Vellore Institute Of Technology, Vellore

Web Programming (BCSE203E)
Slot: L19+L20+L29+L30

Mini project

GALAXY TECH: A TECH-SELLING PLATFORM

Akshath Rajkumar (23BAI0059)

Vishwajith P (23BCI0128)

INDEX

Abstract

In response to the growing demand for tech items in our increasingly digital world, this project endeavours to create an interactive online platform dedicated to offering a wide array of cutting-edge gadgets, electronics, and accessories. Designed to cater to tech enthusiasts of all backgrounds, our website emphasizes a dynamic and user-friendly experience, prioritizing seamless navigation and visually appealing design. Customers can expect detailed product descriptions, high-quality images, and secure payment options, ensuring trust and confidence in their purchasing decisions. Employing responsive design principles, our platform guarantees compatibility across various devices and screen sizes. Ultimately, our goal is to showcase our expertise in web programming while contributing to the digital marketplace, offering users a hassle-free and enjoyable shopping experience for all their tech needs.

1. Introduction

In our rapidly evolving digital landscape, the demand for technology products has reached unprecedented levels. To address this surge in demand and cater to the needs of tech enthusiasts, our project aims to develop an innovative online platform dedicated to the sale of cutting-edge gadgets, electronics, and accessories. This platform will serve as a hub where users can explore a diverse range of tech items, make informed purchasing decisions, and enjoy a seamless shopping experience.

1.1 Project Objectives

- **Create a Dynamic Online Platform:** Develop a user-friendly website with modern web development techniques such as React JS to provide an engaging and interactive experience for users browsing tech products.
- **Offer a Wide Range of Tech Products:** Curate a comprehensive selection of tech items, including gadgets, electronics, and accessories, to cater to the diverse needs and interests of our target audience.
- **Ensure user-friendliness:** Implement easy and efficient options and provide detailed product descriptions and high-quality images to make it simple and clear to use.
- **Prioritize Responsive Design:** Ensure compatibility across various devices and screen sizes by incorporating responsive design principles, enhancing accessibility and usability for all users.

1.2 Project Goals

- **Enhanced User Experience:** Provide customers with intuitive navigation, visually appealing design, and seamless functionality to create an enjoyable shopping experience.
- **Effective Search Functionality:** Implement advanced search features such as card scrollbars to help users quickly find desired products, enhancing user satisfaction and encouraging repeat visits.
- **Contribution to the Digital Marketplace:** Showcase proficiency in web programming while making a meaningful contribution to the digital marketplace, offering a valuable resource for tech enthusiasts worldwide.

2. Technologies Used

2.1 Innovative Web Development with React.js and Vite

Our project leverages React.js and Vite to craft dynamic web applications. React.js facilitates modular development, while Vite accelerates our workflow with lightning-fast build times and instant hot module replacement.

With React, we embrace a modular approach, breaking down complex user interfaces into reusable components. This not only enhances code maintainability but also streamlines development workflows, enabling rapid iteration and seamless collaboration.

Complementing React.js, we harness the power of Vite—a next-generation build tool and development server. Vite reimagines the traditional build process, leveraging modern ES module-based bundling to deliver lightning-fast development experiences. With Vite, we enjoy instantaneous hot module replacement (HMR), eliminating tedious rebuilds and accelerating feedback loops. This efficiency extends to production builds, where Vite's optimized bundling strategies ensure high-performance web applications.

2.2 CSS and JavaScript: Crafting Immersive User Experiences

In addition to React.js and Vite, our project incorporates the cornerstone technologies of web development—CSS and JavaScript. Leveraging the expressive capabilities of CSS, we design captivating user interfaces that captivate and delight users. From fluid animations to responsive layouts, CSS empowers us to bring our creative visions to life.

Meanwhile, JavaScript serves as the backbone of our application logic, enabling dynamic interactions and real-time updates. Whether it's handling user input, or orchestrating complex workflows, JavaScript powers the functionality that underpins our web applications.

3. Project Requirements

3.1 User Authentication and Authorization:

- Implement user authentication features, including registration, login, and password management.
- Ensure secure access control mechanisms to protect sensitive user data and functionalities.

3.2 Product Catalog and Management:

- Create a comprehensive product catalogue featuring various tech items, including gadgets, electronics, and accessories.
- Implement CRUD (Create, Read, Update, Delete) operations for managing products, including adding new products, updating existing ones, and removing outdated items.

3.3 User Interface and Experience:

- Design an intuitive and visually appealing user interface that enhances the browsing and shopping experience.
- Ensure responsive design principles to optimize compatibility across different devices and screen sizes.

3.4 Shopping Cart and Checkout Process:

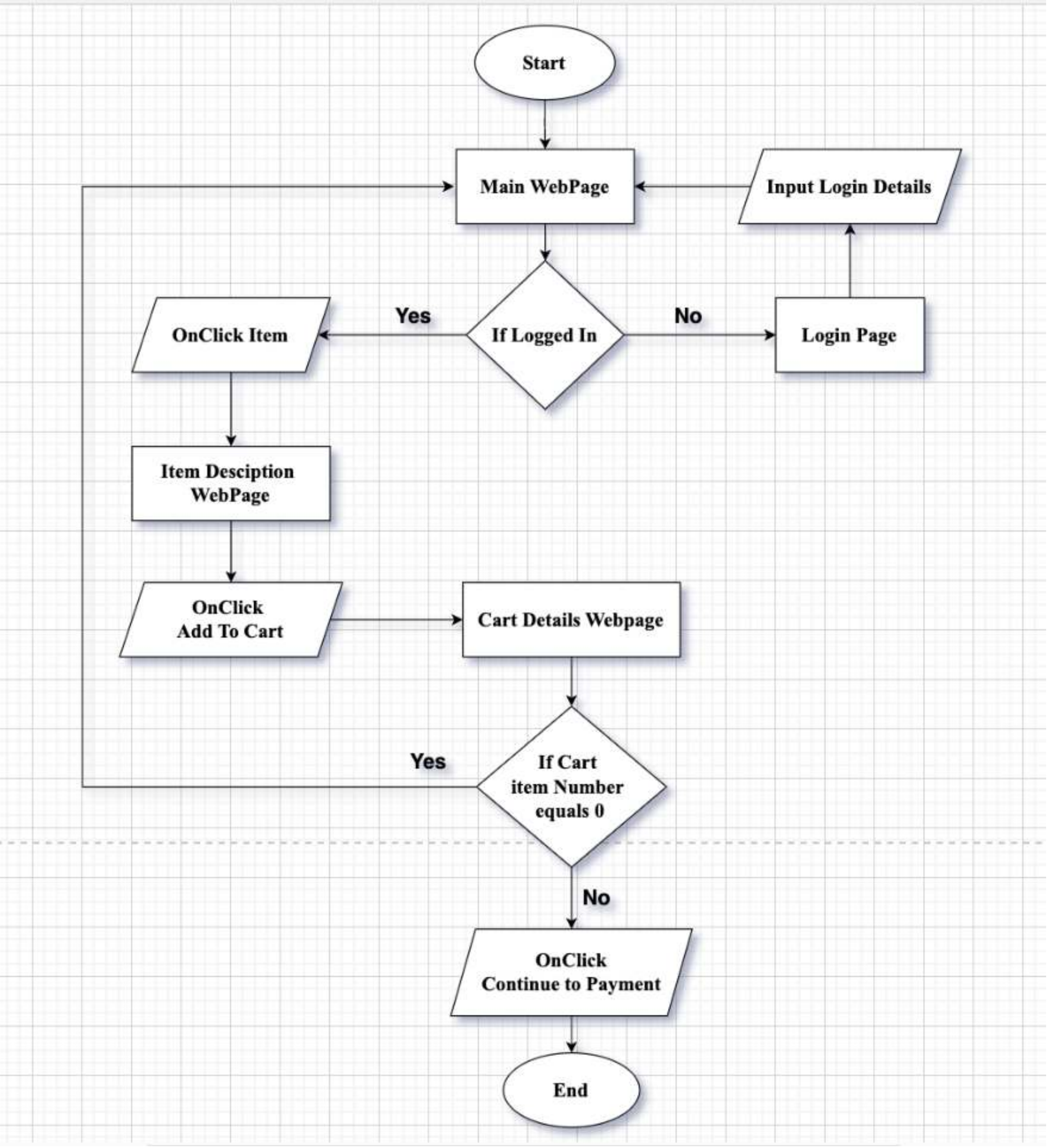
- Develop a shopping cart feature that allows users to add products, update quantities, and remove items before proceeding to checkout.
- Implement a secure and seamless checkout process.

3.5 Performance Optimization:

- Optimize website performance for fast loading times and smooth navigation, minimizing page load times and latency.
- Implement caching mechanisms, lazy loading techniques, and image optimization to improve overall website performance.

4. System Design

4.1 User Interface Flowchart



4.2 User Interface Design (UI):

4.2.1 Clean and Intuitive Layout:

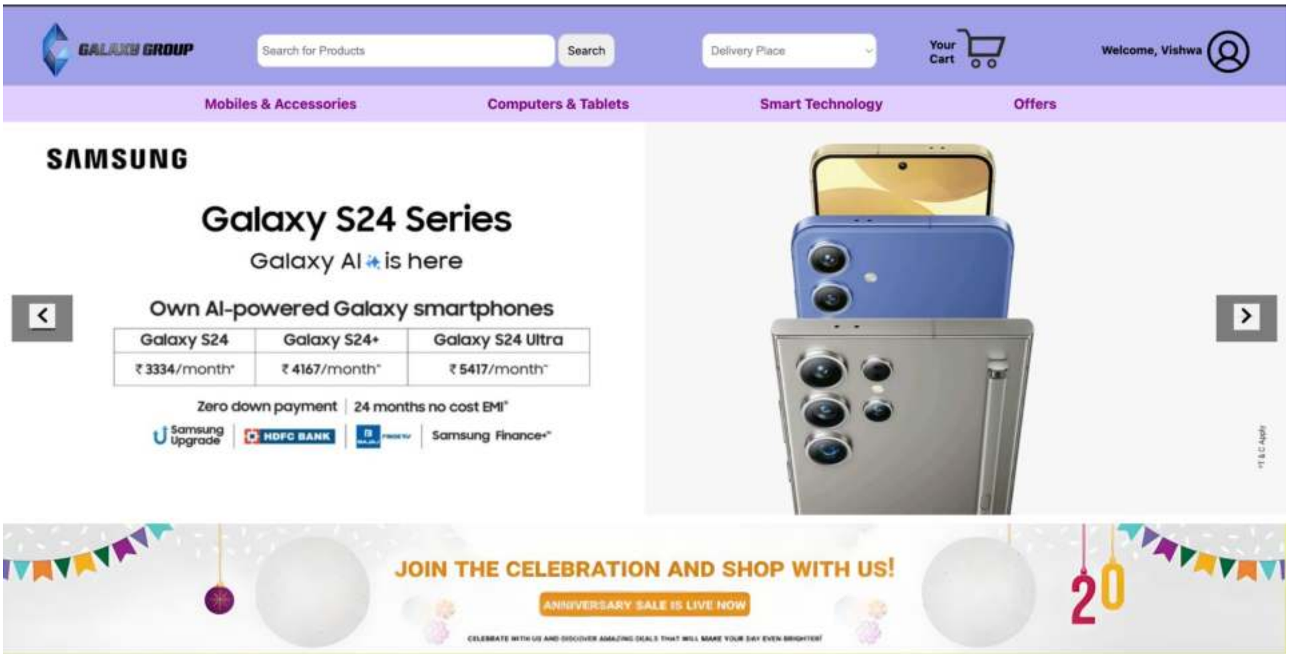
- Maintain a clean and clutter-free layout to avoid overwhelming users.
- Organize content logically with clear navigation menus and categories for easy exploration.

4.2.2 Consistent Branding:

- Maintain consistent branding elements such as colours, typography, and logo placement throughout the interface.
- Reinforce brand identity through visual elements to create a cohesive and memorable experience.

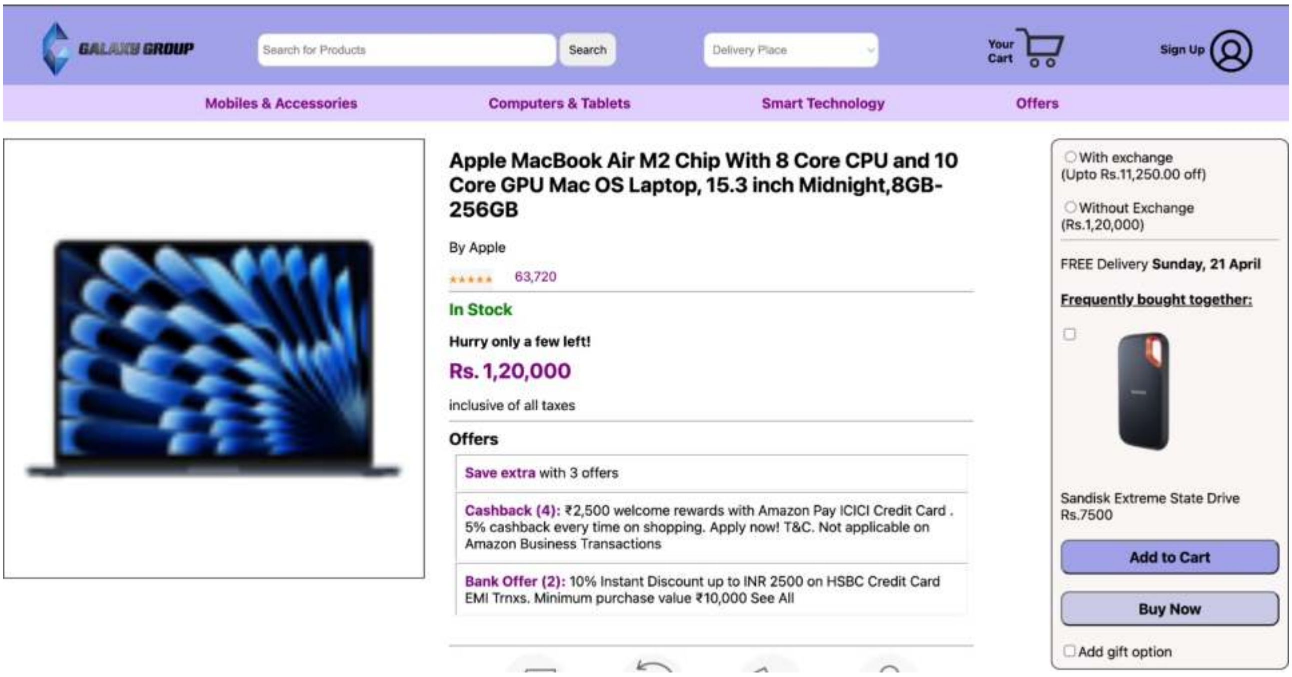
4.2.3 Product Showcase:

- Showcase products prominently with high-quality images and descriptive metadata.
- Allow users to view product details, specifications, and customer reviews to make informed purchase decisions.



Galaxy S24	Galaxy S24+	Galaxy S24 Ultra
₹3334/month*	₹4167/month*	₹5417/month*

Fig (4a)- Main React WebPage



Apple MacBook Air M2 Chip With 8 Core CPU and 10 Core GPU Mac OS Laptop, 15.3 inch Midnight,8GB-256GB

By Apple

★★★★★ 63,720

In Stock

Hurry only a few left!

Rs. 1,20,000

inclusive of all taxes

Offers

Save extra with 3 offers

Cashback (4): ₹2,500 welcome rewards with Amazon Pay ICICI Credit Card . 5% cashback every time on shopping. Apply now! T&C. Not applicable on Amazon Business Transactions

Bank Offer (2): 10% Instant Discount up to INR 2500 on HSBC Credit Card EMI Trnxs. Minimum purchase value ₹10,000 See All

○ With exchange (Upto Rs.11,250.00 off)

○ Without Exchange (Rs.1,20,000)

FREE Delivery **Sunday, 21 April**

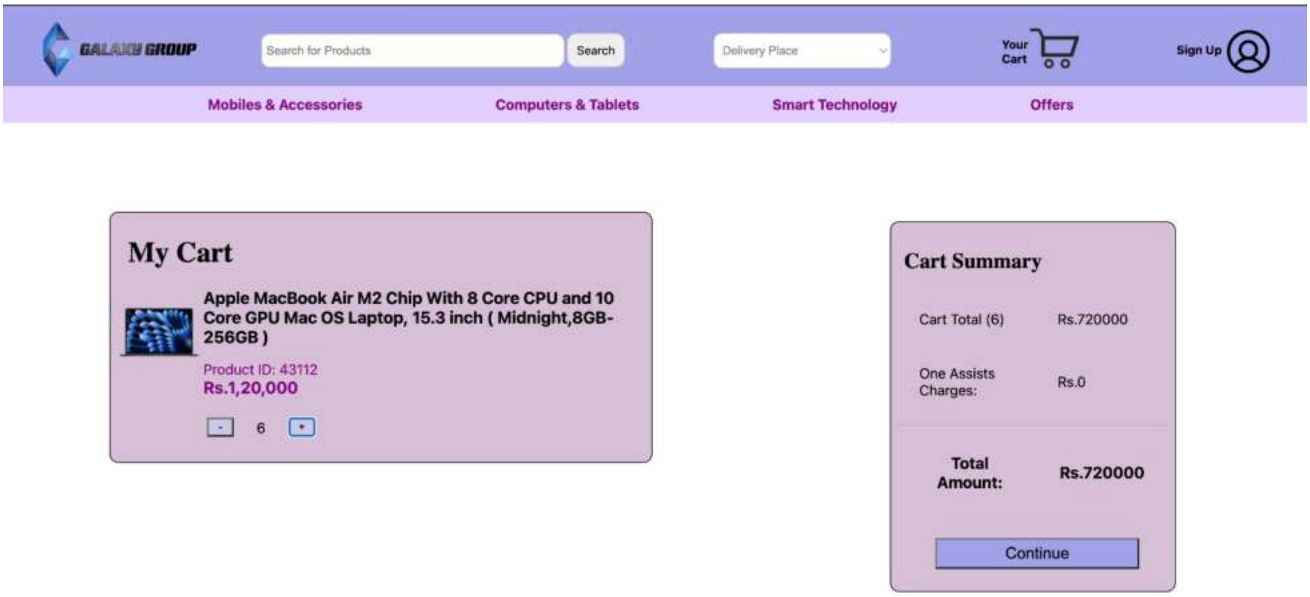
Frequently bought together:

□

Sandisk Extreme State Drive Rs.7500

□ Add gift option

Fig (4b)- Item Details WebPage



My Cart

Apple MacBook Air M2 Chip With 8 Core CPU and 10 Core GPU Mac OS Laptop, 15.3 inch (Midnight,8GB-256GB)

Product ID: 43112

Rs.1,20,000

[-] 6 [+]

Cart Summary

Cart Total (6)	Rs.720000
One Assists Charges:	Rs.0
Total Amount:	Rs.720000

Fig (4c)- Cart Webpage

5. Project Implementation

5.1 Coding Techniques

The various techniques we used in the implementation of the mini project include-

5.1.1 Modularization:

- We broke down our code into smaller, reusable modules, each responsible for a specific functionality or feature.
- Encapsulated related code within modules to improve code organization, maintainability, and reusability.
- Utilized ES6 module syntax (import and export) in JavaScript to define and import modules.

5.1.2 DRY Principle (Don't Repeat Yourself):

- Avoided duplicating code by extracting common functionality into reusable functions, classes, or modules.
- Encapsulated repetitive logic within functions or utility modules and reuse them across different parts of your codebase.
- Followed the single responsibility principle (SRP) to ensure that each function or module has a single, well-defined purpose.

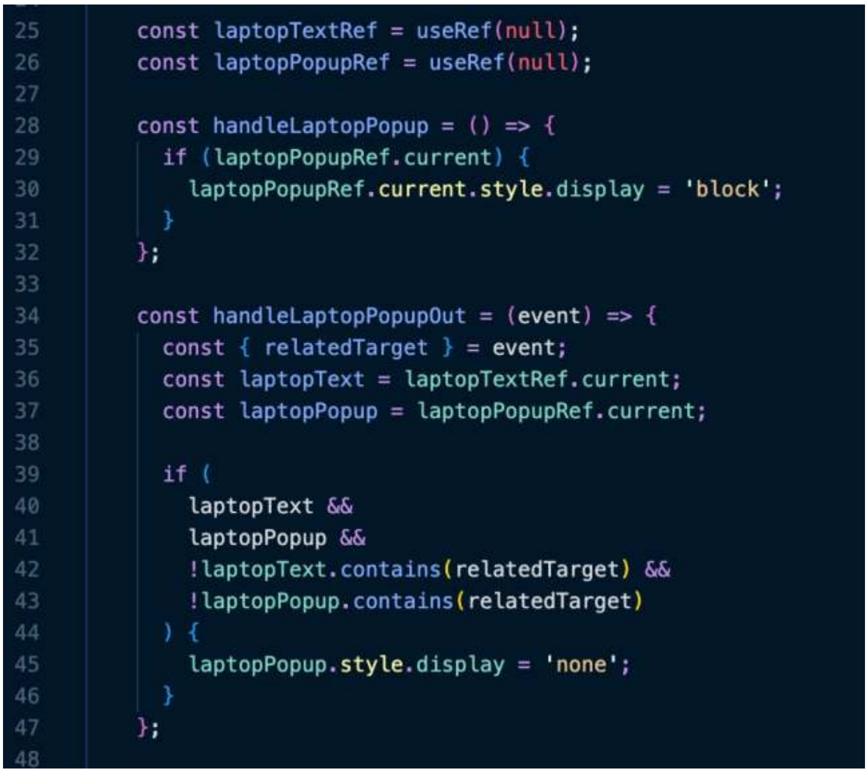
5.1.3 Lazy Loading Techniques:

- Improved website performance by deferring the loading of images that are not immediately visible in the viewport.
- Instead of loading all images when the page loads, images are loaded dynamically as the user scrolls down the page, ensuring that only the images that are currently visible are loaded.
- This technique helps reduce bandwidth usage and speed up page loading times, particularly for web pages with a large number of images or media assets.

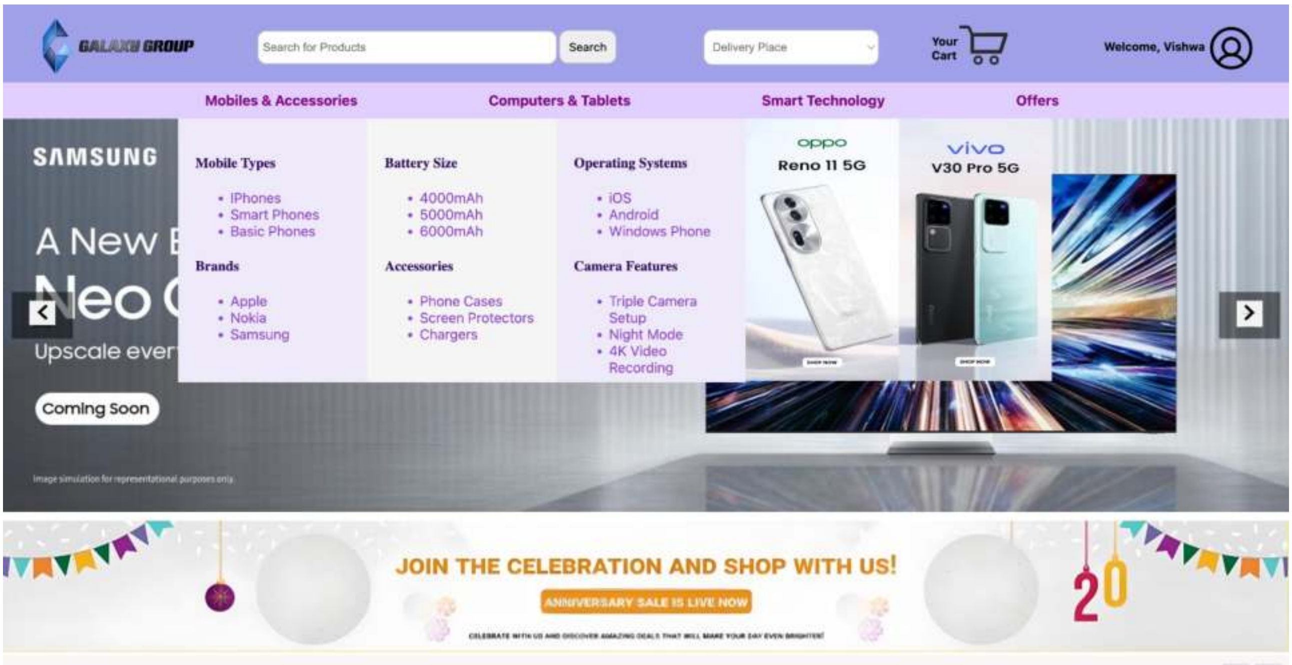
5.2 Key functionalities

5.2.1 Pop-up feature displaying available products

On hovering the mouse over certain texts in the navigation bar, you can get a pop-up display of the various products available under the particular element of the navigation bar.

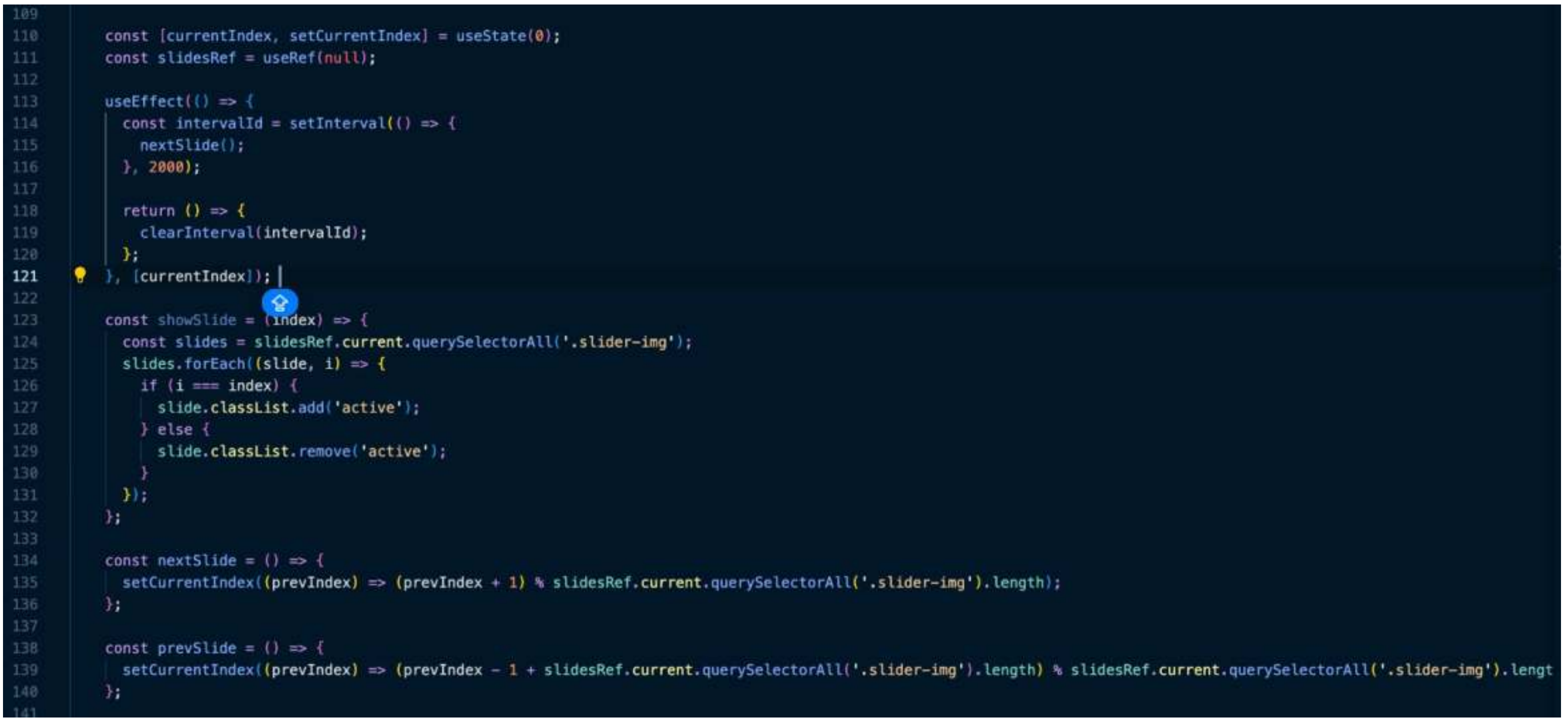


```
const laptopTextRef = useRef(null);
const laptopPopupRef = useRef(null);

const handleLaptopPopup = () => {
  if (laptopPopupRef.current) {
    laptopPopupRef.current.style.display = 'block';
  }
};

const handleLaptopPopupOut = (event) => {
  const { relatedTarget } = event;
  const laptopText = laptopTextRef.current;
  const laptopPopup = laptopPopupRef.current;

  if (
    laptopText &&
    laptopPopup &&
    !laptopText.contains(relatedTarget) &&
    !laptopPopup.contains(relatedTarget)
  ) {
    laptopPopup.style.display = 'none';
  }
};
```

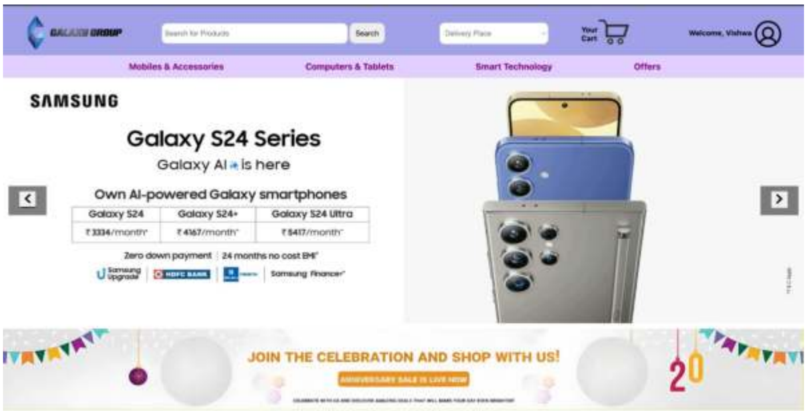


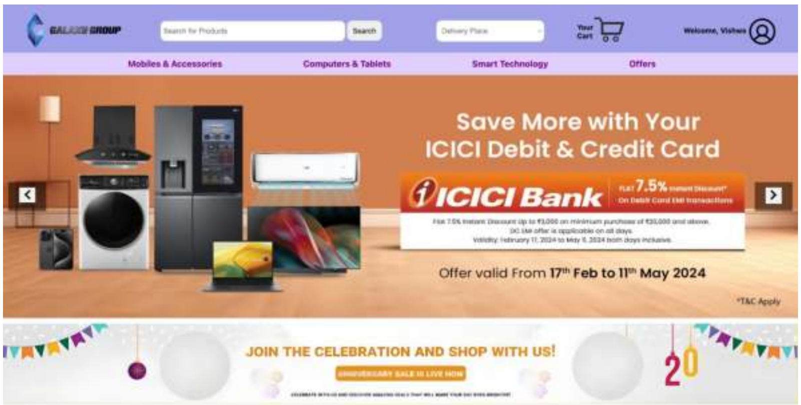
5.2.2 Image slider

To make the webpage visually attractive we included a feature of image slider such the images in the main webpage keep changing periodically. It runs in an infinite loop. The next and previous images can be accessed through the right and left buttons respectively.



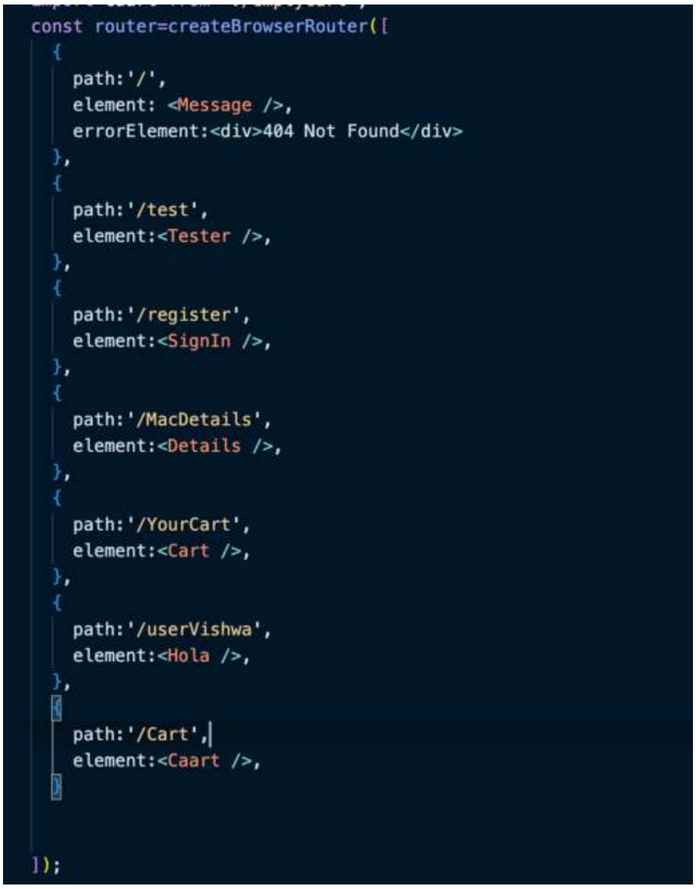
```
  const [currentIndex, setCurrentIndex] = useState(0);
  const slidesRef = useRef(null);

  useEffect(() => {
    const intervalId = setInterval(() => {
      nextSlide();
    }, 2000);

    return () => {
      clearInterval(intervalId);
    };
  }, [currentIndex]);

  const showSlide = (index) => {
    const slides = slidesRef.current.querySelectorAll('.slider-img');
    slides.forEach((slide, i) => {
      if (i === index) {
        slide.classList.add('active');
      } else {
        slide.classList.remove('active');
      }
    });
  };

  const nextSlide = () => {
    setCurrentIndex((prevIndex) => (prevIndex + 1) % slidesRef.current.querySelectorAll('.slider-img').length);
  };

  const prevSlide = () => {
    setCurrentIndex((prevIndex) => (prevIndex - 1 + slidesRef.current.querySelectorAll('.slider-img').length) % slidesRef.current.querySelectorAll('.slider-img').lengt
  };
```



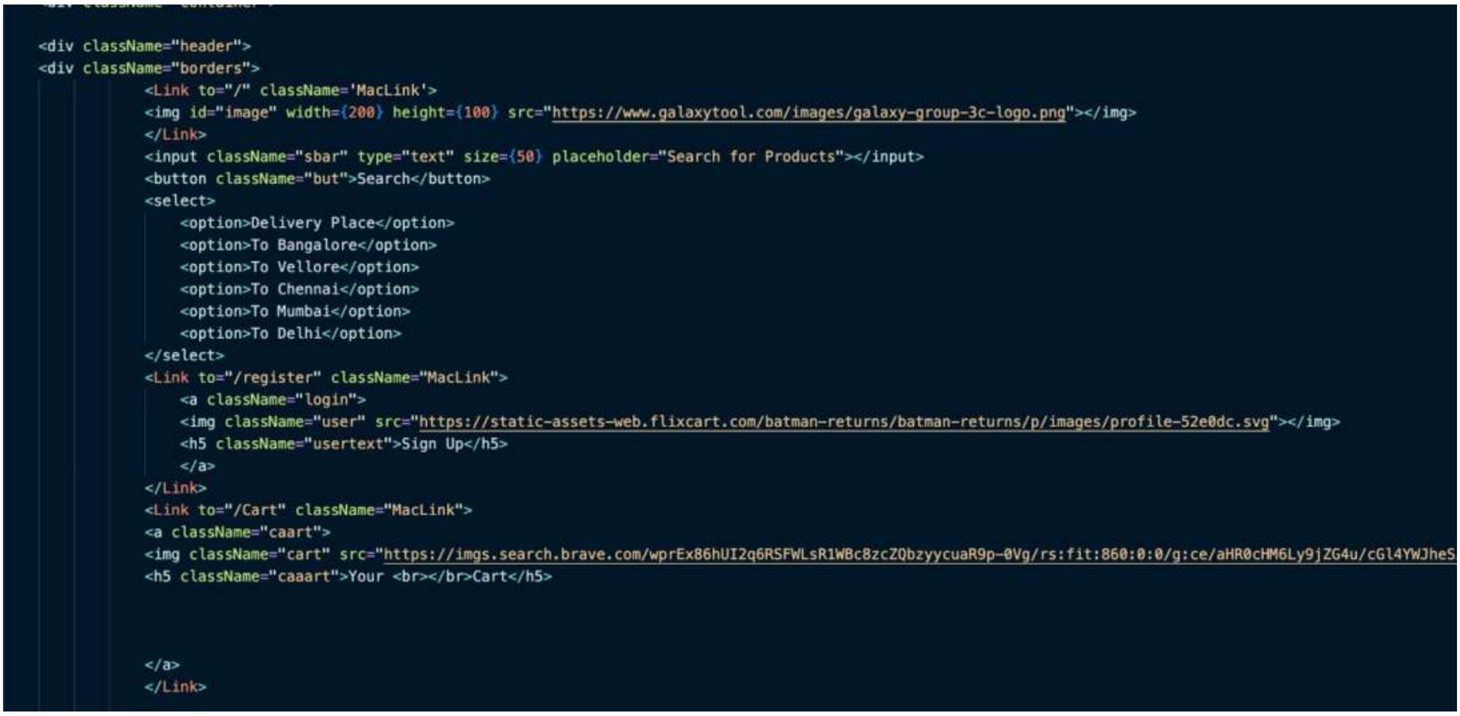


5.2.3 Navigation to various React webpages

Using Router, Link and Navigation we were successfully able to link several React webpages together. This not only increased the functionality of the app but also provided a seamless browsing experience for users as they navigate through different pages of the React web application.



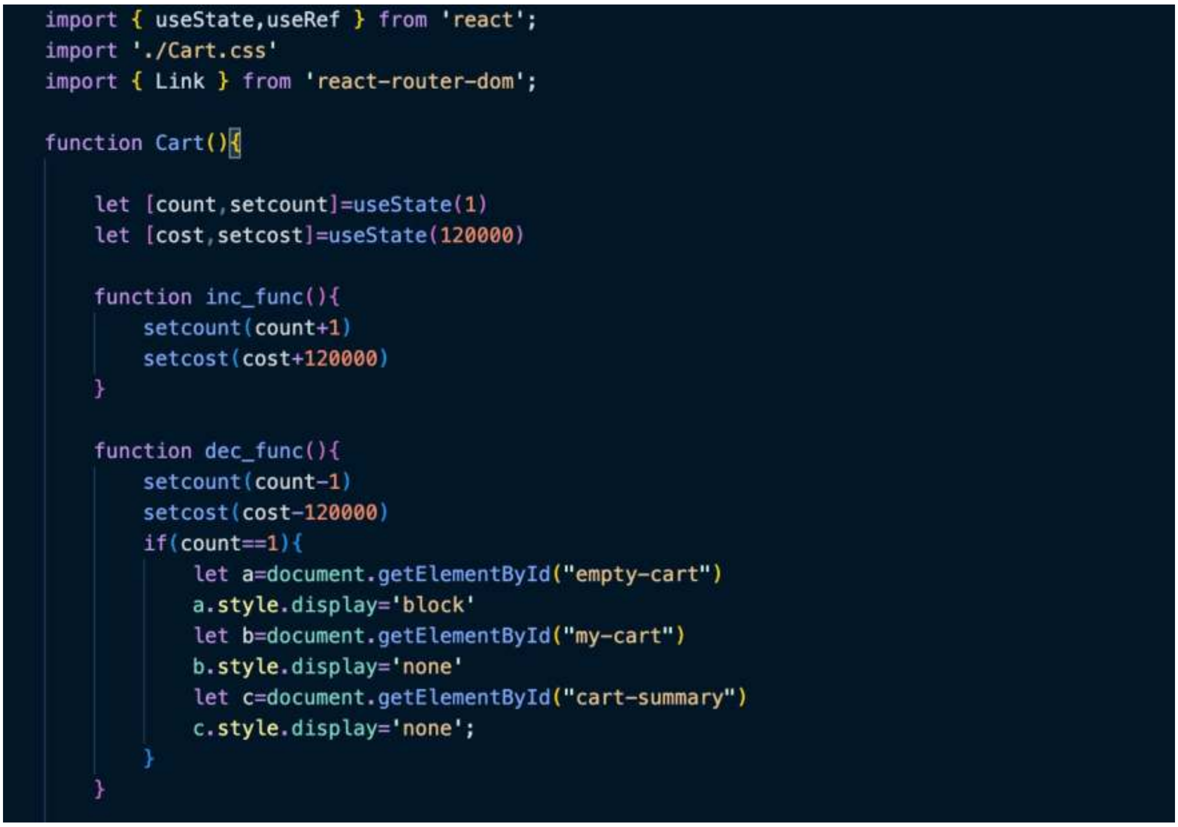
```
const router=createBrowserRouter([
  {
    path:'/',
    element: <Message />,
    errorElement:<div>404 Not Found</div>
  },
  {
    path:'/test',
    element:<Tester />,
  },
  {
    path:'/register',
    element:<SignIn />,
  },
  {
    path:'/MacDetails',
    element:<Details />,
  },
  {
    path:'/YourCart',
    element:<Cart />,
  },
  {
    path:'/userVishwa',
    element:<Hola />,
  },
  {
    path:'/Cart',
    element:<Caart />,
  }


]);
```



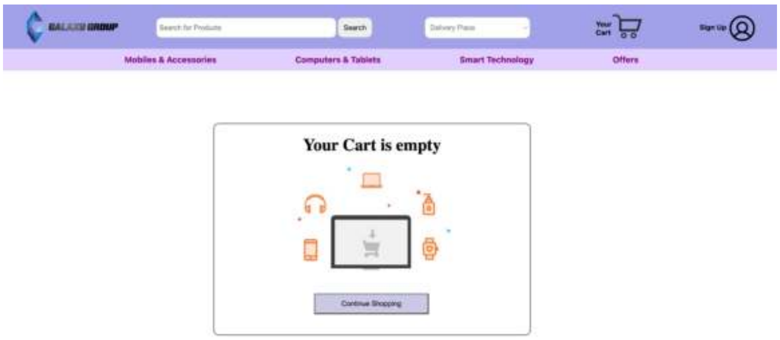
```
<div className="header">
<div className="borders">
            <Link to="/" className='MacLink'>
            <img id="image" width={200} height={100} src="https://www.galaxytool.com/images/galaxy-group-3c-logo.png"></img>
            </Link>
            <input className="sbar" type="text" size={50} placeholder="Search for Products"></input>
            <button className="but">Search</button>
            <select>
                <option>Delivery Place</option>
                <option>To Bangalore</option>
                <option>To Vellore</option>
                <option>To Chennai</option>
                <option>To Mumbai</option>
                <option>To Delhi</option>
            </select>
            <Link to="/register" className="MacLink">
                <a className="login">
                <img className="user" src="https://static-assets-web.flixcart.com/batman-returns/batman-returns/p/images/profile-52e0dc.svg"></img>
                <h5 className="usertext">Sign Up</h5>
                </a>
            </Link>
            <Link to="/Cart" className="MacLink">
            <a className="caart">
            <img className="cart" src="https://imgs.search.brave.com/wprEx86hUI2q6RSFWLsR1WBc8zcZQbzyycuaR9p-0Vg/rs:fit:860:0:0/g:ce/aHR0cHM6Ly9jZG4u/cGl4YWJheS
            <h5 className="caaart">Your <br></br>Cart</h5>



            </a>
            </Link>
```

5.2.4 Dynamically updating the Shopping Cart page

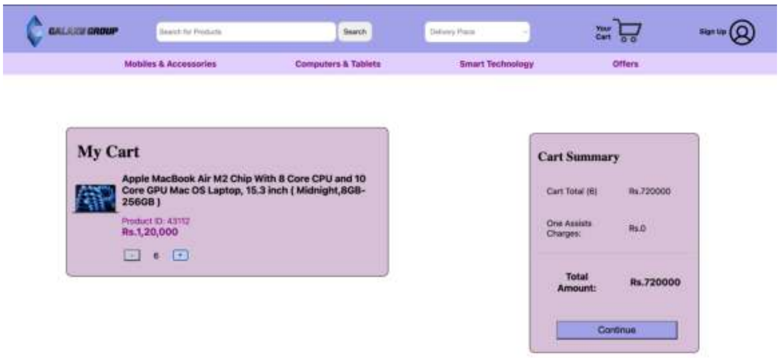
We created a dynamic shopping cart where users can add, and remove the existing items in the cart. Not only that, but the cart shows empty when no order has been placed by the user. The cost and quantity is dynamically updated based on the cart items of the user.



```
import { useState,useRef } from 'react';
import './Cart.css'
import { Link } from 'react-router-dom';

function Cart(){

    let [count,setcount]=useState(1)
    let [cost,setcost]=useState(120000)

    function inc_func(){
        setcount(count+1)
        setcost(cost+120000)
    }

    function dec_func(){
        setcount(count-1)
        setcost(cost-120000)
        if(count==1){
            let a=document.getElementById("empty-cart")
            a.style.display='block'
            let b=document.getElementById("my-cart")
            b.style.display='none'
            let c=document.getElementById("cart-summary")
            c.style.display='none';
        }
    }
```



Your Cart is empty



My Cart

Apple MacBook Air M2 Chip With 8 Core CPU and 10 Core GPU Mac OS Laptop, 15.3 inch (Midnight,8GB-256GB)

Product ID: 43112

Rs.1,20,000

6

Cart Summary

Cart Total (6)	Rs.720000
One Assists Charges:	Rs.0
Total Amount:	Rs.720000

6. Project Deployment

6.1 Version Control:

Our project is hosted on GitHub, utilizing Git for version control. This allows for collaboration among team members and keeps track of changes made to the codebase over time.

6.2 Repository Setup:

We have set up a GitHub repository to host our project. The repository contains all the necessary files and folders required for the application, including source code, assets, and documentation.

6.3 Deployment Workflow:

We utilize GitHub Actions for automated deployment workflows. Whenever changes are pushed to the main branch, GitHub Actions automatically triggers the deployment process, ensuring that the latest version of the application is deployed to the production environment.

6.4 Deployment Status:

The deployment status can be tracked directly from the GitHub repository. We use status badges to indicate the current deployment status, allowing team members to quickly assess the health of the deployment.

7. Future Enhancements

7.1 Enhanced Product Search and Filtering:

- Implement advanced search features such as filtering by price range, brand, category, etc.
- Introduce autocomplete suggestions to help users find products more quickly.

7.2 User Accounts and Profiles:

- Allow users to create accounts to save their preferences, order history, and shopping carts across sessions.
- Implement profile customization options and personalized recommendations based on user preferences and browsing history.

7.3 Social Integration:

- Enable social login options (e.g., Google, Facebook) to streamline the registration and login process.
- Integrate social sharing features to allow users to share products they like with their friends and followers.

7.4 Product Reviews and Ratings:

- Add a feature for users to leave reviews and ratings for products they've purchased.
- Display average ratings and reviews on product pages to help users make informed purchasing decisions.

7.5 Enhanced Checkout Process:

- Streamline the checkout process with features such as guest checkout, saved shipping addresses, and multiple payment options (e.g., credit/debit cards, PayPal, etc.).
- Implement order tracking functionality to allow users to track the status of their orders in real time.

8. Conclusion

In conclusion, the development of our tech selling platform using React combined with Vite has been an enriching journey. Through this project, we have not only gained valuable experience in utilizing modern web development technologies but also created a platform that offers a seamless and enjoyable shopping experience for users.

By leveraging the power of React, we have built a dynamic and responsive user interface that enables users to browse, search, and purchase products with ease. The use of Vite as our build tool has contributed to fast development iterations and improved performance, ensuring a smooth user experience.

As we look to the future, there are numerous opportunities for further enhancement and refinement of our platform. From implementing advanced search and filtering options to enhancing the checkout process and integrating social features, there are many avenues to explore to meet the evolving needs of the users.

In conclusion, we are proud of the progress we have made with our tech-selling platform, and we are excited about the possibilities that lie ahead. Thank you for reviewing this project, and we look forward to the journey ahead as we continue to grow and evolve our skills.

9. References

For JavaScript:

https://www.w3schools.com/js/default.asp

https://stackoverflow.com/

For React JS:

https://react.dev/

https://www.w3schools.com/REACT/DEFAULT.ASP

https://stackoverflow.com/

https://www.geeksforgeeks.org/reactjs-introduction/?ref=header_search

For CSS and Styling:

https://getbootstrap.com/docs/5.3/getting-started/introduction/

10. GitHub Link to project

https://github.com/Vishy204/ReactJS_GalaxyTech

THANK YOU