

Back-End Software Engineer

General interview process overview

- 1) Recruiter Screen (30 minutes)
- 2) Technical Screen (1 hour)
- 3) Team Matching (internal review process, nothing candidates need to prepare for!)
- 4) Full Panel (combination of 4–6 interviews listed below at 1 hour each)
- 5) Hiring Committee (internal review process, nothing candidates need to prepare for!)
- 6) References
- 7) Offer

CoderPad prep

- [CoderPad Sandbox](#)
- [CoderPad Languages and Frameworks](#)

Typical Back-End panel interviews

We are language agnostic for our interviews, but we do expect fluency in the programming language of your choice (i.e, understanding of core data structures, performance/memory & language constructs). Interviews generally consist of: Coding, Algorithms, System Programming, Architecture, Domain Deep Dive, and Cross-Functional

- **[Coding]** This implementation-focused interview evaluates your ability to write production-quality code. You'll be expected to develop functional solutions with clean code organization, comprehensive test coverage, and careful edge case handling. We'll assess your practical knowledge of algorithms and data structures, along with your ability to analyze solution efficiency using Big-O notation.
- **[Algorithm]** This algorithm and data structures interview assesses computer science fundamentals, your analytical reasoning, and your problem-solving methodology. You'll implement solutions while focusing on algorithmic efficiency and appropriate data structure selection. We expect you to analyze time/space complexity using Big-O notation, demonstrate comprehensive testing strategies, and clearly articulate your approach to edge cases and design decisions.

- **[System Programming]** This systems programming interview evaluates your understanding of core concepts, including multi-threading, synchronization, I/O operations, and performance optimization. You'll be asked to design and implement system components using pseudocode, demonstrating knowledge of fundamental systems primitives (files, buffering, caching, storage, concurrency) and their practical applications. We'll cover both single-threaded and multi-threaded approaches, with emphasis on code design principles such as separation of concerns, cohesion, coupling, and the single responsibility principle. While specific language proficiency isn't required, you should be prepared to write detailed pseudocode to explore various technical trade-offs and solutions.
- **[Architecture]** This high-level architectural design interview evaluates your ability to conceptualize end-to-end systems from database to UI. You'll likely whiteboard with CoderPad Draw and create comprehensive system designs that address scalability, reliability, component interactions, and request flows. We expect strong technical judgment, clear articulation of trade-offs, and in-depth knowledge of storage systems, distributed computing concepts, and interface design. The discussion may encompass APIs, networking, concurrency, caching, replication, consistency models, and failure handling. You should demonstrate the ability to design robust, scalable solutions while collaboratively defining requirements and justifying your architectural decisions.
- **[Domain Deep Dive]** This in-depth technical conversational interview explores your experience with architectural and scaling challenges you've solved. We'll examine your problem-solving approach, technical decision-making, and domain expertise. You may be asked to work through a relevant architectural problem related to your background, though it can be leveraged. The goal is to facilitate a meaningful technical exchange where both interviewer and candidate gain new perspectives within your technical domain.
- **[Cross-Functional/Hiring Manager]** This behavioral interview will be conducted by one of our Engineering Leaders, potentially your direct manager. The conversation will explore your career trajectory, job search motivations, and interest in Databricks. You'll discuss significant projects you've contributed to and how they aligned with broader company objectives, along with your approach to problem-solving, decision-making, and professional growth. We're also interested in your experiences

with cross-team collaboration, overcoming challenges, and leadership qualities relevant to your career level.

While this isn't a technical assessment, please be prepared to explain your projects with technical depth, avoiding company-specific terminology that might not be familiar to those outside your current organization.

Front-End/Full-Stack Software Engineer

Typical Front-End/Full-Stack interviews

Interviews generally consist of: Coding, System Programming, Product Design, Front-End Infrastructure, and Cross-Functional

- **[Front-End Code]** This front-end coding interview assesses your web development fundamentals through building a browser-based application that fetches and displays remote data. You'll demonstrate proficiency with asynchronous operations, efficient data fetching, and state management across UI components. We support vanilla JS/DOM, React, Angular, Vue, and Svelte, but if you prefer another framework, please familiarize yourself with configuring it in CoderPad's environment by reviewing the CoderPad resources before your interview.
- **[Front-End Systems]** This front-end design interview evaluates your ability to architect scalable solutions for real product experiences. You'll demonstrate expertise in client-server interfaces, API design, state management, and data synchronization strategies. The deliberately open-ended problem statement requires you to proactively clarify requirements before proposing solutions. Unlike typical design interviews, this session includes hands-on coding components, so be prepared to implement aspects of your architecture. We'll assess both your high-level system design thinking and practical implementation skills.
- **[Product Design]** This collaborative interview involves meeting with a member of the Product Design team. You'll tackle hypothetical design challenges that mirror the journey from Figma specifications to functional implementations. We'll evaluate your approach to interpreting design requirements, anticipating implementation challenges, and navigating potential constraints. The primary focus is on your communication style and collaborative problem-solving with design team members,

which will provide insight into how effectively we would work together in real production scenarios.

- **[Front-End Infrastructure]** This conversational interview evaluates your ability to optimize team productivity with complex code bases. We'll discuss your experience with large-scale refactors, migrations, testing infrastructure, build systems, performance optimization, and code quality initiatives. Come prepared with specific examples demonstrating how you've successfully led technical teams through significant code base transformations and improvements.
- **[Cross-Functional/Hiring Manager]** This behavioral interview will be conducted by one of our Engineering Leaders, potentially your direct manager. The conversation will explore your career trajectory, job search motivations, and interest in Databricks. You'll discuss significant projects you've contributed to and how they aligned with broader company objectives, along with your approach to problem-solving, decision-making, and professional growth. We're also interested in your experiences with cross-team collaboration, overcoming challenges, and leadership qualities relevant to your career level.

While this isn't a technical assessment, please be prepared to explain your answer projects with technical depth, avoiding company-specific terminology that might not be familiar to those outside your current organization.

Database Software Engineer

Typical Database interviews

Interviews generally consist of: Coding, Algorithms, Architecture or Storage, System Programming, Domain Deep Dive, Database Papers, and Cross-Functional

- **[Storage]** This open-ended design interview evaluates candidates for engineering roles on our storage layer team. The deliberately broad requirements allow you to demonstrate your ability to bring clarity to ambiguous problems, propose and iterate on storage layer designs, and address technical challenges with thoughtful solutions. Be prepared to assess trade-offs, optimizations, and how your design would evolve to meet changing requirements. We value both technical depth and the ability to explain complex concepts in accessible terms.

- **[Stream Processing]** This technical interview evaluates candidates with database and/or distributed systems expertise, focusing on real-time data processing architectures. You'll be asked to design solutions for data ingestion pipelines for real-time systems, storage optimization for high-throughput environments, and query processing mechanisms with low-latency requirements. Be prepared to discuss trade-offs, optimizations, scalability considerations, and fault tolerance. We'll evaluate how your design addresses both immediate needs and future growth, as well as your ability to explain complex concepts in accessible terms. We're looking for systematic thinking paired with practical implementation knowledge.
- **[Databases Papers]** This seminar-style interview will evaluate your database expertise, focusing on query optimization and execution strategies. This open-book assessment allows you to reference your notes and the original paper during the discussion. Be prepared to address questions about storage optimization techniques, runtime performance improvements, and general database performance enhancement methods. This module is specifically designed for engineers with database experience to demonstrate their understanding of advanced optimization concepts.
- **[Cross-Functional/Hiring Manager]** This behavioral interview will be conducted by one of our Engineering Leaders, potentially your direct manager. The conversation will explore your career trajectory, job search motivations, and interest in Databricks. You'll discuss significant projects you've contributed to and how they aligned with broader company objectives, along with your approach to problem-solving, decision-making, and professional growth. We're also interested in your experiences with cross-team collaboration, overcoming challenges, and leadership qualities relevant to your career level.

While this isn't a technical assessment, please be prepared to explain your projects with technical depth, avoiding company-specific terminology that might not be familiar to those outside your current organization.

Engineering Managers

Engineering Leadership interviews

Interviews generally consist of: Hiring and People Management, Process and Execution, Domain Deep-Dive, Architecture, and/or System Programming

- **[Hiring and People Management]** This interview uncovers your talent management expertise across the complete lifecycle — from recruiting engineers and leaders to developing established team members. We'll examine your hiring approach from résumé screening to candidate closing, including your track record, evaluation methods, onboarding practices, and key recruitment insights. We'll also delve into your team development strategies, covering performance management, career growth planning, interpersonal conflict resolution, retention tactics and how you navigate challenging personnel situations while fostering a healthy team culture. Through these conversations, we aim to understand your comprehensive approach to building and sustaining high-performing technical teams.
- **[Process and Execution]** This interview evaluates your approach to leading teams and driving results. You'll share strategies for handling complex projects, addressing execution challenges, and implementing effective planning and prioritization frameworks. We'll inquire about your holistic experience in process management, including setting clear team vision, establishing planning methods and OKRs, implementing appropriate tools and workflows, fostering collaborative environments, and navigating delivery challenges. The discussion may also cover specific scenarios, such as how you've handled project cancellations or pivots.

Additional Resources

- [Data Lakehouse](#), [MLflow](#), [Photon](#), [Apache Spark™](#), [GenAI/MosaicML](#)
- [The Data Lakehouse Platform for Dummies](#)