



High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **RetailX - Personalized E-Commerce Experience**. RetailX is a next-generation e-commerce frontend delivering personalized product recommendations, advanced search, and seamless checkout, built with React, Redux Toolkit, and Tailwind CSS.

1. System Architecture Overview

Architecture Description:

RetailX is a modular, single-page application (SPA) structured around a component-based React frontend. State management is centralized via Redux Toolkit. The UI is styled with Tailwind CSS. The frontend communicates with backend APIs for product data, user authentication, recommendations, and payment processing.

Main System Modules

Module	Role/Responsibility
UI Layer	Renders responsive, interactive views using React & Tailwind
State Management	Centralizes app state with Redux Toolkit
Product Engine	Handles product listing, filtering, and recommendations
Search Module	Provides advanced, dynamic product search
Auth Module	Manages user authentication and session state
Checkout Module	Orchestrates cart, payment, and order confirmation
API Integration Layer	Interfaces with backend APIs (products, auth, payments, etc.)

2. Component Interactions

Sequence Step	Interaction Description
1. User Action	User browses, searches, or filters products via UI
2. State Update	Redux updates state; triggers data fetch if needed
3. API Request	API Integration Layer fetches data (products, recommendations, etc.)
4. Data Rendering	UI Layer displays updated products, recommendations, or search results
5. Auth/Checkout	Auth Module manages login; Checkout Module handles payment flow
6. Payment Integration	Checkout Module interacts with payment API for transaction



3. Data Flow Overview

Data Flow	Source Component	Destination Component	Purpose
Product Data Fetch	API Integration	Product Engine, UI Layer	Display product listings & details
User Auth State	Auth Module	State Management, UI	Control access, personalize experience
Recommendations Fetch	API Integration	Product Engine, UI Layer	Show personalized suggestions
Cart/Order Data	Checkout Module	API Integration, UI Layer	Manage cart, process orders
Payment Processing	Checkout Module	Payment API	Complete transactions

4. Technology Stack

Layer/Function	Technology/Framework
UI Framework	React (TypeScript)
State Management	Redux Toolkit
Styling	Tailwind CSS
Language	JavaScript, TypeScript
API Integration	RESTful APIs (external)
Build Tooling	Vite / Webpack
Testing	Jest, React Testing Library

5. Scalability & Reliability

- **Scalability:**
 - Stateless frontend enables horizontal scaling via CDN/static hosting.
 - Modular architecture supports feature expansion and code splitting.
- **Reliability & Security:**
 - Secure API communication (HTTPS, token-based auth).
 - Input validation and error handling at UI and API layers.
 - Payment handled via trusted third-party APIs.
- **Performance:**
 - Optimized asset delivery (code splitting, lazy loading).
 - Responsive design for all device sizes.