

Akademik Arama Sistemi
KOCAELİ ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
YAZILIM LAB. II- 3. Proje

Oğuzhan Koç-190201184 Yusuf Arslan-190201183

oguzhankoc9855@hotmail.com

ysf.arslan2017@gmail.com

ÖZET

Yazılım Laboratuvarı 2 dersi 3.projesi kapsamında bir akademik arama motoru geliştirmeniz beklenmektedir. Akademik arama motoru, akademik çalışmalara ve araştırmacılara ulaşabileceğiniz bir web sayfasıdır. Bu proje ile birlikte gerçek dünya ihtiyaçlarını karşılayacak bazı yetenekler kazandırılacaktır. Bu proje ile web ve veri tabanı uygulamalarının kullanmamız ve web tabanlı bir akademik arama sayfası yapmanız beklenmektedir. Projede veri tabanı olarak Neo4j graf veri tabanı kullanılmıştır.

GİRİŞ

Projede bizden istenilen bir akademik arama yapılan web uygulamasıdır. Bu web uygulamasında iki farklı arayüz bulunmaktadır. Bu arayüzler Yönetici arayüzü ve Kullanıcı arayüzüdür.Yönetici arayüzünde veri tabanına giriş yapılmaktadır.Veri tabanına email adresi ve şifresini kullanarak giriş yapılmaktadır.Veri tabanına araştırmacılara ait bilgiler olan araştırmacıId,araştırmacıAdı,araştırmacıSoyadı,yayınAdı,yayınYılı,yayınYeri ve yayın türü bilgileri de eklenebilmektedir.Kullanıcı arayüzünde ise kullanıcının arama yapabileceği bir sayfa bulunmaktadır.Bu arama motorunda adı soyadı yayın adı yayın yılı yayın türü ve yayın yerine göre arama yapılabilir.

Veri tabanı olarak kullandığımız neo4j de araştırmacı ,yayınlar ve tür tabloları bulunmaktadır.Araştırmacı tablosunda araştırmacının adı ,soyadı;yayın tablosunda yayın adı ve yayın yeri;tür tablosunda ise yayın türü ve yayın yeri bulunmaktadır. Kullanıcının arama sonucuna göre gelen araştırmacının yayınlarında eğer istenirse araştırmacıları grafları üstlerine tıklamak şartıyla gösterilebilir.

I.YÖNTEM

Program Python programlama dili kullanılarak geliştirilmiştir.Bizden istenen veri tabanı olarak Neo4j Aura kullanılmıştır.Framework yapısı olarak Flask frameworkünü tercih edilmiştir. Geliştirme ortamı olarak ise “Visual Studio Code”kullandık.Projemize başlarken öncelikle yol haritamızı çıkardık. Projenin isterlerini analiz edip bu isterler üzerine araştırmalar yaptık.Bu araştırmalarımıza öncelikle kullanacağımız dile karar vermek oldu.Bunun için pythonda karar kıldık.Sonra framework için hangi framework’ü kullanacağımız hakkında kararsız kaldık.Ya önceden kullandığımız Djangoyu yada Flask’ı tercih edecektik.Flask’ın kullanım rahatlığından dolayı Flask’i tercih ettik.Bundan sonraki adım olarak ise database projeye dahil etmekte.Ama bunu denerken Neo4j desktop’ı prçalıştıramadık.Bizde Neo4j Aura’yı projeye entegre etmeye karar verdik.Projemize database’i entegre ettikten sonra tabloları ekledik.Database kullandığımız tablolar şu şekildedir:

User :Bu tabloda adminin kişisel bilgileri(name,password), bulunur.Bu sayede normal kullanıcılar admin ekranına geçiş yapamazlar.

Arastirmaci:Bu tabloda araştırmacının adı ve soyadı tutulur.Bu sayede arama yaparken kullanıcının adına ve soyadına göre yeni bir sorgu gönderebilir yada yeni bir node oluştururken burdaki bilgileri çekebiliriz

Yayin:Bu tabloda yayınların yayın adı ve yayın yılı tutulur.Bu sayede arama yaparken kullanıcının yayın adı ve yılına göre yeni bir sorgu gönderebilir yada yeni bir node oluştururken burdaki bilgileri çekebiliriz

Tur:Bu tabloda araştırmacının yayının türü ve yayın yeri tutulur.Bu sayede arama yaparken kullanıcının adına ve soyadına göre yeni bir sorgu gönderebilir yada yeni bir node oluştururken burdaki bilgileri çekebiliriz.

Kullanılan Sınıflar:

app.py:Bu sınıfta kullanılan fonksiyonlar ile web sayfasına bağlantılar kurulur ve database veriler bağlanır.Bu sınıfta kullanılan önemli fonksiyonlar şöyledir:

baglantikur:Neo4j aura database’ine bağlantı kurulur.Kullanıcının girdiği verilerin gönderilmesini sağlar.

baglantisonlandir:Neo4j aura database ile bağlantıyı kapatır.

vis:Kullanıcının yaptığı search sonucu gelmiş olan verilerden araştırmacı adlarına göre graph yapılarını gösterir.Bunun için DB clasından bilgi alır ve vis.html ye bilgi gönderir.

home:Bu fonksiyon sayesinde kullanıcı eğer admin ise admin ekranına geçiş yapar eğer değilse arama motoruna aramak istediği bilgileri yazarak bilgileri ekranda görür ve istemesi durumunda kullanıcıların graflarını görür.

adminhome:Adminin adını ve şifresini girerek giriş yapıp ekleme yapacağı ekrana geçiş yaptığı fonksiyondur.

admingiris:Bu fonksiyon sayesindecreateBaglanti createTur
createArastirmaci createYayin fonksiyonlarına geçiş yapılabilir.

createBaglanti:Database de var olan tür,araştırmacı ve yayınlar arasında
bağlantı kurulmasını sağlar

createTur:Database'e yeni bir tür ekler

createArastirmaci: Database'e yeni bir araştırmacı ekler

createYayin: Database'e yeni bir yayın ekler.

DB.py:Bu sınıftaki fonksiyonlar sayesinde Neo4j aura'dan veriler
çekilebiliyor.Database'e yeni bir veri eklenebiliyor yada arama yapılabilir.

Bu sınıfta kullanılan önemli fonksiyonlar şöyledir:

create_arastirmaci:Bu fonksiyon sayesinde database de yeni bir
araştırmacı ekleyebiliyoruz.

arastirmaci_sorgu:Database'deki araştırmacıların bilgilerini çekebiliyoruz.

create_tur: Bu fonksiyon sayesinde database de yeni bir
tur ekleyebiliyoruz.

tur_sorgu: Database'deki tür bilgilerini çekebiliyoruz.

create_yayin: Bu fonksiyon sayesinde database de yeni bir
yayın ekleyebiliyoruz.

yayin_sorgu: Database'deki yayın bilgilerini çekebiliyoruz.

a_y_c_sorgu: Database'deki araştırmacı ve yayın arasındaki bağlantıların
bilgilerini çekebiliyoruz.

y_t_c_sorgu: Database'deki tür ve yayın arasındaki bağlantıların
bilgilerini çekebiliyoruz.

create_yayin_to_tur_connection:Yayın ve tür arasında bağlantı kurar.

create_arastirmacilar_to_yayin_connection:arasştırmacı ve yayın arasında
bağlantı kurar

find_ArastirmaciAdi:Araştırmacı adını bulur.

find_Arastirmaci_wrote_yayin:Araştırmacının yazdığı yayınları bulur.

find_Yayin_to_everything:Kişinin yazdığı yayını bulup tüm bilgileri alan
fonksiyondur.

find_YayinYeri:Yayın yerini bulan fonksiyondur.

find_YayinTuru:Yayın türünü bulan fonksiyondur.

find_YayinAdi:Yayın adını bulan fonksiyondur.

search:İstenilen arama türüne göre buluna sonuçları geri döner.

DENEYSEL SONUÇLAR:

Admin

ArastirmaciAdi

Submit

ArastirmaciAdi	ArastirmaciSoyadi	YayinAdi	YayinYili	YayinTuru	YayinYeri
----------------	-------------------	----------	-----------	-----------	-----------

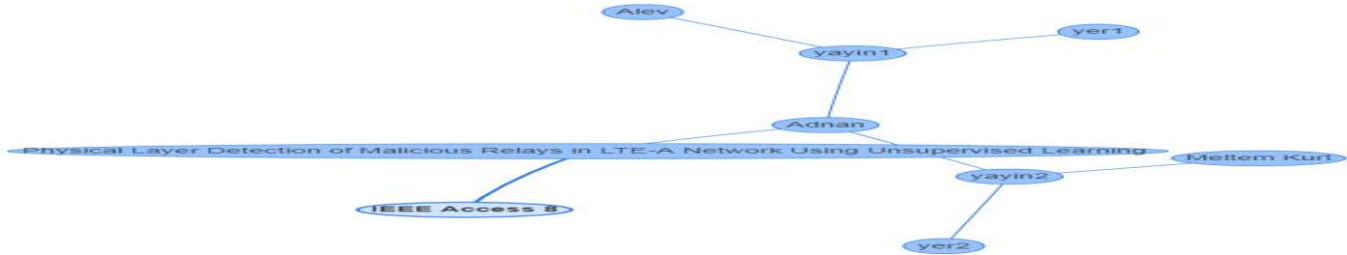
Admin

ArastirmaciAdi

Adnan

Submit

ArastirmaciAdi	ArastirmaciSoyadi	YayinAdi	YayinYili	YayinTuru	YayinYeri
Adnan	Kavak	yayin2	yil2	tur2	yer2
Adnan	Kavak	yayin1	yil1	tur1	yer1
Adnan	Kavak	Physical Layer Detection of Malicious Relays in LTE-A Network Using Unsupervised Learning.	2020	Makale	IEEE Access 8



kullaniciAdi

Admin

password

....

giris

[createBaglanti](#)[Create Tur](#)[Create Yayin](#)[Create Arastirmaci](#)

Choose a Tur: IEEE_Access_8

Choose a Arastirmaci: Adnan

Choose a Yayin: Physical_Layer_Detection_of_Malicious_Relays_in_LTE-A_Network_Using_Unsupervised_Learning.

Makale

[Gönder](#)

Yayın Türü(Tür Adı)

Yayın Yeri

[Gönder](#)

Yayın Adı

Yayın Yılı

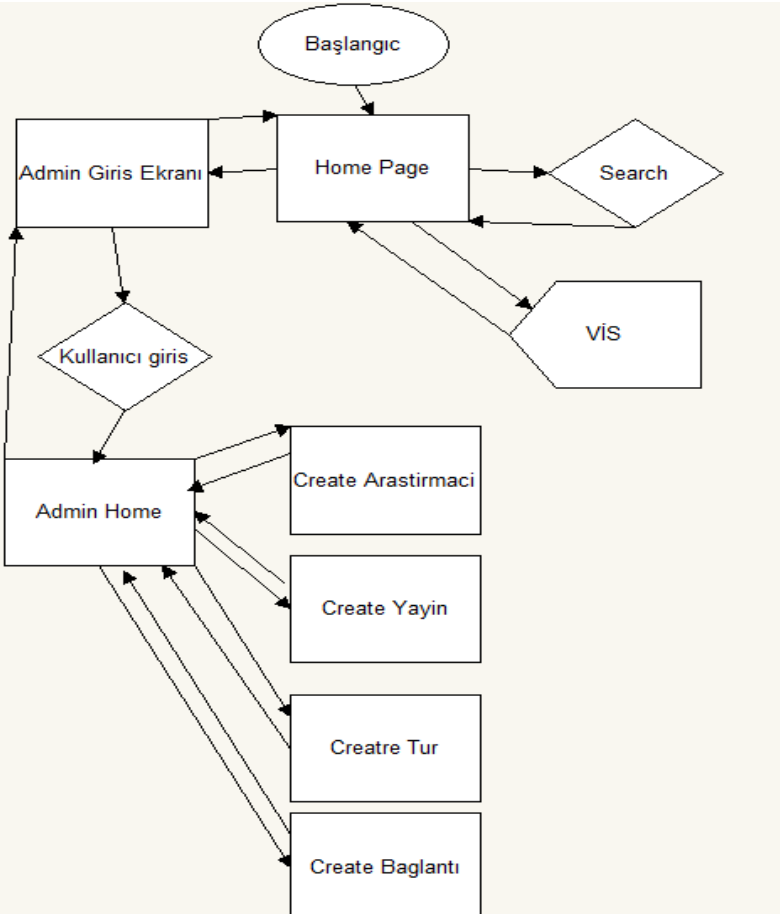
[Gönder](#)

Araştırmacı Adı

Araştırmacı Soyadı

[Gönder](#)

V. AKIS DIAGRAMI



VI. KAYNAKÇA

<https://flask.palletsprojects.com/en/2.0.x/>

<https://www.youtube.com>

<https://stackoverflow.com/>

<https://www.geeksforgeeks.org>www.w3schools.com/

<https://neo4j.com/docs/>