SOMNOG 7
Somali Network Operators Group

WORKSHOPS AND CONFERENCE
22 - 26 December, 2024

THEME:

Digital Public Infrastructure: Laying the Foundations for Somalia's Digital Future

SOMNOG 7
Somali Network Operators Group
WORKSHOPS AND CONFERENCE
22 - 26 December, 2024

React js Fundamentals

A Comprehensive Guide to React Components, JSX, and State Management

Mohamed Abdirahman Ahmed – Software Developer At Tabaarak ICT and Rikaab

What is React.js?

- React.js is a JavaScript library for building dynamic and interactive user interfaces.
- History:
- - Created by Facebook in 2013.
- - Used in applications like Facebook, Instagram, and WhatsApp.
- Core Features:
- - Component-based architecture.
- - Virtual DOM for performance optimization.
- - Declarative programming style for predictable UI.

Why Use React.js?

- Key Benefits:
- - Reusable Components: Build once, use multiple times across the application.
- - Virtual DOM: Efficient updates and rendering for improved performance.
- - Unidirectional Data Flow: Easy debugging and predictable application state.
- - Rich Ecosystem: Includes tools like Redux, React Router, and more.
- - Community Support: Large developer community with abundant resources and libraries.

React Components

- What are Components?
- - Self-contained and reusable pieces of UI.
- - Accept inputs called 'props' and return React elements.
- Types of Components:
- - Functional Components: Simpler syntax, better performance, and support for hooks.
- - Class Components: Less common in modern React.

JSX (JavaScript XML)

- Definition:
- JSX is a syntax extension that allows mixing HTML with JavaScript.
- Features of JSX:
- - Makes React code more readable and maintainable.
- - Transpiled into JavaScript using tools like Babel.
- Example:
- const element = <h1>Hello, world!</h1>;

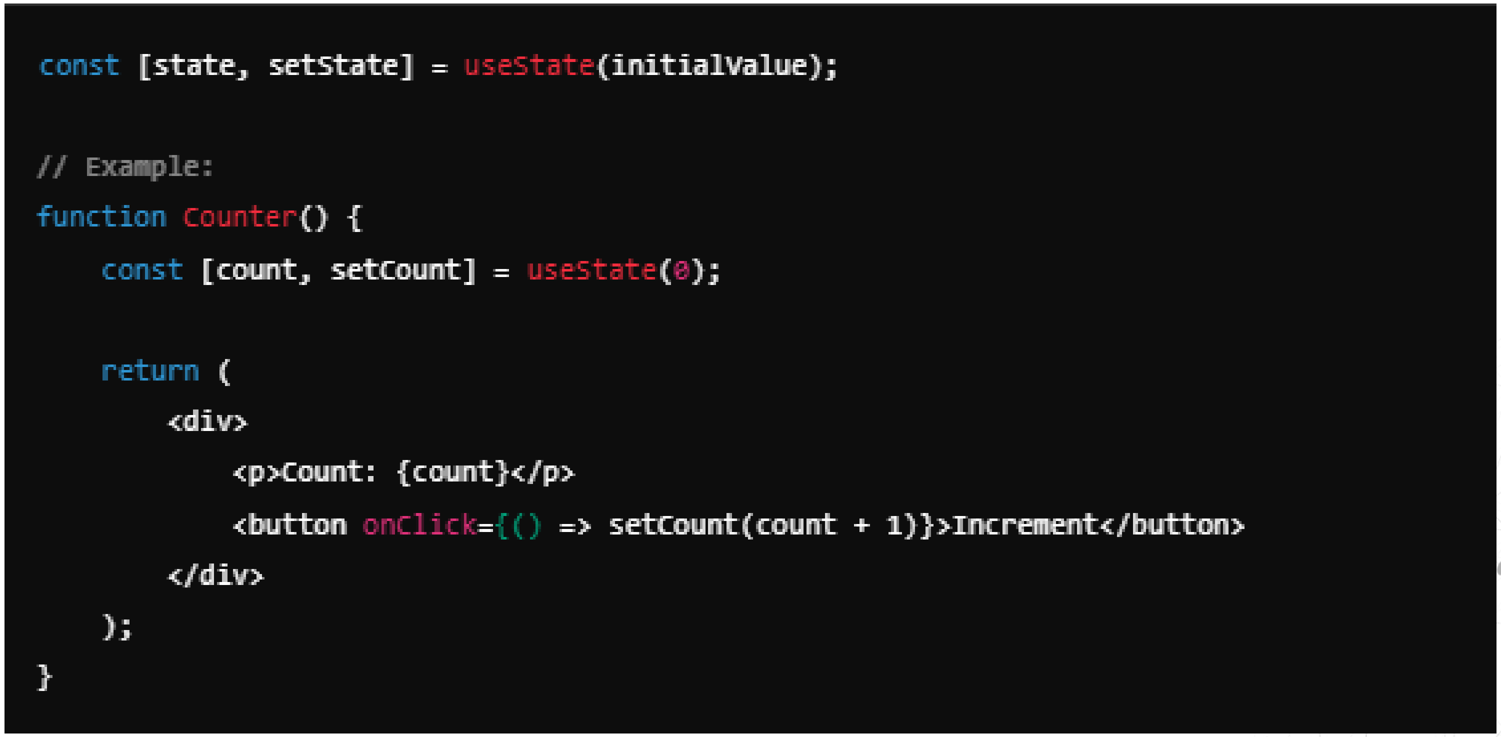
The Virtual DOM

- What is the Virtual DOM?
- - A lightweight copy of the real DOM.
- How It Works:
- 1. React creates a Virtual DOM tree when state changes.
- 2. Compares the new Virtual DOM with the previous one (diffing algorithm).
- 3. Updates only the changed parts of the real DOM.
- Advantages:
- - Faster updates.
- - Minimizes expensive DOM operations.
- - Improves application performance.

Managing State with useState

- What is State?
- - A React object that stores dynamic data and controls component behavior.
- useState Hook:
- - Allows adding state to functional components.

Managing State with useState

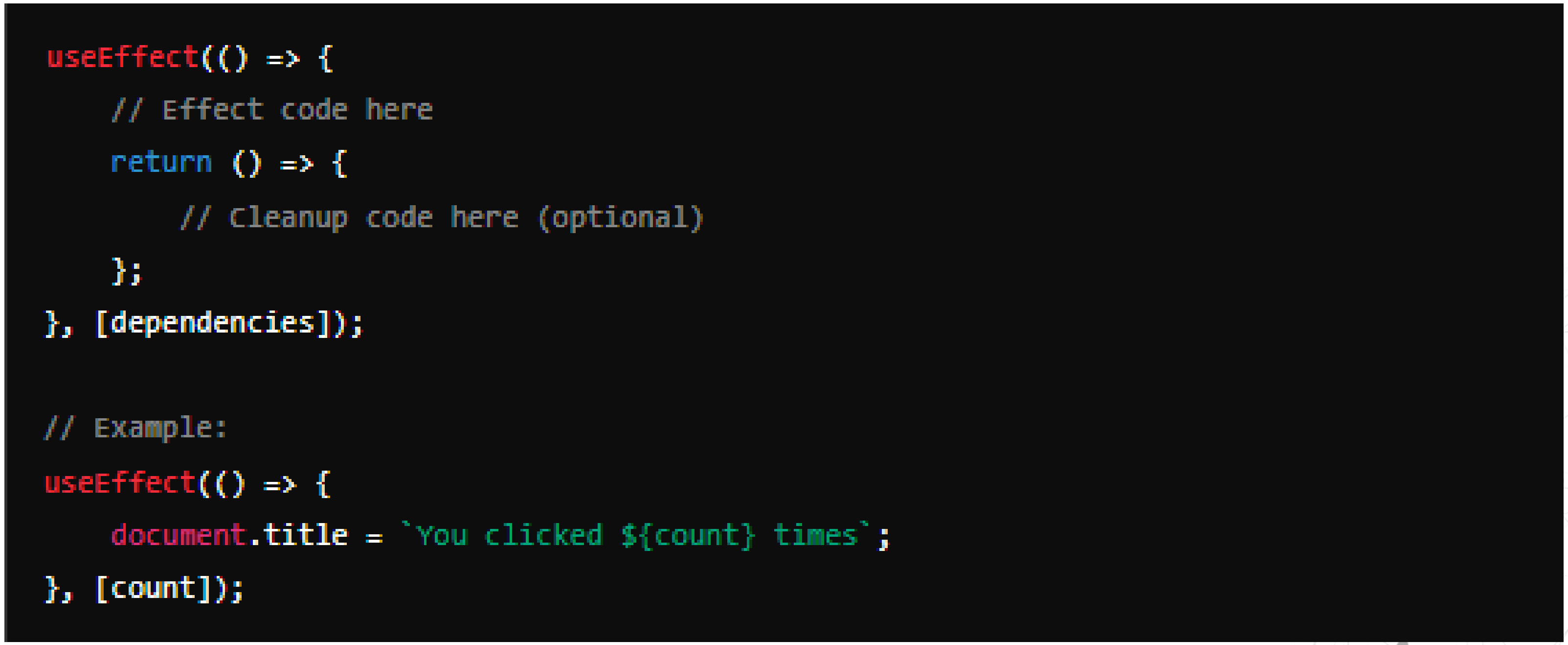


```
const [state, setState] = useState(initialValue);

// Example:
function Counter() {
    const [count, setCount] = useState(0);

    return (
        <div>
            <p>Count: {count}</p>
            <button onClick={() => setCount(count + 1)}>Increment</button>
        </div>
    );
}
```

Side Effects with useEffect

- What are Side Effects?
- - Operations like fetching data, subscribing to events, or directly interacting with the DOM.
- useEffect Hook:
- - Handles side effects in functional components.

Side Effects with useEffect



```
useEffect(() => {
    // Effect code here
    return () => {
        // Cleanup code here (optional)
    };
}, [dependencies]);

// Example:
useEffect(() => {
    document.title = `You clicked ${count} times`;
}, [count]);
```

Summary

- Key Takeaways:
- - React simplifies UI development with components, JSX, and the Virtual DOM.
- - State management is efficient with useState.
- - Side effects are handled using useEffect.
- - Modern React encourages using functional components with hooks.

Thank You!