

# FPS Codebase - Complete File Location Map & Dependency Graph

**Purpose:** Quick reference for locating files and understanding how components interact

## BACKEND FILE LOCATIONS

### Root Directory Structure

fpsV2/	
├─ app/	# Application code
│   └─ Agents/	# Agent orchestrators (7 files)
│   └─ Console/	# CLI commands
│   └─ Exceptions/	# Exception classes
│   └─ Http/	# Controllers, Requests, Middleware
│   └─ Jobs/	# Background queue jobs
│   └─ Mail/	# Email templates/classes
│   └─ Models/	# Eloquent models (39 files)
│   └─ Providers/	# Service providers
│   └─ Services/	# Business logic (50+ files)
├─ config/	# Configuration files
│   └─ app.php	# App config
│   └─ database.php	# Database config
│   └─ cache.php	# Cache config (Memcached)
│   └─ queue.php	# Queue driver (database)
│   └─ uk_tax_config.php	# CRITICAL: All UK tax rules
├─ database/	
│   └─ migrations/	# Database schema migrations
│   └─ factories/	# Model factories for testing
│   └─ seeders/	# Database seeders
├─ routes/	
│   └─ api.php	# CRITICAL: All API routes (420+ lines)
│   └─ web.php	# Web routes (not used in SPA)
├─ resources/	
│   └─ js/	# Vue.js 3 frontend
├─ tests/	# Test suite (Pest)
│   └─ Unit/	# Unit tests
│   └─ Feature/	# Feature/integration tests
│   └─ E2E/	# End-to-end tests
├─ storage/	# File storage
│   └─ app/backups/	# Database backups (important!)
└─ bootstrap/, public/, lang/	# Framework files

### Agent Files (app/Agents/)

```

BaseAgent.php (152 lines)
├─ Methods: analyze(), generateRecommendations(), buildScenarios()
├─ Caching: remember($key, $callback, $ttl)
├─ Tax year: getCurrentTaxYear()
├─ Utilities: calculateAge(), roundToPenny(), formatCurrency()

ProtectionAgent.php
├─ Uses: CoverageGapAnalyzer, AdequacyScorer, RecommendationEngine,
ScenarioBuilder
├─ Returns: analysis with coverage gaps and recommendations

SavingsAgent.php
├─ Uses: EmergencyFundCalculator, ISATracker, GoalProgressCalculator
├─ Returns: savings analysis with ISA tracking

InvestmentAgent.php
├─ Uses: PortfolioAnalyzer, MonteCarloSimulator, AssetAllocationOptimizer
├─ Returns: portfolio analysis with Monte Carlo results

RetirementAgent.php
├─ Uses: PensionProjector, AnnualAllowanceChecker, ContributionOptimizer
├─ Returns: retirement projections and readiness score

EstateAgent.php (200+ lines)
├─ Uses: IHTCalculator, GiftingStrategy, NetWorthAnalyzer, CashFlowProjector
├─ Returns: IHT analysis with recommendations

CoordinatingAgent.php
├─ Calls: All other agents
├─ Aggregates: Unified recommendations
├─ Resolves: Conflicting recommendations

```

## Controllers (app/Http/Controllers/Api/)

### Auth & Core (6 files)

```

AuthController.php
├─ register(RegisterRequest)
├─ login(LoginRequest)
├─ logout()
├─ changePassword()
├─ user()

UserProfileController.php
├─ getProfile()
├─ updatePersonalInfo(UpdatePersonalInfoRequest)
├─ updateIncomeOccupation(UpdateIncomeOccupationRequest)

DashboardController.php
├─ index() → DashboardAggregator
├─ financialHealthScore()
├─ alerts()
├─ dismissAlert()

```

### User Management (4 files)

FamilyMembersController.php  
├─ CRUD for family members (spouse, children)  
└─ Spouse account linking

PersonalAccountsController.php  
├─ getPersonalAccounts()  
├─ calculatePersonalAccounts()  
└─ CRUD for line items (P&L, cashflow)

SpousePermissionController.php  
├─ Request permission to see spouse's data  
├─ Accept/reject permissions  
└─ Revoke permissions

AdminController.php (admin-only)  
├─ User management (CRUD)  
├─ Database backup/restore  
└─ Dashboard statistics

## Module Controllers (14 files)

ProtectionController.php  
├─ index() → all policy data  
├─ analyze() → ProtectionAgent.analyze()  
├─ recommendations() → ProtectionAgent.generateRecommendations()  
├─ scenarios() → ProtectionAgent.buildScenarios()  
└─ CRUD methods for policies (life, CI, IP, disability, sickness/illness)

SavingsController.php  
├─ index() → accounts + goals  
├─ analyze() → SavingsAgent.analyze()  
├─ recommendations()  
├─ scenarios()  
├─ storeAccount(), updateAccount(), deleteAccount()  
├─ storeGoal(), updateGoal(), deleteGoal()  
└─ isaAllowance(\$taxYear) → ISATracker.calculateAllowanceUsage()

InvestmentController.php  
├─ index() → accounts + holdings + goals  
├─ analyze() → InvestmentAgent.analyze()  
├─ recommendations()  
├─ scenarios()  
├─ startMonteCarlo() → Queue job  
├─ getMonteCarloResults(\$jobId)  
└─ CRUD for accounts, holdings, goals, risk profile

RetirementController.php  
├─ index() → pensions + state pension  
├─ analyze() → RetirementAgent.analyze()  
├─ recommendations()  
├─ scenarios()  
├─ checkAnnualAllowance(\$taxYear)  
└─ CRUD for DC/DB/state pensions

EstateController.php (400+ lines)

- └─ index() → Aggregates ALL assets (manual + investment + property + savings)
- └─ analyze() → EstateAgent.analyze()
- └─ recommendations() → EstateAgent.generateRecommendations()
- └─ scenarios() → EstateAgent.buildScenarios()
- └─ calculateIHT() → Calls IHTController
- └─ getNetWorth() → NetWorthAnalyzer
- └─ getCashFlow() → CashFlowProjector
- └─ CRUD: storeAsset(), updateAsset(), destroyAsset()  
       storeLiability(), updateLiability(), destroyLiability()  
       storeGift(), updateGift(), destroyGift()

#### Estate/IHTController.php

- └─ calculateIHT(\$request) → IHTCalculator.calculateIHTLiability()
- └─ calculateSurvivingSpouseIHT() → Second death scenario
- └─ calculateSecondDeathIHTPlanning() → Full married couple analysis
- └─ storeOrUpdateIHTProfile(\$request)

#### Estate/GiftingController.php

- └─ getPlannedGiftingStrategy() → GiftingStrategyOptimizer
- └─ calculateDiscountedGiftDiscount()

#### Estate/LifePolicyController.php

- └─ getLifePolicyStrategy() → LifePolicyStrategyService

#### Estate/TrustController.php

- └─ getTrusts(), createTrust(), updateTrust(), deleteTrust()
- └─ analyzeTrust()
- └─ getTrustAssets() → TrustAssetAggregatorService
- └─ calculateTrustIHTImpact()
- └─ getTrustRecommendations()

#### Estate/WillController.php

- └─ getWill(), storeOrUpdateWill()
- └─ getBequests(), storeBequest(), updateBequest(), deleteBequest()
- └─ getUpcomingTaxReturns()

#### NetWorthController.php

- └─ getOverview() → NetWorthService
- └─ getBreakdown() → assets/liabilities breakdown
- └─ getTrend() → historical snapshots
- └─ getAssetsSummary()
- └─ getJointAssets()
- └─ refresh()

#### PropertyController.php

- └─ CRUD for properties
- └─ calculateSDLT(), calculateCGT(), calculateRentalIncomeTax()
- └─ Nested: mortgages for each property

#### MortgageController.php

- └─ CRUD for mortgages
- └─ calculatePayment()
- └─ amortizationSchedule()

#### HolisticPlanningController.php

- └─ analyze() → CoordinatingAgent
- └─ plan() → Full household plan
- └─ recommendations() → Cross-module recommendations
- └─ cashFlowAnalysis()

```
└─ Recommendation tracking (mark done, in progress, dismiss)

RecommendationsController.php
├─ index() → All recommendations
├─ summary() → Recommendation counts
├─ top() → Top 5 recommendations
├─ completed() → Completed recommendations
└─ Tracking: markDone(), markInProgress(), dismiss(), updateNotes()
```

### Configuration Controllers (2 files)

```
TaxSettingsController.php (admin-only)
├─ getCurrent()
├─ getAll()
├─ create()
├─ update()
└─ setActive($id) → Updates active tax year config

UKTaxesController.php
└─ index() → Returns all tax rules
```

### Services - Detailed Location Map

#### app/Services/Estate/ (20 files - most complex)

```
IHTCalculator.php (400+ lines)
├─ calculateIHTLiability(Collection, IHTProfile, ...)
│   ├── Gross estate value calculation
│   ├── Spouse exemption logic
│   ├── Gift relief analysis
│   ├── Trust IHT values
│   └─ Final IHT at 40%
├─ calculatePETLiability($gifts)
└─ getTaperReliefPercentage($years)

NetWorthAnalyzer.php
├─ generateSummary($userId) → Full net worth statement
├─ calculateNetWorth($userId) → Assets - Liabilities
├─ getTrend($userId) → Historical trend
├─ identifyConcentrationRisk($assets)
└─ getAssetBreakdown()

CashFlowProjector.php (250+ lines)
├─ createPersonalPL($userId, $taxYear) → P&L statement
├─ createCashFlowStatement($userId, $taxYear) → Cash flow
├─ calculateTotalIncome() → All income sources
├─ calculateTotalExpenses()
└─ calculateDiscretionaryIncome()

GiftingStrategy.php
├─ analyzePETs($gifts)
├─ analyzeCLTs($gifts)
├─ calculateTaperRelief()
└─ identifyGiftingOpportunities()
```

### GiftingStrategyOptimizer.php

- └─ optimizeGiftingTimeline(\$userId)
- └─ calculateOptimalAnnualGift()
- └─ projectEstateTaxWithGifting()

### GiftingTimelineService.php

- └─ trackPETStatus(\$gift)
- └─ checkSurvivalStatus()
- └─ generateTimeline()

### SecondDeathIHTCalculator.php (complex)

- └─ calculateSecondDeathScenario(\$user, \$spouse)
- └─ modelSpouse1Death() → Spouse exemption unlimited
- └─ modelSpouse2Death() → Full NRB available
- └─ projectEstateGrowth()
- └─ recommendedLifeCover()

### LifePolicyStrategyService.php

- └─ analyzeLifePolicyNeeds()
- └─ compareWholeOfLifeVsSelfInsurance()
- └─ recommendCoverAmount()

### FutureValueCalculator.php

- └─ calculateFutureValue(\$principal, \$rate, \$years)
- └─ getLifeExpectancy(\$user)
- └─ projectMortgageBalance()
- └─ calculateAssetProjection()

### ActuarialLifeTableService.php

- └─ getLifeExpectancy(\$age, \$gender)
- └─ getSurvivalProbability(\$years)
- └─ getDeathAge()

### TrustService.php

- └─ calculateTrustValue()
- └─ analyzeIHTImpact()
- └─ trackPeriodicCharges()
- └─ recommendTrustType()

### SpouseNRBTrackerService.php

- └─ trackNRBTransfer()
- └─ getAvailableNRB()
- └─ monitorNRBUsage()

### EstateAssetAggregatorService.php

- └─ aggregateAllAssets(\$userId)
  - └─ Manual estate assets
  - └─ Investment accounts
  - └─ Properties
  - └─ Savings accounts
  - └─ Pension values
- └─ categorizeAssets()

### IHTStrategyGeneratorService.php

- └─ generateMitigationStrategies()
- └─ recommendTrustSetup()
- └─ recommendGifting()
- └─ recommendLifeInsurance()

```
LifeCoverCalculator.php
├── calculateMortgageProtection()
├── calculateIncomeLossProtection()
└── calculateIHTLiabilityProtection()
```

**app/Services/Protection/** (5 files)

```
CoverageGapAnalyzer.php
├── calculateProtectionNeeds($profile)
├── calculateTotalCoverage($policies)
├── calculateCoverageGap($needs, $coverage)
├── identifyGaps()
└── getRiskFactors()

AdequacyScorer.php
├── calculateAdequacyScore($gaps, $needs) → %
├── generateScoreInsights()
└── getRiskLevel() → red/amber/green

RecommendationEngine.php
├── generateRecommendations($gaps, $profile)
├── prioritizeRecommendations()
└── calculateImpactScore()

ScenarioBuilder.php
├── modelDeathScenario($profile, $coverage)
├── modelCriticalIllnessScenario()
├── modelDisabilityScenario()
└── calculateFinancialImpact()
```

**app/Services/Savings/** (5 files)

```
EmergencyFundCalculator.php
├── calculateEmergencyFund($expenses)
├── calculateRunway($fund, $expenses) → months
└── getRecommendedTarget()

ISATracker.php (cross-module)
├── calculateAllowanceUsage($userId, $taxYear)
│   ├── Sum savings ISAs
│   ├── Sum investment ISAs
│   ├── Total used vs £20k
│   └── Warnings if exceeded
└── trackByAccountType()

GoalProgressCalculator.php
├── calculateProgressToGoal($goal)
├── calculateMonthlyRequiredSavings()
└── projectGoalCompletion()

LiquidityAnalyzer.php
├── calculateLiquidityRatio()
├── assessLiquidity()
└── identifyIlliquidAssets()
```

```
RateComparator.php
├─ compareAccountRates()
├─ identifyHighYieldAccounts()
└─ recommendBetterRates()
```

**app/Services/Investment/** (5 files)

```
PortfolioAnalyzer.php
├─ analyzeAssetAllocation()
├─ calculatePerformance()
├─ assessGeographicExposure()
└─ identifyConcentration()

MonteCarloSimulator.php
├─ runSimulation($account, $parameters) → Queue job
├─ generateRandomOutcomes()
└─ calculateProbabilities()

AssetAllocationOptimizer.php
├─ optimizeAllocation()
├─ recommendRebalancing()
└─ calculateOptimalWeights()

TaxEfficiencyCalculator.php
├─ calculateTaxLossHarvesting()
├─ assessISAUtilization()
└─ identifyTaxOptimizationOpportunities()

FeeAnalyzer.php
├─ calculateFeeImpact()
├─ compareFeeLevels()
└─ recommendFeeReductions()
```

**app/Services/Retirement/** (5 files)

```
PensionProjector.php
├─ projectAccumulation() → To retirement
├─ projectDrawdown() → From retirement
├─ calculateRetirementDate()
└─ assessRetirementAdequacy()

AnnualAllowanceChecker.php
├─ checkAnnualAllowance($userId, $taxYear)
├─ checkCarryForward() → 3-year rules
└─ identifyAllowanceBreaches()

ContributionOptimizer.php
├─ optimizeContributions()
├─ recommendContributionStrategy()
└─ calculateMaximumAllowedContribution()

ReadinessScorer.php
├─ calculateReadinessScore() → %
└─ assessIncomeAdequacy()
```

```
└─ generateReadinessInsights()  
  
DecumulationPlanner.php  
└─ planRetirementIncome()  
└─ calculateSafeWithdrawalRate()  
└─ projectLongevity()
```

**app/Services/Property/** (3 files)

```
MortgageService.php  
└─ calculateMonthlyPayment()  
└─ generateAmortizationSchedule()  
└─ projectMortgageBalance()  
└─ analyzeMortgageTerms()  
  
PropertyService.php  
└─ CRUD operations  
└─ calculatePropertyValue()  
└─ assessPropertyTax()  
  
PropertyTaxService.php  
└─ calculateSDLT() → Stamp Duty Land Tax  
└─ calculateCGT() → Capital Gains Tax  
└─ calculateRentalIncomeTax()  
└─ identifyTaxEfficiencies()
```

**app/Services/Coordination/** (5 files)

```
HolisticPlanner.php  
└─ createHolisticPlan()  
└─ coordinateModules()  
└─ identifyModuleSynergies()  
  
RecommendationsAggregatorService.php  
└─ aggregateRecommendations() → From all agents  
└─ prioritizeRecommendations()  
└─ resolveConflicts()  
└─ generateUnifiedPlan()  
  
CashFlowCoordinator.php  
└─ calculateHouseholdCashFlow()  
└─ allocateCashFlow()  
└─ identifySurplus/Deficit()  
  
PriorityRanker.php  
└─ rankRecommendations() → By impact & urgency  
└─ calculatePriorityScore()  
└─ groupByPriority()  
  
ConflictResolver.php  
└─ identifyConflicts()  
└─ resolveConflicts()  
└─ suggestOptimalPath()
```

**Other Shared Services** (6 files)

## Onboarding/

- OnboardingService.php
  - getOnboardingSteps()
  - saveStepProgress()
  - completeOnboarding()
  - restartOnboarding()
- EstateOnboardingFlow.php
  - Estate-specific onboarding steps

## Dashboard/

- DashboardAggregator.php
  - generateMainDashboard()
  - calculateFinancialHealthScore()
  - generateAlerts()
  - getQuickStats()

## UserProfile/

- UserProfileService.php
  - updateProfile()
  - validateData()
  - calculateProfileStatistics()
- PersonalAccountsService.php
  - generatePnLStatement()
  - generateCashflowStatement()
  - generateBalanceSheet()

## NetWorth/

- NetWorthService.php
  - aggregateNetWorth()
  - trackNetWorthTrend()
  - identifyHighValueAssets()

## Trust/

- TrustAssetAggregatorService.php
  - aggregateTrustAssets()
- IHTPeriodicChargeCalculator.php
  - calculatePeriodicCharge()
  - trackChargesTimeline()

## Shared/

- CrossModuleAssetAggregator.php
  - aggregateAllModuleAssets()
  - classifyAssetsByModule()

## UKTaxCalculator.php (SHARED)

- calculateIncomeTax()
  - Personal allowance
  - Basic rate (20%)
  - Higher rate (40%)
  - Additional rate (45%)
- calculateNationalInsurance()
- calculateDividendTax()
- calculateSelfEmploymentTax()

## Models (app/Models/ - 39 files)

### Core User Models

```
User.php (Authenticatable, HasApiTokens, Notifiable)
├─ Properties: name, email, password, DOB, marital_status, spouse_id
├─ Relationships: spouse(), household(), familyMembers(), allPolicies()
└─ Methods: getAge(), getTotalIncome(), getIncomeByType()
```

```
FamilyMember.php
├─ User relationship
├─ Role: spouse, child, dependent
└─ Permission tracking
```

```
Household.php
├─ Many users (joint account)
└─ Aggregate household data
```

```
ExpenditureProfile.php
├─ Monthly/annual expenses
├─ Category breakdown
└─ Used by all modules for planning
```

### Protection Module Models

```
ProtectionProfile.php
├─ user_id, income, expenses, dependents
├─ human_capital calculation
└─ Protection analysis baseline
```

```
LifeInsurancePolicy.php, CriticalIllnessPolicy.php, etc.
├─ user_id, sum_assured, term, premium, type
├─ Policy details (inception date, maturity)
└─ Coverage calculation
```

### Savings Module Models

```
SavingsAccount.php
├─ user_id, institution, balance, type (regular/isa)
├─ interest_rate, account_number
└─ Used for emergency fund & ISA tracking
```

```
SavingsGoal.php
├─ user_id, goal_name, target_amount, deadline
├─ current_amount, priority
└─ Goal progress tracking
```

```
ISAAllowanceTracking.php
├─ user_id, tax_year, used_amount, limit (£20k)
└─ Audit trail of ISA contributions
```

## Investment Module Models

### InvestmentAccount.php

- └─ user\_id, provider, current\_value
- └─ account\_type (S&S ISA, GIA, SIPP, VCT, EIS, etc)
- └─ platform, reference
- └─ Used by Estate module for IHT

### Holding.php

- └─ investment\_account\_id, security\_code, quantity, cost\_per\_unit
- └─ current\_value, percentage\_allocation
- └─ Individual securities within accounts

### InvestmentGoal.php

- └─ user\_id, goal\_name, target\_value, target\_date
- └─ current\_value, required\_growth\_rate
- └─ Goal tracking & projections

### RiskProfile.php

- └─ user\_id, risk\_tolerance, questionnaire\_results
- └─ Asset allocation recommendations

## Retirement Module Models

### DCPension.php

- └─ user\_id, scheme\_name, current\_value
- └─ contribution\_rate, employer\_match
- └─ Annual allowance tracking

### DBPension.php

- └─ user\_id, scheme\_name, pension\_accrual
- └─ pension\_in\_payment, scheme\_administrator
- └─ DB scheme details

### StatePension.php

- └─ user\_id, forecast\_amount, state\_pension\_age
- └─ national\_insurance\_credits
- └─ State pension details

### RetirementProfile.php

- └─ user\_id, target\_retirement\_age, desired\_income
- └─ Retirement planning baseline

## Estate Module Models (Estate/ subdirectory)

### Asset.php

- └─ user\_id, asset\_type (property, pension, investment, etc)
- └─ current\_value, ownership\_type (sole/joint/trust)
- └─ is\_iht\_exempt, exemption\_reason
- └─ IHT estate asset

### Liability.php

- └─ user\_id, liability\_type (mortgage, loan, credit card, etc)

- └─ current\_balance, interest\_rate, monthly\_payment
- └─ Deducted from estate for IHT

#### Gift.php (PETs & CLTs)

- └─ user\_id, gift\_date, recipient, gift\_value
- └─ gift\_type (pet, clt, exempt, small\_gift, annual\_exemption)
- └─ status (within\_7\_years, survived\_7\_years)
- └─ taper\_relief\_applicable

#### Trust.php

- └─ user\_id, trust\_name, trust\_type
- └─ current\_value, creation\_date
- └─ Periodic charge tracking, IHT value
- └─ Trust lifecycle management

#### IHTProfile.php

- └─ user\_id, marital\_status, own\_home, home\_value
- └─ nrb\_transferred\_from\_spouse
- └─ charitable\_giving\_percent
- └─ IHT calculation baseline

#### Will.php

- └─ user\_id, death\_scenario (user\_only, both\_die)
- └─ spouse\_primary\_beneficiary, spouse\_bequest\_percentage
- └─ Probate planning

#### Bequest.php

- └─ will\_id, beneficiary, asset\_type, amount/percentage
- └─ Will distribution

#### NetWorthStatement.php

- └─ user\_id, snapshot\_date
- └─ total\_assets, total\_liabilities, net\_worth
- └─ Historical trend tracking

### Net Worth Module Models

#### Property.php

- └─ user\_id, address, current\_value, ownership\_percentage
- └─ property\_type, is\_main\_residence
- └─ Rental income, property tax fields
- └─ Mortgages relationship

#### Mortgage.php

- └─ property\_id, user\_id
- └─ outstanding\_balance, interest\_rate, term\_months
- └─ mortgage\_type (repayment, interest-only, fixed, variable)
- └─ Payment calculations

#### CashAccount.php

- └─ user\_id, account\_type, balance
- └─ Quick cash holdings

#### PersonalAccount.php

- └─ user\_id, account\_type (income, expense, asset, liability)
- └─ amount, description, category
- └─ P&L, cashflow, balance sheet items

```
BusinessInterest.php
├─ user_id, business_name, ownership_percentage
├─ valuation, valuation_method
└─ Business ownership
```

```
Chattel.php
├─ user_id, item_description, valuation
└─ Personal items (art, jewelry, etc)
```

## Configuration Models

```
TaxConfiguration.php
├─ tax_year (e.g., 2025/26)
├─ is_active boolean
├─ All tax rates & allowances (JSON)
└─ Used to override config/uk_tax_config.php
```

```
RecommendationTracking.php
├─ user_id, recommendation_id, status
├─ created_at, completed_at, dismissed_at
└─ Tracks recommendation lifecycle
```

```
OnboardingProgress.php
├─ user_id, current_step, completed_steps
├─ focus_area, started_at, completed_at
└─ Onboarding wizard progress
```

```
SpousePermission.php
├─ user_id, spouse_id, view_*, edit_* boolean fields
└─ Granular cross-spouse permissions
```

---

## FRONTEND FILE LOCATIONS

### Directory Structure

```
resources/js/
├─ App.vue           # Root component
├─ app.js            # Entry point
├─ bootstrap.js      # Axios setup
├─
└─ components/ (150+ Vue files)
    ├─ Admin/
    ├─ Auth/
    ├─ Common/
    ├─ Dashboard/
    ├─ Estate/
    ├─ Protection/
    ├─ Savings/
    ├─ Investment/
    ├─ Retirement/
    ├─ NetWorth/
    └─ Onboarding/
```

- └─ UserProfile/
- └─ Holistic/
- └─ Trusts/
- └─ Shared/
- └─ Actions/
- └─ UKTaxes/
- └─ Global/ (Navbar, Footer, etc)

- └─ views/ (25 Vue files - page level)

- └─ Dashboard.vue
- └─ Estate/
- └─ Protection/
- └─ Savings/
- └─ Investment/
- └─ Retirement/
- └─ NetWorth/
- └─ Trusts/
- └─ UKTaxes/
- └─ Admin/
- └─ Actions/
- └─ Onboarding/
- └─ Login.vue
- └─ Register.vue
- └─ UserProfile.vue
- └─ Settings.vue
- └─ HolisticPlan.vue
- └─ Version.vue

- └─ store/

- └─ index.js
- └─ modules/ (16 .js files)
  - └─ auth.js
  - └─ user.js
  - └─ userProfile.js
  - └─ dashboard.js
  - └─ netWorth.js
  - └─ onboarding.js
  - └─ protection.js
  - └─ savings.js
  - └─ investment.js
  - └─ retirement.js
  - └─ estate.js
  - └─ holistic.js
  - └─ recommendations.js
  - └─ trusts.js
  - └─ spousePermission.js

- └─ services/ (17 .js files)

- └─ api.js
- └─ authService.js
- └─ estateService.js
- └─ protectionService.js
- └─ savingsService.js
- └─ investmentService.js
- └─ retirementService.js
- └─ propertyService.js
- └─ mortgageService.js
- └─ netWorthService.js
- └─ userProfileService.js

```

├── dashboardService.js
├── adminService.js
├── onboardingService.js
├── holisticService.js
├── spousePermissionService.js
├── taxSettingsService.js
├── layouts/ (1 file)
│   └── AppLayout.vue
├── router/ (routing config)
│   └── index.js
└── utils/ (utility functions)
    └── helpers.js

```

## DEPENDENCY RELATIONSHIPS

### Estate Module Complete Chain

```

IHTPlanning.vue
├── props: ihtProfile, user
├── imports: estateService
├── store: estate module
└── calls estateService.calculateIHT()
    ↓
    estateService.js
    ├── api.post('/estate/calculate-ih')
    └── returns response.data
        ↓
        IHTController::calculateIHT()
        ├── receives $request
        ├── validates: StoreIHTRequest
        ├── calls: IHTCalculator.calculateIHTLiability()
        │   ├── Asset::where('user_id', $userId)->get()
        │   ├── InvestmentAccount::where('user_id', $userId)->get()
        │   ├── Property::where('user_id', $userId)->get()
        │   ├── SavingsAccount::where('user_id', $userId)->get()
        │   ├── Liability::where('user_id', $userId)->get()
        │   ├── Mortgage::where('user_id', $userId)->get()
        │   └── Gift::where('user_id', $userId)->get()
        ├── returns {success: true, data: {...}, projection: {...}}
        └── JSON response
            ↓
            Component receives data
            ├── commits to store: estate/setIHTCalculation
            ├── updates component.ihtData
            └── triggers v-if="ihtData" re-render

```

### Cross-Module Asset Aggregation

```
EstateController.index()
├─ Asset::where('user_id')->get()           ← From estate_assets table
├─ InvestmentAccount::where('user_id')->get() ← From investment_accounts table
├─ Property::where('user_id')->get()         ← From properties table
├─ SavingsAccount::where('user_id')->get()   ← From savings_accounts table
├─ Mortgage::where('user_id')->get()        ← From mortgages table
└─ Returns aggregated to frontend:
    {assets: [...], investment_accounts: [...], properties: [...], ...}
```

When calculating IHT:

```
IHTCalculator.calculateIHTLiability()
├─ Assets from estate_assets table
├─ Investment accounts (ISAs not IHT-exempt, included in gross estate)
├─ Properties (with ownership %)
├─ Savings accounts (ISAs not IHT-exempt, included in gross estate)
└─ Total = IHT calculation base
```

## Recommendation Pipeline

```
Component calls: agent.generateRecommendations(analysisData)
↓
Agent loops through analysis results
├─ IHT analysis → "Set up gifting strategy"
├─ Coverage gaps → "Increase life insurance"
├─ ISA usage → "Maximize ISA contributions"
├─ Pension projection → "Increase pension contributions"
└─ (others from each module)
↓
CoordinatingAgent.aggregateRecommendations()
├─ Collects from all agents
├─ PriorityRanker.rankRecommendations()
│   └─ Scores by: impact × urgency
│       └─ Orders by priority
├─ ConflictResolver.resolveConflicts()
│   └─ Identifies: "Pay off debt" vs "Max ISA"
│       └─ Suggests optimal path
└─ Returns: [{id, priority, module, action, rationale, impact}]
↓
Frontend receives array
├─ Vuex store: recommendations/setUnifiedRecommendations
├─ Component: v-for recommendation in recommendations
└─ Display with action buttons (Done, In Progress, Dismiss)
```

## ISA Allowance Tracking (Cross-Module)

```
SavingsController.isaAllowance($taxYear)
├─ calls ISATracker.calculateAllowanceUsage($userId, $taxYear)
└─ Queries:
    └─ SavingsAccount::where([
        'user_id' => $userId,
        'account_type' => 'isa'
```

```
| | | )->sum('current_balance') ← Cash ISA
| | | | InvestmentAccount::where([
| | | | | 'user_id' => $userId,
| | | | | 'account_type' => 's&s_isa'
| | | | ])->sum('current_value') ← Stocks & Shares ISA
| | |
| | | Calculation:
| | | | used = cashISA + stocksISA
| | | | remaining = 20000 - used
| | | | warning = (remaining < 1000)
| | |
| | | Returns {used, remaining, limit: 20000, warning}
| | | ↓
| | | Frontend (ISAAallowanceTracker.vue)
| | | | Shows: "You've used £15,000 of £20,000"
| | | | Displays: progress bar
| | | | Warnings: if within 1% of limit
```

Agent → Service → Model Chain (Protection Example)

```
ProtectionController.analyze()
| ProtectionAgent.analyze($userId)
| | calls User.with(['protectionProfile', 'lifeInsurancePolicies', ...])-
| | >findOrFail()
| | | ProtectionProfile.load()
| | | CoverageGapAnalyzer.calculateProtectionNeeds($profile)
| | | | Models: ExpenditureProfile, ProtectionProfile
| | | CoverageGapAnalyzer.calculateTotalCoverage($policies)
| | | | Models: LifeInsurancePolicy, CriticalIllnessPolicy, etc
| | | CoverageGapAnalyzer.calculateCoverageGap()
| | | AdequacyScorer.calculateAdequacyScore()
| | | RecommendationEngine.generateRecommendations()
| |
| Returns array with all analysis
| | needs, coverage, gaps
| | adequacy_score (%), score_insights
| | recommendations []
| | scenarios (death, CI, disability)
| | ↓
| | Frontend
| | | store.commit('protection/setAnalysis', data)
| | | GapAnalysis.vue → shows gaps
| | | CoverageAdequacyGauge.vue → shows %
| | | Recommendations.vue → shows actions
```

KEY FILE RESPONSIBILITIES MATRIX

Task	Frontend File	Backend Service	Backend Model	Database Table
Calculate IHT	IHTPlanning.vue	IHTCalculator	Asset, Liability, Gift, Trust	assets, liabilities, gifts, trusts

Task	Frontend File	Backend Service	Backend Model	Database Table
Track ISA	ISAAllowanceTracker.vue	ISATracker	SavingsAccount, InvestmentAccount	savings_accounts, investment_accounts
Emergency Fund	EmergencyFund.vue	EmergencyFundCalculator	SavingsAccount, ExpenditureProfile	savings_accounts, expenditure_profiles
Coverage Gaps	GapAnalysis.vue	CoverageGapAnalyzer	ProtectionProfile, *Policy	protection_profiles, *_policies
Portfolio Analysis	PortfolioOverview.vue	PortfolioAnalyzer	InvestmentAccount, Holding	investment_accounts, holdings
Pension Projection	Projections.vue	PensionProjector	DCPension, DBPension, StatePension	dc_pensions, db_pensions, state_pensions
Net Worth	NetWorthOverview.vue	NetWorthService	All models	Multiple tables
Property Tax	PropertyTaxCalculator.vue	PropertyTaxService	Property, Mortgage	properties, mortgages
Gifting Strategy	GiftingStrategy.vue	GiftingStrategyOptimizer	Gift, Trust, Asset	gifts, trusts, assets
Holistic Plan	ExecutiveSummary.vue	CoordinatingAgent	All	All

## CRITICAL FILE PATHS FOR COMMON TASKS

### Add a new policy type to Protection:

1. Create migration: `database/migrations/YYYY_MM_DD_create_[policy]_policies_table.php`
2. Create model: `app/Models/[PolicyType]Policy.php`
3. Add to User model relationships
4. Create Form Request: `app/Http/Requests/Protection/Store[PolicyType]PolicyRequest.php`
5. Update ProtectionController: Add `store[PolicyType]Policy()` method
6. Update routes: `routes/api.php` → Protection prefix
7. Create Vue component: `resources/js/components/Protection/[PolicyType]Form.vue`
8. Update store: `resources/js/store/modules/protection.js`
9. Update service: `resources/js/services/protectionService.js`
10. Update dashboard: `resources/js/views/Protection/ProtectionDashboard.vue`

### Add a new data field to Estate calculation:

1. Create migration: Add column to existing table
2. Update model: Add to `$fillable` and `$casts`
3. Update form request with validation
4. Update IHTCalculator: Incorporate new field in calculation
5. Update API response/database query
6. Update Vue component form
7. Update store to include new field
8. Update API service method signature if needed