* Project Title: Inventory Management
* Team ID: NM2025TMID39840
* Team Leader: SREENATHI B & [16.10.2005.sreenathi@gmail.com](mailto:16.10.2005.sreenathi@gmail.com)
* Team Members: YAMINI S & [yamu061026@gmail.com](mailto:yamu061026@gmail.com)
* Team Members: VENKATASHREE S & [venkatcharancharan85@gmail.com](mailto:venkatcharancharan85@gmail.com)
* Team Members: PRIYADHARSHINI M C & [priyamugil06@gmail.com](mailto:priyamugil06@gmail.com)

Inventory Management

1.Introduction

Inventory management is the process of efficiently supervising and controlling the ordering, storage, and use of goods within an organization. It ensures that the right products are available in the right quantity, at the right place, and at the right time, while minimizing costs and avoiding shortages or overstocking.

It plays a crucial role in supply chain management since inventory represents one of the largest investments for businesses, especially in retail, manufacturing, and distribution. Effective inventory management balances customer demand with inventory costs, helping companies improve cash flow, reduce waste, and increase overall profitability.

2.PROJECT OVERVIEW

* PURPOSE: Inventory management is tracking stock to have the right products at the right time.
* FEATURES:
* **Inventory Management:** Inventory Management helps maintain healthy stock levels in a store and acquire them in time.
* **Stock Updates:** Stock will automatically update on sale of products, and it can be updated on adding new stock.
* **Cart:** Products can be added to cart for a particular sale and quantity can be added to each product.
* **Checkout at Cart**: Upon checkout, cart is cleared, inventory is updated, and a sale record is made.
* **Adding New Products to Inventory:** New products can be added to the inventory by providing product name, image URL, price, stock, tags.
* **Alert View for Depleting Stock:** Depleting stocks are shown in red background, and alert count can be updated as per requirement.
* **Search Functionality for Products:** Products in inventory and product catalog can be searched.

**3.ARCHITECTURE**

* **FRONTEND:**

**In the frontend architecture of an inventory management system, the user interface (UI) and user experience (UX) are central. The frontend is designed to allow users (e.g., managers, staff, or customers) to interact with the inventory system through features such as viewing stock levels, adding or removing items, and generating reports. The frontend often communicates with the backend (which handles the data processing and storage) via APIs or other methods.**

* **BACKEND:**

**Inventory management architecture in the backend typically involves designing a system to efficiently track, manage, and update inventory levels, movements, and related data. Here's a high-level overview of key components and considerations for building an inventory management architecture in the backend.**

* **DATABASE:**

1. **Items/Products Table: Stores details about each item/product like ID, name, description, SKU, etc.**
2. **Inventory Table: Tracks quantities of items at different locations or warehouses.**
3. **Transactions Table: Logs changes to inventory like additions, subtractions due to sales, purchases, adjustments.**
4. **4. Locations/Warehouses Table: Details about different storage locations.**

**4. Setup Instructions**

* **Prerequisites**

Here are the key prerequisites for developing a frontend application using React.js:

* **Node.js and npm**:

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the local environment. It provides a scalable and efficient platform for building network applications.

Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.

* Download: <https://nodejs.org/en/download/>
* Installation instructions: <https://nodejs.org/en/download/package-manager/>
* **React.js**:

React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.

Install React.js, a JavaScript library for building user interfaces.

* Create a new React app:

npx create-react-app my-react-app

Replace my-react-app with your preferred project name.

* Navigate to the project directory:

cd my-react-app

* Running the React App:

With the React app created, you can now start the development server and see your React application in action.

* Start the development server:

npm start

This command launches the development server, and you can access your React app at [http://localhost:3000](about:blank) in your web browser.

* **Version Control**: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.
* **HTML, CSS, and JavaScript**: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.
* INSTALLATION STEPS:

Depends on the specific inventory management system or software. For a custom-built system using a tech stack like Node.js, Express, and MongoDB:

1. Setup Database: Install MongoDB or chosen DB.

2. Install Dependencies: Use npm/yarn to install required packages (e.g., Express, Mongoose).

3. Configure Connection: Connect backend to database.

4. Define Schema/Models: Set up schemas for items, inventory, transactions.

5. Implement APIs: Create endpoints for CRUD operations on inventory.

6. Test: Test APIs for inventory management operations.

7. Deploy: Deploy to server or cloud.

For off-the-shelf inventory management software, steps typically involve:

1. Download/Install Software: Follow vendor instructions.

2. Configure Settings: Setup users, items, locations.

3. Integrate if needed: Connect with other systems like sales or accounting.

5.FOLDER STRUCTURE

A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

6.RUNNING THE APPLICATION

* Frontend:

cd inventory management

npm install

* Backend:

Cd server npm start

* Access:

Visit <http://localhost:3000>

7.API DOCUMENTATION

* Users:

- /api/users/register

- /api/users/login

- /api/users/{id}

* Projects:

- /api/projects

- /api/projects/{id}

* Application:

- /api/items

*- /api/items/{id}*

- /api/items/{id}

* Chats:

-/api/chats

- /api/chats/{conversation}

- /api/chats/unread

- /api/chats/{id}

8.AUTHENTICATION

* To Project Sensitive Data:
* Authentication in inventory management protects sensitive data like inventory levels, item details, and transaction history by ensuring only authorized personnel can access or modify this data. Using mechanisms like role-based access control and secure tokens (e.g., JWT), the system safeguards against unauthorized access, data breaches, or unintended modifications, maintaining data integrity and security.
* To Prevent Unauthorized Access:
* Authentication in inventory management prevents unauthorized access by verifying user identities and restricting actions based on user roles. This ensures only permitted users can view, edit, or manage inventory data, reducing risks of data tampering, theft, or accidental changes. By implementing authentication mechanisms like login credentials and tokens, inventory systems protect against unauthorized entry and maintain data security.
* To Track User Actions:
* Inventory management systems can track user actions (like adds, updates, deletes) via logging or audit trails. This ensures accountability, aids troubleshooting, and supports compliance. By recording user actions with details like user ID, action type, and timestamp, systems maintain a history of changes to inventory dat

9.USER INTERFACE

* LOGIN:
* The login UI in an inventory management system authenticates users to secure access to inventory data. Typically including username and password fields, the login ensures only authorized users can view or manage inventory based on their roles. Post-login, users access inventory features tailored to their permissions.
* DASHBOARD:
* The dashboard in an inventory management system's UI gives users an overview of inventory status, like current stock levels, low stock alerts, and recent activity. It often includes shortcuts to common actions like adding items, generating reports, or viewing details. The dashboard's layout is typically designed for quick access to key inventory info and actions based on user permissions
* ORDERS:
* In the orders section of an inventory management system's UI, users can view, create, or manage orders (like purchase orders or sales orders). The interface typically shows order details like items ordered, quantities, statuses, and dates. Users can often filter orders, mark orders as fulfilled, or trigger inventory updates based on order actions.
* SUPPLIER:
* In the orders section of an inventory management system, users can view orders linked to suppliers, like purchase orders. The UI typically shows supplier details alongside order info like items, quantities, and statuses. Users can manage these orders, track supplier deliveries, and update inventory based on order fulfillment.
* SEARCH:
* Search means looking for specific information or products within a system or database.
* NOTIFICATION:
* A notification is a message or alert that informs you about an event or update.

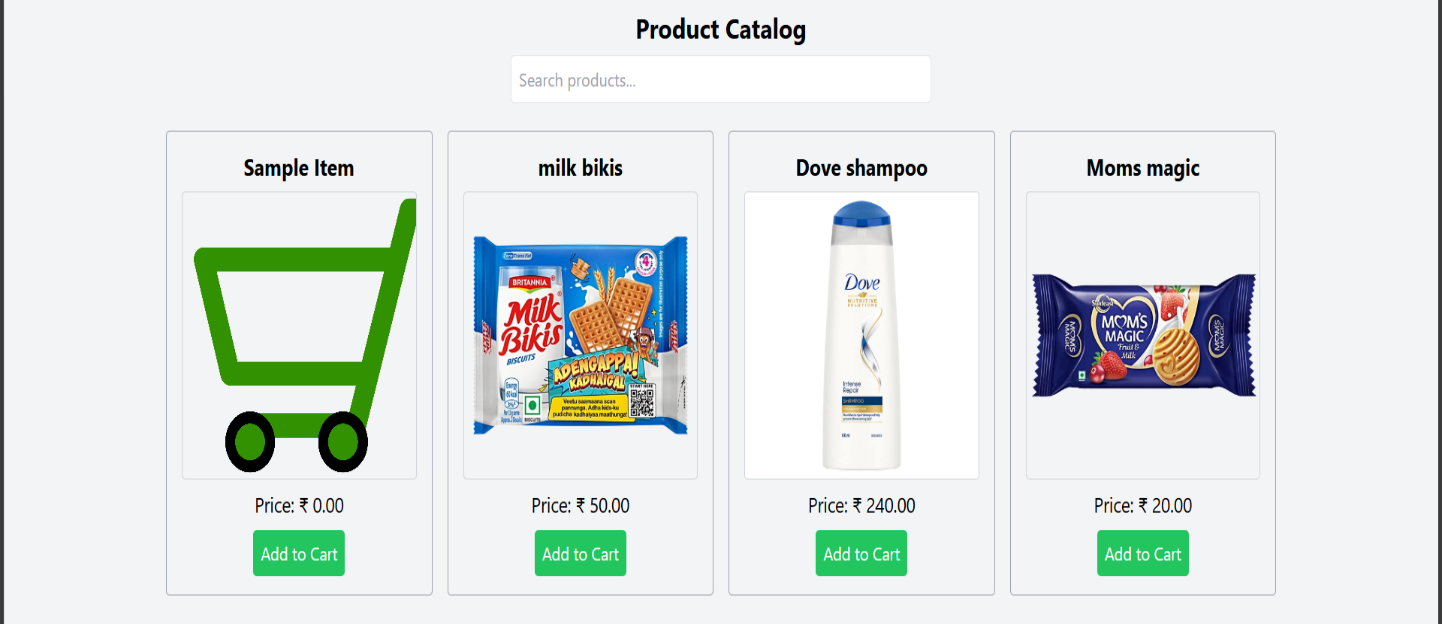
10.TESTING

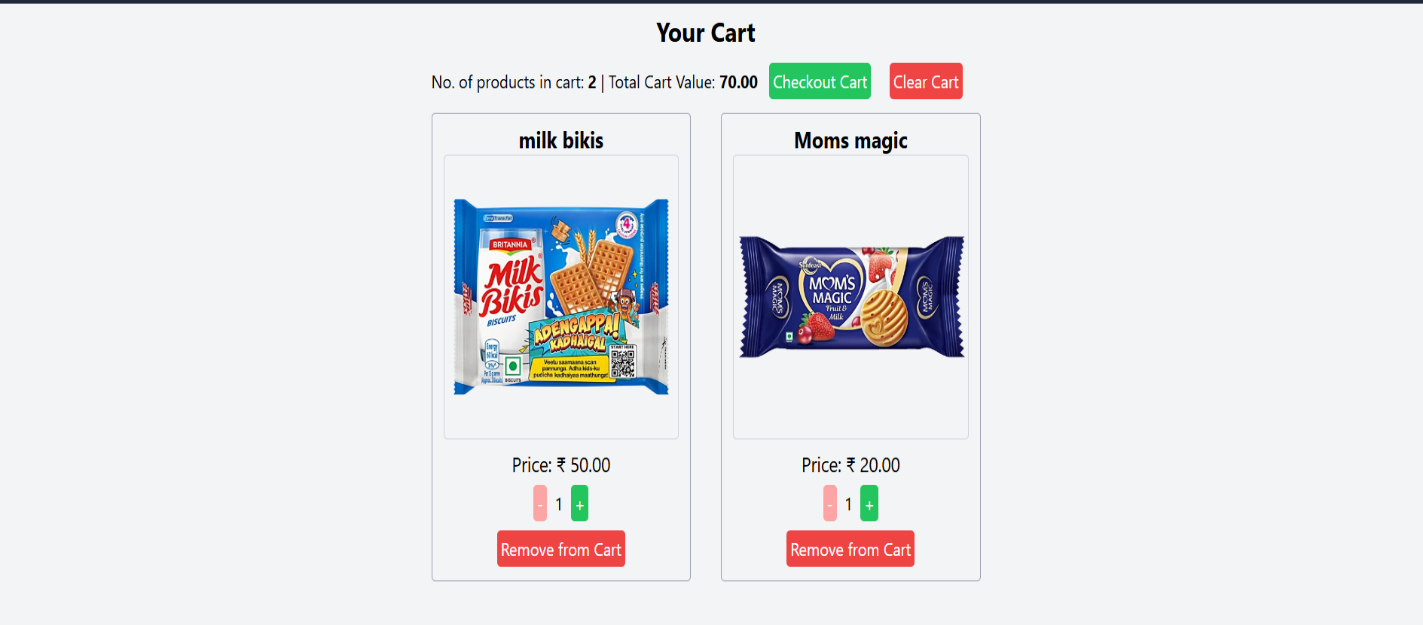
* System is accurate, secure and easy to use before running.
* In inventory management, testing refers to the process of verifying and validating inventory-related processes, system, or products.

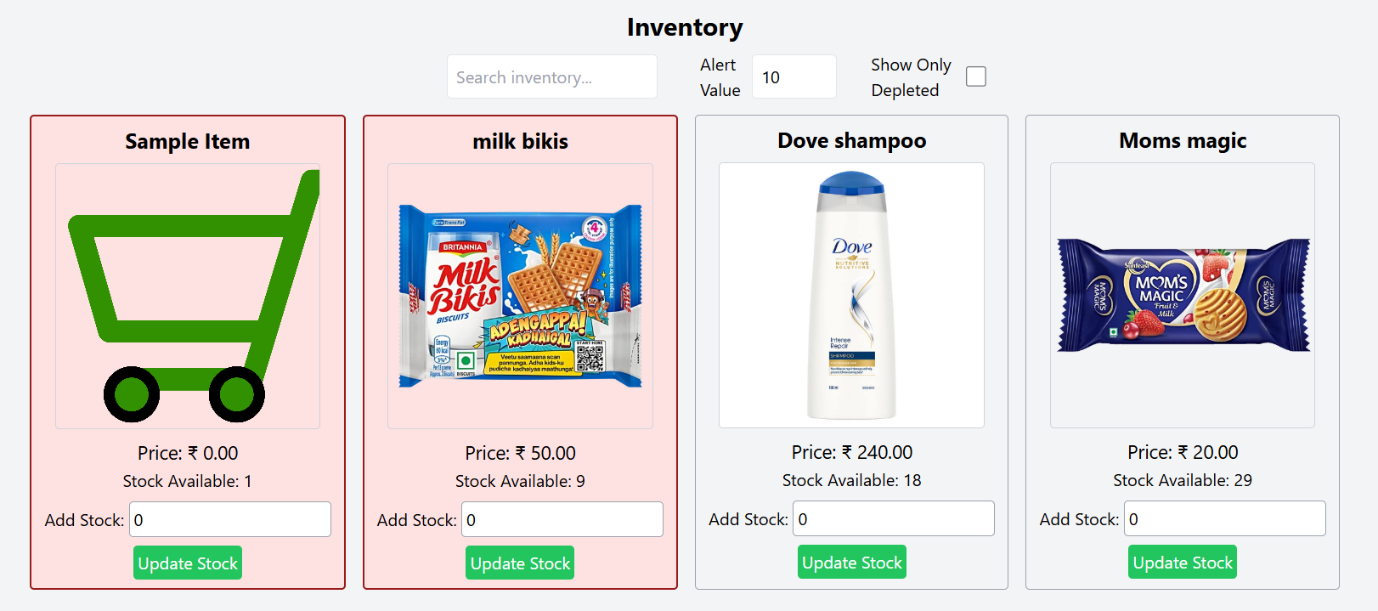
This can include:

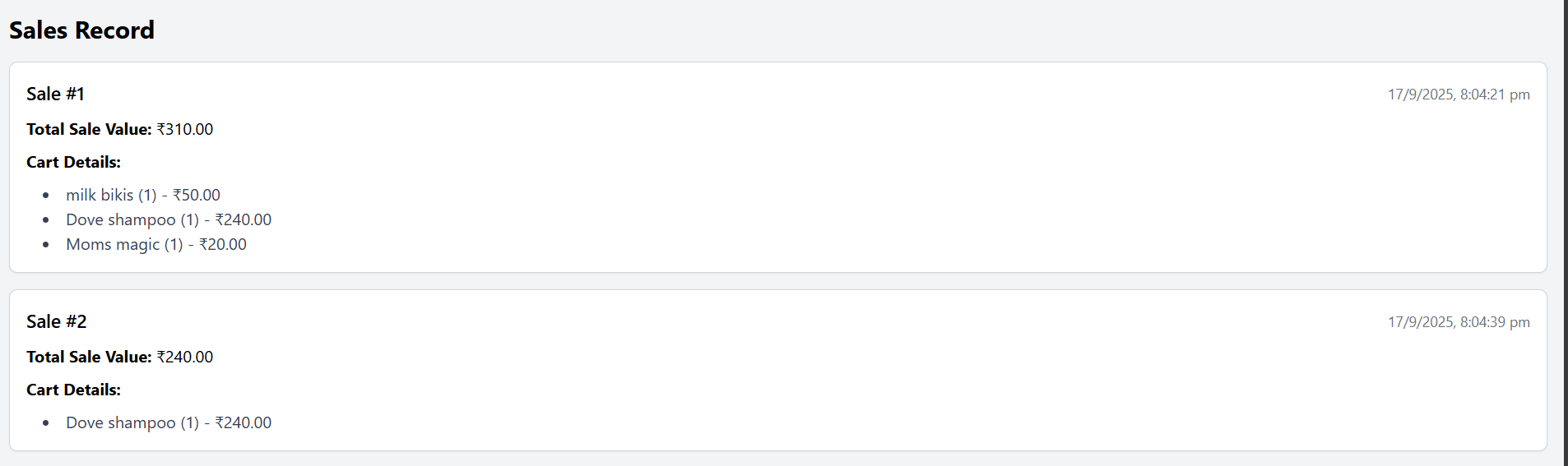
* Inventory audits: Verifying the accuracy of inventory records.
* System testing: Testing inventory management software or systems.
* Product testing: Inspecting products for quality and defects.
* Testing helps ensure that inventory management processes are accurate, efficient, and reliable.
* Tools: Excel, Jira, Testrail.

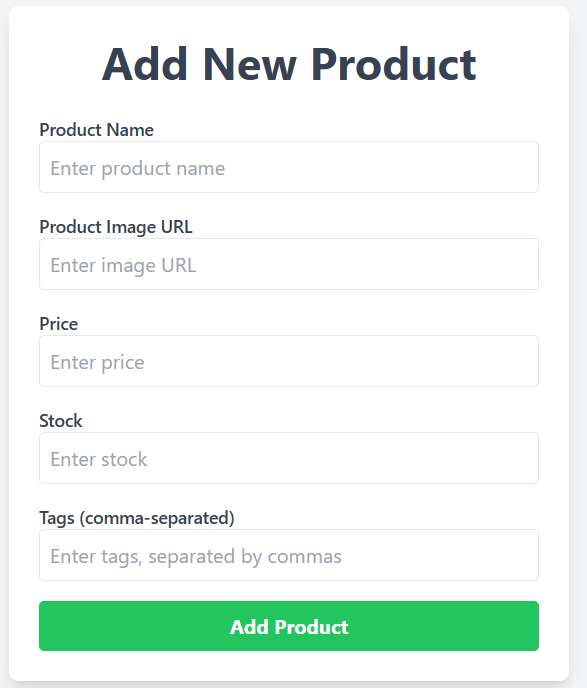
11.SCREENSHOTS

Home:

Cart:

Inventory:

Sales Record:

Add Products:

12.KNOWN ISSUES

1. Stockouts: Running out of stock due to inaccurate forecasting or supply chain.

2. Overstocking: Excess inventory leading to unnecessary holding costs and potential waste.

3. Inaccurate inventory records: Discrepancies between actual and recorded inventory levels.

4. Inventory shrinkage: Loss of inventory due to theft, damage, or errors.

5. Poor supplier management: Delays, quality issues, or unreliable suppliers.

6. Inefficient inventory tracking: Manual errors, lack of visibility, or inadequate tracking system.

7. Demand forecasting challenges: Accurately predicting demand due to seasonality, trends, or market fluctuations.

13.FUTURE ENHANCEMENT

Key trends:

* Artificial Intelligence (AI) and Machine Learning: AI and machine learning are revolutionizing inventory management by enabling predictive analytics for demand forecasting, optimizing stock levels, and automating routine tasks ¹ ² ³.
* Internet of Things (IoT) Integration: IoT devices provide real-time tracking and monitoring of inventory, reducing errors and enhancing supply chain visibility ¹ ⁴.
* Cloud-Based Solutions: Cloud computing makes inventory management more accessible, scalable, and collaborative, allowing real-time access to inventory data from anywhere ¹ ².
* Automation and Robotics: Warehouse automation using robotics and AI-driven software improves efficiency, reduces human error, and enhances accuracy in inventory management ⁴ ⁵.
* Sustainability Focus: Businesses are adopting sustainable practices in inventory management to reduce waste, optimize packaging, and minimize environmental impact ¹ ⁶.
* Omnichannel Inventory Control: Managing inventory across multiple sales channels ensures seamless operations and improved customer satisfaction ⁷.
* Predictive Analytics: Advanced analytics help businesses make data-driven decisions for inventory optimization, demand forecasting, and supply chain management ¹ ² ⁵.