Лабораторная работа 3. Разработка проекта ИС средствами технологии объектного проектирования информационных систем ASP.NET Core MVC

Часовских В.П., сайт http://vikchas.ru

Разработка проекта ИС основывается на технологии объектного проектирования ASP.NET Core MVC.

В процессе выполнения работы Вы научитесь:

- > Создавать веб-приложение.
- > Добавление модели и формирование шаблона.
- > Работать с базой данных.
- > Добавление поиска и проверки.

Необходимая среда для выполнения работы

- Visual Studio 2019 16.4 или более поздней версии с рабочей нагрузкой ASP.NET и разработка веб-приложений
- > Пакет SDK для .NET Core 3.1 или более поздней версии

Необходимо запустить Visual Studio 2019 и последовательно выполнить последующие действия.

Be6-приложение ASP.NET Core (модель-представление-ко	нтроллер)	C# Linux	macOS	Windows	Облако	Служба	Beő		
олевая платформа									
NET Core 3.1 (долгосрочная поддержка)									
проверки подлинности									
er	-								
Настроить для HTTPS									
Включить Docker									
ерационная система Docker									
Vindows Волечно компилицию в среде выполнения Razor									
скиз жонить компилицию в сроде выполнения Razor									
ices жючить компилицию в среде выполнения Razor									
сота жочнък компилиција в сроде выполнения Вахог									
ndroes Включить комплияцию в среде выполнения Razor									
ndows									

Настроить новый проект

Веб-приложение ASP.NET Core (модель-представление-контроллер) ся Linux mecos Windows Обиеко Служба Веб

Имя проекта	
Kafedra02	
Расположение	
C:\Users\sergei\Desktop\CORE3\Kaфegpa	
Имя решения 🕕	
 Поместить решение и проект в одном каталоге 	

Назад Далее

D

															-	٥	>
Создание проекта		Поиск ш	аблонов (л	ALT+'B")			<i>-</i> م	Очи	стить все								
Последние шаблоны проектов		C#			Windows		- B	се типы проектов									
Веб приложение ASP.NET Core (модель представление контроллер)	C#	E/CF	Консолын Проект дл	ое приложен: в создания пр	ме жиложения ко	мандной стр	роки, катар	ре может выполня	ться в среде .NET	Core s Windows	, Linux и Mac C)S					
Пустой ASP.NET Core	CN		Ce U	nux macOS	Windows	Консоль											
Веб приложение ASP.NET (.NET Framework)	CX		Библиоте Проект дл	а классов я создания бы	иблиотеки кла	ссов, испол	ьзующей .NI	ET Standard или .NE	ET Core								
Веб приложение ASP.NET Core	CN		C# A	ndroid Linu	x macOS	Windows	Библиоте	-ca									
Веб приложение ASP.NET (.NET Framework)	Visual Basic	Z"	Пустой AS Пустой ша	P.NET Core 6лон проекта	а для создании	я приложені	ия ASP.NET (Соге. Эгот шаблон	не имеет содерж	имого.							
			C# U	nux macO5	Windows	Облако	Служба	BeG									
		• O	Веб АРІ А Шаблон п представл	SP.NET Core роекта для со ений MVC и в	адания прило юнтроллеров	жения ASP.I ASP.NET.Cor	NET Core c o	бразцом контролл	тера для службы і	HTTP RESTIUL 31	от шаблон мо:	кно также испо.	льзовать для				
			C# U	nux macO5	Windows	Облако	Служба	BeG									
		36	Worker Se Шаблон п	vice ycroro npoeki	а для создани	1я службы W	/orker Servic										
			ce u	nux macOS	Windows	Облако	Служба										
			Веб-прил Шаблон п	жение ASP.N роекта для со	ET Core адания прило	жения ASP.I	NET Core c n	римером содержи	woro Razor Pages	ASP.NET.							
			Ce Li	nux macOS	Windows	Облако	Служба	Be6									
		Ē	Веб-прил Шаблон п для служб	ожение ASP.N роекта для со HTTP RESTful	ЕТ Соге (моде адания прило	ль-предстае жения ASP.I	аление-кант NET Care c a	роллер) бразцом представ	лений MVC и кан	троллеров ASP.	NET Core. Этот	шаблон можне	также использое	ать			
			C? U	nus macOS	Windows	Облако	Служба	Be6									
		gRPC	Служба gl Шаблон п	КРС ASP.NET С роекта для со	iore адания служб	ы gRPC ASP.	NET Core.										
														Назад	Дал	ee	



Обратите внимание на «Настроить для HTTPS» - квадратик пуст.

Настройка стиля сайта

Выполните незначительную настройку меню, макета и домашней страницы сайта.

Откройте файл Views/Shared/_Layout.cshtml и внесите следующие изменения:

```
<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white
border-bottom box-shadow mb-3">
                                     <div class="container">
                                                  <a class="navbar-brand" asp-area="" asp-controller="Home" asp-
action="Index">Kafedra02</a>
                                                  <br/>
<body>
<br/>
target=".navbar-collapse" aria-controls="navbarSupportedContent"
                                                                          aria-expanded="false" aria-label="Toggle navigation">
                                                              <span class="navbar-toggler-icon"></span>
                                                  </button>
                                                  <div class="navbar-collapse collapse d-sm-inline-flex flex-sm-row-
reverse">
                                                              <a class="nav-link text-dark" asp-area="" asp-
controller="Home" asp-action="Index">Home</a>
                                                                          \langle /|i\rangle
                                                                          <a class="nav-link text-dark" asp-area="" asp-
controller="Disziplinas" asp-action="Index">Дисциплинa</a>
                                                                           </div>
                                     </div>
                         </nav>
            </header>
            <div class="container">
                         <main role="main" class="pb-3">
                                     @RenderBody()
                         </main>
            </div>
            <footer class="border-top footer text-muted">
                         <div class="container">
                                     © 2021 - Kafedra02 - <a asp-area="" asp-controller="Home" asp-</pre>
action="Privacy">Privacy</a>
                         </div>
            </footer>
            <script src="~/lib/jquery/dist/jquery.min.js"></script>
            <script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script
            <script src="~/js/site.js" asp-append-version="true"></script>
            @RenderSection("Scripts", required: false)
</body>
</html>
                                              <script
src="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"
                                                 asp-fallback-src="~/lib/bootstrap/dist/js/bootstrap.min.js"
                                                  asp-fallback-test="window.jQuery && window.jQuery.fn &&
window.jQuery.fn.modal"
                                                 crossorigin="anonymous"
                                                 integrity="sha384-
aJ210jlMXNL5UyI1/XNwTMqvzeRMZH2w8c5cRVpzpU8Y5bApTppSuUkhZXN0VxHd">
                         </script>
                         <script src="~/js/site.min.js" asp-append-version="true"></script></script>
            </environment>
            @RenderSection("Scripts", required: false)
</body>
</html>
```

Замените содержимое файла Views/Home/Index.cshtml следующим кодом, который заменяет текст о ASP.NET и MVC описанием этого приложения:

```
ViewBag.Title = "Home Page";
```

```
<hl><hl>style="text-align:center; color:#796310">Курс ASP.NET Core MVC - создание базы данных</hl>
```

```
Создадим в папке Models сущностей (таблицу будущей базы данных) – тип файла класс
Disziplina
И заменим сгенерированный код следующим:
using Microsoft.AspNetCore.Mvc;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
namespace Kafedra02
{
    public class Disziplina
   {
        [HiddenInput(DisplayValue = false)]
        [Display(Name = "Ключ записи")]
        public int DisziplinaID { get; set; }
        [Display(Name = "Название дисциплины")]
        public string Nazvanie { get; set; }
        [Display(Name = "Зачетные единицы")]
        public string ZachetEdin { get; set; }
        [Display(Name = "Курсовая работа")]
        public string KursRabota { get; set; }
        [Display(Name = "Контрольная работа")]
        public string KontRabota { get; set; }
        [Display(Name = "Форма получения оценки")]
        public string FormaOzenki { get; set; }
        [Display(Name = "Ссылка на рабочую программу в папке Uploads/RabProg")]
        public string ObrzavProgramma { get; set; }
        [Display(Name = "Имя файла рабочей программы")]
        public string DDopPole01 { get; set; }
        [Display(Name = "Читает кафедра")]
        public string DDopPole02 { get; set; }
        [Display(Name = "Направление- Бакалавр или Магистр или Аспирант")]
        public string DDopPole03 { get; set; }
        [Display(Name = "Приобретаемые компетенции, записываются через точку с запятой
без пробелов")]
        public string DDopPole04 { get; set; }
        [Display(Name = "Доп. поле 05")]
        public string DDopPole05 { get; set; }
```

```
public virtual ICollection<Disziplina> Disziplinas { get; set; }
}
}
```

```
<mark>@{</mark>
```

Создание контекста базы данных

Контекст базы данных — это основной класс, который координирует функциональные возможности Entity Framework для заданной модели данных. Этот класс coздается путем наследования от класса Microsoft.EntityFrameworkCore.DbContext. В коде указываются сущности, которые включаются в модель данных. Также вы можете настроить реакцию платформы Entity Framework на некоторые события. В этом проекте соответствующий класс называется Kafedra02Context.

В папке проекта создайте папку Data.

В папке *Data* создайте новый файл класса с именем *Kafedra02Context.cs* и замените код шаблона следующим кодом:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.EntityFrameworkCore;
using Kafedra02;
namespace Kafedra02.Data
{
    public class Kafedra02Context : DbContext
    {
        public Kafedra02Context (DbContextOptions<Kafedra02Context> options)
            : base(options)
        {
        }
        public DbSet<Kafedra02.Disziplina> Disziplina { get; set; }
    }
}
```

Регистрация Kafedra02Context

ASP.NET Core по умолчанию реализует технологию внедрения зависимостей. С помощью внедрения зависимостей службы (например, контекст базы данных EF) регистрируются во время запуска приложения. Затем компоненты, которые используют эти службы (например, контроллеры MVC), обращаются к ним через параметры конструктора. Код конструктора контроллера, который получает экземпляр контекста, будет приведен позднее в этом учебнике. Чтобы зарегистрировать Kafedra02Context как службу, откройте файл Startup.cs и добавьте выделенные строки в метод ConfigureServices.

```
using Kafedra02.Data;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
namespace Kafedra02
{
    public class Startup
    {
        public Startup(IConfiguration configuration)
        {
            Configuration = configuration;
        }
        public IConfiguration Configuration { get; }
        // This method gets called by the runtime. Use this method to add services to the
container.
        public void ConfigureServices(IServiceCollection services)
        ł
```

```
services.AddControllersWithViews();
```

services.AddDbContext<Kafedra02Context>(options =>

options.UseSqlServer(Configuration.GetConnectionString("Kafedra02Context")));
}

```
// This method gets called by the runtime. Use this method to configure the HTTP
request pipeline.
        public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
        {
            if (env.IsDevelopment())
            {
                app.UseDeveloperExceptionPage();
            }
            else
            {
                app.UseExceptionHandler("/Home/Error");
            }
            app.UseStaticFiles();
            app.UseRouting();
            app.UseAuthorization();
            app.UseEndpoints(endpoints =>
            {
                endpoints.MapControllerRoute(
                    name: "default",
                    pattern: "{controller=Home}/{action=Index}/{id?}");
           });
      }
   }
}
```

Откройте файл *appsettings.json* и добавьте строку подключения, как показано в следующем примере.

```
{
    "Logging": {
        "LogLevel": {
            "Default": "Information",
            "Microsoft": "Warning",
            "Microsoft.Hosting.Lifetime": "Information"
        }
    },
    "AllowedHosts": "*",
    "ConnectionStrings": {
        "Kafedra02Context": "Server=(localdb)\\mssqllocaldb;Database=KafedraContext-
1;Trusted_Connection=True;MultipleActiveResultSets=true",
    }
}
```

SQL Server Express LocalDB

Строка подключения указывает на базу данных SQL Server LocalDB. LocalDB — это упрощенная версия ядра СУБД SQL Server Express, предназначенная для разработки приложений и не ориентированная на использование в производственной среде. LocalDB запускается по запросу в пользовательском режиме, поэтому настройки не слишком сложны.

Формирования шаблона страниц дисциплины

Используйте средство формирования шаблонов, чтобы создать страницы для операций создания, чтения, обновления и удаления (CRUD) для модели дисциплина.

Для работы с базой данных необходимо добавление пакетов NuGet

В меню Сервис последовательно выберите пункты Диспетчер пакетов NuGet > Консоль диспетчера пакетов

Ввести Install-Package Microsoft.EntityFrameworkCore.SqlServer



Приведенная выше команда добавляет поставщик EF Core для SQL Server. Пакет поставщика устанавливает пакет EF Core в качестве зависимости. Дополнительные пакеты устанавливаются автоматически на этапе формирования шаблонов далее в этом руководстве.

Формирования шаблона страниц дисциплин

Используйте средство формирования шаблонов, чтобы создать страницы для операций создания, чтения, обновления и удаления (CRUD) для модели фильма.





DistribulingsController of A annualtions ison Startup of KaladraContext	s		
Kafedra02	trollers DisziplinasController		· & -
1 📍 🗏 using System;			0
using System.itna; using System.itna; using System.itna; using Nicrosoft.Aspletcore.Wrc; using Nicrosoft.Aspletcore.Wrc; using Nicrosoft.Aspletcore.Wrc; using Kafedrad2.bata; using Kafedrad2.bata; using Kafedrad2.controllers forcest: public class DisziplinasController : Controller forcest: Dote: State State State State State State	Добавить Контроллер МVС с представлениями, и Кисс можник рызіріпа (Kaledna02) Добавление контекста данных		0
Added Vies (ViewsUkispineskylette.totkul Added Vies (ViewsUkispineskylette.totkul Added Vies (ViewsUkispineskylette.cotkul Added Vies (ViewsUkispineskylette.cotkul Added Vies (ViewsUkispineskylette.cotkul	Новый тип контекста денных Каfedra02.Data.Kafedra02Context Добавить	Otwella > Stared Ørwella > Viewstruchtmil Ørwella > Groppan.cs > c* Program.cs > c* Startup.cs	
Documents Approximate International Monousing Approximate International Systems Collections. Immutable 5:80.8° ycmesse ycrawcameen 6:Kafedraw2 Systems Collections. Immutable 5:80.8° ycmesse ycrawcameen 6:Kafedraw2 Systems Outgowers (Sources 5:80.8° ycmesse ycrawcameen 6:Kafedraw2 Systems Outgowers (Sources 5:80.8° ycmesse ycrawcameen 6:Kafedraw2 Systems Outgowers (Sources 5:80.8° ycmesse ycrawcameen 6:Kafedraw2 Systems Collections). Jones (Sources 5:80.8° ycmesse ycrawcameen 6:Kafedraw2 Systems Collections). Jones (Sources 5:80.8° ycmesse) 9:Kafedraw2 Systems Collections). Jones (Sources 5:80.8° ycmesse) 9:Kafedraw2 Systems Collections). Jones (Sources 5:80.8° ycmesse) 9:Kafedraw2 Systems Security. Jennistisis 5:7.0° ycmesse ycrawcameen 6:Kafedraw2 Systems Collective Stretsions 5:7.3° ycmesse ycrawcameen 6:Kafedraw2 Systems Collective (Stretsions 5:7.3° ycmesse ycrawcameen 6:Kafedraw2 Systems Security (Stretsions 5:7.3° ycresse ycrawcameen 6:Kafedraw2 Systems Security (Stretsions 5:7.3° ycrawcamey ycrawcameen 6:Kafedraw2 Systems Security (Stretsions 5:7.3° ycresse ycrawcameen 6:Kafedraw2 Systems Security	Имя контроинера Disciplinas1Controller Добавить С	OTweeva	

Visual Studio создаст следующие компоненты:

- контроллер дисциплин (Controllers/DisziplinasController.cs);
- файлы представления Razor для страниц Create, Delete, Details, Edit и Index (*Views/Disziplinas/*.cshtml*).

Автоматическое создание этих файлов называется формированием шаблонов.

Сформированные страницы пока использовать нельзя, так как база данных не существует. Если запустить приложение и щелкнуть ссылку **Дисциплина** в меню, появится сообщение об ошибке *Невозможно открыть базу* данных или отсутствует таблица. Необходимо создать базу данных.

Первоначальная миграция

Создайте базу данных с помощью функции <u>миграций</u> EF Core. Миграции — это набор средств, позволяющих создавать и обновлять базы данных в соответствии с моделью данных.

В меню Сервис последовательно выберите пункты Диспетчер пакетов NuGet > Консоль диспетчера пакетов

Введите следующие команды:

Add-Migration InitialCreate Update-Database

- Add-Migration InitialCreate. Создает файл миграции Migrations/{метка_времени}_InitialCreate.cs. Аргумент InitialCreate это имя миграции. Можно использовать любое имя, но по соглашению выбирается имя, которое описывает миграцию. Так как это первая миграция, созданный класс содержит код для создания схемы базы данных. Схема базы данных создается на основе модели, указанной в классе Kafedra02Context.
- > Update-Database. Обновляет базу данных до последней миграции, созданной предыдущей командой.