WEB SCRAPİNG

ALİ KUTAY BİLGİLİOĞLU
KOCAELİ ÜNİVERSİTESİ
Bilgisayar Mühendisliği(i.ö.)
200202108@kocaeli.edu.tr

ÖZET

Fiyat karşılaştırılması için kurulan ve verilerini başka sitelerin HTML kaynağından sağlayan yerel ağ üzerinde çalışan bir web sitesi.
Anahtar Kelimeler:web, NoSQL, python, Javascript, HTML, MongoDB, CSS, Web-Driver

I. GİRİŞ

Tasarlanan web sitede kullanıcı sayfa üzerinde yaklaşık 850'ye yakın bilgisayar, fotoğrafları ve açıklamasını görebiliyor.
Web site bu verileri iki siteden kullanmaktadır. Amazon ve Hepsiburada.
Kullanıcı ürün detaylarını görebilmek için ürünlerin 'incele' sayfasına giderek burdan bilgisayarın hangi sitede var olduğunu ve ne kadar olduğunu görebilecektir. Kullanıcı site logolarına tıklayarak ürünün web sayfasına ulaşabilmektedir.

Python programı sayesinde sitelerde gezinirken, sitelerin iskeleti kopyalandı ve filtrelendi, filtre işlemi sonucunda siteden istenen veriler sınıflanfırıldı, bazı durumlar için sınıflandırılan verilerin tekrar kendi içinde sınıflandırılması gerekti.

Sınıflandırılmış olan veriler bir NoSQL (Not-Only Structured Query Language) programı olan MongoDB'de her bir girdi döküman olacak şekilde daha sonrasında aynı veri tabanından web sitenin verileri kullanabilmesi için veri tabanına yazıldı.

II. Temel Bilgiler

Web Scraping işlemi pythonda, pycharm IDE'si üzerinden yazılmış olup Web sitenin oluşturulması için ise Visual Studio Code ortamı kullanılmış olup node.js, javascript, ejs, html gibi araçlar kullanılmıştır.

III. Yöntem ve Program Mimarisi

Bu kısımda programın farklı özelliklerini oluşturmak için kullandığımız araçlar ve yöntemler üzerinde durularak ayrıntılı olarak bilgi verilecektir. Program mimarisi daha detaylı bir şekilde açıklanacaktır.

A. Web Scraping

Chrome web driver pythona yüklenerek öncelikle program üzerinde otomasyon bir şekilde sitelerin gezilebildiğine emin olundu. Sonrasında ise BeatifulSoup kütüphanesi entegre edilip kullanılarak web site sayfalarındaki HTML verisi çıkartıldı, aynı kütüphane kullanılarak HTML verisi ayrıştırıldı ve değişkenlere atıldı, bazı verilerin başlık stringinden ayrıştırılabilmesi için RegEx kullanıldı.

RegEx kullanımı sayesinde string verileri ve dizilimi birbirlerinden çok farklı da olsa String içinde aranan benzer veriler çekilip kullanılmıştır.

Örnek RegEx kullanımı

```
islemci_pattern = re.compile(r'i\d.....|ryzen.....|celeron....|m2|m1', re.IGNORECASE)
```

Daha sonra kullanılabilmesi için ürünlerin href verisi ve ürün görüntüsü için de img src verisi toplandı.

Site içi hızlı kullanımda kullanıcıyı Chrome kimliğinden engelleyen (request yöntemi kullanılmadığı için bir tehlike oluşturmadı) ya da sitenin hızlı kullanıldığını fark edip IP Ban atması veya hızlı kullanım sonucu sitenin Chrome Driver karşısına ReCaptcha yollamasından kaçınmak için program üzerinden site içi kullanım yapılmadı, program sayfa değiştirmek için hali hazırda elinde olan URL stringine bir sayfa numarası atılarak (çerezler de aktif olmadığı için) veri çekilen web sitenin programı engellemesinden kaçınılmıştır. Aynı zamanda bu yöntem ile for döngüsü içindeki bir parametrenin değiştirilmesi ile programın kaç sayfa veri çekeceği rahat bir şekilde belirlenebiliyor.

Amazon ve Hepsiburada'dan çekilen veriler daha sonra web sitesi tarafından karşılaştırılmak üzere iki ayrı collection'a yollanıyor, her web sitenin kendine özgü HTML karakteri olduğu için Hepsiburada'dan veri çekmesi için yazılmış olan python kodu amazon kodundan tamamen farklı bir script içinde yazıldı.

MongoDB üzerinde herhangi bir duplikasyon oluşmaması için her site taraması yapıldığında eski collectionlar droplanır.

B. Web Sitesi(Back-end ve Front-end)

Node.Js tarafında web sitenin gösterilebilmesi için lokal bir sunucu oluşturulur.

Programın her kayıttan sonra tekrar çalışarak kendisini güncelleyebilmesi için npm tarafından nodemon yüklenmiştir. Front-end ve Back-end iletişimin kolaylaştırılabilmesi için Express kullanılmıştır. MongoDB Crud için ise JavaScript uyumlu Mongoose kullanılmıştır. Back-End tarafından HTML'e yollanan verilerin daha kolay işlenebilmesi ve HTML içinde mantıksal ifadeler kullanabilmek için ejs aracı kullanılmıştır.

Web Sitenin arayüz tasarımının güzelleştirilmesi için bootstrap kullanılmıştır.

Main JavaScript programı içinde MongoDB veri tabanına bağlanır. Daha sonra iki collectiondan da veriler çekilir. Çekilen collectionlarda her bir verinin diğer her bir veri ile karşılaştırılması yapılır ve aynı modelde olan bilgisayarların modeli daha sonra kullanılmak üzere bir diziye atılır.

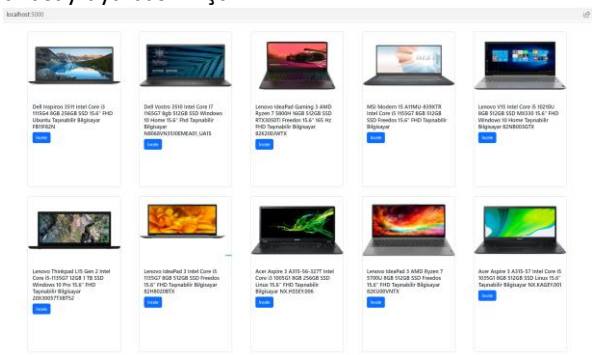
```
const amazon_data = await amazon.find()
const hepsi_data = await hepsiburada.find()
var same_model1 = [];
amazon_data.forEach(function(obj1){
    hepsi_data.forEach(function(obj2){
        if(obj1.model === obj2.model){
            if(obj1.model != null){
                same_model1.push(obj1.model);
            }

        }
    });
})
```

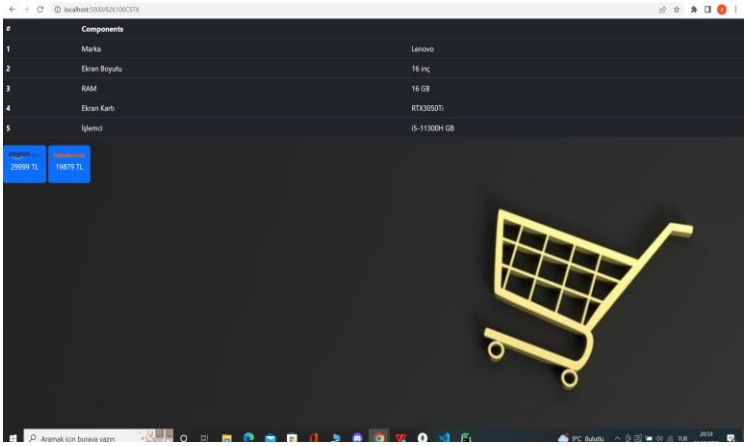
Bu dizi içinde tekrarlı verilerden kurtulunur. app.use kullanılarak kullanıcının tıkladığı ürünün model numarası çekilir ve bu fonksiyon içinde bakılmak istenilen bilgisayarın modeli her iki sitede de mevcut olup olunmadığına bakılır. Bu durum şartlarına göre veriler farklı şekillerde kullanılmak üzere farklı ejs(HTML) formatına yollanabilir.

C. Arayüz

Arayüz için bootstrap(css) kullanılmıştır. Her ürün bir container classına alınarak container içine ürün resmi ve açıklaması konulmuştur. Çok fazla ürün olduğu için bu verileri tek tek html e yazmak yerine HTML içinde ejsloop kullanılarak veri sayısının önemi kalmadan tüm veriler düzenli bir şekilde anasayfaya basılmıştır.



Kullanıcı herhangi bir ürünün incele butonuna bastığında back-end tarafında modelin hangi sitede ya da iki sitede de mi mevcut sorularını cevaplar ve buna göre ilgili HTML e yönlendirilir. Bu sayfada ise tablo içinde ürün bilgileri, fiyat ve web site yönlendirilmesi yapılır.



IV. Deneysel Sonuçlar

Projeyi araştırma aşamasında internette herhangi bir sayfadan veri çekmeyi ve temelde çok ünlü olan akakçe.com, cimri.com gibi sitelerin çalışma prensibi anlaşıldı.

Bu bilgilerin sadece web sitesi kurmak için değil aynı zamanda bir çok siteden bir çok veriyi işleyerek doğru olanı bulma, bilimsel araştırmalar için gereken verinin elde edilmesi için de kullanılabileceğinin farkına varıldı.

Python kullanarak string verilerinin sınıflandırılması ve işlenmesi konusunda fikirler elde edildi

Bootstrap ve ejs kullanılarak web site tasarımının matematiği ve mantığı anlaşılırken bir web sitesinde arka planda gerçekleşen mantıksal ve matematiksel işlemlerin farkındalığını JavaScript kullanarak anladık.

İlk kez genel web tecrübesinin yanı sıra aynı zamanda ilk defa da NoSQL kullandık ve SQL den farkları tarafımızdan farkedildi.

Bulutta tutulan verilerin (resim gibi) nasıl kendi bilgisayarına indirilmeden bilgisayarında web sitesinin internette bu linklere giderek bu verileri kullanabileceğini çıkarttık.

Aynı zamanda JavaScript ve Python kullanımını daha iyi öğrendik.

V. SONUÇ

Pythonda veri çekme ve işleme konusunda deneyimlendik.

JavaScript kullanmayı öğrendik Node.js kullanımı ile yerelde nasıl web sitesi uygulanabileceğini öğrendik.

NoSQL kullanmayı öğrendik ve ilk defa bir veritabanını web sitesi için kullanmış olduk.

Web sitesinde site içinde uzantılarla routing yapılma öğrenildi

VI. Yalancı Kod

- Başla
- Web Driver'ı başlat
- Verilerin kaydedileceği veri tabanı ve collection'a ulaş
- For döngüsünü başlat
- Driver'ı site URL'ine yönlendir
- Döngü indeksini URL üstüne ekle
- Soup ile HTML verilerini al ve bu veriler içinde sadece ürünleri item isminde ayır
- Ürünün sponsorlu ürün olup olmadığını bul, eğer sponsorluysa aynı sayfa içinde sıradaki ürüne geç
- Ürünün markasını bul
- Ürünün model numarasını çıkart
- Ürünün komponent bilgilerini başlıktan çıkar
- Ürünün verilerini sınıflandırılmış bir şekilde MongoDB'ye yolla
- JavaScript üzerinden veritabanına bağlan
- Verilerin hepsini çıkart
- Verileri karşılaştırarak aynı ürünlerin model numarasını çıkart
- Model numarasını saklayan dizi duplikelerinden kurtul
- Site işleyişini çıkart
- Kullanıcının tıkladığı ürünün model numarasını bul
- Aynı modeldeki cihazların iki sitede de olup olmadığını bul
- Iki sitede de bulunuyorsa farklı aynı sitede bulunuyorsa farklı HTML e gönder
- Tablo tanımla ve gönderilen verileri sınıflandırarak tablonun içine yerleştir
- Duruma göre buton sayısı ve buton resmi çıkart
- Butonların işlevsel olabilmesi için ilgili ürün linklerine bağla
- Fiyat bilgilerini yazdır
- Bitiş

VII. KAYNAKÇA

- https://www.w3schools.com/html/html_images_background.asp
- https://getbootstrap.com/docs/4.1/content/tables/
- https://stackoverflow.com/questions/8683528/embed-image-in-a-button-element
- https://www.includehelp.com/node-js/ejs-if-else-statement-ejs-conditions.aspx
- https://expressjs.com/en/guide/routing.html

VIII. Akış Şeması