

99 Full Stack Developer interview questions to hire engineers

Questions

1. Can you explain what the terms 'front-end' and 'back-end' mean in web development, like I'm five?
2. What is HTML, and what's its main job in building a website?
3. What is CSS, and how does it make a website look pretty?
4. What is JavaScript, and what does it do to make a website interactive?
5. What does the DOM stand for, and why is it important?
6. Have you used any version control systems before? If so, how has it helped you?
7. What's the difference between '==' and '===' in JavaScript?
8. What are some common data structures, and why might you choose one over another?
9. Can you describe what an API is in simple terms?
10. What is a database, and why do we need them for web applications?
11. What are HTTP methods? Describe a few of them.
12. What is responsive design, and why is it important?
13. How do you handle errors in your code? Can you give an example?
14. What is a framework? What is a library? Can you give an example of each?
15. What is the difference between authentication and authorization?
16. What are cookies and local storage, and when would you use each?
17. Have you ever debugged code? Tell me about your approach.
18. What does it mean to 'deploy' a website?
19. What are some basic security practices you should keep in mind when developing a web application?
20. What are some of the advantages and disadvantages of using a single-page application (SPA) framework like React or Angular?
21. How do you stay up-to-date with the latest trends and technologies in web development?
22. Describe a time you had to learn a new technology quickly. What was your approach?
23. Let's say a website is loading slowly. What steps would you take to troubleshoot the problem?
24. How do you handle cross-browser compatibility issues in your front-end code?
25. Describe your experience with different architectural patterns like MVC, MVVM, or Flux.
26. How do you approach debugging a complex issue that spans both the front-end and back-end?
27. Explain the difference between authentication and authorization, and how you would implement them in a full-stack application.
28. What are some strategies you use to optimize the performance of a web application, considering both front-end and back-end aspects?
29. Describe a time when you had to refactor a significant portion of a codebase. What challenges did you face, and how did you overcome them?
30. Explain your approach to writing unit and integration tests for both front-end and back-end components.
31. How do you handle security vulnerabilities like XSS and SQL injection in a full-stack application?
32. Describe your experience with different state management libraries/patterns in front-end development (e.g., Redux, Context API).
33. How do you manage and deploy database migrations in a full-stack project?
34. What are the advantages and disadvantages of using serverless functions in a full-stack application?
35. Explain your experience with containerization technologies like Docker and orchestration tools like Kubernetes.
36. How do you approach designing a RESTful API, considering best practices for versioning, error handling, and documentation?
37. Describe your experience with different caching strategies, both on the client-side and server-side.
38. How do you handle user input validation on both the front-end and back-end to ensure data integrity?
39. Explain your understanding of CI/CD pipelines and how they automate the deployment process.
40. How do you monitor and log errors in a production full-stack application?
41. Describe a time when you had to make a trade-off between performance and scalability in a full-stack project.
42. How do you stay up-to-date with the latest trends and technologies in full-stack development?
43. Explain how you would implement real-time functionality (e.g., chat application) in a full-stack application.
44. Explain the concept of RESTful API design and how you would implement authentication and authorization in a RESTful API.
45. How do you handle cross-browser compatibility issues, and what tools do you use for testing across different browsers?
46. Describe your experience with different database systems (SQL and NoSQL) and when you would choose one over the other.
47. What are the advantages and disadvantages of server-side rendering (SSR) versus client-side rendering (CSR), and when would you use each approach?
48. How do you optimize the performance of a web application, focusing on both front-end and back-end aspects?
49. Explain the concept of continuous integration and continuous deployment (CI/CD) and how you would set up a CI/CD pipeline for a full-stack application.
50. How do you handle state management in a complex front-end application, and what are some common state management libraries or patterns you've used?
51. Describe your experience with different testing frameworks and methodologies (unit testing, integration testing, end-to-end testing) for both front-end and back-end code.
52. Explain the concept of microservices architecture and how you would design and implement a full-stack application using microservices.
53. How do you handle security vulnerabilities in a full-stack application, and what are some common security best practices you follow?
54. Describe your experience with containerization technologies like Docker and orchestration tools like Kubernetes, and how you would use them to deploy a full-stack application.
55. Explain the concept of caching and how you would implement caching strategies on both the front-end and back-end to improve performance.
56. How do you handle error logging and monitoring in a full-stack application, and what tools do you use for debugging and troubleshooting?
57. Describe your experience with different front-end frameworks (React, Angular, Vue.js) and their strengths and weaknesses.
58. Explain the concept of responsive web design and how you would ensure that a web application looks good on different devices and screen sizes.
59. How do you handle data validation on both the front-end and back-end to ensure data integrity?
60. Describe your experience with different back-end frameworks (Node.js, Python/Django, Ruby on Rails) and their strengths and weaknesses.
61. Explain the concept of asynchronous programming and how you would handle asynchronous operations in both front-end and back-end code.
62. How do you handle version control and collaboration in a full-stack development team, and what are some common Git workflows you follow?
63. Describe your experience with different cloud platforms (AWS, Azure, Google Cloud) and how you would deploy a full-stack application to the cloud.
64. Explain the concept of web sockets and how you would use them to implement real-time communication in a web application.
65. How do you handle scalability in a full-stack application, and what are some common techniques for scaling both the front-end and back-end?
66. Describe your experience with different API testing tools and how you would ensure the quality and reliability of an API.
67. Explain the concept of progressive web apps (PWAs) and how you would turn a web application into a PWA.
68. How do you handle internationalization (i18n) and localization (l10n) in a full-stack application?
69. Describe your experience with different design patterns (e.g., MVC, MVVM) and how you would apply them in a full-stack application.
70. Explain the concept of GraphQL and how it compares to RESTful APIs.
71. How do you stay up-to-date with the latest trends and technologies in full-stack development?
72. Can you describe a time you had to debug a particularly challenging issue in a full-stack application? What was your approach?
73. How do you approach designing a scalable full-stack application from initial concept to deployment?
74. Describe your experience with different architectural patterns (e.g., microservices, MVC) and when you would choose one over another.
75. Explain your process for debugging complex issues that span both the front-end and back-end of an application.
76. What are your preferred methods for ensuring code quality and maintainability in a full-stack project, and how do you implement them?
77. How do you handle cross-browser compatibility issues and ensure a consistent user experience across different devices and platforms?
78. Describe your experience with different database technologies (SQL vs. NoSQL) and when you would choose one over the other, considering factors like scalability and data consistency.
79. How do you approach performance optimization in a full-stack application, identifying bottlenecks and implementing solutions on both the front-end and back-end?
80. Explain your understanding of RESTful APIs and how you design them to be efficient, secure, and easy to use.
81. What strategies do you use for securing a full-stack application against common vulnerabilities such as XSS, CSRF, and SQL injection?
82. How do you stay up-to-date with the latest trends and technologies in full-stack development, and how do you incorporate them into your work?
83. Describe a time when you had to learn a new technology or framework quickly to meet a project deadline. How did you approach it?
84. Explain your experience with cloud platforms (e.g., AWS, Azure, Google Cloud) and how you leverage them to deploy and manage full-stack applications.
85. How do you approach version control and collaboration in a full-stack development team, and what tools do you prefer to use?
86. Describe your experience with different testing methodologies (e.g., unit testing, integration testing, end-to-end testing) and how you ensure thorough test coverage in a full-stack project.
87. How do you handle state management in a complex full-stack application, especially when dealing with real-time updates and multiple users?
88. Explain your understanding of CI/CD pipelines and how you set them up to automate the build, testing, and deployment process for a full-stack application.
89. How do you approach monitoring and logging in a production full-stack application, and what tools do you use to identify and resolve issues?
90. Describe a challenging full-stack project you worked on and the lessons you learned from it.
91. How do you balance the need for rapid development with the need for long-term maintainability and scalability in a full-stack project?
92. Explain your experience with different front-end frameworks (e.g., React, Angular, Vue.js) and when you would choose one over the other based on project requirements.
93. How do you handle asynchronous operations and concurrency in both the front-end and back-end of a full-stack application?
94. Describe your experience with containerization technologies (e.g., Docker, Kubernetes) and how you use them to deploy and manage full-stack applications.
95. How do you approach code reviews and provide constructive feedback to other developers in a full-stack team?
96. Explain your understanding of different caching strategies and how you implement them to improve the performance of a full-stack application.
97. How do you approach data modeling and schema design in a full-stack application, considering factors like data integrity and query performance?
98. Describe your experience with different message queue systems (e.g., RabbitMQ, Kafka) and how you use them to build asynchronous and scalable full-stack applications.