**CHAPTER 1**

**INTRODUCTION**

* 1. **Objective**

The objective of our web application is to facilitate the buying and selling of products or services over the internet. It aims to provide a convenient and seamless shopping experience for customers while enabling businesses to expand their reach, increase sales, and manage their online operations efficiently. Key objectives of an e-commerce web application may include increasing website traffic, improving user engagement, increasing conversion rates, managing inventory and orders, and providing secure payment processing.

* 1. **Scope**

The scope of our web application typically includes all the features and functionalities required to conduct online transactions, manage orders and inventory, and provide a seamless shopping experience for customers. This may include:

1.Product catalogs and search functionality

2.Shopping cart and checkout process

3.Payment gateway integration

4.Order management and tracking

5.Customer account management and support

6.Inventory management and tracking

7.Sales analytics and reporting

8.Marketing and promotional features

9.Security and fraud prevention measures

The scope of an e-commerce web application can vary depending on the specific business requirements, the size of the business, and the target audience. The web application can be customized and scaled to meet the needs of the business and its customers.

* 1. **Motivation**

The motivation behind an e-commerce web application is to enable businesses to expand their reach, increase their sales, and manage their operations more efficiently by leveraging the internet as a platform for conducting transactions. E-commerce web applications offer several benefits for businesses, including:

* + 1. Global reach: E-commerce enables businesses to sell their products or services to a global audience, breaking down geographical barriers and reaching customers anywhere in the world.
    2. Increased sales: By making products and services available 24/7, e-commerce web applications can increase sales and revenue for businesses.
    3. Efficient operations: E-commerce web applications can help businesses automate various processes such as inventory management, order processing, and customer support, leading to more efficient and streamlined operations.
    4. Improved customer experience: E-commerce web applications offer a convenient and seamless shopping experience for customers, which can lead to increased customer satisfaction and loyalty.
    5. Data-driven insights: E-commerce web applications can provide businesses with valuable data insights such as customer behaviour, sales patterns, and inventory management, which can help businesses make data-driven decisions to optimize their operations and increase sales.

Overall, the motivation behind an e-commerce web application is to provide businesses with a powerful tool for expanding their reach, increasing sales, and improving their operations.

* 1. **Web Application Development Need & Importance**

Web application development is important for several reasons:

1.Accessibility: Web applications can be accessed from anywhere, at any time, as

long as there is an internet connection. This makes them highly accessible to users, regardless of their location.

2.Scalability: Web applications can be easily scaled to accommodate an increasing number of users, without the need for significant infrastructure changes.

3.Cost-effectiveness: Web applications can be developed and deployed at a relatively low cost, compared to traditional desktop applications.

4.Cross-platform compatibility: Web applications can be accessed from a variety of devices and operating systems, making them highly versatile.

5.User-friendly interface: Web applications are designed to be user-friendly, with a simple and intuitive interface that is easy to navigate. Overall, web application development is important for businesses and organizations looking to provide their users with easy access to their products and services, and to stay competitive in the digital age.

**CHAPTER 2**

**METHODOLOGY**

**2.1 Techniques Used**

When building an e-commerce web application with React, some of the techniques that can be used include:

1.Building reusable components: React is component-based architecture allows developers to build reusable UI components that can be used throughout the application. This can help to reduce development time and improve code quality.

2.Optimizing for performance: React's Virtual DOM and the use of techniques like code splitting and lazy loading can help to optimize the performance of the application, making it faster and more responsive.

3.Implementing server-side rendering: Server-side rendering can help to improve the application's initial load time and search engine optimization (SEO), by rendering the application on the server and sending a pre-rendered HTML page to the client.

4.Integrating with APIs: E-commerce web applications typically require integration with external APIs, such as payment gateways, shipping providers, and inventory management systems. React's modular architecture and the use of libraries like Axios can make it easier to integrate with these APIs.

5.Ensuring security: E-commerce web applications must be secure to protect sensitive user information such as credit card details. Techniques such as encryption, input validation, and server-side validation can help to improve the security of the application.

In summary, building an e-commerce web application with React requires a combination of techniques, including building reusable components, using a state management library, optimizing for performance, implementing server-side rendering, integrating with APIs, and ensuring security.

**2.2 Tools Used (**Brief Description about the tools)

1.React js: ReactJS is a declarative, efficient, and flexible JavaScript library

for building reusable UI components. It is an open-source, component-based front-end library which is responsible only for the view layer of the application. It was initially developed and maintained by Facebook and later used in its products like WhatsApp & Instagram.

2.Firebase: Firebase is an app / web development platform that helps to build and grow apps and games users love. Backed by Google and trusted by millions of businesses around

the world. We have used firebase as a database for storing details of users, products and the products which were purchased.

3.Visual Studio Code: Visual Studio Code is a lightweight but powerful source code editor which runs on desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages and runtimes (such as C++, C#, Java, Python, PHP, Go,

.NET). Our whole project was built using vs code.

4.Stripe: Stripe is a payment services provider that lets merchants accept credit and debit cards or other payments. Stripe payments are best suited for businesses that make most of their sales online, as most of its unique features are primarily geared toward online sales.

5 HTML: HTML (Hypertext Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behaviour (JavaScript).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

6.CSS: Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML(including XML dialects such

as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

7.JS: JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Webpages, many non-browser environments also use it, such as Node.js. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g., functional programming) styles.

**CHAPTER 3**

**SYSTEM REQUIREMENTS SPECIFICATION**

**3.1 Software Requirements**

* Operating System: Developers can use a variety of operating systems such as Windows, MacOS, or Linux, depending on their preference.
* Code Editor or Integrated Development Environment (IDE): A code editor or IDE is required to write, edit, and debug the code. Some popular choices include Visual Studio Code, Sublime Text, Atom, or Eclipse.
* Web Browser: Web developers need to test their applications in different browsers such as Chrome, Firefox, Safari, or Edge.
* Version Control System: Developers should use a version control system such as Git to track changes to their code and collaborate with other developers.
* Web Server: Developers need a web server to serve their application to clients. Popular web servers include Apache, Nginx, and IIS.

**3.2 Hardware Requirements**

* Processor: A multi-core processor is recommended for faster performance during development.
* RAM: A minimum of 8 GB of RAM is recommended for web application development.
* Storage: Sufficient storage is required to store code, databases, and other development files.
* Display: A large, high-resolution monitor is recommended for better productivity.
* Internet Connection: A reliable internet connection is necessary for downloading updates, libraries, and frameworks, as well as for testing and deploying the application

**3.3 Functional Requirements**

* User registration and login
* Product catalogue and browsing
* Shopping cart and checkout process
* Order management and tracking
* Payment gateway integration
* Shipping and delivery management
* Customer support and feedback system
* Search and filter functionality
* Wishlist’s and saved items
* Personalized recommendations
* Social sharing and referral features
* Admin dashboard for managing products, orders, and customers.

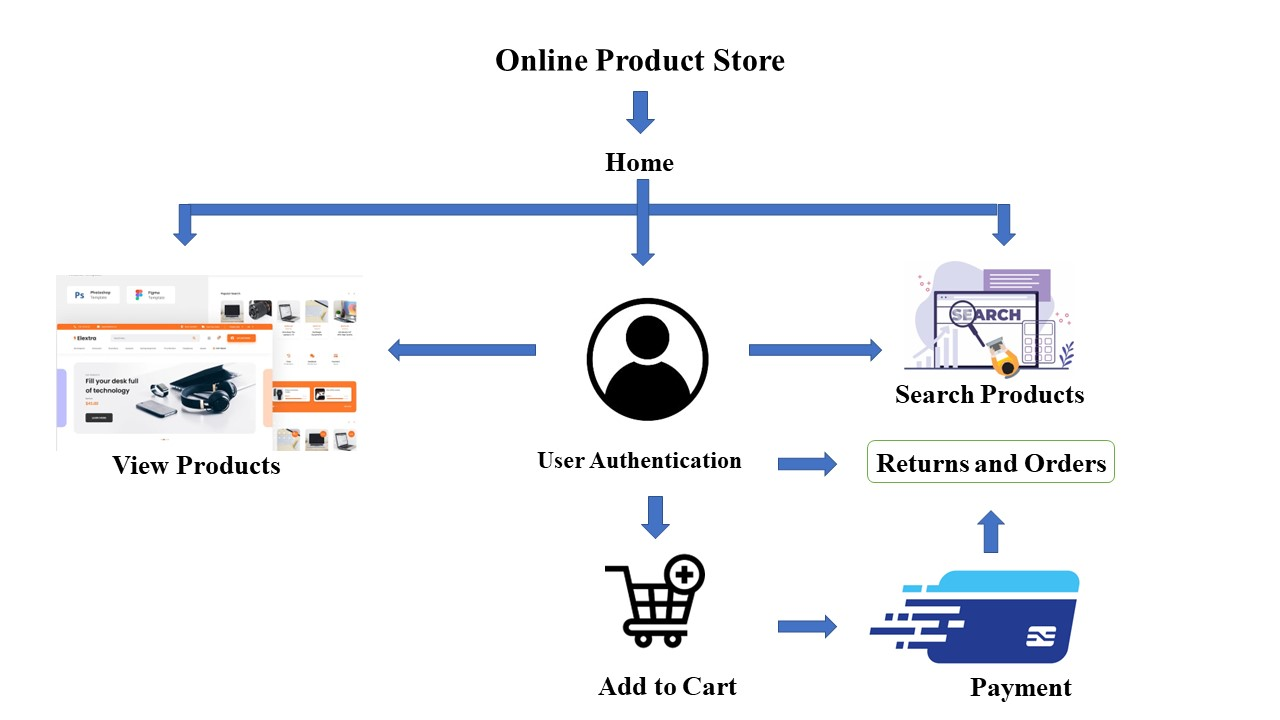
**3.4 Non - Functional Requirements**

* Performance and scalability: the ability of the application to handle large volumes of users and transactions without slowing down or crashing.
* Security: the application should be designed to protect sensitive user information, such as credit card details, and prevent unauthorized access to the system.
* Availability: the application should be always available to users, with minimal downtime for maintenance or updates.
* Usability and user experience: the application should be easy to navigate, with clear and intuitive interfaces, and provide a seamless user experience.
* Compatibility: the application should be compatible with different web browsers, operating systems, and devices, and accessible to users with disabilities.
* Reliability and fault tolerance: the application should be able to recover from errors or failures and maintain data integrity.
* Compliance: the application should comply with relevant regulations and industry standards, such as data protection laws and payment card industry standards.

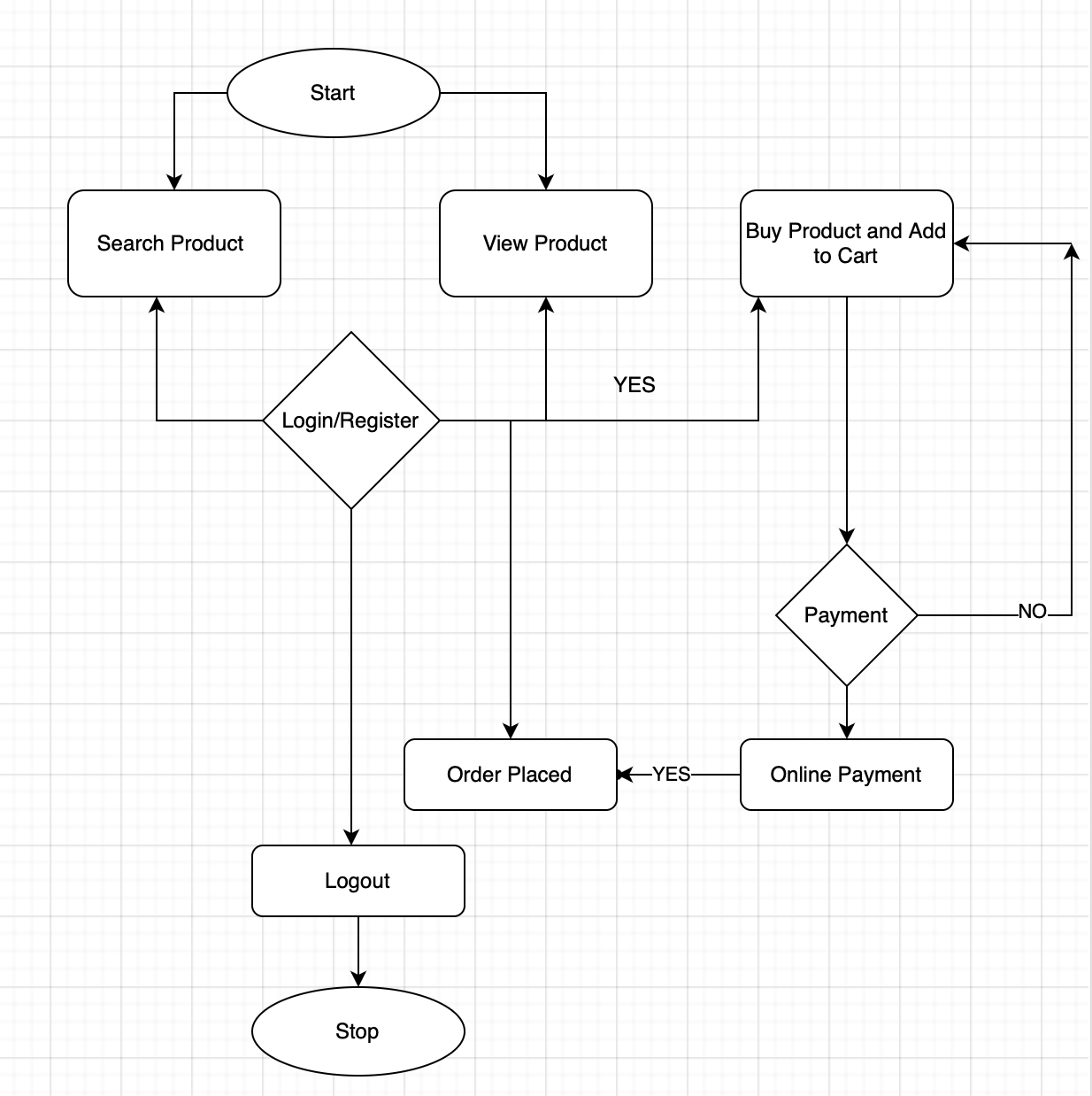
**CHAPTER 4**

**SYSTEM DESIGN AND DEVELOPMENT**

**4.1 Architectural Design**

****

**Flow diagram**



When an user visits our website home page is visible i.e. the user can view and search for products, if he/she wants to purchase the products they will be redirected to the login/signup page . Once the user registers “Add to Cart” button will be enabled . Then the user can proceed to payment page and complete the payment process. Once the payment gets completed user will be redirected to “Return and Orders” page where the payment details and preview of purchased products will be seen .

**CHAPTER 5**

**IMPLEMENTATION**

**5.1 Modules Implemented**

1. Product Catalogue: A module to manage and display products.

* The catalogue also contains a button called “Add to cart” functionality which when clicked adds that product to the cart.

2. Shopping Cart: A module to allow customers to select products and manage their orders.

3. Payment Gateway: A module to facilitate online payment transactions.

* Payment gateway used is Stripe.
* The customer provides their card information, either online or in person.
* Those card details enter Stripe’s payment gateway, which encrypts the data.
* Stripe sends that data to the acquirer, which is a bank that will process the transaction on the merchant’s behalf. In this step, Stripe serves as the merchant (with the business owner as a sub merchant). This means Stripe users don’t have to set up a [merchant account] (https://nerdwallet.com/article/small-business/what-is-a-merchant-account), which can be cumbersome.
* The payment passes through a credit card network, such as Visa or Mastercard, to the cardholder’s issuing bank.
* The issuing bank approves or denies the transaction.
* That signal travels from the issuing bank through the card network to the acquirer, then through the gateway to the customer — who sees a message telling them the payment has been accepted or declined.

4. User Authentication: A module to manage user accounts, including registration, login, and logout.

* First the user should register by providing the required credentials and the credentials will be stored in database.
* When the user enters email id and password it searches the database and if the credential matches the user will be logged in.
* When the user clicks logout, the credentials will be removed from the website but not from the database.

5. Order Management: A module to manage order processing and shipment.

* This page opens if the payment is successful.
  + 1. Search and Filtering: A module to help customers find products by searching or filtering based on various criteria’s find products by searching or filtering based on various criteria.

**CHAPTER 6**

**TESTING and RESULTS**

**6.1 Snapshots of the project and description**

1.Header Component: The header consists of logo, location, search bar , login option and a cart button .



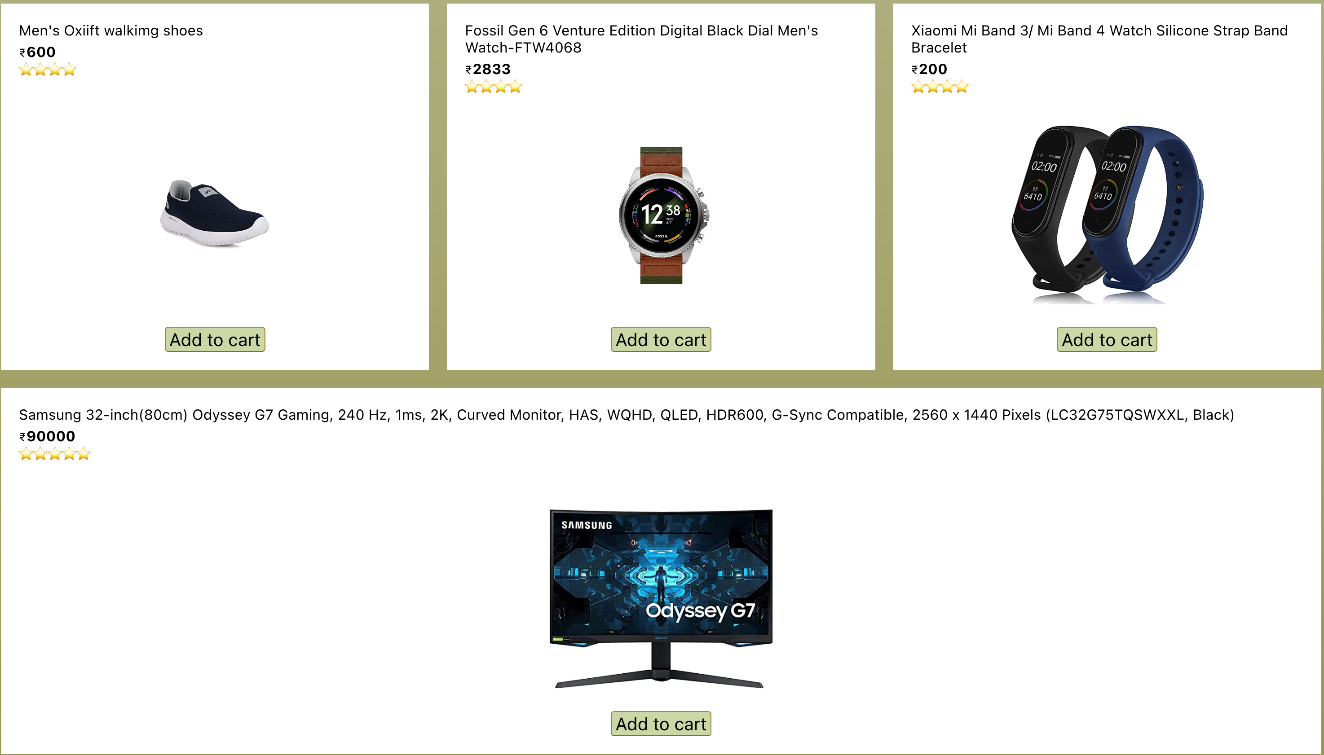
**Fig.6.1.1 Header Component**

→ Location - displays the user location.

→ Login - when clicked on sign-in the user is redirected to the login/signup page.

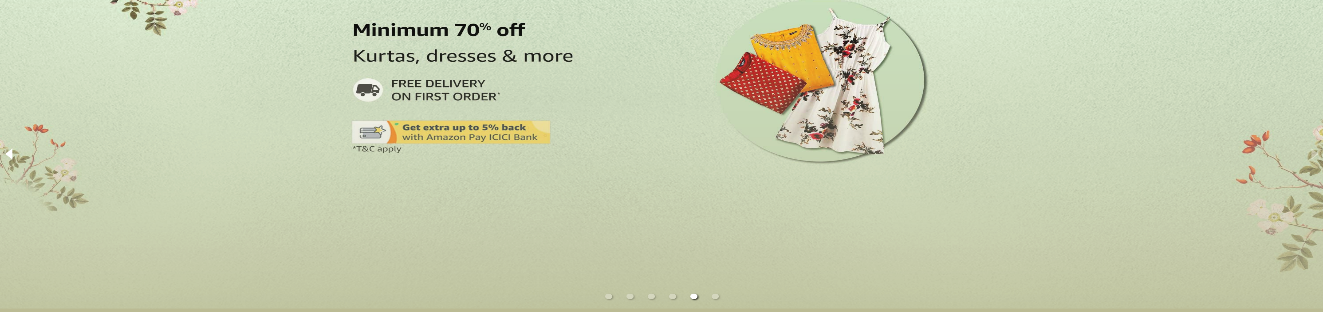
→ Cart - Displays the number of items added to the cart by the user.

2.Card Component (display of products with their details): The card component consists of title, price, rating, image of the product and an “Add to Cart “ button which when clicked gets added to the cart.



**Fig.6.1.2 Card Component**

3.Carousel: Displays a list of images changing at a particular time interval.

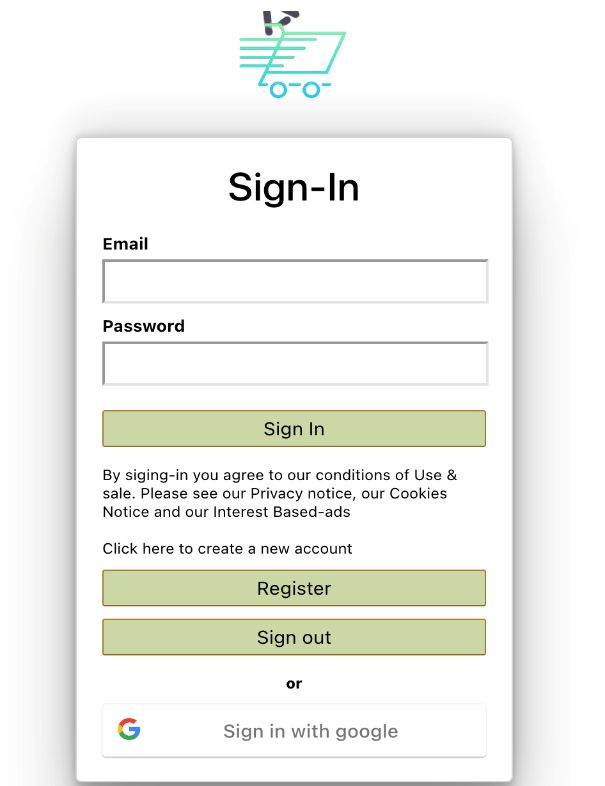


**Fig.6.1.3 Carousel**

4.Login/Signup Page: Here the user gets three options a)Login b)Register c)Logout

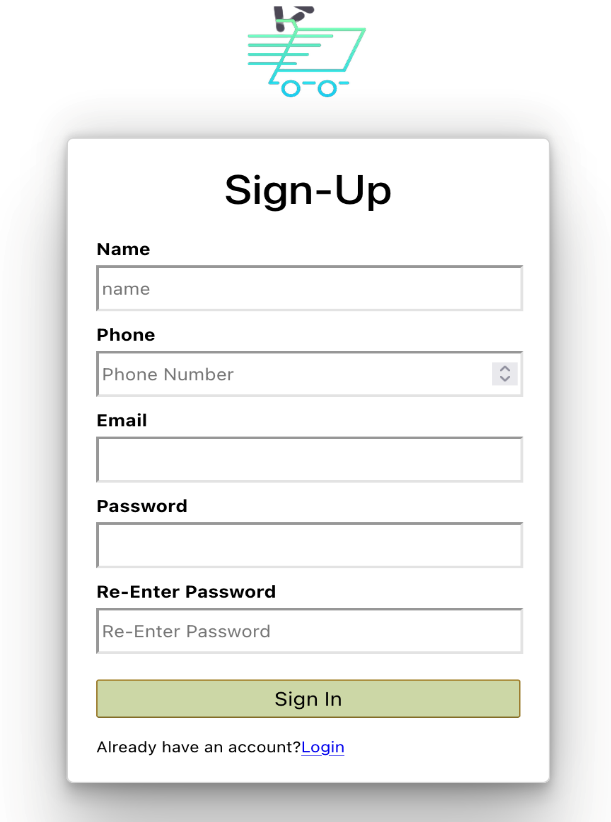
a) Login - the user can login if he/she is already registered with email and password

/ Google.

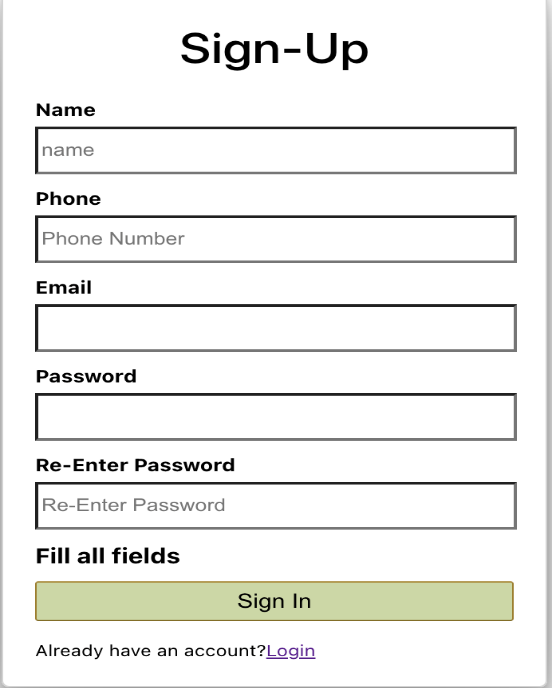
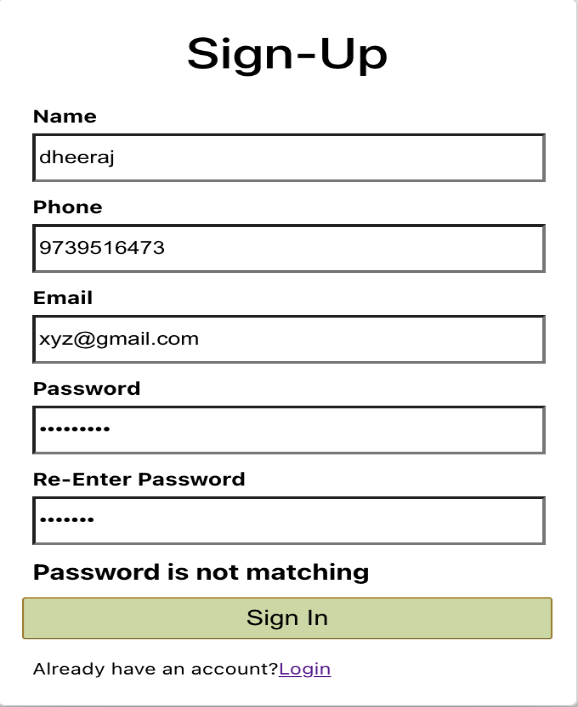


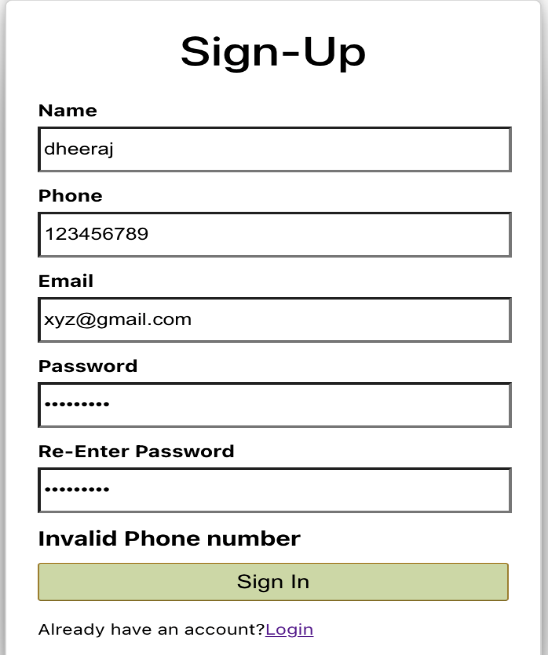
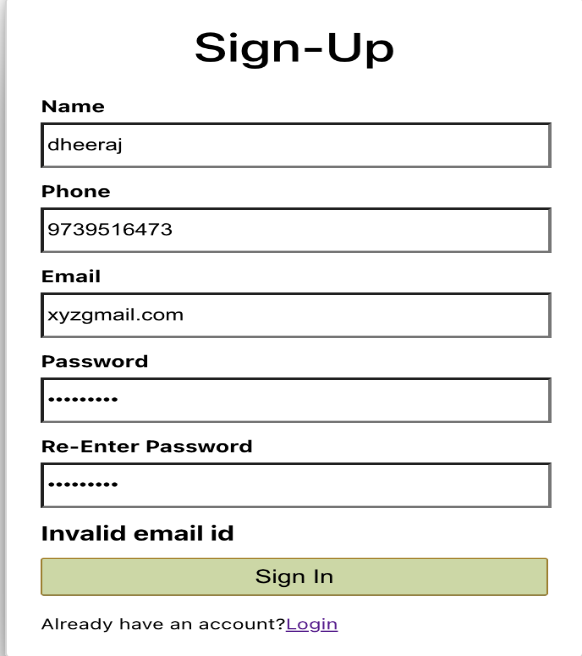
**Fig.6.1.4a Login page**

b) Register /Sign up- Here the user can register if he/she is new to the website.



**Fig.6.1.4b Register/Sign-Up page**

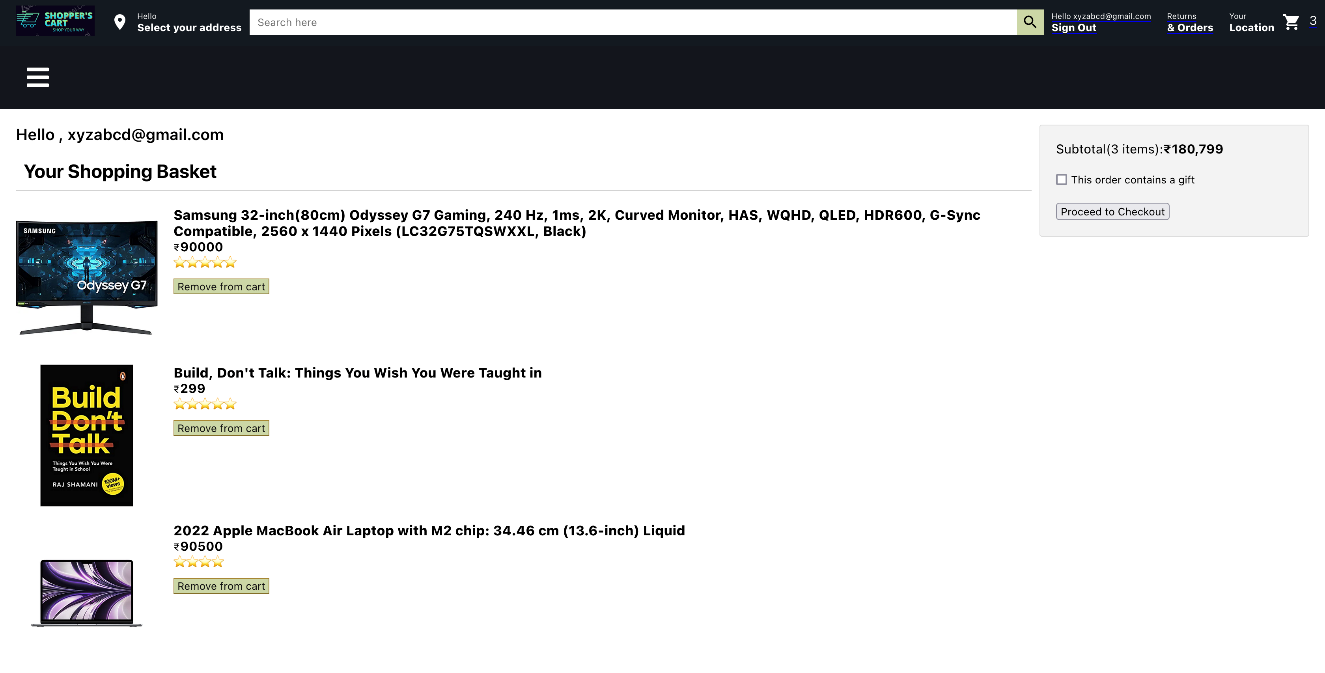
 

**Fig.6.1.4b Testing and Validation**

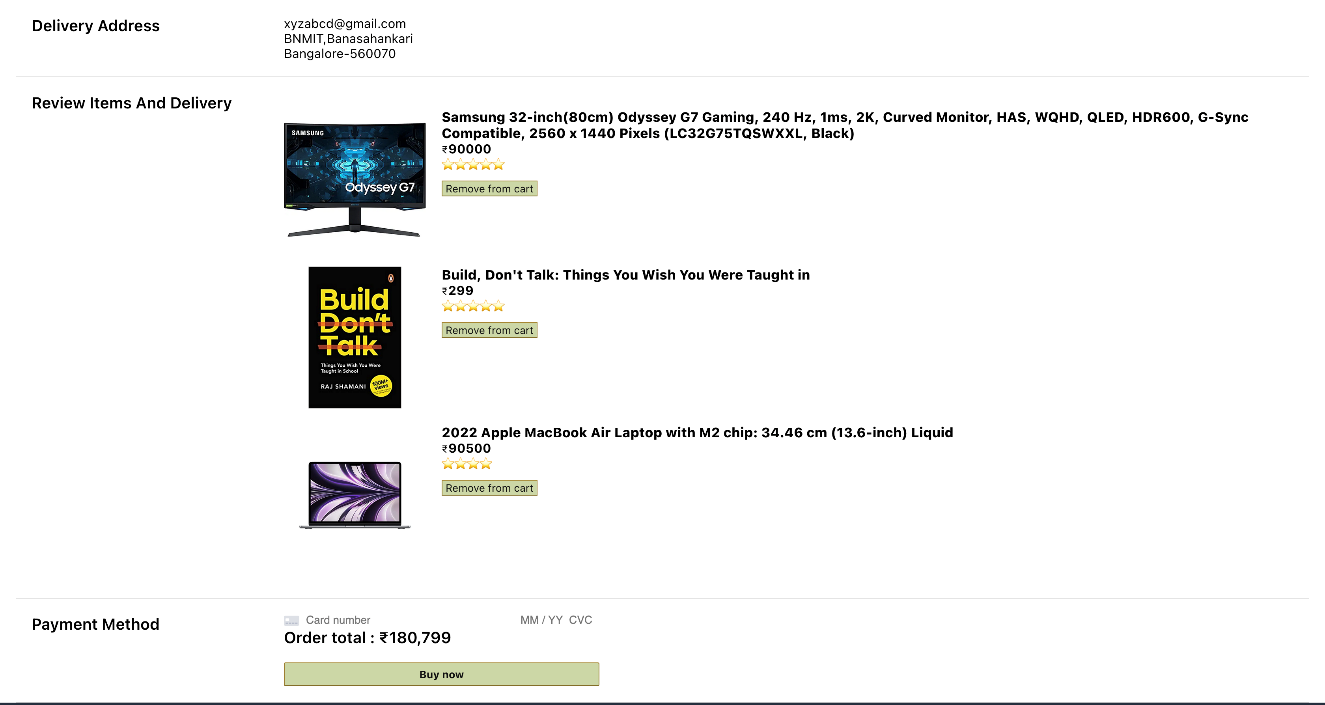
c)Logout - By clicking on logout the user gets logged out from the website.

5.Checkout page - When clicked on cart button the user is redirected to the checkout page where preview of the products added to the cart is visible with a button “Remove from Cart “which when clicked removes the product from the cart and a subtotal window is also visible with the total amount of products which gets updated if a product is added / removed.



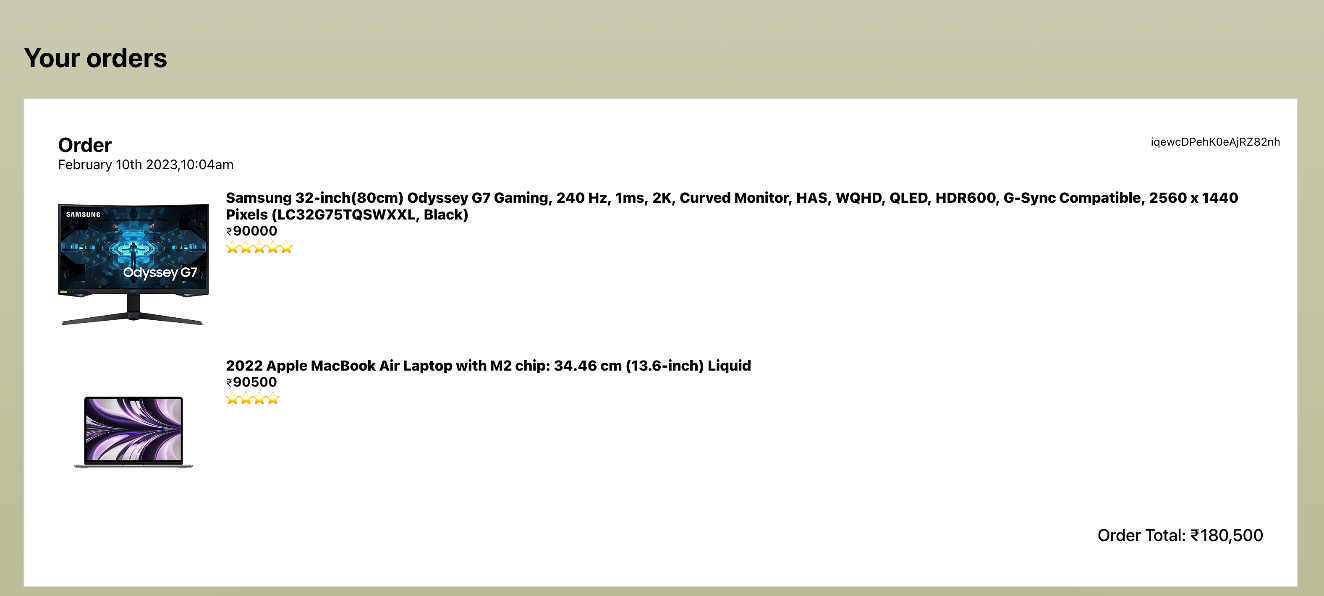
**Fig.6.1.5 Checkout page**

6.Payment Page: When the user is ready to checkout, user is redirected to the payment page where the user’s address, preview of products and the total amount is visible with the payment-card element.



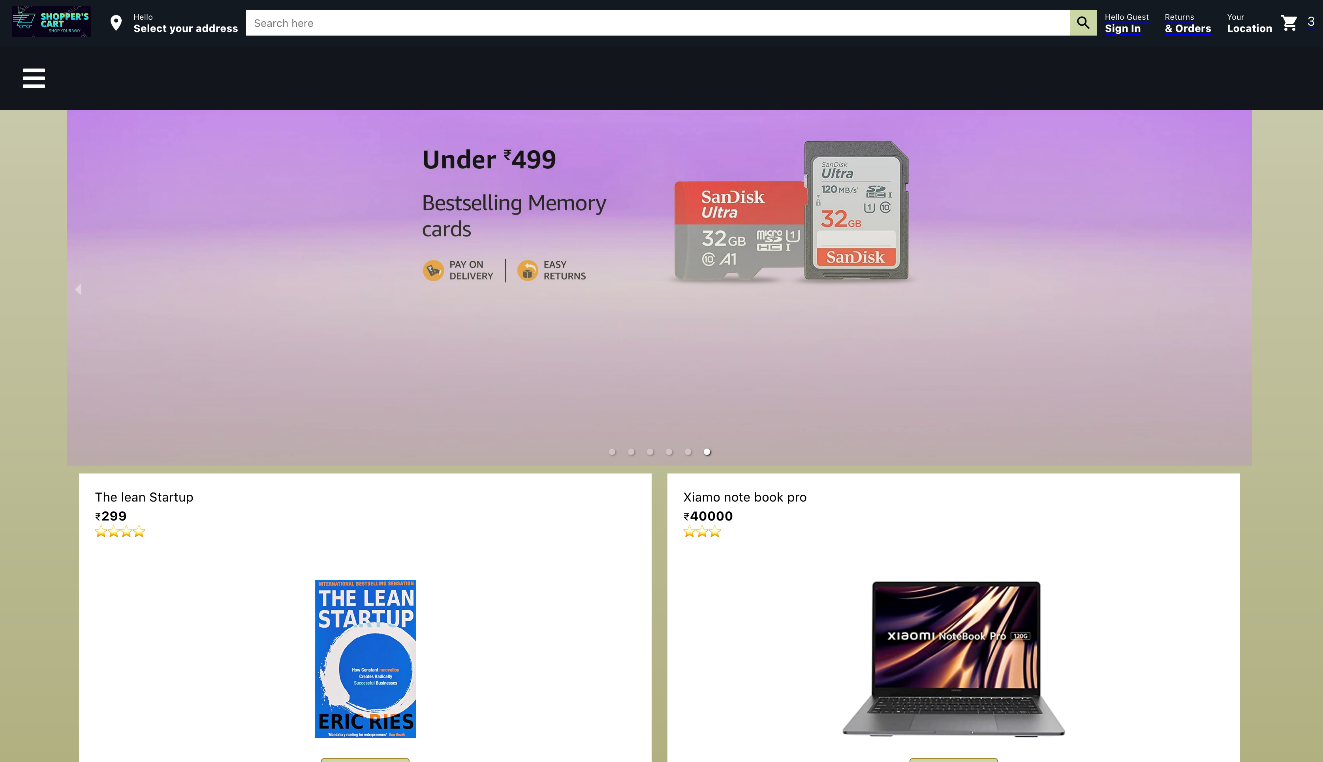
**Fig.6.1.6 Payment page**

7.Returns and Orders Page: Once the payment is successful the user is redirected to the “Returns and Orders Page “where the payment id, payment date and the details of product are visible.



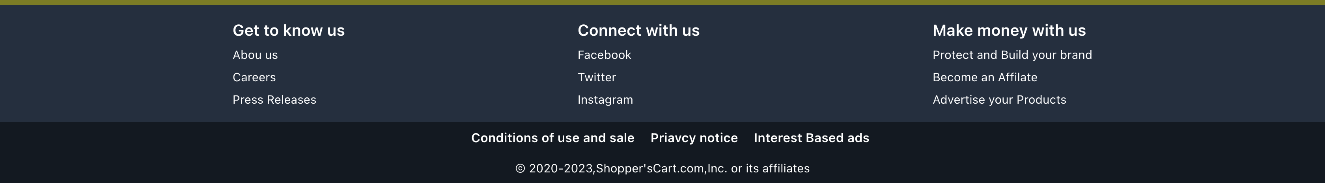
**Fig.6.1.8 Returns and orders page**

8.Home Page: It is a combination of Header, Carousel, Card, and footer component.



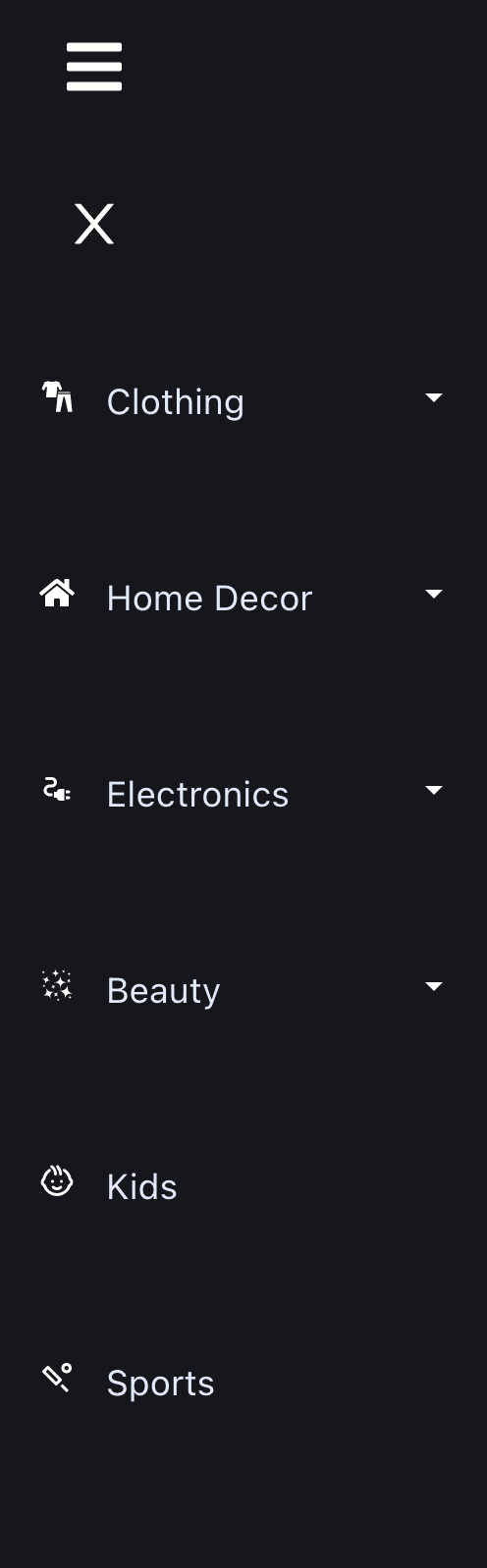
**Fig.6.1.8 Home page**

9.Footer component: Consist details about the website.



**Fig.6.1.9 Footer Component**

10.Sidebar component: It consists of all fields of products available in our store.



**Fig.6.1.10 Sidebar Component**

**6.2 Observation about the project**

Observations made from our e-commerce web application:

1.User interface design: The design of web application is critical to its success. The interface should be user-friendly, easy to navigate, and visually appealing. A well- designed interface can help to improve the user experience and increase engagement.

2.Product catalogue and search: The product catalogue and search functionality are essential components of an e-commerce web application. The catalogue should be

well-organized and easy to browse, and the search functionality should be fast and accurate.

3.Checkout process: The checkout process should be streamlined and easy to use. It should include clear instructions and guidance to help users complete their purchases quickly and easily.

4.Payment processing: Payment processing is a critical component of an e-commerce web application. The application should support secure payment processing through multiple payment gateways and provide clear information about payment options and fees.

5.Order tracking and management: Once an order is placed, the e-commerce web application should provide users with tools to track their orders and manage their account details. This can include features such as order history, shipping status, and account management.

6.Security: E-commerce web applications must be secure to protect user information, including personal and payment details. The application should use encryption,

input validation, and server-side validation to ensure the security of user data.

Overall, an e-commerce web application should provide a seamless and secure shopping experience for users. It should be well-designed, easy to use, and provide users with the tools they need to find, purchase, and manage products and orders.

**CHAPTER 7**

**CONCLUSION AND FUTURE ENHANCEMENT**

In conclusion, an e-commerce web application is a critical component of any online business, providing a platform for customers to browse and purchase products or services. To be successful, an e-commerce web application must be well-designed, easy to use, and secure. Key features such as product catalogue and search, checkout process, payment processing, and order tracking and management must be well-

implemented to ensure a seamless and positive user experience.

In terms of future enhancements, there are several areas that could be focused on to improve the performance and user experience of an e-commerce web application.

Some of these areas include:

1.Personalization: Adding personalized product recommendations and marketing messages can help to improve the user experience and drive sales.

2.Mobile optimization: More and more users are accessing e-commerce web applications from mobile devices. Optimizing the application for mobile use can help to improve accessibility and user engagement.

3.Augmented reality: Integrating augmented reality (AR) technology into the application can provide users with a more interactive and immersive shopping experience.

4.Voice-enabled interfaces: Voice-enabled interfaces can help to streamline the shopping experience and make it more accessible to users with disabilities.

5.Artificial intelligence: Using artificial intelligence (AI) technology to improve search accuracy, product recommendations, and customer service can help to improve the efficiency and effectiveness of the application.

Overall, by incorporating these and other future enhancements, an e-commerce web application can continue to evolve and meet the changing needs of its users, providing a more personalized, engaging, and convenient shopping experience.

**REFERENCES**

1. <https://www.npmjs.com/>
2. <https://reactjs.org/>
3. https://nodejs.org/en/
4. <https://expressjs.com/>
5. <https://dashboard.stripe.com/test/dashboard>
6. https://codesandbox.io/s/new-feather-rsrp6p?file=/src/App.js
7. <https://colorhunt.co/>
8. <https://www.adobe.com/express/feature/image/transparent-background>
9. <https://www.geeksforgeeks.org/reactjs-toast-notification/>
10. https://nodejs.org/en/