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

SAP Commerce, ticari işletmelerin çevrimiçi satış yapmalarını sağlayan bir e-ticaret - platformudur. Başlangıçta Hybris olarak bilinen bu platform, SAP tarafından satın alındıktan sonra SAP Commerce olarak yeniden adlandırılmıştır. SAP Commerce'un ana amacı, işletmelerin müşteri deneyimini geliştirmek, satışları artırmak ve operasyonel verimliliği sağlamaktır.

SAP Commerce, modüler bir yapıya sahiptir ve işletmelerin ihtiyaçlarına göre özelleştirilebilir. Temel özellikleri arasında çevrimiçi mağaza yönetimi, ürün yönetimi, sipariş yönetimi, müşteri yönetimi, pazarlama ve kişiselleştirme araçları bulunur.

SAP Commerce, Java ve Spring Framework kullanılarak geliştirilmiştir ve RESTful API'ler aracılığıyla dış sistemlerle entegrasyon sağlar. Ayrıca, HTML, CSS ve JavaScript gibi web teknolojileri kullanılarak zengin bir kullanıcı arayüzü oluşturulur. Arama ve filtreleme için Apache Solr gibi araçlar kullanılırken, veritabanı olarak MySQL, Oracle veya SAP HANA gibi seçenekler sunulur.

SAP Commerce'un sunduğu avantajları maddeleyecek olursak:

*Çok kanallı satış imkanı,

*Kişiselleştirilmiş müşteri deneyimi,

*Esneklik ve ölçeklenebilirlik,

*Entegre ticaret yönetimi ve kapsamlı analitik raporlama bulunur.Böylece işletmeler, rekabet avantajı elde edebilir ve çevrimiçi satış stratejilerini geliştirebilirler.

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?

Farklı platformlar arasında iletişim kurmak için en yaygın yöntemlerden biri HTTP tabanlı API'lerdir. Bu API'ler, farklı dillerde ve platformlarda yazılmış uygulamalar arasında veri alışverişini sağlar. Örneğin, Java tarafında Spring Boot veya C# tarafında ASP.NET Core gibi framework'ler kullanılarak RESTful API'ler oluşturulabilir. Bu API'ler sayesinde, farklı platformlardaki uygulamalar arasında JSON veya XML formatında veri alışverişi yapılabilir.

Bir diğer iletişim yöntemi ise mesaj kuyruklarıdır. Mesaj kuyrukları, farklı platformlar arasında asenkron iletişim sağlar. Bir platformdan diğerine mesaj gönderildiğinde, mesaj kuyruğu tarafından saklanır ve hedef platform bu mesajı işleyene kadar bekletilir. Bu şekilde, uygulamalar arasında güvenilir ve sıralı mesaj alışverişi sağlanabilir. Örneğin, Java tarafında RabbitMQ veya Apache Kafka, C# tarafında ise MSMQ veya Azure Service Bus gibi mesaj kuyruğu sistemleri kullanılabilir.

İletişim sırasında güvenliği sağlamak için HTTPS kullanımı tavsiye edilir. HTTPS, iletişimi şifreleyerek verilerin güvenliğini sağlar. Ayrıca, kimlik doğrulama için JWT gibi token tabanlı yöntemler tercih edilebilir. Bu yöntemler sayesinde, iletişim sırasında veri güvenliği sağlanır ve

yetkilendirme işlemleri gerçekleştirilir. Uygulanacak yöntemler, proje gereksinimlerine ve kullanılan teknolojilere bağlı olarak değişiklik gösterebilir.

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

Apache Solr, Apache Software Foundation tarafından geliştirilen, açık kaynaklı bir arama platformudur. Solr, metin tabanlı veriler üzerinde hızlı ve etkili arama işlemleri yapmak için kullanılır. Genellikle büyük ölçekli metin tabanlı veri kümelerini indekslemek ve sorgulamak için tercih edilir. İşte Solr'un kullanım alanlarından bazıları:

*Solr, büyük metin belgelerini indeksleyerek hızlı arama sağlar. Özellikle belge yönetimi sistemleri ve web siteleri gibi uygulamalarda yaygın olarak kullanılır.

*E-ticaret siteleri, binlerce ürünü hızlıca arayabilen bir arama motoruna ihtiyaç duyarlar. Solr, ürün kataloğunu indeksleyerek alışveriş yapmak isteyen kullanıcılara hızlı ve doğru sonuçlar sunabilir.

*Solr, büyük veri kümeleri üzerinde analiz yapmak için de kullanılabilir. Log dosyaları, sosyal medya verileri gibi veriler üzerinde yapılan analizlerde kullanılır. Metin analiz özellikleri, kelime sıklığı gibi analizler için uygundur.

Kurumsal çevrelerde, büyük miktarda dokümantasyon ve bilgi yönetimi ihtiyacı vardır. Solr, kurumsal belgeleri indeksleyerek kullanıcıların hızlıca erişebilmesini sağlar ve kullanıcı deneyimini artırır.

Kurumsal bir projede Solr'ün kullanılabileceği iki farklı alan :

**Bir Şirketin İç Bilgi Portalı:* Şirket içindeki dokümantasyon, raporlar, eğitim materyalleri vb. gibi içeriklerin bir araya getirilmesi ve etkili bir şekilde aranması için Solr kullanılabilir. Kullanıcılar, bilgileri hızlıca bulabilir ve erişebilir, böylece iş verimliliği artar.

**E-ticaret Platformu:* Bir kurumsal e-ticaret platformu, milyonlarca ürünü indeksleyip arayabilen bir arama motoruna ihtiyaç duyar. Solr, ürün kataloğunu indeksleyerek hızlı ve doğru sonuçlar sağlayabilir. Aynı zamanda, filtreleme, sıralama özellikleriyle kullanıcıların alışveriş deneyimini iyileştirebilir.

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.**

```java
import java.util.*;

public class RandomNumber {

    public static void main(String[] args) {

        List<Integer> List = generateRandomList( size: 100);

        List<Integer> copyList = new ArrayList<>(List);
        Random rand = new Random();
        int randomIndex = rand.nextInt(copyList.size());
        copyList.remove(randomIndex);

        int missingNumber = findMissingNumber(List, copyList);
        System.out.println("Eksik olan eleman: " + missingNumber);
    }
    1 usage
    public static List<Integer> generateRandomList(int size) {
        List<Integer> list = new ArrayList<>();
        Random rand = new Random();
        for (int i = 0; i < size; i++) {
            list.add(rand.nextInt( bound: 101));
        }
        return list;
    }
```

```java
    1 usage
    public static int findMissingNumber(List<Integer> originalList, List<Integer> copyList) {
        int missingNumber = 0;
        for (int num : originalList) {
            if (!copyList.contains(num)) {
                missingNumber = num;
                break;
            }
        }
        return missingNumber;
    }
}
```