**1. Explain React Events**

* React events are similar to native DOM events but are wrapped in a **cross-browser wrapper** known as the **SyntheticEvent**.
* These events handle user interactions like clicks, keypresses, form submissions, etc.
* React uses **camelCase** for event names instead of lowercase (as in HTML).

Example:

jsx

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

**2. Explain About Event Handlers**

* An event handler is a **function** that runs when an event is triggered by the user (like clicking a button).
* It is passed as a **prop** to the JSX element using the appropriate event attribute.

Example:

jsx

function handleClick() {

alert("Button was clicked!");

}

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

**3. Define Synthetic Event**

* A **SyntheticEvent** is React’s **cross-browser wrapper** around the browser’s native event system.
* It normalizes the events to work **consistently** across different browsers.
* SyntheticEvent wraps native DOM events like click, change, submit, etc., and supports **pooling** to optimize performance.

Example:

jsx

function handleInput(event) {

console.log(event.target.value); // SyntheticEvent object

}

**4. Identify React Event Naming Convention**

* React follows **camelCase** naming for event props.
* Examples:
  + onClick instead of onclick
  + onChange instead of onchange
  + onSubmit instead of onsubmit

| **HTML Event Attribute** | **React Equivalent** |
| --- | --- |
| onclick | onClick |
| onchange | onChange |
| onsubmit | onSubmit |