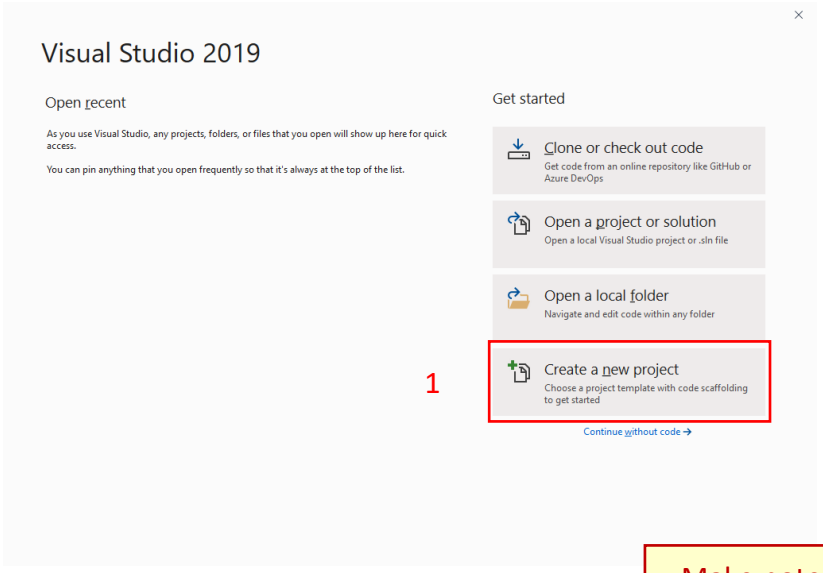
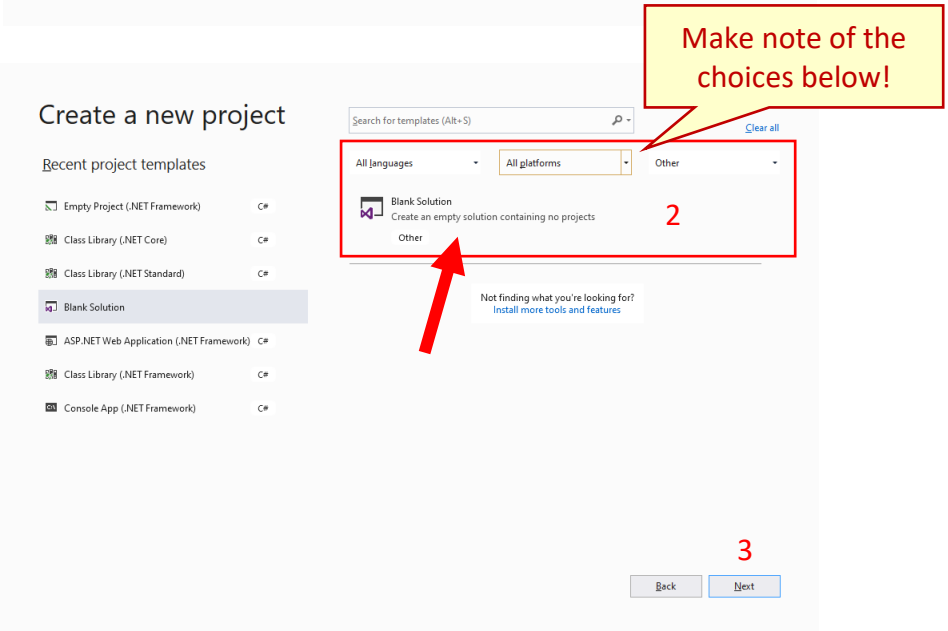
2.1.0 – Create Visual Studio Solution

Introduction

This is the first, practical, lesson for this course. In this lesson, you will learn how to properly create a multi-project (n-tier) Visual Studio solution. Your Visual Studio solution will have 2 Class Library projects and an ASP Website. In this lesson, you only need to focus on the details of how to properly create the solution as all other solutions, in-class, exercises, and Course Project, use these steps.

New Project





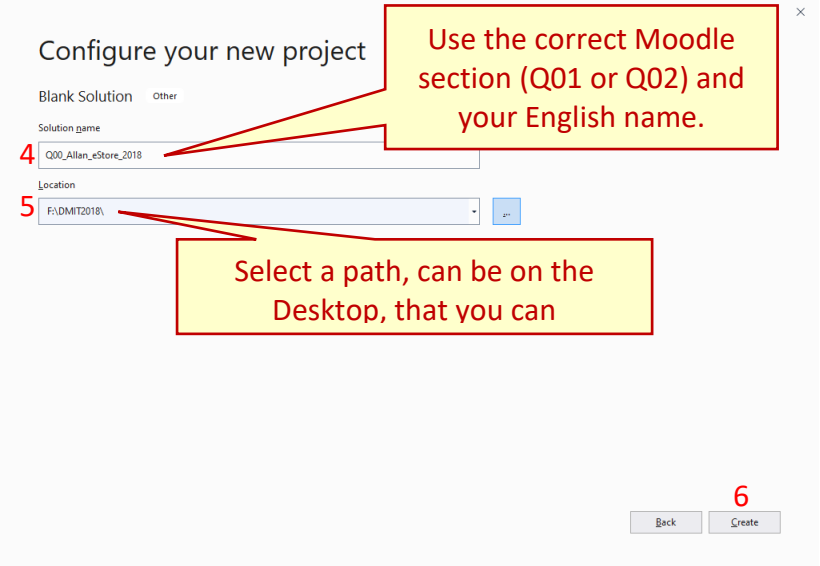


Figure 1: Create New Project Steps

The decisions you need to make are:

1. What is the Name of the entire solution? 4
2. Where would be the best Location for the solution? 5

Generally, the name of the solution will represent the business system you are creating the solution for, which should also match the name of the database you will be using. For this lesson, the Name will be: **Q0#_Name_eStore_2018** (replace the # with 1 or 2 [depends on which group you are in] and use your English name for Name).

The Location can be almost anywhere. By default, it will either be in the `Documents\Visual Studio 2019\Projects` folder, **`C:\Users\yourname\source\repos`**, or in the last solution folder you used. Some people say that you should NOT use the Documents folder, or the Desktop, or a USB drive. In this class, it is preferred that you store/save your work on your **OWN** USB Flash drive (which you **must** bring to class every day). On your USB Flash Drive, you can store/save your code solutions to the root of the drive or create a folder to put all your work in (it is up to you, but **YOU** need to REMEMBER where you stored/saved your work).

You are NOT permitted to share your code solutions with other students, or to use a code solution from another student. If your code solution does not work, for any reason, please see your instructor to get help to get your solution working again.

It is okay, to talk another student through the steps, but DO NOT do their work for them; they will only learn by writing the code themselves!

Once created, your Visual Studio has 0 projects:

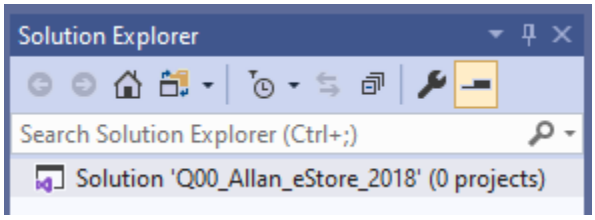


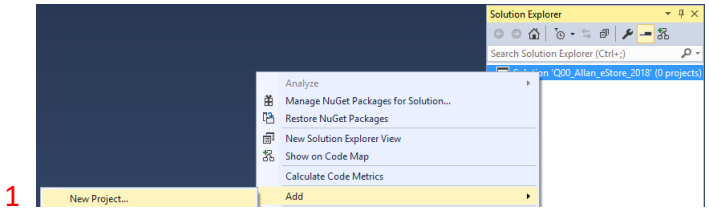
Figure 2: Blank Solution (0 Projects)

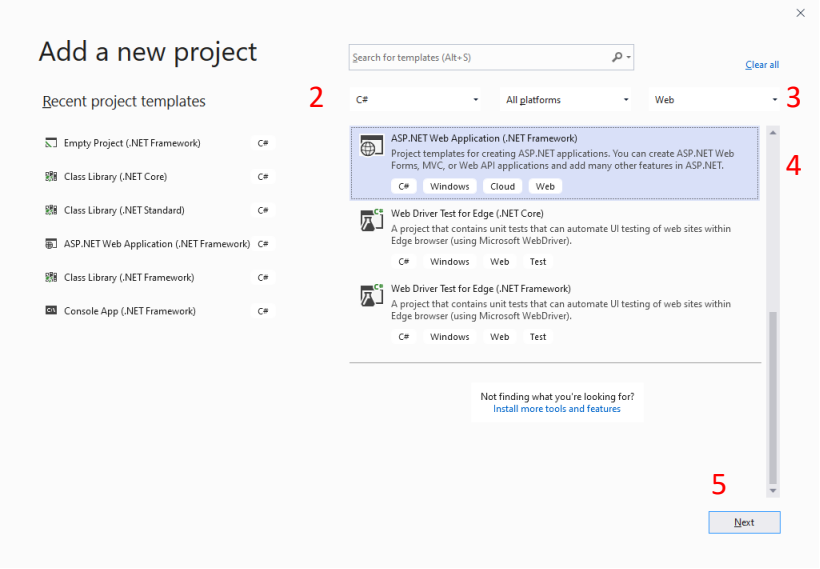
Solution Projects

You will be creating a multi-project (n-tier) solution. To do that you will need to add a web application project and 2 different Class Library projects to your Visual Studio solution. The first Class Library project will be for the Business Logic Layer (BLL) and the Data Access Layer (DAL). The second Class Library project will be for the Entities and Data classes for the solution.

Web Application Project

Right-click the solution name in the Solution Explorer and select Add, New Project:





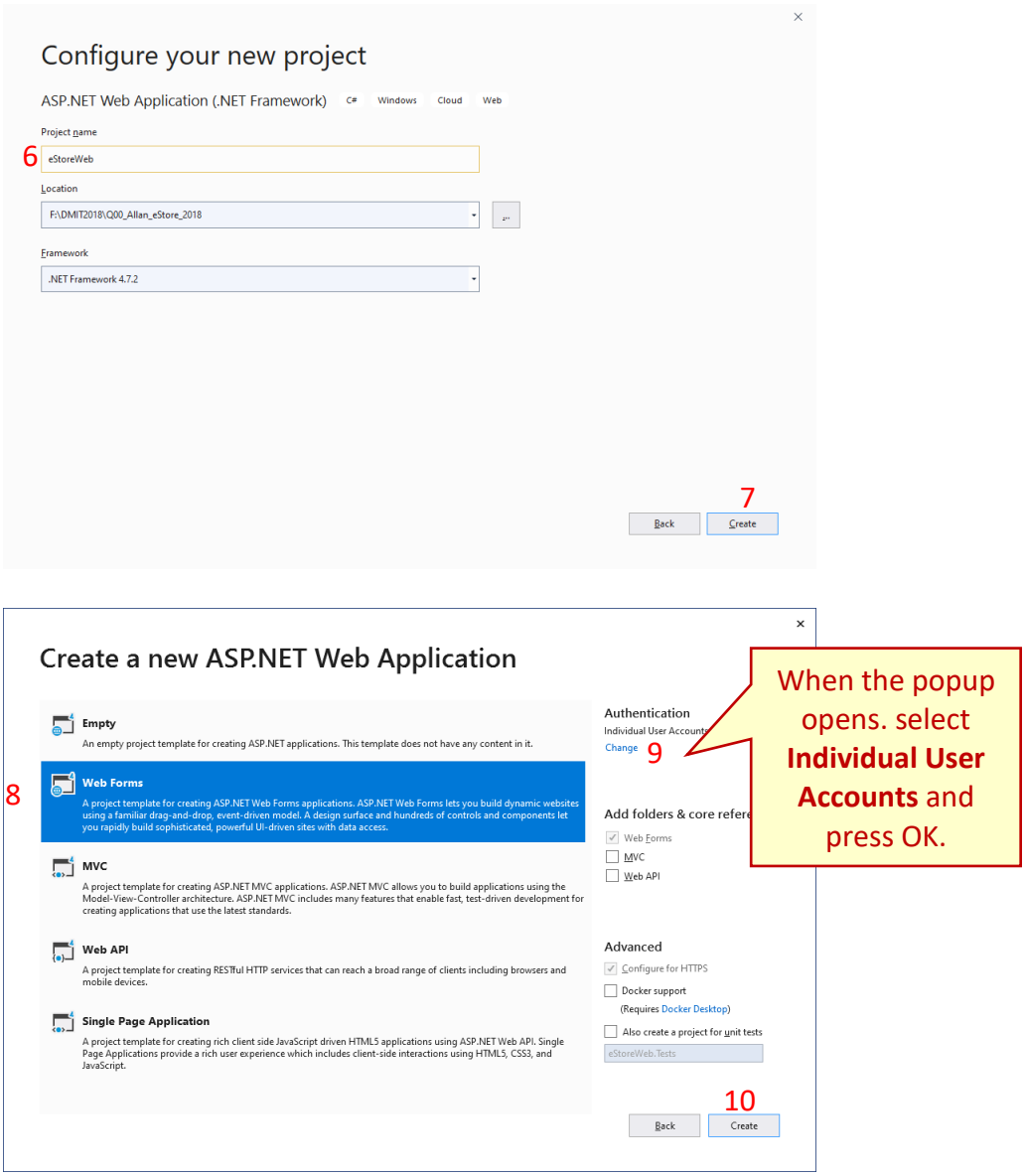


Figure 3: Add Class Web Application Project Steps

Change the Name with **eStoreWeb** (leave the Location as shown, as you need to have this project in the solution's physical file folder).

When you press **OK**, you will have the following screen. Make sure you select **Web Forms**:

The process to create the web application will take several minutes. When completed, you should see:

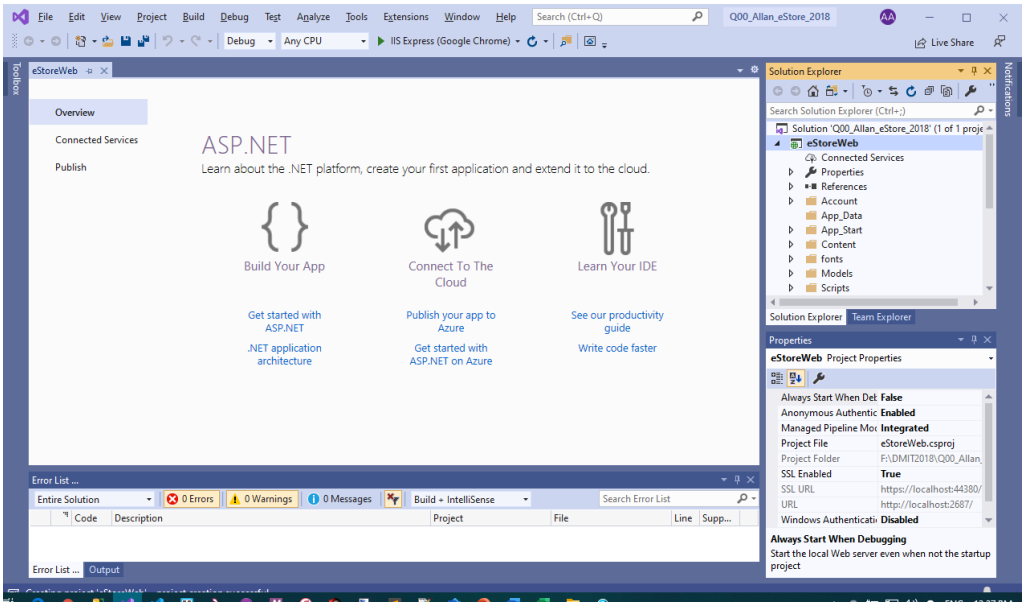
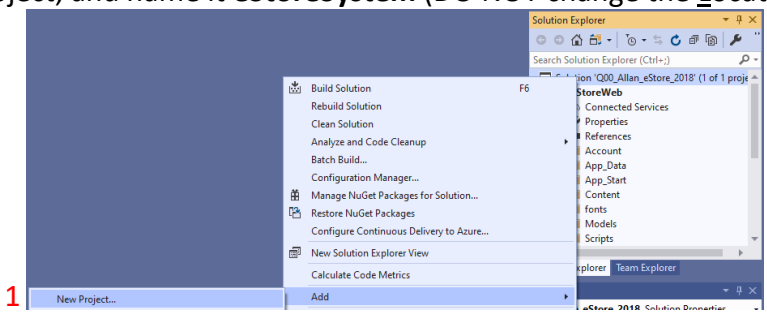


Figure 4: Web Application Project Created

You can close the **Your ASP.NET application** tab as this is just information.

Class Library Projects

Use the same process as before to add a new project to your solution. This time you will select a Class Library project, and name it **eStoreSystem** (DO NOT change the Location!):



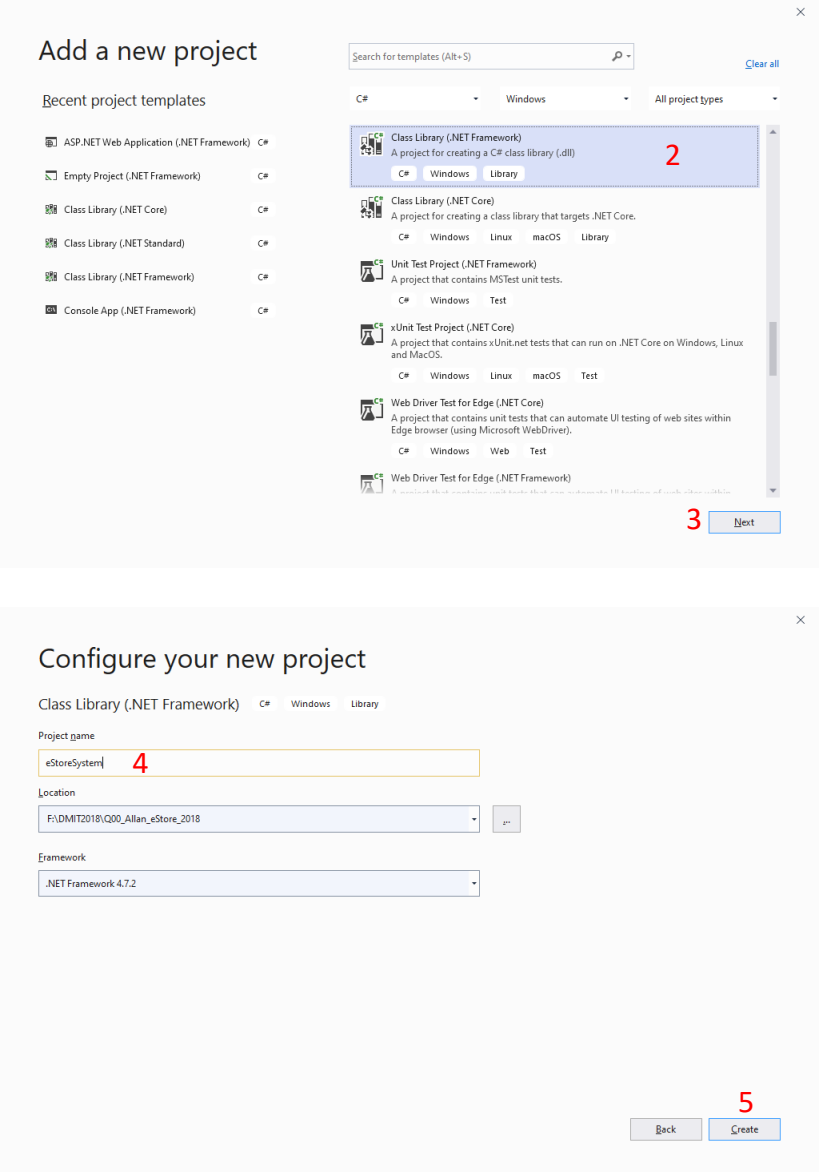


Figure 5: Add First Class Library Project Steps

You now have 2 projects in your solution. Note that Visual Studio created a file called `Class1.cs`.

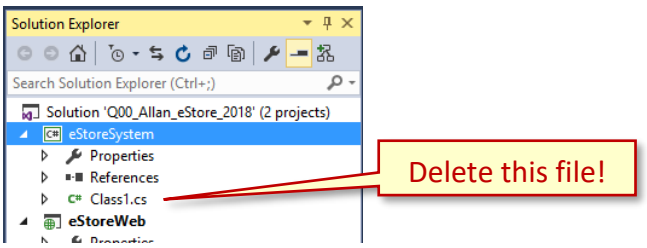


Figure 6: Class Library Project Added

This file **must** be deleted, as you will be doing more code organization and this file is not needed. What you need to do is to add 2 folders to your **eStoreSystem** project: BLL and DAL.

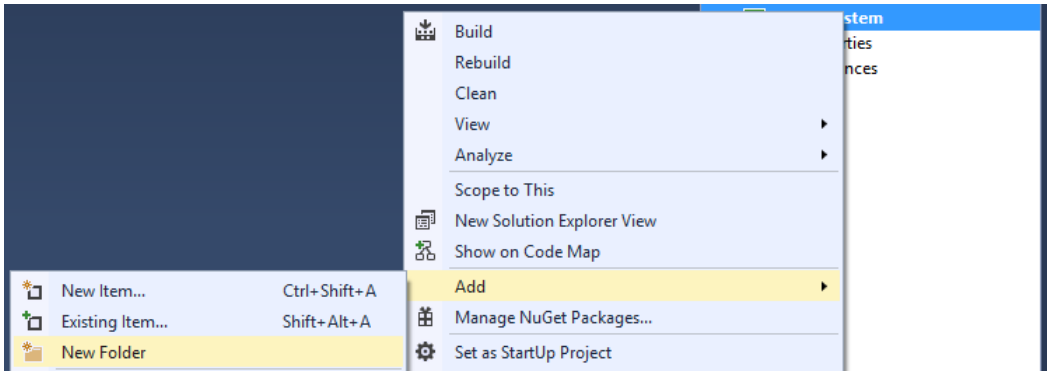


Figure 7: Add New Folder

Your Solution Explorer should now look like:

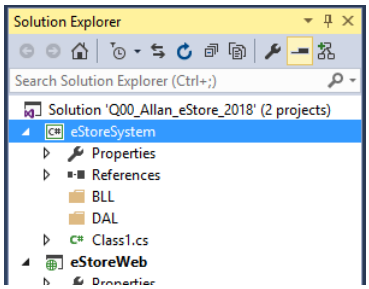


Figure 8: eStoreSystem with BLL and DAL Folders Added

Next, add another Class Library project called **eStoreData** using the same steps as above (DO NOT change the Location!). In this project create 2 folders: **Entities**, and **POCOs**. Remember to delete the `Class1.cs` code file. When finished, your solution folder should look like:

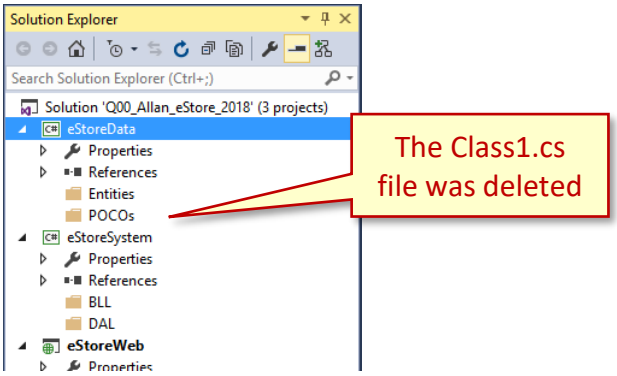


Figure 9: eStoreData Project Added with the Entities and POCOs Folders

Entities Folder

This folder will contain class code files that represent tables in the database.

POCOs Folder

A POCO is a **P**lain **O**ld **C**ommon **O**bject. These code files are NOT Entities, but custom, flat, data objects. No property of a POCO class is a collection.

There is just 1 thing that we **MUST** change: set the startup web page. Scroll the **Solution Explorer** window until you see a file called **Default.aspx**. Right-click the **Default.aspx** file and select **Set As Start Page**.

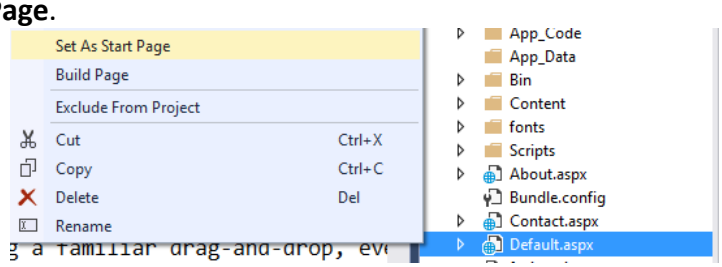


Figure 10: Set Default.aspx as Start Page

Now, the Solution Explorer should look like:

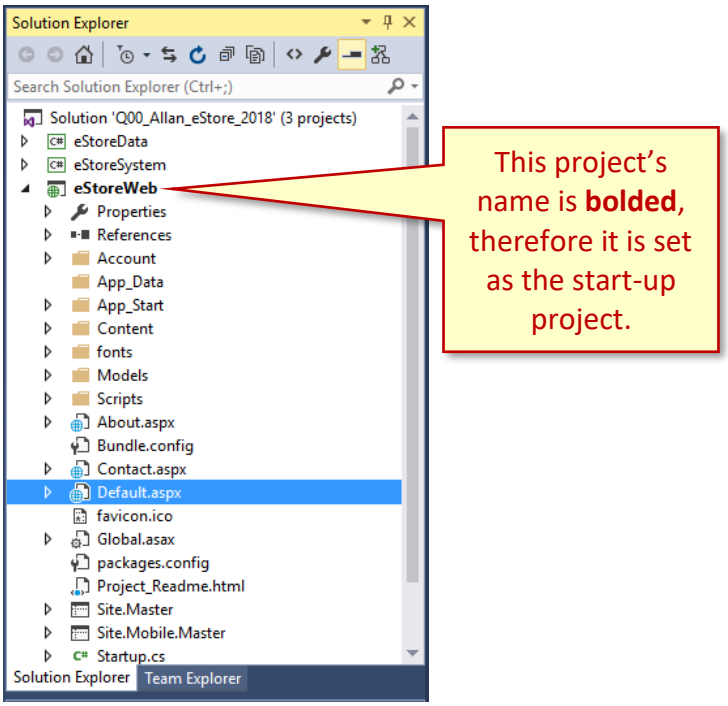


Figure 11: Solution Initial Setup Complete

Press the Save All button:

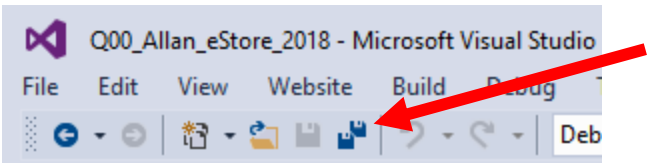


Figure 12: Save All

You could now close Visual Studio.

Exercise

Complete Exercise 2.1.1