**Currency Converter**

**REACT**

Vaishnavi

30/7/2025

Event handling in React is essential for making web applications interactive. Unlike traditional HTML DOM events, React events are wrapped in **Synthetic Events**, which are a cross-browser wrapper around native events, ensuring consistent behavior across platforms. React uses **camelCase naming** for events and associates event handlers using functions, often bound to the component’s state.

In this lab, we will:

Implement event handling in React.

Understand **synthetic events**, **event handlers**, and **this keyword usage**.

Build a small app (eventexamplesapp) that includes:

Counter functionality with increment and decrement buttons.

A button that displays a welcome message using arguments.

A button that triggers a synthetic event.

A **Currency Converter component** that converts Indian Rupees to Euro on button click.

**Objective:**

In React, events allow components to respond to user actions like clicking a button, typing in a field, or submitting a form. These events work similarly to native DOM events but are handled via React's own system, providing a unified and cross-browser compatible interface known as Synthetic Events.

Event handlers are functions that define what action should be taken when a user performs an event. In React, you define these functions inside a component and bind them to JSX elements using props.

SyntheticEvent is React’s cross-browser wrapper around native browser events. It behaves identically across different platforms and ensures consistent functionality. These events are part of React’s internal event system and help in optimizing performance through object reuse.

React uses **camelCase** for event names instead of the lowercase format used in HTML. Additionally, instead of strings, functions are passed as event handlers.

* React: onChange={handleChange}
* HTML: onchange="handleChange()"  
  This design aligns with JavaScript and JSX syntax standards.

**Implementation:**

**Step 1: Create a React App**

****

**Step 2: Use the Component in App.js**

import React, { useState } from 'react';

import './App.css';

function App() {

  const [count, setCount] = useState(0);

  const [inr, setInr] = useState('');

  const [euro, setEuro] = useState('');

  const increment = () => {

    setCount(prev => prev + 1);

  };

  const decrement = () => {

    setCount(prev => prev - 1);

  };

  const sayHello = () => {

    console.log("Hello! This is a static message.");

  };

  const handleIncrementClick = () => {

    increment();

    sayHello();

  };

  const sayWelcome = (msg) => {

    alert(msg);

  };

  const handleSynthetic = (e) => {

    console.log("Synthetic event object:", e);

    alert("I was clicked");

  };

  const handleSubmit = (e) => {

    e.preventDefault();

    const result = (parseFloat(inr) \* 0.00984).toFixed(2);

    setEuro(result);

  };

  return (

    <div className="container">

      <h1>React Event App</h1>

      <div className="counter-box">

        <p>Counter Value: {count}</p>

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

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

      </div>

      <div className="section">

        <button onClick={() => sayWelcome("Welcome to React Page!")}>Say Welcome</button>

      </div>

      <div className="section">

        <button onClick={handleSynthetic}>Click me</button>

      </div>

      <div className="section">

        <h2>Currency Converter (INR ➡ Euro)</h2>

        <form onSubmit={handleSubmit}>

          <input

            type="number"

            value={inr}

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

            placeholder="Enter INR"

            required

          />

          <button type="submit">Convert</button>

        </form>

        {euro && <p>Converted Amount: €{euro}</p>}

      </div>

    </div>

  );

}

**App.css**

body {

  margin: 0;

  padding: 0;

  font-family: 'Segoe UI', sans-serif;

  background-color: #f4f6f8;

}

.container {

  max-width: 600px;

  margin: 40px auto;

  background: white;

  padding: 30px;

  border-radius: 12px;

  box-shadow: 0 0 15px rgba(0, 0, 0, 0.1);

}

h1 {

  color: #444;

  text-align: center;

}

.section,

.counter-box {

  margin: 20px 0;

  text-align: center;

}

button {

  margin: 5px;

  padding: 10px 15px;

  background-color: #007bff;

  color: white;

  border: none;

  border-radius: 5px;

  cursor: pointer;

}

button:hover {

  background-color: #0056b3;

}

input {

  padding: 10px;

  margin-right: 10px;

  width: 60%;

  border: 1px solid #ccc;

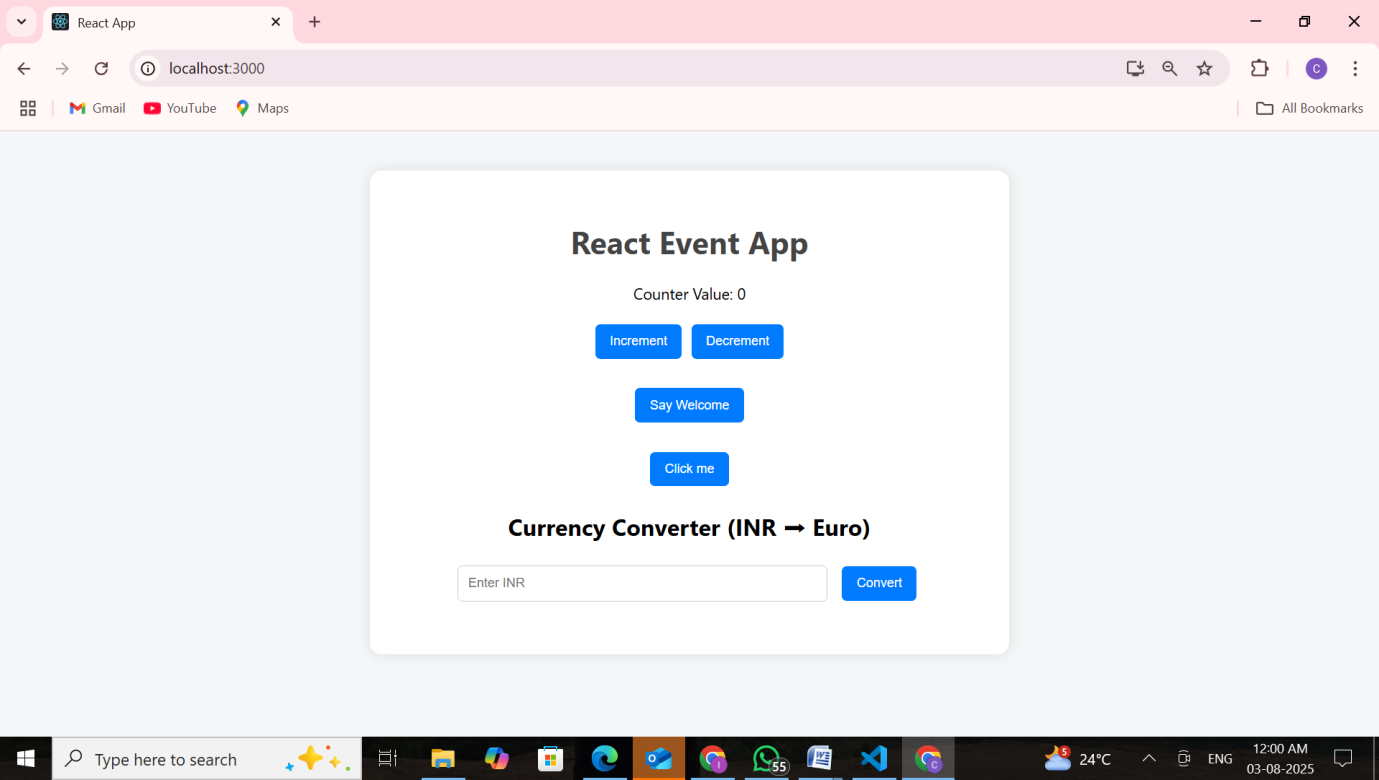
  border-radius: 5px;

}

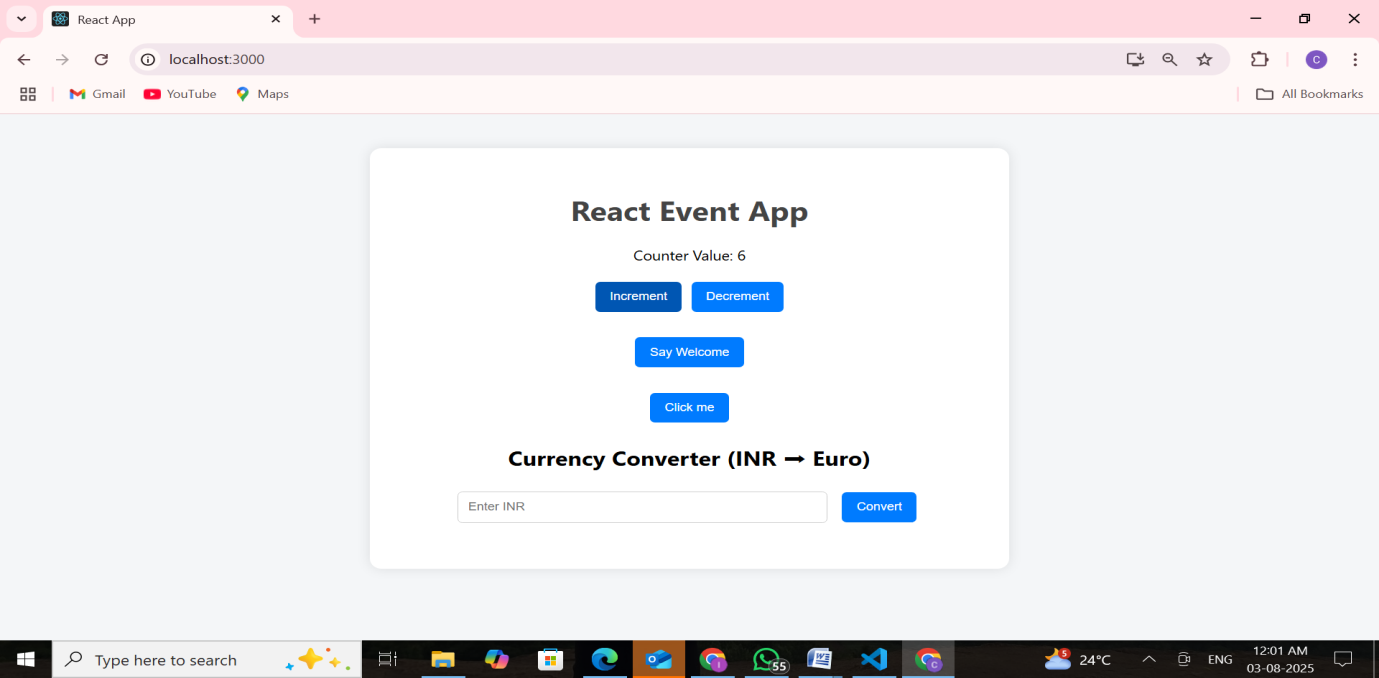
**Step 4: Run the App**

****

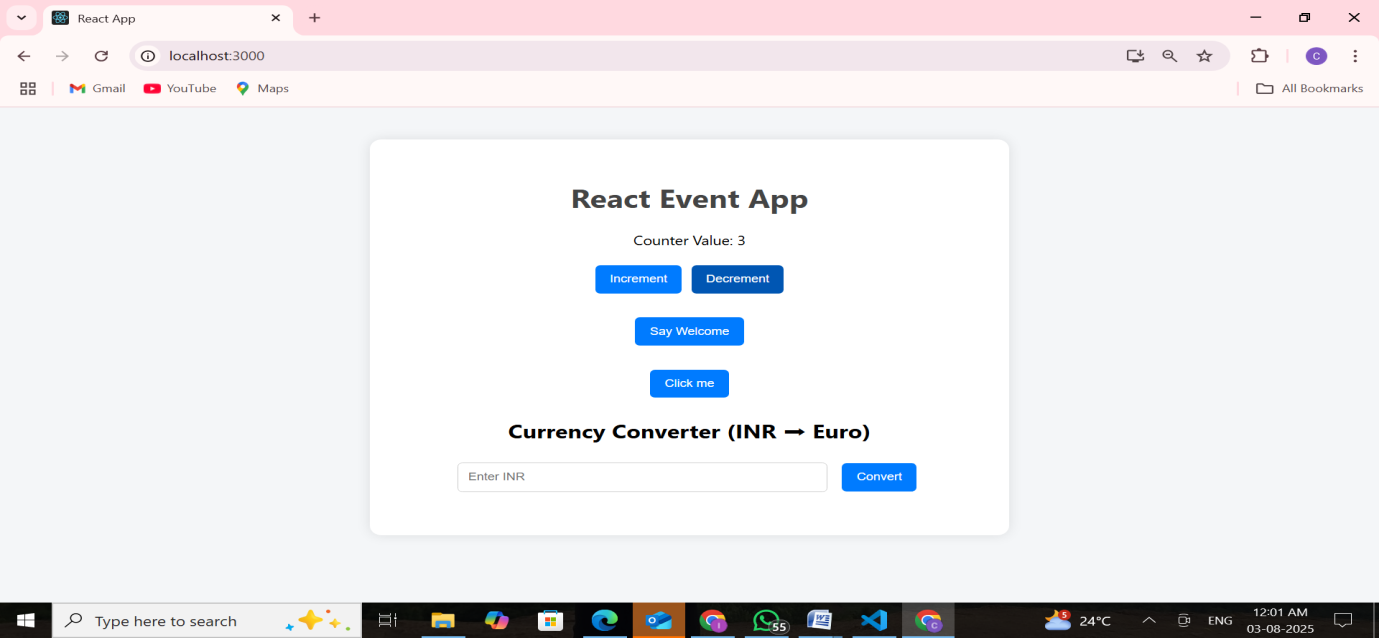
**Output:**

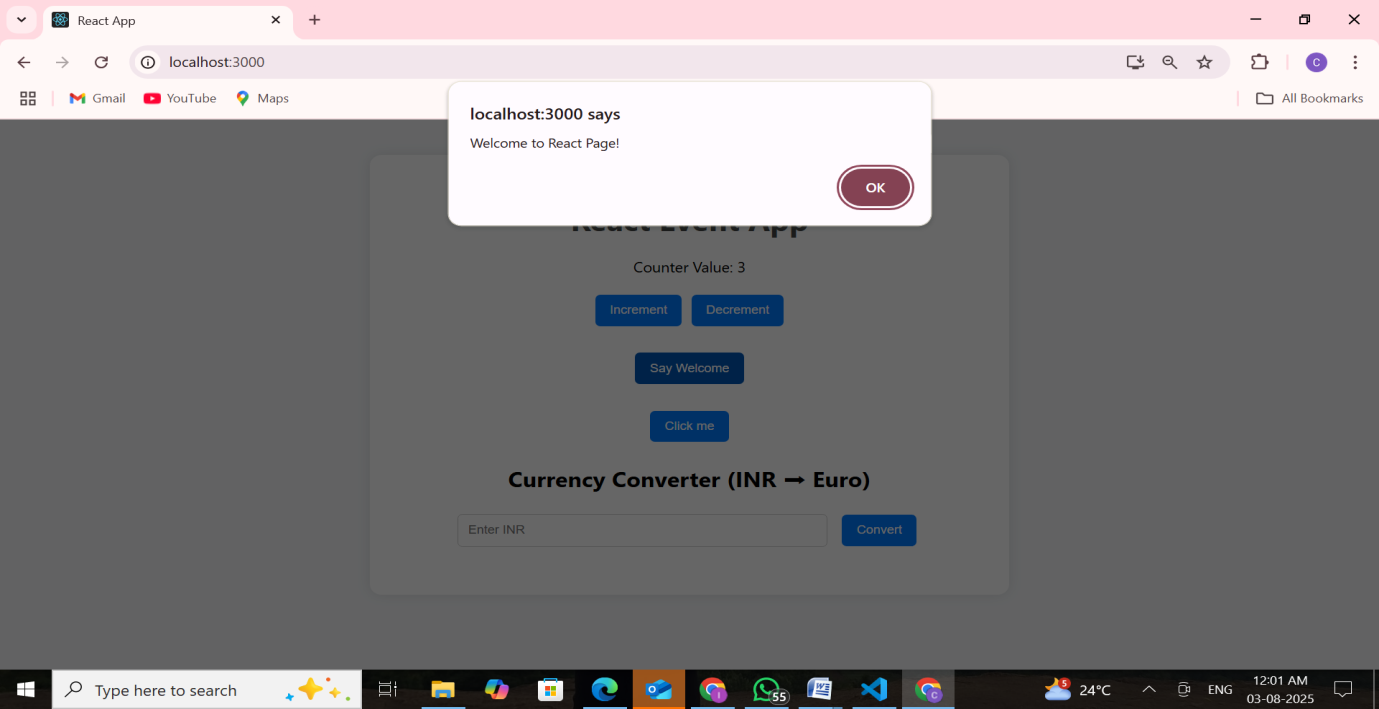


Increment:

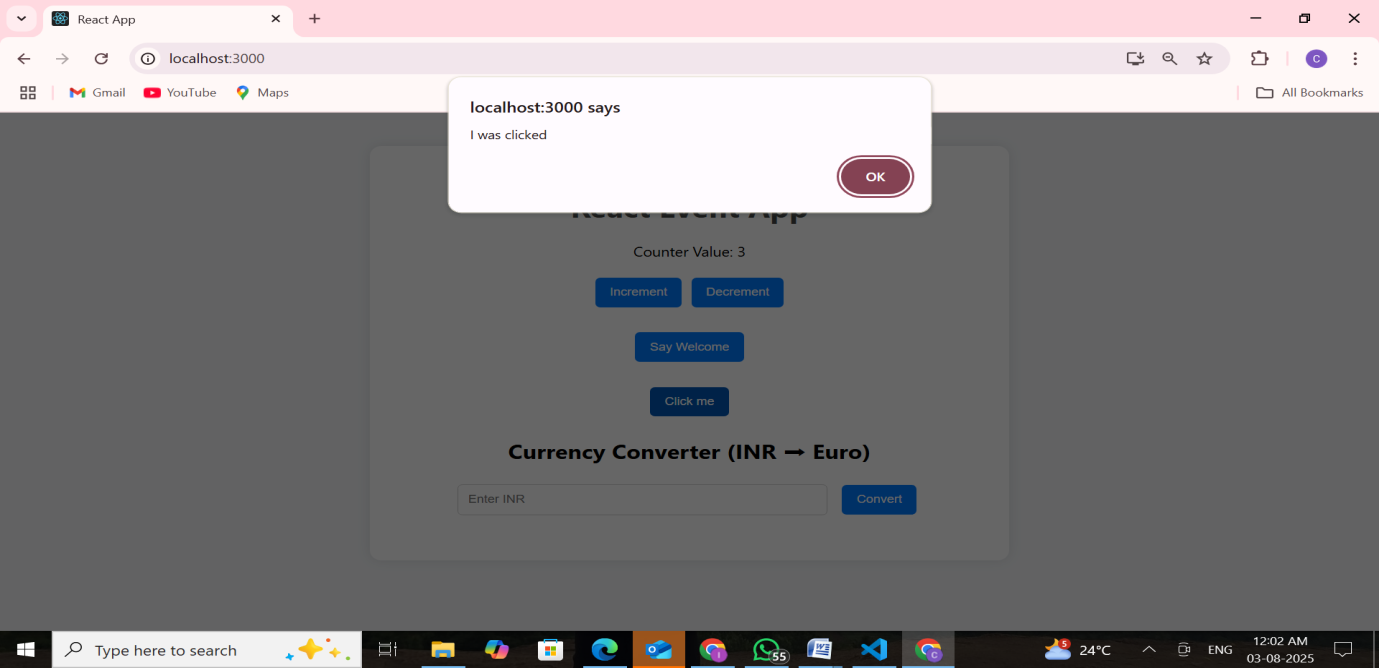


Decrement:



Say welcome: 

Click me:



Currency Converter (Indian to Euro):

