🔹 Angular Basics

1. What is Angular and how is it different from AngularJS?
2. What are the main features of Angular?
3. What is the purpose of Angular in web development?
4. Why would you choose Angular over other frameworks like React or Vue?
5. Explain the high-level architecture of a web application built using Angular.
6. What is the role of the front end, back end, and database in a web application?

🔹 Framework & Development Platform

7. What is the difference between a library and a framework?
8. What are the benefits of using a framework like Angular?
9. What is a development platform in the context of Angular?

🔹 JavaScript & TypeScript

10. How is TypeScript different from JavaScript?
11. What are the basic data types in TypeScript?
12. What is type inference in TypeScript?
13. Explain the difference between "any", "unknown", "union", and "literal" types.
14. How does TypeScript handle object typing?
15. What are enums and tuples in TypeScript?

16. What are access modifiers in TypeScript OOP?

17. What is the difference between an abstract class and an interface?

🔹 Angular Environment & Setup

18. How do you set up an Angular development environment?

19. What is Angular CLI and how is it useful?

20. How do you create a new Angular project using the CLI?

21. What is the purpose of `package.json` in an Angular project?

🔹 Components

22. What is a component in Angular?

23. How do you create a component using Angular CLI?

24. What are the main parts of a component?

25. How do you share data between components?

26. Explain parent-child communication using `@Input()` and `@Output()`.

🔹 Data Binding

27. What is data binding in Angular?

28. What are the different types of data binding?

29. Explain the difference between property binding and interpolation.

30. How does two-way data binding work in Angular?

Directives

31. What are Angular directives?

32. What is the difference between structural and attribute directives?

33. How does the `ngFor` directive work?

34. When would you use `ngIf` vs `ngSwitch`?

35. How do `ngClass` and `ngStyle` help in dynamic styling?

Decorators

36. What are decorators in Angular?

37. What is the difference between `@ViewChild` and `@ContentChild`?

38. How do you use `ng-template`, `ng-content`, and `ng-container`?

39. How do `@ViewChildren` and `@ContentChildren` differ?

Services and Dependency Injection

40. What is a service in Angular?

41. How do you create and inject a service?

42. Explain dependency injection in Angular.

43. How do services help in component communication?

🔹 Pipes

44. What are Angular pipes?

45. What is the difference between a pure and impure pipe?

46. How do you create a custom pipe?

🔹 HTTP & Backend Integration

47. How do you make HTTP requests in Angular?

48. What is `HttpClient` and how do you use it?

49. How do you handle HTTP errors?

50. How do you add headers to HTTP requests?

51. What is JWT authentication and how is it implemented in Angular?

52. What are observables and how are they used with HTTP requests?

🔹 Observables & RxJS

53. What is the difference between a Promise and an Observable?

54. What is a Subject in RxJS?

55. How does `map()` work in RxJS?

56. How do `of()` and `from()` differ?

57. Explain the use of operators like `filter`, `take`, and `switchMap`.

🔹 Routing

58. What is routing in Angular?

59. How do you configure routes in Angular?

60. What is the purpose of `router-outlet`?

61. How do you implement route guards like `CanActivate` and `CanDeactivate`?

62. What is the difference between `routerLink` and `navigateByUrl`?

🔹 Forms in Angular

63. What is the difference between template-driven and reactive forms?

64. How do you implement form validation in Angular?

65. What are form states like `touched`, `dirty`, and `pristine`?

66. How do you dynamically add form controls in a reactive form?

67. What is a `FormArray` and how is it used?
