Exercise 16: Demo about Event Handling

Objectives and Outcomes

This exercise is a demo that showcases event handling in React

Exercises

* Create a functional component called EventHandlingDemo. Inside the component, we use the useState hook to initialize and manage a count state variable.
* Define a function called handleButtonClick that increments the count state when the button is clicked. This function is called in the onClick event handler of the button.

import React, { useState } from 'react';

const EventHandlingDemo = () => {

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

const handleButtonClick = () => {

setCount(count + 1);

};

return (

<div>

<h1>Event Handling Demo</h1>

<p>Count: {count}</p>

<button onClick={handleButtonClick}>Increase Count</button>

</div>

);

};

export default EventHandlingDemo;

The return statement renders a simple UI that displays the current count and a button. When the button is clicked, the handleButtonClick function is called, updating the count and triggering a re-render of the component.

import React from 'react';

import EventHandlingDemo from './EventHandlingDemo';

const App = () => {

return (

<div>

<EventHandlingDemo />

</div>

);

};

export default App;

Conclusion

This demo showcases a simple event handling scenario in React. You can extend and modify it to handle different events and perform various actions based on user interaction.