**⚛️ Day 4 – useState, Events & Reactivity**

**🎯 GOAL:**

* Understand how to make **dynamic + interactive** apps
* Learn React’s most important hook: useState
* Handle **click events** (like count, toggle, etc.)
* Build: **“Like Button” App** & **“Show/Hide” Toggle**

**🔹 1. What is useState?**

React doesn’t change UI magically. You have to **store the changing value in “state.”**

jsx

CopyEdit

const [count, setCount] = useState(0);

This means:

* count → current value
* setCount() → how to **update** the value (and trigger UI update)

🧠 **useState** is React’s way of making variables that **update the screen** when changed.

**🔸 2. Let’s Build: “Like Button” App**

**✅ App.jsx Code:**

jsx

CopyEdit

import { useState } from "react";

function App() {

const [likes, setLikes] = useState(0);

const handleLike = () => {

setLikes(likes + 1);

};

return (

<div>

<h1>👍 Likes: {likes}</h1>

<button onClick={handleLike}>Like +</button>

</div>

);

}

export default App;

**🎉 What’s happening:**

* You click → handleLike runs
* setLikes() updates the value
* React **re-renders** the UI with the new value

**🔹 3. Toggle Visibility Example**

Let’s make a **Show/Hide Jugaad Card**:

jsx

CopyEdit

import { useState } from "react";

function App() {

const [show, setShow] = useState(true);

return (

<div>

<button onClick={() => setShow(!show)}>

{show ? "Hide" : "Show"} Jugaad

</button>

{show && (

<div style={{ padding: "10px", marginTop: "10px", background: "#e0f7fa" }}>

<h3>🧠 Jugaad of the Day:</h3>

<p>Use `map()` to render multiple React cards with just one component.</p>

</div>

)}

</div>

);

}

export default App;

✅ This is **conditional rendering** with a **toggle** button.  
Aise apps me interactivity aata hai — users love it.

**🎯 Challenge for Today:**

**🔸 “Counter App”:**

* Add "➕ Increase" and "➖ Decrease" buttons
* Prevent it from going below zero (optional)

**🔸 “Dark Mode Toggle”:**

* Change background color of page when toggle is clicked
* Use a darkMode state (true/false)
* Add emoji: 🌞 / 🌙 depending on mode

**🧠 Smart Tip of the Day:**

useState() should always be called at the **top level of the component**, not inside conditions or loops.

Also:

jsx

CopyEdit

<button onClick={functionHere}>Click</button> // ✅

<button onClick={functionHere()}>Click</button> // ❌ This runs immediately!

**🧪 Summary:**

| **Concept** | **What it does** |
| --- | --- |
| useState() | Add state to component |
| Event handler | onClick={...} to handle clicks |
| setState() | Triggers re-render with new value |
| Conditional rendering | {state && <Component />} or ternary state ? A : B |