

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.

Хід роботи:

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

Завдання 2. Створення проектів

Частина 1.

1. Створіть проект для консольного додатку з назвою ConsoleToWeb з використанням [dotnet CLI](#)
2. Перетворіть створений консольний додаток на веб-додаток
3. Опишіть виконані кроки у звіті

Кроки виконання:

1. Створено консольний додаток за допомогою dotnet new console -n ConsoleToWeb.
2. Перевірено роботу через dotnet run.
3. В ConsoleToWeb.csproj змінено Sdk="Microsoft.NET.Sdk" на Sdk="Microsoft.NET.Sdk.Web".
4. У Program.cs додано WebApplication та ендпоінт /.
5. Запущено веб-додаток через dotnet run і перевірено у браузері.

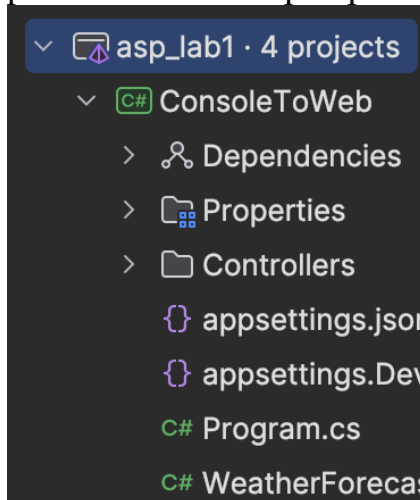


Рис. 1.1. Створено консольний додаток

Частина 2.

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

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.16.000 – Лр.1						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Кравченко К.М.			Звіт з лабораторної роботи №1			Літ.	Арк.	Аркушів	
Перевір.		Українець М.О.							1		
Реценз.								ФІКТ, гр. ІПЗ-22-2			
Н. Контр.											
Зав.каф.											

2. Реалізуйте GET-ендпоінт “/who”, який повертатиме у відповідь ваше ім’я та прізвище
3. Реалізуйте GET-ендпоінт “/time”, який повертатиме у відповідь поточний час на сервері
4. Наведіть лістинг реалізованих обробників у звіті

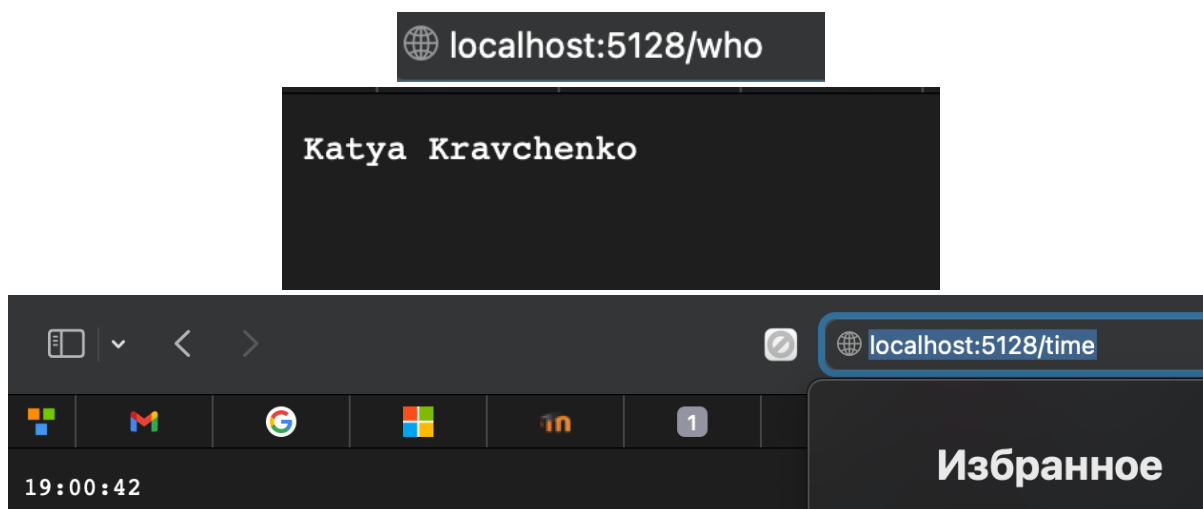


Рис.1.2. – 1.4. Реалізовані GET-ендпоінти

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

Частина 3.

1. Створіть ASP.NET MVC проект будь-яким зручним способом.
2. Реалізуйте контролер з назвою LabController
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
4. Дані для відображення передати з контролера

LabController.cs:

```
using Microsoft.AspNetCore.Mvc;
namespace MvcLab1Project.Controllers;

public class LabController : Controller
{
    public IActionResult Info()
    {
        // Дані, які передамо у View
        ViewData["LabNumber"] = "Лабораторна робота №1";
        ViewData["Topic"] = "Вступ до ASP.NET Core";
        ViewData["Goal"] = "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.";
    }
}
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.16.000 – Лр.1	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        ViewData["Student"] = "Катерина Кравченко";

        return View();
    }
}

```

Views/Lab файл Info.cshtml:

```

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

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

<table border="1" style="border-collapse: collapse; width: 60%; text-align:
left;">
    <tr>
        <th>Поле</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["Student"]</td>
    </tr>
</table>

```

MvcLab1Project [Home](#) [Privacy](#)

Інформація про лабораторну

Поле	Значення
Номер лабораторної роботи	Лабораторна робота №1
Тема	Вступ до ASP.NET Core 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.
Мета	
Виконавець	Катерина Кравченко

Рис. 1.5. Результат роботи обробника /info

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.16.000 – Лр.1	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 3. Робота з middleware

1. Ознайомтеся з [поняттям middleware](#) в ASP.NET
2. Створіть ASP.NET WebAPI проект з назвою MiddlewareSandbox
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.
 - 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.
 - в. 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.
 - г. 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 запиту можна скористатися Postman або подібними утилітами.

Program.cs:

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

// Add services to the container.

builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at
https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

int requestCount = 0;
app.Use(async (context, next) =>
{
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.16.000 – Лр.1	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        requestCount++;
        await next.Invoke();
        await context.Response.WriteAsync($"\\nThe amount of processed requests is
{requestCount}.");
    });

// Middleware 2: Перевірка параметру query string "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.Invoke();
    }
});

// Middleware 3: Логування всіх запитів у консоль
app.Use(async (context, next) =>
{
    Console.WriteLine($"[{DateTime.Now}] {context.Request.Method}
{context.Request.Path}");
    await next.Invoke();
});

// Middleware 4: Перевірка X-API-KEY
const string apiKey = "SECRET123";
app.Use(async (context, next) =>
{
    if (!context.Request.Headers.TryGetValue("X-API-KEY", out var key) || key !=
apiKey)
    {
        context.Response.StatusCode = 403;
        await context.Response.WriteAsync("Forbidden");
        return;
    }
    await next.Invoke();
});

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

app.Run();

```

You've hit a custom middleware!
The amount of processed requests is 1.

Failed to determine the https port for redirect.
[27.10.2025 18:49:41] GET /favicon.ico

Рис. 3.1.-3.2. Виконання запиту

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.16.000 – Лр.1	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Postman:

GET ▼ | https://localhost:44315/ Send ▼

Params Authorization Headers (7) Body Scripts Settings Cookies

Headers 7 hidden

	Key	Value	Description	...	Bulk Edit	Presets
	Key	Value	Description			

Body Cookies Headers (4) Test Results 🕒 403 Forbidden • 3 ms • 190 B • 🔍 Save Response ...

Raw ▼ ▶ Preview 🐞 Debug with AI ▼ 🔍 🔗

1 Forbidden
2 The amount of processed requests is 7.

GET ▼ | https://localhost:44315/ Send ▼

Params ● Authorization Headers (8) Body Scripts Settings Cookies

Headers 7 hidden

	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	X-API-KEY	SECRET123				
	Key	Value	Description			

Body Cookies Headers (7) Test Results 🕒 200 OK • 4 ms • 400 B • 🔍 Save Response ...

Raw ▼ ▶ Preview 📊 Visualize ▼ 🔍 🔗

1 Hello from MiddlewareSandbox!
2 The amount of processed requests is 8.

GET ▼ | https://localhost:44315/test?custom=true Send ▼

Params ● Authorization Headers (7) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	Key	Value	Description		
<input checked="" type="checkbox"/>	custom	true			
	Key	Value	Description		

Body Cookies Headers (4) Test Results 🕒 200 OK • 4 ms • 207 B • 🔍 Save Response ...

Raw ▼ ▶ Preview 📊 Visualize ▼ 🔍 🔗

1 You've hit a custom middleware!
2 The amount of processed requests is 9.

Рис. 3.3. – 3.5 Перевірка запитів у Postman

Пояснення

1. **Middleware 1** – рахує всі запити і додає у відповідь The amount of processed requests is X.
2. **Middleware 2** – перевіряє ?custom= у URL, якщо є – повертає повідомлення, інакше пропускає далі.
3. **Middleware 3** – логування всіх запитів у консоль: метод та шлях.
4. **Middleware 4** – перевірка заголовка X-API-KEY; якщо ключ неправильний або відсутній → 403 Forbidden.

Висновок: У ході лабораторної роботи я ознайомила з основними принципами роботи платформи .NET, навчилася налаштовувати середовище розробки та встановлювати необхідні компоненти для створення проєктів. Було опановано створення різних типів проєктів: консольного додатку, веб-додатку, ASP.NET WebAPI та MVC. Також отримано практичні навички обробки запитів з використанням middleware, включно з підрахунком запитів, логуванням, перевіркою параметрів рядка запиту та контролем доступу через API-ключ. Лабораторна робота дозволила закріпити теоретичні знання і застосувати їх на практиці для створення функціональних веб-додатків.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.16.000 – Лр.1	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		