

Personalized Wealth Management & Goal Tracker

Project Statement:

This is a digital wealth management platform for planning goals (retirement, home, education), building portfolios, and tracking progress with market-linked updates and simulations.

Key Features:

- Goal-based planning & what-if simulations
- Portfolio builder (stocks/ETFs/mutual funds)
- Financial calculators (SIP, retirement, loan payoff)
- Live dashboards with market sync
- Personalized allocation recommendations

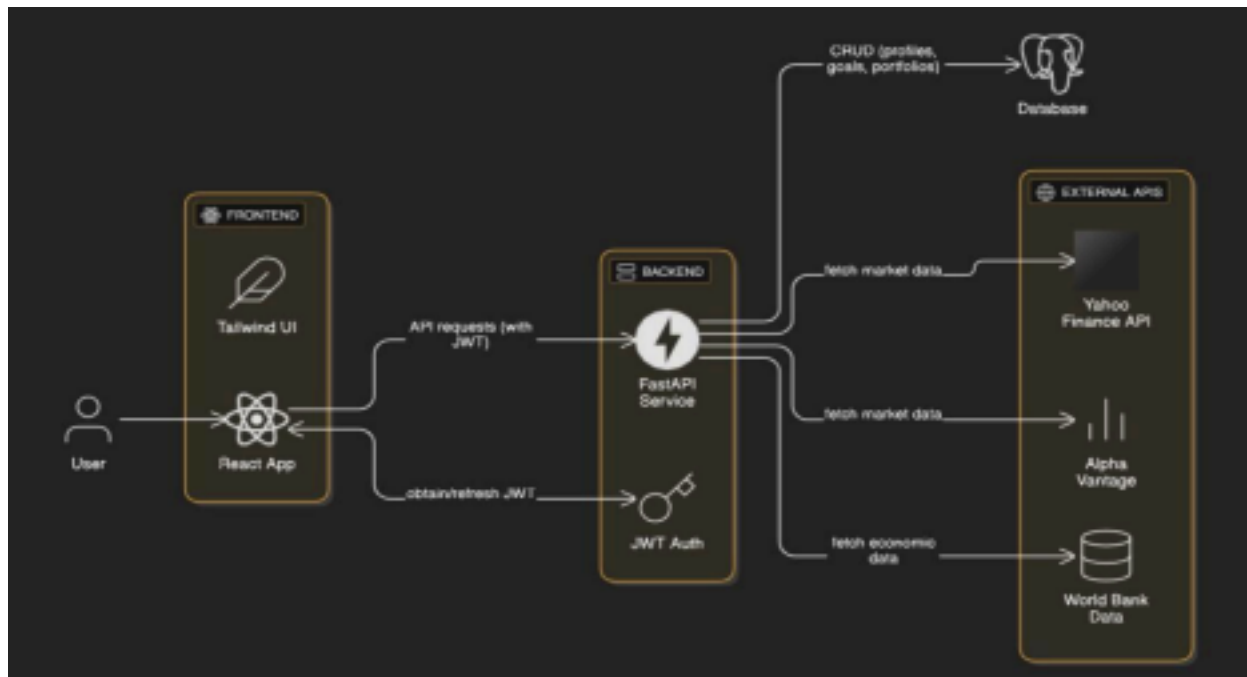
Tech Stack:

- Frontend: React.js + Tailwind CSS
- Backend: FastAPI
- Database: PostgreSQL
- Authentication: JWT (access + refresh)
- **Extras:** Celery/RQ + Redis for scheduled updates; Integrations (Yahoo Finance, Alpha Vantage)

Modules:

- Module A: User Management & Risk Profiling
- Module B: Goals & Simulations
- Module C: Portfolio & Transactions
- Module D: Market Data & Recommendations
- Module E: Dashboards & Reports

Architecture Diagram:



8-Week Milestone Plan

Milestone 1: Weeks 1–2 – Auth, Profile & Foundations

Week 1:

- Set up React + FastAPI project skeleton
- JWT auth (register/login/refresh)
- Create Users table & risk profile fields

Week 2:

- Profile page (risk profile, KYC status)
- Secure routing (protected routes)
- Base Tailwind layout & nav

Expected Output:

Secure auth flow, profile capture, app shell with protected routes.

Milestone 2: Weeks 3–4 – Goals & Portfolio Core

Week 3:

- Goals CRUD (target, date, monthly contribution)
- Goal progress visualization (basic)

Week 4:

- Investments + Transactions CRUD

- Portfolio view (positions, cost basis)

Expected Output:

Goals & portfolio working end-to-end with DB persistence.

Milestone 3: Weeks 5–6 – Market Sync & Simulations**Week 5:**

- Integrate market price fetch (Alpha Vantage/Yahoo)
- Nightly price refresh with Celery

Week 6:

- Simulations module (assumptions → results JSON)
- What-if scenarios on goal timelines

Expected Output:

Live valuations + basic simulation engine.

Milestone 4: Weeks 7–8 – Recommendations & Reports**Week 7:**

- Recommendations engine (suggested allocation JSON)
- Rebalance suggestions per risk profile

Week 8:

- Reports (PDF/CSV export)
- QA, accessibility pass, performance tuning, deployment

Expected Output:

Personalized suggestions, exportable reports, production-ready build.

Expected Project Outcome:

By Week 8, users can set goals, track portfolios with live pricing, run simulations, and receive tailored allocation recommendations with exports.

Database Schema:

- **Users:** id (INT, PK), name (VARCHAR), email (VARCHAR, UNIQUE), password (VARCHAR), risk_profile (ENUM: 'conservative', 'moderate', 'aggressive'), kyc_status (ENUM: 'unverified', 'verified'), created_at

(TIMESTAMP)

- **Goals:** id (INT, PK), user_id (FK to Users.id), goal_type (ENUM: 'retirement', 'home', 'education', 'custom'), target_amount (NUMERIC), target_date (DATE), monthly_contribution (NUMERIC), status (ENUM: 'active', 'paused', 'completed'), created_at (TIMESTAMP)
- **Investments:** id (INT, PK), user_id (FK to Users.id), asset_type (ENUM: 'stock', 'etf', 'mutual_fund', 'bond', 'cash'), symbol (VARCHAR), units (NUMERIC), avg_buy_price (NUMERIC), cost_basis (NUMERIC), current_value (NUMERIC), last_price (NUMERIC), last_price_at (TIMESTAMP)
- **Transactions:** id (INT, PK), user_id (FK to Users.id), symbol (VARCHAR), type (ENUM: 'buy', 'sell', 'dividend', 'contribution', 'withdrawal'), quantity (NUMERIC), price (NUMERIC), fees (NUMERIC), executed_at (TIMESTAMP)
- **Recommendations:** id (INT, PK), user_id (FK to Users.id), title (VARCHAR), recommendation_text (TEXT), suggested_allocation (JSONB), created_at (TIMESTAMP)
- **Simulations:** id (INT, PK), user_id (FK to Users.id), goal_id (INT, NULL, FK to Goals.id), scenario_name (VARCHAR), assumptions (JSONB), results (JSONB), created_at (TIMESTAMP)

