

## Лабораторна робота №2

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

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

**Хід роботи:**

**Завдання 1.** Створення та конфігурація проекту.

3	Портал вакансій	Роботодавці можуть публікувати списки вакансій, а кандидати можуть відгукуватися на вакансії та керувати своїми відгуками.
---	-----------------	--

**Лістинг launchSettings.json у папці Properties:**

```
{
  "iisSettings": {
    "windowsAuthentication": false,
    "anonymousAuthentication": true,
    "iisExpress": {
      "applicationUrl": "http://localhost:5000/",
      "sslPort": 0
    }
  },
  "profiles": {
    "JobPortal": {
      "commandName": "Project",
      "dotnetRunMessages": true,
      "launchBrowser": true,
      "applicationUrl": "http://localhost:5000/",
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    },
    "IIS Express": {
      "commandName": "IISExpress",
      "launchBrowser": true,
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    }
  }
}
```

Далі перейдемо до створення першої сутності моделі даних.

**Лістинг Models.Vacancy:**

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

namespace JobPortal.Models
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.13.000 – Лр.1						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Кравченко К.М.			Звіт з лабораторної роботи №2			Лім.	Арк.	Аркушів	
Перевір.		Українець М.О.								1	
Реценз.								ФІКТ, гр. ІПЗ-22-2			
Н. Контр.											
Зав.каф.											

```
{
    public class Vacancy
    {
        public long? VacancyID { get; set; }
        public string Title { get; set; } = string.Empty;
        public string Description { get; set; } = string.Empty;
        public string Company { get; set; } = string.Empty;
        public string Location { get; set; } = string.Empty;

        [Column(TypeName = "decimal(10, 2)")]
        public decimal Salary { get; set; }

        public DateTime PostedDate { get; set; } = DateTime.Now;
    }
}
```

## Завдання 2. Робота з ORM.

### Лістинг appsettings.json:

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

Entity Framework Core надає доступ до бази даних через клас контексту. Для проекту JobPortal буде додано файл класу з ім'ям PortalDbContext.cs до папки Models.

### Лістинг PortalDbContext.cs:

```
using Microsoft.EntityFrameworkCore;

namespace JobPortal.Models
{
    public class PortalDbContext : DbContext
    {
        public PortalDbContext(DbContextOptions<PortalDbContext> options) :
base(options) { }
        public DbSet<Vacancy> Vacancies => Set<Vacancy>();
    }
}
```

### Необхідні зміни в Program.cs:

```
using Microsoft.EntityFrameworkCore;
using JobPortal.Models;
...
builder.Services.AddDbContext<PortalDbContext>(opts => {
    opts.UseSqlServer(builder.Configuration["ConnectionStrings:JobPortalConnection"]);
});
```

Для реалізації репозиторію для JobPortal, створимо файл з назвою IPortalRepository.cs.

### Лістинг:

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.13.000 – Лр.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```
namespace JobPortal.Models;

public interface IPortalRepository
{
    IQueryable<Vacancy> Vacancies { get; }
}
```

Щоб створити реалізацію інтерфейсу репозиторію, додамо файл класу з іменем EFPortalRepository.cs в папку Models.

**Лістинг:**

```
namespace JobPortal.Models
{
    public class EFPortalRepository : IPortalRepository
    {
        private PortalDbContext context;
        public EFPortalRepository(PortalDbContext ctx)
        {
            context = ctx;
        }
        public IQueryable<Vacancy> Vacancies => context.Vacancies;
    }
}
```

Змінимо Program.cs таким чином, щоб створити сервіс для інтерфейсу IPortalRepository, який використовує EFPortalRepository як клас реалізації.

**Лістинг:**

```
using Microsoft.EntityFrameworkCore;
using JobPortal.Models;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllersWithViews();

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

builder.Services.AddScoped<IPortalRepository, EFPortalRepository>();

var app = builder.Build();
```

```
*****
PS D:\4 курс\321\Project\JobPortal> dotnet ef migrations add Initial
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'
PS D:\4 курс\321\Project\JobPortal>
```

**Рис. 2.1. Успішне виконання команди**

Щоб наповнити БД даними одразу після створення, реалізуємо статичний клас SeedData в папці Models. **Лістинг файлу SeedData.cs:**

```
using Microsoft.EntityFrameworkCore;

namespace JobPortal.Models
{
    public static class SeedData
    {

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.13.000 – Лр.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public static void EnsurePopulated(IApplicationBuilder app)
{
    PortalDbContext context = app.ApplicationServices
        .CreateScope().ServiceProvider
        .GetRequiredService<PortalDbContext>();

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

    if (!context.Vacancies.Any())
    {
        context.Vacancies.AddRange(
            new Vacancy
            {
                Title = "Junior .NET Developer",
                Company = "SoftLine Group",
                Description = "Розробка веб-додатків на платформі .NET,  
робота з базами даних, підтримка існуючих систем.",
                Location = "Київ",
                Salary = 25000,
                PostedDate = DateTime.Now
            },
            new Vacancy
            {
                Title = "Frontend Developer (React)",
                Company = "TechVision",
                Description = "Розробка клієнтської частини веб-додатків з  
використанням React, HTML, CSS.",
                Location = "Львів",
                Salary = 32000,
                PostedDate = DateTime.Now
            },
            new Vacancy
            {
                Title = "Project Manager",
                Company = "Innovate Solutions",
                Description = "Управління командою розробників, планування  
проектів, комунікація з клієнтами.",
                Location = "Одеса",
                Salary = 40000,
                PostedDate = DateTime.Now
            },
            new Vacancy
            {
                Title = "QA Engineer",
                Company = "TestPro",
                Description = "Розробка тест-кейсів, проведення ручного та  
автоматизованого тестування.",
                Location = "Харків",
                Salary = 27000,
                PostedDate = DateTime.Now
            },
            new Vacancy
            {
                Title = "UI/UX Designer",
                Company = "Creative Minds",
                Description = "Створення макетів інтерфейсів, покращення  
користувацького досвіду, робота з Figma.",
                Location = "Дніпро",
                Salary = 29000,
                PostedDate = DateTime.Now
            }
        );
    }
}

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.13.000 – Лр.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        );
        context.SaveChanges();
    }
}
}

```

Останнім кроком є заповнення бази даних при запуску додатку. Для цього додамо виклик методу EnsurePopulated в Program.cs.

#### *Лістинг:*

```

SeedData.EnsurePopulated(app);

app.Run();

```

**Завдання 3.** Виведення даних на сторінці.

#### *Лістинг файлу HomeController.cs:*

```

using Microsoft.AspNetCore.Mvc;
using JobPortal.Models;
using JobPortal.Models.ViewModels;

namespace JobPortal.Controllers
{
    public class HomeController : Controller
    {
        private readonly IPortalRepository repository;
        private const int PageSize = 4; // кількість вакансій на сторінці

        public HomeController(IPortalRepository repo)
        {
            repository = repo;
        }

        public IActionResult Index(int page = 1)
        {
            var vacancies = repository.Vacancies
                .OrderBy(v => v.VacancyID)
                .Skip((page - 1) * PageSize)
                .Take(PageSize);

            var viewModel = new VacanciesListViewModel
            {
                Vacancies = vacancies,
                PagingInfo = new PagingInfo
                {
                    CurrentPage = page,
                    ItemsPerPage = PageSize,
                    TotalItems = repository.Vacancies.Count()
                }
            };

            return View(viewModel);
        }
    }
}

```

#### *Лістинг файлу Views/Home/Index.cshtml:*

```

@model JobPortal.Models.ViewModels.VacanciesListViewModel

<h2>Список вакансій</h2>

@foreach (var v in Model.Vacancies)
{
    <div class="card mb-3 p-3 border border-secondary rounded">

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.13.000 – Лр.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<h3>@v.Title</h3>
<p>@v.Description</p>
<p><strong>Компанія:</strong> @v.Company</p>
<p><strong>Місто:</strong> @v.Location</p>
</div>
}

<!-- Пагінація -->
<div class="pagination">
    @for (int i = 1; i <= Model.PagingInfo.TotalPages; i++)
    {
        if (i == Model.PagingInfo.CurrentPage)
        {
            <span class="btn btn-primary">@i</span>
        }
        else
        {
            <a class="btn btn-outline-primary" asp-action="Index" asp-route-
page="@i">@i</a>
        }
    }
</div>

```

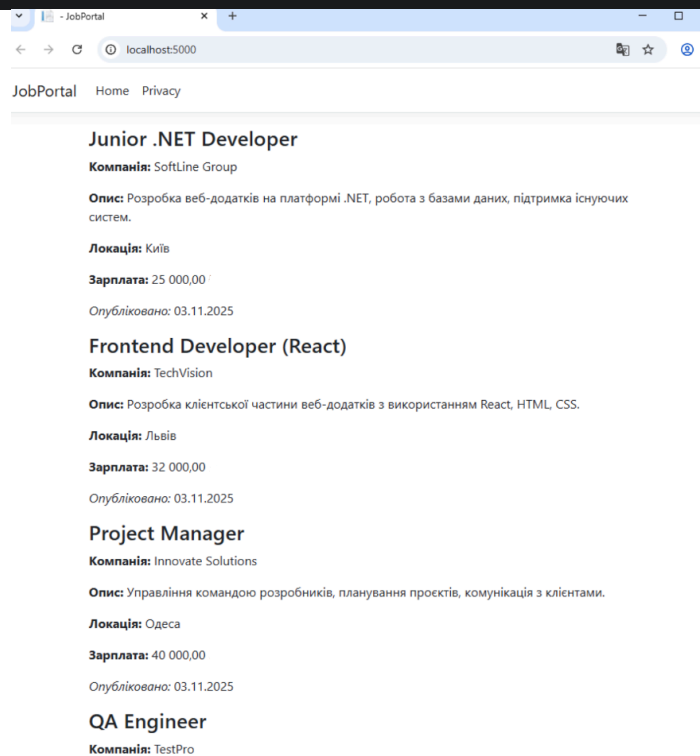


Рис. 3.1. Виведення даних на представлення

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

**Лістинг:**

```

namespace JobPortal.Models.ViewModels
{
    public class PagingInfo

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.13.000 – Лр.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    public int TotalItems { get; set; }
    public int ItemsPerPage { get; set; }
    public int CurrentPage { get; set; }

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

```

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

*Лістинг:*

```

namespace JobPortal.Models.ViewModels
{
    public class VacanciesListViewModel
    {
        public IEnumerable<Vacancy> Vacancies { get; set; } =
            Enumerable.Empty<Vacancy>();
        public PagingInfo PagingInfo { get; set; } = new PagingInfo();
    }
}

```

4. Додайте підтримку пейджингу в контролер. Для цього змініть метод Index таким чином, щоб він приймав номер поточної сторінки 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.

*Лістинг:*

```

using Microsoft.AspNetCore.Mvc;
using JobPortal.Models;
using JobPortal.Models.ViewModels;

namespace JobPortal.Controllers
{
    public class HomeController : Controller
    {
        private readonly IPortalRepository repository;
        private const int PageSize = 4; // кількість вакансій на сторінці

        public HomeController(IPortalRepository repo)
        {
            repository = repo;
        }

        public IActionResult Index(int page = 1)
        {
            var vacancies = repository.Vacancies
                .OrderBy(v => v.VacancyID)
                .Skip((page - 1) * PageSize)
                .Take(PageSize);

            var viewModel = new VacanciesListViewModel
            {
                Vacancies = vacancies,
                PagingInfo = new PagingInfo
                {
                    CurrentPage = page,
                    ItemsPerPage = PageSize,
                    TotalItems = repository.Vacancies.Count()
                }
            };
        }
    }
}

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.13.000 – Лр.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```
        return View(viewModel);
    }
}
```

5. Адаптуйте представлення для відображення списку продуктів з View Model.  
**Лістинг:**

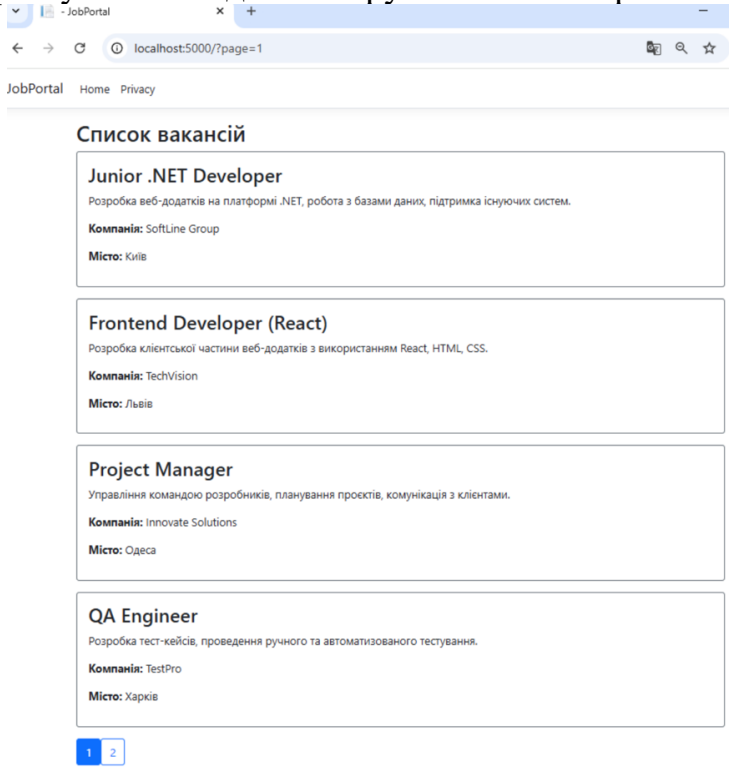
```
@model JobPortal.Models.ViewModels.VacanciesListViewModel

<h2>Список вакансій</h2>

@foreach (var v in Model.Vacancies)
{
    <div class="card mb-3 p-3 border border-secondary rounded">
        <h3>@v.Title</h3>
        <p>@v.Description</p>
        <p><strong>Компанія:</strong> @v.Company</p>
        <p><strong>Місто:</strong> @v.Location</p>
    </div>
}

<!-- Пагінація -->
<div class="pagination">
    @for (int i = 1; i <= Model.PagingInfo.TotalPages; i++)
    {
        if (i == Model.PagingInfo.CurrentPage)
        {
            <span class="btn btn-primary">@i</span>
        }
        else
        {
            <a class="btn btn-outline-primary" asp-action="Index" asp-route-
page="@i">@i</a>
        }
    }
</div>
```

6. Додайте на сторінку елемент для вибору поточної сторінки.





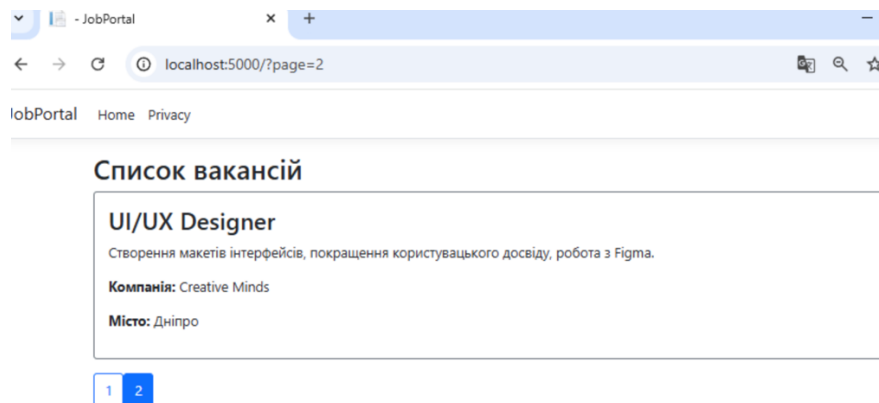


Рис. 4.6.1.- 4.6.2. Виведення даних на представлення

## Завдання 5. Стилiзацiя.

### 1. Встановiть фреймворк Bootstrap.

```
PS D:\4 курс\321\Project\JobPortal> dotnet tool install --global Microsoft.Web.LibraryManager.Cli
Tool 'microsoft.web.librarymanager.cli' was reinstalled with the stable version (version '3.0.71').
PS D:\4 курс\321\Project\JobPortal> libman init -p cdnjs
PS D:\4 курс\321\Project\JobPortal> libman install bootstrap -d wwwroot/lib/bootstrap
>>
D:/4 курс/321/Project/JobPortal/wwwroot/lib/bootstrap/scss/vendors
Installed library "bootstrap@5.3.8" to "wwwroot/lib/bootstrap"
PS D:\4 курс\321\Project\JobPortal>
```

Рис. 5.1. Встановлено фреймворк Bootstrap

### 2. Змiнiть базовий шаблон сторiнки (\_Layout.cshtml). Додайте елементи для хедера та футера.

#### Лiстинг:

```
<!DOCTYPE html>
<html lang="uk">
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>@ViewData["Title"] - JobPortal</title>

<link rel="stylesheet" href="~/lib/bootstrap/css/bootstrap.min.css" />
<link rel="stylesheet" href="~/css/site.css" />
</head>
<body>
<header class="bg-primary text-white p-3 mb-4">
<div class="container">
<h1 class="display-6">JobPortal</h1>
<p class="lead">Портал вакансій для роботодавцiв i кандидатiв</p>
</div>
</header>

<main class="container mb-5">
@RenderBody()
</main>

<footer class="bg-light text-center py-3 border-top">
<div class="container">
<p class="mb-0">&copy; @DateTime.Now.Year - JobPortal. Усi права захищено.</p>
</div>
</footer>

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

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.13.000 – Лр.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```
</body>
</html>
```

### 3. Стилiзуйте список сутностей та пейджинг.

#### *Лiстинг:*

```
@model JobPortal.Models.ViewModels.VacanciesListViewModel

@{
    ViewData["Title"] = "Вакансiї";
}

<div class="row">
    @foreach (var v in Model.Vacancies)
    {
        <div class="col-md-4 mb-4">
            <div class="card h-100 shadow-sm">
                <div class="card-body">
                    <h5 class="card-title">@v.Title</h5>
                    <h6 class="card-subtitle text-muted">@v.Company</h6>
                    <p class="card-text mt-2">@v.Description</p>
                    <p><strong>Мiсцезнаходження:</strong> @v.Location</p>
                    <p><strong>Зарплата:</strong> @v.Salary.ToString("C", new
System.Globalization.CultureInfo("uk-UA"))</p>
                </div>
            </div>
        </div>
    }
</div>

<!-- ♦ Пейджинг -->
<nav aria-label="Навiгацiя сторiнками">
    <ul class="pagination justify-content-center mt-4">
        @for (int i = 1; i <= Model.PagingInfo.TotalPages; i++)
        {
            <li class="page-item @(i == Model.PagingInfo.CurrentPage ? "active" : "")">
                <a class="page-link" asp-action="Index" asp-route-page="@i">@i</a>
            </li>
        }
    </ul>
</nav>
```

#### *Результат:*

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.13.000 – Лр.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

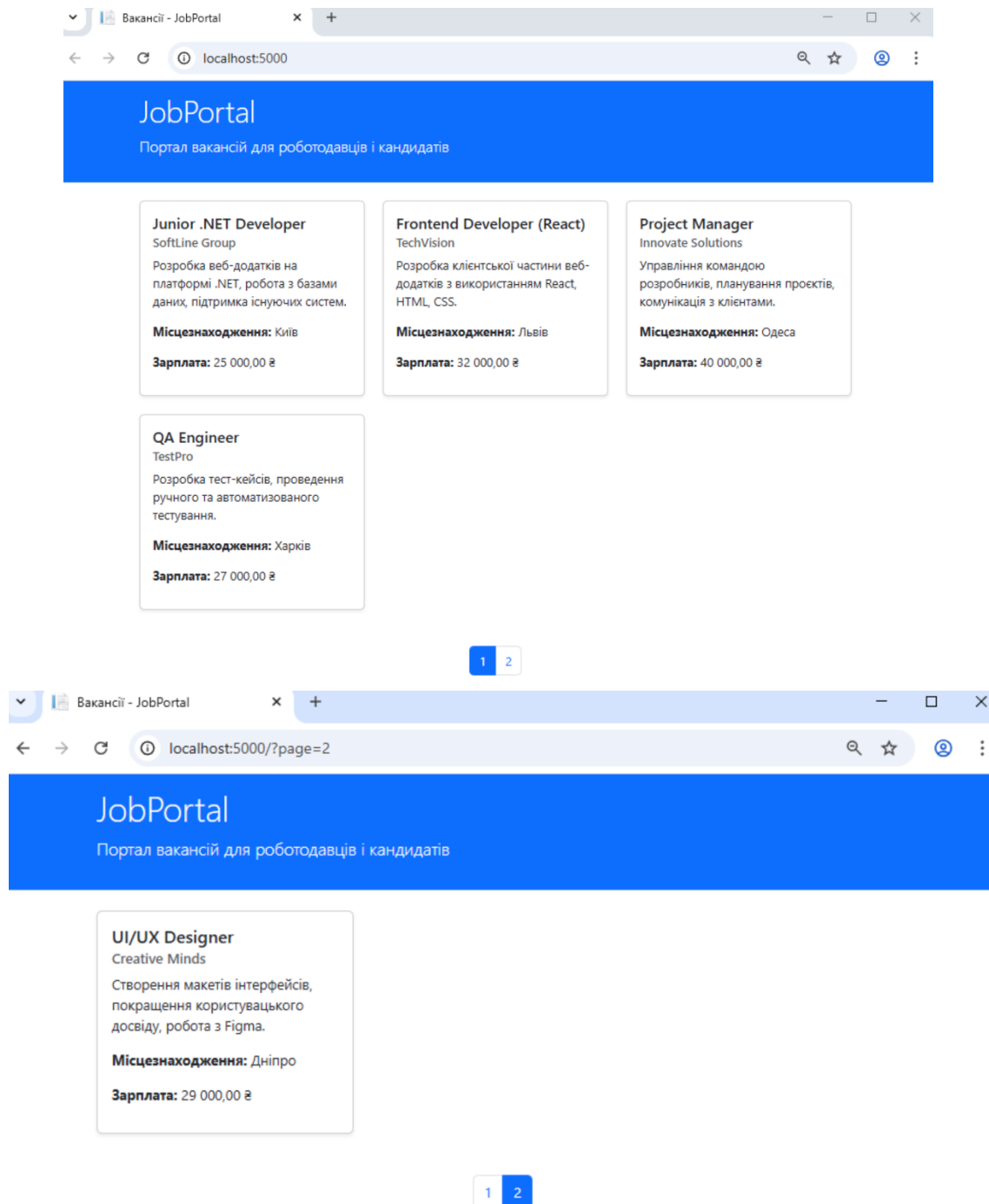


Рис. 5.3.1. – 5.3.2. Фінальний результат

**Висновок:** У ході виконання лабораторної роботи було опановано принципи роботи з архітектурою ASP.NET MVC. Я налаштувала структуру проєкту, розглянула призначення основних компонентів — Model, View та Controller. Також було виконано підключення до бази даних, налаштування рядка з'єднання та реалізовано отримання даних з БД для їх відображення на веб-сторінках. Окрім цього, я отримала практичні навички роботи з представленнями (Views) для виведення інформації користувачу. Таким чином, поставлену мету було досягнуто, а необхідні теоретичні знання та практичні навички — закріплено.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.13.000 – Лр.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		