Logo, company name

Description automatically generated

FIT5032: Internet Applications Development

Studio Assessment Task 1: Studio #3 & #4

Name: Edward Shen

Student Id: 30594863

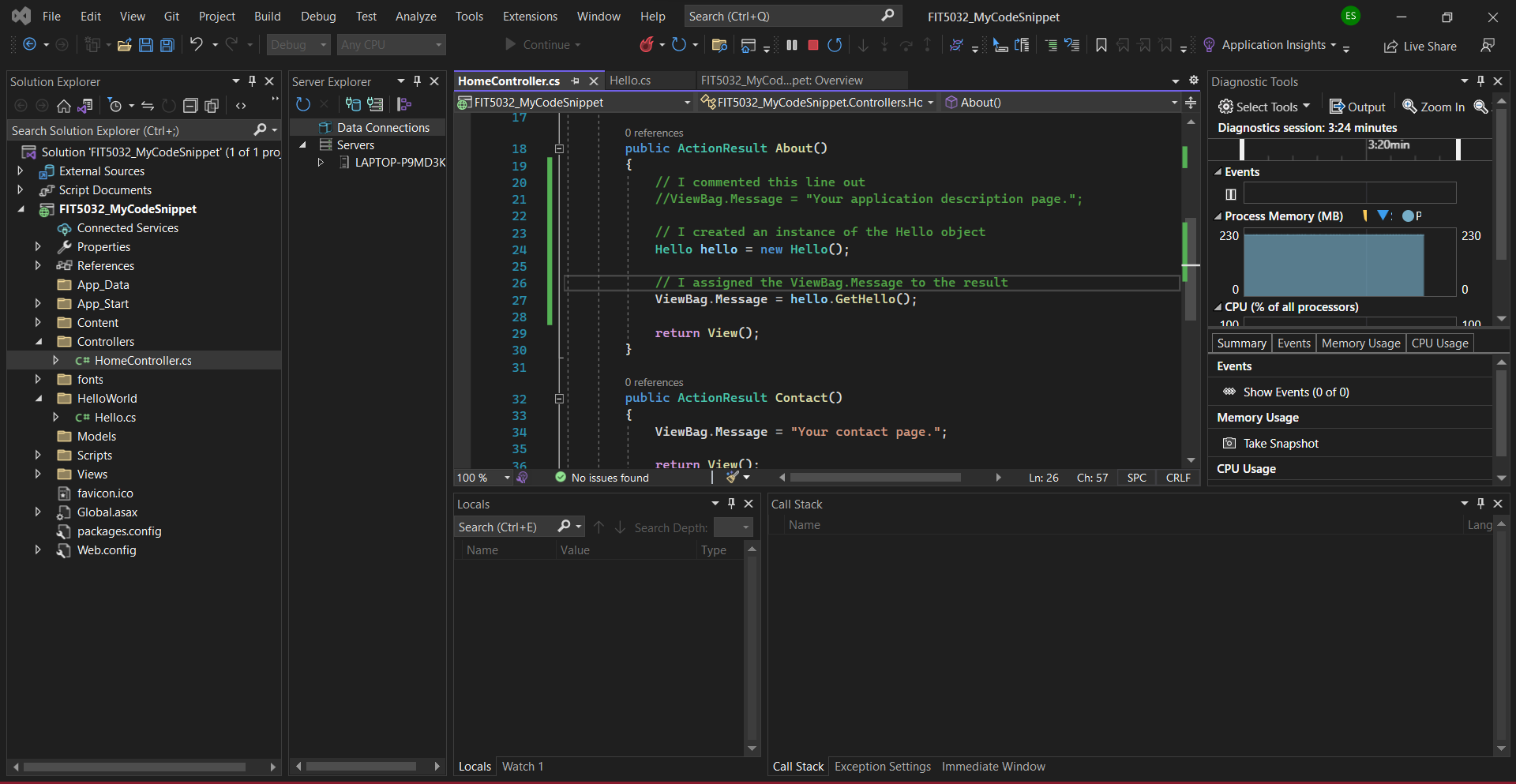
Date of submission: 21st August 2022

Self-Evaluation: High Distinction

Efolio Task 3.1:

Graphical user interface, text, application, email

Description automatically generated



*Code from IDE.*

Efolio Task 3.2:

Github Username: 3drdsh3in

Git Link: https://github.com/3drdsh3in/FIT5032

Task 4.1: Model First Development Screenshots ([Github](https://github.com/3drdsh3in/FIT5032/tree/master/lab4/FIT5032_MyModelFirst)):

Graphical user interface, application

Description automatically generated

Figure 1: Create.cshtml (Running in localhost)

Graphical user interface, application

Description automatically generated

Figure 2: Index.cshtml after entering the student fields (Running in localhost)

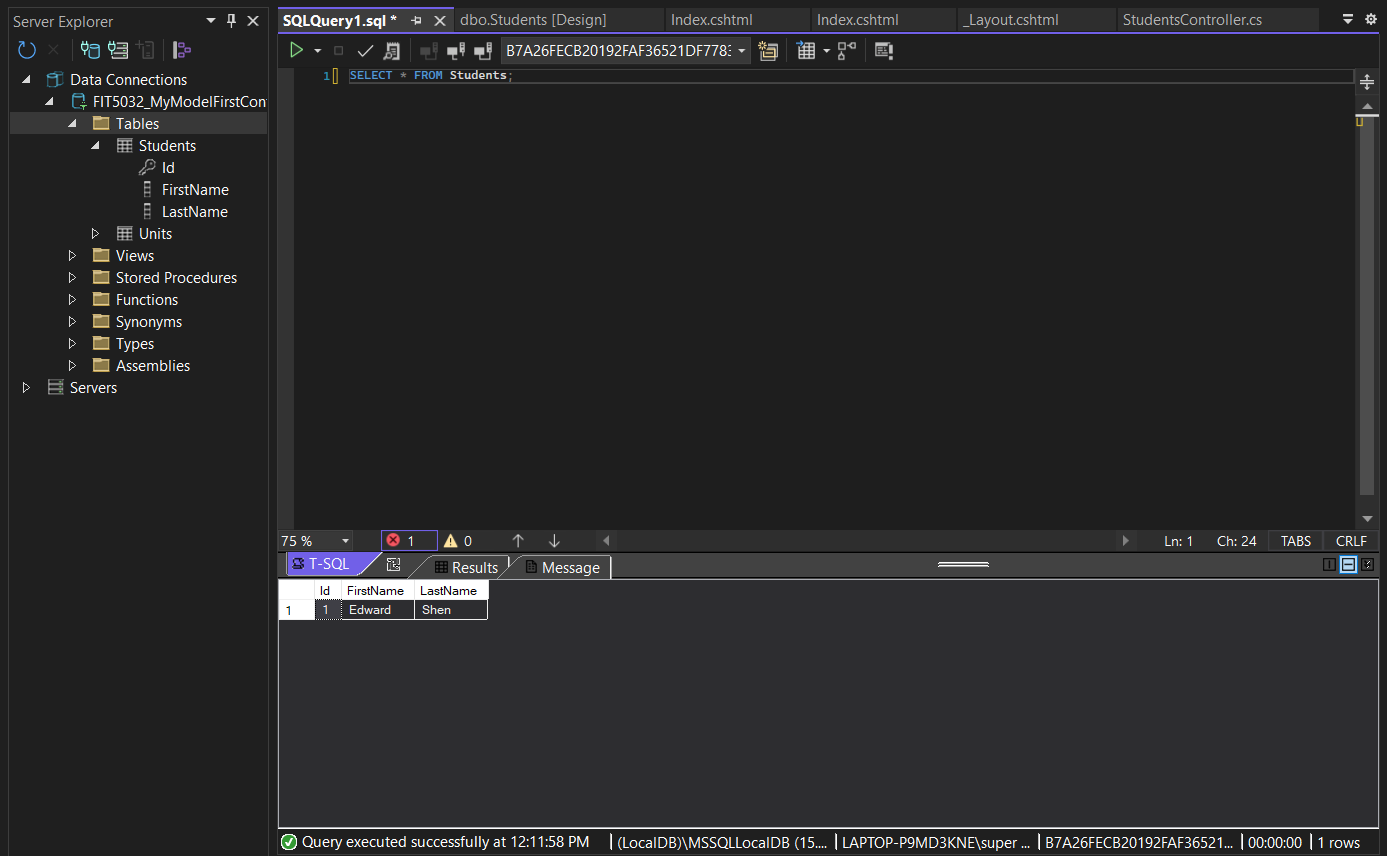


Figure 3: Querying the table after adding the student via the web application.

Text

Description automatically generated

Figure 4: Screenshot of Generated Student Schema. ([Github](https://github.com/3drdsh3in/FIT5032/blob/master/lab4/FIT5032_MyModelFirst/FIT5032_MyModelFirst/Models/FIT5032_MyModelFirst.edmx.sql))

Task 4.2: Code First Development Screenshots ([Github](https://github.com/3drdsh3in/FIT5032/tree/master/lab4/FIT5032_MyCodeFirst)):

Note that the controller and cshtml code I used for this is identical to Task 4.1. The only major difference being the way the Domain classes under /Models is written.

Text

Description automatically generated

Figure 5: Domain Class for Student.cs ([Github](https://github.com/3drdsh3in/FIT5032/blob/master/lab4/FIT5032_MyCodeFirst/FIT5032_MyCodeFirst/Models/Student.cs))

A screenshot of a computer

Description automatically generated

Figure 6: Domain Class for Unit.cs ([Github](https://github.com/3drdsh3in/FIT5032/blob/master/lab4/FIT5032_MyCodeFirst/FIT5032_MyCodeFirst/Models/Unit.cs))

The only major difference being the [Table(“table\_name”, Schema=”db\_name”)] annotation at the top along with the [ForeignKey(“fk\_name”)] annotation.

Note that the FIT5032\_CodeFirstContext.cs class was used to connect to the database ([Github](https://github.com/3drdsh3in/FIT5032/blob/master/lab4/FIT5032_MyCodeFirst/FIT5032_MyCodeFirst/Models/FIT5032_CodeFirstContext.cs)):

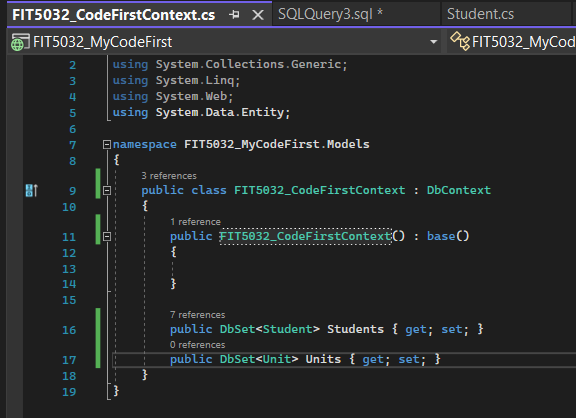


Figure 7: The DbContext subclass I am using to interact w/ SQL Server database.

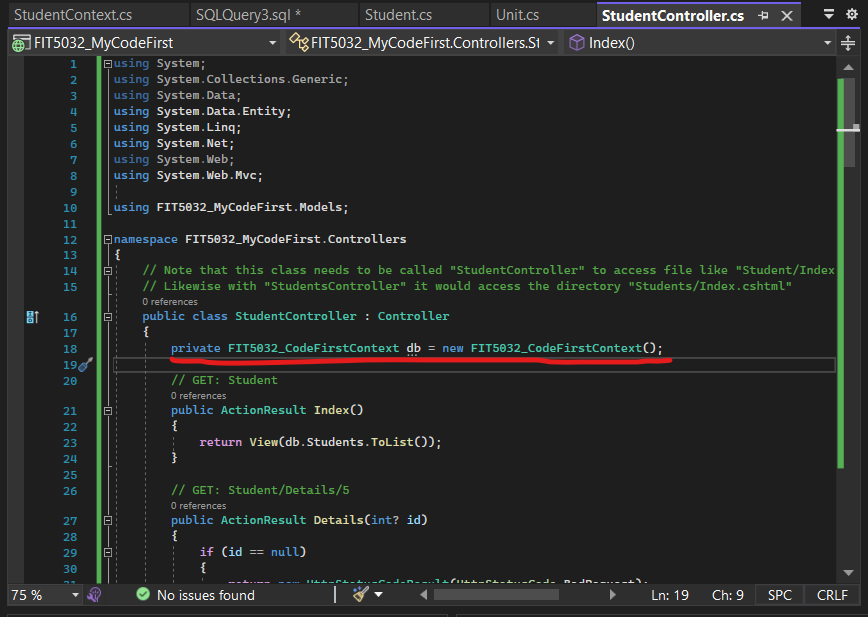


Figure 8: Dependency Injection of the Context class from the previous figure. Also note that the controller code is (basically) the same generated code from the model first approach (I couldn’t be bothered to rewrite it from scratch… but usually we would want to modify the code in the controller to adhere to business requirements… but this should be enough for a quick POC). (Github)

Demo of Entering A row into our SQL Server Database:

Graphical user interface, application

Description automatically generated

Figure 9: Entering it in via our web application:

Graphical user interface, text, application, email

Description automatically generated

Figure 10: It now shows on our Student/Index.cshtml page.

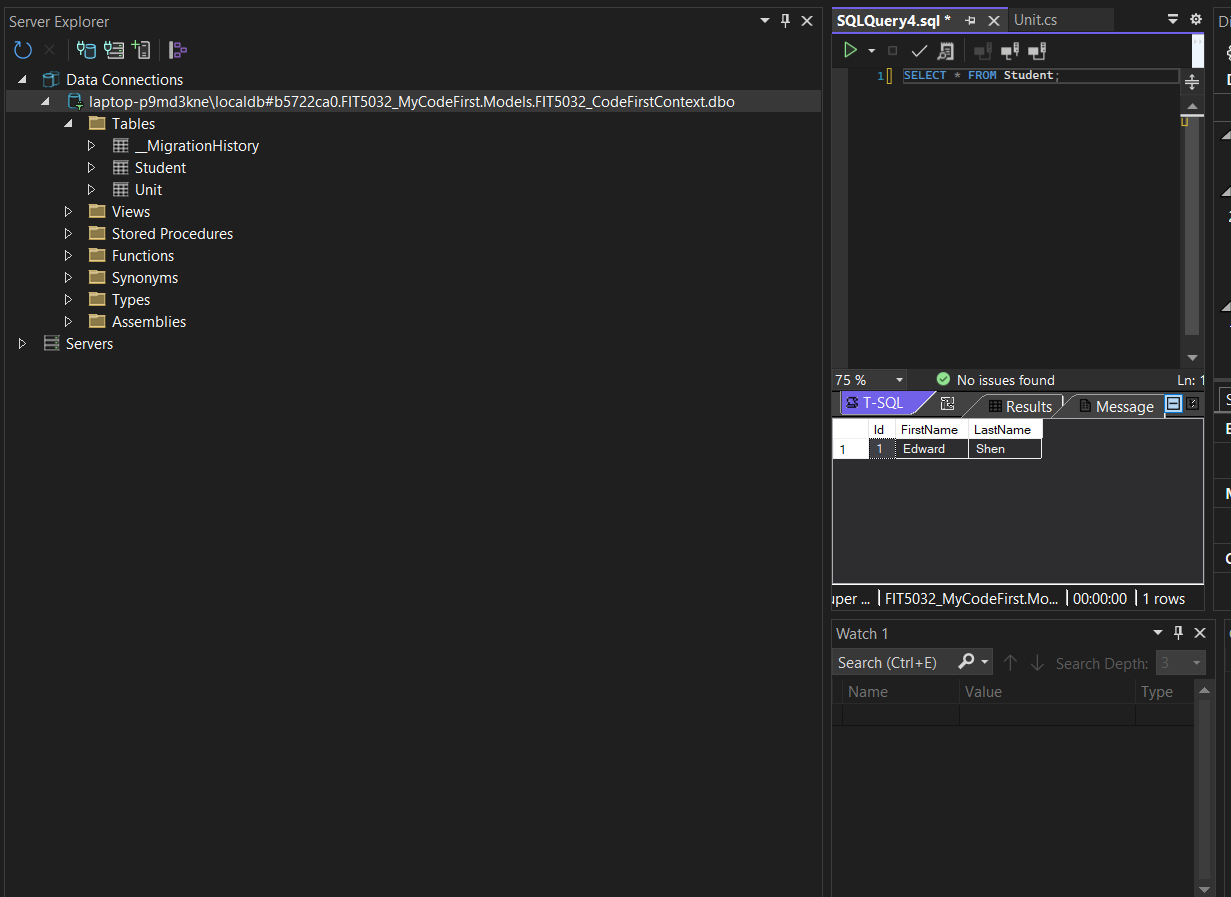


Figure 11: Directly Query the database that the context created to make sure it saved the record. (Note that the name of the data connection itself “FIT5032\_CodeFirstContext.dbo” which was specified in our Student.cs domain class earlier.)

Note I won’t do a demonstration for CRUD on the Unit.cs Domain class just to keep this short (I’m pretty sure this is enough to cover the efolio task anyway and it shouldn’t be too difficult to figure out since we’ve already done it with the student class).