App.js

1.restuarant (react)

import React, { useState } from 'react';

import './App.css';

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: 'briyani' },

    { id: '2', name: 'fish fry' },

    { id: '3', name: 'chicken noodles' },

    { id: '3', name: 'fried rice' },

  ];

  const handleSubmit = (e) => {

    e.preventDefault();

    const newErrors = {};

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

    if (quantity < 2 || 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,

      totalOrder: quantity

    };

    setOrderDetails(order);

    setErrors({});

  };

  const handleReset = () => {

    setCustomerName('');

    setCustomerNumber('');

    setSelectedItem('1');

    setQuantity(2);

    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</a></li>

            </ul>

        </nav>

        <section id="order">

          <div className="form-container">

            <h1>Restaurant Order</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(4, Math.max(2, e.target.value)))} />

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

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

              </div>

            )}

          </div>

        </section>

      </main>

</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(75 5% 5%);

  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: rgb(7, 232, 22);

}

.order-summary {

  border: 1px solid #dddddd;

  padding: 10px;

  margin-top: 20px;

}

output



