Front End Capstone Project
Date – 05.05.2023

Front End Capstone Project

Notes from Eskandar Atrakchi:

1. I have the project committed on my GitHub the link is here https://github.com/EskandarAtrakchi/Skillup_Burger_Shop
2. I have created an extra styling file for SCSS styling accordingly and I named it receipt.scss.
3. I have made a video showing how the final project looks like with its styling and functionalities, the video can be accessed here https://youtu.be/iVAhIdAXau4
4. If you want to run the project the dependencies need to be installed first
5. After (npm start) command I have the project running on http://localhost:3000/
6. The project has been zipped and can be accessed here https://mobidrive.com/sharelink/z/76LAZf5UnyWIsR0hFQhpg3623XCT4hWfMK7prYzYmqyP

Pre-requisites:

This project requires a basic knowledge of:

1. HTML, CSS
2. Node.JS & React
3. Using VS code

Level of Exercise: Beginner

Duration: Approx 10 hours

All exercises are required to be completed for the successful completion of this project Please return this file with the answers in it to <mail ID> on or before <date of submission> -

Objective:

You have an offline shop for your burger-selling business and want to create an online website for expanding your business. For this, you will create a website using React.js for users to buy burgers online.

Overview:

This Capstone Project is divided into 4 exercises:

Exercise 1:

- Creating the Home Pad its Header and Footer
- Creating the “Explore Menu” button for viewing the details of burgers available

Exercise 2:

- Creating the About and Contact pages for website visitors to get information and contact.
- Creating the Sign up and Login pages

Exercise 3:

- Creating the Cart page for selecting and placing orders
- Creating the Shipping page for completing the order details & confirming the order for shipping

Exercise 4:

- Creating the “My Orders” page which will have the order details
- Creating the “Order details” page which will show the complete details for each order

This project can be done in any suitable IDE (Integrated development environment), with VS Code being the recommended IDE.

If you do not have VS Code on your system already, you may install it from here:
https://code.visualstudio.com/download , and set it up on your system.

Setting up your project:

1. Please fork the below repo which contains the project codes

```

**https://github.com/kumar045/Skillup_Burger_Shop.git**

```

Note: Need to create this project on a General GitHub account and share its link instead of above for ensuring privacy and confidentiality of all other GitHub projects present on Shivam’s GitHub account.

2. Clone this repo to VS code/ your suitable IDE for working on the project.
3. You will add the codes as per the instructions for building each component of your online Burger application.

Exercise 1

1.1/ Task 1: Home Page

You will now be completing the Homepage of the Burger Shop project.

Instructions:

1. Go to the file - home/Home.jsx in the “components” folder.
2. Import all the necessary packages

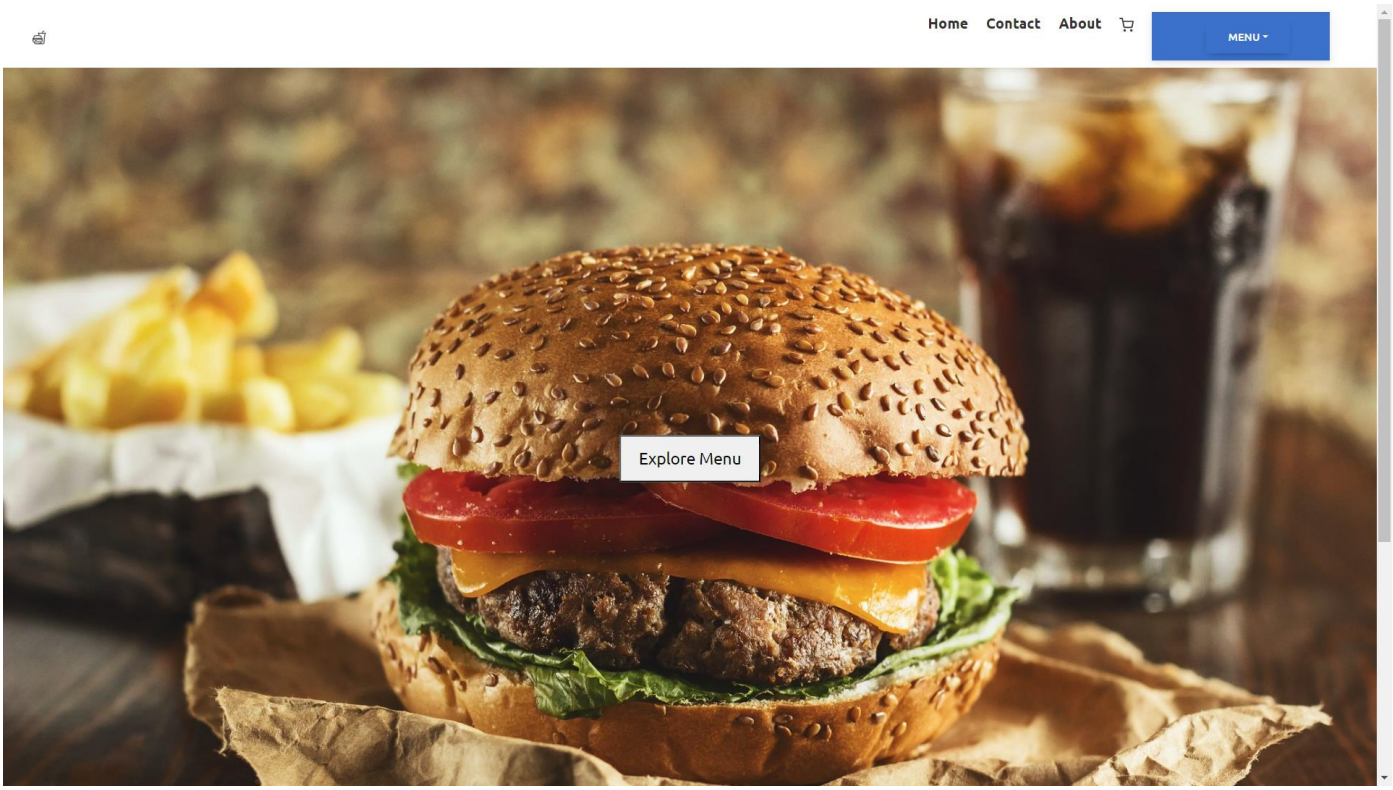
SOLUTION

```
import React, { useState } from "react";
import { motion } from "framer-motion";
```

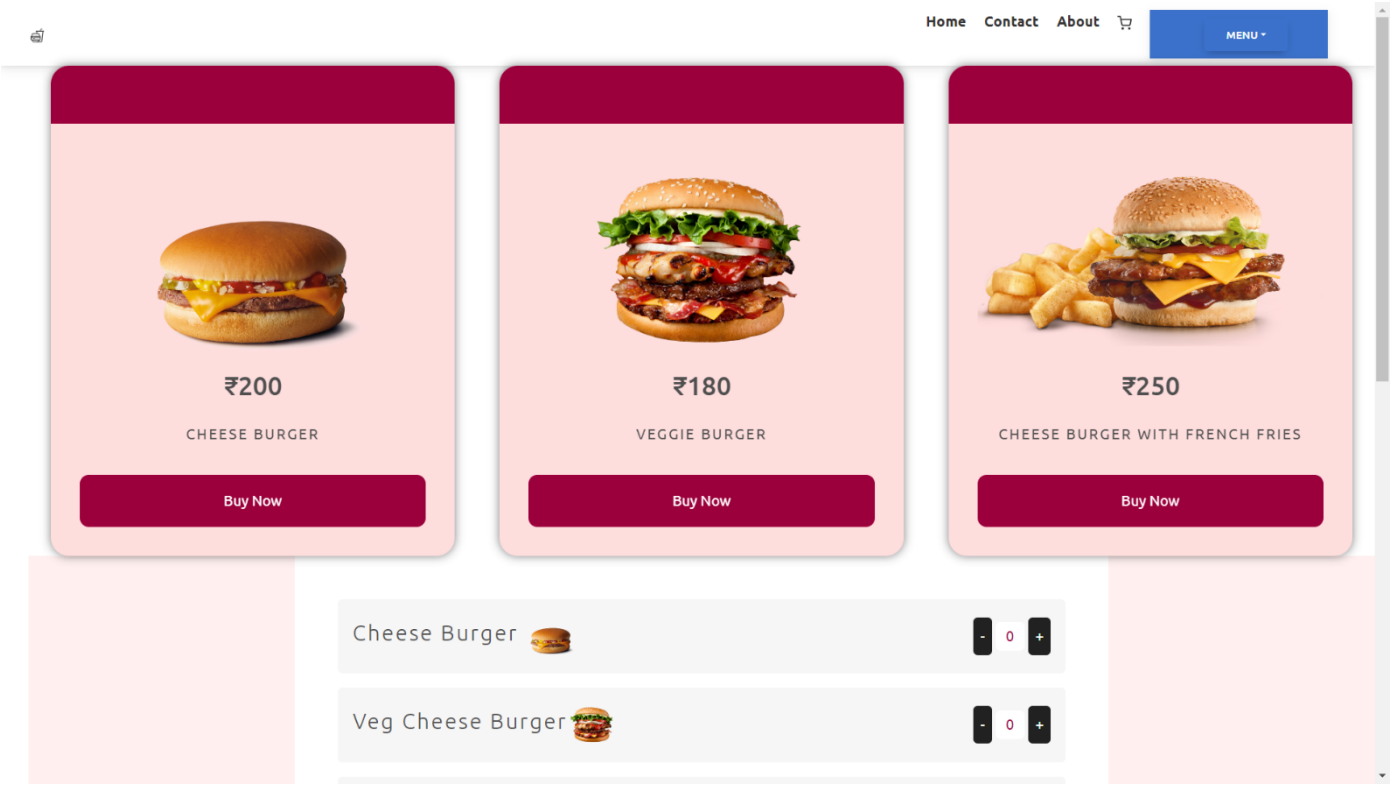
```jsx
import Founder from "./Founder";
import Menu from "./Menu";
import burgerImage from "../../assets/bg.jpg";
import { useNavigate } from "react-router-dom";

const options = {
  initial: {
    x: "-100%",
    opacity: 0,
  },
  whileInView: {
    x: 0,
    opacity: 1,
  },
};

function Home() {
  const [showImage, setShowImage] = useState(true);
  const navigate = useNavigate();

  const handleButtonClick = () => {
    // Navigate to the Menu component when the button is clicked
    navigate("/menu");
  };

  return (
    <div>
      {showImage ? (
        <motion.div {...options}>
          <img
            src={burgerImage}
            alt="Burger"
            style={{ width: "100%", height: "100vh", objectFit: "cover" }}
          />
          <button
            style={{
              position: "absolute",
              top: "50%",
              left: "50%",
              transform: "translate(-50%, -50%)",
              padding: "10px 20px",
              fontSize: "18px",
              cursor: "pointer",
            }}
            onClick={() => setShowImage(false)}
          >
            Explore Menu
          </button>
        </motion.div>
      ) : (
        <>
          <motion.div {...options}>
            {/* <Founder /> */}
```

```
          </motion.div>
          <motion.div {...options}>
            <Menu />
          </motion.div>
        </>
      )}
    </div>
  );
}

export default Home;
```

The outcome is:



And after clicking on the explore menu button this shows



Explaining the component;

1. The user will be pumping into a burger covered page and there is Explore menu button.
2. If the user clicks on Explore menu button the menu component will show.

1.2 /Task 2: Header on Home page

You will now be completing the Header of the Burger Shop project.

<instructions>

1. Go to the file - layout/Header.jsx in the “components” folder.
2. Create a navigation bar with links to Home, Contact, About, shopping cart, and a menu with links to Login, Orders and Logout.

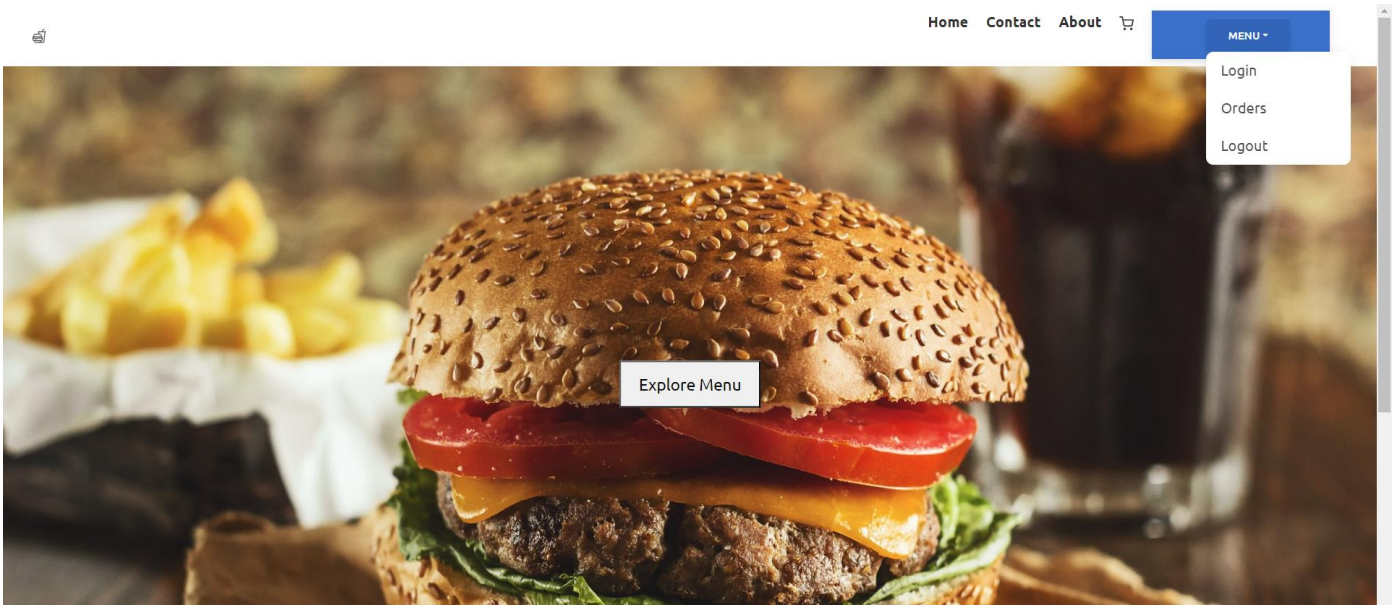
SOLUTION

The header component

```
import React from "react";
import { IoFastFoodOutline } from "react-icons/io5";
import { Link } from "react-router-dom";
import { FiShoppingCart } from "react-icons/fi";
import { motion } from "framer-motion";
import DropdownMenu from "./DropdownMenu";

function Header() {
  return (
    <nav className="navbar">
      <motion.div initial={{ x: "-100%" }} whileInView={{ x: 0 }}>
        <IoFastFoodOutline />
      </motion.div>
      <div className="menu-container">
```

```
        <ul className="nav-links">
          <li>
            <Link to="/">Home</Link>
          </li>
          <li>
            <Link to="/contact">Contact</Link>
          </li>
          <li>
            <Link to="/about">About</Link>
          </li>
          <li>
            <Link to="/cart">
              <FiShoppingCart />
            </Link>
          </li>
          <li>
            <DropdownMenu />
          </li>
        </ul>
      </div>
    </nav>
  );
}

export default Header;
```

And the outcome is:



The DropdownMenu component

```
import React, { useEffect, useState } from "react";
import { Link, useNavigate } from "react-router-dom";

function DropdownMenu() {
  const [navLinks, setNavLinks] = useState([]);
```

```
const navigate = useNavigate(); // Navigate to a different route

useEffect(() => {
  const navs = [
    { name: "Login", path: "/login" },
    { name: "Orders", path: "/myOrders" },
    { name: "Logout", path: "/login" },
  ];
  setNavLinks(navs);
}, []);

const handleLogout = () => {
  // Show confirmation dialog
  const confirmed = window.confirm("Are you sure you want to logout?");

  // If user confirms, redirect to login
  if (confirmed) {
    alert('You have been logged out successfully!!');
    navigate("/login");
  }
};

return (
  <div>
    <nav className="navbar navbar-expand-lg navbar-dark bg-primary">
      <div className="container">
        <a className="navbar-brand" href="#"></a>
        <div className="btn-group">
          <button
            type="button"
            className="btn btn-primary dropdown-toggle"
            data-bs-toggle="dropdown"
            data-bs-display="static"
            aria-expanded="false"
          >
            Menu
          </button>
          <ul className="dropdown-menu dropdown-menu-end">
            {navLinks.map((d, i) => (
              <li key={i}>
                {d.name === "Logout" ? (
                  <button
                    className="dropdown-item"
                    type="button"
                    onClick={handleLogout}
                  >
                    {d.name}
                  </button>
                ) : (
                  <Link to={d.path}>
                    <button className="dropdown-item" type="button">
                      {d.name}
                    </button>
```

```
                    </Link>
                  )}
                </li>
              ))}
            </ul>
          </div>
        </div>
      </nav>
    </div>
  );
}

export default DropdownMenu;
```

Noting that all the routes work fine and there are no error throws in the console or whatsoever.

1.3 /Task 3: Footer on Home page

You will now be completing the Footer of the Burger Shop project.

Instructions:

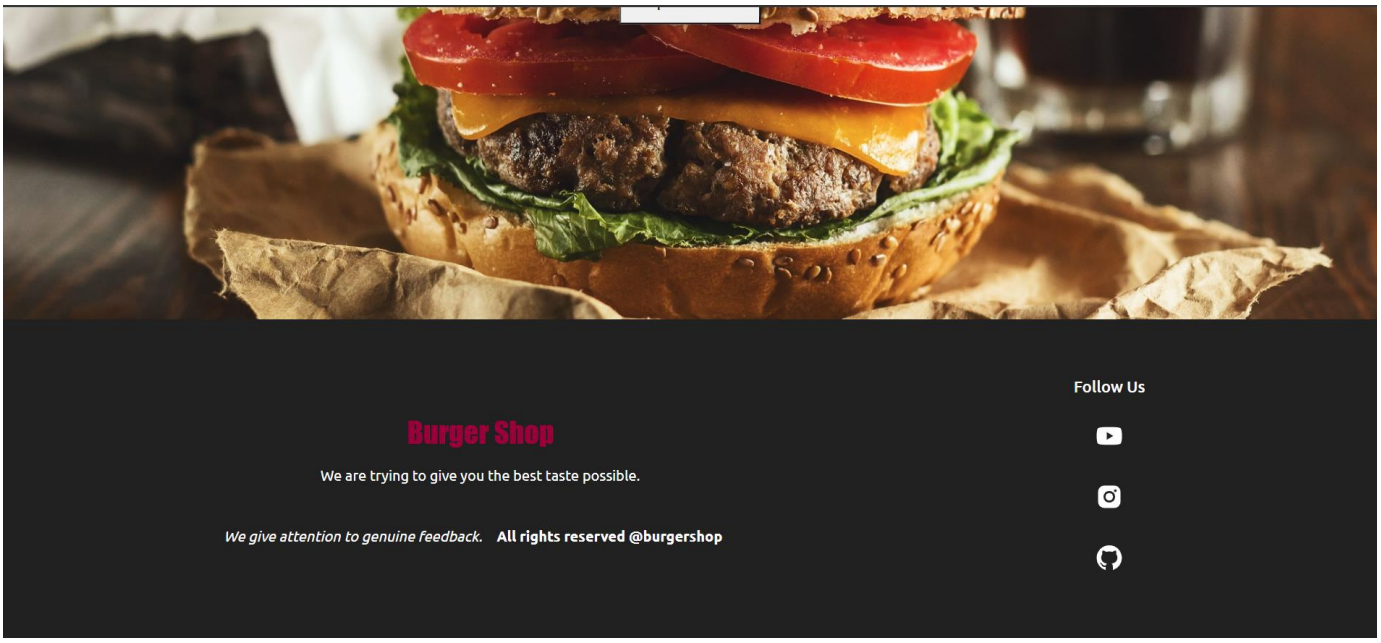
1. Go to the file - layout/Footer.jsx in the “components” folder.
2. Create a social media bar with links to YouTube and Instagram, using anchor tags and icons from the react-icons/ai library.

SOLUTION

```
import React from "react";
import { AiFillInstagram, AiFillYoutube, AiFillGithub } from "react-icons/ai";

function Footer() {
  return (
    <footer>
      <div>
        <h2>Burger Shop</h2>
        <p>We are trying to give you the best taste possible.</p>
        <br />
        <em>We give attention to genuine feedback.</em>
        <strong>All rights reserved @burgershop</strong>
      </div>
      <aside>
        <h4>Follow Us</h4>
        <a href="https://www.youtube.com/@itiscoded/videos">
          <AiFillYoutube />
        </a>
        <a href="https://www.instagram.com/itiscoded/">
          <AiFillInstagram />
        </a>
        <a href="https://github.com/EskandarAtrakchi/Skillup_Burger_Shop">
        <AiFillGithub />
        </a>
      </aside>
    </footer>
  );
```

```
}

export default Footer;
```

And the outcome is:



All the icons are hrefed to my social media and the GitHub will land you to the repo of this project in my GitHub

One of the addition I added is GitHub and I linked it to this project if it clicked on.

1.4 / Task 4: Explore Menu:

You will now be completing the menu of some items available in the Burger Shop using the MenuCard Component

Instructions:

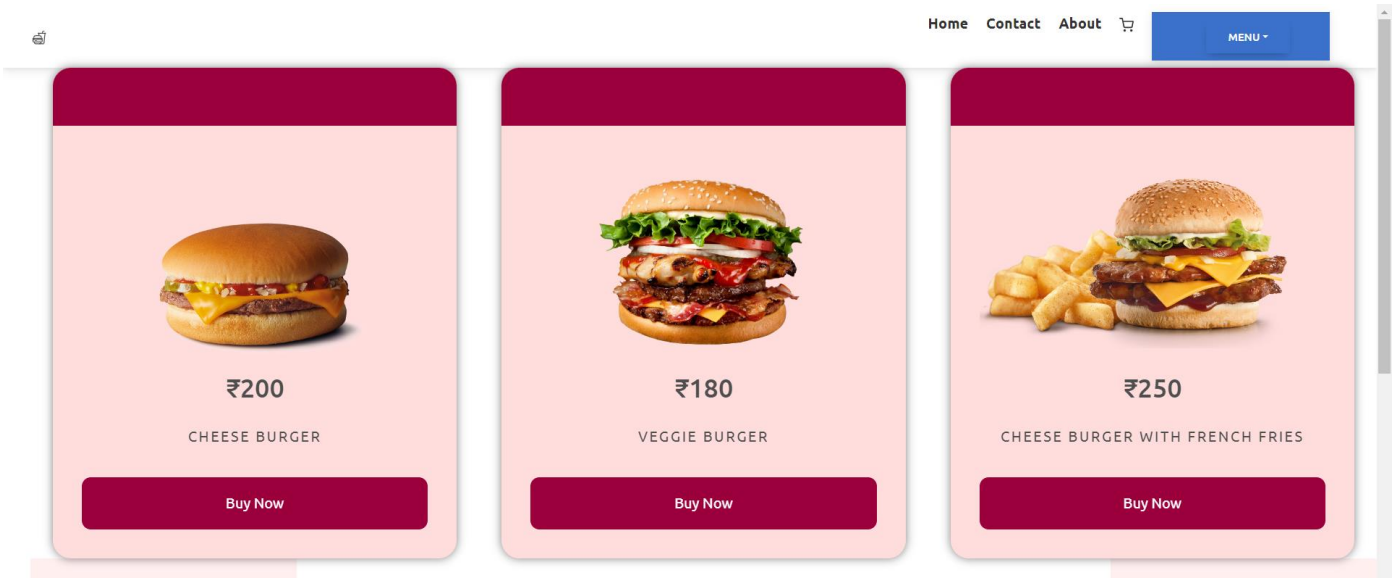
1. Go to the file - home/Menu.jsx in the “components” folder.
2. Create a menu card of the items with the item number, image source, price, title, a function handler for adding to cart, and a delay for animation.

SOLUTION

```
import React, { useState } from "react";
import MenuCard from "./MenuCard";
import burger1 from "../../assets/burger1.png";
import burger2 from "../../assets/burger2.png";
import burger3 from "../../assets/burger3.png";
import Cart from "../cart/Cart";
import { Navigate } from "react-router-dom";

const Menu = () => {
  const [cartItems, setCartItems] = useState({
    1: 0,
    2: 0,
    3: 0,
```

```
  });

  const Handler = (itemId) => {
    setCartItems((prevItems) => ({
      ...prevItems,
      [itemId]: prevItems[itemId] + 1,
    }));
    console.log(`Item ${itemId} added to cart!`);

  };

  return (
    <div className="wappingDiv">
        <div className="menu">
          <MenuCard
            itemNum={1}
            burgerSrc={burger1}
            price={200}
            title="Cheese Burger"
            handler={() => Handler(1)}
            delay={0.1}
          />
          <MenuCard
            itemNum={2}
            burgerSrc={burger2}
            price={180}
            title="Veggie Burger"
            handler={() => Handler(2)}
            delay={0.2}
          />
          <MenuCard
            itemNum={3}
            burgerSrc={burger3}
            price={250}
            title="Cheese Burger with French Fries"
            handler={() => Handler(3)}
            delay={0.3}
          />
          <Cart cartItems={cartItems} />
        </div>
    </div>
  );
};

export default Menu;
```

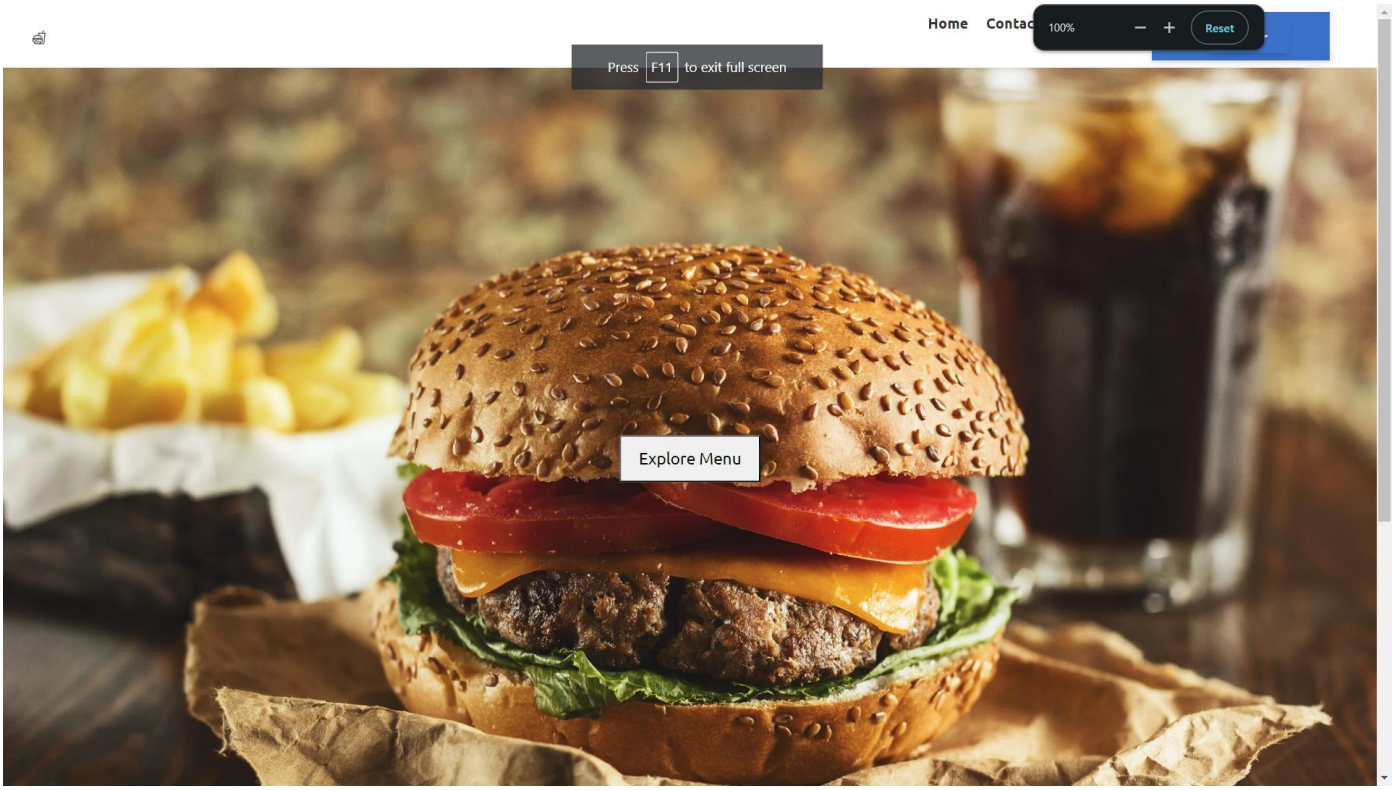
And the outcome is:



The cards styling SCSS has been changed to display the cards in flex mode and this style will be broken to column display when the screen width is less than 600 PX as this code shows it

```
@media screen and (max-width: 600px) {
  #menu > div {
    flex-direction: column;
  }
```

Noting that this shows after the user clicks on the explore menu button.



The Explore menu button is in the middle of the page.

Exercise 2:

2.1/ Task 1: About Page

You will now be completing the About Page of the Burger Shop project.

Instructions:

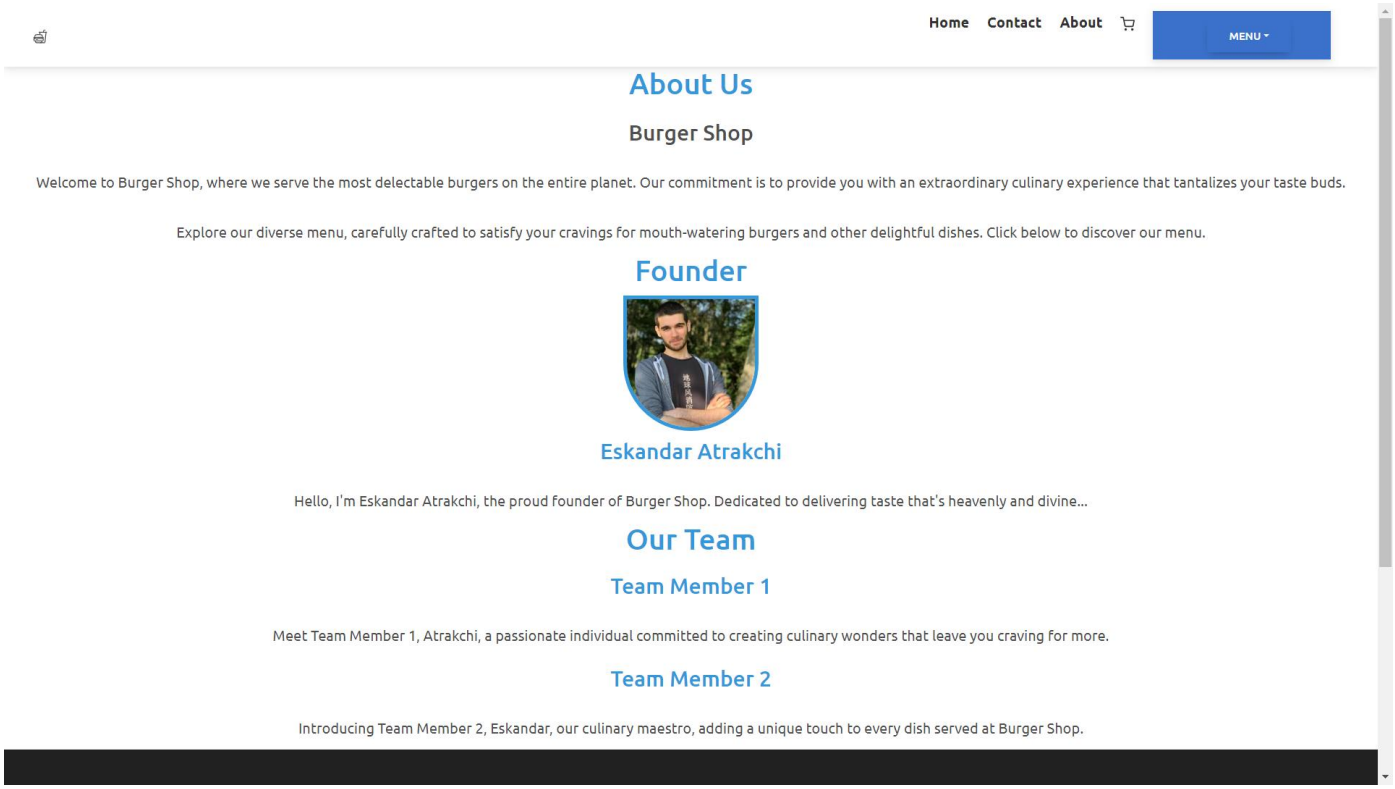
1. Go to the file - about/About.jsx in the “components” folder.
2. Import all the necessary packages and the picture of the founder

SOLUTION

```
import React from "react";
import { RiFindReplaceLine } from "react-icons/ri";
import me from "../../assets/skj.jpg";

const About = () => {
  return (
    <div>
      <h1>About Us</h1>
      <article>
        <h4>Burger Shop</h4>
        <p>
          Welcome to Burger Shop, where we serve the most delectable burgers on
          the entire planet. Our commitment is to provide you with an
          extraordinary culinary experience that tantalizes your taste buds.
        </p>
        <p>
          Explore our diverse menu, carefully crafted to satisfy your cravings
          for mouth-watering burgers and other delightful dishes. Click below
          to discover our menu.
        </p>
      </article>

      <div>
        <h2>Founder</h2>
        <article>
          <div>
            <img src={me} alt="Founder" />
            <h3>Eskandar Atrakchi</h3>
          </div>
          <p>
            Hello, I'm Eskandar Atrakchi, the proud founder of Burger Shop.
            Dedicated to delivering taste that's heavenly and divine...
          </p>
        </article>
      </div>

      <div>
        <h2>Our Team</h2>
        <article>
          <div>
            <h3>Team Member 1</h3>
          </div>
          <p>
```

```
            Meet Team Member 1, Atrakchi, a passionate individual committed to creating
            culinary wonders that leave you craving for more.
          </p>
        </article>

        <article>
          <div>
            <h3>Team Member 2</h3>
          </div>
          <p>
            Introducing Team Member 2, Eskandar, our culinary maestro, adding a unique
            touch to every dish served at Burger Shop.
          </p>
        </article>
      </div>
    </div>
  );
};

export default About;
```

And the outcome is:



According to the brief exercises there is not much required from me to do only to create the about page with some content and image of mine.

2.2 /Task 2: Contact Page

You will now be completing the Contact Page of the Burger Shop project.

Instructions:

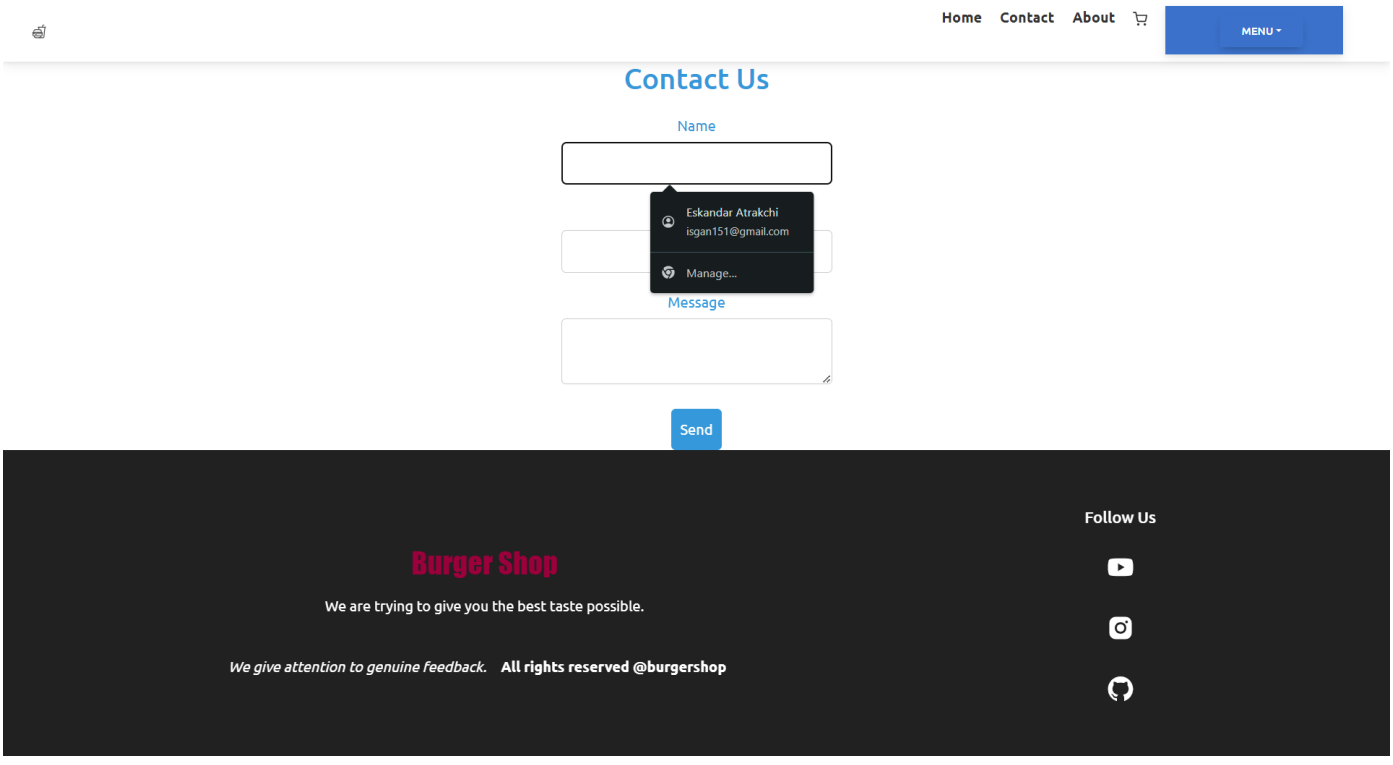
1. Go to the file - contact/Contact.jsx in the “components” folder.
2. Create a Popup with the click of the send button

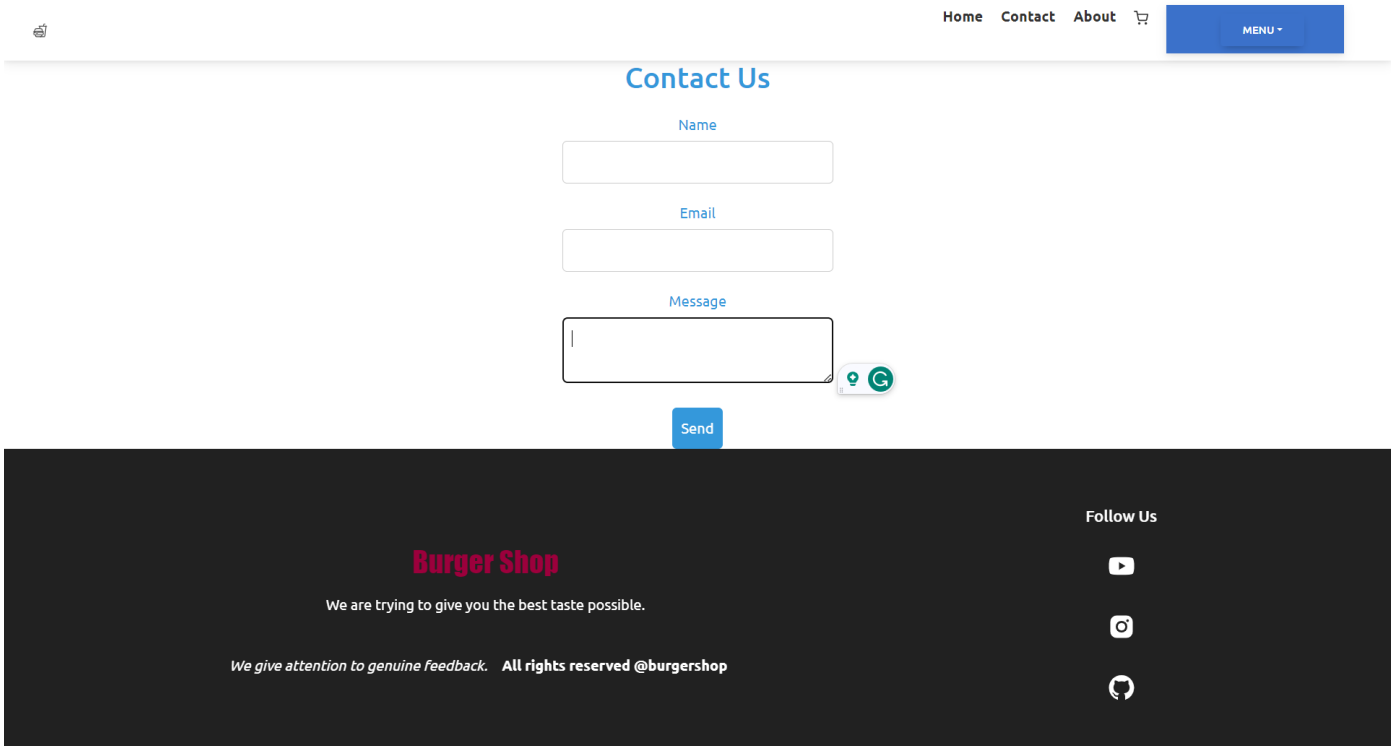
SOLUTION

```
import React, { useState } from 'react';
import Popup from 'reactjs-popup';
import { useNavigate } from 'react-router-dom';

const Contact = () => {
  const [formData, setFormData] = useState({
    name: '',
    email: '',
    message: '',
  });

  const [isSubmitted, setIsSubmitted] = useState(false);
  const navigate = useNavigate();

  const handleChange = (e) => {
    setFormData({
      ...formData,
      [e.target.name]: e.target.value,
    });
  };

  const handleSubmit = (e) => {
    e.preventDefault();

    // Check if the fields are empty
    if (!formData.name || !formData.email || !formData.message) {
      alert('Please fill in all the fields before submitting.');
      return;
    }

    // Handle form submission logic here
    // in the lecture course the guy said no database required for now (e.g., send data to server) so I will just log the data to the console
    //the video is here https://ibm.box.com/s/nlo8r939muitapht5ontxjgdms5bp8yb
    console.log('Form submitted:', formData);

    // Alert user about successful submission
    alert('Form submitted successfully!');

    // Redirect to the home page
    setIsSubmitted(true);
    navigate('/');
  };

  return (
```

```
    <div>
      <h1>Contact Us</h1>
      <form onSubmit={handleSubmit}>
        <div>
          <label>Name</label>
          <input type="text" name="name" value={formData.name} onChange={handleChange} />
        </div>
        <div>
          <label>Email</label>
          <input type="email" name="email" value={formData.email} onChange={handleChange}
/>
        </div>
        <div>
          <label>Message</label>
          <textarea name="message" value={formData.message} onChange={handleChange} />
        </div>
        {/* Popup for confirmation or additional information */}
        <Popup trigger={<button type="submit">Send</button>}>
          <div>
            <h2>Your message is important to us!</h2>
            <p>Try to fill out the fields before submission!</p>
          </div>
        </Popup>
      </form>
    </div>
  );
};

export default Contact;
```

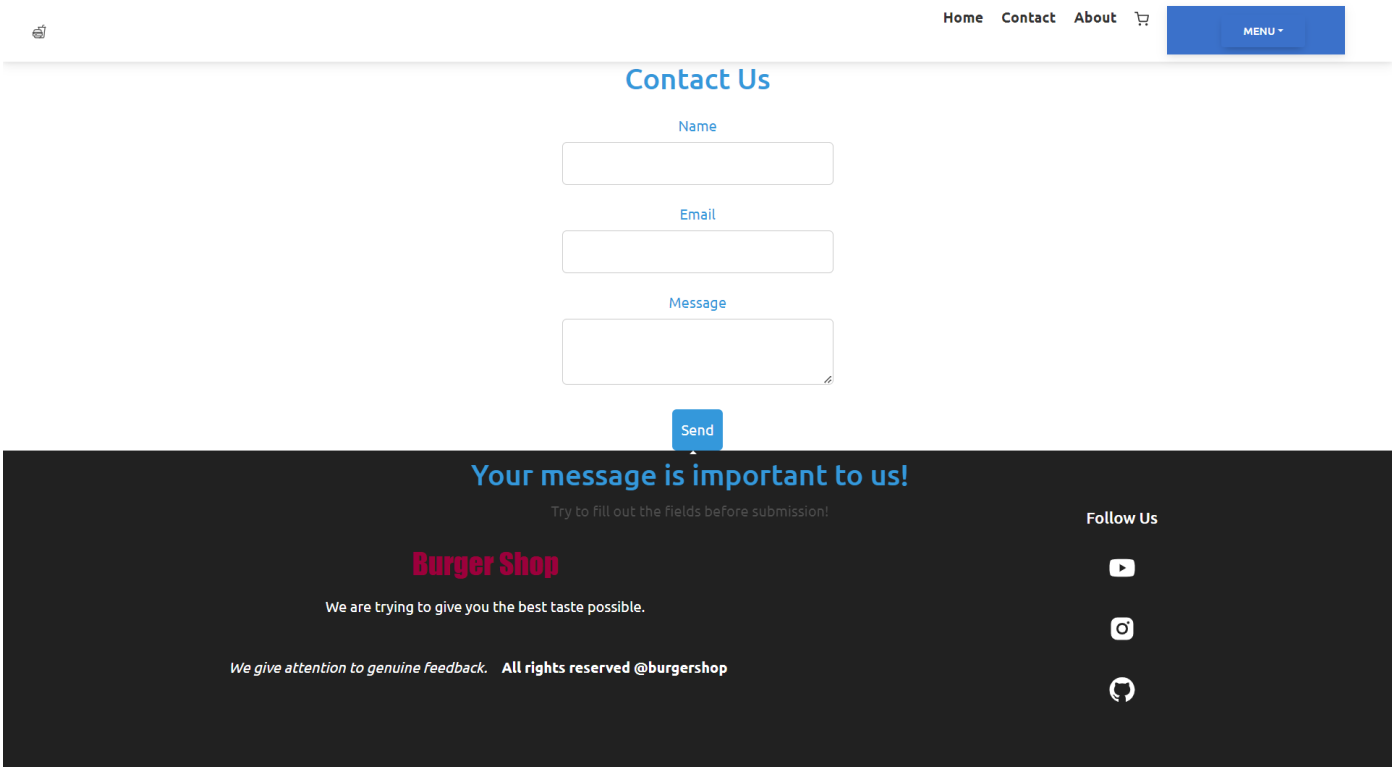
And the outcome is:





The logic is;

1. If the fields are empty the user won't be able to submit their message
2. If the user fails to submit the user will be notified that their message is not submitted
3. If the user is successfully submitted their message, then the user will be notified.
4. If the user successfully submitted their message the user will be redirected to the home component (home page).
5. This pop up will show to the user if the user tries to submit with the fields are empty or invalid data.



2.3 /Task 3: Login & Signup Page

You will now be completing the Login Signup Page of the Burger Shop project.

Instructions:

1. Go to the file - login/Login.jsx in the “components” folder.
2. Create a Login and Signup Tabs for the Login and Signup Form

SOLUTION

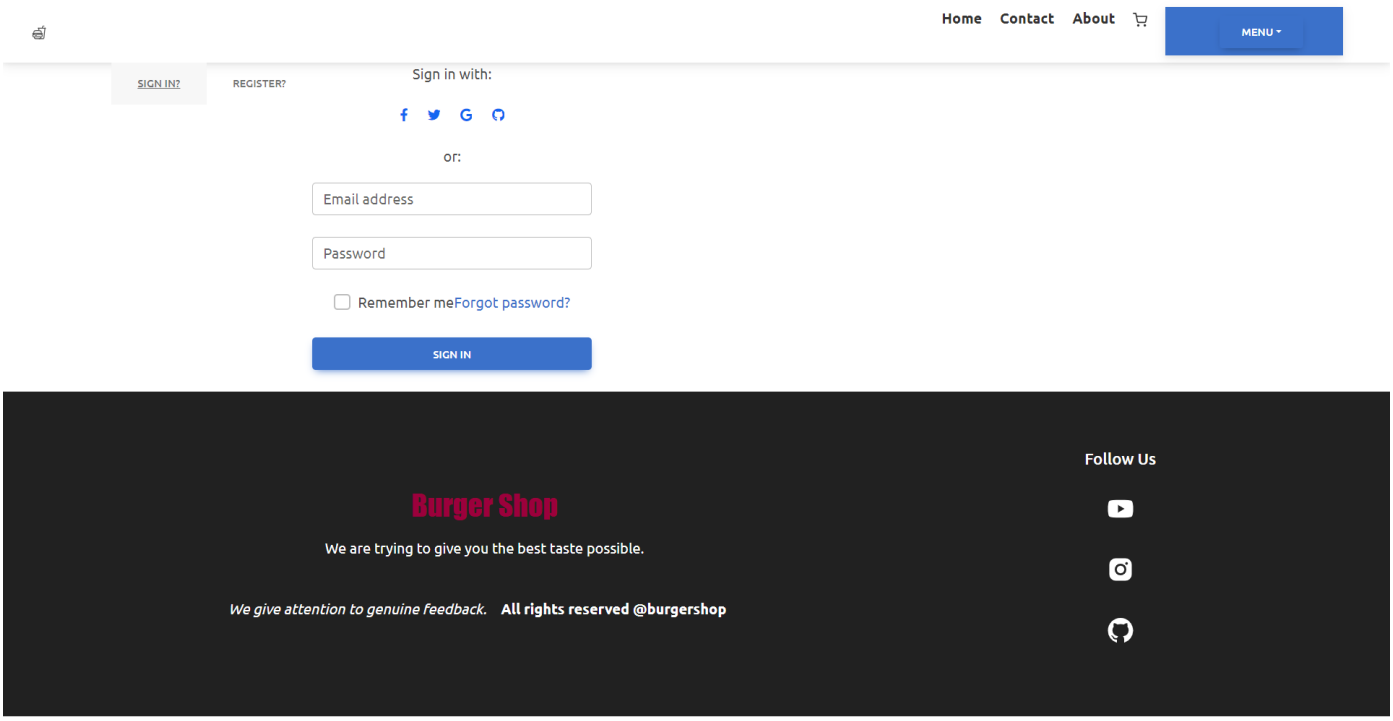
```
import React, { useState } from "react";
import { useNavigate } from "react-router-dom";
import {
  MDBContainer,
  MDBTabs,
  MDBTabsItem,
  MDBTabsLink,
  MDBTabsContent,
  MDBTabsPane,
  MDBBtn,
  MDBIcon,
  MDBInput,
  MDBCheckbox,
} from "mdb-react-ui-kit";

const Login = () => {
  const navigate = useNavigate();
  const [justifyActive, setJustifyActive] = useState("tab1");
  const [formData, setFormData] = useState({
    email: "",
    password: "",
  });

  const handleSignIn = () => {
    // Validate fields
    if (!formData.email || !formData.password) {
      alert("Please fill in all the fields before signing in.");
      return;
    }

    //Assume the user is logged in successfully
    alert("Logged in successfully!");

    // Redirect to the home page after successful login
    navigate("/");
  };

  const handleRegister = () => {
    // Validate fields
    if (!formData.email || !formData.password) {
      alert("Please fill in all the fields before registering.");
```

```
      return;
    }

    // Assume the user is registered successfully
    alert("Registered successfully!");

    // Redirect to the home page after successful registration
    navigate("/");
  };

  const handleChange = (e) => {
    setFormData({
      ...formData,
      [e.target.name]: e.target.value,
    });
  };

  return (
    <MDBContainer>
      <MDBTabs>
        <MDBTabsItem>
          <MDBTabsLink show={justifyActive === "tab1"} onClick={() =>
setJustifyActive("tab1")}>
            Sign In?
          </MDBTabsLink>
        </MDBTabsItem>

        {/*Creating another tab which will be number 2 for registering note delete later,
come back here if the registeration does not work*/}
        {/*Creating another tab for registeration is required in the brief*/}
        <MDBTabsItem>
          <MDBTabsLink show={justifyActive === "tab2"} onClick={() =>
setJustifyActive("tab2")}>
            Register?
          </MDBTabsLink>
        </MDBTabsItem>

        <MDBTabsContent>
          <MDBTabsPane show={justifyActive === "tab1"}>
            <div className="text-center mb-3">
              <p>Sign in with:</p>
              <div className="d-flex justify-content-between mx-auto" style={{ width:
"40%" }}>
                <MDBBtn tag="a" color="none" className="m-1" style={{ color: "#1266f1" }}>
                  <MDBIcon fab icon="facebook-f" size="sm" />
                </MDBBtn>
                <MDBBtn tag="a" color="none" className="m-1" style={{ color: "#1266f1" }}>
                  <MDBIcon fab icon="twitter" size="sm" />
                </MDBBtn>
                <MDBBtn tag="a" color="none" className="m-1" style={{ color: "#1266f1" }}>
                  <MDBIcon fab icon="google" size="sm" />
                </MDBBtn>
                <MDBBtn tag="a" color="none" className="m-1" style={{ color: "#1266f1" }}>
```

```
                <MDBIcon fab icon="github" size="sm" />
              </MDBBtn>
            </div>
            <p className="text-center mt-3">or:</p>
          </div>
          <MDBInput wrapperClass="mb-4" label="Email address" id="form1" type="email"
name="email" value={formData.email} onChange={handleChange} />
          <MDBInput wrapperClass="mb-4" label="Password" id="form2" type="password"
name="password" value={formData.password} onChange={handleChange} />
          <div className="d-flex justify-content-between mx-4 mb-4">
            <MDBCheckbox name="flexCheck" value="" id="flexCheckDefault" label="Remember
me" />
            <a href="!#">Forgot password?</a>
          </div>
          <MDBBtn className="mb-4 w-100" onClick={handleSignIn}>
            Sign in
          </MDBBtn>
        </MDBTabsPane>

        <MDBTabsPane show={justifyActive === "tab2"}>
          <div className="text-center mb-3">
            <p>Register with:</p>
            <div className="d-flex justify-content-between mx-auto" style={{ width:
"40%" }}>
              <MDBBtn tag="a" color="none" className="m-1" style={{ color: "#1266f1" }}>
                <MDBIcon fab icon="facebook-f" size="sm" />
              </MDBBtn>
              <MDBBtn tag="a" color="none" className="m-1" style={{ color: "#1266f1" }}>
                <MDBIcon fab icon="twitter" size="sm" />
              </MDBBtn>
              <MDBBtn tag="a" color="none" className="m-1" style={{ color: "#1266f1" }}>
                <MDBIcon fab icon="google" size="sm" />
              </MDBBtn>
              <MDBBtn tag="a" color="none" className="m-1" style={{ color: "#1266f1" }}>
                <MDBIcon fab icon="github" size="sm" />
              </MDBBtn>
            </div>
            <p className="text-center mt-3">or:</p>
          </div>
          <MDBInput wrapperClass="mb-4" label="Email address" id="form1" type="email"
name="email" value={formData.email} onChange={handleChange} />
          <MDBInput wrapperClass="mb-4" label="Password" id="form2" type="password"
name="password" value={formData.password} onChange={handleChange} />
          <div className="d-flex justify-content-between mx-4 mb-4">
            <MDBCheckbox name="flexCheck" value="" id="flexCheckDefault" label="Remember
me" />
          </div>
          <MDBBtn className="mb-4 w-100" onClick={handleRegister}>
            Register
          </MDBBtn>
        </MDBTabsPane>
      </MDBTabsContent>
    </MDBTabs>
```

```
    </MDBContainer>
  );
};

export default Login;
```

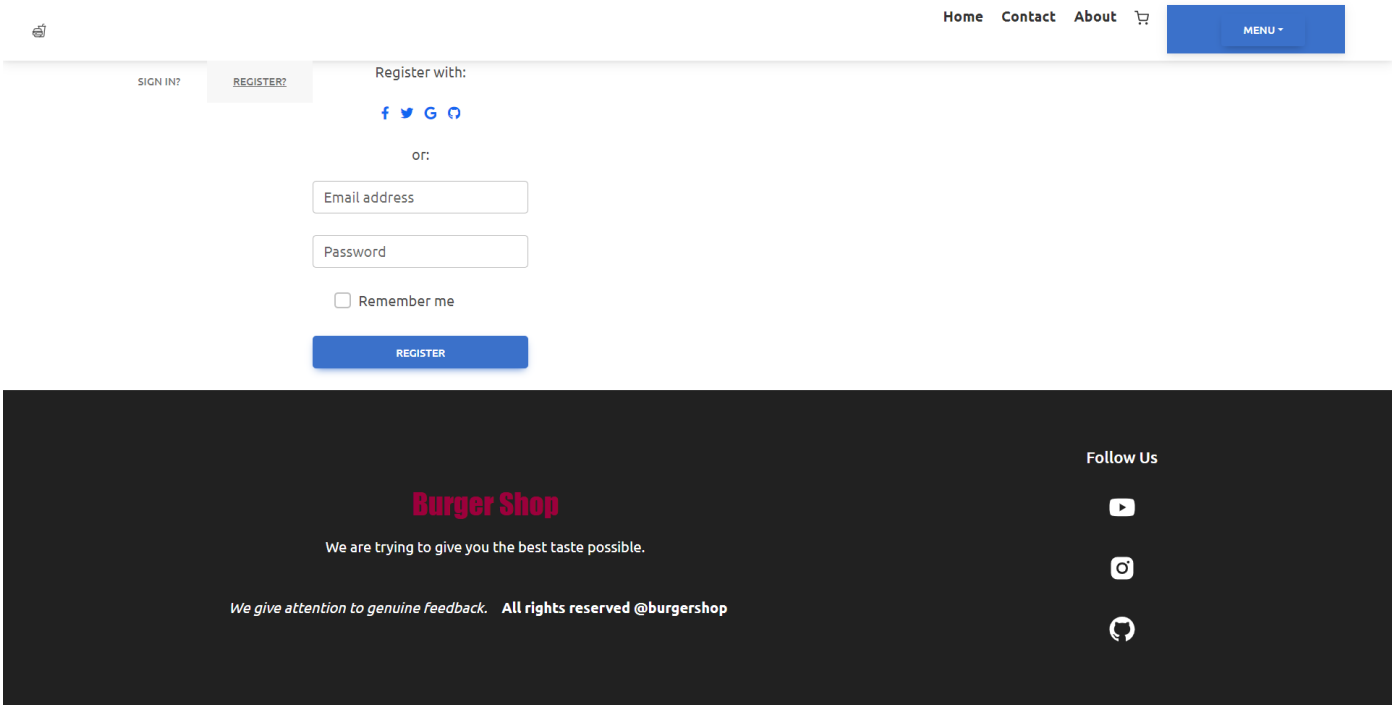
And the outcome is;
For the sign-in tab



And the logic is;

1. **The user can not sign in if the fields are empty.**
2. **The user will be notified if the fields are empty.**
3. **The user will be notified if the sign-in is successful.**
4. **The user will be redirected to the home component (home page) if the sign-in is successful.**

For the Registration tab



And the logic is;

1. **The user can not register if the fields are empty.**
2. **The user will be notified if the fields are empty.**
3. **The user will be notified if the registration is successful.**
4. **The user will be redirected to the home component (home page) if the registration is successful.**

Exercise 3

3.1/ Task 1: Create the “Cart” page

You will now be completing the menu on you Cart for selecting and placing the orders.

2 items namely, Cheeseburger and Veg Cheeseburger are already added. You will add another item - ”Cheeseburger with French Fries

Instructions:

1. Go to the file - cart/Cart.jsx in the “components” folder and add a “CartItem” React element.
2. Set its attributes as:

- title: ”Cheeseburger with French Fries”
- Img: burger3 ie. You need to import the image (“burger3”) from the assets folder to use this as the variable - Set the initial value as 0
- Add increment and decrement counters

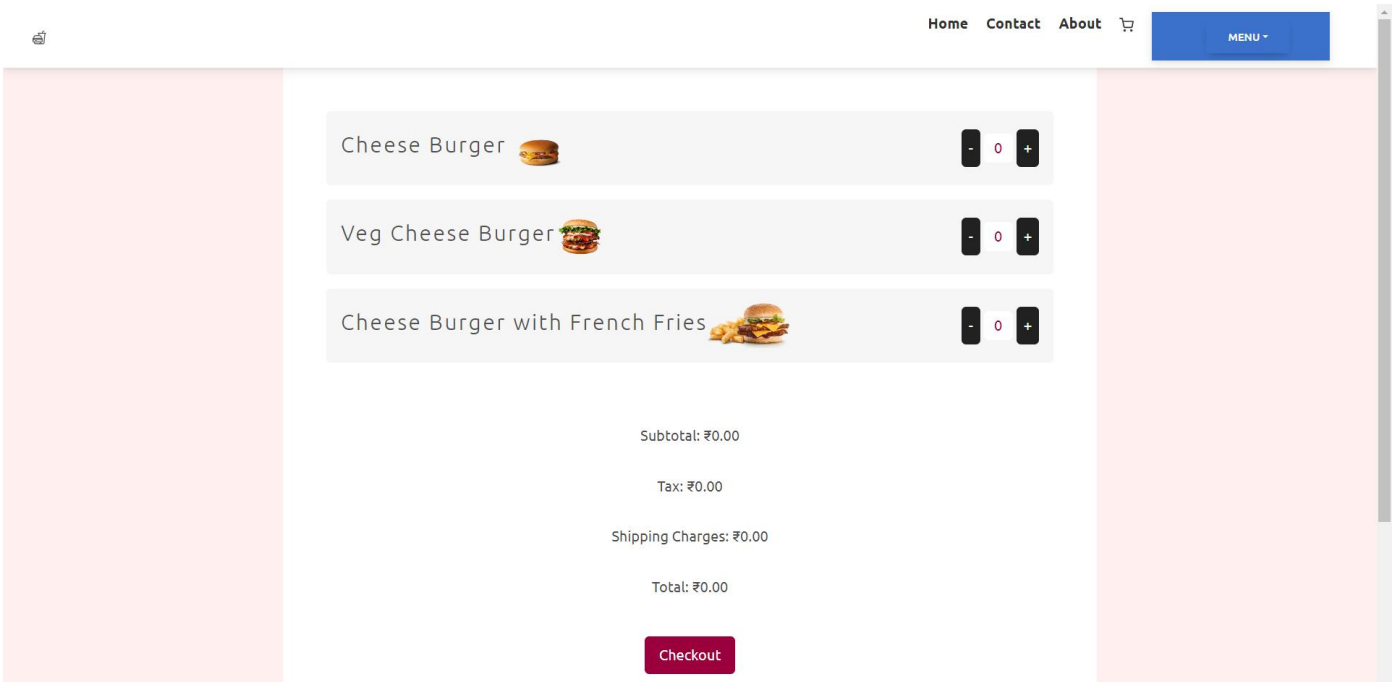
SOLUTION

```
import React, { useState } from "react";
import { Link } from "react-router-dom";
import burger1 from "../../assets/burger1.png";
import burger2 from "../../assets/burger2.png";
import burger3 from "../../assets/burger3.png";

const CartItem = ({ value, title, img, increment, decrement }) => (
```

```
  <div className="cartItem">
    <div>
      <h4>{title}</h4>
      <img src={img} alt="Item" />
    </div>

    <div>
      <button onClick={decrement}>-</button>
      <input type="number" readOnly value={value} />
      <button onClick={increment}>+</button>
    </div>
  </div>
);

const Cart = () => {
  const [cartItems, setCartItems] = useState({
    1: 0, // Quantity for Cheese Burger
    2: 0, // Quantity for Veg Cheese Burger
    3: 0, // Quantity for Cheese Burger with French Fries
  });

  const itemPrices = {
    1: 200, // Price for Cheese Burger
    2: 180, // Price for Veg Cheese Burger
    3: 250, // Price for Cheese Burger with French Fries
  };

  const taxRate = 0.1; // 10%

  const shippingCharges = 0;//I will set this to zero for now, change it to 50 later,
delete this later

  const subtotal = Object.keys(cartItems).reduce(
    (acc, itemId) => acc + cartItems[itemId] * itemPrices[itemId],
    0
  );

  const tax = subtotal * taxRate;
  const total = subtotal + tax + shippingCharges;

  const increment = (itemId) => {
    setCartItems((prevItems) => ({
      ...prevItems,
      [itemId]: prevItems[itemId] + 1,
    }));
  };

  const decrement = (itemId) => {
    if (cartItems[itemId] > 0) {
      setCartItems((prevItems) => ({
        ...prevItems,
        [itemId]: prevItems[itemId] - 1,
      }));
```

```
    }
  };

  return (
    <section className="cart">
      <main>
        <CartItem
          title={"Cheese Burger"}
          img={burger1}
          value={cartItems[1]}
          increment={() => increment(1)}
          decrement={() => decrement(1)}
        />
        <CartItem
          title={"Veg Cheese Burger"}
          img={burger2}
          value={cartItems[2]}
          increment={() => increment(2)}
          decrement={() => decrement(2)}
        />
        <CartItem
          title={"Cheese Burger with French Fries"}
          img={burger3}
          value={cartItems[3]}
          increment={() => increment(3)}
          decrement={() => decrement(3)}
        />
        <article>
          <p>Subtotal: ₹{subtotal.toFixed(2)}</p>
          <p>Tax: ₹{tax.toFixed(2)}</p>
          <p>Shipping Charges: ₹{shippingCharges.toFixed(2)}</p>
          <p>Total: ₹{total.toFixed(2)}</p>
          <Link to="/shipping">Checkout</Link>
        </article>
      </main>
    </section>
  );
};

export default Cart;
```

And the outcome is;



The logic is;

1. **The user can hit on + and increment burgers as required.**
2. **If the – gets clicked the decrement from the last potential number will be implemented.**
3. **The subtotal, tax, shipping charges, and total will be counted accordingly as the user increments and decrements.**

3.2 /Task 2: Shipping page

You will now be completing the order details for confirming your order and shipping The code is partly complete, and you will be completing the rest.

Instructions:

1. Go to the cart/Shipping.jsx file in the “components” folder.
2. Navigate to the element marked COUNTRY DROPDOWN
3. The following code has been given:

- Label element with name “Country”
- Select element for the dropdown
- Options element for the dropdown option (ie. Country names)

4. Write the remaining code as below:

- Use the getAllCountries method to get an array having the details of all countries
- Use the map() method to iterate through this array to output the ISO code for each country as the “value” attribute for the option tag and display the country Name in the dropdown menu.

5. Similarly create a dropdown for selecting the states of a country.

It will be similar to the above with the difference being that:

- The State object of the “country-state-city" library will be used
- The getStatesOfCountry() method will be used and its parameter will be “IN” (for selecting India as the country)

6. An input element for entering Pin code is present

7. Create an input element for Phone number

8. The final code is a <Popup> element for generating a pop-up message “Order Placed” on clicking the “Confirm Order” button. Observe the CSS elements used for styling it.

9. Import the Shipping object (exported from Shipping.jsx) in App.js and create the “Shipping” Route in the Return method of the App() element

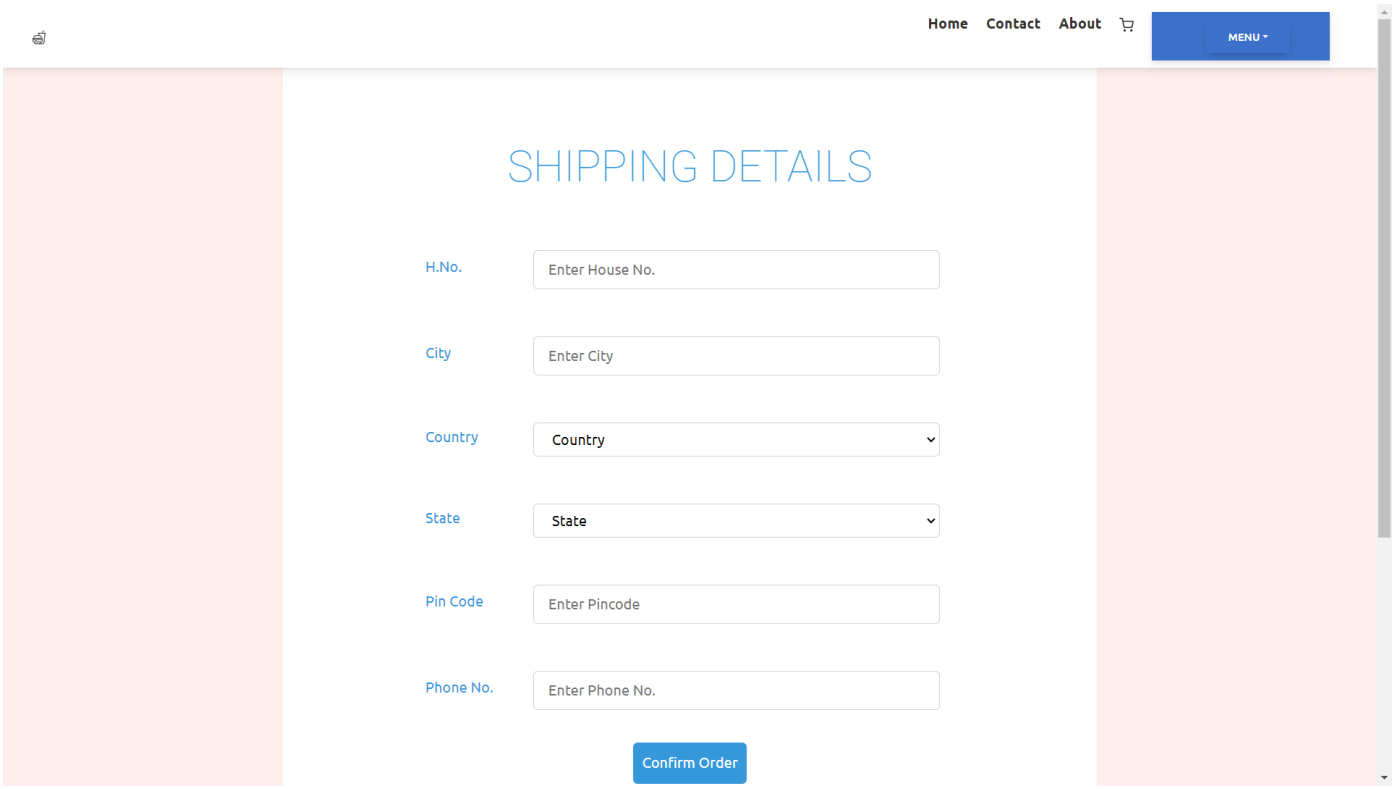
SOLUTION

```
import React, { useState } from "react";
import { Country, State } from "country-state-city";
import Popup from "reactjs-popup";
import { useNavigate } from "react-router-dom";

const Shipping = () => {
  const [selectedCountry, setSelectedCountry] = useState("");
  const [selectedState, setSelectedState] = useState("");
  const [orderPlaced, setOrderPlaced] = useState(false);
  const [formError, setFormError] = useState(false);

  const navigate = useNavigate();

  const handleCountryChange = (e) => {
    const countryIsoCode = e.target.value;
    setSelectedCountry(countryIsoCode);
    setSelectedState(""); // Reset selected state when country changes
  };

  const handleConfirmOrder = () => {
    // Check if any required field is empty
    if (!selectedCountry || !selectedState) {
      setFormError(true);
      return;
    }

    // Mock logic for placing an order
    setOrderPlaced(true);

    // Show the success popup
    setTimeout(() => {
      setOrderPlaced(false);
      setFormError(false); // Reset form error state
      // Alert the user that the order has been placed successfully
      alert("Order placed successfully!");
      // Redirect to the home page
      navigate("/");
    }, 500);
  };

  return (
    <section className="shipping">
```

```
      <main>
        <h1>Shipping Details</h1>
        <form>
          <div>
            <label>H.No.</label>
            <input type="text" placeholder="Enter House No." />
          </div>
          <div>
            <label>City</label>
            <input type="text" placeholder="Enter City" />
          </div>
          <div>
            {/* COUNTRY DROPDOWN */}
            <label>Country</label>
            <select onChange={handleCountryChange} value={selectedCountry}>
              <option value="">Country</option>
              {Country.getAllCountries().map((country) => (
                <option value={country.isoCode} key={country.isoCode}>
                  {country.name}
                </option>
              ))}
            </select>
          </div>
          <div>
            {/* STATE DROPDOWN */}
            <label>State</label>
            <select value={selectedState} onChange={(e) =>
setSelectedState(e.target.value)}>
              <option value="">State</option>
              {selectedCountry &&
                State.getStatesOfCountry(selectedCountry).map((state) => (
                  <option value={state.isoCode} key={state.isoCode}>
                    {state.name}
                  </option>
                ))}
            </select>
          </div>
          <div>
            {/* PIN CODE INPUT */}
            <label>Pin Code</label>
            <input type="number" placeholder="Enter Pincode" />
          </div>
          <div>
            {/* PHONE NUMBER INPUT */}
            <label>Phone No.</label>
            <input type="number" placeholder="Enter Phone No." />
          </div>
          {/* Add the code for contact */}
          <button type="button" onClick={handleConfirmOrder}>
            Confirm Order
          </button>
          {formError && <div style={{ color: "red" }}>Please fill in all required
fields</div>}
```

```
          <Popup open={orderPlaced} position="right center">
            <div style={{ color: "red", position: "absolute", top: "50%", right: "100%",
transform: "translateY(-50%)", backgroundColor: "#fff", padding: "10px", borderRadius:
"5px", boxShadow: "0 0 10px rgba(0, 0, 0, 0.2)" }}>
              Order Placed successfully!
            </div>
          </Popup>
        </form>
      </main>
    </section>
  );
};

export default Shipping;
```

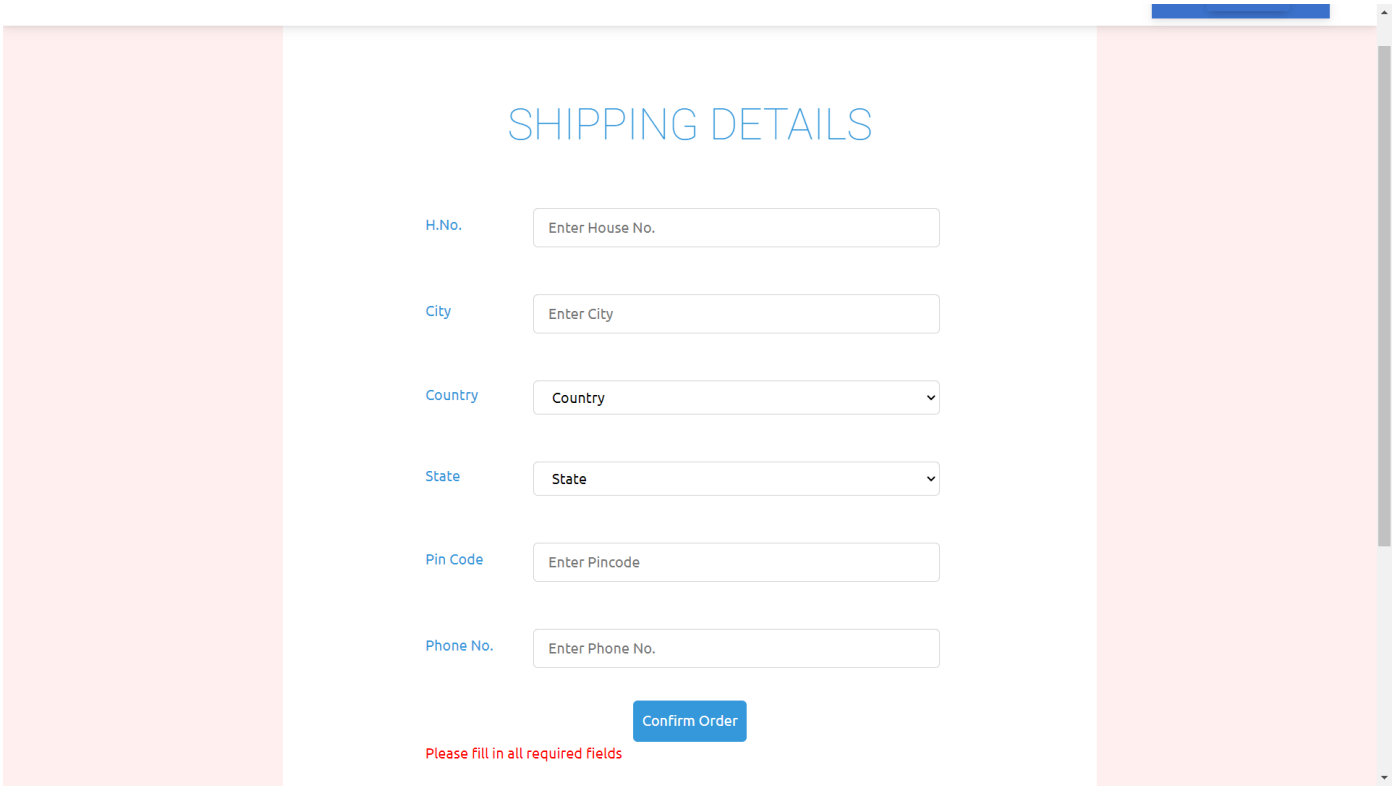
And the outcome is;



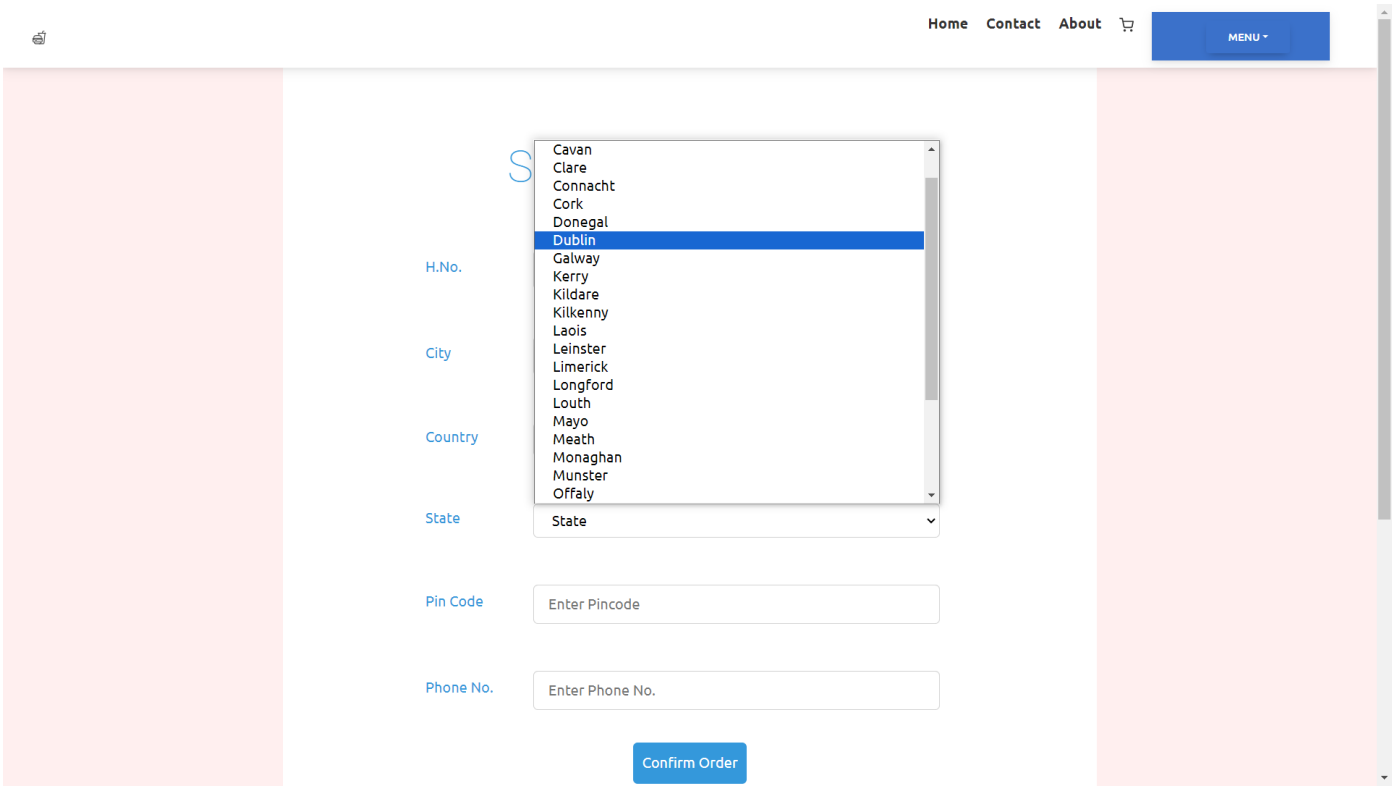
The logic is;

1. The user cannot confirm their order if the fields are empty.
2. The user will be notified if the fields are empty.
3. The state will be empty if the user does not choose any country.
4. The state will be mapped accordingly to the country of the user's choice.
   for example, if and only if the user selects Ireland the Irish states will be presented to the user because of the user's choice of the country.
5. In the PIN Code field the user is allowed to enter only numbers.
6. In Phone No. field the user is allowed to enter only numbers.
7. If the fields are populated successfully then the user will be notified (Order placed successfully!)
8. The user will be redirected to the home component (home page)

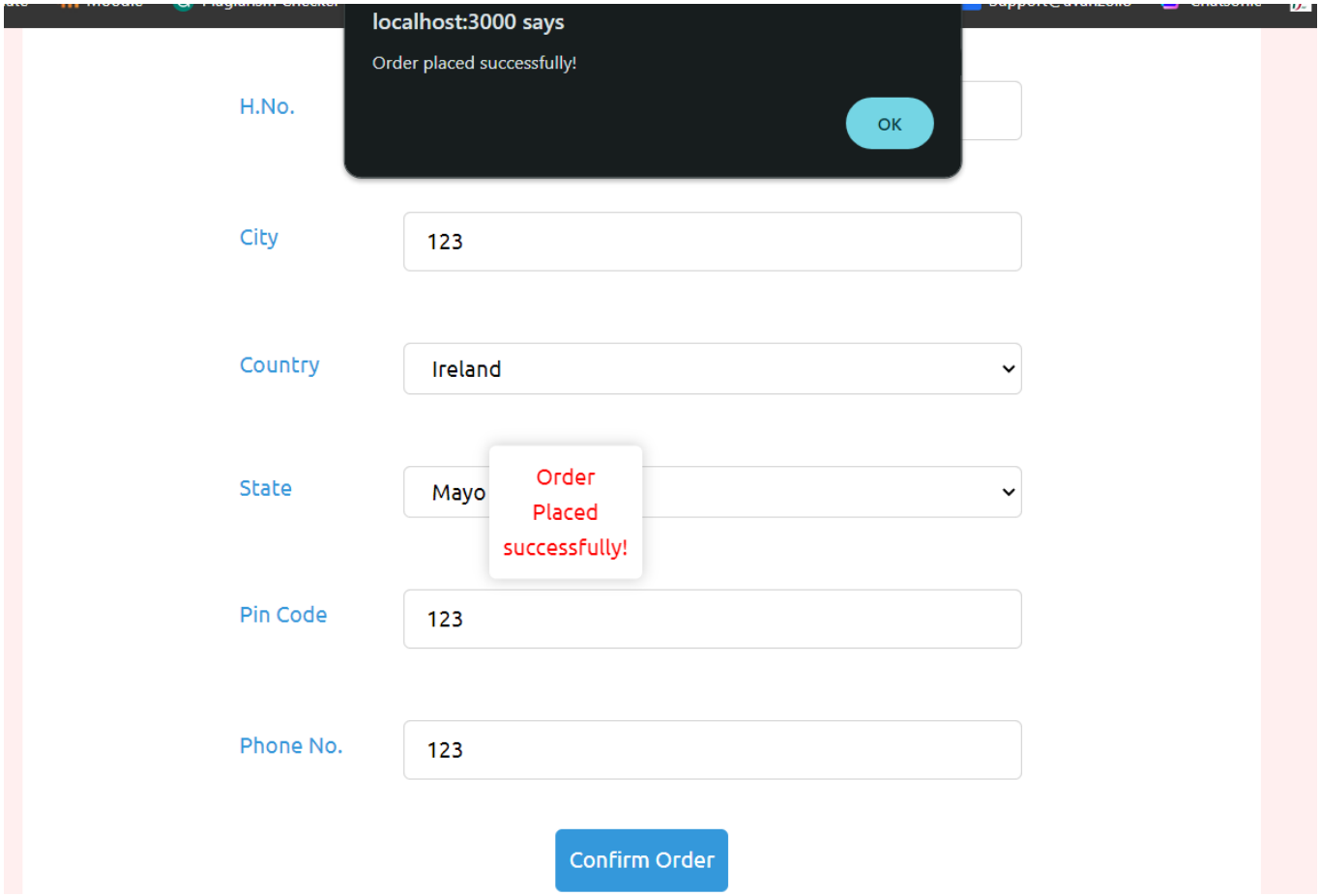
Here is the notified message informing the user that they should fill out the fields



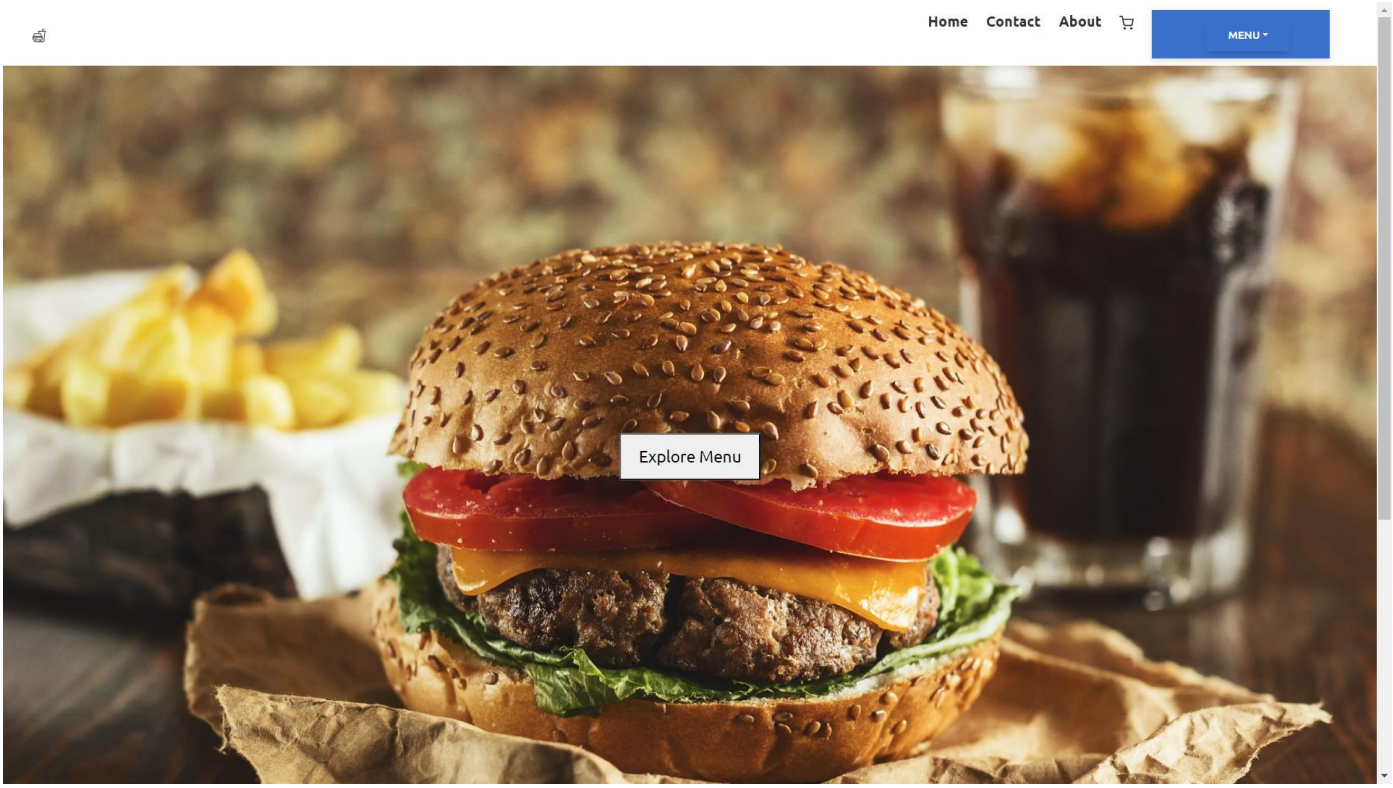
Here are the states if the user chooses Ireland



And here is the notifying alert that the order was placed successfully.



And here is the redirecting of the user to the home page.



Exercise 4

4.1 / Task 1: “My Orders” page

You will now be creating the “My Orders” page which will have the Order Id, Status, Item Qty, Amount, Payment Method, Action for each of your orders.

Navigate to myOrders/myOrders.jsx in the components folder. You will see the code for the <thead> tag present. You will now add the code for the <tbody> tag

1. Use the map method to iterate through each order ID and display the:

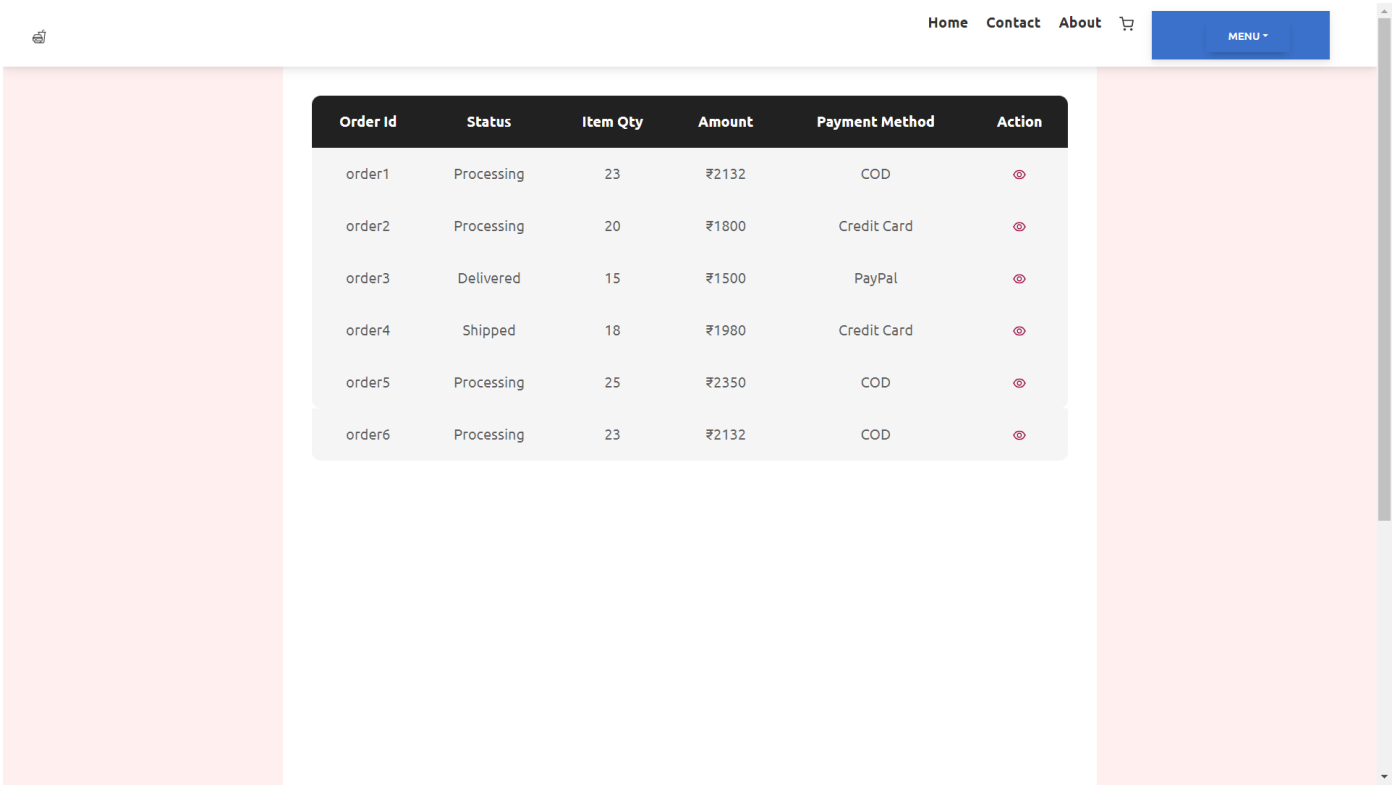
- Status as: Processing
- Item Qty as: 23
- Amount as: Rs ₹2132
- Payment Method as: COD
- An “eye” icon which will redirect to the “Order details” page on clicking

SOLUTION

```
import React from "react";
import { Link } from "react-router-dom";
import { AiOutlineEye } from "react-icons/ai";

const MyOrders = () => {
  const orders = [
    { id: "order1", status: "Processing", qty: 23, amount: 2132, paymentMethod: "COD" },
    { id: "order2", status: "Processing", qty: 20, amount: 1800, paymentMethod: "Credit
Card" },
    { id: "order3", status: "Delivered", qty: 15, amount: 1500, paymentMethod: "PayPal" },
    { id: "order4", status: "Shipped", qty: 18, amount: 1980, paymentMethod: "Credit Card"
},
    { id: "order5", status: "Processing", qty: 25, amount: 2350, paymentMethod: "COD" },
    //This is added by me according to the brief
    { id: "order6", status: "Processing", qty: 23, amount: 2132, paymentMethod: "COD" },
  ];

  return (
    <section className="tableClass">
      <main>
        <table>
          <thead>
            <tr>
              <th>Order Id</th>
              <th>Status</th>
              <th>Item Qty</th>
              <th>Amount</th>
              <th>Payment Method</th>
              <th>Action</th>
            </tr>
```

```
          </thead>

          <tbody>
            {orders.map((order) => (
              <tr key={order.id}>
                <td>{order.id}</td>
                <td>{order.status}</td>
                <td>{order.qty}</td>
                <td>₹{order.amount}</td>
                <td>{order.paymentMethod}</td>
                <td>
                  <Link to={`/order/${order.id}`}>
                    <AiOutlineEye />
                  </Link>
                </td>
              </tr>
            ))}
          </tbody>
        </table>
      </main>
    </section>
  );
};

export default MyOrders;
```

And the outcome is;



The changes I have made are;

1. I have added more order IDs until number 6
2. Orderid number 6 is the order following the data required
    - Status as: Processing

- Item Qty as: 23
- Amount as: Rs ₹2132
- Payment Method as: COD
- An “eye” icon which will redirect to the “Order details” page on clicking

3. I have used const { id } = useParams(); to get the order ID from the URL

4.2 / Task 2: “Order details” page

You will now be completing the Order Details Page of the Burger Shop project.

Instructions:

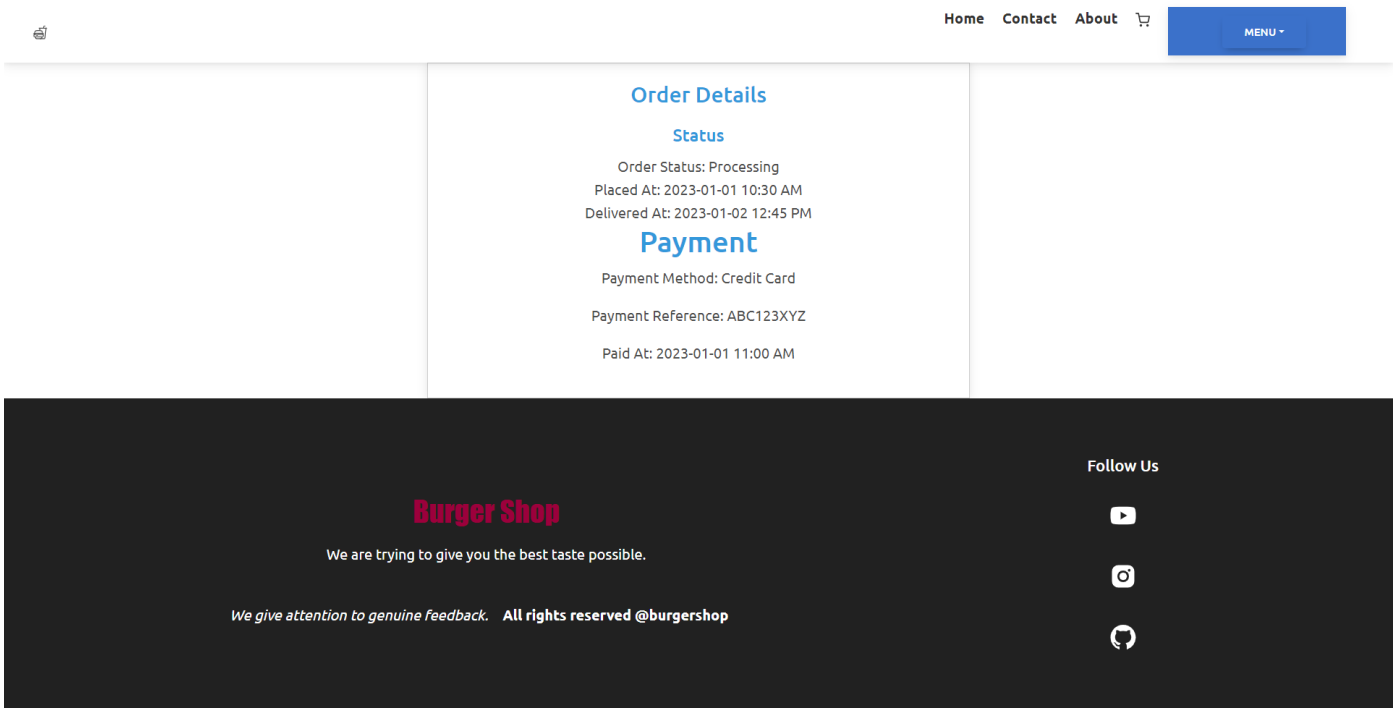
1. Go to the file - myOrders/OrderDetails.jsx in the “components” folder.
2. Create Status section with Order Status, placed at and Delivered At content & Payment section with Payment Method, Payment Reference and Paid At content of the Order Details Page

SOLUTION

```
import React from "react";
import { useParams } from "react-router-dom";

const OrderDetails = () => {
  // useParams hook to get the order ID from the URL
  const { id } = useParams();

  // Mock data for order details
  const orderDetails = {
    id: "order1",
    status: "Processing",
    placedAt: "2023-01-01 10:30 AM",
    deliveredAt: "2023-01-02 12:45 PM",
    paymentMethod: "Credit Card",
    paymentReference: "ABC123XYZ",
    paidAt: "2023-01-01 11:00 AM",
  };

  return (
    <section className="order-details">
      <main>
        <h1>Order Details</h1>

        {/* Status Section */}
        <div>
          <h2>Status</h2>
          <p>Order Status: {orderDetails.status}</p>
          <p>Placed At: {orderDetails.placedAt}</p>
          <p>Delivered At: {orderDetails.deliveredAt}</p>
        </div>

        {/* Payment Section */}
        <div>
          <h2>Payment</h2>
```

```
            <p>Payment Method: {orderDetails.paymentMethod}</p>
            <p>Payment Reference: {orderDetails.paymentReference}</p>
            <p>Paid At: {orderDetails.paidAt}</p>
          </div>
        </main>
      </section>
    );
};

export default OrderDetails;
```

And the outcome is;



I have changed the SCSS styling to make it look as much like an actual receipt.

And the last thing about the imports and routes placed in the App.js here is the solution for that

```
import { BrowserRouter as Router, Route, Routes } from "react-router-dom";
import Home from "./components/home/Home";
import Footer from "./components/layout/Footer";
import Header from "./components/layout/Header";
import Contact from "./components/contact/Contact";
// Add module imports for CART
import Cart from "./components/cart/Cart";
// Add module imports for Shipping
import Shipping from "./components/cart/Shipping";
import Login from "./components/login/Login";
import Profile from "./components/profile/Profile";
// Add module imports for MY ORDERS
```

```
import MyOrders from "./components/myOrders/MyOrders";
import OrderDetails from "./components/myOrders/OrderDetails";
import About from "./components/about/About";

import "./styles/app.scss";
import "./styles/header.scss";
import "./styles/home.scss";
import "./styles/founder.scss";
import "./styles/menu.scss";
import "./styles/footer.scss";
import "./styles/contact.scss";
import "./styles/cart.scss";
import "./styles/shipping.scss";
import "./styles/login.scss";
import "./styles/profile.scss";
import "./styles/table.scss";
import "./styles/orderDetails.scss";
import "./styles/about.scss";
import "./styles/DropdownMenu.scss";
import "./styles/colors.scss";
//I have created this styling just to make the order details look like an actual  reciept order
import "./styles/receipt.scss";

function App() {
  return (
    <Router>
      <Header isAuthenticated={true} />
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/contact" element={<Contact />} />
        <Route path="/about" element={<About />} />
        <Route path="/cart" element={<Cart />} />
        {<Route path="/shipping" element={<Shipping />} />}
        <Route path="/login" element={<Login />} />
        <Route path="/me" element={<Profile />} />
        <Route path="/myOrders" element={<MyOrders />} />
        <Route path="/order/:id" element={<OrderDetails />} />
      </Routes>
      <Footer />
    </Router>
  );
}

export default App;
```

I have imported the rest if the required styling SCSS files and the components.
I have established routes for the components.
For example myOrders component here is the route for it

```
<Route path="/myOrders" element={<MyOrders />} />
```