



Interview Guide | Engineering

Welcome

Thank you for your continued interest in the Chan Zuckerberg Initiative. This document provides an overview of your next day of interviews. We appreciate the time you've invested in our interview process and are excited to have you meet the team!

Please review our [Candidate Resources](#) to learn more about CZI, our culture, and [how to prepare for a Zoom interview](#). If you require additional accommodations, have questions, or need support on the day of your interviews, please contact your Recruiter or a member of the Hiring Experience team. Someone from the team will ensure you are taken care of and that your experience is smooth.



1:1 Interviews

You will meet with 4-5 of our interviewers – some potential teammates and some cross-functional team members – for 60 minutes each. You will mostly meet with one interviewer at a time but there may on occasion be an additional team member observing the interview, we use interviews to help calibrate our interviewing team. At some point during your visit, your Recruiter will meet with you to touch base.



AMA

There will also be a 30 minute “ask me anything” interview, where a member from the team will meet with you to answer any questions you might have, and have the opportunity to learn about us and our work.



Lunch & Breaks

Breaks will be scheduled after 2-3 interviews and if you're interviewing with us during the hours of 11:30 am through 1 pm we will schedule a lunch break. This will be your opportunity to take a mental break, stretch, use the bathroom, grab a snack, etc. Of course, if you need anything at any point during your interview, please let your interviewer know. Your comfort is important to us throughout your interview experience.



Questions

During these meetings, our interviewers are getting a glimpse of your skills and work style. The questions they ask are not meant to quiz or stump you. We are more interested in how you think and how you'd overcome roadblocks so we want to encourage you to think out loud and share your thought process along the way. If something seems unclear, feel free to ask clarifying questions.



You are likely to encounter the following interview types:

1. Code Pair (two rounds)
2. Systems Design
3. Career & Competencies

Code Pair

The Code Pair style of interviews is designed to evaluate your coding ability. To ensure you can demonstrate your problem solving skills and technical fluency in depth across a range of coding challenges, there will be two rounds of this interview type.

During this interview, you will be expected to write (and run) code in the language of your choice. The interviewer will present you with a problem and allow you time to ask clarifying questions. You will then solve the problem by writing code on our pair programming tool via HackerRank. Please be prepared to share your screen upon the interviewer's request. You will be asked to discuss your approach and test your code against various samples of test data.

Please familiarize yourself with the HackerRank [development environment](#) before your interview and review the [interview preparation checklist](#). In addition to these resources, you'll also find a variety of helpful articles to reference before your interviews.

For Frontend specific interviews, please check out this [optional pre work](#) to get a head start on the internal question and familiarize yourself with the HackerRank environment.

System Design

In this interview, you will be discussing how you might design a software system to accomplish a certain purpose. While you won't be coding, you will be expected to articulate tradeoffs and assess how and why you might implement a certain design. These questions don't have a single correct answer or a definitive endpoint. Rather, they give you the chance to demonstrate how you handle complexity and make decisions factoring in a variety of requirements and stakeholders. For this question type, we'll be looking to understand:

- How you break down and clarify complex challenges
- How you evaluate what are the most important parts of a problem to focus on.
- How you identify and prepare for extreme use cases
- How you balance requirements and desires from a variety of stakeholders
- How you design for scale

While we are virtual we have come up with a few alternatives to the traditional white board interview. For this interview you can expect to use either Zoom's whiteboard feature, a Google Drawing, a pen and paper or any other medium you feel comfortable with.

Career and Competencies

Our Career and Competencies interview allows candidates to highlight their experiences and perspectives through behavioral style interviewing. We believe that teams composed of people with diverse experiences will allow us to create better solutions and relationships with the communities in which we serve and help us advance our goals under each of the initiatives.

Questions during this interview will focus on your career as well as how you handle situations. Remember to draw upon your professional and personal experiences to answer these questions. The use of real-life examples makes it easier for the interviewer to ask for more details and get a stronger signal. Don't be worried about sharing your failures as well. Learning through failures is a valuable skill that we look for in a candidate.



General Tips for Coding/Design and Beyond

Think out loud

It's not just about the end solution. What we care about is how you are thinking about a problem and the approach you are taking to get to a destination. Bounce ideas off of your interviewer. Break down the problem out loud. Share what parts of the problem you're struggling with. By communicating your thought process, you are also allowing the interviewer to correct any misunderstandings and provide high-level guidance.

Understand the problem

If you don't fully understand the question, ask! Take the time to fully understand what is being asked before jumping in.

Take a moment to consider alternative approaches

It can be helpful to consider multiple approaches and then choose one. You could choose based on time/space complexity, ease of implementation, extendability, or something else! That said, sharing how you're thinking about these trade-offs with the interviewer is helpful.

Do your research and bring your questions

Do your research on us and think about what motivated you to engage with CZI. This is your opportunity to interview us as well. We value curiosity and want to answer your questions.

Be yourself

Tell us what you're passionate about and what drives you. We want to learn about you and the perspective you bring to CZI. The best tools are built by diverse teams, so we want you to bring your authentic self to the interview.

Stay focused and positive

Don't worry if your solution is not perfect on the first try. Don't let that throw you off your game during the interview. We hit roadblocks every day - it's about how you navigate them that makes the work fun.

Edge Cases

Don't forget about **edge cases**! Once you have a solution, make sure it can handle most, if not all, edge cases.

Coding specific

Your Solution

- Try and understand your high-level solution before starting to code something.
- Understand what your code does and be able to explain it to your interviewer.

Implementation

- Check your code for mistakes or bugs.
- Be open to feedback or hints from your interviewer (avoid being defensive).
- Run through basic test cases when possible. Enumerate edge case tests to show you thought about them and didn't arrive at a solution accidentally.
- Be prepared to defend your solution and point out any errors or mistakes.

We Are Not Looking For.....

Coding language minutiae

While we are looking for general fluency in the language of your choice, don't worry if you forget the exact API of a library function or can't remember the exact syntax of something. Using the internet resourcefully is an expected part of the job, we let you look things up just like you might at work.

Ability to respond to stress

We recognize that interviewing can make someone nervous and not able to perform to their full ability. We are not trying to trick you, intimidate you, or otherwise place added stress on you just to see how you respond.

The perfect solution

Our questions are designed to be complex. We don't expect you to come up with a brilliant, optimal solution in 50 minutes. Rather, we look for how you develop and communicate your approach, how you turn your approach into code, and how you iterate and improve the code you've already written. At the same time, don't rush into the first brute force solution that comes to mind; we do like to see that you can develop multiple approaches and compare them to choose the best one.