REACT ASSIGNMENT

**Explain React Events**

React events are a way to handle user interactions such as clicks, form submissions, keyboard presses, and more, using a consistent and cross-browser-compatible system. Similar to HTML DOM events, React provides built-in support for events but does so using a wrapped system called synthetic events to ensure compatibility across all browsers. Events in React can be attached to JSX elements using props like onClick, onChange, or onSubmit.

**Explain About Event Handlers**

Event handlers in React are functions that are called in response to specific user actions or events, such as a button click or text input. These handlers are passed as props to elements using event attributes like onClick={handleClick}. Typically, event handler functions are defined within the component and may use state or props to manage logic or update the UI. React event handlers can be regular functions or arrow functions, depending on how you want to handle this.

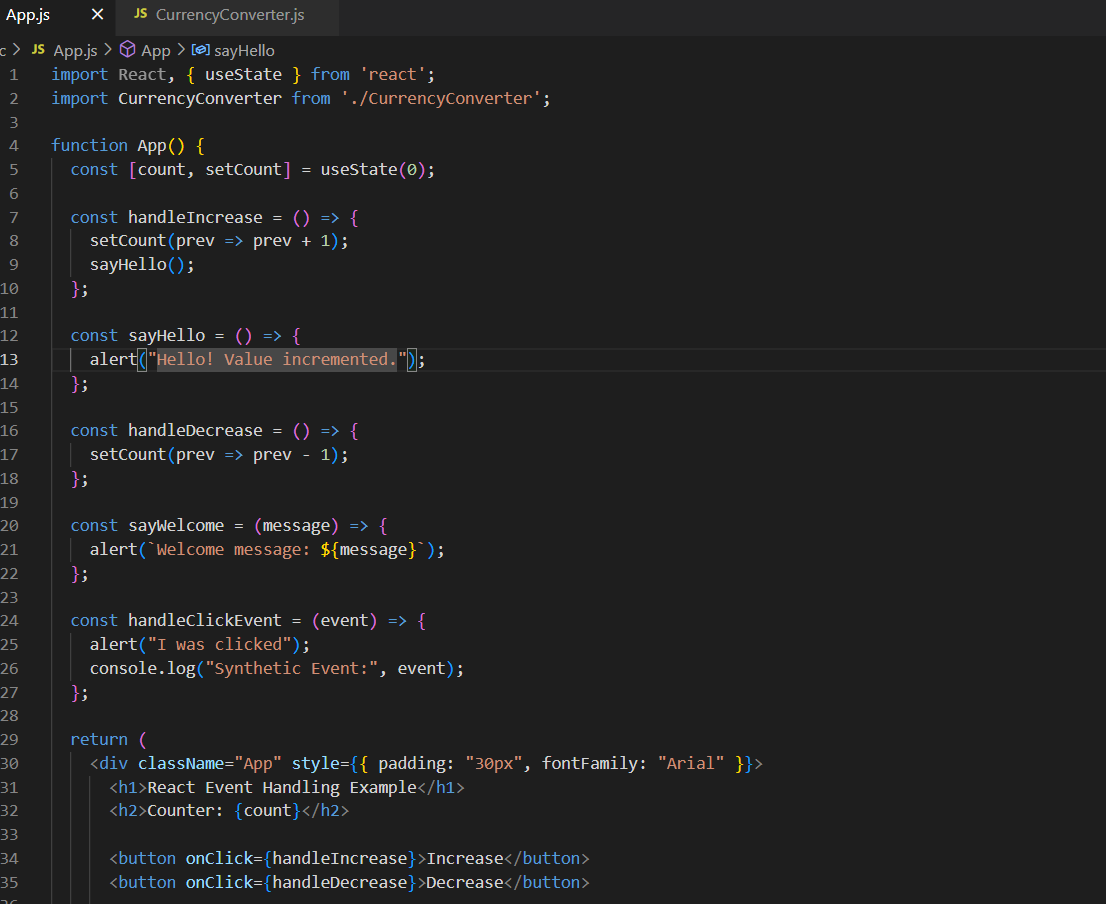
**Define Synthetic Event**

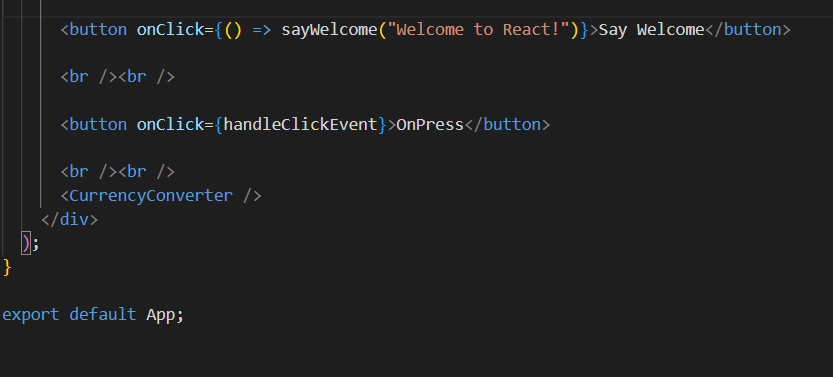
A Synthetic Event in React is a wrapper around the browser’s native event, provided by React to ensure that events behave consistently across all browsers. React creates a synthetic event object that wraps the native DOM event and normalizes its behavior, allowing developers to interact with properties like event.target and event.preventDefault() just as they would with a real event. This system improves performance and cross-browser reliability.

**Identify React Event Naming Convention**

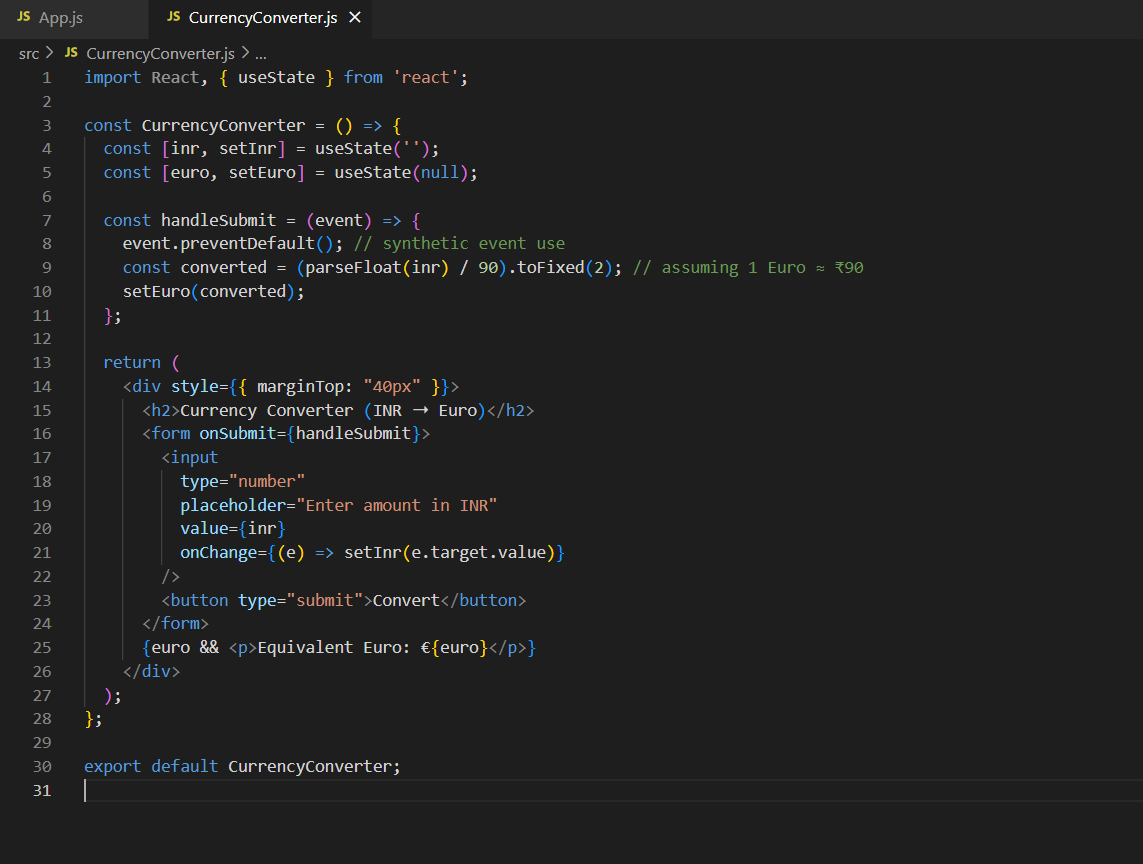
React follows a camelCase naming convention for event attributes, unlike HTML which uses lowercase. For example, onclick in HTML becomes onClick in React, and onchange becomes onChange. Additionally, in JSX, event handlers must be passed as a function (not a string), such as onClick={handleClick} instead of onClick="handleClick()". This ensures React can properly bind and manage events in a declarative way.

**App.js:**

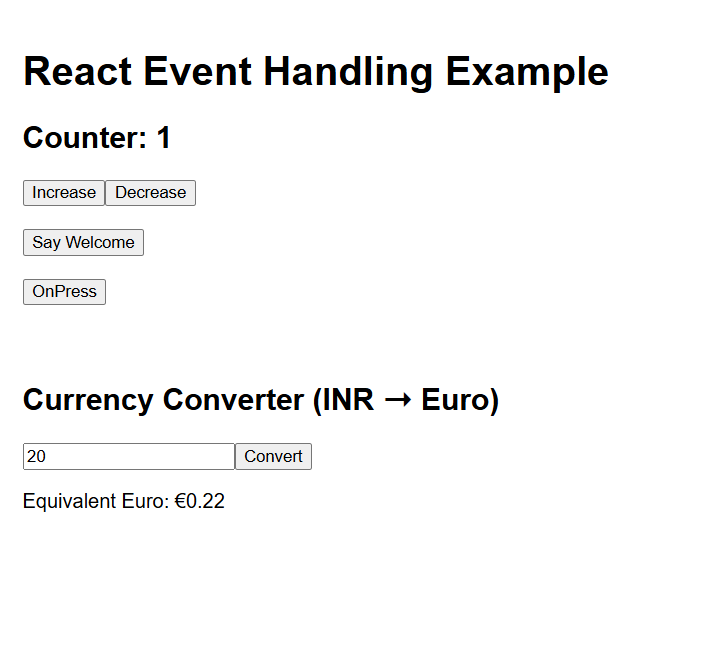




**CurrencyConverter.js:**

****

**OUTPUT:**

****