

## ЛАБОРАТОРНА РОБОТА № 3

### Часткові представлення. Робота з View Components.

### Навігація. Фільтрація. Використання сесії.

**Мета:** отримати навички роботи з частковими представленнями, навчитися працювати з параметрами рядку запиту, набути навичок роботи з сесіями.

### Хід роботи

#### Завдання 1. Створення часткового представлення (Partial View).

У рамках завдання необхідно було винести розмітку одиничного елемента списку (події) в окреме Partial View для зменшення дублювання коду та підвищення читабельності представлень. До виконання завдання HTML-розмітка картки події знаходилась безпосередньо у файлі Views/Home/Index.cshtml всередині циклу foreach. Такий підхід ускладнює підтримку, оскільки будь-які зміни оформлення елемента доводиться робити в основному представленні.

Спочатку у папці Views/Shared було створено Partial View \_EventCard.cshtml, який приймає модель типу Event і відповідає за відображення однієї події. Після цього розмітку одиничного елемента було перенесено з Index.cshtml у новий файл. У головному представленні Index.cshtml замість дублювання розмітки використано Tag Helper <partial> для рендерингу Partial View для кожної події.

					ДУ «Житомирська політехніка».25.121.1.000 – Лр3								
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи				Літ.	Арк.	Аркушів		
Розроб.	Андріюк Д. Р.										1	15	
Перевір.	Українець М.О.								ФІКТ Гр. ІПЗ-22-2[1]				
Керівник													
Н. контр.													
Зав. каф.													

Лістинг коду \_EventCard.cshtml:

```
@model Event

<div class="col-md-4 mb-4">
  <div class="card h-100 shadow-sm">
    <div class="card-body">
      <h5 class="card-title">@Model.Title</h5>
      <p class="card-text">@Model.Description</p>
      <div class="text-muted small">ID: @Model.EventId</div>

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

    <div class="card-footer d-flex justify-content-between align-items-center">
      <span class="fw-bold">
        @Model.BaseTicketPrice.ToString("c")
      </span>

      <form asp-controller="Cart" asp-action="AddToCart" method="post" class="d-inline">
        <button type="submit" class="btn btn-sm btn-primary"
          name="eventId" value="@Model.EventId">
          Book
        </button>
      </form>
    </div>
  </div>
</div>
```

Лістинг фрагменту коду Index.cshtml:

```
<div class="row">
  @foreach (var e in Model.Events ?? Enumerable.Empty<Event>()) {
    <partial name="_EventCard" model="e" />
  }
</div>
```

Після запуску проєкту зовнішній вигляд сторінки залишився незмінним, але структура коду стала більш модульною, що полегшує подальше розширення проєкту (наприклад, підключення кнопок бронювання, або використання картки події в інших сторінках).

		Андріюк Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр3	Арк.
		Українець М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

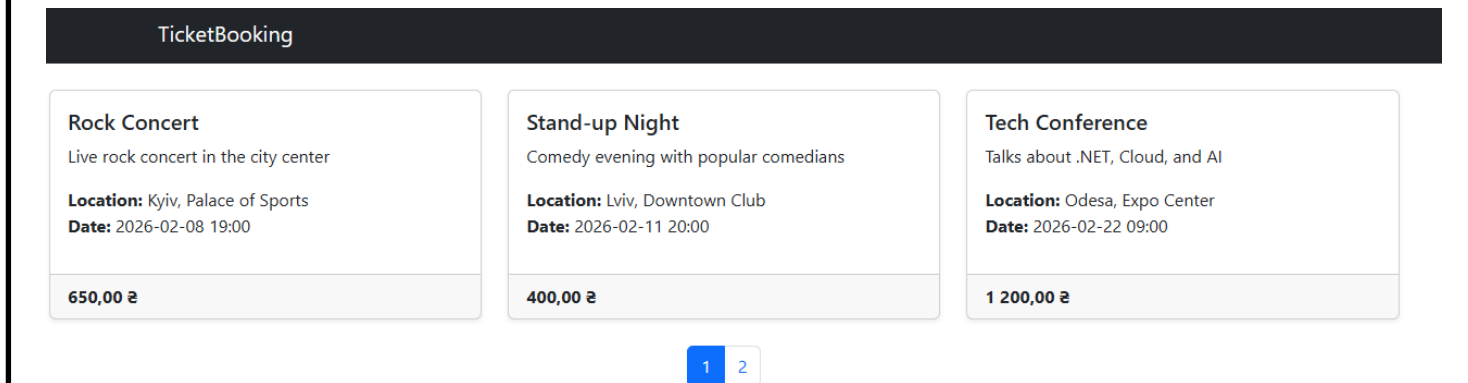


Рис. 1. Результат використання Partial View  
для відображення елементів списку подій.

## Завдання 2. Навігація по категоріям.

Реалізуйте навігацію за категорією (або за іншою ознакою, яка більше підходить за вашим варіантом), дотримуючись вказаних кроків.

Для зручної навігації користувача було реалізовано сайдбар з переліком категорій (ознака фільтрації). Оскільки проєкт TicketBooking працює з подіями, як «категорію» було обрано Location (місце проведення події), що відповідає вимозі завдання щодо альтернативної ознаки для фільтрації.

На першому кроці було створено View Component `NavigationMenuViewComponent`, який спочатку повертав тестовий текст. Це дозволило швидко перевірити коректність підключення компонента у макеті сторінки. Далі у файлі `_Layout.cshtml` додано розмітку двоколонкової структури сторінки: зліва — сайдбар, справа — основний контент. У сайдбар вбудовано виклик View Component через `@await Component.InvokeAsync("NavigationMenu")`.

Лістинг `NavigationMenuViewComponent.cs`:

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

namespace TicketBooking.Components {
    public class NavigationMenuViewComponent : ViewComponent {
        public string Invoke() {
```

		Андріюк Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр3	Арк.
		Українець М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

```

        return "Hello from the Nav View Component";
    }
}

```

Лістинг фрагменту \_Layout.cshtml (main елемент):

```

<main class="container-fluid mt-4">
  <div class="row">
    <aside class="col-12 col-md-3 col-lg-2 mb-3">
      @await Component.InvokeAsync("NavigationMenu")
    </aside>

    <section class="col-12 col-md-9 col-lg-10">
      @RenderBody()
    </section>
  </div>
</main>

```

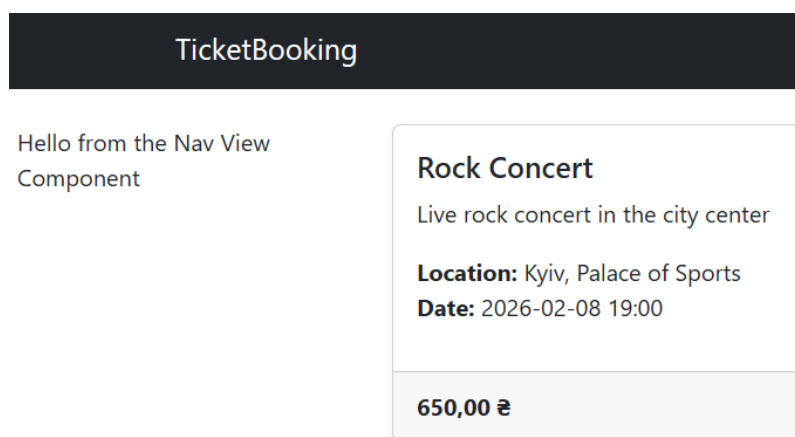


Рис. 2.1. Перевірка підключення  
NavigationMenuViewComponent у шаблоні сторінки.

Після перевірки компонент було доопрацьовано: він отримує у репозиторію список унікальних локацій з бази даних, сортує їх та передає у представлення компонента. Для підтримки підсвічування поточної категорії, View Component читає параметр рядка запиту location та передає його у View як CurrentLocation. Для цього було створено окрему ViewModel NavigationMenuViewModel.

Лістинг коду NavigationMenuViewModel.cs:

```

namespace TicketBooking.Models.ViewModels {
    public class NavigationMenuViewModel {
        public IEnumerable<string> Locations { get; set; } = Enumerable.Empty<string>();
        public string? CurrentLocation { get; set; }
    }
}

```

		Андріюк Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр3	Арк.
		Українець М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

## Лістинг коду оновленого NavigationMenuViewComponent.cs:

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

namespace TicketBooking.Components {
    public class NavigationMenuViewComponent : ViewComponent {
        private readonly IEventRepository _repository;

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

        public IViewComponentResult Invoke() {
            string? currentLocation = HttpContext.Request.Query["location"];

            var locations = _repository.Events
                .Select(e => e.Location)
                .Distinct()
                .OrderBy(x => x)
                .ToList();

            var model = new NavigationMenuViewModel {
                Locations = locations,
                CurrentLocation = string.IsNullOrEmpty(currentLocation) ? null
: currentLocation
            };

            return View(model);
        }
    }
}
```

Для відображення меню компоненту згідно конвенції Razor створено файл Views/Shared/Components/NavigationMenu/Default.cshtml. У ньому сформовано список посилань для навігації, а для активного пункту застосовано клас Bootstrap active. Також посилання передає pageNum=1, щоб при виборі категорії користувач завжди потрапляв на першу сторінку відфільтрованого списку.

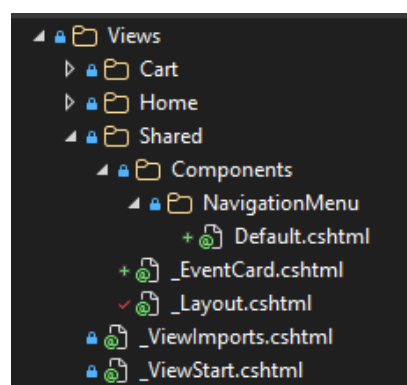


Рис. 2.2. Розміщення представлення View Component за конвенцією Razor.

		Андріюк Д. Р.			ДУ «Житомирська політехніка». 25.121.1.000 – Лр3	Арк.
		Українець М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

### Лістинг коду Default.cshtml:

```
@model TicketBooking.Models.ViewModels.NavigationMenuViewModel

<div class="list-group">
    <a class="list-group-item list-group-item-action @(Model.CurrentLocation == null ?
"active" : "")"
        asp-controller="Home"
        asp-action="Index">
        Home
    </a>

    @foreach (var loc in Model.Locations) {
        <a class="list-group-item list-group-item-action @(loc == Model.CurrentLocation ?
"active" : "")"
            asp-controller="Home"
            asp-action="Index"
            asp-route-location="@loc"
            asp-route-pageNum="1">
            @loc
        </a>
    }
</div>
```

Далі було реалізовано фільтрацію у контролері HomeController. Метод Index приймає параметр location та застосовує фільтр до запиту репозиторію перед пагінацією. Одночасно ViewModel розширено полем CurrentLocation для збереження стану вибраної категорії на представленні.

### Лістинг коду методу Index класу HomeController:

```
public IActionResult Index(string? location, int pageNum = 1) {
    var query = _repository.Events;

    if (!string.IsNullOrEmpty(location)) {
        query = query.Where(e => e.Location == location);
    }

    query = query.OrderBy(e => e.EventId);

    int totalItems = query.Count();
    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
        },
        CurrentLocation = location
    };

    return View(model);
}
```

		Андрійук Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр3	Арк.
		Українець М.О.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

### Лістинг коду оновленого EventsListViewModel:

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

Окремо було змінено генерацію посилань у пейджингу: тепер при переході між сторінками зберігається параметр location, а кількість сторінок обчислюється відповідно до кількості подій у вибраній категорії.

### Лістинг фрагменту пейджингу у Index.cshtml:

```
<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"
                    asp-route-location="@Model.CurrentLocation">
                    @i
                </a>
            </li>
        }
    </ul>
</nav>
```

У результаті реалізації навігації користувач може обрати локацію у сайдбарі, отримати відфільтрований список подій і бачити активну категорію. Пейджинг відображає коректну кількість сторінок для обраного фільтра.

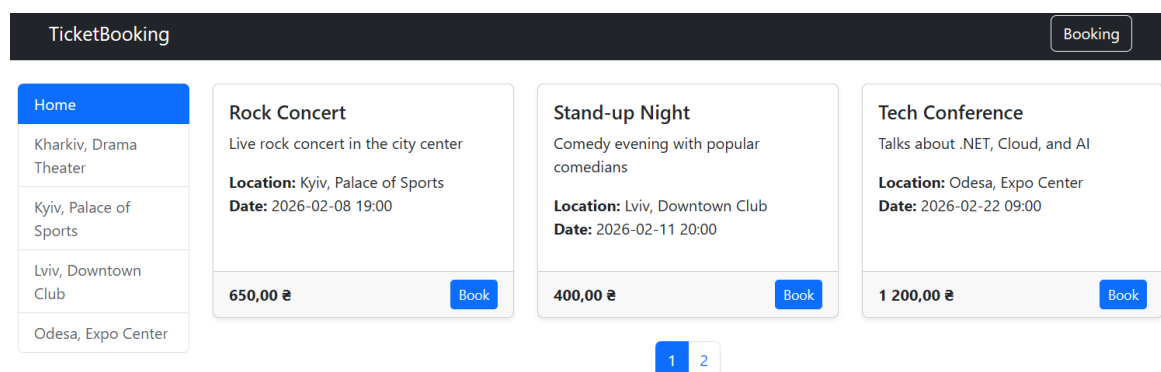


Рис. 2.3.1. Навігація по категоріях. Загальна категорія

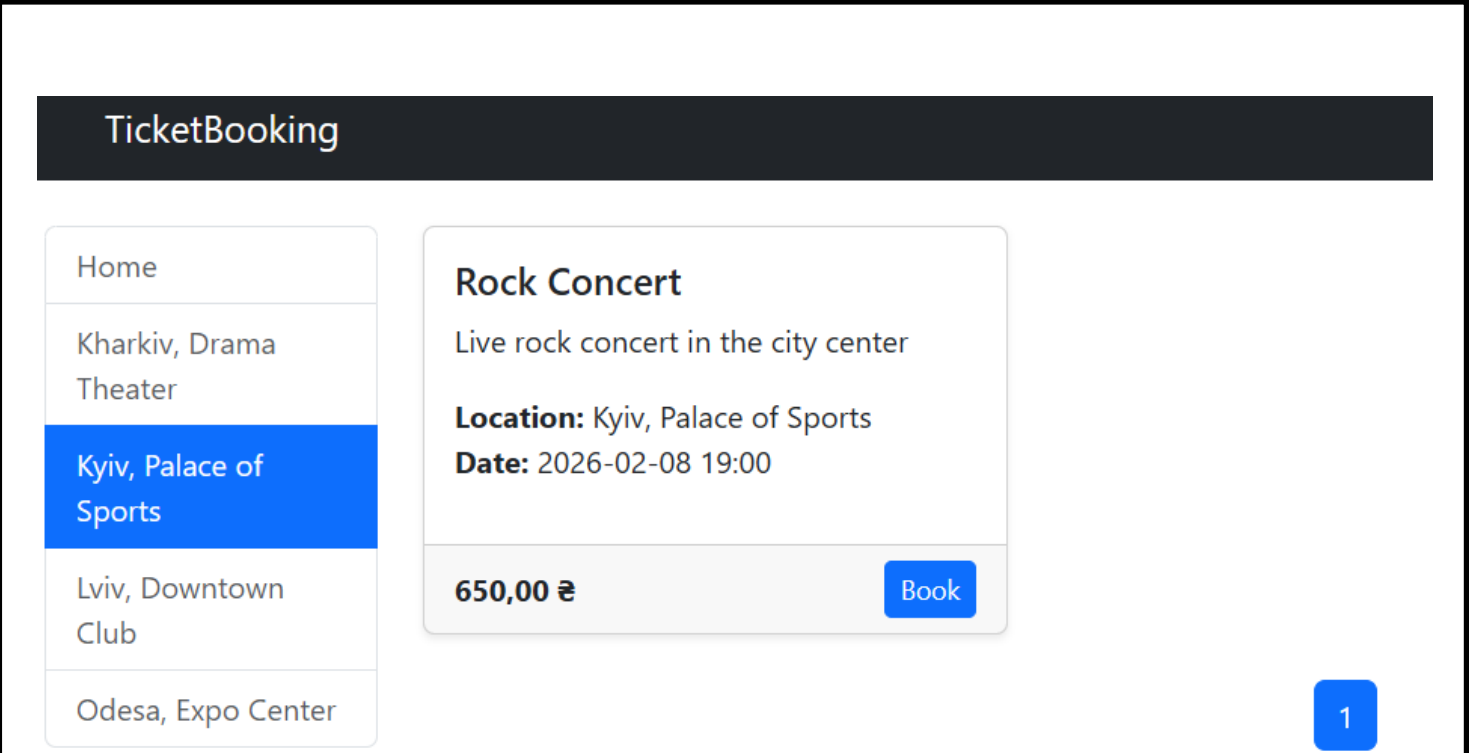


Рис. 2.3.2. Навігація по категоріях та фільтрація подій з коректним пейджингом. Приклад 1

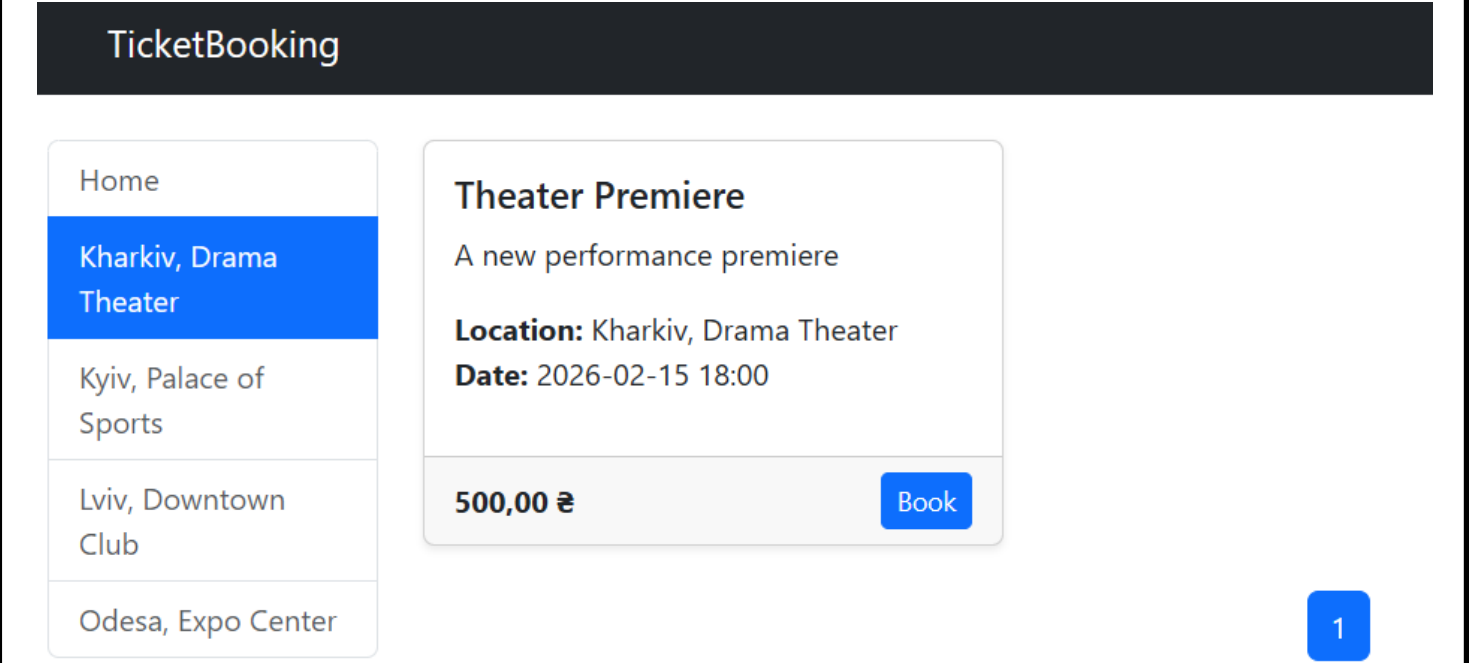


Рис. 2.3.3. Навігація по категоріях та фільтрація подій з коректним пейджингом. Приклад 2



### Завдання 3. Сесії.

Реалізуйте функціональність відповідно до варіанту вашого проекту з використанням механізму сесії.

У цьому завданні було реалізовано збереження стану «кошика бронювання» між HTTP-запитами за допомогою механізму Session. Оскільки HTTP є stateless-протоколом, без використання сесії додаток не може «пам'ятати» вибір користувача між переходами сторінками. Для TicketBooking сесія використовується для зберігання списку вибраних подій (квитків) у «Booking Cart».

Спочатку у файлі Program.cs було увімкнено підтримку сесії: додано AddDistributedMemoryCache, AddSession(...) та middleware UseSession(). Також було зареєстровано IHttpContextAccessor для доступу до HttpContext.Session в класах, що створюються через dependency injection.

Лістинг фрагменту коду Program.cs:

```
builder.Services.AddControllersWithViews();
builder.Services.AddDistributedMemoryCache();

builder.Services.AddSession(options => {
    options.IdleTimeout = TimeSpan.FromMinutes(30);
    options.Cookie.HttpOnly = true;
    options.Cookie.IsEssential = true;
});

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

builder.Services.AddHttpContextAccessor();
builder.Services.AddScoped<Cart>(sp => SessionCart.GetCart(sp));
builder.Services.AddScoped<IEventRepository, EFEventRepository>();

var app = builder.Build();

app.UseStaticFiles();
app.UseSession();
app.MapDefaultControllerRoute();
```

Для збереження складних об'єктів у сесії реалізовано допоміжні розширення SessionExtensions, які серіалізують/десеріалізують об'єкти у JSON. Далі створено моделі CartLine та Cart, які описують елемент кошика (подія + кількість) та сам кошик.

		Андріюк Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр3	Арк.
		Українець М.О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Для збереження кошика у сесії створено клас SessionCart, який отримує або створює екземпляр кошика з сесії та зберігає його назад після змін.

Під час реалізації було виявлено важливий нюанс: для коректної JSON-серіалізації список елементів кошика має бути представленим як публічна властивість, доступна для серіалізатора. Тому властивість Lines у класі Cart була реалізована як public List<CartLine> Lines { get; set; }, що забезпечило коректне збереження та відновлення даних з сесії.

Лістинг коду SessionExtensions.cs:

```
using System.Text.Json;
using Microsoft.AspNetCore.Http;

namespace TicketBooking.Infrastructure {
    public static class SessionExtensions {
        public static void SetJson(this ISession session, string key, object value) {
            session.SetString(key, JsonSerializer.Serialize(value));
        }

        public static T? GetJson<T>(this ISession session, string key) {
            var data = session.GetString(key);
            return data == null ? default : JsonSerializer.Deserialize<T>(data);
        }
    }
}
```

Лістинг коду CartLine.cs:

```
namespace TicketBooking.Models {
    public class CartLine {
        public Event Event { get; set; } = default!;
        public int Quantity { get; set; }
    }
}
```

Лістинг коду Cart.cs:

```
namespace TicketBooking.Models {
    public class Cart {
        public List<CartLine> Lines { get; set; } = new();

        public void AddItem(Event ev, int quantity) {
            var line = Lines.FirstOrDefault(l => l.Event.EventId == ev.EventId);
            if (line == null) {
                Lines.Add(new CartLine { Event = ev, Quantity = quantity });
            } else {
                line.Quantity += quantity;
            }
        }

        public void RemoveLine(long eventId) {
            Lines.RemoveAll(l => l.Event.EventId == eventId);
        }
    }
}
```

		Андрійук Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр3	Арк.
		Українець М.О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        Lines.RemoveAll(l => l.Event.EventId == eventId);
    }

    public void Clear() => Lines.Clear();

    public decimal ComputeTotalValue()
        => Lines.Sum(l => l.Event.BaseTicketPrice * l.Quantity);
    }
}

```

Лістинг коду SessionCart.cs:

```

using TicketBooking.Infrastructure;

namespace TicketBooking.Models {
    public class SessionCart : Cart {
        public static Cart GetCart(IServiceProvider services) {
            var session = services.GetRequiredService<IHttpContextAccessor>()
                .HttpContext!.Session;

            var cart = session.GetJson<SessionCart>("Cart") ?? new SessionCart();
            return cart;
        }

        public void Save(ISession session) {
            session.SetJson("Cart", this);
        }
    }
}

```

Для керування кошиком було створено CartController з діями додавання (AddToCart), видалення (RemoveFromCart) та перегляду кошика (Index). У картці події (Partial View) додано кнопку «Book», яка надсилає POST-запит на додавання події до кошика.

Лістинг коду CartController.cs:

```

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

namespace TicketBooking.Controllers {
    public class CartController : Controller {
        private readonly IEventRepository _repository;
        private readonly Cart _cart;

        public CartController(IEventRepository repository, Cart cart) {
            _repository = repository;
            _cart = cart;
        }

        public IActionResult Index() {
            return View(_cart);
        }

        [HttpPost]
        public IActionResult AddToCart(long eventId, string? returnUrl) {
            var ev = _repository.Events.FirstOrDefault(e => e.EventId == eventId);

```

		Андрійук Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр3	Арк.
		Українець М.О.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if (ev != null) {
            _cart.AddItem(ev, 1);
            (_cart as SessionCart)?.Save(HttpContext.Session);
        }

        return RedirectToAction("Index", new { returnUrl });
    }

    [HttpPost]
    public IActionResult RemoveFromCart(long eventId, string? returnUrl) {
        _cart.RemoveLine(eventId);
        (_cart as SessionCart)?.Save(HttpContext.Session);

        return RedirectToAction("Index", new { returnUrl });
    }

    [HttpPost]
    public IActionResult Clear(string? returnUrl) {
        _cart.Clear();
        (_cart as SessionCart)?.Save(HttpContext.Session);

        return RedirectToAction("Index", new { returnUrl });
    }
}

```

Лістинг фрагменту коду \_EventCard.cshtml:

```

<div class="card-footer d-flex justify-content-between align-items-center">
    <span class="fw-bold">
        @Model.BaseTicketPrice.ToString("c")
    </span>

    <form asp-controller="Cart" asp-action="AddToCart" method="post" class="d-inline">
        <button type="submit" class="btn btn-sm btn-primary"
            name="eventId" value="@Model.EventId">
            Book
        </button>
    </form>
</div>

```

Для відображення кошика створено представлення Views/Cart/Index.cshtml, де показано список обраних подій, їх кількість, ціну та загальну суму. Також додано посилання на кошик у навігаційну панель \_Layout.cshtml.

Лістинг коду Index.cshtml:

```

@model TicketBooking.Models.Cart

<h2>Your Booking</h2>

@if (!Model.Lines.Any())
{
    <div class="alert alert-info">Your booking cart is empty.</div>
}
else
{
    <table class="table table-bordered">

```

		Андріюк Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр3	Арк.
		Українець М.О.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<thead>
<tr>
    <th>Event</th>
    <th class="text-center">Qty</th>
    <th class="text-end">Price</th>
    <th class="text-end">Total</th>
    <th></th>
</tr>
</thead>
<tbody>
@foreach (var line in Model.Lines)
{
    <tr>
        <td>@line.Event.Title</td>
        <td class="text-center">@line.Quantity</td>
        <td class="text-end">@line.Event.BaseTicketPrice.ToString("c")</td>
        <td class="text-end">@((line.Event.BaseTicketPrice *
line.Quantity).ToString("c"))</td>
        <td class="text-end">
            <form asp-action="RemoveFromCart" method="post">
                <input type="hidden" name="eventId" value="@line.Event.EventId" />
                <button type="submit" class="btn btn-sm btn-outline-
danger">Remove</button>
            </form>
        </td>
    </tr>
}
</tbody>
<tfoot>
<tr>
    <td colspan="3" class="text-end"><strong>Total:</strong></td>
    <td class="text-
end"><strong>@Model.ComputeTotalValue().ToString("c")</strong></td>
    <td></td>
</tr>
</tfoot>
</table>

<form asp-action="Clear" method="post">
    <button class="btn btn-warning">Clear</button>
</form>
}

```

Лістинг фрагменту коду \_Layout.cshtml:

```

<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>
        <a class="btn btn-outline-light" asp-controller="Cart" asp-action="Index">
            Booking
        </a>
    </div>
</nav>
</header>

```

		Андріюк Д. Р.			ДУ «Житомирська політехніка».25.121.1.000 – Лр3	Арк.
		Українець М.О.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

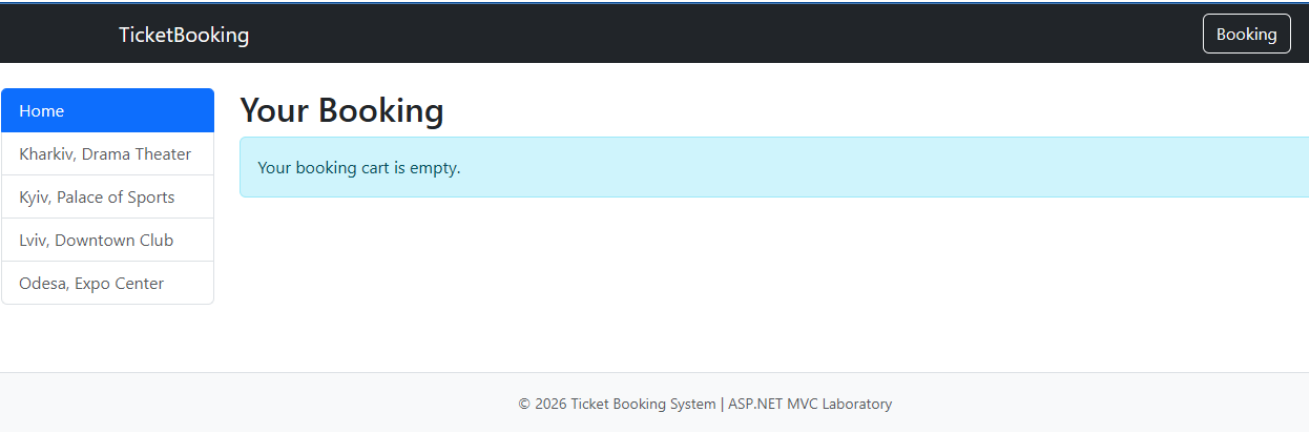


Рис. 3.1. Відображення пустого кошику.

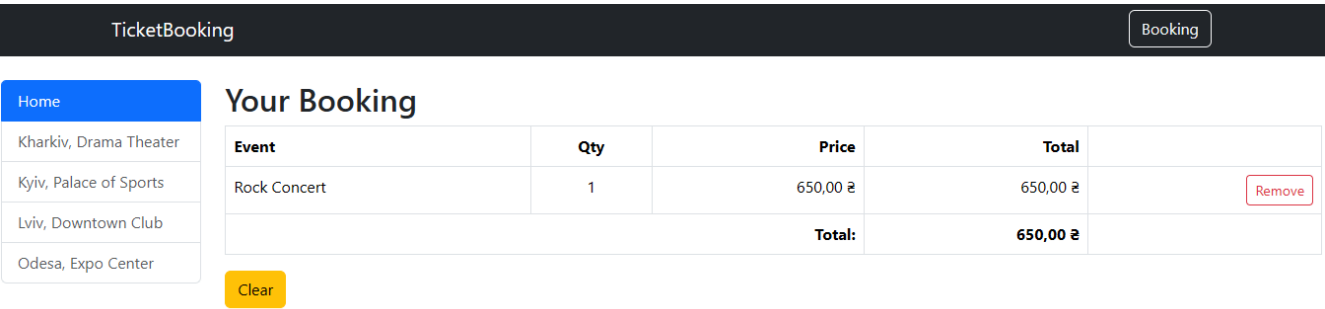


Рис. 3.2. Збережена 1 подія у кошику.

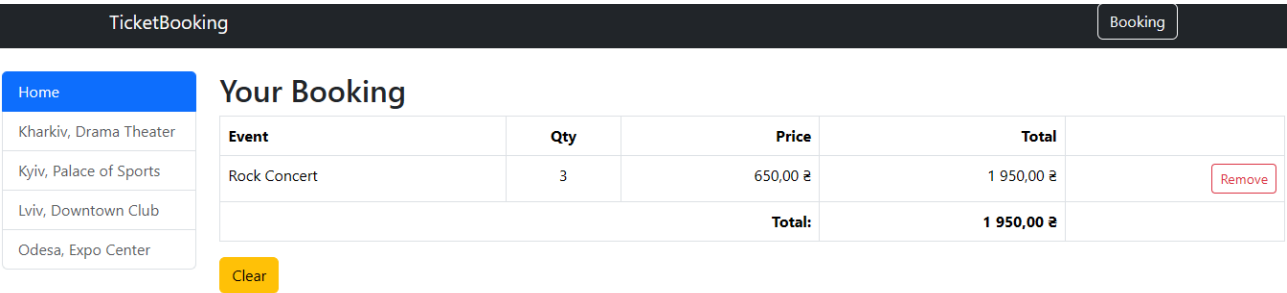


Рис. 3.2. Збережені декілька однакових подій у кошику.

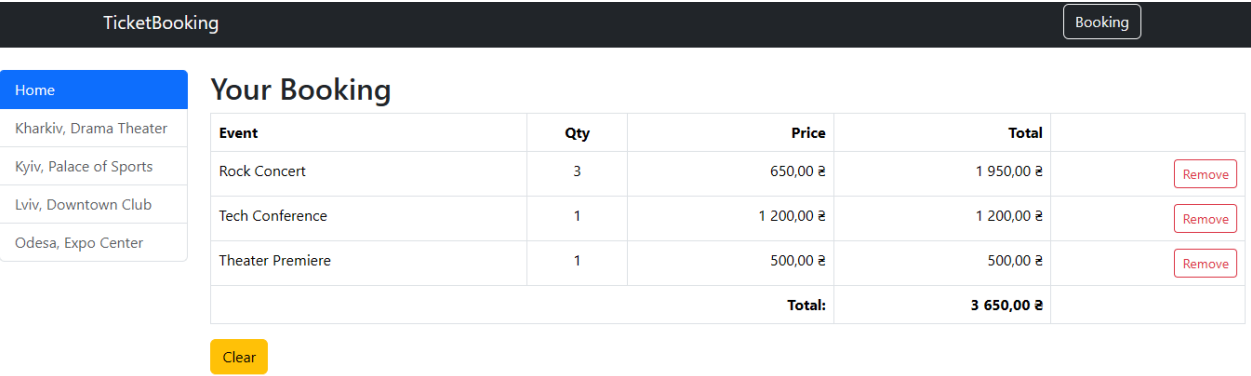


Рис. 3.3. Збережені декілька різних подій у кошику.

**Висновок:** У ході виконання лабораторної роботи №3 було поглиблено навички роботи з представленнями ASP.NET MVC та механізмами взаємодії інтерфейсу з даними. Створено Partial View для відображення одиничної події та інтегровано його в основне представлення, що зробило код більш модульним. Реалізовано View Component для навігації по категоріях (Location), додано фільтрацію через параметри рядка запиту та забезпечено коректну роботу пейджингу з урахуванням фільтрів. Також реалізовано роботу з сесією для збереження стану кошика бронювання між HTTP-запитами, що дозволило додатку підтримувати користувацький стан у stateless-середовищі HTTP.

		Андріюк Д. Р.			ДУ «Житомирська політехніка». 25.121.1.000 – Лр3	Арк.
		Українець М.О.				15
Змн.	Арк.	№ докум.	Підпис	Дата		