

YAZLAB - PROJE 1

Web Scrapping - E-Ticaret

1st Melih Selami URKMEZ
Kocaeli University
Computer Engineering
Kocaeli,Turkey
200202010@kocaeli.edu.tr

2nd Taha PEK
Kocaeli University
Computer Engineering
Kocaeli,Turkey
200202046@kocaeli.edu.tr

Abstract—Projede 2 adet websitesi istenmektedir. Bir sitede scraping ile elde edilen veriler birbirleriyle eşleştirilerek gösterilmektedir. Ve bu sitede kategoriye göre filtreleme, arama barı ile ürün arama veya ilgili fonksiyonların bulunması gerekir. Diğer sitenin ise bir e-ticaret sitesi olması beklenmektedir. Ayrıca bu diğer e-ticaret sitesindeki veriler scraping ile çekilip ilk websitesinde gösterilmelidir. Ayrıca bu diğer websitesinde kategoriye göre ürün filtreleme veya arama barı ile ürün arama gibi fonksiyonların da bulunması gerekir.

I. GİRİŞ

Projede öncelikle scraping(kazma) yöntemi ile 4 adet siteden veri çekmemiz istenmektedir. Bu veriler laptop olarak filtrelenmiştir. Yani sadece ilgili sitelerden laptoplara ait bilgileri çekmemiz gerektiği belirtilmiştir. Bu scraping ile çekilen verilerin veritabanında tutulması gerektiği de söylenmiştir.

Totalde projede 2 adet websitesi istenmektedir. 1. website genel olarak aynı ürünlerin farklı sitelerdeki karşılıklarını birbirleriyle eşleştirip göstermektir. Bu sitede kategoriye göre filtreleme,dinamik arama,sıralama becerisini geliştirme gibi fonksiyonların bulunması istenmektedir. 2. website ise bir e-ticaret sitesi olmalıdır. Yani aynı ürünün birden fazla kez bulunmayacağı ve tek bir siteye ait verilerin bulunduğu e-ticaret sitesi olmalıdır. Ayrıca diğer sitede olduğu gibi bu sitede de kategoriye göre filtreleme , dinamik arama, sıralama becerisini geliştirme gibi fonksiyonların bulunması gerektiği söylenmiştir.

Veritabanı olarak isteyen NOSQL yapıda bir veritabanı isteyen ise MYSQL yapıda bir veritabanı kullanılabileceği belirtilmiştir.

Eğer MYSQL yapıda bir veritabanı kullanılacaksa normalizasyon kurallarına dikkat edilmesi gerekir

Web Scrapping Nedir?

Web kazıma, metin tabanlı biçimlendirme dillerinin (XHTML, HTML, Markdown gibi) taranması, web sunucuları ve/veya uygulamalar arasındaki veri akışının (JSON, XML, YAML vb.) dinlenmesi ve verilerin bu dosyalardan kazınması işlemlerini ifade eder. Bu amaçla pek çok araç ve programlama dili geliştirmeleri mevcuttur.

NOSQL Nedir?

NoSQL terimi, ilişkisel olmayan veritabanı türlerini ifade eder ve bu veritabanları verileri ilişkisel tablolardan farklı bir formatta depolar.NoSQL veritabanları, belirli veri modelleri için özel olarak tasarlanmıştır ve modern uygulamalar oluşturmaya yönelik esnek şemalara sahiptir. NoSQL veritabanları uygun ölçekte geliştirme kolaylığı, işlevselliği ve performansıyla geniş çaplı olarak kabul görmüştür.

Biz projemizde MongoDB olarak bilinen NOSQL yapıda bir veritabanı kullandık. Ayrıca sürekli database'den işlem yapmamak için bilgileri cache'den daha hızlı yakalamak için bilgileri 15 saniye timeout ile Redis'te tuttuk.

Cache Nedir?

Cache, yani önbellek, yüklediğiniz web sayfasının HTML sayfalarını, içerisindeki görselleri ve dokümanları geçici olarak belleğe alır. Böylece bu sayfa yeniden ziyaret edildiğinde daha az bant genişliği kullanılır ve server'a da daha az istek gönderilir. Bu durum, kullanıcı deneyimini iyileştirir.

Ayrıca veritabanını ve cache sistemini ortak olarak kullanmak için ABVAG sunucumuz üzerinden MongoDB ve Redis için docker container oluşturduk. Devamında bunu kendi localhost'umuza port forward ederek sunucu portunu kendi localimize aktararak uzak sunucuya bağlandık.Mimariyi tasarlarlarken mikroservis mimarisini ele almaya çalıştık.

```
melih@kouabvagi:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
b1ad856261f8	redis	"docker-entrypoint.s..."	5 days ago	Up 5 days	0.0.0.0:6379->6379
cb1ad3da8079	mongo:4.4.6	"docker-entrypoint.s..."	2 weeks ago	Up 2 weeks	0.0.0.0:27017->27017

Fig. 1. Sunucu üzerinde çalışan Redis ve MongoDB

Django Nedir?

Python ile yazılmış özgür ve açık kaynak bir web uygulama iskeletidir (framework). Bir web iskeleti, websitesi geliştirmeyi hızlandıran ve kolaylaştıran bir grup bileşendir.

II. TEMEL BİLGİLER

Projenin scraping kısmında 'Python' kullanılmıştır. Web kısmında ise Python'ın Django frameworkü kullanılmıştır. Tasarım için HTML,CSS,Javascript ve

Bootstrap kullanılmıştır. Database olarak NOSQL veritabanı bir MongoDB kullanılmıştır. Cacheleme için ise Redis kullanılmıştır. Ayrıca MongoDB'ye bağlanmak için Mongo'nun bize sunduğu MongoDB Compass client'ı kullanılmıştır.

Projenin yazılım sürecinde git versiyon kontrol sistemi olarak GitHub kullanılmış olup, kodlarımızın gizliliği için Github Education hesabımızın bize sağladığı private repository içerisinde geliştirilmiştir.

III. YÖNTEM

Proje mimarisi en optimum özelliklerle çalışacak şekilde tasarlanmaya çalışılmıştır. Proje tasarlanırken artık her yeni projenin de aktarılmaya çalışıldığı mikroservis mimarisi kullanılmıştır. Böylece site üzerinden gelen isteklere göre cluster üzerinde scale edilebilir bir proje haline getirilmiştir.

Ve django üzerindeki tasarımlar da en az kod tekrarı oluşacak şekilde tasarlanmıştır. Örnek verecek olursak her sayfada olması gereken footer ve navbar genel bir yapıya alınıp her sayfa içerisinde 1 kez include edilmiştir. Böylece 100'lerce satır kod tekrarıdan kurtulunmuştur.

A. Proje Hiyerarşisi

- Backend-Python

Backend-Python klasörü içerisinde 6 adet siteden scraping yapmayı sağlayan kod bulunmaktadır. Scraping yaparken kullandığımız siteleri özetleyecek olursam ; Çiçeksepeti, Teknosa, Hepsiburada, N11, Trendyol ve kendi oluşturduğumuz e-ticaret sitesidir. Bu sitelerden scraping yaparak 'WebScraping' adlı database içerisindeki allproducts kolonuna bilgileri yazmayı sağlar.

- EcommerceWeb

EcommerceWeb klasörü içerisinde oluşturduğumuz e-ticaret sitesinin kodları bulunmaktadır. Bu klasör içerisinde ariyeten 3 adet app klasörü daha bulunmaktadır. Bunlar authentication, crud ve ecommerce'dir.

Authentication app'inin içerisinde giriş, çıkış ve kayıt olma işlemlerini kontrol eden fonksiyonlar bulunmaktadır.

Crud app'inin içerisinde ise ürün güncelleme, ürün silme veya ürün ekleme gibi işlevlerin sağlandığı fonksiyonlar bulunmaktadır.

Ecommerce app'inin içerisinde ise e-ticaret sitesine ait kodlar bulunmaktadır.

- ShoppingApp

ShoppingApp klasörü içerisinde oluşturduğumuz cimeriye benzeyen siteye ait kodlar bulunmaktadır. Ayrıca bu klasör içerisinde 1 adet shopping adında app bulunmaktadır. Bu siteye ait tüm bilgiler ve tanımlamalar bu shopping içerisinde yer alır. Ayrıca klasör içerisindeki

kategoriler.txt dosyası da dinamik olarak doldurulmaktadır. Autocomplete yapıda kullanılır.

IV. PSEUDO KOD(YALANCI KOD)

- 1) Scraping için siteleri tanımla.
- 2) Sitelerden laptop ürünleri al ve listeye at.
- 3) Sitelerden bilgileri çekerken aynı yapıda olmaları için ilgili bilgileri shaperlere at.
- 4) Model numarası vermeyen sitelerden title'da arama yaparak model numarasını al.
- 3) Sitelerden scrap etme işlemi bittikten sonra database'e bağlan.
- 4) Listeye yazdığın verileri kontrollü bir şekilde database'e ekle.
- 5) Eklerken eğer aynı ürünü eklemeye çalışırsa eski ürünü sil yeni ürünü ekle.
- 6) Database'e ekleme işlemi sonlandıktan sonra backend-python kodunu sonlandır.
- 7) Ssh ile sunucuyu çalıştır.

- ShoppingApp Adımları

8) 127.0.0.1:8080 portundan ShoppingApp uygulamasını çalıştır.

9) Home path'ini görüntüle.

10) Eğer en yüksek fiyata göre seçilirse;

- Eşlenmiş olarak bulunan database'deki verileri büyükten küçüğe .orderby(-fiyat) methodu ile sırala.

11) Eğer en düşük fiyata göre seçilirse;

- Eşlenmiş olarak bulunan database'deki verileri küçükten büyüğe .orderby(fiyat) methodu ile sırala.

12) Eğer en yüksek puana göre seçilirse;

- Eşlenmiş olarak bulunan database'deki verilerin puan bilgisini büyükten küçüğe .orderby(-puani) methodu ile sırala.

13) Eğer kullanıcı kategori seçmişse seçilen kategorileri al.

- İlgili kategoriler ile sorgu yarat.

- Yaratılan sorguyu database'de çalıştırarak bulunan ürünleri listele.

14) Eğer kullanıcı arama çubuğundan arama yaparsa;

- İlgili kelimeler için öneride bulun.

- Kullanıcının yazdığı bilgiye göre sorgu yaratarak database'de çalıştır. Geri dönen cevabı ekrana bastır.

15) Eğer kullanıcı ürün title'ına basarsa onu ürün detay bilgilerinin olduğu sayfaya yönlendir.

16) Eğer kullanıcı ilgili sitenin icon'una basarsa onu ilgili siteye yönlendir.

- EcommerceWeb Adımları

17) 127.0.0.1:8000 portundan EcommerceWeb uygulamasını çalıştır. 18) Home path'ini görüntüle. 19) Eğer en yüksek fiyata göre seçilirse;

- Eşlenmiş olarak bulunan database'deki verileri büyükten küçüğe .orderby(-fiyat) methodu ile sırala.

20) Eğer en düşük fiyata göre seçilirse;

- Eşlenmiş olarak bulunan database'deki verileri küçükten

büyükçe .orderby(fiyat) methodu ile sırala.

21)Eğer en yüksek puana göre seçilirse;

- Eşlenmiş olarak bulunan databasedeki verilerin puan bilgisini büyükten küçüğe .orderby(-puani) methodu ile sırala.

22)Eğer kullanıcı kategori seçmişse seçilen kategorileri al.

- İlgili kategoriler ile sorgu yarat.

- Yaratılan sorguyu database'de çalıştırarak bulunan ürünleri listele.

23)Eğer kullanıcı arama çubuğundan arama yaparsa;

- İlgili kelimeler için öneride bulun.

- Kullanıcının yazdığı bilgiye göre sorgu yaratarak database'de çalıştır. Geri dönen cevabı ekrana bastır.

24) Eğer kullanıcı ürün title'ına basarsa onu ürün detay bilgilerinin olduğu sayfaya yönlendir.

25) Eğer kullanıcı giriş yapmamışsa ve staff yetkili bir hesapla giriş yaparsa ürün ekleme,çıkarma ve güncellemeyi aktif et.

26) Eğer kullanıcı giriş yapmışsa ve staff yetkisi olmayan bir hesapla giriş yaparsa sadece ürünleri görüntüle.

27) Eğer kullanıcı kayıt olmak istiyorsa kayıt et. Ettikten sonra ilgili mail'e teşekkür mesajı at.

28) Eğer kullanıcı bilgide güncelleme yaparsa bilgiyi hem ecomemrce database'inde hem de shoppingapp databaseinde güncelle.

29) Eğer kullanıcı ürünü silerse ürünü hem ecommerce databaseinden hem de shoppingapp databaseinden sil.

30) Eğer kullanıcı ürün eklerse ürünü shaperlara sokarak öncelikle database'de var olup olmadığını kontrol et.

31) Var olup olmadığını model numarası üzerinden kontrol et.

-Eğer ürün varsa ürünün var olduğuna dair log'u terminalden döndür.

-Eğer ürün yoksa ürünü database'e ekle.

V. SONUÇ

Bu projede birçok kazanım elde ettik. Ayrıca birçok sorunla karşılaştık. Bu durumun bizi oldukça geliştirdiğini öğrendik. Elde ettiğimiz kazanımlardan bahsedecek olursak;

- NOSQL veritabanı yapısını öğrenmek
 - Django framework'ü ile çalışmak
 - MongoDB ile çalışmak
 - Dinamik bir websitesi oluşturmak
 - Git sürümleri - GitHub Kullanımı
 - HTML.CSS,Javascript,Bootstrap kullanmayı pekiştirmek,
 - Docker mimarisini araştırmak,öğrenmek,kullanmak,
- Kazanımlarını elde ettim.

VI. KAYNAKÇA

- <https://www.hosting.com.tr/blog/django/>
- <https://sadikturan.com/>
- <https://medium.com/featurepreneur/how-to-connect-a-django-project-with-mongodb-99f556cbb5b7>
- <https://www.w3schools.blog/mongodb-tutorial>

- <https://www.youtube.com/watch?v=hbx39adciac>
- <https://stackoverflow.com/questions/37961413/django-import-model-from-another-project>
- <https://realpython.com/beautiful-soup-web-scraper-python/>
- <https://oxylabs.io/blog/python-web-scraping>

VII. DENEYSEL SONUÇLAR

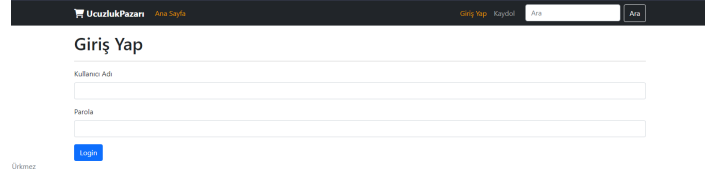


Fig. 2. Giriş Yap Sayfası

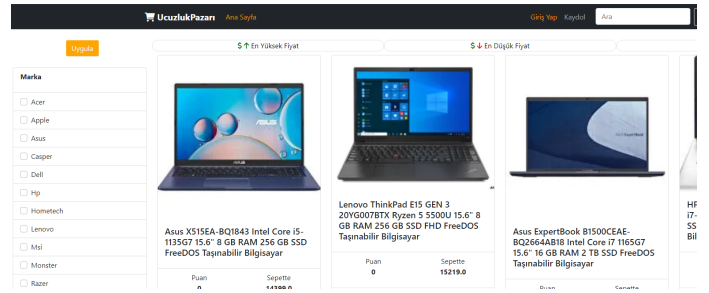


Fig. 3. Ecommerce Home Sayfası

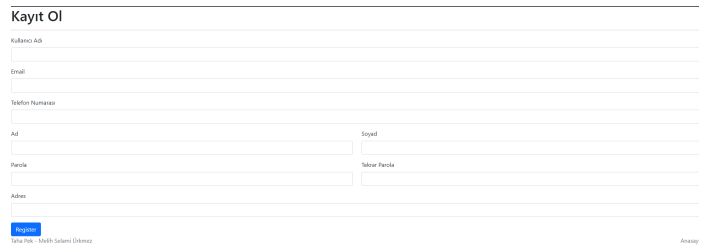


Fig. 4. Ecommerce Kayıt Ol Sayfası

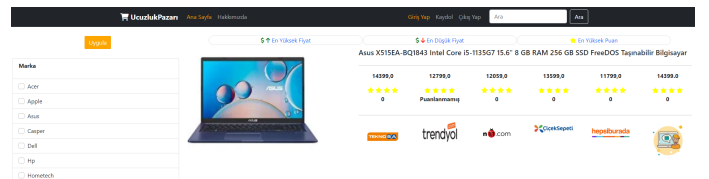


Fig. 5. ShoppingApp Anasayfası

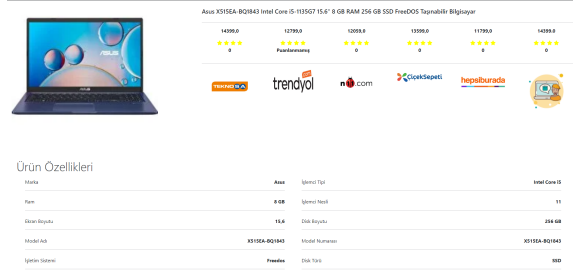


Fig. 6. ShoppingApp Detay Sayfası

UcuzlukPazari [Asus Sayfa](#) Ürün Sil Ürün Ekle Çıkış Yap Hoşgeldin,

Add Product

Marka

Model Adı

Model No

İşletim Sistemi

İşlemci Tipi

İşlemci Hızı

Ram

Disk Boyutu

Disk Türü

Fig. 7. Ürün Ekleme Sayfası

Delete Product

Model No

Fig. 8. Ürün Güncelleme Sayfası

UcuzlukPazari [Asus Sayfa](#) Ürün Sil Ürün Ekle Çıkış Yap Hoşgeldin,

Update Product

Marka

Model Adı

Model No

İşletim Sistemi

İşlemci Tipi

İşlemci Hızı

Ram

Disk Boyutu

Fig. 9. Ürün Silme Sayfası

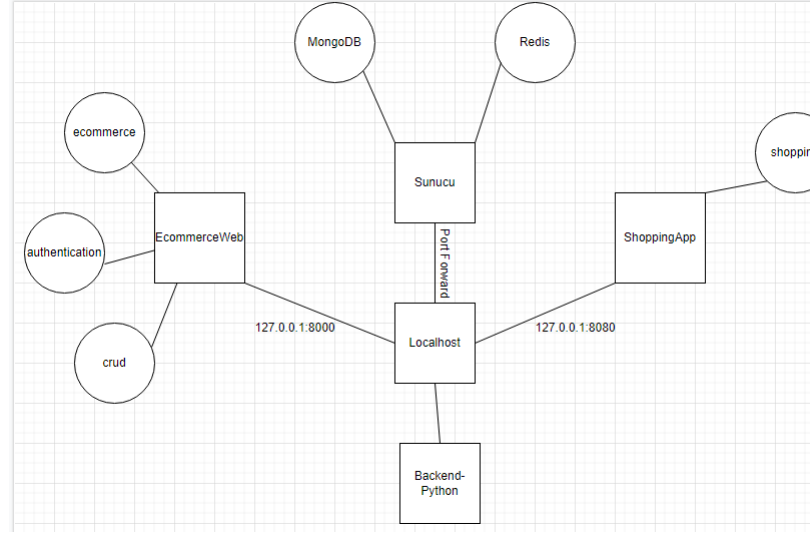


Fig. 10. Projenin Mimari Görüntüsü