @ L.T, C, atol=1e-8)`	Pass	0.007s
When np.linalg.cholesky raises LinAlgError on first call,	`L.shape == (model.n_gauges, model.n_gauges)`; `call_count[0] == 2`	Pass	0.000s
Generator error has message	`r.get_json().get("message")`	Pass	0.005s
Generator error returns 500	`r.status_code == 500`	Pass	0.004s
Generator error status is error	`r.get_json()["status"] == "error"`	Pass	0.004s
Claim ref is built from last 8 chars of property_id.	`len(result) > 1000`	Pass	0.017s

Test Description	Acceptance Criteria	Result	Time
Generate empty sequence lookup	`isinstance(result, bytes)`	Pass	0.021s
Generate returns bytes	`isinstance(result, bytes); len(result) > 0`	Pass	0.022s
Generate without mortgage	`isinstance(result, bytes); result[:4] == b'%PDF'`	Pass	0.017s
Minimal prop_data with a single flood event should not raise.	`isinstance(result, bytes)`	Pass	0.013s
Output is valid pdf	`result[:4] == b'%PDF'`	Pass	0.021s
Pdf size reasonable	`len(result) > 8_000`	Pass	0.021s
Content length header set	`r.headers.get("Content-Length") == str(len(fake_pdf))`	Pass	0.004s
Filename contains prop id	`CLAIM_PROP_ID in cd`	Pass	0.003s
Response body is pdf bytes	`r.data == fake_pdf`	Pass	0.004s
Response has content disposition	`"attachment" in cd; CLAIM_PROP_ID in cd`	Pass	0.004s
Response is pdf content type	`r.content_type == "application/pdf"`	Pass	0.004s
Success returns 200	`r.status_code == 200`	Pass	0.004s
Corrupt storm_sequences.json triggers the except in _load_sequence_lookup (L56-57).	`r.status_code == 200; seq_lookup == {}`	Pass	0.031s
_load_mortgage_record exception (L89-91) returns None; report still generated.	`r.status_code == 200; mort_record is None`	Pass	0.001s
_load_prop_record exception (L71-73) returns {}; report still generated.	`r.status_code == 200; prop_record == {}`	Pass	0.002s
_load_sequence_lookup exception (L56-57) returns empty dict; report still generated.	`r.status_code == 200`	Pass	0.002s
No dir message mentions propertyts	`"propertyts" in msg or "property" in msg or "not found" i...`	Pass	0.000s
No dir returns 404	`r.status_code == 404`	Pass	0.000s
No dir status error	`r.get_json()["status"] == "error"`	Pass	0.000s
No floods message mentions events	`"flood" in msg or "event" in msg`	Pass	0.001s
No floods returns 404	`r.status_code == 404`	Pass	0.002s
No floods status error	`r.get_json()["status"] == "error"`	Pass	0.002s
Unknown property message contains id	`"PROP-GHOST" in r.get_json()["message"]`	Pass	0.000s
Unknown property returns 404	`r.status_code == 404`	Pass	0.000s
Unknown property status error	`r.get_json()["status"] == "error"`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Metadata categories contains expected keys	`key in cats`	Pass	0.000s
Metadata categories station has required subkeys	`key in station_keys`	Pass	0.000s
Schema version is string	`isinstance(GaugeHistoricalDaily.SCHEMA_VERSION, str); len(GaugeHistoricalDaily.SCHEMA_VERSION) > 0`	Pass	0.000s
Classifiers that exist should reference current gauge IDs.	`len(valid) > 0`	Skip	0.004s
training_summary.json must exist alongside classifiers.	—	Skip	0.000s
Job dict says 'failed' =¿ status failed with error message.	`resp.status_code == 200; data["status"] == "failed"` (+2 more)	Pass	0.000s
Failed job with no 'error' key falls back to 'Unknown error'.	`data["status"] == "failed"; data["error"] == "Unknown error"`	Pass	0.000s
If joblib exists on disk, status is 'ready' even if job dict says 'training'.	`data["status"] == "ready"`	Pass	0.001s
No joblib file and no training job =¿ not_trained.	`resp.status_code == 200; data["status"] == "not_trained"` (+1 more)	Pass	0.000s
Job dict says 'ready' (joblib might not exist yet) =¿ status ready.	`resp.status_code == 200; data["status"] == "ready"`	Pass	0.000s
If a .joblib file exists for the gauge, status is 'ready'.	`resp.status_code == 200; data["status"] == "ready"` (+1 more)	Pass	0.001s
When a training job is in progress, status is 'training' with elapsed.	`resp.status_code == 200; data["status"] == "training"` (+4 more)	Pass	0.000s
Unknown gauge with no joblib and no job =¿ not_trained.	`resp.status_code == 200; data["status"] == "not_trained"`	Pass	0.001s
Status endpoint	`'/api/v1/trading/stress/classifier-status/' in cl_js`	Pass	0.000s
Summary endpoint	`'/api/v1/trading/classifiers/summary' in cl_js`	Pass	0.000s
Train all endpoint	`'/api/v1/trading/classifiers/train-all' in cl_js`	Pass	0.000s
Train endpoint	`'/api/v1/trading/stress/train/' in cl_js`	Pass	0.000s
hidePanel calls clCleanupCharts.	`'clCleanupCharts' in full_js`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
'classifiers' included in switchTab views array.	`"'classifiers'" in full_js`	Pass	0.000s
switchTab calls loadClassifiersData.	`'loadClassifiersData()' in full_js`	Pass	0.000s
Tab button array includes classifiers.	`"'classifiers'" in full_js or '"classifiers"' in full_js`	Pass	0.000s
createClassifiersView() called in panel setup.	`'createClassifiersView()' in full_js`	Pass	0.000s
All gauges have .joblib → ready=True.	`data["ready"] is True`; `data["trained"] == data["total"]` (+1 more)	Pass	0.010s
Internal error → 500 with error status.	`resp.status_code == 500`; `data["status"] == "error"` (+1 more)	Pass	0.009s
No .joblib files → ready=False.	`resp.status_code == 200`; `data["status"] == "success"` (+4 more)	Pass	0.010s
Some gauges trained → ready=False, correct counts.	`data["ready"] is False`; `data["trained"] == 2` (+1 more)	Pass	0.008s
classifiers.get_js() returns non-empty string.	`len(cl_js) > 0`	Pass	0.000s
'_clStartBatchTraining' triggers train-all.	`'_clStartBatchTraining' in cl_js`	Pass	0.000s
'clCleanupCharts' destroys Chart.js instances.	`'clCleanupCharts' in cl_js`	Pass	0.000s
'createClassifiersView' builds the Tab 9 DOM.	`'createClassifiersView' in cl_js`	Pass	0.000s
'_clRenderFeatureChart' renders feature importance bar chart.	`'_clRenderFeatureChart' in cl_js`	Pass	0.000s
'loadClassifiersData' fetches summary on tab open.	`'loadClassifiersData' in cl_js`	Pass	0.000s
'_clMetricCard' builds metric display cards.	`'_clMetricCard' in cl_js`	Pass	0.000s
'_clUpdateProgress' updates the batch progress bar.	`'_clUpdateProgress' in cl_js`	Pass	0.000s
'_clRenderDetail' renders single-gauge detail panel.	`'_clRenderDetail' in cl_js`	Pass	0.000s
'_clRenderSummaryTable' renders the all-gauge table.	`'_clRenderSummaryTable' in cl_js`	Pass	0.000s
'_clStartSingleTraining' triggers per-gauge training.	`'_clStartSingleTraining' in cl_js`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Module-level state variables must be declared.	'clSummary' in cl_js; 'clSelectedGaugeId' in cl_js (+2 more)	Pass	0.000s
Avg auc computed	data["num_trained"] == 1; data["avg_auc_roc"] == 0.95	Pass	0.021s
Gauge fields	"gauge_id" in g; "gauge_name" in g (+4 more)	Pass	0.008s
Gauges sorted by longitude	lons == sorted(lons)	Pass	0.009s
No .joblib files → all gauges show has_model=False.	data["num_trained"] == 0; g["has_model"] is False	Pass	0.003s
Returns all gauges	"gauges" in data; data["num_total"] > 0	Pass	0.003s
Returns success	resp.status_code == 200; data["status"] == "success"	Pass	0.012s
When training_summary.json + .joblib exist, metrics appear.	len(trained) == 1; g["has_model"] is True (+4 more)	Pass	0.008s
Above high threshold is high	_classify_property_risk(0.5) == "High"	Pass	0.000s
High threshold	_classify_property_risk(0.03) == "High"	Pass	0.000s
Just below medium is low	_classify_property_risk(0.009) == "Low"	Pass	0.000s
Medium threshold	_classify_property_risk(0.01) == "Medium"	Pass	0.000s
Tiny prob is low	_classify_property_risk(0.005) == "Low"	Pass	0.000s
Zero is negligible	_classify_property_risk(0.0) == "Negligible"	Pass	0.000s
No models to clear → removed=0.	resp.status_code == 200; data["status"] == "success" (+1 more)	Pass	0.001s
Internal error → 500.	resp.status_code == 500; data["status"] == "error"	Pass	0.001s
Clears .joblib files and training_summary.json.	data["status"] == "success"; data["removed"] == 3 (+3 more)	Pass	0.009s
clear-all still returns success when port_stress import fails.	data["status"] == "success"; data["removed"] == 1	Pass	0.001s
Close response includes final P&L.	'final_pnl' in data	Pass	0.031s
After close, trade status is set to Closed in marks file.	marks_file.exists(); marks['PRS-TEST-002']['trade_status'] == 'Closed' (+2 more)	Pass	0.032s

Test Description	Acceptance Criteria	Result	Time
Closing non-existent trade returns 404.	`resp.status_code == 404`	Pass	0.019s
Close-out without close-out_spread_bps returns 400.	`resp.status_code == 400`	Pass	0.000s
Settlement amount reflects the negotiated closeout spread, not fair spread.	`data['status'] == 'success'`; `data['close_spread_bps'] == 400.0` (+2 more)	Pass	0.031s
Closing a trade at a negotiated spread returns success with close spread.	`resp.status_code == 200`; `data['status'] == 'success'` (+3 more)	Pass	0.029s
Closed trade should not appear in stress results.	`resp.status_code == 200`; `t['swap_id'] != 'PRS-CLOSED-001'`	Pass	0.054s
post_damage_ltv is recomputed independently for each storm.	`ltvs["STORM-cl2"] > ltvs["STORM-cl1"]`; `ltvs["STORM-cl2"] > ltvs["STORM-cl3"]`	Pass	0.012s
Each cluster storm individually queryable	`r.status_code == 200`; `data["status"] == "success"`	Pass	0.005s
STORM-cl2 (damage_ratio=0.15) causes more damage than cl1 or cl3.	`damages["STORM-cl2"] > damages["STORM-cl1"]`; `damages["STORM-cl2"] > damages["STORM-cl3"]`	Pass	0.004s
Querying a storm_id that didn't breach alert returns 404.	`r.status_code == 404`	Pass	0.003s
Summary max_depth_m reflects the worst storm in the cluster.	`result["summary"]["max_depth_m"] >= 0.0`	Pass	0.165s
Cluster produces three flood events	`len(data["flood_events"]) == 3`	Pass	0.200s
Cluster three distinct storm ids in events	`storm_ids == {"STORM-c001a", "STORM-c001b", "STORM-c001c"}`	Pass	0.168s
The 'from port.src.book import ...' statement in cmd_port must be	`blotter_idx != -1`; `book_idx != -1` (+1 more)	Pass	0.001s
Importing app.commands.port must not trigger QuantLib.	`hasattr(mod, "cmd_port")`; `hasattr(mod, "register_parser")`	Pass	0.001s
chain_result has expected keys.	`'is_consistent' in chain`; `'stale_steps' in chain` (+1 more)	Pass	3.720s
Every step in graph should have a step_details entry.	`graph_steps == detail_steps`	Pass	3.738s
Pipeline must have at least one step.	`data['num_steps'] > 0`	Pass	5.013s
collect_all() returns a dict with expected keys.	`isinstance(data, dict)`; `'manifest' in data` (+4 more)	Pass	5.452s
step_details is a list of dicts.	`isinstance(data['step_details'], list)`; `isinstance(sd, dict)` (+5 more)	Pass	4.959s

Test Description	Acceptance Criteria	Result	Time
Blue is hex colour	`BLUE == colors.HexColor('#1565C0')`	Pass	0.000s
Dark blue	`DARK_BLUE == colors.darkblue`	Pass	0.000s
Grey	`GREY == colors.HexColor('#757575')`	Pass	0.000s
Light amber	`LIGHT_AMBER == colors.HexColor('#FFF8E1')`	Pass	0.000s
Light blue	`LIGHT_BLUE == colors.HexColor('#E3F2FD')`	Pass	0.000s
Light green	`LIGHT_GREEN == colors.HexColor('#E8F5E9')`	Pass	0.000s
Light red	`LIGHT_RED == colors.HexColor('#FFEBEE')`	Pass	0.000s
Age 50 to 100	`mid_age > young`	Pass	0.000s
Age over 100	`old > young`	Pass	0.000s
Basic returns float	`isinstance(score, float); 0 < score <= 10`	Pass	0.000s
Construction concrete lower	`concrete < brick`	Pass	0.000s
Construction steel lowest	`steel < concrete`	Pass	0.000s
Construction timber higher	`timber > brick`	Pass	0.000s
Construction wood equals timber	`timber == wood`	Pass	0.000s
High ltv multiplier	`high > low`	Pass	0.000s
Ltv 60 to 80 multiplier	`mid > base`	Pass	0.000s
Ltv 80 to 95 multiplier	`mid > base`	Pass	0.000s
None ltv no multiplier	`0 < score <= 10`	Pass	0.000s
Unknown construction type	`0 < score <= 10`	Pass	0.000s
Unknown flood level	`0 < score <= 10`	Pass	0.000s
Very high flood capped	`score == 10.0`	Pass	0.000s
Very low flood lowest score	`vl < vh`	Pass	0.000s
All sections together	`isinstance(result, list); len(tables) >= 2`	Pass	0.000s
Feeding bad data (non-dict regulatory) exercises the except block.	`isinstance(result, list)`	Pass	0.000s
mortgage_data without 'Mortgage' key → uses top-level.	`isinstance(result, list)`	Pass	0.000s
Commit button must call tdCommitMarket().	`'tdCommitMarket()' in js or 'tdCommitMarket(' in js`	Pass	0.000s
Market tab must contain a visible commit button with correct ID.	`'td-commit-btn' in js`	Pass	0.000s
Commit button must NOT have display:none in its own style.	`'display:none' not in own_style`	Pass	0.000s
tdCommitMarket function must be defined.	`'tdCommitMarket = function' in js or \{} 'fu...`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Commit function must POST to hazard-term-structure/commit.	'/trading/hazard-term-structure/commit' in js	Pass	0.000s
After commit, must switch to blotter tab (which loads blotter data).	"switchTab('blotter')" in js	Pass	0.000s
Close out path exercised	r.status_code in (200, 400)	Pass	0.035s
Terrain fields should use —— fallback for missing values.	"result.selectedZone || ''" in js; "result.actualZone || ''" in js (+1 more)	Pass	0.000s
Ea flood zone actual in payload	'ea_flood_zone_actual: result.actualZone' in js	Pass	0.000s
Ea flood zone in payload	'ea_flood_zone: result.selectedZone' in js	Pass	0.000s
Terrain delta bps in payload	'terrain_delta_bps: result.terrainDelta' in js	Pass	0.000s
Commit with cashflows — exercises PDF cashflow schedule section.	r.status_code in (200, 400)	Pass	0.006s
Commit with maturity_date provided explicitly.	r.status_code in (200, 400)	Pass	0.006s
Commit with EA flood zone terrain fields.	r.status_code in (200, 400); r.get_json()["status"] in ("success", "error")	Pass	0.008s
Commit without terrain fields should still work (backward compatibility).	r.status_code in (200, 400)	Pass	0.005s
Empty json returns 400	r.status_code == 400; r.get_json()["status"] == "error"	Pass	0.000s
Commit with enough fields to pass CDM validation.	r.status_code in (200, 400); r.get_json()["status"] in ("success", "error")	Pass	0.006s
No json returns error	r.status_code in (400, 500); r.get_json()["status"] == "error"	Pass	0.004s
Validation fails returns 400	r.status_code == 400; data["status"] == "error" (+1 more)	Pass	0.030s
commit_yield_curve creates yield_curve from default when missing.	updated['yield_curve']['1'] == 0.04; updated['yield_curve']['2'] == 0.045	Pass	0.002s
Advised flag	isinstance(result, list)	Pass	0.000s
Business purpose false	isinstance(result, list)	Pass	0.000s
Business purpose true	isinstance(result, list)	Pass	0.000s
Common section returns elements	isinstance(result, list); len(result) > 2	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Fca reference number	`len(tables) >= 1`	Pass	0.000s
Multiple regulatory flags	`isinstance(result, list)`	Pass	0.000s
Contains header	`"Scenario Comparison" in result`	Pass	0.366s
Contains scenario names	`name in result`	Pass	0.366s
None uses all scenarios	`len(result) > 50`	Pass	0.368s
Returns string	`isinstance(result, str)`	Pass	0.367s
Single scenario	`first_name in result`	Pass	0.074s
Two scenarios	`name in result`	Pass	0.147s
Shim class is same	`New is Shim`	Pass	0.000s
Shim imports generator	`ClaimReportGenerator is not None`	Pass	0.001s
Line 159: _create_summary calls get_entity_summary.	`summary == {'id': 'Y', 'name': 'Ypsilon'}`	Pass	0.000s
Line 155: _extract_id calls get_entity_id.	`loader._extract_id(entity) == 'X'`	Pass	0.000s
Line 163: _get_container_keys returns CONTAINER_KEYS.	`loader._get_container_keys() == ['items']`	Pass	0.000s
Line 167: get_file_status is alias for get_status.	`status['entity_name'] == 'stub'; status['exists'] is True` (+1 more)	Pass	0.000s
Persistent sequences (SHORT gaps) should show stronger compounding than doublets (MEDIUM gaps).	`persistent_excess >= doublet_excess - 0.1`	Pass	0.048s
During the gap, level should drop from Storm 1 peak.	`level_at_gap_end < s1_peak`	Pass	0.000s
For a short-gap persistent sequence, level does not return to baseline during gap.	`level_at_gap_end > default_gauge.base_level + 0.05`	Pass	0.000s
Spec requirement: Storm 2 peak ¿ Storm 1 peak in ¿= 60% of doublets.	`fraction >= 0.60`	Pass	0.037s
Gauge ids match	`{r.gauge_id for r in responses} == {gauge.gauge_id, gauge...`	Pass	0.000s
Near gauge higher response	`near_resp.peak_level >= far_resp.peak_level`	Pass	0.000s
Returns one response per gauge	`len(responses) == 2`	Pass	0.000s
All kernels decrease with distance	`near >= far`	Pass	0.000s
All kernels in 0 1 range	`0.0 <= decay <= 1.0`	Pass	0.000s
Exponential at half	`decay == pytest.approx(math.exp(-0.5), rel=1e-4)`	Pass	0.000s
Exponential at one lambda	`decay == pytest.approx(math.exp(-1.0), rel=1e-4)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Gaussian at one sigma	`decay == pytest.approx(math.exp(-0.5), rel=1e-4)`	Pass	0.000s
Gaussian at two sigma	`decay == pytest.approx(math.exp(-2.0), rel=1e-4)`	Pass	0.000s
Linear beyond cutoff is zero	`decay == 0.0`	Pass	0.000s
Linear midpoint	`decay == pytest.approx(0.75, abs=0.001)`	Pass	0.000s
Zero distance returns one all kernels	`model.compute_decay(0, 100, kernel, 1.0) == 1.0`	Pass	0.000s
Consistent when all clear	`dl_mod._compute_health(chain, lr) == 'CONSISTENT'`	Pass	0.000s
Degraded on stale steps	`dl_mod._compute_health(chain, lr) == 'DEGRADED'`	Pass	0.000s
Degraded on failures	`dl_mod._compute_health(chain, lr) == 'DEGRADED'`	Pass	0.000s
Health colour mapping	`dl_mod._health_colour('CONSISTENT') == dl_mod.GREEN`; `dl_mod._health_colour('DEGRADED') == dl_mod.AMBER` (+2 more)	Pass	0.000s
Inconsistent on broken chain	`dl_mod._compute_health(chain, lr) == 'INCONSISTENT'`	Pass	0.000s
After end uses last point	`intensity >= 0`	Pass	0.000s
At peak time is nonzero	`intensity > 0`	Pass	0.000s
Before start uses first point	`intensity >= 0`	Pass	0.000s
Empty track returns zero	`result == 0.0`	Pass	0.000s
Far from track is lower	`near > far`	Pass	0.000s
dist ¡ 1.0 -¿ short-circuit to _build_flood_event directly.	`result is not None`; `result["storm_id"] == "STORM-001"`	Pass	0.164s
Multiple gauges in nearest -¿ IDW path used.	`result is not None`	Pass	0.251s
gauge_responses has no entry for nearest gauge -¿ None.	`result is None`	Pass	0.164s
Normal IDW path returns dict with storm_id.	`result is not None`; `result["storm_id"] == "STORM-001"`	Pass	0.166s
Response without pulse_peaks -¿ simple event (no compound key).	`result is not None`; `result.get("compound") is not True`	Pass	0.164s
Response with pulse_peaks -¿ compound event (compound=True).	`result is not None`; `result.get("compound") is True` (+1 more)	Pass	0.164s

Test Description	Acceptance Criteria	Result	Time
Line 126: annuity ¡= 0 returns annual_hazard_rate * 10000.	`result > 0`	Pass	0.000s
Negative hazard returns zero	`result == 0.0`	Pass	0.000s
Verify compute_prs_spread uses yield_curve when provided.	`result > 0`	Pass	0.000s
Zero hazard returns zero	`result == 0.0`	Pass	0.000s
Accumulation nonnegative	`resp.accumulation >= 0`	Pass	0.000s
Duration nonnegative	`resp.duration_above_alert >= 0`; `resp.duration_above_warning >= 0` (+1 more)	Pass	0.000s
Flooded flag correct	`resp.flooded == (resp.peak_level >= gauge.flood_alert)`	Pass	0.000s
High intensity storm floods	`resp.peak_level > gauge.base_level`	Pass	0.000s
Peak level at least base	`resp.peak_level >= gauge.base_level`	Pass	0.000s
Peak level is max of series	`resp.peak_level == pytest.approx(max(levels))`	Pass	0.000s
Response ids match	`resp.gauge_id == gauge.gauge_id`; `resp.storm_id == simple_storm.storm_id`	Pass	0.000s
Timeseries length	`len(resp.level_timeseries) == expected_steps`; `len(resp.intensity_timeseries) == expected_steps`	Pass	0.000s
To dict roundtrip	`d['gauge_id'] == gauge.gauge_id`; `'level_timeseries' in d` (+1 more)	Pass	0.000s
To dict without timeseries	`'level_timeseries' not in d`	Pass	0.000s
Pre-wetting the catchment before the same storm raises the peak.	`np.max(wet) > np.max(dry)`	Pass	0.001s
Level should be declining in the later part of the drainage window.	`np.all(np.diff(tail) <= 0.01)`	Pass	0.004s
Extreme storm should produce higher peak than moderate.	`np.max(l_ext) > np.max(l_mod)`	Pass	0.001s
With identical first storm parameters, doublet produces ¿= isolated peak.	`np.max(l_dbl) >= np.max(l_iso)`	Pass	0.001s
Severe storm should raise level above baseline.	`np.max(levels) > default_gauge.base_level + 0.5`	Pass	0.000s
River level must always be ¿= base_level.	`np.all(levels >= default_gauge.base_level - 1e-6)`	Pass	0.006s
Spec constraint: no precipitation after hour 156.	`np.all(p[156:] == pytest.approx(0.0))`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Output shape	`levels.shape == (EVENT_WINDOW_HOURS,)`	Pass	0.000s
Level at hour 167 should be below the peak level (recovering, not diverging).	`final <= peak + 0.01; final_excess < 1.5`	Pass	0.006s
Peak above base	`peak >= default_gauge.base_level`	Pass	0.004s
Peak indices match storm indices	`idx == storm.storm_index`	Pass	0.002s
Returns one peak per storm	`len(peaks) == seq.num_storms`	Pass	0.004s
Compliant when all pass	`dl_mod._compute_verdict(chain, lr) == 'COMPLIANT'`	Pass	0.000s
Empty lineage results	`dl_mod._compute_verdict(chain, {}) == 'PARTIALLY COMPLIANT'`	Pass	0.000s
Non compliant on broken chain	`dl_mod._compute_verdict(chain, lr) == 'NON-COMPLIANT'`	Pass	0.000s
Non compliant on majority failures	`dl_mod._compute_verdict(chain, lr) == 'NON-COMPLIANT'`	Pass	0.000s
Non compliant on missing steps	`dl_mod._compute_verdict(chain, lr) == 'NON-COMPLIANT'`	Pass	0.000s
Partially compliant on failures	`dl_mod._compute_verdict(chain, lr) == 'PARTIALLY COMPLIANT'`	Pass	0.000s
No tests run cannot prove compliance.	`dl_mod._compute_verdict(chain, lr) == 'PARTIALLY COMPLIANT'`	Pass	0.000s
Spatial correlation constants in config.port must be self-consistent.	`SPATIAL_CORR_ENABLED is True; SPATIAL_CORR_MODEL_TYPE == "exponential"` (+9 more)	Pass	0.000s
Config structure	`sc["enabled"] is True; sc["model_type"] == "exponential"` (+3 more)	Pass	0.001s
Round trip params	`loaded.base_range_km == 35.0; loaded.nugget == 0.08`	Pass	0.001s
Save creates file	`path.exists()`	Pass	0.000s
Configure both menus	`handler.property_menu == pm; handler.gauge_menu == gm`	Pass	0.000s
Configure gauge menu	`handler.gauge_menu == new_gm`	Pass	0.000s
Configure max visible	`ns.max_visible == 10`	Pass	0.000s
Configure menus	`mgr.context_menus.property_menu == custom_prop`	Pass	0.000s
Configure no args no change	`handler.server_url == 'http://test:1234'`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Configure no args no change	handler.property_menu == DEFAULT_PROPERTY_MENU; handler.gauge_menu == DEFAULT_GAUGE_MENU	Pass	0.000s
Configure no args no change	ns.position == NotificationPosition.TOP_LEFT; ns.timeout == 7000 (+1 more)	Pass	0.000s
Configure notification position	mgr.notifications.position == NotificationPosition.BOTTOM...	Pass	0.000s
Configure notification timeout	mgr.notifications.timeout == 2000	Pass	0.000s
Configure position	ns.position == NotificationPosition.BOTTOM_LEFT	Pass	0.000s
Configure property menu	handler.property_menu == new_pm	Pass	0.000s
Configure returns self	result is mgr	Pass	0.000s
Configure server url	result is mgr; mgr.backend.server_url == 'http://new:9999'	Pass	0.000s
Configure timeout	ns.timeout == 3000	Pass	0.000s
Configure updates endpoints	handler.endpoints == {'a': '/b', 'c': '/d'}	Pass	0.000s
Configure updates server url	handler.server_url == 'http://new:2222'; handler.endpoints == {'a': '/b'}	Pass	0.000s
Margin positive	MARGIN > 0	Pass	0.000s
Page dimensions	PAGE_W > 0; PAGE_H > 0 (+1 more)	Pass	0.000s
Custom parameters	d.params.base_mean == 50; d.params.tail_threshold == 70	Pass	0.000s
Default parameters	d.params.base_mean == 45.0; d.params.base_std == 15.0 (+4 more)	Pass	0.000s
From params object	d.params.base_mean == 55	Pass	0.000s
From scenario baseline	d is not None; d.params.name == "baseline"	Pass	0.000s
From scenario unknown raises	—	Pass	0.000s
Seed gives deterministic output	s1 == s2	Pass	0.000s
Tail prob between 0 and 1	0.0 < d._tail_prob < 1.0	Pass	0.000s
Construction year shown	isinstance(result, list)	Pass	0.000s
Empty property does not crash	isinstance(self._page().generate_elements({}), list)	Pass	0.000s
Floor level shown	isinstance(result, list)	Pass	0.000s
Has section header	any("Construction" in t for t in texts)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Has table	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
No construction data returns message	`any("No construction" in t for t in texts)`	Pass	0.000s
No construction year	`isinstance(self._page().generate_elements(prop), list)`	Pass	0.000s
Returns list with data	`isinstance(result, list) and len(result) > 0`	Pass	0.000s
Site height shown	`isinstance(result, list)`	Pass	0.000s
Brick construction	`isinstance(result, list)`	Pass	0.000s
Concrete construction	`isinstance(result, list)`	Pass	0.000s
Empty construction type	`isinstance(result, list)`	Pass	0.000s
Masonry construction	`isinstance(result, list)`	Pass	0.000s
Steel construction	`isinstance(result, list)`	Pass	0.000s
Timber construction	`isinstance(result, list)`	Pass	0.000s
Unknown construction type	`isinstance(result, list)`	Pass	0.000s
Custom menus	`handler.property_menu == pm; handler.gauge_menu == gm`	Pass	0.000s
Default init	`handler.property_menu == DEFAULT_PROPERTY_MENU; handler.gauge_menu == DEFAULT_GAUGE_MENU`	Pass	0.000s
Each startup cache variable must be initialised as window.X = null. [var='_preGovAudit']	`f'window.{var}' in src`	Pass	0.000s
Each startup cache variable must be initialised as window.X = null. [var='_preGovBib']	`f'window.{var}' in src`	Pass	0.000s
Each startup cache variable must be initialised as window.X = null. [var='_preStorms']	`f'window.{var}' in src`	Pass	0.000s
Each startup cache variable must be initialised as window.X = null. [var='_tdPreBlotter']	`f'window.{var}' in src`	Pass	0.000s
Each startup cache variable must be initialised as window.X = null. [var='_tdPreEodHistory']	`f'window.{var}' in src`	Pass	0.000s
Each startup cache variable must be initialised as window.X = null. [var='_tdPreGauges']	`f'window.{var}' in src`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Each startup cache variable must be initialised as window.X = null. [var='_tdPreGovDocs']	`f'window.{var}' in src`	Pass	0.000s
Each startup cache variable must be initialised as window.X = null. [var='_tdPreMarket']	`f'window.{var}' in src`	Pass	0.000s
Each startup cache variable must be initialised as window.X = null. [var='_tdPrePortStorms']	`f'window.{var}' in src`	Pass	0.000s
Each startup cache variable must be initialised as window.X = null. [var='_tdPreStressStorms']	`f'window.{var}' in src`	Pass	0.000s
Each startup cache variable must be initialised as window.X = null. [var='_tdPreYieldCurve']	`f'window.{var}' in src`	Pass	0.000s
No startup cache var should have a bare var declaration.	`f'var {var}' not in src`	Pass	0.000s
The preload-done flag must NOT be a var (regression: 2026-03-12 incident).	`'var _tdPreloadDone' not in src`	Pass	0.001s
Window td preload done initialised	`'window._tdPreloadDone = false' in src`	Pass	0.000s
Window td preload done set after settle	`'window._tdPreloadDone = true' in src`	Pass	0.000s
Blotter nulls cache after consume	`'window._tdPreBlotter = null' in src`	Pass	0.000s
Blotter reads window pre blotter	`'window._tdPreBlotter' in src`	Pass	0.000s
Gov docs nulls cache after consume	`'window._tdPreGovDocs = null' in src`	Pass	0.000s
Gov docs reads window pre gov docs	`'window._tdPreGovDocs' in src`	Pass	0.001s
Port stress nulls cache after consume	`'window._tdPrePortStorms = null' in src`	Pass	0.000s
Port stress reads window pre port storms	`'window._tdPrePortStorms' in src`	Pass	0.000s
preloader.py must not assign bare _tdPreloadDone (creates unreliable global).	`not bare_assigns`	Pass	0.000s
trading/preloader.py fallback must also set window._tdPreloadDone.	`'window._tdPreloadDone = true' in src`	Pass	0.001s
Storm portfolio nulls cache after consume	`'window._preStorms = null' in src`	Pass	0.000s
Storm portfolio reads window pre storms	`'window._preStorms' in src`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
showPanel() must read window._tdPreloadDone, not bare _tdPreloadDone.	`'window._tdPreloadDone' in src`	Pass	0.000s
Blotter setup no bare pre blotter	—	Pass	0.000s
Gov docs no bare pre gov docs	—	Pass	0.000s
Port stress no bare pre port storms	—	Pass	0.001s
preloader.py checks should all use window._tdPreloadDone.	`not bad_lines`	Pass	0.001s
Storm table no bare pre storms	—	Pass	0.001s
Tradingdesk no bare preload done	—	Pass	0.001s
Governance JS sub-modules must be syntactically valid.	—	Pass	0.169s
startup.py JS must be syntactically valid.	—	Pass	0.057s
Storm portfolio syntax	—	Pass	0.065s
Trading aggregate submodule syntax	—	Pass	0.059s
Trading blotter submodule syntax	—	Pass	0.066s
Trading curves submodule syntax	—	Pass	0.058s
Trading desk js syntax	—	Pass	0.079s
Trading eod submodule syntax	—	Pass	0.060s
Trading fs01 submodule syntax	—	Pass	0.057s
Trading market submodule syntax	—	Pass	0.066s
Trading port stress submodule syntax	—	Pass	0.060s
Trading stress submodule syntax	—	Pass	0.063s
Aggregate generates	`len(js) > 1000`	Pass	0.000s
Blotter generates	`len(js) > 1000`	Pass	0.000s
Curves generates	`len(js) > 1000`	Pass	0.000s
Eod generates	`len(js) > 1000`	Pass	0.000s
Fs01 generates	`len(js) > 500`	Pass	0.000s
Governance bibliography generates	`len(js) > 500`	Pass	0.000s
Governance docs generates	`len(js) > 500`	Pass	0.000s
Market generates	`len(js) > 1000`	Pass	0.000s
Port stress generates	`len(js) > 1000`	Pass	0.000s
Preloader generates	`len(js) > 1000`	Pass	0.000s
Startup generates	`len(js) > 1000`	Pass	0.000s
Storm sp table generates	`len(js) > 500`	Pass	0.000s
Stress generates	`len(js) > 1000`	Pass	0.000s
Trading desk generates	`len(js) > 50_000`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
blotter/actions.py must call window.refreshMainMapFS01 after close-out.	'window.refreshMainMapFS01' in src	Pass	0.001s
tdApplyFilter must be on window for FS01 cell click to work.	'window.tdApplyFilter' in src	Pass	0.001s
tdRiskCellClick must be on window for blotter filter drill-through.	'window.tdRiskCellClick' in src	Pass	0.001s
_tdPendingFilter must be on window for context-menu → blotter filter.	'window._tdPendingFilter' in blotter_src	Pass	0.001s
refreshMainMapFS01 must be on window for blotter close-out to call it.	'window.refreshMainMapFS01' in td_map_src	Pass	0.000s
_stressStormHint must be set via window.* for cross-panel navigation.	'window._stressStormHint' in hist_src	Pass	0.001s
Stress setup reads window hint	'window._stressStormHint' in stress_src	Pass	0.001s
TradingDeskPanel must export TradingDesk and showTradingDesk.	'window.TradingDesk' in src; 'window.showTradingDesk' in src	Pass	0.001s
Create panel creates blotter view	'createBlotterView' in js	Pass	0.003s
Open panel function exists	'_tdOpenPanel' in js	Pass	0.003s
Open panel sets display flex	"style.display = 'flex'" in js	Pass	0.003s
Pi button calls show panel	'showPanel()' in js	Pass	0.003s
Fallback preloader function must exist for first-open race condition.	'_tdRunPreload' in js	Pass	0.003s
Show panel function exists	'function showPanel()' in js	Pass	0.003s
Show panel reads window preload done	'window._tdPreloadDone' in js	Pass	0.003s
Panel must switch to blotter tab when opened.	"switchTab('blotter')" in js	Pass	0.003s
Generate risk report basic	generate_risk_report(sample_portfolio_data, output_dir=ou...	Pass	0.017s
Generate risk report default	generate_risk_report(sample_portfolio_data, output_dir=ou...	Pass	0.016s
Generate risk report detailed	generate_risk_report(sample_portfolio_data, output_dir=ou...	Pass	0.022s
Diagonal is one	np.allclose(np.diag(C), 1.0)	Pass	0.000s
Longer range -¿ higher off-diagonal correlations.	C_long[mask].mean() > C_short[mask].mean()	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Higher nugget -¿ lower off-diagonal values.	`C_high[mask].mean() < C_low[mask].mean()`	Pass	0.000s
Off diagonal in open interval	`0.0 < C[i, j] < 1.0`	Pass	0.000s
Positive definite	`np.all(eigenvalues > 0)`	Pass	0.005s
Symmetric	`np.allclose(C, C.T)`	Pass	0.000s
Thames positive definite	`np.all(eigenvalues > 0)`	Pass	0.005s
All gauge files corrupt → no storms (but file still written).	`result["total_storms"] == 0`	Pass	0.002s
Lines 151-153: unreadable gauge file is skipped with warning.	`out.exists() and (out / "_index.json").exists(); result["total_storms"] == 1`	Pass	0.003s
Lines 133-134: invalid JSON in storms.json logs warning, continues.	`out.exists() and (out / "_index.json").exists(); result["total_storms"] == 1`	Pass	0.022s
Exactly at 119999 should be band 'A'.	`band == "A"`	Pass	0.000s
Line 159: property value ¡ 120000 returns band 'A'.	`band == "A"`	Pass	0.000s
Line 161: property value = 120000 returns band 'B'.	`band == "B"`	Pass	0.000s
Property value 140000 returns band 'B'.	`band == "B"`	Pass	0.000s
Exactly at 155999 should still be band 'B'.	`band == "B"`	Pass	0.000s
Counts matching files	`_count_dir(tmp_path, 'GAUGE-*.json') == 5`	Pass	0.001s
Empty directory	`_count_dir(tmp_path, '*') == 0`	Pass	0.000s
Returns zero for no matches	`_count_dir(tmp_path, '*.json') == 0`	Pass	0.000s
Returns zero for none dir	`_count_dir(None, '*.json') == 0`	Pass	0.000s
Returns zero for nonexistent dir	`_count_dir(tmp_path / 'nope', '*.json') == 0`	Pass	0.000s
All counterparties have party id	`len(counterparties) > 0; ctpy["CounterpartySet"]["Party"]["PartyID"].startswith("C...`	Pass	0.006s
Cdm structure	`"Party" in ctpy; "PartyID" in ctpy["Party"]` (+1 more)	Pass	0.001s
Custom count	`len(result["data"]) == 3`	Pass	0.001s
Generates counterparty file	`(pipeline_dir / "counterparty.json").exists()`	Pass	0.000s
Lines 263-269: ContactInformation fields mapped.	`mapping.get("primary_email") == "test@bank.com"; mapping.get("city") == "London"`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Lines 280-285: count fields in mapping.	`"counterparty_count" in mapping; mapping["counterparty_count"] == 2`	Pass	0.000s
Minimal data (empty CounterpartySet) returns dict.	`isinstance(mapping, dict)`	Pass	0.000s
Lines 245-289: create_mapping with full data.	`isinstance(mapping, dict); mapping["party_id"] == "CTPY-001"` (+3 more)	Pass	0.000s
Lines 291-292: exception → raises ValueError.	—	Pass	0.000s
Lines 288-289: None values are stripped from result.	`None not in mapping.values()`	Pass	0.000s
Lines 294-301: get_required_fields() returns non-empty list.	`isinstance(fields, list); len(fields) > 0` (+2 more)	Pass	0.000s
Schema is dict	`isinstance(cdm.schema, dict); len(cdm.schema) > 0`	Pass	0.000s
Lines 201-202: empty Counterparties array → error.	`"Counterparties" in errors`	Pass	0.000s
validate({}) returns errors for all required fields.	`isinstance(errors, dict)`	Pass	0.000s
Lines 211-214: invalid role (not Party1/Party2) → error.	`"Counterparties" in errors; any("Invalid" in e for e in errors["Counterparties"])`	Pass	0.000s
Lines 209-210: missing CounterpartyRole → error.	`"Counterparties" in errors; any("CounterpartyRole" in e for e in errors["Counterparti...`	Pass	0.000s
Lines 188-189: missing PartyID → Party errors.	`"Party" in errors; any("PartyID" in e for e in errors["Party"])`	Pass	0.000s
Lines 191-192: missing PartyName → Party errors.	`"Party" in errors; any("PartyName" in e for e in errors["Party"])`	Pass	0.000s
Lines 218-219: missing PartyRef → error.	`"Counterparties" in errors; any("PartyRef" in e for e in errors["Counterparties"])`	Pass	0.000s
Lines 222-225: two CPs with same role → error.	`"Counterparties" in errors`	Pass	0.000s
Valid counterparty data produces empty error dict.	`errors == {}`	Pass	0.000s
Lines 74-75: json.JSONDecodeError in get_counterparty → 500.	`r.status_code == 500; data["status"] == "error"`	Pass	0.002s
Lines 58-59: json.JSONDecodeError → 500.	`r.status_code == 500; data["status"] == "error"`	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Creates output file	`output_file.exists()`	Pass	0.000s
Data is list	`isinstance(result["data"], list); len(result["data"]) == 3`	Pass	0.000s
Default count uses all counterparties	`len(result["data"]) == len(_ALL_COUNTERPARTIES)`	Pass	0.002s
File path is pathlib path	`isinstance(result["file_path"], Path)`	Pass	0.000s
Metadata has total generated	`result["metadata"]["total_generated"] == 4`	Pass	0.000s
Output file is valid json	`"counterparties" in data; "generation_metadata" in data`	Pass	0.002s
Returns dict with required keys	`"data" in result; "file_path" in result` (+1 more)	Pass	0.000s
Zero count returns empty	`result["data"] == []`	Pass	0.000s
Counterparty not found returns empty	`result == []`	Pass	0.002s
Counterparty trace finds master record	`"counterparties" in steps`	Pass	0.003s
Counterparty trace finds prs trades	`len(consumed) >= 1`	Pass	0.002s
Consistent cover	`badge._cellvalues[1][0] == 'CONSISTENT'`	Pass	0.000s
Degraded cover on failures	`badge._cellvalues[1][0] == 'DEGRADED'`	Pass	0.000s
Inconsistent cover on broken chain	`badge._cellvalues[1][0] == 'INCONSISTENT'`	Pass	0.000s
Engines not none	`engine is not None`	Pass	0.000s
Returns dict with engines	`isinstance(engines, dict); "ISDA" in engines` (+2 more)	Pass	0.002s
Default day counter	`curve is not None; curve is not None`	Pass	0.000s
Returns curve and quote	`curve is not None; quote is not None`	Pass	0.000s
Actual360 day counter accepted	`curve is not None`	Pass	0.000s
Quote is simple quote	`isinstance(quote, ql.SimpleQuote)`	Pass	0.000s
Quote value matches input	`quote.value() == pytest.approx(rate, rel=1e-10)`	Pass	0.000s
Survival probability formula	`sp == pytest.approx(math.exp(-lam * 3), rel=0.02)`	Pass	0.000s
Very high hazard rate near zero survival	`s1 == pytest.approx(math.exp(-2.0), rel=0.01)`	Pass	0.000s
Returns hazard curve	`curve is not None`	Pass	0.000s
Survival probability decreases	`sp2 < sp1`	Pass	0.000s
Zero hazard near full survival	`sp > 0.99`	Pass	0.000s
Auto generates id	`"id" in data["meeting"]; len(data["meeting"]["id"]) > 0`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Create returns meeting	`r.status_code == 200`; `data["status"] == "success"` (+1 more)	Pass	0.002s
Custom attendees become participants	`"Alice Smith" in names`	Pass	0.001s
Custom title used	`data["meeting"]["title"] == "Quarterly Review"`	Pass	0.002s
Default chair populated	`data["meeting"]["chair"] == "Johnny Mattimore"`	Pass	0.001s
Default participants created	`len(participants) >= 1`	Pass	0.001s
No json body uses defaults	`r.status_code == 200`; `r.get_json()["status"] == "success"`	Pass	0.002s
Custom max notifications	`ns.max_visible == 8`	Pass	0.000s
Custom timeout	`ns.timeout == 1000`	Pass	0.000s
Invalid position defaults to top right	`ns.position.value == "top-right"`	Pass	0.000s
Valid position	`ns.position.value == "bottom-left"`	Pass	0.000s
create_pdf_report(None) calls collect_all().	—	Pass	0.017s
Creates audit dir	`audit.is_dir()`; `result.exists()`	Pass	0.016s
Creates pdf	`result.exists()`; `result.stat().st_size > 0`	Pass	0.017s
Creates pdf empty data	`result.exists()`	Pass	0.007s
Main prints summary	`'Generating Model Risk Governance Report' in captured`; `'models in inventory' in captured` (+5 more)	Pass	0.017s
Integration: main() with actual data loading against empty dirs.	`out.exists()`	Pass	0.018s
PDF file is created on disk.	`out.exists()`	Pass	0.022s
Generated file has PDF magic bytes.	`out.read_bytes()[:4] == b'%PDF'`	Pass	0.393s
PDF has reasonable size (multi-page report).	`out.stat().st_size > 5000`	Pass	0.033s
PDF renders when no steps have been recorded.	`out.exists()`; `out.read_bytes()[:4] == b'%PDF'`	Pass	0.024s
PDF generates correctly when pipeline has stale/missing steps.	`out.exists()`; `out.read_bytes()[:4] == b'%PDF'` (+1 more)	Pass	0.037s
PDF renders when lineage_results is empty (tests not run).	`out.exists()`; `out.stat().st_size > 5000`	Pass	0.025s
create_pdf_report() returns a Path object.	`isinstance(result, Path)`	Pass	0.040s
Creates registry	`create_registry(str(tmp_path)) is not None`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
With file overrides	`registry.get_gauge_loader().count() == 2`	Pass	0.001s
Test create_registry helper.	`registry.data_dir == populated_data_dir; registry is not None` (+1 more)	Pass	0.002s
Test create_registry with file overrides.	`loader.count() == 0`	Pass	0.002s
Auto generated id starts with storm	`storm.storm_id.startswith("STORM-")`	Pass	0.000s
Decay kernel stored	`storm.decay_kernel == DecayKernel.EXPONENTIAL`	Pass	0.000s
Decay parameter stored	`storm.decay_parameter == pytest.approx(1.2)`	Pass	0.000s
Default num track points	`len(storm.track) == 25`	Pass	0.000s
Default start time is recent	`before <= storm.start_time <= after`	Pass	0.000s
Duration hours stored	`storm.duration_hours == pytest.approx(36.0)`	Pass	0.000s
Explicit id preserved	`storm.storm_id == "MY-STORM-001"`	Pass	0.000s
Explicit name	`storm.name == "Ciara"`	Pass	0.000s
Explicit start time	`storm.start_time == t`	Pass	0.000s
Footprint km stored	`storm.footprint_km == pytest.approx(60.0)`	Pass	0.000s
Num track points	`len(storm.track) == 10`	Pass	0.000s
Peak intensity stored	`storm.peak_intensity == pytest.approx(75.0)`	Pass	0.000s
Returns storm object	`isinstance(storm, Storm)`	Pass	0.000s
Speed is positive	`storm.speed_kmh > 0`	Pass	0.000s
Track end time equals duration	`storm.track[-1].time_hours == pytest.approx(24.0)`	Pass	0.000s
Track longitude moves from start to end	`lons[0] < lons[-1]`	Pass	0.000s
Track points are trackpoint instances	`isinstance(tp, TrackPoint)`	Pass	0.000s
Track start time is zero	`storm.track[0].time_hours == pytest.approx(0.0)`	Pass	0.000s
Track time increases	`all(times[i] <= times[i + 1] for i in range(len(times) - 1))`	Pass	0.000s
Actual360 day counter produces curve	`0.0 < sp <= 1.0`	Pass	0.000s
Implied hazard rate formula	`sp5 == pytest.approx(expected_5yr, rel=0.01)`	Pass	0.000s
Near zero survival positive hazard	`s1 < 0.9`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Single point high risk ts	`curve is not None`	Pass	0.000s
Single point term structure	`curve is not None`	Pass	0.000s
Survival strictly positive	`sp > 0.0`	Pass	0.000s
Custom day counter	`curve is not None`	Pass	0.000s
Default day counter	`curve is not None; curve is not None`	Pass	0.000s
Full survival at start	`abs(sp - 1.0) < 1e-6`	Pass	0.001s
Returns curve	`curve is not None`	Pass	0.000s
Line 83: final_survival ¡= 0 → uses 0.05 default hazard rate.	`curve is not None`	Pass	0.000s
Line 83: final_survival ¿= 1 → uses 0.05 default hazard rate.	`curve is not None`	Pass	0.005s
Excellent credit 800 plus	`isinstance(result, list)`	Pass	0.000s
Fair credit 580 to 669	`isinstance(result, list)`	Pass	0.000s
Good credit 670 to 739	`isinstance(result, list)`	Pass	0.000s
Poor credit under 580	`isinstance(result, list)`	Pass	0.000s
Very good credit 740 to 799	`isinstance(result, list)`	Pass	0.000s
Excellent 800 plus	`isinstance(result, list)`	Pass	0.000s
Fair 580 to 669	`isinstance(result, list)`	Pass	0.000s
Good 670 to 739	`isinstance(result, list)`	Pass	0.000s
Poor under 580	`isinstance(result, list)`	Pass	0.000s
Very good 740 to 799	`isinstance(result, list)`	Pass	0.000s
Gauges returned by list endpoint should have valid timeseries responses.	`response.status_code == 200; ts_response.status_code in [200, 404]`	Pass	0.002s
Gauge hazard endpoint should never return 500 for listed gauges.	`hz_response.status_code != 500`	Pass	0.002s
Properties returned by list endpoint should have flood data if propertyts exists.	`response.status_code == 200; flood_response.status_code in [200, 404]`	Pass	0.002s
Every property in propertyhc should have all fields the pricer needs.	`field in pc`	Pass	0.004s
Elevation should be consistent between propertyts and propertyhc.	`abs(hc_elev - pts_elev) < 0.01`	Pass	0.008s
flood_count in propertyhc should match flooded event count in propertyts.	`pc.get("flood_count") == pts_flooded`	Pass	0.008s
Flood zone should be consistent between propertyts and propertyhc.	`pts_zone == hc_zone`	Pass	0.008s

Test Description	Acceptance Criteria	Result	Time
Nearest gauge IDs should carry through from propertyts to propertyhc.	`hc_gauges.issubset(pts_gauges)`	Pass	0.008s
crud.py lines 63-67: gauge not found -¿ 404.	`r.status_code == 404;` `r.get_json()["status"] == "error"`	Pass	0.029s
crud.py lines 59-72: valid gauge retrieval.	`r.status_code == 200;` `data["status"] == "success"` (+1 more)	Pass	0.031s
crud.py lines 51-53: loader error -¿ 500.	`r.status_code == 500;` `r.get_json()["status"] == "error"`	Pass	0.028s
crud.py line 34: /list_gauges alias works.	`r.status_code == 200;` `r.get_json()["count"] == 1`	Pass	0.029s
crud.py lines 37-38: OPTIONS returns ok.	`r.status_code == 200`	Pass	0.029s
crud.py lines 40-50: successful gauge listing.	`r.status_code == 200;` `data["status"] == "success"` (+2 more)	Pass	0.029s
hd present but daily_observations missing -¿ same paragraph.	`any('No historical daily data' in t for t in texts)`	Pass	0.000s
daily_observations = [] -¿ falsy, caught by first guard.	`any('No historical daily data' in t for t in texts)`	Pass	0.000s
Normal flow: valid obs -¿ chart + table elements returned.	`len(elements) > 3`	Pass	0.091s
ValueError/KeyError in date parse loop -¿ skipped.	`len(elements) > 0`	Pass	0.088s
Invalid date in latest observation -¿ current_month = None.	`len(elements) > 0`	Pass	0.087s
No obs in same month -¿ same_month_levels = all_levels.	`len(elements) > 0`	Pass	0.086s
No hd data -¿ 'No historical daily data' paragraph.	`any('No historical daily data' in t for t in texts)`	Pass	0.000s
level is None -¿ skipped, does not crash.	`len(elements) > 0`	Pass	0.088s
Boe base shown	`isinstance(result, list)`	Pass	0.000s
Current payment shown	`isinstance(result, list)`	Pass	0.000s
Has status header	`any("Status" in t or "Mortgage" in t for t in texts)`	Pass	0.000s
Has table	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
Interest paid shown	`isinstance(result, list)`	Pass	0.000s
Last payment date shown	`isinstance(result, list)`	Pass	0.000s
Lastatus shown	`isinstance(result, list)`	Pass	0.000s
Lending rate shown	`isinstance(result, list)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
No current status returns message	`any("No current status" in t for t in texts)`	Pass	0.000s
No mortgage returns message	`any("No mortgage" in t for t in texts)`	Pass	0.000s
Outstanding balance shown	`isinstance(result, list)`	Pass	0.000s
Principal paid shown	`isinstance(result, list)`	Pass	0.000s
Total payments shown	`isinstance(result, list)`	Pass	0.000s
With mortgage returns list	`isinstance(result, list) and len(result) > 0`	Pass	0.000s
Returns empty history when no snapshots exist.	`resp.status_code == 200; data['history'] == []`	Pass	0.001s
Missing gauge_id returns 400.	`resp.status_code == 400`	Pass	0.000s
After EOD, returns term structure history for that gauge.	`resp.status_code == 200; data['gauge_id'] == 'GAUGE-001'` (+1 more)	Pass	0.041s
POST hazard-term-structure/commit returns 500 on error.	`resp.status_code == 500`	Pass	0.001s
curve-history returns 500 on engine error (lines 82-84).	`resp.status_code == 500`	Pass	0.001s
GET hazard-term-structure returns 500 on engine error.	`resp.status_code == 500`	Pass	0.001s
GET yield-curve returns 500 on engine error.	`resp.status_code == 500`	Pass	0.001s
pnl-series returns 500 on engine error (lines 36-38).	`resp.status_code == 500`	Pass	0.001s
POST hazard-term-structure/reset returns 500 on error.	`resp.status_code == 500`	Pass	0.001s
POST yield-curve/reset returns 500 on engine error.	`resp.status_code == 500`	Pass	0.001s
POST yield-curve returns 500 on engine error.	`resp.status_code == 500`	Pass	0.001s
POST hazard-term-structure/commit with null body -¿ 400 (line 282).	`resp.status_code == 400`	Pass	0.000s
POST yield-curve/commit with null body -¿ 400 (line 145).	`resp.status_code == 400`	Pass	0.000s
POST hazard-term-structure with null body -¿ 400 (line 245).	`resp.status_code == 400`	Pass	0.000s
POST yield-curve with null body -¿ 400 (line 115).	`resp.status_code == 400`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Curves tab should show all gauges on one chart.	'tdRenderAllCurves' in js; 'tdCurveAllData' in js (+1 more)	Pass	0.000s
Curves tab should have alert/warning/severe trigger selector.	'td-curve-trigger' in js; 'severe' in js (+2 more)	Pass	0.000s
Curves tab should use HSL for distinct line colours.	'hsl(' in js	Pass	0.000s
Curves sub-module should render without errors.	len(js) > 0	Pass	0.000s
Legend should strip 'Thames ' prefix for brevity.	'Thames' in js	Pass	0.000s
Curves tab should fetch from market-state endpoint.	'/api/v1/trading/market-state' in js	Pass	0.000s
Good dti under 28	isinstance(result, list)	Pass	0.001s
High dti 36 to 43	isinstance(result, list)	Pass	0.000s
Moderate dti 28 to 36	isinstance(result, list)	Pass	0.000s
No payment dti unknown	isinstance(result, list)	Pass	0.000s
Very high dti over 43	isinstance(result, list)	Pass	0.000s
Empty daily pnl returns drawing	isinstance(result, Drawing)	Pass	0.000s
Mixed positive negative renders	isinstance(result, Drawing)	Pass	0.000s
Negative daily pnl renders	isinstance(result, Drawing)	Pass	0.000s
Commercial higher damage	com_ratio >= res_ratio	Pass	0.000s
Damage capped	calculate_damage_ratio(10.0) <= 1.0	Pass	0.000s
Damage increases with depth	ratios == sorted(ratios)	Pass	0.000s
No damage no flood	calculate_damage_ratio(0.0) == 0.0	Pass	0.000s
GaugeHC file must have required structure.	'hazard_curves' in data; not missing	Pass	0.001s
Gaugets files must have required structure.	'flood_simulation' in data; 'readings' in sim (+1 more)	Pass	0.002s
PropertyHC file must have required structure.	'property_hazard_curves' in data; 'metadata' in data (+2 more)	Pass	0.001s
PropertyTS files must have required fields.	not missing; not missing_loc (+3 more)	Pass	0.003s
Every step in manifest must appear in the static dependency graph.	len(missing) == 0	Pass	0.001s
Data lineage file should exist after a port run.	lineage_path.exists()	Pass	0.000s
data_lineage.json must exist after any port run.	"steps" in data; len(data["steps"]) > 0	Pass	0.000s
Recorded output hashes must match current files on disk.	—	Pass	11.220s

Test Description	Acceptance Criteria	Result	Time
Data summary contains errors count	`"errors" in dl.get_data_summary()["gauge"]`	Pass	0.002s
Data summary contains warnings count	`"warnings" in dl.get_data_summary()["gauge"]`	Pass	0.001s
Data summary loaded flag false when no file	`summary["gauge"]["loaded"] is False`	Pass	0.000s
Data summary loaded flag true for gauge	`dl.get_data_summary()["gauge"]["loaded"] is True`	Pass	0.001s
Get data summary empty before load	`DataLoader(input_dir=tmp_path).get_data_summary() == {}`	Pass	0.000s
Get data summary returns dict	`isinstance(dl.get_data_summary(), dict)`	Pass	0.001s
Get validation summary empty before load	`DataLoader(input_dir=tmp_path).get_validation_summary() =...`	Pass	0.000s
Get validation summary is copy	`s1 is not s2`	Pass	0.001s
Get validation summary returns dict	`isinstance(dl.get_validation_summary(), dict)`	Pass	0.002s
Is data complete depends only on gauge	`dl.is_data_complete() is False`	Pass	0.002s
Is data complete false before load	`DataLoader(input_dir=tmp_path).is_data_complete() is False`	Pass	0.000s
Is data complete false when no gauge	`dl.is_data_complete() is False`	Pass	0.000s
Is data complete true with gauge	`dl.is_data_complete() is True`	Pass	0.002s
Validation summary has gauge key	`"gauge" in dl.get_validation_summary()`	Pass	0.002s
Init with input dir	`dl.input_dir == tmp_path`	Pass	0.000s
Init with string path converts to path	`isinstance(dl.input_dir, Path); dl.input_dir == tmp_path`	Pass	0.000s
Loaded data initialised	`isinstance(dl.loaded_data, LoadedData)`	Pass	0.000s
Private loaders created	`dl._gauge_loader is not None; dl._property_loader is not None` (+1 more)	Pass	0.000s
Validation results empty at init	`dl._validation_results == {}`	Pass	0.000s
Counterparty data loaded	`result.counterparty_data is not None`	Pass	0.002s
Empty dir returns loaded data	`isinstance(result, LoadedData)`	Pass	0.000s
Gauge data count matches records	`result.gauge_data["count"] == len(result.gauge_data["item...`	Pass	0.001s
Gauge data has items key	`"items" in result.gauge_data`	Pass	0.001s
Gaugehd dir counted	`result.gaugehd_count == 1`	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Gaugets dir counted	`result.gaugets_count == 1`	Pass	0.002s
Gaugets dir multiple files counted	`result.gaugets_count == 3`	Pass	0.002s
Hazard curves loaded	`result.hazard_data is not None`	Pass	0.002s
Lookups built after load	`result.mortgage_lookup is not None; result.gauge_flood_info is not None`	Pass	0.001s
No gaugehd dir count stays zero	`result.gaugehd_count == 0`	Pass	0.002s
No gaugets dir count stays zero	`result.gaugets_count == 0`	Pass	0.001s
No propertyts dir count stays zero	`result.propertyts_count == 0`	Pass	0.001s
Property hazard curves loaded	`result.property_hazard_data is not None`	Pass	0.002s
Property hazard summary fields accessible	`"summary" in result.property_hazard_data`	Pass	0.002s
Propertyts dir counted	`result.propertyts_count >= 1`	Pass	0.002s
Storm data list format	`result.storm_data is not None`	Pass	0.003s
Storm data loaded dict format	`result.storm_data is not None`	Pass	0.002s
Storm data unknown format	`result.storm_data is not None`	Pass	0.002s
Storm data with items key	`result.storm_data is not None`	Pass	0.002s
With gauge data loaded	`result.gauge_data is not None`	Pass	0.001s
With mortgage data loaded	`result.mortgage_data is not None`	Pass	0.002s
With property data loaded	`result.property_data is not None`	Pass	0.002s
Collect all keys	`expected_keys == set(result.keys())`	Pass	0.004s
List audit files	`len(result) == 1; result[0]['name'] == 'report.pdf'` (+2 more)	Pass	0.001s
List audit files missing dir	`result == []`	Pass	0.000s
List sensitivity generators	`result == ['hazard']`	Pass	0.001s
List sensitivity generators no dir	`result == []`	Pass	0.000s
Load audit log	`len(result) == 1`	Pass	0.001s
Load audit log dict fallback	`result == []`	Pass	0.001s
Load bcbs	`'principles' in result`	Pass	0.001s
Load coverage	`result == pytest.approx(82.5)`	Pass	0.001s
Load coverage corrupt	`result is None`	Pass	0.001s
Load coverage missing	`result is None`	Pass	0.000s
Load inventory	`result['models'][0]['model_id'] == 'M-TEST'`	Pass	0.001s
Load inventory bad json	`result == {}`	Pass	0.001s
Load inventory missing	`result == {}`	Pass	0.000s
Load junit	`result['total'] == 100; result['failed'] == 2` (+4 more)	Pass	0.001s
Load junit corrupt	`result['total'] == 0`	Pass	0.001s
Load junit missing	`result['total'] == 0`	Pass	0.000s
Load meetings	`len(result) == 1`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Load meetings dict fallback	`result == []`	Pass	0.001s
Load raci	`'roles' in result`	Pass	0.002s
Both empty	`isinstance(self._page().generate_elements({}, {}), list)`	Pass	0.000s
Empty property does not crash	`isinstance(self._page().generate_elements({}), list)`	Pass	0.000s
Has data header	`any("Data" in t for t in texts)`	Pass	0.001s
Has table	`any(isinstance(e, Table) for e in result)`	Pass	0.001s
No sections in property	`isinstance(result, list)`	Pass	0.000s
Returns list	`isinstance(self._page().generate_elements(_make_property()), list); isinstance(r, list)`	Pass	0.001s
Returns non empty	`len(self._page().generate_elements(_make_property())) > 0`	Pass	0.001s
With mortgage data	`isinstance(result, list) and len(result) > 0`	Pass	0.001s
Invalid result fields	`result.is_valid is False; "file not found" in result.errors`	Pass	0.000s
Is named tuple	`isinstance(result, tuple)`	Pass	0.000s
Tuple unpacking	`is_valid is True`	Pass	0.000s
Valid result fields	`result.is_valid is True; result.warnings == ["minor issue"]` (+2 more)	Pass	0.000s
Datetime encoded as isoformat	`"2024-06-15" in result`	Pass	0.000s
Nested numpy in dict	`result["a"] == 1; abs(result["b"] - 2.5) < 1e-6` (+1 more)	Pass	0.000s
Numpy float32 encoded	`abs(result - 1.5) < 0.01`	Pass	0.000s
Numpy float64 encoded as float	`abs(result - 3.14) < 1e-6`	Pass	0.000s
Numpy int32 encoded	`result == 7`	Pass	0.000s
Numpy int64 encoded as int	`result == 99; isinstance(result, int)`	Pass	0.000s
Numpy ndarray encoded as list	`result == [1, 2, 3]`	Pass	0.000s
Unknown type raises type error	—	Pass	0.000s
Line 59-60: datetime -¿ isoformat.	`"2024-06-01" in result`	Pass	0.000s
Line 63-64: np.floating -¿ float.	`"2.71" in result`	Pass	0.000s
Line 61-62: np.integer -¿ int.	`result == "99"`	Pass	0.000s
Line 65-66: np.ndarray -¿ list.	`result == [10, 20]`	Pass	0.000s
Line 67: super().default() -¿ TypeError for unknown.	—	Pass	0.000s
Exponential value	`DecayKernel.EXPONENTIAL.value == "exponential"`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Gaussian value	`IntensityProfile.GAUSSIAN.value == "gaussian"; DecayKernel.GAUSSIAN.value == "gaussian"`	Pass	0.000s
Linear value	`DecayKernel.LINEAR.value == "linear"`	Pass	0.000s
Add decision auto generates id	`r.status_code == 200; decisions[0]["id"] == "D-001"`	Pass	0.001s
Add decision unknown meeting returns 404	`r.status_code == 404`	Pass	0.001s
Delete decision	`r.status_code == 200; len(r.get_json()["meeting"]["decisions"]) == 1`	Pass	0.004s
Delete decision unknown meeting returns 404	`r.status_code == 404`	Pass	0.001s
Second decision increments id	`ids == ["D-001", "D-002"]`	Pass	0.002s
Update decision	`r.status_code == 200; r.get_json()["meeting"]["decisions"][0]["description"] ==...`	Pass	0.002s
Update missing decision returns 404	`r.status_code == 404`	Pass	0.001s
Add decision auto ids	`"D-001" in ids; "D-002" in ids`	Pass	0.002s
Add decision missing meeting returns 404	`r.status_code == 404`	Pass	0.002s
Add decision returns 200	`r.status_code == 200; len(meeting["decisions"]) == 2`	Pass	0.002s
Delete decision missing meeting returns 404	`r.status_code == 404`	Pass	0.002s
Delete decision returns 200	`r.status_code == 200; len(r.get_json()["meeting"]["decisions"]) == 0`	Pass	0.002s
Update decision not found returns 404	`r.status_code == 404`	Pass	0.002s
Update decision returns 200	`r.status_code == 200; decisions[0]["description"] == "Updated decision"`	Pass	0.002s
Delete is idempotent — filtering out a missing ID is still 200.	`r.status_code == 200`	Pass	0.002s
Delete returns success	`r.status_code == 200; r.get_json()["status"] == "success"`	Pass	0.003s
Deleted ref not in list	`ref_id not in ids`	Pass	0.004s
Deleted doc not in list	`doc_id not in ids`	Pass	0.003s

Test Description	Acceptance Criteria	Result	Time
Not found returns 404	`r.status_code == 404; r.status_code == 404`	Pass	0.001s
Valid delete returns success	`r.status_code == 200; data["status"] == "success"`	Pass	0.003s
Delete save failure returns 500	`r.status_code == 500; data["status"] == "error"`	Pass	0.030s
Gaugehd references gauges	`gid in gauge_ids`	Pass	0.025s
Gaugets reads gauge	`gid in gauge_ids`	Pass	0.002s
Hazard reads gauge and storms	`len(hc_curves) > 0; gid in gauge_ids`	Pass	0.001s
Mortgage reads property	`mid in prop_ids`	Pass	0.003s
Propertyhc references properties	`pid in prop_ids`	Pass	0.004s
Propertyts uses nearest gauges	`ng["gauge_id"] in gauge_ids`	Pass	0.001s
Every input declared in STEP_IO must be produced by a transitive	`producers`	Pass	0.000s
When hazard mutates gaugets/ to a different hash, propertyts must	`producer == "hazard"`	Pass	0.000s
Commercial higher damage than residential	`com_damage > res_damage`	Pass	0.000s
Floor level reduces damage	`damages[0] >= damages[3]`	Pass	0.000s
Large depth all near one	`np.all(damages > 0.5)`	Pass	0.000s
Output clipped to 0 1	`np.all((damages >= 0) & (damages <= 1))`	Pass	0.000s
Returns array same length	`len(damages) == len(properties)`	Pass	0.002s
Zero depths all zero damage	—	Pass	0.053s
Line 150: commercial factor = 1.2.	`result[1] > result[0]`	Pass	0.000s
Lines 159-160: adjusted_depths = max(depths - floor_levels, 0).	`result[3] == pytest.approx(0.0)`	Pass	0.000s
Full function returns correct length	`len(result) == len(properties)`	Pass	0.000s
Line 151: industrial factor = 0.9.	`result[2] < result[0]`	Pass	0.000s
Line 165: np.clip(final_damage, 0, 1).	`np.all(result <= 1.0); np.all(result >= 0.0)`	Pass	0.000s
Line 155: .get(str(pt).lower(), 1.0) — 'other' defaults to 1.0.	`result[4] == pytest.approx(result[0])`	Pass	0.000s
Zero depths all zero	—	Pass	0.000s
Annual probability equals count over storms	`abs(prob - 5 / 100) < 1e-6`	Pass	0.088s
Only severe key in depth thresholds	`list(result["depth_thresholds"].keys()) == ["severe"]`	Pass	0.089s
Return period none when zero floods	`rp is None`	Pass	0.087s

Test Description	Acceptance Criteria	Result	Time
Return period positive when floods present	`rp is not None; rp > 0`	Pass	0.087s
Deep flood -¿ damage ratio approaches 1.	`result > 0.5`	Pass	0.163s
Positive depth -¿ positive damage ratio.	`0.0 < result <= 1.0`	Pass	0.163s
0m depth -¿ 0 damage.	`gen._depth_to_damage(0.0) == 0.0`	Pass	0.164s
Delegates to scalar depth damage	`isinstance(result, float); result > 0`	Pass	0.164s
Zero depth	`gen._depth_to_damage(0.0) == 0.0`	Pass	0.164s
Line 80: after loop exhausted, gauges_warning ¡ 3 → 'baseline'.	`_derive_intensity_category(-1, 2) == "baseline"`	Pass	0.000s
Negative severe with five warnings returns moderate	`_derive_intensity_category(-1, 5) == "moderate"`	Pass	0.000s
Lines 78-79: gauges_severe ¡ 0 (doesn't match any threshold),	`_derive_intensity_category(-1, 3) == "moderate"`	Pass	0.000s
Negative severe zero warnings returns baseline	`_derive_intensity_category(-1, 0) == "baseline"`	Pass	0.000s
When severe ¿= 1, warning count is irrelevant.	`_derive_intensity_category(1, 10) == "moderate"`	Pass	0.000s
gauges_severe=0 matches (0, 'baseline') in the severity table.	`_derive_intensity_category(0, 0) == "baseline"`	Pass	0.000s
Both missing returns na	`page._derive_tenor("", "") == "N/A"`	Pass	0.000s
Five year tenor	`page._derive_tenor("2024-01-01", "2029-01-01") == "5Y"`	Pass	0.000s
Invalid date format returns na	`page._derive_tenor("not-a-date", "2026-01-01") == "N/A"`	Pass	0.000s
Missing end returns na	`page._derive_tenor("2024-01-01", "") == "N/A"`	Pass	0.000s
Missing start returns na	`page._derive_tenor("", "2026-01-01") == "N/A"`	Pass	0.000s
Valid dates returns years	`page._derive_tenor("2024-01-01", "2026-01-01") == "2Y"`	Pass	0.000s
Empty property info returns valid	`result in ["Residential", "Buy-to-Let", "Second Home", "H...`	Pass	0.000s
Multi family returns btl or residential	`result in ["Buy-to-Let", "Residential"]`	Pass	0.000s
Rental history returns btl or residential	`result in ["Buy-to-Let", "Residential"]`	Pass	0.000s
Rental income returns btl or residential	`result in ["Buy-to-Let", "Residential"]`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Vacant property	`result in ["Buy-to-Let", "Second Home", "Residential"]`	Pass	0.000s
Buy to let returns investment or primary	`result in ["Investment", "PrimaryResidence"]`	Pass	0.000s
Holiday home returns second residence	`_determine_occupancy_type("Holiday Home") == "SecondResid...`	Pass	0.000s
Residential returns primary or second	`result in ["PrimaryResidence", "SecondResidence"]`	Pass	0.000s
Second home returns second residence	`_determine_occupancy_type("Second Home") == "SecondReside...`	Pass	0.000s
Gauge IDs should be derived from location, not random UUIDs.	—	Skip	0.000s
Haversine known distance	`6.0 < distance < 8.0`	Pass	0.000s
Haversine same point	`calculate_distance_km(lat, lon, lat, lon) == pytest.appro...`	Pass	0.000s
Haversine teddington to richmond	`2.5 < distance < 4.0`	Pass	0.000s
Diagonal zero	`np.allclose(np.diag(D), 0.0)`	Pass	0.000s
51.5N 0.0 -¿ 51.5N 1.0E is approximately 69km.	`65.0 < d < 75.0`	Pass	0.000s
Positive off diagonal	`D[i, j] > 0.0`	Pass	0.000s
Shape	`D.shape == (n, n)`	Pass	0.000s
Symmetric	`np.allclose(D, D.T)`	Pass	0.000s
Thames gauges span ~80km corridor (Reading to Purfleet).	`10.0 < max_d < 120.0`	Pass	0.000s
Thames n gauges	`thames_model.n_gauges == 52`	Pass	0.000s
At zero is max	`RiskAssessor.calculate_distance_risk_factor(0.0) == 1.0`	Pass	0.000s
Close band	`RiskAssessor.calculate_distance_risk_factor(0.3) == 0.8`	Pass	0.000s
Far band	`RiskAssessor.calculate_distance_risk_factor(3.0) == 0.2`	Pass	0.000s
Medium band	`RiskAssessor.calculate_distance_risk_factor(1.5) == 0.4`	Pass	0.000s
Monotone decrease	`all(factors[i] >= factors[i + 1] for i in range(len(facto...`	Pass	0.000s
Near band	`RiskAssessor.calculate_distance_risk_factor(0.7) == 0.6`	Pass	0.000s
Negative returns default	`RiskAssessor.calculate_distance_risk_factor(-1.0) == 0.5`	Pass	0.000s
None returns default	`RiskAssessor.calculate_distance_risk_factor(None) == 0.5`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Very close	`RiskAssessor.calculate_distance_risk_factor(0.05) == 1.0`	Pass	0.000s
Very far	`RiskAssessor.calculate_distance_risk_factor(10.0) == 0.1`	Pass	0.000s
Property 50m from gauge with strong storm must flood.	`depth > 0; ret > 0.98`	Pass	0.000s
Property 50km from gauge with moderate storm should not flood.	`depth == 0.0; ret < 0.01`	Pass	0.000s
More distant properties receive less water — flood depth must	`depths[i] <= depths[i - 1] + 1e-6`	Pass	0.002s
Verify retention factor at key distances with 10km length scale.	`abs(actual - expected) < 0.001`	Pass	0.000s
At typical Thames distances (200-800m), check flood/no-flood	`10 <= ratio <= 90; n_flooded < n_storms`	Pass	0.004s
Close distance label	`"Very close to Thames" in str(elements)`	Pass	0.000s
Zero distance highest risk	`"On Thames River" in str(elements)`	Pass	0.000s
STORM-d1a damage for PROP-A uses only that storm's damage_ratio.	`prop_a["damage_amount"] == pytest.approx(expected_damage)`	Pass	0.004s
Query STORM-d1a: only PROP-A (depth¿0) appears; PROP-B depth=0 excluded.	`r.status_code == 200; "PROP-A" in prop_ids` (+1 more)	Pass	0.004s
Total damage is different for each storm in the doublet.	`total2 > total1`	Pass	0.005s
STORM-d1b causes more damage to PROP-A (deeper flood).	`dmg2 > dmg1`	Pass	0.005s
Query STORM-d1b: both PROP-A and PROP-B have positive depth.	`r.status_code == 200; "PROP-A" in prop_ids` (+1 more)	Pass	0.004s
Response storm_id matches the queried storm.	`data["storm_id"] == "STORM-d1b"`	Pass	0.004s
Each storm computes its own damage_ratio from its own flood depth.	`"damage_ratio" in event; 0.0 <= event["damage_ratio"] <= 1.0`	Pass	0.168s
Doublet events have distinct storm ids	`"STORM-d001a" in event_storm_ids; "STORM-d001b" in event_storm_ids`	Pass	0.167s
Doublet produces two flood events	`result is not None; prop_file.exists()` (+1 more)	Pass	0.172s
floods_at_nearest_gauge == 2 for doublet sequence.	`result["summary"]["floods_at_nearest_gauge"] == 2`	Pass	0.166s
File not found returns 404	`r.status_code == 404`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Non pdf served as attachment	`r.status_code == 200; "attachment" in r.headers.get("Content-Disposition", "")`	Pass	0.001s
Path traversal returns 403	`r.status_code in (403, 404)`	Pass	0.001s
Valid pdf served inline	`r.status_code == 200`	Pass	0.012s
Not found returns 404	`r.status_code == 404; r.status_code == 404`	Pass	0.001s
Valid download returns 200	`r.status_code == 200`	Pass	0.002s
File missing on disk returns 404	`r.status_code == 404; "not found" in data["message"].lower()`	Pass	0.028s
All rates below one	`rate < 1.0`	Pass	0.000s
All rates positive	`rate > 0`	Pass	0.000s
Has all five zones	`set(EA_FLOOD_ZONE_RATES.keys()) == self.EXPECTED_ZONES`	Pass	0.000s
Rate is float[zone 1] [zone='Zone 1']	`isinstance(EA_FLOOD_ZONE_RATES[zone], float)`	Pass	0.000s
Rate is float[zone 2] [zone='Zone 2']	`isinstance(EA_FLOOD_ZONE_RATES[zone], float)`	Pass	0.000s
Rate is float[zone 3] [zone='Zone 3']	`isinstance(EA_FLOOD_ZONE_RATES[zone], float)`	Pass	0.000s
Rate is float[zone 3a] [zone='Zone 3a']	`isinstance(EA_FLOOD_ZONE_RATES[zone], float)`	Pass	0.000s
Rate is float[zone 3b] [zone='Zone 3b']	`isinstance(EA_FLOOD_ZONE_RATES[zone], float)`	Pass	0.000s
Zone 3 (generic) should have same rate as Zone 3a.	`EA_FLOOD_ZONE_RATES['Zone 3'] == EA_FLOOD_ZONE_RATES['Zon...`	Pass	0.000s
Zone ordering 2 gt 1	`EA_FLOOD_ZONE_RATES['Zone 2'] > EA_FLOOD_ZONE_RATES['Zone...`	Pass	0.000s
Zone ordering 3a gt 2	`EA_FLOOD_ZONE_RATES['Zone 3a'] > EA_FLOOD_ZONE_RATES['Zon...`	Pass	0.000s
Zone ordering 3b gt 3a	`EA_FLOOD_ZONE_RATES['Zone 3b'] > EA_FLOOD_ZONE_RATES['Zon...`	Pass	0.000s
Boundary parametrized[0.0-zone 3b] [offset=0.0, expected_zone='Zone 3b']	`_ea_zone_from_elevation({'vertical_offset': offset}) == e...`	Pass	0.000s
Boundary parametrized[0.25-zone 3b] [offset=0.25, expected_zone='Zone 3b']	`_ea_zone_from_elevation({'vertical_offset': offset}) == e...`	Pass	0.000s
Boundary parametrized[0.5-zone 3a] [offset=0.5, expected_zone='Zone 3a']	`_ea_zone_from_elevation({'vertical_offset': offset}) == e...`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Boundary parametrized[1.0-zone 3a] [offset=1.0, expected_zone='Zone 3a']	`_ea_zone_from_elevation({'vertical_offset': offset}) == e...`	Pass	0.000s
Boundary parametrized[1.5-zone 2] [offset=1.5, expected_zone='Zone 2']	`_ea_zone_from_elevation({'vertical_offset': offset}) == e...`	Pass	0.000s
Boundary parametrized[10.0-zone 1] [offset=10.0, expected_zone='Zone 1']	`_ea_zone_from_elevation({'vertical_offset': offset}) == e...`	Pass	0.000s
Boundary parametrized[2.5-zone 2] [offset=2.5, expected_zone='Zone 2']	`_ea_zone_from_elevation({'vertical_offset': offset}) == e...`	Pass	0.000s
Boundary parametrized[3.0-zone 1] [offset=3.0, expected_zone='Zone 1']	`_ea_zone_from_elevation({'vertical_offset': offset}) == e...`	Pass	0.000s
The EAFloodZone field generator should use _ea_zone_from_elevation.	`result == 'Zone 3b'; result == 'Zone 1'`	Pass	0.000s
Missing offset defaults to zone 1	`_ea_zone_from_elevation({}) == 'Zone 1'`	Pass	0.000s
Zone 1 at 30	`_ea_zone_from_elevation({'vertical_offset': 3.0}) == 'Zon...`	Pass	0.000s
Zone 1 at large offset	`_ea_zone_from_elevation({'vertical_offset': 50.0}) == 'Zo...`	Pass	0.000s
Zone 2 at 15	`_ea_zone_from_elevation({'vertical_offset': 1.5}) == 'Zon...`	Pass	0.000s
Zone 2 at 29	`_ea_zone_from_elevation({'vertical_offset': 2.9}) == 'Zon...`	Pass	0.000s
Zone 3a at 05	`_ea_zone_from_elevation({'vertical_offset': 0.5}) == 'Zon...`	Pass	0.000s
Zone 3a at 14	`_ea_zone_from_elevation({'vertical_offset': 1.4}) == 'Zon...`	Pass	0.000s
Zone 3b at 049	`_ea_zone_from_elevation({'vertical_offset': 0.49}) == 'Zo...`	Pass	0.000s
Zone 3b at zero offset	`_ea_zone_from_elevation({'vertical_offset': 0.0}) == 'Zon...`	Pass	0.000s
Insufficient data message	`any("Insufficient" in t for t in texts)`	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0` (+2 more)	Pass	0.001s
Single entry insufficient	`any("Insufficient" in t for t in texts)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Sufficient data creates content	`len(result) > 2`	Pass	0.001s
More than 20 positions -¿ only top 20 movers shown.	`isinstance(result, list)`	Pass	0.001s
Market move pnl creates table	`len(tables) >= 2`	Pass	0.000s
New trade pnl creates table	`len(tables) >= 1`	Pass	0.000s
No positions shows messages	`any("No new trades" in t for t in texts); any("No market move" in t for t in texts)`	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0` (+2 more)	Pass	0.000s
Position with notional=0 triggers fmt_gbp(0) → em-dash path.	`isinstance(result, list); len(result) > 0`	Pass	0.001s
Gauge IDs longer than 12 chars should not crash.	`isinstance(result, list)`	Pass	0.000s
Positions from 2 gauges should produce group subheaders.	`len(tables) >= 1`	Pass	0.000s
No positions shows message	`any("No open positions" in t for t in texts)`	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0` (+2 more)	Pass	0.000s
Single position	`isinstance(result, list)`	Pass	0.000s
With positions creates table	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
Line 48: daily_pnl=0 and running_pnl=0 -¿ fmt_gbp returns em-dash.	`isinstance(result, list)`	Pass	0.000s
Eod id in filename	`"EOD-2026-03-08" in path.name`	Pass	0.005s
Generate empty snapshot	`path.exists()`	Pass	0.004s
Generate report returns path	`path.exists(); path.suffix == ".pdf"`	Pass	0.009s
Generate report without pnl series	`path.exists()`	Pass	0.005s
Lines 49-50: output_dir=None -¿ uses config.get_eod_dir().	`gen.output_dir == eod_dir`	Pass	0.000s
Output dir created	`new_dir.exists()`	Pass	0.001s
Pdf not empty	`path.stat().st_size > 1000`	Pass	0.011s
EOD dates should be weekdays (Mon-Fri).	`not weekend`	Pass	0.002s
blotter/eod/ directory must exist.	`EOD_DIR.exists()`	Pass	0.000s
Each EOD file must be parseable JSON.	`not bad`	Pass	0.011s
Each EOD snapshot should have date, trades, and P&L data.	`not bad`	Pass	0.004s

Test Description	Acceptance Criteria	Result	Time
Each EOD JSON should have a matching PDF report.	—	Pass	0.000s
Should have ~63 EOD snapshots (3 months of business days).	`len(eod_files) >= 50`	Pass	0.000s
trade_marks.json must exist in blotter directory.	`TRADE_MARKS.exists()`	Pass	0.000s
Date appears in paragraph	`any("2026-03-08" in t for t in texts)`	Pass	0.000s
Empty summary	`isinstance(result, list); len(result) > 0`	Pass	0.000s
Has paragraph and table	`any(isinstance(e, Paragraph) for e in result); any(isinstance(e, Table) for e in result)`	Pass	0.000s
Negative pnl snapshot	`isinstance(result, list)`	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0` (+2 more)	Pass	0.016s
All unknown pages still creates pdf	`path.exists()`	Pass	0.001s
Audit log single event	`len(story) >= 5`	Pass	0.000s
Bcbs zero max score	`len(story) >= 4`	Pass	0.000s
Completely empty flood data	`path.exists()`	Pass	0.010s
Completely empty gauge data	`path.exists()`	Pass	0.012s
Completely empty property data	`path.exists()`	Pass	0.014s
Cover no coverage	`len(story) >= 3`	Pass	0.000s
Gauge data with hazard curve and timeseries	`path.exists()`	Pass	0.024s
gauge.json with dict-format flood_gauges is handled.	`resp.status_code == 200; data["status"] == "training"`	Pass	0.003s
Gauge with '-' for all numeric fields.	`result.exists()`	Pass	0.015s
Verify no crash or timeout with 500 property entries.	`path.exists()`	Pass	0.012s
Missing summary key	`path.exists()`	Pass	0.010s
Model with long name	`len(story) >= 3`	Pass	0.000s
Model with no name	`len(story) >= 3`	Pass	0.000s
Mortgage with None for numeric fields.	`result.exists()`	Pass	0.013s
Mrc actions with missing fields	`len(story) >= 3`	Pass	0.000s
Multiple models risk ratings	`len(story) >= 5`	Pass	0.000s
Negative value property	`path.exists()`	Pass	0.015s
Negative/nonsensical values should not crash the generator.	`path.exists()`	Pass	0.010s
None flood gauge header	`path.exists()`	Pass	0.011s

Test Description	Acceptance Criteria	Result	Time
None gauge data key	`path.exists()`	Pass	0.005s
Output is non empty bytes	`isinstance(content, bytes); len(content) > 0`	Pass	0.011s
Pdf content is bytes	`isinstance(content, bytes); len(content) > 0`	Pass	0.020s
Property data with none header	`path.exists()`	Pass	0.008s
Property with None for value field.	`result.exists()`	Pass	0.013s
Raci no activities no triggers	`len(story) >= 3`	Pass	0.000s
Remediation null due date	`len(story) >= 4`	Pass	0.000s
Report with all pages	`path.exists(); path.stat().st_size > 0`	Pass	0.028s
Report with all pages	`path.exists(); path.stat().st_size > 0`	Pass	0.022s
Report with all pages	`path.exists(); path.stat().st_size > 0`	Pass	0.018s
Single page report	`path.exists()`	Pass	0.003s
Single page report	`path.exists()`	Pass	0.003s
If joblib appears between status check and train call, returns ready.	`data["status"] == "ready"`	Pass	0.001s
Training status of one gauge does not affect another.	`data["status"] == "not_trained"`	Pass	0.000s
Validation question no short label	`len(story) >= 1`	Pass	0.000s
Zero properties at risk	`path.exists()`	Pass	0.010s
Zero value property	`path.exists()`	Pass	0.015s
Spec requirement: stronger storms -¿ broader spatial footprint.	`r_moderate < r_severe < r_extreme`	Pass	0.000s
intensity_factor ¡ 1 still gives at least base range.	`r == pytest.approx(small_model.params.base_range_km)`	Pass	0.000s
intensity_factor = 1.0 -¿ range = base_range.	`r == pytest.approx(small_model.params.base_range_km)`	Pass	0.000s
range = base * (1 + rho * (factor - 1)).	`m.effective_range_km(2.0) == pytest.approx(56.0)`	Pass	0.000s
Good elevation over 20m	`isinstance(result, list)`	Pass	0.001s
Low elevation 5 to 10m	`isinstance(result, list)`	Pass	0.001s
Moderate elevation 10 to 20m	`isinstance(result, list)`	Pass	0.001s
No ground level	`isinstance(result, list)`	Pass	0.000s
Very low elevation under 5m	`isinstance(result, list)`	Pass	0.001s
Find the elevation where flooding stops — should be consistent	`depth_below > 0; depth_above == 0.0`	Pass	0.000s
Higher properties must have lower (or equal) flood depth.	`depths[i] <= depths[i - 1] + 1e-6`	Pass	0.002s
Higher floor level should reduce flood depth.	`depth_low_floor >= depth_high_floor`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Property 8m above gauge should not flood (water_at_property ¡ threshold).	`depth == 0.0; est == 0.0`	Pass	0.000s
Property only 0.5m above gauge should flood with a strong storm.	`depth > 0`	Pass	0.000s
Concerning under 1 year	`isinstance(result, list)`	Pass	0.000s
Excellent 5 plus years	`isinstance(result, list)`	Pass	0.000s
Fair 1 to 2 years	`isinstance(result, list)`	Pass	0.000s
Good 2 to 5 years	`isinstance(result, list)`	Pass	0.000s
No curves raises value error	—	Pass	0.001s
Empty storm_id returns 400.	`resp.status_code == 400`	Pass	0.001s
Line 160-161: responses with empty storm_id are skipped.	`result["total_storms"] == 1; data["storms"][0]["storm_id"] == "STORM-valid001"`	Pass	0.003s
storm_id key missing entirely → resp.get('storm_id', '') == '' → skip.	`result["total_storms"] == 1`	Pass	0.003s
Full pipeline generates all outputs	`(pipeline_dir / "gaugets").is_dir(); (pipeline_dir / "gaugehd").is_dir()` (+3 more)	Pass	0.000s
Gauge count propagates	`n == N_GAUGES; len(list((pipeline_dir / "gaugets").glob("GAUGE-*.json"))...` (+2 more)	Pass	0.001s
Pipeline output file integrity	`expected_key in data`	Pass	0.004s
Property count propagates	`n == N_PROPERTIES; json.load(f)["summary"]["total_properties"] == n`	Pass	0.002s
Analyze epc rating all bands	`expected_word in result`	Pass	0.000s
Analyze epc rating case insensitive	`"Excellent" in page._analyze_epc_rating("a")`	Pass	0.000s
Analyze epc rating unknown	`"Unknown" in result`	Pass	0.000s
Exercises energy intensity branch when area is known.	`len(tables) >= 1`	Pass	0.000s
Full data set exercises all branches without exception.	`isinstance(result, list); len(result) > 2`	Pass	0.000s
No energy data fallback message	`any("No energy performance data" in t for t in texts)`	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0`	Pass	0.001s
With building fabric	`isinstance(result, list)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
With energy usage	`isinstance(result, list); len(result) > 0`	Pass	0.000s
With ratings data	`isinstance(result, list); len(result) > 0`	Pass	0.000s
With ratings includes table	`len(tables) >= 1`	Pass	0.000s
Close point pushed away	`dist >= PropertyPortfolioGenerator.MIN_RIVER_DISTANCE_M *...`	Pass	0.000s
Far point unchanged	`new_lat == pytest.approx(lat); new_lon == pytest.approx(lon)`	Pass	0.001s
Returns tuple of two floats	`len(result) == 2; all(isinstance(v, float) for v in result)`	Pass	0.000s
Lines 353-355: FileNotFoundError → logger.error + sys.exit(1).	`1 in exited_with`	Pass	0.001s
Lines 451-453: FileNotFoundError → logger.error + sys.exit(1).	`1 in exited`	Pass	0.002s
Lines 510-512: FileNotFoundError → logger.error + sys.exit(1).	`1 in exited`	Pass	0.001s
Test handling of invalid JSON.	`response.status_code == 400`	Pass	0.000s
Lines 451-453: bad JSON → json.JSONDecodeError → sys.exit(1).	`1 in exited`	Pass	0.003s
Lines 510-512: bad JSON → json.JSONDecodeError → sys.exit(1).	`1 in exited`	Pass	0.002s
Lines 353-355: bad JSON → json.JSONDecodeError → sys.exit(1).	`1 in exited_with`	Pass	0.002s
Test handling of missing content type.	`response.status_code in [400, 415, 500]`	Pass	0.000s
Kent multiplier	`result > 0`	Pass	0.000s
London multiplier	`london > 0`	Pass	0.000s
New property	`result > 0`	Pass	0.000s
Old property	`result > 0`	Pass	0.000s
Returns float	`isinstance(result, float)`	Pass	0.000s
Surrey multiplier	`result > 0`	Pass	0.000s
With area	`result > 0`	Pass	0.000s
With bedrooms	`result > 0`	Pass	0.000s
Zero property value falls through	`result > 0`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
If est_depth says no flood, hydrograph should also show no flood. [distance=100, prop_elev=6.0, floor=0.3, peak=8.0]	`depth == 0.0; depth > 0`	Pass	0.000s
If est_depth says no flood, hydrograph should also show no flood. [distance=1000, prop_elev=6.5, floor=0.5, peak=9.0]	`depth == 0.0; depth > 0`	Pass	0.000s
If est_depth says no flood, hydrograph should also show no flood. [distance=200, prop_elev=8.0, floor=0.3, peak=10.0]	`depth == 0.0; depth > 0`	Pass	0.000s
If est_depth says no flood, hydrograph should also show no flood. [distance=2000, prop_elev=6.0, floor=0.3, peak=7.0]	`depth == 0.0; depth > 0`	Pass	0.000s
If est_depth says no flood, hydrograph should also show no flood. [distance=3000, prop_elev=7.5, floor=0.0, peak=10.0]	`depth == 0.0; depth > 0`	Pass	0.000s
If est_depth says no flood, hydrograph should also show no flood. [distance=500, prop_elev=5.5, floor=0.8, peak=6.0]	`depth == 0.0; depth > 0`	Pass	0.000s
If est_depth says no flood, hydrograph should also show no flood. [distance=500, prop_elev=7.0, floor=0.3, peak=10.0]	`depth == 0.0; depth > 0`	Pass	0.000s
If est_depth says no flood, hydrograph should also show no flood. [distance=5000, prop_elev=6.0, floor=0.0, peak=12.0]	`depth == 0.0; depth > 0`	Pass	0.000s
100 floods gives 50bp	`spread == 50.0`	Pass	0.000s
664 floods gives 332bp	`spread == 332.0`	Pass	0.000s
Spread scales with num storms	`spread_10k == spread_20k * 2`	Pass	0.000s
Zero floods gives zero spread	`spread == 0.0`	Pass	0.000s
Zone hazard rates are still defined for reference/terrain grid.	`EA_FLOOD_ZONE_RATES['Zone 3b'] == 0.05; EA_FLOOD_ZONE_RATES['Zone 3a'] == 0.02` (+2 more)	Pass	0.000s
Above max is zero	`d.exceedance_probability(d.params.max_intensity + 10) == 0.0`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Above threshold uses pareto	`0 < p < 1`	Pass	0.000s
At max intensity is zero	`d.exceedance_probability(d.params.max_intensity) == 0.0`	Pass	0.000s
At min intensity is one	`d.exceedance_probability(d.params.min_intensity) == 1.0`	Pass	0.000s
At threshold continuous	`abs(p_just_below - p_just_above) < 0.05`	Pass	0.000s
Below min is one	`d.exceedance_probability(d.params.min_intensity - 5) == 1.0`	Pass	0.000s
Below threshold uses normal	`0 < p < 1`	Pass	0.000s
Severe floods should be rare.	`prob <= 0.05 + 1e-9`	Pass	0.000s
Normal levels should be common.	`prob > 0.5`	Pass	0.000s
Monotonically decreasing	`all(probs[i] > probs[i + 1] for i in range(len(probs) - 1))`	Pass	0.000s
Return period should be inverse of probability.	`return_period > 1`	Pass	0.000s
Get gauge id missing key	`page._get_gauge_id({}) == "Unknown Gauge"; page._get_gauge_id(None) == "Unknown Gauge"`	Pass	0.000s
Malformed gauge data returns error element	`isinstance(elements, list); len(elements) > 0`	Pass	0.000s
Compliant verdict	`verdict_badge._cellvalues[1][0] == 'COMPLIANT'`	Pass	0.000s
Non compliant verdict	`verdict_badge._cellvalues[1][0] in ( 'PARTIAL...`	Pass	0.000s
Run history shown	`'Run History' in texts`	Pass	0.000s
Extracts ids	`extract_property_ids(data) == {"P1", "P2"}; extract_mortgage_ids(data) == {"M1", "M2"}` (+1 more)	Pass	0.000s
Missing gauge id skipped	`extract_gauge_ids(data) == set()`	Pass	0.000s
None returns empty set	`extract_property_ids(None) == set(); extract_mortgage_ids(None) == set()` (+2 more)	Pass	0.000s
Extracts ids	`extract_property_ids(data) == {"P1", "P2"}; extract_mortgage_ids(data) == {"M1", "M2"}` (+1 more)	Pass	0.000s
None returns empty set	`extract_property_ids(None) == set(); extract_mortgage_ids(None) == set()` (+2 more)	Pass	0.000s
Extracts property ids from mortgages	`extract_mortgage_property_ids(data) == {"P1", "P2"}`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
None returns empty set	`extract_property_ids(None) == set()`; `extract_mortgage_ids(None) == set()` (+2 more)	Pass	0.000s
Empty items returns empty set	`extract_property_ids({"items": []}) == set()`	Pass	0.000s
Extracts ids	`extract_property_ids(data) == {"P1", "P2"}`; `extract_mortgage_ids(data) == {"M1", "M2"}` (+1 more)	Pass	0.000s
Missing property id skipped	`extract_property_ids(data) == set()`	Pass	0.000s
None returns empty set	`extract_property_ids(None) == set()`; `extract_mortgage_ids(None) == set()` (+2 more)	Pass	0.000s
Construction year extracted	`gen._extract_property_info(self._full_record)["construc...`	Pass	0.000s
Flood risk extracted	`gen._extract_property_info(self._full_record)["flood_ri...`	Pass	0.000s
Lat lon extracted	`abs(info["latitude"] - 51.45) < 0.001`; `abs(info["longitude"] + 0.30) < 0.001`	Pass	0.000s
Missing attrs default construction year	`gen._extract_property_info({"PropertyHeader": {"Header": ...`	Pass	0.000s
Missing header defaults property id	`gen._extract_property_info({"PropertyHeader": {}})["prope...`	Pass	0.000s
Missing location defaults postcode and coords	`info["postcode"] == "SW1A 1AA"`; `abs(info["latitude"] - 51.5) < 0.01` (+1 more)	Pass	0.000s
Missing property header returns defaults	`info["property_id"] == ""`; `info["property_value"] == 500000` (+1 more)	Pass	0.000s
Missing risk assessment defaults flood risk	`gen._extract_property_info({"PropertyHeader": {"Header": ...`	Pass	0.000s
Missing valuation defaults property value	`gen._extract_property_info(record)["property_value"] == 5...`	Pass	0.000s
Postcode extracted	`gen._extract_property_info(self._full_record)["postcode...`	Pass	0.000s
Property id extracted from header	`gen._extract_property_info(self._full_record)["property...`	Pass	0.000s
Property type extracted	`gen._extract_property_info(self._full_record)["property...`	Pass	0.000s
Property value extracted	`gen._extract_property_info(self._full_record)["property...`	Pass	0.000s
Age band factors within reasonable bounds	`lo > 0; hi >= lo`	Pass	0.000s
Base area ranges all positive	`lo > 0; hi > lo`	Pass	0.000s
Base price per sqm all positive	`lo > 0; hi > lo`	Pass	0.000s
Condition factors ordered by severity	`excellent_lo > poor_hi`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Epc factors a highest	a_lo > g_hi	Pass	0.000s
Flood risk factors decrease with risk	very_low_lo > very_high_hi	Pass	0.000s
Min max value bounds	MIN_PROPERTY_VALUE == 150_000; MAX_PROPERTY_VALUE == 5_000_000	Pass	0.000s
Proximity zones have range key	'range' in data	Pass	0.000s
Rent per sqm all present	set(RENT_PER_SQM.keys()) == expected_types	Pass	0.000s
Rental yield rates all present	set(RENTAL_YIELD_RATES.keys()) == expected_types	Pass	0.000s
A failure matching FAILURE_METADATA renders a structured table.	len(tables) >= 3	Pass	0.001s
A failure NOT in FAILURE_METADATA falls back to bullet point.	any('test_trade_gauges_exist_in_gauge_json' in t ...	Pass	0.000s
Empty counterparty file uses fallback	len(trades) > 0; all(cid.startswith("CTPY-") for cid in ctpy_ids)	Pass	0.003s
Missing counterparty file uses fallback	len(trades) > 0	Pass	0.002s
Exact lookup found	r.status_code == 200; data["status"] == "success" (+2 more)	Pass	0.002s
Exact lookup not found	r.status_code == 404; "Field not found" in r.get_json()["message"]	Pass	0.001s
Exact lookup wrong report	r.status_code == 404	Pass	0.001s
Exact lookup wrong section	r.status_code == 404	Pass	0.002s
No params 400	r.status_code == 400; "Provide" in r.get_json()["message"]	Pass	0.002s
No registry 404	r.status_code == 404; "not found" in data["message"].lower() (+1 more)	Pass	0.000s
Search by cdm path	data["count"] >= 1	Pass	0.002s
Search by source field	data["count"] == 1; data["results"][0]["field"] == "spread" (+1 more)	Pass	0.002s
Search case insensitive	data["count"] >= 1	Pass	0.002s
Search finds matches	r.status_code == 200; data["status"] == "success" (+2 more)	Pass	0.002s
Search no results	r.status_code == 200; data["count"] == 0 (+1 more)	Pass	0.002s
Search result structure	data["count"] >= 1; "report" in result (+4 more)	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
If only report and search are given (not all 3 exact keys), search is used.	`data["count"] >= 1`	Pass	0.002s
None field def returns empty string	`result == ""`	Pass	0.000s
Numeric field def returns empty string	`result == ""`	Pass	0.000s
String field def returned as is	`result == "literal_value"`	Pass	0.000s
No type defaults to text	`"SomeField" in result`	Pass	0.000s
Unknown type returns empty	`result == ""`	Pass	0.000s
5 year filter	`len(result) < len(flows); len(result) > 0`	Pass	0.002s
Empty flows returns empty	`GaugeHistoricalDaily().filter_by_years([], years=10) == []`	Pass	0.000s
Empty list returns empty	`result == []`	Pass	0.000s
Filter 1 year keeps only recent	`"2010-01-01" not in [f["date"] for f in result]`	Pass	0.000s
Filter keeps recent	`"2000-06-01" not in dates; "2020-06-01" in dates`	Pass	0.000s
Large years returns all	`len(result) == len(flows)`	Pass	0.001s
None years returns all	`len(gen.filter_by_years(flows, years=None)) == 2`	Pass	0.000s
None years returns all	`len(result) == len(flows)`	Pass	0.000s
All dates within range kept	`len(gen.filter_by_years(flows, years=5)) == 3`	Pass	0.000s
Zero years returns only most recent year	`"2020-01-01" in [f["date"] for f in result]`	Pass	0.000s
Empty property does not crash	`isinstance(self._page().generate_elements({}), list)`	Pass	0.000s
Has financial header	`any("Financial" in t for t in texts)`	Pass	0.000s
Has table	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
No area skips value per sqm	`isinstance(result, list)`	Pass	0.000s
No property value	`isinstance(result, list)`	Pass	0.000s
No valuation data	`isinstance(result, list)`	Pass	0.000s
Other valuation field with value in name	`isinstance(result, list)`	Pass	0.000s
Other valuation field without value in name	`isinstance(result, list)`	Pass	0.000s
Returns list	`isinstance(self._page().generate_elements(_make_property()), list)`	Pass	0.000s
Returns non empty list	`len(self._page().generate_elements(_make_property())) > 0`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Value per sqm shown when area present	`isinstance(result, list)`	Pass	0.000s
Atl basin	`"EVT-001" in ids; "EVT-002" in ids` (+1 more)	Pass	0.001s
Case insensitive	`result is not None; len(result) >= 1`	Pass	0.001s
Unknown basin returns empty	`storm_loader.find_by_basin("XYZ") == []`	Pass	0.003s
Wpac basin	`len(result) == 1; storm_loader.get_entity_id(result[0]) == "EVT-003"`	Pass	0.002s
All category 4	`all( s["TCEvent"]["Header"]["Category"] == 4 ...`	Pass	0.001s
Category 3	`"EVT-003" in ids`	Pass	0.001s
Category 5	`len(result) == 1; storm_loader.get_entity_id(result[0]) == "EVT-002"`	Pass	0.001s
No matches returns empty	`storm_loader.find_by_category(99) == []`	Pass	0.001s
Find by county case insensitive	`len(results) == 1`	Pass	0.001s
Find by county empty portfolio	`results == []`	Pass	0.004s
Find by county exact match	`len(results) == 1; results[0]["PropertyHeader"]["Header"]["PropertyID"] == "...`	Pass	0.005s
Property with no County field should not crash.	`results == []`	Pass	0.003s
Find by county no match	`results == []`	Pass	0.001s
Find by county partial match	`len(results) == 2; ids == {"PROP-001", "PROP-003"}`	Pass	0.001s
Case insensitive	`result is not None; len(result) >= 1`	Pass	0.001s
Exact match	`result is not None; storm_loader.get_entity_id(result) == "EVT-001"`	Pass	0.001s
No match returns none	`storm_loader.find_by_name("Nonsense Storm") is None`	Pass	0.001s
Partial match	`result is not None; storm_loader.get_entity_id(result) == "EVT-003"`	Pass	0.001s
Missing season returns empty	`storm_loader.find_by_season(1900) == []`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Season 2017	`len(result) == 1; storm_loader.get_entity_id(result[0]) == "EVT-001"`	Pass	0.007s
Season with multiple	`len(result) == 1`	Pass	0.001s
Dict with 'flood_gauges' list → finds first gauge correctly.	`result["FloodGauge"]["Header"]["GaugeName"] == "Gauge One"`	Pass	0.000s
Dict with 'flood_gauges' list → finds correct gauge by ID.	`result["FloodGauge"]["Header"]["GaugeID"] == "GAUGE-002"`	Pass	0.000s
Plain list of gauge dicts → finds the correct one.	`result["FloodGauge"]["Header"]["GaugeID"] == "GAUGE-B"`	Pass	0.000s
Dict with 'FloodGauge' key (single object) → returns the dict itself.	`result is sample_gauge_data`	Pass	0.000s
Flood gauges key found	`result == data["flood_gauges"][0]`	Pass	0.000s
Dict with neither 'flood_gauges' nor 'FloodGauge' key → ValueError raised.	—	Pass	0.000s
Invalid dict structure raises	—	Pass	0.000s
Invalid type raises	—	Pass	0.000s
Passing a string instead of dict/list → ValueError raised.	—	Pass	0.000s
List input found	`result == gauges[0]`	Pass	0.000s
Malformed gauge entry skipped	`result == data["flood_gauges"][1]`	Pass	0.000s
Gauge ID not in plain list → ValueError raised.	—	Pass	0.000s
Gauge ID not present in data → ValueError raised.	—	Pass	0.000s
Not found raises value error	—	Pass	0.000s
Single gauge dict returns self	`result is data`	Pass	0.000s
Line 361: dict with neither 'flood_gauges' nor 'FloodGauge' → ValueError.	—	Pass	0.000s
Flood gauges key first gauge	`result["FloodGauge"]["Header"]["GaugeID"] == "G-A"`	Pass	0.000s
Flood gauges key last gauge	`result["FloodGauge"]["Header"]["GaugeID"] == "G-C"`	Pass	0.000s
Line 375: gauge_id not present in any entry → ValueError.	—	Pass	0.000s
Integer input raises	—	Pass	0.000s
Line 363: list input → list used directly as gauge collection.	`result["FloodGauge"]["Header"]["GaugeID"] == "G-LIST"`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Malformed entry missing flood gauge key skipped	`result["FloodGauge"]["Header"]["GaugeID"] == "G-REAL"`	Pass	0.000s
Malformed entry with none skipped	`result["FloodGauge"]["Header"]["GaugeID"] == "G-OK"`	Pass	0.000s
Single flood gauge dict returns itself	`result is data`	Pass	0.000s
String input raises	—	Pass	0.000s
Finds existing meeting	`meeting is not None; meeting["id"] == "M1"` (+1 more)	Pass	0.032s
Returns error for missing meeting	`meeting is None; err is not None`	Pass	0.070s
Dict with mortgages found	`result["amount"] == 100`	Pass	0.000s
Dict with mortgages found	`result["amount"] == 100`	Pass	0.000s
Dict with mortgages not found returns none	`result is None`	Pass	0.000s
Dict with mortgages not found returns none	`result is None`	Pass	0.000s
Dict without mortgages key returns data itself	`result is not None`	Pass	0.000s
Empty mortgages list returns none	`result is None`	Pass	0.000s
Lines 406-407: non-dict, non-list -¿ return data as-is.	`result == "some_string"`	Pass	0.000s
Invalid type returns data as is	`result == "some_string"`	Pass	0.000s
List input found	`result == data[0]; result["amount"] == 200`	Pass	0.000s
List input found	`result["amount"] == 200`	Pass	0.000s
List input not found returns none	`result is None`	Pass	0.000s
Distance is positive	`dist > 0`	Pass	0.000s
Returns nearest of three	`nearest.intensity == 80.0; dist == pytest.approx(0.0, abs=1.0)`	Pass	0.000s
Single track point	`nearest is track[0]`	Pass	0.000s
Dict without list key wraps itself	`result is data`	Pass	0.000s
Dict without properties wraps in list	`result == data`	Pass	0.000s
Empty properties list raises	—	Pass	0.000s
Invalid type raises	—	Pass	0.000s
Invalid type raises value error	—	Pass	0.000s
List input	`result == data[0]`	Pass	0.000s
List input found	`result == data[0]; result["amount"] == 200`	Pass	0.000s
List not found raises	—	Pass	0.000s
None raises value error	—	Pass	0.000s
Not found raises	—	Pass	0.000s
Not found raises value error	—	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Portfolio key	`result == data["portfolio"][0]`	Pass	0.000s
Portfolio key found	`result["PropertyHeader"]["Header"]["PropertyID"] == "PA"`	Pass	0.000s
Properties key	`result == data["properties"][0]`	Pass	0.000s
Properties key found	`result["PropertyHeader"]["Header"]["PropertyID"] == "P-002"`	Pass	0.000s
Train gauge creates joblib file	`(model_dir / "GAUGE-001.joblib").exists()`	Pass	0.086s
Train gauge has metrics	`"metrics" in result; "auc_roc" in result["metrics"]` (+1 more)	Pass	0.088s
Train gauge returns dict	`isinstance(result, dict)`	Pass	0.202s
Train gauge single class skipped	`result["status"] == "skipped"; result["reason"] == "single_class"`	Pass	0.002s
Train gauge status trained	`result["status"] == "trained"`	Pass	0.082s
Flood above trigger	`depth > 0.0`	Pass	0.000s
Flood attenuates with distance	`depth_close > depth_far`	Pass	0.000s
High elevation no flood	`depth == 0.0`	Pass	0.000s
No flood below trigger	`depth == 0.0`	Pass	0.000s
Calculate time above warning level.	`duration > 0`	Pass	0.000s
Duration should be in reasonable range.	`5 < duration < 30`	Pass	0.000s
No time above threshold for normal readings.	`duration == 0`	Pass	0.000s
Create mapping returns flat dict	`isinstance(mapping, dict); mapping.get("gauge_id") == "GAUGE-test1234"` (+7 more)	Pass	0.000s
Create mapping with empty input	`isinstance(gauge_cdm.create_mapping({}), dict)`	Pass	0.000s
Get field info for gauge id	`info is not None; info.get("type") == "text"`	Pass	0.000s
Required fields include gauge id	`any("GaugeID" in f for f in gauge_cdm.get_required_fields())`	Pass	0.000s
Schema has flood gauge root	`"FloodGauge" in gauge_cdm.schema`	Pass	0.000s
Schema has header section	`"Header" in gauge_cdm.schema["FloodGauge"]`	Pass	0.000s
Validate empty data returns errors	`isinstance(gauge_cdm.validate({}), dict)`	Pass	0.000s
Validate valid gauge data	`len(gauge_cdm.validate(data)) == 0`	Pass	0.000s
Available gauges	`"GAUGE-001" in gauges`	Pass	0.000s
Available gauges empty dir	`predictor.available_gauges() == []`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Lines 188-192: _load_summary() loads training_summary.json.	'num_gauges' in summary; 'num_trained' in summary	Pass	0.001s
_load_summary caches on second call.	s1 is s2	Pass	0.001s
Predict caches model	"GAUGE-001" in predictor._models	Pass	0.009s
Predict high level higher probability	p_high > p_low	Pass	0.009s
Predict in range	0.0 <= prob <= 1.0	Pass	0.008s
Predict missing model raises	—	Pass	0.000s
Predict returns float	isinstance(prob, float)	Pass	0.010s
Predict series all in range	all(0.0 <= p <= 1.0 for p in result)	Pass	0.008s
Predict series default derivatives	len(result) == 2	Pass	0.009s
Predict series returns list	isinstance(result, list); len(result) == 4	Pass	0.009s
Predict series with derivatives	len(result) == 3	Pass	0.009s
predict_surface returns a 2D probability grid (hours × levels).	'water_levels' in result; 'hours' in result (+3 more)	Pass	0.009s
Directly call build_property_hydrograph and verify it produces depth.	max_depth > 0	Skip	0.003s
Properties with est_depth ¿ 0 should have some flooded events.	—	Pass	0.002s
The IDW interpolation should produce high WSE from alert storms.	water_at_prop > 0	Pass	0.003s
At least one property should have a nearest gauge with alert storms.	pid is not None	Pass	0.005s
At high threshold	RiskAssessor.assess_flood_risk_level(1.0) == "High"	Pass	0.000s
At low threshold	RiskAssessor.assess_flood_risk_level(0.1) == "Low"	Pass	0.000s
At medium threshold	RiskAssessor.assess_flood_risk_level(0.5) == "Medium"	Pass	0.000s
Below low threshold	RiskAssessor.assess_flood_risk_level(0.05) == "Low"	Pass	0.000s
High range	RiskAssessor.assess_flood_risk_level(0.75) == "High"	Pass	0.000s
Medium range	RiskAssessor.assess_flood_risk_level(0.3) == "Medium"	Pass	0.000s
Monotone with depth	levels == order	Pass	0.000s
Negative returns unknown	RiskAssessor.assess_flood_risk_level(-1.0) == "Unknown"	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
None returns unknown	`RiskAssessor.assess_flood_risk_level(None) == "Unknown"; RiskAssessor.assess_ltv_risk_level(None) == "Unknown"`	Pass	0.000s
Very high	`RiskAssessor.assess_flood_risk_level(2.5) == "Very High"`	Pass	0.000s
Zero is very low	`RiskAssessor.assess_flood_risk_level(0.0) == "Very Low"`	Pass	0.000s
Lines 59-74: model builds without error given valid inputs.	`model.properties is not None; model.flood_event == {"GAUGE-000": 1.5}`	Pass	3.572s
Line 72-74: correlation_matrix is set after __init__.	`model.correlation_matrix is not None`	Pass	0.001s
Lines 62-63: custom correlation_distance and base_correlation stored.	`model.correlation_distance == 500; model.base_correlation == 0.6`	Pass	0.001s
Lines 86-88: missing elevation → warning + default 20.0.	`any("elevation" in str(warning.message).lower() for warni...; (model.properties["elevation"] == 20.0).all()`	Pass	0.001s
Lines 94-95: missing floor_level_metres → default 0.0.	`"floor_level_metres" in model.properties.columns; (model.properties["floor_level_metres"] == 0.0).all()`	Pass	0.001s
Line 81: missing 'geometry' column raises ValueError.	—	Pass	0.000s
Lines 83-84: property_id auto-assigned when absent.	`"property_id" in model.properties.columns; model.properties["property_id"].iloc[0].startswith("PROP_")`	Pass	0.001s
Lines 90-92: missing property_type → warning + default 'residential'.	`any("property_type" in str(warning.message).lower() for w...; (model.properties["property_type"] == "residential").all()`	Pass	0.001s
Line 81: missing 'value' column raises ValueError.	—	Pass	0.000s
Line 59: properties is a copy, original not mutated.	`set(props.columns) == original_cols`	Pass	0.001s
Line 71: kdtree built when gauge_data provided.	`model.gauge_data is not None`	Pass	0.005s

Test Description	Acceptance Criteria	Result	Time
Lines 131-139: analyze_risk_clusters() returns DataFrame, sets risk_clusters.	`isinstance(result, pd.DataFrame)`; `model.risk_clusters is not None`	Pass	0.231s
Lines 114-120: analyze_spatial_concentration() returns DataFrame.	`isinstance(result, pd.DataFrame)`	Pass	0.006s
Lines 122-129: calculate_advanced_metrics() returns dict.	`isinstance(result, dict)`	Pass	0.002s
Line 103: calculate_direct_impacts() returns ndarray.	`isinstance(impacts, np.ndarray)`; `len(impacts) == len(model.properties)`	Pass	0.000s
Line 99: calculate_flood_depths() returns ndarray.	`isinstance(depths, np.ndarray)`; `len(depths) == len(model.properties)`	Pass	0.000s
Line 151: _haversine_distance() delegates to haversine_distance.	`isinstance(d, float)`; `d > 0`	Pass	0.000s
Lines 107-112: simulate_portfolio_impact() returns dict.	`isinstance(result, dict)`	Pass	0.026s
Default params produces data	`len(result) > 0`; `"readings" in result[0]`	Pass	0.001s
String start time is parsed	`len(result) == 50`	Pass	0.000s
Lines 235-239: invalid metadata JSON triggers except continue.	`r.status_code == 200`	Pass	0.030s
Lines 280-281: property.json read error triggers except pass.	`r.status_code == 200`; `data["status"] == "success"` (+1 more)	Pass	0.030s
Line 288: property not in valued_ids is skipped; storm filtered out.	`"STORM-E" not in storm_ids`	Pass	0.029s
Line 292: flood events with depth ¡= 0 are skipped; storm filtered out.	`"STORM-F" not in storm_ids`	Pass	0.033s
Lines 218-219: invalid JSON in gaugets file triggers except continue.	`data["status"] == "success"`; `data["count"] >= 1`	Pass	0.030s
Lines 280-292: property flooding enriches storm with depth/damage.	`storms["STORM-D"]["properties_flooded"] == 1`; `storms["STORM-D"]["max_depth_m"] == 1.2` (+1 more)	Pass	0.029s
Line 204: storm responses with empty storm_id are skipped.	`data["status"] == "success"`; `"" not in storm_ids` (+1 more)	Pass	0.032s
Lines 235-237: storms.json enrichment path (not sequences).	`s["intensity_category"] == "extreme"`; `s["effective_precipitation_mm"] == 80.0`	Pass	0.031s

Test Description	Acceptance Criteria	Result	Time
Lines 254-266: storms without metadata get classified by gauges_severe count.	storms["S-CAT"]["intensity_category"] == "catastrophic"; storms["S-CAT"]["name"] == "Catastrophic" (+4 more)	Pass	0.052s
Property without flood_zone in source data should default to Zone 1.	data['property_hazard_curves']['PROP-NOZONE']['flood_zone']	Pass	0.376s
Flood zone is valid zone name	pc['flood_zone'] in valid_zones	Pass	0.274s
Flood zone present in result	'flood_zone' in pc	Pass	0.297s
Flood zone values correct	curves['PROP-001']['flood_zone'] == 'Zone 2'; curves['PROP-002']['flood_zone'] == 'Zone 1' (+1 more)	Pass	0.390s
Unknown zone	"Unknown" in r or "Zone X" in r	Pass	0.000s
Zone 1 low risk	"Low" in r	Pass	0.000s
Zone 2 medium risk	"Medium" in r	Pass	0.000s
Zone 3a high risk	"High" in r	Pass	0.000s
Zone 3b floodplain	"floodplain" in r.lower() or "Functional" in r	Pass	0.000s
Zone in page output	isinstance(result, list)	Pass	0.002s
Large value	'1,000,000' in result; result.startswith('£')	Pass	0.000s
Negative	'5,000' in result	Pass	0.000s
Non numeric fallback	fmt_gbp('N/A') == 'N/A'	Pass	0.000s
None fallback	result == 'None'	Pass	0.000s
Positive float	fmt_gbp(1234.56) == '£1,235'	Pass	0.005s
Zero	fmt_gbp(0) == '£0'	Pass	0.000s
line 286: f'{value:.6f}'.	result == '51.500000'	Pass	0.000s
Negative coordinate	result == '-0.100000'	Pass	0.000s
None delegates to format value	page._format_coordinate(None) == 'Not specified'	Pass	0.000s
None currency returns not specified	result == 'Not specified'	Pass	0.000s
String currency delegates	result == "N/A"	Pass	0.000s
Abbreviation replacements	'GPS' in result	Pass	0.000s
Camel case split	'Gauge' in result	Pass	0.000s
Non numeric delegates	page._format_frequency(None) == 'Not specified'	Pass	0.000s
Positive returns times	page._format_frequency(3) == '3 times'	Pass	0.000s
line 294: 'Never'.	page._format_frequency(0) == 'Never'	Pass	0.000s
Integer value	result == '5.000 m'	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Non numeric delegates to format value	`result == 'Not specified'`	Pass	0.000s
Numeric value with custom unit	`result == '1.234 km'`	Pass	0.000s
line 280: f'{value:.3f} {unit}'.	`result == '3.500 m'`	Pass	0.000s
None percentage returns not specified	`result == 'Not specified'`	Pass	0.000s
String percentage delegates	`result == "N/A"`	Pass	0.000s
Blank/whitespace-only string (line 272) -¿ 'Not specified'.	`page._format_value('') == 'Not specified'; page._format_value(' ') == 'Not specified'`	Pass	0.000s
False returns no	`page._format_value(False) == 'No'`	Pass	0.000s
float.is_integer() -¿ str(int(value)).	`result == '3'`	Pass	0.000s
float with fractional part -¿ 2 decimal places.	`result == '3.14'`	Pass	0.000s
int branch (line 270): str(value).	`result == '42'`	Pass	0.000s
None returns not specified	`page._format_value(None) == 'Not specified'`	Pass	0.000s
Normal string returned	`page._format_value('hello') == 'hello'`	Pass	0.000s
True returns yes	`page._format_value(True) == 'Yes'`	Pass	0.000s
Empty string returns not specified	`page._format_value('') == 'Not specified'`	Pass	0.000s
Whitespace string returns not specified	`page._format_value(' ') == 'Not specified'`	Pass	0.000s
None returns not specified	`page._format_value(None) == 'Not specified'`	Pass	0.000s
Pad foundation	`isinstance(result, list)`	Pass	0.000s
Pile foundation	`isinstance(result, list)`	Pass	0.000s
Raft foundation	`isinstance(result, list)`	Pass	0.000s
Strip foundation	`isinstance(result, list)`	Pass	0.000s
Unknown foundation	`isinstance(result, list)`	Pass	0.000s
Four exceedances high	`"High frequency" in str(elements)`	Pass	0.000s
One exceedance	`"One severe level exceedance" in str(elements)`	Pass	0.000s
Two exceedances moderate	`"Moderate frequency" in str(elements)`	Pass	0.000s
Zero exceedances	`"No severe level exceedances recorded" in str(elements)`	Pass	0.000s
Distance matrix built	`thames_model.dist_matrix.shape == (52, 52)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
From dict	`p.base_range_km == 50.0; p.nugget == 0.10` (+1 more)	Pass	0.030s
Lines 323-325: construct model from a gauge.json file.	`model.n_gauges == 3; model.dist_matrix.shape == (3, 3)`	Pass	0.002s
Loads 52 gauges	`thames_model.n_gauges == 52`	Pass	0.000s
Full js contains iife	`'(function()' in full_governance_js or '(function ()' in ...`	Pass	0.000s
The Leaflet control that adds the Regulatory shield must be present.	`'GovernanceControl' in full_governance_js; 'Regulatory Compliance' in full_governance_js`	Pass	0.000s
Full js contains render audit reports	`'renderAuditReports' in full_governance_js`	Pass	0.000s
Full js contains script tags	`'<script>' in full_governance_js; '</script>' in full_governance_js`	Pass	0.000s
Full js contains show mg panel	`'showMgPanel' in full_governance_js`	Pass	0.000s
Terminal popup added in last session must be present.	`'_openTerminalPopup' in full_governance_js; '_auditTerminalRefreshNow' in full_governance_js`	Pass	0.000s
Full js is str	`isinstance(full_governance_js, str)`	Pass	0.000s
Full js non empty	`len(full_governance_js) > 10_000`	Pass	0.000s
Run node –check on the compiled governance script.	`result.returncode == 0`	Pass	0.194s
alpha=0.3 -¿ fast rise, long tail.	`rise_val > fall_val`	Pass	0.000s
Invalid alpha	`gamma_shape(0.5, 0.0) == 0.0; gamma_shape(0.5, -1.0) == 0.0`	Pass	0.000s
The maximum value over u in (0,1] should be 1.0 at u=alpha. [alpha=0.3]	`max(vals) == pytest.approx(1.0, abs=1e-4)`	Pass	0.000s
The maximum value over u in (0,1] should be 1.0 at u=alpha. [alpha=0.7]	`max(vals) == pytest.approx(1.0, abs=1e-4)`	Pass	0.000s
The maximum value over u in (0,1] should be 1.0 at u=alpha. [alpha=1.0]	`max(vals) == pytest.approx(1.0, abs=1e-4)`	Pass	0.000s
Negative u	`gamma_shape(-0.1, 0.5) == 0.0`	Pass	0.000s
phi(alpha^2) should equal 1.0 (the normalised peak). [alpha=0.3]	`abs(val - 1.0) < 1e-10`	Pass	0.000s
phi(alpha^2) should equal 1.0 (the normalised peak). [alpha=0.5]	`abs(val - 1.0) < 1e-10`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
phi(alpha^2) should equal 1.0 (the normalised peak). [alpha=0.7]	`abs(val - 1.0) < 1e-10`	Pass	0.000s
phi(alpha^2) should equal 1.0 (the normalised peak). [alpha=1.0]	`abs(val - 1.0) < 1e-10`	Pass	0.000s
Zero at zero	`gamma_shape(0.0, 0.3) == 0.0`	Pass	0.000s
Zero beyond one	`gamma_shape(1.01, 0.5) == 0.0`	Pass	0.000s
Matches scalar	`arr[h] == pytest.approx(expected, abs=1e-10)`	Pass	0.000s
Zero outside window	`arr[0] == 0.0; arr[49] == 0.0` (+1 more)	Pass	0.000s
Base level calculated	`abs(gc.base_level - 1.4) < 0.01`	Pass	0.000s
Creates gauge config	`isinstance(gc, GaugeConfig)`	Pass	0.000s
Defaults when no flood stages	`gc.flood_alert == 4.0`	Pass	0.000s
Handles case where FloodGauge key is missing (flat data).	`gc.flood_alert == 3.5`	Pass	0.000s
Flood alert set	`gc.flood_alert == 4.0`	Pass	0.000s
Gauge id set	`gc.gauge_id == "GAUGE-099"`	Pass	0.000s
Historical high set	`gc.historical_high == 6.0`	Pass	0.000s
Lines 32-34: loader.list_all() raises -¿ 500 response.	`response.status_code == 500; data['status'] == 'error'`	Pass	0.000s
Line 19: OPTIONS on /api/v1/gauges returns 200.	`response.status_code == 200; data['status'] == 'ok'`	Pass	0.000s
Empty historical daily shows fallback	`any(t for t in texts)`	Pass	0.000s
No timeseries data shows fallback	`any("No historical" in t or "No" in t for t in texts)`	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0`	Pass	0.000s
With observations includes content	`len(result) >= 2`	Skip	0.094s
With observations returns elements	`len(result) > 2`	Skip	0.102s
Certified conclusion	`"certification standards" in str(result)`	Pass	0.001s
Empty data does not crash	`isinstance(result, list)`	Pass	0.000s
Freq high significance	`"Regular severe events" in str(result)`	Pass	0.001s
Freq low significance	`"Low frequency events" in str(result)`	Pass	0.001s
Freq zero no exceedances conclusion	`"No historical severe flood events" in str(result)`	Pass	0.001s
Freq zero significance	`"No severe exceedances" in str(result)`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Gauge overview summary section	`any("Gauge Overview Summary" in t for t in texts)`	Pass	0.001s
Get gauge id missing	`page._get_gauge_id({}) == "Unknown Gauge"`	Pass	0.000s
Get gauge id present	`page._get_gauge_id(_make_gauge()) == "GAUGE-001"`	Pass	0.000s
Get gauge name missing	`page._get_gauge_name({}) == "Unknown Gauge Name"`	Pass	0.000s
Get gauge name present	`page._get_gauge_name(_make_gauge()) == "Chelsea Bridge"`	Pass	0.000s
Has tables	`any(isinstance(e, Table) for e in result)`	Pass	0.001s
High risk conclusion	`"high flood risk" in str(result)`	Pass	0.001s
hist_high ¿ severe_level → 'Exceeds severe warning' significance.	`"Exceeds severe warning" in str(result)`	Pass	0.001s
hist_high ¡= severe_level → 'Within expected range' significance.	`"Within expected range" in str(result)`	Pass	0.001s
Key metrics summary section	`any("Key Metrics Summary" in t for t in texts)`	Pass	0.001s
Low risk conclusion	`"Low flood risk" in str(result)`	Pass	0.001s
Moderate risk conclusion	`"Moderate flood risk" in str(result)`	Pass	0.001s
Operational status conclusion	`"fully operational" in str(result)`	Pass	0.001s
Operational status summary section	`any("Operational Status Summary" in t for t in texts)`	Pass	0.001s
Report conclusions section	`any("Report Conclusions" in t for t in texts)`	Pass	0.001s
Returns list	`isinstance(result, list); len(result) > 0`	Pass	0.001s
Timeseries alert conclusion	`"ALERT" in str(result)`	Pass	0.001s
Timeseries normal conclusion	`"normal water levels" in str(result)`	Pass	0.001s
Timeseries severe conclusion	`"CRITICAL" in str(result)`	Pass	0.001s
Timeseries warning conclusion	`"WARNING" in str(result)`	Pass	0.001s
Title present	`any("Comprehensive Data Summary" in t for t in texts)`	Pass	0.001s
Historical graph no dates fallback	`isinstance(result, list)`	Pass	0.000s
Historical graph with thresholds	`isinstance(result, list); len(result) > 0`	Pass	0.037s
No storm responses shows message	`any("No storm" in t for t in texts)`	Pass	0.000s
No timeseries data shows no data message	`any("No historical" in t or "No storm" in t for t in texts)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Realised floods no exceedances	`any("No flood events" in t for t in texts)`	Pass	0.000s
Realised floods with exceedances	`len(tables) >= 1`	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0`	Pass	0.000s
With historical daily returns more elements	`len(result) > 2`	Pass	0.094s
With storm responses includes table	`len(tables) >= 1`	Pass	0.000s
Worst storms severity classification	`isinstance(result, list)`	Pass	0.000s
Worst storms table top 10	`len(tables) == 1`	Pass	0.000s
Current level at alert	`"FLOOD ALERT" in str(result)`	Pass	0.001s
Current level at severe	`"SEVERE FLOOD WARNING" in str(result)`	Pass	0.001s
Current level at warning	`"FLOOD WARNING" in str(result)`	Pass	0.001s
Current level below alert is normal	`"Normal" in str(result)`	Pass	0.001s
Empty data does not crash	`isinstance(result, list)`	Pass	0.000s
Get gauge id missing	`page._get_gauge_id({}) == "Unknown Gauge"`	Pass	0.000s
Get gauge id present	`page._get_gauge_id(_make_gauge()) == "GAUGE-001"`	Pass	0.000s
Has tables	`any(isinstance(e, Table) for e in result)`	Pass	0.001s
Historical exceedance section	`any("Historical Threshold Exceedance" in t for t in texts)`	Pass	0.000s
No timeseries shows message	`any("No current timeseries data" in t for t in texts)`	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0`	Pass	0.000s
Threshold analysis section	`any("Threshold Analysis" in t for t in texts)`	Pass	0.000s
Alert-to-warning and warning-to-severe ranges are added to analysis table.	`isinstance(result, list)`	Pass	0.000s
Title present	`any("Flood Stages" in t for t in texts)`	Pass	0.000s
Uk thresholds section	`any("UK Flood Stage Thresholds" in t for t in texts)`	Pass	0.000s
With timeseries shows current status section	`any("Current Status vs Thresholds" in t for t in texts)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Lines 251-254: generate_basic_report() body on real generator.	`path.exists()`; `path.suffix == ".pdf"`	Pass	0.118s
Line 315: analysis branch in gauge_generator.generate_gauge_report.	`path.exists()`	Pass	0.003s
Lines 321-328: auto_open=True — open attempt caught, does not raise.	`path.exists()`	Pass	0.020s
Lines 307-311: generate_gauge_report() in gauge_generator.py, basic type.	`path.exists()`	Pass	0.021s
Lines 316-317: unknown type falls back to generate_report.	`path.exists()`	Pass	0.009s
Line 313: monitoring branch in gauge_generator.generate_gauge_report.	`path.exists()`	Pass	0.003s
Line 119: generate_report(pages_to_include=None) calls _auto_select_pages.	`path.exists()`	Pass	0.009s
Lines 102-103 in __init__.py: auto_open=True, open returns False.	`result == fake_pdf`	Pass	0.020s
Lines 285-286: configure(panel_height=...) updates panel_height.	`obj.panel_height == "600px"`; `obj.panel_height == "700px"` (+1 more)	Pass	0.000s
Lines 283-284: configure(panel_width=...) updates panel_width.	`obj.panel_width == "800px"`; `obj.panel_width == "900px"` (+1 more)	Pass	0.000s
Line 290: get_statistics() returns dict with panel_width and panel_height.	`stats["panel_width"] == "700px"`; `stats["panel_height"] == "500px"` (+4 more)	Pass	0.000s
Lines 480-481: configure(panel_height=...) updates panel_height.	`obj.panel_height == "600px"`; `obj.panel_height == "700px"` (+1 more)	Pass	0.000s
Lines 478-479: configure(panel_width=...) updates panel_width.	`obj.panel_width == "800px"`; `obj.panel_width == "900px"` (+1 more)	Pass	0.000s
Line 485: get_statistics() returns dict.	`stats["panel_width"] == "700px"`; `stats["panel_height"] == "500px"` (+4 more)	Pass	0.000s
Empty return period levels no extra table	`len(tables) >= 1`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
No hazard curve shows fallback	`any("No hazard" in t or "hazard" in t.lower() for t in te...`	Pass	0.000s
Returns list	`isinstance(result, list)`	Pass	0.000s
With hazard curve basic	`len(result) > 2`	Pass	0.000s
With return period levels renders table	`len(tables) >= 2`	Pass	0.000s
GET /api/v1/gauges/¡id¿/hazard returns hazard data for known gauge.	`response.status_code == 200`; `data['status'] == 'success'` (+6 more)	Pass	0.003s
GEV parameters should be present and numeric.	`gev['location'] is not None`; `gev['scale'] is not None` (+3 more)	Pass	0.003s
GET /api/v1/gauges/¡id¿/hazard returns 404 for nonexistent gauge.	`response.status_code == 404`	Pass	0.001s
GET /api/v1/gauges/¡id¿/hazard returns 404 for gauge without hazard data.	`response.status_code == 404`	Pass	0.003s
GET /api/v1/gauges/¡id¿/storms returns storm data for known gauge.	`response.status_code == 200`; `data['status'] == 'success'` (+2 more)	Pass	0.006s
calculate_statistics_from_levels via class.	`isinstance(stats, dict)`	Pass	0.002s
Calculate statistics via class	`isinstance(stats, dict)`	Pass	0.000s
Custom years	`ghd.years_of_history == 30`	Pass	0.000s
Default years	`ghd.years_of_history == 50`	Pass	0.000s
Filter by years explicit	`isinstance(result, list)`	Pass	0.000s
Filter by years none	`result == flows`	Pass	0.000s
generate_from_gauge_portfolio with explicit years overrides default.	`result["years_included"] == 3`	Pass	0.186s
generate_from_gauge_portfolio via class method.	`isinstance(result, dict)`; `result["years_included"] == 5`	Pass	0.197s
When years=None, uses years_of_history.	`result["years_included"] == 10`	Pass	0.002s
Generate from nrfa via class	`output.exists()`; `isinstance(result, dict)`	Pass	0.002s
generate_synthetic_timeseries via class method.	`isinstance(result, list)`; `len(result) > 0`	Pass	0.003s
Generates per gauge files	`gaugets_dir.exists()`; `len(list(gaugets_dir.glob("GAUGE-*.json"))) >= N_GAUGES` (+2 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Output has daily observations	"daily_observations" in data; len(data["daily_observations"]) > 0	Pass	0.016s
Parse nrfa csv via class	len(flows) == 10	Pass	0.001s
Statistics computed	"statistics" in data; "mean_level" in data["statistics"] (+1 more)	Pass	0.002s
Cdm available flag is bool	hasattr(mod, "CDM_AVAILABLE"); isinstance(mod.CDM_AVAILABLE, bool)	Pass	0.000s
Response has count and gauges keys	"count" in data; "gauges" in data (+1 more)	Pass	0.001s
Returns 200 with no gaugehd files	r.status_code == 200; data["status"] == "success" (+1 more)	Pass	0.001s
Returns correct count when files present	r.status_code == 200; data["status"] == "success" (+2 more)	Pass	0.001s
gauge.json must contain at least one gauge.	len(ids) > 0	Pass	0.003s
Gauge IDs in gaugehc.json must exist in gauge.json.	not orphaned	Pass	0.002s
Gauge IDs in gaugets/ files must exist in gauge.json.	not orphaned	Pass	0.001s
Each gaugets file's internal gauge_id must match its filename.	internal_id == gf.stem	Pass	0.002s
gaugets/ files must reference gauge IDs from gauge.json.	len(missing) == 0	Pass	2.904s
Every gauge in gauge.json must have a hazard curve.	len(missing) == 0	Pass	0.013s
gaugehc.json must not contain IDs absent from gauge.json.	len(orphan) == 0	Pass	0.012s
Every gauge referenced by a trade must exist in gauge.json.	len(missing) == 0	Pass	0.007s
Every gauge with trades must have a hazard curve (for PRS pricing).	len(missing) == 0	Pass	0.015s
gauge_random.py must NOT use GAUGE-{i:03d} style sequential IDs.	'GAUGE-{i:' not in src; 'GAUGE-{index' not in src (+3 more)	Pass	0.000s
gauge_random.py uses hashlib for deterministic ID generation.	'hashlib' in src	Pass	0.000s
gauge_random.py contains the GAUGE- prefix.	'GAUGE-' in src	Pass	0.001s
uuid4()[:8] always produces 8 lowercase hex chars — 100 samples.	GAUGE_ID_RE.match(gauge_id); PROP_ID_RE.match(prop_id) (+2 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Accepts valid uuid derived ids	`GAUGE_ID_RE.match('GAUGE-9042bd95'); GAUGE_ID_RE.match('GAUGE-a1b2c3d4')` (+2 more)	Pass	0.000s
Rejects non hex chars	`not GAUGE_ID_RE.match('GAUGE-9042xd95'); not GAUGE_ID_RE.match('GAUGE-9042gd95')`	Pass	0.000s
GAUGE-001, GAUGE-002 are NOT protocol-compliant.	`not GAUGE_ID_RE.match('GAUGE-001'); not GAUGE_ID_RE.match('GAUGE-002')` (+5 more)	Pass	0.000s
Gauge IDs must be lowercase; GAUGE-9042BD95 is invalid.	`not GAUGE_ID_RE.match('GAUGE-9042BD95'); not GAUGE_ID_RE.match('GAUGE-A1B2C3D4')`	Pass	0.000s
Rejects wrong length	`not GAUGE_ID_RE.match('GAUGE-9042bd9'); not GAUGE_ID_RE.match('GAUGE-9042bd951')` (+1 more)	Pass	0.000s
Gauge id present in all responses	`"gauge_id" not in resp; gr.get("gauge_id") == "GAUGE-deadbeef"`	Pass	0.006s
Multiple gauges have correct ids	`found_ids == {"GAUGE-00000001", "GAUGE-00000002", "GAUGE-...`	Pass	0.003s
Corrupted gauge json does not crash	`result["processing_stats"]["successful_properties"] == 2`	Pass	0.519s
Gauge id map empty when no gauge json	`isinstance(gen._gauge_id_map, dict)`	Pass	0.510s
When gauge.json exists with proper structure, gauge IDs are loaded.	`0 in gen._gauge_id_map; gen._gauge_id_map[0] == "GAUGE-111"`	Pass	0.548s
Also supports 'gauges' as the top-level key.	`gen._gauge_id_map.get(0) == "G-AAA"`	Pass	0.512s
Get most recent reading.	`current == flood_event_readings[-1]['level']`	Pass	0.000s
Detect falling water levels.	`trend == "FALLING"`	Pass	0.000s
Detect rising water levels.	`trend == "RISING"`	Pass	0.000s
Detect stable water levels.	`trend == "STABLE"`	Pass	0.000s
Detect peak level in time series.	`peak['level'] > 6.0; 'timestamp' in peak`	Pass	0.000s
Peak should occur around hour 24.	`20 < hours_to_peak < 28`	Pass	0.000s
Every gaugets file should have 168-hour flood simulation.	`len(missing) == 0`	Pass	0.001s
Every gaugets file should have storm_responses with responses.	`len(missing) == 0`	Pass	0.004s
At least some storms should exceed alert threshold.	`total_alert > 0`	Pass	1.429s

Test Description	Acceptance Criteria	Result	Time
Synthetic gaugets specifically should have storm responses.	`len(synth_gaugets) > 0`; `len(responses) > 0`	Pass	1.013s
Count	`GaugeLoader(tmp_path).count() == 4`	Pass	0.005s
Floodgauges key also works	`GaugeLoader(tmp_path).count() == 2`	Pass	0.034s
Load all returns list	`len(GaugeLoader(tmp_path).load_all()) == 3`	Pass	0.073s
Load missing file returns empty	`GaugeLoader(tmp_path).load_all() == []`	Pass	0.001s
Test GaugeID extraction from nested structure.	`gauge_id is not None`; `gauge_id.startswith(("GAUGE-", "SYNTH-"))`	Pass	0.001s
Test gauge summary generation.	`len(summaries) == 3`; `'gaugeId' in summary` (+5 more)	Pass	0.001s
Test loading gauge portfolio.	`len(gauges) == 3`	Pass	0.001s
Caching works	`loader.load_all() is loader.load_all()`	Pass	0.001s
Clear cache	`not loader._cache_valid`	Pass	0.001s
Force reload	`len(loader.load_all(force_reload=True)) == 5`	Pass	0.005s
Test getting active gauges.	`len(active) == 2`; `status == "Active"`	Pass	0.001s
Test getting gauge coordinates.	`coords is not None`; `lat == pytest.approx(51.4309, rel=1e-4)` (+1 more)	Pass	0.001s
Test getting flood stage thresholds.	`stages is not None`; `stages['ActionStage'] == 4.5` (+3 more)	Pass	0.001s
Test getting gauge name by ID.	`name == "Thames at Teddington"`	Pass	0.002s
Test getting name for non-existent gauge.	`name is None`	Pass	0.001s
Exists false	`GaugeLoader(tmp_path).exists("GAUGE-999") is False`	Pass	0.001s
Exists true	`GaugeLoader(tmp_path).exists("GAUGE-000") is True`	Pass	0.003s
Find by id missing returns none	`GaugeLoader(tmp_path).find_by_id("GAUGE-999") is None`	Pass	0.005s
Find by id returns entity	`result is not None`; `result["FloodGauge"]["Header"]["GaugeID"] == "GAUGE-001"`	Pass	0.001s
List all returns summaries	`len(summaries) == 2`; `"gaugeId" in summaries[0]`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Test finding gauge by ID.	gauge is not None; gauge['FloodGauge']['Header']['GaugeName'] == "Thames at ...	Pass	0.001s
Test finding gauge by name.	gauge is not None; gauge_loader.get_entity_id(gauge) == "GAUGE-001"	Pass	0.001s
Test name search is case insensitive.	gauge is not None	Pass	0.001s
Test finding gauges by owner.	len(ea_gauges) == 2; len(usgs_gauges) == 1	Pass	0.001s
Test finding active gauges.	len(active) == 2	Pass	0.001s
Test finding inactive gauges.	len(inactive) == 1; gauge_loader.get_entity_id(inactive[0]) == "GAUGE-003"	Pass	0.001s
Test finding river gauges.	len(river) == 2	Pass	0.001s
Test finding tidal gauges.	len(tidal) == 1	Pass	0.001s
Test finding gauges within radius.	len(nearby) >= 1; len(nearby) >= 2	Pass	0.002s
Test radius search with no matches.	len(nearby) == 0	Pass	0.001s
Get status returns dict	"entity_name" in status; "path" in status (+1 more)	Pass	0.000s
Distance ¡ 100 → 'Very close to Thames'.	"Very close to Thames" in str(result)	Pass	0.001s
Distance == 0 → 'On Thames River' and 'Directly on Thames'.	"On Thames River" in str(result)	Pass	0.001s
Empty data does not crash	isinstance(result, list)	Pass	0.000s
Flood risk section	any("Flood Risk" in t for t in texts)	Pass	0.000s
Geographic coordinates section	any("Geographic Coordinates" in t for t in texts)	Pass	0.000s
Has tables	any(isinstance(e, Table) for e in result)	Pass	0.000s
Latitude 51.4–51.5 → 'Central London area'.	"Central London area" in str(result)	Pass	0.001s
Latitude ¿ 51.5 → 'North London area'.	"North London area" in str(result)	Pass	0.001s
Latitude ¡ 51.4 → 'South London area'.	"South London area" in str(result)	Pass	0.001s
No thames info does not crash	isinstance(result, list)	Pass	0.000s
Positioning summary section	any("Positioning Summary" in t for t in texts)	Pass	0.000s
Returns list	isinstance(result, list); len(result) > 0	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Risk category in location profile	`"High flood risk location" in str(result)`	Pass	0.001s
Risk score shown as 'X/10'.	`"7.5/10" in str(result)`	Pass	0.001s
Thames context section	`any("Thames River Context" in t for t in texts)`	Pass	0.000s
Title present	`any("Geographic Location" in t for t in texts)`	Pass	0.000s
Lines 89-90: exception in create_gauge_mapping raises ValueError.	—	Pass	0.000s
Line 100: get_nrfa_fields returns list of NRFA field names.	`isinstance(fields, list); len(fields) == 15` (+3 more)	Pass	0.000s
Empty data does not crash	`isinstance(result, list)`	Pass	0.000s
Empty readings list does not crash	`isinstance(result, list)`	Pass	0.000s
Get gauge id missing	`page._get_gauge_id({}) == "Unknown Gauge"`	Pass	0.000s
Get gauge id present	`page._get_gauge_id(_make_gauge()) == "GAUGE-001"`	Pass	0.000s
Has tables	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
Historical statistics section	`any("Historical Measurement Statistics" in t for t in texts)`	Pass	0.000s
Measurement configuration section	`any("Measurement Configuration" in t for t in texts)`	Pass	0.000s
No timeseries shows message	`any("No current timeseries data" in t for t in texts)`	Pass	0.000s
Operational status section	`any("Operational Status" in t for t in texts)`	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0`	Pass	0.000s
Title present	`any("Current Measurements" in t for t in texts)`	Pass	0.000s
Current readings table is added when timeseries provided.	`len(tables_with) >= len(tables_without)`	Pass	0.001s
With timeseries shows current readings	`any("Current Readings" in t for t in texts)`	Pass	0.000s
Calculate level range.	`level_range > 2.0`	Pass	0.000s
Calculate maximum level.	`max_level > 6.0`	Pass	0.000s
Calculate mean water level.	`2.5 < mean < 3.5`	Pass	0.000s
Calculate minimum level.	`min_level > 0`	Pass	0.000s
Calculate maximum rate of rise.	`rate > 0`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
When name is empty, uses area_name.	"Westminster" in result; "Thames" in result	Pass	0.000s
No name key falls back to area	"Putney" in result	Pass	0.000s
No hazard curve shows fallback	any("No hazard" in t or "hazard" in t.lower() for t in te...	Pass	0.000s
Returns list	isinstance(result, list)	Pass	0.000s
With nonzero rates renders spreads	len(tables) >= 1	Pass	0.000s
With zero rates renders na	len(tables) >= 1	Pass	0.000s
Basic gauge report produces a valid PDF file.	report_path.exists(); report_path.suffix == '.pdf' (+1 more)	Pass	0.025s
Full gauge report produces a valid PDF.	report_path.exists(); report_path.suffix == '.pdf'	Pass	0.013s
Output directory is created automatically.	new_dir.exists(); report_path.exists()	Pass	0.025s
Generated filename includes the gauge ID.	'GAUGE-test0001' in report_path.name	Pass	0.025s
Gauge report PDF should be non-trivial in size.	report_path.stat().st_size > 5000	Pass	0.025s
Generated file is a valid PDF.	header == b'%PDF-'	Pass	0.032s
Report handles minimal gauge data without crashing.	report_path.exists(); report_path.stat().st_size > 0	Pass	0.020s
Daily minus 15	r['reliability_score'] == 80	Pass	0.000s
Decommissioned	r['reliability_score'] == 0; r['category'] == "Unreliable"	Pass	0.000s
Fully operational	r['reliability_score'] == 95; r['category'] == "Highly Reliable"	Pass	0.000s
Hourly minus 5	r['reliability_score'] == 90	Pass	0.000s
Maintenance required	r['reliability_score'] == 70; r['category'] == "Reliable"	Pass	0.000s
Monthly minus 50	r['reliability_score'] == 20; r['category'] == "Unreliable"	Pass	0.000s
Realtime no adjustment	rt['reliability_score'] == base['reliability_score']	Pass	0.000s
Result keys	'reliability_score' in r; 'category' in r (+1 more)	Pass	0.000s
Score clamped at zero	r['reliability_score'] == 0	Pass	0.000s
Temporarily offline	r['reliability_score'] == 30; r['category'] == "Low Reliability"	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Unknown status	`r['reliability_score'] == 50`; `r['category'] == "Moderately Reliable"`	Pass	0.000s
Weekly minus 30	`r['reliability_score'] == 65`	Pass	0.000s
Generator initialises with output directory.	`gen.output_dir == tmp_path`	Pass	0.001s
Generator lists all registered gauge page modules.	`'title_overview' in pages`; `'sensor_details' in pages` (+7 more)	Pass	0.001s
Validates known gauge page names.	`'title_overview' in valid`; `'bogus' in invalid`	Pass	0.001s
Line 163 (else branch): no time-series or hazard -¿ no analysis pages.	`not any(p in pages for p in analysis_pages)`	Pass	0.001s
Line 163 (if branch): gauge_data.get('hazard_curve') -¿ include analysis.	`"risk_assessment" in pages`	Pass	0.001s
Lines 269-270: generate_analysis_report creates PDF.	`path.exists()`	Pass	0.003s
Lines 208-210: exception in page -¿ logged, continue to next page.	`isinstance(elements, list)`	Pass	0.002s
Lines 190-191: unknown page -¿ warning logged, skipped.	`isinstance(elements, list)`	Pass	0.001s
Lines 176-177: missing GaugeID -¿ uses 'unknown' in filename.	`"unknown" in name`; `name.endswith(".pdf")`	Pass	0.001s
Line 175: gauge_id extracted for filename.	`"GAUGE-001" in name`	Pass	0.001s
Lines 261-263: generate_monitoring_report creates PDF.	`path.exists()`	Pass	0.003s
Line 313: get_page_categories returns dict.	`isinstance(cats, dict)`; `"gauge_info" in cats`	Pass	0.001s
Line 279: list_available_pages returns list of strings.	`isinstance(pages, list)`; `"title_overview" in pages`	Pass	0.001s
Lines 59-60: output_dir=None -¿ uses config.get_gauge_reports_dir().	`gen.output_dir == gauge_dir`	Pass	0.001s
Line 315: validate_pages with all valid pages.	`"title_overview" in valid`; `invalid == []`	Pass	0.001s
validate_pages with some invalid page names.	`"title_overview" in valid`; `"nonexistent_page" in invalid`	Pass	0.001s
Categories dict initialized	`"gauge_info" in gen.categories`; `"operational" in gen.categories` (+2 more)	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Explicit output dir path	`isinstance(gen.output_dir, Path)`	Pass	0.001s
Explicit output dir string	`gen.output_dir == tmp_path`	Pass	0.001s
No output dir uses config	`gen.output_dir == gauge_dir`	Pass	0.001s
Output dir created	`new_dir.exists()`	Pass	0.001s
Pages dict initialized	`expected <= set(gen.pages.keys())`	Pass	0.001s
Excludes timeseries when requested	`"intensity_timeseries" not in d`; `"level_timeseries" not in d`	Pass	0.000s
Flooded is bool	`isinstance(d["flooded"], bool)`; `d["flooded"] is True`	Pass	0.000s
Has required keys	`"gauge_id" in d`; `"storm_id" in d` (+2 more)	Pass	0.000s
Includes timeseries by default	`"intensity_timeseries" in d`; `"level_timeseries" in d` (+1 more)	Pass	0.000s
Returns dict	`isinstance(d, dict)`; `isinstance(d, dict)`	Pass	0.000s
Values rounded	`d["peak_level"] == 5.2`; `d["accumulation"] == 4.5`	Pass	0.000s
Generates elements list	`isinstance(elements, list)`; `len(elements) > 0`	Pass	0.000s
No timeseries shows placeholder	`any("No current timeseries" in t for t in texts)`	Pass	0.000s
With timeseries data	`isinstance(elements, list)`; `len(elements) > 0`	Pass	0.000s
Test POST /api/v1/gauges/report without gaugeId.	`response.status_code == 400`; `'Gauge ID is required' in data['message']`	Pass	0.000s
Test GET /api/v1/gauges/¡id¿.	`response.status_code == 200`; `data['status'] == 'success'` (+1 more)	Pass	0.002s
Test GET /api/v1/gauges/¡id¿ with invalid ID.	`response.status_code == 404`	Pass	0.002s
Test GET /api/v1/gauges/¡id¿/statistics.	`response.status_code == 200`; `data['status'] == 'success'` (+1 more)	Pass	0.003s
Test GET /api/v1/gauges/¡id¿/timeseries.	`response.status_code == 200`; `data['status'] == 'success'` (+1 more)	Pass	0.003s
Test legacy GET /api/v1/list_gauges endpoint.	`response.status_code == 200`; `data['count'] == 3`	Pass	0.001s
Test GET /api/v1/gauges.	`response.status_code == 200`; `data['status'] == 'success'` (+2 more)	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Empty data does not crash	`isinstance(result, list)`	Pass	0.000s
Gauge id in subtitle	`any("GAUGE-001" in t for t in texts)`	Pass	0.001s
Gauge information section	`any("Gauge Information" in t for t in texts)`	Pass	0.001s
Get gauge id missing	`page._get_gauge_id({}) == "Unknown Gauge"`	Pass	0.000s
Get gauge id present	`page._get_gauge_id(_make_gauge()) == "GAUGE-001"`	Pass	0.000s
Has tables	`any(isinstance(e, Table) for e in result)`	Pass	0.001s
Historical sensor statistics section	`any("Historical Sensor Statistics" in t for t in texts)`	Pass	0.001s
Location information section	`any("Location Information" in t for t in texts)`	Pass	0.001s
Measurement configuration section	`any("Measurement Configuration" in t for t in texts)`	Pass	0.001s
Missing SensorDetails → empty tables, no exception.	`isinstance(result, list)`	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0`	Pass	0.001s
Title present	`any("Sensor Details" in t for t in texts)`	Pass	0.001s
With timeseries does not crash	`isinstance(result, list)`	Pass	0.001s
Level at alert threshold.	`status == "ALERT"`	Pass	0.000s
Level between alert and warning.	`status == "ALERT"`	Pass	0.000s
Level below alert = normal.	`status == "NORMAL"`	Pass	0.000s
Level at severe threshold.	`status == "SEVERE"`	Pass	0.000s
Level at warning threshold.	`status == "WARNING"`	Pass	0.000s
Lines 270-271: configure(panel_height=...) updates panel_height.	`obj.panel_height == "600px"; obj.panel_height == "700px"` (+1 more)	Pass	0.000s
Lines 268-269: configure(panel_width=...) updates panel_width.	`obj.panel_width == "800px"; obj.panel_width == "900px"` (+1 more)	Pass	0.000s
Line 275: get_statistics() returns dict.	`stats["panel_width"] == "700px"; stats["panel_height"] == "500px"` (+4 more)	Pass	0.000s
When nearest_gauges lacks gauge_info, thresholds default to empty.	`"gauge_thresholds" in ng; isinstance(ng["gauge_thresholds"], dict)`	Pass	0.089s
Gauge thresholds has alert level	`thresholds["alert_level"] == 3.5`	Pass	0.089s

Test Description	Acceptance Criteria	Result	Time
Gauge thresholds has severe level	`thresholds["severe_level"] == 5.0`	Pass	0.089s
Gauge thresholds has warning level	`thresholds["warning_level"] == 4.5`	Pass	0.088s
Gauge thresholds key exists	`"gauge_thresholds" in ng`	Pass	0.089s
Gauge thresholds alert level	`thresholds["alert_level"] == 3.5`	Pass	0.002s
Gauge thresholds present	`"gauge_thresholds" in ng`	Pass	0.002s
Gauge thresholds severe level	`thresholds["severe_level"] == 5.0`	Pass	0.002s
Gauge thresholds warning level	`thresholds["warning_level"] == 4.5`	Pass	0.002s
Empty timeseries returns short list	`isinstance(result, list); len(result) > 0`	Pass	0.000s
Gauge name in title	`any("Chelsea Gauge" in t for t in texts)`	Pass	0.000s
Generates per gauge files	`gaugets_dir.exists(); len(list(gaugets_dir.glob("GAUGE-*.json"))) >= N_GAUGES` (+2 more)	Pass	0.000s
Missing gauge name does not crash	`isinstance(result, list)`	Pass	0.000s
No timeseries message included	`any("No time series data" in t for t in texts)`	Pass	0.000s
No timeseries returns short list	`len(result) > 0`	Pass	0.000s
Output file has readings	`len(gauge_files) > 0; "flood_simulation" in data` (+1 more)	Pass	0.001s
Requires gauge portfolio	—	Pass	0.000s
Returns list	`isinstance(result, list)`	Pass	0.001s
Timeseries mixed statuses	`isinstance(result, list)`	Pass	0.084s
Readings with non-zero thresholds exercising fill_between paths.	`isinstance(result, list)`	Pass	0.084s
With timeseries returns more elements	`len(result) >= len(empty_result)`	Pass	0.089s
With timeseries table included	`len(tables) >= 1`	Pass	0.084s
DEFAULT_PARAMS and generate() default must both be 168 hours (7 days).	`GaugeTimeSeriesGenerator.DEFAULT_PARAMS["simulation_hours"]`	Pass	0.000s
generate(simulation_hours) default parameter must be 168.	`default == 168`	Pass	0.000s
Convenience function generate_gaugets() default must be 168.	`default == 168`	Pass	0.000s
peak_hour_max must be strictly less than simulation_hours.	`p["peak_hour_max"] < p["simulation_hours"]`	Pass	0.000s
peak_hour_min must be less than peak_hour_max.	`p["peak_hour_min"] < p["peak_hour_max"]`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
168-hour simulation produces significantly more readings than 60-hour.	`files60 and files168; count168 > count60`	Pass	0.512s
Lines 127-130: configure() updates a known sim_param key.	`gen.sim_params['simulation_hours'] == 48`	Pass	0.000s
Lines 127-129: unknown key is silently ignored (if branch is False).	`'nonexistent_param' not in gen.sim_params`	Pass	0.000s
generate(168) should produce ~168 readings per gauge.	`len(files) > 0; sim.get("simulation_hours") == 168` (+1 more)	Pass	0.487s
Explicit override stores the passed value in simulation_hours metadata.	`sim.get("simulation_hours") == 60`	Pass	0.302s
Lines 157-159: gauge.json absent -¿ FileNotFoundError re-raised.	—	Pass	0.000s
simulation_hours is written into each gaugets JSON.	`"simulation_hours" in data["flood_simulation"]`	Pass	0.463s
Line 175: stale GAUGE-*.json files in gaugets/ are deleted before writing.	`stale.exists(); not stale.exists()`	Pass	0.196s
Unparseable install date is silently skipped.	`isinstance(result, list)`	Pass	0.000s
Empty data does not crash	`isinstance(result, list)`	Pass	0.000s
Gauge id in output	`any("GAUGE-001" in t for t in texts)`	Pass	0.000s
Gauge overview section header	`any("Gauge Overview" in t for t in texts)`	Pass	0.000s
Gauge summary section header	`any("Gauge Summary" in t for t in texts)`	Pass	0.000s
Get gauge id missing	`page._get_gauge_id({}) == "Unknown Gauge"`	Pass	0.000s
Get gauge id present	`page._get_gauge_id(_make_gauge()) == "GAUGE-001"`	Pass	0.000s
Has tables	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
Valid install date triggers gauge age row in summary.	`isinstance(result, list); len(result) > 0`	Pass	0.001s
No install date does not crash	`isinstance(result, list)`	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0`	Pass	0.000s
Title paragraph present	`any("Comprehensive Flood Gauge Analysis Report" in t for ...`	Pass	0.000s
Unknown gauge fallback	`any("Unknown Gauge" in t for t in texts)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
With timeseries does not crash	`isinstance(result, list)`	Pass	0.000s
Gauge trace includes classifier summary	`len(clf_steps) >= 2`	Pass	0.004s
Gauge trace includes gaugehd	`"gaugehd" in steps`	Pass	0.013s
Gauge trace includes market state	`len(blotter_steps) >= 2`	Pass	0.005s
Gauge trace includes sequence gauge	`len(stressm_steps) >= 2`	Pass	0.007s
When per-gauge file doesn't exist, falls back to _index.json.	`any("_index.json" in r["file"] for r in stressm_steps)`	Pass	0.002s
When sequence_gauge/ dir doesn't exist, falls back to legacy file.	`any("sequence_gauge_summary" in r["file"] for r in stress...`	Pass	0.003s
Lines 68-69: exception returns validation_error dict.	`"validation_error" in result`; `isinstance(result["validation_error"], list)` (+2 more)	Pass	0.000s
Lines 61,64: invalid menu field value appended to sensor_errors.	`"SensorDetails" in errors`; `any("DataSourceType" in e for e in errors["SensorDetails"])`	Pass	0.000s
Multiple invalid menu fields all collected.	`"SensorDetails" in errors`; `len(errors["SensorDetails"]) == 3`	Pass	0.000s
Valid menu field values produce no SensorDetails errors.	`"SensorDetails" not in errors`	Pass	0.000s
Curve points present	`not missing`	Pass	0.000s
Each gauge has required fields	`not missing`	Pass	0.000s
File exists	`GAUGEHC_PATH.exists()`; `MARKET_STATE.exists()`	Pass	0.000s
alert ¡ warning ¡ severe for every gauge.	`not violations`	Pass	0.000s
Gauge count at least 40	`count >= 40`	Pass	0.000s
Gauge ids are well formed	`not bad`	Pass	0.000s
Has hazard curves key	`"hazard_curves" in gaugehc`	Pass	0.000s
alert ¿ warning ¿ severe (higher trigger = lower probability).	`not violations`	Pass	0.000s
Hazard rates positive	`not violations`	Pass	0.000s
Term structure present	`not missing`	Pass	0.000s
Lines 73-77: main() with no GDF files logs info.	—	Pass	0.000s
Lines 70-71: main() processes files, handles errors.	—	Pass	0.001s
Line 37: output_dir defaults to input_dir when None.	`result == []`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Lines 46-47: error processing a CSV file is logged and skipped.	`result == []`	Pass	0.001s
directory.main() is callable.	`callable(main)`	Pass	0.000s
Each file has required keys	`not bad`	Pass	13.150s
Flood simulation has readings	`not bad`	Pass	12.874s
Gauge id field matches filename	`not bad`	Pass	13.094s
Gaugets dir exists	`GAUGETS_DIR.exists()`	Pass	0.000s
Gaugets gauge id format	`not bad`	Pass	0.001s
Gaugets not empty	`gaugets_files`	Pass	0.000s
simulation_hours should match number of reading entries.	`not mismatches`	Pass	13.117s
Line 72: datetime -¿ isoformat string.	`"2026-01-01" in result`	Pass	0.000s
Lines 75-76: np.float32 is np.floating but not Python float -¿ default() called.	`"2.5" in result`	Pass	0.000s
Lines 73-74: np.integer -¿ int.	`"42" in result`	Pass	0.000s
Lines 77-78: np.ndarray -¿ list via tolist().	`"1.0" in result`	Pass	0.000s
Line 79: unknown type -¿ super().default() -¿ TypeError.	—	Pass	0.000s
gaugets readings using 'level' key (alternate field name) are extracted.	`resp.status_code == 200; max(water_levels) > min(water_levels)`	Pass	0.007s
gaugets readings using 'waterLevel' key are correctly extracted.	`resp.status_code == 200; max(water_levels) > min(water_levels)` (+1 more)	Pass	0.007s
When no gaugets file exists, stress run returns 404.	`resp.status_code == 404; "No gaugets data" in data.get('message', '')`	Pass	0.005s
Lines 241-244: __main__ block runs generate_gaugets(168) and logs results.	`(tmp_path / 'gaugets').exists()`	Pass	0.004s
Random module peak_hour_max must be ¡ simulation_hours.	`DEFAULT_PARAMS["peak_hour_max"] < DEFAULT_PARAMS["simulat...`	Pass	0.000s
gaugets_random.DEFAULT_PARAMS simulation_hours must be 168.	`DEFAULT_PARAMS["simulation_hours"] == 168`	Pass	0.000s
The generator uses input_dir.name as the catchment name.	`gen.input_dir.name == 'thames'`	Pass	0.000s
Daily flows filtered by years	`len(result_3yr["daily_flows"]) > len(result_1yr["daily_fl...`	Pass	0.026s
Explicit gauge id overrides metadata	`result["station_metadata"]["station_id"] == "OVERRIDE-ID"`	Pass	0.015s

Test Description	Acceptance Criteria	Result	Time
Even with completely empty input, PDF should still generate.	`result.exists()`; `result.stat().st_size > 0`	Pass	0.007s
Minimal data (empty arrays) should produce valid PDF.	`result.exists()`	Pass	0.012s
Output file written	`output_path.exists()`; `json.load(f)["data_type"] == "GaugeHistoricalDaily"`	Pass	0.017s
Output path creates parent dirs	`result.exists()`	Pass	0.023s
With populated data, PDF should have multiple pages (from PageBreaks).	`out.stat().st_size > 5000`	Pass	0.021s
Pdf starts with magic bytes	`f.read(5) == b'%PDF-'`	Pass	0.027s
Produces pdf file	`result == out`; `out.exists()` (+1 more)	Pass	0.023s
Returns dict with required keys	`key in result`	Pass	0.015s
Returns path	`isinstance(result, Path)`	Pass	0.020s
Avg basis is float	`isinstance(stats["avg_basis_bps"], float)`	Pass	0.186s
Avg transmission rate between 0 and 1	`0.0 <= stats["avg_transmission_rate"] <= 1.0`	Pass	0.098s
Num storms in stats	`"num_storms" in stats`; `stats["num_storms"] > 0`	Pass	0.096s
Total flood events counted	`stats["total_flood_events"] == 7`	Pass	0.183s
No gauges in portfolio returns empty list.	`result == []`	Pass	0.002s
Line 50: error path extracts GaugeID from FloodGauge.Header.	`len(result) == 0`; `"GAUGE-BADONE" in caplog.text`	Pass	0.002s
Line 50: completely empty entry uses UNKNOWN as gauge_id.	`len(result) == 0`; `"UNKNOWN" in caplog.text`	Pass	0.001s
Good gauges succeed even when bad ones fail.	`len(result) == 1`	Pass	0.185s
Line 60-61: house with 3 bedrooms returns 1 or 2 bathrooms.	`results.issubset({1, 2})`; `len(results) == 2`	Pass	0.000s
Line 62-63: house with 4+ bedrooms returns 2 or 3 bathrooms.	`results.issubset({2, 3})`; `len(results) == 2`	Pass	0.000s
5 bedrooms also takes the 4+ path (line 62-63).	`val in (2, 3)`	Pass	0.000s
Advised flag	`isinstance(generate_boolean_value("AdvisedFlag", self._fd...`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Btl commercial purpose	`generate_boolean_value("BusinessOrCommercialPurpose", sel...; generate_boolean_value("BusinessOrCommercialPurpose", sel...`	Pass	0.000s
Default flag from data	`generate_boolean_value("DefaultFlag", self._fd(is_default...; generate_boolean_value("DefaultFlag", self._fd(is_default...`	Pass	0.000s
Execution only flag	`isinstance(generate_boolean_value("ExecutionOnlyFlag", se...`	Pass	0.000s
First time buyer btl always false	`generate_boolean_value("FirstTimeBuyerFlag", self._fd(mor...`	Pass	0.000s
First time buyer expensive property	`isinstance(result, bool)`	Pass	0.000s
First time buyer normal property	`isinstance(result, bool)`	Pass	0.000s
In arrears flag	`generate_boolean_value("InArrearsFlag", self._fd(is_in_ar...`	Pass	0.000s
Mmr compliant	`generate_boolean_value("MMRCompliantFlag", self._fd()) is...`	Pass	0.000s
Stress compliant	`generate_boolean_value("StressTestCompliantFlag", self._f...`	Pass	0.000s
Unknown field returns bool	`isinstance(generate_boolean_value("UnknownFlag", self._fd()),...`	Pass	0.000s
Count respected	`len(result["data"]) == 3`	Pass	0.000s
Returns dict	`isinstance(result, dict); "data" in result`	Pass	0.000s
Fair section 50 to 75 gets enhancement rec	`any("OtherSection" in rec for rec in r)`	Pass	0.000s
Good sections maintain recommendation	`any("Maintain" in rec or "maintain" in rec for rec in r)`	Pass	0.000s
Max 5 recommendations	`len(r) <= 5`	Pass	0.000s
Poor section under 50 gets recommendation	`any("SomeSection" in rec for rec in r)`	Pass	0.000s
Returns list	`isinstance(self._page().generate_elements(make_property(...); isinstance(r, list)`	Pass	0.000s
Application date is string	`isinstance(result, str); len(result) == 10`	Pass	0.000s
Default date not defaulted	`result is None`	Pass	0.000s
Default date when defaulted	`isinstance(result, str)`	Pass	0.000s
Disbursal date	`isinstance(result, str)`	Pass	0.000s
Maturity date	`isinstance(result, str)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Unknown field	`1 <= result <= 10; isinstance(result, str)`	Pass	0.000s
Application property valuation	`generate_decimal_value("ApplicationPropertyValuation", se...`	Pass	0.000s
Aprc initial rate	`isinstance(result, float)`	Pass	0.000s
Aprc secondary rate	`isinstance(result, float)`	Pass	0.000s
Borrower income	`generate_decimal_value("BorrowerIncome", self._fd()) == 7...`	Pass	0.000s
Current balance	`generate_decimal_value("CurrentBalance", self._fd()) == 2...`	Pass	0.000s
Current lending rate	`abs(result - 4.0) < 0.01`	Pass	0.000s
Current ltv	`abs(result - 75.0) < 0.01`	Pass	0.000s
Current payment	`generate_decimal_value("CurrentPayment", self._fd()) == 1500`	Pass	0.000s
Debt to income ratio	`isinstance(result, float)`	Pass	0.000s
Debt to income zero income	`result == 0.3`	Pass	0.000s
Default field	`isinstance(result, float)`	Pass	0.000s
Interest rate	`isinstance(result, float)`	Pass	0.000s
Loan to value ratio	`abs(result - 80.0) < 0.01`	Pass	0.000s
Ltv field	`abs(result - 75.0) < 0.01`	Pass	0.000s
Original lending rate	`abs(result - 4.0) < 0.01`	Pass	0.000s
Original loan	`generate_decimal_value("OriginalLoan", self._fd()) == 320000`	Pass	0.000s
Original loan amount	`generate_decimal_value("OriginalLoanAmount", self._fd()) ...`	Pass	0.000s
Original ltv	`abs(result - 80.0) < 0.01`	Pass	0.000s
Outstanding balance	`generate_decimal_value("OutstandingBalance", self._fd()) ...`	Pass	0.000s
Purchase value	`generate_decimal_value("PurchaseValue", self._fd()) == 40...`	Pass	0.000s
Empty page list	`elements == []`	Pass	0.002s
Empty page list	`elements == []`	Pass	0.001s
Empty page list returns empty	`elements == []`	Pass	0.001s
Exception in page continues	`isinstance(elements, list); len(elements) > 0`	Pass	0.002s
If one page raises, the remainder are still processed.	`isinstance(elements, list); len(elements) > 0`	Pass	0.001s
Exception in page module continues	`isinstance(elements, list); len(elements) > 0`	Pass	0.002s
First page no page break	`not any(isinstance(e, PageBreak) for e in elements)`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
First page no page break	`not any(isinstance(e, PageBreak) for e in elements)`	Pass	0.002s
First page no page break	`not any(isinstance(e, PageBreak) for e in elements)`	Pass	0.001s
Mixed valid and unknown pages	`len(elements) > 0`	Pass	0.001s
Mixed valid and unknown pages	`len(elements) > 0`	Pass	0.002s
Mixed valid and unknown pages	`len(elements) > 0`	Pass	0.001s
Mortgage data passed to page	—	Pass	0.002s
Multiple pages have page break	`any(isinstance(e, PageBreak) for e in elements)`	Pass	0.002s
Multiple pages have page breaks	`any(isinstance(e, PageBreak) for e in elements)`	Pass	0.002s
Multiple pages inserts page breaks	`has_break`	Pass	0.001s
No gauge id key returns message	`any("No gauge ID" in t for t in texts)`	Pass	0.000s
None timeseries passed to page	—	Pass	0.001s
Returns list	`isinstance(elements, list)`	Pass	0.002s
Returns list	`isinstance(result, list); len(result) > 0`	Pass	0.012s
Returns list	`isinstance(elements, list)`	Pass	0.002s
Returns list	`isinstance(elements, list)`	Pass	0.001s
Timeseries passed to page	—	Pass	0.002s
Unknown page skipped	`len(elements) == 0`	Pass	0.001s
Unknown page skipped	`len(elements) == 0`	Pass	0.002s
Unknown page skipped	`isinstance(elements, list); len(elements) == 0`	Pass	0.001s
With gauge id returns trading header	`any("Trading Activity" in t for t in texts)`	Pass	0.010s
When eod_date is None, uses today's date.	`snapshot['date'] == date.today().isoformat(); snapshot['eod_id'] == expected_id`	Pass	0.000s
Empty property list returns zero mortgages	`result["data"]["mortgages"] == []; result["processing_stats"]["total_mortgages"] == 0`	Pass	0.002s
Missing property file raises file not found	—	Pass	0.000s
All storm sequences	`isinstance(seq, StormSequence)`	Pass	0.000s
Catchment propagated	`seq.catchment_id == "rhine"; storm.catchment_id == "rhine"`	Pass	0.001s
Correct count	`len(small_batch) == 200`	Pass	0.000s
Different seeds differ	`types_a != types_b`	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Force sequence type doublet	`doublets > 50; seq.sequence_type in ("doublet", "isolated")`	Pass	0.002s
Force sequence type isolated	`seq.sequence_type == "isolated"; seq.num_storms == 1`	Pass	0.001s
Reproducible with same seed	`[s.sequence_id for s in a] != [s.sequence_id for s in b]; sa.num_storms == sb.num_storms` (+1 more)	Pass	0.003s
Returns list	`isinstance(small_batch, list)`	Pass	0.000s
Custom weight heavy on extreme shifts the distribution.	`all(c == "extreme" for c in cats)`	Pass	0.007s
All four default weight categories appear in the batch.	`cat in categories`	Pass	0.000s
Multi-storm sequences appear at a meaningful rate for severe+.	`multi_frac > 0.15`	Pass	0.000s
Boolean type	`isinstance(result, bool)`	Pass	0.000s
Date type	`isinstance(result, str)`	Pass	0.000s
Decimal type	`isinstance(result, (int, float))`	Pass	0.000s
Enum type	`result == "Monthly"`	Pass	0.000s
Integer type	`result == 300`	Pass	0.000s
Menu type	`result == "Monthly"`	Pass	0.000s
Non dict field def returns string	`result == "literal"`	Pass	0.000s
Non dict non string returns empty	`result == ""`	Pass	0.000s
Text type	`result == "thames"`	Pass	0.000s
Unknown type returns empty	`result == ""`	Pass	0.000s
Contains flood risk report	`"flood_risk_report" in gen._generate_filename()`	Pass	0.001s
Contains gauge id	`"GAUGE-001" in name`	Pass	0.001s
Contains property id	`"PROP-001" in name`	Pass	0.002s
Contains timestamp	`digits >= 8`	Pass	0.001s
Ends with pdf	`gen._generate_filename(gauge_data).endswith(".pdf")`	Pass	0.001s
Ends with pdf	`gen._generate_filename(prop_data).endswith(".pdf")`	Pass	0.002s
Ends with pdf	`gen._generate_filename().endswith(".pdf")`	Pass	0.001s
Missing gauge id uses unknown	`"unknown" in name; name.endswith(".pdf")`	Pass	0.001s
Missing property id uses unknown	`"unknown" in name; name.endswith(".pdf")`	Pass	0.002s
Partial path missing gauge id	`"unknown" in name`	Pass	0.001s
Partial path missing header	`"unknown" in name`	Pass	0.001s
Partial path missing header	`"unknown" in name`	Pass	0.002s
Partial path missing property id	`"unknown" in name`	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Returns string	`isinstance(name, str)`	Pass	0.001s
Returns string	`isinstance(name, str)`	Pass	0.002s
Returns string	`isinstance(name, str)`	Pass	0.001s
Timestamp in filename	`digits >= 8`	Pass	0.001s
Timestamp in filename	`digits >= 8`	Pass	0.002s
Two calls produce string names	`isinstance(n1, str) and isinstance(n2, str)`	Pass	1.102s
Estimates value when zero	`result["loan_amount"] > 0`	Pass	0.000s
Flood risk passed through	`result["flood_risk"] == "High"`	Pass	0.000s
Returns dict with mortgage id	`"mortgage_id" in result; result["mortgage_id"].startswith("MORT-")`	Pass	0.000s
Uses provided property value	`result["property_value"] == 600000`	Pass	0.000s
When output_dir=None, uses config.get_gaugehd_dir().	`isinstance(result, dict)`	Pass	0.187s
Daily observations generated	`len(result["daily_observations"]) > 0`	Pass	0.186s
Gauge metadata extracted	`result["gauge_metadata"]["gauge_id"] == "GAUGE-TEST01"`	Pass	0.202s
Gauge data without FloodGauge wrapper.	`result["gauge_metadata"]["gauge_id"] == "GAUGE-FLAT"`	Pass	0.186s
Output file created	`(tmp_path / "gauge_GAUGE-TEST01_hd.json").exists()`	Pass	0.186s
Returns dict	`isinstance(result, dict)`	Pass	0.223s
Schema version	`result["schema_version"] == "1.0"`	Pass	0.218s
When output_path=None, uses config.get_gaugehd_dir().	`isinstance(result, dict); (tmp_path / "gaugehd" / "gauge_AUTO-01_hd.json").exists()`	Pass	0.002s
Creates json file	`output.exists()`	Pass	0.002s
Gauge id override	`result["station_metadata"]["station_id"] == "GAUGE-CUSTOM"`	Pass	0.002s
Returns dict	`isinstance(result, dict)`	Pass	0.002s
Schema version in output	`result["schema_version"] == "1.0"; result["data_type"] == "GaugeHistoricalDaily"`	Pass	0.002s
Station metadata extracted	`result["station_metadata"]["station_name"] == "Thames at ...`	Pass	0.002s
Current balance does not exceed original loan	`loan["CurrentBalance"] <= loan["OriginalLoanAmount"]`	Pass	0.002s
Loan details values are positive	`loan["OriginalLoanAmount"] > 0; loan["CurrentBalance"] > 0` (+3 more)	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Ltv within valid range	`10 <= m["Mortgage"]["LoanDetails"]["LTV"] <= 100`	Pass	0.002s
Mortgage data key is list	`isinstance(result["data"]["mortgages"], list)`	Pass	0.002s
Mortgage data structure has mortgage key	`"Mortgage" in m`	Pass	0.002s
Mortgage header has required fields	`"MortgageID" in header`; `"PropertyID" in header` (+1 more)	Pass	0.002s
Mortgage ids key is list	`isinstance(result["data"]["mortgage_ids"], list)`	Pass	0.002s
Output file is valid json	`isinstance(json.load(f), dict)`	Pass	0.003s
Output json contains generation metadata	`"generated_at" in meta`; `"catchment" in meta` (+1 more)	Pass	0.004s
Output json contains mortgages key	`"mortgages" in json.load(f)`	Pass	0.003s
Processing stats contain timing	`stats["start_time"] is not None`; `stats["end_time"] is not None`	Pass	0.002s
Gauge reading returns all fields	`"gaugeId" in reading`; `"waterLevel" in reading` (+1 more)	Pass	0.000s
Gauge reading with string timestamp	`reading["timestamp"] == "2024-01-01T00:00:00"`	Pass	0.000s
report_type='analysis' calls generate_analysis_report.	—	Pass	0.001s
auto_open=True with open_pdf_file raising -¿ logs warning, does not raise.	`result is not None`	Pass	0.001s
auto_open=True tries to open the PDF (lines 95-106).	`mock_open.call_count >= 0`	Pass	0.001s
report_type='basic' calls generate_basic_report.	—	Pass	0.001s
Gauge not found returns 404	`r.status_code == 404`; `data["status"] == "error"`	Pass	0.002s
The /generate_gauge_report alias also works.	`r.status_code == 200`	Pass	0.030s
Missing gauge id returns 400	`r.status_code == 400`; `data["status"] == "error"`	Pass	0.000s
report_type='monitoring' with timeseries calls generate_monitoring_report.	—	Pass	0.001s
No json returns 400	`r.status_code in (400, 415, 500)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Sending JSON null body makes get_json() return None → line 29.	`r.status_code == 400`	Pass	0.000s
Options returns ok	`r.status_code == 200`	Pass	0.000s
Response has file path	`"file_path" in data; data["file_path"].endswith(".pdf")`	Pass	0.030s
Response has pdf base64	`"pdf_base64" in data; len(data["pdf_base64"]) > 0`	Pass	0.030s
Return value is the PDF path.	`result == tmp_path / "gauge_report.pdf"`	Pass	0.001s
Unknown report_type calls generate_report.	—	Pass	0.001s
Valid gauge generates report	`r.status_code == 200; data["status"] == "success"`	Pass	0.030s
gaugehd file exists → exercises lines 62-69.	`r.status_code == 200`	Pass	0.114s
gaugehc.json exists → exercises lines 76-82.	`r.status_code == 200`	Pass	0.062s
Analysis type calls generate analysis	—	Pass	0.001s
Auto open false returns path	`result == fake_pdf`	Pass	0.001s
Auto open true handles exception	`result is not None`	Pass	0.001s
Lines 343-344: auto_open=True + open_pdf_file succeeds → both lines covered.	`result == fake_pdf`	Pass	0.001s
Basic type calls generate basic	—	Pass	0.001s
Monitoring type with timeseries calls monitoring	—	Pass	0.001s
Monitoring without timeseries falls back	—	Pass	0.001s
Returns path	`result == fake_pdf`	Pass	0.001s
Unknown type calls generate report	—	Pass	0.000s
Lines 236-237: generate_gaugets() instantiates generator and calls generate().	`result['data']['num_gauges'] == 5`	Pass	0.000s
Creates output file	`output.exists()`	Pass	0.002s
History entries have date	`'date' in entry`	Pass	0.003s
Metadata lists gauges	`'GAUGE-005' in data['metadata']['gauges']`	Pass	0.002s
Multiple gauges	`'GAUGE-001' in data['history']; 'GAUGE-002' in data['history']`	Pass	0.002s
No eod files returns early	`not output.exists()`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Output has history	'history' in data; 'GAUGE-001' in data['history']	Pass	0.002s
Output has metadata	'metadata' in data; data['metadata']['num_snapshots'] == 2	Pass	0.003s
Tenors sorted numerically	len(entries) == 1; tenor_keys == sorted(tenor_keys, key=int)	Pass	0.003s
Two eod files two history entries	len(entries) == 2	Pass	0.004s
Empty dir produces no file	not output.exists()	Pass	0.000s
History entries have date	entry["date"] == "2026-01-02"	Pass	0.004s
History entries have tenors	"1" in entry and "2" in entry and "3" in entry	Pass	0.002s
History keyed by gauge	"GAUGE-042" in data["history"]	Pass	0.003s
History keyed by trigger	"alert" in gauge_hist; "warning" in gauge_hist	Pass	0.002s
Metadata has generated at	"generated_at" in data["metadata"]	Pass	0.002s
Metadata num snapshots	data["metadata"]["num_snapshots"] == 3	Pass	0.003s
Multiple gauges in metadata	set(data["metadata"]["gauges"]) == {"GAUGE-001", "GAUGE-0...	Pass	0.003s
Multiple snapshots produce multiple entries	len(alert_entries) == 3	Pass	0.005s
Output has metadata and history	"metadata" in data; "history" in data	Pass	0.003s
Single snapshot creates file	output.exists()	Pass	0.002s
Standalone check: _business_days always returns Mon-Fri dates.	len(result) == 63; d.weekday() < 5	Pass	0.000s
Pre-existing EOD-*.json files are deleted before simulation.	not stale.exists()	Pass	0.002s
Creates eod dir	(trading_dir / 'eod').exists()	Pass	0.002s
Empty trades returns zero	count == 0	Pass	0.002s
A trade that fails enrich_trade is skipped gracefully (except branch).	count == 0	Pass	0.015s
More trades should generally yield more (or equal) snapshots.	count > 0	Pass	0.153s
Returns integer	isinstance(result, int)	Pass	0.002s
Same seed should produce same random walk (deterministic).	count1 == count2	Pass	0.004s
With a real trade + real engine, we expect ¿0 snapshots.	count > 0	Pass	0.110s
EOD snapshot JSON files are created in eod/ directory.	len(eod_files) == count	Pass	0.124s

Test Description	Acceptance Criteria	Result	Time
hazard_curve_history.json and trade_pnl_history.json are created.	`(trading_dir / 'hazard_curve_history.json').exists(); (trading_dir / 'trade_pnl_history.json').exists()`	Pass	0.135s
Arrears months current	`generate_integer_value("ArrearsMonths", self._fd()) == 0`	Pass	0.000s
Arrears months in arrears	`1 <= result <= 6`	Pass	0.000s
Behavioral score	`30 <= result <= 100`	Pass	0.000s
Borrower age btl	`35 <= result <= 65`	Pass	0.000s
Borrower age expensive	`35 <= result <= 65`	Pass	0.000s
Borrower age normal	`25 <= result <= 60`	Pass	0.000s
Credit score expensive property	`700 <= result <= 800`	Pass	0.000s
Credit score low rate	`750 <= result <= 850`	Pass	0.000s
Credit score normal	`650 <= result <= 750`	Pass	0.000s
Days in arrears current	`generate_integer_value("DaysInArrears", self._fd()) == 0`	Pass	0.000s
Days in arrears defaulted	`90 <= result <= 365`	Pass	0.000s
Days in arrears in arrears	`1 <= result <= 89`	Pass	0.000s
Family members	`result in [1, 2, 3, 4]`	Pass	0.000s
Loan term	`generate_integer_value("LoanTerm", self._fd()) == 300`	Pass	0.000s
Number of borrowers	`result in [1, 2]`	Pass	0.000s
Original term	`generate_integer_value("OriginalTerm", self._fd()) == 300`	Pass	0.000s
Prepayment risk high rate	`7 <= result <= 10`	Pass	0.000s
Prepayment risk low rate	`1 <= result <= 5`	Pass	0.000s
Remaining term	`result == 240`	Pass	0.000s
Total payments	`result == 60`	Pass	0.000s
Unknown field	`1 <= result <= 10; isinstance(result, str)`	Pass	0.000s
Each location has required keys	`"lat" in loc; "lon" in loc` (+4 more)	Pass	0.000s
Elevation is non negative	`loc["elevation"] >= 0`	Pass	0.000s
With ¡ 3 gauge points, fallback generator is called.	`len(locs) == 3`	Pass	0.000s
Fallback used when no gauge points	`len(locs) == 3`	Pass	0.000s
When primary_idx == 0, triangle is gauges 0, 1, 2 -¿ sorted [0,1,2].	`set(ref) == {0, 1, 2}`	Pass	0.000s
Reference gauge indices sorted	`idxs == sorted(idxs)`	Pass	0.001s
Reference gauge indices three elements	`len(loc["reference_gauge_indices"]) == 3`	Pass	0.000s
Returns correct count	`len(locs) == 10`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
When params has no get_elevation, elevation is sampled from	`len(locs) == 5; 0 <= loc["elevation"] <= 40`	Pass	0.000s
Application channel	`result in ["Retail", "Broker", "Correspondent"]`	Pass	0.000s
Loan purpose btl	`result in ["Purchase", "Refinancing"]`	Pass	0.000s
Loan purpose residential	`result in ["Purchase", "Refinancing", "Home Improvement"]`	Pass	0.000s
Mortgage type	`result == "Buy-to-Let"`	Pass	0.000s
Occupancy type	`result == "Investment"`	Pass	0.000s
Original rate type	`isinstance(result, str) and len(result) > 0`	Pass	0.000s
Payment frequency	`generate_menu_value("PaymentFrequency", {}, 0, self._fd()...`	Pass	0.000s
Repayment type btl	`result in ["Interest only", "Repayment", "Part and part"]`	Pass	0.000s
Repayment type residential	`result in ["Repayment", "Interest only", "Part and part"]`	Pass	0.000s
Unknown field no options	`result == ""`	Pass	0.000s
Unknown field with options	`result in ["X", "Y"]`	Pass	0.000s
Convenience function creates pdf	`output.exists()`	Pass	0.010s
Convenience function returns path	`isinstance(result, Path)`	Pass	0.011s
Lines 203-208: exception -¿ 500.	`r.status_code in (200, 500)`	Pass	0.001s
Lines 174-193: full success path with mocked mortgage report generator.	`r.status_code in (200, 500)`	Pass	0.002s
Missing property id returns 400	`r.status_code == 400; "Property ID" in r.get_json()["message"]`	Pass	0.000s
Mortgage not found returns 404	`r.status_code == 404; "mortgage" in r.get_json()["message"].lower()`	Pass	0.001s
Line 149: get_json() returns None -¿ 400.	`r.status_code in (400, 415, 500)`	Pass	0.000s
No json returns error	`r.status_code in (400, 415)`	Pass	0.000s
Options returns ok	`r.status_code == 200; r.status_code == 200`	Pass	0.000s
Property not found returns 404	`r.status_code == 404`	Pass	0.001s
Mortgage import error returns 500	`r.status_code == 500; data["status"] == "error"`	Pass	0.002s
Raises without property file	—	Pass	0.001s
Returns dict when property file exists	`isinstance(result, dict); "data" in result` (+1 more)	Pass	0.021s

Test Description	Acceptance Criteria	Result	Time
Lines 332-334: open_pdf_file raises -¿ warning logged, no raise.	`result is not None`	Pass	0.001s
Lines 335-336: auto_open=False -¿ skips opening.	—	Pass	0.001s
Lines 327-334: auto_open=True -¿ calls open_pdf_file.	`mock_open.called`	Pass	0.001s
Lines 128-133: exception in report generation -¿ 500.	`r.status_code in (500, 404, 400)`	Pass	0.002s
Lines 94-119: full success path with mocked report generator.	`r.status_code in (200, 500)`; `data["status"] == "success"` (+1 more)	Pass	0.002s
Missing property id returns 400	`r.status_code == 400`; `"Property ID" in r.get_json()["message"]`	Pass	0.000s
Lines 319-320: 'mortgage-focused' with mortgage_data.	—	Pass	0.001s
Line 74: get_json() returns None -¿ 400.	`r.status_code in (400, 415, 500)`	Pass	0.000s
No json returns error	`r.status_code in (400, 415)`	Pass	0.000s
Options returns ok	`r.status_code == 200`; `r.get_json()["status"] == "ok"`	Pass	0.000s
Property not found returns 404	`r.status_code == 404`	Pass	0.001s
Line 317-318: 'property-only' -¿ generate_property_only_report.	—	Pass	0.001s
Lines 321-322: 'risk-focused'.	—	Pass	0.001s
Lines 323-324: unknown type -¿ generate_report.	—	Pass	0.001s
Auto open exception does not raise	`result is not None`	Pass	0.001s
Auto open false skips open	—	Pass	0.001s
Auto open true calls open	—	Pass	0.001s
Full type falls back to generate report	—	Pass	0.000s
Mortgage focused type with mortgage	—	Pass	0.001s
Mortgage focused without mortgage falls back	—	Pass	0.001s
Property only type	—	Pass	0.001s
Returns path	`result == fake_pdf`	Pass	0.001s
Risk focused type	—	Pass	0.001s
Unknown type falls back to generate report	—	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Line 167: log milestone every 50 properties.	`any("50" in m for m in log_messages)`	Pass	0.179s
Line 171: ratio branch when total_storms_at_gauge ¿ 0.	`"gauge_to_property_ratio" in result; result["gauge_to_property_ratio"] >= 0.0`	Pass	0.343s
Line 176: ratio=0 branch when total_storms_at_gauge == 0.	`result["gauge_to_property_ratio"] == 0.0`	Pass	0.170s
All items are strings	`all(isinstance(r, str) for r in recs)`	Pass	0.000s
Deep flood extra items	`len(deep) > len(shallow)`	Pass	0.000s
High risk four or more	`len(recs) >= 4`	Pass	0.000s
High risk mentions flood insurance	`any("flood insurance" in r.lower() for r in recs)`	Pass	0.000s
Low risk	`len(recs) >= 2`	Pass	0.000s
Medium risk	`len(recs) >= 3`	Pass	0.000s
Very low risk	`len(recs) >= 1`	Pass	0.000s
All unknown pages still creates pdf	`path.exists()`	Pass	0.002s
All unknown pages still creates pdf	`path.exists()`	Pass	0.002s
Auto select called when pages none	—	Pass	0.011s
Auto select called when pages none	—	Pass	0.015s
Auto select not called when pages provided	—	Pass	0.003s
Auto select not called when pages provided	—	Pass	0.003s
Auto select pages when none	—	Pass	0.013s
Auto select skipped when provided	—	Pass	0.003s
Custom output filename	`path.name == "my_gauge.pdf"; path.exists()`	Pass	0.011s
Custom output filename	`path.name == "my_prop.pdf"; path.exists()`	Pass	0.018s
Custom output filename	`path.name == "my_report.pdf"; path.exists()`	Pass	0.012s
Custom pages to include	`path.exists()`	Pass	0.005s
Custom pages to include	`path.exists()`	Pass	0.005s
Custom pages to include	`path.exists()`	Pass	0.007s
Empty flood data	`path.exists()`	Pass	0.012s
Empty gauge data	`path.exists()`	Pass	0.009s
Empty property data	`path.exists()`	Pass	0.014s
Full data report	`path.exists(); path.stat().st_size > 0`	Pass	0.021s
Full data report	`path.exists(); path.stat().st_size > 0`	Pass	0.018s
Pdf exists	`gen.generate_report(gauge_data).exists()`	Pass	0.011s

Test Description	Acceptance Criteria	Result	Time
Pdf exists	`gen.generate_report(prop_data).exists()`	Pass	0.015s
Pdf exists	`path.exists()`	Pass	0.012s
Pdf is bytes	`isinstance(content, bytes); len(content) > 0`	Pass	0.008s
Pdf non empty	`path.stat().st_size > 0`	Pass	0.011s
Pdf non empty	`path.stat().st_size > 0`	Pass	0.015s
Pdf non empty	`path.stat().st_size > 0`	Pass	0.012s
Pdf suffix	`gen.generate_report(gauge_data).suffix == ".pdf"`	Pass	0.011s
Pdf suffix	`gen.generate_report(prop_data).suffix == ".pdf"`	Pass	0.015s
Pdf suffix	`path.suffix == ".pdf"`	Pass	0.012s
Returns path	`isinstance(result, Path)`	Pass	0.011s
Returns path	`isinstance(result, Path)`	Pass	0.015s
Returns path	`isinstance(result, Path)`	Pass	0.012s
With mortgage data	`path.exists()`	Pass	0.131s
With timeseries data	`path.exists()`	Pass	0.023s
gauge_integrator should handle flood_gauges key too.	`result is not None`	Pass	0.001s
Found gauge returns path	`result is not None; isinstance(result, Path)`	Pass	0.002s
Gauge not found returns none	`result is None`	Pass	0.002s
Missing file returns none	`result is None`	Pass	0.000s
Output contains gauge id	`"GAUGE-001" in content`	Pass	0.002s
Output dir created	`output_dir.exists()`	Pass	0.001s
Output file created	`result.exists()`	Pass	0.001s
Corrupt json returns none	`result is None`	Pass	0.008s
Non json file returns none	`result is None`	Pass	0.001s
Found property returns path	`result is not None; isinstance(result, Path)`	Pass	0.002s
Missing property file returns none	`result is None`	Pass	0.001s
No mortgage file still generates	`result is not None`	Pass	0.001s
Output contains property id	`"PROP-001" in content`	Pass	0.003s
Output dir created	`output_dir.exists()`	Pass	0.001s
Output file created	`result.exists()`	Pass	0.002s
Property not found returns none	`result is None`	Pass	0.001s
A file with invalid JSON triggers an exception -¿ return None.	`result is None`	Pass	0.001s
Simulated write permission error returns None.	`result is None`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Import error returns 500	`r.status_code == 500; data["status"] == "error"`	Pass	0.002s
Has report generated timestamp	`"Report Generated" in r`	Pass	0.000s
Last updated extracted	`r["Data Last Updated"] == "2025-01-01"`	Pass	0.000s
Mortgage id extracted	`r.get("Mortgage ID") == "MORT-001"`	Pass	0.000s
Property id extracted	`r["Property ID"] == "PROP-999"`	Pass	0.000s
Returns dict	`isinstance(r, dict); isinstance(r, dict)`	Pass	0.000s
Scope combined with mortgage	`"Mortgage" in r["Analysis Scope"]`	Pass	0.000s
Scope property only	`"Property-Only" in r["Analysis Scope"]`	Pass	0.000s
Unknown property id when missing	`r["Property ID"] == "Unknown"`	Pass	0.000s
Uprn extracted	`r["UPRN"] == "123456789"`	Pass	0.000s
Lines 263-264: 'analysis' type.	`path.exists()`	Pass	0.016s
Line 257-258: 'basic' type.	`path.exists()`	Pass	0.017s
Lines 259-260: 'detailed' type.	`path.exists()`	Pass	0.022s
Lines 261-262: 'summary' type.	`path.exists()`	Pass	0.009s
Lines 265-266: unknown type → generate_report.	`path.exists()`	Pass	0.022s
Analysis type returns path	`path.exists()`	Pass	0.010s
Basic type returns path	`isinstance(path, Path); path.exists()`	Pass	0.010s
Omitting report_type should default to 'basic'.	—	Pass	0.011s
Detailed type returns path	`path.exists()`	Pass	0.012s
Output dir as string	`path.exists()`	Pass	0.010s
Pdf is non empty bytes on disk	`isinstance(content, bytes); len(content) > 0`	Pass	0.016s
Summary type returns path	`path.exists()`	Pass	0.008s
Unknown type falls back to generate report	`path.exists()`	Pass	0.012s
Empty gaugets directory must raise FileNotFoundError.	—	Pass	0.001s
Each index entry must have gauge_ids_alert for directory-based filtering.	`"gauge_ids_alert" in entry; "GAUGE-00000001" in entry["gauge_ids_alert"]`	Pass	0.002s
Index has required top level keys	`key in data`	Pass	0.003s
Individual storm files must contain the full gauge_responses array.	`"gauge_responses" in storm; len(storm["gauge_responses"]) == 1`	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Storm id prefix uses config constant	`data["storm_id_prefix"] == STORM_ID_PREFIX`	Pass	0.004s
Storms where no gauge exceeded alert must NOT appear in output.	`result["total_storms"] == 0`	Pass	0.001s
Total storms matches list length	`data["total_storms"] == len(data["storms"])`	Pass	0.009s
Output directory must contain _index.json and per-storm files.	`(out_dir / "_index.json").exists(); (out_dir / f"{sid}.json").exists()`	Pass	0.003s
Account status defaulted	`result == "Default"`	Pass	0.000s
Account status normal	`result in ["Current", "Arrears", "Closed"]`	Pass	0.000s
Borrower employment btl	`isinstance(result, str)`	Pass	0.000s
Catchment id	`generate_text_value("CatchmentID", 0, self._fd()) == "tha...`	Pass	0.000s
Currency	`generate_text_value("currency", 0, self._fd()) == "GBP"`	Pass	0.000s
Employment type residential	`isinstance(result, str)`	Pass	0.000s
Flood risk category	`result == "Medium"`	Pass	0.000s
Lastatus defaulted	`result == "Defaulted"`	Pass	0.000s
Lastatus normal	`result in ["Current", "Completed", "Redeemed"]`	Pass	0.000s
Marital status older	`isinstance(result, str)`	Pass	0.000s
Marital status young	`result in ["Single", "Married", "Civil Partnership"]`	Pass	0.000s
Member id	`result.startswith("MEMBER-")`	Pass	0.000s
Mortgage id	`generate_text_value("MortgageID", 0, self._fd()) == "MORT...`	Pass	0.000s
Mortgage provider	`isinstance(result, str)`	Pass	0.000s
Property id	`generate_text_value("PropertyID", 0, self._fd()) == "PROP...`	Pass	0.000s
Unknown field	`"Text-5-" in result`	Pass	0.000s
Uprn no existing	`result.startswith("UPRN-") or isinstance(result, str)`	Pass	0.000s
Uprn with existing	`result == "UPRN-12345"`	Pass	0.000s
Alias returns same structure	`len(result) == 50; "readings" in result[0]`	Pass	0.000s
CloseOutDate with Payable SettlementDirection.	`pdf_path.exists()`	Pass	0.003s
_generate_trade_pdf creates a PDF without cashflows.	`pdf_path.exists(); pdf_path.stat().st_size > 1000`	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
_generate_trade_pdf with cashflows exercises CF schedule section.	`pdf_path.exists()`	Pass	0.002s
_generate_trade_pdf with CloseOutOf exercises close-out section.	`pdf_path.exists()`	Pass	0.003s
_generate_trade_pdf with CloseOutDate (original trade that was closed).	`pdf_path.exists()`	Pass	0.003s
Pdf failure returns empty list	`pdfs == []`	Pass	0.000s
Pdf partial failure returns successful only	`len(pdfs) == 1`	Pass	0.000s
Creates output file	`output.exists()`	Pass	0.002s
History entry has required keys	`'date' in entry`; `'mark' in entry` (+2 more)	Pass	0.002s
Multiple eods accumulate	`len(trade['history']) == 3`	Pass	0.004s
Multiple trades in same eod	`'PRS-A' in data['trades']`; `'PRS-B' in data['trades']` (+1 more)	Pass	0.003s
No eod files returns early	`not output.exists()`	Pass	0.000s
Num snapshots in metadata	`data['metadata']['num_snapshots'] == 2`	Pass	0.003s
Output has metadata	`'metadata' in data`; `'num_trades' in data['metadata']`	Pass	0.003s
Output has trades	`'trades' in data`; `'PRS-001' in data['trades']`	Pass	0.006s
Position without swap id skipped	`data['trades'] == {}`; `data['metadata']['num_trades'] == 0`	Pass	0.002s
Trade has history	`'history' in trade`; `len(trade['history']) == 1`	Pass	0.002s
Trade metadata preserved	`trade['gauge_id'] == 'GAUGE-001'`; `trade['trigger'] == 'severe'` (+3 more)	Pass	0.002s
Empty dir produces no file	`not output.exists()`	Pass	0.000s
History entry structure	`entry["date"] == "2026-01-02"`; `entry["mark"] == 215.0` (+2 more)	Pass	0.003s
Metadata has generated at	`"generated_at" in data["metadata"]`	Pass	0.003s
Metadata num snapshots	`data["metadata"]["num_snapshots"] == 3`	Pass	0.004s
Multiple days builds history vector	`len(data["trades"]["PRS-001"]["history"]) == 3`	Pass	0.005s
Multiple trades in snapshot	`data["metadata"]["num_trades"] == 3`	Pass	0.003s
Output has metadata and trades	`"metadata" in data`; `"trades" in data`	Pass	0.003s

Test Description	Acceptance Criteria	Result	Time
Position missing swap id is skipped	`"PRS-001" in data["trades"]; len(data["trades"]) == 1`	Pass	0.003s
Single snapshot creates file	`output.exists()`	Pass	0.002s
Trade keyed by swap id	`"PRS-001" in data["trades"]`	Pass	0.003s
Trade metadata fields	`"gauge_id" in t; "trigger" in t` (+3 more)	Pass	0.003s
Explicit years overrides instance default	`result["years_included"] == 1`	Pass	0.013s
Instance years used when no years arg	`result["years_included"] == 2`	Pass	0.015s
If all sequence storms are sub-alert, portfolio flood count is 0.	`result["total_storms_at_gauge"] == 0; result["properties_with_floods"] == 0`	Pass	0.169s
Portfolio summary total_storms_at_gauge counts all alert storms.	`result["total_storms_at_gauge"] == 2`	Pass	0.171s
portfolio_flood_summary.json is written with per-property data.	`summary_path.exists(); summary["summary"]["total_gauges"] == 1` (+1 more)	Pass	0.171s
Generate output path none uses config input dir	`isinstance(result, dict); (tmp_path / "gaugehd_39001.json").exists()`	Pass	0.015s
Banks always have CSAAgreement = True.	`platform["CSAAgreement"] is True`	Pass	0.001s
Counterparties list has two parties	`len(counterparties) == 2`	Pass	0.000s
Counterparty has counterparty set	`"CounterpartySet" in ctpy`	Pass	0.000s
Isda master agreement is true	`platform["ISDAMasterAgreement"] is True`	Pass	0.000s
Party has accounts	`"Accounts" in party; len(party["Accounts"]) >= 1`	Pass	0.000s
Party has contact info	`"ContactInformation" in party; "PrimaryEmail" in party["ContactInformation"]`	Pass	0.000s
Party has id	`"PartyID" in party; party["PartyID"].startswith("CTPY-")`	Pass	0.000s
Party has name	`"PartyName" in party; isinstance(party["PartyName"], str)`	Pass	0.000s
Party has natural person	`"NaturalPersons" in party; len(party["NaturalPersons"]) == 1`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Platform has credit rating	`"CreditRating" in platform; platform["CreditRating"] in [ "AAA", "AA+...`	Pass	0.000s
Unique party ids	`len(set(ids)) == 5`	Pass	0.001s
Generator exception falls through to type	`isinstance(result, float); 0 <= result <= 1000`	Pass	0.000s
Gauge importable without quantlib	`hasattr(mod, "GaugePortfolioGenerator")`	Pass	0.005s
models.hazard.builder must not drag in QuantLib pricing at import.	`mod is not None`	Pass	0.003s
Mortgage importable without quantlib	`hasattr(mod, "MortgagePortfolioGenerator")`	Pass	0.002s
Importing port.src (package __init__) must not require QuantLib.	`hasattr(mod, "GaugePortfolioGenerator")`	Pass	0.002s
Storm multi importable without quantlib	`mod is not None`	Pass	0.009s
Depth damage importable without geopandas	`hasattr(mod, 'scalar_depth_damage')`	Pass	0.001s
__init__.py must not have an executable FloodRiskModel import statement.	`'FloodRiskModel' not in executable`	Pass	0.000s
propertyts imports scalar_depth_damage – must work without geopandas.	`hasattr(mod, 'PropertyTimeSeriesGenerator')`	Pass	0.001s
Scalar depth damage works without geopandas	`0.0 < result <= 1.0`	Pass	0.001s
Spatial importable without geopandas	`hasattr(mod, 'haversine_distance')`	Pass	0.001s
100 years is heritage	`get_age_band(1925, current_year=2025) == 'heritage'`	Pass	0.000s
10 years is modern	`get_age_band(2015, current_year=2025) == 'modern'`	Pass	0.000s
25 years is established	`get_age_band(2000, current_year=2025) == 'established'`	Pass	0.000s
50 years is period	`get_age_band(1975, current_year=2025) == 'period'`	Pass	0.000s
Default current year	`result in AGE_BAND_FACTORS`	Pass	0.000s
Established	`get_age_band(1985, current_year=2025) == 'established'`	Pass	0.000s
Exactly 24 years is modern	`get_age_band(2001, current_year=2025) == 'modern'`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Exactly 49 years is established	`get_age_band(1976, current_year=2025) == 'established'`	Pass	0.000s
Exactly 99 years is period	`get_age_band(1926, current_year=2025) == 'period'`	Pass	0.000s
Exactly 9 years is new build	`get_age_band(2016, current_year=2025) == 'new_build'`	Pass	0.000s
Heritage old	`get_age_band(1800, current_year=2025) == 'heritage'`	Pass	0.000s
Modern	`get_age_band(2005, current_year=2025) == 'modern'`	Pass	0.000s
New build	`get_age_band(2020, current_year=2025) == 'new_build'`	Pass	0.000s
Period	`get_age_band(1950, current_year=2025) == 'period'`	Pass	0.000s
After log entry appears	`data["total_entries"] == 1; len(data["entries"]) == 1`	Pass	0.004s
Empty trail returns 200	`r.status_code == 200; data["status"] == "success"` (+1 more)	Pass	0.001s
Filter by event type	`all(e["event_type"] == "calibration" for e in entries)`	Pass	0.005s
Filter by model id	`r.get_json()["total_entries"] == 1`	Pass	0.004s
Filter by model id no match	`r.get_json()["total_entries"] == 0`	Pass	0.004s
Has returned field	`"returned" in r.get_json()`	Pass	0.001s
Limit parameter	`r.get_json()["returned"] == 2`	Pass	0.008s
Missing file returns empty list	`result == []`	Pass	0.000s
File with two gauges should return both IDs.	`set(result) == {"GAUGE-A", "GAUGE-B"}`	Pass	0.001s
Returns correct id	`"GAUGE-001" in result`	Pass	0.001s
Returns list	`isinstance(result, list)`	Pass	0.001s
Binary file returns empty	`result == []`	Pass	0.001s
Corrupt json returns empty	`result == []`	Pass	0.001s
No data returns 404	`r.status_code == 404; r.get_json()["status"] == "error"`	Pass	0.000s
Principles count	`len(r.get_json()["assessment"]["principles"]) == 2`	Pass	0.002s
Response has assessment	`"assessment" in data; "principles" in data["assessment"]`	Pass	0.002s
With data returns 200	`r.status_code == 200; r.get_json()["status"] == "success"`	Pass	0.002s
Returns 200 if PDF exists, 404 if not yet generated.	`r.status_code in (200, 404)`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Empty references	`r.get_json()["references"] == []`	Pass	0.002s
Has categories key	`"categories" in r.get_json()`	Pass	0.002s
Returns 200	`r.status_code == 200`	Pass	0.001s
Returns success status	`r.get_json()["status"] == "success"`	Pass	0.002s
Returns empty trades list when no PRS files exist.	`resp.status_code == 200; data['status'] == 'success'` (+1 more)	Pass	0.003s
Payer (long protection) trade should have positive gauge_fs01.	`len(payer_trades) > 0; t['gauge_fs01'] >= 0`	Pass	0.025s
Receiver (short protection) trade should have negative gauge_fs01.	`len(rcv_trades) > 0; t['gauge_fs01'] <= 0`	Pass	0.024s
Trades should have gauge_name from gauge locations.	`'Thames at Westminster' in names`	Pass	0.022s
Each trade must have P&L decomposition fields.	`'daily_pnl' in trade; 'market_pnl' in trade` (+2 more)	Pass	0.026s
Each trade must have prev_fair_spread_bps for curve-move indication.	`'prev_fair_spread_bps' in trade`	Pass	0.024s
Each trade must have core identification and pricing fields.	`'swap_id' in trade; 'gauge_id' in trade` (+5 more)	Pass	0.026s
Summary must have all required aggregate fields.	`'num_trades' in summary; 'total_notional' in summary` (+4 more)	Pass	0.026s
Blotter returns enriched trades with status success.	`resp.status_code == 200; data['status'] == 'success'` (+1 more)	Pass	0.021s
Found returns 200	`r.status_code == 200; data["status"] == "success"` (+1 more)	Pass	0.002s
Found second counterparty	`r.status_code == 200`	Pass	0.004s
Not found returns 404	`r.status_code == 404; data["status"] == "error"`	Pass	0.002s
Format	`rid.startswith("run-"); len(rid) == 19`	Pass	0.000s
All pipeline outputs exist and are fresh.	`data["status"] == "success"; data["summary"]["fresh"] == 14` (+4 more)	Pass	0.008s
All present but some stale = degraded.	`data["summary"]["missing"] == 0; data["summary"]["stale"] == 1` (+1 more)	Pass	0.004s
Response includes as_of timestamp.	`"as_of" in data`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Mix of fresh, stale, and missing.	`data["summary"]["fresh"] == 1`; `data["summary"]["stale"] == 1` (+2 more)	Pass	0.002s
When lineage file does not exist, manifest is None but route succeeds.	`r.status_code == 200`; `data["status"] == "success"` (+3 more)	Pass	0.001s
Empty list returns success	`r.status_code == 200`; `data["status"] == "success"` (+1 more)	Pass	0.001s
Has fields	`"owner" in fields`; `"rag_rating" in fields`	Pass	0.000s
Returns 200	`r.status_code == 200`; `r.status_code == 200`	Pass	0.000s
Returns success status	`r.get_json()["status"] == "success"`; `r.get_json()["status"] == "success"`	Pass	0.000s
Empty entity returns none	`storm_loader.get_entity_id({}) is None`	Pass	0.000s
Extracts event id	`storm_loader.get_entity_id(storms[0]) == "EVT-001"`	Pass	0.003s
Missing id returns none	`storm_loader.get_entity_id(entity) is None`	Pass	0.000s
Missing fields return none	`summary["eventId"] is None`; `summary["name"] is None`	Pass	0.000s
Returns summary dict	`summary["eventId"] == "EVT-001"`; `summary["name"] == "Harvey"` (+3 more)	Pass	0.001s
get_eod_snapshot returns snapshot data for existing date.	`result is not None`; `result['date'] == '2026-03-15'`	Pass	0.002s
get_eod_snapshot returns None when no snapshot exists for the date.	`result is None`	Pass	0.000s
Filter by report	`r.status_code == 200`; `data["total_reports"] == 1` (+3 more)	Pass	0.002s
Filter unknown report 404	`r.status_code == 404`; `"available_reports" in data`	Pass	0.002s
Full registry	`r.status_code == 200`; `data["status"] == "success"` (+4 more)	Pass	0.014s
No field lineage file returns 404.	`r.status_code == 404`; `"not found" in data["message"].lower()` (+1 more)	Pass	0.001s
Summary contents	`fr_summary["label"] == "Flood Risk Report"`; `fr_summary["section_count"] == 2` (+1 more)	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Producer step exists in STEP_IO but not in manifest.	`result["produced_by"] == "synthetic_gauges"; result["producer_run_id"] is None`	Pass	0.000s
Known file	`result["produced_by"] == "synthetic_gauges"; result["producer_run_id"] == "r2"` (+1 more)	Pass	0.000s
Unknown file	`result["produced_by"] is None; result["producer_run_id"] is None` (+1 more)	Pass	0.000s
Gauge not found returns 404	`r.status_code == 404; r.get_json()["status"] == "error"`	Pass	0.002s
No curve for gauge returns 404	`r.status_code == 404`	Pass	0.003s
No hazard file returns 404	`r.status_code == 404; "not available" in r.get_json()["message"].lower()` (+1 more)	Pass	0.002s
Response has annual flood probs	`"alert" in probs; "warning" in probs` (+1 more)	Pass	0.003s
Response has curve points	`len(r.get_json()["curve_points"]) == 3`	Pass	0.003s
Response has flood stages	`"FloodAlert" in stages; "FloodWarning" in stages` (+1 more)	Pass	0.003s
Response has gauge id	`r.get_json()["gauge_id"] == GAUGE_ID`	Pass	0.002s
Response has gev parameters	`"location" in gev; "scale" in gev` (+1 more)	Pass	0.003s
Response has term structures	`"alert" in ts; "warning" in ts` (+1 more)	Pass	0.003s
Valid hazard returns 200	`r.status_code == 200; r.get_json()["status"] == "success"`	Pass	0.003s
Days larger than available returns all	`len(data["daily_observations"]) == 12`	Pass	0.002s
Days param limits observations	`len(data["daily_observations"]) == 5`	Pass	0.002s
Days param zero returns all	`len(data["daily_observations"]) == 12`	Pass	0.002s
No history file returns 404	`r.status_code == 404; r.get_json()["status"] == "error"`	Pass	0.001s
Response has daily observations	`"daily_observations" in data; len(data["daily_observations"]) == 12`	Pass	0.002s
Response has flood stages	`"flood_stages" in data`	Pass	0.002s
Response has gauge metadata	`"gauge_metadata" in data`	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Response has statistics	"statistics" in data; data["statistics"]["mean_level"] == 2.5	Pass	0.002s
Valid history returns 200	r.status_code == 200; r.get_json()["status"] == "success"	Pass	0.002s
Ignores other keys	gen._get_gauge_severe_count(gauge_hc) == 3	Pass	0.000s
Large count returned	gen._get_gauge_severe_count(gauge_hc) == 150	Pass	0.000s
Missing key defaults to zero	gen._get_gauge_severe_count(gauge_hc) == 0	Pass	0.000s
Returns severe event count	gen._get_gauge_severe_count(gauge_hc) == 7	Pass	0.000s
Zero count returns zero	gen._get_gauge_severe_count(gauge_hc) == 0	Pass	0.000s
Gauge not found returns 404	r.status_code == 404; r.get_json()["status"] == "error" (+2 more)	Pass	0.001s
Valid gauge returns success	r.status_code == 200; data["status"] == "success" (+4 more)	Pass	0.002s
Gauge not found returns 404	r.status_code == 404; r.get_json()["status"] == "error"	Pass	0.002s
Response has flood simulation	"flood_simulation" in data; "num_timesteps" in data["flood_simulation"]	Pass	0.004s
Response has flood stages	"flood_stages" in r.get_json()	Pass	0.004s
Response has gauge id	r.get_json()["gauge_id"] == GAUGE_ID	Pass	0.003s
Response has storm responses	"storm_responses" in data; "num_storms" in data["storm_responses"]	Pass	0.004s
Storm responses enriched with metadata from storm_sequences.json.	"name" in responses[0] or responses[0].get("name") == ""	Pass	0.004s
Valid gauge returns 200	r.status_code == 200; r.get_json()["status"] == "success"	Pass	0.004s
When exceeded_severe=True, severe count computed from gaugets dir.	r.status_code == 200	Pass	0.003s
Gauge not found returns 404	r.status_code == 404; r.get_json()["status"] == "error" (+2 more)	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Response has count and readings	`"count" in data`; `"readings" in data` (+1 more)	Pass	0.002s
Valid gauge returns success	`r.status_code == 200`; `data["status"] == "success"` (+4 more)	Pass	0.003s
When term structure doesn't have the requested tenor, falls back	`rate == base_rate`	Pass	0.002s
Unknown gauge with no term structure returns 0.0 from flat rate fallback.	`rate == 0.0`	Pass	0.001s
Missing id returns none	`storm_loader.get_track("EVT-NONEXISTENT") is None`; `storm_loader.get_impact("EVT-NONEXISTENT") is None`	Pass	0.001s
Returns impact dict	`isinstance(impact, dict)`; `"TotalDamage" in impact`	Pass	0.001s
Contains timeout	`'30000' in js`	Pass	0.000s
Returns script tag	`'<script>' in js`; `'</script>' in js` (+2 more)	Pass	0.001s
All modules non empty	`len(js) > 50`	Pass	0.000s
All modules return str	`isinstance(js, str)`	Pass	0.000s
Excludes category 1 and 2	`"EVT-004" not in ids`	Pass	0.001s
Includes cat 3 4 5	`"EVT-001" in ids`; `"EVT-002" in ids` (+1 more)	Pass	0.001s
Returns category 3 plus	`all(c >= 3 for c in categories)`	Pass	0.001s
Get created meeting	`r.status_code == 200`; `data["status"] == "success"` (+1 more)	Pass	0.003s
Unknown id returns 404	`r.status_code == 404`; `r.get_json()["status"] == "error"`	Pass	0.001s
Meeting not found returns 404	`r.status_code == 404`; `r.get_json()["status"] == "error"`	Pass	0.001s
Valid meeting content type pdf	`"application/pdf" in r.content_type`	Pass	0.004s
Valid meeting returns bytes	`len(r.data) > 100`	Pass	0.004s
Valid meeting returns pdf	`r.status_code == 200`	Pass	0.004s
Model has validation questions	`"validation_questions" in model`; `len(model["validation_questions"]) == 9`	Pass	0.003s
Model not found returns 404	`r.status_code == 404`; `r.status_code == 404` (+1 more)	Pass	0.002s
Response has audit entries	`"audit_entries" in r.get_json()`	Pass	0.003s
Valid model returns success	`r.status_code == 200`; `data["status"] == "success"` (+1 more)	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
No inventory returns 404	`r.status_code == 404; r.get_json()["status"] == "error"` (+3 more)	Pass	0.001s
Category distribution	`"category_distribution" in data`	Pass	0.002s
Model has review status	`"review_status" in model; model["review_status"] in ("On Track", "Upcoming", "Due S...`	Pass	0.002s
Model has risk rating	`"risk_rating" in model`	Pass	0.002s
Model has validation coverage	`"validation_coverage" in model`	Pass	0.002s
Response has metadata	`"metadata" in data`	Pass	0.004s
Returns 200	`r.status_code == 200; r.status_code == 200`	Pass	0.002s
Returns models list	`"models" in data; data["total_models"] >= 1`	Pass	0.002s
Returns success status	`r.get_json()["status"] == "success"; r.get_json()["status"] == "success"`	Pass	0.002s
Tier distribution	`"tier_distribution" in data`	Pass	0.002s
No inventory returns 404	`r.status_code == 404; r.get_json()["status"] == "error"` (+3 more)	Pass	0.001s
Not scheduled review status	`r.status_code == 200; model["review_status"] == "Not Scheduled"`	Pass	0.002s
Overdue review status	`r.status_code == 200; model["review_status"] == "Overdue"`	Pass	0.002s
Returns expected keys	`result["NRFAStationID"] == "39001"; result["MeanFlow"] == pytest.approx(55.0)` (+2 more)	Pass	0.000s
Truncates data quality notes	`len(result["DataQualityNotes"]) <= 500`	Pass	0.000s
All expected keys present	`key in result`	Pass	0.000s
Catchment area always none	`result["CatchmentArea"] is None`	Pass	0.000s
Custom flow units	`gen.get_nrfa_metadata_for_cdm({}, {"units": "ft3/s"}, "f....`	Pass	0.000s
Database source from station metadata	`result["DatabaseSource"] == "TestDB"`	Pass	0.000s
Default flow units m3 per s	`gen.get_nrfa_metadata_for_cdm({}, {}, "f.json")["FlowUnit...`	Pass	0.000s
Empty data quality notes when no description	`gen.get_nrfa_metadata_for_cdm({}, {}, "f.json")["DataQual...`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Missing database source returns none	gen.get_nrfa_metadata_for_cdm({}, {}, "f.json")["Database...	Pass	0.000s
Missing stats keys return none	result[key] is None	Pass	0.000s
Response has 'storms' list with count ¿ 0.	'storms' in data; data['count'] > 0	Pass	0.001s
Returns 200 with status=success.	resp.status_code == 200; data['status'] == 'success'	Pass	0.002s
Each storm has storm_id, name, intensity_category, gauges_severe,	'storm_id' in s; 'name' in s (+5 more)	Pass	0.002s
Returns 404 when stress_storms not found.	resp.status_code == 404	Pass	0.000s
When gauges_severe is equal, storms sorted by gauges_warning desc.	s1['gauges_warning'] >= s2['gauges_warning']	Pass	0.002s
Storms sorted by gauges_severe desc — STORM_PORT_SEVERE must come first.	len(storms) >= 2; storms[0]['storm_id'] == STORM_PORT_SEVERE (+1 more)	Pass	0.002s
Get preloader js matches get js	_get_preloader_js() == get_js()	Pass	0.000s
Get preloader js returns string	isinstance(result, str); 'DOMContentLoaded' in result	Pass	0.000s
Found returns property	r.status_code == 200; data["status"] == "success" (+1 more)	Pass	0.001s
Not found returns 404	r.status_code == 404; r.get_json()["status"] == "error"	Pass	0.001s
Close high risk high	get_proximity_zone(50, 'High') == 'close_high_risk'	Pass	0.000s
Close high risk medium	get_proximity_zone(100, 'Medium') == 'close_high_risk'	Pass	0.000s
Close high risk very high	get_proximity_zone(10, 'Very High') == 'close_high_risk'	Pass	0.000s
Close low risk low	get_proximity_zone(150, 'Low') == 'close_low_risk'	Pass	0.000s
Close low risk very low	get_proximity_zone(100, 'Very Low') == 'close_low_risk'	Pass	0.000s
Exactly 200m is medium zone	get_proximity_zone(200, 'Low') == 'medium'	Pass	0.000s
Exactly 500m is far	get_proximity_zone(500, 'Low') == 'far'	Pass	0.000s
Far zone	get_proximity_zone(600, 'Medium') == 'far'	Pass	0.000s
Medium zone	get_proximity_zone(300, 'Very Low') == 'medium'	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Medium zone upper bound	`get_proximity_zone(499, 'Very High') == 'medium'`	Pass	0.000s
Returns none when quantlib unavailable	`result is None`	Pass	0.001s
Model not found returns 404	`r.status_code == 404`; `r.status_code == 404`	Pass	0.002s
Response has risk rating	`"risk_rating" in data`; `"calculated_rating" in rr` (+1 more)	Pass	0.002s
Risk rating has component scores	`"component_scores" in rr`	Pass	0.003s
Valid model returns success	`r.status_code == 200`; `data["status"] == "success"` (+1 more)	Pass	0.002s
Gauges affects many	`"properties" in downstream`; `"gaugehd" in downstream` (+1 more)	Pass	0.000s
Leaf has no downstream	`downstream == []`	Pass	0.000s
Mid chain	`"hazard" in downstream`; `"propertyts" in downstream`	Pass	0.000s
Custom statistics	`stats['property_menu_items'] == 2`; `stats['gauge_menu_items'] == 1` (+1 more)	Pass	0.000s
Default statistics	`stats['property_menu_items'] == len(DEFAULT_PROPERTY_MENU)`; `stats['gauge_menu_items'] == len(DEFAULT_GAUGE_MENU)` (+1 more)	Pass	0.000s
Get statistics keys	`stats['server_url'] == 'http://test:1234'`; `stats['total_endpoints'] == 2` (+1 more)	Pass	0.000s
Get statistics single endpoint	`stats['total_endpoints'] == 1`; `stats['endpoints'] == ['only']`	Pass	0.000s
Returns dict with expected keys	`'notifications' in stats`; `'backend_handler' in stats` (+1 more)	Pass	0.000s
Returns expected keys	`'default_position' in stats`; `'auto_dismiss_timeout' in stats` (+3 more)	Pass	0.000s
Statistics are dicts	`isinstance(stats['notifications'], dict)`; `isinstance(stats['backend_handler'], dict)` (+1 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Statistics reflect config	`stats['default_position'] == 'bottom-center'; stats['auto_dismiss_timeout'] == 9000` (+1 more)	Pass	0.000s
Mid step	`"gauges" in result["upstream"]; "mortgages" in result["downstream"]` (+1 more)	Pass	0.000s
Root step	`result["upstream"] == []; "properties" in result["downstream"]` (+3 more)	Pass	0.000s
Step not in manifest	`result["last_run"] is None; result["last_timestamp"] is None`	Pass	0.000s
Cluster returns 3 or 4	`count in (3, 4)`	Pass	0.000s
Doublet returns 2	`get_storm_count(SequenceType.DOUBLET, rng) == 2`	Pass	0.000s
Isolated returns 1	`get_storm_count(SequenceType.ISOLATED, rng) == 1`	Pass	0.000s
Persistent returns 4 or 5	`count in (4, 5)`	Pass	0.000s
Each storm has peak_level_m.	`'peak_level_m' in s; s['peak_level_m'] > 0`	Pass	0.003s
Storms have required identification fields.	`'storm_id' in s; 'name' in s` (+3 more)	Pass	0.008s
Returns 404 when stress_storms not found.	`resp.status_code == 404`	Pass	0.000s
Missing gauge_id returns total count (used by startup preloader).	`resp.status_code == 200; data['status'] == 'success'` (+2 more)	Pass	0.002s
Returns empty list for gauge with no storms.	`resp.status_code == 200; data['storms'] == []`	Pass	0.002s
Returns storm list for specified gauge.	`resp.status_code == 200; data['status'] == 'success'` (+2 more)	Pass	0.006s
Storms sorted by peak level descending (worst first).	`peaks == sorted(peaks, reverse=True)`	Pass	0.003s
Each storm returned at most once	`len(result) == 1`	Pass	0.003s
Empty for remote location	`result == []`	Pass	0.001s
Finds nearby storm	`"EVT-001" in ids`	Pass	0.003s
Large radius finds more	`len(large) >= len(small)`	Pass	0.001s
Each gauge has gauge_id, gauge_name, lon, lat, trade_count.	`'gauge_id' in g; 'gauge_name' in g` (+3 more)	Pass	0.019s

Test Description	Acceptance Criteria	Result	Time
Gauges with trades show non-zero trade_count.	`sum(counts) > 0`	Pass	0.022s
Returns gauge list with trade counts.	`resp.status_code == 200`; `data['status'] == 'success'` (+2 more)	Pass	0.022s
Gauges sorted west to east (ascending longitude).	`lons == sorted(lons)`	Pass	0.024s
Missing id returns none	`storm_loader.get_track("EVT-NONEXISTENT") is None`; `storm_loader.get_impact("EVT-NONEXISTENT") is None`	Pass	0.001s
Returns track for existing storm	`track is not None`; `isinstance(track, list)` (+1 more)	Pass	0.001s
Track has lat lon	`"Latitude" in track[0]`; `"Longitude" in track[0]`	Pass	0.001s
get_trade_mark returns mark data for an existing trade.	`mark['fair_spread_bps'] == 250.0`	Pass	0.001s
get_trade_mark returns empty dict for unknown trade.	`mark == {}`	Pass	0.000s
get_trade_mark returns merged data after multiple updates.	`mark['fair_spread_bps'] == 200.0`; `mark['mtm'] == 5000.0`	Pass	0.002s
If the PDF exists, it should be served with 200.	`r.status_code == 200`; `r.content_type == "application/pdf"`	Pass	0.030s
Missing pdf returns 404	`r.status_code == 404`; `r.get_json()["status"] == "error"`	Pass	0.000s
When yield_curve is empty dict, falls back to risk_free_rate.	`rate == 0.042`	Pass	0.000s
When yield_curve key is missing, falls back to risk_free_rate.	`rate == 0.05`	Pass	0.000s
When tenor falls exactly between integer keys but loop exits, returns fallback.	`0.035 <= rate <= 0.045`	Pass	0.000s
Apply documents function exists	`'_applyDocuments' in src`	Pass	0.000s
Checks gov docs cache	`'window._tdPreGovDocs' in src`	Pass	0.000s
Clears cache after use	`'window._preStorms = null' in src`; `'window._tdPreGovDocs = null' in src`	Pass	0.000s
For each zone and elevation, spread should decrease as distance increases.	`g[di][ei] >= g[di + 1][ei]`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
For each zone and distance, spread should decrease as elevation increases.	`g[di][ei] >= g[di][ei + 1]`	Pass	0.000s
Zone2 gt zone1 everywhere	`g2[di][ei] >= g1[di][ei]`	Pass	0.000s
Zone3a gt zone2 everywhere	`g3a[di][ei] >= g2[di][ei]`	Pass	0.000s
Zone3b gt zone3a everywhere	`g3b[di][ei] >= g3a[di][ei]`	Pass	0.000s
Zone1 origin	`grid['zones']['Zone 1']['grid'][0][0] == expected`	Pass	0.000s
Zone2 mid grid	`grid['zones']['Zone 2']['grid'][10][5] == expected`	Pass	0.000s
Zone3a far high	`grid['zones']['Zone 3a']['grid'][20][10] == expected`	Pass	0.000s
Zone3b origin	`grid['zones']['Zone 3b']['grid'][0][0] == expected`	Pass	0.000s
Distance count	`len(grid['distances']) == 21`	Pass	0.000s
Elevation count	`len(grid['elevations']) == 11`	Pass	0.000s
Grid dimensions per zone	`len(zone_data['grid']) == 21`; `len(row) == 11`	Pass	0.000s
Has all zones	`set(grid['zones'].keys()) == set(EA_FLOOD_ZONE_RATES.keys())`	Pass	0.000s
Has distances	`grid['distances'] == GRID_DISTANCES_M`	Pass	0.000s
Has elevations	`grid['elevations'] == GRID_ELEVATIONS_M`	Pass	0.000s
Params present	`p['elev_scale_m'] == ELEV_SCALE_M`; `p['tenor'] == GRID_TENOR` (+2 more)	Pass	0.000s
Zone rates match config	`zone_data['rate'] == EA_FLOOD_ZONE_RATES[zone_name]`	Pass	0.000s
Hash directory	`result["type"] == "directory"`; `result["hash"] is not None` (+1 more)	Pass	0.001s
Hash file	`result["type"] == "file"`; `result["hash"] is not None`	Pass	0.001s
Hash missing	`result == {"hash": None, "type": "missing"}`	Pass	0.000s
Hash empty directory	`count == 0`; `len(digest) == 64`	Pass	0.000s
Hash multiple files deterministic	`d1 == d2`; `c1 == c2 == 2`	Pass	0.001s
Hash respects pattern	`count_json == 1`; `count_csv == 1`	Pass	0.001s
Hash single file	`count == 1`; `len(digest) == 64`	Pass	0.001s
Hash binary file	`result == expected`	Pass	0.007s
Hash empty file	`result == expected`	Pass	0.000s
Hash known content	`result == expected`	Pass	0.002s
Approximate one degree lat	`110_000 < d < 112_000`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Approximate one degree lon at equator	`110_000 < d < 112_000`	Pass	0.000s
Equatorial degree approx 111km	`110 < d < 113`	Pass	0.000s
London to paris approx 340km	`320 < d < 360`	Pass	0.000s
Same point is zero	`model.haversine_km(0.0, 0.0, 0.0, 0.0) == pytest.approx(0...`	Pass	0.000s
Same point returns zero	`d == pytest.approx(0.0, abs=1e-6)`	Pass	0.000s
Symmetric	`d1 == pytest.approx(d2, rel=1e-6)`	Pass	0.000s
Symmetry	`d1 == pytest.approx(d2, rel=1e-9)`	Pass	0.000s
GET /api/v1/trading/hazard-term-structure must return HTS data.	`resp.status_code == 200`; `data['status'] == 'success'` (+1 more)	Pass	0.052s
API response must match the market_state.json file.	`set(api_hts.keys()) == set(file_hts.keys())`	Pass	0.055s
gaugehc.json must exist and contain hazard curves.	`path.exists()`; `len(curves) > 0`	Pass	0.014s
Each gauge curve must have annual_hazard_rate for all triggers.	`key in gc`; `gc[key] > 0`	Pass	0.010s
market_state.json must exist with hazard_term_structure.	`path.exists()`; `len(hts) > 0`	Pass	0.004s
Every real gauge in gaugehc should have a market state entry.	`not missing`	Pass	0.014s
Each gauge's HTS must have rates for tenors 1-5.	`tenor in rates`; `rates[tenor] > 0`	Pass	0.004s
num_sequences used by property pricer must match storm_sequences.json.	`expected != num_storms or expected == num_storms`; `expected > 0` (+1 more)	Pass	0.208s
hazard.py lines 100-102: exception reading hazard file -¿ 500.	`r.status_code == 500`; `r.get_json()["status"] == "error"`	Pass	0.029s
Custom day count	`curve is not None`	Pass	0.000s
High hazard gives low survival	`sp < 0.1`	Pass	0.000s
Returns credit curve	`curve is not None`	Pass	0.002s
Survival prob decreases	`sp2 < sp1`	Pass	0.000s
Zero hazard gives survival near one	`sp >= 0.999`	Pass	0.000s
Commit returns affected_trades for the target gauge.	`resp.status_code == 200`; `data['status'] == 'success'` (+3 more)	Pass	0.028s
Missing gauge_id returns 400.	`resp.status_code == 400`	Pass	0.000s
Missing rates returns 400.	`resp.status_code == 400`	Pass	0.000s
Committing GAUGE-001 only affects its trades (2 trades).	`data['affected_trades'] == 2`	Pass	0.029s

Test Description	Acceptance Criteria	Result	Time
Reset all gauges.	`resp.status_code == 200; 'all' in data['message']`	Pass	0.003s
Reset one gauge.	`resp.status_code == 200; data['status'] == 'success'` (+1 more)	Pass	0.003s
GET returns per-gauge per-trigger structure.	`resp.status_code == 200; data['status'] == 'success'` (+1 more)	Pass	0.002s
Tenor outside 1-10 returns 400.	`resp.status_code == 400`	Pass	0.000s
Missing gauge_id returns 400.	`resp.status_code == 400`	Pass	0.000s
POST updates one gauge/trigger/tenor point.	`resp.status_code == 200; data['status'] == 'success'`	Pass	0.005s
Header footer callable	—	Pass	0.002s
Header footer callable	—	Pass	0.006s
Header footer callable	—	Pass	0.002s
Header footer draws confidential	`mock_canvas.drawCentredString.called`	Pass	0.003s
Header footer draws confidential footer	`mock_canvas.drawCentredString.called`	Pass	0.002s
Header footer draws header strings	`mock_canvas.drawString.call_count >= 1; mock_canvas.drawRightString.call_count >= 2`	Pass	0.003s
Header footer draws strings	`mock_canvas.drawString.call_count >= 1; mock_canvas.drawRightString.call_count >= 2`	Pass	0.002s
Health 200 at api v1	`r.status_code == 200`	Pass	0.001s
Health 200 at root	`r.status_code == 200`	Pass	0.000s
A single function must handle both route decorators.	`api_idx != -1 and plain_idx != -1 and fn_idx != -1; api_idx < fn_idx` (+1 more)	Pass	0.000s
health.py must declare /api/v1/health as a route alias.	`'/api/v1/health' in src`	Pass	0.001s
Backward-compat: /health must still exist.	`'"/health"' in src`	Pass	0.000s
Health returns ok status	`data['status'] == 'ok'`	Pass	0.000s
Test GET /health/detailed returns detailed status.	`response.status_code in [200, 503]; data['status'] in ['healthy', 'degraded']` (+2 more)	Pass	0.000s
Test GET /health returns health status.	`response.status_code == 200; data['status'] == 'ok'` (+1 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Test GET / redirects to catchment selector.	`response.status_code == 302`	Pass	0.000s
Medium height 10 to 15m	`isinstance(result, list)`	Pass	0.000s
No height skips classification	`isinstance(result, list)`	Pass	0.000s
Standard height under 10m	`isinstance(result, list)`	Pass	0.000s
Tall structure over 15m	`isinstance(result, list)`	Pass	0.000s
Generate monitoring high score	`"Weekly" in r.values()`	Pass	0.000s
Generate monitoring low score	`"Annually" in r.values()`	Pass	0.000s
Generate monitoring medium score	`"Monthly" in r.values()`	Pass	0.000s
Generate recommendations high score	`any("URGENT" in rec for rec in r)`	Pass	0.000s
Generate recommendations low score	`any("LOW" in rec for rec in r)`	Pass	0.000s
Generate recommendations medium score	`any("HIGH PRIORITY" in rec or "MODERATE" in rec for rec i...`	Pass	0.000s
Identify key factors high risk	`any("High Risk Factor" in k for k in r.keys())`	Pass	0.000s
Identify key factors medium risk	`any("Medium Risk Factor" in k for k in r.keys())`	Pass	0.000s
Status colour fresh	`col == dl_mod.GREEN`	Pass	0.000s
Status colour missing	`col == dl_mod.RED`	Pass	0.000s
Status colour stale	`col == dl_mod.AMBER`	Pass	0.000s
Status colour unknown	`col == dl_mod.STEEL`	Pass	0.000s
Status label unknown uppercased	`dl_mod._status_label('degraded') == 'DEGRADED'`	Pass	0.000s
Status label values	`dl_mod._status_label('fresh') == 'FRESH'; dl_mod._status_label('stale') == 'STALE'` (+1 more)	Pass	0.000s
Alert colour amber	`'#FFC107' in hist_js`	Pass	0.000s
Daily observations field	`'daily_observations' in hist_js`	Pass	0.000s
Renders historical chart	`'hist-timeseries-chart' in hist_js`	Pass	0.000s
Severe colour red	`'#F44336' in hist_js`	Pass	0.000s
Shows flood alert threshold	`'FloodAlert' in hist_js`	Pass	0.000s
Shows flood warning threshold	`'FloodWarning' in hist_js`	Pass	0.000s
Shows severe flood threshold	`'SevereFloodWarning' in hist_js`	Pass	0.000s
Subsamples observations	`'7' in hist_js; 'sampleObs' in hist_js`	Pass	0.000s
Warning colour orange	`'#FF9800' in hist_js`	Pass	0.000s
Cache keyed by gauge id	`'_gaugeId' in hist_js`	Pass	0.000s
Caches history data	`'_histData' in hist_js`	Pass	0.000s
Uses cached data on revisit	`'_histData._gaugeId' in hist_js or \{} '_his...`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Exceeded severe label	"Exceeded severe warning level" in str(elements)	Pass	0.001s
Within severe level	"4.000m" in str(elements)	Pass	0.001s
Cleanup function	'_cleanupHistCharts' in hist_js	Pass	0.000s
Stats bar flood exceedances	'flood_exceedances' in hist_js	Pass	0.000s
Stats bar mean level	'mean_level' in hist_js or 'Mean' in hist_js	Pass	0.000s
Stats bar record years	'record_years' in hist_js or 'Record' in hist_js	Pass	0.000s
Should fetch from /api/v1/gauges/¡gauge_id¿/history.	'/api/v1/gauges/' in js; '/history' in js	Pass	0.000s
Should fetch from /api/v1/trading/stress/storms.	'/api/v1/trading/stress/storms' in js	Pass	0.000s
Right panel should show storm scenarios list.	'hist-storms-panel' in js; 'hist-storms-list' in js (+1 more)	Pass	0.000s
Should render a time series chart.	'hist-timeseries-chart' in js; 'renderHistorical' in js	Pass	0.000s
Historical tab JS should render without errors.	len(js) > 0	Pass	0.000s
Should have function to render storm list.	'_renderStormList' in js; 'hist-storm-row' in js	Pass	0.000s
Chart should show flood stage threshold lines.	'FloodAlert' in js; 'FloodWarning' in js (+1 more)	Pass	0.000s
Storm rows should show trigger-colored dots.	'#FFC107' in js; '#FF9800' in js (+1 more)	Pass	0.000s
Storm row click should navigate to Stress Test tab.	'_navigateToStress' in js; '_stressStormHint' in js (+1 more)	Pass	0.000s
Should use addEventListener (not inline onclick) for storms.	'addEventListener' in js	Pass	0.000s
ghc_historical.get_js() returns substantial JS output.	len(hist_js) > 1000	Pass	0.000s
Output long enough to contain real function definitions.	len(hist_js) > 3000	Pass	0.000s
Historical tab fetches from /gauges/¡gauge_id¿/history.	'/api/v1/gauges/' in hist_js; '/history' in hist_js	Pass	0.000s
Fetch URL is built using gaugeId variable.	'gaugeId' in hist_js	Pass	0.000s
history.py lines 88-90: corrupted gaugehd file -¿ 500.	r.status_code == 500; r.get_json()["status"] == "error"	Pass	0.030s
history.py lines 144-146: statistics loader exception -¿ 500.	r.status_code == 500	Pass	0.030s

Test Description	Acceptance Criteria	Result	Time
history.py lines 119-121: timeseries loader exception -¿ 500.	`r.status_code == 500`	Pass	0.029s
Assess historical risks active subsidence	`"Ground Stability Risk" in result; "High" in result["Ground Stability Risk"]`	Pass	0.000s
Assess historical risks high env	`"Environmental Risk" in result; "High" in result["Environmental Risk"]`	Pass	0.000s
Assess historical risks low env	`"Low" in result["Environmental Risk"]`	Pass	0.000s
Assess historical risks minor subsidence	`"Medium" in result["Ground Stability Risk"]`	Pass	0.000s
Assess historical risks moderate env	`"Medium" in result["Environmental Risk"]`	Pass	0.000s
Assess historical risks moderate flood	`"Medium" in result["Historical Flood Risk"]`	Pass	0.000s
Assess historical risks no flood damage	`"Low" in result["Historical Flood Risk"]`	Pass	0.000s
Assess historical risks severe flood	`"Historical Flood Risk" in result; "High" in result["Historical Flood Risk"]`	Pass	0.000s
Assess historical risks stable ground	`"Low" in result["Ground Stability Risk"]`	Pass	0.000s
Build flood history bad json	`any("Error" in t or "error" in t.lower() for t in texts)`	Pass	0.001s
Build flood history no flood days	`any("No flood events" in t for t in texts)`	Pass	0.002s
Build flood history no hd file	`any("No historical" in t or "historical" in t.lower() for...`	Pass	0.000s
Build flood history no reference gauges	`any("No reference gauges" in t for t in texts)`	Pass	0.000s
Build flood history with flood data	`isinstance(result, list); len(tables) >= 1`	Pass	0.001s
Empty flood events	`isinstance(result, list)`	Pass	0.000s
Empty property does not crash	`isinstance(result, list)`	Pass	0.000s
Has history header	`any("History" in t or "Flood" in t for t in texts)`	Pass	0.000s
No flood history	`isinstance(result, list)`	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
With environmental issues	`isinstance(result, list); len(tables) >= 1`	Pass	0.000s
With fire incidents	`isinstance(result, list)`	Pass	0.000s
With ground conditions	`isinstance(result, list)`	Pass	0.000s
With mortgage data	`isinstance(result, list)`	Pass	0.000s
Hmda minimal fields	`isinstance(result, list)`	Pass	0.000s
Hmda rate spread formatted	`isinstance(result, list)`	Pass	0.000s
Hmda section creates table	`len(tables) >= 1`	Pass	0.000s
When gaugets file has invalid JSON, exception is caught	`resp.status_code == 200; data['status'] == 'success'`	Pass	0.119s
When gaugets file has ¿= STORM_HOURS readings, hydrograph is truncated.	`resp.status_code == 200`	Pass	0.053s
When gaugets JSON is corrupt, route returns 404.	`resp.status_code == 404; data['status'] == 'error'` (+1 more)	Pass	0.029s
When gaugets has fewer than STORM_HOURS readings, pads with last value.	`resp.status_code == 200; len(data['hourly']) == 168`	Pass	0.032s
Remaining exactly zero	`len(levels) == 5; abs(levels[0] - 3.0) < 0.01`	Pass	0.000s
run_series result matches manual step-by-step.	`qf == pytest.approx(qf_series[h], rel=1e-6)`	Pass	0.000s
Wet initial state produces higher early quickflow.	`qf_wet[:6].mean() > qf_dry[:6].mean()`	Pass	0.000s
No rain no quickflow	`np.all(qf == pytest.approx(0.0))`	Pass	0.000s
Quickflow positive	`np.all(qf >= 0.0)`	Pass	0.000s
Shape	`qf.shape == (168,); len(states) == 168`	Pass	0.000s
Soil moisture drains to near zero after long dry period.	`states[-1].soil_moisture < 1.0`	Pass	0.000s
Dry start no precip no quickflow	`qf == pytest.approx(0.0); new_state.soil_moisture == pytest.approx(0.0)` (+1 more)	Pass	0.000s
Groundwater accumulates from soil	`new_state.groundwater > 0.0`	Pass	0.000s
Negative precip treated as zero	`qf >= 0.0; new_state.soil_moisture >= 0.0`	Pass	0.000s
Precip increases soil moisture	`new_state.soil_moisture > 0.0`	Pass	0.000s
Precip produces quickflow	`qf > 0.0`	Pass	0.000s
Soil moisture drains without rain	`new_state.soil_moisture < 50.0`	Pass	0.000s
Elevated antecedent state produces more quickflow per mm.	`qf_wet > qf_dry`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Idw gauge spreads computed with two gauges	`"severe" in idw; len(idw["severe"]) == len(TENORS)`	Pass	0.087s
With empty nearest_gauges, idw_gauge_spreads is still a dict.	`isinstance(result.get("idw_gauge_spreads"), dict)`	Pass	0.087s
Fair income 30k to 50k	`isinstance(result, list)`	Pass	0.000s
Good income 50k to 100k	`isinstance(result, list)`	Pass	0.000s
High income over 100k	`isinstance(result, list)`	Pass	0.000s
Limited income under 30k	`isinstance(result, list)`	Pass	0.000s
Total income with secondary	`isinstance(result, list)`	Pass	0.000s
f_imperv=1 -¿ Y_max=0 -¿ no infiltration -¿ depth unchanged.	—	Pass	0.000s
Reduces depth	`np.all(result <= raw + 1e-10); result[50] < raw[50]`	Pass	0.000s
After Y_max is consumed, depth passes through unchanged.	`result[0] < raw[0]; result[1] == pytest.approx(raw[1])`	Pass	0.000s
Zero depth	—	Pass	0.000s
Input dir accepts string	`isinstance(gen.input_dir, Path)`	Pass	0.000s
Input dir set	`gen.input_dir == tmp_path`	Pass	0.000s
Output path explicit	`gen.output_path == out`	Pass	0.000s
Styles initialised	`gen.title_style is not None; gen.body_style is not None` (+1 more)	Pass	0.000s
Known inputs	`"properties" in m["gauge.json"]; "gaugehd" in m["gauge.json"]`	Pass	0.000s
Area = float('inf') skips all threshold checks, hits fallback.	`isinstance(result, tuple); len(result) == 2`	Pass	0.000s
Base rates by level	`result == pytest.approx(rate)`	Pass	0.000s
Capped at 10 pct	`result == pytest.approx(0.10)`	Pass	0.000s
Depth increases premium	`with_depth > no_depth`	Pass	0.000s
Depth multiplier formula	`rate == pytest.approx(expected)`	Pass	0.000s
Unknown level uses default	`result == pytest.approx(0.005)`	Pass	0.000s
Zero depth no effect	`base == zero`	Pass	0.000s
Unknown integer field returns int	`isinstance(result, int); 1 <= result <= 10`	Pass	0.000s
Integer type returns int	`isinstance(result, int); 0 <= result <= 100`	Pass	0.000s
Baseline alert only	`self._run(tmp_path, n_severe=0, n_warning=0) == "baseline"`	Pass	0.002s
Catastrophic 10 severe	`self._run(tmp_path, n_severe=10) == "catastrophic"`	Pass	0.010s

Test Description	Acceptance Criteria	Result	Time
Extreme 6 severe	`self._run(tmp_path, n_severe=6) == "extreme"`	Pass	0.007s
Moderate 1 severe	`self._run(tmp_path, n_severe=1) == "moderate"`	Pass	0.003s
Severe 3 severe	`self._run(tmp_path, n_severe=3) == "severe"`	Pass	0.009s
Running ‘python -m models.intensity‘ executes cli.main().	`result.returncode == 0; "Storm Intensity" in result.stdout or "usage" in result.s...`	Pass	1.660s
Beta value	`IntensityProfile.BETA.value == "beta"`	Pass	0.000s
Gaussian value	`IntensityProfile.GAUSSIAN.value == "gaussian"; DecayKernel.GAUSSIAN.value == "gaussian"`	Pass	0.000s
Triangular value	`IntensityProfile.TRIANGULAR.value == "triangular"`	Pass	0.000s
All profiles produce valid storm	`isinstance(storm, Storm); len(storm.track) > 0`	Pass	0.000s
Beta profile creates storm	`storm is not None; len(storm.track) > 0`	Pass	0.000s
Gaussian profile peak near midpoint	`8 <= peak_idx <= 16`	Pass	0.000s
Triangular max equals peak intensity	`max(tp.intensity for tp in storm.track) == pytest.approx(...`	Pass	0.000s
Triangular profile peak near midpoint	`8 <= peak_idx <= 16`	Pass	0.000s
Unknown sequence type falls back to 1 storm.	`count == 1`	Pass	0.001s
Monotone increasing	`all(levels[i] < levels[i + 1] for i in range(len(levels) ...`	Pass	0.000s
Negative intensity returns zero	`model.intensity_to_level(-10.0, gauge) == 0.0`	Pass	0.000s
Positive intensity positive level	`model.intensity_to_level(50.0, gauge) > 0.0`	Pass	0.000s
Sensitivity scales output	`model.intensity_to_level(50, g2) == pytest.approx( ...`	Pass	0.000s
Uses historical high when set	`lv_with != lv_without`	Pass	0.000s
Zero intensity returns zero	`model.intensity_to_level(0.0, gauge) == 0.0`	Pass	0.000s
ContextMenuHandler JS renders without f-string errors.	`len(js) > 0; '<script>' in js`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
GaugeHazardCurve JS renders without f-string errors.	`len(js) > 0; '<script>' in js`	Pass	0.000s
GaugePDFPanel JS renders without f-string errors.	`len(js) > 0; 'GaugePDFPanel' in js`	Pass	0.000s
GaugeStormAnalysis JS renders without f-string errors.	`len(js) > 0; '<script>' in js`	Pass	0.000s
NotificationSystem JS renders without f-string errors.	`len(js) > 0; '<script>' in js`	Pass	0.001s
PropertyHazardCurvePanel JS renders without f-string errors.	`len(js) > 0; '<script>' in js`	Pass	0.000s
PropertyPDFPanel JS renders without f-string errors.	`len(js) > 0; 'PropertyPDFPanel' in js`	Pass	0.000s
PropertyStormAnalysis JS renders without f-string errors.	`len(js) > 0; '<script>' in js` (+1 more)	Pass	0.000s
PropertyStormAnalysis JS must guard against empty flood_events array.	`'flood_events.length === 0' in js`	Pass	0.000s
TradingDeskPanel should be in __all__.	`'TradingDeskPanel' in __all__`	Pass	0.000s
InteractivityManager should have trading_desk attribute.	`hasattr(mgr, 'trading_desk')`	Pass	0.000s
Setup map adds preloader before notifications	`preloader_idx != -1; notifications_idx != -1` (+1 more)	Pass	0.000s
Startup preloader imported in init	`'from .startup import StartupPreloader' in src`	Pass	0.001s
Startup preloader instantiated in init	`'self.startup_preloader = StartupPreloader()' in src`	Pass	0.000s
Elevated rate 5 to 7	`isinstance(result, list)`	Pass	0.000s
High rate over 7	`isinstance(result, list)`	Pass	0.000s
Low rate under 3	`isinstance(result, list)`	Pass	0.000s
Moderate rate 3 to 5	`isinstance(result, list)`	Pass	0.000s
No rate unknown	`isinstance(result, list)`	Pass	0.000s
Line 64: empty yield_curve returns default fallback 0.04.	`result == 0.04`	Pass	0.000s
Line 64: empty yield_curve returns fallback.	`result == 0.05`	Pass	0.000s
Line 77: fallthrough at end of for loop returns last key rate.	`interpolate_yield_rate(curve, 4.0) == 0.04; result == pytest.approx(0.035)`	Pass	0.000s
Line 71: t ¿= keys[-1] returns the last key's rate.	`result == 0.045`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Line 71: t == keys[-1] returns the last key's rate.	`result == 0.045`	Pass	0.000s
Interpolation at a grid point should return the exact grid value.	`abs(val - expected) < 1e-9`	Pass	0.000s
At origin	`abs(val - expected) < 1e-9`	Pass	0.000s
Distance ¿ 5000m should clamp to 5000m.	`abs(val_clamped - val_edge) < 1e-9`	Pass	0.000s
Elevation ¿ 5.0m should clamp to 5.0m.	`abs(val_clamped - val_edge) < 1e-9`	Pass	0.000s
Negative distance should clamp to 0.	`abs(val_clamped - val_edge) < 1e-9`	Pass	0.000s
Negative elevation should clamp to 0.	`abs(val_clamped - val_edge) < 1e-9`	Pass	0.000s
Moving to a higher-risk zone should increase spread.	`s_high > s_low`	Pass	0.000s
Terrain delta between same zone should be exactly zero.	`abs(s1 - s2) < 1e-12`	Pass	0.000s
All four corners of each cell should match the grid exactly. [zone='Zone 1']	`abs(val - expected) < 1e-9`	Pass	0.000s
All four corners of each cell should match the grid exactly. [zone='Zone 2']	`abs(val - expected) < 1e-9`	Pass	0.000s
All four corners of each cell should match the grid exactly. [zone='Zone 3']	`abs(val - expected) < 1e-9`	Pass	0.000s
All four corners of each cell should match the grid exactly. [zone='Zone 3a']	`abs(val - expected) < 1e-9`	Pass	0.000s
All four corners of each cell should match the grid exactly. [zone='Zone 3b']	`abs(val - expected) < 1e-9`	Pass	0.000s
Interpolation at midpoint should be between bracketing values.	`lo <= val <= hi`	Pass	0.000s
File exists	`INVENTORY_PATH.exists()`	Pass	0.000s
File is valid json	`isinstance(data, dict)`	Pass	0.001s
Has at least one model	`len(models) > 0`	Pass	0.000s
Has metadata	`"metadata" in inventory; "framework" in inventory["metadata"]`	Pass	0.000s
Has model chain	`"model_chain" in inventory`	Pass	0.000s
Has models key	`"models" in inventory`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Get models includes risk fields	`response.status_code == 200`; `data["status"] == "success"` (+4 more)	Pass	0.002s
One prob returns min	`d.inverse_exceedance(1) == d.params.min_intensity`	Pass	0.000s
Result in bounds	`d.params.min_intensity <= v <= d.params.max_intensity`	Pass	0.000s
Roundtrip above threshold	`abs(recovered - target) < 3.0`	Pass	0.000s
Roundtrip below threshold	`abs(recovered - target) < 2.0`	Pass	0.000s
Zero prob returns max	`d.inverse_exceedance(0) == d.params.max_intensity`	Pass	0.000s
Line 125: _inverse_normal(1.5) returns +inf.	`result == float('inf')`	Pass	0.000s
Line 123: _inverse_normal(-0.5) returns -inf.	`result == float('-inf')`	Pass	0.000s
Line 125: _inverse_normal(1.0) returns +inf.	`result == float('inf')`	Pass	0.000s
Line 123: _inverse_normal(0) returns -inf.	`result == float('-inf')`	Pass	0.000s
Lines 170-171: p ¿ p_high (0.97575) uses the upper tail formula.	`result == pytest.approx(2.326, abs=0.02)`	Pass	0.000s
Lines 170-171: p=0.98 also triggers high-p branch.	`result == pytest.approx(2.054, abs=0.02)`	Pass	0.000s
Lines 170-171: high-p branch returns positive value.	`result > 0`	Pass	0.000s
Lines 170-171: verify roundtrip CDF(inverse_CDF(p)) p for high p.	`recovered == pytest.approx(p, abs=0.001)`	Pass	0.000s
Below average 2 to 4pct	`isinstance(result, list)`	Pass	0.000s
Excellent over 8pct	`isinstance(result, list)`	Pass	0.000s
Fair 4 to 6pct	`isinstance(result, list)`	Pass	0.000s
Good 6 to 8pct	`isinstance(result, list)`	Pass	0.000s
Poor under 2pct	`isinstance(result, list)`	Pass	0.000s
Get raises on unknown key	—	Pass	0.000s
JSONFileConfig.get('hazard_curves') returns the hazard curves filename.	`result == JSONFileConfig.HAZARD_CURVES`	Pass	0.000s
Datetime encoded as isoformat	`"2024-01-15" in result`	Pass	0.000s
Numpy float32 encoded	`abs(result - 1.5) < 0.01`	Pass	0.000s
Numpy float encoded	`abs(result - 3.14) < 1e-6`	Pass	0.000s
Numpy int32 encoded	`result == 7`	Pass	0.000s
Numpy int encoded	`result == 42`; `isinstance(result, int)`	Pass	0.000s
Numpy ndarray encoded as list	`result == [1, 2, 3]`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Unknown type raises type error	—	Pass	0.000s
Charts reference first_trigger_hour for KO line.	`'first_trigger_hour' in stress_js`	Pass	0.000s
Charts have KO annotation line.	`'koLine' in stress_js`	Pass	0.000s
Shows KO H{n} label for triggered trades.	`'KO H' in stress_js`	Pass	0.000s
References triggered_hour for KO status.	`'triggered_hour' in stress_js`	Pass	0.000s
Custom label width	`t._colWidths[0] == pytest.approx(lw); t._colWidths[1] == pytest.approx(AVAIL_WIDTH - lw)`	Pass	0.000s
Default label width	`t._colWidths[0] == pytest.approx(2.8 * inch)`	Pass	0.000s
Multiple pairs	`len(t._cellvalues) == 3`	Pass	0.000s
Returns table	`isinstance(t, Table); isinstance(t, Table)`	Pass	0.000s
Numeric values are str()-ified in Paragraph.	`isinstance(t, Table)`	Pass	0.000s
LTV ¿ 1 (e.g. 75 instead of 0.75) should still work.	`isinstance(result, list)`	Pass	0.000s
Ltv over 60 low medium	`isinstance(result, list)`	Pass	0.000s
Ltv over 70 medium	`isinstance(result, list)`	Pass	0.000s
Ltv over 80 medium high	`isinstance(result, list)`	Pass	0.000s
Ltv over 90 high risk	`isinstance(result, list)`	Pass	0.000s
Ltv under 60 low risk	`isinstance(result, list)`	Pass	0.000s
High ltv 90 to 95	`isinstance(result, list)`	Pass	0.001s
Low ltv under 70	`isinstance(result, list)`	Pass	0.000s
Low medium ltv 70 to 80	`isinstance(result, list)`	Pass	0.001s
Medium ltv 80 to 90	`isinstance(result, list)`	Pass	0.001s
Very high ltv over 95	`isinstance(result, list)`	Pass	0.001s
At high boundary	`RiskAssessor.assess_ltv_risk_level(0.95) == "High"`	Pass	0.000s
At low boundary	`RiskAssessor.assess_ltv_risk_level(0.6) == "Low"`	Pass	0.000s
At moderate boundary	`RiskAssessor.assess_ltv_risk_level(0.8) == "Moderate"`	Pass	0.000s
Critical ltv	`RiskAssessor.assess_ltv_risk_level(0.98) == "Critical"`	Pass	0.000s
High ltv	`RiskAssessor.assess_ltv_risk_level(0.88) == "High"`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Low ltv	`RiskAssessor.assess_ltv_risk_level(0.5) == "Low"`	Pass	0.000s
Moderate ltv	`RiskAssessor.assess_ltv_risk_level(0.7) == "Moderate"`	Pass	0.000s
None returns unknown	`RiskAssessor.assess_flood_risk_level(None) == "Unknown"; RiskAssessor.assess_ltv_risk_level(None) == "Unknown"`	Pass	0.000s
Normalises 95 pct	`RiskAssessor.assess_ltv_risk_level(95) == "High"`	Pass	0.000s
Normalises percentage input	`RiskAssessor.assess_ltv_risk_level(70) == "Moderate"`	Pass	0.000s
Bad date string gracefully skipped	`isinstance(elements, list)`	Pass	0.000s
Moderate recent activity	`"Moderate recent activity" in str(elements)`	Pass	0.000s
No recent activity	`"No recent severe activity" in str(elements)`	Pass	0.000s
Recent activity within 1 year	`"Recent severe level activity" in str(elements)`	Pass	0.000s
Body style alignment	`s.alignment == TA_CENTER`	Pass	0.000s
Body style bold	`s.fontName == 'Helvetica-Bold'`	Pass	0.000s
Body style default	`isinstance(s, ParagraphStyle); s.fontName == 'Helvetica'`	Pass	0.000s
Build header footer callable	`callable(cb)`	Pass	0.000s
White hdr style returns paragraph style	`isinstance(s, ParagraphStyle); s.fontName == 'Helvetica-Bold'`	Pass	0.000s
Legacy file uses mtime cache — second call returns cached.	`r1["storms"][0]["storm_id"] == "LEGACY-002"; r2 is r1`	Pass	0.005s
Falls back to stress_storms.json when _index.json absent.	`result is not None; result["storms"][0]["storm_id"] == "LEGACY-001"`	Pass	0.004s
OSError on legacy file stat returns existing cache.	`result == {"storms": [{"storm_id": "cached-legacy"}]}`	Pass	0.003s
Coverage available false when no index	`r.get_json()["coverage_available"] is False`	Pass	0.001s
Coverage available true when index present	`r.get_json()["coverage_available"] is True`	Pass	0.001s
Empty dir returns empty list	`data["reports"] == []`	Pass	0.001s
File has size and modified	`reports[0]["size"] > 0; "modified" in reports[0]`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
File type label json	`r.get_json()["reports"][0]["type"] == "JSON"`	Pass	0.001s
File type label pdf	`r.get_json()["reports"][0]["type"] == "PDF"`	Pass	0.001s
File type label unknown	`r.get_json()["reports"][0]["type"] == "XYZ"`	Pass	0.001s
No dir returns empty list	`r.status_code == 200; data["status"] == "success"` (+2 more)	Pass	0.000s
Pdf file listed	`len(data["reports"]) == 1; data["reports"][0]["filename"] == "report.pdf"`	Pass	0.001s
Subdirectories not listed	`r.get_json()["reports"] == []`	Pass	0.001s
sys.exit(0) fires before the gauge file is opened.	—	Pass	0.002s
sys.exit(0) fires before the property file is opened.	—	Pass	0.002s
Lines 324-327: all category/page pairs are logged.	—	Pass	0.001s
Lines 412-419: –list-categories iterates category dict, sys.exit(0).	—	Pass	0.002s
Lines 469-476: –list-categories iterates category dict, sys.exit(0).	—	Pass	0.002s
Lines 321-328: –list-categories → get_page_categories() iterated, sys.exit(0).	—	Pass	0.001s
No file returns empty list	`r.status_code == 200; data["status"] == "success"` (+2 more)	Pass	0.000s
Summary fields	`first["counterparty_id"] == "CTPY-001"; first["name"] == "Test Bank PLC"` (+2 more)	Pass	0.002s
With data returns summary	`r.status_code == 200; data["status"] == "success"` (+1 more)	Pass	0.002s
No storms file returns 404	`r.status_code == 404`	Pass	0.000s
Options returns ok	`r.status_code == 200`	Pass	0.000s
Storm has expected fields	`len(storms) >= 1; "storm_id" in s` (+3 more)	Pass	0.006s
Storm name is capitalised intensity_category, not empty.	`s["name"] == s["intensity_category"].capitalize()`	Pass	0.004s
Storms list count	`data["count"] >= 1`	Pass	0.005s
Storms sorted by gauges_severe descending (most severe first).	`counts == sorted(counts, reverse=True)`	Pass	0.005s

Test Description	Acceptance Criteria	Result	Time
With data returns success	`r.status_code == 200`; `data["status"] == "success"` (+1 more)	Pass	0.003s
Created meeting appears in list	`len(data["meetings"]) == 1`; `data["meetings"][0]["title"] == "Board Meeting"`	Pass	0.003s
Empty list returns success	`r.status_code == 200`; `data["status"] == "success"` (+1 more)	Pass	0.001s
List summary has required keys	`key in m`	Pass	0.003s
Sorted by date desc	`dates == sorted(dates, reverse=True)`	Pass	0.005s
List mortgages empty	`r.status_code == 200`; `r.get_json()["count"] == 0`	Pass	0.001s
List mortgages exception returns 500	`r.status_code == 500`; `r.get_json()["status"] == "error"`	Pass	0.001s
List mortgages success	`r.status_code == 200`; `data["status"] == "success"` (+2 more)	Pass	0.001s
sys.exit(0) fires before the flood file is opened.	—	Pass	0.001s
sys.exit(0) fires before the gauge file is opened.	—	Pass	0.002s
sys.exit(0) fires before the property file is opened.	—	Pass	0.005s
Lines 405-410: –list-pages iterates available pages, sys.exit(0).	—	Pass	0.002s
Lines 462-467: –list-pages iterates available pages, sys.exit(0).	—	Pass	0.002s
Lines 314-319: –list-pages → list_available_pages() iterated, sys.exit(0).	—	Pass	0.002s
Loader exception returns 500	`r.status_code == 500`; `r.get_json()["status"] == "error"`	Pass	0.001s
Success empty list	`r.status_code == 200`; `data["status"] == "success"` (+2 more)	Pass	0.001s
Success with properties	`r.status_code == 200`; `r.get_json()["count"] == 2`	Pass	0.001s
Empty dir returns zero count	`r.status_code == 200`; `data["status"] == "success"` (+2 more)	Pass	0.001s
Returns trades list key	`"trades" in r.get_json()`	Pass	0.001s
Corrupt trade file returns 500	`r.status_code == 500`; `r.get_json()["status"] == "error"`	Pass	0.029s
list_prs_trades reads PRS-*.json files and returns them.	`r.status_code == 200`; `data["count"] == 1` (+1 more)	Pass	0.029s

Test Description	Acceptance Criteria	Result	Time
Loads list json	`result == [1, 2, 3]`	Pass	0.001s
Loads nested json	`_load(p) == data`	Pass	0.001s
Loads valid json	`_load(p) == {'key': 'value'}`	Pass	0.001s
Returns empty dict for directory	`_load(tmp_path) == {}`	Pass	0.000s
Returns empty dict for invalid json	`_load(p) == {}`	Pass	0.001s
Returns empty dict for missing file	`_load(tmp_path / 'nonexistent.json') == {}`	Pass	0.000s
Classifier count	`data['classifier_count'] == 2`	Pass	0.010s
Eod count defaults zero on import error	`isinstance(data['eod_count'], int)`	Pass	0.008s
Gaugehd baselines loaded	`len(data['gaugehd_baselines']) == 2`	Pass	0.008s
Gaugehd count	`data['gaugehd_count'] == 2`	Pass	0.009s
Gaugets count	`data['gaugets_count'] == 2`	Pass	0.009s
Loads counterparties	`len(data['counterparties']) == 1`	Pass	0.010s
Loads gaugehc	`'GAUGE-0001' in data['gaugehc']['hazard_curves']`	Pass	0.006s
Loads gauges	`len(data['gauges']) == 2`	Pass	0.008s
Loads mortgages	`len(data['mortgages']) == 1`	Pass	0.009s
Loads properties	`len(data['properties']) == 1`	Pass	0.008s
Loads propertyhc	`'PROP-0001' in data['propertyhc']['property_hazard_curves']`	Pass	0.007s
Loads seq summary	`data['seq_summary']['num_sequences'] == 10000`	Pass	0.007s
Loads stress storms index	`len(data['stress_storms']['storms']) == 1`	Pass	0.017s
Falls back to stress_storms.json when _index.json is absent.	`len(data['stress_storms']['storms']) == 2`	Pass	0.007s
When input dir is empty, everything defaults safely.	`data['gauges'] == []`; `data['properties'] == []` (+10 more)	Pass	0.001s
Propertyts count	`data['propertyts_count'] == 1`	Pass	0.009s
Stress storms empty when neither exists	`data['stress_storms'] == {}`	Pass	0.005s
config import may fail in test env — defaults to 0.	`isinstance(data['trade_count'], int)`	Pass	0.011s
Training summary	`data['training_summary']['avg_auc_roc'] == pytest.approx(...`	Pass	0.009s
Loads entries	`result[0]["event"] == "test"`	Pass	0.001s
Returns empty list on bad json	`_load_audit_log() == []`	Pass	0.001s
Returns empty list when missing	`_load_audit_log() == []`; `_load_meetings() == []`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Loads valid data	"principles" in result; "roles" in result	Pass	0.002s
Returns none when missing	_load_bcbs239() is None; _load_raci() is None	Pass	0.000s
Loads bibliography	result["references"][0]["id"] == "R1"	Pass	0.001s
Returns default when missing	result == {"references": [], "categories": []}	Pass	0.000s
Line 208: file exists with empty counterparties → fallback list.	len(result) == 21; result[0]["id"] == "CTPY-001" (+1 more)	Pass	0.003s
Line 207-209: file doesn't exist → fallback list.	len(result) == 21	Pass	0.000s
When file has entries, those are returned (not fallback).	len(result) == 1; result[0]["id"] == "CTPY-REAL" (+1 more)	Pass	0.002s
Corrupt counterparty file leaves data none	dl.loaded_data.counterparty_data is None	Pass	0.001s
Missing counterparty file leaves data none	dl.loaded_data.counterparty_data is None	Pass	0.000s
Valid counterparty file loads	dl.loaded_data.counterparty_data is not None; "counterparties" in dl.loaded_data.counterparty_data	Pass	0.002s
Missing gaugehc returns empty and default storms	curves == {}; n_storms == 1000	Pass	0.000s
Present gaugehc returns curves	"GAUGE-001" in curves; n_storms == 500	Pass	0.001s
Present gaugehc returns num storms	n_storms == 250	Pass	0.001s
Lines 59-64: get_reports_dir raises -¿ fallback to get_output_path.	result == []	Pass	0.000s
Line 67: prs_dir doesn't exist -¿ return empty list.	result == []	Pass	0.000s
Lines 75-76: trade with matching gauge_id -¿ appended to list.	len(result) == 1	Pass	0.002s
Trade for different gauge not appended.	result == []	Pass	0.002s
Computes summer mean	bl['summer'] == pytest.approx(1.0667, abs=0.001)	Pass	0.002s
Computes winter mean	bl['winter'] == pytest.approx(1.8)	Pass	0.002s
Handles missing monthly means	len(baselines) == 1; baselines[0]['winter'] is None (+1 more)	Pass	0.003s

Test Description	Acceptance Criteria	Result	Time
Loads baselines	`len(baselines) == 2`; `bl['gauge_id'] in ('GAUGE-0001', 'GAUGE-0002')` (+1 more)	Pass	0.005s
Returns empty for nonexistent dir	`loader._load_gaugehd_baselines(tmp_path / 'nope') == []`	Pass	0.000s
Skips files without mean level	`loader._load_gaugehd_baselines(ghd) == []`	Pass	0.001s
Skips invalid files	`loader._load_gaugehd_baselines(ghd) == []`	Pass	0.001s
Empty gauge list raises	—	Pass	0.001s
File not found raises	—	Pass	0.000s
Gauge has id	`result[0]["gauge_id"] == "GAUGE-999"`	Pass	0.001s
Gauge that maps to no gauge_id raises ValueError.	—	Pass	0.001s
Loads gauges	`isinstance(result, list)`; `len(result) == 1`	Pass	0.002s
Empty portfolio returns empty list	`result == []`	Pass	0.003s
Loads gauge from portfolio	`len(result) == 1`; `isinstance(result[0], GaugeConfig)` (+1 more)	Pass	0.002s
Loads documents	`_load_gov_documents()[0]["doc_id"] == "D1"`	Pass	0.001s
Returns empty list when missing	`_load_gov_documents() == []`	Pass	0.000s
Corrupt hazard file leaves hazard data none	`dl.loaded_data.hazard_data is None`	Pass	0.001s
Corrupt property hazard file leaves phc none	`dl.loaded_data.property_hazard_data is None`	Pass	0.001s
Loaded data has gauge	`"TEST-G001" in result["hazard_curves"]`	Pass	0.001s
Loads from file	`"hazard_curves" in result`	Pass	0.004s
Missing hazard file leaves hazard data none	`dl.loaded_data.hazard_data is None`	Pass	0.000s
Missing property hazard file leaves phc none	`dl.loaded_data.property_hazard_data is None`	Pass	0.000s
Valid hazard file populates hazard data	`dl.loaded_data.hazard_data is not None`; `"hazard_curves" in dl.loaded_data.hazard_data`	Pass	0.001s
Empty json object returns empty dict	`result == {}`	Pass	0.001s
Invalid json raises json decode error	—	Pass	0.002s
Missing file raises file not found	—	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Returns full structure	`result["meta"]["version"] == 2`; `"G-001" in result["hazard_curves"]`	Pass	0.002s
Loads valid file	`result["models"][0]["model_id"] == "M1"`	Pass	0.001s
Returns none on bad json	`_load_inventory() is None`	Pass	0.001s
Returns none when missing	`_load_inventory() is None`	Pass	0.000s
Loads existing results	`result['total'] == 15`; `result['failed'] == 1`	Pass	0.002s
Returns dict when missing	`isinstance(result, dict)`; `len(result) == 0`	Pass	0.000s
Returns empty skeleton for corrupt JSON.	`result == {"runs": {}, "steps": {}}`	Pass	0.001s
Load corrupt json returns skeleton	`result == {"runs": {}, "steps": {}}`	Pass	0.053s
Load missing returns skeleton	`result == {"runs": {}, "steps": {}}`	Pass	0.000s
Load valid manifest	`result == data`	Pass	0.023s
Reads and parses an existing manifest.	`'gauges' in result['steps']`	Pass	0.002s
Returns empty skeleton when file doesn't exist.	`isinstance(result, dict)`; `'runs' in result` (+1 more)	Pass	0.000s
Lines 89-91: exception -¿ returns {}.	`result == {}`	Pass	0.000s
Loads list	`_load_meetings()[0]["id"] == "M1"`	Pass	0.001s
Returns empty list when missing	`_load_audit_log() == []`; `_load_meetings() == []`	Pass	0.000s
When mortgage.json has invalid JSON, the exception is caught	`r.status_code == 200`; `len(props) == 1` (+1 more)	Pass	0.030s
Missing property json returns 404	`r.status_code == 404`	Pass	0.002s
Missing property json sequence returns 404	`r.status_code == 404`	Pass	0.002s
Loads valid data	`"principles" in result`; `"roles" in result`	Pass	0.002s
Returns none when missing	`_load_bcbs239() is None`; `_load_raci() is None`	Pass	0.000s
Adding a gauge to gaugehc.json causes reconciliation on next load.	`"GAUGE-A" in state["base_rates"]`; `"GAUGE-B" not in state["base_rates"]` (+2 more)	Pass	0.003s
Loading with same gauges doesn't trigger reconciliation.	`mtime1 == mtime2`	Pass	0.058s
Reconciled gauge has all triggers with 5 tenors each.	`trigger in ts`; `len(ts[trigger]) == 5` (+1 more)	Pass	0.004s

Test Description	Acceptance Criteria	Result	Time
Reconciled state is persisted so second load doesn't re-reconcile.	"GAUGE-B" in disk_state["base_rates"]; "GAUGE-B" in disk_state["hazard_term_structure"]	Pass	0.005s
Corrupt storm file leaves storm data none	dl.loaded_data.storm_data is None	Pass	0.001s
Missing storm file leaves storm data none	dl.loaded_data.storm_data is None	Pass	0.000s
Storms dict with storms key	dl.loaded_data.storm_data is not None	Pass	0.001s
Storms list format	dl.loaded_data.storm_data is not None	Pass	0.001s
Storms other type	dl.loaded_data.storm_data is not None	Pass	0.001s
File not found raises	—	Pass	0.000s
Loads list	isinstance(result, list); len(result) == 5	Pass	0.002s
Storm has id	result[0]["storm_id"] == "STORM-0000"	Pass	0.001s
Storm loaded by searching legacy storms list.	result is not None; result["storm_id"] == "LEGACY-STORM-001" (+1 more)	Pass	0.000s
When path.stat() raises OSError, return existing cache.	result == {"storms": [{"storm_id": "cached"}]}	Pass	0.000s
When path.stat() raises OSError and no cache, returns None.	result is None	Pass	0.000s
Empty data list records invalid result	dl._validation_results["empty_type"].is_valid is False; "No data loaded" in dl._validation_results["empty_type"]....	Pass	0.000s
Empty data list returns none	dl._load_with_validation(mock_loader, "empty_type") is None	Pass	0.000s
Exception in loader returns none	dl._load_with_validation(mock_loader, "error_type") is None	Pass	0.001s
Exception records error in validation	dl._validation_results["error_type"].is_valid is False; "disk error" in dl._validation_results["error_type"].erro...	Pass	0.000s
None return from loader returns none	dl._load_with_validation(mock_loader, "none_type") is None	Pass	0.000s
Successful load records valid result	"gauge_test" in dl._validation_results; dl._validation_results["gauge_test"].is_valid is True	Pass	0.001s
Successful load returns data dict	result is not None; result["count"] == 2 (+1 more)	Pass	0.000s
Validation summary count matches items	dl._validation_results["three_items"].summary["count"] == 3	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Validation summary has type field	`dl._validation_results["xtype"].summary["type"] == "xtype"`	Pass	0.000s
Can set gauge data	`d.gauge_data is not None`	Pass	0.000s
Can set multiple fields	`d.gaugets_count == 3; d.gaugehd_count == 5`	Pass	0.000s
Counts default to zero	`d.gaugets_count == 0; d.gaugehd_count == 0` (+1 more)	Pass	0.000s
Defaults are none	`getattr(d, field) is None`	Pass	0.000s
Is dataclass	`dataclasses.is_dataclass(LoadedData)`	Pass	0.000s
Lookup defaults are none	`getattr(d, field) is None`	Pass	0.000s
Test creating a registry.	`registry.data_dir == populated_data_dir; registry is not None` (+1 more)	Pass	0.000s
Test getting list of available loader types.	`'property' in loaders; 'mortgage' in loaders` (+3 more)	Pass	0.000s
Clear all caches	`not loader._cache_valid`	Pass	0.004s
Get all status	`"gauge" in status; "entity_name" in status["gauge"]`	Pass	0.003s
Invalidate all caches	—	Pass	0.000s
Test clearing all loader caches.	`prop_loader._cache is None; gauge_loader._cache is None`	Pass	0.007s
Test that different loader types are separate.	`prop_loader is not gauge_loader`	Pass	0.000s
Test that loader instances are cached.	`loader1 is loader2`	Pass	0.000s
Check data files all missing	`all(v is False for v in result.values())`	Pass	0.000s
Check data files present	`result["property"] is True; result["gauge"] is False`	Pass	0.000s
File overrides	`registry.get_gauge_loader().count() == 3`	Pass	0.002s
Get data dir	`LoaderRegistry(data_dir=tmp_path).get_data_dir() == tmp_path`	Pass	0.000s
Test providing custom filenames.	`len(props) == 1; loader.get_entity_id(props[0]) == "CUSTOM-001"`	Pass	0.001s
Test getting loader by name string.	`loader is not None; loader.count() == 3`	Pass	0.001s
Test getting unknown loader type.	`loader is None`	Pass	0.000s
Get available loaders	`"property" in available; "gauge" in available` (+1 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Get gauge loader	`isinstance(LoaderRegistry(data_dir=tmp_path).get_gauge()...`	Pass	0.000s
Get loader by name	`LoaderRegistry(data_dir=tmp_path).get_loader("gauge") is ...`	Pass	0.000s
Get loader unknown name returns none	`LoaderRegistry(data_dir=tmp_path).get_loader("nonexistent...`	Pass	0.000s
Get mortgage loader	`isinstance(LoaderRegistry(data_dir=tmp_path).get_mortgage()...`	Pass	0.000s
Get property loader	`isinstance(LoaderRegistry(data_dir=tmp_path).get_property()...`	Pass	0.000s
Get storm loader	`isinstance(LoaderRegistry(data_dir=tmp_path).get_storm()...`	Pass	0.000s
Get timeseries loader	`isinstance(LoaderRegistry(data_dir=tmp_path).get_timeseries()...`	Pass	0.000s
Loader cached on repeated access	`registry.get_property_loader() is registry.get_property_l...`	Pass	0.000s
Test checking data file existence.	`status['property'] is True; status['mortgage'] is True` (+3 more)	Pass	0.000s
Test getting status for all loaders.	`'property' in status; 'gauge' in status` (+1 more)	Skip	0.000s
Test getting GaugeLoader.	`isinstance(loader, GaugeLoader); loader.count() == 3`	Pass	0.001s
Test getting MortgageLoader.	`isinstance(loader, MortgageLoader); loader.count() == 3`	Pass	0.001s
Test getting PropertyLoader.	`isinstance(loader, PropertyLoader); loader.count() == 3`	Pass	0.001s
Test getting StormLoader.	`isinstance(loader, StormLoader)`	Pass	0.000s
Test getting TimeseriesLoader.	`isinstance(loader, TimeseriesLoader)`	Pass	0.000s
Address subsection header	`any("Primary Address" in t for t in texts)`	Pass	0.000s
Administrative subsection header	`any("Administrative Areas" in t for t in texts)`	Pass	0.000s
Area classification subsection header	`any("Area Classification" in t for t in texts)`	Pass	0.000s
Density high	`isinstance(result, list)`	Pass	0.000s
Density low	`isinstance(result, list)`	Pass	0.000s
Density low medium	`isinstance(result, list)`	Pass	0.000s
Density medium	`isinstance(result, list)`	Pass	0.000s
Density very high	`isinstance(result, list)`	Pass	0.000s
Empty property does not crash	`isinstance(result, list)`	Pass	0.000s
Geographic subsection header	`any("Geographic Information" in t for t in texts)`	Pass	0.000s
Has section header	`any("Location Details" in t for t in texts)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Has tables	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
Missing lat/lon means coordinates rows are omitted — no crash.	`isinstance(result, list)`	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0`	Pass	0.000s
With mortgage data does not crash	`isinstance(result, list)`	Pass	0.000s
Generated locations must include vertical_offset.	`'vertical_offset' in loc; loc['vertical_offset'] >= 0.0`	Pass	0.000s
Debug does not raise	—	Pass	0.000s
Debug level does not raise	—	Pass	0.000s
Error does not raise	—	Pass	0.000s
Error level does not raise	—	Pass	0.000s
Info does not raise	—	Pass	0.000s
Info level does not raise	—	Pass	0.000s
Log does not raise	—	Pass	0.000s
Log with empty string	—	Pass	0.000s
Success falls back gracefully	—	Pass	0.000s
Success level falls back gracefully	—	Pass	0.000s
Unknown level falls back to info	—	Pass	0.000s
Warning does not raise	—	Pass	0.000s
Warning level does not raise	—	Pass	0.000s
Appends entry	`len(entries) == 1; entries[0]["action"] == "test_action"` (+1 more)	Pass	0.047s
Failures in audit logging must not propagate.	—	Pass	0.000s
Corrupted audit log → start fresh, don't raise.	`len(entries) == 1`	Pass	0.003s
Creates audit file	`os.path.exists(audit_path)`	Pass	0.001s
Defaults applied when no json	`r.status_code == 200; entry["event_type"] == "usage"` (+1 more)	Pass	0.003s
Entry has expected fields	`entry["model_id"] == "MKM-TEST-001"; entry["event_type"] == "calibration"` (+2 more)	Pass	0.002s
Entry has required keys	`"timestamp" in entry; "model_id" in entry` (+3 more)	Pass	0.002s
Existing file appended	`len(entries) == 2; "existing" in entries[0]`	Pass	0.003s

Test Description	Acceptance Criteria	Result	Time
Model not found returns 404	`r.status_code == 404`; `r.get_json()["status"] == "error"` (+1 more)	Pass	0.002s
Multiple calls accumulate	`len(entries) == 2`	Pass	0.027s
Parameters stored	`r.get_json()["entry"]["parameters"] == {"key": "value"}`	Pass	0.003s
Valid model returns 200	`r.status_code == 200`; `r.get_json()["status"] == "success"`	Pass	0.003s
ltv_ratio ¿ 1 is interpreted as a percentage and divided by 100.	`score > 0`	Pass	0.000s
Compare mode	`"Scenario Comparison" in captured.out`	Pass	0.366s
Custom notional and spread	`rc == 0`	Pass	0.002s
Default describe mode	`len(captured.out) > 0`	Pass	0.079s
Default gauge selection returns zero	`rc == 0`	Pass	0.002s
Exceedance json mode	`"exceedance_probability" in data`	Pass	0.000s
Exceedance mode	`"70" in captured.out`	Pass	0.000s
Explicit gauge returns zero	`rc == 0`	Pass	0.002s
Flat flag dispatches flat mode	`rc == 0`	Pass	0.002s
Json default mode	`"summary_statistics" in data`	Pass	0.287s
List flag logs gauge names	`"THAMES-TEST-99" in combined`	Pass	0.002s
List flag returns zero	`rc == 0`	Pass	0.002s
main() creates audit directory if it doesn't exist.	`audit.exists()`; `out.exists()`	Pass	3.878s
Lines 120-122: default (no –nrfa-dir) calls generate_all_gauge_histories.	—	Pass	0.304s
main() exits cleanly if jscpd fails.	—	Pass	0.001s
Lines 107-108: without –catchment, config.CATCHMENT is untouched.	`config.CATCHMENT == original`	Pass	0.303s
main() prints file/clone/pct stats to stdout.	`'Clones' in out`; `'PDF saved' in out`	Pass	0.237s
main() prints step count and consistency status.	`'pipeline steps' in output`; `'Consistency' in output` (+1 more)	Pass	3.840s
main() writes data_lineage_report.pdf to audit dir.	`out.exists()`; `out.stat().st_size > 5000`	Pass	3.894s
main() writes code_duplication_report.pdf to audit dir.	`(tmp_path / 'code_duplication_report.pdf').exists()`; `(tmp_path / 'code_duplication_report.pdf').stat().st_size...`	Pass	0.244s

Test Description	Acceptance Criteria	Result	Time
Line 102: -c short flag for catchment.	—	Pass	0.296s
Line 101: -y short flag for years.	—	Pass	0.182s
main() logs errors for unreadable files and continues.	—	Pass	0.001s
Lines 107-108: –catchment sets config.CATCHMENT.	`config.CATCHMENT == "severn"`	Pass	0.310s
main() processes any *_gdf.csv files found in config.input_dir.	—	Pass	0.021s
main() handles an empty directory gracefully.	—	Pass	0.000s
Lines 117-119: –nrfa-dir calls process_nrfa_directory.	`list(gaugehd_dir.glob("gauge_*_hd.json"))`	Pass	0.002s
Line 101: –years flag is passed through.	—	Pass	0.207s
Missing gauge logs available gauges	`"THAMES-G042" in combined`	Pass	0.003s
Missing gauge logs error	`any("GHOST-GAUGE" in r.message for r in caplog.records)`	Pass	0.002s
Missing gauge returns one	`rc == 1`	Pass	0.002s
Return period json mode	`"return_period_years" in data; "intensity" in data`	Pass	0.000s
Return period mode	`"100" in captured.out`	Pass	0.000s
Samples json mode	`"samples" in data; len(data["samples"]) == 3`	Pass	0.000s
Samples mode	`"5 samples" in captured.out`	Pass	0.000s
Scenario mode	`len(captured.out) > 0`	Pass	0.074s
Seed reproducible	`out1 == out2`	Pass	0.001s
Trigger alert explicit	`rc == 0`	Pass	0.002s
Trigger severe explicit	`rc == 0`	Pass	0.002s
Trigger warning explicit	`rc == 0`	Pass	0.002s
Add to map calls add child	—	Pass	0.001s
Get js returns script	`"<script>" in js; "refreshMainMapFS01" in js`	Pass	0.000s
Get statistics returns dict	`isinstance(result, dict); result["component"] == "main_map_fs01"`	Pass	0.000s
Report with 2% duplication should render without error.	`analysis['total']['percentage'] < 3; len(pdf) > 0`	Pass	0.231s
Report with ¿8% duplication renders correctly.	`pdf[:4] == b'%PDF'`	Pass	0.228s
Pdf header	`pdf[:4] == b'%PDF'`	Pass	0.246s

Test Description	Acceptance Criteria	Result	Time
Pdf nonempty	`len(pdf) > 2000`	Pass	0.230s
Returns bytes	`isinstance(result, bytes)`	Pass	1.481s
Peak precipitation is in the storm's active window.	`peak_in_storm >= peak_outside`	Pass	0.000s
Drainage window (hours 156-167) must be precipitation-free.	`np.all(p[156:] == pytest.approx(0.0))`	Pass	0.001s
Non negative	`np.all(p >= 0.0)`	Pass	0.000s
Output shape	`p.shape == (EVENT_WINDOW_HOURS,)`	Pass	0.000s
Sum of hourly series == sum of storm precipitation_mm values.	`p.sum() == pytest.approx(expected, rel=1e-4)`	Pass	0.001s
Col widths propagated	`t._colWidths == cw`	Pass	0.000s
Custom alt color	`isinstance(t, Table)`	Pass	0.000s
When no alt_color passed, ALT_ROWS_BLUE is applied.	`isinstance(t, Table)`	Pass	0.000s
Empty rows	`isinstance(t, Table); len(t._cellvalues) == 1`	Pass	0.000s
Repeat rows set	`t.repeatRows == 1`	Pass	0.000s
Returns table	`isinstance(t, Table); isinstance(t, Table)`	Pass	0.000s
Table has correct row count	`len(t._cellvalues) == 4`	Pass	0.000s
Add action with malformed existing id	`r.status_code == 200; "A-002" in ids`	Pass	0.002s
Add decision with malformed existing id	`r.status_code == 200; "D-002" in ids`	Pass	0.002s
Add participant with malformed existing id	`r.status_code == 200; "P-002" in ids`	Pass	0.002s
Click-through should hide trading desk before opening gauge.	`'TradingDesk.hide' in js`	Pass	0.000s
Map should have New Trade link.	`'window.tdNewTrade' in js; 'New Trade' in js`	Pass	0.000s
Map should have tdViewHazardCurve navigating to gauge panel.	`'window.tdViewHazardCurve' in js; 'GaugeHazardCurve' in js`	Pass	0.000s
P&L bar must show 'From Market' metric.	`'From Market' in js; 'fromMarket' in js`	Pass	0.000s
Blotter must have a Mkt P&L column.	`"'market_pnl'" in js or '"market_pnl"' in js; 'Mkt P' in js`	Pass	0.000s
Market tab commit notification must show P&L impact.	`'total_pnl_impact' in js; 'gross_pnl_impact' in js`	Pass	0.000s
Hazard column must show up/down arrows when fair != prev_fair.	`'prev_fair_spread_bps' in js; '\{}\{}u25b2' in js` (+1 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Entry level under 250k	`isinstance(result, list)`	Pass	0.000s
Middle 250k to 500k	`isinstance(result, list)`	Pass	0.000s
Premium over 1m	`isinstance(result, list)`	Pass	0.000s
Upper middle 500k to 1m	`isinstance(result, list)`	Pass	0.000s
Base rates not empty	`br`	Pass	0.000s
File exists	`GAUGEHC_PATH.exists(); MARKET_STATE.exists()`	Pass	0.000s
Has required keys	`not missing`	Pass	0.000s
Hazard term structure has triggers	`"alert" in triggers`	Pass	0.000s
Hazard term structure not empty	`hts`	Pass	0.000s
Last updated present	`"last_updated" in ms`	Pass	0.000s
Yield curve has all tenors	`not missing`	Pass	0.000s
Yield curve rates in range	`not bad`	Pass	0.000s
computeMaturityDate function exists.	`'computeMaturityDate' in prs_js`	Pass	0.000s
currentRollDate function exists.	`'currentRollDate' in prs_js`	Pass	0.000s
lastFridayOfMonth function exists.	`'lastFridayOfMonth' in prs_js`	Pass	0.000s
showMaturityPopup function exists.	`'showMaturityPopup' in prs_js`	Pass	0.000s
Mcob advice rejection reason	`isinstance(result, list)`	Pass	0.000s
Mcob all fields returns list	`isinstance(result, list); len(result) > 5`	Pass	0.000s
Mcob section creates tables	`len(tables) >= 2`	Pass	0.001s
Mcob with minimal fields	`isinstance(result, list)`	Pass	0.000s
Serve missing document returns 404	`r.status_code == 404`	Pass	0.002s
Upload empty filename returns 400	`r.status_code == 400`	Pass	0.003s
Upload no file returns 400	`r.status_code == 400; r.get_json()["status"] == "error"`	Pass	0.002s
Upload not found meeting returns 404	`r.status_code == 404`	Pass	0.001s
Unknown menu no options returns empty	`result == ""`	Pass	0.000s
Unknown menu with options cycles	`result == "OptionB"`	Pass	0.000s
Unknown menu with options wraps	`result == "X"`	Pass	0.000s
Duration from storms json	`storm["duration_hours"] == 72; storm["peak_position"] == pytest.approx(0.3)` (+1 more)	Pass	0.003s
Fallback to config defaults	`storm["duration_hours"] == STRESS_STORM_DEFAULT_DURATION_...; storm["peak_position"] == pytest.approx(STRESS_STORM_DEFA...`	Pass	0.004s

Test Description	Acceptance Criteria	Result	Time
Passing a non-existent storms_json_path must not crash.	`out.exists() and (out / "_index.json").exists()`	Pass	0.002s
Distant point returns larger distance	`d_far > d_near`	Pass	0.000s
Point on gauge returns small distance	`d < 10_000`	Pass	0.000s
Returns float	`isinstance(d, float)`; `d >= 0`	Pass	0.000s
Add minute item	`r.status_code == 200`; `isinstance(minutes, list)` (+1 more)	Pass	0.002s
Delete minute item legacy format returns 400	`r.status_code == 400`	Pass	0.003s
Update minute item legacy format returns 400	`r.status_code == 400`	Pass	0.003s
Update minute item not found returns 404	`r.status_code == 404`	Pass	0.003s
Update minutes text	`r.status_code == 200`; `r.get_json()["meeting"]["minutes"] == "# Minutes\{}n\{}nAll a...`	Pass	0.002s
Update minutes unknown meeting returns 404	`r.status_code == 404`	Pass	0.001s
Add minute item missing meeting returns 404	`r.status_code == 404`	Pass	0.002s
Add minute item returns 200	`r.status_code == 200`; `isinstance(meeting["minutes"], list)` (+1 more)	Pass	0.002s
When minutes is a string, deleting returns 400.	`r.status_code == 400`	Pass	0.003s
Delete minute item missing meeting returns 404	`r.status_code == 404`	Pass	0.002s
Delete minute item returns 200	`r.status_code == 200`; `len(meeting["minutes"]) == 1` (+1 more)	Pass	0.004s
When minutes is a string (legacy), updating an item returns 400.	`r.status_code == 400`	Pass	0.003s
Update minute item not found returns 404	`r.status_code == 404`	Pass	0.002s
Update minute item returns 200	`r.status_code == 200`; `item["text"] == "Updated text."`	Pass	0.002s
Add minute item when minutes is string	`r.status_code == 200`; `isinstance(meeting["minutes"], list)` (+2 more)	Pass	0.003s

Test Description	Acceptance Criteria	Result	Time
Update minutes missing meeting returns 404	`r.status_code == 404`	Pass	0.002s
Update minutes returns 200	`r.status_code == 200`; `r.get_json()["meeting"]["minutes"] == "New minutes text"`	Pass	0.002s
Property hazard returns 404 when propertyhc.json missing.	`response.status_code == 404`; `data['status'] == 'error'`	Pass	0.000s
PropertyHC summary returns 404 with helpful message when file missing.	`response.status_code == 404`; `data['status'] == 'error'` (+1 more)	Pass	0.000s
PropertyTS summary returns 404 when directory missing.	`response.status_code == 404`; `data['status'] == 'error'`	Pass	0.000s
Alert storms collected severe counted	`result["summary"]["floods_at_nearest_gauge"] == 3`; `result["summary"]["severe_at_nearest_gauge"] == 2`	Pass	0.164s
Sub alert storms not in flood events	`"STORM-m001b" not in event_storm_ids`; `"STORM-m001a" in event_storm_ids` (+2 more)	Pass	0.167s
Has kv table	`callable(gen._kv_table)`	Pass	0.000s
Has load all	`callable(gen._load_all)`	Pass	0.000s
Has make table	`callable(gen._make_table)`	Pass	0.000s
Has section methods	`hasattr(gen, name)`	Pass	0.000s
_load_all() output can be consumed by all section methods.	—	Pass	0.013s
model_auc should be None since predictor is removed.	`resp.status_code == 200`; `data['model_auc'] is None`	Pass	0.033s
All models have model id	`"model_id" in m`; `isinstance(m["model_id"], str)` (+1 more)	Pass	0.000s
Model ids are unique	`len(ids) == len(set(ids))`	Pass	0.000s
Model ids follow naming convention	`mid.startswith("MKM-")`	Pass	0.000s
Create mapping returns flat dict	`isinstance(mapping, dict)`; `mapping.get("gauge_id") == "GAUGE-test1234"` (+7 more)	Pass	0.000s
Required fields include mortgage id	`any("MortgageID" in f for f in mortgage_cdm.get_required_...`	Pass	0.000s
Schema has mortgage root	`len(list(mortgage_cdm.schema.keys())) > 0`	Pass	0.000s
Validate valid mortgage	`isinstance(mortgage_cdm.validate(data), dict)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Lines 362-363: exception in create_mapping raises ValueError.	—	Pass	0.000s
Line 280: valid data returns empty dict.	`mortgage_cdm.validate(data) == {}`	Pass	0.000s
Lines 282-283: exception in validate returns validation_error.	`"validation_error" in result`; `isinstance(result["validation_error"], list)` (+1 more)	Pass	0.000s
Line 268: header_errors list appended to errors dict when non-empty.	`"Header" in errors`; `len(errors["Header"]) >= 2`	Pass	0.000s
Line 262: Missing CatchmentID produces header error.	`"Header" in errors`; `any("CatchmentID" in e for e in errors["Header"])` (+2 more)	Pass	0.000s
Line 259: Missing MortgageID produces header error.	`"Header" in errors`; `any("MortgageID" in e for e in errors["Header"])`	Pass	0.000s
Line 275: Missing OriginalLoan produces FinancialTerms error.	`"FinancialTerms" in errors`; `any("OriginalLoan" in e for e in errors["FinancialTerms"])`	Pass	0.000s
Line 265: Missing PropertyID produces header error.	`"Header" in errors`; `any("PropertyID" in e for e in errors["Header"])`	Pass	0.000s
Line 278: terms_errors appended when non-empty.	`"FinancialTerms" in errors`	Pass	0.000s
Cost analysis subsection	`any("Cost Analysis" in t for t in texts)`	Pass	0.000s
Net upfront ¿ 5% of loan → High assessment.	`isinstance(result, list)`	Pass	0.000s
Net upfront ¡ 0.5% → Low assessment.	`isinstance(result, list)`	Pass	0.000s
Net upfront 0.5–1.5% → Low-Medium assessment.	`isinstance(result, list)`	Pass	0.000s
Net upfront 1.5–3% → Medium assessment.	`isinstance(result, list)`	Pass	0.000s
Net upfront 3–5% → Medium-High assessment.	`isinstance(result, list)`	Pass	0.000s
Discount points + original rate triggers estimated points/rate reduction rows.	`isinstance(result, list)`	Pass	0.000s
Empty mortgage dict does not crash	`isinstance(result, list)`	Pass	0.000s
Has tables	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
Lender credits reduces net cost	`isinstance(result, list)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
No financial terms returns message	`any("No cost information available" in t for t in texts)`	Pass	0.000s
No mortgage data returns message	`any("No mortgage data available" in t for t in texts)`	Pass	0.000s
Origination setup subsection	`any("Origination" in t for t in texts)`	Pass	0.000s
Penalties subsection	`any("Penalties" in t for t in texts)`	Pass	0.000s
Returns list with data	`isinstance(result, list); len(result) > 0`	Pass	0.000s
Section header present	`any("Mortgage Costs" in t for t in texts)`	Pass	0.000s
TotalLoanCosts + OriginalLoan triggers percentage row.	`isinstance(result, list)`	Pass	0.000s
Total costs overview subsection	`any("Total Loan Costs" in t for t in texts)`	Pass	0.000s
LoanToValueRatio shown when different from OriginalLTV.	`isinstance(result, list)`	Pass	0.000s
Core loan terms subsection	`any("Core Loan Terms" in t for t in texts)`	Pass	0.000s
Dti high risk	`isinstance(result, list)`	Pass	0.000s
Dti low risk	`isinstance(result, list)`	Pass	0.000s
Dti medium high risk	`isinstance(result, list)`	Pass	0.000s
Dti medium risk	`isinstance(result, list)`	Pass	0.000s
Dti subsection	`any("Debt-to-Income" in t for t in texts)`	Pass	0.000s
Has tables	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
Interest rates subsection	`any("Interest Rates" in t for t in texts)`	Pass	0.000s
Loan as % of purchase value row is computed when both fields present.	`isinstance(result, list)`	Pass	0.000s
Ltv high risk	`isinstance(result, list)`	Pass	0.000s
Ltv low medium risk	`isinstance(result, list)`	Pass	0.000s
Ltv low risk	`isinstance(result, list)`	Pass	0.000s
Ltv medium risk	`isinstance(result, list)`	Pass	0.000s
Ltv subsection	`any("Loan-to-Value" in t for t in texts)`	Pass	0.000s
Ltv very high risk	`isinstance(result, list)`	Pass	0.000s
No financial terms returns message	`any("No detailed financial terms available" in t for t in...`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
No mortgage data returns message	`any("No mortgage data available" in t for t in texts)`	Pass	0.000s
Returns list with data	`isinstance(result, list); len(result) > 0`	Pass	0.000s
Section header present	`any("Detailed Mortgage Financial Terms" in t for t in texts)`	Pass	0.000s
Init no output dir uses config	`gen.output_dir == reports_dir; reports_dir.exists()`	Pass	0.001s
Generate filename non dict raises attribute error	`"unknown" in fname`	Pass	0.001s
Generate filename none mortgage data	`"unknown" in fname; fname.endswith(".pdf")`	Pass	0.001s
Page exception is caught	`isinstance(elements, list); len(elements) > 0`	Pass	0.002s
Unknown page in order is skipped	`isinstance(elements, list); len(elements) > 0`	Pass	0.002s
Generator does not use sequential format	`'GAUGE-{i:' not in src; 'GAUGE-{index' not in src` (+3 more)	Pass	0.000s
Generator source uses mort prefix	`'MORT-' in src`	Pass	0.001s
Generator source uses uuid	`'uuid' in src; 'uuid' in src`	Pass	0.000s
Uuid4 produces protocol matching ids	`GAUGE_ID_RE.match(gauge_id); PROP_ID_RE.match(prop_id)` (+2 more)	Pass	0.000s
Accepts valid	`PROP_ID_RE.match('PROP-a3f7bc12'); PROP_ID_RE.match('PROP-00000000')` (+6 more)	Pass	0.000s
Rejects sequential	`not PROP_ID_RE.match('PROP-001'); not PROP_ID_RE.match('PROP-1234')` (+3 more)	Pass	0.000s
Test getting delinquent mortgages.	`len(delinquent) == 1; delinquent[0]['MortgageID'] == "MORT-003"`	Pass	0.001s
Test exposure aggregation by lender.	`"First National Bank" in exposure; "Regional Credit Union" in exposure` (+2 more)	Pass	0.002s
Test getting property IDs that have mortgages.	`len(prop_ids) == 3; "PROP-001" in prop_ids` (+2 more)	Pass	0.001s
Test total loan exposure calculation.	`total == 1640000`	Pass	0.001s
List all returns summaries	`len(MortgageLoader(tmp_path).list_all()) == 2`	Pass	0.007s
Load all returns list	`len(MortgageLoader(tmp_path).load_all()) == 3`	Pass	0.007s

Test Description	Acceptance Criteria	Result	Time
Test MortgageID extraction.	`mort_id is not None; mort_id.startswith("MORT-")`	Pass	0.001s
Test mortgage summary generation.	`len(summaries) == 3; 'mortgageId' in summary` (+3 more)	Pass	0.001s
Test loading mortgage portfolio.	`len(mortgages) == 3`	Pass	0.002s
Find by id	`MortgageLoader(tmp_path).find_by_id("MORT-001") is not None`	Pass	0.001s
Mortgage without top-level PropertyID falls back to nested Mortgage.Header.PropertyID.	`MortgageLoader(tmp_path).find_by_property_id("PROP-NESTED")`	Pass	0.001s
Missing returns none	`MortgageLoader(tmp_path).find_by_id("MORT-999") is None`	Pass	0.001s
Test finding mortgage by ID.	`mort is not None; mort['PropertyID'] == "PROP-001"` (+1 more)	Pass	0.001s
Test finding mortgages by lender.	`len(morts) == 2`	Pass	0.002s
Test lender search is case insensitive.	`len(morts) == 2`	Pass	0.002s
Test finding mortgage by property ID.	`mort is not None; mort['MortgageID'] == "MORT-001"`	Pass	0.005s
Test finding mortgage for property without mortgage.	`mort is None`	Pass	0.001s
Test finding active mortgages.	`len(morts) == 2`	Pass	0.002s
Test finding delinquent mortgages.	`len(morts) == 1; morts[0]['MortgageID'] == "MORT-003"`	Pass	0.001s
Application subsection	`any("Application Information" in t for t in texts)`	Pass	0.001s
Denial reason hidden when default	`isinstance(result, list)`	Pass	0.000s
Denial reason shown when not default	`isinstance(result, list)`	Pass	0.000s
Empty mortgage dict does not crash	`isinstance(result, list)`	Pass	0.000s
mortgage_data without outer 'Mortgage' key is handled via fallback.	`isinstance(result, list)`	Pass	0.000s
Has tables	`any(isinstance(e, Table) for e in result)`	Pass	0.001s
Key financial terms subsection	`any("Key Financial Terms" in t for t in texts)`	Pass	0.000s
LTV row is computed when both PurchaseValue and OriginalLoan are present.	`isinstance(result, list)`	Pass	0.000s
Mortgage features subsection	`any("Mortgage Features" in t for t in texts)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Mortgage identification subsection	`any("Mortgage Identification" in t for t in texts)`	Pass	0.000s
No mortgage data returns message	`any("No mortgage data available" in t for t in texts)`	Pass	0.000s
None mortgage data returns message	`any("No mortgage data available" in t for t in texts)`	Pass	0.000s
Returns list with data	`isinstance(result, list); len(result) > 0`	Pass	0.000s
Section header present	`any("Mortgage Overview" in t for t in texts)`	Pass	0.000s
Test that mortgage PropertyID links to valid property.	`prop is not None`	Pass	0.003s
Test finding mortgage for each property.	`prop is not None; mort is not None`	Pass	0.003s
Generate elements returns list	`isinstance(elements, list); len(elements) > 0`	Pass	0.002s
Generate filename fallback	`fname.endswith(".pdf"); "unknown" in fname`	Pass	0.001s
Generate filename with id	`"MORT-test0001" in fname; fname.endswith(".pdf")`	Pass	0.001s
Generate report creates pdf	`output.exists(); output.suffix == ".pdf"`	Pass	0.011s
Generate report returns path	`isinstance(result, Path)`	Pass	0.011s
Init creates output dir	`output_dir.exists()`	Pass	0.001s
Page order set	`"title" in gen.page_order; gen.page_order[0] == "title"`	Pass	0.001s
Pages initialised	`"title" in gen.pages; "mortgage_overview" in gen.pages`	Pass	0.001s
Elevated ltv medium flood	`"Moderate Risk" in r`	Pass	0.000s
High flood overrides	`"High Risk" in r and "flood" in r.lower()`	Pass	0.000s
High ltv medium flood	`"High Risk" in r`	Pass	0.000s
Low flood risk	`"Low Risk" in r`	Pass	0.000s
Medium flood default	`"Moderate Risk" in r`	Pass	0.000s
Minimal risk	`"Minimal" in r`	Pass	0.000s
Negative value critical	`"Critical Risk" in r`	Pass	0.000s
Negative value high	`"High Risk" in r`	Pass	0.000s
Negative value moderate	`"Moderate Risk" in r`	Pass	0.000s
None ltv no crash	`isinstance(r, str)`	Pass	0.000s
Very high flood overrides	`"High Risk" in r`	Pass	0.000s
Elements non empty	`len(elements) > 0`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Empty property data does not crash	`isinstance(elements, list)`	Pass	0.000s
Extracts mortgage id	`"MORT-test0001" in full_text`	Pass	0.000s
No mortgage data returns elements	`isinstance(elements, list)`; `len(elements) > 0`	Pass	0.000s
Returns list	`isinstance(elements, list)`	Pass	0.012s
Table elements included	`len(tables) >= 1`	Pass	0.000s
Line 108: _save_meetings fails → 500.	`r.status_code == 500`; `"Failed" in r.get_json()["message"]`	Pass	0.001s
Line 195: document file not found → 404.	`r.status_code == 404`	Pass	0.002s
Line 196: serve an uploaded document.	`r.status_code == 200`	Pass	0.004s
Line 162: file with empty filename → 400.	`r.status_code == 400`	Pass	0.003s
Line 155: meeting not found → 404.	`r.status_code == 404`	Pass	0.001s
Line 158: no file in request → 400.	`r.status_code == 400`	Pass	0.003s
Line 184: save failure after upload → 500.	`r.status_code == 500`	Pass	0.003s
Lines 164-186: upload a file to a meeting.	`r.status_code == 200`; `doc["filename"] == "report.pdf"`	Pass	0.004s
Line 179: meeting without 'documents' key gets it created.	`r.status_code == 200`	Pass	0.005s
Line 144: _save_meetings fails on update → 500.	`r.status_code == 500`	Pass	0.003s
Count gauges exceeding warning level.	`count >= 0`; `count <= len(sample_gauges_df)`	Pass	0.002s
Find gauge with highest current level.	`highest is not None`; `'gauge_id' in highest` (+1 more)	Pass	0.003s
First storm gets greek name	`name`; `name[0].isupper()`	Pass	0.002s
Name from storms json takes priority	`data["storms"][0]["name"] == "CustomStormName"`	Pass	0.002s
Gauge nav menu must use the same actions as the right-click menu.	`item['action'] in js`	Pass	0.000s
Menu items must be serialized from Python lists, not hardcoded in JS.	`gauge_json in js`; `prop_json in js`	Pass	0.000s
Property nav menu must use the same actions as the right-click menu.	`item['action'] in js`	Pass	0.000s
Loader should return both real and synthetic gauges.	`len(synth) > 0`; `len(real) > 0`	Pass	0.003s

Test Description	Acceptance Criteria	Result	Time
GaugeLoader.get_entity_summary must produce 'gaugeId' key.	len(summaries) > 0; 'gaugeId' in sample (+1 more)	Pass	0.002s
Nav menu JS must use 'gaugeId' (API contract), not 'gauge_id'.	'g.gaugeId' in js or "g['gaugeId']" in js or 'gaugeId' in js; 'g.gauge_id' not in js	Pass	0.000s
Nav menu JS must use 'name' for gauge display name.	'.name' in js or "['name']" in js	Pass	0.000s
Nav JS must access _prePropertyTS.data.properties for property list.	'_prePropertyTS' in js; '.properties' in js	Pass	0.000s
Nav menu JS must use 'property_id' for property IDs.	'property_id' in js	Pass	0.000s
Portfolio flood summary properties must have 'property_id'.	len(properties) > 0; 'property_id' in properties[0]	Pass	0.002s
Filter must check for 'SYNTH' prefix on gaugeId.	'SYNTH' in js	Pass	0.000s
Nav menus included in get js	'nav-menu-container' in js	Pass	0.001s
Nav menus js contains gauge select	'nav-menu-container' in js; 'nav-gauge-select' in js (+1 more)	Pass	0.000s
Nav menus js contains property select	'nav-prop-select' in js; '_prePropertyTS' in js	Pass	0.000s
Nav menus js filters synthetic gauges	'SYNTH' in js	Pass	0.000s
Nav menus js has action bar	'nav-action-bar' in js	Pass	0.000s
Nav menus js includes menu actions	item['action'] in js; item['action'] in js	Pass	0.000s
When segment is a point (seg2 ¡ 1e-18), returns start point distance.	abs(nx - 51.5) < 1e-9; abs(ny + 0.1) < 1e-9 (+1 more)	Pass	0.000s
Distance is non negative	d >= 0	Pass	0.000s
Midpoint projects to midpoint	abs(nx - 51.5) < 0.01; abs(ny - 0.0) < 0.01 (+1 more)	Pass	0.000s
Point before start clamps to start	abs(nx - 51.0) < 0.01; abs(ny - 0.0) < 0.01	Pass	0.000s
Point past end clamps to end	abs(nx - 52.0) < 0.01; abs(ny - 0.0) < 0.01	Pass	0.000s
Damage amount computed correctly	prop["damage_amount"] == pytest.approx(expected)	Pass	0.004s
Negative equity flag set on property	data["properties"][0]["negative_equity"] is True	Pass	0.004s
LTV ¿ 100% when outstanding ¿ post-damage value.	prop["post_damage_ltv"] > 100.0	Pass	0.004s

Test Description	Acceptance Criteria	Result	Time
Severe storm triggers negative equity	`r.status_code == 200; data["portfolio"]["mortgages_in_negative_equity"] == 1`	Pass	0.004s
Same for mortgage-report: null JSON body -¿ line 168 -¿ 400.	`r.status_code in (400, 500)`	Pass	0.000s
Sending content-type application/json with null body -¿ get_json() = None -¿ 400.	`r.status_code in (400, 500)`	Pass	0.000s
All modules no newline in js strings	`failures == {}`	Pass	0.004s
Audit reports no newline in js strings	`offenders == []`	Pass	0.001s
Full compiled js no newline in js strings	`offenders == []`	Pass	0.004s
No mortgage message	`any("No mortgage data" in t for t in texts)`	Pass	0.000s
No mortgage returns list	`isinstance(result, list); len(result) > 0`	Pass	0.000s
Gauge that exists in storm data but has no trades returns 404.	`resp.status_code == 404; 'No open trades' in data.get('message', '')`	Pass	0.032s
Empty regulatory returns message	`any("No regulatory information" in t for t in texts)`	Pass	0.000s
Returns list	`isinstance(result, list)`	Pass	0.000s
Line 95: _normal_pdf(0) should return 1/sqrt(2*pi).	`result == pytest.approx(expected, rel=1e-10)`	Pass	0.000s
Line 95: PDF decreases as —x— increases.	`d._normal_pdf(0) > d._normal_pdf(1) > d._normal_pdf(2)`	Pass	0.000s
Line 95: _normal_pdf at positive value.	`result == pytest.approx(expected, rel=1e-10)`	Pass	0.000s
Line 95: _normal_pdf is symmetric.	`d._normal_pdf(2.0) == pytest.approx(d._normal_pdf(-2.0))`	Pass	0.000s
Normalize float returns empty	`loader._normalize_to_list(3.14) == []`	Pass	0.000s
Normalize integer returns empty	`result == []`	Pass	0.001s
Normalize string returns empty	`result == []`	Pass	0.000s
Normalize true returns empty	`loader._normalize_to_list(True) == []`	Pass	0.000s
–stressm alone must always run stressm regardless of –nostress.	—	Pass	0.544s
–nostress must be a registered argument on the port subparser.	`args.nostress is True`	Pass	0.005s
With –all –nostress, generate_stressm must not be called.	—	Pass	0.091s

Test Description	Acceptance Criteria	Result	Time
Number type returns float	`isinstance(result, float); 0 <= result <= 1000`	Pass	0.000s
CalledProcessError → returns False.	`result is False`	Pass	0.000s
macOS: subprocess.run(['open', ...]) is called.	`result is True; mock_run.call_args[0][0][0] == "open"`	Pass	0.001s
Darwin success	`result is True; mock_run.call_args[0][0][0] == "open"`	Pass	0.000s
Darwin success	`result is True`	Pass	0.000s
Generic exception returns false	`result is False`	Pass	0.000s
Generic exception returns false	`result is False`	Pass	0.000s
Any other exception → returns False.	`result is False`	Pass	0.001s
Linux: subprocess.run(['xdg-open', ...]) is called.	`result is True; mock_run.call_args[0][0][0] == "xdg-open"`	Pass	0.000s
Linux success	`result is True; "xdg-open" in args`	Pass	0.000s
Linux success	`result is True`	Pass	0.000s
Subprocess called process error returns false	`result is False`	Pass	0.000s
Subprocess failure returns false	`result is False`	Pass	0.000s
Unknown platform uses webbrowser	`result is True`	Pass	0.000s
Unknown platform uses webbrowser	`result is True`	Pass	0.001s
Unknown platform: falls back to webbrowser.open().	`result is True`	Pass	0.000s
Windows: subprocess.run(['cmd', '/c', 'start', ...]) is called.	`result is True; mock_run.call_args[0][0][:3] == ["cmd", "/c", "start"]`	Pass	0.000s
Windows case insensitive	`isinstance(result, bool)`	Pass	0.001s
Windows success	`result is True`	Pass	0.000s
Windows success	`result is True`	Pass	0.000s
Lines 432-433: –gauge-id → _find_gauge_by_id called.	—	Pass	0.022s
Lines 484-486: –mortgage-file → mortgage JSON is opened and parsed.	—	Pass	0.006s
Lines 489-490: –property-id with no mortgage file → _find_property_by_id called.	—	Pass	0.005s

Test Description	Acceptance Criteria	Result	Time
Lines 491-492: both –property-id and –mortgage-file → both finders called.	—	Pass	0.006s
Lines 427-429: –timeseries-file → timeseries JSON opened and parsed.	—	Pass	0.007s
Known outputs	`m["gauge.json"] == "synthetic_gauges"`; `m["property.json"] == "properties"` (+1 more)	Pass	0.000s
Dir output empty	`not _outputs_exist("stressm", tmp_path)`	Pass	0.000s
Dir output missing	`not _outputs_exist("stressm", tmp_path)`	Pass	0.000s
Dir output present non empty	`_outputs_exist("stressm", tmp_path)`	Pass	0.001s
File output missing	`not _outputs_exist("gauges", tmp_path)`	Pass	0.000s
File output present	`_outputs_exist("gauges", tmp_path)`	Pass	0.000s
Partial outputs missing	`not _outputs_exist("stressm", tmp_path)`	Pass	0.000s
Unknown step	`not _outputs_exist("nonexistent", tmp_path)`	Pass	0.000s
Very high flood risk + arrears + very high LTV + poor credit -¿ CRITICAL.	`isinstance(result, list)`	Pass	0.001s
Very low flood risk + good payment + low LTV + excellent credit -¿ LOW.	`isinstance(result, list)`	Pass	0.000s
Property only high flood risk	`isinstance(result, list)`	Pass	0.000s
Property only very low flood risk	`isinstance(result, list)`	Pass	0.000s
Effective rating is valid	`rating in valid_ratings`	Pass	0.000s
Overall risk rating has component scores	`isinstance(cs, dict)`	Pass	0.000s
Overall risk rating has required keys	`key in orr`	Pass	0.000s
Clear override with null	`r.status_code == 200`; `r.get_json()["status"] == "success"`	Pass	0.005s
Invalid rating returns 400	`r.status_code == 400`; `r.get_json()["status"] == "error"`	Pass	0.002s
Model not found returns 404	`r.status_code == 404`; `r.status_code == 404`	Pass	0.002s
No reason field returns 400	`r.status_code == 400`	Pass	0.002s
No reason returns 400	`r.status_code == 400`	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Override stored in model	`rr["mrc_override"] == "Conditional"`; `rr["effective_rating"] == "Conditional"`	Pass	0.004s
Response has audit entry	`"audit_entry" in data`; `data["audit_entry"]["event_type"] == "risk_rating_override"`	Pass	0.003s
Valid override returns success	`r.status_code == 200`; `r.get_json()["status"] == "success"`	Pass	0.003s
Line 301: audit overflow during override.	`r.status_code == 200`	Pass	0.036s
Line 248: inventory not found → 404.	`r.status_code == 404`	Pass	0.001s
Line 282: save failure → 500.	`r.status_code == 500`	Pass	0.002s
All hourly P(flood) values should be in [0, 1] (None after knock-out).	`resp.status_code == 200`; `0.0 <= h['p_flood'] <= 1.0`	Pass	0.032s
Once water exceeds severe, P(flood) = 1.0 until knock-out blanks it.	`resp.status_code == 200`; `found_one` (+1 more)	Pass	0.029s
P(flood) for non-severe gauges should be w/s ratio.	`resp.status_code == 200`; `0.0 <= g['p_flood'] <= 1.0`	Pass	0.033s
Custom gauge basket size	`cdm.gauge_basket_size == 10`	Pass	0.000s
Default gauge basket size	`cdm.gauge_basket_size == 20`	Pass	0.000s
Lines 333-334: exception in create_mapping raises ValueError.	—	Pass	0.000s
Verify all sections of create_mapping produce correct keys.	`mapping["swap_id"] == "PRS-1"`; `mapping["notional"] == 1000000` (+3 more)	Pass	0.000s
validate({}) returns errors.	`isinstance(errors, dict)`	Pass	0.000s
Lines 255-256: Empty GaugeBasket.	`"GaugeSet" in errors`; `any("GaugeBasket" in e for e in errors["GaugeSet"])`	Pass	0.000s
Lines 263-264: exception returns validation_error.	`"validation_error" in result`; `isinstance(result["validation_error"], list)` (+1 more)	Pass	0.000s
Lines 258-259: gauge errors collected.	`"GaugeSet" in errors`	Pass	0.000s
Lines 244-245: header errors collected.	`"Header" in errors`; `len(errors["Header"]) >= 2`	Pass	0.000s
Line 241-242: Missing CatchmentID.	`"Header" in errors`; `any("CatchmentID" in e for e in errors["Header"])` (+2 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Line 251-252: Missing GaugeSetID.	`"GaugeSet" in errors; any("GaugeSetID" in e for e in errors["GaugeSet"])`	Pass	0.000s
Line 238-239: Missing SwapID.	`"Header" in errors; any("SwapID" in e for e in errors["Header"])`	Pass	0.000s
Line 261: valid data returns empty dict.	`prs_cdm.validate(data) == {}`	Pass	0.000s
buildPRSControls function exists.	`'buildPRSControls' in prs_js`	Pass	0.000s
Counterparty selector present.	`'prs-counterparty' in prs_js`	Pass	0.000s
Payer/Receiver direction selector present.	`'payer' in prs_js.lower(); 'receiver' in prs_js.lower()` (+1 more)	Pass	0.000s
Maturity/tenor selector present.	`'prs-maturity' in prs_js`	Pass	0.000s
Notional input field present.	`'prs-notional' in prs_js`	Pass	0.000s
Trigger level selector present.	`'prs-trigger' in prs_js`	Pass	0.000s
ghc_prs.get_js() returns valid JS.	`len(prs_js) > 1000`	Pass	0.000s
Fetches yield curve from /api/v1/trading/yield-curve.	`'/api/v1/trading/yield-curve' in prs_js`	Pass	0.000s
Each trade has physical swap root	`not bad`	Pass	0.010s
Each trade has swap id	`not bad`	Pass	0.005s
Prs directory exists	`PRS_DIR.exists()`	Pass	0.000s
Swap id matches filename	`not bad`	Pass	0.005s
Thames Central book has 16 trades.	`len(prs_files) >= 16`	Pass	0.000s
Trade statuses are valid	`not bad`	Pass	0.005s
Trades are present	`len(prs_files) >= 1`	Pass	0.000s
Trades have gauge set	`not bad`	Pass	0.005s
Book thames uses prs prefix	`'PRS-' in src`	Pass	0.001s
Book thames uses uuid	`'uuid' in src`	Pass	0.000s
uuid4().hex[:8].upper() — the book generator pattern — always matches.	`PRS_ID_RE.match(swap_id)`	Pass	0.000s
Some paths may produce lowercase — also accepted.	`PRS_ID_RE.match('PRS-a8b3c4d5'); PRS_ID_RE.match('PRS-ffffffff')`	Pass	0.000s
book.py uses uuid4().hex[:8].upper() → uppercase hex.	`PRS_ID_RE.match('PRS-A8B3C4D5'); PRS_ID_RE.match('PRS-FFFFFFFF')` (+1 more)	Pass	0.000s
PRS-TEST-001 style IDs used in test fixtures are NOT production format.	`not PRS_ID_RE.match('PRS-TEST-001'); not PRS_ID_RE.match('PRS-VAUXHALL')`	Pass	0.000s
Port src all does not include book symbols	`not overlap`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
The package __init__.py must not contain ‘from .book import‘.	`"from .book import" not in src`	Pass	0.001s
Package __init__.py must not expose generate_thames_central_book.	`not hasattr(_pkg, "generate_thames_central_book")`	Pass	0.000s
Contains claim ref paragraph	`len(tables) > 0`	Pass	0.001s
No flood events still renders	`len(result) > 0`	Pass	0.001s
Coordinates that are not floats should not raise.	`isinstance(result, list)`	Pass	0.001s
Returns list	`isinstance(result, list); len(result) > 0` (+4 more)	Pass	0.001s
Stats reflect events	`any(isinstance(e, Paragraph) for e in result)`	Pass	0.001s
Without mortgage	`isinstance(result, list); len(result) > 0`	Pass	0.001s
Events with no sequence_id (None) should render without error.	`isinstance(result, list)`	Pass	0.001s
No flood events	`isinstance(result, list); len(result) > 0` (+2 more)	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0` (+4 more)	Pass	0.001s
Stats bar present	`len(paragraphs) > 0`	Pass	0.001s
Table included	`any(isinstance(e, Table) for e in result)`	Pass	0.001s
Events with sequence_id not in lookup default to ’isolated’.	`isinstance(result, list)`	Pass	0.001s
Contains tables	`len(tables) >= 2`	Pass	0.001s
Events with damage ratio ¿ 5% trigger red background rows.	`isinstance(result, list)`	Pass	0.001s
Multiple sequences grouped	`len(tables) >= 3`	Pass	0.001s
No flood events	`isinstance(result, list); len(result) > 0` (+2 more)	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0` (+4 more)	Pass	0.002s
Zero property value should not raise a ZeroDivisionError.	`isinstance(result, list)`	Pass	0.001s
Contains tables	`any(isinstance(e, Table) for e in result)`	Pass	0.001s
High damage + large mortgage triggers negative equity highlighting.	`isinstance(result, list); any(isinstance(e, Table) for e in result)`	Pass	0.001s
No flood events with mortgage	`isinstance(result, list)`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
With mortgage	`isinstance(result, list); len(result) > 0`	Pass	0.001s
Without mortgage	`isinstance(result, list); any('No mortgage' in t for t in texts)` (+2 more)	Pass	0.000s
Contains reference table	`len(tables) >= 2`	Pass	0.001s
Assessed damage = prop_value * max damage_ratio per sequence.	`isinstance(result, list)`	Pass	0.001s
When damage exceeds property value, LTV is very high.	`isinstance(result, list)`	Pass	0.001s
Returns list	`isinstance(result, list); len(result) > 0`	Pass	0.001s
Signature block present	`len(paragraphs) > 3`	Pass	0.001s
Without mortgage	`isinstance(result, list); any('No mortgage' in t for t in texts)` (+2 more)	Pass	0.001s
sp-panel-title element ID must be referenced in the JS.	`'sp-panel-title' in table_js`	Pass	0.000s
Title must start with 'Portfolio Storm Impact'.	`'Portfolio Storm Impact' in table_js`	Pass	0.000s
Panel title must use options[selectedIndex].textContent.	`'selectedIndex' in table_js; 'textContent' in table_js`	Pass	0.000s
Title assignment must NOT directly concatenate bare stormId.	`"Portfolio Storm Impact \{}\{}u2014 ' + stormId" not in table_js`	Pass	0.000s
Fat tail higher exceedance	`p_fat > p_thin`	Pass	0.000s
Pareto above threshold between 0 and 1	`0 < p < 1`	Pass	0.000s
Pareto at threshold is one	`d._pareto_exceedance(d.params.tail_threshold) == 1.0`	Pass	0.000s
Pareto below threshold is one	`d._pareto_exceedance(d.params.tail_threshold - 5) == 1.0`	Pass	0.000s
Empty file	`metadata == {}; flows == []`	Pass	0.000s
Flow values are floats	`isinstance(f["flow_cumecs"], float)`	Pass	0.006s
Flows have date and flow	`len(flows) == 5; "date" in f` (+1 more)	Pass	0.001s
Flows have date and flow cumecs	`len(flows) > 0; "date" in flows[0]` (+1 more)	Pass	0.007s
Invalid rows skipped	`len(flows) == 2`	Pass	0.002s
Metadata contains station id	`"station_id" in metadata; metadata["station_id"] == "39001"`	Pass	0.001s
Metadata contains station name	`metadata.get("station_name") == "My Test Gauge"`	Pass	0.006s

Test Description	Acceptance Criteria	Result	Time
Metadata contains station name	`metadata.get("station_name") == "Test River"`	Pass	0.001s
Metadata station id	`metadata.get("station_id") == "40007"`	Pass	0.006s
Returns metadata and flows	`isinstance(metadata, dict); isinstance(flows, list)`	Pass	0.006s
Returns tuple	`isinstance(metadata, dict); isinstance(flows, list)`	Pass	0.002s
Short row skipped	`len(flows) == 2`	Pass	0.003s
Lines with fewer than 2 columns are skipped.	`len(flows) == 1`	Pass	0.001s
All metadata categories parsed	`metadata.get("file_timestamp") == "2024-01-01"; metadata.get("database_id") == "7"` (+4 more)	Pass	0.002s
Flow row with non numeric value skipped	`len(flows) == 2`	Pass	0.001s
Metadata row with no value uses empty string	`metadata.get("station_name") == ""; metadata.get("station_id") == "39001"` (+1 more)	Pass	0.001s
Metadata row with three columns stores value	`metadata.get("dataType_units") == "m3/s"`	Pass	0.001s
Add participant auto generates id	`r.status_code == 200; new_p["id"].startswith("P-")`	Pass	0.001s
Add participant unknown meeting returns 404	`r.status_code == 404`	Pass	0.001s
Delete participant	`r.status_code == 200; pid not in remaining`	Pass	0.003s
Delete participant unknown meeting returns 404	`r.status_code == 404`	Pass	0.001s
Update missing participant returns 404	`r.status_code == 404`	Pass	0.001s
Update participant	`r.status_code == 200; updated["status"] == "Attended"`	Pass	0.003s
Add participant auto id	`"P-001" in ids; "P-002" in ids`	Pass	0.002s
Add participant missing meeting returns 404	`r.status_code == 404`	Pass	0.002s
Add participant returns 200	`r.status_code == 200; len(meeting["participants"]) == 2`	Pass	0.002s
Delete participant missing meeting returns 404	`r.status_code == 404`	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Delete participant returns 200	`r.status_code == 200; len(r.get_json()["meeting"]["participants"]) == 0`	Pass	0.002s
Update participant not found returns 404	`r.status_code == 404`	Pass	0.002s
Update participant returns 200	`r.status_code == 200; p["role"] == "CFO"`	Pass	0.002s
All holiday fields shown	`isinstance(result, list)`	Pass	0.000s
Holidays taken shown	`isinstance(result, list)`	Pass	0.000s
Last holiday date shown	`isinstance(result, list)`	Pass	0.000s
Without holiday fields, section should not appear.	`not any("Holiday" in t for t in texts)`	Pass	0.000s
Total holidays shown	`isinstance(result, list)`	Pass	0.000s
Arrears balance shown	`isinstance(result, list)`	Pass	0.000s
Five missed payments poor	`isinstance(result, list)`	Pass	0.000s
Four missed payments concerning	`isinstance(result, list)`	Pass	0.000s
Highest arrears shown	`isinstance(result, list)`	Pass	0.000s
In arrears flag true	`isinstance(result, list)`	Pass	0.000s
In arrears score 5	`isinstance(result, list)`	Pass	0.000s
Many missed payments over 3	`isinstance(result, list)`	Pass	0.000s
No missed payments good	`isinstance(result, list)`	Pass	0.000s
One missed payment good	`isinstance(result, list)`	Pass	0.000s
Some missed payments 1 to 3	`isinstance(result, list)`	Pass	0.000s
Three missed payments concerning	`isinstance(result, list)`	Pass	0.000s
Two missed payments good	`isinstance(result, list)`	Pass	0.000s
Zero missed payments excellent	`isinstance(result, list)`	Pass	0.000s
Documentation pdf known model no file 404	`r.status_code == 404`	Pass	0.000s
Documentation pdf unknown model 404	`r.status_code == 404`	Pass	0.001s
Mrc tor pdf responds	`r.status_code in (200, 404)`	Pass	0.001s
Parameter inventory pdf responds	`r.status_code in (200, 404)`	Pass	0.000s
Results pdf known model no file 404	`r.status_code == 404`	Pass	0.000s
Results pdf unknown model 404	`r.status_code == 404`	Pass	0.000s
Line 323: valid model dir but PDF file missing → 404.	`r.status_code == 404; "not found" in r.get_json()["message"].lower()`	Pass	0.001s
Line 325: model documentation PDF served when file exists.	`r.status_code == 200`	Pass	0.001s
Line 357: MRC ToR PDF missing → 404.	`r.status_code == 404`	Pass	0.001s
Line 359: MRC ToR PDF served.	`r.status_code == 200`	Pass	0.004s

Test Description	Acceptance Criteria	Result	Time
Line 337: valid model dir but test_results.pdf missing → 404.	`r.status_code == 404`	Pass	0.003s
Line 339: test results PDF served when file exists.	`r.status_code == 200`	Pass	0.002s
STORM-SEVERE-001 (peak 6.5m) should come before STORM-MODERATE-002 (4.8m).	`severe_idx < moderate_idx`	Pass	0.003s
GET /trading/stress/storms returns storms sorted by peak_level_m DESC.	`len(storms) >= 2; peaks == sorted(peaks, reverse=True)`	Pass	0.004s
storms[0] must have the highest peak_level_m.	`storms[0]['peak_level_m'] >= storms[-1]['peak_level_m']`	Pass	0.003s
When seg_start == seg_end, norm ~ 0 -¿ degenerate fallback path.	`abs(lon - (-0.30)) < 1e-9`	Pass	0.000s
Offset moves point	`dist >= PropertyPortfolioGenerator.MIN_OFFSET_M * 0.5`	Pass	0.000s
Returns two floats	`isinstance(lat, float); isinstance(lon, float)`	Pass	0.000s
Persistent all events have readings	`"flood_depth_m" in event; "damage_ratio" in event` (+2 more)	Pass	0.168s
Persistent four events recorded	`result["summary"]["floods_at_nearest_gauge"] == 4`	Pass	0.165s
Create mapping returns flat dict	`isinstance(stormts_cdm.create_mapping(data), dict); isinstance(prs_cdm.create_mapping(data), dict)`	Pass	0.000s
List all fields has gauge set	`any("GaugeSet" in f for f in prs_cdm.list_all_fields())`	Pass	0.000s
Required fields include swap id	`any("SwapID" in f for f in prs_cdm.get_required_fields())`	Pass	0.000s
Schema has root	`isinstance(stormts_cdm.schema, dict); len(stormts_cdm.schema) > 0` (+2 more)	Pass	0.000s
FAIL if any output from STEP_IO is missing or empty on disk.	—	Pass	0.985s
At least one trained classifier must be documented in training_summary.	`stress_storms_dir.exists(); summary_path.exists()` (+1 more)	Skip	0.000s
Directories that exist but are empty are just as broken as missing.	`not empty`	Pass	0.065s

Test Description	Acceptance Criteria	Result	Time
Line 48: fmt_gbp(0) returns em-dash.	`isinstance(result, list)`	Pass	0.000s
After an EOD snap, series has one entry.	`len(data['series']) == 1`; `'date' in entry` (+4 more)	Pass	0.036s
Returns empty series when no EOD snapshots exist.	`resp.status_code == 200`; `data['status'] == 'success'` (+1 more)	Pass	0.001s
Missing positions key returns drawing	`isinstance(result, Drawing)`	Pass	0.000s
Negative gauge P&L -¿ bar drawn to the left.	`isinstance(result, Drawing)`	Pass	0.000s
No positions returns drawing	`isinstance(result, Drawing)`	Pass	0.000s
Positions exist but all running_pnl = 0 -¿ gauge_pnl has values.	`isinstance(result, Drawing)`	Pass	0.000s
In –all mode, resolve_prerequisites should not be called.	`run_all is True`	Pass	0.000s
Running –propertyts with stale prereqs should enable their flags.	`args.gauges is True`; `args.properties is True` (+3 more)	Pass	0.000s
In –strict mode, resolve_prerequisites should not be called.	`not run_all`; `args.strict is True`	Pass	0.000s
JS must POST to /api/v1/trading/stress/portfolio-run.	`'/api/v1/trading/stress/portfolio-run' in ps_js`	Pass	0.000s
JS must fetch from /api/v1/trading/stress/portfolio-storms.	`'/api/v1/trading/stress/portfolio-storms' in ps_js`	Pass	0.000s
portfolio-run endpoint must use POST method.	`'POST' in ps_js`	Pass	0.000s
Full Detail button must navigate to Tab 7 via switchTab('stress').	`"switchTab('stress')" in ps_js`	Pass	0.000s
'Full Detail' link button must be present — navigates to Tab 7 gauge view.	`'Full Detail' in ps_js`	Pass	0.000s
_psRenderGaugeTable must be defined — renders per-gauge trade rows.	`'_psRenderGaugeTable' in ps_js`	Pass	0.000s
_psRenderGaugePnlTab must be defined.	`'_psRenderGaugePnlTab' in ps_js`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
tdStressGaugeHint must be set — passes selected gauge to Tab 7 pre-selection.	`'tdStressGaugeHint' in ps_js`	Pass	0.000s
_stressStormHint must be set — passes selected storm to Tab 7 auto-selection.	`'_stressStormHint' in ps_js`	Pass	0.000s
ps- IDs must not collide with Tab 7's td-stress-gauge ID.	`'td-stress-gauge' not in ps_js`	Pass	0.000s
Content area uses ps-content ID.	`'ps-content' in ps_js`	Pass	0.000s
Storm selector uses ps- prefix (not td-stress-*).	`'ps-storm-sel' in ps_js`	Pass	0.000s
Sub-tab button IDs are built using ps-tab- prefix.	`"ps-tab-'" in ps_js or "'ps-tab-' +" in ps_js or 'ps-tab-...'; "'pflood'" in ps_js` (+3 more)	Pass	0.000s
Rendered full JS must be of substantial length — sanity check that	`len(full_js) > 200000`	Pass	0.000s
loadPortStressData() must be called in the portstress switchTab branch.	`'loadPortStressData' in full_js`	Pass	0.000s
psResult must survive f-string rendering — absent if brace escaping broke.	`'psResult' in full_js`	Pass	0.000s
'Port Stress' tab label must appear in the rendered trading desk.	`'Port Stress' in full_js`	Pass	0.000s
portStressView variable must appear — panel appends it to DOM.	`'portStressView' in full_js`	Pass	0.000s
createPortStressView() call must appear in panel construction.	`'createPortStressView' in full_js`	Pass	0.000s
switchTab must handle tab === 'portstress' branch.	`"tab === 'portstress'" in full_js`	Pass	0.000s
tabs array must include the portstress entry.	`"id: 'portstress'" in full_js`	Pass	0.000s
switchTab views array must include 'portstress'.	`"'portstress'" in full_js`	Pass	0.000s
hidePanel must call psCleanupCharts() to destroy chart instances.	`'psCleanupCharts' in full_js`	Pass	0.000s
psResult state variable must appear inside the IIFE, not after it.	`idx > 0; idx < iife_close`	Pass	0.000s
P(flood) tab contains the heatmap canvas and chart area container.	`'ps-heatmap-canvas' in ps_js; 'ps-pflood-chart-area' in ps_js`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
_psRenderPFloodTab must be defined.	`'_psRenderPFloodTab' in ps_js`	Pass	0.000s
Threshold bar colours must be present: severe red, warning orange, alert amber.	`'#c62828' in ps_js`; `'#e65100' in ps_js` (+1 more)	Pass	0.000s
P(flood) tab creates a Chart.js instance.	`'Chart(' in ps_js`	Pass	0.000s
Canvas element uses ps-portpnl-canvas ID.	`'ps-portpnl-canvas' in ps_js`	Pass	0.000s
Portfolio P&L tab uses the shared fmtGBP formatter.	`'fmtGBP' in ps_js`	Pass	0.000s
_psRenderPortPnlTab must be defined.	`'_psRenderPortPnlTab' in ps_js`	Pass	0.000s
Horizontal bar chart needs indexAxis: 'y' in Chart.js options.	`'indexAxis' in ps_js`	Pass	0.000s
Severity rows must navigate to Gauge P&L sub-tab via psSwitchSubTab.	`'psSwitchSubTab' in ps_js`	Pass	0.000s
All four severity group labels must be present in the JS output.	`'SEVERE' in ps_js`; `'WARNING' in ps_js` (+2 more)	Pass	0.000s
_psRenderSeverityTab must be defined.	`'_psRenderSeverityTab' in ps_js`	Pass	0.000s
Severity row click must set psSelectedGaugeId before switching sub-tab.	`'psSelectedGaugeId' in ps_js`	Pass	0.000s
Severe group must use the #ffebee background colour.	`ps_js.lower().find('#ffebee') >= 0`	Pass	0.000s
All four sub-tab render functions must be present.	`'_psRenderPFloodTab' in ps_js`; `'_psRenderPortPnlTab' in ps_js` (+2 more)	Pass	0.000s
'psCleanupCharts' must be defined — destroys Chart.js instances.	`'psCleanupCharts' in ps_js`	Pass	0.000s
'createPortStressView' must be defined — builds the Tab 8 DOM.	`'createPortStressView' in ps_js`	Pass	0.000s
'loadPortStressData' must be defined — fetches storm list on tab open.	`'loadPortStressData' in ps_js`	Pass	0.000s
'psStormChanged' must be defined — handles storm dropdown selection.	`'psStormChanged' in ps_js`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
'psSwitchSubTab' must be defined — controls the 4 sub-tabs.	`'psSwitchSubTab' in ps_js`	Pass	0.000s
port_stress.get_js() returns non-empty string without errors.	`len(ps_js) > 0`	Pass	0.000s
Module-level state variables must be declared.	`'psResult' in ps_js`; `'psActiveSubTab' in ps_js` (+1 more)	Pass	0.000s
Return value still has the correct structure after classifier training.	`result["num_sequences"] == 20`; `"type_counts" in result`	Pass	18.371s
train_classifier=True without gauge_id covers lines 364-384.	`result["num_gauges"] == 3`; `stressm_dir.exists()` (+1 more)	Pass	18.753s
Print summary stats for the flood portfolio.	—	Pass	0.115s
Verify properties receive the expected number of storm events.	—	Pass	0.076s
Known storm returns success	`r.status_code == 200`; `r.get_json()["status"] == "success"`	Pass	0.004s
Mortgaged property has ltv fields	`len(mortgaged) >= 1`; `"outstanding_balance" in p` (+3 more)	Pass	0.004s
No pts dir returns 404	`r.status_code == 404`; `r.get_json()["status"] == "error"` (+2 more)	Pass	0.000s
Options returns ok	`r.status_code == 200`; `r.status_code == 200`	Pass	0.000s
Portfolio damage pct non negative	`pct >= 0`	Pass	0.004s
Portfolio totals count	`port["properties_affected"] == len(r.get_json()["properti...`	Pass	0.004s
Properties sorted by damage descending	`amounts == sorted(amounts, reverse=True)`; `amounts == sorted(amounts, reverse=True)`	Pass	0.003s
Property entry has financial fields	`"property_id" in prop`; `"property_value" in prop` (+9 more)	Pass	0.003s
Response has portfolio	`"portfolio" in data`; `"total_properties" in port` (+4 more)	Pass	0.004s
Response has properties list	`"properties" in data`; `len(data["properties"]) >= 1` (+2 more)	Pass	0.004s
Response has storm id	`r.get_json()["storm_id"] == sid`	Pass	0.004s
Unknown storm returns 404	`r.status_code == 404`; `r.get_json()["status"] == "error"`	Pass	0.004s

Test Description	Acceptance Criteria	Result	Time
No pts dir returns 404	`r.status_code == 404`; `data["status"] == "error"`	Pass	0.000s
Options returns 200	`r.status_code == 200`	Pass	0.000s
Unknown storm returns 404	`r.status_code == 404`; `data["status"] == "error"`	Pass	0.004s
Mortgage enrichment present	`entry["has_mortgage"] is True`; `entry["outstanding_balance"] == pytest.approx(280_000)` (+2 more)	Pass	0.004s
Property entry has damage fields	`entry["property_id"] == "PROP-001"`; `entry["flood_depth_m"] == pytest.approx(0.5)` (+2 more)	Pass	0.004s
Response has portfolio key	`"portfolio" in data`; `key in portfolio`	Pass	0.004s
Response has properties list	`"properties" in data`; `len(props) >= 1`	Pass	0.004s
Storm id in response	`data["storm_id"] == "STORM-001"`	Pass	0.004s
Valid storm returns 200	`r.status_code == 200`; `data["status"] == "success"`	Pass	0.004s
API must not return name == storm_id (produces 'ID (ID)' in UI).	`s['name']`	Pass	0.002s
At least one storm must have a human-readable name (not a raw STORM-xxx id).	`named`	Pass	0.004s
In gauge_results, severe gauges appear before warning gauges.	`levels == sorted(levels)`	Pass	0.035s
Warning gauges appear before alert gauges in results.	`levels[i] <= levels[i + 1]`	Pass	0.031s
Gauges with num_trades ¿ 0 have non-empty trades list.	`len(g['trades']) > 0`	Pass	0.032s
Each trade has swap_id, trigger, notional, mtm, cash_price, stress_pnl,	`found_trade`; `'swap_id' in t` (+8 more)	Pass	0.031s
Exception in _load_stress_storms causes 500.	`resp.status_code == 500`	Pass	0.008s
When two storms have equal gauges_severe, the one with	`storms[0]['intensity_category'] == 'catastrophic'`	Pass	0.000s
patch _get_engines to raise RuntimeError -¿ 500.	`resp.status_code == 500`	Pass	0.004s
No stress_storms/ + cache cleared -¿ portfolio-run returns 404.	`resp.status_code == 404`	Pass	0.000s
POST with no body returns 400 or 500.	`resp.status_code in (400, 500)`	Pass	0.001s
Payer (is_payer=True) has positive notional in trade details.	`t['notional'] > 0`	Pass	0.023s

Test Description	Acceptance Criteria	Result	Time
portfolio_stress_pnl == sum(g['stress_pnl'] for g in gauges).	abs(data['portfolio_stress_pnl'] - expected_total) < 0.01	Pass	0.025s
Receiver (is_payer=False) has negative notional in trade details.	t['notional'] < 0	Pass	0.026s
When peak_level_m ¿= severe_level, p_flood should be 1.0.	gauge_001 is not None; gauge_001['p_flood'] == 1.0	Pass	0.026s
stress_pnl = cash_price - mtm; cash_price = signed_notional * p_flood.	data['status'] == 'success'; abs(t['stress_pnl'] - (t['cash_price'] - t['mtm'])) < 0.01	Pass	0.023s
Gauge not in storm's gauge_responses has p_flood=0.0 and impacted=False.	lambeth['p_flood'] == 0.0; lambeth['impacted'] is False	Pass	0.027s
Missing sequences message mentions doublet	"storm_sequences" in msg or "not found" in msg	Pass	0.027s
Missing sequences returns 404	r.status_code == 404	Pass	0.027s
No pts dir returns 404	r.status_code == 404	Pass	0.000s
No pts dir status error	pts_client_no_data.get(URL).get_json()["status"] == "error"	Pass	0.000s
Options returns ok	r.status_code == 200	Pass	0.000s
Missing mortgage json returns 200	client.get(URL).status_code == 200	Pass	0.118s
Missing mortgage json zero portfolio mortgages	client.get(URL).get_json()["total_portfolio_mortgages"] =...	Pass	0.117s
Missing property json returns 200	client.get(URL).status_code == 200	Pass	0.299s
Missing property json status success	client.get(URL).get_json()["status"] == "success"	Pass	0.120s
Missing property json zero portfolio value	client.get(URL).get_json()["total_portfolio_value"] == 0.0	Pass	0.116s
Mortgage impairment max is positive	mi["max"] > 0	Pass	0.093s
Mortgage impairment value	mi["max"] == pytest.approx(140000.0, rel=1e-3)	Pass	0.089s
Tail storm has mortgage impairment	len(tail) >= 1; tail[0]["mortgage_impairment"] > 0	Pass	0.092s
Prob loss pct is 50	client_two_storms.get(URL).get_json()["prob_loss_pct"] ==...	Pass	0.091s
Storm count is two	client_two_storms.get(URL).get_json()["storm_count"] == 2	Pass	0.092s
Storms with damage is one	client_two_storms.get(URL).get_json()["storms_with_damage"]	Pass	0.093s
Storms zero damage is one	client_two_storms.get(URL).get_json()["storms_zero_damage"]	Pass	0.091s

Test Description	Acceptance Criteria	Result	Time
Tail storms contains the damaged storm	`"STORM-0001" in storm_ids`	Pass	0.090s
Tail storms limited to 50	`len(client_two_storms.get(URL).get_json()["tail_storms"]) <=...`	Pass	0.090s
No mortgage impairment when above water	`mi["max"] == 0.0`	Pass	0.091s
Prob loss pct is 100	`pts_env["client"].get(URL).get_json()["prob_loss_pct"] ==...`	Pass	0.092s
Property damage cond mean positive	`pts_env["client"].get(URL).get_json()["property_damage"]["c...`	Pass	0.098s
Property damage max is positive	`pts_env["client"].get(URL).get_json()["property_damage"]["m...`	Pass	0.100s
Storm count is one	`pts_env["client"].get(URL).get_json()["storm_count"] == 1`	Pass	0.091s
Storms with damage is one	`pts_env["client"].get(URL).get_json()["storms_with_damage"]...`	Pass	0.090s
Storms zero plus with damage equals total	`data["storms_with_damage"] + data["storms_zero_damage"] =...`	Pass	0.094s
Total portfolio mortgages	`pts_env["client"].get(URL).get_json()["total_portfolio_mo...`	Pass	0.096s
Total portfolio value	`pts_env["client"].get(URL).get_json()["total_portfolio_va...`	Pass	0.092s
Histogram bin has lo hi count	`"lo" in bin0 and "hi" in bin0 and "count" in bin0`	Pass	0.108s
Mort histogram has 50 bins	`len(pts_env["client"].get(URL).get_json()["mort_histogram"]...`	Pass	0.091s
Mortgage impairment sub keys	`key in mi`	Pass	0.091s
Prop histogram has 50 bins	`len(pts_env["client"].get(URL).get_json()["prop_histogram"]...`	Pass	0.093s
Property damage sub keys	`key in pd`	Pass	0.091s
Returns 200	`pts_env["client"].get(URL).status_code == 200`	Pass	0.094s
Status success	`pts_env["client"].get(URL).get_json()["status"] == "success"`	Pass	0.094s
Tail storm has required fields	`len(storms) >= 1; f in s`	Pass	0.090s
Tail storms is list	`isinstance(pts_env["client"].get(URL).get_json()["tail_st...`	Pass	0.090s
Tail storms sorted descending	`damages == sorted(damages, reverse=True)`	Pass	0.089s
Top level keys present	`key in data`	Pass	0.094s
Portfolio value from known property	`client_unknown_prop.get(URL).get_json()["total_portfolio_va...`	Pass	0.093s
Unknown prop id is skipped	`data["storms_with_damage"] == 0`	Pass	0.091s
Zero depth cond mean is zero	`client_zero_depth.get(URL).get_json()["property_damage"]["c...`	Pass	0.097s
Zero depth prob loss is zero	`client_zero_depth.get(URL).get_json()["prob_loss_pct"] ==...`	Pass	0.097s
Zero depth property damage max is zero	`client_zero_depth.get(URL).get_json()["property_damage"]["m...`	Pass	0.094s
Zero depth storms with damage is zero	`client_zero_depth.get(URL).get_json()["storms_with_damage"]...`	Pass	0.092s

Test Description	Acceptance Criteria	Result	Time
trading/preloader.py must set window._tdPreloadDone (not bare assignment).	'window._tdPreloadDone = true' in src	Pass	0.000s
startup.py must initialise via window._tdPreloadDone (not bare var).	'window._tdPreloadDone = false' in src; 'var _tdPreloadDone = false' not in src	Pass	0.000s
startup.py must set window._tdPreloadDone = true after all fetches settle.	'window._tdPreloadDone = true' in src	Pass	0.000s
tradingdesk.py showPanel() must read window._tdPreloadDone, not bare _tdPreloadDone.	'window._tdPreloadDone' in js	Pass	0.000s
All three tabs must fetch from /trading/stress/storms.	'/trading/stress/storms' in js	Pass	0.000s
Gauge Stress tab auto-selects storms[0] (worst case).	'storms[0]' in ghc_stress_js or 'storms.length > 0' in gh...	Pass	0.000s
Gauge Stress tab storm dropdown uses all required storm fields.	field in ghc_stress_js	Pass	0.000s
Historical tab storm list uses all required storm fields.	field in hist_js	Pass	0.000s
Trading Desk storm dropdown includes peak_level_m in the label.	'peak_level_m' in td_stress_js	Pass	0.000s
Trading Desk Stress tab also auto-selects worst case storm.	'withTrades' in td_stress_js or \{} 'storms....	Pass	0.000s
Trading Desk Stress tab storm dropdown uses all required storm fields.	field in td_stress_js	Pass	0.000s
Alert trigger	result["trigger_level"] == "alert"; result["trigger_threshold_m"] == 4.0	Pass	0.000s
Custom notional	result["notional"] == 5_000_000	Pass	0.000s
Flat hazard curve	"fair_spread_bps" in result; result["use_term_structure"] is False	Pass	0.000s
Gauge not in hazard returns 404	r.status_code == 404	Pass	0.002s
Invalid trigger level returns 400	r.status_code == 400; r.get_json()["status"] == "error"	Pass	0.000s
Missing gauge id returns 400	r.status_code == 400	Pass	0.000s
No hazard file returns 404	r.status_code == 404; "not available" in r.get_json()["message"].lower() (+1 more)	Pass	0.000s
No json returns 400	r.status_code in (400, 500)	Pass	0.000s
Options returns ok	r.status_code == 200	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Result has fair spread bps	`"fair_spread_bps" in result; result["fair_spread_bps"] > 0`	Pass	0.000s
Result has npv	`"npv" in result; isinstance(result["npv"], float)`	Pass	0.000s
Result has survival probabilities	`len(sp) == 3; "1yr" in sp`	Pass	0.000s
Returns dict	`isinstance(result, dict)`	Pass	0.013s
Severe trigger	`result["trigger_level"] == "severe"; result["trigger_threshold_m"] == 5.0`	Pass	0.000s
Term structure true	`result["use_term_structure"] is True`	Pass	0.000s
Annual hazard rate matches input	`r["annual_hazard_rate"] == gauge["annual_hazard_rate_warn...`	Pass	0.000s
Gauge id in result	`r["gauge_id"] == gauge["gauge_id"]`	Pass	0.000s
Gauge name in result	`r["gauge_name"] == gauge["gauge_name"]`	Pass	0.000s
Severe annual hazard rate matches input	`r["annual_hazard_rate"] == gauge["annual_hazard_rate_seve...`	Pass	0.000s
hazard.py lines 174-176: exception during pricing -¿ 500.	`r.status_code == 500`	Pass	0.031s
hazard.py lines 146-147: gauge not in hazard data -¿ 404.	`r.status_code == 404`	Pass	0.029s
hazard.py lines 139-140: no gaugehc.json -¿ 404.	`r.status_code == 404`	Pass	0.028s
hazard.py line 123: no JSON body -¿ 400.	`r.status_code in (400, 500)`	Pass	0.028s
hazard.py lines 121-123: JSON null body -¿ 400.	`r.status_code == 400; r.get_json()["status"] == "error"`	Pass	0.027s
All trigger levels work: alert, warning, severe.	`r.status_code == 200`	Pass	0.117s
hazard.py lines 133-172: successful PRS pricing.	`r.status_code == 200; data["status"] == "success"` (+1 more)	Pass	0.126s
hazard.py: PRS pricing with default params.	`r.status_code == 200`	Pass	0.118s
Npv is sum of legs	`r["npv"] == pytest.approx( r["protection_leg_...`	Pass	0.000s
Premium leg negative	`r["premium_leg_npv"] < 0`	Pass	0.000s
Protection leg positive	`r["protection_leg_npv"] > 0`	Pass	0.000s
Nonzero recovery rate stored	`r["recovery_rate"] == pytest.approx(0.4)`	Pass	0.000s
Recovery rate affects fair spread	`r0["fair_spread_bps"] > r40["fair_spread_bps"]`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Higher risk free rate changes npv	`r_low["npv"] != pytest.approx(r_high["npv"], rel=1e-4)`	Pass	0.000s
Risk free rate not in output	`"risk_free_rate" not in r`	Pass	0.000s
Fair spread is float	`isinstance(r["fair_spread"], float)`	Pass	0.000s
Fair upfront is float	`isinstance(r["fair_upfront"], float)`	Pass	0.000s
Npv is float	`isinstance(r["npv"], float)`	Pass	0.000s
Running spread bps identity	`r["running_spread_bps"] == pytest.approx(r["running_sprea...`	Pass	0.000s
Survival probs all in zero one	`0.0 < val <= 1.0`	Pass	0.000s
Tenor 1yr gives one survival entry	`len(r["survival_probabilities"]) == 1; "1yr" in r["survival_probabilities"]`	Pass	0.000s
Use term structure stored false	`r["use_term_structure"] is False`	Pass	0.000s
Use term structure stored true	`r["use_term_structure"] is True`	Pass	0.000s
Not trained no metrics printed	`"AUC" not in out`	Pass	0.000s
Not trained status	`"insufficient_data" in out; "GAUGE-bad" in out` (+1 more)	Pass	0.000s
Trained shows feature importance	`"log_h_s" in out; "log_t_end" in out`	Pass	0.000s
Trained shows label info	`"severe_warning" in out; "5.50" in out`	Pass	0.000s
Trained status prints metrics	`"trained" in out; "0.9200" in out` (+1 more)	Pass	0.000s
Percentile between alert and warning -¿ '← ALERT' marker (line 480).	`"ALERT" in out`	Pass	0.000s
All sequence types reported	`"isolated" in out; "doublet" in out` (+1 more)	Pass	0.000s
Categories with zero peaks are skipped via 'continue' (line 493).	`"moderate" in out; "catastrophic" not in out`	Pass	0.000s
Output contains gauge id	`"GAUGE-test-pgp" in out`	Pass	0.000s
Percentile ¿= severe -¿ '← SEVERE' marker.	`"SEVERE" in out`	Pass	0.000s
Percentile between warning and severe -¿ '← WARNING' marker.	`"WARNING" in out`	Pass	0.000s
Summary with full data does not crash	—	Pass	0.005s
Summary with no data does not crash	—	Pass	0.000s
Summary with partial data does not crash	—	Pass	0.000s
Logs gauge name	`isinstance(result["gauge_name"], str)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Logs without error	`len(caplog.records) >= 0`	Pass	0.001s
Alert trigger in log	`"ALERT" in combined`	Pass	0.000s
All survival years in log	`key in combined`	Pass	0.000s
Contract terms header in log	`"Contract Terms" in combined`	Pass	0.000s
Fair spread in log	`"Fair Spread" in combined`	Pass	0.000s
Gauge id in log	`result["gauge_id"] in combined`	Pass	0.000s
Gauge name in log	`result["gauge_name"] in combined`	Pass	0.000s
Npv value in log	`"NPV" in combined`	Pass	0.000s
Premium leg in log	`"Premium Leg" in combined`	Pass	0.000s
Protection leg in log	`"Protection Leg" in combined`	Pass	0.000s
Recovery rate in log	`"Recovery Rate" in combined`	Pass	0.000s
Severe trigger in log	`"SEVERE" in combined`	Pass	0.000s
Survival curve label in log	`"Survival Curve" in combined`	Pass	0.000s
Trigger level in log	`"WARNING" in combined`	Pass	0.000s
Empty directory returns empty list	`result == []`	Pass	0.000s
Non matching files skipped	`result == []`	Pass	0.000s
Output dir none uses input dir	`isinstance(result, list); any("gaugehd_39999" in r for r in result)`	Pass	0.015s
Output files created	`all(Path(p).exists() for p in result)`	Pass	0.015s
Process directory error continues	`isinstance(result, list)`	Pass	0.017s
Processes matching files	`len(result) == 2`	Pass	0.030s
Line 61/80: custom glob pattern selects different files.	`len(result) == 1`	Pass	0.002s
Line 87-88: error on one file doesn't stop processing others.	`len(result) >= 1`	Pass	0.003s
Multiple CSV files in directory are all processed.	`len(result) == 2`	Pass	0.003s
Line 81: station_id extracted by stripping _gdf from stem.	`len(result) == 1; "99999" in result[0]`	Pass	0.002s
Line 288: empty gauge_lookup -¿ no nearest -¿ None.	`result is None`	Pass	0.163s
Line 283: empty prop_id -¿ None.	`result is None`	Pass	0.164s
Line 283: prop_lat == 0 -¿ None.	`result is None`	Pass	0.164s
Few events still uses event count	`result["has_gev"] is False; result["pricing_method"] == "event_count"`	Pass	0.097s
Many events uses event count	`result["flood_count"] == 10; result["pricing_method"] == "event_count"` (+1 more)	Pass	0.089s

Test Description	Acceptance Criteria	Result	Time
Spread equals flood count over storms	`result["flood_count"] == 5; result["term_structure"]["severe"]["prs_spread_bps"][0] =...`	Pass	0.098s
Non flooded events excluded from depth count	`result["flood_count"] == 4`	Pass	0.087s
Zero flooded events uses event count pricing	`result["has_gev"] is False; result["pricing_method"] == "event_count"` (+1 more)	Pass	0.088s
Max depth positive for flooded property	`result["summary"]["max_depth_m"] > 0; result["summary"]["mean_depth_m"] > 0`	Pass	0.088s
With zero flood events, gauge_flood_count=0 so transmission_rate=0.	`result["nearest_gauges"][0]["flood_transmission_rate"] ==...`	Pass	0.101s
Zero depths for non flooded	`result["summary"]["max_depth_m"] == 0.0; result["summary"]["mean_depth_m"] == 0.0`	Pass	0.092s
Property with elevation ¿ gauge peak -¿ depth 0.	`result["summary"]["floods_at_property"] == 0`	Pass	0.168s
Gauge response with exceeded_alert=False -¿ 0 gauge storms.	`result["summary"]["floods_at_nearest_gauge"] == 0`	Pass	0.165s
Summary dict has floods_at_nearest_gauge etc.	`"property_id" in s; "floods_at_nearest_gauge" in s` (+2 more)	Pass	0.164s
Full processing writes a JSON file for the property.	`result is not None; (pts_dir / "PROP-001.json").exists()`	Pass	0.165s
If a single property raises during generation, failed_properties	`result["processing_stats"]["failed_properties"] >= 1; result["processing_stats"]["successful_properties"] >= 2`	Pass	0.514s
getSampleStyleSheet() includes 'Title' by default, so line 65 branch	`'Title' in page.styles`	Pass	0.000s
Trade with property_id computes basis_delta_bps and basis_dv01.	`enriched['property_id'] == 'PROP-001'; 'basis_delta_bps' in enriched` (+2 more)	Pass	0.002s
Create mapping returns flat dict	`isinstance(mapping, dict); mapping.get("gauge_id") == "GAUGE-test1234"` (+7 more)	Pass	0.000s
Required fields not empty	`len(property_cdm.get_required_fields()) > 0`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Schema has property root	`len(list(property_cdm.schema.keys()))` `> 0`	Pass	0.000s
Validate valid property	`isinstance(property_cdm.validate(data), dict)`	Pass	0.000s
Existing property returns success	`r.status_code == 200;` `r.get_json()["status"] == "success"` (+3 more)	Pass	0.002s
Include readings false strips readings	`"readings" not in event`	Pass	0.003s
Include readings true keeps readings	`"readings" in data["flood_events"][0];` `len(data["flood_events"][0]["readings"])` `== 168`	Pass	0.003s
Missing property returns 404	`r.status_code == 404;` `r.get_json()["status"] == "error"` (+1 more)	Pass	0.000s
Options returns ok	`r.status_code == 200; r.status_code == 200` (+1 more)	Pass	0.000s
Depth thresholds	`"severe" in dt;` `"annual_probability" in dt["severe"]`	Pass	0.004s
Elevation m	`"elevation_m" in pc;` `isinstance(pc["elevation_m"], (int, float))`	Pass	0.004s
Flood count	`"flood_count" in pc;` `isinstance(pc["flood_count"], int)`	Pass	0.004s
Flood zone	`"flood_zone" in pc;` `pc["flood_zone"] in EA_FLOOD_ZONE_RATES`	Pass	0.004s
Flood zone in propertyhc should match the propertyts source.	`pc.get("flood_zone") == pts.get("flood_zone")`	Pass	0.009s
Floor level m	`"floor_level_m" in pc`	Pass	0.004s
Gev params is none	`pc.get("gev_params") is None`	Pass	0.004s
Has gev is false	`pc.get("has_gev") is False`	Pass	0.004s
Location	`"lat" in loc or "latitude" in loc`	Pass	0.004s
Nearest gauge has basis bps	`"severe" in basis;` `"values" in basis["severe"]` (+1 more)	Pass	0.004s
Nearest gauge has flood counts	`"property_flood_count" in ng;` `"gauge_flood_count" in ng` (+1 more)	Pass	0.004s
Nearest gauges	`len(ngs) >= 1;` `"gauge_id" in ng` (+2 more)	Pass	0.004s
Only severe threshold should exist (no any_flood or moderate).	`set(dt.keys()) == {"severe"}`	Pass	0.005s

Test Description	Acceptance Criteria	Result	Time
Pricing method is event count	`pc.get("pricing_method") == "event_count"`	Pass	0.004s
Property id	`pc.get("property_id") == prop_id`	Pass	0.004s
Spread should equal flood_count / num_storms * 10000.	`abs(spreads[0] - expected) < 0.1`	Pass	0.004s
Spreads non negative	`s >= 0`	Pass	0.004s
Summary	`"avg_basis_bps" in summary; "flood_transmission_rate" in summary`	Pass	0.004s
Term structure exists	`"tenors" in ts; isinstance(ts["tenors"], list)` (+1 more)	Pass	0.004s
Term structure has severe	`"severe" in ts; "prs_spread_bps" in severe`	Pass	0.004s
All tenors should have the same spread (storms are independent).	`all(s == spreads[0] for s in spreads)`	Pass	0.004s
GET /api/v1/properties/¡id¿/hazard returns hazard data for property with curves.	`response.status_code == 200; data['status'] == 'success'` (+2 more)	Pass	0.002s
GET /api/v1/properties/¡id¿/hazard returns 404 for unknown property.	`response.status_code == 404`	Pass	0.002s
GET /api/v1/properties/¡id¿/hazard returns 404 for property without curves.	`response.status_code == 404; data['status'] == 'error'` (+1 more)	Pass	0.002s
GET /api/v1/propertyhc/basis returns basis table.	`response.status_code == 200; data['status'] == 'success'` (+5 more)	Pass	0.002s
GET /api/v1/propertyhc/summary returns portfolio hazard summary.	`response.status_code == 200; data['status'] == 'success'` (+3 more)	Pass	0.002s
Data contains hazard curve	`"hazard_curve" in r.get_json()["data"]`	Pass	0.002s
Data contains nearest gauges	`"nearest_gauges" in r.get_json()["data"]`	Pass	0.002s
Data contains property id	`r.get_json()["data"]["property_id"] == "PROP-001"`	Pass	0.002s
Data contains summary	`"summary" in r.get_json()["data"]`	Pass	0.002s
Known property returns 200	`r.status_code == 200`	Pass	0.002s
Known property status success	`r.get_json()["status"] == "success"`	Pass	0.002s
No data returns 404	`r.status_code == 404; r.status_code == 404`	Pass	0.000s
No data status error	`r.get_json()["status"] == "error"`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Options returns 200	`r.status_code == 200`; `r.status_code == 200`	Pass	0.000s
Options status ok	`r.get_json()["status"] == "ok"`; `r.get_json()["status"] == "ok"`	Pass	0.000s
Response has data key	`"data" in r.get_json()`; `"data" in r.get_json()`	Pass	0.002s
Second property also works	`r.status_code == 200`; `r.get_json()["data"]["property_id"] == "PROP-002"`	Pass	0.002s
Unknown property message contains id	`"PROP-GHOST" in r.get_json()["message"]`	Pass	0.002s
Unknown property returns 404	`r.status_code == 404`	Pass	0.002s
Unknown property status error	`r.get_json()["status"] == "error"`	Pass	0.002s
configure(panel_width, panel_height) updates both.	`obj.panel_width == "1000px"`; `obj.panel_height == "800px"`	Pass	0.000s
configure() with no args leaves values unchanged.	`obj.panel_width == original_width`; `obj.panel_height == original_height`	Pass	0.000s
Lines 303-304: configure(panel_height=...) updates panel_height.	`obj.panel_height == "700px"`	Pass	0.000s
Lines 301-302: configure(panel_width=...) updates panel_width.	`obj.panel_width == "900px"`	Pass	0.000s
Line 308: get_statistics() returns dict with panel_width and panel_height.	`stats["panel_width"] == "750px"`; `stats["panel_height"] == "550px"`	Pass	0.000s
get_statistics() always returns a dict.	`isinstance(stats, dict)`; `"panel_width" in stats` (+1 more)	Pass	0.000s
PropertyID must be populated in Header (not just PropertyAttributes).	`len(empty_header_ids) == 0`	Pass	0.003s
property.json must contain at least one property.	`len(ids) > 0`	Pass	0.004s
Property IDs in propertyhc.json must exist in property.json.	`not orphaned`	Pass	0.002s
Properties with hazard curves should also have timeseries data.	`not missing`	Pass	0.001s
propertyts/ files must reference property IDs from property.json.	`len(missing) == 0`	Pass	1.873s
Property IDs in propertyts/ files must exist in property.json.	`not orphaned`	Pass	0.004s

Test Description	Acceptance Criteria	Result	Time
Generator does not use sequential format	'GAUGE-{i:' not in src; 'GAUGE-{index' not in src (+3 more)	Pass	0.000s
Generator source uses prop prefix	'PROP-' in src	Pass	0.001s
Generator source uses uuid	'uuid' in src; 'uuid' in src	Pass	0.000s
Uuid4 produces protocol matching ids	GAUGE_ID_RE.match(gauge_id); PROP_ID_RE.match(prop_id) (+2 more)	Pass	0.000s
Accepts valid	PROP_ID_RE.match('PROP-a3f7bc12'); PROP_ID_RE.match('PROP-00000000') (+6 more)	Pass	0.000s
Rejects sequential	not PROP_ID_RE.match('PROP-001'); not PROP_ID_RE.match('PROP-1234') (+3 more)	Pass	0.000s
Rejects uppercase	not PROP_ID_RE.match('PROP-A3F7BC12'); not STORM_ID_RE.match('STORM-C7F4B8E2')	Pass	0.000s
Test total portfolio value calculation.	total == 2050000	Pass	0.002s
Test value aggregation by state.	'FL' in values; 'TX' in values (+2 more)	Pass	0.001s
Custom filename	PropertyLoader(tmp_path, filename="my_props.json").count(...	Pass	0.004s
Load all returns list	len(PropertyLoader(tmp_path).load_all()) == 3	Pass	0.003s
Missing file empty	PropertyLoader(tmp_path).load_all() == []	Pass	0.000s
Test PropertyID extraction from nested structure.	prop_id is not None; prop_id.startswith("PROP-")	Pass	0.002s
Test property summary generation.	len(summaries) == 3; 'propertyId' in summary (+4 more)	Pass	0.002s
Test loading property portfolio.	len(properties) == 3	Pass	0.001s
Test getting coordinates.	coords is not None; lat == pytest.approx(25.7617, rel=1e-4) (+1 more)	Pass	0.002s
Test getting coordinates for non-existent property.	coords is None	Pass	0.002s
Test getting location data.	location is not None; location['City'] == "Miami" (+1 more)	Pass	0.002s
Test getting location for non-existent property.	location is None	Pass	0.001s
Test finding properties within radius.	len(nearby) >= 1; len(nearby) == 2 (+1 more)	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Find by id	`PropertyLoader(tmp_path).find_by_id("PROP-001")` is not None	Pass	0.001s
List all has property id	`"propertyId" in summaries[0]`	Pass	0.001s
Test finding property by address.	`prop is not None; "PROP-001" == property_loader.get_entity_id(prop)`	Pass	0.002s
Test address search is case insensitive.	`prop is not None`	Pass	0.001s
Test address search with no match.	`prop is None`	Pass	0.002s
Test finding property by ID.	`prop is not None; prop['PropertyHeader']['Location']['City'] == "Miami"`	Pass	0.001s
Test finding properties by state.	`len(fl_props) == 2; len(tx_props) == 1`	Pass	0.006s
Test state search is case insensitive.	`len(props) == 2`	Pass	0.002s
Test finding properties by zip code.	`len(props) == 1; property_loader.get_entity_id(props[0]) == "PROP-001"`	Pass	0.001s
Test finding multiple properties in same zip.	`len(props) == 1`	Pass	0.001s
Passing an unhandled type triggers super().default() → TypeError.	—	Pass	0.000s
Lines 62-64: ground_level falls back to default_elevation.	`mapping["ground_level_meters"] == 10.0; mapping["elevation"] == 10.0`	Pass	0.000s
Lines 136-137: exception in create_mapping raises ValueError.	—	Pass	0.000s
Found returns mortgage	`r.status_code == 200; data["status"] == "success"` (+2 more)	Pass	0.001s
Loader exception returns 500	`r.status_code == 500`	Pass	0.001s
Not found returns 404	`r.status_code == 404; r.get_json()["status"] == "error"`	Pass	0.001s
Options returns ok	`r.status_code == 200; r.status_code == 200`	Pass	0.000s
All properties have nearest gauges	`len(missing) == 0`	Pass	0.000s
Each property should have at most 1 synthetic gauge.	`len(bad) == 0`	Pass	0.000s
Every nearest gauge should have gauge_elevation_m.	`len(bad) == 0`	Pass	0.001s
Line 258: continue when prop_id not in prop_values → no properties → 404.	`r.status_code == 404`	Pass	0.003s
Line 193: break when prop_id not in prop_values → no properties → 404.	`r.status_code == 404`	Pass	0.003s

Test Description	Acceptance Criteria	Result	Time
Output json contains generation metadata	`"generated_at" in meta`; `"catchment" in meta` (+1 more)	Pass	0.526s
Output json contains properties key	`"properties" in data`	Pass	0.782s
Property header has property id	`"PropertyHeader" in prop`; `"PropertyAttributes" in prop["PropertyHeader"]`	Pass	0.790s
Result data contains locations	`len(result["data"]["locations"]) == 3`	Pass	0.784s
Full property+mortgage report produces a valid PDF file.	`report_path.exists()`; `report_path.suffix == '.pdf'` (+1 more)	Pass	0.028s
Mortgage-focused report produces a valid PDF.	`report_path.exists()`; `report_path.suffix == '.pdf'` (+1 more)	Pass	0.021s
Output directory is created automatically if it doesn't exist.	`new_dir.exists()`; `report_path.exists()`	Pass	0.020s
Generated filename includes the property ID.	`'PROP-test0001' in report_path.name`	Pass	0.020s
Full report PDF should be non-trivial in size (multi-page).	`report_path.stat().st_size > 10000`	Pass	0.028s
Generated file is a valid PDF (starts with %PDF).	`header == b'%PDF-'`	Pass	0.028s
Property-only report (no mortgage) produces a valid PDF.	`report_path.exists()`; `report_path.suffix == '.pdf'` (+1 more)	Pass	0.020s
Report handles minimal/sparse property data without crashing.	`report_path.exists()`; `report_path.stat().st_size > 0`	Pass	0.014s
Risk-focused report produces a valid PDF.	`report_path.exists()`; `report_path.suffix == '.pdf'` (+1 more)	Pass	0.015s
Auto-select includes mortgage pages when mortgage data present.	`'mortgage_overview' in pages`; `'mortgage_details' in pages`	Pass	0.002s
Auto-select excludes mortgage pages when no mortgage data.	`'title_overview' in pages`; `'mortgage_overview' not in pages` (+1 more)	Pass	0.002s
Generator initialises with output directory.	`gen.output_dir == tmp_path`	Pass	0.002s
Generator lists all registered page modules.	`'title_overview' in pages`; `'location' in pages` (+2 more)	Pass	0.002s
Validates known page names.	`'title_overview' in valid`; `'location' in valid` (+1 more)	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Line 174-175: mortgage_data -¿ includes mortgage pages.	`"mortgage_overview" in pages`	Pass	0.002s
Line 174 (else): no mortgage_data -¿ no mortgage pages.	`"mortgage_overview" not in pages`	Pass	0.002s
Lines 214-216: exception in page -¿ logged, continue.	`isinstance(elements, list)`	Pass	0.002s
Lines 198-199: unknown page -¿ skip.	`isinstance(elements, list)`	Pass	0.002s
Lines 184-185: missing PropertyID -¿ 'unknown'.	`"unknown" in name`	Pass	0.002s
Lines 257-264: mortgage-focused report.	`path.exists()`	Pass	0.016s
Lines 251-255: property-only report.	`path.exists()`	Pass	0.014s
Lines 266-276: risk-focused report without mortgage data.	`path.exists()`	Pass	0.009s
Lines 266-276: risk-focused report with mortgage data.	`path.exists()`	Pass	0.011s
Line 313: get_page_categories.	`"property" in cats; "mortgage" in cats`	Pass	0.002s
Line 285: list_available_pages.	`isinstance(pages, list); "title_overview" in pages`	Pass	0.002s
Lines 69-70: output_dir=None -¿ uses config.get_property_reports_dir().	`gen.output_dir == prop_dir`	Pass	0.002s
Line 315: validate_pages.	`"title_overview" in valid; "bogus_page" in invalid`	Pass	0.002s
Categories dict initialized	`"property" in gen.categories; "mortgage" in gen.categories` (+1 more)	Pass	0.002s
Explicit output dir path	`isinstance(gen.output_dir, Path)`	Pass	0.002s
Explicit output dir string	`gen.output_dir == tmp_path`	Pass	0.002s
No output dir uses config	`gen.output_dir == prop_dir`	Pass	0.002s
Output dir created	`new_dir.exists()`	Pass	0.002s
Property pages dict initialized	`expected <= set(gen.pages.keys())`	Pass	0.002s
Test POST /api/v1/properties/report without body.	`response.status_code == 400`	Pass	0.000s
Test POST /api/v1/properties/report without propertyId.	`response.status_code == 400; 'Property ID is required' in data['message']`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Test POST /api/v1/properties/report with invalid property.	`response.status_code == 404`	Pass	0.002s
Test GET /api/v1/properties/¡id¿.	`response.status_code == 200`; `data['status'] == 'success'` (+1 more)	Pass	0.002s
Test GET /api/v1/properties/¡id¿ with invalid ID.	`response.status_code == 404`; `data['status'] == 'error'`	Pass	0.002s
Test legacy POST /api/v1/generate_property_report endpoint.	`response.status_code == 404`	Pass	0.002s
Test GET /api/v1/properties.	`response.status_code == 200`; `data['status'] == 'success'` (+2 more)	Pass	0.002s
Test OPTIONS preflight request.	`response.status_code == 200`	Pass	0.000s
No data returns 404	`r.status_code == 404`; `r.status_code == 404` (+2 more)	Pass	0.000s
Options returns 200	`r.status_code == 200`; `r.status_code == 200` (+2 more)	Pass	0.000s
Unknown property returns 404	`r.status_code == 404`; `r.status_code == 404`	Pass	0.002s
Unknown property status error	`r.get_json()["status"] == "error"`; `r.get_json()["status"] == "error"`	Pass	0.002s
Valid property has data	`"data" in r.get_json()`; `"data" in r.get_json()`	Pass	0.002s
Valid property returns 200	`r.status_code == 200`; `r.status_code == 200`	Pass	0.003s
Valid property status success	`r.get_json()["status"] == "success"`; `r.get_json()["status"] == "success"`	Pass	0.002s
Shd has term structure	`"term_structure" in pc`; `"severe" in pc["term_structure"]`	Pass	0.100s
Shd output file is propertyshd json	`(output_dir / "propertyshd.json").exists()`; `not (output_dir / "propertyhc.json").exists()`	Pass	0.099s
shd mode should look for propertytsd/ not propertyts/.	—	Pass	0.000s
No data returns 404	`r.status_code == 404`; `r.status_code == 404` (+2 more)	Pass	0.000s
Options returns 200	`r.status_code == 200`; `r.status_code == 200` (+2 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Unknown property returns 404	`r.status_code == 404`; `r.status_code == 404`	Pass	0.002s
Unknown property status error	`r.get_json()["status"] == "error"`; `r.get_json()["status"] == "error"`	Pass	0.002s
Valid property has data	`"data" in r.get_json()`; `"data" in r.get_json()`	Pass	0.002s
Valid property returns 200	`r.status_code == 200`; `r.status_code == 200`	Pass	0.002s
Valid property status success	`r.get_json()["status"] == "success"`; `r.get_json()["status"] == "success"`	Pass	0.002s
She has term structure	`"term_structure" in pc`; `"severe" in pc["term_structure"]`	Pass	0.106s
She output file is propertyshe json	`(output_dir / "propertyshe.json").exists()`	Pass	0.098s
she mode should look for propertyt-se/ not propertyts/.	—	Pass	0.000s
Existing property returns success	`r.status_code == 200`; `r.get_json()["status"] == "success"` (+3 more)	Pass	0.004s
Flood events tagged with sequence type	`len(events) >= 1`; `"sequence_type" in events[0]`	Pass	0.004s
Missing property returns 404	`r.status_code == 404`; `r.get_json()["status"] == "error"` (+1 more)	Pass	0.000s
Nearest gauges enriched with flood stages	`len(gauges) >= 1`; `"flood_stages" in gauges[0]`	Pass	0.004s
Options returns ok	`r.status_code == 200`; `r.status_code == 200` (+1 more)	Pass	0.000s
Response has flood events	`"flood_events" in data`; `len(data["flood_events"]) >= 1`	Pass	0.004s
Response has nearest gauges	`"nearest_gauges" in data`	Pass	0.004s
Response has property info	`"property_info" in data`; `"elevation_m" in data["property_info"]`	Pass	0.003s
Lines 157-158: gauge.json missing triggers except pass.	`r.status_code == 200`; `ng["flood_stages"] == {}`	Pass	0.029s
Lines 134-135: storm_sequences.json missing triggers except pass.	`r.status_code == 200`; `data["status"] == "success"` (+1 more)	Pass	0.029s
Property trace includes mortgage	`"mortgages" in steps`	Pass	0.003s
Property trace includes propertyhc	`"propertyhc" in steps`	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Flood zone in propertyts is derived from synthetic gauge elevation,	`zone in EA_FLOOD_ZONE_RATES`	Pass	0.009s
Propertyts has elevation	`"elevation_m" in pts; isinstance(pts["elevation_m"], (int, float))`	Pass	0.010s
Propertyts has flood events	`"flood_events" in pts; isinstance(pts["flood_events"], list)`	Pass	0.010s
Propertyts has flood zone	`"flood_zone" in pts; pts["flood_zone"] in EA_FLOOD_ZONE_RATES`	Pass	0.009s
Propertyts has floor level	`"floor_level_m" in pts`	Pass	0.009s
Propertyts has location	`"lat" in loc; "lon" in loc`	Pass	0.009s
Propertyts has nearest gauges	`len(ngs) > 0; "gauge_id" in ng` (+1 more)	Pass	0.009s
Propertyts has property id	`pts.get("property_id") == prop_id`	Pass	0.032s
GET /api/v1/propertyts/storms returns storms with correct structure.	`response.status_code == 200; data['status'] == 'success'` (+8 more)	Pass	0.009s
Storm list must only include storms with property damage.	`len(non_flooding) == 0`	Pass	0.009s
GET /api/v1/properties/¡id¿/floods returns flood events for known property.	`response.status_code == 200; data['status'] == 'success'` (+2 more)	Pass	0.002s
Property with few flood events should still return data from propertyts.	`response.status_code == 200; data['status'] == 'success'` (+1 more)	Pass	0.001s
GET /api/v1/properties/¡id¿/floods returns 404 for unknown property.	`response.status_code == 404; data['status'] == 'error'`	Pass	0.000s
GET /api/v1/properties/¡id¿/storms returns storm analysis data.	`response.status_code == 200; data['status'] == 'success'` (+4 more)	Pass	0.004s
GET /api/v1/properties/¡id¿/storms returns 404 for unknown property.	`response.status_code == 404`	Pass	0.000s
Property with zero flood events should return empty array, not error.	`response.status_code == 200; data['status'] == 'success'` (+4 more)	Pass	0.005s
GET /api/v1/propertyts/summary returns portfolio flood summary.	`response.status_code == 200; data['status'] == 'success'` (+1 more)	Pass	0.002s
Generates per property files	`pts_dir.exists(); (pts_dir / "portfolio_flood_summary.json").exists()`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Per property files have flood data	"property_id" in data; "nearest_gauges" in data (+1 more)	Pass	0.001s
Portfolio summary has statistics	"total_properties" in s; s["total_properties"] == N_PROPERTIES (+1 more)	Pass	0.002s
np.float64 is converted to Python float by encoder.	isinstance(result, float); abs(result - 3.14) < 1e-10	Pass	0.000s
Lines 69-70: exception returns validation_error dict.	"validation_error" in result; isinstance(result["validation_error"], list) (+2 more)	Pass	0.000s
get_required_fields returns expected paths.	len(fields) == 4; "PropertyHeader.Header.PropertyID" in fields (+1 more)	Pass	0.000s
Line 65: location_errors appended when non-empty.	"Location" in errors; len(errors["Location"]) == 2	Pass	0.000s
Line 60: Missing LatitudeDegrees.	"Location" in errors; any("LatitudeDegrees" in e for e in errors["Location"])	Pass	0.000s
Line 62: Missing LongitudeDegrees.	"Location" in errors; any("LongitudeDegrees" in e for e in errors["Location"])	Pass	0.000s
Valid data returns no errors.	property_validate(data) == {}	Pass	0.000s
Close to water increases score	near['vulnerability_score'] > far['vulnerability_score']	Pass	0.000s
Flooded depth computed	r['flood_depth'] == pytest.approx(1.0); r['vulnerability_score'] > 0	Pass	0.000s
High risk recommendations present	any("flood" in rec.lower() for rec in r['recommendations'])	Pass	0.000s
No flood	r['flood_depth'] == 0.0; r['vulnerability_score'] == 0	Pass	0.000s
None elevation returns unknown	r['risk_level'] == 'Unknown'; r['flood_depth'] is None (+1 more)	Pass	0.000s
None flood level returns unknown	r['risk_level'] == 'Unknown'	Pass	0.000s
Result dict keys	set(r.keys()) == {'flood_depth', 'risk_level', 'vulnerabi...	Pass	0.000s
Score capped at 100	r['vulnerability_score'] <= 100	Pass	0.000s
No mortgage returns zero fill	r.status_code == 200; prop["has_mortgage"] is False (+4 more)	Pass	0.003s

Test Description	Acceptance Criteria	Result	Time
No mortgage sequence	`r.status_code == 200; prop["has_mortgage"] is False`	Pass	0.003s
Basis entry has avg basis bps	`"avg_basis_bps" in entry`	Pass	0.002s
Basis entry has flood count	`"flood_count" in entry`	Pass	0.002s
Basis entry has nearest gauges	`"nearest_gauges" in entry`	Pass	0.002s
Basis entry has property id	`"property_id" in entry`	Pass	0.002s
Basis table count matches	`data["count"] == len(data["basis_table"])`	Pass	0.002s
Basis table length	`r.get_json()["count"] == 2`	Pass	0.002s
Results must be sorted highest avg_basis_bps first.	`bases == sorted(bases, reverse=True)`	Pass	0.002s
Nearest gauge has distance km	`"distance_km" in entry["nearest_gauges"][0]`	Pass	0.002s
Nearest gauge has gauge id	`"gauge_id" in entry["nearest_gauges"][0]`	Pass	0.002s
No data returns 404	`r.status_code == 404`	Pass	0.000s
No data status error	`r.get_json()["status"] == "error"`	Pass	0.000s
Options returns 200	`r.status_code == 200`	Pass	0.000s
Options status ok	`r.get_json()["status"] == "ok"`	Pass	0.000s
Response has basis table	`"basis_table" in r.get_json()`	Pass	0.003s
Response has count	`"count" in r.get_json()`	Pass	0.002s
Returns 200 with data	`r.status_code == 200`	Pass	0.002s
Status is success	`r.get_json()["status"] == "success"`	Pass	0.002s
Empty basis table is list	`isinstance(r.get_json()["basis_table"], list)`	Pass	0.002s
Empty count is zero	`r.get_json()["count"] == 0`	Pass	0.002s
Empty returns 200	`r.status_code == 200`	Pass	0.002s
Distribution avg basis bps	`"avg_basis_bps" in dist; dist["avg_basis_bps"] >= 0`	Pass	0.002s
Distribution avg flood count	`"avg_flood_count" in dist; dist["avg_flood_count"] > 0`	Pass	0.002s
Distribution max depth m	`"max_depth_m" in dist; dist["max_depth_m"] > 0`	Pass	0.002s
Distribution num properties	`dist["num_properties"] == 2`	Pass	0.002s
No data message mentions command	`"port" in r.get_json()["message"].lower() or "propertyhc"...`	Pass	0.001s
No data returns 404	`r.status_code == 404; r.status_code == 404`	Pass	0.000s
No data status is error	`r.get_json()["status"] == "error"`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Options returns 200	`r.status_code == 200`; `r.status_code == 200`	Pass	0.000s
Options status ok	`r.get_json()["status"] == "ok"`; `r.get_json()["status"] == "ok"`	Pass	0.000s
Response has data key	`"data" in r.get_json()`; `"data" in r.get_json()`	Pass	0.002s
Response has distribution	`"distribution" in r.get_json()["data"]`	Pass	0.001s
Response has metadata	`"metadata" in r.get_json()["data"]`	Pass	0.002s
Response has summary	`"summary" in r.get_json()["data"]`	Pass	0.002s
Returns 200 with data	`r.status_code == 200`	Pass	0.002s
Status is success	`r.get_json()["status"] == "success"`	Pass	0.002s
Empty curves avg flood count is zero	`r.get_json()["data"]["distribution"]["avg_flood_count"] == 0`	Pass	0.002s
Empty curves num properties is zero	`r.get_json()["data"]["distribution"]["num_properties"] == 0`	Pass	0.002s
Empty curves returns 200	`r.status_code == 200`	Pass	0.002s
No data returns 404	`r.status_code == 404`; `r.status_code == 404` (+2 more)	Pass	0.000s
No data status error	`r.get_json()["status"] == "error"`; `r.get_json()["status"] == "error"`	Pass	0.000s
Options returns 200	`r.status_code == 200`; `r.status_code == 200` (+2 more)	Pass	0.000s
Options status ok	`r.get_json()["status"] == "ok"`; `r.get_json()["status"] == "ok"`	Pass	0.000s
Response has metadata	`"metadata" in r.get_json()["data"]`; `"metadata" in r.get_json()["data"]`	Pass	0.002s
Response has summary	`"summary" in r.get_json()["data"]`; `"summary" in r.get_json()["data"]`	Pass	0.002s
Returns 200 with data	`r.status_code == 200`; `r.status_code == 200`	Pass	0.002s
Status is success	`r.get_json()["status"] == "success"`; `r.get_json()["status"] == "success"`	Pass	0.002s
No data returns 404	`r.status_code == 404`; `r.status_code == 404` (+2 more)	Pass	0.000s
No data status error	`r.get_json()["status"] == "error"`; `r.get_json()["status"] == "error"`	Pass	0.000s
Options returns 200	`r.status_code == 200`; `r.status_code == 200` (+2 more)	Pass	0.000s
Options status ok	`r.get_json()["status"] == "ok"`; `r.get_json()["status"] == "ok"`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Response has metadata	`"metadata" in r.get_json()["data"]; "metadata" in r.get_json()["data"]`	Pass	0.001s
Response has summary	`"summary" in r.get_json()["data"]; "summary" in r.get_json()["data"]`	Pass	0.002s
Returns 200 with data	`r.status_code == 200; r.status_code == 200`	Pass	0.002s
Status is success	`r.get_json()["status"] == "success"; r.get_json()["status"] == "success"`	Pass	0.002s
When no natural measures, recommendation should appear.	`isinstance(result, list)`	Pass	0.000s
Natural measures section rendered	`any("Natural" in t for t in texts)`	Pass	0.000s
Empty property returns list	`isinstance(result, list); len(result) > 0`	Pass	0.000s
Empty protection measures dict shows fallback	`any("No protection" in t for t in texts)`	Pass	0.000s
No protection measures shows fallback	`any("No protection" in t or "Protection" in t for t in te...`	Pass	0.000s
Full protection fewer recommendations	`isinstance(result, list)`	Pass	0.001s
Generate protection recommendations returns dict	`isinstance(result, dict)`	Pass	0.000s
Missing critical measures in recommendations	`isinstance(result, list)`	Pass	0.000s
Recommendations all natural false	`"Natural Solutions" in result`	Pass	0.000s
Recommendations flood re eligible	`"Insurance" in result`	Pass	0.000s
Recommendations missing critical	`"Priority Installations" in result`	Pass	0.000s
Recommendations section always rendered	`any("Recommendation" in t for t in texts)`	Pass	0.001s
All installed excellent rating	`any("Resilience" in t or "Protection" in t for t in texts)`	Pass	0.000s
Building measures listed	`isinstance(result, list)`	Pass	0.000s
8 of 10 = 80% → Excellent.	`isinstance(result, list)`	Pass	0.000s
4 of 10 = 40% → Fair.	`isinstance(result, list)`	Pass	0.000s
6 of 10 = 60% → Good.	`isinstance(result, list)`	Pass	0.000s
2 of 10 = 20% → Limited.	`isinstance(result, list)`	Pass	0.000s
0 of 10 = 0% → Poor.	`isinstance(result, list)`	Pass	0.000s
Critical measures listed	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
Emergency measures listed	`isinstance(result, list)`	Pass	0.000s
Uncategorised measure still shown	`isinstance(result, list)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Excess amount shown	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
Insurance premium shown	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
Other risk fields shown	`isinstance(result, list); len(result) > 1`	Pass	0.000s
Recommendations include flood re	`isinstance(result, list)`	Pass	0.000s
Risk assessment section rendered	`any("Insurance" in t for t in texts)`	Pass	0.000s
All entity ID suffixes are exactly 8 characters.	`all(v == 8 for v in suffix_lengths.values())`	Pass	0.000s
str(uuid.uuid4())[:8] always yields lowercase hex.	`suffix == suffix.lower(); re.match(r'^[0-9a-f]{8}$', suffix)`	Pass	0.001s
All four entity types use str(uuid4())[:8] — same derivation, same format.	`pattern.match(entity_id)`	Pass	0.000s
All exports	`'ClaimReportGenerator' in pkg.__all__`	Pass	0.000s
Import from package	`ClaimReportGenerator is not None`	Pass	0.000s
Current exceeds original is capped	`result["Mortgage"]["LoanDetails"]["CurrentBalance"] <= 40...`	Pass	0.000s
Interest over 15 capped	`result["Mortgage"]["LoanDetails"]["InterestRate"] == 12.0`	Pass	0.000s
Interest under 1 raised	`result["Mortgage"]["LoanDetails"]["InterestRate"] == 2.0`	Pass	0.000s
Ltv over 100 capped at 95	`result["Mortgage"]["LoanDetails"]["LTV"] == 95`	Pass	0.000s
Ltv under 10 set to 60	`result["Mortgage"]["LoanDetails"]["LTV"] == 60`	Pass	0.000s
Valid data unchanged	`result["Mortgage"]["LoanDetails"]["CurrentBalance"] == 35...; result["Mortgage"]["LoanDetails"]["LTV"] == 80` (+1 more)	Pass	0.000s
Current exceeds original is capped to 95pct	`result["Mortgage"]["LoanDetails"]["CurrentBalance"] <= 30...`	Pass	0.000s
Current within original unchanged	`result["Mortgage"]["LoanDetails"]["CurrentBalance"] == 35...`	Pass	0.000s
Delegates to random module when method present	`result is sentinel`	Pass	0.000s
Ltv over 100 set to 95	`result["Mortgage"]["LoanDetails"]["LTV"] == 95`	Pass	0.000s
Ltv under 10 set to 60	`result["Mortgage"]["LoanDetails"]["LTV"] == 60`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Valid ltv unchanged	`result["Mortgage"]["LoanDetails"]["LTV"] == 75`	Pass	0.000s
Get raci matrix	`response.status_code == 200; data["status"] == "success"` (+3 more)	Pass	0.002s
Get raci matrix file missing	`response.status_code == 404`	Pass	0.000s
Update raci activity	`response.status_code == 200; data["status"] == "success"` (+3 more)	Pass	0.003s
Update raci activity invalid role	`response.status_code == 400`	Pass	0.002s
Update raci escalation	`response.status_code == 200; data["status"] == "success"` (+3 more)	Pass	0.003s
Update raci role	`response.status_code == 200; data["status"] == "success"` (+3 more)	Pass	0.003s
Update raci role not found	`response.status_code == 404`	Pass	0.002s
No raci data returns 404	`r.status_code == 404; r.status_code == 404`	Pass	0.001s
Save raci failure returns 500	`r.status_code == 500; r.get_json()["status"] == "error"` (+4 more)	Pass	0.001s
Tier emphasis updated	`r.status_code == 200; activities[0]["tier_emphasis"] == "Tier 1 focus"`	Pass	0.002s
No raci data returns 404	`r.status_code == 404; r.status_code == 404`	Pass	0.001s
Save raci failure returns 500	`r.status_code == 500; r.get_json()["status"] == "error"` (+4 more)	Pass	0.002s
Save raci failure returns 500	`r.status_code == 500; r.get_json()["status"] == "error"` (+4 more)	Pass	0.002s
Remaining zero uses base amplitude	`isinstance(level, float)`	Pass	0.000s
Failure metadata covers key tests	`name in dl_mod.FAILURE_METADATA; 'principle' in meta` (+2 more)	Pass	0.000s
Recon table has principle column	`'BCBS Principle(s)' in header_row`	Pass	0.000s
Record and reload	`"gaugehd" in manifest["steps"]; manifest["steps"]["gaugehd"]["run_id"] == "run-test-001"` (+4 more)	Pass	0.007s
Record with auto run id	`entry["run_id"].startswith("run-")`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Empty EndDate returns default tenor of 5.	`result == 5`	Pass	0.000s
Invalid date format returns default tenor of 5 (ValueError path).	`result == 5`	Pass	0.000s
Missing EndDate key returns default tenor of 5.	`result == 5`	Pass	0.000s
None EndDate returns default tenor of 5 (TypeError path).	`result == 5`	Pass	0.000s
Valid far-future EndDate returns positive tenor.	`result >= 1`	Pass	0.000s
Compliant run is suitable	`'suitable for regulatory risk reporting' in texts`	Pass	0.000s
Issues run not suitable	`'not suitable for regulatory risk reporting' in texts; 'COMPLIANT' not in texts.replace( 'PARTIALLY ...`	Pass	0.001s
Remediation lists stale steps	`'Stale data' in texts; 'hazard' in texts`	Pass	0.001s
When lineage_results has a known failure, remediation cites it.	`'PARTIALLY COMPLIANT' in texts or 'NON-COMPLIANT' in texts`	Pass	0.001s
Shows strengths and weaknesses	`'Strengths' in texts; 'Weaknesses' in texts`	Pass	0.001s
Steps with empty output directories are skipped.	`"propertyts" in result["skipped"]`	Pass	0.019s
Repair with no existing manifest creates entries for all steps.	`len(result["repaired"]) == len(STEP_IO); result["skipped"] == []` (+2 more)	Pass	0.023s
Repaired hashes must match current file content.	`recorded_hash == hash_file(gauge_file)`	Pass	0.008s
Running repair twice produces identical hashes.	`out_info["hash"] == data2["steps"][step]["outputs"][out_n...`	Pass	0.035s
Input hashes recorded by repair should match the producer's output hash.	`gauges_out == props_in`	Pass	0.022s
Repair keeps run_id, timestamp, generator from existing entries.	`"gauges" in result["repaired"]; gauges_entry["run_id"] == "run-original"` (+3 more)	Pass	0.036s
Repair uses repair-YYYYMMDD-HHMMSS as run_id for new entries.	`gauges_entry["run_id"].startswith("repair-")`	Pass	0.004s
Steps whose outputs don't exist on disk are skipped.	`"gauges" in result["repaired"]; "properties" in result["skipped"]` (+1 more)	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Repair adds a runs entry listing all repaired steps.	`len(repair_runs) == 1; set(data["runs"][repair_runs[0]]) == set(result["repaired"])`	Pass	0.024s
Default report_type (no –report-type flag) hits the else branch	—	Pass	0.005s
Lines 349-351: success-logging block executes without error.	—	Pass	0.004s
Lines 444-445: monitoring without timeseries_data → condition False → else.	—	Pass	0.005s
Lines 442-443: –report-type analysis → generate_analysis_report path.	—	Pass	0.004s
Lines 344-345: –report-type analysis → generate_analysis_report path executed.	—	Pass	0.004s
Lines 438-439: –report-type basic → generate_basic_report path.	—	Pass	0.006s
Lines 338-339: –report-type basic → generate_basic_report path executed.	—	Pass	0.004s
Lines 340-341: –report-type detailed → generate_detailed_report path executed.	—	Pass	0.004s
Lines 346-347: unmapped report_type → generate_report(pages_to_include=args.pages).	—	Pass	0.004s
Lines 444-445: –report-type full (not basic/monitoring/analysis) → else branch.	—	Pass	0.005s
Lines 440-441: –report-type monitoring + timeseries → monitoring path.	—	Pass	0.005s
Lines 499-500: –report-type mortgage-focused + mortgage file → focused report.	—	Pass	0.006s
Lines 499 condition False (no mortgage_data) → falls to else branch	—	Pass	0.005s
Lines 497-498: –report-type property-only → generate_property_only_report.	—	Pass	0.005s

Test Description	Acceptance Criteria	Result	Time
Lines 501-502: –report-type risk-focused → generate_risk_focused_report.	—	Pass	0.007s
Lines 342-343: –report-type summary → generate_summary_report path executed.	—	Pass	0.004s
Lines 447-449: success-logging block runs after generation.	—	Pass	0.005s
Lines 506-508: success-logging block runs after a successful generation.	—	Pass	0.004s
reports.py lines 87-88: gaugehd file read error -¿ warning, continues.	`r.status_code == 200`	Pass	0.036s
reports.py lines 135-140: general Exception -¿ 500.	`r.status_code == 500; r.get_json()["status"] == "error"`	Pass	0.030s
reports.py lines 100-101: hazard file read error -¿ warning, continues.	`r.status_code == 200`	Pass	0.030s
reports.py lines 128-133: ImportError -¿ 500.	`r.status_code == 500; r.get_json()["status"] == "error"`	Pass	0.029s
reports.py lines 70-72: storm_responses loaded -¿ timeseries_data set.	`r.status_code == 200`	Pass	0.029s
reports.py lines 73-74: timeseries load failure -¿ warning, continues.	`r.status_code == 200`	Pass	0.030s
Field is dict[overall risk rating] [field='overall_risk_rating']	`field in m; isinstance(m[field], dict)`	Pass	0.000s
Field is dict[coverage] [field='test_coverage']	`field in m; isinstance(m[field], dict)`	Pass	0.000s
Field is list[assumptions] [field='assumptions']	`field in m; isinstance(m[field], list)`	Pass	0.000s
Field is list[downstream models] [field='downstream_models']	`field in m; isinstance(m[field], list)`	Pass	0.000s
Field is list[limitations] [field='limitations']	`field in m; isinstance(m[field], list)`	Pass	0.000s
Field is list[upstream models] [field='upstream_models']	`field in m; isinstance(m[field], list)`	Pass	0.000s
Field is string[category] [field='category']	`field in m; isinstance(m[field], str)` (+1 more)	Pass	0.000s
Field is string[description] [field='description']	`field in m; isinstance(m[field], str)` (+1 more)	Pass	0.000s
Field is string[name] [field='name']	`field in m; isinstance(m[field], str)` (+1 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Field is string[owner] [field='owner']	`field in m; isinstance(m[field], str)` (+1 more)	Pass	0.000s
Field is string[status] [field='status']	`field in m; isinstance(m[field], str)` (+1 more)	Pass	0.000s
Field is string[version] [field='version']	`field in m; isinstance(m[field], str)` (+1 more)	Pass	0.000s
Case insensitive	`resolve_model_id("GEV") == "MKM-GH-001"`	Pass	0.000s
Depth damage maps to dd	`resolve_model_id("depth_damage") == "MKM-DD-001"`	Pass	0.000s
Distribution maps to si	`resolve_model_id("distribution") == "MKM-SI-001"`	Pass	0.000s
Mkm prefix returned as is	`resolve_model_id("MKM-GH-001") == "MKM-GH-001"`	Pass	0.002s
Shorthand gev resolved	`resolve_model_id("gev") == "MKM-GH-001"`	Pass	0.000s
Shorthand intensity resolved	`resolve_model_id("intensity") == "MKM-SI-001"`	Pass	0.000s
Shorthand mortgage resolved	`resolve_model_id("mortgage") == "MKM-MP-001"`	Pass	0.000s
Shorthand prs resolved	`resolve_model_id("prs") == "MKM-PR-001"`	Pass	0.000s
Spatial maps to sp	`resolve_model_id("spatial") == "MKM-SP-001"`	Pass	0.000s
Unknown shorthand returned as is	`resolve_model_id("unknown_model") == "unknown_model"`	Pass	0.000s
If all prerequisites are fresh, returns empty list.	`result == []`	Pass	0.001s
Upstream never recorded but outputs exist -¿ skip it.	`"gauges" not in result`	Pass	0.000s
Upstream never recorded and outputs missing -¿ needs to run.	`"gauges" in result`	Pass	0.000s
Multiple targets: prerequisites are union of both.	`"gauges" in result; "counterparties" in result`	Pass	0.000s
Without data_dir, missing manifest entries -¿ prerequisite needed.	`"gauges" in result`	Pass	0.000s
Root steps (gauges) have no prerequisites.	`result == []`	Pass	0.000s
Upstream recorded but inputs stale -¿ needs to run.	`"properties" in result`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Target steps themselves should never appear in prerequisites.	"propertyts" not in result	Pass	0.000s
Prerequisites are returned in valid topological order.	result.index("gauges") < result.index("properties"); result.index("gauges") < result.index("gaugehd") (+1 more)	Pass	0.000s
Deep chain: if gauges missing, all downstream prereqs needed.	"gauges" in result; "properties" in result (+4 more)	Pass	0.001s
Unknown step returns empty list.	result == []	Pass	0.000s
Retention mentions reproducibility	'Principle 5/6' in texts; 'Reproducibility' in texts	Pass	0.000s
100yr greater than 10yr	i100 > i10	Pass	0.000s
Ordering 10 50 100	i10 < i50 < i100	Pass	0.000s
Result in bounds	d.params.min_intensity <= v <= d.params.max_intensity	Pass	0.000s
Storms per year affects result	many > few	Pass	0.000s
Returns dict with total storms	isinstance(result, dict); "total_storms" in result (+1 more)	Pass	0.002s
Returns elapsed s	"elapsed_s" in result; result["elapsed_s"] >= 0	Pass	0.004s
revalue_all(trades, None) loads market state internally.	len(enriched) == 1; enriched[0]['swap_id'] == 'PRS-TEST001' (+1 more)	Pass	0.003s
Combined mode with mortgage	isinstance(result, list) and len(result) > 0	Pass	0.001s
Empty property does not crash	isinstance(self._page().generate_elements({}), list)	Pass	0.000s
Has risk header	any("Risk" in t for t in texts)	Pass	0.000s
Has table	any(isinstance(e, Table) for e in result)	Pass	0.001s
Property only mode	isinstance(result, list) and len(result) > 0	Pass	0.000s
Returns list	isinstance(self._page().generate_elements(make_property)),	Pass	0.000s
Canal distance shown	isinstance(result, list)	Pass	0.000s
Empty property does not crash	isinstance(self._page().generate_elements({}), list)	Pass	0.000s
Flood risk type shown	isinstance(result, list)	Pass	0.000s
Has risk header	any("Risk" in t for t in texts)	Pass	0.001s
Has table	any(isinstance(e, Table) for e in result)	Pass	0.000s
Lake distance shown	isinstance(result, list)	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Last flood date shown	`isinstance(result, list)`	Pass	0.000s
No defence scheme	`isinstance(result, list)`	Pass	0.001s
No ea zone	`isinstance(result, list)`	Pass	0.000s
No last flood date	`isinstance(result, list)`	Pass	0.000s
Returns list	`isinstance(self._page().generate_elements(make_property()), ...`	Pass	0.001s
Line 257: non-numeric → delegates to _format_value.	`page._format_count(None) == "Not specified"`	Pass	0.000s
Line 256: numeric → comma-formatted.	`"1,234,567" in result`	Pass	0.000s
Line 239: non-numeric → delegates to _format_value.	`result == "Not specified"`	Pass	0.000s
Line 238: numeric → £ formatted.	`"£" in result; "1,234,567" in result`	Pass	0.000s
Line 251: non-numeric → delegates to _format_value.	`page._format_depth("N/A") == "N/A"`	Pass	0.000s
Line 250: numeric → formatted with 'm'.	`"1.50m" in result`	Pass	0.000s
Lines 204-215: camelCase → 'Camel Case'.	`"Risk" in result`	Pass	0.000s
Lines 207-213: abbreviation replacements applied.	`"ID" in result`	Pass	0.000s
Line 245: non-numeric → delegates to _format_value.	`page._format_percentage(None) == "Not specified"`	Pass	0.000s
Line 244: numeric → formatted with %.	`"%" in result; "12.3" in result`	Pass	0.000s
Line 222: False → 'No'.	`page._format_value(False) == "No"`	Pass	0.000s
Line 222: True → 'Yes'.	`page._format_value(True) == "Yes"`	Pass	0.000s
Lines 230-231: blank string → 'Not specified'.	`page._format_value(" ") == "Not specified"`	Pass	0.000s
Lines 226-227: float with fractional part → 2dp string.	`result == "3.14"`	Pass	0.000s
Lines 224-225: float with no fractional part → int string.	`page._format_value(5.0) == "5"`	Pass	0.000s
Lines 228-229: integer → plain string.	`page._format_value(42) == "42"`	Pass	0.000s
Lines 232-233: non-empty string → passed through.	`page._format_value("hello") == "hello"`	Pass	0.000s
Line 220: None → 'Not specified'.	`page._format_value(None) == "Not specified"`	Pass	0.000s
High risk	`classify_risk_level(3.0) == "HIGH"`	Pass	0.000s
Low risk	`classify_risk_level(0.3) == "LOW"`	Pass	0.000s
Medium risk	`classify_risk_level(1.0) == "MEDIUM"`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Minimal risk	`classify_risk_level(0.0) == "MINIMAL"`	Pass	0.000s
Get risk rating calculation	`response.status_code == 200; data["status"] == "success"` (+5 more)	Pass	0.002s
Risk rating clear override	`response.status_code == 200; rr["mrc_override"] is None` (+1 more)	Pass	0.005s
Risk rating override	`response.status_code == 200; rr["mrc_override"] == "Unacceptable"` (+3 more)	Pass	0.004s
Risk rating override invalid rating	`response.status_code == 400`	Pass	0.002s
Risk rating override requires reason	`response.status_code == 400`	Pass	0.002s
Line 210: inventory not found → 404.	`r.status_code == 404`	Pass	0.002s
Lines 126-134: _auto_select_pages returns known pages.	`isinstance(pages, list); len(pages) > 0`	Pass	0.001s
Empty gauge data	`RiskReportGenerator(output_dir).generate_basic_report(dat`	Pass	0.015s
Lines 162-164: page raises → continue to next.	`isinstance(elements, list)`	Pass	0.002s
Lines 148-149: unknown page → warning, skipped.	`isinstance(elements, list)`	Pass	0.001s
Line 265: get_page_categories.	`isinstance(cats, dict)`	Pass	0.001s
High risk portfolio report	`RiskReportGenerator(output_dir).generate_detailed_report(`	Pass	0.023s
Line 261: list_available_pages.	`isinstance(pages, list); len(pages) > 0`	Pass	0.001s
Minimal data report	`RiskReportGenerator(output_dir).generate_basic_report(mi`	Pass	0.014s
Missing mortgage data	`RiskReportGenerator(output_dir).generate_detailed_report(`	Pass	0.021s
Lines 60-62: output_dir=None → uses config.get_risk_reports_dir().	`gen.output_dir == risk_dir`	Pass	0.000s
Lines 270-336: validate_pages.	`"title" in valid; "nonexistent" in invalid`	Pass	0.001s
Analysis report generation	`report_path.exists(); report_path.stat().st_size > 0`	Pass	0.016s
Basic report generation	`report_path.exists(); report_path.suffix == ".pdf"` (+1 more)	Pass	0.021s
Custom filename	`report_path.name == custom_name; report_path.exists()`	Pass	0.016s
Custom page selection	`report_path.exists()`	Pass	0.012s
Detailed report generation	`report_path.exists(); report_path.stat().st_size > 0`	Pass	0.025s

Test Description	Acceptance Criteria	Result	Time
Summary report generation	`report_path.exists()`; `report_path.stat().st_size > 0`	Pass	0.009s
Categories dict initialized	`"overview" in gen.categories`; `"analysis" in gen.categories` (+1 more)	Pass	0.001s
Explicit output dir path	`isinstance(gen.output_dir, Path)`	Pass	0.001s
Explicit output dir string	`gen.output_dir == tmp_path`	Pass	0.001s
output_dir=None → falls back to config.get_reports_dir('risk').	`gen.output_dir == risk_dir`	Pass	0.001s
Output dir created	`new_dir.exists()`	Pass	0.001s
Pages dict initialized	`expected <= set(gen.pages.keys())`	Pass	0.001s
Get page categories	`isinstance(categories, dict)`; `key in categories`	Pass	0.001s
List available pages	`isinstance(pages, list)`; `key in pages`	Pass	0.001s
Validate pages invalid	`"title" in valid`; `"nonexistent_page" in invalid`	Pass	0.001s
Validate pages valid	`len(valid) == 2`; `len(invalid) == 0`	Pass	0.001s
Complete workflow all types	`p.exists()`; `p.stat().st_size > 0`	Pass	0.062s
Complete workflow basic	`report_path.exists()`; `report_path.suffix == ".pdf"` (+1 more)	Pass	0.016s
High risk label	`"High Risk" in str(elements)`	Pass	0.001s
Low risk label	`"Low Risk" in str(elements)`	Pass	0.000s
Moderate risk label	`"Moderate Risk" in str(elements)`	Pass	0.001s
Non numeric score formatted	`isinstance(elements, list)`	Pass	0.000s
Very high risk label	`"Very High Risk" in str(elements)`	Pass	0.003s
Clay soil adds 1	`result["score"] >= 1`	Pass	0.000s
Close river adds 1	`result["score"] >= 1`	Pass	0.000s
High flood risk adds 3	`result["score"] >= 3`	Pass	0.000s
High score level	`result["level"] == "High Risk"`	Pass	0.000s
Low elevation adds 2	`result["score"] >= 2`	Pass	0.000s
Low flood risk adds 1	`result["score"] >= 1`	Pass	0.000s
Low medium score level	`result["level"] in ("Medium Risk", "Low-Medium Risk")`	Pass	0.000s
Low risk level	`result["level"] == "Low Risk"`	Pass	0.000s
Medium flood risk adds 2	`result["score"] >= 2`	Pass	0.000s
Medium score level	`"Medium" in result["level"]` or `result["score"] >= 4`	Pass	0.000s
Moderate elevation adds 1	`result["score"] >= 1`	Pass	0.000s
No risk factors message	`"No major" in result["primary_factors"]`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Peat soil adds 2	`result["score"] >= 2`	Pass	0.000s
Returns all keys	`k in result`	Pass	0.000s
Very close river adds 2	`result["score"] >= 2`	Pass	0.000s
Very high flood risk adds 4	`result["score"] >= 4`	Pass	0.000s
Very high score level	`result["level"] == "Very High Risk"`	Pass	0.000s
Very low elevation adds 3	`result["score"] >= 3`	Pass	0.000s
Close 100 to 500m	`isinstance(result, list)`	Pass	0.001s
Good distance over 1000m	`isinstance(result, list)`	Pass	0.000s
Moderate 500 to 1000m	`isinstance(result, list)`	Pass	0.001s
Very close under 100m	`isinstance(result, list)`	Pass	0.001s
GET /governance/data-lineage gracefully handles _check_staleness error.	`resp.status_code == 200`; `data["pipeline_steps"] == []`	Pass	0.001s
GET /governance/data-lineage/staleness returns 500 on error.	`resp.status_code == 500`; `data["status"] == "error"`	Pass	0.001s
GET /governance/data-lineage/trace returns 500 on _trace_data error.	`resp.status_code == 500`; `data["status"] == "error"` (+1 more)	Pass	0.001s
Appends started header	`any("=" * 10 in ln for ln in lines)`; `any("Audit run complete" in ln for ln in lines)`	Pass	0.001s
Error line appended on exception	`any("ERROR" in ln for ln in lines)`	Pass	0.001s
Handles exception gracefully	`tr_mod._running is False`	Pass	0.001s
Sets running false on completion	`tr_mod._running is False`	Pass	0.001s
Setting _batch_job to None mid-run stops processing.	`call_count == 1`; `cl_mod._batch_job is None`	Pass	0.000s
Failed gauge training is recorded as 'failed' in results.	`cl_mod._batch_job["status"] == "complete"`; `results[0]["status"] == "trained"` (+2 more)	Pass	0.001s
Batch trains gauges and records timings.	`cl_mod._batch_job["status"] == "complete"`; `cl_mod._batch_job["completed"] == 2` (+5 more)	Pass	0.008s
Each gauge has gauge_id, gauge_name, p_flood, p_flood_pct, threshold,	`'gauge_id' in g`; `'gauge_name' in g` (+8 more)	Pass	0.028s
Response has gauges_severe as list of gauge_ids; for STORM_PORT_SEVERE	`'gauges_severe' in data`; `isinstance(data['gauges_severe'], list)` (+2 more)	Pass	0.024s

Test Description	Acceptance Criteria	Result	Time
gauges_warning list includes GAUGE-002.	'gauges_warning' in data; isinstance(data['gauges_warning'], list) (+2 more)	Pass	0.024s
Response has 'gauges' list.	'gauges' in data; isinstance(data['gauges'], list)	Pass	0.033s
Response has portfolio_stress_pnl, portfolio_mtm, num_gauges.	'portfolio_stress_pnl' in data; 'portfolio_mtm' in data (+1 more)	Pass	0.025s
POST with non-existent storm_id returns 404.	resp.status_code == 404	Pass	0.002s
POST with no body or empty storm_id returns 400.	resp.status_code == 400	Pass	0.000s
POST STORM-PORT-001 returns 200 with status=success.	resp.status_code == 200; data['status'] == 'success'	Pass	0.022s
Each hour has water_level, p_flood, portfolio fields.	'hour' in h; 'water_level' in h (+4 more)	Pass	0.034s
Non-existent storm returns 404.	resp.status_code == 404	Pass	0.001s
Severe storm triggers knock-out (P(flood)=100%).	data['hourly'][first_breach]['p_flood'] == 1.0; h['p_flood'] is None	Pass	0.213s
Missing gauge_id or storm_id returns 400.	resp.status_code == 400	Pass	0.000s
No JSON body returns error.	resp.status_code in (400, 500)	Pass	0.001s
Response includes portfolio-level stress summary.	'num_trades' in summary; 'total_notional' in summary (+5 more)	Pass	0.033s
Response includes probability surface grid.	'probability_surface' in data; 'water_levels' in surface (+7 more)	Pass	0.029s
POST returns 168-hour forecast array.	resp.status_code == 200; data['status'] == 'success' (+1 more)	Pass	0.034s
Response includes per-trade summary.	'trades' in data; len(data['trades']) == 2 (+4 more)	Pass	0.035s
Surface water levels cap at severe – no rows above severe.	max(surface['water_levels']) <= severe	Pass	0.033s
Surface must cover full 168h storm horizon regardless of KO.	max(surface['hours']) >= 164	Pass	0.036s
Empty gauge_id returns 400.	resp.status_code == 400	Pass	0.000s
No open trades at gauge returns 404.	resp.status_code == 404	Pass	0.002s
Null JSON body returns 400.	resp.status_code == 400	Pass	0.000s
Gaugets file present uses hydrograph built from real readings.	resp.status_code == 200; data['status'] == 'success'	Pass	0.034s

Test Description	Acceptance Criteria	Result	Time
Already running message	`data["status"] == "running"`; `"message" in data` (+1 more)	Pass	0.000s
Already running returns 202	`resp.status_code == 202`	Pass	0.000s
Response has started status	`data["status"] == "started"`	Pass	0.001s
Starts run returns 202	`resp.status_code == 202`	Pass	0.001s
Empty running pnl returns drawing	`isinstance(result, Drawing)`	Pass	0.000s
All positive	`np.all(M > 0.0)`	Pass	0.000s
Empirical pairwise correlation approximately matches model correlation for large samples.	`abs(emp_corr[i, j] - model_corr[i, j]) < 0.15`	Pass	0.006s
Higher intensity -¿ longer range -¿ more spatially coherent (lower variance).	`np.mean(vars_extreme) < np.mean(vars_moderate)`	Pass	0.009s
Normalisation ensures catchment mean is preserved.	`M.mean() == pytest.approx(1.0, abs=1e-10)`	Pass	0.000s
Pre-normalisation, E[M] = exp(sigma^2/2) / exp(sigma^2/2) = 1 theoretically.	`np.mean(pre_norms) == pytest.approx(1.0, abs=0.05)`	Pass	0.002s
Reproducible with seed	`np.allclose(M1, M2)`	Pass	0.000s
Shape	`M.shape == (small_model.n_gauges,)`	Pass	0.000s
When rng is None, falls back to np.random.randn (line 251).	`M.shape == (model.n_gauges,)`; `M.mean() == pytest.approx(1.0, abs=1e-10)` (+1 more)	Pass	0.000s
Catastrophic weights: 15% doublet, 45% cluster, 40% persistent.	`doublets < 150`	Pass	0.003s
Defined categories[catastrophic] [cat='catastrophic']	`result in [SequenceType.DOUBLET, SequenceType.CLUSTER, Se...`	Pass	0.000s
Defined categories[extreme] [cat='extreme']	`result in [SequenceType.DOUBLET, SequenceType.CLUSTER, Se...`	Pass	0.000s
Defined categories[moderate] [cat='moderate']	`result in [SequenceType.DOUBLET, SequenceType.CLUSTER, Se...`	Pass	0.000s
Defined categories[severe] [cat='severe']	`result in [SequenceType.DOUBLET, SequenceType.CLUSTER, Se...`	Pass	0.000s
Returns sequence type	`isinstance(result, SequenceType)`; `result in [SequenceType.DOUBLET, SequenceType.CLUSTER, Se...`	Pass	0.000s
Unknown category uses DEFAULT_TYPE_WEIGHTS.	`isinstance(result, SequenceType)`	Pass	0.000s
Lines 122-123: 30% chance subsequent storm goes down.	`below_base > 10`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Storm index 0 has downward variation (up to -20%).	`all(r <= base + 0.001 for r in results)`	Pass	0.001s
First storm returns positive	`result > 0`	Pass	0.000s
Result should be within +/-20% of base for first storm.	`all(r >= lower - 0.001 for r in results); all(r <= upper + 0.001 for r in results)`	Pass	0.001s
70% chance of maintaining/increasing subsequent storm intensity.	`above_base > 100`	Pass	0.000s
Subsequent storm positive	`result > 0`	Pass	0.000s
max(0.1, ...) prevents zero or negative intensity.	`result >= 0.1`	Pass	0.000s
Different seeds different output	`d1.sample(10) != d2.sample(10)`	Pass	0.000s
Sample n returns correct length	`len(samples) == 100`	Pass	0.000s
Sample single	`d.params.min_intensity <= v <= d.params.max_intensity`	Pass	0.000s
Samples within bounds	`all(d.params.min_intensity <= s <= d.params.max_intensity...`	Pass	0.001s
Seeded reproducibility	`s1 == s2`	Pass	0.000s
Bool passed through	`result["flag"] is True`	Pass	0.000s
Empty params	`result == {}`	Pass	0.000s
Float passed through	`result["rate"] == 0.05`	Pass	0.000s
Int passed through	`result["n"] == 42`	Pass	0.000s
Long list summarised	`isinstance(result["items"], str); "15 items" in result["items"]`	Pass	0.000s
None passed through	`result["x"] is None`	Pass	0.000s
Object converted to string	`result["obj"] == "my_object"`	Pass	0.000s
Short list converted to str	`isinstance(result["items"], list); len(result["items"]) == 3`	Pass	0.000s
String passed through	`result["name"] == "test"`	Pass	0.000s
Tuple treated like list	`isinstance(result["t"], list)`	Pass	0.000s
Identity no antecedent	`compute_saturation_factor(0.0) == 1.0`	Pass	0.000s
Known value	`compute_saturation_factor(100.0) == pytest.approx(expected)`	Pass	0.000s
Monotonic	`1.0 < s1 < s2 < s3`	Pass	0.000s
Negative precip	`compute_saturation_factor(-10.0) == 1.0`	Pass	0.000s
Saves and returns response	`resp is not None`	Pass	0.056s
Returns false on oserror	`result is False; _save_audit_log([]) is False` (+1 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Saves entries	result is True; json.loads(p.read_text())[0]["event"] == "created"	Pass	0.003s
Returns false on oserror	_save_bcbs239({}) is False; _save_raci({}) is False (+2 more)	Pass	0.000s
Saves data	_save_bcbs239({"principles": []}) is True; _save_raci({"roles": []}) is True	Pass	0.000s
Returns false on oserror	_save_bcbs239({}) is False; _save_raci({}) is False (+2 more)	Pass	0.000s
Saves bibliography	_save_bibliography({"references": [], "categories": []}) ...	Pass	0.000s
Save error is reraised	—	Pass	0.002s
Creates gauge file	(tmp_path / "gaugets" / "GAUGE-001.json").exists()	Pass	0.000s
If gauge file already exists, merge storm_responses into it.	"flood_simulation" in data; "storm_responses" in data	Pass	0.002s
Multiple gauges	(gaugets_dir / "GAUGE-001.json").exists(); (gaugets_dir / "GAUGE-002.json").exists()	Pass	0.000s
Saved file has responses	"storm_responses" in data; len(data["storm_responses"]["responses"]) == 3	Pass	0.001s
Returns false on oserror	_save_bcbs239({}) is False; _save_raci({}) is False (+2 more)	Pass	0.000s
Saves documents	_save_gov_documents([{"doc_id": "D1"}]) is True	Pass	0.000s
Creates file	output_path.exists(); result == output_path	Pass	0.000s
Creates parent dirs	output_path.exists()	Pass	0.000s
File has hazard curves	"GAUGE-001" in data["hazard_curves"]	Pass	0.001s
File has metadata	"metadata" in data; data["metadata"]["catchment_id"] == "thames"	Pass	0.001s
Metadata merged	data["metadata"]["num_storms"] == 100	Pass	0.001s
Returns false on oserror	result is False; _save_audit_log([]) is False (+1 more)	Pass	0.000s
Saves data	result is True; json.loads(p.read_text())["models"] == []	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Creates parent dirs	`out.exists()`	Pass	0.008s
Load missing file raises	—	Pass	0.000s
Load returns correct count	`len(loaded) == len(small_batch)`	Pass	0.020s
Load wrong schema raises	—	Pass	0.002s
Loaded sequences still valid	`result["invalid"] == 0`	Pass	0.021s
Round trip num storms	`orig.num_storms == back.num_storms`	Pass	0.018s
Round trip precipitation	`abs(orig.total_precipitation_mm - back.total_precipitatio...`	Pass	0.022s
Round trip sequence type	`orig.sequence_type == back.sequence_type`	Pass	0.019s
Round trip storm fields	`os_.storm_index == bs_.storm_index; abs(os_.duration_hours - bs_.duration_hours) < 0.01` (+1 more)	Pass	0.020s
Round trip total duration	`abs(orig.total_duration_hours - back.total_duration_hours...`	Pass	0.019s
Save creates file	`out.exists()`	Pass	0.008s
Schema version in file	`raw["schema_version"] == SCHEMA_VERSION`	Pass	0.020s
Save cleans up on failure	`len(list(tmp_path.glob("*.tmp"))) == 0`	Pass	0.001s
Save creates file	`result == data`	Pass	0.058s
Save overwrites existing	`result == new_data`	Pass	0.016s
Returns false on oserror	`result is False; _save_audit_log([]) is False` (+1 more)	Pass	0.000s
Saves meetings	`result is True; json.loads(p.read_text())[0]["id"] == "M1"`	Pass	0.003s
Returns false on oserror	`_save_bcbs239({}) is False; _save_raci({}) is False` (+2 more)	Pass	0.000s
Saves data	`_save_bcbs239({"principles": []}) is True; _save_raci({"roles": []}) is True`	Pass	0.000s
Creates file	`out.exists()`	Pass	0.000s
Duration stats present	`d["min"] > 0; d["max"] <= 156` (+1 more)	Pass	0.001s
Num sequences matches	`data["num_sequences"] == len(small_batch)`	Pass	0.001s
Precipitation stats present	`p["min"] > 0; p["max"] >= p["min"]` (+1 more)	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Schema version	`data["schema_version"] == SCHEMA_VERSION`	Pass	0.002s
Type counts sum to total	`sum(data["sequence_type_counts"].values()) == len(small_b...`	Pass	0.001s
Type fractions sum to one	`abs(total_frac - 1.0) < 0.001`	Pass	0.001s
Above max depth returns one	`scalar_depth_damage(100.0) == pytest.approx(1.0)`	Pass	0.000s
Between control points interpolated	`0.25 < d < 0.40`	Pass	0.000s
Damage between 0 and 1	`0 <= scalar_depth_damage(depth) <= 1.0`	Pass	0.000s
Half meter damage	`d == pytest.approx(DAMAGE_POINTS[2], abs=1e-6)`	Pass	0.000s
Max depth returns one	`scalar_depth_damage(DEPTH_POINTS[-1]) == pytest.approx(1.0)`	Pass	0.000s
Monotonically increasing	`all(damages[i] <= damages[i + 1] for i in range(len(damag...`	Pass	0.000s
Negative depth returns zero	`scalar_depth_damage(-1.0) == 0.0; scalar_depth_damage(-100.0) == 0.0`	Pass	0.000s
One meter damage	`d == pytest.approx(DAMAGE_POINTS[3], abs=1e-6)`	Pass	0.000s
Returns float	`isinstance(scalar_depth_damage(0.75), float)`	Pass	0.000s
Small depth returns small damage	`d == pytest.approx(DAMAGE_POINTS[1], abs=1e-9)`	Pass	0.000s
Zero depth returns zero	`scalar_depth_damage(0.0) == 0.0`	Pass	0.000s
Every control point must return a value in [0, 1].	`0.0 <= val <= 1.0`	Pass	0.000s
Very small positive depth (between 0 and first control point).	`0.0 < d < 0.05`	Pass	0.000s
'base_level_m' field — used in Stress tab storm info bar.	`'base_level_m' in s`	Pass	0.004s
'duration_hours' field — used in Stress tab storm info bar.	`'duration_hours' in s`	Pass	0.003s
'effective_precipitation_mm' field — shown in storm row.	`'effective_precipitation_mm' in s`	Pass	0.004s
'gauges_severe' field — severity indicator in storm row.	`'gauges_severe' in s`	Pass	0.003s
'intensity_category' field — colour-coded in storm row.	`'intensity_category' in s`	Pass	0.004s
'level_change_m' field — used in Stress tab storm info bar.	`'level_change_m' in s`	Pass	0.003s

Test Description	Acceptance Criteria	Result	Time
'max_trigger' field — used for trigger colour coding in Historical list.	`'max_trigger' in s`	Pass	0.003s
'name' field — shown in storm row title.	`'name' in s`	Pass	0.004s
'peak_level_m' field — primary sort key and shown in Trading Desk label.	`'peak_level_m' in s`	Pass	0.004s
'peak_position' field — used in hydrograph synthesis.	`'peak_position' in s`	Pass	0.004s
Hour label element must be present to display current hour.	`'sp-sim-hour-label' in sim_js`	Pass	0.000s
Scrubber element must have the expected DOM ID.	`'sp-sim-scrubber' in sim_js`	Pass	0.000s
scrubber.max must equal STORM_SIMULATION_HOURS - 1 (0-indexed).	`f"scrubber.max = '{expected}'" in sim_js`	Pass	0.000s
Regression guard: the old wrong value '59' must not be present.	`"scrubber.max = '59'" not in sim_js`	Pass	0.000s
scrubber.min must be '0' so the timeline starts at hour 0.	`"scrubber.min = '0'" in sim_js`	Pass	0.000s
Scrubber element must be an input[type=range].	`"scrubber.type = 'range'" in sim_js`	Pass	0.000s
Heading	`_has_paragraph_containing(els, '7. Property Flood Time Se...'; _has_paragraph_containing(els, '9. Counterparties')` (+2 more)	Pass	0.000s
Kv table	`_has_table(els)`	Pass	0.000s
Returns list	`isinstance(els, list); isinstance(els, list)` (+2 more)	Pass	0.000s
Cover healthy	`len(story) > 0`	Pass	0.000s
Cover issues	`len(story) > 0`	Pass	0.000s
Exec summary healthy	`len(story) > 0`	Pass	0.001s
Exec summary issues	`len(story) > 0`	Pass	0.001s
Quality metrics healthy	`len(story) > 0`	Pass	0.000s
Quality metrics with stale details	`len(story) > 0`	Pass	0.001s
Reconciliation no results	`len(story) > 0`	Pass	0.000s
Reconciliation with failures	`len(story) > 0`	Pass	0.000s
Reconciliation with results	`len(story) > 0`	Pass	0.000s
Remediation compliant	`len(story) > 0`	Pass	0.000s
Remediation with issues	`len(story) > 2`	Pass	0.001s
Retention policy	`len(story) > 0`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Source documentation	len(story) > 0	Pass	0.001s
Step with no parameters should still render.	len(story) > 0	Pass	0.000s
Topology	len(story) > 0	Pass	0.000s
Handles Counterparty key instead of CounterpartySet.	_has_table(els)	Pass	0.000s
Empty message	_has_paragraph_containing(els, 'No counterparties generat...	Pass	0.000s
Heading	_has_paragraph_containing(els, '7. Property Flood Time Se...; _has_paragraph_containing(els, '9. Counterparties') (+2 more)	Pass	0.000s
Returns list	isinstance(els, list); isinstance(els, list) (+2 more)	Pass	0.000s
Table present	_has_table(els)	Pass	0.000s
Heading	_has_paragraph_containing(els, '3. Mortgages'); _has_paragraph_containing(els, '4. Historical Gauge Data') (+2 more)	Pass	0.000s
No baselines message	_has_paragraph_containing(els, '5 gaugehd files')	Pass	0.000s
Range computed	_has_paragraph_containing(els, 'Seasonal range')	Pass	0.000s
Returns list	isinstance(els, list); isinstance(els, list) (+4 more)	Pass	0.000s
Seasonal averages	_has_paragraph_containing(els, 'Avg winter baseline')	Pass	0.000s
Table when baselines	_has_table(els)	Pass	0.000s
Empty gauges message	_has_paragraph_containing(els, 'No gauges generated'); not _has_table(els)	Pass	0.000s
Multiple gauges	_has_paragraph_containing(els, '2 gauges')	Pass	0.000s
Returns list	isinstance(els, list); isinstance(els, list) (+4 more)	Pass	0.000s
Section heading	_has_paragraph_containing(els, '1. Gauge Network')	Pass	0.000s
Summary line counts	_has_paragraph_containing(els, '5 time series'); _has_paragraph_containing(els, '3 historical daily')	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Table present	`_has_table(els)`; `_has_table(els)`	Pass	0.000s
Handles exceedance_probabilities + return_levels format.	`_has_table(els)`	Pass	0.000s
Non-dict gauge curve entry should produce dashes, not crash.	`_has_table(els)`	Pass	0.000s
Heading	`_has_paragraph_containing(els, '3. Mortgages')`; `_has_paragraph_containing(els, '4. Historical Gauge Data')` (+2 more)	Pass	0.000s
No curves message	`_has_paragraph_containing(els, 'No hazard curves available')`	Pass	0.000s
Returns list	`isinstance(els, list)`; `isinstance(els, list)` (+4 more)	Pass	0.000s
Table with curves	`_has_table(els)`; `_has_paragraph_containing(els, '1 property hazard curves')`	Pass	0.000s
Count in summary	`_has_paragraph_containing(els, '2 mortgages linked')`	Pass	0.000s
Empty mortgages message	`_has_paragraph_containing(els, 'No mortgages generated')`	Pass	0.000s
Heading	`_has_paragraph_containing(els, '3. Mortgages')`; `_has_paragraph_containing(els, '4. Historical Gauge Data')` (+2 more)	Pass	0.000s
Returns list	`isinstance(els, list)`; `isinstance(els, list)` (+4 more)	Pass	0.000s
Table present	`_has_table(els)`; `_has_table(els)`	Pass	0.000s
Empty properties message	`_has_paragraph_containing(els, 'No properties generated')`	Pass	0.000s
Heading present	`_has_paragraph_containing(els, '2. Properties')`	Pass	0.000s
Property count in summary	`_has_paragraph_containing(els, '2 properties')`	Pass	0.000s
Returns list	`isinstance(els, list)`; `isinstance(els, list)` (+4 more)	Pass	0.000s
Table with properties	`_has_table(els)`	Pass	0.000s
Heading	`_has_paragraph_containing(els, '7. Property Flood Time Se...`; `_has_paragraph_containing(els, '9. Counterparties')` (+2 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
No phc section when empty	`not _has_paragraph_containing(els, '8. Property Hazard')`	Pass	0.000s
Phc section added	`_has_paragraph_containing(els, '8. Property Hazard Curves'); _has_paragraph_containing(els, '2 property hazard curves')`	Pass	0.000s
Returns list	`isinstance(els, list); isinstance(els, list)` (+2 more)	Pass	0.000s
Heading	`_has_paragraph_containing(els, '3. Mortgages'); _has_paragraph_containing(els, '4. Historical Gauge Data')` (+2 more)	Pass	0.000s
Kv table present	`_has_table(els)`	Pass	0.000s
Returns list	`isinstance(els, list); isinstance(els, list)` (+4 more)	Pass	0.000s
With full summary	`_has_table(els)`	Pass	0.000s
If input_dir is a real path, data size is appended.	`_has_table(els)`	Pass	0.001s
Heading	`_has_paragraph_containing(els, '7. Property Flood Time Se...; _has_paragraph_containing(els, '9. Counterparties')` (+2 more)	Pass	0.001s
Kv table present	`_has_table(els)`	Pass	0.001s
Returns list	`isinstance(els, list); isinstance(els, list)` (+2 more)	Pass	0.001s
All builders exported	`hasattr(sections_mod, name); callable(getattr(sections_mod, name))`	Pass	0.000s
All list	`len(sections_mod.__all__) == 12`	Pass	0.000s
Case insensitive	`seq_type_color('DOUBLET') == colors.lightyellow`	Pass	0.000s
Cluster	`seq_type_color('cluster') == colors.lightsalmon`	Pass	0.000s
Doublet	`seq_type_color('doublet') == colors.lightyellow`	Pass	0.000s
Isolated	`seq_type_color('isolated') == colors.lightblue`	Pass	0.000s
None returns white	`seq_type_color(None) == colors.white`	Pass	0.000s
Persistent	`seq_type_color('persistent') == colors.mistyrose`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Unknown returns white	`seq_type_color('unknown') == colors.white`	Pass	0.000s
Response declares the max_depth damage model.	`data["damage_model"] == "max_depth"`	Pass	0.005s
Each property entry includes num_sequence_floods.	`prop_a["num_sequence_floods"] >= 1`	Pass	0.005s
Response includes num_storms_in_sequence.	`"num_storms_in_sequence" in data; data["num_storms_in_sequence"] >= 1`	Pass	0.005s
Damage is max depth, not 0.10 + 0.18 = 0.28.	`prop_a["damage_ratio"] < 0.20`	Pass	0.005s
PROP-A: second storm (0.65m) is worst — sequence damage uses 0.18 ratio.	`prop_a["damage_ratio"] == pytest.approx(0.18); prop_a["damage_amount"] == pytest.approx(350_000 * 0.18)` (+1 more)	Pass	0.005s
PROP-B: first storm depth=0 so excluded, second storm 0.55m -¿ included.	`"PROP-B" in prop_ids`	Pass	0.005s
PROP-C is flooded by STORM-other, not STORM-test1 — must not appear.	`"PROP-C" not in prop_ids`	Pass	0.005s
Sequence total damage ¡ sum of individual storm damages (not additive).	`seq_damage < dmg_d1a + dmg_d1b`	Pass	0.006s
Sequence endpoint options	`r.status_code == 200`	Pass	0.000s
Sequence endpoint returns 200	`r.status_code == 200; data["status"] == "success"`	Pass	0.005s
Sequence id in response	`data["sequence_id"] == "STORM-test1"`	Pass	0.004s
Unknown sequence returns 404	`r.status_code == 404`	Pass	0.005s
Worst storm (17% damage on 300k) -¿ 249k ¡ 280k outstanding -¿ negative equity.	`data["portfolio"]["mortgages_in_negative_equity"] == 1; prop["negative_equity"] is True`	Pass	0.004s
Sequence num floods is four	`data["properties"][0]["num_sequence_floods"] == 4`	Pass	0.004s
Sequence damage uses storm p3 (1.8m, ratio=0.17), not p1 or p2.	`prop["worst_storm_id"] == "STORM-p3"; prop["damage_ratio"] == pytest.approx(0.17)`	Pass	0.004s
Querying seqA1 does not return seqB damage.	`dmg_a != dmg_b; dmg_a != dmg_a + dmg_b`	Pass	0.003s
Seqa first storm returns only its damage	`prop["damage_amount"] == pytest.approx(400_000 * 0.12)`	Pass	0.004s

Test Description	Acceptance Criteria	Result	Time
Seqb storm returns only seqb damage	`prop["damage_amount"] == pytest.approx(400_000 * 0.06)`	Pass	0.003s
Known sequence returns success	`r.status_code == 200; r.get_json()["status"] == "success"`	Pass	0.004s
No pts dir returns 404	`r.status_code == 404; r.get_json()["status"] == "error"` (+2 more)	Pass	0.000s
Options returns ok	`r.status_code == 200; r.status_code == 200`	Pass	0.000s
Properties sorted by damage descending	`amounts == sorted(amounts, reverse=True); amounts == sorted(amounts, reverse=True)`	Pass	0.004s
Property entry has financial fields	`"property_id" in prop; "property_value" in prop` (+9 more)	Pass	0.004s
Property entry has num sequence floods	`"num_sequence_floods" in prop; prop["num_sequence_floods"] >= 1`	Pass	0.003s
Property entry has worst storm id	`"worst_storm_id" in prop`	Pass	0.003s
Response has damage model	`r.get_json()["damage_model"] == "max_depth"`	Pass	0.004s
Response has num storms	`"num_storms_in_sequence" in r.get_json(); r.get_json()["num_storms_in_sequence"] >= 1`	Pass	0.005s
Response has portfolio	`"portfolio" in data; "total_properties" in port` (+4 more)	Pass	0.004s
Response has properties list	`"properties" in data; len(data["properties"]) >= 1` (+2 more)	Pass	0.003s
Response has sequence id	`r.get_json()["sequence_id"] == seq_id`	Pass	0.003s
Unknown sequence returns 404	`r.status_code == 404; r.get_json()["status"] == "error"`	Pass	0.004s
Describe returns string	`isinstance(desc, str); "base_mean" in desc.lower() or "Base mean" in desc`	Pass	0.073s
Params roundtrip	`recovered.base_mean == 52; recovered.name == "custom"`	Pass	0.000s
Tail probability matches internal	`d.to_dict()['tail_probability'] == pytest.approx(d._tail_...`	Pass	0.000s
To dict keys	`'parameters' in result; 'tail_probability' in result`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Returns 404 when file missing	`r.status_code in (200, 404, 500)`	Pass	0.001s
Lines 33-35: send_file when HTML file actually exists.	`r.status_code in (200, 404, 500)`	Pass	0.030s
Returns json error or html	`r.status_code in (200, 404, 500)`	Pass	0.001s
No coverage dir returns 404	`r.status_code == 404`	Pass	0.000s
With coverage index returns 200	`r.status_code == 200`	Pass	0.002s
The response body is valid JSON with an error status.	`r.status_code in (500, 503)`; `data is not None` (+1 more)	Pass	0.001s
Without gauge.json, visualization fails with 500.	`r.status_code in (500, 503)`	Pass	0.001s
No gauge data returns error status	`data["status"] == "error"`	Pass	0.001s
Lines 97-99: exception in serve_visualization_for_catchment → 500.	`r.status_code in (302, 500)`	Pass	0.027s
Matching is case-insensitive.	`r.status_code == 302`	Pass	0.000s
'thames' catchment redirects to main visualization route.	`r.status_code == 302`	Pass	0.000s
Unknown catchment returns 400	`r.status_code == 400`; `data["status"] == "error"` (+1 more)	Pass	0.000s
Unsupported catchment message	`r.status_code == 400`; `"thames" in r.get_json()["message"].lower()`	Pass	0.000s
Lines 59-65: ImportError → 503.	`r.status_code in (500, 503)`	Pass	0.029s
Lines 50-57: mock TCEventVisualization to return existing file.	`r.status_code in (200, 500)`	Pass	0.029s
main() calls app.run() with configured host/port/debug.	`len(run_calls) == 1`; `run_calls[0]["host"] == config.SERVER_HOST` (+1 more)	Pass	0.027s
server.main() is callable (test by importing, not running).	`callable(main)`	Pass	0.000s
Simulate what routes/gauges.py:generate_report() does.	`report_path.exists()`; `report_path.suffix == '.pdf'` (+1 more)	Pass	0.024s
Test with actual Thames gauge data if available.	`report_path.exists()`; `report_path.stat().st_size > 5000`	Pass	0.029s
Simulate what routes/properties.py:generate_report() does.	`report_path.exists()`; `report_path.suffix == '.pdf'` (+1 more)	Pass	0.028s
Test with actual Thames portfolio data if available.	`report_path.exists()`; `report_path.stat().st_size > 10000`	Pass	0.042s
Case insensitive thames	`r.status_code == 200`	Pass	0.000s
Missing catchment id returns 400	`r.status_code == 400`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
No json returns 400	`r.status_code in (400, 415, 500)`	Pass	0.000s
Line 64: non-empty body but null/empty catchment → 400.	`r.status_code == 400; "Catchment ID is required" in data["message"]`	Pass	0.000s
Options returns 200	`r.status_code == 200`	Pass	0.000s
Unknown catchment returns 400	`r.status_code == 400; data["status"] == "error"`	Pass	0.000s
Valid thames returns success	`r.status_code == 200; data["status"] == "success"` (+2 more)	Pass	0.000s
Existing mortgage dict updated	`data["Mortgage"]["Header"]["ExistingField"] == "keep"; data["Mortgage"]["Header"]["MortgageID"] == "MORT-new"`	Pass	0.000s
Initialises header key when absent	`"Header" in data["Mortgage"]`	Pass	0.000s
Initialises mortgage key when absent	`"Mortgage" in data`	Pass	0.000s
Missing property id key defaults empty string	`data["Mortgage"]["Header"]["PropertyID"] == ""`	Pass	0.000s
Sets catchment id	`data["Mortgage"]["Header"]["CatchmentID"] == "thames"`	Pass	0.000s
Sets loan details fields	`loan["OriginalLoanAmount"] == fd["loan_amount"]; loan["CurrentBalance"] == fd["current_balance"]` (+3 more)	Pass	0.000s
Sets mortgage id	`data["Mortgage"]["Header"]["MortgageID"] == "MORT-abc12"`	Pass	0.000s
Sets property id in header	`data["Mortgage"]["Header"]["PropertyID"] == "PROP-xyz"`	Pass	0.000s
Catchment id set in attrs	`data["PropertyHeader"]["PropertyAttributes"]["CatchmentID"]`	Pass	0.000s
Elevation sets ground level	`loc["RiskAssessment"]["GroundLevelMeters"] == pytest.appr...`	Pass	0.000s
Empty reference gauge indices no ref key	`"ReferenceGauges" not in data.get("PropertyHeader", {})`	Pass	0.000s
Lat lon set in location	`loc["LatitudeDegrees"] == pytest.approx(51.45); loc["LongitudeDegrees"] == pytest.approx(-0.31)`	Pass	0.000s
No elevation in location does not set ground level	`"RiskAssessment" not in loc`	Pass	0.000s
No reference gauge indices no ref key	`"ReferenceGauges" not in data.get("PropertyHeader", {})`	Pass	0.000s
Property id set in attrs	`data["PropertyHeader"]["PropertyAttributes"]["PropertyID"]`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Reference gauges fallback when not in map	`ref == ["GAUGE-001", "GAUGE-002", "GAUGE-003"]`	Pass	0.000s
Reference gauges set from gauge id map	`ref == ["GAUGE-abc", "GAUGE-def", "GAUGE-ghi"]`	Pass	0.076s
Adds all components	`mock_html.add_child.call_count > 0`	Pass	0.076s
Copyright element added	`len(calls) > 0`	Pass	0.075s
Returns self	`result is mgr`	Pass	0.077s
Body style font size	`styled.body_style.fontSize == 9`	Pass	0.000s
Kv style font size	`styled.kv_style.fontSize == 9`	Pass	0.000s
Kv style left indent	`styled.kv_style.leftIndent == 12`	Pass	0.000s
Required keys	`key in styles`	Pass	0.000s
Returns stylesheet	`isinstance(styles, StyleSheet1)`	Pass	0.000s
Section style font size	`styled.section_style.fontSize == 13`	Pass	0.000s
Section style text color	`styled.section_style.textColor == DARK_BLUE`	Pass	0.000s
Styles are paragraph styles	`isinstance(styles[key], ParagraphStyle)`	Pass	0.000s
Subsection style font size	`styled.subsection_style.fontSize == 10`	Pass	0.000s
Subsection style text color	`styled.subsection_style.textColor == BLUE`	Pass	0.000s
Subtitle style font size	`styled.subtitle_style.fontSize == 10`	Pass	0.000s
Subtitle style text color	`styled.subtitle_style.textColor == GREY`	Pass	0.000s
Title style font name	`styled.title_style.fontName == 'Helvetica-Bold'`	Pass	0.000s
Title style font size	`styled.title_style.fontSize == 18`	Pass	0.000s
Title style text color	`styled.title_style.textColor == DARK_BLUE`	Pass	0.000s
In shd mode the property elevation should match the gauge,	`result is not None; data["elevation_m"] <= 4.0`	Pass	0.165s
In she mode all gauge distances should be 0.	`result is not None; ng["distance_m"] == 0.0`	Pass	0.166s
Mode shd should write PROP-*.json into propertytsd/ not propertyts/.	`gen._MODE_DIRS["shd"] == "propertytsd"`	Pass	0.000s
She writes to propertytse dir	`gen._MODE_DIRS["she"] == "propertytse"`	Pass	0.164s
All defined categories	`isinstance(result, bool)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Catastrophic has 85% chance — over many trials, majority should be True.	`sum(results) > 140`	Pass	0.000s
Minimal has 5% chance — most should be False.	`sum(results) < 30`	Pass	0.000s
Returns bool	`isinstance(result, bool)`	Pass	0.000s
Unknown category uses default 0.15 probability.	`50 < sum(results) < 350`	Pass	0.001s
sp_sim.get_js() must return a non-empty string.	`len(sim_js) > 100`	Pass	0.000s
spSeekTo() must be defined — called by scrubber oninput.	`'spSeekTo' in sim_js`	Pass	0.000s
spSetSpeed() must be defined — called by speed buttons.	`'spSetSpeed' in sim_js`	Pass	0.000s
spStopAnim() must be defined — cleans up animation interval.	`'spStopAnim' in sim_js`	Pass	0.000s
Business days is three months	`60 <= NUM_BUSINESS_DAYS <= 66`	Pass	0.000s
Daily vol is small	`0.0001 <= DAILY_HAZARD_VOL <= 0.001`	Pass	0.000s
Random walk floors at 0.0001 — vol must not exceed that floor.	`DAILY_HAZARD_VOL > 0; MAX_TOTAL_MOVE > 0`	Pass	0.000s
Cumulative cap should allow meaningful drift.	`MAX_TOTAL_MOVE >= 5 * DAILY_HAZARD_VOL`	Pass	0.000s
Max move bounds vol	`MAX_TOTAL_MOVE > DAILY_HAZARD_VOL; MAX_TOTAL_MOVE <= 0.01`	Pass	0.000s
Test complete gauge analysis workflow.	`results['success'] is True; results['gauge']['gauge_id'] == "THAMES_TEDDINGTON"` (+2 more)	Pass	0.003s
Handle non-existent gauge gracefully.	`results['success'] is False; len(results['errors']) > 0`	Pass	0.001s
Analysis should include flood status.	`'status' in results['flood_analysis']; results['flood_analysis']['status'] in ["NORMAL", "ALERT"...`	Pass	0.002s
Test loading gauge metadata from JSON.	`gauges is not None; len(gauges) == 4` (+2 more)	Pass	0.002s
Test loading gauge time series.	`readings is not None; "THAMES_TEDDINGTON" in readings` (+1 more)	Pass	0.002s
All mortgages fail returns empty list	`result["processing_stats"]["failed_mortgages"] == 2; result["data"]["mortgages"] == []`	Pass	0.004s

Test Description	Acceptance Criteria	Result	Time
When _generate_single_mortgage raises, the generator continues and	`result["processing_stats"]["failed_mortgages"] == 1`; `result["processing_stats"]["successful_mortgages"] == 2` (+1 more)	Pass	0.003s
Find nearest gauge	`gauge_id == "THAMES_TEDDINGTON"`; `distance < 1.0`	Pass	0.000s
Full analysis high risk property	`results["success"] is True`; `results["property"]["property_id"] == "THAMES_TEST_002"` (+1 more)	Pass	0.003s
Full analysis low risk property	`results["success"] is True`; `results["property"]["property_id"] == "THAMES_TEST_003"` (+2 more)	Pass	0.002s
Load gauge data	`gauge_meta is not None`; `len(gauge_meta) == 3` (+2 more)	Pass	0.002s
Load property data	`properties is not None`; `len(properties) == 3` (+3 more)	Pass	0.004s
Clay medium high	`"Medium" in r or "high" in r.lower()`	Pass	0.000s
Peat high risk	`"High" in r`	Pass	0.000s
Rock keyword	`"Low" in r`	Pass	0.000s
Rock low risk	`"Low" in r`	Pass	0.000s
Sand medium risk	`"Medium" in r`	Pass	0.000s
Unknown soil	`"Assessment" in r or "chalk" in r`	Pass	0.000s
STORM-cccccccc (alert-only, no warning/severe) must be last.	`sorted_storms[-1]["storm_id"] == "STORM-cccccccc"`	Pass	0.000s
STORM-bbbbbbbb (2 severe) must rank before all others.	`sorted_storms[0]["storm_id"] == "STORM-bbbbbbbb"`	Pass	0.000s
gauges_severe must be non-increasing.	`severe_counts == sorted(severe_counts, reverse=True)`	Pass	0.000s
Defaults	`p.base_range_km == 40.0`; `p.nugget == 0.05` (+2 more)	Pass	0.000s
From dict	`p.base_range_km == 50.0`; `p.nugget == 0.10` (+1 more)	Pass	0.000s
Load from file	`p.base_range_km == 40.0`; `p.nugget == 0.05`	Pass	0.002s
Detailed report uses more pages than basic, so should produce larger file.	`detailed.stat().st_size >= basic.stat().st_size`	Pass	0.030s
Generate analysis report custom filename	`path.name == "analysis.pdf"`	Pass	0.003s
Generate analysis report custom filename	`path.name == "analysis.pdf"`	Pass	0.010s

Test Description	Acceptance Criteria	Result	Time
Generate analysis report exists	`gen.generate_analysis_report(gauge_data).exists()`	Pass	0.003s
Generate analysis report exists	`gen.generate_analysis_report(minimal_flood_data).exists()`	Pass	0.010s
Generate analysis report with timeseries	`path.exists()`	Pass	0.003s
Generate basic report custom filename	`path.name == "basic.pdf"`	Pass	0.022s
Generate basic report exists	`gen.generate_basic_report(gauge_data).exists()`	Pass	0.023s
Generate basic report exists	`gen.generate_basic_report(minimal_flood_data).exists()`	Pass	0.038s
Generate basic report non empty	`path.stat().st_size > 0`	Pass	0.023s
Generate basic report non empty	`path.stat().st_size > 0`	Pass	0.022s
Generate basic with timeseries	`path.exists()`	Pass	0.023s
Generate detailed report exists	`gen.generate_detailed_report(minimal_flood_data).exists()`	Pass	0.015s
Detailed report includes mortgage_analysis — verify call reaches that page.	—	Pass	0.019s
Generate monitoring report custom filename	`path.name == "mon.pdf"`	Pass	0.003s
Generate monitoring report exists	`gen.generate_monitoring_report(gauge_data, ts_data).exists()`	Pass	0.003s
Generate monitoring report no timeseries	`path.exists()`	Pass	0.003s
Generate mortgage focused report custom filename	`path.name == "mort_focused.pdf"`	Pass	0.016s
Generate mortgage focused report exists	`gen.generate_mortgage_focused_report(prop_data, mort_data...`	Pass	0.016s
Generate property only report custom filename	`path.name == "prop_only.pdf"`	Pass	0.015s
Generate property only report exists	`gen.generate_property_only_report(prop_data).exists()`	Pass	0.015s
Generate risk focused report custom filename	`path.name == "risk_only.pdf"`	Pass	0.009s
Generate risk focused report with mortgage	`path.exists()`	Pass	0.011s
Generate risk focused report without mortgage	`path.exists()`	Pass	0.010s
Generate summary report custom filename	`path.name == "summary.pdf"`	Pass	0.008s
Generate summary report exists	`gen.generate_summary_report(minimal_flood_data).exists()`	Pass	0.008s
Mortgage focused includes mortgage context pages	—	Pass	0.016s
Risk focused with mortgage includes current status	—	Pass	0.011s

Test Description	Acceptance Criteria	Result	Time
Risk focused without mortgage excludes current status	—	Pass	0.010s
Distance first path	`"distance_effect_bps" in df`; `"elevation_effect_bps" in df`	Pass	0.004s
Distance first sums to basis	`abs(total - prop) < 0.5`	Pass	0.004s
Elevation first path	`"elevation_effect_bps" in ef`; `"distance_effect_bps" in ef`	Pass	0.004s
Elevation first sums to basis	`abs(total - prop) < 0.5`	Pass	0.004s
Gauge spread present	`"gauge_spread_bps" in sd`; `isinstance(sd["gauge_spread_bps"], (int, float))`	Pass	0.004s
property_spread_bps should equal the 5yr severe spread from term_structure.	`abs(sd["property_spread_bps"] - ts_spreads[4]) < 0.1`	Pass	0.004s
Property spread present	`"property_spread_bps" in sd`	Pass	0.004s
propertyshd.json and propertyshe.json must exist for decomposition.	`shd.exists()`; `she.exists()`	Pass	0.000s
Shd spread present	`"shd_spread_bps" in sd`	Pass	0.004s
She spread present	`"she_spread_bps" in sd`	Pass	0.004s
Only gauge_GAUGE-*_hd.json files are removed, not other files.	`other.exists()`	Pass	0.186s
Multiple stale files are all removed.	`not s.exists()`	Pass	0.189s
Lines 37-39: old gauge_GAUGE-*_hd.json files are removed.	`stale.exists()`; `not stale.exists()`	Pass	0.188s
After clear-all, every gauge should show null metrics.	`resp.get_json()["status"] == "success"`; `g["has_model"] is False` (+2 more)	Pass	0.022s
Gauge with both .joblib AND summary entry should show metrics.	`gauge["has_model"] is True`; `gauge["status"] == "trained"` (+2 more)	Pass	0.015s
Gauge with summary entry but NO .joblib should show null metrics.	`gauge["has_model"] is False`; `gauge["status"] == "not_trained"` (+4 more)	Pass	0.011s
All fresh full health	`data["health_pct"] == 100.0`	Pass	0.007s
All missing zero health	`r.status_code == 200`; `data["status"] == "success"` (+2 more)	Pass	0.001s
Has as of	`"as_of" in data`	Pass	0.001s
Partial health	`data["health_pct"] == pytest.approx(7.1, abs=0.1)`	Pass	0.001s
Stale does not count as fresh	`data["health_pct"] == 0.0`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Every dataset key must appear in _startupDetail – no silent blanks.	`f"key === '{k}'" in src`	Pass	0.000s
Blotter stat uses trades length	`"key === '_tdPreBlotter'" in src; "data.trades.length + ' trades'" in src`	Pass	0.000s
Eod history stat uses history length	`"key === '_tdPreEodHistory'" in src; "data.history.length + ' snapshots'" in src`	Pass	0.000s
Gauge hist stat uses count	`"key === '_preGaugeHist'" in src; 'data.count' in src`	Pass	0.000s
Gauges stat uses gauges length	`"key === '_tdPreGauges'" in src; "data.gauges.length + ' gauges'" in src`	Pass	0.000s
Gov audit stat uses events length	`"key === '_preGovAudit'" in src; "data.total_entries + ' events'" in src`	Pass	0.000s
Gov bib stat uses references length	`"key === '_preGovBib'" in src; "data.references.length + ' refs'" in src`	Pass	0.000s
Gov docs stat uses documents length	`"key === '_tdPreGovDocs'" in src; "data.documents.length + ' docs'" in src`	Pass	0.000s
Market stat uses object keys gauges	`"key === '_tdPreMarket'" in src; "Object.keys(data.gauges).length" in src` (+1 more)	Pass	0.000s
Mortgages stat uses mortgages length	`"key === '_preMortgages'" in src; "data.mortgages" in src`	Pass	0.000s
Port storms stat uses storms length	`"key === '_tdPrePortStorms'" in src; "data.storms.length + ' storms'" in src`	Pass	0.000s
Pre storms stat uses storms length	`"key === '_preStorms'" in src; "data.storms.length + ' storms'" in src`	Pass	0.000s
Property ts stat uses total properties	`"key === '_prePropertyTS'" in src; 'data.data.summary.properties_with_floods' in src` (+1 more)	Pass	0.000s
Stress storms stat uses storms length	`"key === '_tdPreStressStorms'" in src; "data.storms.length + ' scenarios'" in src`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Yield curve stat uses object keys	"key === '_tdPreYieldCurve'" in src; "Object.keys(data.yield_curve).length" in src (+1 more)	Pass	0.000s
Add to map calls add child	—	Pass	0.001s
Add to map output contains script	'<script>' in rendered; 'window._tdPreloadDone' in rendered	Pass	0.001s
Add to map method defined	'def add_to_map' in src	Pass	0.000s
All fourteen endpoints present	ep in js	Pass	0.000s
Must NOT defer to a button click – runs on DOMContentLoaded.	'DOMContentLoaded' in js	Pass	0.000s
'Market state' was renamed to 'Hazard curves'.	'Hazard curves' in js; "'Market state'" not in js and '"Market state"' not in js	Pass	0.000s
Sets window._tdPreloadDone so any IIFE can read it via global scope.	'window._tdPreloadDone = true' in _src('src/visual/intera...	Pass	0.000s
Shows progress popup	'startup-preloader-popup' in js; 'startup-pre-bar' in js	Pass	0.000s
Startup preloader class defined	'class StartupPreloader' in src	Pass	0.000s
Declares pre gauge hist	'window._preGaugeHist' in _src('src/visual/interactivity/...	Pass	0.000s
Declares pre gov audit	'window._preGovAudit' in _src('src/visual/interactivity/s...	Pass	0.000s
Declares pre gov bib	'window._preGovBib' in _src('src/visual/interactivity/sta...	Pass	0.000s
Declares pre property ts	'window._prePropertyTS' in _src('src/visual/interactivity...	Pass	0.000s
Declares pre storms	'window._preStorms' in _src('src/visual/interactivity/sta...	Pass	0.000s
Declares td pre blotter	'window._tdPreBlotter' in _src('src/visual/interactivity/...	Pass	0.000s
Declares td pre eod history	'window._tdPreEodHistory' in _src('src/visual/interactivi...	Pass	0.000s
Declares td pre gauges	'window._tdPreGauges' in _src('src/visual/interactivity/s...	Pass	0.000s
Declares td pre gov docs	'window._tdPreGovDocs' in _src('src/visual/interactivity/...	Pass	0.000s
Declares td pre market	'window._tdPreMarket' in _src('src/visual/interactivity/s...	Pass	0.000s
Declares td pre port storms	'window._tdPrePortStorms' in _src('src/visual/interactivi...	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Declares td pre stress storms	'window._tdPreStressStorms' in _src('src/visual/interacti...	Pass	0.000s
Declares td pre yield curve	'window._tdPreYieldCurve' in _src('src/visual/interactivi...	Pass	0.000s
Line 46: GET /apple-touch-icon.png returns the PNG.	r.status_code == 200; "image/png" in r.content_type	Pass	0.002s
Line 51: GET /apple-touch-icon-precomposed.png returns the PNG.	r.status_code == 200; "image/png" in r.content_type	Pass	0.001s
Line 41: GET /favicon.ico returns the icon file.	r.status_code == 200; r.content_type in ("image/x-icon", "image/vnd.microsoft.i...	Pass	0.001s
Stats bar spread should use adjustedPropSpread.	idx_spread > idx_bar	Pass	0.000s
Zone tag only shown when selected zone differs from actual.	'selectedZone !== actualZone' in js	Pass	0.000s
Zone tag present	'zoneTag' in js	Pass	0.000s
Zone tag should show 'vs' the actual zone.	'(vs ' in js	Pass	0.000s
avg_auc/avg_accuracy are None when results have no metrics.	data["avg_auc"] is None; data["avg_accuracy"] is None	Pass	0.001s
Status computes avg auc and accuracy	data["avg_auc"] == 0.85; data["avg_accuracy"] == 0.9	Pass	0.001s
Click handler defined	'_navigateToStress' in hist_js	Pass	0.000s
Click handler iife safe	'addEventListener' in hist_js; 'hist-storm-row' in hist_js	Pass	0.000s
Navigates to tab 5	'switchTab(5)' in hist_js; 'switchTab(4)' not in hist_js.replace('switc...	Pass	0.000s
Sets stress storm hint	'window._stressStormHint' in hist_js	Pass	0.000s
Storm hint set before switch tab	idx_hint > 0; idx_switch > 0 (+1 more)	Pass	0.000s
Storm id passed to navigate	'data-storm-id' in hist_js; 'getAttribute' in hist_js	Pass	0.000s
Every gauge in gaugehc.json should have a trained classifier.	not missing	Skip	0.000s
Every gauge referenced in stress_storms should have a trained classifier.	not missing	Skip	0.000s
Every gauge in stress_storms should have a hazard curve in gaugehc.json.	not missing	Skip	0.001s

Test Description	Acceptance Criteria	Result	Time
Storm IDs in stress_storms must overlap with storm IDs in propertyts/.	`len(overlap) > 0`	Skip	0.000s
Every storm_id in stress_storms.json must exist in gaugets/GAUGE-*.json.	`not missing`	Skip	0.000s
Storm ids match flood events	`actual_ids == expected_ids`	Pass	0.088s
Each storm has damage ratio	`"damage_ratio" in sd; 0 <= sd["damage_ratio"] <= 1`	Pass	0.000s
Each storm has exceeded severe	`"exceeded_severe" in sd; isinstance(sd["exceeded_severe"], bool)`	Pass	0.000s
Each storm has flood depth m	`"flood_depth_m" in sd; isinstance(sd["flood_depth_m"], (int, float))`	Pass	0.000s
Each storm has flooded flag	`"flooded" in sd; isinstance(sd["flooded"], bool)`	Pass	0.000s
Each storm has gauge peak m	`"gauge_peak_m" in sd; isinstance(sd["gauge_peak_m"], (int, float))`	Pass	0.000s
Each storm has retention factor	`"retention_factor" in sd; isinstance(sd["retention_factor"], (int, float))`	Pass	0.000s
Each storm has storm id	`"storm_id" in sd; isinstance(sd["storm_id"], str)`	Pass	0.000s
Flooded storms have positive depth	`sd["flood_depth_m"] > 0`	Pass	0.000s
Non flooded storms have zero depth	`len(non_flooded) > 0; sd["flood_depth_m"] == 0`	Pass	0.000s
Non flooded storm in details	`len(non_flooded) == 1; non_flooded[0]["storm_id"] == "S2"` (+1 more)	Pass	0.002s
Storm details count	`len(storms) == 3`	Pass	0.002s
Storm details fields	`"storm_id" in storm; "gauge_peak_m" in storm` (+5 more)	Pass	0.003s
Storm details present	`r.status_code == 200; "storm_details" in data`	Pass	0.002s
Storm details values	`storm["storm_id"] == "S1"; storm["gauge_peak_m"] == 5.5` (+1 more)	Pass	0.002s
Storm details empty when no floods	`result["storm_details"] == []`	Pass	0.088s

Test Description	Acceptance Criteria	Result	Time
Non-flooded events appear in storm_details (unlike flood_count).	`len(result["storm_details"]) == 4`; `result["flood_count"] == 3`	Pass	0.088s
Storm details key exists	`"storm_details" in result`	Pass	0.091s
Storm details length matches flood events	`len(result["storm_details"]) == 5`	Pass	0.087s
Every storm dropdown must show the storm_id in the option text.	`not violations`	Pass	0.001s
All dropdown option builders must use sentinel or pipe separators.	`not violations`	Pass	0.001s
The storm display format is centralised in config/format.py.	`' | ' in js`; `'storm_id' in js` (+2 more)	Pass	0.000s
'precipitation' as a label suffix must not appear.	`not violations`	Pass	0.001s
'severe levels' (old space-separated format) must not appear.	`not violations`	Pass	0.003s
port_stress/setup.py uses sentinel with show_warning=True.	`'_storm_opt' in src`; `'show_warning=True' in src`	Pass	0.000s
sp_table.py delegates storm label to config.format via sentinel.	`'_storm_opt' in src`; `'__STORM_OPT__' in src` (+1 more)	Pass	0.000s
Custom variable name is used throughout — gsa_timeline uses 'r'.	`js.startswith('(r.name')`; `'s.storm_id' not in js`	Pass	0.000s
show_peak=True adds peak water level after mm.	`'peak' in js_peak`	Pass	0.000s
show_warning=True adds gauges_warning count between severe and mm.	`'warn' in js_warn`; `'warn' not in js_base`	Pass	0.000s
trading/stress/setup.py uses sentinel with show_peak=True.	`'_storm_opt' in src`; `'show_peak=True' in src`	Pass	0.001s
Visual files must not contain hard-coded storm display format strings.	`pat not in src`	Pass	0.001s
Lines 255-256: exception in create_mapping -¿ raises ValueError.	—	Pass	0.000s
Lines 213-256: create_mapping with all sections populated.	`mapping["storm_event_id"] == "STORM-001"`; `mapping["storm_name"] == "Eunice"` (+1 more)	Pass	0.000s
Create mapping returns flat dict	`isinstance(mapping, dict)`; `mapping.get("gauge_id") == "GAUGE-test1234"` (+7 more)	Pass	0.000s
Line 258-260: get_required_fields() returns list with StormEventID.	`isinstance(fields, list)`; `any("StormEventID" in f for f in fields)`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Required fields include storm id	`any("StormEventID" in f for f in storm_cdm.get_required_f...`	Pass	0.000s
Schema has storm root	`len(list(storm_cdm.schema.keys())) > 0`	Pass	0.000s
validate({}) returns errors for missing fields.	`isinstance(errors, dict)`	Pass	0.000s
Lines 192-193: exception in validate -¿ {'validation_error': [...]}.	`isinstance(result, dict); "validation_error" in result`	Pass	0.000s
validate() with missing CatchmentID returns Header errors.	`"Header" in errors; any("CatchmentID" in e for e in errors["Header"])`	Pass	0.000s
Lines 175-193: validate() with missing StormEventID returns Header errors.	`"Header" in errors; any("StormEventID" in e for e in errors["Header"])`	Pass	0.000s
validate() with valid data returns empty dict.	`storm_cdm.validate(data) == {}`	Pass	0.000s
Unknown intensity profile falls to else: intensity = peak * t_frac.	`storm.storm_id.startswith("STORM-"); len(storm.track) == 5` (+1 more)	Pass	0.000s
Effective precipitation mm consistent	`'effective_precipitation_mm' in hist_js`	Pass	0.000s
Gauges severe consistent	`'gauges_severe' in hist_js`	Pass	0.000s
Intensity category consistent	`'intensity_category' in hist_js`	Pass	0.000s
Peak level m consistent	`'peak_level_m' in hist_js`	Pass	0.000s
Storm id consistent	`'storm_id' in hist_js`	Pass	0.000s
Storm IDs in gaugets/ must come from storm_sequences.json.	`len(overlap) > 0`	Pass	0.415s
Storm IDs in propertyts must come from storm_sequences.json.	`len(overlap) > 0`	Pass	1.750s
storm_multi.make_storm_id() produces STORM-{8hex} IDs.	`STORM_ID_RE.match(sid)`	Pass	0.000s
storm generator must NOT use STORM-{i:03d} or STORM-{i:04d}.	`'STORM-{i:' not in src; 'STORM-{i+' not in src`	Pass	0.000s
Single storm generator uses storm prefix	`'STORM-' in src or 'STORM_ID_PREFIX' in src`	Pass	0.001s
Single storm generator uses uuid	`'uuid' in src`	Pass	0.000s
Uuid4 produces protocol matching ids	`GAUGE_ID_RE.match(gauge_id); PROP_ID_RE.match(prop_id)` (+2 more)	Pass	0.000s
Accepts valid	`PROP_ID_RE.match('PROP-a3f7bc12'); PROP_ID_RE.match('PROP-00000000')` (+6 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Human-readable names like STORM-FLOOD are not protocol-compliant.	not STORM_ID_RE.match('STORM-FLOOD'); not STORM_ID_RE.match('STORM-SAFE') (+1 more)	Pass	0.000s
STORM-001, STORM-0001 violate the agreed protocol.	not GAUGE_ID_RE.match('GAUGE-001'); not GAUGE_ID_RE.match('GAUGE-002') (+5 more)	Pass	0.000s
Rejects uppercase	not PROP_ID_RE.match('PROP-A3F7BC12'); not STORM_ID_RE.match('STORM-C7F4B8E2')	Pass	0.000s
STORM-PORT-001 uses a compound prefix — not protocol-compliant.	not STORM_ID_RE.match('STORM-PORT-001'); not STORM_ID_RE.match('STORM-PORT-002')	Pass	0.000s
Storm from list is runnable	resp.status_code == 200; len(data['storms']) > 0 (+2 more)	Pass	0.013s
Worst storm auto selection is runnable	run_resp.status_code == 200	Pass	0.006s
Storm list endpoint must be /api/v1/propertyts/storms.	'/api/v1/propertyts/storms' in table_js	Pass	0.000s
Empty storm list must display a user-readable message.	'No flooding storms found' in table_js	Pass	0.000s
JS must check window._preStorms cache before fetching.	'_preStorms' in table_js	Pass	0.000s
Shows overflow count	'more storms' in hist_js or '+ ' in hist_js	Pass	0.000s
Shows top 30 storms	'30' in hist_js	Pass	0.000s
Sorted by peak level	'sorted by peak level' in hist_js or \{} 'pe...	Pass	0.000s
Storm with empty name falls back to storm_id.	resp.status_code == 200; s['name'] == s['storm_id']	Pass	0.042s
Apply storm list function exists	'_applyStormList' in src	Pass	0.000s
Checks pre storms cache	'window._preStorms' in src	Pass	0.000s
Cache var should be nulled after consuming to allow refresh.	'window._preStorms = null' in src; 'window._tdPreGovDocs = null' in src	Pass	0.000s
Uses effective precipitation	'effective_precipitation_mm' in hist_js	Pass	0.000s
Uses gauges severe	'gauges_severe' in hist_js	Pass	0.000s
Uses intensity category	'intensity_category' in hist_js	Pass	0.000s
Uses max trigger for colour	'max_trigger' in hist_js	Pass	0.000s
Uses peak level m	'peak_level_m' in hist_js	Pass	0.000s
Uses storm id	'storm_id' in hist_js	Pass	0.000s
Uses storm name field	's.name' in hist_js or 'storm.name' in hist_js	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Historical tab must NOT use /gauges/¡id¿/storms for the storm list.	'_loadStormScenarios' in hist_js; fn_idx >= 0 (+1 more)	Pass	0.000s
_loadStormScenarios function must be defined.	'_loadStormScenarios' in hist_js	Pass	0.000s
Storm list renders in hist-storms-list element.	'hist-storms-list' in hist_js; 'hist-storms-panel' in hist_js	Pass	0.000s
Storm list must be loaded from the same endpoint as Stress tab.	'/api/v1/trading/stress/storms' in hist_js	Pass	0.000s
Sorts by peak level for auto-selection.	'peak_level_m' in stress_js	Pass	0.000s
Storm select element present.	'stress-storm-select' in stress_js	Pass	0.000s
Lines 172-173: exception in create_mapping raises ValueError.	—	Pass	0.000s
Lines 150-170: create_mapping with full Parameters section.	mapping["time_series_id"] == "TS-001"; mapping["time_step_minutes"] == 60	Pass	0.000s
Create mapping returns flat dict	isinstance(stormts_cdm.create_mapping(data), dict); isinstance(prs_cdm.create_mapping(data), dict)	Pass	0.000s
Line 185: create_reading_mapping returns flat dict.	mapping["timestamp"] == "2022-02-18T06:00"; mapping["rainfall_mm"] == 12.5 (+1 more)	Pass	0.000s
Lines 194-199: get_required_fields() returns list.	isinstance(fields, list); any("TimeSeriesID" in f for f in fields)	Pass	0.000s
Required fields include ts id	any("TimeSeriesID" in f for f in stormts_cdm.get_required...	Pass	0.000s
Schema has root	isinstance(stormts_cdm.schema, dict); len(stormts_cdm.schema) > 0 (+2 more)	Pass	0.000s
validate({}) returns errors for both required fields.	isinstance(errors, dict); "Header" in errors	Pass	0.000s
Lines 137-138: exception in validate -¿ {'validation_error': [...]}.	isinstance(result, dict); "validation_error" in result	Pass	0.000s
validate() with missing CatchmentID returns Header errors.	"Header" in errors	Pass	0.000s
Lines 120-138: validate() with missing TimeSeriesID returns Header errors.	"Header" in errors; any("TimeSeriesID" in e for e in errors["Header"])	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
validate() with valid data returns empty dict.	`stormts_cdm.validate(data) == {}`	Pass	0.000s
Custom kernel	`d["decay_kernel"] == "exponential"`	Pass	0.000s
Decay kernel serialised as string	`isinstance(d["decay_kernel"], str); d["decay_kernel"] == "gaussian"`	Pass	0.000s
Has storm id	`storm.to_dict()["storm_id"] == "STORM-0001"`	Pass	0.000s
Has track	`"track" in d; len(d["track"]) == 2`	Pass	0.000s
Intensity profile serialised as string	`isinstance(d["intensity_profile"], str); d["intensity_profile"] == "triangular"`	Pass	0.000s
Returns dict	`isinstance(d, dict); isinstance(d, dict)`	Pass	0.000s
Track has coords	`"longitude" in point; "latitude" in point` (+2 more)	Pass	0.000s
storms.py lines 91-92: corrupted gaugets file -¿ skipped in severe count loop.	`r.status_code == 200`	Pass	0.031s
storms.py line 80: gaugets dir doesn't exist -¿ skip severe counting.	`r.status_code == 200`	Pass	0.032s
storms.py line 61: storm_sequences.json doesn't exist -¿ skip enrichment.	`r.status_code == 200; data["storm_responses"]["num_storms"] == 1`	Pass	0.030s
storms.py lines 75-76: corrupted storm_sequences.json -¿ silently skipped.	`r.status_code == 200`	Pass	0.031s
storms.py lines 113-115: exception in storm loading -¿ 500.	`r.status_code == 500; r.get_json()["status"] == "error"`	Pass	0.028s
Calls proc wait	—	Pass	0.000s
_stream_proc should push each line from proc.stdout into _output_lines.	`"line one" in lines; "line two" in lines` (+1 more)	Pass	0.000s
Strips newlines	`"stripped" in lines; "stripped\{}n" not in lines`	Pass	0.000s
Hits /trading/stress/run.	`'/trading/stress/run' in stress_js`	Pass	0.000s
Hits /trading/stress/storms.	`'/trading/stress/storms' in stress_js`	Pass	0.000s
stress/run returns 500 on engine error.	`resp.status_code == 500`	Pass	0.004s
stress/gauges returns 500 when gauge loader raises.	`resp.status_code == 500`	Pass	0.001s
stress/storms returns 500 when storm loader raises.	`resp.status_code == 500`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
_tdPopulateGaugeDropdown must survive f-string rendering.	'_tdPopulateGaugeDropdown' in full_js	Pass	0.000s
tdStressGaugeChanged must survive f-string rendering in tradingdesk.py.	'tdStressGaugeChanged' in full_js	Pass	0.000s
Rendered JS must not have unescaped f-string brace artifacts.	len(full_js) > 10000	Pass	0.000s
td-stress-gauge element must be present for JS to find it.	"getElementById('td-stress-gauge')" in stress_js	Pass	0.000s
_tdPopulateGaugeDropdown must exist – builds gauge selector.	'_tdPopulateGaugeDropdown' in stress_js	Pass	0.000s
_tdPopulateStormDropdown must exist – builds storm list for selected gauge.	'_tdPopulateStormDropdown' in stress_js	Pass	0.000s
td-stress-storm element must be present for JS to find it.	"getElementById('td-stress-storm')" in stress_js	Pass	0.000s
Stress functions must be INSIDE the IIFE, not after it.	idx_gauge_changed > 0; idx_gauge_changed < iife_close	Pass	0.000s
gauge onchange must call tdStressGaugeChanged.	'tdStressGaugeChanged()' in stress_js	Pass	0.000s
tdStressGaugeChanged must exist – called when gauge is selected.	'function tdStressGaugeChanged' in stress_js	Pass	0.000s
storm onchange must call tdStressStormChanged.	'tdStressStormChanged()' in stress_js	Pass	0.000s
tdStressStormChanged must exist – called when storm is selected.	'function tdStressStormChanged' in stress_js	Pass	0.000s
'First with trades' fallback must only run when sel.value is still empty.	"if (!sel.value && tdStressGauges.length > 0)" in js	Pass	0.000s
Hint selection must happen before the 'first with trades' fallback.	idx_hint > 0; idx_fallback > idx_hint	Pass	0.000s
tdStressGaugeHint must be set to null after use to prevent stale hint.	'tdStressGaugeHint = null' in js	Pass	0.000s
When no market hint, tdBlotterFilters.gauge_id must be the fallback.	'tdBlotterFilters' in js; 'tdBlotterFilters.gauge_id' in js	Pass	0.000s
loadStressData must accept a gaugeHint parameter.	'function loadStressData(gaugeHint)' in js	Pass	0.000s
loadStressData must store gaugeHint in tdStressGaugeHint.	'if (gaugeHint) tdStressGaugeHint = gaugeHint' in js	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
_tdPopulateGaugeDropdown must use tdStressGaugeHint for pre-selection.	'tdStressGaugeHint' in js; idx_hint_use > idx_populate	Pass	0.000s
tdStressGaugeHint state variable must exist in stress JS.	'tdStressGaugeHint' in js	Pass	0.000s
switchTab('stress') must use blotter filter as primary hint.	idx_blotter > 0; idx_market > 0 (+4 more)	Pass	0.000s
switchTab('stress') in tradingdesk must pass gauge hint to loadStressData.	"loadStressData(stressHint)" in js; "tdSelectedGauge" in js (+1 more)	Pass	0.000s
switchTab('stress') must fall back to first blotter trade gauge when no filter set.	"tdBlotterData" in js; "tdBlotterData[0].gauge_id" in js	Pass	0.000s
peak_position=0 uses decay-only path.	len(levels) == 10; all(isinstance(v, float) for v in levels)	Pass	0.000s
remaining=0 uses frac=1.0 path.	len(levels) == 5	Pass	0.000s
Both stress charts must show KO vertical annotation line.	'koLine' in js; 'first_trigger_hour' in js	Pass	0.000s
Stress table must show KO status per trade.	'triggered_hour' in js; 'KO H' in js	Pass	0.000s
Stats bar must show knock-out count.	'num_triggered' in js; 'Knocked Out' in js	Pass	0.000s
Storm id exact match required	resp.status_code == 200; resp.status_code == 404	Pass	0.008s
Storm list has all run inputs	data['gauge_id'] == 'GAUGE-001'; 'storm_id' in s	Pass	0.003s
Response includes trigger levels for threshold annotation on charts.	'alert_level' in data; 'warning_level' in data (+1 more)	Pass	0.008s
Response includes hydrograph_source string describing the data origin.	'hydrograph_source' in data	Pass	0.006s
Response includes storm_name and intensity_category for header display.	data['storm_name'] == 'Severe Storm Alpha'; data['intensity_category'] == 'severe'	Pass	0.007s
Hourly array must be 168 elements (7-day storm window).	len(data['hourly']) == 168	Pass	0.007s
Surface table must cover the full 168h window regardless of knock-out.	max(surface['hours']) >= 164	Pass	0.007s
If the index has not changed, the cached object is returned.	id(data1) == id(data2)	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Replacing stress_storms/_index.json on disk is picked up on next	`len(data1['storms']) == 1; len(data2['storms']) == 2`	Pass	0.019s
Thames catchment has 52 gauges; all must be represented.	`len(gauge_ids) == 52`	Pass	0.025s
All gauge_id values in gauge_responses must match GAUGE-/SYNTH-xxxx format.	`not bad`	Pass	0.028s
All storm_id values must use STORM- prefix.	`not bad`	Pass	0.001s
At least one storm exceeds alert	`any_alert`	Pass	0.000s
Each storm has gauge responses	`not empty`	Pass	0.000s
Each storm has required fields	`not missing`	Pass	0.003s
File exists	—	Pass	0.000s
Gauge ids well formed	`not bad`	Pass	0.028s
Gauge responses have required fields	`not missing`	Pass	0.096s
Has storms	`len(storms) > 0`	Pass	8.940s
Intensity categories present	`cats`	Pass	0.001s
Minimum storm count	`len(storms) >= MIN_STORM_COUNT`	Pass	0.000s
Peak level exceeds base	`not violations`	Pass	0.034s
Peak levels positive	`not violations`	Pass	0.036s
Storm ids unique	`not dupes`	Pass	0.467s
Storm ids well formed	`not bad`	Pass	0.001s
Storms cover multiple gauges	`len(gauges) >= 10`	Pass	0.030s
Storms must be sorted: gauges_severe DESC, gauges_warning DESC.	`severe_counts == sorted(severe_counts, reverse=True)`	Pass	0.001s
Total storms key matches list length	`data["total_storms"] == len(data["storms"])`	Pass	0.034s
Trigger summary max trigger valid	`not bad`	Pass	0.001s
Every storm must have a fully-populated trigger_summary dict.	`not missing`	Pass	0.007s
No gauge id response has count	`isinstance(data['count'], int); data['count'] > 0`	Pass	0.002s
Startup preloader fetches without gauge_id – must not return 400.	`r.status_code == 200`	Pass	0.002s
All storms in file returned – not filtered by gauge.	`'storms' in data; data['count'] == len(data['storms'])` (+1 more)	Pass	0.002s
No gauge id returns success	`data['status'] == 'success'`	Pass	0.002s
gauge_id param still filters to matching storms.	`data['status'] == 'success'; all(True for _ in data['storms'])`	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
When stress_storms.json is absent, returns 404.	`resp.status_code == 404`; `'not found' in data.get('message', '').lower()`	Pass	0.000s
ghc_stress.get_js() returns valid JS.	`len(stress_js) > 1000`	Pass	0.000s
Chart canvas element must use td-prefixed ID.	`'td-stress-chart-canvas' in js`	Pass	0.000s
Chart sub-tabs must exist for Flood Probability, Stress P&L, and Surface.	`'td-stress-ctab-0' in js`; `'td-stress-ctab-1' in js` (+3 more)	Pass	0.000s
Stress tab must have both chart renderers.	`'_tdRenderProbabilityChart' in js`; `'_tdRenderStressPnlChart' in js`	Pass	0.000s
Stress tab must have cleanup function.	`'tdCleanupStressCharts' in js`	Pass	0.000s
Stress tab must have gauge selector dropdown.	`'td-stress-gauge' in js`; `'tdStressGaugeChanged' in js`	Pass	0.000s
Stress tab must fetch from stress/-gauges endpoint.	`'/api/v1/trading/stress/gauges' in js`	Pass	0.000s
Stress tab must have createStressView and loadStressData.	`'createStressView' in js`; `'loadStressData' in js`	Pass	0.000s
Charts must have KO annotation line.	`'koLine' in js`; `'first_trigger_hour' in js`	Pass	0.000s
Stress tab must have Property Damage link.	`'tdStressOpenDamage' in js`; `'Property Damage' in js`	Pass	0.000s
Stress tab must have Storm Simulation link.	`'tdStressOpenGauge' in js`; `'Storm Simulation' in js`	Pass	0.000s
Stress sub-module should render without errors.	`len(js) > 0`	Pass	0.000s
Stress tab must have stats bar with KO info.	`'_tdRenderStressStats' in js`; `'td-stress-stats-bar' in js` (+1 more)	Pass	0.000s
Stress tab must have storm selector dropdown.	`'td-stress-storm' in js`; `'tdStressStormChanged' in js`	Pass	0.000s
Stress tab must render probability surface table.	`'probability_surface' in js`; `'td-stress-surface-wrap' in js` (+2 more)	Pass	0.000s
Stress tab must have trade table renderer.	`'_tdRenderStressTable' in js`; `'td-stress-table-wrap' in js`	Pass	0.000s
Surface table must be horizontally scrollable for 168h of columns.	`'overflow:auto' in js`; `'width:100%' not in snippet`	Pass	0.000s
Should fetch from /api/v1/trading/stress/storms.	`'/api/v1/trading/stress/storms' in js`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Should use cash pricing formula.	'cash_price' in js or 'cashPrice' in js or 'portfolio_cas...	Pass	0.000s
Should have two chart sub-tabs: Flood Probability and Stress P&L.	'Flood Probability' in js; 'Stress P' in js (+2 more)	Pass	0.000s
Should have storm selector drop-down.	'stress-storm-select' in js; '_populateStormSelect' in js	Pass	0.000s
Stress test JS should render without errors.	len(js) > 0	Pass	0.000s
Should POST to /api/v1/trading/stress/run.	'/api/v1/trading/stress/run' in js; '_runStressScenario' in js	Pass	0.000s
Should show P(flood) in chart.	'p_flood' in js	Pass	0.000s
Should pick up _stressStormHint from Historical tab.	'_stressStormHint' in js; '_stressStormHint = null' in js	Pass	0.000s
Should render trade table at peak hour.	'stress-table-wrap' in js; '_renderStressTable' in js	Pass	0.000s
Chart tab clicks should use addEventListener (IIFE-safe).	'addEventListener' in js	Pass	0.000s
String with options picks from list	result in ["Brick", "Stone", "Timber"]	Pass	0.000s
String without options returns empty	result == ""	Pass	0.000s
Colour constants	hasattr(styles_mod, name)	Pass	0.000s
Get styles keys	expected == set(S.keys())	Pass	0.000s
Rag colour maps	key in styles_mod.RAG_COLOURS; key in styles_mod.RAG_BG	Pass	0.000s
Section rule appends flowables	len(story) == 3	Pass	0.000s
Tbl style returns tablestyle	isinstance(ts, TableStyle)	Pass	0.000s
No data returns 404	r.status_code == 404; r.get_json()["status"] == "error"	Pass	0.000s
Options returns ok	r.status_code == 200; r.status_code == 200 (+1 more)	Pass	0.000s
With data returns success	r.status_code == 200; r.get_json()["status"] == "success"	Pass	0.002s
avg_auc_roc is None when no gauges are trained.	data["avg_auc_roc"] is None	Pass	0.009s
Verify all expected gauge fields are in the response.	field in g	Pass	0.025s
avg_auc_roc averages correctly over multiple trained gauges.	data["num_trained"] == 2; data["avg_auc_roc"] == 0.85	Pass	0.011s
Internal error → 500.	resp.status_code == 500; data["status"] == "error"	Pass	0.008s

Test Description	Acceptance Criteria	Result	Time
Gauge with .joblib but no auc_roc in summary still counted as trained.	`data["num_trained"] == 1`; `data["avg_auc_roc"] == 0.0`	Pass	0.010s
Exceedance probs decreasing	`stats['prob_exceed_60'] >= stats['prob_exceed_70'] >= ...`	Pass	0.000s
Fat tail higher p99 than thin	`fat['p99'] > thin['p99']`	Pass	0.148s
Mean in reasonable range	`30 < stats['mean'] < 70`	Pass	0.000s
Percentiles ordered	`stats['p10'] <= stats['p25'] <= stats['p50'] <= \{} ...`	Pass	0.000s
Required keys present	`k in stats`	Pass	0.000s
Cap	`result[50] == pytest.approx(11.0)`	Pass	0.000s
Empty pulses	`len(result) == 168`	Pass	0.000s
Non overlapping sequential	`result[25] == pytest.approx(7.0)`; `result[55] == pytest.approx(8.0)` (+1 more)	Pass	0.000s
Overlapping exceeds single	`result[55] == pytest.approx(9.0)`; `result[45] == pytest.approx(7.0)`	Pass	0.000s
Single pulse	`result[55] == pytest.approx(8.0)`; `result[0] == pytest.approx(5.0)`	Pass	0.000s
Surface footer references KO trimming.	`'trimmed at KO' in stress_js`	Pass	0.000s
Surface table reads probability_surface from data.	`'probability_surface' in stress_js`	Pass	0.000s
Renders an HTML table with monospace font.	`'border-collapse' in stress_js`; `'font-family:monospace' in stress_js`	Pass	0.000s
Surface table applies trigger-band background colours.	`'#FFEBEE' in stress_js`; `'#FFF3E0' in stress_js` (+1 more)	Pass	0.000s
Surface tab creates a wrapping div.	`'stress-surface-wrap' in stress_js`	Pass	0.000s
No synthetic falls back to distance order	`result[0]['distance_m'] <= result[1]['distance_m'] <= res...`; `not r['gauge_id'].startswith('SYNTH')`	Pass	0.000s
Real gauges sorted by distance after synthetic	`result[0]['gauge_id'].startswith('SYNTH')`; `not result[1]['gauge_id'].startswith('SYNTH')` (+2 more)	Pass	0.000s
Synthetic at position zero	`result[0]['gauge_id'] == 'SYNTH-001'`	Pass	0.000s
Synthetic first even when far	`result[0]['gauge_id'] == 'SYNTH-001'`	Pass	0.000s
Synthetic gauge info preserved	`synth['gauge_info']['elevation'] == 5.0`; `synth['gauge_info']['severe_level'] == 7.0`	Pass	0.000s
Three gauges returned	`len(result) == 3`	Pass	0.000s
No synth returns none	`synth is None`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Synth prefix detection	`synth is not None`; `synth['gauge_id'] == 'SYNTH-123'`	Pass	0.000s
Lines 72-76: water_level ¡= severe_level → no flooding.	—	Pass	0.000s
Line 96: flood_depths[i] = min(depth, 5.0).	`np.all(depths <= 5.0)`	Pass	0.000s
Lines 89-92: distance_factor reduces depth proportionally.	`depths[0] > 0`; `depths[1] > 0`	Pass	0.000s
Lines 72: len(gauge_data) == 0.	—	Pass	0.000s
Lines 93-94: distance ¿= max_distance → depth = 0.0.	`depths[0] == 0.0`	Pass	0.000s
Lines 72-96: water_level ¿ severe_level, close to Thames.	`np.any(depths > 0)`	Pass	0.000s
Lines 72: gauge_data is None.	—	Pass	0.000s
Every property should have a synthetic gauge at position [0].	`first.startswith("SYNTH")`	Pass	0.004s
Zone must match the elevation offset above the synthetic gauge.	`offset >= lo`; `lo <= offset < hi`	Pass	0.004s
Should generate enough synthetics for the portfolio.	`len(synths) >= 100`	Pass	0.002s
Synthetic elevation should be interpolated between flanking real gauges.	`min_real <= elev <= max_real`	Pass	0.004s
No two synthetics within dedup distance.	`d >= SYNTH_DEDUP_DISTANCE_M`	Pass	0.016s
Synthetic has elevation	`elev > 0`	Pass	0.002s
Synthetic has flood stages	`stages["FloodAlert"] > 0`; `stages["FloodWarning"] > stages["FloodAlert"]` (+1 more)	Pass	0.002s
Synthetic gauges should be near the river polyline.	`min_dist < 1000`	Pass	0.033s
Synthetics exist	`len(synths) > 0`	Pass	0.002s
loadPortfolioImpact() must be defined.	`'loadPortfolioImpact' in table_js`	Pass	0.000s
Portfolio impact fetch must use the correct endpoint path.	`'portfolio-impact' in table_js`	Pass	0.000s
Sort state variables must be initialised.	`'spSortCol' in table_js`; `'spSortAsc' in table_js`	Pass	0.000s
Summary card labels must be present.	`'Properties Affected' in table_js`; `'Total Damage' in table_js`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Alt rows amber	`len(ALT_ROWS_AMBER) == 1; ALT_ROWS_AMBER[0][0] == 'ROWBACKGROUNDS'`	Pass	0.000s
Alt rows blue	`len(ALT_ROWS_BLUE) == 1; ALT_ROWS_BLUE[0][0] == 'ROWBACKGROUNDS'`	Pass	0.000s
Alt rows green	`len(ALT_ROWS_GREEN) == 1; ALT_ROWS_GREEN[0][0] == 'ROWBACKGROUNDS'`	Pass	0.000s
Hdr style first command is background	`HDR_STYLE[0][0] == 'BACKGROUND'`	Pass	0.000s
Hdr style has grid	`'GRID' in cmds; 'BOX' in cmds`	Pass	0.000s
Hdr style is list	`isinstance(HDR_STYLE, list); len(HDR_STYLE) > 0`	Pass	0.000s
Only severe threshold should exist in term_structure.	`"severe" in ts; "any_flood" not in ts` (+1 more)	Pass	0.099s
PRS spread should be non-negative for all tenors.	`all(s >= 0 for s in spreads)`	Pass	0.098s
All tenors should have the same spread (events are independent).	`len(set(spreads)) == 1`	Pass	0.098s
Property Spread row should show adjustedPropSpread.	`'adjustedPropSpread.toFixed(1)' in js`	Pass	0.000s
Detail column should show the selected zone name.	`"selectedZone + '</td></tr>'" in js`	Pass	0.000s
Negative terrain delta (risk reduction) should be green.	`"#2E7D32" in js`	Pass	0.000s
Positive terrain delta (risk increase) should be orange.	`"#E65100" in js`	Pass	0.000s
Terrain delta variable	`'terrainDelta' in js`	Pass	0.000s
Terrain effect label	`'Terrain Effect' in js`	Pass	0.000s
Terrain row should have purple-tinted background.	`'#F3E5F5' in js`	Pass	0.000s
Terrain row should only appear when terrainDelta !== 0.	`'if (terrainDelta !== 0)' in js`	Pass	0.000s
terrainRow should be inserted in Path 1 (after Gauge Spread).	`idx_terrain1 < idx_distance1`	Pass	0.000s
terrainRow should be inserted in Path 2 (after header).	`idx_terrain2 < idx_elev2`	Pass	0.000s
Metadata has terrain grid	`grid is not None`	Pass	0.004s
Terrain grid distances	`len(grid["distances"]) == 21; grid["distances"][0] == 0` (+1 more)	Pass	0.004s

Test Description	Acceptance Criteria	Result	Time
Terrain grid elevations	`len(grid["elevations"]) == 11; grid["elevations"][0] == 0.0` (+1 more)	Pass	0.004s
Terrain grid has all zones	`set(grid["zones"].keys()) == set(EA_FLOOD_ZONE_RATES.keys())`	Pass	0.004s
Terrain grid has all zones	`set(grid'zones'].keys()) == set(EA_FLOOD_ZONE_RATES.keys())`	Pass	0.271s
Terrain grid has distances	`'distances' in grid; len(grid['distances']) == 21`	Pass	0.288s
Terrain grid has elevations	`'elevations' in grid; len(grid['elevations']) == 11`	Pass	0.272s
Terrain grid present	`'terrain_grid' in data['metadata']`	Pass	0.292s
Spread should decrease with distance for a given zone/elevation.	`g[i][0] >= g[i + 1][0]`	Pass	0.004s
Terrain grid zone3b gt zone1 at origin	`s3b > s1`	Pass	0.004s
Terrain grid zone dimensions	`len(zdata["grid"]) == 21; len(row) == 11`	Pass	0.004s
Terrain grid zone has grid array	`'grid' in zone_data; 'rate' in zone_data` (+2 more)	Pass	0.290s
Zone 3b spread ¿ Zone 3a ¿ Zone 2 ¿ Zone 1 at origin (d=0, h=0).	`s3b > s3a > s2 > s1`	Pass	0.004s
Moving from Zone 3b to Zone 1 should yield negative delta.	`delta < 0`	Pass	0.008s
Moving from Zone 1 to Zone 3b should yield positive delta.	`delta > 0`	Pass	0.008s
Delta between same zone should be exactly zero.	`abs(s1 - s2) < 1e-12`	Pass	0.008s
Ensure the Python reference produces known values at grid points.	`val > 0; val > val1`	Pass	0.009s
Actual zone from phcdata	`'phcData.flood_zone' in js`	Pass	0.000s
Avg distance computed from gauges	`'avgDistM' in js; 'nearestGauges' in js`	Pass	0.000s
Core bilinear formula should be present.	`'v00 * (1 - td) * (1 - te)' in js`	Pass	0.000s
When selectedZone === actualZone, terrainDelta should stay 0.	`'selectedZone !== actualZone' in js`	Pass	0.000s
Elevation diff computed	`'avgElevDiff' in js; 'phcData.elevation_m' in js`	Pass	0.000s
Distance and elevation should be clamped to grid bounds.	`'Math.max(distances[0]' in js; 'Math.min(distance_m' in js`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Interpolate accepts terrain grid	'terrainGrid' in js	Pass	0.000s
Interpolate function present	'function interpolateTerrainSpread(' in js	Pass	0.000s
Return includes terrain fields	'terrainDelta: terrainDelta' in js; 'selectedZone: selectedZone' in js (+1 more)	Pass	0.000s
Selected zone read from dropdown	"getElementById('phc-ea-zone')" in js	Pass	0.000s
Terrain delta computed	'terrainDelta' in js	Pass	0.000s
Coverage booleans are bool	isinstance(val, bool)	Pass	0.000s
Coverage has required keys	key in tc	Pass	0.000s
Coverage has file	"test_file" in tc; isinstance(tc["test_file"], str)	Pass	0.000s
GAUGE-001, GAUGE-002, GAUGE-9042bd95 are backward-compat exceptions.	compat_ids == {'GAUGE-001', 'GAUGE-002', 'GAUGE-9042bd95'}	Pass	0.000s
gauge.json and gaugehc.json gauge IDs must be identical sets.	gauge_json_ids == gaugehc_ids	Pass	0.000s
The new gauge entries in SAMPLE_GAUGEHC follow GAUGE-{8hex}.	not violations	Pass	0.000s
New gauge IDs added in conftest (GAUGE-{8hex}) follow the protocol.	GAUGE_ID_RE.match(gid)	Pass	0.000s
Gauge IDs in SAMPLE_STRESS_STORMS gauge_responses are from _data.	not violations	Pass	0.000s
_data.py STORM_PORT_* constants follow the protocol.	STORM_ID_RE.match(STORM_PORT_SEVERE); STORM_ID_RE.match(STORM_PORT_ALERT)	Pass	0.000s
All storm_id values in SAMPLE_PORT_STRESS_STORMS follow STORM-{8hex}.	STORM_ID_RE.match(sid)	Pass	0.000s
All storm_id values in SAMPLE_STRESS_STORMS follow STORM-{8hex}.	STORM_ID_RE.match(sid)	Pass	0.000s
_data.py STORM_SEVERE and STORM_WARNING follow the protocol.	STORM_ID_RE.match(STORM_SEVERE); STORM_ID_RE.match(STORM_WARNING)	Pass	0.000s
Has lines field	"lines" in data; isinstance(data["lines"], list)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Has running field	`"running" in data; isinstance(data["running"], bool)`	Pass	0.000s
Lines includes appended	`"test line" in data["lines"]`	Pass	0.000s
Not running initially	`data["running"] is False`	Pass	0.000s
Returns 200	`resp.status_code == 200`	Pass	0.000s
Returns not found when no file	`data["status"] == "not_found"`	Pass	0.000s
Returns ok when file exists	`data["status"] == "ok"`	Pass	0.002s
Returns summary fields	`"summary" in data; "failures" in data` (+3 more)	Pass	0.002s
Running field is bool	`"running" in data; isinstance(data["running"], bool)`	Pass	0.000s
Historical high exceeds severe by less than 1m	`"Elevated risk" in str(elements)`	Pass	0.000s
Historical high exceeds severe by more than 1m	`"High risk" in str(elements)`	Pass	0.000s
Historical high within severe	`"within severe warning levels" in str(elements)`	Pass	0.000s
Moderate alert range	`"Moderate alert range" in str(elements)`	Pass	0.000s
Narrow alert range	`"Narrow alert range" in str(elements)`	Pass	0.000s
Wide alert range	`"Wide alert range" in str(elements)`	Pass	0.000s
Alert status	`"ELEVATED RISK" in str(self._elements("Alert"))`	Pass	0.001s
Normal status	`"NORMAL" in str(self._elements("Normal"))`	Pass	0.001s
Severe status	`"CRITICAL RISK" in str(self._elements("Severe"))`	Pass	0.001s
Timeseries with no level	`isinstance(elements, list)`	Pass	0.000s
Timeseries with no readings	`isinstance(elements, list)`	Pass	0.000s
Warning status	`"HIGH RISK" in str(self._elements("Warning"))`	Pass	0.001s
Test getting timeseries for a gauge from portfolio.	`len(active_gauges) > 0; len(readings) > 0`	Pass	0.002s
Test that timeseries gauge IDs exist in portfolio.	`gauge is not None`	Pass	0.002s
Count returns number of gauge files	`TimeseriesLoader(tmp_path).count() == 2`	Pass	0.001s
Get available gauge ids	`"GAUGE-001" in ids; "GAUGE-002" in ids`	Pass	0.001s

Test Description	Acceptance Criteria	Result	Time
Get available gauge ids missing dir	`TimeseriesLoader(tmp_path / "nonexistent").get_available_...`	Pass	0.000s
List all returns summaries	`isinstance(TimeseriesLoader(tmp_path).list_all(), list)`	Pass	0.002s
Load all from directory	`len(TimeseriesLoader(tmp_path).load_all()) >= 1`	Pass	0.005s
Missing dir returns empty	`TimeseriesLoader(tmp_path / "nonexistent").load_all() == []`	Pass	0.000s
Test that entity ID is the timestamp.	`entity_id is not None; "2025-01-01" in entity_id`	Pass	0.002s
Test timeseries summary generation.	`len(summaries) == 3; 'timestamp' in summary` (+3 more)	Pass	0.003s
Test loading timeseries data.	`len(records) == 3`	Pass	0.003s
Clear cache	`not loader._cache_valid`	Pass	0.001s
Force reload	`"GAUGE-002" in loader.get_available_gauge_ids()`	Pass	0.002s
Invalidate cache	`not loader._cache_valid`	Pass	0.000s
Test getting most recent reading.	`latest is not None; latest['hour'] == 2` (+1 more)	Pass	0.001s
Test getting peak (highest) reading.	`peak is not None; peak['level'] == 4.8`	Pass	0.002s
Test getting readings for a specific gauge.	`len(readings) == 3; 'timestamp' in reading` (+1 more)	Pass	0.002s
Test getting readings using gauge name.	`len(readings) == 3`	Pass	0.001s
Test getting readings for nonexistent gauge.	`readings == []`	Pass	0.000s
Readings with unparseable timestamps are skipped (all-gauges path).	`len(result) == 1`	Pass	0.003s
Readings with unparseable timestamps are skipped (gauge_id path).	`len(result) == 1`	Pass	0.028s
Get readings in range all gauges	`isinstance(result, list)`	Pass	0.002s
Get readings in range for gauge	`isinstance(result, list)`	Pass	0.002s
Get lareading	`TimeseriesLoader(tmp_path).get_latest_reading("GAUGE-001")`	Pass	0.001s.
Get lareading missing returns none	`TimeseriesLoader(tmp_path).get_latest_reading("GAUGE-999")`	Pass	0.000s.
Get peak reading	`TimeseriesLoader(tmp_path).get_peak_reading("GAUGE-001") ...`	Pass	0.001s
Get peak reading missing returns none	`TimeseriesLoader(tmp_path).get_peak_reading("GAUGE-999") ...`	Pass	0.000s
Get readings for gauge	`isinstance(result, list) and len(result) >= 1`	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Gauge ID not found — fallback to gauge_name search.	`isinstance(result, list)`	Pass	0.003s
Get readings missing gauge returns empty	`TimeseriesLoader(tmp_path).get_readings_for_gauge("GAUGE-0...`	Pass	0.000s
Get storm responses	`len(responses) == 1; responses[0]["storm_id"] == "STORM-0001"`	Pass	0.004s
Get storm responses missing gauge	`TimeseriesLoader(tmp_path).get_storm_responses("GAUGE-999...`	Pass	0.000s
Test getting list of gauge IDs in timeseries.	`len(gauge_ids) == 2; "GAUGE-001" in gauge_ids` (+1 more)	Pass	0.001s
Test calculating gauge statistics.	`stats['gauge_id'] == "GAUGE-001"; stats['reading_count'] == 3` (+4 more)	Pass	0.002s
Get gauge statistics missing gauge	`stats["reading_count"] == 0; "error" in stats`	Pass	0.000s
Readings with no numeric level field return error in stats.	`"error" in stats`	Pass	0.011s
Test statistics for non-existent gauge.	`stats['reading_count'] == 0; 'error' in stats`	Pass	0.000s
Get gauge statistics with data	`"reading_count" in stats; stats["reading_count"] >= 1`	Pass	0.001s
Exists by timestamp	`loader.exists(result[0]["timestamp"]) is True`	Pass	0.001s
Find by id not found returns none	`TimeseriesLoader(tmp_path).find_by_id("1999-01-01T00:00:00...`	Pass	0.001s
Get file status alias	`"entity_name" in TimeseriesLoader(tmp_path).get_file_stat...`	Pass	0.000s
Get status returns dict	`"entity_name" in status; "exists" in status`	Pass	0.000s
Test getting readings in time range.	`len(readings) == 2`	Skip	0.000s
Test getting readings in range for specific gauge.	`len(readings) >= 2; reading.get('gaugeId') == "GAUGE-001" or 'level' in reading`	Skip	0.000s
Bad gauge data is skipped	`len(result["readings"]) >= 1`	Pass	0.000s
Avg per gauge seconds	`abs(result - 160 / 15) < 0.001`	Pass	0.000s
Avg per gauge seconds empty	`_avg_per_gauge_seconds({"runs": []}) == 0.0`	Pass	0.000s
Load timings corrupt json	`result == {"runs": []}`	Pass	0.001s
Load timings missing file	`result == {"runs": []}`	Pass	0.006s
Load timings valid	`len(result["runs"]) == 1; result["runs"][0]["num_gauges"] == 5`	Pass	0.008s

Test Description	Acceptance Criteria	Result	Time
Save timings truncates to 20	`len(saved["runs"]) == 20; saved["runs"][0]["i"] == 10`	Pass	0.002s
ConstructionYear triggers building age row in summary table.	`isinstance(result, list); len(result) > 0`	Pass	0.000s
Empty property does not crash	`isinstance(result, list)`	Pass	0.000s
Missing nested key falls back to 'Unknown Property'.	`result == "Unknown Property"`	Pass	0.000s
Get property id present	`result == "PROP-001"`	Pass	0.000s
Has tables	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
Has title paragraph	`any("Comprehensive Property Analysis Report" in t for t i...`	Pass	0.000s
Property id in output	`any("PROP-001" in t for t in texts)`	Pass	0.000s
Property overview section header	`any("Property Overview" in t for t in texts)`	Pass	0.000s
Property summary section header	`any("Property Summary" in t for t in texts)`	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0`	Pass	0.000s
Unknown property id fallback	`any("Unknown Property" in t for t in texts)`	Pass	0.000s
With mortgage data does not crash	`isinstance(result, list)`	Pass	0.000s
Step owners covers all graph steps	`step in dl_mod.STEP_OWNERS`	Pass	0.000s
Topology renders owner	`len(tables) >= 1; 'Owner' in header_row`	Pass	0.000s
Gauge trace should find origin + derived + consumed steps.	`len(result) >= 3; "gauges" in steps` (+1 more)	Pass	0.003s
Each trace entry should have a context description.	`r.get("context")`	Pass	0.003s
Gauge with trained classifier should show classifiers step.	`"classifiers" in steps`	Pass	0.004s
Gauge referenced in PRS trades should show blotter step.	`len(consumed) >= 1`	Pass	0.003s
None lineage with no matching files returns empty.	`result == []`	Pass	0.004s
Non-existent gauge returns empty trace.	`result == []`	Pass	0.004s
Property trace should find origin and derived steps.	`len(result) >= 2; "properties" in steps`	Pass	0.001s
Trade trace should find the PRS file as origin.	`len(result) >= 1; result[0]["role"] == "origin"`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Missing data id	`r.status_code == 400`	Pass	0.001s
Missing data type	`r.status_code == 400`	Pass	0.001s
Missing data_type or data_id returns 400.	`r.status_code == 400; "required" in r.get_json()["message"]`	Pass	0.001s
Fallback scan when no direct traces key matches.	`r.status_code == 200; data["found"] is True` (+1 more)	Pass	0.004s
No lineage file =¿ found=False, empty trace.	`r.status_code == 200; data["found"] is False`	Pass	0.002s
Trace not found	`r.status_code == 200; data["found"] is False` (+1 more)	Pass	0.002s
Valid trace	`r.status_code == 200; data["status"] == "success"` (+4 more)	Pass	0.009s
Whitespace only params	`r.status_code == 400`	Pass	0.001s
Gauge id chain	`len(chain) == 4; chain[0]["step"] == "gauges"` (+4 more)	Pass	0.000s
Property id chain	`len(chain) == 7; all(c["recorded"] is False for c in chain)`	Pass	0.000s
Unknown data type	`chain == []`	Pass	0.000s
Posts to /api/v1/prs/commit.	`'/api/v1/prs/commit' in prs_js`	Pass	0.000s
Commit payload includes payer flag.	`'isPayer' in prs_js`	Pass	0.000s
Commit returns swap id	`data['status'] == 'success'; data['swap_id'].startswith('PRS-')`	Pass	0.007s
Commit writes to temp dir	`trade_file.exists()`	Pass	0.012s
Trade trace includes eod	`any("eod" in r["file"] for r in blotter_steps)`	Pass	0.002s
Trade trace includes marks	`any("trade_marks" in r["file"] for r in blotter_steps)`	Pass	0.001s
Panel should add to a Folium map without error.	`len(html._children) > 0`	Pass	0.025s
Blotter sub-module should render with FS01 column.	`'createBlotterView' in js; 'loadBlotterData' in js` (+2 more)	Pass	0.000s
EOD sub-module should render.	`'createEodView' in js; 'loadEodData' in js`	Pass	0.000s
Aggregate View sub-module should render.	`'createMapView' in js; 'loadMapData' in js` (+2 more)	Pass	0.000s
Market sub-module should render.	`'createMarketView' in js; 'loadMarketData' in js`	Pass	0.000s
Ensure no unescaped single braces that could cause NameError.	`len(js) > 1000`	Pass	0.000s
Panel should contain all 7 tab IDs and labels.	`'blotter' in js; 'market' in js` (+7 more)	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Panel should export TradingDesk to window.	'window.TradingDesk' in js; 'window.showTradingDesk' in js	Pass	0.000s
Panel should contain the Pi button.	'Π' in js	Pass	0.000s
TradingDeskPanel JS should render without errors.	len(js) > 0; '<script>' in js (+1 more)	Pass	0.000s
FS01 Matrix sub-module should render.	'createRiskView' in js; 'loadRiskData' in js (+1 more)	Pass	0.000s
get_statistics() returns panel_width and panel_height.	stats["panel_width"] == "420px"; stats["panel_height"] == "600px"	Pass	0.000s
Td preloader does not redeclare td pre blotter	'window._tdPreBlotter = null' not in src	Pass	0.000s
Td preloader does not redeclare td pre market	'window._tdPreMarket = null' not in src	Pass	0.000s
Td preloader does not redeclare td preload done	'var _tdPreloadDone = false' not in src	Pass	0.000s
POST train-all returns 500 when config.get_classifiers_dir raises.	resp.status_code == 500; data["status"] == "error" (+1 more)	Pass	0.009s
If training is already in progress, return training status.	resp.status_code == 200; data["status"] == "training" (+1 more)	Pass	0.000s
If joblib already exists, POST returns ready immediately.	resp.status_code == 200; data["status"] == "ready" (+1 more)	Pass	0.001s
Corrupt gauge.json =¿ 500.	resp.status_code == 500; data["status"] == "error" (+1 more)	Pass	0.002s
Gauge not in gauge.json =¿ 404.	resp.status_code == 404; data["status"] == "error" (+1 more)	Pass	0.003s
Missing gauge.json =¿ 404.	resp.status_code == 404; "gauge.json not found" in data["message"]	Pass	0.004s
Missing storm_sequences.json =¿ 404.	resp.status_code == 404; "storm_sequences.json not found" in data["message"]	Pass	0.003s
A previously failed job can be restarted.	resp.status_code == 200; data["status"] == "training"	Pass	0.003s
Successful POST starts training and returns training status.	resp.status_code == 200; data["status"] == "training" (+4 more)	Pass	0.005s
severe_warning=100m ensures no events breach it -¿ fallback to alert.	result["label_threshold"] == "flood_alert"	Pass	6.021s
Model file written	stressm_dir.exists()	Pass	6.175s
Result has gauge id	result.get("gauge_id") == "GAUGE-clf001"	Pass	6.128s

Test Description	Acceptance Criteria	Result	Time
Result has label threshold	`"label_threshold" in result; result["label_threshold"] in ("severe_warning", "flood_al...`	Pass	6.094s
Returns result dict	`isinstance(result, dict)`	Pass	6.195s
severe_warning=0 exercises the else branch in log_hs computation.	`isinstance(result, dict)`	Pass	6.152s
Exception during training sets job status to 'failed'.	`job["status"] == "failed"; job["error"] is not None` (+1 more)	Pass	0.002s
Successful training sets job status to 'ready'.	`job["status"] == "ready"; job["error"] is None`	Pass	0.003s
Monthly rent and annual rent	`isinstance(result, list)`	Pass	0.000s
No transaction data	`isinstance(self._page().generate_elements(prop), list)`	Pass	0.000s
Rental yield computed	`isinstance(result, list)`	Pass	0.000s
Sale date shown	`isinstance(result, list)`	Pass	0.000s
Sale price and appreciation	`any(isinstance(e, Table) for e in result)`	Pass	0.000s
Sales section rendered	`any("Transaction" in t for t in texts)`	Pass	0.000s
Assess yield average	`page._assess_yield(5.0) == "Average yield"`	Pass	0.000s
Assess yield below average	`page._assess_yield(3.0) == "Below average yield"`	Pass	0.000s
Assess yield excellent	`page._assess_yield(9.0) == "Excellent yield"`	Pass	0.000s
Assess yield good	`page._assess_yield(7.0) == "Good yield"`	Pass	0.000s
Assess yield poor	`page._assess_yield(1.0) == "Poor yield"`	Pass	0.000s
Empty property does not crash	`isinstance(result, list)`	Pass	0.000s
Empty transactions	`isinstance(result, list)`	Pass	0.000s
Has transactions header	`any("Transaction" in t or "History" in t for t in texts)`	Pass	0.000s
Market analysis appreciation	`"Capital Appreciation" in result`	Pass	0.000s
Market analysis few vacancies	`"Good" in result["Rental Stability"]`	Pass	0.000s
Market analysis many vacancies	`"Concerning" in result["Rental Stability"]`	Pass	0.000s
Market analysis marketing above 30	`"Marketing Efficiency" in result; "Good" in result["Marketing Efficiency"]`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Market analysis marketing above 90	`"Below average" in result["Marketing Efficiency"]`	Pass	0.000s
Market analysis never rented	`"Rental History" in result; "No rental history" in result["Rental History"]`	Pass	0.000s
Market analysis no vacancy	`"Rental Stability" in result; "Excellent" in result["Rental Stability"]`	Pass	0.000s
Market analysis yield	`"Gross Rental Yield" in result`	Pass	0.000s
No transaction history	`isinstance(result, list)`	Pass	0.000s
Returns list	`isinstance(result, list); len(result) > 0`	Pass	0.001s
Sales extended marketing	`isinstance(result, list)`	Pass	0.000s
Sales quick sale	`isinstance(result, list)`	Pass	0.000s
Sales very long marketing	`isinstance(result, list)`	Pass	0.000s
With mortgage data	`isinstance(result, list)`	Pass	0.000s
With rental data	`isinstance(result, list); any(isinstance(e, Table) for e in result)`	Pass	0.000s
With sales data	`isinstance(result, list); any(isinstance(e, Table) for e in result)`	Pass	0.000s
Flood arrival hour should increase with distance.	`len(arrivals) >= 2; arrivals[i][1] >= arrivals[i - 1][1]`	Pass	0.001s
More distant properties should have shorter flood duration	`flooded_durations[i][1] <= flooded_durations[i - 1][1]`	Pass	0.001s
Deeper water flows faster → shorter travel time.	`times[i] < times[i - 1]`	Pass	0.000s
Travel time must increase monotonically with distance.	`times[i] > times[i - 1]`	Pass	0.000s
Gauges alert count	`single_storm_data["trigger_summary"]["gauges_alert"] == 2`	Pass	0.000s
Gauges impacted matches alert	`ts["gauges_impacted"] == ts["gauges_alert"]`	Pass	0.000s
Gauges severe count	`single_storm_data["trigger_summary"]["gauges_severe"] == 1`	Pass	0.000s
Gauges warning count	`single_storm_data["trigger_summary"]["gauges_warning"] == 1`	Pass	0.000s
Max trigger is alert when alert only	`data["storms"][0]["trigger_summary"]["max_trigger"] == "a...`	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Max trigger is severe	`single_storm_data["trigger_summary"]["max_trigger"] == "s...`	Pass	0.000s
Max trigger is warning when no severe	`data["storms"][0]["trigger_summary"]["max_trigger"] == "w...`	Pass	0.002s
Trigger summary present	`"trigger_summary" in single_storm_data`	Pass	0.000s
Assessment date updated	`r.get_json()["assessment"]["assessment_date"] is not None`	Pass	0.002s
No data file returns 404	`r.status_code == 404`	Pass	0.000s
Principle not found returns 404	`r.status_code == 404`	Pass	0.002s
Score 1 derives non compliant status	`r.status_code == 200; p["status"] == "Non-compliant"`	Pass	0.002s
Score 4 derives fully compliant status	`p["status"] == "Fully Compliant"`	Pass	0.002s
Update evidence field	`r.status_code == 200; p["evidence"] == "New evidence gathered"`	Pass	0.002s
Update gaps and remediation	`p["gaps"] == "Minor gap"; p["remediation"] == "Action plan created"`	Pass	0.002s
Valid update returns 200	`r.status_code == 200; r.get_json()["status"] == "success"`	Pass	0.002s
Not found returns 404	`r.status_code == 404`	Pass	0.002s
Update changes field	`ref["author"] == "New Author"`	Pass	0.004s
Update returns success	`r.status_code == 200; r.get_json()["status"] == "success"`	Pass	0.003s
Update with new category adds to list	`"NewCat" in r.get_json()["categories"]`	Pass	0.004s
Update sets updated at	`r.get_json()["meeting"]["updated_at"] is not None`	Pass	0.003s
Update title	`r.status_code == 200; r.get_json()["meeting"]["title"] == "Updated Title"`	Pass	0.003s
Update unknown id returns 404	`r.status_code == 404`	Pass	0.001s
Update action missing meeting returns 404	`r.status_code == 404`	Pass	0.002s
Update decision missing meeting returns 404	`r.status_code == 404`	Pass	0.002s
Update minute item missing meeting returns 404	`r.status_code == 404`	Pass	0.002s
Update participant missing meeting returns 404	`r.status_code == 404`	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Empty value returns 400	`r.status_code == 400`	Pass	0.002s
Invalid choice value returns 400	`r.status_code == 400`	Pass	0.002s
Invalid field returns 400	`r.status_code == 400; r.get_json()["status"] == "error"`	Pass	0.002s
Model not found returns 404	`r.status_code == 404; r.status_code == 404` (+1 more)	Pass	0.002s
No reason returns 400	`r.status_code == 400`	Pass	0.002s
Response has audit entry	`"audit_entry" in data; data["audit_entry"]["event_type"] == "field_update"`	Pass	0.003s
Update lifecycle stage	`r.status_code == 200`	Pass	0.003s
Update validation status	`r.status_code == 200`	Pass	0.003s
Valid choice field update returns success	`r.status_code == 200; r.get_json()["new_value"] == "Amber"`	Pass	0.003s
Valid text field update returns success	`r.status_code == 200; data["status"] == "success"` (+2 more)	Pass	0.003s
No inventory returns 404	`r.status_code == 404; r.get_json()["status"] == "error"` (+3 more)	Pass	0.001s
Save failure returns 500	`r.status_code == 500; r.get_json()["status"] == "error"`	Pass	0.002s
Activity not found returns 404	`r.status_code == 404`	Pass	0.002s
Invalid role id returns 400	`r.status_code == 400`	Pass	0.002s
Notes updated	`activities[0]["notes"] == "New note"`	Pass	0.003s
Setting C or I to None (null) is valid.	`r.status_code == 200`	Pass	0.003s
Valid activity update returns success	`r.status_code == 200; r.get_json()["status"] == "success"`	Pass	0.003s
Escalation tier threshold updated	`triggers[0]["tier_threshold"]["1"] == "Immediate"`	Pass	0.003s
Response has audit entry	`"audit_entry" in r.get_json(); "audit_entry" in r.get_json()`	Pass	0.003s
Trigger not found returns 404	`r.status_code == 404`	Pass	0.002s
Valid escalation update returns success	`r.status_code == 200; r.get_json()["status"] == "success"`	Pass	0.003s
Backup field updated	`owner["backup"] == "Owner B"`	Pass	0.003s
No raci file returns 404	`r.status_code == 404`	Pass	0.029s
Response has audit entry	`"audit_entry" in r.get_json(); "audit_entry" in r.get_json()`	Pass	0.003s
Role not found returns 404	`r.status_code == 404`	Pass	0.002s
Valid role update returns success	`r.status_code == 200; r.get_json()["status"] == "success"`	Pass	0.003s

Test Description	Acceptance Criteria	Result	Time
When all gauges are skipped, avg_auc_roc is 0.	`summary["num_trained"] == 0`; `summary["avg_auc_roc"] == 0.0`	Pass	0.001s
Creates training_summary.json when it does not exist.	`summary_path.exists()`; `summary["num_gauges"] == 1` (+5 more)	Pass	0.001s
Merges a new gauge result into an existing summary.	`summary["num_gauges"] == 2`; `summary["num_trained"] == 2` (+2 more)	Pass	0.003s
Re-training a gauge replaces its entry, not duplicates it.	`summary["num_gauges"] == 1`; `summary["num_trained"] == 1` (+2 more)	Pass	0.002s
A 'skipped' gauge does not contribute to avg_auc_roc.	`summary["num_gauges"] == 2`; `summary["num_trained"] == 1` (+2 more)	Pass	0.004s
All valid statuses	`r.status_code == 200`	Pass	0.011s
Invalid status returns 400	`r.status_code == 400`; `r.get_json()["status"] == "error"`	Pass	0.002s
Model not found returns 404	`r.status_code == 404`; `r.get_json()["status"] == "error"` (+1 more)	Pass	0.002s
Question not found returns 404	`r.status_code == 404`	Pass	0.002s
Risk rating recalculated	`rr["last_calculated"] is not None`	Pass	0.003s
Valid update has model and audit entry	`"model" in data`; `"audit_entry" in data`	Pass	0.003s
Valid update returns success	`r.status_code == 200`; `r.get_json()["status"] == "success"`	Pass	0.003s
update_yield_curve creates yield_curve from default when missing.	`'1' in updated['yield_curve']`; `updated['yield_curve']['1'] == 0.055`	Pass	0.003s
Empty filename returns 400	`r.status_code == 400`	Pass	0.001s
No file returns 400	`r.status_code == 400`; `data["status"] == "error"`	Pass	0.000s
Upload appears in list	`len(data["documents"]) == 1`	Pass	0.003s
Upload response has id	`"id" in data["document"]`	Pass	0.002s
Upload stores description	`data["document"]["description"] == "My description"`	Pass	0.002s
Valid upload returns 200	`r.status_code == 200`; `data["status"] == "success"` (+1 more)	Pass	0.002s
Upload save failure returns 500	`r.status_code == 500`; `data["status"] == "error"` (+1 more)	Pass	0.158s

Test Description	Acceptance Criteria	Result	Time
_add with file outside input_dir falls back to absolute path.	`len(eod_entries) >= 1`	Pass	0.001s
_search_dir_files matches entity_id in filename stem — exercised	`len(eod_entries) >= 1`	Pass	0.001s
_search_dir_files with nonexistent dir returns [] — exercised by	`isinstance(result, list)`	Pass	0.002s
_search_json with unreadable file returns False — exercised by	`isinstance(result, list)`	Pass	0.003s
Get page categories contains expected keys	`key in cats`	Pass	0.001s
Get page categories contains expected keys	`key in cats`	Pass	0.001s
Get page categories returns copy	`"__new_key__" not in gen.categories`	Pass	0.001s
Get page categories returns copy	`"__new_key__" not in gen.categories`	Pass	0.002s
Mutating the returned dict must not affect the generator's internal state.	`"__new_key__" not in gen.categories`	Pass	0.001s
Get page categories returns dict	`isinstance(cats, dict)`	Pass	0.001s
Get page categories returns dict	`isinstance(cats, dict)`	Pass	0.002s
Get page categories returns dict	`isinstance(cats, dict)`	Pass	0.001s
List available pages contains all expected	`expected <= set(pages)`	Pass	0.002s
List available pages contains all expected	`expected <= set(pages)`	Pass	0.001s
List available pages contains expected	`expected <= set(pages)`	Pass	0.001s
List available pages non empty	`len(gen.list_available_pages()) > 0`	Pass	0.001s
List available pages non empty	`len(gen.list_available_pages()) > 0`	Pass	0.001s
List available pages returns list	`isinstance(pages, list)`	Pass	0.001s
List available pages returns list	`isinstance(pages, list)`	Pass	0.002s
List available pages returns list	`isinstance(pages, list)`	Pass	0.001s
Validate pages all invalid	`valid == []`; `"bogus_one" in invalid` (+1 more)	Pass	0.001s
Validate pages all invalid	`valid == []`; `"no_such_page" in invalid` (+1 more)	Pass	0.002s
Validate pages all invalid	`valid == []`; `"bogus_one" in invalid` (+1 more)	Pass	0.001s
Validate pages all valid	`"title_overview" in valid`; `"sensor_details" in valid` (+1 more)	Pass	0.001s
Validate pages all valid	`"title_overview" in valid`; `"data_summary" in valid` (+1 more)	Pass	0.002s

Test Description	Acceptance Criteria	Result	Time
Validate pages all valid	"title" in valid; "appendix" in valid (+1 more)	Pass	0.001s
Validate pages empty list	valid == []; invalid == []	Pass	0.001s
Validate pages empty list	valid == []; invalid == []	Pass	0.002s
Validate pages empty list	valid == []; invalid == []	Pass	0.001s
Validate pages mixed	"title_overview" in valid; "nonexistent_page" in invalid	Pass	0.001s
Validate pages mixed	"title_overview" in valid; "nonexistent_page" in invalid	Pass	0.002s
Validate pages mixed	"title" in valid; "nonexistent_page" in invalid	Pass	0.001s
The error message should be a string representation of the exception.	"validation_error" in result; isinstance(result["validation_error"][0], str)	Pass	0.000s
Another path to trigger the except branch.	"validation_error" in result	Pass	0.000s
Lines 251-252: passing data that causes an exception in the try	"validation_error" in result; isinstance(result["validation_error"], list) (+1 more)	Pass	0.000s
Consistent chain	result["is_consistent"]; result["stale_steps"] == [] (+1 more)	Pass	0.002s
Empty manifest	not result["is_consistent"]; len(result["missing_steps"]) == 15	Pass	0.000s
Stale step detected	not result["is_consistent"]; "properties" in result["stale_steps"]	Pass	0.000s
Alt key structure	validate_gauge_exists("GAUGE-002", gauge_file) is True	Pass	0.001s
Existing gauge returns true	validate_gauge_exists("GAUGE-001", gauge_file) is True	Pass	0.001s
Missing file returns false	validate_gauge_exists("GAUGE-001", tmp_path / "missing.js...	Pass	0.000s
Nonexistent gauge returns false	validate_gauge_exists("GAUGE-XXX", gauge_file) is False	Pass	0.001s
Binary file returns false	result is False	Pass	0.001s
Corrupt json returns false	result is False	Pass	0.001s
Zero validation errors across the whole batch.	result["invalid"] == 0	Pass	0.000s
Medium batch all valid	result["invalid"] == 0	Pass	0.001s
No negative durations	storm.duration_hours > 0	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
No negative precipitation	seq.total_precipitation_mm > 0; storm.precipitation_mm > 0	Pass	0.000s
Last storm precipitation must end by hour 156 (1e-9 FP tolerance).	last_end <= MAX_PRECIP_END_HOUR + _FP_TOL	Pass	0.000s
Line 196: audit log overflow during VQ update.	r.status_code == 200	Pass	0.036s
Line 130: inventory not found → 404.	r.status_code == 404	Pass	0.001s
Line 176: save failure → 500.	r.status_code == 500	Pass	0.002s
Update triggers risk recalculation	rr["calculated_score"] is not None; rr["calculated_rating"] in ["Acceptable", "Conditional", ... (+1 more)	Pass	0.006s
Update validation question	response.status_code == 200; data["status"] == "success" (+5 more)	Pass	0.003s
Update validation question invalid status	response.status_code == 400	Pass	0.002s
Update validation question not found	response.status_code == 404	Pass	0.002s
Base pct wins when higher than depth factor	var == pytest.approx(350_000)	Pass	0.000s
Depth 1 to 2m	var == pytest.approx(500_000)	Pass	0.000s
Depth half to 1m	var == pytest.approx(250_000)	Pass	0.000s
Depth over 2m	var == pytest.approx(800_000)	Pass	0.000s
Depth under half m	var == pytest.approx(100_000)	Pass	0.000s
Negative property value	RiskAssessor.calculate_value_at_risk(-100_000, "High") ==...	Pass	0.000s
Risk level percentages	var == pytest.approx(1_000_000 * pct)	Pass	0.000s
Zero depth uses base pct	var == pytest.approx(150_000)	Pass	0.000s
Zero property value	RiskAssessor.calculate_value_at_risk(0, "High") == 0.0	Pass	0.000s
When verbose=False, the per-counterparty log lines are absent.	"[1/2]" not in caplog.text; "Wrote 2 counterparties" not in caplog.text	Pass	0.000s
Lines 169-170: when verbose=True, each counterparty is logged.	len(result["data"]) == 3; "[1/3]" in caplog.text (+2 more)	Pass	0.001s
Line 187: verbose=True logs final summary with count and path.	"Wrote 2 counterparties" in caplog.text	Pass	0.000s
create_visualization() returns a TCEventVisualization instance.	isinstance(viz, TCEventVisualization)	Pass	0.000s
All addressed	_vq_summary(model) == "3/3"	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Empty questions	`_vq_summary(model) == "0/0"`	Pass	0.000s
None addressed	`_vq_summary(model) == "0/2"`	Pass	0.000s
Some not applicable	`_vq_summary(model) == "1/2"`	Pass	0.000s
_walk_upstream should not infinite loop on revisited nodes.	`result.count("gauges") == 1`	Pass	0.000s
Deep chain	`"gauges" in result; "propertyhc" in result`	Pass	0.000s
Mid chain	`"gauges" in result; result.index("gauges") < result.index("properties")`	Pass	0.000s
Root has no upstream	`result == ["gauges"]`	Pass	0.000s
GET returns yield_curve dict.	`resp.status_code == 200; data['status'] == 'success'` (+1 more)	Pass	0.003s
Default yield curve has standard tenor points.	`len(yc) > 0`	Pass	0.002s
Tenor outside 1-30 returns 400.	`resp.status_code == 400`	Pass	0.000s
POST without JSON body returns error.	`resp.status_code in (400, 500)`	Pass	0.002s
POST updates one tenor point.	`resp.status_code == 200; data['status'] == 'success'` (+1 more)	Pass	0.003s
Commit returns affected_trades and P&L impact fields.	`resp.status_code == 200; data['status'] == 'success'` (+4 more)	Pass	0.032s
Commit should affect all open trades.	`data['affected_trades'] == 16`	Pass	0.027s
Empty rates dict returns 400.	`resp.status_code == 400`	Pass	0.000s
No JSON body returns error.	`resp.status_code in (400, 500)`	Pass	0.001s
Reset returns to default shape.	`resp.status_code == 200; data['status'] == 'success'` (+1 more)	Pass	0.005s
In SHD mode (zero elevation diff), zone should be Zone 3b.	`FloodMixin._zone_from_offset(0.0) == 'Zone 3b'`	Pass	0.000s
In SHE mode (zero distance), zone should reflect real elevation.	`FloodMixin._zone_from_offset(2.0) == 'Zone 2'; FloodMixin._zone_from_offset(0.3) == 'Zone 3b'`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
The zone should be derived from prop_elevation - synthetic_elevation,	`FloodMixin._zone_from_offset(1.0) == 'Zone 3a'; FloodMixin._zone_from_offset(0.3) == 'Zone 3b'` (+1 more)	Pass	0.000s
All zone options present	`zone in js`	Pass	0.000s
The change listener array must include the zone dropdown ID.	`"'phc-ea-zone'" in js`	Pass	0.000s
Default selection reads from phcData.flood_zone.	`'phcData.flood_zone' in js`	Pass	0.000s
Label present	`'EA Zone:' in js`	Pass	0.000s
Select element present	`'id="phc-ea-zone"' in js`	Pass	0.000s
Zones should appear in risk order: 3b, 3a, 3, 2, 1.	`idx_3b < idx_3a < idx_3 < idx_2 < idx_1`	Pass	0.000s
Bounded zones have finite upper[zone 2] [zone='Zone 2']	`hi is not None; hi > 0`	Pass	0.000s
Bounded zones have finite upper[zone 3a] [zone='Zone 3a']	`hi is not None; hi > 0`	Pass	0.000s
Bounded zones have finite upper[zone 3b] [zone='Zone 3b']	`hi is not None; hi > 0`	Pass	0.000s
Upper bound of each zone == lower bound of the next.	`hi == lo`	Pass	0.000s
Has four zones	`set(EA_FLOOD_ZONE_ELEVATION_BOUNDS.keys()) == set(self.EX...`	Pass	0.000s
Lower bound non negative[zone 1] [zone='Zone 1']	`lo >= 0.0`	Pass	0.000s
Lower bound non negative[zone 2] [zone='Zone 2']	`lo >= 0.0`	Pass	0.000s
Lower bound non negative[zone 3a] [zone='Zone 3a']	`lo >= 0.0`	Pass	0.000s
Lower bound non negative[zone 3b] [zone='Zone 3b']	`lo >= 0.0`	Pass	0.000s
Lower bounds strictly increasing	`lowers[i] < lowers[i + 1]`	Pass	0.000s
Zone 1 has no upper bound	`hi is None`	Pass	0.000s
Zone 3b starts at zero	`lo == 0.0`	Pass	0.000s
Boundary[0.0-zone 3b] [offset=0.0, expected='Zone 3b']	`FloodMixin._zone_from_offset(offset) == expected`	Pass	0.000s
Boundary[0.25-zone 3b] [offset=0.25, expected='Zone 3b']	`FloodMixin._zone_from_offset(offset) == expected`	Pass	0.000s
Boundary[0.49-zone 3b] [offset=0.49, expected='Zone 3b']	`FloodMixin._zone_from_offset(offset) == expected`	Pass	0.000s

Test Description	Acceptance Criteria	Result	Time
Boundary[0.5-zone 3a] [offset=0.5, expected='Zone 3a']	`FloodMixin._zone_from_offset(offset) == expected`	Pass	0.000s
Boundary[1.0-zone 3a] [offset=1.0, expected='Zone 3a']	`FloodMixin._zone_from_offset(offset) == expected`	Pass	0.000s
Boundary[1.49-zone 3a] [offset=1.49, expected='Zone 3a']	`FloodMixin._zone_from_offset(offset) == expected`	Pass	0.000s
Boundary[1.5-zone 2] [offset=1.5, expected='Zone 2']	`FloodMixin._zone_from_offset(offset) == expected`	Pass	0.000s
Boundary[2.5-zone 2] [offset=2.5, expected='Zone 2']	`FloodMixin._zone_from_offset(offset) == expected`	Pass	0.000s
Boundary[2.99-zone 2] [offset=2.99, expected='Zone 2']	`FloodMixin._zone_from_offset(offset) == expected`	Pass	0.000s
Boundary[3.0-zone 1] [offset=3.0, expected='Zone 1']	`FloodMixin._zone_from_offset(offset) == expected`	Pass	0.000s
Boundary[5.0-zone 1] [offset=5.0, expected='Zone 1']	`FloodMixin._zone_from_offset(offset) == expected`	Pass	0.000s
Boundary[50.0-zone 1] [offset=50.0, expected='Zone 1']	`FloodMixin._zone_from_offset(offset) == expected`	Pass	0.000s
Negative offset (property below river) should still give Zone 3b.	`FloodMixin._zone_from_offset(0.0) == 'Zone 3b'`	Pass	0.000s
All four EA flood zones should appear in the portfolio.	`zone in stats and len(stats[zone]["flood_counts"]) > 0`	Pass	0.004s
Average spreads should decrease from Zone 3b → 3a → 2 → 1.	`avgs[present[i]] >= avgs[present[i + 1]]`	Pass	0.097s
Zone 3b properties should flood more than Zone 1 on average.	`avg_3b > avg_1`	Pass	0.004s
Zone 3b properties should have higher spreads than Zone 1 on average.	`avg_3b > avg_1`	Pass	0.004s
Zone 3b properties should be closer to gauge level than Zone 1.	`avg_3b < avg_1`	Pass	0.004s
Average offset above gauge should increase from Zone 3b → 3a → 2 → 1.	`avgs[present[i]] <= avgs[present[i + 1]]`	Pass	0.004s