Web İndeksleme Uygulaması

Cemre Can Kaya   
Bilgisayar Mühendisliği  
Kocaeli Üniversitesi  
190201137

Engin Yenice  
Bilgisayar Mühendisliği  
Kocaeli Üniversitesi  
190201133

*Özet*— Verilen bir URL’deki web sayfa içeriğine göre diğer birden fazla web sayfasını benzerlik bakımından indeksleyip sıralayan web tabanlı bir uygulama geliştirmek. Böylece bu proje sayesinde web indeksleme yöntemleri hakkında bilgi edinilmesini ve web tabanlı uygulama yazma becerisinin geliştirilmesi amaçlanmaktadır.

Anahtar Kelimeler—anahtar, kelime, frekans, havuz, liste, website, link, url, semantic, benzerlik

# Giriş

Programın arka planı (backend) **C#**, ön planı(frontend) **Angular Framework**’ü ile geliştirilmiştir. Verilen bir URL’deki web sayfa içeriğine göre diğer birden fazla web sayfasını benzerlik bakımından indeksleyip sıralayan web tabanlı bir uygulama geliştirilmiştir. Bu proje sayesinde web indeksleme yöntemleri hakkında bilgi edinilmiştir.

# Yontem

## Eklenecek

# Proje kodlama yapıları

Projenin arka planı (backend) kurumsal mimariye uygun bir şekilde geliştirilmiştir. Proje yapısı gereği 5 temel katmana parçalanmıştır.

Projenin ön planı (frontend) parçala yönet mantığı ile geliştirilmiş olup her bir işlem parçacığı ayrı bir bileşende (component) üzerinde yapılmaya çalışılmıştır.

## Arka Plan (Backend) Yapısı

Proje yapısı gereği 5 temel katmana parçalanmıştır. Bu katmanların detayları bu başlık altında açıklanmıştır.

### Core Katmanı

Bu katman proje içerisinde bulunması gereken temel bileşenleri bulundurmaktadır.

#### **Entities:**

Oluşturduğumuz nesnelerin daha somut ve yönetilebilmesi için temel arayüzleri (interface) bu klasör içerisinde tutulmaktadır.

* **IEntity:** Temel nesnelerimizin temel arayüz (interface) sınıfıdır.
* **IDto:** Uzun ismi **Veri iletim nesnesi (Data transfer object)** olarak geçmektedir. Temel nesnelerimizden Kullanıcı arayüzüne (UI (User Interface (Frontend))) göndermek istediğimiz nesne tanımlamalarının temel arayüz (interface) sınıfıdır.

#### **Utilities:**

Projenin genelinde kullanabileceğimiz araçlar bu klasör altında tutulmaktadır.

* **Results:** İş (Business) katmanında kullanacağımız methodların geriye dönüş değerlerinin daha yönetilebilir ve daha düzenli olması için oluşturduğumuz bir araç sınıfıdır. Bu sınıfı temel olarak özetlenecek olursak. İçerisinde temel olarak 2 adet değişken bulunmaktadır. Bu değişkenler mesaj ve başarı (message, success) durumu olarak isimlendirilmektedir. Veri gönderilmesi durumunda miras verdiği alt sınıfda ise data (veri) değişkeninin bulunduğu ayrı bir dönüş tipi bulunmaktadır. Sınıf içerisinde ki değişkenlerin daha kolay yönetilmesi için başarı (success) durumuna göre alt sınıflar oluşturulmuştur. Bu sınıfların çağrılması durumunda başarı durumu otomatik olarak belirlenmektedir.

### Entities Katmanı

Entity ve Dto arayüzlerinden (interface) örnek alan veri sınıfları bulunmaktadır.

#### **Concrete**

Entity arayüzünden (interface) örnek alınan veri sınıfları bulunmaktadır.

#### **Dto:**

Dto arayüzünden (interface) örnek alınan veri sınıfları bulunmaktadır.

### DataAccess Katmanı

Veri havuzunun yönetilmesinden görevli katmandır. Bu katmanımızı ileride geliştirmeye açık olması için arayüzler (interface) kullanarak geliştirdik. Bu sayede ileride bir gerektiğinde başka yapılara geçmeyi planladık. Şuanda veri havuzumuzu bellekte (In Memory) olarak turuyoruz.

#### Abstract

Veri sınıflarımızın arayüzlerinin tutulduğu klasör.

#### Concrete

Veri sınıflarımızın tutulduğu klasör.

* **WordToExcludeDal:** Anahtar kelime olmasını istemediğimiz kelimelerin listesinin tutulduğu bir listedir.
* **TagAndPointDal:** Site içerisinde belirli taglara ait özel puanlama yapılmaktadır. Bu puanlama listesinin puanlarının belirlendiği liste bu sınıfta tutulmaktadır.

|  |  |
| --- | --- |
| **HTML Etiketler** | **Puan (Her bir kelime için)** |
| **Title** | 10 |
| **H1** | **9** |
| **H2** | **8** |
| **H3** | **7** |
| **H4** | **6** |
| **H5** | **5** |
| **H6** | **4** |
| **B** | **3** |
| **Strong** | **3** |
| **U** | **2** |
| **P** | **2** |

* **MemoryGlobalSemanticWord:** Semantik kelimelerin bulunduğu bir listeyi bellekte tutmaktadır. Program ilk açıldığında bu liste hafızaya alınmaktadır.

### Business Katmanı

Projenin iş kodlarının yazıldığı katmandır. Bu gerekli işlemlerin yönetildiği katmandır. Temel olarak 4 klasöre bölünmüştür.

#### Abstract

Business sınıflarımızın arayüzlerinin tutulduğu klasör.

#### Concrete

Business sınıflarımızın tutulduğu klasör.

* **IndexerManager:** WebAPI tarafından gönderilen isteklerin gerekli helper sınıflarına yönlendirilmesini sağlayan ve dönen sonuçların geri WebAPI tarafına yönlendiren sınıftır.

#### DependencyResolvers

Bağımlılıkların çözülmesi ve isim havuzuna aktarılması için kullanılmaktadır. (Paket olarak Autofac kullanılmıştır.)

#### Helpers

Projenin akışını sürdürecek helper sınıfları yazılmıştır. Helper sınıflarımızın arayüzlerinin tutulduğu abstract klasörü ve örnek alındığı concrete klasörü bulunmaktadır.

* **WebSiteOperation:** Gönderilen web site adresi ile iletişim kurmakla görevlidir. Gönderilen web sitelerinin bilgilerini tespit etmekle görevlidir.
* **HtmlCleaner:** Gönderilen web site nesnesi içerisinde bulunan html kodlarının temizlenmesi görevini yapmaktadır.
* **KeywordOperation:** Gönderilen website nesnesi üzerinde kelime, anahtar kelime, frekans, semantic kelime bulma görevlerini yapmaktadır.

### WebAPI katmanı

Ön Plan (Frontend) kısmından gelen istekleri karşılayıp gerekli dönüşleri yapmakla görevli olan katmandır. Ön Plan (Frontend) tarafından 5 farklı isteğe karşılık verebilecek 5 adet Controller bulunmaktadır.

#### StageOneController

Sayfada geçen kelimelerin frekanslarını hesaplayan methodlarını çağırmakla görevlidir. (1. Madde)

#### StageTwoController

Anahtar kelime çıkarma methodlarını çağırmakla görevlidir (2. Madde) [Otomatik olarak 1. Madde ile bağlantılıdır.]

#### StageThreeController

URL ve URL havuzu arasındaki benzerlik skorlaması methodlarını çağırmakla görevlidir (3. Madde) [Otomatik olarak 2. Madde ile bağlantılıdır.]

#### StageFourController

Site indexleme ve sıralama methodlarını çağırmakla görevlidir. (4. Madde) [Otomatik olarak 3. Madde ile bağlantılıdır.]

#### StageFiveController

Semantik analiz methodlarını çağırmakla görevlidir (5. Madde) [Otomatik olarak 4. Madde ile bağlantılıdır.]

## Ön Plan (Frontend) Yapısı

Projenin kullanıcı arayüzü (User Interface) ekranıdır. Arka plan (Backend) tarafında yazılan kodların kullanıcıya görüntüsel olarak aktarıldığı taraftır. Angular Framework ile geliştirilmiştir. NG-ZORRO bileşen (component) paketi kullanılmıştır.

### Ana Bileşen (Component) (app.component)

Projenin çalıştığı ana bileşendir. Bu bileşen içerisinde sabit ve dinamik bileşenler çağrılmaktadır.



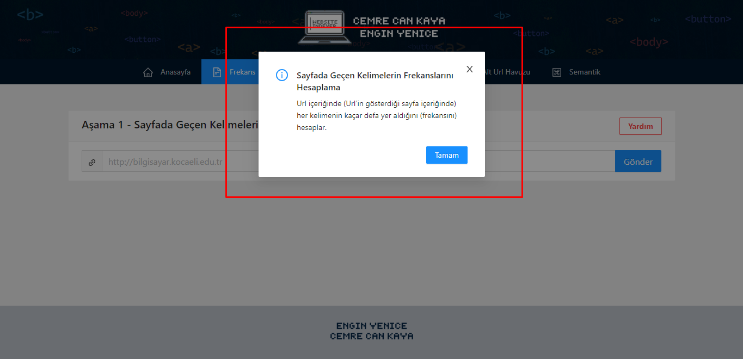
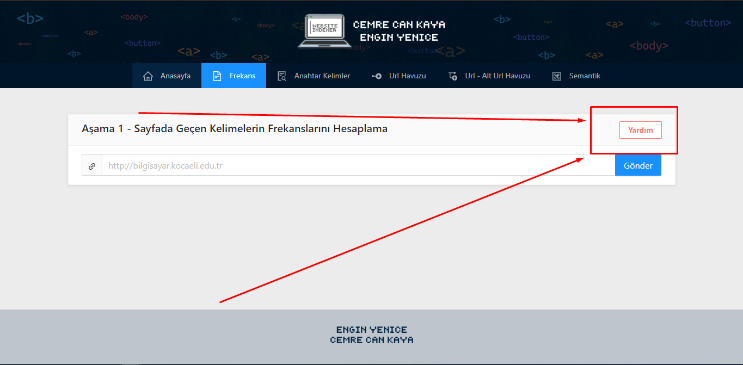
Sabit bileşenler direk ana bileşen içerisinden çağrılmaktadır. Dinamik bileşenler ise rotalama (routing) yapısı ile ekrana getirilmektedir.

# Nasıl Kullanılır

## Temel Birleşenler

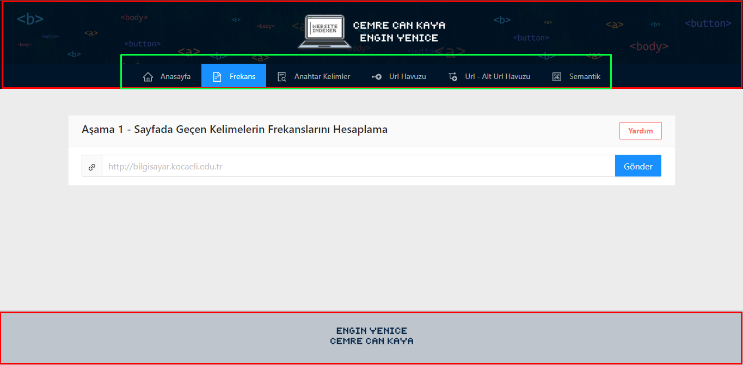
### Yardım

Kullanıcının bulunduğu sayfanın temel işleyişiyle ilgili bilgiler vermektedir.



### Menü ve Footer

Sayfalar arası geçişi yöneten ve sayfa hakkında bilgi veren alanlardır.



## Anasayfa

Eklenecek

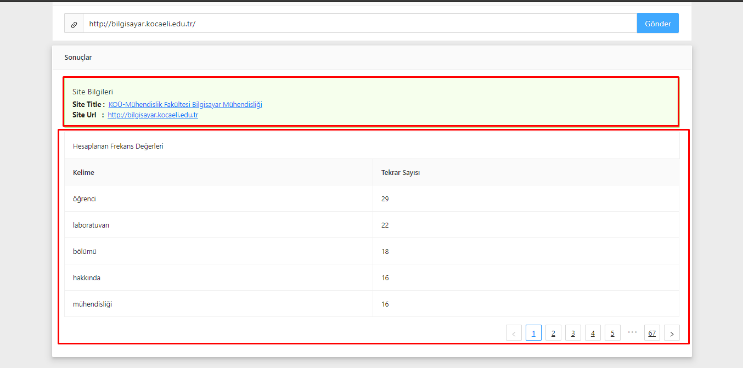
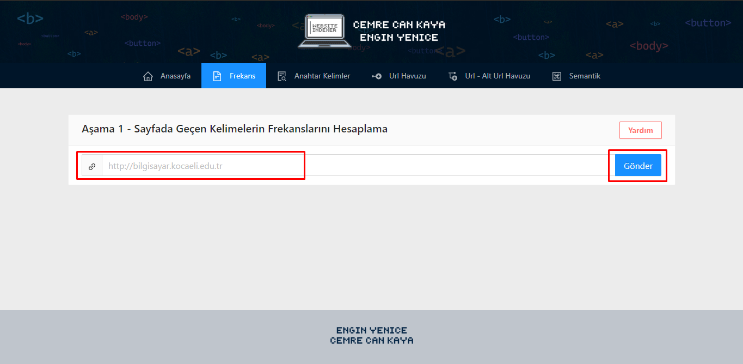
## Frekans

Sayfanın ortasında bulunan veri girişi nesnesi (input) içerisine control edilmesi istenen link girilmektedir. Ardından gönder butonuna tıklayarak site hakkında sonuçlara ulaşabilirsiniz.

Gelen sonuç içerisinde temel site bilgileri ve frekans değerleri gösterilmektedir.

Frekans tablosu:

Kelime ve site içerisinde geçen tekrar sayısı gösterilmektedir.



## Anahtar Kelimeler

Sayfanın ortasında bulunan veri girişi nesnesi (input) içerisine control edilmesi istenen link girilmektedir. Ardından gönder butonuna tıklayarak site hakkında sonuçlara ulaşabilirsiniz.

Gelen sonuç içerisinde temel site bilgileri ve belirlenen anahtar kelimeler gösterilmektedir.

Anahtar Kelime Tablosu:

Anahtar kelime, site içerisinde geçen tekrar sayısı ve hesaplama sonucu aldığı skor değeri gösterilmektedir.

