



JAVA CHALLENGE

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

Kişilere veya işletmelerce e-ticaret süreçlerini yönetmek için kullanılabilecek çok amaçlı bir platformdur. Çok kanallı satıştan müşteri ilişkilerine kadar, ürün bilgi yönetimi, sipariş yönetimi ve pazarlama otomasyonu gibi alanlarda zengin özelliklere sahiptir. Java, Spring Boot, SQL Hybris Accelerator ve Cloud Foundry gibi teknolojilerle geliştirilmiştir.

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

SOAP, REST gibi web servisleri, platformlar arasında iletişimi kolaylaştırır, XML veya JSON gibi formatlarla veri alışverişi yapılabilir. Güvenlik için de HTTPS ve kimlik doğrulama çözümleri kullanılabilir.

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

SOLR, açık kaynaklı, yüksek performanslı bir arama moturu platformudur. NoSQL veri tabanı ailesine aittir ve metin, belge , e-posta ve medya dosyası gibi çeşitli veri türlerini indekslemek ve aramak için kullanılır. E-ticaret sitelerinde ürün arama/kataloglama, kütüphane gibi platformlarda belge arama/indeksleme, log dosyalarını indekslemek ve olayları analiz etmek için kullanılabilir.

1.Siber Güvenlik: Bir siber güvenlik ekibi, ağ trafiğini ve güvenlik olaylarını indekslemek için kullanabilir. Bu, tehditleri daha hızlı tespit etmelerine ve siber saldırılara karşı aktif olarak savunma yapmalarına yardımcı olabilir. Fidyeye yazılımı ve diğer zararlı yazılımlar tarafından kullanılan dosyaları tanımlamak için kullanılır. Bu, siber saldırıları önlemeye ve sistemleri fidye yazılımı saldırılarından korumaya yardımcı olabilir.

2.Kişiselleştirilmiş Öneri: Geçmiş aramalara, satın alma geçmişine ve demografik verilere dayalı olarak kişiselleştirilmiş ürün önerileri sunmak için kullanılabilir. Bir kıyafet e-ticaret platformunda, müşterilerin geçmiş aramalarına ve satın almalarına dayanarak otomatik ürün önerileri sunmak için kullanılabilir

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.

NOT: Lütfen internette gördüğünüz veya yapay zekadan oluşturduğunuz metotları kullanmayınız.

En hızlı çözümü üretememiş olsanız bile kendi oluşturduğunuz eşsiz ve mantıklı bir algoritma kullanmayı tercih ediniz.

4.soru kodu ve github linki : <https://github.com/Onur-Akman/java-backend-challenge/tree/main/Question%204>

```
1  import java.util.Random;
2  import java.util.ArrayList;
3  import java.util.Collections;
4  import java.util.List;
5
6  public class Main {
7  public static void main(String[] args) {
8      Random rand = new Random();
9      List<Integer> nums1 = new ArrayList<>(); // İlk listemizi olusturdum.
10     for(int x=0;x<100;x++) {
11         int exp1 = rand.nextInt(); // Rastgele olarak 100 adet sayı olusturup 1.listeye ekledim.
12         nums1.add(exp1);
13     }
14     System.out.println("Birinci liste: " + nums1);
15     List<Integer> nums2 = new ArrayList<>(); // İkinci listemizi olusturdum.
16     nums2.addAll(nums1); // Birinci listedeki tüm elemanları ikinci listeye ekledim.
17     System.out.println("Kopya liste: " + nums2);
18     int exp2 = rand.nextInt( bound: 101); // 0-100 arasında sayı Üretiyorum.
19     nums2.remove(exp2); // Random Ürettiğim sayının olduğu indeksteki değeri siliyorum.
20     System.out.println("Silinen ikinci liste : " + nums2);
21     for (Integer num : nums1) {
22         if (!nums2.contains(num)) { // 2.liste eğer ilk listeden bir elemanı içermiyorsa ekrana yazdıracak.
23             System.out.println("Eksik Eleman: " + num);
24             break;
25         }
26     }
27 }
28 }
```

5. Product, Customer, Cart ve Order tablolarının ve bu tablolarının miras aldığı bir Base Entity'nin bulunduğu Spring Boot ile geliştirilmiş bir proje oluşturun. Bir müşterinin bir sepeti (cart) ve birden fazla siparişi (order) olabilecek şekilde ilişkilendirme işlemini gerçekleştirin. Sepetin ve siparişin toplam fiyat bilgisi her işlemde (sepete ekleme, çıkarma, miktar artırıp azaltma) hesaplınsın ve sepete kaydedilsin. Bir müşteri sipariş geçtikten sonra, sipariş içerisindeki ürünlerin fiyatı daha sonradan güncellendiğinde müşteri satın aldığı anki fiyatı geçmişe yönelik olarak görebilsin. Bunun için farklı bir tablo tutabilirsiniz. Bu tablo üzerinde ürün, fiyat, miktar gibi bilgiler tutulabilir. Ürün üzerinde stok takibi yapılsın, bir ürünün stoğu bittikten sonra o üründen daha fazla sipariş verilemesin.

Yazılması beklenen servisler:

AddCustomer

GetProduct

CreateProduct

UpdateProduct

DeleteProduct

GetCart

UpdateCart

EmptyCart

PlaceOrder

GetOrderForCode

GetAllOrdersForCustomer

AddProductToCart

RemoveProductFromCart

Challenge Notu

Birinci, ikinci ve üçüncü soruları araştırmalarınız ve kendi yorumlarınız ile açıklayınız. Dördüncü sorudaki isteğe ait kaynak kodların ekran görüntülerini (ve varsa açıklamalarınızı) ilk üç soruyla aynı dokümanda yeni bir sayfaya ekleyiniz. Beşinci soruda projenizi tamamladıktan sonra kaynak kodunuzu bir repository'e (Github, Bitbucket vs.) ekleyin. Proje için kısa bir doküman hazırlayın. İlk 4 soru için dokümanınızı, 5. soru için proje dokümanınızı ve kaynak kodunuzun olduğu repository linkini maile ekleyiniz. Süre (haftasonu dahil) 5 gündür.

5.soru GitHub linki: <https://github.com/Onur-Akman/java-backend-challenge/tree/main/Question%205>