**Ex 11**

**App.js**

import React, { useState } from 'react';

import CurrencyConvertor from './CurrencyConvertor';

const App = () => {

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

const increaseCounter = () => {

setCount(prev => prev + 1);

greetUser();

};

const decreaseCounter = () => {

setCount(prev => prev - 1);

};

const greetUser = () => {

console.log("Hi Poshika! You're increasing the count");

};

const welcomeMessage = (msg) => {

alert(msg);

};

const syntheticEventHandler = (e) => {

alert("Poshika clicked this button!");

console.log("Synthetic event triggered by Poshika:", e);

};

return (

<div style={{ padding: "40px", fontFamily: "Segoe UI" }}>

<h1>React Events Practice - Poshika</h1>

<h2>Current Value: {count}</h2>

<button onClick={increaseCounter}>Add Count</button>

<button onClick={decreaseCounter} style={{ marginLeft: "10px" }}>Reduce Count</button>

<div style={{ marginTop: "20px" }}>

<button onClick={() => welcomeMessage("Welcome to React Events, Poshika!")}>Show Welcome</button>

</div>

<div style={{ marginTop: "20px" }}>

<button onClick={syntheticEventHandler}>Synthetic Click Event</button>

</div>

<hr style={{ margin: "30px 0" }} />

<CurrencyConvertor />

</div>

);

};

export default App;

**CurrencyConvertor.js**

import React, { useState } from 'react';

const CurrencyConvertor = () => {

const [inr, setInr] = useState('');

const [convertedEuro, setConvertedEuro] = useState(null);

const convertCurrency = (e) => {

e.preventDefault();

const rate = 0.011;

const euro = (parseFloat(inr) \* rate).toFixed(2);

setConvertedEuro(euro);

};

return (

<div>

<h2>INR to Euro Converter</h2>

<form onSubmit={convertCurrency}>

<input

type="number"

value={inr}

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

placeholder="Amount in INR"

/>

<button type="submit" style={{ marginLeft: "10px" }}>Convert to EUR</button>

</form>

{convertedEuro && (

<p style={{ marginTop: "15px" }}>

₹{inr} = <strong>€{convertedEuro}</strong>

</p>

)}

</div>

);

};

export default CurrencyConvertor;

**Output :**







