

YAZILIM LABORATUVARI-1

PROJE-2

1. Zehra KARABEKTAŞ

Bilgisayar Mühendisliği

Kocaeli Üniversitesi

Kocaeli, Türkiye

zehrakarabektas679@gmail.com

Özetçe—Bu proje, kullanıcıların etkinlikler oluşturabileceği, katılabileceği ve etkinlikler etrafında sosyal etkileşim kurabileceği web tabanlı Etkinlik Planlama Platformu geliştirme amaçlanmıştır.

Anahtar Kelimeler — Web Uygulaması; Veritabanı Yönetimi; C#; MSSQL ; ASP.NET ; HTML ; MVC Mimarisi ; API Entegrasyonu; Etkinlik Yönetimi; Kişiselleştirilmiş Etkinlik Önerileri .

I. ÖZET

Bu proje, kullanıcıların etkinlikler oluşturabileceği bulunabileceği bir web tabanlı Akıllı Etkinlik Planlama Platformu geliştirmeyi amaçlamaktadır. Kullanıcılar, kişiselleştirilmiş etkinlik önerileri alacak, etkinliklerde sohbet edebilecek ve etkinlikleri harita üzerinden takip edebilecektir. Proje, web programlama, veritabanı tasarımı, dinamik kullanıcı deneyimi ve gerçek zamanlı veri işleme konularında bilgi kazandırmayı hedeflemektedir. Aynı zamanda, kural tabanlı kişiselleştirilmiş etkinlik önerileri sunan bir sistem ve API entegrasyonları ile harita ve rota planlama işlevlerini içermektedir. Ana bileşenler arasında kullanıcı arayüzü, etkinlik yönetimi, kullanıcı profilleri, sohbet özellikleri ve admin paneli bulunmakta olup, bu sistem, kullanıcıların etkinliklere katılımını artırmak ve sosyal etkileşimi teşvik etmek için çeşitli fonksiyonları barındırmaktadır. Proje ayrıca, zaman çakışmalarını önlemek için bir algoritma, etkinliklerin harita üzerinde gösterilmesi, ve kullanıcıların etkinliklere katılımı üzerinden puanlar kazanmasını sağlayacak bir oyunlaştırma sistemi de içermektedir. Sonuç olarak, bu proje modern teknolojileri kullanarak etkinlik planlama ve yönetim süreçlerini kolaylaştırmak ve sosyal etkileşimi güçlendirmek için kapsamlı bir platform sunmayı amaçlamaktadır.

II. GİRİŞ

Bu projede, kullanıcılara yönelik bir dizi işlevsellik sağlanarak, web tabanlı bir Akıllı Etkinlik Planlama Platformu oluşturulmuştur. Uygulama, kullanıcı etkileşimini artırmak ve etkinlik planlama sürecini daha verimli hale getirmek amacıyla aşağıdaki özellikleri sunmaktadır: Kullanıcılar, platforma kaydolabilir, mevcut hesaplarına giriş yapabilir, şifrelerini E-posta yolu ile sıfırlayabilir ve profillerini güncelleyebilirler. Bu işlemler, kullanıcı doğrulama ve güvenlik önlemleriyle desteklenmiştir. Ayrıca, kullanıcılar etkinlikler oluşturabilir, var olan etkinlikleri güncelleyebilir, silebilir ve katılım sağlayabilirler. Etkinlikler, kullanıcıların ilgi alanlarına ve tercihleri doğrultusunda önerilen etkinliklerle etkileşime girer. Kullanıcıların geçmiş etkinlik katılımlarına ve ilgi alanlarına dayalı olarak, kişiselleştirilmiş etkinlik önerileri

sağlanmaktadır. Bu sistem, kullanıcı deneyimini daha ilgi çekici hale getirmeyi amaçlamaktadır. Etkinliklerin zaman dilimlerini dikkate alarak, kullanıcılara çakışan etkinlikler hakkında uyarılar verilir. Böylece kullanıcılar, etkinliklerini çakışma olmadan planlayabilirler. Ayrıca, platform üzerinde kullanıcılar diğer kullanıcılarla mesajlaşarak etkinlikler hakkında bilgi alışverişinde bulunabilirler. Etkinliklerin konumları harita üzerinde gösterilmekte, katılımcılara en uygun rotalar önerilmektedir. Kullanıcılar, katıldıkları etkinliklere göre puan ve başarı kazanabilirler. Bu bilgiler, kullanıcıların profil sayfalarında görüntülenmektedir. Adminler, etkinliklerin onaylanması ve düzenlenmesi işlemleri ile kullanıcı yönetimini gerçekleştirebilirler. Bu yetkiler, platformun güvenliği ve işleyişi açısından kritik öneme sahiptir.

Kullanıcı dostu bir arayüz, kullanıcı etkileşimini kolaylaştıracak şekilde tasarlanmış olup, modern web tasarım prensiplerine uygun olarak geliştirilmiştir. Kullanıcı güvenliği, şifreleme yöntemleri, güvenli oturum yönetimi ve doğru yetkilendirme mekanizmaları kullanılarak sağlanmıştır. Projede, MVC (Model-View-Controller) mimarisi benimsenmiştir. MVC mimarisi, uygulamanın verilerini (Model), kullanıcı arayüzünü (View) ve iş mantığını (Controller) birbirinden ayırarak, daha modüler ve bakım kolaylığı sağlayan bir yapı sunmaktadır. Bu sayede, uygulama geliştirme süreci daha verimli hale gelmiş ve bakım, güncelleme işlemleri daha basit bir şekilde yapılabilir olmuştur. Bu özellikler ve kullanılan teknolojiler, platformun kullanıcı dostu, güvenli ve fonksiyonel olmasını sağlamaktadır. MVC yapısının sağladığı esneklik ve sürdürülebilirlik sayesinde, proje sürekli olarak geliştirilebilecek ve ölçeklenebilir bir yapıya sahiptir.

III. YÖNTEM

Bu proje, modern web uygulamaları geliştirmek için yaygın kullanılan bir dizi teknoloji ve mimariyi kullanılarak inşa edilmiştir. Geliştirme sürecinde MVC (Model-View-Controller) mimarisi, MSSQL veritabanı, C# programlama dili, Google Maps API ve Bootstrap gibi popüler kütüphaneler kullanılmıştır.

A. Veritabanı Tasarımı

Sistemde Kullanıcılar, Etkinlikler, Katılımcılar, İlgi Alanları, Mesajlar, Puanlar, Bildirimler ve Etkinlik Kategorileri gibi temel tablolar bulunmaktadır. Tasarımda, birden çoğa (one-to-many) ve çoktan çoğa (many-to-many) ilişkiler, veri yapısını düzenli ve tutarlı bir şekilde kurmak için kullanılmıştır. Aynı zamanda normalizasyonun temel ilkeleri

yerine getirerek sistemin güvenilirliğini ve verimliliğini arttırmıştır. Migration işlemleri ile tablolar oluşturuldu.

Kullanıcılar tablosu, platformdaki kullanıcıların bilgilerini saklar. Kullanıcı adı, şifre, e-posta, konum, ad, soyad, doğum tarihi, cinsiyet, telefon numarası ve profil fotoğrafı gibi detaylar bu tabloda yer alır. Ayrıca kullanıcıların sistemdeki rolü de bu tabloda tanımlıdır. Kullanıcıların ilgi alanları, ayrı bir yapı olan Kullanıcı İlgi Alanları tablosu ile ilişkilendirilmiştir. Burada Kullanıcılar ile Kullanıcı İlgi Alanları arasında birden çoğa (one-to-many) ilişki kurulmuştur.

Etkinlikler tablosu, platformdaki tüm etkinliklerin bilgilerini içerir. Etkinlik adı, açıklama, tarih, saat, süresi, konumu, kategorisi ve ilgi alanı gibi özellikler bu tabloda yer alır. Ayrıca, etkinliği düzenleyen kullanıcının bilgisi de tutulur. Kullanıcılar ve Etkinlikler tablosu arasında etkinliği oluşturan kullanıcıyı belirlemek için birden çoğa (one-to-many) ilişki bulunmaktadır. Etkinlikler ayrıca ilgili ilgi alanları ilişkilendirilerek sınıflandırılmıştır; burada Etkinlikler ile İlgi alanları arasında bir- birden çoğa (one-to-many) ilişkisi kurulmuştur.

Katılımcılar tablosu, kullanıcıların hangi etkinliklere katıldığını izlemek için tasarlanmıştır. Bu tablo, kullanıcılar ve etkinlikler arasında bir çoktan çoğa (many-to-many) ilişki kurarak kullanıcıların etkinlik katılımlarını kayıt altına alınmıştır.

İlgi Alanları tablosu, kullanıcıların ve etkinliklerin ilişkilendirilebileceği çeşitli kategorileri içerir. Kullanıcılar ve İlgi Alanları arasındaki ilişki, Kullanıcı İlgi Alanları tablosu aracılığıyla bir çoktan çoğa (many-to-many) ilişki şeklinde modellenmiştir. Bu yapı, kullanıcıların tercih ettikleri ilgi alanlarına göre kişiselleştirilmiş öneriler sunulmasını sağlar.

Mesajlar ve Etkinlik Mesajları tabloları, kullanıcılar arasında iletişim kurmayı ve etkinliklerle ilişkili mesajlaşmayı destekler. Mesajlar tablosunda, Kullanıcılar ile Mesajlar arasında birden çoğa (one-to-many) ilişki bulunmaktadır. Benzer şekilde, Etkinlik Mesajları tablosunda da Kullanıcılar ile Etkinlik Mesajları arasında bir birden çoğa (one-to-many) ilişki kurulmuştur.

Puanlar tablosu, kullanıcıların platformdaki aktiviteleri sonucunda kazandıkları puanları ve bu puanların kazanılma tarihlerini içerir. Kullanıcılar ve Puanlar tablosu arasında bir birden çoğa (one-to-many) ilişki bulunmaktadır. Bu tablo, kullanıcı motivasyonunu artırmak için ödül sisteminin bir parçası olarak tasarlanmıştır.

Bildirimler tablosu, kullanıcılara önemli olaylarla ilgili bilgi vermek amacıyla kullanılır. Kullanıcılar ve Bildirimler tablosu arasında bir birden çoğa (one-to-many) ilişki bulunmaktadır. Bu tablo sayesinde kullanıcılar, etkinlik davetleri veya mesaj bildirimleri gibi önemli gelişmelerden haberdar olabilir.

Etkinlik Kategorileri tablosu, etkinliklerin kategorize edilmesini sağlar ve bu kategorilerle ilgili ek bilgiler (örneğin, kategori görselleri) sunar. Ayrıca, kategori onay durumu gibi bilgiler bu tabloda yer alır. Etkinlik Kategorileri ile İlgi Alanları arasında bir birden çoğa (one-to-many) ilişki bulunmaktadır.

Bu veritabanı tasarımı, one-to-many ve many-to-many ilişkiler, kullanıcıların etkinliklere katılımını, ilgi alanlarını ve

sosyal etkileşimlerini takip etmek için etkili bir sistem oluşturulmuş olup gerçekleştirilmiştir. Ayrıca, ilişkisel model sayesinde veri bütünlüğü korunmuş ve tutarlı bir yapı oluşturulmuştur.

B. Kullanıcı Arayüzü (GUI) Tasarımı

Sistem için tasarlanan kullanıcı arayüzü, kullanıcıların etkinlikleri görüntüleyebileceği, katılabileceği ve kendi etkinliklerini oluşturabileceği işlevsel bir yapı sunmaktadır. Web arayüzü, hem kullanıcılar hem de adminler için özel olarak özelleştirilmiş düzenlere sahip olup, web sayfasına iki farklı giriş sistemi eklenerek işlemler kolaylaştırılmıştır. Platformda, kullanıcılar ve adminler için farklı özelleştirilmiş sayfalar bulunmaktadır. Ana sayfada kullanıcılar, kişiselleştirilmiş etkinlik önerilerini görür ve harita üzerinde, ilgi alanlarına, katıldıkları etkinliklere ve benzer ilgi alanlarına sahip yalnızca onaylı etkinlikleri markerlarla görüntüler. Adminlerin ana sayfasında ise tüm etkinlikler, harita üzerinde farklı markerlarla, onay durumlarına göre (onaylı, onay bekleyen, reddedilen etkinlikler) gösterilmektedir. Adminler, tüm etkinlikleri görüntüleyip yönetebilirken, kullanıcılar yalnızca onaylı etkinlikleri ve kendi oluşturdukları etkinlikleri görüntüleyebilir ve yönetebilirler. Profil sayfası, kullanıcıların kişisel bilgilerini güncelleyebileceği, etkinlik sayfası ise etkinliklerin detaylarını görüntüleyebileceği sayfalardır. Mesajlar sayfası, kullanıcıların yalnızca kendi etkinliklerine ait sohbetleri yönetebilmelerini sağlarken, adminler tüm mesajları yönetebilir. Kullanıcılar, katıldıkları ve düzenledikleri etkinliklerin sohbetlerine erişebilirken, adminler etkinliklerin onay durumlarını ve kullanıcı hesaplarını yönetebileceği sayfalara sahiptir.

C. Yazılım Mimarisi

Uygulama MVC (Model-View-Controller) yani katmanlı mimari prensiplerine dayanarak yazılmıştır. Dört ana katman üzerine kuruldu. Bunlar: EntityLayer (Varlık Katmanı), DataAccessLayer (Veri Erişim Katmanı), BusinessLayer (İş Mantığı Katmanı) ve Sunum Katmanı. Bu katmanlar, uygulamanın modüler ve esnek bir yapıya sahip olmasını sağlayarak, her bir katmanın kendi sorumluluklarına odaklanmasını ve bakımının kolaylaşması sağlandı.. EntityLayer veritabanı ile doğrudan etkileşimi yönetildi ve sağlandı, DataAccessLayer katmanında veri işlemleri gerçekleştirildi, BusinessLayer katmanında iş mantığını uygulandı ve Sunum katmanında kullanıcı ile etkileşimi yönetildi.

D. Algoritmalar ve Yöntemler

Bu projede, ASP.NET MVC çerçevesi kullanılarak bir etkinlik planlama uygulaması geliştirilmiştir. Projede kullanılan yöntemler, kullanıcıların etkinlik oluşturma, katılım, bildirim ve öneri alımı gibi süreçlerde etkili ve güvenli bir deneyim yaşamalarını sağlamaktadır. Uygulamada veri işlemleri Entity Framework üzerinden gerçekleştirilmiştir. Veri erişim katmanı (DAL) ile iş katmanı (Business Layer) arasında bir bağlantı sağlanmış, Manager sınıfları kullanılarak veriler işlenmiştir. Bu katmanda kullanıcı, etkinlik, bildirim ve ilgi alanı gibi

varlıklar üzerinde ekleme, silme, güncelleme ve listeleme işlemleri yapılmıştır. FluentValidation kütüphanesi, kullanıcıdan gelen verilerin doğruluğunu kontrol etmek için kullanılmıştır. Örneğin, kullanıcı kayıt veya etkinlik oluşturma işlemlerinde, giriş verileri doğrulanarak sistemin güvenliği artırılmıştır. Validasyon hataları, kullanıcıya geri bildirim olarak sunulmuş ve eksik veya hatalı veri girişlerinin önüne geçilmiştir. Rol tabanlı bir yetkilendirme sistemi kullanılarak kullanıcıların erişim hakları yönetilmiştir. Admin ve kullanıcı rolleri için özel filtreler (AdminAttribute, KullaniciAttribute) tanımlanmıştır. Örneğin, etkinlikleri onaylama veya silme işlemleri yalnızca admin yetkisi olan kullanıcılar tarafından gerçekleştirilebilmektedir. Aynı şekilde, anonim kullanıcıların belirli sayfalara erişimi kısıtlanmıştır. Uygulama, kullanıcıların profil fotoğrafları ve etkinlik görselleri gibi dosyaları sunucuya yüklemelerine imkan tanımıştır. Dosya yükleme işlemlerinde, çakışmaların önüne geçmek için benzersiz dosya isimlendirme yöntemleri kullanılmıştır. Şifre yenileme işlemleri, SMTP protokolü üzerinden e-posta gönderimi ile desteklenmiştir. Kullanıcılar, şifrelerini unutmaları durumunda e-posta aracılığıyla yeni bir şifre oluşturabilmişlerdir. Bu işlem, Gmail servisi kullanılarak güvenli bir şekilde gerçekleştirilmiştir.

Etkinlik öneri mekanizması, kullanıcıların ilgi alanlarına ve katıldıkları etkinliklere göre kişiselleştirilmiş öneriler sunmuştur. Kullanıcının tercih ettiği ilgi alanlarına uygun etkinlikler sıralanmış, aynı zamanda benzer alanlara yönelik öneriler yapılmıştır. Öneri algoritması, kullanıcının geçmiş davranışlarını dikkate alarak dinamik bir şekilde çalışmaktadır. Etkinliklerin yönetimi, farklı onay durumlarına göre kategorize edilmiştir. Onay bekleyen, onaylanmış veya reddedilmiş etkinlikler sistematik bir şekilde işlenmiştir. Admin kullanıcılar, bu etkinliklerin durumlarını değiştirebilme yetkisine sahiptir. Son olarak, uygulama kullanıcı oturum bilgilerini Session nesnesi ile yönetmiştir. Kullanıcı adı, profil fotoğrafı, toplam puan gibi bilgiler oturum sırasında saklanmış ve uygulamanın çeşitli bölümlerinde kullanılmak üzere erişilmiştir. Bu yöntem, kullanıcı deneyimini iyileştirmek ve dinamik içerik sunmak için etkili bir çözüm olmuştur. Genel olarak, bu projede kullanılan yöntemler ve teknolojiler, uygulamanın güvenli, ölçeklenebilir ve kullanıcı dostu bir yapıya sahip olmasını sağlamıştır. Projenin mimarisi, modern yazılım geliştirme standartlarına uygun bir şekilde tasarlanmıştır.

Bu projede, etkinlik önerileri sunarken kullanılan yöntem şu şekilde çalışmaktadır: Kullanıcının katıldığı etkinlikler, ilgi alanları ve bu ilgi alanlarına benzer ilgi alanları veri tabanından çekilerek liste haline getirilmiştir. Bu işlemler Business Layer tarafında yapılmış, veriler işlendikten sonra Controller'a gönderilmiştir. Controller, işlenen verileri aldıktan sonra gerekli filtrelemeleri yaparak, kullanıcının daha önce katılmadığı ve düzenlemediği etkinlikleri belirlemiştir. Ayrıca, sadece onaylı olan etkinlikler seçilmiştir. Filtrelenen etkinlikler, ViewBag aracılığıyla HTML sayfasına iletilmiştir. HTML sayfasında ise, Geocode yöntemi kullanılarak etkinliklerin konumları belirlenmiş ve haritada farklı renklerdeki işaretçiler (marker) ile kullanıcıya sunulmuştur.

Bu sayede, etkinlikler görsel olarak kullanıcıya kolayca ulaşılabilir hale getirilmiştir.

IV. DENEYSEL SONUÇLAR

Bu bölümde, projenin genel görünümü ve işlevleri hakkında bilgi verilmektedir. Projenin yapısı ve gerçekleştirilen işlevler detaylı bir şekilde sunulmuştur. Kullanıcı arayüzü, ASP.NET ile tasarlanmıştır.

A. Ana sayfa

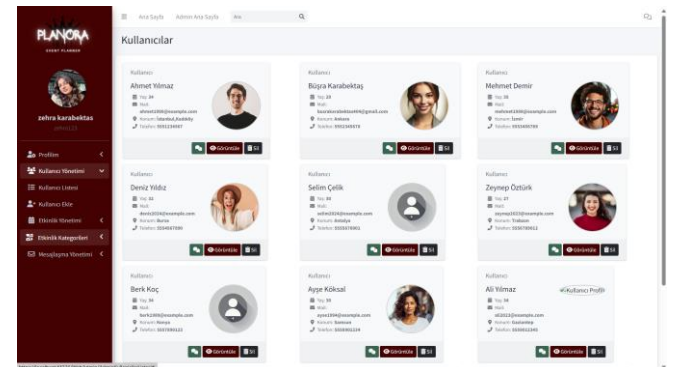


B. Admin Ana Sayfa



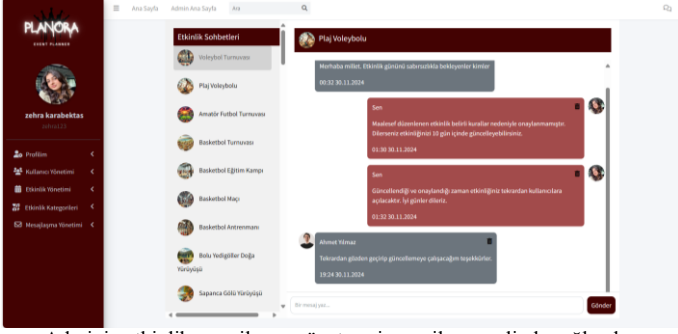
Harita üzerinde tüm etkinlikler markerlar ile admin ana sayfasında gösterildi.

C. Admin Kullanıcı Yönetimi



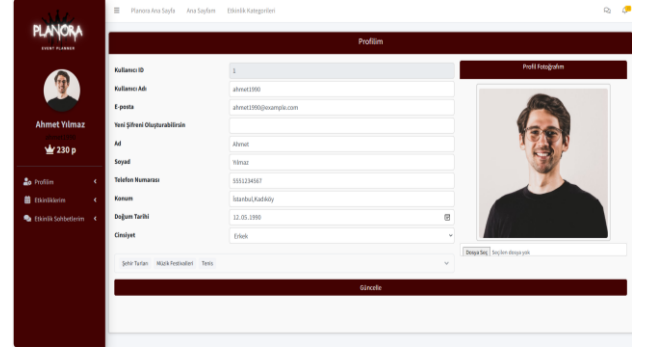
Admin sayfasında kullanıcı yönetimi sağlandı.

D. Admin Mesaj Yönetimi



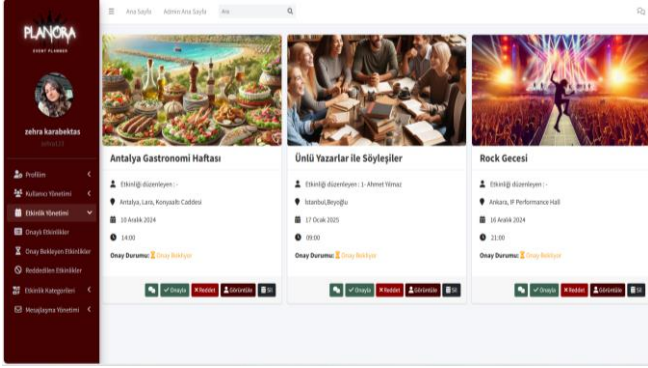
Adminin etkinlik mesajlarını yönetmesi mesajlar panelinde sağlandı.

G. Kullanıcı Profili



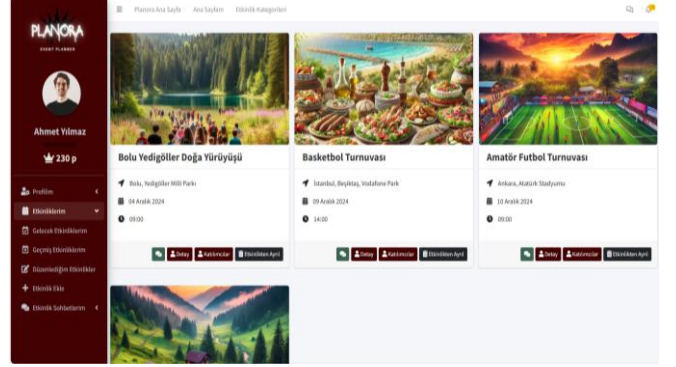
Kullanıcının profilini güncelleme sayfası.

E. Admin Etkinlik Yönetimi



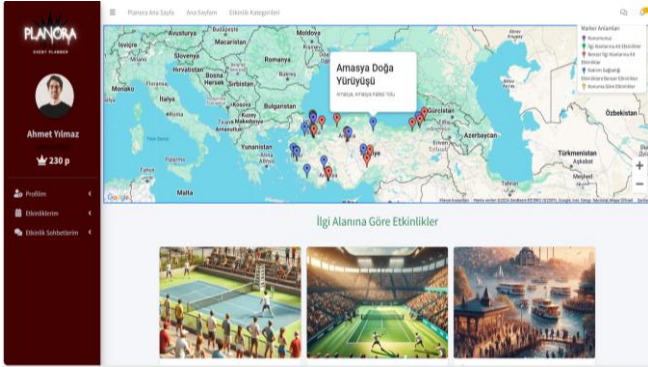
Adminlerin etkinlik silme güncelleme onaylama işlemleri

H. Kullanıcının Gelecek ve Geçmişte Düzenlediği Etkinlikler



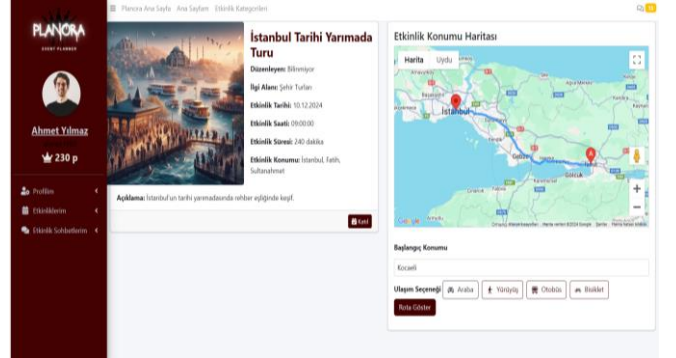
Kulanıcı etkinlikleri listeli şekilde gösterildi.

F. Kullanıcı Ana Sayfa



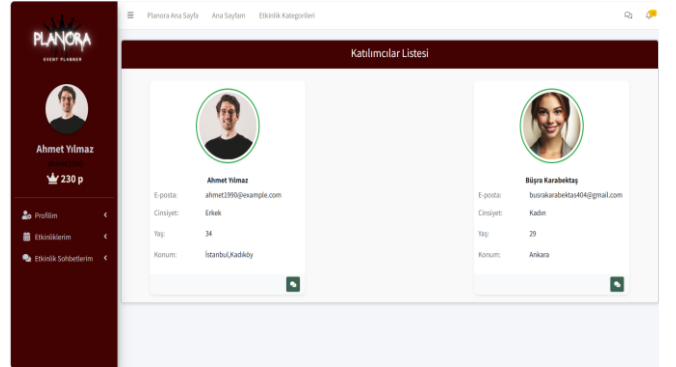
Kullanıcı ana sayfasında öneriler harita üzerinde gösterildi.

İ. Etkinlik Detay Sayfası



Etkinliklerin haritada rota göstergesi ve bilgi detayları.

J. Katılımcı Sayfası



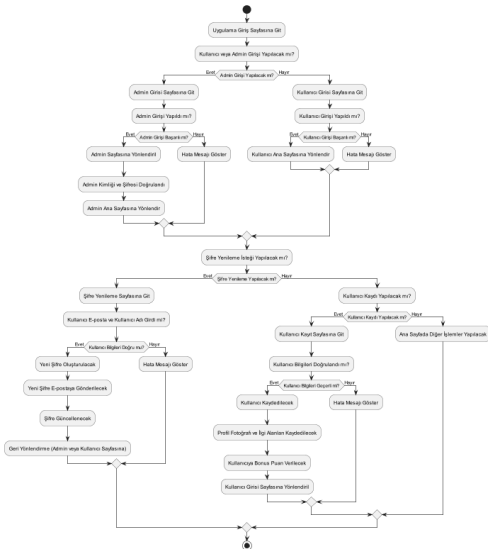
Etkinliğe katılımcı listesi.

Bu proje, web programlama, veritabanı yönetimi, kural tabanlı öneri sistemi ve iletişim gibi modern teknolojilerin bir arada kullanılarak kapsamlı bir web uygulama geliştirme süreci geçirilmiştir. Hem teknik hem de kullanıcı deneyimi açısından güçlü bir altyapı sağlayarak sosyal etkinlikleri planlamak ve yönetmek için yenilikçi bir web sitesi sayfası ortaya çıkarılmıştır.

- [1] Google Maps API,Erişim adresi: <http://developers.google.com/maps>
- [2] Microsoft. (n.d.). *Create a C# Windows Forms App in Visual Studio*. Erişim adresi: <https://learn.microsoft.com/tr-tr/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022>
- [3] Google,Google Cloud Platform,Erişim adresi: <https://cloud.google.com>
- [4] Youtube.com
- [5] Bootstrap, Bootstrap Framework ve Icons, Erişim adresi: <http://getbootstrap.com>

[illegible]

Akış Diyagramları



```

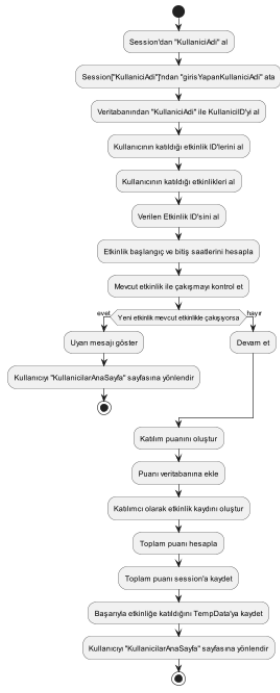
graph TD
    Start(( )) --> A[Session'ın "KullanıcıAdı" al]
    A --> B[Session'ın "KullanıcıAdı"ndan "girisYapilanKullanıcıAdı" al]
    B --> C[Veritabanından "KullanıcıAdı" ile KullanıcıOlul al]
    C --> D[Etkirlik modelini doğrula]
    D --> E{evet}
    D --> F{hayır}
    F --> G[Hata mesajlarını ekle]
    G --> H[Onaylılgı alanlarını al ve ViewBag'a at]
    H --> I(( ))
    E --> J[İlgili alanı ID'sini ekleme kontrol et]
    J --> K{evet}
    J --> L{SelecedilgiKategori nolu değeri}
    L --> M{evet}
    L --> N{hayır}
    M --> O[Yeni ilgi alanını ekle]
    O --> P[Yeni ilgi alanı ID'sini ekleme ate]
    P --> Q{ }
    N --> R[Yeni ilgi alanı girilmez]
    R --> S[Hata mesajını göster]
    S --> T(( ))
    Q --> U[Etkirlik fotoğrafını kontrol et]
    U --> V{evet}
    U --> W{Yeni fotoğraf varsa}
    W --> X[Yeni fotoğraf kaydet]
    W --> Y[Fotoğraf kaydedilmez]
    X --> Z{ }
    Y --> Z
    Z --> AA[Etkirlik bilgilerini veritabanına ekle]
    AA --> AB[Etkirlik oluşturma puanı ekle]
    AB --> AC[Toplam puanı hesapla]
    AC --> AD[Toplam puanı session'a kaydet]
    AD --> AE[Başarı mesajını TempData'ya kaydet]
    AE --> AF[Kullanıcıyı "Düzenlenen Etkirlikler Listesi" sayfasına yönlendir]
    AF --> AG(( ))
  
```

```

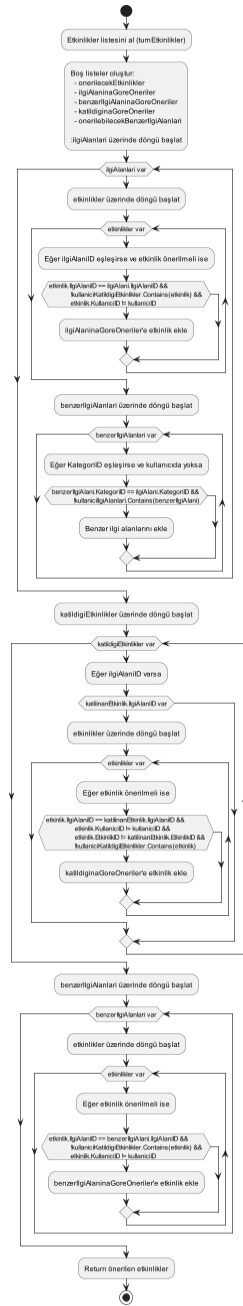
graph TD
    Start(( )) --> A[Mevcut ekinlik veritabanından al]
    A --> B[Ekinlik modelini doğrula]
    B --> C{Ekinlik doğrulaması geçerli değilse}
    C -- hayır --> D[Hata mesajlarını ekle]
    D --> E[Onaylı ilgi alanlarını al ve ViewBag'e ata]
    E --> End1(( ))
    C -- evet --> F[İlgili alan ID'sini kontrol et]
    F -- hayır --> G[Yeni ilgi alanı girilince]
    G -- hayır --> H[Mevcut ilgi alanını koru]
    H --> I[Hata mesajı göster]
    I --> J[Onaylı ilgi alanlarını al ve ViewBag'e ata]
    J --> End2(( ))
    G -- evet --> K[Yeni ilgi alanını ekle]
    K --> L[Yeni ilgi alanı ID'sini ekinliğe ata]
    F -- evet --> L
    L --> M{ }
    M --> N[Ekinlik fotoğrafını kontrol et]
    N -- hayır --> O[Mevcut fotoğrafı koru]
    N -- evet --> P[Yeni fotoğraf varsa]
    P -- hayır --> O
    P -- evet --> Q[Fotoğrafın geçerli olup olmadığını kontrol et]
    Q -- hayır --> R[Fotoğraf kaydedilmez]
    R --> O
    Q -- evet --> S[Yeni fotoğrafı kaydet]
    S --> M
    O --> T{ }
    T --> U[Ekinlik bilgilerini veritabanına güncelle]
    U --> V[Başarıyla güncellendi mesajı göster]
    V --> W[Kullanıcıya "Düzenlenen Ekinlikler Listesi" sayfasına yönlendir]
    W --> End3(( ))
  
```

The flowchart illustrates the E-Business Registration Process. It begins with a start node leading to 'Mevcut ekinlik veritabanından al' (Get from existing event database). This is followed by 'Ekinlik modelini doğrula' (Verify event model). A decision point 'Ekinlik doğrulaması geçerli değilse' (Event verification is not valid) branches the flow. If 'hayır' (no), it goes to 'Hata mesajlarını ekle' (Add error messages) and then 'Onaylı ilgi alanlarını al ve ViewBag'e ata' (Get approved interests and assign to ViewBag), leading to an end node. If 'evet' (yes), it proceeds to 'İlgili alan ID'sini kontrol et' (Check related area ID). Another decision point 'Selektörlüğünü ID null değilse' (If selection ID is not null) follows. If 'hayır', it goes to 'Yeni ilgi alanı girilince' (When new interest is entered). If 'hayır' here, it goes to 'Mevcut ilgi alanını koru' (Keep existing interest), then 'Hata mesajı göster' (Show error message), and finally 'Onaylı ilgi alanlarını al ve ViewBag'e ata' (Get approved interests and assign to ViewBag), leading to an end node. If 'evet' at either decision point, it goes to 'Yeni ilgi alanını ekle' (Add new interest) and then 'Yeni ilgi alanı ID'sini ekinliğe ata' (Assign new interest ID to event). This leads to a merge point before 'Ekinlik fotoğrafını kontrol et' (Check event photo). A decision 'Yeni fotoğraf varsa' (If new photo exists) follows. If 'hayır', it goes to 'Mevcut fotoğrafı koru' (Keep existing photo). If 'evet', it goes to 'Fotoğrafın geçerli olup olmadığını kontrol et' (Check if photo is valid). Another decision 'Fotoğraf geçerli ise' (If photo is valid) follows. If 'hayır', it goes to 'Fotoğraf kaydedilmez' (Photo is not saved) and then 'Mevcut fotoğrafı koru' (Keep existing photo). If 'evet', it goes to 'Yeni fotoğrafı kaydet' (Save new photo). Both 'Fotoğraf kaydedilmez' and 'Yeni fotoğrafı kaydet' lead to a merge point before 'Ekinlik bilgilerini veritabanına güncelle' (Update event information to database). This is followed by 'Başarıyla güncellendi mesajı göster' (Show successfully updated message) and finally 'Kullanıcıya "Düzenlenen Ekinlikler Listesi" sayfasına yönlendir' (Redirect user to "List of Edited Events" page), leading to the final end node.

Şekil 4. Etkinlik Güncelleme



Şekil 5. Etkinlik Katıl



Şekil 5. Kullanıcı Etkinlik Öneri Algoritma Akışı