Yazilim Labaratuvari 1 1.PROJE

OMER FARUK ULUSOY
200202032

MUHAMMED FATIH OZEN
200202050

I. OZET

Bu döküman yazilim labaratuvarı 1 dersi 1. projesi için çözümümüzü açıklamaya yönelik oluşturulmuştur.Dökümanda projenin tanımı,arastırma yöntem ve deneysel sonuçlar,algoritmalar gibi programın oluşumunu açıklayan başlıklara yer verilmiştir.Dökümanın sonunda projeyi hazırlarken kullanılan kaynaklar bulunmaktadır.

II. PROJENIN TANIMI

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. Web scaping ile fiyat karsilastirma site yapiniz.

A. Projenin Konusu

Web scaping ile fiyat karsilastirma sitesi yapiniz.Ayriyetten kendi e-ticaret sitesini yapiniz

III. ARAŞTIRMA YÖNTEM VE DENEYSEL SONUÇLAR

Bu projeyi gerceklestirmek icin oncelikle java spring boot,rest api ve react teknolojilerini, ve asil islem olan web Scraping i arastirdik.Sonrasında proje için gerekli olan dependencileri maven ile kurduk.Bu proje için izlediğimiz yöntem böl,parçala ve yönettir.Tüm isterleri fonksiyonlara parçalayarak mantığını kavrayıp içeriğini tek tek yazdık.Yapamadığımız fonksiyonları başka bir dosyada açıp orada yapıp asıl dosyaya gömdük.Projeyi yaparken kalem kağıt kullanmaya özen gösterdik.Sonuç olarak projeden algoritma analizimizin geliştiğini fonksiyonlarla programlamaya adım atmayı gerçekleştirmeyi takım arkadaşı olarak proje yapmayı verimli bulduk.

IV. ALGORITMA-KOD BILGISI

Bu kısımda projeyi açıklayacağız.
Bu proje için izlediğimiz yöntem daha öncede söylediğimiz gibi böl,parçala ve yönet prensibidir.Açıklamaya başlamadan önce yapmamız gereken isterler şunlardır;

- Siteleden WebScapring,
- WebScarping yapilan sitelerin fiyat karsilastirmasi,
- e-site yapimi,
- e-karsilastirma site yapimi.

A. KULLANILAN TEKNOLOJILER

- Java

  Java, 1995 yılında Sun Microsystems tarafından yayınlanmış bir hesaplama platformu ve programlama dilidir. Basit başlangıçlar ile ortaya çıkmıştır ve birçok hizmet ve uygulamanın oluşturulduğu güvenilir platformu sağladığı için bugünün dijital dünyasında en büyük paya sahiptir. Gelecekte kullanmak üzere tasarlanmış yeni, inovatif ürünler ve dijital hizmetler de Java'yı temel almaya devam ediyor.

- Java-Jsoup

  Jsoup html parse etmemizi sağlayan bir kütüphanedir.Peki bu kütüphaneyle ne yapabiliriz.Projemize dahil ettiğimiz andan itibaren projemizden bir web sayfasını response edebiliriz ve geri dönen değerlerle çalışmalar veya bir web sayfasının içindeki bir veriyi alabiliriz.Jsoup neden ihtiyaç duyarız:
  Jsoup web servis olmadığı bir web projesinde mobil veya başka projeden verilere erişmekte kullanabiliriz.Dezavantajı ise projemizde kullanıldıktan sonra html in etiketleri (tag leri) değişince projemiz çalışmayacaktır.Html i parse ederken divlerin css attiribute lerinden faydalanırız

- Maven

  Maven genellikle Java platformunda yer alan komutların derlenmesi sırasında kullanılan otomasyon ve inşa aracıdır.
  Java programlama dili ile uygulama geliştirirken çeşitli kütüphaneler kullanamak isteyebiliriz.
  Örneğin; Java ile PDF dosyası oluşturmak için Apache PDFBox, iText, JPOD gibi çeşitli kütüphaneleri kullanabiliriz.
  Her kütüphane için gerekli olan JAR dosyalarını indirmek ve projeye uygun olarak yerleştirmek (classpath) gerekir.

Ancak sadece kütüphanelerin indirilmesi ve projeye dahil edilmesi yetmeyecektir.
Ayrıca her yeni güncelleme sonrası güncel dosyaların takip edilmesi gerekecektir.
Maven proje dosyasına eklenen bağımlılıklar ile kolay bir şekilde indirmeyi ve proje yerleştirmeyi sağlar.
Kullanılan kütüphaneler proje dosyasında yer aldığından taşınabilirlik sağlanmış olur.
Sunmuş olduğu dizin yapısı sayesinde diğer geliştiricilerin projeyi takibini kolaylaşır.

- Spring-SpringBoot

  Spring, Java ile uygulama oluşturmaya yönelik kolaylaştırılmış ve modüler bir yaklaşım sağlayan açık kaynak bir projedir. Spring projeleri ailesi Java geliştirmenin erken aşamalarındaki karmaşıklıklara bir yanıt niteliğinde 2003 yılında başlamıştır ve Java uygulamaları geliştirmeye yönelik destek sunmaktadır. Spring adı tek başına genellikle uygulama çerçevesinin kendisini veya tüm proje ya da modül grubunu ifade eder. Spring Boot, Spring çerçevesinin bir uzantısı olarak oluşturulmuş belirli tek bir modüldür.
  Spring çerçevesinin, Spring Boot'un ve Java'nın birlikte nasıl çalıştığı ile ilgili bu bilgileri de edindiğimize göre Spring Boot'u, Java çerçevesi olan Spring içinde web uygulaması ve mikro hizmetler geliştirmeyi kolaylaştırıp hızlandıran araç olarak tanımlayabiliriz.
  Bazen, dil ve/veya platform Java olduğundan, "Java Spring Boot", "Java Spring çerçevesi" veya "Spring Boot çerçevesi" olarak adlandırılır. Ancak Java için Yürüt ve Hazırda Beklet gibi başka üçüncü taraf çerçeveler olduğundan, "Spring" ve "Spring Boot" demek daha doğru olur.

- Restful Api

  RESTful API, iki bilgisayar sisteminin internet üzerinden güvenli bir şekilde bilgi alışverişi yapmak için kullandığı bir arabirimdir. Çoğu iş uygulaması, çeşitli görevleri gerçekleştirmek için diğer dahili ve üçüncü taraf uygulamalarla iletişim kurmak zorundadır. Örneğin, aylık olarak maaş bordroları oluştururken faturalandırmayı otomatik hale getirmek ve dahili bir zaman çizelgesi uygulamasıyla iletişimde olmak için dahili hesap sisteminizin, verileri müşterinin bankacılık sistemiyle paylaşması gerekir. RESTful API'ler; güvenli, güvenilir ve verimli yazılım iletişim standartlarını izlediğinden bu bilgi alışverişini destekler.

- BootStrap

  Bootstrap, kullanılabilir kod parçalarından oluşan açık kaynaklı ve ücretsiz bir web uygulaması geliştirme araç takımıdır. Sahip olduğu CSS ve JavaScript taslakları, web sitelerinin ve mobil uygulamaların kullanıcılara görünen bileşenleri için kullanılır. HTML, CSS, Less, Sass ve JavaScript ile yazılmış olan Bootstrap, tamamen etkileşimli ve duyarlı web uygulamaları geliştirmek için kullanılabilecek öğrenmesi kolay bir alternatiftir.

- ReactJS

  Kullanıcı arayüzü oluşturmak için kullanılan React en çok tercih edilen JavaScript kütüphanesidir. Aracın bileşenleri Facebook tarafından geliştirilmiş ve 2013 yılında açık kaynak kodlu bir JavaScript olarak piyasaya sunulmuştur. Netflix, American Express, WhatsApp ve Instagram gibi büyük şirketler tarafından da kullanılan React kullanımı kolay bir araçtır. Kullanıcı arayüzü görevlerini verimli şekilde tamamlamaya odaklanan araç, birkaç gün kullanım sonucunda kolayca anlaşılabilir.

V. Yalanci Kod

BAŞLA

Başla
Öncelikle java ile jsoup kütüphanesi sayesinde web Scrabing işlemlerini gerçekleştirdik.
Ardından gelen verileri gerekli fonksiyonlara göndererek if koşulları ile eşleşme kontrollerini gerçekleştirdik.
Sonra bu eşleşen verileri yeni bir tabloya kaydettik. Spring boot,mongodb teknolojilerini kullanarak Restful apı oluşturduk ve
anotasyonlar sayesinde frontend kısmına verileri aktardık.
Frontend kısmında sayfa ilk çalıştığında önce bir yükleniyor işareti çıkıyor ve o sırada backend tarafından
axios.get() işlemi ile verileri çekiyoruz.Veriler geldiği anda sayfa yükleniyor ve karışık bir şekilde ana sayfaya basılıyor
ana sayfada bulunan butonlar ve işlevlerini şu şekilde açıklayabiliriz:

büyükten küçüğe fiyat sırala butonu:
axios.get("/mysite/bilgi=fiyatbilgi2=y")
.then(response =
setSampleData(response.data)
).catch(error = console.log(err))
bu butona basıldığında backend kısmmına bir http isteği gönderilir ve backend de hazırlanan fonksiyonlar sayesinde veriler fiyatlarına göre büyükten küçüğe sıralı bir şekilde ekrana gelir.

büyükten küçüğe puan sırala butonu:
axios.get("/mysite/bilgi=puanbilgi2=y")
.then(response =
setSampleData(response.data)

).catch(error = console.log(err))
bu butona basıldığında backend kısmmına bir http isteği gönderilir ve backend de hazırlanan fonksiyonlar sayesinde veriler puanlarına göre büyükten küçüğe sıralı bir şekilde ekrana gelir.

küçükten büyüğe fiyat sırala butonu:
axios.get(”/mysite/bilgi=fiyatbilgi2=d”)
.then(response =
setSampleData(response.data)
).catch(error = console.log(err))
bu butona basıldığında backend kısmmına bir http isteği gönderilir ve backend de hazırlanan fonksiyonlar sayesinde veriler fiyatlarına göre küçükten büyüğe sıralı bir şekilde ekrana gelir.

küçükten büyüğe puan sırala butonu:
axios.get(”/mysite/bilgi=fiyatbilgi2=d”)
.then(response =
setSampleData(response.data)
).catch(error = console.log(err))
bu butona basıldığında backend kısmmına bir http isteği gönderilir ve backend de hazırlanan fonksiyonlar sayesinde veriler puanlarına göre küçükten büyüğe sıralı bir şekilde ekrana gelir.

Header kısmında bir buton daha bulunuyor ancak bu buton admin girişi için kullanılan butondur.Butona basıldığında kullanıcı adı ve şifresi doğruysa admin paneli açılır örnek bir kod parçası aşşağıdaki gibidir.
if(inputs.name === ”fatih” inputs.password === ”123456”)
window.location.replace(”/admin”)

ana sayfada filter menüsünü şu şekilde açıklayalım:
Bu menüde markalar,fiyat aralıkları,ram boyutları,disk boyutları,disk tipleri,işletim sistemleri gibi özellikler bulunmaktadır.
¡details
¡input
type=”checkBox”
name=”fiyat”
value=”15000-25000”
/

.
.
.
¡/details
Bu özelliklerden herhangi biri seçildiğinde sayfa otomatik olarak yenilenir ve o seçilen özelliğe ait verileri getirir. marka özelliklerinden 1 tanesi seçildiği anda marka seçenekleri kilitlenir ve seçim yapmayı engeller ancak diğer özelliklerden seçim yapma kısıtı yoktur.Yani aynı özellikten 1 den fazla seçenek seçmek engellenmektedir.
Search işlemine gelirsek şöyle bir örnek kod ile açıklayalım:
const onChange = (event) =
setSampleData(event.target.value)

bu fonskşiyonda sampleData her güncellendiğinde App.js dosyasındaki şu kod devreye giriyor:
useEffect(()=
const filtered = samplData.filter((element)=
element.siteIsmi.toLowerCase().indexOf(search.toLowerCase())
!== -1 OR
element.modelNo.toLowerCase().indexOf(search.toLowerCase())
!==-1OR
element.marka.toLowerCase().indexOf(search.toLowerCase())
!==-1
})
setData(filtred)
,[sampleData])

Bu kod parçasında girilen input site ismine , markaya yada model numarasına uyuyor mu diye kontrol yapılı ve koşulu sağlayan elemanlar ekrana basılır.
Admin panelini ise şu şekilde tanıtalım:
admin panelinde ana safada oluşan verilerin card componenti içerisinde ekstra delete butonu,ve save butonu bulunmaktadır.Admin panelinin headerında ise search inputu ve add product butonu bulunmaktadır.
add product butonuna basılınca product sayfasına yönlendirir ve gerekli bilgileri girebileceği inputlar bulunmaktadır.inputlar eksiksiz şekilde doldurulursa kaydet butonuna basıldığında
axios.post(”/add”,inputs)
.then(()=
window.location.replace(”/”)
)
.catch(error = console.log(error))
kod parçacığı çalışır ve backend de hazırlanan
@postMapping() işelemi çağırılır.Bu işlem çağırıldığında gerekli fonksiyonlar çağırılarak gelen veriler veri tabanına eklenir.
Admin panelinde delete butonuna basıldığında:
axios.delete(”delete$id”)
.then(window.location.replace(”/”)
.catch(err = console.log(err)))
kod parçacığı çalışır ve backend de hazılanan
@deleteMapping() işlemi çağırılır gelen id ye ait verileri silmek için gerekli fonksiyonlara gönderilir ve veri tabanından silme işlemi gerçekleşir.
Admin panelinde save butonuna basıldığında:
axios.put(”save$id”,inputs)
.then(window.locaiton.replace(”/admin”))
.catch(err= console.log(err))
kod parçası çalışır ve backend de hazılanan
@putmapping() işlemi çağırılır ve gönderilen id ye ait veriler gönderilen veriler ile güncellenir.
ana sayfada başlığa tıklandığında product sayfası açılır ve tıklanılan ürüne ait bilgiler ekrana gelir bu sayfada
axios.get(”/mysite$id”)
.then(res =
setİnputs(res)

)
.catch(err= console.log(err))
kod parçacığı çağırılır ve veriler sayfa yenilensede gözükmeye devam eder.
cimri sitesine gelirsek:
Bu sitede ise search ile çalışan fonksiyondan bir parça kod şöyle gösterebiliriz:

useEffect(()=
const filtred = sampleData.filter(
(element) =
element.map((e) =
if (
(e.marka
e.marka.toLowerCase().indexOf(search.toLowerCase()) !==-1) OR
e.siteIsmi.toLowerCase().indexOf(search.toLowerCase())
!==-1 OR
e.modelNo.toLowerCase().indexOf(search.toLowerCase() !== -1))
return true;

return false;
)
.indexOf(true) !== -1
)
bu kod parçası girilen inputun markaya yada site ismine yada model numarasına eşit olanları ekrana basar.
card componentinin içinde fazladan site ismi ve ifyat gösterebilmek için şöyle bir kod parçası yazdık:

let trendyolSayac = 0
let hepsiburadaSayac = 0
const siteIsmi = []
const fiyat = []
const siteLinki = []

props.SiteIsmi.forEach((element) =
if (element.siteIsmi.toLowerCase() === ”trendyol” trendyolSayac === 0)
trendyolSayac++
siteIsmi.push(element.siteIsmi)
siteLinki.push(element.siteLinki)
fiyat.push(element.fiyat)
else if (
element.siteIsmi.toLowerCase() === ”hepsiburada”
hepsiburadaSayac === 0
)
hepsiburadaSayac++
siteIsmi.push(element.siteIsmi)
siteLinki.push(element.siteLinki)
fiyat.push(element.fiyat)
else if (
element.siteIsmi.toLowerCase() !== ”trendyol”
element.siteIsmi.toLowerCase() !== ”hepsiburada”
)
siteIsmi.push(element.siteIsmi)
fiyat.push(element.fiyat)
siteLinki.push(element.siteLinki)

)
BITIR

VI. KAYNAKCA

- https://www.bezkoder.com/spring-boot-mongodb-crud/
- https://github.com/bezkoder/spring-boot-mongodb-pagination/blob/master/src/main/java/com/bezkoder/spring/data/mong
- https://reactjs.org/

- https://jsoup.org/