Training Day 12

4th July 2025

TOPICS COVERED

- **Destructuring in JavaScript**

Destructuring is a JavaScript expression that allows you to unpack values from arrays or properties from objects into distinct variables. It helps write cleaner and more readable code, especially when working with complex data structures like objects and arrays.

1. Array Destructuring

You can extract values from an array and assign them to variables in one line.

```
const numbers = [10, 20, 30];

const [a, b, c] = numbers;

console.log(a); // 10

console.log(b); // 20

console.log(c); // 30
```

Skip Elements:

```
const [x, , z] = [1, 2, 3];

console.log(z); // 3
```

Default Values:

```
const [p = 5, q = 10] = [7];

console.log(p); // 7

console.log(q); // 10
```

Rest operator:

```
const numbers = [1, 2, 3, 4, 5];

const [first, second, ...rest] = numbers;

console.log(first);  // 1

console.log(second); // 2
```

```
console.log(rest);   // [3, 4, 5]
```

2. Object Destructuring

Allows you to extract properties from an object into individual variables.

```
const user = {
  name: "Raghav",
  age: 24,
  country: "India"
};

const { name, age } = user;
console.log(name); // Raghav
console.log(age);  // 24
```

Renaming Variables:

```
const { name: username } = user;
console.log(username); // Raghav
```

Default Values:

```
const { email = "Not provided" } = user;
console.log(email); // Not provided
```

- **Why Use Destructuring?**

Reduces code clutter

Makes accessing object/array properties concise

Useful with APIs, state management, and component props in React

- **What is useState in React?**

useState is a Hook that allows you to add state (i.e., data that can change over time) to functional components.

Before hooks, only class components could manage state. Now, with useState, functional components can do it too.

Syntax of useState:

const [stateVariable, setStateFunction] = useState(initialValue);

stateVariable: current state value

setStateFunction: function used to update the state

initialValue: the starting value for the state

Example: Counter App

```
import React, { useState } from 'react';
function Counter() {
  const [count, setCount] = useState(0);
  return (
    <div>
      <h2>Count: {count}</h2>
      <button onClick={() => setCount(count + 1)}>Increment</button>
    </div>
  );
}
```

What's happening:

count starts at 0

Clicking the button updates count using setCount

React automatically re-renders the component with the new state

TOOLS USED

Visual Studio Code (VS Code)

Chrome Browser (JavaScript Console)