High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components of **FinSight - Interactive Financial Analytics Portal**. FinSight is a frontend analytics platform for finance professionals, providing dynamic data visualization, advanced filtering, and real-time updates. The platform leverages React, Redux Toolkit, and Tailwind CSS to deliver a responsive, modular user experience, with secure authentication and integration with external financial data APIs.

System Architecture Overview

Architecture Summary:
FinSight is a single-page application (SPA) structured around modular React components, centralized state management, and secure API integration.

Module/Component	Role/Responsibility
UI Layer (React + Tailwind)	Renders responsive, interactive user interface and visualizations
State Management (Redux Toolkit)	Manages global app state, including user, data, and UI state
API Integration Layer	Handles communication with external financial data APIs
Authentication Module	Manages user login, registration, and session handling
Forms & Filters Module	Provides complex forms and advanced filtering capabilities
Real-Time Update Engine	Subscribes to and processes live data updates

---

Component Interactions

Interaction Sequence
1. User authenticates via Authentication Module.
2. Upon login, UI Layer loads dashboard and triggers data fetch via API Integration Layer.
3. API Integration Layer retrieves financial data and dispatches to Redux store.
4. State changes propagate to UI Layer, updating visualizations and forms.
5. User applies filters/forms; Forms & Filters Module updates Redux state and triggers data refresh.
6. Real-Time Update Engine listens for live data, updating Redux state and UI in real time.

Data Flow Overview

Source/Trigger	Data Flow Path

User Actions	UI Layer → Forms/Filters → Redux Toolkit → API Integration (if needed)
API Data Fetch	API Integration → Redux Toolkit → UI Layer (visualizations, tables, etc.)
Real-Time Updates	Real-Time Engine → Redux Toolkit → UI Layer
Authentication Events	Auth Module → Redux Toolkit → UI Layer

Technology Stack

Layer/Function	Technology/Frameworks
UI/Frontend	React, Tailwind CSS, HTML, CSS
State Management	Redux Toolkit, Redux Thunk
Type Safety	TypeScript
API Integration	REST APIs (external financial data)
Authentication	JWT/OAuth (via API)
Data Visualization	Chart.js, D3.js (or similar)
Tooling	Webpack, ESLint, Prettier

Scalability & Reliability

- **Scalability:** Modular React components and Redux state slices enable feature expansion and team scaling. API integration is abstracted for easy extension to new data sources.
- **Reliability:** Centralized error handling in API and state layers; real-time updates use resilient WebSocket or polling strategies.
- **Security:** Authentication flows use secure token-based mechanisms; sensitive data is never stored client-side. Input validation and sanitization are enforced in forms.

End of Document