

Q: Tell me about your biggest challenge in that project.

A: The toughest part was aligning technical decisions with tight deadlines. I handled it by breaking the task into smaller deliverables and setting micro-deadlines. It kept progress measurable and the pressure manageable.

Q: What was your biggest achievement in that role?

A: Delivering a fully functional feature ahead of schedule — not by overworking, but by optimizing communication and cutting unnecessary rework. That taught me execution beats overplanning.

Q: If something didn't go as planned, what did you do?

A: I don't panic. I backtrack the root cause fast, communicate it clearly to the team, and propose a fix. My focus is always on solving, not assigning blame.

Q: How do you handle deadlines?

A: By assuming they're non-negotiable. I set internal checkpoints earlier than the final date so I have a buffer for debugging and reviews.

Q: What did you learn from that internship/project?

A: Technical skills were obvious, but what stuck with me is clarity — writing cleaner code, better communication, and knowing when to ask for help instead of wasting time solo.

Q: What's one thing you'd improve about your last work?

A: Documentation. I've seen how much time unclear notes waste. Now I maintain a simple README or Notion doc for everything I touch.

Q: How do you prioritize tasks?

A: I use the 80/20 rule — focus on what impacts the end result most. If it doesn't move the project or customer forward, it's secondary.

Q: What do you do when you disagree with a teammate?

A: I don't argue feelings — I argue data. I show results or performance metrics. If the data proves me wrong, I accept it and move on.

Q: How do you handle pressure or multitasking?

A: Pressure's normal. I make a quick list, sort by urgency, and focus on completing one small chunk at a time. Momentum kills overwhelm.

Q: Why should we hire you?

A: Because I don't just execute — I take ownership. I don't wait for instructions; I find what needs to be done and get it moving.

Q: What's your biggest strength?

A: Adaptability. I've worked in different tech stacks and roles — instead of freezing when something's new, I break it down, learn fast, and deliver.

Q: What's your weakness?

A: I sometimes dive too deep into fixing details. I've learned to step back and ask — 'Is this detail changing the result or just eating time?' That keeps me balanced.

Q: Where do you see yourself in 2–3 years?

A: In a role where I'm trusted with ownership — maybe leading small modules or mentoring juniors. I care more about depth and impact than titles.

Q: How do you stay updated with tech/industry?

A: I follow engineers, not influencers. I read real GitHub repos, release notes, and newsletters like TLDR or Dev.to — that's where real stuff happens.

Q: Do you have any questions for us? (Never say "No.")

A: Yeah — what does success look like in this role after 3 months? I like knowing what I'll be measured on before I start.