Store Management Project Documentation

# 1. Introduction

• Project Title: Store Manager

• Team ID: NM2025TMID39573

• Team Leader: KUSHAL V & kv5735154@gmail.com

• Team Members:  
 – DHARANI S & yamunadharani7@gmail.com   
 – SANJAY M & msanjaym909@gmail.com

–SANJAY PRASAD A & sanjayprasath2208@gmail.com

# 2. Project Overview

• Purpose: Store Manager helps in keeping track of inventory, managing stock efficiently, and monitoring usage.

• Features:  
 – Inventory tracking  
 – Stock updates in real-time  
 – User-friendly interface  
 – Integration with database  
 – Admin control for better management

# 3. Architecture

• Frontend: React.js with Bootstrap and Material UI

• Backend: Node.js and Express.js

• Database: MongoDB for storing inventory data  
• Tools: Visual Studio Code, Git

# 4. Pre-Requisites

• Download and install Node.js (LTS) from https://nodejs.org/en/download/  
• Open Windows PowerShell as Administrator  
• Run the command: set-executionPolicy unrestricted  
• Confirm with 'Y' and press Enter

# 5. Setup Instructions

• From the SmartInternz portal, go to Projects → Access Resources → Guided Projects → Go To Workspace  
• Access the project flow and download the code folder  
• Extract the downloaded zip file  
• Open the folder in Visual Studio Code  
• Open a new terminal and run: npm install  
• After dependencies are installed, run: npm start  
• Application will be accessible at http://localhost:3000

# 6. Folder Structure

Store-Manager/  
|-- client/ # React frontend  
| |-- components/  
| |-- pages/  
|-- server/ # Node.js backend  
| |-- routes/  
| |-- models/  
| |-- controllers/

# 7. API Documentation

• Inventory:  
 – /api/inventory/add  
 – /api/inventory/update/:id  
 – /api/inventory/delete/:id  
 – /api/inventory/list  
• Authentication:  
 – /api/user/register  
 – /api/user/login

# 8. Running the Application

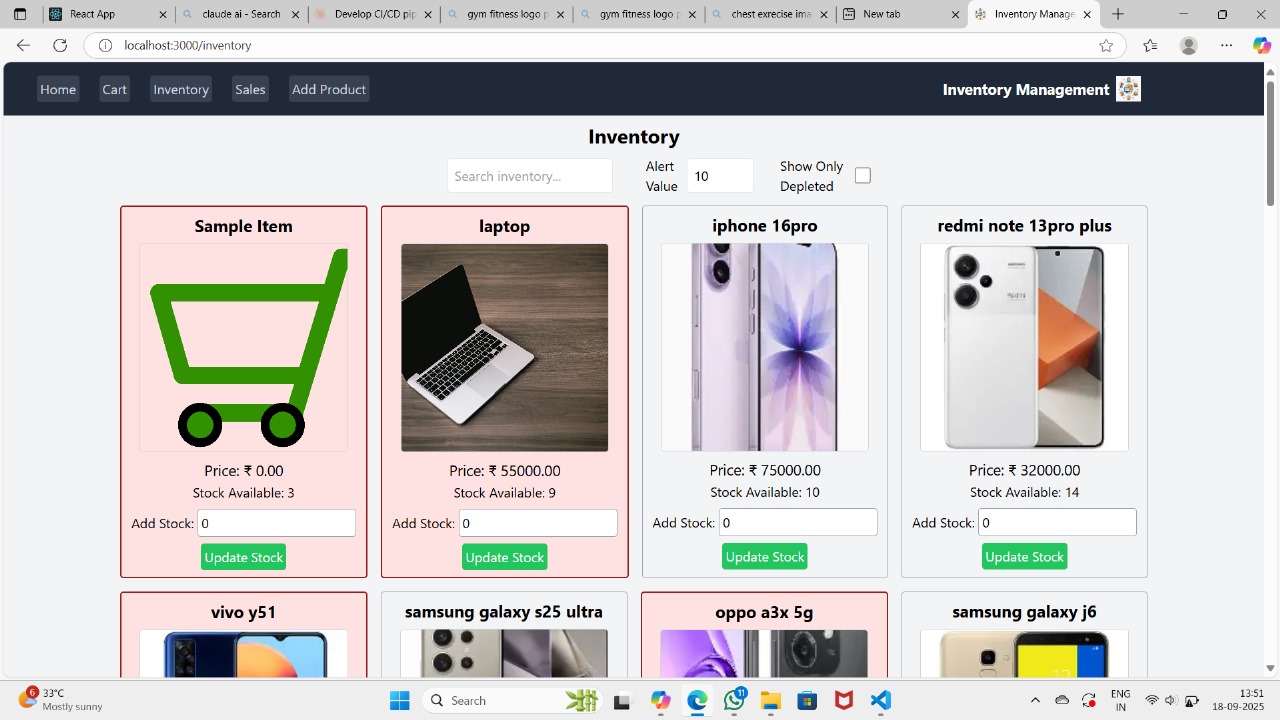
• Frontend:  
 cd client  
 npm start  
• Backend:  
 cd server  
 npm start  
  
• Access via: http://localhost:3000

# 9. Testing

• Manual testing during project milestones  
• Tools: Postman for API testing, Chrome Dev Tools for frontend testing

# 10. Screenshots

# 



# 11. Future Enhancements

• Adding analytics dashboard for inventory insights  
• Implementing barcode/QR scanning for stock management  
• Role-based access control for users and admins  
• Cloud integration for remote access