**Проект Educational Center**

ASP.NET MVC ученически проект

Използвани тегнологии:

* VisualStudio 2022
* C#
* HTML
* EntityFramework
* Microsoft SQL Server

Цел на проекта:

* Създаване на проложение лесно за използване от ученици и учители. Улесняване на създаването и оправляване на онлайн/училищни курсове и самоинициатива.

Изготвил: Ръководител:

Николай Симеонов Госпожа Набила Сарвари

Функционалности на **User Controller**:

* **StudentView**

Методът **StudentView** връща изглед с списък на всички студенти.

* **StudentCreate**

Методът **StudentCreate** връща изглед за създаване на нов студент.

* **StudentCreate (HttpPost)**

Методът **StudentCreate** с **HTTP POST** приема данни от формата за създаване на студент и ги добавя в базата данни.

* **StudentUpdate**

Методът **StudentUpdate** връща изглед за редактиране на съществуващ студент, идентифициран по уникален идентификатор.

* **StudentUpdate (HttpPost)**

Методът **StudentUpdate** **с HTTP POST** приема актуализирани данни за студента и ги записва в базата данни.

* **StudentDelete**

Методът **StudentDelete** с **HTTP DELETE** премахва студент от базата данни по зададен уникален идентификатор.

* **TeacherView**

Методът **TeacherView** връща изглед със списък на всички учители.

* **TeacherCreate**

Методът **TeacherCreate** връща изглед за създаване на нов учител.

* **TeacherCreate (HttpPost)**

Методът TeacherCreate с **HTTP POST** приема данни от формата за създаване на учител, както и списък от курсове, и ги добавя в базата данни.

* **TeacherUpdate**

Методът **TeacherUpdate** връща изглед за редактиране на съществуващ учител, идентифициран по уникален идентификатор.

* **TeacherUpdate (HttpPost)**

Методът **TeacherUpdate** с **HTTP POST** приема актуализирани данни за учителя и ги записва в базата данни.

* **TeacherDelete**

Методът **TeacherDelete** с **HTTP DELETE** премахва учител от базата данни по зададен уникален идентификатор.

Функционалности на **Course Controller**:

* **Index**

Методът **Index** връща изглед с информация за всички курсове, включително студенти и учители, свързани с тях.

* **CreateCourse**

Методът **CreateCourse** връща изглед за създаване на нов курс.

* **CreateCourse (HttpPost)**

Методът **CreateCourse** с **HTTP POST** приема данни от формата за създаване на курс и ги добавя в базата данни.

* **CourseUpdate**

Методът **CourseUpdate** връща изглед за редактиране на съществуващ курс, идентифициран по уникален идентификатор.

* **CourseUpdate (HttpPost)**

Методът **CourseUpdate** с **HTTP POST** приема актуализирани данни за курса и ги записва в базата данни.

* **CourseDelete**

Методът **CourseDelete** с **HTTP DELETE** премахва курс от базата данни по зададен уникален идентификатор.

* **StudentRemove**

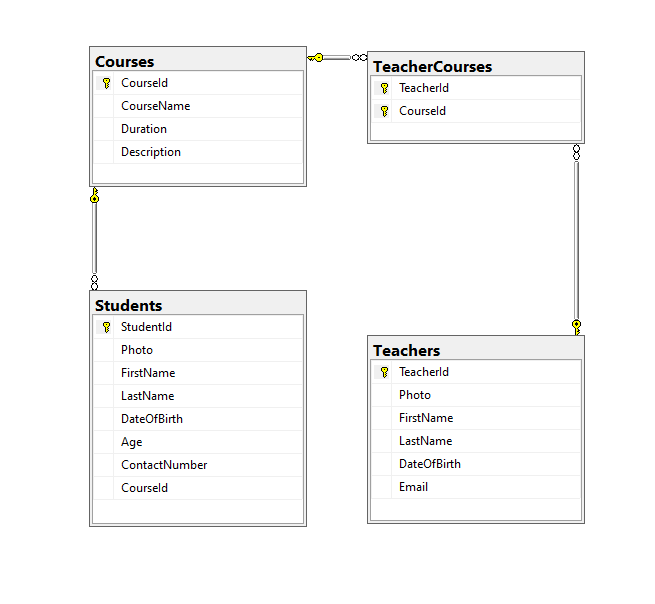
Методът **StudentRemove** с **HTTP POST** премахва студент от курса и базата данни.

* **TeacherRemove**

Методът **TeacherRemove** **с HTTP POST** премахва учител от курсa.

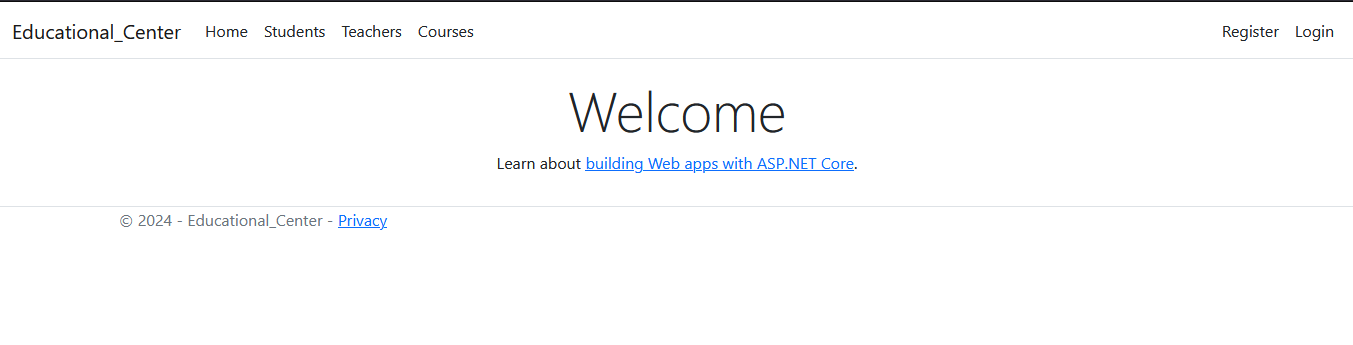
* **Приложения**

**Фиг.1** Диаграма на базата данни:



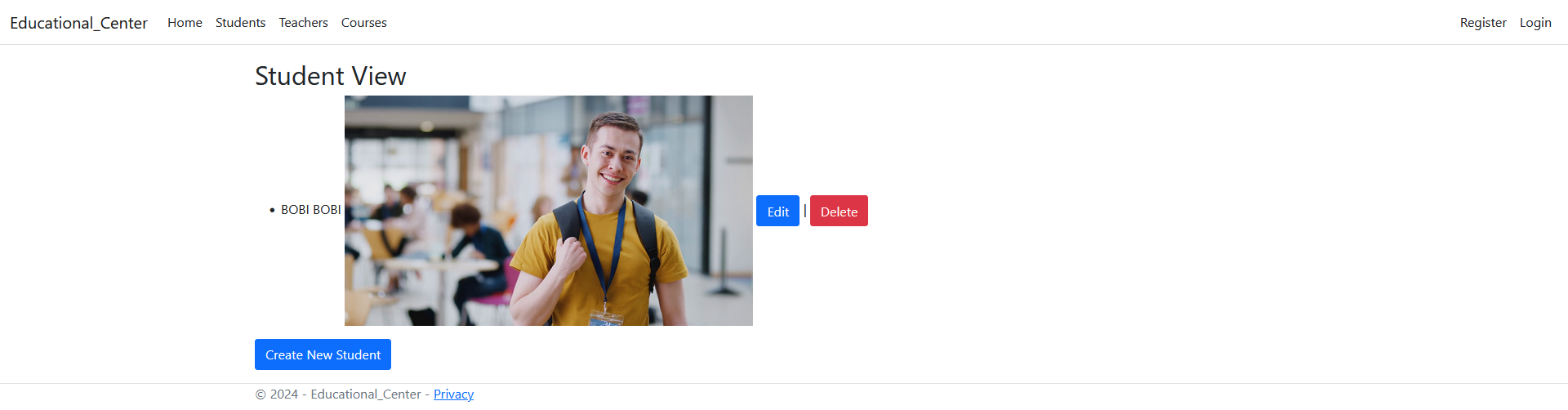
**Фиг.2**

Изглед на проекта в основната страница:



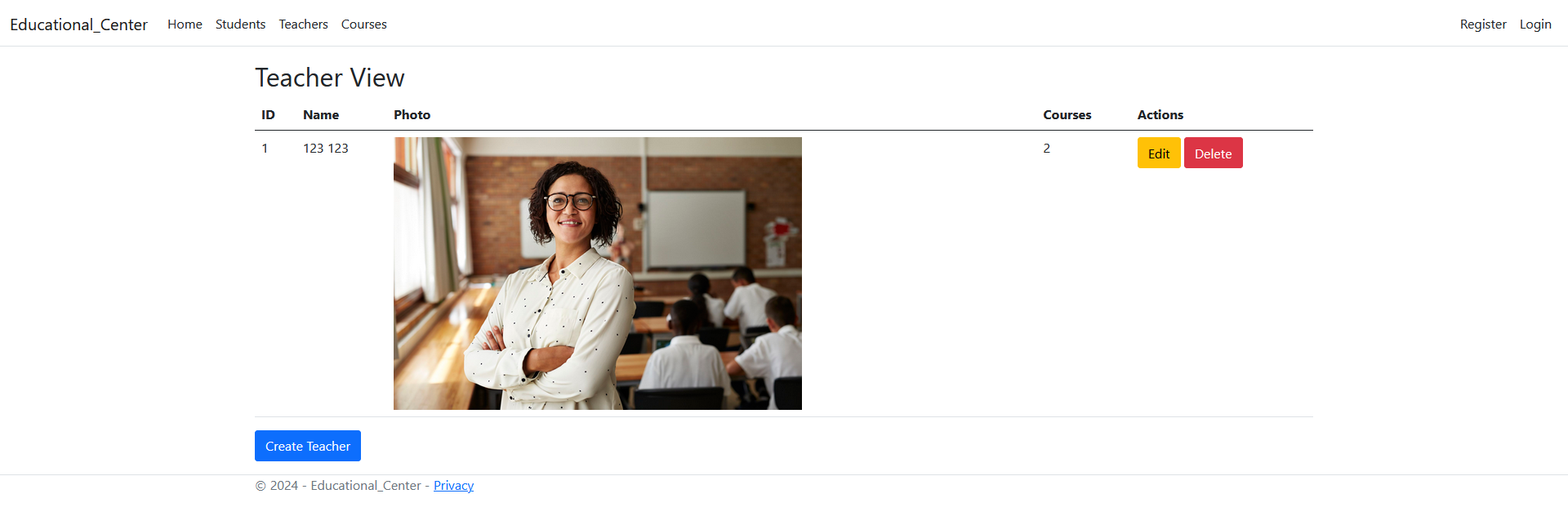
**Фиг.3**

Изглед на страницата **Students**:

****

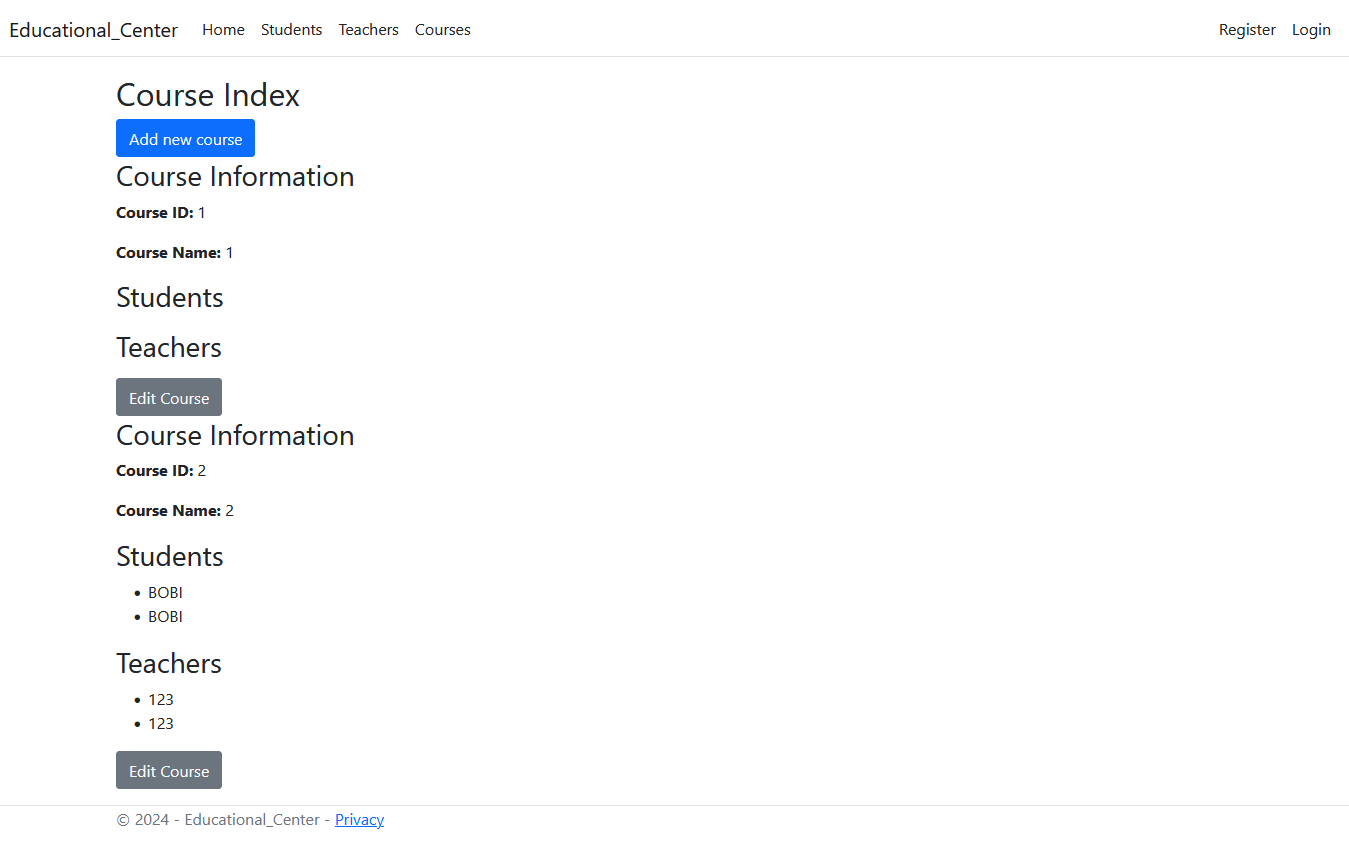
**Фиг.4**

Изглед на страницата **Teachers**:



**Фиг.5**

Изглед на страницата **Courses**:

****

**Фиг.6**

Изглед на проекта в разработната среда:

