

Alright 🙌 let's go step-by-step and explain **Form Handling** in React with **detailed code comments** so you clearly see what's happening.

🔥 Form Handling in React

When you submit a form in plain HTML, the browser **refreshes the page** by default.

In React, we usually **stop that behavior** so we can handle the form ourselves — e.g., validate inputs, send data to a backend, etc.

📄 Example with Full Comments

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

function MyForm() {

  // 💡 State to store input value
  const [name, setName] = useState("");

  // 📁 Function to handle form submission
  const handleSubmit = (e) => {
    e.preventDefault(); // 🚫 Stop browser from refreshing
    console.log("Form submitted!");
    console.log("Name entered:", name); // 🖨 Show the value in console
  };

  return (
    <form onSubmit={handleSubmit}>
      {/* 🔥 Controlled input: value comes from state, changes update state */}
      <input
        type="text"
        value={name} // Shows current value from state
        onChange={(e) => setName(e.target.value)} // Updates state as you type
```

```
/>
```

```
{/* 📄 Clicking this triggers onSubmit event */}
```

```
<button type="submit">Submit</button>
```

```
</form>
```

```
);
```

```
}
```

```
export default MyForm;
```

Step-by-step:

1. **useState** → Stores the form data (name).
 2. **onSubmit={handleSubmit}** → Runs handleSubmit function when form is submitted.
 3. **e.preventDefault()** → Stops browser from refreshing the page.
 4. **console.log()** → You can check if form data is being received properly before sending it to a backend.
 5. **Controlled Input** →
 - **value={name}** → Input shows value from state
 - **onChange** updates state with what the user types
-

Why prevent default?

- If you don't, React will reload the page and you'll lose all your app's state.
 - We want to handle everything inside React instead of letting the browser do it.
-

If you want, I can now merge **Controlled Inputs (part 1)** and **Form Handling (part 2)** into a **single example** so you see them working together like in a real form.