**ЛАБОРАТОРНА РОБОТА № 2**

**Робота з ASP.NET MVC. Конфігурація проекту. Відображення даних з БД на веб-сторінках.**

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

**Хід роботи:**

**Завдання 1.** Створення та конфігурація проекту



Рис. 1. Тема проекту за варіантом.

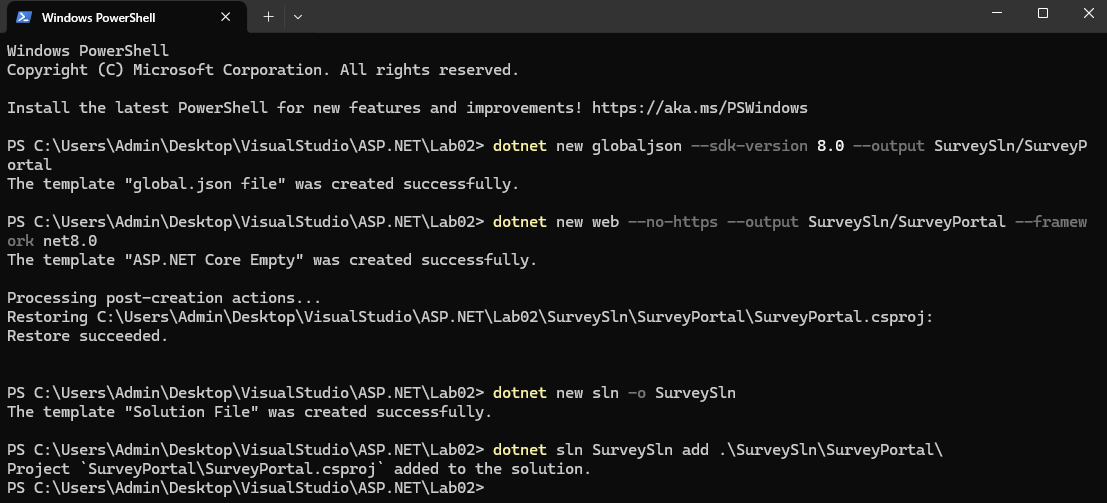


Рис. 2. Створення проекту.

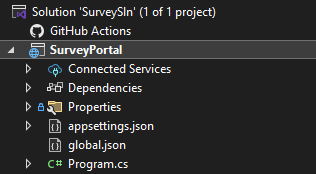


Рис. 3. Структура проекту.

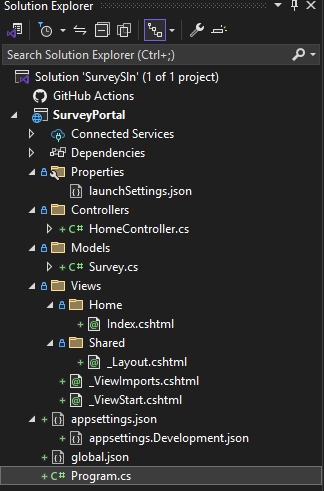


Рис. 4. Структура проекту після виконання завдання 1.

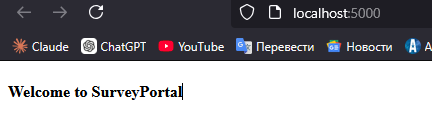


Рис. 5. Результат.

Завдання 2. Робота з ORM

Налаштувати роботу з БД, використовуючи інструкції.

Після встановлення пакетів, створення міграції, створення SeedData, інтерфейсу та EFSurveyRepository.cs структура програми виглядає наступним чином:

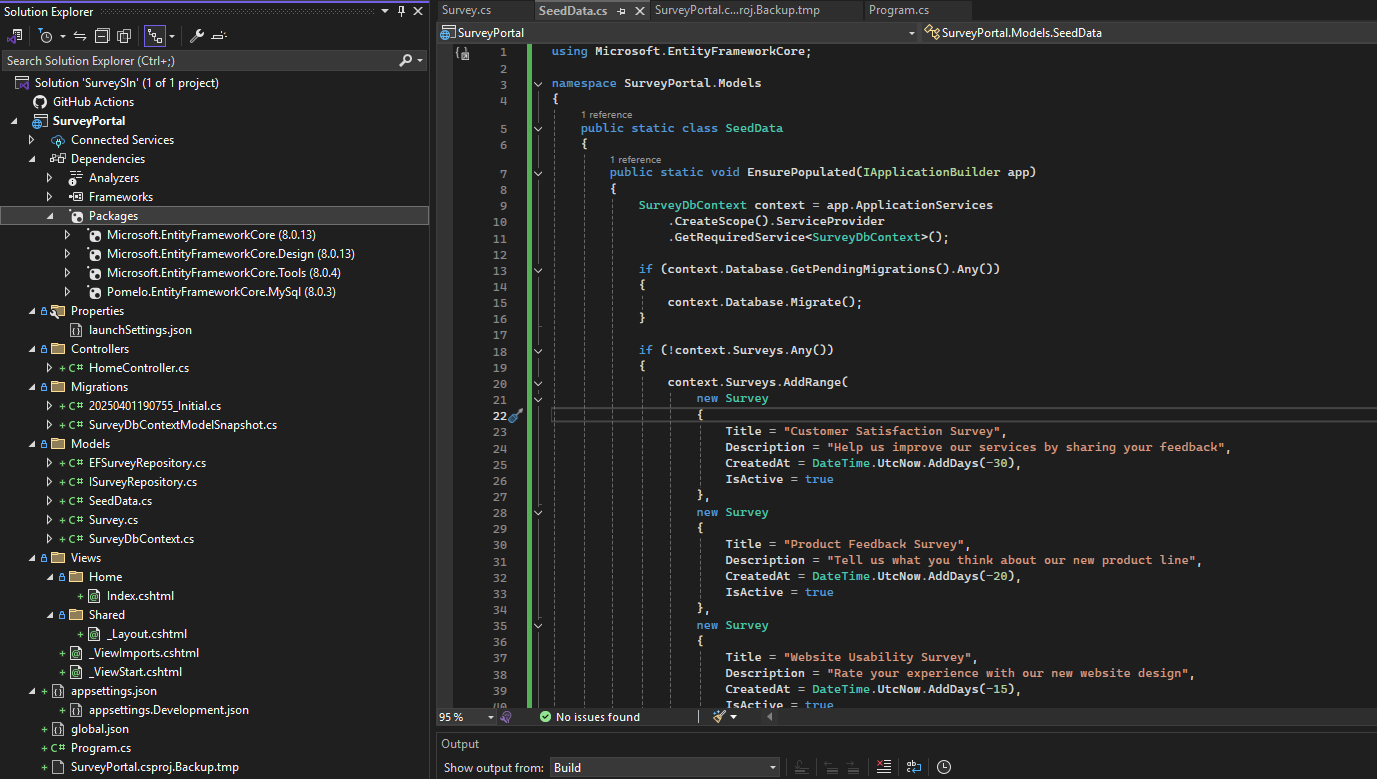


Рис. 6. Структура програми.

**Листинг Program.cs:**

using Microsoft.EntityFrameworkCore;

using SurveyPortal.Models;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllersWithViews();

builder.Services.AddDbContext<SurveyDbContext>(opts =>

{

var connectionString = builder.Configuration.GetConnectionString("SurveyPortalConnection");

opts.UseMySql(connectionString, ServerVersion.AutoDetect(connectionString));

});

builder.Services.AddScoped<ISurveyRepository, EFSurveyRepository>();

var app = builder.Build();

app.UseStaticFiles();

app.MapDefaultControllerRoute();

SeedData.EnsurePopulated(app);

app.Run();

**Листинг SeedData.cs:**

using Microsoft.EntityFrameworkCore;

namespace SurveyPortal.Models

{

public static class SeedData

{

public static void EnsurePopulated(IApplicationBuilder app)

{

SurveyDbContext context = app.ApplicationServices

.CreateScope().ServiceProvider

.GetRequiredService<SurveyDbContext>();

if (context.Database.GetPendingMigrations().Any())

{

context.Database.Migrate();

}

if (!context.Surveys.Any())

{

context.Surveys.AddRange(

new Survey

{

Title = "Customer Satisfaction Survey",

Description = "Help us improve our services by sharing your feedback",

CreatedAt = DateTime.UtcNow.AddDays(-30),

IsActive = true

},

new Survey

{

Title = "Product Feedback Survey",

Description = "Tell us what you think about our new product line",

CreatedAt = DateTime.UtcNow.AddDays(-20),

IsActive = true

},

new Survey

{

Title = "Website Usability Survey",

Description = "Rate your experience with our new website design",

CreatedAt = DateTime.UtcNow.AddDays(-15),

IsActive = true

},

new Survey

{

Title = "Market Research Survey",

Description = "Help us understand your preferences and needs",

CreatedAt = DateTime.UtcNow.AddDays(-10),

IsActive = true

},

new Survey

{

Title = "Employee Satisfaction Survey",

Description = "Internal survey for company staff",

CreatedAt = DateTime.UtcNow.AddDays(-5),

IsActive = false

},

new Survey

{

Title = "Event Feedback Survey",

Description = "Share your thoughts about our recent conference",

CreatedAt = DateTime.UtcNow.AddDays(-2),

IsActive = true

}

);

context.SaveChanges();

}

}

}

}

**Листинг EFSurveyRepository.cs та ISurveyRepository.cs:**

namespace SurveyPortal.Models

{

public interface ISurveyRepository

{

public interface IStoreRepository

{

IQueryable<Survey> Surveys { get; }

}

}

}

namespace SurveyPortal.Models

{

public class EFSurveyRepository : ISurveyRepository

{

private readonly SurveyDbContext \_context;

public EFSurveyRepository(SurveyDbContext context)

{

\_context = context;

}

public IQueryable<Survey> Surveys => \_context.Surveys;

}

}

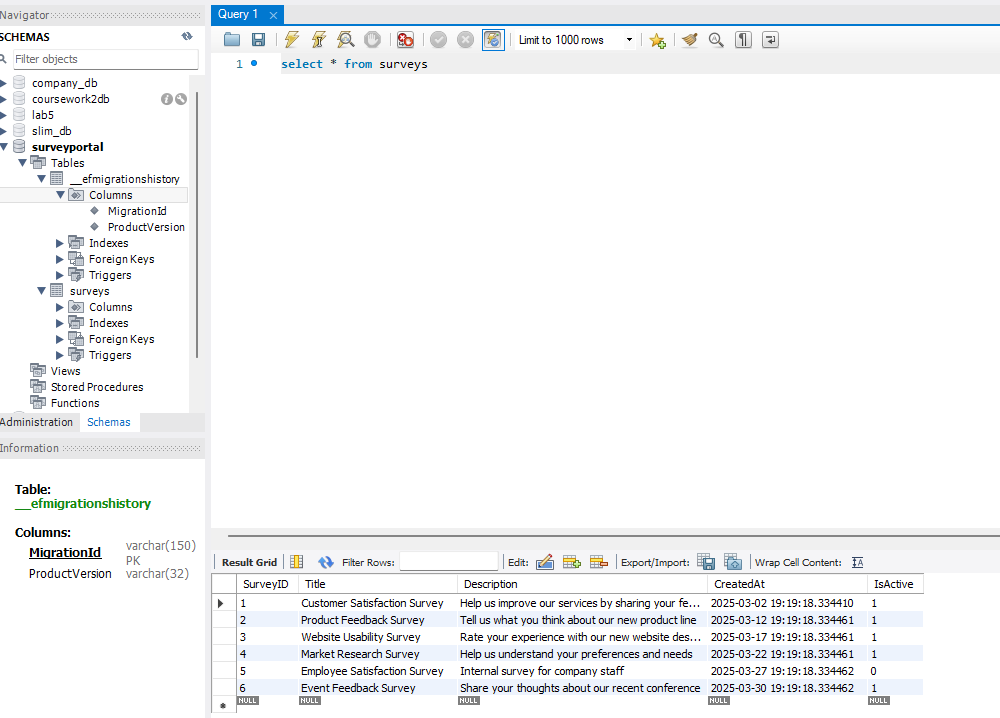
****

Рис. 7. База даних, що була створена автоматично.

**Завдання 3. Виведення даних на сторінці**

**Змініть контролер представлення для виведення даних з БД.**

**Листинг HomeController.cs**

using Microsoft.AspNetCore.Mvc;

using SurveyPortal.Models;

namespace SurveyPortal.Controllers

{

public class HomeController : Controller

{

private ISurveyRepository \_repository;

public HomeController(ISurveyRepository repository)

{

\_repository = repository;

}

public IActionResult Index() => View(\_repository.Surveys);

}

}

**Листинг Index.cshtml**

@model IQueryable<Survey>

@foreach (var p in Model ?? Enumerable.Empty<Survey>())

{

<div>

<h2>@p.Title</h2>

@p.Description

<p>Created: @p.CreatedAt</p>

@if(p.IsActive)

{

<h4>Survey is Active</h4>

}

else

{

<h4>Survey is Over</h4>

}

</div>

}

****

Рис. 8. Результат.

**Завдання 4. Реалізуйте посторінковий вивід записів.**

1. У папці Models створіть підпапку ViewModels. View Model – це сутність, створена спеціально для передачі даних між контролером та представленням, яка створюється під час роботи додатку і не зберігається в БД.

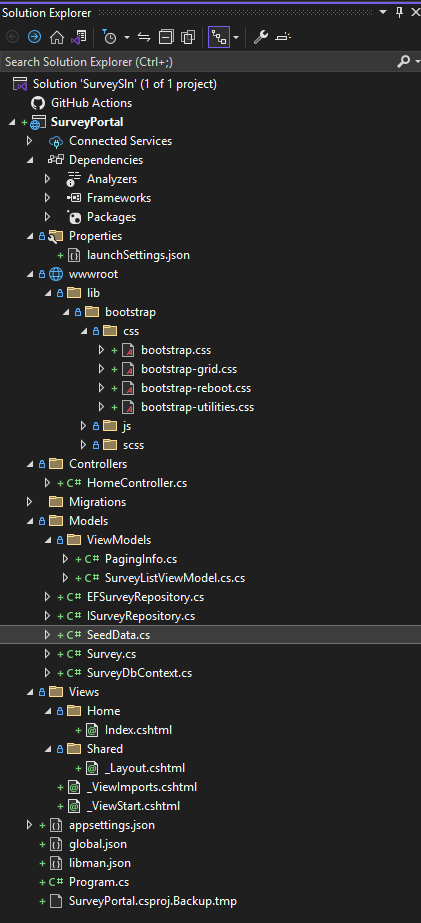


Рис. 9. Структура програми після виконання завдання(bootstrap завантажено).

1. В папці ViewModels створіть клас PagingInfo для збереження даних про пейджинг (кількість записів на сторінці, загальна кількість записів, кількість сторінок тощо).

**Листинг PagingInfo:**

namespace SurveyPortal.Models.ViewModels

{

public class PagingInfo

{

public int TotalItems { get; set; }

public int ItemsPerPage { get; set; }

public int CurrentPage { get; set; }

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

}

}

1. В папці ViewModels створіть View Model для посторінкового відображення списку сутностей. Вона повинна включати в собі колекцію сутностей та інформацію про пейджинг.

**Листинг SurveyListViewModel.cs:**

namespace SurveyPortal.Models.ViewModels

{

public class SurveyListViewModel

{

public IEnumerable<Survey> Surveys { get; set; } = new List<Survey>();

public PagingInfo PagingInfo { get; set; } = new PagingInfo();

}

}

1. Додайте підтримку пейджингу в контролер. Для цього змініть метод Index таким чином, щоб він приймав номер поточної сторінку та реалізуйте в ньому запит до БД для отримання елементів саме для цієї сторінки. У представлення передавайте екземпляр створеної View Model на кроці номер 3.
2. Адаптуйте представлення для відображення списку продуктів з View Model.
3. Додайте на сторінку елемент для вибору поточної сторінки.

**Листинг Index.cshtml (bootstrap):**

@model SurveyPortal.Models.ViewModels.SurveyListViewModel

<div class="row">

@foreach (var p in Model.Surveys)

{

<div class="col-md-6 mb-4">

<div class="card shadow-sm">

<div class="card-body">

<h2 class="card-title">@p.Title</h2>

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

<p><strong>Created:</strong> @p.CreatedAt.ToString("yyyy-MM-dd HH:mm:ss")</p>

<span class="badge bg-@(p.IsActive ? "success" : "secondary")">

@(p.IsActive ? "Active" : "Over")

</span>

</div>

</div>

</div>

}

</div>

<nav>

<ul class="pagination justify-content-center">

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

{

<li class="page-item @(i == Model.PagingInfo.CurrentPage ? "active" : "")">

<**a** class="page-link" **asp-controller**="Home" **asp-action**="Index" **asp-route-page**="@i">@i</**a**>

</li>

}

</ul>

</nav>

**Завдання 5. Стилізація**

1. Встановіть фреймворк Bootstrap
2. Змініть базовий шаблон сторінки (\_Layout.cshtml). Додайте елементи для хедера та футера.

**Листинг \_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"] - SurveyPortal</title>

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

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

</head>

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

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

<div class="container">

<h1 class="mb-0">Survey Portal</h1>

</div>

</header>

<main class="flex-grow-1 container mt-4">

@RenderBody()

</main>

<footer class="bg-dark text-white text-center py-3 mt-auto">

<p>&copy; @DateTime.Now.Year - Survey Portal</p>

</footer>

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

</body>

</html>

1. Стилізуйте список сутностей та пейджинг.

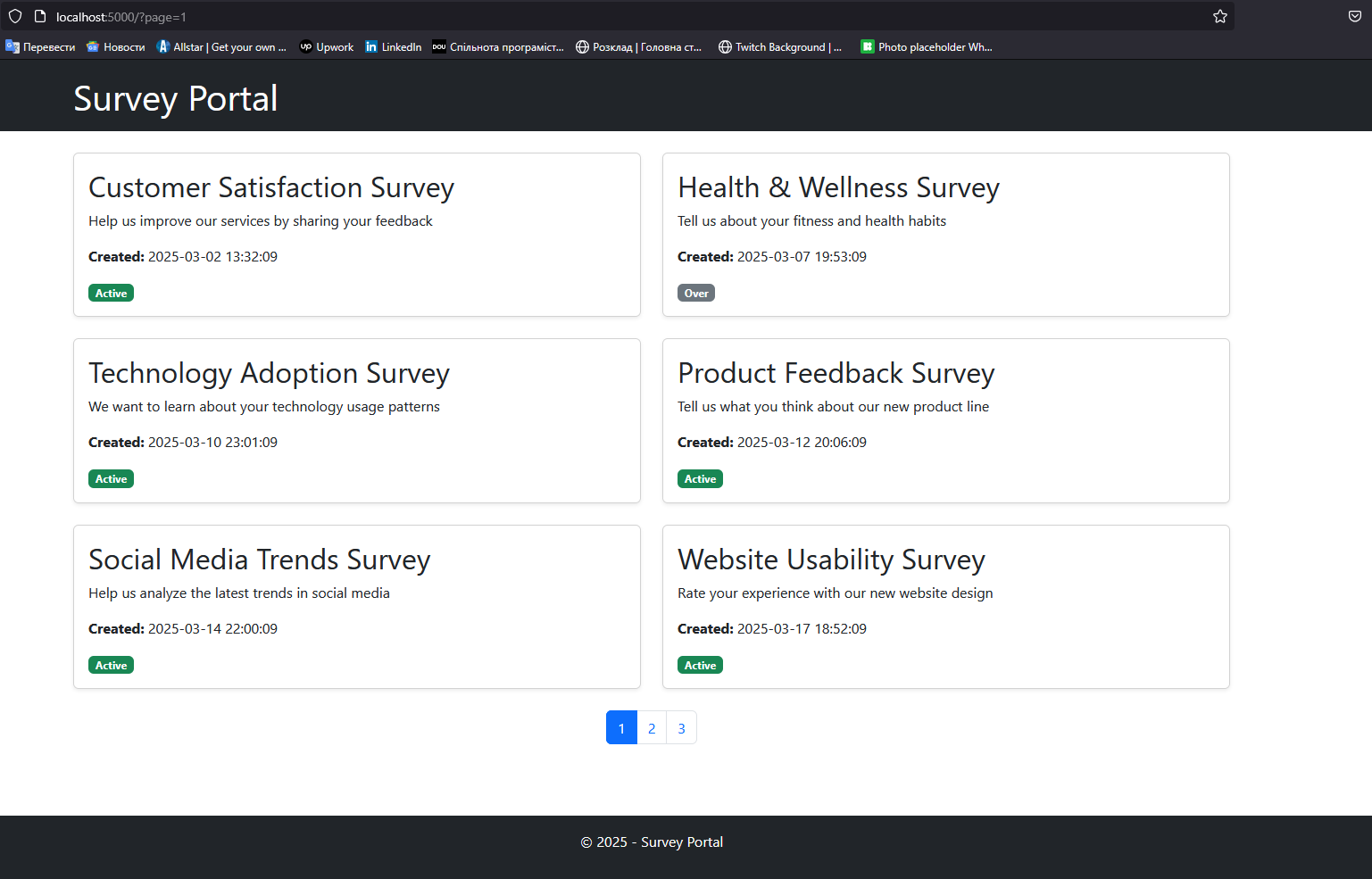


Рис. 10. Фінальний результат.

Посилання на репозиторій: <https://github.com/JanRizhenko/ASP.NET>

***Висновок:*** У лабораторній роботі було ознайомлено з архітектурою MVC, налаштуванням підключення до БД та виконанням запитів для відображення даних на веб-сторінках. Опановано створення основних компонентів MVC та роботу з представленнями.