Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Институт информационных технологий

Факультет компьютерных технологий

Лабораторная работа № 9

Работа с приложениями ASP

Выполнил: студент гр. 881062 Литасов О.А.

Проверил: Потоцкий Д.С.

Минск 2020

## Задания

1.    Выполнить все примеры, приведенные в теоретической части и разобраться в неясных местах.

2.    Создать собственную базу данных в ACCESS.

3.    Разработать серверный ASP.NET-сценарий, который обращается к вашей базе данных (например, временя начала лекции или какая-то иная справочная информация (наличие или цена товара, характеристика человека и пр.)).

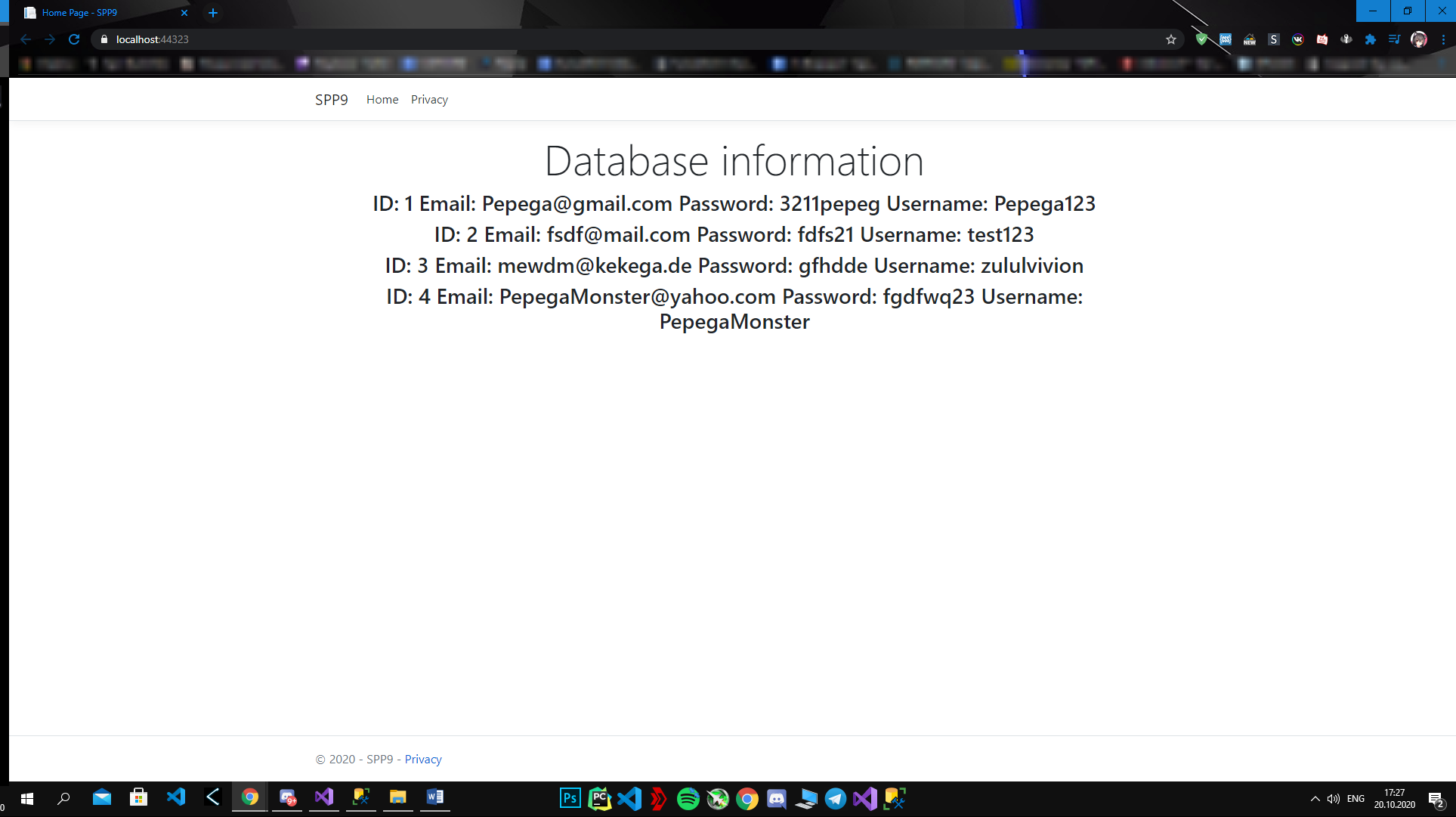
4.    В ASP-сценарии использовать собственный элемент пользователя в качестве интерфейса с базой данных.

5.    Содать клиентский сайт (средствами HTML) для доступа к ASP.NET-приложению.

6.     Подготовить отчет и защитить работу.

## Скриншот выполнения программы:

Вывод информации из базы данных на Html страницу



## Код программы:

Front

@{

ViewData["Title"] = "Home Page";

}

<div class="text-center">

<h1 class="display-4">Database information</h1>

@{

foreach (var inf in ViewBag.data)

{

<h3> ID: @inf.Id Email: @inf.Email Password: @inf.Password Username: @inf.Username </h3>

}

}

</div>

Back

using System.Collections.Generic;

using System.Diagnostics;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Logging;

using SPP9.Models;

namespace SPP9.Controllers

{

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

public HomeController(ILogger<HomeController> logger)

{

\_logger = logger;

}

public IActionResult Index()

{

//создание объекта базы данных

using (SPPLR2Context context = new SPPLR2Context())

{

//создание объекта юзеров

var users = context.Users;

List<Users> Us = new List<Users>();

//запись всех юзеров в лист

foreach (var u in users)

{

Us.Add(u);

}

ViewBag.data = Us;

//вызов страницы idex.html

return View();

}

}

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 });

}

}

}

Создание объекта базы данных

using System;

using Microsoft.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore.Metadata;

namespace SPP9.Models

{

public partial class SPPLR2Context : DbContext

{

public SPPLR2Context()

{

}

public SPPLR2Context(DbContextOptions<SPPLR2Context> options)

: base(options)

{

}

public virtual DbSet<Users> Users { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

if (!optionsBuilder.IsConfigured)

{

optionsBuilder.UseSqlServer("Server=.\\SQLExpress;Database=SPPLR2;Trusted\_Connection=True;");

}

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Users>(entity =>

{

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.Email)

.HasMaxLength(30)

.IsUnicode(false)

.IsFixedLength();

entity.Property(e => e.Password)

.HasColumnName("password")

.HasMaxLength(30)

.IsUnicode(false)

.IsFixedLength();

entity.Property(e => e.Username)

.HasMaxLength(30)

.IsUnicode(false)

.IsFixedLength();

});

OnModelCreatingPartial(modelBuilder);

}

partial void OnModelCreatingPartial(ModelBuilder modelBuilder);

}

}