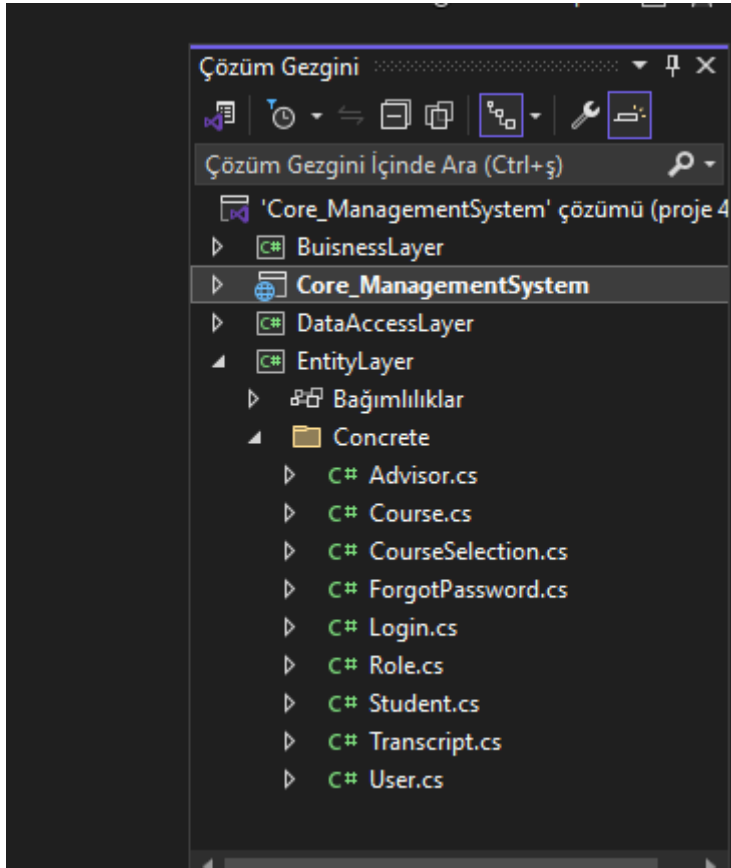
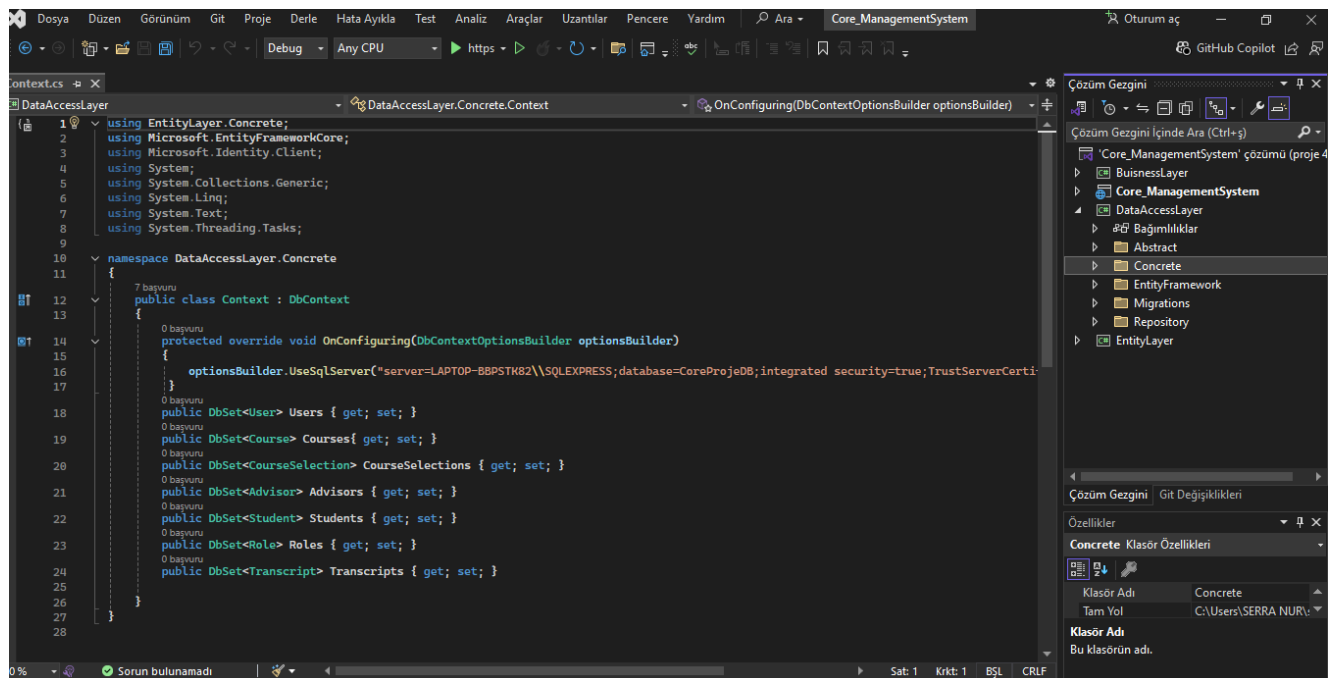


STUDENT MANAGEMENT SYSTEM(Core_ManagementSystem)

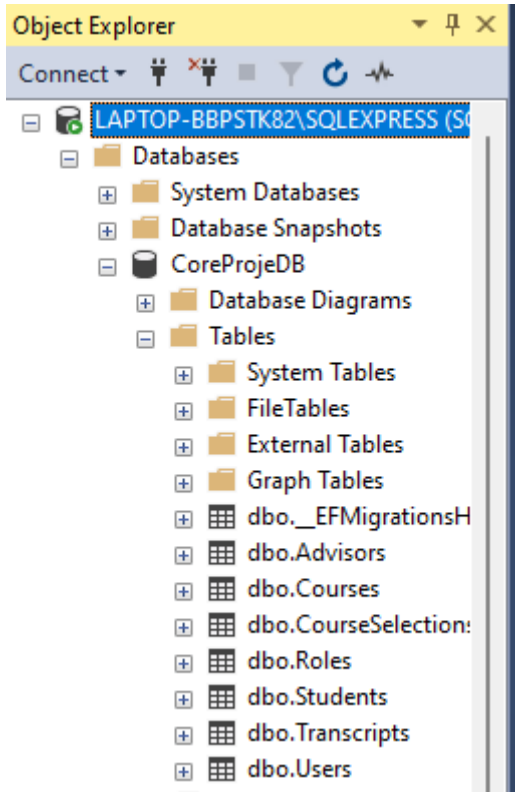
ÇOK KATMANLI MİMARİ YAPISININ OLUŞTURULMASI



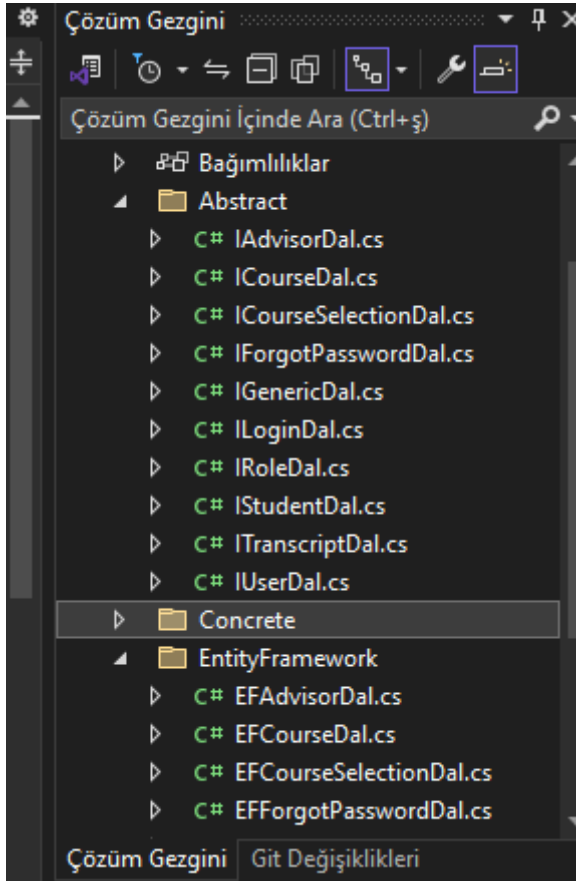
DATA ACCESS LAYER İÇİNDE DbContext Sınıfı VE BAĞLANTI YAPILANDIRILMASI



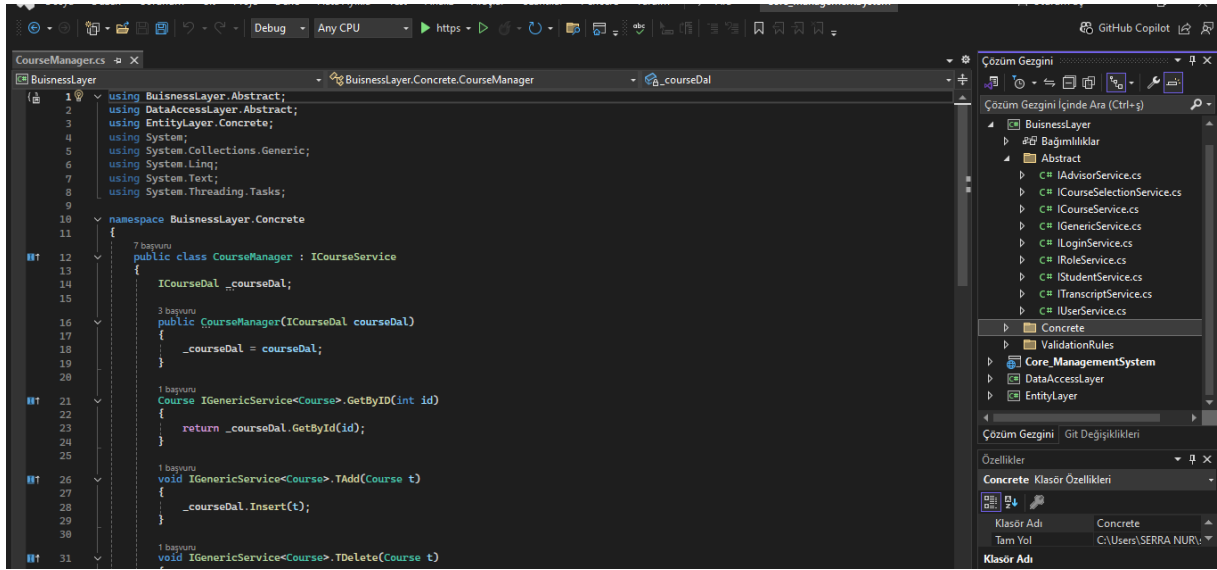
MİGRATIONS İŞLEMİ VE OLUŞTURULAN DATABASE



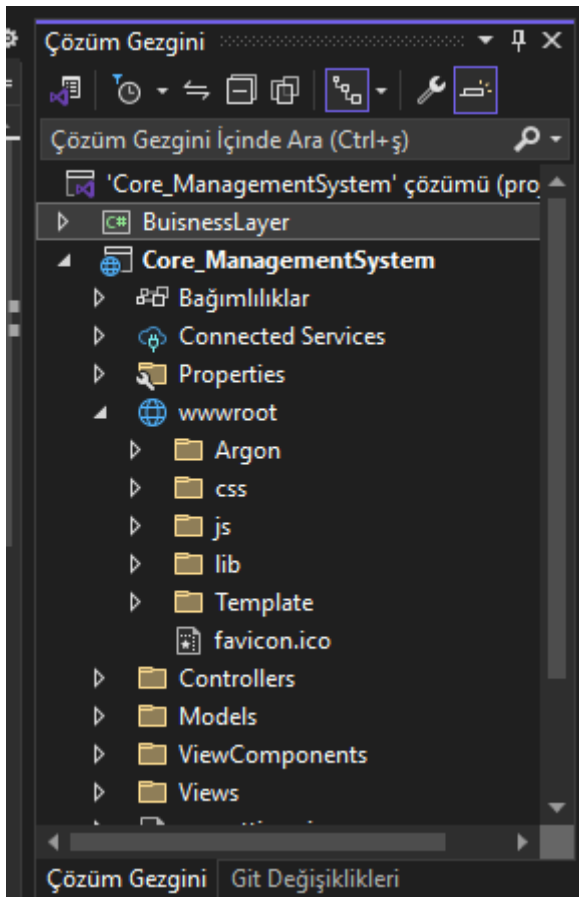
INTERFACE VE EFDAL



BÜİSSSLAYER KATMANI VE GENERİC SERVİSLER, MANAGER SINIFLARI



TEMANIN OLUŞTURULMASI VE PROJEYE DAHİL EDİLMESİ(Template)

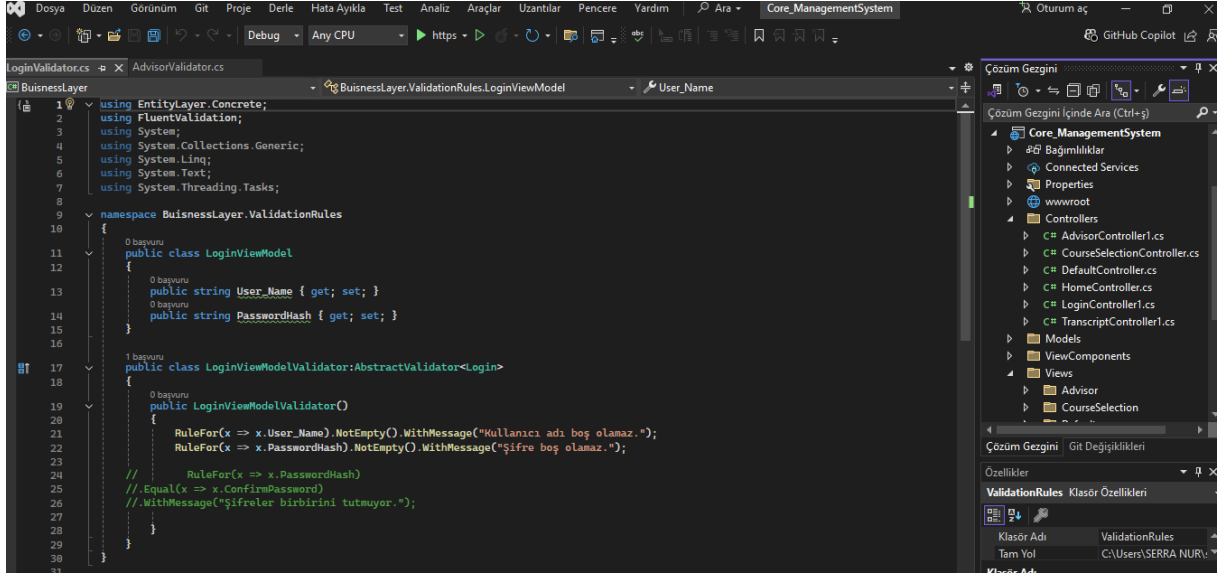


The screenshot displays the Visual Studio IDE with the following components:

- Top Bar:** Shows the file path `Core_ManagementSystem` and the current file `AdvisorList.cs`.
- Code Editor:** Contains the C# code for `AdvisorList` class, which inherits from `ViewComponent`. The code includes using statements for `BuisnessLayer`, `DataAccessLayer`, `Microsoft.AspNetCore.Mvc`, and `Core_ManagementSystem.Models`. It defines a `AdvisorList` class with a `Invoke()` method that populates `advisors`, `students`, and `courses` lists and returns a `View`.
- Solution Explorer:** Located on the right, it shows the project structure. The `Advisor` folder is selected, containing `Advisor`, `CourseSelected`, `Login`, `Transcript`, and `Views` subfolders. The `Views` folder is expanded, showing `Advisor`, `CourseSelection`, `Default`, `Home`, `Login`, `Shared`, and `Transcript` subfolders. The `_ViewImports.cshtml` file is also visible.
- Output Window:** At the bottom right, it shows the `Özellikler` (Properties) window for the `Advisor` class, displaying the `Klasör Adı` (Folder Name) as `Advisor` and the `Tam Yol` (Full Path) as `C:\Users\SERRA NUR\...`.

FLUENT VALIDATION

Controller ve Validation Rules Kısımları



```
1 using EntityLayer.Concrete;
2 using FluentValidation;
3 using System;
4 using System.Collections.Generic;
5 using System.Linq;
6 using System.Text;
7 using System.Threading.Tasks;
8
9 namespace BuisnessLayer.ValidationRules
10 {
11     0 başvuru
12     public class LoginViewModel
13     {
14         0 başvuru
15         public string User_Name { get; set; }
16         0 başvuru
17         public string PasswordHash { get; set; }
18     }
19
20     1 başvuru
21     public class LoginViewModelValidator:AbstractValidator<Login>
22     {
23         0 başvuru
24         public LoginViewModelValidator()
25         {
26             RuleFor(x => x.User_Name).NotEmpty().WithMessage("Kullanıcı adı boş olamaz.");
27             RuleFor(x => x.PasswordHash).NotEmpty().WithMessage("Şifre boş olamaz.");
28             //
29             RuleFor(x => x.PasswordHash)
30             // .Equal(x => x.ConfirmPassword)
31             // .WithMessage("Şifreler birbirini tutmuyor.");
32         }
33     }
34 }
```