**Question 1: How are events handled in React compared to vanilla JavaScript? Explain the concept of synthetic events.**

**Event Handling in React vs. Vanilla JavaScript**

1. **Event Binding:**
   * In **vanilla JavaScript**, events are added using methods like addEventListener() directly on DOM elements.
   * In **React**, event handlers are passed as props (e.g., onClick={handleClick}) to JSX elements.
2. **Event Naming:**
   * In vanilla JavaScript, event names are lowercase (e.g., onclick, onmouseover).
   * In React, event names use camelCase (e.g., onClick, onMouseOver).
3. **Event Handling:**
   * In vanilla JavaScript, you usually call event.preventDefault() manually in an event listener.
   * In React, events are wrapped in a **Synthetic Event** system, but you still call event.preventDefault() as usual.
4. **Event Binding in Classes:**
   * In vanilla JavaScript, binding is not an issue when using regular functions.
   * In React class components, event handlers need to be explicitly bound (this.handleClick = this.handleClick.bind(this);) or defined as arrow functions to preserve this context.

**Synthetic Events in React**

* **Definition:**  
  Synthetic events are React’s wrapper around native browser events. They provide a unified API across different browsers to ensure consistency.
* **Benefits of Synthetic Events:**
  + Works across all browsers.
  + Improves performance through **event pooling** (events are reused to reduce memory consumption).
  + Provides the same event properties as native events (e.g., event.target, event.type).
* **Example of Synthetic Events:**

function Button() {

function handleClick(event) {

console.log("Button clicked!", event.type); // Synthetic event

}

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

}

**Question 2: What are some common event handlers in React.js? Provide examples of onClick, onChange, and onSubmit.**