Kocaeli Üniversitesi Yazılım Laboratuvarı 2 Proje 1

Edanur Çövüt-190201035

17 March 2024

1 Özet

Bu doküman Yazılım Laboratuvarı 2 dersi 1. Projesi için hazırlanmıştır. Dokümanda projenin tanımı, çözüme yönelik yapılan araştırmalar, kullanılan yöntemler,proje hazırlanırken kullanılan geliştirme ortamı ve kod bilgisi gibi programın oluşumunu açıklayan başlıklar bulunmaktadır. En sonda da kaynakça kısmı bulunmaktadır.

2 Giriş

Bu projede 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 geliştirilmesi beklenmektedir. Projenin amacı web scraping ile bir web sayfasından bilgiye erişim sağlama; MongoDB veritabanı ile Elasticsearch sorgu yapılarını kullanma ve web kodlama becerilerinin geliştirilmesidir.

3 Yöntem

Bu projenin amacı akademik arama motorlarından web scraping yaparak, aratılan akademik yayınlara ait bilgilerin kaydedilmesiyle bu bilgilerin aratılması, görüntülenmesi ve sorguların yapılabilmesini sağlayan arayüz geliştirmektir. Web scraping (web kazıma), web sitelerinden veya web sayfalarından veri çekme işlemidir. Bu işlem, bir otomasyon aracı veya bir bilgisayar programı kullanılarak yapılır ve internet üzerindeki çeşitli kaynaklardan bilgi toplamak için kullanılır. Web scraping (web kazıma) işlemi, web sayfalarındaki metin, görüntü, bağlantı, veritabanlarından veri ve diğer bilgileri almak için kullanılabilir.

Web scraping (web kazıma) işlemi şu temel adımlardan oluşur: Bir hedef web sitesi veya web sayfası belirlenir. Bir web scraping aracı veya programı kullanılarak hedef web sitesine istek gönderilir ve sayfa içeriği alınır. Alınan sayfa içeriği analiz edilir ve istenilen verileri çıkarmak için belirli kurallar veya desenler kullanılır. Elde edilen veriler daha sonra kullanım için saklanabilir veya başka bir işleme tabi tutulabilir. Web scraping (web kazıma); bilgi toplama, fiyat karşılaştırmaları yapma, pazar araştırması yapma, haber izleme, otomasyon ve daha birçok amaç için kullanılabilir. Ancak, web

scraping'in yasal ve etik sınırlamaları vardır. Web siteleri, kullanıcıların izni olmadan verilerini çekmeye karşı korumalıdır ve bazı siteler scraping'i yasaklar veya sınırlar. Bu nedenle, web scraping yapmadan önce ilgili yasal düzenlemelere ve web sitesi politikalarına uymak önemlidir. Web scraping için birçok farklı programlama dili ve araç mevcuttur. Hangi dil ve aracı kullanmanız gerektiği, projenin karmaşıklığına, hedef web sitesinin yapısına ve kişisel tercihe bağlıdır. Bu projede yaygın olarak kullanılan Python programlama dili kullanıldı. Python, web scraping için en popüler programlama dillerinden biridir. Beautiful Soup ve Requests gibi kütüphaneler sayesinde web sayfalarını çözümlemek ve veri çekmek kolaydır. Ayrıca, Selenium gibi bir otomasyon aracı, dinamik web siteleriyle etkileşim kurulmasına yardımcı olabilir.

MongoDB kullanımı

MongoDB, belge tabanlı bir NoSQL veritabanı yönetim sistemidir. MongoDB hakkında temel bilgiler:

Belge Tabanlı Veritabanı: MongoDB, JSON benzeri bir formatta belgeleri saklar. Her belge, anahtar-değer çiftlerinden oluşan bir JSON nesnesidir.

Esnek Şema: İlişkisel veritabanlarından farklı olarak, MongoDB'de her belge kendi şemasını tanımlar. Bu, veri modellemesinde esneklik sağlar ve uygulama gereksinimlerine göre dinamik olarak değişebilir.

Gelişmiş Sorgu Desteği: MongoDB, karmaşık sorguları destekleyen zengin bir sorgu diline sahiptir. Bu dil, belgeleri filtrelemek, sıralamak, birleştirmek ve dönüştürmek için kullanılır.

Açık Kaynak Kodlu: MongoDB, açık kaynaklı bir projedir ve topluluk desteğiyle geliştirilmektedir. Bu, kullanıcıların kodu inceleyebilmesini, değiştirebilmesini ve katkıda bulunabilmesini sağlar.

İşletim Sistemlerinde Çalışma: MongoDB, Linux, Windows ve macOS gibi çeşitli işletim sistemlerinde çalışabilir. Ayrıca, bulut tabanlı hizmetler üzerinde de kullanılabilir.

Veri Toplama (Web Scraping):

Python programlama dili kullanarak web scraping için script oluşturuldu. Kütüphane kullanarak web sayfalarını analiz edildi ve istenen veriler çekildi. Web scraping script'inizi geliştirirken hedef web sitesinin robots.txt dosyasını ve kullanım koşullarını dikkate alın. Aşırı sorgu göndermemeye özen gösterin ve web sitesini aşırı yüklemekten kaçının. Verilerin İşlenmesi ve Temizlenmesi:

Web scraping ile topladığınız veriler genellikle düzensiz olabilir. Bu nedenle, verileri işleyerek istenmeyen karakterleri kaldırın, eksik veya yanlış verileri düzeltin ve gerekirse veri formatını standartlaştırın. MongoDB Veritabanı Oluşturma:

MongoDB'de verilerinizi saklamak için bir veritabanı ve koleksiyonlar oluşturun. MongoDB'de veritabanı ve koleksiyonları oluşturmak için MongoDB Shell'i veya bir MongoDB yönetim aracı kullanabilirsiniz. Verilerin MongoDB'ye Aktarılması:

Web scraping ile topladığınız verileri işleyerek MongoDB'ye aktarın. Bu işlemi Python'daki pymongo gibi bir MongoDB sürücüsü kullanarak gerçekleştirebilirsiniz. Verileri uygun MongoDB belgeleri olarak düzenleyip, MongoDB koleksiyonlarına ekleyebilirsiniz. Verilerin Sorgulanması ve Analizi:

MongoDB'de sakladığınız verileri sorgulayarak analiz edin. Bu, veriler üzerinde filtreleme, sıralama, gruplama ve diğer iş-

lemleri gerçekleştirmenizi sağlar. PyMongo veya MongoDB Shell'i kullanarak sorgular oluşturabilirsiniz. Güvenlik ve Performans İyileştirmeleri:

Performansı artırmak için uygun indeksler oluşturun ve MongoDB'nin ölçeklenebilirlik özelliklerinden yararlanın. Web scraping ile topladığınız verileri MongoDB'de saklayarak, verilerinizi etkili bir şekilde depolayabilir, sorgulayabilir ve analiz edebilirsiniz. Bu şekilde, web scraping ile topladığınız verileri daha işlevsel hale getirebilir ve çeşitli analizler için kullanabilirsiniz.

4 Sonuç

Projenin başlangıcında, belirlenen hedeflere ulaşmak için üç ana adım bulunmaktadır: Web Scraping, Veritabanı Yönetimi ve Web Arayüzü Geliştirme.

1. Web Scraping

Web scraping işlemi için Python programlama dilini kullanarak, BeautifulSoup ve requests gibi kütüphanelerden yararlandık. Springer arama motorundan yapılan kullanıcı sorgusuna göre en az ilk 10 akademik yayının bilgilerini topladık. Bu işlemi gerçekleştirmek için, kullanıcının girdiği anahtar kelimeleri alarak, Springer arama motorunun web sayfasına bir HTTP isteği gönderdik. Ardından, gelen web sayfasının HTML yapısını BeautifulSoup kütüphanesi aracılığıyla analiz ettik ve istediğimiz bilgileri çıkardık. Gerekli durumlarda, yayınların detaylarını içeren sayfalara yönlendirme linklerini takip ederek ikincil web sayfalarından da bilgi topladık.

2. Veritabanı Yönetimi

Web scraping ile elde edilen verileri saklamak için MongoDB veritabanını tercih ettik. Python programlama dili kullanılarak, pymongo kütüphanesi aracılığıyla MongoDB'ye eriştik ve toplanan yayın bilgilerini veritabanına kaydettik. Veritabanında her bir yayın için belirlenen özelliklere uygun bir veri yapısı oluşturduk ve yayınların detaylarını kaydettik.

3. Web Arayüzü Geliştirme

Web arayüzünü oluşturmak için JavaScript kullandım. Kullanıcıya yayınları aramak için bir metin alanı sağladık ve girilen anahtar kelimelere göre ilgili arama motorundan yayınları çekip web sayfasına getirdik. Ayrıca, veritabanındaki tüm kayıtları ilk açılışta web sayfasına getirdik ve kullanıcıya yayınların listesini sunarak, istediği yayına tıklayarak detaylı bilgilere ulaşma imkanı sunduk. Dinamik arama ve filtreleme işlemleri için gerekli kodları yazarak, kullanıcının isteklerini karşıladık. Bu yöntemlerin kombinasyonuyla, web scraping ile bilgi toplama, MongoDB ile veri depolama ve JS ile web arayüzü geliştirme süreçlerini başarıyla tamamladık.

5 Deneysel Sonuçlar

1. Web Scraping Sonuçları:

Web scraping işlemi sırasında, belirlenen akademik arama motoru üzerinden kullanıcıların girdiği anahtar kelimelere göre en az ilk 10 akademik yayının bilgileri başarıyla çekilmiştir. BeautifulSoup ve requests gibi Python kütüphaneleri kullanılarak web sayfaları analiz edilmiş ve istenilen verilere erişilmiştir. Ayrıca, ikincil web sayfalarına yönlendirme linkleri takip edilerek detaylı bilgilerin alınması sağlanmıştır. Bu süreçte, web scraping işlemi istenilen verilere başarıyla erişmiş ve verilerin toplanması sağlan-

mıştır.

2. Veritabanı Yönetimi Sonuçları:

MongoDB veritabanı kullanılarak web scraping ile toplanan veriler veritabanına başarılı bir şekilde kaydedilmiştir. Her bir yayın için belirlenen özelliklere uygun bir veri yapısı oluşturulmuş ve yayın bilgileri MongoDB koleksiyonlarında saklanmıştır. Veritabanı yönetimi işlemi sırasında, verilerin düzenli ve tutarlı bir şekilde depolanması sağlanmıştır. Bu sayede, verilere kolayca erişilebilir ve sorgulanabilir bir yapı oluşturulmuştur.

3. Web Arayüzü Geliştirme Sonuçları:

JS kullanılarak geliştirilen web arayüzü, kullanıcı dostu ve etkileşimli bir yapıya sahiptir. Kullanıcıya yayınları aramak için bir metin alanı sunulmuş ve girilen anahtar kelimelere göre ilgili arama motorundan yayınlar çekilerek web sayfasına getirilmiştir. Ayrıca, veritabanındaki tüm kayıtların listesi kullanıcıya sunulmuş ve kullanıcı istediği yayına tıklayarak detaylı bilgilere ulaşabilmiştir. Dinamik arama ve filtreleme işlemleri için gerekli kodlar yazılarak, kullanıcının isteklerini karşılayacak bir arayüz sağlanmıştır.

Genel Sonuçlar:

Proje kapsamında gerçekleştirilen deneyler, web scraping, MongoDB veritabanı yönetimi ve JS ile web arayüzü geliştirme süreçlerinin başarılı bir şekilde tamamlandığını göstermektedir. Elde edilen sonuçlar, projenin amaçlarına ulaşılmasında başarılı olduğunu ve projenin istenilen özelliklere göre sorguların yapılabildiği bir web arayüzüyle sonuçlandığını göstermektedir.

6 Geliştirme Ortamı

Projeyi Windows sistemde, VSCode üzerinde geliştirip yine VSCode kullanarak derledik.

6.0.1 İstatistik

Program kodu 160 satırlık tek dosyadan oluşmaktadır. models içerisinde scrapped.js, router içerisinde index.js, model.js, controllers içerisinde extraModel.js, model.js, runner.js router içerisinde ve ex.py, index.js, package-lock.json, package.json ve scraper.py belgeleri bulunmaktadır Kullandığım kütüphaneler kabaca aşağıdaki gibidir: bs4 beautifulsoup pandas requests

6.0.2 Programın Derlenmesi

Programın kaynak kodu tek dosyadan oluşmaktadır. Bu dosyayı Pycharm ile derleyebilirsiniz.

7 Kaynakça

1)https://medium.com/kaveai/web-scraping-453e96a86195
2)https://www.codiasoft.com/blog/web-scraping-web-kazima-nedir-neden-yapilir/ 3)https://www.springer.com/gp
4)https://www.youtube.com/watch?v=XVv6mJpFOb0
5)https://www.youtube.com/watch?v=8dTpNajxaH0
6)https://www.w3schools.com/mongodb/