**WEEK-7-ReactJS-HOL 11**

**SUPERSET ID 6393676**

**Objectives**

**• Explain React events**

React events are interactions triggered by users (like clicks, typing, form submissions) that React handles similarly to native DOM events, but using a unified and consistent interface. React wraps native events inside a synthetic event system to provide cross-browser compatibility.

**• Explain about event handlers**

Event handlers are functions in React that respond to user actions. They are assigned as props to React elements using camelCase syntax. For example, onClick={handleClick} attaches a click handler to a button element.

**• Define Synthetic event**

A SyntheticEvent is a cross-browser wrapper around the browser’s native event. It provides a consistent interface for handling events in React and works identically across all browsers, abstracting away browser-specific quirks.

**• Identify React event naming convention**

React uses camelCase naming for event handlers (unlike HTML which uses lowercase). For example:  
✅ onClick, onChange, onSubmit  
❌ onclick, onchange

Create a React Application “eventexamplesapp” to handle various events of the form elements in HTML.

1. Create “Increment” button to increase the value of the counter and “Decrement” button to decrease the value of the counter. The “Increase” button should invoke multiple methods.
   1. To increment the value
   2. Say Hello followed by a static message.

**APP.JS**

import React from 'react';

import EventHandlingExample from './EventHandlingExample';

function App() {

  return (

    <div className="App">

      <EventHandlingExample />

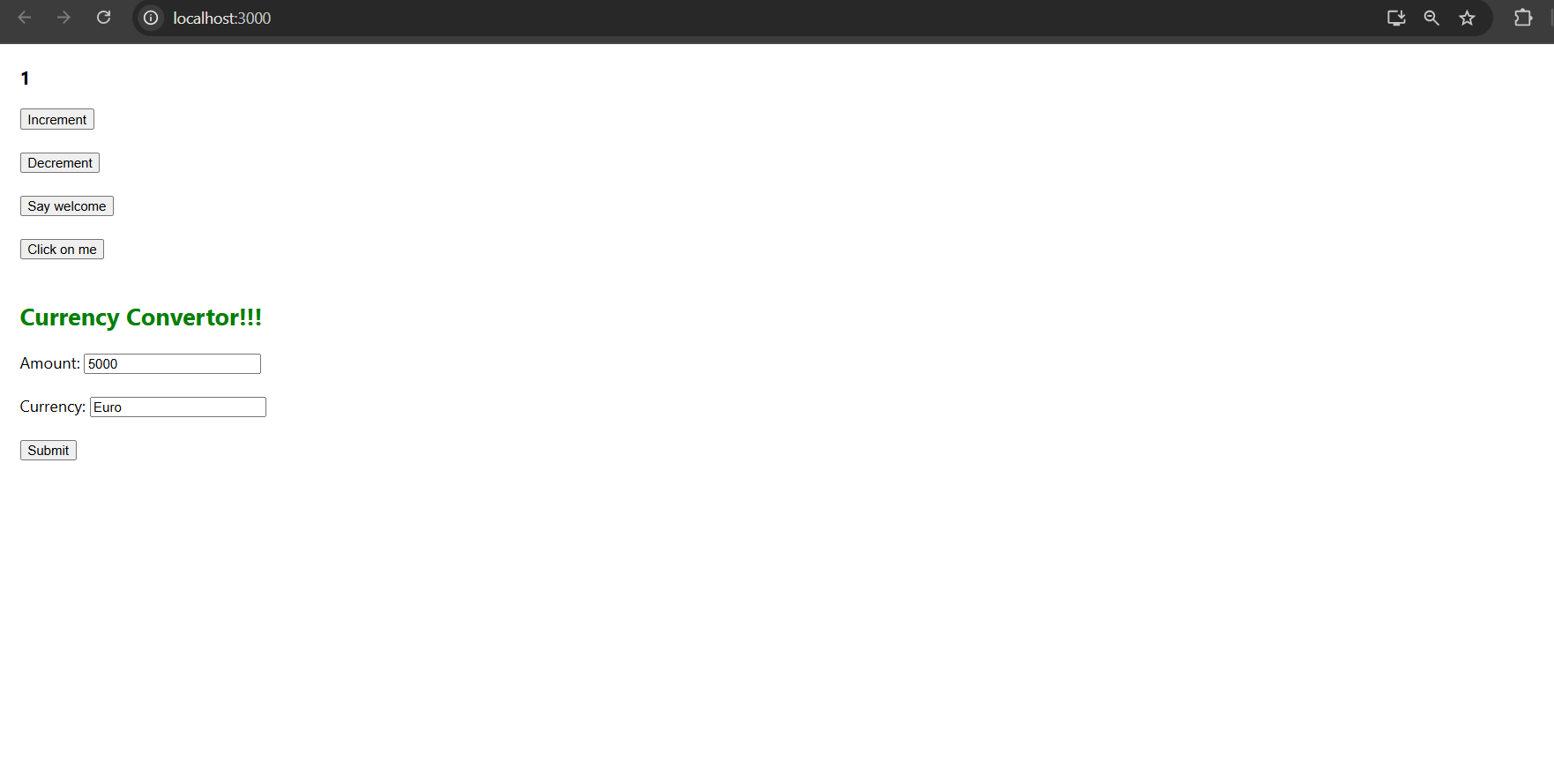
    </div>

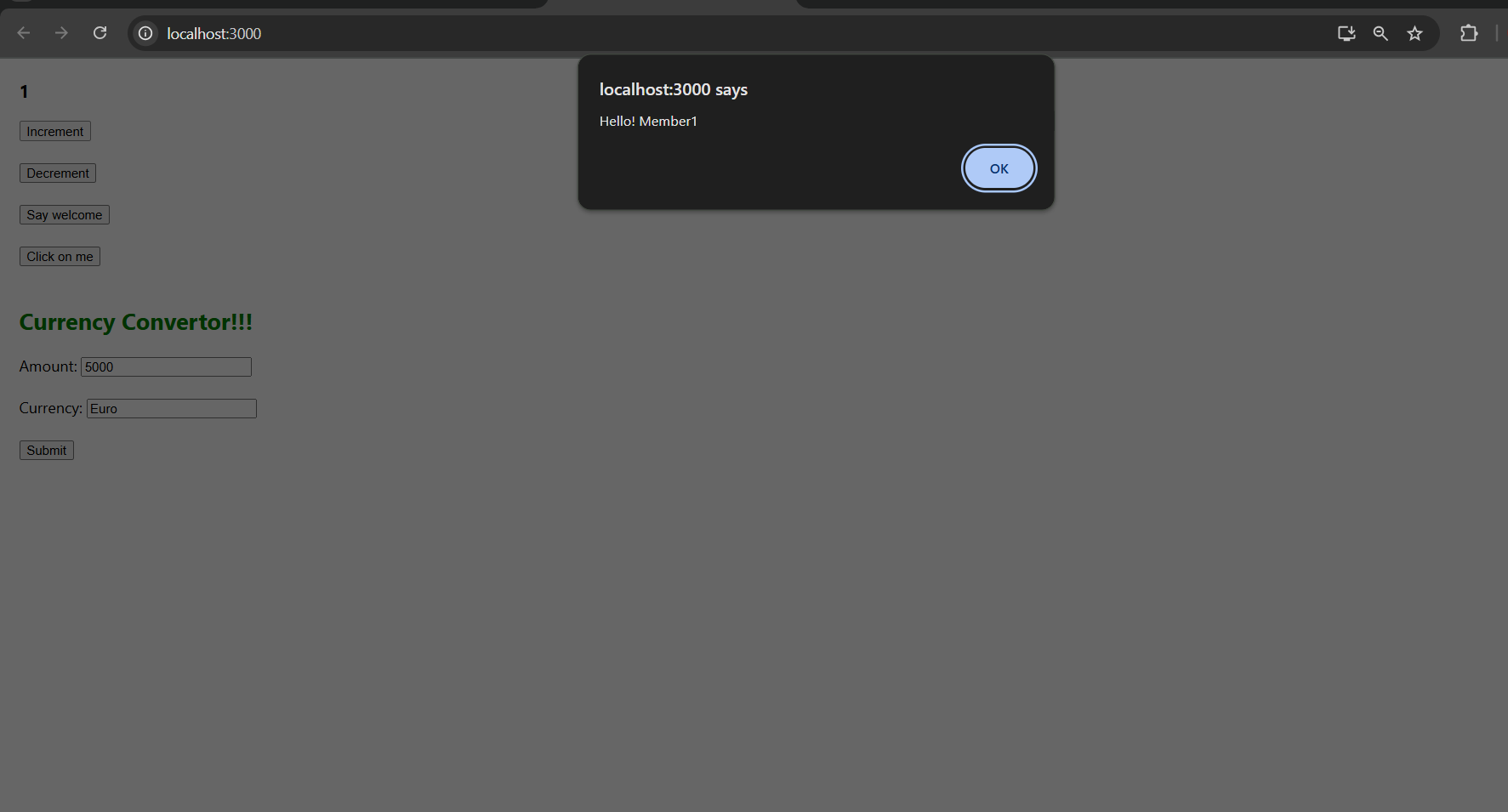
  );

}

export default App;

**OUTPUT**

****

****

