

1. Testing Levels Covered

- Unit Testing: Done at the function/method level. (e.g., login() or add_samsung_to_cart() methods)
- Functional Testing: Each feature tested for expected behavior (e.g., login, cart addition)
- Smoke Testing: Used @pytest.mark.smoke to verify critical paths like login and add-to-cart
- Regression Testing: Used @pytest.mark.regression to ensure new features don't break old ones

Level	Description
Unit Testing	Function-level validation like login() or add_samsung_to_cart()
Functional Testing	Validates features like login or cart addition per requirements
Smoke Testing	Uses @pytest.mark.smoke to confirm core flows like login/cart work
Regression Testing	Ensures existing flows are stable with new changes using markers

2. Test Structure

Layer	Purpose
tests/	Contains all test cases (e.g., test_login.py, test_login_and_cart.py)
pages/	Page Object Model classes like LoginPage, CartPage
utils/	Helpers like take_screenshot(), popup handlers, Excel logging, logger

3. Test Design Approach

- Page Object Model (POM): Each web page is a class with methods for user actions and locators
- Modular Code: Reusable actions like login(), add_to_cart() etc., for test maintainability
- Utility Helpers: Abstracted logic for screenshots, logging, alerts, and Excel reporting

4. Assertion Strategy

Test Case	Assertion Used
test_login.py	assert "Welcome" not in page_source
test_cart_addition()	assert "Samsung galaxy s6" in page_source
test_product_price_assertion()	assert "\$360" in page_source

5. Test Reporting

- HTML Report: Generated using pytest-html with:
pytest.main(["--html=reports/test_cart_report.html", "--self-contained-html"])
- Excel Report: Custom Excel log written via openpyxl to test_results.xlsx

6. Reusability and Maintainability

- Page classes like HomePage and CartPage can be extended easily
- Utilities like take_screenshot(), log_to_excel(), handle_popup() allow reuse across tests

7. Error Handling Strategy

- All test logic is wrapped in try...except
- Failures trigger screenshot and Excel logging
- logger captures detailed tracebacks
- Popup alerts notify on UI about execution status

8. Test Strategy Summary

Feature	Description
Framework Type	Selenium with PyTest using Page Object Model
Execution Tools	PyTest CLI, HTML Report, Excel Log
Assertion Style	Standard PyTest assert statements
Error Recovery	Try-Except + Alerts + Screenshots + Logging
Modular Utilities	Screenshot, Logger, Excel Log, Popups
Test Coverage	Login, Cart Addition, Price Assertion, Alerts
Markers Used	@pytest.mark.smoke, @pytest.mark.regression