

Proje konusu: Bir öğrenci kayıt uygulaması oluşturarak veritabanı CRUD işlemlerini(Ekleme,Silme,Güncelleme,Listeleme) gerçekleştirmek.

Proje için kullanılacak teknolojiler:

- .NET Core MVC teknolojisi ile katmanlı uygulama tasarımı
 - Model → Veri modellemesi
 - View → Arayüz tasarımları
 - Controller → Arayüz ve veritabanı arasındaki işleyiş
- Sqlite Veritabanı(Sunucu gerektirmez,giriş seviyesi için ideal)

ASP.NET MVC → Windows'a özel eski bir teknoloji. (sadece Windows Server'larda çalışıyor.

ASP.NET Core MVC → Windows + Linux + Mac her yerde çalışabilen, **modern** bir teknoloji.

Proje uygulama adımları:

1 Visual Studio'da ASP.NET Core MVC Projesi Aç

- Visual Studio'yu aç.
- **Create a new project** → seç.
- **ASP.NET Core Web App (Model-View-Controller)** seç → İleri.
- Proje adı: **StudentApp**
- Framework: **.NET 8** seç (veya varsa .NET 7, 6 da olur.)
- Authentication: **None** bırak.
- ✓ HTTPS ✓ MVC seçili olsun.

Create de → Proje oluşsun.

2 Entity Framework Core ve SQLite Yükle

Tools > NuGet Package Manager > Manage NuGet Packages for Solution aç.

Browse sekmesine gel, şunları sırayla kur:

- `Microsoft.EntityFrameworkCore`
- `Microsoft.EntityFrameworkCore.Sqlite`
- `Microsoft.EntityFrameworkCore.Tools`

✅ Bunlar kurulduysa her şey tamam!

3 Model Oluştur (**Models/Student.cs**)

Proje içinde **Models** klasörüne sağ tık → Add → Class → **Student.cs** dosyası oluştur.

```
using System.ComponentModel.DataAnnotations;

namespace StudentApp.Models
{
    public class Student
    {
        public int Id { get; set; }

        [Required]
        public string Name { get; set; }

        [Required]
        public string Surname { get; set; }
    }
}
```

4 DbContext Oluştur (**Data/AppDbContext.cs**)

Yeni bir klasör: **Data** → sağ tık → Add → Class → **AppDbContext.cs** oluştur.

İçine şunu yaz:

```
using Microsoft.EntityFrameworkCore;
using StudentApp.Models;
```

```
namespace StudentApp.Data
{
    public class AppDbContext : DbContext
    {
        public AppDbContext(DbContextOptions<AppDbContext> options) : base(options) { }

        public DbSet<Student> Students { get; set; }
    }
}
```

5 Veritabanı Ayarlarını Yap (Program.cs)

Program.cs dosyasını aç. Şu satırı

`builder.Services.AddControllersWithViews();` altına ekle:

```
builder.Services.AddDbContext<AppDbContext>(options =>
    options.UseSqlite("Data Source=student.db"));
```

6 Controller ve View Oluştur

Controllers klasörüne sağ tık → Add → Controller →

MVC Controller with views, using Entity Framework seç.

- Model class: `Student`
- Data context class: `AppDbContext`
- Controller name: `StudentsController`

Oluştur → Otomatik Controller + View'lar gelecek.

7 Migration Yap ve Veritabanını Oluştur

Visual Studio'da:

- View > Terminal (veya Tools > NuGet Package Manager > Package Manager Console) aç.
- Şunları sırayla yaz:

Add-Migration InitialCreate

Update-Database

✅ Otomatik olarak `student.db` dosyası oluşur!

ASP.NET Core MVC projesi açtığında varsayılan olarak `HomeController` gelir ve `Views/Home/Index.cshtml` dosyası açılış sayfasıdır.

Senin oluşturduğun `StudentsController`'a gitmek için **oraya bir link** ekleyebiliriz.

✅ Yapılacak işlem:

1. `Views/Home/Index.cshtml` dosyasını aç.
2. Aşağıdaki gibi bir `<a>` etiketi ekle:

<pre><p> <a asp-controller="Students" asp-action="Index">Öğrenci Listesine Git </p></pre>

Bu link seni `StudentsController` > `Index` action'ına yönlendirir.



Artık localde çalıştırabilirsin:

F5 → Students Controller'ına git → Yeni öğrenci ekle → Listele!

A. GitHub'a ilk hazırlık

1. GitHub Hesabın yoksa:

- GitHub'a git → <https://github.com>
- Ücretsiz hesap oluştur.

2. Bilgisayarında Git yüklü mü kontrol et:

- Visual Studio yüklüyse büyük ihtimalle Git de yüklüdür.
- Kontrol etmek için `cmd` ya da `PowerShell` aç, şunu yaz:

```
git --version
```

- Eğer sürüm numarası gelirse tamamdır.

B. Visual Studio ile GitHub bağlantısı

1. Visual Studio'da proje açıkken:

- Sol üst menüden **View > Git Changes** panelini aç.
(ya da alt çubuktan `Git Changes` sekmesini tıkla)

2. Sol üstteki “Create Git Repository” seçeneğine tıkla.

3. Açılan pencerede:

- **Name:** Repo adı (örneğin: `StudentProject`)
- **Description:** (isteğe bağlı)
- **GitHub Account:** GitHub hesabını seç veya giriş yap.
- **Private / Public:** Tercihine göre seç.
- Alt kısımdaki `Push to GitHub` kutusunu işaretle.

 Son olarak **Create and Push** butonuna bas.

✓ C. Artık GitHub'da yayında!

- Visual Studio otomatik olarak:
 - `.git` klasörünü oluşturur
 - Tüm dosyaları ekler
 - İlk commit'i yapar
 - GitHub'a gönderir

🎉 GitHub hesabına git → Repositories sekmesine bak.
Projen orada olacak!

🧠 Ekstra: Değişiklik yaptığında ne yapacaksın?

1. Değişiklik yap.
2. Sol alttaki **Git Changes** panelinde:
 - `Message` kısmına kısa bir açıklama yaz (örn. "Ogrenci ekleme eklendi")
 - `Commit All and Push` butonuna tıkla.
3. GitHub'daki repo otomatik güncellenmiş olur.