

---

# AMAZON CLONE REPORT

---



MOUNIKA BHARGAVI GIRIDI

# **ABSTRACT**

Nowadays, the world has moved online and Amazon is a prime example of a website with all the key elements making up a good e-commerce site. Earlier amazon was put together using HTML, CSS and JavaScript . But as the time passed and when the different frameworks came into the spotlight , the website got updated .We are building an e-commerce application using MERN Stack, which is inspired by Amazon. Through this project, we have discovered a way to build a functional clone of Amazon's e-commerce website.

# CONTENTS

<b>S. N o</b>	<b>Topics</b>	<b>Page. No</b>
	<b>List of Figures</b>	3
	<b>Abstract</b>	1
1	<b>Introduction</b>	
	1.1 Motivation	4
	1.2 Objective of the Project	4
	1.3 Problem Statement	4
2	<b>Existing System</b>	5
	2.1 Literature survey	
3	<b>Proposed Methodology</b>	
	3.1 System Specifications	6
	3.2 System Design	6
	3.3 Proposed Work	7
4	<b>Implementation and Results</b>	8
5	<b>Conclusion and Future Scope</b>	35
	<b>Bibliography</b>	36

## List of Figures

Figure No	Name of the Figure	Page No
3.1	Flow chart	3
4.1	Fig 4.1	5
4.2	Fig 4.2	5
4.3	Fig 4.3	6
4.4	Fig 4.4	6
4.5	Fig 4.5	7
4.6	Front Page	7
4.7	Login Page	8
4.8	User login/registration form	8
4.9	Submitting details and resume	9
4.10	Result Page	9
4.11	Adminlogin/registration form	10
4.12	Opening Resumes	10
4.13	Folder where admin can view parsed resumes	11

# 1. INTRODUCTION

## 1.1 MOTIVATION:

We wanted to build an e-commerce application using MERN Stack , which is inspired by Amazon. We wanted to know how the Amazon website has been built with various features. Our project aim is to provide a theoretical contribution in understanding the present status of online shopping.

## 1.2 OBJECTIVE OF THE PROJECT:

Businesses, in any field, have a lot of competition. They are always on the lookout for a proven way to increase their business revenue. Restaurants, retail shops, and vape stores - whatever a business may want to sell, if the business doesn't have an e-commerce website, they are leaving money on the table! The world has moved online – a fact that businesses have to accept and put up a website to address. Amazon is a prime example of a website with all the key elements making up a good e-commerce site. The e-commerce website of Amazon was initially put together with simple HTML, CSS and JavaScript. But as time progressed and different frameworks came into the limelight, the website got a makeover. So, our objective is to build a functional clone of the Amazon e-commerce website, relying on the MERN stack.

## 1.3 PROBLEM STATEMENT:

We are implementing a project which is a user-friendly Amazon-clone. Our project allows users to save time by shopping different categories through a wide range of wholesale retailers.

## 2.

## EXISTING SYSTEM

### 2.1 LITERATURE SURVEY:

The e-commerce website of Amazon was initially put together with simple HTML, CSS and JavaScript. But as time progressed and different frameworks came into the limelight, the website got a makeover. Our project provides insight into the phases involved in development of a low complexity website, we did it by choosing the easiest programming language to read and write in ReactJs since it's fast and efficient . MongoDB which is document based Database software for zero configuration and maintenance. The low complexity and zero to low maintenance website.

### 3. PROPOSED METHODOLOGY

#### 3.1 SYSTEM SPECIFICATIONS:

Operating System: Windows 10

Languages used: In order to run in VScode

and modify this program on your personal machine, you will need to have installed the following apps

- VS Code
- React JS
- Node JS
- Express JS
- MongoDB

#### 3.2 SYSTEM DESIGN:

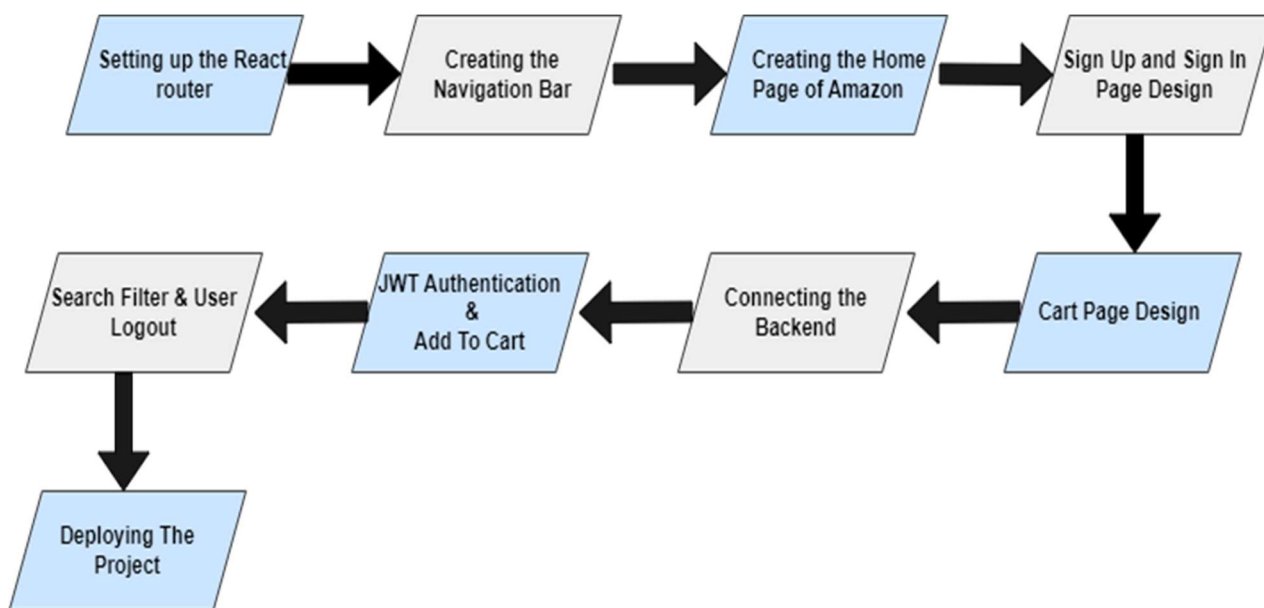


Fig 3.1: Flow chart

### 3.3 PROPOSED WORK:

**This proposed project consists of:**

- **Homepage:**  
Homepage is the default page of the website. It consists of a navigation bar, all the deals related to products and a footer page. In the navigation bar, there is a search icon, sign in button, cart icon and profile button. And the footer consists of information about us.
- **Products:**  
The Product page consists of details about the products. Whenever we click on a particular product then it displays all information related to that product.
- **Sign In/ Sign Up:**  
If the user is new to the website then first he/she must create an account by filling up all the credentials and after registering he/she can login in into their accounts using their email and password given during registration.
- **Cart Page:**  
Cart page displays all the products added to the cart by the user along with the total amount. Here the user even have an option to delete the products
- **Order Page:**  
Order page displays the products which are ready to place the order.



## 4. IMPLEMENTATION AND RESULTS

### 1. Here are the steps that are used in the project:

01. home page
02. footer
03. create account
04. sign-in
05. profile
06. cart
07. order
08. products

### 01. Homepage:

Homepage is the default page of the website. It consists of a navigation bar, all the deals related to products and a footer page. In the navigation bar, there is a search icon, sign in button, cart icon and profile button. And the footer consists of information about us.

In here we used some special commands and packages for creating the header which are login symbol, avatar, search etc..

```
import Carousel from 'react-bootstrap/Carousel';
```

The carousel is a slideshow for cycling through a series of content, built with CSS 3D transforms and a bit of JavaScript. It works with a series of images, text, or custom markup. It also includes support for previous/next controls and indicators.

we used this carousel for image sliding in the home page and we used the multi carousel to slide multiple slide at one section

```
import Slider from './Slider';
```

```
import './home.css';
```

```
JS connection.js Home.jsx Card.jsx Banner.jsx X
client > src > components > home > Banner.jsx > ...
1  import React from 'react';
2  import Carousel from 'react-bootstrap/Carousel';
3
4  const Banner = () => {
5    return (
6      <div className='banner'>
7        <Carousel
8          variant='dark'
9          indicators={false}
10         >
11
12          <Carousel.Item>
13            <div className='carousel-img-wrapper'>
14              
19            </div>
20          </Carousel.Item>
21
22          <Carousel.Item>
23            <div className='carousel-img-wrapper'>
24              
29            </div>
30          </Carousel.Item>
31
32          <Carousel.Item>
33            <div className='carousel-img-wrapper'>
```

Fig 4.1 - home page

In the Banner section we divided the categories like laptops ,smartwatches etc..

For that we imported the card sections like row ,column

```
import React from 'react';
```

```
import Container from 'react-bootstrap/Container';
```

```
import Row from 'react-bootstrap/Row';
```

```
import Col from 'react-bootstrap/Col';
```

```
import Card from './Card';
```

```
import MultiCard from './MultiCard';
```

React-Bootstrap is a front-end framework that was designed keeping react in mind. We can use the following approach in ReactJS to use the react-bootstrap Container, Row, Col Component.

- **Container** Component provides a way to center and horizontally pad the contents of our application. It is used when the user wants the responsive pixel width.
- **Row** Component provides a way to represent a row in the grid system. It is used when we want to display data in the form of rows.
- **Col** Component provides a way to represent a column in the grid system. It is used when we want to display data in the form of columns

```

JS connection.js Home.jsx Card.jsx CategoryCards.jsx X Banner.jsx
client > src > components > home > CategoryCards.jsx > ...
1  import React from 'react';
2  import Container from 'react-bootstrap/Container';
3  import Row from 'react-bootstrap/Row';
4  import Col from 'react-bootstrap/Col';
5  import Card from './Card';
6  import MultiCard from './MultiCard';
7  import './home.css';
8
9  const Cards = () => {
10   return (
11     <div className='cards-section'>
12       <div className='cards-container container-fluid'>
13         <div className='row cards'>
14           <div className='col-6 col-sm-4 col-xl-3 multi-card'>
15             <MultiCard name="Shop by Category" img="category-1" a="Smartwatches" b="Ta
16           </div>
17           <div className='col-6 col-sm-4 col-xl-3'>
18             <Card name="Health and Personal Care" img="category-2" bottom="Shop now"
19           </div>
20           <div className='col-6 col-sm-4 col-xl-3'>
21             <Card name="Get Fit at Home" img="category-3" bottom="Explore now" />
22           </div>
23           <div className='col-6 col-sm-4 col-xl-3'>
24             <Card name="Shop Mother's Day Gifts" img="category-4" bottom="Shop now" />
25           </div>
26           <div className='col-6 col-sm-4 col-xl-3'>
27             <Card name="Computer & Accessories" img="category-5" bottom="Shop now" />
28           </div>
29           <div className='col-6 col-sm-4 col-xl-3'>

```

**Cards** are surfaces that display content and actions on a single topic.

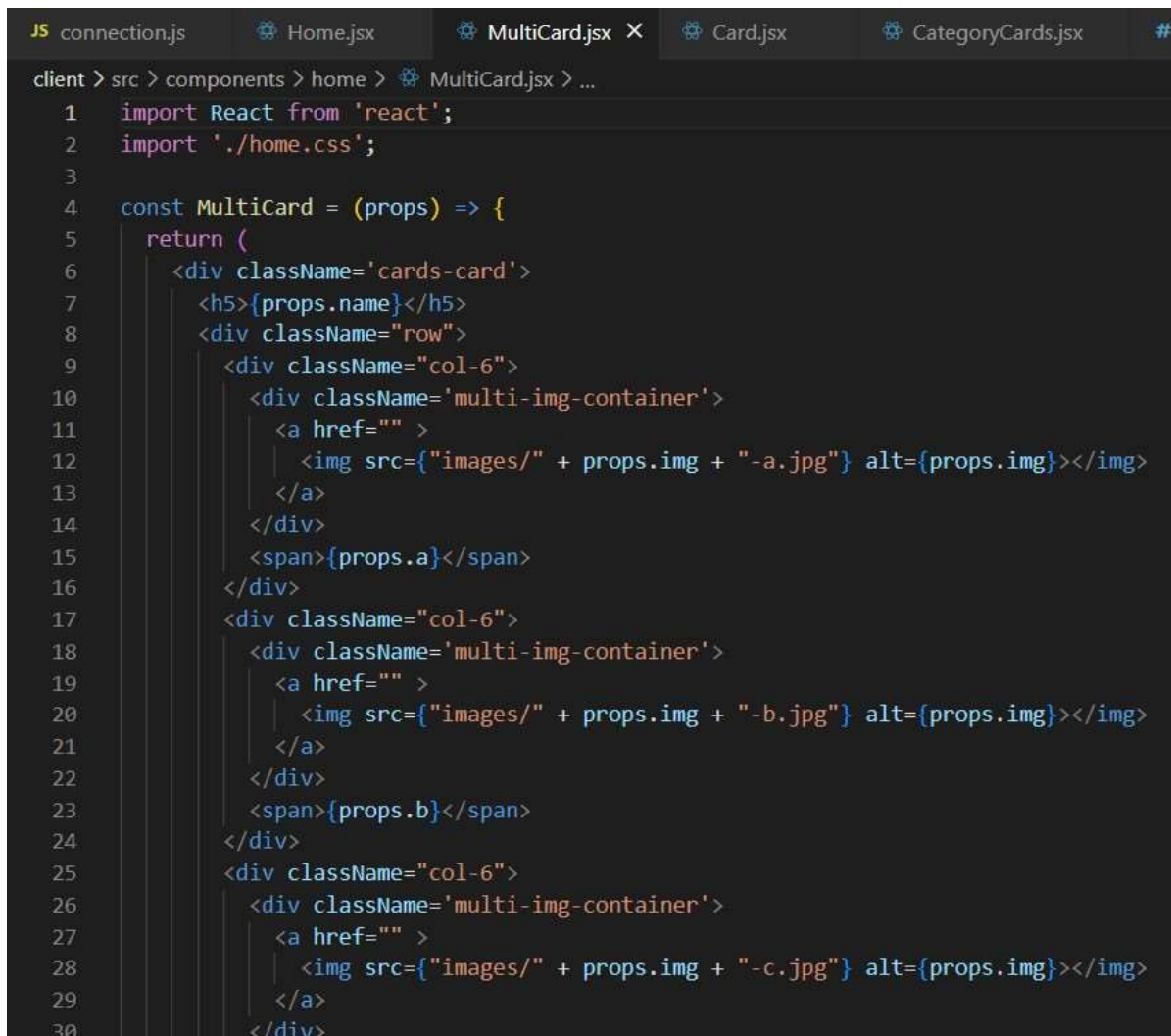
They should be easy to scan for relevant and actionable information. Elements, like text and images, should be placed on them in a way that clearly indicates hierarchy.

Although cards can support multiple actions, UI controls, and an overflow menu, use restraint and remember that cards are entry points to more complex and detailed information.

## Outlined Card

Set variant="outlined" to render an outlined card.

```
<Card variant="outlined">{card}</Card>
```



```
client > src > components > home > MultiCard.jsx > ...
1  import React from 'react';
2  import './home.css';
3
4  const MultiCard = (props) => {
5    return (
6      <div className='cards-card'>
7        <h5>{props.name}</h5>
8        <div className="row">
9          <div className="col-6">
10             <div className='multi-img-container'>
11               <a href="" >
12                 <img src={"images/" + props.img + "-a.jpg"} alt={props.img}></img>
13               </a>
14             </div>
15             <span>{props.a}</span>
16           </div>
17           <div className="col-6">
18             <div className='multi-img-container'>
19               <a href="" >
20                 <img src={"images/" + props.img + "-b.jpg"} alt={props.img}></img>
21               </a>
22             </div>
23             <span>{props.b}</span>
24           </div>
25           <div className="col-6">
26             <div className='multi-img-container'>
27               <a href="" >
28                 <img src={"images/" + props.img + "-c.jpg"} alt={props.img}></img>
29               </a>
30             </div>
```

## 02. footer

In the footer section we gave the information about amazon clone and what are the business the site with ,contact details etc..

A footer is an important element of a website's design since it signals to the user that they have reached the end of the webpage and provides useful links to other areas of the website that the user may want to visit.

### Prerequisite:

- Basic knowledge of *npm* & *create-react-app* command.
- Basic knowledge of styled-components.

```
npx create-react-app react-footer
```

Now go to your **react-footer** folder by typing the given command in the terminal:

```
cd react-footer
```

**Required module:** Install the dependencies required in this project by typing the given command in the terminal.

```
npm install --save styled-components
```

Now create the **components** folder in **src** then go to the components folder and create two files by the name **Footer.js** and **FooterStyles.js**

```
client > src > components > footer > ⚙ Footer.jsx > ...
 1  import React from 'react';
 2  import SignIn from './SignIn';
 3  import TopFooter from './TopFooter';
 4  import MiddleFooter from './MiddleFooter';
 5  import BottomFooter from './BottomFooter';
 6  import './footer.css';
 7
 8  const Footer = () => {
 9
10    return (
11      <footer>
12        <SignIn />
13        <TopFooter />
14        <MiddleFooter />
15        <BottomFooter />
16      </footer>
17    )
18  }
19
20  export default Footer;
```

```
import React from 'react';

import LanguageIcon from '@mui/icons-material/Language';

import $ from 'jquery';

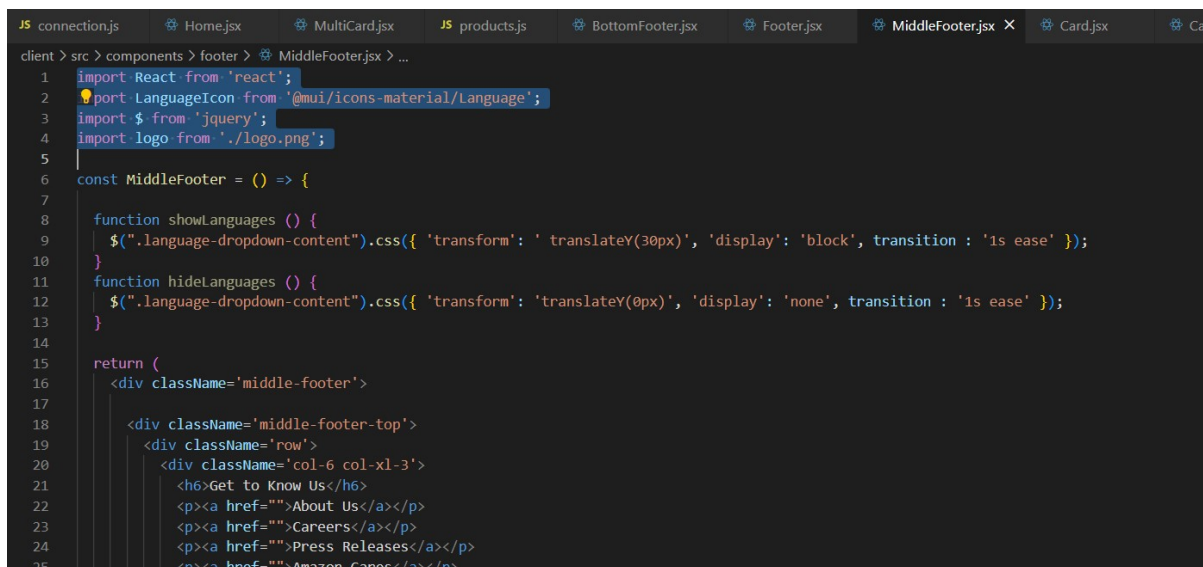
import logo from './logo.png';
```

Include popular icons in your React projects easily with react-icons, which utilizes ES6 imports that allows you to include only the icons that your project is using.

## Installation (for standard modern project)

```
npm install react-icons --save
```

Icons in the computing world are defined as **pictorial representations** or **ideograms** displayed on a computer screen that conveys a meaning. Icons convey characteristics that separate them from each other and make a thing exist uniquely. Under all of these aspects, they are also called symbols popularly that quickly comprehend thoughts about a software tool, a function, a file, a data. Icons represent real-world entities that contain detailed illustrations and are designed so that one symbol speaks it all.



```
JS connection.js Home.jsx MultiCard.jsx JS products.js BottomFooter.jsx Footer.jsx MiddleFooter.jsx X Card.jsx Cat
client > src > components > footer > MiddleFooter.jsx > ...
1 import React from 'react';
2 import LanguageIcon from '@mui/icons-material/Language';
3 import $ from 'jquery';
4 import logo from './logo.png';
5
6 const MiddleFooter = () => {
7
8   function showLanguages () {
9     $(".language-dropdown-content").css({ 'transform': 'translateY(30px)', 'display': 'block', 'transition': '1s ease' });
10  }
11  function hideLanguages () {
12    $(".language-dropdown-content").css({ 'transform': 'translateY(0px)', 'display': 'none', 'transition': '1s ease' });
13  }
14
15  return (
16    <div className='middle-footer'>
17      <div className='middle-footer-top'>
18        <div className='row'>
19          <div className='col-6 col-xl-3'>
20            <h6>Get to Know Us</h6>
21            <p><a href=''>About Us</a></p>
22            <p><a href=''>Careers</a></p>
23            <p><a href=''>Press Releases</a></p>
24            <p><a href=''>Amazon Cares</a></p>
```

## 03. create account

Users and Sellers can register into the website and login with their e-mail or phone number and password. Window-shopping doesn't require any kind of authentication. **For Signup page the imported packages:**



```

import React, { useState } from 'react';

import axios from 'axios';

import CountryCode from './CountryCode';

import './login-register.css';

import ArrowRightIcon from '@mui/icons-material/ArrowRight';

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

import Alert from '@mui/material/Alert/Alert';

import AlertTitle from '@mui/material/AlertTitle/AlertTitle';

```

This package take country-list as reference, and make it more friendly to react-select

Maps ISO 3166-1-alpha-2 codes to English country names and match react-select options props.

## Install

```
npm install react-select-country-list --save
```

or

```
yarn add react-select-country-list
```

```

<Alert variant="outlined" className='alert success-alert'>Registered successfully! Please <NavLink
to='/login'>login</NavLink></Alert>

<div className='form-details'>

  <h3>Create Account</h3>

  <form method='POST' onSubmit={ sendData }>

    <label htmlFor='name'>Your name</label>

    <input type='text' name='name' id='name' placeholder='First and last name' onChange={ formUpdate } value={
signUpInfo.name } required />

```

```

<label htmlFor='number'>Mobile number</label>

<div className='mobile-number'>

  <CountryCode />

  <input type='text' name='number' id='number' placeholder='Mobile number' onChange={ formUpdate } value={
signUpInfo.number } required />

</div>

<label htmlFor='email'>Email</label>

  <input type='email' name='email' id='email' placeholder='Email Address' onChange={ formUpdate } value={
signUpInfo.email } required />

<label htmlFor='password'>Password</label>

  <input type='password' name='password' id='password' placeholder='Password (At least 6 characters)' onChange={
formUpdate } value={ signUpInfo.password } required />

<label htmlFor='confirmPassword'>Confirm Password</label>

  <input type='password' name='confirmPassword' id='confirmPassword' placeholder='Confirm Password' onChange={
formUpdate } value={ signUpInfo.confirmPassword } required />

<button type='submit' id='submit'>Continue</button>

</form>

```

## 04. sign-in



For sign page ,we imported

```
import React, { useState } from 'react';  
  
import './login-register.css';  
  
import { NavLink, useNavigate } from 'react-router-dom';  
  
import Alert from '@mui/material/Alert/Alert';  
  
import AlertTitle from '@mui/material/AlertTitle/AlertTitle';  
  
import axios from 'axios';
```

**Alerts** are urgent interruptions, requiring acknowledgment, that inform the user about a situation. Material UI for React has this component available for us, and it is very easy to integrate. We can use Alert Component in ReactJS using the following approach.

### Creating React Application And Installing Module:

**Step 1:** Create a React application using the following command.

```
npx create-react-app foldername
```

**Step 2:** After creating your project folder i.e. foldername, move to it using the following command.

```
cd foldername
```

**Step 3:** After creating the ReactJS application, Install the **material-ui** modules using the following command.

```
npm install @material-ui/core
```

```
npm install @material-ui/lab
```

**React MUI** is a UI library providing predefined robust and customizable components for React for easier web development. The MUI design is based on Material Design by Google.

In this article, we are going to discuss the **React MUI AlterTitle API**. The Alert component allows the user to show important tips or messages on the page.

### Import AlterTitle API:

```
import Alert from '@mui/material/Alert';
```

```
import AlertTitle from '@mui/material/AlertTitle';
```

**Approach:** Let us create a React project and install React MUI module. Then we will create a UI that will showcase React MUI AlertTitle API.

## Creating React Project:

**Step 1:** To create a react app, you need to install react modules through npx command. “npx” is used instead of “npm” because you will be needing this command in your app’s lifecycle only once.

```
npx create-react-app project_name
```

**Step 2:** After creating your react project, move into the folder to perform different operations.

```
cd project_name
```

**Step 3:** After creating the ReactJS application, Install the required module using the following command:

```
npm install @mui/material @emotion/react @emotion/styled
```

```
npm install @mui/lab
```

The **<Link>** and **<NavLink>** are the components provided by react-router-dom to navigate around the react application. Generally, we use anchor tags for this purpose while navigating. Then what’s the difference? Let’s find out.

Anchor tags will reload the page and re-render all the components. While **<Link>** and **<NavLink>** will only re-render updated components matched with the URL path of the Route without reloading. It helps the Single-Page Applications to work faster while routing.

### **<Link> Component Props:**

- **to:** String or object which specifies the pathname.
- **replace:** Replaces the pathname in the history stack with new.
- **innerRef:** Passes ref to the element rendered by the component.

## Modules Required:

- npm
- create-react-app
- react-router-dom

## Creating React Application And Installing Module:

**Step 1:** Create a new react application by the following command using terminal:

```
npx create-react-app <project_name>
```

**Step 2:** Go to the project folder by the following command:

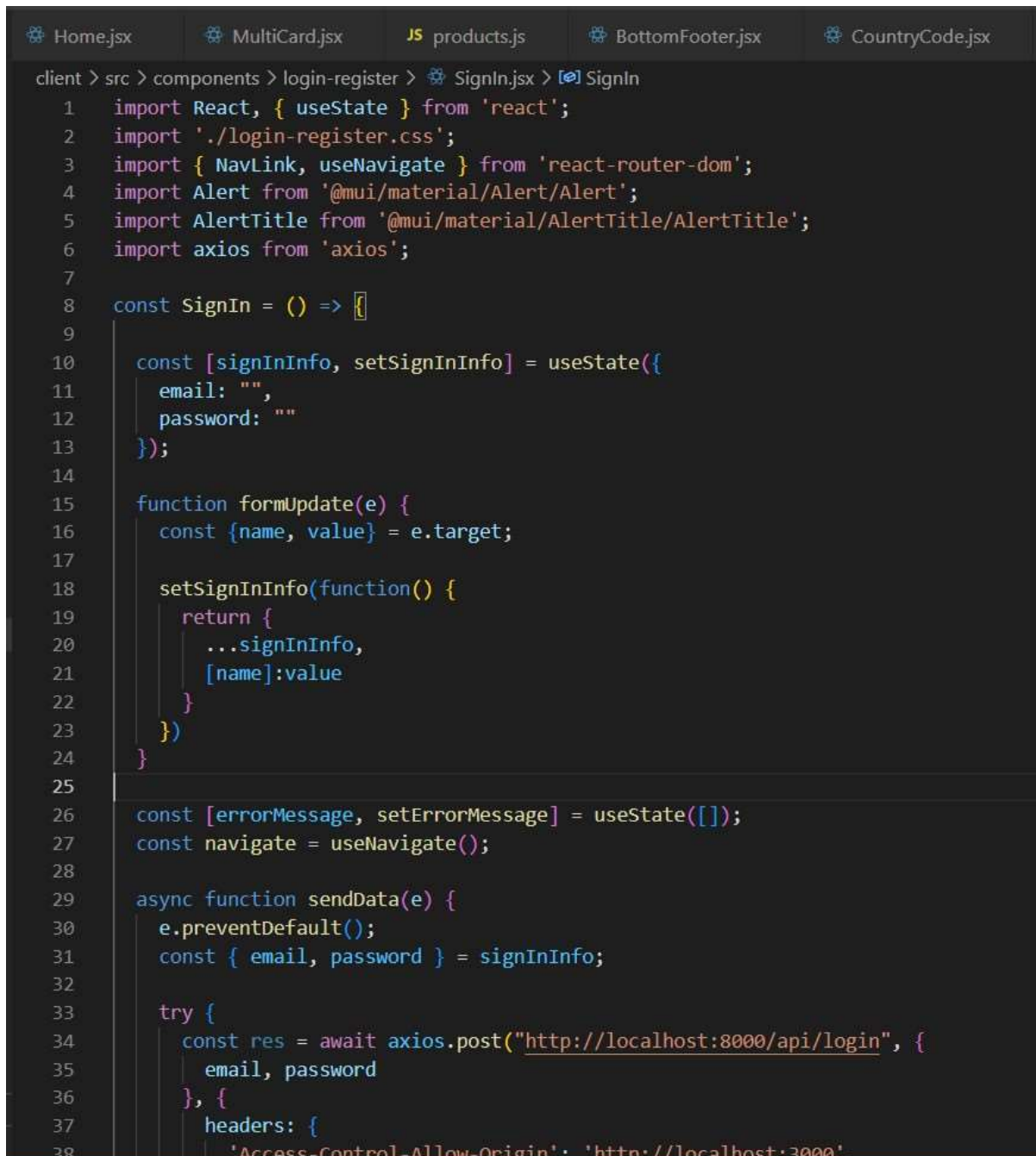
```
cd <project_name>
```

**Step 3:** Install dependency react-router-dom using the following command:

```
npm install react-router-dom
```

**Note:** To check whether the dependency has been installed or not , go to package.json and checkout in the dependencies.

**Step 4:** Create a new folder named components in the **src** folder and add **Home.js**, **About.js** and **Contact.js** files to it.



```
client > src > components > login-register > SignIn.jsx > SignIn
1  import React, { useState } from 'react';
2  import './login-register.css';
3  import { NavLink, useNavigate } from 'react-router-dom';
4  import Alert from '@mui/material/Alert/Alert';
5  import AlertTitle from '@mui/material/AlertTitle/AlertTitle';
6  import axios from 'axios';
7
8  const SignIn = () => {
9
10     const [signInInfo, setSignInInfo] = useState({
11       email: "",
12       password: ""
13     });
14
15     function formUpdate(e) {
16       const {name, value} = e.target;
17
18       setSignInInfo(function() {
19         return {
20           ...signInInfo,
21           [name]:value
22         }
23       })
24     }
25
26     const [errorMessage, setErrorMessage] = useState([]);
27     const navigate = useNavigate();
28
29     async function sendData(e) {
30       e.preventDefault();
31       const { email, password } = signInInfo;
32
33       try {
34         const res = await axios.post("http://localhost:8000/api/login", {
35           email, password
36         }, {
37           headers: {
38             'Access-Control-Allow-Origin': 'http://localhost:3000'
```

## 05. Profile

```
import React, { useEffect, useState } from 'react';
```

```
import NameBanner from './NameBanner';

import UserDetails from './UserDetails';

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

import './profile.css';

import axios from 'axios';

import Loader from '../loader/Loader';
```

**Name banner** - It is a simple line text banner component (like warning, error, success). Styles (font and background) can be customized. By default the banner has a fade in and fade out animation of 2s. This can be used in two ways, banner can be shown for a specific amount of time using visibleTime prop, or can be handled manually by making the title text null or changing the string.

Install it from npm and include it in your React build process (using Webpack, Browserify, etc).

```
npm i react-js-banner
```

Import Banner in your react component.

```
import Banner from 'react-js-banner';
```

First things You should have a react app created already and install react-router-dom from npm (just for this example, you don't need react-router-dom for useContext). Now create a file in your source folder and name it userDetails.jsx this is the file that will be creating the context.

```
const Profile = () => {

  const [isLoading, setIsLoading] = useState(true);
  const [userData, setUserData] = useState();

  const navigate = useNavigate();

  useEffect(function() {
    async function fetchUser() {
      try {
        const res = await axios.get("http://localhost:8000/api/getAuthUser", {
          withCredentials: true
        })

        if (res) {
          setUserData(res.data);
          setIsLoading(false);
        }
      } catch (error) {
        if (error.response.data.message === "No token provided") {
          navigate('/login');
        } else {
          console.log(error);
        }
      }
    }
  }, []);

  fetchUser();
}
```

```
client > src > components > profile > UserDetails.jsx > ...
1  import React from 'react';
2  import './profile.css';
3  import PersonIcon from '@mui/icons-material/Person';
4  import EmailIcon from '@mui/icons-material/Email';
5  import PhoneIphoneIcon from '@mui/icons-material/PhoneIphone';
6
7  const UserDetails = (props) => {
8    return (
9      <div className='user-details'>
10        <div className='user-detail'>
11          <PersonIcon className='icon' />
12          <h5>{props.user.name}</h5>
13        </div>
14        <div className='user-detail'>
15          <EmailIcon className='icon' />
16          <h5>{props.user.email}</h5>
17        </div>
18        <div className='user-detail'>
19          <PhoneIphoneIcon className='icon' />
20          <h5>{props.user.number}</h5>
21        </div>
22      </div>
23    )
24  }
25
26  export default UserDetails;
```

## 06. Cart

Packages imported in cart section:

```
import React, { useEffect, useState } from 'react';

import Loader from '../loader/Loader';

import axios from 'axios';

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

import Alert from '@mui/material/Alert';

import './cart.css';
```



```
import CartProduct from './CartProduct';
```

```
import SubTotal from './SubTotal';
```

## useCart

The `useCart` hook exposes all the getter/setters for your cart state. `setItems(items)` The `setItems` method should be used to set all items in the cart. This will overwrite any existing cart items. A quantity default of 1 will be set for an item implicitly if no quantity is specified.

```
// Creating an array of products ordered
const orderedProducts = [];

for (let i = 0; i < cartArr.length; i++) {
  let product = {
    id: cartArr[i].cartItem.id,
    name: cartArr[i].cartItem.name,
    qty: cartArr[i].qty,
    img: cartArr[i].cartItem.url
  }
  orderedProducts.push(product);
}

let orderAmount = 0;
for (let i = 0; i < cartArr.length; i++) {
  orderAmount += cartArr[i].qty * cartArr[i].cartItem.accValue;
}

function loadRazorpay() {
  const script = document.createElement("script");
  script.src="https://checkout.razorpay.com/v1/checkout.js";

  script.onerror = () => {
    alert("Razorpay SDK failed to load. Try again later");
  };
  script.onload = async () => {
    try {
      const res = await axios.post("http://localhost:8000/api/create-order", {
        amount: orderAmount + '00'
      });
    }
  }
}
```

## 07. orders

packages imported in order section:

```
import React, { useEffect, useState } from 'react';
```

```
import NameBanner from './NameBanner';

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

import './profile.css';

import axios from 'axios';

import 'bootstrap/dist/css/bootstrap.min.css';

import 'bootstrap/dist/js/bootstrap.bundle.min';

import OrderTop from './OrderTop';

import OrderedProduct from './OrderedProduct';

import Loader from './loader/Loader';
```

Higher-order components or HOC is the advanced method of reusing the component functionality logic. It simply takes the original component and returns the enhanced component.

### Syntax:

```
const EnhancedComponent = higherOrderComponent(OriginalComponent);
```

### Reason to use Higher-Order component:

- Easy to handle
- Get rid of copying the same logic in every component
- Makes code more readable

Using npm:

```
$ npm install react-ordering --save
```



```
# Navbar.css  Navbar.jsx  NameBanner.jsx  Profile.jsx  UserDetails.jsx  Cart.jsx  CartProduct.jsx  OrderTop.jsx
client > src > components > profile > OrderTop.jsx > ...
1  import React from 'react';
2  import './profile.css';
3
4  const OrderTop = (props) => {
5    let order = props.order;
6    let date = order.date;
7
8    let monthArr = ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November"];
9    let slash = 0;
10   let start = 0;
11   let monthStr = '';
12   let day = '';
13   let year = ''
14
15   for (let i = 0; i < date.length; i++) {
16     if (date[i] === '/') {
17       if (slash === 0) {
18         day = date.substring(start, i);
19         start = i + 1;
20         slash++;
21       } else if (slash === 1) {
22         monthStr = date.substring(start, i);
23         start = i + 1;
24         slash++;
25       }
26     } else if (slash === 2) {
27       year = date.substring(start, date.length);
28     }
29   }
30
31   let month = monthArr[parseInt(start) + 1];
32   let fullDate = day + " " + month + " " + year;
33 }
```

## 08.products

packages imported in product section:

```
import React, { useEffect, useState } from 'react';

import axios from 'axios';

import './product.css';

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

import Loader from '../loader/Loader';
```

```

const Product = () => {

  // Loader
  const [isLoading, setIsLoading] = useState(true);

  const {id} = useParams("");

  // Fetching individual product from API
  const [product, setProduct] = useState();

  useEffect(function() {
    async function fetchSingleProduct() {
      try {
        const res = await axios.get('http://localhost:8000/api/product/' + id);
        setProduct(res.data);
        setIsLoading(false);
      } catch (error) {
        console.log(error);
      }
    }

    fetchSingleProduct();
  }, [])

  // To navigate to a diff page
  const navigate = useNavigate();
  // Add to cart
  async function addToCart(id) {
    try {
      const res = await axios.post('http://localhost:8000/api/addtocart/' + id, {
        product
      }, {
        headers: {
          'Content-Type': 'application/json'
        },
      },

```

## 4.1 OUTPUT RESULT:

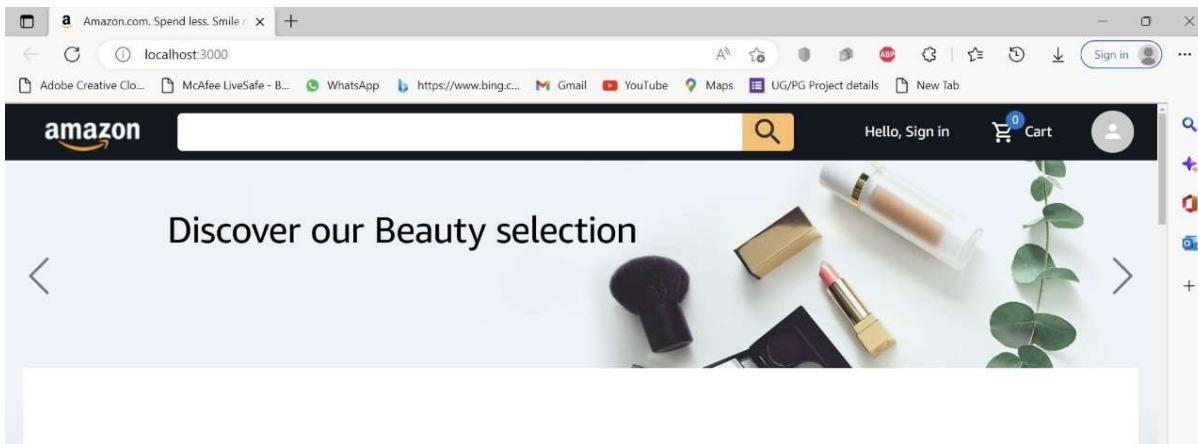


Fig 4.1.1 -homepage

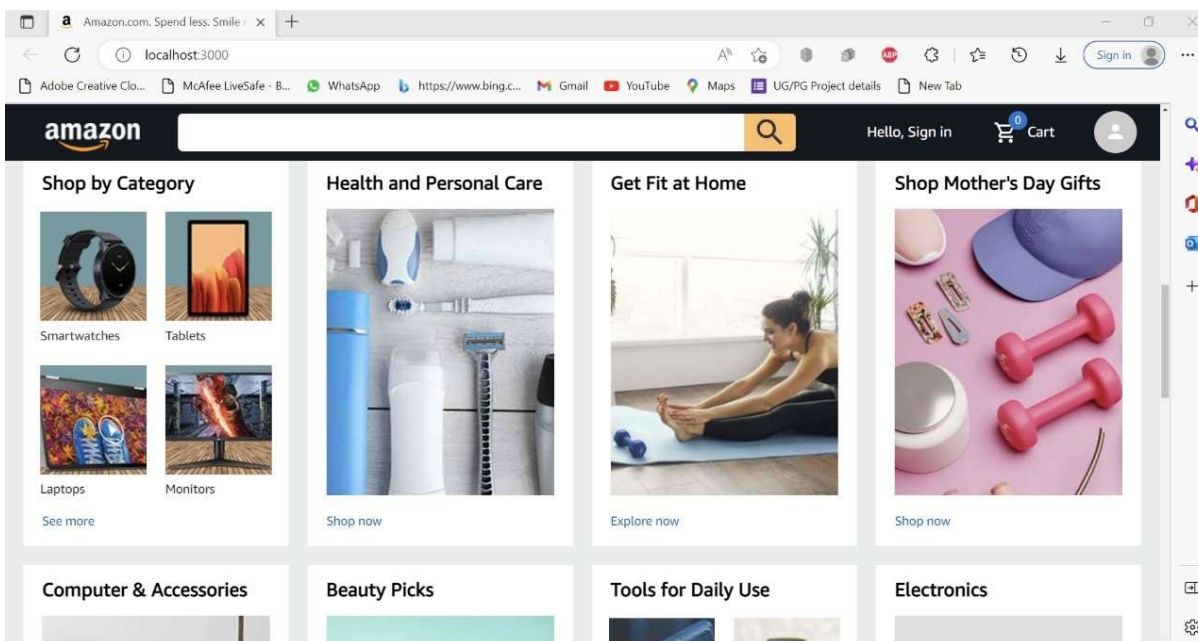


Fig 4.1.2 -products list

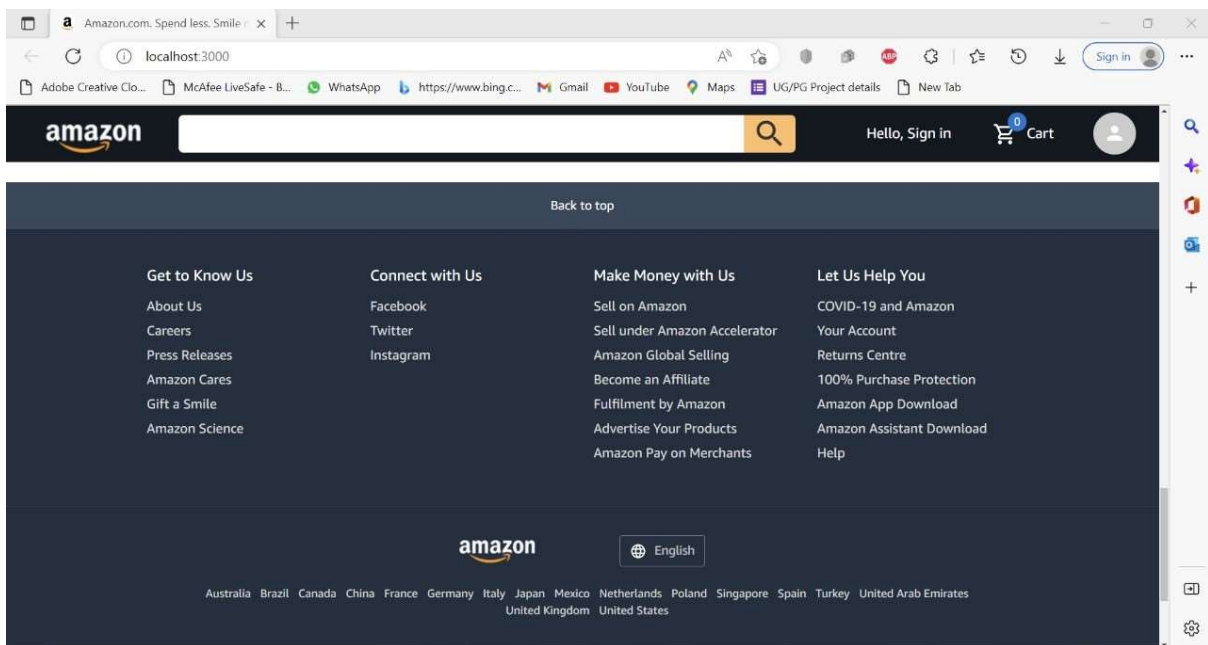


Fig 4.1.3 -Footer

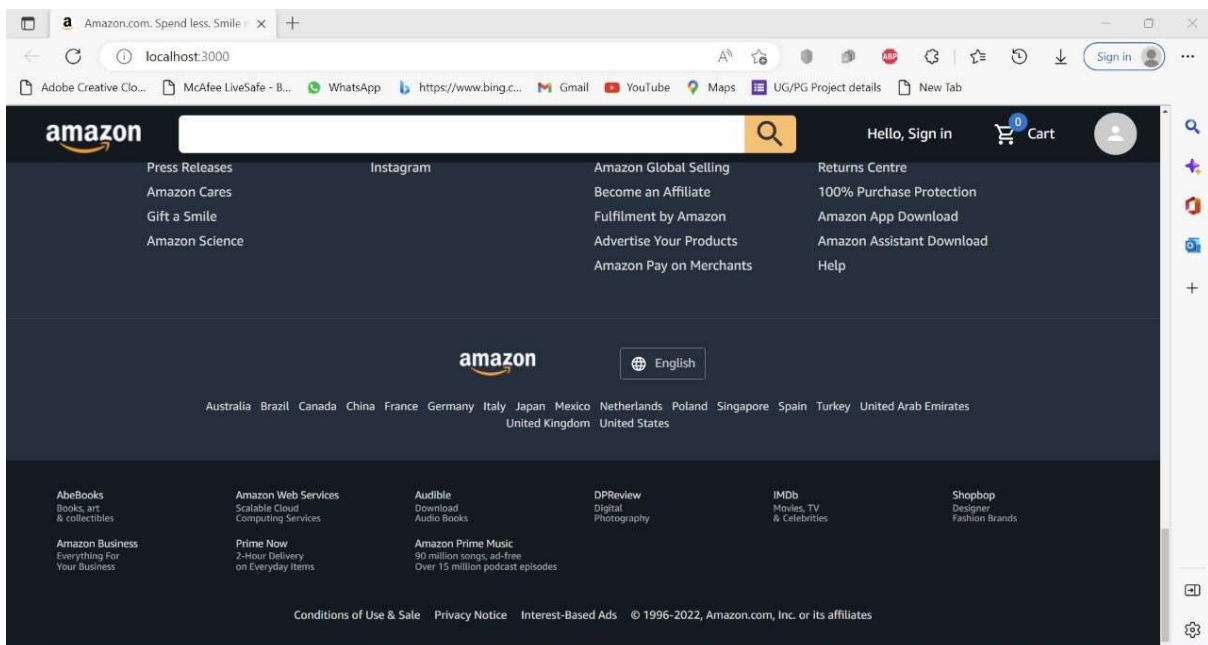


Fig 4.1.4 -amazon buiness

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/register'. The page features the Amazon logo at the top left. The main content is a 'Create Account' form with the following fields: 'Your name' (First and last name), 'Mobile number' (with a dropdown for 'India (+91)' and a text input for the number), 'Email' (Email Address), 'Password' (Password (At least 6 characters)), and 'Confirm Password' (Confirm Password). A yellow 'Continue' button is positioned below the 'Confirm Password' field. At the bottom of the form, there is a link that says 'Already have an account? Sign in >'. The browser's taskbar at the bottom shows several open applications, including Adobe Creative Cloud, McAfee LiveSafe, WhatsApp, and various web browsers.

Fig 4.1.5 -create account

This screenshot shows the same 'Create Account' form as in Fig 4.1.5, but with validation errors. A red-bordered box at the top of the form contains a warning icon and the text 'There were some errors'. The form fields are now populated with the following data: 'Your name' is 'kaira', 'Mobile number' is 'India (+91)' with the number '1234567891', 'Email' is 'kai@gmail.com', 'Password' is masked with '\*\*\*\*\*', and 'Confirm Password' is also masked with '\*\*\*\*\*'. The yellow 'Continue' button remains at the bottom of the form. The browser window and taskbar are identical to the previous figure.

Fig 4.1.6 -invalid details

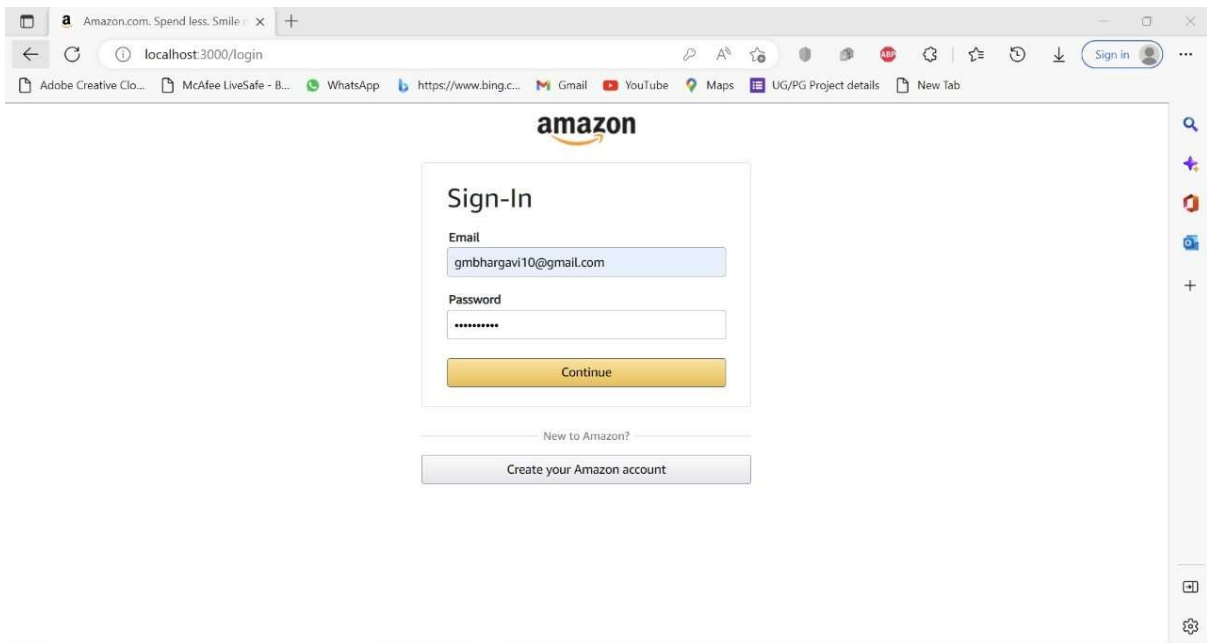


Fig 4.1.7 -Sign-in

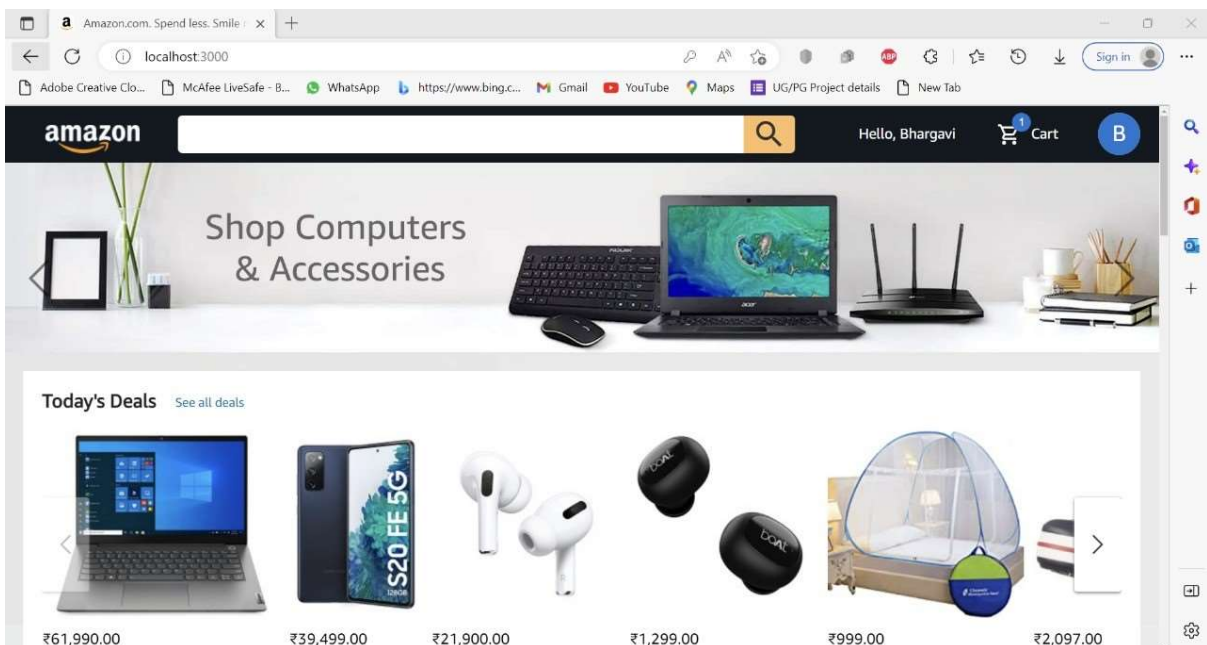


Fig 4.1.8 -after login

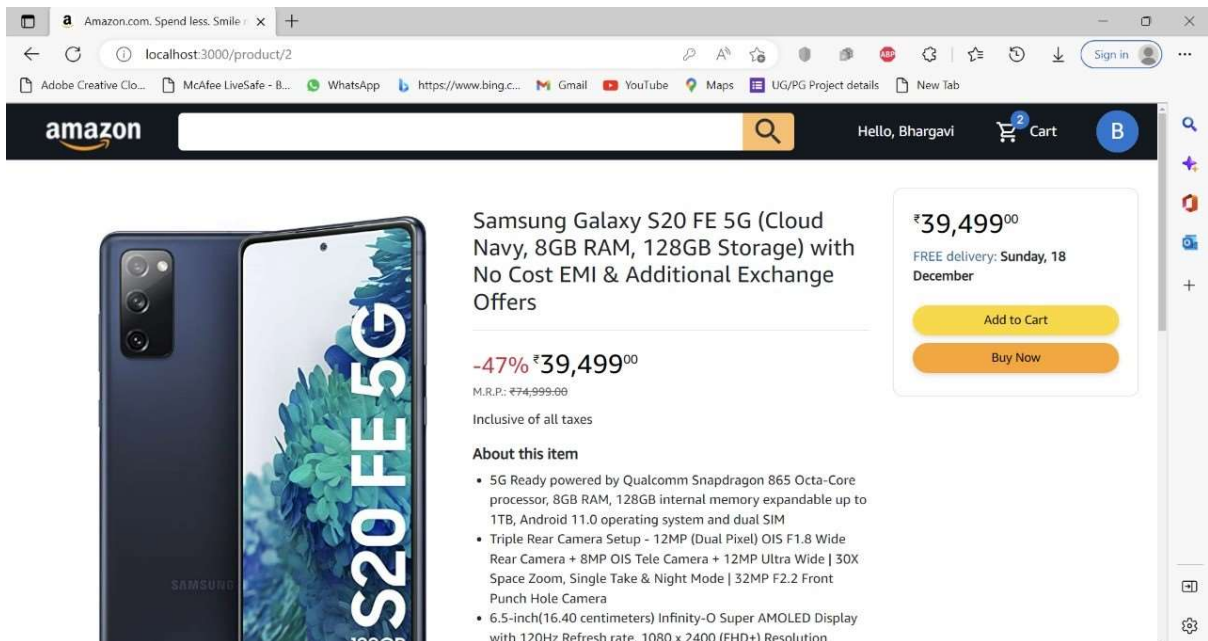


Fig 4.1.9 - product details

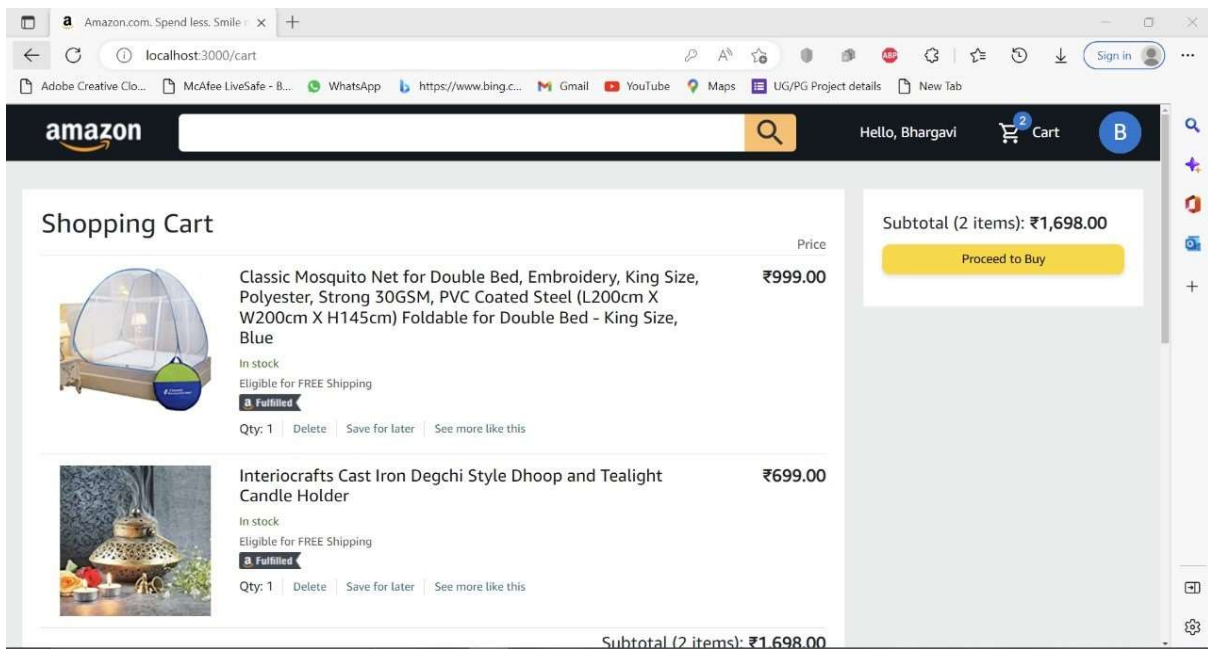


Fig 4.1.10 - proceed to buy



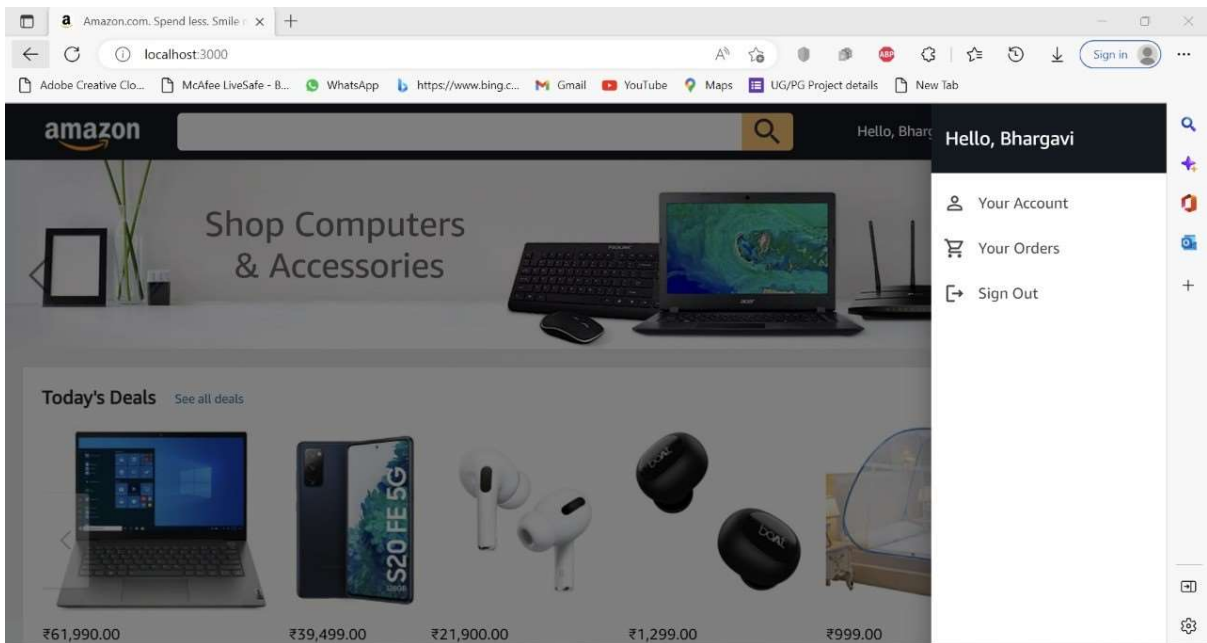


Fig 4.1.11 - profile, your orders and sign out

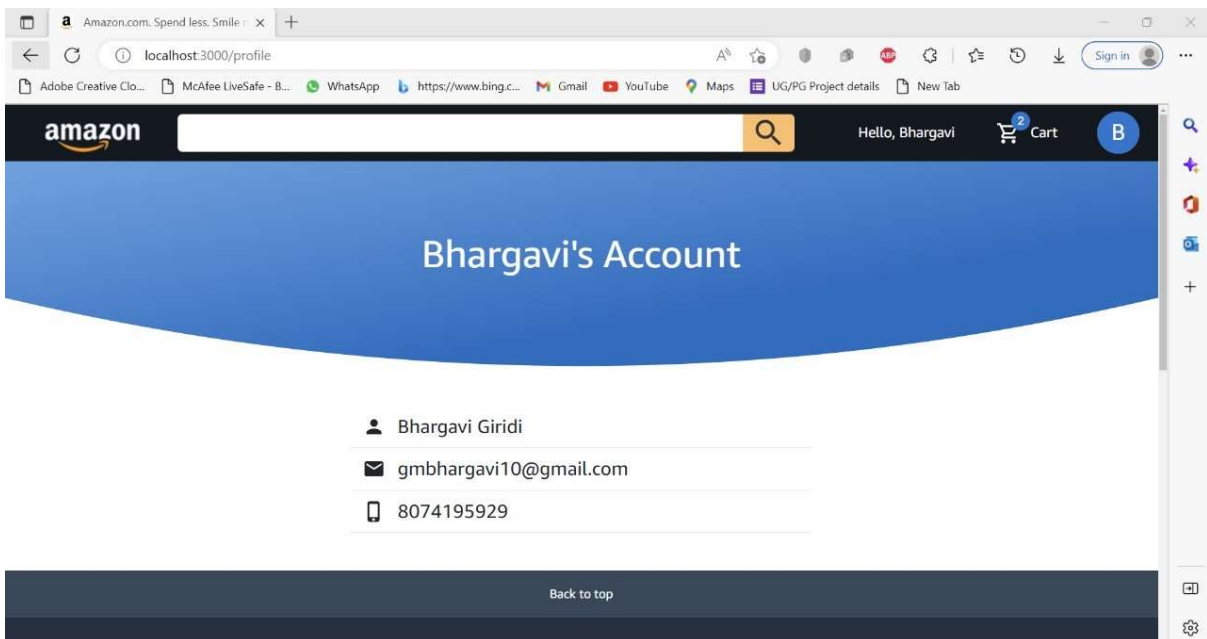


Fig 4.1.12 - user profile



## **5.CONCLUSION AND FUTURE SCOPE**

The system that we have developed with free of errors and at the same time it is efficient and less time consuming for shopping. The purpose of our project is to develop a web application for purchasing items from e-commerce websites(Amazon Clone). This project enabled us to gain valuable information and practical knowledge on several topics like designing web pages using React JS, usage of responsive templates and management of databases using MongoDB. Also, the project helped to understand about the development phases of a project and software development life cycle.

In future application we are willing to add few features:

-> payment mode

## **BIBLIOGRAPHY**

- [https://www.amazon.in/ref=nav\\_logo](https://www.amazon.in/ref=nav_logo)
- <https://medium.com/cleverprogrammer/amazon-clone-using-react-the-ultimate-guide-fba2b36f3458>
- <https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/dp-manage-pipeline-clone-console.html>