***Лабораторна робота №3***

**Тема: 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.

**Завдання 1**

Створення часткового представлення (Partial View)

Лістінг коду VehiclesSummary.cshtml (код вже з доданими куками та пейджингом):

@model VehicleRentalService.Models.ViewModels.VehicleListViewModel

<link rel="stylesheet" href="@Url.Content("~/css/card-styles.css")" />

@{

foreach (var v in Model.Vehicles)

{

<div class="card">

<div class="card-body">

<h5 class="card-title">@v.Name</h5>

<p class="card-text">@v.Description</p>

<div class="d-flex justify-content-between align-items-center">

<span class="h5 mb-0">@v.PricePerMinute/minute</span>

</div>

</div>

<div class="card-footer d-flex justify-content-between bg-light">

<**form** **asp-controller**="Home" **asp-action**="SetCookies" method="post">

<input type="number" name="Quantity" value="1" min="1" class="form-control-sm" />

<input type="hidden" name="VehicleType" value="@v.GetType().Name" />

<input type="hidden" name="VehicleId" value="@v.VehicleId" />

<button class="btn btn-primary btn-sm" type="submit">Add to Cart</button>

</**form**>

</div>

</div>

}

<div>

Page @Model.PagingInfo.CurrentPage / @Model.PagingInfo.TotalPages

</div>

<div>

@for (int i = 1; i <= Model.PagingInfo.TotalPages; i++)

{

if (i == Model.PagingInfo.CurrentPage)

{

<span>@i</span>

}

Else

{

<a href="@Url.Action("Index", new { page = i, category = ViewData["CurrentCategory"] })">@i</a>

}

}

</div>

}

Лістінг головної сторінки із використанням Partial View:

@section Sidebar {

<aside class="col-md-3 col-lg-2 bg-light border-end min-vh-100 p-3">

@await Component.InvokeAsync("NavigationMenu")

</aside>

}

@await Html.PartialAsync("\_VehiclesSummary")

**Завдання 2**

Навігація по категоріям

Лістінг NavigationViewComponent:

using Microsoft.AspNetCore.Mvc;

using VehicleRentalService.Models;

namespace VehicleRentalService.Components

{

public class NavigationMenuViewComponent : ViewComponent

{

public readonly string[] \_vehicleTypes = new string[]

{

"Bike",

"Car",

"Scooter"

};

public IViewComponentResult Invoke()

{

return View(\_vehicleTypes);

}

}

}

Лістінг \_Layout.cshtml:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>@ViewData["Title"] VehicleRentalService</title>

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

<**link** rel="stylesheet" href="~/css/layouts.css" />

</head>

<body class="d-flex flex-column min-vh-100">

<header class="bg-dark text-white p-3">

<div class="container">

<a class="logo" href="/">Vehicle Rental Service</a>

<a class="shopping-cart-link" href="/ShoppingCart">

<**img** src="~/images/shopping-cart.png" class="shopping-cart-logo" />

</a>

</div>

</header>

<div class="container-fluid flex-grow-1">

<div class="row">

@RenderSection("Sidebar", required: false)

<main class="col-md-9 col-lg-10 p-4">

@RenderBody()

</main>

</div>

</div>

<footer class="bg-light text-center p-3 mt-auto">

<div class="container">

&copy; 2025 Vehicle Rental Service

</div>

</footer>

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

</body>

</html>

Виклик VehiclesSummary видалено із \_Layout, бо для виконання завдання із сессіями, потрібно додати інші сторінки, на яких виклик VehiclesSummary не потрібен

Частина HomeController.cs із виведенням транспортних засобів по категоріям(вимушено довелось перенести із NavigationMenuViewComponent до HomeController):

public IActionResult Index(string category ="Car", int page = 1)

{

ViewData["CurrentCategory"] = category;

var totalItems = category switch

{

"Car" => \_repository.Cars.Count(),

"Bike" => \_repository.Bikes.Count(),

"Scooter" => \_repository.Scooters.Count(),

\_ => 0

};

#if DEBUG

Console.WriteLine($"Items count:{totalItems}");

#endif

var vehicles = category switch

{

"Car" => \_repository.Cars

.Cast<Vehicle>()

.OrderBy(v => v.VehicleId)

.Skip((page - 1) \* \_pageSize)

.Take(\_pageSize),

"Bike" => \_repository.Bikes

.Cast<Vehicle>()

.OrderBy(v => v.VehicleId)

.Skip((page - 1) \* \_pageSize)

.Take(\_pageSize),

"Scooter" => \_repository.Scooters.Cast<Vehicle>()

.OrderBy(v => v.VehicleId)

.Skip((page - 1) \* \_pageSize)

.Take(\_pageSize),

\_ => Enumerable.Empty<Vehicle>()

};

var ViewModelVh = new VehicleListViewModel

{

Vehicles = vehicles,

PagingInfo = new PagingInfo

{

CurrentPage = page,

ItemsPerPage = \_pageSize,

TotalItems = totalItems,

}

};

return View(ViewModelVh);

}

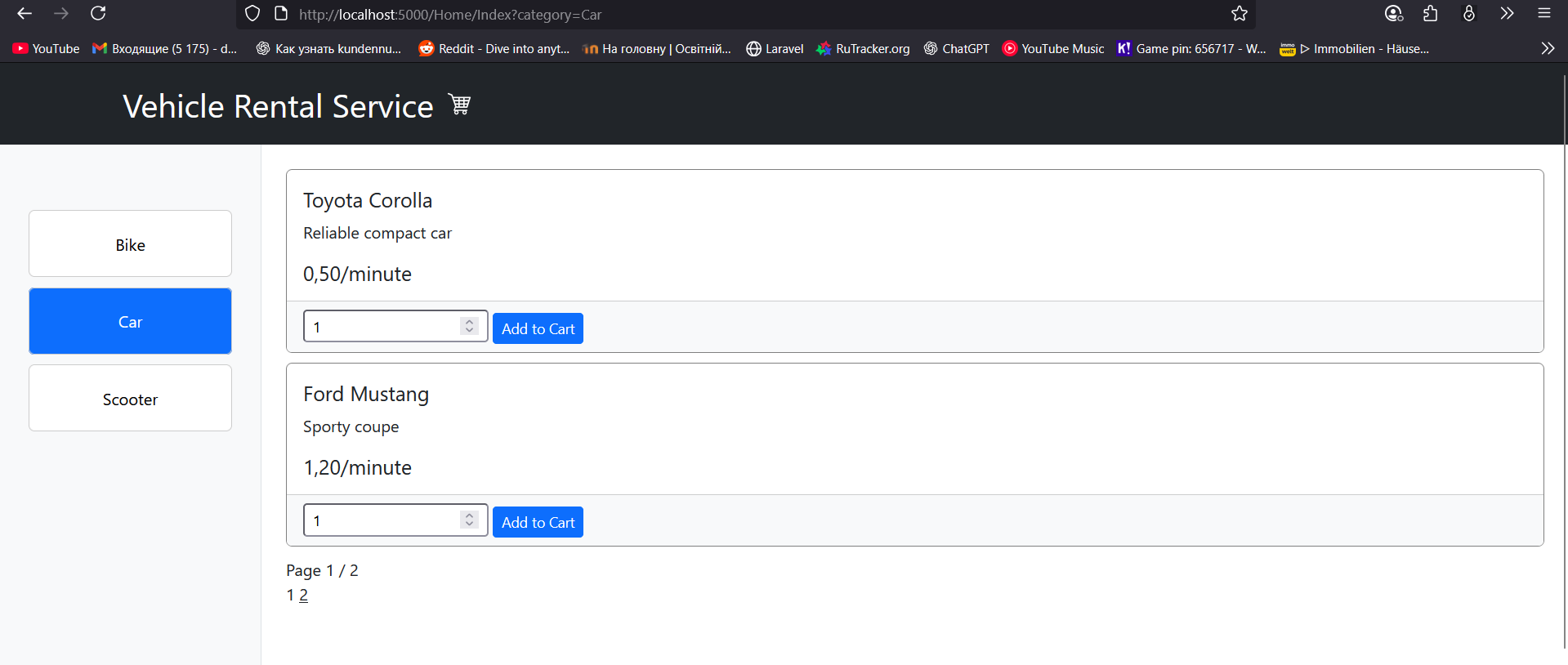


Рис.1.Результат виконання завдання 1 та 2

Як можна побачити, було додано три категорії: Велосипеди, автівки та самокати. В наявності можливість збільшення кількості ТЗ та додавання у кошик(кошик та сессії будуть розроблені у завданні 3). Також реалізован

пейджинг. За замовчуванням, на однієї сторінці відображається виключно 2 ТЗ.

Завдання 3

Реалізуйте функціональність відповідно до варіанту вашого проекту з використанням механізму сесії

1)Додамо сессії до Program.cs та налаштовуємо:

builder.Services.AddDistributedMemoryCache();

builder.Services.AddSession(options =>

{

options.IdleTimeout = TimeSpan.FromSeconds(9999);

options.Cookie.HttpOnly = true;

options.Cookie.IsEssential = true;

options.Cookie.Name = "CartData";

});

app.UseSession();

2) Додавання кнопки придбання автівки:

@model VehicleRentalService.Models.ViewModels.VehicleListViewModel

<link rel="stylesheet" href="@Url.Content("~/css/card-styles.css")" />

@{

foreach (var v in Model.Vehicles)

{

<div class="card">

<div class="card-body">

<h5 class="card-title">@v.Name</h5>

<p class="card-text">@v.Description</p>

<div class="d-flex justify-content-between align-items-center">

<span class="h5 mb-0">@v.PricePerMinute/minute</span>

</div>

</div>

<div class="card-footer d-flex justify-content-between bg-light">

<**form** **asp-controller**="Home" **asp-action**="SetCookies" method="post">

<input type="number" name="Quantity" value="1" min="1" class="form-control-sm" />

<input type="hidden" name="VehicleType" value="@v.GetType().Name" />

<input type="hidden" name="VehicleId" value="@v.VehicleId" />

<button class="btn btn-primary btn-sm" type="submit">Add to Cart</button>

</**form**>

</div>

</div>

}

<div>

Page @Model.PagingInfo.CurrentPage / @Model.PagingInfo.TotalPages

</div>

<div>

@for (int i = 1; i <= Model.PagingInfo.TotalPages; i++)

{

if (i == Model.PagingInfo.CurrentPage)

{

<span>@i</span>

}

else

{

<a href="@Url.Action("Index", new { page = i, category = ViewData["CurrentCategory"] })">@i</a>

}

}

</div>

}

Див. далі:

<div class="card-footer d-flex justify-content-between bg-light">

<**form** **asp-controller**="Home" **asp-action**="SetCookies" method="post">

<input type="number" name="Quantity" value="1" min="1" class="form-control-sm" />

<input type="hidden" name="VehicleType" value="@v.GetType().Name" />

<input type="hidden" name="VehicleId" value="@v.VehicleId" />

<button class="btn btn-primary btn-sm" type="submit">Add to Cart</button>

</**form**>

</div>

При натисканні на кнопку викликається метод SetCookies, який і записує данні в кукі.

public IActionResult Index()

{

//getting all data from cookies

var json = HttpContext.Session.GetString("cart");

List<CartItem> cartContent = string.IsNullOrEmpty(json)

? new List<CartItem>()

: JsonSerializer.Deserialize<List<CartItem>>(json);

var tripBeginString = HttpContext.Session.GetString("tripBegin");

DateTime tripBegin = string.IsNullOrEmpty(tripBeginString)

? DateTime.Now

: DateTime.Parse(tripBeginString, null, System.Globalization.DateTimeStyles.RoundtripKind);

var tripEndString = HttpContext.Session.GetString("tripEnd");

DateTime tripEnd = string.IsNullOrEmpty(tripEndString)

? DateTime.Now

: DateTime.Parse(tripEndString, null, System.Globalization.DateTimeStyles.RoundtripKind);

var shoppingCart = new ShoppingCart(\_repository)

{

CartItems = cartContent ?? new List<CartItem>(),

TripBegin = tripBegin,

TripEnd = tripEnd

};

var viewModel = new CartViewModel

{

TripBegin = shoppingCart.TripBegin,

TripEnd = shoppingCart.TripEnd,

Duration = shoppingCart.Duration,

Total = shoppingCart.CalculateTotalCartPrice(),

Items = shoppingCart.CartItems.Select(item =>

{

var vehicle = \_repository.FindById(item.VehicleType, item.Id);

return new CartItemViewModel

{

Id = item.Id,

VehicleName = vehicle?.Name ?? "Unknown",

VehicleType = item.VehicleType,

Quantity = item.Quantity,

PricePerMinute = vehicle?.PricePerMinute ?? 0,

Subtotal = (vehicle?.PricePerMinute ?? 0) \* (decimal)shoppingCart.Duration.TotalMinutes \* item.Quantity

};

}).ToList()

};

return View(viewModel);

}

3)Також було додано саму корзину, де будуть виводитись товари, які були додані до сесіїю. Було створено контролер та представлення.  
  
ShoppingCartController:

using System.Reflection.Metadata.Ecma335;

using System.Text.Json;

using Microsoft.AspNetCore.Mvc;

using Microsoft.IdentityModel.Tokens;

using VehicleRentalService.Models;

using VehicleRentalService.Models.ViewModels;

namespace VehicleRentalService.Controllers

{

public class ShoppingCartController : Controller

{

private readonly IServiceRepository \_repository;

public ShoppingCartController(IServiceRepository repository)

{

\_repository = repository;

}

public IActionResult Index()

{

//getting all data from cookies

var json = HttpContext.Session.GetString("cart");

List<CartItem> cartContent = string.IsNullOrEmpty(json)

? new List<CartItem>()

: JsonSerializer.Deserialize<List<CartItem>>(json);

var tripBeginString = HttpContext.Session.GetString("tripBegin");

DateTime tripBegin = string.IsNullOrEmpty(tripBeginString)

? DateTime.Now

: DateTime.Parse(tripBeginString, null, System.Globalization.DateTimeStyles.RoundtripKind);

var tripEndString = HttpContext.Session.GetString("tripEnd");

DateTime tripEnd = string.IsNullOrEmpty(tripEndString)

? DateTime.Now

: DateTime.Parse(tripEndString, null, System.Globalization.DateTimeStyles.RoundtripKind);

var shoppingCart = new ShoppingCart(\_repository)

{

CartItems = cartContent ?? new List<CartItem>(),

TripBegin = tripBegin,

TripEnd = tripEnd

};

var viewModel = new CartViewModel

{

TripBegin = shoppingCart.TripBegin,

TripEnd = shoppingCart.TripEnd,

Duration = shoppingCart.Duration,

Total = shoppingCart.CalculateTotalCartPrice(),

Items = shoppingCart.CartItems.Select(item =>

{

var vehicle = \_repository.FindById(item.VehicleType, item.Id);

return new CartItemViewModel

{

Id = item.Id,

VehicleName = vehicle?.Name ?? "Unknown",

VehicleType = item.VehicleType,

Quantity = item.Quantity,

PricePerMinute = vehicle?.PricePerMinute ?? 0,

Subtotal = (vehicle?.PricePerMinute ?? 0) \* (decimal)shoppingCart.Duration.TotalMinutes \* item.Quantity

};

}).ToList()

};

return View(viewModel);

}

[HttpPost]

public IActionResult RemoveCookies(long Id, VehicleType vehicleType)

{

var json = HttpContext.Session.GetString("cart");

if (!string.IsNullOrEmpty(json))

{

var cart = JsonSerializer.Deserialize<List<CartItem>>(json);

var itemToRemove = cart.FirstOrDefault(c => c.Id == Id && c.VehicleType == vehicleType);

if (itemToRemove != null)

{

cart.Remove(itemToRemove);

HttpContext.Session.SetString("cart", JsonSerializer.Serialize(cart));

}

}

return RedirectToAction("Index", "ShoppingCart");

}

}

}

Представлення:  
@model VehicleRentalService.Models.ViewModels.CartViewModel

<link rel="stylesheet" href="@Url.Content("~/css/shopping-cart-styles.css")" />

@{

@foreach (var v in Model.Items)

{

<div class="card">

<div class="card-body">

<h5 class="card-title">@v.VehicleName</h5>

<div class="d-flex justify-content-between align-items-center">

<span class="h5 mb-0">@v.PricePerMinute/minute</span>

</div>

</div>

<div class="card-footer d-flex justify-content-between bg-light">

<span>@v.Quantity</span>

<**form** **asp-controller**="ShoppingCart" **asp-action**="RemoveCookies" method="post">

<input type="hidden" name="VehicleType" value="@v.VehicleType" />

<input type="hidden" name="Id" value="@v.Id" />

<button class="btn btn-primary btn-sm" name="remove-btn" type="submit">Remove</button>

</**form**>

</div>

</div>

}

}

<**form** method="post" **asp-controller**="Orders" **asp-action**="PlaceOrder" id="orderForm">

<div class="mb-3">

<label for="TripStart">Start Time</label>

<input type="datetime-local" id="TripStart" name="TripBegin" value="@Model.TripBegin.ToString("yyyy-MM-ddTHH:mm")" class="form-control" />

</div>

<div class="mb-3">

<label for="TripEnd">End Time</label>

<input type="datetime-local" id="TripEnd" name="TripEnd" value="@Model.TripEnd.ToString("yyyy-MM-ddTHH:mm")" class="form-control" />

</div>

<button class="btn btn-primary btn-lg" type="submit">Place Order</button>

</**form**>

Також було створено 3 моделі-представлення: CartViewModel, AddToCartViewModel та CartItemViewModel:

CartViewModel:

using Microsoft.AspNetCore.Mvc;

namespace VehicleRentalService.Models.ViewModels

{

public class CartViewModel

{

public List<CartItemViewModel> Items { get; set; } = new();

public DateTime TripBegin { get; set; }

public DateTime TripEnd { get; set; }

public TimeSpan Duration { get; set; }

public decimal Total { get; set; }

}

}

AddToCartViewModel:

using Microsoft.AspNetCore.Mvc;

namespace VehicleRentalService.Models.ViewModels

{

public class AddToCartViewModel

{

public long VehicleId { get; set; }

public VehicleType VehicleType { get; set; }

public int Quantity { get; set; } = 1;

}

}

CartItemViewModel:

namespace VehicleRentalService.Models.ViewModels

{

public class CartItemViewModel

{

public long Id { get; set; }

public string VehicleName { get; set; } = "";

public VehicleType VehicleType { get; set; }

public int Quantity { get; set; }

public decimal PricePerMinute { get; set; }

public decimal Subtotal { get; set; }

}

}

Висновки: під час виконання Л/Р 3 було отримано навичок роботи з частковими представленнями, навчився працювати з параметрами рядку запиту, набув навичок роботи з сесіями.