Project Documentation

STORE MANAGER: KEEP TRACK OF INVENTORY

1.Introduction

Project Title: Inventory Management

Team Members:

1. SRINIVASAN.J(TEAM LEADER)

2. PRATHISH.R (SCRIPT WRITING)

3. RACHIN.K (DOCUMENTATION)

4. RAGASANDHIYA.V (VOICE OVER)

5. POOJA.V (DOCUMENTATION)

2.Project overview

• Purpose: To provide a user-friendly interface for store managers to efficiently track,

manage, and update product inventory. The system allows for viewing stock levels,

• Features:

 Inventory Dashboard: Displays all products in a card-based layout, showing the product image, name, price, and available stock.

 Stock Updates: Functionality to add stock to any product and update the count in real-time.

 Search and Filter: Users can search for specific items and filter the view to show only depleted products or items below a certain alert value.

 Navigation: Includes navigation for "Home", "Cart", "Inventory", "Sales", and "Add Product" pages.

3. Architecture

• Component Structure: The application is structured with major components such

as Navbar, inventory Page, and Product Card. The Inventory Page would fetch and display

a list of products, rendering a Product Card for each one.

● State Management: A state management approach like react Context API or

Use State/use Reducer hooks is used to manage the product list and search/filter

criteria.

● Routing: The navigation links suggest that a library like react-router - dom is used to

handle client-side routing between different pages (/inventory, /cart, /sales, etc.).

4.Setup instructions

● Prerequisites: Node.js and npm (or yarn).

● Installation:

1. Clone the repository: git clone <your-repo-url>

2. Navigate to the client directory: cd client

3. Install dependencies: npm install

4. npm start

5. Folder Structure

● Client: The React application is organized with folders for:

○ components: Contains reusable UI elements like Product Card, Button, Input.

○ pages: Contains top-level components for each route, such as InventoryPage.js,

SalesPage.js.

○ assets: For static files like images and stylesheets.

6. Running the Application

● To start the frontend development server, run the following command in the client

directory: npm start.

7. API Documentation

● Key Components:

○ Inventory Page: Fetches and displays the list of all inventory items. Manages the

state for search and filter inputs.

○ Product Card: A reusable component that displays details for a single product

(image, name, price, stock). It receives product data as props and handles the local

state for the "Add Stock" input field.

8. Authentication

● CSS Frameworks/Libraries: Based on the visual style, the project uses a CSS

framework like Bootstrap or Material-UI for the layout and components, or custom CSS for

styling, this project doesn’t have authentication.

9. Screenshots or Demo

●demo videos of this project is created and submitted in git hub link.

10.Testing

• Manually tested in visual studio code terminal.

11. Known Issues

• No issues.

12. Future Enhancements

● Potential future improvements could include a sales analytics dashboard, user

authentication for different roles (e.g., manager vs. employee), or barcode scanning for

faster stock updates.