

## ЛАБОРАТОРНА РОБОТА № 1

**Тема:** Вступ до ASP.NET Core

**Мета роботи:** 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.

**Хід роботи:**

### Завдання 1. Встановлення інтегрованого середовища розробки (IDE) та необхідних компонентів

Since I already had Visual Studio 2022 installed and packages configured for ASP.Net and web development (Fig. 1), I will move on to the next task.

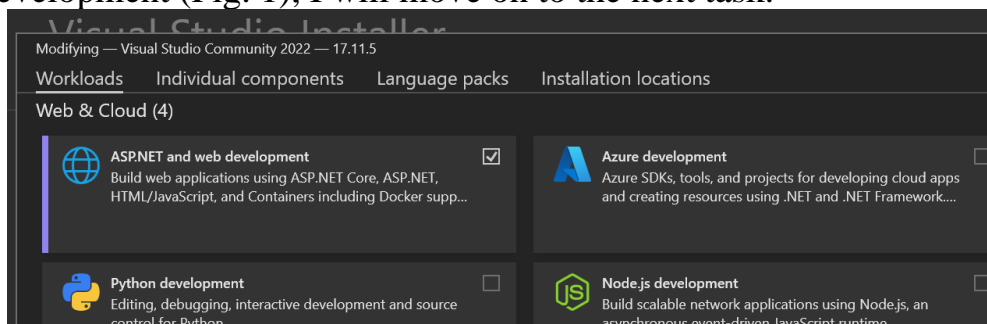


Рис. 1 – Required packages installed

### Завдання 2. Створення проектів

#### Частина 1.

1. Створіть проект для консольного додатку з назвою ConsoleToWeb з використанням (рис. 2) [dotnet CLI](#)

```
PS D:\ASP_development\pr1> dotnet new console -n ConsoleToWeb
The template "Console App" was created successfully.

Processing post-creation actions...
Restoring D:\ASP_development\pr1\ConsoleToWeb\ConsoleToWeb.csproj:
  Determining projects to restore...
  Restored D:\ASP_development\pr1\ConsoleToWeb\ConsoleToWeb.csproj (in 156 ms).
Restore succeeded.

PS D:\ASP_development\pr1> cd ConsoleToWeb
PS D:\ASP_development\pr1\ConsoleToWeb> dotnet run
Hello, World!
PS D:\ASP_development\pr1\ConsoleToWeb>
```

Рис. 2 – Create project

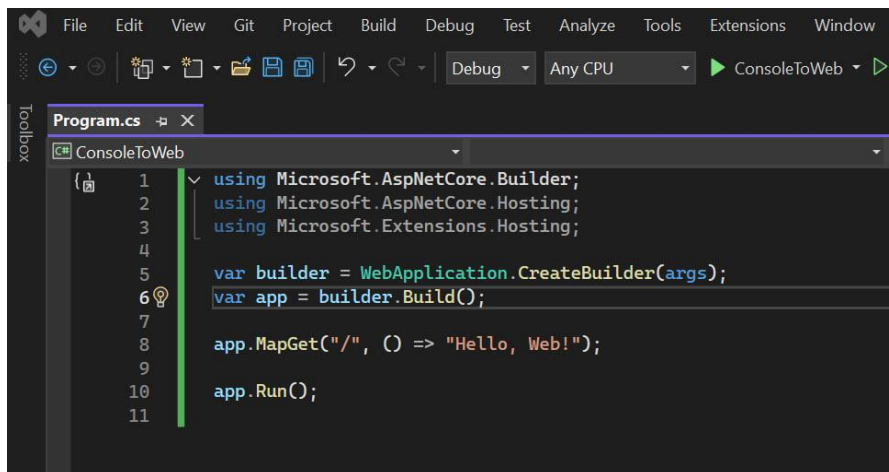
					ДУ «Житомирська політехніка».24.121.11.000 – Лр1		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Ліщинський В. В.					
Перевір.		Українець М. О.					
Керівник							
Н. контр.							
Зав. каф.							
					Звіт з лабораторної роботи		
					Лім.	Арк.	Аркушів
						1	
					ФІКТ, зр. ІПЗк-24-1		

2. Перетворіть створений консольний додаток на веб-додаток (рис. 2)
3. Опишіть виконані кроки у звіті

```
PS D:\ASP_development\pr1\ConsoleToWeb> dotnet add package Microsoft.AspNetCore.App
Determining projects to restore...
Writing C:\Users\mbyk\AppData\Local\Temp\tmp0mro2d.tmp
info : X.509 certificate chain validation will use the default trust store selected by .NET for code signing.
info : X.509 certificate chain validation will use the default trust store selected by .NET for timestamping.
info : Adding PackageReference for package 'Microsoft.AspNetCore.App' into project 'D:\ASP_development\pr1\ConsoleToWeb\ConsoleToWeb.csproj'.
info : GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.aspnetcore.app/index.json
info : OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.aspnetcore.app/index.json 425ms
info : Restoring packages for D:\ASP_development\pr1\ConsoleToWeb\ConsoleToWeb.csproj...
info : GET https://api.nuget.org/v3-flatcontainer/microsoft.aspnetcore.app/index.json
info : OK https://api.nuget.org/v3-flatcontainer/microsoft.aspnetcore.app/index.json 221ms
info : GET https://api.nuget.org/v3-flatcontainer/microsoft.aspnetcore.app/2.2.8/microsoft.aspnetcore.app.2.2.8.nupkg
info : OK https://api.nuget.org/v3-flatcontainer/microsoft.aspnetcore.app/2.2.8/microsoft.aspnetcore.app.2.2.8.nupkg 193ms
info : GET https://api.nuget.org/v3-flatcontainer/microsoft.aspnetcore.app/2.2.8/microsoft.aspnetcore.app.2.2.8.nupkg
info : GET https://api.nuget.org/v3-flatcontainer/microsoft.aspnetcore.app/2.2.8/microsoft.aspnetcore.app.2.2.8.nupkg
info : GET https://api.nuget.org/v3-flatcontainer/microsoft.aspnetcore.app/2.2.8/microsoft.aspnetcore.app.2.2.8.nupkg
info : GET https://api.nuget.org/v3-flatcontainer/microsoft.aspnetcore.app/2.2.8/microsoft.aspnetcore.app.2.2.8.nupkg
```

Рис. 2 – Convert to a web app

First, I added the necessary packages for the web application to function. Then I wrote the greeting itself (pic. 3)



```
1 using Microsoft.AspNetCore.Builder;
2 using Microsoft.AspNetCore.Hosting;
3 using Microsoft.Extensions.Hosting;
4
5 var builder = WebApplicationBuilder.CreateBuilder(args);
6 var app = builder.Build();
7
8 app.MapGet("/", () => "Hello, Web!");
9
10 app.Run();
11
```

Рис. 3 – Привітання

Then from the console I ran the program (pic. 4)

```
PS D:\ASP_development\pr1\ConsoleToWeb> dotnet run
C:\Program Files\dotnet\sdk\8.0.403\Sdks\Microsoft.NET.Sdk\targets\Microsoft.NET.Sdk.targets(1080,5): warning: A PackageReference to Microsoft.AspNetCore.App is not necessary when Microsoft.AspNetCore.App is used, the shared framework will be referenced automatically. Otherwise, a PackageReference to Microsoft.AspNetCore.App is not necessary when Microsoft.AspNetCore.App is used, the shared framework will be referenced automatically. [D:\ASP_development\pr1\ConsoleToWeb\ConsoleToWeb.csproj]
info : Microsoft.Hosting.Lifetime[14]
info : Now listening on: http://localhost:5000
info : Microsoft.Hosting.Lifetime[0]
info : Application started. Press Ctrl+C to shut down.
info : Microsoft.Hosting.Lifetime[0]
info : Hosting environment: Production
info : Microsoft.Hosting.Lifetime[0]
info : Content root path: D:\ASP_development\pr1\ConsoleToWeb
```

Рис. 4 – Launching a web application

And I followed the link: <http://localhost:5000/> (pic. 5)

		Ліщинський В.В.			ДУ «Житомирська політехніка».24.121.11.000 – Лр1	Арк.
		Українець М. О.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

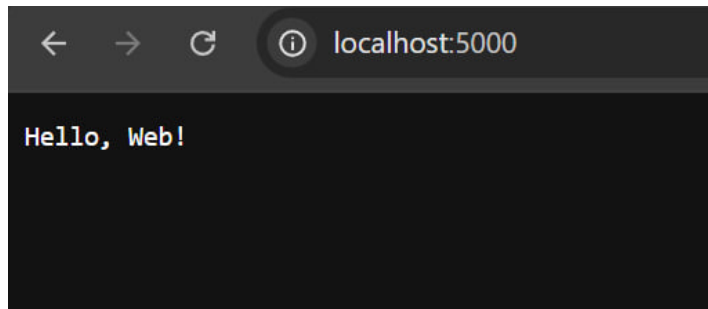


Рис. 5 – Перегляд веб-сторінки

## Частина 2.

1. Створіть ASP.NET WebAPI проект без авторизації з назвою WebFromCli з використанням [dotnet CLI](#) (рис. 6)

```
PS D:\ASP_development\pr1> dotnet new webapi -n WebFromCli --no-https
The template "ASP.NET Core Web API" was created successfully.

Processing post-creation actions...
Restoring D:\ASP_development\pr1\WebFromCli\WebFromCli.csproj:
  Determining projects to restore...
  Restored D:\ASP_development\pr1\WebFromCli\WebFromCli.csproj (in 5,44 sec).
Restore succeeded.

PS D:\ASP_development\pr1>
```

Рис. 6 – Create WebAPI project

2. Реалізуйте GET-ендпоінт “/who”, який повертатиме у відповідь ваше ім’я та прізвище (рис. 7)

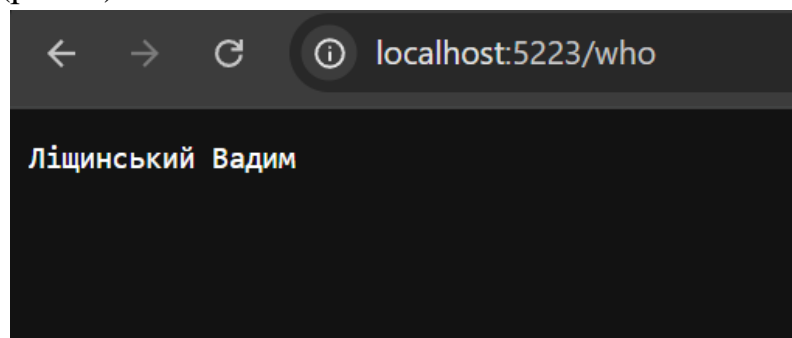


Рис. 7 – Implementing the "/who" endpoint

3. Реалізуйте GET-ендпоінт “/time”, який повертатиме у відповідь поточний час на сервері (рис. 8)

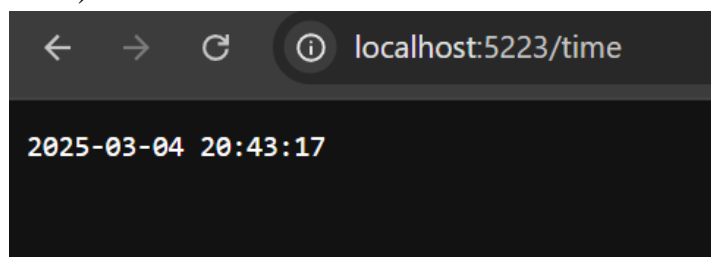


Рис. 8 – Implementing the "/time" endpoint

#### 4. Наведіть лістинг реалізованих обробників у звіті

Listing of the “/who” endpoint:

```
app.MapGet("/who", () => "Ліщинський Вадим");
```

Listing of the “/time” endpoint:

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

### Частина 3.

#### 1. Створіть ASP.NET MVC проект будь-яким зручним способом (рис. 9).

```
PS D:\ASP_development\pr1> dotnet new mvc -n MvcLab
The template "ASP.NET Core Web App (Model-View-Controller)" was created successfully.
This template contains technologies from parties other than Microsoft, see https://aka.ms/aspnetcore/8.0-third-party-notices for details.

Processing post-creation actions...
Restoring D:\ASP_development\pr1\MvcLab\MvcLab.csproj:
  Determining projects to restore...
  Restored D:\ASP_development\pr1\MvcLab\MvcLab.csproj (in 190 ms).
Restore succeeded.

PS D:\ASP_development\pr1>
```

Рис. 9 – Create MVC project

#### 2. Реалізуйте контролер з назвою LabController (рис. 10)

```
LabController.cs
using Microsoft.AspNetCore.Mvc;

namespace MvcLab.Controllers
{
    public class LabController : Controller
    {
        public IActionResult Index()
        {
            return View();
        }
    }
}
```

Рис. 10 – Controller implementation

#### 3. В створеному контролері реалізуйте обробник /info, який повертатиме View з даними про номер лабораторної роботи, тему, мету та ім'я та прізвище виконавця в табличному вигляді (рис. 11)

		Ліщинський В.В.			ДУ «Житомирська політехніка».24.121.11.000 – Пр1	Арк.
		Українець М. О.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

1  @model dynamic
2
3  <!DOCTYPE html>
4  <html>
5  <head>
6      <title>Лабораторна робота</title>
7  <style>
8      table {
9          width: 50%;
10         border-collapse: collapse;
11         margin: 20px 0;

```

Рис. 11 – Handler /info

#### 4. Дані для відображення передати з контролера (рис. 12)

I created a separate View (Info) to receive data from the LabController controller and set the default path in Program.cs to Lab/Info.cshtml (Lab/Info) so that when the web application is opened, the data is immediately received.

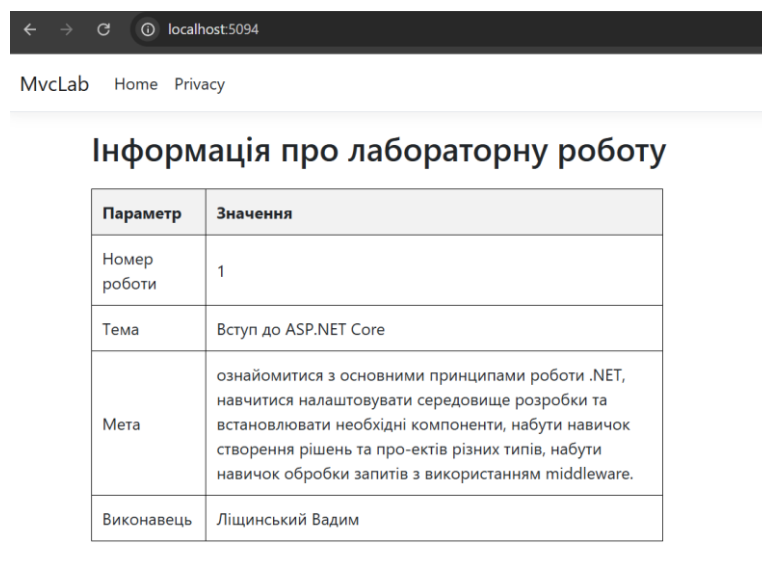


Рис. 12 – Information display

### Завдання 3. Робота з middleware

1. Ознайомтеся з [поняттям middleware](#) в ASP.NET
2. Створіть ASP.NET WebAPI проект з назвою MiddlewareSandbox (рис. 13)

```

PS D:\ASP_development\pr1> dotnet new webapi -n MiddlewareSandbox
The template "ASP.NET Core Web API" was created successfully.

Processing post-creation actions...
Restoring D:\ASP_development\pr1\MiddlewareSandbox\MiddlewareSandbox.csproj:
  Determining projects to restore...
  Restored D:\ASP_development\pr1\MiddlewareSandbox\MiddlewareSandbox.csproj (in 805 ms).
Restore succeeded.

```

Рис. 13 – Create WebAPI project

### 3. Реалізуйте наступні middleware

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

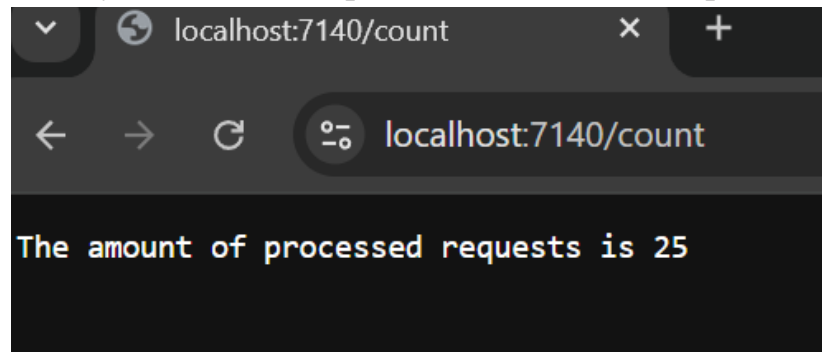


Рис. 14 –Counter implementation

- б. Створіть middleware, яке аналізує параметри рядка запиту. Якщо в рядку запиту (query string) присутній параметр “custom”, то повертати у відповідь “You’ve hit a custom middleware!”, інакше пропускати запит далі.(рис. 15)

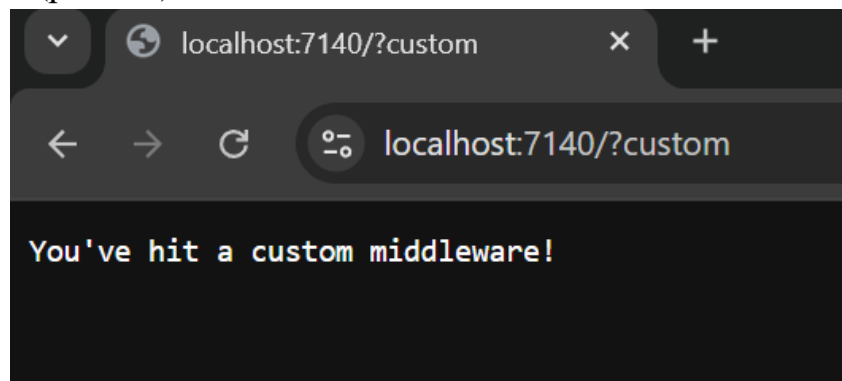


Рис. 15 –Customer implementation

- с. 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. (рис. 16)

```

info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7140
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5288
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: D:\ASP_development\pr1\Middlewaresandbox
info: MiddlewareSandbox.Middlewares.LoggerMiddleware[0]
      Request received: GET /who

```

Рис. 16 –Logger implementation

- d. Реалізуйте middleware, яке перевіряє наявність API-ключа у заголовках запиту. Для цього слід Перевіряти, чи є в запиті заголовок X-API-KEY. Якщо ключ неправильний або відсутній, повертати 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 запити можна скористатися Postman або подібними утилітами. (рис. 17)

```

Windows PowerShell
PS D:\ASP_development\pr1\Middlewaresandbox> curl.exe -X GET https://localhost:7140/who -H "X-API-KEY: my-key"
PS D:\ASP_development\pr1\Middlewaresandbox> curl.exe -X GET https://localhost:7140/who -H "X-API-KEY: just_key"
Forbidden: Invalid or missing API key.
PS D:\ASP_development\pr1\Middlewaresandbox> |

```

Рис. 17 –UseApi implementation

**Conclusions:** During the laboratory work, I got acquainted with the basic principles of .NET robots, learned how to configure the development environment and install the necessary components, acquired skills in creating solutions and projects of various types, and acquired skills in processing requests using middleware.

		Ліщинський В.В.			ДУ «Житомирська політехніка».24.121.11.000 – Лр1	Арк.
		Українець М. О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		