

# Full Stack Development with MERN

## Frontend Development Report

Date	19 July 2024
Team ID	SWTID1720150432
Project Name	EagerEats-Food Ordering App
Maximum Marks	

### Project Title: Eager Eats

Date: 19 July 2024

Prepared by: PRAVEEN S





---

### Objective

The objective of this report is to document the frontend development progress and key aspects of the user interface implementation for the “Eager Eats” project..

---

### Technologies Used

-  **Frontend Framework:** React.js
-  **State Management:** Context API
-  **UI Framework/Libraries:** Bootstrap
-  **API Libraries:** Axios

### Project Structure

/src

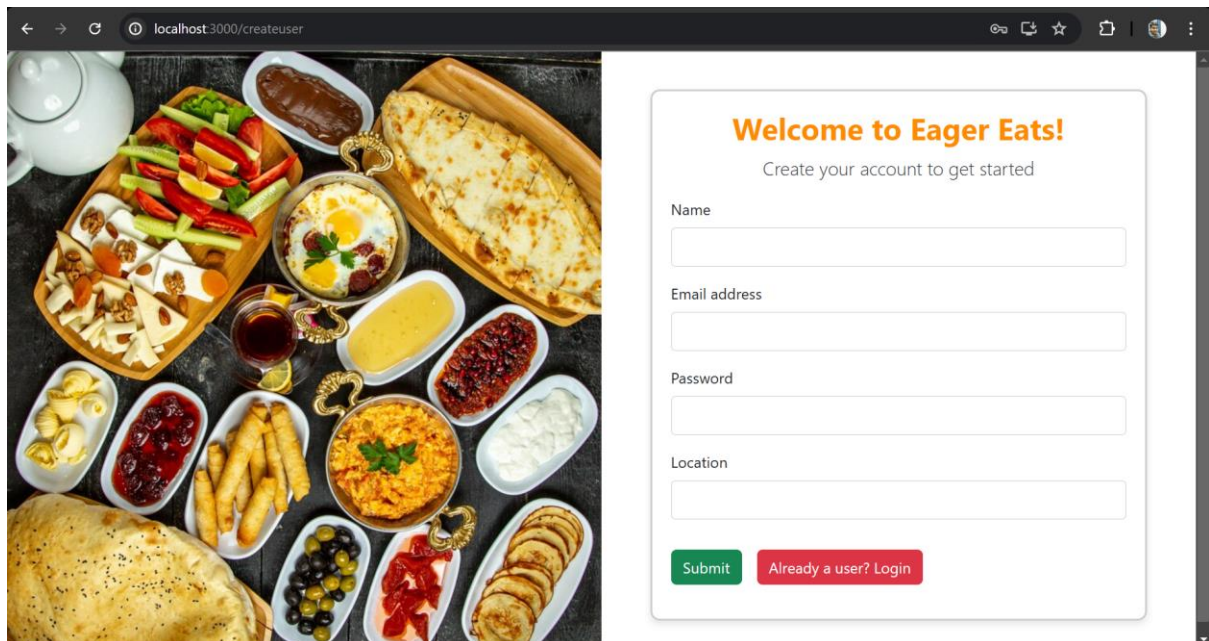
└─ App.js: Responsible for routing and main application layout.

└─ /components: Contains reusable UI components used across the application.

└─ /pages: Includes different pages for the Web App.

**Login page :**

The login page allows users to enter their email and password to access the application. It features a welcoming message and options for submitting credentials or signing up as a new user. The form is styled using Bootstrap for a clean and responsive design.



## **Add to cart page :**

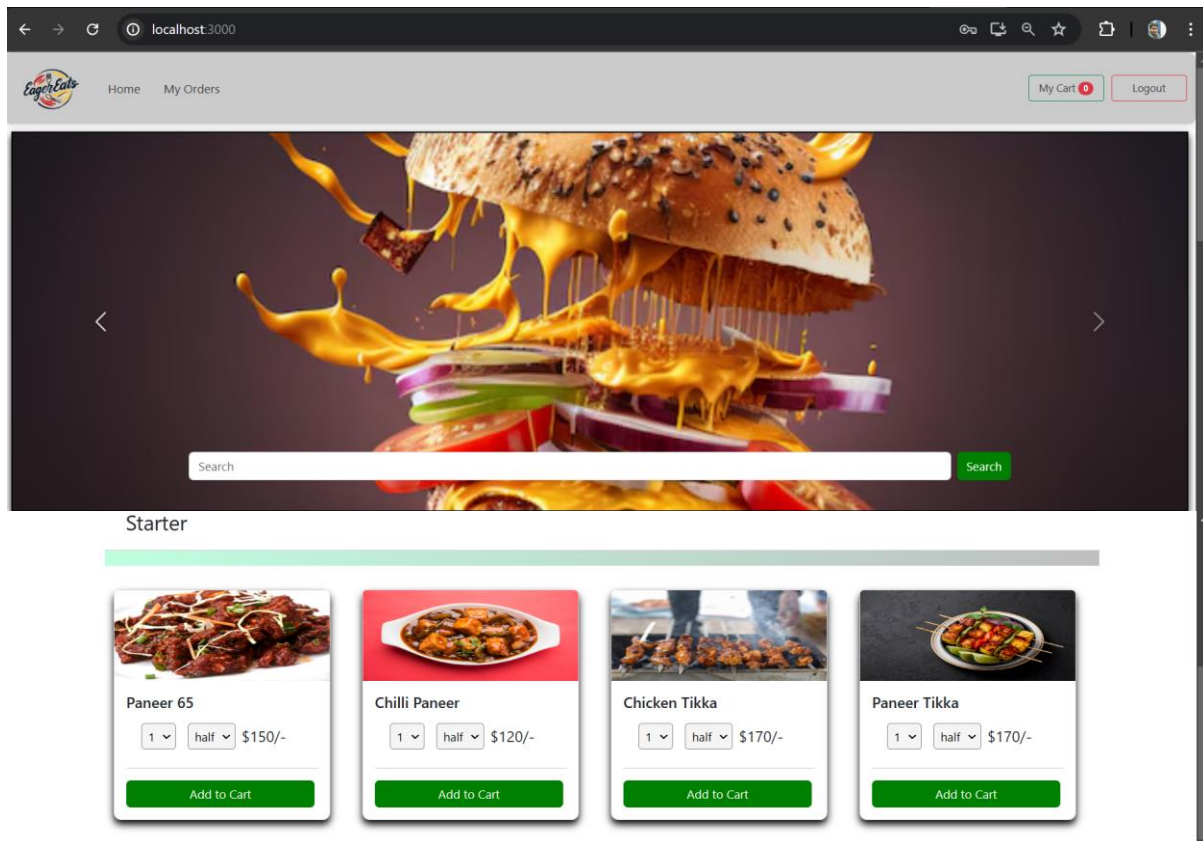
The Add to Cart functionality allows users to select items from the menu and add them to their cart. The cart displays a summary of the selected items, their quantities, options, and total amount. Users can proceed to checkout or remove items from the cart.

### **Top Section:**

- The top section displays the available food items with options to add them to the cart.
- Each item card includes a name, image, price, and an "Add to Cart" button.

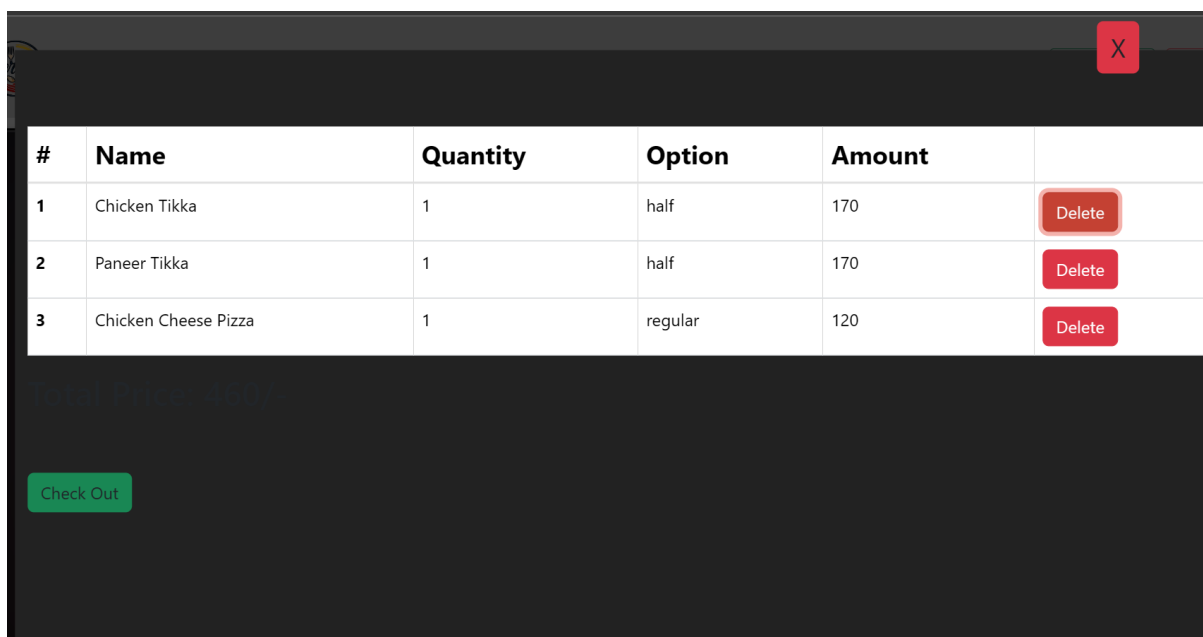
### **Bottom Section:**

- The bottom section is the cart summary which lists all the items added to the cart.
- It shows the item name, quantity, selected options, and the total amount for each item.
- Users can delete items or proceed to checkout from this section.



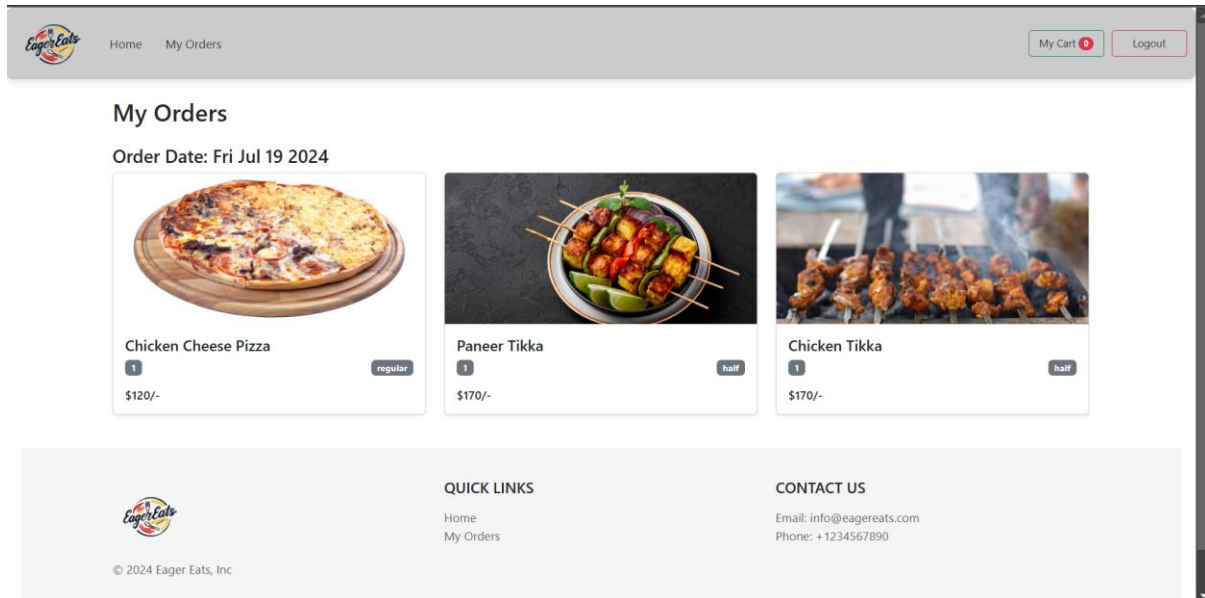
## My Cart Page

The My Cart page allows users to review the items they have added to their cart. It includes options to adjust quantities, select item options, and view the total amount before proceeding to checkout. The cart interface is user-friendly and provides an overview of the selected items, facilitating easy modifications or deletions.



## My Orders Page

The My Orders page displays a list of orders placed by the user, including the order date, items ordered, and their respective details such as quantity and price. This page helps users keep track of their past and current orders.



## Key Components

- **App.js:** Responsible for routing and the main application layout.
- **components:** Contains reusable UI components used across the application.

🔗 Button.jsx, Navbar.jsx, SignIn.jsx, SignUp.jsx, TextInput.jsx, ProductCategoryCard.jsx, ProductsCard.jsx

- **pages:** Includes different pages for the web app.

🔗 Authentication.jsx, Cart.jsx, FoodDetails.jsx, FoodListing.jsx, Home.jsx

- **redux:** State management using Redux.

🔗 store.js, reducers/SnackbarSlice.js, reducers/UserSlice.js

## Routing

Routing is managed using React Router. Here are the main routes:

- /home: Landing page of the application.
- /dashboard: Dashboard displaying user data and statistics.
- /profile: User profile management.

## State Management (If Applicable)

State management is achieved using Redux. The store is set up in `store.js`, and reducers are defined in `SnackbarSlice.js` and `UserSlice.js`.

## Integration with Backend

The frontend communicates with the backend APIs hosted on [backend URL]. Key endpoints include:

- **GET /api/data:** Retrieves data for display.
- **POST /api/user/login:** Handles user authentication.

## User Interface (UI) Design

The UI design follows a modern and clean design principle. It is implemented using Material-UI, ensuring a consistent and responsive design across different devices.

## Third-Party Integrations (If any)

The project integrates several third-party libraries and services:

- 🚀 **Material-UI:** For UI components.
- 🚀 **Axios:** For API requests.