

# Student Information System

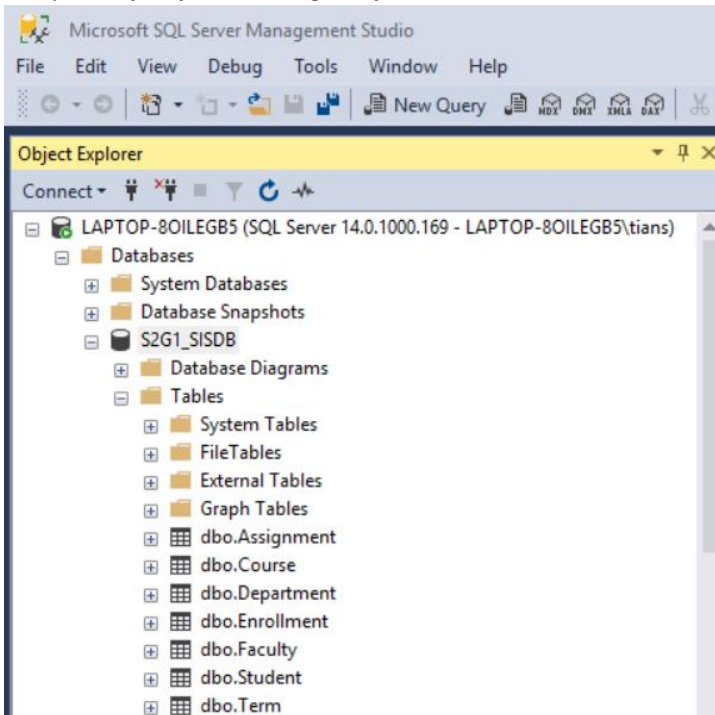
## 1. Azure and Github

Source code published on github: <https://github.com/hongshuidang/courseRegisterSystem>

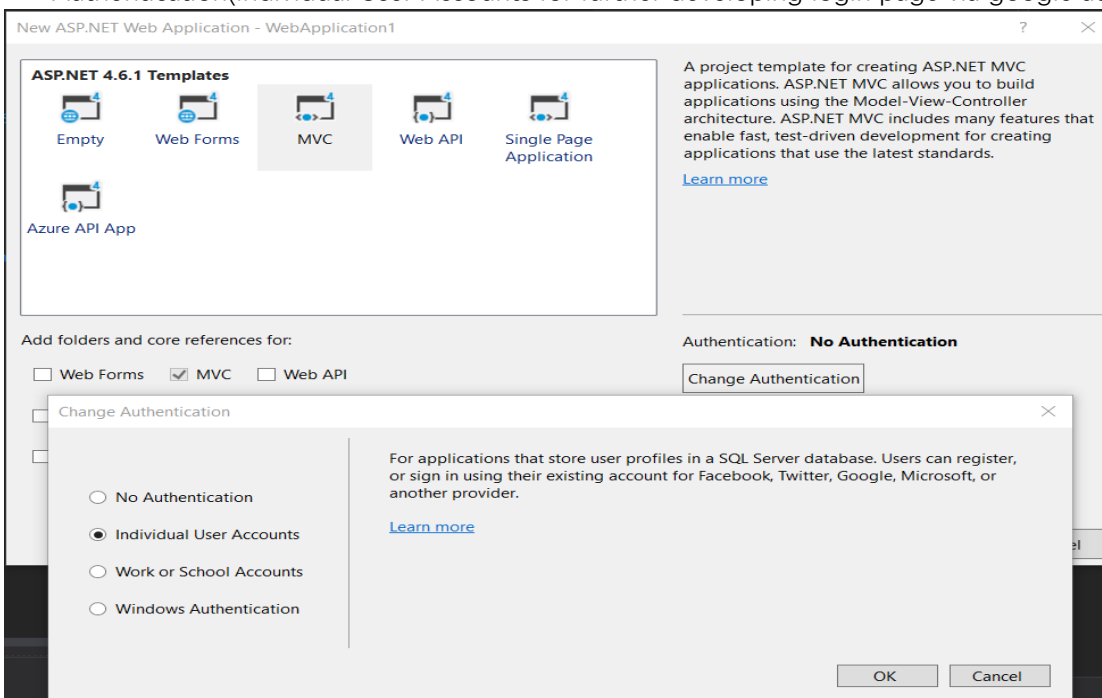
Application deployed on Azure: <https://s2g1sis.azurewebsites.net>

## 2. Prerequisite

You have designed the schema in your local Sql server Database, including attributes and their domain, and constraints like primary key and foreign keys.



## 3. Create a new project in Visual Studio Enterprise IDE: File->New->ASP.NET Web Application->MVC->Change Authentication(Individual User Accounts for further developing login page via google account).



4. Create Model: Models->Add->New Item->Data->ADO.NET Entity Data Model->EF Designer from database(Server name can be found in Master Data Services)

Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:  
Microsoft SQL Server (SqlClient) Change...

Server name:  
LAPTOP-8OILEGB5 Refresh

Log on to the server

Authentication: Windows Authentication

User name:

Password:

☐ Save my password

Connect to a database

☒ Select or enter a database name:  
S2G1\_SISDB ▼


☐ Attach a database file:  
 Browse...

Logical name:

Advanced...

Test Connection OK Cancel

Entity Data Model Wizard

 **Choose Your Data Connection**

**Which data connection should your application use to connect to the database?**

S2G1\_SISDBEntities (S2G1\_SISAPP) New Connection...

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set it in my application code.

☐ Yes, include the sensitive data in the connection string.

Connection string:


```
metadata=res://*/Models.Model2.csdl|res://*/Models.Model2.ssdl|
res://*/Models.Model2.msl;provider=System.Data.SqlClient;provider connection string="data
source=LAPTOP-8OILEGB5;initial catalog=S2G1_SISDB;integrated
security=True;multipleactiveresultsets=True;application name=EntityFramework"
```

☒ Save connection settings in Web.Config as:

S2G1\_SISDBEntities1

< Previous Next > Finish Cancel

Entity Data Model Wizard

 **Choose Your Database Objects and Settings**

**Which database objects do you want to include in your model?**

☒ Tables

☒ dbo

☒ Assignment

☒ Course

☒ Department

☒ Enrollment

☒ Faculty

☒ Student

☒ Term

☐ mdm

☐ Views

☐ Stored Procedures and Functions

☒ Pluralize or singularize generated object names

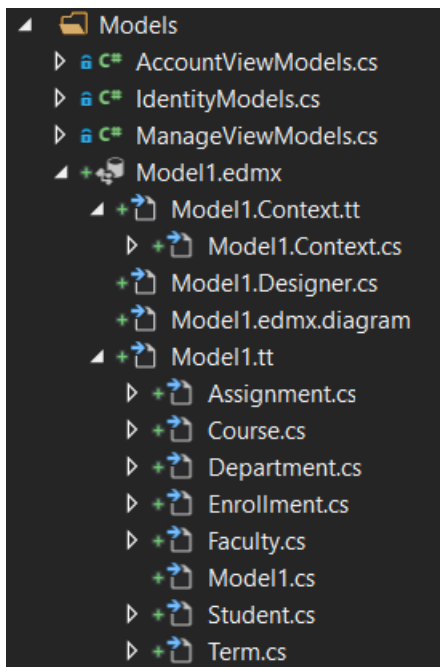
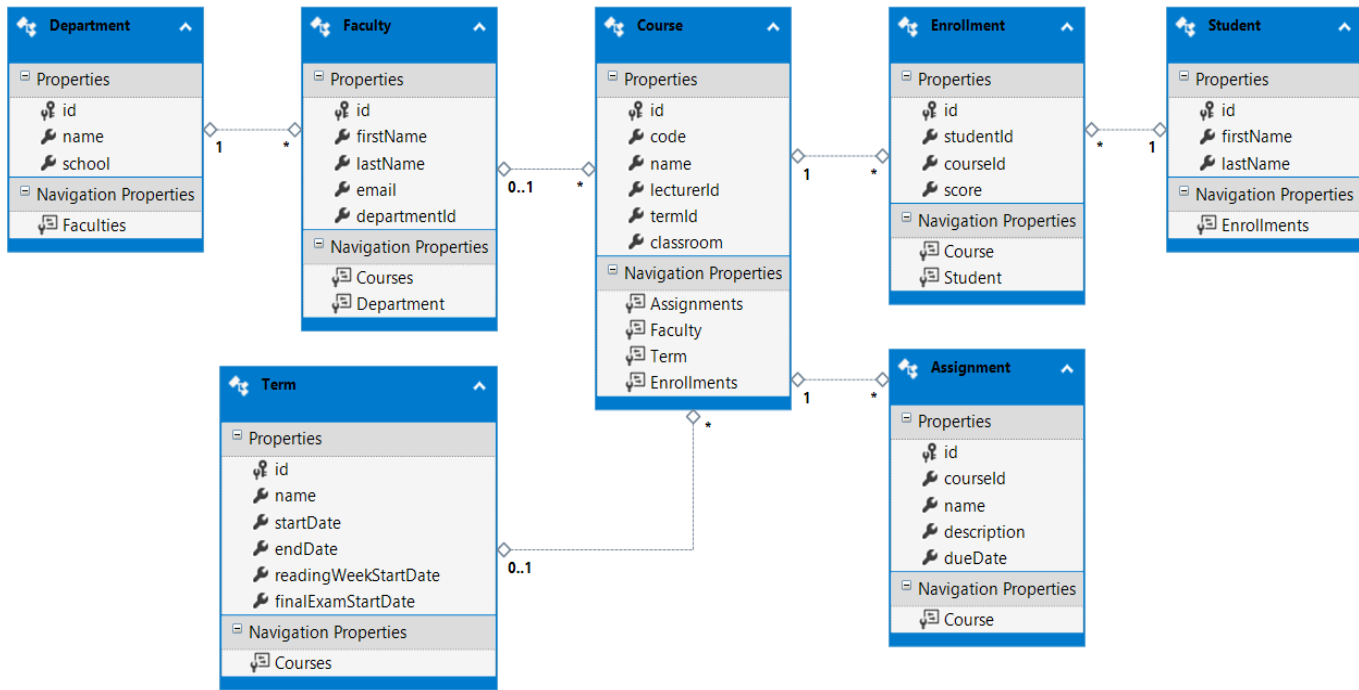
☒ Include foreign key columns in the model

☒ Import selected stored procedures and functions into the entity model

Model Namespace:  
S2G1\_SISDBModel1

< Previous Next > Finish Cancel

After this step, a class diagram and all entity classes are generated.



```

namespace CourseRegister.Models
{
    using System;
    using System.Collections.Generic;

    3 references | 0 changes | 0 authors, 0 changes
    public partial class Assignment
    {
        0 references | 0 changes | 0 authors, 0 changes
        public int id { get; set; }
        0 references | 0 changes | 0 authors, 0 changes
        public int courseId { get; set; }
        0 references | 0 changes | 0 authors, 0 changes
        public string name { get; set; }
        0 references | 0 changes | 0 authors, 0 changes
        public string description { get; set; }
        0 references | 0 changes | 0 authors, 0 changes
        public Nullable<System.DateTime> dueDate { get; set; }

        0 references | 0 changes | 0 authors, 0 changes
        public virtual Course Course { get; set; }
    }
}

```

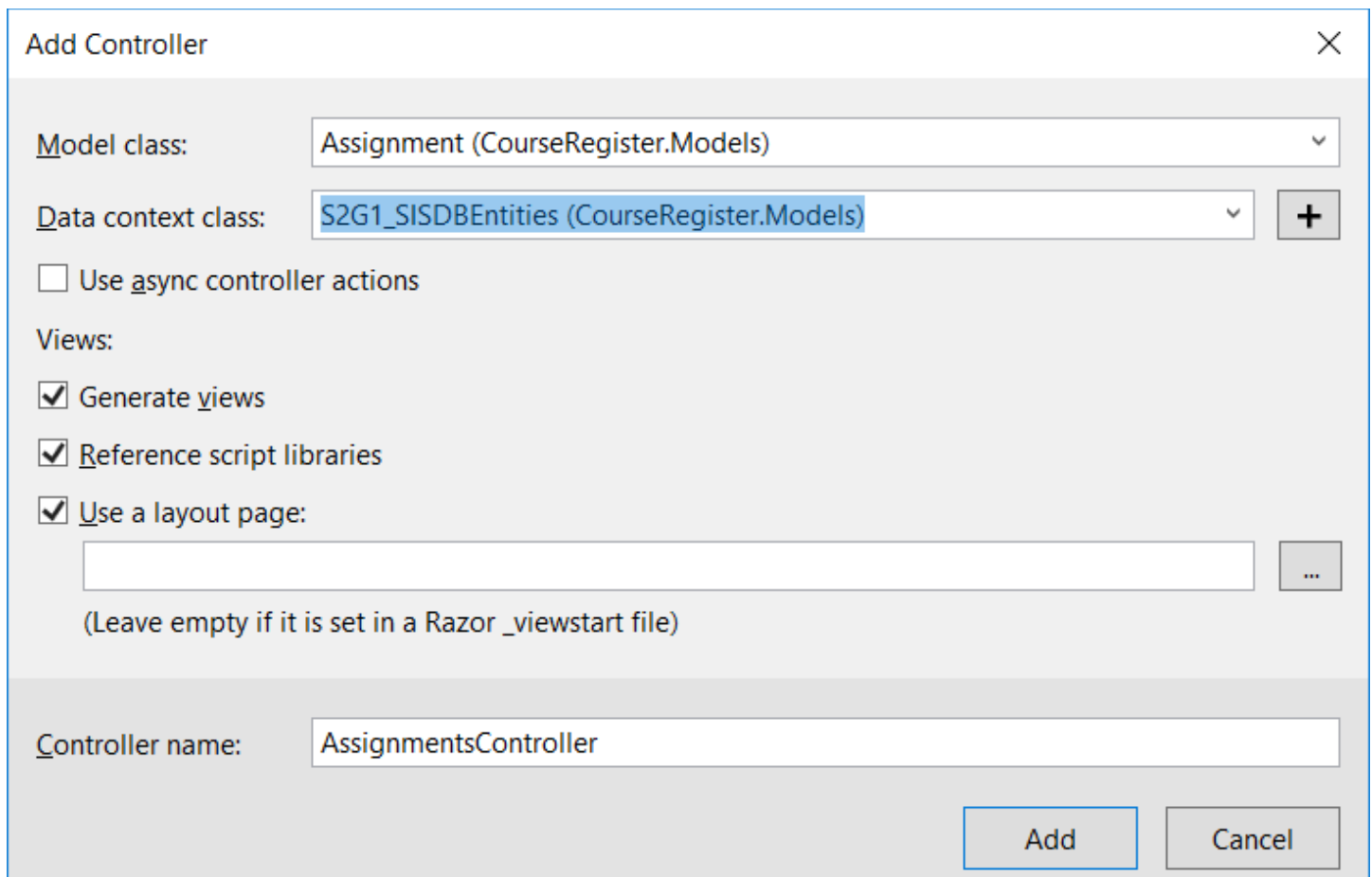
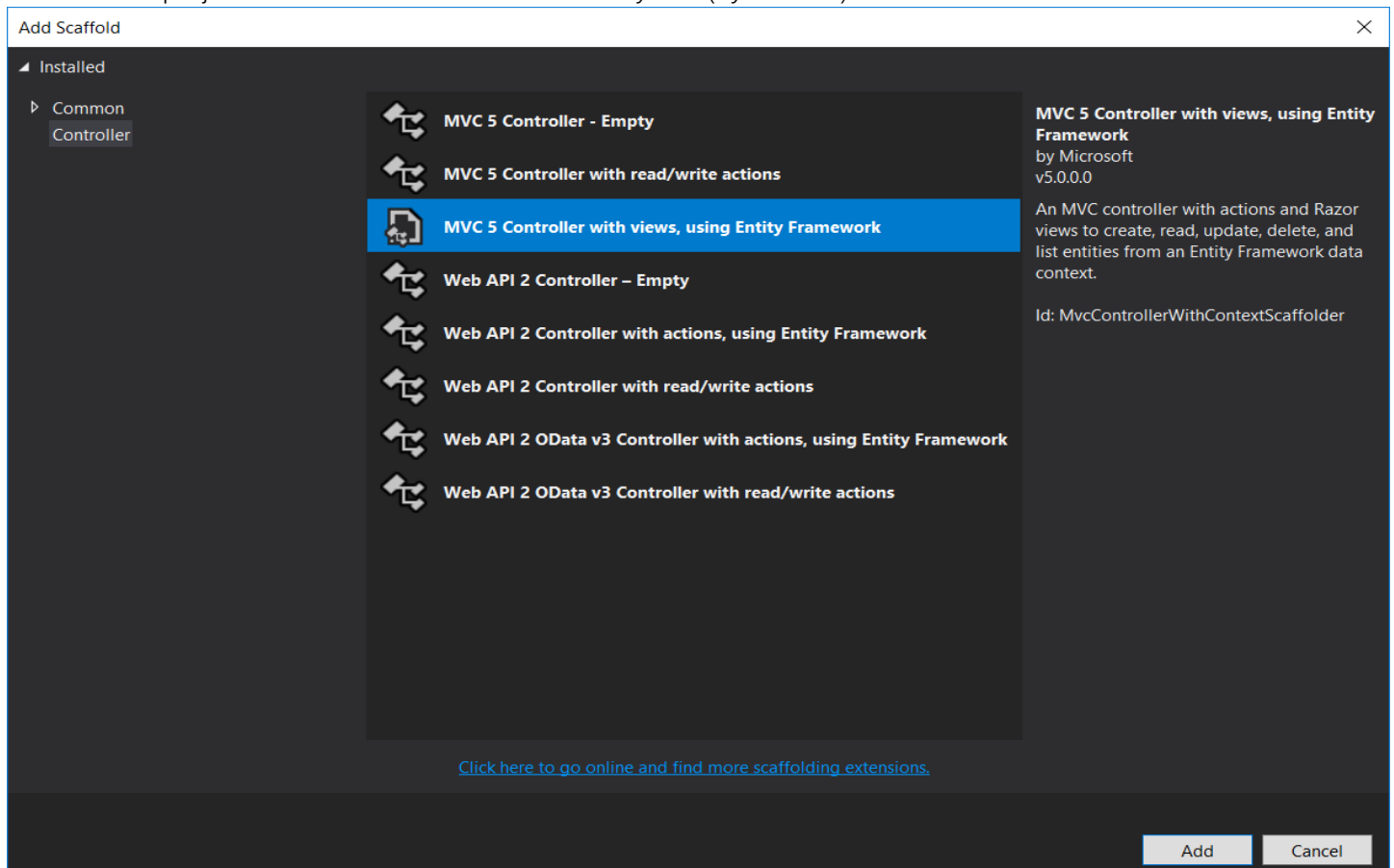
Add also, a connection string is generated in Web.config file:

```

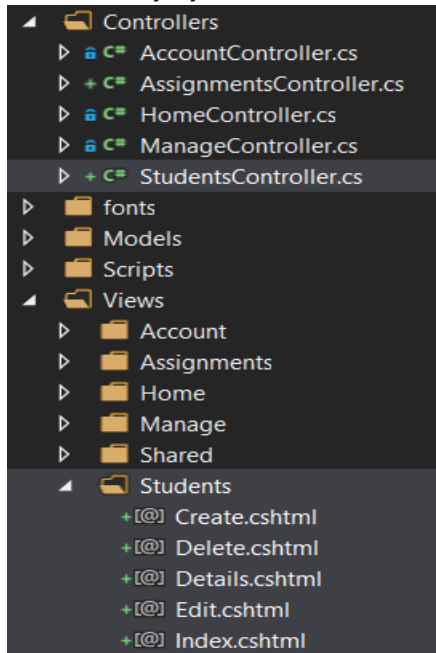
<add name="S2G1_SISDBEntities"
connectionString="metadata=res://*/Models.Model1.csdl|res://*/Models.Model1.ssdl|res://*/Models.Model1.msl;provider=System.Data.SqlClient;provider connection string="data source=LAPTOP-801LEG5;initial catalog=S2G1_SISDB;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework";"
providerName="System.Data.EntityClient" />

```

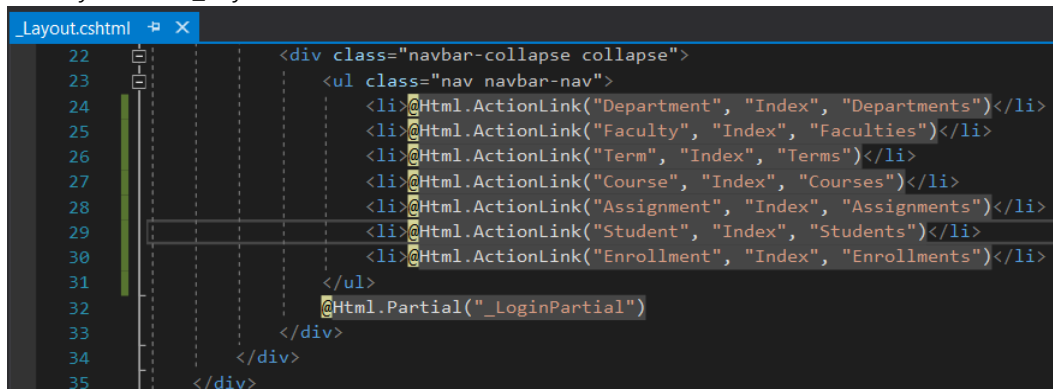
5. Build the project and then create controllers one by one (by scaffold):



**S2G1\_SISDBEntities** is the name of the connection string and **CourseRegister.Models** is the namespace (CourseRegister is the previous name of the project and solution). After this step, you can see the controller and its relevant views generated automatically by scaffold.



Modify Shared/\_Layout.cshtml and Home/Index.cshtml files.



6. Run application:

## Student Information System(SIS)

To be exact, SIS is a course register system constructed via ASP.NET MVC scaffold.

[Source Code »](#)

[Department](#)   [Faculty](#)   [Term](#)   [Course](#)  
[Assignment](#)   [Student](#)   [Enrollment](#)

© 2019 - SIS By S2G1

### 7. Remedy

7.1 Change the date type from intrinsic one to smalldatetime:

```
using System.ComponentModel.DataAnnotations;
```

```
[DataType(DataType.Date)]
[DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}", ApplyFormatInEditMode = true)]
0 references | 0 changes | 0 authors, 0 changes
public Nullable<System.DateTime> startDate { get; set; }
```

Start Date	End Date	Reading Week Start Date	Final Exam Start Date
2018-09-03	2018-12-19	2018-10-08	2018-12-16
2019-01-04	2019-04-19	2019-02-16	2019-04-15

7.2 Change the column name displayed:

```
[Display(Name = "Student's First Name")]
0 references | 0 changes | 0 authors, 0 changes
public string firstName { get; set; }
```

Faculty's First Name	Faculty's Last Name	Faculty's Email	Name of Department
Si	Li	si.li@uwindsor.ca	Computer Science
San	Zhang	san.zhang@uwindsor.ca	Computer Science

7.3 Attribute range

```
[Range(0, 100)]
[Display(Name = "Score")]
0 references | 0 changes | 0 authors, 0 changes
public int score { get; set; }
```

Score

The field Score must be between 0 and 100.

7.4 Foreign key: Department.id is the primary key and Faculty.departmentId is the foreign key referring to the Department

entity.


In Faculty.cs:

```
[ForeignKey("Department")]
[Display(Name = "Department Id")]
3 references | 0 changes | 0 authors, 0 changes
public int departmentId { get; set; }
```

```
1 reference | 0 changes | 0 authors, 0 changes
public virtual Department Department { get; set; }
```

In FacultiesController.cs

```
// GET: Faculties/Create
0 references | 0 changes | 0 authors, 0 changes
public ActionResult Create()
{
    ViewBag.departmentId = new SelectList(db.Departments, "id", "name");
    return View();
}
```

 **SelectList**.SelectList(System.Collections.IEnumerable items, string dataValueField, string dataTextField) (+ 8 overloads)  
Initializes a new instance of the **SelectList** class by using the specified items for the list, the data value field, and the data text field.

That is why in the create web page you can see the label departmentId while the value list shows you the name list rather than the department id list.

## Create

Faculty

id	<input type="text"/>
Faculty's First Name	<input type="text"/>
Faculty's Last Name	<input type="text"/>
Faculty's Email	<input type="text"/>
departmentId	<div>Computer Science ▼</div>
<div>Create</div>	

## Index

[Create New](#)

Faculty's First Name	Faculty's Last Name	Faculty's Email	Name of Department	
Si	Li	si.li@uwindsor.ca	Computer Science	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
San	Zhang	san.zhang@uwindsor.ca	Computer Science	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Wu	Wang	wu.wang@uwindsor.ca	Master of Business Administration	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

While in fact the attribute the action inserts into database is not the value of department name, but right the department id. It is a little bit tricky but meaningful. If you really want to show value list of department id on web page, just modify the third parameter in the SelectList function above.

```
// POST: Faculties/Create ...
[HttpPost]
[ValidateAntiForgeryToken]
0 references | 0 changes | 0 authors, 0 changes
public ActionResult Create([Bind(Include = "id,firstName,lastName,email,departmentId")] Faculty faculty)
{
    if (ModelState.IsValid)
    {
        db.Faculties.Add(faculty);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    ViewBag.departmentId = new SelectList(db.Departments, "id", "name", faculty.departmentId);
    return View(faculty);
}
```

SQLQuery9.sql - LA...OILEGB5\tians (57))\*

```
select * from Faculty
```

100 %

Results Messages

	id	firstName	lastName	email	departmentId
1	F-001	Si	Li	si.li@uwindsor.ca	1
2	F-002	San	Zhang	san.zhang@uwindsor.ca	1
3	F-003	Wu	Wang	wu.wang@uwindsor.ca	2

7.5 If the database table does not contain an attribute called id or className+Id (for e.g. StudentId), scaffold may fail to generate entity class. To walk around, a [key] annotation should be added in front of the primary non-id attribute.

```
[Key]
[ForeignKey("Instructor")]
public int InstructorID { get; set; }
```

## 8. Share it onto Github

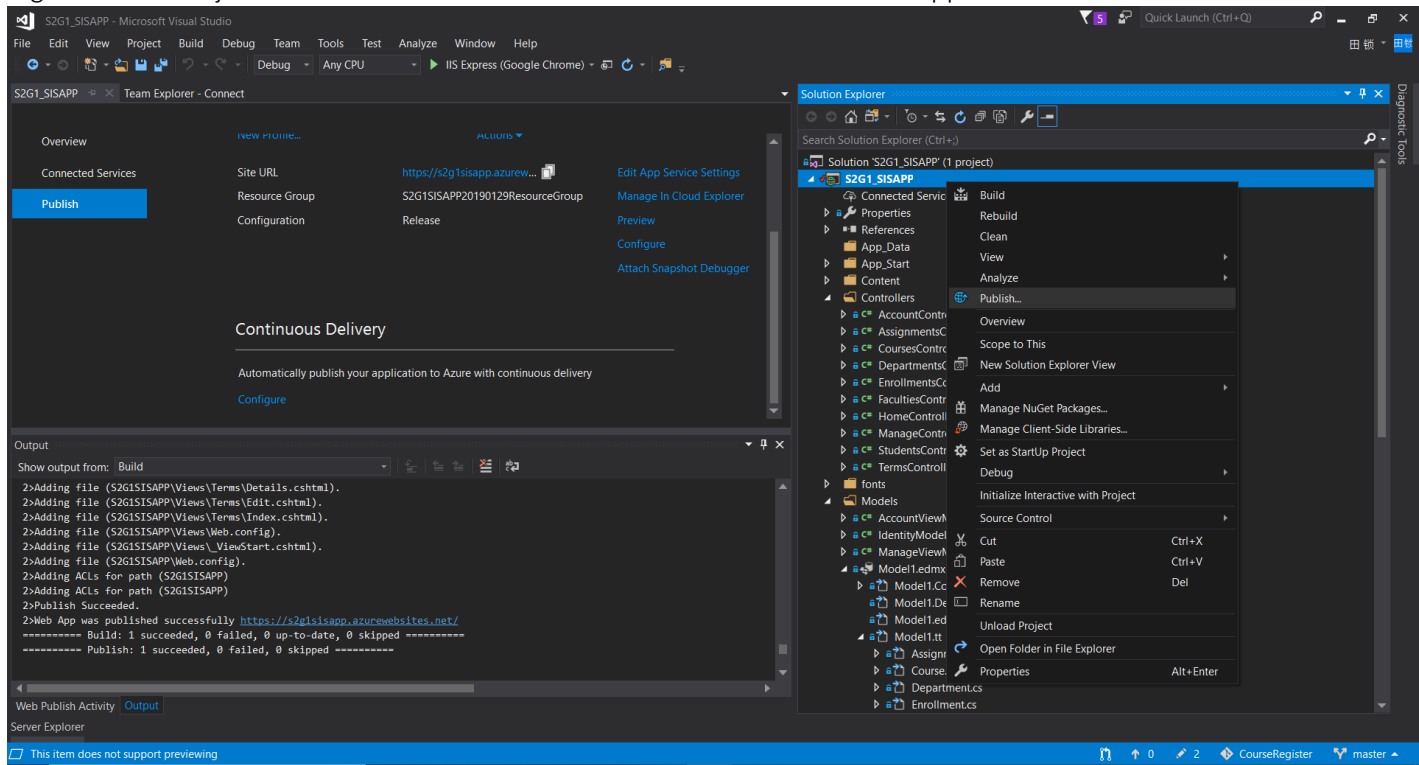
Download Github Extension for Visual Studio and install it. VS->View->Team Explorer->Connect. Publish to Github and then Push, Pull, etc

The screenshot displays the Visual Studio IDE with the Github extension installed. The Team Explorer on the left shows the 'courseRegisterSystem' repository connected to a local path. The Solution Explorer on the right shows the project structure with files like Course.cs, Department.cs, Enrollment.cs, Faculty.cs, Model1.cs, Student.cs, and Term.cs. The Output window at the bottom shows the progress of resolving deltas for the repository.

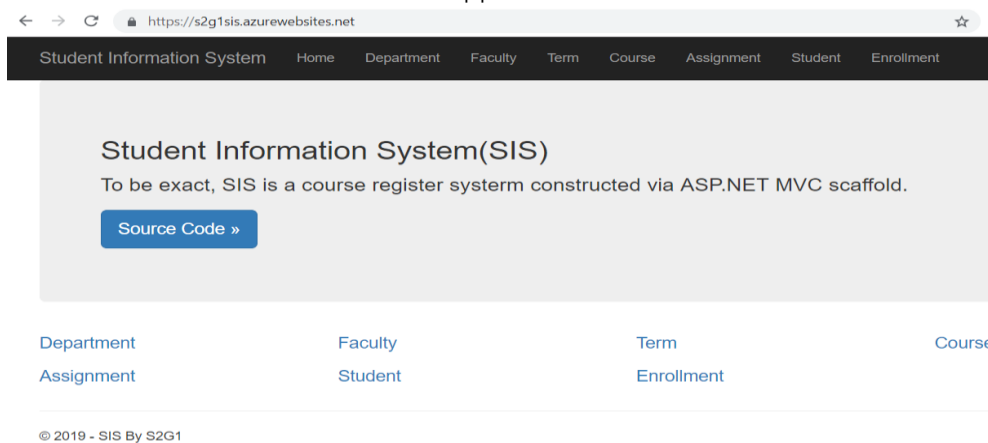


## 9. Launch the application to Azure

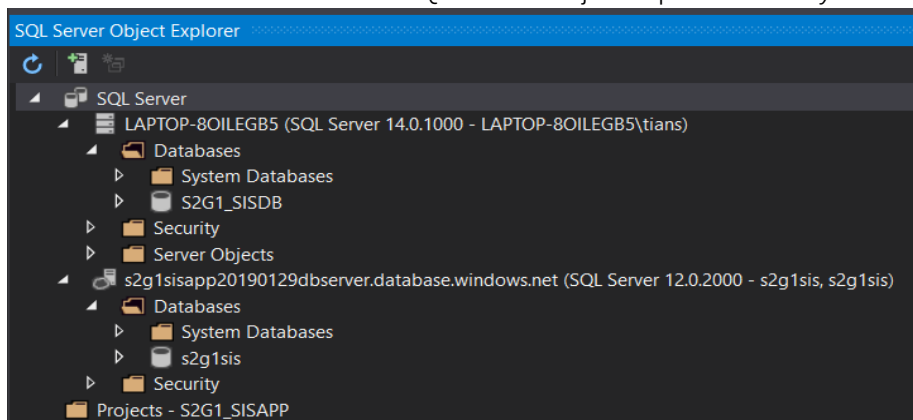
Right click on Project name->Publish->Create a free Azure account->Web app+DB



A web page with an Azure url will pop up. It is cool, but you cannot access any further functions until now, because of the connection issue between Azure app and Azure DB.



Go back to Visual Studio: View->SQL Server Object Explorer where you can edit local DB and Azure DB as well.



If you get error about firewall rule, go back to Azure portal authorizing your working ip.

Home > SQL databases > s2g1sis > Firewall settings

## Firewall settings

s2g1sisapp20190129dbserver (SQL server)

Save Discard Add client IP

Connections from the IPs specified below provides access to all the databases in s2g1sisapp20190129dbserver.

Allow access to Azure services

ON OFF

Client IP address 24.57.19.96

RULE NAME	START IP	END IP	
			...
ClientIPAddress_2019-1-29_...	24.57.19.96	24.57.19.96	...

Right click on s2g1sis/Tables->Add New Table, paste the content of the .sql script file generated from your local DB, and then you can see the newly created tables in your Azure DB.

s2g1sisapp20190129dbserver.database.windows.net (SQL Server 12.0.2000 - s2g1sis, s2g1sis)

- Databases
  - System Databases
  - s2g1sis
    - Tables
      - System Tables
      - FileTables
      - External Tables
      - dbo.Assignment
      - dbo.Course
      - dbo.Department
      - dbo.Enrollment
      - dbo.Faculty
      - dbo.Student
      - dbo.Term

Add New Table... Refresh

Copy the connection string from Azure DB portal and put it into VS: Publish->Configure. Do not forget to substitute the username and password with the real values you have created in the Azure DB setup wizard.

### s2g1sis - Connection strings

SQL database

Search (Ctrl+ /)

Diagnose and solve problems Quick start Query editor (preview)

Settings

- Configure
- Geo-Replication
- Connection strings

ADO.NET JDBC ODBC PHP

ADO.NET (SQL authentication)

```
Server=tcp:s2g1sisapp20190129dbserver.database.windows.net,1433;Initial Catalog=s2g1sis;Persist Security Info=False;User ID=(your_username);Password=(your_password);MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;
```

Download ADO.NET driver for SQL server

Publish

**Publish**

Connection

**Settings**

**s2g1sis - Web Deploy \***

Configuration: Release

File Publish Options

Databases

ApplicationDbContext

Server=tcp:s2g1sisapp20190129dbserver.database.windows.net,1433;Initial ...

☒ Use this connection string at runtime (update destination web.config)

☐ Execute Code First Migrations (runs on application start)

S2G1\_SISDBEntities

Server=tcp:s2g1sisapp20190129dbserver.database.windows.net,1433;Initial ...

< Prev Next > Save Cancel

Now you can operate your Azure DB with the CRUD actions.

← → ↻ https://s2g1sis.azurewebsites.net/Faculties ☆

Student Information System Home Department Faculty Term Course Assignment Student Enrollment Register

## Index

[Create New](#)

Faculty's First Name	Faculty's Last Name	Faculty's Email	Name of Department	
Sufei	Mas	sufei@uwindsor.ca	Computer Science	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Julia	Roberts	julia.roberts@uwindsor.ca	MBA	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

© 2019 - SIS By S2G1

## 10. Reference

<https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/getting-started-with-ef-using-mvc/creating-a-more-complex-data-model-for-an-asp-net-mvc-application>

<https://docs.microsoft.com/en-ca/azure/app-service/app-service-web-get-started-dotnet-framework>

<https://docs.microsoft.com/en-ca/azure/app-service/app-service-web-tutorial-dotnet-sqldatabase#publish-to-azure-with-sql-database>