Sheryians Coding School

100 Essential Interview Questions for MERN Developers

Partition Gift for Sheryians Cohort 1.0

Prepared by Sheryians | Cohort 1.0

This is the parting gift for all students of Cohort 1.0. If you want to learn the fundamentals behind these 100 interview questions, join Sheryians latest Cohort.

MongoDB (20 Questions)

1. Explain the aggregation pipeline and its stages.
2. How does indexing work in MongoDB, and what are compound indexes?
3. What's the difference between find(), findOne(), and aggregate()?
4. How do you optimize MongoDB queries for performance?
5. What is schema design and how do you model relationships in MongoDB (1:1, 1:N, N:N)?
6. Explain data validation and how to enforce it in MongoDB.
7. What are transactions in MongoDB and when would you use them?
8. What is a capped collection?
9. How does MongoDB handle concurrency?
10. Explain change streams in MongoDB.
11. What's the difference between populate() and $lookup?
12. How do you perform pagination in MongoDB efficiently?
13. What is the use of the $facet stage?
14. How does MongoDB handle horizontal scaling?
15. Difference between embedded documents vs. referenced documents.
16. How do you back up and restore a MongoDB database?
17. How can you enforce uniqueness on a field?
18. What are MongoDB Atlas Triggers?
19. What is TTL indexing and when would you use it?
20. How do you monitor and profile MongoDB queries?

Express.js (20 Questions)

1. Explain middleware and the types of middleware in Express.
2. How does request and response flow in Express?
3. What is the use of next() in Express?
4. How do you implement global error handling?
5. How do you structure a scalable Express application?
6. How do you secure an Express API (rate limiting, headers)?
7. Explain route parameter vs. query parameter.
8. What’s the role of CORS in Express and how to configure it?
9. How do you handle file uploads?
10. How would you implement logging in Express?
11. What is the use of express-validator?
12. How do you prevent SQL/NoSQL injection in Express?
13. What is Helmet and how does it help with security?
14. How do you handle role-based authorization in Express?
15. What are virtual routes and when are they useful?
16. Difference between synchronous and asynchronous middleware.
17. How do you optimize performance in Express apps?
18. What is a proxy in Express and how to set it?
19. What is the use of app.locals and res.locals?
20. How can you implement request tracing in Express?

React.js (30 Questions)

1. Explain the React reconciliation algorithm.
2. What are controlled vs. uncontrolled components?
3. How does React handle context and why use it?
4. What are the common performance optimization techniques in React?
5. How does memoization work in React (React.memo, useMemo)?
6. How does React handle batching of state updates?
7. What are custom hooks and when would you use them?
8. Explain lazy loading and code splitting in React.
9. What's the difference between useEffect and useLayoutEffect?
10. How does React's concurrent mode work?
11. What is the virtual DOM and how does React use it?
12. How do you implement SSR (Server-Side Rendering) with React?
13. Difference between lifting state and prop drilling.
14. What is suspense in React?
15. How do you manage state in large applications (Redux, Context, Zustand)?
16. How does hydration work in React with SSR?
17. How do you secure a React frontend?
18. Explain error boundaries in React.
19. How does React Router handle dynamic routes?
20. How do you handle access control (auth-guard) in React?
21. How do you use useReducer and what is it best for?
22. How do you handle WebSocket or real-time data in React?
23. What is the role of key in React lists?
24. Explain compound components pattern.
25. How does React DevTools work under the hood?
26. What is the StrictMode in React and why use it?
27. How would you implement theme switching in React?
28. How do you test React components (Jest, RTL)?
29. How do you implement HOC (Higher-Order Components)?
30. What is the role of forwardRef in React?

Node.js (20 Questions)

1. What is the event loop in Node.js and how does it work?
2. Explain the difference between process.nextTick() and setImmediate().
3. How does Node.js handle asynchronous I/O?
4. What are streams and how do you use them?
5. How does clustering work in Node.js?
6. Explain how you'd implement a caching layer in Node.
7. How do you handle large file uploads efficiently?
8. What are child processes and how are they used?
9. What is the role of buffers in Node.js?
10. How do you monitor and debug Node.js performance?
11. What are worker threads and when would you use them?
12. Explain how you'd secure a Node.js application.
13. How do you manage environment variables securely?
14. What's the difference between CommonJS and ES modules?
15. How do you handle rate limiting in Node.js?
16. How do you implement JWT authentication in Node.js?
17. How does dependency injection work in Node?
18. What is the role of fs/promises in Node?
19. How can you prevent memory leaks?
20. How would you implement logging and metrics?

System Design / DevOps (10 Questions)

1. How would you design a scalable MERN-based e-commerce platform?
2. How do you deploy a MERN app on cloud platforms (e.g., Vercel, Heroku, AWS)?
3. What is CI/CD and how would you implement it for a MERN app?
4. How do you manage environment configurations for different environments?
5. What’s your approach to logging and monitoring in production?
6. How would you implement role-based access control (RBAC)?
7. How do you ensure your REST APIs are RESTful?
8. What is the difference between monolithic and microservice architectures?
9. How would you handle versioning in your APIs?
10. How do you handle scaling a database with growing user base?