

FIT5032 - Internet Applications Development

WEEK 08B - FILE UPLOAD

Last updated: 29th July 2018

Author: Jian Liew

Objectives

Estimated Time To Complete - ~30 mins

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

- How to upload files (images, jpeg & png)

DoubtFire Submission

- **None**

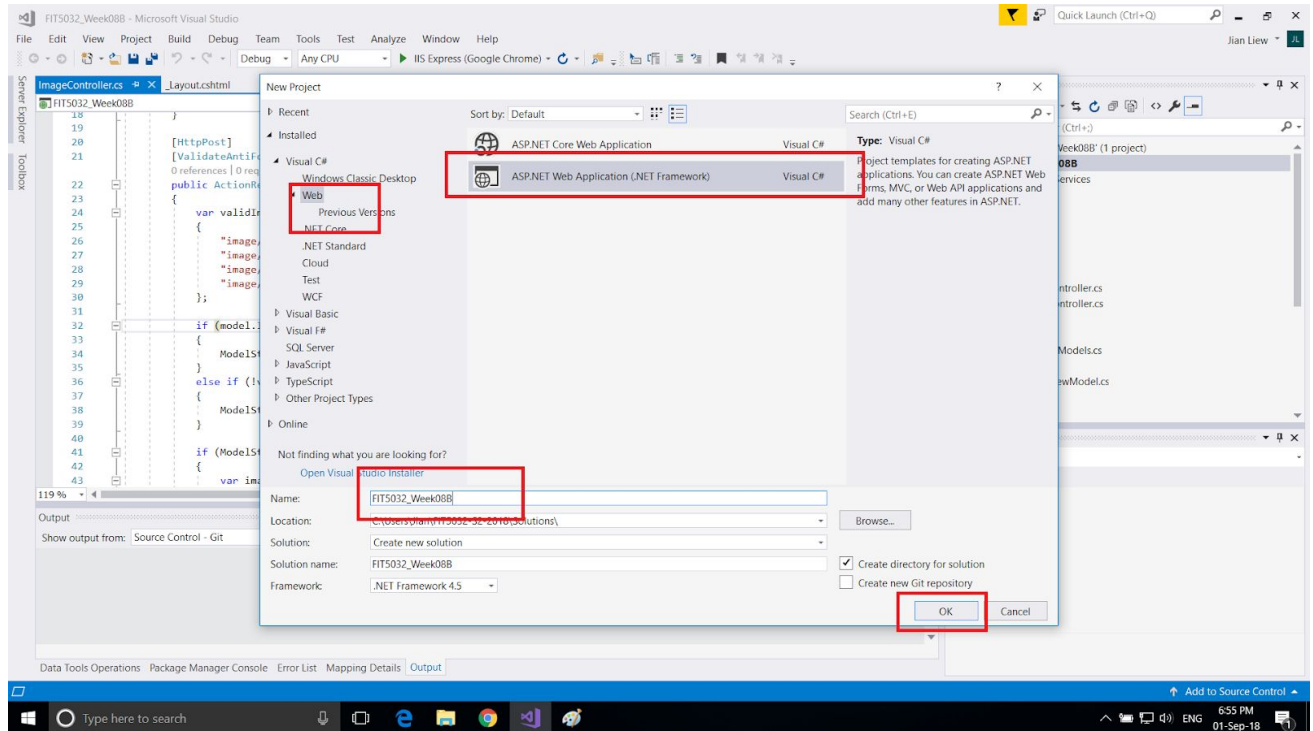
Tutorial Structure

The tutorials in this unit are designed to be of a **self-paced and self-taught structure**. So, the aim of your tutor is to aid you in your learning experience. He or she will not go through all the materials that are covered in the labs and will **not do the exercise one by one as a class**. If help is needed, you can either post on the Moodle forums or you can email your tutor asking for clarification. This is slightly different in comparison to other units, where your tutor will lead the discussions. **Please take note of this, you will only be provided help if you make it known that you need help.**

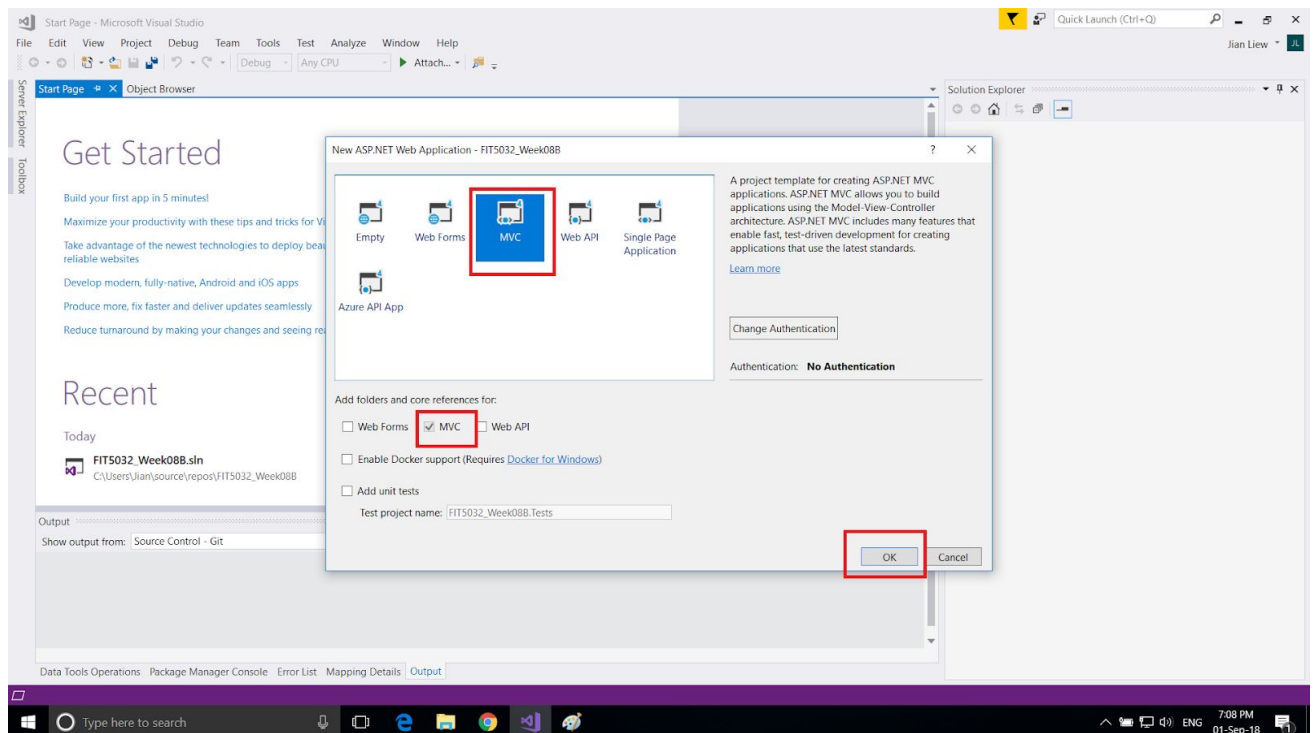
There is no need to complete this tutorial.

Step 1

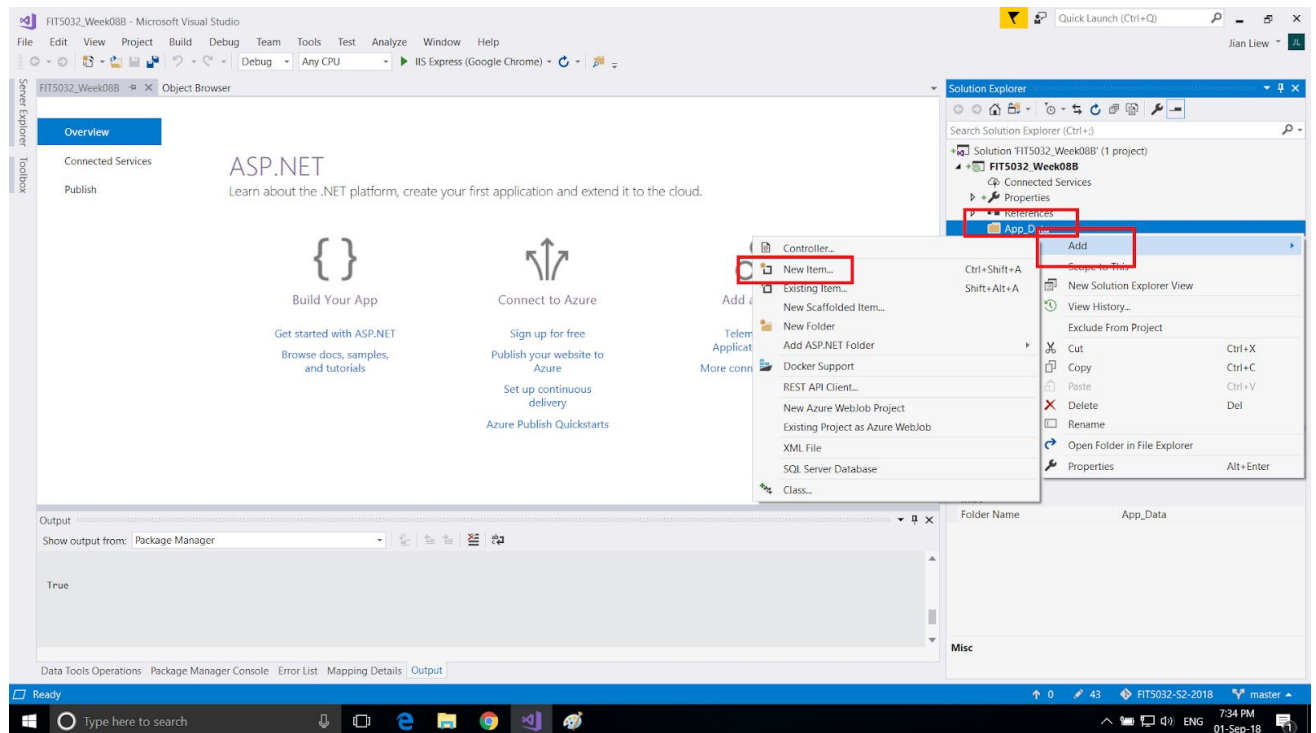
Please remember that it is possible to zoom into the images in this document.



Step 2

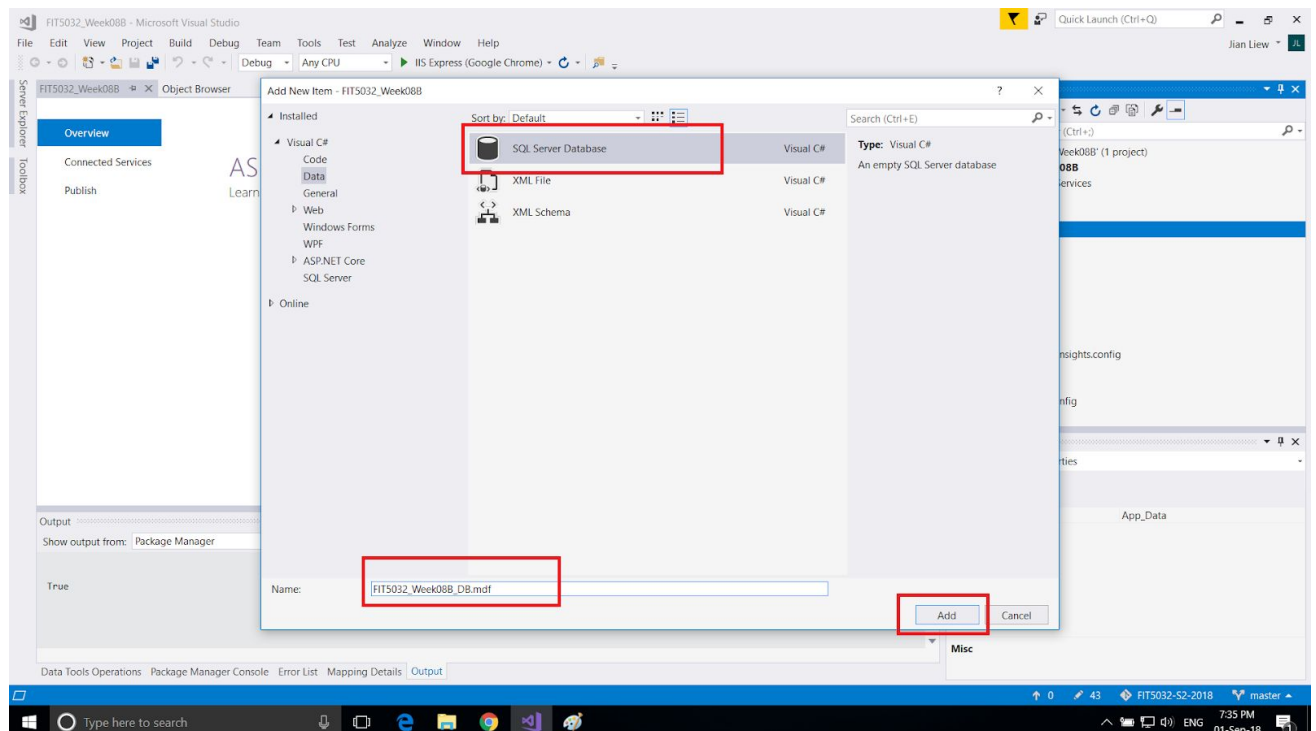


Step 3



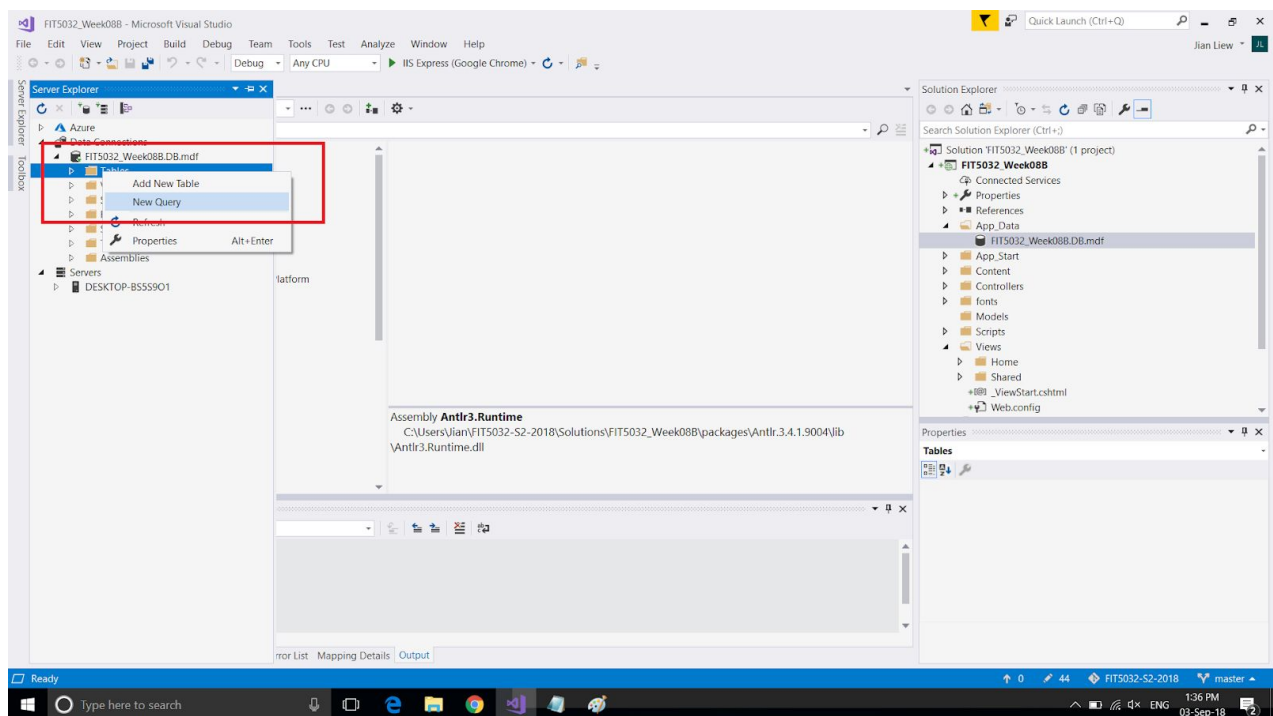
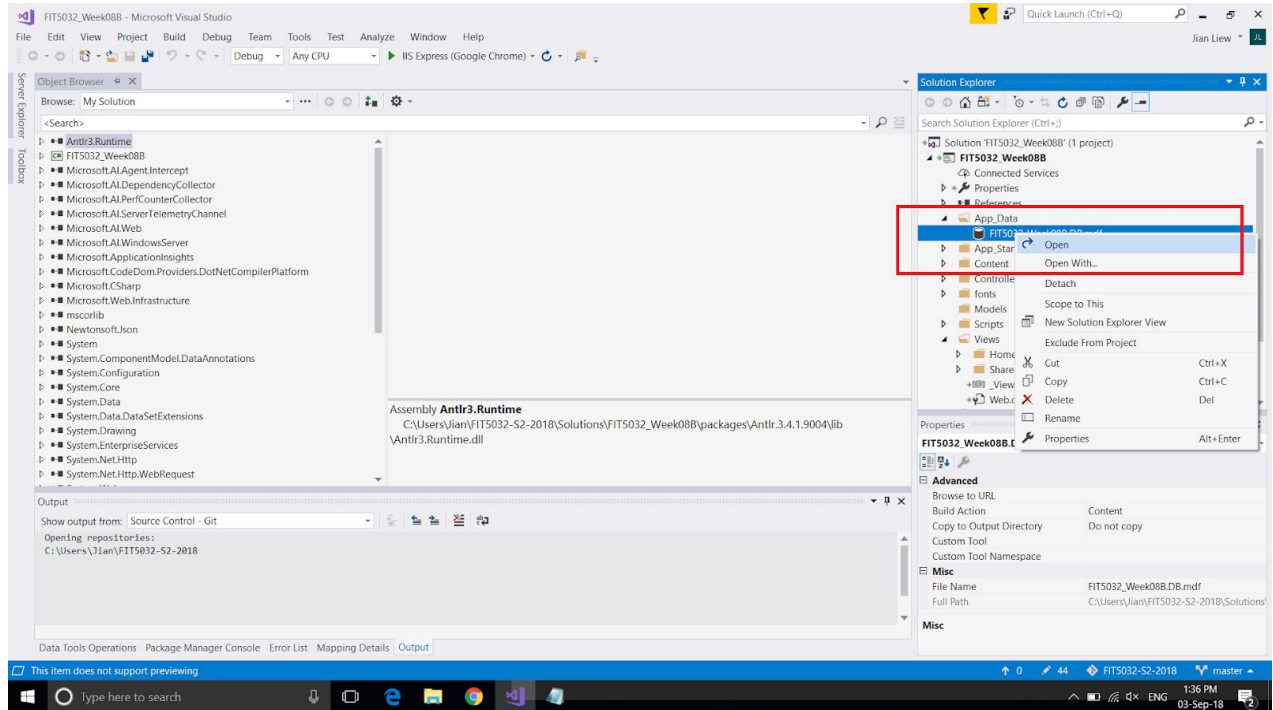
Step 4

I will use a database first approach to create a simple database to store my images.



Step 5

Here, there are several approaches that can be used. I can choose to store the images in my database as either a Binary Large Object (BLOB) or the path to the image itself, however the more common approach is to **store the path to the image itself on the server**. You can choose to use a BLOB instead if you want to however, it is not the preferred approach. If you have a file server, you can dedicate the server just for this purpose.



The DDL to create the table is provided as follows.

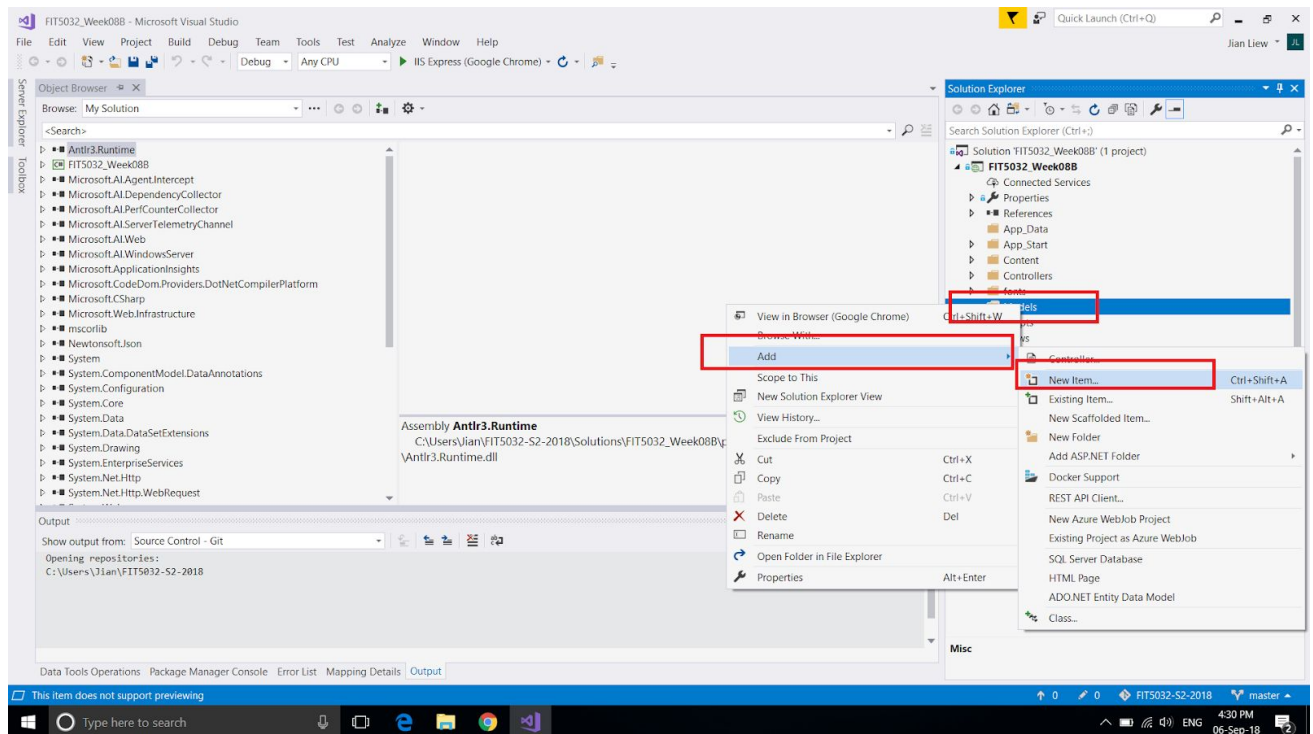
```
CREATE TABLE [dbo].[Images]
(
    [Id] INT NOT NULL PRIMARY KEY IDENTITY(1,1),
    [Path] VARCHAR(50) NOT NULL,
    [Name] VARCHAR(50) NOT NULL
)
```

Please remember to run the SQL query.

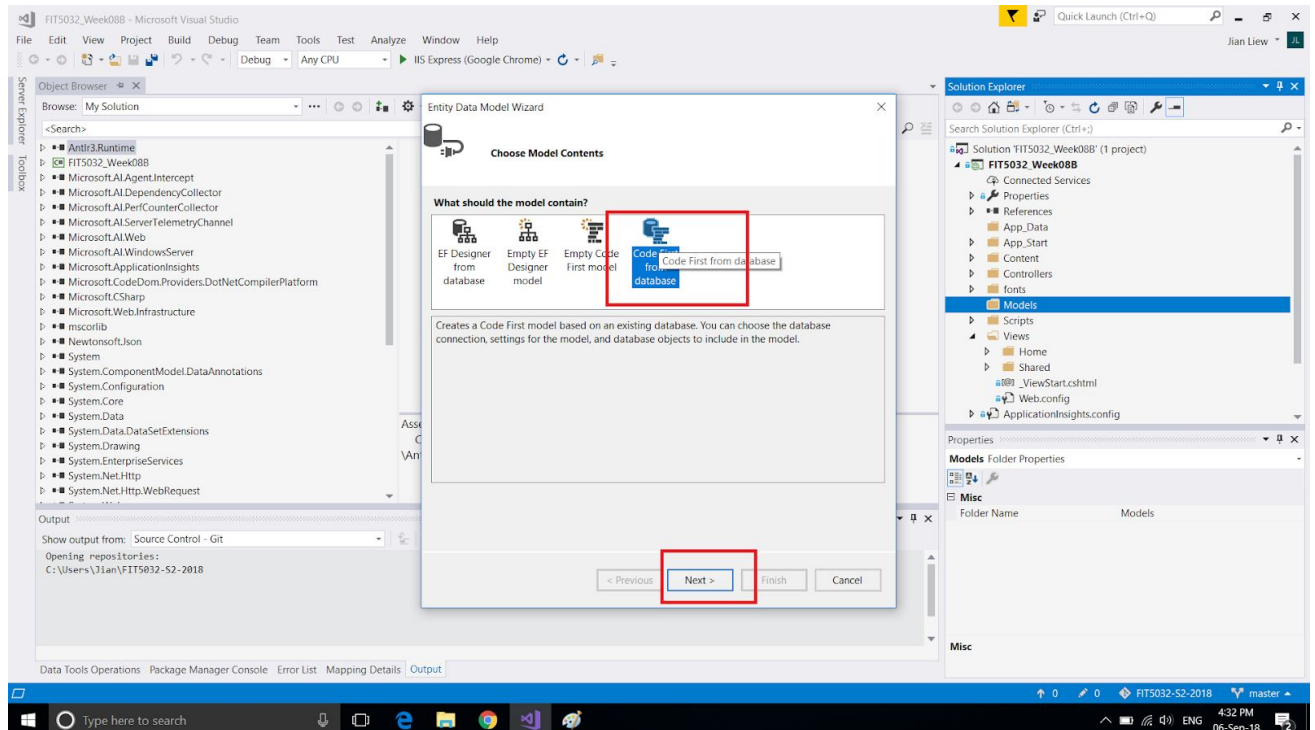
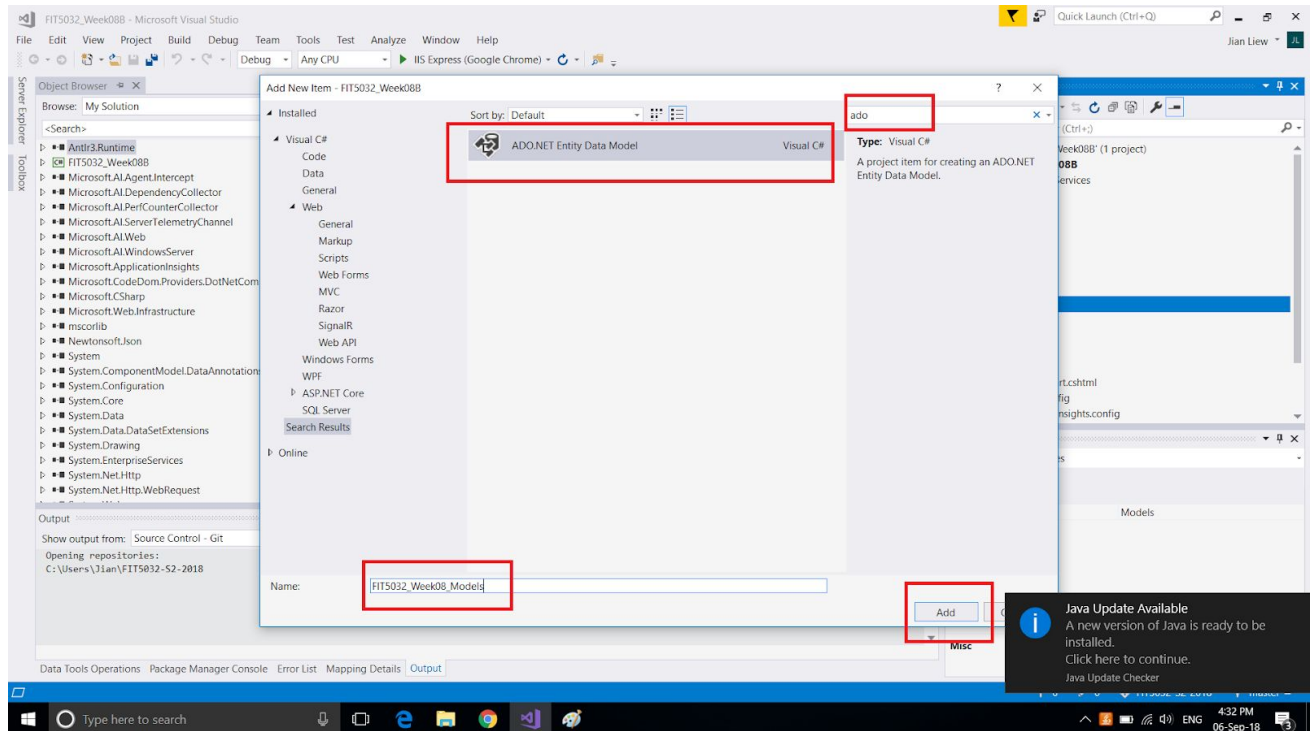
Step 6

After that you can use the automatic generation feature to create the models, controllers and views. However, you will need to make changes so that it is working as intended.

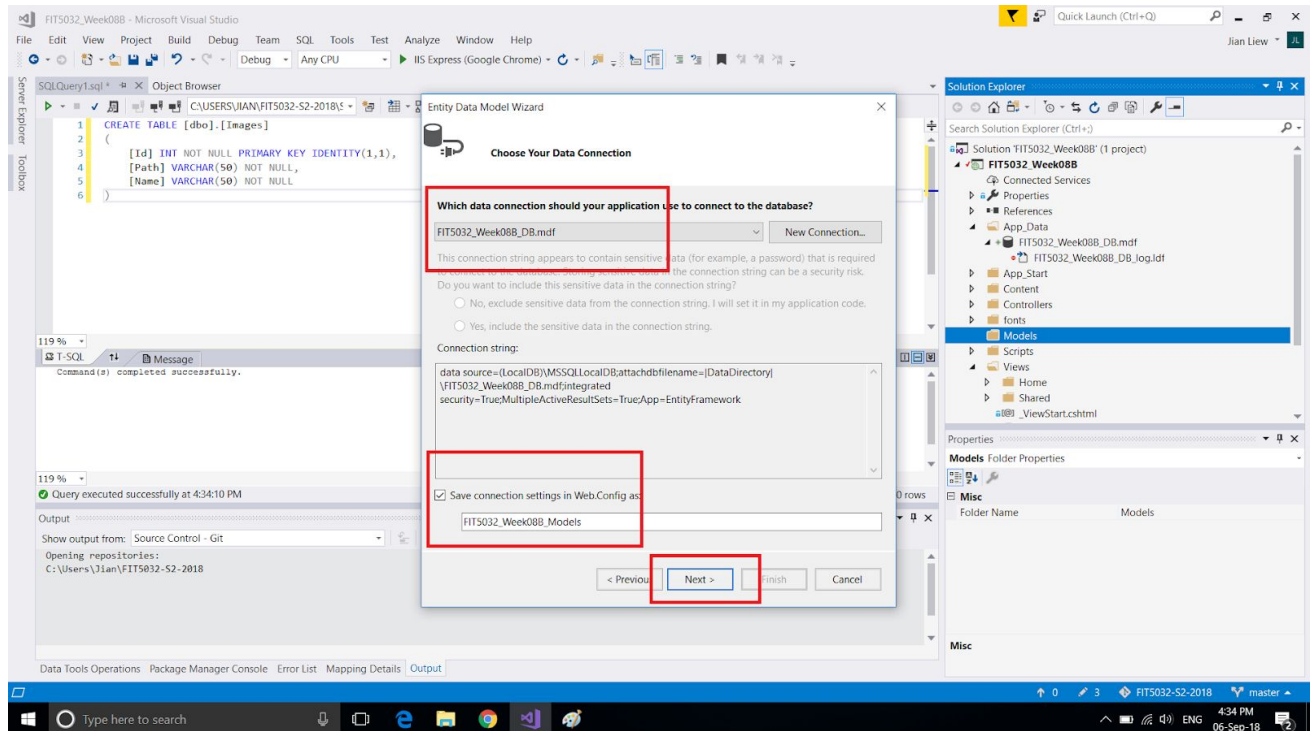
Right Click Models → Add → New Item



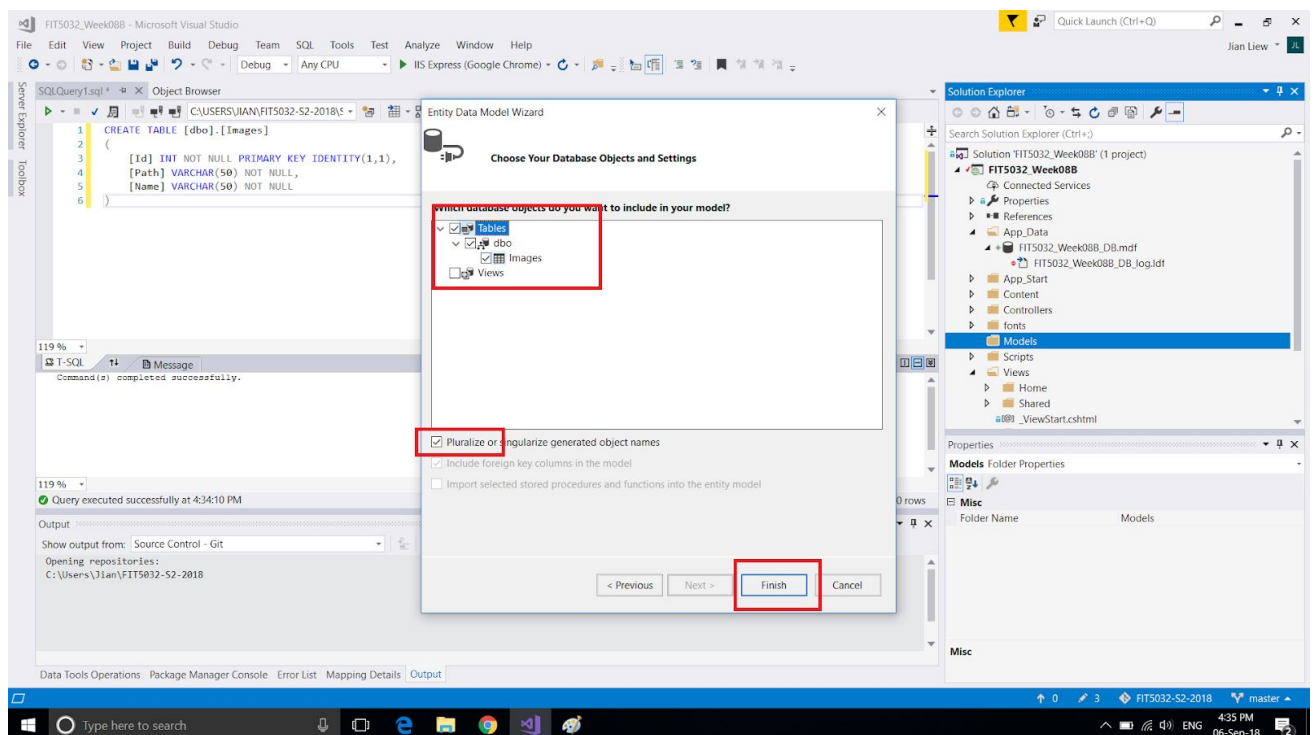
Step 7



Step 8

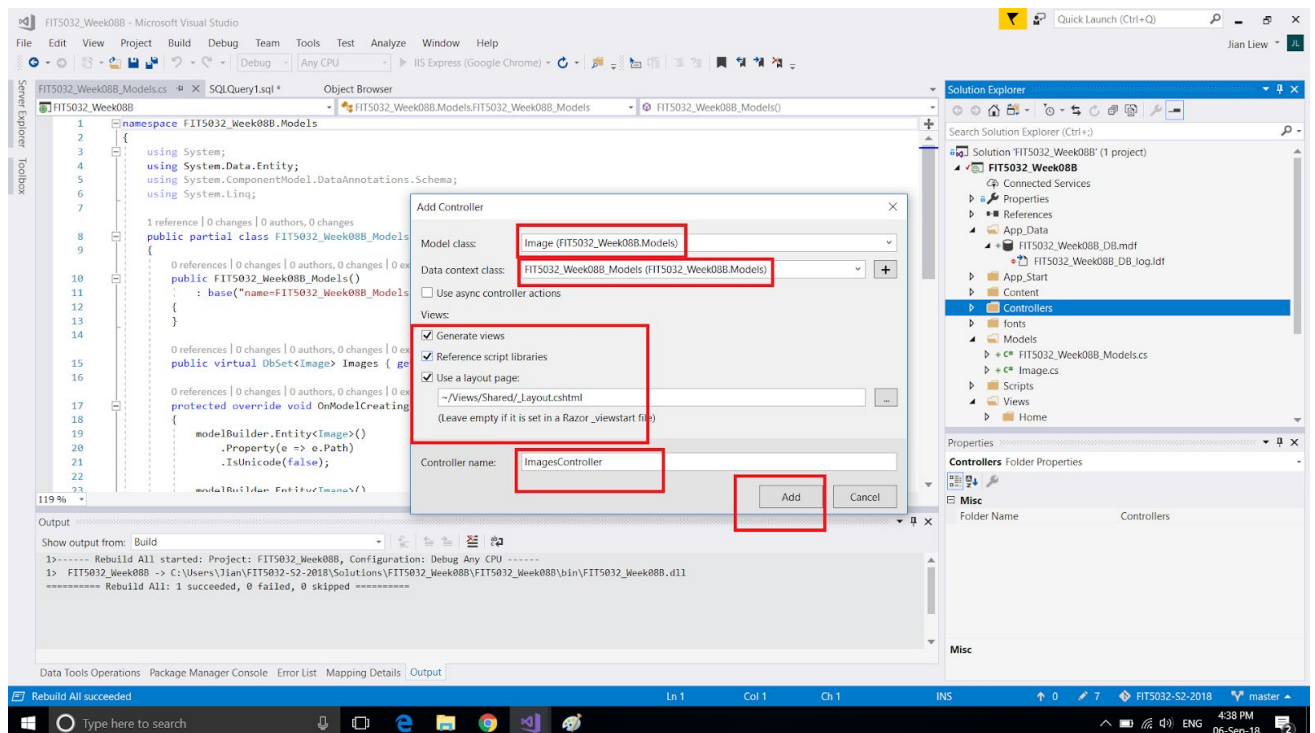
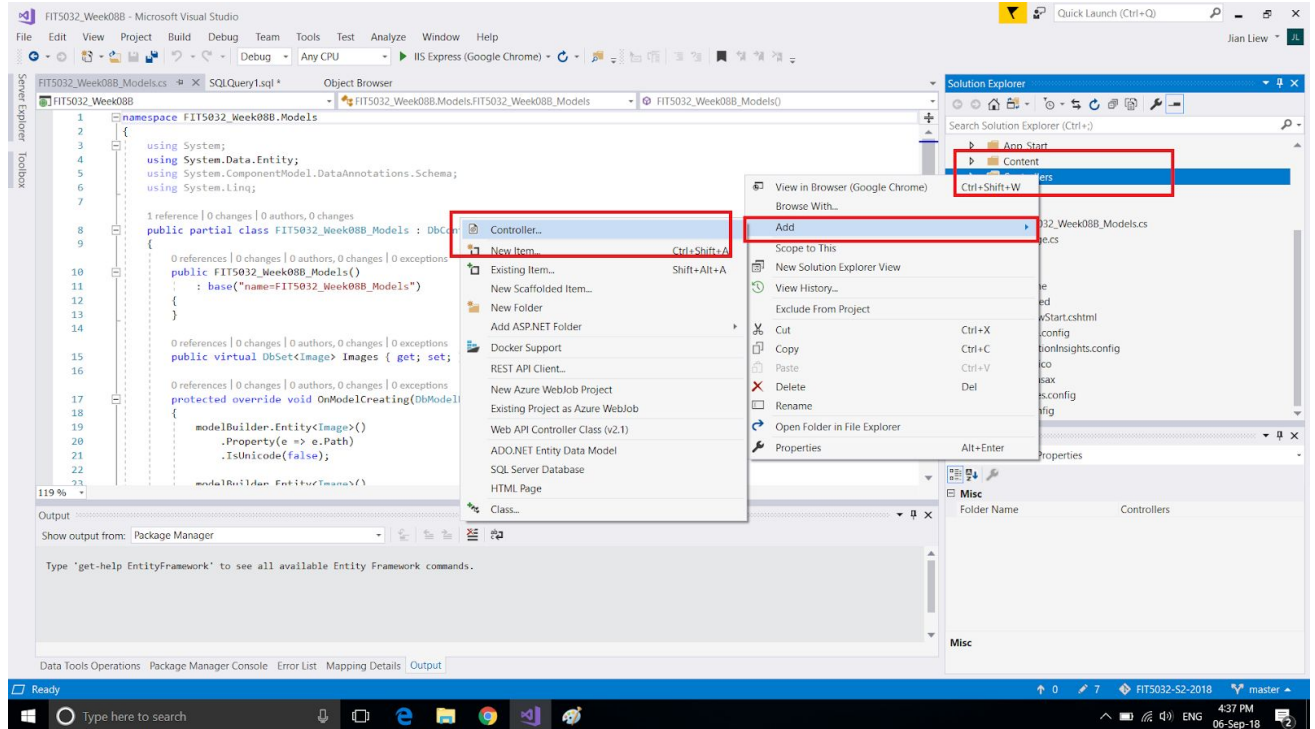


Step 9



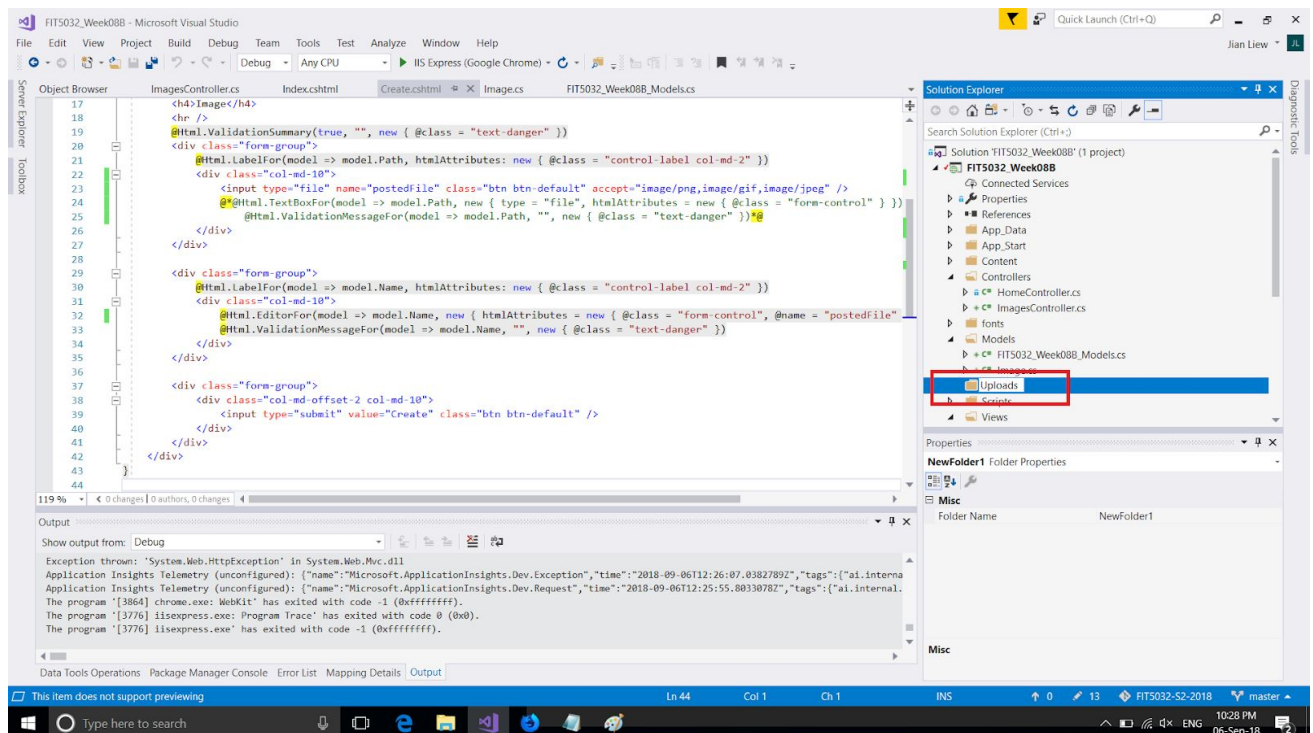
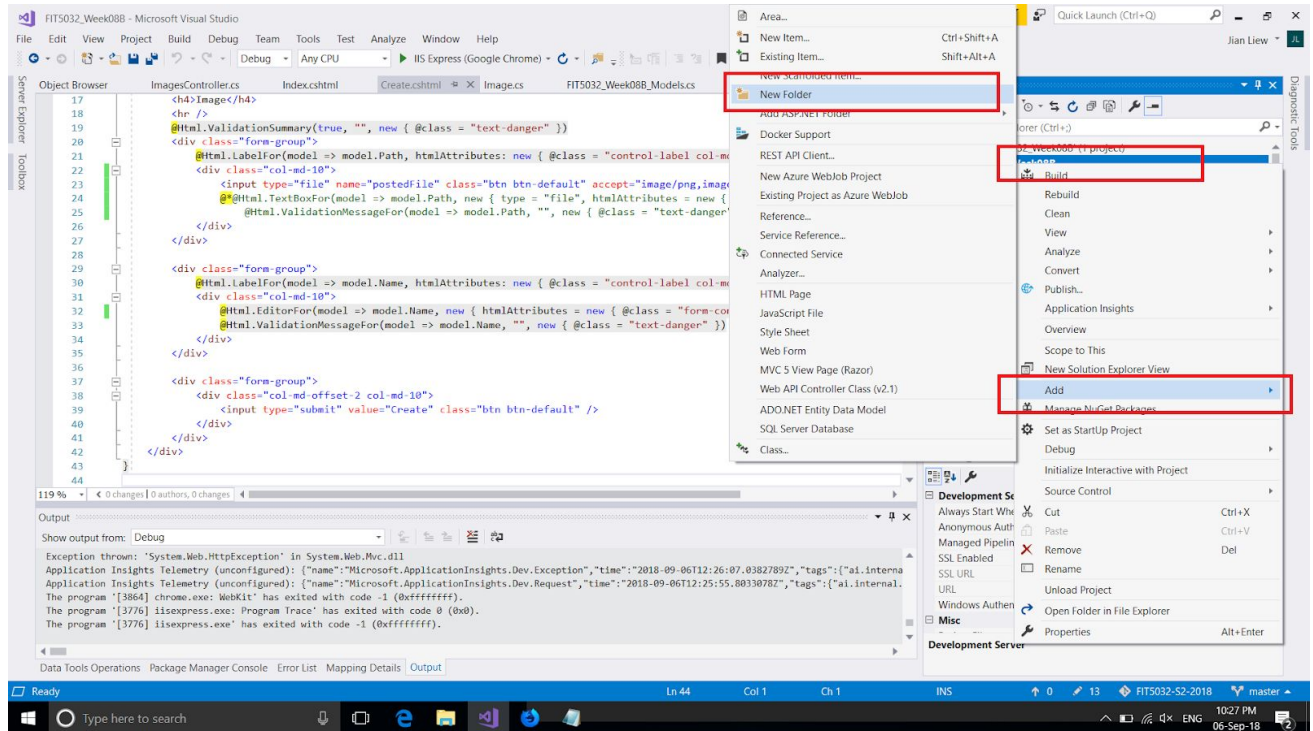
Step 10

Please remember to Rebuild your project after step 9.



Step 11

Create a folder called "Uploads", this folder will be used to store all the uploads.



Step 12

You will need to make the following changes.

ImagesController.cs

```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "Id,Name")] Image image,
    HttpPostedFileBase postedFile)
{
    ModelState.Clear();
    var myUniqueFileName = string.Format(@"{0}", Guid.NewGuid());
    image.Path = myUniqueFileName;
    TryValidateModel(image);

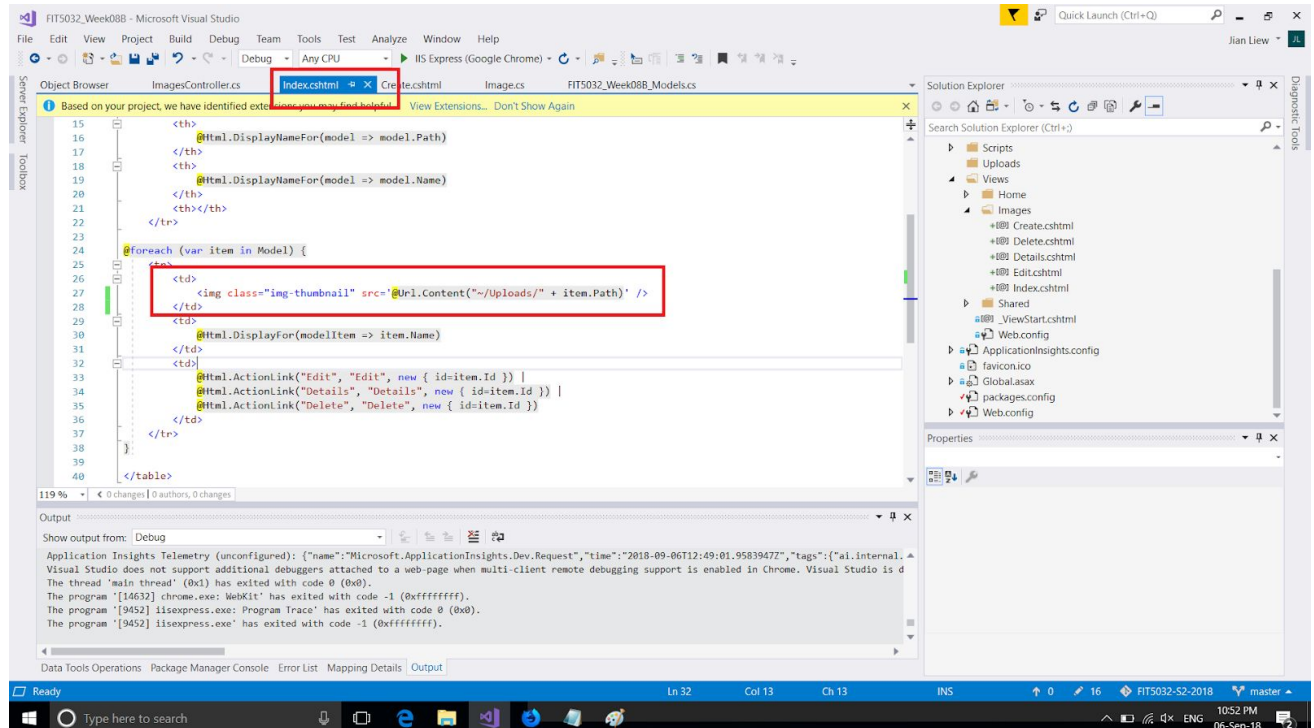
    if (ModelState.IsValid)
    {
        string serverPath = Server.MapPath("~/Uploads/");
        string fileExtension = Path.GetExtension(postedFile.FileName);
        string filePath = image.Path + fileExtension;
        image.Path = filePath;
        postedFile.SaveAs(serverPath + image.Path);

        db.Images.Add(image);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    return View(image);
}
```


Step 15

At the Index.cshtml (Of the Images View). Please type out the changes.

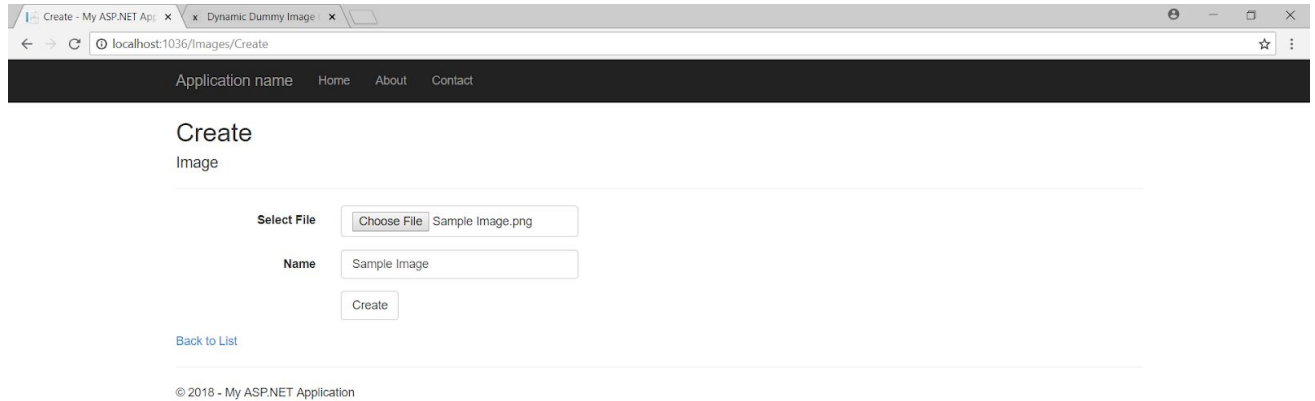


Step 16

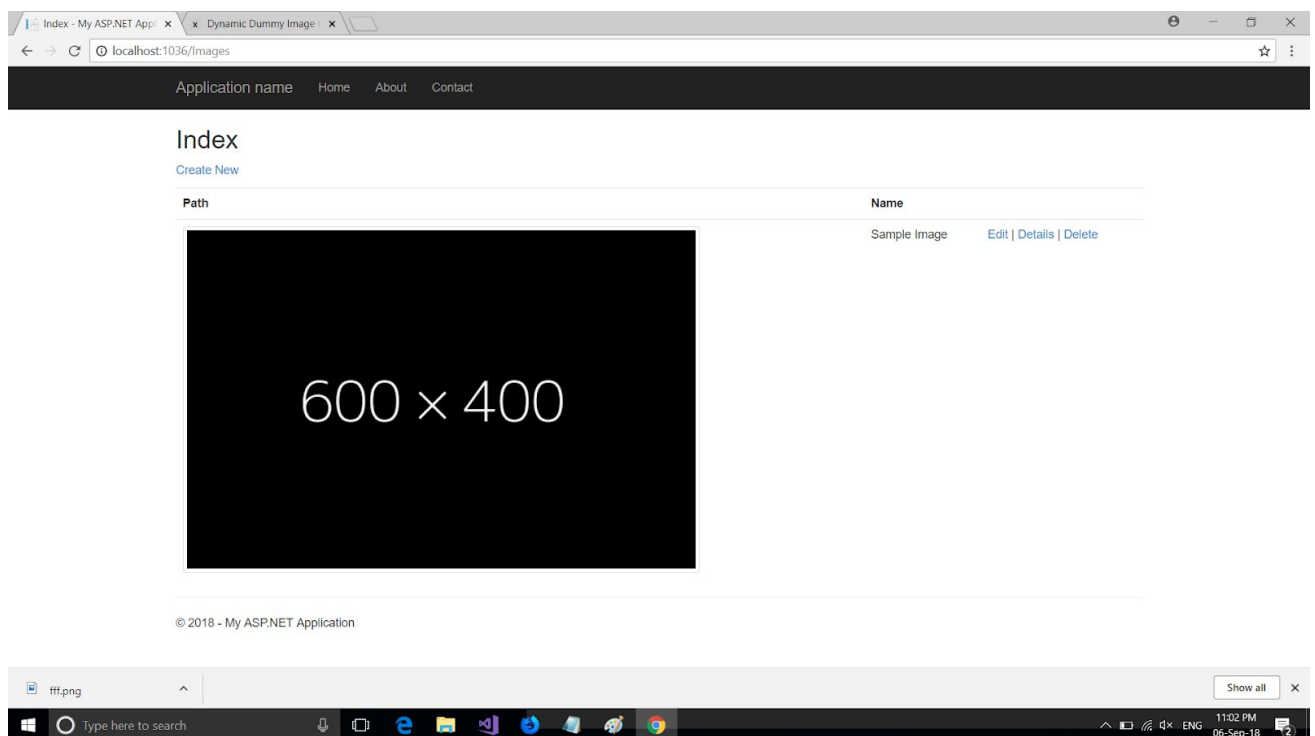
You can also introduce a hyperlink at the shared layout so that the user can navigate to the images view.

Upon the completion of this, you can try running the solution. You should see something like the images below.

At the /Images/Create



At the /Images/



Explanation

Here, I took a very simplistic approach to achieve this goal, and there are quite of numbers of issues not addressed. For example, you should introduce more validations which will validate the size of the image as well.

You can also figure out how to fix the "edit" feature so that it will be working as intended. If you want to, you can also further improve upon this project. If you want to know how to improve this project ask your tutor how to do so.

Conclusion

Upon the completion of this supplementary materials, you should be able to understanding how image upload is handled in a simple manner.

Another approach would be to use a BLOB to store the image itself in the database. However with this approach, it would be harder to see the images as it needs to be queried instead.