

YAZILIM LABORATUVARI 1. PROJE

Buğra Oktay Ateş

I. ÖZET

Bu doküman Yazılım Laboratuvarı 2 dersinin 1. Projesini açıklamaya yönelik oluşturulmuştur. Dökümanda giriş, yöntem, sonuç, proje hazırlanırken kullanılan geliştirme ortamı gibi programın oluşumunu açıklayan başlıklara yer verilmiştir. Doküman sonunda projemi hazırlarken kullandığım kaynaklar ve proje derlenirken dikkat edilmesi gereken hususlar bulunmaktadır.

II. GİRİŞ(PROJE TANITIMI)

Google Akademik gibi akademik arama motorları üzerinden web scraping (web kazıma) yöntemiyle aratılan akademik yayınlara ait bilgilerin kaydedildiği bir veritabanıyla birlikte bu bilgilerin webden aratılması, görüntülenmesi ve istenilen özelliklere göre sorguların yapılmasına imkan sağlayacak bir web arayüzü geliştirmeniz beklenmektedir.

Amaç: Proje sayesinde web scraping ile bir web sayfasından bilgiye erişim sağlama; MongoDB veritabanı ile Elasticsearch sorgu yapılarını kullanma ve web kodlama becerilerinin geliştirilmesi amaçlanmaktadır.

Programlama Dili: Proje veritabanı ve sorguları için MongoDB ile Elasticsearch yapısı kullanılmalıdır. Web arayüzü, istenilen bir web programlama dili kullanılarak oluşturulacaktır..

Proje aşağıda detayları verilen 3 ana kısımlardan oluşmaktadır:

1. Web Scraping: En az bir akademik arama motorundan web scraping kullanılarak kullanıcının gireceği anahtar kelimelere göre listelenmiş en az ilk 10 akademik yayının bilgileri, kendi oluşturacağınız web sayfasında görüntülenmelidir. Kullanıcının arama yapmak için kullanacağı anahtar kelimeler oluşturacağınız kendi web sayfanızdaki bir text alanı üzerinden girilecektir.

Web scraping işlemi için siteye ait html bilgiler kullanılarak veya siteye request isteği yapılarak istenen veriye erişilmelidir. (Erişilecek siteye yönelik hazır web API ler kullanılmamalıdır.)

İstenen yayına ilişkin bilgiler doğrudan akademik arama motorunun sayfasından çekilebileceği gibi arama motoru sayfasındaki link üzerinden yönlendirilecek diğer bir web sayfasından da elde edilebilir. (İkincil sitelere geçiş yapılarak web scraping yapılması durumunda artı puan verilecektir.)

İstenen her yayın için pdf dosyası mutlaka indirilmelidir. Daha sonra tercihe göre yayın bilgileri ya web sayfası üzerinden ya da indirilmiş pdf dosyasının içeriğinden elde edilebilir.

2. Veritabanı: Web scraping ile elde edilen veriler MongoDB kullanılarak veritabanına kaydedilecektir.

—Veritabanında tutulması gereken bilgiler şunlardır:

- Yayın id,
- Yayın adı,
- Yazarların isimleri,
- Yayın türü (araştırma makalesi, derleme, konferans, kitap vb.),
- Yayımlanma tarihi,
- Yayıncı adı (yayının yayımlandığı konferans ismi; dergi veya kitap yayınevi),
- Anahtar kelimeler (Arama motorunda aratılan),
- Anahtar kelimeler (Makaleye ait),
- Özet,
- Referanslar,
- Alıntı sayısı,
- Doi numarası (Eğer varsa),
- URL adresi

3. Web Sayfası:

Erişilen yayınların bilgilerinin kullanıcıya gösterilmesi için bir web sayfası oluşturmanız beklenmektedir.

Web sayfasında aratılacak yayınlar için bir text alanı oluşturulmalı ve bu text alanı girilecek anahtar kelimeler üzerinden ilgili arama motorunun yayınları aratıp bilgilerini web sayfasına getirmesi sağlanmalıdır.

Web sayfası ilk açıldığında veritabanında bulunan tüm kayıtlar sayfaya getirilmelidir.

Listeleme işleminde yayınların isimleri sırasına uygun biçimde listelenmelidir. Listelenen bir

makale ismine tıkladığında o makaleye özgü bilgiler ayrı bir sayfada gösterilmelidir.

Web sayfasından herhangi bir çalışmaya yönelik dinamik arama işlemi yapılabilir. Ayrıca arama sırasında yazım yanlış olması durumunda sistem düzeltilmiş öneride bulunmalıdır. Örnek: deep learning – yazımı düzeltilmiş sonuçları görüyorsunuz: deep learning şeklinde olmalıdır.

Web sayfasında dinamik filtreleme işlemi de ayrıca yer almalıdır. Filtreleme işlemi veritabanında yer alan yayının tüm özellikleri için hem ayrı ayrı hem de birlikte uygulanabilir olmalıdır.

Web sayfasında en son veya en önce yayımlanma tarihine göre sıralama yapılabilir ayrıca yine atf sayısına göre de en az veya en çok olacak şekilde sıralama yapılabilir.

III. YÖNTEM

Bu bölümde projeyi baştan sona nasıl yaptığımı anlatacağım.

Amaç: Bu proje, Puppeteer kullanarak DergiPark sitesinden belirli bir anahtar kelimeye göre ilk 10 makaleyi çekmeyi amaçlamaktadır.

Kullanılan Araçlar:

Puppeteer: Tarayıcı otomasyonu ve kontrolü sağlamak için kullanıldı. **Node.js:** Puppeteer ile etkileşim sağlamak için kullanıldı. **Visual Studio Code:** Kodlama ortamı olarak kullanıldı. **Uygulama Adımları:**

Puppeteer kütüphanesinin kurulumu. Tarayıcıyı başlatma ve DergiPark sitesine gitme. Anahtar kelime araması yapma. Arama sonuçlarını analiz etme ve ilk 10 makaleyi çekme. Makale bilgilerini toplama (başlık, yazarlar, yayın tarihi, özet vb.). Toplanan verileri düzenleme ve raporlama.

Projeyi ilk olarak hangi nodejs web scraping kütüphanesi ile yapmam gerekli onu araştırdım ve bana uygun kütüphanenin Puppeteer olduğuna karar verdim. "Puppeteer", bir Google Chrome tarayıcısının otomatik olarak kontrol edilmesine olanak tanıyan bir Node.js kütüphanesidir. Web sayfalarını açabilir, içeriklerini değiştirebilir, formları doldurabilir, tıklamalar yapabilir, kaydırabilir, ekran görüntüsü alabilir ve daha birçok işlemi gerçekleştirebilir. Genellikle web scraping, otomasyon testleri ve otomatikleştirilmiş web iş akışları gibi alanlarda kullanılır. Puppeteer, tarayıcıyı kontrol etmek için bir dizi API sağlar ve karmaşık işlemleri otomatikleştirmek için kullanıcıya geniş bir esneklik sağlar.

Scraping kütüphanesini seçtikten sonra sıra front-end kısmına geldi. HTML ve CSS ve Bootstrap kullanarak bir arama çubuğu ve arama çubuğunun altına web sitesinden çekilen bilgilerin ekranda gösterilmesi için 10 adet kart oluşturdum. Arama çubuğuna yazılan anahtar kelimeleri back-end e gönderen bu yapı sayesinde anahtar kelime ile web scraping yapabilmekteyiz.

Web Scraping Akademi Yayın Bilgileri

Yayın ID: 21

Yayın Adı: Beyaz Kod Verileri Işığında 112 Çalışanlarına Yönelik Şiddet; Retrospektif Değerlendirme

Yazarların İsimleri: Serap BATI Necmettin Erbakan Üniversitesi Seydişehir Sağlık Bilimleri Fakültesi 0000-0003-3176-6837 Türkiye Hasan KUÇUKKENDİRCİ MİRAM FACULTY OF MEDICINE 0000-0001-9015-7367 Türkiye Osman ULUSAL Konya İl Sağlık Müdürlüğü 0000-0001-8319-5306 Türkiye

Yayın Türü: Makale

Birincil Dil: Türkçe

Yayımlanma Tarihi: 25 Nisan 2021

Anahtar Kelimeler (Arama Motorunda): kod

Anahtar Kelimeler (Makaleye Ait): beyazkod,sağlıkçısı,sağlıkşiddet

URL Adresi: <https://dergipark.org.tr/tr/download/article-file/1356684>

Referanslar: Referans1: Acik, Y., Devenci S. E., Günes, G., Gulbayrak, C., Dabak, S., Saka, G., Vural G., Can, G., Bilgin, N. G., Dündar, P. E., Erguder, T., & Tokdemir, M. (2008). Experience of workplace violence during medical specialty training in Turkey. *Occupational Medicine*, 58 (5), 361-366. <https://doi.org/10.1093/occmed/kqn045> Referans2: Al, B., Zengin, B., Derya, Y., Gökçen, C., Yılmaz, DA., & Yıldırım, C. (2012). Increased violence towards health care staf. *The Journal Of Academic Emergency Medicine*, 11, 115-24. <https://doi.org/10.5152/jaem.2012.033> Referans3: Atan, ŞÜ., & Dönmez, S. (2011). Hemşirelere Karşı İşyeri Şiddeti. *Adli Tıp Dergisi*, 25 (11), 71-80. https://www.journalagent.com/adlittip/pdfs/ADLITIP_25_1_71_80.pdf Referans4: Ayranıcı, Ü., Yenilmez, Ç., Balci, Y., & Kaptanoğlu, C. (2006). Identification of violence in Turkish health care settings. *Journal of Interpersonal Violence*, 21, 276-296. <https://doi.org/10.1177/0886260506282565> Referans5: Çelebi, İ. (2016). Çanakale 112

Back-end e gönderilen anahtar kelime ile dergipark sitesindeki arama çubuğuna istediğimiz kelimeleri yazdırabilmekteyiz. Bu sayede anahtar kelimeye göre çıkan ilk 10 makalenin bililerine ulaşabiliriz. Yaptığım program ilek olarak degi park sitesine girip anahtar kelimeyi arama çubununa yazıp enter tuşuna basmaktadır. Sonra çıkan sayfalardan ilk makaleye girip onun bilgilerini mongo db veritabanına yazdırmaktadır. Yazdırma işlemi bittikten makalenin pdf dosyasına gidip dosyayı önceden belirlediğim masa üstündeki "pdfs" klasörüne id numarası ismini alarak indirilmektedir. Tabi bu anlattığım işlemler ilk 10 makale için sırası ile yapılmaktadır. Tüm işlemler bitince web arayüzümüzde bakaleye ağıt bilgiler gösterilmektedir. Bu bilgilere ek olarak detay bilgiler öğrenmek istersek detaylar linkine tıklayarak makalenin detaylarını öğrenebiliriz.

Ayrıca veritabanındaki tüm bilgileri ekrana yazdıran bir tüm bilgiler sayfasında mevcuttur. Bu sayfada tüm makalelerin bilgilerinin erişilebilirliği gibi makaleler içinde deki özelliklere göre filtreleme işlemi yapılabilmektedir. Ek olarak makaleleri id

numarasına göre büyükten küçüğe veya küçükten büyüğe doru sıralama yapılabilmektedir. Aynı şekilde yayınlanma tarihine göre de yeniden eskiye veya eskiden yeniye olmak üzere sıralama işlemi yapılabilmektedir.

IV. SONUÇ

Özetle bizden Google Akademik gibi akademik arama motorları üzerinden web scraping (web kazıma) yöntemiyle aratılan akademik yayınlara ait bilgilerin kaydedildiği bir veritabanıyla birlikte bu bilgilerin webden aratılması, görüntülenmesi ve istenilen özelliklere göre sorguların yapılmasına imkan

sağlayacak bir web arayüzü geliştirmemiz beklenmekteydi bunu gerçekleştirdik.

V. KAYNAKÇA

https://stackoverflow.com
https://getbootstrap.com/docs/5.3/utilities/spacing/margin-and-padding
https://www.w3schools.com
https://chat.openai.com/?model=text-davinci-002-render-sha

VI. KOD ÖRNEĞİ

```
app.post('/ara', (req, res) => { adlar = []; (async () => { const browser = await puppeteer.launch( { headless: false }); const page = await browser.newPage(); await page.goto('https://dergipark.org.tr/tr/'); await page.type('input[name="q"]', req.body.arama); await page.keyboard.press('Enter'); for (let i = 1; i <= 11; i++) { var s = 'search .article-cards .article-card:nth-child(i)a'; await page.waitForSelector(s); await page.click(s); var id = await isimkaydet(page, adlar, req.body.arama); if (id !== -1 & id !== -2) await anahtarkelime(page, id); await ozet(page, id); await yazar(page, id); await referans(page, id); var s1 = 'article-toolbar a'; await page.waitForSelector(s1); await page.click(s1); if (id !== -1 & id !== -2) const update = { set : { Url : page.url() }; const result = await db.collection("Ogrenci").updateOne({ d : id, update }); //await downloadPdf(page.url(), 'C:/Users/bugra/OneDrive/Masaüstü/pdfs/id.pdf'); await page.goBack(); // Tarayıcıyı kapatın await browser.close(); results = adlar; await res.render("app", results ); }); }; app.get('/app', (req, res) => { results = ""; res.render("app", results ); }); const puppeteer = require('puppeteer'); (async () => { const browser = await puppeteer.launch(); const page = await browser.newPage(); await page.goto('https://www.dergipark.org.tr/'); // Anahtar kelimeye göre arama yapma const anahtarKelime = 'anahtar kelime'; await page.type('search-text', anahtarKelime); await page.click('search-button');
```

```

// İlk 10 makaleyi çekme
const makaleler = await
page.evaluate(() => {
  const makaleListesi = [];
  const makaleler = document.querySelectorAll('.result-item');
  makaleler.forEach(makale => {
    const baslik = makale.querySelector('.result-title').innerText;
    const yazarlar = makale.querySelector('.result-authors').innerText;
    const yayinTarihi = makale.querySelector('.result-date').innerText;
    const ozet = makale.querySelector('.result-summary').innerText;
    makaleListesi.push( baslik, yazarlar, yayinTarihi, ozet );
  });
  return makaleListesi.slice(0, 10);
});

console.log('Toplam ' + makaleler.length + ' makale bulundu:');
console.log(makaleler);

await browser.close();
})();

```