МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА МЕРЕЖЕВИХ ТА ІНТЕРНЕТ ТЕХНОЛОГІЙ

Лабораторне заняття №5

Тема: ГЛОБАЛІЗАЦІЯ ТА ЛОКАЛІЗАЦІЯ ЗАСТОСУНКУ ASP.NET CORE

Виконав студент групи: МІТ-41

Кухарчук Богдан Петрович

Хід роботи

1. Налаштування сервісів локалізації та Middleware

Для забезпечення підтримки локалізації у файл Program.cs було додано реєстрацію відповідних сервісів: AddLocalization (із вказанням шляху до ресурсів), AddViewLocalization та AddDataAnnotationsLocalization. Також було налаштовано конвеєр (middleware) через UseRequestLocalization, де визначено список підтримуваних культур (en-US, uk-UA) та культуру за замовчуванням.

```
// НАЛАШТУВАННЯ ЛОКАЛІЗАЦІЇ (ЛР 5)

// 1. Реєстрація сервісів локалізації
builder.Services.AddLocalization(options => options.ResourcesPath = "Resources");

// 2. Налаштування списку культур для сервісів (щоб працював SelectLanguagePartial)
builder.Services.Configure<RequestLocalizationOptions>(options =>
{
    var supportedCultures = new[]
    {
        new CultureInfo("en-US"), // Англійська (США)
        new CultureInfo("uk-UA"), // Українська
        new CultureInfo("es")     // Іспанська
    };

    options.DefaultRequestCulture = new RequestCulture("en-US");
    options.SupportedCultures = supportedCultures;
    options.SupportedUICultures = supportedCultures;
});
```

Рисунок 5.1 Реєстрація сервісів та налаштування Middleware локалізації у Program.cs.

2. Створення файлів ресурсів (.resx)

У проєкті було створено структуру папок Resources/Controllers та Resources/Views/Home для зберігання файлів перекладів. Створено файли ресурсів для контролера HomeController та представлення Index.cshtml для двох мов: англійської (.en-US.resx) та української (.uk-UA.resx). У ці файли додано пари "ключ-значення" для перекладу текстових повідомлень.

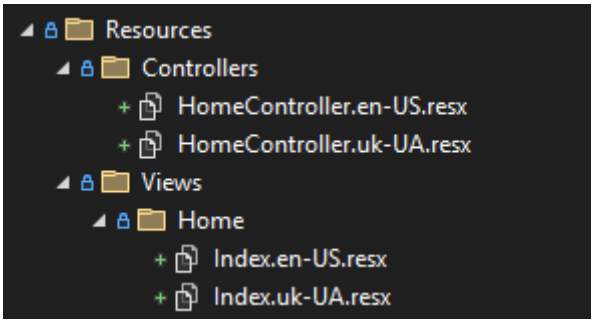

Рисунок 5.2. Структура папок та файлів ресурсів у проєкті.

Name	uk-UA (Ukrainian (Ukraine))	en-US (English (United States))
HeaderTitle	Ласкаво просимо!	Welcome!

Рисунок 5.3. Приклад наповнення файлу ресурсів для української та англійської мови.

3. Локалізація контролера

У контролері HomeController було використано сервіс IStringLocalizer<HomeController>. Він інжектується через конструктор і використовується для отримання локалізованих рядків з файлів ресурсів (наприклад, повідомлення "WelcomeMessage"). Отримане значення передається у представлення через ViewData.

```
//  додано параметр MyConfiguration у конструктор
public HomeController(ILogger<HomeController> logger, IWebRepository repository, MyConfiguration config, IStringLocalizer<HomeController> localizer)
{
    _logger = logger;
    _repository = repository;
    _config = config;
    _localizer = localizer; // [ЛАБ 5] Зберігаємо
}
[AllowAnonymous]
public async Task<IActionResult> IndexAsync()
{
    // [ЛАБ 5] Використовуємо переклад
    ViewData["MessageFromController"] = _localizer["WelcomeMessage"];

    var users = await _repository.ReadAll<WebApplicationUser>().ToListAsync();
```

Рисунок 5.4. Використання IStringLocalizer у контролері для отримання локалізованого рядка.

4. Локалізація представлення (View) та перевірка форматів

На головній сторінці Index.cshtml було використано сервіс IViewLocalizer для перекладу заголовка сторінки. Також на сторінку додано блок для демонстрації форматування даних (дати та валюти), які автоматично змінюються залежно від обраної культури.

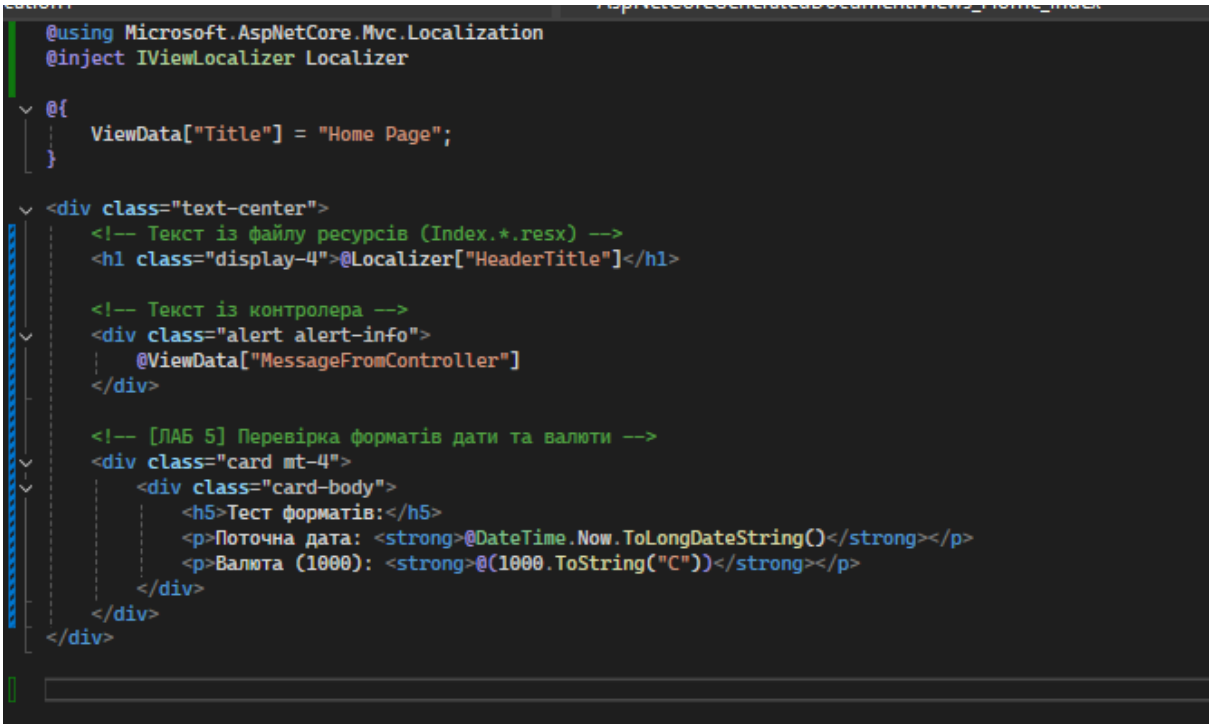

Рисунок 5.5. Використання IViewLocalizer та виведення форматованих даних на сторінці.

5. Реалізація перемикання мов

Для зміни мови інтерфейсу користувачем було створено спеціальний контролер SelectLanguageController, який встановлює cookie з обраною культурою. Також створено часткове представлення _SelectLanguagePartial.cshtml, яке відображає випадаючий список доступних мов, та інтегровано його в навігаційну панель (_Layout.cshtml).

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

namespace WebApplication1.Controllers
{
    public class SelectLanguageController : Controller
    {
        [HttpPost]
        public IActionResult SetLanguage(string culture, string returnUrl)
        {
            // Зберігаємо вибір мови в Cookie
            Response.Cookies.Append(
                CookieRequestCultureProvider.DefaultCookieName,
                CookieRequestCultureProvider.MakeCookieValue(new RequestCulture(culture)),
                new CookieOptions { Expires = DateTimeOffset.UtcNow.AddYears(1) }
            );

            // Повертаємо користувача назад
            return LocalRedirect(returnUrl);
        }
    }
}
```

Рисунок 5.6. Контролер для обробки зміни мови та встановлення cookie.

```
@using Microsoft.AspNetCore.Builder
@using Microsoft.AspNetCore.Localization
@using Microsoft.AspNetCore.Mvc.Localization
@using Microsoft.Extensions.Options

@inject IViewLocalizer Localizer
@inject IOptions<RequestLocalizationOptions> LocOptions

@{
    var requestCulture = Context.Features.Get<IRequestCultureFeature>();
    var cultureItems = LocOptions.Value.SupportedUICultures
        .Select(c => new SelectListItem { Value = c.Name, Text = c.NativeName })
        .ToList();
    var returnUrl = string.IsNullOrEmpty(Context.Request.Path) ? "~/" : $"~{Context.Request.Path.Value}";
}

<div class="mx-3">
    <form id="selectLanguage" asp-controller="SelectLanguage" asp-action="SetLanguage"
          asp-route-returnUrl="@returnUrl" method="post">
        <select name="culture"
                onchange="this.form.submit();"
                asp-for="@requestCulture.RequestCulture.UICulture.Name"
                asp-items="cultureItems"
                class="form-select form-select-sm">
        </select>
    </form>
</div>
```

Рисунок 5.7. Часткове представлення з формою вибору мови.

Результат роботи

Продемонстровано роботу застосунку з різними локалями. При виборі англійської мови (English) інтерфейс відображається англійською, дата — у форматі США (місяць/день/рік), валюта — у доларах ($).

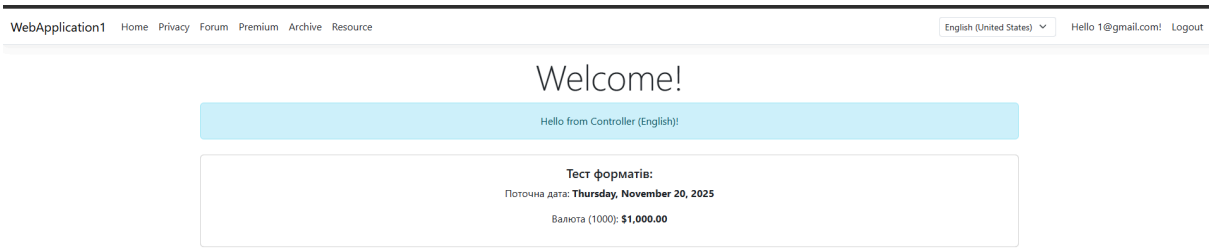

Рисунок 5.8. Відображення сайту з локалізацією en-US (англійська).

При перемиканні на **українську мову**, текст змінюється на український, дата відображається у звичному форматі (день.місяць.рік), а валюта — у гривнях (₴).

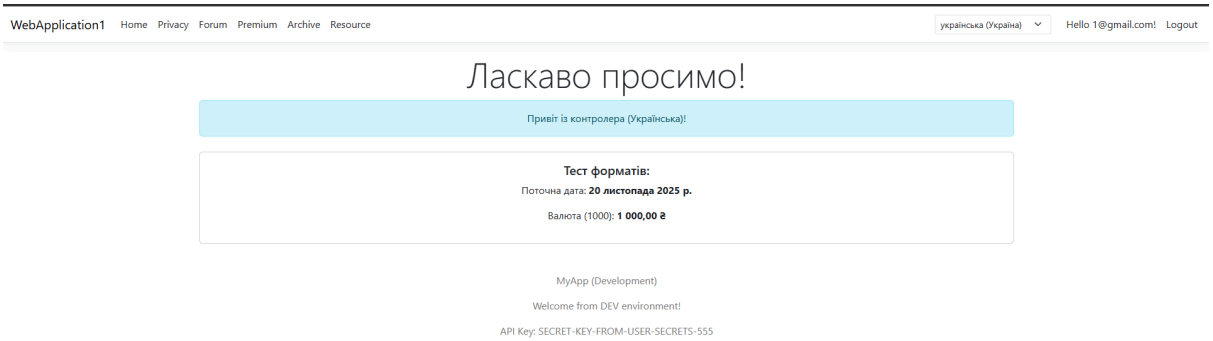

Рисунок 5.9. Відображення сайту з локалізацією uk-UA (українська).

Висновок

У ході виконання лабораторної роботи було реалізовано підтримку багатомовності та культурних особливостей у веб-застосунку ASP.NET Core. Було проведено комплексне налаштування інфраструктури локалізації, що включало реєстрацію необхідних сервісів та конфігурацію конвеєра обробки запитів для автоматичного визначення поточної культури. Для перекладу текстового вмісту було створено відповідні файли ресурсів (.resx) та використано механізми типізованої локалізації як на рівні контролерів, так і на рівні представлень Razor. Додатково

розроблено зручний функціонал для ручного перемикання мови користувачем, що базується на збереженні налаштувань у cookies, а також забезпечено автоматичну адаптацію форматів дат, чисел та валют відповідно до регіональних стандартів. Усі зміни зафіксовано у віддаленому репозиторії GitHub, що підтверджує готовність проєкту до використання міжнародною аудиторією.