Project Documentation

Project Title

1.Introduction:

* Project Title: Store Manager
* Team leader: Yuvarani.S yuvasriyuvarani58@gmail.com
* Team member:

Yuvashree.N [yuvashreenagapooshamam2006@gmail.com](mailto:yuvashreenagapooshamam2006@gmail.com)

Lakshana .R [lakshanaraja1986@gmail.com](mailto:lakshanaraja1986@gmail.com)

Dhanalakshmi.R [dhanalakshmi032506@gmail.com](mailto:dhanalakshmi032506@gmail.com)

Soniya.S [soniyasureshsoniya92@gmail.com](mailto:soniyasureshsoniya92@gmail.com)

2.Project Overview:

The Store Manager project is a software application designed to streamline and simplify the day-to-day operations of a retail store. It helps store owners and staff efficiently manage inventory, sales, billing, customer information, and reporting within a centralized system.

**1. Store Operations Management**

* Oversee daily operations of the software store (physical or online).
* Ensure smooth functioning of sales, billing, and inventory processes.
* Maintain store policies and standard operating procedures.
* Supervise billing and invoicing processes.
* Manage store budget, expenses, and profitability.
* Generate financial and sales reports for management review.

3.Architecture:

### ****Client Layer (Frontend / User Interface)****

* **Users:** Admin, Store Manager, Staff, Customers
* **Purpose:** Provides an easy-to-use interface for interacting with the system.
* **Features:**
  + Product browsing and search
  + Billing and checkout screen
  + Customer management dashboard
  + Reports & analytics view

(Example technologies: React, Angular, Vue.js, HTML/CSS/JS)

4.Setup Instruction:

### ****Prerequisites****

Before starting, install the required tools:

* **Operating System:** Windows / Linux / macOS
* **Software:**
  + Node.js (if backend is in Node) / Python (if backend in Django/Flask) / Java (if backend in Spring Boot)
  + MySQL or PostgreSQL (for database)
  + Git (for version control)
  + VS Code or any IDE (for development)
  + Web browser (Chrome/Firefox)

5.Folder structure:

* **backend/** → API, business logic, database models.
* **frontend/** → UI components, pages, services.
* **database/** → SQL schema, migrations, sample data.
* **docs/** → Architecture diagrams, project report, setup guide.
* **tests/** → Automated tests for both frontend & backend.

6.Running the application:

### ****Start the Database****

* Make sure **MySQL/PostgreSQL** is installed and running.
* Create the database (if not already done):

CREATE DATABASE software store;

7.Componenet Documentation:

## **Authentication Component**

* **Purpose:**  
  Manages user login, registration, and access control.
* **Inputs:**
  + Username, password
* **Outputs:**
  + Access token / session ID
  + User role (Admin, Manager, Staff)
* **Interactions:**
  + Validates user with the **Database**.
  + Provides role-based access to different modules.

8.State Management:

### ****Server State (Fetched from backend)****

* Data that comes from an API or database.
* Example:
  + Sales data from backend for generating reports.
  + Inventory stock counts.
* Tools:
  + React Query, Axios, SWR
  + Angular HttpClient
  + Vue Query

9.User interface:

## **Login & Authentication Screen**

* **Features:**
  + Username & Password fields
  + "Login" button
  + "Forgot Password" option
* **Purpose:** Secure access with role-based login (Admin, Manager, Staff).
* **UI Elements:**
  + Simple form with input boxes
  + Error messages for invalid login
  + Logo + branding

10.Styling:

## **Color Palette**

Use a professional and minimal color scheme:

* **Primary Color (Buttons, Highlights):** #2563eb (Blue)
* **Secondary Color (Headers, Cards):** #1e293b (Dark Gray/Blue)
* **Background:** #f8fafc (Light Gray/White)
* **Text (Primary):** #0f172a (Almost Black)
* **Text (Secondary):** #64748b (Muted Gray)
* **Alerts:**
  + Success → Green #16a34a
  + Error → Red #dc2626
  + Warning → Yellow #f59e0b

11.Testing:

### ****System Testing****

* **Purpose:** Validate the entire application workflow.
* **Examples:**
  + Admin logs in → adds new product → performs a sale → checks reports.
  + Customer purchase generates correct invoice.

12.Screenshot:

**Software Store Management**, covering roles, responsibilities, architecture, and best practices:



13.Known issues:

* **Stock mismanagement:** Overselling a product due to inaccurate stock tracking.
* **License key errors:** Duplicate, expired, or invalid keys causing customer complaints.
* **Obsolete software:** Older versions still listed without clear upgrade paths

14.Future enchanment:

### ****Enhanced User Experience****

* **AI-based Recommendations:** Suggest software based on user behavior, purchase history, or trends.
* **Personalized Dashboards:** Users see software suited to their OS, past purchases, or interests.
* **Voice Search & Chatbot Support:** Quick searches, troubleshooting, and installation guidance via AI assistants.
* **Multi-language Support:** Localized UI for global users.
* **Dark Mode & Accessibility Options:** Improved usability for all users.