Chapter 5: useEffect Hook

1 Introduction

Is chapter mein humne **useEffect** hook ke baare mein seekha, jo React mein side effects handle karta hai, jaise API calls, timers, ya DOM manipulation. Humne dependency array aur cleanup function bhi dekha.

2 5.1 What is useEffect?

useEffect ek React hook hai jo side effects ko handle karta hai, jo component ke bahar ke system se interact karte hain, jaise:

- API se data fetch karna.
- Timers set karna.
- Console mein log karna.

2.1 Syntax

```
import React, { useEffect } from 'react';

function Component() {
  useEffect(() => {
    // Side effect code yaha
  }, [/* dependencies */]);
}
```

2.2 Real-World Example

Ek food delivery app mein, jab app load hota hai, menu ka data API se fetch karna `useEffect` se hota hai.

3 5.2 Basic useEffect Example

Yeh example console mein message log karta hai jab component mount hota hai.

3.1 Code Example: Logging with useEffect

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>useEffect Example</title>
  <script
     src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js
  <script
     src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.develo
```

```
  <script
     src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body>
  <div id="root"></div>
  <script type="text/babel">
    function App() {
      React.useEffect(() => {
        console.log("Component mounted! Welcome to the app!");
      }, []); // Khali dependency array

      return (
        <div className="text-center p-4">
          <h1 className="text-3xl font-bold
             text-blue-600">useEffect Demo</h1>
          <p className="text-lg">Check the console for the
             message!</p>
        </div>
      );
    }

    const root =
       ReactDOM.createRoot(document.getElementById('root'));
    root.render(<App />);
  </script>
</body>
</html>
```

3.2 Explanation

- `useEffect(() => ..., [])`: Component mount hone pe ek baar chalta hai.
- Khali `[]` dependency array: Sirf mount pe chalta hai.
- Output: Console mein `Component mounted! Welcome to the app!`

4 5.3 useEffect with Dependencies

Dependency array mein variables daal kar `useEffect` ko specific state changes pe chala sakte hain.

4.1 Code Example: Counter with useEffect

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Counter with useEffect</title>
  <script
     src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js
```

```
  <script
     src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.develo
  <script
     src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body>
  <div id="root"></div>
  <script type="text/babel">
    function CounterApp() {
      const [count, setCount] = React.useState(0);

      React.useEffect(() => {
        console.log(`Count updated to: ${count}`);
      }, [count]);

      const handleIncrement = () => {
        setCount(count + 1);
      };

      return (
        <div className="text-center p-4">
          <h1 className="text-3xl font-bold
             text-blue-600">Counter with useEffect</h1>
          <p className="text-xl">Count: {count}</p>
          <button
            className="bg-green-500 text-white px-4 py-2 m-2
               rounded"
            onClick={handleIncrement}
          >
            Increase
          </button>
        </div>
      );
    }

    const root =
       ReactDOM.createRoot(document.getElementById('root'));
    root.render(<CounterApp />);
  </script>
</body>
</html>
```

4.2 Explanation

- `useEffect`: `count` change hone pe console mein log karta hai.
- `[count]`: Dependency array mein `count` hai, toh sirf `count` ke change pe chalta hai.
- Output: Console mein `Count updated to: 0`, `Count updated to: 1`, etc.

4.3 Real-World Example

Restaurant app mein order count badhne pe `useEffect` se total price calculate kar sakte hain.

5 5.4 Cleanup in useEffect

Cleanup function resources (jaise timers) ko free karta hai jab component unmount hota hai, taaki memory leaks na ho.

5.1 Code Example: Timer with Cleanup

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Timer with useEffect</title>
  <script
     src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js
  <script
     src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.develo
  <script
     src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body>
  <div id="root"></div>
  <script type="text/babel">
    function TimerApp() {
      const [seconds, setSeconds] = React.useState(0);

      React.useEffect(() => {
        const interval = setInterval(() => {
          setSeconds(seconds => seconds + 1);
        }, 1000);

        // Cleanup function
        return () => {
          clearInterval(interval);
          console.log("Timer cleared!");
        };
      }, []);

      return (
        <div className="text-center p-4">
          <h1 className="text-3xl font-bold text-blue-600">Timer
             App</h1>
          <p className="text-xl">Seconds: {seconds}</p>
        </div>
      );
    }
```

```
36
37      const root =
           ReactDOM.createRoot(document.getElementById('root'));
38      root.render(<TimerApp />);
39    </script>
40  </body>
41  </html>
```

5.2 Explanation

- `setInterval`: Har second `seconds` state badhata hai.
- `return () => clearInterval(interval)`: Component unmount hone pe timer clear karta hai.
- Output: Screen pe `Seconds: 0, 1, 2...`, console mein `Timer cleared!` (unmount pe).

6 Common Mistakes

- **Dependency Array Bhoolna**: Bina array ke `useEffect` har render pe chalta hai, jo performance issues deta hai.
- **Cleanup Na Karna**: Timers ya subscriptions ke liye cleanup nahi kiya toh memory leaks ho sakte hain.
- **Wrong Dependencies**: Dependency array mein wohi variables daalo jo `useEffect` mein use ho rahe hain.

7 Interview Tips

- **useEffect kya hai?**
  - Hook jo side effects handle karta hai, jaise API calls, timers.
- **Dependency array ka role?**
  - Control karta hai kab `useEffect` chalega. Khali `[]` matlab sirf mount pe.
- **Cleanup kyun zaroori?**
  - Resources free karta hai taaki memory leaks na ho.

8 Assignment: Practice Time

8.1 Task 1: Welcome Message

- `Welcome` component banao jo `useEffect` se console mein "Welcome to my app!" log kare (mount pe).
- Tailwind se style karo.

8.2 Task 2: State Change Logger

- `Counter` component banao jo `useState` se count track kare.
- `useEffect` se har count change pe console mein log kare: "Count is now: [count]".
- Button se count badhao.

8.3 Task 3: Simple Timer

- `Timer` component banao jo har second number badhaye.
- `useEffect` mein `setInterval`, cleanup mein `clearInterval`.
- Seconds display karo, Tailwind se style karo.