

ЛАБОРАТОРНА РОБОТА № 1

Початок роботи з ASP.NET Core 5. Middleware

Мета роботи: ознайомитися зі структурою ASP.NET Code додатків, навчитися створювати найпростіші та використовувати наявні компоненти middleware у ASP.NET Core

Завдання на лабораторну роботу

Задання 1.

- відеозаписи лекцій №01-02:
 - ASP.NET Core 5: лекція 1 (частина 1)
 - ASP.NET Core 5: лекція 1 (частина 2)
- <https://metanit.com/sharp/aspnet5/1.1.php>
- <https://metanit.com/sharp/aspnet5/1.2.php>
- <https://metanit.com/sharp/aspnet5/1.3.php>
- <https://metanit.com/sharp/aspnet5/2.13.php>
- <https://metanit.com/sharp/aspnet5/2.1.php>
- <https://metanit.com/sharp/aspnet5/2.2.php>
- <https://metanit.com/sharp/aspnet5/2.3.php>
- <https://metanit.com/sharp/aspnet5/2.22.php>
- <https://metanit.com/sharp/aspnet5/2.4.php>
- <https://metanit.com/sharp/aspnet5/2.18.php>
- <https://metanit.com/sharp/aspnet5/2.21.php>
- <https://metanit.com/sharp/aspnet5/2.5.php>
- <https://metanit.com/sharp/aspnet5/2.14.php>
- <https://metanit.com/sharp/aspnet5/17.1.php>
- <https://metanit.com/sharp/aspnet5/18.6.php>

Рис 1. Опрацювання теоретичного матеріалу

					ДУ «Житомирська політехніка».22.126.4.000 - Лр1							
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Козак Ю.О			Звіт з лабораторної роботи				Літ.	Арк.	Аркушів	
Перевір.		Чижмотря О.В									1	ZZ
Керівник									ФІКТ Гр. ICT-20-1			
Н. контр.												
Зав. каф.												

Завдання 2. Створення нового проекту

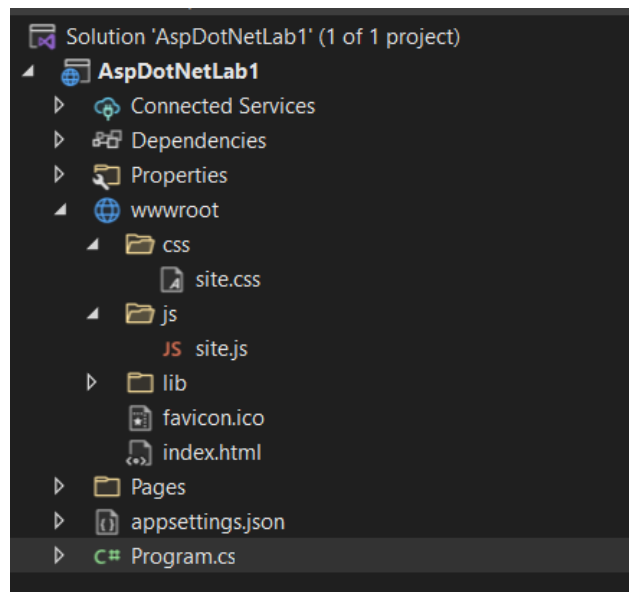


Рис 2. Структура проекту

Завдання 3. Реалізація статичного сервера

3. Реалізація статичного сервера.

Статичний сервер дозволяє передавати у браузер файли, які зберігаються у відповідній папці на сервері.

Розміщуємо у папці `wwwroot` декілька статичних файлів з довільним контентом, наприклад файл `index.html` та таблицю стилів, яка у ньому підключена:

Відкриваємо файл **Startup.cs**, знаходимо метод **Configure**.

Відмітимо, що у методі **Configure** вже буде програмний код, який був згенерований автоматично при створенні проекту:

Запускаємо проект і дивимось, що виводиться при зверненні до кореня сайту `/` та до сторінки `/index.html`:

```
// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseDeveloperExceptionPage();
    // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}

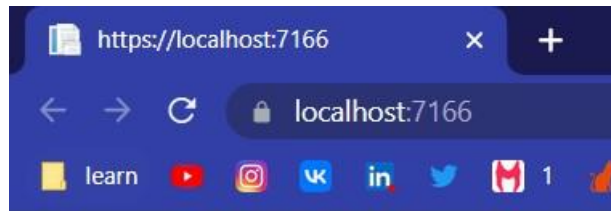
app.UseHttpsRedirection();
app.UseStaticFiles();
```

Рис 3. Підключення обробки запитів у режимі відлагодження

```
app.UseEndpoints(endpoints =>
{
    endpoints.MapGet("/", async context =>
    {
        await context.Response.WriteAsync("Hello World!");
    });
});
```

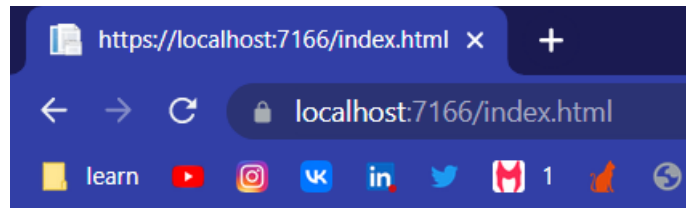
Рис 4. Підключення ендпоінтів

		Козак Ю.О.			ДУ «Житомирська політехніка».22.126.4.000 - Лр1	Арк.
		Чижмотря О.В				2
Змн.	Арк.	№ докум.	Підпис	Дата		



Hello World!

Рис 5. Дефолтна сторінка

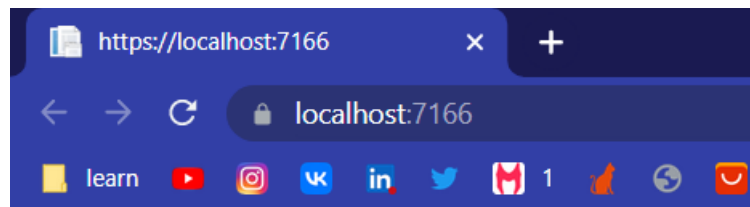


Header from index.html

Рис 6. Вигляд переходу за посиланням до статичного серверу

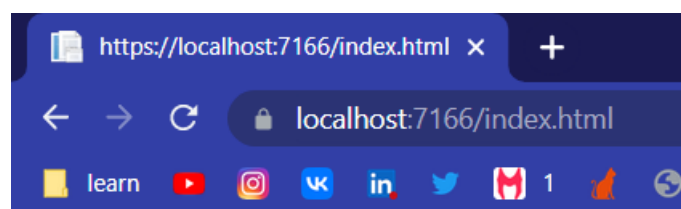
```
app.UseFileServer();
app.UseRouting();
```

Рис 7. Зміна порядку



Header from index.html

Рис 8. Зникнення дефолтної сторінки



Header from index.html

Рис 9. До запиту до статичного серверу змін не відбулося

		Козак Ю.О.			ДУ «Житомирська політехніка».22.126.4.000 - Лр1	Арк.
		Чижмоторя О.В				3
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаємо ще один статичний сервер до проекту.
Для цього створимо ще одну папку з назвою **“static”**, у ній - підкаталог **“files”**, у ньому - файл **page.html**:

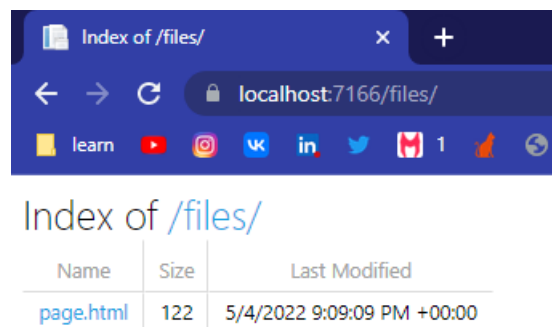


Рис 10. Вигляд переліку файлів у статичній директорії

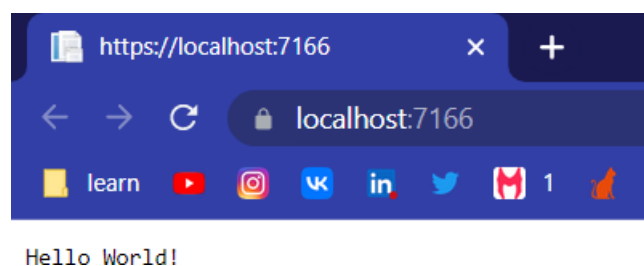


Рис 11. Вигляд після запиту до сервера

Завдання 4. Реалізація обробника помилки

```
@page
@model ErrorModel
@{
    ViewData["Title"] = "Error";
}

<h1 class="text-danger">Error.</h1>
<h2 class="text-danger">An error occurred while processing your request.</h2>

@if (Model.ShowRequestId)
{
    <p>
        <strong>Request ID:</strong> <code>@Model.RequestId</code>
    </p>
}

<h3>Development Mode</h3>
<p>
    Swapping to the <strong>Development</strong> environment displays detailed information about the error that occurred.
</p>
<p>
    <strong>The Development environment shouldn't be enabled for deployed applications.</strong>
    It can result in displaying sensitive information from exceptions to end users.
    For local debugging, enable the <strong>Development</strong> environment by setting the <strong>ASPNETCORE_ENVIRONMENT</strong> environment variable to <strong>Development</strong> and restarting the app.
</p>
```

Завдання 5. Обробка маршрутів

```
app.UseRouting();
app.UseFileServer();

app.UseEndpoints(endpoints =>
{
    endpoints.MapGet("/", async context =>
    {
        await context.Response.WriteAsync("Hello World!");
    });
    endpoints.MapGet("/home/index", async context =>
    {
        await context.Response.WriteAsync("Hello in home's index file");
    });
    endpoints.MapGet("/home/about", async context =>
    {
        await context.Response.WriteAsync("Hello in home's about file\nNow it clear");
    });
});

app.UseFileServer(new FileServerOptions
{
    EnableDirectoryBrowsing = true,
    FileProvider = new PhysicalFileProvider(Path.Combine(Directory.GetCurrentDirectory(),
@"static"))
});
```

Висновок: Сьогодні на лабораторній роботі ми ознайомитися зі структурою ASP.NET Code додатків, також навчилися створювати найпростіші та використовувати наявні компоненти middleware у ASP.NET Core

		Козак Ю.О.			ДУ «Житомирська політехніка».22.126.4.000 - Лр1	Арк.
		Чижевотря О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		5