Exercise 17: Demo about Render and Commit

Objectives and Outcomes

In this exercise, through a simple demo that illustrates the concepts of rendering and committing in React.

Exercises

* Create a new React component called RenderAndCommitDemo. This component will have a button that, when clicked, changes the state and triggers a re-rendering.

import React, {useState} from 'react';

const RenderAndCommitDemo = () => {

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

const handleClick = () => {

setCount(count + 1);

};

return (

<div>

<h1>Render and Commit Demo</h1>

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

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

</div>

);

};

export default RenderAndCommitDemo;

* Render the RenderAndCommitDemo component in the root of our application.

import React from 'react';

import './Component/RenderAndCommitDemo'

import RenderAndCommitDemo from './Component/RenderAndCommitDemo';

function App() {

return (

<React.StrictMode>

<RenderAndCommitDemo />

</React.StrictMode>

);

}

export default App;

Conclusion

In this demo, when the button is clicked and the state is updated, React triggers the reconciliation process. It compares the previous virtual DOM with the updated virtual DOM and determines that the count value has changed. React then applies the necessary changes to the actual DOM, updating the displayed count value on the screen.