Project Documentation

Store Manager: Keep Track of Inventor

1. Introduction

- Project Title: Store Manager – Inventory Tracking System
- Team size: 4
- Team ID: NM2025TMID34878
- Team Leader: **HARISH R** (24ucs519@avasctnj.edu.in)
- Team Members**: KALAIKKANNAN S** (24ucs521@avasctnj.edu.in)
- Team Members: **DEEPAN K** (24ucs510@avasctnj.edu.in)
- Team Members**: HARISH M** (24ucs518@avasctnj.edu.in)

2. Project Overview

Purpose:

Store Manager helps businesses manage and monitor their stock levels effectively. It allows store owners to track product availability, update stock in real time, automate reorder levels, and generate insightful reports.

Features:

Add, update, and delete inventory items

Real-time stock level monitoring

Automatic low-stock alerts and reorder system

Categorization of products (e.g., perishable, non-perishable)

Sales & purchase history tracking

Admin control panel

3. Architecture

Frontend: ReactJS with Bootstrap / Material UI

Backend: Node.js with Express.js (REST API)

Database: MongoDB (stores inventory, categories, stock movements, and user info)

4. Setup Instructions

Prerequisites:

Node.js

MongoDB

Installation Steps:

1. Clone the repository from GitHub
2. Install client dependencies → cd client && npm install
3. Install server dependencies → cd server && npm install
4. Run the project → npm start

5. Folder Structure

```
StoreManager/
  ├── client/      # React Frontend
  ├── server/      # Node.js Backend
  ├── models/      # Database Schemas
  └── controllers/  # API Logic
```

6. Running the Application

Frontend:

```
cd client

npm start
```

Backend:

```
cd server

npm start
```

Access App: http://localhost:3000

7. API Documentation

Users:

POST /api/users/register

POST /api/users/login

Products:

POST /api/products/create

GET /api/products

PUT /api/products/:id

DELETE /api/products/:id

Inventory Movements:

POST /api/inventory/addStock

POST /api/inventory/removeStock

GET /api/inventory/history

8. Authentication

JWT-based authentication for login security

Role-based access (Admin, Store Manager, Staff)

9. User Interface

Dashboard (Overview of stock levels)

Product Management Page

Low Stock Alerts Page

Sales & Purchase History Page

Admin Control Panel

10. Testing

Manual testing during development

Tools: Postman (API testing), Jest (Unit testing)

11. Screenshots or Demo

(Attach screenshots of UI/dashboard here)

12. Known Issues

[List any current limitations/bugs

13. Future Enhancements

Barcode/QR code scanning for product management

Automated supplier order placement

Mobile application support