Module 18) Reactjs for Full Stack

Introduction to React.js

Question 1: What is React.js? How is it different from other JavaScript frameworks and libraries?

ANS:- React.js is a JavaScript library for building user interfaces, particularly single-page applications where performance and dynamic. React uses a UI development and virtual DOM to optimize rendering performance. because it's a lightweight UI library focused on building user interfaces using components and JSX syntax**.**

Question 2: Explain the core principles of React such as the virtual DOM and componentbased architecture.

ANS:- React works on two main ideas: Virtual DOM and Component-based architecture. The Virtual DOM is like a copy of the real webpage. Component-based architecture means the whole UI is made up of small, reusable pieces called components.

Question 3: What are the advantages of using React.js in web development?

ANS:- React.js is popular in web development because it’s fast, easy to use, and lets developers build dynamic user interfaces using reusable components with efficient updates through the virtual DOM, making apps more responsive, maintainable, and scalable.

2. JSX (JavaScript XML)

Question 1: What is JSX in React.js? Why is it used?

ANS:- JSX (JavaScript XML) is a syntax extension for JavaScript used in React.js. It allows you to write HTML-like code inside your JavaScript code. JSX makes the structure of the user interface (UI) easier to visualize and write.

Question 2: How is JSX different from regular JavaScript? Can you write JavaScript inside JSX?

ANS:- JSX is not regular JavaScript—it is a syntax extension of JavaScript that allows you to write HTML-like tags inside JavaScript code. This syntax is not understood by browsers, so it must be transpiled (converted) into regular JavaScript using tools like Babel before execution.

Question 3: Discuss the importance of using curly braces {} in JSX expressions.

ANS:- In JSX, curly braces `{}` are used to embed JavaScript expressions inside HTML-like code. They allow you to dynamically insert values, variables, function calls, or expressions into the JSX structure.

3. Components (Functional & Class Components)

Question 1: What are components in React? Explain the difference between functional components and class components.

ANS:-

Feature	Functional Component	Class Component
Definition	Function	ES6 Class extending `React.Component`
Syntax	Simple, concise	Verbose
State Management	Using Hooks (`useState`)	Using `this.state` and `this.setState()`
Lifecycle Methods	Using Hooks (`useEffect`)	Built-in lifecycle methods
`this` keyword	Not used	Required
Performance	Slightly better (lighter)	Slightly heavier (more boilerplate)
Modern Usage	Recommended	Legacy (still valid but less preferred)

Question 2: How do you pass data to a component using props?

ANS:- React in pass data to a component using props. Props are like arguments you pass to a function. they allow you to customize components by supplying external data.

Question 3: What is the role of render() in class components?

ANS:- The render() method in React class components is responsible for returning the JSX that defines the UI of the component. It is called automatically whenever the component's state or props change.

4. Props and State

Question 1: What are props in React.js? How are props different from state?

ANS:- Props (short for properties) are read-only inputs that are passed from a parent component to a child component in React. They allow components to be dynamic and reusable by accepting external data.

Question 2: Explain the concept of state in React and how it is used to manage component data.

ANS:- In React, state is a built-in object that allows components to store and manage dynamic data. It is used to control the behavior and rendering of a component based on internal or user-driven changes.

Unlike props (which are passed from parent to child and are read-only), state is local to the component and can be updated over time.

Question 3: Why is this.setState() used in class components, and how does it work?

ANS:- In React class components, `this.setState()` is used to update the component's state. It is the only correct way to change the value of `state` and trigger a re-render of the component with updated data.

> Directly modifying state (e.g., `this.state.count = 5`) is not allowed because it won't trigger a re-render, and it can lead to unpredictable behavior.