

Controllers

- HomeController.cs
- StudentsController.cs

Data

- ApplicationDbContext.cs
- Migrations klasörü (EF Core migration dosyaları)

Models

- Student.cs
- Course.cs
- ErrorViewModel.cs
- ViewModels klasörü (ileride ViewModel'ler için)

Repositories

- IStudentRepository.cs
- StudentRepository.cs

Views

- Home (varsayılan)
- Shared (Layout vb.)
- Students
 - Index.cshtml
 - Create.cshtml
 - Detail.cshtml
- _ViewImports.cshtml
- _ViewStart.cshtml

app.db → SQLite veritabanı dosyası

appsettings.Development.json → connection string

Models/Student.cs

```
namespace mvcInterface.Models;

public class Student
{
    public int Id { get; set; }
    public string Name { get; set; } = string.Empty;
    public int Age { get; set; }
}
```

Models/Course.cs

```
namespace mvcInterface.Models;

public class Course
{
    public int Id { get; set; }
    public string Name { get; set; } = string.Empty;
}
```

Data/AppDbContext.cs

```
using Microsoft.EntityFrameworkCore;
using mvcInterface.Models;
namespace mvcInterface.Data;

public class AppDbContext : DbContext
{
    public AppDbContext(DbContextOptions<AppDbContext> options)
        : base(options) { }
    public DbSet<Student> Students => Set<Student>();
    public DbSet<Course> Courses => Set<Course>();
}
```

Repositories/IStudentRepository.cs

```
using mvcInterface.Models;
namespace mvcInterface.Repositories;
public interface IStudentRepository
{
    IEnumerable<Student> GetAll();
    Student? GetById(int id);
    void Add(Student student);
    void Delete(int id);
    void Save();}
}
```

Repositories/StudentRepository.cs

```
using mvcInterface.Data;
using mvcInterface.Models;
using Microsoft.EntityFrameworkCore;
namespace mvcInterface.Repositories;
public class StudentRepository : IStudentRepository
{
    private readonly AppDbContext _context;
    public StudentRepository(AppDbContext context)
    {
        _context = context;
    }
    public IEnumerable<Student> GetAll()
    => _context.Students
        .OrderBy(s => s.Name)
        .ToList();
    public Student? GetById(int id)
    => _context.Students.FirstOrDefault(s => s.Id == id);
    public void Add(Student student)
    {
        _context.Students.Add(student);
    }
    public void Delete(int id)
    {
        var s = _context.Students.Find(id);
        if (s != null)
        {
            _context.Students.Remove(s);
        }
    }
    public void Save()
    {
        _context.SaveChanges();
    }
}
```

appsettings.Development.json (veya appsettings.json)

```
"ConnectionStrings": {  
  "DefaultConnection": "Data Source=app.db"  
}
```

Program.cs

```
using Microsoft.EntityFrameworkCore;  
using mvcInterface.Data;  
using mvcInterface.Repositories;  
var builder = WebApplication.CreateBuilder(args);  
builder.Services.AddControllersWithViews();  
// DbContext kaydı  
builder.Services.AddDbContext<AppDbContext>(options =>  
    options.UseSqlite(builder.Configuration.GetConnectionString("DefaultConnection")));  
// Repository kaydı  
builder.Services.AddScoped<IStudentRepository, StudentRepository>();  
  
var app = builder.Build();  
if (!app.Environment.IsDevelopment()) {  
    app.UseExceptionHandler("/Home/Error");  
    app.UseHsts();  
}  
app.UseHttpsRedirection();  
app.UseStaticFiles();  
app.UseRouting();  
app.UseAuthorization();  
// Varsayılan rota: Students/Index  
app.MapControllerRoute(  
    name: "default",  
    pattern: "{controller=Students}/{action=Index}/{id?}");  
app.Run();
```

dotnet ef migrations add InitialCreate

dotnet ef database update

Controllers/StudentsController.cs

```
using Microsoft.AspNetCore.Mvc;
using mvcInterface.Models;
using mvcInterface.Repositories;
namespace mvcInterface.Controllers;
public class StudentsController : Controller
{
    private readonly IStudentRepository _repo;
    public StudentsController(IStudentRepository repo)
    {
        _repo = repo;
    }
    // Listeleme
    public IActionResult Index()
    {
        var students = _repo.GetAll();
        return View(students);
    }
    // Detay
    public IActionResult Detail(int id)
    {
        var student = _repo.GetById(id);
        if (student == null) return NotFound();
        return View(student);
    }
    // Create GET
    [HttpGet]
    public IActionResult Create()
    {
        return View(new Student());
    }
    // Create POST
    [HttpPost]
    [ValidateAntiForgeryToken]
    public IActionResult Create(Student model)
    {
        if (!ModelState.IsValid)
        {
            return View(model);
        }
        _repo.Add(model);
        _repo.Save();

        return RedirectToAction(nameof(Index));
    }
}
```

```
// Silme (isteğe bağlı)
[HttpPost]
[ValidateAntiForgeryToken]
public IActionResult Delete(int id)
{
    _repo.Delete(id);
    _repo.Save();
    return RedirectToAction(nameof(Index));
}
```

_ViewImports.cshtml

```
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
@using mvcInterface.Models
@using mvcInterface.Models.ViewModels
```

Students/Index.cshtml

```
@model IEnumerable<Student>
@{
    ViewData["Title"] = "Students";
}
<h1>Students</h1>
<p>
    <a asp-action="Create" class="btn btn-primary">Yeni Öğrenci Ekle</a>
</p>
<table class="table table-bordered table-striped">
    <thead>
        <tr>
            <th>Ad</th>
            <th>Yaş</th>
            <th style="width:150px">İşlemler</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var s in Model)
        {
            <tr>
                <td>@s.Name</td>
                <td>@s.Age</td>
                <td>
                    <a asp-action="Detail" asp-route-id="@s.Id" class="btn btn-info btn-sm">Detay</a>
                    <form asp-action="Delete" method="post" class="d-inline">
                        <input type="hidden" name="id" value="@s.Id" />
                        <button type="submit" class="btn btn-danger btn-sm"
                            onclick="return confirm('Silinsin mi?');">Sil</button>
                    </form>
                </td>
            </tr>
        }
    </tbody>
</table>
```

```
</tbody>
</table>
```

Students/Create.cshtml

```
@model Student
@{
    ViewData["Title"] = "Yeni Öğrenci";
}
<h1>Yeni Öğrenci Ekle</h1>
<div asp-validation-summary="ModelOnly" class="text-danger"></div>
<form asp-action="Create" method="post" class="row g-3">
    <div class="col-md-6">
        <label asp-for="Name" class="form-label"></label>
        <input asp-for="Name" class="form-control" />
        <span asp-validation-for="Name" class="text-danger"></span>
    </div>
    <div class="col-md-3">
        <label asp-for="Age" class="form-label"></label>
        <input asp-for="Age" class="form-control" />
        <span asp-validation-for="Age" class="text-danger"></span>
    </div>
    <div class="col-12">
        <button type="submit" class="btn btn-success">Kaydet</button>
        <a asp-action="Index" class="btn btn-secondary">Listeye Dön</a>
    </div>
</form>
@section Scripts {
    <partial name="_ValidationScriptsPartial" />
}
```

Students/Detail.cshtml

```
@model Student
@{
    ViewData["Title"] = "Öğrenci Detay";
}
<h1>Öğrenci Detay</h1>
<dl class="row">
    <dt class="col-sm-2">Ad</dt>
    <dd class="col-sm-10">@Model.Name</dd>
    <dt class="col-sm-2">Yaş</dt>
    <dd class="col-sm-10">@Model.Age</dd>
</dl>
<p>
    <a asp-action="Index" class="btn btn-secondary">Listeye Dön</a>
</p>
```