High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **RetailIQ - Smart Retail Analytics Dashboard**. RetailIQ is a modern, responsive dashboard for retail analytics, providing advanced data visualization, inventory management, and secure user authentication. The solution leverages React, TypeScript, Tailwind CSS, and Redux Toolkit to deliver a scalable and maintainable platform.

1. System Architecture Overview

Architecture Summary:
RetailIQ is a single-page application (SPA) built with a modular frontend architecture. It interacts with backend APIs for data retrieval, authentication, and inventory management. State management is centralized using Redux Toolkit.

Module/Component	Role/Responsibility
UI Layer	Renders dashboard, visualizations, and forms
State Management (Redux)	Centralizes app state, handles async data flows
API Layer	Interfaces with backend for data, auth, inventory
Auth Module	Manages user login, registration, and session
Visualization Module	Renders charts, graphs, and analytics widgets
Inventory Module	Manages inventory data and related operations
Responsive Design Layer	Ensures multi-device compatibility (Tailwind CSS)

2. Component Interactions

Interaction Sequence
1. User accesses dashboard; Auth Module checks authentication state.
2. Upon login, UI Layer requests analytics/inventory data via API Layer.
3. API responses update Redux state; UI Layer subscribes to state changes and updates views.
4. Visualization Module renders data-driven charts/graphs.
5. Inventory actions (add/update/remove) are dispatched to API Layer, updating backend and Redux state.

3. Data Flow Overview

Source	Flow Direction	Destination	Purpose

User Actions	UI Layer	Redux/State	Triggers state updates, API calls
Redux Actions	Redux Middleware	API Layer	Handles async data fetch/update
API Responses	API Layer	Redux Store	Updates state with backend data
Redux Store	State Subscription	UI Layer	Renders updated data/visualizations

4. Technology Stack

Layer/Function	Technology/Framework
UI/Frontend	React, TypeScript, Tailwind CSS
State Management	Redux Toolkit
Data Visualization	Chart.js or Recharts (JS)
API Communication	Fetch API / Axios (JS)
Authentication	JWT-based (via API)
Styling/Responsive	Tailwind CSS
Markup	HTML5, CSS3

5. Scalability & Reliability

- **Scalability:** Modular React components and Redux state management enable easy feature expansion and maintenance. Responsive design ensures usability across devices.
- **Reliability:** Centralized state and API error handling improve robustness. Authentication secures user data and access.
- **Security:** JWT-based authentication, secure API calls, and input validation protect sensitive data.

End of Document