

Project Documentation

Store Manager: Keep Track of Inventory

1. Introduction

- **Project Title:** Store Manager: Keep Track of Inventory
- **Team ID:** NM2025TMID38607
- **Team Leader:** DIVYA BHARATHI P & Mail ID: divya84008@gmail.com
- **Team Members:**
 - NADHIYA K & Mail ID: nadhiyak981@gmail.com
 - MADHUMITHA B & Mail ID: Madhubabu132007@gmail.com
 - MONISHA S & Mail ID: monishasuresh0129@gmail.com

2. Project Overview

- **Purpose:** Store Manager is designed to help businesses monitor and control their inventory efficiently, ensuring optimal stock levels, minimizing losses, and maximizing profit. It streamlines tasks such as stock entry, adjustments, sales tracking, and low stock notifications.
- **Features:**
 - Product registration and categorization
 - Real-time inventory tracking and updates
 - Sales and purchase management
 - Automated low-stock alerts via notification/email
 - Stock adjustment logs
 - Barcode and QR code support for products
 - User role management (Admin, Staff)
 - Comprehensive analytics and inventory reporting

3. Architecture

- **Frontend:** React.js implemented with Bootstrap and Material UI for responsive user interfaces
- **Backend:** Node.js and Express.js handle API requests, server-side business logic, and authentication
- **Database:** MongoDB maintains product details, inventory records, user roles, and transaction logs

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. State Management

- **Global State:** Implemented using React Context API to manage inventory data across components.
- **Local State:** Managed using useState for component-specific interactions such as form input, search query, and modal visibility.

8. Component Documentation

- The project uses modular components to manage and display inventory efficiently. Each component handles a specific task to keep the code clean and reusable.

Key Components:

- **ProductForm**
 - Allows users to add or edit product details.
 - Includes fields like product name, quantity, and price.
- **InventoryList**

- Displays all products in a table or list view.
- Shows real-time updates after any changes.
- **SearchBar**
 - Helps users quickly find specific products by name.
 - Filters the inventory list based on input.
- **ActionButtons**
 - Provides options to edit or delete individual products.
 - Ensures quick inventory management.
- **AlertBox(Optional)**
 - Displays success or error messages after actions like adding or deleting a product.

9. User Interface

- The user interface (UI) of the Store Manager project is designed to be simple, clean, and user-friendly. It helps users easily manage and track inventory without technical complexity.
- **UI Highlights:**
 - **Dashboard View**
 - Displays all inventory items in a structured table or card format.
 - Includes product name, quantity, and price.
 - **Add/Edit Product Form**
 - Clearly labeled input fields for product details.
 - Buttons for submitting or resetting the form.
 - **Search Bar**

- Positioned at the top for easy access.
- Helps users quickly filter products by name.

- **Action Buttons**

- Edit and Delete icons for each product row.
- Prompts confirmation for deletion.

- **Responsive Design**

- Optimized for desktops, tablets, and mobile devices.
- Ensures smooth navigation on all screen sizes.

10. Styling

- **Minimalistic Layout**

- Simple design with clear sections for products, forms, and buttons.
- Focus on usability and clarity.

- **Consistent Color Scheme**

- Light background with contrasting colors for buttons and alerts.
- Different colors indicate different actions (e.g., green for add, red for delete).

- **Responsive Design**

- Works well on desktops, tablets, and mobiles.
- Layout adjusts based on screen size.

- **Interactive Elements**

- Hover effects on buttons and rows.
- Clear visual feedback for user actions.

- **Styling ensures a pleasant user experience, making the app both functional and visually appealing.**

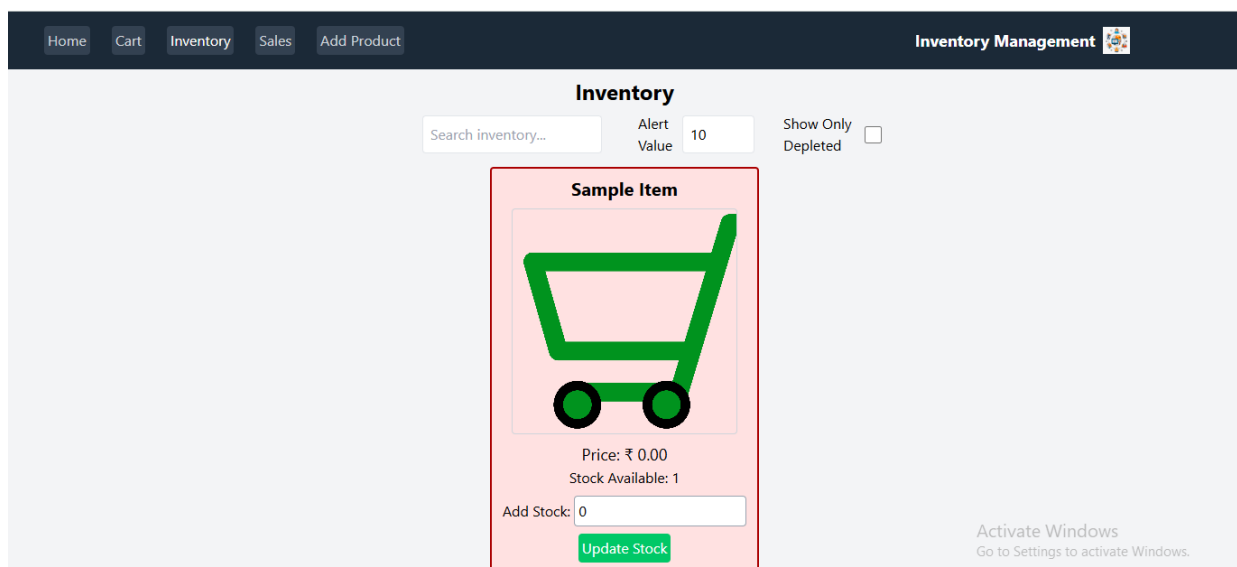
11. Testing

- Testing ensures that all features in the Store Manager app work as expected and handle user input correctly.
 - Manual testing in Chrome and Firefox
 - Console logging for API errors
 - Responsive layout checks
 - Future: Jest for unit testing (App.test.js, setupTests.js)

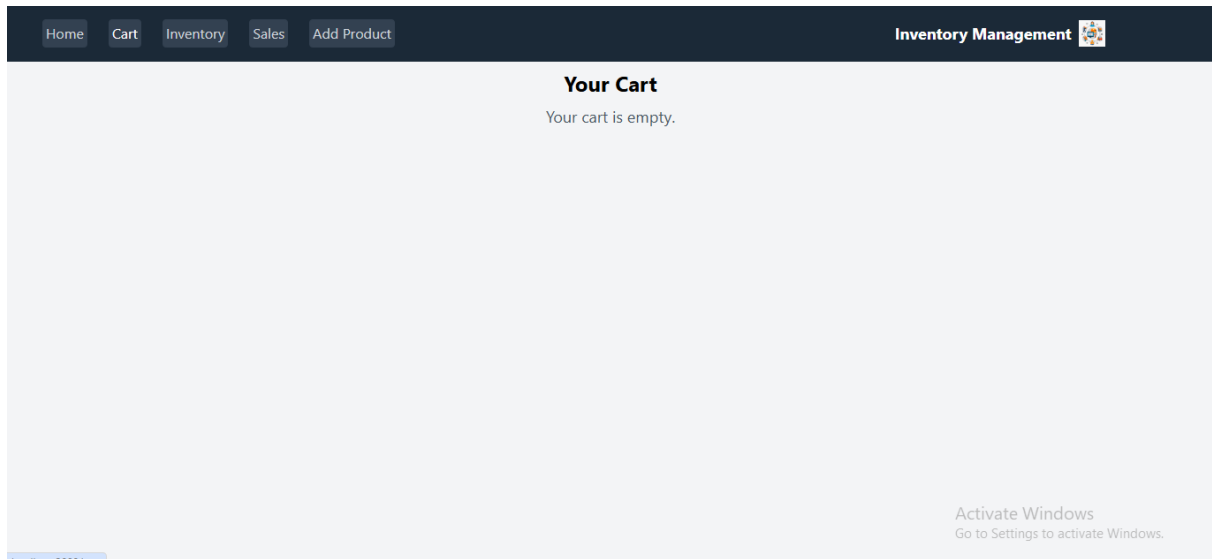
12. Screenshots

Sample Product Output

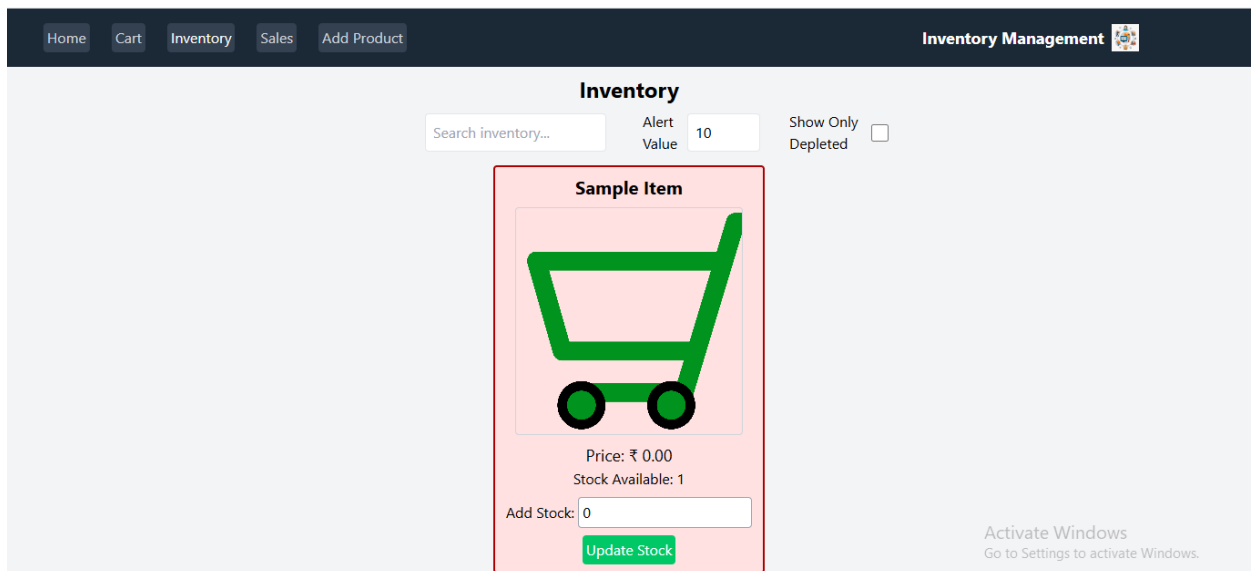
Home Page



Cart Page




Inventory page



Sales Page

HomeCartInventorySalesAdd Product

Inventory Management

Sales Record


No sales recorded yet.

localhost:3000/sales

Activate Windows
Go to Settings to activate Windows.

Add Product

HomeCartInventorySalesAdd Product

Inventory Management

Add New Product

Product Name

Enter product name

Product Image URL

Enter image URL

Price

Enter price

Stock

Enter stock

Tags (comma-separated)

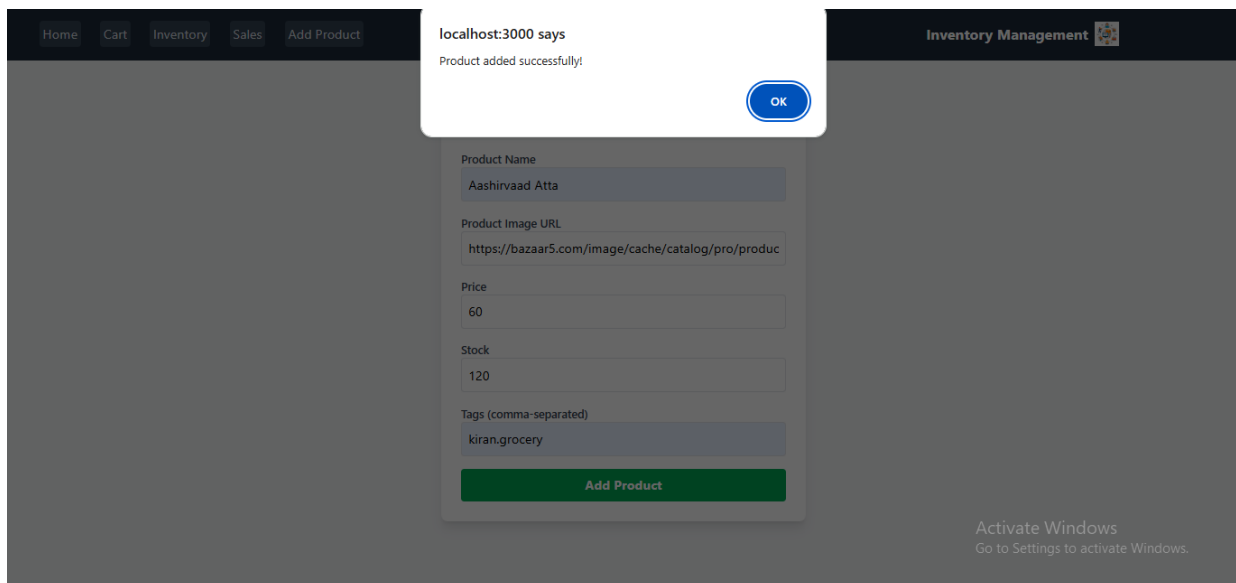
Enter tags, separated by commas

Add Product

localhost:3000/add-product

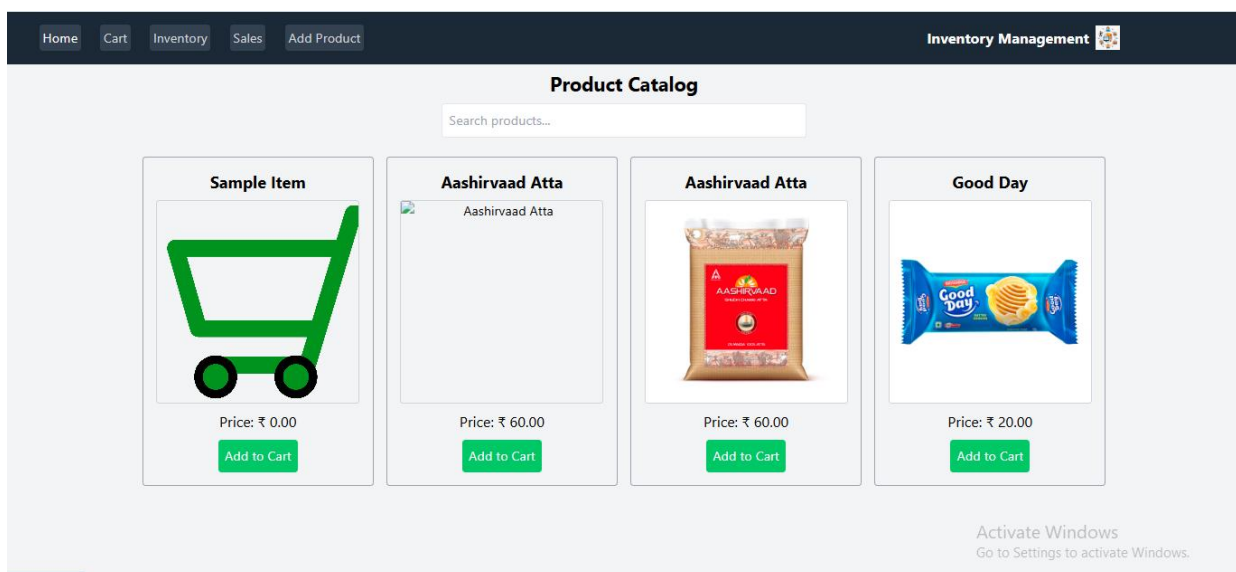
Activate Windows
Go to Settings to activate Windows.

Original Output after adding the Product details in the above screenshot



Now the click the Navigation Bar and then added product and each page will display

Home Page



Card Page

HomeCartInventorySalesAdd Product


Inventory Management

Your Cart

No. of products in cart: 1 | Total Cart Value: 60.00

Checkout CartClear Cart

Aashirvaad Atta



Price: ₹ 60.00

-1+

Remove from Cart

Activate Windows
Go to Settings to activate Windows.

localhost:3000/cart

Inventory Page

HomeCartInventorySalesAdd Product

Inventory Management


Inventory

Search inventory...

Alert Value10

Show Only Depleted

Sample Item

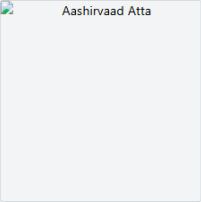


Price: ₹ 0.00
Stock Available: 1

Add Stock:

Update Stock

Aashirvaad Atta




Price: ₹ 60.00
Stock Available: 37

Add Stock:

Update Stock

Aashirvaad Atta




Price: ₹ 60.00
Stock Available: 37

Add Stock:

Update Stock

Good Day



Price: ₹ 20.00
Stock Available: 10

Add Stock:

Update Stock

Aashirvaad Atta

Activate Windows
Go to Settings to activate Windows.

Sales Page

HomeCartInventorySalesAdd Product

Inventory Management

Sales Record

Sale #1

17/09/2025, 22:17:49

Total Sale Value: ₹120.00

Cart Details:

Aashirvaad Atta (2) - ₹120.00

Sale #2

17/09/2025, 23:21:53

Total Sale Value: ₹180.00

Cart Details:

Aashirvaad Atta (3) - ₹180.00

Activate Windows
Go to Settings to activate Windows.

localhost:3000/sales

By filling the details in the add product page you can get a different Products

HomeCartInventorySalesAdd Product

Inventory Management

Add New Product

Product Name

Aashirvaad Atta

Product Image URL

https://bazaar5.com/image/cache/catalog/pro/produc

Price

60

Stock

120

Tags (comma-separated)

kiran.grocery

Add Product

Activate Windows
Go to Settings to activate Windows.

localhost:3000/add-product

13. Known Issues

- While the application performs its core functions well, a few limitations are noted:
- **No Login/Authentication**
 - Any user can access and modify inventory without restriction.
- **No Data Persistence**
 - Inventory data is lost when the app is closed or refreshed (if not connected to a database).

These issues can be addressed in future updates to improve security and reliability.

14. Future Enhancements

- To improve functionality and user experience, the following features can be added in future versions:
- **Authentication**
 - Secure login system to restrict access and track user actions.
- **Database Integration**
 - Store inventory data permanently using a database like MySQL or MongoDB.
- **Low Stock Alerts**
 - Automatically notify the user when product stock is low.
- **Report Generation**

- Export inventory reports in PDF or Excel format.

- **Barcode Scanning Support**

- Add products quickly using barcode input for faster inventory updates.