

**MIT Art, Design and Technology University**

**MIT School of Computing, Pune**

**Department of Information Technology**

|  |
| --- |
| **Lab Manual** |

# **Practical - Web Programming**

# **Class – TY-IT-SMAD**

# **Batch - SMAD**

# **Mr. Sainjal Kalnekar**

**A.Y. 2024 – 2025 (SEM-IV)**

|  |  |
| --- | --- |
| **Assignment No 1: Project Assignment based on Unit-I and Unit-II** | |
| **Project-I** | Project Title: ReactJS E-Commerce Website with Page RoutingProject Statement: Objective: Develop a fully functional E-commerce website using ReactJS that includes page routing functionality for smooth navigation between different sections like Home, Product Listing, Product Details, Cart, and Checkout. Project Features:  1. Navigation Bar with Routing:    * Create a Navigation Bar at the top of the website with links to:      + Home: Displays featured products and general information about the store.      + Products: Lists all available products for purchase.      + Cart: Displays the products added to the shopping cart.      + Checkout: Allows the user to review and complete the purchase.      + Profile: Displays user account information and order history.    * Implement React Router for routing between these pages. 2. Home Page:    * The Home page should showcase highlighted products or categories.    * Implement a product carousel or featured section.    * Include a search bar to allow users to search for products directly. 3. Product Listing Page:    * Create a Product Listing page that fetches products dynamically from an API or static JSON file.    * Each product in the list should show:      + Product image.      + Product name.      + Price.      + A button to add the product to the cart.    * Implement React Router links on product items to navigate to the Product Details page. 4. Product Details Page:    * When a user clicks on a product, navigate them to the Product Details page.    * The Product Details page should display:      + Full-size product image.      + Detailed product description.      + Price.      + Add to Cart button.    * Allow users to select product quantity before adding it to the cart. 5. Cart Page:    * Implement a Cart page that displays products added to the cart.    * Each cart item should include:      + Product name.      + Product quantity.      + Price per item and total cost for the item.      + A button to remove items from the cart.    * Display a total price for all items in the cart.    * Include buttons for Continue Shopping and Proceed to Checkout. 6. Checkout Page:    * Create a Checkout page where users can review their cart, enter shipping information, and confirm the order.    * Display an order summary with total cost, shipping details, and payment options.    * Provide a Place Order button to finalize the transaction. 7. User Profile Page:    * Create a Profile page where users can view their account information and order history.    * Include a form to update the profile details (name, email, shipping address). 8. State Management with Context API/Redux:    * Use React Context API or Redux to manage the global state of the cart, allowing users to add, remove, and update quantities of products across different pages.    * Ensure that the cart data is preserved during navigation between pages. 9. Responsive Design:    * Ensure the website is responsive and works well on both desktop and mobile devices using CSS Flexbox or CSS Grid.  Technical Requirements:ReactJS: For building the user interface.React Router: For implementing page routing functionality.State Management: Use React Context API or Redux to manage cart state and other global data.API Integration: Fetch products from a static JSON file or a mock API (e.g., JSONPlaceholder or a custom API).Responsive Styling: Use CSS/SCSS to ensure the website is mobile-friendly.Optional: Add localStorage or sessionStorage to persist cart data between page reloads.Expected Outcome: By completing this project, you will:   * Learn how to implement page routing in a ReactJS application using React Router. * Build a dynamic E-commerce website with features like product listings, cart management, and checkout functionality. * Gain experience with state management and ensuring the cart data persists across different pages. * Develop a responsive, user-friendly interface that functions well on both mobile and desktop screens. |
| **Assignment No 2: Project Assignment based on Unit-III, IV and V** | |
| **Project-II** | Project Title: React Native E-Commerce ApplicationProject Statement: Objective: Build a fully functional E-commerce mobile application using React Native that integrates navigation, core components, user interaction, dynamic list rendering, and state management. Project Features:  1. User Authentication:    * Implement a simple login screen where users can enter their credentials (username and password).    * After successful login, navigate to the Home Screen. 2. Navigation:    * Set up react-navigation to allow users to navigate between the following screens using StackNavigator:      + Home Screen: Displays a list of products.      + Product Details Screen: Shows details of the selected product.      + Cart Screen: Displays products added to the shopping cart.      + Profile Screen: Displays user profile details.      + Settings Screen: Allows the user to modify their account settings. 3. Core Components & UI:    * Use core React Native components such as View, Text, Image, StyleSheet, and Button to build a user-friendly interface.    * Apply Flexbox for layout, ensuring responsiveness across different screen sizes. 4. Product List and Cart:    * On the Home Screen, render the list of products using FlatList. Each item should display a product image, name, and price.    * Implement a button for each product to add it to the Cart. 5. Handling User Interactions:    * Implement touch events to handle interactions such as adding products to the cart, navigating to the product details, and updating the cart.    * Use onPress to trigger actions like adding a product to the cart or navigating between screens. 6. Cart Management:    * Use React Context API for global state management to handle the cart’s data (adding/removing products).    * Show the cart’s content and total price on the Cart Screen. 7. Product Details:    * When a user clicks on a product in the Home Screen, navigate to the Product Details Screen and display detailed information about the product, such as the description, image, and price. 8. Responsive Design:    * Ensure that the app is responsive, and elements adjust properly to different screen sizes using Flexbox. 9. Rendering Lists with SectionList:    * On the Home Screen, group products by categories (e.g., electronics, clothing) and use SectionList to render them in separate sections with headers.  Technical Requirements:React Native with Expo CLI for development and testing.React Navigation for screen transitions.State Management using React Context API or useState, useReducer.Dynamic data rendering using FlatList and SectionList.Styling using StyleSheet and Flexbox for layout.Expected Outcome: By completing this project, students will gain hands-on experience with:   * Setting up and using React Native and React Navigation. * Building a fully interactive mobile application with dynamic data rendering and state management. * Applying Flexbox for responsive layouts and managing user interactions. * Creating a simple e-commerce app with multi-screen navigation, product lists, and state-driven cart functionality. |

# 

## **Experiment No.1**

## 

**Problem Statement:**   
  
In today’s digital age, having an online presence is important for restaurants to attract and retain customers. **Little Lemon**, a Mediterranean restaurant, needs a responsive and user-friendly website that allows visitors to explore the menu, learn about the restaurant, and contact the team easily. Additionally, the site should support user registration and login to enhance the customer experience. This project focuses on building a simple and functional web application using React to meet these needs and improve customer interaction.

## **Objective:**

Create an engaging and functional restaurant website using React that includes the following pages:

* **Home**: The landing page that gives users a warm welcome and introduces the theme and offerings of Little Lemon.
* **Menu/Product**: Displays a list of dishes offered by the restaurant, including names, prices, and possibly images or descriptions.
* **Cart**: Allows users to view selected items (static in this version), helping them keep track of their intended order.
* **About Us**: Shares the background, values, and mission of the restaurant, helping build a connection with customers.
* **Contact Us**: Provides essential contact information, such as phone number, email, and location, making it easy for users to reach out.
* **Registration**: Enables new users to sign up by entering their email and password to create an account.
* **Login**: Lets returning users sign in with their credentials to access personalized features.

## **Theory:**

The project uses **React** for building dynamic and reusable UI components, which allows for a fast and efficient development process. Each page (such as Home, Menu, Cart, etc.) is designed as a separate component, enabling modularity and better code management.

**React Router** is used to manage navigation between pages without refreshing the browser, providing a smooth single-page application (SPA) experience. This enhances user interaction by making transitions seamless and quick.

For styling, the project uses **basic CSS** to create a clean and responsive layout, ensuring the website works well on both desktop and mobile devices.

Temporary data such as user login information or cart details is stored using **localStorage** or component-level **React state**. This allows users to maintain session-like behavior (e.g., staying logged in or preserving cart items) without connecting to a backend server, keeping the app simple and fully client-side.

## **Code:**

A. Home page:

code:  
// src/pages/HomePage.js

import React from "react";

import { Link } from "react-router-dom";

import img1 from "../assets/icons\_assets/restauranfood.jpg";

const HomePage = () => {

return (

<section className="grid grid-cols-4 px-10 lg:grid-cols-12 md:grid-cols-8 gap-x-5 lg:px-16 bg-yellow-100 h-fit">

<div className="col-start-1 col-span-2 lg:col-start-3 lg:col-end-7 md:col-start-1 md:col-span-5">

<h1 className="text-5xl font-bold pt-6 text-green-800 font-serif">Little Lemon</h1>

<p className="pt-3 pb-6 text-yellow-900 text-3xl font-semibold">Chicago</p>

<p className="pb-6 text-yellow-900 font-sans text-lg">

We are a family-owned Mediterranean restaurant, focused on traditional recipes served with a modern twist.

</p>

<Link to="/Little-Lemon/booking">

<button className="w-full lg:w-52 h-14 mb-4 bg-green-700 text-white rounded-lg text-xl font-semibold hover:bg-green-800 transition duration-150 ease-in-out">

Reserve a table

</button>

</Link>

</div>

<img

src={img1}

alt="Restaurant food"

className="object-cover col-start-3 col-span-2 mt-10 ml-10 rounded-lg

justify-self-center self-center lg:col-start-8 lg:col-span-3

md:col-start-6 md:col-span-4 md:h-72 lg:h-[475px] lg:w-80"

/>

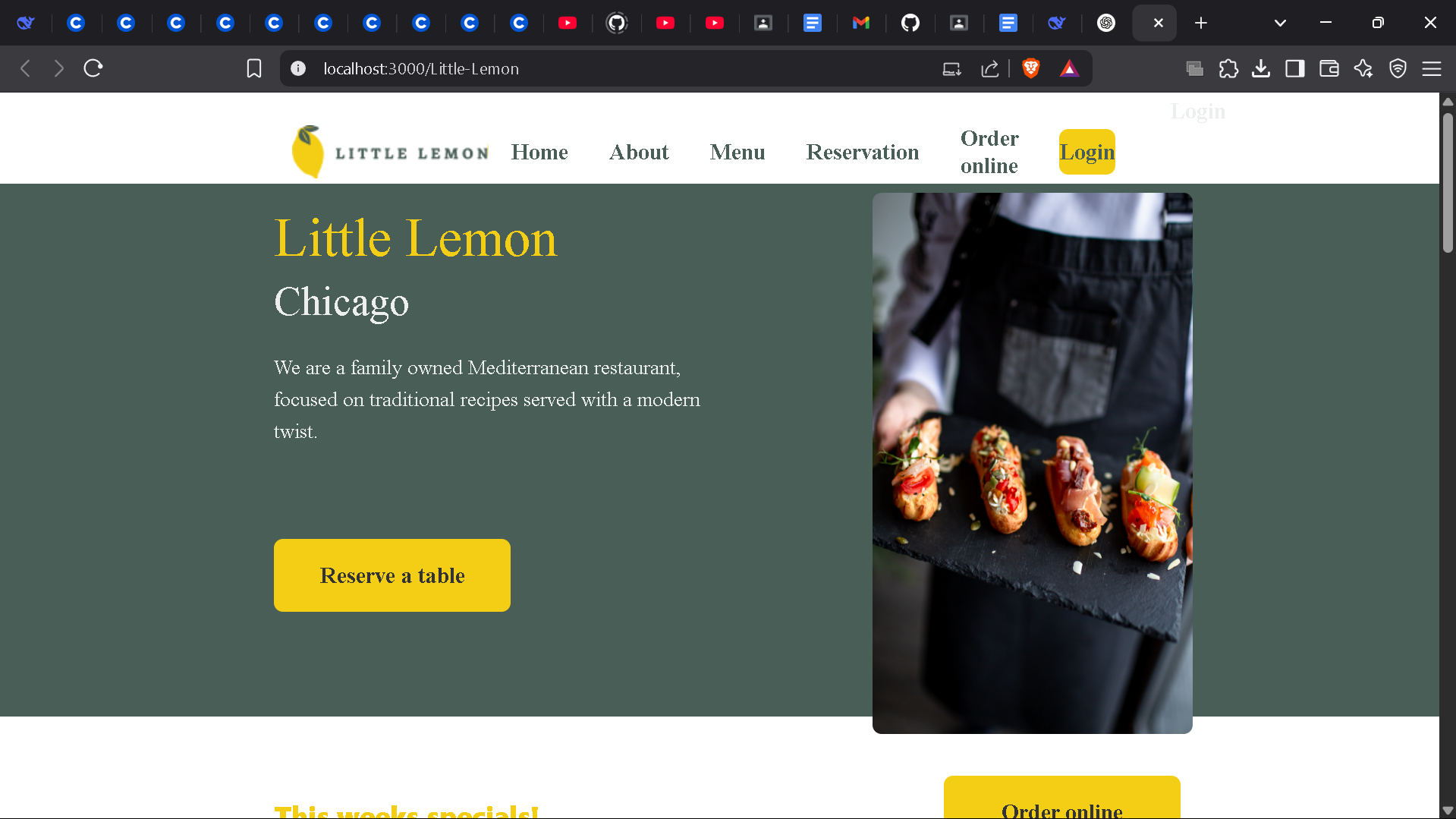
</section>

);

};

export default HomePage;

## **Output:**

A. Index/Home page output:  


## 

## **Code:**

B. menu/product page:

code:  
// src/pages/MenuPage.js

import React from "react";

const dishes = [

{ name: "Greek Salad", price: "$12", description: "Fresh veggies and feta." },

{ name: "Bruschetta", price: "$8", description: "Grilled bread with tomatoes." },

{ name: "Lemon Dessert", price: "$6", description: "Citrus flavored treat." }

];

const MenuPage = () => {

return (

<div className="p-10 bg-white">

<h2 className="text-4xl font-bold mb-6">Menu</h2>

<div className="grid md:grid-cols-3 gap-6">

{dishes.map((dish, index) => (

<div key={index} className="border p-4 rounded-lg shadow-md hover:shadow-xl">

<h3 className="text-2xl font-semibold">{dish.name}</h3>

<p className="text-gray-600">{dish.description}</p>

<p className="font-bold mt-2">{dish.price}</p>

</div>

))}

</div>

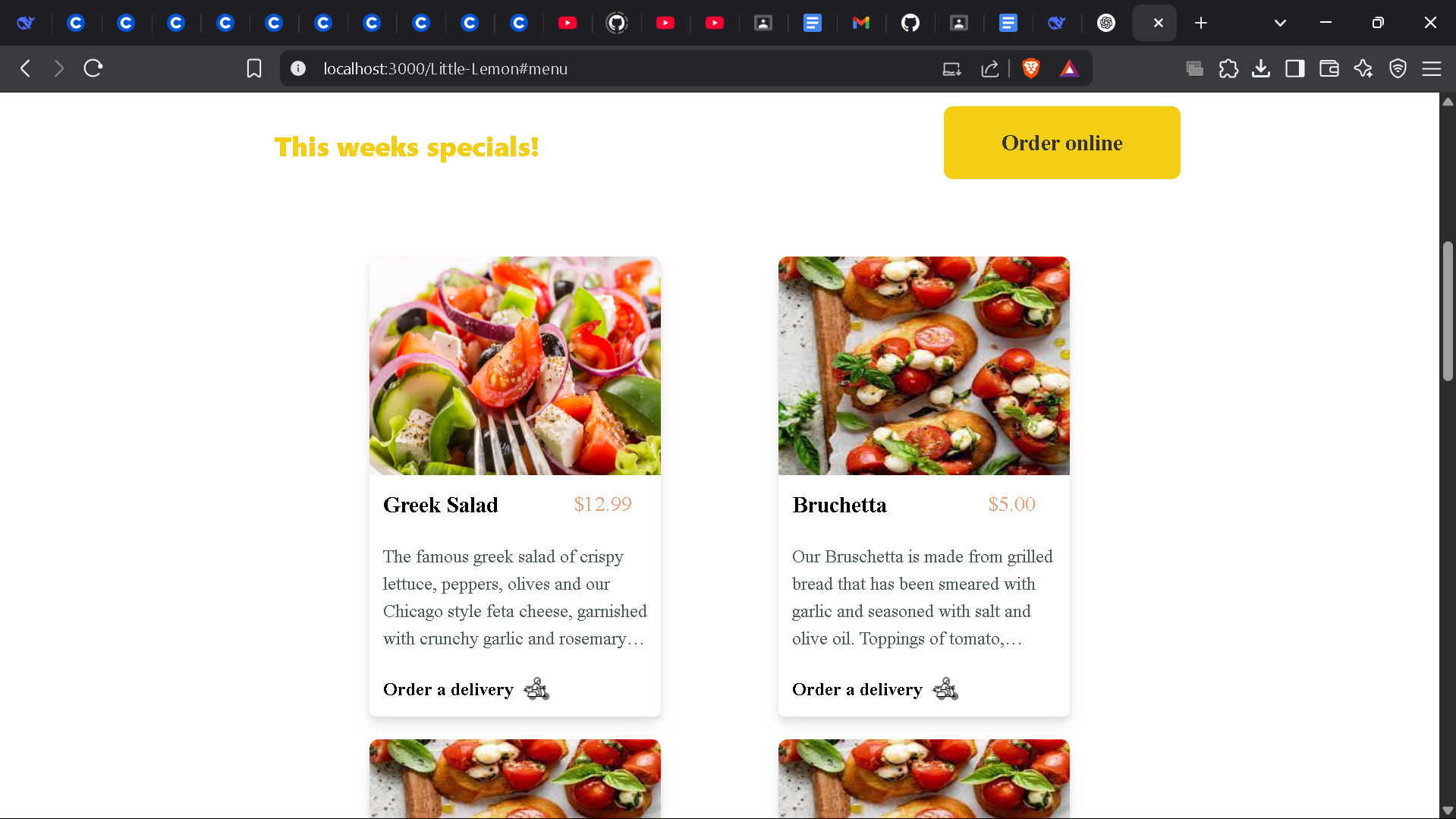
</div>

);

};

export default MenuPage;

## **Output:**

B. menu/product page output:  


## **Code:**

C. cart page:

code:

## **Output:**

C. cart page output:

## **Code:**

D. about us page:

code:  
import img1 from "../assets/icons\_assets/Mario and Adrian A.jpg"

import img2 from "../assets/icons\_assets/Mario and Adrian b.jpg"

const About = ()=>{

    return (

        <section className="grid grid-cols-4 auto-rows-auto px-10 mt-10 lg:grid-cols-12 md:grid-cols-8 gap-x-5 gl:px-16 " id="about">

            <div className="col-start-1 col-span-4 lg:col-start-3 lg:col-end-7 md:col-start-1 md:col-span-5">

                    <h1 className="font-main text-5xl font-medium pt-6 text-primary-1">Little Lemon</h1>

                    <p className="pt-3 pb-6 font-main text-Highlight-2 text-3xl">Chicago</p>

                    <p className="lg:pb-10 md:pb-9 pb-6 text-Highlight-2 font-second lg:text-lg text-base">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>

            </div>

            <div className="col-start-1 col-span-4 justify-self-end col-end-4 flex h-fit justify-start md:col-start-6 md:justify-start md:col-span-3 md:mt-10 lg:col-start-7 lg:col-span-4 self-center">

                <img alt="photo\_1" src={img1} className="w-36 h-52 rounded-md mt-12 hover:z-50 lg:w-60 lg:h-80 "/>

                <img alt="photo\_2" src={img2} className="w-36 h-52 rounded-md ml-[-46px] z-40 hover:z-50 lg:w-60 lg:h-80 "/>

            </div>

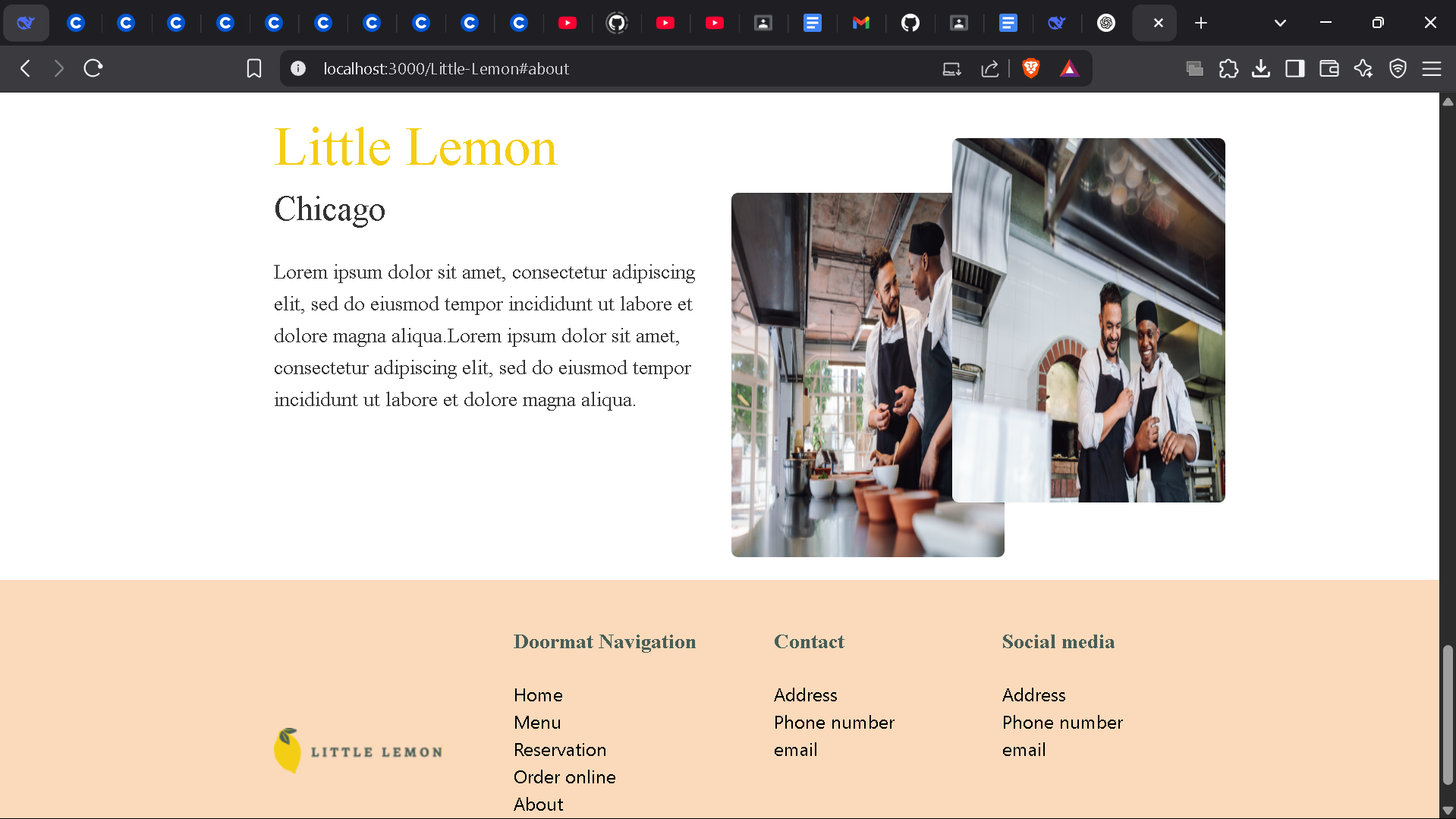
        </section>

    )

}

export default About;

## **Output:**

D. about us page output:  


## **Code:**

E. contact us page:

code:

// src/pages/ContactPage.js

import React from "react";

const ContactPage = () => {

return (

<div className="p-10 bg-gray-50">

<h2 className="text-4xl font-bold mb-4">Contact Us</h2>

<p>📍 123 Main St, Chicago</p>

<p>📞 (555) 123-4567</p>

<p>📧 info@littlelemon.com</p>

</div>

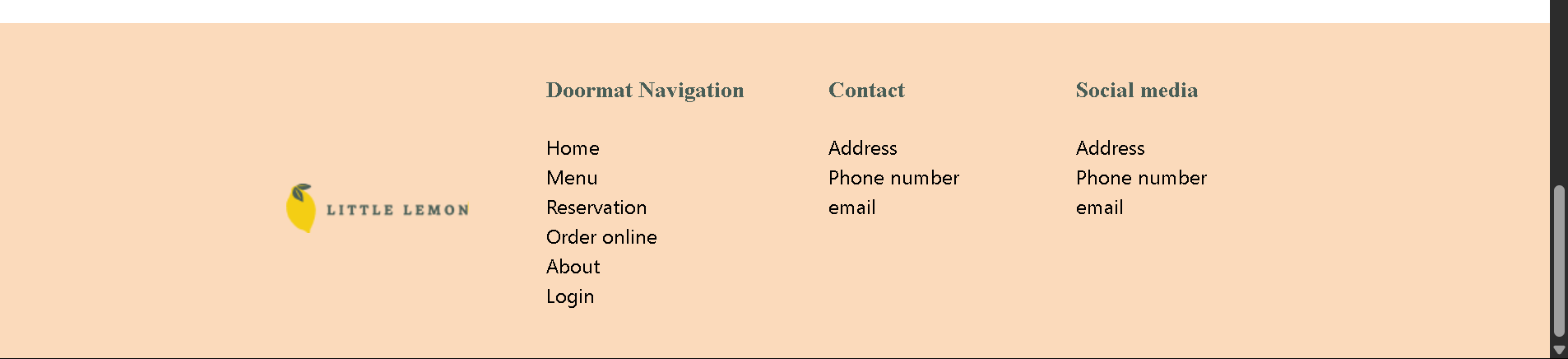
);

};

export default ContactPage;

## **Output:**

E. contact us page output:



## **Code:**

F. registration page:

code:  
// src/pages/RegistrationPage.js

import React, { useState } from 'react';

function RegistrationPage() {

const [email, setEmail] = useState('');

const [password, setPassword] = useState('');

const handleRegister = () => {

localStorage.setItem('userEmail', email);

localStorage.setItem('userPassword', password);

alert('Registration successful!');

};

return (

<div>

<h2>Register</h2>

<input type="email" placeholder="Email" onChange={(e) => setEmail(e.target.value)} />

<input type="password" placeholder="Password" onChange={(e) => setPassword(e.target.value)} />

<button onClick={handleRegister}>Register</button>

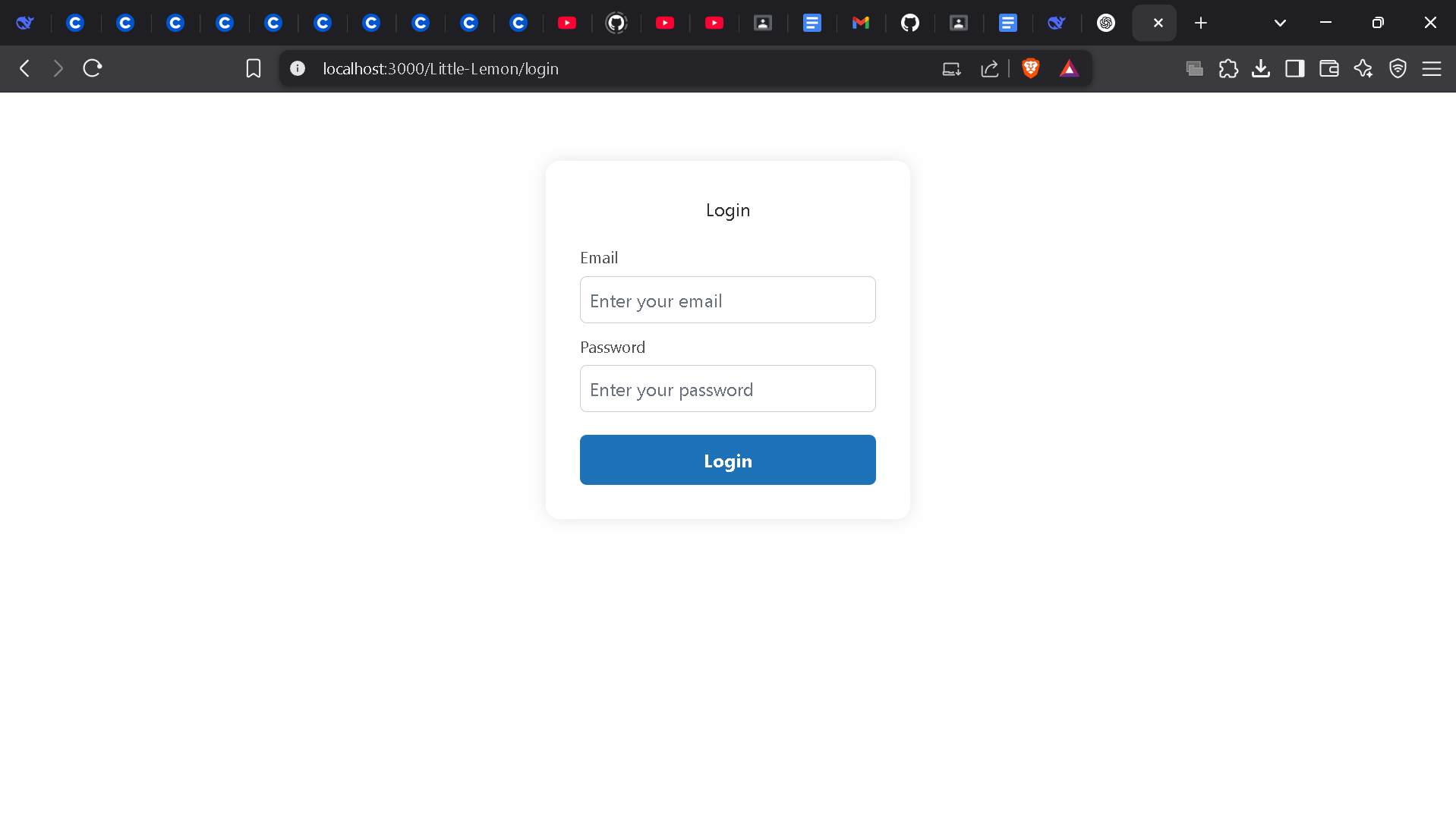
</div>

);

}

export default RegistrationPage;

## **Output:**

F. registration page output:  


## **Code:**

G. login page:

code:  
// src/componant/LoginPage.js

import React, { useState } from 'react';

import { useNavigate } from 'react-router-dom';

function LoginPage() {

const [email, setEmail] = useState('');

const [password, setPassword] = useState('');

const [message, setMessage] = useState('');

const navigate = useNavigate();

const handleLogin = () => {

const storedEmail = localStorage.getItem('userEmail');

const storedPassword = localStorage.getItem('userPassword');

if (email === storedEmail && password === storedPassword) {

setMessage('✅ You are logged in successfully!');

setTimeout(() => navigate('/Little-Lemon/'), 2000);

} else {

setMessage('❌ Invalid credentials. Please try again.');

}

};

return (

<div>

<h2>Login</h2>

<input type="email" placeholder="Email" onChange={(e) => setEmail(e.target.value)} />

<input type="password" placeholder="Password" onChange={(e) => setPassword(e.target.value)} />

<button onClick={handleLogin}>Login</button>

<p>{message}</p>

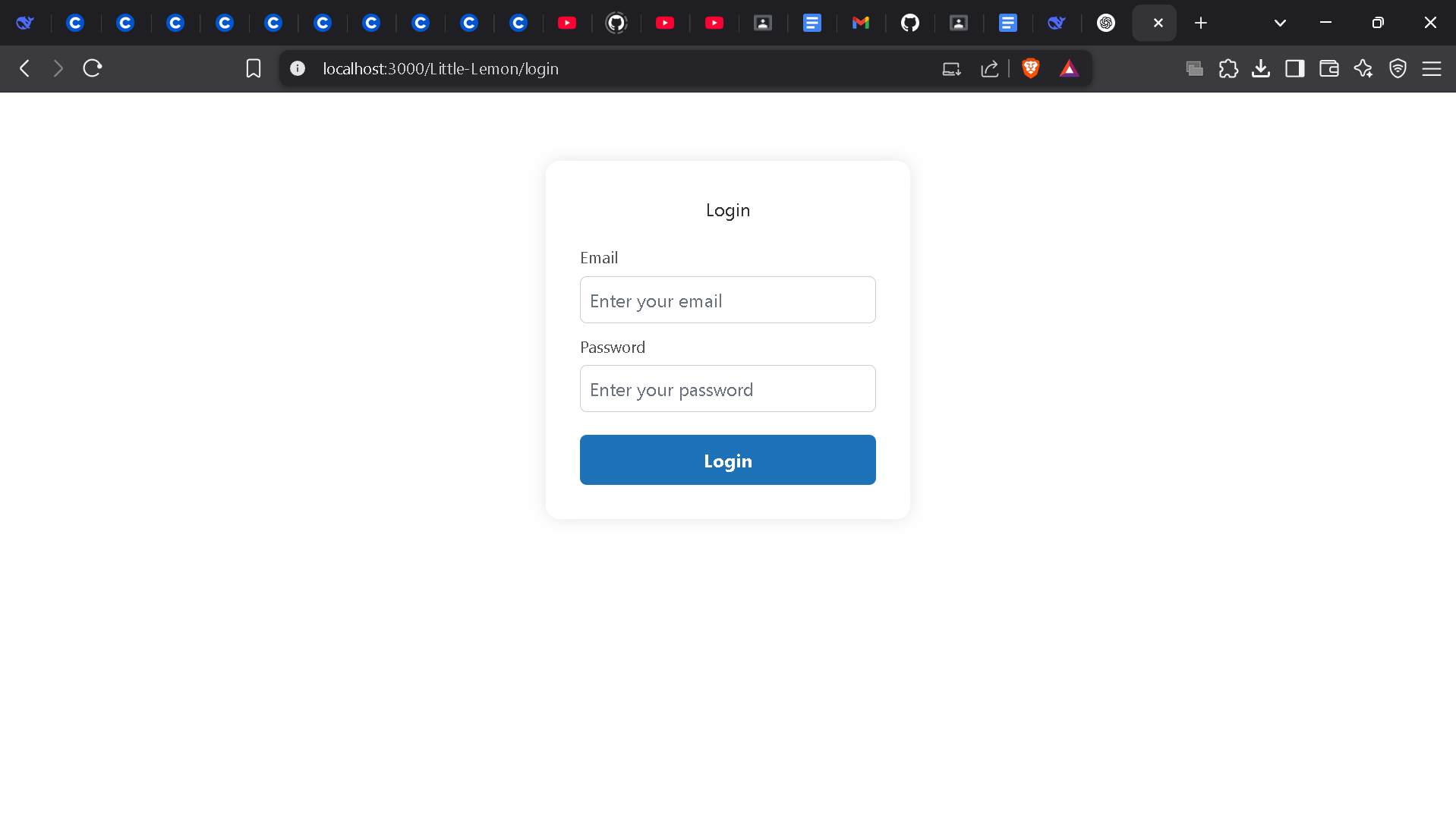
</div>

);

}

export default LoginPage;

## **Output:**

G. login page output:  


## **Conclusion:**

In this project, we built a multi-page **Little Lemon** restaurant website using **React**. The app allows users to browse the **menu**, add items to the **cart**, learn about the restaurant, and manage accounts through **registration** and **login**.

Key features include:

* **Home Page**: A welcome message.
* **Menu/Product Page**: Displays dishes with prices.
* **Cart Page**: Shows selected items (can be expanded later).
* **About Us Page**: Details about the restaurant.
* **Contact Us Page**: Displays contact info.
* **Registration Page**: Users can create an account.
* **Login Page**: Users log in with their credentials.

Using **React**, **React Router**, and **localStorage**, we created an interactive website that can be further enhanced with cart features, state management, and backend integration.