Project Documentation

Store Manager – Keep Track of Inventory

1. Introduction

- Project Title: Store Manager : keep track of inventory
- Team ID :[NM2025TMID40444]
- Team Leader: Dhanalakshmi.M [19052007dhanalakshmi@gmail.com]
- Team Members:
1 - Monisha.R – r.monishabubu1@gmail.com
2 - Keerthika.P – n.p.keerthika1627@gmail.com
3 - Kaviya.K – kkaviya7617@gmail.com

2. project overview

Purpose:

 Store Manager is a front-end application designed to help users manage store inventory efficiently. It allows adding, editing, deleting, and viewing products in a clean and responsive interface.

- Features:
- Product listing and management
- Search and filter functionality
- Responsive design for desktop and mobile
- User-friendly UI for quick access to inventory data

3. Architecture

- Frontend: React.js with Bootstrap and tailwind CSS
- Backend: Node.js and Express.js managing server logic and API endpoints
- Database: MongoDB stores user data, project information, applications, and chat messages.

4. Setup Instructions

- Prerequisites:
- Node.js
- Git
- React .js
- Visual Studio Code

- Installation Steps:
`bash

#Clone the repository
git clone https://github.com/yourusername/store-manager.git

#Navigate to project folder
cd store-manager

#Install dependencies
  npm install

#Start the development server
  npm start
5. Folder Structure
```
store-manager/
├── public/
├── src/
│   ├── components/
│   ├── pages/
│   ├── assets/
│   ├── App.js
│   ├── index.js
├── package.json
└── README.md
```
6. Running the Application

- Frontend:
  `bash
  cd store-manager
  npm start
  `
- Access:
  Visit http://localhost:3000 in your browser
7. API Documentation

- /api/products – Get all products
- /api/products/:id – Get, update, or delete a specific product
- /api/products/create – Add a new product
8. Authentication
- JWTbased authentication
- Protected routes for admin access
9. User Interface

- Home Page
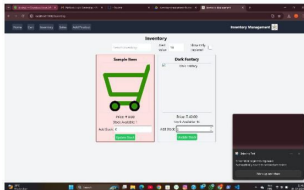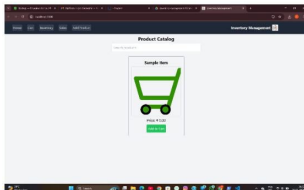- Add Product Form
- Inventory Dashboard
- Product Catalog
- Search and Filter Panel
10. Testing

- Manual testing during milestones
• Tools: Postman, Chrome Dev Tools

11. Screenshots or Demo

12. Known Issues

- No persistent storage without backend
- Limited to local state management

13. Future Enhancements

- Backend integration with Node.js and MongoDB
- User authentication and role-based access
- Export inventory reports
- Cloud deployment