Project Documentation: Store Manager –
Inventory Tracking System

1. Introduction

- **Project Title**: Store Manager – Inventory Tracking System
- **Team ID**:NM2025TMID46901
- **Team leader**:Thanuja saraswathi.S&thanujasaraswathineet2023@gmail.com

S.NO	NAME	MAIL ID
1	**Leenadevi.S**	**leenadevisugumaran@gmail.com**
2	**Kavya.P**	**kaviyapushparaj7@gmail.com**
3	**Kaviya.V**	**vkaviya666@gmail.com**
4	**Kaviya.C**	**kaviyamurugammal3@gmail.com**

2. Project Overview

- **Purpose**:
  The system helps store managers track and manage inventory effectively. It enables users to add, update, and monitor products, generate reports, and receive alerts when stock levels are low.
- **Features**:
  - Product management (add, update, delete, view).
  - Real-time stock tracking.
  - Automatic stock reduction after sales.
  - Low-stock alerts and reorder level notifications.
  - Sales and inventory reports.
  - User authentication with role-based access.

---

3. Architecture

- **Component Structure**:
  - `Navbar` – navigation across modules.
  - `Dashboard` – summary of inventory and sales.
  - `ProductList` – list of all products with search and filter.
  - `ProductForm` – add or edit product details.
  - `StockManager` – manage stock levels.
  - `SalesManager` – record and track sales.
  - `Reports` – generate and display sales/inventory reports.
  - `Login/Register` – authentication and role management.
- **State Management**:
  - Global state handled using **Redux Toolkit** (or Context API).
  - Local state for component-level updates (e.g., form inputs).
- **Routing**:
  - Implemented using **React Router** for navigation:
    - `/login`, `/dashboard`, `/products`, `/sales`, `/reports`, `/settings`.

---

4. Setup Instructions

- **Prerequisites**:
  - Node.js, npm/yarn, React, Redux Toolkit, MySQL/MongoDB (backend), Git.
- **Installation**:
- `git clone <repository-url>`
- `cd store-manager-frontend`
- `npm install`
- `npm start`

---

5. Folder Structure

```
store-manager-frontend/
├── public/
├── src/
│   ├── components/ (Navbar, ProductForm, SalesForm, etc.)
│   ├── pages/ (Dashboard, Products, Sales, Reports)
│   ├── redux/ (slices, store)
│   ├── utils/ (helper functions, hooks)
│   ├── assets/ (images, icons)
│   └── App.js
├── package.json
```

---

6. Running the Application

- **Frontend**:
- `npm start`
- **Backend** (example if using Node/Express):
- `npm run server`

---

7. Component Documentation

- **Key Components**:
  - `ProductList`: Displays all inventory items.
  - `ProductForm`: Add/edit products.
  - `SalesForm`: Record sales transactions.
  - `ReportViewer`: Displays inventory and sales reports.
- **Reusable Components**:
  - `Button`, `Modal`, `InputField`, `Table`, `Card`.

---

8. State Management

- **Global State**:
  - Products, sales, user authentication stored in Redux.
  - Reducers manage stock changes when sales occur.
- **Local State**:
  - Form input handling (React useState).
  - UI toggles (e.g., modal open/close).

---

9. User Interface

- Dashboard with sales/stock graphs.
- Product management table with search/filter.
- Stock update and alerts on low stock.
- Reports with export option (PDF/Excel).

---

10. Styling

- **CSS Frameworks/Libraries**: Tailwind CSS / Bootstrap.
- **Theming**: Custom dark/light mode theme.

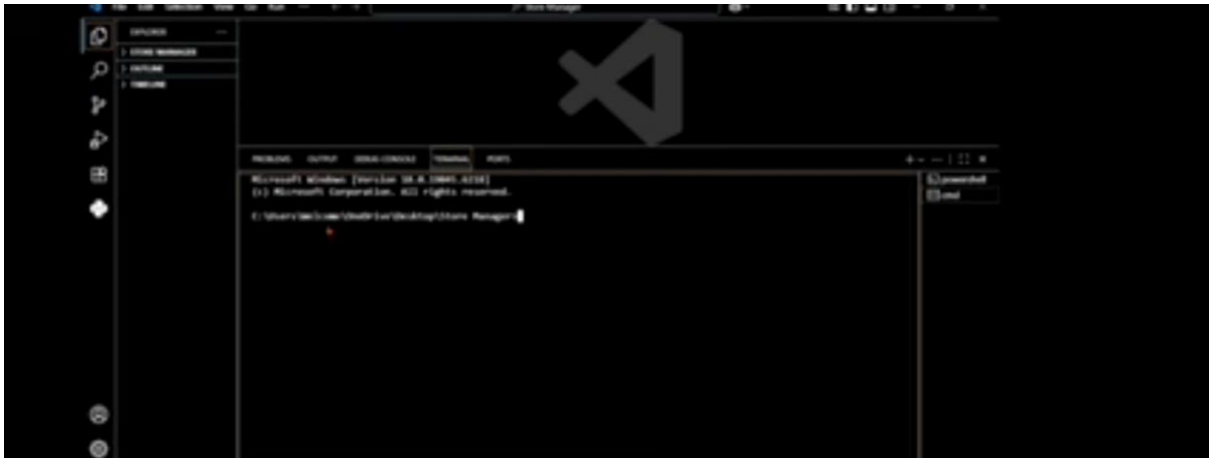
---

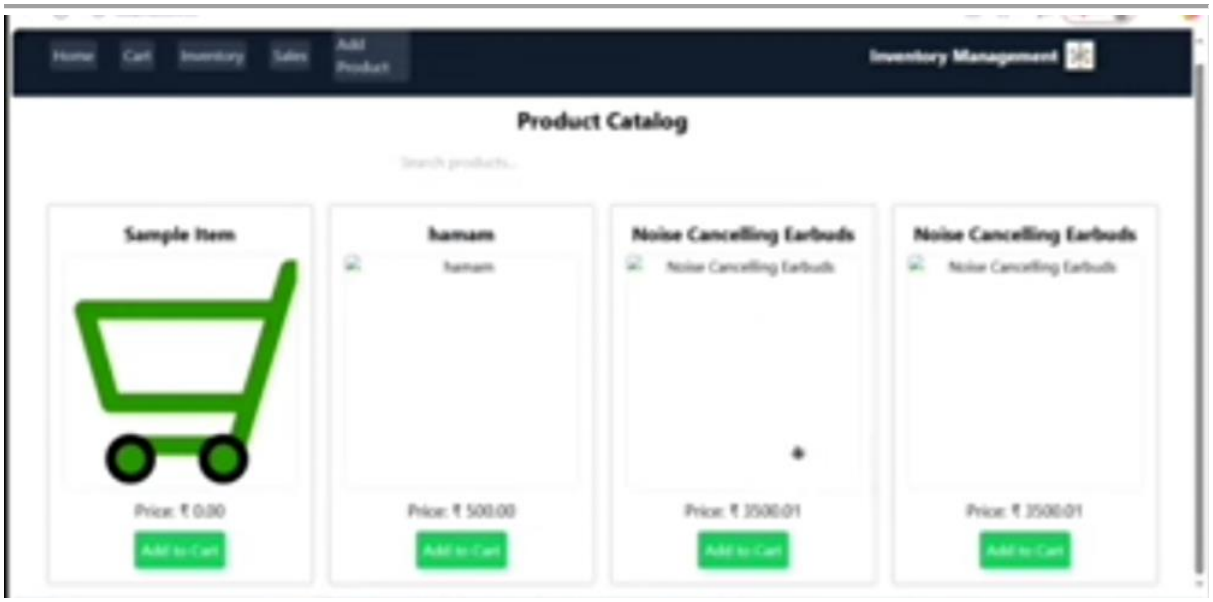
11. Testing

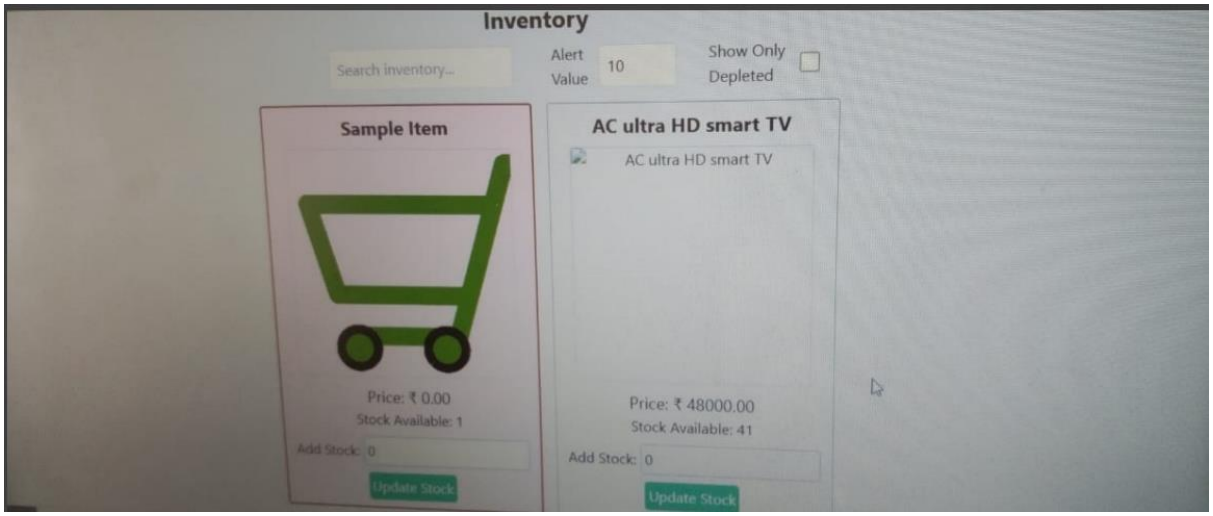
- **Testing Strategy**:
  - Unit testing with **Jest**.
  - Integration testing with **React Testing Library**.
  - End-to-end testing with **Cypress**.
- **Code Coverage**:
  - Jest and Istanbul used to ensure test coverage.

---

12. Screenshots or Demo




Home
Cart
Inventory
Sales
Add Product
Inventory Management
Product Catalog
Search products...
Sample Item
Price: ₹ 0.00
Add to Cart
hamam
hamam
Price: ₹ 500.00
Add to Cart
Noise Cancelling Earbuds
Noise Cancelling Earbuds
Price: ₹ 3500.01
Add to Cart
Noise Cancelling Earbuds
Noise Cancelling Earbuds
Price: ₹ 3500.01
Add to Cart



13. Known Issues

- Delay in real-time sync if backend server is slow.
- Export to Excel may require additional configuration.

---

14. Future Enhancements

- Barcode scanner integration.
- AI-powered demand forecasting.
- Mobile app version.
- Multi-branch inventory management?