**1. Coding Challenges**

These tests usually assess your ability to write code that solves specific problems. They can be in the form of:

* **Algorithm problems**: Sorting, searching, or solving puzzles using algorithms (e.g., dynamic programming, recursion).
* **Data structures**: Understanding and implementing arrays, linked lists, trees, graphs, stacks, and queues.
* **Complexity analysis**: Analyzing time and space complexity of algorithms (Big O notation).

**Where to practice**:

* **LeetCode**: Offers a wide range of coding challenges and interview questions.
* **HackerRank**: Provides coding challenges with solutions and tutorials, often aligned with specific programming languages.
* **CodeSignal**: Another platform offering coding challenges for interview preparation.
* **Codewars**: Focuses on problem-solving through gamified challenges.

**2. System Design Interviews**

In this type of test, you'll be asked to design a system (like a large-scale application) to test your ability to architect solutions, handle scalability, and think through the technical requirements.

**How to prepare**:

* Practice designing systems such as **social media apps**, **e-commerce platforms**, or **chat services**.
* Understand **scalability**, **load balancing**, **database design**, and how to manage data flow across different parts of the system.

**Resources**:

* **System Design Primer** (GitHub repo): A great guide to system design concepts.
* **Educative.io**: Offers a course specifically on system design interview prep.

**3. Technical Interviews (Whiteboard or Live Coding)**

In this type of interview, you might be asked to write code on the spot, either on a whiteboard or in an online code editor, while explaining your thought process.

**Tips**:

* Practice explaining your thought process as you solve problems. Employers want to see how you think and approach problems.
* Focus on **clean code** and try to follow **best practices** (like proper variable naming, error handling, etc.).

**Live coding platforms**:

* **CoderPad**
* **Interviewing.io**
* **Pramp**

**4. Behavioral/Soft Skills Test**

In addition to technical tests, you might also face behavioral questions to assess how well you work in teams, how you handle challenges, and how you communicate. These are typically designed to see if you are a good cultural fit for the company.

**Common questions**:

* Tell me about a time you faced a challenge in a project.
* How do you handle tight deadlines?
* Can you describe a time when you had to learn a new technology quickly?

**5. Technical Knowledge Tests**

These might be written or online tests that assess your knowledge of specific technologies (e.g., **React**, **Node.js**, **databases**, **cloud platforms**).

**Topics you might encounter**:

* **Programming languages** (JavaScript, Python, Java, etc.)
* **Web development frameworks** (React, Angular, Django, Flask)
* **Databases** (SQL vs NoSQL, database design, queries)
* **Version control systems** (Git, GitHub)
* **Testing frameworks** (Jest, Mocha)