

VERİ YAPILARI 2. ÖDEV RAPORU

Adı Soyadı: Ahmet Taşdemir

Öğrenci No: G231210068

Ders Grubu: 2-B

Tarih: 20Aralık 2025

GitHub: <https://github.com/Ahmettasdemirr/web-proje>

Giriş:

Bu proje, Web Programlama kapsamında, teorik ve pratik bilgilerin gerçek bir probleme uygulanarak çözüm üretmesini hedeflemektedir. Projenin temel konusu, bir Spor Salonu (Fitness Center) Yönetim ve Randevu Sistemi geliştirmektir. Geliştirilen sistem, spor salonunun sunduğu hizmetleri, antrenörlerin uzmanlık alanlarını ve üyelerin randevularını yönetebilecek nitelikte olup, Web Programlama dersinin önemli bir gereksinimi olan **REST API** kullanımını içermektedir. Projenin ana odak noktası, bir müşteri veya yönetici tarafından sorgulanan belirli bir zaman aralığında müsait olan eğitmenleri listelemektir. Geliştirme süreci, güncel teknolojiler olan ASP.NET Core MVC, C#, Entity Framework Core ve SQL Server kullanılarak gerçekleştirilmiştir.

Veritabanı Modeli ve Varlık İlişkileri

Sistemin veri tabanı yapısı, projenin gereksinimlerini karşılamak üzere dört ana varlık üzerine kurulmuştur. **Trainers** tablosu, salonlarda görev yapan antrenörlerin temel bilgilerini tutarken, **Services** tablosu ise sunulan hizmet türlerini (fitness, yoga, pilates vb.) tanımlar. **Appointments** tablosu, eğitmenlerin önceden alınmış randevu kayıtlarını ve bu randevuların başlangıç/bittiş saatlerini saklayarak müsaitlik hesaplamasının temelini oluşturur. Son olarak, **TrainerServices** tablosu, bir eğitmenin birden fazla hizmette uzman olabileceği (Çoka-Çok) ilişkisini yöneten bir ara tablodur. Veri modeli bu şekilde tasarılanarak, Antrenör Yönetimi gereksinimine uygun olarak eğitmenlerin uzmanlık alanları sisteme entegre edilmiştir.

Uygulama Mimarisi ve İş Akışı

Uygulama, veri erişimini ve iş mantığını merkezi olarak yöneten bir API katmanı ve kullanıcı etkileşimini sağlayan bir MVC katmanı olmak üzere ayrılmıştır. Bu ayrim, projenin REST API kullanımı gereksinimini yerine getirmektedir.

Müsaitlik Sorgusu (REST API Kullanımı)

Sistemin en kritik işlevi, TrainersApiController.cs içinde yer alan GetAvailableTrainers metodu ile gerçekleştirilir. Bu metot, sorgulanın tarih, saat ve süre parametrelerini alarak Veritabanında uygun antrenörleri getirme işlemini gerçekleştirir. Metot içerisinde öncelikle sorgulanın randevunun başlangıç ve bitiş zamanları kesin olarak hesaplanır. Ardından, tüm eğitmenler ve randevu detayları Entity Framework Core kullanılarak veritabanından çekilir. Çekilen veri üzerinde uygulanan ana LINQfiltresi, bir eğitmenin Appointments (Randevular) koleksiyonunda sorgulanın zaman dilimi ile herhangi bir çakışma olup olmadığını kontrol eder. Çakışma bulunmayan eğitmenler müsait kabul edilir ve TrainerApiResult modeline dönüştürülerek JSON formatında döndürülür. Bu iş akışı sayesinde, randevu saati uygun değilse sistem kullanıcıyı uyarma gereksinimini dolaylı olarak karşılamaktadır

MVC ve Hata Yönetimi

ReportController.cs, kullanıcı arayüzü ile API arasında köprü görevi görür. Controller, HttpClientFactory kullanarak API'ye isteği gönderir ve yanıtı işler. Proje, sağlam bir hata yönetimi sunar. API'den beklenmeyen bir durum kodu (404 Not Found veya 500 Internal Server Error) alındığında, Controller yanıt kodunu ve API'den gelen detaylı hata metnini yakalayarak arayüze aktarır. Örneğin, başlangıçta serviceId=1 parametresinin veritabanında karşılığının bulunmaması nedeniyle API'den dönen 404 hatası bu mekanizma sayesinde net bir şekilde kullanıcıya sunulabilmiştir.

Geliştirme Sürecindeki Çözülen Kritik Sorunlar

Proje, geliştirme sırasında iş mantığı ve teknik uygulamadan kaynaklanan iki önemli sorunla karşılaşmış ve bu sorunlar başarılı bir şekilde çözülmüştür.

- Hizmet ID Uyumsuzluğu Sorunu:** Projenin erken aşamasında, Controller'da varsayılan olarak kullanılan serviceId=1 değerinin veritabanında karşılığı olmadığı için tüm sorgular NotFound (404) hatasıyla başarısız olmuştur. Bu sorun, varsayılan Service ID değerinin veritabanında kesinlikle mevcut olan en küçük Service ID (örneğin 3) olarak güncellenmesiyle çözülmüştür.
- Yanlış İş Mantığı ve Projeksiyon Hatası:** Başlangıçta API, sadece sorgulanan hizmeti verebilecek eğitmenleri getiriyordu. Ancak talebin, *uzmanlık ayrimi yapmaksızın* sadece **zamanı müsait olan tüm eğitmenleri** listeleye yönünde değişmesi üzerine, API içerisindeki uzmanlık filtreleme (.Where(t => t.TrainerServices.Any(...))) kodu kaldırılmıştır. Bu değişiklik sonrasında, eğitmenlerin uzmanlık alanlarını listeleyen projeksiyon (string.Join kullanımı) Entity

Framework'ün SQL'e çeviremediği **CS8072** hatasına neden olmuştur. Bu teknik sorun, sorgunun `ToList()` ile erken bir aşamada C# belleğine çekilmesi ve `string.Join` gibi karmaşık işlemlerin bellek üzerinde güvenle çalıştırılmasıyla kesin olarak giderilmiştir.

Sonuç ve Proje Teslimi

Geliştirilen sistem, Spor Salonu Yönetim Sistemi projesinin Raporlama (Müsaitlik Sorgusu) ve REST API Kullanımı gereksinimlerini eksiksiz yerine getirmiştir. Proje, hem frontend hem de backend veri doğrulama prensiplerine uygun olarak geliştirilmiştir. Projenin nihai teslimi, 20.12.2025 tarihinde üzerinden gerçekleştirilecek olup, raporla birlikte tüm kaynak kodları, öğrenci numarası ve gibi kapak bilgileri eksiksiz sunulacaktır. Proje tanıtımı, veritabanı modeli ve bu rapor, projenin tüm kapsamını detaylandırmaktadır.

