

Programming 3 Assignment 2

Assignment Overview

- The specifications associated with this assignment will incorporate the Model View Controller design pattern and specifically ASP.NET MVC to continue the process of building a CRUD application along with a relational database. This assignment will focus on the Controller and View components of the MVC pattern.
- Work associated with this assignment will reinforce the student's learning in the following areas:
 - Controllers
 - Views
 - cshtml formatting
 - Viewbag dynamic property

Evaluation

- The evaluation of this assignment will be in the form of a Code Review and Run Time Demonstration.
- During the Code Review evaluation will be based on:
 - Comments
 - Documentation
 - Adherence to Standards
 - Coding choices (i.e. refactoring, no repeated code, readability etc...)
 - Implementation of Requirements
 - Verbal discussion of code
- During the Run-Time Demonstration evaluation will be based on:
 - A variety of run-time tests

Programming 3 Assignment 2

Specifications

MVC Project

- Continue with the ASP.NET MVC project called **BITCollege_FL** based on the following specifications:

Home Controller

- Modify the **About** ActionResult method by changing the ViewBag.Message value to “BIT College Data Maintenance System”.

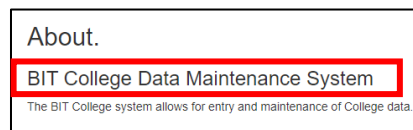


Figure 1

_Layout.cshtml

- Modify the _Layout.cshtml file found in the Views/Shared folder as follows:
 - Modify the ActionLink in the navbar-header class such that the first link appears as “BIT College”
 - Modify the footer such that the timestamp is followed by “BITCollege_FL” where FL are replaced by your first and last initials

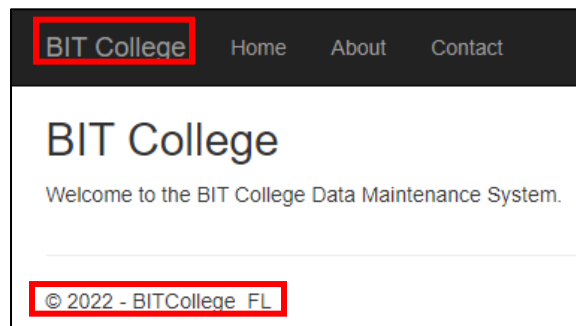


Figure 2

Index.cshtml

- Modify the content in Index.cshtml file in the Home views folder based on the following figure:
 - Note:** In the figure below, the text “BIT College” has been modified to use <h2>

Programming 3 Assignment 2

- **Note:** The figure below contains the entire content to be displayed in the Index page. Please delete any of the remaining content.

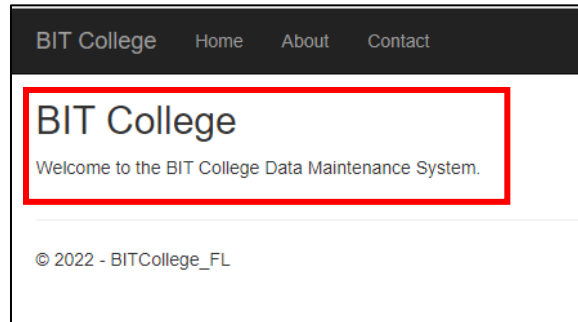


Figure 3

About.cshtml

- Modify the content in About.cshtml file in the Home views folder based on the following figure:

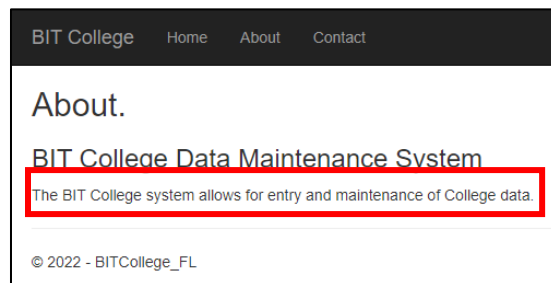


Figure 4

Contact.cshtml

- Modify the Contact.cshtml file in the Home views folder based on the following figure:
 - **Note:** The figure below contains the entire content to be displayed in the Index page. Please delete any of the remaining content.

Programming 3 Assignment 2

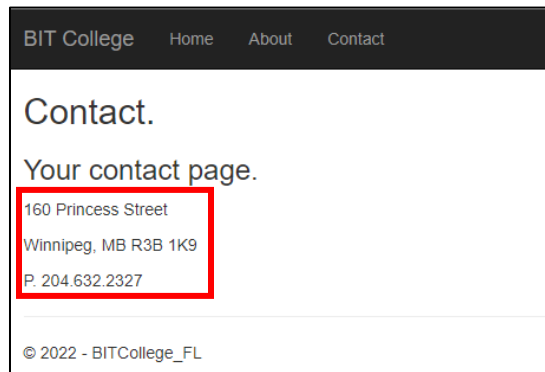


Figure 5

BITCollege Controllers and Views

- Before proceeding, ensure that all feedback has been addressed from the first assignment. This includes any spelling errors in the property names.
- Define Controllers based on each of the Models defined in the first assignment.
- The Controllers must use the Scaffolding Option which will generate *MVC 5 Controller with views, using Entity Framework*
 - The name of the Data Context Class generated should correspond with the name of the MVC project (e.g. **BITCollege_FLContext**)
- When creating Controllers for Abstract Classes, do not generate views
- When Creating Controllers/Views or when running the MVC application, if an error occurs such as the one shown below, a migration is necessary
 - Refer to the **Code First Migrations** topic in the Course Notes – or seek assistance from your instructor or an EA to proceed with your migration

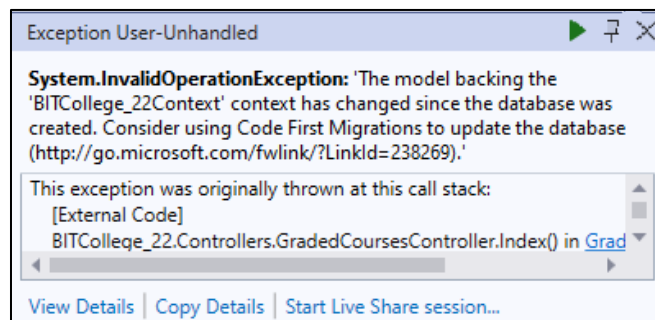


Figure 6

- **Upon creating the first controller**, modify the Web.Config file based on the instructions provided in the Course Notes.
 - Refer to the **Design Patterns – MVC – View Controller** topic in the Course Notes.

Programming 3 Assignment 2

Sub Class Controllers

- In your Data Context class, DbSet entries will not be generated automatically for the sub classes of the Abstract Classes. As you create the controllers for these sub classes, update your Data Context Class (found in the Data folder), by adding entries for each of the sub class models using the following format:

```
public System.Data.Entity.DbSet<Namespace.Models.ModelName> PluralizedModelName { get; set; }
```

Code 1

- The Controllers for the Course and GradePointState sub classes will need to be further modified such that all database queries are based on the appropriate sub class.
 - In most cases, the queries within the controller action methods will result in a syntax error prompting you to clarify the subtype in the expression, similar to the syntax error shown below:

```
AuditCourse auditCourse = db.Courses.Find(id);
```

Code 2

- In some cases, the error won't reveal itself until runtime. In those cases, you will see a run-time error message similar to the one shown below:

```
The model item passed into the dictionary is of type  
'System.Collections.Generic.List`1['  
'System.Collections.Generic.IEnumerable`1['  
]', but this dictionary requires a model item of type  
'System.Collections.Generic.IEnumerable`1['
```

Figure 7

- In both cases, the error can be addressed by either changing the query to explicitly use the pluralized subtype shown in first line of code below, or by casting to the subtype within the query shown in the second line of code below.

```
AuditCourse auditCourse = db.AuditCourses.Find(id);  
  
AuditCourse auditCourse = (AuditCourse)db.Courses.Find(id);
```

Code 3

Programming 3 Assignment 2

Controller and View Customization

Note

- In the figures that follow, data is displayed to show the results of the customizations that follow. Data entry will take place during the final step of this assignment; however, data can be entered now to assist with customization, as the views will not populate without any data.
- The derived Description properties in GradePointState and Course will require modification at this point. Use methods of the string class to further modify GetType().Name as such that it displays the same as the figures below
- There might be slight differences in the styling of the cshtml pages. For example, the dropdown arrows, or Create button may appear differently. This is not a concern as it's related to either system settings, or slightly different Visual Studio versions.

Controller Customization

StudentsController

- As necessary, modify the StudentsController such that the Create AND Edit views will display foreign key fields as selectlists.
 - The display value for the AcademicProgramId foreign key is ProgramAcronym.
 - The display value for the GradePointStateId foreign key is Description.
 - The figures below show the selectlists for the Create views.

The figure shows two side-by-side screenshots of a web form titled 'Create Student'. The left screenshot shows the 'GradePointStateId' dropdown menu open, displaying a list with 'Suspended' selected. The right screenshot shows the 'AcademicProgramId' dropdown menu open, displaying a list with 'PMM' selected. Both forms include fields for 'Student Number' and 'First Name'.

Figure 8

RegistrationsController

- As necessary, modify the RegistrationsController such that the Create AND Edit views will display foreign key fields as selectlists.

Programming 3 Assignment 2

- The display value for the StudentId foreign key is FullName.
- The display value for the CourseId foreign key is Title.
- The figures below show the selectlists for the Create views.

The figure shows two side-by-side screenshots of a web form titled 'Create Registration'. The left screenshot shows the 'StudentId' dropdown menu open, displaying a list of names: 'Zelda Zimmerman' (selected), 'Zelda Zimmerman', and 'Yancy Yale'. Below it is a 'Registration Number' text box and a 'Date' label. The right screenshot shows the 'CourseId' dropdown menu open, displaying a list of course titles: 'Science' (selected), 'Science', 'Ovens', and 'General Safety'. Below it is a 'Date' label.

Figure 9

AcademicProgramsController

Discussion

- The default foreign key behaviour dictates that cascading deletions will take place such that dependent records will be automatically deleted when the parent record is deleted in the database.
- This is not the desired behaviour in the AcademicProgram – Student relationship or the AcademicProgram – Course relationship. That is, when a program is deleted, we don't want the related Student or Course records to be deleted – yet we want all the remaining benefits of a defined Foreign Key relationship.
- There is no annotation that can be used to prevent these cascading deletions from occurring. Instead, the Program controller must be modified to detach the related child records from the AcademicProgram being deleted, thus preserving the child records.

Requirement

- Modify the [HttpPost] DeleteConfirmed ActionResult method of the AcademicProgramsController as follows:
 - Following the code that identifies the record for deletion (AcademicProgram object), but prior to the actual delete (Remove), use the Clear method to clear

Programming 3 Assignment 2

the relationship between the AcademicProgram object to be deleted and the corresponding Student and Course records.

```
AcademicProgram academicProgram = db.AcademicPrograms.Find(id);

/** INSERT THE CLEAR CODE HERE
academicProgram.Student.Clear();
academicProgram.Course.Clear();
/**

db.AcademicPrograms.Remove(academicProgram);
```

Code 4

View Customization

Student

- As necessary, modify the Student **Index** view by removing/rearranging fields such that it displays as shown in the figure below
 - Ensure the FullName and FullAddress display instead of the separate fields that make up FullName and FullAddress
 - Ensure the ProgramAcronym and GradePointState Description are displayed as shown.

Index					
Create New					
Student Number	Name	Address	Program	State	Grade Point Average
10000000	Zelda Zimmerman	160 Princess Street Winnipeg MB	VT	Suspended	Edit Details Delete
10000001	Yancy Yale	555 Main Street Winnipeg MB	PBP	Suspended	Edit Details Delete
© 2022 - BITCollege_FL					

Figure 10

Registration

- As necessary, modify the Registration Index View to display as shown in the figure below
 - Ensure the Course's Title and the Student's FullName are displayed as shown

Programming 3 Assignment 2

Index						
Create New						
Title	Name	Registration Number	Date	Grade	Notes	
General Safety	Zelda	100	2022-01-01	Ungraded	Online Registration	Edit Details Delete
Ovens	Yancy	101	2022-01-01	Ungraded	Online Registration	Edit Details Delete
© 2022 - BITCollege_FL						

Figure 11

BIT College Views

Site.css

- Modify the Site.css file found in the Content folder as necessary to meet the following requirements:
 - Refer to **Site.css** discussion in the **Module 02 Design Patterns - MVC - View Controller** topic in the Course Notes.
 - Define an Id which is to be applied to Index View Table Headers and indicate that white space (or \n) should be interpreted as a line feed.
 - Apply text-align: right formatting to **all Model Fields** that have been defined as double, or double? data type.
 - The figure below shows the right aligned double fields in the GradedCourse Edit View:

Edit

GradedCourse

AcademicProgramId

PMM

Course Number

G-100000

Title

Science

Credit Hours

5

Tuition

500

Notes

PMM Required

Assignments

70.00%

Exams

30.00%

Save

[Back to List](#)

© 2022 - BITCollege_FL

Figure 12

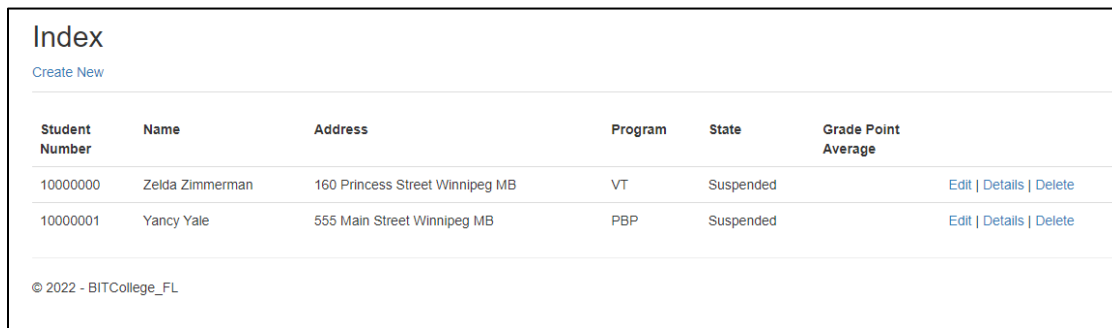
- **Note:** this right-alignment will only appear in the Create and Edit views. There is no right-alignment requirement for the other views.

Programming 3 Assignment 2

- **Note:** In the screenshot above, the percent and currency formatting will apply only if the DisplayFormat annotation has been applied to editing as well
 - **Optional:** To carry display formatting to the fields meant for editing, modify the DisplayFormat annotation to use the **ApplyFormatInEditMode = true** overload

Index Views

- Modify the following Index Views by applying the line feed **Id** defined in the **Site.css** to the <tr> tag associated with the headings :
 - AuditCourse Index.cshtml
 - GradedCourse Index.cshtml
 - MasteryCourse Index.cshtml
 - Registration Index.cshtml
 - Student Index.cshtml
- Below is the Student Index View showing the line feeds applied to the headings:



The screenshot shows a web page titled 'Index' with a 'Create New' link. Below is a table with columns: Student Number, Name, Address, Program, State, and Grade Point Average. There are two rows of student data. At the bottom, there is a copyright notice: © 2022 - BITCollege_FL.

Student Number	Name	Address	Program	State	Grade Point Average
10000000	Zelda Zimmerman	160 Princess Street Winnipeg MB	VT	Suspended	Edit Details Delete
10000001	Yancy Yale	555 Main Street Winnipeg MB	PBP	Suspended	Edit Details Delete

© 2022 - BITCollege_FL

Figure 13

Data Entry

- Populate the generated database by using the Create Views for each of the Models using the following data:

GPA State Subtypes

- The GPAState Subtypes will eventually (next assignment) implement the Singleton Pattern. However, at this point at least one record is needed in the GPAState table to proceed with the data entry that follows. For now, enter **only the following record** using the **SuspendedState controller and views**:

Programming 3 Assignment 2

SuspendedStates

Table 1

Lower Limit	Upper Limit	Tuition Rate Factor
0.00	1.00	1.1

AcademicPrograms

Table 2

Program Acronym	Description
PMM	Precision Metal Machining
VT	Veterinary Technology
PBP	Professional Baking and Patisserie
CA	Culinary Arts
ECE	Early Childhood Educator
CP	Carpentry

Students

Table 3

Grade Point State	Academic Program	Student Number	First Name	Last Name	Address	City	Province
Suspended	VT	10000000	Zelda	Zimmerman	160 Princess Street	Winnipeg	MB
Suspended	PBP	10000001	Yancy	Yale	555 Main Street	Winnipeg	MB

Student data continued...

Date Created	Grade Point Average	Fees	Notes
{today's date}	Null {leave blank}	\$0.00	New Student
{today's date}	Null	\$500.00	New Student

Course Subtypes

GradedCourses

Table 4

Academic Program	Course Number	Title	Credit Hours	Tuition Amount	Notes	Assignments	Exams
PMM	G-100000	Science	5	\$500.00	PMM Required	0.70	0.30

Programming 3 Assignment 2

MasteryCourses

Table 5

Academic Program	Course Number	Title	Credit Hours	Tuition Amount	Notes	Maximum Attempts
PBP	M-10000	Ovens	0	\$1500.00	Required to proceed	99

AuditCourses

Table 6

Academic Program	Course Number	Title	Credit Hours	Tuition Amount	Notes
VT	A-1000	General Safety	0	\$500.00	Recommended to proceed

Registrations

Table 7

Student	Course	Registration Number	Registration Date	Grade	Notes
Zelda Zimmerman	General Safety	100	{today's date}	Null {leave blank}	Online Registration
Yancy Yale	Ovens	101	{today's date}	Null {leave blank}	Online Registration

Test

- Test your solution to ensure it meets all of the requirements as defined above.

Backup

- Backup your solution.
- Create a backup of your database and its corresponding log file at this point
- The database and log file can be found in the following directory:
- C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA
 - **Note:** It may be necessary to stop the SQL Server Service to perform this backup.
 - **Note:** If you stop the service to perform the backup, make sure you restart the service.

Programming 3 Assignment 2

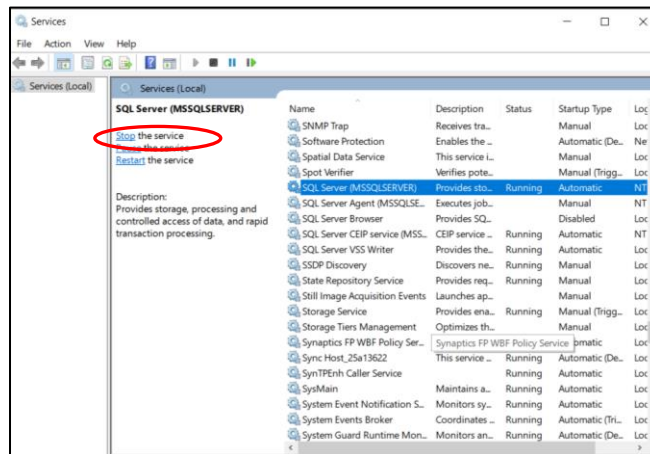


Figure 14

Evaluation: In Person/Hybrid Delivery

- This assignment must be evaluated on or before the due date/time specified in Learn.
 - The assignment is due at the beginning of class on the due date specified. Your instructor will evaluate the assignment on the students' machine in a random order. Remote evaluations will be based on a first come, first served basis.