

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 tasarımını yöneten prensipleri tanımlayan üst düzey yapıdır. Önemi, sistemin ölçülebilirliği, bakımı, performansı ve güvenliği gibi kalite özniteliklerini 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 Ayrimi) 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 soyutlanması önerir.

5. Soru: KISS prensibi mimaride nasıl uygulanır?

Cevap: Sistemlerin gereksiz karmaşıklikten arındırılması gerektiğini savunur. Bir çözüm çalışıysa ve basitse, daha karmaşık bir çözümü tercih edilmelidir.

6. Soru: YAGNI ne demektir?

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

7. Soru: Teknik Borc 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 olceklemeye, mevcut sunucunun gücünü artırmaktır. Yatay olceklemeye 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: Tasarım Desenleri (Design Patterns)

11. Soru: Singleton Pattern nedir?

Cevap: Bir sınıfın sadece bir örnekinin oluşturulmasını garanti eden desendir. Veritabanı bağlantı havuzları için kullanılır.

12. Soru: Factory Pattern ne ise yarar?

Cevap: Nesne oluşturma mantığını istemciden gizler. İstemci hangi sınıfın örnekleneceğini bilmeden nesne talep eder.

13. Soru: Strategy Pattern hangi sorunu cozer?

Cevap: Bir işlemin birden fazla algoritmayla yapılabileceği durumlarda, çalışma zamanında algoritmayı değiştirmeyi sağlar.

14. Soru: Observer Pattern nedir?

Cevap: Bir nesnedeki değişikliklerin, ona bağlı diğer nesnelere otomatik olarak bildirilmesini sağlar.

15. Soru: Adapter Pattern ne zaman kullanılır?

Cevap: Uyumsuz arayzlara sahip sınıfların birlikte çalışmasını sağlamak için kullanılır.

16. Soru: Decorator Pattern nedir?

Cevap: Bir nesneye dinamik olarak yeni özellikler eklemek için kullanılır.

17. Soru: Proxy Pattern'in kullanım amacı nedir?

Cevap: Bir nesneye erişimi kontrol etmek için vekil bir nesne kullanılmasıdır.

18. Soru: Dependency Injection neden önemlidir?

Cevap: Sınıfların bağımlıklarını kendilerinin oluşturması yerine dışardan almasını sağlar. Test edilebilirliği artırır.

BOLUM 3: Mimari Stiller

19. Soru: Monolitik Mimari nedir?

Cevap: Tüm uygulamanın tek bir kod tabanı ve tek bir deploy edilebilir birim olarak geliştirilmesidir.

20. Soru: Mikroservis Mimarisi nedir?

Cevap: Uygulamanın küçük, bağımsız, kendi veritabanına sahip servisler bütünü olarak tasarılanmasıdır.

21. Soru: Layered Mimari nedir?

Cevap: Yazılımin sunum, iş mantığı, veri erişimi gibi yatay katmanlara ayrılmasıdır.

22. Soru: Hexagonal Mimari nedir?

Cevap: İş mantığını dış dunyadan izole etmeyi amaçlar. Bağımlılıklar dışarıdan içeriye doğrudur.

23. Soru: Serverless Mimari ne demektir?

Cevap: Sunucu yönetimiyle ugrasmadan, sadece kodun çalıştırıldığı bulut modelidir.

24. Soru: Event-Driven Architecture nedir?

Cevap: Bileşenlerin durum değişikliklerini olay olarak yayınladığı asenkron mimaridir.

25. Soru: SOA ile Mikroservis farkı nedir?

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

BOLUM 4: Veri Yonetimi ve Veritabanları

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

Cevap: Veri yapısı sabit ve tutarlılık kritikse SQL; veri semasi esnek ve yazma hızı kritikse NoSQL seçilir.

27. Soru: ACID özellikleri nelerdir?

Cevap: Atomicity, Consistency, Isolation, Durability.

28. Soru: CAP Teoremi nedir?

Cevap: Dagitik bir sistemde Consistency, Availability ve Partition Tolerance özelliklerinden aynı anda sadece ikisinin sağlanabileceğini belirtir.

29. Soru: Database Sharding nedir?

Cevap: Buyuk bir veritabanını daha kucuk parçalara bolerek farklı sunucularda tutmaktadır.

30. Soru: Replication neden yapılır?

Cevap: Verinin kopyasını başka sunucularda tutarak okuma performansını artırmak ve yedek sağlamak için.

31. Soru: Caching stratejileri nelerdir?

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

32. Soru: Eventual Consistency nedir?

Cevap: Dagitik sistemlerde verinin tüm düğümlerde bir süre sonra tutarlı hale geleceğini kabul eden yaklaşımdır.

BOLUM 5: Iletisim ve API

33. Soru: REST ile GraphQL arasındaki fark nedir?

Cevap: REST kaynak odaklıdır ve sabit veri doner. 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'udur.

35. Soru: Senkron ve Asenkron iletişim farkı nedir?

Cevap: Senkron iletişimde istemci cevabı bekler. Asenkron iletişimde istemci isteği 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: Yonlendirme, kimlik doğrulama, rate limiting ve SSL sonlandırma yapar.

BOLUM 6: Guvenlik

38. Soru: Authentication ve Authorization farkı nedir?

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

39. Soru: JWT nasıl çalışır?

Cevap: Kullanıcı login olduğunda sunucu imzalı bir token üretir. İstemci sonraki isteklerde bu token'i gönderir.

40. Soru: OAuth2 nedir?

Cevap: Bir kullanıcının şifresini paylaşmadan, uygulamalarının verilerine başka uygulamaların erişmesine izin veren standartdır.

41. Soru: SQL Injection nedir?

Cevap: Kullanıcı girdisi üzerinden veritabanı sorgusunu manipule etme saldırısıdır. Parametreli sorgular ile onlenir.

42. Soru: CORS nedir?

Cevap: Tarayıcıların, bir alan adından sunulan uygulamanın, farklı alan adındaki kaynaklara erişmesini kısıtlayan mekanizmadır.

BOLUM 7: İleri Duzey & Resilience

43. Soru: Circuit Breaker Pattern nedir?

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

44. Soru: Bulkhead Pattern nedir?

Cevap: Sistem kaynaklarını izole ederek bir parcadaki hatanın tüm sistemi çökertmesini engellemektir.

45. Soru: Distributed Tracing neden gereklidir?

Cevap: Mikroservislerde bir isteğin hangi servislerden geçtiğini, nerede yavaşladığını takip etmek için.

46. Soru: Saga Pattern nedir?

Cevap: Dagitik sistemlerde, birden fazla servise yayılan işlemleri yönetmek için kullanılır.

47. Soru: Infrastructure as Code nedir?

Cevap: Sunucu ve ağ altyapısının kod dosyalarıyla yönetilmesidir.

48. Soru: Blue-Green Deployment nedir?

Cevap: Kesinti olmadan surum geçisi yapmak için iki özdes ortam kullanmaktadır.

49. Soru: Canary Release nedir?

Cevap: Yeni versiyonun önce küçük bir yüzdeye açılması, sorun yoksa herkese yayılmasıdır.

50. Soru: Sidecar Pattern nedir?

Cevap: Ana uygulamanın yanına yardımcı bir süreç ekleyerek loglama, izleme gibi işlerin soyutlanmasıdır.