

## Практическая работа №16

### Технология Razor Pages

**Цель работы:** получение навыков написания клиентского веб-приложения с использованием ASP.NET Razor Pages.

#### Теоретическая часть

Для выполнения задания вам потребуется СУБД (MS SQL Server или другая реляционная).

#### Шаг 1

Запустите скрипт для создания БД.

```
use master
go
create database Students
go
use Students
go
create table Student (
  Id int identity(0,1),
  Name varchar(100) not null,
  Birthdate date not null,
  Phone varchar(20) not null,
  constraint PK_Student primary key(Id))
```

#### Шаг 2

Создайте проект типа ASP.NET Core Web App с именем StudentsRazorPages.

Обратите внимание, что в данном примере рассматривается версия .NET 7. Для более ранних версий некоторые шаги могут отличаться.

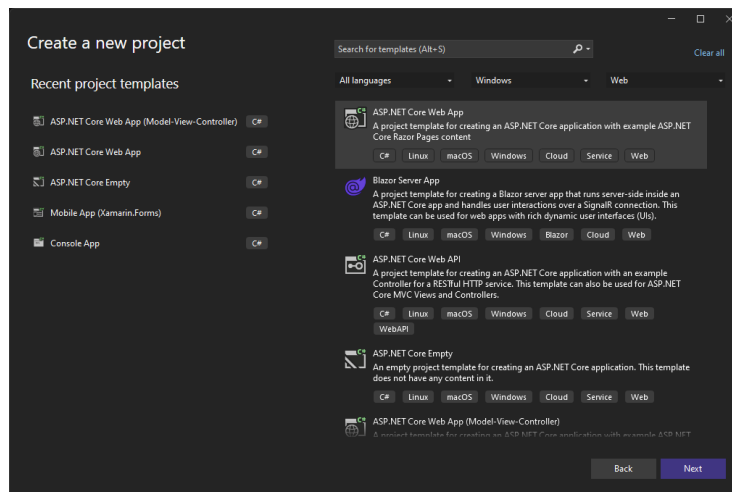


Рисунок 1 – Выбор типа проекта

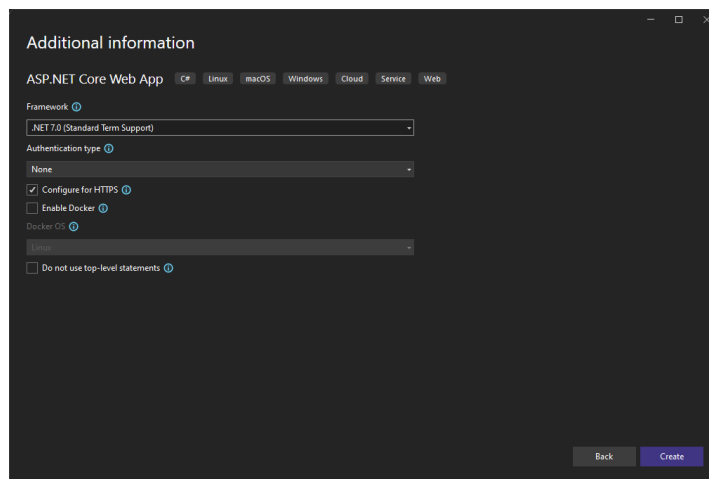


Рисунок 2 – Выбор версии фреймворка

Изучите содержимое проекта.

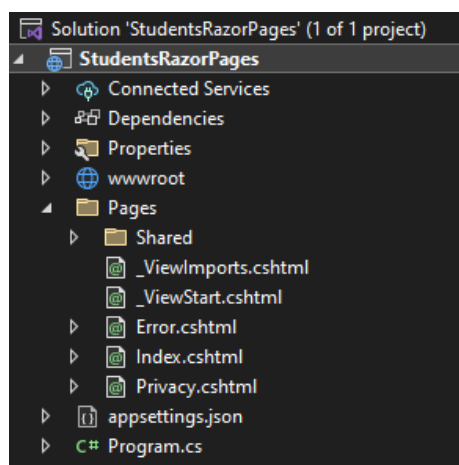


Рисунок 3 – Созданный проект

Папка Pages содержит страницы Razor и вспомогательные файлы. Каждая страница Razor — это пара файлов:

- Файл .cshtml с разметкой HTML и кодом C# использует синтаксис Razor.
- Файл .cshtml.cs с кодом C#, который обрабатывает события страницы.

### Шаг 3

Добавьте в проект папку Models и создайте в папке Models класс Student с кодом:

```
using System.ComponentModel.DataAnnotations;

namespace StudentsRazorPages.Models
{
    public class Student
    {
        [Key]
        public int Id { get; set; }

        [Required]
        [StringLength(100)]
        public string Name { get; set; }

        [Required]
        [DataType(DataType.Date)]
        public DateTime Birthdate { get; set; }

        [Required]
        [StringLength(20)]
        public string Phone { get; set; }
    }
}
```

### Шаг 4

Создайте папку Pages/Students.

Добавьте в проект Razor Pages для выполнения операций со студентами (Pages/Students – Add – New Scaffolded Item – Razor Pages using Entity Framework (CRUD)).

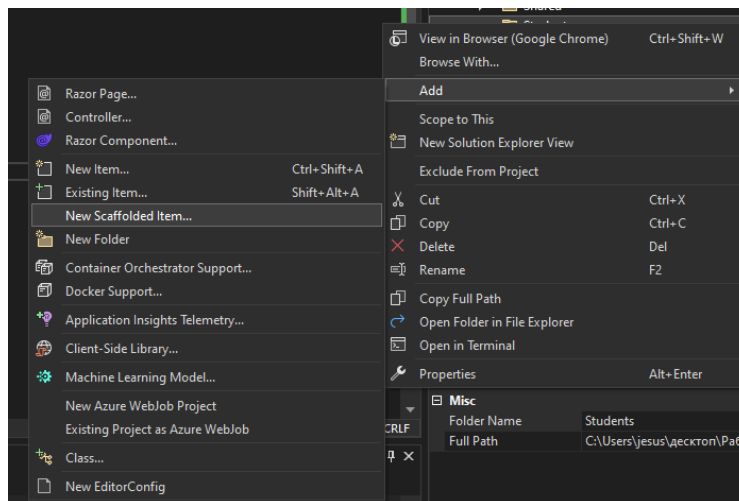


Рисунок 3 – Добавление элемента

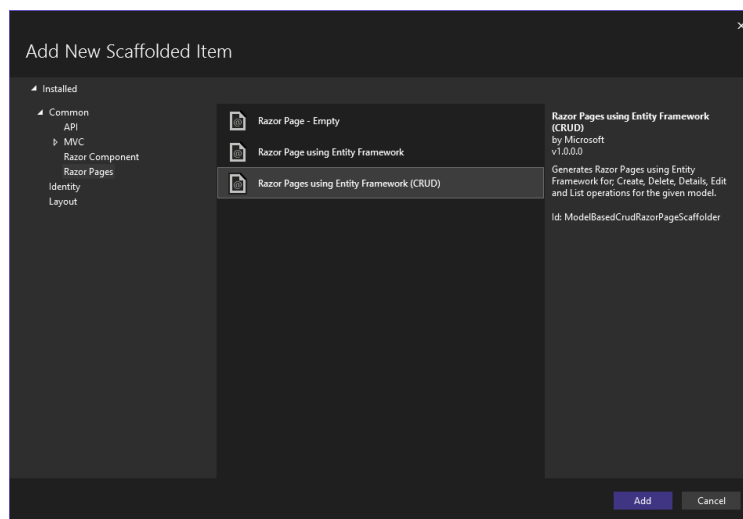


Рисунок 4 – Выбор типа контроллера

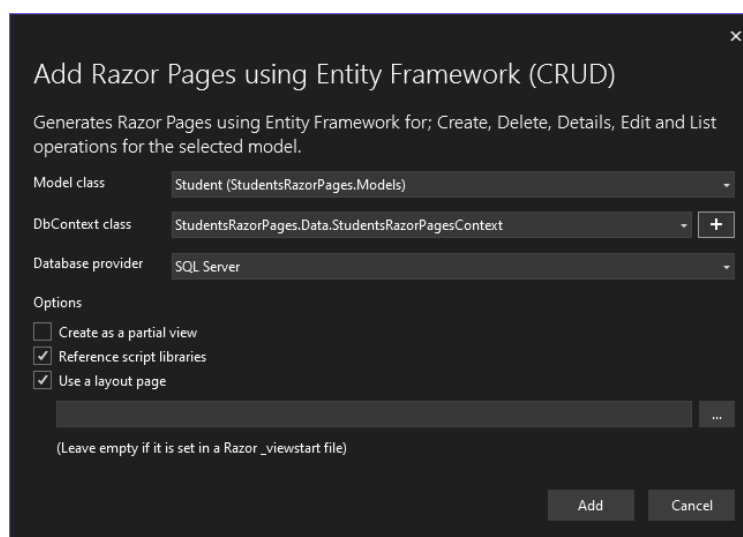


Рисунок 5 – Выбор модели для контроллера

В результате в проекте появятся страницы для операций CRUD с классом модели. Изучите новые файлы в проекте.

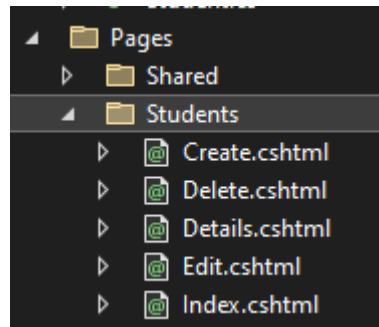


Рисунок 6 – Созданные файлы

Более подробно прочитать про созданные страницы можно в [руководстве](#).

## Шаг 5

В файле appsettings.json измените строку подключения к БД.

"StudentsRazorPagesContext":

"Server=имяПК\\SQLEXPRESS;Database=Students;TrustServerCertificate=True;Trusted\_Connection=True;MultipleActiveResultSets=true"

Запустите приложение.

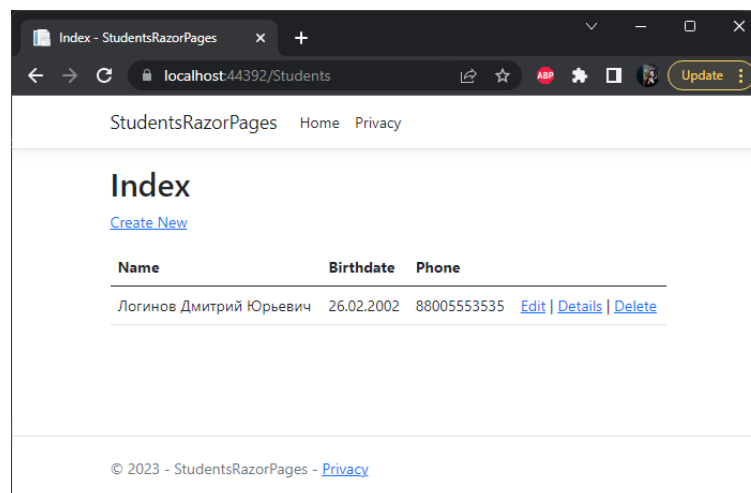


Рисунок 7 – Students.Index

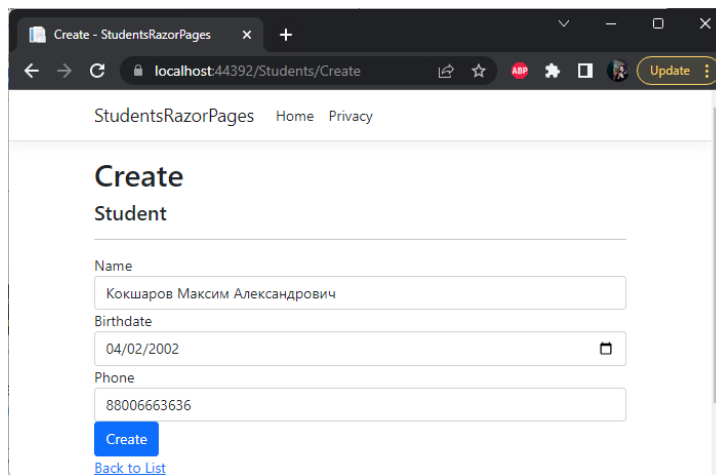


Рисунок 8 – Добавление нового студента

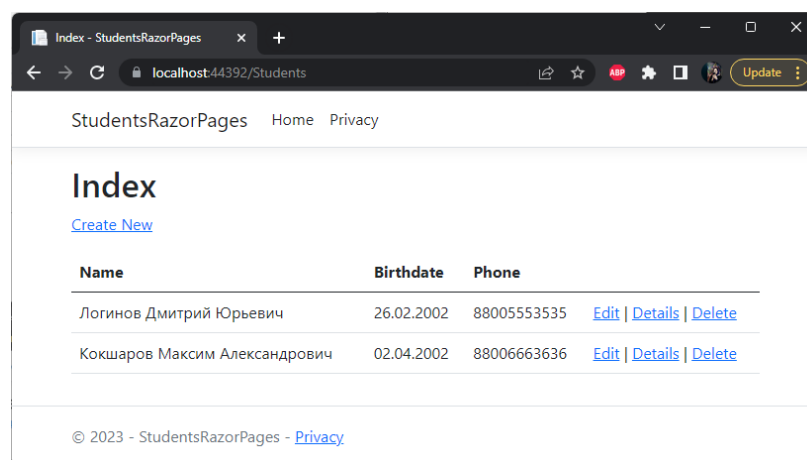


Рисунок 9 – Новая запись

## Задание

Доработайте приложение в соответствии со следующими требованиями:

- 1) Необходимо убрать ненужные веб-страницы. При запуске приложения должна открываться страница со списком студентов (Students.Index).
- 2) Добавьте возможность поиска студента по ФИО.
- 3) Добавьте проверку номера телефона на соответствие шаблону +7(xxx)xxx-xx-xx
- 4) Добавьте в БД таблицу Journal, в которой будет храниться информация о посещаемости студентов (студент, дата, название предмета, присутствие).

- 5) Внесите изменения в приложение для работы с новой таблицей.
- 6) Используйте [руководство](#) (выберите нужную версию фреймворка)