



INTERVIEW PREP

Front-end engineer

WHAT DOES A FRONT-END ENGINEER (FEE) DO AT AMAZON?

Well, we build, create, and deliver delightful user experiences to our customers at a global scale. We own the front-end software development life cycle, design solutions, and work on coding, testing, implementing, maintaining and iterating solutions. With a strong understanding and applicable knowledge of front-end engineering fundamentals, things like inventing new types of customer experiences and optimizing existing ones to support hundreds of millions of users. We support a diverse customer base across a wide range of devices and browsers.

Want to become an Amazon front-end engineer? Let's walk through some helpful tips for the interview process.



Front-end engineer tips

Be familiar with prominent programming languages, including the syntax of the language. Ultimately, pick the one you're most comfortable with and stick with it. Know how to use languages, libraries, and rendering technologies (e.g., *web*: JavaScript, Node.js, templating languages, HTML, CSS; *mobile*: Java, Objective C, Swift, C#, webviews, view frameworks).

Showcase your knowledge of front-end system design. i.e. reusable components, separation of concerns (view models from business logic), application state management, and basic n-tier computing concepts (front-end, middle-tier, back-end).

Understand the inner workings of common data structures. Be able to compare and contrast their usage in various applications (e.g., retrieving JSON and using it to populate and power a user interface).

Research application performance concepts and technology. i.e resource caching (images, fonts), content delivery, asynchronous programming, and real user metrics.

Write syntactically correct code—no pseudo code. Ensure it is scalable, robust, and well-tested.

Use object-oriented design or functional programming (logical and maintainable code) best practices to build lasting, scalable software.

Be familiar with the devices and/or browsers which run your software. Topics such as system availability, efficient resource usage (CPU, battery, screen size), and performance implications.

Be familiar with device and/or browser topics such as security, native API methods, local storage, and compatibility.

FEE technical tips

Be familiar and ready to solve in-depth technical questions on concepts like front-end application design, data structures, and algorithms . This will likely include qualifying requirements, checking edge cases, and white boarding your solutions with our engineers. In-person interviews tend to be more in-depth than the types of questions asked during your phone interview.

Be prepared to discuss technologies listed on your resume. i.e. if you list JavaScript or Ruby as technical competencies, expect technical questions about your experiences with these technologies. It is helpful to review the job description before your interview to align your qualifications against the job's specific requirements and responsibilities. When listing frameworks, be prepared to speak about these in depth and compare to other frameworks.

Feel free to brush up on problem solving and core Computer Science fundamentals. "Cracking the coding interview" is a good read.

White boarding

Be prepared to white board. Practice writing code, front-end system design, and creating rough UI wireframes. Consider logical, maintainable, and scalable code or design before you begin drafting.

Interact with your interviewer. You will be asked several questions related to design. Engage with your interviewer with necessary questions to complete the exercise.

Dig for clarification. Your interviewer will not try to trick you. Questions may be intentionally vague to push your innovation.

Begin drawing a diagram once you've done enough digging to begin white boarding your system design solution. Start with shapes to represent different software components and data sources, and then arrows connecting them to show web services, APIs, and interactions between components.

Know how your solution solves the problem. If you suggest technology to help solve, understand how that technology works.

Think out loud as you write out your code or system design.

Show us your ability to solve problems.

Application performance is a critical component of front-end software design. Consider how to store and retrieve data, client-side vs server-side processing, browser/device rendering efficiency, and data flows. Keep this in mind when diagramming and designing your software systems.

Operational performance is a critical component of front-end software design. How will you ensure this component or application is working at an acceptable level of performance? If a problem occurs, what will be involved to trouble shoot and resolve quickly? What are the possible points of failure and how can they be made more robust against failure?

Keep the customer front of mind. Who is the customer, and what problem are you solving for them?

Write a list of requirements on the board, and keep asking questions. This should be the first thing you write out.

LEARN MORE

Dive into our [Leadership Principles](#)

Discover [Amazon's peculiar culture](#)

Explore [Interviewing at Amazon](#) for FAQs, prep guides, and more

QUESTIONS? REACH OUT TO YOUR RECRUITING POC

AMAZON IS AN EQUAL OPPORTUNITY EMPLOYER

