Чугунов В.Ю  
КІТ-119а

Лабораторна робота №8

**ОСНОВИ ASP.NET MVC**

Завдання:

1. Створити веб-застосунок за допомогою ASP.NET MVC.
2. Реалізувати додавання студенту у список, його видалення та редагування. Використовувати валідацію даних при їх введенні.

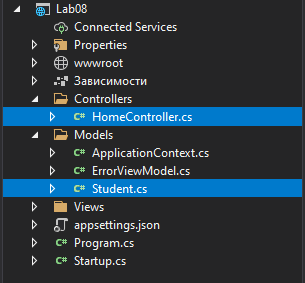
Структура проекту  


Рисунок 1 – Структура проекту

Опис класів

Student – відповідна модель студента;

HomeController – клас, що відповідає за роботу із класом Student, відповідає класу контролеру;

**Текст файлу Student.cs**

namespace Lab08.Models

{

public class Student

{

public int StudentId { get; set; }

public string FirstName { get; set; }

public string SecondName { get; set; }

public string Faculty { get; set; }

public int AcademicPerformance { get; set; }

public int Age { get; set; }

}

}

**Текст файлу HomeController.cs**

using Lab08.Models;

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

using Microsoft.Extensions.Logging;

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Threading.Tasks;

namespace Lab08.Controllers

{

public class HomeController : Controller

{

private ApplicationContext \_db;

public HomeController(ApplicationContext context)

{

\_db = context;

}

public async Task<IActionResult> Edit(int? id)

{

if (id != null)

{

Student student = await \_db.Students.FirstOrDefaultAsync(p => p.StudentId == id);

if (student != null)

return View(student);

}

return NotFound();

}

[HttpPost]

public async Task<IActionResult> Edit(Student student)

{

\_db.Students.Update(student);

await \_db.SaveChangesAsync();

return RedirectToAction("Index");

}

[HttpGet]

[ActionName("Delete")]

public async Task<IActionResult> ConfirmDelete(int? id)

{

if (id != null)

{

Student student = await \_db.Students.FirstOrDefaultAsync(p => p.StudentId == id);

if (student != null)

return View(student);

}

return NotFound();

}

[HttpPost]

public async Task<IActionResult> Delete(int? id)

{

if (id != null)

{

Student student = await \_db.Students.FirstOrDefaultAsync(p => p.StudentId == id);

if (student != null)

{

\_db.Students.Remove(student);

await \_db.SaveChangesAsync();

return RedirectToAction("Index");

}

}

return NotFound();

}

public async Task<IActionResult> Index()

{

return View(await \_db.Students.ToListAsync());

}

public IActionResult Create()

{

return View();

}

[HttpPost]

public async Task<IActionResult> Create(Student student)

{

\_db.Students.Add(student);

await \_db.SaveChangesAsync();

return RedirectToAction("Index");

}

public IActionResult Privacy()

{

return View();

}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]

public IActionResult Error()

{

return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

}

}

}

**Текст файлу Programe.cs**

using Microsoft.AspNetCore.Hosting;

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.Hosting;

using Microsoft.Extensions.Logging;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace Lab08

{

public class Program

{

public static void Main(string[] args)

{

CreateHostBuilder(args).Build().Run();

}

public static IHostBuilder CreateHostBuilder(string[] args) =>

Host.CreateDefaultBuilder(args)

.ConfigureWebHostDefaults(webBuilder =>

{

webBuilder.UseStartup<Startup>();

});

}

}

**РЕЗУЛЬТАТ РОБОТИ**

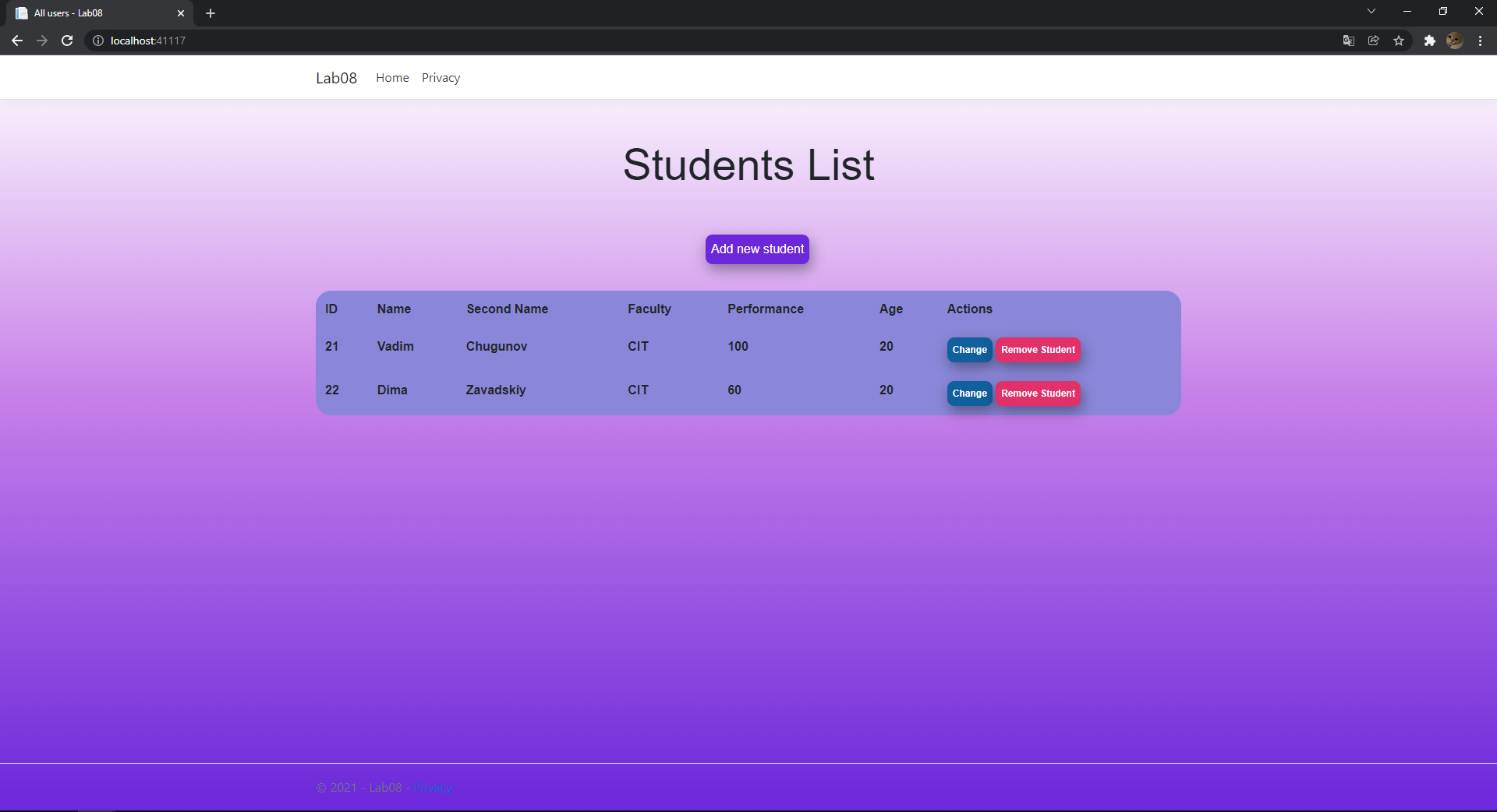


Рисунок 2 - Головна сторінка

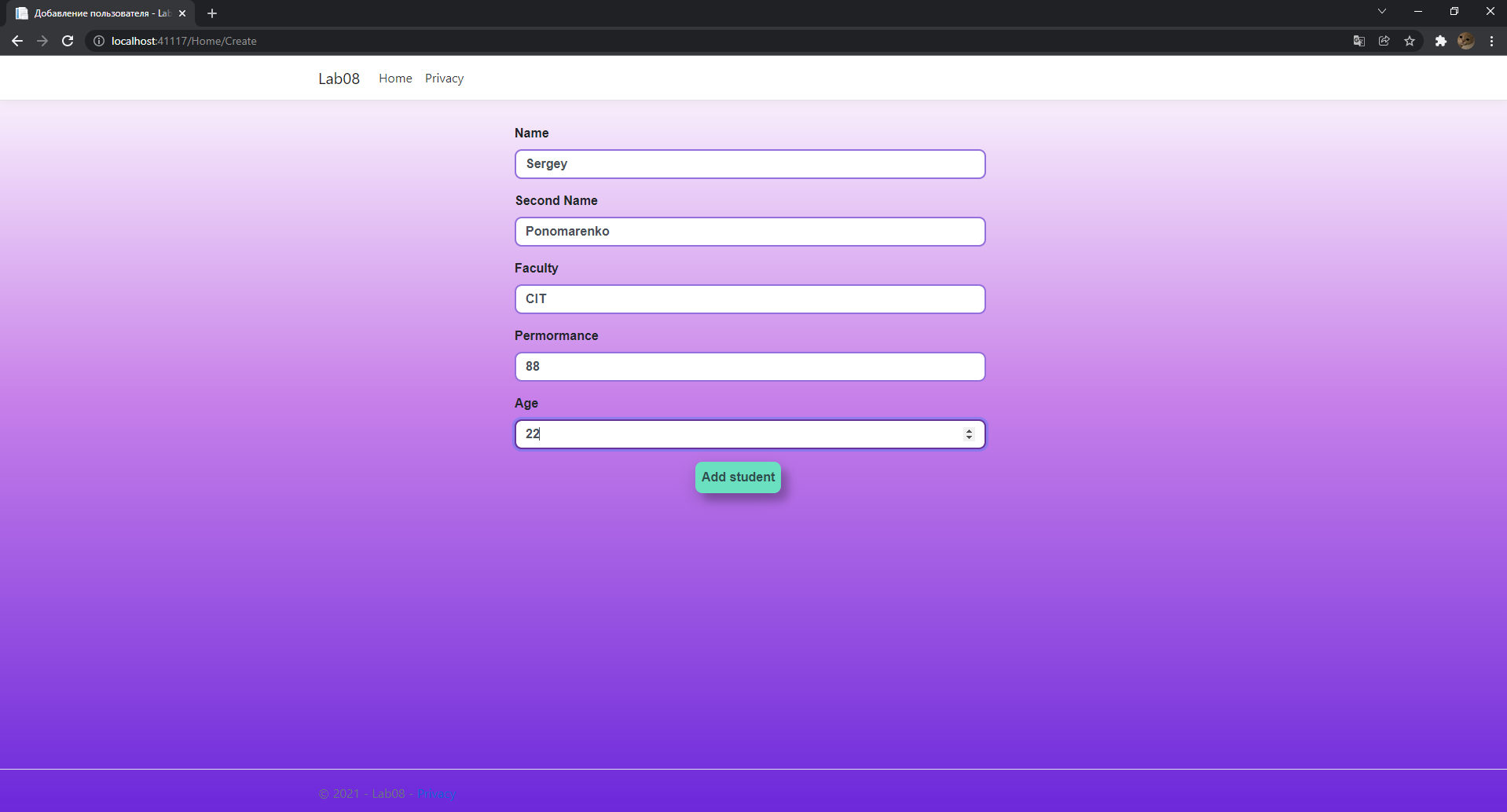


Рисунок 3 - Сторінка додавання студента

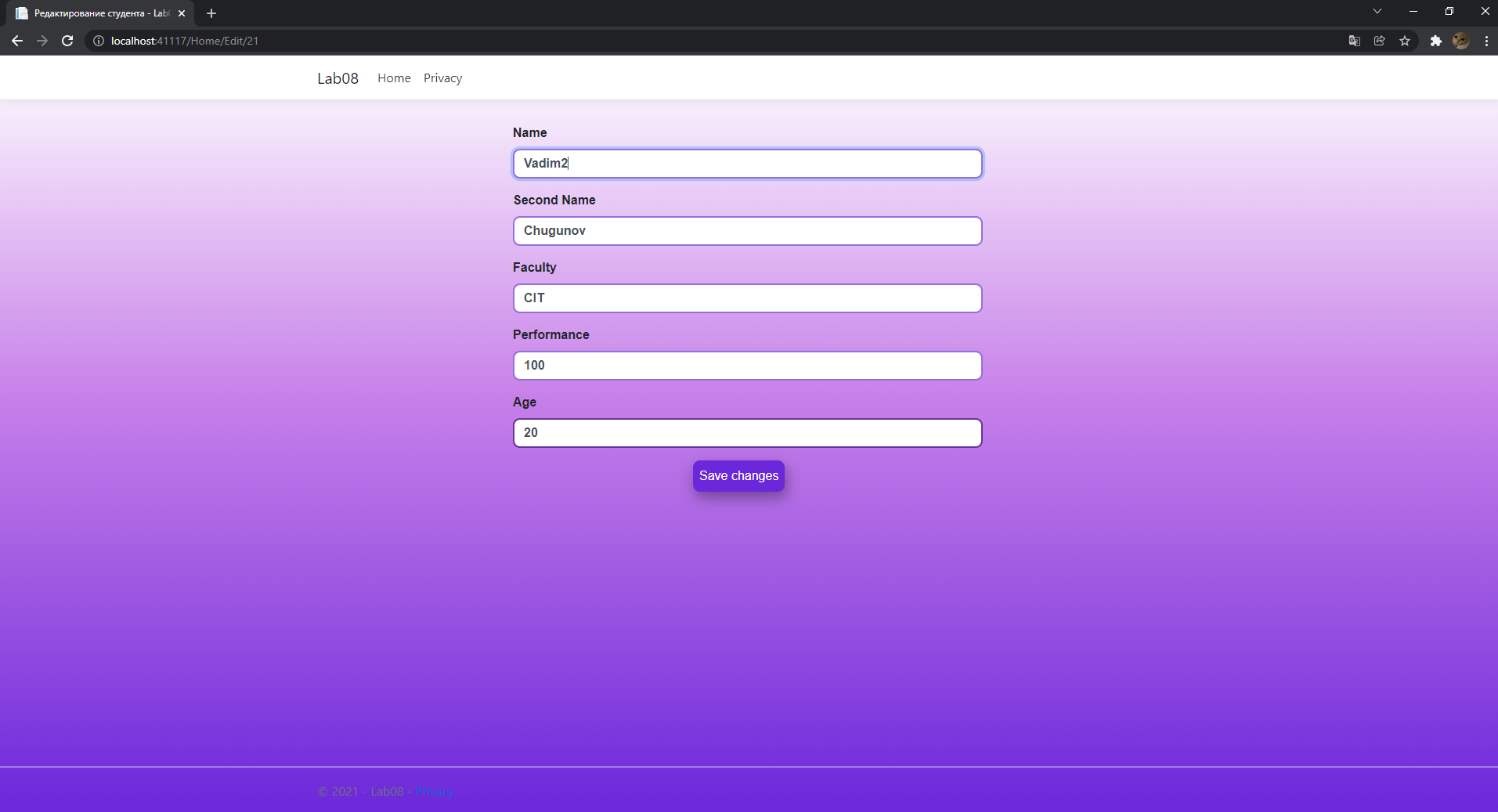


Рисунок 4 - Сторінка оновлення даних студента

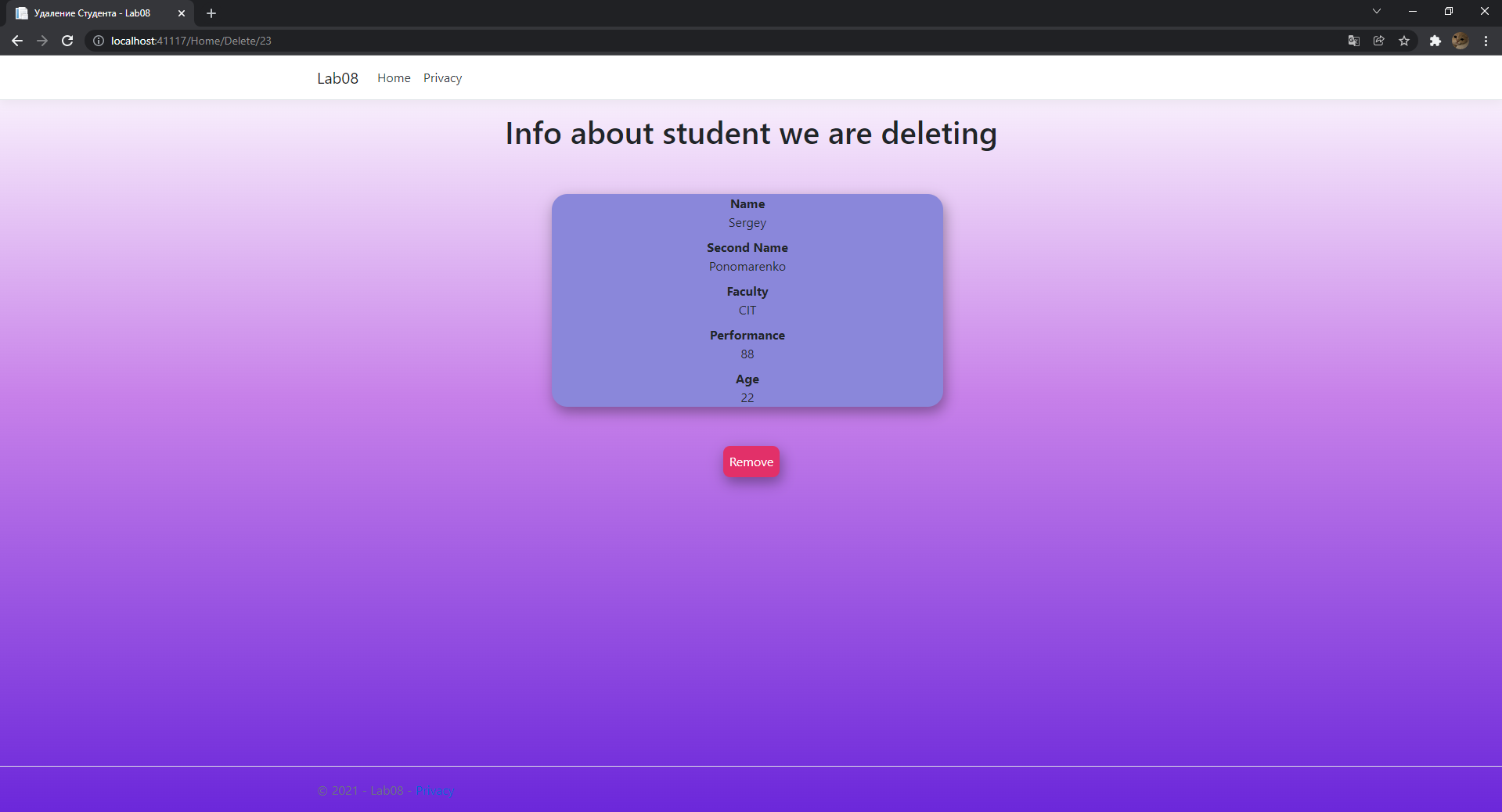


Рисунок 5 - Сторінка видалення студента

**ВИСНОВОК**

Під час виконання лабораторної роботи, набули практичні навички з створення веб-застосунка. Реалізовали додавання студенту у список, його видалення та редагування. Створили валідацію даних студента.