

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BSM 401 YAZILIM TESTİ DERSİ PROJE ÖDEVİ



Fakülte Anabilim Dalı : **BİLGİSAYAR MÜHENDİSLİĞİ**

Ders Hocası : **Dr.Öğr.Üyesi Muhammed Fatih ADAK**

İsim : **Beyzanur Odabaş**

Numara : **B201210100**

Grup : **Yazılım Testi / 1-A**

Github Linki :

<https://github.com/beyzanurodabas/Django-Web-Selenium-Test>

1. Ödevde İstenilen

SABİS dokümanlardaki template kullanılarak Django python web projesi yapılması beklenmektedir. Bu projede login,register,industry ekleme/listeleme/tek açma ve blog ekleme/listeleme/tek açma sayfaları güncellenmelidir. Web projesinde database olarak Postgresql kullanılmalı ve docker ile ayağa kaldırılması beklenmektedir.

Site çalışır durumda iken 2. bir proje olarak Python konsol uygulaması yapılmalıdır. Selenium kütüphanesi kullanılarak proje ile ilgili 30 farklı test gerçekleştirilmelidir. Bu testlerin en az 8 tanesinde xpath kullanılmalıdır. Testler kategorilere bölünmeli ve çalıştırılmalıdır.

2. Öğrendiklerim

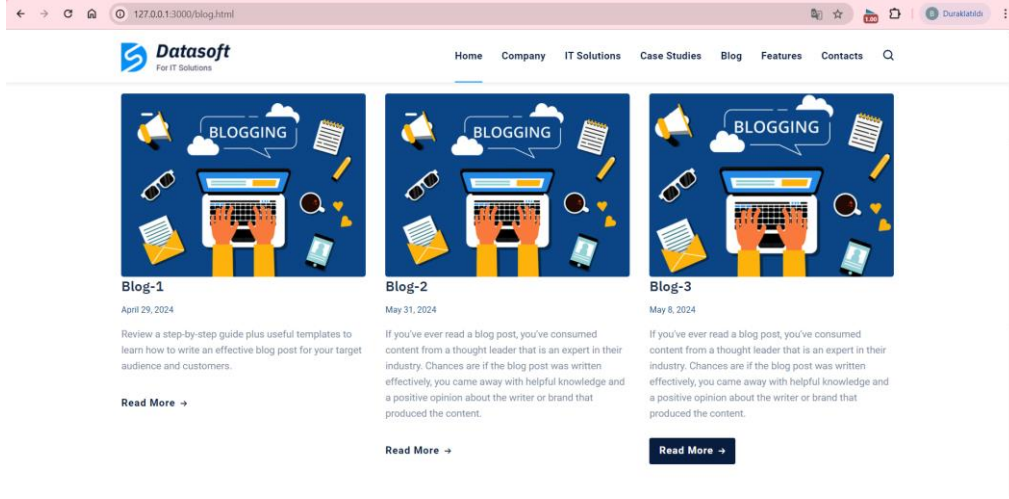
- Selenium’da xpath,name,ID,CLASS_NAME,CSS_SELECTOR kullanımını öğrendim.
- Docker compose kullanımını öğrendim.
- Projede blog-industry ekleme sayfalarında kullanıcının yükleyeceği resimleri docker ile nasıl gösterebileceğimi öğrendim. (dockerfile dosyası oluşturmayı öğrendim.)

3. Ödevde Yaptıklarım

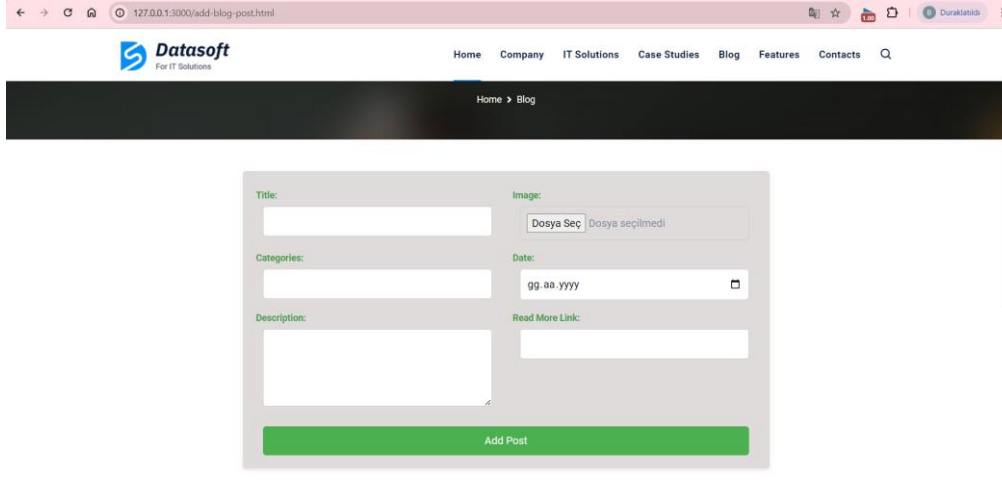
- Sabis dokümanlardaki tüm html sayfaları kullanılarak django web projesi yapılmıştır. Template’de login-register alanları popup olduğu için Login.html ve Register.html olarak iki sayfa tasarlanmıştır. Bu sayfaların header’ı ve footer’ı da template’den alınmıştır.
- Blog ekleme ve industries ekleme sayfaları için add-industries.html ve add-blog-post.html sayfaları eklenmiş ve bu sayfalarda form oluşturulmuştur. Oluşturulan blog ve industries’lerin tek açma kısımları veritabanındaki id’ler kullanılarak gerçekleştirilmiştir. Veritabanı olarak postgresQL ve testdb isimli database kullanılmıştır.
- Kullanıcılar blog ve industries’lere yorum yapabilmektedir.
- Kullanıcının eklediği blog ve industries resimleri/file’ları için sitenin docker ile ayağa kaldırıldığında gözükmesi için Dockerfile dosyasında düzenlemeler yapılmıştır. settings.py dosyasında MEDIA_ROOT VE MEDIA_URL değişkenleri tanımlanmıştır.
- Web kısmı tamamlandığında selenium ile bu web projesi için 30 farklı test gerçekleştirilmiştir. Bu testler 13 kategoriye ayrılmıştır. Örneğin form kontrol testlerinde blog ekleme ve form ekleme işlemlerinin doğru bir şekilde gerçekleşip gerçekleşmediği test edilmiştir. Buton rengi,boyutu,sayfa başlığı içeriğinin doğru olması gibi imaj kontrol testleri gerçekleştirilmiştir. Popup testlerinde video ve search butonlarının tıklanabilirliği ve popup’ın başarıyla açılıp kapanabildiği test edilmiştir. Testlerde 29 farklı XPATH kullanılmıştır.

Aşağıda web projesinde blogların listelendiği ve blog ekleme işleminin gerçekleştiği form sayfası görülmektedir:

Read more diyerek ilgili blogu tek açma işlemi gerçekleşmektedir ve kullanıcı ilgili blogu yorum yapabilmektedir.



Bu sayfada kullanıcı blog ekleme işlemlerini gerçekleştirmektedir:



4. Eksik Bırakılan Yerler

- Projede eksik bir yer bırakılmamıştır.

5. Zorlandığım Kısımlar

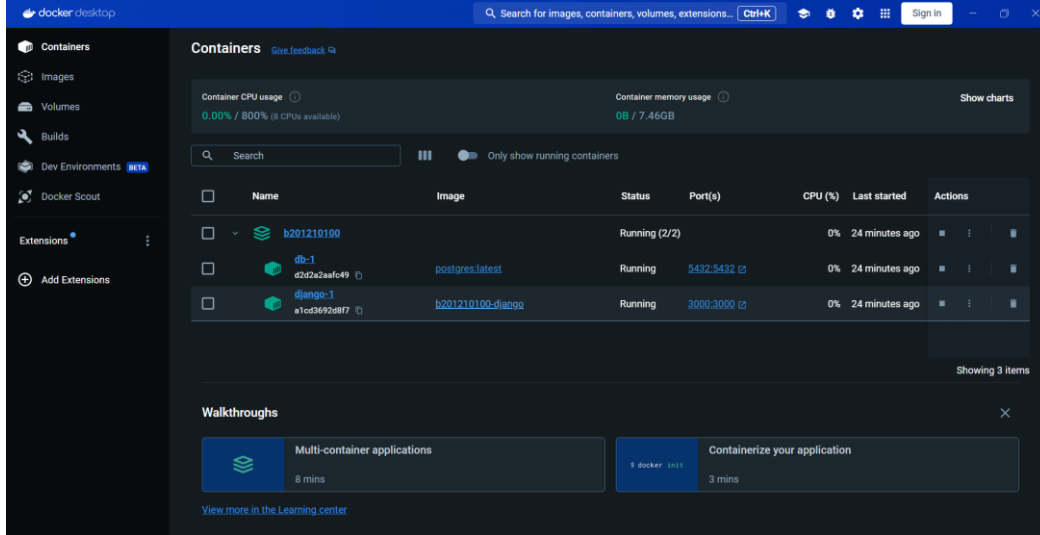
- 30 farklı test senaryosu üretmeye çalışırken zorlandım.
- Docker ile siteyi ayağa kaldırırken aynı zamanda resimleri docker ile sayfada gösterirken zorlandım.
- Template css'lerini göstermekte zorlandım. Daha sonra import hatalarından kaynaklı olduğunu gördüm.
- Blog ve Industries sayfalarını tek açma kısmında zorlandım.

6. Projenin Çalıştırılması

- yazılımTesti_2\test_selenium\b201210100> dizini altında **docker-compose up** veya **docker-compose up -d** diyerek web projesi docker ile ayağa kaldırılmalıdır.

```
PS C:\Users\beyzanur\Desktop\yazılımTesti_2\test_selenium\b201210100> docker-compose up
time="2024-05-19T13:17:27+03:00" level=warning msg="C:\\Users\\beyzanur\\Desktop\\yazılımTesti_2\\test_selenium\\b201210100\\docker-com
pose.yml: 'version' is obsolete"
[+] Running 2/0
 ✓ Container b201210100-db-1 Created
0.0s
 ✓ Container b201210100-django-1 Created
0.0s
Attaching to db-1, django-1
```

Aşağıda docker ile database ve django servislerinin farklı container'larda çalıştırılmış olduğunu görüyoruz:



- yazılımTesti_2\test_selenium> dizini altında **python test.py** diyerek testler çalıştırılmalıdır. Menüden seçim yapılarak testler başlatılmalıdır.

Örneğin testlerin python konsol uygulaması olarak çalıştırılmış hali:



Testler ile ilgili Önemli Not:

- Bazı testlerde blog sayfasında veya industries sayfasında read more dedikten sonra ilgili blog'u /industries'i tek açabiliyor mu test edilmektedir. Bu işlemlerin test edilebilmesi için ise en az bir tane blog veya industries olması gerekmektedir. Eğer hiç blog/industries eklemeyen test yapmaya çalışılırsa "Test başarısız. Hiç blog gönderisi olmadığı için yorum yapma testi başarısız olmuştur. Lütfen önce blog ekleyin." Uyarısı verilecektir. Testlerin başarılı olabilmesi için lütfen önce blog-industries ekleyin.
- Login testlerinde ise başarılı veya başarısız şifre denemeleri test edilmiştir. test_true_login() ve test_false_login() testlerinin başarılı olması için "beyz" username'li ve "123" password'lu bilgiler ile lütfen kayıt olun. Yoksa testler başarısız olacaktır.
- Comment kontrol testlerinde yorum yapabilmek için en az bir blog/industries sayfası olmalıdır ki yorum testi gerçekleşsin. Bu nedenle bu testler hiç blog/industries eklemeyen gerçekleşirse "Test başarısız. Hiç blog gönderisi olmadığı için yorum yapma testi başarısız olmuştur. Lütfen önce blog ekleyin." Uyarısı verilecektir.