Training Day 13

7th July 2025

TOPICS COVERED

- **Reconciliation in React (JavaScript)**

Reconciliation is the process React uses to update the DOM efficiently.

When your component's state or props change:

React creates a new Virtual DOM tree based on the new state/props.

It compares this new tree with the previous Virtual DOM using a process called diffing.

React then updates only the parts of the real DOM that actually changed.

Why is this efficient?

Updating the real DOM is expensive and slow.

React minimizes DOM operations by batching and diffing updates.

Example of Reconciliation:

```
function App() {
  const [count, setCount] = React.useState(0);
  return (
    <div>
      <h1>Hello</h1>
      <p>{count}</p>
      <button onClick={() => setCount(count + 1)}>Add</button>
    </div>
  );
}
```

When setCount is called, React:

Creates a new virtual DOM with updated count

Diffs it with the old one

Sees that only <p> changed → updates just that part in the real DOM

- **Asynchronous Nature of useState**

The useState update is asynchronous — meaning it doesn’t update the state immediately after calling the setter (setStateFunction) inside a component.

Example:

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

function handleClick() {
  setCount(count + 1);
  console.log(count); // Still shows old value!
}
```

count is still the old value because React hasn't updated the state yet.

Why is it async?

React batches multiple state updates together to optimize performance.

Updating the state immediately would cause too many re-renders.