Unlock all answers Utkarsh Jain

★★★★★

"Huge timesaver. Worth the money"

★★★★★

"It's an excellent tool"

★★★★★

"Fantastic catalogue of questions"

your next tech interview with

Explore our carefully curated catalog of interview essentials covering full-stack, data structures and algorithms, system design, data science, and machine learning interview questions

Start preparing now

Web & Mobile Dev | Data Structures & Algorithms | System Design | Machine Learning & Data Science

ADO.NET
Agile & Scrum
Android
Angular
AngularJS
ASP.NET
ASP.NET MVC
ASP.NET Web API
AWS
Azure
C++
C#
Cosmos DB
CSS
Dependency Injection
DevOps
Django
Entity Framework
Express.js
Flutter
GIT
Golang
GraphQL
HTML5
Ionic
Java
JavaScript
jQuery
Kotlin
Laravel
LINQ
MongoDB
.NET Core
Next.js
Node.js
Objective-C
OOP
PHP
PWA
Python
React
React Native
Reactive Programming
Reactive Systems
Redis
Redux

Ruby Ruby on Rails Rust Spring SQL Swift

T-SQL Testing TypeScript UX Design Vue.js

WCF Web Security WebSockets WPF Xamarin

100 React interview questions

Only coding challenges Reset progress Search...

Topic progress: 0%

React Basics

1. What is **React** and why is it used? Question ○

2. How is **React** different from **Angular** or **Vue**? Question ○

3. What is a **React component**? Question ○

4. How do you create a **component** in **React**? Question ○

5. What is **JSX** and why do we use it in **React**? Question ○

6. Can you explain the **virtual DOM** in **React**? Question ○

7. What are the differences between a **class component** and a **functional component**? Question ○

8. How do you handle **events** in **React**? Question ○

9. What are **state** and **props** in **React**? Question ○

10. How do you pass **data** between **components** in **React**? Question ○

React State Management

11. What is a **stateful component**? Question ○

12. Can you explain how **useState** works? Question ○

13. How do you update the **state** of a **parent component** from a **child component**? Question ○

14. What is **lifting state up** in **React**? Question ○

15. When do you use **Redux** or **Context API** for state management? Question ○

React Lifecycle & Hooks

16. Explain the **lifecycle methods** of a **React class component**. 🔒 Question ○

17. How do **hooks** work in **React**? 🔒 Question ○

18. Can you describe the **useEffect hook** and its purpose? 🔒 Question ○

19. How do you **fetch data** with **hooks** in **React**? 🔒 Question ○

20. What rules do you have to follow when using **hooks**? 🔒 Question ○

Component Communication

21. How do **props** work in **React**? 🔒 Question ○

22. What is **prop drilling** and how can you avoid it? 🔒 Question ○

23. Explain the **Context API** and its use cases. 🔒 Question ○

24. How do you use **render props**? 🔒 Question ○

25. What is the **children prop**? 🔒 Question ○

Performance Optimization

26. Why is **performance optimization** important in **React**? 🔒 Question ○

27. What is **React.memo** and when would you use it? 🔒 Question ○

28. How does **PureComponent** differ from **Component** in **React**? 🔒 Question ○

29. Can you explain the concept of **reconciliation** in **React**? 🔒 Question ○

30. How can you prevent unnecessary **re-renders** in **React**? 🔒 Question ○

Styling in React

31. How do you apply **styles** in a **React application**? 🔒 Question ○

32. What is **CSS-in-JS** and how do you implement it in **React**? 🔒 Question ○

33. Can you describe how **Styled-Components** work? 🔒 Question ○

34. What are the advantages of using **Sass** or **LESS** in a **React project**? 🔒 Question ○

35. How do you use **inline styles** in **React**? 🔒 Question ○

React Routing

36. What is **React Router**? 🔒 Question ○
37. How do you create **dynamic routes** in **React**? 🔒 Question ○
38. How would you pass **data** to **routes** in **React Router v5+**? 🔒 Question ○
39. How do you programmatically navigate using **React Router**? 🔒 Question ○
40. What are **route guards** and how can you implement them in **React**? 🔒 Question ○

React Patterns

41. What are **higher-order components (HOCs)**? 🔒 Question ○
42. Explain the **container/presenter** (smart/dumb) component pattern. 🔒 Question ○
43. How would you implement a **compound component pattern** in **React**? 🔒 Question ○
44. Explain the use of **custom hooks** in **React**. 🔒 Question ○
45. What is a **render prop pattern**? 🔒 Question ○

Form Handling

46. How do you handle **forms** in **React**? 🔒 Question ○
47. What is **controlled** and **uncontrolled components**? 🔒 Question ○
48. How do you **validate forms** in **React**? 🔒 Question ○
49. What is **Formik** and how is it used in **React forms**? 🔒 Question ○
50. How do you handle **file uploads** in **React**? 🔒 Question ○

React with TypeScript

51. What are the benefits of using **TypeScript** with **React**? 🔒 Question ○
52. How do you define **types** for **props** and **state** in **TypeScript** with **React**? 🔒 Question ○
53. Explain how to use **interfaces** with **React components** and **TypeScript**. 🔒 Question ○
54. How do **TypeScript generics** enhance **react components**? 🔒 Question ○

Testing in React

55. Why is **testing** important in **React**? 🔒 Question ○

56. What are some common **testing libraries** for **React**? 🔒 Question ○

57. How do you test a **React component** with **Jest**? 🔒 Question ○

58. Can you explain the difference between **shallow rendering** and **mount rendering** in **Enzyme**? 🔒 Question ○

59. What is **react-testing-library** and how is it different from **Enzyme**? 🔒 Question ○

Advanced React Topics

60. What are **React fragments** and why are they useful? 🔒 Question ○

61. What is **React portal** and when would you use it? 🔒 Question ○

62. How does **error boundary** work in **React**? 🔒 Question ○

63. What is **server-side rendering** and how is it done with **React**? 🔒 Question ○

64. Can you explain the concept of **suspense** and **lazy loading** in **React**? 🔒 Question ○

React and SEO

65. How does **React** affect **SEO**? 🔒 Question ○

66. What strategies would you use to make a **React application SEO-friendly**? 🔒 Question ○

67. How can **server-side rendering** improve **SEO** with **React applications**? 🔒 Question ○

React Native

68. What is **React Native** and how is it different from **React**? 🔒 Question ○

69. How do you bridge **native modules** in **React Native**? 🔒 Question ○

70. Can you describe the **layout system** in **React Native**? 🔒 Question ○

State Management Libraries & GraphQL

71. What is **Apollo Client** and how does it integrate with **React**? 🔒 Question ○

72. How do you manage **local state** in **Apollo Client**? 🔒 Question ○

73. What is **Redux** and how does it contrast with the **Context API**? 🔒 Question ○

74. Can you detail the **Redux workflow**? 🔒 Question ○

75. How do you handle **side effects** in **Redux applications**? 🔒 Question ○

React Development Environment & Tooling

76. How do you set up a **React project** from scratch? 🔒 Question ○

77. What is **Babel** and why do we use it with **React**? 🔒 Question ○

78. What is **Webpack** and what role does it play in **React development**? 🔒 Question ○

79. How does **hot module replacement** work in **React**? 🔒 Question ○

80. What are the features of **create-react-app** and how do you **eject** from it? 🔒 Question ○

Integrations and API Handling

81. How do you handle **API calls** in **React**? 🔒 Question ○

82. What is **Axios** and how is it used over **fetch** in **React applications**? 🔒 Question ○

83. How would you handle **WebSocket connections** in a **React application**? 🔒 Question ○

84. What are some strategies used to connect a **React front end** to a **backend server**? 🔒 Question ○

Deployment and Optimization

85. How would you deploy a **React application**? 🔒 Question ○

86. How do you optimize the performance of a **React application** for **production**? 🔒 Question ○

87. What are **service workers** and how can they benefit a **React application**? 🔒 Question ○

88. How do you configure **HTTPS** in a **React app**? 🔒 Question ○

Accessibility in React

89. Why is **accessibility** important in **web development**? 🔒 Question ○

90. How can you make a **React application** accessible? 🔒 Question ○

91. What is **ARIA** and how it is used in **React**? 🔒 Question ○

Internationalization and Localization

92. What is **internationalization (i18n)** in **React**? 🔒 Question ○

93. How do you implement **localization (l10n)** in a **React app**? 🔒 Question ○

React Code Structure & Best Practices

94. How do you structure large **React applications**? Question ○

95. What are some **best practices** when writing **React code**? Question ○

96. How do you ensure **code quality** and **maintainability** in a **React project**? Question ○

React and Git Workflows

97. How do you manage **feature branches** in **React development** with **Git**? Question ○

98. What are your strategies for resolving **merge conflicts** in **React projects**? Question ○

React Interviews Problem Solving and Scenarios

99. How would you handle a **feature request** or **bug report** in an ongoing **React project**? Question ○

100. Describe your process for optimizing a **component** that has complex **state logic** and several **child components**. Question ○

Unlock interview insights

Get the inside track on what to expect in your next interview. Access a collection of high quality technical interview questions with detailed answers to help you prepare for your next coding interview.

Track progress

Simple interface helps to track your learning progress. Easily navigate through the wide range of questions and focus on key topics you need for your interview success.

Save time

Save countless hours searching for information on hundreds of low-quality sites designed to drive traffic and make money from advertising.