**Hergele Backend Case Study**  
  
1.   Mongodbde ücretsiz olarak bir database açılması ve bu databasede   aşağıda görülen yapıda örnek 10 adet döküman oluşturulması beklenmektedir. Potansiyel olarak collectiondaki döküman sayısı arttığında buna çözüm olacak indexlerin oluşturulması ve scale amaçlı yapılacak ayarların, adımların kısaca açıklanması beklenmektedir.

{

“userNo”:””,

“authCode”:””,

"name": "",

"surname": "",

"birthDate": ""

"phoneNumber": "",

"email": "",

“selectedCard”:””,  
 “allCards”: [“”,””],

"balance": 0,

}

2. Verilen gereksinimlere göre Flask kullanarak aşağıda istenen REST API'leri yazmanız beklenmektedir.

<https://dev.param.com.tr/tr/api/kart-saklama>

<https://dev.param.com.tr/tr/api/kart-saklamali-odeme>

<https://dev.param.com.tr/tr/api/islem-iptal-ve-iadeleri>

<https://dev.param.com.tr/tr/api/kart-saklama-listesi>

Yukarıdaki verilen Param ödeme sisteminin test apilarını(<https://dev.param.com.tr/tr>), Flask REST API olarak oluşturunuz.

 Birinci endpointte kart saklama ,

İkinci endpointte saklı karttan para çekme,

Üçüncü endpointte çekilen paranın iadesi ,

Dördüncü endpointte ise o kullanıcıya ait kartların listelenmesi beklenmektedir.

  Kullanıcı doğrulama tarafında Header Authorization kullanılması beklenmektedir.  
 Örnek Header Yapısı:

Filo => filo ismi (“hergele”)

Authorization => authorization code

  UserNo => kullanıcının sisteme kayıtlı numarası

Kullanıcı doğrulaması yaparken, 1. adımda oluşturduğunuz database’e bağlanarak kullanıcıyı doğrulamalısınız. REST API’nin sadece POST requestlere cevap verecek şekilde yazılması beklenmektedir. Bu süreçte Param api’lerinden dönen yanıtlar basit bir düzen içerisinde databasede farklı bir collectionda saklanmalıdır. Yazılan endpointlerin çalıştığını gösteren istek yapılarının postman üzerinden collection olarak paylaşılması gerekmektedir.

3. AWS sunucularında kendi açacağınız hesabınız ile 2 adet instance açmanız beklenmektedir. Açtığınız instancelarda 2. adımda yazdığınız flask REST API’nin 80. Veya 443. Porttan yayın yapması beklenmektedir. Instancelarınızdan birinin fail olması durumunda yine amazon servislerini kullanarak aktif olan diğer sunucuya yönlendirme sağlamanız ve fail olan sunucunun down olduğunu mail ile iletecek bir yapı kurmanız beklenmektedir. Hali hazırda oluşturduğunuz failover senaryosu, tek başına veya ek bir çözüm ile birlikte, TCP socket scriptlerinde sorun yaşandığında da yönlendirme sağlayacak şekilde olmalıdır. Oluşturduğunuz sistemi özetleyen bir yazıyı, test amaçlı uyguladığınız yöntemleri ve sonuçlarını gösteren görseller(tablolar) ile birlikte paylaşınız.

Ekler:  
<https://www.mongodb.com/>  
[https://aws.amazon.com/tr/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=\*all&awsf.Free%20Tier%20Categories=\*all](https://aws.amazon.com/tr/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=*all&awsf.Free%20Tier%20Categories=*all)