

E-Learn

online platform for courses

www.E-Learn.com





Online learning platform



Welcome, everyone!

I'm excited to present our Online Learning Platform, designed to transform the way education is experienced.

In today's digital age, online learning has become essential. Today, I'll highlight how our platform simplifies course management, student enrollment, and progress tracking, all built with ASP.NET Core and MVC architecture.

The platform is designed to enhance the learning experience for students while making administration easier for institutions.



Vision And Mission

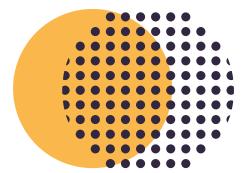
Vision

To become the leading platform for online learning, empowering institutions to deliver high-quality education while providing students with a flexible, engaging, and accessible learning experience.

Mission

- 01 Provide a seamless and user-friendly platform for managing online courses and student progress.
- 02 Enhance educational accessibility by offering flexible, scalable solutions for institutions and learners.
- 03 Ensure a secure, efficient, and high-performance system that simplifies administration and improves the overall learning experience.



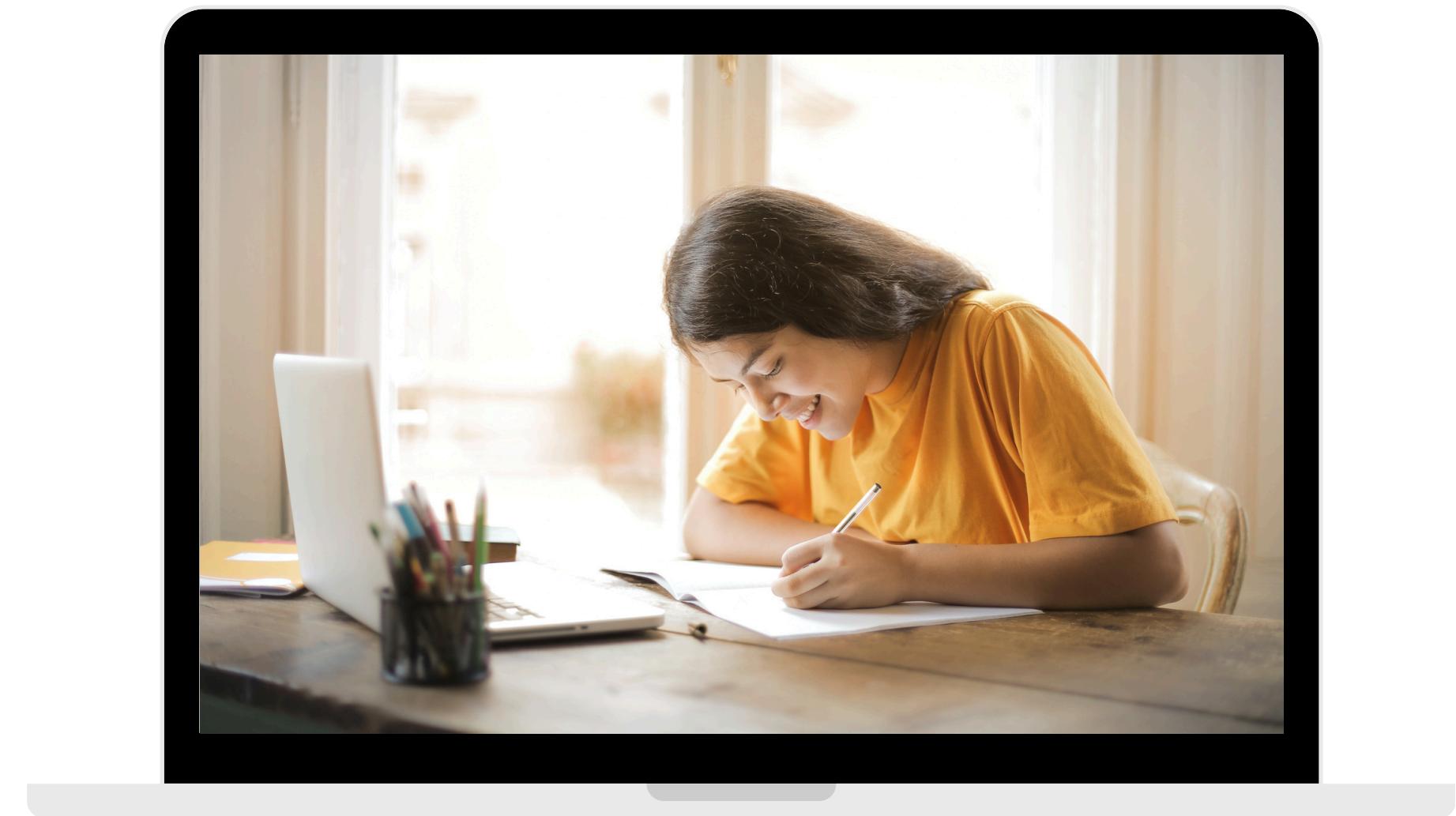


Our Service

- | | |
|--|---|
| <p>→ Micro Learning 01</p> <hr/> | <p>06 Mobile Learning ←</p> <hr/> |
| <p>→ Game Based Learning 02</p> <hr/> | <p>07 Learning Paths ←</p> <hr/> |
| <p>→ Virtual Reality Learning 03</p> <hr/> | <p>08 Scenario-Based Learning ←</p> <hr/> |
| <p>→ Translation & Localization 04</p> <hr/> | <p>09 Certification ←</p> <hr/> |
| <p>→ Interactive Videos 05</p> <hr/> | <p>10 Performance Support ←</p> <hr/> |



Always Be Learnable



Team Contribution :

Abdullah: Model Development (Data Models)

Nosshy: Controller Development (CRUD Operations)

Kareem: Repository Design Pattern (Data Access)

Samar: Design and implement

Ahmed: Identity and Authorization (User Management)

1

Model Development (Data Models) :

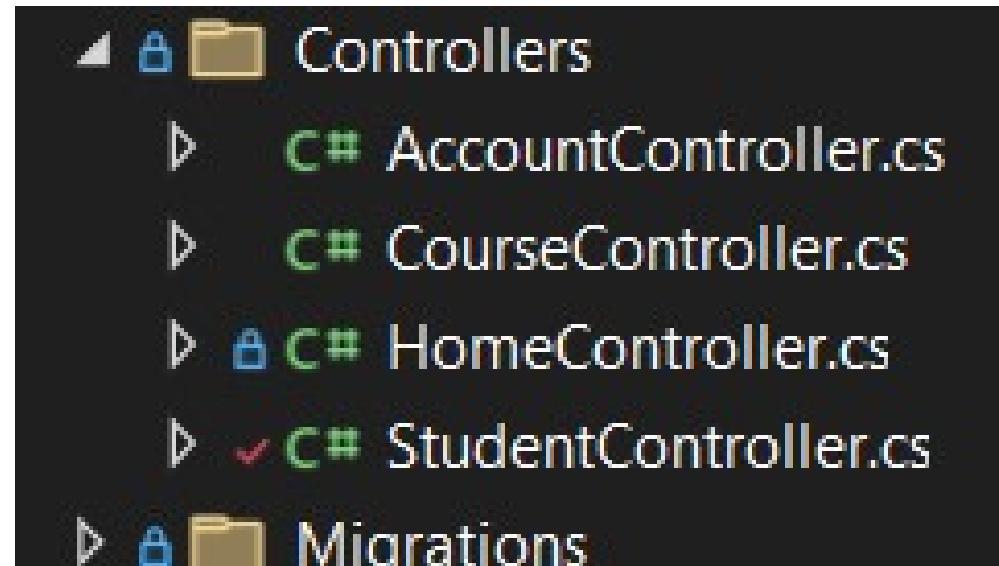
Design and implement the data models for the application, including Student, Course, and Task models.

Define relationships between models (e.g., one-to-many relationships between courses and tasks).

Create Data Transfer Objects (DTOs) if necessary for better data handling.

Create shared layout files to maintain a consistent UI across the application.

For example



The screenshot shows the Visual Studio code editor with the 'Course.cs' file open. The code defines a class named 'Course' with properties for Id, Title, Description, and Enrollments. The 'Title' and 'Description' properties are annotated with [Required] and [StringLength(200)]. The 'Enrollments' property is of type ICollection<Enrollment>. The code editor interface includes tabs for 'Program.cs', '_AdvancedLayout.cshtml', 'Index.cshtml', 'Edit.cshtml', 'Details.cshtml', and 'Delete.cshtml'. The status bar at the bottom shows the path 'Elearn.DAL' and the file name 'Elearn.DAL.Models.Course'.

```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search Elearn P
course.cs X Program.cs _AdvancedLayout.cshtml Index.cshtml Edit.cshtml Details.cshtml Delete.cshtml
Elearn.DAL
Elearn.DAL.Models.Course
1     using System.ComponentModel.DataAnnotations;
2
3     namespace Elearn.DAL.Models
4     {
5         public class Course
6         {
7             public int Id { get; set; }
8             [Required]
9             [StringLength(200)]
10            public string Title { get; set; }
11            public string Description { get; set; }
12
13            public ICollection<Enrollment> Enrollments { get; set; }
14        }
15    }
16
17
18
```

2

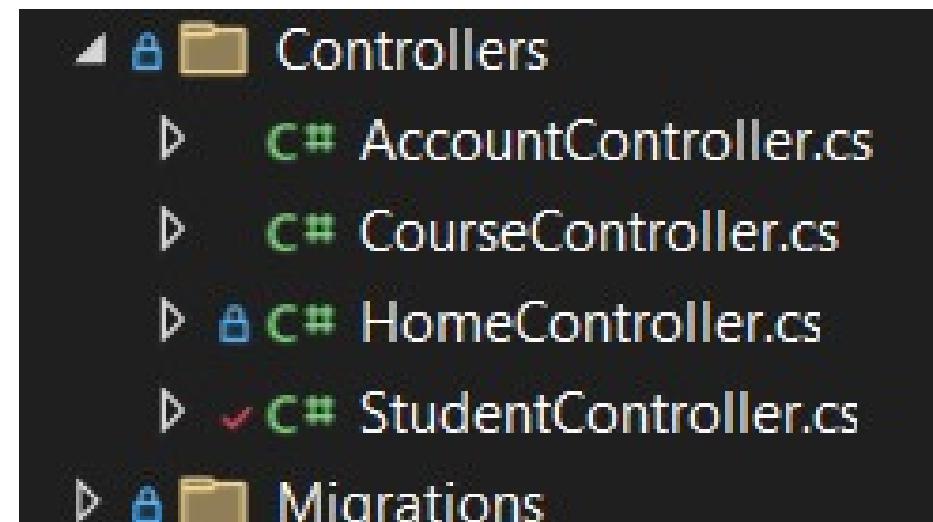
Controller Development (CRUD Operations):

Implement controllers to manage CRUD operations for the models (Create, Read, Update, Delete).

Develop actions for each controller that handle the business logic and interact with the models and views.

Ensure proper routing and handle incoming requests for course and task management.

For example



```
1  using Elearn.DAL.Models;
2  using ELearn.BLL.ViewModels.StudentVM;
3  using Microsoft.AspNetCore.Identity;
4  using Microsoft.AspNetCore.Mvc;
5
6  namespace Elearn_Project.Controllers
7  {
8      public class AccountController : Controller
9      {
10         private readonly UserManager<Student> _userManager;
11         private readonly SignInManager<Student> _signInManager;
12
13         public AccountController(UserManager<Student> userManager, SignInManager<Student> signInManager)
14         {
15             _userManager = userManager;
16             _signInManager = signInManager;
17         }
18
19         [HttpGet]
20         public IActionResult SignUp()...
21
22         [HttpPost]
23         public async Task<IActionResult> SignUp(CreateStudentVM studentVM)...
24
25         [HttpGet]
26         public IActionResult Login()...
27
28         [HttpPost]
29         public async Task<IActionResult> Login(LoginVM loginVM)...
30     }
31 }
```

3

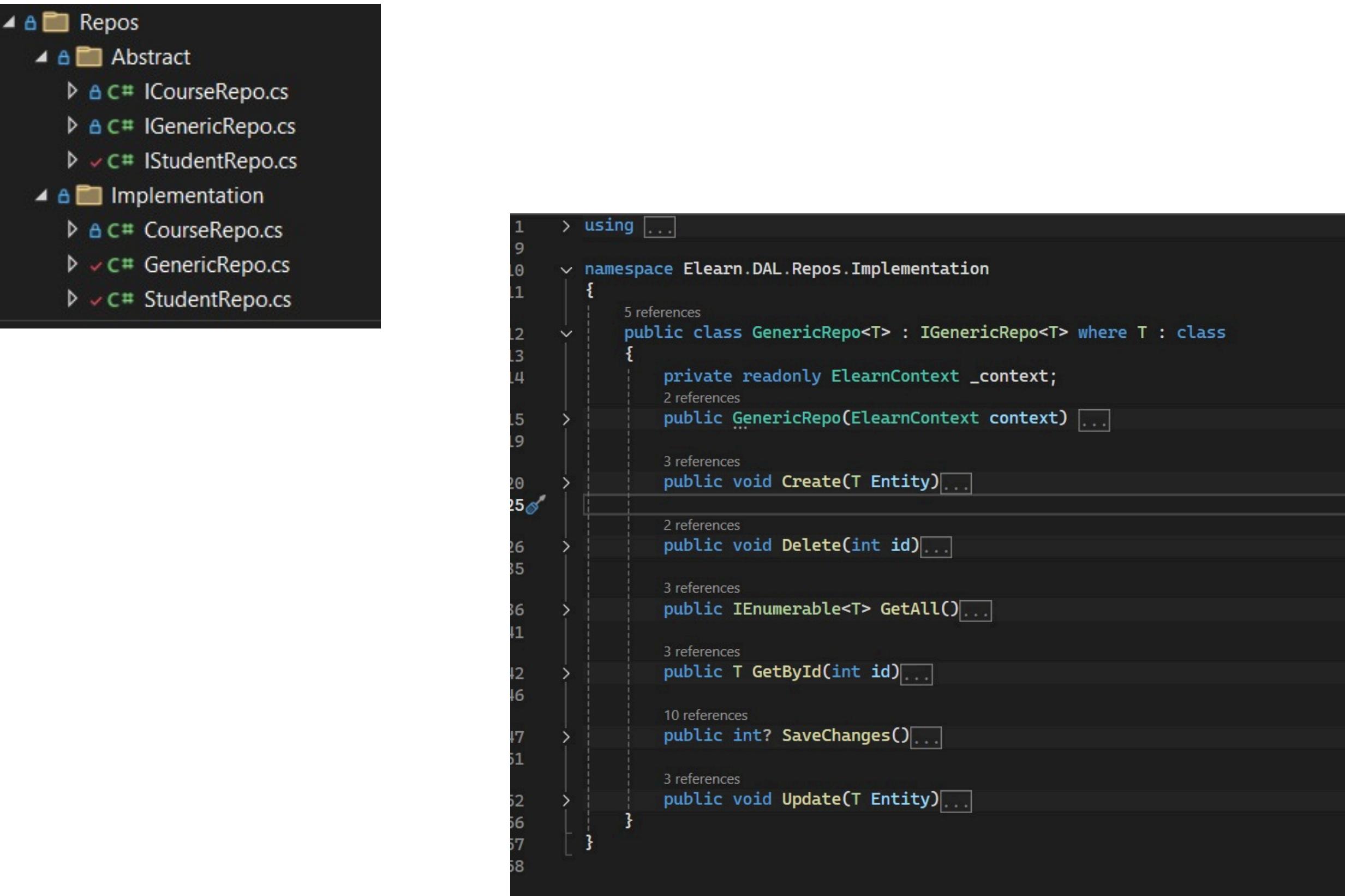
Repository Design Pattern (Data Access) :

Implement the repository design pattern to separate data access logic from the business logic.

Create repository interfaces and concrete classes for managing data operations (e.g.,
ICourseRepository, IStudentRepository).

Utilize dependency injection to inject repository instances into controllers.

For example



The image shows a file explorer on the left and a code editor on the right. The file explorer displays a directory structure under 'Repos' with two main folders: 'Abstract' and 'Implementation'. The 'Abstract' folder contains three files: 'ICourseRepo.cs', 'IGenericRepo.cs', and 'IStudentRepo.cs'. The 'Implementation' folder contains three files: 'CourseRepo.cs', 'GenericRepo.cs', and 'StudentRepo.cs'. The 'GenericRepo.cs' file is selected in the code editor, showing its implementation:

```
1 > using ...
9
10 < namespace Elearn.DAL.Repos.Implementation
11 {
12     < 5 references
13     public class GenericRepo<T> : IGenericRepo<T> where T : class
14     {
15         < 2 references
16         private readonly ElearnContext _context;
17         < 2 references
18         public GenericRepo(ElearnContext context) ...
19
20         < 3 references
21         public void Create(T Entity)...
22
23         < 2 references
24         public void Delete(int id)...
25
26         < 3 references
27         public IEnumerable<T> GetAll()...
28
29         < 3 references
30         public T GetById(int id)...
31
32         < 10 references
33         public int? SaveChanges()...
34
35         < 3 references
36         public void Update(T Entity)...
37
38     }
39 }
```

4

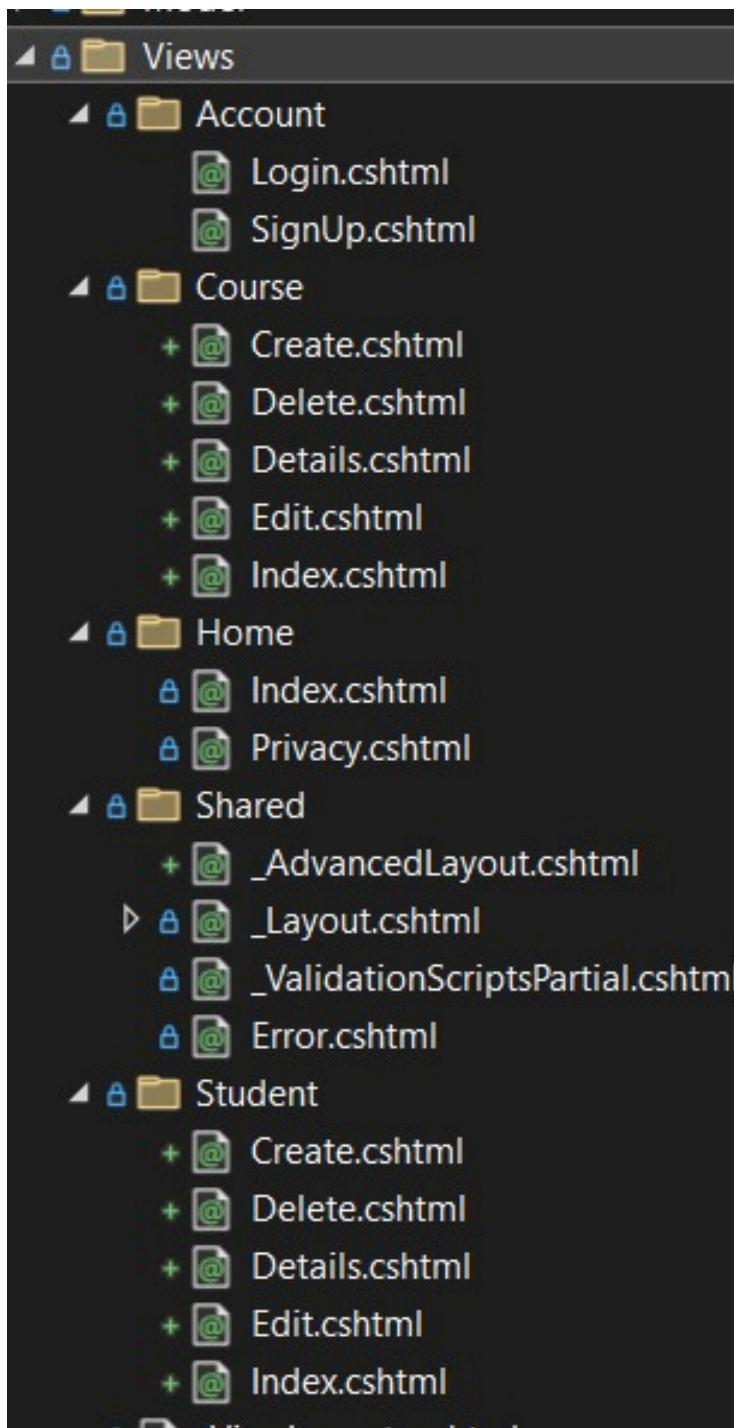
Design and implement:

Design and implement Razor views for displaying courses, managing tasks, and user profiles.

Utilize HTML, CSS, JavaScript, jQuery, and Bootstrap for responsive design and interactivity.

HTML, CSS, JavaScript, jQuery, Bootstrap, Razor.

For example



```
@model ELearn.BLL.ViewModels.StudentVM.LoginVM



## Login



<form asp-action="Login" method="post">
    <div class="form-group">
        <label asp-for="Email"></label>
        <input asp-for="Email" class="form-control" />
        <span asp-validation-for="Email" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="Password"></label>
        <input asp-for="Password" class="form-control" type="password" />
        <span asp-validation-for="Password" class="text-danger"></span>
    </div>
    <br>
    <button type="submit" class="btn btn-primary">Login</button>
</form>
    <div class="form-group">
        <p>Don't have an account? <a asp-action="SignUp">Sign up here</a>.</p>
    </div>
</div>
@section Scripts {
    <partial name="_ValidationScriptsPartial" />
}


```

5

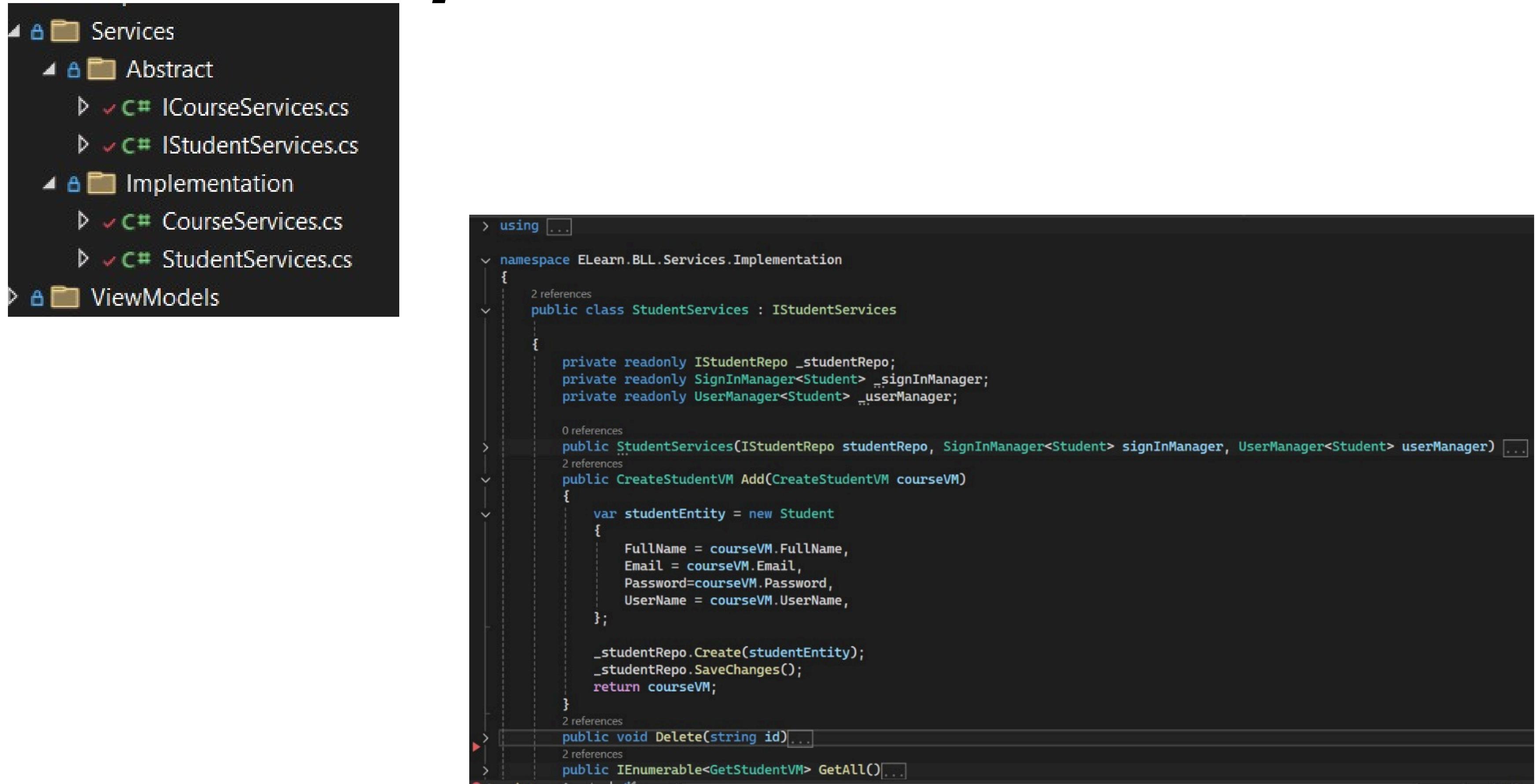
Identity and Authorization (User Management) :

Implement user authentication and authorization using ASP.NET Core Identity.

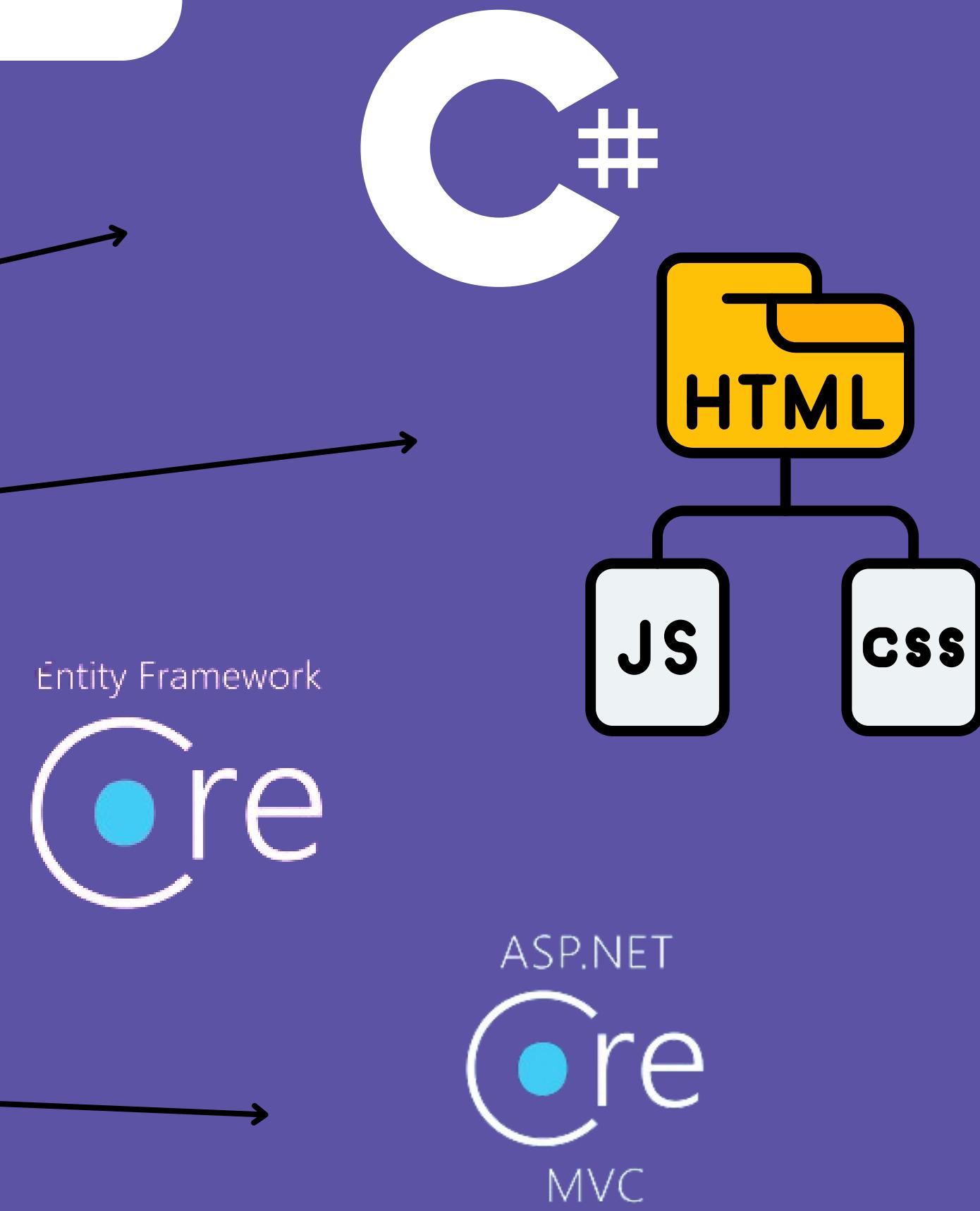
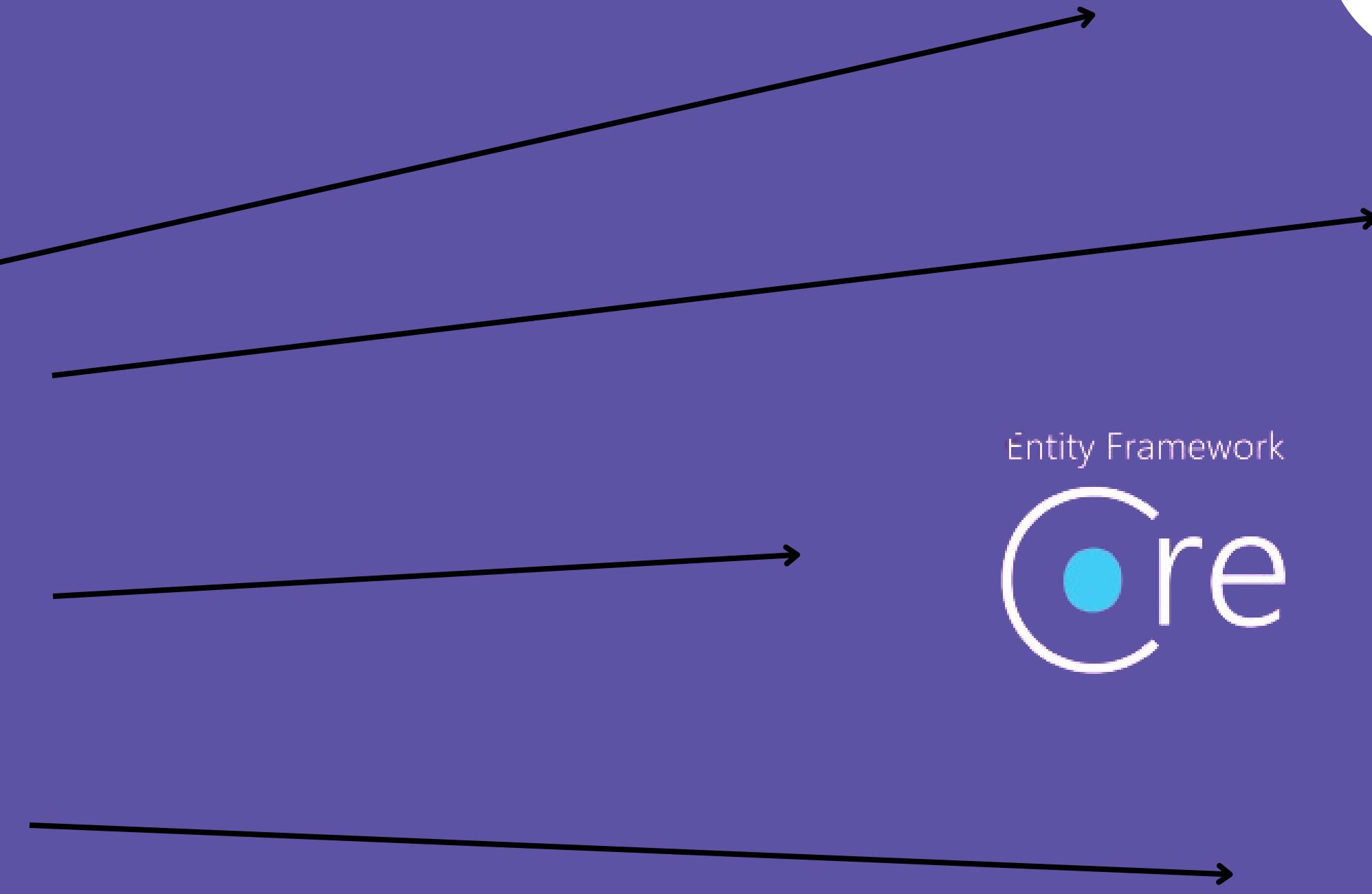
Develop login, registration, and account management functionalities.

Set up role-based access control to restrict access to certain actions based on user roles (students and instructors).

For example



Technologies



Tools



our app

E-learning

Dashboard

Student

Course

Sign Out

Privacy

Home

Students List

Full Name	Email	User Name	Actions
Abdullah Mohamed Saed	abdallaamoe@gmail.com	Abdullah	<button>Details</button> <button>Edit</button> <button>Delete</button>
El Sayed	aadds11@gmail.com	Sayed	<button>Details</button> <button>Edit</button> <button>Delete</button>
Abdulla1	abdallaamoe@gmail.com	A122	<button>Details</button> <button>Edit</button> <button>Delete</button>
Karim Mohamed Saed	abdallaamoe@gmail.com	Abdullah	<button>Details</button> <button>Edit</button> <button>Delete</button>
El Sayed	aadds11@gmail.com	Sayed	<button>Details</button> <button>Edit</button> <button>Delete</button>

[Create New Student](#)

© 2024-2025 E-Learning Management System All rights reserved.

Version 2.4.18

E-learning

Dashboard

Student

Course

Sign Out

Privacy

Home

Create Student

FullName
Ahmed

Email
aadds11@gmail.com

UserName
Ahmed11

Password

[Create](#) [Back to List](#)

DEPI Team - Web Developer
Member since Nov. 2012

Sign Out

© 2024-2025 E-Learning Management System All rights reserved.

Version 2.4.18

our app

The screenshot shows the 'Create Course' form. The title 'Create Course' is at the top. It has two input fields: 'Title' (containing 'C++') and 'Description' (containing 'C++ Description'). Below the form are 'Create' and 'Cancel' buttons. The footer of the page includes the copyright notice '© 2024-2025 E-Learning Management System All rights reserved.' and the version 'Version 2.4.18'.

Conclusion

The Online Learning Platform aims to streamline the learning process for students while empowering instructors with robust management tools.

ASP.NET Core provides a solid foundation for building a scalable and maintainable application.



About Us

We are a dedicated team of developers graduating from the **Digital Egypt Pioneers initiative**, specializing in the **.NET Backend** track. Our journey in this program has equipped us with the skills to build modern, scalable, and efficient backend solutions.

As part of our graduation project, we are focused on leveraging the power of ASP.NET Core and other cutting-edge technologies to solve real-world problems and deliver impactful solutions. Our passion for technology, combined with our strong educational foundation, drives us to contribute to the tech community in Egypt and beyond.



Our Team



Abdullah Mohamed



Mohamed Nossy



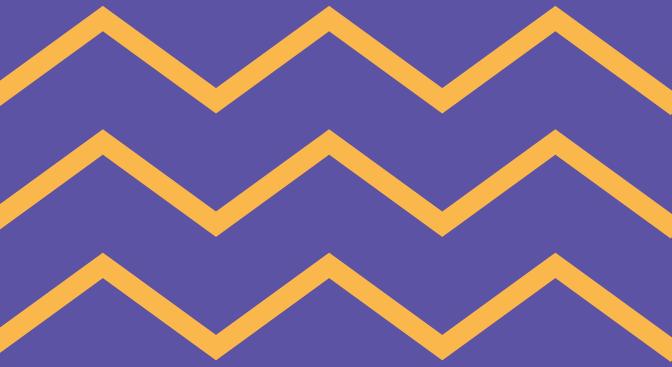
Kareem Sameh



Ahmed Abdelhamed



Samar Nafea



Thanks