**Project Documentation**

**1. Introduction**

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

**Team ID:** [NM2025TMID42703]

**Team Leader:** [SHREEMATHI.P & [mathi2007p@gmail.com](mailto:mathi2007p@gmail.com)]

**Team Member:**

[BHUVANA.V & [bhuvanavenkat0904@gmail.com](mailto:bhuvanavenkat0904@gmail.com)]

[SADHANA.K & [sadhanasadhuk@gmail.com](mailto:sadhanasadhuk@gmail.com)]

[TAMILARASI.S & [tamilsiva1434@gmail.com](mailto:tamilsiva1434@gmail.com)]

[VIJAYASANTHI.S & [vijayasanthiselvam2006@gmail.com](mailto:vijayasanthiselvam2006@gmail.com)]

**2. Project Overview**

**Purpose:**

Store Manager is an inventory management system designed to help businesses track stock, manage product details, and streamline operations. It ensures real-time updates of product quantities, sales, and purchases, reducing manual errors and improving efficiency.

**Features:**

* Product listing with categories
* Real-time stock tracking
* Low-stock alerts and notifications
* Add, edit, delete products
* Sales and purchase tracking
* User authentication (Admin & Staff)
* Reports and analytics dashboard

**3. Architecture**

**Frontend:** React.js with Bootstrap / Material UI

**Backend:** Node.js with Express.js handling APIs and server logic

**Database:** Mongo DB for storing product details, stock levels, users, and sales records

**4. Setup Instructions**

**Prerequisites:**

* Node.js
* Mongo DB
* Git
* React.js
* Express.js
* Visual Studio Code

**Installation Steps:**

# Clone the repository

git clone <repository-link>

# Install client dependencies

cd client

npm install

# Install server dependencies

cd ../server

npm install

StoreManage/

|-- client/ # React frontend

| |-- components/

| |-- pages/

|

|-- server/ # Node.js backend

| |-- routes/

| |-- models/

| |-- controllers/

|

|--database/ # MongoDB collections

**6. Running the Application**

**Frontend:**

cd client

npm start

**Backend:**

cd serve

npm start

**Access:** Visit http://localhost:3000

**7. API Documentation**

**User:**

* POST /api/user/register
* POST /api/user/login

**Products:**

* POST/api/products/add
* GET /api/products/list
* PUT /api/products/update/:id
* DELETE /api/products/delete/:id

**Inventory:**

* GET /api/inventory/stock
* POST /api/inventory/update

**Reports:**

* GET /api/reports/sales
* GET /api/reports/stock-status

**8. Authentication**

* JWT-based authentication for secure login
* Role-based access (Admin & Staff)
* Middleware to protect private routes

**9. User Interface**

* Login & Registration Page
* Dashboard with stock overview
* Product Management Page
* Sales & Purchases Page
* Low-Stock Alerts Page
* Reports & Analytics Dashboard
* Admin Control Panel

**10. Testing**

**Methods:** Manual testing at each milestone

**Tools:** Postman, Chrome DevTools

**11. Screenshots or Demo**

(<https://drive.google.com/file/d/1al5ziwC9FBXAtpULhkBTI5Cg8j55BDxQ/view?usp=drivesdk>)

**12. Known Issues**

* Bulk product import may slow down for very large files
* Limited offline functionality

**13. Future Enhancements**

* Barcode/QR code scanner integration
* Multi-store inventory tracking
* Automated invoice generation
* Mobile app for Android/iOS