

Лабораторна робота №1

Тема: Вступ до ASP.NET Core

Мета: ознайомитися з основними принципами роботи .NET, навчитися налаштовувати середовище розробки та встановлювати необхідні компоненти, набути навичок створення рішень та проектів різних типів, набути навичок обробки запитів з використанням middleware.

Завдання 2. Створення проектів

Частина 1.

Створіть проект для консольного додатку з назвою ConsoleToWeb з використанням dotnet CLI

Перетворіть створений консольний додаток на веб-додаток

Опишіть виконані кроки у звіті

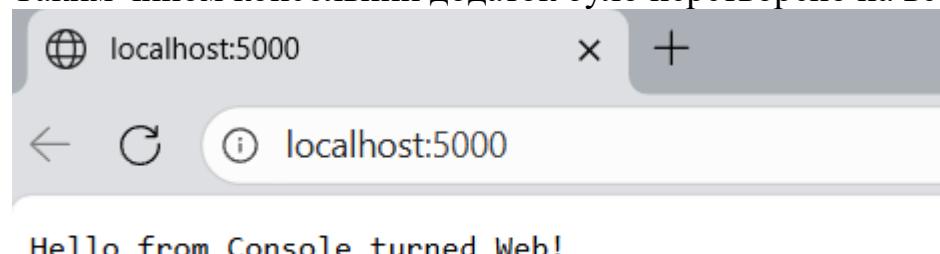
Створив проект у VScode за допомогою **dotnet new console -n ConsoleToWeb**. Потім запустив проект для перевірки чи все працює.

```
PS D:\labs\projects> cd ConsoleToWeb
PS D:\labs\projects\ConsoleToWeb> dotnet run
Hello, World!
```

Відредаговано файл Program.cs, додано використання WebApplication.

Додаток запущено командою **dotnet run**, у браузері відкривається сторінка з повідомленням “Hello from Console turned Web!”.

Таким чином консольний додаток було перетворено на веб-додаток.



Лістинг програми:

```
var builder = WebApplication.CreateBuilder(args);
var app = builder.Build();

app.MapGet("/", () => "Hello from Console turned Web!");

app.Run();
```

Частина 2.

Створіть ASP.NET WebAPI проект без авторизації з назвою WebFromCli з використанням dotnet CLI

Реалізуйте GET-ендпоінт '/who', який повертаємо у відповідь ваше ім'я та	ДУ «Житомирська політехніка».25.121.10.000 – Пр 1
Змн.	Арк.
Розроб.	Ганчевський О.О.
Перевір.	Українець М.О.
Керівник	
Н. контр.	
Зав. каф.	
Звіт з лабораторної роботи №1	
ФІКТ Гр. ІПЗ-22-2[1]	
Літ.	Арк.
	1
	6

прізвище

Реалізуйте GET-ендпоїнт “/time”, який повертаємо у відповідь поточний час на сервері

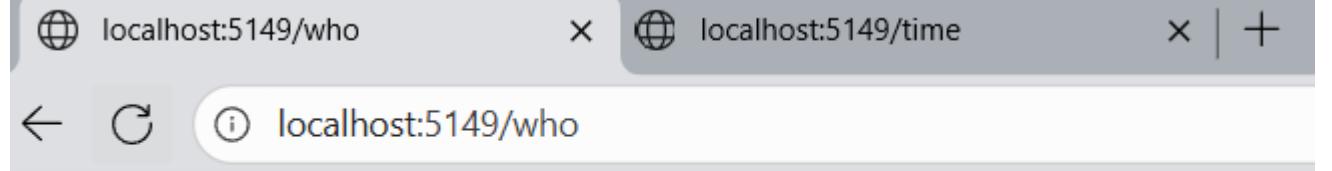
Наведіть лістинг реалізованих обробників у звіті

Створив проект за допомогою **dotnet new webapi -n WebFromCli --no-https --auth None**.

Реалізував обробники

```
app.MapGet("/who", () => "Олександр Ганчевський");
```

```
app.MapGet("/time", () => DateTime.Now.ToString("HH:mm:ss"));
```



Олександр Ганчевський



16:55:20

Лістинг програми:

```
var builder = WebApplication.CreateBuilder(args);
var app = builder.Build();
```

```
app.MapGet("/who", () => "Олександр Ганчевський");
```

```
app.MapGet("/time", () => DateTime.Now.ToString("HH:mm:ss"));
```

```
app.Run();
```

Частина 3.

Створіть ASP.NET MVC проект будь-яким зручним способом.

Реалізуйте контролер з назвою LabController

В створеному контролері реалізуйте обробник /info, який повертаємо View з даними про номер лабораторної роботи, тему, мету та ім’я та прізвище виконавця в табличному вигляді

Дані для відображення передати з контролера

Створив проект за допомогою **dotnet new mvc -n MvcLab**.

Написав контроллер:

```
using Microsoft.AspNetCore.Mvc;
```

```
namespace MvcLab.Controllers
```

		Ганчевський О. О.										Арк.
		Українець М.О.										
Змн.	Арк.	№ докум.	Підпис	Дата								2

```
{  
    public class LabController : Controller  
    {  
        public IActionResult Info()  
        {  
  
            ViewData["LabNumber"] = "Лабораторна робота №1";  
            ViewData["Topic"] = "Створення проектів ASP.NET з використанням dotnet  
CLI";  
            ViewData["Goal"] = "Навчитись створювати WebAPI та MVC-додатки через  
CLI";  
            ViewData["Author"] = "Олександр Ганчевський";  
  
            return View();  
        }  
    }  
}
```

Код View (Info.cshtml):

```
@{
    ViewData["Title"] = "Інформація про лабораторну";
}

<h2>@ViewData["Title"]</h2>

<table border="1" cellpadding="6" cellspacing="0">
    <tr>
        <th>Пункт</th>
        <th>Значення</th>
    </tr>
    <tr>
        <td>Номер лабораторної роботи</td>
        <td>@ViewData["LabNumber"]</td>
    </tr>
    <tr>
        <td>Тема</td>
        <td>@ViewData["Topic"]</td>
    </tr>
    <tr>
        <td>Мета</td>
        <td>@ViewData["Goal"]</td>
    </tr>
    <tr>
        <td>Виконавець</td>
        <td>@ViewData["Author"]</td>
    </tr>
</table>
```

Результат роботи:

В браузері за адресою <http://localhost:5000/Lab/Info> відображається таблиця з інформацією про лабораторну роботу.

		<i>Ганчевський О. О.</i>				
		<i>Українець М.О.</i>				
Змн.	Арк.	№ докум.	<i>Підпис</i>	<i>Дата</i>	ДУ «Житомирська політехніка».25. 121.10.000 – Пр 1	Арк. 3

Пункт	Значення
Номер лабораторної роботи	Лабораторна робота №1
Тема	Створення проектів ASP.NET з використанням dotnet CLI
Мета	Навчитись створювати WebAPI та MVC-додатки через CLI
Виконавець	Олександр Ганчевський

Завдання 3. Робота з middleware

Ознайомтеся з поняттям middleware в ASP.NET

Створіть ASP.NET WebAPI проект з назвою MiddlewareSandbox

Реалізуйте наступні middleware

Реалізуйте middleware, яке рахує кількість запитів до сервера і повертає це число у відповіді, наприклад: The amount of processed requests is X.

Створіть middleware, яке аналізує параметри рядка запиту. Якщо в рядку запиту (query string) присутній параметр “custom”, то повернати у відповідь “You’ve hit a custom middleware!”, інакше пропускати запит далі. Приклад запиту зображеного на рисунку 3.1.

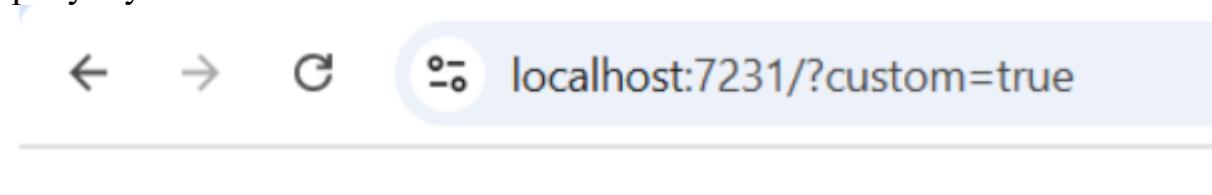


Рис. 3.1. Приклад виконання запиту

Створіть middleware, яке логуватиме у консоль інформацію про всі запити. В логах відображати метод запиту (GET, POST тощо) та його шлях (/user, /product тощо). Для перевірки POST та інших методів запиту можна скористатися Postman або іншою подібною утилітою.

Реалізуйте middleware, яке перевіряє наявність API-ключа у заголовках запиту. Для цього слід перевірити, чи є в запиті заголовок X-API-KEY. Якщо ключ неправильний або відсутній, повернати 403 Forbidden, не передавати запит для виконання далі. Якщо ключ в заголовку запиту співпадає з тим, який заданий на сервері, передавати запит далі. Для додавання відповідного запиту можна скористатися Postman або подібними утилітами.

Лістинг програми:

```
using Microsoft.AspNetCore.Builder;
```

		Ганчевський О. О.			ДУ «Житомирська політехніка».25. 121.10.000 – Пр1	Арк.
		Українець М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

```

using Microsoft.AspNetCore.Http;
using Microsoft.Extensions.Hosting;

var builder = WebApplication.CreateBuilder(args);
var app = builder.Build();

// 1 Middleware – підрахунок кількості запитів

int requestCount = 0;
app.Use(async (context, next) =>
{
    requestCount++;
    if (context.Request.Path == "/count")
    {
        await context.Response.WriteAsync($"The amount of processed requests is {requestCount}.");
    }
    else
    {
        await next();
    }
});

// 2 Middleware – перевірка параметра custom
app.Use(async (context, next) =>
{
    if (context.Request.Query.ContainsKey("custom"))
    {
        await context.Response.WriteAsync("You've hit a custom middleware!");
    }
    else
    {
        await next();
    }
});

// 3 Middleware – логування запитів

app.Use(async (context, next) =>
{
    var method = context.Request.Method;
    var path = context.Request.Path;
    Console.WriteLine($"[{DateTime.Now}] {method} {path}");
    await next();
});

// 4 Middleware – перевірка API-ключа

```

Змн.	Арк.	Ганчевський О. О.		
		Українець М.О.		

ДУ «Житомирська політехніка».25. **121.10.000** – Пр1

Арк.

5

```

const string validApiKey = "12345"; // свій ключ
app.Use(async (context, next) =>
{
    if (!context.Request.Headers.TryGetValue("X-API-KEY", out var key) || key != validApiKey)
    {
        context.Response.StatusCode = StatusCodes.Status403Forbidden;
        await context.Response.WriteAsync("Forbidden: Invalid or missing API key.");
        return;
    }

    await next();
});

// Приклад фінальної кінцевої точки

app.MapGet("/", () => "Hello from Middleware Sandbox!");

app.Run();

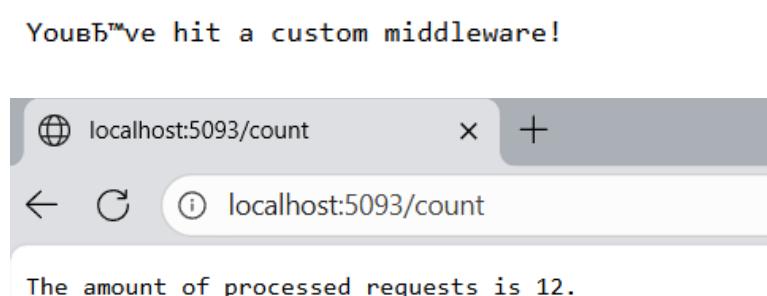
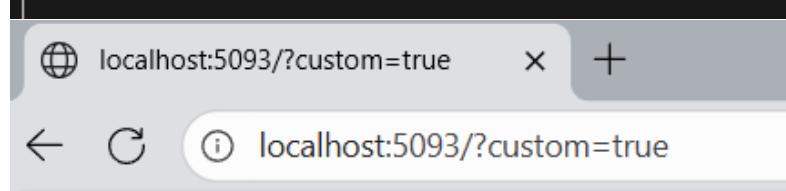
```

Результат програми:

```

[13.11.2025 17:27:37] GET /
[13.11.2025 17:27:37] GET /favicon.ico
[13.11.2025 17:27:52] GET /favicon.ico
[13.11.2025 17:28:25] GET /favicon.ico
[13.11.2025 17:30:02] GET /.well-known/appspecific/com.chrome.devtools.json
[13.11.2025 17:31:33] GET /favicon.ico
[13.11.2025 17:31:50] GET /favicon.ico

```



Висновок: ознайомився з основними принципами роботи .NET, навчився налаштовувати середовище розробки та встановлювати необхідні компоненти, набув навичок створення рішень та проектів різних типів, набув навичок обробки запитів з використанням middleware.

		Ганчевський О. О.			ДУ «Житомирська політехніка».25. 121.10.000 – Пр1	Арк.
		Українець М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		6