Forms and Two-Way Binding

In this lecture, we will learn how to handle forms in React. Forms allow users to input data, and with two-way binding, we can keep the input value in sync with component state.

Controlled Components

In React, form inputs are usually controlled by state. This means the value of the input is stored in state and updated on every change.

Example with a text input:

```
import { useState } from "react"

function App() {
  const [name, setName] = useState("")

  function handleChange(event) {
    setName(event.target.value)
  }

  return (
    <div>
      <input type="text" value={name} onChange={handleChange} />
      <p>Your name is: {name}</p>
    </div>
  )
}

export default App
```

Here:

- `value={name}` means the input’s value comes from state.

- `onChange` updates the state whenever the user types.

This is called two-way binding: the input updates the state, and the state updates the input.

Handling Multiple Inputs

For multiple inputs, you can have separate state variables:

```
import { useState } from "react"

function LoginForm() {
  const [email, setEmail] = useState("")
  const [password, setPassword] = useState("")

  function handleSubmit(event) {
    event.preventDefault()
    alert(`Email: ${email}, Password: ${password}`)
  }

  return (
    <form onSubmit={handleSubmit}>
      <div>
        <label>Email: </label>
        <input type="email" value={email} onChange={e =>
setEmail(e.target.value)} />
      </div>
      <div>
        <label>Password: </label>
        <input type="password" value={password} onChange={e =>
setPassword(e.target.value)} />
      </div>
      <button type="submit">Login</button>
    </form>
  )
}

export default LoginForm
```

The `handleSubmit` function prevents the page from refreshing and shows the input values in an alert.

Recap

- Controlled components keep form inputs in sync with state.
- Two-way binding means user input updates state, and state updates the input field.
- Forms in React usually require event handlers for `onChange` and `onSubmit`.