**PROJECT DOCUMENTATION**

**Project Title: Store Maner - Keep Track of Inventory**

**Team ID: NM2025TMID39995**

Team Leader: Bathurunisha. H

(Email: [sac2427bca5282@ssacollegechennai.com](mailto:sac2427bca5282@ssacollegechennai.com))

Team Members:

• Aswitha Penitta Rani. J

(Email: [sac2427bca5233@ssacollegechennai.com](mailto:sac2427bca5233@ssacollegechennai.com))

• Bhuvaneshwari. V

(Email: [sac2427bca5349@ssacollegechennai.com](mailto:sac2427bca5349@ssacollegechennai.com))

• Divya. B

(Email: [sac2427bca5339@ssacollegechennai.com](mailto:sac2427bca5339@ssacollegechennai.com))

**1. Introduction**

Store Manager is a smart inventory management system that helps businesses efficiently keep track of products, manage stock levels, and reduce manual errors. It ensures that inventory is updated in real time and provides reports to support better decision-making.

**2. Project Overview**

Purpose:   
The purpose of Store Manager is to provide an easy-to-use system that allows businesses to maintain their inventory, monitor sales, and avoid stock shortages or excess storage.

**Features:**• Add, update, and delete products  
• Track stock levels in real time  
• Generate inventory reports  
• Notifications for low stock  
• User dashboard for quick insights  
• Role-based access for admin and staff

**3. Architecture**

Frontend: React.js with Tailwind CSS

Backend: Node.js with Express.js

Database: MongoDB

**4. Setup Instruction**

Prerequisites:  
• Install Node.js  
• Install MongoDB

Installation Steps:  
• Clone the repository  
• Run `npm install`  
• Start server using `npm start`

**5. Folder Structure**

StoreManager/  
■-- public/  
■-- src/  
■-- components/  
■-- pages/

■-- styles/  
■-- App.js  
■-- index.js  
■-- package.json  
■-- README.md

**6. Running in Application**

Frontend: Run `npm start` inside client folder.

Backend: Run `npm run dev` for the server.

Access: Open http://localhost:3000 in the browser.

**7. API Documentation**

**User:**  
• POST /api/user/register → Register new user  
• POST /api/user/login → User login

**Projects:**  
• GET /api/projects → Fetch all projects  
• POST /api/projects → Create new project

**Chats:**  
• GET /api/chats → Fetch user chats  
• POST /api/chats → Create new chat

**8. Authentication**

JWT-based authentication system is implemented to secure user accounts and protect private routes.

**9. User Interface**

The application provides a clean and user-friendly interface with dashboards, product management, and stock status indicators for quick monitoring.

**10. Testing**

Manual testing was performed to validate product entry, stock updates, and API calls. Postman and browser DevTools were used for debugging and validation.

**11. Demo Link**

<https://drive.google.com/file/d/1SZU_Q6ZiJNn47lReJxTQO42GU3wyDF0a/view?usp=drive_link>

**12. Known Issues**

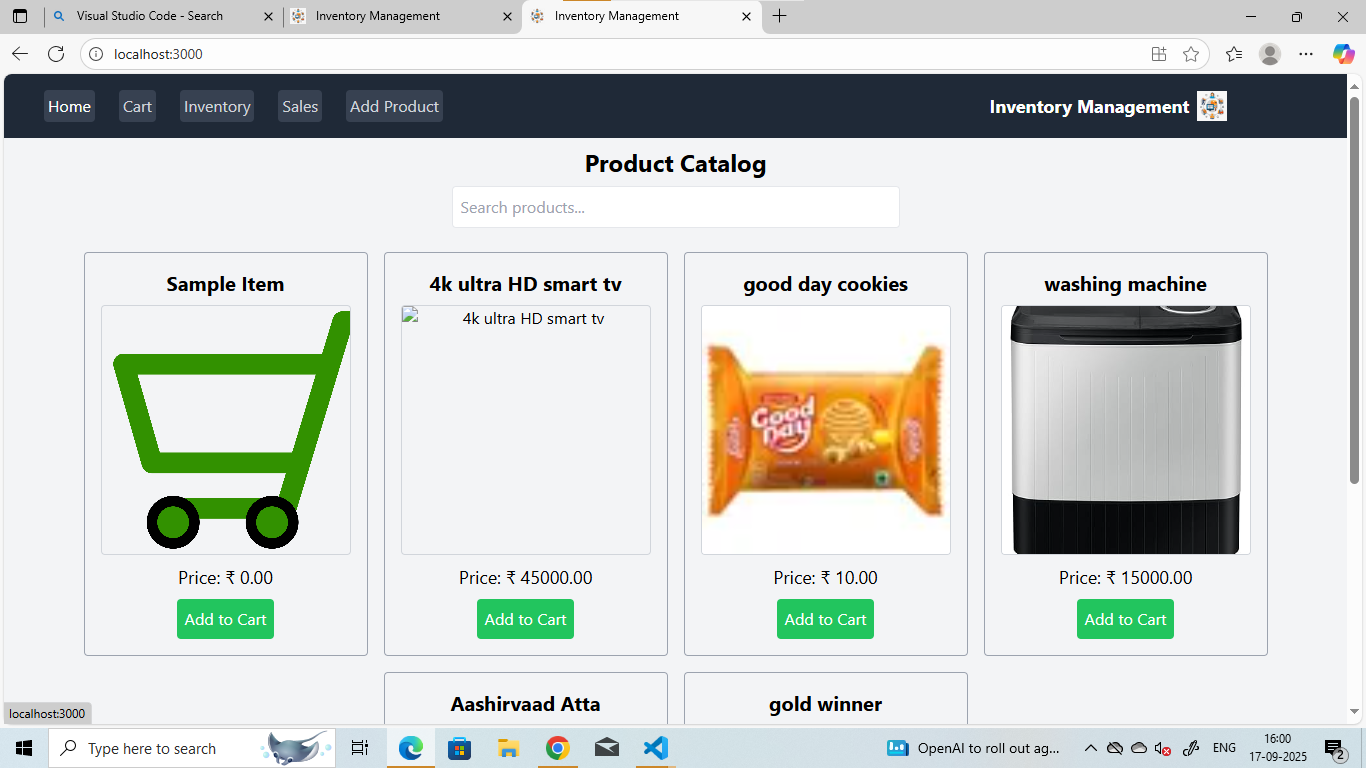
• Limited reporting features in the first version  
• Requires internet connection for real-time updates  
• No offline support initially

**13. Future Enhancement**

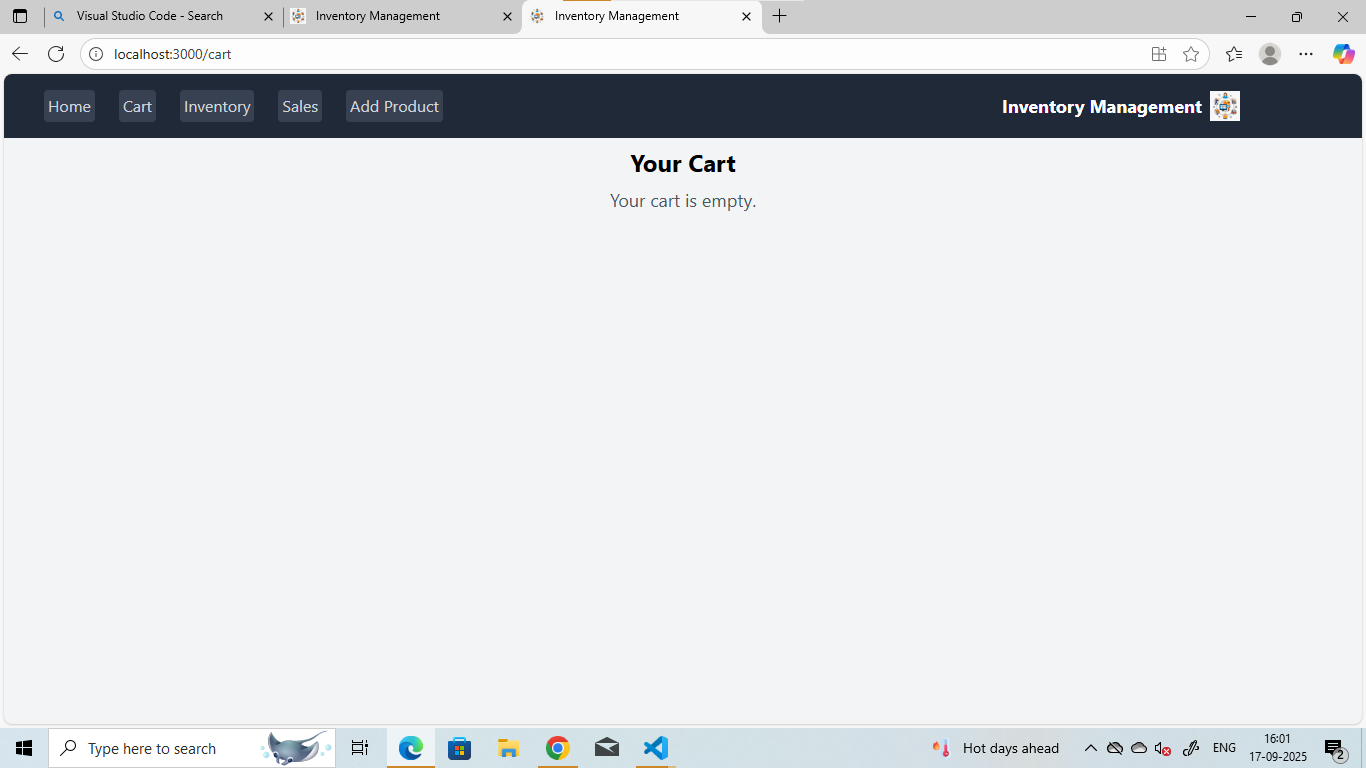
• AI-based sales prediction  
• Barcode scanner integration  
• Mobile app version

• Multi-warehouse management  
• Export reports in Excel/PDF

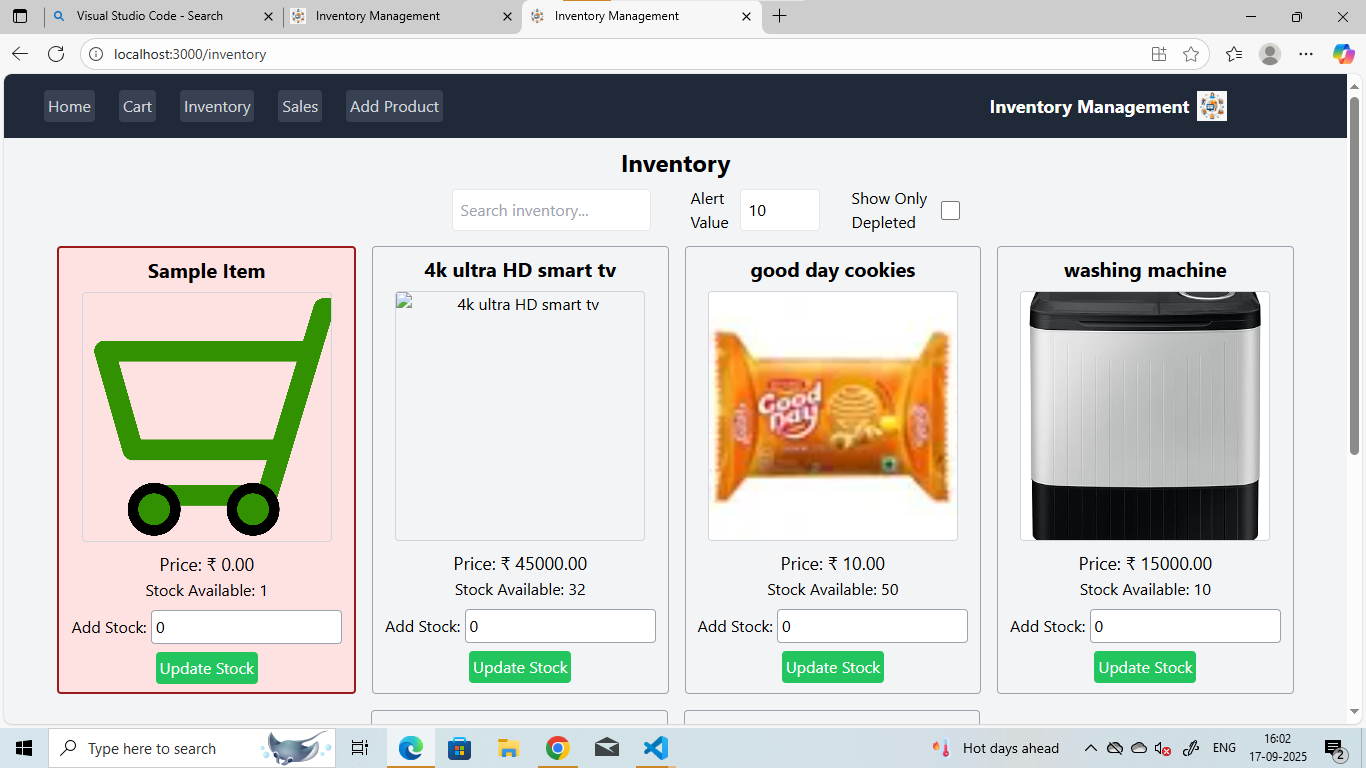
**Screenshots:**



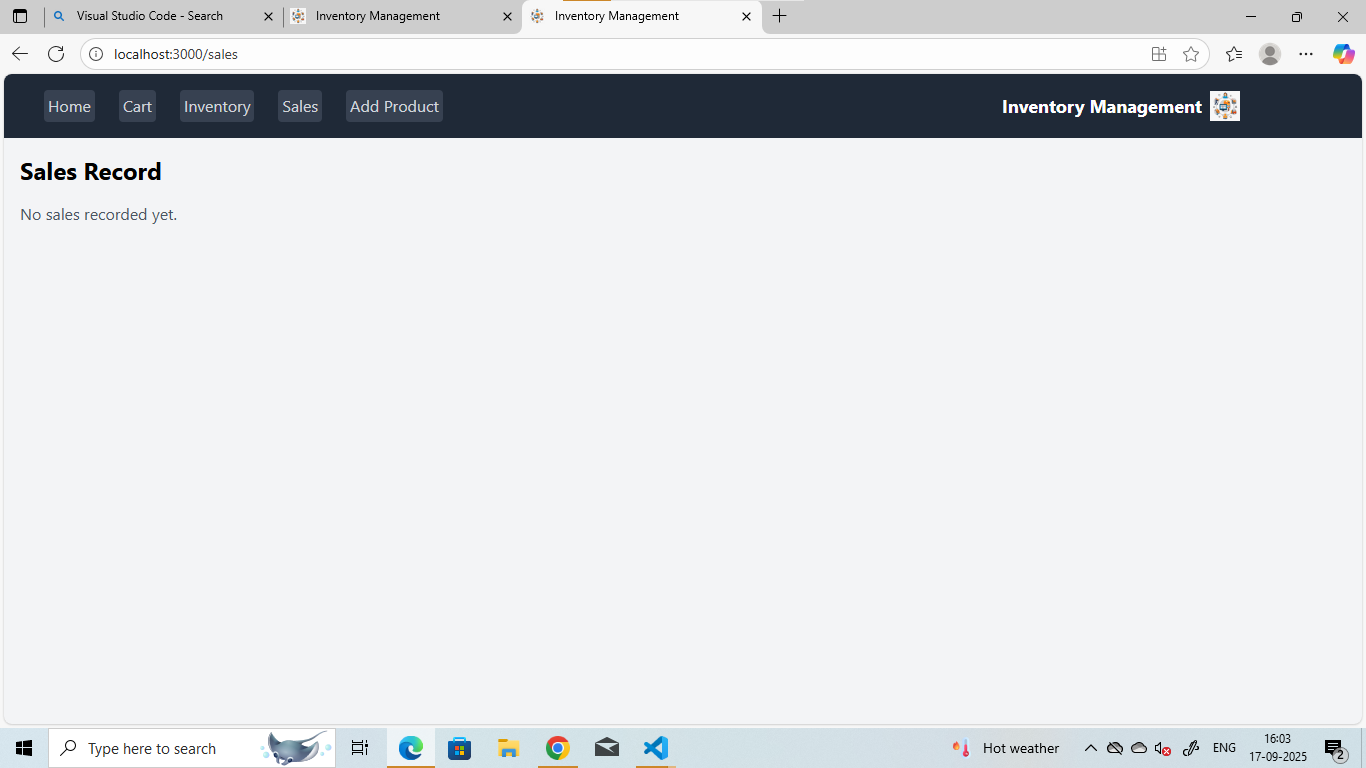
**HOME PAGE**



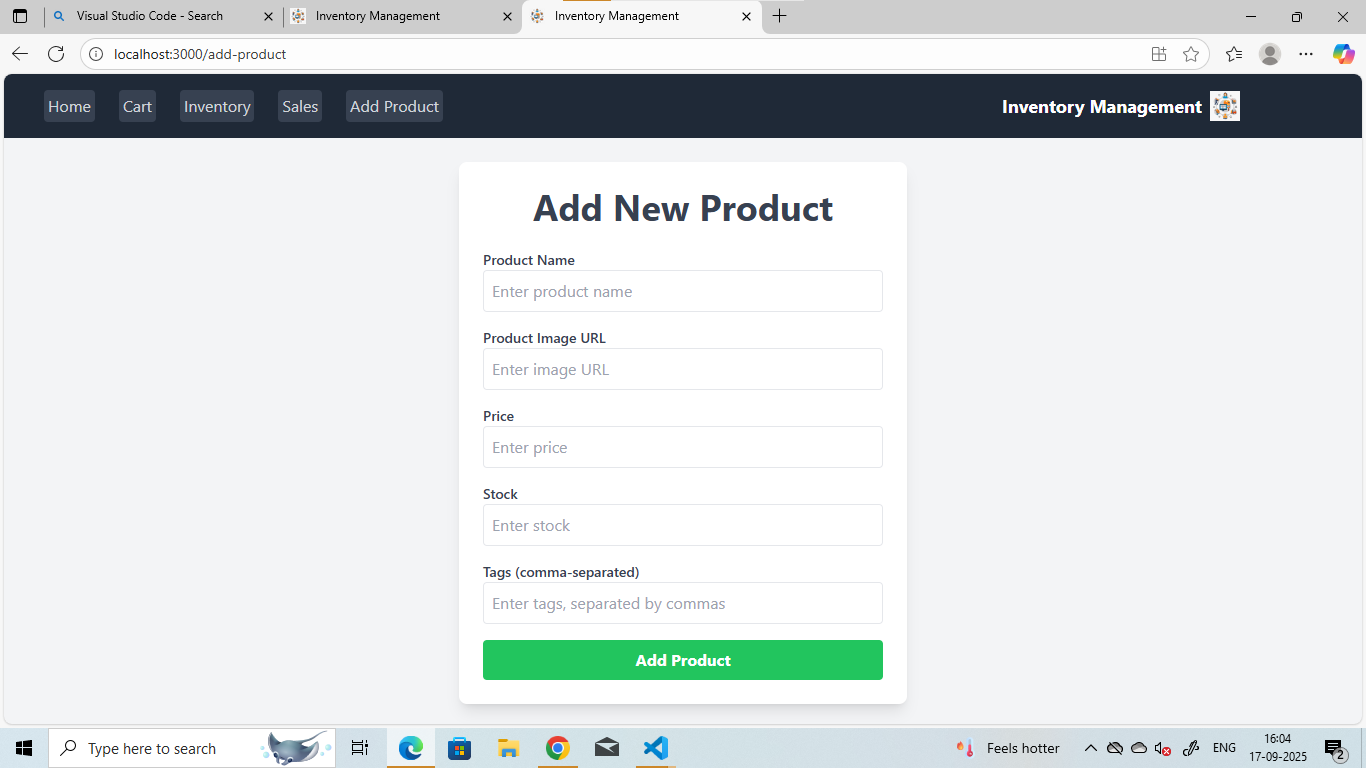
**CART PAGE**



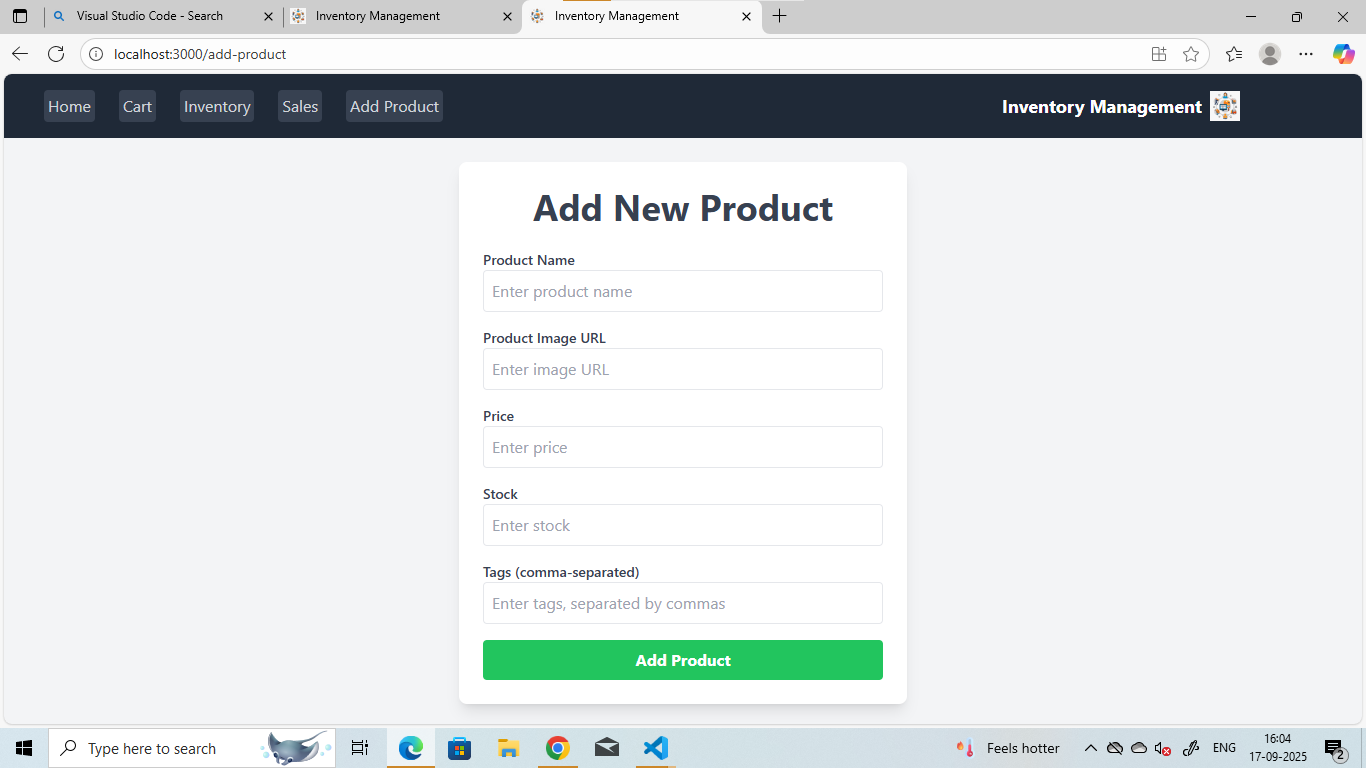
**INVENTORY**



**SALES**

****

**ADD NEW PRODUCT**



**ADD NEW PRODUCTS**