



**T.C.  
SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ  
WEB PROGRAMLAMA**

**SPOR SALONU YÖNETİM VE RANDEVU SİSTEMİ**

**B221210573 - Khaled Abdullatif**

**Ders Grubu: 2. Öğretim A**

**GitHub Bağlantısı: [https://github.com/Khaledb2212/Web\\_Programlama\\_2025](https://github.com/Khaledb2212/Web_Programlama_2025)**

**SAKARYA**

**Aralık, 2025**

**WEB PROGRAMLAMA Dersi**

# SPOR SALONU YÖNETİM VE RANDEVU SİSTEMİ

---

## Özet

Web tabanlı uygulamalar, gerçek hayatı işletmelerin hizmetlerini dijital ortamda yönetebilmesi için kritik öneme sahiptir. Özellikle spor salonları gibi üyelik, hizmet, eğitmen ve randevu süreçlerinin yoğun olduğu işletmelerde, merkezi bir yönetim sistemi hem operasyonel verimliliği artırmakta hem de kullanıcı deneyimini iyileştirmektedir.

Bu proje kapsamında, Web Programlama dersinde öğrenilen kayıtlar kullanılarak **ASP.NET Core MVC** ile bir **Spor Salonu (Fitness Center) Yönetim ve Randevu Sistemi** geliştirilmiştir. Proje, bir taraftan randevu ve hizmet yönetimini sağlarken diğer taraftan yenilikçi bir özellik olarak **yapay zekâ destekli kişisel egzersiz/diyet önerisi** sunmaktadır.

---

## SİSTEMİN AMAÇ VE KAPSAMI

Projenin temel hedefleri aşağıdaki şekilde özetlenebilir:

- Spor salonuna ait hizmetlerin (fitness, yoga vb.) tanımlanması ve yönetilmesi
- Antrenörlerin sisteme eklenmesi, uzmanlık alanlarının belirlenmesi ve müsaitlik saatlerinin tutulması
- Üyelerin uygun hizmet ve antrenöre göre randevu alabilmesi
- Randevu çakışmalarının kontrol edilmesi ve uygun değilse kullanıcıya uyarı verilmesi
- Randevuların detaylarının (hizmet, süre, ücret, eğitmen) saklanması ve süreçlerin yönetilmesi
- Projede REST API kullanımı ile veritabanı işlemlerinin gerçekleştirilmesi
- Yapay zekâ entegrasyonu ile üyeye kişiselleştirilmiş öneriler sunulması

## TEKNOLOJİLER VE GELİŞTİRME ORTAMI

Projede aşağıdaki teknolojiler kullanılmıştır:

- **ASP.NET Core MVC** (istemci/arayüz katmanı)
- **ASP.NET Core Web API** (REST servis katmanı)
- **C#**
- **SQL Server** (veritabanı)
- **Entity Framework Core** (ORM)
- **LINQ** (sorgulama ve filtreleme)
- **ASP.NET Core Identity** (kullanıcı/rol yönetimi)
- **Bootstrap 5, HTML5, CSS3, JavaScript, jQuery** (arayüz ve etkileşim)

Yapay zekâ kısmında:

- Görsel üretimi için **Banana** platformu kullanılmıştır. Banana, AI modellerinin çalıştırılabilıldığı bir **ücretli (paid) üçüncü taraf servis** olup, bu projede “gelecekteki görünüm” görselinin üretilmesi amacıyla API üzerinden entegre edilmiştir.

## SİSTEM MİMARİSİ

Proje, katmanlı yapıyı destekleyecek şekilde iki ana bileşen olarak tasarlanmıştır:

### 4.1 Web\_API (Servis Katmanı)

Veritabanı ile iletişim yalnızca **Web\_API** üzerinden sağlanmaktadır. Bu yaklaşım sayesinde:

- Veritabanı erişimi tek bir katmanda kontrol altına alınmıştır
- Güvenlik ve yetkilendirme daha yönetilebilir hale gelmiştir
- Modülerlik artmış, istemci (MVC) tarafı veritabanından bağımsız hale gelmiştir

API içinde:

- CRUD işlemleri REST prensiplerine uygun tasarlanmıştır
- LINQ sorguları ile filtreleme yapan üç noktalar sunulmuştur (ör. antrenör listeleme, uygunluk kontrolü, randevu listeleme)

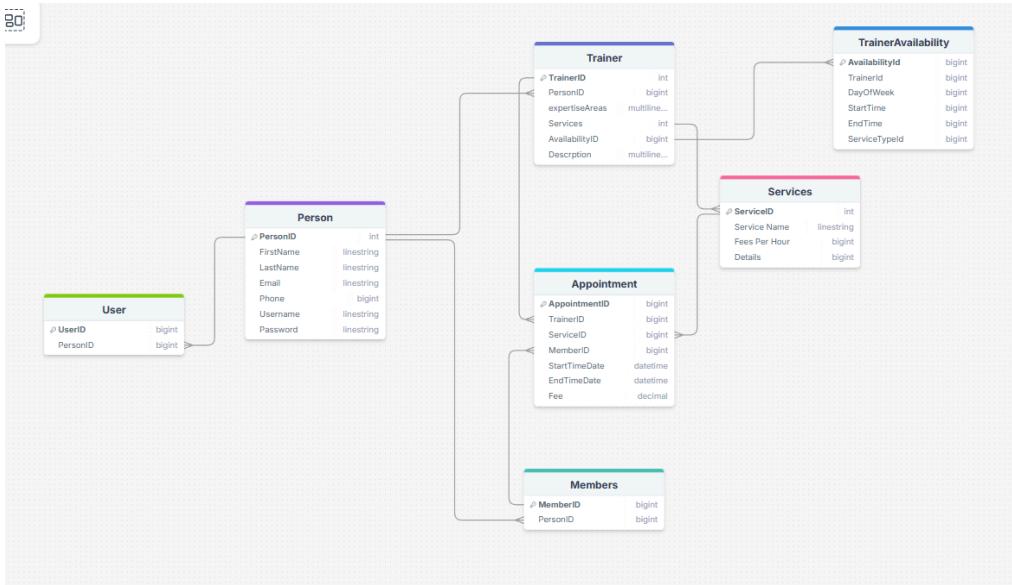
## **4.2 Web\_Project (MVC – İstemci Katmanı)**

Kullanıcı arayüzü temsil eden MVC projesi:

- API üç noktalarına HTTP üzerinden istek atarak verileri alır
- Kullanıcıya uygun sayfaları rol bazlı yetkilendirme ile sunar
- Bootstrap tabanlı arayüz ile kullanıcı deneyimini güçlendirir

---

## **5. VERİTABANI TASARIMI**



Veritabanı olarak **SQL Server** kullanılmıştır. Entity Framework Core ile tablolar oluşturulmuş ve migration yapısı ile geliştirme süreci yönetilmiştir.

Temel tablolar:

- **Members**: Üye bilgileri
- **Trainers**: Antrenör bilgileri
- **TrainerAvailabilities**: Antrenör müsaitlik saatleri
- **Services**: Spor salonu hizmetleri (tür, süre, ücret vb.)
- **Appointments**: Üye randevuları (hizmet, süre, ücret, tarih/saat, eğitmen vb.)

Ayrıca **Identity** tabloları ile kullanıcı giriş, rol ve yetkilendirme süreçleri yönetilmiştir:

- AspNetUsers, AspNetRoles, AspNetUserRoles vb.

# KULLANICI YÖNETİMİ VE ROL BAZLI YETKILENDİRME

Sistemde rol bazlı authorization, hem arayüz (UI) hem de erişim (endpoint/page) seviyesinde uygulanmıştır. Bu sayede kullanıcı yalnızca yetkili olduğu içeriklere erişebilir.

Rol senaryoları:

- **Üye (Member):**

- Randevu alma (Book)
- Kendi randevularını görüntüleme (MyAppointments)
- Antrenör/hizmet bilgilerini görüntüleme

- **Antrenör (Trainer):**

- Sadece Trainer Dashboard erişimi
- Kendi randevularını ve uygunluklarını yönetme

- **Admin:**

- Sadece Admin panel erişimi
- Sistem genelinde yönetim ve CRUD işlemleri

Bu yapı ile:

- Yetkisi olmayan kullanıcılar menüde ilgili alanları göremez

- Doğrudan URL ile erişmeye çalışsa bile sistem erişimi engeller

## **REST API ve LINQ KULLANIMI**

Proje gereksinimlerine uygun şekilde veritabanına erişim REST API üzerinden sağlanmıştır. API içinde LINQ kullanılarak filtreleme yapılan en az bir modül bulunmaktadır.

Örnek senaryolar:

- Tüm antrenörleri listeleme
- Belirli tarih/saat aralığında uygun antrenörleri getirme
- Üyenin randevularını listeleme
- Hizmet türüne göre antrenör filtreleme

Bu kısım, ders kapsamında istenen “API + LINQ ile filtreleme” gereksinimini karşılamaktadır.

## **YAPAY ZEKÂ ENTEGRASYONU (AI MODÜLÜ)**

Projede yenilikçi bir özellik olarak, kullanıcıya kişiselleştirilmiş öneriler sunan bir AI modülü geliştirilmiştir.

Kullanıcı şu seçeneklerle sisteme bilgi sağlayabilir:

- Sadece fotoğraf yükleme
- Sadece boy/kilo/yaş bilgisi girme
- Fotoğraf + boy/kilo/yaş bilgilerini birlikte girme

Sistem, bu girdilere göre:

- Kişiye özel egzersiz ve diyet planı üretir
- Daha sağlıklı yaşam için öneriler sunar
- Planın uygulanması durumunda “gelecekteki görünümü” temsil eden bir görsel üretir

Görsel üretim sürecinde:

- **Banana** platformu kullanılmıştır
- Banana, API üzerinden model çalıştırılmaya imkân veren **ücretli (paid) üçüncü taraf bir servis** olup, sistemde kullanıcıya sunulan “gelecek görünüm görseli” bu servis aracılığıyla üretilmektedir

Bu entegrasyon, projeyi yalnızca CRUD tabanlı klasik bir sistem olmaktan çıkarıp daha güncel, teknolojik ve kullanıcı odaklı bir yapıya taşımaktadır.