1.import React, { useState } from 'react';

import './App.css'; // Import CSS for styling

const App = () => {

const [customerName, setCustomerName] = useState('');

const [customerNumber, setCustomerNumber] = useState('');

const [selectedItem, setSelectedItem] = useState('1');

const [quantity, setQuantity] = useState(1);

const [specialInstructions, setSpecialInstructions] = useState('');

const [termsAccepted, setTermsAccepted] = useState(false);

const [orderDetails, setOrderDetails] = useState(null);

const [errors, setErrors] = useState({});

const menuItems = [

{ id: '1', name: 'idli' },

{ id: '2', name: 'Dosa' },

{ id: '3', name: 'Vada' },

];

const handleSubmit = (e) => {

e.preventDefault();

const newErrors = {};

if (customerNumber.length !== 10) newErrors.customerNumber = 'Contact number must be exactly 10 digits.';

if (quantity < 1 || quantity > 3) newErrors.quantity = 'Quantity must be between 1 and 3.';

if (Object.keys(newErrors).length > 0) {

setErrors(newErrors);

return;

}

const order = {

customerName,

customerNumber,

menuItem: menuItems.find(item => item.id === selectedItem).name,

quantity,

specialInstructions,

totalOrder: quantity

};

setOrderDetails(order);

setErrors({});

};

const handleReset = () => {

setCustomerName('');

setCustomerNumber('');

setSelectedItem('1');

setQuantity(1);

setSpecialInstructions('');

setTermsAccepted(false);

setOrderDetails(null);

setErrors({});

};

return (

<div className="App">

<header className="App-header">

<h1>Restaurant Page</h1>

</header>

<main className="container">

<nav>

<ul>

<li><a href="#order">Order Now</a></li>

<li><a href="#gallery">Gallery</a></li>

</ul>

</nav>

<section id="order">

<div className="form-container">

<h1>Restaurant Order Form</h1>

<form onSubmit={handleSubmit}>

<label>

Customer Name:

<input type="text" value={customerName} onChange={(e) => setCustomerName(e.target.value)} />

</label>

<label>

Customer Number:

<input type="text" value={customerNumber} onChange={(e) => setCustomerNumber(e.target.value)} maxLength="10" />

{errors.customerNumber && <p className="error">{errors.customerNumber}</p>}

</label>

<label>

Menu Item:

<select value={selectedItem} onChange={(e) => setSelectedItem(e.target.value)}>

{menuItems.map(item => (

<option key={item.id} value={item.id}>{item.name}</option>

))}

</select>

</label>

<label>

Quantity:

<input type="number" value={quantity} onChange={(e) => setQuantity(Math.min(3, Math.max(1, e.target.value)))} />

{errors.quantity && <p className="error">{errors.quantity}</p>}

</label>

<label>

Special Instructions:

<textarea value={specialInstructions} onChange={(e) => setSpecialInstructions(e.target.value)} />

</label>

{errors.terms && <p className="error">{errors.terms}</p>}

<button type="submit">Order Now</button>

<button type="button" onClick={handleReset}>Reset</button>

</form>

{orderDetails && (

<div className="order-summary">

<h2>Order Details</h2>

<p>Customer Name: {orderDetails.customerName}</p>

<p>Customer Number: {orderDetails.customerNumber}</p>

<p>Menu Item: {orderDetails.menuItem}</p>

<p>Quantity: {orderDetails.quantity}</p>

<p>Special Instructions: {orderDetails.specialInstructions}</p>

<p>Total Order: {orderDetails.totalOrder}</p>

</div>

)}

</div>

</section>

</main>

<footer>

<p>&copy; 2024 Restaurant</p>

</footer>

</div>

);

};

export default App;

App.css

.App {

font-family: Arial, sans-serif;

}

.App-header {

font-family: Arial, sans-serif;

background-color: ;

padding: 10px;

text-align: center;

}

nav ul {

list-style: none;

padding: 0;

margin: 0;

text-align: center;

}

nav ul li {

display: inline;

margin: 0 10px;

}

.container {

max-width: 800px;

margin: 0 auto;

padding: 20px;

}

.form-container {

background-color: hwb(138 0% 37%);

padding: 20px;

color: black;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

label {

display: block;

margin: 10px 0;

}

input[type="text"], input[type="number"], textarea, select {

width: 100%;

padding: 8px;

box-sizing: border-box;

}

button {

padding: 10px 15px;

margin: 10px;

cursor: pointer;

}

.error {

color: red;

}

.order-summary {

border: 1px solid #dddddd;

padding: 10px;

margin-top: 20px;

}

.gallery {

margin-top: 20px;

display: flex;

flex-wrap: wrap;

justify-content: center;

}

.gallery img {

width: 100px;

height: auto;

margin: 5px;

}

footer {

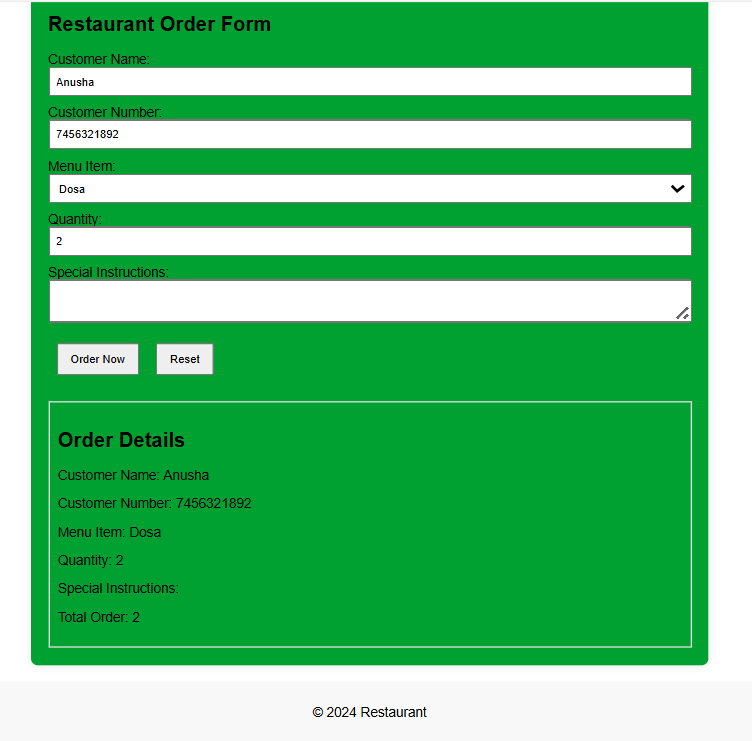
background-color: #f8f8f8;

padding: 10px;

text-align: center;

}

Output



2. import React from 'react';

import Navbar from './components/Navbar';

import Footer from './components/Footer';

import Products from './components/Products';

function App() {

  return (

    <div>

      <Navbar />

      <Products />

      <Footer />

    </div>

  );

}

export default App;

import React from 'react';

import '../App.css';

const Navbar = () => {

  return (

    <div className='navbar'>

      <h1>Shopping App</h1>

    </div>

  );

}

export default Navbar;

import React from 'react';

import '../App.css';

const Footer = () => {

  return (

    <div className='footer'>

      <p>Shopping App 2024&copy;</p>

    </div>

  );

}

export default Footer;

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

import '../App.css';

import axios from 'axios';

const Products = () => {

  const [loading, setLoading] = useState(true);

  const [data, setData] = useState([]);

  useEffect(() => {

    axios.get('https://fakestoreapi.com/products')

      .then(res => {

        setData(res.data);

      })

      .catch(e => console.log(e))

      .finally(() => setLoading(false));

  }, []);

  return (

    <div className='products-container'>

      {loading && <h1>Loading...</h1>}

      {!loading && data.map((product) => (

        <div className="card" key={product.id}>

          <h2>{product.title}</h2>

          {console.log(product.image)} {/\* Log the image URL \*/}

          <img src={product.image} alt={product.title} />

          <p>{product.category}</p>

          <p>${product.price}</p>

        </div>

      ))}

    </div>

  );

};

export default Products;

\* {

  box-sizing: border-box;

  padding: 0;

  margin: 0;

}

.navbar {

  width: 100%;

  height: 10vh;

  background-color: black;

  display: flex;

  align-items: center;

  justify-content: space-between;

  color: white;

  padding: 10px;

  font-weight: bold;

}

.products-container {

  width: 100%;

  min-height: auto;

  display: grid;

  grid-template-columns: repeat(4, 1fr);

  justify-items: center;

  margin-top: 20px;

  margin-bottom: 20px;

}

.card {

  display: flex;

  align-items: center;

  justify-content: center;

  flex-direction: column;

  margin: 20px 0;

}

img {

  width: 250px;

  height: 250px;

  transition: all 0.3s ease;

}

img:hover {

  transform: scale(1.1);

}

.footer {

  width: 100%;

  height: 10vh;

  background-color: black;

  display: flex;

  align-items: center;

  color: white;

  justify-content: center;

  font-size: 19px;

  font-weight: bold;

  margin-top: auto;

}

Output

