

BOLUM 1: Temel Kavramlar ve Prensipler (Basic)

1. Soru: Yazılım Mimarisi nedir ve neden önemlidir?

Cevap: Yazılım mimarisi, bir sistemin organizasyonunu, bileşenlerini, bu bileşenlerin birbirleriyle ilişkilerini ve tasarımı yöneten prensipleri tanımlayan üst düzey yapıdır. Önemi, sistemin ölçeklenebilirliği, bakımı, performansı ve güvenliği gibi kalite özelliklerini garanti altına almasından gelir.

2. Soru: SOLID prensipleri nelerdir?

Cevap: Nesne yönelimli tasarımın 5 temel prensibidir: Single Responsibility (Tek Sorumluluk), Open/Closed (Açık/Kapalı), Liskov Substitution (Liskov Yerine Gecme), Interface Segregation (Arayüz Ayrımı) ve Dependency Inversion (Bağımlılıkların Tersine Çevrilmesi).

3. Soru: Coupling ve Cohesion arasındaki fark nedir?

Cevap: Cohesion, bir modülün kendi içindeki elemanların birbirine ne kadar sıkı bağlı olduğunu ifade eder (Yüksek olması istenir). Coupling ise modüllerin birbirine ne kadar bağımlı olduğunu ifade eder (Düşük olması istenir).

4. Soru: DRY (Don't Repeat Yourself) prensibi nedir?

Cevap: Bilginin veya mantığın tekrarlanmaması gerektiğini savunur. Aynı kodun kopyalanıp yapıştırılması yerine, fonksiyonlar veya sınıflar aracılığıyla soyutlanmasını önerir.

5. Soru: KISS prensibi mimaride nasıl uygulanır?

Cevap: Sistemlerin gereksiz karmaşıklıktan arındırılması gerektiğini savunur. Bir çözüm çalışıyorsa ve basitse, daha karmaşık bir çözüme tercih edilmelidir.

6. Soru: YAGNI ne demektir?

Cevap: Buna ihtiyacın olmayacak anlamına gelir. Gelecekte lazım olur diye simdiden özellik eklemeyin, sadece mevcut gereksinimleri karşılayın demektir.

7. Soru: Teknik Borç nedir?

Cevap: Hızlı teslimat yapmak için kısa vadeli, kalitesiz çözümlerin seçilmesiyle oluşan ekstra geliştirme maliyetidir.

8. Soru: Scalability nedir?

Cevap: Bir sistemin artan iş yükünü, kaynak ekleyerek performans kaybı yaşamadan karşılayabilme yeteneğidir.

9. Soru: Vertical ve Horizontal Scaling farkı nedir?

Cevap: Dikey ölçekleme, mevcut sunucunun gücünü artırmaktır. Yatay ölçekleme ise sisteme yeni sunucular eklemektir.

10. Soru: Stateless ve Stateful uygulama farkı nedir?

Cevap: Stateless uygulamalar sunucu tarafında oturum verisini tutmaz. Stateful uygulamalar ise oturum bilgisini sunucuda tutar.

BOLUM 2: Tasarim Desenleri (Design Patterns)

11. Soru: Singleton Pattern nedir?

Cevap: Bir sinifin sadece bir orneginin olusturulmasini garanti eden desendir. Veritabani baglanti havuzlari icin kullanilir.

12. Soru: Factory Pattern ne ise yarar?

Cevap: Nesne olusturma mantigini istemciden gizler. Istemci hangi sinifin orneklenecegini bilmeden nesne talep eder.

13. Soru: Strategy Pattern hangi sorunu cozer?

Cevap: Bir islemin birden fazla algoritmayla yapilabilecegi durumlarda, calisma zamaninda algoritmayi degistirmeyi saglar.

14. Soru: Observer Pattern nedir?

Cevap: Bir nesnedeki degisikliklerin, ona bagimli diger nesnelere otomatik olarak bildirilmesini saglar.

15. Soru: Adapter Pattern ne zaman kullanilir?

Cevap: Uyumsuz arayuzlere sahip siniflari birlikte calismasini saglamak icin kullanilir.

16. Soru: Decorator Pattern nedir?

Cevap: Bir nesneye dinamik olarak yeni ozellikler eklemek icin kullanilir.

17. Soru: Proxy Pattern'in kullanim amaci nedir?

Cevap: Bir nesneye erisimi kontrol etmek icin vekil bir nesne kullanilmasidir.

18. Soru: Dependency Injection neden onemlidir?

Cevap: Siniflari bagimliklerini kendilerinin olusturmasi yerine disaridan almasini saglar. Test edilebilirligi artırir.

BOLUM 3: Mimari Stiller

19. Soru: Monolitik Mimari nedir?

Cevap: Tum uygulamanin tek bir kod tabani ve tek bir deploy edilebilir birim olarak gelistirilmesidir.

20. Soru: Mikroservis Mimarisi nedir?

Cevap: Uygulamanin kucuk, bagimsiz, kendi veritabanina sahip servisler butunu olarak tasarlanmasidir.

21. Soru: Layered Mimari nedir?

Cevap: Yazilimin sunum, is mantigi, veri erisimi gibi yatay katmanlara ayrilmasidir.

22. Soru: Hexagonal Mimari nedir?

Cevap: Is mantigini dis dunyadan izole etmeyi amaclar. Bagimliliklar disaridan iceriye dogrudur.

23. Soru: Serverless Mimari ne demektir?

Cevap: Sunucu yonetimiyle ugrasmadan, sadece kodun calistirildigi bulut modelidir.

24. Soru: Event-Driven Architecture nedir?

Cevap: Bilesenlerin durum degisikliklerini olay olarak yayinladigi asenkron mimaridir.

25. Soru: SOA ile Mikroservis farki nedir?

Cevap: SOA daha kurumsal, ESB kullanan yapidir. Mikroservisler daha granuler ve merkeziyetsizdir.

BOLUM 4: Veri Yonetimi ve Veritabanlari

26. Soru: SQL ve NoSQL veritabani ne zaman secilmelidir?

Cevap: Veri yapisi sabit ve tutarlilik kritikse SQL; veri semasi esnek ve yazma hizi kritikse NoSQL secilir.

27. Soru: ACID ozellikleri nelerdir?

Cevap: Atomicity, Consistency, Isolation, Durability.

28. Soru: CAP Teoremi nedir?

Cevap: Dagitik bir sistemde Consistency, Availability ve Partition Tolerance ozelliklerinden ayni anda sadece ikisinin saglanabilecegini belirtir.

29. Soru: Database Sharding nedir?

Cevap: Buyuk bir veritabanini daha kucuk parcalara bolerek farkli sunucularda tutmaktir.

30. Soru: Replication neden yapilir?

Cevap: Verinin kopyasini baska sunucularda tutarak okuma performansini artirmak ve yedek saglamak icin.

31. Soru: Caching stratejileri nelerdir?

Cevap: Write-Through, Write-Back, Cache-Aside.

32. Soru: Eventual Consistency nedir?

Cevap: Dagitik sistemlerde verinin tum dugumlerde bir sure sonra tutarli hale gelecegini kabul eden yaklasimdir.

BOLUM 5: İletişim ve API

33. Soru: REST ile GraphQL arasındaki fark nedir?

Cevap: REST kaynak odaklıdır ve sabit veri döner. GraphQL ise istemcinin tam olarak istediği veriyi sorgulayabildiği bir sorgulama dilidir.

34. Soru: gRPC nedir?

Cevap: Google tarafından geliştirilen, Protocol Buffers kullanan, yüksek performanslı bir RPC framework'üdür.

35. Soru: Senkron ve Asenkron iletişim farkı nedir?

Cevap: Senkron iletişimde istemci cevabı bekler. Asenkron iletişimde istemci istegi atar ve işine devam eder.

36. Soru: Idempotency nedir?

Cevap: Bir işlemin birden fazla kez uygulanmasının sonucu değiştirmemesidir.

37. Soru: API Gateway'in görevleri nelerdir?

Cevap: Yönlendirme, kimlik doğrulama, rate limiting ve SSL sonlandırma yapar.

BOLUM 6: Guvenlik

38. Soru: Authentication ve Authorization farki nedir?

Cevap: Authentication 'Sen kimsin?' sorusuna cevap verir. Authorization 'Neleri yapmaya iznin var?' sorusuna cevap verir.

39. Soru: JWT nasıl çalışır?

Cevap: Kullanici login oldugunda sunucu imzali bir token uredir. Istemci sonraki isteklerde bu token'i gonderir.

40. Soru: OAuth2 nedir?

Cevap: Bir kullanicinin sifresini paylasmadan, uygulamalarinin verilerine baska uygulamalarin erismesine izin veren standardir.

41. Soru: SQL Injection nedir?

Cevap: Kullanici girdisi uzerinden veritabani sorgusunu manipule etme saldirisidir. Parametreli sorgular ile onlenir.

42. Soru: CORS nedir?

Cevap: Tarayicilarin, bir alan adindan sunulan uygulamanin, farkli alan adindaki kaynaklara erismesini kisitlayan mekanizmadir.

BOLUM 7: Ileri Duzey & Resilience

43. Soru: Circuit Breaker Pattern nedir?

Cevap: Bir servis hata verdiginde, surekli istek atip sistemi yormak yerine sigortayi attirip hemen hata donmeyi saglayan desendir.

44. Soru: Bulkhead Pattern nedir?

Cevap: Sistem kaynaklarini izole ederek bir parcadaki hatanin tum sistemi cokertmesini engellemektir.

45. Soru: Distributed Tracing neden gereklidir?

Cevap: Mikroservislerde bir istegin hangi servislerden gectigini, nerede yavasledigini takip etmek icin.

46. Soru: Saga Pattern nedir?

Cevap: Dagitik sistemlerde, birden fazla servise yayilan islemleri yönetmek icin kullanilir.

47. Soru: Infrastructure as Code nedir?

Cevap: Sunucu ve ag altyapisinin kod dosyalarıyla yönetilmesidir.

48. Soru: Blue-Green Deployment nedir?

Cevap: Kesinti olmadan surum gecisi yapmak icin iki ozdes ortam kullanmaktır.

49. Soru: Canary Release nedir?

Cevap: Yeni versiyonun once kucuk bir yuzdeye acilmasi, sorun yoksa herkese yayilmasidir.

50. Soru: Sidecar Pattern nedir?

Cevap: Ana uygulamanin yanina yardimci bir surec ekleyerek loglama, izleme gibi islerin soyutlanmasidir.