**Project** **Documentation**

**Project Title: Store Manager – Keep Track of Inventory**

**Team ID: NM2025TMID36474**

**Team Leader:** SARAL ESTHER RANI J–24cssaralestherranij088@gmail. com

**Team Members:**

1. NETHRA R–24csnethrar062@gmail.com
2. SUMITHRA R–24cssumithra101@gmail.com
3. PREMALATHA B –24cspremalatha70@gmail.com

**1.Introduction:**

A Store Manager inventory management system is designed to help businesses efficiently monitor, control, and optimize the flow of goods across all supply chain touchpoints. It ensures accurate record-keeping for stock levels, product tracking, and replenishment needs, which minimizes losses, reduces waste, and supports seamless business operations .

**2. Project Overview :**

* + The system allows for unlimited creation of stocks, customers, orders, and invoices to streamline store management.
  + It manages stock and orders using barcode scanners, enabling easy product search and transaction recording.
  + Stock quantities are automatically updated; the system sends reminders for low stock items to prevent shortages.
  + It supports customer management by tracking payment statuses and order histories.
  + Order management is simplified by allowing tax, discounts, and quantity controls, ensuring that orders do not exceed available stock.
  + Data security features like pin lock and multiple currency support enhance usability.
  + Additional capabilities include importing/exporting data, generating sales reports, and customizable product visibility settings.

**Features**:

* Real-time Stock Monitoring
* Restock Alerts and Reorder Points
* Order and Sales Management
* Barcode Scanning
* Multiple Location Management
* Customer and Supplier Management
* Automated Reordering
* Data Reporting and Analytics
* Offline Functionality
* Security and Multi-currency Support
* Inventory Auditing and Verification
* Demand Forecasting

**3.Architecture :**

* + **Frontend**: React.js with Bootstrap / Material UI
  + **Backend**: Node.js & Express.js handling server logic and API endpoints
  + **Database**: MongoDB storing user profiles, news articles, preferences, and analytics

**4.Setup Instructions:**

Prerequisites:

* Node.js
* MongoDB
* Git
* React.js
* Express.js
* Mongoose
* Visual Studio Code

*Installation Steps:*

* Clone the repository: git clone
* Install client dependencies: cd client && npm install
* Install server dependencies: cd ../server && npm install

**5.Folder Structure :**

Store-Manager/

├── client/ # React frontend

│ ├── components/

│ └── pages/

├── server/ # Node.js backend

│ ├── routes/

│ ├── models/

│ └── controllers/

**6.Running the Application:**

* **Frontend**: cd client && npm start
* **Backend**: cd server && npm start
* **Access**: Visit http://localhost:3000

**7.API Documentation:**

*User Management:*

* POST /api/user/register
* POST /api/user/login

Inventory:

* POST /api/inventory/add
* GET /api/inventory/:id
* PUT /api/inventory/update/:id
* DELETE /api/inventory/delete/:id

Reports:

* GET /api/reports/sales
* GET /api/reports/purchases

**8.Authentication**:

* JWT-based authentication for secure login.
* Middleware to protect private routes.

**9.User Interface :**

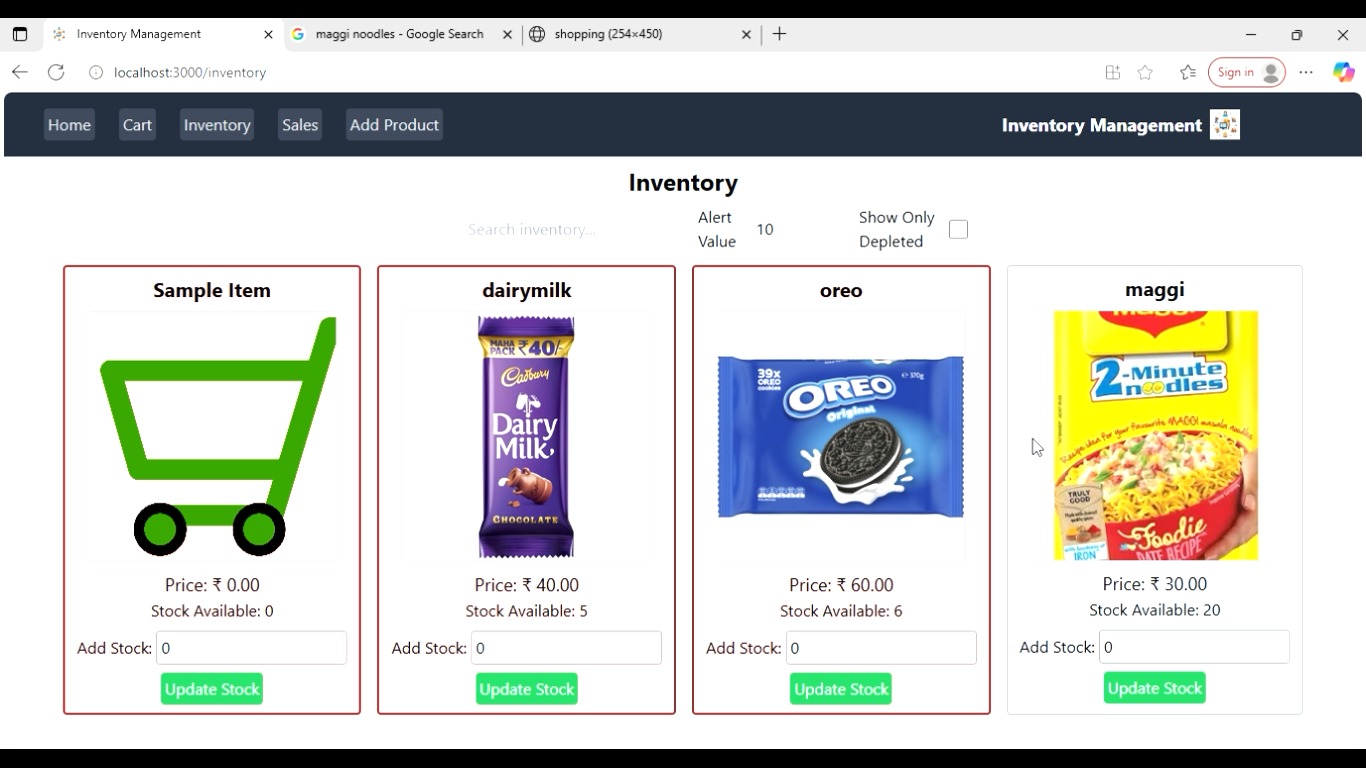
* Dashboard (Inventory overview)
* Add/Edit Inventory Page
* Reports Page (Sales, Purchases, Stock)
* Admin Panel (User management)

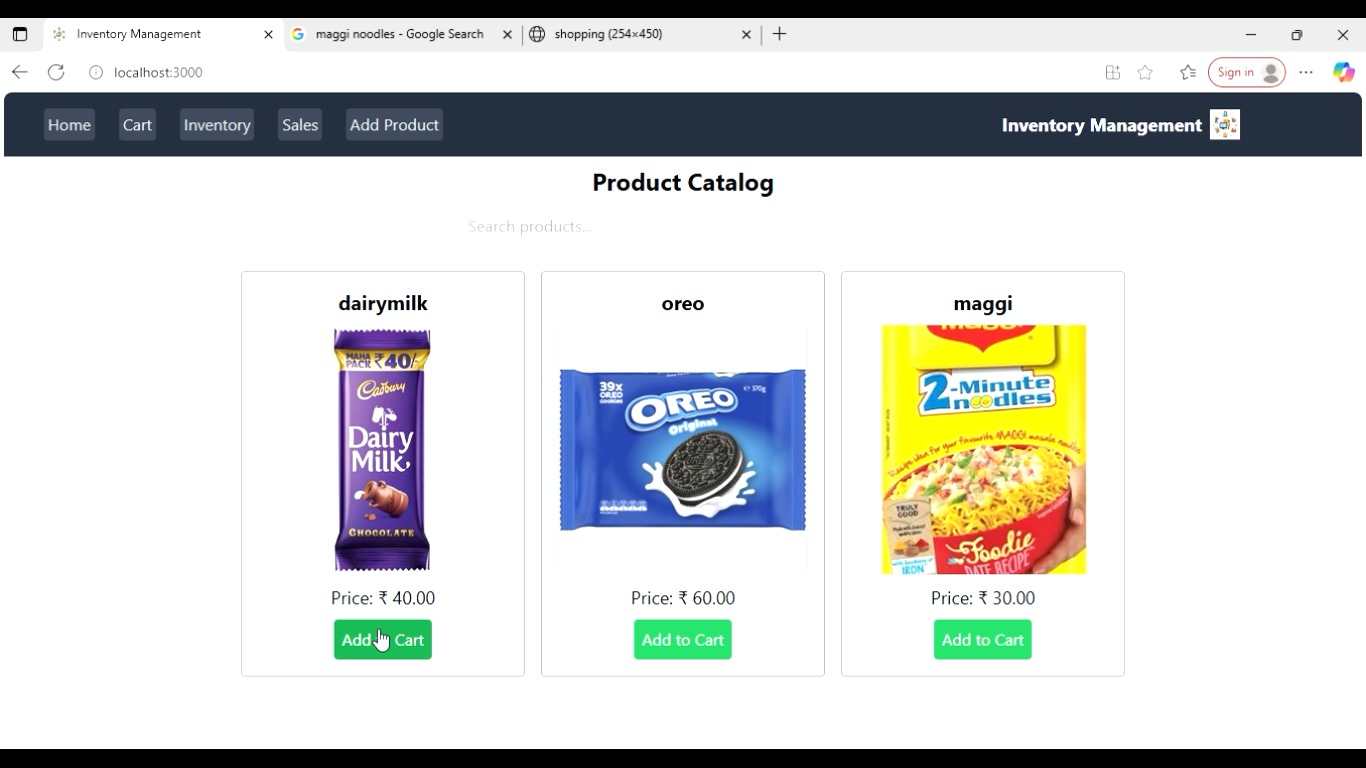
**10.Testing:**

* Manual testing at each milestone.
* Tools: Postman, Chrome Dev Tools

**11.screenshot :**

**demo video :https://drive.google.com/file/d/1Ca-O8EDApIdLUhkeQ7QmsyFbgV0I8mTO/view?usp=drivesdk**

* 

**12.Issues:**

* Poor Demand Forecasting
* Overstocking
* Understocking
* Lack of Real-time Inventory Visibility
* Inefficient Tracking and Ordering Systems
* Inconsistent Data Across Channels
* Warehouse Inefficiencies
* Inventory Defects and Waste
* Returned Inventory Issues
* Poor Supplier Coordination

**13.Future Enhancements:**

* Artificial Intelligence (AI) and Machine Learning
* Automation and Robotics
* Cloud-Based Inventory Systems
* Advanced Data Analytics
* Sustainability Focus
* Predictive Maintenance and Smart Warehousing

**14.Conclusion :**

The “Store Manager – Keep Track of Inventory” system is essential for efficient retail operations, providing real-time stock tracking, automated alerts, and order management. It helps reduce errors, prevent stockouts, and improve customer satisfaction. By optimizing inventory levels, the system minimizes costs and maximizes sales. Future advancements like AI and IoT will enhance forecasting and automation. Overall, this system supports better decision-making, operational efficiency, and business growth in competitive markets