

GOVERNMENT ARTS AND SCIENCE COLLEGE, ALANGUDI

Department of computer science

INTERNSHIP REPORT

Store Manager: Keep Track of Inventory

Virtual Internship Program

Organized by
SMART INTERNZ

submitted by :

Team id	NM2025TMID35273
Team size	4
Team leader	MAHESWARI A
Team member	ABITHA M
Team member	ARCHANA K
Team member	BHUVANESHWARI R

➤ INSTALLATION:

- **Node.js and npm:**

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

- **React.js:**

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:
- Start the development server:

```
npm start
```

- **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.
- **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.

- Git: Download and installation instructions can be found at: <https://git-scm.com/downloads>

- **Development Environment:** Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

- Visual Studio Code: Download from <https://code.visualstudio.com/download>

- Sublime Text: Download from <https://www.sublimetext.com/download>

- WebStorm: Download from <https://www.jetbrains.com/webstorm/download>

To get the Application project from drive:

Follow below steps:

Install Dependencies:

- Navigate into the cloned repository directory and install libraries:

```
cd store  
npm install
```

- **Start the Development Server:**

- To start the development server, execute the following command:

```
npm start
```

Access the App:

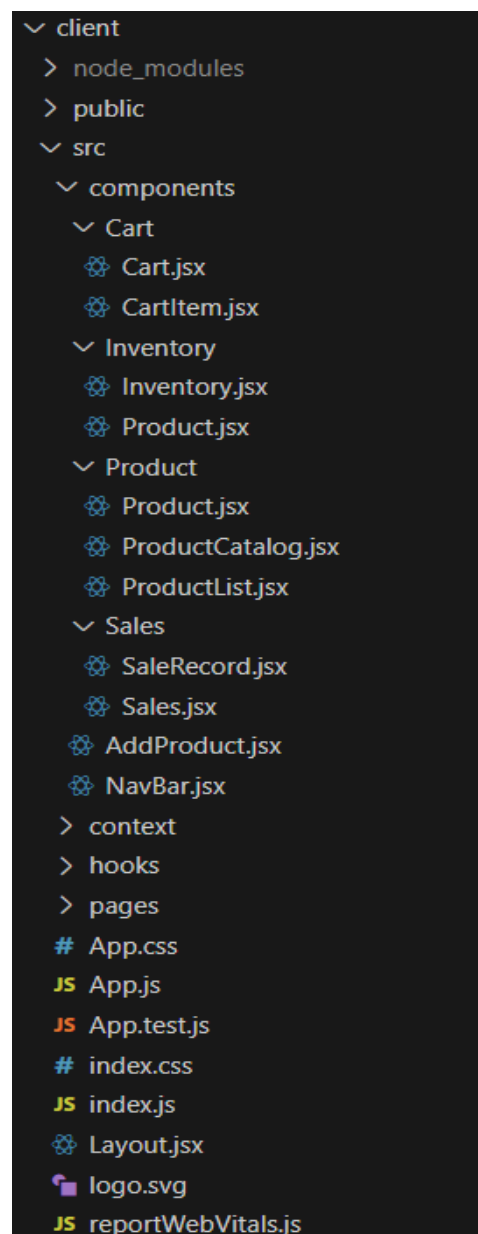
- Open your web browser and navigate to <http://localhost:3000>.

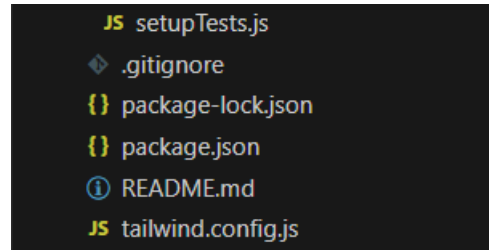
- You should see the application's homepage, indicating that the installation and setup were successful.

You have successfully installed and set up the application on your local machine. You can now proceed with further customization, development, and testing as needed.

➤ Project structure

The image is of the folder structure which shows all the files and folders that have been used in project development.





➤ Project Flow

DemoLink :

https://drive.google.com/file/d/1i_BNxMjY5DM2irPCnzyKCRlwqI_Kn7d/view?usp=sharing

CodeReference :

<https://drive.google.com/drive/folders/1BPYYmXEAEmpDlvCQnz5HDGmhXloYzJ5c?usp=sharing>

➤ Project setup

- **Project Setup**

- **Step 1: Initialize a new React application.**

- Use create-react-app or Vite for project setup.

- **Step 2: Install dependencies.**

- Add tailwindcss, react, react-dom, and other necessary packages.

```
"dependencies": {  
  "cra-template": "1.2.0",  
  "react": "^19.0.0",  
  "react-dom": "^19.0.0",  
  "react-router-dom": "^7.1.1",  
  "react-scripts": "5.0.1",  
  "web-vitals": "^4.2.4"  
},
```

```
"devDependencies": {  
  "tailwindcss": "^3.4.17"  
}
```

- Set up the tailwind.config.js file.
- Add the required styles in index.css.
- For further reference, use the following resources
 - <https://react.dev/learn/installation>
 - <https://react-bootstrap-v4.netlify.app/getting-started/introduction/>
 - <https://axios-http.com/docs/intro>
 - <https://reactrouter.com/en/main/start/tutorial>
 -

➤ Project Development

Setting Up Context and Reducers:

- **Step 1: Create context files for Inventory, Cart, and Sales.**
 - Use createContext() and define default states.
- **Step 2: Define reducers for state management.**
 - Inventory Reducer: Handle stock addition, updates, and sales.
 - Cart Reducer: Manage adding/removing items and quantities.
 - Sales Reducer: Track sales records.
- **Step 3: Set up ContextProvider components.**
 - Wrap the app in these providers in index.js or App.js.
- **Step 4: Persist data in localStorage.**
 - Save and load context states to/from localStorage.

Milestone 3 - Developing Core Components

- **Step 1: Build the Inventory Component.**
 - Display a list of products with stock details.
 - Implement search and alert value input.
- **Step 2: Build the Product Component.**

- Show product details, including an image, price, and stock.
- Add functionality to update stock.
- **Step 3: Build the Cart Component.**
 - Display items added to the cart with quantity and total value.
 - Add buttons for incrementing, decrementing, and removing items.
 - Include checkout functionality to update inventory and create sales records.
- **Step 4: Build the Sales Component.**
 - Display a list of sales records with details like date, cart items, and total sale value.
 - Sort records from the latest to the oldest.

Milestone 4 - Implementing Utilities

- **Step 1: Create reusable helper functions.**
 - Format currency.
 - Sort or filter data.
- **Step 2: Add input validation.**
 - Prevent invalid values for stock or cart quantities.
- **Step 3: Handle edge cases.**
 - Avoid negative stock updates.
 - Restrict checkout when the cart is empty.

Milestone 5 - Styling with TailwindCSS

- **Step 1: Create responsive layouts.**
 - Use flexbox (flex, flex-wrap) and grid utilities.
- **Step 2: Style individual components.**

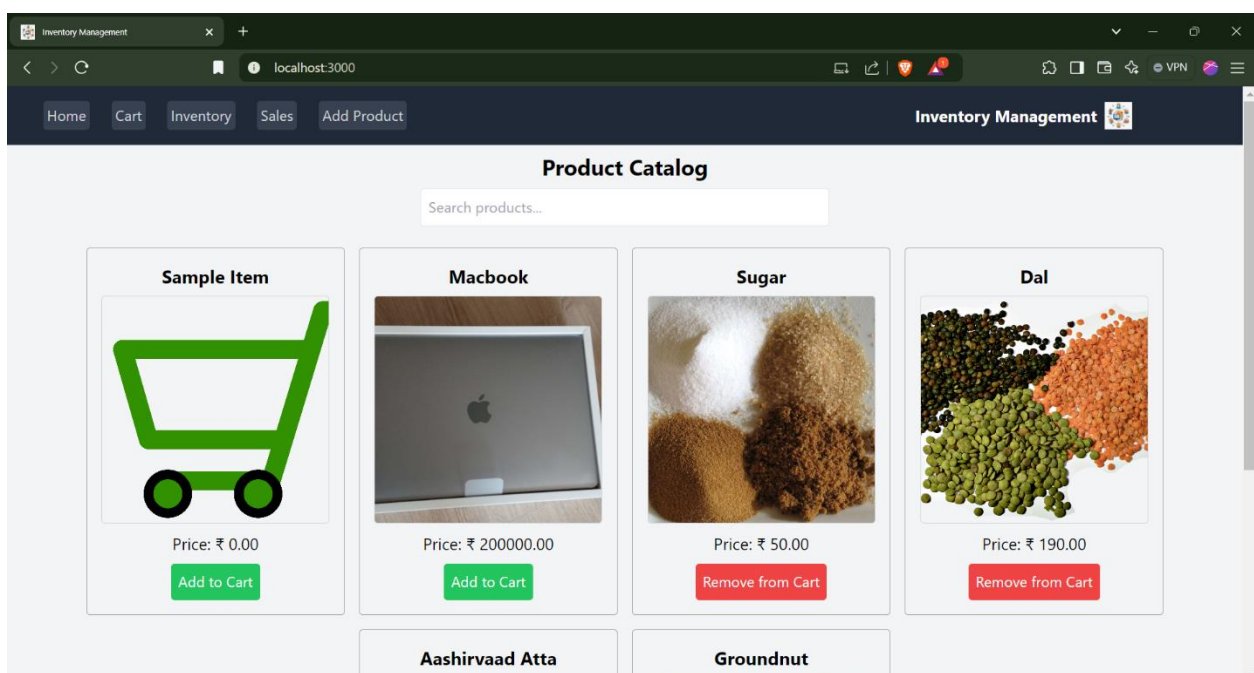
- Inventory: Add hover effects and responsive search bar.
- Cart: Highlight selected products and display alerts on low stock.
- Sales: Use a clean table or card layout for sale records.
- **Step 3: Ensure consistency.**
 - Apply a theme or consistent color palette.

Milestone 6 - Testing and Debugging

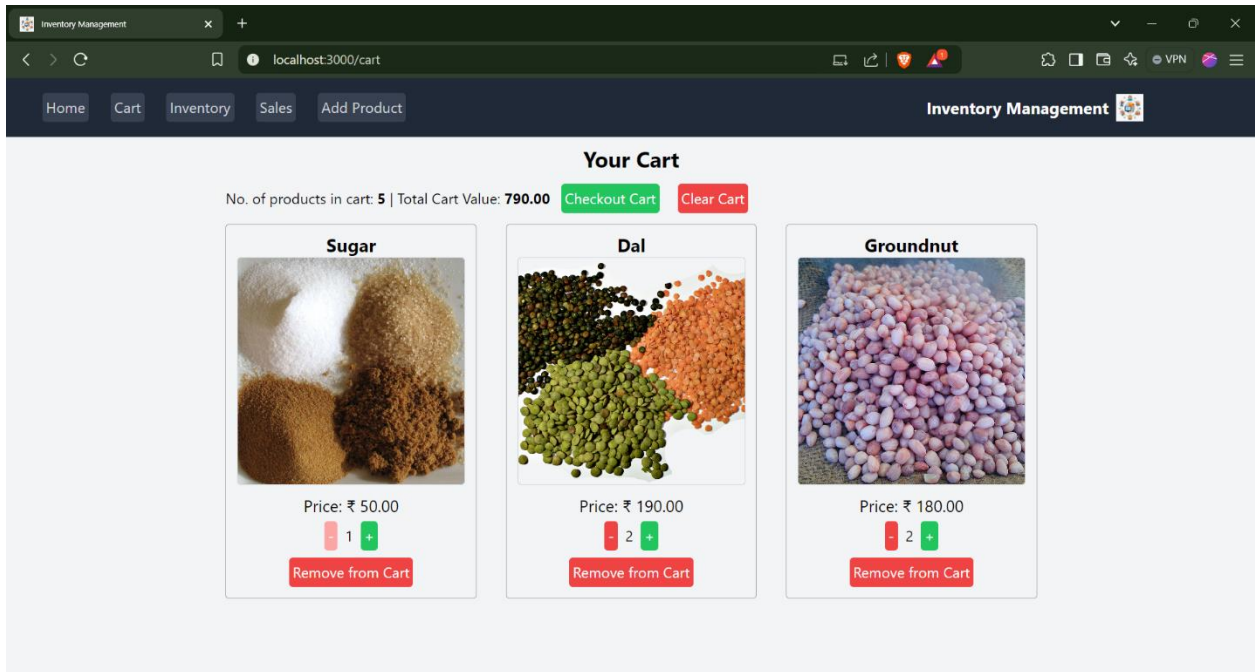
- **Step 1: Test individual components.**
 - Ensure correct rendering and functionality.
- **Step 2: Test context and reducers.**
 - Verify state updates and interactions with localStorage.
- **Step 3: Debug UI/UX issues.**
 - Check for responsiveness and usability.
 -

➤ Project Implementation & Execution

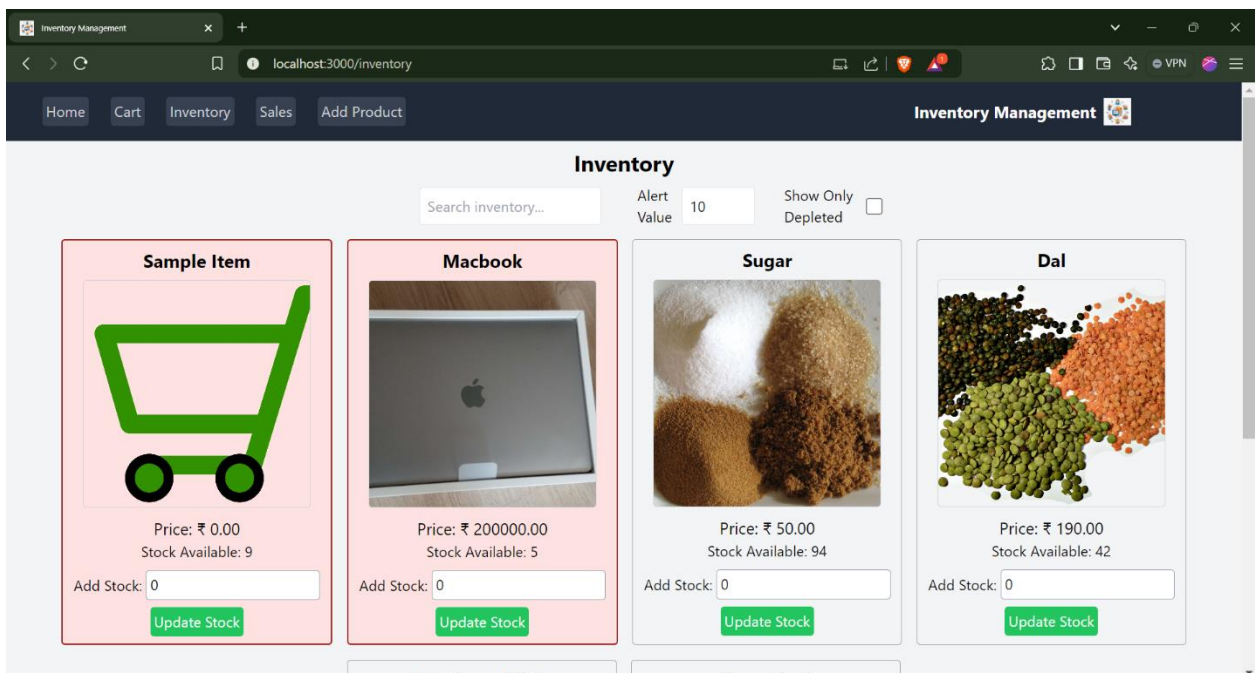
Product Catalog:



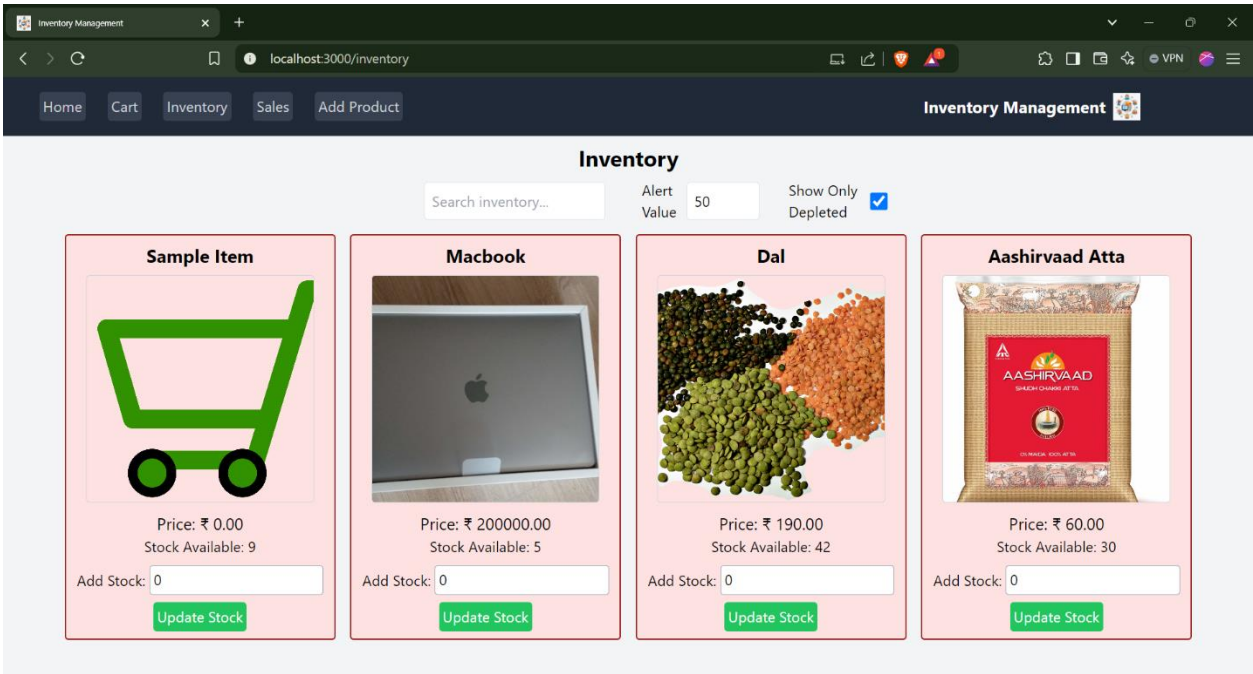
Cart:



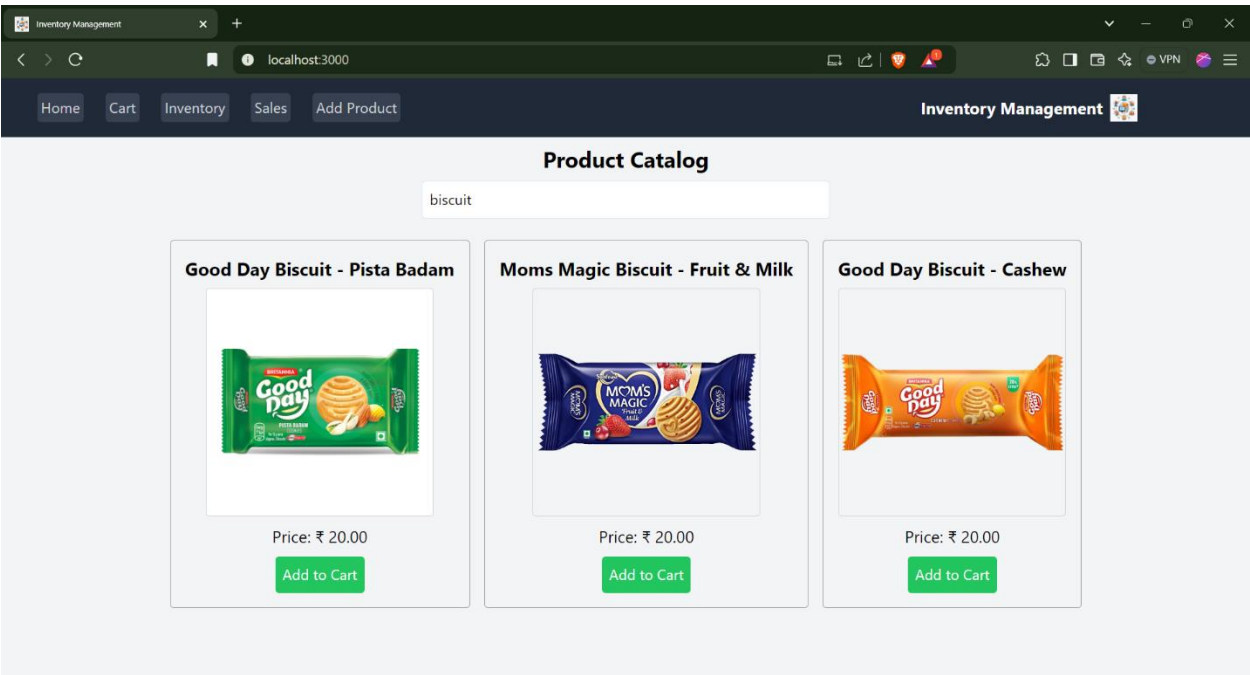
Inventory:



Depleted Stock:



Search Functionality:



Sales:

Inventory Management

localhost:3000/sales

HomeCartInventorySalesAdd Product

Inventory Management

Sales Record

Sale #1

1/3/2025, 4:10:46 PM

Total Sale Value: ₹100.00

Cart Details:

- Sugar (2) - ₹100.00

Sale #2

1/3/2025, 4:16:32 PM

Total Sale Value: ₹4000000.00

Cart Details:

- Macbook (20) - ₹4000000.00

Sale #3

1/3/2025, 4:45:37 PM

Total Sale Value: ₹650.00

Cart Details:

- Dal (2) - ₹380.00
- Aashirvaad Atta (2) - ₹120.00
- Sugar (3) - ₹150.00

Add Product:

Inventory Management

localhost:3000/add-product

HomeCartInventorySalesAdd Product

Inventory Management

Add New Product

Product Name

Enter product name

Product Image URL

Enter image URL

Price

Enter price

Stock

Enter stock

Tags (comma-separated)

Enter tags, separated by commas

Add Product

