Міністерство освіти України

Національний технічний університет "ХПІ"

кафедра "Комп’ютерних наук та інформаційних технологій"

**Звіт**

**Лабораторна робота 2**

з дисципліни "Проєктування серверних застосунків"

Виконав: студент групи КН-922В

Мельник Б. В.

Перевірив:

Івашко А. В.

Харків 2025

**Зміст**

[Пояснення роботи файлів: 1](#_Toc193289622)

[Завдання 1 3](#_Toc193289623)

[Текст програми: 3](#_Toc193289624)

[Результат роботи програми: 15](#_Toc193289625)

[Алгоритм побудови проєкту і кодів: 16](#_Toc193289626)

[Функціональність: 16](#_Toc193289627)

[Висновок 16](#_Toc193289628)

[Завдання 2 18](#_Toc193289629)

[Текст програми: 18](#_Toc193289630)

[Результат роботи програми: 30](#_Toc193289631)

[Алгоритм побудови проєкту і кодів: 31](#_Toc193289632)

[Функціональність: 31](#_Toc193289633)

[Висновок: 31](#_Toc193289634)

[Завдання 3 32](#_Toc193289635)

[Текст програми: 32](#_Toc193289636)

[Результат роботи програми: 43](#_Toc193289637)

[Алгоритм побудови проєкту і кодів: 44](#_Toc193289638)

[Функціональність: 45](#_Toc193289639)

[Висновок: 45](#_Toc193289640)

[\*Завдання 4: 46](#_Toc193289641)

[Текст програми: 46](#_Toc193289642)

[Результат роботи програми: 57](#_Toc193289643)

[Алгоритм побудови проєкту і кодів: 58](#_Toc193289644)

[Функціональність: 59](#_Toc193289645)

[Висновок: 59](#_Toc193289646)

[Завдання 5 60](#_Toc193289647)

[Текст програми: 60](#_Toc193289648)

[Результат роботи програми: 71](#_Toc193289649)

[Алгоритм побудови проєкту і кодів: 72](#_Toc193289650)

[Функціональність: 73](#_Toc193289651)

[Висновок: 73](#_Toc193289652)

[Завдання 6 74](#_Toc193289653)

[Текст програми: 74](#_Toc193289654)

[Результат роботи програми: 84](#_Toc193289655)

[Алгоритм побудови проєкту і кодів: 85](#_Toc193289656)

[Функціональність: 86](#_Toc193289657)

[Висновок: 86](#_Toc193289658)

[Завдання 7 87](#_Toc193289659)

[Текст програми: 87](#_Toc193289660)

[Результат роботи програми: 97](#_Toc193289661)

[Алгоритм побудови проєкту і кодів: 98](#_Toc193289662)

[Функціональність: 99](#_Toc193289663)

[Висновок: 99](#_Toc193289664)

[\*Завдання 8 100](#_Toc193289665)

[Текст програми: 100](#_Toc193289666)

[Результати роботи програми: 110](#_Toc193289667)

[Алгоритм побудови проєкту і кодів: 111](#_Toc193289668)

[Функціональність: 111](#_Toc193289669)

[Висновок: 112](#_Toc193289670)

[Завдання 9 113](#_Toc193289671)

[Текст програми: 113](#_Toc193289672)

[Результат роботи програми: 123](#_Toc193289673)

[Алгоритм побудови проєкту і кодів: 124](#_Toc193289674)

[Функціональність: 125](#_Toc193289675)

[Висновок: 125](#_Toc193289676)

[Завдання 10 126](#_Toc193289677)

[Текст програми: 126](#_Toc193289678)

[Результат роботи програми: 136](#_Toc193289679)

[Алгоритм побудови проєкту і кодів: 137](#_Toc193289680)

[Функціональність: 138](#_Toc193289681)

[Висновок: 138](#_Toc193289682)

[Завдання 11 139](#_Toc193289683)

[Текст програми: 139](#_Toc193289684)

[Результат роботи програми: 149](#_Toc193289685)

[Алгоритм побудови проєкту і кодів: 150](#_Toc193289686)

[Функціональність: 151](#_Toc193289687)

[Висновок: 151](#_Toc193289688)

[\*Завдання 12 152](#_Toc193289689)

[Текст програми: 152](#_Toc193289690)

[Результат роботи програми: 162](#_Toc193289691)

[Алгоритм побудови проєкту і кодів: 163](#_Toc193289692)

[Функціональність: 164](#_Toc193289693)

[Висновок: 164](#_Toc193289694)

[Завдання 13 165](#_Toc193289695)

[Текст програми: 165](#_Toc193289696)

[Результати роботи програми: 173](#_Toc193289697)

[Алгоритм побудови проєкту і кодів: 174](#_Toc193289698)

[Функціональність: 175](#_Toc193289699)

[Висновок: 175](#_Toc193289700)

[Завдання 14 176](#_Toc193289701)

[Текст програми: 176](#_Toc193289702)

[Результати роботи програми: 184](#_Toc193289703)

[Алгоритм побудови проєкту і кодів: 185](#_Toc193289704)

[Функціональність: 185](#_Toc193289705)

[Висновок: 186](#_Toc193289706)

[Завдання 15 187](#_Toc193289707)

[Текст програми: 187](#_Toc193289708)

[Результати роботи програми: 195](#_Toc193289709)

[Алгоритм побудови проєкту і кодів: 196](#_Toc193289710)

[Функціональність: 196](#_Toc193289711)

[Висновок: 196](#_Toc193289712)

[\*Завдання 16 197](#_Toc193289713)

[Текст програми: 197](#_Toc193289714)

[Результати роботи програми: 205](#_Toc193289715)

[Алгоритм побудови проєкту і кодів: 206](#_Toc193289716)

[Функціональність: 207](#_Toc193289717)

[Висновок: 207](#_Toc193289718)

[Загальний Висновок 208](#_Toc193289719)

**Тема:** Шляхи використання можливостей форм у засобах ASP.NET MVC WEB-застосунків. Частина 1.

**Мета роботи:** набути практику та досвід у використанні різноманітних шля-хів отримання даних із форм на поданнях під час розробки та використання MVC WEB-застосунків.

**Програмні засоби розробки:** середовище програмування (IDE-Integrated Development Environment) MS Visual Studio, програмний фреймворк ASP.NET версії не нижче 6.0 та дуже бажано використання дизайнерського фреймво-рку BOOTSTRAP, або якогось іншого відповідного аналогу, MySQL Server, або якась аналогічна СУБД.

**Індивідуальні завдання**

**Варіант №12**

# **Пояснення роботи файлів:**

* **Index.cshtml:**

Це головний вигляд (View), який містить HTML-розмітку форми. У ньому розміщено елементи введення (текстові поля, чекбокси, комбобокси, списки тощо), а також логіка для відображення результатів (через ViewBag або модель).

* **HomeController.cs:**

Контролер, що обробляє HTTP-запити до застосунку. Він містить методи GET (повертає порожню форму або початкові дані) та POST (обробляє дані форми, зчитує їх через Request, IFormCollection або через параметри/модель, а потім передає дані назад до вигляду).

* **\_Layout.cshtml:**

Спільний шаблон (Layout) для всіх сторінок сайту. Містить загальну розмітку: шапку (header), навігаційне меню, місце для основного контенту (RenderBody) та футер. Забезпечує єдиний стиль оформлення для всіх сторінок.

* **site.css:**

Глобальний файл стилів (CSS) для сайту, що містить загальні правила оформлення: розміри шрифту, відступи, кольори, стилі для елементів, фокус, тощо. Використовується для забезпечення єдиного зовнішнього вигляду застосунку.

* **PersonModel.cs (Завдання 4):**

Модель, що описує дані особи (наприклад, ім’я, телефон, електронна пошта, дата народження). Використовується для суворої типізації форми, щоб автоматично прив’язувати введені дані до властивостей моделі.

* **SelectionModel.cs (Завдання 8):**

Модель, яка містить властивості для вибору (наприклад, місто та транспорт). Вона використовується для суворої типізації даних форми, забезпечуючи автоматичну прив’язку даних чекбоксів або полів введення до моделі.

* **InterestsModel.cs (Завдання 12):**

Модель для чекбоксів, що відображає інтереси (наприклад, Спорт, Мандрівки, Майстрування, Малювання) із встановленими дефолтними значеннями. Забезпечує сувору типізацію даних форми при обробці вибору інтересів.

* **CitySelectionModel.cs (Завдання 16):**

Модель, що містить властивості для вибору міста із списків (наприклад, ComboCity та ListCity). Вона використовується для суворої типізації даних, що вводяться через комбобокс і листбокс, що дозволяє легко управляти вибором користувача.

# **Завдання 1**

Створити застосунок, стартове подання (Index) якого містить форму згідно до власного варіанту. За допомогою стандартного класу Request контролер, викликаний кнопкою Відіслати приймає дані з форми та пе-редає їх назад до подання для їхньої демонстрації під формою, кнопка Скинути має повертати форму у «дефолтний» (початковий) стан, тобто очистити текстові поля.

|  |  |
| --- | --- |
| № варіанту | Вигляд форми |
| 1 | 2 |
| 12 |  |

## **Текст програми:**

**Index.cshtml:**

@{

// Встановлюємо Layout і заголовок сторінки

Layout = "\_Layout";

ViewData["Title"] = "Завдання 1 (Варіант 12) — Абсолютне позиціонування";

}

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Завдання 1 (Варіант 12) — Абсолютне позиціонування</title>

<style>

/\* Стилі для контейнера форми \*/

.form-container {

position: relative; /\* щоб позиціонувати дочірні елементи \*/

width: 600px;

height: 400px;

border: 1px solid #ccc; /\* простий бордер \*/

margin-bottom: 20px;

padding: 10px;

background: rgba(255, 255, 255, 0.8); /\* напівпрозорий фон \*/

}

/\* Стилі для міток \*/

.labelName, .labelPhone, .labelEmail, .labelBirthdate {

font-weight: bold;

color: #333;

}

.labelName {

position: absolute;

left: 20px;

top: 20px;

}

.labelPhone {

position: absolute;

left: 20px;

top: 60px;

}

.labelEmail {

position: absolute;

left: 50px;

top: 100px;

}

.labelBirthdate {

position: absolute;

left: 60px;

top: 140px;

}

/\* Стилі для кнопок \*/

.buttonSubmit, .buttonReset {

border: none;

border-radius: 5px;

color: #fff;

padding: 8px 16px;

font-weight: bold;

cursor: pointer;

transition: transform 0.3s ease, box-shadow 0.3s ease;

}

.buttonSubmit {

position: absolute;

left: 180px;

top: 20px;

background: linear-gradient(45deg, #007bff, #00aaff); /\* градієнт \*/

}

.buttonSubmit:hover {

transform: scale(1.05);

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

}

.buttonReset {

position: absolute;

left: 260px;

top: 190px;

background: linear-gradient(45deg, #dc3545, #e4606d);

}

.buttonReset:hover {

transform: scale(1.05);

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

}

/\* Стилі для полів введення \*/

.inputName, .inputPhone, .inputEmail, .inputBirthdate {

position: absolute;

width: 200px;

padding: 6px;

border: 1px solid #ccc;

border-radius: 5px;

transition: transform 0.3s ease, border-color 0.3s ease;

}

.inputName {

left: 305px;

top: 20px;

}

.inputPhone {

left: 310px;

top: 60px;

}

.inputEmail {

left: 315px;

top: 100px;

}

.inputBirthdate {

left: 315px;

top: 140px;

}

.inputName:focus, .inputPhone:focus, .inputEmail:focus, .inputBirthdate:focus {

border-color: #007bff;

transform: scale(1.02);

}

</style>

</head>

<body>

<div class="form-container">

<!-- Форма для введення даних користувача -->

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

<div class="labelName">Ім’я</div>

<button type="submit" name="action" value="submit" class="buttonSubmit">Відіслати</button>

<input type="text" name="Name" value="@ViewBag.Name" class="inputName" />

<div class="labelPhone">Телефон</div>

<input type="text" name="Phone" value="@ViewBag.Phone" class="inputPhone" />

<div class="labelEmail">Поштова скринька</div>

<input type="text" name="Email" value="@ViewBag.Email" class="inputEmail" />

<div class="labelBirthdate">Дата народження</div>

<input type="date" name="Birthdate" value="@ViewBag.Birthdate" class="inputBirthdate" />

<button type="submit" name="action" value="reset" class="buttonReset">Скинути</button>

</form>

</div>

<!-- Відображення введених даних, якщо вони є -->

@if (!string.IsNullOrEmpty(ViewBag.Name)

|| !string.IsNullOrEmpty(ViewBag.Phone)

|| !string.IsNullOrEmpty(ViewBag.Email)

|| !string.IsNullOrEmpty(ViewBag.Birthdate))

{

<hr />

<h3>Введені дані:</h3>

<p>Ім’я: @ViewBag.Name</p>

<p>Телефон: @ViewBag.Phone</p>

<p>Поштова скринька: @ViewBag.Email</p>

<p>Дата народження: @ViewBag.Birthdate</p>

}

</body>

</html>

**HomeController.cs:**

using Microsoft.AspNetCore.Mvc;

namespace Task\_1.Controllers

{

public class HomeController : Controller

{

[HttpGet]

public IActionResult Index()

{

// Повертаємо порожню форму при першому завантаженні сторінки

return View();

}

[HttpPost]

public IActionResult Index(string action)

{

// Визначаємо, яка кнопка була натиснута

if (action == "reset")

{

// Якщо "Скинути", очищуємо дані

ViewBag.Name = "";

ViewBag.Phone = "";

ViewBag.Email = "";

ViewBag.Birthdate = "";

}

else

{

// Зчитуємо дані з форми

string name = Request.Form["Name"];

string phone = Request.Form["Phone"];

string email = Request.Form["Email"];

string birthdate = Request.Form["Birthdate"];

// Передаємо дані у ViewBag для виведення на сторінці

ViewBag.Name = name;

ViewBag.Phone = phone;

ViewBag.Email = email;

ViewBag.Birthdate = birthdate;

}

// Повертаємо сторінку з даними або очищеною формою

return View();

}

}

}

**\_Layout.cshtml:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Task\_1</title>

<!-- Підключення Bootstrap і власного CSS -->

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

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

</head>

<body>

<header>

<!-- Навігаційне меню сайту -->

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">

<div class="container-fluid">

<!-- Назва/логотип сайту -->

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Task\_1</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse"

data-bs-target=".navbar-collapse" aria-controls="navbarSupportedContent"

aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<!-- Основний контейнер для контенту сторінки -->

<div class="container">

<main role="main" class="pb-3">

@RenderBody() <!-- Динамічно підставляємо вміст конкретної сторінки -->

</main>

</div>

<!-- Футер сайту -->

<footer class="border-top footer text-muted">

<div class="container">

&copy; 2025 - Task\_1 - <a asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</div>

</footer>

<!-- Підключення JavaScript файлів -->

<script src="~/lib/jquery/dist/jquery.min.js"></script>

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

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

**site.css:**

/\* Скидання відступів для html та body \*/

html, body {

margin: 0;

padding: 0;

height: 100%;

}

/\* Робимо body flex-контейнером і задаємо фон \*/

body {

display: flex;

flex-direction: column;

/\* Фонове зображення \*/

background: url('https://img.goodfon.ru/original/1920x1080/2/fe/lake-tahoe-nevada-usa-lake-trees-sunrise-mountains-clear-wat.jpg') no-repeat center center fixed;

background-size: cover;

}

/\* Напівпрозорий шар для кращої читабельності \*/

body::before {

content: "";

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: rgba(255, 255, 255, 0.3);

z-index: -1;

}

/\* Контейнер для основного вмісту \*/

.container {

flex: 1 0 auto;

}

/\* Футер завжди залишається внизу \*/

footer {

flex-shrink: 0;

}

/\* Базовий розмір шрифту \*/

html {

font-size: 14px;

}

@media (min-width: 768px) {

html {

font-size: 16px;

}

}

/\* Стилі для фокуса елементів (кнопок, полів введення) \*/

.btn:focus, .btn:active:focus, .btn-link.nav-link:focus,

.form-control:focus, .form-check-input:focus {

box-shadow: 0 0 0 0.1rem white, 0 0 0 0.25rem #258cfb;

}

/\* Стилі для кнопок із градієнтом \*/

button, .btn {

background: linear-gradient(45deg, #007bff, #00aaff);

border: none;

border-radius: 5px;

color: #fff;

padding: 8px 16px;

font-weight: bold;

text-shadow: 1px 1px 2px rgba(0,0,0,0.3);

transition: transform 0.3s ease, box-shadow 0.3s ease, background 0.3s ease;

}

/\* Ефект при наведенні на кнопки \*/

button:hover, .btn:hover {

transform: scale(1.05);

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

background: linear-gradient(45deg, #0056b3, #007bff);

}

/\* Ефект при натисканні на кнопки \*/

button:active, .btn:active {

transform: scale(0.98);

}

/\* Стилі для полів введення \*/

input.form-control {

border-radius: 5px;

border: 1px solid #ccc;

transition: border-color 0.3s ease;

}

/\* Зміна рамки при фокусі \*/

input.form-control:focus {

border-color: #007bff;

}

input.form-control,

textarea.form-control,

select.form-control {

border: 1px solid #ccc;

border-radius: 5px;

padding: 8px;

transition: transform 0.3s ease, border-color 0.3s ease, box-shadow 0.3s ease;

}

/\* Трохи збільшуємо поле при наведенні \*/

input.form-control:hover,

textarea.form-control:hover,

select.form-control:hover {

transform: scale(1.02);

}

/\* Ефект фокусу на полях введення \*/

input.form-control:focus,

textarea.form-control:focus,

select.form-control:focus {

border-color: #007bff;

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

transform: scale(1.03);

}

/\* Стилі для футера \*/

footer.footer,

footer.footer .container,

footer.footer a {

color: green !important;

}

footer.footer,

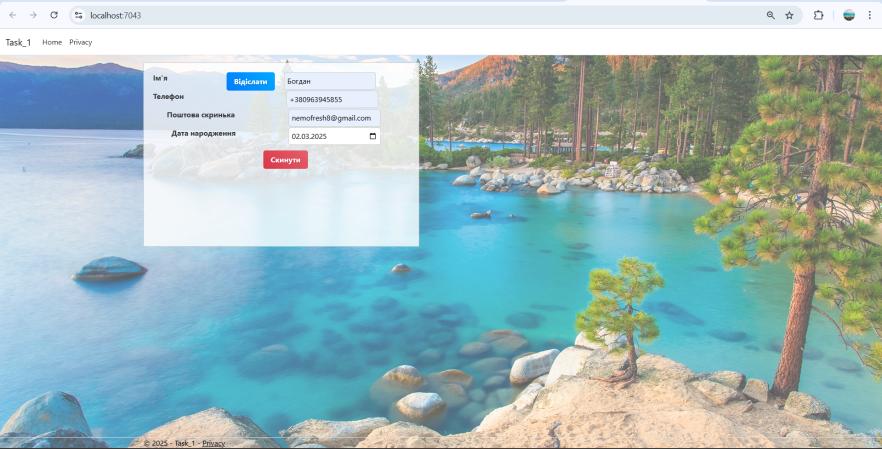
footer.footer a,

footer.footer .container {

color: black !important;

}

## **Результат роботи програми:**



Изображение выглядит как вода, текст, снимок экрана, озеро

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как вода, снимок экрана, дерево, Лазурь

Контент, сгенерированный ИИ, может содержать ошибки.

## **Алгоритм побудови проєкту і кодів:**

1. **Створити новий ASP.NET Core MVC-проєкт** (або додати в наявний).
2. **Додати контролер** HomeController із двома методами Index:
   * HttpGet повертає порожній вигляд з формою.
   * HttpPost обробляє кнопки:
     + Якщо натиснуто «Скинути» – обнулюємо значення.
     + Якщо натиснуто «Відіслати» – зчитуємо з Request.Form дані та записуємо у ViewBag.
3. **Створити вигляд Index.cshtml** з формою, розміткою (абсолютне позиціонування) та двома кнопками:
   * **Відіслати (submit)** – надсилає форму в HomeController.
   * **Скинути (reset)** – повертає порожній стан полів.
4. **Додати/налаштувати \_Layout.cshtml** (головний шаблон) для спільних стилів, шапки та футера.
5. **Підключити стилі (Bootstrap, site.css)** для оформлення, у тому числі фонового зображення та анімації наведення.
6. **Перевірити роботу**: при заповненні форми та натисканні «Відіслати» дані відображаються під формою; при натисканні «Скинути» – поля очищуються.

## **Функціональність:**

* Застосунок відображає форму з полями: Ім’я, Телефон, Поштова скринька, Дата народження.
* При натисканні «Відіслати» введені дані відправляються до контролера і повертаються назад у вигляд для відображення.
* Кнопка «Скинути» очищує всі поля форми.

## **Висновок**

У результаті отримали простий ASP.NET Core MVC-застосунок, що демонструє:

* Роботу з формами (заповнення та надсилання).
* Використання Request.Form для зчитування даних у контролері.
* Виведення зібраної інформації на сторінці.
* Очищення форми до початкового стану. Це приклад базової взаємодії між користувачем і сервером у MVC-додатку.

# **Завдання 2**

Виконати функції минулого завдання, використавши для отримання даних форми замість класу Request колекцію типу IFormCollection.

## **Текст програми:**

**Index.cshtml:**

@{

Layout = "\_Layout"; // Використовуємо спільний Layout

}

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Завдання 2 (Варіант 12) — Абсолютне позиціонування</title>

<style>

/\* Контейнер форми – відносне позиціонування для внутрішніх елементів \*/

.form-container {

position: relative;

width: 600px; /\* Фіксована ширина \*/

height: 400px; /\* Фіксована висота \*/

border: 1px solid #ccc;

margin-bottom: 20px;

padding: 10px;

background: rgba(255, 255, 255, 0.8); /\* Легкий фон для контрасту \*/

}

/\* Заголовки та мітки для полів форми \*/

.labelName, .labelPhone, .labelEmail, .labelBirthdate {

font-weight: bold;

color: #333;

}

.labelName {

position: absolute;

left: 20px;

top: 20px;

}

.labelPhone {

position: absolute;

left: 20px;

top: 50px;

}

.labelEmail {

position: absolute;

left: 50px;

top: 80px;

}

.labelBirthdate {

position: absolute;

left: 60px;

top: 110px;

}

/\* Стилі для кнопок із анімацією \*/

.buttonSubmit, .buttonReset {

border: none;

border-radius: 5px;

color: #fff;

padding: 8px 16px;

font-weight: bold;

cursor: pointer;

transition: transform 0.3s ease, box-shadow 0.3s ease;

}

.buttonSubmit {

position: absolute;

left: 180px;

top: 20px;

background-color: #007bff; /\* Синій фон \*/

}

.buttonSubmit:hover {

transform: scale(1.05);

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

}

.buttonReset {

position: absolute;

left: 260px;

top: 190px;

background: linear-gradient(45deg, #dc3545, #e4606d);

}

.buttonReset:hover {

transform: scale(1.05);

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

}

/\* Стилі для полів введення даних \*/

.inputName, .inputPhone, .inputEmail, .inputBirthdate {

position: absolute;

width: 200px;

padding: 6px;

border: 1px solid #ccc;

border-radius: 5px;

transition: transform 0.3s ease, border-color 0.3s ease;

}

.inputName {

left: 305px;

top: 20px;

}

.inputPhone {

left: 310px;

top: 60px;

}

.inputEmail {

left: 315px;

top: 100px;

}

.inputBirthdate {

left: 315px;

top: 140px;

}

/\* Ефект при фокусі на полях введення \*/

.inputName:focus, .inputPhone:focus, .inputEmail:focus, .inputBirthdate:focus {

border-color: #007bff;

transform: scale(1.02);

}

</style>

</head>

<body>

<!-- Форма для вводу даних -->

<form method="post" asp-action="Index" asp-controller="Home" class="form-container">

<div class="labelName">Ім’я</div>

<button type="submit" name="action" value="submit" class="buttonSubmit">

Відіслати

</button>

<input type="text" name="Name" value="@ViewBag.Name" class="inputName" />

<div class="labelPhone">Телефон</div>

<input type="text" name="Phone" value="@ViewBag.Phone" class="inputPhone" />

<div class="labelEmail">Поштова скринька</div>

<input type="text" name="Email" value="@ViewBag.Email" class="inputEmail" />

<div class="labelBirthdate">Дата народження</div>

<input type="date" name="Birthdate" value="@ViewBag.Birthdate" class="inputBirthdate" />

<button type="submit" name="action" value="reset" class="buttonReset">

Скинути

</button>

</form>

<!-- Вивід даних, якщо вони введені -->

@if (!string.IsNullOrEmpty(ViewBag.Name)

|| !string.IsNullOrEmpty(ViewBag.Phone)

|| !string.IsNullOrEmpty(ViewBag.Email)

|| !string.IsNullOrEmpty(ViewBag.Birthdate))

{

<hr />

<h3>Введені дані:</h3>

<p>Ім’я: @ViewBag.Name</p>

<p>Телефон: @ViewBag.Phone</p>

<p>Поштова скринька: @ViewBag.Email</p>

<p>Дата народження: @ViewBag.Birthdate</p>

}

</body>

</html>

**HomeController.cs:**

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Http; // Потрібно для IFormCollection

namespace Task\_2.Controllers

{

public class HomeController : Controller

{

// Обробка GET-запиту для сторінки

[HttpGet]

public IActionResult Index()

{

// Просто повертаємо View без даних

return View();

}

// Обробка POST-запиту з форми

[HttpPost]

public IActionResult Index(IFormCollection form, string action)

{

if (action == "reset")

{

// Якщо натиснули кнопку "Скинути", очищуємо дані

ViewBag.Name = "";

ViewBag.Phone = "";

ViewBag.Email = "";

ViewBag.Birthdate = "";

}

else

{

// Зчитуємо дані з форми

string name = form["Name"];

string phone = form["Phone"];

string email = form["Email"];

string birthdate = form["Birthdate"];

// Записуємо дані у ViewBag для виведення

ViewBag.Name = name;

ViewBag.Phone = phone;

ViewBag.Email = email;

ViewBag.Birthdate = birthdate;

}

// Повертаємо View з оновленими даними

return View();

}

}

}

**\_Layout.cshtml:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Task\_2</title>

<!-- Підключення Bootstrap та основних стилів -->

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

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

<link rel="stylesheet" href="~/Task\_2.styles.css" asp-append-version="true" />

</head>

<body>

<header>

<!-- Навігаційне меню сайту -->

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">

<div class="container-fluid">

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Task\_2</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse"

aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<!-- Основний контейнер для динамічного контенту -->

<div class="container">

<main role="main" class="pb-3">

@RenderBody() <!-- Тут буде вміст конкретної сторінки -->

</main>

</div>

<!-- Футер сайту -->

<footer class="border-top footer text-muted">

<div class="container">

&copy; 2025 - Task\_2 - <a asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</div>

</footer>

<!-- Підключення JavaScript файлів -->

<script src="~/lib/jquery/dist/jquery.min.js"></script>

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

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

**site.css:**

/\* Встановлюємо базовий розмір шрифту \*/

html {

font-size: 14px;

}

@media (min-width: 768px) {

html {

font-size: 16px; /\* Збільшуємо шрифт на більших екранах \*/

}

}

/\* Стилі для фокусу елементів \*/

.btn:focus, .btn:active:focus, .btn-link.nav-link:focus,

.form-control:focus, .form-check-input:focus {

box-shadow: 0 0 0 0.1rem white, 0 0 0 0.25rem #258cfb;

}

/\* Забезпечуємо мінімальну висоту для html \*/

html {

position: relative;

min-height: 100%;

}

/\* Фонове зображення для body \*/

body {

margin-bottom: 60px;

background: url('https://img.goodfon.ru/original/1920x1080/2/fe/lake-tahoe-nevada-usa-lake-trees-sunrise-mountains-clear-wat.jpg') no-repeat center center fixed;

background-size: cover;

}

/\* Напівпрозорий шар для покращення читабельності тексту \*/

body::before {

content: "";

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: rgba(255, 255, 255, 0.3); /\* Прозорість налаштована для кращого фону \*/

z-index: -1;

}

/\* Стилізація заголовків \*/

h1 {

color: #333;

}

/\* Мітки форм \*/

.form-label {

font-weight: bold;

}

/\* Стилі для кнопок із градієнтом та анімацією \*/

button, .btn {

background: linear-gradient(45deg, #007bff, #00aaff);

border: none;

border-radius: 5px;

color: #fff;

padding: 8px 16px;

font-weight: bold;

text-shadow: 1px 1px 2px rgba(0,0,0,0.3);

transition: transform 0.3s ease, box-shadow 0.3s ease, background 0.3s ease;

}

/\* Ефект при наведенні на кнопку \*/

button:hover, .btn:hover {

transform: scale(1.05);

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

background: linear-gradient(45deg, #0056b3, #007bff);

}

/\* Ефект при натисканні на кнопку \*/

button:active, .btn:active {

transform: scale(0.98);

}

/\* Стилі для полів введення \*/

input.form-control {

border-radius: 5px;

border: 1px solid #ccc;

transition: border-color 0.3s ease;

}

input.form-control:focus {

border-color: #007bff;

}

/\* Додаткове стилізування для полів форми \*/

input.form-control,

textarea.form-control,

select.form-control {

border: 1px solid #ccc;

border-radius: 5px;

padding: 8px;

transition: transform 0.3s ease, border-color 0.3s ease, box-shadow 0.3s ease;

}

/\* Ефект при наведенні на поле введення \*/

input.form-control:hover,

textarea.form-control:hover,

select.form-control:hover {

transform: scale(1.02);

}

/\* Ефект при фокусі на полі введення \*/

input.form-control:focus,

textarea.form-control:focus,

select.form-control:focus {

border-color: #007bff;

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

transform: scale(1.03);

}

/\* Стилі для футера \*/

footer.footer,

footer.footer .container,

footer.footer a {

color: black !important;

}

## **Результат роботи програми:**

Изображение выглядит как вода, снимок экрана, озеро, дерево

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как вода, текст, озеро, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как вода, снимок экрана, дерево, озеро

Контент, сгенерированный ИИ, может содержать ошибки.

## **Алгоритм побудови проєкту і кодів:**

1. **Створити або відкрити ASP.NET Core MVC-проєкт.**
2. **Додати контролер** HomeController з методами Index:
   * HttpGet: просто повертає порожній вигляд (форму).
   * HttpPost: приймає параметр типу IFormCollection form і зчитує з нього дані (замість Request.Form). Якщо натиснуто «Скинути» – обнуляємо поля, інакше зчитуємо та передаємо у ViewBag.
3. **Створити вигляд** Index.cshtml, в якому:
   * Розміщено форму з полями (Ім’я, Телефон, Поштова скринька, Дата народження) та двома кнопками «Відіслати» і «Скинути».
   * Виводимо введені дані під формою, якщо вони не порожні.
4. **Налаштувати \_Layout.cshtml** (якщо потрібно) для загального оформлення та підключення стилів.
5. **Перевірити роботу**: при натисканні «Відіслати» дані відображаються, при «Скинути» – поля очищуються.

## **Функціональність:**

* Застосунок відображає форму з полями для введення особистих даних.
* Дані зчитуються за допомогою IFormCollection у контролері.
* Користувач може надіслати (і побачити введене) або скинути (очистити) інформацію.

## **Висновок:**

Реалізовано ту саму логіку, що і в попередньому завданні, але замість Request використано колекцію IFormCollection для отримання даних з форми. Це демонструє різні способи обробки POST-запитів в ASP.NET Core MVC без зміни основного функціоналу.

# **Завдання 3**

Виконати функції першого завдання, використавши для отримання даних форми замість класу Request слабку типізацію форми, додавши до кожного поля форми відповідну змінну.

## **Текст програми:**

**Index.cshtml:**

@{

Layout = "\_Layout"; // Використовуємо спільний макет

ViewData["Title"] = "Завдання 3 (Варіант 12) — Абсолютне позиціонування"; // Встановлюємо заголовок сторінки

}

<!-- Стилі для форми (можна винести в окремий CSS-файл) -->

<style>

/\* Контейнер форми – всі дочірні елементи позиціонуються відносно нього \*/

.form-container {

position: relative;

width: 600px; /\* Фіксована ширина контейнера \*/

height: 400px; /\* Фіксована висота контейнера \*/

border: 1px solid #ccc; /\* Тонка рамка \*/

margin-bottom: 20px; /\* Відступ знизу \*/

padding: 10px; /\* Внутрішній відступ \*/

background: rgba(255, 255, 255, 0.8); /\* Напівпрозорий фон \*/

}

/\* Стилі для міток, які позначають поля форми \*/

.labelName, .labelPhone, .labelEmail, .labelBirthdate {

font-weight: bold;

color: #333;

}

.labelName {

position: absolute;

left: 20px;

top: 20px;

}

.labelPhone {

position: absolute;

left: 20px;

top: 50px;

}

.labelEmail {

position: absolute;

left: 50px;

top: 80px;

}

.labelBirthdate {

position: absolute;

left: 60px;

top: 110px;

}

/\* Стилі для кнопок з анімацією \*/

.buttonSubmit, .buttonReset {

border: none;

border-radius: 5px;

color: #fff;

padding: 8px 16px;

font-weight: bold;

cursor: pointer; /\* Зміна курсора при наведенні \*/

transition: transform 0.3s ease, box-shadow 0.3s ease; /\* Анімація для приємного ефекту \*/

}

.buttonSubmit {

position: absolute;

left: 180px;

top: 20px;

background-color: #007bff; /\* Синій фон для кнопки відправки \*/

}

.buttonSubmit:hover {

transform: scale(1.05); /\* Збільшення кнопки при наведенні \*/

box-shadow: 0 4px 8px rgba(0,0,0,0.2); /\* Легка тінь \*/

}

.buttonReset {

position: absolute;

left: 260px;

top: 190px;

background: linear-gradient(45deg, #dc3545, #e4606d); /\* Градієнт для кнопки скидання \*/

}

.buttonReset:hover {

transform: scale(1.05);

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

}

/\* Стилі для полів введення даних \*/

.inputName, .inputPhone, .inputEmail, .inputBirthdate {

position: absolute;

width: 200px; /\* Фіксована ширина полів \*/

padding: 6px;

border: 1px solid #ccc; /\* Рамка для виділення поля \*/

border-radius: 5px;

transition: transform 0.3s ease, border-color 0.3s ease; /\* Анімація при фокусі \*/

}

.inputName {

left: 305px;

top: 20px;

}

.inputPhone {

left: 310px;

top: 60px;

}

.inputEmail {

left: 315px;

top: 100px;

}

.inputBirthdate {

left: 315px;

top: 140px;

}

/\* Зміна стилю поля при фокусі \*/

.inputName:focus, .inputPhone:focus, .inputEmail:focus, .inputBirthdate:focus {

border-color: #007bff;

transform: scale(1.02); /\* Невелике збільшення для візуального ефекту \*/

}

</style>

<!-- Форма з абсолютним позиціонуванням -->

<div class="form-container">

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

<!-- Поле для введення імені -->

<div class="labelName">Ім’я</div>

<!-- Кнопка для відправки форми -->

<button type="submit" name="action" value="submit" class="buttonSubmit">

Відіслати

</button>

<input type="text" name="Name" value="@ViewBag.Name" class="inputName" />

<!-- Поле для введення телефону -->

<div class="labelPhone">Телефон</div>

<input type="text" name="Phone" value="@ViewBag.Phone" class="inputPhone" />

<!-- Поле для введення поштової скриньки -->

<div class="labelEmail">Поштова скринька</div>

<input type="text" name="Email" value="@ViewBag.Email" class="inputEmail" />

<!-- Поле для введення дати народження -->

<div class="labelBirthdate">Дата народження</div>

<input type="date" name="Birthdate" value="@ViewBag.Birthdate" class="inputBirthdate" />

<!-- Кнопка для скидання форми -->

<button type="submit" name="action" value="reset" class="buttonReset">

Скинути

</button>

</form>

</div>

<!-- Відображення даних, якщо користувач щось ввів -->

@if (!string.IsNullOrEmpty(ViewBag.Name)

|| !string.IsNullOrEmpty(ViewBag.Phone)

|| !string.IsNullOrEmpty(ViewBag.Email)

|| !string.IsNullOrEmpty(ViewBag.Birthdate))

{

<hr />

<h3>Введені дані:</h3>

<p>Ім’я: @ViewBag.Name</p>

<p>Телефон: @ViewBag.Phone</p>

<p>Поштова скринька: @ViewBag.Email</p>

<p>Дата народження: @ViewBag.Birthdate</p>

}

**HomeController.cs:**

using Microsoft.AspNetCore.Mvc;

namespace Task\_3.Controllers

{

public class HomeController : Controller

{

// GET-запит: просто повертаємо форму без попередніх даних

[HttpGet]

public IActionResult Index()

{

return View();

}

// POST-запит: отримуємо дані з форми

// Параметри Name, Phone, Email, Birthdate відповідають іменам полів форми,

// а параметр action визначає, яку кнопку було натиснуто (submit або reset)

[HttpPost]

public IActionResult Index(string Name, string Phone, string Email, string Birthdate, string action)

{

if (action == "reset")

{

// Якщо натиснуто "Скинути", очищуємо всі поля

ViewBag.Name = "";

ViewBag.Phone = "";

ViewBag.Email = "";

ViewBag.Birthdate = "";

}

else

{

// Якщо натиснуто "Відіслати", зберігаємо отримані дані для подальшого відображення

ViewBag.Name = Name;

ViewBag.Phone = Phone;

ViewBag.Email = Email;

ViewBag.Birthdate = Birthdate;

}

return View(); // Повертаємо форму з оновленими даними

}

}

}

**\_Layout.cshtml:**

@\* Views/Shared/\_Layout.cshtml \*@

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Task\_3</title>

@RenderSection("Title", required: false) <!-- Розділ для додаткових заголовків, якщо потрібно -->

<!-- Підключення Bootstrap і основного CSS -->

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

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

</head>

<body>

<header>

<!-- Верхнє меню сайту -->

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">

<div class="container-fluid">

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Task\_3</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse"

data-bs-target=".navbar-collapse" aria-controls="navbarSupportedContent"

aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<!-- Основний контейнер для контенту -->

<div class="container">

<main role="main" class="pb-3">

@RenderBody() <!-- Тут відображається вміст конкретної сторінки -->

</main>

</div>

<!-- Футер сайту -->

<footer class="border-top footer text-muted">

<div class="container">

&copy; 2025 - Task\_3 -

<a asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</div>

</footer>

<!-- Підключення JavaScript бібліотек -->

<script src="~/lib/jquery/dist/jquery.min.js"></script>

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

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

**site.css:**

/\* Скидання відступів та встановлення висоти 100% для html і body \*/

html, body {

margin: 0;

padding: 0;

height: 100%;

}

/\* Використовуємо flex-контейнер для організації розміщення елементів на сторінці \*/

body {

display: flex;

flex-direction: column;

/\* Фонове зображення з адаптивним масштабуванням \*/

background: url('https://img.goodfon.ru/original/1920x1080/2/fe/lake-tahoe-nevada-usa-lake-trees-sunrise-mountains-clear-wat.jpg') no-repeat center center fixed;

background-size: cover;

}

/\* Напівпрозорий шар для покращення читабельності тексту \*/

body::before {

content: "";

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: rgba(255, 255, 255, 0.3);

z-index: -1;

}

/\* Контейнер для основного контенту, який розтягується \*/

.container {

flex: 1 0 auto;

}

/\* Футер завжди знаходиться внизу сторінки \*/

footer {

flex-shrink: 0;

}

/\* Базовий розмір шрифту для всієї сторінки \*/

html {

font-size: 14px;

}

@media (min-width: 768px) {

html {

font-size: 16px;

}

}

/\* Стилі для фокусу елементів (кнопок, полів введення) \*/

.btn:focus, .btn:active:focus, .btn-link.nav-link:focus,

.form-control:focus, .form-check-input:focus {

box-shadow: 0 0 0 0.1rem white, 0 0 0 0.25rem #258cfb;

}

/\* Стилі для кнопок із градієнтом і анімацією \*/

button, .btn {

background: linear-gradient(45deg, #007bff, #00aaff);

border: none;

border-radius: 5px;

color: #fff;

padding: 8px 16px;

font-weight: bold;

text-shadow: 1px 1px 2px rgba(0,0,0,0.3);

transition: transform 0.3s ease, box-shadow 0.3s ease, background 0.3s ease;

}

button:hover, .btn:hover {

transform: scale(1.05);

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

background: linear-gradient(45deg, #0056b3, #007bff);

}

button:active, .btn:active {

transform: scale(0.98);

}

/\* Стилі для полів введення з плавними переходами \*/

input.form-control {

border-radius: 5px;

border: 1px solid #ccc;

transition: border-color 0.3s ease;

}

input.form-control:focus {

border-color: #007bff;

}

input.form-control,

textarea.form-control,

select.form-control {

border: 1px solid #ccc;

border-radius: 5px;

padding: 8px;

transition: transform 0.3s ease, border-color 0.3s ease, box-shadow 0.3s ease;

}

input.form-control:hover,

textarea.form-control:hover,

select.form-control:hover {

transform: scale(1.02);

}

input.form-control:focus,

textarea.form-control:focus,

select.form-control:focus {

border-color: #007bff;

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

transform: scale(1.03);

}

/\* Стилі для футера – чорний текст \*/

footer.footer,

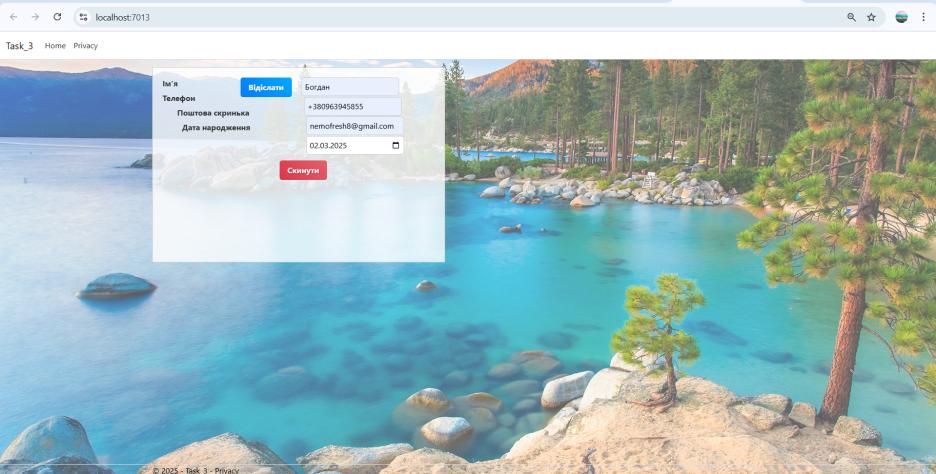
footer.footer a,

footer.footer .container {

color: black !important;

}

## **Результат роботи програми:**



Изображение выглядит как вода, текст, озеро, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как вода, снимок экрана, дерево, Лазурь

Контент, сгенерированный ИИ, может содержать ошибки.

## **Алгоритм побудови проєкту і кодів:**

1. **Створити або відкрити ASP.NET Core MVC-проєкт.**
2. **Додати контролер** HomeController з двома методами Index:
   * **GET**: просто повертає порожню форму (View).
   * **POST**: приймає як параметри змінні, назви яких відповідають іменам полів форми (string Name, string Phone, ...), а також параметр string action для визначення, яку кнопку натиснули («submit» чи «reset»).
3. **Якщо action == "reset"** – очищуємо поля, інакше зберігаємо значення у ViewBag.
4. **Створити вигляд** Index.cshtml, де:
   * Є форма з полями name="Name", name="Phone", name="Email", name="Birthdate".
   * Дві кнопки: «Відіслати» (submit) і «Скинути» (reset), що передаються в той самий метод Index (POST).
   * Виведення значень під формою, якщо вони не порожні.
5. **Налаштувати \_Layout.cshtml** і стилі (за потреби).
6. **Перевірити**: при відправленні форми дані відображаються, при скиданні – поля очищуються.

## **Функціональність:**

* Застосунок відображає форму з полями (Ім’я, Телефон, Пошта, Дата народження).
* Дані зчитуються як параметри методу (слабка типізація: string Name, тощо).
* Користувач може надіслати введені дані або очистити форму.

## **Висновок:**

Механізм прив’язки ASP.NET Core MVC дозволяє автоматично отримувати значення з полів форми, якщо назви параметрів у методі контролера збігаються з атрибутами name у вигляді. Це ще один спосіб обробки даних безпосередньо в дії контролера, не використовуючи Request чи IFormCollection.

# **\*Завдання 4:**

Виконати функції першого завдання, використавши для отримання даних форми замість класу Request сувору типізацію форми, додавши до полів форми відповідну модель.

## **Текст програми:**

**Index.cshtml:**

@model Task\_4.Models.PersonModel

@{

Layout = "\_Layout"; // Використовуємо спільний макет для сторінок

ViewData["Title"] = "Завдання 4 (Варіант 12) — Абсолютне позиціонування зі строгою типізацією"; // Встановлюємо заголовок сторінки

}

<style>

/\* Контейнер форми – тут розміщуються всі елементи, що позиціонуються абсолютно \*/

.form-container {

position: relative;

width: 600px; /\* Фіксована ширина \*/

height: 400px; /\* Фіксована висота \*/

border: 1px solid #ccc; /\* Легка рамка \*/

margin-bottom: 20px;

padding: 10px;

background: rgba(255, 255, 255, 0.8); /\* Напівпрозорий фон \*/

}

/\* Стилі для міток полів \*/

.labelName, .labelPhone, .labelEmail, .labelBirthdate {

font-weight: bold;

color: #333;

}

.labelName {

position: absolute;

left: 20px;

top: 20px;

}

.labelPhone {

position: absolute;

left: 20px;

top: 50px;

}

.labelEmail {

position: absolute;

left: 50px;

top: 80px;

}

.labelBirthdate {

position: absolute;

left: 60px;

top: 110px;

}

/\* Стилі для кнопок із анімацією \*/

.buttonSubmit, .buttonReset {

border: none;

border-radius: 5px;

color: #fff;

padding: 8px 16px;

font-weight: bold;

cursor: pointer; /\* Зміна курсора при наведенні \*/

transition: transform 0.3s ease, box-shadow 0.3s ease;

}

.buttonSubmit {

position: absolute;

left: 180px;

top: 20px;

background-color: #007bff; /\* Синій фон для відправки \*/

}

.buttonSubmit:hover {

transform: scale(1.05); /\* Збільшення кнопки при наведенні \*/

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

}

.buttonReset {

position: absolute;

left: 260px;

top: 190px;

background: linear-gradient(45deg, #dc3545, #e4606d); /\* Градієнт для кнопки скидання \*/

}

.buttonReset:hover {

transform: scale(1.05);

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

}

/\* Стилі для полів введення – абсолютне позиціонування полів \*/

.inputName, .inputPhone, .inputEmail, .inputBirthdate {

position: absolute;

width: 200px; /\* Фіксована ширина поля \*/

padding: 6px;

border: 1px solid #ccc; /\* Рамка для виділення поля \*/

border-radius: 5px;

transition: transform 0.3s ease, border-color 0.3s ease;

}

.inputName {

left: 305px;

top: 20px;

}

.inputPhone {

left: 310px;

top: 60px;

}

.inputEmail {

left: 315px;

top: 100px;

}

.inputBirthdate {

left: 315px;

top: 140px;

}

/\* Зміна стилю поля при фокусі – змінюється колір рамки та виконується невелике збільшення \*/

.inputName:focus, .inputPhone:focus, .inputEmail:focus, .inputBirthdate:focus {

border-color: #007bff;

transform: scale(1.02);

}

</style>

<div class="form-container">

<!-- Форма для введення даних із строго типізованою моделлю -->

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

<div class="labelName">Ім’я</div>

<button type="submit" name="action" value="submit" class="buttonSubmit">Відіслати</button>

<!-- Використовуємо asp-for для автоматичної прив'язки до моделі -->

<input type="text" asp-for="Name" class="inputName" />

<div class="labelPhone">Телефон</div>

<input type="text" asp-for="Phone" class="inputPhone" />

<div class="labelEmail">Поштова скринька</div>

<input type="text" asp-for="Email" class="inputEmail" />

<div class="labelBirthdate">Дата народження</div>

<input type="date" asp-for="Birthdate" class="inputBirthdate" />

<button type="submit" name="action" value="reset" class="buttonReset">Скинути</button>

</form>

</div>

<div>

<!-- Відображення введених даних з моделі -->

<h3>Введені дані:</h3>

<p>Ім’я: @Model.Name</p>

<p>Телефон: @Model.Phone</p>

<p>Поштова скринька: @Model.Email</p>

<p>Дата народження: @Model.Birthdate</p>

</div>

**HomeController.cs:**

using Microsoft.AspNetCore.Mvc;

using Task\_4.Models;

namespace Task\_4.Controllers

{

public class HomeController : Controller

{

// GET-запит: повертаємо форму з новою (порожньою) моделлю

[HttpGet]

public IActionResult Index()

{

// Створюємо новий об'єкт PersonModel із початковими значеннями

return View(new PersonModel());

}

// POST-запит: отримуємо дані з форми через строго типізовану модель

[HttpPost]

public IActionResult Index(PersonModel model, string action)

{

if (action == "reset")

{

// Якщо натиснуто "Скинути", очищуємо стан моделі та створюємо нову порожню модель

ModelState.Clear(); // Очищення помилок валідації, якщо були

model = new PersonModel();

}

// Повертаємо подання з отриманою або очищеною моделлю

return View(model);

}

}

}

**\_Layout.cshtml:**

@\* Views/Shared/\_Layout.cshtml \*@

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Task\_4</title>

<!-- Підключення Bootstrap та основного CSS -->

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

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

</head>

<body>

<header>

<!-- Верхнє меню сайту -->

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">

<div class="container-fluid">

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Task\_4</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse"

data-bs-target=".navbar-collapse" aria-controls="navbarSupportedContent"

aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<!-- Контейнер для основного вмісту -->

<div class="container">

<main role="main" class="pb-3">

@RenderBody() <!-- Тут вставляється контент конкретної сторінки -->

</main>

</div>

<!-- Футер сайту -->

<footer class="border-top footer text-muted">

<div class="container">

&copy; 2025 - Task\_4 - <a asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</div>

</footer>

<!-- Підключення JavaScript бібліотек -->

<script src="~/lib/jquery/dist/jquery.min.js"></script>

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

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

**PersonModel.cs:**

namespace Task\_4.Models

{

public class PersonModel

{

// Властивість для імені; за замовчуванням – порожній рядок

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

// Властивість для номера телефону

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

// Властивість для електронної пошти

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

// Властивість для дати народження (як рядок)

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

}

}

**site.css:**

/\* Скидання відступів та встановлення висоти 100% для html і body \*/

html, body {

margin: 0;

padding: 0;

height: 100%;

}

/\* Організовуємо елементи сторінки за допомогою flex-контейнера (вертикальна розкладка) \*/

body {

display: flex;

flex-direction: column;

background: url('https://img.goodfon.ru/original/1920x1080/2/fe/lake-tahoe-nevada-usa-lake-trees-sunrise-mountains-clear-wat.jpg') no-repeat center center fixed;

background-size: cover;

}

/\* Напівпрозорий шар для покращення читабельності тексту на фоні \*/

body::before {

content: "";

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: rgba(255, 255, 255, 0.3);

z-index: -1;

}

/\* Контейнер для основного вмісту, який займає всю доступну висоту \*/

.container {

flex: 1 0 auto;

}

/\* Футер завжди знаходиться внизу сторінки \*/

footer {

flex-shrink: 0;

}

/\* Встановлення базового розміру шрифту \*/

html {

font-size: 14px;

}

@media (min-width: 768px) {

html {

font-size: 16px;

}

}

/\* Стилі для фокусу елементів (кнопок, полів введення) \*/

.btn:focus, .btn:active:focus, .btn-link.nav-link:focus,

.form-control:focus, .form-check-input:focus {

box-shadow: 0 0 0 0.1rem white, 0 0 0 0.25rem #258cfb;

}

/\* Стилі для кнопок із градієнтом і анімацією \*/

button, .btn {

background: linear-gradient(45deg, #007bff, #00aaff);

border: none;

border-radius: 5px;

color: #fff;

padding: 8px 16px;

font-weight: bold;

text-shadow: 1px 1px 2px rgba(0,0,0,0.3);

transition: transform 0.3s ease, box-shadow 0.3s ease, background 0.3s ease;

}

button:hover, .btn:hover {

transform: scale(1.05);

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

background: linear-gradient(45deg, #0056b3, #007bff);

}

button:active, .btn:active {

transform: scale(0.98);

}

/\* Стилі для полів введення із плавними переходами \*/

input.form-control {

border-radius: 5px;

border: 1px solid #ccc;

transition: border-color 0.3s ease;

}

input.form-control:focus {

border-color: #007bff;

}

input.form-control,

textarea.form-control,

select.form-control {

border: 1px solid #ccc;

border-radius: 5px;

padding: 8px;

transition: transform 0.3s ease, border-color 0.3s ease, box-shadow 0.3s ease;

}

input.form-control:hover,

textarea.form-control:hover,

select.form-control:hover {

transform: scale(1.02);

}

input.form-control:focus,

textarea.form-control:focus,

select.form-control:focus {

border-color: #007bff;

box-shadow: 0 4px 8px rgba(0,0,0,0.2);

transform: scale(1.03);

}

/\* Оновлені стилі для футера: чорний текст \*/

footer.footer,

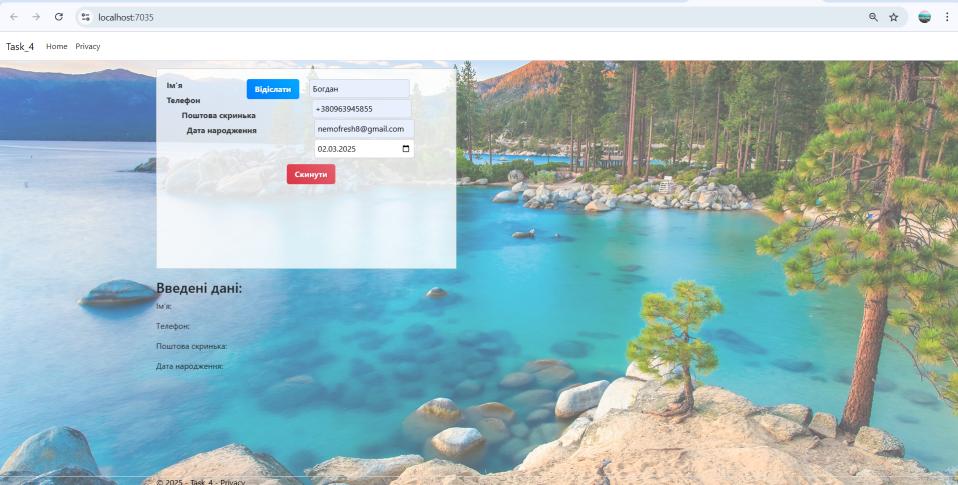
footer.footer a,

footer.footer .container {

color: black !important;

}

## **Результат роботи програми:**



Изображение выглядит как вода, текст, озеро, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как вода, снимок экрана, озеро, текст

Контент, сгенерированный ИИ, может содержать ошибки.

## **Алгоритм побудови проєкту і кодів:**

1. **Створення проєкту:**
   * Запустити ASP.NET Core MVC-проєкт.
2. **Створення моделі:**
   * Створити клас **PersonModel** з властивостями: *Name*, *Phone*, *Email*, *Birthdate*.
3. **Налаштування контролера:**
   * У **HomeController** реалізувати метод **GET**, який повертає порожню модель, та **POST**:
     + Приймає строго типізовану модель як параметр.
     + За умовою *action* (якщо "reset") очищує дані, створюючи нову модель.
4. **Створення виду:**
   * У **Index.cshtml** використовувати директиву @model для прив’язки моделі.
   * Використовувати атрибут asp-for для полів форми, що забезпечує автоматичну прив’язку.
5. **Налаштування загального оформлення:**
   * Використати спільний макет **\_Layout.cshtml** та підключити CSS-стилі для оформлення форми.

## **Функціональність:**

* Форма відображає поля введення для імені, телефону, електронної пошти та дати народження.
* Дані з форми автоматично прив’язуються до моделі **PersonModel** завдяки строгій типізації.
* Після натискання кнопки "Відіслати" дані відображаються на сторінці, а кнопка "Скинути" повертає модель до початкового стану.

## **Висновок:**

Використання строгої типізації з моделлю **PersonModel** спрощує прив’язку даних, покращує читабельність коду та знижує ймовірність помилок, забезпечуючи більш надійну роботу ASP.NET Core MVC-застосунку.

# **Завдання 5**

Створити застосунок, стартове подання (Index) якого містить форму згідно до власного варіанту (Варіант №12, картинка прикріплена). За допомогою стандартного класу Request контролер, викликаний кнопкою Відіслати приймає дані з форми та пе-редає їх назад до подання для їхньої демонстрації під формою, кнопка Скинути має повертати форму у «дефолтний» (початковий) стан, тобто відновити комбінацію вибраних радіо-кнопок.

|  |  |
| --- | --- |
| № варіанту | Вигляд форми |
| 1 | 2 |
| 12 |  |

## **Текст програми:**

**Index.cshtml:**

@{

Layout = "\_Layout"; // Використовуємо власне подання (не стандартний спільний Layout)

}

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Завдання 5 (Варіант №12)</title>

<style>

/\* Контейнер форми — усі абсолютні елементи позиціонуються відносно нього \*/

.form-container {

position: relative;

width: 600px; /\* Ширина форми \*/

height: 400px; /\* Висота форми \*/

background-color: rgba(255, 255, 255, 0.8); /\* Напівпрозорий білий фон \*/

border: 1px solid #ccc;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);

margin-bottom: 20px;

}

/\* --- ЛІВА КОЛОНКА (Місто) --- \*/

/\* Заголовок для колонки міст \*/

.cityTitle {

position: absolute;

left: 20px;

top: 20px;

font-weight: bold;

}

/\* Текстові підписи для міст \*/

.cityLabelKharkiv {

position: absolute;

left: 90px;

top: 40px;

}

.cityLabelKyiv {

position: absolute;

left: 105px;

top: 65px;

}

.cityLabelOdesa {

position: absolute;

left: 60px;

top: 85px;

}

.cityLabelZap {

position: absolute;

left: 20px;

top: 105px;

}

/\* Радіо-кнопки для міст \*/

.cityRadioKharkiv {

position: absolute;

left: 150px;

top: 40px;

}

.cityRadioKyiv {

position: absolute;

left: 150px;

top: 65px;

}

.cityRadioOdesa {

position: absolute;

left: 150px;

top: 85px;

}

.cityRadioZap {

position: absolute;

left: 150px;

top: 105px;

}

/\* Кнопка "Вибрати" для міст \*/

.chooseButton {

position: absolute;

left: 140px;

top: 130px;

}

/\* --- ПРАВА КОЛОНКА (Транспорт) --- \*/

/\* Заголовок для колонки транспорту \*/

.transportTitle {

position: absolute;

left: 300px;

top: 20px;

font-weight: bold;

}

/\* Текстові підписи для транспорту \*/

.transportLabelBus {

position: absolute;

left: 385px;

top: 40px;

}

.transportLabelTrain {

position: absolute;

left: 405px;

top: 65px;

}

.transportLabelPlane {

position: absolute;

left: 350px;

top: 85px;

}

/\* Радіо-кнопки для транспорту \*/

.transportRadioBus {

position: absolute;

left: 460px;

top: 40px;

}

.transportRadioTrain {

position: absolute;

left: 460px;

top: 65px;

}

.transportRadioPlane {

position: absolute;

left: 460px;

top: 85px;

}

/\* Кнопка "Відмовитися" для транспорту \*/

.cancelButton {

position: absolute;

left: 395px;

top: 130px;

}

</style>

</head>

<body>

<!-- Форма з абсолютним позиціонуванням -->

<form method="post" asp-action="Index" asp-controller="Home" class="form-container">

<!-- Ліва колонка (Місто) -->

<div class="cityTitle">Виберіть місто:</div>

<div class="cityLabelKharkiv">Харків</div>

<div class="cityLabelKyiv">Київ</div>

<div class="cityLabelOdesa">Одеса</div>

<div class="cityLabelZap">Запоріжжя</div>

<div class="cityRadioKharkiv">

<input type="radio" name="City" value="Kharkiv" @(ViewBag.City == "Kharkiv" ? "checked" : "") />

</div>

<div class="cityRadioKyiv">

<input type="radio" name="City" value="Kyiv" @(ViewBag.City == "Kyiv" ? "checked" : "") />

</div>

<div class="cityRadioOdesa">

<input type="radio" name="City" value="Odesa" @(ViewBag.City == "Odesa" ? "checked" : "") />

</div>

<div class="cityRadioZap">

<input type="radio" name="City" value="Zaporizhzhia" @(ViewBag.City == "Zaporizhzhia" ? "checked" : "") />

</div>

<div class="chooseButton">

<!-- Кнопка для підтвердження вибору міста -->

<button type="submit" name="action" value="choose" class="btn btn-primary">Вибрати</button>

</div>

<!-- Права колонка (Транспорт) -->

<div class="transportTitle">Виберіть транспорт:</div>

<div class="transportLabelBus">Автобус</div>

<div class="transportLabelTrain">Потяг</div>

<div class="transportLabelPlane">Літак</div>

<div class="transportRadioBus">

<input type="radio" name="Transport" value="Bus" @(ViewBag.Transport == "Bus" ? "checked" : "") />

</div>

<div class="transportRadioTrain">

<input type="radio" name="Transport" value="Train" @(ViewBag.Transport == "Train" ? "checked" : "") />

</div>

<div class="transportRadioPlane">

<input type="radio" name="Transport" value="Plane" @(ViewBag.Transport == "Plane" ? "checked" : "") />

</div>

<div class="cancelButton">

<!-- Кнопка для відміни вибору -->

<button type="submit" name="action" value="cancel" class="btn btn-danger">Відмовитися</button>

</div>

</form>

@\* Обчислення відображуваних назв на основі вибраних кодів \*@

@{

// Отримуємо вибрані значення з ViewBag

string cityCode = ViewBag.City as string ?? "";

string transportCode = ViewBag.Transport as string ?? "";

// Використовуємо switch для перетворення коду в назву міста

string cityName = cityCode switch

{

"Kharkiv" => "Харків",

"Kyiv" => "Київ",

"Odesa" => "Одеса",

"Zaporizhzhia" => "Запоріжжя",

\_ => ""

};

// Аналогічно для транспорту

string transportName = transportCode switch

{

"Bus" => "Автобус",

"Train" => "Потяг",

"Plane" => "Літак",

\_ => ""

};

}

@if (ViewBag.ShowResult == true)

{

<hr />

<div class="result-text">

<h3>Вибрані параметри:</h3>

<p>Місто: @cityName</p>

<p>Транспорт: @transportName</p>

</div>

}

</body>

</html>

**HomeController.cs:**

using Microsoft.AspNetCore.Mvc;

namespace Task\_5.Controllers

{

public class HomeController : Controller

{

// GET: /Home/Index

[HttpGet]

public IActionResult Index()

{

// Встановлюємо дефолтні значення для радіо-кнопок

ViewBag.City = "Kharkiv"; // За замовчуванням – Харків

ViewBag.Transport = "Train"; // За замовчуванням – Потяг

ViewBag.ShowResult = false; // Перший запуск – результат не показується

return View();

}

// POST: /Home/Index

[HttpPost]

public IActionResult Index(string action)

{

if (action == "cancel")

{

// Якщо натиснуто "Відмовитися", повертаємо дефолтний стан

ViewBag.City = "Kharkiv";

ViewBag.Transport = "Train";

ViewBag.ShowResult = false; // Результат не показується

}

else

{

// Якщо натиснуто "Вибрати", зчитуємо вибір користувача з форми

string city = Request.Form["City"];

string transport = Request.Form["Transport"];

// Зберігаємо отримані значення для подальшого відображення

ViewBag.City = city;

ViewBag.Transport = transport;

ViewBag.ShowResult = true; // Показуємо результат

}

return View();

}

}

}

**\_Layout.cshtml:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Task\_5</title>

<!-- Підключення Bootstrap та додаткових стилів -->

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

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

<link rel="stylesheet" href="~/Task\_5.styles.css" asp-append-version="true" />

</head>

<body>

<header>

<!-- Верхнє меню сайту -->

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light border-bottom box-shadow mb-3">

<div class="container-fluid">

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Task\_5</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse"

aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<div class="container">

<main role="main" class="pb-3">

@RenderBody() <!-- Тут буде підставлятися вміст конкретної сторінки -->

</main>

</div>

<footer class="border-top footer text-muted">

<div class="container">

&copy; 2025 - Task\_5 - <a asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</div>

</footer>

<!-- Підключення JavaScript бібліотек -->

<script src="~/lib/jquery/dist/jquery.min.js"></script>

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

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

**site.css:**

html {

font-size: 14px; /\* Базовий розмір шрифту для менших екранів \*/

}

@media (min-width: 768px) {

html {

font-size: 16px; /\* Збільшуємо розмір шрифту для великих екранів \*/

}

}

/\* Фокус, активні стани та інші ефекти для елементів \*/

.btn:focus, .btn:active:focus, .btn-link.nav-link:focus, .form-control:focus, .form-check-input:focus {

box-shadow: 0 0 0 0.1rem white, 0 0 0 0.25rem #258cfb;

}

html {

position: relative;

min-height: 100%;

}

body {

margin-bottom: 60px;

background: url('https://s1.1zoom.me/b5440/715/USA\_Parks\_Mountains\_Lake\_Stones\_Forests\_California\_556636\_1920x1080.jpg') no-repeat center center fixed;

background-size: cover;

}

/\* Колір тексту результату \*/

.result-text {

color: white;

}

/\* Покращення вигляду хедера та футера \*/

.navbar, .footer {

background-color: rgba(255, 255, 255, 0.9);

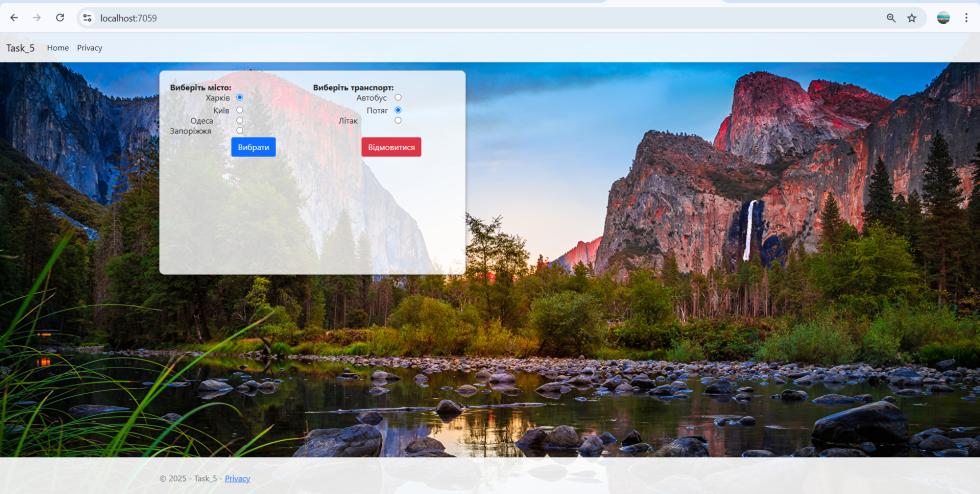
}

.footer {

padding: 10px 0;

}

## **Результат роботи програми:**



Изображение выглядит как текст, снимок экрана, растение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, растение, небо

Контент, сгенерированный ИИ, может содержать ошибки.

## **Алгоритм побудови проєкту і кодів:**

1. **Створити або відкрити ASP.NET Core MVC-проєкт.**
2. **Додати контролер** HomeController з двома методами Index:
   * **GET**: встановлює початкові значення радіокнопок (наприклад, місто = "Kharkiv", транспорт = "Train") та ховає результат.
   * **POST**:
     + Якщо action == "cancel", повертає всі значення до початкового стану.
     + Якщо action == "choose", зчитує з Request.Form значення City та Transport, зберігає їх у ViewBag і вмикає відображення результату.
3. **Створити вигляд** Index.cshtml, де:
   * Є радіокнопки для вибору міста та транспорту.
   * Кнопки «Вибрати» і «Відмовитися» надсилають форму у той самий метод Index (POST).
   * Під формою відображається результат, якщо ViewBag.ShowResult == true.
4. **Підключити і налаштувати \_Layout.cshtml** (спільний макет) та файли стилів для оформлення.

## **Функціональність:**

* Застосунок відображає дві групи радіокнопок: «Виберіть місто» та «Виберіть транспорт».
* При натисканні «Вибрати» вибрані варіанти зберігаються та виводяться на екран.
* При натисканні «Відмовитися» значення повертаються до початкових (дефолтних) налаштувань.

## **Висновок:**

Реалізовано форму з радіокнопками, яка приймає дані через Request.Form, зберігає їх у ViewBag та відображає користувачеві. Цей приклад демонструє базову взаємодію з радіокнопками в ASP.NET Core MVC та роботу з кількома кнопками в одній формі.

# **Завдання 6**

Виконати функцї минулого завдання, використавши для отримання да-них форми замість класу Request колекцію типу IFormCollection.

## **Текст програми:**

**Index.cshtml:**

@{

Layout = "\_Layout"; // Використовуємо власне подання, не базове \_Layout.cshtml

}

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Завдання 6 (Варіант №12)</title>

<style>

/\* Контейнер форми – всі абсолютно позиціоновані елементи розташовуються відносно нього \*/

.form-container {

position: relative;

width: 600px; /\* Задаємо ширину форми \*/

height: 400px; /\* Задаємо висоту форми \*/

background-color: rgba(255, 255, 255, 0.8);

border: 1px solid #ccc;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);

margin-bottom: 20px;

}

/\* --- ЛІВА КОЛОНКА (місто) --- \*/

.cityTitle {

position: absolute;

left: 20px;

top: 20px;

font-weight: bold;

}

.cityLabelKharkiv {

position: absolute;

left: 90px;

top: 40px;

}

.cityLabelKyiv {

position: absolute;

left: 105px;

top: 65px;

}

.cityLabelOdesa {

position: absolute;

left: 60px;

top: 85px;

}

.cityLabelZap {

position: absolute;

left: 20px;

top: 105px;

}

/\* Радіо-кнопки для міст \*/

.cityRadioKharkiv {

position: absolute;

left: 150px;

top: 40px;

}

.cityRadioKyiv {

position: absolute;

left: 150px;

top: 65px;

}

.cityRadioOdesa {

position: absolute;

left: 150px;

top: 85px;

}

.cityRadioZap {

position: absolute;

left: 150px;

top: 105px;

}

/\* Кнопка "Вибрати" для лівої колонки \*/

.chooseButton {

position: absolute;

left: 140px;

top: 130px;

}

/\* --- ПРАВА КОЛОНКА (транспорт) --- \*/

.transportTitle {

position: absolute;

left: 300px;

top: 20px;

font-weight: bold;

}

.transportLabelBus {

position: absolute;

left: 385px;

top: 40px;

}

.transportLabelTrain {

position: absolute;

left: 405px;

top: 65px;

}

.transportLabelPlane {

position: absolute;

left: 350px;

top: 85px;

}

/\* Радіо-кнопки для транспорту \*/

.transportRadioBus {

position: absolute;

left: 460px;

top: 40px;

}

.transportRadioTrain {

position: absolute;

left: 460px;

top: 65px;

}

.transportRadioPlane {

position: absolute;

left: 460px;

top: 85px;

}

/\* Кнопка "Відмовитися" для правої колонки \*/

.cancelButton {

position: absolute;

left: 395px;

top: 130px;

}

</style>

</head>

<body>

<!-- Форма з абсолютним позиціонуванням елементів -->

<form method="post" asp-action="Index" asp-controller="Home" class="form-container">

<!-- Ліва колонка: місто -->

<div class="cityTitle">Виберіть місто:</div>

<div class="cityLabelKharkiv">Харків</div>

<div class="cityRadioKharkiv">

<input type="radio" name="City" value="Kharkiv" @(ViewBag.City == "Kharkiv" ? "checked" : "") />

</div>

<div class="cityLabelKyiv">Київ</div>

<div class="cityRadioKyiv">

<input type="radio" name="City" value="Kyiv" @(ViewBag.City == "Kyiv" ? "checked" : "") />

</div>

<div class="cityLabelOdesa">Одеса</div>

<div class="cityRadioOdesa">

<input type="radio" name="City" value="Odesa" @(ViewBag.City == "Odesa" ? "checked" : "") />

</div>

<div class="cityLabelZap">Запоріжжя</div>

<div class="cityRadioZap">

<input type="radio" name="City" value="Zaporizhzhia" @(ViewBag.City == "Zaporizhzhia" ? "checked" : "") />

</div>

<div class="chooseButton">

<!-- Кнопка підтвердження вибору міста -->

<button type="submit" name="action" value="choose" class="btn btn-primary">Вибрати</button>

</div>

<!-- Права колонка: транспорт -->

<div class="transportTitle">Виберіть транспорт:</div>

<div class="transportLabelBus">Автобус</div>

<div class="transportRadioBus">

<input type="radio" name="Transport" value="Bus" @(ViewBag.Transport == "Bus" ? "checked" : "") />

</div>

<div class="transportLabelTrain">Потяг</div>

<div class="transportRadioTrain">

<input type="radio" name="Transport" value="Train" @(ViewBag.Transport == "Train" ? "checked" : "") />

</div>

<div class="transportLabelPlane">Літак</div>

<div class="transportRadioPlane">

<input type="radio" name="Transport" value="Plane" @(ViewBag.Transport == "Plane" ? "checked" : "") />

</div>

<div class="cancelButton">

<!-- Кнопка скидання вибору -->

<button type="submit" name="action" value="cancel" class="btn btn-danger">Відмовитися</button>

</div>

</form>

@\* Логіка перетворення вибраних кодів на відображувані назви \*@

@{

string cityCode = ViewBag.City as string ?? "";

string transportCode = ViewBag.Transport as string ?? "";

string cityName = cityCode switch

{

"Kharkiv" => "Харків",

"Kyiv" => "Київ",

"Odesa" => "Одеса",

"Zaporizhzhia" => "Запоріжжя",

\_ => ""

};

string transportName = transportCode switch

{

"Bus" => "Автобус",

"Train" => "Потяг",

"Plane" => "Літак",

\_ => ""

};

}

@if (ViewBag.ShowResult == true)

{

<hr />

<div class="result-text">

<h3>Вибрані параметри:</h3>

<p>Місто: @cityName</p>

<p>Транспорт: @transportName</p>

</div>

}

</body>

</html>

**HomeController.cs:**

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Http;

namespace Task\_6.Controllers

{

public class HomeController : Controller

{

[HttpGet]

public IActionResult Index()

{

// Встановлюємо початкові значення для форми при першому завантаженні

ViewBag.City = "Kharkiv";

ViewBag.Transport = "Train";

ViewBag.ShowResult = false;

return View();

}

[HttpPost]

public IActionResult Index(IFormCollection form, string action)

{

if (action == "cancel")

{

// Якщо натиснуто "Відмовитися", повертаємо дефолтні значення

ViewBag.City = "Kharkiv";

ViewBag.Transport = "Train";

ViewBag.ShowResult = false;

}

else

{

// Якщо натиснуто "Вибрати", зчитуємо значення з форми

string city = form["City"];

string transport = form["Transport"];

ViewBag.City = city;

ViewBag.Transport = transport;

ViewBag.ShowResult = true;

}

return View();

}

}

}

**\_Layout.cshtml:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Task\_6</title>

<!-- Підключаємо Bootstrap та додаткові стилі -->

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

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

<link rel="stylesheet" href="~/Task\_6.styles.css" asp-append-version="true" />

</head>

<body>

<header>

<!-- Верхнє меню сайту -->

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light border-bottom box-shadow mb-3">

<div class="container-fluid">

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Task\_6</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse"

aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<div class="container">

<main role="main" class="pb-3">

@RenderBody() <!-- Тут буде відображено контент конкретної сторінки -->

</main>

</div>

<footer class="border-top footer text-muted">

<div class="container">

&copy; 2025 - Task\_6 - <a asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</div>

</footer>

<!-- Підключення необхідних JavaScript бібліотек -->

<script src="~/lib/jquery/dist/jquery.min.js"></script>

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

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

**site.css:**

html {

font-size: 14px; /\* Базовий розмір шрифту для маленьких екранів \*/

}

@media (min-width: 768px) {

html {

font-size: 16px; /\* Збільшуємо розмір шрифту на великих екранах \*/

}

}

/\* Фокус, активні стани та інші ефекти для елементів \*/

.btn:focus, .btn:active:focus, .btn-link.nav-link:focus, .form-control:focus, .form-check-input:focus {

box-shadow: 0 0 0 0.1rem white, 0 0 0 0.25rem #258cfb;

}

html {

position: relative;

min-height: 100%;

}

body {

margin-bottom: 60px;

background: url('https://s1.1zoom.me/b5440/715/USA\_Parks\_Mountains\_Lake\_Stones\_Forests\_California\_556636\_1920x1080.jpg') no-repeat center center fixed;

background-size: cover;

}

/\* Колір тексту для блоку результатів \*/

.result-text {

color: white;

}

/\* Оновлення стилів для навбара та футера \*/

.navbar, .footer {

background-color: rgba(255, 255, 255, 0.9);

}

## **Результат роботи програми:**

Изображение выглядит как текст, снимок экрана, растение, пейзаж

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, растение, небо

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, растение, небо

Контент, сгенерированный ИИ, может содержать ошибки.

## **Алгоритм побудови проєкту і кодів:**

1. **Створити ASP.NET Core MVC-проєкт** або відкрити вже існуючий.
2. **Створити контролер** HomeController з методами:
   * **GET (Index)**: встановлює початкові значення (наприклад, місто = "Kharkiv", транспорт = "Train") і вимикає показ результату (ViewBag.ShowResult = false).
   * **POST (Index)**: приймає параметр типу IFormCollection form:
     + Якщо action == "cancel", повертає все до дефолтних значень.
     + Якщо action == "choose", зчитує з form["City"] і form["Transport"], зберігає у ViewBag та вмикає показ результату.
3. **Створити вигляд** Index.cshtml, де:
   * Є дві групи радіокнопок для вибору міста й транспорту.
   * Кнопки «Вибрати» та «Відмовитися» надсилають форму у метод Index (POST).
   * Після вибору виводиться результат, якщо ViewBag.ShowResult == true.
4. **Налаштувати \_Layout.cshtml** та стилі для загального оформлення.

## **Функціональність:**

* Застосунок відображає форму з радіокнопками для вибору міста й транспорту.
* При натисканні «Вибрати» відображаються обрані значення.
* При натисканні «Відмовитися» форма повертається до початкового стану (заповнені дефолтні радіокнопки, результат не показується).

## **Висновок:**

Таким чином, приклад демонструє базову взаємодію з формою, використовуючи IFormCollection для отримання даних з полів і керування логікою відображення вибраних параметрів.

# **Завдання 7**

Виконати функції першого завдання, використавши для отримання даних форми замість класу Request слабку типізацію форми, додавши до кожної радіо-кнопки форми відповідну змінну.

## **Текст програми:**

**Index.cshtml:**

@{

Layout = "\_Layout"; // Використовуємо власне подання (без стандартного спільного Layout)

}

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Завдання 7 (Варіант №12)</title>

<style>

/\* Контейнер форми – всі абсолютно позиціоновані елементи розташовуються відносно нього \*/

.form-container {

position: relative;

width: 600px; /\* Задаємо ширину форми \*/

height: 400px; /\* Задаємо висоту форми \*/

background-color: rgba(255, 255, 255, 0.8);

border: 1px solid #ccc;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);

margin-bottom: 20px;

}

/\* --- ЛІВА КОЛОНКА (Місто) --- \*/

.cityTitle {

position: absolute;

left: 20px;

top: 20px;

font-weight: bold;

}

.cityLabelKharkiv {

position: absolute;

left: 90px;

top: 40px;

}

.cityLabelKyiv {

position: absolute;

left: 105px;

top: 65px;

}

.cityLabelOdesa {

position: absolute;

left: 60px;

top: 85px;

}

.cityLabelZap {

position: absolute;

left: 20px;

top: 105px;

}

/\* Радіо-кнопки для міст \*/

.cityRadioKharkiv {

position: absolute;

left: 150px;

top: 40px;

}

.cityRadioKyiv {

position: absolute;

left: 150px;

top: 65px;

}

.cityRadioOdesa {

position: absolute;

left: 150px;

top: 85px;

}

.cityRadioZap {

position: absolute;

left: 150px;

top: 105px;

}

/\* Кнопка "Вибрати" для лівої колонки \*/

.chooseButton {

position: absolute;

left: 140px;

top: 130px;

}

/\* --- ПРАВА КОЛОНКА (Транспорт) --- \*/

.transportTitle {

position: absolute;

left: 300px;

top: 20px;

font-weight: bold;

}

.transportLabelBus {

position: absolute;

left: 385px;

top: 40px;

}

.transportLabelTrain {

position: absolute;

left: 405px;

top: 65px;

}

.transportLabelPlane {

position: absolute;

left: 350px;

top: 85px;

}

/\* Радіо-кнопки для транспорту \*/

.transportRadioBus {

position: absolute;

left: 460px;

top: 40px;

}

.transportRadioTrain {

position: absolute;

left: 460px;

top: 65px;

}

.transportRadioPlane {

position: absolute;

left: 460px;

top: 85px;

}

/\* Кнопка "Відмовитися" для правої колонки \*/

.cancelButton {

position: absolute;

left: 395px;

top: 130px;

}

/\* Стиль для блоку результату – білий текст \*/

.result-text {

color: white;

}

</style>

</head>

<body>

<!-- Форма з абсолютним позиціонуванням елементів -->

<form method="post" asp-action="Index" asp-controller="Home" class="form-container">

<!-- Ліва колонка: місто -->

<div class="cityTitle">Виберіть місто:</div>

<div class="cityLabelKharkiv">Харків</div>

<div class="cityRadioKharkiv">

<input type="radio" name="City" value="Kharkiv" @(ViewBag.City == "Kharkiv" ? "checked" : "") />

</div>

<div class="cityLabelKyiv">Київ</div>

<div class="cityRadioKyiv">

<input type="radio" name="City" value="Kyiv" @(ViewBag.City == "Kyiv" ? "checked" : "") />

</div>

<div class="cityLabelOdesa">Одеса</div>

<div class="cityRadioOdesa">

<input type="radio" name="City" value="Odesa" @(ViewBag.City == "Odesa" ? "checked" : "") />

</div>

<div class="cityLabelZap">Запоріжжя</div>

<div class="cityRadioZap">

<input type="radio" name="City" value="Zaporizhzhia" @(ViewBag.City == "Zaporizhzhia" ? "checked" : "") />

</div>

<div class="chooseButton">

<!-- Кнопка підтвердження вибору міста -->

<button type="submit" name="action" value="choose" class="btn btn-primary">Вибрати</button>

</div>

<!-- Права колонка: транспорт -->

<div class="transportTitle">Виберіть транспорт:</div>

<div class="transportLabelBus">Автобус</div>

<div class="transportRadioBus">

<input type="radio" name="Transport" value="Bus" @(ViewBag.Transport == "Bus" ? "checked" : "") />

</div>

<div class="transportLabelTrain">Потяг</div>

<div class="transportRadioTrain">

<input type="radio" name="Transport" value="Train" @(ViewBag.Transport == "Train" ? "checked" : "") />

</div>

<div class="transportLabelPlane">Літак</div>

<div class="transportRadioPlane">

<input type="radio" name="Transport" value="Plane" @(ViewBag.Transport == "Plane" ? "checked" : "") />

</div>

<div class="cancelButton">

<!-- Кнопка для скидання вибору -->

<button type="submit" name="action" value="cancel" class="btn btn-danger">Відмовитися</button>

</div>

</form>

@\* Логіка перетворення вибраних кодів у відображувані назви \*@

@{

string cityCode = ViewBag.City as string ?? "";

string transportCode = ViewBag.Transport as string ?? "";

// В залежності від вибраного коду визначаємо назву міста

string cityName = cityCode switch

{

"Kharkiv" => "Харків",

"Kyiv" => "Київ",

"Odesa" => "Одеса",

"Zaporizhzhia" => "Запоріжжя",

\_ => ""

};

// Аналогічно визначаємо назву транспорту

string transportName = transportCode switch

{

"Bus" => "Автобус",

"Train" => "Потяг",

"Plane" => "Літак",

\_ => ""

};

}

@if (ViewBag.ShowResult == true)

{

<hr />

<div class="result-text">

<h3>Вибрані параметри:</h3>

<p>Місто: @cityName</p>

<p>Транспорт: @transportName</p>

</div>

}

</body>

</html>

**HomeController.cs:**

using Microsoft.AspNetCore.Mvc;

namespace Task\_7.Controllers

{

public class HomeController : Controller

{

[HttpGet]

public IActionResult Index()

{

// При першому завантаженні встановлюємо дефолтні значення

ViewBag.City = "Kharkiv";

ViewBag.Transport = "Train";

ViewBag.ShowResult = false;

return View();

}

// Параметри City, Transport та action отримуються безпосередньо з форми

[HttpPost]

public IActionResult Index(string City, string Transport, string action)

{

if (action == "cancel")

{

// Якщо натиснуто "Відмовитися", повертаємо дефолтний стан

ViewBag.City = "Kharkiv";

ViewBag.Transport = "Train";

ViewBag.ShowResult = false;

}

else

{

// Якщо натиснуто "Вибрати", зберігаємо значення з форми для відображення

ViewBag.City = City;

ViewBag.Transport = Transport;

ViewBag.ShowResult = true;

}

return View();

}

}

}

**\_Layout.cshtml:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Task\_7</title>

<!-- Підключення Bootstrap та додаткових стилів -->

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

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

<link rel="stylesheet" href="~/Task\_7.styles.css" asp-append-version="true" />

</head>

<body>

<header>

<!-- Верхнє меню сайту -->

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light border-bottom box-shadow mb-3">

<div class="container-fluid">

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Task\_7</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse"

aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<div class="container">

<main role="main" class="pb-3">

@RenderBody() <!-- Тут відображається контент конкретної сторінки -->

</main>

</div>

<footer class="border-top footer text-muted">

<div class="container">

&copy; 2025 - Task\_7 - <a asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</div>

</footer>

<!-- Підключення необхідних JavaScript бібліотек -->

<script src="~/lib/jquery/dist/jquery.min.js"></script>

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

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

**site.css:**

html {

font-size: 14px; /\* Базовий розмір шрифту для маленьких екранів \*/

}

@media (min-width: 768px) {

html {

font-size: 16px; /\* Збільшуємо шрифт на великих екранах \*/

}

}

/\* Фокус, активні стани та інші ефекти для елементів \*/

.btn:focus, .btn:active:focus, .btn-link.nav-link:focus, .form-control:focus, .form-check-input:focus {

box-shadow: 0 0 0 0.1rem white, 0 0 0 0.25rem #258cfb;

}

html {

position: relative;

min-height: 100%;

}

body {

margin-bottom: 60px;

background: url('https://s1.1zoom.me/b5440/715/USA\_Parks\_Mountains\_Lake\_Stones\_Forests\_California\_556636\_1920x1080.jpg') no-repeat center center fixed;

background-size: cover;

}

/\* Колір тексту для блоку результату \*/

.result-text {

color: white;

}

/\* Оновлення стилів для навбара та футера \*/

.navbar, .footer {

background-color: rgba(255, 255, 255, 0.9);

}

## **Результат роботи програми:**

Изображение выглядит как текст, растение, снимок экрана, на открытом воздухе

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, растение, небо

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, растение, снимок экрана, на открытом воздухе

Контент, сгенерированный ИИ, может содержать ошибки.

## **Алгоритм побудови проєкту і кодів:**

1. **Створення проєкту:** Запустити ASP.NET Core MVC-проєкт або відкрити існуючий.
2. **Реалізація контролера:**
   * У **HomeController** створити метод **GET Index**, який встановлює дефолтні значення (наприклад, City = "Kharkiv", Transport = "Train") та вимикає показ результату (ShowResult = false).
   * Реалізувати метод **POST Index**, який приймає параметри (слабка типізація) string City, string Transport та string action. Якщо action == "cancel", повертаються дефолтні значення; інакше зберігаються отримані дані та активується показ результату.
3. **Створення вигляду:**
   * У **Index.cshtml** розмістити форму з радіокнопками, де кожна кнопка має відповідну змінну (атрибут name відповідає параметрам методу контролера).
   * Додати кнопки для підтвердження вибору (action = "choose") та для скидання вибору (action = "cancel").
   * Виводити результат вибору (назви міста та транспорту) під формою, якщо ViewBag.ShowResult == true.
4. **Налаштування макету:** Використати власний макет **\_Layout.cshtml** та файли стилів для оформлення сторінки.

## **Функціональність:**

* Користувач обирає місто та транспорт через радіокнопки, які автоматично передають свої значення як параметри в метод контролера.
* Після натискання кнопки «Вибрати» дані з форми зберігаються та відображаються на сторінці; кнопка «Відмовитися» повертає форму до початкового (дефолтного) стану.

## **Висновок:**

Використання слабкої типізації для обробки даних форми спрощує передачу значень з радіокнопок, забезпечуючи зрозуміле та легке управління даними у контролері. Це дозволяє ефективно керувати станом форми, демонструючи базову взаємодію між користувачем і сервером в ASP.NET Core MVC.

# **\*Завдання 8**

Виконати функції першого завдання, використавши для отримання даних форми замість класу Request сувору типізацію форми, додавши до радіо-кнопок форми відповідну модель.

## **Текст програми:**

**Index.cshtml:**

@model Task\_8.Models.SelectionModel

@{

Layout = "\_Layout"; // Використовуємо власне подання (не спільний \_Layout)

}

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Завдання 8 (Варіант №12) – Сувора типізація</title>

<style>

/\* Контейнер форми: всі абсолютно позиціоновані елементи розташовуються відносно нього \*/

.form-container {

position: relative;

width: 600px; /\* Фіксована ширина форми \*/

height: 400px; /\* Фіксована висота форми \*/

background-color: rgba(255, 255, 255, 0.8);

border: 1px solid #ccc;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);

margin-bottom: 20px;

}

/\* --- ЛІВА КОЛОНКА (Місто) --- \*/

.cityTitle {

position: absolute;

left: 20px;

top: 20px;

font-weight: bold;

}

.cityLabelKharkiv {

position: absolute;

left: 90px;

top: 40px;

}

.cityLabelKyiv {

position: absolute;

left: 105px;

top: 65px;

}

.cityLabelOdesa {

position: absolute;

left: 60px;

top: 85px;

}

.cityLabelZap {

position: absolute;

left: 20px;

top: 105px;

}

/\* Радіо-кнопки для міст \*/

.cityRadioKharkiv {

position: absolute;

left: 150px;

top: 40px;

}

.cityRadioKyiv {

position: absolute;

left: 150px;

top: 65px;

}

.cityRadioOdesa {

position: absolute;

left: 150px;

top: 85px;

}

.cityRadioZap {

position: absolute;

left: 150px;

top: 105px;

}

/\* Кнопка "Вибрати" \*/

.chooseButton {

position: absolute;

left: 140px;

top: 130px;

}

/\* --- ПРАВА КОЛОНКА (Транспорт) --- \*/

.transportTitle {

position: absolute;

left: 300px;

top: 20px;

font-weight: bold;

}

.transportLabelBus {

position: absolute;

left: 385px;

top: 40px;

}

.transportLabelTrain {

position: absolute;

left: 405px;

top: 65px;

}

.transportLabelPlane {

position: absolute;

left: 350px;

top: 85px;

}

/\* Радіо-кнопки для транспорту \*/

.transportRadioBus {

position: absolute;

left: 460px;

top: 40px;

}

.transportRadioTrain {

position: absolute;

left: 460px;

top: 65px;

}

.transportRadioPlane {

position: absolute;

left: 460px;

top: 85px;

}

/\* Кнопка "Відмовитися" \*/

.cancelButton {

position: absolute;

left: 395px;

top: 130px;

}

/\* Стиль для блоку з результатом – білий текст \*/

.result-text {

color: white;

}

</style>

</head>

<body>

<!-- Форма з абсолютним позиціонуванням елементів -->

<form method="post" asp-action="Index" asp-controller="Home" class="form-container">

<!-- Ліва колонка (Місто) -->

<div class="cityTitle">Виберіть місто:</div>

<div class="cityLabelKharkiv">Харків</div>

<div class="cityRadioKharkiv">

<input type="radio" asp-for="City" value="Kharkiv" />

</div>

<div class="cityLabelKyiv">Київ</div>

<div class="cityRadioKyiv">

<input type="radio" asp-for="City" value="Kyiv" />

</div>

<div class="cityLabelOdesa">Одеса</div>

<div class="cityRadioOdesa">

<input type="radio" asp-for="City" value="Odesa" />

</div>

<div class="cityLabelZap">Запоріжжя</div>

<div class="cityRadioZap">

<input type="radio" asp-for="City" value="Zaporizhzhia" />

</div>

<div class="chooseButton">

<!-- Кнопка для підтвердження вибору міста -->

<button type="submit" name="action" value="choose" class="btn btn-primary">Вибрати</button>

</div>

<!-- Права колонка (Транспорт) -->

<div class="transportTitle">Виберіть транспорт:</div>

<div class="transportLabelBus">Автобус</div>

<div class="transportRadioBus">

<input type="radio" asp-for="Transport" value="Bus" />

</div>

<div class="transportLabelTrain">Потяг</div>

<div class="transportRadioTrain">

<input type="radio" asp-for="Transport" value="Train" />

</div>

<div class="transportLabelPlane">Літак</div>

<div class="transportRadioPlane">

<input type="radio" asp-for="Transport" value="Plane" />

</div>

<div class="cancelButton">

<!-- Кнопка для скидання вибору -->

<button type="submit" name="action" value="cancel" class="btn btn-danger">Відмовитися</button>

</div>

</form>

@\* Якщо показ результату увімкнено, відображаємо вибрані параметри \*@

@if (ViewBag.ShowResult == true)

{

<hr />

<div class="result-text">

<h3>Вибрані параметри:</h3>

<p>Місто: @Model.City</p>

<p>Транспорт: @Model.Transport</p>

</div>

}

</body>

</html>

**HomeController.cs:**

using Microsoft.AspNetCore.Mvc;

using Task\_8.Models;

namespace Task\_8.Controllers

{

public class HomeController : Controller

{

[HttpGet]

public IActionResult Index()

{

// Повертаємо модель із дефолтними значеннями

var model = new SelectionModel();

ViewBag.ShowResult = false; // Результат спочатку не показується

return View(model);

}

[HttpPost]

public IActionResult Index(SelectionModel model, string action)

{

if (action == "cancel")

{

// Якщо натиснуто "Відмовитися" – очищуємо ModelState і повертаємо нову модель

ModelState.Clear();

model = new SelectionModel();

ViewBag.ShowResult = false;

}

else

{

// Інакше – показуємо результат вибору

ViewBag.ShowResult = true;

}

return View(model);

}

}

}

**\_Layout.cshtml:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Task\_8</title>

<!-- Підключення Bootstrap та інших стилів -->

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

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

<link rel="stylesheet" href="~/Task\_8.styles.css" asp-append-version="true" />

</head>

<body>

<header>

<!-- Верхнє меню сайту -->

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light border-bottom box-shadow mb-3">

<div class="container-fluid">

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Task\_8</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse"

aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<div class="container">

<main role="main" class="pb-3">

@RenderBody() <!-- Контент конкретної сторінки -->

</main>

</div>

<footer class="border-top footer text-muted">

<div class="container">

&copy; 2025 - Task\_8 - <a asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</div>

</footer>

<!-- Підключення JavaScript бібліотек -->

<script src="~/lib/jquery/dist/jquery.min.js"></script>

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

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

**SelectionModel.cs:**

namespace Task\_8.Models

{

public class SelectionModel

{

// За замовчуванням встановлюємо значення для міста та транспорту

public string City { get; set; } = "Kharkiv";

public string Transport { get; set; } = "Train";

}

}

**site.css:**

html {

font-size: 14px; /\* Базовий розмір шрифту для маленьких екранів \*/

}

@media (min-width: 768px) {

html {

font-size: 16px; /\* Збільшуємо шрифт на великих екранах \*/

}

}

.btn:focus, .btn:active:focus, .btn-link.nav-link:focus, .form-control:focus, .form-check-input:focus {

box-shadow: 0 0 0 0.1rem white, 0 0 0 0.25rem #258cfb;

}

html {

position: relative;

min-height: 100%;

}

body {

margin-bottom: 60px;

background: url('https://s1.1zoom.me/b5440/715/USA\_Parks\_Mountains\_Lake\_Stones\_Forests\_California\_556636\_1920x1080.jpg') no-repeat center center fixed;

background-size: cover;

}

.result-text {

color: white; /\* Білий текст для блоку з результатом \*/

}

/\* Стилізація navbar та футера \*/

.navbar, .footer {

background-color: rgba(255, 255, 255, 0.9);

}

## **Результати роботи програми:**

Изображение выглядит как текст, растение, снимок экрана, пейзаж

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, растение, небо

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, растение, снимок экрана, пейзаж

Контент, сгенерированный ИИ, может содержать ошибки.

## **Алгоритм побудови проєкту і кодів:**

1. **Створення моделі:**
   * Створити клас **SelectionModel** з властивостями *City* і *Transport*, встановивши дефолтні значення (наприклад, "Kharkiv" та "Train").
2. **Реалізація контролера:**
   * У **HomeController** реалізувати метод **GET Index**, який повертає нову модель та встановлює ViewBag.ShowResult = false.
   * У методі **POST Index** приймати строго типізовану модель **SelectionModel** разом із параметром action.
     + Якщо action == "cancel", очищати ModelState та повертати нову модель (дефолтні значення).
     + Інакше – залишати отримані значення, встановлювати ViewBag.ShowResult = true для відображення результату.
3. **Створення вигляду:**
   * У **Index.cshtml** використовувати директиву @model Task\_8.Models.SelectionModel та атрибут asp-for для прив’язки радіокнопок до властивостей моделі.
   * Додати кнопки для підтвердження вибору (choose) та для скидання (cancel).
   * Якщо ViewBag.ShowResult == true, виводити вибрані значення з моделі.

## **Функціональність:**

* Застосунок відображає форму з радіокнопками для вибору міста та транспорту, прив’язаними до строго типізованої моделі.
* При відправленні форми дані автоматично зв’язуються з властивостями моделі.
* При натисканні «Вибрати» обрані значення з моделі відображаються на сторінці, а кнопка «Відмовитися» повертає форму до початкових (дефолтних) значень.

## **Висновок:**

Використання суворої типізації через модель **SelectionModel** забезпечує автоматичну прив’язку даних форми, покращує читабельність коду та знижує ризик помилок, що сприяє більш ефективній та зрозумілій розробці ASP.NET Core MVC-застосунку.

# **Завдання 9**

Створити застосунок, стартове подання (Index) якого містить форму згідно до власного варіанту. За допомогою стандартного класу Request контролер, викликаний кнопкою Відіслати приймає дані з форми та пе-редає їх назад до подання для їхньої демонстрації під формою, кнопка Скинути має повертати форму у «дефолтний» (початковий) стан, тобто відновити комбінацію вибраних «чекбоксів».

## **Текст програми:**

**Index.cshtml:**

@{

Layout = "\_Layout"; // Використовуємо загальний Layout

}

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Завдання 9 (Варіант №12)</title>

<style>

/\* Стилі для контейнера форми \*/

.form-container {

position: relative;

width: 600px; /\* Ширина форми \*/

height: 300px; /\* Висота форми \*/

border: 1px solid #ccc;

margin-bottom: 20px;

background-color: rgba(255, 255, 255, 0.9);

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0,0,0,0.1);

padding: 10px;

}

/\* Заголовок форми \*/

.interestsTitle {

position: absolute;

left: 150px;

top: 20px;

font-weight: bold;

font-size: 1.5em;

color: #333;

}

/\* Стилі для "Спорт": текст та чекбокс \*/

.interestSportLabel {

position: absolute;

left: 240px;

top: 60px;

font-size: 1.2em;

color: #444;

}

.interestSportCheck {

position: absolute;

left: 320px;

top: 60px;

}

/\* Стилі для "Мандрівки" \*/

.interestTravelLabel {

position: absolute;

left: 140px;

top: 100px;

font-size: 1.2em;

color: #444;

}

.interestTravelCheck {

position: absolute;

left: 278px;

top: 100px;

}

/\* Стилі для "Майстрування" \*/

.interestCraftLabel {

position: absolute;

left: 83px;

top: 140px;

font-size: 1.2em;

color: #444;

}

.interestCraftCheck {

position: absolute;

left: 278px;

top: 140px;

}

/\* Стилі для "Малювання" \*/

.interestDrawLabel {

position: absolute;

left: 135px;

top: 180px;

font-size: 1.2em;

color: #444;

}

.interestDrawCheck {

position: absolute;

left: 320px;

top: 180px;

}

/\* Стилі для кнопок \*/

.chooseButton, .cancelButton {

position: absolute;

top: 220px;

}

.chooseButton {

left: 170px;

}

.cancelButton {

left: 260px;

}

.chooseButton button, .cancelButton button {

padding: 8px 16px;

font-size: 1em;

border: none;

border-radius: 5px;

cursor: pointer;

background-color: #258cfb;

color: #fff;

transition: background-color 0.3s;

}

.cancelButton button {

padding: 8px 16px;

font-size: 1em;

border: none;

border-radius: 5px;

cursor: pointer;

background-color: red;

color: #fff;

transition: background-color 0.3s;

}

.chooseButton button:hover, .cancelButton button:hover {

background-color: #1a6fb3;

}

/\* Стилізація чекбоксів \*/

input[type="checkbox"] {

width: 20px;

height: 20px;

}

/\* Стилі для відображення результатів \*/

.result-header,

.result-content {

color: white;

}

</style>

</head>

<body>

<!-- Форма для вибору інтересів -->

<form method="post" asp-action="Index" asp-controller="Home" class="form-container">

<div class="interestsTitle">Виберіть ваші інтереси:</div>

<!-- Чекбокс для "Спорт" -->

<div class="interestSportLabel">Спорт</div>

<div class="interestSportCheck">

<input type="checkbox" name="Interests" value="Спорт"

@(((List<string>)ViewBag.SelectedInterests).Contains("Спорт") ? "checked" : "") />

</div>

<!-- Чекбокс для "Мандрівки" -->

<div class="interestTravelLabel">Мандрівки</div>

<div class="interestTravelCheck">

<input type="checkbox" name="Interests" value="Мандрівки"

@(((List<string>)ViewBag.SelectedInterests).Contains("Мандрівки") ? "checked" : "") />

</div>

<!-- Чекбокс для "Майстрування" -->

<div class="interestCraftLabel">Майстрування</div>

<div class="interestCraftCheck">

<input type="checkbox" name="Interests" value="Майстрування"

@(((List<string>)ViewBag.SelectedInterests).Contains("Майстрування") ? "checked" : "") />

</div>

<!-- Чекбокс для "Малювання" -->

<div class="interestDrawLabel">Малювання</div>

<div class="interestDrawCheck">

<input type="checkbox" name="Interests" value="Малювання"

@(((List<string>)ViewBag.SelectedInterests).Contains("Малювання") ? "checked" : "") />

</div>

<!-- Кнопки для вибору або скасування -->

<div class="chooseButton">

<button type="submit" name="action" value="choose">Вибрати</button>

</div>

<div class="cancelButton">

<button type="submit" name="action" value="cancel">Відмовитися</button>

</div>

</form>

@if (ViewBag.ShowResult == true)

{

<hr />

<h3 class="result-header">Обрані інтереси:</h3>

@\* Вивід обраних інтересів у список \*@

var selected = (List<string>)ViewBag.SelectedInterests;

if (selected.Count > 0)

{

<ul class="result-content">

@foreach (var interest in selected)

{

<li>@interest</li>

}

</ul>

}

else

{

<p class="result-content">Нічого не обрано.</p>

}

}

</body>

</html>

**HomeController.cs:**

using Microsoft.AspNetCore.Mvc;

namespace Task\_9.Controllers

{

public class HomeController : Controller

{

// GET: /Home/Index

[HttpGet]

public IActionResult Index()

{

// Задаємо дефолтні інтереси

ViewBag.SelectedInterests = new List<string> { "Спорт", "Майстрування" };

// Спочатку не показуємо результат

ViewBag.ShowResult = false;

return View();

}

// POST: /Home/Index

[HttpPost]

public IActionResult Index(string action)

{

if (action == "cancel")

{

// Якщо користувач натиснув "Відмовитися" - повертаємо дефолтні значення

ViewBag.SelectedInterests = new List<string> { "Спорт", "Майстрування" };

ViewBag.ShowResult = false;

}

else

{

// Якщо натиснуто "Вибрати", зчитуємо вибрані чекбокси

var selected = Request.Form["Interests"]; // Отримуємо дані з форми

List<string> selectedList = selected.ToList(); // Перетворюємо в список

ViewBag.SelectedInterests = selectedList;

ViewBag.ShowResult = true;

}

return View();

}

}

}

**\_Layout.cshtml:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Task\_9</title>

<!-- Підключення Bootstrap -->

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

<!-- Підключення основного стилю -->

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

<!-- Підключення стилів для Task\_9 -->

<link rel="stylesheet" href="~/Task\_9.styles.css" asp-append-version="true" />

</head>

<body class="task9-background">

<header>

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">

<div class="container-fluid">

<!-- Назва сайту -->

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Task\_9</a>

<!-- Кнопка для мобільного меню -->

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse" aria-controls="navbarSupportedContent"

aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<div class="container">

<main role="main" class="pb-3">

@RenderBody() <!-- Тут рендериться вміст конкретної сторінки -->

</main>

</div>

<footer class="border-top footer text-white">

<div class="container">

&copy; 2025 - Task\_10 - <a asp-area="" asp-controller="Home" asp-action="Privacy" class="text-white">Privacy</a>

</div>

</footer>

<!-- Підключення jQuery та Bootstrap JS -->

<script src="~/lib/jquery/dist/jquery.min.js"></script>

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

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

**site.css:**

html {

font-size: 14px;

}

/\* Збільшуємо розмір шрифту на великих екранах \*/

@media (min-width: 768px) {

html {

font-size: 16px;

}

}

/\* Стилізація фокусу для різних елементів \*/

.btn:focus, .btn:active:focus, .btn-link.nav-link:focus, .form-control:focus, .form-check-input:focus {

box-shadow: 0 0 0 0.1rem white, 0 0 0 0.25rem #258cfb;

}

/\* Загальні стилі для html \*/

html {

position: relative;

min-height: 100%;

}

/\* Відступ знизу сторінки \*/

body {

margin-bottom: 60px;

}

/\* Фон для Task\_9 із заданою картинкою \*/

body.task9-background {

background: url('https://images.squarespace-cdn.com/content/v1/57c517e329687fa2d85e0267/1595215578708-KRGZ8CZ9IVMBW14WIPX4/0K6A3286.jpg?format=2500w') no-repeat center center fixed;

background-size: cover;

}

/\* Стилізація футера \*/

footer.footer {

color: white !important;

}

footer.footer a {

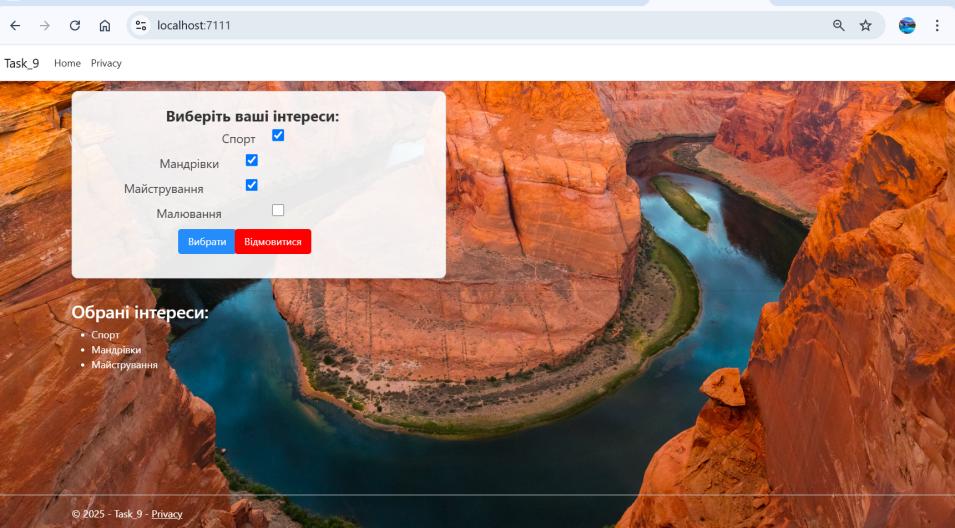
color: white !important;

}

## **Результат роботи програми:**

Изображение выглядит как текст, снимок экрана, вода, природа

Контент, сгенерированный ИИ, может содержать ошибки.



Изображение выглядит как текст, вода, снимок экрана, природа

Контент, сгенерированный ИИ, может содержать ошибки.

## **Алгоритм побудови проєкту і кодів:**

* **Створення проєкту:** Запустити або відкрити ASP.NET Core MVC-проєкт із загальним макетом (\_Layout.cshtml).
* **Реалізація контролера:**
  + **GET метод:** Повертає View із встановленими дефолтними значеннями чекбоксів (наприклад, "Спорт" і "Майстрування") та вимикає відображення результату.
  + **POST метод:** За допомогою класу **Request** зчитує значення чекбоксів (атрибут name="Interests") і перетворює їх на список. Якщо користувач натискає «Відмовитися», повертає дефолтний список і приховує результат, інакше – зберігає обрані значення та відображає їх.
* **Створення вигляду (Index.cshtml):**
  + Розмістити форму з чекбоксами для кожного інтересу.
  + Використовувати умовний вираз для встановлення атрибуту checked, якщо відповідне значення міститься у ViewBag.
  + Додати кнопки «Вибрати» (для підтвердження вибору) та «Відмовитися» (для скидання до дефолтних значень).
  + При наявності значення ViewBag.ShowResult відобразити список обраних інтересів під формою.

## **Функціональність:**

* **Введення:** Користувач може обрати інтереси (наприклад, "Спорт", "Мандрівки", "Майстрування", "Малювання") за допомогою чекбоксів.
* **Обробка:** При натисканні «Вибрати» дані з форми зчитуються через клас **Request** і перетворюються на список, який зберігається в ViewBag.
* **Відображення:** Обрані інтереси відображаються під формою. При натисканні «Відмовитися» форма повертається до початкового (дефолтного) стану.

## **Висновок:**

Застосунок демонструє базову взаємодію з формою в ASP.NET Core MVC, де використовується клас **Request** для зчитування даних з чекбоксів. Реалізація дозволяє легко скинути вибір до початкових значень, що сприяє зручності використання застосунку та ефективній обробці даних.

# **Завдання 10**

Виконати функції минулого завдання, використавши для отримання даних форми замість класу Request колекцію типу IFormCollection.

## **Текст програми:**

**Index.cshtml:**

@{

Layout = "\_Layout"; // Використовуємо спільний Layout

}

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Завдання 10 (Варіант №12)</title>

<style>

/\* Стилі для контейнера форми \*/

.form-container {

position: relative;

width: 600px; /\* Ширина форми \*/

height: 300px; /\* Висота форми \*/

border: 1px solid #ccc;

margin-bottom: 20px;

background-color: rgba(255, 255, 255, 0.9);

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0,0,0,0.1);

padding: 10px;

}

/\* Заголовок форми \*/

.interestsTitle {

position: absolute;

left: 150px;

top: 20px;

font-weight: bold;

font-size: 1.5em;

color: #333;

}

/\* Стилі для напису та чекбоксу "Спорт" \*/

.interestSportLabel {

position: absolute;

left: 240px;

top: 60px;

font-size: 1.2em;

color: #444;

}

.interestSportCheck {

position: absolute;

left: 320px;

top: 60px;

}

/\* Стилі для "Мандрівки" \*/

.interestTravelLabel {

position: absolute;

left: 140px;

top: 100px;

font-size: 1.2em;

color: #444;

}

.interestTravelCheck {

position: absolute;

left: 278px;

top: 100px;

}

/\* Стилі для "Майстрування" \*/

.interestCraftLabel {

position: absolute;

left: 83px;

top: 140px;

font-size: 1.2em;

color: #444;

}

.interestCraftCheck {

position: absolute;

left: 278px;

top: 140px;

}

/\* Стилі для "Малювання" \*/

.interestDrawLabel {

position: absolute;

left: 135px;

top: 180px;

font-size: 1.2em;

color: #444;

}

.interestDrawCheck {

position: absolute;

left: 320px;

top: 180px;

}

/\* Стилі для кнопок \*/

.chooseButton, .cancelButton {

position: absolute;

top: 220px;

}

.chooseButton {

left: 170px;

}

.cancelButton {

left: 260px;

}

.chooseButton button, .cancelButton button {

padding: 8px 16px;

font-size: 1em;

border: none;

border-radius: 5px;

cursor: pointer;

background-color: #258cfb;

color: #fff;

transition: background-color 0.3s;

}

.cancelButton button {

padding: 8px 16px;

font-size: 1em;

border: none;

border-radius: 5px;

cursor: pointer;

background-color: red;

color: #fff;

transition: background-color 0.3s;

}

.chooseButton button:hover, .cancelButton button:hover {

background-color: #1a6fb3;

}

/\* Стилізація чекбоксів \*/

input[type="checkbox"] {

width: 20px;

height: 20px;

}

/\* Стилі для виведення результатів \*/

.result-header,

.result-content {

color: white;

}

</style>

</head>

<body>

<!-- Форма з абсолютним позиціонуванням елементів -->

<form method="post" asp-action="Index" asp-controller="Home" class="form-container">

<div class="interestsTitle">Виберіть ваші інтереси:</div>

<!-- Чекбокс для "Спорт" -->

<div class="interestSportLabel">Спорт</div>

<div class="interestSportCheck">

<input type="checkbox" name="Interests" value="Спорт"

@(((List<string>)ViewBag.SelectedInterests).Contains("Спорт") ? "checked" : "") />

</div>

<!-- Чекбокс для "Мандрівки" -->

<div class="interestTravelLabel">Мандрівки</div>

<div class="interestTravelCheck">

<input type="checkbox" name="Interests" value="Мандрівки"

@(((List<string>)ViewBag.SelectedInterests).Contains("Мандрівки") ? "checked" : "") />

</div>

<!-- Чекбокс для "Майстрування" -->

<div class="interestCraftLabel">Майстрування</div>

<div class="interestCraftCheck">

<input type="checkbox" name="Interests" value="Майстрування"

@(((List<string>)ViewBag.SelectedInterests).Contains("Майстрування") ? "checked" : "") />

</div>

<!-- Чекбокс для "Малювання" -->

<div class="interestDrawLabel">Малювання</div>

<div class="interestDrawCheck">

<input type="checkbox" name="Interests" value="Малювання"

@(((List<string>)ViewBag.SelectedInterests).Contains("Малювання") ? "checked" : "") />

</div>

<!-- Кнопки для дій -->

<div class="chooseButton">

<button type="submit" name="action" value="choose">Вибрати</button>

</div>

<div class="cancelButton">

<button type="submit" name="action" value="cancel">Відмовитися</button>

</div>

</form>

@if (ViewBag.ShowResult == true)

{

<hr />

<h3 class="result-header">Обрані інтереси:</h3>

// Отримуємо список вибраних інтересів із ViewBag

var selected = (List<string>)ViewBag.SelectedInterests;

if (selected.Count > 0)

{

<ul class="result-content">

@foreach (var interest in selected)

{

<li>@interest</li>

}

</ul>

}

else

{

<p class="result-content">Нічого не обрано.</p>

}

}

</body>

</html>

**HomeController.cs:**

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Http; // Для роботи з IFormCollection

using System.Collections.Generic;

using System.Linq;

namespace Task\_10.Controllers

{

public class HomeController : Controller

{

// GET: /Home/Index

[HttpGet]

public IActionResult Index()

{

// Задаємо початкові значення: "Спорт" та "Майстрування"

ViewBag.SelectedInterests = new List<string> { "Спорт", "Майстрування" };

// Спочатку результат не показується

ViewBag.ShowResult = false;

return View();

}

// POST: /Home/Index

[HttpPost]

public IActionResult Index(IFormCollection form, string action)

{

if (action == "cancel")

{

// Якщо користувач натиснув "Відмовитися", повертаємо дефолтні значення

ViewBag.SelectedInterests = new List<string> { "Спорт", "Майстрування" };

ViewBag.ShowResult = false;

}

else

{

// Якщо натиснуто "Вибрати", читаємо значення чекбоксів із форми

var selected = form["Interests"]; // Отримуємо вибрані значення

List<string> selectedList = selected.ToList(); // Перетворюємо їх у список

ViewBag.SelectedInterests = selectedList;

ViewBag.ShowResult = true;

}

return View();

}

}

}

**\_Layout.cshtml:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Task\_10</title>

<!-- Підключення Bootstrap CSS -->

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

<!-- Основні стилі сайту -->

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

<!-- Додаткові стилі для Task\_10 -->

<link rel="stylesheet" href="~/Task\_10.styles.css" asp-append-version="true" />

</head>

<body class="task10-background">

<header>

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">

<div class="container-fluid">

<!-- Назва сайту -->

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Task\_10</a>

<!-- Кнопка для відкриття мобільного меню -->

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse" aria-controls="navbarSupportedContent"

aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<div class="container">

<main role="main" class="pb-3">

@RenderBody() <!-- Тут буде вставлено вміст конкретної сторінки -->

</main>

</div>

<footer class="border-top footer text-white">

<div class="container">

&copy; 2025 - Task\_10 - <a asp-area="" asp-controller="Home" asp-action="Privacy" class="text-white">Privacy</a>

</div>

</footer>

<!-- Підключення jQuery та Bootstrap JS -->

<script src="~/lib/jquery/dist/jquery.min.js"></script>

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

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

**site.css:**

html {

font-size: 14px; /\* Базовий розмір шрифту \*/

}

@media (min-width: 768px) {

html {

font-size: 16px; /\* Збільшуємо шрифт на більших екранах \*/

}

}

/\* Стилізація для елементів при фокусі \*/

.btn:focus, .btn:active:focus, .btn-link.nav-link:focus, .form-control:focus, .form-check-input:focus {

box-shadow: 0 0 0 0.1rem white, 0 0 0 0.25rem #258cfb;

}

html {

position: relative;

min-height: 100%;

}

body {

margin-bottom: 60px; /\* Задаємо нижній відступ \*/

}

/\* Фон для Task\_10 із заданою картинкою \*/

body.task10-background {

background: url('https://images.squarespace-cdn.com/content/v1/57c517e329687fa2d85e0267/1595215578708-KRGZ8CZ9IVMBW14WIPX4/0K6A3286.jpg?format=2500w') no-repeat center center fixed;

background-size: cover;

}

footer.footer {

color: white !important;

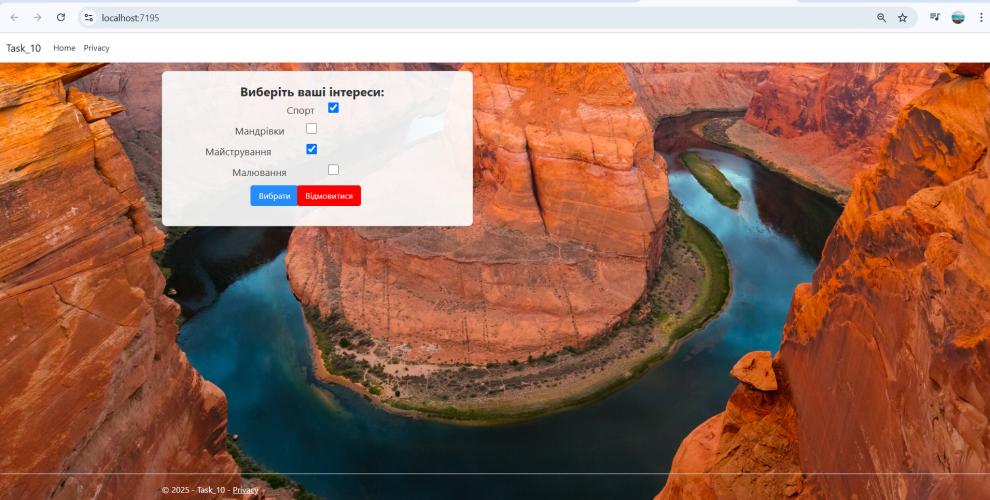
}

footer.footer a {

color: white !important;

}

## **Результат роботи програми:**



Изображение выглядит как текст, снимок экрана, вода, природа

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как природа, снимок экрана, вода, геология

Контент, сгенерированный ИИ, может содержать ошибки.

## **Алгоритм побудови проєкту і кодів:**

1. **Створення проєкту:**
   * Запустити ASP.NET Core MVC-проєкт із загальним макетом (\_Layout.cshtml).
2. **Реалізація контролера (HomeController):**
   * **GET метод:** Повертає View із дефолтним списком інтересів (наприклад, "Спорт" і "Майстрування") і вимикає показ результату (ViewBag.ShowResult = false).
   * **POST метод:** Приймає параметр типу **IFormCollection form** і рядковий параметр **action**.
     + Якщо **action == "cancel"**, повертає дефолтні значення.
     + Якщо **action == "choose"**, зчитує вибрані чекбокси з **form["Interests"]**, перетворює їх на список та встановлює **ViewBag.ShowResult = true**.
3. **Створення вигляду (Index.cshtml):**
   * Розмістити форму з чекбоксами, де для кожного інтересу перевіряється, чи міститься він у ViewBag.SelectedInterests (для встановлення атрибуту *checked*).
   * Додати кнопки «Вибрати» та «Відмовитися», які відправляють форму на відповідний метод контролера.
   * При наявності **ViewBag.ShowResult** відобразити обрані інтереси під формою.

## **Функціональність:**

* **Введення:** Користувач обирає інтереси через чекбокси.
* **Обробка:** При натисканні «Вибрати» дані з форми зчитуються через колекцію **IFormCollection** і перетворюються у список, який зберігається у ViewBag.
* **Відображення:** Обрані інтереси відображаються під формою, а кнопка «Відмовитися» скидає вибір до початкових значень.

## **Висновок:**

Використання колекції **IFormCollection** для зчитування даних форми забезпечує альтернативний спосіб обробки чекбоксів у ASP.NET Core MVC. Це дозволяє ефективно керувати даними форми та повернути її до початкового стану, що сприяє зручності використання застосунку.

# **Завдання 11**

Виконати функції першого завдання, використавши для отримання да-них форми замість класу Request слабку типізацію форми, додавши до кожного «чекбоксу» форми відповідну змінну.

## **Текст програми:**

**Index.cshtml:**

@{

Layout = "\_Layout"; // Використовуємо спільний Layout

}

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Завдання 11 (Варіант №12)</title>

<style>

/\* Контейнер форми \*/

.form-container {

position: relative;

width: 600px; /\* Ширина форми \*/

height: 300px; /\* Висота форми \*/

border: 1px solid #ccc;

margin-bottom: 20px;

background-color: rgba(255, 255, 255, 0.9);

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0,0,0,0.1);

padding: 10px;

}

/\* Заголовок форми \*/

.interestsTitle {

position: absolute;

left: 150px;

top: 20px;

font-weight: bold;

font-size: 1.5em;

color: #333;

}

/\* Стилізація міток для кожного інтересу \*/

.sportLabel,

.travelLabel,

.craftLabel,

.drawLabel {

position: absolute;

font-size: 1.2em;

color: #444;

}

.sportLabel {

left: 240px;

top: 60px;

}

.travelLabel {

left: 140px;

top: 100px;

}

.craftLabel {

left: 83px;

top: 140px;

}

.drawLabel {

left: 135px;

top: 180px;

}

/\* Розташування чекбоксів \*/

.sportCheck {

position: absolute;

left: 320px;

top: 60px;

}

.travelCheck {

position: absolute;

left: 278px;

top: 100px;

}

.craftCheck {

position: absolute;

left: 278px;

top: 140px;

}

.drawCheck {

position: absolute;

left: 320px;

top: 180px;

}

/\* Стилізація кнопок \*/

.chooseButton,

.cancelButton {

position: absolute;

top: 220px;

}

.chooseButton {

left: 170px;

}

.cancelButton {

left: 260px;

}

.chooseButton button,

.cancelButton button {

padding: 8px 16px;

font-size: 1em;

border: none;

border-radius: 5px;

cursor: pointer;

background-color: #258cfb;

color: #fff;

transition: background-color 0.3s;

}

.cancelButton button {

padding: 8px 16px;

font-size: 1em;

border: none;

border-radius: 5px;

cursor: pointer;

background-color: red;

color: #fff;

transition: background-color 0.3s;

}

.chooseButton button:hover,

.cancelButton button:hover {

background-color: #1a6fb3;

}

/\* Стилізація чекбоксів \*/

input[type="checkbox"] {

width: 20px;

height: 20px;

}

/\* Стилізація результатів \*/

.result-header,

.result-content {

color: white;

}

</style>

</head>

<body>

<!-- Форма для вибору інтересів -->

<form method="post" asp-action="Index" asp-controller="Home" class="form-container">

<div class="interestsTitle">Виберіть ваші інтереси:</div>

<!-- Чекбокси для кожного інтересу -->

<div class="sportLabel">Спорт</div>

<div class="sportCheck">

<input type="checkbox" name="Sport" value="true"

@((bool)ViewBag.Sport ? "checked" : "") />

</div>

<div class="travelLabel">Мандрівки</div>

<div class="travelCheck">

<input type="checkbox" name="Travel" value="true"

@((bool)ViewBag.Travel ? "checked" : "") />

</div>

<div class="craftLabel">Майстрування</div>

<div class="craftCheck">

<input type="checkbox" name="Craft" value="true"

@((bool)ViewBag.Craft ? "checked" : "") />

</div>

<div class="drawLabel">Малювання</div>

<div class="drawCheck">

<input type="checkbox" name="Draw" value="true"

@((bool)ViewBag.Draw ? "checked" : "") />

</div>

<!-- Кнопки для дій -->

<div class="chooseButton">

<button type="submit" name="action" value="choose">Вибрати</button>

</div>

<div class="cancelButton">

<button type="submit" name="action" value="cancel">Відмовитися</button>

</div>

</form>

@if (ViewBag.ShowResult == true)

{

<hr />

<h3 class="result-header">Обрані інтереси:</h3>

<ul class="result-content">

@\* Якщо чекбокс активний, показуємо відповідний пункт \*@

@if ((bool)ViewBag.Sport)

{

<li>Спорт</li>

}

@if ((bool)ViewBag.Travel)

{

<li>Мандрівки</li>

}

@if ((bool)ViewBag.Craft)

{

<li>Майстрування</li>

}

@if ((bool)ViewBag.Draw)

{

<li>Малювання</li>

}

</ul>

@\* Якщо нічого не вибрано, виводимо повідомлення \*@

@if (!(bool)ViewBag.Sport && !(bool)ViewBag.Travel && !(bool)ViewBag.Craft && !(bool)ViewBag.Draw)

{

<p class="result-content">Нічого не обрано.</p>

}

}

</body>

</html>

**HomeController.cs:**

using Microsoft.AspNetCore.Mvc;

namespace Task\_11.Controllers

{

public class HomeController : Controller

{

[HttpGet]

public IActionResult Index()

{

// Встановлюємо початкові значення: Спорт та Майстрування увімкнені

ViewBag.Sport = true;

ViewBag.Travel = false;

ViewBag.Craft = true;

ViewBag.Draw = false;

ViewBag.ShowResult = false; // На старті не показуємо результат

return View();

}

[HttpPost]

public IActionResult Index(bool Sport, bool Travel, bool Craft, bool Draw, string action)

{

if (action == "cancel")

{

// Якщо користувач натиснув "Відмовитися", повертаємо початкові налаштування

ViewBag.Sport = true;

ViewBag.Travel = false;

ViewBag.Craft = true;

ViewBag.Draw = false;

ViewBag.ShowResult = false;

}

else

{

// Якщо натиснуто "Вибрати", приймаємо дані з форми

ViewBag.Sport = Sport;

ViewBag.Travel = Travel;

ViewBag.Craft = Craft;

ViewBag.Draw = Draw;

ViewBag.ShowResult = true;

}

return View();

}

}

}

**\_Layout.cshtml:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Task\_11</title>

<!-- Підключення Bootstrap -->

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

<!-- Основні стилі сайту -->

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

<!-- Додаткові стилі для Task\_11 -->

<link rel="stylesheet" href="~/Task\_11.styles.css" asp-append-version="true" />

</head>

<body class="task11-background">

<header>

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">

<div class="container-fluid">

<!-- Назва сайту -->

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Task\_11</a>

<!-- Кнопка для мобільного меню -->

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse"

aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<div class="container">

<main role="main" class="pb-3">

@RenderBody() <!-- Тут буде вставлено вміст конкретної сторінки -->

</main>

</div>

<footer class="border-top footer text-white">

<div class="container">

&copy; 2025 - Task\_11 - <a asp-area="" asp-controller="Home" asp-action="Privacy" class="text-white">Privacy</a>

</div>

</footer>

<!-- Підключення jQuery та Bootstrap JS -->

<script src="~/lib/jquery/dist/jquery.min.js"></script>

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

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

**site.css:**

html {

font-size: 14px; /\* Базовий розмір шрифту \*/

}

@media (min-width: 768px) {

html {

font-size: 16px; /\* Збільшуємо шрифт на більших екранах \*/

}

}

/\* Стилізація фокусу для кнопок та інших елементів \*/

.btn:focus, .btn:active:focus, .btn-link.nav-link:focus, .form-control:focus, .form-check-input:focus {

box-shadow: 0 0 0 0.1rem white, 0 0 0 0.25rem #258cfb;

}

html {

position: relative;

min-height: 100%;

}

body {

margin-bottom: 60px; /\* Відступ знизу сторінки \*/

}

/\* Фон для Task\_11 із зображенням гори \*/

body.task11-background {

background: url('https://images.squarespace-cdn.com/content/v1/57c517e329687fa2d85e0267/1595215578708-KRGZ8CZ9IVMBW14WIPX4/0K6A3286.jpg?format=2500w') no-repeat center center fixed;

background-size: cover;

}

footer.footer {

color: white !important;

}

footer.footer a {

color: white !important;

}

## **Результат роботи програми:**

Изображение выглядит как снимок экрана, природа, вода, текст

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как вода, текст, снимок экрана, природа

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как природа, вода, снимок экрана, текст

Контент, сгенерированный ИИ, может содержать ошибки.

## **Алгоритм побудови проєкту і кодів:**

1. **Створення проєкту:**
   * Запустити ASP.NET Core MVC-проєкт із спільним Layout (\_Layout.cshtml).
2. **Реалізація контролера (HomeController):**
   * **GET Index:** Повертає View із встановленими дефолтними значеннями для чекбоксів (наприклад, Sport = true, Travel = false, Craft = true, Draw = false) і встановлює прапорець ShowResult = false.
   * **POST Index:** Приймає дані форми через параметри (bool Sport, bool Travel, bool Craft, bool Draw, string action).
     + Якщо action = "cancel", повертаються дефолтні значення та ShowResult = false.
     + Інакше – дані з форми зберігаються у ViewBag і встановлюється ShowResult = true для відображення вибраних інтересів.
3. **Створення вигляду (Index.cshtml):**
   * Розмістити форму з чекбоксами, кожен з яких має ім’я, що відповідає параметрам контролера (Sport, Travel, Craft, Draw).
   * Використовувати умовні вирази для встановлення атрибуту **checked** за допомогою ViewBag.
   * Додати кнопки для дій: «Вибрати» (action="choose") та «Відмовитися» (action="cancel").
   * При ShowResult = true відобразити список обраних інтересів.

## **Функціональність:**

* **Введення:** Користувач обирає інтереси через чекбокси (Спорт, Мандрівки, Майстрування, Малювання).
* **Обробка:** Дані форми передаються до контролера через слабку типізацію (bool-параметри), де визначається, чи зберегти вибір або повернути дефолтні значення.
* **Відображення:** При успішному виборі (ShowResult = true) відображається список обраних інтересів; при скиданні – форма повертається до початкового стану.

## **Висновок:**

Використання слабкої типізації для обробки чекбоксів дозволяє легко керувати станом форми, забезпечуючи зрозумілу логіку скидання та відображення даних. Це сприяє підвищенню зручності використання застосунку та покращенню якості коду.

# **\*Завдання 12**

Виконати функції першого завдання, використавши для отримання даних форми замість класу Request сувору типізацію форми, додавши до «чеккбоксів» форми відповідну модель.

## **Текст програми:**

**Index.cshtml:**

@model Task\_12.Models.InterestsModel

@{

Layout = "\_Layout"; // Встановлюємо власний Layout, не використовуючи спільний

}

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Завдання 12 (Варіант №12) – Сувора типізація чекбоксів</title>

<style>

/\* Контейнер форми – основний блок для всіх елементів \*/

.form-container {

position: relative;

width: 600px; /\* Фіксована ширина \*/

height: 300px; /\* Фіксована висота \*/

border: 1px solid #ccc;

margin-bottom: 20px;

background-color: rgba(255, 255, 255, 0.9);

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0,0,0,0.1);

padding: 10px;

}

/\* Заголовок форми \*/

.interestsTitle {

position: absolute;

left: 150px;

top: 20px;

font-weight: bold;

font-size: 1.5em;

color: #333;

}

/\* Мітки для чекбоксів \*/

.sportLabel,

.travelLabel,

.craftLabel,

.drawLabel {

position: absolute;

font-size: 1.2em;

color: #444;

}

.sportLabel {

left: 240px;

top: 60px;

}

.travelLabel {

left: 140px;

top: 100px;

}

.craftLabel {

left: 83px;

top: 140px;

}

.drawLabel {

left: 135px;

top: 180px;

}

/\* Розташування чекбоксів \*/

.sportCheck {

position: absolute;

left: 320px;

top: 60px;

}

.travelCheck {

position: absolute;

left: 278px;

top: 100px;

}

.craftCheck {

position: absolute;

left: 278px;

top: 140px;

}

.drawCheck {

position: absolute;

left: 320px;

top: 180px;

}

/\* Стилізація кнопок \*/

.chooseButton,

.cancelButton {

position: absolute;

top: 220px;

}

.chooseButton {

left: 170px;

}

.cancelButton {

left: 260px;

}

.chooseButton button,

.cancelButton button {

padding: 8px 16px;

font-size: 1em;

border: none;

border-radius: 5px;

cursor: pointer;

background-color: #258cfb;

color: #fff;

transition: background-color 0.3s;

}

.cancelButton button {

padding: 8px 16px;

font-size: 1em;

border: none;

border-radius: 5px;

cursor: pointer;

background-color: red;

color: #fff;

transition: background-color 0.3s;

}

.chooseButton button:hover,

.cancelButton button:hover {

background-color: #1a6fb3;

}

/\* Стилізація чекбоксів \*/

input[type="checkbox"] {

width: 20px;

height: 20px;

}

/\* Стилі для результатів \*/

.result-header,

.result-content {

color: white;

}

</style>

</head>

<body>

<!-- Форма для вибору інтересів з суворою типізацією -->

<form method="post" asp-action="Index" asp-controller="Home" class="form-container">

<div class="interestsTitle">Виберіть ваші інтереси:</div>

<!-- Чекбокс "Спорт" – прив'язаний до властивості Sport у моделі -->

<div class="sportLabel">Спорт</div>

<div class="sportCheck">

<input type="checkbox" asp-for="Sport" value="true" />

</div>

<!-- Чекбокс "Мандрівки" – прив'язаний до властивості Travel -->

<div class="travelLabel">Мандрівки</div>

<div class="travelCheck">

<input type="checkbox" asp-for="Travel" value="true" />

</div>

<!-- Чекбокс "Майстрування" – прив'язаний до властивості Craft -->

<div class="craftLabel">Майстрування</div>

<div class="craftCheck">

<input type="checkbox" asp-for="Craft" value="true" />

</div>

<!-- Чекбокс "Малювання" – прив'язаний до властивості Draw -->

<div class="drawLabel">Малювання</div>

<div class="drawCheck">

<input type="checkbox" asp-for="Draw" value="true" />

</div>

<!-- Кнопки для підтвердження або скидання вибору -->

<div class="chooseButton">

<button type="submit" name="action" value="choose">Вибрати</button>

</div>

<div class="cancelButton">

<button type="submit" name="action" value="cancel">Відмовитися</button>

</div>

</form>

@if (ViewBag.ShowResult == true)

{

<hr />

<h3 class="result-header">Обрані інтереси:</h3>

<ul class="result-content">

@\* Якщо відповідна властивість моделі true – виводимо пункт \*@

@if (Model.Sport)

{

<li>Спорт</li>

}

@if (Model.Travel)

{

<li>Мандрівки</li>

}

@if (Model.Craft)

{

<li>Майстрування</li>

}

@if (Model.Draw)

{

<li>Малювання</li>

}

</ul>

@\* Якщо нічого не вибрано, показуємо повідомлення \*@

@if (!Model.Sport && !Model.Travel && !Model.Craft && !Model.Draw)

{

<p class="result-content">Нічого не обрано.</p>

}

}

</body>

</html>

**HomeController.cs:**

using Microsoft.AspNetCore.Mvc;

using Task\_12.Models;

namespace Task\_12.Controllers

{

public class HomeController : Controller

{

[HttpGet]

public IActionResult Index()

{

// Створюємо нову модель з дефолтними значеннями (Спорт та Майстрування увімкнені)

var model = new InterestsModel();

ViewBag.ShowResult = false; // На початку результат не показується

return View(model);

}

[HttpPost]

public IActionResult Index(InterestsModel model, string action)

{

if (action == "cancel")

{

// Якщо користувач натиснув "Відмовитися" – повертаємо модель з початковими значеннями

model = new InterestsModel();

ViewBag.ShowResult = false;

ModelState.Clear(); // Очищуємо ModelState, щоб скинути попередні дані

}

else

{

// Якщо вибір підтверджено, показуємо результати з моделі

ViewBag.ShowResult = true;

}

return View(model);

}

}

}

**\_Layout.cshtml:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Task\_12</title>

<!-- Підключення Bootstrap для базових стилів -->

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

<!-- Основні стилі сайту -->

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

<!-- Додаткові стилі для Task\_12 -->

<link rel="stylesheet" href="~/Task\_12.styles.css" asp-append-version="true" />

</head>

<body class="task12-background">

<header>

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">

<div class="container-fluid">

<!-- Назва сайту, посилання на Home -->

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Task\_12</a>

<!-- Кнопка для мобільного меню -->

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse"

aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<div class="container">

<main role="main" class="pb-3">

@RenderBody() <!-- Тут буде розміщено вміст конкретної сторінки -->

</main>

</div>

<footer class="border-top footer text-white">

<div class="container">

&copy; 2025 - Task\_12 - <a asp-area="" asp-controller="Home" asp-action="Privacy" class="text-white">Privacy</a>

</div>

</footer>

<!-- Підключення jQuery та Bootstrap JS -->

<script src="~/lib/jquery/dist/jquery.min.js"></script>

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

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

**InterestsModel.cs:**

namespace Task\_12.Models

{

public class InterestsModel

{

// За замовчуванням увімкнено "Спорт" та "Майстрування"

public bool Sport { get; set; } = true;

public bool Travel { get; set; } = false;

public bool Craft { get; set; } = true;

public bool Draw { get; set; } = false;

}

}

**site.css:**

html {

font-size: 14px; /\* Базовий розмір шрифту \*/

}

@media (min-width: 768px) {

html {

font-size: 16px; /\* Збільшуємо шрифт на великих екранах \*/

}

}

/\* Стилізація фокусу для елементів форми \*/

.btn:focus, .btn:active:focus, .btn-link.nav-link:focus, .form-control:focus, .form-check-input:focus {

box-shadow: 0 0 0 0.1rem white, 0 0 0 0.25rem #258cfb;

}

html {

position: relative;

min-height: 100%;

}

body {

margin-bottom: 60px; /\* Відступ знизу сторінки \*/

}

/\* Фон для Task\_12 із зображенням гори \*/

body.task12-background {

background: url('https://images.squarespace-cdn.com/content/v1/57c517e329687fa2d85e0267/1595215578708-KRGZ8CZ9IVMBW14WIPX4/0K6A3286.jpg?format=2500w') no-repeat center center fixed;

background-size: cover;

}

footer.footer {

color: white !important;

}

footer.footer a {

color: white !important;

}

## **Результат роботи програми:**

Изображение выглядит как снимок экрана, природа, вода, текст

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как снимок экрана, вода, текст, природа

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, природа, вода

Контент, сгенерированный ИИ, может содержать ошибки.

## **Алгоритм побудови проєкту і кодів:**

1. **Створення моделі:**
   * Розробити клас **InterestsModel** з булевими властивостями (Sport, Travel, Craft, Draw) із встановленими дефолтними значеннями (наприклад, Sport і Craft – true, інші – false).
2. **Реалізація контролера:**
   * **GET метод:** Повертає View з новою моделлю **InterestsModel** і встановлює ViewBag.ShowResult = false, щоб не відображати результат при першому завантаженні.
   * **POST метод:** Приймає строго типізовану модель та рядковий параметр **action**. Якщо **action == "cancel"**, модель скидається до початкових значень (через створення нового об’єкта і очищення ModelState); інакше – зберігаються отримані дані, і ViewBag.ShowResult встановлюється на true.
3. **Створення вигляду (Index.cshtml):**
   * Використати директиву @model Task\_12.Models.InterestsModel для прив’язки моделі.
   * Розмістити чекбокси із використанням атрибута asp-for, що дозволяє автоматично пов’язувати їх з властивостями моделі.
   * Додати кнопки для дій (choose та cancel) і, при ShowResult = true, вивести список вибраних інтересів.

## **Функціональність:**

* Застосунок дозволяє користувачу вибрати інтереси через чекбокси, де значення автоматично пов’язуються із строго типізованою моделлю.
* При натисканні кнопки «Вибрати» дані з моделі відображаються, а кнопка «Відмовитися» повертає форму до початкового стану.

## **Висновок:**

Використання суворої типізації з моделлю **InterestsModel** покращує надійність і зручність обробки даних форми. Це забезпечує чистоту коду, автоматичну валідацію та простоту скидання форми до дефолтних значень, що сприяє ефективній розробці ASP.NET Core MVC-застосунку.

# **Завдання 13**

Створити застосунок, стартове подання (Index) якого містить форму зі списком, що розкривається (combo box, або «комбобокс») з N містами країни згідно свого варіанту (Варіант №12) та список, що прокручується (list box «листбокс») з аналогічним контентом. За допомогою стандартного класу Request контролер, викликаний кнопкою Відіслати приймає дані з форми та передає їх назад до подання для їхньої демонстрації під формою.

|  |  |  |
| --- | --- | --- |
| **№ варіанту** | **Країна** | **Кількість міст (N)** |
| 12 | Єгипет | 5 |

## **Текст програми:**

**Index.cshtml:**

@{

Layout = "\_Layout";

// Створюємо список міст Єгипту для комбобоксу та листбоксу

var cities = new List<string> { "Каїр", "Александрія", "Гіза", "Луксор", "Асуан" };

}

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Завдання 13 (Варіант №12)</title>

<style>

/\* Контейнер форми: позиціонуємо елементи абсолютно \*/

.form-container {

position: relative;

width: 600px;

height: 350px;

border: 1px solid #ccc;

margin-bottom: 20px;

padding: 10px;

background-color: rgba(255, 255, 255, 0.9);

border-radius: 10px;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

}

/\* Стилі для комбобоксу \*/

.comboLabel {

position: absolute;

left: 20px;

top: 20px;

font-weight: bold;

}

.comboBox {

position: absolute;

left: 20px;

top: 50px;

width: 200px;

border-radius: 5px;

border: 1px solid #ccc;

padding: 5px;

}

/\* Стилі для листбоксу \*/

.listLabel {

position: absolute;

left: 320px;

top: 20px;

font-weight: bold;

}

.listBox {

position: absolute;

left: 320px;

top: 50px;

width: 200px;

height: 100px;

border-radius: 5px;

border: 1px solid #ccc;

padding: 5px;

}

/\* Розташування кнопок \*/

.buttonChoose {

position: absolute;

left: 150px;

top: 280px;

}

.buttonReset {

position: absolute;

left: 270px;

top: 280px;

}

/\* Стиль для блоку з результатами \*/

.result-text {

color: white;

}

</style>

</head>

<body>

<!-- Форма для вибору міста через комбобокс та листбокс -->

<form method="post" asp-action="Index" asp-controller="Home" class="form-container">

<!-- Комбобокс -->

<div class="comboLabel">Комбобокс (розкривний список):</div>

<select name="comboCity" class="comboBox">

@foreach (var city in cities)

{

// Якщо місто співпадає з вибраним – додаємо selected

if (city == (ViewBag.ComboCity as string))

{

<option value="@city" selected>@city</option>

}

else

{

<option value="@city">@city</option>

}

}

</select>

<!-- Листбокс -->

<div class="listLabel">Листбокс (список):</div>

<select name="listCity" class="listBox" size="5">

@foreach (var city in cities)

{

// Аналогічно перевіряємо для листбоксу

if (city == (ViewBag.ListCity as string))

{

<option value="@city" selected>@city</option>

}

else

{

<option value="@city">@city</option>

}

}

</select>

<!-- Кнопки для дій -->

<div class="buttonChoose">

<button type="submit" name="action" value="choose" class="btn btn-primary">Відіслати</button>

</div>

<div class="buttonReset">

<button type="submit" name="action" value="reset" class="btn btn-secondary">Скинути</button>

</div>

</form>

@if (ViewBag.ShowResult == true)

{

<hr />

<!-- Відображення вибраних параметрів -->

<div class="result-text">

<h3>Вибрані параметри:</h3>

<p>Комбобокс: @ViewBag.ComboCity</p>

<p>Листбокс: @ViewBag.ListCity</p>

</div>

}

</body>

</html>

**HomeController.cs:**

using Microsoft.AspNetCore.Mvc;

namespace Task\_13.Controllers

{

public class HomeController : Controller

{

// GET: /Home/Index

[HttpGet]

public IActionResult Index()

{

// За замовчуванням вибираємо "Каїр" для комбобоксу та листбоксу

ViewBag.ComboCity = "Каїр";

ViewBag.ListCity = "Каїр";

ViewBag.ShowResult = false;

return View();

}

// POST: /Home/Index

[HttpPost]

[ActionName("Index")]

public IActionResult IndexPost()

{

// Зчитуємо значення кнопки з форми через Request

string action = Request.Form["action"];

if (action == "reset")

{

// Якщо натиснули "Скинути" – повертаємо дефолтні значення

ViewBag.ComboCity = "Каїр";

ViewBag.ListCity = "Каїр";

ViewBag.ShowResult = false;

}

else

{

// Інакше зчитуємо обрані значення з комбобоксу та листбоксу через Request

string comboCity = Request.Form["comboCity"];

string listCity = Request.Form["listCity"];

ViewBag.ComboCity = comboCity;

ViewBag.ListCity = listCity;

ViewBag.ShowResult = true;

}

return View();

}

}

}

**\_Layout.cshtml:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Task\_13</title>

<!-- Підключення Bootstrap -->

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

<!-- Основні стилі сайту -->

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

<!-- Додаткові стилі для Task\_13 -->

<link rel="stylesheet" href="~/Task\_13.styles.css" asp-append-version="true" />

</head>

<body>

<header>

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light custom-navbar border-bottom box-shadow mb-3">

<div class="container-fluid">

<!-- Назва сайту, посилання на Home -->

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Task\_13</a>

<!-- Кнопка для мобільного меню -->

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse" aria-controls="navbarSupportedContent"

aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<div class="container">

<main role="main" class="pb-3">

@RenderBody() <!-- Тут буде розміщено контент конкретної сторінки -->

</main>

</div>

<footer class="border-top footer text-white">

<div class="container">

&copy; 2025 - Task\_13 - <a asp-area="" asp-controller="Home" asp-action="Privacy" class="text-white">Privacy</a>

</div>

</footer>

<script src="~/lib/jquery/dist/jquery.min.js"></script>

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

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

**site.css:**

html {

font-size: 14px; /\* Базовий розмір шрифту \*/

}

@media (min-width: 768px) {

html {

font-size: 16px; /\* Шрифт для екранів ширших 768px \*/

}

}

.btn:focus, .btn:active:focus, .btn-link.nav-link:focus, .form-control:focus, .form-check-input:focus {

box-shadow: 0 0 0 0.1rem white, 0 0 0 0.25rem #258cfb;

}

html {

position: relative;

min-height: 100%;

}

body {

margin-bottom: 60px;

/\* Фон зображення для сторінки Task\_13 \*/

background: url('https://static.independent.co.uk/2023/01/19/17/Petit%20Jean%20State%20Park%203%20-%20Credit%20James%20Norbury%20and%20Guy%20Davies.jpg') no-repeat center center fixed;

background-size: cover;

}

/\* Стиль для навігаційного бару \*/

.custom-navbar {

background-color: rgba(255, 255, 255, 0.8) !important;

}

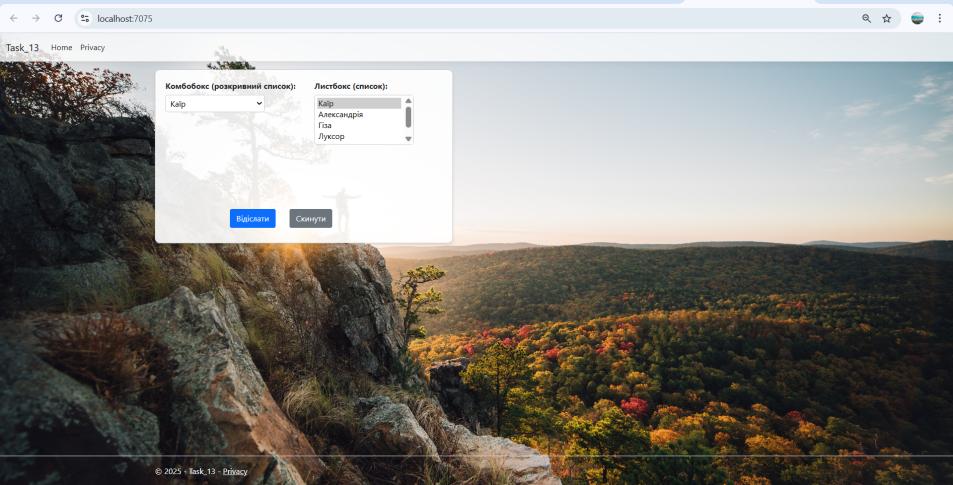
/\* Додаткове правило для футера \*/

footer.footer {

color: white;

}

## **Результати роботи програми:**



Изображение выглядит как текст, снимок экрана, дерево, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как снимок экрана, растение, на открытом воздухе, текст

Контент, сгенерированный ИИ, может содержать ошибки.

## **Алгоритм побудови проєкту і кодів:**

1. **Створення проєкту та налаштування View:**
   * Запустити ASP.NET Core MVC-проєкт із спільним Layout.
   * У файлі **Index.cshtml** створити список міст (наприклад, «Каїр», «Александрія», «Гіза», «Луксор», «Асуан») для заповнення комбобоксу та листбоксу.
2. **Реалізація контролера (HomeController):**
   * **GET метод:** Встановлює дефолтне значення для комбобоксу та листбоксу (наприклад, "Каїр") і встановлює прапорець, що результат не показується.
   * **POST метод:** За допомогою стандартного класу **Request** зчитує значення, вибрані користувачем із полів «comboCity» та «listCity».
     + Якщо натиснуто кнопку «Скинути», повертає дефолтне значення; інакше – оновлює ViewBag зі зчитаними значеннями та встановлює прапорець для відображення результату.
3. **Відображення результату:**
   * У View за допомогою умовного блоку перевіряється, чи встановлено ViewBag.ShowResult. Якщо так – відображаються обрані значення комбобоксу та листбоксу під формою.

## **Функціональність:**

* Користувач бачить два списки (комбобокс та листбокс) із заданим списком міст.
* При виборі значення і натисканні кнопки «Відіслати» контролер зчитує вибрані значення через **Request** і відображає їх під формою.
* Кнопка «Скинути» повертає форму до початкового стану із встановленими дефолтними значеннями.

## **Висновок:**

Даний приклад демонструє базову обробку даних форми в ASP.NET Core MVC із використанням комбобоксу та листбоксу. Застосунок ефективно зчитує вибрані значення через клас **Request** та повертає їх для демонстрації, що є важливим елементом взаємодії користувача із серверною логікою.

# **Завдання 14**

Виконати функції минулого завдання, використавши для отримання даних форми замість класу Request колекцію типу IFormCollection.

## **Текст програми:**

**Index.cshtml:**

@{

Layout = "\_Layout";

// Створюємо список міст Єгипту – 5 міст

var cities = new List<string> { "Каїр", "Александрія", "Гіза", "Луксор", "Асуан" };

}

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Завдання 14 (Варіант №12)</title>

<style>

/\* Контейнер форми: усі елементи розташовуються відносно цього блоку \*/

.form-container {

position: relative;

width: 600px;

height: 350px;

border: 1px solid #ccc;

margin-bottom: 20px;

padding: 10px;

background-color: rgba(255, 255, 255, 0.9);

border-radius: 10px;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

}

/\* Стилі для комбобоксу \*/

.comboLabel {

position: absolute;

left: 20px;

top: 20px;

font-weight: bold;

}

.comboBox {

position: absolute;

left: 20px;

top: 50px;

width: 200px;

border-radius: 5px;

border: 1px solid #ccc;

padding: 5px;

}

/\* Стилі для листбоксу \*/

.listLabel {

position: absolute;

left: 320px;

top: 20px;

font-weight: bold;

}

.listBox {

position: absolute;

left: 320px;

top: 50px;

width: 200px;

height: 100px;

border-radius: 5px;

border: 1px solid #ccc;

padding: 5px;

}

/\* Кнопки \*/

.buttonChoose {

position: absolute;

left: 150px;

top: 280px;

}

.buttonReset {

position: absolute;

left: 270px;

top: 280px;

}

/\* Стиль для блоку з результатами \*/

.result-text {

color: white;

}

</style>

</head>

<body>

<!-- Форма для вибору міста через комбобокс та листбокс -->

<form method="post" asp-action="Index" asp-controller="Home" class="form-container">

<!-- Комбобокс -->

<div class="comboLabel">Комбобокс (розкривний список):</div>

<select name="comboCity" class="comboBox">

@foreach (var city in cities)

{

// Якщо місто співпадає з вибраним, додаємо атрибут selected

if (city == (ViewBag.ComboCity as string))

{

<option value="@city" selected>@city</option>

}

else

{

<option value="@city">@city</option>

}

}

</select>

<!-- Листбокс -->

<div class="listLabel">Листбокс (список):</div>

<select name="listCity" class="listBox" size="5">

@foreach (var city in cities)

{

// Аналогічна перевірка для листбоксу

if (city == (ViewBag.ListCity as string))

{

<option value="@city" selected>@city</option>

}

else

{

<option value="@city">@city</option>

}

}

</select>

<!-- Кнопки -->

<div class="buttonChoose">

<button type="submit" name="action" value="choose" class="btn btn-primary">Відіслати</button>

</div>

<div class="buttonReset">

<button type="submit" name="action" value="reset" class="btn btn-secondary">Скинути</button>

</div>

</form>

@if (ViewBag.ShowResult == true)

{

<hr />

<!-- Відображення результатів вибору -->

<div class="result-text">

<h3>Вибрані параметри:</h3>

<p>Комбобокс: @ViewBag.ComboCity</p>

<p>Листбокс: @ViewBag.ListCity</p>

</div>

}

</body>

</html>

**HomeController.cs:**

using Microsoft.AspNetCore.Http; // Для роботи з IFormCollection

using Microsoft.AspNetCore.Mvc;

namespace Task\_14.Controllers

{

public class HomeController : Controller

{

// GET: /Home/Index

[HttpGet]

public IActionResult Index()

{

// За замовчуванням вибираємо "Каїр" для комбобоксу та листбоксу

ViewBag.ComboCity = "Каїр";

ViewBag.ListCity = "Каїр";

ViewBag.ShowResult = false;

return View();

}

// POST: /Home/Index

[HttpPost]

public IActionResult Index(IFormCollection form)

{

// Зчитуємо дію, натиснуту у формі (choose або reset)

string action = form["action"];

if (action == "reset")

{

// Якщо натиснули "Скинути", повертаємо дефолтні значення

ViewBag.ComboCity = "Каїр";

ViewBag.ListCity = "Каїр";

ViewBag.ShowResult = false;

}

else

{

// Якщо натиснули "Відіслати", читаємо значення з комбобоксу та листбоксу

string comboCity = form["comboCity"];

string listCity = form["listCity"];

ViewBag.ComboCity = comboCity;

ViewBag.ListCity = listCity;

ViewBag.ShowResult = true;

}

return View();

}

}

}

**\_Layout.cshtml:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Task\_14</title>

<!-- Підключення Bootstrap -->

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

<!-- Основні стилі сайту -->

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

<!-- Додаткові стилі для Task\_13 -->

<link rel="stylesheet" href="~/Task\_14.styles.css" asp-append-version="true" />

</head>

<body>

<header>

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light custom-navbar border-bottom box-shadow mb-3">

<div class="container-fluid">

<!-- Назва сайту з посиланням на Home -->

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Task\_14</a>

<!-- Кнопка для мобільного меню -->

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse"

aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<div class="container">

<main role="main" class="pb-3">

@RenderBody() <!-- Тут буде відображено контент конкретної сторінки -->

</main>

</div>

<footer class="border-top footer text-white">

<div class="container">

&copy; 2025 - Task\_14 - <a asp-area="" asp-controller="Home" asp-action="Privacy" class="text-white">Privacy</a>

</div>

</footer>

<script src="~/lib/jquery/dist/jquery.min.js"></script>

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

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

**site.css:**

html {

font-size: 14px; /\* Базовий розмір шрифту \*/

}

@media (min-width: 768px) {

html {

font-size: 16px; /\* Збільшуємо шрифт на широких екранах \*/

}

}

.btn:focus, .btn:active:focus, .btn-link.nav-link:focus, .form-control:focus, .form-check-input:focus {

box-shadow: 0 0 0 0.1rem white, 0 0 0 0.25rem #258cfb;

}

html {

position: relative;

min-height: 100%;

}

body {

margin-bottom: 60px;

/\* Фонова картинка для сторінки Task\_14 \*/

background: url('https://static.independent.co.uk/2023/01/19/17/Petit%20Jean%20State%20Park%203%20-%20Credit%20James%20Norbury%20and%20Guy%20Davies.jpg') no-repeat center center fixed;

background-size: cover;

}

/\* Стиль для навігаційного бару \*/

.custom-navbar {

background-color: rgba(255, 255, 255, 0.8) !important;

}

## **Результати роботи програми:**

Изображение выглядит как текст, снимок экрана, небо, на открытом воздухе

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, растение, небо

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как снимок экрана, текст, на открытом воздухе, растение

Контент, сгенерированный ИИ, может содержать ошибки.

## **Алгоритм побудови проєкту і кодів:**

1. **Створення проєкту:**
   * Запустити ASP.NET Core MVC-проєкт із спільним Layout (файл \_Layout.cshtml).
2. **Реалізація View (Index.cshtml):**
   * Створити список міст (наприклад, «Каїр», «Александрія», «Гіза», «Луксор», «Асуан») для заповнення комбобоксу та листбоксу.
   * Додати форму з двома елементами: комбобокс (select) і листбокс (select із атрибутом size="5").
   * Додати кнопки «Відіслати» (action = "choose") та «Скинути» (action = "reset") для керування вибором.
3. **Реалізація контролера (HomeController):**
   * **GET метод:** Встановлює дефолтні значення (наприклад, "Каїр") для комбобоксу та листбоксу і встановлює ViewBag.ShowResult = false.
   * **POST метод:** Використовуючи IFormCollection, зчитує значення полів "comboCity" та "listCity".
     + Якщо вибрано «Скинути», повертаються дефолтні значення.
     + Якщо «Відіслати» – зчитані значення записуються у ViewBag і встановлюється ViewBag.ShowResult = true для відображення результату.
4. **Відображення результату:**
   * У View перевірте ViewBag.ShowResult; при true відобразити вибрані параметри під формою.

## **Функціональність:**

* Користувач обирає місто із комбобоксу та листбоксу, що заповнені списком міст Єгипту.
* При натисканні «Відіслати» контролер зчитує дані через IFormCollection і відображає їх під формою.
* Кнопка «Скинути» повертає форму до початкових значень.

## **Висновок:**

Використання IFormCollection для зчитування даних із комбобоксу та листбоксу дозволяє ефективно обробляти вибір користувача та керувати станом форми. Такий підхід сприяє зручній взаємодії між користувачем і серверною логікою в ASP.NET Core MVC.

# **Завдання 15**

Виконати функції першого завдання, використавши для отримання да-них форми замість класу Request слабку типізацію форми, додавши до кожного списку форми відповідну змінну.

## **Текст програми:**

**Index.cshtml:**

@{

Layout = "\_Layout";

// Створюємо список з 5 міст Єгипту

var cities = new List<string> { "Каїр", "Александрія", "Гіза", "Луксор", "Асуан" };

}

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Завдання 15 (Варіант №12)</title>

<style>

/\* Контейнер форми – базовий блок для всіх елементів \*/

.form-container {

position: relative;

width: 600px;

height: 350px;

border: 1px solid #ccc;

margin-bottom: 20px;

padding: 10px;

background-color: rgba(255, 255, 255, 0.9);

border-radius: 10px;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

}

/\* Стилі для комбобоксу \*/

.comboLabel {

position: absolute;

left: 20px;

top: 20px;

font-weight: bold;

}

.comboBox {

position: absolute;

left: 20px;

top: 50px;

width: 200px;

border-radius: 5px;

border: 1px solid #ccc;

padding: 5px;

}

/\* Стилі для листбоксу \*/

.listLabel {

position: absolute;

left: 320px;

top: 20px;

font-weight: bold;

}

.listBox {

position: absolute;

left: 320px;

top: 50px;

width: 200px;

height: 100px;

border-radius: 5px;

border: 1px solid #ccc;

padding: 5px;

}

/\* Кнопки для дій \*/

.buttonChoose {

position: absolute;

left: 150px;

top: 280px;

}

.buttonReset {

position: absolute;

left: 270px;

top: 280px;

}

/\* Стиль для блоку з результатами \*/

.result-text {

color: white;

}

</style>

</head>

<body>

<!-- Форма для вибору міста через комбобокс і листбокс -->

<form method="post" asp-action="Index" asp-controller="Home" class="form-container">

<!-- Комбобокс -->

<div class="comboLabel">Комбобокс (розкривний список):</div>

<select name="comboCity" class="comboBox">

@foreach (var city in cities)

{

// Якщо місто співпадає з вибраним, додаємо атрибут selected

if (city == (ViewBag.ComboCity as string))

{

<option value="@city" selected>@city</option>

}

else

{

<option value="@city">@city</option>

}

}

</select>

<!-- Листбокс -->

<div class="listLabel">Листбокс (список):</div>

<select name="listCity" class="listBox" size="5">

@foreach (var city in cities)

{

// Аналогічна перевірка для списку

if (city == (ViewBag.ListCity as string))

{

<option value="@city" selected>@city</option>

}

else

{

<option value="@city">@city</option>

}

}

</select>

<!-- Кнопки для відправки або скидання вибору -->

<div class="buttonChoose">

<button type="submit" name="action" value="choose" class="btn btn-primary">Відіслати</button>

</div>

<div class="buttonReset">

<button type="submit" name="action" value="reset" class="btn btn-secondary">Скинути</button>

</div>

</form>

@if (ViewBag.ShowResult == true)

{

<hr />

<!-- Виведення результатів вибору -->

<div class="result-text">

<h3>Вибрані параметри:</h3>

<p>Комбобокс: @ViewBag.ComboCity</p>

<p>Листбокс: @ViewBag.ListCity</p>

</div>

}

</body>

</html>

**HomeController.cs:**

using Microsoft.AspNetCore.Mvc;

namespace Task\_15.Controllers

{

public class HomeController : Controller

{

// GET: /Home/Index

[HttpGet]

public IActionResult Index()

{

// За замовчуванням встановлюємо "Каїр" для комбобоксу та листбоксу

ViewBag.ComboCity = "Каїр";

ViewBag.ListCity = "Каїр";

ViewBag.ShowResult = false;

return View();

}

// POST: /Home/Index

// Слабка типізація: параметри форми приймаються як рядки

[HttpPost]

public IActionResult Index(string comboCity, string listCity, string action)

{

if (action == "reset")

{

// Якщо натиснуто "Скинути", повертаємо значення за замовчуванням

ViewBag.ComboCity = "Каїр";

ViewBag.ListCity = "Каїр";

ViewBag.ShowResult = false;

}

else

{

// Інакше зчитуємо вибрані значення з форми

ViewBag.ComboCity = comboCity;

ViewBag.ListCity = listCity;

ViewBag.ShowResult = true;

}

return View();

}

}

}

**\_Layout.cshtml:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Task\_15</title>

<!-- Підключення Bootstrap -->

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

<!-- Основні стилі сайту -->

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

<!-- Додаткові стилі для Task\_15 -->

<link rel="stylesheet" href="~/Task\_15.styles.css" asp-append-version="true" />

</head>

<body>

<header>

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light custom-navbar border-bottom box-shadow mb-3">

<div class="container-fluid">

<!-- Назва сайту з посиланням на головну сторінку -->

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Task\_15</a>

<!-- Кнопка для відкриття мобільного меню -->

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse"

aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<div class="container">

<!-- Основний контент сторінки -->

<main role="main" class="pb-3">

@RenderBody()

</main>

</div>

<footer class="border-top footer text-white">

<div class="container">

&copy; 2025 - Task\_15 - <a asp-area="" asp-controller="Home" asp-action="Privacy" class="text-white">Privacy</a>

</div>

</footer>

<script src="~/lib/jquery/dist/jquery.min.js"></script>

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

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

**site.css:**

html {

font-size: 14px; /\* Базовий розмір шрифту \*/

}

@media (min-width: 768px) {

html {

font-size: 16px; /\* Збільшення шрифту на великих екранах \*/

}

}

.btn:focus, .btn:active:focus, .btn-link.nav-link:focus, .form-control:focus, .form-check-input:focus {

box-shadow: 0 0 0 0.1rem white, 0 0 0 0.25rem #258cfb;

}

html {

position: relative;

min-height: 100%;

}

body {

margin-bottom: 60px;

/\* Фонова картинка для сторінки Task\_15 \*/

background: url('https://static.independent.co.uk/2023/01/19/17/Petit%20Jean%20State%20Park%203%20-%20Credit%20James%20Norbury%20and%20Guy%20Davies.jpg') no-repeat center center fixed;

background-size: cover;

}

/\* Стиль для навігаційного бару \*/

.custom-navbar {

background-color: rgba(255, 255, 255, 0.8) !important;

}

## **Результати роботи програми:**

Изображение выглядит как снимок экрана, текст, растение, пейзаж

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, растение, небо

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, на открытом воздухе, растение

Контент, сгенерированный ИИ, может содержать ошибки.

## **Алгоритм побудови проєкту і кодів:**

1. **Створення View:**
   * У файлі **Index.cshtml** створити список міст (наприклад, "Каїр", "Александрія", "Гіза", "Луксор", "Асуан") для заповнення комбобоксу та листбоксу.
   * Додати форму з двома елементами (select) із атрибутами name="comboCity" і name="listCity", а також кнопки «Відіслати» (action = "choose") і «Скинути» (action = "reset").
   * Використовувати умовну логіку для встановлення атрибуту **selected** залежно від збережених значень (через ViewBag).
2. **Реалізація контролера:**
   * **GET метод:** Встановлює дефолтні значення для комбобоксу та листбоксу (наприклад, "Каїр") і встановлює ViewBag.ShowResult = false.
   * **POST метод:** За допомогою слабкої типізації (рядкові параметри, напр. comboCity, listCity, action) зчитує значення з форми. При натисканні «Скинути» повертає дефолтні значення, інакше – зберігає вибрані значення в ViewBag та встановлює ViewBag.ShowResult = true для відображення результату.

## **Функціональність:**

* Користувач обирає місто через комбобокс і листбокс із заданого списку міст Єгипту.
* При натисканні «Відіслати» вибрані значення зчитуються та відображаються під формою, а кнопка «Скинути» повертає форму до початкових значень.

## **Висновок:**

Використання слабкої типізації для обробки даних комбобоксу та листбоксу дозволяє легко зчитувати та демонструвати вибрані значення. Такий підхід забезпечує зручність користувача і простоту керування станом форми в ASP.NET Core MVC.

# **\*Завдання 16**

Виконати функції першого завадння, використавши для отримання даних форми замість класу Request сувору типізацію форми, додавши до списків форми відповідну модель.

## **Текст програми:**

**Index.cshtml:**

@model Task\_16.Models.CitySelectionModel

@{

Layout = "\_Layout";

// Створюємо список міст Єгипту – 5 міст

var cities = new List<string> { "Каїр", "Александрія", "Гіза", "Луксор", "Асуан" };

}

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Завдання 16 (Варіант №12)</title>

<style>

/\* Контейнер форми у вигляді «карточки» \*/

.form-container {

position: relative;

width: 600px;

height: 350px;

border: 1px solid #ccc;

margin-bottom: 20px;

padding: 10px;

background-color: rgba(255, 255, 255, 0.9);

border-radius: 10px;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

}

/\* Стилі для комбобоксу \*/

.comboLabel {

position: absolute;

left: 20px;

top: 20px;

font-weight: bold;

}

.comboBox {

position: absolute;

left: 20px;

top: 50px;

width: 200px;

border-radius: 5px;

border: 1px solid #ccc;

padding: 5px;

}

/\* Стилі для листбоксу \*/

.listLabel {

position: absolute;

left: 320px;

top: 20px;

font-weight: bold;

}

.listBox {

position: absolute;

left: 320px;

top: 50px;

width: 200px;

height: 100px;

border-radius: 5px;

border: 1px solid #ccc;

padding: 5px;

}

/\* Розташування кнопок \*/

.buttonChoose {

position: absolute;

left: 150px;

top: 280px;

}

.buttonReset {

position: absolute;

left: 270px;

top: 280px;

}

/\* Стиль для виводу результатів \*/

.result-text {

color: white;

}

</style>

</head>

<body>

<!-- Форма для вибору міста через комбобокс та листбокс -->

<form method="post" asp-action="Index" asp-controller="Home" class="form-container">

<!-- Комбобокс -->

<div class="comboLabel">Комбобокс (розкривний список):</div>

<select asp-for="ComboCity" name="comboCity" class="comboBox">

@foreach (var city in cities)

{

// Якщо місто співпадає з вибраним у моделі – додаємо selected

if (city == Model.ComboCity)

{

<option value="@city" selected>@city</option>

}

else

{

<option value="@city">@city</option>

}

}

</select>

<!-- Листбокс -->

<div class="listLabel">Листбокс (список):</div>

<select asp-for="ListCity" name="listCity" class="listBox" size="5">

@foreach (var city in cities)

{

if (city == Model.ListCity)

{

<option value="@city" selected>@city</option>

}

else

{

<option value="@city">@city</option>

}

}

</select>

<!-- Кнопки -->

<div class="buttonChoose">

<button type="submit" name="action" value="choose" class="btn btn-primary">Відіслати</button>

</div>

<div class="buttonReset">

<button type="submit" name="action" value="reset" class="btn btn-secondary">Скинути</button>

</div>

</form>

@if (ViewBag.ShowResult == true)

{

<hr />

<!-- Вивід вибраних параметрів -->

<div class="result-text">

<h3>Вибрані параметри:</h3>

<p>Комбобокс: @Model.ComboCity</p>

<p>Листбокс: @Model.ListCity</p>

</div>

}

</body>

</html>

**HomeController.cs:**

using Microsoft.AspNetCore.Mvc;

using Task\_16.Models;

namespace Task\_16.Controllers

{

public class HomeController : Controller

{

// GET: /Home/Index

[HttpGet]

public IActionResult Index()

{

// Створюємо модель з дефолтними значеннями ("Каїр" для обох)

var model = new CitySelectionModel();

ViewBag.ShowResult = false;

return View(model);

}

// POST: /Home/Index

// Сувора типізація – дані форми прив'язані до властивостей моделі

[HttpPost]

public IActionResult Index(CitySelectionModel model, string action)

{

if (action == "reset")

{

// Якщо натиснуто "Скинути" – повертаємо дефолтну модель і очищаємо ModelState

model = new CitySelectionModel();

ViewBag.ShowResult = false;

ModelState.Clear(); // Очищуємо ModelState, щоб не було попередніх помилок

}

else

{

// Якщо натиснуто "Відіслати", приймаємо модель з форми

ViewBag.ShowResult = true;

}

return View(model);

}

}

}

\_Layout.cshtml:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Task\_16</title>

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

<!-- Підключення основних стилів сайту -->

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

<!-- Додаткові стилі для Task\_16 -->

<link rel="stylesheet" href="~/Task\_16.styles.css" asp-append-version="true" />

</head>

<body>

<header>

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light custom-navbar border-bottom box-shadow mb-3">

<div class="container-fluid">

<!-- Назва сайту, посилання на головну сторінку -->

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Task\_16</a>

<!-- Кнопка для мобільного меню -->

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse"

aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<div class="container">

<!-- Основний контент сторінки -->

<main role="main" class="pb-3">

@RenderBody()

</main>

</div>

<footer class="border-top footer text-white">

<div class="container">

&copy; 2025 - Task\_16 - <a asp-area="" asp-controller="Home" asp-action="Privacy" class="text-white">Privacy</a>

</div>

</footer>

<script src="~/lib/jquery/dist/jquery.min.js"></script>

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

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

**CitySelectionModel.cs:**

namespace Task\_16.Models

{

public class CitySelectionModel

{

// Властивості для комбобоксу та листбоксу

// За замовчуванням встановлено "Каїр" для обох

public string ComboCity { get; set; } = "Каїр";

public string ListCity { get; set; } = "Каїр";

}

}

**site.css:**

html {

font-size: 14px; /\* Базовий розмір шрифту \*/

}

@media (min-width: 768px) {

html {

font-size: 16px; /\* Шрифт для екранів ширших 768px \*/

}

}

.btn:focus, .btn:active:focus, .btn-link.nav-link:focus, .form-control:focus, .form-check-input:focus {

box-shadow: 0 0 0 0.1rem white, 0 0 0 0.25rem #258cfb;

}

html {

position: relative;

min-height: 100%;

}

body {

margin-bottom: 60px;

/\* Фонова картинка для сторінки Task\_16 \*/

background: url('https://static.independent.co.uk/2023/01/19/17/Petit%20Jean%20State%20Park%203%20-%20Credit%20James%20Norbury%20and%20Guy%20Davies.jpg') no-repeat center center fixed;

background-size: cover;

}

/\* Стиль для навігаційного бару \*/

.custom-navbar {

background-color: rgba(255, 255, 255, 0.8) !important;

}

## **Результати роботи програми:**

Изображение выглядит как снимок экрана, на открытом воздухе, облако, небо

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, растение, на открытом воздухе

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как снимок экрана, текст, растение, пейзаж

Контент, сгенерированный ИИ, может содержать ошибки.

## **Алгоритм побудови проєкту і кодів:**

1. **Створення моделі:**
   * Розробити клас **CitySelectionModel** з властивостями *ComboCity* і *ListCity*, які за замовчуванням встановлено в "Каїр".
2. **Реалізація контролера:**
   * **GET метод:** Повертає View з новою моделлю (де значення ComboCity і ListCity дорівнюють "Каїр") і встановлює ViewBag.ShowResult = false.
   * **POST метод:** Приймає строго типізовану модель і параметр action. При натисканні «Скинути» модель перезавантажується до дефолтних значень (ModelState очищається), інакше – значення з форми залишаються, а ViewBag.ShowResult встановлюється в true для відображення результату.
3. **Створення вигляду:**
   * Використовувати директиву @model Task\_16.Models.CitySelectionModel для прив’язки моделі.
   * Створити комбобокс і листбокс, заповнені списком міст, де атрибут asp-for забезпечує автоматичну прив’язку до властивостей моделі.
   * Додати кнопки для відправки («Відіслати») та скидання («Скинути») форми, після чого, при наявності ViewBag.ShowResult, відобразьте обрані значення.

## **Функціональність:**

* Користувач обирає місто із комбобоксу та листбоксу з переліку міст Єгипту.
* Після натискання кнопки «Відіслати» вибрані значення з моделі відображаються під формою.Натискання «Скинути» повертає форму до початкового стану з дефолтним вибором ("Каїр").

## **Висновок:**

Використання строго типізованої моделі (**CitySelectionModel**) для обробки даних форми покращує контроль за передачею даних, знижує ризик помилок і сприяє чистоті коду. Це забезпечує ефективну і зручну взаємодію між користувачем і сервером в ASP.NET Core MVC.

# **Загальний Висновок**

У ході виконання лабораторної роботи було досліджено різні методи обробки даних форм у середовищі **ASP.NET Core MVC**. Було реалізовано кілька варіантів отримання даних, зокрема:

* **Використання класу Request** для зчитування даних із форми.
* **Робота з колекцією IFormCollection** для отримання значень чекбоксів та інших елементів управління.
* **Впровадження слабкої та суворої типізації через моделі** для автоматичного прив’язування даних форми.
* **Обробка подій кнопок (вибір, скидання значень)**, що дозволяє динамічно змінювати стан форми та контролювати відображення введених даних.

У результаті проведених експериментів було підтверджено, що **сувора типізація через моделі є найефективнішим способом обробки даних**, оскільки вона зменшує кількість потенційних помилок і робить код більш структурованим. Проте використання Request та IFormCollection також має свої переваги, особливо у випадках, коли потрібно обробляти невизначену кількість параметрів.

Таким чином, робота дозволила отримати практичний досвід у **взаємодії між користувачем та сервером через форми**, а також продемонструвала різні підходи до обробки введених даних у веб-застосунках, що працюють на основі **ASP.NET Core MVC**.