REACT
INTERVIEW + CODING
IMPORTANT QUESTIONS

Q 1. What is React?

Ans: React is an open-source JavaScript library for building user interfaces. It was developed and is maintained by Facebook and a community of individual developers and companies.

React is commonly used for building web applications and user interfaces, but it can also be used for mobile app development.

It allows developers to create reusable and interactive user interface components, making it easier to build complex and dynamic front-end applications.

Q 2. What are the features of React?

Ans: These are the important features of react:

- **Component-Based:** React is built around the concept of reusable UI components, making it easy to manage and maintain code.
- **Virtual DOM:** React's virtual DOM efficiently updates the actual DOM, improving rendering performance.
- **JSX:** JSX allows developers to write HTML-like code within JavaScript for defining component structures.

- **Unidirectional Data Flow:** Data flows in one direction, simplifying data management and updates.
- **Reusability:** React components can be reused across the application, promoting code reusability.
- **Declarative:** React allows you to describe what your UI should look like, and it takes care of updating the DOM accordingly.
- **Community and Ecosystem:** React has a large and active developer community and a rich ecosystem of libraries and tools.
- **Performance Optimization:** React provides tools for optimizing performance, such as memoization and code splitting.
- **Support for Server-Side Rendering (SSR):** React can render on the server side, improving SEO and initial load times.
- **Developer Tools:** React offers browser extensions and tools for debugging and inspecting components.
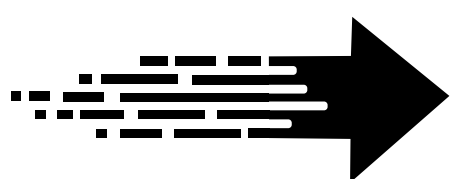
Explore Our Popular Courses

</> Data Structure & Algorithms
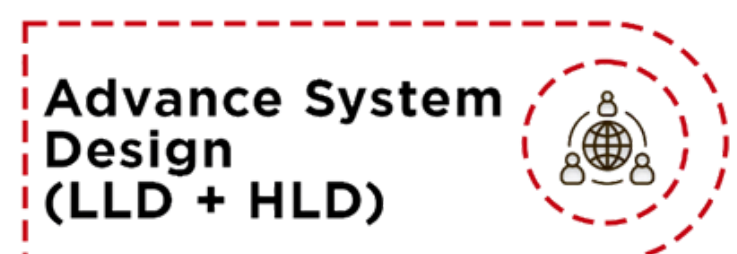
Advance System Design (LLD + HLD)

Advanced Data Science & Machine Learning

MERN Full Stack Development

Q 3. What is the difference between React and other JavaScript frameworks?

Ans: React is a JavaScript library for building user interfaces. It is declarative, efficient, and flexible. React is used by companies like Facebook, Instagram, and Netflix to build complex web applications.

Other popular JavaScript frameworks include Angular and Vue.js. These frameworks also have their own strengths and weaknesses, but here are some of the key differences between React and other JavaScript frameworks:

Feature	React	Angular	Vue.js
Type	Library	Framework	Framework
Declarative approach	Yes	Yes	Yes
Virtual DOM	Yes	No	Yes
Component-based architecture	Yes	Yes	Yes
JSX	Optional	Required	Optional
Learning curve	Easier	Steeper	Easier
Performance	Faster	Slower	Faster
Best suited for	SPAs, mobile apps, websites	Complex enterprise applications	Small to medium-sized applications

Q 4. What is JSX?

Ans: JSX stands for JavaScript XML. It is a syntax extension for JavaScript that allows you to write HTML-like code in your JavaScript.
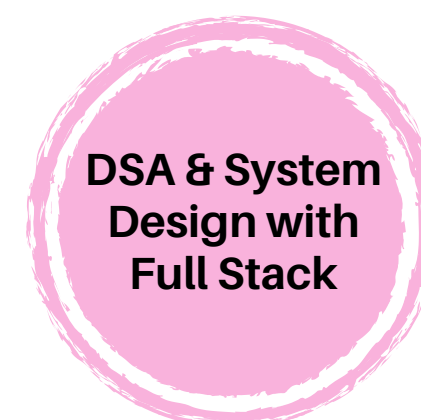
JSX is not required to use React, but it is highly recommended. JSX makes it easier to write and understand your UI code, as you don't have to switch between two separate languages.

Q 5. What is the Virtual DOM?

Ans: The Virtual DOM is a lightweight representation of the real DOM. It is a JavaScript object that describes the current state of the UI.

When the state of the UI changes, React updates the Virtual DOM. Then, React compares the Virtual DOM to the real DOM and efficiently updates the real DOM only with the changes that are needed.

Want to up your Skill?
Join our Popular courses

DSA with System Design (HLD + LLD)

Advanced Data Science & Machine Learning

MERN Full Stack Web Development

DSA & System Design with Full Stack

Q 6. How does the Virtual DOM work?

Ans: The Virtual DOM is a JavaScript object that describes the current state of the UI. It is a tree structure, with each node in the tree representing an element in the UI. The Virtual DOM also includes information about the attributes and children of each element.

To update the real DOM, React uses a process called reconciliation. Reconciliation compares the Virtual DOM to the real DOM and determines the minimal set of changes needed to bring them into alignment. React then applies these changes to the real DOM.

Reconciliation is a very efficient process. React uses a diffing algorithm to compare the Virtual DOM to the real DOM. This algorithm is able to identify the smallest possible set of changes needed to update the real DOM.

Virtual DOM
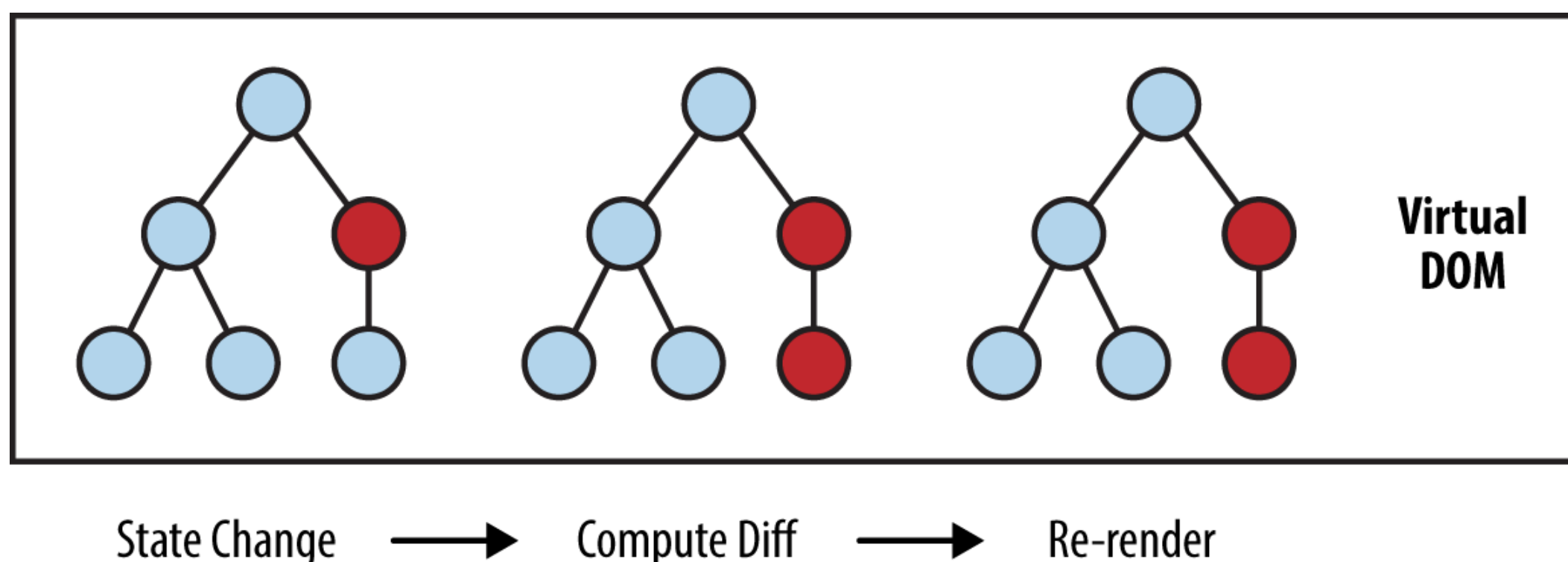
State Change → Compute Diff → Re-render

Why Choose **AlgoTutor**?
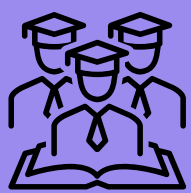
100 % PLACEMENT ASSISTANCE

1:1 PERSONAL MENTORSHIP FROM INDUSTRY EXPERTS

100 % SUCCESS RATE

23 LPA(AVG.)CTC

200+ SUCCESSFUL ALUMNI

147%(AVG.) SALARY HIKE

LEARN FROM SCRATCH

CAREER SERVICES

EXPLORE MORE

+91-7260058093   www.algotutor.io   info@algotutor.io

Q 7. What are the benefits of using the Virtual DOM?

Ans: The benefits of using the Virtual DOM in web development include:

- Improved performance by reducing direct manipulation of the real DOM.
- Efficient updates through batched and optimized changes.
- Cross-platform compatibility in various environments.
- Easier abstraction, as it abstracts browser differences.
- Declarative programming for more predictable code.
- Fine-grained control over rendering for optimized performance.
- Integration with third-party libraries.
- Simplified testing and debugging.
- Encouragement of component reusability for modular development.

For Admission Enquiry
+91-7260058093
info@algotutor.io

Q 8. What are the different types of components in React?

Ans: There are two main types of components in React:

- **Functional components:** Functional components are pure functions that take props as input and return an element as output. They are the simplest and most common type of component in React.
- **Class components:** Class components are more complex than functional components, but they offer more features, such as state management and lifecycle hooks.

Q 9. What is the difference between state and props?

Ans: State is data that is local to a React component and can be changed by the component itself.

Props are data that are passed to a React component from its parent component and cannot be changed by the child component.

Here is a table that summarizes the key differences between state and props:

Feature	State	Props
Definition	Data that is local to a React component and can be changed by the component itself	Data that are passed to a React component from its parent component and cannot be changed by the child component
Mutable	Yes	No
Owned by	Component	Parent component
Updated by	Component	Parent component

Q 10. How do you pass props from one component to another?

Ans: To pass props from one component to another in React, we need to use the props attribute.

The props attribute is an object that contains the data that you want to pass to the child component.

Q 11. What is the one-way data flow in React?

Ans: One-way data flow in React is a design pattern that ensures that data flows in one direction, from parent to child components.

This helps to make React applications more predictable and easier to debug.

Q 12. **What are the different lifecycle methods in React?**

Ans: There are three main phases in the React component lifecycle:
- Mounting: This is the phase when a component is first created and inserted into the DOM.
- Updating: This is the phase when a component's state or props change.
- Unmounting: This is the phase when a component is removed from the DOM.

ReactJS Component's Lifecycle
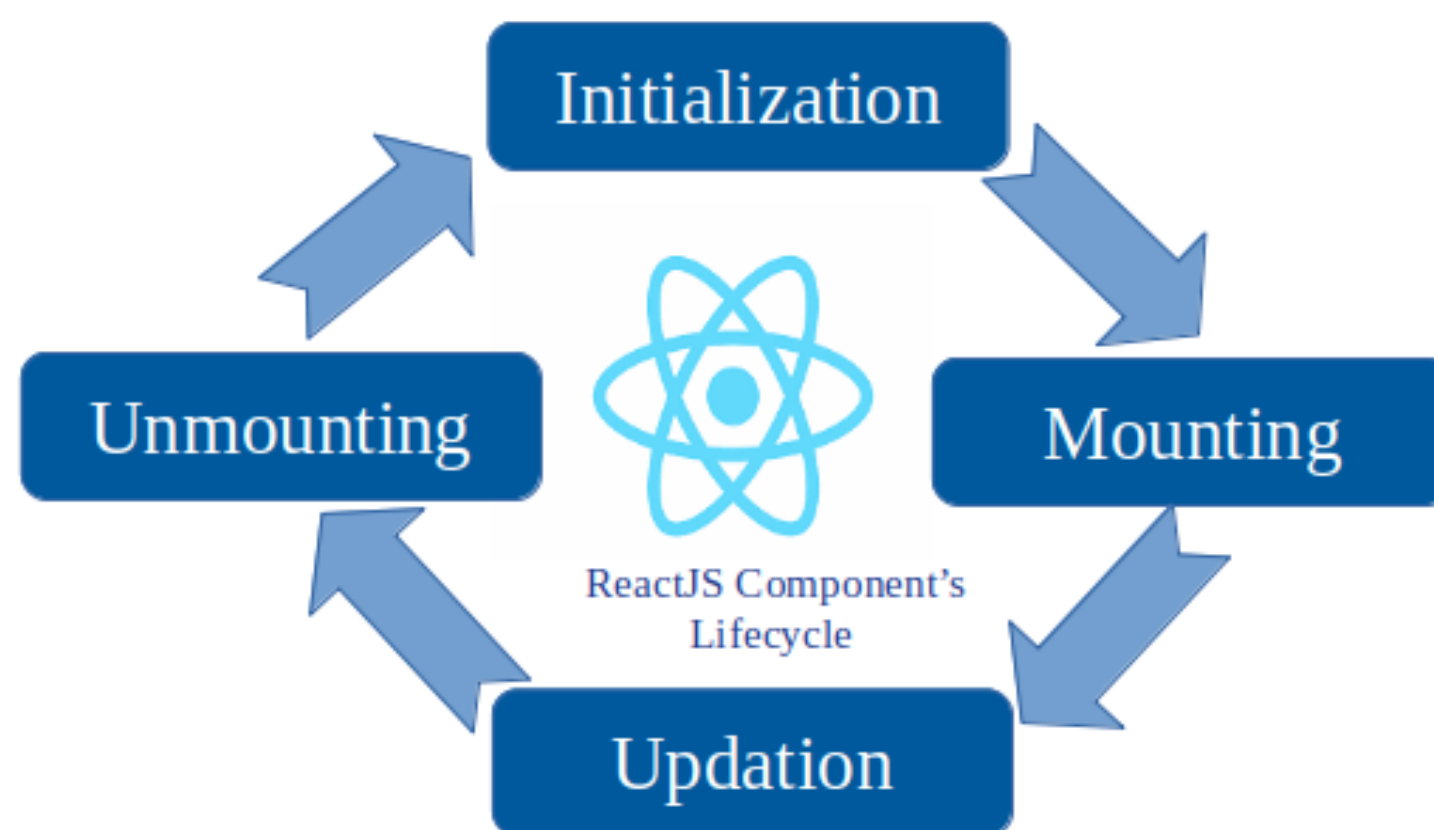
Q 13. **What are the benefits of using React hooks?**

Ans: React hooks are a feature introduced in React 16.8 that allow you to use state and other React features in functional components.

They offer several benefits over the older class components and are now the recommended way to work with state and side-effects in React applications.

Here are some of the key benefits of using React hooks:

- **Simpler code:** Hooks allow you to write more concise and readable code by avoiding the need to write a class.
- **Improved reusability:** Hooks can be easily reused across different components, making your code more modular and maintainable.
- **Easier testing and debugging:** Hooks are easier to test and debug than class components, as they are simply functions that can be called in isolation.
- **Reduced bundle size:** Hooks can help to reduce the bundle size of your application, as they are smaller and more efficient than class components.

Q 14. What is the difference between a controlled and uncontrolled component?

Ans: The main difference between a controlled and uncontrolled component is how the component handles its state.

Controlled components use React state to manage their state. This means that the parent component is responsible for updating the state of the controlled component.

Uncontrolled components do not use React state to manage their state. Instead, they manage their own internal state.

Q 15. What is the best way to handle events in React?

Ans: The best way to handle events in React is by using arrow functions or binding event handlers in the constructor, or by using class properties (property initializer syntax) for class components.

For functional components, you can use the useState and useCallback hooks to handle events efficiently and prevent unnecessary re-renders.
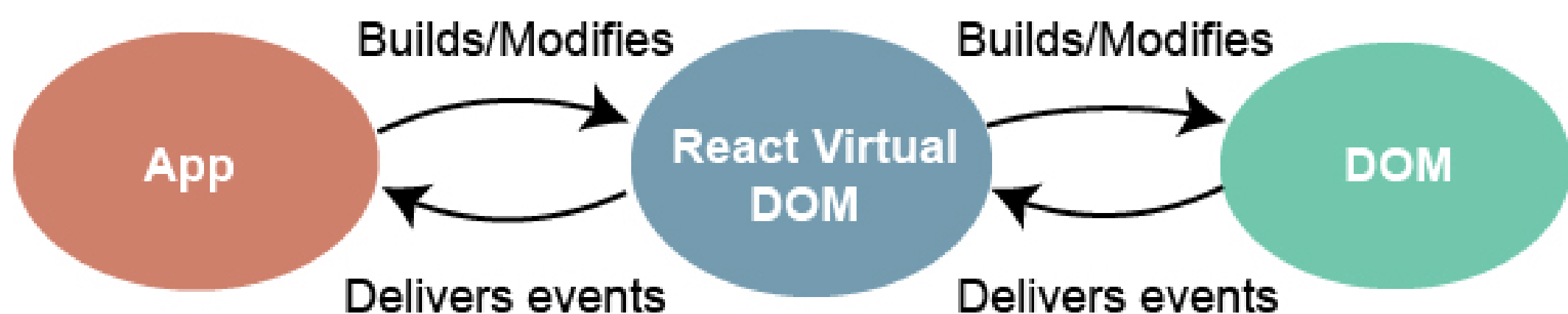
Q 16. What is the best way to manage state in React?

Ans: The best way to manage state in React is by using React hooks like useState and useReducer in functional components and by using the built-in this.state and this.

setState methods in class components. Use context API or third-party libraries like Redux for global state management when needed.

Q 17. What are some common performance optimization techniques in React?

Ans: Here are some common performance optimization techniques in React:

- Use React.memo() and useCallback() to prevent unnecessary re-renders.
- Use code splitting to load code only when it is needed.
- Use lazy loading to load images and other resources only when they are needed.
- Use React.PureComponent to prevent unnecessary re-renders.
- Use immutable data structures.
- Use a performance profiling tool to identify performance bottlenecks.

Q 18. What are some of the most popular React libraries and tools?

Ans: Some popular React libraries and tools include Redux, React Router, Axios, Styled-components, Material-UI, Formik, Jest, Enzyme, Storybook, and Webpack.

These tools assist with state management, routing, HTTP requests, styling, UI components, form handling, testing, documentation, and build processes in React applications.

Q 19. **What are some of the challenges of developing React applications?**

Ans: Challenges of developing React applications include a learning curve, state management complexities, performance optimization, build tool configuration, choosing the right libraries, prop drilling, SEO for SPAs, accessibility concerns, global CSS management, maintenance, cross-browser compatibility, error handling, and deployment/setup complexities.

Q 20. **Explain the concept of conditional rendering in React.**

Ans: Conditional rendering in React allows you to show or hide components or content based on certain conditions. You can use JavaScript logic and conditional statements to determine what should be rendered in response to changes in state or props.

Q 21. **What is a higher-order component (HOC)?**

Ans: A higher-order component (HOC) is a function that takes a component as an argument and returns a new component.

HOCs are used to add additional functionality to components without modifying the components themselves.

Q 22. What is context?

Ans: Context is a way for React components to share data without having to pass props down through the component tree.

This can be useful for sharing data that is needed by many components, such as the current user or the current locale.

Q 23. What is error handling in React?

Ans: Error handling in React is the process of handling errors that occur in React applications. This can include errors that occur during rendering, in lifecycle methods, or in event handlers.

Q 24. What is testing in React?

Ans: Testing in React is the process of verifying that your React components and applications work as expected.

This can be done by writing automated tests that run your code and check for errors or unexpected behavior.

Q 25. What is internationalization (i18n) in React?

Ans: Internationalization (i18n) is the process of making software applications adaptable to different locales and languages.

This includes translating the text in the application, as well as formatting dates, numbers, and currencies according to the locale.

Q 26. What is the significance of the key prop in React?

Ans: The key prop in React is significant because it helps React identify which items in a list have changed, are added, or are removed. This allows React to efficiently update the UI with the most recent changes, rather than having to update the entire list.

The key prop should be a unique identifier for each item in the list. It can be a string, number, or object. It is important to note that the key prop should be unique for each item in the list, even if the items themselves are identical.

Q 27. What is the difference between React and React Native ?

Ans: React and React Native are two different but related technologies.

React is a JavaScript library for building user interfaces. It is declarative, efficient, and flexible. React is used to build the user interface of web applications, such as Facebook, Instagram, and Airbnb.

React Native is a framework for building native mobile applications for iOS and Android using React. It allows developers to use React, a popular JavaScript library for building user interfaces, to create mobile applications that look and feel truly native.

Both React and React Native are based on the React component model, which means that mobile applications are built using reusable components. This makes development faster and easier, and it also makes it easier to share code between different platforms.
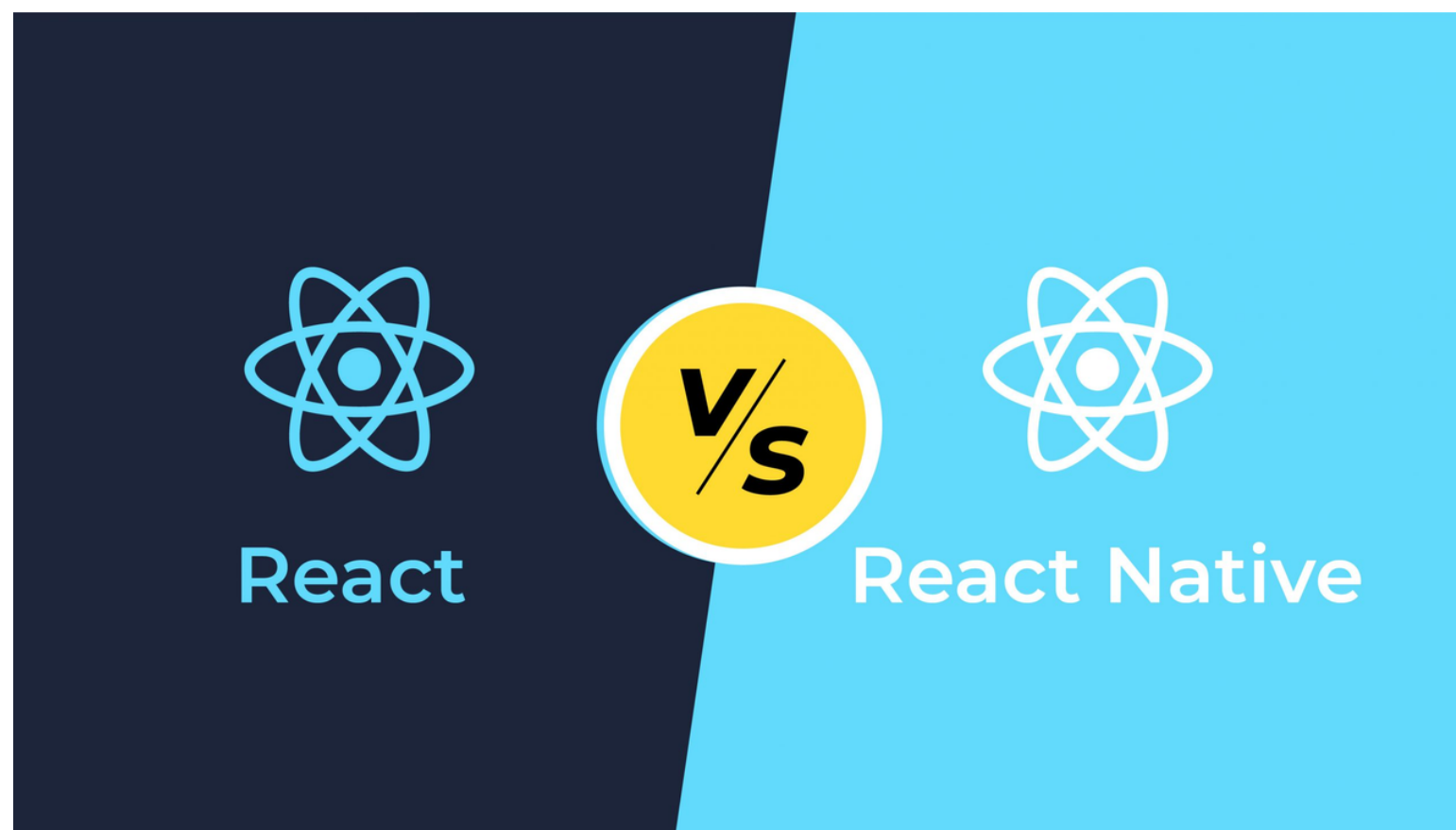
Feature	React	React Native
Platform	Web	iOS and Android
JavaScript library	Yes	Yes
Native components	No	Yes
Performance	Very good	Excellent
Community	Very large and active	Large and active

For Admission Enquiry

+91-7260058093

info@algotutor.io

There Are some Important React Coding Question

1. Write a React component to display a list of items.
2. Write a React component to handle user input.
3. Write a React component to fetch data from an API.
4. Write a React component to implement a simple routing system.
5. Write a React component to implement a state management pattern, such as Redux.
6. Write a React component to implement a common UI element, such as a button, input field, or modal.
7. Write a React component to implement a complex UI element, such as a table, chart, or map.
8. Write a React component to implement a performance optimization technique, such as memoization or lazy loading.
9. Write a React component to implement an accessibility feature, such as keyboard navigation or screen reader support.
10. Write a React Native component to implement a native UI element, such as a button, image, or list.
11. Write a React Native component to implement a complex native UI element, such as a map, navigation bar, or tab bar.
12. Write a React Native component to implement a performance optimization technique, such as bridging or asynchronous rendering.