1. Introduction

- Project Title: Store Manager : keep track of inventory

  Team Id:NM2025TMID42660

- Team size:5

  - Team Leader:Puja S (Documentation)
  - Email Id:pujasaravanan2513@gmail.com
  - Team member:Dharshini V (Coding)
  - Email Id:venkatpriya2430@gmail.com
  - Team member:Subashini G C (Demo video)
  - Email Id:subashinichandran79@gmail.com
  - Team member:Naviya K (Coding)
  - Email Id:navyaknavya2006@gmail.com
  - Team member:Sathya R (Documentaion)
  - Email Id:sathyasathya57074@gmail.com

____

2. Project Overview

- Purpose:

The Store Manager application aims to simplify the day-to-day management of store operations for Naan Mudhalvan. It enables store managers to track inventory, sales, employee data, and customer interactions in real-time, with a focus on improving efficiency and decision-making.

- Features:

o Dashboard: Overview of store performance, recent sales, stock levels, etc.

o Inventory Management: Add, update, and remove products; track stock levels.

o Sales Tracking: Real-time sales data, detailed reports, and trends.

o Employee Management: Track working hours, performance, and payroll.

o Customer Management: Store customer details, transaction history, and preferences.

____

3. Architecture

- Component Structure:

o App: The root component that initializes the application and routing.

o Dashboard: Displays key metrics and visualizations like sales trends, stock levels, etc.

o Inventory: Manages product inventory, including CRUD operations.

o Sales: Displays daily, weekly, and monthly sales reports.

o Employees: Allows for managing employee data like shifts, attendance, and performance.

o Customers: Manages customer information and order history.

• State Management:

We are using Redux for global state management to ensure the consistent flow of information across the application. For local component state, React's useState is employed.

• Routing:

React Router is used to handle navigation:

o /dashboard - Main dashboard with key metrics.

o /inventory - Inventory management page.

o /sales - Sales tracking and reports.

o /employees - Employee management page.

o /customers - Customer management and details.

____

4. Setup Instructions

• Prerequisites:

o Node.js (>= 14.x)

o npm (>= 7.x)

• Installation:

1. Clone the repository:
2. git clone https://github.com/naan-mudhalvan/store-manager.git
3. Navigate to the project directory:
4. cd store-manager
5. Install dependencies:
6. npm install
7. Set up environment variables (create a .env file in the root directory and add the necessary keys):
8. REACT_APP_API_URL=https://your-api-url.com
9. REACT_APP_AUTH_SECRET=your-auth-secret

____

5. Folder Structure

• Client:

o src/

 components/: Reusable components like buttons, tables, modals.

	pages/: Pages representing different routes: Dashboard, Inventory, Sales, Employees, Customers.

	assets/: Static assets like logos, icons, etc.

	utils/: Helper functions, custom hooks, and utility classes (e.g., useFetchData.js, useFormValidation.js).

	services/: API services for CRUD operations (e.g., inventoryService.js, salesService.js).

- Utilities:

o useAuth.js: Custom hook for handling user authentication.

o useInventory.js: Custom hook for managing inventory state.

o useSales.js: Custom hook for managing sales data.

____

6. Running the Application

- Frontend:

To start the frontend locally:

- cd client
- npm start

This will start the React development server at http://localhost:3000.

____

7. Component Documentation

- Key Components:

o Dashboard:

	Purpose: Displays high-level metrics such as total sales, stock levels, and employee performance.

	Props:

	salesData: An object with sales statistics (total, daily, monthly).

	inventoryData: An array of inventory data (product name, quantity, price).

o Inventory:

	Purpose: Allows the manager to add, remove, or update products in the inventory.

	Props:

	inventoryList: A list of products in the store.

	onAddProduct: Callback function for adding a new product.

	onRemoveProduct: Callback function for removing a product.

•	Reusable Components:

o	Button:

□	Props:

□	label: The text shown on the button.

□	onClick: Function to execute on button click.

o	Table:

□	Props:

□	columns: Array of column definitions (e.g., name, price, quantity).

□	data: Data to display in the table (e.g., inventory list, sales data).

____

8. State Management

•	Global State:

The application uses Redux for global state management to handle user authentication, inventory data, sales data, and employee records.

o	Redux Store contains:

□	auth: User authentication state.

□	inventory: Data for products and stock.

□	sales: Sales reports.

□	employees: Employee records.

•	Local State:

Each page component uses React's use State to handle local state for user interactions, such as adding a new product or generating sales reports.

____

9. User Interface

•	Screenshots :

o	Dashboard View: A graph showing sales trends over time, and a list of top-performing products.

o	Inventory Management: A table displaying product names, quantities, and actions to edit or delete.

o	Sales Tracking: Visual representation of daily/weekly/monthly sales.

____

---

10. Testing

- Testing Strategy:

o Unit Tests: Using Jest and React Testing Library to write unit tests for individual components.

o Integration Tests: Testing interaction between components like adding/removing inventory items and viewing updated sales data.

o End-to-End Tests: Cypress is used for full flows such as logging in, viewing inventory, and making a sale.

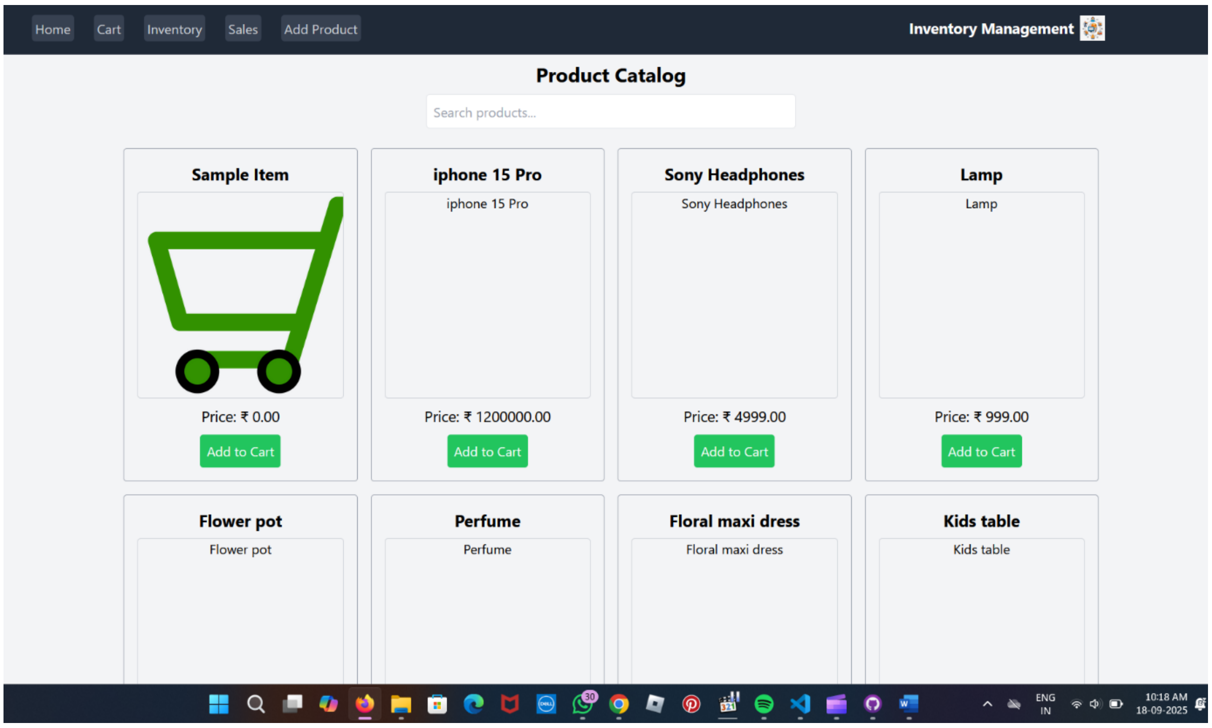
- Code Coverage:

We use Jest’s built-in coverage tool to ensure all components are well-tested. Targeted coverage is 80%.

---

11. Screenshots or Demo

- Screenshots:



Product Catalog

Sample Item
Price: ₹ 0.00

iphone 15 Pro
iphone 15 Pro
Price: ₹ 1200000.00

Sony Headphones
Sony Headphones
Price: ₹ 4999.00

Lamp
Lamp
Price: ₹ 999.00

Flower pot
Flower pot

Perfume
Perfume

Floral maxi dress
Floral maxi dress

Kids table
Kids table

Include images of the dashboard, inventory table, and sales report graphs.

- Demo Link: https://drive.google.com/file/d/1N4GcmzdK_rzD6oI6WQrYAIrMYmxFgfWT/view?usp=drive_link

You can share a live demo link to showcase the project:

Live Demo.

---

12. Known Issues

- Bug 1: The inventory update sometimes fails to reflect in the dashboard in real-time.
- Bug 2: Mobile view of the dashboard could use some responsive adjustments.

---

13. Future Enhancements

- Product Recommendations: Implement AI-based recommendations for store managers to restock high-demand items.
- Multi-store Support: Allow store managers to manage multiple stores from a single account.
- Analytics: Advanced analytics for sales forecasting and product performance.