

INTRODUCTION:

Project Title: Store Manager : keep Track Of Inventory

Team Members:

1. MALLIKA. M

2. MYTHILI. A

3. MEGALA.S

4. MAHALAXMI.S

5. MARIYAM BEE.S

Team Members Roles:

1.MALLIKA.M- Coding

2.MYTHILI.A - Coding Development

3.MAGALA.S - Document

4.MAHALAXMI.S – Document

5.MARIYAM BEE- Demo Video

Store Manager – Inventory Tracking Frontend

Project Overview

Purpose:

- The application helps store managers efficiently manage inventory by tracking stock levels, product details, and updates in real time.

Goals:

- Provide a user-friendly dashboard for viewing and managing inventory.
- Enable CRUD operations (Create, Read, Update, Delete) for products.
- Allow filtering, sorting, and searching for products.
- Provide alerts for low stock and out-of-stock items.

Features:

Dashboard with product list and quick stats.

Add, edit, and delete products.

Search and filter inventory.

Low-stock notifications.

Responsive design for desktop and mobile.

Architecture

Component Structure

App – Root component handling routing and global state.

Dashboard – Displays key inventory stats and product list.

ProductList – Renders table/grid of products with actions.

ProductForm – Used for adding/editing product details.

Header / Sidebar – Navigation and branding.

NotificationBar – Shows low-stock alerts.

Reusable UI Components – Button, Modal, Input, Dropdown, etc.

State Management

Global State: Managed with Context API (or Redux if needed) for product data and user preferences.

Local State: Managed via useState inside individual components for form inputs, modal open/close states, etc.

Routing

Using react-router-dom:

/ → Dashboard

/products → Product List

/products/add → Add Product

/products/edit/:id → Edit Product

Setup Instructions

Prerequisites

Node.js ($\geq 18.x$)

npm or yarn

Installation

Clone the repository

```
git clone
```

Navigate to client folder

```
cd store-manager-frontend
```

Install dependencies

```
npm install
```

Configure environment variables

Example: .env file

```
REACT_APP_API_URL=http://localhost:5000
```

```
/api
```

Folder Structure

store-manager-frontend/

|— public/ # Static assets

|— src/

| |— assets/ # Images, icons,

fonts

| └─ components/ # Reusable ui

components

| └─ pages/ # Page-level

components (Dashboard, ProductList, etc.)

| └─ context/ # Context API

providers

| └─ hooks/ # Custom hooks

(e.g., useFetch, useForm)

| └─ utils/ # Helper functions

| └─ App.js # Root component

| └─ index.js # Entry point

Running the Application

Start development server

npm start

Runs on <http://localhost:3000>

Component Documentation

Key Components

ProductList: Displays inventory table, props → products, onEdit, onDelete.

ProductForm: Handles adding/editing, props → initialValues, onSubmit.

NotificationBar: Displays stock alerts, props → alerts

Reusable Components

Button – Variants: primary, secondary, danger.

Modal – Used for confirmation dialogs and forms.

InputField – Handles validation and error messages.

State Management

Global State

Products Context: Manages all product data fetched from API.

UI Context: Handles theme and sidebar toggle.

Local State

Form inputs (useState) inside ProductForm.

Modal open/close state in Dashboard and ProductList.

User Interface

Dashboard – Total items, low stock count, quick links.

Product Table – Sort, search, filter by category.

Forms – Add/Edit product with validation

Styling

Frameworks: Tailwind CSS + custom styles.

Theming: Light/Dark mode toggle.

Testing

Testing Strategy

Unit Tests – Component logic (Jest + React Testing Library).

Integration Tests – API calls and context behavior.

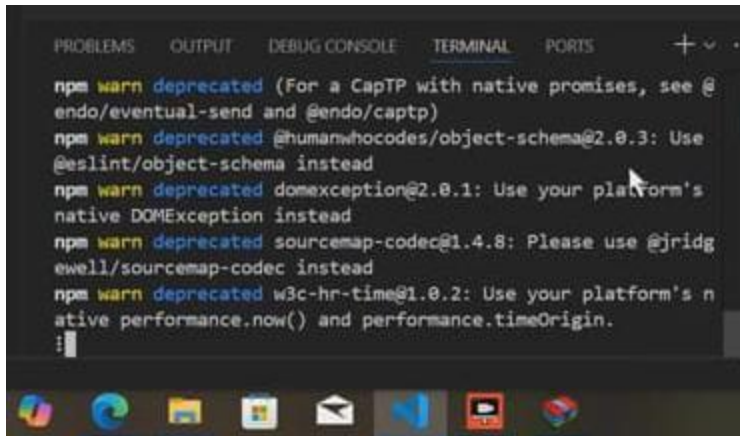
E2E Tests – User flows (Cypress).

Code Coverage

Enforced via Jest coverage reports.

Screenshots or Demo

Demo link

A screenshot of a Visual Studio Code terminal window. The terminal shows several npm deprecation warnings. The warnings are: 1. 'npm warn deprecated (For a CapTP with native promises, see @endo/eventual-send and @endo/captp)' 2. 'npm warn deprecated @humanwhocodes/object-schema@2.0.3: Use @eslint/object-schema instead' 3. 'npm warn deprecated domexception@2.0.1: Use your platform's native DOMException instead' 4. 'npm warn deprecated sourcemap-codec@1.4.8: Please use @jridgeway/sourcemap-codec instead' 5. 'npm warn deprecated w3c-hr-time@1.0.2: Use your platform's native performance.now() and performance.timeOrigin.' The terminal is dark-themed with light blue and yellow text. The VS Code interface is visible in the background, including the 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS' tabs at the top. The Windows taskbar is visible at the bottom with icons for Edge, File Explorer, Mail, and other applications.

```
npm warn deprecated (For a CapTP with native promises, see @
endo/eventual-send and @endo/captp)
npm warn deprecated @humanwhocodes/object-schema@2.0.3: Use
@eslint/object-schema instead
npm warn deprecated domexception@2.0.1: Use your platform's
native DOMException instead
npm warn deprecated sourcemap-codec@1.4.8: Please use @jridg
ewell/sourcemap-codec instead
npm warn deprecated w3c-hr-time@1.0.2: Use your platform's n
ative performance.now() and performance.timeOrigin.
:
```

Known Issues

Pagination not implemented yet.

Low-stock alerts only update on page reload.

Future Enhancements

Barcode scanner integration.

Export inventory data (CSV, Excel, PDF).

Role-based authentication (Admin, Staff).

Push notifications for stock alerts.

