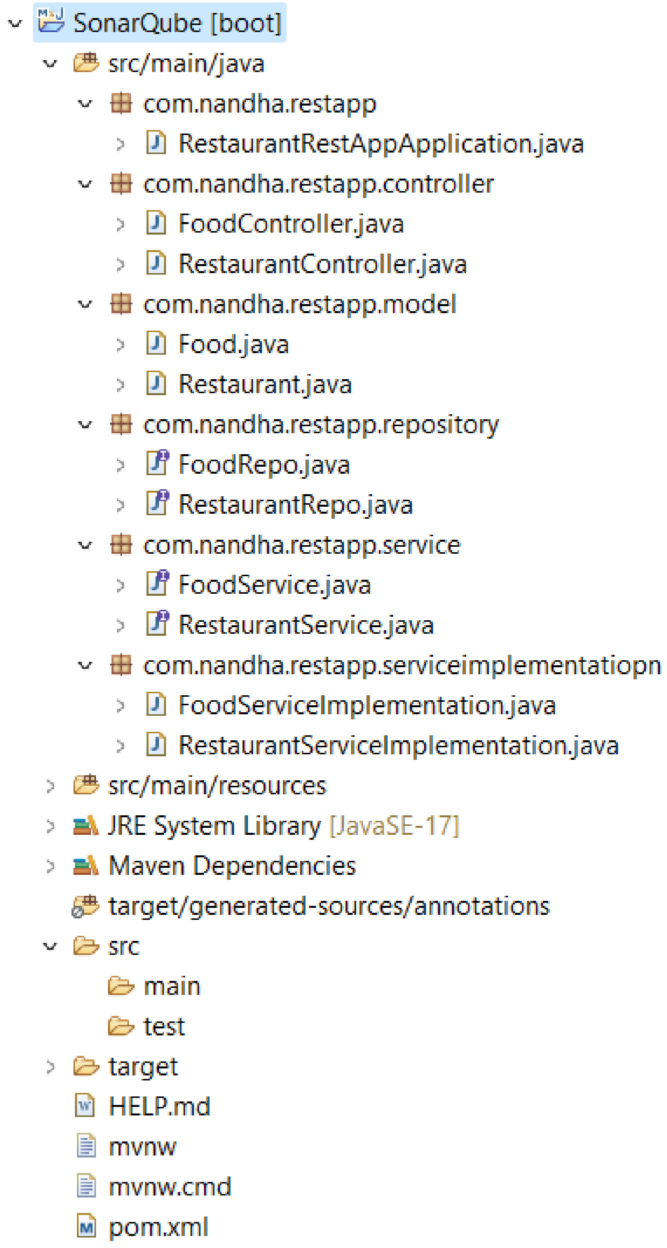
Name: Nandhakumaran H

Employee ID: 12225

Case Study: Restaurant Management System



```
SonarQube [boot]
  src/main/java
    com.nandha.restapp
      RestaurantRestAppApplication.java
    com.nandha.restapp.controller
      FoodController.java
      RestaurantController.java
    com.nandha.restapp.model
      Food.java
      Restaurant.java
    com.nandha.restapp.repository
      FoodRepo.java
      RestaurantRepo.java
    com.nandha.restapp.service
      FoodService.java
      RestaurantService.java
    com.nandha.restapp.serviceimplementatiopn
      FoodServiceImplementation.java
      RestaurantServiceImplementation.java
  src/main/resources
  JRE System Library [JavaSE-17]
  Maven Dependencies
  target/generated-sources/annotations
  src
    main
    test
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

BackendCode:

RestaurantRestAppApplication,java

```java
package com.nandha.restapp;

import org.springframework.boot.SpringApplication;
import
org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class RestaurantRestAppApplication {

public static void main(String[] args) {
SpringApplication.run(RestaurantRestAppApplication.class, args);
}

}
```

FoodController.java

```java
package com.nandha.restapp.controller;

import java.util.List;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import com.nandha.restapp.model.Food;
import com.nandha.restapp.service.FoodService;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;

@CrossOrigin("http://localhost:3001,http://localhost:3000,http:/
/localhost:3002,http://localhost:3003")
@RestController
@RequestMapping("/food")
public class FoodController {
```

```java
FoodService service;

public FoodController(FoodService service) {
super();
this.service = service;
}

@PostMapping
public Food insertFood(@RequestBody Food fd) {

return service.addFood(fd);
}

@GetMapping("{id}")
public Food getFoodById(@PathVariable("id") int id) {
return service.getFood(id);
}

@GetMapping("/all")
public List<Food> viewallFood() {
return service.getAllFood();
}

@PutMapping
public Food updateFood(@RequestBody Food fd) {

return service.updateFood(fd);
}

@DeleteMapping("{id}")
public String deleteFoodById(@PathVariable("id") int id) {

return service.deleteFoodById(id);
}
}
```

RestaurantController.java

```java
package com.nandha.restapp.controller;
```

```
import java.util.List;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import com.nandha.restapp.model.Restaurant;
import com.nandha.restapp.service.RestaurantService;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;

@CrossOrigin("http://localhost:3001,http://localhost:3000,http:/
/localhost:3002,http://localhost:3003")
@RestController
@RequestMapping("/restaurant")
public class RestaurantController {

RestaurantService service;

public RestaurantController(RestaurantService service) {
super();
this.service = service;
}

@PostMapping
public Restaurant insertRestaurant(@RequestBody Restaurant res)
{

return service.addRestaurant(res);
}

@GetMapping("{id}")
public Restaurant getRestaurantById(@PathVariable("id") int id)
{
return service.getRestaurant(id);
}

@GetMapping("/all")
public List<Restaurant> viewAllRestaurant() {
return service.getAllRestaurant();
}

@PutMapping
public Restaurant updateRestaurant(@RequestBody Restaurant res)
{

return service.updateRestaurant(res);
}

@DeleteMapping("{id}")
public String deleteRestaurantById(@PathVariable("id") int id) {

return service.deleteRestaurantbyId(id);
}
}
```

Food.java

```
package com.nandha.restapp.model;

import jakarta.persistence.CascadeType;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.Table;

@Entity
@Table(name = "Food")
public class Food {

@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
@Column(name = "fid")
private int fid;

@Column(name = "foodName")
private String foodName;

@Column(name = "price")
private int price;

@Column(name = "quantity")
private int quantity;

@Column(name = "totalAmount")
private int totalAmount;

@ManyToOne(targetEntity = Restaurant.class, cascade =
CascadeType.MERGE)
@JoinColumn(name = "restaurantId")
private Restaurant restaurant;

public Food() {
super();
}

public Food(int fid, String foodName, int price, int quantity,
int totalAmount, Restaurant restaurant) {
super();
this.fid = fid;
this.foodName = foodName;
this.price = price;
this.quantity = quantity;
this.totalAmount = totalAmount;
this.restaurant = restaurant;
}

public int getFid() {
return fid;
}

public void setFid(int fid) {
```

```
this.fid = fid;
}

public String getFoodName() {
return foodName;
}

public void setFoodName(String foodName) {
this.foodName = foodName;
}

public int getPrice() {
return price;
}

public void setPrice(int price) {
this.price = price;
}

public int getQuantity() {
return quantity;
}

public void setQuantity(int quantity) {
this.quantity = quantity;
}

public int getTotalAmount() {
return totalAmount;
}

public void setTotalAmount(int totalAmount) {
this.totalAmount = totalAmount;
}

public Restaurant getRestaurant() {
return restaurant;
}

public void setRestaurant(Restaurant restaurant) {
```

```java
this.restaurant = restaurant;
}



}
```

Restaurant.java

```java
package com.nandha.restapp.model;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;

@Entity
@Table(name = "restaurant")
public class Restaurant {

@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "restaurantId")
private int restaurantId;

@Column(name = "name")
private String name;

@Column(name = "ratings")
private int ratings;

@Column(name = "phnum")
private long phnum;

public Restaurant() {
super();
}
```

```
public Restaurant(int restaurantId, String name, int ratings,
long phnum) {
super();
this.restaurantId = restaurantId;
this.name = name;
this.ratings = ratings;
this.phnum = phnum;
}

public int getRestaurantId() {
return restaurantId;
}

public void setRestaurantId(int restaurantId) {
this.restaurantId = restaurantId;
}

public String getName() {
return name;
}

public void setName(String name) {
this.name = name;
}

public int getRatings() {
return ratings;
}

public void setRatings(int ratings) {
this.ratings = ratings;
}

public long getPhnum() {
return phnum;
}

public void setPhnum(long phnum) {
this.phnum = phnum;
}


}
```

FoodRepo.java

```
package com.nandha.restapp.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
import com.nandha.restapp.model.Food;

@Repository
public interface FoodRepo extends JpaRepository<Food, Integer> {

}
```

RestaurantRepo.java

```
package com.nandha.restapp.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
import com.nandha.restapp.model.Restaurant;

@Repository
public interface RestaurantRepo extends
JpaRepository<Restaurant, Integer> {

}
```

FoodService.java

```
package com.nandha.restapp.service;
```

```java
import java.util.List;
import com.nandha.restapp.model.Food;

public interface FoodService {

public Food addFood(Food fd);

public Food getFood(int id);

public List<Food> getAllFood();

public Food updateFood(Food fd);

public String deleteFoodById(int id);
}
```

RestaurantService.java

```java
package com.nandha.restapp.service;

import java.util.List;
import com.nandha.restapp.model.Restaurant;

public interface RestaurantService {

public Restaurant addRestaurant(Restaurant res);

public Restaurant getRestaurant(int id);

public List<Restaurant> getAllRestaurant();

public Restaurant updateRestaurant(Restaurant res);

public String deleteRestaurantbyId(int id);
}
```

FoodServiceImplementation.java

```
package com.nandha.restapp.serviceimplementatiopn;

import java.util.List;
import org.springframework.stereotype.Service;
import com.nandha.restapp.model.Food;
import com.nandha.restapp.repository.FoodRepo;
import com.nandha.restapp.service.FoodService;

@Service
public class FoodServiceImplementation implements FoodService {

FoodRepo repo;

public FoodServiceImplementation(FoodRepo repo) {
super();
this.repo = repo;
}

public Food addFood(Food fd) {
return repo.save(fd);
}

@Override
public Food getFood(int id) {
return repo.findById(id).orElse(null);
}
public List<Food> getAllFood() {
return repo.findAll();
}

public Food updateFood(Food fd) {
return repo.save(fd);
}

public String deleteFoodById(int id) {
repo.deleteById(id);
return null;
}
}
```

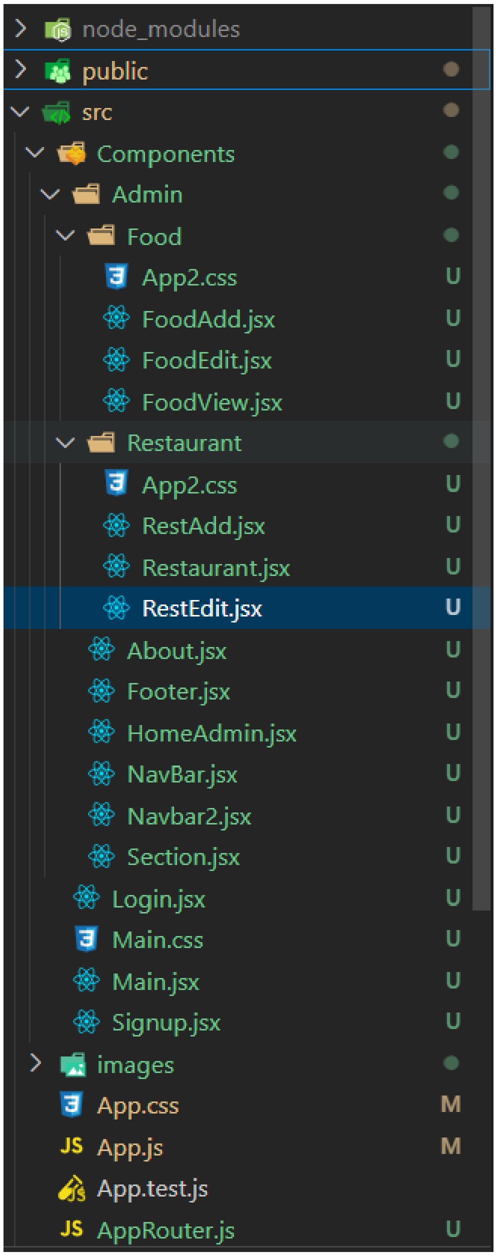
RestaurantServiceImplementation.java

```
package com.nandha.restapp.serviceimplementatiopn;

import java.util.List;
import org.springframework.stereotype.Service;
import com.nandha.restapp.model.Restaurant;
import com.nandha.restapp.repository.RestaurantRepo;
import com.nandha.restapp.service.RestaurantService;

@Service
```

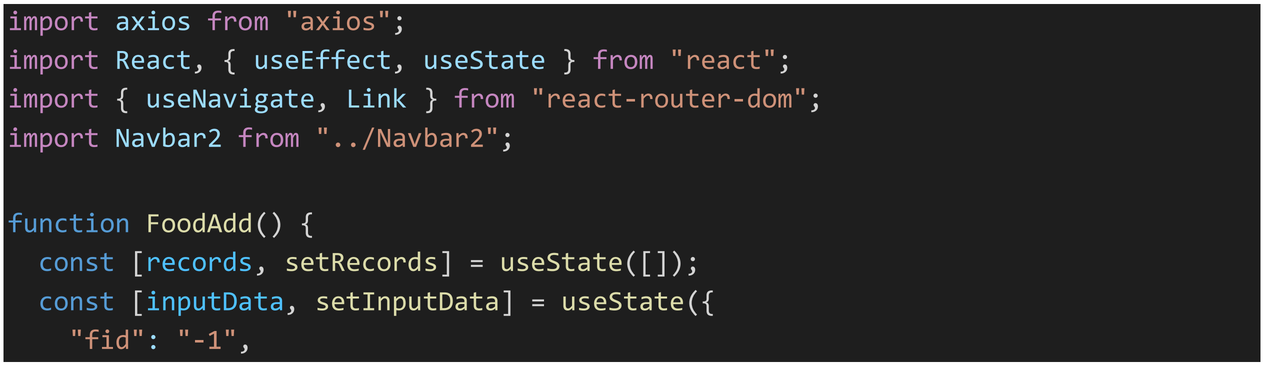
```
public class RestaurantServiceImplementation implements RestaurantService {

RestaurantRepo repo;

public RestaurantServiceImplementation(RestaurantRepo repo) {
super();
this.repo = repo;
}

@Override
public Restaurant addRestaurant(Restaurant res) {
return repo.save(res);
}

@Override
public Restaurant getRestaurant(int id) {
return repo.findById(id).orElse(null);
}

@Override
public List<Restaurant> getAllRestaurant() {
return repo.findAll();
}

@Override
public Restaurant updateRestaurant(Restaurant res) {
```

```
return repo.save(res);

}


@Override

public String deleteRestaurantbyId(int id) {

repo.deleteById(id);

return null;

}


}
```

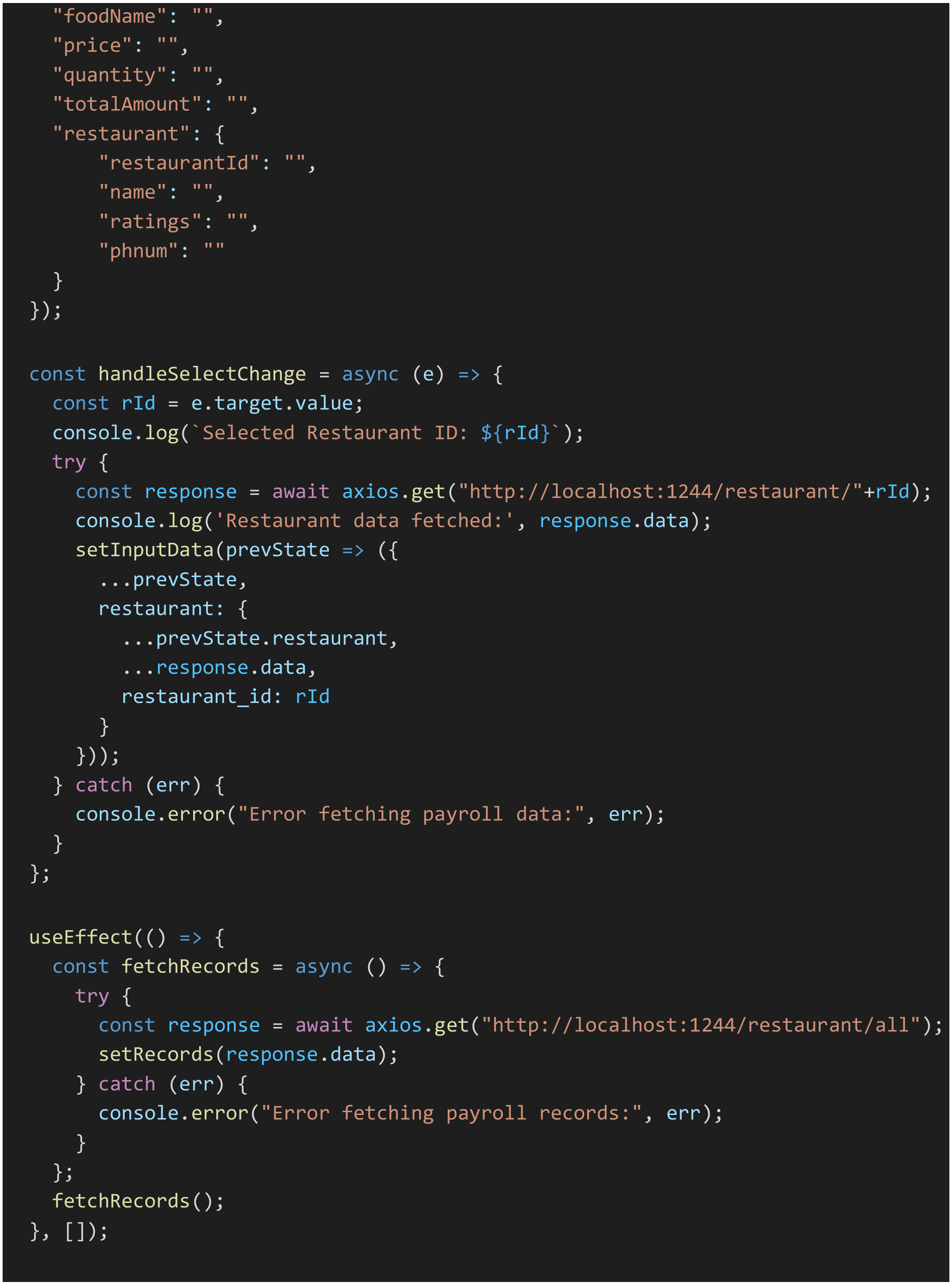
Frontend Code:



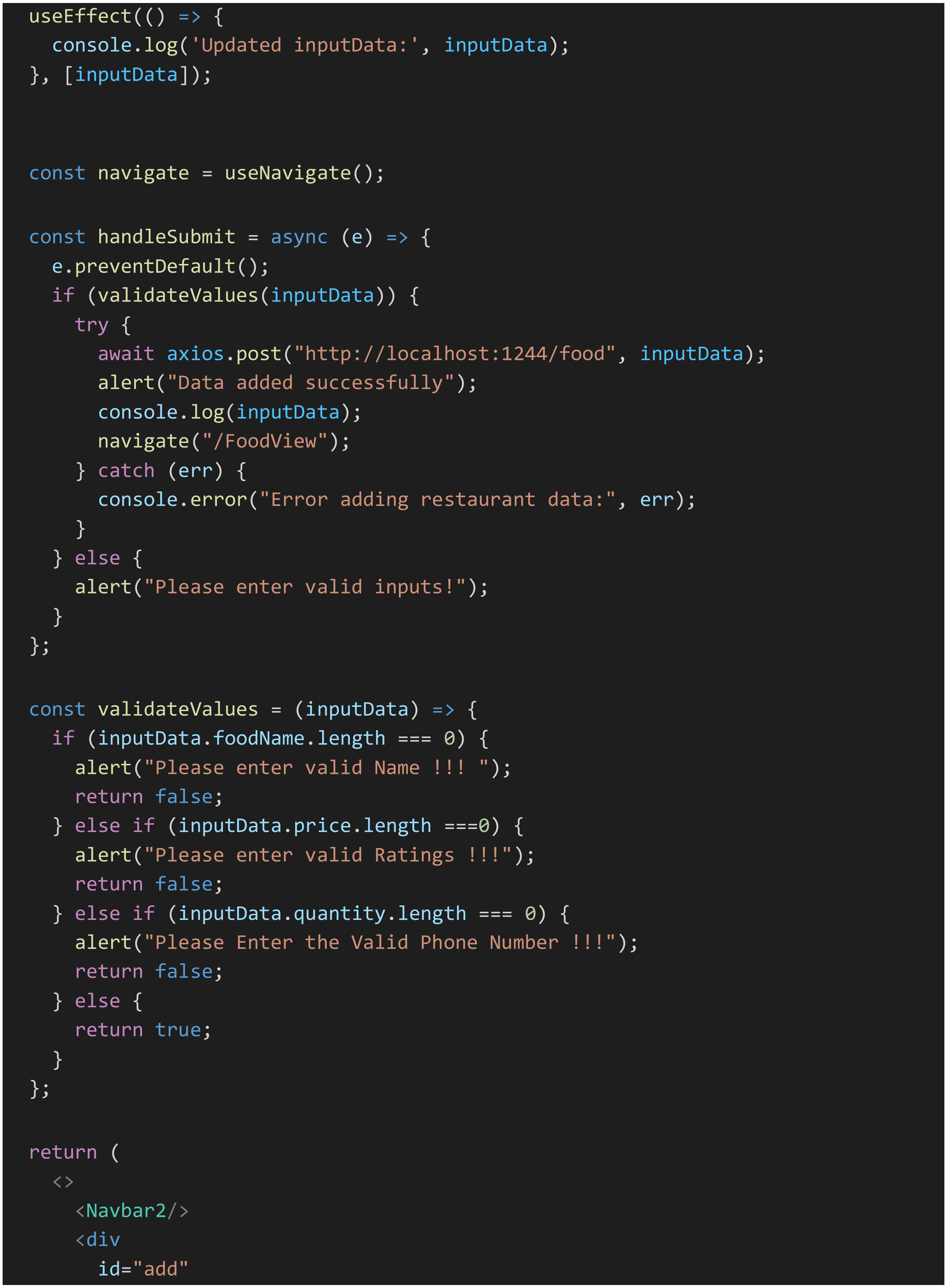
```
> node_modules
> public
v src
  v Components
    v Admin
      v Food
          App2.css          U
          FoodAdd.jsx       U
          FoodEdit.jsx      U
          FoodView.jsx      U
      v Restaurant
          App2.css          U
          RestAdd.jsx       U
          Restaurant.jsx    U
          RestEdit.jsx      U
        About.jsx           U
        Footer.jsx          U
        HomeAdmin.jsx       U
        NavBar.jsx          U
        Navbar2.jsx         U
        Section.jsx         U
      Login.jsx             U
      Main.css              U
      Main.jsx              U
      Signup.jsx            U
  > images
    App.css                 M
    App.js                  M
    App.test.js
    AppRouter.js            U
```



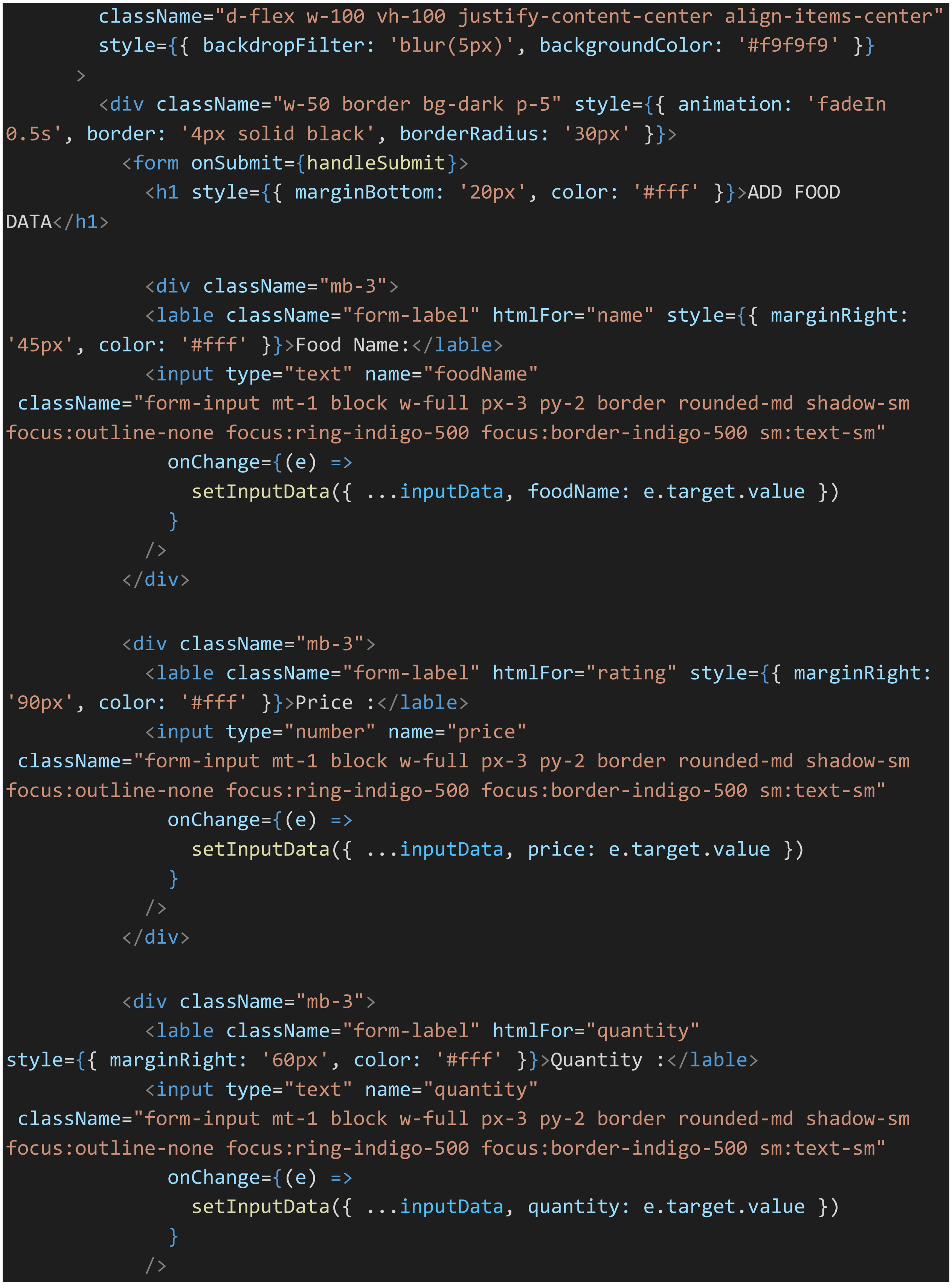
```
import axios from "axios";
import React, { useEffect, useState } from "react";
import { useNavigate, Link } from "react-router-dom";
import Navbar2 from "../Navbar2";

function FoodAdd() {
  const [records, setRecords] = useState([]);
  const [inputData, setInputData] = useState({
    "fid": "-1",
```



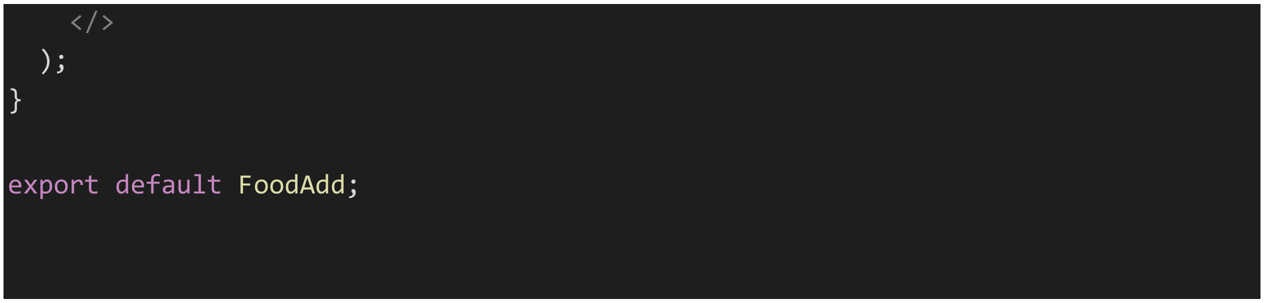
```
  "foodName": "",
  "price": "",
  "quantity": "",
  "totalAmount": "",
  "restaurant": {
      "restaurantId": "",
      "name": "",
      "ratings": "",
      "phnum": ""
  }
});

const handleSelectChange = async (e) => {
  const rId = e.target.value;
  console.log(`Selected Restaurant ID: ${rId}`);
  try {
    const response = await axios.get("http://localhost:1244/restaurant/"+rId);
    console.log('Restaurant data fetched:', response.data);
    setInputData(prevState => ({
      ...prevState,
      restaurant: {
        ...prevState.restaurant,
        ...response.data,
        restaurant_id: rId
      }
    }));
  } catch (err) {
    console.error("Error fetching payroll data:", err);
  }
};

useEffect(() => {
  const fetchRecords = async () => {
    try {
      const response = await axios.get("http://localhost:1244/restaurant/all");
      setRecords(response.data);
    } catch (err) {
      console.error("Error fetching payroll records:", err);
    }
  };
  fetchRecords();
}, []);
```



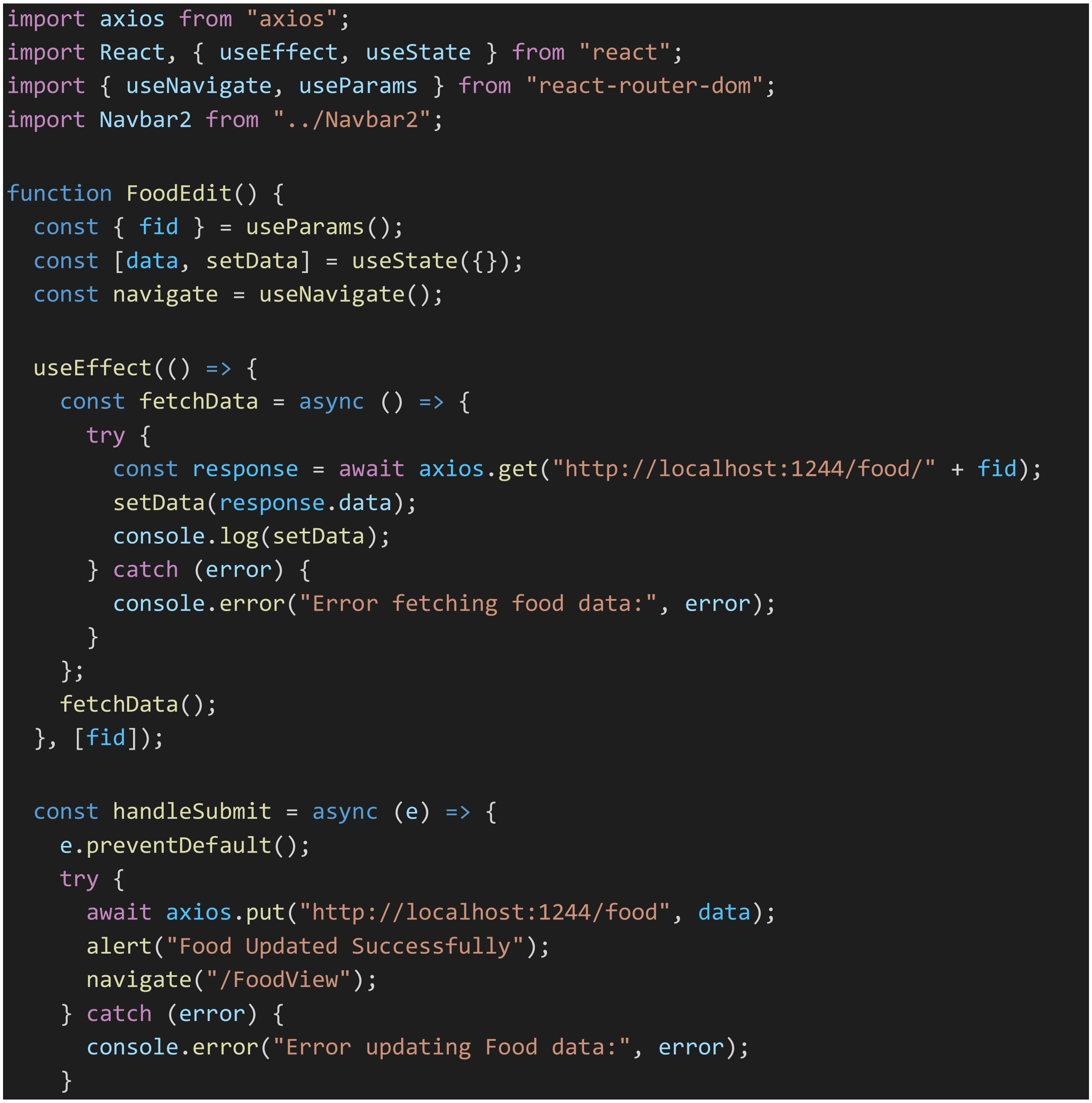
```
useEffect(() => {
  console.log('Updated inputData:', inputData);
}, [inputData]);



const navigate = useNavigate();

const handleSubmit = async (e) => {
  e.preventDefault();
  if (validateValues(inputData)) {
    try {
      await axios.post("http://localhost:1244/food", inputData);
      alert("Data added successfully");
      console.log(inputData);
      navigate("/FoodView");
    } catch (err) {
      console.error("Error adding restaurant data:", err);
    }
  } else {
    alert("Please enter valid inputs!");
  }
};

const validateValues = (inputData) => {
  if (inputData.foodName.length === 0) {
    alert("Please enter valid Name !!! ");
    return false;
  } else if (inputData.price.length ===0) {
    alert("Please enter valid Ratings !!!");
    return false;
  } else if (inputData.quantity.length === 0) {
    alert("Please Enter the Valid Phone Number !!!");
    return false;
  } else {
    return true;
  }
};

return (
  <>
    <Navbar2/>
    <div
      id="add"
```



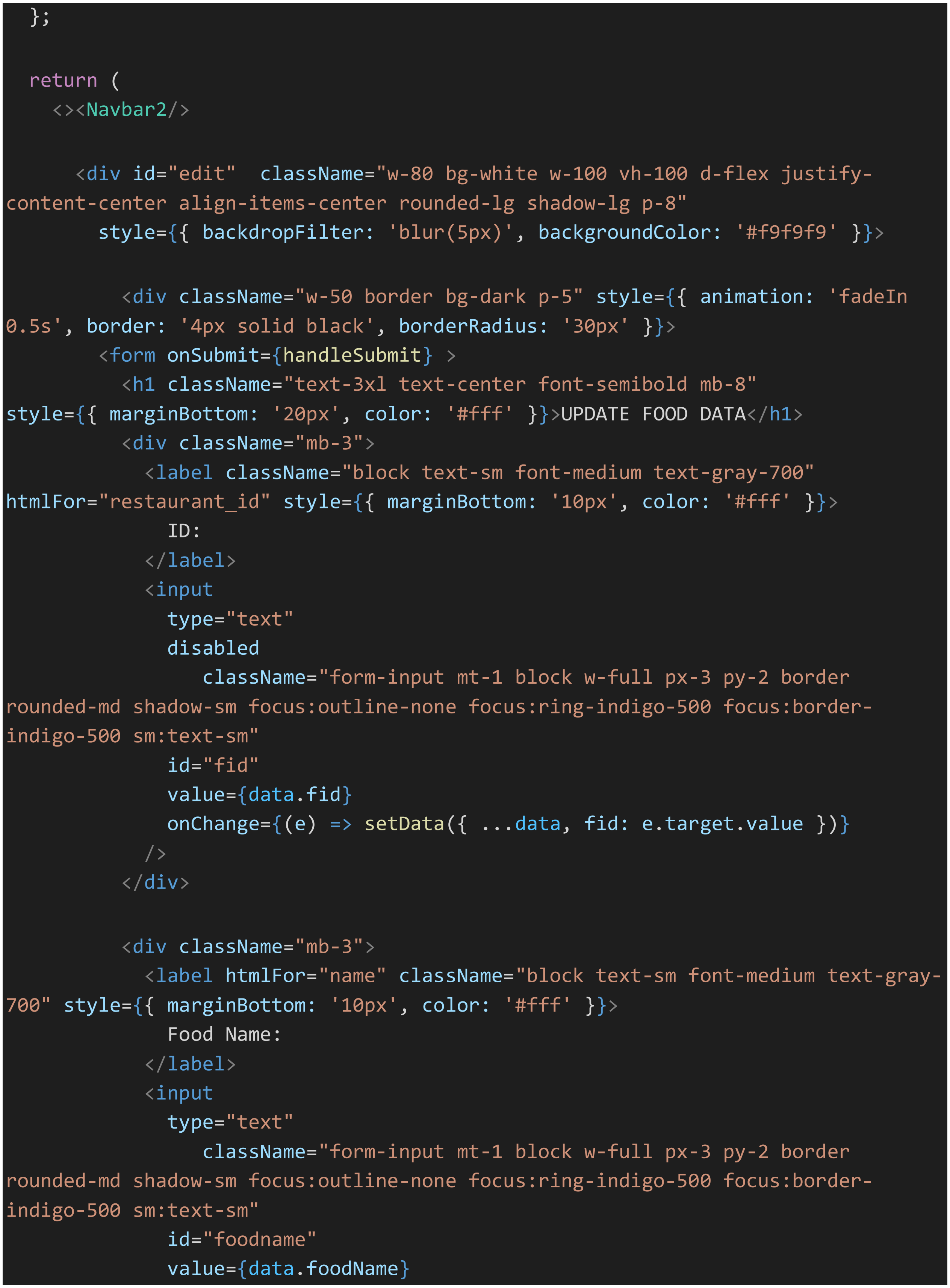
```
        className="d-flex w-100 vh-100 justify-content-center align-items-center"
        style={{ backdropFilter: 'blur(5px)', backgroundColor: '#f9f9f9' }}
      >
        <div className="w-50 border bg-dark p-5" style={{ animation: 'fadeIn
0.5s', border: '4px solid black', borderRadius: '30px' }}>
          <form onSubmit={handleSubmit}>
            <h1 style={{ marginBottom: '20px', color: '#fff' }}>ADD FOOD
DATA</h1>

            <div className="mb-3">
            <lable className="form-label" htmlFor="name" style={{ marginRight:
'45px', color: '#fff' }}>Food Name:</lable>
            <input type="text" name="foodName"
 className="form-input mt-1 block w-full px-3 py-2 border rounded-md shadow-sm
focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm"
              onChange={(e) =>
                setInputData({ ...inputData, foodName: e.target.value })
              }
            />
          </div>

          <div className="mb-3">
            <lable className="form-label" htmlFor="rating" style={{ marginRight:
'90px', color: '#fff' }}>Price :</lable>
            <input type="number" name="price"
 className="form-input mt-1 block w-full px-3 py-2 border rounded-md shadow-sm
focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm"
              onChange={(e) =>
                setInputData({ ...inputData, price: e.target.value })
              }
            />
          </div>

          <div className="mb-3">
            <lable className="form-label" htmlFor="quantity"
style={{ marginRight: '60px', color: '#fff' }}>Quantity :</lable>
            <input type="text" name="quantity"
 className="form-input mt-1 block w-full px-3 py-2 border rounded-md shadow-sm
focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm"
              onChange={(e) =>
                setInputData({ ...inputData, quantity: e.target.value })
              }
            />
          </div>

          <div className="mb-3">
            <lable className="form-label" htmlFor="totalamount"
style={{ marginRight: '20px', color: '#fff' }}>Total Amount :</lable>
            <input type="text" name="totalAmount"
 className="form-input mt-1 block w-full px-3 py-2 border rounded-md shadow-sm
focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm"
              value={parseInt(inputData.price)*parseInt(inputData.quantity)}
              onBlur={(e) =>
                setInputData({ ...inputData, totalAmount: e.target.value })
              }

            />
          </div >

          <div className="mb-3">
            <label className="form-label" style={{ marginRight: '70px', color:
'#fff' }}>Restaurant:</label>
            <select
name="selectedId" value={inputData.restaurant.restaurantId}
onChange={handleSelectChange}  className="form-input mt-1 block w-full px-3 py-2
border rounded-md shadow-sm focus:outline-none focus:ring-indigo-500
focus:border-indigo-500 sm:text-sm">
              <option value="">Select Restaurant</option>
              {records.map((d, i) => (
                <option key={i} value={d.restaurantId}>{d.restaurantId}</option>
              ))}
            </select>
          </div>

            <button type="submit" className="btn btn-info" style={{ fontSize:
'16px', padding: '10px 20px' }}>
              Submit
            </button>

            <Link to="#" className="btn btn-primary" style={{ marginLeft:
'10px' }}>
              Menu
            </Link>
          </form>
        </div>
      </div>
```



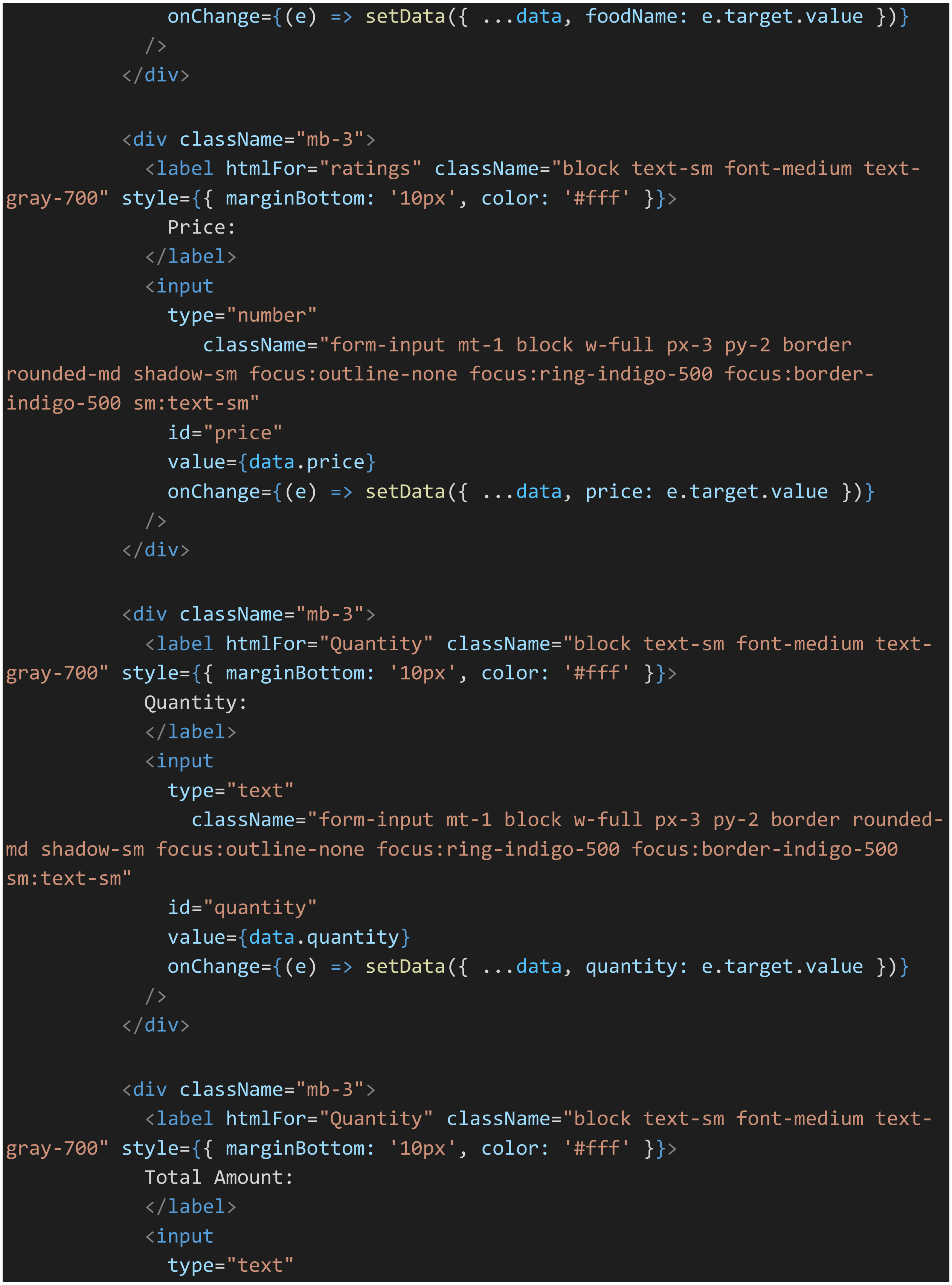
```
    </>
  );
}

export default FoodAdd;
```



```
import axios from "axios";
import React, { useEffect, useState } from "react";
import { useNavigate, useParams } from "react-router-dom";
import Navbar2 from "../Navbar2";

function FoodEdit() {
  const { fid } = useParams();
  const [data, setData] = useState({});
  const navigate = useNavigate();

  useEffect(() => {
    const fetchData = async () => {
      try {
        const response = await axios.get("http://localhost:1244/food/" + fid);
        setData(response.data);
        console.log(setData);
      } catch (error) {
        console.error("Error fetching food data:", error);
      }
    };
    fetchData();
  }, [fid]);

  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      await axios.put("http://localhost:1244/food", data);
      alert("Food Updated Successfully");
      navigate("/FoodView");
    } catch (error) {
      console.error("Error updating Food data:", error);
    }
```



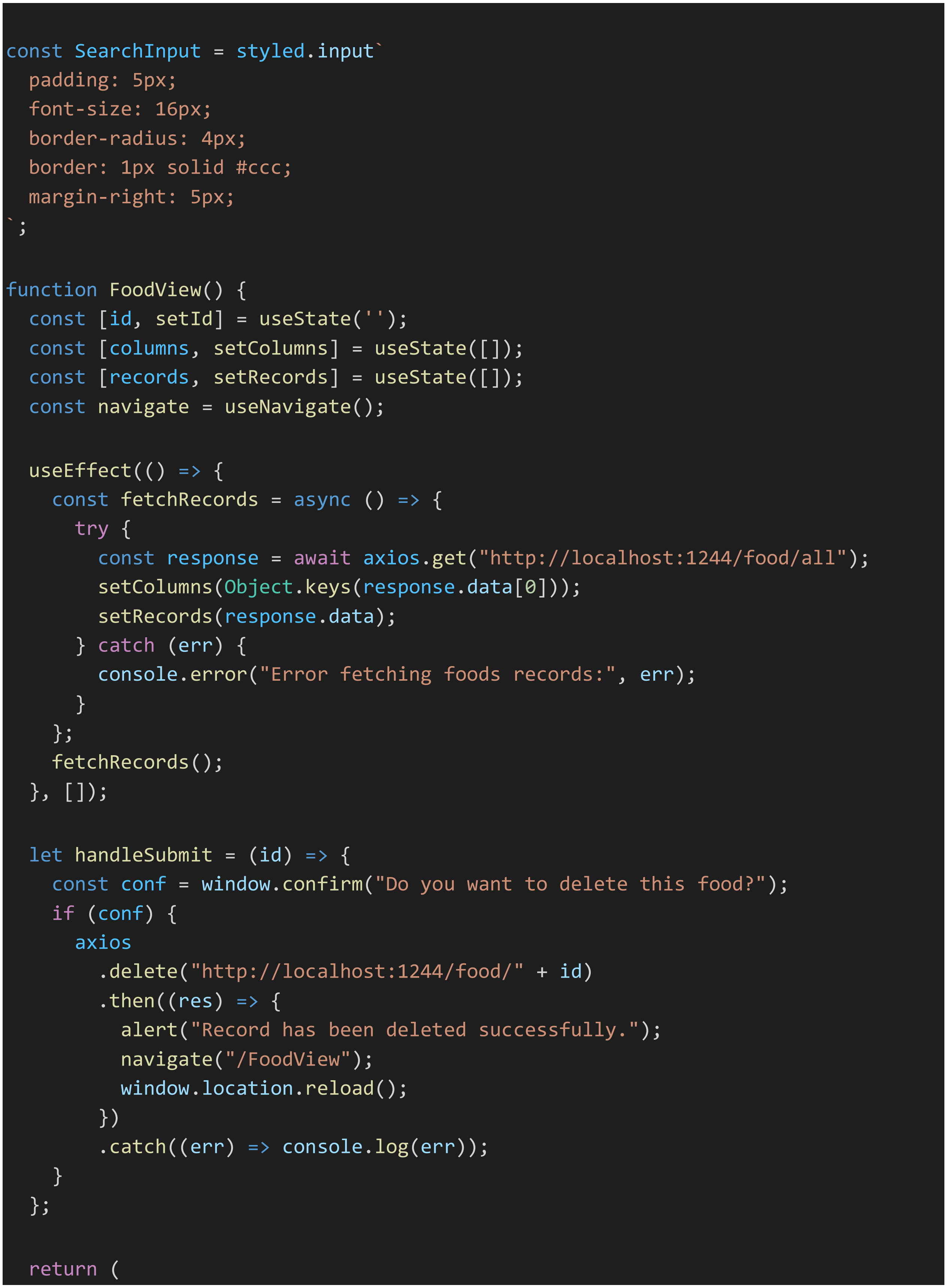
```
  };

  return (
    <><Navbar2/>

      <div id="edit"  className="w-80 bg-white w-100 vh-100 d-flex justify-
content-center align-items-center rounded-lg shadow-lg p-8"
        style={{ backdropFilter: 'blur(5px)', backgroundColor: '#f9f9f9' }}>

          <div className="w-50 border bg-dark p-5" style={{ animation: 'fadeIn
0.5s', border: '4px solid black', borderRadius: '30px' }}>
        <form onSubmit={handleSubmit} >
          <h1 className="text-3xl text-center font-semibold mb-8"
style={{ marginBottom: '20px', color: '#fff' }}>UPDATE FOOD DATA</h1>
          <div className="mb-3">
            <label className="block text-sm font-medium text-gray-700"
htmlFor="restaurant_id" style={{ marginBottom: '10px', color: '#fff' }}>
              ID:
            </label>
            <input
              type="text"
              disabled
                 className="form-input mt-1 block w-full px-3 py-2 border
rounded-md shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-
indigo-500 sm:text-sm"
              id="fid"
              value={data.fid}
              onChange={(e) => setData({ ...data, fid: e.target.value })}
            />
          </div>

          <div className="mb-3">
            <label htmlFor="name" className="block text-sm font-medium text-gray-
700" style={{ marginBottom: '10px', color: '#fff' }}>
              Food Name:
            </label>
            <input
              type="text"
                 className="form-input mt-1 block w-full px-3 py-2 border
rounded-md shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-
indigo-500 sm:text-sm"
              id="foodname"
              value={data.foodName}
```



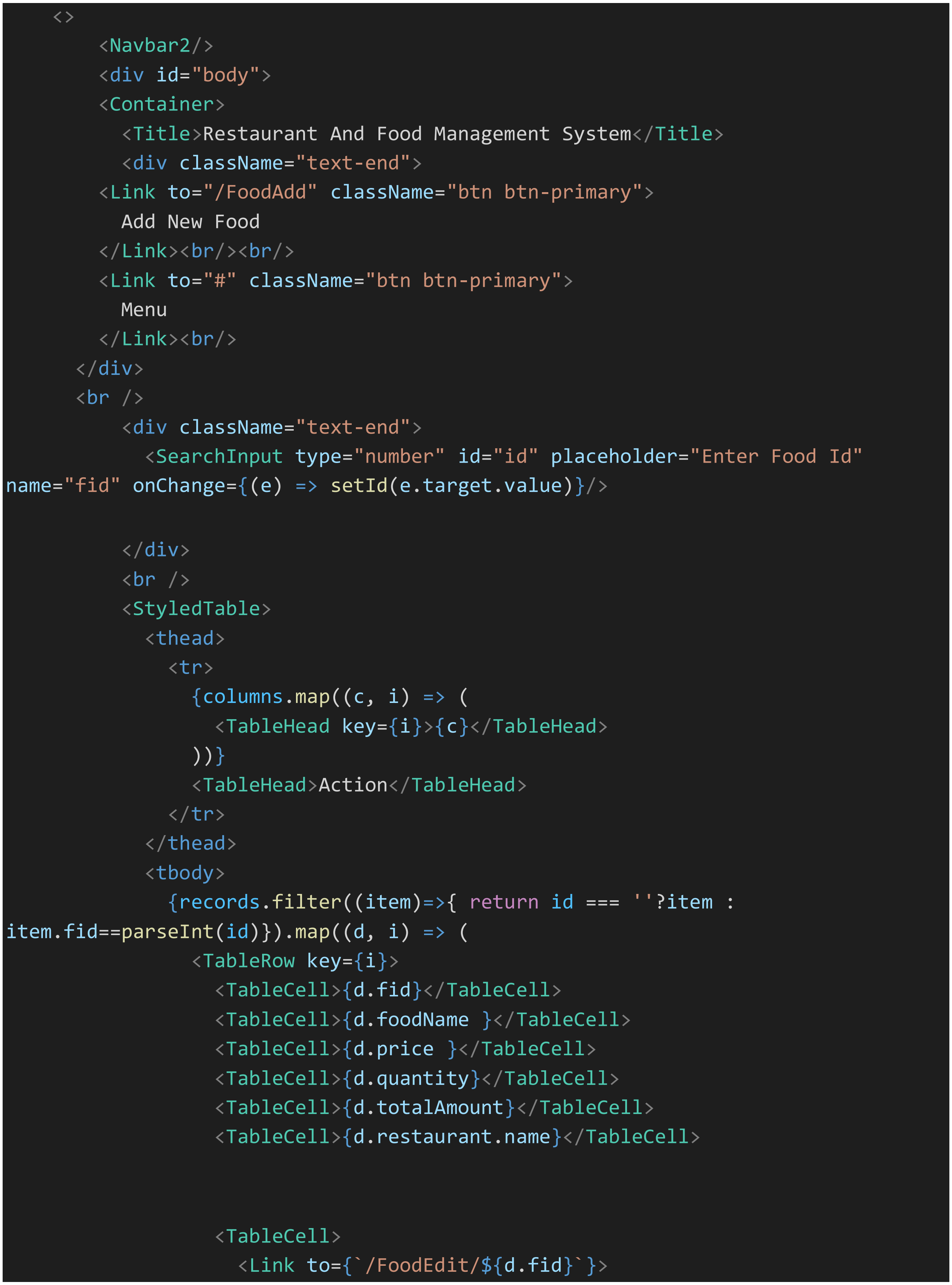
```
              onChange={(e) => setData({ ...data, foodName: e.target.value })}
            />
          </div>

          <div className="mb-3">
            <label htmlFor="ratings" className="block text-sm font-medium text-gray-700" style={{ marginBottom: '10px', color: '#fff' }}>
              Price:
            </label>
            <input
              type="number"
                 className="form-input mt-1 block w-full px-3 py-2 border rounded-md shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm"
              id="price"
              value={data.price}
              onChange={(e) => setData({ ...data, price: e.target.value })}
            />
          </div>

          <div className="mb-3">
            <label htmlFor="Quantity" className="block text-sm font-medium text-gray-700" style={{ marginBottom: '10px', color: '#fff' }}>
            Quantity:
            </label>
            <input
              type="text"
                className="form-input mt-1 block w-full px-3 py-2 border rounded-md shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm"
              id="quantity"
              value={data.quantity}
              onChange={(e) => setData({ ...data, quantity: e.target.value })}
            />
          </div>

          <div className="mb-3">
            <label htmlFor="Quantity" className="block text-sm font-medium text-gray-700" style={{ marginBottom: '10px', color: '#fff' }}>
            Total Amount:
            </label>
            <input
              type="text"
```

```
                 className="form-input mt-1 block w-full px-3 py-2 border rounded-md shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm"
              id="totalAmount"
              value={parseInt(data.price)*parseInt(data.quantity)}
              onBlur={(e) =>
                setData({ ...data, totalAmount: e.target.value })
              }
            />
          </div>

          <button type="submit" className="w-full bg-indigo-500 text-white p-3 rounded-md font-semibold hover:bg-indigo-600 transition duration-300">
            Update
          </button>
        </form></div>
      </div>
    </>
  );
}

export default FoodEdit;
```

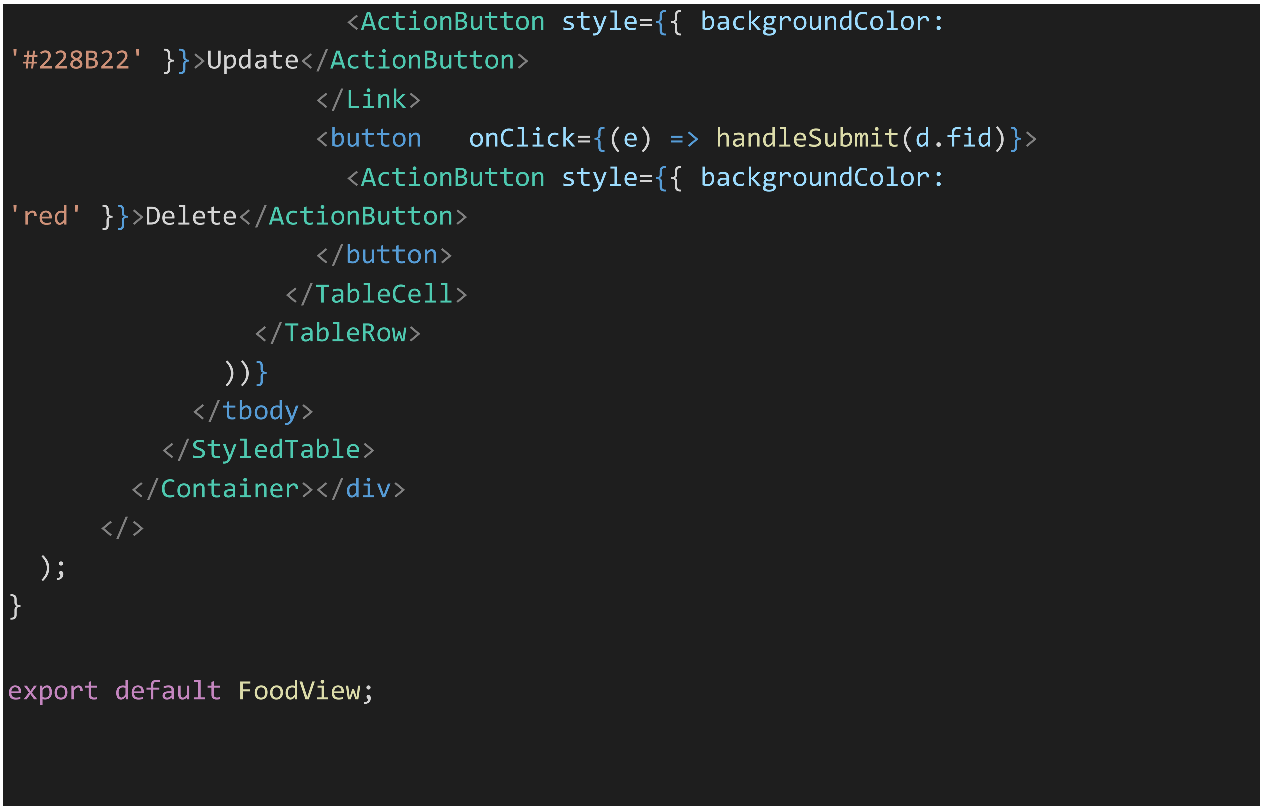
```
import React, { useEffect, useState } from "react";
import axios from "axios";
import { Link, useNavigate } from "react-router-dom";
import Navbar2 from "../Navbar2";
import "./App2.css";

import styled from 'styled-components';

const Container = styled.div`
  background-color: #f9f9f9;
  padding: 20px;
`;

const Title = styled.h1`
  text-align: center;
  background-color: #f9f9f9;
  padding: 10px;
  margin-bottom: 20px;
  color: #333;
`;

const StyledTable = styled.table`
  width: 100%;
  border-collapse: collapse;
  background-color: #fff;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
  border-radius: 8px;
`;

const TableHead = styled.th`
  background-color: #f9f9f9;
  color: #333;
  padding: 10px;
  text-align: left;
`;

const TableRow = styled.tr`
  &:nth-child(even) {
    background-color: #f5f5f5;
  }
`;

const TableCell = styled.td`
  padding: 10px;
`;

const ActionButton = styled.button`
  background-color: #007bff;
  color: #fff;
  padding: 5px 10px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  margin-right: 5px;

  &:hover {
    background-color: #0056b3;
  }
`;
```



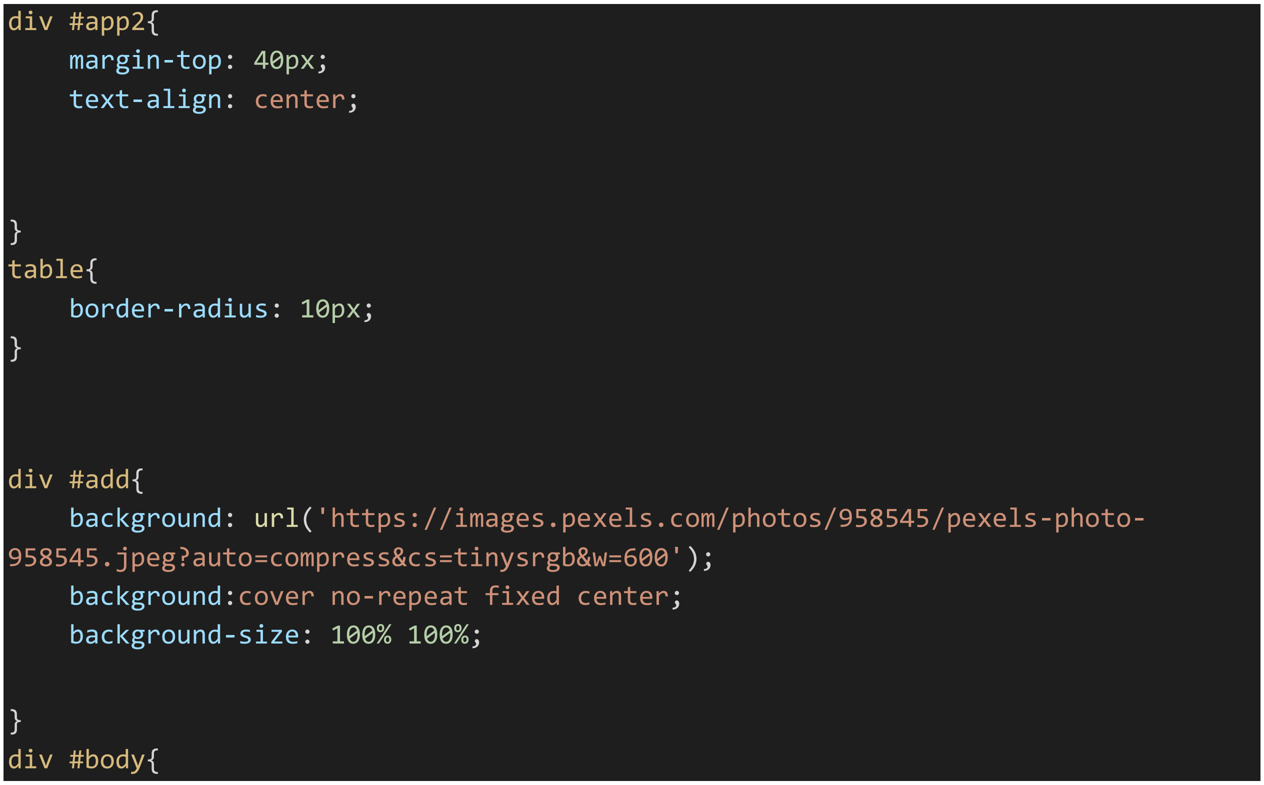
```
const SearchInput = styled.input`
  padding: 5px;
  font-size: 16px;
  border-radius: 4px;
  border: 1px solid #ccc;
  margin-right: 5px;
`;

function FoodView() {
  const [id, setId] = useState('');
  const [columns, setColumns] = useState([]);
  const [records, setRecords] = useState([]);
  const navigate = useNavigate();

  useEffect(() => {
    const fetchRecords = async () => {
      try {
        const response = await axios.get("http://localhost:1244/food/all");
        setColumns(Object.keys(response.data[0]));
        setRecords(response.data);
      } catch (err) {
        console.error("Error fetching foods records:", err);
      }
    };
    fetchRecords();
  }, []);

  let handleSubmit = (id) => {
    const conf = window.confirm("Do you want to delete this food?");
    if (conf) {
      axios
        .delete("http://localhost:1244/food/" + id)
        .then((res) => {
          alert("Record has been deleted successfully.");
          navigate("/FoodView");
          window.location.reload();
        })
        .catch((err) => console.log(err));
    }
  };

  return (
```



```
    <>
        <Navbar2/>
        <div id="body">
        <Container>
          <Title>Restaurant And Food Management System</Title>
          <div className="text-end">
        <Link to="/FoodAdd" className="btn btn-primary">
          Add New Food
        </Link><br/><br/>
        <Link to="#" className="btn btn-primary">
          Menu
        </Link><br/>
      </div>
      <br />
          <div className="text-end">
            <SearchInput type="number" id="id" placeholder="Enter Food Id"
name="fid" onChange={(e) => setId(e.target.value)}/>

          </div>
          <br />
          <StyledTable>
            <thead>
              <tr>
                {columns.map((c, i) => (
                  <TableHead key={i}>{c}</TableHead>
                ))}
                <TableHead>Action</TableHead>
              </tr>
            </thead>
            <tbody>
              {records.filter((item)=>{ return id === ''?item :
item.fid==parseInt(id)}).map((d, i) => (
                <TableRow key={i}>
                  <TableCell>{d.fid}</TableCell>
                  <TableCell>{d.foodName }</TableCell>
                  <TableCell>{d.price }</TableCell>
                  <TableCell>{d.quantity}</TableCell>
                  <TableCell>{d.totalAmount}</TableCell>
                  <TableCell>{d.restaurant.name}</TableCell>


                  <TableCell>
                    <Link to={`/FoodEdit/${d.fid}`}>
```

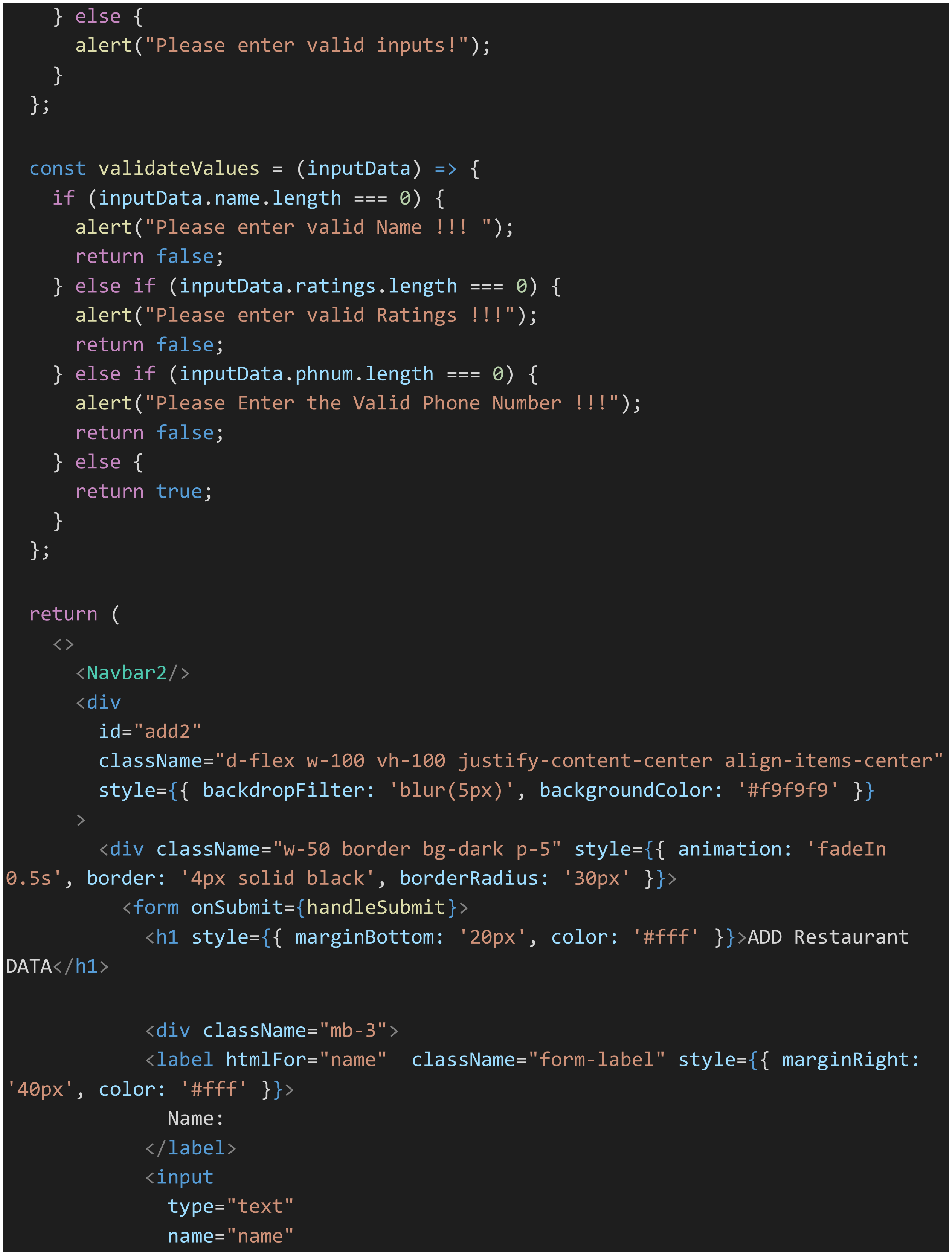


```
                      <ActionButton style={{ backgroundColor:
'#228B22' }}>Update</ActionButton>
                    </Link>
                    <button   onClick={(e) => handleSubmit(d.fid)}>
                      <ActionButton style={{ backgroundColor:
'red' }}>Delete</ActionButton>
                    </button>
                  </TableCell>
                </TableRow>
              ))}
            </tbody>
          </StyledTable>
        </Container></div>
      </>
  );
}

export default FoodView;
```

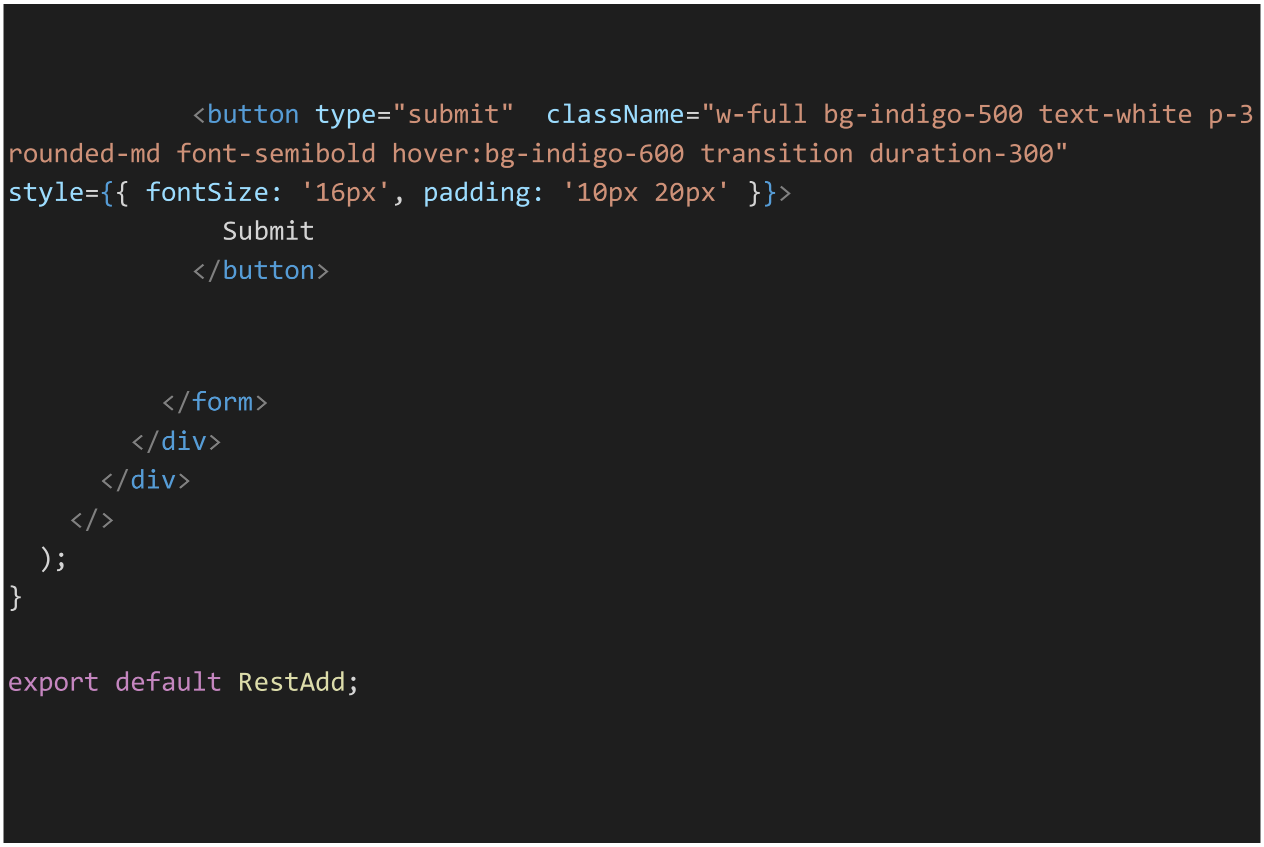


```
div #app2{
    margin-top: 40px;
    text-align: center;


}
table{
    border-radius: 10px;
}


div #add{
    background: url('https://images.pexels.com/photos/958545/pexels-photo-
958545.jpeg?auto=compress&cs=tinysrgb&w=600');
    background:cover no-repeat fixed center;
    background-size: 100% 100%;

}
div #body{
    background: url('https://images.pexels.com/photos/1076885/pexels-photo-1076885.jpeg?auto=compress&cs=tinysrgb&w=600');
    background:cover no-repeat fixed center;
    background-size: 100% 100%;

}
div #edit{

    background: url('https://images.pexels.com/photos/691114/pexels-photo-691114.jpeg?auto=compress&cs=tinysrgb&w=600');
    background:cover no-repeat fixed center;
    background-size: 100% 100%;
}
div{
    font-family: Verdana, Geneva, Tahoma, sans-serif;
}
```

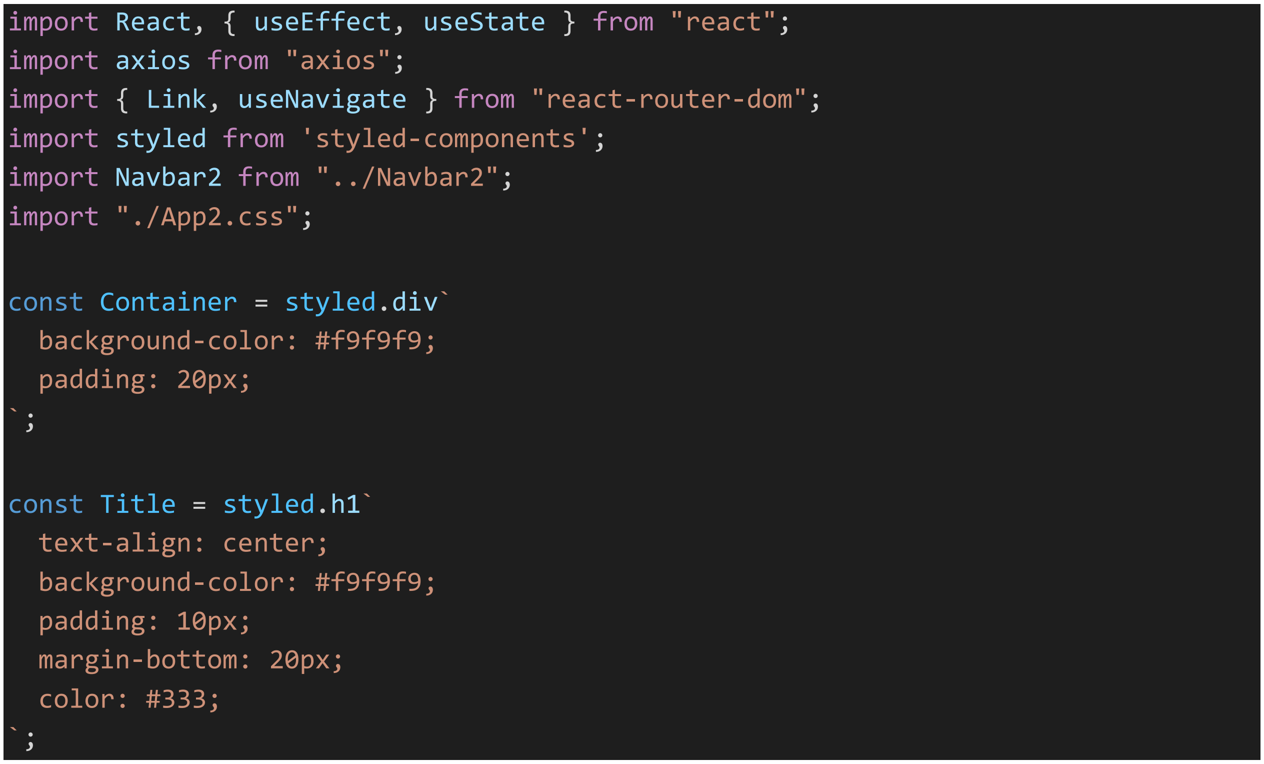
```
import axios from "axios";
import React, { useState } from "react";
import { useNavigate } from "react-router-dom";
import Navbar2 from "../Navbar2";

function RestAdd() {
  const [inputData, setInputData] = useState({
    name: "",
    ratings: "",
    phnum: "",
  });

  const navigate = useNavigate();

  const handleSubmit = async (e) => {
    e.preventDefault();
    if (validateValues(inputData)) {
      try {
        await axios.post("http://localhost:1244/restaurant", inputData);
        alert("Data added successfully");
        navigate("/Restaurant");
      } catch (err) {
        console.error("Error adding restaurant data:", err);
      }
```



```
    } else {
      alert("Please enter valid inputs!");
    }
  };

  const validateValues = (inputData) => {
    if (inputData.name.length === 0) {
      alert("Please enter valid Name !!! ");
      return false;
    } else if (inputData.ratings.length === 0) {
      alert("Please enter valid Ratings !!!");
      return false;
    } else if (inputData.phnum.length === 0) {
      alert("Please Enter the Valid Phone Number !!!");
      return false;
    } else {
      return true;
    }
  };

  return (
    <>
      <Navbar2/>
      <div
        id="add2"
        className="d-flex w-100 vh-100 justify-content-center align-items-center"
        style={{ backdropFilter: 'blur(5px)', backgroundColor: '#f9f9f9' }}
      >
        <div className="w-50 border bg-dark p-5" style={{ animation: 'fadeIn
0.5s', border: '4px solid black', borderRadius: '30px' }}>
          <form onSubmit={handleSubmit}>
            <h1 style={{ marginBottom: '20px', color: '#fff' }}>ADD Restaurant
DATA</h1>

            <div className="mb-3">
            <label htmlFor="name"  className="form-label" style={{ marginRight:
'40px', color: '#fff' }}>
              Name:
            </label>
            <input
              type="text"
              name="name"
              className="form-input mt-1 block w-full px-3 py-2 border rounded-md
shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-indigo-500
sm:text-sm"
              onChange={(e) => setInputData({ ...inputData, name:
e.target.value })}
              required
            />
          </div>

          <div className="mb-3">
            <label htmlFor="ratings" className="form-label" style={{ marginRight:
'30px', color: '#fff' }}>
              Ratings:
            </label>
            <input
              type="number"
              name="ratings"
             className="form-input mt-1 block w-full px-3 py-2 border rounded-md
shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-indigo-500
sm:text-sm"
              onChange={(e) => setInputData({ ...inputData, ratings:
e.target.value })}
              required
            />
          </div>

          <div className="mb-6">
            <label htmlFor="phnum" className="form-label" style={{ color:
'#fff' }}>
              Phone Number:
            </label>
            <input
              type="text"
              name="phnum"
              className="form-input mt-1 block w-full px-3 py-2 border rounded-md
shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-indigo-500
sm:text-sm"
              onChange={(e) => setInputData({ ...inputData, phnum:
e.target.value })}
              required
            />
          </div>
```

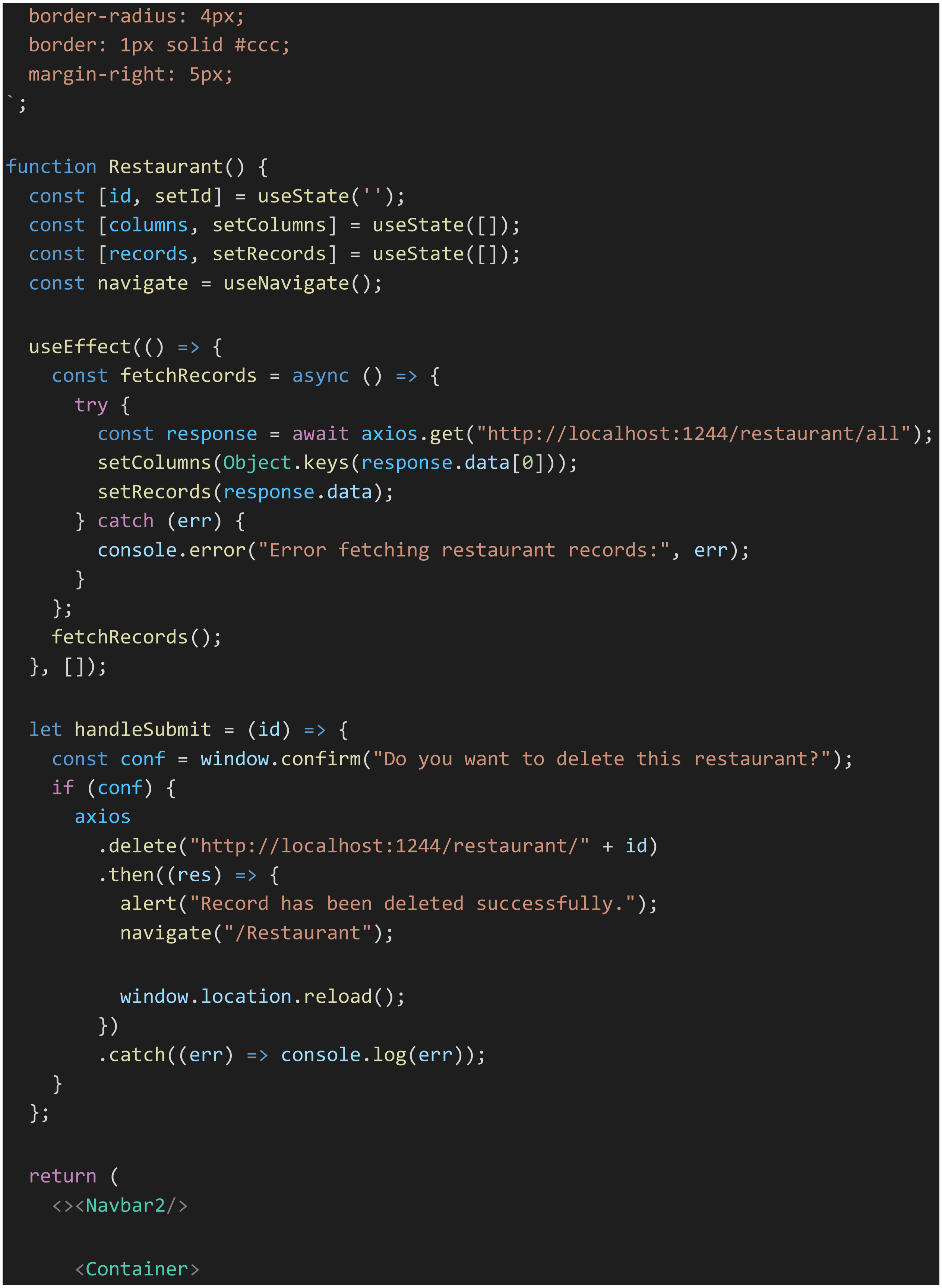


```
            <button type="submit"  className="w-full bg-indigo-500 text-white p-3
rounded-md font-semibold hover:bg-indigo-600 transition duration-300"
style={{ fontSize: '16px', padding: '10px 20px' }}>
              Submit
            </button>



          </form>
        </div>
      </div>
    </>
  );
}

export default RestAdd;
```

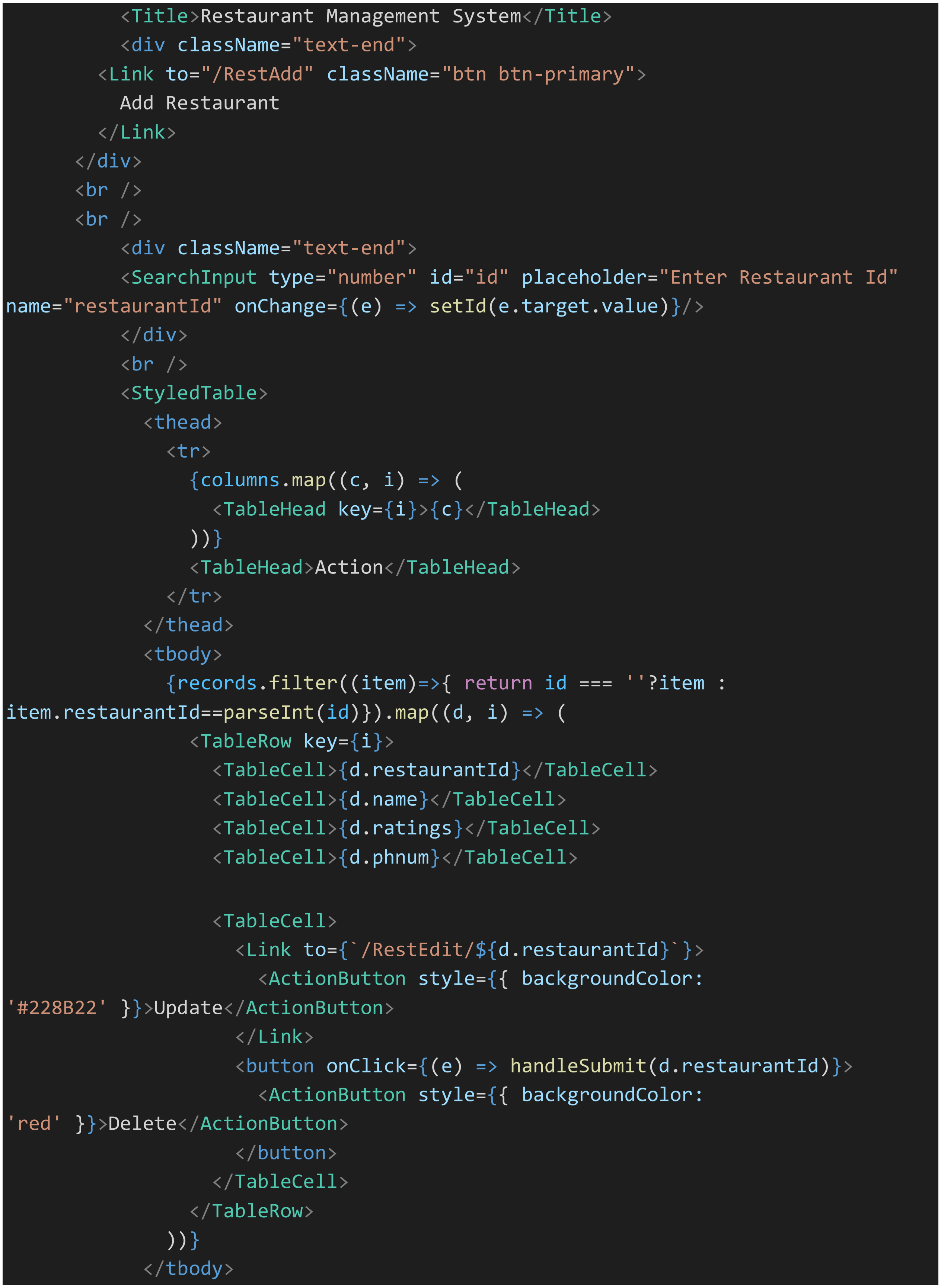


```
import React, { useEffect, useState } from "react";
import axios from "axios";
import { Link, useNavigate } from "react-router-dom";
import styled from 'styled-components';
import Navbar2 from "../Navbar2";
import "./App2.css";

const Container = styled.div`
  background-color: #f9f9f9;
  padding: 20px;
`;

const Title = styled.h1`
  text-align: center;
  background-color: #f9f9f9;
  padding: 10px;
  margin-bottom: 20px;
  color: #333;
`;
```

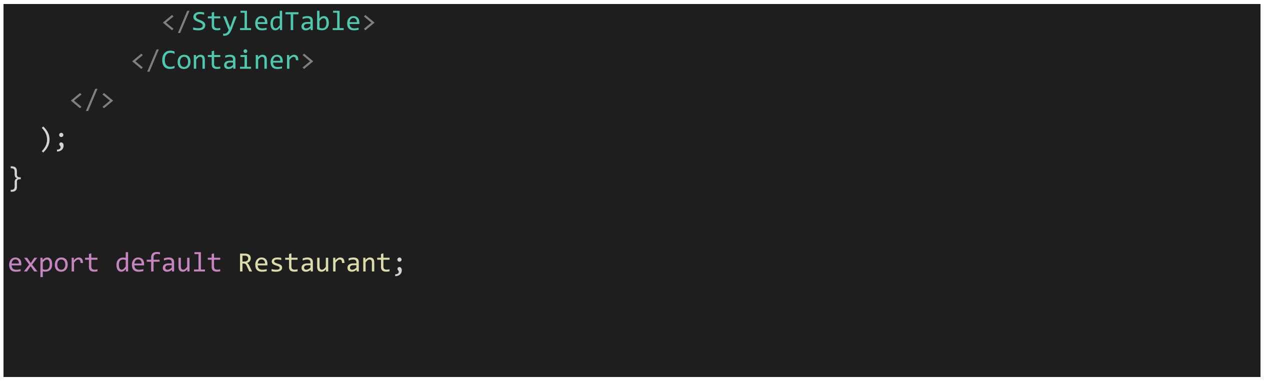
```
const StyledTable = styled.table`
  width: 100%;
  border-collapse: collapse;
  background-color: #fff;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
  border-radius: 8px;
`;

const TableHead = styled.th`
  background-color: #f9f9f9;
  color: #333;
  padding: 10px;
  text-align: left;
`;

const TableRow = styled.tr`
  &:nth-child(even) {
    background-color: #f5f5f5;
  }
`;

const TableCell = styled.td`
  padding: 10px;
`;

const ActionButton = styled.button`
  background-color: #007bff;
  color: #fff;
  padding: 5px 10px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  margin-right: 5px;

  &:hover {
    background-color: #0056b3;
  }
`;

const SearchInput = styled.input`
  padding: 5px;
  font-size: 16px;
```



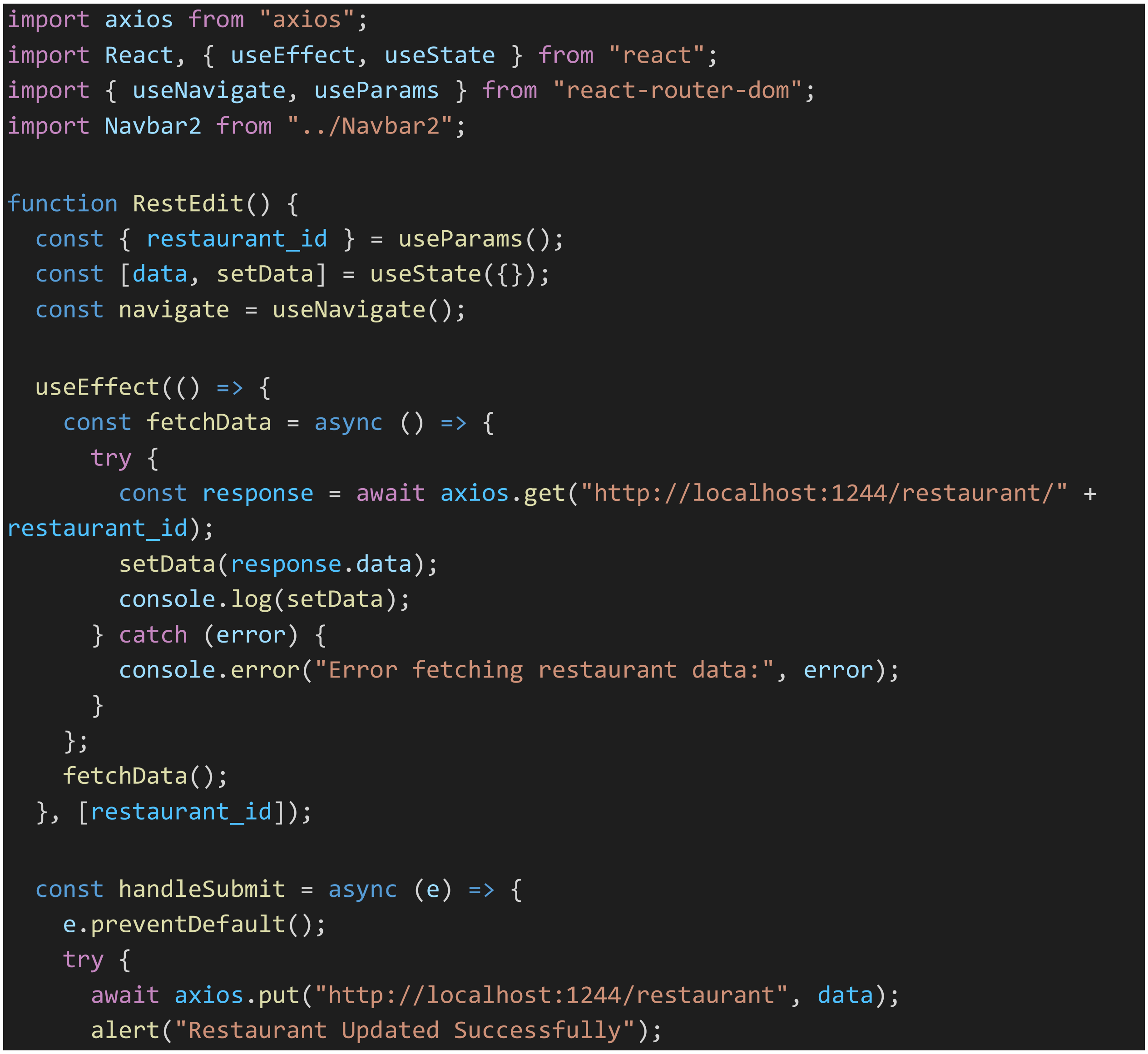
```
  border-radius: 4px;
  border: 1px solid #ccc;
  margin-right: 5px;
`;

function Restaurant() {
  const [id, setId] = useState('');
  const [columns, setColumns] = useState([]);
  const [records, setRecords] = useState([]);
  const navigate = useNavigate();

  useEffect(() => {
    const fetchRecords = async () => {
      try {
        const response = await axios.get("http://localhost:1244/restaurant/all");
        setColumns(Object.keys(response.data[0]));
        setRecords(response.data);
      } catch (err) {
        console.error("Error fetching restaurant records:", err);
      }
    };
    fetchRecords();
  }, []);

  let handleSubmit = (id) => {
    const conf = window.confirm("Do you want to delete this restaurant?");
    if (conf) {
      axios
        .delete("http://localhost:1244/restaurant/" + id)
        .then((res) => {
          alert("Record has been deleted successfully.");
          navigate("/Restaurant");

          window.location.reload();
        })
        .catch((err) => console.log(err));
    }
  };

  return (
    <><Navbar2/>

      <Container>
```



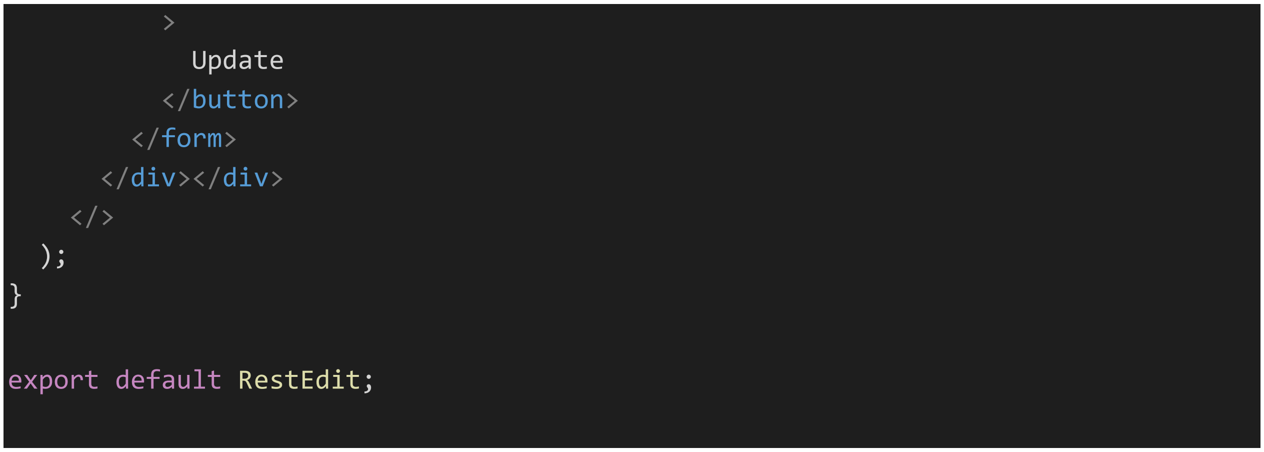
```
          <Title>Restaurant Management System</Title>
          <div className="text-end">
        <Link to="/RestAdd" className="btn btn-primary">
          Add Restaurant
        </Link>
      </div>
      <br />
      <br />
          <div className="text-end">
          <SearchInput type="number" id="id" placeholder="Enter Restaurant Id"
name="restaurantId" onChange={(e) => setId(e.target.value)}/>
          </div>
          <br />
          <StyledTable>
            <thead>
              <tr>
                {columns.map((c, i) => (
                  <TableHead key={i}>{c}</TableHead>
                ))}
                <TableHead>Action</TableHead>
              </tr>
            </thead>
            <tbody>
              {records.filter((item)=>{ return id === ''?item :
item.restaurantId==parseInt(id)}).map((d, i) => (
                <TableRow key={i}>
                  <TableCell>{d.restaurantId}</TableCell>
                  <TableCell>{d.name}</TableCell>
                  <TableCell>{d.ratings}</TableCell>
                  <TableCell>{d.phnum}</TableCell>

                  <TableCell>
                    <Link to={`/RestEdit/${d.restaurantId}`}>
                      <ActionButton style={{ backgroundColor:
'#228B22' }}>Update</ActionButton>
                    </Link>
                    <button onClick={(e) => handleSubmit(d.restaurantId)}>
                      <ActionButton style={{ backgroundColor:
'red' }}>Delete</ActionButton>
                    </button>
                  </TableCell>
                </TableRow>
              ))}
            </tbody>
```



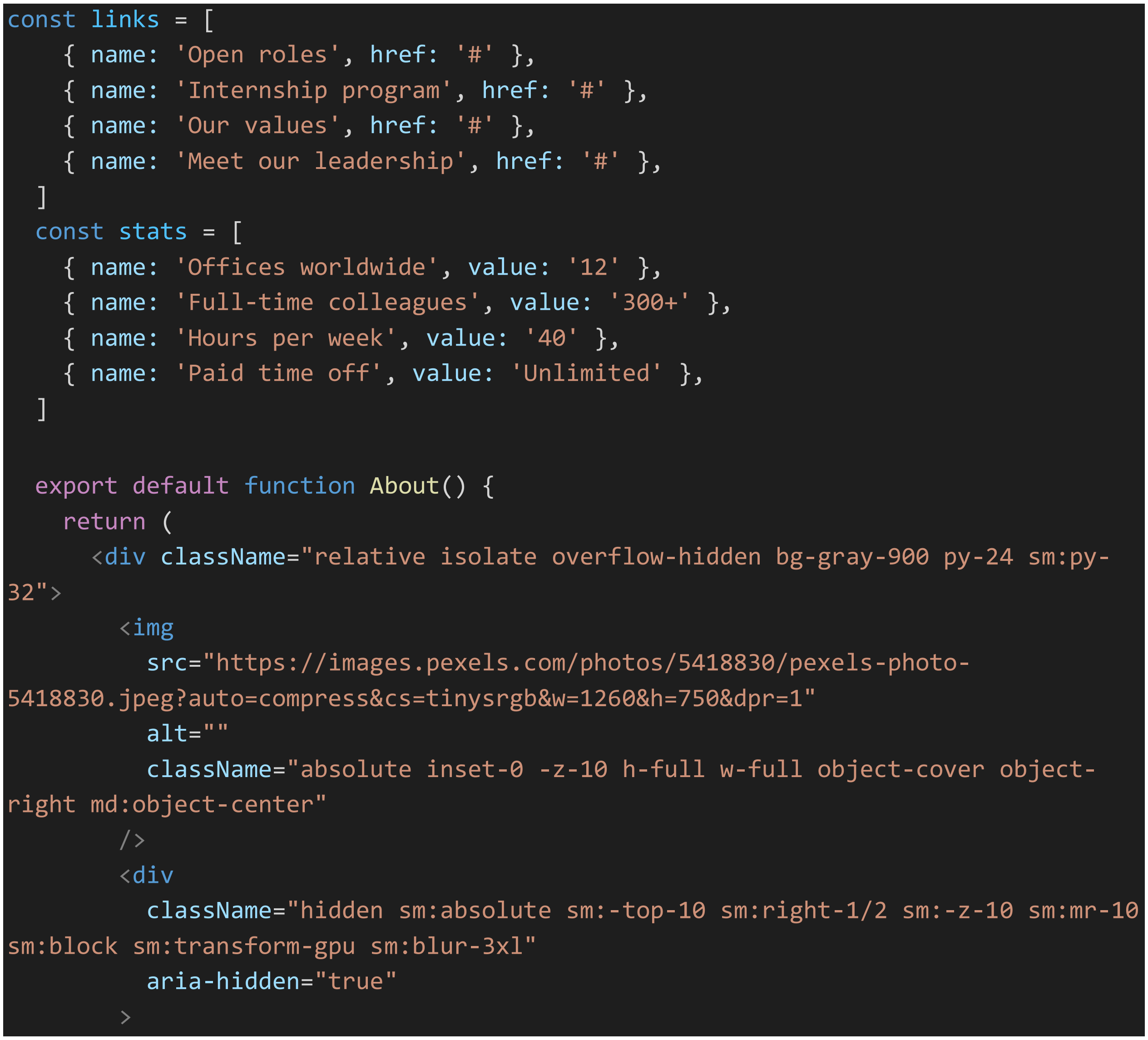
```
          </StyledTable>
        </Container>
    </>
  );
}

export default Restaurant;
```



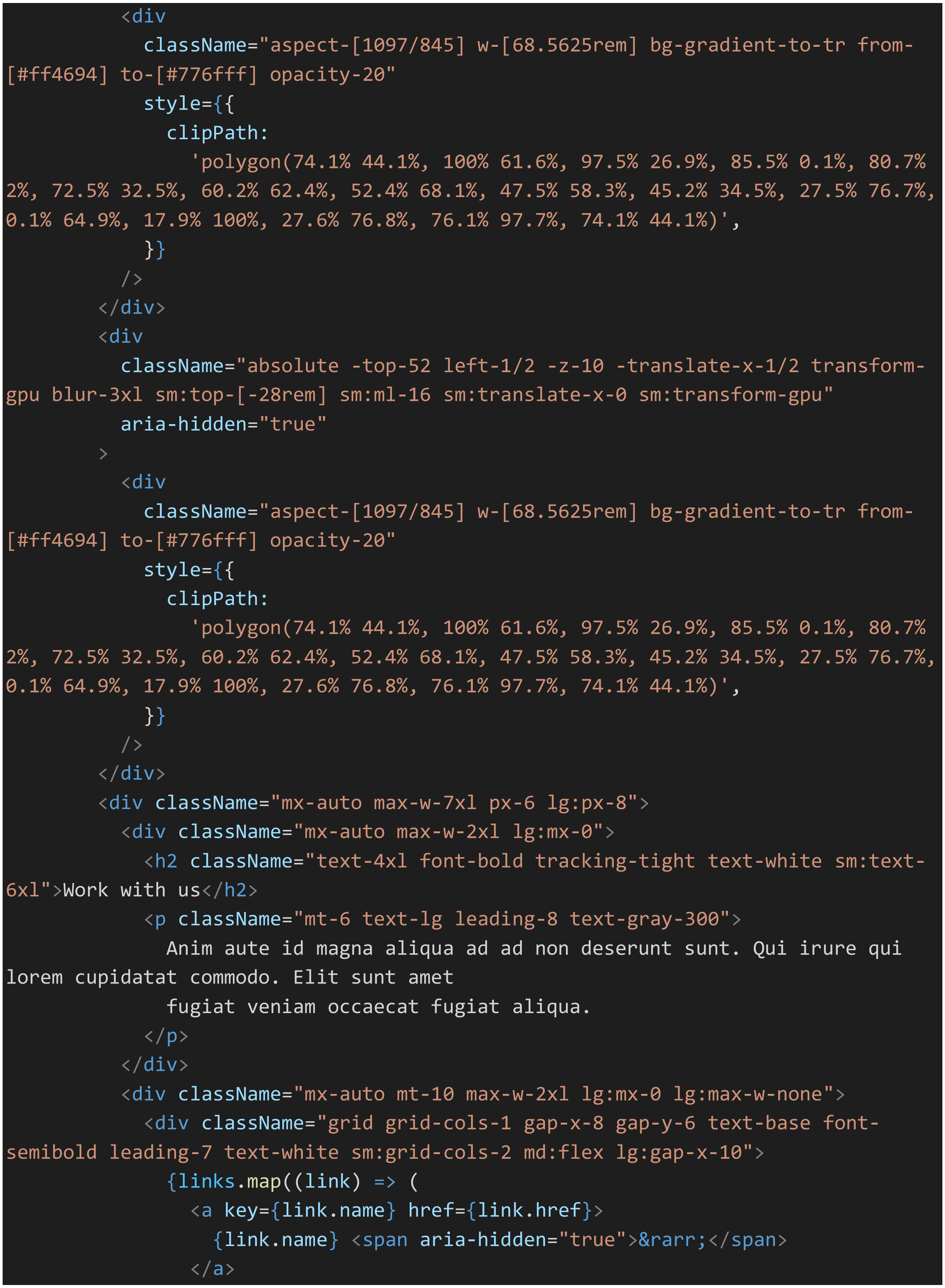
```
import axios from "axios";
import React, { useEffect, useState } from "react";
import { useNavigate, useParams } from "react-router-dom";
import Navbar2 from "../Navbar2";

function RestEdit() {
  const { restaurant_id } = useParams();
  const [data, setData] = useState({});
  const navigate = useNavigate();

  useEffect(() => {
    const fetchData = async () => {
      try {
        const response = await axios.get("http://localhost:1244/restaurant/" +
restaurant_id);
        setData(response.data);
        console.log(setData);
      } catch (error) {
        console.error("Error fetching restaurant data:", error);
      }
    };
    fetchData();
  }, [restaurant_id]);

  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      await axios.put("http://localhost:1244/restaurant", data);
      alert("Restaurant Updated Successfully");
      navigate("/Restaurant");
    } catch (error) {
      console.error("Error updating restaurant data:", error);
    }
  };

  return (
    <><Navbar2/>

      <div id="edit" className="w-80 bg-white w-100 vh-100 d-flex justify-
content-center align-items-center rounded-lg shadow-lg p-8"
style={{ backdropFilter: 'blur(5px)', backgroundColor: '#f9f9f9' }}>

      <div className="w-50 border bg-dark p-5" style={{ animation: 'fadeIn 0.5s',
border: '4px solid black', borderRadius: '30px' }}>
        <form onSubmit={handleSubmit}>
          <h1  className="text-3xl text-center font-semibold mb-8"
style={{ marginBottom: '20px', color: '#fff' }}>UPDATE RESTAURANT DATA</h1>
          <div className="mb-3">
            <label htmlFor="restaurantId" className="block text-sm font-medium
text-gray-700" style={{ marginBottom: '10px', color: '#fff' }}>
              ID:
            </label>
            <input
              type="text"
              disabled
              className="form-input mt-1 block w-full px-3 py-2 border rounded-md
shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-indigo-500
sm:text-sm"
              id="restaurantId"
              value={data.restaurantId}
              onChange={(e) => setData({ ...data, restaurantId:
e.target.value })}
            />
          </div>

          <div className="mb-3">
            <label htmlFor="name" className="block text-sm font-medium text-gray-
700" style={{ marginBottom: '10px', color: '#fff' }}>
              Name:
            </label>
            <input
              type="text"
              className="form-input mt-1 block w-full px-3 py-2 border rounded-md shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm"
              id="name"
              value={data.name || ""}
              onChange={(e) => setData({ ...data, name: e.target.value })}
            />
          </div>

          <div className="mb-3">
            <label htmlFor="ratings" className="block text-sm font-medium text-gray-700" style={{ marginBottom: '10px', color: '#fff' }}>
              Ratings:
            </label>
            <input
              type="number"
              className="form-input mt-1 block w-full px-3 py-2 border rounded-md shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm"
              id="ratings"
              value={data.ratings || ""}
              onChange={(e) => setData({ ...data, ratings: e.target.value })}
            />
          </div>

          <div className="mb-3">
            <label htmlFor="phnum" className="block text-sm font-medium text-gray-700" style={{ marginBottom: '10px', color: '#fff' }}>
              Phone Number:
            </label>
            <input
              type="text"
              className="form-input mt-1 block w-full px-3 py-2 border rounded-md shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm"
              id="phnum"
              value={data.phnum || ""}
              onChange={(e) => setData({ ...data, phnum: e.target.value })}
            />
          </div>

          <button type="submit" className="w-full bg-indigo-500 text-white p-3 rounded-md font-semibold hover:bg-indigo-600 transition duration-300"
```



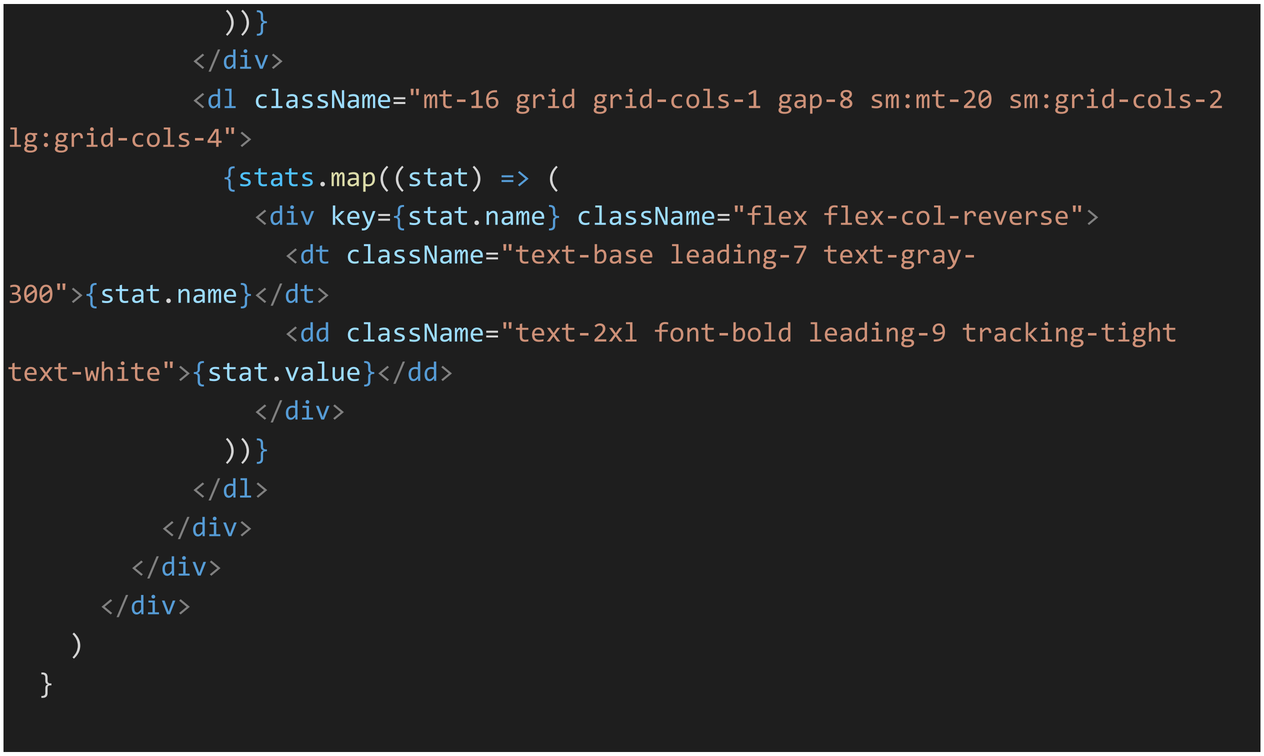
```
          >
            Update
          </button>
        </form>
      </div></div>
    </>
  );
}

export default RestEdit;
```



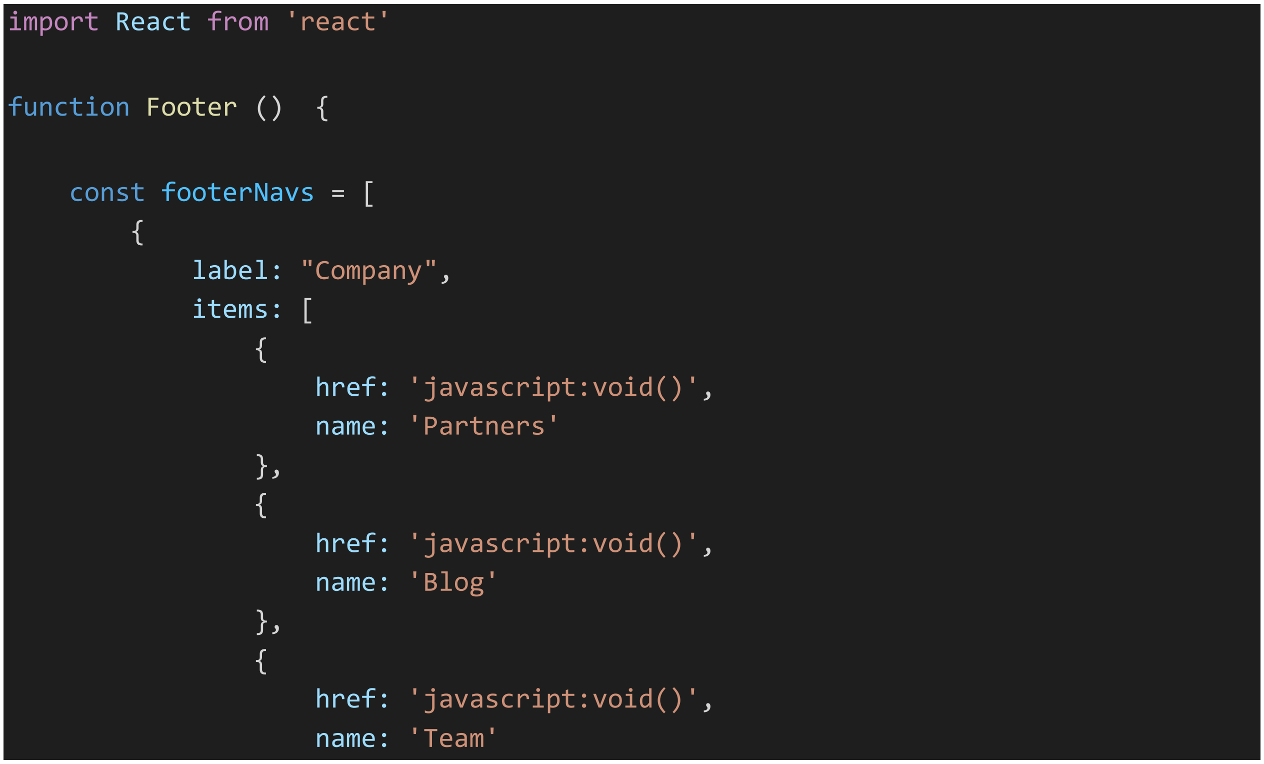
```
const links = [
    { name: 'Open roles', href: '#' },
    { name: 'Internship program', href: '#' },
    { name: 'Our values', href: '#' },
    { name: 'Meet our leadership', href: '#' },
  ]
  const stats = [
    { name: 'Offices worldwide', value: '12' },
    { name: 'Full-time colleagues', value: '300+' },
    { name: 'Hours per week', value: '40' },
    { name: 'Paid time off', value: 'Unlimited' },
  ]

  export default function About() {
    return (
      <div className="relative isolate overflow-hidden bg-gray-900 py-24 sm:py-
32">
        <img
          src="https://images.pexels.com/photos/5418830/pexels-photo-
5418830.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1"
          alt=""
          className="absolute inset-0 -z-10 h-full w-full object-cover object-
right md:object-center"
        />
        <div
          className="hidden sm:absolute sm:-top-10 sm:right-1/2 sm:-z-10 sm:mr-10
sm:block sm:transform-gpu sm:blur-3xl"
          aria-hidden="true"
        >
```



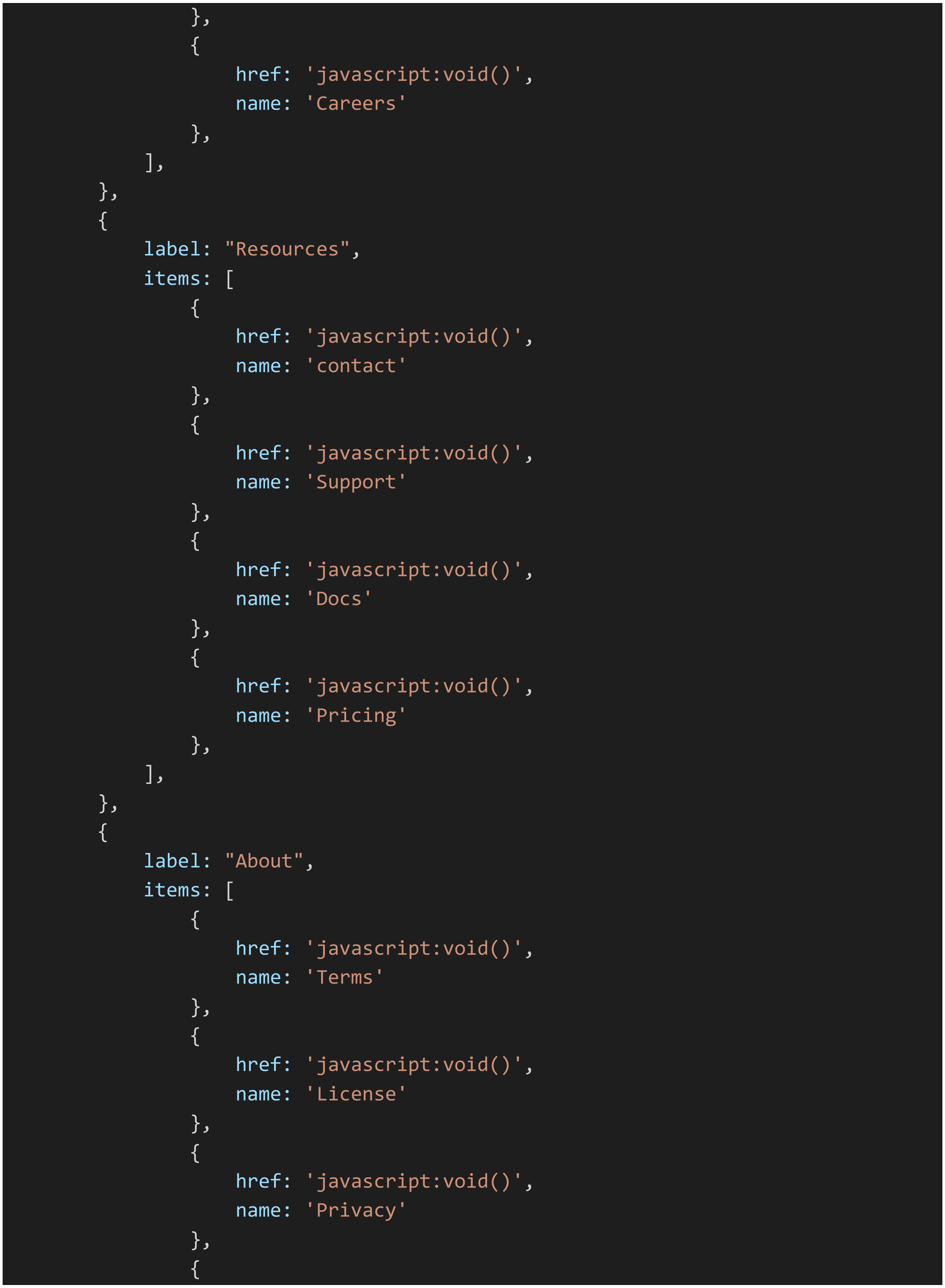
```
          <div
            className="aspect-[1097/845] w-[68.5625rem] bg-gradient-to-tr from-[#ff4694] to-[#776fff] opacity-20"
            style={{
              clipPath:
                'polygon(74.1% 44.1%, 100% 61.6%, 97.5% 26.9%, 85.5% 0.1%, 80.7% 2%, 72.5% 32.5%, 60.2% 62.4%, 52.4% 68.1%, 47.5% 58.3%, 45.2% 34.5%, 27.5% 76.7%, 0.1% 64.9%, 17.9% 100%, 27.6% 76.8%, 76.1% 97.7%, 74.1% 44.1%)',
            }}
          />
        </div>
        <div
          className="absolute -top-52 left-1/2 -z-10 -translate-x-1/2 transform-gpu blur-3xl sm:top-[-28rem] sm:ml-16 sm:translate-x-0 sm:transform-gpu"
          aria-hidden="true"
        >
          <div
            className="aspect-[1097/845] w-[68.5625rem] bg-gradient-to-tr from-[#ff4694] to-[#776fff] opacity-20"
            style={{
              clipPath:
                'polygon(74.1% 44.1%, 100% 61.6%, 97.5% 26.9%, 85.5% 0.1%, 80.7% 2%, 72.5% 32.5%, 60.2% 62.4%, 52.4% 68.1%, 47.5% 58.3%, 45.2% 34.5%, 27.5% 76.7%, 0.1% 64.9%, 17.9% 100%, 27.6% 76.8%, 76.1% 97.7%, 74.1% 44.1%)',
            }}
          />
        </div>
        <div className="mx-auto max-w-7xl px-6 lg:px-8">
          <div className="mx-auto max-w-2xl lg:mx-0">
            <h2 className="text-4xl font-bold tracking-tight text-white sm:text-6xl">Work with us</h2>
            <p className="mt-6 text-lg leading-8 text-gray-300">
              Anim aute id magna aliqua ad ad non deserunt sunt. Qui irure qui lorem cupidatat commodo. Elit sunt amet
              fugiat veniam occaecat fugiat aliqua.
            </p>
          </div>
          <div className="mx-auto mt-10 max-w-2xl lg:mx-0 lg:max-w-none">
            <div className="grid grid-cols-1 gap-x-8 gap-y-6 text-base font-semibold leading-7 text-white sm:grid-cols-2 md:flex lg:gap-x-10">
              {links.map((link) => (
                <a key={link.name} href={link.href}>
                  {link.name} <span aria-hidden="true">&rarr;</span>
                </a>
```



```
              ))}
            </div>
            <dl className="mt-16 grid grid-cols-1 gap-8 sm:mt-20 sm:grid-cols-2 lg:grid-cols-4">
              {stats.map((stat) => (
                <div key={stat.name} className="flex flex-col-reverse">
                  <dt className="text-base leading-7 text-gray-300">{stat.name}</dt>
                  <dd className="text-2xl font-bold leading-9 tracking-tight text-white">{stat.value}</dd>
                </div>
              ))}
            </dl>
          </div>
        </div>
      </div>
    )
  }
```



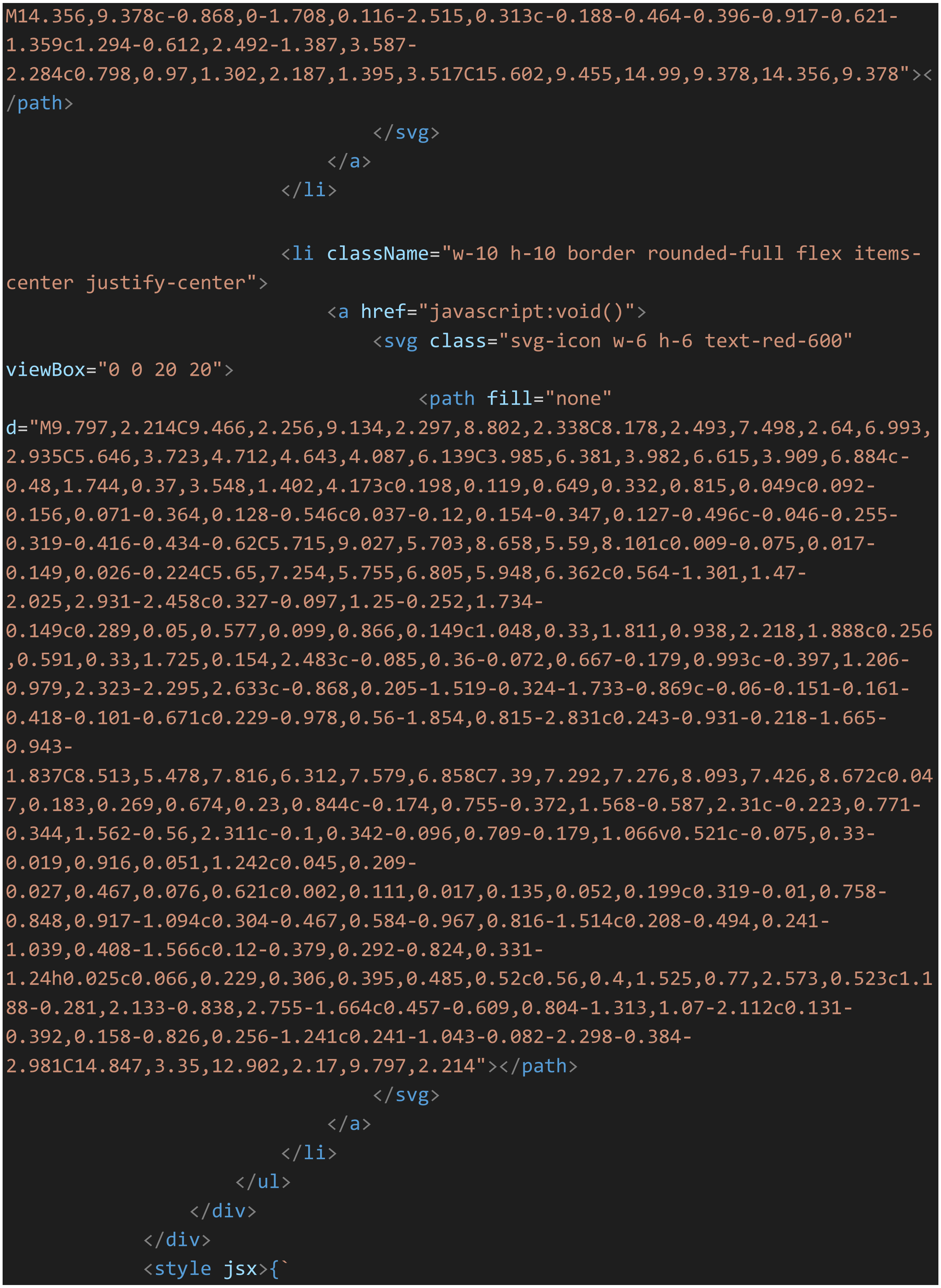
```
import React from 'react'

function Footer ()  {

    const footerNavs = [
        {
            label: "Company",
            items: [
                {
                    href: 'javascript:void()',
                    name: 'Partners'
                },
                {
                    href: 'javascript:void()',
                    name: 'Blog'
                },
                {
                    href: 'javascript:void()',
                    name: 'Team'
```



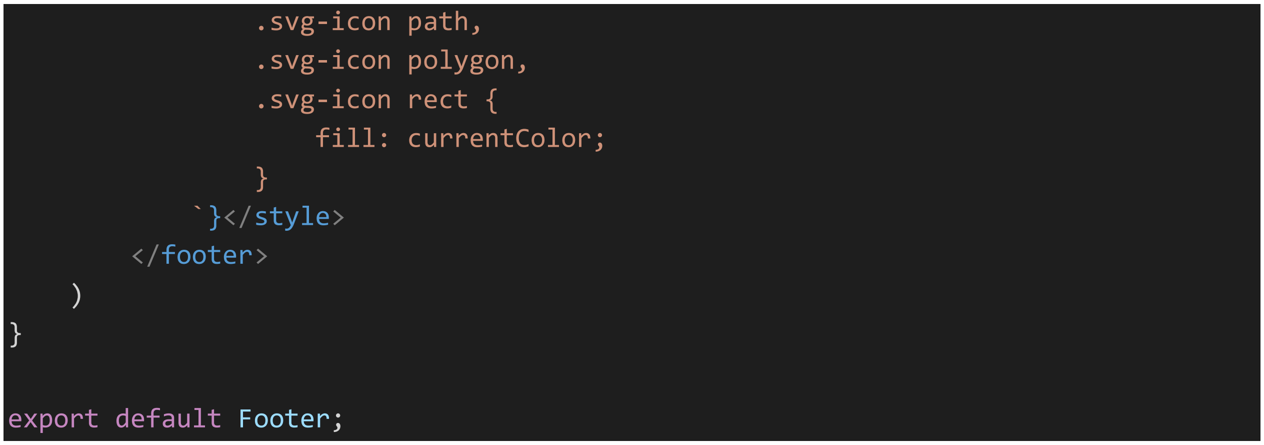
```
                },
                {
                    href: 'javascript:void()',
                    name: 'Careers'
                },
            ],
        },
        {
            label: "Resources",
            items: [
                {
                    href: 'javascript:void()',
                    name: 'contact'
                },
                {
                    href: 'javascript:void()',
                    name: 'Support'
                },
                {
                    href: 'javascript:void()',
                    name: 'Docs'
                },
                {
                    href: 'javascript:void()',
                    name: 'Pricing'
                },
            ],
        },
        {
            label: "About",
            items: [
                {
                    href: 'javascript:void()',
                    name: 'Terms'
                },
                {
                    href: 'javascript:void()',
                    name: 'License'
                },
                {
                    href: 'javascript:void()',
                    name: 'Privacy'
                },
                {
                    href: 'javascript:void()',
                    name: 'About US'
                },
            ]
        }
    ]

    return (
        <footer className="text-gray-500 bg-white px-4 py-5 max-w-screen-xl mx-
auto md:px-8">
            <div className="gap-6 justify-between md:flex">
                <div className="flex-1">
                    <div className="max-w-xs">
                        <img
src={"https://cdn2.iconfinder.com/data/icons/international-food/64/fried_rice-
256.png"} className="w-32" />
                        <p className="leading-relaxed mt-2 text-[15px]">
                            We were love to hear from You
                        </p>
                    </div>
                    <form
                        onSubmit={(e) => e.preventDefault()}
                    >
                        <label className="block pt-4 pb-2">
                            Stay up to date
                        </label>
                        <div className="max-w-sm flex items-center border
rounded-md p-1">
                            <input
                                type="email"
                                placeholder="Enter your email"
                                className="w-full p-2.5 outline-none"
                            />
                            <button
                                className="p-2.5 rounded-md text-white bg-indigo-
600 outline-none shadow-md focus:shadow-none sm:px-5"
                            >
                                Subscribe
                            </button>
                        </div>
                    </form>
                </div>
                <div className="flex-1 mt-10 space-y-6 items-center justify-
between sm:flex md:space-y-0 md:mt-0">
```

```
                    {
                        footerNavs.map((item, idx) => (
                            <ul
                                className="space-y-4"
                                key={idx}
                            >
                                <h4 className="text-gray-800 font-medium">
                                    { item.label }
                                </h4>
                                {
                                    item.items.map(((el, idx) => (
                                        <li key={idx}>
                                            <a
                                                href={el.href}
                                                className="hover:underline
hover:text-indigo-600"

                                            >
                                                { el.name }
                                            </a>
                                        </li>
                                    )))
                                }
                            </ul>
                        ))
                    }
                </div>
            </div>
            <div className="mt-8 py-6 border-t items-center justify-between
sm:flex">
                <div className="mt-4 sm:mt-0">
                    &copy; 2024 All Copyrights rights reserved.
                </div>
                <div className="mt-6 sm:mt-0">
                    <ul className="flex items-center space-x-4">
                        <li className="w-10 h-10 border rounded-full flex items-
center justify-center">
                            <a href="javascript:void()">
                                <svg class="svg-icon w-6 h-6 text-blue-400"
viewBox="0 0 20 20">
                                    <path fill="none" d="M18.258,3.266c-
0.693,0.405-1.46,0.698-2.277,0.857c-0.653-0.686-1.586-1.115-2.618-1.115c-1.98,0-
3.586,1.581-
3.586,3.53c0,0.276,0.031,0.545,0.092,0.805C6.888,7.195,4.245,5.79,2.476,3.654C2.1
```

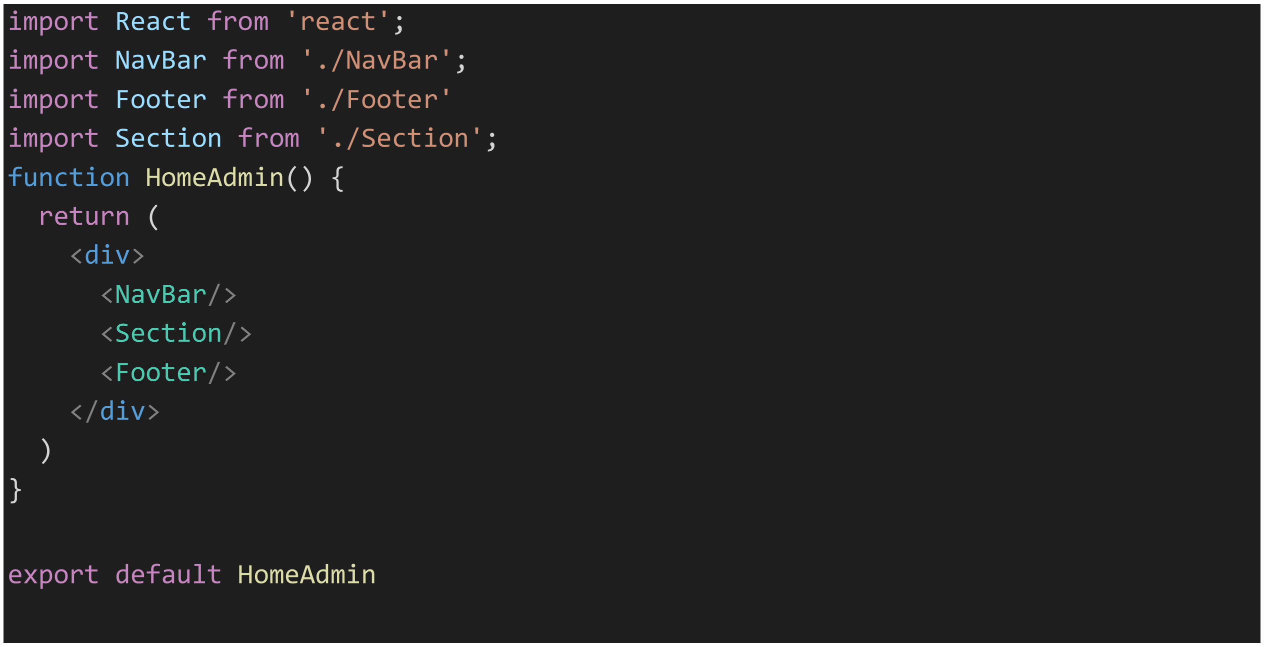
```
67,4.176,1.99,4.781,1.99,5.429c0,1.224,0.633,2.305,1.596,2.938C2.999,8.349,2.445,
8.19,1.961,7.925C1.96,7.94,1.96,7.954,1.96,7.97c0,1.71,1.237,3.138,2.877,3.462c-
0.301,0.08-0.617,0.123-0.945,0.123c-0.23,0-0.456-0.021-0.674-
0.062c0.456,1.402,1.781,2.422,3.35,2.451c-1.228,0.947-2.773,1.512-4.454,1.512c-
0.291,0-0.575-0.016-0.855-0.049c1.588,1,3.473,1.586,5.498,1.586c6.598,0,10.205-
5.379,10.205-10.045c0-0.153-0.003-0.305-0.01-0.456c0.7-0.499,1.308-1.12,1.789-
1.827c-0.644,0.28-1.334,0.469-
2.06,0.555C17.422,4.782,17.99,4.091,18.258,3.266"></path>
                                </svg>
                            </a>
                        </li>

                        <li className="w-10 h-10 border rounded-full flex items-
center justify-center">
                            <a href="javascript:void()">
                                <svg class="svg-icon w-6 h-6 text-blue-700"
viewBox="0 0 20 20">
                                    <path fill="none" d="M11.344,5.71c0-
0.73,0.074-1.122,1.199-1.122h1.502V1.871h-2.404c-2.886,0-3.903,1.36-
3.903,3.646v1.765h-1.8V10h1.8v8.128h3.601V10h2.403l0.32-2.718h-
2.724L11.344,5.71z"></path>
                                </svg>
                            </a>
                        </li>

                        <li className="w-10 h-10 border rounded-full flex items-
center justify-center">
                            <a href="javascript:void()">
                                <svg class="svg-icon w-6 h-6 text-blue-500"
viewBox="0 0 20 20">
                                    <path fill="none" d="M10,2.531c-4.125,0-
7.469,3.344-7.469,7.469c0,4.125,3.344,7.469,7.469,7.469c4.125,0,7.469-
3.344,7.469-7.469C17.469,5.875,14.125,2.531,10,2.531
M10,3.776c1.48,0,2.84,0.519,3.908,1.384c-1.009,0.811-2.111,1.512-
3.298,2.066C9.914,6.072,9.077,5.017,8.14,4.059C8.728,3.876,9.352,3.776,10,3.776
M6.903,4.606c0.962,0.93,1.82,1.969,2.53,3.112C7.707,8.364,5.849,8.734,3.902,8.75C
4.264,6.976,5.382,5.481,6.903,4.606 M3.776,10c2.219,0,4.338-0.418,6.29-
1.175c0.209,0.404,0.405,0.813,0.579,1.236c-2.147,0.805-3.953,2.294-
5.177,4.195C4.421,13.143,3.776,11.648,3.776,10 M10,16.224c-1.337,0-2.572-0.426-
3.586-1.143c1.079-1.748,2.709-3.119,4.659-
3.853c0.483,1.488,0.755,3.071,0.784,4.714C11.271,16.125,10.646,16.224,10,16.224
M13.075,15.407c-0.072-1.577-0.342-3.103-0.806-4.542c0.673-0.154,1.369-
0.243,2.087-
0.243c0.621,0,1.22,0.085,1.807,0.203C15.902,12.791,14.728,14.465,13.075,15.407
```



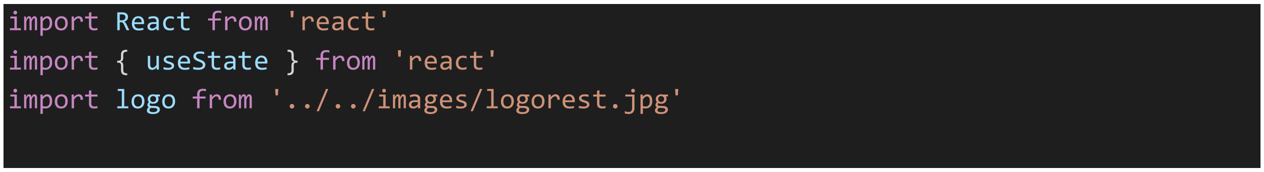
```
M14.356,9.378c-0.868,0-1.708,0.116-2.515,0.313c-0.188-0.464-0.396-0.917-0.621-
1.359c1.294-0.612,2.492-1.387,3.587-
2.284c0.798,0.97,1.302,2.187,1.395,3.517C15.602,9.455,14.99,9.378,14.356,9.378"><
/path>
                                </svg>
                            </a>
                        </li>

                        <li className="w-10 h-10 border rounded-full flex items-
center justify-center">
                            <a href="javascript:void()">
                                <svg class="svg-icon w-6 h-6 text-red-600"
viewBox="0 0 20 20">
                                    <path fill="none"
d="M9.797,2.214C9.466,2.256,9.134,2.297,8.802,2.338C8.178,2.493,7.498,2.64,6.993,
2.935C5.646,3.723,4.712,4.643,4.087,6.139C3.985,6.381,3.982,6.615,3.909,6.884c-
0.48,1.744,0.37,3.548,1.402,4.173c0.198,0.119,0.649,0.332,0.815,0.049c0.092-
0.156,0.071-0.364,0.128-0.546c0.037-0.12,0.154-0.347,0.127-0.496c-0.046-0.255-
0.319-0.416-0.434-0.62C5.715,9.027,5.703,8.658,5.59,8.101c0.009-0.075,0.017-
0.149,0.026-0.224C5.65,7.254,5.755,6.805,5.948,6.362c0.564-1.301,1.47-
2.025,2.931-2.458c0.327-0.097,1.25-0.252,1.734-
0.149c0.289,0.05,0.577,0.099,0.866,0.149c1.048,0.33,1.811,0.938,2.218,1.888c0.256
,0.591,0.33,1.725,0.154,2.483c-0.085,0.36-0.072,0.667-0.179,0.993c-0.397,1.206-
0.979,2.323-2.295,2.633c-0.868,0.205-1.519-0.324-1.733-0.869c-0.06-0.151-0.161-
0.418-0.101-0.671c0.229-0.978,0.56-1.854,0.815-2.831c0.243-0.931-0.218-1.665-
0.943-
1.837C8.513,5.478,7.816,6.312,7.579,6.858C7.39,7.292,7.276,8.093,7.426,8.672c0.04
7,0.183,0.269,0.674,0.23,0.844c-0.174,0.755-0.372,1.568-0.587,2.31c-0.223,0.771-
0.344,1.562-0.56,2.311c-0.1,0.342-0.096,0.709-0.179,1.066v0.521c-0.075,0.33-
0.019,0.916,0.051,1.242c0.045,0.209-
0.027,0.467,0.076,0.621c0.002,0.111,0.017,0.135,0.052,0.199c0.319-0.01,0.758-
0.848,0.917-1.094c0.304-0.467,0.584-0.967,0.816-1.514c0.208-0.494,0.241-
1.039,0.408-1.566c0.12-0.379,0.292-0.824,0.331-
1.24h0.025c0.066,0.229,0.306,0.395,0.485,0.52c0.56,0.4,1.525,0.77,2.573,0.523c1.1
88-0.281,2.133-0.838,2.755-1.664c0.457-0.609,0.804-1.313,1.07-2.112c0.131-
0.392,0.158-0.826,0.256-1.241c0.241-1.043-0.082-2.298-0.384-
2.981C14.847,3.35,12.902,2.17,9.797,2.214"></path>
                                </svg>
                            </a>
                        </li>
                    </ul>
                </div>
            </div>
            <style jsx>{`
```



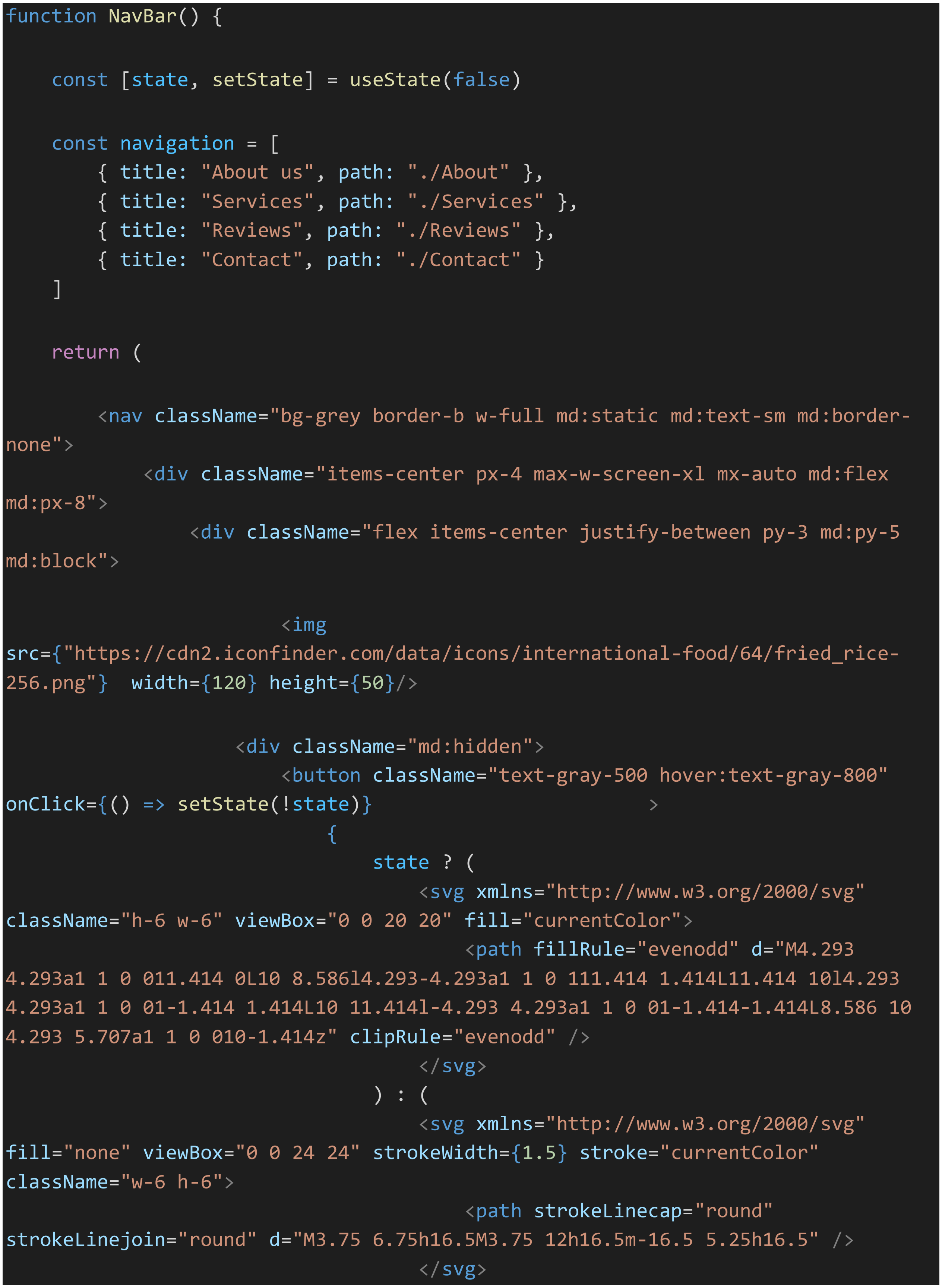
```
                .svg-icon path,
                .svg-icon polygon,
                .svg-icon rect {
                    fill: currentColor;
                }
            `}</style>
        </footer>
    )
}

export default Footer;
```



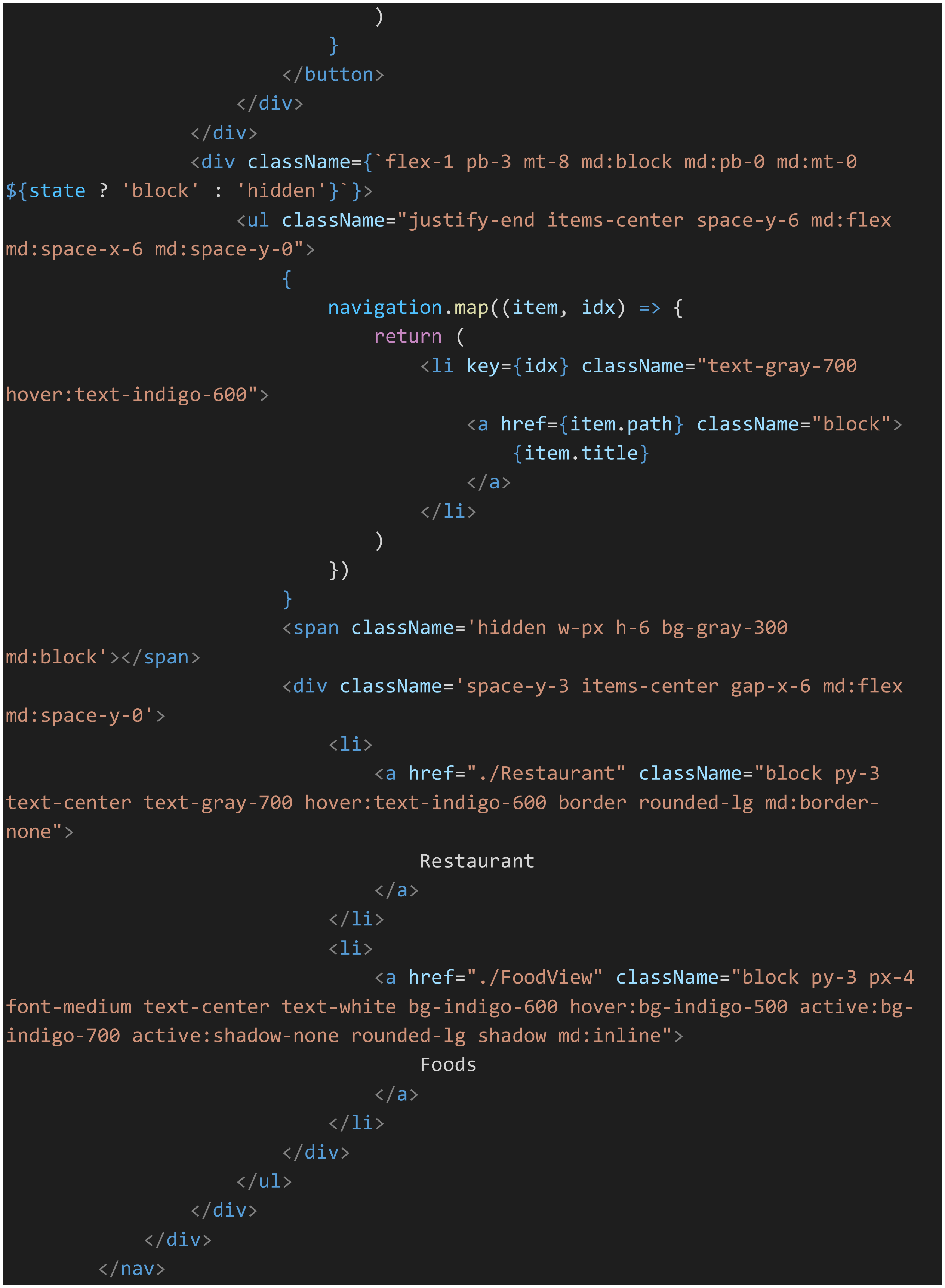
```
import React from 'react';
import NavBar from './NavBar';
import Footer from './Footer'
import Section from './Section';
function HomeAdmin() {
  return (
    <div>
      <NavBar/>
      <Section/>
      <Footer/>
    </div>
  )
}

export default HomeAdmin
```



```
import React from 'react'
import { useState } from 'react'
import logo from '../../images/logorest.jpg'
```



```
function NavBar() {

    const [state, setState] = useState(false)

    const navigation = [
        { title: "About us", path: "./About" },
        { title: "Services", path: "./Services" },
        { title: "Reviews", path: "./Reviews" },
        { title: "Contact", path: "./Contact" }
    ]

    return (

        <nav className="bg-grey border-b w-full md:static md:text-sm md:border-
none">
            <div className="items-center px-4 max-w-screen-xl mx-auto md:flex
md:px-8">
                <div className="flex items-center justify-between py-3 md:py-5
md:block">

                        <img
src={"https://cdn2.iconfinder.com/data/icons/international-food/64/fried_rice-
256.png"}  width={120} height={50}/>

                    <div className="md:hidden">
                        <button className="text-gray-500 hover:text-gray-800"
onClick={() => setState(!state)}                        >
                            {
                                state ? (
                                    <svg xmlns="http://www.w3.org/2000/svg"
className="h-6 w-6" viewBox="0 0 20 20" fill="currentColor">
                                        <path fillRule="evenodd" d="M4.293
4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293
4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10
4.293 5.707a1 1 0 010-1.414z" clipRule="evenodd" />
                                    </svg>
                                ) : (
                                    <svg xmlns="http://www.w3.org/2000/svg"
fill="none" viewBox="0 0 24 24" strokeWidth={1.5} stroke="currentColor"
className="w-6 h-6">
                                        <path strokeLinecap="round"
strokeLinejoin="round" d="M3.75 6.75h16.5M3.75 12h16.5m-16.5 5.25h16.5" />
                                    </svg>
```



```
                                )
                            }
                        </button>
                    </div>
                </div>
                <div className={`flex-1 pb-3 mt-8 md:block md:pb-0 md:mt-0
${state ? 'block' : 'hidden'}`}>
                    <ul className="justify-end items-center space-y-6 md:flex
md:space-x-6 md:space-y-0">
                        {
                            navigation.map((item, idx) => {
                                return (
                                    <li key={idx} className="text-gray-700
hover:text-indigo-600">
                                        <a href={item.path} className="block">
                                            {item.title}
                                        </a>
                                    </li>
                                )
                            })
                        }
                        <span className='hidden w-px h-6 bg-gray-300
md:block'></span>
                        <div className='space-y-3 items-center gap-x-6 md:flex
md:space-y-0'>
                            <li>
                                <a href="./Restaurant" className="block py-3
text-center text-gray-700 hover:text-indigo-600 border rounded-lg md:border-
none">
                                    Restaurant
                                </a>
                            </li>
                            <li>
                                <a href="./FoodView" className="block py-3 px-4
font-medium text-center text-white bg-indigo-600 hover:bg-indigo-500 active:bg-
indigo-700 active:shadow-none rounded-lg shadow md:inline">
                                    Foods
                                </a>
                            </li>
                        </div>
                    </ul>
                </div>
            </div>
        </nav>
```

```
    )
}

export default NavBar
```

```
import {
    Disclosure,
    DisclosureButton,
    DisclosurePanel,
    Menu,
    MenuButton,
    MenuItem,
    MenuItems,
    Transition,
  } from '@headlessui/react'
  import { Bars3Icon, BellIcon, XMarkIcon } from '@heroicons/react/24/outline'

  const navigation = [
    { name: 'Home', href: '/Admin', current: false },
    { name: 'RestaurantAdder', href: '/RestAdd', current: false },
    { name: 'FoodAdder', href: '/FoodAdd', current: false },

    { name: 'RestaurantDetails', href: '/Restaurant', current: false },
    { name: 'FoodDetails', href: '/FoodView', current: false },



  ]

  function classNames(...classes) {
    return classes.filter(Boolean).join(' ')
  }

  export default function Navbar2() {
    return (
      <Disclosure as="nav" className="bg-gray-800">
        {({ open }) => (
          <>
            <div className="mx-auto max-w-7xl px-2 sm:px-6 lg:px-8">
              <div className="relative flex h-16 items-center justify-between">
                <div className="absolute inset-y-0 left-0 flex items-center sm:hidden">
                  {/* Mobile menu button*/}
                  <DisclosureButton className="relative inline-flex items-center justify-center rounded-md p-2 text-gray-400 hover:bg-gray-700 hover:text-white focus:outline-none focus:ring-2 focus:ring-inset focus:ring-white">
                    <span className="absolute -inset-0.5" />
                    <span className="sr-only">Open main menu</span>
                    {open ? (
                      <XMarkIcon className="block h-6 w-6" aria-hidden="true" />
                    ) : (
                      <Bars3Icon className="block h-6 w-6" aria-hidden="true" />
                    )}
                  </DisclosureButton>
                </div>
                <div className="flex flex-1 items-center justify-center sm:items-stretch sm:justify-start">
                  <div className="flex flex-shrink-0 items-center">
                    <img
                      className="h-8 w-auto"
                      src="https://cdn2.iconfinder.com/data/icons/international-food/64/fried_rice-256.png"
                      alt="Your Company"
                    />
                  </div>
                  <div className="hidden sm:ml-6 sm:block">
                    <div className="flex space-x-4">
                      {navigation.map((item) => (
                        <a
                          key={item.name}
                          href={item.href}
                          className={classNames(
                            item.current ? 'bg-gray-900 text-white' : 'text-gray-300 hover:bg-gray-700 hover:text-white',
                            'rounded-md px-3 py-2 text-sm font-medium',
                          )}
                          aria-current={item.current ? 'page' : undefined}
                        >
                          {item.name}
                        </a>
                      ))}
                    </div>
                  </div>
                </div>
                <div className="absolute inset-y-0 right-0 flex items-center pr-2 sm:static sm:inset-auto sm:ml-6 sm:pr-0">
                  <button
                    type="button"
                    className="relative rounded-full bg-gray-800 p-1 text-gray-400 hover:text-white focus:outline-none focus:ring-2 focus:ring-white focus:ring-offset-2 focus:ring-offset-gray-800"
                  >
                    <span className="absolute -inset-1.5" />
                    <span className="sr-only">View notifications</span>
                    <BellIcon className="h-6 w-6" aria-hidden="true" />
                  </button>

                  {/* Profile dropdown */}
                  <Menu as="div" className="relative ml-3">
                    <div>
                      <MenuButton className="relative flex rounded-full bg-gray-800 text-sm focus:outline-none focus:ring-2 focus:ring-white focus:ring-offset-2 focus:ring-offset-gray-800">
                        <span className="absolute -inset-1.5" />
                        <span className="sr-only">Open user menu</span>
                        <img
                          className="h-8 w-8 rounded-full"
                          src="https://images.pexels.com/photos/3613388/pexels-photo-3613388.jpeg?auto=compress&cs=tinysrgb&w=600"
                          alt=""
                        />
                      </MenuButton>
                    </div>
                    <Transition
                      enter="transition ease-out duration-100"
                      enterFrom="transform opacity-0 scale-95"
                      enterTo="transform opacity-100 scale-100"
                      leave="transition ease-in duration-75"
                      leaveFrom="transform opacity-100 scale-100"
                      leaveTo="transform opacity-0 scale-95"
                    >
                      <MenuItems className="absolute right-0 z-10 mt-2 w-48 origin-top-right rounded-md bg-white py-1 shadow-lg ring-1 ring-black ring-opacity-5 focus:outline-none">
                        <MenuItem>
```

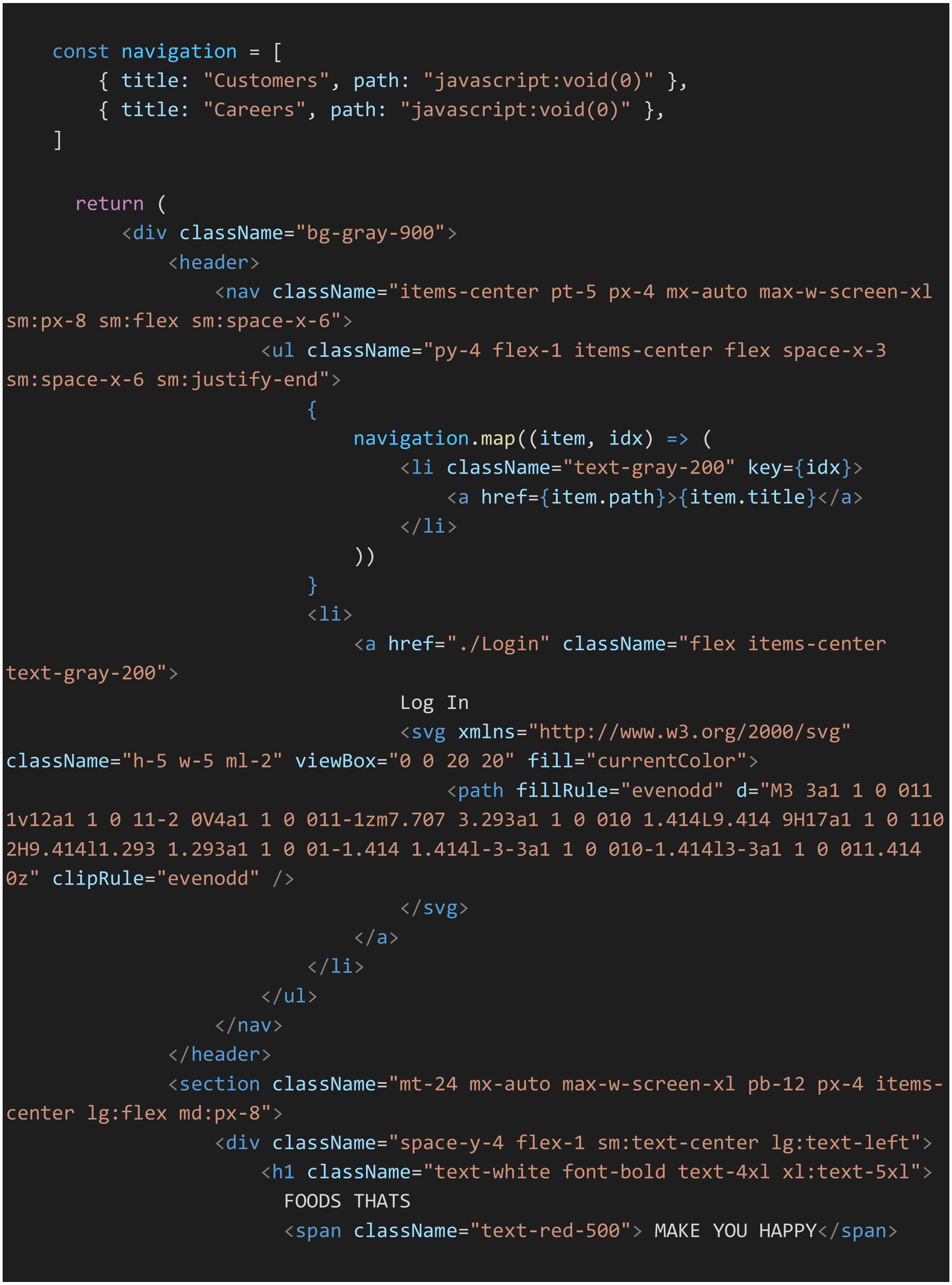
```
                          {({ focus }) => (
                            <a
                              href="#"
                              className={classNames(focus ? 'bg-gray-100' : '', 'block px-4 py-2 text-sm text-gray-700')}
                            >
                              Your Profile
                            </a>
                          )}
                        </MenuItem>
                        <MenuItem>
                          {({ focus }) => (
                            <a
                              href="#"
                              className={classNames(focus ? 'bg-gray-100' : '', 'block px-4 py-2 text-sm text-gray-700')}
                            >
                              Settings
                            </a>
                          )}
                        </MenuItem>
                        <MenuItem>
                          {({ focus }) => (
                            <a
                              href="#"
                              className={classNames(focus ? 'bg-gray-100' : '', 'block px-4 py-2 text-sm text-gray-700')}
                            >
                              Sign out
                            </a>
                          )}
                        </MenuItem>
                      </MenuItems>
                    </Transition>
                  </Menu>
                </div>
              </div>
            </div>

            <DisclosurePanel className="sm:hidden">
              <div className="space-y-1 px-2 pb-3 pt-2">
                {navigation.map((item) => (
                  <DisclosureButton
                    key={item.name}
                    as="a"
                    href={item.href}
                    className={classNames(
                      item.current ? 'bg-gray-900 text-white' : 'text-gray-300 
hover:bg-gray-700 hover:text-white',
                      'block rounded-md px-3 py-2 text-base font-medium',
                    )}
                    aria-current={item.current ? 'page' : undefined}
                  >
                    {item.name}
                  </DisclosureButton>
                ))}
              </div>
            </DisclosurePanel>
          </>
        )}
      </Disclosure>
    )
  }
```

```
import React from 'react';

function Section() {
  return (
    <div style={{
      backgroundImage: `url('https://wallpapers.com/images/hd/food-4k-
m37wpodzrcbv5gvw.jpg')`,
      backgroundSize: 'cover',
      height: '900px',
      position: 'relative', // Necessary for absolute positioning of video
    }}>
    </div>
  );
}

export default Section;
```

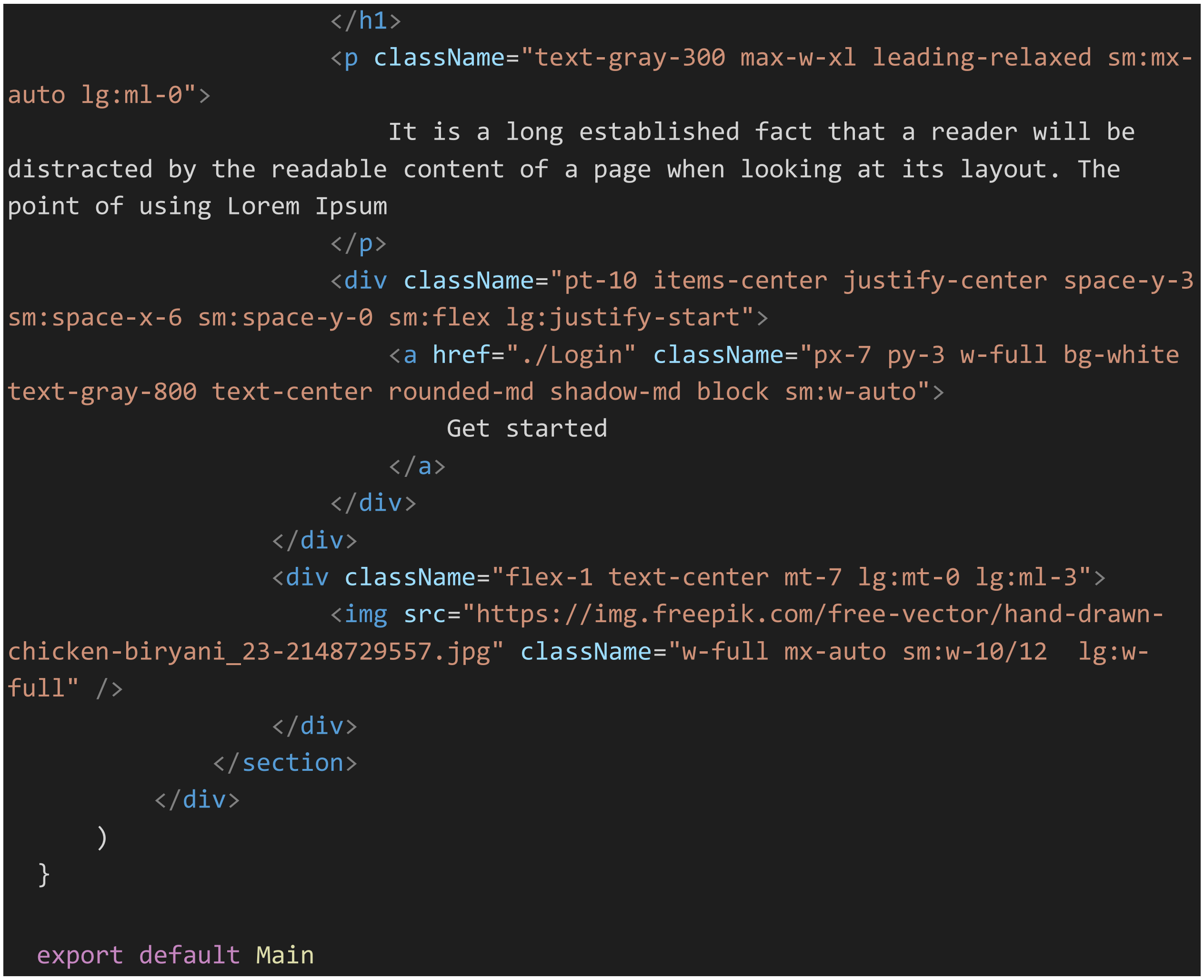
```
import React from 'react';

function Section() {
  return (
    <div style={{
      backgroundImage: `url('https://wallpapers.com/images/hd/food-4k-m37wpodzrcbv5gvw.jpg')`,
      backgroundSize: 'cover',
      height: '900px',
      position: 'relative', // Necessary for absolute positioning of video
    }}>
    </div>
  );
}

export default Section;
```

```
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap');

body {
  font-family: 'Poppins', sans-serif !important;
}
```

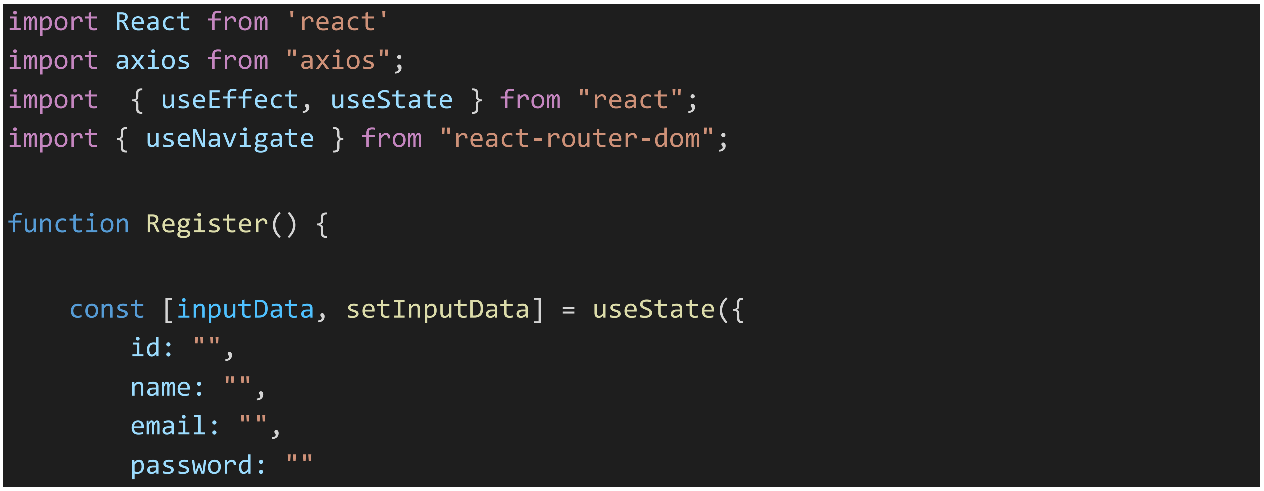
```
import React from "react"
import logo from '../images/logorest.jpg'

function Main() {
```



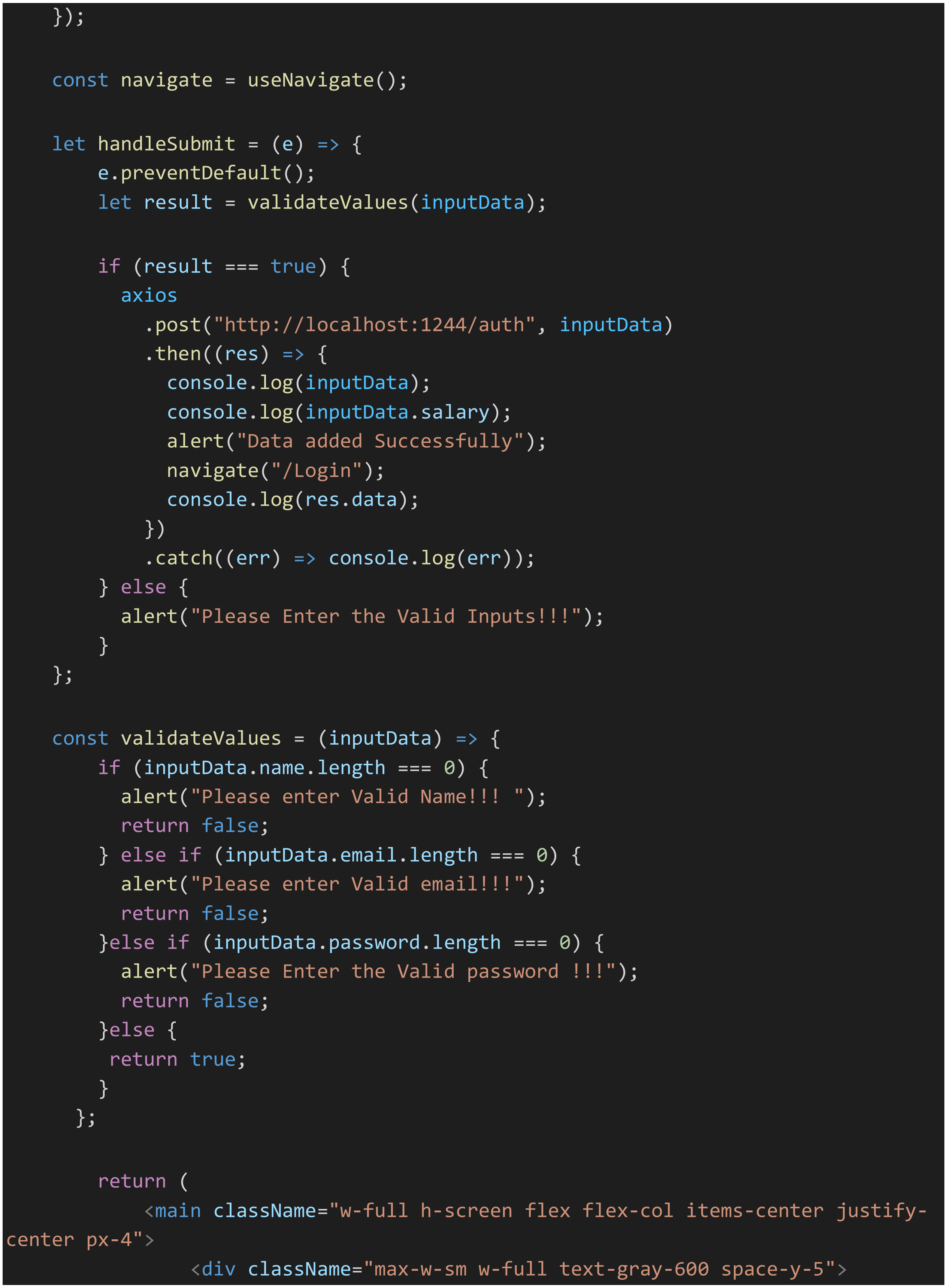
```
    const navigation = [
        { title: "Customers", path: "javascript:void(0)" },
        { title: "Careers", path: "javascript:void(0)" },
    ]

      return (
          <div className="bg-gray-900">
              <header>
                  <nav className="items-center pt-5 px-4 mx-auto max-w-screen-xl
sm:px-8 sm:flex sm:space-x-6">
                      <ul className="py-4 flex-1 items-center flex space-x-3
sm:space-x-6 sm:justify-end">
                          {
                              navigation.map((item, idx) => (
                                  <li className="text-gray-200" key={idx}>
                                      <a href={item.path}>{item.title}</a>
                                  </li>
                              ))
                          }
                          <li>
                              <a href="./Login" className="flex items-center
text-gray-200">
                                  Log In
                                  <svg xmlns="http://www.w3.org/2000/svg"
className="h-5 w-5 ml-2" viewBox="0 0 20 20" fill="currentColor">
                                      <path fillRule="evenodd" d="M3 3a1 1 0 011
1v12a1 1 0 11-2 0V4a1 1 0 011-1zm7.707 3.293a1 1 0 010 1.414L9.414 9H17a1 1 0 110
2H9.414l1.293 1.293a1 1 0 01-1.414 1.414l-3-3a1 1 0 010-1.414l3-3a1 1 0 011.414
0z" clipRule="evenodd" />
                                  </svg>
                              </a>
                          </li>
                      </ul>
                  </nav>
              </header>
              <section className="mt-24 mx-auto max-w-screen-xl pb-12 px-4 items-
center lg:flex md:px-8">
                  <div className="space-y-4 flex-1 sm:text-center lg:text-left">
                      <h1 className="text-white font-bold text-4xl xl:text-5xl">
                        FOODS THATS
                        <span className="text-red-500"> MAKE YOU HAPPY</span>
```



```
                      </h1>
                      <p className="text-gray-300 max-w-xl leading-relaxed sm:mx-auto lg:ml-0">
                          It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum
                      </p>
                      <div className="pt-10 items-center justify-center space-y-3 sm:space-x-6 sm:space-y-0 sm:flex lg:justify-start">
                          <a href="./Login" className="px-7 py-3 w-full bg-white text-gray-800 text-center rounded-md shadow-md block sm:w-auto">
                              Get started
                          </a>
                      </div>
                  </div>
                  <div className="flex-1 text-center mt-7 lg:mt-0 lg:ml-3">
                      <img src="https://img.freepik.com/free-vector/hand-drawn-chicken-biryani_23-2148729557.jpg" className="w-full mx-auto sm:w-10/12  lg:w-full" />
                  </div>
              </section>
          </div>
      )
  }

  export default Main
```

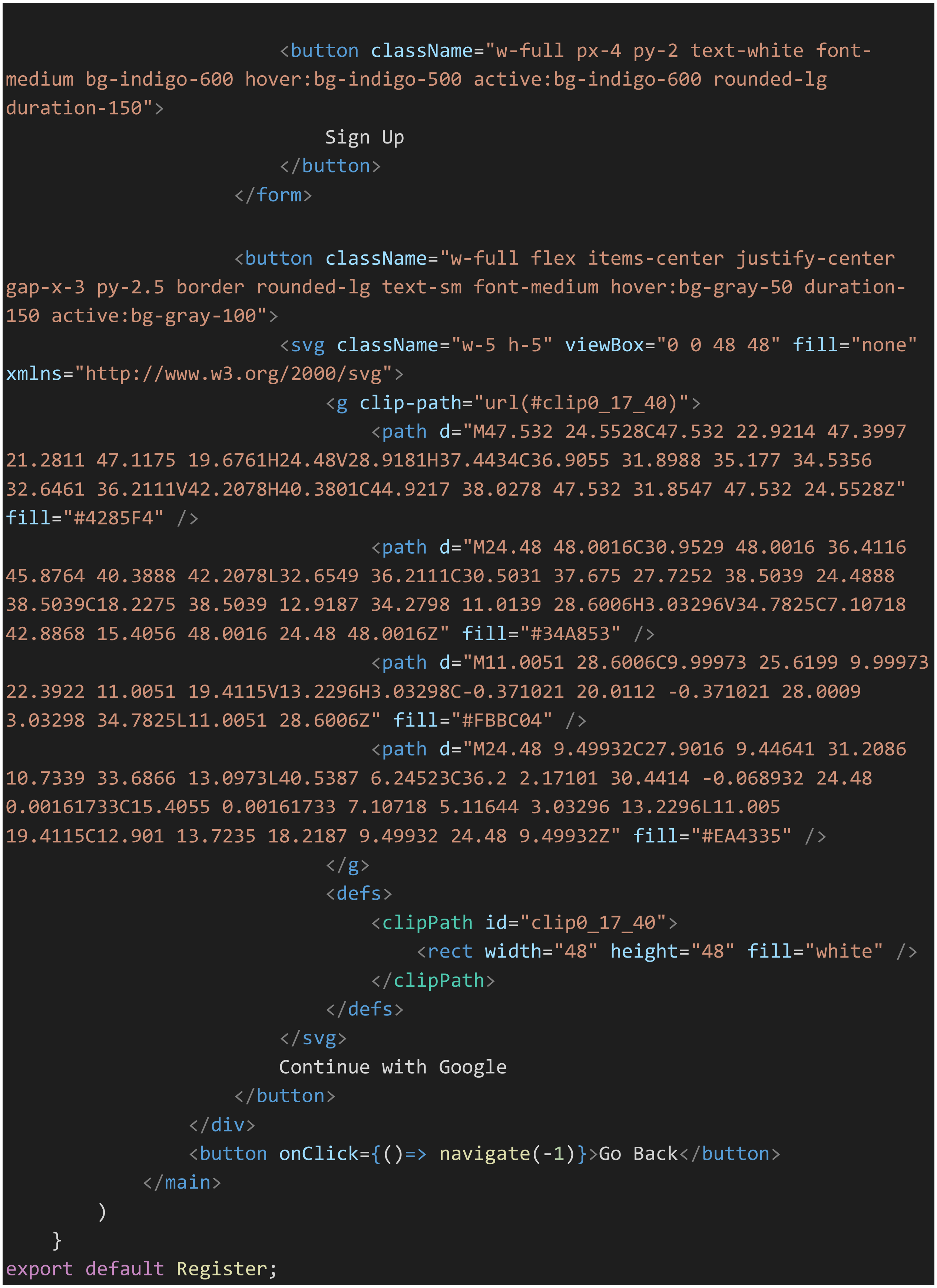


```
import React from 'react'
import axios from "axios";
import  { useEffect, useState } from "react";
import { useNavigate } from "react-router-dom";

function Register() {

    const [inputData, setInputData] = useState({
        id: "",
        name: "",
        email: "",
        password: ""
```

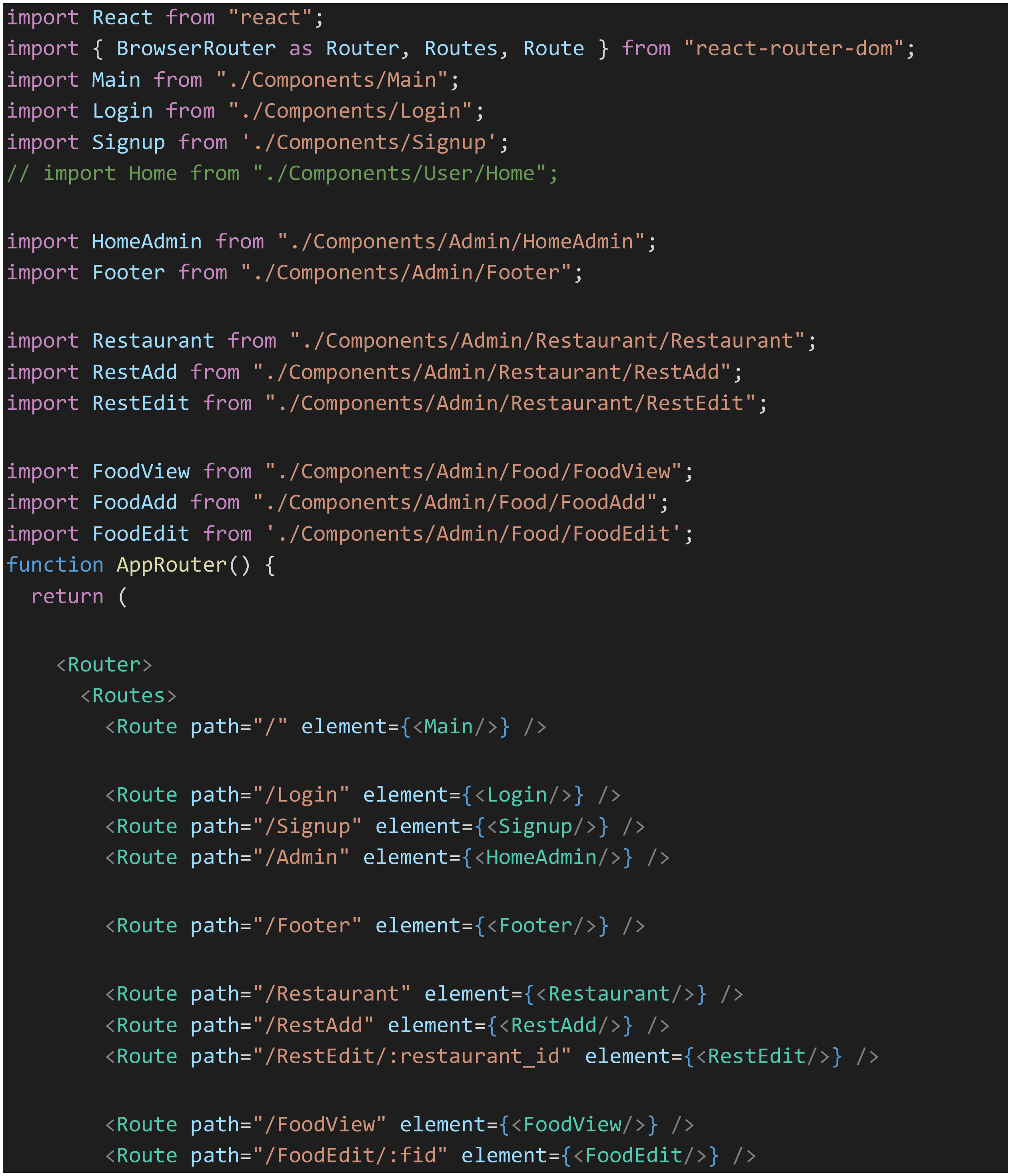


```
    });

    const navigate = useNavigate();

    let handleSubmit = (e) => {
        e.preventDefault();
        let result = validateValues(inputData);

        if (result === true) {
          axios
            .post("http://localhost:1244/auth", inputData)
            .then((res) => {
              console.log(inputData);
              console.log(inputData.salary);
              alert("Data added Successfully");
              navigate("/Login");
              console.log(res.data);
            })
            .catch((err) => console.log(err));
        } else {
          alert("Please Enter the Valid Inputs!!!");
        }
    };

    const validateValues = (inputData) => {
        if (inputData.name.length === 0) {
          alert("Please enter Valid Name!!! ");
          return false;
        } else if (inputData.email.length === 0) {
          alert("Please enter Valid email!!!");
          return false;
        }else if (inputData.password.length === 0) {
          alert("Please Enter the Valid password !!!");
          return false;
        }else {
         return true;
        }
      };

        return (
            <main className="w-full h-screen flex flex-col items-center justify-
center px-4">
                <div className="max-w-sm w-full text-gray-600 space-y-5">
```

```
                    <div className="text-center pb-8">
                        <div className="mt-5">
                            <h3 className="text-gray-800 text-2xl font-bold
sm:text-3xl">Create a New Account</h3>
                        </div>
                    </div>
                    <form onSubmit={handleSubmit} className="space-y-5">

                        <div>
                            <label className="font-medium">Name</label>
                            <input type="text" name="name" required className="w-
full mt-2 px-3 py-2 text-gray-500 bg-transparent outline-none border
focus:border-indigo-600 shadow-sm rounded-lg"
                                onChange={(e) =>
                                    setInputData({ ...inputData, name:
e.target.value })
                                }
                            />
                        </div>

                        <div>
                            <label className="font-medium">Email</label>
                            <input type="text" name="email" required
className="w-full mt-2 px-3 py-2 text-gray-500 bg-transparent outline-none border
focus:border-indigo-600 shadow-sm rounded-lg"
                                onChange={(e) =>
                                    setInputData({ ...inputData, email:
e.target.value })
                                }
                            />
                        </div>

                        <div>
                            <label className="font-medium">Password</label>
                            <input type="password" name="password" required
className="w-full mt-2 px-3 py-2 text-gray-500 bg-transparent outline-none border
focus:border-indigo-600 shadow-sm rounded-lg"
                                onChange={(e) =>
                                    setInputData({ ...inputData, password:
e.target.value })
                                }
                            />
                        </div>
```

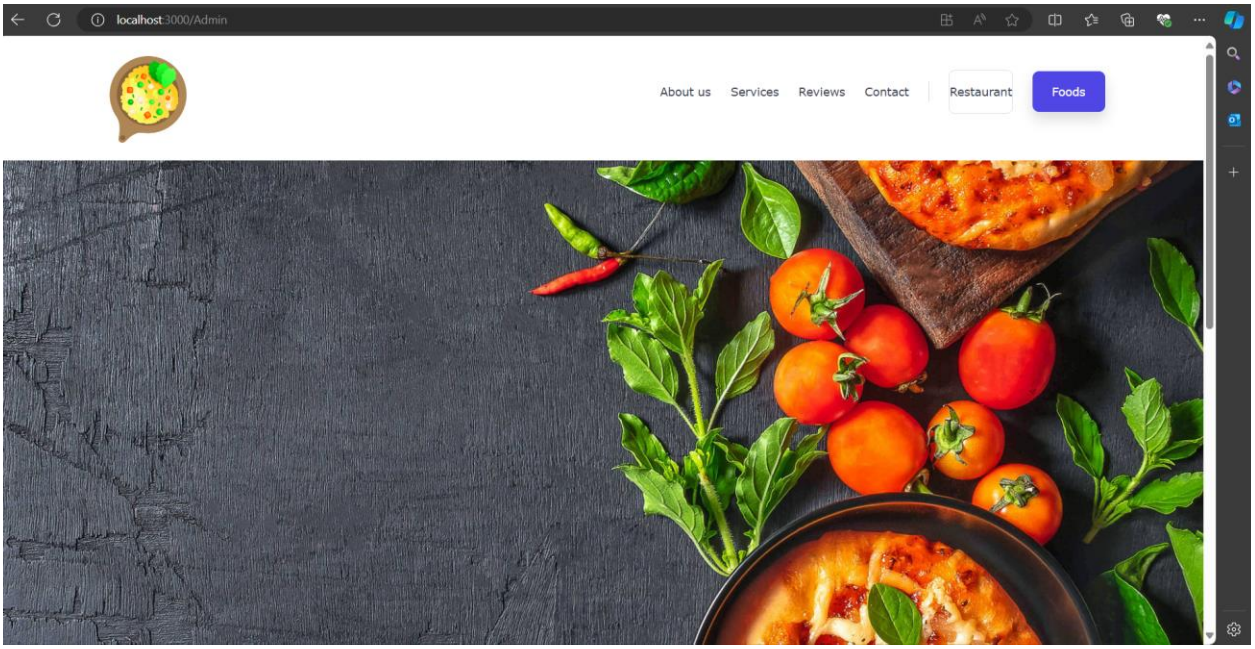


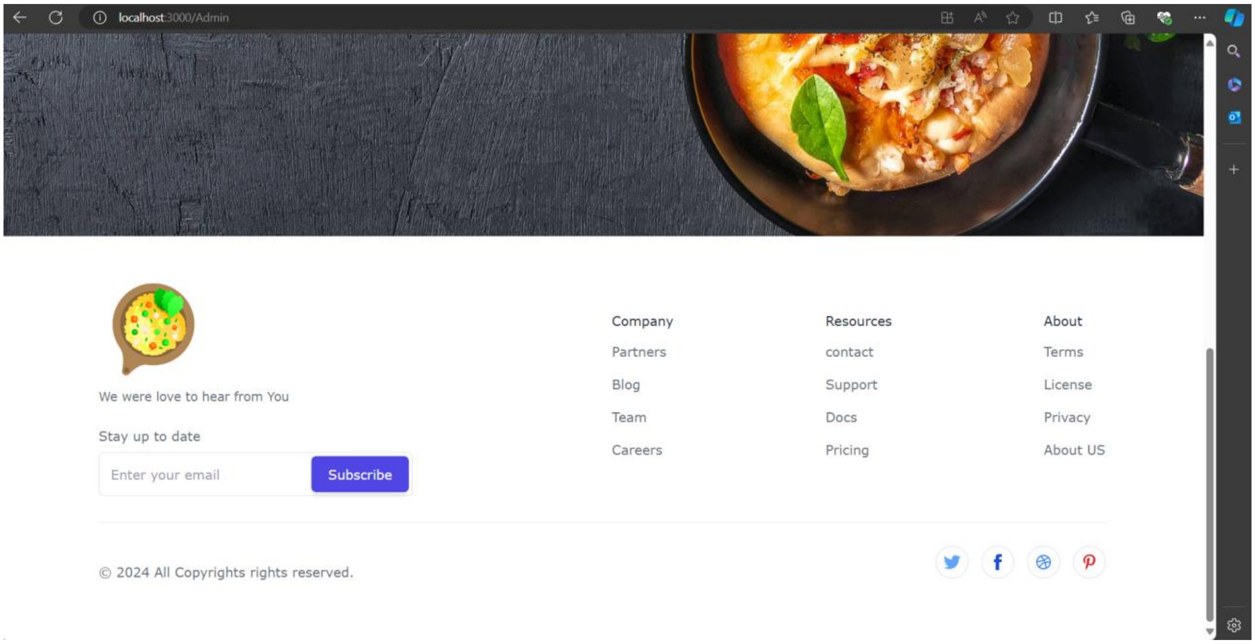
```
                        <button className="w-full px-4 py-2 text-white font-medium bg-indigo-600 hover:bg-indigo-500 active:bg-indigo-600 rounded-lg duration-150">
                            Sign Up
                        </button>
                    </form>

                    <button className="w-full flex items-center justify-center gap-x-3 py-2.5 border rounded-lg text-sm font-medium hover:bg-gray-50 duration-150 active:bg-gray-100">
                        <svg className="w-5 h-5" viewBox="0 0 48 48" fill="none" xmlns="http://www.w3.org/2000/svg">
                            <g clip-path="url(#clip0_17_40)">
                                <path d="M47.532 24.5528C47.532 22.9214 47.3997 21.2811 47.1175 19.6761H24.48V28.9181H37.4434C36.9055 31.8988 35.177 34.5356 32.6461 36.2111V42.2078H40.3801C44.9217 38.0278 47.532 31.8547 47.532 24.5528Z" fill="#4285F4" />
                                <path d="M24.48 48.0016C30.9529 48.0016 36.4116 45.8764 40.3888 42.2078L32.6549 36.2111C30.5031 37.675 27.7252 38.5039 24.4888 38.5039C18.2275 38.5039 12.9187 34.2798 11.0139 28.6006H3.03296V34.7825C7.10718 42.8868 15.4056 48.0016 24.48 48.0016Z" fill="#34A853" />
                                <path d="M11.0051 28.6006C9.99973 25.6199 9.99973 22.3922 11.0051 19.4115V13.2296H3.03298C-0.371021 20.0112 -0.371021 28.0009 3.03298 34.7825L11.0051 28.6006Z" fill="#FBBC04" />
                                <path d="M24.48 9.49932C27.9016 9.44641 31.2086 10.7339 33.6866 13.0973L40.5387 6.24523C36.2 2.17101 30.4414 -0.068932 24.48 0.00161733C15.4055 0.00161733 7.10718 5.11644 3.03296 13.2296L11.005 19.4115C12.901 13.7235 18.2187 9.49932 24.48 9.49932Z" fill="#EA4335" />
                            </g>
                            <defs>
                                <clipPath id="clip0_17_40">
                                    <rect width="48" height="48" fill="white" />
                                </clipPath>
                            </defs>
                        </svg>
                        Continue with Google
                    </button>
                </div>
                <button onClick={()=> navigate(-1)}>Go Back</button>
            </main>
        )
    }
export default Register;
```



```
import React from "react";
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import Main from "./Components/Main";
import Login from "./Components/Login";
import Signup from './Components/Signup';
// import Home from "./Components/User/Home";

import HomeAdmin from "./Components/Admin/HomeAdmin";
import Footer from "./Components/Admin/Footer";

import Restaurant from "./Components/Admin/Restaurant/Restaurant";
import RestAdd from "./Components/Admin/Restaurant/RestAdd";
import RestEdit from "./Components/Admin/Restaurant/RestEdit";

import FoodView from "./Components/Admin/Food/FoodView";
import FoodAdd from "./Components/Admin/Food/FoodAdd";
import FoodEdit from './Components/Admin/Food/FoodEdit';
function AppRouter() {
  return (

    <Router>
      <Routes>
        <Route path="/" element={<Main/>} />

        <Route path="/Login" element={<Login/>} />
        <Route path="/Signup" element={<Signup/>} />
        <Route path="/Admin" element={<HomeAdmin/>} />

        <Route path="/Footer" element={<Footer/>} />

        <Route path="/Restaurant" element={<Restaurant/>} />
        <Route path="/RestAdd" element={<RestAdd/>} />
        <Route path="/RestEdit/:restaurant_id" element={<RestEdit/>} />

        <Route path="/FoodView" element={<FoodView/>} />
        <Route path="/FoodEdit/:fid" element={<FoodEdit/>} />
        <Route path="/FoodAdd" element={<FoodAdd/>} />
      </Routes>
    </Router>

  );
}

export default AppRouter;
```

Output ScreenShots:

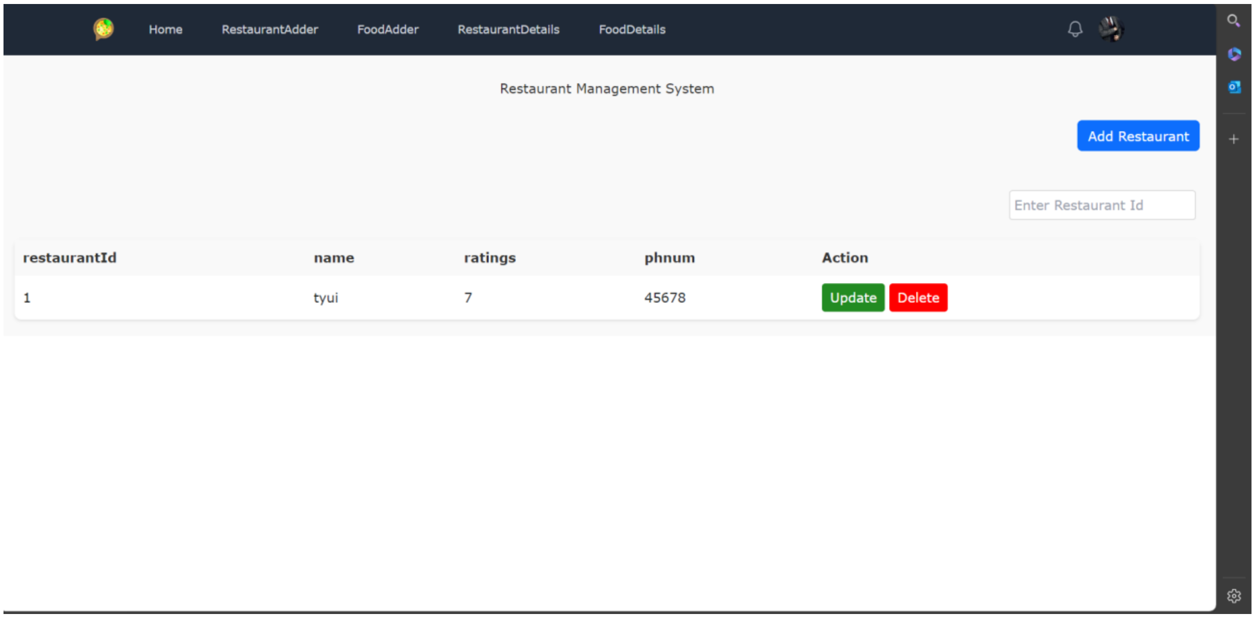




We were love to hear from You

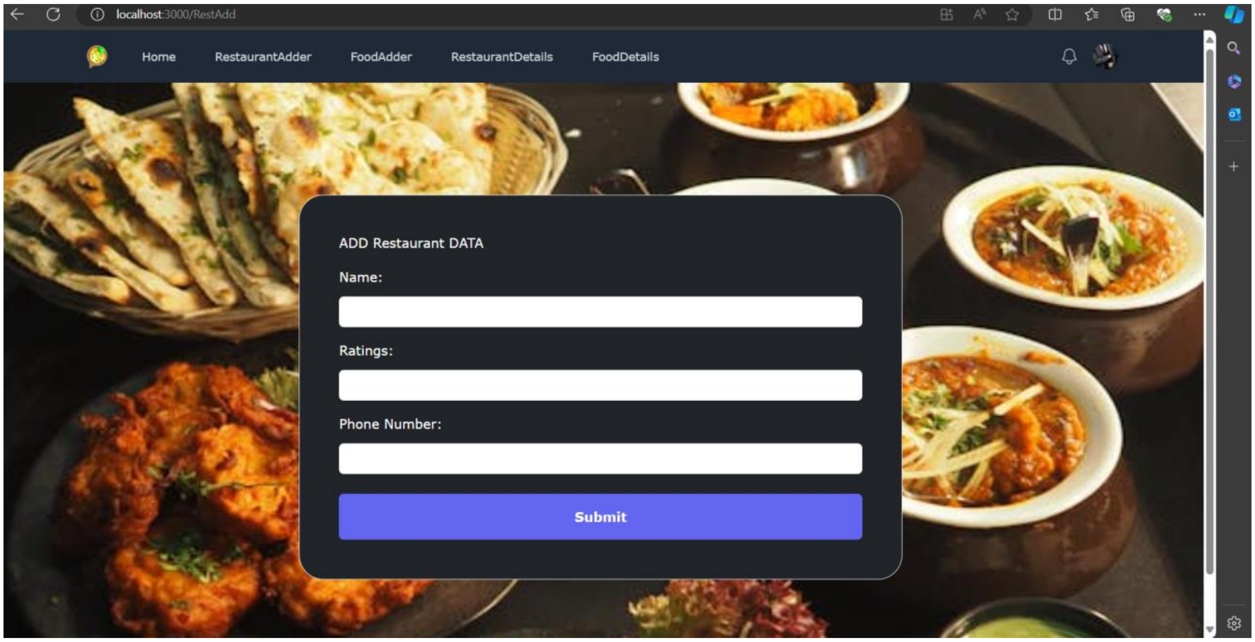
Stay up to date

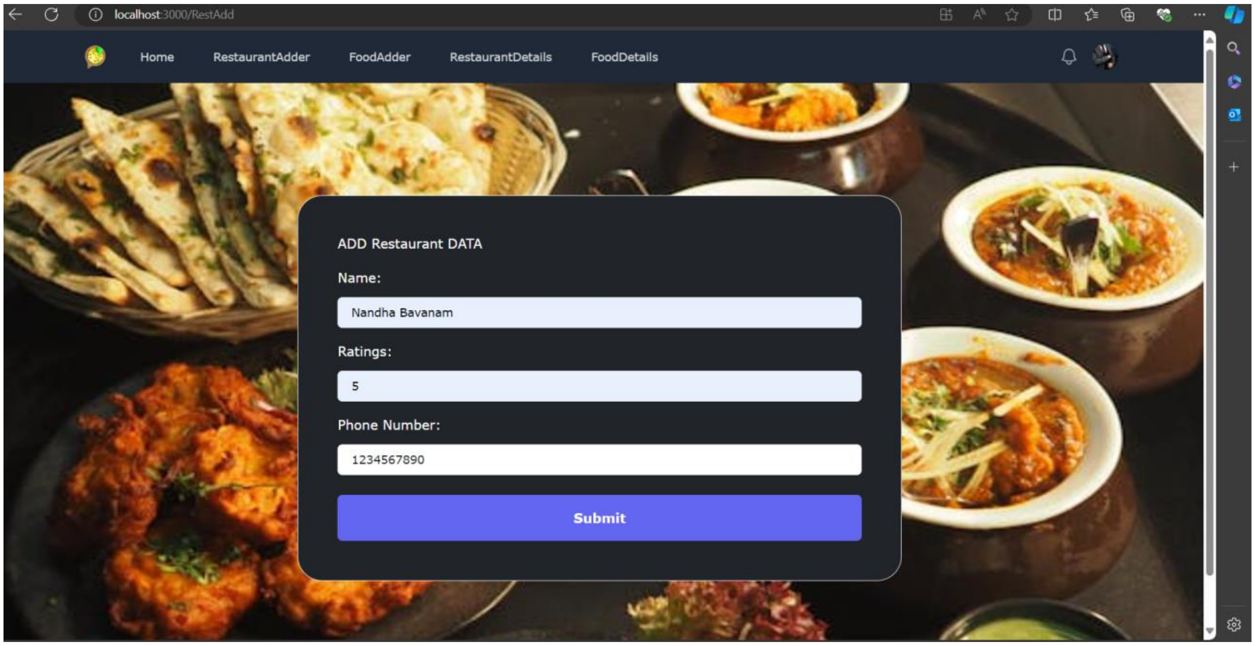
© 2024 All Copyrights rights reserved.



Restaurant Management System

restaurantId	name	ratings	phnum	Action
1	tyui	7	45678	Update Delete

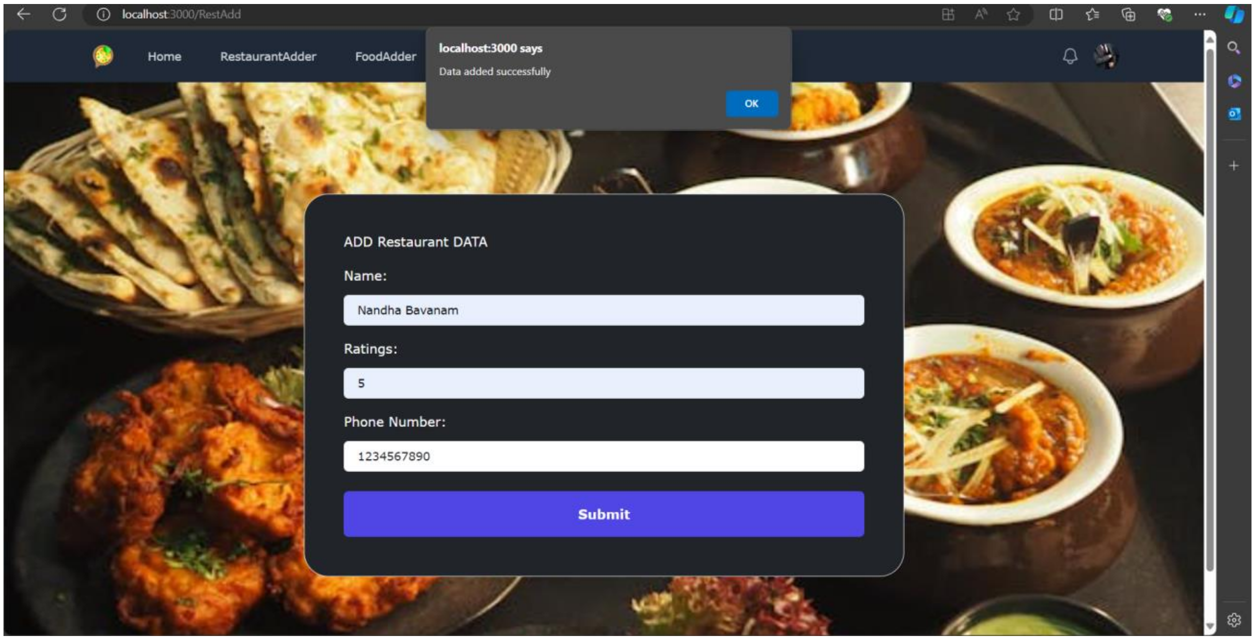




Name: Nandha Bavanam

Ratings: 5

Phone Number: 1234567890



localhost:3000 says

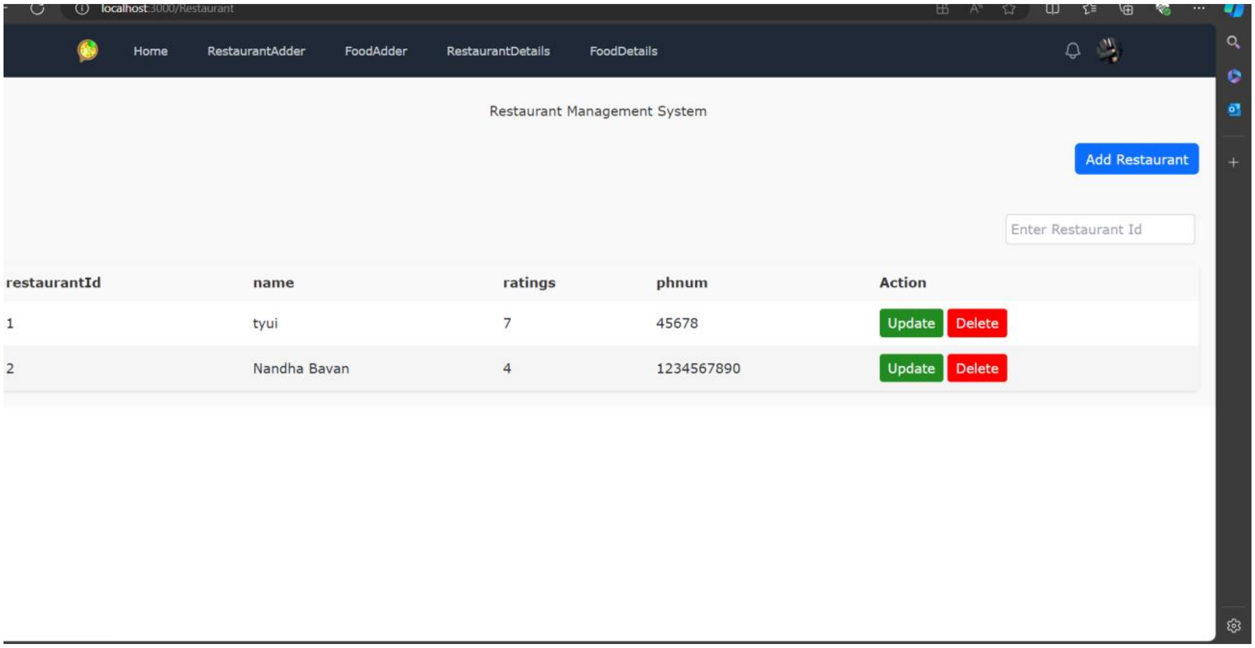
Data added successfully

ADD Restaurant DATA

Name: Nandha Bavanam

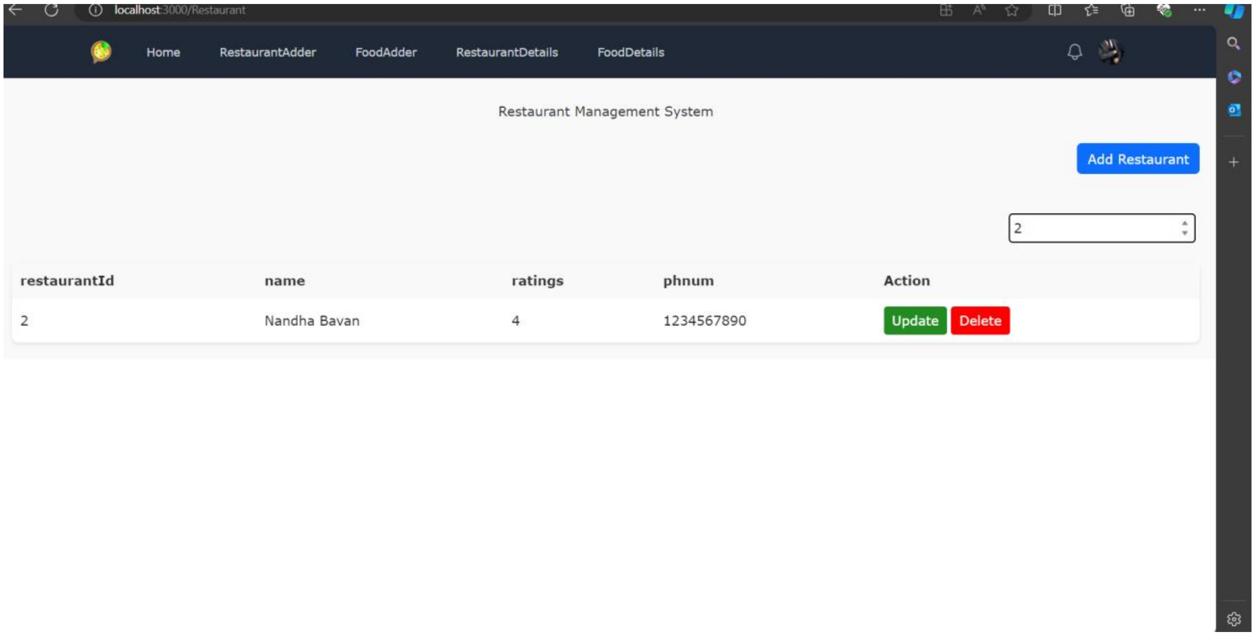
Ratings: 5

Phone Number: 1234567890



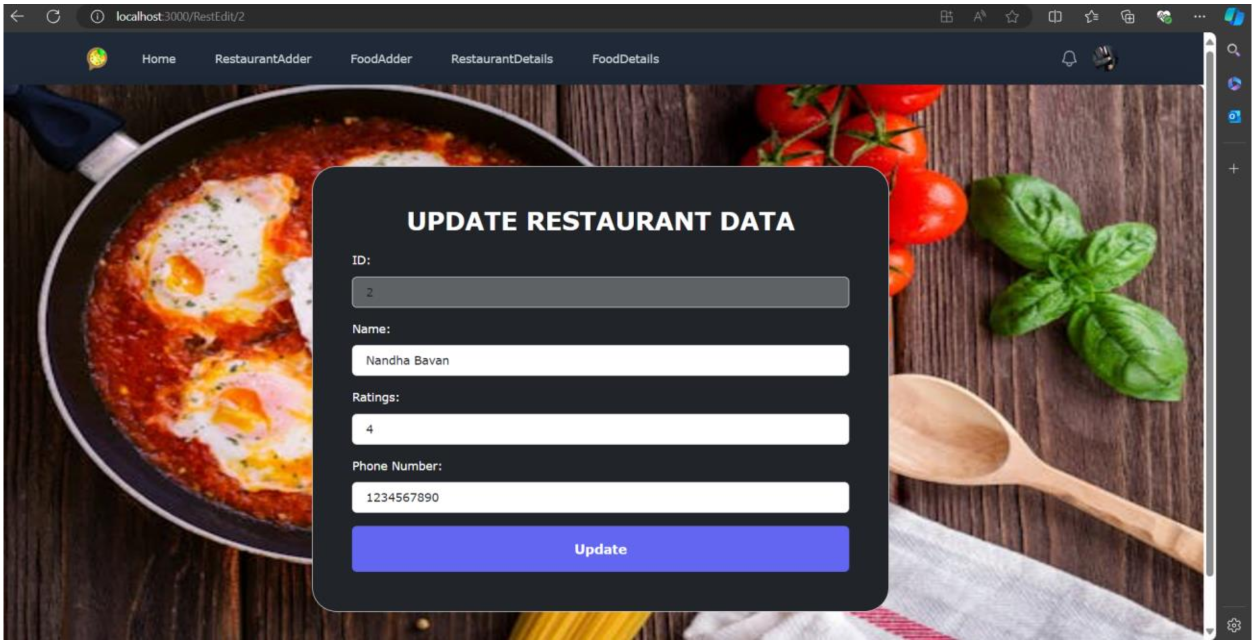
Restaurant Management System

restaurantId	name	ratings	phnum	Action
1	tyui	7	45678	Update Delete
2	Nandha Bavan	4	1234567890	Update Delete



Restaurant Management System

restaurantId	name	ratings	phnum	Action
2	Nandha Bavan	4	1234567890	Update Delete



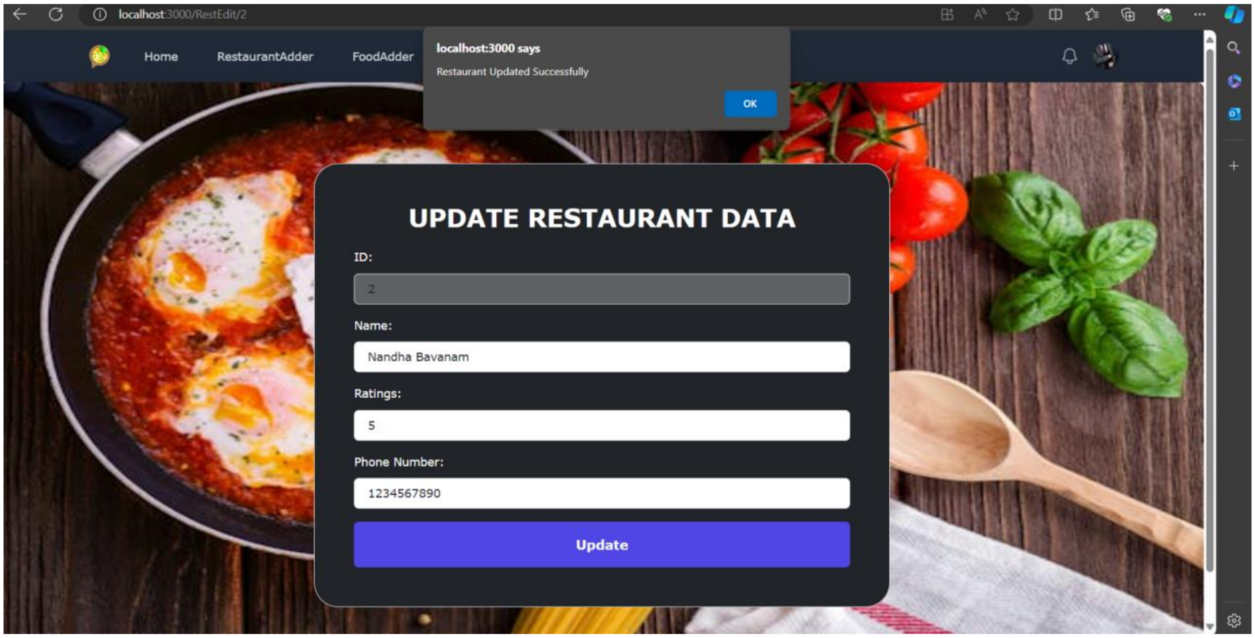
UPDATE RESTAURANT DATA

ID: 2

Name: Nandha Bavan

Ratings: 4

Phone Number: 1234567890



localhost:3000 says

Restaurant Updated Successfully

OK

UPDATE RESTAURANT DATA

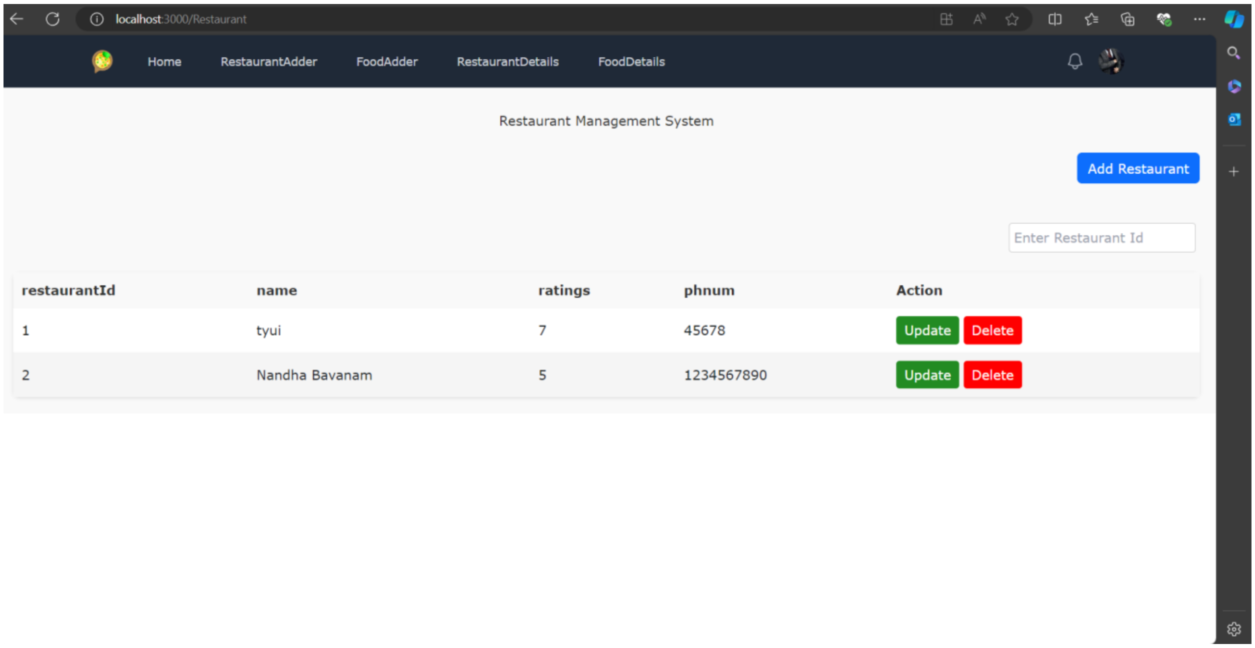
ID: 2

Name: Nandha Bavanam

Ratings: 5

Phone Number: 1234567890

Update

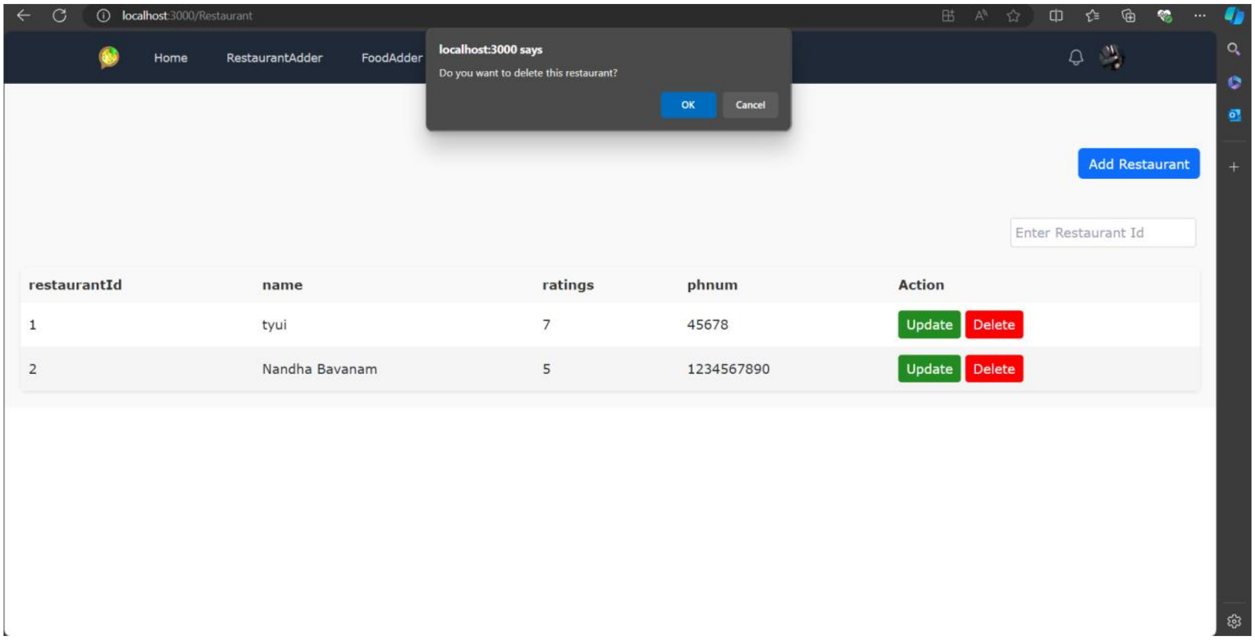


Restaurant Management System

Add Restaurant

Enter Restaurant Id

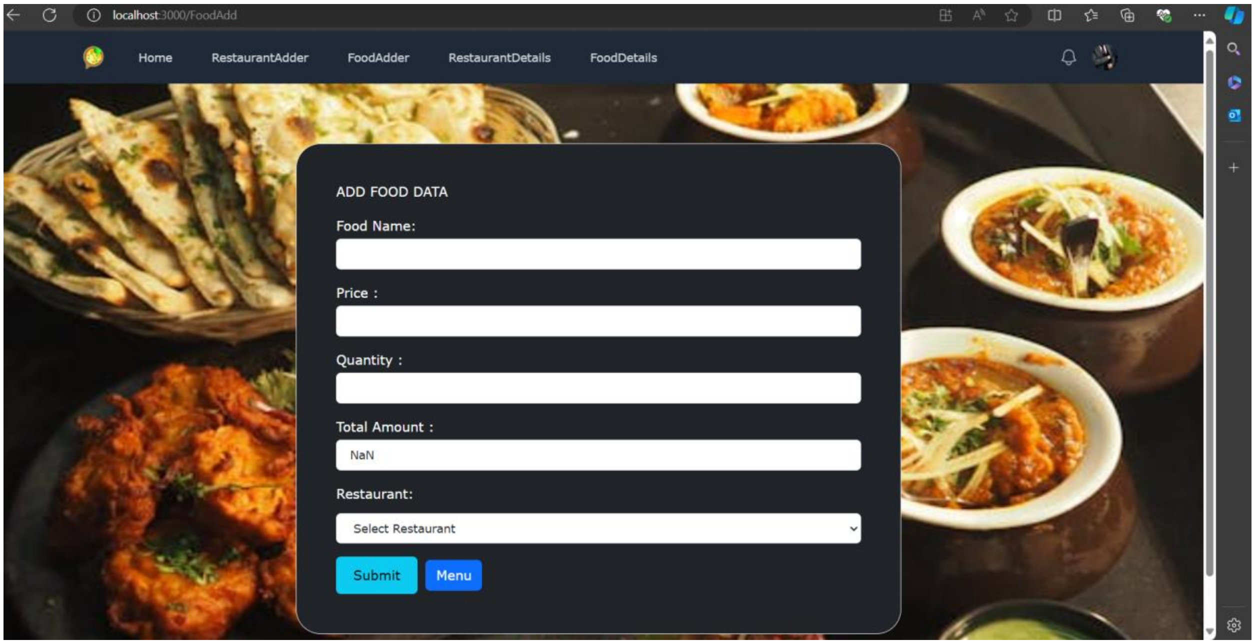
restaurantId	name	ratings	phnum	Action
1	tyui	7	45678	Update Delete
2	Nandha Bavanam	5	1234567890	Update Delete



localhost:3000 says

Do you want to delete this restaurant?

restaurantId	name	ratings	phnum	Action
1	tyui	7	45678	Update Delete
2	Nandha Bavanam	5	1234567890	Update Delete



ADD FOOD DATA

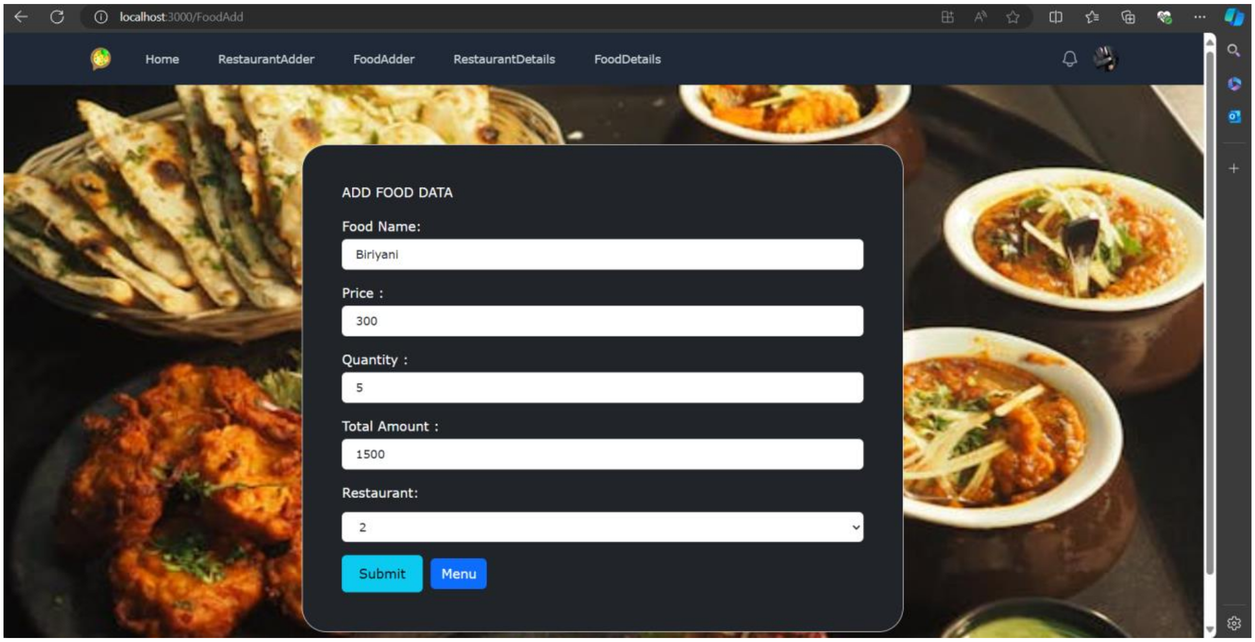
Food Name:

Price :

Quantity :

Total Amount : NaN

Restaurant: Select Restaurant



ADD FOOD DATA

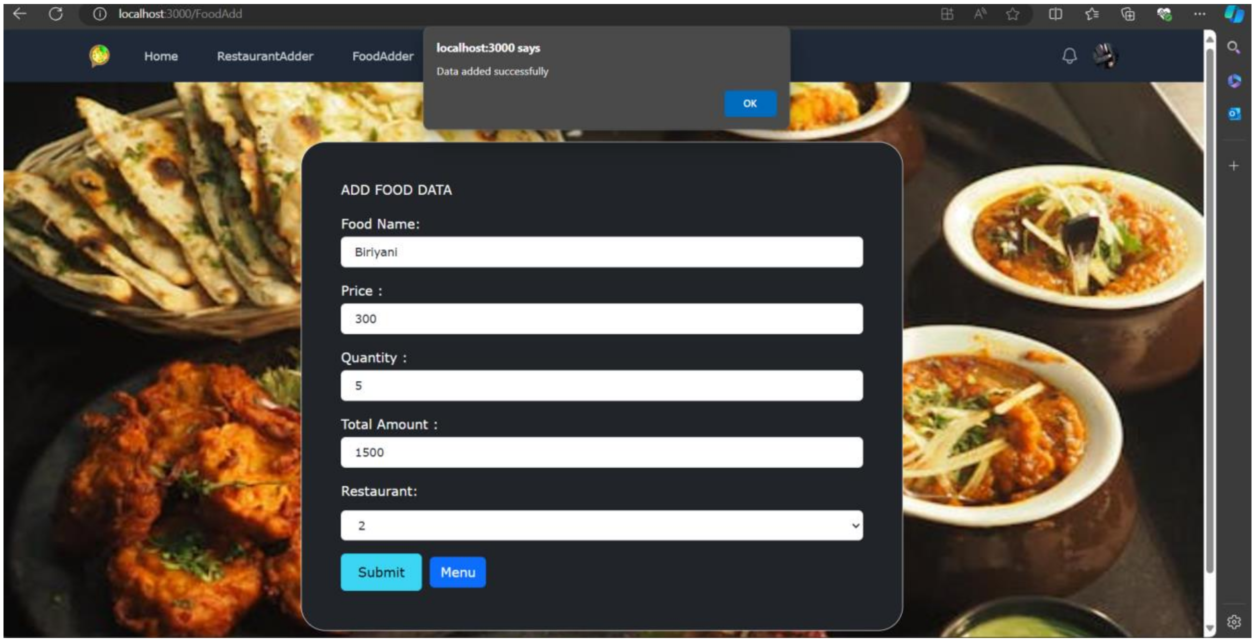
Food Name: Biriyani

Price : 300

Quantity : 5

Total Amount : 1500

Restaurant: 2



localhost:3000 says

Data added successfully

ADD FOOD DATA

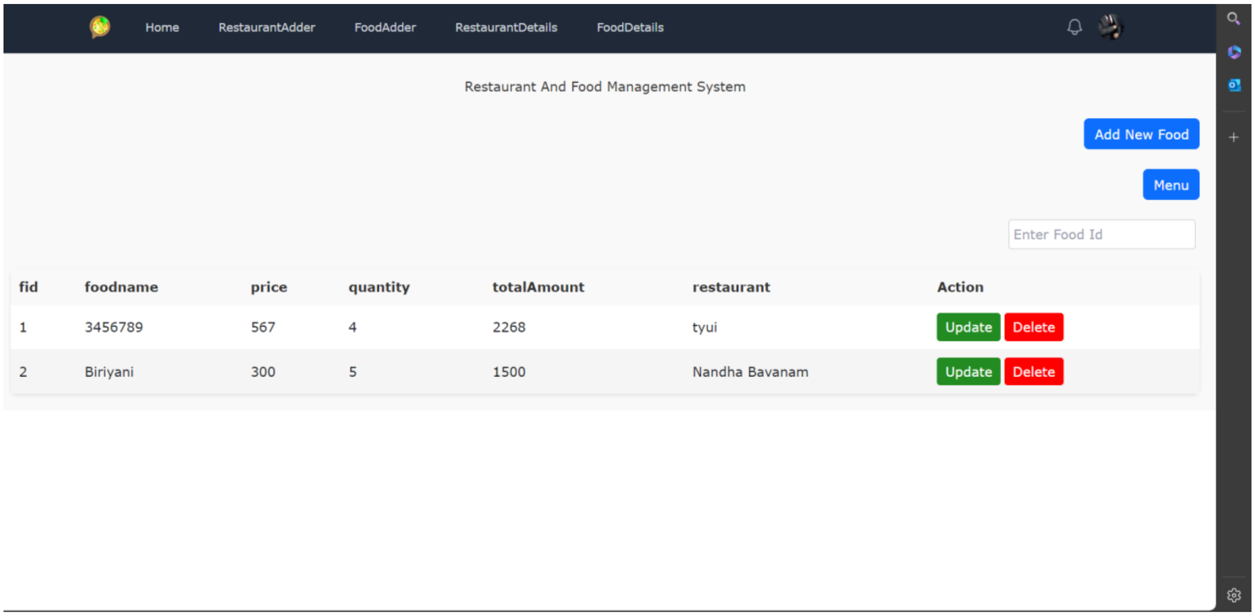
Food Name: Biriyani

Price : 300

Quantity : 5

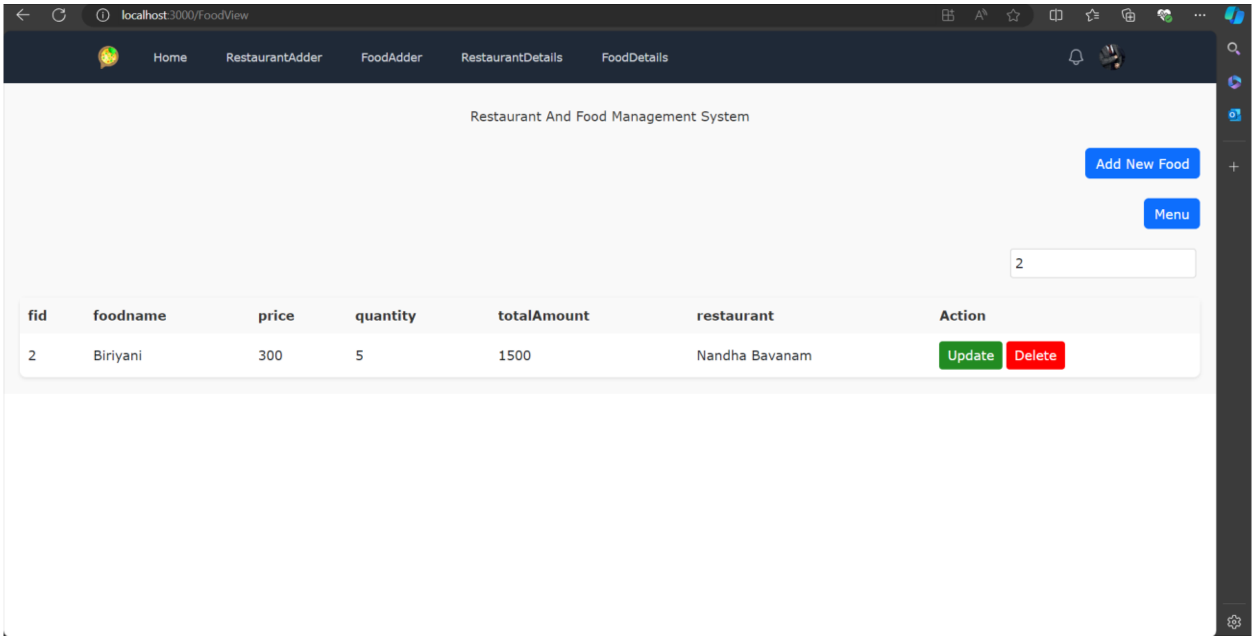
Total Amount : 1500

Restaurant: 2



Restaurant And Food Management System

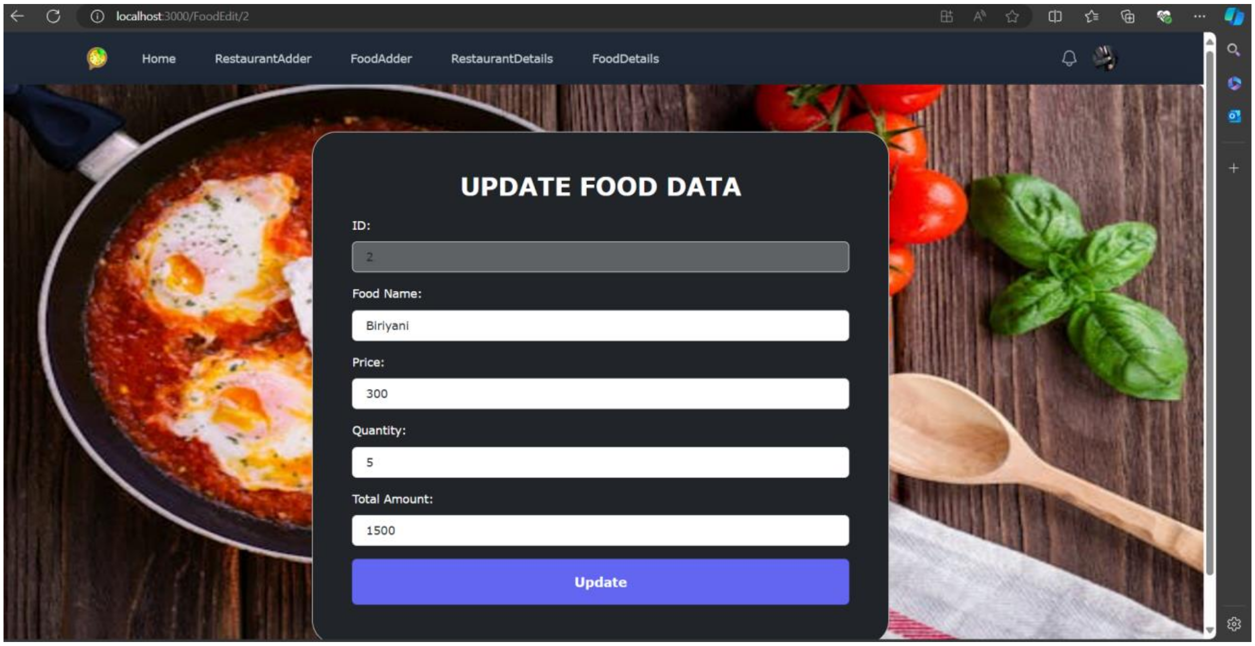
fid	foodname	price	quantity	totalAmount	restaurant	Action
1	3456789	567	4	2268	tyui	Update Delete
2	Biriyani	300	5	1500	Nandha Bavanam	Update Delete



Restaurant And Food Management System

2

fid	foodname	price	quantity	totalAmount	restaurant	Action
2	Biriyani	300	5	1500	Nandha Bavanam	Update Delete



UPDATE FOOD DATA

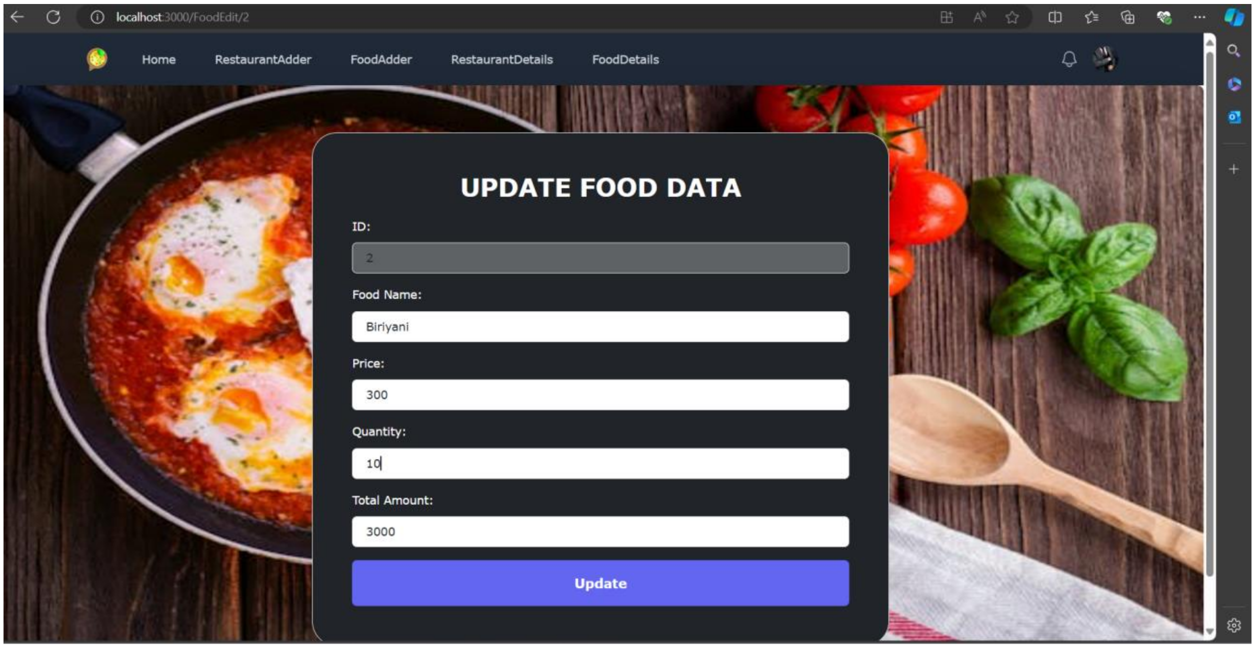
ID: 2

Food Name: Biriyani

Price: 300

Quantity: 5

Total Amount: 1500



UPDATE FOOD DATA

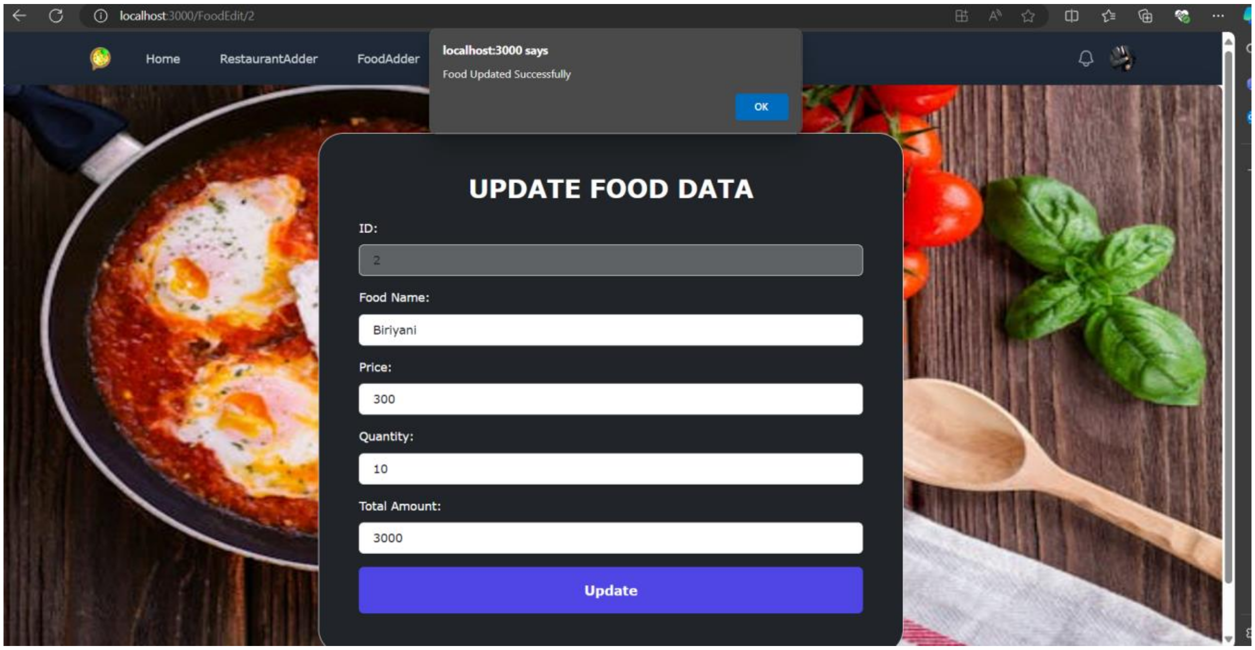
ID: 2

Food Name: Biriyani

Price: 300

Quantity: 10

Total Amount: 3000



localhost:3000 says

Food Updated Successfully

UPDATE FOOD DATA

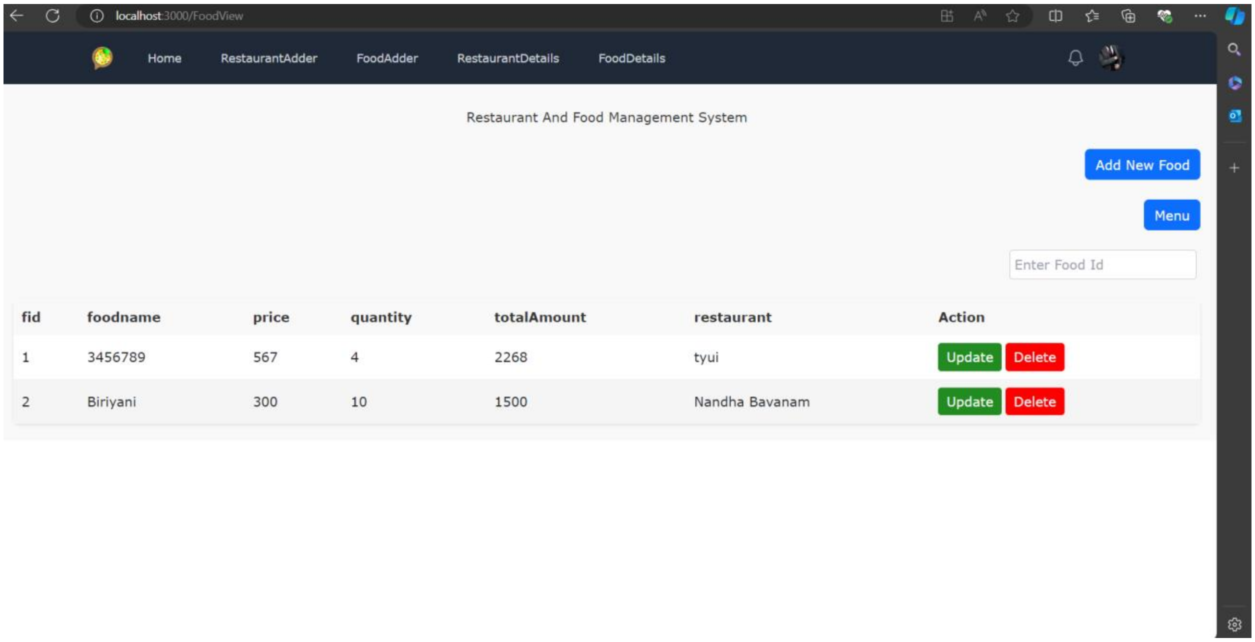
ID: 2

Food Name: Biriyani

Price: 300

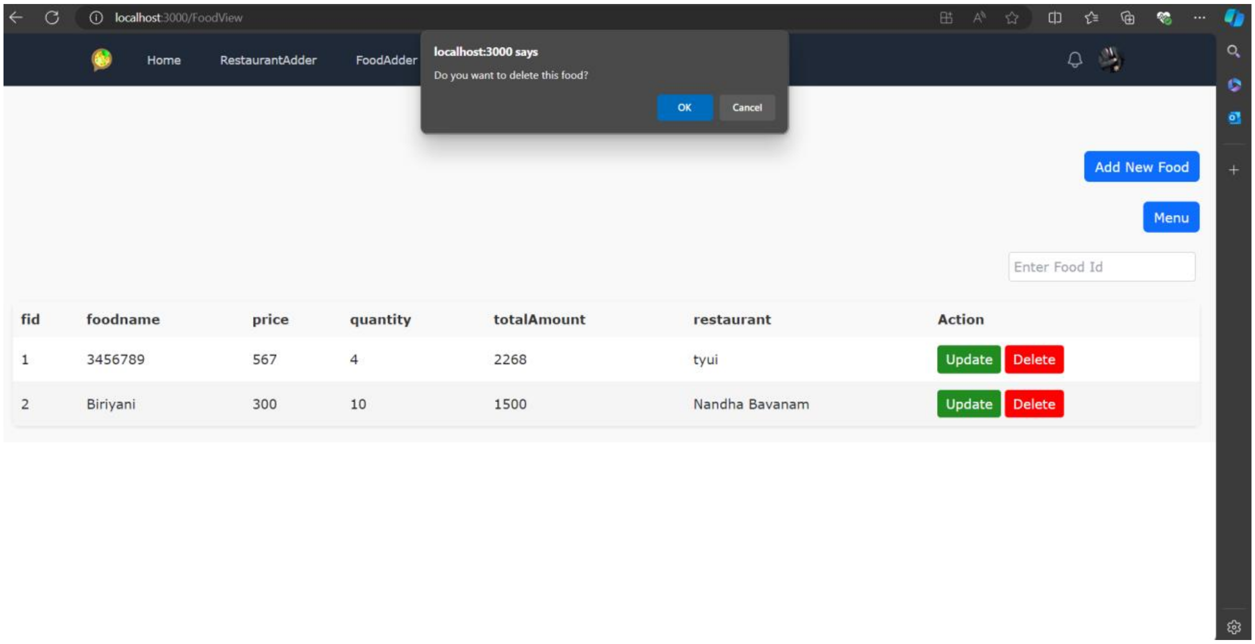
Quantity: 10

Total Amount: 3000



Restaurant And Food Management System

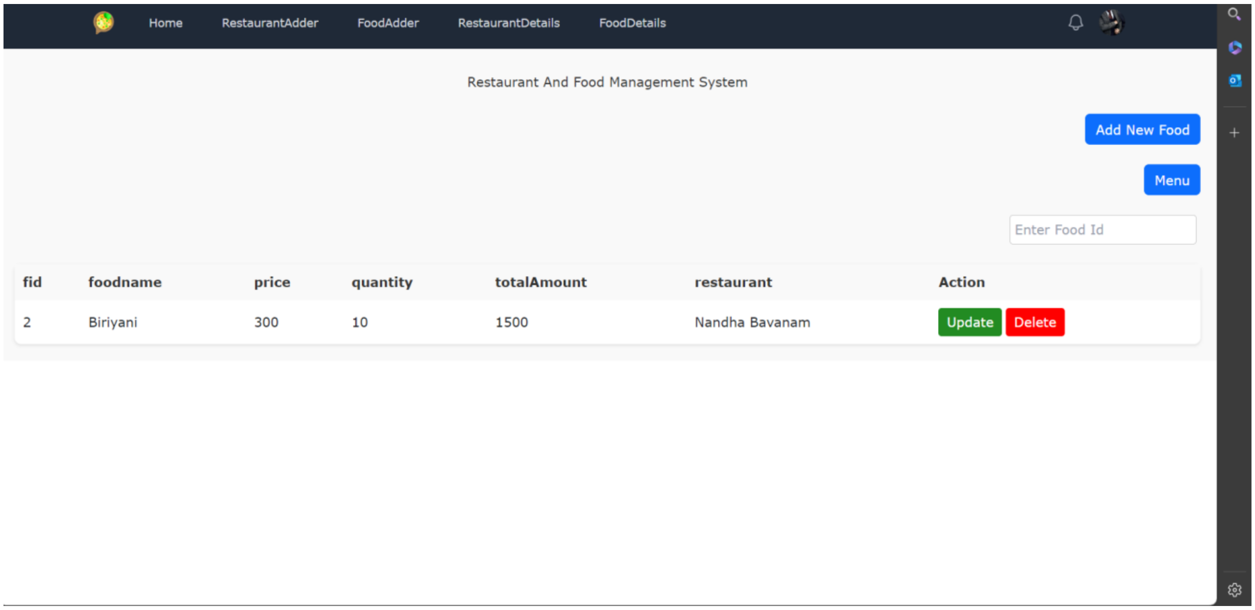
fid	foodname	price	quantity	totalAmount	restaurant	Action
1	3456789	567	4	2268	tyui	Update Delete
2	Biriyani	300	10	1500	Nandha Bavanam	Update Delete



localhost:3000 says

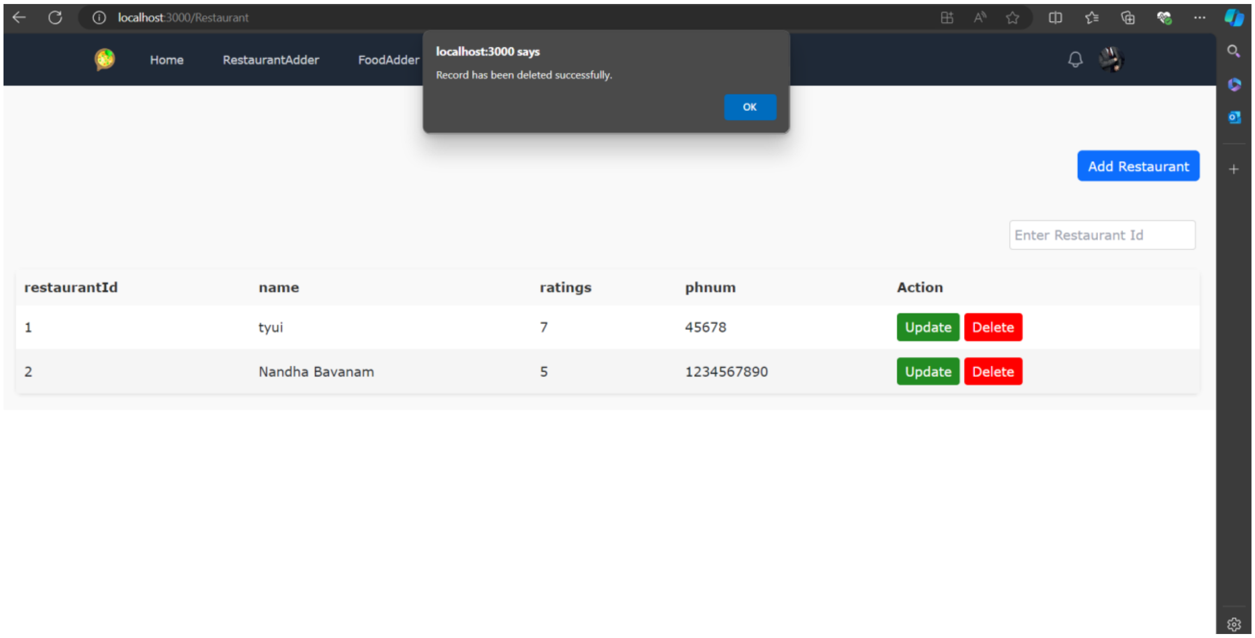
Do you want to delete this food?

fid	foodname	price	quantity	totalAmount	restaurant	Action
1	3456789	567	4	2268	tyui	Update Delete
2	Biriyani	300	10	1500	Nandha Bavanam	Update Delete



Restaurant And Food Management System

fid	foodname	price	quantity	totalAmount	restaurant	Action
2	Biriyani	300	10	1500	Nandha Bavanam	Update Delete



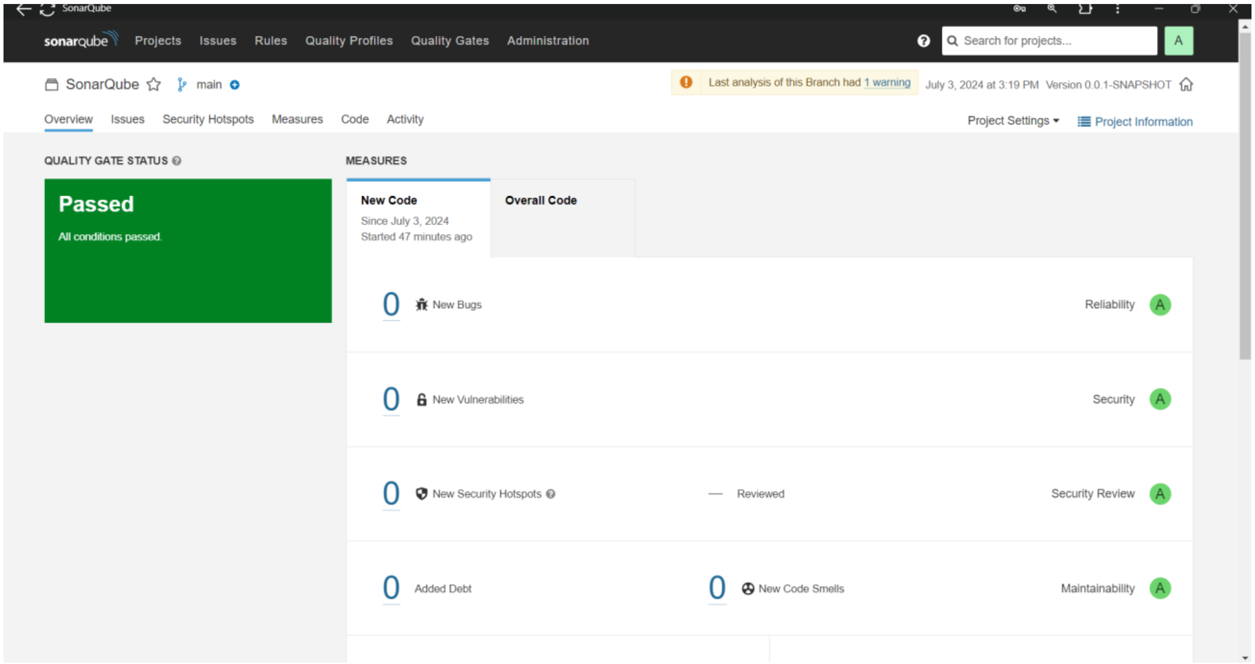
localhost:3000 says

Record has been deleted successfully.

OK

restaurantId	name	ratings	phnum	Action
1	tyui	7	45678	Update Delete
2	Nandha Bavanam	5	1234567890	Update Delete

SonarQube ScreenShot Before Peer Review:



Last analysis of this Branch had 1 warning

July 3, 2024 at 3:19 PM Version 0.0.1-SNAPSHOT

QUALITY GATE STATUS

Passed

All conditions passed.

MEASURES

New Code
Since July 3, 2024
Started 47 minutes ago

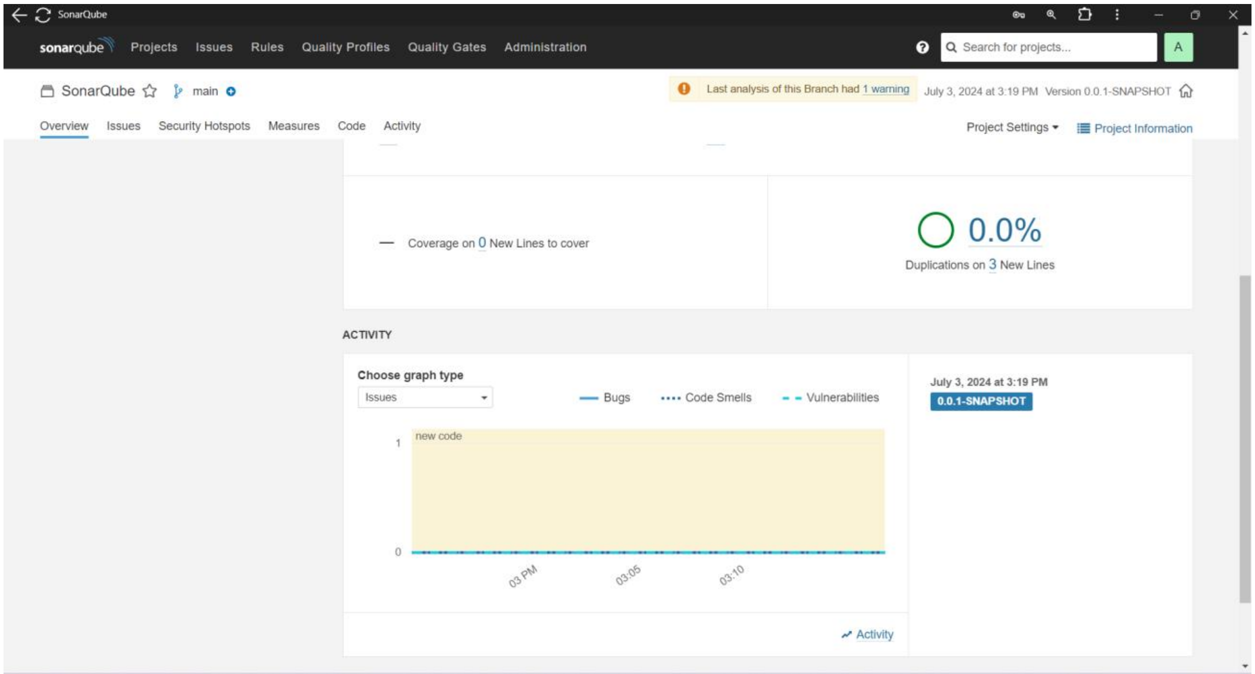
Overall Code

0 New Bugs — Reliability A

0 New Vulnerabilities — Security A

0 New Security Hotspots — Reviewed — Security Review A

0 Added Debt — 0 New Code Smells — Maintainability A



Last analysis of this Branch had 1 warning July 3, 2024 at 3:19 PM Version 0.0.1-SNAPSHOT

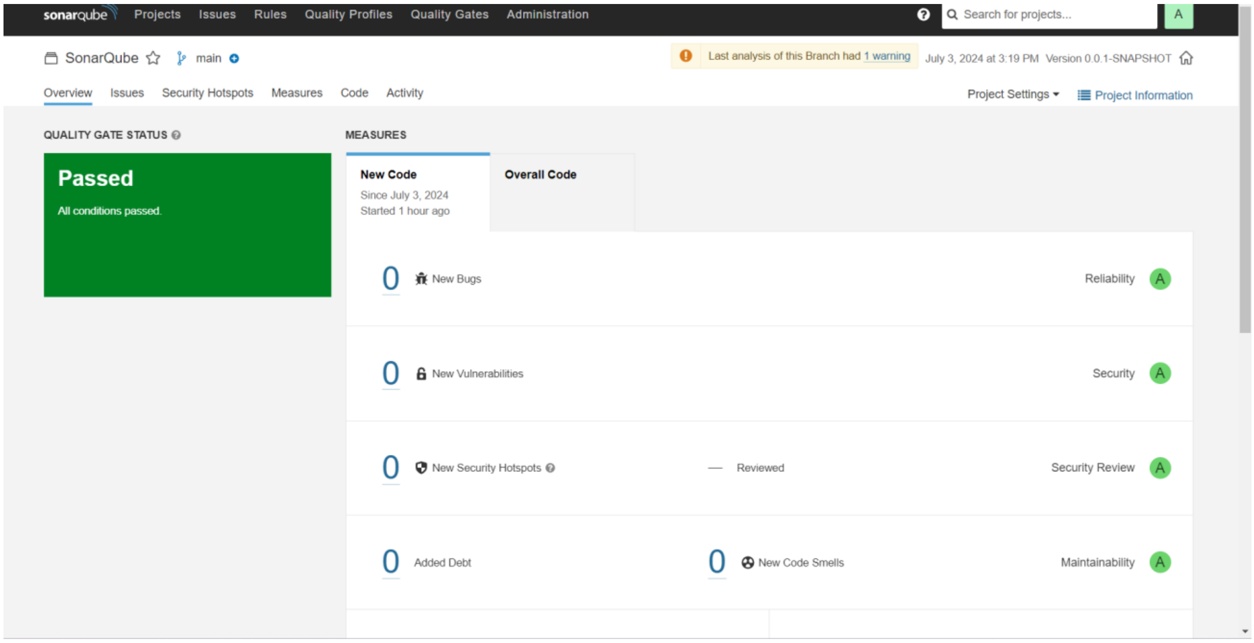
— Coverage on 0 New Lines to cover

0.0%
Duplications on 3 New Lines

ACTIVITY

July 3, 2024 at 3:19 PM
0.0.1-SNAPSHOT

SonarQube ScreenShot After Peer Review:



Last analysis of this Branch had 1 warning July 3, 2024 at 3:19 PM Version 0.0.1-SNAPSHOT

QUALITY GATE STATUS

Passed
All conditions passed.

MEASURES

New Code
Since July 3, 2024
Started 1 hour ago

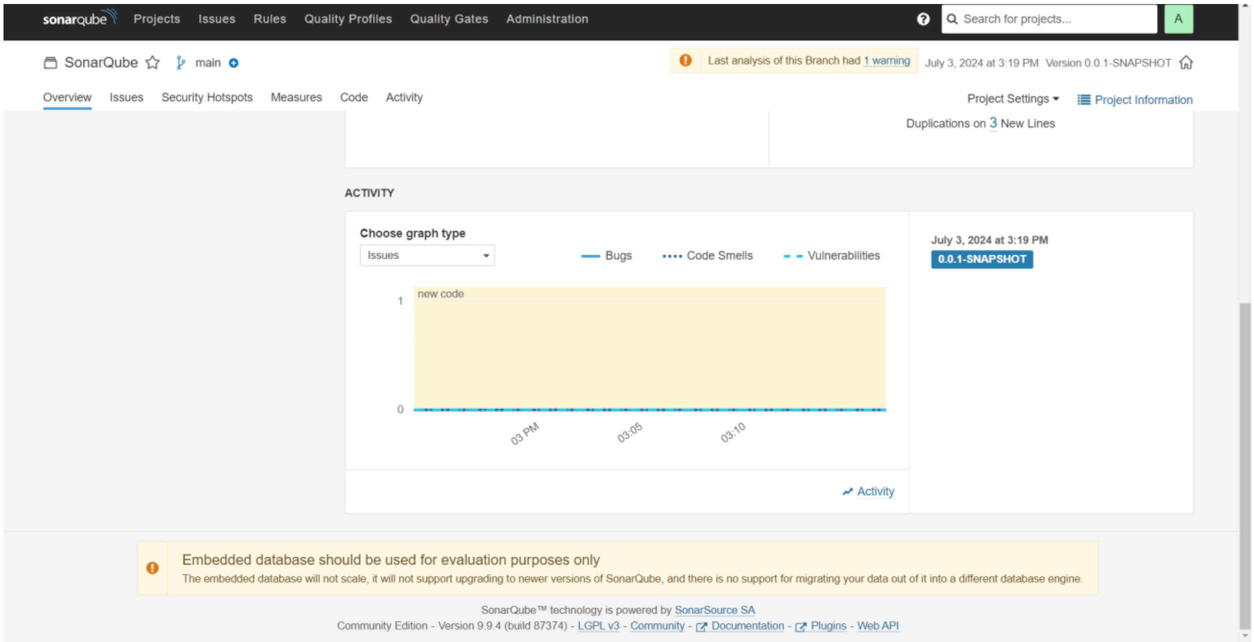
Overall Code

0 New Bugs Reliability A

0 New Vulnerabilities Security A

0 New Security Hotspots — Reviewed Security Review A

0 Added Debt 0 New Code Smells Maintainability A



Last analysis of this Branch had 1 warning July 3, 2024 at 3:19 PM Version 0.0.1-SNAPSHOT

Duplications on 3 New Lines

July 3, 2024 at 3:19 PM

Embedded database should be used for evaluation purposes only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA

Community Edition - Version 9.9.4 (build 87374) - LGPL v3 - Community - Documentation - Plugins - Web API