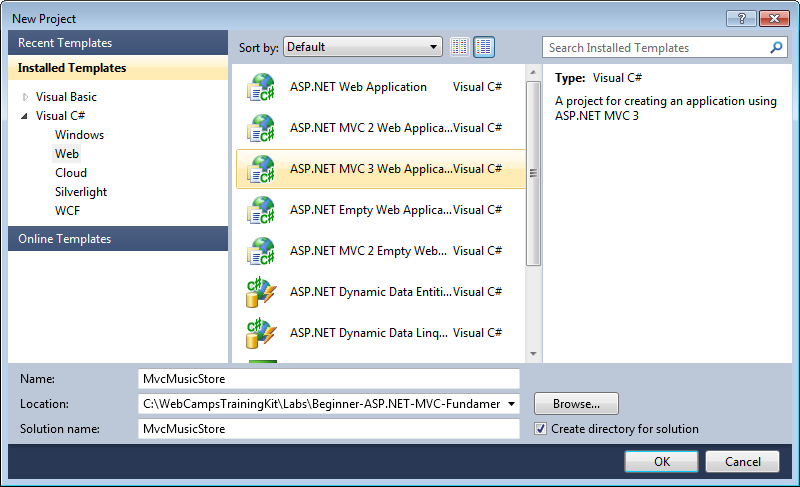
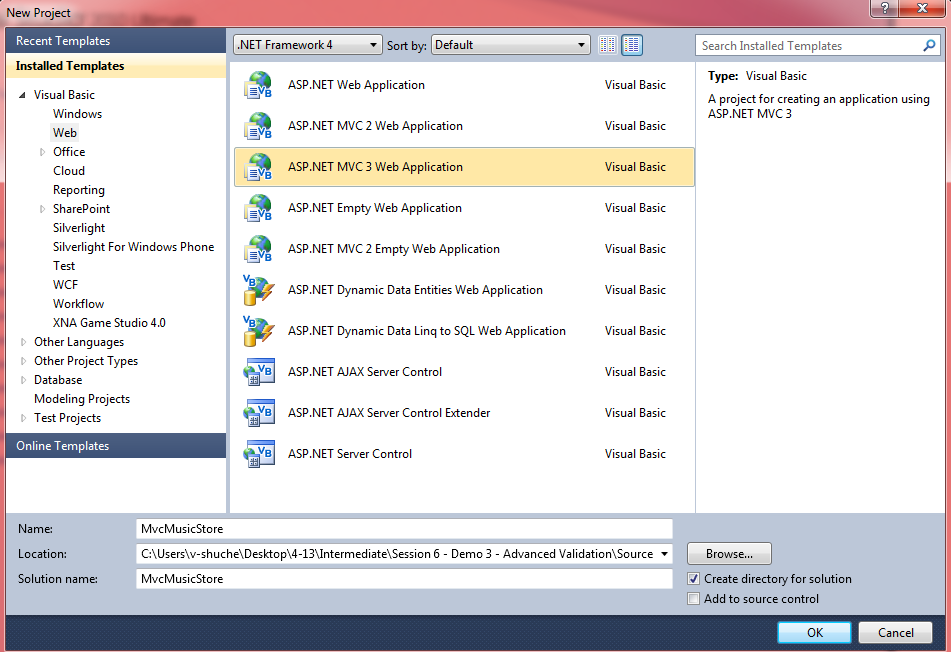
Lưu ý để thực hiện các bài thực hành cần cài đặt

* Visual Studio 2010
* ASP.NET MVC 3

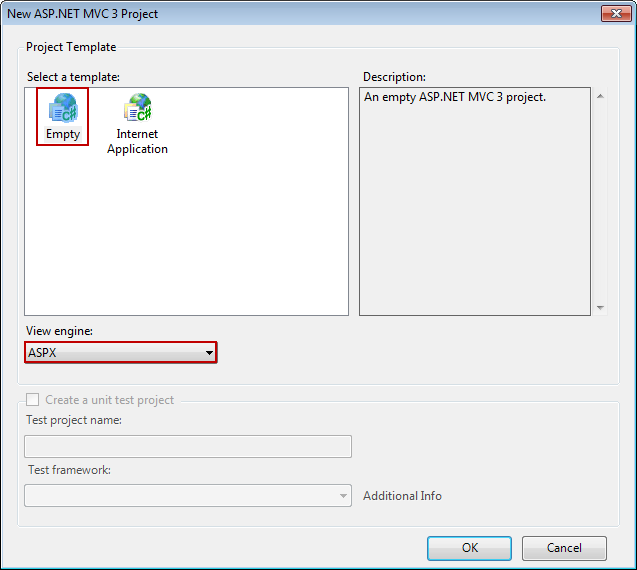
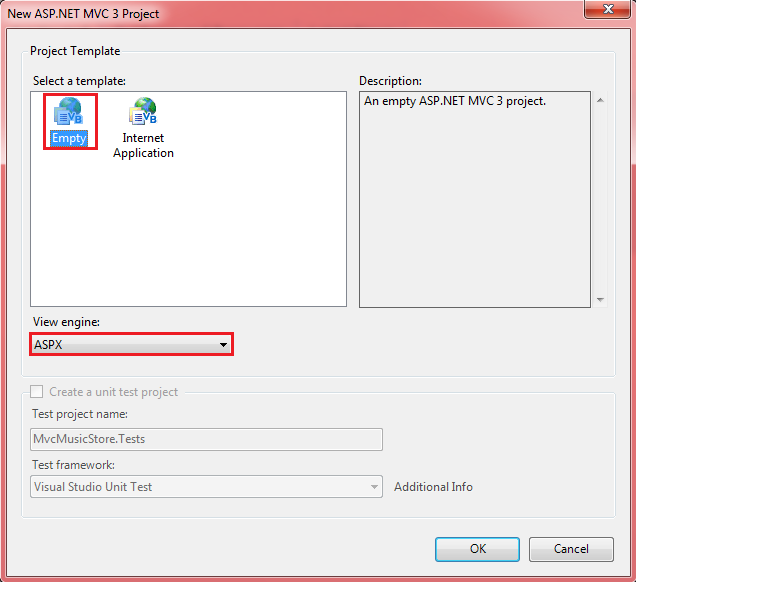
BÀI 1:TẠO MỘT ỨNG DỤNG WEB TRÊN NỀN TẢNG ASP.NET MVC

Trong bài tập này, bạn sẽ tìm hiểu làm thế nào để tạo ra một ứng dụng ASP.NET MVC trong Visual Studio 2010 cũng như cách tổ chức các thư mục của nó. Ngoài ra, bạn sẽ tìm hiểu làm thế nào để thêm một Controller mới và nó hiển thị một chuỗi đơn giản ở màn hình ứng dụng.

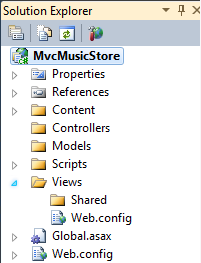
Task 1 – Tạo ASP.NET MVC Dự án Ứng dụngWeb

1. Trong nhiệm vụ này, bạn sẽ tạo ra một ứng dụng ASP.NET MVC  sử dụng template MVC Visual Studio.
   1. **Start** | **All Programs** | **Microsoft Visual Studio 2010 Express** | **Microsoft Visual Web Developer 2010 Express**.
   2. **File** -> **New Project**.
   3. **New Project** ->**ASP.NET MVC 3 Web Application** **Visual [C#|Basic],** **Web** .
   4. Thay đổi tên **Name** thành **MvcMusicStore**.
   5. **C:\WebCampsTrainingKit\Labs\Beginner-ASP.NET-MVC-Fundamentals MVC3\Source\Ex01-CreatingMusicStoreProject\Begin**. Chọn **OK**.
      1. 
      2. Figure 11
      3. Create New Project Dialog Box - C#
      4. 
      5. Figure 12
      6. Create New Project Dialog Box - VB
   6. **New ASP.NET MVC 3 Project** ->**Empty**

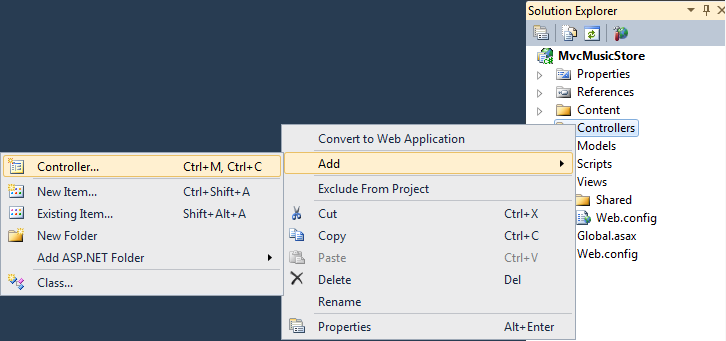
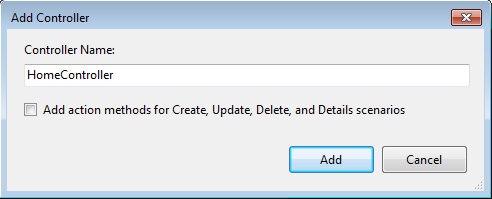
**View engine** : **ASPX**. Chọn **OK**.

* + 1. 
    2. Figure 13
    3. New ASP.NET MVC 3 Project Dialog Box - C#
    4. 
    5. Figure 14
    6. New ASP.NET MVC 3 Project Dialog Box - VB

Task 2 – Xem Qua cấu trúc của Solution

* 1. ASP.NET MVC giúp bạn tạo ra các ứng dụng Web được cấu trúc để hỗ trợ các mô hình MVC. Một ứng dụng Web MVC mới được cấu hình để có các thư mục cần thiết, và các tập tin cấu hình.
     1. 
     2. Figure 15
     3. ASP.NET MVC Folder structure in Solution Explorer
     4. **Content**. Thư muc này khuyến khích là chứa nhữa các file hình ảnh, css….
     5. **Controllers.** Nó có tác dụng điều hướng những request và nắm giữ những model và truyền tải dữ liệu tới View.
     6. **Models**. Thư mục chứa những lớp cho việc hiện thực hoá dữ liệu
     7. **Scripts**. Thư mục này khuyến khích chứa các thư viện javascipt
     8. **Views**. Thư mục chứa các trang web trình diễn cho người xem (aspx, .ascx, .cshtml , .master files)
     9. **Views\Shared.** Mặc định, cũng có mục thư mục tên Shared trong thư mục Views. Thư mục Shared này có thể được sử dụng để chứa các trang master page của ứng dụng web.
     10. **Global.asax** là file để thiết lập đường dẫn URL, và **Web.config** là file để thiết lập ứng dụng.

Task 3 –Thêm một Controller

* 1. Chuột phải vào thư mục **Controllers** trong Solution Explorer, chọn **Add** một **Controller** .
     1. 
     2. Figure 16
     3. Add a Controller Command
  2. Cửa sổ **Add Controller** xuất hiện .Bạn đặt tên là **HomeController** và nhấn **Add**.
     1. 
     2. Figure 17
     3. Add Controller Dialog
  3. File **HomeController.cs** được tạo trong thư mục **Controllers** .Để HomeController trả về một chuỗi, thay thế phương thức Index() vơi dòng code sau
     1. C#

**public string Index()**

**{**

**return "Hello from Home";**

**}**

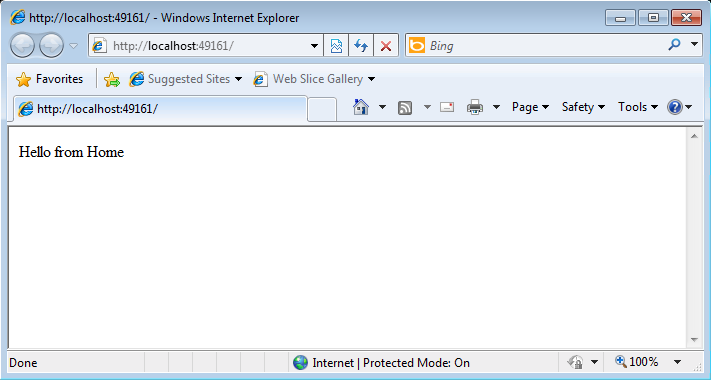
* + 1. Visual Basic

**Public Function Index() As String**

**Return "Hello from Home"**

**End Function**

Task 4 – Chạy ứng dụng

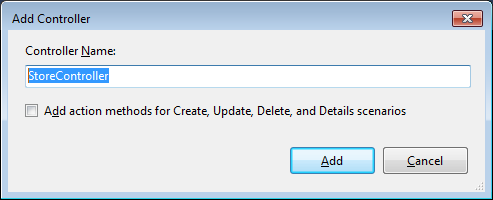
* + 1. Nhấn F5 để chạy các ứng dụng. Dự án được biên dịch và các máy chủ WebASP.NET được xây dựng trên Visual Web Developer bắt đầu chạy. Visual Web Developer sẽ tự động mở một trình duyệt web.  
    2. Figure 18
    3. Application running in a web browser
  1. Đóng trình duyệt.

Bài 2: Tạo một Controller

Trong bài tập này, bạn sẽ tìm hiểu làm thế nào để tạo ra Controller để thực hiện chức năng đơn giản của Music Store. Đó là bộ điều khiển sẽ xác định mọi phương pháp hành động để xử lý từng yêu cầu cụ thể:Liệt kê tất cả các thể loại nhạc trong Music store.

* + Liệt kê tất cả những ablums trong một loại nhạc cụ thể.
  + Chi tiết một albums.

Task 1 – Thêm một StoreController .

* 1. **Start** | **All Programs** | **Microsoft Visual Studio 2010 Express** | **Microsoft Visual Web Developer 2010 Express**.
  2. **Thêm một**  **Controller** command. Thay đổi **Controller** **Name** thành **StoreController**, và chọn **Add**.
     1. 
     2. Figure 19
     3. Add Controller Dialog

Task 2 – Điều chỉnh StoreController’s Actions

* 1. **Thay thế chuỗi trả về trong hàm Index() :**
     1. C#

**public string Index()**

**{**

**return "Hello from Store.Index()";**

**}**

* + 1. Visual Basic

**Public Function Index() As String**

**Return "Hello from Store.Index()"**

**End Function**

* 1. Thêm một phương thức Browse() làm cho trả về một chuỗi.
     1. C#

**//**

**// GET: /Store/Browse**

**public string Browse()**

**{**

**return "Hello from Store.Browse()";**

**}**

**//**

**// GET: /Store/Details**

**public string Details()**

**{**

**return "Hello from Store.Details()";**

**}**

* + 1. Visual Basic

**'**

**'GET: /Store/Browse**

**Public Function Browse() As String**

**Return "Hello from Store.Browse()"**

**End Function**

**'**

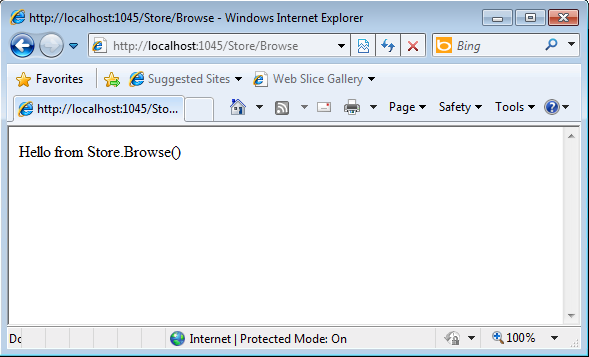
**'GET: /Store/Details**

**Public Function Details() As String**

**Return "Hello from Store.Details()"**

**End Function**

Task 3 – Chạy ứng dụng

* 1. Chạy một ứng dụng trên một trình duyệt.
  2. Nhấn F5 để chạy ứng dụng.
  3. Thay đổi đường dẫn như bên dưới bạn sẽ thấy kết quả hiện ra trong trình duyệt
     1. **/Store**. → **“Hello from Store.Index()”**.
     2. **/Store/Browse**. → **“Hello from Store.Browse()”**.
     3. **/Store/Details**. → **“Hello from Store.Details()”**.
        1. 
        2. Figure 20
        3. Browsing /Store/Browse
  4. Đóng trình duyệt

BÀI 3:TRUYỀN MỘT THAM SỐ TỚI CONTROLLER

Trong bài này chúng ta sẽ học cách sử dụng cách truyền tham số từ đường dẫn vào Controller

Task 1 – Thêm một tham số vào đường dẫn và truyền vào StoreController

C#

**//**

**// GET: /Store/Browse?genre=Disco**

**public string Browse(string genre)**

**{**

**string message = HttpUtility.HtmlEncode("Store.Browse, Genre = " + genre);**

**return message;**

**}**

* + 1. Visual Basic

**'**

**'GET: /Store/Browse?genre=Disco**

**Public Function Browse(ByVal genre As String) As String**

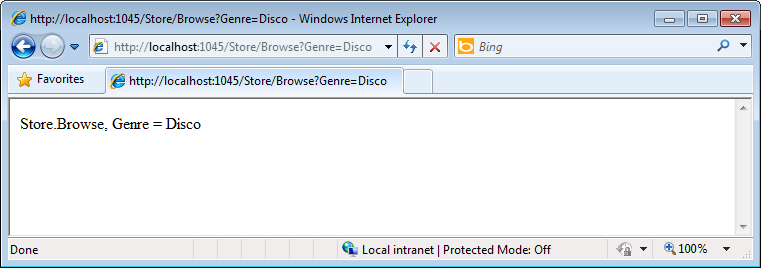
**Dim message As String = HttpUtility.HtmlEncode("Store.Browse, Genre = " & genre)**

**Return message**

**End Function**

* + 1. **Chú ý:** Bạn sử dụng HttpUtility.HtmlEncode để chống người dùng chèn script độc hại vào chương trình. Ví dụ như:
    2. **/Store/Browse?Genre=<script>window.location=’http://hackersite.com’</script>**.

Task 2 – Chạy ứng dụng

* 1. Nhấn F5 để chạy ứng dụng.
  2. Thay đổi URL thành **/Store/Browse?Genre=Disco** để truyền tham số vào
     1. 
     2. Figure 21
     3. Browsing /Store/Browse?Genre=Disco
  3. Đóng trình duyệt

Task 3 –Thêm một Id vào một Controller

* 1. Trong URL **Store/Detail/5**, **Id** sẽ được truyền vào giá trị là **5**.
  2. Trong phương thức Details có một tham số Id để lấy từ đường dẫn URL
     1. C#

**//**

**// GET: /Store/Details/5**

**public string Details(int id)**

**{**

**string message = "Store.Details, ID = " + id;**

**return message;**

**}**

* + 1. Visual Basic

**'**

**'GET: /Store/Details/5**

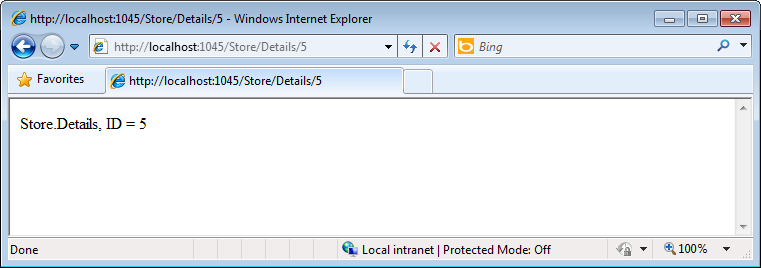
**Public Function Details(ByVal id As Integer) As String**

**Dim message As String = "Store.Details, ID = " & id**

**Return message**

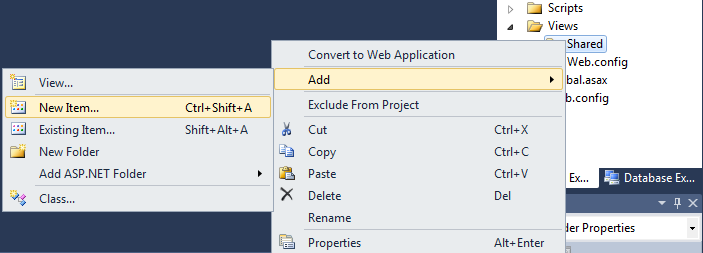
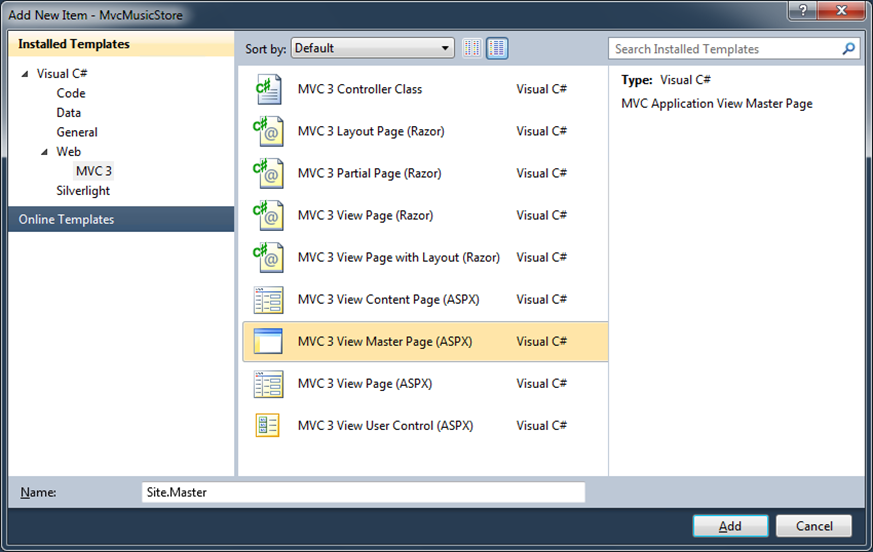
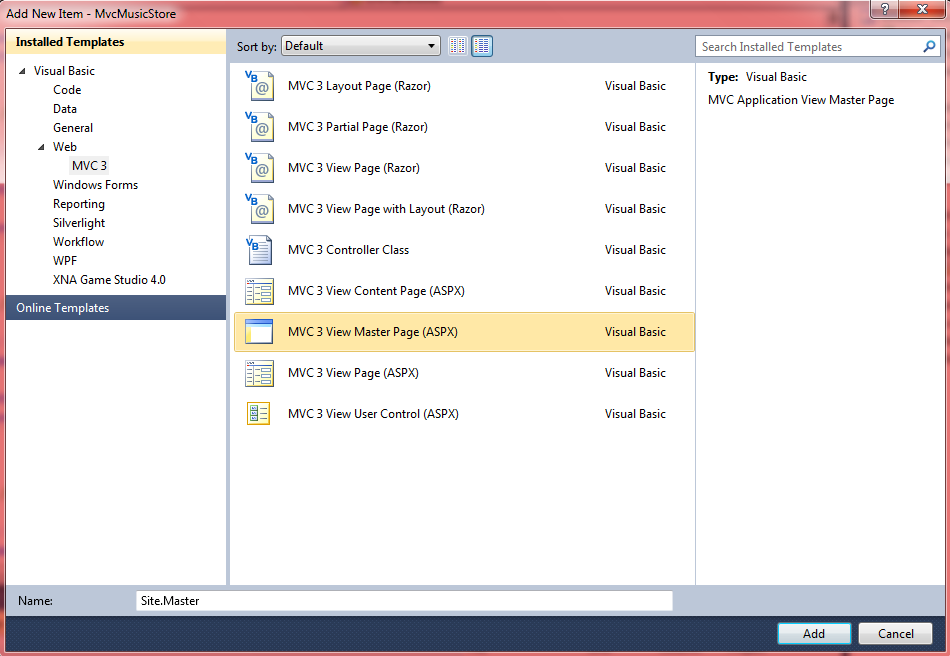
**End Function**

Task 4 – Chạy ứng dụng

* 1. Nhấn F5 để chạy ứng dụng.
  2. Thay đổi URL thành **/Store/Details/5** để truyền giá trị tham số id.
     1. 
     2. Figure 22
     3. Browsing /Store/Details/5

Bài 4:Tạo một View

Task 1 – Thêm vào 1 MasterPage

* 1. Mater Pages như là một layout của site
  2. **Start** | **All Programs** | **Microsoft Visual Studio 2010 Express** | **Microsoft Visual Web Developer 2010 Express**.
  3. **Trong thư mục Shared bạn Add thêm New Item chọn Master page**
     1. 
     2. Figure 23
     3. Adding a new item
     4. ****
     5. Figure 24
     6. Add New Item Dialog – C#
     7. 
     8. Figure 25
     9. Add New Item Dialog - VB
     10. HTML(C#)

<%@ Master Language="C#" Inherits="System.Web.Mvc.ViewMasterPage" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >

<head runat="server">

<title><asp:ContentPlaceHolder ID="TitleContent" runat="server" /></title>

</head>

<body>

<div>

<asp:ContentPlaceHolder ID="MainContent" runat="server">

</asp:ContentPlaceHolder>

</div>

</body>

</html>

* + 1. HTML(Visual Basic)

<%@ Master Language="VB" Inherits="System.Web.Mvc.ViewMasterPage" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >

<head runat="server">

<title><asp:ContentPlaceHolder ID="TitleContent" runat="server" /></title>

</head>

<body>

<div>

<asp:ContentPlaceHolder ID="MainContent" runat="server">

</asp:ContentPlaceHolder>

</div>

</body>

</html>

* 1. Thêm vào một header trong master page
     1. HTML(C#)

<%@ Master Language="C#" Inherits="System.Web.Mvc.ViewMasterPage" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >

<head runat="server">

<title><asp:ContentPlaceHolder ID="TitleContent" runat="server" /></title>

</head>

<body>

<div>

**<div id="header">**

**<h1>ASP.NET MVC MUSIC STORE</h1>**

**<ul id="navlist">**

**<li class="first"><a href="/" id="current">Home</a></li>**

**<li><a href="/Store/">Store</a></li>**

**</ul>**

**</div>**

<asp:ContentPlaceHolder ID="MainContent" runat="server">

</asp:ContentPlaceHolder>

</div>

</body>

</html>

* + 1. HTML(Visual Basic)

<%@ Master Language="VB" Inherits="System.Web.Mvc.ViewMasterPage" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >

<head runat="server">

<title><asp:ContentPlaceHolder ID="TitleContent" runat="server" /></title>

</head>

<body>

<div>

**<div id="header">**

**<h1>ASP.NET MVC MUSIC STORE</h1>**

**<ul id="navlist">**

**<li class="first"><a href="/" id="current">Home</a></li>**

**<li><a href="/Store/">Store</a></li>**

**</ul>**

**</div>**

<asp:ContentPlaceHolder ID="MainContent" runat="server">

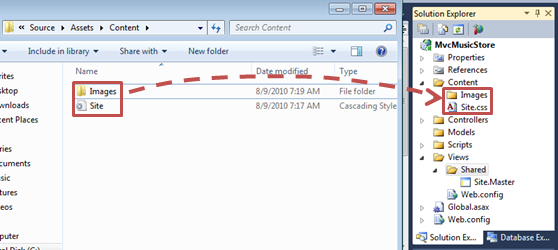
</asp:ContentPlaceHolder>

</div>

</body>

</html>

Task 2 – Thêm vào một Site CSS

* + 1. 
    2. Figure 26
    3. Dragging style contents

Thêm vào một tag link để liên kết tới file css.

* + 1. HTML

<head runat="server">

**<link href="/Content/Site.css" rel="Stylesheet" type="text/css" />**

<title><asp:ContentPlaceHolder ID="TitleContent" runat="server" /></title>

</head>

Task 3 – Thêm một View Template

* 1. Để sử dụng một View Template, trong phương thức Index() cần trả về một View();
     1. C#

public class HomeController : Controller

{

//

// GET: /Home/

**public ActionResult Index()**

**{**

**return View();**

**}**

}

* + 1. Visual Basic

Public Class HomeController

Inherits Controller

'

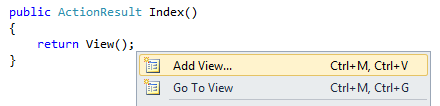
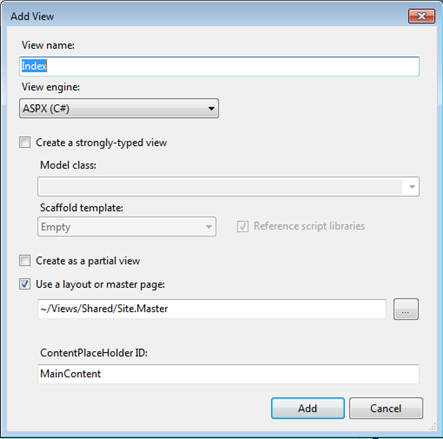
'GET: /Home/

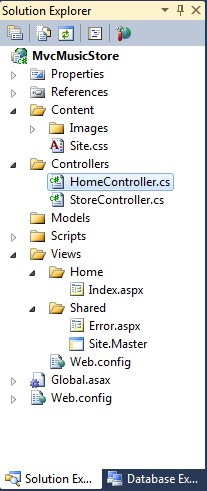
**Public Function Index() As ActionResult**

**Return View()**

**End Function**

End Class

1. Bây giờ, bạn cần phải thêm một trang View thích hợp. Để làm điều này, kích phải vào bên trong phương thức hành động Index và chọn Add View. Điều này sẽ mở hộp thoại Add Xem.
   * 1. 
     2. Figure 28
2. Một hộp thoại Add View xuất hiện. Cho phép tạo một trang View
   * 1. 
3. Visual Studio tạo một trang có tên ***Index.aspx* bên trong thư mục** Views\Home



1. Trang View này được tạo dựa trên **Site.master** và chứa 2 thẻ **<asp:content>**. Thay đổi tiêu đề thành **Home**, và thay đổi nội dung chính thành **This is the Home Page** như code bên dưới
   * 1. HTML

<asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">

Home

<h2>This is the Home Page</h2>

</asp:Content>

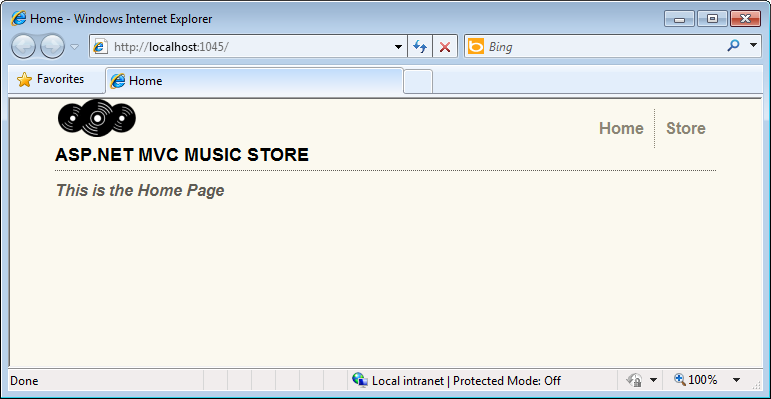
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

Home

<h2>This is the Home Page</h2>

</asp:Content>

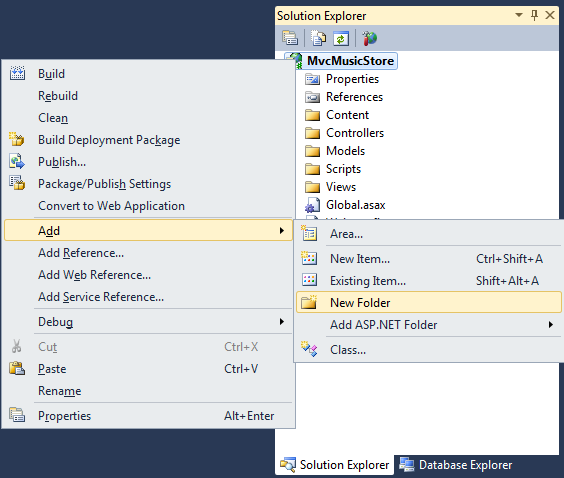
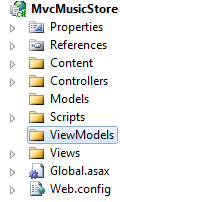
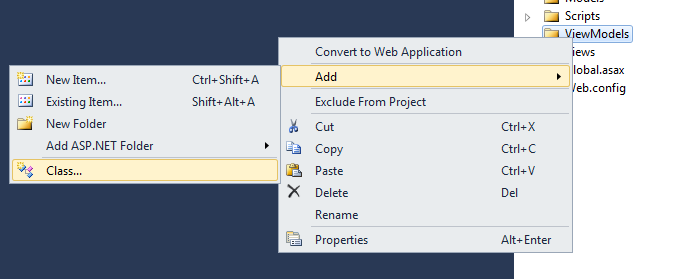
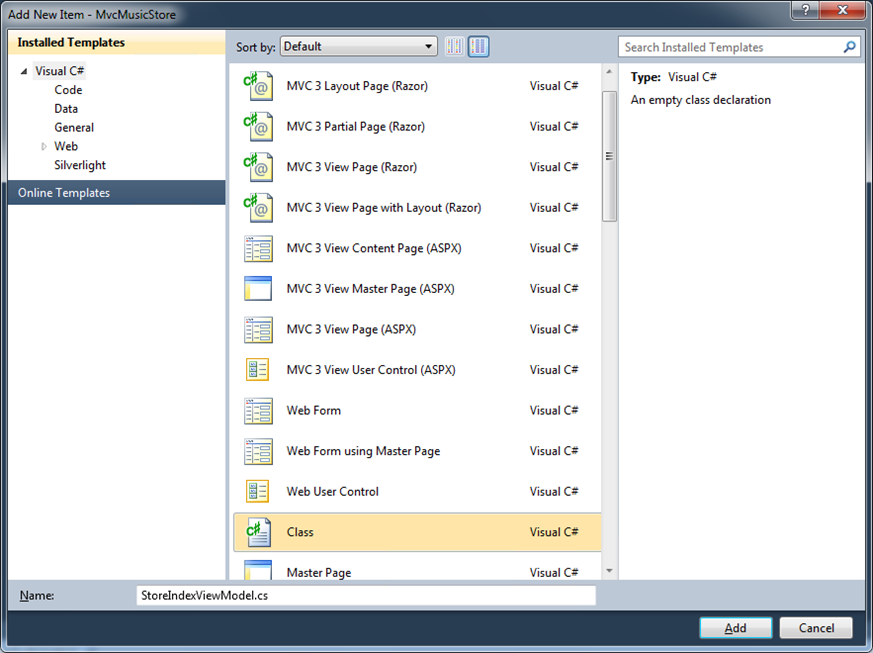
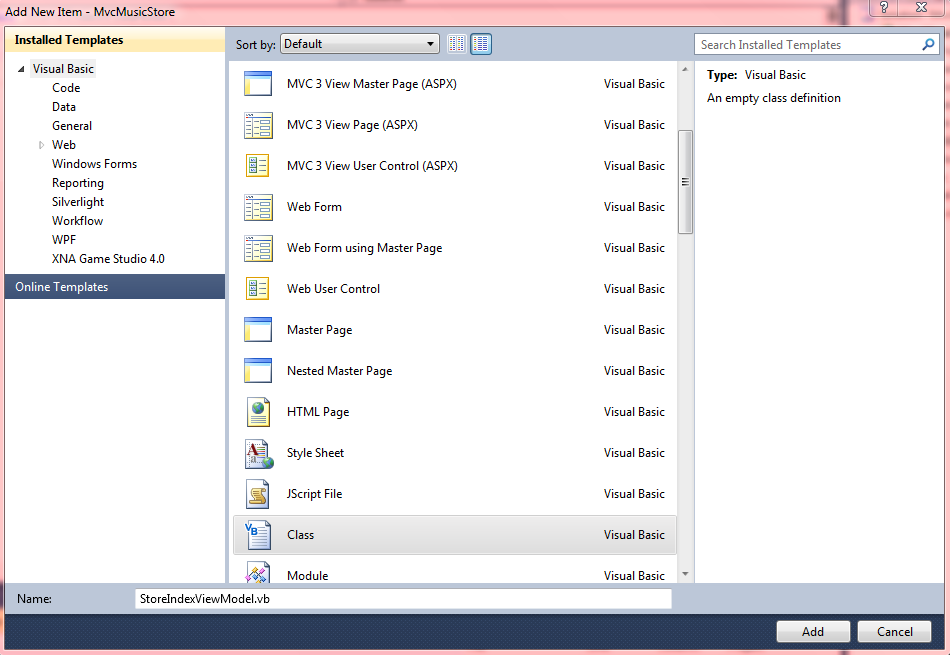
1. Ấn F5 để chạy ứng dụng.



* + - 1. Figure 33
      2. Home Index View using the MasterPage and style defined

Bài 5:Tạo một View Model

Task 1 – Tạo một lớp Model

* 1. **Start** | **All Programs** | **Microsoft Visual Studio 2010 Express** | **Microsoft Visual Web Developer 2010 Express**.
  2. Tạo một thư mục **ViewModels** .
     1. 
     2. Figure 34
     3. Adding a new folder
     4. 
     5. Figure 35
     6. ViewModels folder in Solution Explorer
  3. Tạo một lớp ViewModels
     1. 
     2. Figure 36
     3. Adding a new Class
  4. Đăt tên là **StoreIndexViewModel** .
     1. 
     2. Figure 37
     3. Creating StoreIndexViewModel class – C#
     4. 
     5. Figure 38
     6. Creating StoreIndexViewModel class - VB

Task 2 – Thêm những thuộc tính vào lớp ViewModel

Ta sẽ thêm 2 thuộc tính

* Một là NumberOfGenres
* Hai là list Genres
  + 1. C#

public class StoreIndexViewModel

{

**public int NumberOfGenres { get; set; }**

**public List<string> Genres { get; set; }**

}

* + 1. Visual Basic

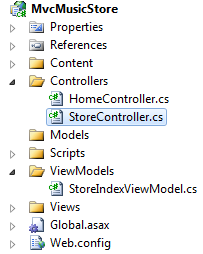
**Public Class StoreIndexViewModel**

**Public Property NumberOfGenres As Integer**

**Public Property Genres As List(Of String)**

**End Class**

Task 3 – Cập nhật lại StoreController để sử dụng StoreIndexViewModel

* 1. Mở lớp **StoreController** .
     1. 
     2. Figure 39
     3. Opening StoreController class
  2. Để sử dụng StoreIndexViewModel ta thêm vào namespace.
     1. C#

**using MvcMusicStore.ViewModels;**

* + 1. Visual Basic

**VB no need to add namespace**

* 1. Thay thế phương thức index() lại như sau:
     1. C#

**public ActionResult Index()**

**{**

**// Create a list of genres**

**var genres = new List<string> {"Rock", "Jazz", "Country", "Pop", "Disco"};**

**// Create our view model**

**var viewModel = new StoreIndexViewModel {**

**NumberOfGenres = genres.Count(),**

**Genres = genres**

**};**

**return View(viewModel);**

**}**

* + 1. Visual Basic

**Public Function Index() As ActionResult**

**'Create list of genres**

**Dim genres = New List(Of String) From {"Rock", "Jazz", "Country", "Pop", "Disco"}**

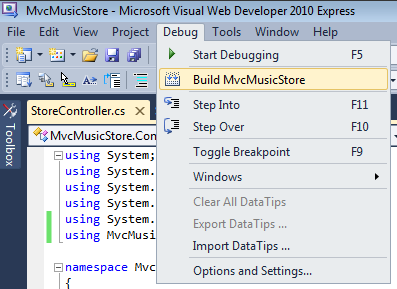
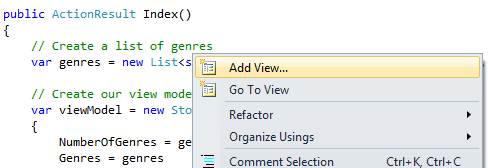
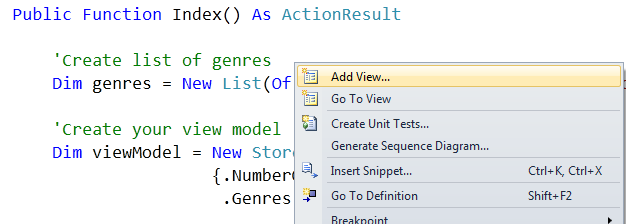
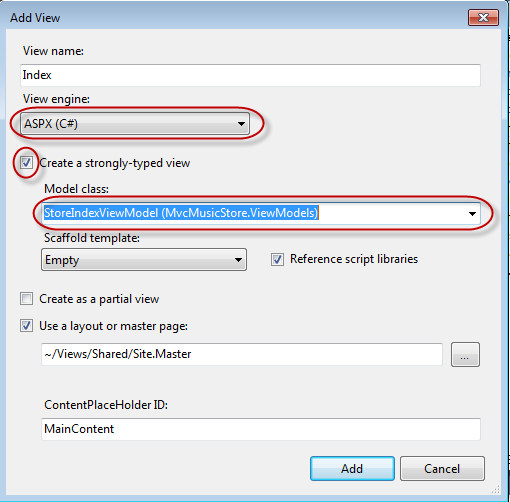
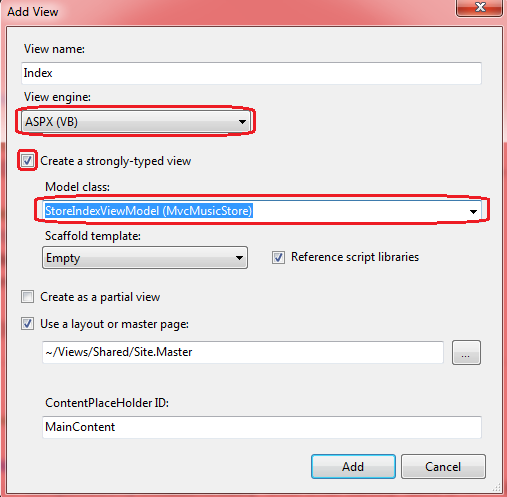
**'Create your view model**

**Dim viewModel = New StoreIndexViewModel With {.NumberOfGenres = genres.Count, .Genres = genres}**

**Return View(viewModel)**

**End Function**

Task 4 – Tạo một View mà sử dụng StoreIndexViewModel

* 1. Trong task này, ta sẽ tạo một trang View sử dụng đói tượng StoreIndexViewModel để hiện thị danh sách thể loại.
  2. Ta build lại chương trình trước khi tạo một trang View mới.
     1. 
     2. Figure 40
     3. Building the project
  3. **Chúng ta Add vào 1 view mới.**
     1. 
     2. Figure 41
     3. Adding a View – C#
     4. 
     5. Figure 42
     6. Adding a View - VB
  4. Ta sẽ chọn như hình bên dưới để chương trình tự động sinh ra code như mong muốn của chúng ta là thể hiện một list .
     1. 
     2. Figure 43
     3. Add View Dialog – C#
     4. 
     5. Figure 44
     6. Add View Dialog - VB

Task 5 – Cập nhật lại View Template

* 1. Trong file index.aspx ta thay thế lại như sau:
     1. HTML

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

**<h3> Browse Genres</h3>**

**<p> Select from <%: Model.NumberOfGenres %> genres</p>**

</asp:Content>

* 1. Ta tạo một tag <ul> để thể hiện danh sách các thể loại (gerne).
     1. HTML(C#)

<%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage<MvcMusicStore.ViewModels.StoreIndexViewModel>" %>

<asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">

**Store Genres**

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

<h3>Browse Genres</h3>

<p>Select from <%: Model.NumberOfGenres %> genres</p>

**<ul>**

**<% foreach (string genreName in Model.Genres) { %>**

**<li>**

**<%: genreName %>**

**</li>**

**<% } %>**

**</ul>**

</asp:Content>

* + 1. HTML(Visual Basic)

<%@ Page Title="" Language="VB" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage(Of MvcMusicStore.StoreIndexViewModel)" %>

<asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">

**Store Genres**

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

<h3>Browse Genres</h3>

<p>Select from <%: Model.NumberOfGenres %> genres</p>

**<ul>**

**<% For Each string genreName in Model.Genres %>**

**<li>**

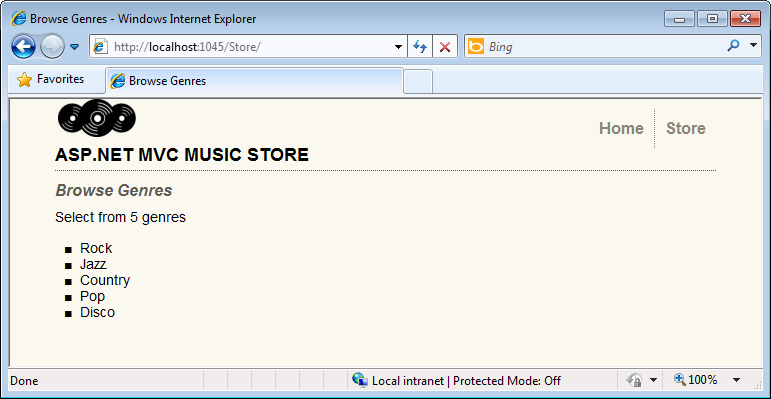
**<%: genreName %>**

**</li>**

**<% Next genreName %>**

**</ul>**

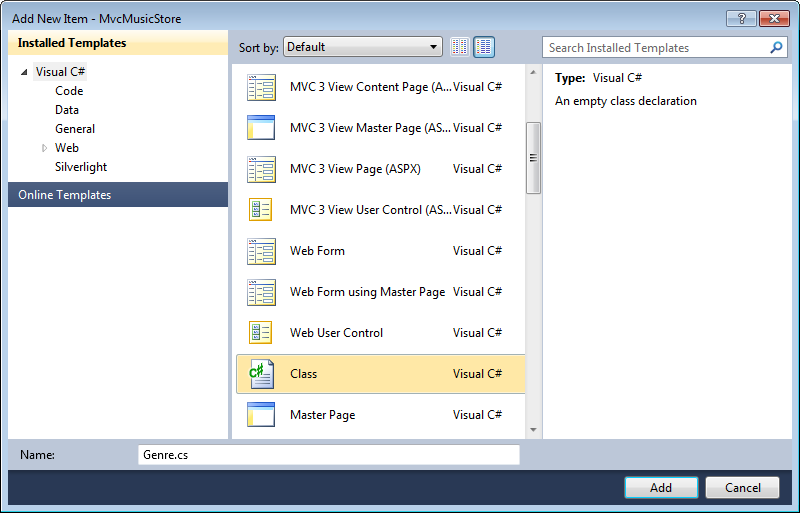
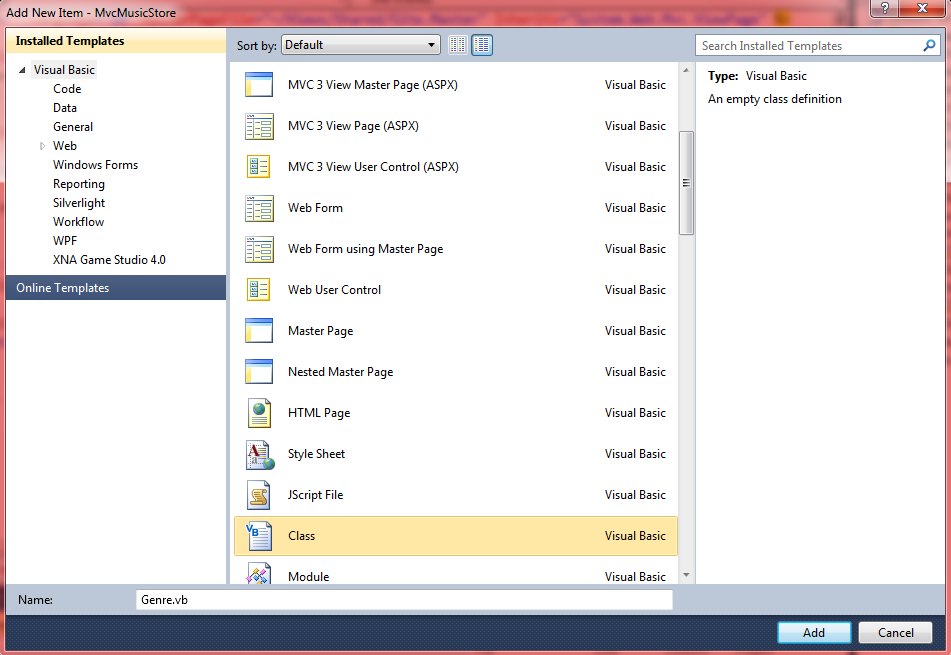
</asp:Content>

* 1. Nhấn F5 để chạy chương trình với URL **/Store** bạn sẽ thấy kết quả như bên dưới.
     1. 
     2. Figure 46
     3. View displaying a list of genres
  2. Đóng trình duyệt.

Bài 6:Sử dụng những tham số trong View

Trong bài này chúng ta sẽ học cách sử dụng các tham số được truyền vào View.

Task 1 – Thêm Một lớp Model

* 1. **Start | All Programs | Microsoft Visual Studio 2010 Express | Microsoft Visual Web Developer 2010 Express**.
  2. Thêm một lớp Genre trong Model. Nhắp phải vào thư mục **Models** và lựa **Add** một **Class** mới, đặt tên là **Genre**.
     1. 
     2. Figure 47
     3. Add Genre Model Class – C#
     4. 
     5. Figure 48
     6. Add Genre Model Class - VB
  3. Thêm một thuộc tính Name trong lớp **Genre**.
     1. C#

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace MvcMusicStore.Models

{

public class Genre

{

**public string Name { get; set; }**

}

}

* + 1. Visual Basic

**Public Class Genre**

**Public Property Name As String**

**End Class**

* 1. Tương tự, tạo thêm một lớp Album và thêm 2 thuộc tính Title và Genre.
     1. C#

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace MvcMusicStore.Models

{

public class Album

{

**public string Title { get; set; }**

**public Genre Genre { get; set; }**

}

}

* + 1. Visual Basic

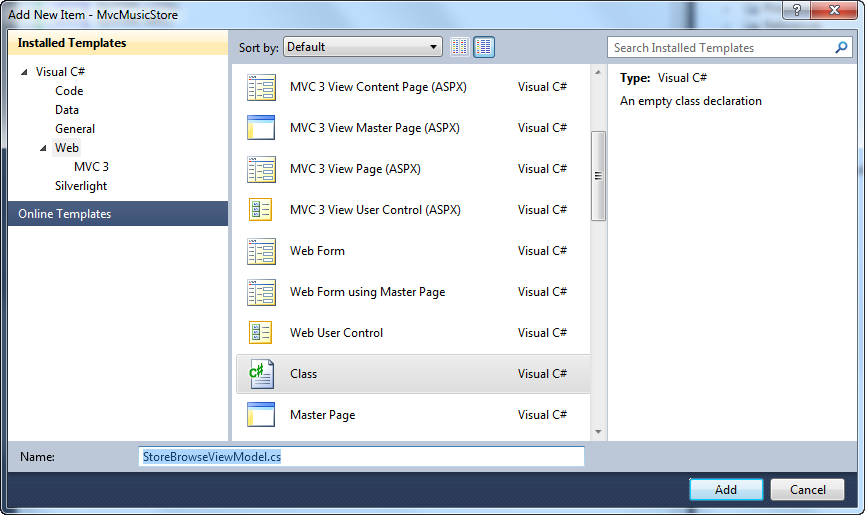
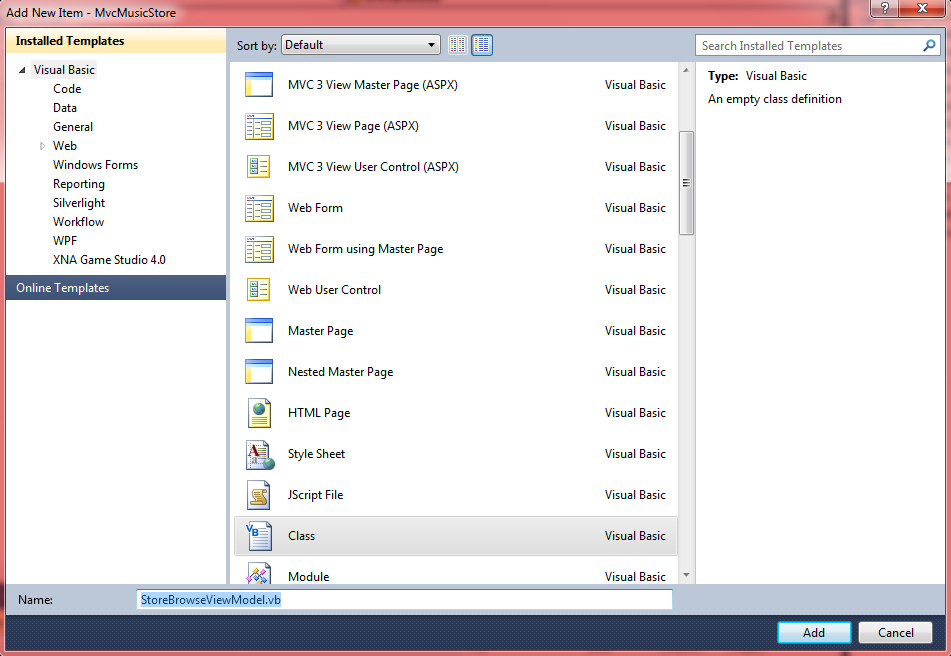
**Public Class Album**

**Public Property Title As String**

**Public Property Genre As Genre**

**End Class**

Task 2 – Thêm một StoreBrowseViewModel

* 1. Bổ sung một lớp **StoreBrowseViewModel**. Nhắp phải chuột vào thư mục **ViewModels**, chọn **Add**\**Class, đặt tên là** **StoreBrowseViewModel**
     1. 
     2. Figure 49
     3. Adding StoreBrowseViewModel Class – C#
     4. 
     5. Figure 50
     6. Adding StoreBrowseViewModel Class - VB
     7. C#

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

**using MvcMusicStore.Models;**

namespace MvcMusicStore.ViewModels

{

public class StoreBrowseViewModel

{

}

}

* 1. Thêm 2 thuộc tính Genre và List Albums
     1. C#

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using MvcMusicStore.Models;

namespace MvcMusicStore.ViewModels

{

public class StoreBrowseViewModel

{

**public Genre Genre { get; set; }**

**public List<Album> Albums { get; set; }**

}

}

* + 1. Visual Basic

Public Class StoreBrowseViewModel

**Public Property Genre As Genre**

**Public Property Albums As List(Of Album)**

End Class

Task 3 – Sử dung ViewModel bên trong một StoreController

* 1. Bạn phải bổ sung thêm **using MvcMusicStore.Models** trong **StoreController** để sử dụng
     1. C#

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using MvcMusicStore.ViewModels;

**using MvcMusicStore.Models;**

* 1. Thay thế phương thức Browse(). Bạn sẽ tạo một Genre và 2 Album mới.
     1. C#

**//**

**// GET: /Store/Browse?genre=Disco**

**public ActionResult Browse(string genre)**

**{**

**var genreModel = new Genre()**

**{**

**Name = genre**

**};**

**var albums = new List<Album>()**

**{**

**new Album() { Title = genre + " Album 1" },**

**new Album() { Title = genre + " Album 2" }**

**};**

**var viewModel = new StoreBrowseViewModel()**

**{**

**Genre = genreModel,**

**Albums = albums**

**};**

**return View(viewModel);**

**}**

* + 1. Visual Basic

**'**

**'GET: /Store/Browse?genre=Disco**

**Public Function Browse(ByVal genre As String) As ActionResult**

**Dim genreModel = New Genre With {.Name = genre}**

**Dim albums = New List(Of Album) From {New Album With {.Title = genre & " Album 1"},**

**New Album With {.Title = genre & " Album 2"}}**

**Dim viewModel = New StoreBrowseViewModel With**

**{.Genre = genreModel,**

**.Albums = albums}**

**Return View(viewModel)**

**End Function**

* 1. Trong phương thức **Details** bạn sẽ tạo mộ đối tượng **Album** và trả về View.
     1. C#

**//**

**// GET: /Store/Details/5**

**public ActionResult Details(int id)**

**{**

**var album = new Album { Title = "Sample Album" };**

**return View(album);**

**}**

* + 1. Visual Basic

**'**

**'GET: /Store/Details/5**

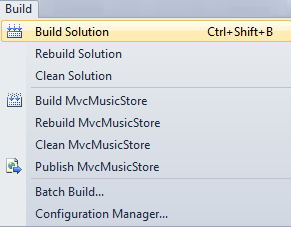
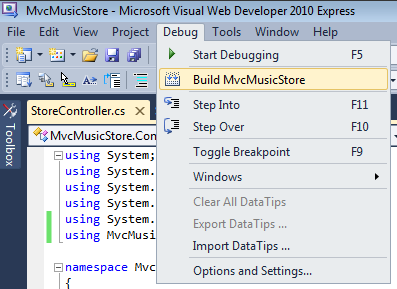
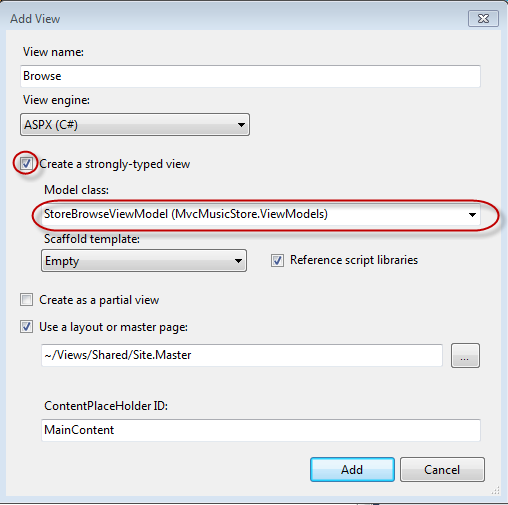
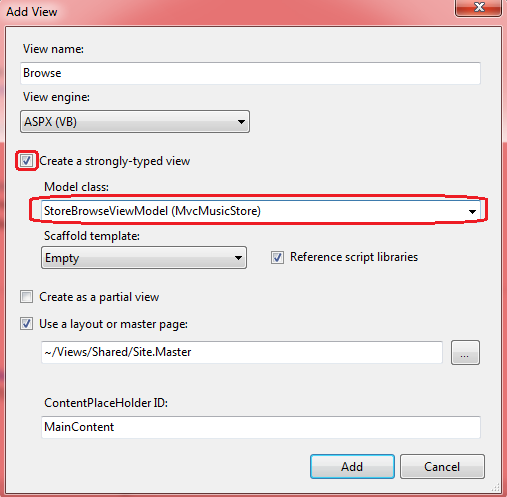
**Public Function Details(ByVal id As Integer) As ActionResult**

**Dim album = New Album With {.Title = "Sample Album"}**

**Return View(album)**

**End Function**

Task 4 – Add thêm một View mới cho phương thức Browse

* 1. Build lại chương trình trước khi add view mới.
     1. 
     2. Figure 51
     3. Building the project in Visual Web Developer 2010
     4. 
     5. Figure 52
     6. Building the project in Visual Studio 2010
  2. Nhắp chuột phải vào phương thức Browse trong StoreController và nhắp Add View.
     1. 
     2. Figure 53
     3. Adding a Browse View – C#
     4. 
     5. Figure 54
     6. Adding a Browse View - VB
  3. Thay đổi File Browse.aspx lại như sau.
     1. HTML(C#)

<%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage<MvcMusicStore.ViewModels.StoreBrowseViewModel>" %>

<asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">

**Browse Albums**

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

**<h2>Browsing Genre: <%: Model.Genre.Name %></h2>**

**<ul>**

**<% foreach (var album in Model.Albums)**

**{ %>**

**<li><%: album.Title %></li>**

**<% } %>**

**</ul>**

</asp:Content>

* + 1. HTML(Visual Basic)

<%@ Page Title="" Language="VB" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage(Of MvcMusicStore.StoreBrowseViewModel)" %>

<asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">

**Browse Albums**

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

**<h2>Browsing Genre: <%: Model.Genre.Name %></h2>**

**<ul>**

**<% For Each album In Model.Albums**

**%>**

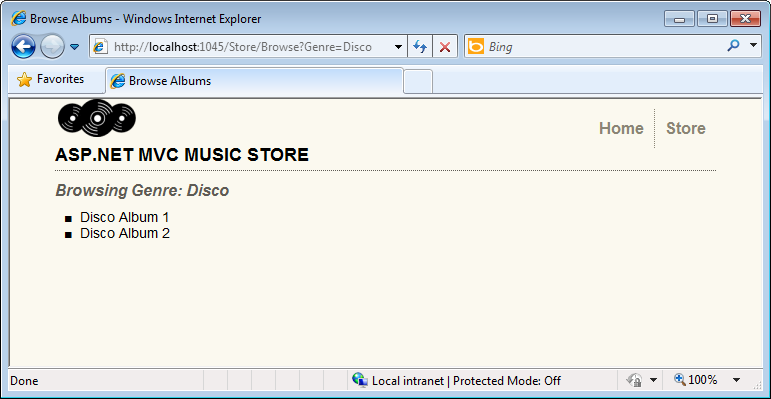
**<li><%: album.Title %></li>**

**<% Next album %>**

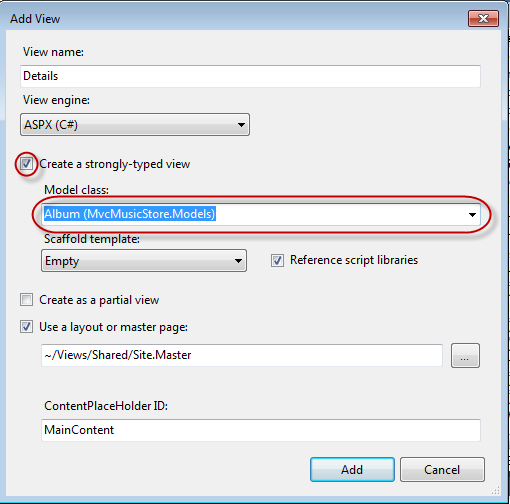
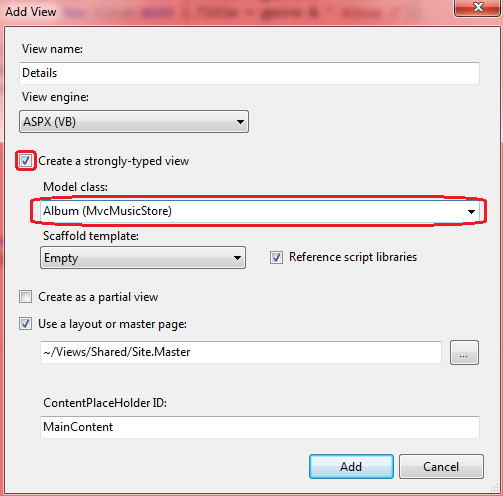
**</ul>**

</asp:Content>

Task 5 – Chạy chương trình

* 1. Nhấn F5 để tạo chương trình.
  2. Thay đổi URL thành **/Store/Browse?Genre=Disco** ta sẽ thấy xuất hiện 2 Ablums
     1. 
     2. Figure 55
     3. Browsing Store Disco Albums

Task 6 – Trình diễn thông tin về một Album

* 1. Nhắp chuột phải trong phương thức **Details** trong **StoreController** và nhắp **Add View**
     1. 
     2. Figure 56
     3. Adding a Details View – C#
     4. 
     5. Figure 57
     6. Adding a Details View - VB
  2. Điều chỉnh lại lại Details.aspx để hiện ra một Ablums.
     1. HTML(C#)

<%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage<MvcMusicStore.Models.Album>" %>

<asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">

Details

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

<h2>Album: <%: Model.Title %></h2>

</asp:Content>

* + 1. HTML(Visual Basic)

<%@ Page Title="" Language="VB" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage(Of MvcMusicStore. Album)" %>

<asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">

Details

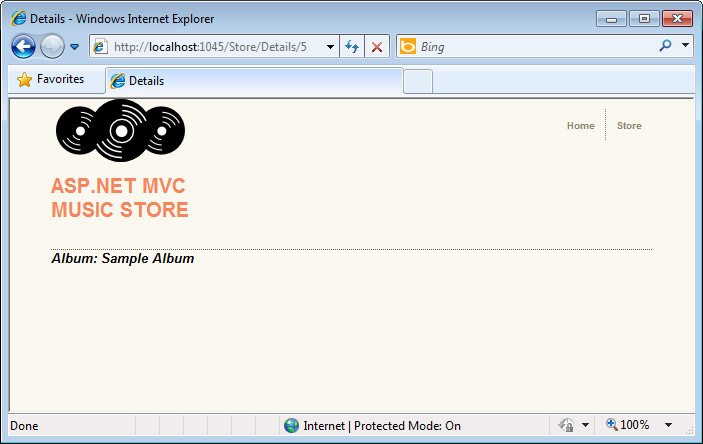
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

<h2>Album: <%: Model.Title %></h2>

</asp:Content>

Task 7 – Chạy ứng dụng

* 1. Bấm F5 để chạy ứng dụng.
  2. Thay đổi URL thành **/Store/Details/5** để hiên thông tin Albums có Id là 5.
     1. 
     2. Figure 58
     3. Browsing Album’s Detail

Task 8 – Thêm Links liên kết giữa các trang.

Trong trang Views/Store/Index.aspx thay đổi code như sau:

* + 1. C#

<%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage<MvcMusicStore.ViewModels.StoreIndexViewModel)" %>

<asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">

Browse Genres

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

<h2>Browse Genres</h2>

<p>Select from **<%: Model.NumberOfGenres** %></p>

<ul>

<% foreach (string genreName in Model.Genres)

{ %>

<li>

**<%: Html.ActionLink(genreName, "Browse", new { genre = genreName }, null) %>**

</li>

<% } %>

</ul>

</asp:Content>

* + 1. HTML(Visual Basic)

<%@ Page Title="" Language="VB" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage(Of MvcMusicStore. StoreIndexViewModel)" %>

<asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">

Browse Genres

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

<h2>Browse Genres</h2>

<p>Select from **<%: Model.NumberOfGenres** %></p>

<ul>

<% For Each genreName In Model.Genres

%>

<li>

**<%: Html.ActionLink(genreName, "Browse", new With {Key .genre = genreName }, Nothing) %>**

</li>

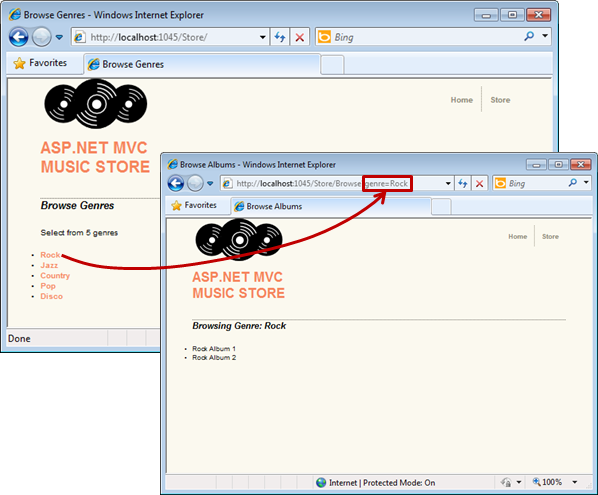
<% Next genreName %>

</ul>

</asp:Content>

Task 9 – Chạy ứng dụng.

* 1. Nhấn F5 để chạy ứng dụng.
  2. Thay đổi URL thành **/Store**/. Sau đó nhắp vào danh sách để kiểm tra



* + 1. Figure 59
    2. Browsing Genres with links to Browse page

Task 10 – Sử dụng Dynamic ViewModel Collection để truyền các giá trị.

Trong task này, chúng ta sẽ sử dụng ViewBag dùng để truyền dữ liệu từ controller tới các view.

* 1. Thay đổi phương thức Index() như bên dưới.
     1. C#

public ActionResult Index()

{

// Create list of genres

var genres = new List<string> { "Rock", "Jazz", "Country", "Pop", "Disco" };

// Create your view model

var viewModel = new StoreIndexViewModel

{

NumberOfGenres = genres.Count,

Genres = genres

};

**ViewBag.Starred = new List<string> { "Rock", "Jazz" };**

return View(viewModel);

}

* + 1. Visual Basic

Public Function Index() As ActionResult

'Create list of genres

Dim genres = New List(Of String) From {"Rock", "Jazz", "Country", "Pop", "Disco"}

'Create your view model

Dim viewModel = New StoreIndexViewModel With

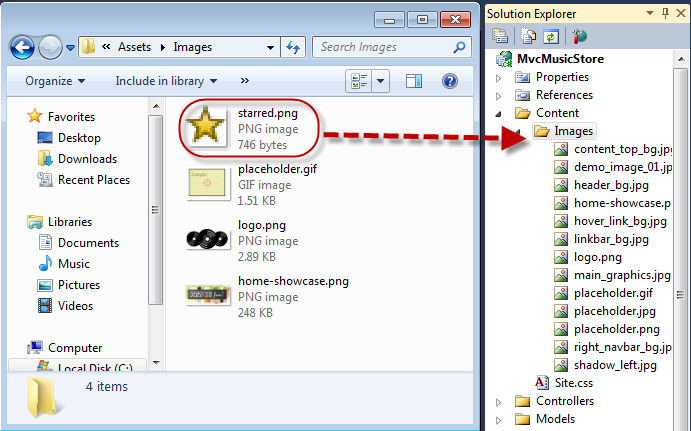
{.NumberOfGenres = genres.Count,

.Genres = genres}

**ViewBag.Starred = New List(Of String) From {"Rock", "Jazz"}**

Return View(viewModel)

End Function

* 1. Tạo một icon ngôi sao cho Ablum bằng cách add thêm hình vào thư mục Content \Images.
     1. 
     2. Figure 60
     3. Adding star image to the solution
     4. HTML(C#)

<%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage<MvcMusicStore.ViewModels.StoreIndexViewModel>" %>

<asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">

Browse Genres

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

<h2>Browse Genres</h2>

<p>Select from <%: Model.NumberOfGenres %></p>

<ul>

<% foreach (string genreName in Model.Genres)

{ %>

<li>

<%: Html.ActionLink(genreName, "Browse", new { genre = genreName }, null) %>

**<% if (ViewBag.Starred.Contains(genreName)) { %>**

**<img src="../../Content/Images/starred.png" alt="Starred element" />**

**<% } %>**

</li>

<% } %>

</ul>

**<br/>**

**<h5> <img src="../../Content/Images/starred.png" alt="Starred element" /> Starred genres 20% Off! </h5>**

</asp:Content>

* + 1. HTML(Visual Basic)

<%@ Page Title="" Language="VB" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage(Of MvcMusicStore. StoreIndexViewModel)" %>

<asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">

Browse Genres

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

<h2>Browse Genres</h2>

<p>Select from <%: Model.NumberOfGenres %></p>

<ul>

<% For Each genreName In Model.Genres

%>

<li>

<%: Html.ActionLink(genreName, "Browse", new With {Key .genre = genreName }, Nothing) %>

**<% If ViewBag.Starred.Contains(genreName) Then%>**

**<img src="../../Content/Images/starred.png" alt="Starred element" />**

**<% End If %>**

</li>

<% Next genreName %>

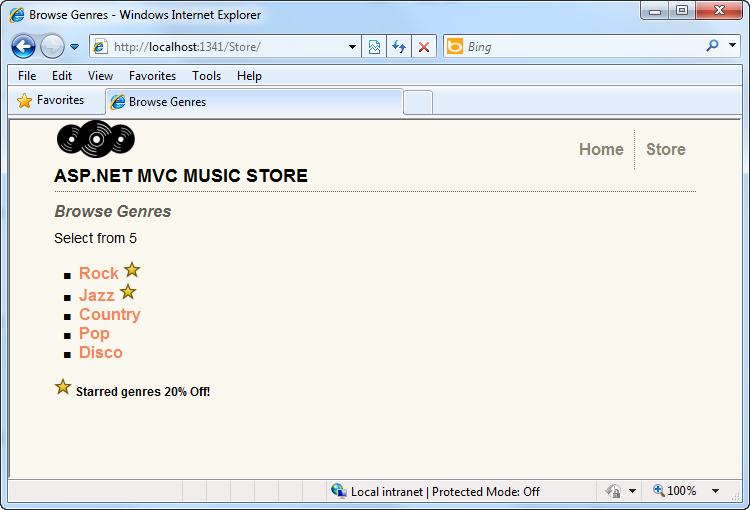
</ul>

**<br/>**

**<h5> <img src="../../Content/Images/starred.png" alt="Starred element" /> Starred genres 20% Off! </h5>**

</asp:Content>

Task 11 – Chạy ứng dụng

* + 1. 
    2. Figure 61
    3. Browsing Genres with starred elements