

FRONTEND DEVELOPMENT WITH REACT.JS

Store Manager: Keep Track of Inventory

1) Introduction

- **Project Title:** Store Manager – Keep Track of Inventory
- **Team Members:**
 - Ganesh. A – Demonstration
 - Nethaji. P – Editing
 - Sakthivelan. C – Documentation
 - Vishnu Jayaraj. J – Documentation
 - Bharath. K – Coding and Debugging

2) Project Overview

- **Purpose**
 - The project aims to help businesses easily manage the processes of ordering, stocking, storing, and utilizing inventory items.
 - It serves to organize warehouse or storage spaces, enabling quick identification and retrieval of products as needed.
 - Effective inventory management through the store manager helps prevent both overstocking (which ties up capital and increases risk of waste) and understocking (which leads to missed sales and dissatisfied customers).
- **Goals**
 - **Maintain optimal stock levels:** Ensure products are available when needed, preventing stockouts and excess accumulation.
 - **Minimize costs:** Reduce operational expenses such as storage, handling, and waste through efficient inventory practices.

- **Streamline operations:** Automate repetitive tasks, enable real-time tracking, and improve overall warehouse and inventory workflows.
- **Improve customer satisfaction:** Guarantee that the required items are always in stock, leading to timely order fulfillment and enhanced service quality.

- **Features**

- **Dashboard views:** Centralized overview showing current stock levels, item categories, and trends for easy status monitoring.
- **Add, update, delete items:** Forms and tables for quick editing, enabling seamless data entry and maintenance of product records.
- **Barcode/QR code scanning:** Integrated scanning for fast product identification, reducing manual entry errors and speeding up stocktaking.
- **Real-time tracking:** Instant updates as items are sold, restocked, or moved between locations, keeping data accurate and current.
- **Multi-location management:** Lets users view and control inventory across several warehouses or stores from one interface.
- **Automated alerts:** Notifications for low stock, reorder points, and inventory discrepancies to support proactive management.
- **Role-based custom views:** Different interfaces and permissions for managers, staff, or admins, ensuring security and usability.

- **Functionality**

- **Order management:** Generates and tracks purchase orders for inventory replenishment and supplier management.

- **Reporting and analytics:** Provides downloadable reports (e.g., CSV exports), detailed insights, and inventory turnover metrics for data-driven decisions.
- **Inventory movement tracking:** Records product transfers, returns, and restocking, with real-time visibility for accurate control.
- **Search and filter options:** Advanced querying to quickly locate products, sort by category or status, and optimize inventory reviews.

3) Architecture

- **Components Structure**
 - **App.js:** Root component, sets up router and context providers.
 - **Pages:** Each distinct view is a separate page (e.g., Home, AddInventory, Category, SaleHistory, YourCart, Login).
 - **Components:** Reusable UI elements such as Navbar, SearchBar (with mic), CategoryCard, ProductList, etc.
 - **Interaction:** Pages are composed of these components, passing props for data and handling events.
- **State Management**
 - **Global State:** Managed via React Context API, for features like user authentication and cart contents.
 - **Local State:** Managed in each component using the useState hook, for form fields, modal visibility, etc.
- **Routing**
 - **Library Used:** react-router-dom
 - **Structure:** Routes are defined in App.js using <Routes> and <Route>. Navigation is handled using <Link> components for client-side routing.

4) Setup Instructions

- **Prerequisite**
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.
 - a) Download: <https://nodejs.org/en/download/>
 - b) 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.
 - a) Create a new React app:
 - i. `npx create-react-app my-react-app`
 - ii. Replace my-react-app with your preferred project name.
 - b) Navigate to the project directory: `cd my-react-app`
 - c) Running the React App: With the React app created, you can now start the development server and see your React application in action.
 - d) Start the development server:
 - i. `npm start`
This command launches the development server, and you can access your React app at <http://localhost:3000> in your web browser.
- **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:

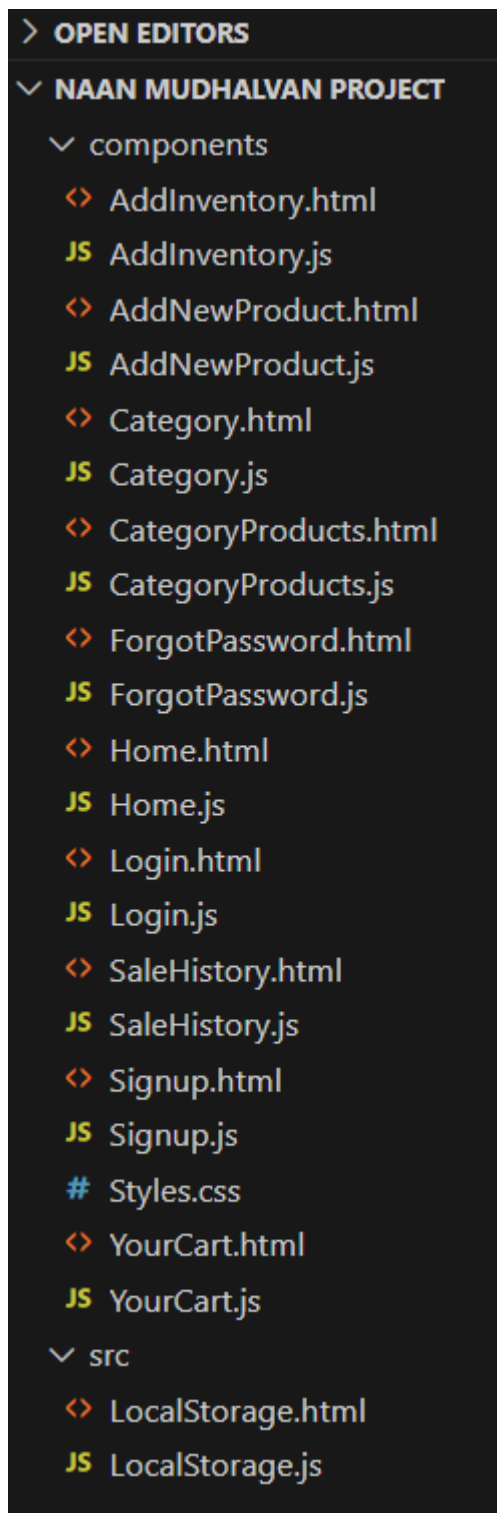
Follow below steps:

- **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.

5) Folder Structure

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



6) Running the Applications

Start the frontend server locally:

- Navigate to the client directory: `cd client`
- Run: `npm start`

The application is available at <http://localhost:3000>.

7) Components Documentation

➤ Key Components

- a) **Navbar**: Displays navigation and user info.
- b) **CategoryCard**: Renders a category with image and name; receives category, image, and onSelect props.
- c) **SearchBar**: Input field with search and mic icon; receives onSearch, placeholder props.

➤ Reusable Components

- a) **Button**: Configurable for style and event.
- b) **Modal**: Used for popups, configurable via props.
- c) **InputField**: For text input, props for value and onChange.

8) State Management

- **Global State**: Managed using React Context (e.g., UserContext for authentication, CartContext for cart), provided at the top level so all components can access global state.
- **Local State**: Handled by useState within components, for things like form fields, toggles, and transient UI state.

9) User Interface

- **Home**: Shows complete products availability.
- **Add Inventory**: Form for adding products.
- **Category Selection**: Grid of categories with images.
- **Search Bar**: Allows text input for search.
- **Cart**: Displays items in cart.

10) Styling

➤ CSS Frameworks / Libraries

- a) Custom CSS (Styles.css) used for most UI.
- b) No third-party frameworks (e.g., Bootstrap) unless specified.

c) Optionally, Styled-Components for scoped styling.

➤ **Theming**

a) Colour variables and CSS custom properties enable easy theme changes (light/dark modes).

11) Testing

➤ **Testing Strategy**

- a) **Unit Tests:** For components and utilities using Jest and React Testing Library.
- b) **Integration Tests:** For flows like login and cart.
- c) **End-to-End Tests:** (Optional) Can use Cypress.

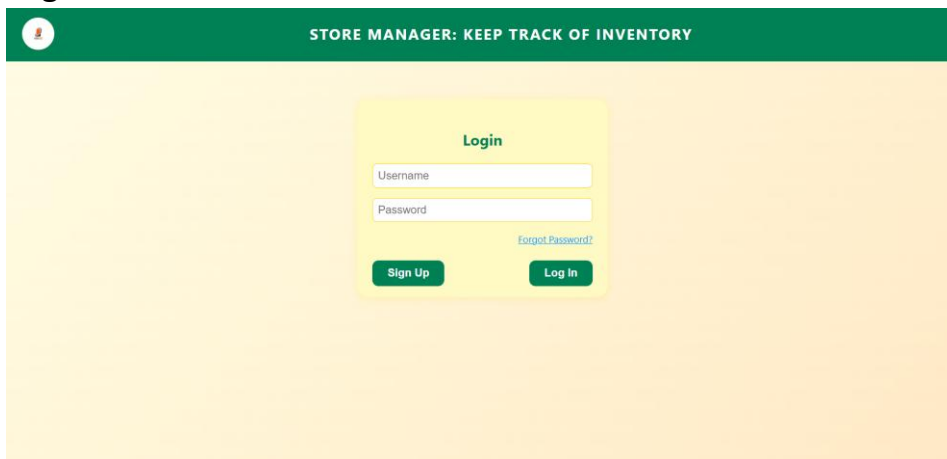
➤ **Code Coverage**

- a) Run `npm test -- --coverage` to generate coverage reports.

12) Screenshots and Demo


➤ **Screenshots**

Login



The screenshot shows a web application interface. At the top, there is a dark green header bar with a small circular logo on the left and the text "STORE MANAGER: KEEP TRACK OF INVENTORY" in white. Below the header, the main background is a light yellow gradient. In the center, there is a white rounded rectangle containing the login form. The form has a title "Login" in green. It includes two input fields: "Username" and "Password". Below the password field is a blue link that says "Forgot Password?". At the bottom of the form are two green buttons: "Sign Up" and "Log In".


Signup

 **STORE MANAGER: KEEP TRACK OF INVENTORY**

Sign Up

Sign Up


Forgot Password

 **STORE MANAGER: KEEP TRACK OF INVENTORY**

Forgot Password?


Change Password

Home

 **STORE MANAGER: KEEP TRACK OF INVENTORY** Hello, Ganesh A [Logout](#)

[HOME](#) [ADD INVENTORY](#) [CATEGORY](#) [SALE HISTORY](#) [YOUR CART](#)

All Categories




Apple
Category: Fruits and Vegetables
Stock: 149
Price: ₹ 30

-

0

+




Mango
Category: Fruits and Vegetables
Stock: 129
Price: ₹ 30

-

0

+




Banana
Category: Fruits and Vegetables
Stock: 99
Price: ₹ 60

-

0

+




Orange
Category: Fruits and Vegetables
Stock: 129
Price: ₹ 40

-

0

+








Grapes
Category: Fruits and Vegetables
Stock: 89
Price: ₹ 75

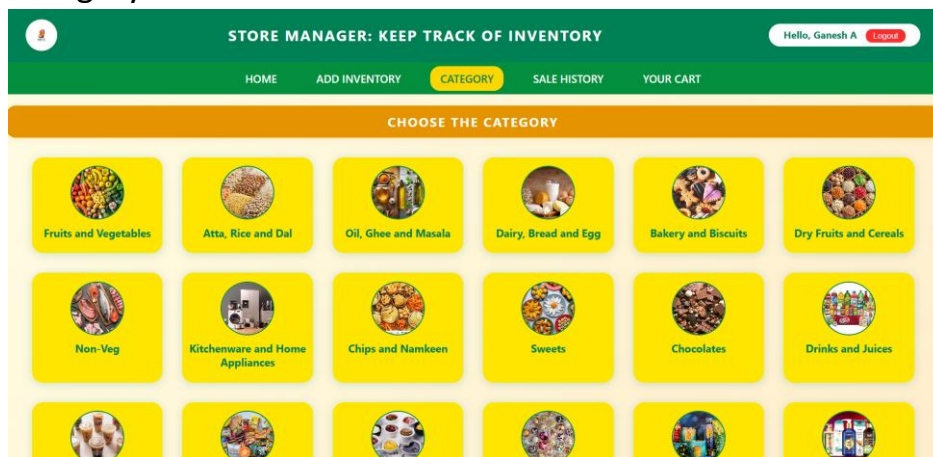
-

0

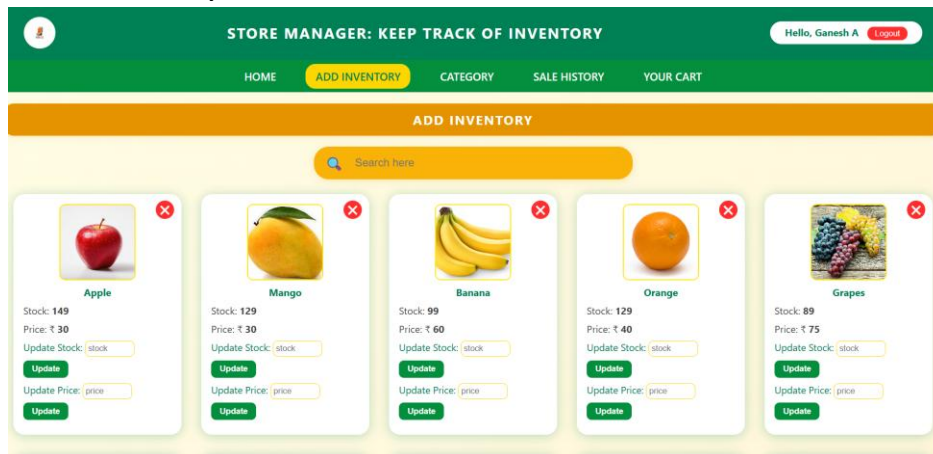
+



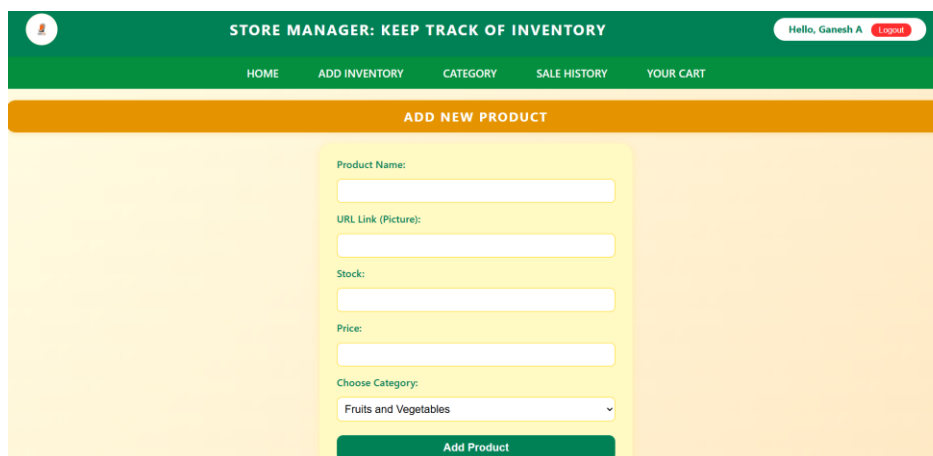
Category




Add Inventory



Add Products



Category Products




STORE MANAGER: KEEP TRACK OF INVENTORY

Hello, Ganesh A [Logout](#)

HOMEADD INVENTORYCATEGORYSALE HISTORYYOUR CART

Fruits and Vegetables Products




Apple

Stock: 149

Price: ₹ 30

- 0 +



Mango

Stock: 129

Price: ₹ 30

- 0 +




Banana

Stock: 99

Price: ₹ 60

- 0 +




Orange

Stock: 129

Price: ₹ 40

- 0 +




Grapes


Stock: 89

Price: ₹ 75


- 0 +




Potato




Tomato



Brinjal




Pumpkin



Watermelon

Your Cart



STORE MANAGER: KEEP TRACK OF INVENTORY

Hello, Ganesh A [Logout](#)


HOMEADD INVENTORYCATEGORYSALE HISTORYYOUR CART

YOUR CART

Product	Qty	Price
Brinjal	1	₹35
Muskmelon	1	₹60
Cucumber	1	₹40
Corn	2	₹80
Pomegranate	1	₹30
Litchi	1	₹40
Total (18% GST incl.)		₹336

Proceed

Sales History



STORE MANAGER: KEEP TRACK OF INVENTORY

Hello, Ganesh A [Logout](#)

HOMEADD INVENTORYCATEGORYSALE HISTORYYOUR CART

SALE HISTORY

Sale NO. #3

15/9/2025, 11:14:11 pm

Product	Qty	Price
Brinjal	1	₹35
Muskmelon	1	₹60
Cucumber	1	₹40
Corn	2	₹80
Pomegranate	1	₹30
Litchi	1	₹40
Total (18% GST incl.)		₹336

Sale NO. #2

15/9/2025, 9:28:07 am

Product	Qty	Price
Corn	1	₹40
Onion	1	₹30
Lady finger	1	₹45
Watermelon	1	₹55
Potato	1	₹25
Tomato	1	₹30

➤ **Demo**

Link:

<https://drive.google.com/file/d/10zW8yq6e8mcXAgvChnGetBH8w34krhw0/view>

13) Known Issues

- Speech recognition (mic) works only in Chrome/Edge, not Firefox/Safari.
- Mobile responsiveness needs improvement on some pages.
- Unsplash category images may not load offline.

14) Future Enhancement

- Backend integration for real-time inventory.
- Improved mobile and responsive design.
- Progressive Web App (PWA) features.
- More UI animations and transitions.
- Advanced search and filtering.
- Enhanced theming and user customization.