**WEEK – 7(HandsOn)**

**React**

**11. ReactJS-HOL**

**Objectives**

**1. Explain React Events**

React events are how you handle user interactions in a React application — like clicks, typing, hovering, submitting forms, etc.

React uses its own cross-browser wrapper for native DOM events — called the Synthetic Event System**.**

**Example:**

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

**2. Explain Event Handlers**

**Event handlers** are functions that respond when a specific event occurs.

They can be:

* Inline
* Arrow functions
* Defined as separate named functions

**Example**: Event handler in React

function handleClick() {

alert("Button Clicked!");

}

function App() {

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

}

**3. Define Synthetic Event**

A **Synthetic Event** is a **wrapper around the browser’s native event** system provided by React.

* It normalizes events across all browsers.
* You don’t need to worry about cross-browser differences (like IE vs Chrome).

**Example:**

function handleChange(event) {

console.log(event.target.value); // event is a SyntheticEvent

}

**4. React Event Naming Conventions**

React follows **camelCase** for event names, unlike traditional HTML.

| **HTML** | **React** |
| --- | --- |
| onclick | onClick |
| onchange | onChange |
| onmouseover | onMouseOver |
| onsubmit | onSubmit |

**CODE:**

Src/App.js:

import React, { useState } from "react";

function App() {

  // Counter state

  const [counter, setCounter] = useState(5);

  // Currency converter states

  const [amount, setAmount] = useState("");

  const [currency, setCurrency] = useState("");

  // Increment button - triggers two functions (alert + increase counter)

  const handleIncrement = () => {

    alert("Hello! Member1");

    setCounter(counter + 1);

  };

  // Decrement button

  const handleDecrement = () => {

    setCounter(counter - 1);

  };

  // Say welcome button with argument

  const handleWelcome = () => {

    alert("welcome");

  };

  // Universal click event

  const handleClick = () => {

    alert("I was clicked");

  };

  // Currency converter submit

  const handleSubmit = (e) => {

    e.preventDefault();

    // Example: if converting to Euro and amount is entered

    if (currency.trim().toLowerCase() === "euro" && amount) {

      // Let's say 1 unit = 80 INR (so 80 INR = 1 Euro, as per your screenshot logic: 80 x 80 = 6400)

      const amt = parseFloat(amount);

      if (!isNaN(amt)) {

        alert(`Converting to Euro Amount is ${amt \* 80}`);

      }

    }

  };

  return (

    <div style={{ margin: "30px" }}>

      {/\* Counter display \*/}

      <div style={{ fontSize: "20px", marginBottom: "10px" }}>{counter}</div>

      {/\* Buttons for Increment/Decrement/Welcome/Click \*/}

      <button onClick={handleIncrement}>Increment</button>

      <br />

      <button onClick={handleDecrement}>Decrement</button>

      <br />

      <button onClick={handleWelcome}>Say welcome</button>

      <br />

      <button onClick={handleClick}>Click on me</button>

      {/\* Heading \*/}

      <h1 style={{

        color:"green",

        marginTop: "40px",

        fontFamily: "Arial",

        fontWeight: "bold"

      }}>Currency Convertor!!!</h1>

      {/\* Currency Converter Form \*/}

      <form onSubmit={handleSubmit}>

        <table>

          <tbody>

            <tr>

              <td style={{ fontSize: "18px" }}>Amount:</td>

              <td>

                <input

                  type="text"

                  value={amount}

                  onChange={e => setAmount(e.target.value)}

                  style={{ margin: "10px", height: "22px", fontSize: "16px" }}

                />

              </td>

            </tr>

            <tr>

              <td style={{ fontSize: "18px" }}>Currency:</td>

              <td>

                <input

                  type="text"

                  value={currency}

                  onChange={e => setCurrency(e.target.value)}

                  style={{ margin: "10px", height: "30px", fontSize: "16px" }}

                />

              </td>

            </tr>

            <tr>

              <td colSpan={2}>

                <button

                  type="submit"

                  style={{

                    marginLeft: "70px",

                    width: "60px",

                    height: "28px"

                  }}

                >Submit</button>

              </td>

            </tr>

          </tbody>

        </table>

      </form>

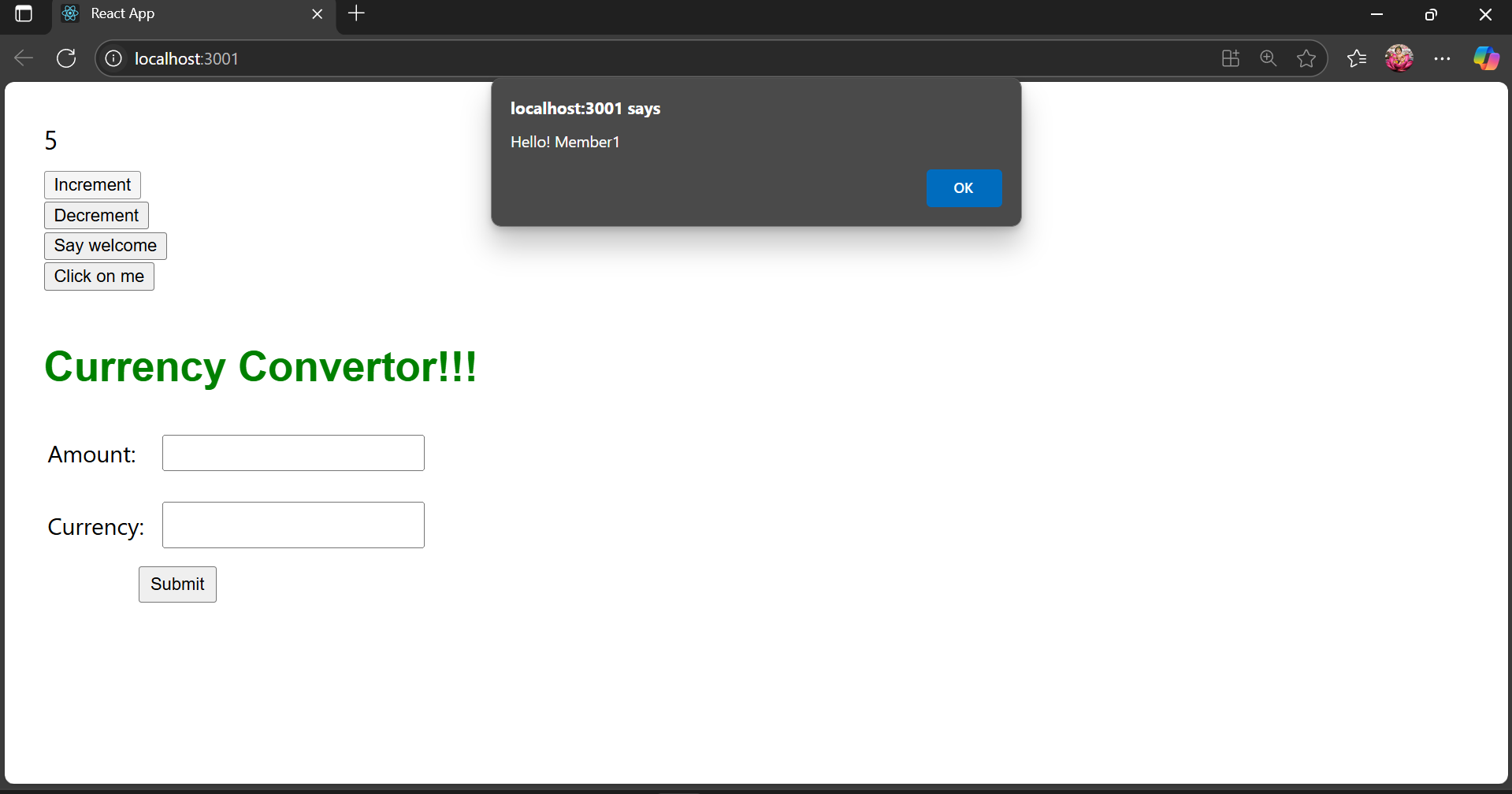
    </div>

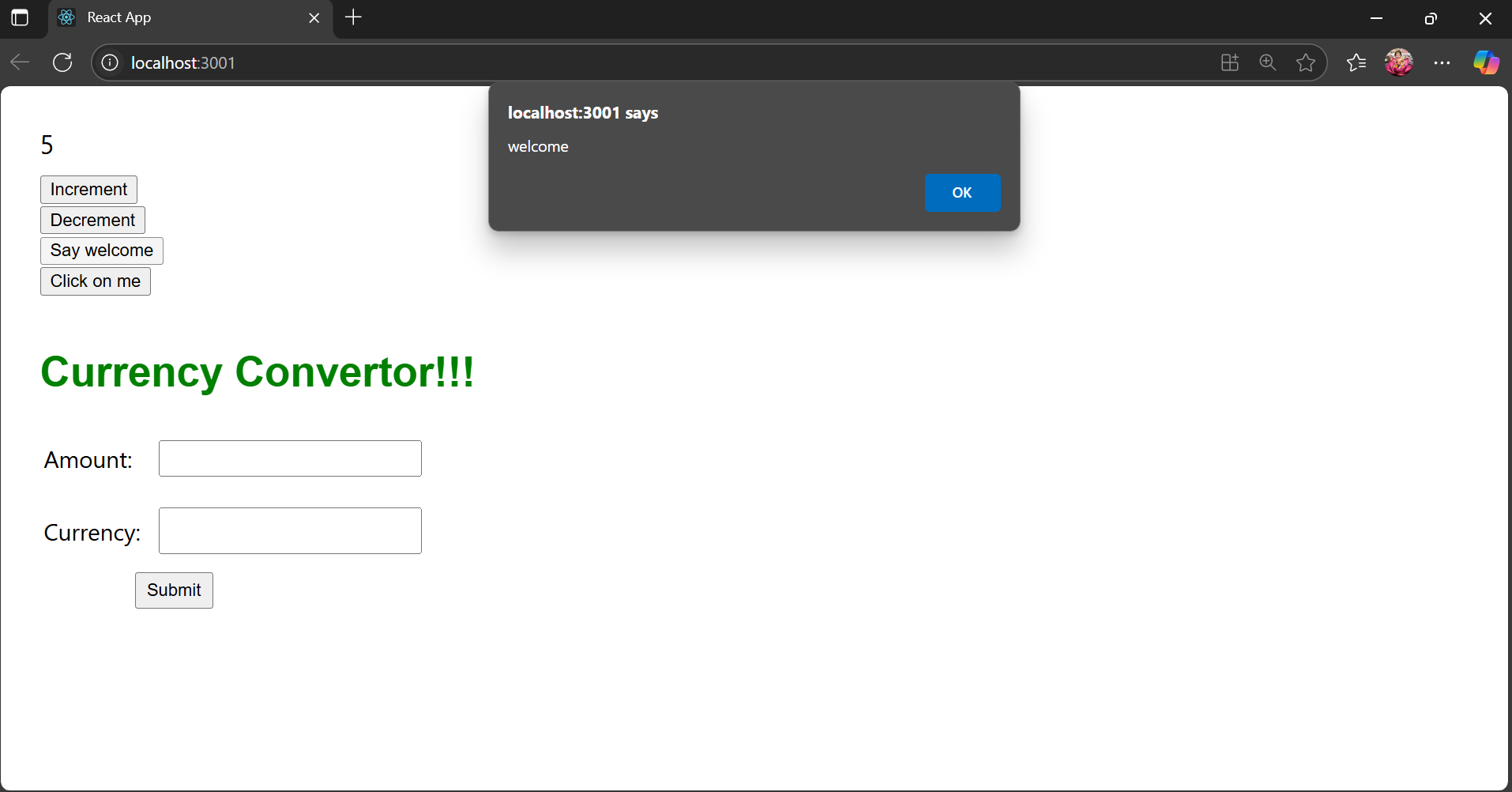
  );

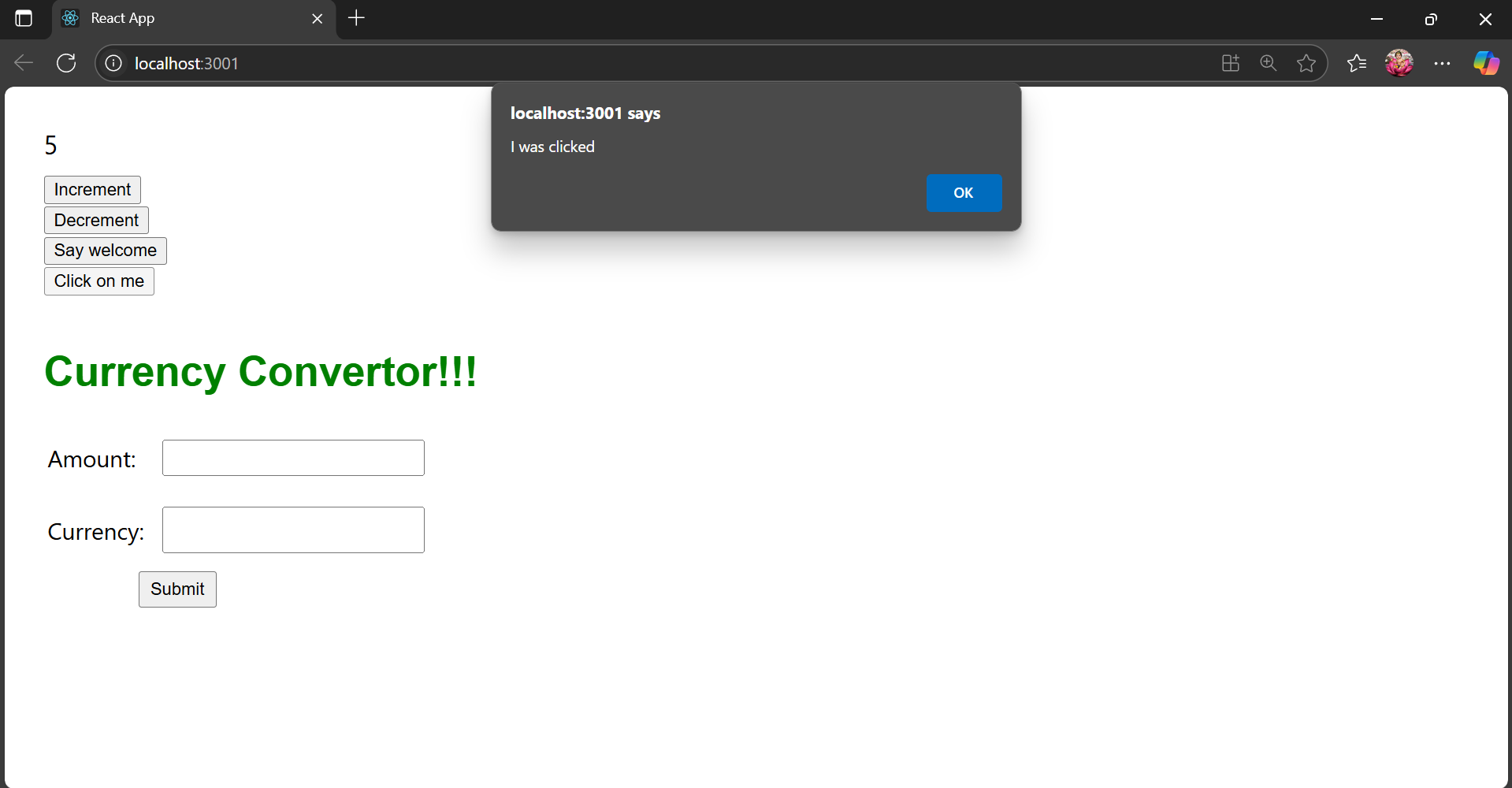
}

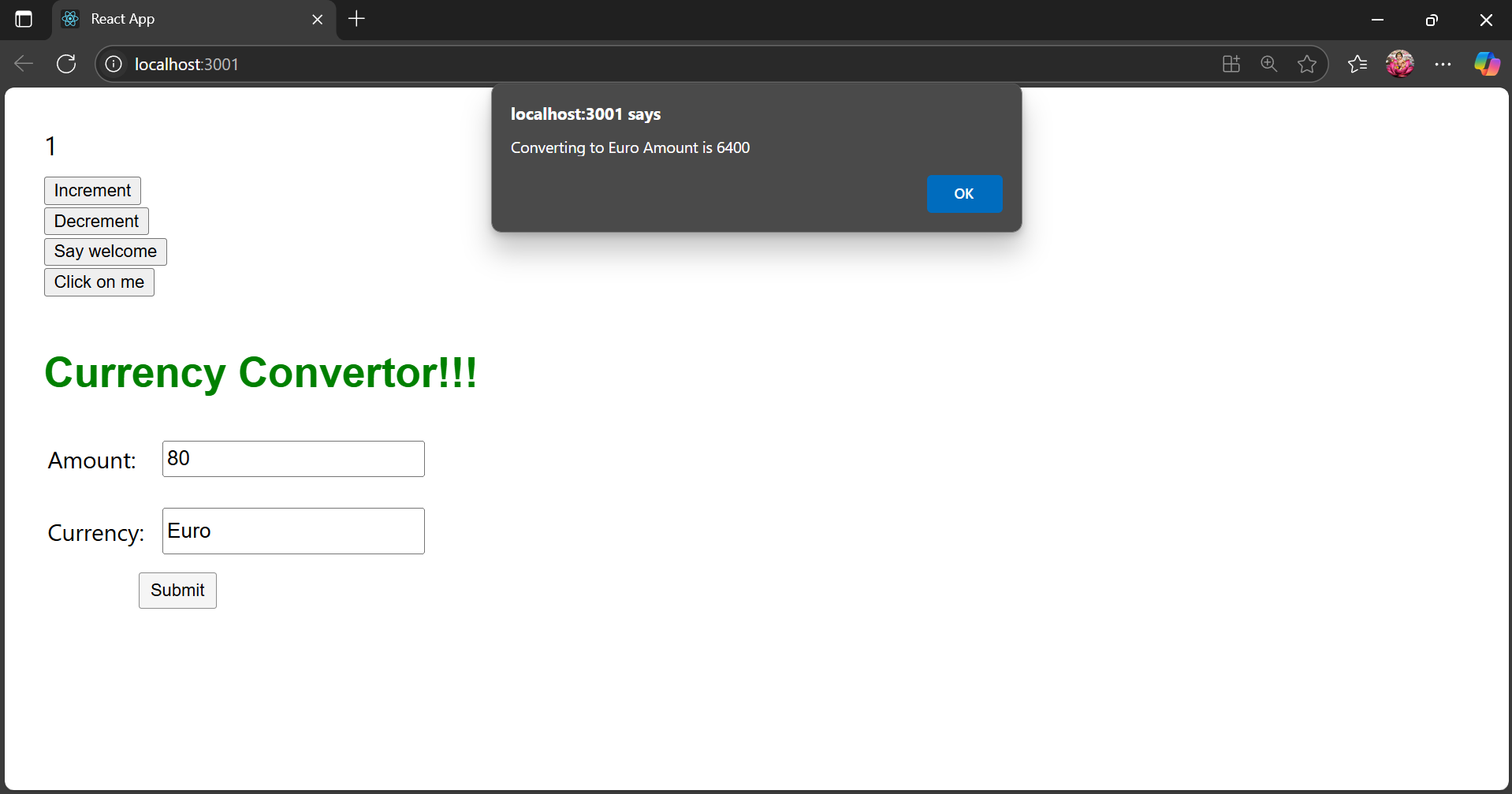
export default App;

**OUTPUT:**

****

****

****

****