Day 3: State & Event Handling in React

Today, you'll learn about **state management** using `useState` and handle **user interactions** with events in React.

---

1 What is State in React?

State is a built-in object that **stores dynamic data** and **updates the UI** when changed.

📌 **Difference Between Props & State**

Feature	Props	State
Data Flow	Parent → Child	Within the Component
Mutability	Immutable (cannot change)	Mutable (can change)
Usage	Pass data to child	Manage local component data

---

2 Using `useState` for State Management

📌 **First, import `useState` from React**

```
import { useState } from "react";
```

Example: Counter App

```
import { useState } from "react";

function Counter() {
  const [count, setCount] = useState(0); // State variable

  return (
    <div>
      <h2>Counter: {count}</h2>
      <button onClick={() => setCount(count + 1)}>Increment</button>
      <button onClick={() => setCount(count - 1)}>Decrement</button>
    </div>
  );
}

export default Counter;
```

📌 **Breakdown:**

- `useState(0)` initializes **count** with `0`.
- `setCount(count + 1)` updates state and re-renders UI.
- Clicking **buttons** updates the counter.

✅ **State updates automatically trigger a re-render!**

3️⃣ Handling Events in React

React uses **event handlers** to handle user interactions like clicks, input changes, etc.

📌 **Example: Click Event**

```jsx
function ButtonClick() {
  function handleClick() {
    alert("Button Clicked!");
  }

  return <button onClick={handleClick}>Click Me</button>;
}

export default ButtonClick;
```

✅ **Notice:** No `()` after `handleClick` in `onClick`.

4️⃣ Updating State with Input Events

📌 **Example: Controlled Input Field**

```jsx
import { useState } from "react";

function TextInput() {
  const [text, setText] = useState("");

  return (
    <div>
      <input type="text" onChange={(e) => setText(e.target.value)} />
      <p>Entered Text: {text}</p>
    </div>
  );
}

export default TextInput;
```

✅ **Live updates as you type!**

5️⃣ Combining State & Events: Toggle Theme App

📌 Example: Light/Dark Mode

```
import { useState } from "react";

function ThemeToggle() {
  const [isDark, setIsDark] = useState(false);

  return (
    <div style={{ background: isDark ? "#333" : "#fff", color: isDark ? "#fff" : "#
000", padding: "20px" }}>
      <h2>Theme: {isDark ? "Dark" : "Light"}</h2>
      <button onClick={() => setIsDark(!isDark)}>Toggle Theme</button>
    </div>
  );
}

export default ThemeToggle;
```

✅ **Clicking the button toggles between Dark & Light mode!**

📌 Summary of Day 3

- ✔ **State** stores dynamic data
- ✔ `useState` manages **component state**
- ✔ **Events** handle user interactions
- ✔ **Controlled Inputs** update state on change
- ✔ **Built a Toggle Theme App**

🚀 Next Step: Day 4 - Conditional Rendering & Lists