

Web Sitesi Uygulaması

Umut SÜTCÜ

umutsutcou@outlook.com

ÖZET

Bu doküman yazılım laboratuvarı 1 dersinin 1.projesi olan Web scraping ile farklı alışveriş sitelerinden istenilen ürün hakkında bilgi elde etmek. Bu elde edilen bilgilerle bir veritabanı oluşturmak. Oluşturulan veritabanındaki bilgilerin filtrelenmesi ve sıralanması sağlamak. Bu veritabanındaki verilerle bir e-ticaret sitesi oluşturmak ve cimri akakçe bir karşılaştırma sitesi oluşturmak.

PROJENİN TANIMI

Web Scraping: 4 adet E-Ticaret (Trendyol, Hepsiburada, n11, Amazon) sitesi üzerinden web scraping kullanılarak bilgisayar(notebook) kategorisine ait bilgiler alınacaktır. Ayrıca E-ticaret sitelerine benzer yapıda kendiniz bir web sitesi oluşturacak (sadece bilgisayar(notebook) kategorisi olacak) ve web scraping kullanarak sitedeki bilgiler alınacaktır. Web Scraping kullanılırken siteye ait html bilgiler veya siteye request isteği ile bilgiler alınmalıdır.

Veritabanı: Web scraping ile elde edilen veriler veritabanına kaydedilecektir. Kullanılacak veritabanı proje uygun olarak herhangi bir tanesini kullanabilirsiniz (MongoDB gibi veri tabanları kullanılması artı puan verilecektir.) Veritabanında aşağıdaki tabloda belirtilen özellik bilgilerinin kullanılması gerekmektedir. Veritabanındaki tabloların normalizasyon kurallarına uygun olacak şekilde tasarlanması ve organize edilmelidir.

Web Sayfası: Farklı sitelerden çekilen ürün bilgilerinin gösterilmesi için bir web sayfası oluşturmanız beklenmektedir. Örnek web sayfası aşağıdaki resimde görülmektedir. Bu sayfa ilk açıldığında veritabanında yer alan tüm kayıtlar listelenmelidir. Listeleme işleminde ürün bilgilerinin yer aldığı ana başlık bulunmalıdır. Herhangi bir ürün başlığına tıklandığında ürün hakkındaki bilgilerin (fiyat ve ürün özellikleri) olduğu sayfaya yönlendirilecektir.. Listeleme işleminde ürüne ait en düşük 3 fiyat ve bunların ait olduğu E- ticaret sitelerinin linkleri isim ve logolarıyla birlikte yer alacaktır. İsim veya logonun üzerine tıklayınca ürünün bulunduğu E-ticaret sitesinin sayfası açılacaktır. Web sayfasından herhangi bir ürünle ilgili dinamik arama işlemi yapılabilir. Bu arama alanında bir ürünün isim, model ya da bulunduğu E-ticaret sitesine göre yapılmalıdır. Ayrıca Arama sırasında yazım yanlış olması durumunda sistem düzeltilmiş öneride bulunmalıdır. Örnek: hepsi burada – yazımı düzeltilmiş sonuçları görüyorsunuz:

Hepsiburada şeklinde olmalıdır. Web sayfasında dinamik filtreleme işlemi de ayrıca yer almalıdır. Filtreleme işlemi veritabanında yer alan ürünün tüm özellikleri için uygulanabilir olmalıdır. Yukarıdaki görselde e-ticaret sitelerinde bulunan laptop ürünleri markasına göre örnek olarak filtrelenmiştir. Web sayfasında en düşük fiyattan en yüksek fiyata, en yüksek fiyattan en düşük fiyata ve en yüksek puandan en düşük fiyata şeklinde sıralama işlemleri yapılabilir. Laptop bilgilerinin yer aldığı alanda fiyat bilgisi için de aynı durum geçerli olacaktır.

Duplicate : Scraping ile çekilen bilgiler güncellenmesi gerekmektedir. Güncel bilgiler veritabanında kaydedilmelidir. Kaydedilen veriler ile önceki kayıtlı veriler benzer bilgiler içermektedir. Bundan dolayı Duplicate oluşmaması için veritabanında kontrol işlemi yapılmalı ve veritabanı ile web sayfasında sadece güncel bilgiler yer almalıdır. Veritabanında Near Duplicate kontrolü yapılacaktır. Veritabanında Benzer isimlerde kaydedilmiş birden fazla kayıt olabilir. Bu durum aynı isimlerde farklı kayıtlar olduğu anlamına gelmektedir. Örneğin bir web sitesinden ürünün işletim sistemi bilgisi Dos olarak diğerinden ise FreeDos olarak kayıt edildiğini düşünelim. Bu iki durum aynı bilgiyi temsil etmektedir. Bundan dolayı Dos olamı diğer unique bir etiket altında temsil edilmelidir.

Kendinize ait bir E-ticaret sitesi oluşturmalısınız ve bu site Admin tarafından güncellenebilir olmalıdır. Sitede yapılacak değişikliklerde (Bir notebook ürünün fiyatının değiştirilmesi Notebook ürünü ait bir kaydın kaldırılması Yeni notebook ürünün eklenmesi Notebook ürünün puanında değişiklik yapılması Ürün bilgilerinin güncellenmesi) bilgilerin web scraping ile çekilmesi gerekmektedir. Alınan tüm bu değişiklikler farklı sitelerdeki ürünlerin listelendiği diğer web sitesinde anlık olarak gösterilmelidir. Ürün bilgisinin gösterildiği web sayfasında tüm notebook ürünlerinin listelenmesi ve ürün bilgilerinin yer aldığı ana başlıklarının oluşturulması gerekmektedir. Bu başlığa tıklayınca ürün hakkında bilgilerin (fiyat bilgileri dahil) olduğu sayfa açılmalıdır. Ayrıca ürünlerin listelendiği sayfadaki E-ticaret linki üzerine tıklayınca ürünün bu E-ticaret sitesindeki sayfasına kullanıcı yönlendirilmelidir. a) Dinamik arama ile (yukarıda istenilen tüm arama kriterleri dahil olacak şekilde) Asus .vb modele göre arama işleminin yapılması b) Dinamik arama ile (yukarıda istenilen tüm arama kriterleri dahil olacak şekilde) hepsiburada .vb E-ticaret sitelerinin aranması a) Ürün bilgilerinin gösterildiği web say-

fası üzerinde Filtreleme işleminin yapılması b) Fiyat bilgisine göre küçükten büyüğe veya büyüktten küçüğü sıralama ayrıca puanı yüksek olan ürüne göre sıralama işleminin yapılması Sizin oluşturacağınız E- Ticaret sitesine kayıtlı olan bir ürünün aynı bilgileri ile kayıt edilmeli ve veritabanında Duplicate Kontrolü yapılmalıdır. Sizin oluşturacağınız E- Ticaret sitesinde iki farklı kayıt işleminin yapılması ve Near Duplicate kontrolünün yapılması

ARAŞTIRMA VE YÖNTEMLER

İlk önceliğim olarak e-ticaret sitelerinden veri çekebilmek için python kütüphanelerini araştırmak oldu. BeautifulSoup kütüphanesinin proje için çok fazla kaynağı olduğunu fark ettim. İnternet sitelerinden veri çekebilmek için requests kütüphanesinden faydalanadım. Çektiğim verileri düzgün bir şekilde parçalamak için biraz html bilgimin olmasını gerektiğini fark ettim. Bunun üzerine hızlı bir şekilde html baktım. Daha sonra çektiğim verileri atacağım bir database baktım. Önerilen databaselerden mongodbyi seçtim. Bunun için mongodbnin nasıl kullanıldığını ve NoSQL mantığını biraz araştırdım. Verileri düzgün bir şekilde database e attıktan sonra duplicate kontrollerini gerçekleştirdim.

Bunlardan sonra websitesi yapımı için pythonun django framework'ünü kullandım. Projede en çok zorlandığım yer de bu oldu. Websitesi yapmak için htmlden biraz da olsa anlamak gerektiğini fark ettim. Bu beni çok zorladı. İnternetteki hazır kodları kullanarak siteyi biraz daha güzelleştirdim.

n11	289	173.12KB	614B	84KB	1	24KB	24KB
n11_v2	945	569KB	617B	368KB	1	80KB	80KB
n11_v3	412	249.57KB	621B	104KB	1	28KB	28KB
test2	32	17.37KB	556B	24KB	1	20KB	20KB
test_n11	965	590.71KB	627B	364KB	1	80KB	80KB
test_trendyol	984	542.49KB	565B	292KB	1	64KB	64KB
trendyol	280	155.26KB	568B	84KB	1	24KB	24KB
trendyol_v2	546	303.64KB	570B	260KB	1	60KB	60KB
trendyol_v3	456	252.35KB	567B	124KB	1	28KB	28KB
vatan	298	163.69KB	563B	76KB	1	24KB	24KB

Database'deki verilerin genel bir örneği

```
_id: 2
Marka: "Monster"
adres: "https://www.trendyol.com/monster/abra-a7-v13-2-3-intel-core-i5-11400h-"
isim: "Monster Abra A7 V13.2.3 Intel Core i5-11400h 16gb Ram 500gb Ssd 4gb Rt_"
fiyat: "19.499 TL"
Cpu: ""
Cpu_cekirdek: "6"
Disk_Tur: "SSD"
Bellek_Tur: "DDR4"
Ekran_Karti_Model: "Nvidia GeForce RTX 3050 Ti"
Bellek_Kapasite: "16 GB"
Cpu_Model: "Intel Core i5-11400H"
Disk_Kapasite: "500 GB"
Ekran_Boyut: "17,3 inç"
```

Database'deki verilerin alınış şekli

Sitelerden webscrapping çektiğim verileri daha sonra rahat bir şekilde kullanabilmek için düzgün bir şekilde detaylara ayırıp tek tek aldım. Bu verileri daha sonra websitemde düzgün bir şekilde kullandım.

```
siteId="https://www.vatanbilgisayar.com/notebook/page="
header_bilgileri{"User-Agent":"Mozilla/5.0 (Linux; Android 10; SM-A200U) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.4577.62 Mobile"}

for i in range(1,20):
    new_dict=dict()
    sayfacevabi=requests.get(new_headers+header_bilgileri)
    print(new)
    print(sayfacevabi.status_code)
    if sayfacevabi.status_code==200:
        htmlicerik=sayfacevabi.content
        soup=BeautifulSoup(htmlicerik,"html.parser")
        for icerik in soup.find_all("div",{"class":"product-list product-list-list-page"}):
            bilgisayarlinikicerik.find("a")
            yonilink-bilgisayarlinik.get("href")
            yonilink="https://www.vatanbilgisayar.com/"+yonilink
            url_lin_listesi.append(yonilink)
            cevap=requests.get(yonilink)
            print(yonilink)
            if cevap.status_code==200:
                yonicerik=cevap.content
                soup2=BeautifulSoup(yonicerik,"html.parser")
                for dom in soup2.find_all("h1",{"class":"product-list-product-name"}):
```

Webscrapping için kullandığım kodun bir kısmı

Webscrapping için beautifulsoup ve requests kütüphanelerinden çokça faydalandım. Yukarıdaki fotoğrafta da gösterildiği gibi verileri tek tek parçaladım.

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark"> ...
<div class="container mt-3">
  <div class="row">
    <div class="col-3"> ...
    </div>
    <div class="col-9">
      <div class="row">
        {% for kategori in kategoriler %}
          <div class="col-4">
            <div class="card" style="width: 100%;">
              
              <div class="card-body">
                <h5 class="card-title">{{kategori.Marka}}</h5>
                <p class="card-text">{{kategori.isim}}</p>
                <a href="{% url 'details' kategori.id %}" class="btn btn-primary">Ürün adresi</a>
              </div>
            </div>
          </div>
        {% endfor %}
      </div>
    </div>
  </div>
```

Websitesinin genel görüntüsünü için kullandığım html

Websitesi yapımında tecrübem ve html bilgim çok olmadığı için bootstrap sitesinden hazır templateler aldım bu templateleri kendi koduma göre uyarlayayıp kodumun içine ekledim

```
class laptop(models.Model):
    isim = models.TextField()
    Model = models.CharField(max_length=200)
    Marka = models.CharField(max_length=200)
    fiyat = models.CharField(max_length=200)
    Cpu = models.CharField(max_length=200)
    Cpu_cekirdek = models.CharField(max_length=200)
    Disk_Tur = models.CharField(max_length=200)
    Bellek_Tur = models.CharField(max_length=200)
    Ekran_Karti_Model = models.CharField(max_length=200)
    Cpu_Model = models.CharField(max_length=200)
    Disk_Kapasite = models.CharField(max_length=200)
    Ekran_Boyut = models.CharField(max_length=200)
    Bellek_Kapasite = models.TextField()
```

Admin sitesi için kullandığım bir veri ekleme için kullandığım kod

Admin paneli için djangonun hazır admin panelini kullandım. Hazır admin panelinin üzerine veri eklemek ve eklenen verileri mongodbye aktarmak için bir djongo kütüphanesini kullandım.

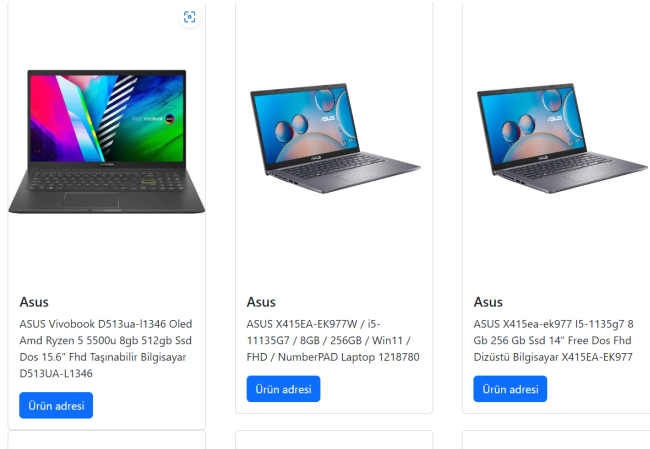
KOD BİLGİSİ(YALANCI KOD)

Kodu txt dosyası olarak ek olarak gönderdim.

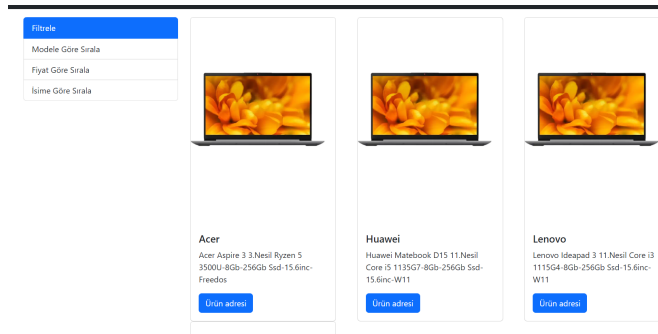
SONUÇ

Sonuç olarak python ile database bağlantıları nasıl yapılır. Veritabanında NoSQL mantığını kavradım. NoSQL veritabanlarında normalizasyon yapılarının nasıl yapıldığını kavradım. Daha sonra websitesi yapmak için python'un django framework'ünün nasıl kullanıldığını ve websitesi yapmak için html ve css kodlarının nasıl kullanıldığını öğrenmiş oldum. Websitesi ile veritabanının bağlantısının nasıl yapıldığını ve yapılan bağlantının dinamik kalmasını nasıl sağlanacağını görmüş oldum. Python ve django framework'ünün kullanımı üzerine ve NoSQL mantığını çalışmış olduk.

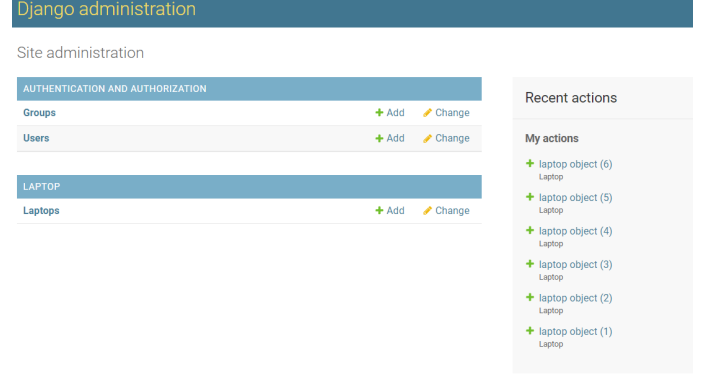
Aşağıda da projemize ait ekran görüntüleri:



Cimri benzeri sitemin bir ekran görüntüsü



Kendi e-ticaret sitemin bir ekran görüntüsü



Websitenin admin panelinin bir ekran görüntüsü

Cpu:	<input type="text"/>
Cpu çekirdek:	<input type="text"/>
Disk Tür:	<input type="text"/>
Bellek Tür:	<input type="text"/>
Ekran Kartı Model:	<input type="text"/>
Cpu Model:	<input type="text"/>
Disk Kapasite:	<input type="text"/>
Ekran Boyut:	<input type="text"/>
Bellek Kapasite:	<input type="text"/>

[Save and add another](#) [Save and continue editing](#) [SAVE](#)

Kendi sitem için laptop ekleme ekranının bir görüntüsü

Admin panelinden eklenen verileri mongodb'ye otomatik olarak ekliyorum. Veritabanına eklenen veriler daha sonra çağrılıyor.

KAYNAKÇA

<https://getbootstrap.com>

https://www.w3schools.com/python/python_mongodb_getstarted.asp

<https://www.w3schools.com/python/>

<https://www.mongodb.com/docs/>

<https://www.geeksforgeeks.org/python-programming-language/?ref=shm>

<https://sadienvrenseker.com/>

<https://www.youtube.com/watch?v=nZwvTo-fuOAl&list=PLz2cqWVPJE62ESrjIK6On61M5IsLoIiJ>