

Alright, let's go through **Basic Validation with useState** step-by-step and I'll rewrite your example with **clear code comments** so you fully understand what's happening.

---

### **Basic Validation with useState**

Here, we'll check if an **email** entered by the user is valid (very basic check — just checking if it contains "@").

---

### **Full Example with Comments**

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

function EmailForm() {

  // 💡 State for storing the email input value
  const [email, setEmail] = useState("");

  // 💡 State for storing an error message
  const [error, setError] = useState("");

  // 📄 Function runs when the form is submitted
  const handleSubmit = (e) => {

    e.preventDefault(); // 🚫 Stop page reload

    // 🔍 Validation check: email must include '@'
    if (!email.includes("@")) {

      setError("Invalid email"); // ❌ Set error message
    } else {

      setError(""); // ✅ Clear error message

      console.log("Form submitted!"); // 📄 Show success in console
    }
  };
};
```

```

return (
  <form onSubmit={handleSubmit}>
    {/* 📌 Controlled input: value comes from state */}
    <input
      type="email"
      placeholder="Enter your email"
      value={email}
      onChange={(e) => setEmail(e.target.value)} // 📝 Update state as user types
    />

    {error && <p style={{ color: "red" }}>{error}</p>}

    <button type="submit">Submit</button>
  </form>
);
}

export default EmailForm;

```

### 🔍 How it works:

1. **email state** → Stores the current text in the input.
2. **error state** → Stores the error message (empty if no error).
3. **When Submit is clicked:**
  - If email doesn't have "@" → Show "Invalid email".
  - Else → Clear the error and log "Form submitted!".
4. **{error && ...}** → This is a conditional render.

- If error has text, it will display the <p> tag.
- If error is empty, nothing will show.

---

💡 This is called **basic inline validation** — it happens when you click Submit.

If you want, I can show you how to do **live validation** (it shows error instantly while typing) using the same `useState` method. That's the next step after this in most form handling tutorials.