PROJECT DOCUMENTATION -

STORE MANAGER KEEP TRACK OF INVENTORY

1. **INTRODUCTION :**

* PROJECT TITLE:STORE MANAGER KEEP TRACK OF INVENTORY
* TEAM ID : NM2025TMID37852
* TEAM LEADER:Lakshya v

[lakshyavin07@gmail.com](mailto:lakshyavin07@gmail.com)

* **TEAM MEMBERS**:

Priya dharshini S

Jaishree suresh

Harini D

Samrin ashiba s

**2.PROJECT OVERVIEW**:

* **PURPOSE**:

The purpose of this system is to help store managers maintain accurate, real-time control over inventory so that the business can run efficiently and profitably.

* **FEATURES**:

1.Product Management

* Add, edit, or delete products
* Assign product codes, categories, and

pricing

2.Stock Monitoring

* Real-time tracking of available stock Minimum stock alerts and automatic reorder reminders

3.Sales & Purchase Integration

* Stock automatically updates with each sale or purchase Generate invoices and purchase orders

**3.ARCHITECTURE**

* **Component documentation**
* User Interface (UI) Component
* Product Management Component
* Inventory Tracking Component
* Sales & Billing Component

* **State management**
* State management keeps track of product

availability, sales, purchases, and user actions.It ensures that every action (sale, purchase, return, or stock update) immediately changes the system's state, so managers always see the most accurate inventory data.

* **Routing**

1. Authentication Routes

* Login page-User enters credentials
* Role-based access[Admin,manager,staff ]
* Logout jlogout-ends session

2.Dashboard Route

* ·Dashboard (/dashboard)- Main homepage after login.
* Quick overview of stock levels.
* Alerts for low stock or expiring items.
* Sales & purchase summaries

3. Product Management Routes

* Product List (/products) - View all products.
* Add Product (/products/add)- Enter newproduct details
* Edit Product (/products/edit/:id)- Update existing product info
* Delete Product (action route)- Remove product from system

**4.SETUP INSTRUCTIONS**

* **Prerequisites**
* The prerequisites ensure that the system has the necessary hardware, software, user skills, and security setup for smooth functioning
* **Installation**

1.Hardware Setup

* Ensure the system meets minimum

hardware requirements (Processor: i3 or higher, RAM:4GB+, Storage: 250GB+)· Connect optional devices like barcode scanner and receipt/invoice printer if needed

2.Software Setup

1. Install Required Software Packages

* Web Server (Apache/Nginx)
* Database (MySQL/Post/MongoDB)

Backend Framework (Node.js/Django/ Spring Boot)

* Frontend Framework(React.js/Angular/

Vue.js)

2.Install Browser (Google Chrome/Firefox/

Edge) for accessing the system

* Create a new database for the inventory system.
* Import the database schema (tables for products,sales, purchases, suppliers users)
* Configure connection settings in the

backend (DB host, username, password)

5**.FOLDER STRUCTURE**

* **Client**
* The clients of the system include store managers, staff, administrators, suppliers, and business owners. Indirectly, even customers benefit because inventory is always up-to-date, ensuring product availability
* **Utilities**

1.Stock Control Utility

* Tracks real-time stock levels.
* Prevents overstocking or stock shortages
* Ensures products are always available when SSneeded.

2.Sales Management Utility

* Automatically deducts stock during sales.
* Generates invoices and receipts
* Helps in monitoringperformance.

3.Purchase & Supplier Utility

* Manages supplier information
* Creates purchase orders when stock is low
* Tracks supplier history and performance

**6.RUNNING THE APPLICATION**

* **Fronted**
* The frontend acts as the user-facing layer of the inventory system, ensuring store staff and managers can interact easily with products sales, purchases, and reports in a clear and responsive interface

**7.COMPONENT DOCUMENT**

* **Key Components**

1.User Interface(UI) Component

* Provides dashboards and navigation Menus
* Allows managers and staff to interact with

the system

2.Product Management Component

* Stores and manages product details (name,

SKU, category, price)

* Supports adding, editing, and deleting products.

3.Inventory Tracking Component

* Monitors stock levels in real-time
* Generates alerts for low stock or

out-of-stock items

* **Reusable Components**
* The reusable components include Authentication, Product Forms, Search & Filters, Tables, Notifications, Repor Generator, Validation, API Services and Dashboard Widgets - all designed to be used multiple times across the system for efficiency and consistency.

**8.STATE MANAGEMENT**

* **Global State**

1.Product State

* Stores details of all products (ID,name,Category, price, quantity
* shared across Inventory, Sales, and Purchase modules

2. Inventory State

* Current stock levels of all items.
* Updates automatically when sales or

purchases occur.

* Accessible by Dashboard, Reports, and Alerts

3.Sales State

* Stores all completed, pending, and returned sales transactions.
* Used by Reports, Inventory (to dect Stock), and Billing
* **Local state**

* The Local State of the Store Manager

Inventory System includes form inputs,

search filters, transaction progress, Ul

controls,session data, and validation

whichare temporary and specific to individual module

**9.USER INTERFACE**

* The User Interface (UI) is the part of the

system that allows users (store managers,

staff, and admins) to interact easily with inventory system. A well-designed Ul improves usability, reduces errors, and speeds up

inventory management

**10.STYLING**

* **CSS frameworks/libraries**
* CSS Frameworks/Libraries like Boot strap.Tailwind CSS, Materialize, and Bulma are used to design responsive, visually appealing, and user-friendly frontend interfaces for the Store Manager Inventory System, saving development time and ensuring consistency.
* **Theming**

1.Color Scheme

* a consistent color palette for dashboards, tables,buttons, and alerts

2.Typography

* Consistent font styles and sizes for headings, labels, tables, and buttons.

**11.TEXTING**

* **TEXTING STARATEGY**
* The Testing Strategy of the Store Manager Inventory System includes Unit Testing Integration Testing, System Testing Functional Testing, Performance Testing, Security Testing, UAT, and Regression Testing to ensure the system is accurate, secure, and reliable.
* **CODE COVERAGE**

1.Importance of Code Coverage

* Ensures critical functionalities are tested
* ldentifies untested or redundant code

2.Statement Coverage

* Ensures every line of code executes atleast once

3.Branch Coverage

* Tests all possible paths in decision-making structures.

**12. DEMO**

* Click here for demo video

<https://drive.google.com/file/d/15xJmcfsw0HI3x_yR7svoUL0IF3GRlidZ/view?usp=drivesdk>

**13.Known Issues**

* The known issues of the Store Manager Inventory System include performance limitations, multi-user synchronization challenges, network dependency, mobile responsiveness, scanner integration, user errors, reporting limitations, and potential security gaps.

**14.Future Enhancements**

1.Mobile Application

* Develop a mobile app version for Android/IOS
* Allows store managers and staff to access inventory and sales on the go

2.Cloud-Based System

* Host the system on the cloud for better accessibility and backup
* Enables multi-store management and real-time data synchronization

3.Barcode/QR Code Integration

* Add barcode scanning for faster product

entry, sales, and stock updates.

* Reduces manual e and improves efficiency

***THANKYOU***