**⚛️ Day 5 – Controlled Components + Forms + Input Handling**

**🎯 GOAL:**

* Understand **controlled components** (very important term!)
* Use useState to control **input fields**
* Build: Login Form, Live Greeting, Simple Search Filter

**🔹 1. What is a Controlled Component?**

When an input field’s value is **fully controlled by React state** — it’s called a **Controlled Component**.

🧠 Example:

jsx

CopyEdit

const [username, setUsername] = useState("");

<input value={username} onChange={(e) => setUsername(e.target.value)} />

➡️ Now, whatever you type in the input is reflected in username, and vice versa.  
React becomes **the boss** of the form!

**🔸 2. Example: Live Greeting Form**

**✅ App.jsx:**

jsx

CopyEdit

import { useState } from "react";

function App() {

const [name, setName] = useState("");

return (

<div style={{ textAlign: "center", marginTop: "40px" }}>

<h2>👋 Live Greeting</h2>

<input

type="text"

placeholder="Enter your name"

value={name}

onChange={(e) => setName(e.target.value)}

style={{ padding: "10px", fontSize: "1rem" }}

/>

<p style={{ fontSize: "1.2rem", marginTop: "20px" }}>

{name ? `Namaste, ${name}!` : "Type your name above 👆"}

</p>

</div>

);

}

export default App;

✅ This is the simplest example of controlled components.

**🔸 3. Mini Project: Login Form**

jsx

CopyEdit

import { useState } from "react";

function App() {

const [email, setEmail] = useState("");

const [password, setPassword] = useState("");

const [submitted, setSubmitted] = useState(false);

const handleSubmit = (e) => {

e.preventDefault(); // prevent page reload

setSubmitted(true);

};

return (

<div style={{ maxWidth: "300px", margin: "40px auto", padding: "20px", border: "1px solid #ccc" }}>

<h2>🔐 Login</h2>

<form onSubmit={handleSubmit}>

<input

type="email"

placeholder="Email"

value={email}

onChange={(e) => setEmail(e.target.value)}

style={{ width: "100%", marginBottom: "10px", padding: "8px" }}

/>

<input

type="password"

placeholder="Password"

value={password}

onChange={(e) => setPassword(e.target.value)}

style={{ width: "100%", marginBottom: "10px", padding: "8px" }}

/>

<button type="submit" style={{ width: "100%", padding: "10px" }}>

Login

</button>

</form>

{submitted && (

<p style={{ marginTop: "20px" }}>

👋 Welcome, user with email <b>{email}</b>

</p>

)}

</div>

);

}

export default App;

**🔸 4. Challenge: Simple Search Filter**

jsx

CopyEdit

const fruits = ["Apple", "Banana", "Mango", "Orange", "Papaya"];

function App() {

const [search, setSearch] = useState("");

const filteredFruits = fruits.filter((fruit) =>

fruit.toLowerCase().includes(search.toLowerCase())

);

return (

<div style={{ textAlign: "center", marginTop: "40px" }}>

<h2>🍎 Fruit Search</h2>

<input

type="text"

placeholder="Search fruits..."

value={search}

onChange={(e) => setSearch(e.target.value)}

style={{ padding: "10px", fontSize: "1rem" }}

/>

<ul style={{ listStyle: "none", padding: 0 }}>

{filteredFruits.map((fruit, index) => (

<li key={index}>{fruit}</li>

))}

</ul>

</div>

);

}

**🧠 Smart Tip of the Day:**

Don’t use .value from DOM directly.  
Always use value={...} + onChange={...} — that’s the **React way**.

And don’t forget:

* e.preventDefault() on form submit
* Use trim() to clean inputs if needed

**🔥 Summary of Day 5**

| **Concept** | **Example** |
| --- | --- |
| useState for inputs | const [value, setValue] = useState("") |
| Controlled component | value={value} onChange={...} |
| Form submit | onSubmit={handleSubmit} |
| Search/filtering | array.filter(...includes(...)) |