nn **Store Manager App Documentation**

n **Introduction**

The Store Manager App is a web-based application designed to help businesses manage their inventory, sales, and staff efficiently. It allows store owners and managers to keep track of products, stock levels, customer transactions, and sales reports in a simple, user-friendly interface.

n **Project Overview**

- Purpose: To simplify store operations by providing a centralized platform for inventory and sales management.
- Users: Store owners, managers, and employees.
- Features:
  Product CRUD operations (Add, Edit, Delete, View)
  Inventory tracking with alerts for low stock
  Sales management (billing, invoices)
  Employee/user role management
  Dashboard with analytics

nn **Architecture**

- Frontend: React (Next.js or CRA) with Tailwind CSS / Material UI
- Backend: Node.js + Express.js (REST API)
- Database: MongoDB (or PostgreSQL / MySQL alternative)
- State Management: Redux Toolkit / Context API
- Authentication: JWT-based authentication & role-based access control
- Deployment: Docker + CI/CD (optional), deployed on Vercel/Netlify (frontend) & Heroku/Render (backend)

nn **Setup Instructions**

1. Clone the repository:

```
git clone https://github.com/yourusername/store-manager-app.git
cd store-manager-app
```

2. Install dependencies:

```
npm install
```

3. Configure environment variables (.env):

```
MONGO_URI=your-mongodb-url
JWT_SECRET=your-secret-key
PORT=5000
```

4. Start backend:

```
cd backend npm
run dev
```

5. Start frontend:

```
cd frontend
npm start
```

n **Folder Structure**

```
store-manager-app/
nnn backend/ # Express.js API
n nnn models/ # Database models
n nnn routes/ # API routes
n nnn controllers/ # Business logic
n nnn middleware/ # Auth & error handling n
nnn server.js # Entry point

nnn frontend/ # React app
n nnn src/
n n nnn components/ # UI components n
n nnn pages/ # Page views
n n nnn redux/ # State management
n n nnn styles/ # Global styles
n n nnn App.js # Root component

nnn .env.example # Sample environment file
nnn README.md
```

nn Running the Application

- Development Mode: Run backend and frontend separately (npm run dev for backend, npm start for frontend).
- Production Build:

```
cd frontend
npm run build
```

n Component Documentation

- ProductCard → Displays product details (name, price, stock).
- ProductForm → Form for adding/editing products.
- Cart → Manages items before checkout.
- Dashboard → Displays analytics & reports.
- Navbar → Navigation with role-based links.

n State Management

- Redux Toolkit (recommended):

productsSlice: Manages product state.

authSlice: Handles user login/logout.

cartSlice: Manages cart items.

n User Interface

- Clean, responsive design with Tailwind CSS.
- Dark/Light theme support (optional).
- Dashboard with charts (using Recharts/Chart.js).

n Styling

- Tailwind CSS for utility-first styling.
- ShadCN/UI or Material UI for prebuilt components.

n Testing

- Unit Tests: Jest + React Testing Library (frontend), Mocha/Chai (backend).
- Integration Tests: Supertest for API endpoints.

- Run tests with:

```
npm test
```

n **Demo Video**

n inventory-management-frontend-demo.mp4 - Google Drive

n **Known Issues**

- No offline support yet.
- Limited reporting features.
- Role management is basic (admin/user only).

n **Future Enhancements**

- n Barcode scanner support for faster checkout.
- n Mobile app version (React Native).
- n Advanced analytics with AI-powered insights.
- n Multi-store management.
- n Integration with payment gateways (Stripe/PayPal).