KEEP TRACK OF INVENTORY

TEAM ID : NM2025TMID29991

TEAM LEADER : S.**Tharini** 202400015@sigc.edu

TEAM MEMBERS : >> R.Shalinipriya & 202400383@sigc.edu

>> K.Riya sri & 202400186@sigc.edu

>> S.Sivasakthini & 202400351@sigc.edu

PROJECT OVERVIEW :

This project aims to develop a reliable and efficient system for monitoring, managing, and optimizing inventory levels across a business or organization. Whether it's physical goods in a warehouse, office supplies, or digital assets, the system ensures accurate tracking, timely updates, and streamlined operations.

KEY FEATURES :

>>Real-Time Inventory Tracking : Provides up-to-date visibility into stock levels across locations, helping avoid stockouts and overstocking2.

>>Cycle Counting : Allows periodic checks of inventory subsets without disrupting operations, improving accuracy over time.

>>Barcoding & Scanning : Speeds up data entry and reduces human error by using barcode labels and handheld scanners.

>>Automated Reordering & Stock Alerts : Sends alerts when stock falls below a threshold and can auto-generate purchase orders.

>>Centralized Inventory Management : Consolidates data from multiple warehouses or stores into one dashboard for easy monitoring.

>>Quality Control & Inspection : Tracks batch numbers, expiry dates, and warranty details to ensure product quality.

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.

SET UP INSTRUCTIONS :

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

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

API DOCUMENTATION

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

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

TESTING :

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

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

XXXX