

**KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



PROJENİN KONUSU:

Kullanıcılarına, etkinliklere katılamayacakları durumlarda biletlerini satma veya bu tür biletleri satın alma imkanı sunan platform

Yazılım Mühendisliği Mimari Tasarım Raporu

**Sinan KOCATÜRK- 410452
Hüseyin Uğur Aydın- 412610
Enes Ceviz- 402498**

2022-2023 GÜZ DÖNEMİ



IEEE Etik Kuralları IEEE Code of Ethics



Mesleğime karşı şahsi sorumluluğumu kabul ederek, hizmet ettiğim toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmaya söz verdiğimi ve aşağıdaki etik kurallarını kabul ettiğimi ifade ederim:

1. Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;
2. Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;
3. Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;
4. Her türlü rüşveti reddetmek;
5. Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;
6. Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;
7. Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriye kabul etmek ve eleştiriye yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;
8. Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği, veya cinsiyet ifadesi üzerinden ayrımcılık yapma durumuna girişmemek;
9. Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;
10. Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.

IEEE Yönetim Kurulu tarafından Ağustos 1990'da onaylanmıştır.

ÖNSÖZ

Bu platform, kullanıcıların etkinliklere katılamadıkları zamanlarda biletlerini satma veya alım yapma imkanı sunarak erişilebilirliği artırır. Kullanıcılar kullanmadıkları biletleri satarak ek gelir elde edebilir ve son dakika kararlarını kolayca uygulayabilirler.

Kültürel ve sosyal etkileşimi artırırken, daha fazla insanın etkinliklere katılma olasılığını artırır. Kullanıcılar, bilet satın alma ve satma süreçlerini daha kolay ve kullanıcı dostu bir şekilde yönetebilirler. Bu proje aynı zamanda veri toplama ve analiz fırsatları sunarak gelecekte daha iyi etkinlik planlamaları yapmanıza yardımcı olabilir.

Sinan KOCATÜRK- 410452
Hüseyin Uğur Aydın- 412610
Enes Ceviz- 402498
Trabzon 2023

İÇİNDEKİLER

	Sayfa No
IEEE ETİK KURALLARI.....	II
ÖNSÖZ.....	III
İÇİNDEKİLER.....	IV
ÖZET.....	V
1. GENEL BİLGİLER.....	1
1.1. Giriş.....	1
1.2. Problem tanımı ve projenin başlatılma gerekçesi.....	1,2
2 Mimari Tasarım Raporu.....	3
2.1 UML ve FLOW diyagramları	4
2.2 Dosya Yapısı.....	5
2.3 UI.....	6
2.4 Fonksiyonel gereksinimlere Mimari çözümler.....	13
2.4.1 Frontend Gereksinimlere Mimari Çözümler.....	14
2.4.2 Backend Gereksinimler Mimari Çözümler.....	15

ÖZET

Yapacağımız projede Front-End ve Back-End kısımlarını Python dili Django Framework'u ve Bootstrap ile geliştirilecektir. Database olarak da SQLite kullanılacaktır.

Proje Yöneticisi Enes Ceviz ve Hüseyin Uğur Aydın Front-End kısmında, Back-End kısmında Sinan Kocatürk görevli olacaktır.

1. GENEL BİLGİLER

1.1.Giriş

Etkinlikler, kültürel ve sosyal bağları güçlendirmenin kilit bir yoludur, fakat herkesin bu etkinliklere katılma şansı her zaman mümkün olmayabilir. İnsanlar çeşitli nedenlerle etkinliklere katılamayabilir ve bu durumda biletlerini boşa harcamış olurlar. Hem katılımcılar hem de etkinlik organizatörleri için bu, bir kayıp anlamına gelir.

Bu sorunu çözmek için geliştirilen bir platform, katılamayan kişilere biletlerini satabilme veya başkalarından bilet alabilme imkanı sunuyor. Bu sayede, biletleri kullanılmayan kişilerin zararını minimize ederken, diğer kullanıcıların da uygun fiyatlı biletlerle etkinliklere katılma fırsatını artırıyor. Platform, erişilebilirliği ve kullanılabilirliği artırarak, daha fazla kişinin kültürel etkinliklere katılmasını sağlar.

Kullanıcı dostu bir arayüz ile donatılmış olan bu platform, bilet alım-satım işlemlerini basitleştirerek geniş bir kullanıcı kitlesine hitap ediyor. Ayrıca, topladığı verilerle gelecekteki etkinliklerin daha iyi planlanmasına yardımcı olma potansiyeline sahip. Bu, etkinliklerin daha geniş bir kitleye ulaşmasını ve başarılarını artırmasını sağlayabilir.

Sonuç olarak, bu platform, etkinliklere daha kolay erişim sağlayarak, insanların kültürel ve sosyal etkileşimlerini artırmada önemli bir rol oynayabilir.

1.2.Problem tanımı ve projenin başlatılma gerekçesi

Problem tanımı: Etkinliklere katılmak isteyen bireylerin karşılaştığı temel zorluklardan biri, çeşitli nedenlerle katılamayacakları durumlarda biletlerinin boşa gitmesidir. Bu, hem bilet sahipleri için maddi kayıp hem de etkinlik organizatörleri için katılımcı eksikliği ve gelir kaybı anlamına gelmektedir. Ayrıca, bilet fiyatlarının yüksek olması veya bilet bulunabilirliğinin sınırlı olması gibi faktörler, potansiyel katılımcıların etkinliklere erişimini kısıtlamaktadır. Bu durum, kültürel ve sosyal etkileşim fırsatlarının azalmasına yol açmakta ve geniş bir kullanıcı kitlesinin etkinliklere katılımını engellemektedir.

Projenin başlatılma gerekçesi:

Bilet Kullanımında Etkinlik:

Etkinlik biletlerinin kullanılmaması, sıkça karşılaşılan bir sorundur. Bilet sahipleri, beklenmedik durumlar nedeniyle etkinliklere katılamayabilirler ve bu durumda biletleri boşa gider. Projenin temel gerekçesi, bu kullanılmayan biletlerin değerinin korunmasını ve yeniden kullanılabilir hale getirilmesini sağlamaktır.

Mali Kayıpların Azaltılması:

Kullanılmayan biletler, hem bilet sahipleri hem de etkinlik organizatörleri için mali kayıplara yol açar. Proje, bu mali kayıpları azaltmayı hedeflerken, aynı zamanda biletlerini satmak isteyen kişilere ek gelir elde etme fırsatı sunar.

Erişilebilirlik ve Katılım Fırsatlarının Artırılması:

Biletlerin yüksek maliyeti veya sınırlı bulunabilirliği, potansiyel katılımcıların etkinliklere erişimini kısıtlayabilir. Proje, daha uygun fiyatlı biletlerin sağlanması ve geniş bir kitleye ulaşılması yoluyla, kültürel ve sosyal etkinliklere katılım fırsatlarını genişletmeyi amaçlar.

Kullanıcı Dostu Çözüm:

Etkinlik biletlerinin alımı ve satımı süreçlerinin karmaşık olması, kullanıcıları caydırabilir. Bu projenin bir diğer temel gerekçesi, bu süreçleri basitleştiren, kullanıcı dostu bir platform sağlamak ve böylece geniş bir kullanıcı kitlesine ulaşmaktır.

Veri Toplama ve Analiz:

Etkinlik organizatörleri için bilet satış verilerinin analizi, gelecekteki etkinliklerin daha iyi planlanmasına yardımcı olabilir. Proje, veri toplama ve analiz araçları sunarak, etkinliklerin etkinliğini ve kapsamını artırmayı hedefler.

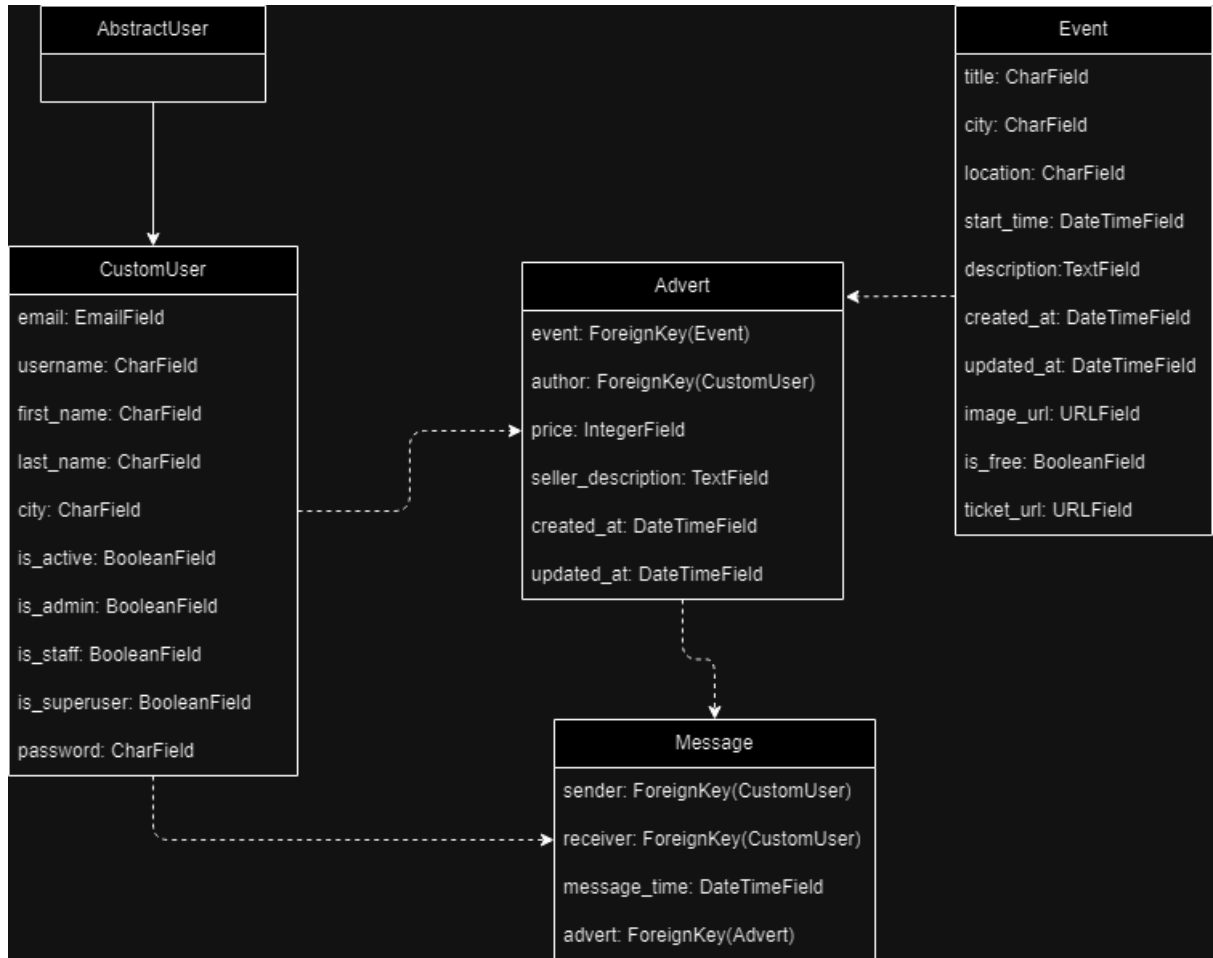
Kültürel ve Sosyal Etkileşimin Teşviki:

Son olarak, bu proje, insanların kültürel ve sosyal etkileşimlerini artırmayı ve toplum içindeki bağları güçlendirmeyi hedefler. Böylece, daha fazla insanın çeşitli etkinliklere katılımını sağlayarak toplumsal uyum ve zenginlik katkıda bulunmayı amaçlar.

2. Mimari Tasarım Raporu

2.1. Model sınıfların UML diyagramları ve Proje Kullanımının Akış Şeması

2.1.1. Model class'lar UML diyagramı



Şekil 2.1.1 Tabloda yer alan düz çizgiler inheritance, noktalı çizgiler composition ilişkisi göstermektedir.

En önemli class olan CustomUser sınıfı django.models yapısındaki user sınıfını miras alıyor. Bunun üzerine kendi ihtiyacımız olan data memberları dahil edebiliyoruz.

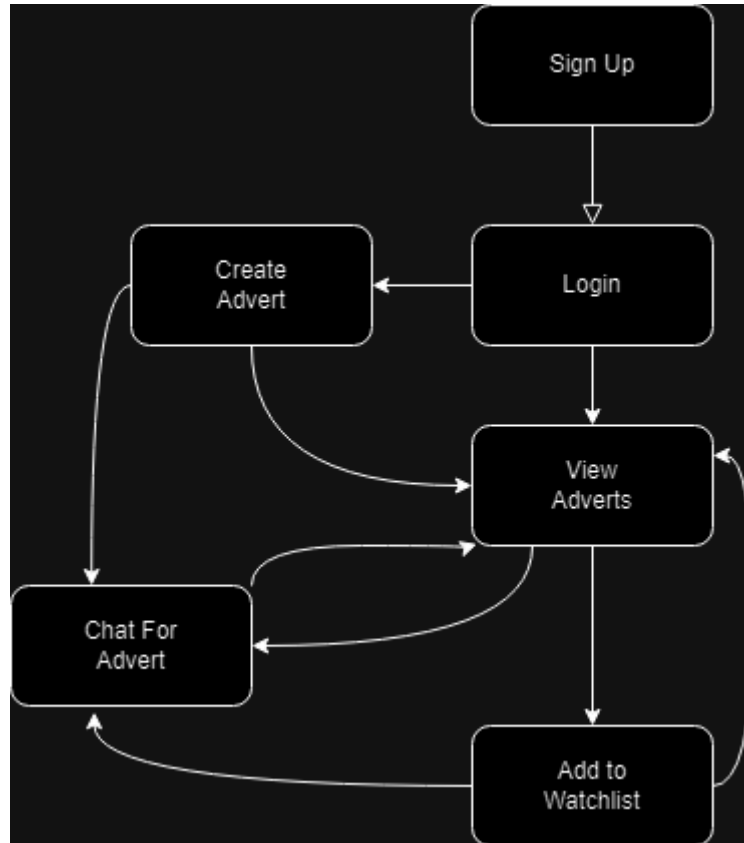
CustomUser sınıfı özet olarak standart user sınıfında olmayan first_name, last_name, city, username gibi data memberlar içeriyor. Burada email ve username benzersiz olmak zorunda.

Advert sınıfı, kullanıcıların oluşturduğu ilanların bilgilerini içeriyor.

Event sınıfı, API ile çekilen bileti satılabilecek olan gelecek etkinliklerin bilgilerini içeriyor.

Message sınıfı, kullanıcıların birbirine gönderdikleri mesajların bilgilerini içeriyor.

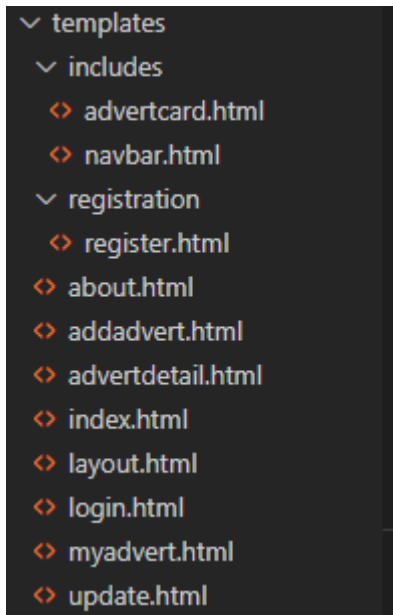
2.1.2 Proje Kullanımının Akış Şeması



Kullanıcı siteye kayıt olduktan sonra ilan oluşturabilmekte, oluşturulan ilanları görüntüleyebilmekte, ilanları izleme listesine ekleyebilmekte ve satıcı/alıcılarla mesajlaşabilmektedir.

2.2. Dosya yapısı

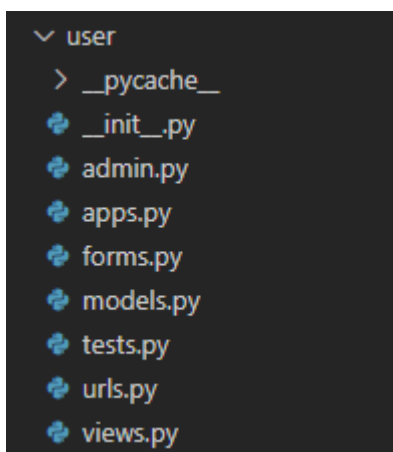
2.2.1 Frontend Dosya yapısı



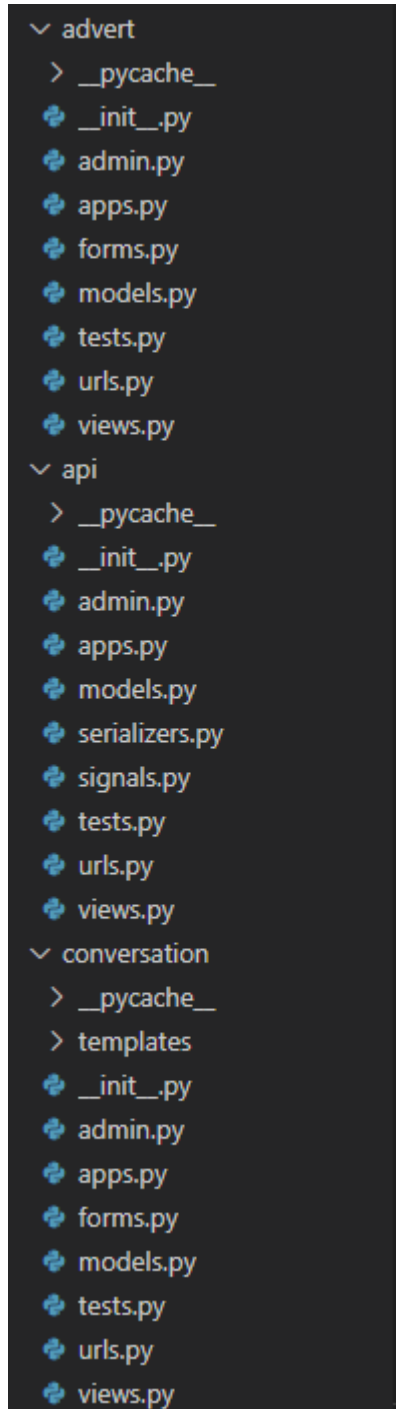
Şekil 2.2.1 html dosyalarının bulunduğu templates klasörü



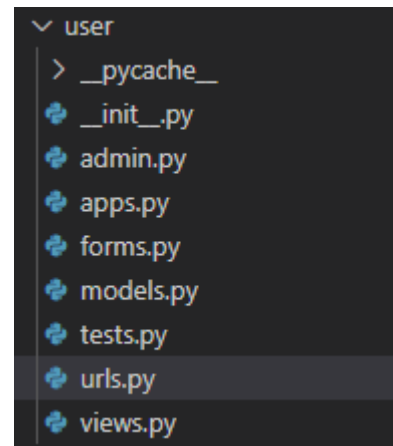
Şekil 2.2.2 static kodların dosya yapısı



Şekil 2.2.3 user sınıfın dosya yapısı



Şekil 2.2.5 Backend genel dosya yapısı



Şekil 2.2.6 user dosyası

2.3 UI

Web uygulamanın Geliştirme aşamasından görüntüler

Giriş Yap

Kullanıcı adı *

11111

Parola *

.....

Şifremi unuttum

Hesabın yok mu? **Kaydol**

Giriş Yap

Şekil 2.3.1 login screen

Görüntüler çalışan uygulamadan ekran alıntısıdır.

Giriş Yap

Kullanıcı adı *

Parola *



Lütfen bu alanı doldurun.

Şifremi unuttum

Hesabın yok mu? **Kaydol**


Giriş Yap

Şekil 2.3.2 yazmış olduğumuz validator'lerin kontrollü ile textbox boşken uygulamaya giremiyor


Görüntüler çalışan uygulamadan ekran alıntısıdır.

Başarıyla Giriş Yaptınız


Hoş geldin, 111




İstanbul Jan. 12, 2024, 7:30 p.m.
Ayçin Asan - Evet Ne Söyluyoduk
★★★★☆ (13) 213
232 TL



Bursa Jan. 27, 2024, 6:30 p.m.
Mirkelam
★★★★☆ (13) sinan11
122 TL



İstanbul Dec. 31, 2023, 8:59 p.m.
Hediye Paketi - Biletix
★★★★☆ (13) 213
123 TL



İstanbul Jan. 12, 2024, 7:30 p.m.
Ayçin Asan - Evet Ne Söyluyoduk
★★★★☆ (13) 123
423 TL

Şekil 2.3.3 Anasayfa

Görüntüler çalışan uygulamadan ekran alıntısıdır.

İlan Ekleme Ekranı

Ekleme istediğiniz etkinliği seçiniz, 111

Event*

Seller description*

Price*

İlan Oluştur

Şekil 2.3.4. İlan ekleme sayfası

Görüntüler çalışan uygulamadan ekran alıntısıdır. Veriler backend'den gelmektedir.

Kontrol Paneli

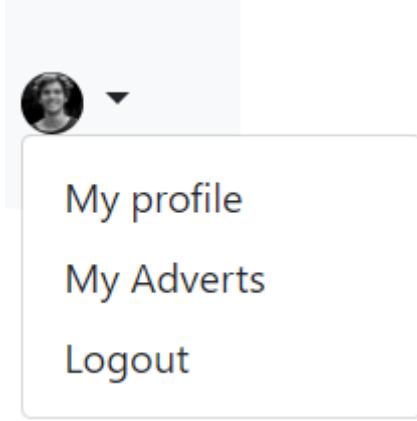
Hoşgeldiniz, 111

Etkinlik	Fiyat	Tarih	Yer	Açıklama	Güncelle	Sil
Depeche Mode Tribute- Moodie Petch	12	Dec. 20, 2023, 6:30 p.m.	İstanbul	acil satılık	Güncelle	Sil

İlan Ekle

Şekil 2.3.5 ilanlarım sayfası

Görüntüler çalışan uygulamadan ekran alıntısıdır. Veriler backend'den gelmektedir



Şekil 2.3.6 dropdown menü

Kayıt Ol

Kullanıcı adı*

E-posta*

Parola*

Parolayı Doğrula*

Kayıt Ol

Şekil 2.3.13 Register sayfası

Görüntüler çalışan uygulamadan ekran alıntısıdır

2.4 Fonksiyonel Gereksinimlere Mimari çözümler

Projemizde geliştirme aşamasında uygulamanın kullanım sırasında filtreleme gibi fonksiyonlarda tutarsızlık olabileceği için Türkiye'deki tüm etkinlikleri sitemizde bulundurabileceğimiz bir app oluşturduk. Bu sayede kullanıcı ilan oluşturma ekranında elindeki bileti seçebilir.

```
def fetch_and_save_events(request):  
  
    api_url =   
    api_key =   
  
    params = {  
        'take': 9999,  
        'format_ids' : '19',  
    }  
    headers = {  
        'X-Etkinlik-Token': api_key  
    }  
  
    response = requests.get(api_url, params=params, headers=headers)  
  
    if response.status_code == 200:  
        events_data = response.json().get('items', [])  
  
        for event_data in events_data:  
            event, created = Event.objects.get_or_create(  
                id=event_data.get('id'),  
                defaults={  
                    'title': event_data.get('name'),  
                    'location': event_data.get('venue', {}).get('name', ''),  
                    'city': event_data.get('venue', {}).get('city', {}).get('name', ''),  
                    'start_time': event_data.get('start'),  
                    'description': event_data.get('content'),  
                    'image_url': event_data.get('poster_url'),  
                    'is_free': event_data.get('is_free', False),  
                    'ticket_url': event_data.get('ticket_url', ''),  
                }  
            )  
        )
```

Yukarıdaki örnekte ise yapının api hizmeti veren platformdan verileri çeken kodu görüyoruz. Burada veritabanına sadece yazı giriyor. Görsellerin sadece url adresleri bize verildiği için harici depolama sorunu olmuyor.

2.4.1 Frontend Mimari çözümleri

Django'nun sağladığı yapılar kullanılarak kod tekrarı en aza indirildi ve en optimal şekilde veriler frontend tarafında gösterildi.

```
{% load static %}

<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0/dist/css/bootstrap.min.css" integrity="sha384-G6584921zKFCJ58gg4lwaF40Vt1650zqf1638F4d3456789012345678901234567890" crossorigin="anonymous">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@4.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-G6584921zKFCJ58gg4lwaF40Vt1650zqf1638F4d3456789012345678901234567890" crossorigin="anonymous">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/all.min.css" integrity="sha384-1z2o2t46EuMiybFPOSAoiWNI/hYO6RQwMz+yLq854S9lGnZFkwcm9q5t2r5v0W4s3r9" crossorigin="anonymous" referrerpolicy="no-referrer" />

  <!-- block head -->
  <!-- endblock head -->

  <title>İtinalıkPass</title>
</head>
<body class="container">
  <!-- include "includes/navbar.html" -->

  <!-- if messages -->
  <!-- for message in messages -->
  <!-- if message.tags == "info" -->
  <div class="alert alert-info" role="alert">{{ message }}</div>
  <!-- else -->
  <div class="alert alert-danger" role="alert">{{ message }}</div>
  <!-- endif -->
  <!-- endfor -->

  <!-- if not all_adverts -->
  <div class="col-md-12">
    <div class="alert alert-danger" role="alert">
      <h4 class="alert-heading">Uyar</h4>
      <p>Henüz hiçbir ilan yayınlanmamış.</p>
      <hr>
      <!-- if request.user.is_authenticated -->
      <p class="mb-0">İlan yayınlamak için <a href="{% url 'advert:addadvert' %}">buraya</a> tıklayın.</p>
      <!-- else -->
      <p class="mb-0">İlan yayınlamak için <a href="{% url 'user:login' %}">buraya</a> tıklayın.</p>
      <!-- endif -->
    </div>
  </div>
  <!-- else -->
  <!-- for advert in all_adverts -->
  <div class="col-md-12">
    <div class="card">
      <!-- include "includes/advertcard.html" -->
    </div>
  </div>
  <!-- endfor -->
  <!-- endif -->
</body>
</html>
```

layout.html sayesinde bütün sayfalarda kullanılması gereken temel gereksinimler tek seferde entegre edildi.

```
{% if not all_adverts %}

<div class="col-md-12">
  <div class="alert alert-danger" role="alert">
    <h4 class="alert-heading">Uyar</h4>
    <p>Henüz hiçbir ilan yayınlanmamış.</p>
    <hr>
    <!-- if request.user.is_authenticated -->
    <p class="mb-0">İlan yayınlamak için <a href="{% url 'advert:addadvert' %}">buraya</a> tıklayın.</p>
    <!-- else -->
    <p class="mb-0">İlan yayınlamak için <a href="{% url 'user:login' %}">buraya</a> tıklayın.</p>
    <!-- endif -->
  </div>
</div>

{% else %}

{% for advert in all_adverts %}

  <div class="col-md-12">
    <div class="card">
      <!-- include "includes/advertcard.html" -->
    </div>
  </div>
</div>

{% endfor %}

{% endif %}
```

Django'nun sağlamış olduğu yapı kullanılarak frontend ile backend fonksiyonları arasında bağlantı sağlanıp veriler aktarıldı.

2.4.2 Backend Mimari Çözümleri

Şekil 2.2.5'te görüldüğü üzere modüller ayrı Django app'lerine ayrılarak modülerlik sağlanmıştır. Kullanıcılardan gelen istekler ilgili modüllerde işlenerek response verilmektedir

2.4.2.1 URLs: Her uygulamanın içerdiği URL yapıları urls.py dosyasında bulunmaktadır.

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', index, name= "index"),  
    path('user/', include("user.urls")),  
    path('event/', include("event.urls")),  
    path('advert/', include("advert.urls")),  
    path('api/', include("api.urls")),  
    path('chat/', include("chat.urls")),  
]
```

Ana uygulamanın urlpatterns yapısı.

2.4.2.2 Views: Belirli sayfalara ait kullanıcılardan gelen isteklere ve front-end'e verilecek olan cevapların düzenlendiği kısımdır.

```
def advertdetail(request, id):  
    advert = get_object_or_404(Advert, id=id)  
  
    context = {  
        "advert" : advert  
    }  
  
    return render(request, "advertdetail.html", context)
```

Advert uygulamasında bulunan örnek bir view fonksiyonu.

2.4.2.3 Models: Veritabanına kaydetmek üzere bilgileri içeren sınıflardır. Her uygulama kendi ile ilişkili modelleri içermektedir.

```
class Advert(models.Model):
    event = models.ForeignKey(Event, on_delete=models.CASCADE, related_name='adverts', null=True)
    author = models.ForeignKey(User, on_delete=models.CASCADE, null=True)
    price = models.IntegerField(null=True)
    seller_description = models.TextField(null=True)
    created_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.author.username
    class Meta:
        ordering = ['-created_at']
```

Advert sınıfına ait örnek bir model yapısı.

Proje Yöneticisi

Sinan Kocatürk

Enes Ceviz

Hüseyin Uğur Aydın