

ЛАБОРАТОРНА РОБОТА № 2

Робота з ASP.NET MVC. Конфігурація проекту.

Відображення даних з БД на веб-сторінках.

Мета: Ознайомитися з архітектурою MVC, набути навичок створення головних компонентів архітектури MVC, навчитися конфігурувати підключення до БД та виконувати запити на читання даних, набути навичок роботи з представленнями (Views).

Хід роботи

Завдання 1. Створення та конфігурація проекту.

Створити проект, який реалізує архітектуру MVC, слідуючи інструкціям.

Тема проекту – «Веб-сайт для бронювання квитків на заходи».

Короткий опис функціоналу: Користувачі можуть шукати події, бронювати квитки та керувати бронюванням, а адміністратори можуть додавати та змінювати події.

На першому етапі було створено проєкт ASP.NET MVC для реалізації веб-сайту бронювання квитків на заходи. Для забезпечення однакової версії середовища розробки використано файл global.json, у якому зафіксовано версію .NET SDK 10.0. Це дозволяє уникнути проблем сумісності під час запуску проєкту на різних комп'ютерах.

```
D:\Навчання\ASP.NET\Labs\lab2>dotnet new globaljson --sdk-version 10.0 --output TicketBookingSln/TicketBooking.Web
The template "global.json file" was created successfully.
```

Рис. 1.1. Створення файлу global.json з фіксацією версії .NET SDK.

					ДУ «Житомирська політехніка».25.121.1.000 – Лр2								
Змн.	Арк.	№ докум.	Підпис	Дата									
Розроб.		Андріюк Д. Р.			Звіт з лабораторної роботи				Літ.	Арк.	Аркушів		
Перевір.		Українець М.О.									1	17	
Керівник									ФІКТ Гр. ІПЗ-22-2[1]				
Н. контр.													
Зав. каф.													

Проект було створено за допомогою шаблону ASP.NET Core Empty з подальшим перетворенням на MVC-додаток. Для цього було створено solution та додано до нього веб-проект. Також у файлі launchSettings.json налаштовано HTTP-порт для локального запуску додатку.

Лістинг файлу launchSettings.json:

```
{
  "$schema": "https://json.schemastore.org/launchsettings.json",
  "profiles": {
    "http": {
      "commandName": "Project",
      "dotnetRunMessages": true,
      "launchBrowser": true,
      "applicationUrl": "http://localhost:5000",
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    }
  }
}
```

```
D:\Навчання\ASP.NET\Labs\lab2>dotnet new web --no-https --output TicketBookingSln
TicketBooking --framework net10.0
The template "ASP.NET Core Empty" was created successfully.

Processing post-creation actions...
Restoring D:\Навчання\ASP.NET\Labs\lab2\TicketBookingSln\TicketBooking\TicketBo
oking.csproj:
Restore succeeded.
```

Рис. 1.2. Створення додатку.

```
D:\Навчання\ASP.NET\Labs\lab2>dotnet new sln -o TicketBookingSln
The template "Solution File" was created successfully.
```

Рис. 1.3. Створення проекту.

```
D:\Навчання\ASP.NET\Labs\lab2>dotnet sln TicketBookingSln add TicketBookingSln/
TicketBooking.Web
Project 'TicketBooking.Web\TicketBooking.Web.csproj' added to the solution.
```

Рис. 1.4. Додавання додатку в проект.

		Андріюк Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр2	Арк.
		Українець М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

У структурі проєкту створено папки Models, Controllers, Views, Views/Home, Views/Shared, що відповідає класичній архітектурі MVC. У файлі Program.cs виконано налаштування сервісів MVC та маршрутизації за замовчуванням. Додатково налаштовано рушій представлень Razor шляхом створення файлів _ViewImports.cshtml, _ViewStart.cshtml та базового шаблону _Layout.cshtml.

Далі було налаштовано файл Program.cs, у якому зареєстровано сервіси MVC та стандартну маршрутизацію. Також налаштовано рушій представлень Razor шляхом створення файлів _ViewImports.cshtml, _ViewStart.cshtml та базового шаблону _Layout.cshtml.

Лістинг файлу Program.cs:

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

var app = builder.Build();
app.UseStaticFiles();
app.MapDefaultControllerRoute();
app.Run();
```

Лістинг файлу _ViewImports.cshtml:

```
@using TicketBooking.Models
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
```

Лістинг файлу _ViewStart.cshtml:

```
@{
    Layout = "_Layout";
}
```

Лістинг файлу _Layout.cshtml:

```
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Ticket Booking</title>
</head>
<body>
    <div>
        @RenderBody()
    </div>
</body>
</html>
```

		Андріюк Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр2	Арк.
		Українець М.О.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

Для перевірки працездатності додатку створено контролер HomeController з методом Index та відповідне представлення Index.cshtml, яке відображає стартову сторінку.

Лістинг контролеру HomeController:

```
using Microsoft.AspNetCore.Mvc;

namespace TicketBooking.Controllers {
    public class HomeController : Controller {
        public IActionResult Index() => View();
    }
}
```

Лістинг файлу Index.cshtml:

```
<h4>Welcome to Ticket Booking</h4>
<p>Initial page for Ticket Booking</p>
```

Лістинг моделі Event:

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace TicketBooking.Models {
    public class Event {
        public long? EventId { get; set; }

        public string Title { get; set; } = string.Empty;
        public string Description { get; set; } = string.Empty;

        public DateTime StartDate { get; set; }

        public string Location { get; set; } = string.Empty;

        [Range(0, double.MaxValue, ErrorMessage = "Ticket's price can't be negative")]
        [Column(TypeName = "decimal(10, 2)")] // 99999999.99 max value
        public decimal BaseTicketPrice { get; set; }
    }
}
```

Після запуску проєкту у браузері відображається стартова сторінка веб-додатку.

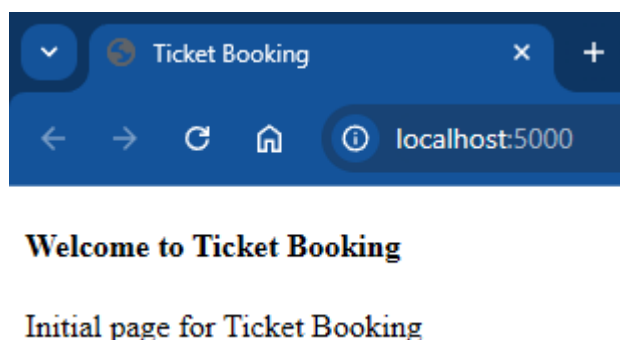


Рис. 1.5. Початкова сторінка веб-додатку Ticket Booking.

		Андріюк Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр2	Арк.
		Українець М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

Завдання 2. Робота з ORM. Налаштувати роботу з БД.

На другому етапі було реалізовано роботу з базою даних за допомогою ORM Entity Framework Core. Для цього попередньо перевірено наявність SQL Server LocalDB у системі.

```
D:\Навчання\ASP.NET\Labs\lab2>SqlLocalDB
Microsoft (R) SQL Server Express LocalDB Command Line Tool
Version 15.0.4382.1
Copyright (c) Microsoft Corporation. All rights reserved.

Usage: SqlLocalDB operation [parameters...]
```

Рис. 2.1. Перевірка наявності SQL Server LocalDB.

Для роботи з базою даних встановлено необхідні пакети Entity Framework Core для SQL Server, а також інструмент командного рядка dotnet-ef для керування міграціями.

```
D:\Навчання\ASP.NET\Labs\lab2\TicketBookingSln\TicketBooking>dotnet add package
Microsoft.EntityFrameworkCore.Design
info : X.509 certificate chain validation will use the default trust store sele
cted by .NET for code signing.
info : X.509 certificate chain validation will use the default trust store sele
cted by .NET for timestamping.
info : Adding PackageReference for package 'Microsoft.EntityFrameworkCore.Desig
n' into project 'D:\Навчання\ASP.NET\Labs\lab2\TicketBookingSln\TicketBooking\T
icketBooking.csproj'.
info :   GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entity
frameworkcore.design/index.json
info :   OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityf
rameworkcore.design/index.json 196ms
info :   GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entity
frameworkcore.design/page/0.0.1-alpha/3.1.3.json
```

Рис. 2.2. Встановлення пакету Entity Framework Core Design.

		Андрійук Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр2	Арк.
		Українець М.О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		


```
D:\Навчання\ASP.NET\Labs\lab2\TicketBookingSln\TicketBooking>dotnet add package
Microsoft.EntityFrameworkCore.SqlServer
info : X.509 certificate chain validation will use the default trust store sele
cted by .NET for code signing.
info : X.509 certificate chain validation will use the default trust store sele
cted by .NET for timestamping.
info : Adding PackageReference for package 'Microsoft.EntityFrameworkCore.SqlSe
rver' into project 'D:\Навчання\ASP.NET\Labs\lab2\TicketBookingSln\TicketBookin
g\TicketBooking.csproj'.
info :   GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entity
frameworkcore.sqlserver/index.json
info :   OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityf
rameworkcore.sqlserver/index.json 317ms
info :   GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entity
frameworkcore.sqlserver/page/0.0.1-alpha/3.1.2.json
```

Рис. 2.3. Встановлення пакету Entity Framework Core SqlServer.

```
D:\Навчання\ASP.NET\Labs\lab2\TicketBookingSln\TicketBooking>dotnet add package
Microsoft.EntityFrameworkCore.Tools
info : X.509 certificate chain validation will use the default trust store sele
cted by .NET for code signing.
info : X.509 certificate chain validation will use the default trust store sele
cted by .NET for timestamping.
info : Adding PackageReference for package 'Microsoft.EntityFrameworkCore.Tools
' into project 'D:\Навчання\ASP.NET\Labs\lab2\TicketBookingSln\TicketBooking\Ti
cketBooking.csproj'.
info :   GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entity
frameworkcore.tools/index.json
info :   OK https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entityf
rameworkcore.tools/index.json 370ms
info :   GET https://api.nuget.org/v3/registration5-gz-semver2/microsoft.entity
frameworkcore.tools/page/1.0.0-msbuild1-final/3.1.0.json
```

Рис. 2.4. Встановлення пакету Entity Framework Core Tools.

```
D:\Навчання\ASP.NET\Labs\lab2\TicketBookingSln>dotnet tool install --global dotnet-ef
You can invoke the tool using the following command: dotnet-ef
Tool 'dotnet-ef' (version '10.0.2') was successfully installed.

D:\Навчання\ASP.NET\Labs\lab2\TicketBookingSln>dotnet ef --version
Entity Framework Core .NET Command-line Tools
10.0.2
```

Рис. 2.5.1 Встановлення CLI-інструментів.

```
D:\Навчання\ASP.NET\Labs\lab2\TicketBookingSln\TicketBooking>dotnet ef migrations add Initial
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'
```

Рис. 2.5.2 Встановлення CLI-інструментів.

		Андріюк Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр2	Арк.
		Українець М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

У файлі appsettings.json визначено рядок підключення до бази даних LocalDB з назвою TicketBookingDb.

Лістинг файлу конфігурації appsettings.json:

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "TicketBookingConnection":
"Server=(localdb)\\MSSQLLocalDB;Database=TicketBookingDb;MultipleActiveResultSets=true"
  }
}
```

Для доступу до даних створено клас контексту TicketBookingDbContext, який наслідує DbContext та містить властивість DbSet<Event> для роботи з подіями.

Лістинг класу TicketBookingDbContext:

```
using Microsoft.EntityFrameworkCore;

namespace TicketBooking.Models {
    public class TicketBookingDbContext : DbContext {
        public TicketBookingDbContext(DbContextOptions<TicketBookingDbContext> options)
            : base(options) { }

        public DbSet<Event> Events => Set<Event>();
    }
}
```

З метою відокремлення логіки доступу до даних реалізовано патерн репозиторію. Для цього створено інтерфейс IEventRepository та його реалізацію EFEventRepository.

Лістинг інтерфейсу IEventRepository:

```
namespace TicketBooking.Models {
    public interface IEventRepository {
        IQueryable<Event> Events { get; }
    }
}
```

Лістинг класу EFEventRepository:

```
namespace TicketBooking.Models {
    public class EFEventRepository : IEventRepository {
        private readonly TicketBookingDbContext _context;
```

		Андріюк Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр2	Арк.
		Українець М.О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        public EFEventRepository(TicketBookingDbContext context) {
            _context = context;
        }

        public IQueryable<Event> Events => _context.Events;
    }
}

```

Для автоматичного створення та наповнення бази даних реалізовано клас SeedData, який застосовує міграції та додає початкові записи подій.

Лістинг класу SeedData:

```

using Microsoft.EntityFrameworkCore;

namespace TicketBooking.Models {
    public class SeedData {
        public static void EnsurePopulated(IApplicationBuilder app) {
            using var scope = app.ApplicationServices.CreateScope();
            var context =
scope.ServiceProvider.GetRequiredService<TicketBookingDbContext>();

            if (context.Database.GetPendingMigrations().Any()) {
                context.Database.Migrate();
            }

            if (!context.Events.Any()) {
                context.Events.AddRange(
                    new Event {
                        Title = "Rock Concert",
                        Description = "Live rock concert in the city center",
                        Location = "Kyiv, Palace of Sports",
                        StartDate = DateTime.Today.AddDays(7).AddHours(19),
                        BaseTicketPrice = 650m
                    },
                    new Event {
                        Title = "Stand-up Night",
                        Description = "Comedy evening with popular comedians",
                        Location = "Lviv, Downtown Club",
                        StartDate = DateTime.Today.AddDays(10).AddHours(20),
                        BaseTicketPrice = 400m
                    },
                    new Event {
                        Title = "Tech Conference",
                        Description = "Talks about .NET, Cloud, and AI",
                        Location = "Odesa, Expo Center",
                        StartDate = DateTime.Today.AddDays(21).AddHours(9),
                        BaseTicketPrice = 1200m
                    },
                    new Event {
                        Title = "Theater Premiere",
                        Description = "A new performance premiere",
                        Location = "Kharkiv, Drama Theater",
                        StartDate = DateTime.Today.AddDays(14).AddHours(18),
                        BaseTicketPrice = 500m
                    }
                );

                context.SaveChanges();
            }
        }
    }
}

```

		Андріюк Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр2	Арк.
		Українець М.О.				8
Змн.	Арк.	№ докум.	Підпис	Дата		


```

    }
}
}

```

Після реалізації моделі даних створено початкову міграцію та виконано оновлення бази даних.

```

D:\Навчання\ASP.NET\Labs\lab2\TicketBookingSln\TicketBooking>dotnet ef database update
Build started...
Build succeeded.
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (217ms) [Parameters=[], CommandType='Text', CommandTimeout='60']
      CREATE DATABASE [TicketBookingDb];
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (53ms) [Parameters=[], CommandType='Text', CommandTimeout='60']
      IF SERVERPROPERTY('EngineEdition') <> 5
      BEGIN
        ALTER DATABASE [TicketBookingDb] SET READ_COMMITTED_SNAPSHOT ON;
      END;
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (5ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
      SELECT 1
info: Microsoft.EntityFrameworkCore.Migrations[20411]
      Acquiring an exclusive lock for migration application. See https://aka.ms/efcore-docs-migrations-lock for more information if this takes too long.
      Acquiring an exclusive lock for migration application. See https://aka.ms/efcore-docs-migrations-lock for more information if this takes too long.
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (29ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
      DECLARE @result int;
      EXEC @result = sp_getapplock @Resource = '__EFMigrationsLock', @LockOwner = 'Session', @LockMode = 'Exclusive';
      SELECT @result
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (7ms) [Parameters=[], CommandType='Text', CommandTimeout='30']

```

Рис. 2.6. Створення та застосування міграцій Entity Framework Core.

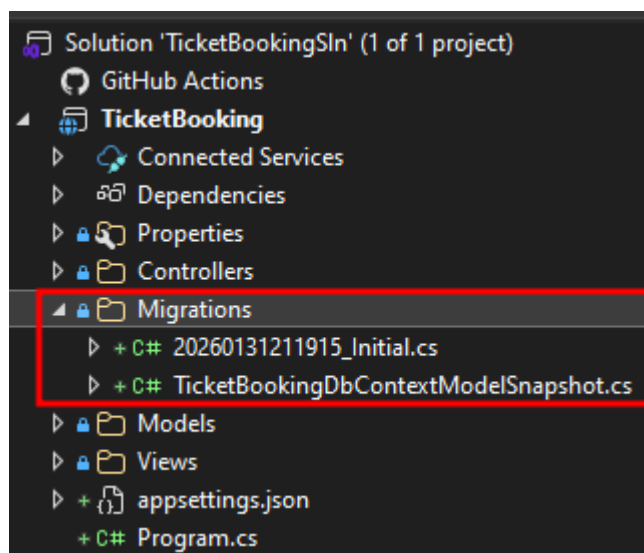


Рис. 2.7. Структура міграцій Entity Framework Core.

		Андріюк Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр2	Арк.
		Українець М.О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг файлу Program.cs:

```
using Microsoft.EntityFrameworkCore;
using TicketBooking.Models;

var builder = WebApplication.CreateBuilder(args);
builder.Services.AddControllersWithViews();

builder.Services.AddDbContext<TicketBookingDbContext>(opts => {
    opts.UseSqlServer(builder.Configuration["ConnectionStrings:TicketBookingConnection"]);
});

builder.Services.AddScoped<IEventRepository, EFEventRepository>();

var app = builder.Build();
app.UseStaticFiles();
app.MapDefaultControllerRoute();
SeedData.EnsurePopulated(app);
app.Run();
```

Завдання 3. Виведення даних на сторінці.

Змініть контролер представлення для виведення даних з БД.

На цьому етапі реалізовано виведення даних з бази даних на веб-сторінці. Контролер HomeController було змінено таким чином, щоб він отримував доступ до репозиторію подій через dependency injection та передавав колекцію подій у представлення.

Лістинг оновлення контролеру HomeController:

```
using Microsoft.AspNetCore.Mvc;
using TicketBooking.Models;

namespace TicketBooking.Controllers {
    public class HomeController : Controller {
        private readonly IEventRepository _repository;

        public HomeController(IEventRepository repository) {
            _repository = repository;
        }

        public IActionResult Index() => View(_repository.Events);
    }
}
```

У представленні Index.cshtml реалізовано перебір отриманої колекції подій та виведення інформації про кожну подію: назви, опису, місця проведення, дати та вартості квитків.

		Андріюк Д. Р.			ДУ «Житомирська політехніка». 25.121.1.000 – Лр2	Арк.
		Українець М.О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг оновлення файлу Index.cshtml:

```
@model IQueryable<Event>

@foreach (var e in Model ?? Enumerable.Empty<Event>()) {
    <div style="margin-bottom: 16px; padding-bottom: 12px; border-bottom: 1px solid #ccc;">
        <h3>@e.Title</h3>
        <div>@e.Description</div>

        <div><b>Location:</b> @e.Location</div>
        <div><b>Date:</b> @e.StartDate.ToString("yyyy-MM-dd HH:mm")</div>

        <h4>@e.BaseTicketPrice.ToString("c")</h4>
    </div>
}
```

У результаті запуску додатку у браузері відображається список подій, отриманих з бази даних.

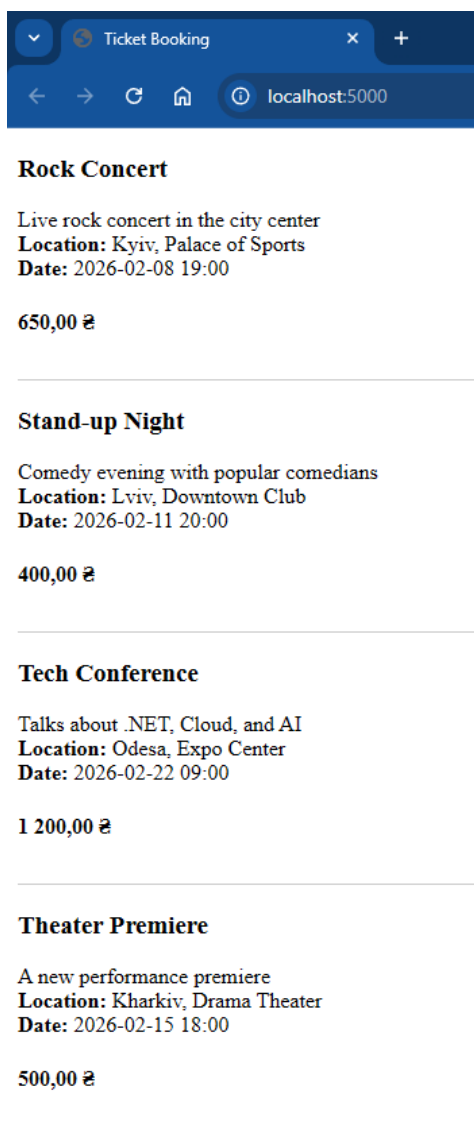


Рис. 3. Виведення даних з бази даних на веб-сторінці.

		Андріюк Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр2	Арк.
		Українець М.О.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 4. Реалізуйте посторінковий вивід записів.

Для підвищення зручності користування додатком реалізовано посторінковий вивід записів. Для цього у папці Models/ViewModels створено класи PagingInfo та EventsListViewModel.

Лістинг класу PagingInfo:

```
namespace TicketBooking.Models.ViewModels {
    public class PagingInfo {
        public int TotalItems { get; set; }
        public int ItemsPerPage { get; set; }
        public int CurrentPage { get; set; }

        public int TotalPages =>
            (int)Math.Ceiling((decimal)TotalItems / ItemsPerPage);
    }
}
```

Лістинг класу EventsListViewModel:

```
namespace TicketBooking.Models.ViewModels {
    public class EventsListViewModel {
        public IEnumerable<Event> Events { get; set; } = Enumerable.Empty<Event>();
        public PagingInfo PagingInfo { get; set; } = new PagingInfo();
    }
}
```

Метод Index контролера було модифіковано: тепер він приймає номер поточної сторінки та формує запит до бази даних з використанням методів Skip та Take.

Лістинг оновлення контролеру HomeController:

```
using Microsoft.AspNetCore.Mvc;
using TicketBooking.Models;
using TicketBooking.Models.ViewModels;

namespace TicketBooking.Controllers {
    public class HomeController : Controller {
        private readonly IEventRepository _repository;
        private const int PageSize = 3;

        public HomeController(IEventRepository repository) {
            _repository = repository;
        }

        public IActionResult Index(int pageNum = 1) {
            var query = _repository.Events
                .OrderBy(e => e.EventId);

            var totalItems = query.Count();
        }
    }
}
```

		Андрійук Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр2	Арк.
		Українець М.О.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        var eventsOnPage = query
            .Skip((pageNum - 1) * PageSize)
            .Take(PageSize)
            .ToList();

        var model = new EventsListViewModel {
            Events = eventsOnPage,
            PagingInfo = new PagingInfo {
                CurrentPage = pageNum,
                ItemsPerPage = PageSize,
                TotalItems = totalItems
            }
        };

        return View(model);
    }
}

```

У представленні реалізовано елементи навігації між сторінками, що дозволяє переходити між сторінками списку подій.

Лістинг оновлення файлу Index.cshtml:

```

@model TicketBooking.Models.ViewModels.EventsListViewModel

@foreach (var e in Model.Events ?? Enumerable.Empty<Event>()) {
    <div style="margin-bottom: 16px; padding-bottom: 12px; border-bottom: 1px solid #ccc;">
        <h3>@e.Title</h3>
        <div>@e.Description</div>

        <div><b>Location:</b> @e.Location</div>
        <div><b>Date:</b> @e.StartDate.ToString("yyyy-MM-dd HH:mm")</div>

        <h4>@e.BaseTicketPrice.ToString("c")</h4>
    </div>
}

<div style="margin-top: 16px;">
    @for (int i = 1; i <= Model.PagingInfo.TotalPages; i++) {
        if (i == Model.PagingInfo.CurrentPage) {
            <span style="margin-right: 6px; font-weight: bold;">@i</span>
        } else {
            <a style="margin-right: 6px;"
                asp-controller="Home"
                asp-action="Index"
                asp-route-pageNum="@i">@i</a>
        }
    }
</div>

```




Рис. 4.1. Посторінковий вивід записів у веб-додатку. Перша сторінка

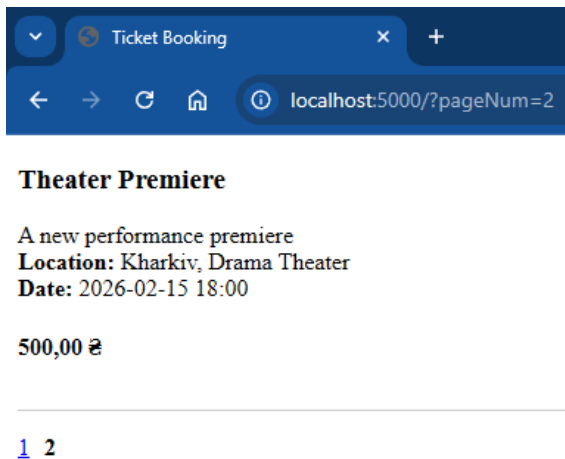


Рис. 4.2. Посторінковий вивід записів у веб-додатку. Наступна сторінка.

Завдання 5. Реалізуйте посторінковий вивід записів.

		Андріюк Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр2	Арк.
		Українець М.О.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

На завершальному етапі виконано стилізацію веб-додатку з використанням фреймворку Bootstrap. Для цього встановлено Library Manager (LibMan) та підключено бібліотеку Bootstrap у папку wwwroot.

```
D:\Навчання\ASP.NET\Labs\lab2\TicketBookingSln\TicketBooking>dotnet tool install --global Microsoft.Web.LibraryManager.Cli

You can invoke the tool using the following command: libman
Tool 'microsoft.web.librarymanager.cli' (version '3.0.71') was successfully installed.

D:\Навчання\ASP.NET\Labs\lab2\TicketBookingSln\TicketBooking>libman --version

3.0.71+33c04f70a4.RR

D:\Навчання\ASP.NET\Labs\lab2\TicketBookingSln\TicketBooking>libman init -p cdnjs

D:\Навчання\ASP.NET\Labs\lab2\TicketBookingSln\TicketBooking>libman install bootstrap -d wwwroot/lib/bootstrap
Downloading file https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.8/css/bootstrap-grid.css...
Downloading file https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.8/css/bootstrap-grid.css.map...
Downloading file https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.8/css/bootstrap-grid.min.css...
Downloading file https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.8/css/bootstrap-grid.min.css.map...
```

Рис. 5.1. Підключення Bootstrap за допомогою LibMan.

У базовому шаблоні _Layout.cshtml додано навігаційну панель та футер, а також підключено CSS і JavaScript файли Bootstrap.

Лістинг оновлення файлу _Layout.cshtml:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <title>Ticket Booking</title>

    <link rel="stylesheet" href="~/lib/bootstrap/css/bootstrap.min.css" />
</head>
<body>

    <header>
        <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
            <div class="container">
                <a class="navbar-brand" asp-controller="Home" asp-action="Index">
                    TicketBooking
                </a>
            </div>
        </nav>
    </header>

    <main class="container mt-4">
        @RenderBody()
    </main>

    <footer class="bg-light text-center text-muted mt-5 py-3 border-top">
        <div class="container">
            <small>
```

```

        © @DateTime.Now.Year Ticket Booking System |
        ASP.NET MVC Laboratory
    </small>
</div>
</footer>

<script src="~/lib/bootstrap/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

Список подій стилізовано з використанням Bootstrap Grid та компонентів Card, а пейджинг реалізовано за допомогою стандартного компонента pagination.

Лістинг оновлення файлу Index.cshtml:

```

@model TicketBooking.Models.ViewModels.EventsListViewModel

<div class="row">
    @foreach (var e in Model.Events ?? Enumerable.Empty<Event>()) {
        <div class="col-md-4 mb-4">
            <div class="card h-100 shadow-sm">
                <div class="card-body">
                    <h5 class="card-title">@e.Title</h5>
                    <p class="card-text">@e.Description</p>

                    <ul class="list-unstyled mb-3">
                        <li><strong>Location:</strong> @e.Location</li>
                        <li><strong>Date:</strong> @e.StartDate.ToString("yyyy-MM-dd
HH:mm")</li>
                    </ul>
                </div>

                <div class="card-footer text-end">
                    <span class="fw-bold">
                        @e.BaseTicketPrice.ToString("c")
                    </span>
                </div>
            </div>
        </div>
    }
</div>

<nav aria-label="Page navigation">
    <ul class="pagination justify-content-center">

        @for (int i = 1; i <= Model.PagingInfo.TotalPages; i++) {
            <li class="page-item @(i == Model.PagingInfo.CurrentPage ? "active" : "")">
                <a class="page-link"
                    asp-controller="Home"
                    asp-action="Index"
                    asp-route-pageNum="@i">
                    @i
                </a>
            </li>
        }

    </ul>
</nav>

```

		Андрійук Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр2	Арк.
		Українець М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		16

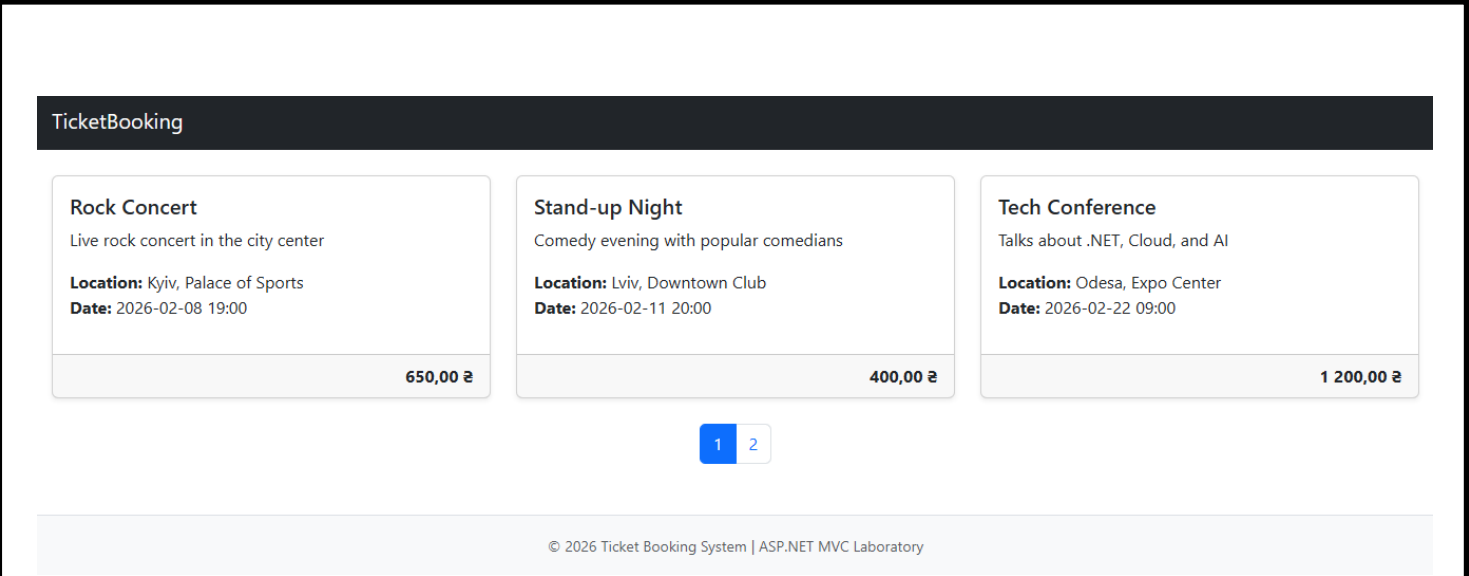


Рис. 5.1. Стилізований інтерфейс веб-додатку Ticket Booking. Перша сторінка

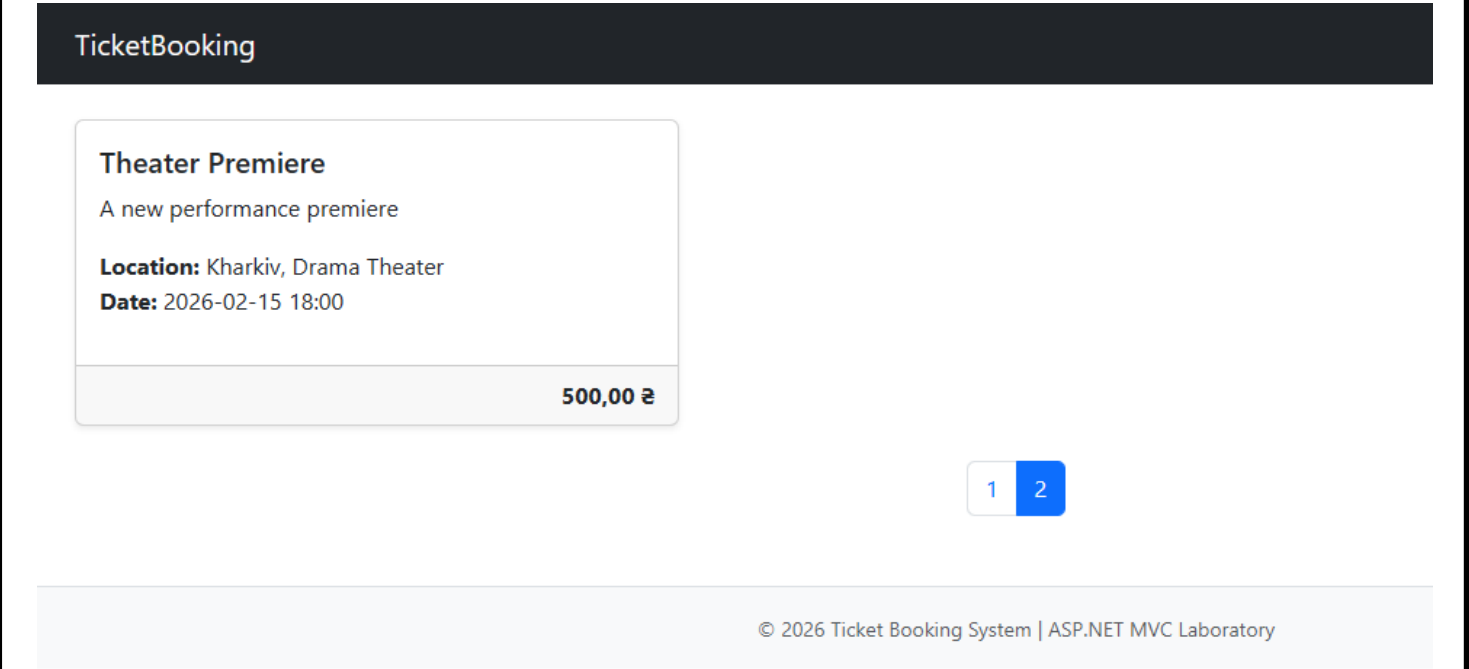


Рис. 5.2. Стилізований інтерфейс веб-додатку Ticket Booking. Друга сторінка

Висновок: У ході виконання лабораторної роботи було поетапно створено веб-додаток на базі ASP.NET MVC. Реалізовано архітектуру MVC, підключення до бази даних за допомогою Entity Framework Core, виведення та посторінкову навігацію даних, а також стилізацію інтерфейсу з використанням Bootstrap. Отримані практичні навички дозволяють застосовувати ASP.NET MVC для розробки сучасних веб-додатків з розширюваною архітектурою.