**STORE INVENTORY MANAGEMENT SYSTEM**

**Project Documentation**

1. **Introduction**

The **Store Inventory Management System** is designed to help businesses efficiently track, manage, and update their stock levels. It minimizes manual errors, provides real-time insights, and ensures smooth day-to-day operations.

* **Project Title:** Store Inventory Management System
* **Team ID:** *[NM2025TMID37246]*
* **Team Leader:** *Rohith.H.M -* [*bcacrohith@gmail.com*](mailto:bcacrohith@gmail.com)
* **Team Members:**
* *Priyan.E –* [*bcacpriyan@gmail.com*](mailto:bcacpriyan@gmail.com)
* *Prithish.S –* [*bcacprithish@gmail.com*](mailto:bcacprithish@gmail.com)
* *Premnath.C –* [*bcacpremnath@gmail.com*](mailto:bcacpremnath@gmail.com)

**2. Project Overview**

* **Purpose:**  
  The Store Inventory Management System streamlines the process of tracking, managing, and updating stock levels for retail or wholesale businesses. It ensures accurate inventory data, reduces manual errors, and improves operational efficiency.
* **Features:**
* Add, edit, and delete product records
* Real-time stock level updates
* Low-stock alerts and notifications
* Category and supplier management
* Sales and purchase tracking
* Role-based access control (Admin, Staff)
* Export reports in PDF/Excel

**3. Architecture**

* **Frontend:** React.js with Bootstrap and Material UI
* **Backend:** Node.js with Express.js for API handling
* **Database:** MongoDB for storing product, category, supplier, and transaction data

**4. Setup Instructions**

* **Prerequisites:**
  + Node.js
  + MongoDB
  + Git
  + React.js
  + Express.js
  + Mongoose
  + 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-Inventory/

│-- 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**

* **Products:**
  + POST /api/products/create
  + GET /api/products/:id
  + PUT /api/products/:id
  + DELETE /api/products/:id
* **Categories:**
  + POST /api/categories/create
  + GET /api/categories
* **Suppliers:**
  + POST /api/suppliers/create
  + GET /api/suppliers
* **Transactions:**
* POST /api/transactions/sale
* POST /api/transactions/purchase

**8. Authentication**

* JWT-based authentication for secure login
* Middleware to protect admin and staff routes

**9. User Interface**

* Dashboard with stock overview
* Product management page
* Category and supplier management pages
* Sales and purchase entry forms
* Reports page

**10. Testing**

* Manual testing during development milestones
* Tools: Postman, Chrome DevTools

**11. Screenshots or Demo**

<https://drive.google.com/file/d/1Dl102HbQOAssYOk7lc8Ok6YuXnozNRRI/view?usp=sharing>

**12. Known Issues**

* Barcode scanner integration pending
* Mobile UI optimization in progress
* Can’t edit already created inventory

**13. Future Enhancements**

* AI-based demand forecasting
* Multi-store inventory synchronization
* Integration with POS systems