CHAPTER 1
INTRODUCTION

In the vibrant world of online retail, the "Shopper" website has emerged as a versatile and user-centric E-commerce platform, catering to the diverse clothing needs of men, women, and kids. This chapter provides an in-depth overview of the project, outlining its objectives, features, and the underlying motivations that have driven its development.

1.1 PROJECT OVERVIEW

1.1.1 E-COMMERCE AND THE SHOPPER WEBSITE

The "Shopper" website is a testament to the transformative power of E-commerce, particularly in the realm of fashion. Offering a curated selection of clothing for men, women, and kids, the platform transcends geographical boundaries, providing an accessible and enjoyable shopping experience for users across the globe.

1.1.2 FEATURES OF THE SHOPPER WEBSITE

- **Product Catalog:**

The heart of the "Shopper" website lies in its extensive product catalog. Tailored for men, women, and kids, this catalog showcases a diverse array of clothing items, from stylish apparel to comfortable loungewear.

- **User Authentication:**

Prioritizing user security, the "Shopper" website implements a robust user authentication system. This ensures that each user's personal information and transaction details are safeguarded, fostering a sense of trust and confidence.

- **Shopping Cart Functionality:**

An intuitive shopping cart system allows users to effortlessly add, remove, and modify items. This feature is designed to streamline the shopping experience, ensuring flexibility and convenience.

1.1.3 USER AUTHENTICATION AND SECURITY

Security is paramount in the "Shopper" website. The user authentication system employs encryption techniques, safeguarding sensitive user data and ensuring that each shopping journey is secure and private.

1.1.4 LIMITATIONS OF THE CURRENT SYSTEM

As with any system, the "Shopper" website has its limitations. This section addresses potential challenges, acknowledging areas for improvement and future development to enhance the overall user experience.

1.1.5 MOTIVATION BEHIND DEVELOPING SHOPPER

The "Shopper" website is born out of the recognition that the online shopping landscape continues to evolve. The motivation to develop this platform arises from the desire to offer a distinctive and enjoyable online shopping experience for individuals seeking quality clothing for men, women, and kids.

1.2 MOTIVATION AND OBJECTIVES

1.2.1 ADDRESSING DIVERSE CLOTHING NEEDS

The primary motivation behind the "Shopper" website is to address the diverse clothing needs of individuals across different age groups and genders. The project seeks to become a one-stop destination for fashionable and comfortable clothing for men, women, and kids.

1.2.2 OBJECTIVES OF THE "SHOPPER" PROJECT

The overarching objectives of the "Shopper" website project include:

- Providing a seamless and enjoyable online shopping experience for users.

- Ensuring the security and privacy of user information through robust authentication measures.
- Curating a diverse product catalog that caters to the specific preferences and trends in men's, women's, and kids' fashion.
- Continuous improvement and innovation to stay abreast of evolving E-commerce trends and technologies.

CHAPTER 2
HARDWARE AND SOFTWARE USE

In the development and deployment of the "Shopper" website, specific hardware and software requirements are essential to ensure optimal performance, security, and scalability.

2.1 HARDWARE REQUIREMENTS

2.1.1 Server Infrastructure

The server infrastructure for the "Shopper" website should meet the following hardware specifications:

- **Processor:** Multi-core processors (e.g., Intel Xeon, AMD Ryzen) for handling concurrent requests efficiently.
- **Memory (RAM):** At least 8GB RAM to accommodate the demands of the MongoDB database and Node.js server.
- **Storage:** SSD storage for improved data retrieval speed and overall system responsiveness.
- **Network:** High-speed internet connectivity to facilitate smooth communication between server and clients.

2.1.2 Client Devices

End-users accessing the "Shopper" website should have devices that meet the following requirements:

- **Desktops/Laptops:** Modern web browsers (Chrome, Firefox, Safari) on Windows, macOS, or Linux operating systems.

2.2 SOFTWARE REQUIREMENTS

2.2.1 Operating System

- **Server:** Linux-based operating system (e.g., Ubuntu) for hosting Node.js and MongoDB.

- **Client:** Compatible with Windows, macOS, and Linux for desktop users. Responsive design ensures compatibility with various mobile operating systems (iOS, Android).

2.2.2 Backend Technologies

- **Node.js:** Version 14 or higher as the server-side runtime environment.
- **Express.js:** A minimalist web application framework for Node.js, facilitating the creation of robust APIs.
- **MongoDB:** A NoSQL database, version 4.x, for storing product data, user information, and other relevant data.

2.2.3 Frontend Technologies

- **React.js:** Version 17 or higher for building the interactive user interface.

2.2.4 Additional Tools and Libraries

- **npm:** Package managers for installing and managing project dependencies.
- **Webpack:** A module bundler for optimizing and bundling frontend assets.
- **Babel:** For transpiling modern JavaScript code to ensure compatibility with a wide range of browsers.

2.2.5 Development Tools

- **Code Editor:** Visual Studio Code editor for development.
- **Version Control:** Git for source code version control, preferably hosted on platforms like GitHub or GitLab.
- **Postman:** For testing and documenting APIs during development.

2.2.6 Deployment and Hosting

- **Cloud Service:** Platform-as-a-Service (PaaS) like Heroku, AWS, or similar for hosting the application.
- **Database Hosting:** MongoDB Atlas or a self-hosted MongoDB instance.

These hardware and software requirements form the foundation for the development and deployment of the "Shopper" website, ensuring a smooth and reliable user experience across various devices.

CHAPTER 3

DATA FLOW DIAGRAM

The data flow diagram for the "Shopper" e-commerce website involves a seamless flow of information. Users register with personal details, which are authenticated before generating a session token. As users browse products, requests for information flow to the database, and selected items are added to the shopping cart, updating the user's cart details.

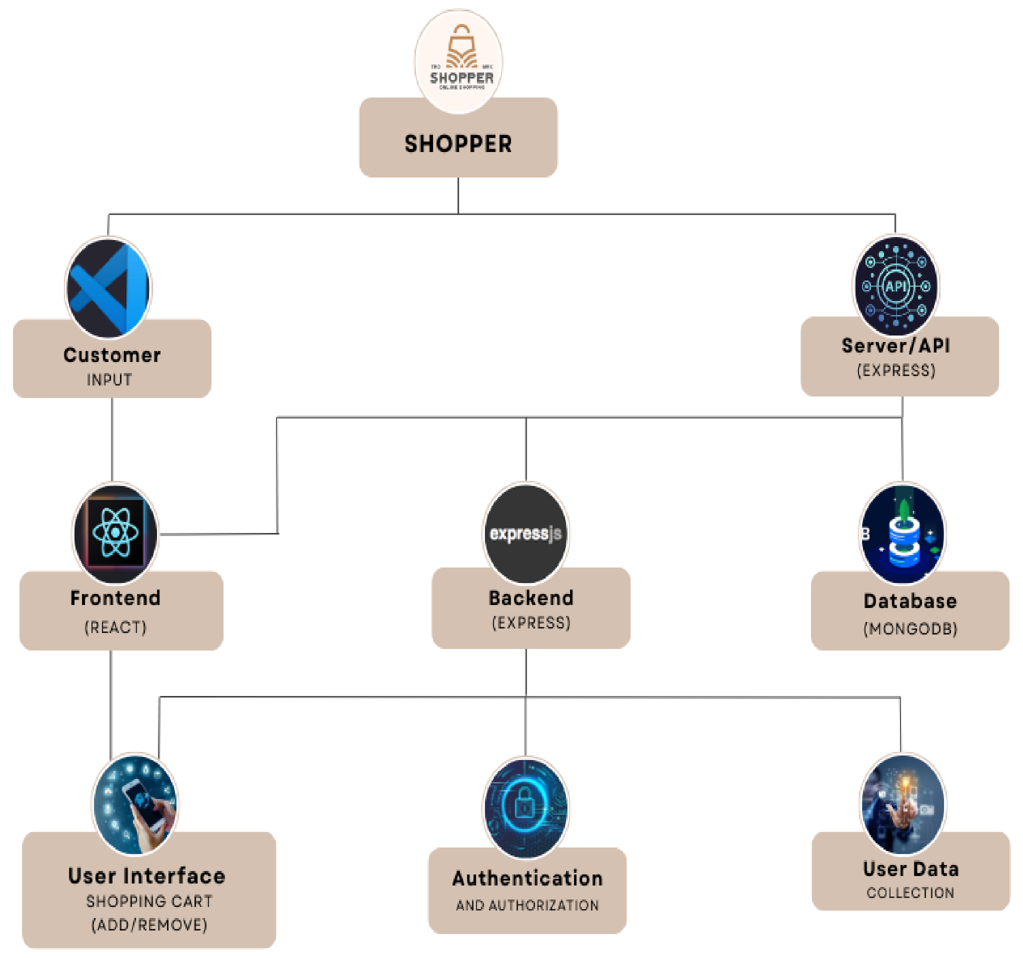


Fig 3.1 Data Flow Diagram

3.1 Frontend:

- This is the part of the website that users see and interact with. It's built with React, a JavaScript library for building user interfaces.
- The frontend includes the user interface (UI), which is the layout and design of the website, as well as the shopping cart, where users can add and remove items they want to buy.

3.2 Backend:

- This is the part of the website that users don't see. It's built with Express, a Node.js framework for building web applications.
- The backend includes the server/API, which processes user requests and sends data to the frontend. It also includes the database, which stores product information, user data, and other data that the website needs to run.

3.3 Authentication and Authorization:

- This system ensures that only authorized users can access certain parts of the website and make purchases.
- It typically involves users creating an account and logging in with a username and password.

3.4 User Data Collection:

- The website collects data about users, such as their name, email address, and purchase history.
- This data can be used to personalize the shopping experience, send targeted marketing emails, and improve the website overall.

3.5 Database:

- The website stores data in a MongoDB database.
- This database stores all of the information that the website needs to run, such as product information, user data, and orders.

3.6 Express:

- Express is a Node.js framework that is used to build the backend of the website.
- It provides a number of features that make it easy to develop web applications, such as routing, middleware, and templating.

3.6 React:

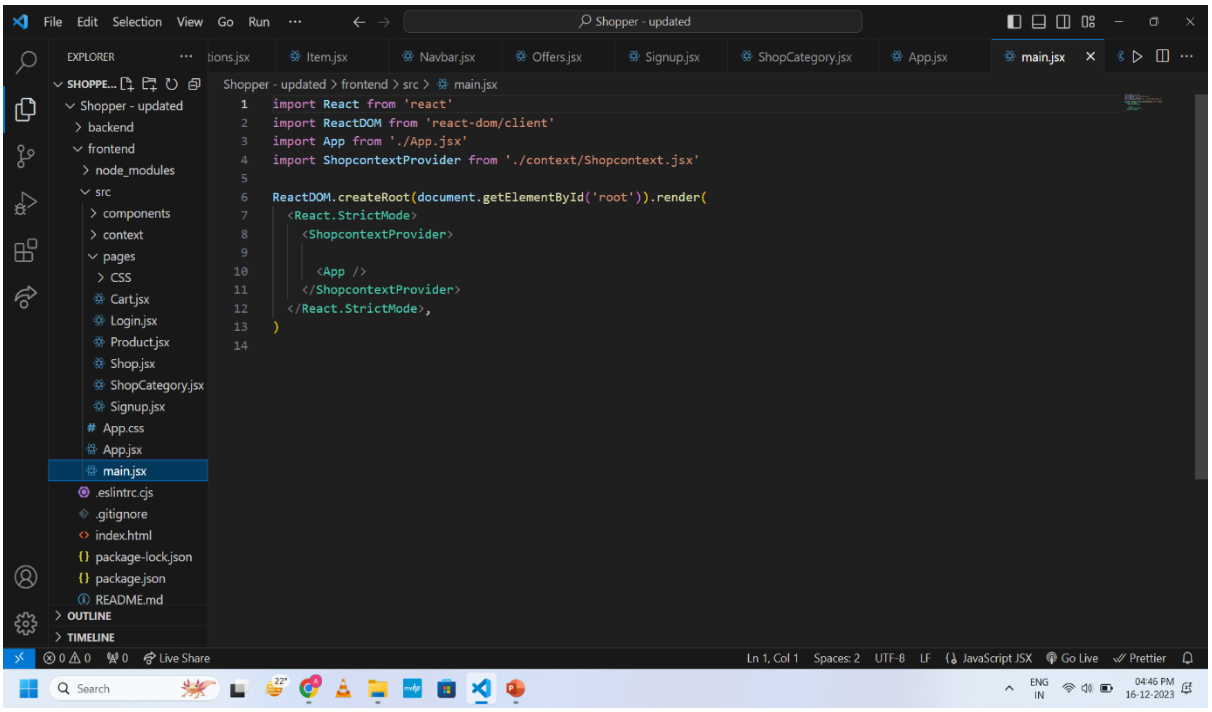
- React is a JavaScript library that is used to build the frontend of the website.
- It provides a number of features that make it easy to build user interfaces, such as virtual DOM, components, and state management.

CHAPTER 4

PROJECT MODULES

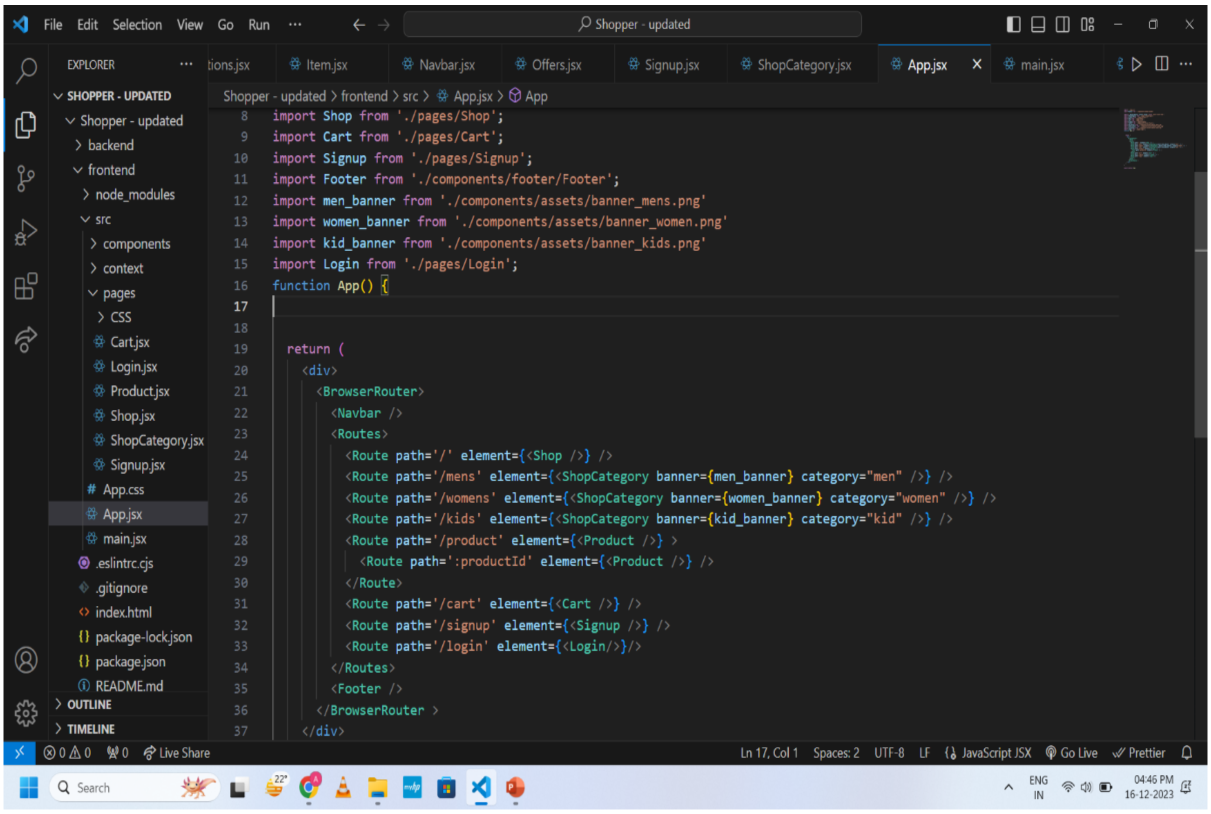
The "Modules" page of our project documentation provides an overview of the system's major components, outlining their features and functionalities. The User Authentication module ensures secure user access, allowing account creation, login, and password recovery. Product Management encompasses a dynamic catalog, search capabilities, and user reviews. The Shopping Cart module facilitates seamless product transactions, enabling users to add, remove, and view items. Lets0 see how we are created our Shopper Application:

Fig 4.1: Main.jsx



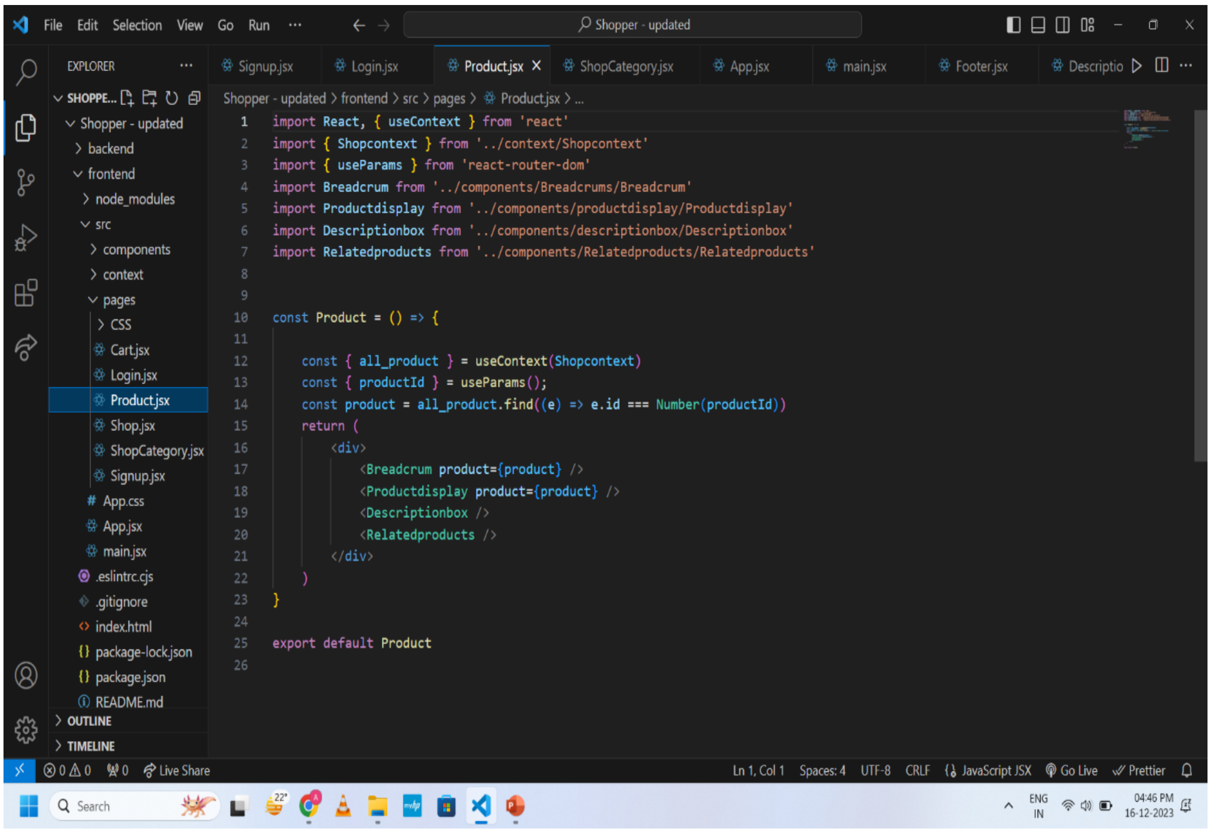
```
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App.jsx'
import ShopcontextProvider from './context/Shopcontext.jsx'

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <ShopcontextProvider>

      <App />
    </ShopcontextProvider>
  </React.StrictMode>,
)
```

Fig 4.2 App.jsx



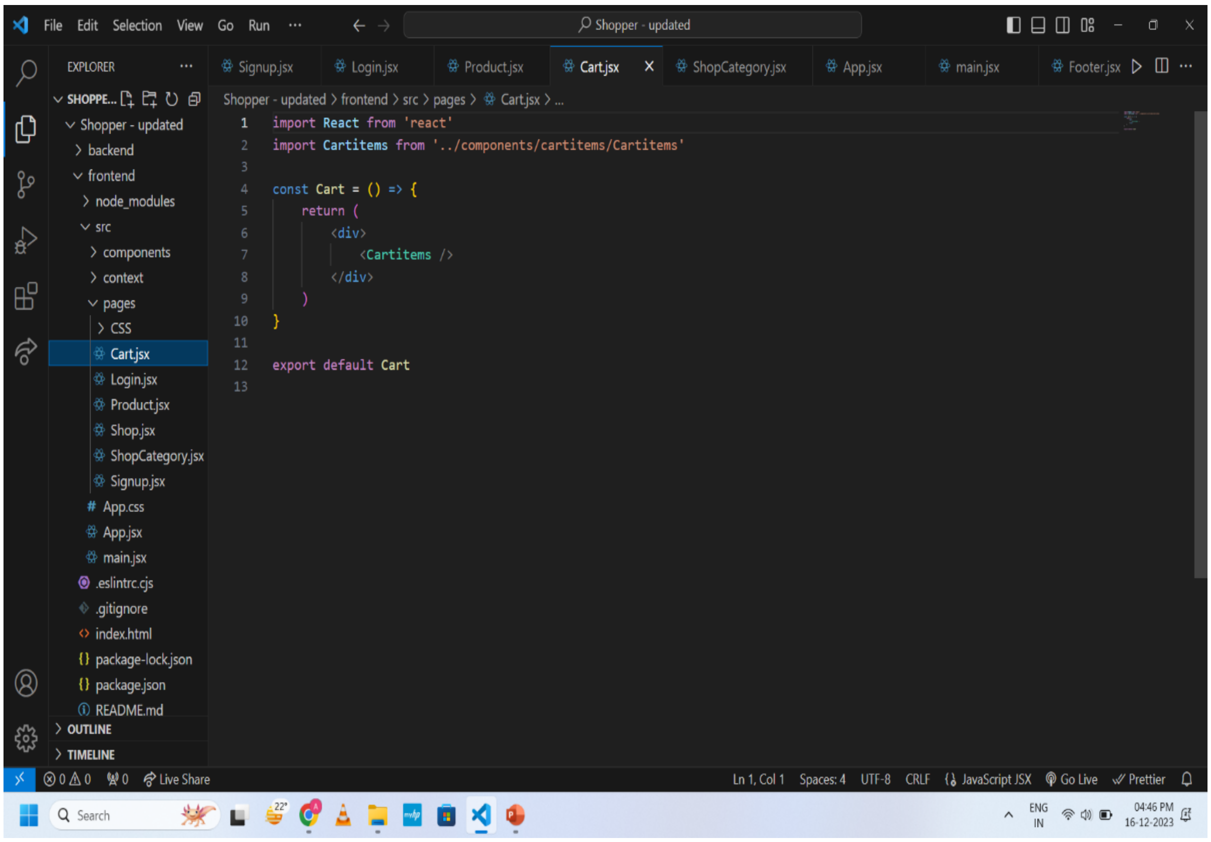
```
import Shop from './pages/Shop';
import Cart from './pages/Cart';
import Signup from './pages/Signup';
import Footer from './components/footer/Footer';
import men_banner from './components/assets/banner_mens.png'
import women_banner from './components/assets/banner_women.png'
import kid_banner from './components/assets/banner_kids.png'
import Login from './pages/Login';
function App() {


  return (
    <div>
      <BrowserRouter>
        <Navbar />
        <Routes>
          <Route path='/' element={<Shop />} />
          <Route path='/mens' element={<ShopCategory banner={men_banner} category="men" />} />
          <Route path='/womens' element={<ShopCategory banner={women_banner} category="women" />} />
          <Route path='/kids' element={<ShopCategory banner={kid_banner} category="kid" />} />
          <Route path='/product' element={<Product />} >
            <Route path=':productId' element={<Product />} />
          </Route>
          <Route path='/cart' element={<Cart />} />
          <Route path='/signup' element={<Signup />} />
          <Route path='/login' element={<Login/>}/>
        </Routes>
        <Footer />
      </BrowserRouter >
    </div>
```

Fig 4.3 Product.jsx



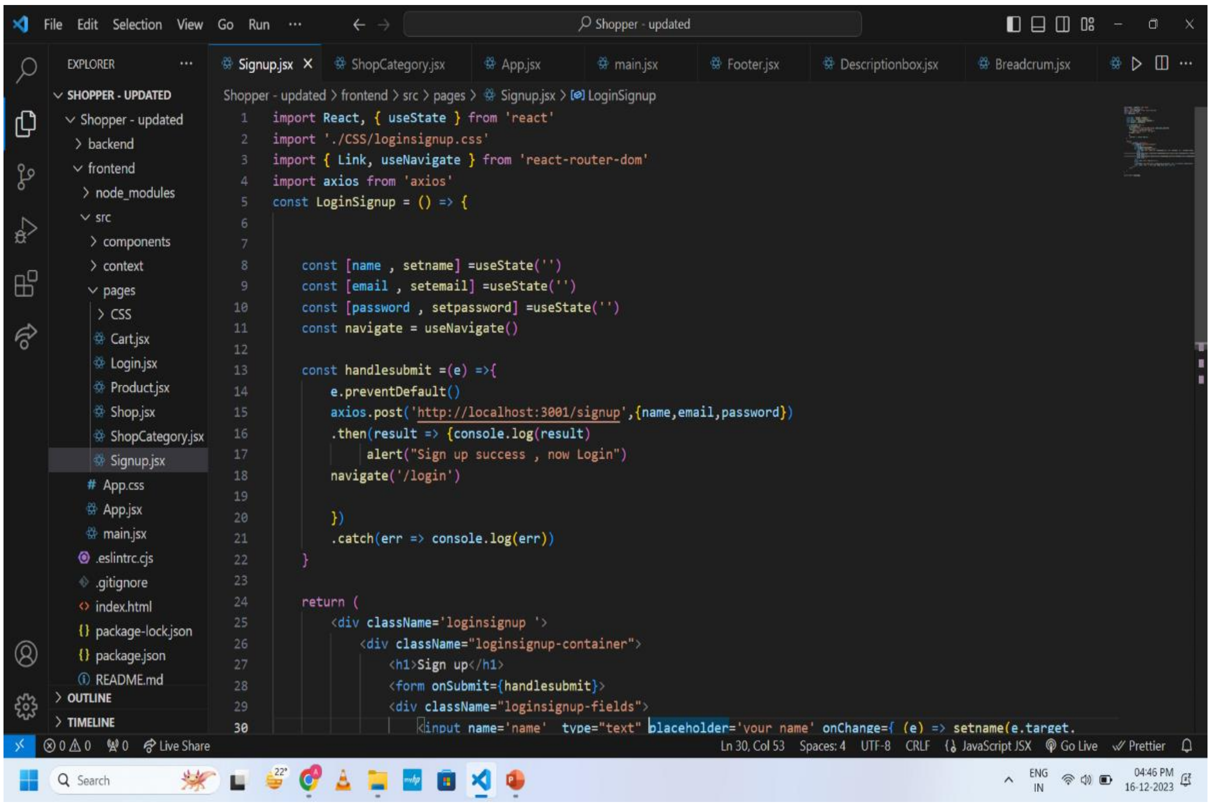
```
import React, { useContext } from 'react'
import { Shopcontext } from '../context/Shopcontext'
import { useParams } from 'react-router-dom'
import Breadcrum from '../components/Breadcrums/Breadcrum'
import Productdisplay from '../components/productdisplay/Productdisplay'
import Descriptionbox from '../components/descriptionbox/Descriptionbox'
import Relatedproducts from '../components/Relatedproducts/Relatedproducts'


const Product = () => {

    const { all_product } = useContext(Shopcontext)
    const { productId } = useParams();
    const product = all_product.find((e) => e.id === Number(productId))
    return (
        <div>
            <Breadcrum product={product} />
            <Productdisplay product={product} />
            <Descriptionbox />
            <Relatedproducts />
        </div>
    )
}

export default Product
```

Fig 4.4 Cart.jsx



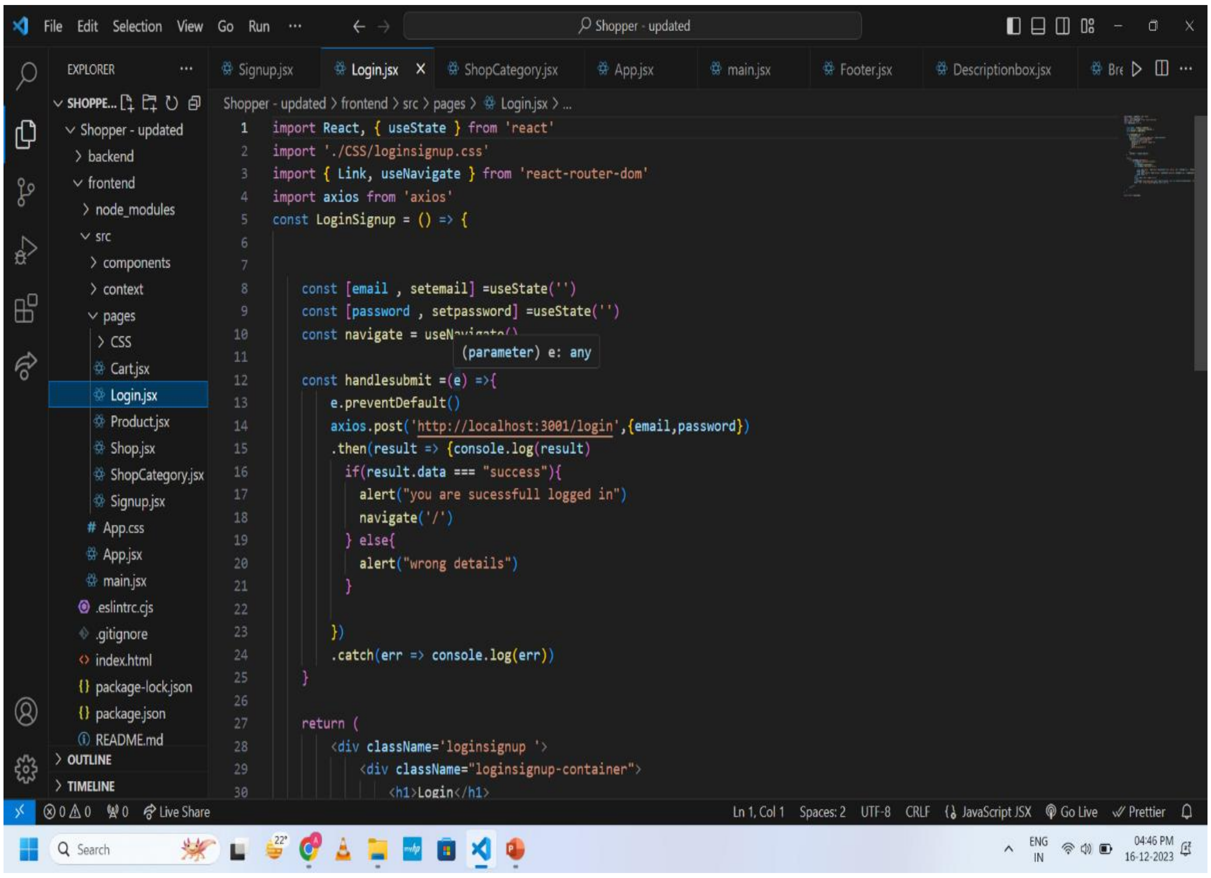
```jsx
import React from 'react'
import Cartitems from '../components/cartitems/Cartitems'

const Cart = () => {
    return (
        <div>
            <Cartitems />
        </div>
    )
}

export default Cart
```

Fig 4.5 SignUp.jsx



```jsx
import React, { useState } from 'react'
import './CSS/loginsignup.css'
import { Link, useNavigate } from 'react-router-dom'
import axios from 'axios'
const LoginSignup = () => {


    const [name , setname] =useState('')
    const [email , setemail] =useState('')
    const [password , setpassword] =useState('')
    const navigate = useNavigate()

    const handlesubmit =(e) =>{
        e.preventDefault()
        axios.post('http://localhost:3001/signup',{name,email,password})
        .then(result => {console.log(result)
            alert("Sign up success , now Login")
        navigate('/login')

        })
        .catch(err => console.log(err))
    }

    return (
        <div className='loginsignup '>
            <div className="loginsignup-container">
                <h1>Sign up</h1>
                <form onSubmit={handlesubmit}>
                <div className="loginsignup-fields">
                    <input name='name'  type="text" placeholder='your name' onChange={ (e) => setname(e.target.
```

Fig 4.6 Login.jsx

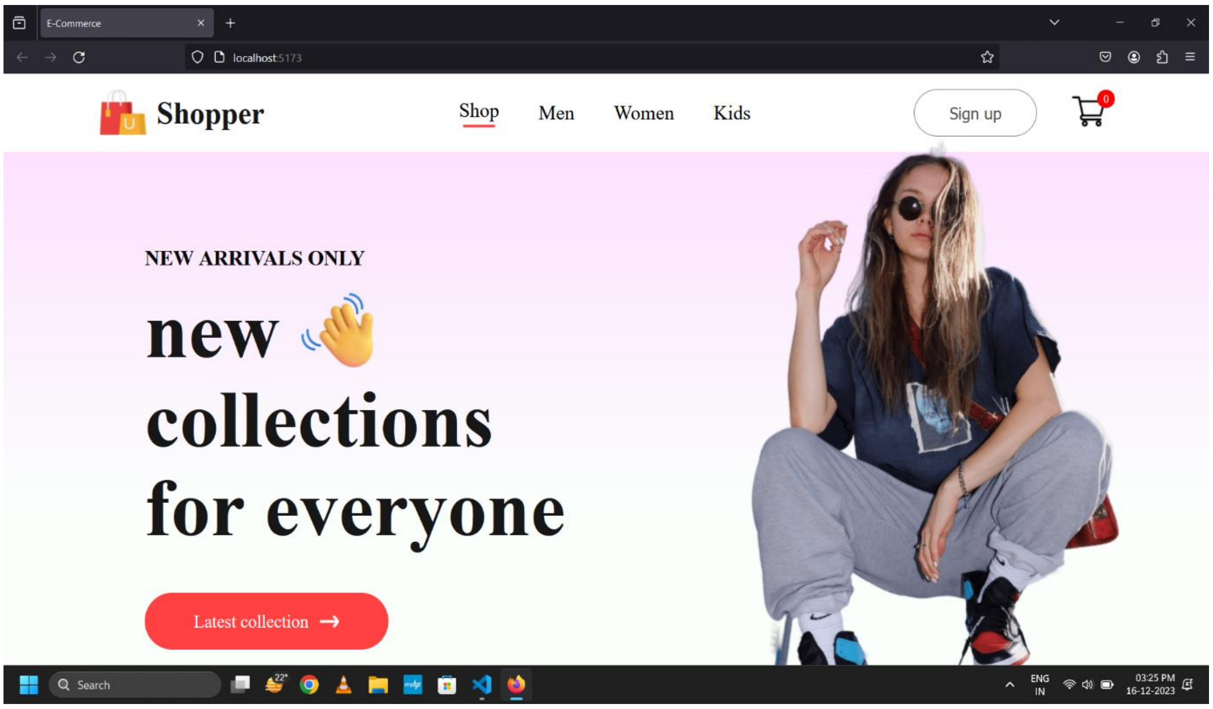


```
import React, { useState } from 'react'
import './CSS/loginsignup.css'
import { Link, useNavigate } from 'react-router-dom'
import axios from 'axios'
const LoginSignup = () => {


    const [email , setemail] =useState('')
    const [password , setpassword] =useState('')
    const navigate = useNavigate()

    const handlesubmit =(e) =>{
        e.preventDefault()
        axios.post('http://localhost:3001/login',{email,password})
        .then(result => {console.log(result)
          if(result.data === "success"){
            alert("you are sucessfull logged in")
            navigate('/')
          } else{
            alert("wrong details")
          }

        })
        .catch(err => console.log(err))
    }

    return (
        <div className='loginsignup '>
            <div className="loginsignup-container">
                <h1>Login</h1>
```

CHAPTER 5

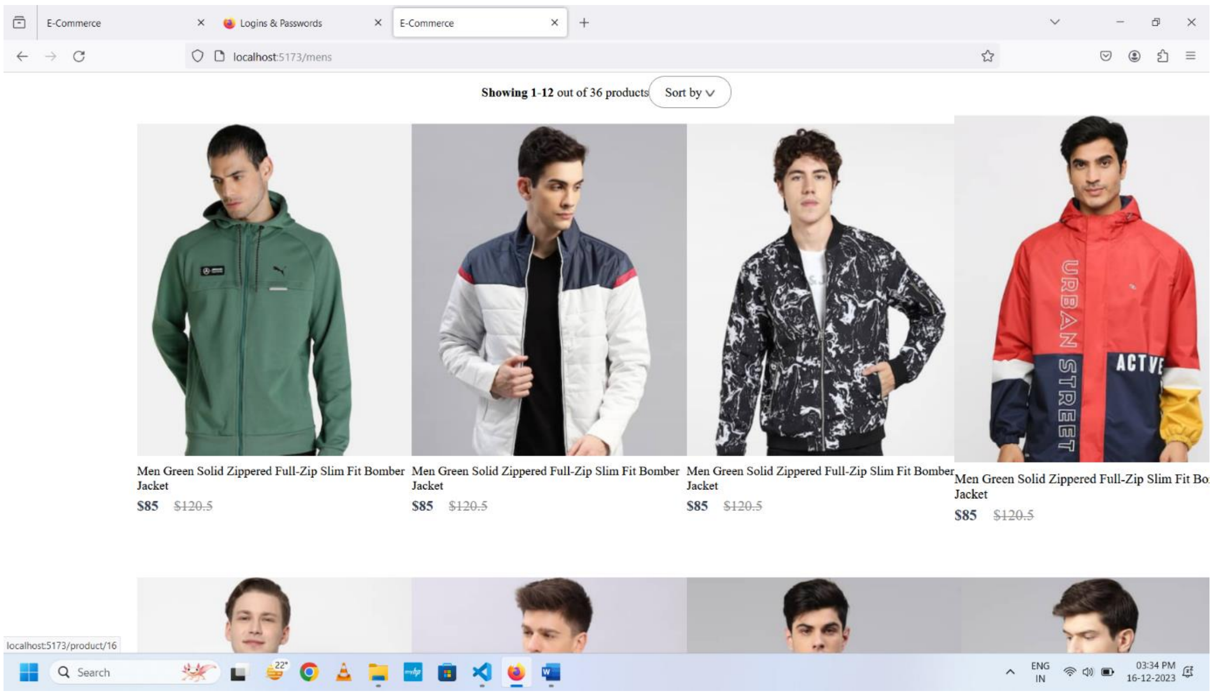
PROJECT SNAPSHOTS

5.1 HOMEPAGE

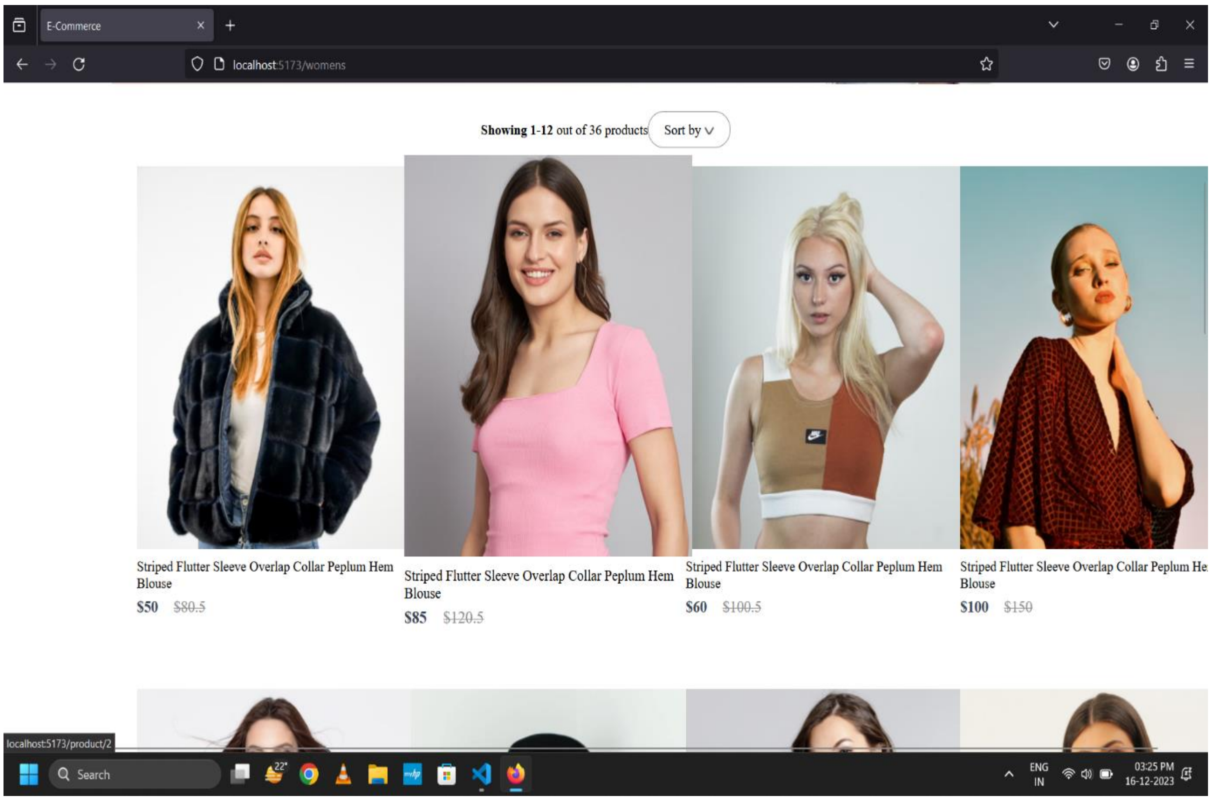


5.2 PRODUCT CATALOG

5.2.1 Mens Category


E-Commerce
Logins & Passwords
localhost:5173/mens
Showing 1-12 out of 36 products
Sort by
Men Green Solid Zippered Full-Zip Slim Fit Bomber Jacket
$85 $120.5
Men Green Solid Zippered Full-Zip Slim Fit Bomber Jacket
$85 $120.5
Men Green Solid Zippered Full-Zip Slim Fit Bomber Jacket
$85 $120.5
Men Green Solid Zippered Full-Zip Slim Fit Bo
Jacket
$85 $120.5
localhost:5173/product/16

5.2.2 Female Category



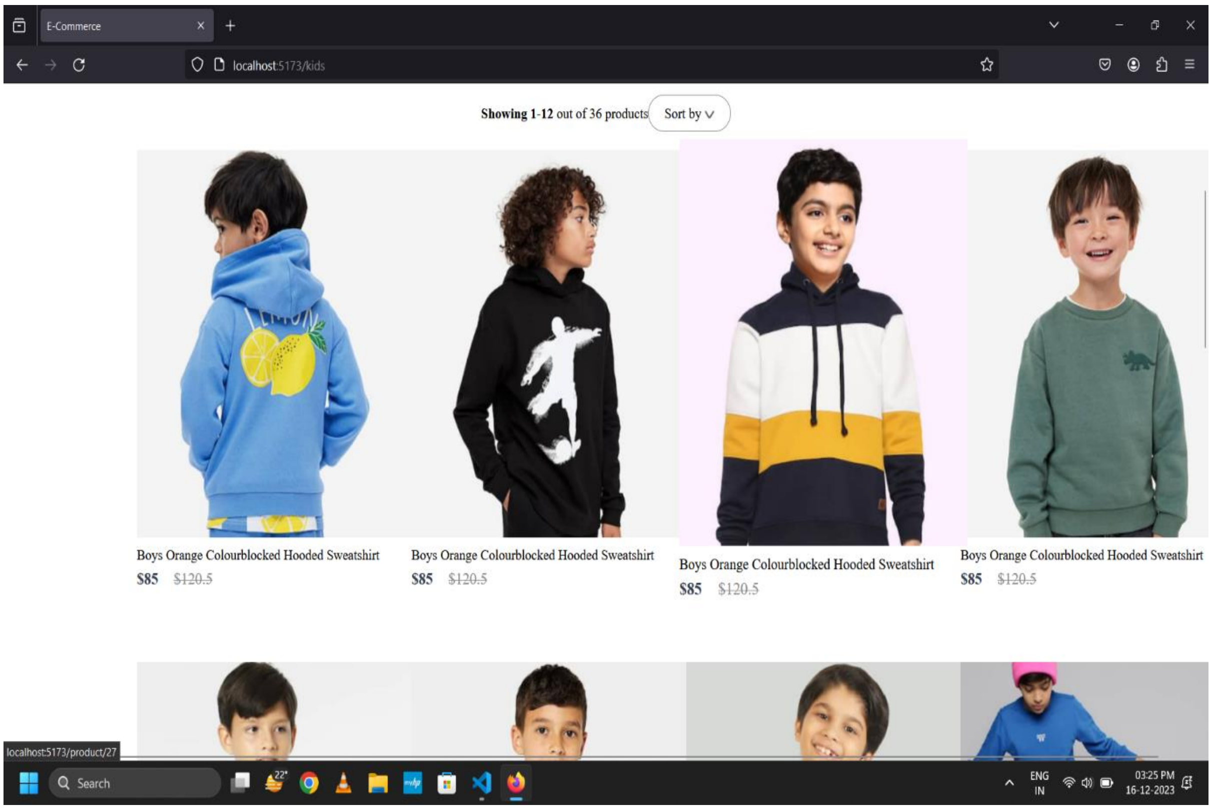
Striped Flutter Sleeve Overlap Collar Peplum Hem Blouse
$50 ~~$80.5~~

Striped Flutter Sleeve Overlap Collar Peplum Hem Blouse
$85 ~~$120.5~~

Striped Flutter Sleeve Overlap Collar Peplum Hem Blouse
$60 ~~$100.5~~

Striped Flutter Sleeve Overlap Collar Peplum He Blouse
$100 ~~$150~~

5.2.3 Kids Category



Boys Orange Colourblocked Hooded Sweatshirt
$85 ~~$120.5~~

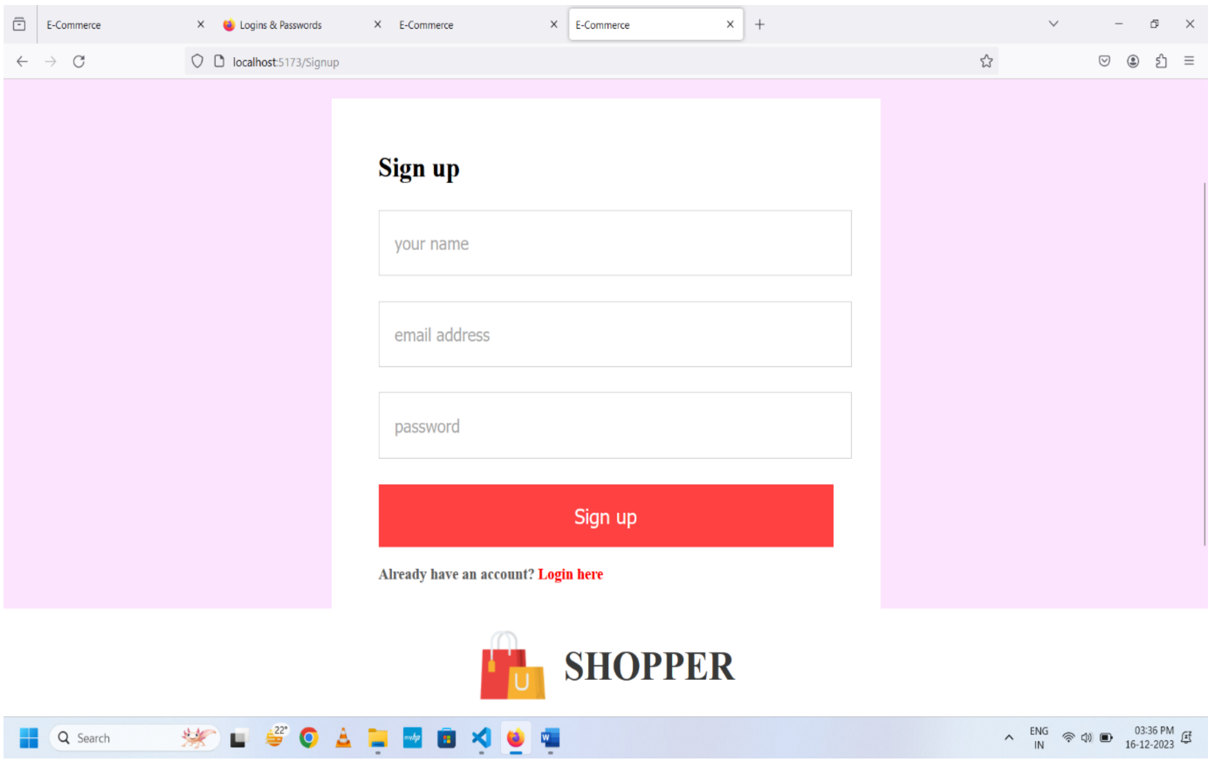
Boys Orange Colourblocked Hooded Sweatshirt
$85 ~~$120.5~~

Boys Orange Colourblocked Hooded Sweatshirt
$85 ~~$120.5~~

Boys Orange Colourblocked Hooded Sweatshirt
$85 ~~$120.5~~

5.3 USER AUTHENTICATION SCREEN

5.3.1 SignUp Page



Sign up

your name

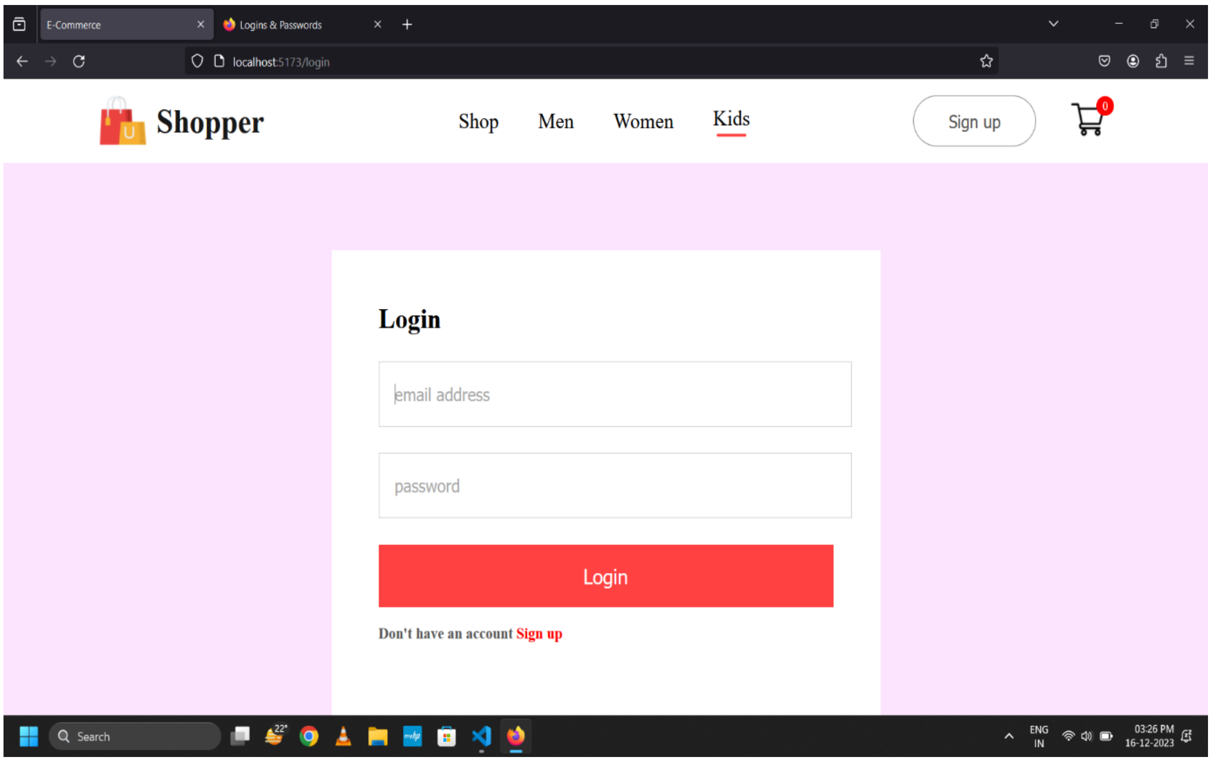
email address

password

Sign up

Already have an account? Login here

5.3.2 Login Page



Login

email address

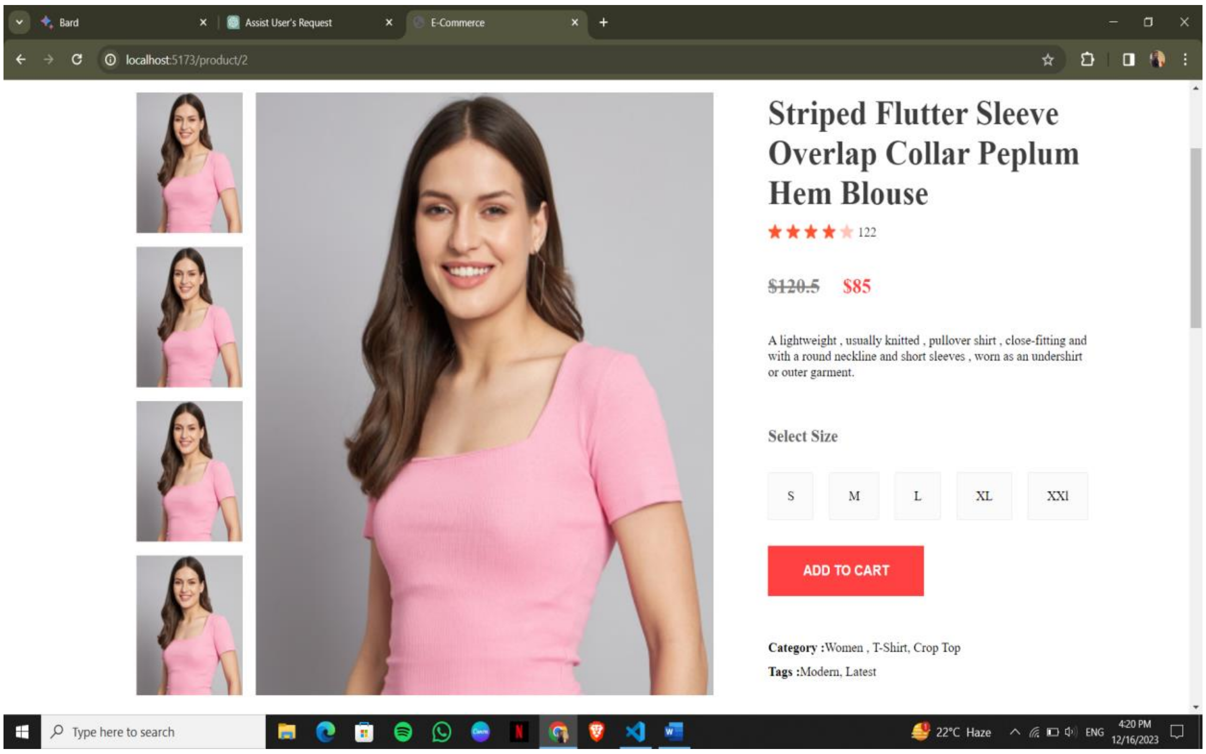
password

Login

Don't have an account Sign up

5.4 SHOPPING CART

5.4.1 Add to Cart Page



Striped Flutter Sleeve Overlap Collar Peplum Hem Blouse

★★★★☆ 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.

Select Size

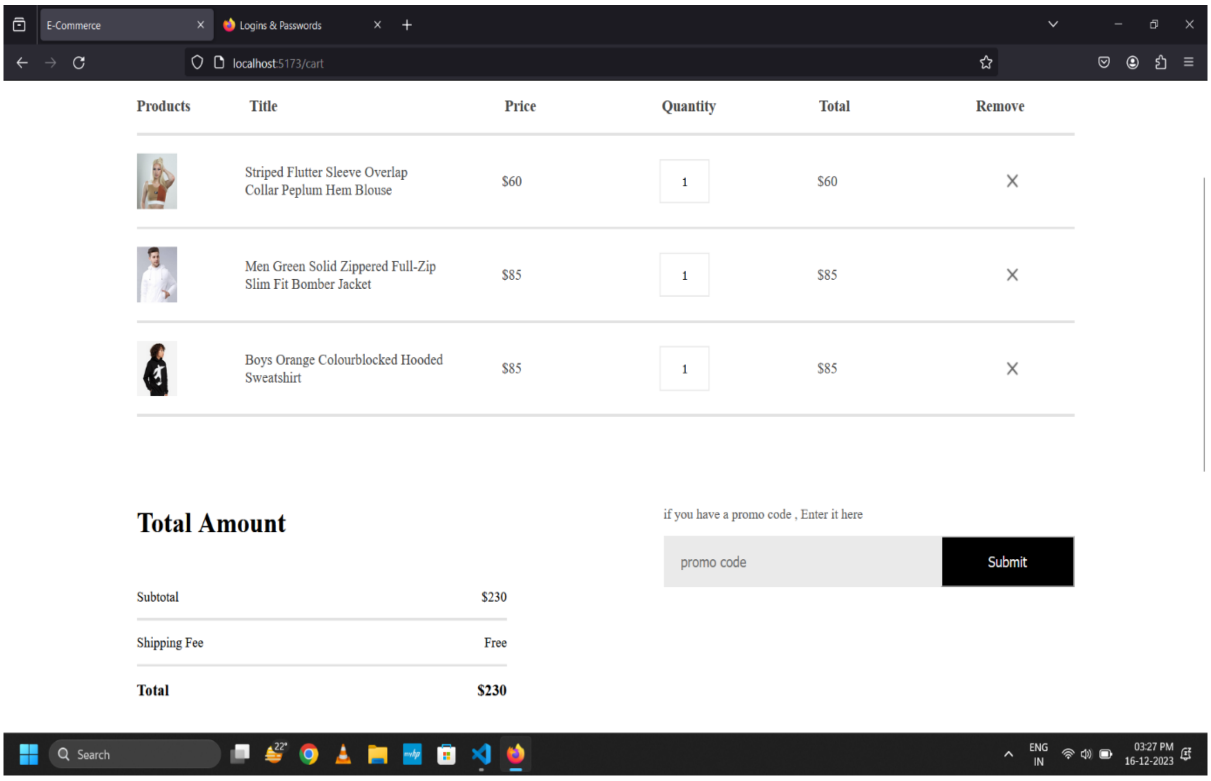
S M L XL XXl

ADD TO CART

Category : Women , T-Shirt, Crop Top

Tags : Modern, Latest

5.4.2 Checkout Page of Total Cart Amount



Products	Title	Price	Quantity	Total	Remove
	Striped Flutter Sleeve Overlap Collar Peplum Hem Blouse	$60	1	$60	×
	Men Green Solid Zippered Full-Zip Slim Fit Bomber Jacket	$85	1	$85	×
	Boys Orange Colourblocked Hooded Sweatshirt	$85	1	$85	×

Total Amount

Subtotal	$230
Shipping Fee	Free
Total	**$230**

if you have a promo code , Enter it here

promo code Submit

CHAPTER 6
LIMITATIONS

6.1 Technical Limitations

6.1.1 Scalability Challenges

Despite efforts to ensure scalability, the "Shopper" website may face challenges in handling a sudden surge in user traffic. This section discusses potential bottlenecks and outlines strategies for addressing scalability issues.

6.1.2 Performance Concerns

Technical limitations may manifest in performance issues, particularly during peak usage periods. This section examines potential performance bottlenecks and explores optimizations to enhance the overall speed and responsiveness of the website.

6.1.3 Integration Complexity

Integrating third-party services or future enhancements might pose technical challenges. This subsection addresses potential complexities in integration and suggests strategies to streamline the process.

6.2 Usability Limitations

6.2.1 User Experience Challenges

Despite efforts to create an intuitive design, the "Shopper" website may still face usability challenges. This section identifies potential issues related to navigation, layout, or accessibility that could impact the overall user experience.

6.2.2 Device Compatibility

The website's responsiveness might vary across different devices and browsers. This subsection highlights potential usability limitations related to device compatibility and proposes strategies to enhance cross-device functionality.

6.2.3 Learning Curve for Users

Users, especially those unfamiliar with online shopping, may encounter challenges in navigating the website. This part discusses potential usability barriers and suggests improvements to minimize the learning curve.

6.3 Security Limitations

6.3.1 Vulnerabilities and Threats

This section outlines potential security vulnerabilities such as data breaches, SQL injection, or cross-site scripting. It discusses the proactive measures taken to mitigate these threats and emphasizes the importance of ongoing security assessments.

6.3.2 Data Encryption Concerns

Despite encryption measures in place, there may be concerns related to data security during transmission. This subsection explores encryption challenges and proposes enhancements to strengthen data protection mechanisms.

6.3.3 User Authentication Risks

Issues related to user authentication, such as weak passwords or session management, may pose security risks. This part discusses potential vulnerabilities and suggests improvements to fortify the user authentication process.

CHAPTER 7
FUTURE SCOPE

"Future Scope," envisions the trajectory of the "Shopper" website beyond its current state. The narrative explores avenues for growth, including scaling strategies to accommodate a burgeoning user base, the integration of advanced technologies such as augmented reality and machine learning for an enhanced user experience, and the potential expansion of product categories to cater to a wider audience. Emphasizing the importance of continuous improvement in security, the chapter also envisions the implementation of enhanced security measures to safeguard user data and protect against emerging threats. Additionally, considering the prevalence of mobile devices, the exploration of mobile application development is discussed as a strategic move to further engage users.

CHAPTER 8
CONCLUSION

"Conclusion," the project's achievements and milestones are succinctly recapped, reflecting on challenges encountered and the strategies deployed to overcome them. Gratitude is expressed to contributors, team members, and stakeholders for their roles in the project's success. The impact of the "Shopper" website on the online shopping landscape is contemplated, and final thoughts encapsulate the journey of developing the platform, its significance, and the promising potential it holds for the future of E-commerce.

REFERENCES

Frontend :

GreatStackDev :

YouTube: https://www.youtube.com/@GreatStackDev

Backend :

CodeWithHarry:

YouTube. https://www.youtube.com/c/CodeWithHarry

Sign Up and Login :

codingstar6809:

YouTube. https://www.youtube.com/@codingstar6809

Websites : React js docs , express js docs , etc