

# Chapter 8: Forms in React

## 1 Introduction

Is chapter mein humne **forms** aur **controlled components** ke baare mein seekha. React mein forms ke inputs ka value state ke saath sync hota hai, jisse user input handle karna asaan hota hai.

## 2 8.1 What are Controlled Components?

**Controlled components** form elements hote hain jinke value ko React ka state control karta hai. Har input ka value state mein store hota hai, aur **onChange** se update hota hai.

### 2.1 Why Controlled Components?

- Input tracking asaan hota hai.
- Validation aur conditional logic apply kar sakte hain.
- Form data manipulate karna simple hai.

### 2.2 Real-World Example

Restaurant app mein order form jahan user naam aur food item select karta hai.

## 3 8.2 Handling Text Input

Text inputs ko value aur **onChange** se controlled banate hain.

### 3.1 Code Example: Text Input

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Text Input Form</title>
6   <script
7     src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js
8   <script
9     src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.developo
10  <script
11    src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js
12  <script src="https://cdn.tailwindcss.com"></script>
13 </head>
14 <body>
15   <div id="root"></div>
16   <script type="text/babel">
17     function OrderForm() {
18       const [name, setName] = React.useState('');
```

```

16
17     const handleChange = (event) => {
18         setName(event.target.value);
19     };
20
21     const handleSubmit = (event) => {
22         event.preventDefault();
23         alert('Order placed for: ${name}');
24     };
25
26     return (
27         <div className="text-center p-4">
28             <h1 className="text-3xl font-bold text-blue-600">Order
29                 Form</h1>
30             <form onSubmit={handleSubmit}>
31                 <input
32                     type="text"
33                     value={name}
34                     onChange={handleChange}
35                     placeholder="Enter your name"
36                     className="border p-2 m-2 rounded"
37                 />
38                 <button
39                     type="submit"
40                     className="bg-green-500 text-white px-4 py-2 m-2
41                         rounded"
42                 >
43                     Submit
44                 </button>
45             </form>
46             <p className="text-lg">Name: {name}</p>
47         </div>
48     );
49
50     const root =
51         ReactDOM.createRoot(document.getElementById('root'));
52     root.render(<OrderForm />);
53 </script>
</body>
</html>

```

## 3.2 Explanation

- `useState`: name state input value store karta hai.
- `value={name}`: Input state se sync.
- `onChange`: User input se state update.
- `onSubmit`: Form submit pe alert.

- Output: Name type karne pe display, submit pe alert.

## 4 8.3 Handling Multiple Inputs

Multiple inputs ke liye object state aur name attribute use hota hai.

### 4.1 Code Example: Multiple Inputs

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Multiple Inputs Form</title>
6   <script
7     src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js
8   <script
9     src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.developo
10  <script
11    src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js
12  <script src="https://cdn.tailwindcss.com"></script>
13 </head>
14 <body>
15   <div id="root"></div>
16   <script type="text/babel">
17     function OrderForm() {
18       const [formData, setFormData] = React.useState({
19         name: '',
20         address: ''
21       });
22
23       const handleChange = (event) => {
24         const { name, value } = event.target;
25         setFormData({ ...formData, [name]: value });
26       };
27
28       const handleSubmit = (event) => {
29         event.preventDefault();
30         alert('Order placed for ${formData.name} at
31           ${formData.address}');
32       };
33
34       return (
35         <div className="text-center p-4">
36           <h1 className="text-3xl font-bold text-blue-600">Order
37             Form</h1>
38           <form onSubmit={handleSubmit}>
39             <input
40               type="text"
41               name="name"
42               value={formData.name}
43               onChange={handleChange}

```

```

39         placeholder="Enter your name"
40         className="border p-2 m-2 rounded"
41     />
42     <input
43         type="text"
44         name="address"
45         value={formData.address}
46         onChange={handleChange}
47         placeholder="Enter your address"
48         className="border p-2 m-2 rounded"
49     />
50     <button
51         type="submit"
52         className="bg-green-500 text-white px-4 py-2 m-2
53             rounded"
54     >
55         Submit
56     </button>
57 </form>
58 <p className="text-lg">Name: {formData.name}</p>
59 <p className="text-lg">Address: {formData.address}</p>
60 </div>
61 );
62 }
63
64 const root Kost =
65     ReactDOM.createRoot(document.getElementById('root'));
66     root.render(<OrderForm />);
67 </script>
68 </body>
69 </html>
70 \end{lstlisting}
71
72 \subsection{Explanation}
73 \begin{itemize}
74     \item \texttt{formData}: Object state \texttt{name} aur
75         \texttt{address} store karta hai.
76     \item \texttt{handleChange}: \texttt{name} attribute se field
77         update karta hai.
78     \item Output: Name aur address display, submit pe alert.
79 \end{itemize}
80
81 \section{8.4 Handling Checkboxes}
82 Checkboxes ke liye \texttt{checked} attribute aur boolean state
83 use hota hai.
84
85 \subsection{Code Example: Checkbox}
86 \begin{lstlisting}[language=HTML]
87 <!DOCTYPE html>
88 <html lang="en">
89 <head>

```

```

85 <meta charset="UTF-8">
86 <title>Checkbox Form</title>
87 <script
    src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js
88 <script
    src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.develop
89 <script
    src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js
90 <script src="https://cdn.tailwindcss.com"></script>
91 </head>
92 <body>
93 <div id="root"></div>
94 <script type="text/babel">
95     function CheckboxForm() {
96         const [isVeg, setIsVeg] = React.useState(false);
97
98         const handleChange = (event) => {
99             setIsVeg(event.target.checked);
100         };
101
102         const handleSubmit = (event) => {
103             event.preventDefault();
104             alert('Preference: ${isVeg ? 'Vegetarian' :
                'Non-Vegetarian'}');
105         };
106
107         return (
108             <div className="text-center p-4">
109                 <h1 className="text-3xl font-bold text-blue-600">Food
                    Preference</h1>
110                 <form onSubmit={handleSubmit}>
111                     <label className="flex items-center justify-center
                        m-2">
112                         <input
113                             type="checkbox"
114                             checked={isVeg}
115                             onChange={handleChange}
116                             className="mr-2"
117                         />
118                         Vegetarian
119                     </label>
120                     <button
121                         type="submit"
122                         className="bg-green-500 text-white px-4 py-2 m-2
                            rounded"
123                     >
124                         Submit
125                     </button>
126                 </form>
127                 <p className="text-lg">
                    Preference: {isVeg ? 'Vegetarian' : 'Non-Vegetarian'}
128

```

```

129         </p>
130     </div>
131 );
132 }
133
134 const root =
135     ReactDOM.createRoot(document.getElementById('root'));
136     root.render(<CheckboxForm />);
137 </script>
138 </body>
</html>

```

## 4.2 Explanation

- isVeg: Boolean state checkbox ke status ko track karta hai.
- checked={isVeg}: Checkbox state se sync.
- Output: Checkbox check/uncheck pe preference display, submit pe alert.

## 5 8.5 Handling Select Dropdowns

Dropdowns ke liye value aur onChange use hota hai.

### 5.1 Code Example: Select Dropdown

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Select Form</title>
6   <script
7     src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js"
8   </script>
9   <script
10    src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.development.js"
11  </script>
12  <script
13    src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js"
14  </script>
15  <script src="https://cdn.tailwindcss.com"></script>
16 </head>
17 <body>
18   <div id="root"></div>
19   <script type="text/babel">
20     function FoodSelectForm() {
21       const [food, setFood] = React.useState('');
22
23       const handleChange = (event) => {
24         setFood(event.target.value);
25       };
26
27       const handleSubmit = (event) => {
28         event.preventDefault();

```

```

23     alert('Selected food: ${food}');
24 };
25
26 return (
27     <div className="text-center p-4">
28         <h1 className="text-3xl font-bold
29             text-blue-600">Select Food</h1>
30         <form onSubmit={handleSubmit}>
31             <select
32                 value={food}
33                 onChange={handleChange}
34                 className="border p-2 m-2 rounded"
35             >
36                 <option value="" disabled>Select a food</option>
37                 <option value="Chai">Chai</option>
38                 <option value="Samosa">Samosa</option>
39                 <option value="Idli">Idli</option>
40             </select>
41             <button
42                 type="submit"
43                 className="bg-green-500 text-white px-4 py-2 m-2
44                     rounded"
45             >
46                 Submit
47             </button>
48         </form>
49         <p className="text-lg">Selected: {food || 'None'}</p>
50     </div>
51 );
52
53 }
54
55 const root =
56     ReactDOM.createRoot(document.getElementById('root'));
57 root.render(<FoodSelectForm />);
58 </script>
59 </body>
60 </html>

```

## 5.2 Explanation

- food: State dropdown value store karta hai.
- value={food}: Dropdown state se sync.
- Output: Selected food display, submit pe alert.

## 6 Common Mistakes

- Forgetting `event.preventDefault()`: Page refresh ho jayega.
- Not Syncing Value: value state se nahi joda toh uncontrolled input.

- **Missing name in Multiple Inputs:** `handleChange` ke liye zaroori.

## 7 Interview Tips

- **Controlled components kya hain?**
  - Form elements jinke value state se control hote hain.
- **Uncontrolled vs Controlled?**
  - Controlled: State se sync, predictable.
  - Uncontrolled: DOM se value, kam common.
- **Form submission kaise handle karte hain?**
  - `onSubmit` aur `event.preventDefault()` se.

## 8 Assignment: Practice Time

### 8.1 Task 1: Simple Text Input Form

- `UserForm` component banao jo `useState` se name track kare.
- Input value display aur submit pe alert: `Hello, [name]!`.
- Tailwind se style karo.

### 8.2 Task 2: Multiple Input Form

- `OrderForm` component banao jo `useState` se name aur phone track kare.
- Values display aur submit pe alert: `Order for [name], Phone: [phone]`.
- Tailwind se style karo.

### 8.3 Task 3: Checkbox and Dropdown

- `FoodPreference` component banao jo checkbox (`isVeg`) aur dropdown track kare.
- Values display karo.
- Submit pe alert: `Preference: [isVeg ? 'Vegetarian' : 'Non-Veg'], Food: [food]`.
- Tailwind se style karo.