**Kocaeli Üniversitesi**

**Bilgisayar Mühendisliği Bölümü**

**Yazılım Laboratuvarı II**

**GPS Takibi İçin Web Sitesi**

*Alperen Tan Oğuz Narlı*

[*190201054@kocaeli.edu.tr*](mailto:190201054@kocaeli.edu.tr)[*180201074@kocaeli.edu.tr*](mailto:180201074@kocaeli.edu.tr)

**Giriş:**

Yazılım Laboratuvarı II dersinin kapsamında bizden firmaların kendilerine araçlarını harita üzerinden takip etmesini sağlayacak web uygulaması yazmamız beklenmektedir. Biz de bu doğrultuda Python, JavaScript ve HTML-CSS dillerini kullanarak bize verilen isterlere göre web sitesi geliştirdik. Bizden istenen şartları sağlamak için projede Django Web Framework, GoogleApi, Bootstrap CDN, MongoDB, SqLite ve AJAX yapılarını kullandık.

**Özet:**

Uygulamaya üye olan kullanıcı, kendisinin sahip olduğu araçların istenen tarih aralığında geçmiş koordinatlarını veri tabanından çekerek görüntüleyebilir ve dilerse Google Haritalar altyapısını kullanarak harita altığı üzerinden de güzergah kontrolü yapabilir.

**Kullanıcı Paneli:**

Geliştirdiğimiz web uygulamasında kullanılan back-end teknolojisi olarak geçtiğimiz bölümlerde bahsettiğimiz gibi Django kullanılmıştır. Web uygulamamızı geliştirilirken ilk yapılan kullanıcı panelinin geliştirilmesi olmuştur.

Bu doğrultuda ilk olarak Django’nun terminal üzerinde sağladığı kullanıcı oluşturma özelliği ile kullanıcı oluşturduk. Ardından Model kısmında kullanıcının tüm giriş çıkış zamanını tutan verileri kullanıcı yapısıyla ilişkisel olarak oluşturulmuştur. Ayrıca kullanıcıya bağlı kuryelerin bilgilerini tutan verinin modeli de tanımlanmıştır. Bu işlemlerin ardından devamında migrate edilip ilgili modellerle ilgili veri tabanı tablosu oluşturulması sağlanmıştır.

Bu işlemlerin ardından View(Controller) kullanıcı sisteme giriş çıkış yaptığında gerçekleşecek aksiyonla ilgili fonksiyonlar yazılmıştır. Bu fonksiyonlar return olarak request tipi, hangi html üzerinde aksiyon olacağı ve ilgili templatelere göndereceği verileri ve hangi form üzerinden işlem yapacağı döndürmektedir. Form üzerinde işlem yapabilmek için ise Django Forms yapısından formun içereceği bilgilerin tanımlanması önemlidir. Django’nun sağladığı object query yapısı ile veri tabanında kullanıcı giriş çıkış bilgileri kaydedilmiştir. Template kısmında ise giriş ekranı için crispy forms kullanılmıştır. Crispy forms pythonla yazılmış özel yapıdır. Crispy form ekrana Django Forms yapısında belirttiklerimize göre giriş kutucuğu oluşmaktadır. Formda butona basıldığında post request atılmaktadır. View kısmında buna göre request yollanmıştır.

Kullanıcıların giriş çıkış verisi içinde bir controller fonksiyonu mevcuttur. Bu fonksiyon kullanıcı verilerini döndürmektedir. Front-end kısmında tablo şeklinde sergilenmektedir.

**Araç Verilerinin Gösterilmesi:**

Harita için uygulamamızda Google Map kullanılmıştır. Bunun için ilk olarak haritanın olduğu template hazırlanmıştır. Harita için gerekli olan Javascript kodu Google’ın web sayfasından alınıp kullanılmıştır. Haritanın bulunduğu sayfa için gerekli olan aksiyonu sağlayan view(controller) fonksiyonu yazılmalıdır. Ancak bu fonksiyon sadece haritanın gösterileceği sayfaya yönlendirmek için kullanılır. Önemli olan verilerin anlık gösterilmesidir. Bunun için GET request alan asenkron çalışan ayrı bir controller yazılmıştır. Bu fonksiyonun front-end kısmında aksiyon alması için AJAX yapısı eklenmişti. AJAX yapısı sayesinde asenkron şekilde web sitemiz üzerinde başka sayfaya yönlendirilmeden işlem yapmamızı sağlamaktadır. Bu yapıyla 5 saniyede bir aksiyon alarak mevcut zamandan 30 dk öncesine kadar tüm verileri göstermektedir.

Araçların istediğimiz tarih aralığında verilere erişmemiz için yapıda geliştirilmiştir. Bu yapı için önce kullanıcıya bağlı araçlar tabloda gösterilmiştir. Bu tablodan araç seçtikten sonra başka sayfaya yönlendirilir ve araca dair tüm verileri gösterilir. Tablonun altında yer alan giriş kutucuklarında tarih aralığı girerek istenen saat aralığında veriler görüntülenebilmektedir.

**Yalancı kod:**

connect(database)

list[]

for i in vehicleIDs:

id=str(i.vehicle\_id)

query = {"vehicleID": id}

data = find(query, {'\_id': False})

for x in data:

list.append(x.copy())

car\_data["data"]=list

return jsonResponse(data)

show\_detail:

for x in data:

now = datetime.now()

now.replace(year=2018, month=11)

if (0 < ((changednowdate).total\_seconds()) < 1800):

dictlist.append(x.copy())

return dictlist

show\_sorteddata:

list2=[]

firstDate = form.data.get('firstDate')

secondDate = form.data.get('secondDate')

inputdate = datetime.strptime((firstDate), '%Y-%m-%d %H:%M')

inputdate2 = datetime.strptime((secondDate), '%Y-%m-%d %H:%M')

for x in data:

date = datetime.strptime((x['date']), '%Y-%m-%d %H:%M')

if ((inputdate <= date <= inputdate2) or (inputdate2 <= date <= inputdate)): list2.append(x.copy())

return list2

googlemap.html:

function(data):

clearallMarkers();

var today = new Date();

today.setFullYear(2018);

today.setMonth(10);

button.onclick(getelement);

var mark = data["data"];

for each vehicleID:

for (let i=0;i<mark.length;i++){

var dateData = new Date(mark[i]["date"]);

var diffMs = today.getTime() - dateData.getTime();

var diffmins = (diffMs / 1000) / 60;

if(diffmins>=0 && diffmins <=timeTxt){

placeMarker(lat,lng);

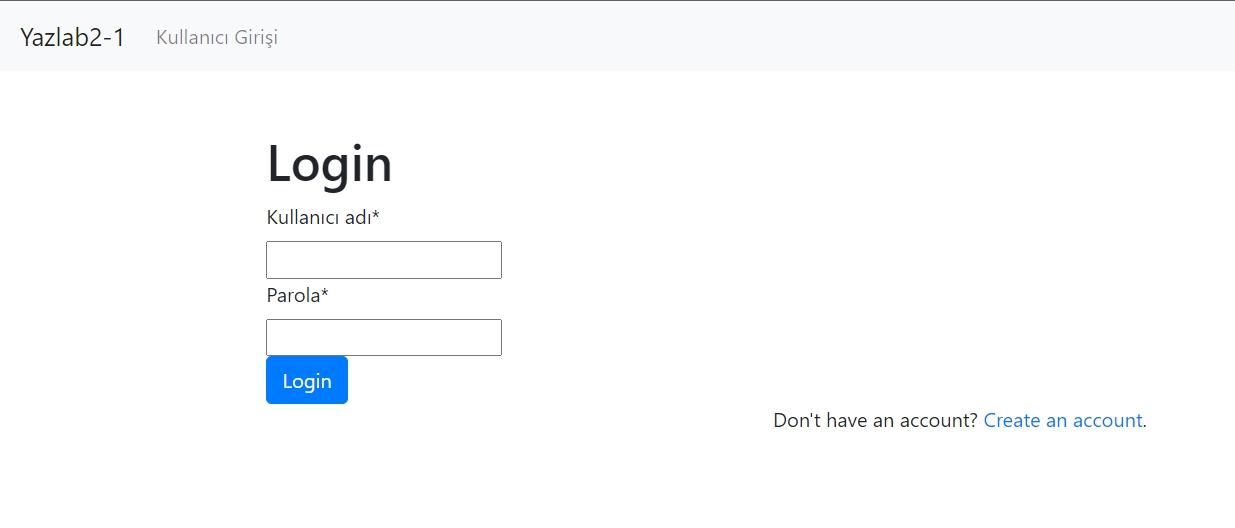
}

}

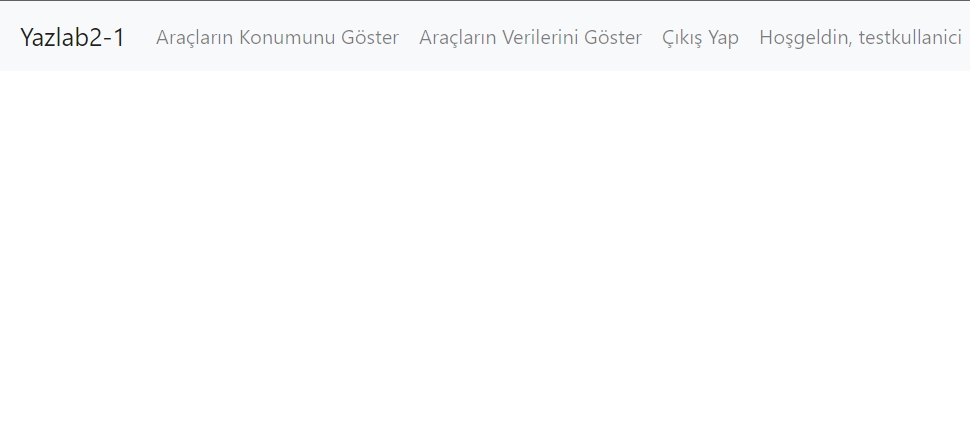
**Sonuç:**

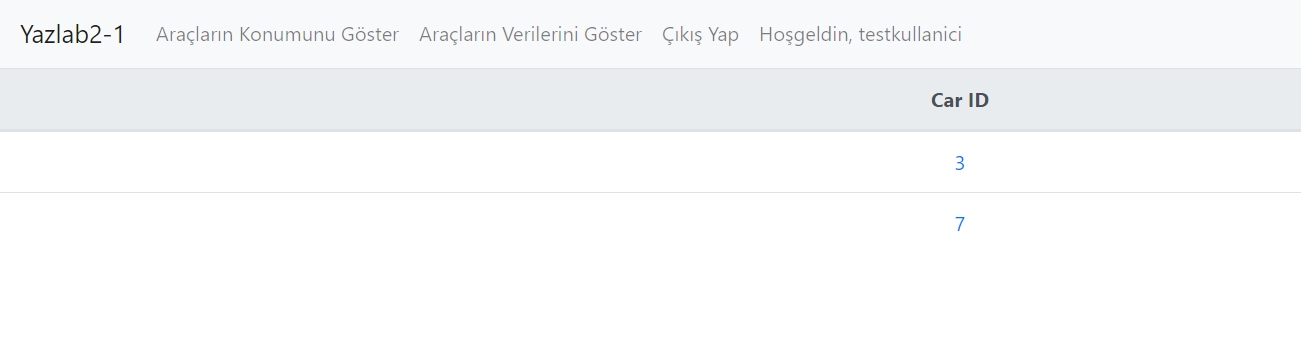
Bu projede Python ve JavaScript programlama diliyle web sitesi mantığının temelini ve veri tabanından çekilen verilerin anlık kullanıcılara yansıtılmasını kodlayarak öğrendik.

**Deneysel Sonuçlar:**

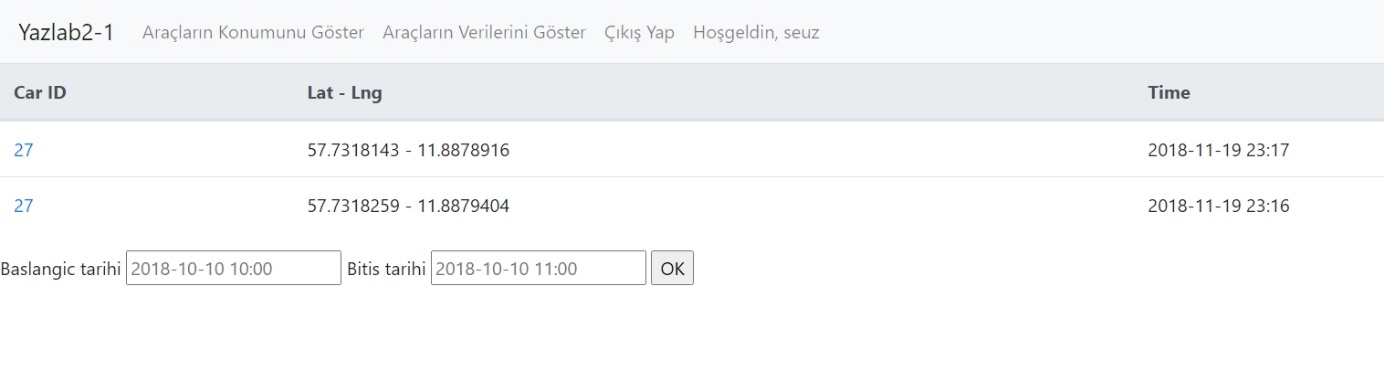
****

*Giriş ekranı*

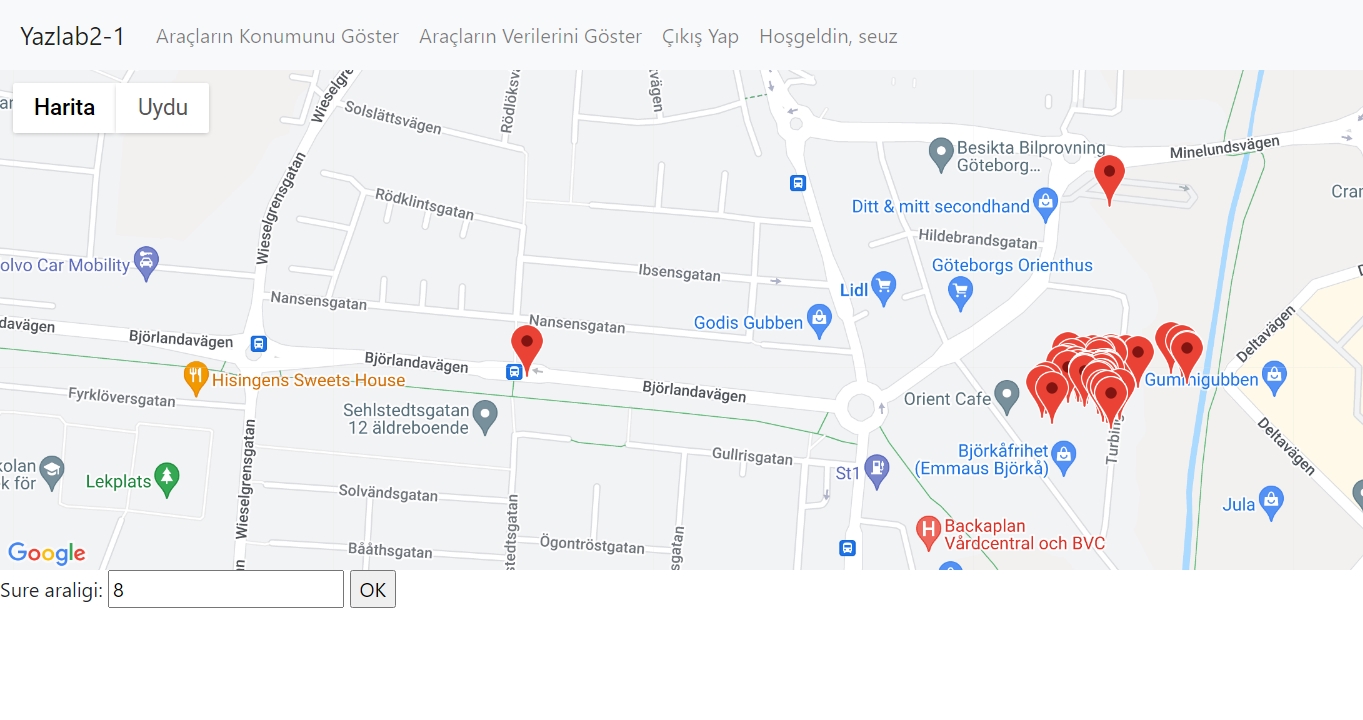
*****Ana ekran*

****

*Veri seçme ekranı*

****

*Tarih seçme ekranı*

**

*Harita ekranı*

**Kaynakça:**

[*https://tutorial.djangogirls.org/tr/*](https://tutorial.djangogirls.org/tr/)

[*https://ordinarycoders.com/blog/article/django-user-register-login-logout*](https://ordinarycoders.com/blog/article/django-user-register-login-logout)

[*https://www.pluralsight.com/guides/work-with-ajax-django*](https://www.pluralsight.com/guides/work-with-ajax-django)

[*https://developers.google.com/maps/documentation/javascript/adding-a-google-map*](https://developers.google.com/maps/documentation/javascript/adding-a-google-map)

[*https://www.youtube.com/watch?v=v2TSTKlrPwo*](https://www.youtube.com/watch?v=v2TSTKlrPwo)

[*https://stackoverflow.com/questions*](https://stackoverflow.com/questions)

[*https://www.w3schools.com/python/*](https://www.w3schools.com/python/)

[*https://www.mongodb.com*](https://www.mongodb.com)

[*https://www.geeksforgeeks.org/*](https://www.geeksforgeeks.org/)