Comprehensive Tech Interview Questions

🖥️ Backend Development

Easy

1. What is an **API**? Explain the difference between an API and a web service.
2. What are the most common **HTTP methods** and what do they do? (e.g., GET, POST, PUT, DELETE)
3. Explain the difference between **SQL** and **NoSQL** databases.

Medium

4. What is **middleware** in the context of a web framework like Express or Django? Provide an example.
5. Explain the difference between **Authentication** and **Authorization**. How would you implement token-based authentication (e.g., JWT)?
6. Compare and contrast **REST** and **GraphQL**. When would you choose one over the other?
7. What is the purpose of an **Object-Relational Mapper (ORM)**? What are its pros and cons?

Hard

8. Describe the challenges of a **microservices architecture**. How do you handle inter-service communication?
9. Explain the concept of a **message queue** (like RabbitMQ or Kafka). Why is it useful in a distributed system?
10. How would you design a system to handle millions of concurrent connections? Discuss concepts like **load balancing**, **database scaling** (read replicas, sharding), and **caching strategies**.

🎨 Frontend Development

Easy

1. Explain the **CSS Box Model**.
2. What is the **DOM** (Document Object Model)?
3. What's the difference between == and === in JavaScript?

Medium

4. Explain the concept of **state** in a framework like React or Vue. What is the difference between props and state?
5. How do you make a website **responsive**? Discuss media queries and modern

approaches like flexbox or grid.
6. What is the browser's **Critical Rendering Path**? How can you optimize it for faster page loads?
7. What is an **SPA** (Single Page Application) and how does it differ from a traditional multi-page application?

Hard

8. How does **code splitting** work in a modern frontend application using a bundler like Webpack?
9. Explain common security vulnerabilities like **Cross-Site Scripting (XSS)** and **Cross-Site Request Forgery (CSRF)** and how to prevent them.
10. What is **WebAssembly (Wasm)** and what problem does it solve? When would it be a good choice for a project?

📊 Data Structures & Algorithms (DSA)

Easy

1. What is the difference between an **Array** and a **Linked List**?
2. Explain **Big O notation**. What does O(n) mean?
3. Write a function to check if a string is a **palindrome**.

Medium

4. What is the difference between a **stack** and a **queue**? Implement one of them.
5. Explain how **Binary Search** works and what its time complexity is. What is the main prerequisite for using it?
6. Traverse a binary tree in three different ways: **in-order, pre-order, and post-order**.
7. Explain the concept of **recursion** with a simple example like calculating a factorial.

Hard

8. What is **Dynamic Programming**? Solve the Fibonacci sequence problem using both memoization and tabulation.
9. Explain **Dijkstra's algorithm** for finding the shortest path in a graph.
10. Design an algorithm to find the **Lowest Common Ancestor (LCA)** of two nodes in a binary search tree. What is the complexity?

🗃️ Database Design

Easy

1. What is a **primary key** and a **foreign key**?
2. What is an **index** in a database? Why is it useful?
3. What are the three main types of relationships in a relational database (One-to-One, One-to-Many, Many-to-Many)?

Medium

4. Explain the concept of **database normalization**. What are 1NF, 2NF, and 3NF?
5. What do the **ACID** properties (Atomicity, Consistency, Isolation, Durability) stand for?
6. Explain different types of **SQL joins** (INNER, LEFT, RIGHT, FULL OUTER).
7. What is a **database transaction**?

Hard

8. Explain the **CAP theorem** in the context of distributed databases.
9. Describe different database scaling strategies like **vertical scaling, horizontal scaling (sharding), and read replicas**.
10. How would you diagnose and optimize a **slow-running query**? Discuss tools and techniques like EXPLAIN PLAN.

🚀 DevOps

Easy

1. What is **CI/CD**? What is the difference between Continuous Integration, Continuous Delivery, and Continuous Deployment?
2. What is **Git**? Explain the difference between git pull and git fetch.
3. What is the difference between a **container** (like Docker) and a **virtual machine (VM)**?

Medium

4. What is **Infrastructure as Code (IaC)**? Name a popular tool and explain its purpose.
5. Explain the purpose of a **Dockerfile**. What are some common instructions?
6. What is **container orchestration**? Why is a tool like Kubernetes needed?
7. What is a **pipeline** in the context of CI/CD? Describe the typical stages.

Hard

8. Explain the architecture of **Kubernetes** at a high level (Control Plane vs. Worker Nodes).
9. What is a **service mesh** (like Istio or Linkerd) and what problems does it solve in a microservices architecture?
10. Compare **Blue-Green** and **Canary** deployment strategies. What are the pros and cons of each?

🧠 Machine Learning

Easy

1. What is the difference between **Supervised** and **Unsupervised** learning? Give an example of each.
2. Explain **overfitting** and **underfitting**. How can you prevent them?
3. What is a **feature** in the context of a dataset?

Medium

4. Explain the **Bias-Variance Tradeoff**.
5. What are **Precision** and **Recall**? When would you prioritize one over the other?
6. How does **Gradient Descent** work?
7. Explain the difference between **Classification** and **Regression** problems.

Hard

8. Explain how **backpropagation** works in a neural network.
9. Describe the architecture of a **Transformer model**. What is the "attention" mechanism?
10. What is **MLOps**? Describe a typical MLOps pipeline for deploying and maintaining an ML model in production.

🏛️ System Design

Easy

1. What is a **load balancer** and why is it used?
2. What is the purpose of a **cache**? Where can you introduce caching in a web application stack?
3. Explain the basic **client-server architecture**.

Medium

4. How would you design a **URL shortening service** like TinyURL?
5. What is a **Content Delivery Network (CDN)** and how does it improve performance?
6. Explain the difference between **stateless** and **stateful** architecture. Why is stateless preferred for scalability?
7. What is an **API Gateway**?

Hard

8. Design a large-scale system like the **Twitter/X news feed**. Discuss the tradeoffs you'd make.
9. Discuss strategies for ensuring **high availability** and **fault tolerance** in a distributed system.
10. Explain **data partitioning (sharding)** strategies for a massive database. What are the challenges involved?

🔒 Cybersecurity

Easy

1. What is **phishing**? How can a user spot a phishing attempt?
2. What is the difference between a **virus** and a **worm**?
3. What is a **firewall** and what does it do?

Medium

4. Explain the **CIA Triad** (Confidentiality, Integrity, Availability).
5. What is the difference between **symmetric** and **asymmetric** encryption?
6. Describe what an **SQL Injection (SQLi)** attack is and how to prevent it.
7. What is the principle of **least privilege**?

Hard

8. Explain how a **Man-in-the-Middle (MITM)** attack works.
9. How would you design a system to mitigate **DDoS (Distributed Denial of Service)** attacks?
10. What is a **zero-day vulnerability**? Why is it so dangerous?