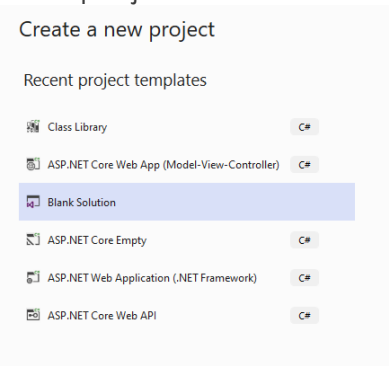
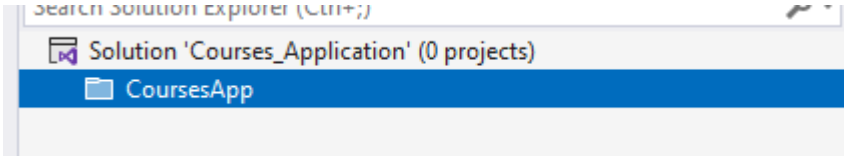
ROADMAP TO 100%
ИНТЕГРИРАНИ СИСТЕМИ – ONION ARCHITECTURE

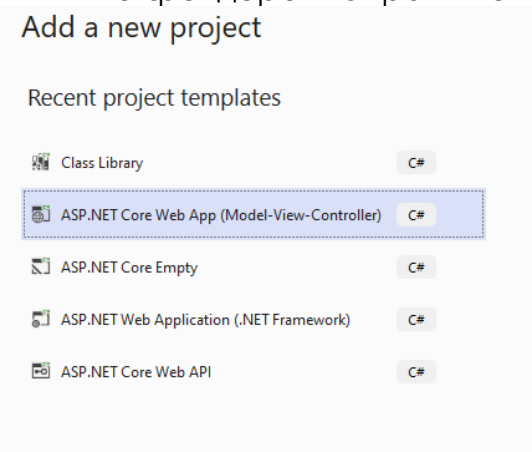
1. КРЕИРАЊЕ И СЕТИРАЊЕ ПРОЕКТ

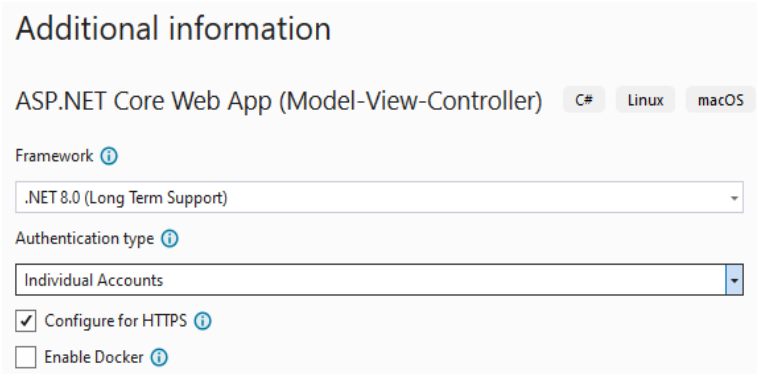
New project > Blank Solution



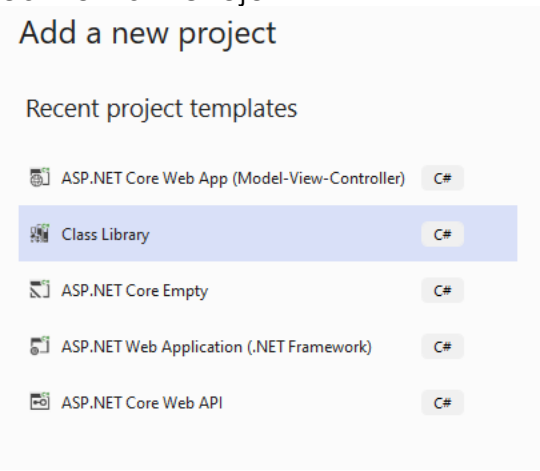


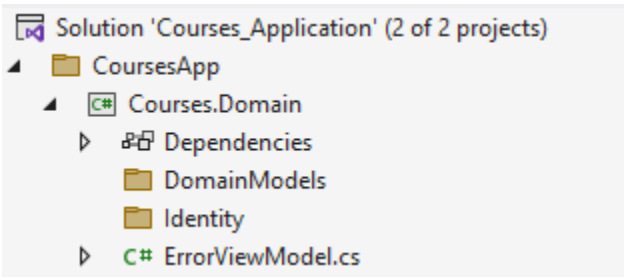
Во фолдерот направи нов проект (Courses.Web) и тоа





За Domain слојот

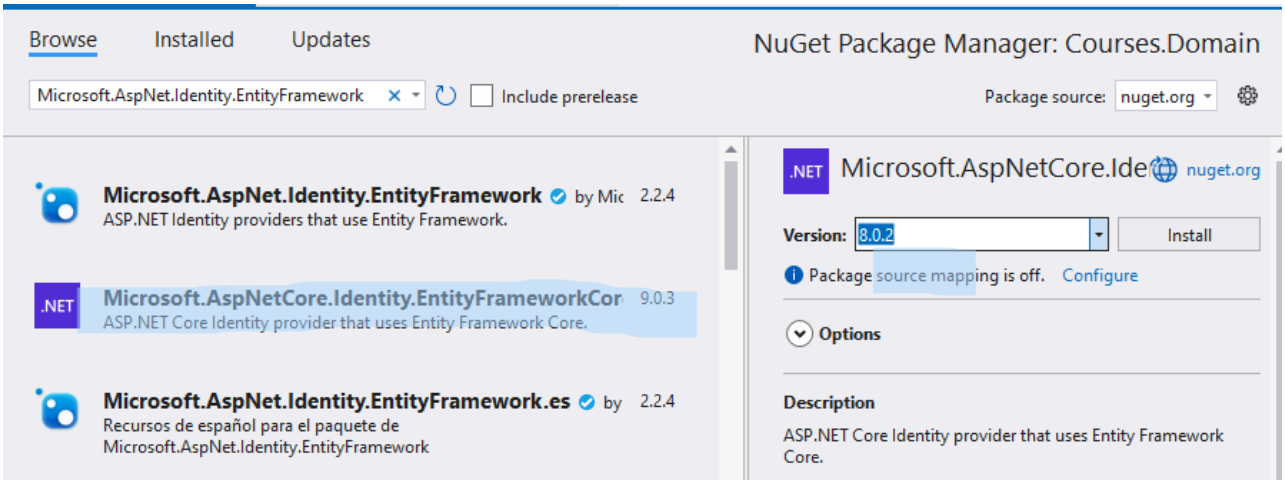




Додаваме два фолдери и од .Web делот бришиме модели, а моделот за грешка го префрламе тука.

2. ДОДАВАЊЕ APPLICATION USER

- Со десен клик на слоевите одиме во Manage NuGetPackages и таму **спуштаме иста верзија** од ASP.NET Core Entity Framework



- **Во Domain>Identity>ApplicationUser.cs (креираме класа).** Во класата треба да додадеме плус информации кои сакаме да ги содржи нашата регистрација на корисници.

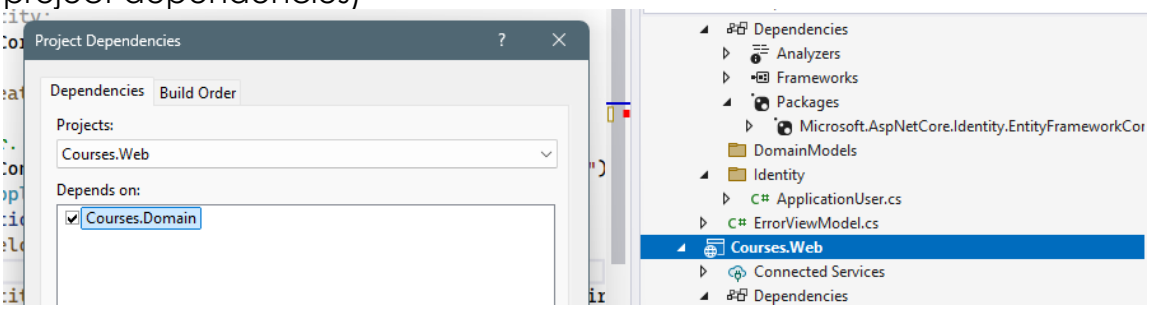
```
using Microsoft.AspNetCore.Identity;

namespace Courses.Domain.Identity
{
   public class ApplicationUser : IdentityUser
   {
        //ОВАЈ ДЕЛ СЕ МЕНУВА ВО ЗАВИСНОСТ ОД БАРАЊАТА
      public string? FirstName { get; set; }
      public string? LastName { get; set; }
      public DateTime DateOfBirth { get; set; }


   }
}
```

- **За да се импортира нов корисник треба да се направат следните три измени во фајловите**

1. **Program.cs**
   Додади прво зависност од Домеин слојот во Веб слојот (десен клик – project dependencies)




```
builder.Services.AddDefaultIdentity<ApplicationUser>(options => options.SignIn.RequireC
    .AddEntityFrameworkStores<ApplicationDbContext>();
builder.Services.AddControllersWithViews();
```

2. **ApplicationDbContext.cs (Web>>Data)**

```
using Courses.Domain.Identity;
using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore;

namespace Courses.Web.Data
{
    6 references
    public class ApplicationDbContext : IdentityDbContext <ApplicationUser>
    {
```

3. **Views/Shared/_LoginPartial.cshtml**

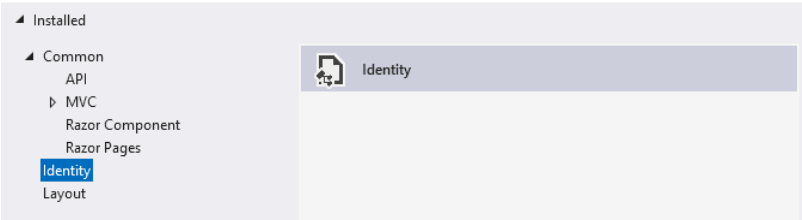
```
@using Courses.Domain.Identity
@using Microsoft.AspNetCore.Identity
@inject SignInManager<ApplicationUser> SignInManager
@inject UserManager<ApplicationUser> UserManager
```

ДОДАДИ ЈА ПРВАТА МИГРАЦИЈА (add-migration Initial + update-database)

3. ИЗМЕНА НА REGISTER

- Десен клик на Веб папката >> Add new scaffolded item>> Identity

Add New Scaffolded Item



Од листата бираме Register и долу ја одбираме нашата база и креираме. Така креираното се наоѓа во Areas>Identity>Pages>>Account>>Register.chtml

1. **Register.cshtml.cs**

   Во оваа класа додаваме се што имаме додадено во ApplicationUser

```
public class InputModel
{
    /// <summary>
    ///     This API supports the ASP.NET Core Identity
    ///     directly from your code. This API may change
    /// </summary>
    [Required]
    [EmailAddress]
    [Display(Name = "Email")]

    7 references
    public string Email { get; set; }
    [Required]
    0 references
    public string FirstName { get; set; }
    [Required]
    0 references
    public string LastName { get; set; }
    [Required]
    0 references
    public DateTime DateOfBirth { get; set; }
```

```
0 references
public async Task<IActionResult> OnPostAsync(string returnUrl = null)
{
    returnUrl ??= Url.Content("~/");
    ExternalLogins = (await _signInManager.GetExternalAuthenticationSchemesAsync()).To
    if (ModelState.IsValid)
    {
        var user = CreateUser();

        await _userStore.SetUserNameAsync(user, Input.Email, CancellationToken.None);
        await _emailStore.SetEmailAsync(user, Input.Email, CancellationToken.None);

        user.FirstName = Input.FirstName;
        user.LastName = Input.LastName;
        user.DateOfBirth = Input.DateOfBirth;
        var result = await _userManager.CreateAsync(user, Input.Password);
```

2. Register.cshtml -> додаваме полиња за истите променливи

```
<div class="form-floating mb-3">
    <input asp-for="Input.FirstName" class="form-control" autocomplete="firstname" aria-required="true" />
    <label asp-for="Input.FirstName">Name</label>
    <span asp-validation-for="Input.FirstName" class="text-danger"></span>
</div>

<div class="form-floating mb-3">
    <input asp-for="Input.LastName" class="form-control" autocomplete="surname" aria-required="true" />
    <label asp-for="Input.LastName">Surname</label>
    <span asp-validation-for="Input.LastName" class="text-danger"></span>
</div>

<div class="form-floating mb-3">
    <input asp-for="Input.DateOfBirth" class="form-control" autocomplete="dateofbirth" aria-required="true" />
    <label asp-for="Input.DateOfBirth">Date of birth</label>
    <span asp-validation-for="Input.DateOfBirth" class="text-danger"></span>
</div>
```

4. КРЕИРАЊЕ МОДЕЛИ + CRUD ОПЕРАЦИИ

Моделите ги правиме во Domain>DomainModels. Ако имаме 1:M секогаш од страната на M вметнуваме ИД од другата страна како и цел објект од истиот. Во случај на M:M релација се прави нов модел и тоа изгледа вака:

```
public class Student
{
    public int Id { get; set; }
    public string Name { get; set; }

    // Наведување на релацијата со наставници
    public ICollection<StudentTeacher> StudentTeachers { get; set; }
}
```

```
public class StudentTeacher
{
    public int StudentId { get; set; }
    public Student Student { get; set; }

    public int TeacherId { get; set; }
    public Teacher Teacher { get; set; }
}
```

```
public class Teacher
{
    public int Id { get; set; }
    public string Name { get; set; }

    // Наведување на релацијата со студенти
    public ICollection<StudentTeacher> StudentTeachers { get; set; }
}
```

```
public class ApplicationDbContext : DbContext
{
    public DbSet<Student> Students { get; set; }
    public DbSet<Teacher> Teachers { get; set; }
    public DbSet<StudentTeacher> StudentTeachers { get; set; }
```

Откако ќе ги направиме моделите треба да се оди во **ApplicationDbContext и да се додадат сите модели!**

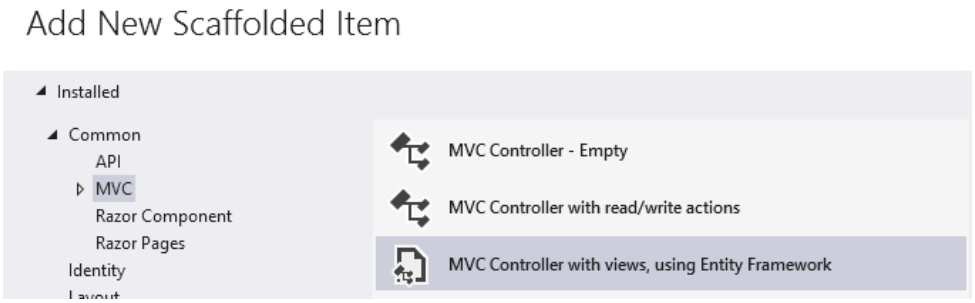
```
namespace Courses.Web.Data
{
    7 references
    public class ApplicationDbContext : IdentityDbContext <ApplicationUser
    {
        0 references
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext>
            : base(options)
        {
        }

        0 references
        public virtual DbSet<Student> Students { get; set; }
        0 references
        public virtual DbSet<Enrollment> Enrollments { get; set; }
        0 references
        public virtual DbSet<Course> Courses { get; set; }
        0 references
        public virtual DbSet<Lecture> Lectures { get; set; }
    }
}
```

НОВА МИГРАЦИЈА!!

5. КРЕИРАЊЕ КОНТРОЛЕРИ И VIEW-А

Само десен клик на папката контролери, Add new scaffolded item, MVC



Ова се повторува за сите модели

6. ДОДАВАЊЕ КОПЧИЊА ЗА НАВИГАЦИЈА

Оди во Views/Shared/Layouts и само како веќе постоечките додади нови за ИНДЕКС страните на твоите контролери

```
            </a>
        <li class="nav-item">
            <a class="nav-link text-dark" asp-area="" asp-controller="Students" asp-action="Index">Students</a>
        </li>
```