**Отчёт по лабораторной работе №5**

**Фесько Ксения Александровна**

**Группа 1720**

Структура проекта довольно большая.

Создала простое приложение с регистрацией и авторизацией студентов.

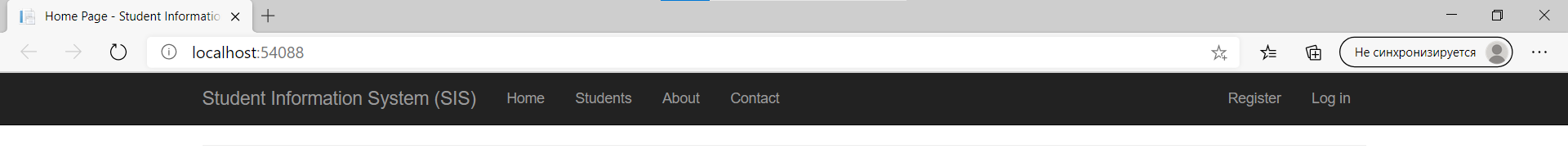
Также есть возможность удаления студента из базы данных, редактирование данных и просмотр информации о студенте.

В форме студент прописано только одно поле, по желанию можно прописать больше, я не успела, для проверки прописала, чтобы посмотреть работает ли.

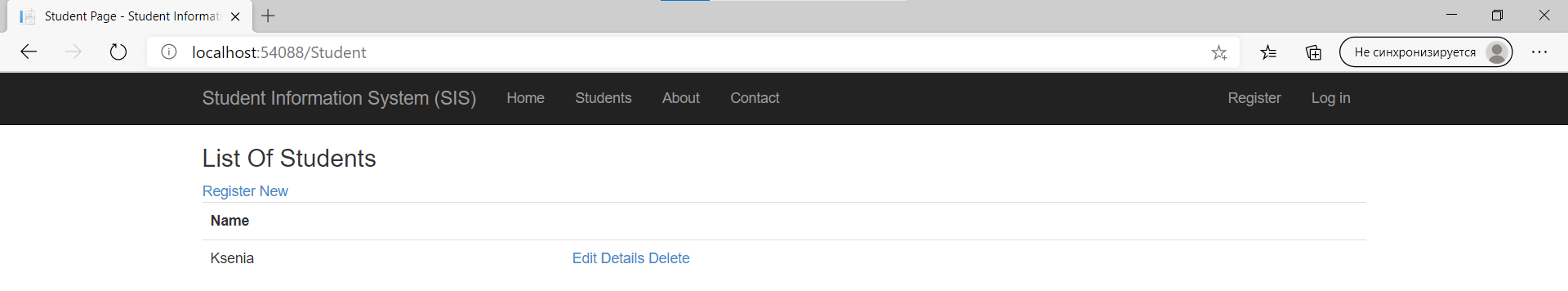
В лабораторной реализовано подключение к базе данных.

Результаты работы будут показаны в виде скриншотов.

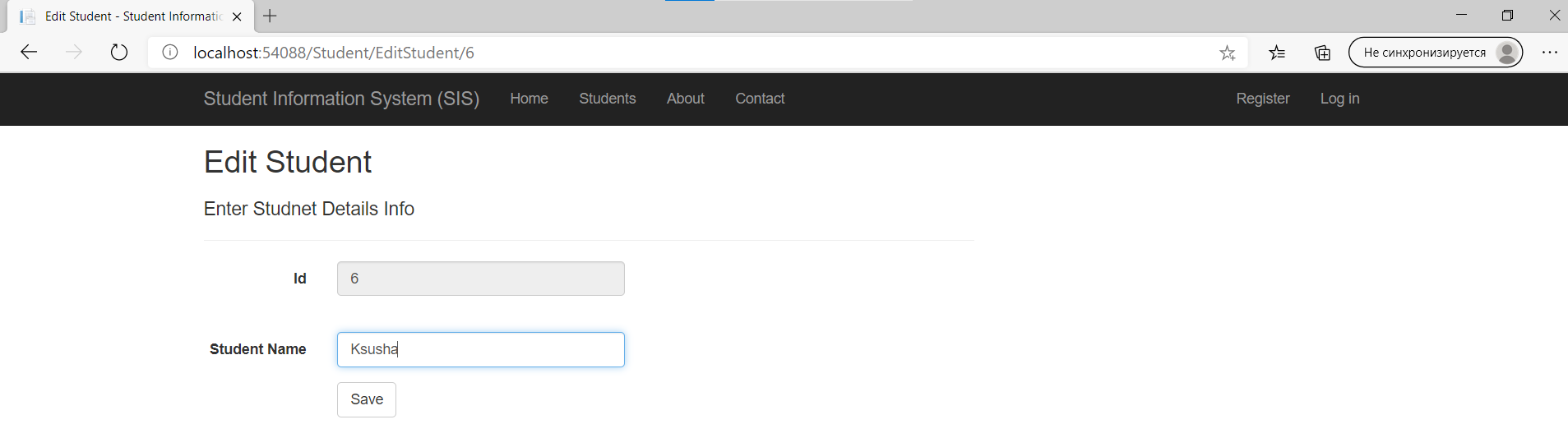
Обычная шапка профиля.



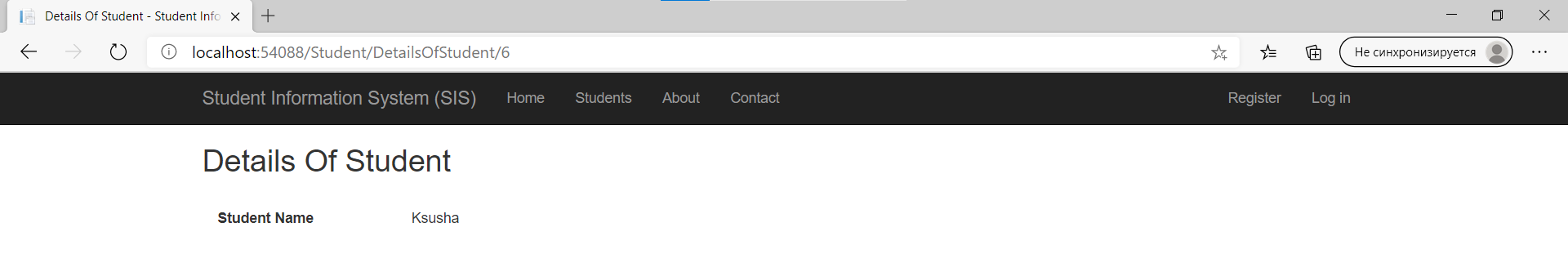
При нажатии на кнопку «Students» отображается лист студентов, в котором можно редактировать, удалять и просматривать данные о студенте.



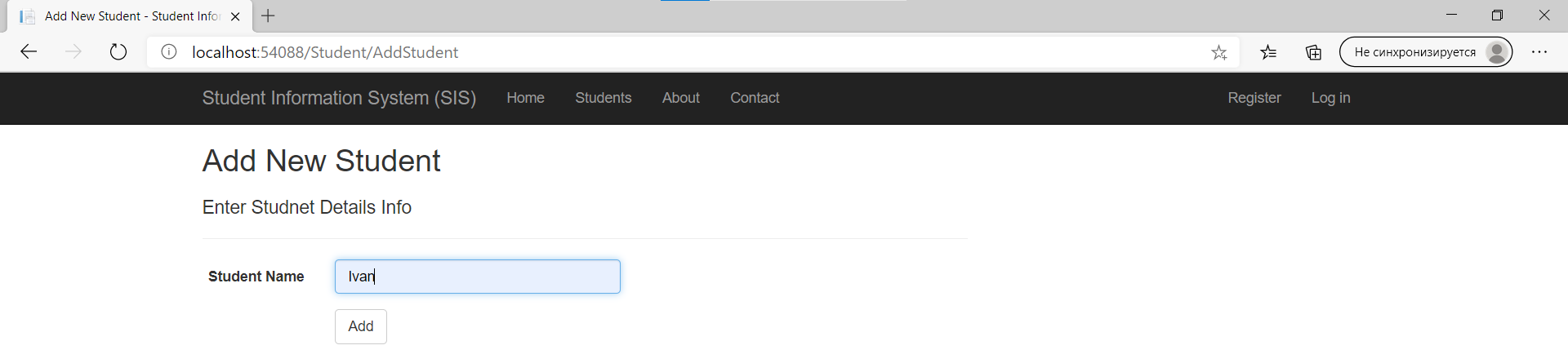
Окно редактирования.

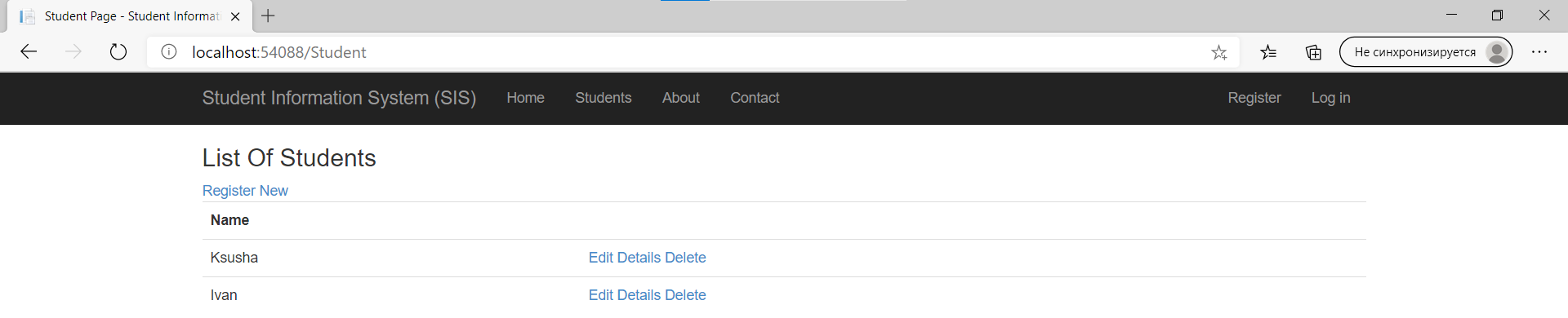


Описание, как я уже писала, можно прописать больше полей.

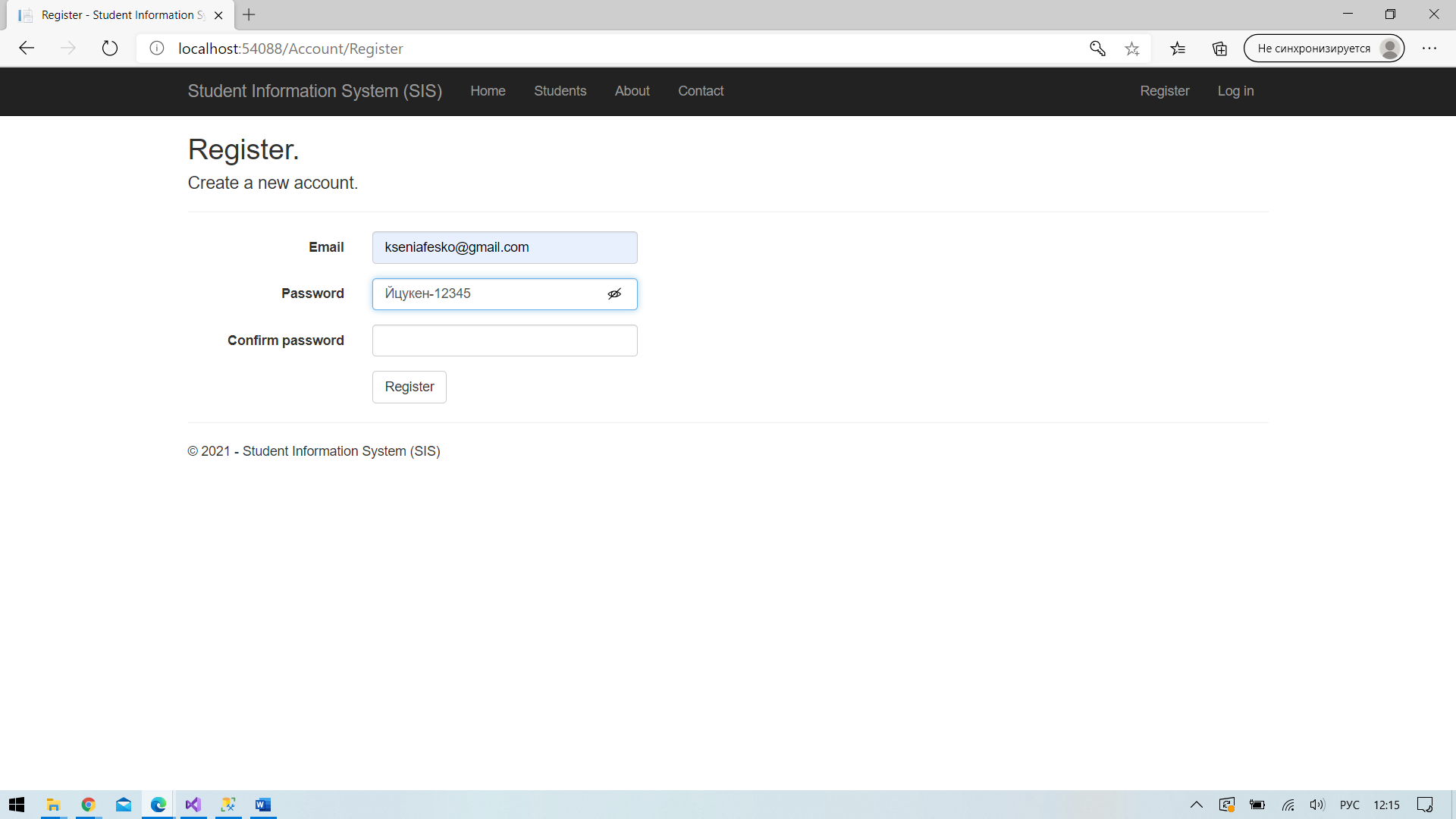


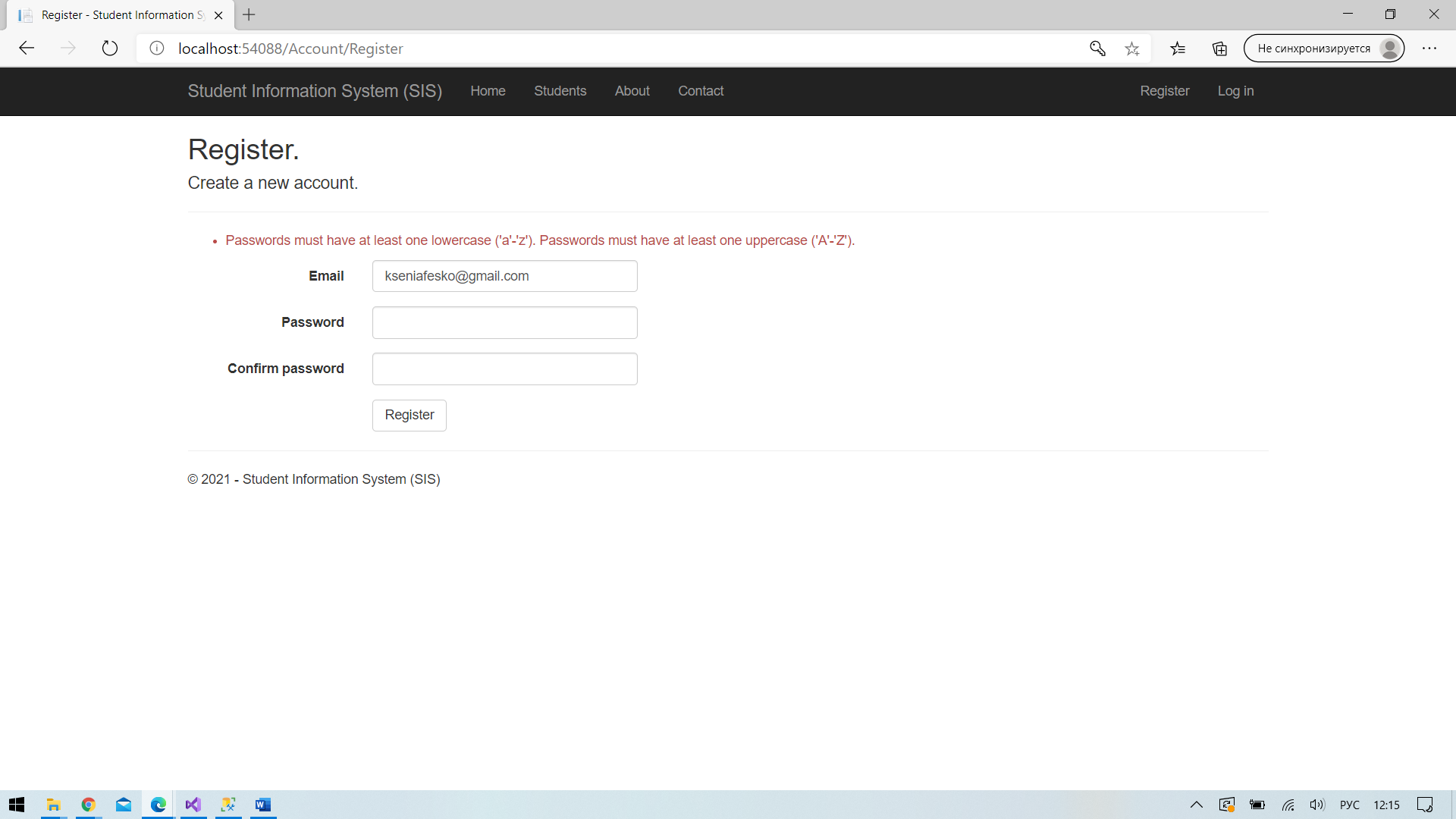
Добавление нового студента в список



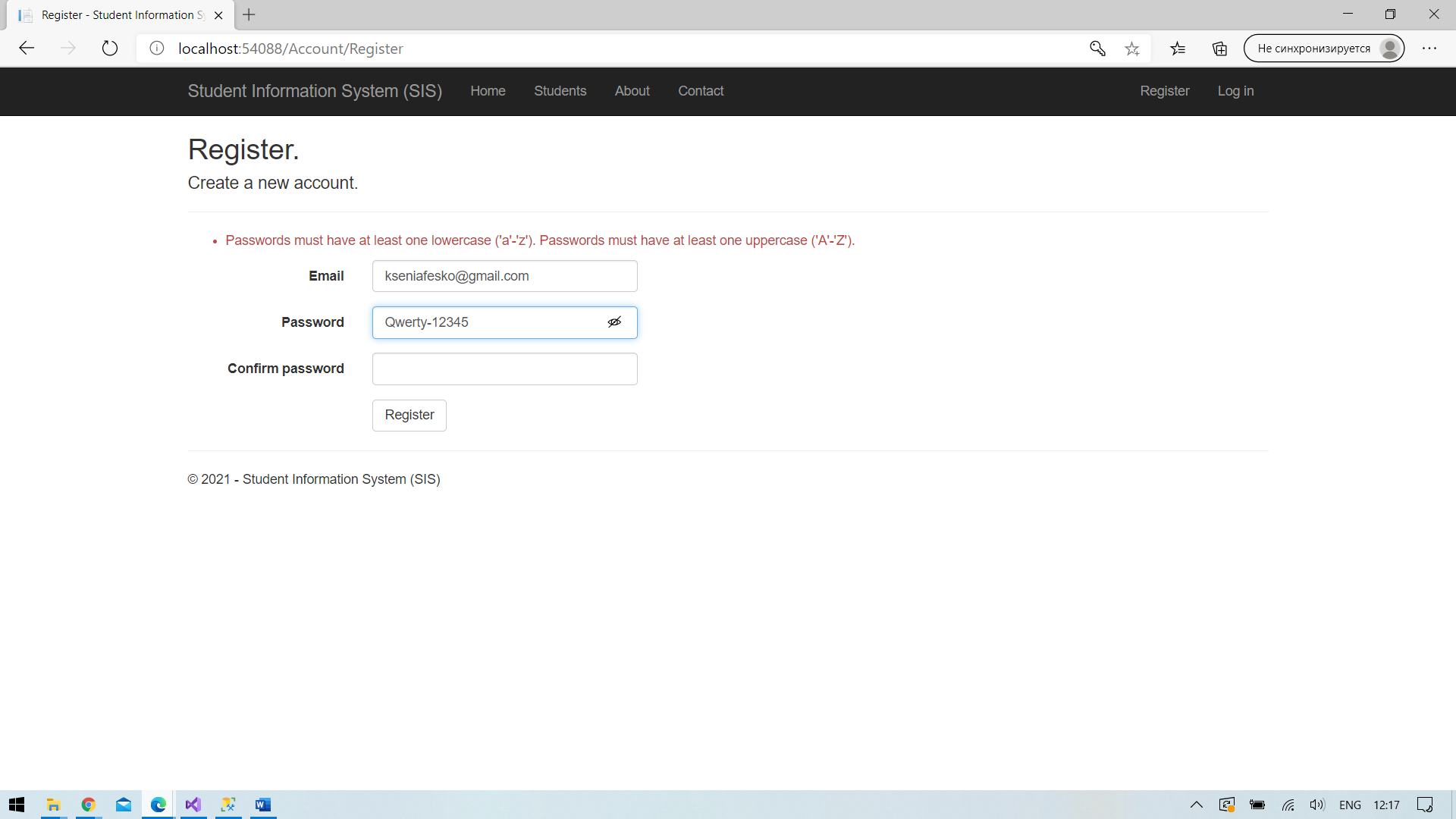


Нажав на кнопку «Register», открывается окно регистрации. В работе осуществлена валидация. Необходимо вводить пароль соответствующий требованиям.

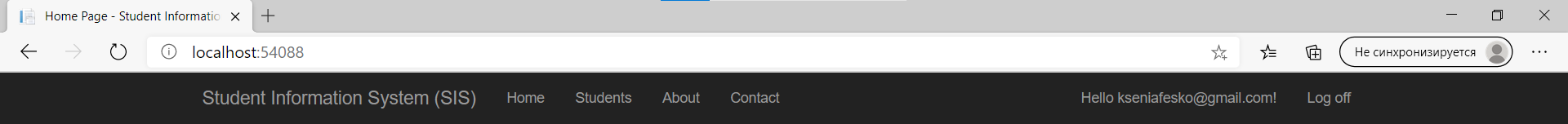




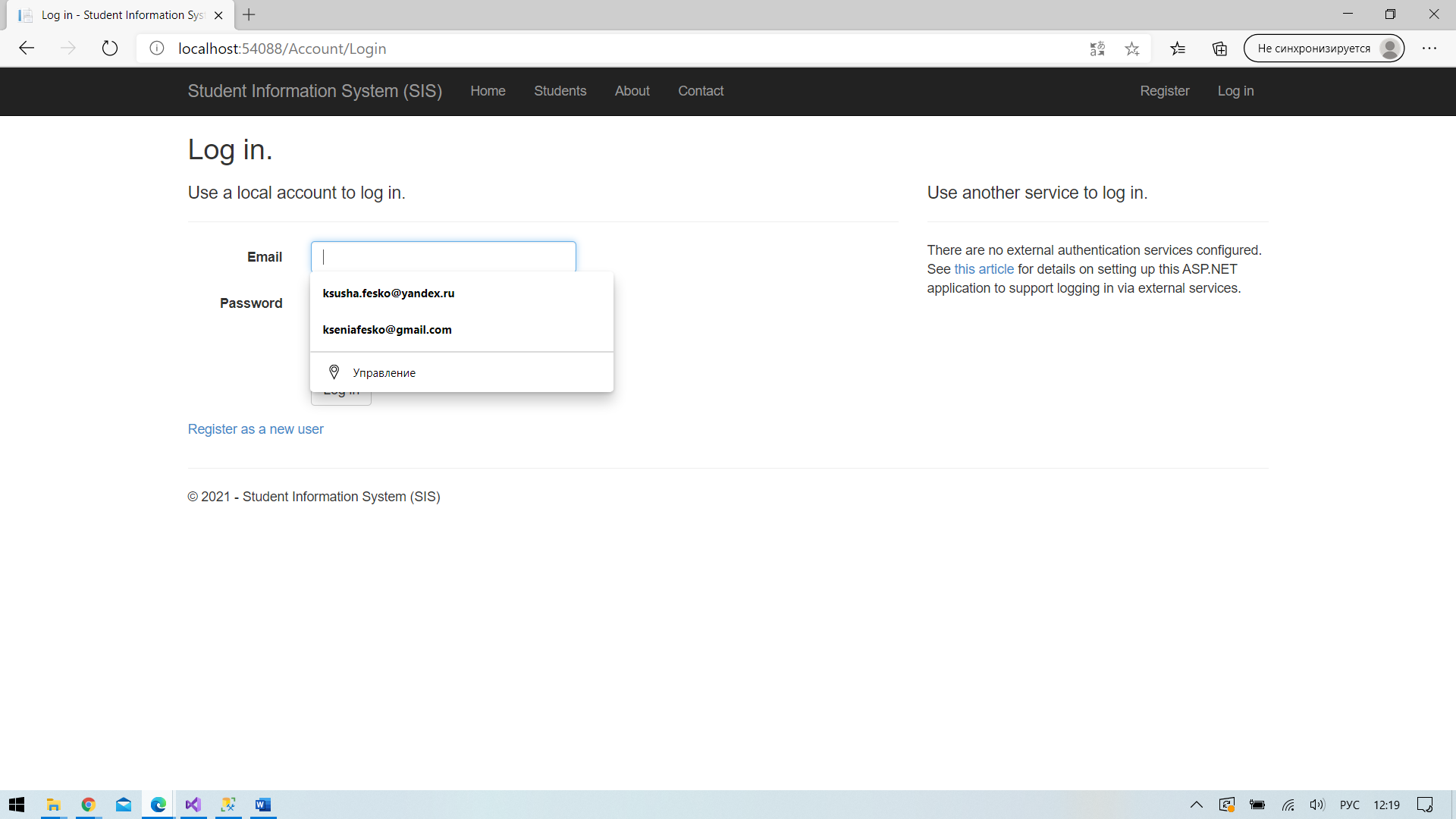
Пример хорошего пароля.



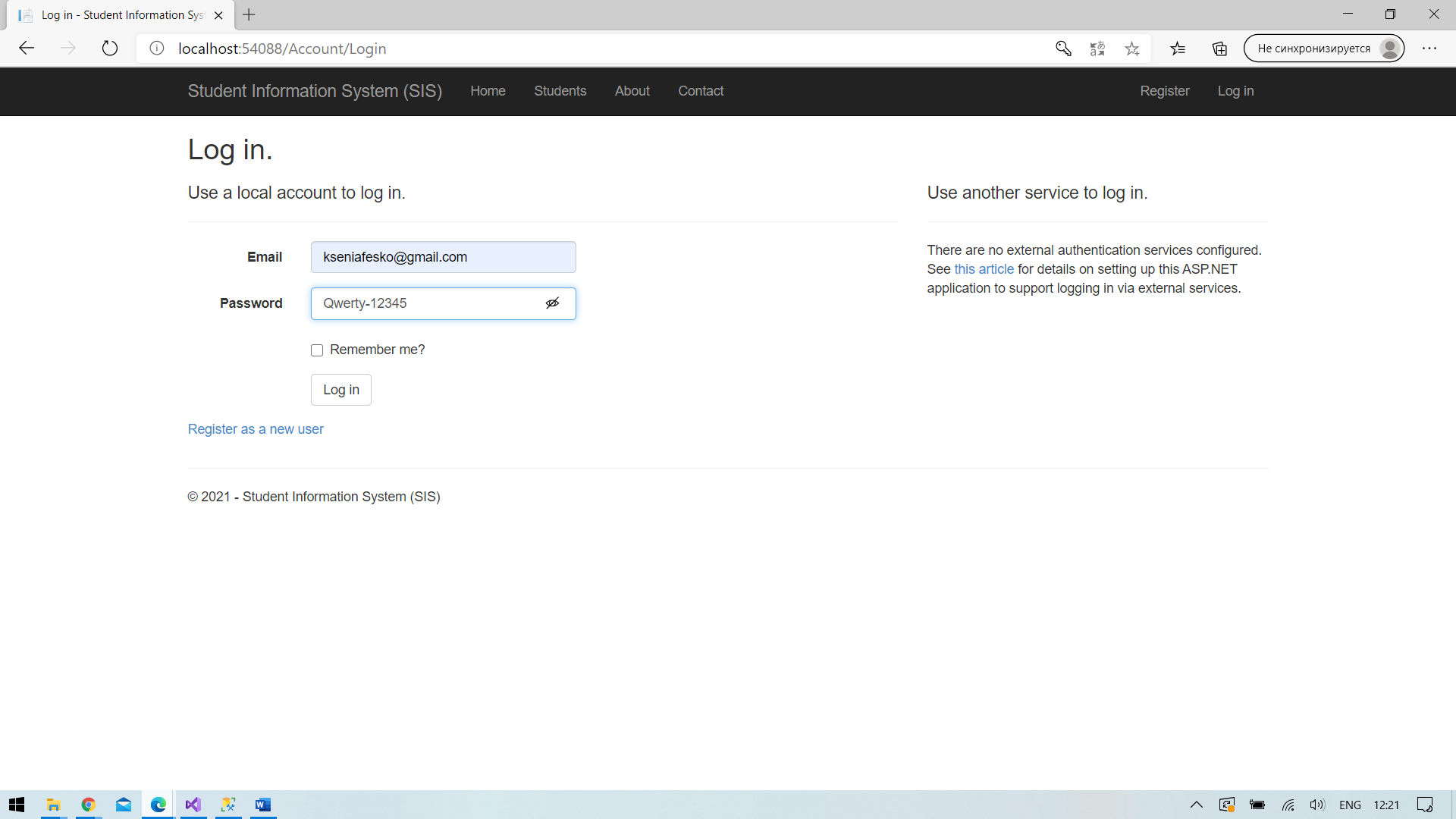
При успешной регистрации в шапке сайта будет надпись с вашим e-mail.



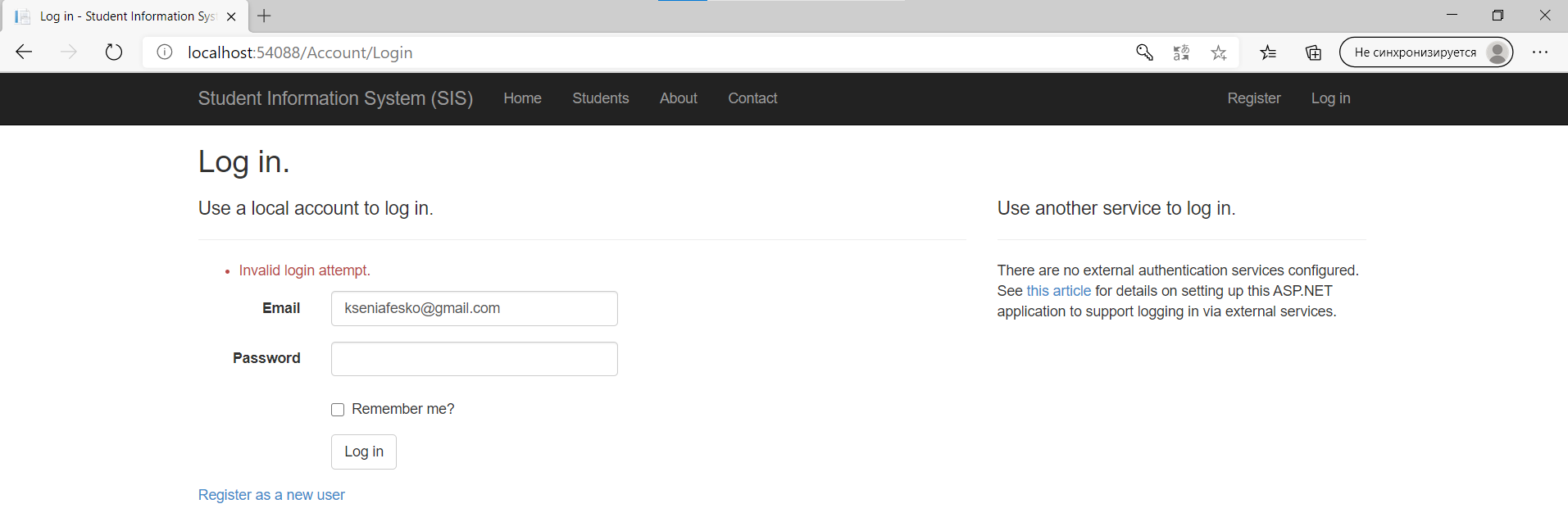
При выходе из учётной записи («Log off»), нажав далее на кнопку «Log in», пользователя переносит на страницу авторизации.



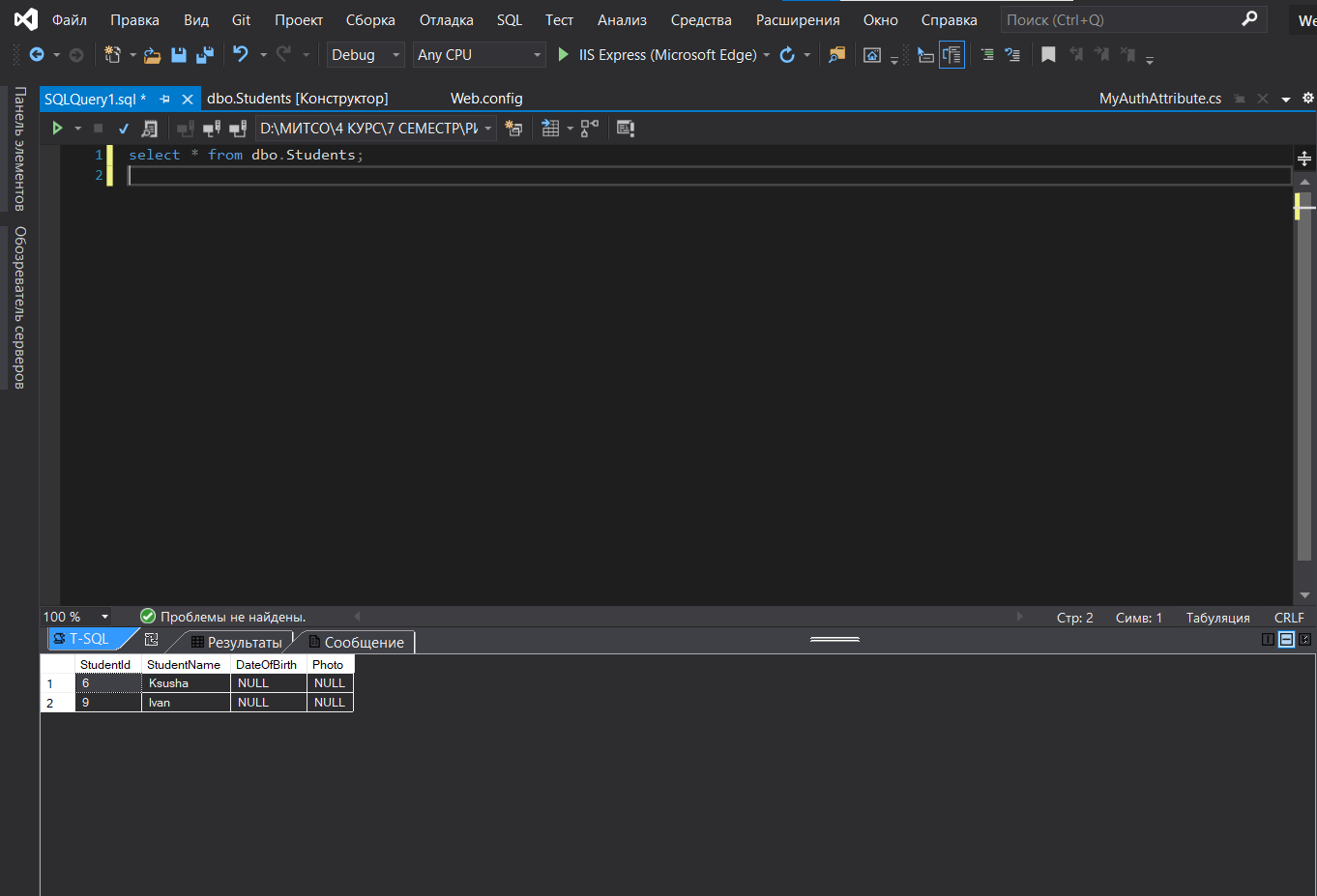
Далее выбираем логин, пишем пароль и входим в систему.



При неправильном введении данных выводится ошибка.



При написании запроса, в базе данных выводится информация, которая заносилась в форме.



**Вопросы для самопроверки**

1. **Что такое платформа ASP.NET?**

Платформа ASP.NET MVC представляет собой фреймворк для создания

сайтов и веб-приложений с помощью реализации паттерна MVC.

1. **Какие функции выполняет платформа ASP.NET?**

Она позволяет создавать динамические страницы HTML и многое другое.

1. **Как происходит компиляция кода в ASP.NET?**

<https://intuit.ru/studies/courses/4455/712/lecture/21296>

1. **Что такое web-форма?**

Web-формы в ASP.NET — это основной элемент динамического Web-сайта. Web-формы определяются как программируемые Web-страницы, которые служат пользовательским интерфейсом в приложениях ASP.NET. Web-форма — это комбинация кода HTML, программного кода и элементов управления. Web-формы в приложении ASP.NET представлены файлами с расширением .aspx.

1. **Для чего предназначены web-формы и какие функции они выполняют?**

Web-формы определяются как программируемые Web-страницы, которые служат пользовательским интерфейсом в приложениях ASP.NET.

1. **Какие основные элементы управления используются на web-формах?**

Главное специфическое для ASP.NET содержание Web-формы — это специальные теги с атрибутами. На форме может быть три типа тегов с атрибутами:

* страничные атрибуты, (@ Page), которые определяют общие параметры для всей Web-формы;
* атрибуты тела Web-формы (body), которые определяют общие параметры отображения содержимого страницы;
* атрибуты форм (form), которые определяют параметры работы с формой — набором элементов управления (не путать с Web-формой).

1. **Какую структуру имеет ASPX-файл?**

Файлы ASPX создаются веб-сервером и содержат сценарии и исходные коды, которые помогают сообщать браузеру, как следует открывать и отображать веб-страницу.

Такие файлы (также известные под названием "веб-формы .NET" могут содержать VBScript-код, написанный на C#, а также XHTML-разметку. Этот веб-формат поддерживает, кроме того, динамический код. Сценарии и команды, содержащиеся в файле, обрабатываются на веб-сервере, который отправляет содержимое в браузер пользователя в нужном формате (HTML).

Рассмотрим пример простой aspx-страницы.

<%-- 1. Директива страницы --%>

<%@ Page Language="C#" AutoEventWireup="true"

CodeBehind="Default.aspx.cs" Inherits="WebApp.\_Default" %>

<%-- 4. Блок серверного кода --%>

<script runat="server">

protected void btnSend\_Click(object sender, EventArgs e)

{

tbxName.Text = tbxName.Text.ToUpper();

}

</script>

<%-- 3. Обычный HTML --%>

<html>

<body>

<%-- 5. Блок рендерного кода --%>

<% for (int i = 0; i < 10; i++) Response.Write("Hello"); %>

<form id="form1" runat="server">

<div>

Input name:

<%-- 2. Серверные элементы управления --%>

<asp:TextBox ID="tbxName" runat="server" />

<asp:Button ID="btnSend" runat="server"

OnClick="btnSend\_Click" />

</div>

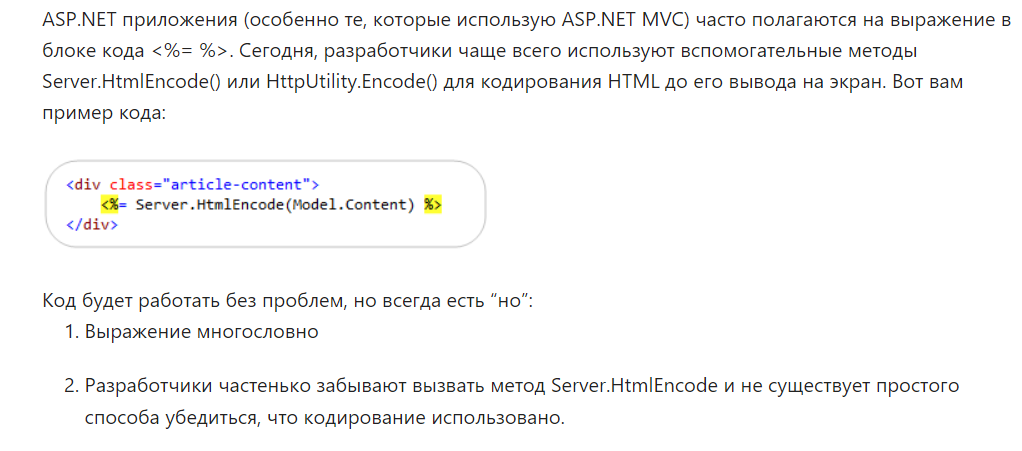
</form>

</body>

+</html>

1. **Для чего предназначена конструкция <%= %>?**

Для HTML-кодирования.



**9. На каких языках программирования возможно написание приложений**

**для ASP.NET-платформы?**

Разработчики могут писать код для ASP.NET, используя языки программирования, входящие в комплект .NET Framework (C#, Visual Basic.NET, J# и JScript .NET).

**10. Назовите основные файлы, содержащиеся в создаваемом проекте**

**ASP.NET?**

|  |  |  |
| --- | --- | --- |
| Папка | Описание | Примечание |
| /App\_Data | Предназначена для хранения закрытых данных: документов XML, баз данных | Здесь могут храниться только те базы данных, где структурные компоненты хранятся в виде файлов. Не файлы здесь храниться не могут. |
| /App\_Start | Предназначена для хранения конфигураций маршрутов, фильтров, и пакетов содержимого. | Маршруты – это ассоциации URL с файлами или кодом, который хранится на сервере.  Фильтры – программная логика, выполняемая до вызова действий контроллера.  Пакеты содержимого – это встраиваемые компоненты контента с помощью менеджера пакетов.  Маршруты, фильтры и пакеты рассматриваются подробнее в соответствующих разделах. |
| /bin | Папка исполняемых файлов. Сюда помещается скомпилированная сборка приложения вместе с теми, на которые ссылается. | В Обозревателе решений не отображается. Чтобы отобразилась нужно щелкнуть на кнопке Show all files (Показать все файлы) |
| /Areas | Папка для областей. | Область позволяет разделить проект на части. Полезно при больших проектах. |
| /Content | Папка для хранения статических данных: изображений, CSS – файлов. |  |
| /Scripts | Папка для хранения скриптов. | Скрипты – библиотеки Javasript, например, JQuery |
| /Models | Классов, типов предметной области. | В больших проектах программисты эту папку удаляют, помещая всю модель предметной области в отдельный проект. |
| /Controllers | Папка для хранения контроллеров |  |
| /Views | Папка для хранения представлений | Представления группируются по контроллерам, поскольку ими создаются. /Views/Home |
| /Views/Shared | Папка для хранения представлений не специфичных для определенного контроллера. | Обычно здесь хранятся компоновки, другое название мастер-страницы. |
|  | | |
| Файлы | Описание | Примечание |
| /Views/Web.config | Файл конфигураций. Нужен чтобы запретить ISS серверу обрабатывать представления. | Все представления должны визуализироваться через методы-действия. Все представления, попавшие в этот файл, игнорируются ISS сервером. |
| /Global.asax | Глобальный класс приложения здесь регистрируются все компоненты приложения и код, который должен быть выполнен при запуске. |  |
| /Web.config | Конфигурационный файл.  Служит для задания специфичных конфигураций разным частям проекта. |  |