



Product Requirements & Specification Document

Project Name

RetailIQ - Smart Retail Analytics Dashboard

Description

RetailIQ is a futuristic, startup-focused analytics dashboard for retail businesses. Built with React, TypeScript, Tailwind CSS, and Redux Toolkit, it provides advanced data visualization, inventory management, and secure user authentication. The dashboard is fully responsive for seamless use across devices.

1. Goals & Objectives

Goal	Description
Advanced Analytics	Deliver actionable insights via interactive data visualizations
Inventory Management	Enable real-time inventory tracking and management
Secure User Authentication	Ensure secure, role-based access
Responsive Design	Provide optimal experience on desktop, tablet, and mobile
Modern, Futuristic UI	Reflect startup and futuristic themes using Tailwind CSS

2. Core Features

Feature	Description
Dashboard Overview	Key metrics (sales, revenue, inventory, trends) in interactive widgets
Data Visualization	Charts (bar, line, pie), heatmaps, and trend analysis
Inventory Management	View, add, edit, and remove inventory items; low-stock alerts
User Authentication	Sign up, login, logout, password reset; JWT-based; role-based access
User Roles	Admin (full access), Manager (analytics, inventory), Staff (inventory only)
Notifications	Real-time alerts for low stock, sales spikes, and system events
Settings/Profile	User profile management, password change, notification preferences

3. Technical Specifications

3.1 Architecture

- Frontend:** React (with TypeScript), Tailwind CSS, Redux Toolkit



- **State Management:** Redux Toolkit
- **Authentication:** JWT (JSON Web Token)
- **API Integration:** RESTful endpoints (mocked or real)
- **Responsiveness:** Tailwind CSS breakpoints for multi-device support

3.2 Technology Stack

Layer	Technology
UI	React, Tailwind
Logic	TypeScript, Redux
Styling	Tailwind CSS
Auth	JWT, Redux
Data	REST API
Markup	HTML, CSS

4. User Stories

As a...	I want to...	So that...
Admin	manage users and inventory	I can control access and stock
Manager	view analytics and manage inventory	I can optimize sales and stock
Staff	update inventory status	I keep stock data accurate
Any User	securely log in and manage my profile	my data is safe and up to date

5. UI/UX Requirements

- **Theme:** Futuristic, clean, startup-inspired
- **Navigation:** Sidebar with icons, collapsible on mobile
- **Charts:** Responsive, interactive (hover, filter, zoom)
- **Accessibility:** WCAG 2.1 AA compliance
- **Branding:** Customizable logo and color palette

6. Non-Functional Requirements

Requirement	Specification
Performance	<2s load time for main dashboard
Security	JWT, HTTPS, input validation, role-based access
Scalability	Modular components, scalable state management
Maintainability	TypeScript types, code linting, clear structure



Responsiveness	Full support for desktop, tablet, and mobile
----------------	--

7. API & Data

7.1 Example API Endpoints

```
GET    /api/metrics/overview
GET    /api/inventory
POST   /api/inventory
PUT    /api/inventory/:id
DELETE /api/inventory/:id
POST   /api/auth/login
POST   /api/auth/register
```

7.2 Data Models (TypeScript)

```
type User = { id: string; name: string; role: 'admin' | 'manager' | 'staff'; email: string; }
type InventoryItem = { id: string; name: string; stock: number; threshold: number; }
type Metric = { date: string; sales: number; revenue: number; }
```

8. Acceptance Criteria

ID	Criteria
AC1	Users can log in, log out, and reset password securely
AC2	Dashboard displays real-time metrics and interactive charts
AC3	Inventory can be managed (CRUD) with low-stock alerts
AC4	Role-based access restricts features appropriately
AC5	UI is responsive and visually aligns with the futuristic startup theme

9. Milestones & Timeline

Milestone	Description	Target Date
UI/UX Design	Wireframes, theme, components	Week 1
Authentication	JWT, role-based access	Week 2
Dashboard & Analytics	Metrics, charts, overview	Week 3
Inventory Management	CRUD, alerts, notifications	Week 4
Testing & QA	Unit, integration, E2E tests	Week 5
Deployment	Production build, documentation	Week 6



10. Risks & Mitigations

Risk	Mitigation
Data security breaches	Use JWT, HTTPS, input validation
Performance bottlenecks	Optimize rendering, lazy load heavy charts
Inconsistent UI/UX	Enforce design system, component library

11. Out of Scope

- Payment processing
- Third-party integrations (beyond REST API)
- Advanced AI/ML analytics

12. Appendix

- **References:** [React](#), [Redux Toolkit](#), [Tailwind CSS](#)
- **Contact:** Product Owner, Lead Developer

End of Document