

Day 13 – [7th July 2025]

TOPICS COVERED

Virtual DOM in React:

React uses a Virtual DOM, which is a lightweight JavaScript copy of the actual DOM.

- Instead of updating the whole DOM every time a change occurs, React:
- Creates a virtual representation of the UI
- Compares it with the previous virtual DOM (diffing)
- Updates only the changed parts in the real DOM (reconciliation)

This makes React apps faster and more efficient.

State Variables in React:

In React, we use the `useState` hook to declare state variables that hold dynamic data.

Example:

```
import { useState } from "react";

function App() {

  const [count, setCount] = useState(0); // count is a state variable

  return <button onClick={() => setCount(count + 1)}>Clicked {count}</button>;

}
```

Every time we update a state variable using its setter function (`setCount`), React re-renders the component to reflect the latest state.

Rendering and Mounting Phase:

React components go through several lifecycle phases:

- Mounting Phase – When the component is inserted into the DOM for the first time.
- Rendering – Whenever the component or its state/props change, React re-renders the UI.
- Updating – Happens after a state/prop change.
- Unmounting – When the component is removed from the DOM.

In functional components, mounting and rendering are handled internally, and we can use the `useEffect()` hook to perform tasks during lifecycle events.

Reconciliation in React:

Reconciliation is the process React uses to compare the previous virtual DOM with the new one after a state or prop update.

It identifies what changed and updates only those specific parts in the actual DOM.

React uses a key property in lists to identify individual items during reconciliation efficiently.

This diffing and updating system is what keeps React apps fast and optimized.

Import and Export in React:

React supports two types of importing and exporting modules or components: default and named.

Default Export & Import:

Used when a file exports a single main value or component.

Example:

```
// File: MyComponent.js  
  
export default function MyComponent() { return <h1>Hello</h1>; }  
  
// Import:  
  
import MyComponent from './MyComponent';
```

Named Export & Import:

Used when a file exports multiple values or components.

Example:

```
// File: utils.js  
  
export const add = (a, b) => a + b;  
export const subtract = (a, b) => a - b;  
  
// Import:  
  
import { add, subtract } from './utils';  
  
We can also rename imports using as:  
  
import { add as addition } from './utils';
```

TOOLS USED:

Visual Studio Code

React with Vite

Chrome DevTools

Node.js + npm

TASK:

Read about:

- Props
- useEffect
- API calling using useEffect