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

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Text Input Form</title>
  <script
     src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js
  <script
     src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.develo
  <script
     src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body>
  <div id="root"></div>
  <script type="text/babel">
    function OrderForm() {
      const [name, setName] = React.useState('');

      const handleChange = (event) => {
        setName(event.target.value);
      };

      const handleSubmit = (event) => {
        event.preventDefault();
        alert(‘Order placed for: ${name}‘);
      };

      return (
        <div className="text-center p-4">
          <h1 className="text-3xl font-bold text-blue-600">Order Form</h1>
          <form onSubmit={handleSubmit}>
            <input
              type="text"
              value={name}
              onChange={handleChange}
              placeholder="Enter your name"
              className="border p-2 m-2 rounded"
            />
            <button
              type="submit"
              className="bg-green-500 text-white px-4 py-2 m-2 rounded"
            >
              Submit
            </button>
          </form>
          <p className="text-lg">Name: {name}</p>
        </div>
      );
    }

    const root = ReactDOM.createRoot(document.getElementById(’root’));
    root.render(<OrderForm />);
  </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

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Multiple Inputs Form</title>
  <script
     src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js
  <script
     src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.develo
  <script
     src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body>
  <div id="root"></div>
  <script type="text/babel">
    function OrderForm() {
      const [formData, setFormData] = React.useState({
        name: ’’,
        address: ’’
      });

      const handleChange = (event) => {
        const { name, value } = event.target;
        setFormData({ ...formData, [name]: value });
      };

      const handleSubmit = (event) => {
        event.preventDefault();
        alert(‘Order placed for ${formData.name} at
           ${formData.address}‘);
      };

      return (
        <div className="text-center p-4">
          <h1 className="text-3xl font-bold text-blue-600">Order
             Form</h1>
          <form onSubmit={handleSubmit}>
            <input
              type="text"
              name="name"
              value={formData.name}
              onChange={handleChange}
              placeholder="Enter your name"
              className="border p-2 m-2 rounded"
            />
            <input
              type="text"
              name="address"
              value={formData.address}
              onChange={handleChange}
              placeholder="Enter your address"
              className="border p-2 m-2 rounded"
            />
            <button
              type="submit"
              className="bg-green-500 text-white px-4 py-2 m-2 rounded"
            >
              Submit
            </button>
          </form>
          <p className="text-lg">Name: {formData.name}</p>
          <p className="text-lg">Address: {formData.address}</p>
        </div>
      );
    }

    const root Kost = ReactDOM.createRoot(document.getElementById(’root’));
    root.render(<OrderForm />);
  </script>
</body>
</html>
\end{lstlisting}

\subsection{Explanation}
\begin{itemize}
  \item \texttt{formData}: Object state \texttt{name} aur \texttt{address} store karta hai.
  \item \texttt{handleChange}: \texttt{name} attribute se field update karta hai.
  \item Output: Name aur address display, submit pe alert.
\end{itemize}

\section{8.4 Handling Checkboxes}
Checkboxes ke liye \texttt{checked} attribute aur boolean state use hota hai.

\subsection{Code Example: Checkbox}
\begin{lstlisting}[language=HTML]
<!DOCTYPE html>
<html lang="en">
<head>
```

```
  <meta charset="UTF-8">
  <title>Checkbox Form</title>
  <script src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js
  <script src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.develo
  <script src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body>
  <div id="root"></div>
  <script type="text/babel">
    function CheckboxForm() {
      const [isVeg, setIsVeg] = React.useState(false);

      const handleChange = (event) => {
        setIsVeg(event.target.checked);
      };

      const handleSubmit = (event) => {
        event.preventDefault();
        alert(‘Preference: ${isVeg ? ’Vegetarian’ : ’Non-Vegetarian’}‘);
      };

      return (
        <div className="text-center p-4">
          <h1 className="text-3xl font-bold text-blue-600">Food Preference</h1>
          <form onSubmit={handleSubmit}>
            <label className="flex items-center justify-center m-2">
              <input
                type="checkbox"
                checked={isVeg}
                onChange={handleChange}
                className="mr-2"
              />
              Vegetarian
            </label>
            <button
              type="submit"
              className="bg-green-500 text-white px-4 py-2 m-2 rounded"
            >
              Submit
            </button>
          </form>
          <p className="text-lg">
            Preference: {isVeg ? ’Vegetarian’ : ’Non-Vegetarian’}
          </p>
        </div>
      );
    }

    const root = ReactDOM.createRoot(document.getElementById('root'));
    root.render(<CheckboxForm />);
  </script>
</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

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Select Form</title>
  <script src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js
  <script src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.develo
  <script src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body>
  <div id="root"></div>
  <script type="text/babel">
    function FoodSelectForm() {
      const [food, setFood] = React.useState('');

      const handleChange = (event) => {
        setFood(event.target.value);
      };

      const handleSubmit = (event) => {
        event.preventDefault();
        alert(`Selected food: ${food}`);
      };

      return (
        <div className="text-center p-4">
          <h1 className="text-3xl font-bold text-blue-600">Select Food</h1>
          <form onSubmit={handleSubmit}>
            <select
              value={food}
              onChange={handleChange}
              className="border p-2 m-2 rounded"
            >
              <option value="" disabled>Select a food</option>
              <option value="Chai">Chai</option>
              <option value="Samosa">Samosa</option>
              <option value="Idli">Idli</option>
            </select>
            <button
              type="submit"
              className="bg-green-500 text-white px-4 py-2 m-2 rounded"
            >
              Submit
            </button>
          </form>
          <p className="text-lg">Selected: {food || 'None'}</p>
        </div>
      );
    }

    const root = ReactDOM.createRoot(document.getElementById('root'));
    root.render(<FoodSelectForm />);
  </script>
</body>
</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.