

```
import React, { useState } from 'react';
import './App.css';

const App = () => {
  const [customerName, setCustomerName] = useState("");
  const [customerNumber, setCustomerNumber] = useState("");
  const [selectedItem, setSelectedItem] = useState('1');
  const [itemSize, setItemSize] = useState('Medium');
  const [quantity, setQuantity] = useState(1);
  const [specialInstructions, setSpecialInstructions] = useState("");
  const [toppings, setToppings] = useState([]);
  const [orderType, setOrderType] = useState('pickup');
  const [termsAccepted, setTermsAccepted] = useState(false);
  const [orderDetails, setOrderDetails] = useState(null);
  const [errors, setErrors] = useState({});

  const menuItems = [
    { id: '1', name: 'Pizza' },
    { id: '2', name: 'Burger' },
    { id: '3', name: 'Pasta' },
  ];

  const images = [
    '/images/pizza.jpg',
    '/images/burger.jpg',
    '/images/pasta.jpg',
  ];

  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 (!termsAccepted) {
  newErrors.terms = 'You must accept the terms and conditions.';
}

if (Object.keys(newErrors).length > 0) {
  setErrors(newErrors);
  return;
}

const order = {
  customerName,
  customerNumber,
  menuItem: menuItems.find(item => item.id === selectedItem).name,
  itemSize,
  quantity,
  toppings,
  specialInstructions,
  orderType,
```

```
    totalOrder: quantity,  
  };
```

```
  setOrderDetails(order);  
  setErrors({});  
};
```

```
const handleReset = () => {  
  setCustomerName("");  
  setCustomerNumber("");  
  setSelectedItem('1');  
  setItemSize('Medium');  
  setQuantity(1);  
  setSpecialInstructions("");  
  setToppings([]);  
  setOrderType('pickup');  
  setTermsAccepted(false);  
  setOrderDetails(null);  
  setErrors({});  
};
```

```
const handleToppingsChange = (e) => {  
  const { value, checked } = e.target;  
  if (checked) {  
    setToppings([...toppings, value]);  
  } else {  
    setToppings(toppings.filter(topping => topping !== value));  
  }  
};
```

```
return (  
  <div className="App">  
    <header className="App-header">  
      <h1>Restaurant Page</h1>  
    </header>  
    <main className="container">  
      <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>

    Size:

    <select
      value={itemSize}
      onChange={(e) => setItemSize(e.target.value)}
    >
      <option value="Small">Small</option>
      <option value="Medium">Medium</option>
      <option value="Large">Large</option>
    </select>
  </label>

  <label>

    Quantity:

    <input

```

```

    type="number"
    value={quantity}
    min="1"
    max="3"
    onChange={(e) => setQuantity(Math.min(3, Math.max(1, e.target.value)))}
  />
  {errors.quantity && (
    <p className="error">{errors.quantity}</p>
  )}
</label>
<label>
  Extra Toppings:
  <div className="toppings">
    <label>
      <input
        type="checkbox"
        value="Cheese"
        checked={toppings.includes('Cheese')}
        onChange={handleToppingsChange}
      />
      Cheese
    </label>
    <label>
      <input
        type="checkbox"
        value="Pepperoni"
        checked={toppings.includes('Pepperoni')}
        onChange={handleToppingsChange}
      />

```

Pepperoni

</label>

<label>

<input

type="checkbox"

value="Mushrooms"

checked={toppings.includes('Mushrooms')}

onChange={handleToppingsChange}

/>

Mushrooms

</label>

</div>

</label>

<label>

Order Type:

<div className="order-type">

<label>

<input

type="radio"

value="pickup"

checked={orderType === 'pickup'}

onChange={(e) => setOrderType(e.target.value)}

/>

Pickup

</label>

<label>

<input

type="radio"

value="delivery"

```

        checked={orderType === 'delivery'}
        onChange={(e) => setOrderType(e.target.value)}
      />
      Delivery
    </label>
  </div>
</label>
<label>
  Special Instructions:
  <textarea
    value={specialInstructions}
    onChange={(e) => setSpecialInstructions(e.target.value)}
  />
</label>
<label>
  <input
    type="checkbox"
    checked={termsAccepted}
    onChange={(e) => setTermsAccepted(e.target.checked)}
  />
  I accept the terms and conditions
</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-modal">

  <div className="modal-content">

    <h2>Order Details</h2>

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

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

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

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

    <p>Toppings: {orderDetails.toppings.join(', ')}</p>

    <p>Order Type: {orderDetails.orderType}</p>

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

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

  </div>

</div>

  )}

</div>

</section>

<section id="gallery">

  <h2>Popular Dishes</h2>

  <div className="gallery">

    {images.map((src, index) => (

      <img key={index} src={src} alt={`Dish ${index + 1}`} />

    ))}

  </div>

</section>

</main>

<footer>

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

</footer>

</div>
```

```
);  
};
```

```
export default App;
```

App.css

```
body {  
  font-family: 'Roboto', sans-serif;  
  background-color: #e0e4e7;  
  color: #444;  
  margin: 0;  
  padding: 0;  
}
```

```
.App-header {  
  background-color: #1a3c5a;  
  color: #ffffff;  
  padding: 25px;  
  text-align: center;  
  border-bottom: 4px solid #f5b041;  
}
```

```
.container {  
  padding: 30px;  
  max-width: 1000px;  
  margin: 0 auto;  
  background-color: #ffffff;
```

```
border-radius: 12px;
box-shadow: 0 10px 20px rgba(0, 0, 0, 0.15);
}
```

```
nav ul {
  list-style: none;
  padding: 0;
  display: flex;
  justify-content: center;
  margin-bottom: 25px;
}
```

```
nav ul li {
  margin: 0 15px;
}
```

```
nav ul li a {
  text-decoration: none;
  color: #f5b041;
  font-weight: 500;
  font-size: 1rem;
  transition: color 0.3s ease;
}
```

```
nav ul li a:hover {
  color: #d35400;
}
```

```
.form-container {
```

```
background-color: #ffffff;
border-radius: 8px;
padding: 20px;
box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);
transition: transform 0.3s ease, box-shadow 0.3s ease;
}
```

```
.form-container:hover {
  transform: translateY(-3px);
  box-shadow: 0 8px 20px rgba(0, 0, 0, 0.15);
}
```

```
.form-container h1 {
  margin-top: 0;
  margin-bottom: 20px;
  color: #1a3c5a;
  font-size: 1.7rem;
  text-align: center;
}
```

```
label {
  display: block;
  margin-bottom: 8px;
  font-weight: 600;
  color: #2c3e50;
}
```

```
input[type="text"],
input[type="number"],
```

```
select,
textarea {
  width: 100%;
  padding: 10px;
  margin-top: 5px;
  margin-bottom: 20px;
  border: 1px solid #ced4da;
  border-radius: 4px;
  box-sizing: border-box;
  transition: border-color 0.3s ease;
}
```

```
input[type="text"]:focus,
input[type="number"]:focus,
select:focus,
textarea:focus {
  border-color: #f5b041;
}
```

```
input[type="checkbox"],
input[type="radio"] {
  margin-right: 6px;
}
```

```
textarea {
  height: 90px;
}
```

```
button[type="submit"],
```

```
button[type="button"] {  
  padding: 10px 20px;  
  background-color: #f5b041;  
  color: white;  
  border: none;  
  border-radius: 4px;  
  cursor: pointer;  
  font-size: 0.95rem;  
  transition: background-color 0.3s ease, transform 0.3s ease;  
  margin-top: 12px;  
}
```

```
button[type="button"] {  
  background-color: #7f8c8d;  
}
```

```
button:hover {  
  background-color: #d35400;  
  transform: translateY(-2px);  
}
```

```
button[type="button"]:hover {  
  background-color: #99a3a4;  
}
```

```
.error {  
  color: #d35400;  
  font-size: 0.875rem;  
  margin-top: -8px;
```

```
margin-bottom: 12px;  
}
```

```
.gallery {  
  display: grid;  
  grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));  
  gap: 15px;  
  margin-top: 18px;  
}
```

```
.gallery img {  
  width: 100%;  
  height: auto;  
  border-radius: 8px;  
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);  
  transition: transform 0.3s ease, box-shadow 0.3s ease;  
}
```

```
.gallery img:hover {  
  transform: scale(1.03);  
  box-shadow: 0 6px 15px rgba(0, 0, 0, 0.15);  
}
```

```
.order-summary-modal {  
  position: fixed;  
  top: 0;  
  left: 0;  
  width: 100%;  
  height: 100%;
```

```
background-color: rgba(0, 0, 0, 0.5);  
display: flex;  
align-items: center;  
justify-content: center;  
}
```

```
.modal-content {  
  background-color: white;  
  padding: 25px;  
  border-radius: 8px;  
  max-width: 480px;  
  width: 100%;  
  box-shadow: 0 8px 20px rgba(0, 0, 0, 0.2);  
}
```

```
.modal-content h2 {  
  margin-top: 0;  
  color: #1a3c5a;  
  font-size: 1.4rem;  
}
```

```
footer {  
  background-color: #1a3c5a;  
  color: #ffffff;  
  text-align: center;  
  padding: 15px;  
  position: fixed;  
  bottom: 0;  
  width: 100%;  
}
```



```
box-shadow: 0 -3px 8px rgba(0, 0, 0, 0.1);
font-size: 0.85rem;
}
```

Output:

Restaurant Page

Restaurant Order Form

Customer Name:

Shivani

Customer Number:

7675443222

Menu Item:

Pasta

Size:

Small

Quantity:

1

Order Type:

☒ Pickup

☐ Delivery

Special Instructions:

☒ I accept the terms and conditions

Order Now

Reset

© 2024 Restaurant