

A. Behavioral Interviews - Definitive Overview and Preparation Guide

Guide to behavioral interviews for front end / web developers / software engineers, written by ex-interviewers at FAANG. Find out about evaluation criteria at big tech, efficient strategies to prepare, and top behavioral interview questions.

Behavioral interviews are one of the most overlooked interview rounds, especially for software engineer interviews. Whilst technical interviews are complex and require a lot of time to prepare for, the importance of behavioral interviews cannot be overstated.

There are however a few common challenges faced when preparing for them:

How to prepare when there are so many different types of questions that could come out?

Indeed, it's impossible to fully prepare for any behavioral question that could come out, since interviewers can ask anything they want. However, we can categorize the most common questions into a few main themes and tackle them methodologically.

What is the best way to conduct yourself to showcase "fit"? Many candidates struggle to appear likable in an interview setting and to connect with the interviewer. We will cover some tips to tackle this based on our experience.

In this article, we will cover:

Behavioral interview signals and evaluation criteria.

Recommended preparation strategy.

Best way to conduct oneself on the actual interview.

Behavioral Interview Signals and Evaluation Criteria

In big tech companies where structured behavioral interviews are conducted, interviewers are often trained to conduct behavioral interviews in a specific manner.

They are provided specific guidelines, which commonly include:

Evaluation criteria: Typically split into certain major categories such as Communication, Proactivity, etc., which they need to evaluate candidates on. Each category will typically also include specific guidance as to how to grade a candidate as good, poor, average, etc.

Level expectations: Interviewers are typically asked to provide an expected level after interviewing the candidate. To ensure consistency between interviewers, a leveling guide is also provided.

Below, we present the typical behavioral interview evaluation criteria as well as the typical level expectations for most companies. The recommended preparation strategy provides suggestions on how you might prepare based on these criteria.

The Interviewer's Perspective

Behavioral interviewers are typically required to give a rating on a few broad categories, such as Collaboration, Proactivity, etc. For each evaluation category, they are provided with criteria as to

what should constitute a good or bad rating. Ratings are commonly done on a 5-point rating scale, such as one of Very good / Good / Sufficient / Poor / Very Poor.

In addition to these, interviewers can also mark if there wasn't sufficient information to make a judgment, or if they didn't manage to ask a question that touches on a specific signal as they ran out of time.

Ratings have to be done with regard to the situation's complexity. A more difficult or complex situation handled well will typically receive a higher rating and leveling recommendation.

At the end of the interview do they reconcile all the ratings and qualitative feedback to decide:

Whether to pass or fail the candidate

If passed, what level the candidate should receive

The implication of the above is that we can typically expect behavioral interviews in big tech to have some structure, as interviewers will try to ask at least 1 question from each category to evaluate candidates.

Additionally, it is important for candidates to avoid rambling on for long periods of time on one question as interviewers may run out of time. Poor communication or missing the point of the question could also constitute insufficient signals.

Apart from that, interviewers are typically encouraged to utilize follow-up questions to dig deeper into the candidate's actual motivations and understanding behind their surface-level actions.:

Why do you think you did (insert action)?

Why did you not do (insert action)?

How would you do things differently in hindsight?

Hence, candidates should also prepare a solid understanding of each alternative mode of action's advantages, disadvantages, and reflect on their intrinsic motivations.

Typical Behavioral Interview Evaluation Criteria

Here are the common evaluation categories and criteria from our experience conducting behavioral interviews for software engineers in big tech companies:

1. Collaboration

Handling disagreements

Working as a team

Working with diverse personalities and skills

Conveying complex ideas simply

Giving constructive feedback

Active listening

2. Driving Results and Problem Solving

Impact-driven mindset

Proactivity to make progress despite obstacles or roadblocks

Influencing others to deliver against objectives

Balancing analysis with decisive action

Identifying best solutions and executing on them

Identifying the right problems

Identifying the most critical objectives

Resourcefulness and data-driven mindset

Creativity and innovation

Identifying tradeoffs and sustainable solutions

Measuring results, iterating and following through

3. Growth Mindset

Having self awareness of strengths and development areas

Actively soliciting feedback from others and creating feedback loops

Taking feedback and setbacks well, viewing them as opportunities

Takes personal accountability for failures

Reflecting, modeling best practices and applying lessons to new situations

Energized by setting and working towards challenging goals

4. Adaptability and flexibility

Performing well in uncertainty

Remaining calm and focused, serving as stable presence for others

Flexible and quick to adapt to unexpected situations

Typical Behavioral Interview Leveling Expectations

Behavioral interviews is one of the interview formats where it is possible for interviewers to assess the candidate's maturity — from Junior Software Engineer all the way to extremely senior positions depending on the stories they tell, their responses and decisions made. Raw number of years of working experience doesn't necessarily equate to specific levels as some people mature faster than others. Leveling guidance is usually made with a consideration of raw years of experience and leveling expectations. The levels mentioned below are based on Google and Facebook's levels where L3 is the equivalent of Junior Software Engineers / fresh graduates.

Understand the expected behaviors for the commonly-defined engineering levels and craft your responses ensuring they match your current level or the level you are targeting.

Junior Software Engineer (L3)

Takes direction from managers and/or senior members of the team.

Executes on well-defined tasks without too much guidance.

Primarily improves their own skills and craft.

Software Engineer (L4)

Understands project purpose and helps to achieve the team's objectives.

Splits larger projects (typically feature work) given to them into smaller tasks for timely execution.

Suitably balance between delegating tasks and performing tasks themselves.

Improves their own skills while possibly mentoring more junior members, interns, or new hires.

Collaborates cross-functionally with guidance from senior members of the team.

Senior Software Engineer (L5)

Leads development on complex tasks and projects within the team.

Identifies well thought out solutions to ambiguous projects of large scope

Breaks solution down into smaller projects and delegating some of them to team members.

Proactively seeks new ideas and directions to build the product better.

Mentors multiple junior members of the team and help them to improve by being a role model.

Works cross-functionally independently and drives complex, ambiguous discussions.

Staff Software Engineer (L6)

Understands business goals and advises managers and organization leaders to achieve the goals.

Leads or greatly influences direction of a team of engineers.

Demonstrates domain expertise and is looked up to by other engineers.

Conducts clear, long-term planning across teams/within the organization and drives consensus.

Mentors tech leads and junior members of the team.

Works on projects requiring interactions with engineers and cross-functional partners across multiple teams.

Senior Staff Software Engineer (L7) and beyond

Owns organization/company-wide business/engineering goals and delivers on them.

Influences or leads product and engineering roadmaps within the organization/company.

Leads solution development and delivery for highly complex and ambiguous areas.

Recommended Preparation Strategy

As mentioned above, based on the evaluation criteria we can typically expect behavioral interviews in big tech to have some structure, as interviewers will try to ask at least 1 question from each category to evaluate candidates.

Apart from these questions, we can also expect general but common questions such as "Tell me about yourself" or "Why join this company".

There is quite a lot of variance in terms of the questions that could be asked. While it is too time consuming to prepare for all of them, we can categorize the 80% most common questions into themes and tackle them methodically, per our following guides:

"Tell me about yourself..."

"Walk me through your resume..."

"Why join this team or company?"

"Do you have any questions for me?"

"Tell me about a time when..." (categorized into below themes)

Driving results and problem solving

Collaboration

Growth mindset

Adaptability and flexibility

Prepare a Few Good, General Stories Based on Common Themes

The general strategy to tackle the large variance of "Tell me about a time when..." questions to be asked is to prepare 3-5 general and well-thought out stories that could be used to address multiple required traits. For instance, an experience collaborating extensively with others could display your traits in communication, team work, leadership, conflict management, etc.

Whenever an interviewer asks you a question, you could run through the stories you have prepared and adapt them to answer the specific question.

Here are some tips to select and prepare good project experiences as stories:

Check against the evaluation criteria above and try to pick broad experiences which can be used to address multiple criteria. Then, write down short bullet points of how you exhibited behavior that fulfilled required traits.

Choose experiences that showcase your level expectations. For instance, if you are applying for a Staff Software Engineer position, ensure that your experiences exhibit your influence on the direction of entire teams.

Do not reuse the same stories too much in the same interview loop. Interviewers tend to discuss candidate performance together, hence overusing the same story could come off as repetitive and insufficient breadth of experience.

As mentioned above, interviewers tend to prioritize follow-up questions to understand the candidate's true psyche, motivations and understanding. For each story, prepare a solid understanding of each alternative mode of action's advantages, disadvantages, and reflect on your intrinsic motivations.

Structure Initial Answers with the STAR Framework

Having prepared a story you could use, you need to structure them well when answering specific questions. One requirement interviewers are always asked to look out for is the candidates actual ability to communicate during the interview (as opposed to what they claim), which is evaluated as follows:

Well-organized and easy to follow thought process

Good intuition on which details should be elaborated and which to keep simple

Leaves others clear on purpose and actions

Demonstrates active listening

To achieve the above, you need to structure your answers well to ensure that it is concise and gets to the point below 3 minutes. One rough guideline you could use is a framework like STAR:

Situation: Present basic details of the scenario, just enough to understand the challenge you were in. (Do not spend too much time on this)

Task: What did you need to achieve or do in the scenario? (Do not spend too much time on this)

Action: What did you do? This is where you exhibit required traits based on the approach you took to solve the challenge or situation.

Results: What was the outcome of your actions?

Here is an example of applying STAR to answering a behavioral interview question "Tell me about a time you had to manage multiple conflicting priorities and how you handled it":

Situation

In my current job as Front End Engineer in a startup, I was once in a situation where I had to deliver several important features for an e-commerce campaign at very short notice, since different teams were making feature requests at the time.

Task

As I recognized that it was not possible for me to achieve all of them with good quality and also in a timely manner. I had to find a way to deconflict their priorities.

Action

What I did was to arrange a quick meeting with all relevant product and engineering stakeholders to co-prioritize and assign the appropriate resources to support this project. I listed the features requested from every team and worked with all the relevant stakeholders to identify each feature's contribution to business objectives, while also roughly estimating the engineering effort for each one.

This helped us to deprioritize features that had very high engineering effort but little contribution to the objectives.

For projects that were not realistically possible to achieve within the timelines, we were able to negotiate for more engineering resources to delegate these feature work to.

After that, to ensure my own timelines were met, I broke down my features into smaller tasks and planned them into daily and weekly milestones, reviewing my progress regularly with the team.

Result

With this co-prioritization and planning effort, I was able to achieve all the required features by the stipulated timeline.

Important warning about your answers

A final note — in order for the interviewer to follow your thought process or understand how brilliant your solutions were, it is VERY important for them to understand the context, purpose and situation. The problem or situation should be described very clearly and it should assume NO prior understanding. Many candidates fall into the trap of assuming the interviewer can follow their stories without giving enough context, especially when problems are domain-specific like fintech or blockchain. In such situations, interviewers might end up marking you as having bad communication, or rate the category as having insufficient signals.

Practice makes perfect

After preparing your stories and having a rough sense of STAR, practice using your prepared stories and the STAR format to answer our list of 50 common behavioral interview questions. Instead of memorizing the answers, practice just answering them verbally each time. This allows you to cement your stories and familiarize with STAR.

How best to conduct yourself in a behavioral interview

Understanding the interviewer's intention

Some candidates think of behavioral interviews as a test or an exam, where you have to study or memorize to get the right points or answer. That is not what the interviewer is looking out for. To understand how interviewers themselves feel, try to picture yourself interviewing someone to join a project you are actually currently working on. That is, put yourself in the hiring manager's position.

A behavioral interview usually happens after the technical rounds are completed. Candidates that have passed the technical rounds are considered technically competent. However, as a manager, you hardly really know this stranger (the candidate). Interviews are a conversation for you to get to know this stranger as a person. It's just like a typical chat in a social setting, somewhat like a speed date, except the interviewer needs to get to know you enough to decide if you can work well.

Understandably, as an interviewer, you would generally try to sieve out:

Is this candidate going to be a poor performer? e.g. Often miss on deadlines, not accountable, not driven, gets into regular conflict with others, communicates poorly, does not take feedback well. Are they lying or exaggerating on their CV?

Is the candidate going to be a superstar, driven teammate? e.g. communicates well, resourceful and independent in solving problems, able to manage stakeholders, proactive and works hard in enacting new initiatives to drive the team forward.

Is the candidate going to be easy to manage? e.g. takes feedback well, likable, sociable and does not give you problems.

If a candidate were to recite answers back to you or appear stoic and unexcited, would you hire them into your team? Probably not. For interviewers, it's as much about seeing you as your true, natural self as it is to know if your thinking process is aligned with their goals.

One size fits most

Interviewers always mention team "fit" as a criteria for evaluating you. But there are so many different teams out there — how would you know how to conduct yourself to display fit?

One way to do so is to ask the recruiter or other interviewers in prior rounds to the behavioral round. Commonly, prior to starting the resume screening / interview process, the hiring managers would have aligned internally on the key criteria they are looking out for in the new candidate. For example, they could be looking for someone who is primarily more proactive as they want more independent drivers in the team. As such, interviewers will tend to look out for those specific traits and ask questions to sieve out those traits.

However, some fail-safe, general tips to conduct yourself

Friendly and positive: Friendly and positive people tend to fit well in any team.

Proactive and driven: Go-getters are correlated with higher performance.

Enthusiastic: Demonstrate clear interest about the role, team, and the company.

B. The Most Important Behavioral Interview Questions to Prepare for

Top behavioral interview questions to practice for front end / web developers / software engineers. Includes additional questions for experienced level engineers.

In this article, we will cover the top questions (ordered from highest frequency) that candidates should practice with before their interviews.

As alluded to in the behavioral interview preparation overview, most behavioral interview questions can be categorized into 8 main themes for efficient preparation. We recommend candidates to prepare using this strategy and then refer to the list below for practice.

Top 16 most common questions

Tell me about yourself.

Walk me through your resume.

Why have you applied for this role and team?

Do you have any questions for me? (end of interview)

Can you tell me about a time you improved upon your work upon receiving criticism or feedback?

Can you describe a past project where you had to work with a difficult stakeholder or teammate and how you dealt with it?

Can you describe a time you creatively solved an engineering problem or achieved meaningful metric improvement?

Can you describe a time when you had to make an important engineering decision and how you decided between tradeoffs?

Can you give an example of a time when you had to work on a team project and had to compromise with others to reach a solution?

What do you think is your greatest area for improvement?

Describe a time you failed and had to alter course and adopt a new approach. How did you know change was necessary?

Can you describe a time when you had to work with a tight timeline but was not given a clear direction on how to proceed?

Can you describe a time when you had to adapt to changing priorities or requirements?

Tell me about a time you learnt a new skill outside of work and applied it to your work.

Tell me about a time you learnt a new skill set after observing others leveraging it to great success

Can you describe a time when you had to effectively communicate technical information to a non-technical audience and how you approached it?

Additional Questions for Leaders (Tech leads or Engineering Managers)

Team Management and Leadership

How do you approach the process of setting and achieving goals for your team?

Can you tell me about a time when you had to work with a tight budget or limited resources to achieve an objective?

How do you approach collaboration and communication with your team and stakeholders?
Describe a time when you had to handle an underperforming team member.
Can you describe a time when you had to adjust your management style to fit the needs of your team?

Cross Functional Collaboration

How do you approach the process of gathering and incorporating user or client feedback into your work?

Can you give an example of a time when you had to balance business needs with engineering complexity?

Additional Questions for Practice

We've grouped the most common behavioral questions into 8 major themes. Check out additional practice questions while studying them:

"Tell me about yourself..."

"Walk me through your resume..."

"Why join this team or company?"

"Do you have any questions for me?"

"Tell me about a time when..." (categorized into below themes)

Driving results and problem solving

Collaboration

Growth mindset

Adaptability and flexibility

C. Behavioral Interview Guidebook

Tips for Making a Strong First Impression in your Self Introduction

How to frame your interview self introduction effectively in different interview contexts, for front end / web developers / software engineers. Refer to examples of good self introductions.

The self introduction is one of the things you can always expect to encounter in an interview, regardless of interview format. Hence the return on investment of preparing for it is rather high. Most of the time, your self introduction should be short and sweet, delivering a good impression in the fewest amount of words.

In this article you will find:

Recommended preparation for different interview settings

The 1-min pitch formula

Portfolio or case study showcase

Examples of good self introductions

Tips and stories from hiring managers

Recommended Preparation

In essence, you should always prepare a self introduction that is under 1 min in almost all cases.

Prior to the interview, it is best to understand more about the evaluation criteria for the round from a recruiter or hiring manager — specifically if you will be evaluated on behavioral/fit aspects or purely on technical criteria. In big tech companies, coding rounds may totally exclude behavioral related criteria, hence your self introduction will most likely just serve as a "warm-up" to the main interview.

However, in behavioral interviews or technical interviews in a startup or a company which prioritizes practical experience over LeetCode-style interviews, you may want to extend your self introduction to a quick portfolio showcase, or allude to the fact that you have a portfolio of your projects that they could view after the interview or allocate around 5 min at the end of the interview to run through.

However, do note that in whichever case, if you are doing coding interviews with fixed format in the round, you should keep your self introduction short to ensure that you have more time on the actual coding test. Any extra time you take on your self introduction will be taken away from the time you have to answer technical questions.

Interview Format	Company Type	Duration	Content
Coding or technical interviews	Big tech with fixed evaluation criteria	< 1 min	1 min pitch
Others which may emphasize less on standard LeetCode-style coding tests		~1 min	1 min pitch + allude to portfolio showcase
Behavioral or culture fit interviews	Any	~1 min	1 min pitch + allude to portfolio showcase
To tackle all the possible cases, you should prepare a 1-min pitch and a project portfolio (recommended but optional).			

The 1-min Pitch

The Big Picture

Most of the time, pitches will cover these 4 items:

Breadth and depth of job experience

Top achievements

If less than 5 years of working experience, educational background

Motivation for applying for role

When developing your pitch, keep the objective in mind — you want to leave a good first impression as a potential ideal candidate for the role. To do this, you should come off as confident, likable, on top of demonstrating key requirements of the role. Think about what you would look out for if you were to hire a candidate for this role?

Do not attempt to go too deep — focus on covering the main points and enunciating them confidently and smoothly. As software engineers, we prove our mettle through projects and code. Talking too much too early may come off as over-eager and words alone don't hold much value.

In essence, to hiring managers, the self introduction serves to answer these questions about the candidate for them:

Is this candidate passionate about front end / software engineering?

Can they describe things in a clear and concise way?

What kind of person are they? Confident or arrogant?

Would I want to work with this person? Do they seem coachable?

Example pitch points for front end / web developers

On top of role-specific requirements such as team-specific frameworks and technologies, front end hiring managers typically focus on the following 4 criteria:

CriteriaExample

Understanding of front end fundamentals: HTML, CSS, JavaScript and related areas "I have been building front end applications for a few years now and also contributed to popular open source projects like Lodash and jQuery."

"I was a Teaching Assistant for my college's web development course and mentored students working on projects which involved building full stack web applications."

Breadth and depth of front end technologies the candidate knows "I used React, Tailwind, Next.js, Prisma and MySQL to build a Twitter clone as part of my software engineering team project."

Initiative to keep up with modern front end technologies "I learnt Astro and rebuilt my personal blog using that because Astro is great for building content-driven websites."

Relevant front end projects that the candidate has worked on, complexity of those projects

"In my spare time I built a crypto price tracking app in React to learn how to build data visualization-heavy client apps and also solve a personal pain point of tracking my portfolio."

Refer to below for examples of good self introductions.

Portfolio showcase

If you were to extend into a showcase of your portfolio, you would want to ensure that your portfolio is minimal and easy to run through during the interview

The description for each project should encompass:

Purpose of project: What it does for users

Frameworks/technologies used

Technical complexity involved

Refer to this article for a more detailed understanding on how to conduct a portfolio showcase (coming soon).

Examples of Good Self Introductions

For Experienced Engineers

I worked at Airbnb as a Front End Engineer for 6 years and was the lead developer on the project to redesign Airbnb's booking experience. As part of my work, I contributed to Airbnb's

main design system by building several UI components like the carousel, ratings widget, and refactoring existing components. I also had experience contributing to Airbnb's open source front end projects like Enzyme and react-dates.

In my spare time, I enjoy learning new front end technologies and catching up with the ever-evolving front end ecosystem. Most recently I've been exploring new JavaScript frameworks like Astro and SolidJS to expose myself to new ways of building user interfaces and improve my skills as a Front End Engineer.

For Fresh Graduates / College Students

I graduated from Northeastern University in 2020 with a degree in Computer Science. My interests are in Front End development and I enjoy building delightful products on the web because of the freedom to develop whatever comes to mind and the fast feedback loop.

I took a web development course while in school which taught me the fundamentals of full stack development. With these skills, I managed to secure software engineering internships at companies like Uber and Palantir. At Uber, I built an internal tool to help developers manage their test rider and driver accounts using React, Redux, and GraphQL.

I'm interested in the Front End Engineer role at Google because I use Google products on a daily basis and am very impressed by their well thought out product experience. It'd be a dream come true to work on such products that impact the lives of billions.

Hear it from the Hiring Managers

Typically I use the self introduction as an easy warm up to kick start the interview. No technical skills required, but it tests communication and ability to describe things in a clear and concise way.

— Engineering Manager, LinkedIn.

Candidates shouldn't ramble on too deeply when introducing themselves as it was never meant to be a deep-dive, just a quick overview.

— Talent Acquisition, Coinbase.

Could be a good idea to finish with what tech stacks or projects you've worked with, and what you hope to work on in the future/where you want to take your career.

— Engineering Manager, Indeed.

Your interviewer is also a person. The foremost thing on an interviewer's mind is whether they want this person to join their team — it's not just about the skills. We want to know who you are, then what you are capable of. Of course, adding personal flavor with relevance to the job is best.

— Senior Engineer, Dropbox.

D. Behavioral Interview Guidebook

How to Answer "Why do you want to work Here" in Behavioral Interviews

Learn the best way to answer questions about company motivation concisely and effectively, for front end / web developers / software engineers. Written by ex-interviewers at big tech.

As a hiring manager, "Why do you want to work here" is an extremely common question that is probably asked in over 70% of interviews. Yet, most candidates fail to answer this well. In this article, you will learn from ex-interviewers how to answer this question in an effective and concise manner Front End Engineers/Software Engineers:

The interviewer's intention

Suggested answer framework

Example questions and answers

The Interviewer's Intention

Most interviewers ask this question with a single objective in mind — to understand more about you through your motivations to join the company. They want to know whether you were interested enough to do your research, or if there was a special reason that constituted a strong motivation from you.

Suggested Answer Framework

To succeed, there are 3 basic steps to build your answer:

1. It should take between 60 to 90 seconds

Never take more than 60 to 90 seconds. Any longer might constitute a monologue. For reference, 1 min is around 15 statements.

2. Gather deep insights about the company/team/position gained from current/previous employees of the company

If you've been an interviewer before, you'd know that most candidates do not impress in the "why do you want to work in this company / team" question because they tend to answer it based the company's public information, which may not necessarily be representative of the individual team or department's culture or situation. For instance, imagine you are evaluating 2 candidates for a game-development team:

Candidate A: "Company A is very environmentally friendly and I want to be a part of the earth-saving efforts" (as the team's manager, you know none of your team members ever talk about earth-saving as your work is completely unrelated to it)

Candidate B: "Team A is working on games based on Technology B and this is something I happen to be very interested in."

To stand out as someone who is truly interested in the role, you should optimally strive to find insights specific to the team you are joining, and use that to justify your interest (and therefore fit).

Here are some common avenues to get team-specific insights:

Gain more insights about the team's stack or practices from the job description (in some cases, there could be many details you could tap on)

Search up the team on LinkedIn and understand more about their individual work scope, past projects, backgrounds and interests

Try to score an informational interview or coffee chat or short video call with the team members (after doing basic research online)

Search up the team on Blind or even Reddit (go as specific in your query as possible)

Here are some useful questions you could ask:

What are some of the unique challenges or learning opportunities that developers get on your team versus other companies?

What are the kinds of projects I could expect to work on if I were to join your team?

How does the way this team operates differ from other teams in the company?

Does this team work closely with other teams? Which ones and how so?

Can you walk me through the process of a typical project?

3. Briefly explain why joining this company/role/team is a strategic and logical step for your career

In so doing, you could also take the chance to cement the interviewer's impression of your passion for the field or the efforts you have made to build your expertise.

Example questions and answers

"Why do you want to work in the TikTok Shop team?"

I've worked as a freelance web developer while in school for almost 4 years. Having grown significant experience as an individual contributor, the next step for me is to work on larger, more complex projects as part of a bigger team to take my development skills to the next level.

The top 3 things about TikTok commerce for me are:

I like to know what my development efforts are able to tangibly drive impact. At the current stage, TikTok Shop is still at a relatively nascent stage of the product lifecycle where that is possible.

TikTok Shop serves the wide masses and has to deal with associated challenges such as accessibility, localization and e-commerce in a relatively new format, which allows me to develop my skills in that area.

ByteDance (TikTok's parent company) internally develops many cutting edge technologies and also uses open source technologies. I'm excited to learn from the company's experienced engineers on how to scale these technologies to a worldwide audience.

E. Useful Questions to ask your Interviewer at the End of a Software Engineering Interview

Learn the best questions to ask at the end of your front end / web developer / software engineer interviews. Written by ex-interviewers at big tech.

In our anxiety to secure a job, some of us might forget that joining a bad team might be worse than waiting it out for a good opportunity. An interview is in fact an invaluable opportunity for the candidate to speak to someone who actually works in the company and find out more about the practices there. Few key aspects you should evaluate about a job before joining:

Will the work be interesting to you?

Is there room for career growth?

Are the teammates/your manager collaborative and easy to work with?

Good questions asked at the end of the interview help a candidate achieve 2 objectives - leave a good impression and help the candidate assess if the company is truly suitable for them.

In this article, we have provided a large list of useful questions you might ask to achieve this. Questions are organized by interviewer/interview round as well as objective.

Screening rounds with HR/recruiters

The recruiter is usually not working directly in the team who is hiring you. Hence, it could be less insightful to ask questions about the team's day-to-day or on-the-ground situation. However, you could still find out information that recruiters would know, such as the candidate evaluation criteria or general company policies.

Get to know the criteria the hiring team is looking out for

These questions are very useful to understand what the hiring team is prioritizing, which they would probably have communicated with HR/recruiters beforehand so that they could help screen candidates accordingly.

"What are the skills, characteristics or experiences that the team is looking for in an ideal candidate?"

These are invaluable as you would be able to pinpoint the key criteria the team is prioritizing in hiring, which allows you to specifically showcase those required skills, characteristics or experiences in the next rounds with the hiring team.

This question is extremely useful as it's often hard to guess what the interviewer is really looking out for. Hiring is often a hit or miss and hiring managers are usually especially excited to find candidates who seem to be a perfect fit to their internal hiring criteria.

"When is the anticipated starting date for this position?"

More often than not, starting dates could be a key criteria when a position has to be filled urgently. You could check this with the recruiter early if you have a delayed start date, to avoid wasting their time and yours.

Get to know more about the hiring team

"Will I have the opportunity to meet my potential manager or colleagues during the interview process?"

If you were to meet the hiring team within the process, it would be good to know which rounds it would be, and if possible, who's who. This will help you prepare relevant questions or conduct background checks beforehand.

"Can you tell me about the team I'll be working with directly, including who I will report to and the team's structure?"

This question may or may not be useful depending on how familiar the recruiter is with the hiring team (sometimes not at all). However, you might still be able to get good insights on the structure or dynamics of the team.

Get to know more from the company's perspective

"What are the current goals that the company is focused on, and how does this team work to support hitting those goals?"

This question gives you insights on the importance and position of the team within the company, including how the team contributes to the company's key goals. This could provide critical insights on the team's key deliverables and priorities, and correspondingly how you could best serve the team. In bad times, lower priority teams could also be laid off.

"Can you give me some examples of the most and least desirable aspects of the company's culture?"

While HR/recruiting might not be familiar with the team's day-to-day, they would most likely know about the company's policies and culture. This could give you some useful insights which could also carry over to the next interview.

"How frequently do employees make themselves available outside of working hours"

In a similar vein, we want to know if there is a general company culture for or against overtime work (without being too obvious). This could be a dealbreaker for some people, and should be cleared up asap.

Rounds with the hiring team/team you will join

As a front end developer / web developer, rounds with the engineering team usually means rounds with the hiring team itself.

Questions to ask in these rounds are highly critical in elucidating the experience you would likely get when you actually join the team (Note: This section is focused on questions to ask your potential team mates except manager. Questions to ask your potential manager are in the next section).

Check if the day-to-day realities of the job aligns with your goals

"Can you show me examples of projects I'd be working on and the technologies I will use? Will I be maintaining projects or building from scratch?"

This question is useful to elucidate the exact kind of work you would be doing, as well as some of the technologies used. In general, having the opportunity to build products from ground up could be a better learning opportunity than doing small, incremental features.

The types of technologies used could also be a criteria, as learning to use modern stacks (and having mentors proficient in that area) will likely be more beneficial to your career growth.

"Can you describe a typical day or week for a developer in your team?" / "Can you walk me through the process of a typical project"

This question helps to clarify quite a lot of things - the kinds of tasks that a developer might be expected to do, stakeholders that you will work with, as well as the project management process and lifecycle.

"What are some of the challenges you expect the person in this position to face?"

This question is useful in uncovering some potential issues in the codebase or workflows which could be a dealbreaker for certain people. It might also uncover dysfunctional teams or processes.

"What does it look like during the busiest and toughest times for this role?"

Asking this question could help uncover unreasonably high expectations or pressure for the role, elucidating more about work life balance without explicitly asking about it.

"Will there be opportunities for stretch assignments where I can learn and use new skills?"

If you're someone who seeks to grow your skills or portfolio in the role, be sure to ask this question to know if you will have reasonable autonomy to work on more complex projects on request.

Check if your potential manager's managerial style is appropriate for your growth

What type of information is shared with employees? Is revenue, profits, expenses, salary ranges, etc. shared?

This question differentiates companies by their levels of transparency. Some candidates might find this important, as it could be frustrating to have little information and hence little control over your compensation and promotion schedule.

Who I will I report to? Can you tell me more about their style of supervision?

This question elucidates if your potential manager is someone who is hands-off or tends to look into the details of your work. This could have a large impact on your experience on the job.

Understand more about the team dynamics and culture

"What is the team structure and how does this role fit into the team?"

This question is useful as it helps you to paint a picture of the role in context to the overall team structure, how big the team is and roughly what each part of the team does.

"How would you describe the work environment here—is the work typically more collaborative or more independent?"

This question indirectly elucidates the likely culture of the team - do people work in silos or do they communicate frequently with one another.

"Can you give me some examples of the most and least desirable aspects of the team's culture?"

Another open-ended, useful question which allows you to gain more insights which may not surface from a more general ask.

"How frequently do team members make themselves available outside of business hours?"
This question indirectly elucidates the culture of the team with respect to work life balance.

Rounds with your potential manager

In these rounds, it is useful to take the chance to find out your manager's managerial style, the performance review process/criteria, as well as his/her expectations of the role (which could give you an idea of how well you might perform on the job)

Can you tell me about the kind of supervision you provide?

Thinking back to people you've seen do this work previously, what differentiated the ones who were good from the ones who were really great at it?

What are you hoping this person will accomplish in their first six months and in their first year?

What has been the top accomplishments of the organization over the past year?

Rounds with roles outside the team (who might not know specifically about team's operations)

Questions to ask the founder

What are the current goals that the company is focused on, and how does this team work to support hitting those goals?

What direction do you see this company heading in over the next few years?

What are the current focus areas for the company as a whole?

Questions to ask the team's collaborators

It is common to interview with someone you may not work directly with. In these cases, its best to cast a wider net and ask questions about the general company policy or culture. You could still ask them about your team if they are close collaborators.

What's one challenge you occasionally or regularly face in your job?

Can you tell me about the team I'll be working with?

What learning and professional development opportunities are available in the company?

Questions to avoid

Here are a few examples of what not to ask at the end of your interview:

Avoid asking about salary and benefits early on

Only start discussing salary when the team is already keen on hiring you (i.e. at the end of the process), unless they ask you about your salary requirements.

What's the starting salary?

Can you tell me about your health insurance?

What are your paid leave policies?

If I get the job, when can I take time off for vacation?

How often do you give out raises?
How often do you hand out bonuses?
What kind of perks and benefits can I expect?
Avoid asking questions that should be public information
What does the company do?
Where are the offices of the company located?
Who is the CEO of the company?

F. Answering "Tell me about a time when..." for Problem Solving Skills and Drive for Results

Learn how to answer behavioral questions on problem solving skills and drive for results, for front end / web developers / software engineers. Refer to sample answers.

As mentioned in our behavioral interview preparation overview, problem solving and drive for results is one of the 8 main categories of questions to prepare for.

In this guide, you will learn how to tackle them:

Evaluation criteria in detail

Abstracting possible questions into common themes

Suggested answer framework

Possible nature of follow-up questions

Sample questions and answers

Evaluation Criteria in Detail

Problem solving and drive for results are grouped together into 1 category as they are highly related in nature. When describing one's experience in solving problems or to achieve certain results / objectives, their mindset or drive to do so can also be inferred.

When rating candidates under this category, interviewers are often looking at the following criteria:

Identifying best solutions and executing on them

Identifying the right problems

Identifying the most critical objectives

Resourcefulness and data-driven mindset

Creativity and innovation

Identifying tradeoffs and sustainable solutions

Measuring results, iterating and following through

Impact-driven mindset

Proactivity to make progress despite obstacles or roadblocks

Influencing others to deliver against objectives

Balancing analysis with decisive action

Abstracting possible questions into common themes

As mentioned in our behavioral interview preparation overview, it is impractical to prepare answers specifically for every behavioral question out there. However, by batching specific

questions into similar themes and preparing stories that cover a large number of question requirements, we can reduce the number of stories to prepare to around 3-5 stories.

There's simply too many types of problem solving and drive for results behavioral questions that could be asked, such as:

Can you tell me about a time when you had to use data to drive engineering decisions?

Can you provide an example of a time when you had to troubleshoot and fix a complex issue in a project?

Can you describe a time you creatively solved an engineering problem or achieved meaningful metric improvement?

Can you describe a time when you had to make an important engineering decision and how you decided between tradeoffs?

However, when we look at 80% of questions under this category, questions typically ask for how a problem was solved or a specific trait required for effective problem solving, such as creativity, using data or trade-off evaluation. The source of the problem or objective, as well as whether the candidate was resilient in the face of obstacles can be inferred from answers to these questions as well.

Suggested Answer Framework

As always, the STAR format is the simplest and most effective framework that we recommend to structure your story.

Despite the large number of question possibilities, we can abstract all the requirements for problem solving behavioral questions by thinking about a robust problem solving process:

Problem identification: Identifying the correct root cause or root problem to focus on from surface level symptoms.

Metrics/target setting: Identify key metrics that would signify success in solving the problem (if any).

Information gathering: Resourcefulness in gathering information from different sources and a data-driven mindset (using data to drive decisions).

Solution brainstorming: Creatively generate solutions that solve the root cause

Solution evaluation: Evaluate trade-offs of each solution and select the most optimal one

Monitoring and adjustment: Monitor effectiveness of the solution by measuring key metrics.

Adjust strategy if needed.

Hence, you just have to ensure that at least 1 of your prepared stories or projects covers all of the above steps. In doing so, the story could be re-used for all affiliated traits under problem solving, such as data-driven mindset, resourcefulness, creativity, and working with feedback from users.

Of course, you might need to tailor the details of your story to focus on the specific aspect asked in the specific question.

We recommend you to also pick problem solving stories that can doubly serve to get signals on the following

Proactivity/initiative-taking: you had taken the initiative to look into the problem, gather information and solve it.

Leadership: you had led the problem solving process

Teampayer: you had to work as part of a team to solve the problem

Sample Story

Situation

I was the tech lead for an e-commerce website selling luxury goods. The website was built as an Angular 1.5 single page application.

In recent years, the product was showing its age – developer experience was not great and the website performance was poor. Initial loading speed was over 3 seconds and the conversion rate was around 0.8%.

Task

I was tasked with improving the performance and conversion of the website.

Action

1. Problem identification

Conversion is tied to performance and UX.

Website performance has been on a gradual decline over the past few years.

UX hasn't been looked at in a while. [Improve phrasing]

2. Information gathering

Looked at nature of bugs in the past year, categorized them according to their root causes to identify hotspots and major problematic areas.

Gathered feedback from team regarding areas of improvements.

Brainstorming session with the team to think of ways to improve.

In order to improve, firstly we need to know how we are doing.

Double checked that our performance and conversion tracking was working correctly.

Started tracking new metrics from Lighthouse and Core Web Vitals.

Worked with Data Scientists to come up with dashboards for performance and conversion and gained some insights:

Identified that some countries had lower conversion rates.

Mobile users had lower conversion rates as compared to desktop.

Worked with UX Designers and UX Researchers to identify problems in the end-to-end shopping experience on the website.

UI elements were too spaced apart and required a lot of scrolling which affected bounce rate because some users didn't bother scrolling.

3. Solution brainstorming

User Interface: Server side rendering is crucial for its performance and SEO benefits. Made choices around good performance.

View: Migrating from Angular.js 1.6 to Angular 13 was a huge undertaking and there was no significant time saving by staying on Angular.

Next.js: A few of our developers have experience with React and Next.js as a meta framework for building SSR applications was rapidly rising in popularity. We really desired the fast initial load and app-like behavior that Next.js provided.

Svelte: Reactive model was appealing and the programming model easier to understand as compared to React's, however the ecosystem is small and there aren't that many libraries.

Styling: The stylesheet was getting very bloated due to many classes being added over the years and being hard to remove.

Tailwind CSS: Tailwind CSS was among the hottest CSS methodologies and its atomic CSS approach scales well for growing code bases.

Styled Components: CSS-in-JS was something we were also looking at, but Styled Components was tied to React and runtime style injection was bad for performance.

Performance-centric mindset. Read many performance case studies on web.dev and engineering blogs of other e-commerce companies, gathered a list of important performance techniques and processes:

Set performance budget for each page (under 300kb).

Run performance benchmarks before merging Pull Requests.

Lazy loading of non-critical components Lazy load below-the-fold contents.

Split JavaScript on page level instead of a single bundle (handled by Next.js).

Use WebP format for images.

Host images on a CDN instead Adaptive loading of images so that mobile devices load a smaller image Consolidated duplicate JavaScript libraries (data-fns and moment.js), switched to lodash-es, removed all usages of jQuery - Looked at data to identify less used features and removed them from the code, reducing JS size on the product details pages by 200+kb.

Search Engine Optimization (SEO)

Used SEO tools like Ahrefs to continuously monitor SEO.

Worked with the marketing team to ensure marketing copy included important keywords as shown by Ahrefs.

Adjusted page URLs to include SEO keywords

User Experience improvements

Single page checkout experience as opposed to two-page checkout to reduce clicking.

Reduced height of many UI elements.

Fixed checkout button that wouldn't be missed.

Payment improvements

Analyzed Stripe checkout and implemented country-specific address fields.

Initially only had one available payment method: credit card. Got the help of Data Scientists to evaluate the popularity of new payment methods and whether they were worth adding. We later added PayPal, Google Pay, and Apple Pay payment methods as well.

4. Solution evaluation

View and Rendering: Chose Next.js because it is backed by Vercel and has the largest community of all. React is also the most popular UI library out there and also easiest to hire jobs for.

Styling: Tailwind because it's a pretty reliable and futureproof option.

5. Monitoring and adjustment

Rolled out the new website behind an A/B test while monitoring the performance and conversion rates over a period of 2 months.

Countries which previously saw lower conversion rates experienced nearly 50% improvements in conversion rates.

Result

Lighthouse score has improved to 92.

Loading speed has improved to 1.5 seconds

Conversion has improved to 2.5%

Developer velocity has improved in the recent survey and it is now easier to hire people into the team because more people know React over other frameworks.

Possible Nature of Follow-up Questions

As alluded to in our behavioral interview preparation overview, interviewers are encouraged to rely more on follow-up questions to really understand the candidate's thought process and motivations, which typically fall under these categories:

Why do you think you did (insert action)?

Why did you not do (insert action)?

How would you do things differently in hindsight?

For questions on problem solving or drive for results, interviewers will most likely probe for questions to help them understand a bit more about:

The source of the task/problem/objective (Understanding level of proactivity and initiative):

Was the project or task initiated by you? And to what extent?

Was the underlying idea from you, or just the plan to execute?

How did you get buy-in from stakeholders to even work on it?

Role and actual contribution by candidate:

Was there a team involved with solving the problem or achieving the objective?

Which actions were initiated on your own accord or contributed solely by you, and what did others do?

Prioritizing the problem or objective to achieve:

Was there a reason why this particular problem or objective was prioritized to be addressed or done against others?

Was there another problem or root objective that should have been more important?

Was this problem not already resolved by another team in the company? Why did the solution have to be reinvented?

Selection of appropriate metrics/objectives, and if they were measured post-launch:

Was there any quantitative or qualitative target set, and how were they decided?

How were they measured post-launch, and what was the result?

If an issue was resolved, what measures were put in place to ensure the same problem didn't happen again?

Using sufficient information to drive decisions:

What types of research or data did you rely on to make decisions / choose a problem to resolve?

How much time did you spend on research and gathering information?

How did you balance between gathering information / planning and actually executing on the plan?

Choosing the right solution:

What other solutions were considered, and what were the pros and cons of each?

Why was the final solution chosen?

Who came up with the original idea of the solutions? How were they derived?

Sample Problem Solving Questions and Answers

Tailoring the above sample story to answer specific questions.

Can you tell me about a time when you had to use data to drive engineering decisions?

Can you provide an example of a time when you had to troubleshoot and fix a complex issue in a project?

Can you describe a time you solved an engineering problem or achieved meaningful metric improvement?

Can you describe a time when you had to make an important engineering decision and how you decided between tradeoffs?

G. Answering "Tell me about a time when..." for Collaboration Skills

Learn how to answer behavioral questions on collaboration, including communication and teamwork, for front end / web developers / software engineers. Refer to sample answers.

As mentioned in our behavioral interview preparation overview, collaboration is one of the 8 main categories of questions to prepare for.

Collaboration questions are probably the most common behavioral questions asked in front end behavioral interviews, as they encompass a large number of related traits which can be grouped for ease of story/project preparation.

In this guide, you will learn how to tackle them:

Evaluation criteria in detail

Abstracting possible questions into common themes

Suggested answer framework

An example of a good sample story

Possible nature of follow-up questions

Sample questions and answers

Evaluation Criteria in Detail

When rating candidates under this category, interviewers are often looking at the following criteria:

Handling disagreements

Working as a team

Working with diverse personalities and skills

Conveying complex ideas simply

Giving constructive feedback

Active listening

Abstracting Collaboration Questions

As mentioned in our behavioral interview preparation overview, it is impractical to prepare answers specifically for every behavioral question out there. However, by batching specific questions into similar themes and preparing stories that cover a large number of question requirements, we can reduce the number of stories to prepare to around 3-5 stories.

Collaboration is such a theme which can group sub-requirements like communication, team work, adaptability and coachability. For each sub-requirement, we have summarized the common questions and also abstracted out the traits that interviewers are looking out for across these questions:

Communication

Example Questions

Can you describe a time when you had to effectively communicate technical information to a non-technical audience and how you approached it?

Can you describe a time when you had to adapt your communication style to effectively communicate with someone who had a different background or perspective?

Can you give an example of a time when you had to communicate important information under time pressure or in a high-stress situation?

How do you ensure that your message is understood and well-received by your audience?

Traits Interviewers Look Out For

Convey complex ideas simply

Active listening

Teamwork

Example Questions

Can you describe a past project where you had to work with a difficult stakeholder or teammate and how you dealt with it?

Can you tell me about a time when you had to give constructive feedback to a team member or colleague?

How do you approach working with team mates who are not meeting their deadlines or responsibilities?

Can you provide an example when you had to work with a team where there were disagreements between members?

How do you handle working on a team with people who have different skill sets and personalities?

Can you give an example of a time when you had to adapt to a team member's working style in order to complete a project successfully?

How do you ensure that all team members are heard and their ideas are considered in group discussions?

Can you give an example of a time when you had to work on a team project and had to compromise with others to reach a solution?

Can you describe a time when you had to collaborate with people from different departments or organizations to complete a project?

Traits Interviewers Look Out For

Working as a team

Working with diverse working styles, skills and personalities

Handling disagreements (others and own)

Giving constructive feedback

Suggested Answer Framework

As always, the STAR format is the simplest and most effective framework that we recommend to structure your story.

Based on the abstraction above, we can see that interviewers are looking for specific traits for every sub-requirement. We have created a quick cheatsheet for suggested approaches to exhibit these desired traits.

Ideally, you should select stories that can cover as much of the below traits as possible, so that you don't have to remember too many stories.

Convey Complex Ideas Simply

First understand your audience's level of knowledge. Put yourself in their shoes, explain in their language.

Understand what they need to know (and want to know) and prioritize the key takeaways.

Only dive deeper where appropriate for important details.

Break down the idea into component parts.

Use visual aids, such as diagrams or charts.

Give examples.

Note: For communication, traits like conveying complex ideas simply and active listening may be evaluated through your actual interview performance instead - e.g. do you interrupt the interviewer, do you listen and understand their question correctly, do you communicate your thought process in an easy to digest manner.

Active Listening

Focus on understanding rather than replying.

Withhold judgment until the person is done talking.

Use non-verbal cues to show that you are engaged.

Summarize and repeat back in your own words.

Ask clarifying questions.

Never interrupt.

Working as a Team

Set clear goals and ensure that everyone understands them. Proactively seek alignment and perspectives.

Assign and communicate roles and responsibilities. Collaborate and delegate.

Set up regular progress check-ins and address blockers. Always include the appropriate stakeholders and share timely information with them.

Measure impact and recognize achievements.

Working with Diverse Working Styles, Skills and Personalities

Openly recognize unique perspectives and skills that each team member can bring to the table.

Encourage open and honest communication, and active listening.

Create a welcoming and supportive environment even to differences in opinions. Use them as opportunities to grow.

Proactively include a range of perspectives and voices in decision-making.

Provide equal opportunities to every team member, including access to channels, resources and support.

Handling Disagreements

Facilitate open and productive communication between relevant parties.

Frame conflict as a way to enhance teamwork and improve current solutions.

Clarify the source of conflict (and whether there actually is conflict)

Give each party equal time to air out their views and concerns without judgment (assume positive intent). Set ground rules if needed to cultivate active listening and understanding. Pivot conversation away from emotions and towards solutions. Tactfully address unproductive dialogue.

Summarize and validate the standpoint from each party, reflecting back at them.

Identifying underlying sources of conflict between the standpoints.

Brainstorm and run through the options available to best meet common goals. (Show skill and logic in finding common ground). Use data and facts to drive resolution with others, weighing pros and cons, instead of just relying on opinions.

Agree on the best solution and determine each party's responsibilities. Bring in relevant parties to support resolutions.

Giving Constructive Feedback

Give the feedback in private.

Remind them of what you already appreciate of them.

Describe specific behaviors that were directly observed (not inferred) and can be changed, e.g. "did not study the documentation" versus "not prepared".

Describe the impact of said behaviors.

Point it out as an opportunity for growth rather than a fault.

Offer solutions.

Sample Story

Although there seem to be a large number of required traits, you can cover 90% of them with a very common situation that occurs in engineering teams:

You had to work within a cross-functional team (e.g. with business stakeholders or product managers or designers) on a high pressure situation, where priorities and requirements were changing rapidly.

There's a conflict of interest between business/design and engineering, such as business/design pushing (demanding) for tighter deadlines, while from engineering perspective rushing for those deadlines will result in large tech debt.

You had to provide constructive feedback to business/design on the problem. In so doing, you needed to convey technical concepts to the non-technical audience. Further, you solicited feedback from them to understand how engineering could collaborate better to avoid this problem again. In so doing, you regularly ran through the list of demands or requirements with them to see if any could be dropped for faster engineering delivery. Additionally, you catered for their need on timeline accountability by providing regular updates. Add in specific details according to your situation.

Situation

In my current job as front end engineer in a startup, I had to lead development on a highly urgent marketing project, working cross functionally with marketing and design. At some point, marketing had expected the feature to ship within the next week, but there was a significant roadblock from engineering due to dependencies on a partner API team whose deliverables were delayed due to recent departures of two senior members. The situation had become heated as marketing did not understand the roadblocks engineering had faced.

Task

I had to resolve the misunderstanding to ensure smooth working relationship between both teams.

Action

In order to do so, I spoke to marketing privately and gave them time to clearly explain the urgency and concerns or misconceptions marketing might have towards engineering. I then explained the role of the partner teams, focusing specifically on their impact on the feature as opposed to complex technical information. In so doing, we were able to brainstorm different ways we could ship the feature even without the external dependencies. In parallel, I also spoke to the partner team's manager on the importance of the project to the company and the possibility of reprioritizing their work to help unblock us. Apart from that, I also solicited feedback on how engineering could better collaborate with marketing to avoid such misunderstandings in future.

Result

Due to the tight collaboration between engineering and marketing, we were able to ship the feature in a timely manner. As marketing prioritized timelines and accountability, we started providing regular updates and discussed alternatives for every potential blocker. With this, every feature was successfully and smoothly delivered there after.

Possible Nature of Follow-up Questions

As alluded to in our behavioral interview preparation overview, interviewers are encouraged to rely more on follow-up questions to really understand the candidate's thought process and motivations, which typically fall under these categories:

Why do you think you did (insert action)?

Why did you not do (insert action)?

How would you do things differently in hindsight?

For questions on collaboration, interviewers will most likely probe for questions to help them understand a bit more about:

Understanding if and why certain stakeholders were involved or not involved

Which stakeholders were involved in the discussion, and why?

Why were certain stakeholders like (insert stakeholder) excluded from the group?

Understanding group dynamics and how it affected the strategy to handle the group

Which personalities were present in the group and how did it affect dynamics? Did this understanding affect your strategy to navigate group discussions?

Understanding the candidate's mindset to conflict

How do you think conflicts like this affect the working environment? Would you prefer to avoid them?

Understanding how the resolution was derived; did the candidate merely defer to others as opposed to relying on facts and data?

Were there any data and information which were leveraged in making the decision?

Was there an analysis of each solution's pros and cons?

Sample Collaboration Questions and Answers

The sample story above already answers most (12) of the example questions above. Questions which require slight modifications are covered below. If you haven't already, do review our tips on your interview disposition to increase your chances of setting a good impression.

How do you ensure that your message is understood and well-received by your audience?

You could answer the question with a slight modification of the sample story, focused on the explanation of the technical information:

To ensure that my message is well understood, after explaining, I will provide a few examples to illustrate my points and also ask my audience a few questions to confirm their understanding. In a project I did a few months ago, I had to explain to the business team why and how an urgent feature had been delayed by external team API delays. To do so, I focused on the key takeaways that business needed — which was how the feature was impacted by such a delay, and utilized visual aids as well as used simple and clear language. I also gave examples while asking questions to ascertain their understanding of my explanation. This allowed the business team to understand the dependency well enough to brainstorm ways to deliver the feature even without the dependencies.

How do you approach working with team mates who are not meeting their deadlines or responsibilities?

Once it is clear that their behaviors may constitute a concerning pattern, I would let them know in a timely manner. For instance, I had a teammate on a project who had missed on bug fix timelines regularly, and once it happened a number of times, I scheduled a private meeting with him to talk to them about it. I focused on framing the conversation as a growth opportunity for him on top of what is already appreciated about them, which was his knowledge and mentorship as a senior developer. Instead of describing a general behavior such as "being late", I mentioned the specific tickets that were missed and the impact that had occurred from missing

them, as well as its effect on the morale of the team. I then offered a few solutions, such as enacting a new reminder bot within the channel for tickets that were due soon. He was able to understand where I was coming from and our team was able to meet our deadlines much better after that.

H. Answering "Tell me about a time when..." for Growth Mindset

Learn how to answer behavioral questions on growth mindset, for front end / web developers / software engineers. Refer to sample answers.

As mentioned in our behavioral interview preparation overview, growth mindset is one of the 8 main categories of questions to prepare for.

In this guide, you will learn how to tackle them:

Evaluation criteria in detail

Abstracting possible questions into common themes

Suggested answer framework

Sample questions and answers

Evaluation Criteria in Detail

When rating candidates under this category, interviewers are often looking at the following criteria:

Having self awareness of strengths and development areas.

Actively soliciting feedback from others and creating feedback loops.

Taking feedback and setbacks well, viewing them as opportunities.

Takes personal accountability for failures.

Reflecting, modeling best practices and applying lessons to new situations.

Energized by setting and working towards challenging goals.

Proactively learning new skills and applying to work.

Abstracting Growth Mindset Questions

Common Questions

What do you think is your greatest area for improvement?

Can you tell me about a time you improved upon your work upon receiving criticism or feedback?

Tell me about a time you set a difficult goal for yourself and achieved it.

Tell me about a time you learnt a new skill outside of work and applied it to your work.

Tell me about a time you learnt a new skill set after observing others leveraging it to great success.

How do you stay up to date on the latest front end technologies and techniques?

Can you provide an example of a time when you had to adapt to a new technical environment or framework?

Imagine you need to implement a new system that you do not have much experience with. What steps do you take to gather information and ensure successful execution?

Themes Identified

As mentioned in our behavioral interview preparation overview, it is impractical to prepare answers specifically for every behavioral question out there. However, by batching specific questions into similar themes and preparing stories that cover a large number of question requirements, we can reduce the number of stories to prepare to around 3-5 stories.

These are the themes which seem to recur:

Receiving and managing feedback.

Mindset towards challenges and stretch goals.

Continued learning.

Suggested Answer Framework

As always, the STAR format is the simplest and most effective framework that we recommend to structure your story.

Here are our recommended approaches for each identified theme, with which you can structure your story around:

Receiving and Managing Feedback/Setbacks

Actively solicit feedback from others where possible e.g. after every large project.

Take personal accountability for mistakes and produce an action plan to address them.

Have a healthy mindset towards feedback / setbacks

View them as opportunities to improve yourself

Focus on being constructive and solutions-focused than emotions-driven.

Take steps to reflect upon past experiences in order to model best practices and put them into action in the next relevant situation.

Mindset Towards Challenges / Stretch Goals

Derive energy from setting and working towards stretch goals, seeing them as opportunities for growth.

Display consistency and discipline in following through with detailed plan.

Continued Learning

A good answer combines a consistent schedule and effort placed into learning, as well as how the learning was proactively applied to work. e.g. "I learned about X because I was doing Y. This impacted the way that I was tackling Z at work because it made me realize W and V".

The points below are key ideas you can structure your answer around technical learning; however, you should still add on concrete examples of events, newsletters or communities you have actually participated in, to showcase your willingness to keep up with trends.

Follow relevant blogs, newsletters, podcasts that cover the latest developments in front end technologies and techniques. Give some examples (which you do actually follow) and the content or tech they provide.

Twitter Influencers

Dan Abramov — Co-creator of Redux, React core team.

Evan You — Creator of Vue.js and Vite.

Rich Harris — Creator of Svelte and Rollup.

Kent C. Dodds — Popular React ecosystem course creator.

Josh W. Comeau — Popular front end course creator.

Blogs & Magazines

CSS-Tricks

ui.dev

JavaScript Weekly

Best of JavaScript

Podcasts

Syntax

JS Party

The Changelog

Attend tech conferences and meetups to learn new developments, network and get hands-on experience with the latest tools and technologies.

Experiment and build projects using the latest technologies and techniques. This could involve building small projects on your own or contributing to open source projects.

Stay engaged with the broader front end development community by participating in online forums, following relevant Twitter accounts and hashtags, and joining local user groups or meetups.

Sample Questions and Answers

How do you stay up to date on the latest front end technologies and techniques?

I have been working at a big tech company for the past two years now and the company uses a mix of external and internal technologies for our front end stack. Because we have a huge code base and established practice of doing things, we tend to not upgrade libraries that often and change our technology choices. It was only after a year into the job that I realized that I have been too comfortable in my role since I have fully ramped up on the code base and haven't learnt anything new for the past year. It was then that I decided that I have to be proactive in my learning in order to keep my skills sharp and relevant.

The Front End ecosystem moves really fast and there's a joke that there's a new JavaScript library emerging every day. Thankfully the ecosystem is more stable compared to 2015/2016 when a new wave of front end tooling first emerged. To keep myself updated without suffering from JavaScript fatigue, I spend a weekend every month going through front end newsletters like "This Week in React" and "JavaScript Weekly" and reading engineering blogs. If I see something interesting or is relevant to my job, I will dive deeper into them by trying out small examples and building small prototypes in my company's code base. For example, I recently discovered React Query, which is a data fetching library for React that uses a declarative paradigm for fetching data and am trying to incorporate it into my work. I also recently discovered tRPC, a library that enforces type safety between client and server boundaries, which is an issue that has caused some bugs for us.

To make this learning process more fun and collaborative, I started a front end social club within my company by creating a new Slack channel and inviting a few like-minded coworkers to join. Anyone can share front end news they find interesting there and discuss front end technologies.

It has been pretty well-received, with over 30 people currently in the channel and activity nearly everyday. A side effect of this initiative is that people also start asking questions about front end issues they are facing at work, and we created another Slack channel for people to get help on front end issues. Feedback has been really positive so far!

This answer uses the STAR technique in a subtle fashion and demonstrates many qualities interviewers would like to see in candidates:

Passion: Keen to learn more about front end domain.

Growth: Identified that the lack of growth would be a problem.

Initiative: Acted on the problem of lack of growth.

Problem solving: Thought of ways to solve the lack of growth, and executed on it.

Building camaraderie: Rallying like minded folks for a common cause.

Helps improves others: Created Slack channels for sharing knowledge and supporting others.

Can you provide an example of a time when you had to adapt to a new technical environment or framework?

When I joined my current company back in 2019, I was fresh out of college and had no experience with the front end stack that the company was using, which used React, Apollo, Styled Components, TypeScript for their front end stack and a Django + GraphQL back end. Back in college, I was mainly using Vue.js for front end development as it was the JavaScript framework of choice taught in the web development class and haven't used GraphQL before.

It was definitely a huge learning curve for me at the start and I was really afraid of underperforming due to my unfamiliarity with the tech stack. Thankfully I had a mentor who gave me a lot of guidance on the tech stack and came up with a number of onboarding tasks which progressively got harder. I first spent a few days reading the documentation websites of the technologies and trying out the examples on their websites. I made it a point to understand the problems that these libraries were solving and how they were better than prior art because I think knowing that is important for fully appreciating the library and using the right tool for the right job. I also looked up some resources that compared Vue.js to React, as they were quite similar yet had some differences. That helped me to learn React faster (fully grasping React hooks still took me some time though). When I had time at nights or on weekends, I'd explore building small projects using these new technologies and also rebuild my personal blog using Gatsby (because it used both React and GraphQL).

Within two months, I felt like I had learnt so much and was comfortable with most of the code base. I could build end to end features without much guidance from my mentor. To help future new employees who might face the same onboarding challenge, I jotted down my learnings in our internal wiki along with links to the best resources for learning the topic. My manager was surprised by the initiative and commended me for that. Today, it is part of our official engineering onboarding resource and I update it every once in a while. A few new joiners have also thanked me for sharing my knowledge in the wiki and making their onboarding process smoother.

Analysis of the answer:

Coachable: Follows guidance/suggestions by the mentor.

Problem solving: Came up with a few ways to help themselves onboard faster.

Initiative: Initiative to start the wiki.

Empathy: See that others might face the same issues and started the wiki.

Helps improves others: Newjoiners benefit from the wiki.

Resource = <https://www.greatfrontend.com/behavioral-interview-guidebook/>