

# Akıllı Etkinlik Planlama Platformu

Alperen Karacan 210201096

—Yazılım laboratuvarı dersinin ikinci projesi olarak istenen akıllı etkinlik planlama uygulaması genel anlamda web tabanlı etkinlik platformlarının demosu niteliğinde bir proje olarak pratik uygulamaların kapsamlı özelliklerini ister olarak içermektedir. Platformun temel işlevinin bir kayıt tabanlı etkinlik düzenleme ve katılım ortamı sağlamasıdır. Bu uygulamadaki önemli kısımlardan ilki bireysel kayıt yönetimi ve platform yetkilisi için ortam hazırlığıdır. Kayıt tabanı oluşturulduktan sonra etkinlik oluşturma, paylaşma ve görüntüleme işlevleriyle kullanıcıların üst düzey bir etkileşimle etkinlikleri haber vermesi ya da etkinliklerden haberdar olmasını sağlamaktadır. Bunların yanında gelişmiş web altyapısının bize sunduğu dinamik metin ekleme ve konum servisleri entegrasyonları sayesinde etkinlikler hakkında daha yeterli ve düzgün ifade edilmiş bilgilerin bulunması yanı sıra oldukça yararlı olan bir destek mekanizması işlevi gören yorum yapabilme özelliğini sunmalıdır. Ayrıca çeşitli yönetici paneli özellikleriyle platformun denetim, kontrol ve yönetiminin de tam olarak sağlanması ve platformun erişilebilirliği garanti edilmiştir. Tüm bu özelliklerle bu proje bir etkinlik yönetim platformu projesinin prototipini tam olarak ifade etmektedir.

## I. ÖZET

Proje sürecimizde belirlediğimiz adımlar, çalışma düzenimizin daha sistematik ve etkili olmasını sağladı. Python ve JavaScript teknolojilerini temel alarak, proje gereksinimlerine uygun bir yapıyı adım adım inşa ettik. Çalışmaya başlamadan önce kullanıcı arayüzü ve veritabanı tasarımı üzerine yoğunlaştık. Kullanıcıların rahatça etkileşimde bulunabileceği bir arayüz oluşturmak, projenin temel taşlarından biri oldu. JavaScript ile arayüz tasarımını şekillendirirken, veri akışını sağlamak için Python tabanlı API entegrasyonlarını kullandık. Backend geliştirme sürecinde genellikle API'leri entegre ederek temel işlevleri uygulamaya geçirdik. Veritabanı yapılandırmasını tamamlarken, API çağrıları üzerinden veri alışverişi sağlayarak sistemin dinamik bir yapıya sahip olmasını sağladık. Backend tarafında API entegrasyonu büyük ölçüde işimizi kolaylaştırdı; birçok işlevi

sıfırdan yazmak yerine var olan API'leri entegre ederek zaman kazandık ve verimli bir yapı kurduk. Frontend ile backend arasındaki bu bağlantı sayesinde, kullanıcılar gerçek zamanlı verilere kolayca erişebildi. Özellikle dinamik arama ve filtreleme özelliklerinde, API'lerin sunduğu veri işlemlerini kullandık ve test süreçlerinde istenen performans elde ettik. Projenin sonunda, geliştirdiğimiz yapı kullanıcıların ihtiyaçlarını karşılayacak şekilde şekillendi ve test sonuçlarında belirlenen tüm gereksinimlerin yerine getirildiği doğrulandı. Bu proje, var olan API'leri verimli kullanmanın önemini anlamamıza ve Python ile JavaScript entegrasyonunda değerli bir deneyim kazanmamıza olanak sağladı.

## II. GİRİŞ

Projenin aşamalarından bahsedeceğim bu kısımda yolculuğun içinde bulunacak ve projenin gelişim ve değişimine şahit olarak tamamlanması konusunda tam olarak fikir sahibi olacaksınız. Öncelikle yöntemi (yani bu projenin tam olarak nasıl bir üretim aşaması geçirdiğini ), sonucu, deneysel sonuçları ve benzer projeler için gerekebilecek ve bize bu süreçte yardımcı olan kaynakları göreceksiniz. Arayüz ve veri tabanının şekillendirilmesiyle başlayacağımız bu yolculukta bu iki kısmın back end e oop ile nasıl entegre edilerek end to end bir masaüstü uygulaması yapıldığını göstereceğiz.

## III. YÖNTEM

Öncelikle veri tabanının tasarlanması üzerine yöneldik ve bunu aradan çıkararak işlevlerin daha düzenli bir şekilde oluşturulabilmesinin sağlanacağını göstermiş olduk. Tablolarımızı manuel (arayüz aracılığıyla) oluşturarak kolay bir şekilde bu kısmı yaptık .

Kullanıcılar Tablosu Kullanıcı bilgilerini saklar.

Sütunlar: ID, Kullanıcı Adı, Şifre, E-posta, Konum, İlgi Alanları, Ad, Soyad, Doğum Tarihi, Cinsiyet, Telefon Numarası, Profil Fotoğrafı. Etkinlikler Tablosu Etkinlik detaylarını içerir.

Sütunlar: ID, Etkinlik Adı, Açıklama, Tarih, Saat, Etkinlik Süresi, Konum, Kategori. Katılımcılar Tablosu Kullanıcılar ve etkinlikler arasında many-to-many ilişkisini temsil eder.

Sütunlar: Kullanıcı ID, Etkinlik ID. Mesajlar Tablosu Kullanıcılar arasındaki mesajlaşma detaylarını içerir.

Sütunlar: Mesaj ID, Gönderici ID, Alıcı ID, Mesaj Metni, Gönderim Zamanı. Puanlar Tablosu Kullanıcıların kazandığı puanları saklar.

Sütunlar: Kullanıcı ID, Puanlar, Kazanılan Tarih. Örnek olarak events tablosu Fig 1 de görülmektedir.

id	name	description	date	location	category	organizer_id	is_approved
(PK) integer	character varying (150)	text	timestamp without time zone	character varying (150)	character varying (100)	integer	boolean
1	sa	idi	2024-11-09 17:15:00	İstanbul	Eğlence	4	true
2	Veri Bilimi Konferansı	Veri bilimi ve yapay zeka alanındaki yenilikleri paylaşıldığı konferans.	2024-11-10 17:04:04.000184	İzmir	Konferans	10	true
3	deneme	deneme	2024-11-09 17:15:00	İstanbul	komedisi	6	true
4	16 Kaam	deneme	2024-11-09 17:15:00	sivas	Eğlence	6	true
5	11 Kaam2	deneme2	2024-11-09 17:15:00	sivas	deneme	6	true
6	17 Kaam	deneme	2024-11-12 20:40:00	Adıyaman	deneme	6	true
7	18 de	deneme	2024-11-27 02:25:00	Adana	id	4	true
8	20 tekin	deneme	2024-11-09 17:15:00	İstanbul	Teknoloji	4	true
9	21 teknoloji	deneme	2024-11-09 17:15:00	İstanbul	Teknoloji	4	true
10	22 de	deneme2	2024-11-21 12:00:00	Ankara	Müzik	6	true
11	23 sda	deneme	2024-11-21 12:00:00	İstanbul	Sanat	6	true
12	24 29 Kaam	deneme2	2024-11-29 12:00:00	İstanbul	Sanat	6	true
13	25 kaam	deneme	2024-11-30 12:00:00	Bolu	Kıyık	4	true
14	2 Flaklı ile Web Geliştirme	Flaklı frameworkü ile web uygulamaları geliştirme eğitimi.	2024-11-29 12:00:00	Ankara	Teknoloji	10	true
15	26 29 Kaam	deneme	2024-11-29 12:00:00	Kocaeli	Eğitim	6	true
16	14 Marisa deneme film	deneme2	2024-12-11 12:00:00	Manisa	Film	6	true
17	27 sivas deneme	deneme	2024-12-26 12:00:00	Sivas	Kıyık	6	true
18	23 1 Anak	Etkinlik Organizasyonu Denemesi	2024-12-01 12:00:00	Bingöl	Seyahat	6	true
19	1 Python Workshop	Temel ve İleri seviye Python konuları	2024-11-09 17:04:04.000184	İstanbul	Eğitim	10	true
20	10 Python	Python Çalışması	2024-11-09 17:15:00	İstanbul	Teknoloji	6	true

Fig. 1. Örnek olarak events tablosu

Ardından kullanıcı ekranı tasarımına geçtik. Kullanıcı ekranı, uygulamanın giriş noktası olarak kullanıcılara hesap oluşturma, giriş yapma ve profillerini yönetme imkanı sağlar. Bu ekran, kullanıcıların hesaplarına erişebilmesi ve güncel bilgilerle profilini düzenleyebilmesi için tasarlanmıştır. Kullanıcılar, kayıt formu aracılığıyla temel bilgilerini sisteme ekler ve şifre belirleyerek hesap güvenliğini sağlar. Ayrıca, giriş ekranı kullanıcıların e-posta ve şifre bilgileri ile doğrulama yapmasını ve uygulamaya erişmesini mümkün kılar. Şifre sıfırlama özelliği ise kullanıcının hesap güvenliğini artırır ve gerektiğinde erişim sorunlarını çözer. Bu işlevi gerçekleştirmesini sağlayan kod Fig 2 de gösterilmiştir.

```
@bp.route('/register', methods=['GET', 'POST'])
def register():
    form = RegistrationForm()
    if form.validate_on_submit():
        user = User(
            username=form.username.data,
            email=form.email.data,
            first_name=form.first_name.data,
            last_name=form.last_name.data
        )
        user.set_password(form.password.data)
        db.session.add(user)
        db.session.commit()
        flash('Kayıt başarılı! Lütfen e-postanızı doğrulayın.', 'success')
        return redirect(url_for('auth.login'))
    return render_template('register.html', form=form)

@bp.route('/login', methods=['GET', 'POST'])
def login():
    form = LoginForm()
    if form.validate_on_submit():
        user = User.query.filter_by(email=form.email.data).first()
        if user and user.check_password(form.password.data):
            login_user(user, remember=form.remember.data)
            return redirect(url_for('main.index'))
        else:
            flash('Giriş başarısız. Lütfen bilgilerinizi kontrol edin.', 'danger')
    return render_template('login.html', form=form)
```

Fig. 2. Kullanıcı ekranı işlevinin kodu

Kullanıcı ekranından sonra da admin ekranının yapımına başladık. Admin ekranı, sistem yöneticilerinin kullanıcıları ve etkinlikleri yönetebildiği özel bir kontrol paneli sunar. Bu ekran sayesinde, yöneticiler kullanıcıların rollerini değiştirebilir, kullanıcıları silebilir ve sistemdeki etkinlikleri onaylayarak yayınlanmalarını sağlayabilir. Ayrıca, kullanıcıların toplam puanlarını görüntüleyebilir ve sistem performansını değerlendirebilir. Bu özellikler, sistemin düzgün işleyişi ve kullanıcı yönetimi açısından oldukça önemlidir. Bu işlevi gerçekleştirmesini sağlayan kod Fig 3 de ifade edilmiştir.

```
@bp.route('/users')
@login_required
@admin_required
def user_list():
    users = db.session.query(
        User.id, User.username, User.email, User.role, func.coalesce(func.sum(Point.score), 0).label('total_score')
    ).outerjoin(Point, User.id == Point.user_id).group_by(User.id).all()
    return render_template('admin/user_list.html', users=users)

@bp.route('/approve_events')
@login_required
@admin_required
def approve_events():
    pending_events = Event.query.filter_by(is_approved=False).all()
    return render_template('admin/approve_events.html', events=pending_events)
```

Fig. 3. Admin ekranı işlevinin kodu

Admin ekranı ardından mail doğrulamasına geçmiştik ki bu aşama genel güvenliğin temelini oluşturduğundan dolayı bu kadar erken gerçekleştirilmektedir. Mail doğrulama, kullanıcıların hesaplarını etkinleştirmesi ve sisteme güvenli bir şekilde erişmesi için kritik bir özelliktir. Kayıt işlemi sırasında, kullanıcının e-posta adresine doğrulama bağlantısı gönderilir. Kullanıcı bu bağlantıya tıklayarak hesabını doğrular ve hesabın sahte olmadığını kanıtlar. Bu mekanizma, hem kullanıcı güvenliğini artırır hem de spam hesapları engeller. Bu işlevse Fig 4 te görülebilmektedir.

```
def generate_confirmation_token(email):
    serializer = URLSafeTimedSerializer(current_app.config['SECRET_KEY'])
    return serializer.dumps(email, salt='email-confirmation-salt')

@bp.route('/confirm/<token>')
def confirm_email(token):
    email = confirm_token(token)
    if not email:
        flash('Doğrulama bağlantısı geçersiz.', 'danger')
    else:
        user = User.query.filter_by(email=email).first_or_404()
        user.is_confirmed = True
        db.session.commit()
        flash('Hesabınız başarıyla doğrulandı.', 'success')
        return redirect(url_for('auth.login'))
```

Fig. 4. Mail doğrulamasının gerçekleştirilmesini sağlayan kod

Mail doğrulaması ardından platformun asıl amacı olan etkinlik yönetimine geçilmiştir. Etkinlik oluşturma ve katılım özellikleri, kullanıcıların sisteme katkıda bulunmasını ve diğer kullanıcılarla etkileşim kurmasını sağlar. Kullanıcılar, etkinlikler oluşturabilir ve bunlara katılabilir. Oluşturulan etkinlikler, admin tarafından onaylandıktan sonra sistemde yayınlanır. Katılım durumunda, kullanıcıya puan verilerek aktif katılım teşvik edilir. Bu aşama görece ara bir aşama olduğundan yalancı koduna yer verilmiştir ve yalancı kod Fig 5 te gösterilmiştir.

```
1. Kullanıcı giriş yapmış mı? (login_required)
- Hayır: Kullanıcı giriş sayfasına yönlendirilir.
- Evet: Devam edilir.

2. Etkinlik oluşturma işlemi başlatılır.
- Etkinlik formu kullanıcıya gösterilir.
- Kullanıcı formu doldurup gönderdi mi? (validate_on_submit)
- Hayır: Form yeniden gösterilir.
- Evet:
  - Form verilerinden yeni bir etkinlik nesnesi oluşturulur:
    - Etkinlik Adı
    - Açıklama
    - Tarih
    - Konum
    - Kategori
    - Organizatör: Giriş yapan kullanıcı
  - Onay Durumu: Veriyolun olarak "False" (Admin onayı gereklidir)
  - Etkinlik veritabanına kaydedilir (db.session.add ve db.session.commit).
  - Kullanıcıya "Etkinlik oluşturuldu, yönetici onayından sonra yayınlanacak." mesajı gösterilir.
  - Etkinlik oluşturma sayfasına geri yönlendirilir.

3. Kullanıcı, oluşturulan etkinliğe katılmak ister mi? (login_event)
- Kullanıcı etkinlik ID'sine göre etkinliği seçer.
- Etkinlik veritabanında sorgulanır:
  - Etkinlik bulunamazsa, 404 hatası döndürülür.
- Kullanıcı daha önce bu etkinliğe katılmış mı? (EventAttendee sorgusu)
- Evet: Kullanıcıya "Zaten bu etkinliğe katıldınız." mesajı gösterilir.
- Hayır:
  - Katılım bilgisi (EventAttendee) oluşturulur:
    - Kullanıcı ID
    - Etkinlik ID
  - Katılım bilgisi veritabanına kaydedilir (db.session.add ve db.session.commit).
  - Kullanıcıya "Etkinliğe katıldınız!" mesajı gösterilir.
- Kullanıcı etkinlik detayları sayfasına yönlendirilir.

4. İşlem tamamlanır.
```

Fig. 5. Etkinlik yönetiminin yalancı kodu

Ardından sıra etkinlik yorumlarıyla kullanıcıların dinamik iletişimine geldi. Etkinlik yorumları, kullanıcıların etkinlikler hakkında görüşlerini paylaşmasını ve diğer katılımcılarla etkileşimde bulunmasını sağlar. Kullanıcılar, etkinlik detay sayfasında yorum yapabilir ve önceki yorumları görüntüleyebilir. Bu işlevi sağlayan kod Fig 6 da görülebilir.

```
@bp.route('/event/<int:event_id>', methods=['GET', 'POST'])
@login_required
def event_messages(event_id):
    event = Event.query.get_or_404(event_id)
    form = MessageForm()
    if form.validate_on_submit():
        message = Message(sender=current_user, event=event, content=form.content.data)
        db.session.add(message)
        db.session.commit()
        flash('Mesaj gönderildi!', 'success')
    messages = Message.query.filter_by(event_id=event.id).all()
    return render_template('event_messages.html', event=event, messages=messages, form=form)
```

Fig. 6. Yorum yapabilme işlevinin kazandırıldığı kod

Devamında kullanıcının kişisel bir deneyim alabilmesi için eklenen takvim bölümü bulunmaktadır. Kullanıcıların katıldığı etkinlikler bir takvim üzerinde listelenir. Bu özellik, kullanıcıların geçmiş ve gelecekteki etkinliklerini kolayca takip etmesine olanak tanır. Bu işlevin kodu ise Fig 7 de belirtilmiştir.

```
@bp.route('/calendar', methods=['GET'])
@login_required
def calendar():
    """Kullanıcının katıldığı etkinlikleri listeleyen takvim sayfası"""
    events = Event.query.join(EventAttendee).filter_by(user_id=current_user.id).all()
    return render_template('calendar.html', events=events)
```

Fig. 7. Takvim işlevini içeren kod

Son olarak Google Maps api si aracılığıyla sağlanan ve etkinlikler hakkında en önemli bilgilerden biri olan konum bilgisinin sağlanması bölümündeyiz. Etkinlik konumu özelliği, kullanıcıların harita üzerinde etkinliklerin yerini görmesini sağlar. Bu sayede kullanıcılar, etkinliğe ulaşım planlamalarını kolayca yapabilir. Bu işlevin kazandırılmasını sağlayan kod ise Fig 8 de görülebilir.





**Akıllı Etkinlik Planlama** Yönetici Paneli Profil Etkinlik Oluştur Takvim Önerilen Etkinlikler Etkinlik Haritası Çıkış Yap

### Kullanıcı Listesi

[Etkinlik Oluştur](#)

ID	Kullanıcı Adı	E-posta	Ad	Soyad	Rol	Durum	Puan	İşlemler
6	AlperenDeneme	alperenkaracpc@gmail.com	Alperen	Karacan	<span>Yeni</span>	<span>Yeni</span>	237	<a href="#">Sil</a>
4	denemedeir	alperenkaracan5@gmail.com	Alperen	Kara	<span>Yeni</span>	<span>Yeni</span>	86	<a href="#">Admin</a>
5	AlperenK	alperenkaracan34@gmail.com	Alperen	Kara	<span>Yeni</span>	<span>Yeni</span>	10	<a href="#">Sil</a>
2	Alperen	deneme123@gmail.com	Alperen	K	<span>Yeni</span>	<span>Yeni</span>	10	<a href="#">Sil</a>
8	Mehmet Ali Kır	kirmali.197@gmail.com	Mehmet Ali	Kır	<span>Yeni</span>	<span>Yeni</span>	10	<a href="#">Sil</a>
1	admin	admin@example.com	Adminim	User	<span>Yeni</span>	<span>Yeni</span>	7	<a href="#">Sil</a>

Fig. 13. Adminin kullanıcıları görüntülediği sayfanın grafik tabanlı çıktısı

**Akıllı Etkinlik Planlama** Yönetici Paneli Profil Etkinlik Oluştur Takvim Önerilen Etkinlikler Etkinlik Haritası Çıkış Yap


### Önerilen Etkinlikler

<b>tekin</b> Kategori: Teknoloji Tarih: 2024-11-09 17:15 Konum: İstanbul İsim: <a href="#">Detaylar</a>	<b>teknoloji</b> Kategori: Teknoloji Tarih: 2024-11-29 13:30 Konum: İstanbul İsim: <a href="#">Detaylar</a>	<b>sda</b> Kategori: Sanat Tarih: 2024-11-21 12:00 Konum: İstanbul İsim: <a href="#">Detaylar</a>	<b>29 Kasım</b> Kategori: Sanat Tarih: 2024-11-29 12:00 Konum: İstanbul İsim: <a href="#">Detaylar</a>
<b>kitapım</b> Kategori: Kitap Tarih: 2024-11-20 12:00 Konum: Etiler İsim: <a href="#">Detaylar</a>	<b>Flask ile Web Geliştirme</b> Kategori: Teknoloji Tarih: 2024-11-29 13:00 Konum: İstanbul İsim: Flask frameworkü ile web uygulamaları geliştirme <a href="#">Detaylar</a>	<b>sivas deneme</b> Kategori: Sanat Tarih: 2024-12-26 12:00 Konum: Sivas İsim: <a href="#">Detaylar</a>	

Fig. 17. Önerilen etkinliklerin sayfasının grafik tabanlı çıktısı

**Akıllı Etkinlik Planlama** Yönetici Paneli Profil Etkinlik Oluştur Takvim Önerilen Etkinlikler Etkinlik Haritası Çıkış Yap

### Profil



Kullanıcı Adı: denemedeir

E-posta: deneme@gmail.com

Ad: Alperen

Soyad: Karacan

Rol: Admin

Profil Resmi Yükle: [Dinle](#) [Sil](#)

[Görüntüle](#)

Fig. 14. Kullanıcı profili sayfasının puanlar ve etkinlikler harici grafik tabanlı çıktısı

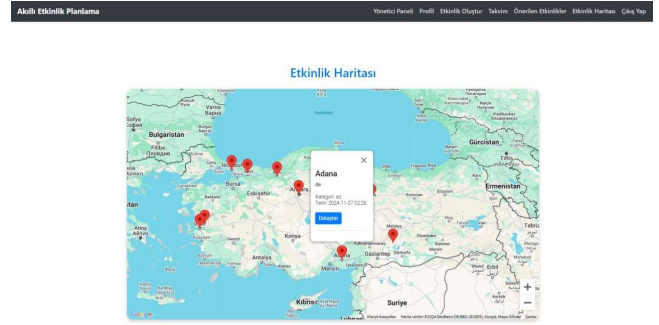


Fig. 18. Etkinliklerin konum işlevi kısmının grafik tabanlı çıktısı

**Akıllı Etkinlik Planlama** Yönetici Paneli Profil Etkinlik Oluştur Takvim Önerilen Etkinlikler Etkinlik Haritası Çıkış Yap

### Etkinlik Oluştur

Etkinlik Adı:

Açıklama:

Tarih ve Saat:

Konum:

Kategori:

[Kaydet](#)

Fig. 15. Etkinlik oluşturma sayfasının grafik tabanlı çıktısı

**Python**

Açıklama: Python Çalışması

Tarih: 2024-11-09 17:15

Konum: İstanbul

Kategori: Teknoloji

Düzenleyen: AlperenDeneme

[Etkinlikten Ayrılmak](#) [Etkinliği Düzenle](#) [Etkinliği Sil](#) [Mesajlar](#)

**Katılımcılar**

denemedeir

Fig. 19. Etkinliğin düzelenebilmesini sağlayan sayfanın grafik tabanlı çıktısı

**Akıllı Etkinlik Planlama** Yönetici Paneli Profil Etkinlik Oluştur Takvim Önerilen Etkinlikler Etkinlik Haritası Çıkış Yap

### Etkinlik Takvimi

[Görüntüle](#) [Yeni](#) [Sil](#)

**Aralık 2024**

Paz	Paz	Sal	Çar	Per	Cum	Cen
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Fig. 16. Takvim sayfasının grafik tabanlı çıktısı

**Python Workshop - Mesajlar**

Açıklama: Temel ve İleri seviye Python konuları

Tarih: 2024-11-09 17:04

Konum: İstanbul

**Mesajlar**

admin: deneme 2024-11-10 21:07

denemedeir: merhaba 2024-11-10 23:16

AlperenDeneme: merhaba 2024-11-12 14:53

**Mesaj Gönder**

Mesaj:

[Gönder](#)

Fig. 20. Aktif etkinliklerdeki mesajla iletişim işlevinin grafik tabanlı çıktısı

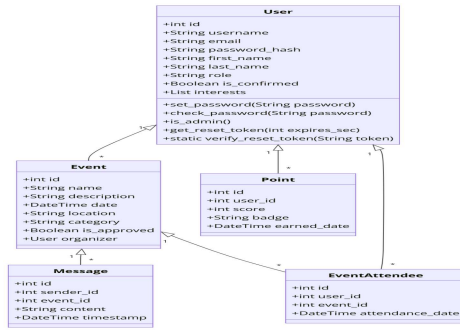


Fig. 21. Projenin UML diyagramı

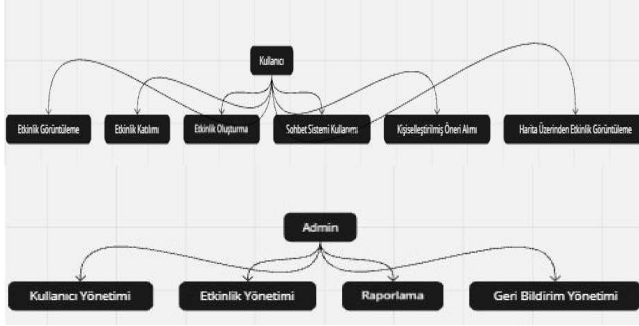


Fig. 22. Projenin use case diyagramı

## VI. KAYNAKÇA

stackoverflow  
chatGPT  
Youtube  
Google Maps API Keys