**创建ContosoUniversity项目的步骤**

**一、创建 mvc core新项目<br/>**

**二、项目名称**：ContosoUniversity<br/>

**三、增加菜单项**，在Views/Shared/\_Layout.cshtml的菜单中增加如下代码

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="About">About</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Students" asp-action="Index">学生</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Courses" asp-action="Index">课程</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Instructors" asp-action="Index">讲师</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Departments" asp-action="Index">部门</a>

</li>

**四、在Models文件夹中添加Student类**

public class Student

{

public int ID { get; set; }

public string LastName { get; set; }

public string FirstMidName { get; set; }

public DateTime EnrollmentDate { get; set; }

public ICollection<Enrollment> Enrollments { get; set; }

}

**五、继续添加Enrollment和Course类**

1、Enrollment类

public enum Grade

{

A, B, C, D, F

}

public class Enrollment

{

public int EnrollmentID { get; set; }

public int CourseID { get; set; }

public int StudentID { get; set; }

public Grade? Grade { get; set; }

public Course Course { get; set; }

public Student Student { get; set; }

}

2、Course类

public class Course

{

[DatabaseGenerated(DatabaseGeneratedOption.None)]

public int CourseID { get; set; }

public string Title { get; set; }

public int Credits { get; set; }

public ICollection<Enrollment> Enrollments { get; set; }

}

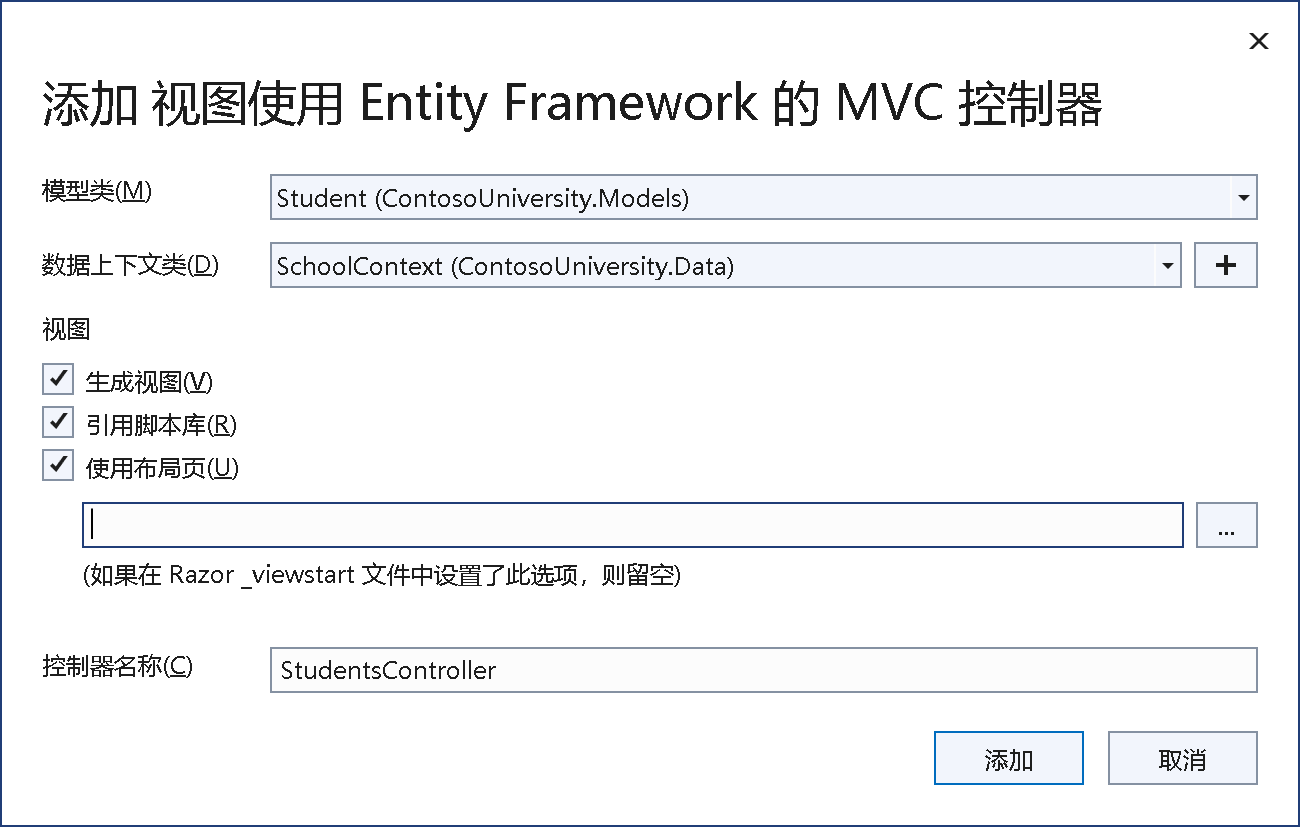
注意：引用 using System.ComponentModel.DataAnnotations.Schema;

**六、编译并运行，查看程序是否能正确运行。**

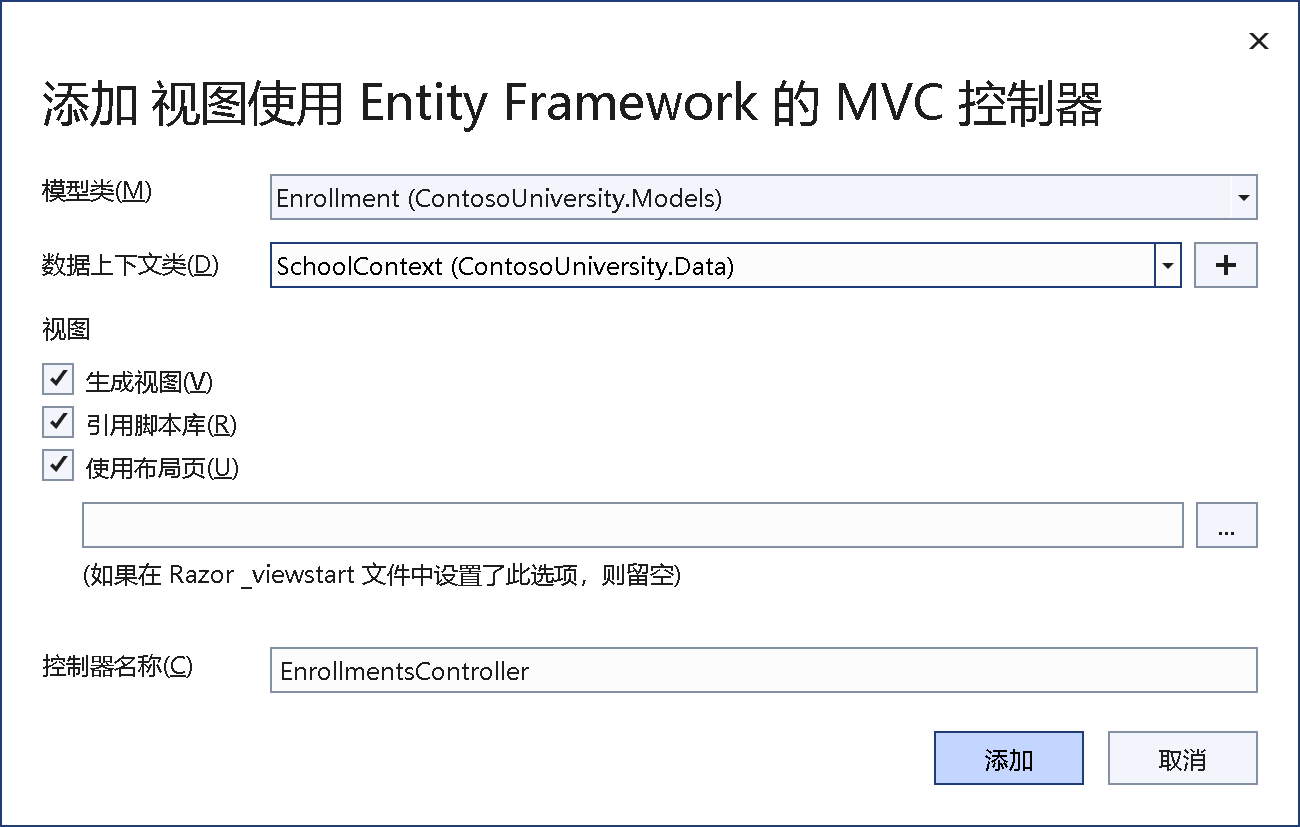
**七、如果是.net 6.0的项目**（5.0的项目调过此步骤），需要修改项目文件，方法为：在“解决方案资源管理器”的项目名称上

右键->属性->生成->可为 Null 类型 选项设为禁用

**八、添加Controller**：在Controllers上右键->添加->控制器，按如下方式添加，如果出错误多试几次。



**九、以此类推，**继续添加 Enrollment和Course模型类。



数据上下文类无需修改或增加。

十、创建数据库

1、add-migration init

2、update-database

十一、数据自动填充

在data文件夹下创建DbInitializer类，内容为：

using ContosoUniversity.Models;

using System;

using System.Linq;

namespace ContosoUniversity.Data

{

public static class DbInitializer

{

public static void Initialize(SchoolContext context)

{

context.Database.EnsureCreated();

// Look for any students.

if (context.Student.Any())

{

return; // DB has been seeded

}

var students = new Student[]

{

new Student{FirstMidName="Carson",LastName="Alexander",EnrollmentDate=DateTime.Parse("2005-09-01")},

new Student{FirstMidName="Meredith",LastName="Alonso",EnrollmentDate=DateTime.Parse("2002-09-01")},

new Student{FirstMidName="Arturo",LastName="Anand",EnrollmentDate=DateTime.Parse("2003-09-01")},

new Student{FirstMidName="Gytis",LastName="Barzdukas",EnrollmentDate=DateTime.Parse("2002-09-01")},

new Student{FirstMidName="Yan",LastName="Li",EnrollmentDate=DateTime.Parse("2002-09-01")},

new Student{FirstMidName="Peggy",LastName="Justice",EnrollmentDate=DateTime.Parse("2001-09-01")},

new Student{FirstMidName="Laura",LastName="Norman",EnrollmentDate=DateTime.Parse("2003-09-01")},

new Student{FirstMidName="Nino",LastName="Olivetto",EnrollmentDate=DateTime.Parse("2005-09-01")}

};

foreach (Student s in students)

{

context.Student.Add(s);

}

context.SaveChanges();

var courses = new Course[]

{

new Course{CourseID=1050,Title="Chemistry",Credits=3},

new Course{CourseID=4022,Title="Microeconomics",Credits=3},

new Course{CourseID=4041,Title="Macroeconomics",Credits=3},

new Course{CourseID=1045,Title="Calculus",Credits=4},

new Course{CourseID=3141,Title="Trigonometry",Credits=4},

new Course{CourseID=2021,Title="Composition",Credits=3},

new Course{CourseID=2042,Title="Literature",Credits=4}

};

foreach (Course c in courses)

{

context.Course.Add(c);

}

context.SaveChanges();

var enrollments = new Enrollment[]

{

new Enrollment{StudentID=1,CourseID=1050,Grade=Grade.A},

new Enrollment{StudentID=1,CourseID=4022,Grade=Grade.C},

new Enrollment{StudentID=1,CourseID=4041,Grade=Grade.B},

new Enrollment{StudentID=2,CourseID=1045,Grade=Grade.B},

new Enrollment{StudentID=2,CourseID=3141,Grade=Grade.F},

new Enrollment{StudentID=2,CourseID=2021,Grade=Grade.F},

new Enrollment{StudentID=3,CourseID=1050},

new Enrollment{StudentID=4,CourseID=1050},

new Enrollment{StudentID=4,CourseID=4022,Grade=Grade.F},

new Enrollment{StudentID=5,CourseID=4041,Grade=Grade.C},

new Enrollment{StudentID=6,CourseID=1045},

new Enrollment{StudentID=7,CourseID=3141,Grade=Grade.A},

};

foreach (Enrollment e in enrollments)

{

context.Enrollment.Add(e);

}

context.SaveChanges();

}

}

}

十二、打开 Program.cs文件，添加如下代码

using (var scope = app.Services.CreateScope())

{

var services = scope.ServiceProvider;

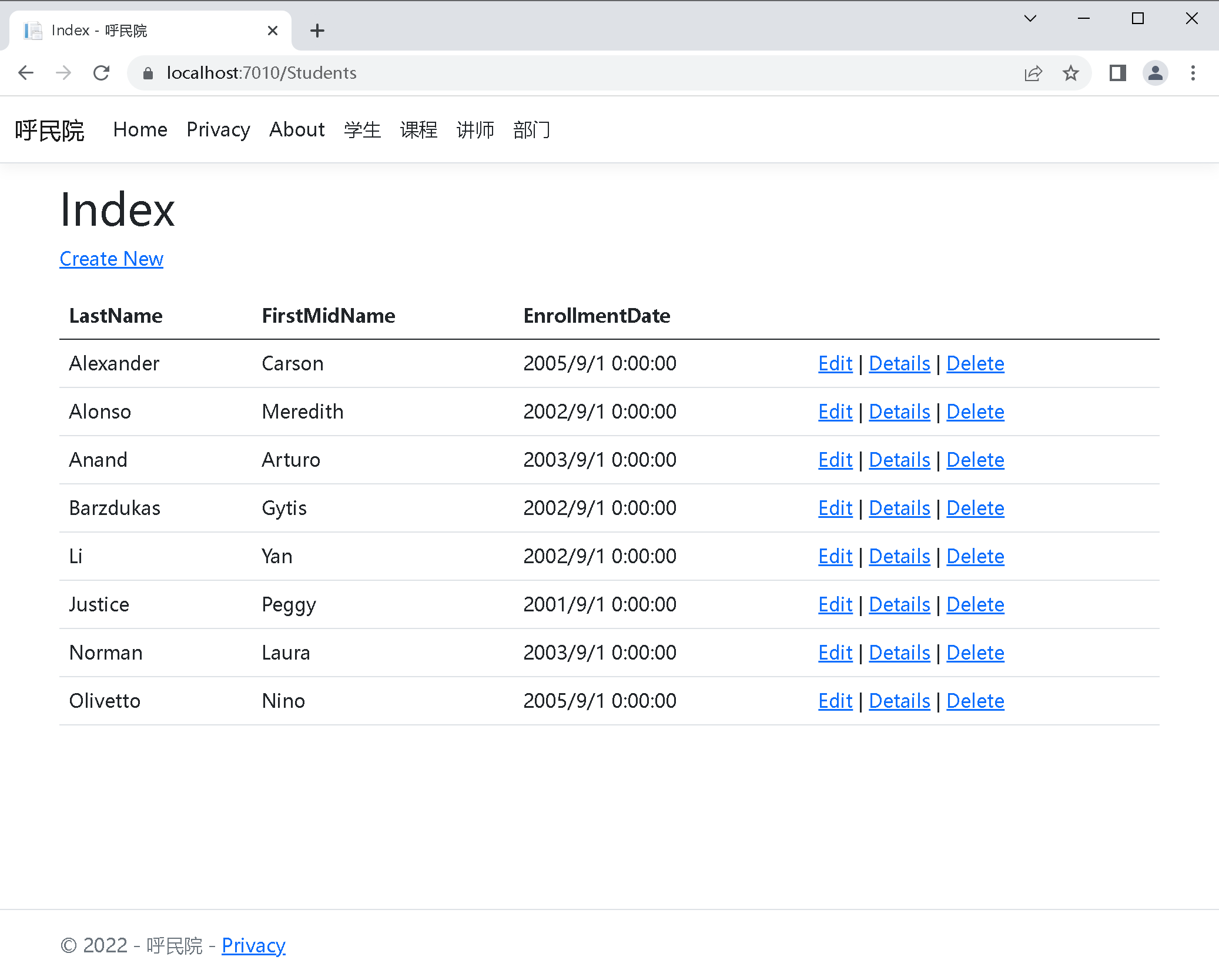
var context = services.GetRequiredService<SchoolContext>();

context.Database.EnsureCreated();

DbInitializer.Initialize(context);

}

执行程序，查看学生和课程内容是否填充正确，正确显示如下图所示内容：



十三、在appsettings.json文件的

"Microsoft.AspNetCore": "Warning"下面添加如下代码

,"Microsoft.EntityFrameworkCore.Database.Command": "Information"

十四、打开controllers里的StudentsController.cs，并修改Details函数的var student改为如下部分

var student = await \_context.Student

.Include(s=>s.Enrollments)

.ThenInclude(e=>e.Course)

.AsNoTracking()

.FirstOrDefaultAsync(m => m.ID == id);

十五、打开Views/Student/Details.cshtml文件，在

<dt class="col-sm-2">

@Html.DisplayNameFor(model => model.Enrollments)

</dt>

下面增加如下代码：

<dd class="col-sm-10">

<table class="table">

<tr>

<th>课程名称</th>

<th>成绩</th>

</tr>

@foreach (var item in Model.Enrollments)

{

<tr>

<td>

@Html.DisplayFor(modelItem => item.Course.Title)

</td>

<td>

@Html.DisplayFor(modelItem => item.Grade)

</td>

</tr>

}

</table>

</dd>

附件：

**SQLServer安装方法：**

**打开vs 2022或2019，选择菜单的“工具”->获取工具和功能，之后如下图所示步骤安装即可。**

