Beginner / L1 – Angular Fundamentals

1. What is Angular?
2. Difference between AngularJS and Angular?
3. What are the building blocks of Angular?
4. What is a component?
5. What is a module (NgModule)?
6. What is a template?
7. What is data binding in Angular?
8. Explain one-way vs two-way binding.
9. What is ngModel?
10. What are directives?
11. Difference between structural and attribute directives?
12. What is *ngIf?
13. What is *ngFor?
14. What is interpolation?
15. What is event binding?
16. What is property binding?
17. What is a pipe?
18. Name some built-in Angular pipes.
19. What is the Angular CLI?
20. How do you create a component using CLI?

Intermediate / L2 – Components, Services, Forms

21. What are lifecycle hooks in Angular?
22. Explain ngOnInit.
23. Difference between constructor and ngOnInit?
24. What is dependency injection?
25. How does Angular DI work?
26. What is a service?
27. Why do we use services?
28. How do you share data between components?

29. What are template-driven forms?
30. What are reactive forms?
31. Difference between template-driven and reactive forms?
32. What is FormBuilder?
33. What are validators?
34. How do you create a custom validator?
35. What is form control?
36. What is form group?
37. What is form array?
38. What is @Input()?
39. What is @Output()?
40. What is EventEmitter?

Advanced / L3 – Routing, RxJS, Performance

41. What is Angular Router?
42. What is lazy loading?
43. How do you implement lazy loading?
44. What are route guards?
45. Types of route guards in Angular?
46. What is an observable?
47. Difference between Observable and Promise?
48. What is RxJS?
49. What is a Subject?
50. Difference between Subject and BehaviorSubject?
51. What is async pipe?
52. What is HttpClient?
53. How do you handle HTTP errors?
54. What is an HTTP interceptor?
55. Use cases of interceptors?
56. What is ChangeDetectionStrategy.OnPush?
57. How does Angular change detection work?

58. What is Zone.js?
59. What is trackBy in *ngFor?
60. How do you optimize Angular application performance?

Expert – Architecture, Security, Testing

61. What is Ahead-of-Time (AOT) compilation?
62. Difference between AOT and JIT?
63. What is Ivy renderer?
64. What are standalone components?
65. What is Angular Signals?
66. How does Angular handle state management?
67. Difference between NgRx and services?
68. How do you secure Angular applications?
69. What is XSS and how does Angular prevent it?
70. How do you handle authentication and authorization?
71. What is role-based access control (RBAC)?
72. How do you write unit tests in Angular?
73. Difference between Jasmine and Jest?
74. What is TestBed?
75. How do you test services?
76. How do you mock HTTP calls?
77. What is end-to-end testing?
78. What is Cypress?
79. How do you deploy an Angular application?
80. What are environment files?

Scenario-Based / Real-World Questions

81. How would you design a large Angular application?
82. How do you handle memory leaks in Angular?
83. How would you implement global error handling?
84. How do you improve load time in Angular?

85. How do you manage API retries?
86. How do you implement caching?
87. How do you share state across modules?
88. How do you handle version upgrades?
89. How do you migrate to standalone components?
90. How do you handle large forms efficiently?

Tricky / Conceptual Questions

91. Why should you unsubscribe from Observables?
92. When should you not unsubscribe?
93. What is providedIn: 'root'?
94. What is a pure vs impure pipe?
95. What is view encapsulation?
96. Types of view encapsulation?
97. What is ChangeDetectorRef?
98. What is NgZone?
99. What is dynamic component loading?
100. When would you choose Angular over React?