ASP.NET Core Web API (20 Questions)

1. What is middleware in ASP.NET Core, and how does the request pipeline work?
2. Difference between filters and middleware?
3. Explain the Dependency Injection (DI) system in ASP.NET Core.
4. How do you implement global exception handling?
5. What are Action Results vs. IActionResult vs. ActionResult?
6. How do you version APIs in ASP.NET Core?
7. How do you secure APIs with JWT authentication?
8. What’s the difference between Authentication and Authorization?
9. How do you enable CORS in ASP.NET Core?
10. How do you improve API performance (caching, compression, response optimization)?
11. Difference between synchronous and asynchronous APIs?
12. How do you configure logging in ASP.NET Core (ILogger, Serilog, NLog)?
13. Explain the Repository and Unit of Work patterns in ASP.NET Core.
14. How do you handle validation (Data Annotations, Fluent Validation)?
15. How do you test ASP.NET Core APIs (unit tests, integration tests)?
16. Explain how routing works (conventional vs. attribute routing).
17. Difference between REST, SOAP, and gRPC APIs?
18. How do you implement rate limiting/throttling in ASP.NET Core?
19. How do you upload and download files via Web API?
20. How would you design a scalable microservice-based API using ASP.NET Core?

Angular (20 Questions)

1. What is the difference between AngularJS and Angular (2+)?
2. Explain Angular architecture (Modules, Components, Services).
3. What are Angular lifecycle hooks?
4. Difference between template-driven and reactive forms?
5. How does Angular’s change detection mechanism work?
6. What is NgZone, and how does Angular detect changes?
7. Explain Dependency Injection (DI) in Angular.
8. How do you communicate between components (Input, Output, services, RxJS)?
9. What are Angular Directives (structural vs. attribute)?
10. Explain RxJS Observables vs. Promises.
11. Difference between switchMap, mergeMap, concatMap, exhaustMap?
12. What are Angular pipes, and how to create a custom pipe?
13. How does Angular routing work? What are route guards?
14. What is Lazy Loading, and why is it used?
15. How do you optimize Angular performance (trackBy, OnPush, AOT)?
16. What is Angular’s HttpClient, and how do interceptors work?
17. Explain Angular modules (feature modules, shared modules).
18. What is NgRx, and when should you use it?
19. How do you handle error handling in Angular (global vs. local)?
20. How do you secure Angular apps (JWT, XSS prevention, CSP)?

SQL (20 Questions)

1. Difference between INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN.
2. What are clustered vs. non-clustered indexes?
3. Explain primary key vs. unique key.
4. What is normalization? Explain 1NF, 2NF, 3NF.
5. What are transactions? What are ACID properties?
6. What are isolation levels in SQL Server?
7. Difference between DELETE, TRUNCATE, and DROP.
8. What are Common Table Expressions (CTE)?
9. What is a window function? Example use cases.
10. Difference between UNION and UNION ALL.
11. How do you optimize slow queries?
12. Difference between HAVING and WHERE.
13. What is a stored procedure? Advantages?
14. What are views? Indexed views vs. normal views.
15. Difference between CHAR, VARCHAR, NVARCHAR.
16. What is a trigger? When would you use it?
17. Explain deadlocks and how to avoid them.
18. What is a covering index?
19. Difference between correlated and non-correlated subquery.
20. How do you design a database for performance and scalability?

SQL – Practical Exercises (10)

1. Query to get the 2nd highest salary in a table.
2. Query to find employees who earn more than their manager.
3. Query to find duplicate rows in a table.
4. Query to delete duplicate rows but keep one.
5. Query to pivot rows into columns.
6. Query to get department-wise average salary using GROUP BY.
7. Query to fetch the first 3 records from each department.
8. Query using ROW_NUMBER() to rank employees by salary.
9. Query to find orders placed in the last 30 days.
10. Query to find customers who bought products in all categories.

Dapper (20 Questions)

1. What is Dapper, and why use it over Entity Framework?
2. How does Dapper differ from ADO.NET?
3. How do you query data with Dapper?
4. How do you map query results to objects in Dapper?
5. What is the difference between Query and QueryFirst/Single?
6. How do you execute stored procedures in Dapper?
7. How do you handle multiple result sets in Dapper?
8. How does Dapper handle parameterized queries?
9. How do you insert, update, and delete records using Dapper?
10. How do you use transactions with Dapper?
11. Difference between Execute and Query methods?
12. How do you map one-to-many relationships in Dapper?
13. How do you map many-to-many relationships?
14. How do you handle bulk inserts in Dapper?
15. What are dynamic parameters in Dapper?
16. How does Dapper support async operations?
17. How do you cache queries in Dapper?
18. What are some performance considerations with Dapper?
19. When would you prefer EF Core over Dapper?
20. How do you integrate Dapper in a clean architecture project?

Entity Framework Core (20 Questions)

1. Difference between EF Core and EF6.
2. What is DbContext?
3. How do you configure EF Core in ASP.NET Core?
4. What are migrations in EF Core?
5. Difference between Database First, Model First, and Code First approaches.
6. What are shadow properties in EF Core?
7. How do you configure relationships (1-1, 1-many, many-many)?
8. Difference between eager loading, lazy loading, and explicit loading.
9. How do you handle concurrency in EF Core?
10. How do you optimize EF Core performance?
11. What are global query filters?
12. Difference between Find() and FirstOrDefault().
13. How do you implement transactions in EF Core?
14. How do you handle raw SQL queries in EF Core?
15. Difference between Value Objects and Entities in EF Core.
16. What is the purpose of the Change Tracker?
17. How does EF Core support LINQ queries?
18. Difference between AsNoTracking() and normal queries.
19. How do you seed data in EF Core?
20. What are some pitfalls when using EF Core in large projects?

JavaScript (20 Questions)

1. Difference between var, let, and const.
2. What is hoisting in JavaScript?
3. What are closures? Give an example.
4. Explain the prototype chain in JavaScript.
5. Difference between == and ===.
6. What is the event loop in JavaScript?
7. Difference between synchronous and asynchronous programming.
8. What are promises? Difference between promise and callback.
9. What is async/await, and how does it work internally?
10. What are higher-order functions?
11. What are arrow functions, and how do they differ from normal functions?
12. Difference between call(), apply(), and bind().
13. What is event delegation?
14. Difference between localStorage, sessionStorage, and cookies.
15. What are modules in JavaScript (ES6 imports/exports)?
16. Difference between shallow copy and deep copy.
17. What is the difference between null and undefined?
18. What is a generator function?
19. How does JavaScript handle memory management?
20. How does debouncing and throttling work in JS?

TypeScript (10 Questions)

1. What is TypeScript, and why use it over JavaScript?
2. What are interfaces in TypeScript?
3. Difference between type and interface.
4. What are generics in TypeScript?
5. What are decorators in TypeScript?
6. How does TypeScript handle null and undefined?
7. What is the difference between unknown and any?
8. What are enums in TypeScript?
9. What is strict mode in TypeScript?
10. How does TypeScript support async/await?

JWT & Security (10 Questions)

1. What is JWT, and why is it used?
2. What are the main parts of a JWT (Header, Payload, Signature)?
3. How does JWT authentication work in ASP.NET Core?
4. Difference between JWT and OAuth2.
5. How do you store JWT securely in a browser? (localStorage vs. cookies).
6. How do you refresh JWT tokens?
7. What are some common JWT vulnerabilities?
8. How do you handle token expiration?
9. What are claims in JWT?
10. What are best practices for securing APIs with JWT?