



Welcome to your prep guide for your onsite Engineering interview at H-E-B Digital! Our engineers, leaders, and recruiters have put together this guide so you will know what to expect and how to prepare.

We know the interviewing and hiring processes can feel overwhelming, but we are working to make sure that *you* don't feel overwhelmed when interviewing at H-E-B. Our goal is that all of our candidates – from entry level to leaders – have access to the same information and resources to help make this process as comfortable as it can be for you.

Interview Overview

Your interview process is sectioned into the following 50-minute interviews, some of which will vary depending on your area of expertise. If you have any questions about the scheduled interviews, please reach out to your recruiter.

[Technical Problem Review](#)

[Career & Culture](#)

[Project Retrospective](#)

[Leadership](#)

[System Design & Architecture](#)

Technical Problem Review

For this review, we'll be discussing the code you wrote as part of the take-home exercise to produce a coding sample – or review an existing project of your own that is publicly available. We'll dig into algorithms, data structures, design patterns, and the scalability of the solution.

After you have completed the Technical Problem Review, your interviewer will share the feedback from your problem review session with the Foundational Technology hiring team. Your recruiter will reach out to you within 48-hours to schedule your Onsite/Zoom interviews if your candidacy has been selected for next steps.

How to prepare:

- Make sure you've reviewed your code and can walk someone through your decisions.
- If you've used any 3rd party libraries, make sure you know general details of their underlying implementation and complexities.
- The reviewer may ask you about test cases you haven't considered, or introduce new complexities not included originally, think out loud as you work through how your code may be adjusted to handle them.
- Have your code open in your IDE so you can share your screen.
- Ask questions! The interviewer is not here to trick you, they are here to help you and are trying to get a sense of what it would be like working with you on a problem collaboratively.

Career & Culture

This interview helps us assess your capability in supporting organizational health and influencing the technical direction of your project or company. Your interviewer will ask you to showcase success stories and situations that present how you, as a leader, have navigated the complex business problems that affected the company at large while being aligned with the core values and culture of your organization.

Example questions:

- What did you do on your very best day at work?
- What are some of your career goals? How do you approach achieving them?
- Tell me about four people whose careers you've fundamentally improved?

Project Retrospective

This discussion-based interview covers project implementation. The interviewer will ask you to present a project you've previously worked on. Come prepared to discuss your project at a

high-level with the interviewer and be able to provide details to explain the successes and failures of the effort and what you learned from them. We don't expect a presentation and we discourage slide decks; however, we accept examples and diagrams that can help you get your point across.

What we look for:

- High level product / project requirements, planning, and general business understanding – can you explain ‘the why’?
- Negotiating with stakeholders and peers.
- Setting, articulating, and driving impactful metrics.
- Planning for scale, iterating, learning from failures, making adjustments and trade-offs.

Leadership

This is a fairly unstructured interview, and is generally focused on philosophy and process around your leadership experience. Be prepared with a few stories that highlight your team leadership, mentorship of others, team-building, and knowledge sharing skills. Think of examples that show both successes and failures in which you’ve navigated the complexity of being part of a team, helping more junior members of the team grow, your role within the team, and supporting the people on your team and within the organization.

Example questions:

- What makes a good team?
- Tell me about a time you had to get buy-in from your team to accomplish something.
- Describe the best code review you ever participated in, what was great about it?

System Design & Architecture

Depending on the role you are pursuing as a candidate, this part of the interview may include one or two interviews. If you are unclear, please reach out to the recruiter for additional information.

This interview covers the design of complex systems and the tradeoffs within your design. The scope of the question can vary widely. It’s a challenging and deep technical discussion around product ideas, usability issues, scalability, data structures, and technologies used. One question will typically draw directly on your previous experience (e.g. file system design, device drivers, ad serving systems, web caching systems). Examples include: design a hotel reservation system, design an existing system like Twitter, Facebook, Google, Uber, etc.

For this interview, there's no right or wrong answer. The interviewer will observe how you design and architect a system. We'll look to you as the expert here and ask you to drive the design, starting with defining the high-level goals and then proposing a solution.

Things to focus on during your discussion:

- Testability – How will you gain confidence that the system works?
- Usability – What does the customer experience look like for the consumer?
- Extensibility – Will it be easy to iterate or change in the future?
- Security – How can the system survive DDOS, spoofing, tampering, repudiation, privilege attacks?
- Portability – Could this system be run in different environments, OSs, hardware?
- Availability – How does the system survive failures?
- Scalability – How can the system grow over time?
- Operational characteristics – Is it easy to debug, run in production, or minimize service disruption?