MONASH University

FIT5032 - Internet Applications Development

Week 05E - Working with Rich Text - WYSIWYG Editor (TinyMCE)

Last updated: 5th Oct 2018
Author: Jian Liew

Introduction

There is no need to complete this tutorial. It functions as a supplementary material to showcase how to use a simple JavaScript API (TinyMCE) for the creation of Rich Text. (WYSIWYG) editor.

Working with Rich Text is a very common use case to allow users to mark up their post. For example, when a user would like to post a content, the user would like to mark up their content so that it is more presentable. There are quite a number of JS libraries that allow this feature to be completed, for example QuillJS or CKEditor. However, this tutorial aims to showcase how to use TinyMCE.

Objectives

Estimated Time To Complete - 30 mins

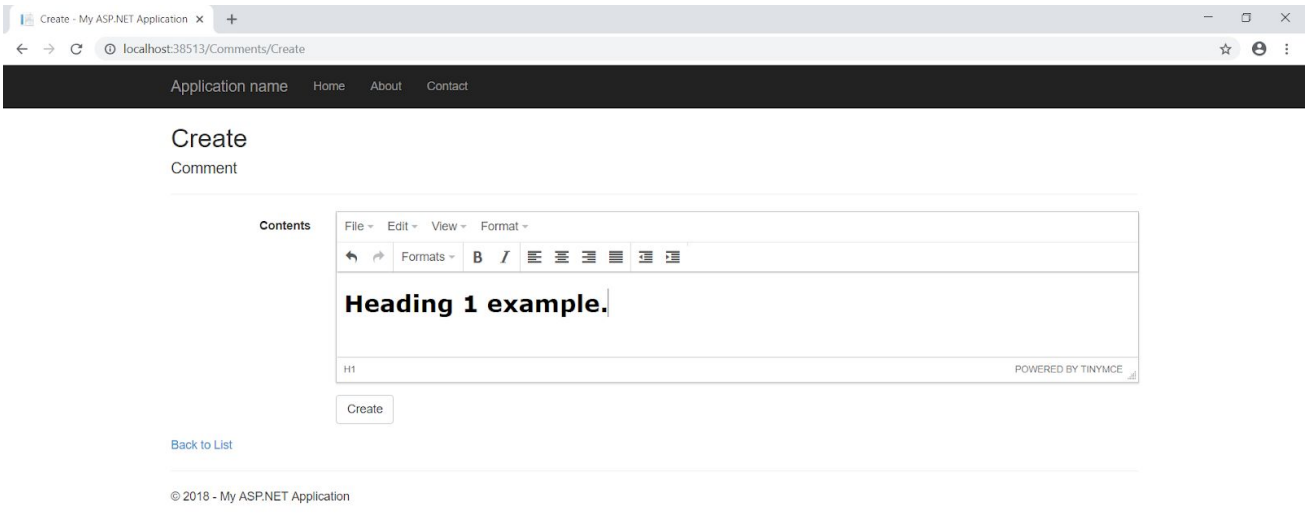
Upon the completion of this tutorial, you will gain a basic understanding of

- How to use TinyMCE at a basic level
- How to use show "raw" HTML with ASP.NET MVC.

DoubtFire Submission

- None

Upon the completion of this tutorial. The end result of your web application would look like the following.



This is a really basic example, on how you can use TinyMCE as your WYSIWYG editor.

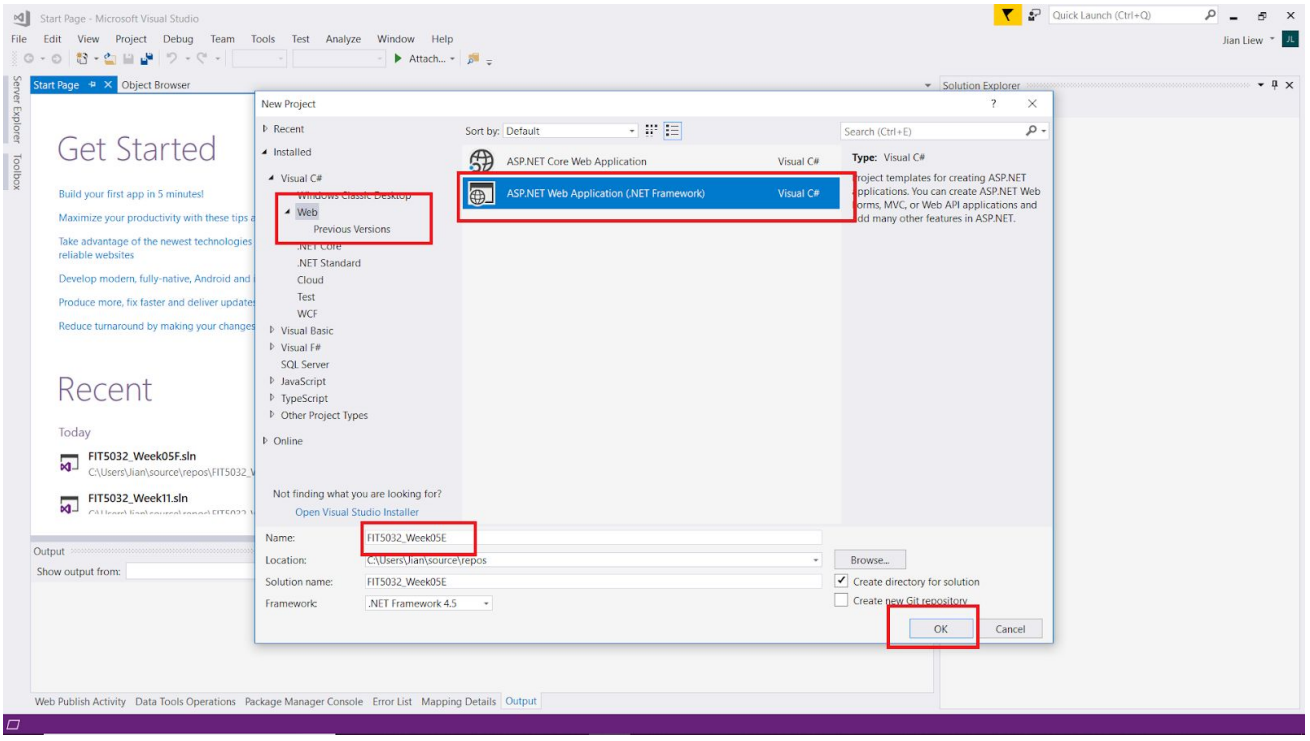
Certain features that are present upon the completion of this tutorials are

1. It uses a database first approach to create a single table with an attribute that is capable of storing values big enough for a text area.
2. It uses the scaffolding feature to create both the controllers and views.

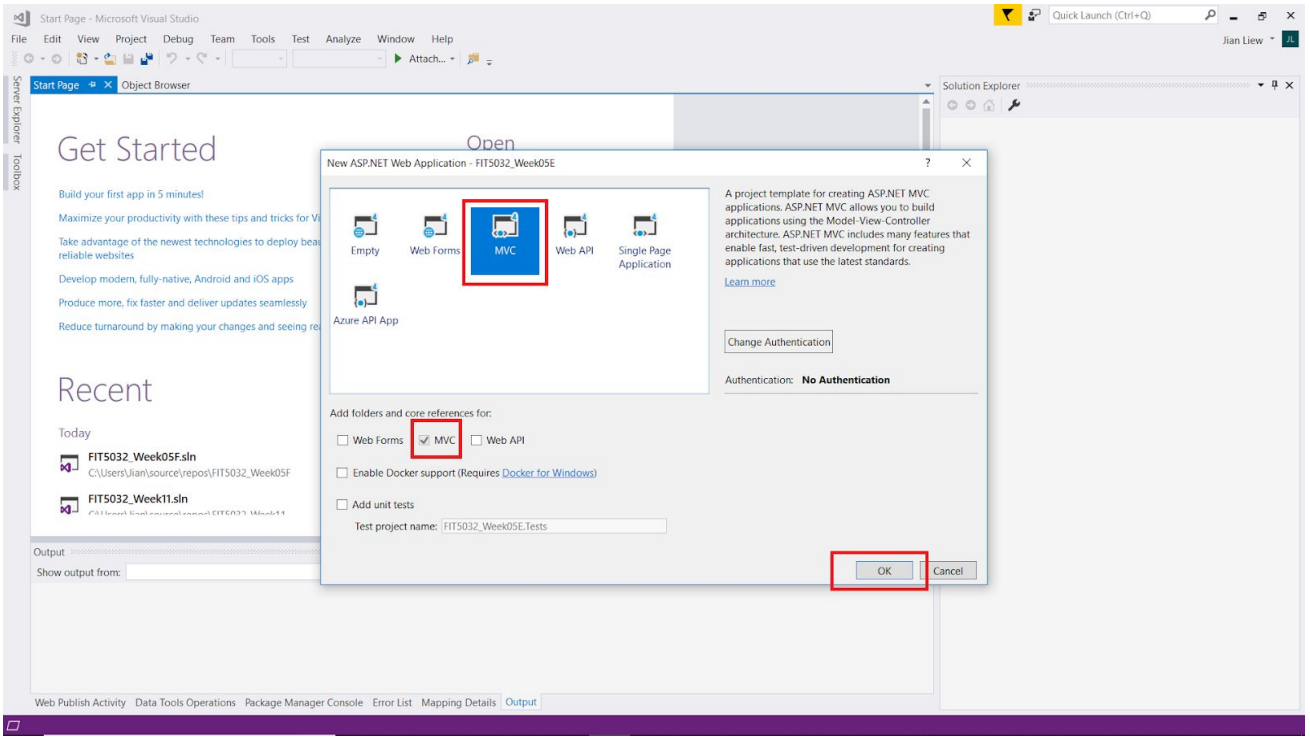
Please remember that this uses the LocalDb as it is the development environment. If you plan to deploy this, the strategy of creating the database will differ slightly.

Please remember that it is possible to zoom into the pictures in this document.

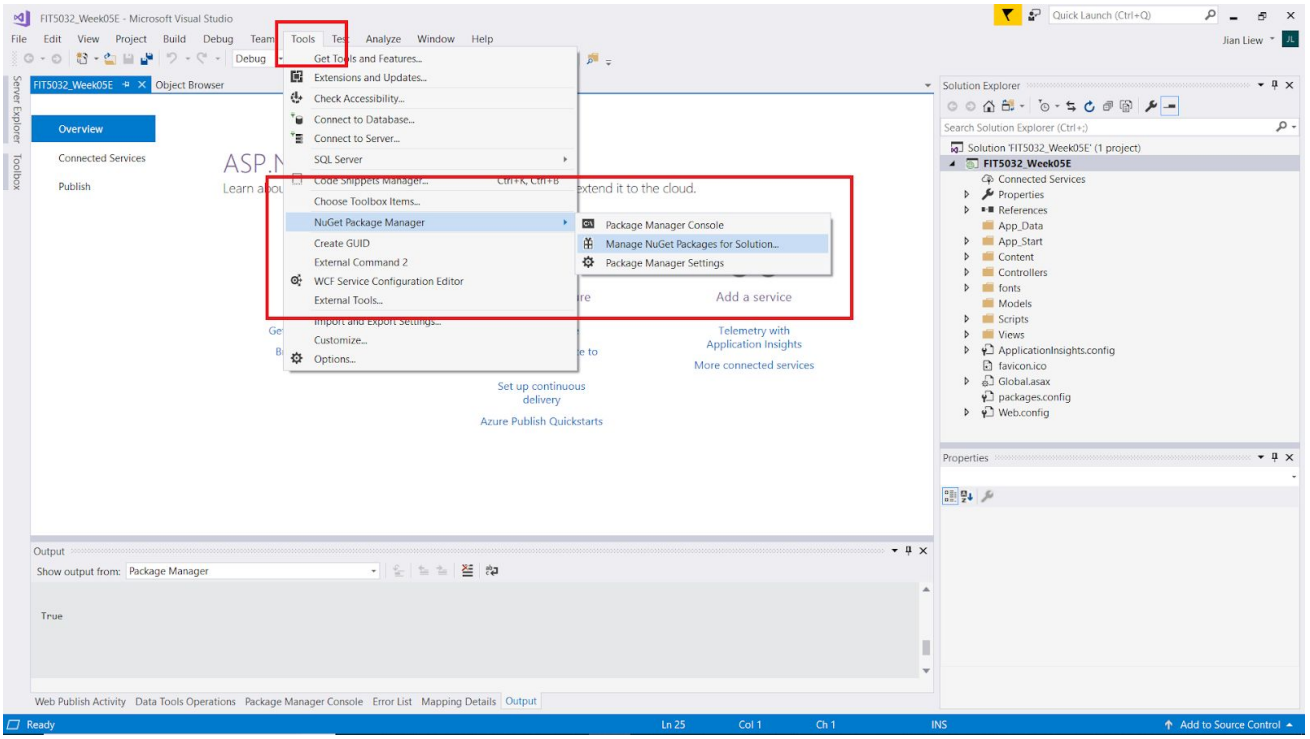
Step 1


Start Page - Microsoft Visual Studio
New Project
Recent
Installed
Visual C#
Web
Previous Versions
.NET Standard
Cloud
Test
WCF
Visual Basic
Visual F#
SQL Server
JavaScript
TypeScript
Other Project Types
Online
ASP.NET Core Web Application
ASP.NET Web Application (.NET Framework)
Not finding what you are looking for?
Open Visual Studio Installer
Name: FIT5032_Week05E
Location: C:\Users\Jian\source\repos
Solution name: FIT5032_Week05E
Framework: .NET Framework 4.5
Create directory for solution
Create new Git repository
OK
Cancel
Get Started
Recent
FIT5032_Week05F.sln
FIT5032_Week11.sln

Step 2


Start Page - Microsoft Visual Studio
New ASP.NET Web Application - FIT5032_Week05E
Empty
Web Forms
MVC
Web API
Single Page Application
Azure API App
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.
Learn more
Change Authentication
Authentication: No Authentication
Add folders and core references for:
Web Forms
MVC
Web API
Enable Docker support (Requires Docker for Windows)
Add unit tests
Test project name: FIT5032_Week05E.Tests
OK
Cancel
Get Started
Recent
FIT5032_Week05F.sln
FIT5032_Week11.sln

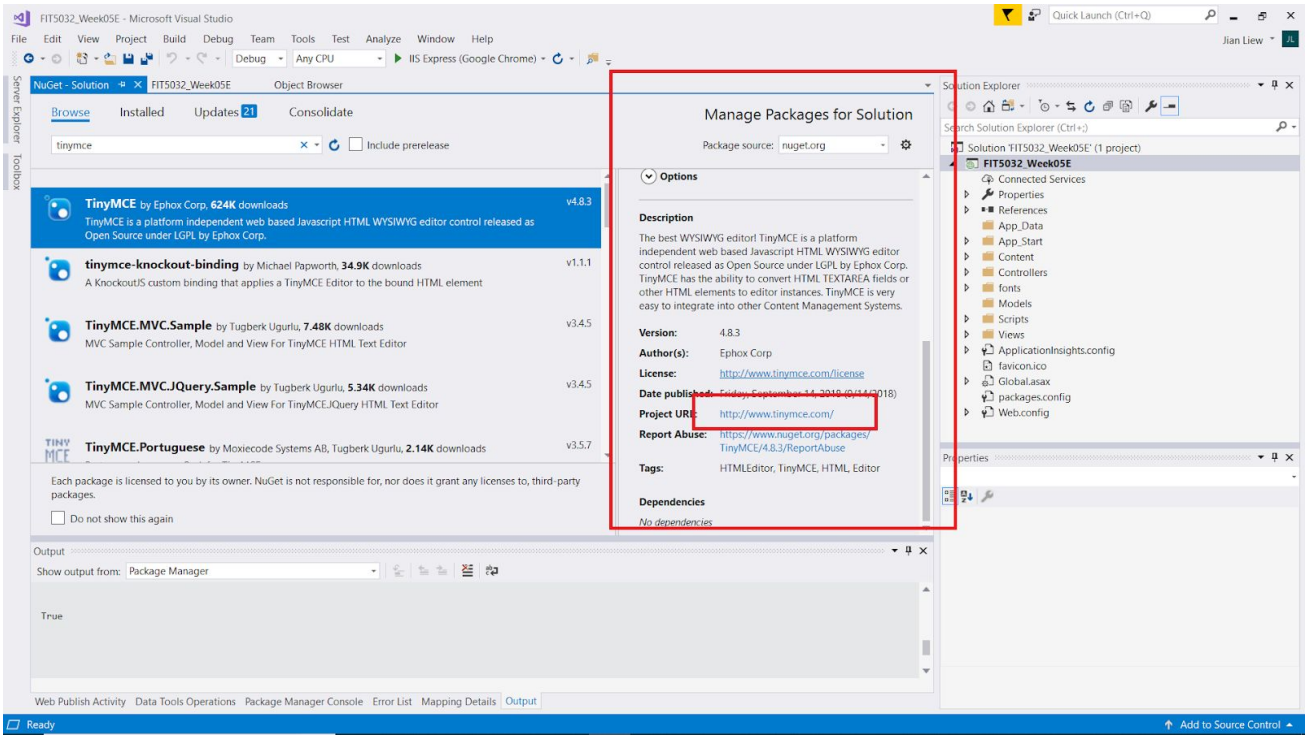
Step 3


FIT5032_Week05E - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Quick Launch (Ctrl+Q)
Jian Liew
Debug
Get Tools and Features...
Extensions and Updates...
Check Accessibility...
Connect to Database...
Connect to Server...
SQL Server
Code Snippets Manager... Ctrl+K, Ctrl+B
Choose Toolbox Items...
NuGet Package Manager
Create GUID
External Command 2
WCF Service Configuration Editor
External Tools...
Import and Export Settings...
Customize...
Options...
Package Manager Console
Manage NuGet Packages for Solution...
Package Manager Settings
FIT5032_Week05E
Object Browser
Overview
Connected Services
Publish
Add a service
Telemetry with Application Insights
More connected services
Set up continuous delivery
Azure Publish Quickstarts
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'FIT5032_Week05E' (1 project)
FIT5032_Week05E
Connected Services
Properties
References
App_Data
App_Start
Content
Controllers
fonts
Models
Scripts
Views
ApplicationInsights.config
favicon.ico
Global.asax
packages.config
Web.config
Properties
Output
Show output from: Package Manager
True
Web Publish Activity
Data Tools Operations
Package Manager Console
Error List
Mapping Details
Output
Ready
Ln 25
Col 1
Ch 1
INS
Add to Source Control

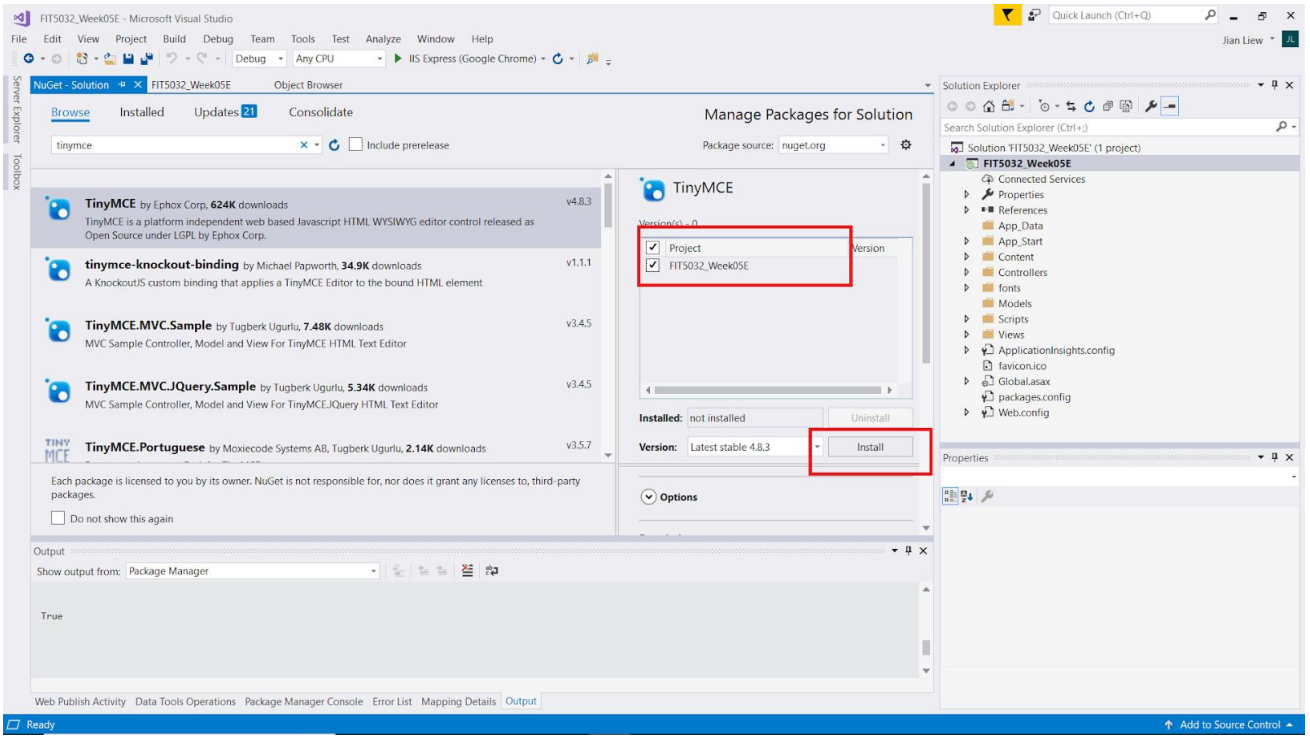
Step 4

Step 5

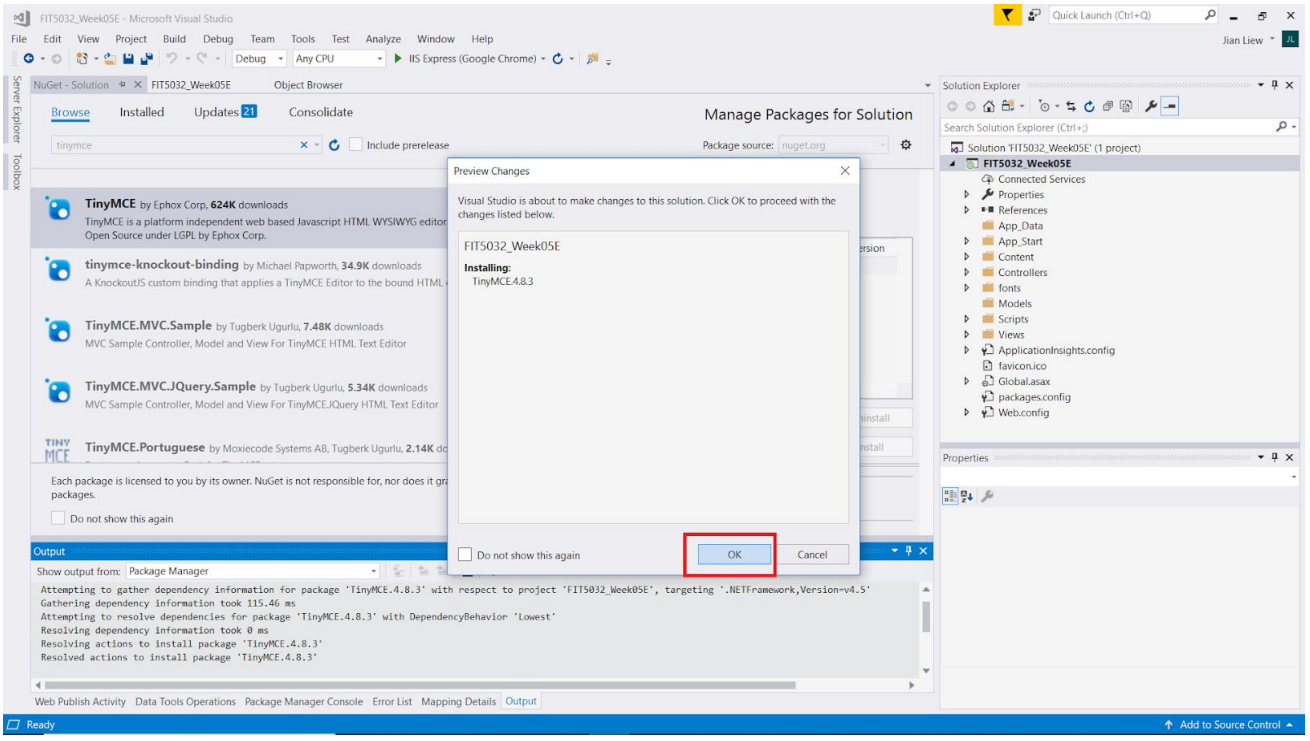
Please remember that **reading documentation is a very important skill as a developer**. When selecting a package to use it is highly recommended to use something with a lot of downloads.



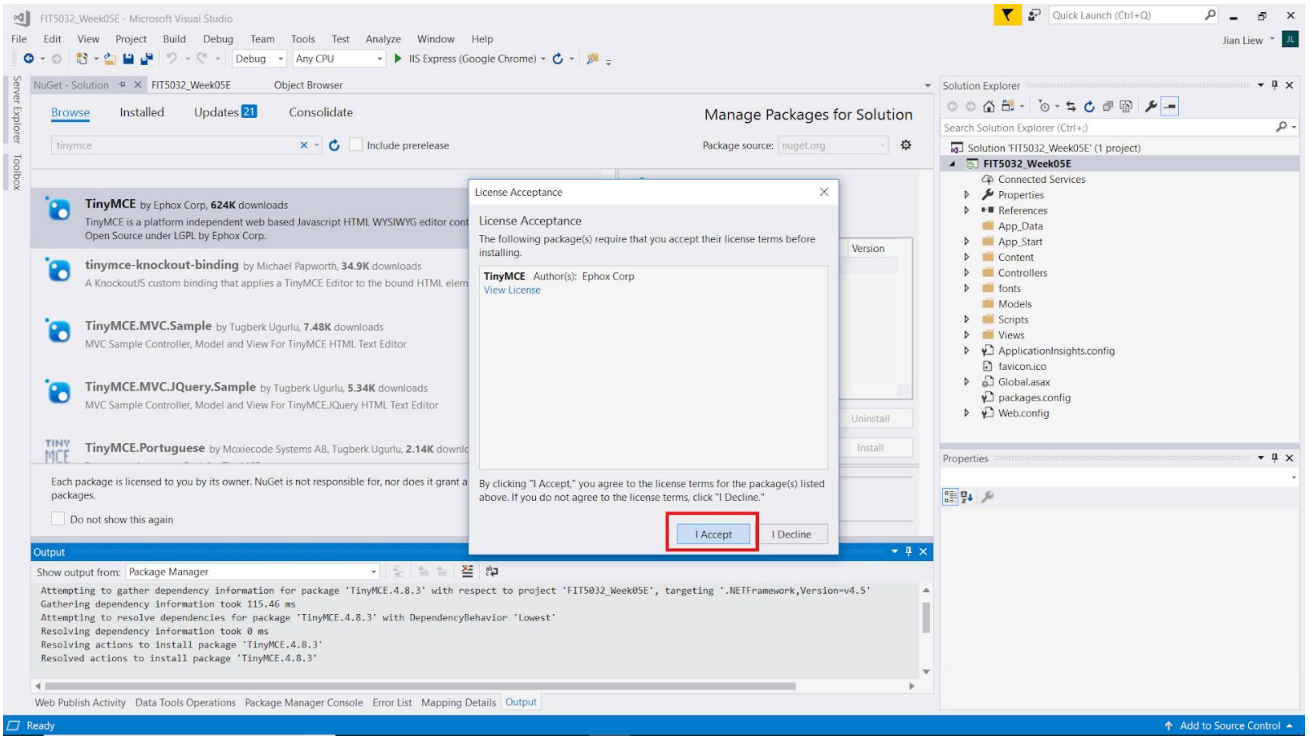
Step 6



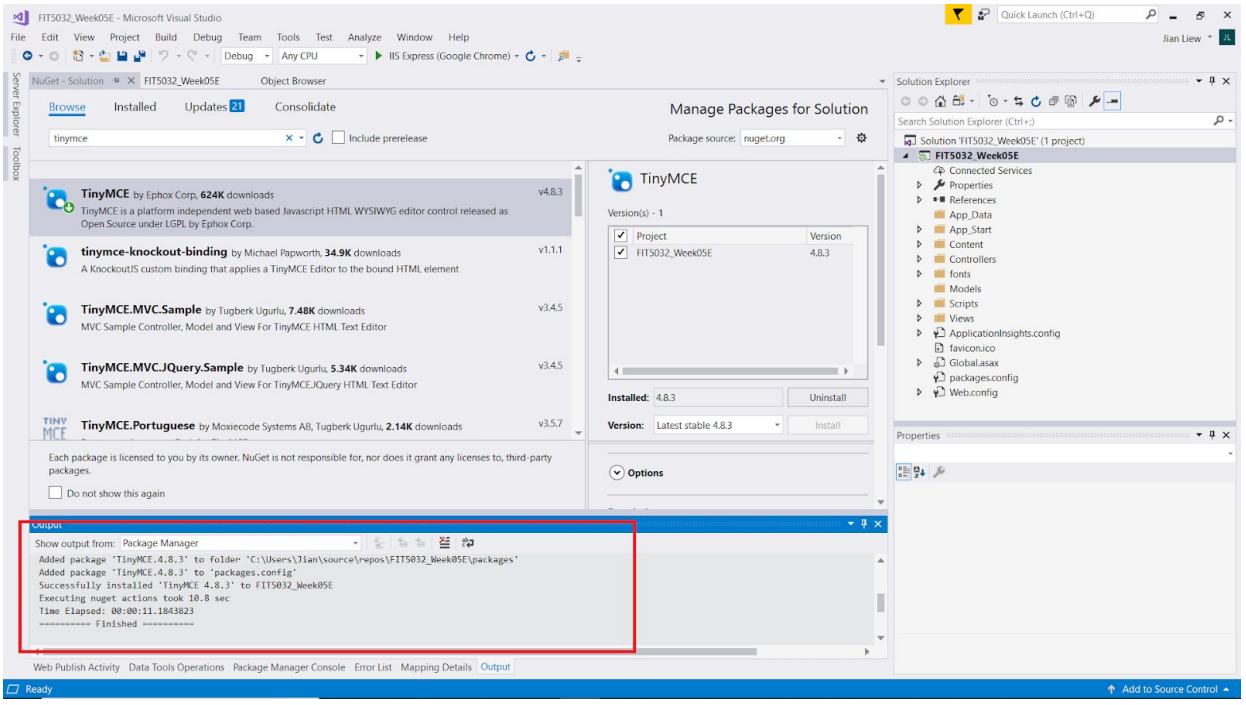
Step 7


Preview Changes
Visual Studio is about to make changes to this solution. Click OK to proceed with the changes listed below.
FIT5032_Week05E
Installing:
TinyMCE.4.8.3
Do not show this again
OK
Cancel
Manage Packages for Solution
TinyMCE by Ephox Corp, 624K downloads
tinymce-knockout-binding by Michael Papworth, 34.9K downloads
TinyMCE.MVC.Sample by Tugberk Ugurlu, 7.48K downloads
TinyMCE.MVC.JQuery.Sample by Tugberk Ugurlu, 5.34K downloads
TinyMCE.Portuguese by Moxiecode Systems AB, Tugberk Ugurlu, 2.14K
Output
Attempting to gather dependency information for package 'TinyMCE.4.8.3' with respect to project 'FIT5032_Week05E', targeting '.NETFramework,Version=v4.5'
Gathering dependency information took 115.46 ms
Attempting to resolve dependencies for package 'TinyMCE.4.8.3' with DependencyBehavior 'Lowest'
Resolving dependency information took 0 ms
Resolving actions to install package 'TinyMCE.4.8.3'
Resolved actions to install package 'TinyMCE.4.8.3'
Ready
Add to Source Control

Step 8

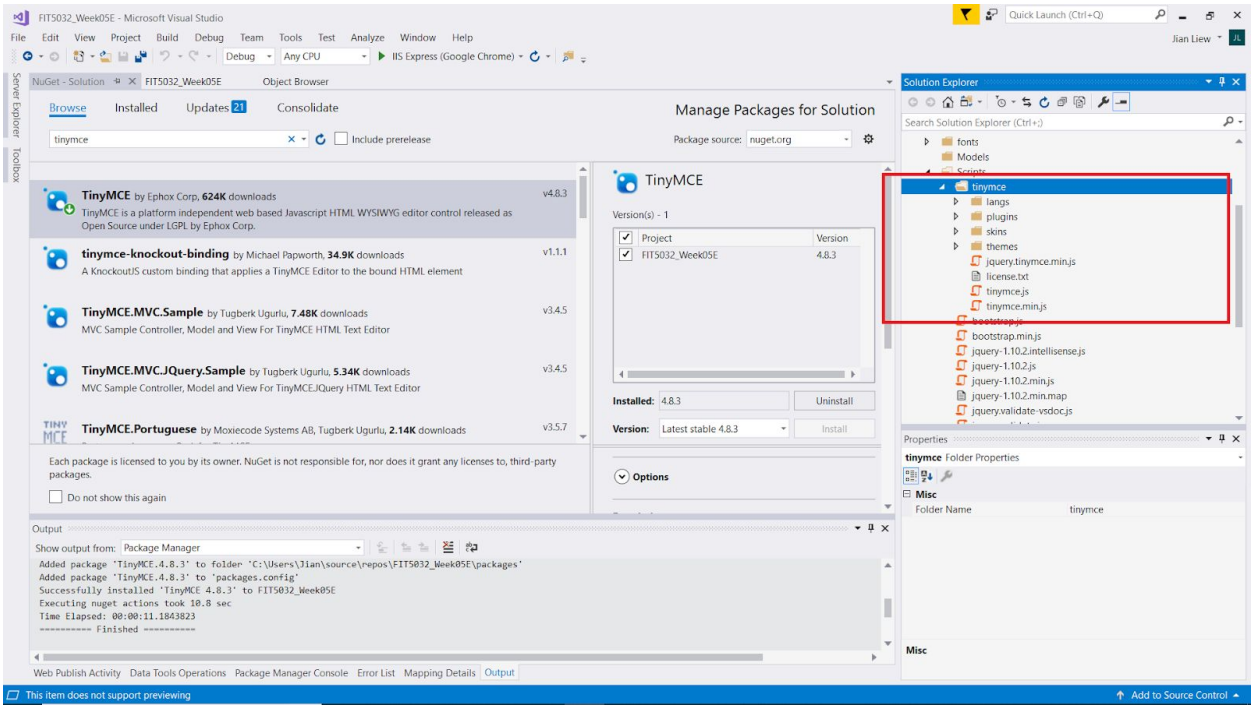

License Acceptance
License Acceptance
The following package(s) require that you accept their license terms before installing.
TinyMCE Author(s): Ephox Corp
View License
By clicking "I Accept," you agree to the license terms for the package(s) listed above. If you do not agree to the license terms, click "I Decline."
I Accept
I Decline
Manage Packages for Solution
Output
Attempting to gather dependency information for package 'TinyMCE.4.8.3' with respect to project 'FIT5032_Week05E', targeting '.NETFramework,Version=v4.5'
Gathering dependency information took 115.46 ms
Attempting to resolve dependencies for package 'TinyMCE.4.8.3' with DependencyBehavior 'Lowest'
Resolving dependency information took 0 ms
Resolving actions to install package 'TinyMCE.4.8.3'
Resolved actions to install package 'TinyMCE.4.8.3'
Ready
Add to Source Control

Step 9



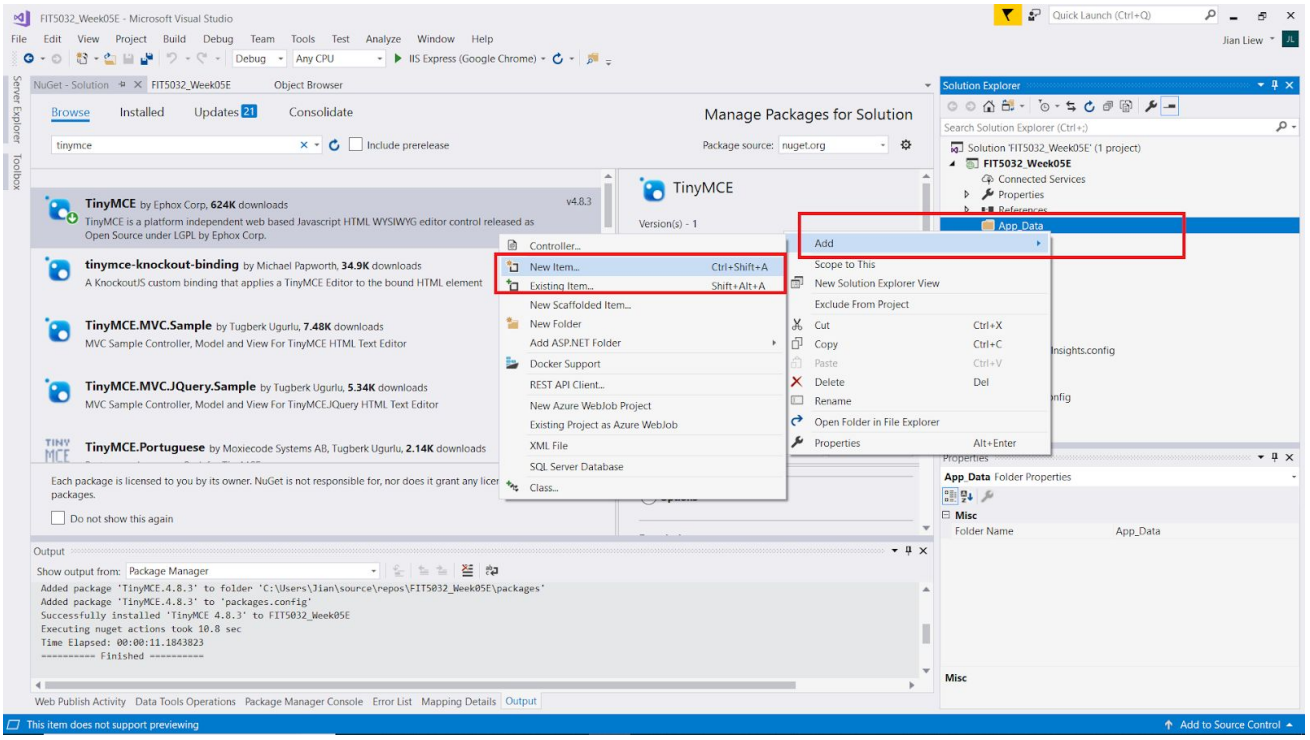
Step 10

You will notice that upon the completion of the "install", there will be a new folder created called "tinymce" under the Scripts folder. Do keep in mind that this installation process merely means that it downloads the need source files and you will still need to implement it so it is working.

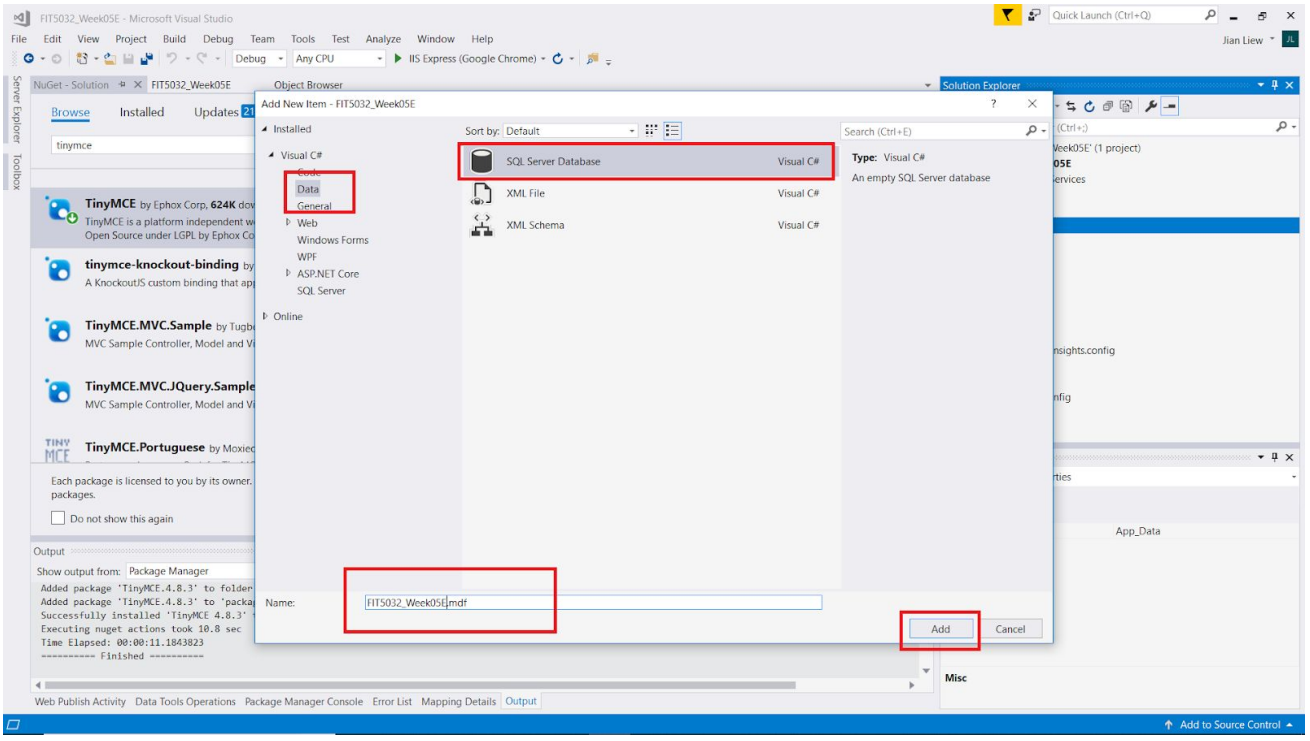


Step 11

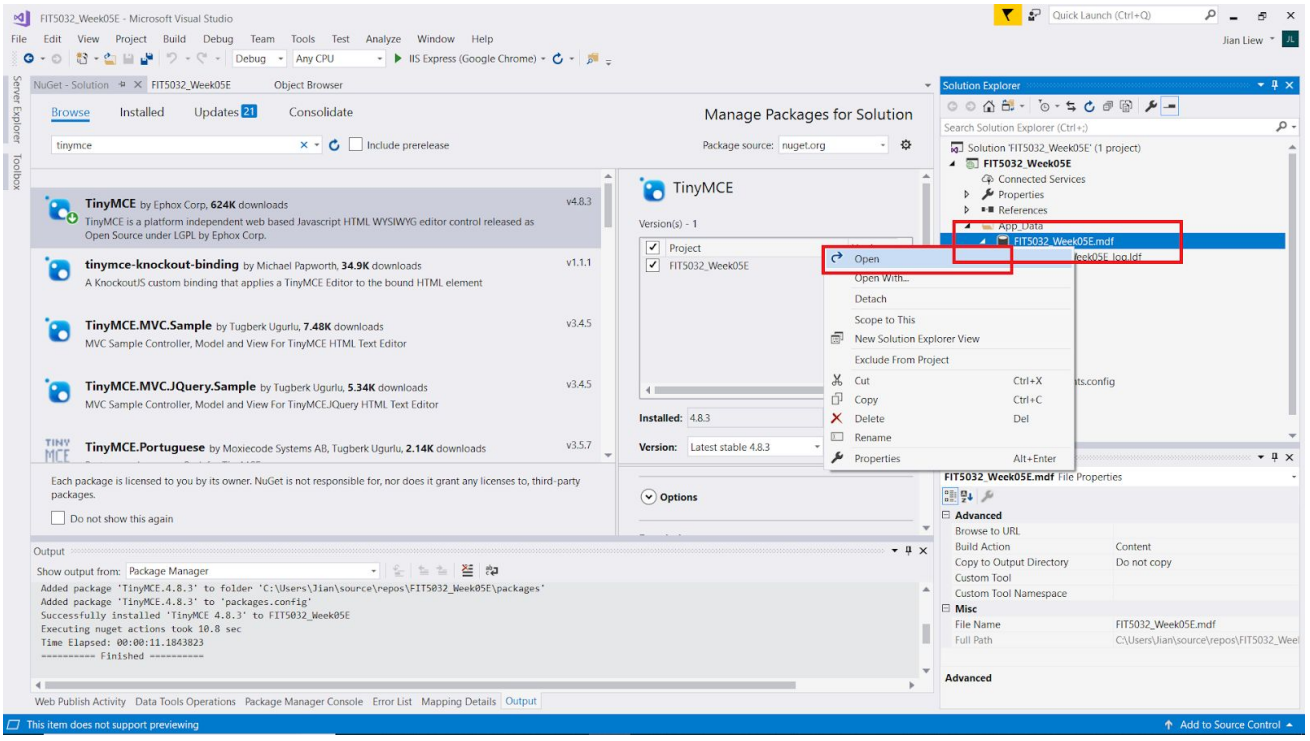
I will now use a database first approach to create a simple database that will allow me to store "Message" information.



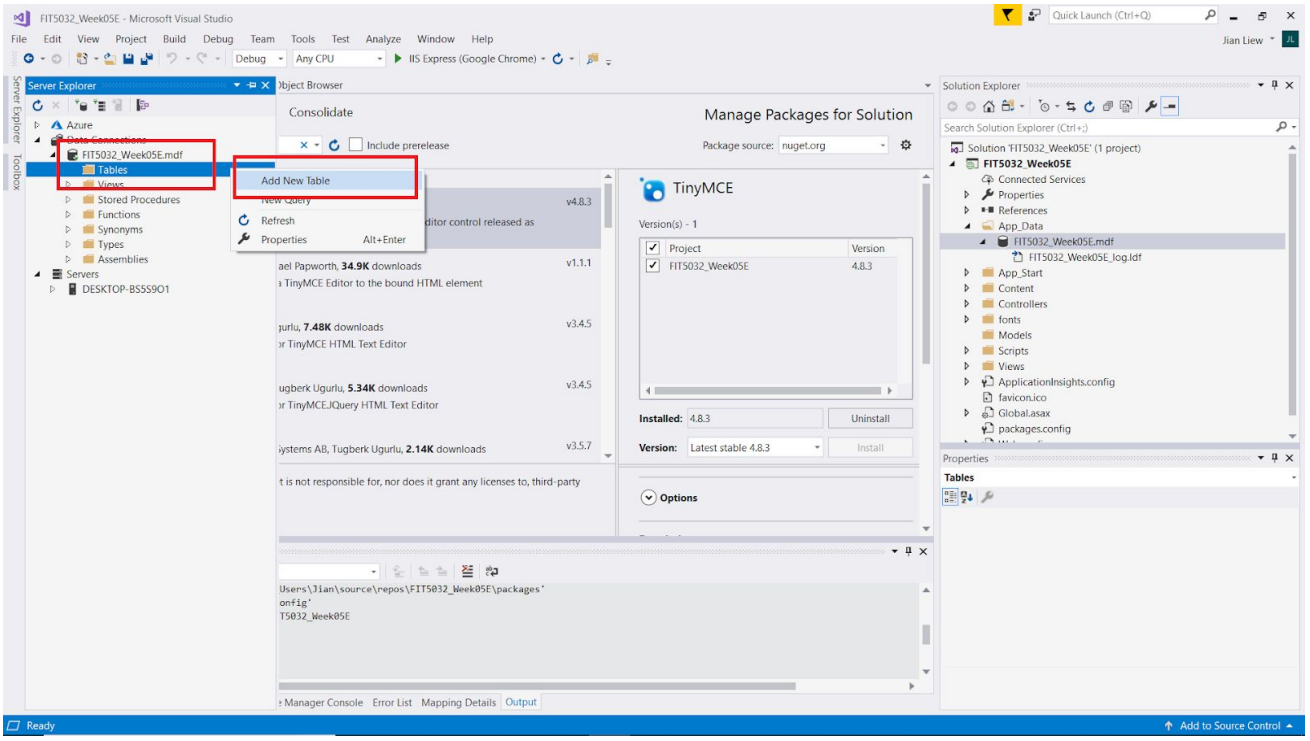
Step 12



Step 13

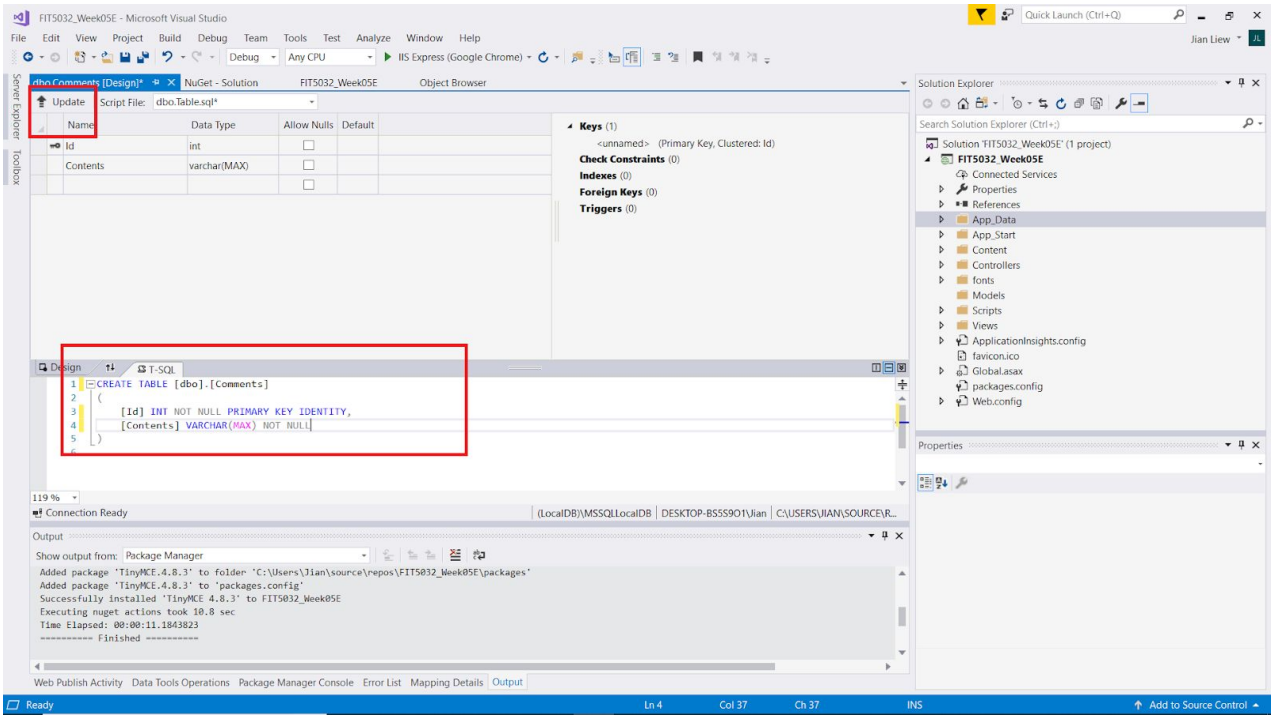


Step 14

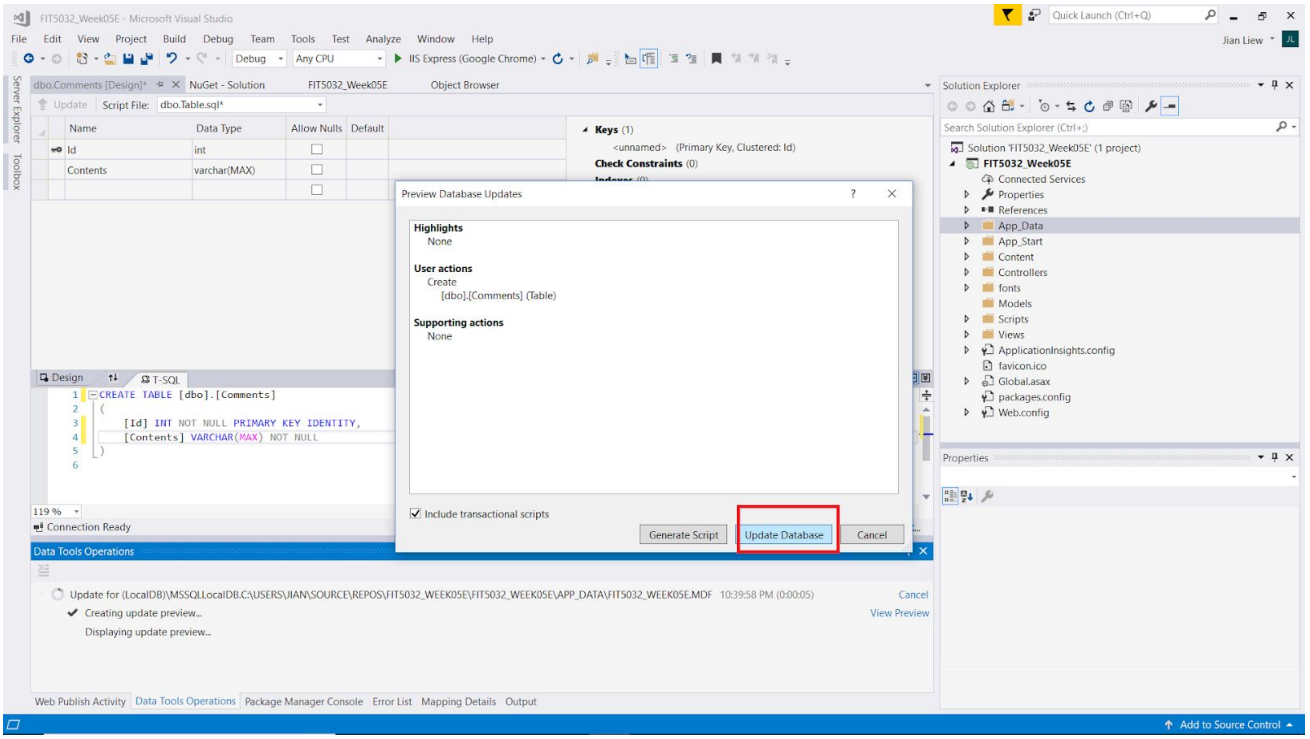


Step 15

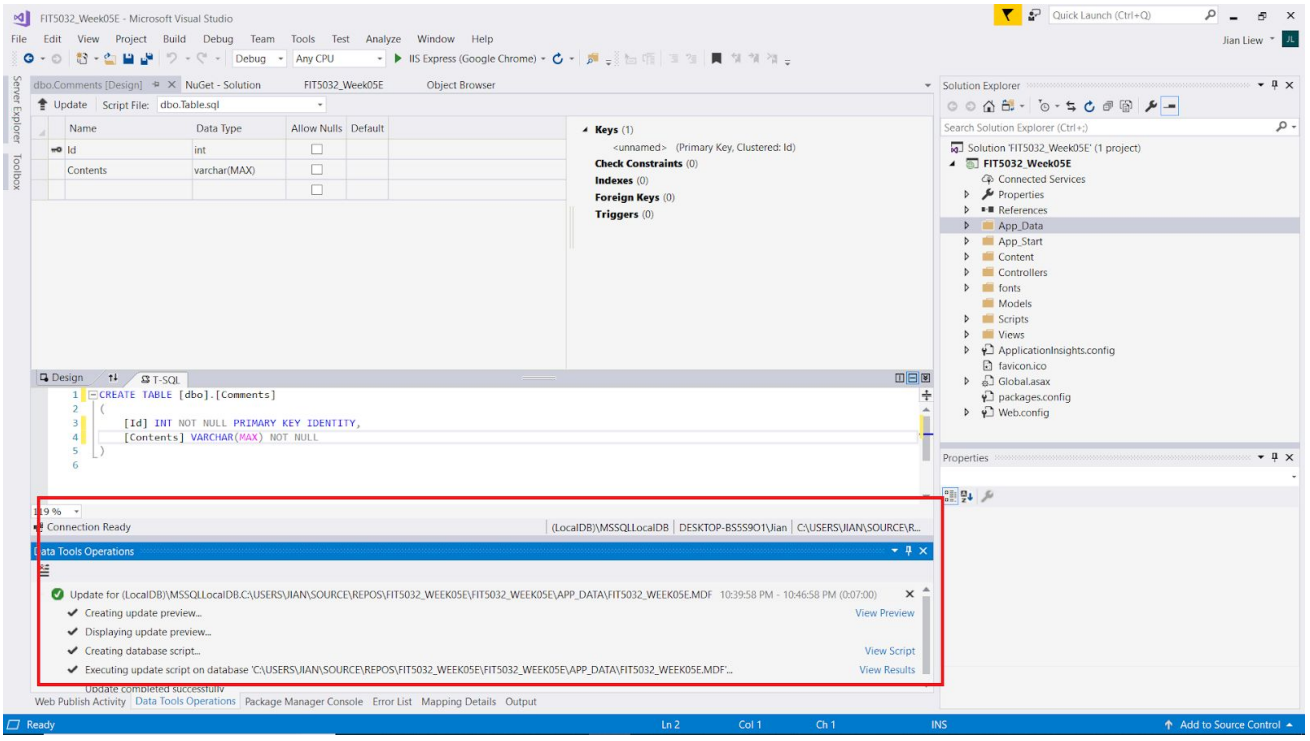
Here, I have introduced an Identity to my Primary Key column. Please remember to type the query in correctly. After that, you can hit the "Update" button. There is no difference between doing it this way or just executing a new query.



Step 17

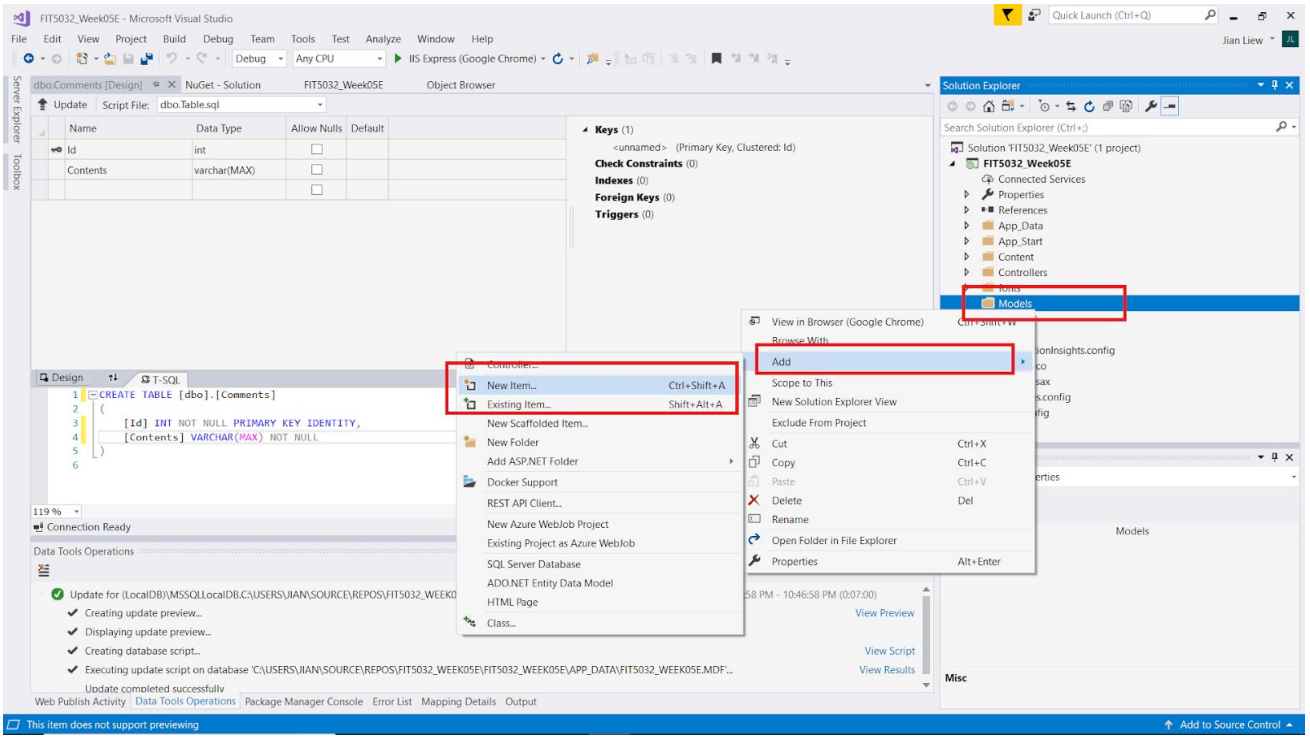


Step 18

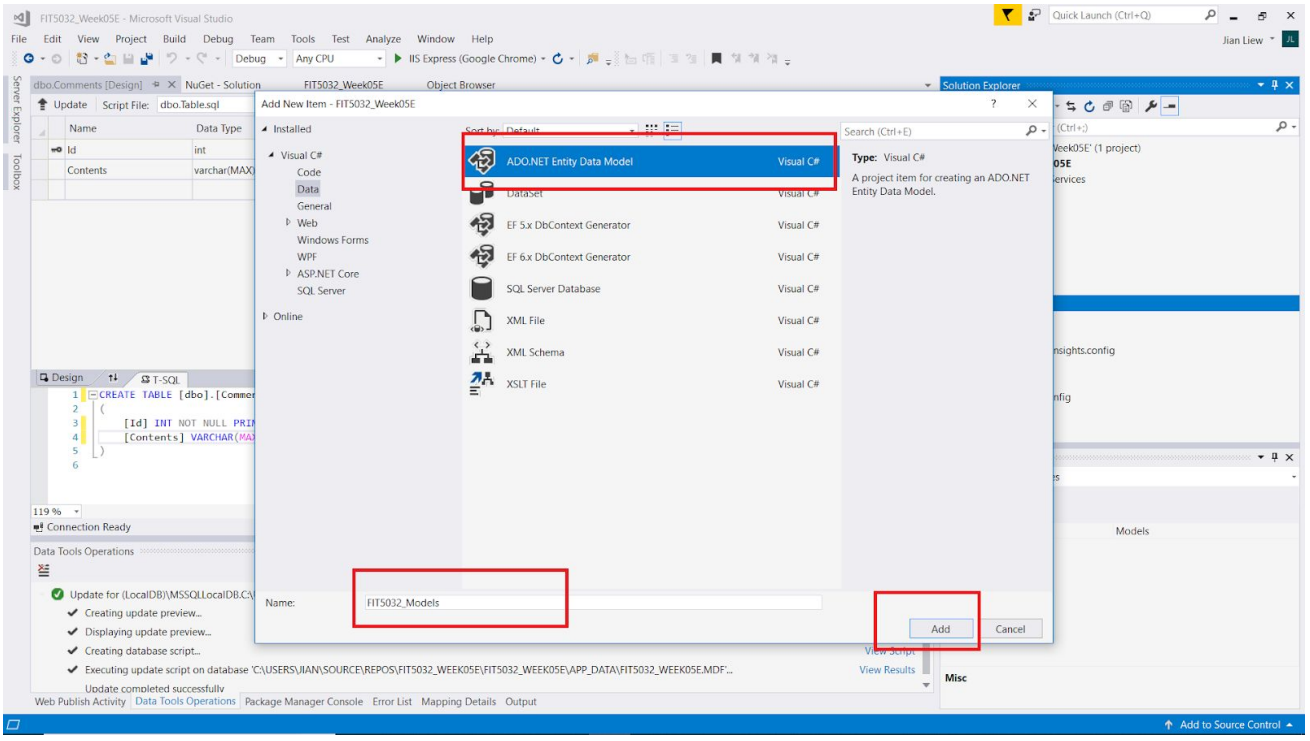


Step 19

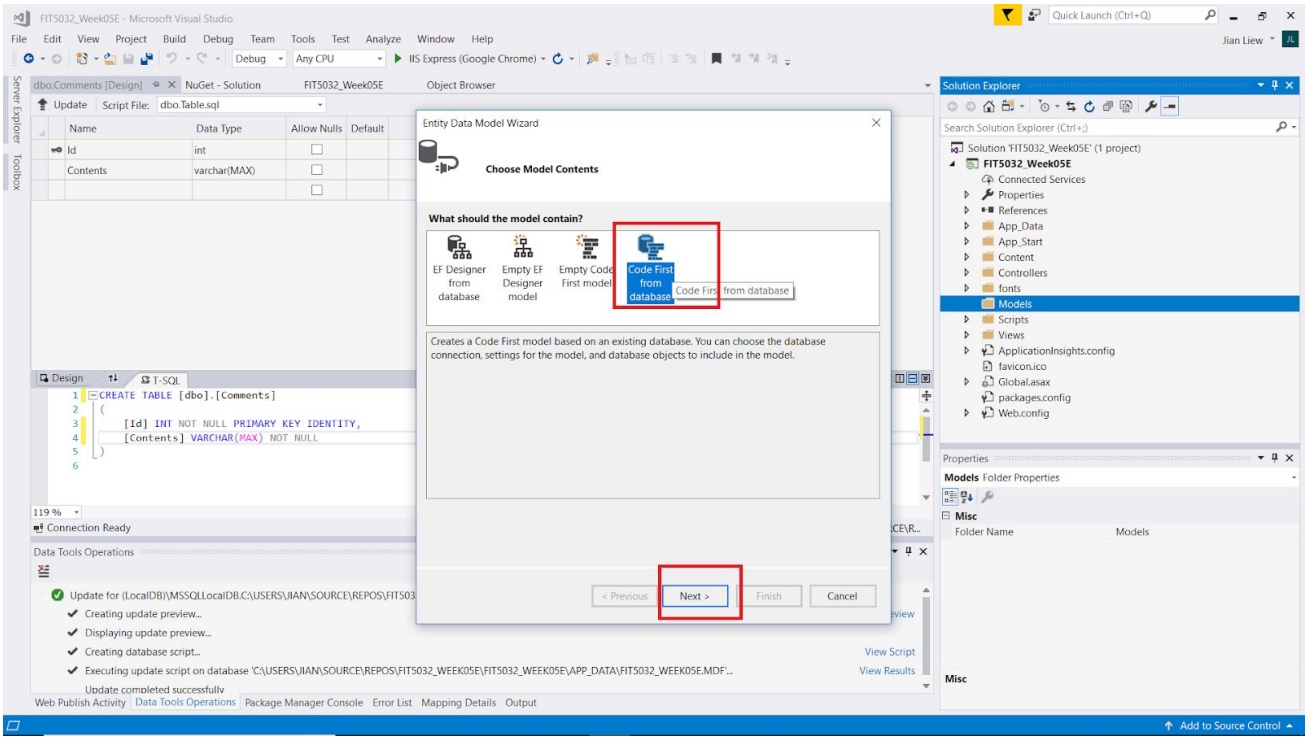
I will now create, my Models, Controllers and Views using a DB First approach.



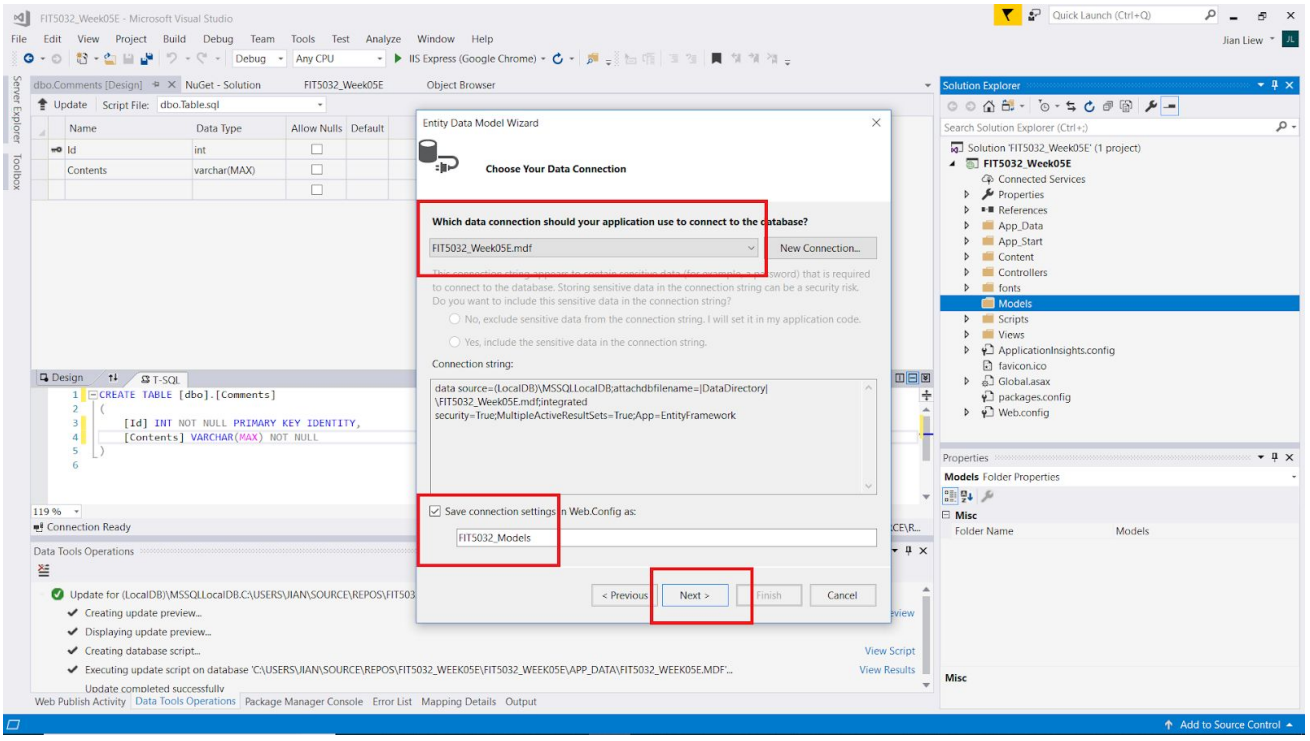
Step 20



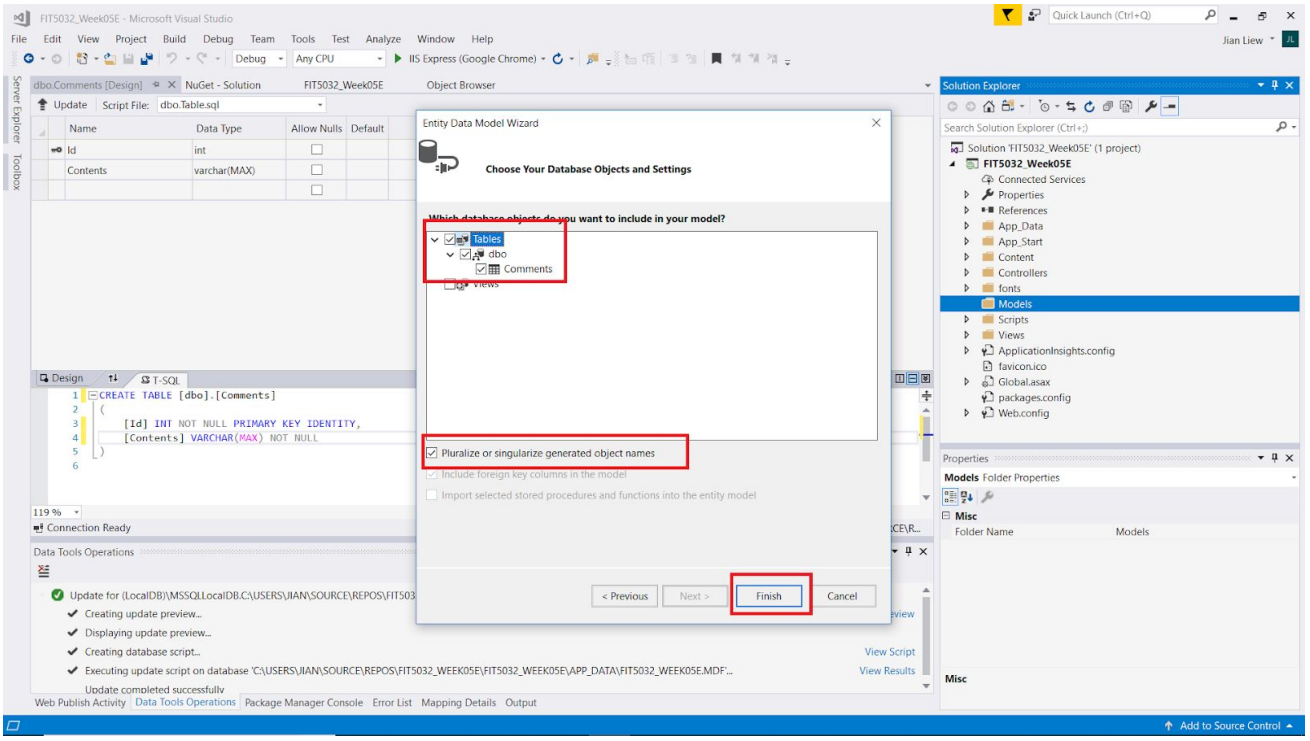
Step 21



Step 22

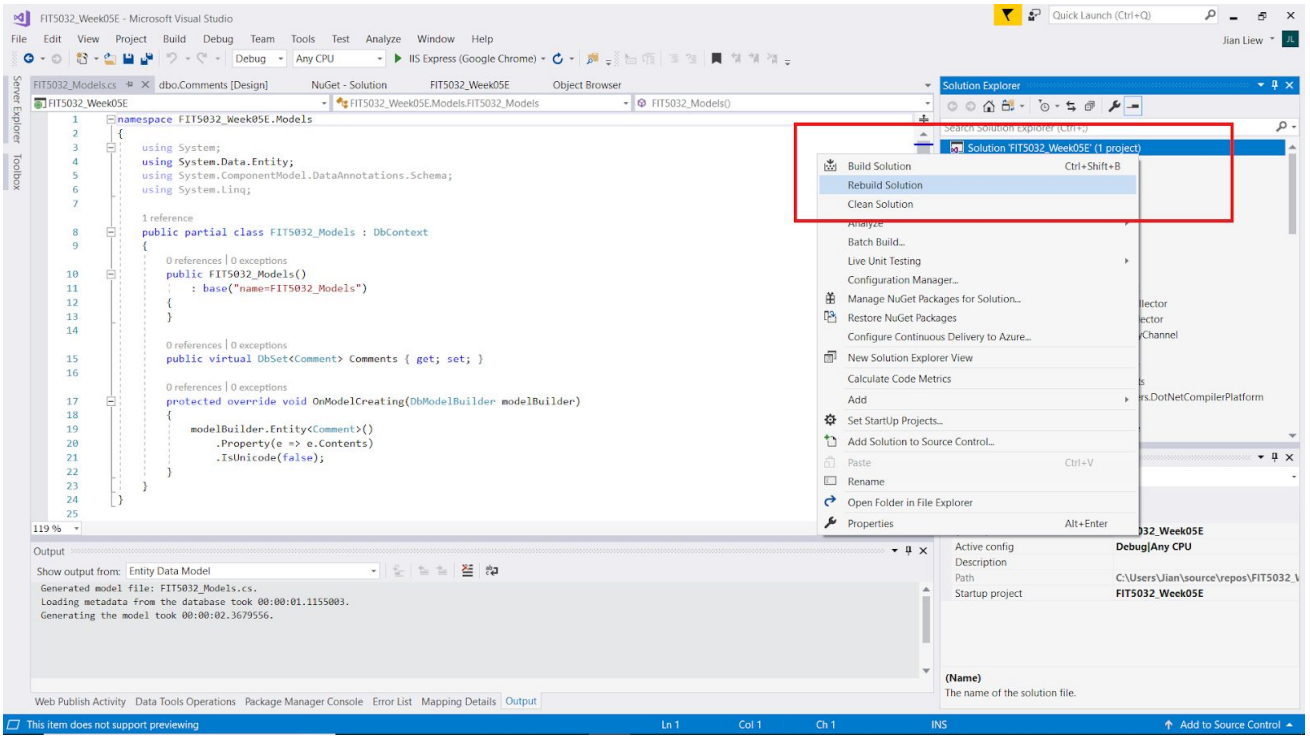


Step 23

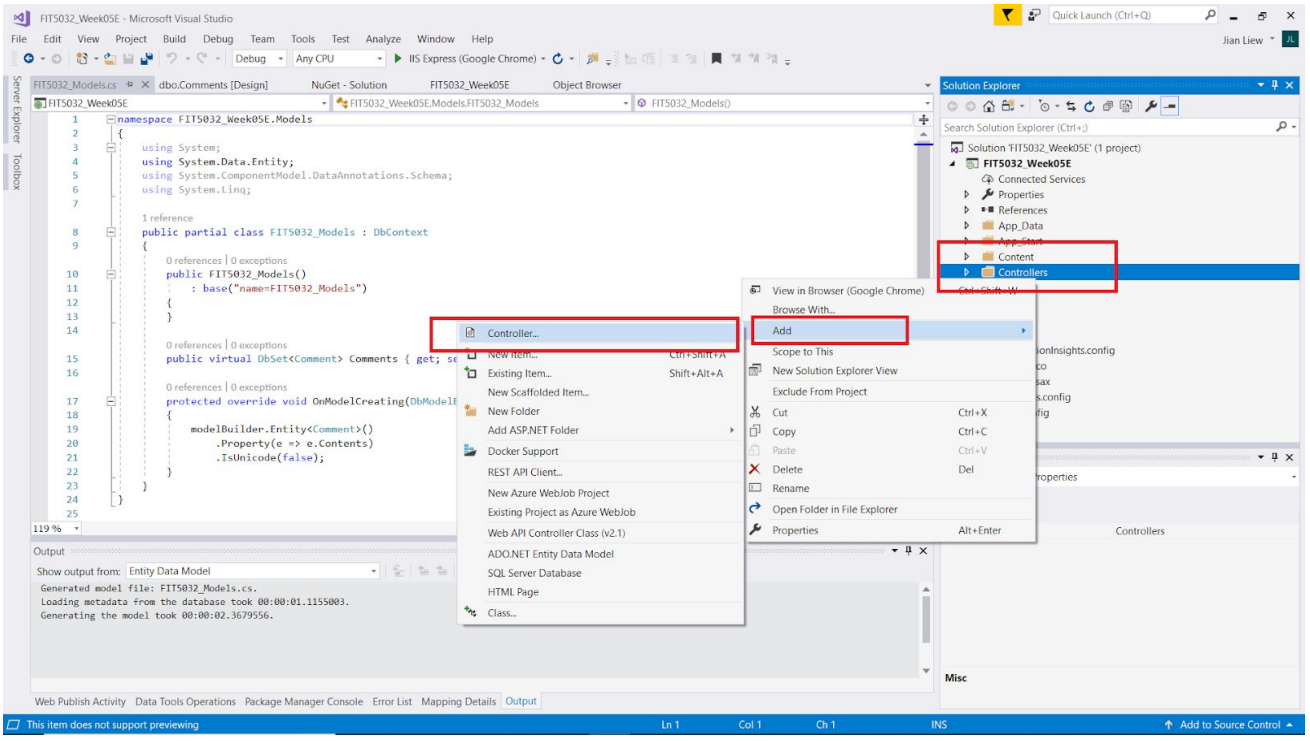


Step 24

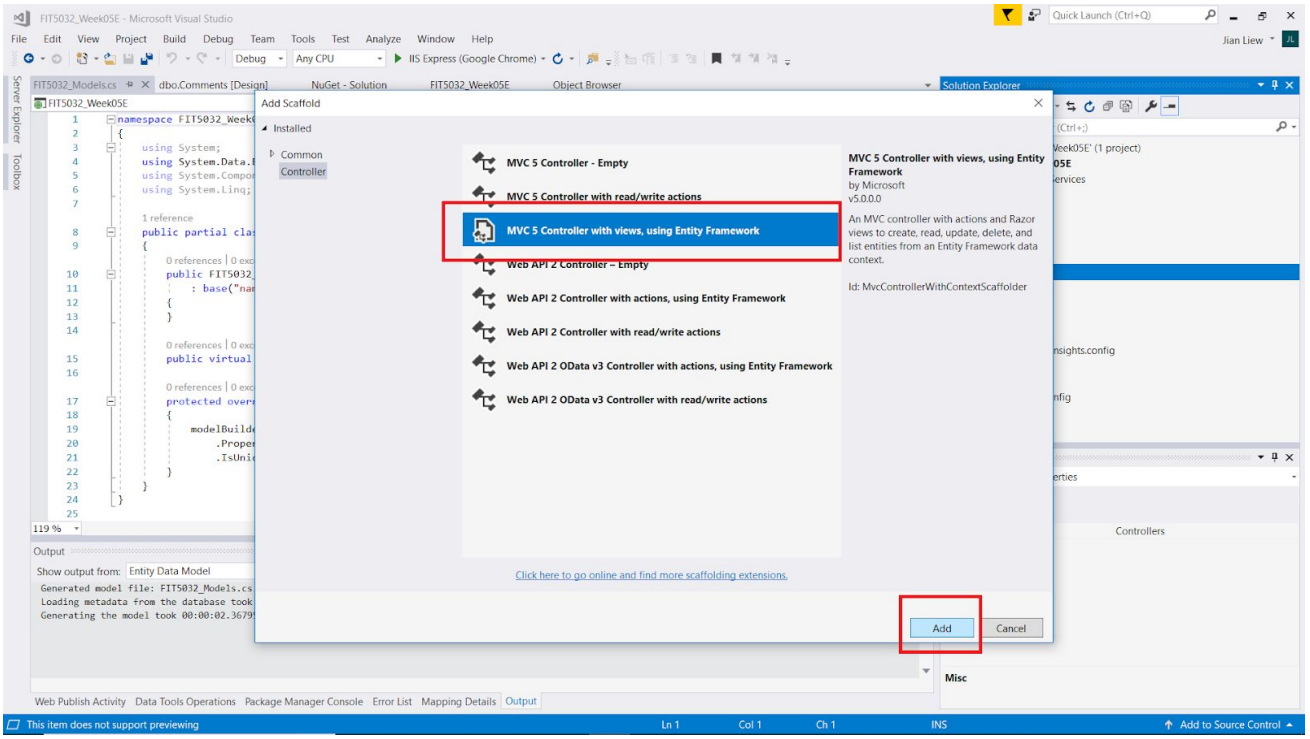
Please "Rebuild" your Solution.



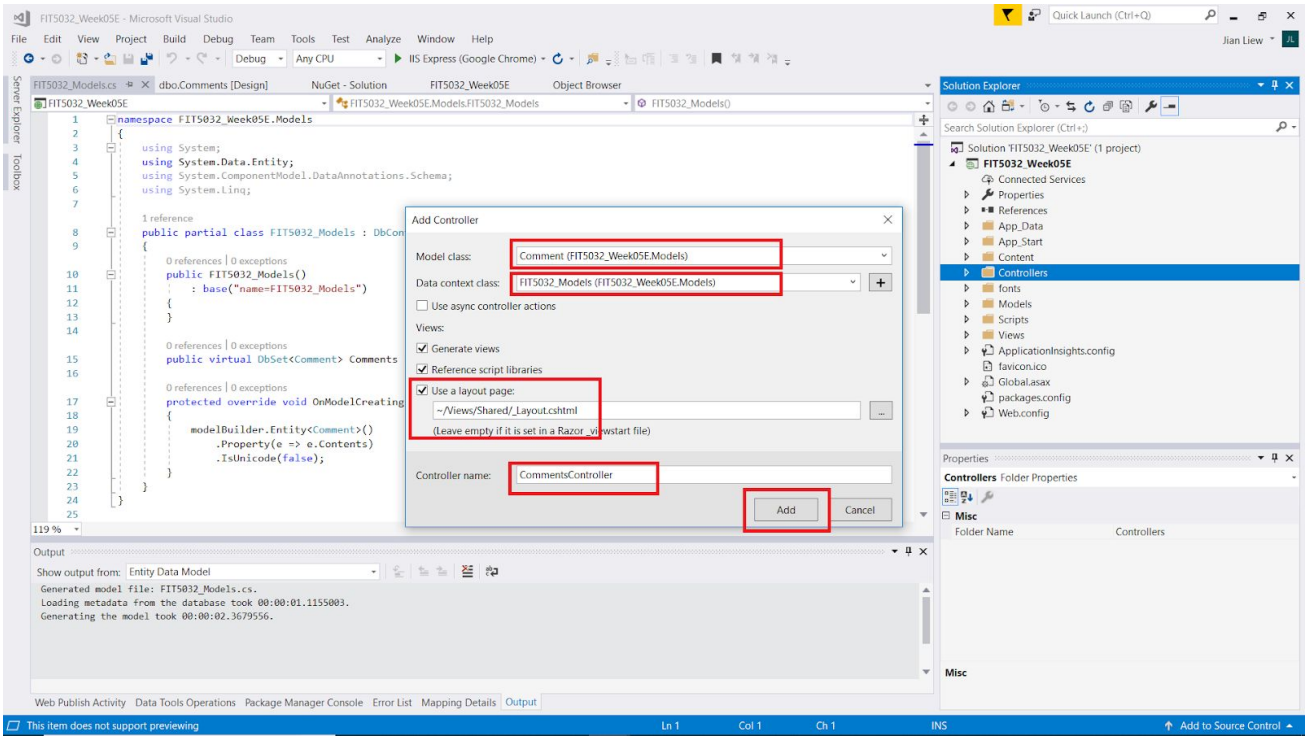
Step 25



Step 26



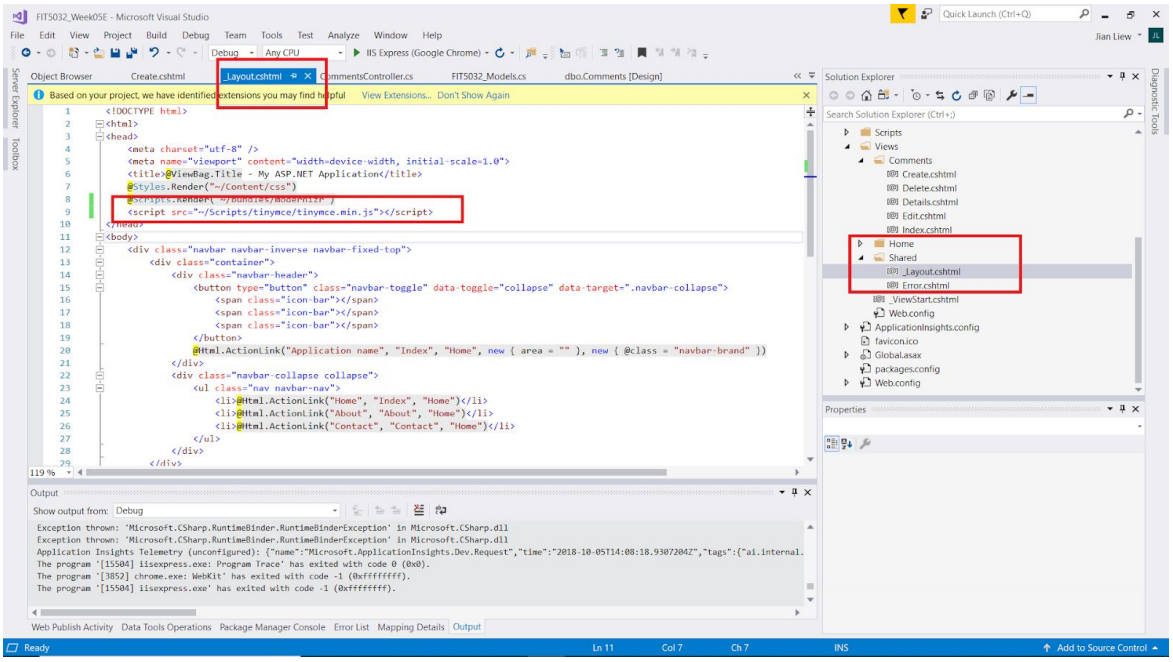
Step 27



Step 28

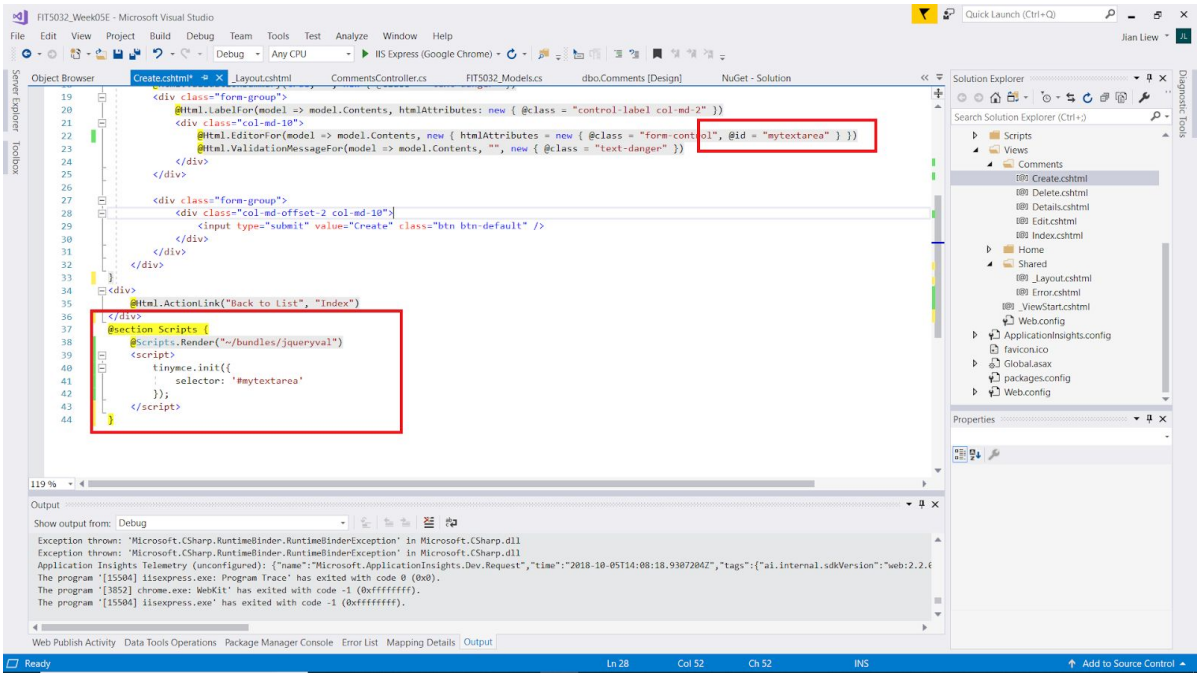
After that in order to get it working, a few steps need to be done.

You will first need to load the needed library. (You can do this at the _Layout.cshtml)



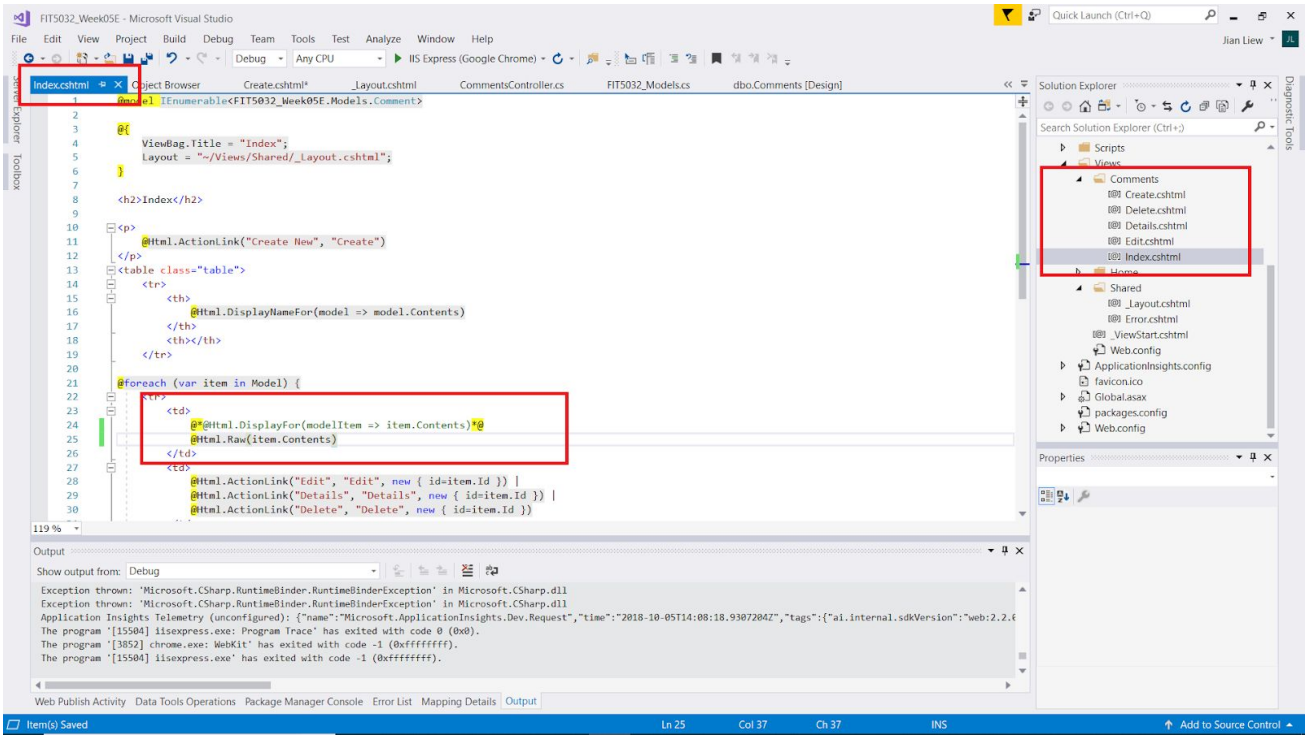
Step 29

You will also now need to introduce the correct id and write the correct scripts to load it. (Create.cshtml)



Step 30

At the Index.cshtml of Comments. Make the changes below. (Using a Html.Raw instead since we are using a markup now).

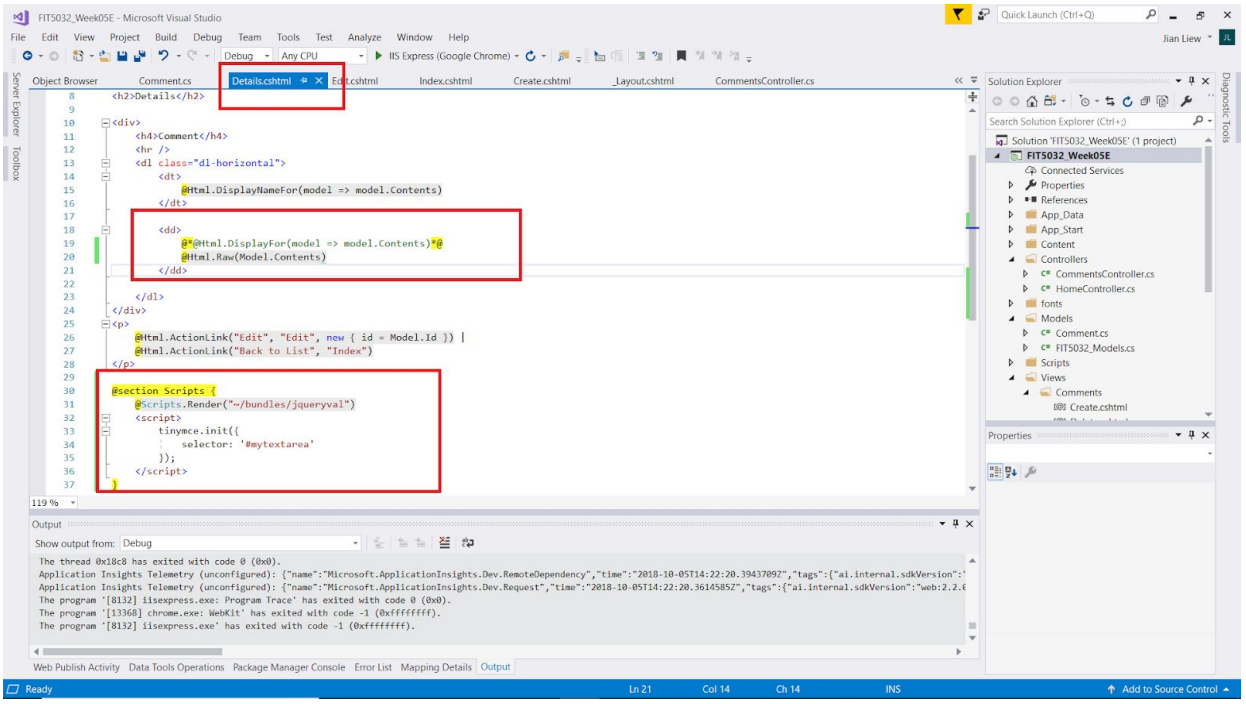


Step 31

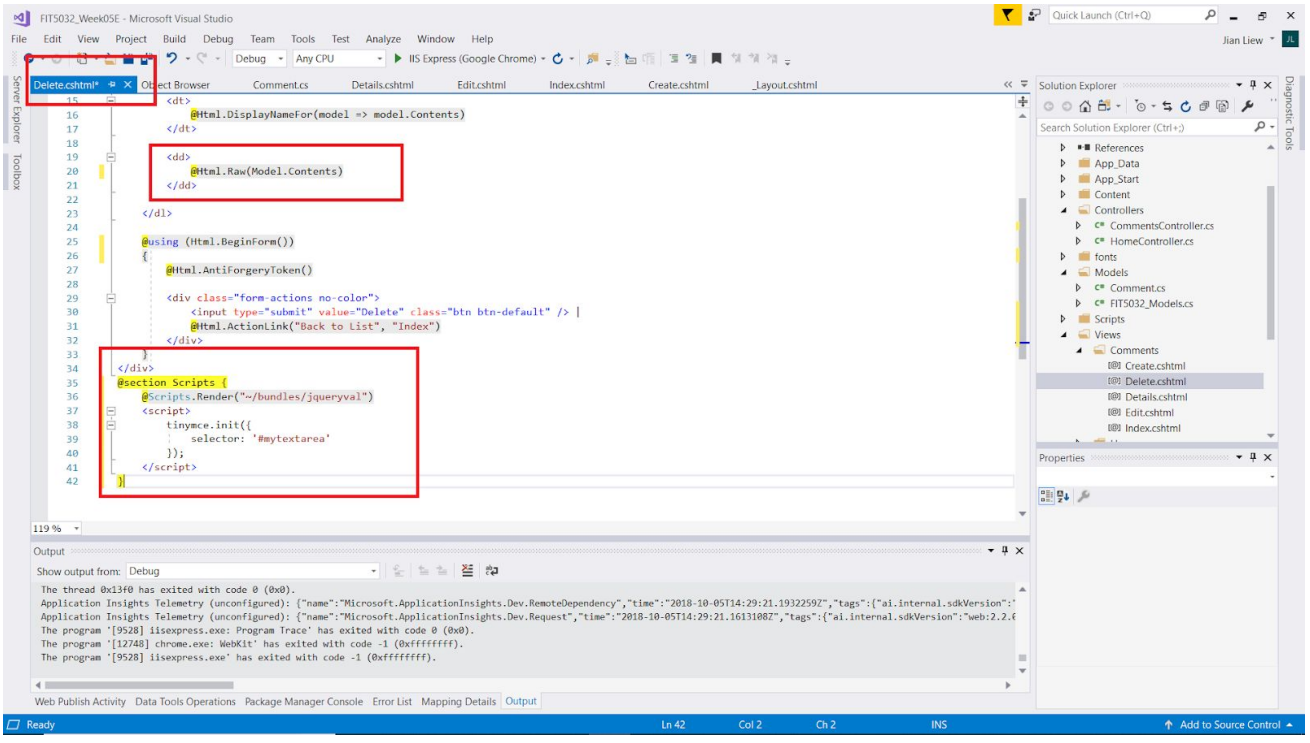
Make similar changes to the Edit.cshtml of Comments.

Step 32

At the Details.cshtml of Comments make the following changes.

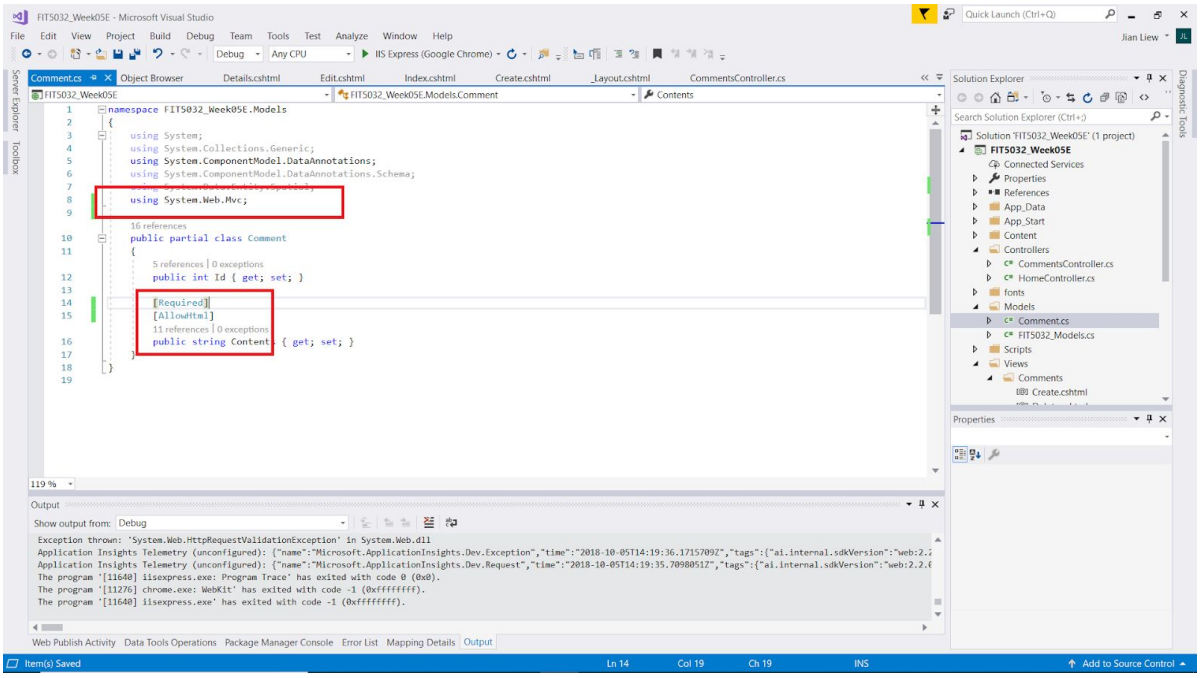


Step 33



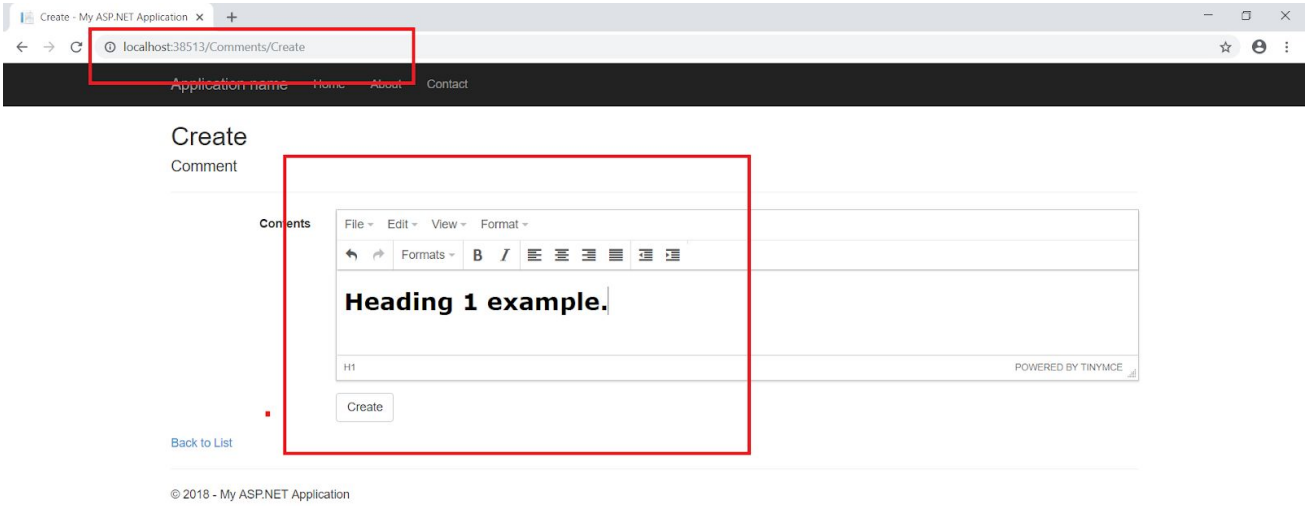
Step 34 (Important!)

You will also need to make the following changes at the Comment.cs of the Models. The [AllowHtml] attribute is now needed.

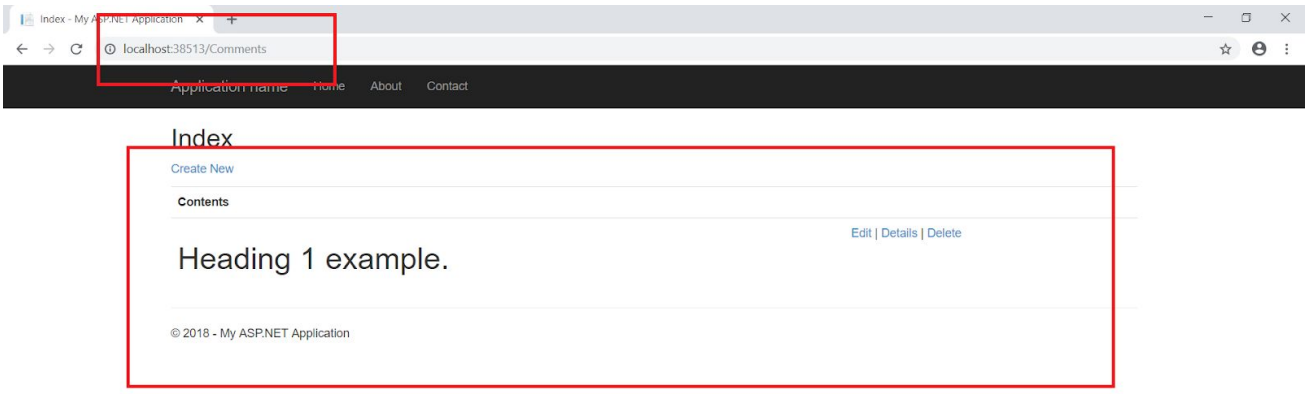


Step 34

Once you have completed that, the end result should be something very similar to this.



Step 35



Step 36

You should introduce a hyperlink to this pages, so that the user experience will increase.

Explanation

In this supplementary tutorial, you would have learn how to use a simple JavaScript library to enable a WYSIWYG like feature. If you are interested, you can further look up how to configure the TinyMCE library to allow for more fancy features.

Conclusion

The objective of this supplementary material, is

- To showcase how to use TinyMCE at a basic level.
- How to edit Razor view for certain desired displays.