

1) SAP Commerce, önceki adıyla Hybris Commerce, müşteri deneyimi yönetimi ve çok kanallı ticaretin birleşimi için tasarlanmış bir e-ticaret platformudur. SAP Commerce, işletmelere çeşitli amaçlarla kullanılabilecek bir dizi özellik sunar:

1. Çok Kanallı Ticaret: Fiziksel mağazalar, online alışveriş siteleri, mobil uygulamalar gibi farklı kanallar aracılığıyla müşterilere ürün ve hizmet sunma yeteneği.
2. Dijital Deneyim Yönetimi: Zengin içerik yönetimi, kişiselleştirme ve kullanıcı dostu arayüzlerle müşteri deneyimini artırma yeteneği.
3. Müşteri İlişkileri Yönetimi (CRM): Müşteri bilgilerini merkezi bir konumda toplama, müşteri profilleri oluşturma ve müşteri davranışlarına dair verileri analiz etme yeteneği.
4. Sipariş ve Stok Yönetimi: Sipariş işlemlerini takip etme, stok yönetimi ve lojistik süreçlerini optimize etme yeteneği.
5. Analitik ve Raporlama: İş zekası araçları ile satış verilerini analiz etme, raporlama yapma ve iş süreçlerini geliştirme yeteneği.

SAP Commerce, Java tabanlı bir yazılım çerçevesi olan Spring Framework üzerine inşa edilmiştir. Ayrıca, çeşitli açık kaynak teknolojilerini içerir ve geniş bir teknoloji yelpazesıyla entegre edilebilir. RESTful web servisleri, Apache Solr gibi arama motorları, hızlandırılmış cephe teknolojileri, JavaServer Faces (JSF), ve çeşitli veritabanı yönetim sistemleri ile uyumludur.

Özetle, SAP Commerce, müşteri odaklı bir e-ticaret deneyimi sağlamak ve işletmelerin çok kanallı ticaret ve müşteri ilişkilerini etkili bir şekilde yönetmelerine yardımcı olmak amacıyla tasarlanmış bir platformdur.

2) Birbirinden bağımsız iki platform arasında request-response ilişkisi kurmak ve bu iletişimi güvenli hale getirmek için birkaç yaygın kullanılan yöntem bulunmaktadır. İşte bu tür bir iletişimi sağlamak için kullanılabilecek bazı yöntemler: RESTful Web Servisleri veya API'lar, SOAP Web Servisleri, Message Queues (Mesaj Kuyrukları), gRPC (Google Remote Procedure Call)

İletişim sırasında güvenliği sağlamak için ise şu yöntemler kullanılabilir: HTTPS (SSL/TLS), API Anahtarları veya Tokenlar, OAuth veya JWT (JSON Web Tokens), Firewall ve Güvenlik Duvarları, Veri Şifreleme

3) Solr Apache Lucene projesinin bir parçası olan tam metin arama, çok yönlü arama, belgeleri indeksleme gibi özellikleri olan açık kaynak kodlu oldukça esnek bir arama motorudur. Temelinde arama motoru olmasına rağmen verilerin replikasyonu ve çoklu core'lara izin vermesi gibi birçok özelliğiyle bir arama motorundan daha fazlasıdır. Solr, genel bir amaçlı arama platformu olduğundan, kullanım alanları çok çeşitlidir ve birçok farklı endüstri ve kurumsal projede uygulanabilir.

Kullanım alanları: E-ticaret ve Ürün Arama, Belge ve İçerik Yönetimi Haber ve Medya Siteleri ,Büyük Veri Analizi ve İstatistiksel Çalışmalar

Müşteri İlişkileri Yönetimi (CRM): Bir kurumsal projede, müşteri bilgilerini yönetmek ve aramak için Solr kullanılabilir. Müşteri ilişkileri yönetimi (CRM) sistemleri, müşteri bilgilerini indeksleme ve hızlı arama yetenekleriyle müşteri temsilcilerinin veya satış ekibinin veriye kolayca erişmelerini sağlayabilir.

İnsan Kaynakları Yönetimi: İnsan kaynakları yönetimi projelerinde, personel dosyaları, CV'ler, yetenek havuzları gibi büyük veri setlerini yönetmek ve aramak amacıyla Solr kullanılabilir. İnsan kaynakları ekipleri, işe alım süreçlerini daha etkili hale getirmek için Solr'ü kullanarak aday verilerini hızlı bir şekilde sorgulayabilirler.

4)

```
J test.java
1  # enoca-proje
2  # Challenge
3  import java.util.ArrayList;
4  import java.util.Collections;
5  import java.util.List;
6  import java.util.Random;
7  public class Main {
8      public static void main(String[] args) {
9
10         // 1. Adım: 100 adet random sayıya sahip bir liste oluşturun
11         List<Integer> originalList = generateRandomList(100);
12
13         // 2. Adım: Listenin bir kopyasını oluşturun
14         List<Integer> copyList = new ArrayList<>(originalList);
15
16         // 3. Adım: 0 ile 100 arasında rastgele bir sayı üretin
17         Random random = new Random();
18         int randomIndex = random.nextInt(100);
19
20         // 4. Adım: Kopya listedeki bu random sayının olduğu indisteki değeri silin
21         copyList.remove(randomIndex);
22
23         // 5. Adım: Hangi elemanın eksik olduğunu bulan metot
24         int missingElement = findMissingElement(originalList, copyList);
25
26         // Sonuçları ekrana yazdır
27         System.out.println("Original Liste: " + originalList);
28         System.out.println("Kopya Liste: " + copyList);
29         System.out.println("Eksik Eleman: " + missingElement);
30     }
31
32     // Metot: Hangi elemanın eksik olduğunu bulan metot
33     private static int findMissingElement(List<Integer> originalList, List<Integer> copyList) {
34         Collections.sort(originalList);
35         Collections.sort(copyList);
36
37         for (int i = 0; i < originalList.size(); i++) {
38             if (!originalList.get(i).equals(copyList.get(i))) {
39                 return originalList.get(i);
40             }
41         }
42
43         // Eğer buraya kadar gelinirse, listenin son elemanı eksiktir.
44         return originalList.get(originalList.size() - 1);
45     }
46
47     // Metot: Belirtilen sayıda random sayı içeren bir liste oluşturur
48     private static List<Integer> generateRandomList(int size) {
49         List<Integer> randomList = new ArrayList<>();
50         Random random = new Random();
51
52         for (int i = 0; i < size; i++) {
53             randomList.add(random.nextInt(101)); // 0 ile 100 arasında random sayı üret
54         }
55
56         return randomList;
57     }
58 }
59
```

Java hakkında çok bir bilgim olmadığı için yapay zekadan yardım aldım.

