



**T.C.**  
**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

YAZILIM TESTİ 2.ÖDEV RAPORU

**B201210084 - Abdulkadir BAŞTUĞ**

**SAKARYA**

Yazılım Testi Dersi

Dr.Öğr.Üyesi MUHAMMED FATİH ADAK

## Özet

Bu projede docker üzerinden ayağa kaldırılabilen; login,register, blog yazısı , industry ekleme listeleme ve tek açma işlemlerini yapan bir django web projesinin yapılması ve bu proje için selenium kütüphanesi kullanılarak en az 30 adet test yazılması istenmiştir. Bu 30 test çeşitli kategorilere ayrılıp python konsol uygulamasında ekrana gelecek menü yardımıyla çalıştırılması istenmiştir.

## Projede Yapılanlar

İlk olarak gerekli veritabanı ayarları static template dosyalarının oluşturulması ve yollarının belirtilmesi gibi kurulum aşamaları gerçekleştirilmiştir. Tasarım için dokümanlarda gönderilen dosyalar kullanıldı. Her sitede belirli bir şablon olması üzerine slider section sabit tutuldu ve sayfanın içeriğine göre diğer kısımlar dokümanlarda gönderilen dosyalardaki form, input, button vs kullanılarak düzenlendi.

Veritabanı için postgresql kullanıldı. Kullanıcı girişleri için yeni model oluşturulmadı. Sadece default app üzerinden migrate edildi. Register formundan alınan bilgiler viewda yeni user oluşturmak için kullanıldı. Böylece login formundaki bilgilerle databasede yeni oluşturulan user eşleşmeleri sağlandığında siteye giriş yapılabilir hale geldi. Siteye giriş yapmadan sitenin diğer sayfalara erişimi her view üzerinden denetlendi ve eğer kullanıcı girişi yapılmadan siteye direkt erişilmek istendiğinde login sayfasına redirect işlemi gerçekleştirildi.

Blog ve industry kayıt işlemleri için blog ve industry kayıtlarını veritabanında tutabilmek için model oluşturuldu ve migrate edilip veritabanında yeni tablolar oluşturuldu. Kayıt için ayrı 2 siteye yönlendirip formlar aracılığıyla siteden alınan bilgiler viewda gerekli kontroller sağlanıp veritabanına kaydedildi.

Veritabanına kaydedilen blog ve industry lerin tamamını tek bir sayfada listeleyebilmek için viewda veritabanındaki tüm nesneler çekilip bir nesne ile html sayfasına gönderildi ve html sayfasında da for döngüsüyle hepsi listelendi. Tek tek erişebilmek için ise read more kısımları eklendi ve farklı bir html sayfasına yönlendirme yapılarak hangi id ye sahip nesne için read more butonuna tıklandıysa o id için singlepost veya singleindustry sayfalarında daha detaylı bir content ile tek sayfada gösterildi.

Logout işlemi için sağ üste bir buton yerleştirildi ve tıklandığında gerekli çıkış işlemi view ile sağlanıp tekrardan login sayfasına redirect işlemi gerçekleştirildi.

Kullanıcı oturumu açık unutursa diye sesion\_cookie\_age 300 olarak yani 5 dk süre içinde otomatik olarak çıkış yapacak şekilde ayarlandı.

Login testlerinde olması gereken kullanıcının giriş yapıp yapılmadığı, yanlış bilgilerle giriş, show-password un işlevselliğinin ve kullanıcı giriş yaptıktan sonra kullanıcı adının anasayfada gözükmesinin testi yapılmıştır.

Navigasyon testlerinde setUp da kullanıcı girişi yaptırılmış ve ulaşabilmesi gereken tüm sayfalara ulaşip ulaşamadığının testleri yapılmıştır.

Kullanıcı kayıt testlerinde random kullanıcı adı email üretilerek başarılı kayıt işlemi, aynı kullanıcı adının daha önceden kaydı varsa hata vermesi, aynı emailin daha önce kaydı varsa hata vermesi ve email inputunun email formatından başka bir formatta input girildiğinde hata vermesi testleri gerçekleştirilmiştir.

Blog ve industry kayıt testlerinde random image ve random başlıklar üretilip kayıtlarının başarıyla gerçekleştiğinin, daha önceden var olan bir name veya title ile kaydedilme çalıştığı zaman ise hata vermesinin testleri yapılmıştır. Aynı veriyi 2 kez kaydetmek için xpath kullanılmıştır.

Session testlerinde oturum süresinin 5 dk olmasını ve logout butonunun doğru çalışmış çalışmadığının testi gerçekleştirildi.

Security testlerinde ise login yapılmadan url kısmına veya sitedeki linklere tıklayarak erişmemesi gereken sayfalara erişemediğini tekrardan login sayfasına yönlendirildiğinin testleri xpath kullanılarak yapılmıştır.

**Önemli Not:** Projenin dockerda ayağa kaldırılabilmesin için settings.py dosyasında database host db yapılmalıdır ve buildden sonra ayağa kaldırmadan migrate yapılmalıdır ve daha sonra docker-compose up –build ile ayağa kaldırılır. Ayağa kaldırdıktan sonra 127.0.0.1:8000 adresinden projeye erişilebilir. Dockerda kullanıcı adı kanla, şifresi 1234 olan kullanıcı kayıt işlemi gerçekleştirilmiş ve giriş testi başarıyla sonuçlanmıştır.

#### Test Caseler

kullanıcı adı:kanla,

mail:kadir5538@gmail.com

şifre:123456789a olan bir kullanıcının var olduğu varsayılarak çalıştırılmalıdır. Ayrıca session testi 5 dk sürdüğünden istenirse test\_session.py daki dosyada süre ve settings.py daki süre 1 dk ya düşürülerek test daha hızlı gerçekleştirilebilir.

Image testleri local tempde random resimler oluşturuyor. Bu dosyalara ulaşmak için Run içine %temp% yazılabilir.