CDAC Mumbai

Module: WPT
Topic: Interview Questions Set-3

1. What is Node.js, and how does it differ from traditional server-side technologies?
2. Explain the event-driven, non-blocking I/O model in Node.js and why it's beneficial.
3. How does the Node.js runtime handle multiple requests simultaneously with a single-threaded architecture?
4. What are some core modules in Node.js, and what are they used for? (e.g., fs, http, path)
5. What is npm, and how does it help in managing dependencies in Node.js projects?
6. Explain how you would create a basic HTTP server in Node.js. Provide example code.
7. What is the purpose of the package.json file, and what key information does it contain?
8. How do you handle asynchronous operations in Node.js? Explain the role of callbacks, promises, and async/await.
9. What are streams in Node.js, and how are they used? Provide an example.
10. Explain error handling in Node.js. How would you handle errors globally across an application?
11. What is ExpressJS, and why is it commonly used in web development?
12. How do you set up a basic ExpressJS server? Provide example code.
13. What is middleware in ExpressJS? How does it work?
14. How do you define routes in ExpressJS?
15. What is the difference between app.get(), app.post(), app.put(), and app.delete() in Express?
16. How can you handle errors in ExpressJS applications?
17. Explain how to set up a static file server in ExpressJS.
18. What is req.params vs. req.query in ExpressJS, and how are they used?
19. How can you use environment variables in Express applications?
20. How do you secure an ExpressJS application? Give examples of best practices.
21. What is AJAX, and how does it differ from a traditional HTTP request?
22. Explain the core steps in making an AJAX request using JavaScript.
23. What is the role of XMLHttpRequest in AJAX?
24. How do you perform an AJAX request with jQuery? Provide example code.
25. What are the different states of an XMLHttpRequest, and what do they mean?
26. Explain the role of JSON in AJAX. Why is JSON preferred?
27. What is fetch() in JavaScript, and how does it improve AJAX functionality?
28. How do you handle errors in AJAX requests using fetch?
29. Explain the concept of CORS (Cross-Origin Resource Sharing) in AJAX requests.
30. What is the difference between synchronous and asynchronous AJAX calls?
31. What is ReactJS, and what problem does it solve in web development?

32. What is JSX, and how does it differ from HTML?
33. Explain the React lifecycle methods in detail.
34. Explain the role of props in React components.
35. What is the purpose of state in a React component?
36. Differentiate between functional and class-based components in React.
37. How does React use the virtual DOM to improve performance?
38. What are React Hooks, and why were they introduced? Give examples of different hooks available in React.
39. What is the role of useState in React? How is it used?
40. Explain useEffect and provide an example of its usage.
41. What are controlled components in React, and why are they important for form handling?
42. What are the new features introduced in React 18?
43. How does one pass data between components in React?
44. How does React handle forms and controlled components?
45. What is useContext, and how does it simplify state management?
46. Explain the purpose of React's useReducer Hook and provide an example.
47. What is React Router, and how does it manage navigation in a React app?
48. How do you implement conditional rendering in React?
49. What is prop drilling, and how can it be minimized?
50. What is the purpose of React.memo()? How does it work?
51. Explain the role of React.forwardRef and when it should be used.
52. What is React.lazy() and Suspense? How do they support code-splitting?
53. How can you optimize performance in a React app?
54. What is higher-order component (HOC) in React? Provide an example.
55. How do you handle errors in React using error boundaries?
56. How does one implement lazy loading in React?
57. What is React Context API, and when would you use it?
58. What is Redux, and why would you use it in a React application?
59. Explain the concept of actions, reducers, and store in Redux.
60. How does the useSelector Hook work in a React-Redux application?
61. What is middleware in Redux? Give an example.
62. Explain the difference between useDispatch and useSelector.
63. What is the connect() function in Redux? How does it connect React to Redux?
64. Explain how to manage asynchronous operations in Redux.
65. How can you combine multiple reducers in a Redux app?
66. What is the purpose of Redux DevTools, and how do you set it up?