**PROJECT DOCUMENTATION**

**STORE MANAGER – KEEP TRACK OF INVENTORY**

**1.INTRODUCTION**

**PROJECT TITLE** **–** STORE MANAGER – KEEP TRACK OF INVENTORY

**TEAM ID –** NM2025TMID29970

**TEAM LEADER -**

**NAME**: R.DHARSHINI

**EMAIL ID**: 202400362@sigc.edu

**TEAM MEMBERS** –

**NAME**: R.POOJA SRI

**EMAIL ID**: [202400106@sigc.edu](mailto:202400106@sigc.edu)

**NAME**: E.KESHIKA

**EMAIL** **ID**: 202400103@sigc.edu

**NAME**: G.LOSEKA SRI

**EMAIL** **ID**: 202400546@sigc.edu

**2.PROJECT OVERVIEW**

**PURPOSE** – This project is to help the store manager efficiently monitor and manage the inventory of the store. It aims to ensure that all products are accounted for, properly stocked and available when needed without overstocking or running out of items.

**FEATURES** -

* Real – Time Inventory Tracking
* Product Categorization
* Stock in /out management
* Supplier information management
* Cost and pricing management
* Barcode/QR code integration
* User access control

**3.ARCHITECTURE**

* **FRONTEND** – HTML,CSS,Java script, Frame work(optional)
* **BACKEND** – Node.js with Express or Python with Flask/Django, REST API handling
* **DATABASE** – MySQL, PostgreSQL

**4.SETUP** **INSTRUCTIONS**

**PREREQUISITES** –

* Node.js
* MongoDB
* Git
* React.js
* Express.js – Visual studio code

**INSTALLATION STEPS**

* Requirement Analysis
* Choose the software/platform
* Population options:Odoo,Zoho Inventory,Square,Tally,or custom-bulit system

**5.FOLDER STRUCTURE**

Src/models/ - Defines DB table

src/routes/ - Handles user action

Src/services/ - Core logic

Templates/ - HTML Pages for UI

Static/ - CSS, JS, Images

Migration/ - Keeps track of database schema changes

**6. RUNNING THE APPLICATION**

**FRONTEND:**

cd client

npm start

**BACKEND**:

cd server

npm start

**ACCESS:**

Open browser → [http://localhost:3000](http://localhost:3000/)

**7.API DOCUMENTATION**

**1.USER**

POST/api/auth/login

GET/api/products/{id}

**2.PROJECT**

CREATE/api/products/{id}

POST/api/transaction/in

**3.CHAT**

GET/api/reports/low-stock

**8.AUTHENTICATION**

User logs in with credentials – gets JWT token

User sends token in every API request (header)

Backend verifies token – grants/denies access based on role

**9.USER INTERFACE**

**1.LOGIN PAGE**

- Username input

- Password input

- Login button

**2.DASHBOARD**

- Shows key stats at a glance

**3.PRODUCT LIST PAGE**

- With minimal details

**4.REPORTS PAGE**

**10.TESTING**

Testing is done after implementation and before final submission to ensure core features worked correctly.

Tools – Logbook/Register, Stock Cards, Calculator, File records.

**11.DEMO LINK**

<https://drive.google.com/drive/folders/1He427USdz84OK7k1zBz7AbaQ76Dw_06m?usp=drive_link>

**12. KNOWN ISSUES**

* Currently does not support multiple store branches
* Graphs may load slowly with large datasets

**13. FUTURE ENHANCEMENTS**

* Multi-branch store support
* Mobile app version
* Real-time push notifications
* AI-driven demand forecasting
* Barcode/QR scanner integration

**THANK YOU**