

**index.js:**

```
import React from 'react';  
import ReactDOM from 'react-dom';  
import './styles.css'; // Import the CSS file  
import App from './App'; // Import the App component
```

```
ReactDOM.render(  
  <React.StrictMode>  
    <App />  
  </React.StrictMode>,  
  document.getElementById('root')  
);
```

**App.js:**

```
import React, { useState } from 'react';  
  
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: '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,  
    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)}
```

```
placeholder="Enter your name"
```

```
/>
```

```
</label>
```

```
<label>
```

Customer Number:

```
<input
  type="text"
  value={customerNumber}
  onChange={(e) => setCustomerNumber(e.target.value)}
  maxLength="10"
  placeholder="Enter your contact number"
/>

{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)))}
    min="1"
    max="3"
  />
  {errors.quantity && <p className="error">{errors.quantity}</p>}
</label>
```

```
<label>
  Special Instructions:
  <textarea
    value={specialInstructions}
    onChange={(e) => setSpecialInstructions(e.target.value)}
    placeholder="Enter any special instructions"
  />
</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">
    <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>
```

```
<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;
```

```
Styles.css:
```

```
.App {
```

```
  font-family: Arial, sans-serif;
```

```
  color: #333;
```

```
}
```

```
.App-header {
```

```
  background-color: #4CAF50;
```

```
  padding: 10px;
```

```
  text-align: center;
```

```
  color: white;
```

```
}
```

```
nav ul {
```

```
  list-style: none;
```

```
  text-align: center;
```



```
padding: 0;  
}
```

```
nav ul li {  
    display: inline;  
    margin: 0 8px;  
}
```

```
nav ul li a {  
    text-decoration: none;  
    color: #4CAF50;  
}
```

```
.container {  
    padding: 10px;  
}
```

```
.form-container {  
    padding: 15px;  
    background-color: #f9f9f9;  
    border: 1px solid #ddd;  
    border-radius: 4px;  
}
```

```
label {  
    margin: 8px 0;
```

```
display: block;  
}
```

```
input, textarea, select {  
    width: 100%;  
    padding: 8px;  
    margin-top: 5px;  
    border: 1px solid #ccc;  
    border-radius: 3px;  
}
```

```
button {  
    padding: 8px 12px;  
    margin: 8px 5px;  
    border: none;  
    border-radius: 3px;  
    background-color: #4CAF50;  
    color: white;  
    cursor: pointer;  
}
```

```
.error {  
    color: red;  
    font-size: 12px;  
}
```

```
footer{  
  
  background-color: #f1f1f1;  
  
  padding: 8px;  
  
  text-align: center;  
  
}
```

## Output:

