**11. REACTJS – HOL**

**Title: React Event Handling Lab**

**1. Explain React Events**

React events are actions that a user performs — like clicking a button, typing into a textbox, or submitting a form. React handles these events using **JSX syntax** similar to HTML, but with **camelCase naming**, and uses **functions or methods** to respond to these actions.

**2. Explain Event Handlers**

Event handlers are **functions** that get triggered when a specific event occurs. In React, you attach event handlers directly to elements using attributes like onClick, onChange, etc.

Example:

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

Here, handleClick is the event handler.

**3. Define Synthetic Event**

A Synthetic Event is a **React wrapper** around the browser’s native event. It normalizes the event object so that it behaves **consistently across all browsers**, providing better performance and compatibility.

**4. Identify React Event Naming Convention**

React uses **camelCase** for event names (unlike lowercase in HTML).  
For example:

* HTML: <button onclick="doSomething()">
* React: <button onClick={doSomething}>

**5. Implement Event Handling in React**

To handle events in React:

1. Define an event handler method (function)
2. Attach it using the JSX attribute like onClick

Example:

function handleClick() {

alert('Button clicked!');

}

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

**6. Use this Keyword**

When using **class components**, this refers to the component instance. You often bind this in constructors to ensure the handler accesses the correct context.

Example (class component):

this.handleClick = this.handleClick.bind(this);

In **functional components**, this is generally not used — hooks like useState manage data instead.

**7. Use Synthetic Events**

Synthetic events are used by default in React. You can access event data like this:

function handleInputChange(event) {

console.log(event.target.value); // Synthetic event object

}

**Tools/Technologies Used**

| **Tool** | **Purpose** |
| --- | --- |
| React | Frontend framework |
| JSX | JavaScript XML for UI |
| Node.js + NPM | Package and runtime support |
| Visual Studio Code | Code editor |
| JavaScript (ES6) | Event handling and logic |

**Components and Description**

| **Component Name** | **Purpose** |
| --- | --- |
| Counter | Handles increment & decrement events with chained methods |
| SayWelcome | Demonstrates passing parameters in event handlers |
| SyntheticEvent | Handles synthetic event (onClick) |
| CurrencyConverter | Converts INR to Euro using event handlers |

**Step 1: Create the React App**

npx create-react-app eventexamplesapp

cd eventexamplesapp

code .

**Step 2: Clean Up Boilerplate**

**TYPE CODE :**

import React from 'react';

import './App.css';

import EventExample from './EventExample';

import CurrencyConvertor from './CurrencyConvertor';

function App() {

return (

<div className="App">

<h1>React Event Handling Lab</h1>

<EventExample />

<CurrencyConvertor />

</div>

);

}

export default App;

**Step 3: Create EventExample.js**

**Code:**

import React, { Component } from 'react';

class EventExample extends Component {

constructor(props) {

super(props);

this.state = { count: 0 };

}

increment = () => {

this.setState({ count: this.state.count + 1 });

};

sayHello = () => {

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

};

handleIncrease = () => {

this.increment();

this.sayHello();

};

handleDecrement = () => {

this.setState({ count: this.state.count - 1 });

};

sayWelcome = (msg) => {

alert(`Welcome: ${msg}`);

};

onPress = () => {

alert("I was clicked");

};

render() {

return (

<div>

<h2>Counter: {this.state.count}</h2>

<button onClick={this.handleIncrease}>Increment</button>

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

<br /><br />

<button onClick={() => this.sayWelcome("Welcome")}>Say Welcome</button>

<br /><br />

<button onClick={this.onPress}>Synthetic Event (OnPress)</button>

</div>

);

}

}

export default EventExample;

**Step 4: Create CurrencyConvertor.js**

**Code:**

jsx

CopyEdit

import React, { useState } from 'react';

function CurrencyConvertor() {

const [rupees, setRupees] = useState('');

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

const handleSubmit = () => {

const result = parseFloat(rupees) \* 0.011; // Approx conversion

setEuro(result.toFixed(2));

};

return (

<div style={{ marginTop: '30px' }}>

<h2>Currency Convertor</h2>

<input

type="number"

placeholder="Enter amount in INR"

value={rupees}

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

/>

<button onClick={handleSubmit}>Convert</button>

{euro && <p>EUR: €{euro}</p>}

</div>

);

}

export default CurrencyConvertor;

**Step 5: Run the App**npm start

**Step 6 : Output**

