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

Ans) It is an open-source javascript library developed by facebook for building user interfaces, particularly for single-page applications (spas). It is used for handling the view layer in web and mobile apps, allow us to create reusable UI components.

1. Components: React applications are made up of components, which are contains modules that render some output.
2. JSX: React uses JSX JavaScript XML a syntax extension that allows you to write HTML structures in the same file that contains JavaScript code. This makes the code easier to understand and debug, as it avoids the usage of complex JavaScript DOM structures.
3. Virtual DOM: React creates a virtual DOM in memory, where it does all the necessary manipulating, before making the changes in the browser DOM. This approach improves performance because it minimizes the number of costly DOM operations.
4. : Explain the core principles of React such as the virtual DOM and componentbased architecture

Ans)1)Virtual DOM directly manipulating the real DOM which is slow and expensive React works with the Virtual DOM to calculate the most efficient way to update the UI.

Ii)Initial Render: When a React component is rendered, it creates a Virtual DOM tree that creates the structure of the actual DOM.

Iii)State/Props Change: When the state or props of a component change, React creates a new Virtual DOM tree.

Component-Based Architecture React is built on the of components, which are reusable which contains pieces of UI. This promotes modules, reusability and maintainability.

Component- A component is a JavaScript function or class that returns a piece of UI written in JSX.

Components can be nested within other components to build complex UIs.

Ex.

function Greeting(props) {

return <h1>Hello, {props.name}!</h1>;

}

Class Components: Written as ES6 classes. Can manage state and lifecycle methods.

Ex.

class Greeting extends React.Component {

render() {

return

<h1>Hello, {this.props.name}!</h1>; } }

Reusability: Components can be reused across different parts of an application or even in different projects.

Ex.: A Button component can be used in multiple places with different text or styles.

1. What are the advantages of using React.js in web development?

Ans)Component-Based Architecture Reusability: Components are reusable reducing code duplication and improving maintainability.

Modularity: Breaking the UI into smaller components makes the code easier to manage and consise.

Encapsulation: Each component manages its own logic, state, and UI, making debugging and testing easier.

Virtual DOM for Improved Performance React's Virtual DOM minimizes direct manipulation of the real DOM, which is slow and expensive. React makes faster rendering and a smoother user experience.