### Exercise 11

### React Events

In React, events are user interactions that trigger a response, such as a mouse click, key press, or form submission. They are similar to native DOM events but with some key differences in how they are handled. Events allow you to create dynamic and interactive user interfaces by responding to these actions.

### Event Handlers

An event handler is a function that is called when a specific event occurs. In React, you define these functions and then attach them to JSX elements. For example, to handle a click on a button, you would define a function and then attach it using an attribute like onClick.

**Example:**

JavaScript

function handleClick() {  
  console.log('Button was clicked!');  
}  
  
<button onClick={handleClick}>Click Me</button>

### Synthetic Event

React's event system is built on something called the SyntheticEvent. A SyntheticEvent is a cross-browser wrapper around the browser's native event. This abstraction ensures that events behave consistently across all browsers, eliminating the need for developers to write browser-specific code. The SyntheticEvent object contains all the properties and methods you would expect from a native event, such as target, preventDefault(), and stopPropagation().

### React Event Naming Convention

React uses a camelCase naming convention for events, which differs from the lowercase convention used in standard HTML. The general convention is as follows:

* **Event Attribute:** The event attribute in JSX is named in camelCase, such as onClick, onChange, onSubmit, and onMouseOver.
* **Handler Function:** The function that handles the event is typically prefixed with handle, followed by the name of the event. For example, the handler for an onClick event is often named handleClick.

**Example:**

JavaScript

function handleChange(event) {  
  // Logic to handle the change event  
  console.log(event.target.value);  
}  
  
<input type="text" onChange={handleChange} />

**App.js**

import React, { useState } from 'react';

import './App.css';

const App = () => {

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

  const handleIncrement = () => {

    setCount(prevCount => prevCount + 1);

  };

  const handleSayHello = () => {

    alert('Hello from a static message!');

  };

  const handleDecrement = () => {

    setCount(prevCount => prevCount - 1);

  };

  const handleSayWelcome = (message) => {

    alert(message);

  };

  const handleSyntheticEvent = () => {

    alert('I was clicked! (This is a synthetic event)');

  };

  return (

    <div className="App">

      <header>

        <h1>Event Examples App</h1>

        <p>Demonstrating various React event handling techniques.</p>

      </header>

      <main>

        <section>

          <h2>Counter: {count}</h2>

          <div className="button-group">

            <button

              onClick={() => {

                handleIncrement();

                handleSayHello();

              }}

            >

              Increment (Multi-method)

            </button>

            <button onClick={handleDecrement}>

              Decrement

            </button>

          </div>

        </section>

        <section>

          <h2>Passing Arguments to Event Handlers</h2>

          <div className="button-group">

            <button onClick={() => handleSayWelcome('Welcome!')}>

              Say Welcome

            </button>

          </div>

        </section>

        <section>

          <h2>Synthetic Event (onClick)</h2>

          <div className="button-group">

            <button onClick={handleSyntheticEvent}>

              OnPress (Synthetic Event)

            </button>

          </div>

        </section>

        <CurrencyConvertor />

      </main>

    </div>

  );

};

const CurrencyConvertor = () => {

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

  const [euros, setEuros] = useState(0);

  const exchangeRate = 85;

  const handleRupeesChange = (event) => {

    setRupees(event.target.value);

  };

  const handleSubmit = (event) => {

    event.preventDefault();

    const calculatedEuros = (parseFloat(rupees) / exchangeRate).toFixed(2);

    setEuros(calculatedEuros);

  };

  return (

    <section className="currency-convertor">

      <h2>Currency Convertor</h2>

      <form onSubmit={handleSubmit}>

        <div>

          <label htmlFor="rupees">Enter Amount in Rupees</label>

          <input

            id="rupees"

            type="number"

            value={rupees}

            onChange={handleRupeesChange}

            placeholder="e.g., 5000"

          />

        </div>

        <div className="result">

          <p>Converted to Euros:</p>

          <p className="euros">€ {euros}</p>

        </div>

        <button type="submit">

          Convert

        </button>

      </form>

    </section>

  );

};

export default App;

**Output:**





