Technical Interview Questions

Senior Software Engineer Position Assessment

Position: Senior Software Engineer

Experience Level: 8+ years

Interview Focus: Technical competency and role alignment

FOUNDATIONAL MASTERY ASSESSMENT

BASIC

1.

Explain the difference between `let`, `const`, and `var` in JavaScript.

Assesses fundamental JavaScript knowledge.

Follow up: Can you provide examples where using `const` might be problematic?

BASIC

2.

Describe the event loop in JavaScript and its importance in asynchronous programming.

Tests understanding of JavaScript's asynchronous nature.

Follow up: How does the event loop handle promises and async/await?

BASIC

3.

What are the different HTTP methods (GET, POST, PUT, DELETE) and when would you use each?

Evaluates understanding of RESTful API principles.

Follow up: Explain idempotency in the context of HTTP methods.

BASIC

4.

Explain the concept of normalization in databases and its benefits.

Assesses database design knowledge.

Follow up: What are the different normal forms (e.g., 1NF, 2NF, 3NF)?

BASIC

5.

Describe your experience with Git and common Git commands (e.g., `git clone`, `git add`, `git commit`, `git push`, `git branch`).

Evaluates version control proficiency.

Follow up: How do you resolve merge conflicts?

APPLIED TECHNICAL COMPETENCY

INTERMEDIATE

6.

Explain your experience with React.js, including state management (e.g., Redux, Context API) and component lifecycle.

Focuses on practical React.js skills.

Follow up: How would you optimize a large React application for performance?

INTERMEDIATE

7.

Describe your experience building and deploying RESTful APIs using Node.js and Express.js.

Assesses backend development skills.

Follow up: How do you handle error handling and logging in your APIs?

INTERMEDIATE

8.

How would you design a database schema for a social media application (users, posts, comments, likes)?

Evaluates database design skills.

Follow up: What database technology would you choose and why?

INTERMEDIATE

9.

Explain your experience with AWS services (EC2, S3, Lambda, RDS). Describe a scenario where you used these services.

Focuses on cloud platform experience.

Follow up: How did you ensure security and cost optimization in your AWS deployments?

INTERMEDIATE

10.

Describe your experience with CI/CD pipelines and the tools you've used (e.g., Jenkins, GitHub Actions).

Assesses DevOps experience.

Follow up: How do you ensure code quality and prevent regressions in your CI/CD pipeline?

INTERMEDIATE

11.

Describe a time you had to debug a complex issue in a production environment. What was your approach?

Evaluates problem-solving and debugging skills.

Follow up: What tools and techniques did you use?

INTERMEDIATE

12.

Explain your understanding of different software design patterns (e.g., MVC, Singleton, Factory). Provide examples.

Assesses software architecture knowledge.

Follow up: When would you choose one pattern over another?

EXPERIENCE-DEPTH VALIDATION

INTERMEDIATE

13.

Describe your experience leading the development of a microservices architecture. What were the challenges and how did you overcome them?

Explores experience with microservices.

Follow up: How did you ensure consistency and communication between services?

INTERMEDIATE

14.

Explain your role in designing and implementing the React-based dashboard that reduced customer support tickets by 40%. What metrics did you track?

Validates the impact of past work.

Follow up: How did you measure the success of this project?

INTERMEDIATE

15.

Describe your experience mentoring junior developers. What strategies did you use to help them grow?

Assesses mentoring capabilities.

Follow up: Provide a specific example of a mentee you helped.

INTERMEDIATE

16.

Discuss your experience with the migration of a legacy system to cloud infrastructure. What were the key considerations?

Evaluates experience with system migration.

Follow up: What challenges did you encounter during the migration process?

INTERMEDIATE

17.

Explain how you optimized database queries to improve application performance by 50% at StartupXYZ. What techniques did you employ?

Explores database optimization skills.

Follow up: How did you measure the performance improvement?

INTERMEDIATE

18.

Describe your contributions to the open-source projects you've worked on. What motivated you to contribute?

Assesses collaborative spirit and open-source contributions.

Follow up: What challenges did you face in contributing to open source?

ADVANCED ARCHITECTURAL THINKING

ADVANCED

19.

Design a scalable architecture for a real-time chat application using WebSockets. Consider factors like scalability, reliability, and security.

Tests advanced architectural design skills.

Follow up: How would you handle message delivery guarantees and prevent message loss?

ADVANCED

20.

Discuss your experience with Docker and Kubernetes. How have you used them to improve the deployment and management of your applications?

Explores containerization and orchestration experience.

Follow up: Describe your approach to managing container images and deployments.

ADVANCED

21.

How would you design a system to handle a sudden surge in traffic to your application? Discuss strategies for handling peak loads.

Evaluates scalability and resilience understanding.

Follow up: What are some common load balancing techniques?

ADVANCED

22.

Explain your understanding of GraphQL and its advantages over REST. When would you choose GraphQL over REST?

Assesses knowledge of alternative API architectures.

Follow up: Discuss potential challenges and considerations when using GraphQL.

ADVANCED

23.

Describe your approach to designing and implementing a system for handling asynchronous tasks and background processes.

Tests understanding of asynchronous task management.

Follow up: What technologies or patterns would you use?

EXPERT-LEVEL ROLE ALIGNMENT

ADVANCED

24.

Based on your experience, what are some key considerations for building a highly scalable and maintainable web application for a company with 10M+ users?

Assesses high-level architectural thinking for large-scale systems.

Follow up: How would you approach monitoring and alerting for such a system?

ADVANCED

25.

Describe a situation where you had to make a difficult technical decision with significant impact. How did you approach the decision-making process?

Evaluates decision-making skills and technical judgment.

Follow up: What were the outcomes and what did you learn from the experience?