

Most asked question in javascript interview

1. Difference between javascript and other programming languages.
2. **Explain execution context in javascript, event loop works.**
3. Difference between let , const and var.
4. What are primitive and non primitive data types?
5. **What is the Nullish coalescing operator?**
6. What is the arrow function and how is it different from normal function?
7. What is IIFE?
8. **What is closures concept in javascript , explain with examples.**
9. What is this keyword?
10. What is prototype and prototype inheritance?
11. Difference between map, filter and find method.
12. **What is the difference between forEach and map method.**
13. What is the callback function? list down the use cases.
14. **What is callback hell? How to avoid it.**
15. What is a promise , explain with an example.
16. What are promise methods?
17. **What is the difference between promise.race and promise.any method.**
18. What is event bubbling and event capturing?
19. **What is the event delegation concept in javascript?**
20. **Difference between localStorage, sessionStorage and cookies.**
21. What is the immutability concept in javascript?
22. Explain the difference between map and set.
23. What is the code splitting concept in javascript.
24. **Explain memorization concepts in javascript with examples.**
25. What is the strict mode in javascript?
26. **Difference between call(), bind() and apply() method with example.**
27. What is the currying concept in javascript?
28. Difference between shallow and deep copy in object.

29. Explain Promise chaining concept with example.
30. Explain Object.freeze and Object.seal method.
31. **Explain design pattern in javascript.**
32. **Explain the concept debouncing and throttling.**

Most asked question in Reactjs interview

1. What is reactjs and how is it different from other libraries/frameworks?
2. What is JSX? What is the difference between JSX and TSX in reactjs.
3. **Explain the concept of virtual DOM , Shadow DOM in reactjs.**
4. How is virtual DOM different from real DOM?
5. What are the different types of components in reactjs?
6. **Difference between class and functional component.**
7. What is the state of reactjs?
8. What are props and the difference between props and states.
9. **What is props drilling and alternative to it?**
10. **Difference between controlled and uncontrolled components.**
11. How re-render happens when states change in reactjs?
12. What happens if you try to update the state directly in reactjs?
13. How to share data among sibling components?
14. What is a styled component? Have you implemented theming in reactjs?
15. What is the react server component?
16. **Explain lifecycle methods of components.**
17. **How do functional components handle the lifecycle method of class components?**
18. **How to handle side effects in class and functional components?**
19. What are react hooks? List out hooks you have used till date?
20. Difference between useState and useRef hooks in form handling.
21. **What are the differences between useState, useContext and useReducer?**

22. What are the rules of hooks in react?
23. **What are custom hooks? Have you used it in a project, provide an example?**
24. **What is a Higher order Component? Explain with examples.**
25. **What is the react reconciliation process?**
26. How do react compares virtual DOM with real DOM?
27. What are context API hooks? Explain with an example?
28. **What is the difference between useCallback and useMemo hooks.**
29. **Difference between context API and redux.**
30. What is the redux toolkit? How does it simplify redux?
31. What are middleware in redux?
32. How would you implement a global state without redux?
33. **What is a synthetic event in react?**
34. What is the difference between a synthetic event and a real DOM event?
35. How to pass arguments to the event handler in react?
36. How do you handle dynamic routing in react?
37. How do you manage form validation in react? Explain with examples.
38. How would you implement lazy loading with routes?
39. How do you optimize forms with hundreds of input fields in react?
40. How would I implement a multi-step form with conditional rendering between steps?
41. **What are the keys to react? Why are they important in lists?**
42. **What is react.memo and how does it work?**
43. What is the react pure component? explain with roles and use cases.
44. **What is the concept called virtualization in react?**
45. How do you optimize rendering of large lists in react?
46. How to avoid unnecessary re-rendering in react?
47. **What is the purpose of error boundaries in react?**
48. How do you handle errors in asynchronous operations in react?
49. How do you fetch data in react? explain with examples

- 50. What is Axios and how is it different from fetch API?**
51. How do you securely store API keys in react?
52. How would you implement retries logic for failed API requests in react?
53. What are fragments in react? Why are they used?
54. What is the significance of React.StrictMode?
55. How do you identify and fix the memory leaks in react?
56. How do you build react applications for production?
- 57. What is the major difference between webpack and babel?**
58. How do you configure webpack for react projects?
- 59. How do you optimize a react application for performance?**
60. Explain the purpose of .env file in react?

Most asked question in NodeJS/ Express JS and MongoDB interview

- 1. Explain the event loop in Node js**
2. Explain the internal architecture of Node Js
3. What is npm and list out the uses of it.
- 4. What are the different phases of event loop**
- 5. Explain the execution flow of Promise, async/await, setTimeout, setImmediate, process.nextTick()**
6. What is the purpose of package.json file in nodejs
7. What is http module and create a server using it
8. How do you create secure server setup using express and https
9. How will u update any npm existing packages
10. List out some of the global objects in node js
11. Explain stream and buffer and their difference
- 12. What is microtask and macrotask in node js**
13. Explain piping the data between streams

14. What are the features of express js
15. What happens, if we pass parameter to next function inside middleware
16. How we can jump from one middleware to another skipping some.
- 17. What is difference between PATCH and PUT API**
- 18. Explain the middleware concept in Node js**
19. How to create optional path API endpoints in express js
20. Explain error handling concept in Express js
- 21. What JWT and explain the structure of JWT token**
- 22. How can we modify the signature of JWT?**
- 23. What is the authentication and authorization concept ?**
24. How do you implement role based and permission based access control in express js
25. Explain in detail best practice of API
26. Explain input validation and data sanitization concept in express js
27. Explain the concept rate limiting in express js
28. Explain the concept of logging and monitoring in node js application
29. What is difference between stateless and stateful API
30. How can we achieve state management in node js application
- 31. How do you scale node js application using cluster module**
- 32. What is a worker thread ? how does it handle CPU-intensive tasks**
- 33. Explain the working with concurrency and parallelism for async task**
- 34. How can you optimise database queries**
35. What is the redis database? Explain the uses of it
- 36. How we can increase the performance of node js application**
37. What is the profiling concept in nodejs
- 38. What is connection pooling concept for database**
39. Write a middleware for role-based access control (RBAC) in Node.js
40. Demonstrate the integration of OAuth2.0 for third-party authentication in a Node.js app.

41. Implement a refresh token mechanism for JWT authentication in Node.js.
- 42. What is role of indexes in performance optimization**
- 43. What is the aggregation pipeline in mongoDB? Explain in detail**
44. What are transactions? How can we achieve it in mongoDB
45. How transaction helps in rollback in case of failure
46. What is difference between hashing, encryption and encoding concept
47. How do you identifies slow queries
48. What is replica sets and sharding mechanism in mongodb
49. How do you configure failover and disaster recovery in mongodb
50. What is load balancer and how to implement in nodejs
51. How to avoid schema pitfalls
52. Measure care taken to write efficient queries
53. How to implement health checks in mongodb
- 54. What are the solid principles for software development?**
55. What is kafka? Explain the use cases of kafka
56. How can we minimize middleware overhead

Most asked questions in the Managerial round as Tech lead/ Sr. Software Engineer

- 1. How do you ensure your team delivers high-quality code within deadlines?**
2. Describe a situation where you had to mentor a junior developer. How did you handle it?
- 3. Can you explain the SOLID principles and how you apply them in your designs?**
- 4. Explain the design pattern in javascript**
5. How do you decide which tech stack is suitable for a specific project?

- 6. what is difference between development, FT, UAT , pre-prod and production environment**
7. How do you ensure 100% code coverage in your projects?
- 8. Can you explain the difference between unit testing, integration testing, and functional testing?**
9. How do you secure sensitive data in a MERN stack application?
10. Describe your experience working in an Agile/Scrum environment
- 11. Explain the meetings like sprint planning, daily standup, backlog, sprint review, sprint retrospective**
- 12. How do you document your code and project architecture for your team?**
13. How do you onboard new developers to your team and projects?
- 14. What tools do you use to track project progress and manage tasks?**
15. What are the key considerations when designing RESTful APIs?
16. What is your approach to deploying and scaling a MERN stack application?
- 17. How would you handle a situation where the application is experiencing a memory leak?**
18. A feature you implemented in React is causing performance issues. How would you debug it?
19. What steps would you take if your database suddenly becomes a bottleneck in your application?
20. If your team is stuck on a critical bug, how would you guide them to a solution?
- 21. What is your approach to conducting code reviews?**
22. How do you optimize MongoDB queries for better performance?
23. How do you manage error handling in an Express.js application?
24. Explain how you would secure an Express.js API.
25. How do you optimize React applications for performance?
26. How do you secure a Node.js application?
- 27. How would you handle database sharding and replication in MongoDB?**