МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

«Курский государственный университет»

Кафедра программного обеспечения и администрирования   
информационных систем

Отчёт

по лабораторной работе 1

«Изучение технологии .Net Core MVC FrameWork»

по дисциплине

**«**Программирование на C#»

Выполнила: студентка группы 413

Ершова Анастасия Сергеевна

Проверил: доцент

кафедры ПОиАИС   
Жиляев Д.Л.

Курск

2021

Цель работы: изучение основ разработки приложений с использованием технологии .Net Core MVC FrameWork.

Задание:

Реализовать страницу личного кабинета (аккаунта) пользователя.  
1. В верхнее меню добавить пункт меню Account  
2. При нажатии на пункт меню:  
Если пользователь не авторизован, открывается форма логина пользователя, содержащая поля для ввода логина и пароля и кнопку Войти. Адрес формы: /Home/Login. При нажатии на кнопку Войти выполняется авторизация пользователя и переход на страницу аккаунта пользователя, если логин и пароль есть в базе пользователей. Если логина и пароля нет в базе пользователей, выполняется возврат на страницу /Login, и на ней над полями ввода выводится соответствующее сообщение  
Если пользователь авторизован, выполняется переход на страницу аккаунта пользователя.  
3. База пользователей хранится в коде приложения в виде инициализированной коллекции:  
List<User> Users = new List<User>  
{  
new User {login = .., password = .., FullName = …, BIrthDate = ….}  
new User {login = .., password = .., FullName = …, BIrthDate = ….}  
}  
  
Пароли допускается хранить в незашифрованном виде  
4. Форма аккаунта пользователя содержит информацию о пользователе (полное имя и дата рождения) в виде текста.  
5. Реализовать пункт верхнего меню "Выйти", выполняющий выход пользователя из системы (снятие признака "авторизован" и затем переход на главную страницу сайта.  
Авторизованного пользователя допускается хранить во ViewBag.

Листинг программы:

HomeController:

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Logging;

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Threading.Tasks;

using WebApplication1.Models;

namespace WebApplication1.Controllers

{

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

private UserService \_userService;

public HomeController(ILogger<HomeController> logger)

{

\_logger = logger;

\_userService = new UserService();

}

[HttpGet]

public IActionResult Privacy()

{

ViewData["Title"] = TempData["CurrentData"];

return View();

}

[HttpPost]

public IActionResult Privacy(LoginModel model)

{

return View(model);

}

public IActionResult Index()

{

return View();

}

[HttpGet]

public IActionResult Account()

{

return View();

}

[HttpPost]

public IActionResult Account(LoginModel model)

{

model.Success = \_userService?.CheckCredentials(model.login, model.password);

if (model.Success.HasValue && model.Success.Value)

{

UserService userService = new UserService();

var CurrentUser1 = userService.Users.Where(x => x.login == model.login && x.password == model.password);

TempData["CurrentData"] = CurrentUser1.First().fullname + "\nДата рождения: " + CurrentUser1.First().birthday;

return RedirectToAction("Privacy", "Home");

}

return RedirectToAction("Error", "Home");

}

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

public IActionResult Error()

{

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

}

[HttpPost]

public IActionResult Index(LoginModel model)

{

return RedirectToAction("Index", "Home");

}

}

}

UserService:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace WebApplication1.Models

{

public class UserService

{

public List<User> Users = new List<User>

{

new User {login="nastya", password="nnn123", fullname="Anastasiya", birthday="01.01.2000"},

new User {login="alena", password="aaa123", fullname="Alena", birthday="02.02.2001"},

new User {login="tanya", password="ttt123", fullname="Tatyana", birthday="01.01.1999"}

};

public bool F = false;

public bool CheckCredentials(string login, string password)

{

bool F = Users.Any(u=>u.login==login && u.password==password);

return F;

}

}

}

Account:

@model LoginModel

<**form** action="/Home/Account" method="POST">

@Html.LabelFor(x => Model.login)

@Html.EditorFor(x => Model.login)

@Html.LabelFor(x => Model.password)

@Html.EditorFor(x => Model.password, new { type = "password" })<br/>

<input type="submit" value="Send"/>

</**form**>

@if (!Model?.Success ?? false)

{

<div>Имя пользователя или пароль неверны</div>

}

Privacy:

@{

}

<h1>@ViewData["Title"]</h1>

<p1> <input type="submit" value="выйти из аккаунта" > </p1>

<p>Use this page to detail your site's privacy policy.</p>

Index:

@{

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

}

<div class="text-center">

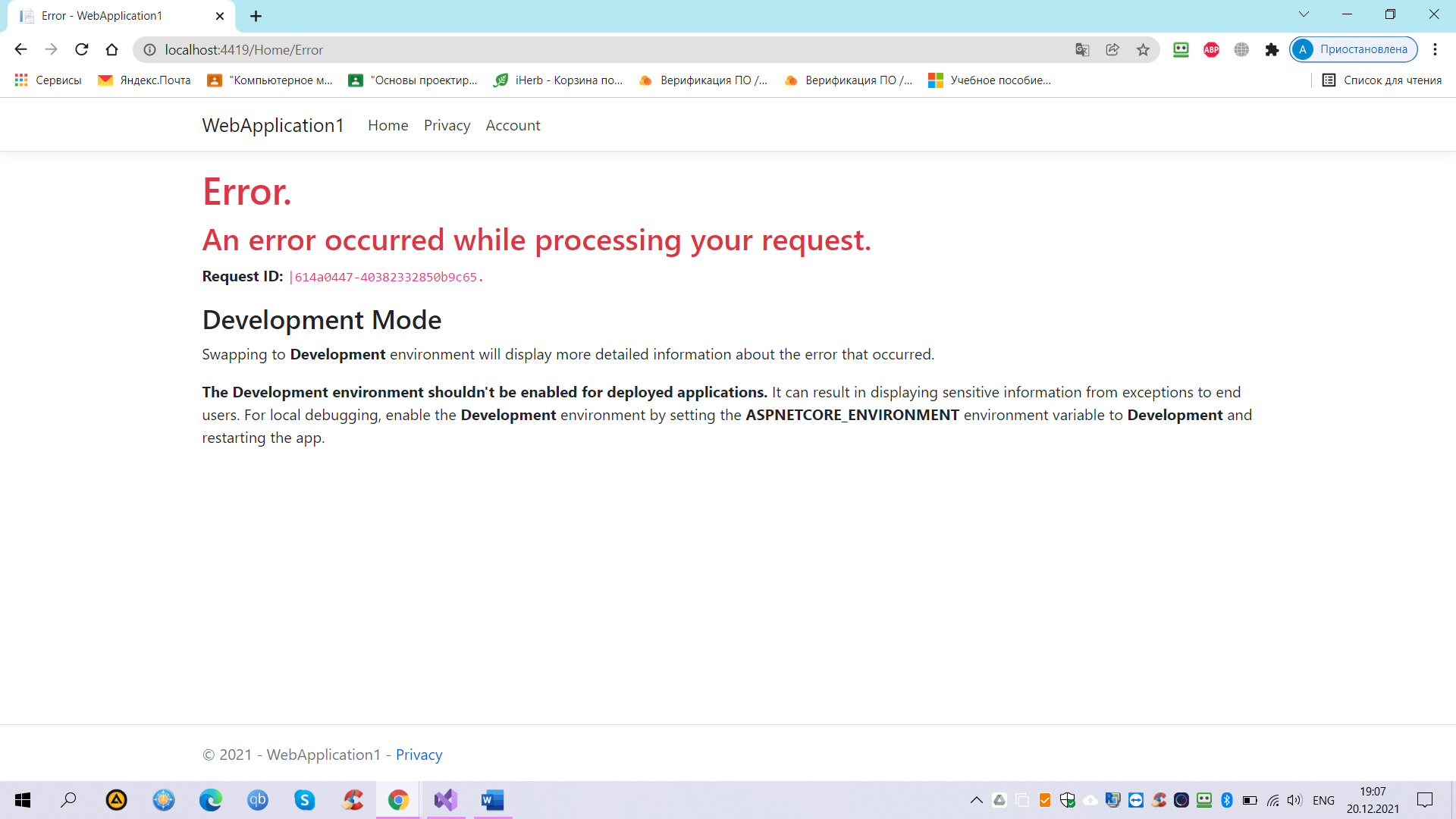
<h1 class="display-4">Welcome</h1>

<p>Learn about <a href="https://docs.microsoft.com/aspnet/core">building Web apps with ASP.NET Core</a>.</p>

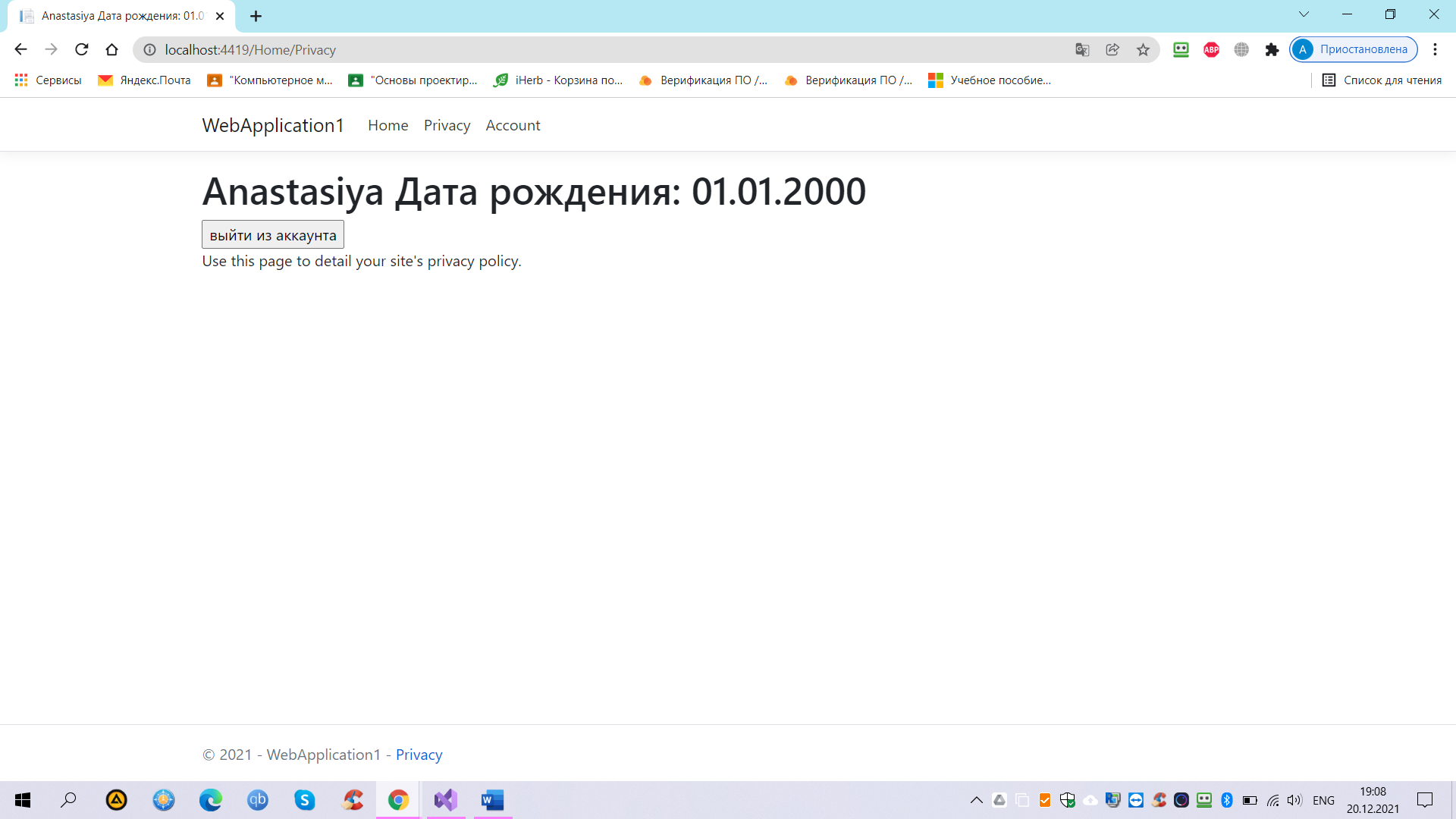
</div>

Тестирование:

Результат тестирования при вводе неверных данных:



Результат тестирования при вводе верных данных:



Вывод: изучила основы разработки приложений с использованием технологии .Net Core MVC FrameWork.