

Project Documentation: Store Manager (Inventory Tracking System)

1. Introduction

Project Title: Store Manager – INSIGHT STREAM:Navigate the news landscape

Team Members:

TEAM ID: NM2025TMID28746

TEAM LEADER:RAMYA.M & ramyamurugesan185@gmail.com

TEAM MEMBER:PRIYADHARSHINI.S & priyadharshinibsc24@gmail.com

TEAM MEMBER:DEEPA.V & deepadd12007@gmail.com

TEAM MEMBER:DURGA DEVI.V & ddurgadevi928@gmail.com

2. Project Overview

Purpose:

The purpose of this project is to help store managers efficiently keep track of inventory, stock levels, and sales data.

Features:

Add, update, and delete inventory items

View stock levels in real-time

Categorize items by type, supplier, or availability

Generate sales and inventory reports

Low-stock alerts and notifications

3. Architecture

Component Structure:

Dashboard: Displays stock overview and reports

Inventory List: Shows all products with details

Product Form: For adding/updating inventory items

Reports: Sales and stock reports visualization

Notifications: Low-stock and expiry alerts

State Management:

Global State: Managed with Redux to handle inventory list, authentication, and notifications.

Local State: Used in forms for temporary inputs before submission.

Routing:

/dashboard – Inventory overview

/inventory – Inventory list

/add-item – Add new inventory item

/reports – Sales & stock reports

4. Setup Instructions

Prerequisites: Node.js, npm, React, Redux, React Router

Installation:

```
git clone https://github.com/your-repo/store-manager.git
```

```
cd store-manager
```

```
npm install
```

```
npm start
```

5. Folder Structure

Client:

src/

components/

Dashboard/

Inventory/

ProductForm/

Reports/

redux/

assets/

utils/

App.js

Utilities: Helper functions for calculations (e.g., total stock value), API services, and validation functions.

6. Running the Application

npm start

Runs the app in development mode at <http://localhost:3000>.

7. Component Documentation

Key Components:

Dashboard: Displays quick stats (total items, low stock, etc.)

InventoryList: Shows products with filters & search

ProductForm: Add or edit product details

Reusable Components:

Button, Card, Modal, Notification

8. State Management

Global State:

Inventory items, authentication status, notifications

Local State:

Form field inputs, temporary filter/search states

9. User Interface

Examples:

Dashboard showing stock count

Inventory table with edit/delete options

Report charts

10. Styling

CSS Frameworks/Libraries: Tailwind CSS, Styled Components

Theming: Dark mode and light mode toggle

11. Testing

Testing Strategy:

Unit tests with Jest & React Testing Library

Integration tests for Redux state

Code Coverage: Jest + Istanbul coverage reports

12. Screenshots or Demo

Inventory Dashboard screenshot

Product Form screenshot

Reports with charts

13. Known Issues

API latency may cause delayed updates on reports

Exporting reports to CSV/PDF under development
