

# İSTANBUL MEDENİYET ÜNİVERSİTESİ

## MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ İNTERNET PROGRAMLAMA DERSİ FINAL PROJESİ

**Villa Kiralama Sitesi**

22120205059 Azra Öykü Ulukan

22120205383 Çağla Sarf

**Ders Sorumlusu**

Dr. Öğr. Üyesi Muhammet Sinan BAŞARSLAN

Mayıs, 2025

İstanbul Medeniyet Üniversitesi, İstanbul

# **İÇİNDEKİLER**

- 1. GİRİŞ**
  - 1.1. Projenin Amacı
  - 1.2. Proje Kısıtları
- 2. MATERİYAL VE YÖNTEM**
  - 2.1. C#: Projede Kullanılan Yazılım Dili
  - 2.2. Visual Studio: IDE
  - 2.3. MongoDB: Database İçin Kullanılan Tasarım Platformu
- 3. UYGULAMA**
  - 3.1. Uygulamanın Database Tasarımı
    - 3.1. MongoDB Bağlantısı ve DbContext Yapılandırması Kesit
    - 3.2. MongoDB Koleksiyonları Kesiti
    - 3.3. MongoDB VillaDb Banners Kesiti
  - 3.2. Uygulamanın Ekranları ve İşlevleri
    - 3.4. Villa Ana Sayfa
    - 3.5. Mevcut Kullanıcıların Oturum Oluşturabileceği Form Ekranı
    - 3.6. Yeni Kullanıcıların Hesap Oluşturabileceği Form Ekranı
    - 3.7. Seçilen Villaya Ait Sık Sorulan Soruların Yanıtlanacağı Bilgi Bölümü
    - 3.8. Firma Tecrübesini ve Başarılarını Yansıtan İstatistik Kartları
    - 3.9. Kullanıcının Villa Türleri Arasında Seçim Yapabileceği Filtreleme Arayüzü
    - 3.10. Kullanıcının Danışmanlarla İletişim Kurabileceği Form ve Firmanın Konumunu Gösteren Harita Entegrasyonu
    - 3.11. Login Doğrulama Kodları, AccountController.cs
    - 3.12. Admin Paneli Ekranı
    - 3.13. Yeni Banner Ekleme Ekranı
    - 3.14. Banner Güncelleme Ekranı
    - 3.15. Gelen Mesajlar Ekranı ve Bildirim Kutusu
- 4. SONUÇ**
- 5. EK**
- 6. KAYNAKÇA**



# 1. GİRİŞ

Bu bölümde projenin amacı ve projede karşılaşılan kısıtlara yer verilmiştir.

## 1.1. Proje'nin Amacı

.NET 7 ve ASP.NET Core teknolojileri kullanılarak geliştirilmiş bir **Villa Kiralama Sitesi** oluşturmaktır. Bu proje, kullanıcıların farklı bölgelerdeki villaları inceleyebildiği, kiralama tarihlerini seçerek rezervasyon yapabildiği, kullanıcı yönetimi ve içerik yönetimi gibi işlemlerin yapılabildiği bir web platformudur. Proje ayrıca bir **admin paneli** içermektedir. Admin paneli üzerinden yeni villa ekleme, var olanları düzenleme/silme, kullanıcıları görüntüleme gibi yönetimsel işlemler gerçekleştirilebilmektedir. Uygulama arayüzü HTML, CSS ve JavaScript ile desteklenmiş, görsel uyumluluk ve kullanıcı deneyimi ön planda tutulmuştur. Veritabanı olarak da MongoDB kullanılmıştır.

## 1.2. Proje Kısıtları

**Kısıt 1:** Projede NoSQL veritabanı olarak **MongoDB** kullanılmıştır. Ancak, ilişkisel veritabanı yapılarında olduğu gibi JOIN işlemleri doğrudan desteklenmediğinden, ilişkili veri sorgulamalarında bazı kısıtlamalar ve performans sorunları yaşanabilemektedir. Bu sebeple, bazı veriler ayrı koleksiyonlar halinde tutulmuş ve uygulama düzeyinde ilişkilendirme yapılmıştır. Bu durum karmaşık sorguların yönetimini zorlaştırmaktadır.

**Kısıt 2:** Admin panelinde kullanılan bazı dinamik HTML içerikler, tarayıcı uyumluluğu açısından farklı sonuçlar verebilir. Özellikle eski tarayıcılar, modern CSS veya JavaScript özelliklerini tam olarak desteklemediğinden dolayı arayüzde bozulmalar yaşanabilir.

**Kısıt 3:** Geliştirme sürecinde .NET 7'nin bazı yeni özellikleri kullanılmıştır. Ancak bu özellikler, daha önceki .NET sürümleriyle uyumlu değildir. Projenin farklı ortamlarda çalıştırılabilmesi için hedef ortamın .NET 7 desteklediğinden emin olunmalıdır.

**Kısıt 4:** Proje geliştirilmeden önce kullanılacak teknolojilerin araştırılıp öğrenilmesinden dolayı geliştirme aşaması bir süre gecikmiştir.

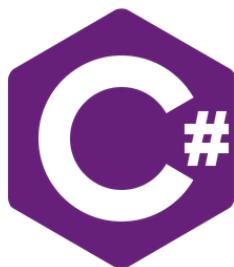
# 2. MATERİYAL METOD

**Projede kullanılan yazılım dili: C#**

- **Projede kullanılan geliştirme ortamı:** Visual Studio
- **Uygulamada kullanılan veri tabanı için seçilen platform:** Firebase (Firestore)

## 2.1. C#: Projede Kullanılan Yazılım Dili

- Bu projede, Microsoft tarafından geliştirilen **C# (C Sharp)** yazılım dili kullanılmıştır.
- **Peki neden C# tercih edilmiştir?** C#, .NET platformunun temel yazılım dili olup nesne yönelimli (OOP) yapısıyla güçlü, okunabilir ve sürdürülebilir bir kod geliştirme imkânı sunar. ASP.NET Core ile birlikte kullanıldığında web uygulamaları için modern, hızlı ve güvenli çözümler geliştirilebilir. C# ayrıca statik tip denetimi, async/await yapısı, LINQ sorguları ve zengin kütüphane desteği sayesinde profesyonel web uygulamaları geliştirmek için ideal bir tercihtir.



Şekil 2.1. C# Dili Logo

## 2.2. Visual Studio: IDE

Proje geliştirilirken, geliştirme ortamı olarak **Visual Studio 2022** tercih edilmiştir. Visual Studio, .NET ve C# ile uyumlu güçlü bir IDE olup, gelişmiş hata ayıklama araçları, kod tamamlama (IntelliSense), GUI tasarım araçları ve entegrasyon destekleriyle profesyonel uygulama geliştirme sürecini hızlandırmaktadır. Özellikle ASP.NET Core ve MongoDB ile çalışırken sunmuş olduğu hazır şablonlar, ekengliler ve veritabanı bağlantı yönetimi sayesinde verimli bir geliştirme deneyimi sunar.



Şekil 2.2. Visual Studio Logo

### **2.3. MongoDB: Database İçin Kullanılan Tasarım Platformu**

- MongoDB, esnek yapısıyla özellikle modern web uygulamaları için tercih edilen, belge (document) tabanlı bir NoSQL veritabanı yönetim sistemidir.
- Bu projede verilerin depolanması için MongoDB kullanılmıştır. MongoDB, JSON benzeri BSON formatı ile verilerin esnek ve ölçeklenebilir şekilde saklanması sağlar. Ayrıca ilişkisel olmayan yapısıyla hızlı sorgulamalar ve kolay veri modellemesi sunar.
- Projede kullanıcı bilgileri, villa detayları, rezervasyon kayıtları gibi tüm veriler MongoDB üzerinde koleksiyonlar halinde tutulmuş; CRUD işlemleri C# ile yazılmış servisler aracılığıyla gerçekleştirilmiştir.



**Şekil 2.3.** MongoDB

Logo

## 3. UYGULAMA

### 3.1. Uygulamanın Database Tasarımı

Uygulamanın veri tabanı tasarımı için **MongoDB** tercih edilmiştir. MongoDB, belge tabanlı (document-oriented) yapısıyla ilişkisel olmayan esnek veri yapılarının kullanılmasına olanak tanır. JSON benzeri BSON formatı ile veri saklanmakta, böylece hızlı ve esnek bir veri yönetimi sağlanmaktadır.

```
11  using Villa.Entity.Entities;
12  using Villa.WebUI.Extensions;
13
14  var builder = WebApplication.CreateBuilder(args);
15
16  // Add services to the container.
17
18  builder.Services.AddServiceExtensions();
19  builder.Services.AddAutoMapper(Assembly.GetExecutingAssembly());
20
21  var mongoDatabase = new MongoClient(builder.Configuration.GetConnectionString("MongoConnection"))
22      .GetDatabase(builder.Configuration.GetSection("DatabaseName").Value);
23
24  builder.Services.AddDbContext<VillaContext>(option =>
25  {
26      option.UseMongoDB(mongoDatabase.Client, mongoDatabase.DatabaseNamespace.DatabaseName);
27  });
28
29  builder.Services.AddIdentity<AppUser, AppRole>().AddEntityFrameworkStores<VillaContext>();
30
31  builder.Services.AddValidatorsFromAssembly(Assembly.GetExecutingAssembly());
32  builder.Services.AddControllersWithViews();
```

Şekil 3.1. MongoDB Bağlantısı ve DbContext Yapılandırması Kesit

Veri tabanı tasarımindaki temel koleksiyonlar (collections) kullanılmıştır.

```
10  using System.Threading.Tasks;
11  using Villa.Entity.Entities;
12
13  namespace Villa.DataAccess.Context
14  {
15      public class VillaContext : IdentityDbContext<AppUser, AppRole, ObjectId>
16      {
17          public VillaContext(DbContextOptions options) : base(options)
18          {
19          }
20      }
21
22      public DbSet<Banner> Banners { get; set; }
23      public DbSet<Contact> Contacts { get; set; }
24      public DbSet<Counter> Counters { get; set; }
25      public DbSet<Deal> Deals { get; set; }
26      public DbSet<Feature> Features { get; set; }
27      public DbSet<Message> Messages { get; set; }
28      public DbSet<Product> Products { get; set; }
29      public DbSet<Quest> Quests { get; set; }
30      public DbSet<Video> Videos { get; set; }
31      public DbSet<SubHeader> SubHeaders { get; set; }
32
33
```

Şekil 3.2. MongoDB Koleksiyonları Kesiti

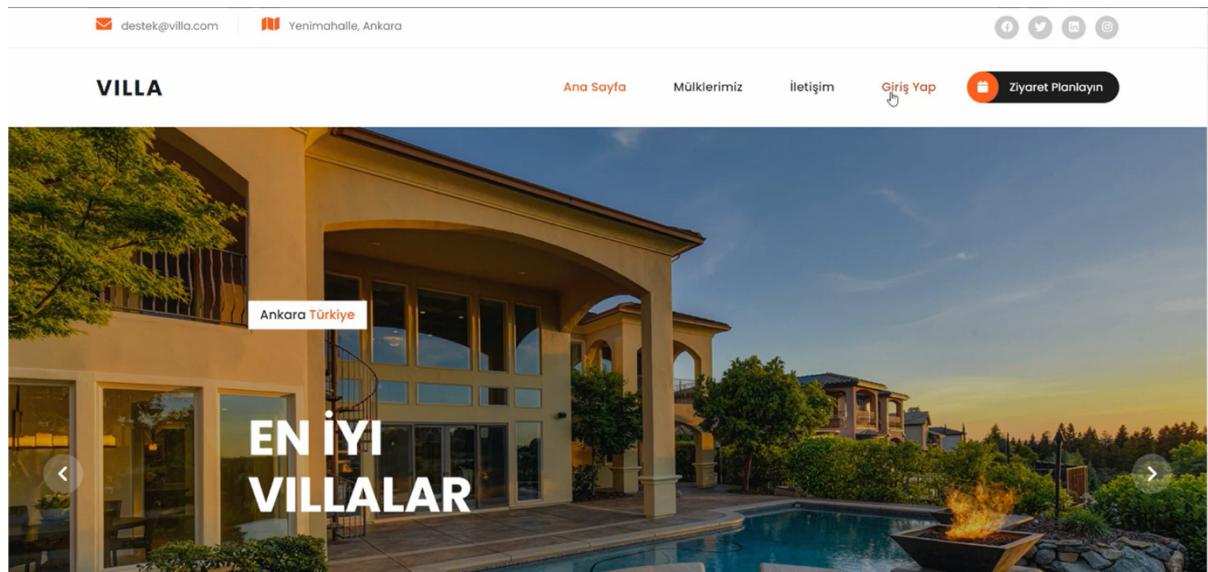
The screenshot shows the MongoDB Compass interface with the following details:

- URL: localhost:27017 > VillaDb > Banners
- Collection: Banners
- Document Count: 3
- Fields:
  - \_id: ObjectId('6817fd6d136f99a57b88b8e3')
  - City: "İstanbul"
  - ImageUrl: "test"
  - Title: "Villa"
- Second Document:
  - \_id: ObjectId('6819ff472d02d3a7933cac60')
  - City: "azra"
  - ImageUrl: "azra"
  - Title: "azra"
- Third Document:
  - \_id: ObjectId('682b20d6db21caab86e373e8')
  - City: "b2"
  - ImageUrl: "b2"
  - Title: "b2"

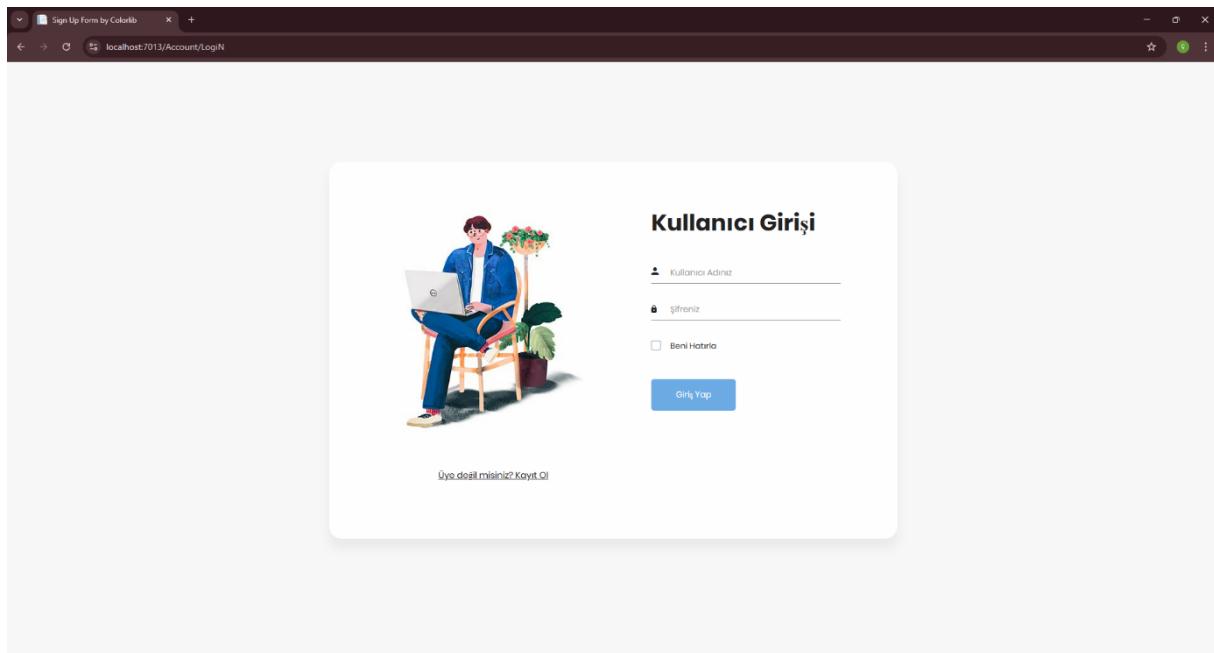
Şekil 3.3. MongoDB VillaDb Banners Kesiti

### 3.2. Uygulamanın Ekranları ve İşlevleri

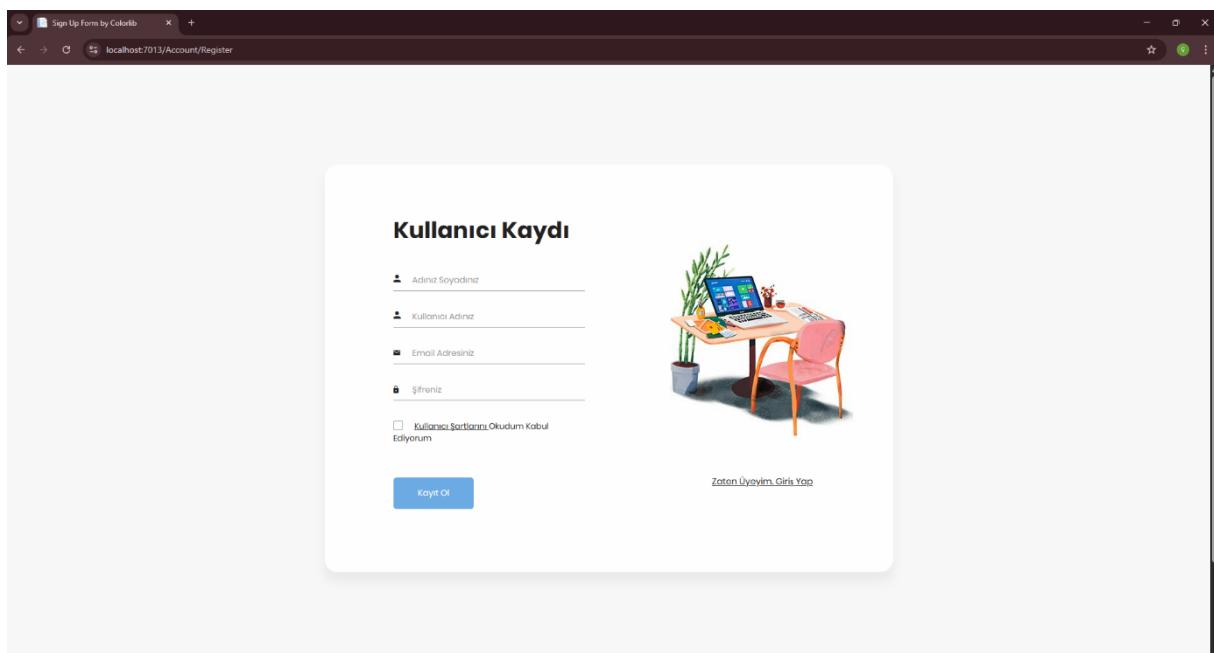
Uygulama başlatıldığında kullanıcıyı doğrudan bir banner görseli ve tanıtım metinleriyle desteklenmiş ana sayfa karşılamaktadır. Bu sayfada üst menüde “Ana Sayfa”, “Mülklerimiz”, “İletişim”, “Giriş Yap” ve “Ziyaret Planlayın” gibi yönlendirici butonlar yer almaktadır. Kullanıcı giriş yapmadığı durumda sayfa genel bilgilendirme niteliğinde olup, villa seçenekleri ve firma hakkında bilgiler sunulmaktadır. “Giriş Yap” butonuna tıklayan kullanıcı, kullanıcı adı ve şifre alanlarını içeren giriş (Login) ekranına yönlendirilir. Giriş ekranında, eksik veya hatalı bilgi girişi yapılması durumunda kullanıcıya uyarı mesajı (popup) gösterilmektedir. Ekranın alt kısmında “Üye değil misiniz? Kayıt Ol” bağlantısı sayesinde yeni kullanıcılar kayıt (Register) ekranına geçiş yapabilmektedir. Kayıt ekranında ad-soyad, kullanıcı adı, e-posta adresi ve şifre gibi zorunlu bilgiler talep edilmekte, isteğe bağlı olarak profil fotoğrafı ve biyografi gibi gelişmiş kullanıcı bilgileri de eklenebilmektedir (geliştirilebilir özellik). Kayıt ekranının altında yer alan “Zaten üyeyim, Giriş Yap” bağlantısı ile tekrar giriş ekranına dönüş mümkündür. Başarılı oturum açma sonrası kullanıcı, sisteme giriş yapmış olur ve sayfa üzerinden filtrelenebilir villa türleri (apartman dairesi, müstakil, çatı katı vb.), detaylı mülk bilgileri ve villa kartlarıyla karşılaşır. Villa kartlarında konum, fiyat, metrekare, kat bilgisi, oda sayısı ve park alanı gibi bilgiler kullanıcıya açıkça sunulmaktadır. Ayrıca “Ziyaret Planlayın” butonu ile iletişim formuna yönlendirme yapılmakta, danışmanlarla birebir iletişim kurulabilmektedir. Harita entegrasyonu ile firmanın lokasyonu detaylı olarak gösterilirken, mesaj formu üzerinden ad-soyad, e-posta ve mesaj bilgileriyle talepler iletilerebilmektedir.



Şekil 3.4. Villa Ana Sayfa



**Şekil 3.5.** Mevcut Kullanıcıların Oturum Oluşturabileceği Form Ekranı



**Şekil 3.6.** Yeni Kullanıcıların Hesap Oluşturabileceği Form Ekranı

## VILLA

Ana Sayfa

Mülklerimiz

İletişim

Giriş Yap

Ziyaret Planlayın



### Deniz Manzalı En Güzel Villa

#### Ödeme İşlemleri nasıl yapılmaktadır?

İster kredi isterse nakit veya senet şeklinde ödeme sistemimiz mevcuttur.

#### Ne zaman mülkleri görebiliriz?

Günün her anında bizi arayıp randevu oluşturabilirsiniz.

#### Evlerin güvenlik sistemi var mı?

Evet bütün evlerimiz 7/24 kameralı ile izlenmektedir.

300 m<sup>2</sup>  
Toplam Alan

Kontrat Türü  
Kontrata Uygun

Ödeme Türü  
Kredi veya Nakit

Güvenlik Durumu  
7/24 Kontrol Altında

Şekil 3.7. Seçilen Villaya Ait Sık Sorulan Soruların Yanıtlandığı Bilgi Bölümü

## VILLA

Ana Sayfa

Mülklerimiz

İletişim

Giriş Yap

Ziyaret Planlayın

34 Tamamlanan Villalar

10 Yıllık Tecrübe

24 Ödül Kazandık

| EN İYİ TEKLİFLERİMİZ

### İstedığınız Türde Villalarımızı

Şekil 3.8. Firma Tecrubesini ve Başarılarını Yansıtan İstatistik Kartları

**VILLA**

## Villa'lарımızı İnceleyin.

Metrekare Bilgisi	<b>200 m<sup>2</sup></b>
Bulunduğu Kat	<b>3</b>
Oda Sayısı	<b>4</b>
Park Alanı	<b>Var</b>
Ödeme Türü	<b>Kredi</b>



**Daire Villa'lарımız hakkında bilgiler**

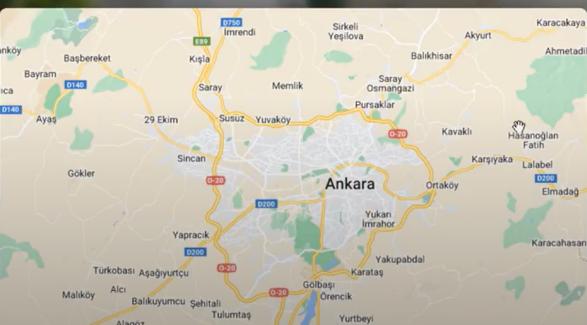
Apartman dairesi içerisinde villa sahibi olmak istereniz bizimle iletişime geçebilirsiniz.

**Ziyaret Planlayın**

Şekil 3.9. Kullanıcının Villa Türleri Arasında Seçim Yapabileceği Filtreleme Arayüzü

**VILLA**

## Danışmanlarımıza İletişime Geçebilirsiniz



Ad Soyad  
Adınız Soyadınız...

Email  
E-mail Adresiniz...

Konu  
Konu...

Mesaj

Şekil 3.10. Kullanıcının Danışmanlarla İletişim Kurabileceği Form ve Firmanın Konumunu Gösteren Harita Entegrasyonu

```
[HttpPost]
0 references
public async Task<IActionResult> Register(RegisterDto registerDto)
{
    var user = new AppUser
    {
        NameSurname = registerDto.NameSurname,
        Email = registerDto.Email,
        UserName = registerDto.UserName
    };

    var result = await _userManager.CreateAsync(user, registerDto.Password);

    if (!result.Succeeded)
    {
        foreach (var item in result.Errors)
        {
            ModelState.AddModelError("", item.Description);
        }

        return View();
    }

    return RedirectToAction("Login");
}

0 references
public IActionResult Login()
{
    return View();
}

[HttpPost]
0 references
public async Task<IActionResult> Login(LoginDto loginDto)
{
    var user = await _userManager.FindByNameAsync(loginDto.UserName);
    if (user == null)
    {
        ModelState.AddModelError("", "Kullanıcı adı veya şifre Hatalı");
        return View();
    }

    var result = await _signInManager.CheckPasswordSignInAsync(user, loginDto.Password, false);
    if (!result.Succeeded)
    {
        ModelState.AddModelError("", "Kullanıcı adı veya şifre Hatalı");
        return View();
    }

    return RedirectToAction("Index", "Banner");
}
```

Şekil 3.11. Login Doğrulama Kodları, AccountController.cs

**Admin paneli**, web sitesinin içerik ve kullanıcı yönetimi işlemlerinin pratik bir şekilde yapılabilmesi için sade ve kullanıcı dostu bir arayüz ile tasarlanmıştır. Panelde sol tarafta yer alan sabit menü, yöneticinin sitenin tüm yönetim alanlarına hızlıca erişebilmesini sağlamaktadır.

Gönderilen örnek ekran görselinde, “Banner Listesi” bölümü yer almaktadır. Bu sayfa sayesinde yönetici kullanıcılar, sitede farklı sayfalarda kullanılmak üzere tanımlanmış banner içeriklerini (şehir ve başlık bilgileriyle) görüntüleyebilmekte, yeni banner ekleyebilmekte veya mevcut banner’ları “Sil” ya da “Güncelle” seçenekleri ile düzenleyebilmektedir.

- Tablo yapısı sayesinde banner’lar kolayca listelenmektedir.
- “Yeni Banner Ekle” butonu ile sisteme dinamik olarak yeni içerikler dahil edilebilmektedir.
- Sil ve Güncelle butonları, içeriklerin yönetimini hızlı ve doğrudan gerçekleştirmeye imkanı sunmaktadır.

Tasarımda modern bir admin panel şablonu tercih edilmiş, minimal görsel öğelerle kullanıcı arayüzünün sade ve anlaşılır olması hedeflenmiştir. Panel, yöneticilerin villa ekleme, mesajları yönetme, soru-cevap içerikleri oluşturma, videoları kontrol etme gibi çeşitli modüllere erişimini kolaylaştıracak şekilde bölmelere ayrılmıştır.

Bu yapı, yöneticilerin sistem üzerinde tam yetkiyle ve güvenli biçimde işlem yapabilmesini sağlamaktadır.

The screenshot shows the 'Banner Listesi' (Banner List) page within the Admin Panel. The left sidebar has a purple header 'ADMIN PANELI' and a 'Banner' menu item selected. The main content area has a search bar at the top. Below it, the title 'Banner Listesi' is displayed with the sub-instruction 'Banner listenizi aşağıdadır.' A table lists three banners:

#	Şehir	Başlık	İşlemler
1	Istanbul	Villa	<a href="#">Sil</a> <a href="#">Güncelle</a>
2	azra	azra	<a href="#">Sil</a> <a href="#">Güncelle</a>
3	b2	b2	<a href="#">Sil</a> <a href="#">Güncelle</a>

A 'Yeni Banner Ekle' button is located at the bottom left of the table. The top right corner shows 'Layout > Blank'. The footer contains the copyright notice 'Copyright © Designed & Developed by Quixkit 2019'.

**Şekil 3.12.** Admin Paneli Ekranı

**Banner Listesi**

Banner listeniz aşağıdadır.

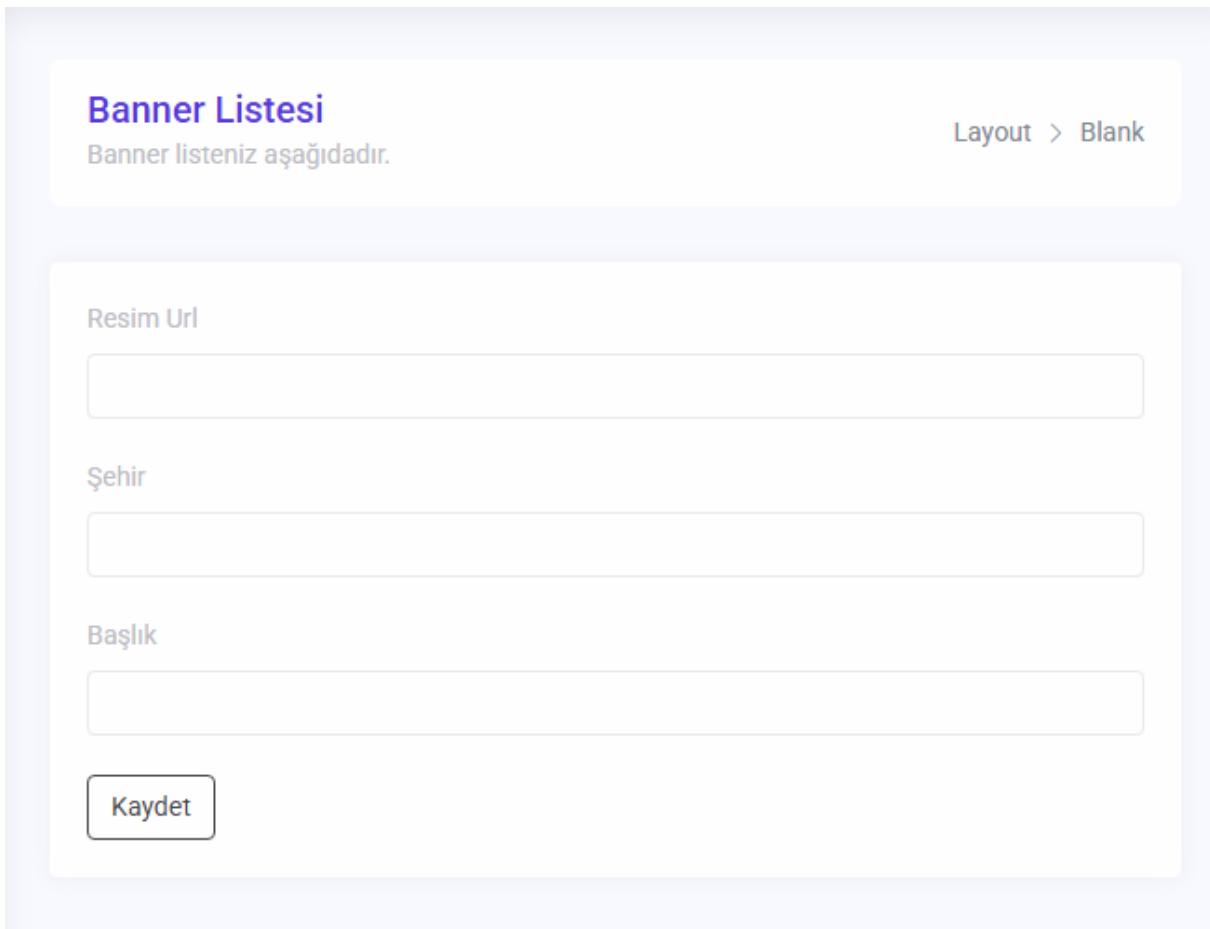
Layout > Blank

Resim Url

Şehir

Başlık

Kaydet



**Şekil 3.13.** Yeni Banner Ekleme Ekranı

**Banner Listesi**

Banner listeniz aşağıdadır.

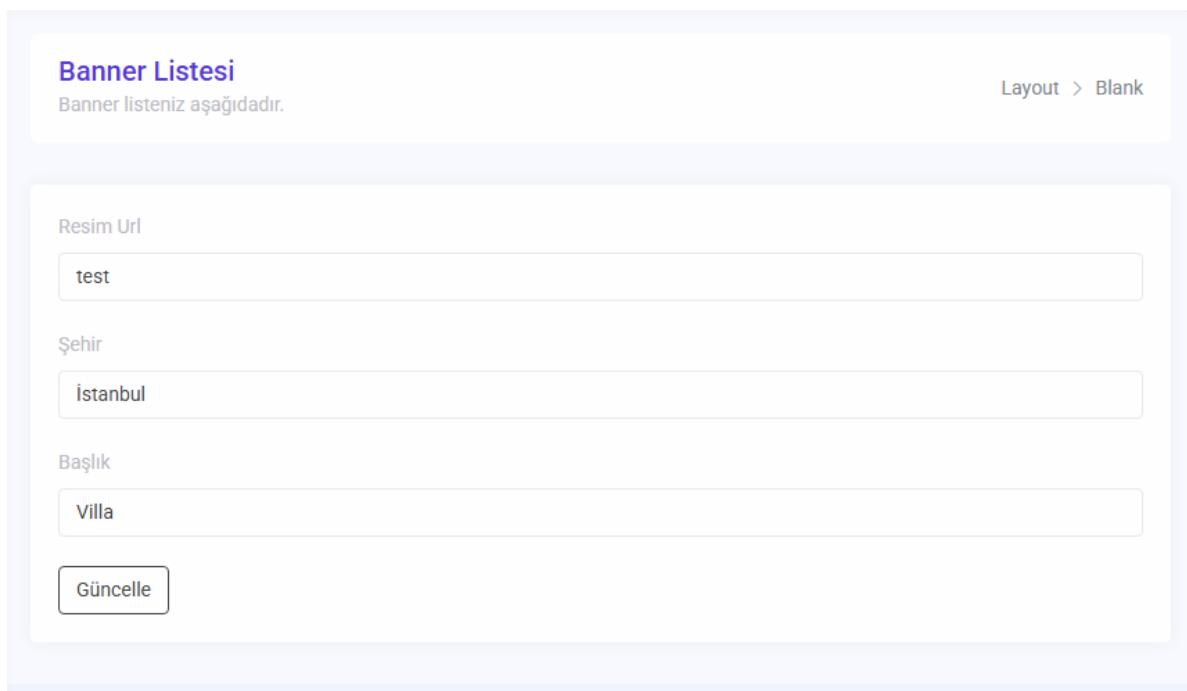
Layout > Blank

Resim Url

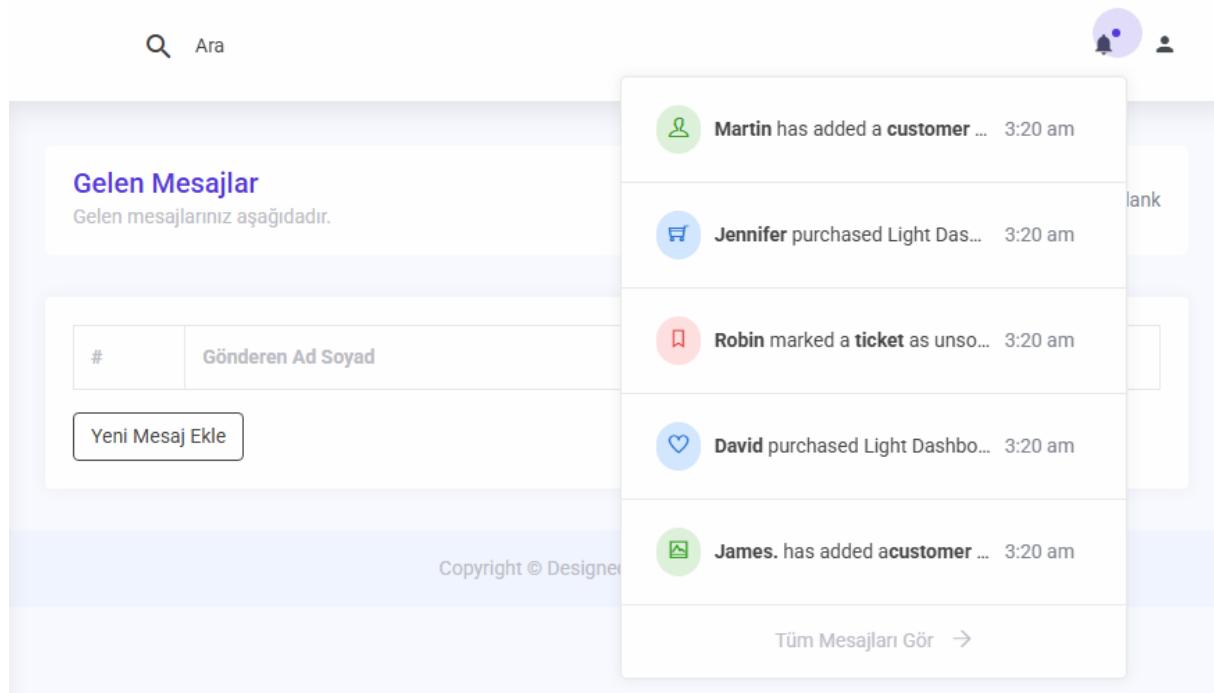
Şehir

Başlık

Güncelle



**Şekil 3.14.** Banner Güncelleme Ekranı



**Şekil 3.15.** Gelen Mesajlar Ekranı ve Bildirim Kutusu

## 4. SONUÇ

Bu projede, C# programlama dili ve **ASP.NET Core** web geliştirme çatısı kullanılarak bir **Villa Kiralama Web Sitesi** geliştirilmiştir. Uygulama; kullanıcıların villa

listelerini görüntüleyebildiği, detaylı bilgi alabildiği, tarih seçenek rezervasyon oluşturabildiği ve yönetici paneli üzerinden içeriklerin kolayca yönetilebilediği işlevsel bir platform sunmaktadır.

Kullanıcılar, siteye kayıt olduktan sonra giriş yapabilir, diledikleri villalar hakkında bilgi alabilir ve kiralama işlemlerini gerçekleştirebilir. Yönetici (admin) tarafı ise kullanıcıları, rezervasyonları ve villa içeriklerini takip edip düzenleme imkanı sağlar.

Projede kullanılan MongoDB veritabanı ile esnek ve ölçeklenebilir veri yönetimi sağlanmış, modern web geliştirme ihtiyaçlarına uygun bir yapı oluşturulmuştur. **Visual Studio** IDE'si ile proje yönetimi ve geliştirme süreci verimli bir şekilde yürütülmüştür.

Kullanıcı arayüzü HTML, CSS ve JavaScript teknolojileriyle sade ve kullanıcı dostu bir tasarıma sahip olacak şekilde hazırlanmıştır. Mobil uyumlu (responsive) tasarım ilkeleri göz önünde bulundurularak kullanıcı deneyimi iyileştirilmiştir.

Bu proje, modern yazılım geliştirme yaklaşımlarını yansıtan, işlevsel ve yönetilebilir bir villa kiralama platformu oluşturmak adına başarılı bir örnek teşkil etmektedir.

## 5. EK

Projeye ait kodlar GitHub üzerinde bir repoda public olarak tutulmaktadır. Proje linki kaynaklar bölümünde verilmiştir.

## **6. KAYNAKÇA**

- [1] <https://github.com/azraoykulukan/Villa>
- [2] [https://www.youtube.com/playlist?list=PLdmlz6\\_BFbF4764pH35xVBT2R9ieYiX\\_A](https://www.youtube.com/playlist?list=PLdmlz6_BFbF4764pH35xVBT2R9ieYiX_A)
- [3] <https://chatgpt.com/>