Technical Interview Preparation Guide

E-Solutions

11/1/25

Technical Interview Preparation Guide

BEFORE THE TECHNICAL INTERVIEW

• Study the job description and list all required technical skills (languages, tools, frameworks, platforms).

• Review fundamentals: Data Structures, Algorithms, OOPs, Databases, OS, Networking, System Design.

• Prepare clear explanations for your past projects architecture, decisions, challenges, results.

• Practice coding on platforms like LeetCode, HackerRank, CodeSignal or according to your expertise.

• Update GitHub/portfolio with your latest work.

• Prepare 4–5 technical questions for the interviewer (stack, roadmap, expectations).

DAY OF THE INTERVIEW

• Ensure laptop, internet, compiler, and IDE are working.

• Keep resume + notes ready.

• Dress professionally.

• Join 5 minutes early; keep your environment distraction-free.

• Communicate clearly while explaining technical concepts.

DURING THE INTERVIEW

• Think aloud—explain your reasoning before coding.

• Clarify requirements before solving.

• Break the problem into steps.

• Offer a working solution first, then optimize.

• If stuck—share your thought process.

• Never bluff; explain how you'd approach learning unfamiliar concepts.

AFTER THE INTERVIEW

• Send a thank-you email summarizing your interest.

• Revise concepts you missed.

• Follow up within a week if required.

ROLE-WISE TOPICS TO REVISE

• Developers: DSA, algorithms, OOPs, SQL, APIs, debugging, Git, System Design.

• QA: STLC, test cases, automation tools, API testing, JIRA.

• DevOps/Cloud: Docker, Kubernetes, CI/CD, AWS/Azure/GCP.

• Data Engineers: SQL, ETL, pipelines, warehousing, Python.

• Networking: Routing, switching, protocols, firewalls.

• Cybersecurity: OWASP, SIEM, vulnerabilities, encryption.

• Engineering: Core subject knowledge, formulas, standards, drawings.

CODING INTERVIEW TIPS

• Start with a simple explanation of the problem.

• Use pseudocode when needed.

• Write clean, readable code.

• Test edge cases.

• Explain time/space complexity.

• Share scalability improvements.

SYSTEM DESIGN BASICS

• Understand client–server architecture.

• Load balancing, caching, queues, microservices.

• Draw simple diagrams.

• Discuss trade-offs and choices.

TECHNICAL BEHAVIORAL QUESTIONS

• Describe a technical problem you solved.

• A time you optimized a system.

• Working across teams.

• A production issue you resolved under pressure.

QUESTIONS TO ASK THE INTERVIEWER

• What tech stack does the team use?

• What challenges is the engineering team solving now?

• How is success measured in this role?

• What does the product architecture look like?

• What tools/frameworks should I know from day one?

THANK YOU EMAIL EXAMPLE

Dear [Interviewer Name],

Thank you for the technical discussion regarding the [Position Name]. I enjoyed learning more about the role and your tech stack.

I'm confident that my experience in [mention skills/technologies] aligns well with the expectations.

Looking forward to hearing from you.

Best regards,

[Your Name]