**Görev 1. Sunucu Mimarisi Tasarımı**

**1. İstemci Tarafı**

Web İstemcileri: Web uygulamasına erişen tarayıcılar.

Mobil İstemciler: Platforma RESTful API'leri aracılığıyla erişen mobil uygulamalar.

**2. Yük Dengeleyiciler**

Rol: Hiçbir sunucunun aşırı yük altında kalmamasını sağlamak için gelen ağ trafiğini birden fazla sunucuya dağıtır.

Sorumluluk: İş yükü dağıtımını iyileştirin ve uygulamanın kullanılabilirliğini ve güvenilirliğini artırır.

**3. Web Sunucuları**

Rol: İstemcilerden gelen HTTP isteklerini yönetir (örneğin, statik içerik sunmak, oturumları yönetmek).

Sorumluluk: Dinamik içerik isteklerini uygulama sunucularına ileterek kullanıcı etkileşimlerini işler ve yanıtlar.

**4. Uygulama Sunucuları**

Rol: Uygulamanın temel mantığını yürütür, iş mantığını yönetir ve kullanıcı oturumlarını yönetir.

Sorumluluk: Kullanıcı eylemlerini işlemek ve dinamik içerik oluşturmak için veritabanı sunucuları, önbellek sunucuları ve diğer mikro hizmetlerle etkileşimde bulunur.

**5. Veritabanı Sunucuları**

Rol: Verileri depolar ve alır.

Sorumluluk: Kullanıcı verilerini, içerik verilerini ve diğer kalıcı verileri korur. Ölçeklenebilirlik ve yüksek kullanılabilirlik için çoğaltma ve parçalamayı kullanır.

Bileşenler:

Birincil Veritabanı: Yazma işlemlerini gerçekleştirir ve replika veritabanlarıyla senkronizasyon sağlar.

Çoğaltma Veritabanları: Birincil veritabanının yükünü boşaltmak için okuma işlemlerini gerçekleştirir.

**6. Önbellek Sunucuları**

Rol: Sık erişilen verileri geçici olarak depolar.

Sorumluluk: Ortak sorguları ve kullanıcı oturumlarını önbelleğe alarak veritabanı sunucularındaki yükü azaltır.

Bileşenler:

Redis/Memcached: Önbelleğe alma için kullanılan bellek içi veri yapısı depoları.

**7. İçerik Dağıtım Ağı (CDN)**

Rol: Statik içeriği (resimler, videolar, komut dosyaları) en yakın coğrafi konumlardaki kullanıcılara dağıtır.

Sorumluluk: İçerik yükleme sürelerini iyileştirir ve web sunucularındaki yükü azaltır.

**8. Mikro hizmetler**

Rol: Uygulama içindeki belirli işlevleri yönetir (ör. kimlik doğrulama, bildirimler, mesajlaşma).

Sorumluluk: Modülerliği, ölçeklenebilirliği ve bakım kolaylığını geliştirir.

**9. Mesaj Kuyrukları**

Rol: Mikro hizmetler arasındaki eşzamansız iletişimi yönetir.

Sorumluluk: Ölçeklenebilirliği artırmak için güvenilir mesaj iletimi ve ayrıştırma hizmetleri sağlar.

**10. İzleme ve Günlük Kaydı**

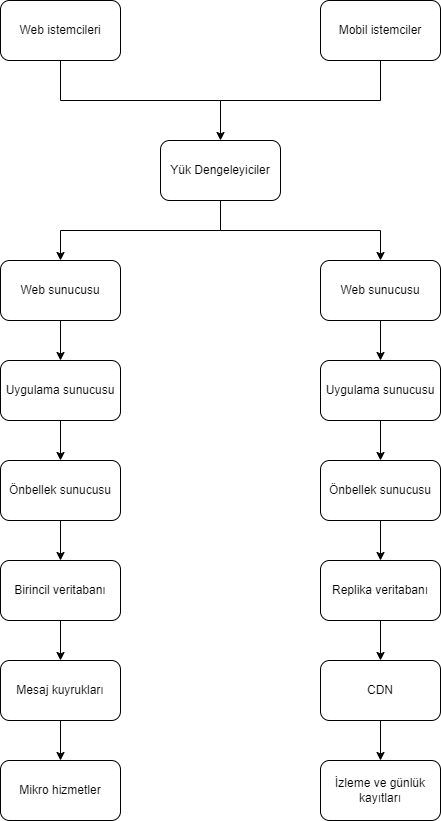
Rol: Sistem performansını ve hataları izler ve günlüğe kaydeder.

Sorumluluk: Sorunların hızlı bir şekilde tanımlanmasına ve çözülmesine olanak tanıyarak sistem durumu ve performansına ilişkin öngörüler sağlar.

Bileşenler:

Prometheus/Grafana: İzleme ve görselleştirme için.

ELK Stack (Elasticsearch, Logstash, Kibana): Günlük kaydı ve analiz için.



**Soru 1**

Öğrenim hayatım boyunca yaptığım projeler gerçek dünya tabanlı projeler olmadığı için güvenlik önlemleri gözetilmeden yaptık.

Cross-Site Scripting (XSS), saldırganların kötü amaçlı kodları (genellikle JavaScript) başka bir kullanıcının tarayıcısında çalıştırmasına olanak tanıyan bir güvenlik açığı türüdür. XSS saldırıları genellikle kullanıcı girdilerinin yeterince doğrulanmadığı veya temizlenmediği web uygulamalarında meydana gelir.

SQL Injection (SQL Enjeksiyonu), saldırganların kötü amaçlı SQL kodlarını bir web uygulamasının veri tabanına eklemesine olanak tanıyan bir güvenlik açığı türüdür. Bu saldırılar, uygulamanın SQL sorgularını oluştururken kullanıcı girdilerini doğrudan kullanmasından kaynaklanır.

**Soru 2**

**1. JVM Ayarları**

Heap Boyutunu Ayarlama: -Xms ve -Xmx parametreleri ile başlangıç ve maksimum heap boyutlarını ayarlayın.

Garbage Collection Ayarları: Uygulamanıza uygun bir garbage collection (GC) algoritması seçin (örneğin, G1GC, ZGC).

JVM İzleme: -XX:+PrintGCDetails ve -XX:+PrintGCTimeStamps gibi seçeneklerle GC loglarını izleyin ve analiz edin.

**2. Veritabanı Endekslemesi**

Doğru Endeks Kullanımı: Sıkça sorgulanan sütunlara endeks ekleyin.

Sorgu Optimizasyonu: Sorgularınızı optimize edin, gereksiz veri çekmekten kaçının.

Bağlantı Havuzu: Veri tabanı bağlantı havuzu (connection pool) kullanarak bağlantı açma ve kapama maliyetini azaltın.

**3. Kod İncelemesi**

Kod Profilleme: Profiling araçları (JProfiler, VisualVM) kullanarak performans darboğazlarını belirleyin.

Verimli Algoritmalar: Daha verimli veri yapıları ve algoritmalar kullanın.

İstemci-Sunucu İletişimi: Gereksiz ağ isteklerini azaltın, veri sıkıştırma kullanın.

Concurrency (Eşzamanlılık): Paralel işlemler ve thread yönetimini optimize edin.

**Soru 4**

**JSON (JavaScript Object Notation)**

Özellikler:

Basit ve hafif

İnsan tarafından okunabilir

JavaScript ile uyumlu

**Güvenlik Önlemleri:**

Girdi doğrulama ve temizleme

JSONP kullanırken dikkat

XSS saldırılarına karşı koruma

Doğru Content-Type başlığı kullanımı

**XML (eXtensible Markup Language)**

Özellikler:

Esnek yapı

İnsan tarafından okunabilir

Genişletilebilir

**Güvenlik Önlemleri:**

XML doğrulama (DTD veya XSD)

XXE saldırılarına karşı koruma

Girdi doğrulama ve temizleme

XML imzaları ve şifreleme

DoS saldırılarına karşı koruma

**Soru 5**

Elimizde totalde 100 adet order var ve customer id 123 olan 15 order var. Soruda bizden istenilen customer id’si 123 olmayan toplam kaç order olduğudur. Bunun sonucu da 85’tir.