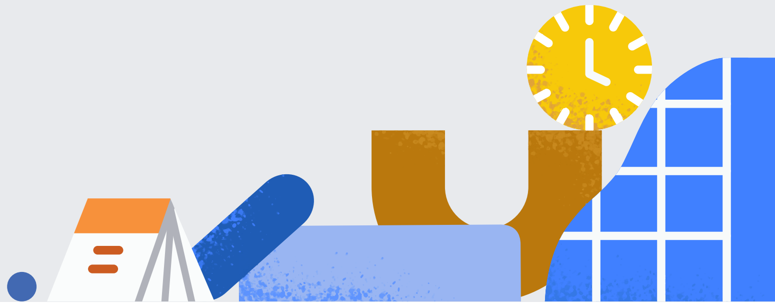
FACEBOOK

MACHINE LEARNING

Onsite Interview Guide



What You'll Find in This Guide

Welcome to your prep guide for your machine learning (ML) onsite interview at the Facebook company. Our ML software engineers and recruiters put together this guide so you know what to expect and how to prepare.

Interview overview

What will your onsite interview be like?

Know that this upcoming interview will be quite different from your initial: you'll be onsite with us and the screen will be longer, nonstop and generally more challenging. Prepare to spend a total of four and a half hours on the Facebook campus for up to six 45-minute interviews. You'll also have a break for lunch, which is unweighted and an opportunity for you to ask questions and relax. During your interviews, you might have an interview with an additional person in the room. Know that they are just there to audit the interview, and that your focus is with the interviewing engineer.

Here's a high-level view of what to expect.

Content

You can expect to cover the following areas in your interviews: at least two coding sessions, one design / architecture session focusing on systems design, one design / architecture session focusing on ML, and a career interview that will focus on you, your work, motivation, and coding skills.

Agenda

Your conversations with our engineers will be divided into the following time blocks:

- **Two 45-minute** Coding Rounds.
- **One 45-minute** Machine Learning Practical Design.
- **One 45-minute** Distributed Systems Design.
- **One 45-minute** Lunch with an engineer.
- **One 45-minute** Career Interview.

Often, you will find a 6th interview, in any of the four competencies above, to better calibrate you and the interview process.

Tips to Keep in Mind

A common mistake is to look at interview problems, recognize them, and sort of understand them, but not to be able to actually code them. So, you should prepare by practicing writing code by hand, without a computer. If you prepare by solving interview-style questions in a timed way, you'll increase the chance that your coding ability will come through during the interview process. Good luck!

Coding rounds

What can you expect?

These coding interviews will be similar to your initial interviews. The engineers will be looking for accurate, bug-free, fast, and well-thought-out code. They'll want to hear your thought process throughout, so be sure to provide a narrative as you go through the code. As in your initial tech screen, you're welcome to code in whatever language you feel most comfortable, but choosing one that is going to assist in getting an optimal solution in the most speedy and efficient manner is key.

Not all interviewers follow the exact same breakdown, but the following is typical:

- **Introductions:** The first five minutes will be an introduction, possibly with brief questions about your background.
- **Coding:** The next 30 – 35 minutes will be one or more coding problems.
- **Ask Us Anything:** We try to reserve the final five minutes for your questions for the interviewer. This part gives you a chance to learn more about Facebook from someone in Engineering and gives your interviewer a chance to learn more about your interests.

How to prep

You should be able to whiteboard solutions from simple to medium difficulty, programming interview questions in under 15 minutes. To prepare, it's not enough to read through sample questions and recognize the concepts. You'll need to practice writing code without a computer, simulating a timed interview environment. When you have a solution, review it and confirm it's something that you'd approve if it were submitted to you as a proposed part of your codebase. Make sure that it's correct, that you've taken into account the edge cases, it's efficient, and it clearly reflects the ideas that you're trying to express in your code.

In addition to reviewing the CS fundamentals, these tips may be helpful:

- **Understand the problem** you have to solve. It's OK to ask for clarifications or to talk through the problem.
- **Think about different algorithms and algorithmic** techniques (sorting, divide-and-conquer, dynamic programming / memorization, recursion).
- **Think about data structures**, particularly the ones used most often (Array, Stack / Queue, Hashset / Hashmap / Hashtable / Dictionary, Tree / Binary Tree, Heap, Graph, Bloom Filter, etc.).
- **Modifying the problem** or thinking about it in smaller pieces may be helpful.
- **Practice coding on a whiteboard.**

Machine learning practical design round

What to expect during this interview

The ML practical design interview is 45 minutes and focuses on your ability to build ML systems at Facebook scale. A strong performance in this interview indicates that you'd be successful in most applied ML problems here at Facebook.

What we ask

- Design a personalized news ranking system.
- Design a product recommendation system.
- Design an evaluation framework for ads ranking.

The idea is to pick any product feature and understand how to model it using ML. We're not looking for you to know and memorize every ML algorithm out there. You should be able to use your existing toolsets to model the problem and break down the components involved in building a large-scale ML system.

What we're looking for

- Can you visualize the entire problem and solution space?
- Can you come up with relevant ML features for your model?
- Can you detect flaws in ML systems and suggest improvements?
- Can you design consistent evaluation and deployment techniques?
- Do you understand architecture requirements (storage, perf etc.) of your system?
- Can you model product requirements into your ML system?
- Are you able to overcome common ML problems such as overfitting, cold start, data collection and logging?

Remember that a good design will touch on the following different components:

- Problem formulation:
  - Optimization function.
  - Supervision signal.
- Feature engineering:
  - Data source.
  - Representation.
- Model architecture.
- Evaluation metrics.
- Deployment (A / B testing).

Tips to Keep in Mind

- If you've mentioned any specific modeling techniques or projects in your resume, be prepared to discuss the class of algorithms that they belong to. Spend some time re-thinking your design decisions, how you might have taken a different approach, and what other insights you have gained from your experiences. You've got this!
- Identify your focus. There's not enough time to discuss every detail of the design, so find the interesting and hard problems to focus the discussion on.

How to prep

To practice, take any well-known app and pick a system that could benefit from ML. For instance, how would you design a friend recommendation system for Twitter? Consider that originally the system was implemented using a handful of static rules for a small set of people. Now, consider you want to deprecate those rules and want to take advantage of ML, so you can easily extend that functionality to millions of people. Then:

- Brush up on basic ML theory and be comfortable with concepts like overfitting and regularization.
- Practice the ability to convert intuitive ideas to concrete features. For example, "number of likes" can be a good feature suggestion but a better feature might also involve normalization, smoothing, and bucketing.
- Think about the end-to-end problem. What will you do after you train the model and the model doesn't perform well? How do you go about debugging an ML model? How do you evaluate and continuously deploy an ML model?
- Consider the questions you'd like to ask the interviewer about the problem. What type of information will help you make better design decisions or to identify the topic you'd like to tackle?
- Analyze your approach and point out pros / cons. Having a good toolset of several different algorithms and understanding the tradeoffs is helpful. For example, be able to explain advantages of logistic regression compared to SVM.

Work out the above problems on paper or a whiteboard and just think about the ways to break them down. Watch public videos and learn how Google search ranking works.

Distributed systems design round

What to expect during this interview

What we ask

- Design a key-value store.
- Design a search for an internet search engine.
- Architect a world-wide video distribution system.
- Build Facebook Chat.

We may focus on specific types of systems (search, ads, distributed learning systems) if you claim expertise in there. We may be more product-focused if that's what you've been working on. We don't expect you to know complex algorithms that you likely wouldn't know off the top of your head (like quad trees or Paxos).

What we're looking for

We're looking for communication. You'll be steering the conversation and it'll be up to you to understand the problem and ask clarifying questions. Beyond that, we're not looking for you to be an expert in ALL of these, but you should know enough of them to weigh design considerations and know when to consult an expert:

- **Familiarity with complex systems.**
- **Concurrency** (threads, deadlock, starvation, consistency, coherence).
- **Abstraction** (understanding how OS, filesystem, and database works).
- **Real-world performance** (relative performance RAM, disk, your network, SSD).
- **Availability and reliability** (durability, understanding how things can fail).
- **Data storage** (RAM vs. durable storage, compression, byte sizes).
- **Byte math**.

Assessment criteria

In addition to the topics above, your interviewer will be trying to get a sense for how you think. We want to see that you can:

- Arrive at an answer in the face of constraints.
- Visualize the entire problem and solution space.
- Make tradeoffs like consistency, availability, partitioning, and performance.
- Give ballpark numbers on QPS supported, number of machines needed using a modern computer.
- Consider Facebook and some of the unique challenges we face.
- Propose a design for a system that breaks down the problem into components that can be built independently, with the ability to talk in detail about any piece of the design.
- Identify the bottlenecks as the system scales and understand the limitations in your design.
- Understand how to adapt the solution when requirements change.
- Draw diagrams that clearly describe the relationship between the different components in the system.
- Calculate (back-of-the-envelope) the physical resources necessary to make this system work.

How to prep

Take any well-known app and think about how you could improve on its design. Reflect on and review the complex systems you've already designed. What would you change about your approach if you did it all over again? What worked well?

- Think about how you'd design a system that Facebook (or Twitter, Google, Uber, Dropbox, etc.) already has. It's a good exercise to think through the complicated, high-scale systems that you already use every day. How would you design it from the ground up?
- Read engineering blogs about approaches that work and false starts big companies have had along the way.
- Start with requirements. Your interviewer might ask: "How would you architect the backend for a messaging system?" Obviously, this question is extremely vague. Where do you even start? You could start with some requirements:
  - How many users are we talking about?
  - How many messages sent?
  - How many messages read?
  - What are the latency requirements for sender-to-receiver message delivery?
  - How will messages be stored?
  - What operations does this data store need to support?
  - What operations is it optimized for?
  - How do you push new messages to clients? Do you push at all, or rely on a pull-based model?
- This is a good online tutorial about design questions.

Tips to Keep in Mind

- Familiarize yourself with our 5 Core Values (Be Bold, Focus on Impact, Move Fast, Be Open, and Build Social Value). This is how we work together to make the world more open and connected. We look for people who believe in these values and practice them daily.

Career interview

What can you expect?

You'll meet with our leadership engineers to talk about two subjects: you and coding. They'll ask about you and your history, your resume, and your motivation to help the team assess whether you'll thrive in Facebook's peer-to-peer, minimal-process, unstructured engineering organization. The coding part is a shorter version of the coding interviews earlier in the day. We include a coding question in this interview to supplement and get additional coding signal.

How to prep

- Know yourself: Take the time to review your own resume as your interviewer will almost certainly ask about key events in your work history.
- Motivation and collaboration: Different interviewers will ask different questions, typically seeking two signals in particular:
  - Motivation: What's your motivation? Why do you want to work at Facebook? Why are you working as a software engineer?
  - Collaboration: How do you collaborate with peers? How do you resolve conflicts?

Tips to Keep in Mind

- Be yourself! Be open and honest about your successes and failures.
- Be humble and focus on teamwork, leadership, and mentorship qualities.

- Have concrete examples or anecdotes: Support each question with examples. Some typical behavioral interview questions are:
    - What were some of the best things you've built?
    - What are you proud of?
    - What could you have done better?
    - What were some successful collaborations you've had?
    - Tell me about a time when you advocated for and pushed your own ideas forward despite opposition?
    - How do you deal with conflict?
    - How do you like to give and receive feedback?
    - What kinds of technologies are you most excited about?
    - Why Facebook?

Appendix / resources

Links to exercises, information, and guides to help you prepare

The resources in this guide are specific to the content and activities in your onsite interview. Take a look through the list as you prepare.

Search Engine Engineering

- **High Scalability**

Machine Learning at Facebook

- **Facebook Research: Machine Learning**

About Facebook

- **5 Core Values**
- **Facebook Quarterly Earnings**

Thanks for taking the time to review this guide and good luck in the interview - you'll do great!