**Store manger-inventory tracking system**

**Documentation**

# 1. Introduction

* **Project Title: Store Manger - Inventory Tracking System**
* **Team Leader: Nivetha W**
* **Team members: Mohana Priya S**
* **Team members: Sathana M**
* **Team members: Sangeetha V**
* **Team members: Priya Bharathi T**

# 2.Project Overview

* **Purpose:**  To store manager is a software solution designed to help retail businesses manage their day-to-day operations efficiently. It centralizes inventory, sales, employee management, and customer interactions into one streamlined platform

### **Key Features**

### 🛒 **Inventory**: Add/update products, stock alerts

### 💳 **Sales**: POS system, billing, discounts

### 👥 **Employees**: Roles, attendance

### 📊 **Reports**: Sales trends, charts

### 🧍 **Customers**: Profiles, history

### 🔐 **Security**: Login, role-based access

# 3.Architecture

# Component Structure:

Product: List, view, and manage products.

Inventory: Track stock levels and updates.

Cart: Add/remove items, calculate totals.

Checkout: Billing and payment flow.

User: Login, profile, and roles.

Admin: Dashboard for analytics and control

**State Management:**

* Context API—used for managing global s **Local State**: useState for simple UI logic.
* Share global state (e.g. auth).
* **Redux/Zustand**: Centralized store for complex flows.

Routing**:**

* **Frontend Routes**:
  + /products, /cart, /checkout, /admin, /login
* **Backend API**:
  + GET /api/products, POST /api/cart, PUT /api/inventory/:id

# 4. Setup Instructions

* **Prerequisites:**

▪ -Node.js(v18 or higher)

-npm or yam

* **Installation:**

1 .Clone the Repository:

’’’bash

git clone <http://github.com/nivetha2007/Store-Manager-Inventoty-Tracking.git> cd

,,,

2.Install Dependencies:

’’’bash

Npm install

,,,

.3Run the Application:

’’’bash

npm

’’’

# 5. Folder Structure

* Store Manager/
* ├── Inventory/
* │ └── Products.xlsx
* ├── Sales/
* │ └── Daily Reports/
* ├── Staff/
* │ └── Employee Info.docx
* ├── Settings/
* │ └── Store Profile.txt

# 6. Running the Application

1. **Install App**: Choose a lightweight app like Vyapar, MyBillBook, or Loyverse POS.
2. **Create Account**: Sign up with store details (name, address, tax info).
3. **Add Products**: Enter items with price, quantity, and category.
4. **Set Up POS**: Enable billing interface and payment options (cash, card, UPI).
5. **Start Selling**: Use POS to bill customers and issue receipts.
6. **Track Inventory**: Monitor stock levels and set low-stock alerts.
7. **View Reports**: Check daily/weekly sales and top-selling items.
8. **Manage Staff**: Add employees and assign roles (optional).
9. **Backup Data**: Regularly save data to cloud or local storage.

# 7. Component Documentation

Key Components:

 **ProductList**: Displays products with filters/search.

 **ProductDetail**: Shows individual product info.

 **InventoryManager**: Tracks stock levels and updates.

 **Cart**: Manages selected items and totals.

 **Checkout**: Handles billing and payment.

 **UserAuth**: Login, register, role-based access.

 **AdminDashboard**: Analytics, settings, user control.

Reusable components:

* **Button**: Standard clickable UI.
* **InputField**: Text input with validation.
* **Modal**: Pop-up for alerts/forms.
* **Loader**: Spinner for async tasks.
* **Navbar**: Top navigation bar.
* **Sidebar**: Menu for admin/user.
* **Card**: Display product/info blocks.
* **Toast**: Notification messages.

# 8. State Management

 **Local State**

* Used for component-specific logic.
* Managed with useState, useReducer, etc.
* Examples: modal visibility, form inputs, button loading.

 **Global State**

* Shared across multiple components.
* Stores app-wide data like cart, user, inventory.
* Ensures consistency and avoids prop drilling.

# 9 User Interface

Dashbard view of inventory with graphical representation.

Interactive tables for product listing and stock details.

User-friendly controls for adding,updating,and deleting stock items.

# 10. Styling

**🧱 Layout**

* **Card view** for mobile
* **Table view** for tablet/desktop
* Use collapsible sections for details
* **🎨 Colors**
* Background: #FFFFFF
* Text: #333333
* Button: #007BFF (blue)
* Low stock: #FFA500 (orange)
* Out of stock: #FF4C4C (red)
* **🔠 Font**
* Font: Roboto or Open Sans
* Size: 14px (labels), 12px (table text)
* **📋 Product Card Example**
* plaintext
* Apple Juice
* SKU: AJ001 | ₹50 | Stock: 25
* [Edit] [Delete]
* **📊 Table Row Example**
* plaintext
* | Apple Juice | AJ001 | ₹50 | 25 | ✏️ 🗑️ |
* **📱 Mobile Features**
* Sticky “Add Product” button
* Swipe for edit/delete
* Responsive layout

# 11. Testing

* **Testing Strategy:**

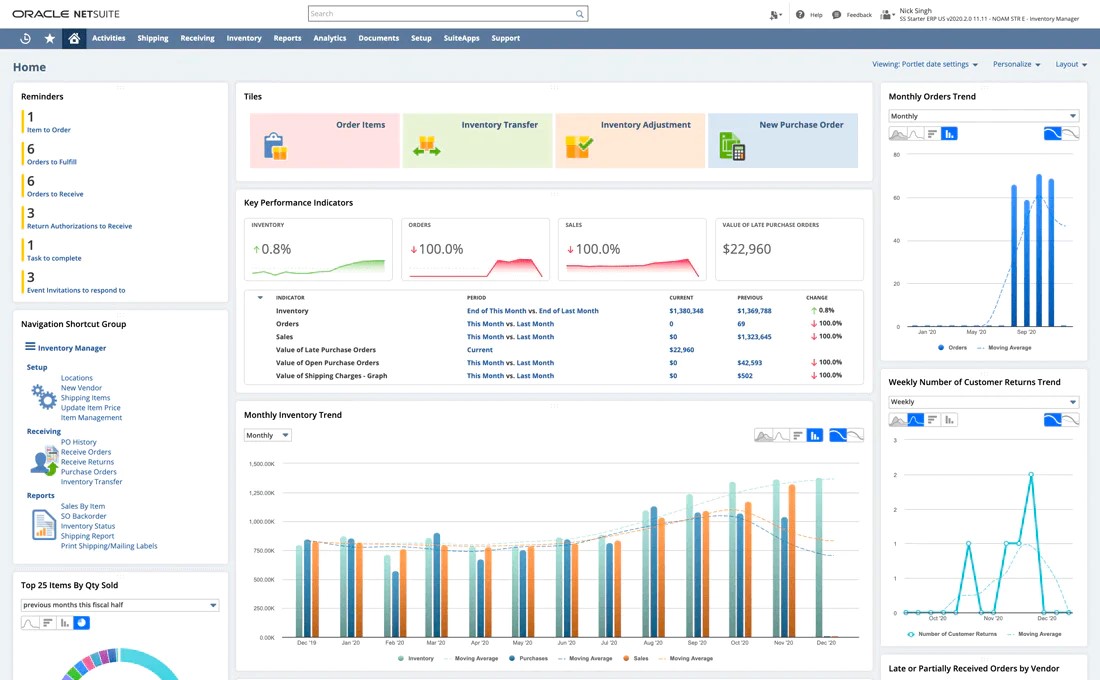
Unit test for individual components.

Integration test for database and backend APIs.

User acceptances testing to validate system performance

**12.Screenshots or Demo**

**screenhots:**

****

****

****

**13. Project Link Known Issues**

Link**:**  https://github.com/nivetha2007w/-Store-manager-Keep-track-of-inventory-.git

14. **Known Issues**

Limited multilingual support.

Mobile view may require optimization for smaller screens.

Reports generation speed may vary with large datasets

# 15. Future Enhancements

1. Add multi-store support to manage inventory across branches

2.Implement role-based access for secure user permissions

3.Enable offline mode with sync-on-connect functionality

4.Integrate advanced reporting for sales and stock analytics

5.Include barcode scanning to speed up checkout and inventory updates