High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **FinSight - Real-Time Financial Portfolio Tracker**. FinSight is a secure, full-stack platform enabling users to track investments, analyze portfolio performance, and receive real-time updates, with advanced data visualization and integration with third-party market data APIs.

---

1. System Architecture Overview

Architecture Description:
FinSight follows a modular, service-oriented architecture with a React-based frontend, FastAPI backend, and PostgreSQL database. Secure authentication, role-based access, and real-time data updates are core features. The system integrates with external APIs for market data and supports concurrent financial operations.

Main System Components:

Component	Role/Responsibility
Web Frontend (React)	User interface, data visualization, user input, session handling
API Gateway (FastAPI)	REST API, business logic, authentication, role management
Database (PostgreSQL)	Persistent storage for users, portfolios, transactions
Market Data Integrator	Fetches real-time data from third-party APIs
Auth Service (JWT)	Secure authentication, token issuance, session validation

---

2. Component Interactions

Sequence Step	Interaction Description
1	User interacts with the React frontend (UI actions, data requests)
2	Frontend sends API requests to FastAPI backend (via HTTPS, JWT-secured)
3	FastAPI backend authenticates user, checks roles, processes business logic
4	Backend queries/updates PostgreSQL for user, portfolio, and transaction data
5	For market data, backend invokes Market Data Integrator to fetch from external APIs
6	Backend returns processed data to frontend for visualization and user feedback

---

3. Data Flow Overview

Data Flow	Source	Destination	Purpose
User credentials/auth	Frontend	Auth Service	Login, JWT issuance
Portfolio/transaction requests	Frontend	Backend API	Portfolio management, transaction ops
Market data fetch	Backend	Third-party APIs	Real-time price and market info
Portfolio/transaction storage	Backend	PostgreSQL	Persistent data storage
Visualization data	Backend	Frontend	Data for charts, tables, analytics

---

4. Technology Stack

Layer	Technology/Frameworks
Frontend	React, TypeScript, Tailwind CSS
Backend	FastAPI (Python), JWT (Auth)
Database	PostgreSQL
Integration	REST APIs (third-party market data)
Security	HTTPS, JWT-based authentication

---

5. Scalability, Reliability & Security

- **Scalability:** Stateless backend enables horizontal scaling; database supports connection pooling; API rate limiting for external integrations.
- **Reliability:** Transaction history and concurrency control ensure data integrity; error handling and monitoring for critical operations.
- **Security:** JWT-based authentication, role-based access control, encrypted data in transit (HTTPS), and secure storage of sensitive information.

End of Document