Project Documentation

Store Manager: Keep Track of Inventory

1. Introduction

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

- **Team ID :** NM2025TMID37772
- **Team Leader:** CHARUSMIRTHA M (charuharish1010@gmail.com)

- **Team Members:**
  - DEEPIKA K G (g.deepika1911@gmail.com)
  - NAVYA V (vnnavya2007@gmail.com)
  - MANJARI P (Manjariponnurangam@gmail.com)

2. Project Overview

- **Purpose:** The Inventory Management System helps businesses manage stock efficiently by providing real-time product tracking, sales monitoring, and inventory updates.

- **Features:**
  - Product management (add, update, delete items)
  - Inventory tracking with stock availability
  - Sales and billing management
  - Shopping cart functionality
  - Alert system for low-stock items
  - Search and filter options for products

3. Architecture

- **Frontend:** React.js with Bootstrap and Material UI

- **Backend:** Node.js with Express.js

- **Database:** MongoDB stores user data, project information, applications, and chat messages

- **Other Tools:** Git, Visual Studio Code

4. Setup Instructions

- **Prerequisites:**
  - Node.js
  - MongoDB
  - Git
  - React.js
  - Express.js

Installation Steps:

```
# Clone the repository git clone

# Install client dependencies cd
client npm install

# Install server dependencies cd
../server npm install
```

5. Folder Structure

```
Inventory-Management/
|-- 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 https://localhost:3000

7. API Documentation

- **Products:**
    - POST /api/products/create
    - GET /api/products/:id
    - PUT /api/products/update/:id
    - DELETE /api/products/delete/:id
- **Cart:**
    - POST /api/cart/add
    - GET /api/cart
- **Sales:**
    - POST /api/sales/create
    - GET /api/sales/history

8. Authentication

- JWT-based authentication for secure login
- Middleware protects private routes

9. User Interface

- **Home Page** – Navigation to Cart, Inventory, Sales
- **Inventory Page** – Product list with stock & price details
- **Cart Page** – Items added for checkout
- **Sales Page** – Sales reports and transaction details
- **Add Product Page** – Admin can add/manage products

10. Testing

- Tools Used: Postman, Chrome DevTools
- Testing Approach: Manual testing for CRUD operations, API validations, and stock alerts

11. Screenshots or Demo

demo video (1) (1) (1).mp4

12. Known Issues

- Limited search filters
- No multi user role support (only admin and user)

13. Future Enhancements

- Role-based access control (Admin, Staff, Customer)
- Integration with barcode scanner for faster inventory updates
- Advanced analytics and sales reports
- Cloud deployment

-