

JAVA CHALLENGE CEVAPLAR

1- SAP Commerce (Hybris) nedir? Hangi amaçlarla kullanılır? Kullandığı teknolojiler nelerdir? Kısaca açıklayınız.

SAP Commerce, SAP tarafından geliştirilmiş önemli ölçüde e-ticaret çözümleri sunan bir platformdur. Bu platform, işletmelere çok kanallı bir ticaret deneyimi sunar. Büyük çaplı işletmelerin ticari operasyonlarını dijital ortamda optimize etmeleri için kullanılır. Tüm müşterilerin özel ihtiyaçlarına göre uyarlanabilen; online, mobil, mağaza, depo, sosyal medya, çağrı merkezi gibi müşteriye temas edilen noktaları birleştirerek; tekil bir müşteri deneyimi yaşanmasına olanak kılar. SAP Commerce genel olarak Java ve Spring tabanlıdır. Ürün arama ve indeksleme için Apache Solr kullanır. Diğer sistemlerle haberleşmesi için RESTFUL, SOAP gibi web servislerini destekler. MySQL, HSQldb, Microsoft SQL gibi veritabanlarıyla uyumlu çalışır. Angular gibi modern frontend framework'u kullanır. Cloud teknolojisini kullanarak bulut teknolojisinin gücünü kullanır. B2B, B2C gibi e-ticaret çözümleri sunan hızlandırıcı modüllerle birlikte gelir.

2- Birbirinden bağımsız iki platformun birbiriyle haberleşmesi nasıl sağlanabilir? Örneğin, X platformu Java ile yazılmış olsun, Y platform u C# ile. Bu iki platformun birbiri ile iletişim halinde request-response ilişkisi kurması gerekiyor. Bu yapıyı nasıl sağlarız? Bu iletişim sırasında güvenlik nasıl sağlanır?

Böyle bir yapıda en yaygın kullanılan yöntemler web servisleri ve API'ler üzerinden gerçekleşebilir. İletişim güvenliğini sağlamak için çeşitli güvenlik önlemleri alabiliriz. Restful API kullanabiliriz. Bu yapı İki bağımsız uygulamanın birbiriyle HTTP üzerinden veri alışverişini sağlar. Restful API'ler JSON ve XML formatta veri iletimi sağlar, en yaygın olanı JSON'dur. Örneğin Java Spring Boot kullanarak, C# Asp.Net Core kullanarak bir REST API oluşturabiliriz. Bu iki yapının CRUD yöntemleriyle haberleşmesi sağlanabilir. Aynı zamanda SOAP kavramını da haberleşme için kullanabiliriz. Araştırdığımda SOAP XML tabanlı bir protokol olduğunu ve platformdan bağımsız olduğunu öğrendim. Java JAX-WS gibi bir kütüphane ile SOAP servisi oluşturabilirdik. C# tarafında ise WCF ile SOAP servis yapısı oluşturabilirdik. SOAP ile iki sistemin aynı protokole göre iletişim kurması sağlanabilir. Eğer asenkron bir haberleşme istiyorsak Message Queue kullanabiliriz. Bu yapının büyük sistemlerde yaygın bir şekilde kullanımı varmış. Diğer bir iletişim türü ise gRPC framework'üdür. Rest yerine Protobuf gibi bir format kullanırmış ve genellikle daha hızlı olduğu görülmüş. gRPC iki platform arasında fonksiyon çağrıları yapar ve bu çağrılar network üzerinden gerçekleştirmiş. Anlaşıldığı üzere ihtiyaca göre iletişim türleri mevcuttur.

Güvenlik kısmında ise HTTPS, API Keys, JWT gibi yapılar kullanabiliriz.

3- SOLR Nedir? Kullanım alanlarını araştırınız. Kurumsal bir projede kullanılabilecek iki farklı kullanım alanı örneği veriniz.

Apache Solr, açık kaynaklı bir arama ve indeksleme motorudur. Büyük miktardaki veriyi hızlı bir şekilde arama, indeksleme ve sorgulama için kullanılır. Solr, Lucene kütüphanesi üzerine inşa edilmiş ve zengin sorgulama yeteneklerine sahiptir. Veri analizi,

sınıflandırma, otomatik tamamlama ve daha birçok gelişmiş arama fonksiyonları içerir. Örnek olarak bir e-ticaret sitesini düşünebiliriz. E-ticaret siteleri geniş ürün katalogları ve çok sayıda müşteri arama istekleriyle çalışır. SOLR, bu tür platformlarda ürün arama, filtreleme, sıralamada kullanılır. Örneğin kullanıcının girdiği anahtar kelimelere göre ürün araması yapmak ve kullanıcıya en alakalı ürünleri hızlıca sunmak için SOLR kullanılabilir. Başka bir kullanım alanı ise büyük şirketlerde görülüyor. Büyük şirketlerde çok fazla sayıda belge, rapor, belge ve bilgi notu vardır. Bu verilerin etkili bir şekilde aranabilmesi ve yönetilmesi için SOLR kullanılabilir. Özetlemek gerekirse Apache SOLR büyük veri kümeleri üzerinde hızlı ve esnek arama yapabilmek için bir güçlü çözüm olduğunu görüyoruz. Verdiğim örneklerle de kullanım alanlarının oldukça geniş olduğunu da vurgulamak gerekir.

4- Aşağıdaki algoritma için uygun çözümü üretin.

- Java'da 100 adet random sayıya sahip bir liste oluşturun.
- Daha sonra bu listenin bir kopyasını oluşturun.
- 0 ile 100 arasında rastgele bir sayı üretin.
- Kopya listedeki bu random sayının olduğu indisteski değeri silin.
- Şimdi elinizde iki adet liste var ve kopya listede orjinal listeye göre bir eleman eksik.
- Hangi elemanın eksik olduğunu bulan bir metot oluşturun.

```
1  import java.util.ArrayList;
2  import java.util.List;
3  import java.util.Random;
4
5  public class RandomNumberList {
6
7      public static void main(String[] args) {
8
9          final int ListNumber = 100;
10
11          List<Integer> originalList = GenerateRandomList.generateRandomList(ListNumber);
12
13          List<Integer> copyList = new ArrayList<>(originalList);
14
15          Random random = new Random();
16          int randomIndex = random.nextInt(ListNumber);
17          int removedElement = copyList.get(randomIndex);
18
19          copyList.remove(randomIndex);
20
21          int missingElement = FindMissingElement.findMissingElement(originalList, copyList);
22
23          System.out.println("Original Liste: " + originalList);
24          System.out.println("Kopya Liste (Bir eleman eksik): " + copyList);
25          System.out.println("Silinen Eleman: " + removedElement);
26          System.out.println("Eksik Eleman: " + missingElement);
27      }
28
29  }
```

```

1  import java.util.ArrayList;
2  import java.util.List;
3  import java.util.Random;
4
5  public class GenerateRandomList { 1 usage
6
7  @    public static List<Integer> generateRandomList(int size) { 1 usage
8      List<Integer> list = new ArrayList<>();
9      Random random = new Random();
10     for (int i = 0; i < size; i++) {
11         list.add(random.nextInt( bound: size + 1));|
12     }
13     return list;
14 }
15
16 }
17

```

```

1  import java.util.List;
2
3  public class FindMissingElement { 1 usage
4
5  @    public static int findMissingElement(List<Integer> originalList, List<Integer> copyList) { 1 usage
6
7      int originalSum = 0;
8      int copySum = 0;
9
10     for (int num : originalList) {
11         originalSum += num;
12     }
13
14     for (int num : copyList) {
15         copySum += num;
16     }
17
18     return originalSum - copySum;
19 }
20
21 }
22

```