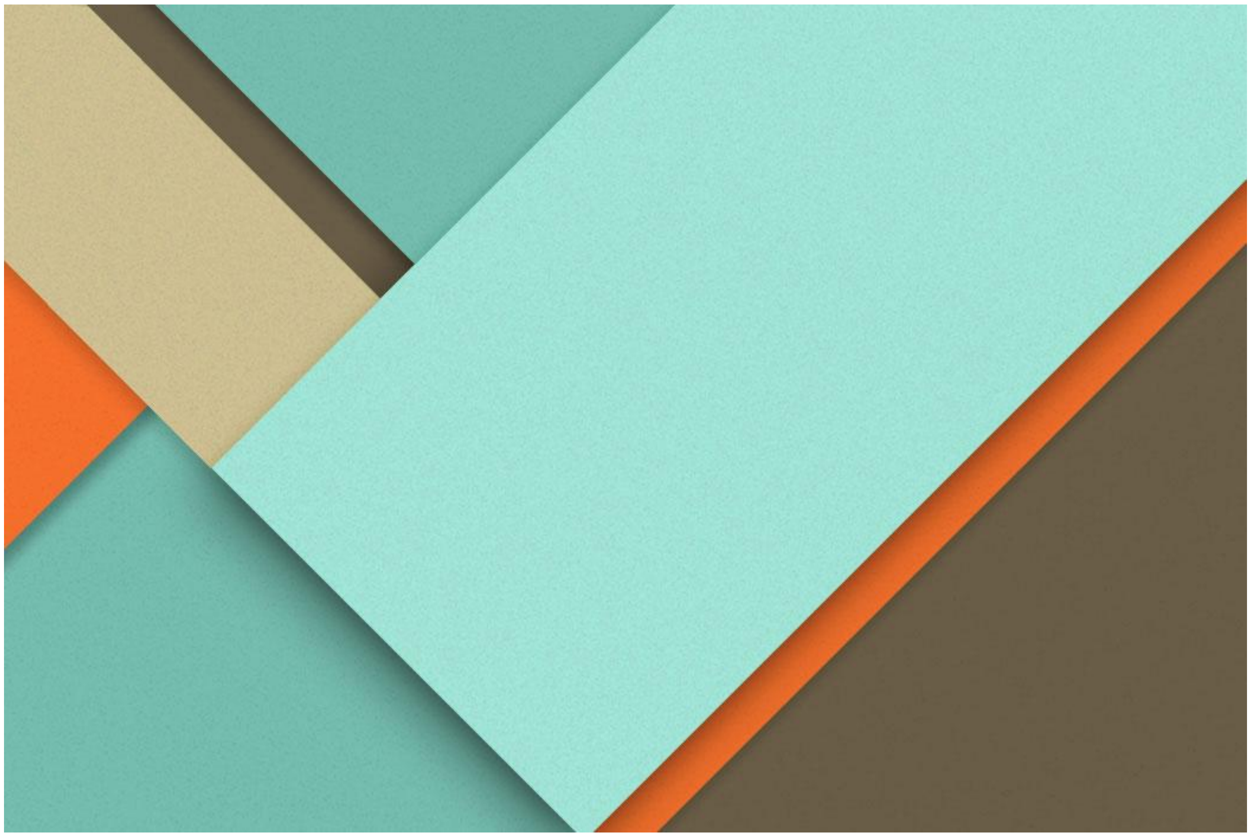


FEE - CS186

25.10.2023

—

Student Name:
AAKASHDEEP SINGH SEDHA
2110990007
G-5

Faculty name:
Lavish Arora

Overview

This project represents a culmination of my efforts to explore and apply front-end technologies to create impactful user experiences.

Goals

1. **Enhance User Experience:** The primary goal of this project is to improve the overall user experience by creating an intuitive, visually appealing, and efficient e-commerce platform.

2. **Increase Sales and Conversion Rates**: To increase the client's online sales and conversion rates by providing a feature-rich and user-friendly e-commerce platform that encourages purchases.

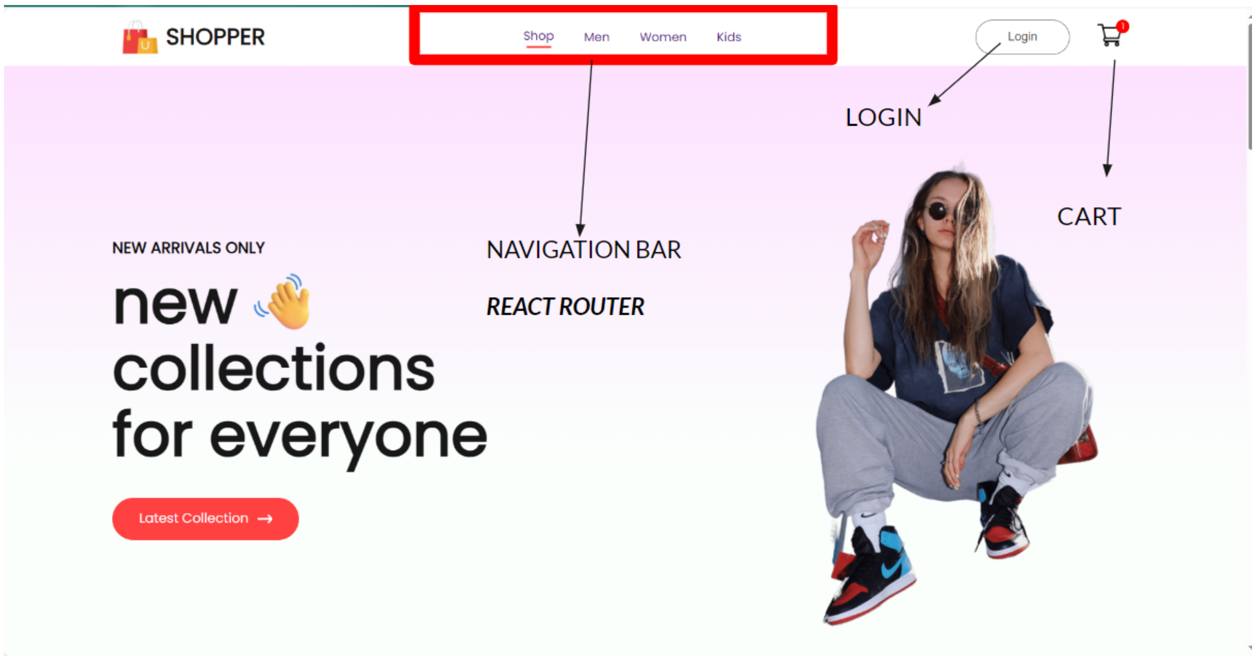
3. **Implement Dynamic Rendering:** Implement dynamic rendering techniques for product cards and pages to engage users and showcase products more effectively.

4. **Integrate Shopping Cart Functionality:** Integrate shopping cart functionality using Redux to enable users to add, remove, and manage items in their cart seamlessly.

Specifications

SHOPPER

Project Name: Shopper



Functional requirements

1. Product Listings:

Display product cards with dynamic rendering.

Sort and filter products by category, price, and other relevant criteria.

Implement a search feature with autocomplete functionality.

2. Display detailed product information:

Include product images, descriptions, pricing, and customer reviews.

Allow users to select product variations (e.g., size, color).

3. **Shopping Cart:**

Implement a shopping cart using Redux for user-friendly item management.

Allow users to add, remove, or update cart items.

Display the cart's total price and items count.

Technology Stack:

Front-end Development:

- Utilize the power of HTML, CSS, and JavaScript to create the foundation of the user interface.

- Implement React, a JavaScript library, to build interactive and dynamic web pages that enhance user engagement and interactivity.

- Employ responsive design principles, ensuring the website is fully accessible on various devices.

State Management:

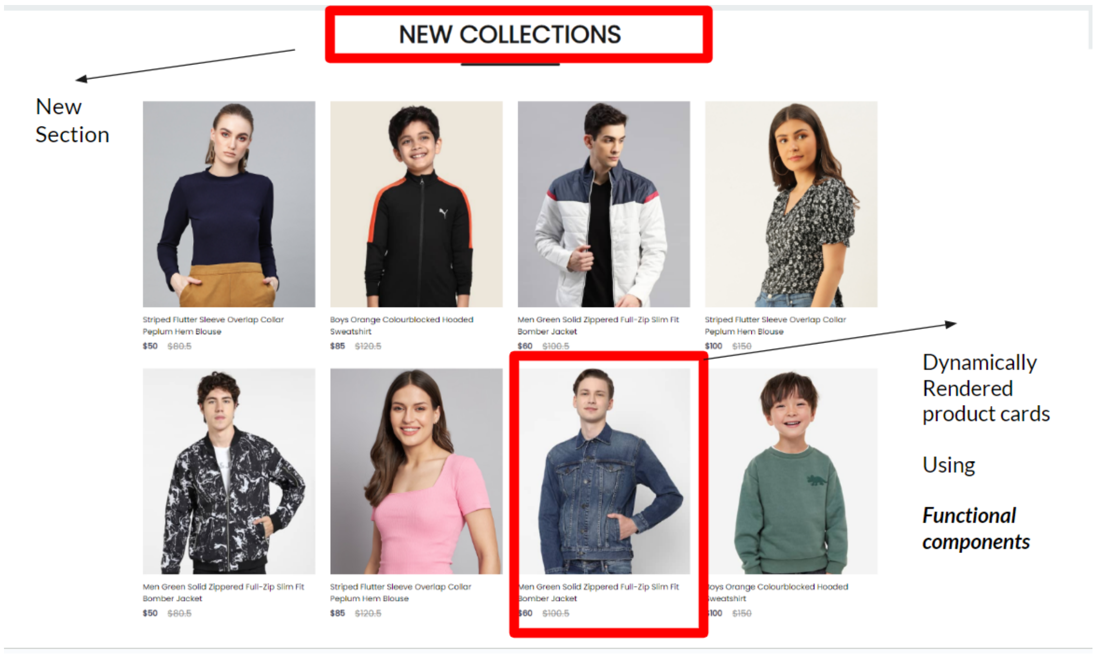
- Utilize Redux, a robust state management library, to efficiently manage the application's state and ensure seamless data flow between components.

- Maintain a centralized and predictable state, enhancing data consistency and simplifying debugging and testing processes.

E-commerce Platform (MongoDB):

- Employ MongoDB as the back-end database to store product data, user information, and order details.

Functionality 1: Dynamic Product Card Rendering


NEW COLLECTIONS
New Section
Striped Flutter Sleeve Overlap Collar Peplum Hem Blouse
$50 ~~$80.5~~
Boys Orange Colourblocked Hooded Sweatshirt
$85 ~~$120.5~~
Men Green Solid Zippered Full-Zip Slim Fit Bomber Jacket
$60 ~~$100.5~~
Striped Flutter Sleeve Overlap Collar Peplum Hem Blouse
$100 ~~$150~~
Men Green Solid Zippered Full-Zip Slim Fit Bomber Jacket
$50 ~~$80.5~~
Striped Flutter Sleeve Overlap Collar Peplum Hem Blouse
$85 ~~$120.5~~
Men Green Solid Zippered Full-Zip Slim Fit Bomber Jacket
$60 ~~$100.5~~
Boys Orange Colourblocked Hooded Sweatshirt
$100 ~~$150~~
Dynamically Rendered product cards
Using
Functional components

How to Implement:

In React-based e-commerce projects, dynamic product card rendering is typically implemented through the following best practices:

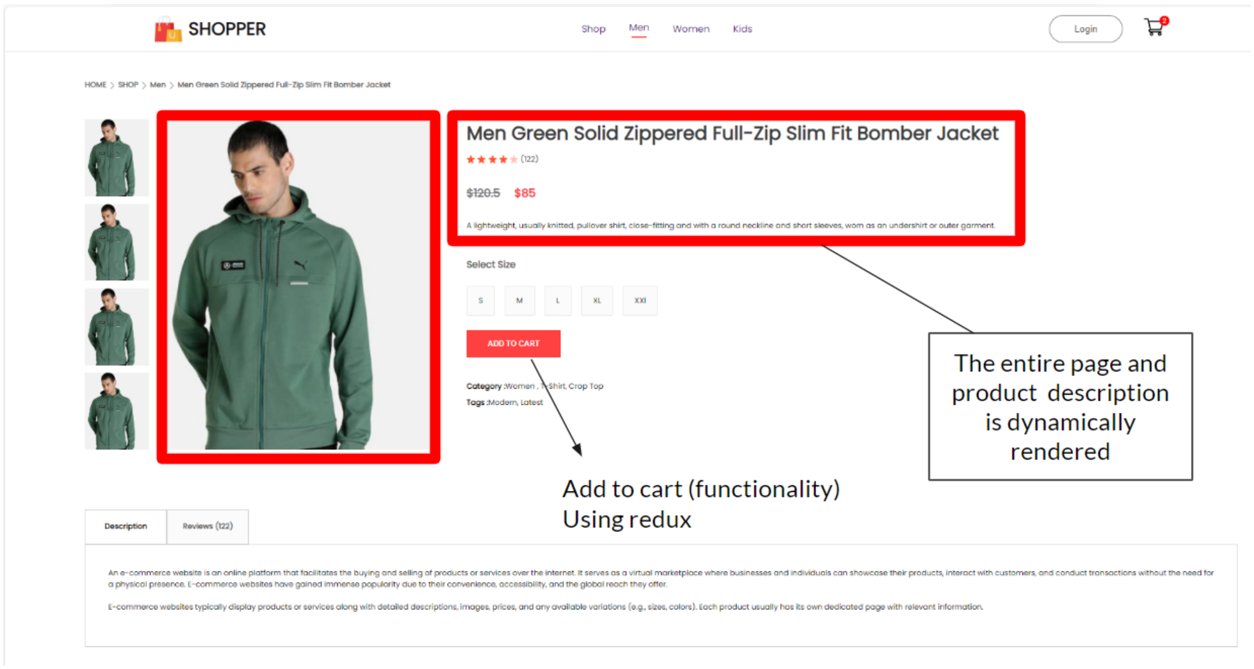
- **Component-Based Architecture: Create reusable and modular React components for product cards. Each card component should receive data as props, making it easy to render dynamic content.**

- **State Management: Utilize state management libraries like Redux or React's built-in state to manage the product card's dynamic data, such as images, prices, and descriptions.**

- **Conditional Rendering: Implement conditional rendering within the product card component to display additional information or interactable elements upon user interactions, such as hovering or clicking on the card.**

- **CSS Transitions: Use CSS transitions and animations to provide smooth and visually appealing effects when users interact with product cards, such as transitioning images or displaying product details.**

Functionality 2: Dynamic Product Page Rendering


SHOPPER
Shop
Men
Women
Kids
Login
HOME > SHOP > Men > Men Green Solid Zippered Full-Zip Slim Fit Bomber Jacket
Select Size
S
M
L
XL
XXl
ADD TO CART
The entire page and product description is dynamically rendered
Add to cart (functionality) Using redux
Description
Reviews (122)

Men Green Solid Zippered Full-Zip Slim Fit Bomber Jacket

★★★★☆ (122)

~~$120.5~~ $85

A lightweight, usually knitted, pullover shirt, close-fitting and with a round neckline and short sleeves, worn as an undershirt or outer garment.

Category :Women , T-Shirt, Crop Top

Tags :Modern, Latest

An e-commerce website is an online platform that facilitates the buying and selling of products or services over the internet. It serves as a virtual marketplace where businesses and individuals can showcase their products, interact with customers, and conduct transactions without the need for a physical presence. E-commerce websites have gained immense popularity due to their convenience, accessibility, and the global reach they offer.

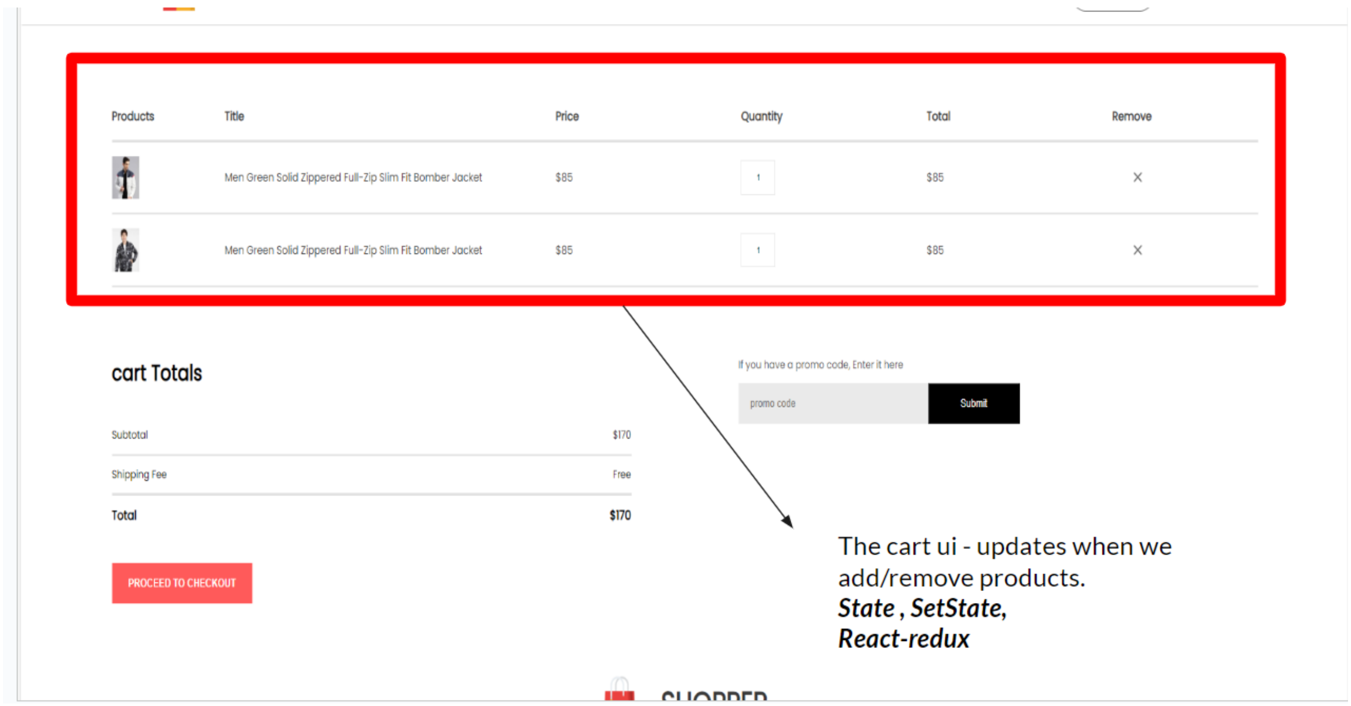
E-commerce websites typically display products or services along with detailed descriptions, images, prices, and any available variations (e.g., sizes, colors). Each product usually has its own dedicated page with relevant information.

How to Implement:

- **Component-Based Architecture: In a React-based e-commerce project, dynamic product page rendering begins with a component-based approach. Each product page is typically represented by a dedicated component. These components are designed to be reusable and adaptable, accepting dynamic data as props.**

- **Conditional Rendering: Conditional rendering is a key technique to dynamically update product pages. Components for product details, images, and descriptions should be structured to display content based on user interactions. For example, additional product information can be revealed when a user clicks on a specific section of the page.**

Functionality 3:Cart functionality



Products	Title	Price	Quantity	Total	Remove
	Men Green Solid Zippered Full-Zip Slim Fit Bomber Jacket	$85	1	$85	×
	Men Green Solid Zippered Full-Zip Slim Fit Bomber Jacket	$85	1	$85	×

cart Totals

Subtotal	$170
Shipping Fee	Free
Total	**$170**

The cart ui - updates when we add/remove products.
***State* , *SetState*,**
React-redux

Procedure: Adding an Item to the Cart

- **Add to Cart Button: On the product page, an "Add to Cart" button is displayed for each product.**

- **User Interaction: When a user clicks the "Add to Cart" button, it triggers a JavaScript function that initiates the process of adding the selected item to the shopping cart.**

- **Redux Action: A Redux action is dispatched, containing information about the selected product, such as its unique identifier, name, price, and quantity.**

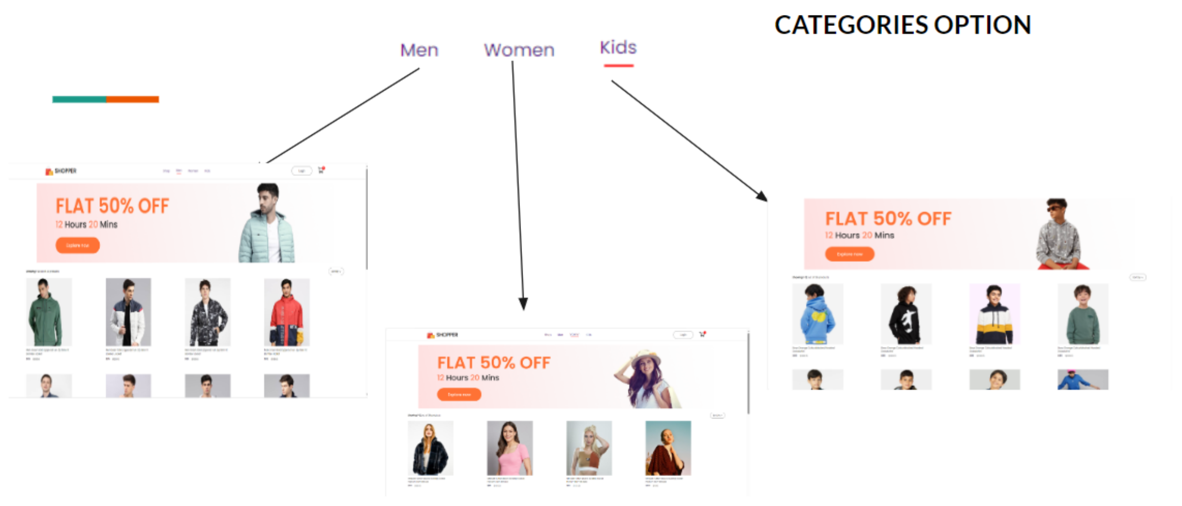
- **Redux Reducer: The dispatched action is processed by a specific Redux reducer dedicated to cart management. This reducer updates the cart state by adding the selected item.**

- **Shopping Cart Component: A shopping cart component is connected to the Redux store and reflects the updated cart state. It displays the added item in the cart's interface, showing the product name, quantity, and total price.**

- **Visual Feedback: After adding an item to the cart, users typically receive visual feedback, such as a confirmation message or an animated icon indicating that the product has been successfully added to the cart.**

- **Cart Management: Users can interact with the cart component to view its contents, adjust quantities, remove items, or proceed to the checkout process. The cart's state is continuously updated based on user interactions.**

Functionality 4 : Routing across different pages



React Routing is a fundamental concept in React applications that allows you to create multi-page web experiences. It's all about managing the content displayed to users as they move between different parts of your web application.

- **Route Mapping: You define a mapping between URLs and the content (React components) you want to display when a user visits a specific URL. For example, you might say that when a user goes to /products, they should see the "Products" page.**

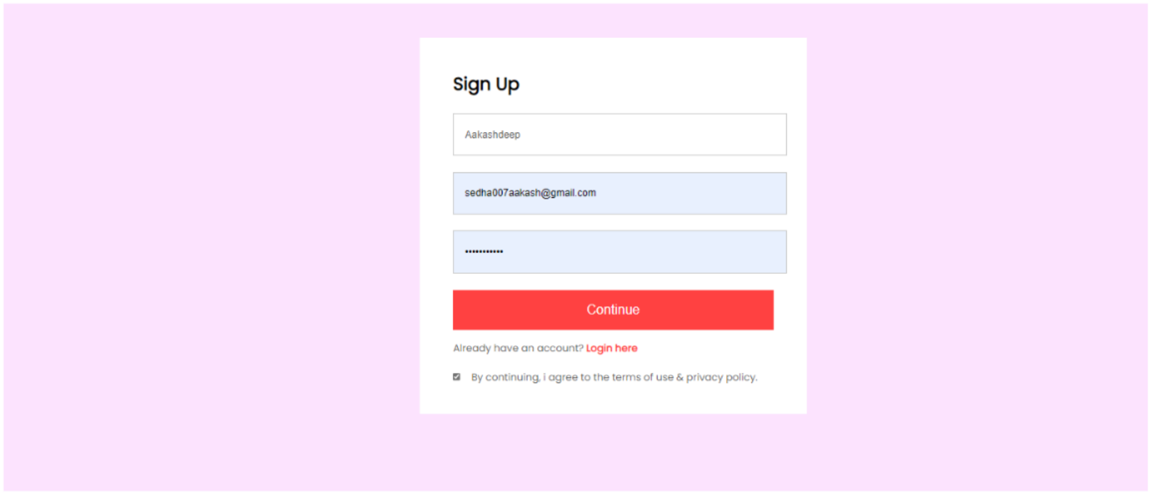
- **Links: You create links or buttons in your app that, when clicked, change the URL in the browser. React Routing intercepts these clicks and displays the associated content without actually refreshing the entire page.**

- **Dynamic Content: You can make your URLs dynamic. For example, /products/123 might display details for product number 123. React Routing allows you to capture this dynamic part of the URL and pass it to the component that needs it.**

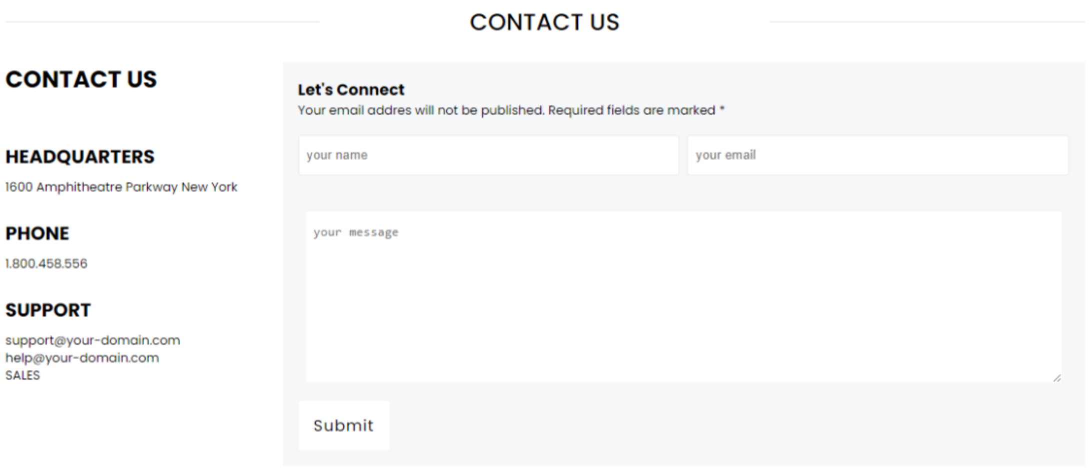
- **Fallback Routes: You can set up "fallback" or "not found" routes. If a user tries to visit a URL that doesn't match any of your defined routes, you can show a friendly "404 - Page Not Found" page.**

Some other pages

SIGNUP / LOGIN PAGE


Sign Up
Aakashdeep
sedha007aakash@gmail.com
•••••••••••
Continue
Already have an account? Login here
By continuing, i agree to the terms of use & privacy policy.

CONTACT US PAGE



CONTACT US

HEADQUARTERS

1600 Amphitheatre Parkway New York

PHONE

1.800.458.556

SUPPORT

support@your-domain.com
help@your-domain.com
SALES

Let's Connect

Your email addres will not be published. Required fields are marked *

Conclusion

In conclusion, e-commerce project stands as a testament to the seamless integration of modern web technologies and best practices. Our commitment to enhancing user experiences, increasing sales and conversions, and delivering dynamic and responsive features has resulted in a user-friendly, visually appealing, and efficient platform. Through dynamic product page rendering, effective shopping cart functionality with Redux, responsive design, and comprehensive routing, we have successfully achieved our project goals.

We extend our heartfelt gratitude and special thanks to Lavish Arora, whose guidance, mentorship, and support were instrumental in the success of this project. Your invaluable insights and expertise have been a driving force throughout the journey.

With a keen focus on performance, security, and compliance, our e-commerce platform is poised for growth and poised to meet the evolving needs of our users. It is our hope that this project serves as a foundation for creating exceptional online shopping experiences, fostering user engagement, and ultimately, contributing to the success of our clients and their businesses.