**Project Documentation**

**1. Introduction**

**Project Title: Store Manager**

**Team ID:** NM2025TMID36629

**Team Leader**: Jithendran .M- [jithendranm10@gmail.com](mailto:jithendranm10@gmail.com)

**Team Members:**

Hemavathy. N [-hemaharinihemaharini1@gmail.com](mailto:-hemaharinihemaharini1@gmail.com)

Jagathratchagan .E [-jagathratchagan077@gmail.com](mailto:-jagathratchagan077@gmail.com)

Kabilesh.S- [kabilesh.s1234@gmail.com](mailto:kabilesh.s1234@gmail.com)

**2. Project Overview**

**Purpose:**

The Store Manager application is designed to help small and medium businesses manage their products, inventory, sales, and billing in an easy-to-use system.

**Features:**

Product catalog with search option

Add, update, and delete products

Shopping cart and sales management

Inventory tracking with stock details

Dashboard for product & sales management

Admin control panel

**3. Architecture**

**Frontend:**

React.js with Bootstrap / Material UI

**Backend:**

Node.js with Express.js

**Database:**

MongoDB for storing product, sales, and user data

**4. Setup Instructions**

**Prerequisites:**

Node.js

MongoDB

Git

React.js

Express.js

Visual Studio Code

**Installation Steps:**

# Clone the repository

git clone <repository-url>

# 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:**

POST /api/user/register

POST /api/user/login

**Products:**

POST /api/products/create

GET /api/products/:id

PUT /api/products/:id

DELETE /api/products/:id

**Sales:**

POST /api/sales/create

GET /api/sales/:id

**Inventory:**

GET /api/inventory

PUT /api/inventory/:id

**8. Authentication**

JWT-based authentication for secure login

Middleware protects private routes

**9. User Interface**

### **Inventory Management**

* Stock Overview
* Low Stock Alerts
* Supplier Details
* Incoming Shipments
* Return/Refund Logs

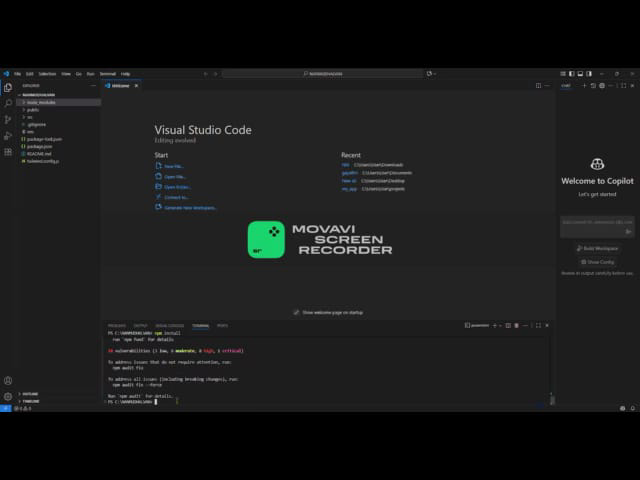
**10. Testing**

Manual Testing done for product creation, cart management, and sales flow

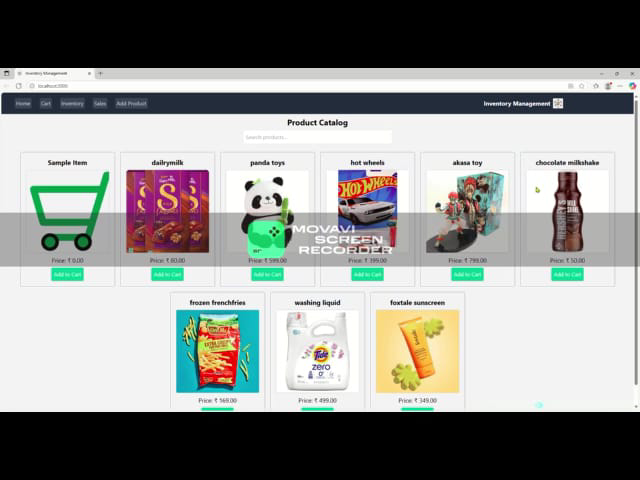
Tools: Postman, Chrome DevTools

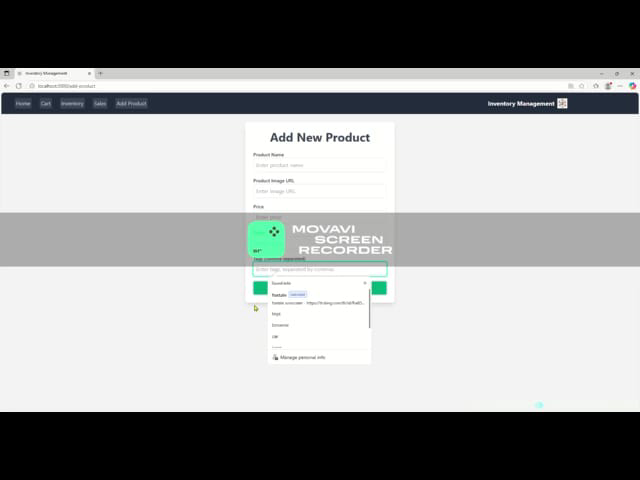
**11. Screenshots / Demo**

**🔹 Development in VS Code**



**🔹 Product Catalog page**

* 🔹 **Add New Product page**



**12. Future Enhancements**

Barcode scanning integration

Mobile app version

Cloud deployment for multi-store support

Sales prediction & analytics with AI

**Demo link**

<https://drive.google.com/file/d/1ilBTUMoVD9QJ_GKaKDhQ4pBL19nskXrR/view?usp=drive_link>