| **National University of Computer and Emerging Sciences** |
| --- |
| Lab Manual 8  “ASP.NET Database Connectivity and Data Population” |

| **Database Systems Lab** |
| --- |
| **Spring 2023** |

Department of Computer Science

FAST-NU, Lahore, Pakistan

**Table of Content**

[Objectives 2](#_30j0zll)

[Task Distribution 2](#_1fob9te)

[Master Pages 3](#_3znysh7)

[Database Connectivity 11](#_2jxsxqh)

**Table of Figures**

[Figure 1: Create a new web application 3](#_2et92p0)

[Figure 2: Add new item from solution explorer 4](#_tyjcwt)

[Figure 3: Add a new master page named MyMaster.Master 4](#_3dy6vkm)

[Figure 4: Add an existing item called MyCssFile.Css 5](#_1t3h5sf)

[Figure 5: Add new folder named Images 6](#_4d34og8)

[Figure 6: Add existing items to a folder 6](#_2s8eyo1)

[Figure 7: Add images from Resources/images folder to Images folder of web application 7](#_17dp8vu)

[Figure 8: Add CSS file MyCssFile.css in MyMasterPage.Master 7](#_3rdcrjn)

[Figure 9: Modify MyMasterPage.Master 8](#_26in1rg)

[Figure 10: Add a new Home.aspx page 9](#_lnxbz9)

[Figure 11: Add MyMasterPage.Master to Home.aspx 9](#_35nkun2)

[Figure 12: Verify Home.aspx and Search.aspx in solution explorer 10](#_1ksv4uv)

[Figure 13: Execute project to check home page and search page 10](#_44sinio)

[Figure 14: Connection string values](about:blank) 12

[Figure 15: Add connection string to Web.config file 12](#_3whwml4)

[Figure 16: Add a new myDAL.cs class in a newly created DAL folder 13](#_2bn6wsx)

[Figure 17: Add namespaces and reference to connection string in myDAL.cs file 14](#_qsh70q)

[Figure 18: Modify myDAL.cs file with SelectItem() function to get all items 15](#_3as4poj)

[Figure 19: Modify Home.aspx page and add a GridView 16](#_1pxezwc)

[Figure 20: Modify Home.aspx.cs file to load data in GridView 17](#_49x2ik5)

[Figure 21: Execute project to view the loaded GridView](about:blank) 18

[Figure 22: Modify Search.aspx page to add search button and GridView](about:blank) 19

[Figure 23: Add SearchItem() in myDAL.cs file to search single item](about:blank) 20

[Figure 24: Modify Search.aspx.cs file to apply server side logic](about:blank) 21

[Figure 25: Search item Soap in the database](about:blank) 22

[Figure 26: Search for an item not present in the database](about:blank) 22

# Objectives

* Understanding ASP.NET Master Pages
* Database connectivity with ASP.Net web forms application
* Populating data in GridView using SQL query and stored procedures

# Task Distribution

| Total Time | 60 Minutes |
| --- | --- |
| Introduction | 10 Minutes |
| Demo + Practice | 50 Minutes |

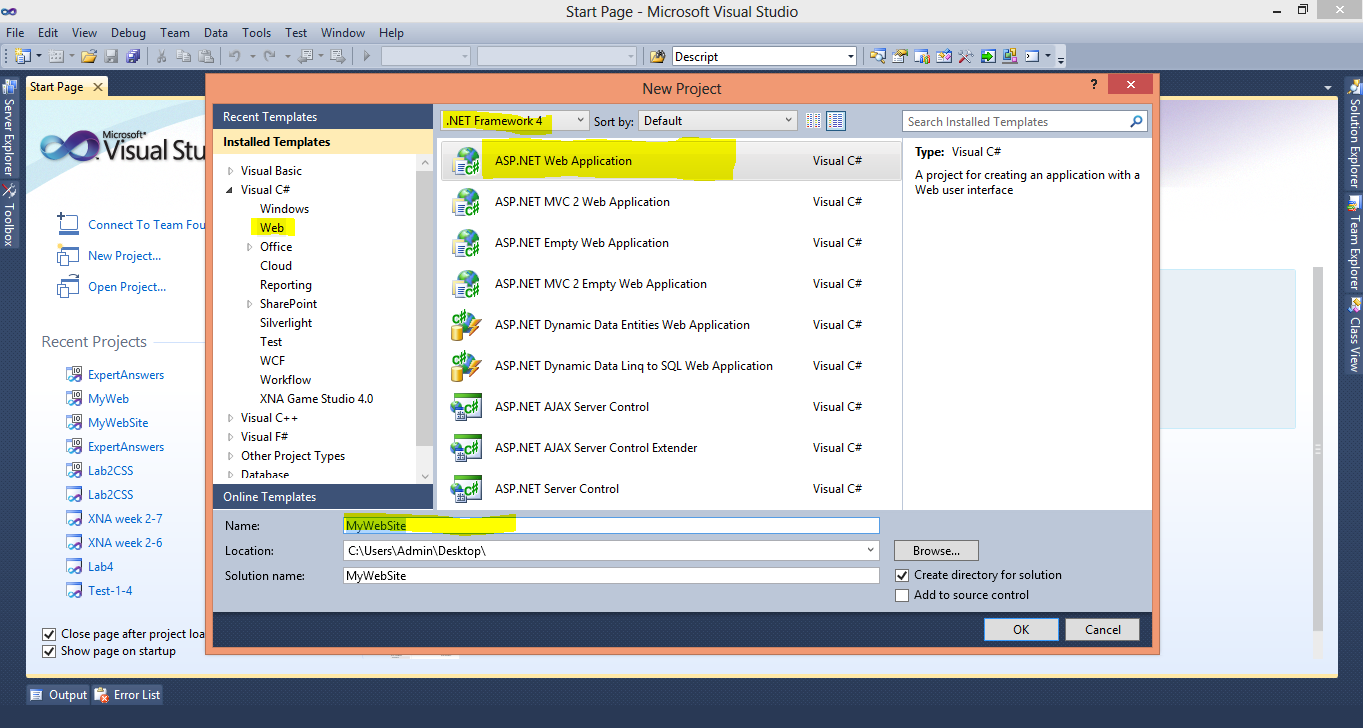
# Master Pages

* Master pages allow you to create a consistent look and behavior for all the pages (or group of pages) in your web application.
* A master page provides a template for other pages, with shared layout and functionality.
* They allow you to centralize the common functionality of your pages so that you can make updates in just one place.
* The master page defines placeholders for the content, which can be overridden by content pages.
* The output result is a combination of the master page and the content page.
* The content pages contain the content you want to display.
* When users request the content page, ASP.NET merges the pages to produce output that combines the layout of the master page with the content of the content page.
* File name extension **.master**

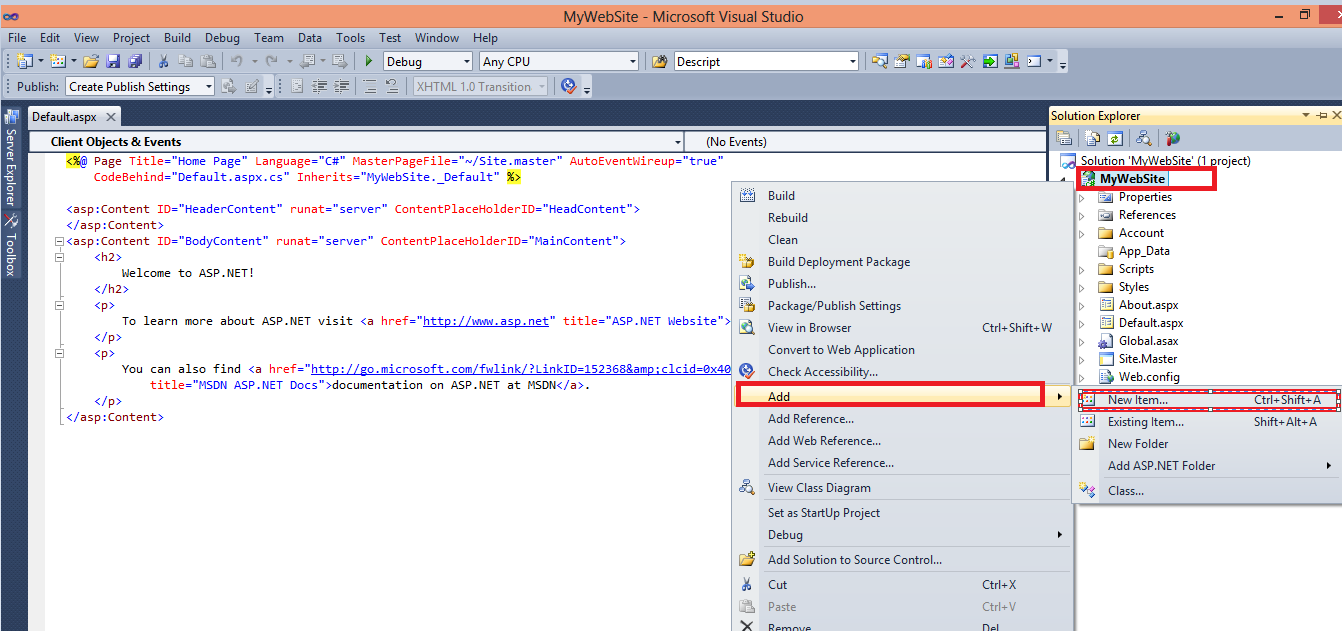
For more info visit:

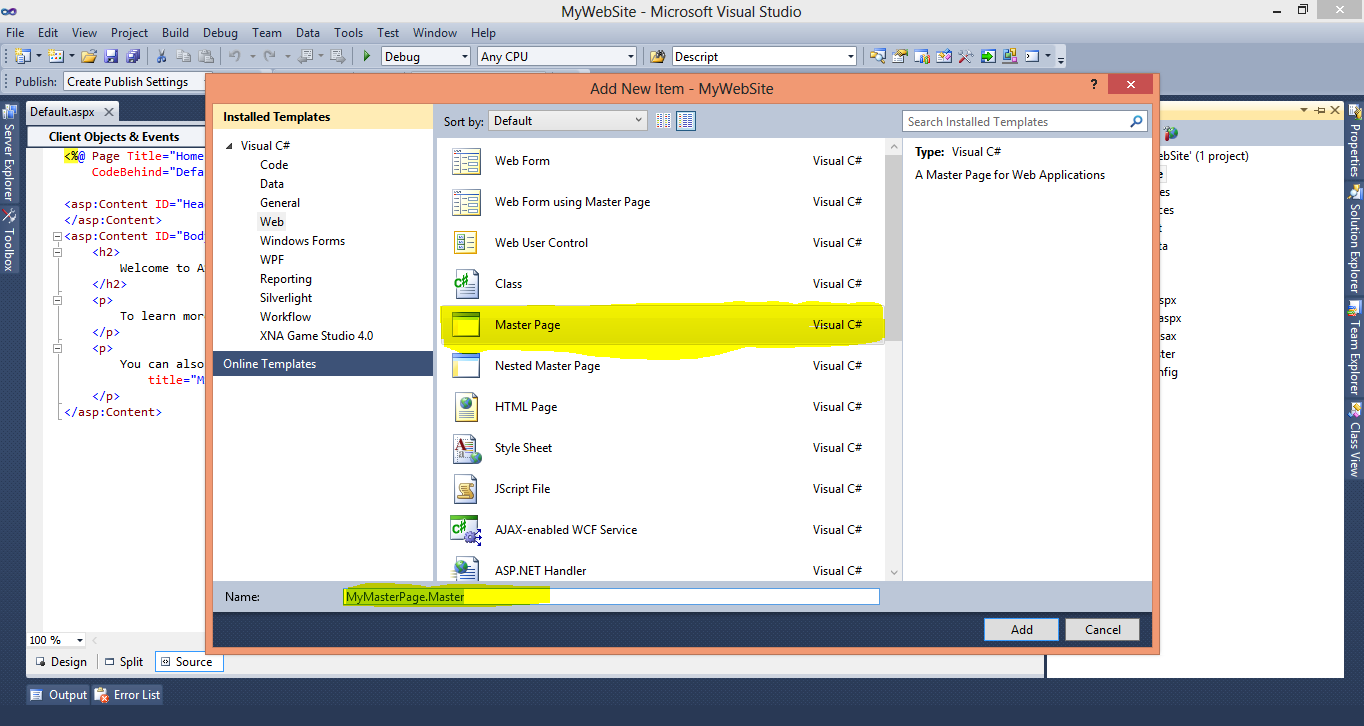
<https://docs.microsoft.com/en-us/previous-versions/aspnet/wtxbf3hh(v=vs.100)>

**The following exercise will demonstrated how to create and use Master Pages**

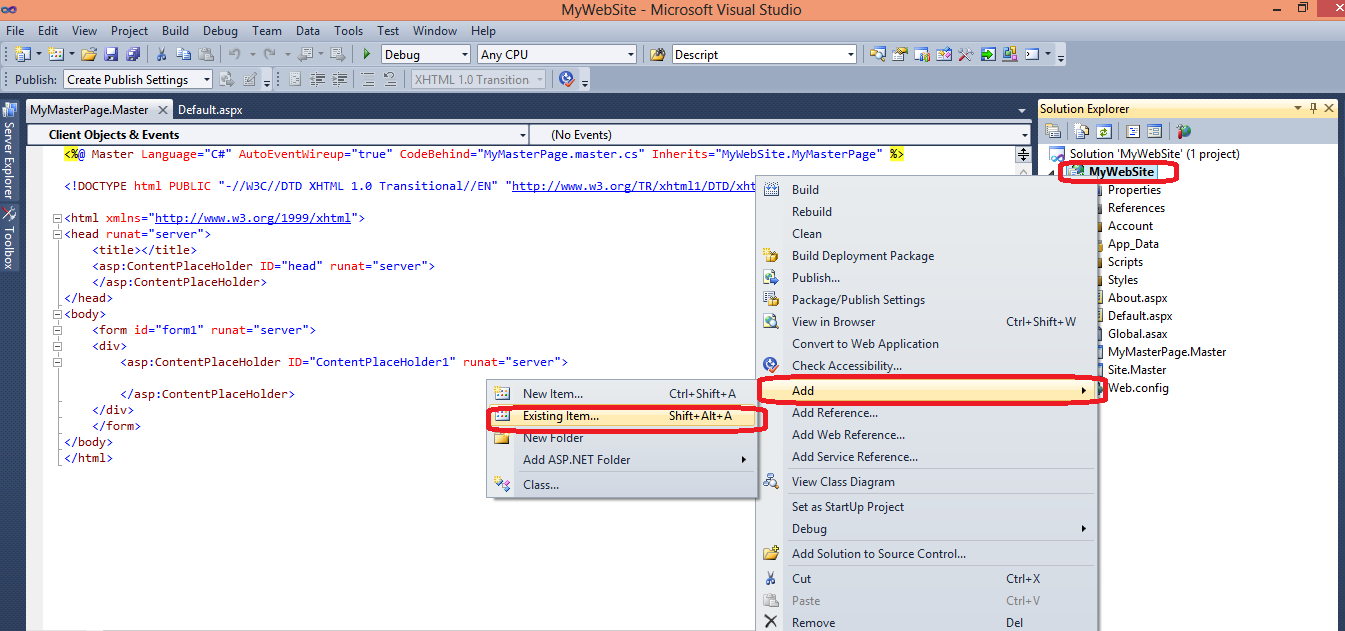
1. Create a New Project in Visual Studio, selecting ASP.Net Web Application and .Net Framework 4 (your latest version) in middle pane and Visual C# Web in left pane. Name your project **MyWebSite** (follow figure 1)

*Figure 1: Create a new web application*

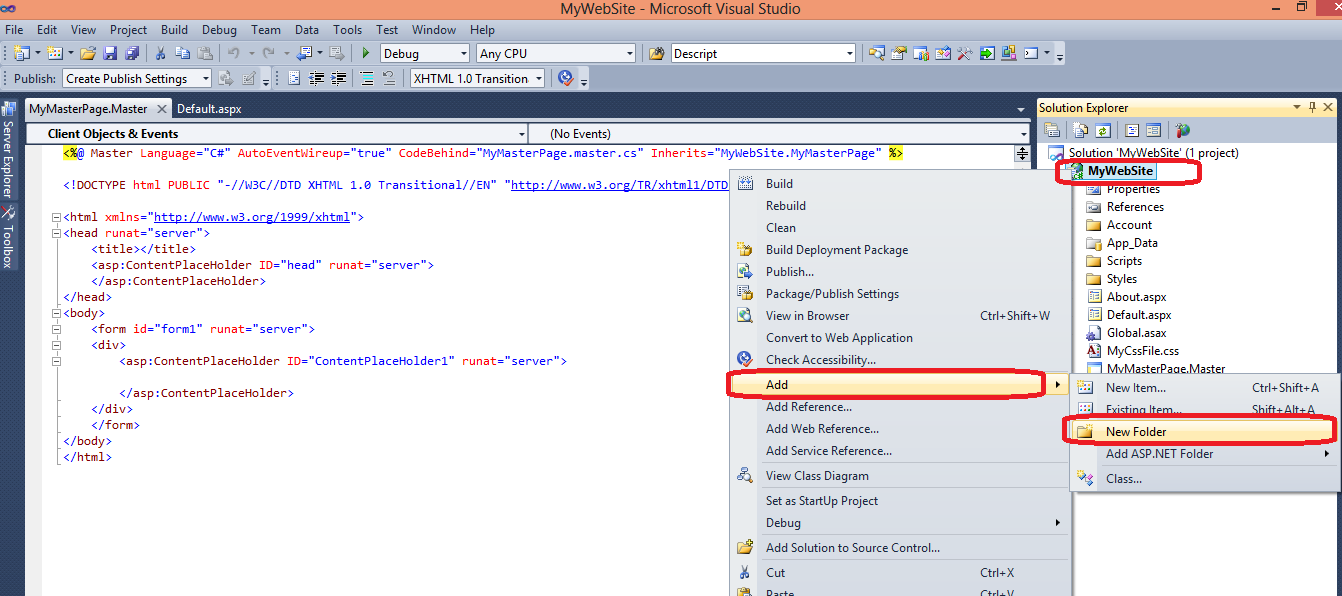
1. Add Master Page in your Project, using Add New Item option from Solution Explorer (follow figure 2 and 3). Name this Master Page as **MyMasterPage.Master.**

*Figure 2: Add new item from solution explorer*

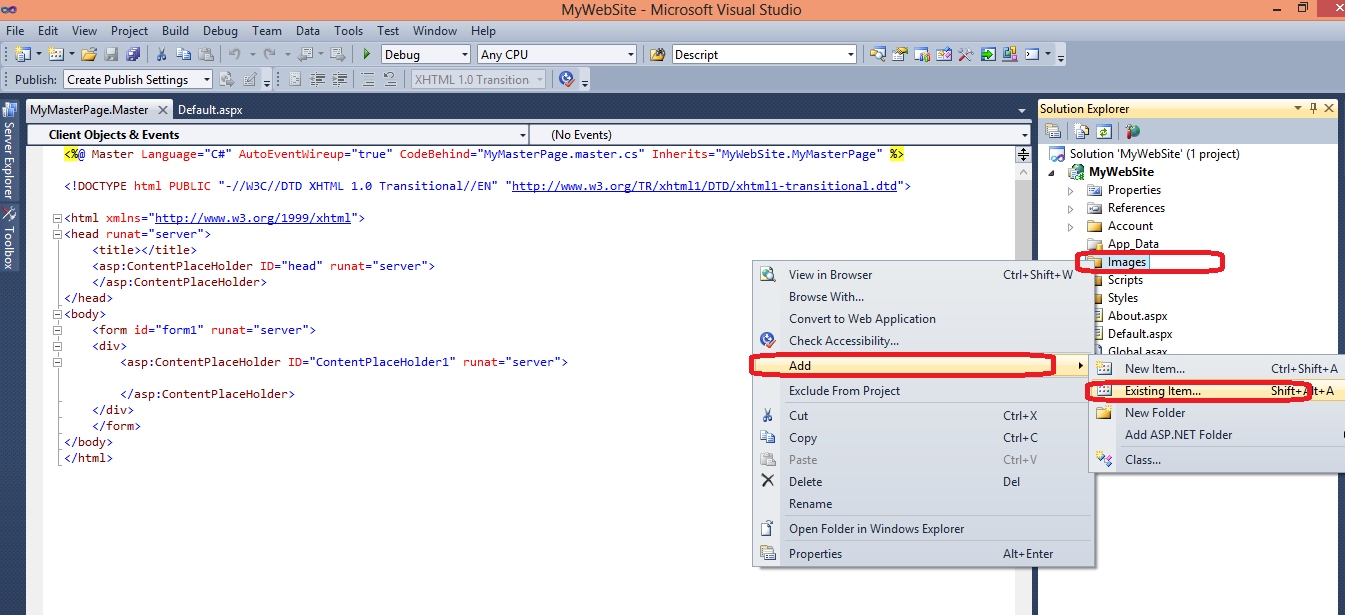
*Figure 3: Add a new master page named MyMaster.Master*

1. Now, we will add CSS file and images in our project that will be used in styling the Master Page. We have used the snow\_glass\_215 template given in the link below.
2. Add the CSS file (**MyCssFile.CSS** given in Resource folder along with manual) in your project using Add Existing Item (follow figure 4)

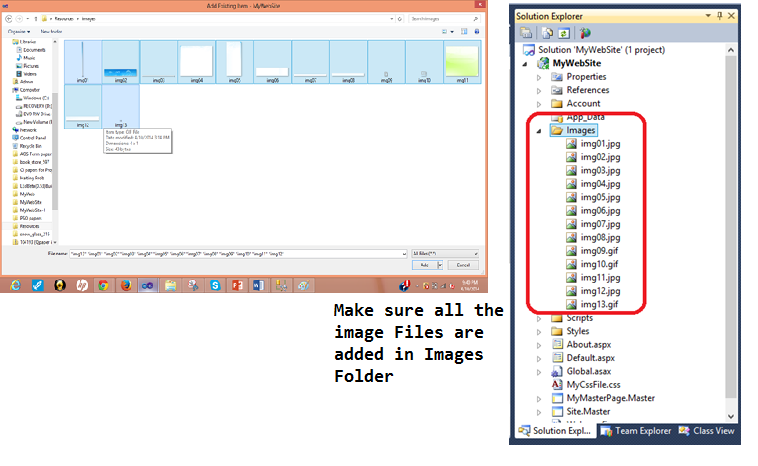
*Figure 4: Add an existing item called MyCssFile.Css*

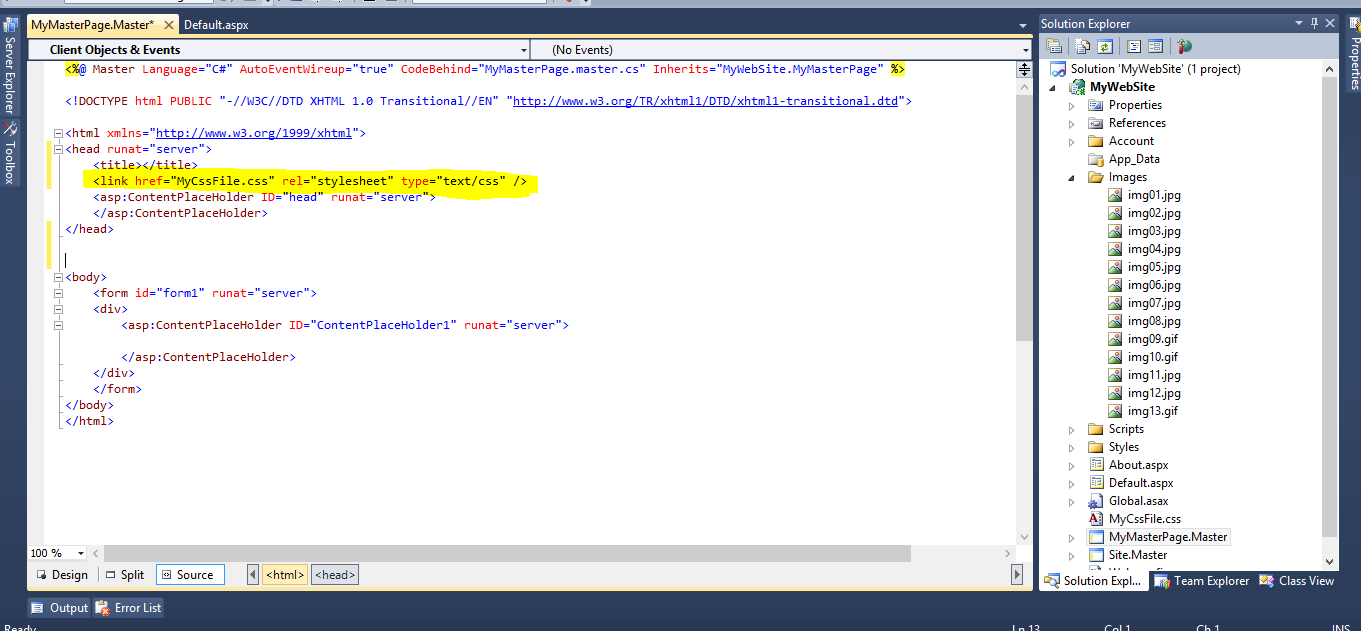
1. After adding CSS file create new folder in your project named Images using New Folder option (follow figure 5)

*Figure 5: Add new folder named Images*

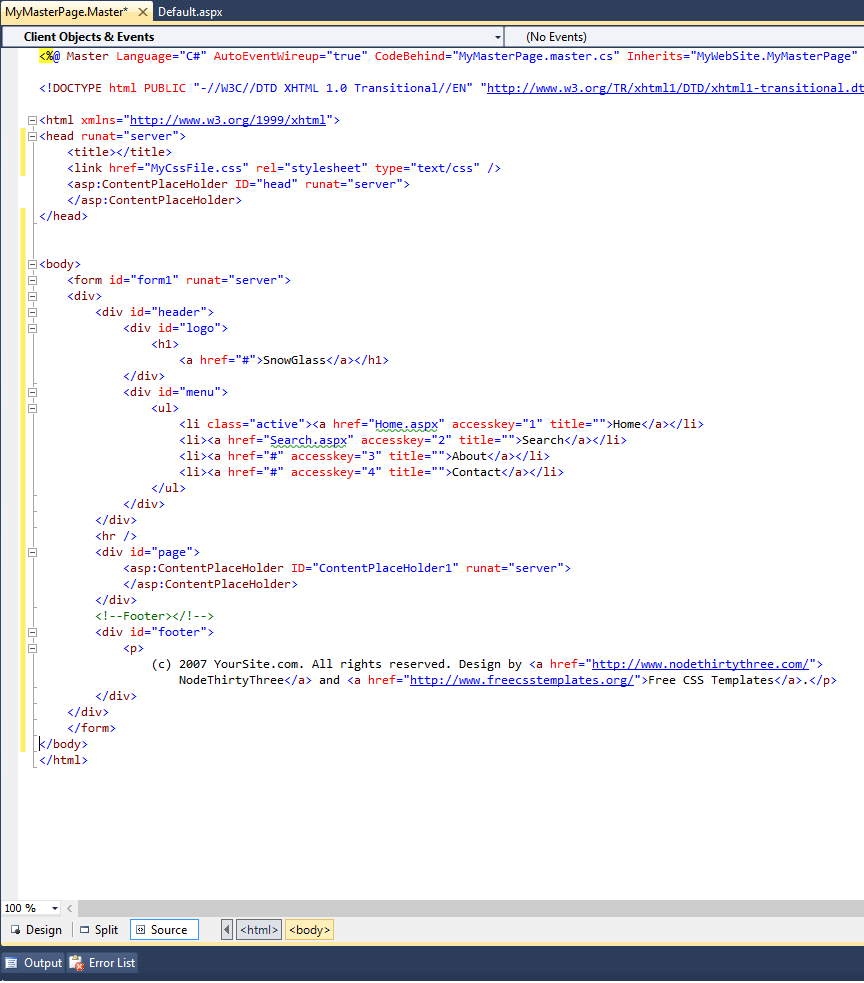
1. Now, add all the images given in Resources\images folder in this Images folder (follow figure 6 and 7)

*Figure 6: Add existing items to a folder*

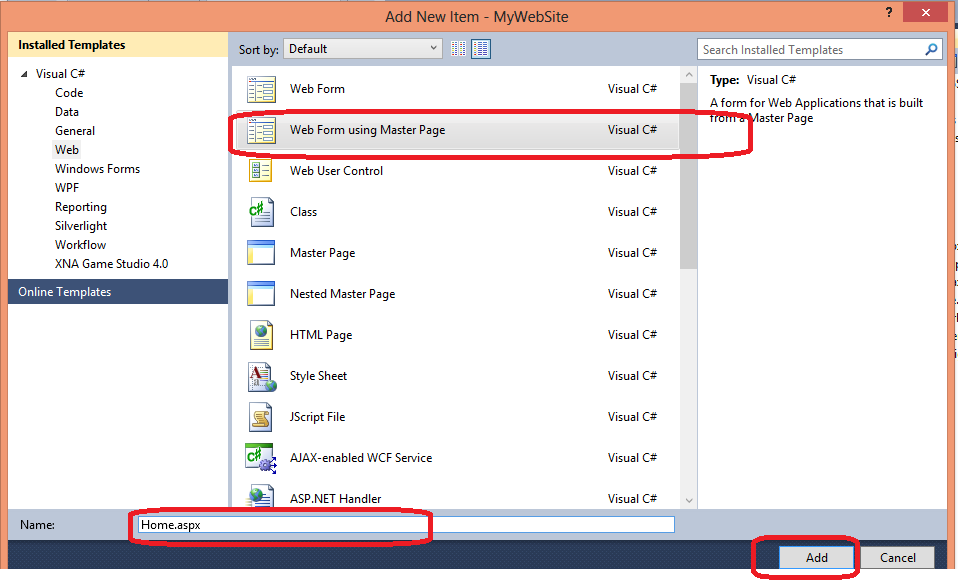
*Figure 7: Add images from Resources/images folder to Images folder of web application*

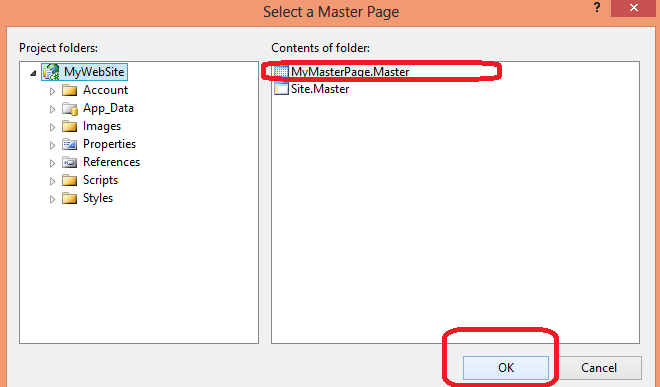
1. Now, open **MyMasterPage.Master** and drag drop **MyCssFile.css** file in <head> tag (follow figure 8)

*Figure 8: Add CSS file MyCssFile.css in MyMasterPage.Master*

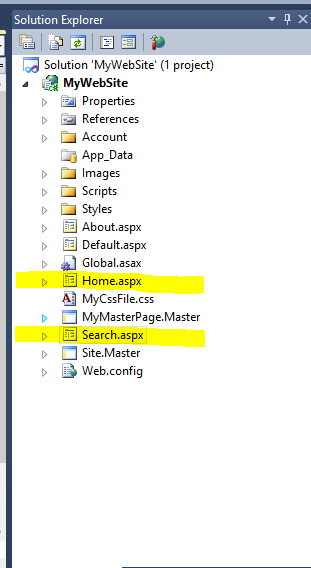
1. Open the **MasterPage\_Body.txt** file given in Resources folder and copy all the contents, replace everything inside the <body> tag of **MyMasterPage.Master** with copied content (follow figure 9)

*Figure 9: Modify MyMasterPage.Master*

1. Now, create two web forms name **Home.aspx** and **Search.aspx** using Add New Item, which will use our Master Page **MyMasterPage.Master**.Add the new web forms using **MyMasterPage.Master** (follow figure 10 and 11)

*Figure 10: Add a new Home.aspx page*

*Figure 11: Add MyMasterPage.Master to Home.aspx*

1. Confirm that the two web forms are added form **Solution Explorer** (follow figure 12)

*Figure 12: Verify Home.aspx and Search.aspx in solution explorer*

1. Execute your project and see the result in browser. Click on home and search and see the change in address bar (follow figure 13)

*Figure 13: Execute project to check home page and search page*

1. **Achievement Unlocked!** You have successfully created an **ASP.NET** Master Page name **MyMasterPage.Master** and used it in **Home.aspx** and **Search.aspx**. Save your work for the next exercise.

# Database Connectivity

This exercise will show how to connect the web site with the MSSQL database and how to access the data present in the database.

1. Open the **DataBaseQueries.sql** file in **Resources** folder, and execute it in MSSQL Server.
2. This will create a new database with name **DBConnectDemo,** an **items** table, and **SearchItem** stored procedure**.** We will use the data from this table and result from this procedure in our web site.
3. Use the same web project in previous exercise. Create a database connection string in **Web.config** file

**Format of Connection String**

*<connectionStrings>*

*<add*

*name="SQLDbConnection"*

*connectionString="Data Source=SQlServerName; Initial Catalog=YouDatabaseName; User Id=userid; Password= password"*

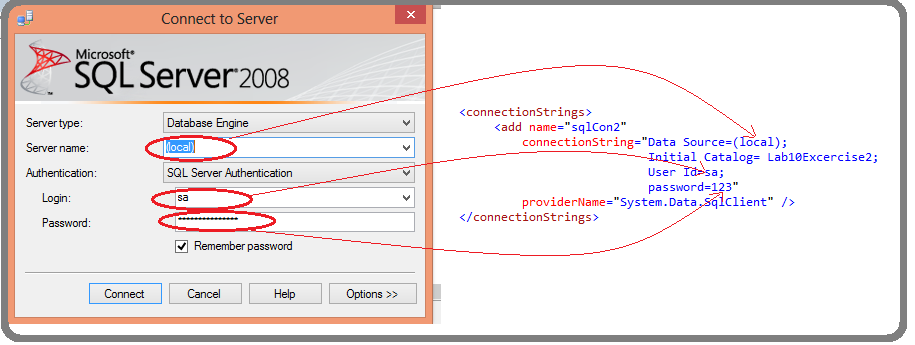
*providerName="System.Data.SqlClient" />*

*</connectionStrings>*

**More info about connection string**<http://msdn.microsoft.com/en-us/library/jj653752%28v=vs.110%29.aspx>

* **Data Source**: server name i.e. **cactus** or **local**
* **Initial Catalog**: database name which has your table, stored procedure and views i.e. **DBConnectDemo**
* **User Id**: login name for database
* **Password**: password for database

An easy way to get these values are from MSSQL Server Connect to Server window (follow figure 14)

**Our Sample Connection Strings**

* Connection string with window authentication for local host

*<connectionStrings>*

*<add*

*name="**sqlCon1**"*

*connectionString="Data Source=(local); Initial Catalog=Lab10Excercise2; Integrated Security=True"*

*providerName="System.Data.SqlClient" />*

*</connectionStrings>*

* Connection string with SQL server authentication for server name MYSQLSERVER

*<connectionStrings>*

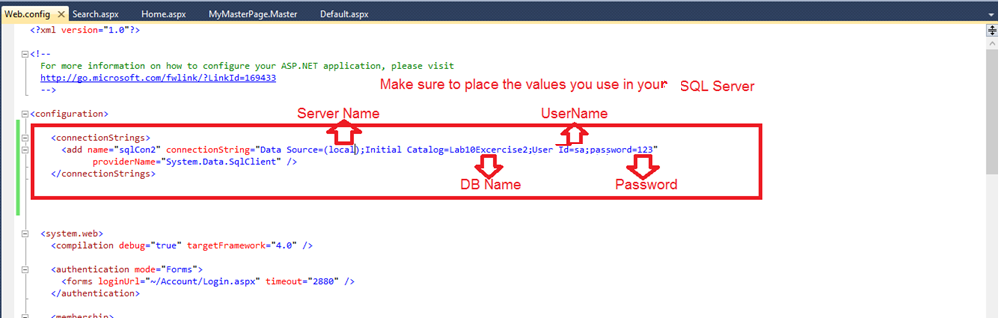
*<add*

*name="sqlCon1"*

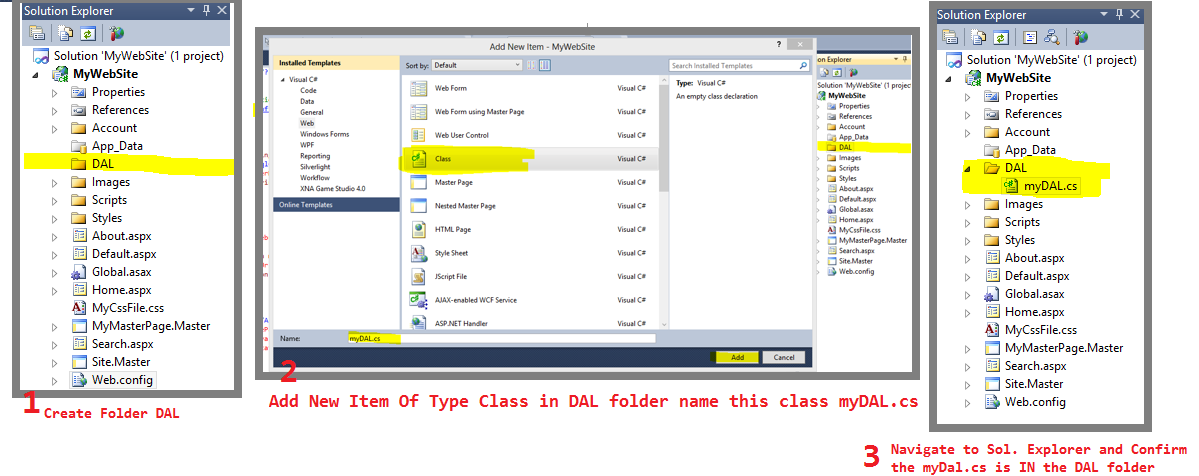
*connectionString="Data Source=(local); Initial Catalog=Lab10Exercise2; User ID=sa; password=123"*

*providerName="System.Data.SqlClient" />*

*</connectionStrings>*

1. Add your connection string in **Web.config** file (follow figure 15)

*Figure 15: Add connection string to Web.config file*

1. Creating DAL (Data Access Layer), to get data from MSSQL. Create new folder in your project named DAL, in this folder Add New Item of type Class and name it **myDAL.cs** (follow figure 16)

*Figure 16: Add a new myDAL.cs class in a newly created DAL folder*

1. Open **myDal.cs** file and add the reference to database connection string, namespaces for SQL and Datasets (follow figure 17)

**Reference to Connection String**

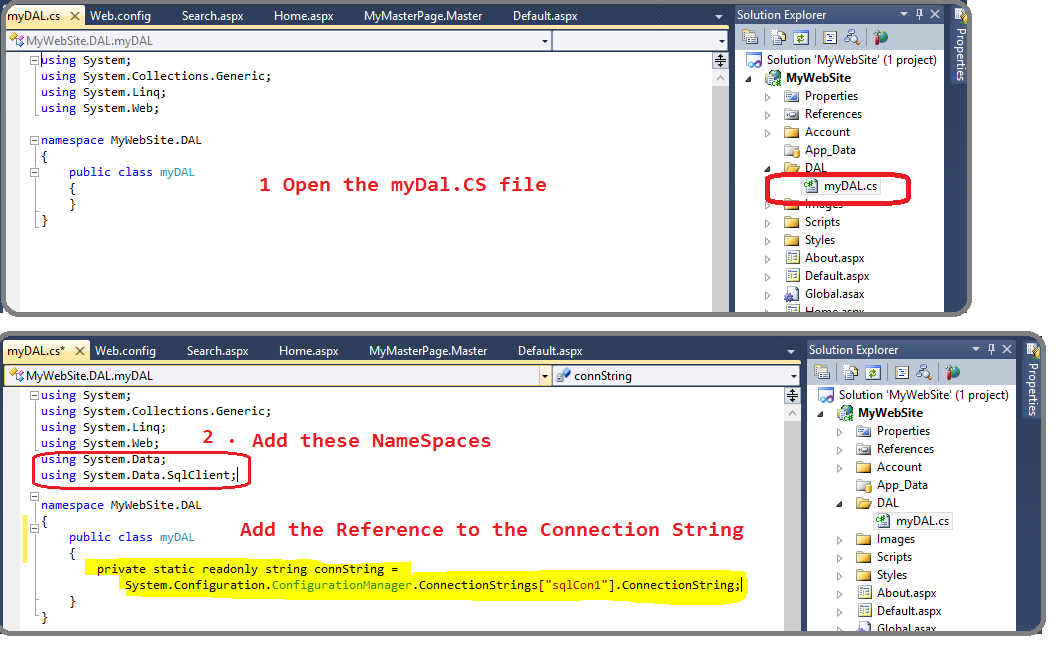
*private static readonly string connString =*

*System.Configuration.ConfigurationManager.ConnectionStrings["sqlCon1"].ConnectionString;*

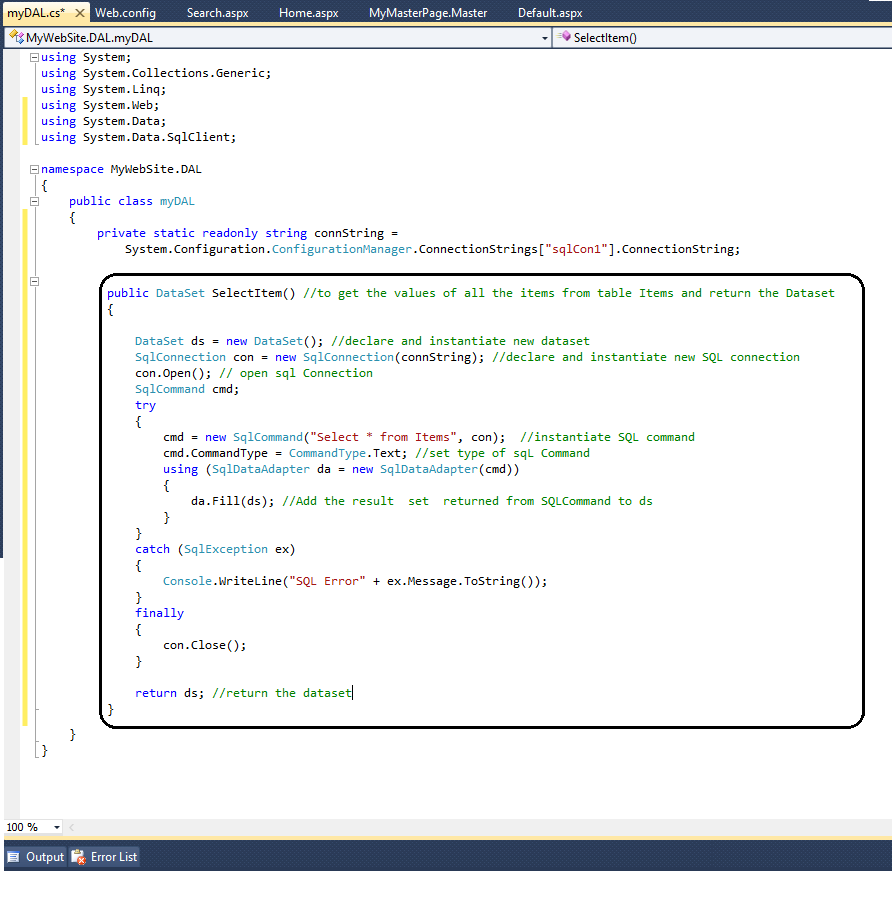
**Namespaces for SQL and Datsets**

*using System.Data;*

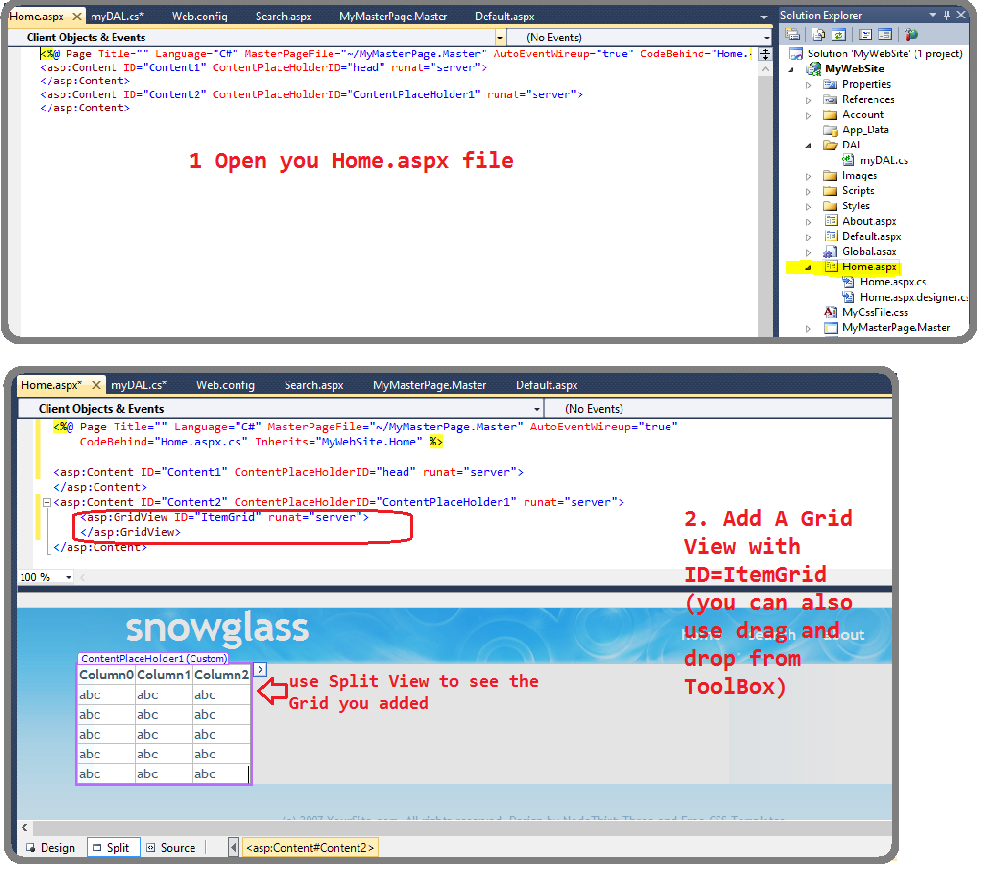
*using System.Data.SqlClient;*



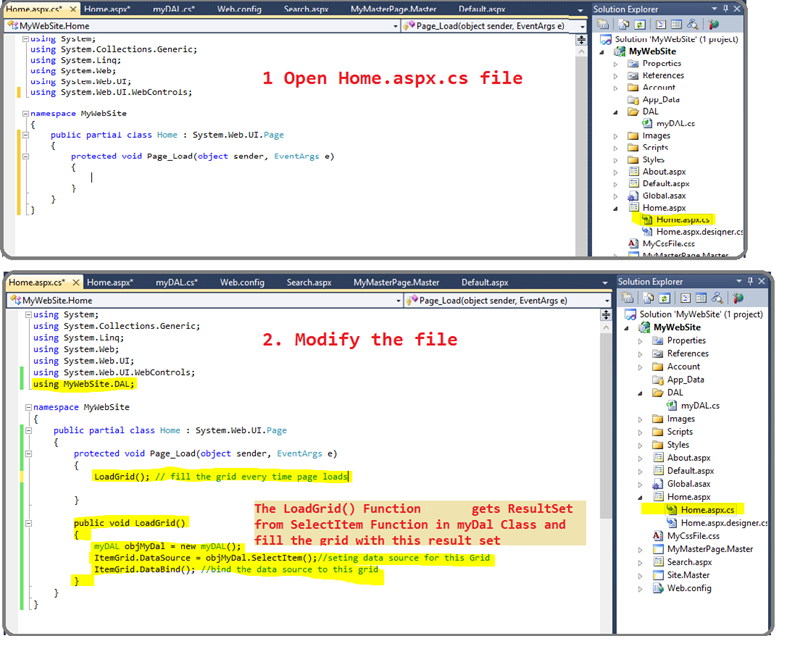
*Figure 17: Add namespaces and reference to connection string in myDAL.cs file*

1. Create a function in **myDal.cs** class that will perform a simple ***select \* from Item***query on Database and get the result set. Copy and paste the function ***SelectItem()*** from **Function\_SelectItem.txt** file (follow figure 18)

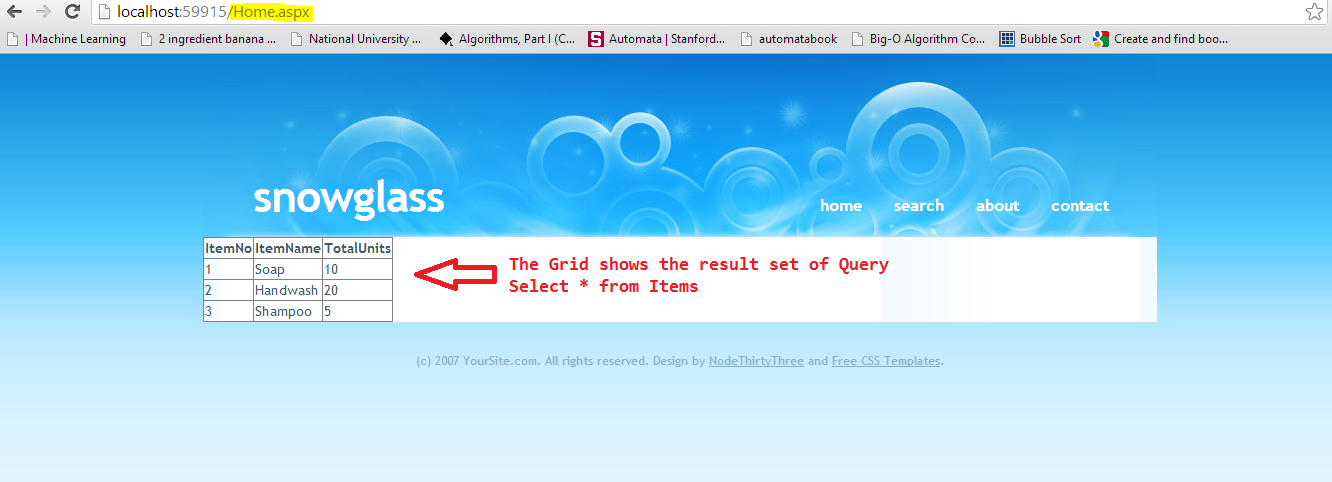
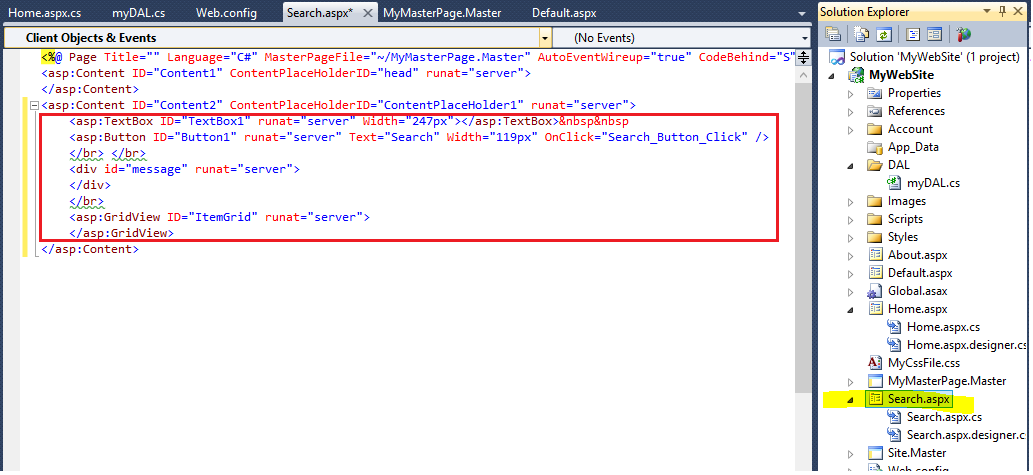
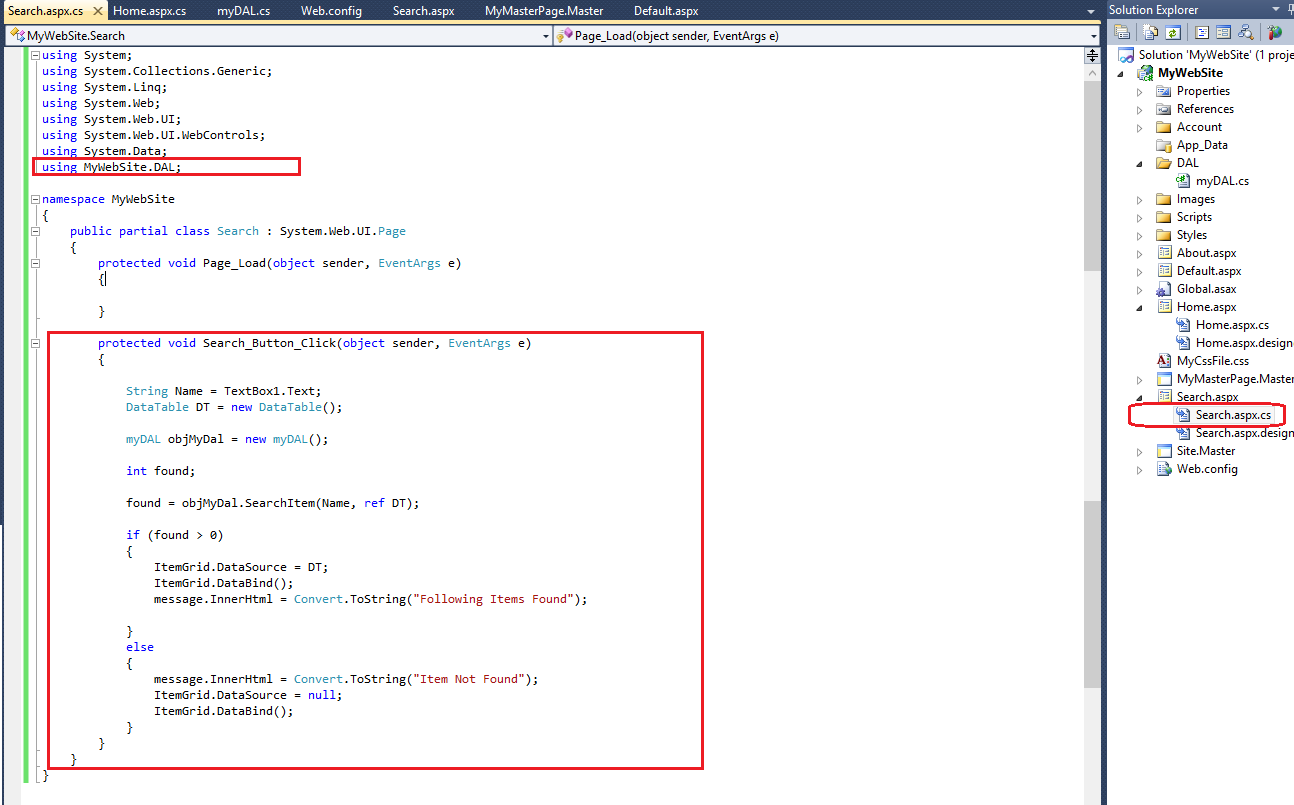
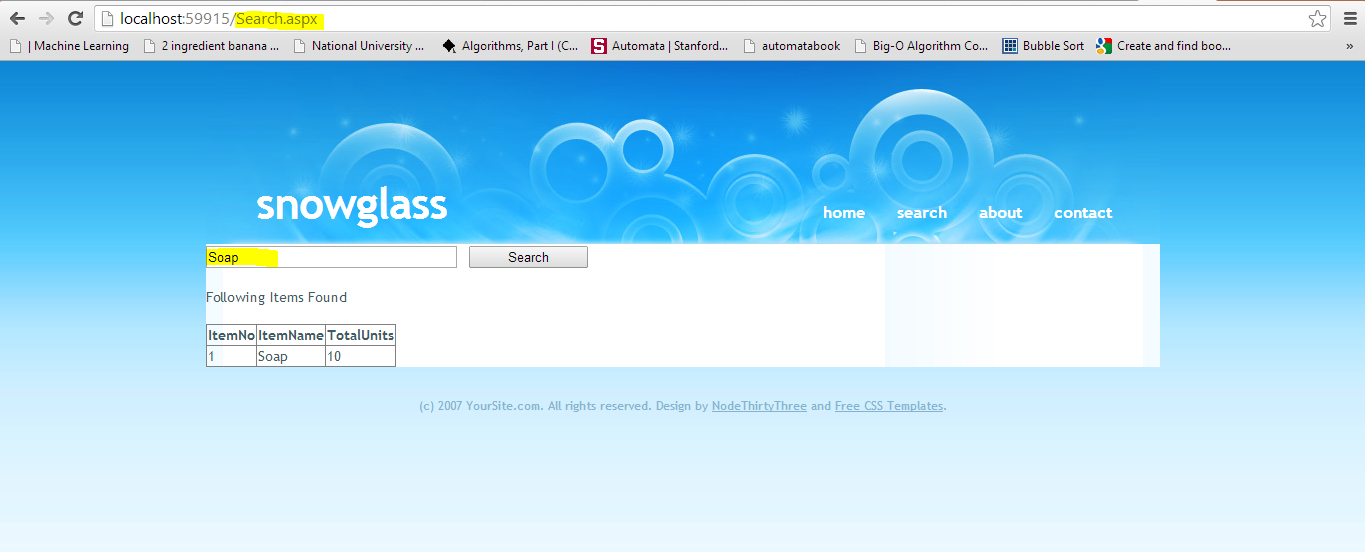
*Figure 18: Modify myDAL.cs file with SelectItem() function to get all items*

1. Use this **SelectItem()** function for displaying the result on home page **Home.aspx** by adding a **GridView** with **ID=“ItemGrid”**. You can see that the grid already looks like a table, all you need to do is fill the values in it from your query. The contents of grid are changed or updated from server side which is the **aspx.cs** file. Open your **Home.aspx.cs** file and modify it using the changes given in **Home\_aspx\_CS.txt** file (follow figure 19 and 20)

*Figure 19: Modify Home.aspx page and add a GridView*



*Figure 20: Modify Home.aspx.cs file to load data in GridView*

1. Execute your project, if everything goes right, the home page should look like the page given below. You have successfully used the result of a simple query on your website. **Achievement Unlocked!** (follow figure 21)
2. Now, we will call MSSQL stored procedure from ASP.NET. We will learn how to pass input parameters, how to get output parameters and result sets from stored procedures. We have already created a web form **Search.aspx** in our web application and **SearchItem** stored procedure in our **DBConnectDemo** database. Now, we will add the search functionality on our page, by getting the **ItemName** from user and querying the database for that item using **SearchItem** MSSQL stored procedure. The procedure will return the dataset and output found=1, if any item with that name exists, and found=0 if no item is found.
   1. Open **Search.aspx** page and modify the code, adding **TextBox** and **Button** and a **GridView** to show the result of search. Code is given in file **Search\_Aspx.txt**
   2. Open **myDal.cs** file and add **SearchItem()** function in it from the **Function\_Searchitems.txt** file (follow figure 23)
   3. Use this new function **SearchItem()** in server side **Search.aspx.cs**. Open **Search.aspx.cs** file and modify with code given in **Search\_Aspx\_CS.txt** file (follow figure 24)
   4. Execute your project. Type “Soap” in TextBox and click Search. The following results should appear. **Achievement Unlocked!** (follow figure 25)
   5. Now, type some random string in TextBox and click Search. The following results should appear (follow figure 26)