Kütüphane Yönetim Sistemi – Proje Dokümantasyonu & README

# 1. Proje Hakkında

Bu proje, ASP.NET Core MVC kullanılarak geliştirilmiş bir Kütüphane Yönetim Sistemi’dir. Proje ile kullanıcılar kitap ve öğrenci kayıtlarını yönetebilir, kitap ödünç alma işlemlerini takip edebilir. Projede temel olarak CRUD işlemleri, Layout & PartialView kullanımı, statik dosyalar, routing ve MVC servisleri uygulanmıştır.

# 2. Kullanılan Teknolojiler ve Özellikler

- C# / ASP.NET Core MVC

- Visual Studio

- Entity Framework Core (opsiyonel)

- Razor View Engine

- wwwroot klasörü ile statik dosya yönetimi

- Routing Konfigürasyonu

- Layout & PartialView

# 3. Proje Yapısı

KütüphaneYonetim/  
│  
├── Controllers/  
│ ├── HomeController.cs  
│ ├── BooksController.cs  
│ └── StudentsController.cs  
│  
├── Models/  
│ ├── Book.cs  
│ └── Student.cs  
│  
├── Views/  
│ ├── Shared/  
│ │ ├── \_Layout.cshtml  
│ │ └── \_Footer.cshtml  
│ ├── Home/  
│ │ ├── Index.cshtml  
│ │ └── About.cshtml  
│ ├── Books/  
│ │ ├── Create.cshtml  
│ │ └── List.cshtml  
│ └── Students/  
│ ├── Create.cshtml  
│ └── List.cshtml  
│  
├── wwwroot/  
│ ├── css/  
│ ├── js/  
│ └── images/  
│  
├── appsettings.json  
└── Program.cs

# 4. Program.cs Konfigürasyonu

```csharp  
using Microsoft.AspNetCore.Builder;  
using Microsoft.Extensions.DependencyInjection;  
  
var builder = WebApplication.CreateBuilder(args);  
  
// MVC Servislerini ekleme  
builder.Services.AddControllersWithViews();  
  
var app = builder.Build();  
  
// wwwroot klasörü için statik dosya kullanımı  
app.UseStaticFiles();  
  
// Routing konfigürasyonu  
app.UseRouting();  
  
// Varsayılan routing  
app.MapControllerRoute(  
 name: "default",  
 pattern: "{controller=Home}/{action=Index}/{id?}"  
);  
  
app.Run();  
```

Açıklama:  
- AddControllersWithViews(): MVC servislerini aktif eder.  
- UseStaticFiles(): wwwroot içindeki CSS, JS, resim gibi dosyaların kullanılmasını sağlar.  
- MapControllerRoute(): İstekleri doğru controller ve aksiyon metoduna yönlendirir.

# 5. Layout & PartialView Kullanımı

- \_Layout.cshtml: Tüm sayfaların ortak yapısı  
- \_Footer.cshtml: Sabitlenmiş footer

<footer>  
 <p>&copy; 2025 Tüm Hakları Saklıdır - [Adınız Soyadınız]</p>  
</footer>

# 6. Sayfalar

- Ana Sayfa (Home/Index): Giriş sayfası, hoş geldiniz mesajı  
- Hakkında (Home/About): Proje hakkında kısa bilgi  
- Öğrenci & Kitap İşlemleri: CRUD işlemleri

# 7. Kullanılan Özelliklerin Açıklamaları

| Özellik | Açıklama |  
|---------|----------|  
| MVC Servisleri | Kontroller, modeller ve görünümler arası iletişimi sağlar |  
| Routing | URL yönlendirmelerini yönetir |  
| Layout & PartialView | Ortak görünüm sağlar |  
| wwwroot | CSS, JS, resim gibi statik dosyaları sunar |  
| CRUD İşlemleri | Kitap ve öğrenci ekleme, silme, listeleme |

# 8. Ekran Görüntüleri

1. Ana Sayfa: [Buraya ekran görüntüsü ekleyin]  
2. Hakkında Sayfası: [Buraya ekran görüntüsü ekleyin]  
3. Kitap Listeleme: [Buraya ekran görüntüsü ekleyin]  
4. Öğrenci Ekleme: [Buraya ekran görüntüsü ekleyin]

# 9. Kurulum ve Çalıştırma

1. Projeyi Visual Studio’da açın  
2. Program.cs dosyasındaki yapılandırmaları kontrol edin  
3. Gerekli NuGet paketlerini yükleyin  
4. Uygulamayı çalıştırın (Ctrl + F5)  
5. Tarayıcıda https://localhost:5001/ ile ana sayfaya ulaşın

# 10. Testler

- Sayfa yönlendirmeleri doğru çalışıyor mu?  
- CRUD işlemleri sorunsuz mu?  
- Layout ve footer doğru render ediliyor mu?