**STORE MANAGER- Keep Track of inventory**

**FULL STACK DEVELOPMENT**

**NM ID: NM2025TMID37605**

**INTRODUCTION**

**Project Title: Store Manager- keep Track of inventory**

**A Store Manager is responsible for overseeing the daily operations of the store while ensuring efficient inventory management.**

**The role involves keeping accurate records of stock, monitoring inventory levels, and coordinating with suppliers to ensure timely replenishment.**

**By implementing proper tracking systems and analyzing sales patterns, the Store Manager helps prevent stockouts and overstocking, reduces losses, and maintains smooth store operations.**

**Along with supervising staff and ensuring customer satisfaction, effective inventory control allows the Store Manager to contribute directly to the store’s profitability and overall success.**

**Team Leader:K.Vagithunisha &** [**vagithunisha086@gmail.com**](mailto:vagithunisha086@gmail.com)

**Team Members: Rathisree.P & rathisreep113@gmail.com**

**Rakshitha.S & r88818657@gmail.com**

**Ranjani.A & rranjani004@gmail.com**

**PROJECT OVERVIEW**

**Purpose:**

* **The Store Manager is to ensure the efficient management of store operations with a primary focus on accurate inventory control.**
* **By keeping track of stock levels, monitoring product movement, and coordinating timely replenishment, the Store Manager helps maintain operational continuity and customer satisfaction.**
* **Through systematic inventory tracking, analysis of demand patterns, and effective communication with suppliers.**
* **The Store Manager minimizes losses, prevents stock discrepancies, and supports the store in achieving its sales and profitability goals.**

**Features:**

* **Accurate Stock Monitoring – Maintains real-time records of inventory to ensure stock levels are always updated.**
* **Efficient Replenishment – Tracks product movement and coordinates timely reorders to prevent shortages or overstocking.**
* **Data-Driven Decisions – Analyzes sales patterns and demand trends to forecast inventory needs effectively.**
* **Loss Prevention – Implements tracking systems to minimize errors, reduce wastage, and avoid discrepancies.**
* **Supplier Coordination – Facilitates seamless communication with vendors for smooth procurement and replenishment.**
* **Operational Efficiency – Ensures inventory availability supports daily store operations and customer satisfaction.**
* **Reporting & Documentation – Prepares inventory reports to support audits, management reviews, and business planning.**
* **Centralized Dashboard – Provides a complete overview of stock levels, sales, and product movement in real time.**
* **Stock Alerts & Notifications – Sends automated alerts for low stock, overstock, or nearing expiry items.**
* **Inventory Adjustment Controls – Allows manual updates for damaged, returned, or transferred items.**
* **Supplier & Purchase Management – Tracks supplier details, purchase orders, and delivery timelines.**
* **User Access & Permissions – Grants role-based access to staff for secure and controlled inventory handling.**
* **Sales & Inventory Integration – Syncs sales data with stock levels to update inventory automatically.**
* **Analytics & Reports – Generates detailed reports on inventory trends, turnover rates, and stock valuation.**

**ARCHITECTURE:**

**The project follows a MERN( Mongo db, Express, React, Node JS)**

* **Frontend (React): Provides the UI components, pages, and routing.**
* **Backend (Node.js + Express): Handles API requests and business logic.**
* **Database (MongoDB): Stores user details, recipes, and favorites.**

**Component Structure 🧩:**

* **Header, Footer, Navigation bar**
* **Recipe List Component**
* **Favorites Component**
* **Authentication Component (Login/Signup)**

**State Management:**

* **Local state for component interactions**
* **Global state using Context API**

**Routing:**

* **React Router is used for navigation between pages such as Home, Recipes, Favorites, and Profile.**

**Setup Instructions**

**Prerequisites ⚙️:**

**Node.js (>=16)**

* **MongoDB**
* **Npm or yarn package manager**

**Installation Steps:**

1. **Clone the repository: `git clone [repo link]`**
2. **Navigate to backend folder: `cd backend`**
3. **Install dependencies: `npm install`**
4. **Setup `.env` file with MongoDB URI & secrets**
5. **Start backend server: `npm run dev`**
6. **Navigate to frontend folder and run: `npm start`**

**FOLDER STRUCTURE**

**📂 Client (Frontend):**

* **Components/**
* **Pages/**
* **Assets/**
* **Utils/**

**📂 Backend:**

* **Server.js**
* **Models/**
* **Routes/**
* **Controllers/**
* **Config/**

**Running the Application**

**Frontend: `npm start` (runs on** [**http://localhost:3000**](http://localhost:3000)**)**

**Backend: `npm run dev` (runs on** [**http://localhost:5000**](http://localhost:5000)**)**

**Ensure MongoDB is running before starting the servers.**

**COMPONENT DOCUMENTATION**

**Key Components:**

* **Product list: Displays available products**
* **Store products : Manages user-saved products**
* **: Handles login/signup**

**Reusable Components:**

* **Buttons**
* **Cards**
* **Modals**

**STATE MANAGEMENT**

**Global State 🌍:**

* **Managed with Context API for auth and favorites**

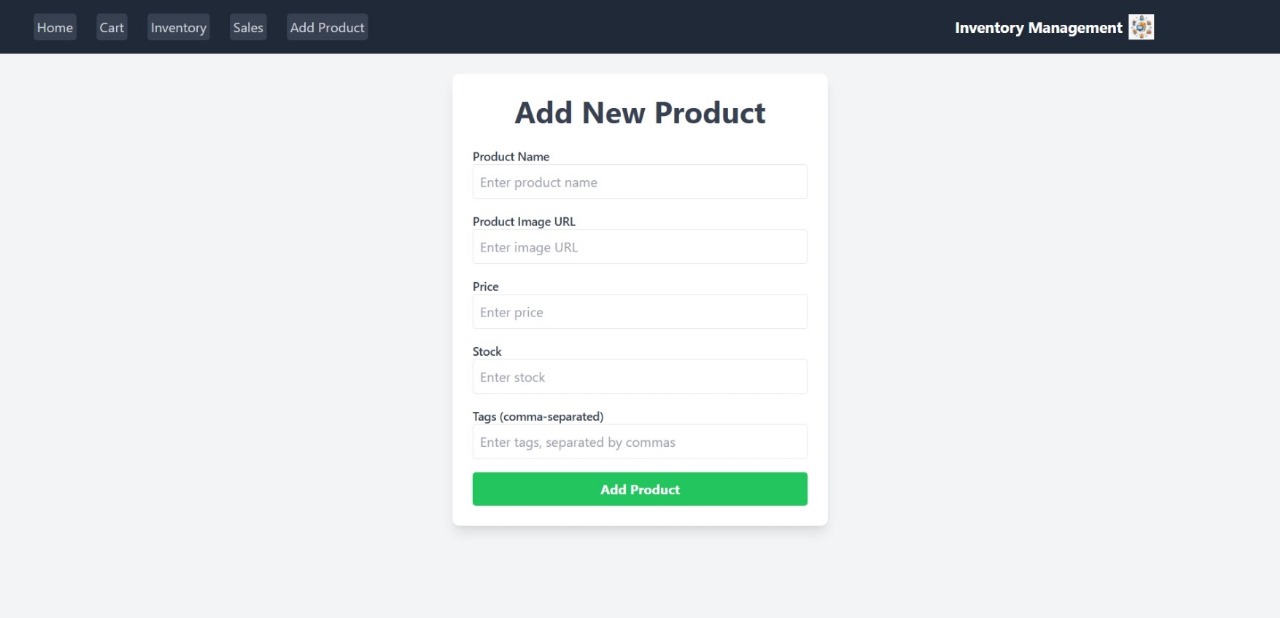
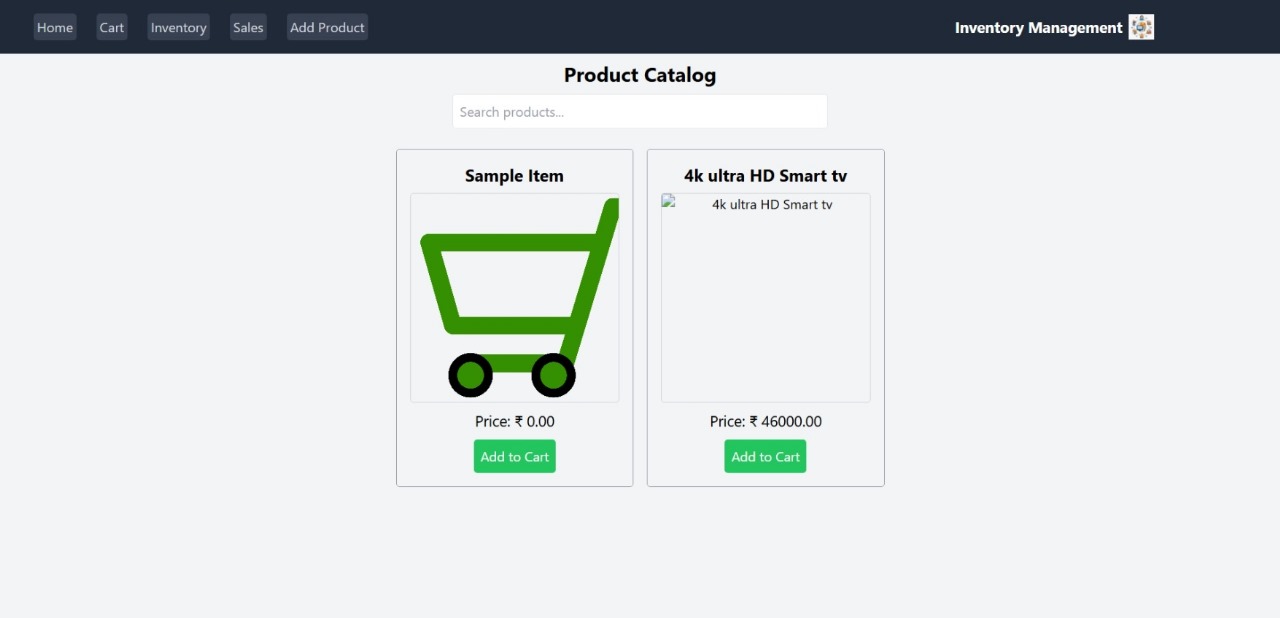
**Local State ⚡:**

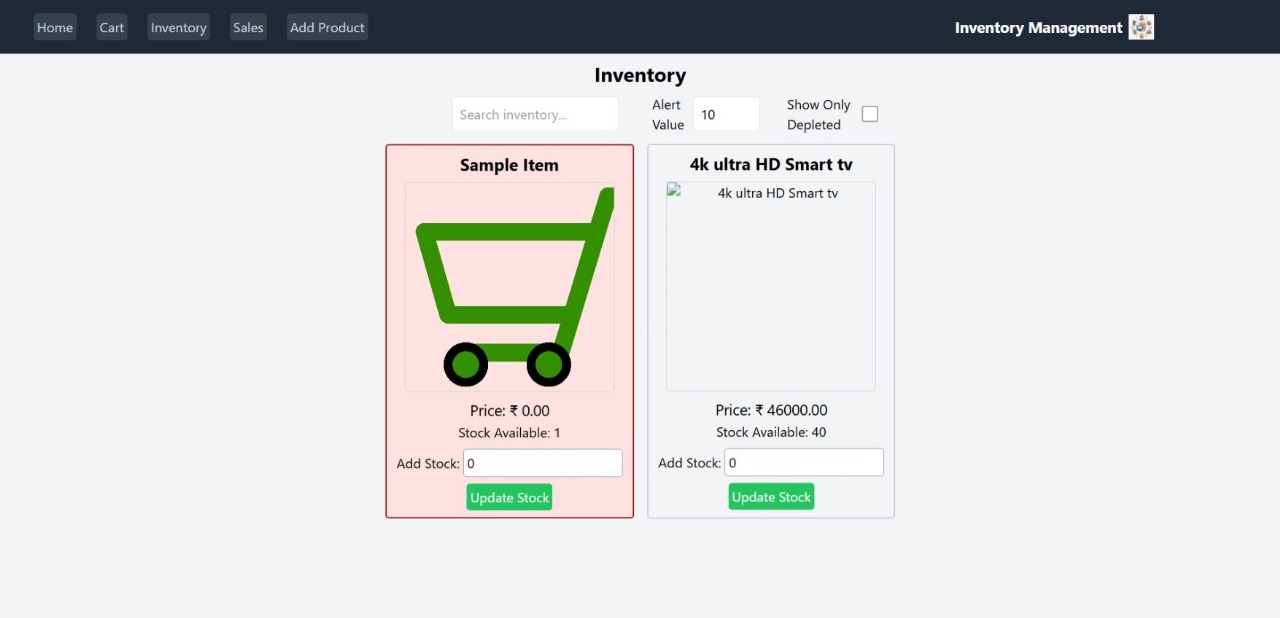
* **Handled via useState in individual components**

**USER INTERFACE**

**UI Highlights 🎨:**

* **Clean, minimal design**
* **Responsive layout for mobile & desktop**
* **Intuitive navigation bar**
* **Products images with and titles**
* **Favorites page for saved recipes**

****

****

**STYLING**

**Styling with CSS and Tailwind 💅**

* **Responsive grid system**
* **Reusable styles for buttons and forms**
* **Theming support for light/dark mode (optional)**

**TESTING**

**Testing Strategy ✅:**

* **Unit testing with Jest**
* **Component testing with React Testing Library**
* **API testing with Postman**

**Code Coverage:**

* **Ensured through automated test scripts**

**SCREENSHOTS AND DEMO**

**Add screenshots or demo link here 📸**

**(E.g.), Home page, Recipe list, Favorites page, Login page**

**Known Issues**

* **Limited dataset of recipes**
* **No role-based access (admin features missing)**
* **Some UI components need optimization for large screens**

**FUTURE ENHANCEMENTS**

**🚀 Planned Improvements:**

* **Add search & filter functionality**
* **Implement recipe categories**
* **Enable recipe image upload**
* **Add admin dashboard for managing recipes**
* **Integrate external recipe APIs for more data**