**Enuca Challenge**

**1-)**MVC Model - View - Controller kavramının kısaltmasıdır ve yazılım geliştirme sürecinde sıkça kullanılan bir tasarım kalıbıdır. Bir uygulamanın farklı bileşenlerinin birbirinden ayrılmasını sağlayarak daha modüler, esnek bir kod tabanı oluşturmayı amaçlar . Mvc’nin temel amacı veri işleme sunum ve kullanıcı etkileşimlerinin birbirinden bağımsız olmasını sağlamaktadır .Java'da, MVC deseni genellikle bir web uygulaması veya masaüstü uygulaması geliştirirken kullanılır. Örneğin, bir web uygulaması için, Controller kısmı web sunucusu tarafından sağlanan bir servlet veya Spring MVC Controller olarak tasarlanabilir. Model kısmı, veritabanı işlemlerini gerçekleştirmek için kullanılabilir. View kısmı ise HTML, CSS ve JavaScript ile tasarlanabilir.Object Oriented programlama ise, nesne yönelimli programlamanın kısaltmasıdır. nesnelerin birbirleriyle etkileşim halinde olduğu ve bu nesnelerin birbirleriyle ilişkili olan veriler ve işlevler içerdiği bir programlama yaklaşımıdır.Katmanları, sınıflar ve nesneler şeklinde oluşturulur. OOP'de sınıflar, nesnelerin özelliklerini ve davranışlarını tanımlayan şablonlardır. Nesneler, sınıfların örnekleridir ve verileri ve işlevleri içerir. OOP, kod tekrarını azaltır ve yazılım projelerinin daha organize edilmesine olanak tanır.

**2-)** Farklı platformlar arasında haberleşme sağlamak için RESTful API veya SOAP protokolü gibi ortak bir protokol kullanılabilir. Her iki platformda da bu protokolleri destekleyen teknolojiler mevcuttur. Veri alışverişi için kullanılacak protokol belirlendikten sonra, platformlar arasında haberleşme sağlanabilir.X platformu Java ile yazılmış olsun, Y platformu C# ile yazılmış olsun, iki platformun da desteklediği bir protokol (örneğin, REST) kullanarak request-response ilişkisi kurabilirsiniz. Bu, farklı platformlar arasında etkili bir iletişim sağlayabilir.

**3-)** Web sayfasında ekrana Backend veya bir başka yapı tarafından güncellenen bilgileri refresh etmeden anlık olarak yansıtmak için, WebSocket teknolojisi kullanılabilir. WebSocket teknolojisi, istemci (web sayfası) ve sunucu arasında sürekli açık bir bağlantı sağlar ve herhangi bir zamanda veri akışı gerçekleştirilmesine izin verir.

**4-)code**

**5-)**Öncelikle, verilen IP adresi, port, kullanıcı adı ve şifre ile sunucuya SSH protokolü üzerinden erişirim. SSH protokolü, güvenli bir şekilde sunucuya uzaktan erişim sağlar ve benim işletim sistemimde bir terminal açarak sunucuda komut çalıştırmama olanak sağlar. Sunucuya erişmek için terminali açarım. “ssh kullaniciadi@ipadresi -p port ” komutunu kullanarak sunucuya SSH bağlantısı kurabilirim. Daha sonra ekrana kullanıcının şifresini girerek sunucuya bağlanırım.Sunucuya bağlandıktan sonra sunucya dosya atmak için “scp” komutunu kullanırım. Mesela örnek olarak :

“scp /local/path/to/file kullaniciadi@ipadresi:/remote/path/to/file”

Bu komut, “ /local/path/to/file ” adlı dosyayı sunucuya “/remote/path/to/file” adlı konuma kopyalar.

Sunucudan dosya çekmek içinde “scp” komutunu kullanırım.

Örneğin, sunucudan “/remote/path/to/file” adlı dosyayı indirmek için aşağıdaki komutu kullanabilirim:

“scp kullaniciadi@ipadresi:/remote/path/to/file /local/path/to/file”

Bu komut, “/remote/path/to/file” adlı dosyayı sunucudan indirir ve “/local/path/to/file” konumuna kaydeder.

**6-)** code

**7-)**Solr sorguları Lucene sorgu syntax'ını kullanır. "updatedAt" alanının veri tipi "long" olduğundan, tarih filtrelemesi için Epoch zamanı kullanılabilir. Ayrıca Solr, tarih aralığı sorgulamaları için özel syntax'lar da sağlar.

2020 Ocak ayından sonra güncellenen kayıtları getirmek için, Solr query'si aşağıdaki şekilde olabilir:

**http://example?fq=updatedAt:[2020-01-01T00:00:00Z TO \*]**

Bu sorgu "updatedAt" alanı 2020 Ocak ayından sonrasını içeren tüm kayıtları döndürecektir.