

# Yazılım Laboratuvarı 2.Proje

## Ödevi

Batın Erke Öztürk  
Bilgisayar Mühendisliği

1.Öğretim

Kocaeli Üniversitesi

Kocaeli Türkiye

batinerkeozturk@gmail.com

Halil Han Kavak  
Bilgisayar Mühendisliği

1.Öğretim

Kocaeli Üniversitesi

Kocaeli Türkiye

halilhankavak10@gmail.com

**Abstract**—Bu doküman Kocaeli Üniversitesi Bilgisayar Mühendisliği bölümü ,Yazılım Laboratuvarı Dersi 2.proje ödevi için hazırlanmıştır.

### 1.Özet

Bu proje, kullanıcıların etkinlikler düzenleyebileceği, bu etkinliklere katılabileceği ve etkinliklere özel sohbet alanlarında iletişim kurabileceği bir Web Tabanlı Akıllı Etkinlik Planlama Platformu geliştirmeyi amaçlamaktadır. Platform, kullanıcıların etkinlik planlamasını kolaylaştırarak katılımcılar arasında etkileşim ortamı sunmayı hedeflemektedir. Etkinlik sahipleri, etkinlik detaylarını oluşturup yönetebilirken; katılımcılar, etkinliklere kayıt olma, detayları görüntüleme ve katıldıkları etkinliklerin sohbet alanlarında etkinlik hakkında konuşma,sohbet etme hakkına sahip olacaklardır.

### 2.Giriş

Bu projede, kullanıcıların etkinlik oluşturabileceği, etkinliklere katılabileceği ve kullanıcı ile etkinlik bilgilerini yönetebileceği web tabanlı bir sistem geliştirilmesi amaçlanmıştır. Kullanıcı arayüzü; giriş yapma, kaydolma ve bilgilerini güncelleme ekranları ile etkinlik ekleme, güncelleme, silme, etkinliklere katılma, etkinlik detaylarını görüntüleme ve bir admin sayfasını içermektedir. Kullanıcılar, kayıt işlemleri sırasında ad, soyad, doğum tarihi, profil fotoğrafı, cinsiyet, telefon numarası, konum, e-posta, hobiler ve şifre gibi bilgilerini sisteme girebilmektedir. Etkinlik ekleme ekranında, etkinlik adı, süresi, zamanı, açıklaması ve kategorisi gibi detaylar tanımlanabilmektedir. Etkinlik detay sayfasında, etkinlikle ilgili bilgiler görüntülenebilmekte ve her etkinlik için ayrı bir mesajlaşma paneli bulunmaktadır. Bu panel, gerçek zamanlı olmamakla birlikte, etkinlik katılımcılarının daha önce gönderdiği mesajları incelemeye olanak sağlar. Admin kullanıcıları ise kullanıcı ve etkinlik silme işlemleri yapabilir.

Kullanıcılar, bilgilerini güncelleyebileceği bir panelde profil fotoğrafı ekleyebilir veya değiştirebilir. Ayrıca, etkinliklere katılım ekranında seçilen etkinliğin bilgileri kontrol edilip uygunluk durumuna göre katılım sağlanabilmektedir.

### 3.Yöntem

Projemizde Backend tarafında **Java Spring Boot**, Frontend tarafında ise **React.js** kullanılmıştır. Veritabanı yönetim sistemi olarak **PostgreSQL** tercih edilmiş ve oluşturulan veritabanı, Backend ve Frontend ile entegre edilmiştir. Bu entegrasyon sayesinde, uygulama üzerinden veri ekleme, güncelleme ve sorgulama işlemleri gerçekleştirilerek verilerin dinamik bir şekilde işlenmesi sağlanmıştır.

Proje kapsamında **Admins, Events, Users, Participant, Creator, Point** ve **Messages** olmak üzere toplam yedi veritabanı tablosu oluşturulmuştur.

- **Admins** tablosu, yöneticilere ait adminid (primary key), adminname, adminpassword ve adminemail sütunlarından oluşmaktadır.
- **Events** tablosunda, etkinlik bilgileri için id (primary key), eventname, eventdescription, eventdate, eventtime, eventduration, eventlocation, eventcategory, eventlocationlat ve eventlocationlon sütunları yer almaktadır.
- **Users** tablosunda kullanıcı bilgileri için username, password, email, location, areaofinterest, name, surname, dateofbirth, gender, number, profilphotopath, locationlat ve locationlon sütunları bulunmaktadır.

- **Messages** tablosu, etkinlik sohbet mesajlarını saklamak için messageid (primary key), userid, eventid, messagetext ve sendingtime sütunlarından oluşmaktadır.
- **Participant** tablosunda etkinliklere katılım için participantid ve eventid sütunları primary key olarak tutulmaktadır.
- **Point** tablosunda kullanıcıların puan bilgileri pointid (primary key), userid, points ve date sütunları ile yönetilmektedir.
- **Creator** tablosu ise etkinlik düzenleyicilerinin creatorid ve eventid bilgilerini primary key olarak barındırmaktadır.

- Projenin harita bazlı işlevleri için **Google Maps API** kullanılmış ve konum bilgileri locationlat (enlem) ve locationlon (boylam) sütunları ile saklanmıştır. Kullanıcıların toplam puanları; katıldıkları etkinlik sayısına (her biri 10 puan), oluşturdukları etkinlik sayısına (her biri 15 puan) ve ilk katılım bonusuna (20 puan) göre hesaplanmıştır. Bu hesaplama işlemleri **PointController**, **PointRepository** ve **PointService** sınıflarıyla gerçekleştirilmiştir.

- Profil fotoğrafı yükleme işlemi, kullanıcının seçtiği dosyanın FormData nesnesi aracılığıyla backend'e gönderilmesiyle yapılmaktadır. axios kullanılarak, fotoğraf verisi ve kullanıcı ID'si doğru şekilde API'ye gönderilmekte, böylece backend tarafında bu fotoğraf kaydedilip kullanıcı profili güncellenmektedir. Bu çözüm ile kullanıcılar kendi profil fotoğraflarını kolayca yükleyebilir ve veritabanında güncellenmiş görsel bilgisiyle saklanabilir.

- Mesajlar, sadece userID ve messageText bilgileriyle geliyor. Kullanıcı adını (username) göstermek için her mesajdaki userID'yi, kullanıcı listesindeki ilgili kullanıcıyla eşleştirmek gerekiyor.

-Eğer users listesinden userID ile eşleşen bir kullanıcı bulunamazsa, mesajlar "Unknown" şeklinde bir kullanıcı adıyla görüntüleniyor.

#### 4. Deneyisel Sonuçlar

- Projemizde haritada etkinlik ve kullanıcıları gösterirken etkinlikleri kırmızı konum işareti ile gösterirken , kullanıcıyı ise mavi renkte daire sembolü ile gösterdik.

- Projemizde harita işlevselliğini sağlamak amacıyla **Google Maps API** kullanılmıştır. Bu kapsamda, API erişimi için gerekli olan bir anahtar alınmış ve harita entegrasyonu sağlanarak,

kullanıcıların ve etkinliklerin konumlarının görsel olarak projede gösterilmesi gerçekleştirilmiştir.

- Projemizde, farklı coğrafi bölgelerden ve farklı ülkelerden kullanıcıların aynı etkinliğe katılabilme olasılığı göz önünde bulundurulmuştur. Kullanıcıların farklı zaman dilimlerinde bulunması, etkinliklerin doğru saatlerde takip edilmesini zorlaştırabilirdi. Bu nedenle, sistemde tüm kullanıcılar için tek bir zaman dilimi kullanılması tercih edilmiştir.

Zaman dilimi farklılıklarını ortadan kaldırmak amacıyla, etkinliklerin tarih ve saat bilgileri UTC (Coordinated Universal Time) zaman dilimi kullanılarak kaydedilmiştir.

LocalDateTime.now() metodu ile tüm kullanıcılar, tek bir zaman diliminde düşünülerek işlem yapılmıştır. Bu yaklaşım, kullanıcılar arasında tutarlı bir zaman göstergesinin sağlanmasını ve zaman diliminden kaynaklanan hataların engellenmesini mümkün kılmıştır.

- Projemizde, frontend ve backend farklı portlarda çalışırken (örneğin, frontend 6060 portunda, backend ise 8080 portunda) tarayıcıda CORS hatası meydana geliyordu. Bu hata, frontend uygulamasının backend'e HTTP istekleri göndermeye çalışırken, tarayıcının güvenlik nedeniyle bu istekleri engellemesinden kaynaklanmaktadır. Projede, frontend ve backend farklı portlarda çalışırken (örneğin, frontend 6060 portunda, backend ise 8080 portunda) tarayıcıda CORS (Cross-Origin Resource Sharing) hatası meydana geliyordu. Bu hata, frontend uygulamasının backend'e HTTP istekleri göndermeye çalışırken, tarayıcının güvenlik nedeniyle bu istekleri engellemesinden kaynaklanmaktadır.

- Mesajlar görüntülendiğinde, yeni mesajların sayfanın altında kalmaması için bir otomatik kaydırma mekanizması uygulandı. Bir useRef kullanılarak son mesajın bulunduğu yere kaydırma yapıldı. Bu kod, messages state'i her güncellendiğinde çalışarak sayfanın son mesajı görecektir şekilde kaydırılmasını sağlıyor.

- Yeni bir mesaj gönderildikten sonra, messages state'i eski mesajların üzerine yazılırsa önceki mesajlar kaybolabilir. Yeni mesaj, mevcut mesajlar listesinin sonuna eklenerek güncelleniyor. setMessages([...messages, savedMessage]); Bu sayede, eski mesajlar korunarak yeni mesajlar ekleniyor.

#### Etkinlik Uygulaması Giriş Ekranı Sayfası

# EVENT APP

## LOGIN

Username or Email :

Password :

Login


Don't have an account?  
Register now.

Şekil 1.1.

#### Etkinlik Uygulaması Güncelleme Ekranı Sayfası

# EVENT APP

## User Info



Username :

Password :

Email :

Location :

Areas Of Interest :

Name :

Surname :

Date Of Birth :

Gender :

Number :


Update

Şekil 3.1.

#### Etkinlik Uygulaması Kayıt Ekranı Sayfası

# EVENT APP

## REGISTER



Username :

Password :

Email :

Location :

Areas Of Interest :

Name :

Surname :

Date Of Birth :

Gender :

Number :


Register

You have an account?  
Login now.

Şekil 2.1.

#### Etkinlik Uygulaması Anasayfası

# EVENT APP



ADD EVENT

UPDATE EVENT

DELETE EVENT

JOIN EVENT

### Your Events

Las Vegas GP

Abu Dhabi GP

Map Satellite



Şekil 4.1.

# EVENT APP

## UPDATE EVENT

Event Name :

Event Description :

Event Date :

Event Time :

Event Duration :

Event Location :

Event Category :

My Events

UPDATE

Şekil 6.2.

# EVENT APP

## DELETE EVENT

Event Name :

Event Description :

Event Date :

Event Time :

Event Duration :

Event Location :

Event Category :

My Events

DELETE

Şekil 6.3.

# Event App

## Admin Page

Select User:

Select a user

Select Event:

Select an event

Username:

Event Name:

Delete User

Delete Event

Şekil 5.1.

# EVENT APP

## ADD EVENT

Event Name :

Event Description :

Event Date :

Event Time :

Event Duration :

Event Location :

Event Category :

ADD

Şekil 6.1.

# EVENT APP

## JOIN EVENT

Event Name :

Event Description :

Event Date :

Event Time :

Event Duration :

Event Location :

Event Category :

Event List

Şekil 6.4.

# EVENT APP

## EVENT DETAIL

Event Name :

Event Description :

Event Date :

Event Time :

Event Duration :

Event Location :

Event Category :

Messages

Şekil 6.5.

### 5.Sonuç

-Bu projeyle birlikte sonuç olarak birçok kazanım elde ettik.Bunlar :

- Web programlama konusunda ilerleme kaydettik.

-Bu projeyle, Backend, Frontend ve veritabanı yönetim sistemini birlikte kullanarak bir uygulama geliştirilmiş, böylece aradaki bağlantının işleyişi daha net bir şekilde kavradık.

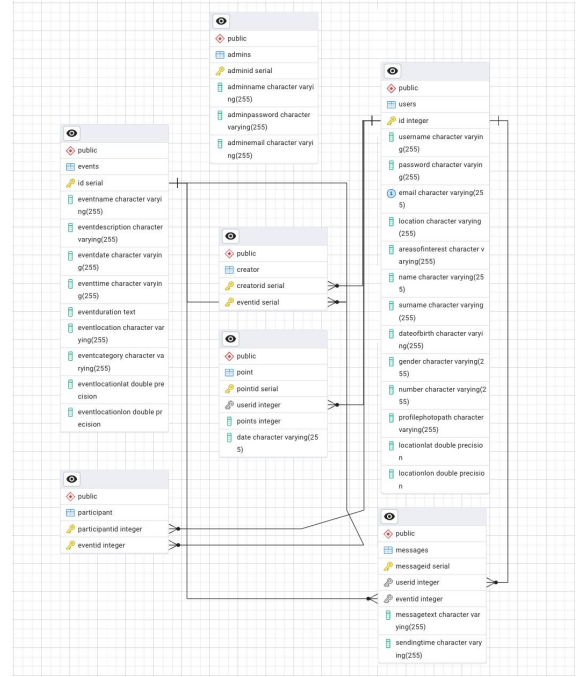
-Bu projeyle, etkinliğin ve kullanıcının aktif olarak haritada konumunu göstermeyi öğrendik.

-Mesaj Paneli ile nickname ile mesaj yazabilme ve önceden o etkinlik sohbetinde olan mesajları göstermeyi öğrendik.

### 6.Kaynakça

- [1] <https://stackoverflow.com>
- [2] Google Maps API
- [3] Google Görseller (Profil resimleri için)
- [4] <https://spring.io/projects/spring-boot>
- [5] <https://www.youtube.com/watch?v=eycjk3APuol>

### 7.Class Diyagramları



### 8.Yalancı Kod

MainPage.jsx

Başla

MainPage

Değişkenleri Tanımla:

```

navigate
role
userEvents
mapCenter
isLoading
    
```

Harita Ayarlarını Tanımla:

```

mapContainerStyle = {
    
```

```
genişlik: "100%",  
yükseklik: "400px",  
kenarYuvarlama: "20px",  
üsttenMargin: "1000px",  
}
```

Fonksiyon fetchEventDetails(eventId):

Deneme:

```
cevap = axios'dan GET isteği gönder  
(http://localhost:8080/events/ + eventId)  
geri döndür cevap.verisi  
Yakala:  
hata "Etkinlik detayları alınırken hata:", yazdır
```

Fonksiyon fetchUserEvents():

kullanıcıyı localStorage'dan al

Eğer kullanıcı yoksa:

```
hata "Kullanıcı bilgileri eksik" yazdır  
geri dön
```

kullanıcıId = kullanıcı.id

Deneme:

```
cevap = axios'dan GET isteği gönder  
(http://localhost:8080/participants/events/ + kullanıcıId)  
eventIds = cevap.verisi
```

```
eventsWithDetails = eventIds'den her eventId için  
fetchEventDetails'i çağır  
userEvents = null olmayan detayları ayarla
```

```
Eğer kullanıcı.locationlat ve kullanıcı.locationlon varsa:  
mapCenter'ı ayarla
```

Yakala:

```
hata "Kullanıcı etkinlikleri alınırken hata:", yazdır
```

useEffect ile:

```
role'ü localStorage'dan al ve ayarla  
fetchUserEvents'i çağır
```

Fonksiyon handleAdminPageClick():

Eğer role "admin" değilse:

```
uyarı "Admin sayfasına erişemezsiniz." yazdır  
geri dön  
navigate("/adminpage")
```

Fonksiyon handleAccountSettingsClick():

Eğer role "user" değilse:

```
uyarı "Kullanıcı bilgileri sayfasına erişemezsiniz." yazdır  
geri dön  
navigate("/userinfo")
```

Fonksiyon handleAddEventClick():

Eğer role "user" değilse:

```
uyarı "Etkinlik ekleme sayfasına erişemezsiniz." yazdır  
geri dön  
navigate("/addeventpage")
```

Fonksiyon handleUpdateEventClick():

Eğer role "user" değilse:

```
uyarı "Etkinlik güncelleme sayfasına erişemezsiniz." yazdır  
geri dön  
navigate("/updateeventpage")
```

Fonksiyon handleDeleteEventClick():

Eğer role "user" değilse:

```
uyarı "Etkinlik silme sayfasına erişemezsiniz." yazdır  
geri dön  
navigate("/deleteeventpage")
```

Fonksiyon handleJoinEventClick():

Eğer role "user" değilse:

```
uyarı "Etkinlik katılma sayfasına erişemezsiniz." yazdır  
geri dön  
navigate("/joineventpage")
```

Fonksiyon handleEventDetailClick(event):

```
eventToStore oluştur ve eventId'yi kaydet  
event nesnesini localStorage'a kaydet  
navigate("/eventdetailpage")
```

JSX Dön:

```
Ana kapsayıcı div oluştur  
Başlık "Event App" ekle  
Admin sayfasına gitmek için tıkladığında  
handleAdminPageClick'i çağır  
Kullanıcı ayarlarına gitmek için tıkladığında  
handleAccountSettingsClick'i çağır
```

Kullanıcı etkinliklerine dair başlık ekle

Eğer userEvents varsa:

Liste halinde etkinlikleri göster

Aksi halde:

"Henüz katıldığınız etkinlik yok." mesajı göster

Harita alanı oluştur

Eğer harita yüklendiyse:

Haritayı göster

Kullanıcının konumunu ve etkinliklerin konumlarını  
işaretle

Aksi halde:

"Harita yükleniyor..." mesajı göster

Son

```
Login.jsx

Başla Login

Değişkenleri Tanımla:
  kullanıcıAdı
  sifre
  girisBasarisiz
  navigate

Fonksiyon handleSubmit(e):
  e.preventDefault()

  Eğer kullanıcıAdı veya sifre boşsa:
    uyarı "Lütfen her iki alanı doldurun."
    geri dön

  loginVerisi = { kullanıcıAdı, sifre }
  adminLoginVerisi = { adminAdı: kullanıcıAdı, adminSifre:
sifre }

  localStorage'dan user, admin ve role'ü kaldır

  Gönderilen Login Verisini konsola yazdır

  isUserLoginFailed = false
  isAdminLoginFailed = false

  Deneme:
    Kullanıcı girişi kontrolü için POST isteği gönder
    (http://localhost:8080/user/login, loginVerisi)
    Eğer yanıt varsa:
      role'ü "user" olarak ayarla
      user'ı localStorage'a kaydet
      navigate("/mainpage")
      geri dön
    Aksi halde:
      isUserLoginFailed = true

  Yakala:
    hata "Kullanıcı girişi başarısız" yazdır

  Deneme:
    Admin girişi kontrolü için POST isteği gönder
    (http://localhost:8080/admin/login, adminLoginVerisi)
    Eğer yanıt varsa:
      role'ü "admin" olarak ayarla
      admin'i localStorage'a kaydet
      navigate("/mainpage")
    Aksi halde:
      uyarı "Geçersiz admin kullanıcı adı/e-posta veya şifre."
      isAdminLoginFailed = true
```

Yakala:  
hata "Admin girişi başarısız" yazdır

Eğer isUserLoginFailed ve isAdminLoginFailed ise:  
girisBasarisiz'i true yap  
uyarı "Giriş başarısız. Bilgilerinizi kontrol edin ve tekrar deneyin."

JSX Dön:  
login-container div oluştur  
"EVENT APP" başlığını ekle  
"LOGIN" başlığını ekle  
form oluştur ve handleSubmit ile gönderimi sağla  
username/email girişi için input alanı oluştur  
sifre girişi için input alanı oluştur  
Submit butonunu ekle

Kayıt mesajı ekle ve navigate("/register") ile yönlendir

Son