

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА ПРОГРАМУВАННЯ»

«Програмна технологія DOTNET»

*Звіт з лабораторної роботи №5*

*Тема: «ASP.NET CORE MVC»*

Виконав:  
ст. гр. КІТ-118в  
Соколенко Д.Г.

Перевірив:  
Бартош М.В.

Харків – 2020

**Мета:** отримати практичний досвід у розробці програм на С# із додаванням виведення контейнеру у веб-версію з використанням asp.net core mvc.

### **Завдання:**

Виведення контейнер з попередньої роботи у веб-версію з використанням asp.net core mvc.

### **Хід роботи**

Структура проекту (рис.1):

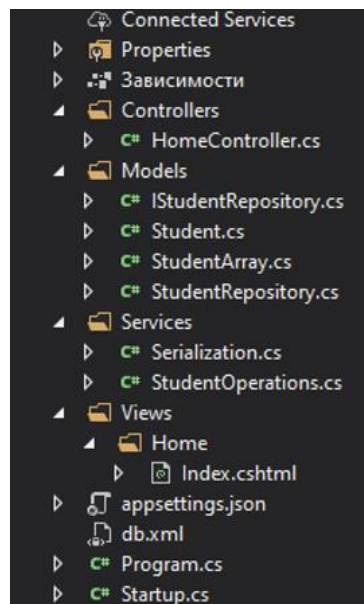


Рисунок 1 – Структура проекту

Клас Startup – налаштовує веб-додаток

```
public class Startup
{
    // This method gets called by the runtime. Use this method to add services to the container.
    Ссылка: 0
    public void ConfigureServices(IServiceCollection services)
    {
        services.AddMvc();
        services.AddLogging();
        services.AddSingleton<IStudentRepository, StudentRepository>();
    }

    // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
    Ссылка: 0
    public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
    {
        if (env.IsDevelopment())
        {
            app.UseDeveloperExceptionPage();
        }

        app.UseRouting();

        app.UseEndpoints(endpoints =>
        {
            endpoints.MapControllerRoute(
                name: "default",
                pattern: "{controller=Home}/{action=Index}/{id?}");
        });
    }
}
```

Рисунок 2 – Клас Startup

Клас HomeController – оброблює запити користувача

```
[Route("api/[controller]")]
ссылка: 1
public class HomeController : Controller
{
    Ссылка: 0
    public HomeController(IStudentRepository studItems)
    {
        StudItems = studItems;
    }

    Ссылка: 3
    public IStudentRepository StudItems { get; set; }

    Ссылка: 0
    public IActionResult Index()
    {
        return View(StudItems.GetAll());
    }

    [HttpGet("{id}", Name = "GetStud")]
    Ссылка: 0
    public IActionResult GetById(long id)
    {
        var item = StudItems.Find(id);
        if (item == null)
        {
            return NotFound();
        }
        return new ObjectResult(item);
    }
}
```

Рисунок 3 – Клас HomeController

Клас IStudentRepository – виконує операції з даними про студентів

```
ссылка: 4
public interface IStudentRepository
{
    Ссылка: 1
    void Add(Student item);
    Ссылка: 2
    IEnumerable<Student> GetAll();
    Ссылка: 2
    Student Find(long id);
    Ссылка: 1
    Student Remove(long id);
}
```

Рисунок 4 – Клас IStudentRepository

Файл Index.cshtml – використовується для створення View

```
@model IEnumerable<DotNetLab8.Models.Student>

<style type="text/css">
    table, th, td {
        border: 1px solid black;
        border-collapse: collapse;
    }
</style>

<h3>Students</h3>

<table style="width: 100%;">
    <tr>
        <th>Student</th>
        <th>Age</th>
        <th>Enter date</th>
        <th>Group</th>
        <th>Faculty</th>
        <th>Specialty</th>
        <th>Academic performance</th>
    </tr>
    @foreach (var s in Model)
    {
        <tr>
            <td>@($"{s.Surname} {s.Name} {s.Patronymic}")</td>
            <td>@($"{s.Age}")</td>
            <td>@($"{s.EnterDate}")</td>
            <td>@($"{s.Group}")</td>
            <td>@($"{s.Faculty}")</td>
            <td>@($"{s.Specialty}")</td>
            <td>@($"{s.AcademicPerformance}")</td>
        </tr>
    }
</table>
```

Рисунок 4 – Файл Index.cshtml

Отримуємо таблицю студентів

Студент	Дата рождения	Дата поступления	Возраст	Факультет	Индекс группы	Специальность
Соколенко Дмитрий Григорьевич	06.09.2001 00:00:00	01.09.2018 00:00:00	20	КИТ	в	123
Фалопьев Олег Викторович	01.03.2000 00:00:00	01.09.2018 00:00:00	21	КИТ	в	123
Гелпина Мая Владимировна	07.07.2003 00:00:00	01.09.2018 00:00:00	18	КИТ	в	123

Рисунок 5 – Результат виконання програми SpaseX

Students

Student	Age	DateOfBirth	Enter date	Group	Faculty	Specialty	Academic performance	Add	Remove
Sokolenko Dmitry Grigorovich	19	06.09.2001	01.09.2018	CIT-v	CIT	IT-123	98	<a href="#">Add</a>	<a href="#">Remove</a>

Рисунок 6 – перезапуск програми після модифікації

Edit Student

Value Name	Current Value	Saved Value
Surname	Sokolenko	Sokolenko
Name	Dmitry	Dmitry
Patronymic	Grigorovich	Grigorovich
DateOfBirth	06.09.2001 0:00:00	06.09.2001 0:00:00
Enter date	01.09.2018 0:00:00	01.09.2018 0:00:00
GroupIndex	v	v
Faculty	CIT	CIT
Specialty	IT-123	IT-123
Academic performance	98	98
Save		

Рисунок 7 – поле редагування

**Edit Student**

Value Name	Current Value	Saved Value
Surname	Sokolenko	Sokolenko
Name	Dmitry	Danilo
Patronymic	Grigorovich	Grigorovich
DateOfBirth	06.09.2001 0:00:00	06.09.2001 0:00:00
Enter date	01.09.2018 0:00:00	01.09.2018 0:00:00
GroupIndex	v	a
Faculty	CIT	KN
Specialty	IT-123	IT-122
Academic performance	98	60
Save		

Рисунок 8 – відредаговані поля у редакторі

Student	Age	DateOfBirth	Enter date	Group	Faculty	Specialty	Academic performance	Add	Remove
Sokolenko Danilo Grigorovich	19	06.09.2001	01.09.2018	KN-a	KN	IT-122	60	Edit	Remove

Рисунок 9 – змінений об'єкт типу Студент

**Висновки:** в результаті виконання лабораторної роботи отримав навички розробки програм використовуючи технологію ASP.NET Core MVC.