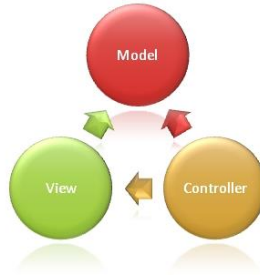


1.1 Giới thiệu ASP.NET MVC

ASP.NET MVC là một framework web ứng dụng mã nguồn mở thực thi kiến trúc MVC (model-view-controller). MVC là một kiến trúc phần mềm chia nhỏ một ứng dụng thành ba thành phần riêng biệt bao gồm model, view, và controller.



- **Models:** các đối tượng model là các phần của ứng dụng có nhiệm vụ lưu trữ miền dữ liệu của ứng dụng. Thông thường, các đối tượng model lấy và lưu trữ thông tin, trạng thái của các đối tượng từ một lớp được ánh xạ từ một bảng trong CSDL.
- **Views:** là các thành phần hiển thị các thông tin trên các giao diện. Thông thường, các thông tin được lấy từ dữ liệu của các model.
- **Controllers:** là các thành phần chịu trách nhiệm xử lý tương tác với người dùng, nó làm việc với model và hiển thị thông tin lên các view. Trong một ứng dụng MVC, view chỉ có trách nhiệm hiển thị thông tin, controller sẽ điều khiển việc tương tác người dùng từ các thông tin nhập của người dùng và trả về kết quả cho người dùng thông qua các view.

1.2 ASP.NET MVC cơ bản

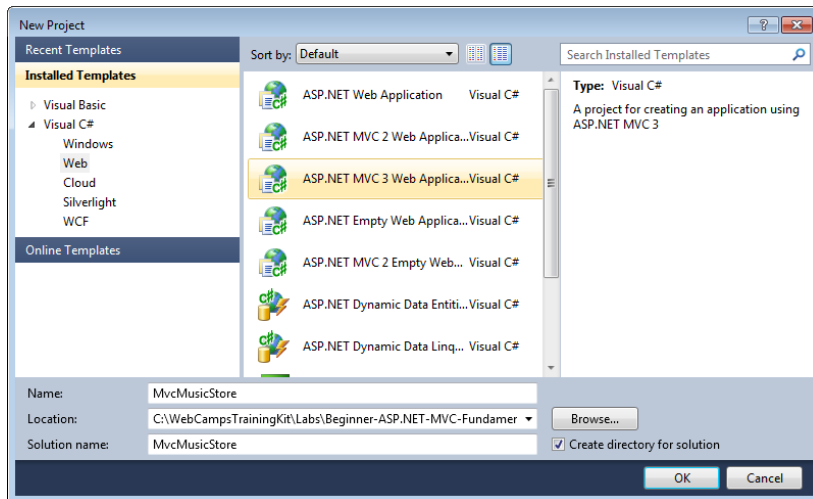
1.2.1 Tạo Một Ứng Dụng Web Trên Nền Tảng Asp.Net MVC

Trong phần này, chúng ta sẽ tìm hiểu làm thế nào để tạo ra một ứng dụng ASP.NET MVC sử dụng ASP.NET MVC 3 trong Microsoft Visual Web Developer 2010 Express (hoặc Microsoft Visual Studio 2010) cũng như cách tổ chức các thư mục của nó. Ngoài ra, chúng ta sẽ tìm hiểu làm thế nào để thêm một Controller mới và hiển thị một chuỗi đơn giản ở màn hình ứng dụng.

Bước 1 – Tạo Project ASP.NET MVC

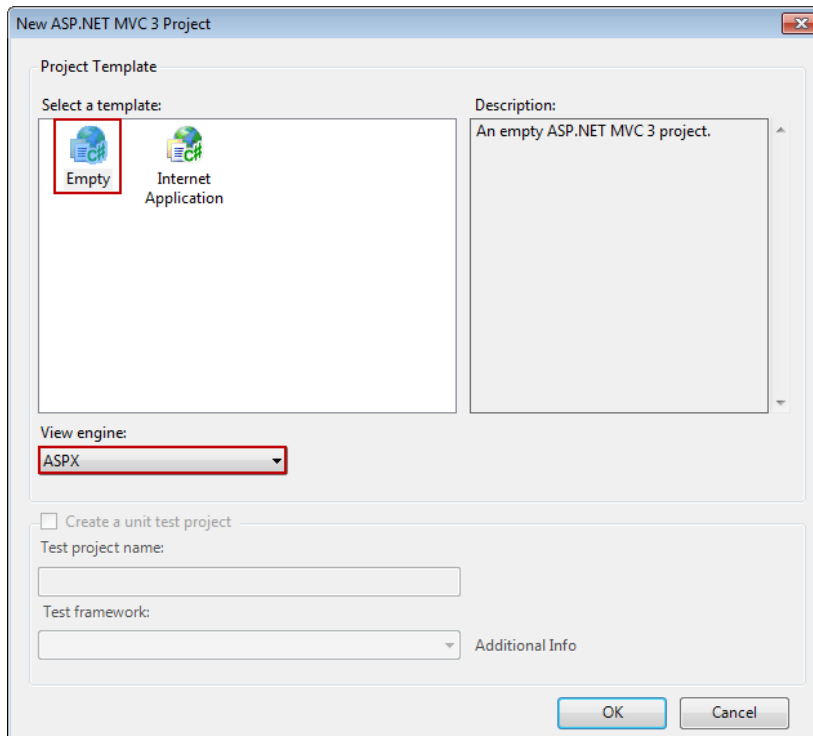
Trong phần này, chúng ta sẽ tạo ra một ứng dụng ASP.NET MVC sử dụng template MVC.

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#, Web .**
4. Thay đổi tên thành **MvcMusicStore**.
5. Lựa chọn đường dẫn ví dụ **\Source\Ex01-CreatingMusicStoreProject\Begin**. Chọn **OK**.



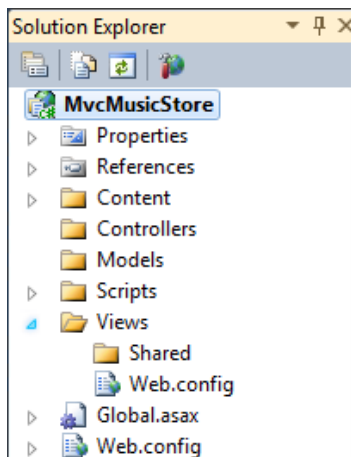
6. **New ASP.NET MVC 3 Project > Empty**

View engine : ASPX. Chọn **OK**.



Bước 2 – Xem qua cấu trúc của Solution

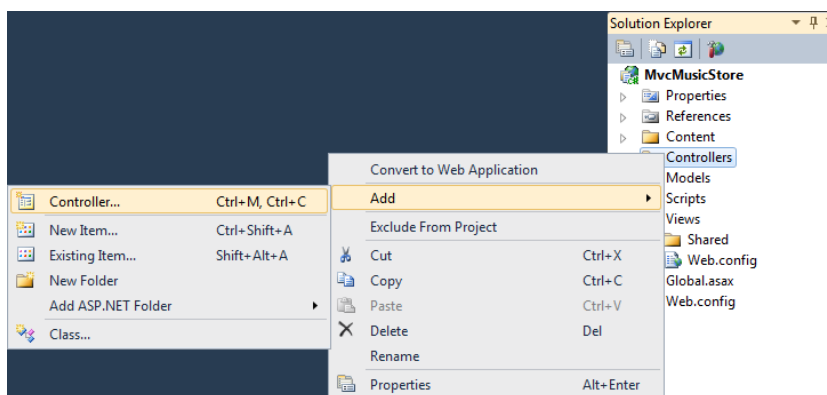
ASP.NET MVC giúp chúng ta tạo ra các ứng dụng Web được cấu trúc để hỗ trợ các mô hình MVC. Một ứng dụng Web MVC mới được cấu hình để có các thư mục cần thiết, và các tập tin cấu hình.



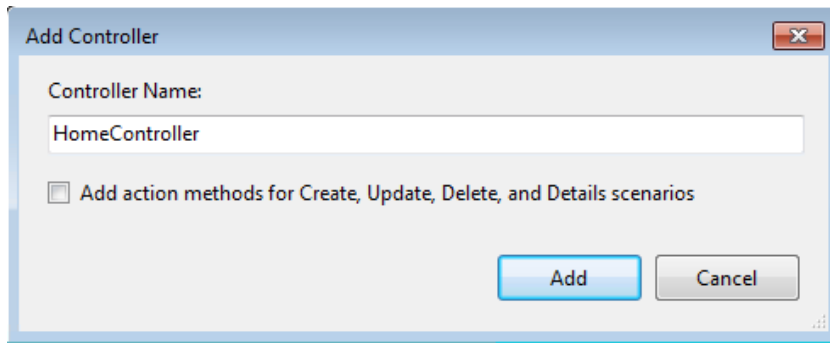
- **Content.** Thư mục này được khuyến khích là chứa các file hình ảnh, css....
- **Controllers.** Nó có tác dụng điều hướng những request và kết hợp với những model, truyền tải dữ liệu tới View.
- **Models.** Thư mục chứa những lớp cho việc hiện thực hoá dữ liệu
- **Scripts.** Thư mục này được khuyến khích chứa các thư viện javascript
- **Views.** Thư mục chứa các trang web trình diễn cho người xem (aspx, .ascx, .cshtml , .master files)
- **Views\Shared.** Thư mục Shared này có thể được sử dụng để chứa các trang master page của ứng dụng web.
- **Global.asax** là file để thiết lập đường dẫn URL, và **Web.config** là file để thiết lập ứng dụng.

Bước 3 –Thêm một Controller

1. Chuột phải vào thư mục **Controllers** trong Solution Explorer, và chọn **Add > Controller** để thêm một controller mới.



2. Cửa sổ **Add Controller** xuất hiện. Chúng ta đặt tên là **HomeController** và nhấn **Add**.

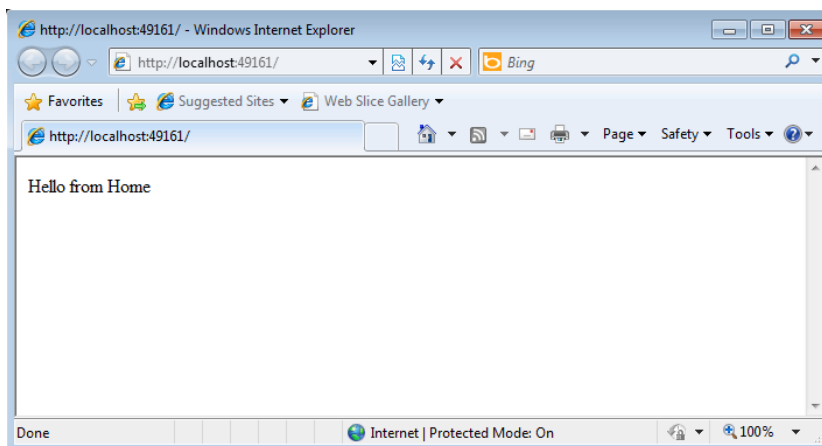


3. File **HomeController.cs** được tạo trong thư mục **Controllers**. Để HomeController trả về một chuỗi, thay thế phương thức Index() với dòng code sau

```
public string Index()
{
    return "Hello from Home";
}
```

Bước 4 – Chạy ứng dụng

Nhấn F5 để chạy ứng dụng. Project được biên dịch và các máy chủ Web ASP.NET được xây dựng trên Visual Web Developer bắt đầu chạy. Visual Web Developer sẽ tự động mở một trình duyệt web.



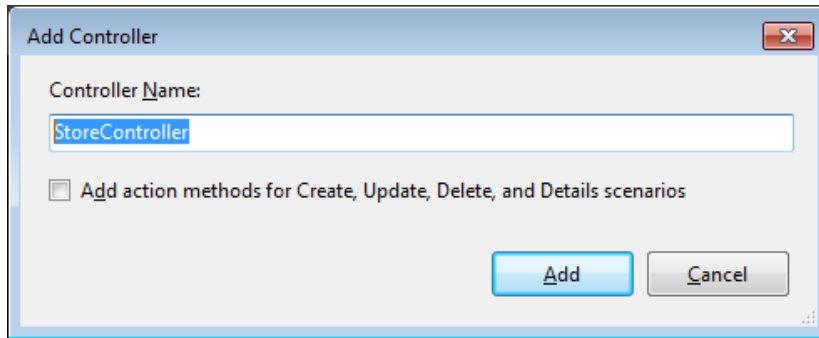
1.2.2 Tạo một Controller

Trong phần này, chúng ta sẽ tìm hiểu làm thế nào để tạo ra Controller để thực hiện chức năng đơn giản của Music Store. Controller sẽ xác định mọi hành động xử lý từng yêu cầu cụ thể:

- Liệt kê tất cả những albums trong một loại nhạc cụ thể.
- Chi tiết một albums.

Bước 1 – Thêm một StoreController .

1. Start | All Programs | Microsoft Visual Studio 2010 Express | Microsoft Visual Web Developer 2010 Express.
2. **Thêm một Controller.** Thay đổi **Controller Name** thành **StoreController**, và nhấn nút **Add**.



Bước 2 – Điều chỉnh các hành động trong StoreController

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

```
public string Index()
{
    return "Hello from Store.Index()";
}
```

2. Thêm một phương thức Browse() để trả về một chuỗi.

```
//
// GET: /Store/Browse

public string Browse()
{
    return "Hello from Store.Browse()";
}

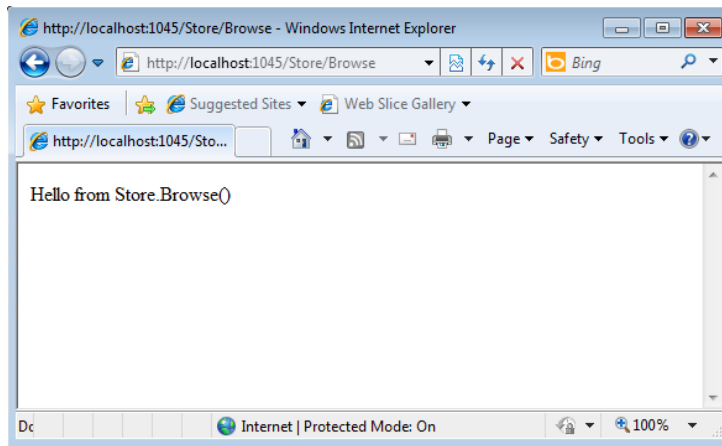
//
// GET: /Store/Details

public string Details()
{
    return "Hello from Store.Details()";
}
```

Bước 3 – Chạy ứng dụng

Chạy ứng dụng trên một trình duyệt.

1. Nhấn F5 để chạy ứng dụng.
2. Thay đổi đường dẫn như bên dưới chúng ta sẽ thấy kết quả hiện ra trong trình duyệt
 - a. /Store. → “Hello from Store.Index()”.
 - b. /Store/Browse. → “Hello from Store.Browse()”.
 - c. /Store/Details. → “Hello from Store.Details()”.



3. Đóng trình duyệt

1.2.3 Truyền một tham số tới Controller

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

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

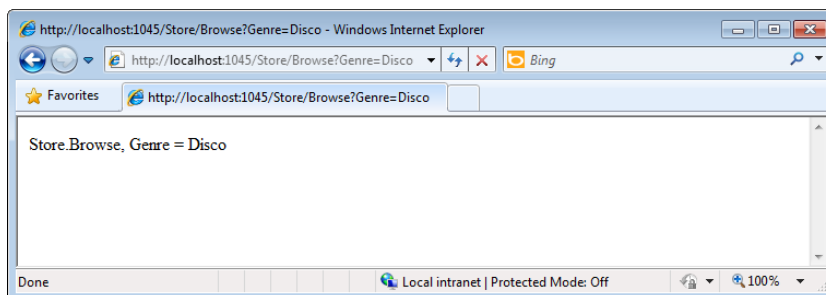
```
//  
// GET: /Store/Browse?genre=Disco  
  
public string Browse(string genre)  
{  
    string message = HttpUtility.HtmlEncode("Store.Browse, Genre = " + genre);  
    return message;  
}
```

Chú ý: Chúng ta sử dụng `HttpUtility.HtmlEncode` để chống người dùng chèn script độc hại vào chương trình. Ví dụ như:

`/Store/Browse?Genre=<script>>window.location='http://hackersite.com'</script>.`

Bước 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



3. Đóng trình duyệt

Bước 3 – Thêm một Id vào một Controller

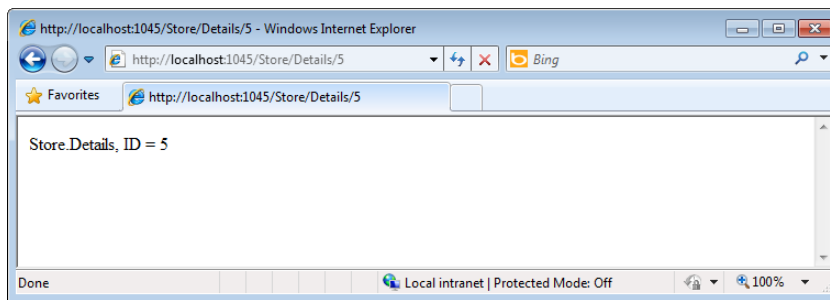
Trong URL `Store/Detail/5`, **Id** sẽ được truyền vào giá trị là **5**.

Trong phương thức Details có một tham số Id để lấy từ đường dẫn URL

```
//  
// GET: /Store/Details/5  
  
public string Details(int id)  
{  
    string message = "Store.Details, ID = " + id;  
  
    return message;  
}
```

Bước 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.

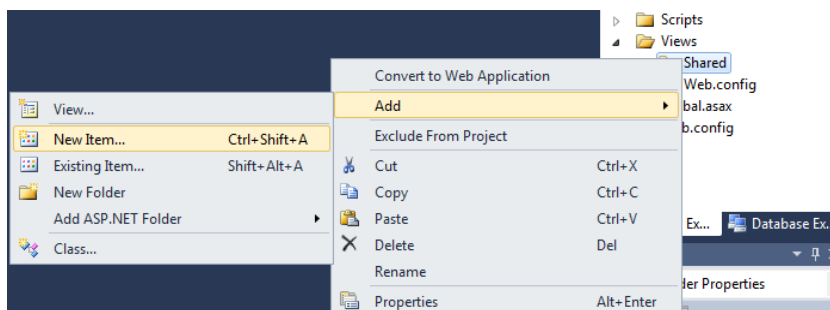


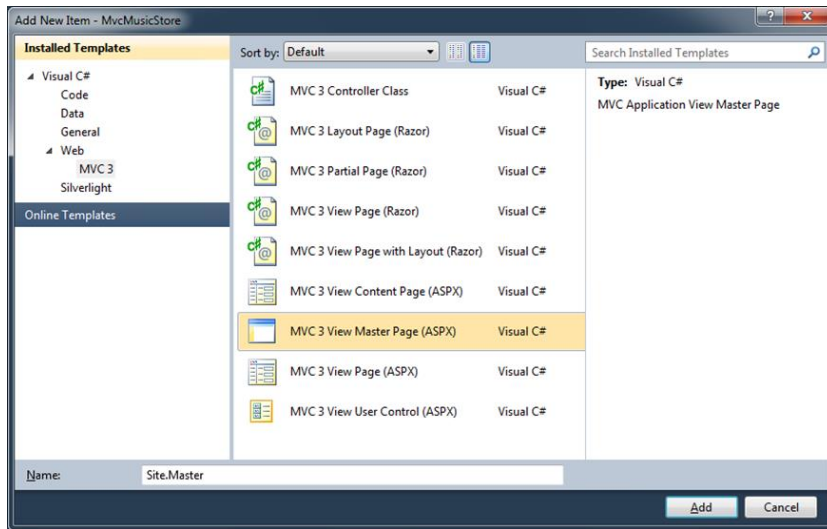
1.2.4 Tạo một View

Bước 1 – Thêm vào 1 MasterPage

Master Page như là một layout của trang web.

1. Start | All Programs | Microsoft Visual Studio 2010 Express | Microsoft Visual Web Developer 2010 Express.
2. Trong thư mục **Shared** chúng ta bổ sung một item mới là một trang Master page





```
<%@ 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>
```

3. Thêm vào một header trong master page

```
<%@ 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">
```

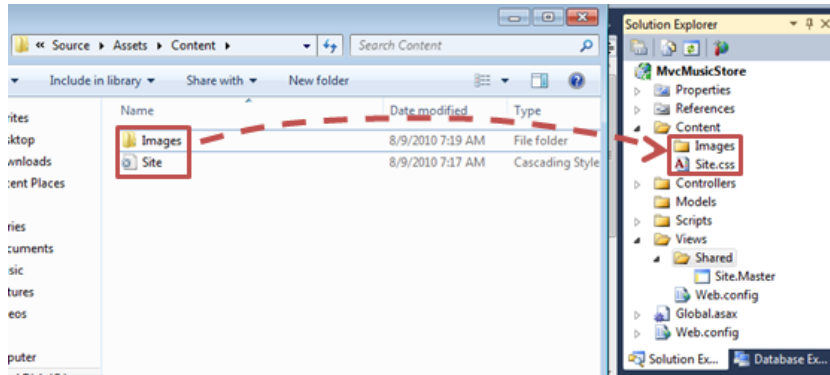


```

        </asp:ContentPlaceholder>
    </div>
</body>
</html>

```

Bước 2 – Thêm vào một Site CSS



Chú ý: các tài nguyên này có thể download từ https://docs.google.com/a/uit.edu.vn/file/d/0Bxc4-oCY_UPIURQSDgwN2J6ZEK/edit?usp=drive_web

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

```

<head runat="server">
    <link href="/Content/Site.css" rel="Stylesheet" type="text/css" />
    <title><asp:ContentPlaceholder ID="TitleContent" runat="server" /></title>
</head>

```

Bước 3 – Thêm một template View

- Để sử dụng một Template View, trong phương thức Index() cần trả về một View();

```

public class HomeController : Controller
{
    //
    // GET: /Home/

    public ActionResult Index()
    {
        return View();
    }
}

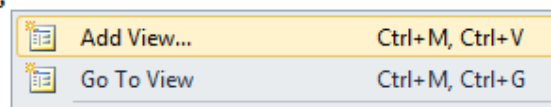
```

- Bây giờ, chúng ta cần phải thêm một trang View thích hợp. Để làm điều này, kích phải vào bên trong phương thức **Index** và chọn **Add View**. Điều này sẽ mở hộp thoại **Add View**.

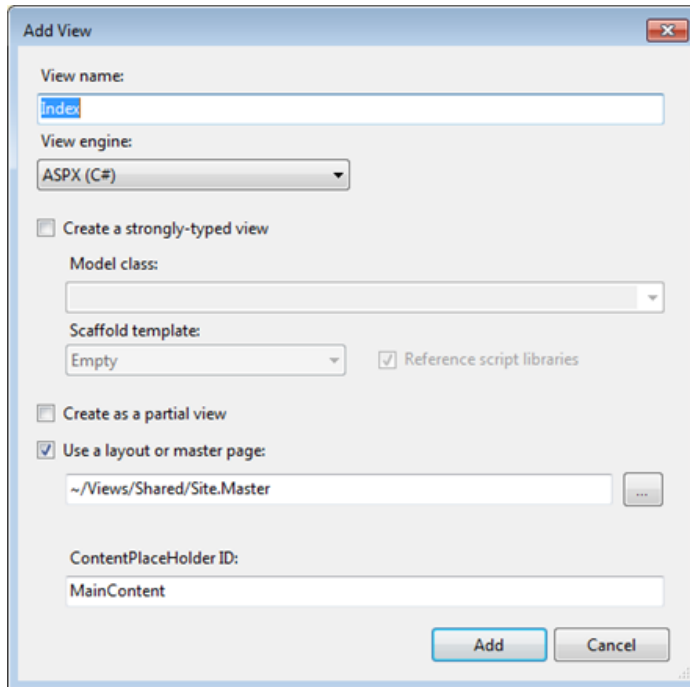
```

public ActionResult Index()
{
    return View();
}

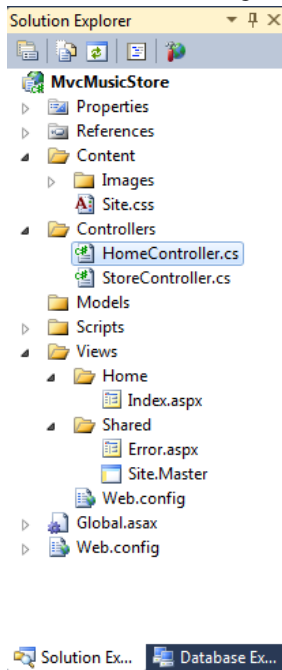
```



- Hộp thoại **Add View** xuất hiện. Cho phép tạo một trang View



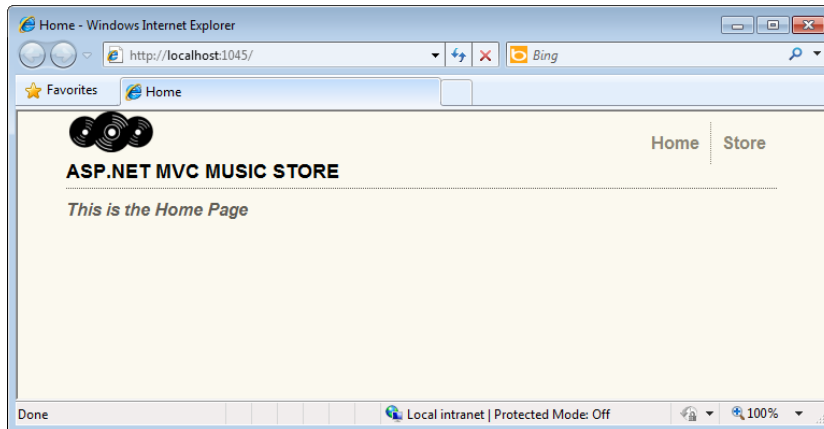
4. Visual Studio tạo một trang có tên **Index.aspx** bên trong thư mục Views\Home



5. 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

```
<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>
```

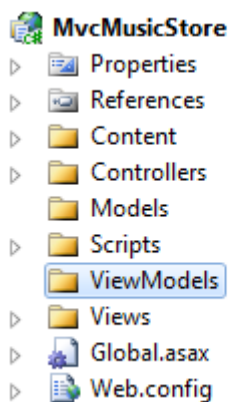
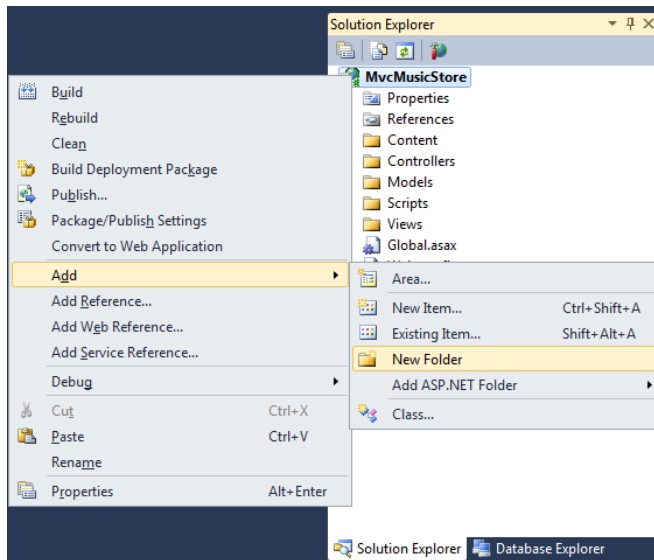
6. Ấn F5 để chạy ứng dụng.



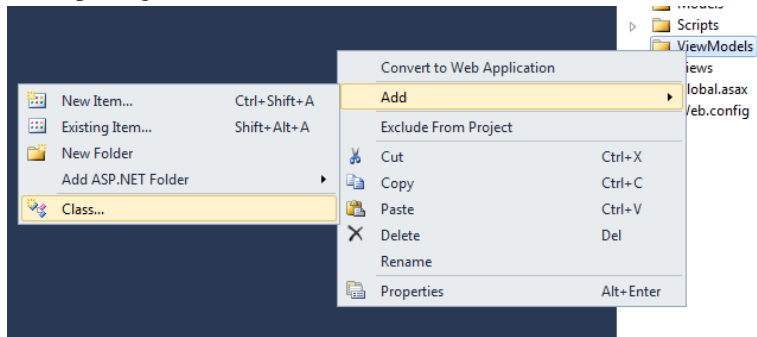
1.2.5 Tạo một View Mode

Bước 1 – Tạo một lớp Model

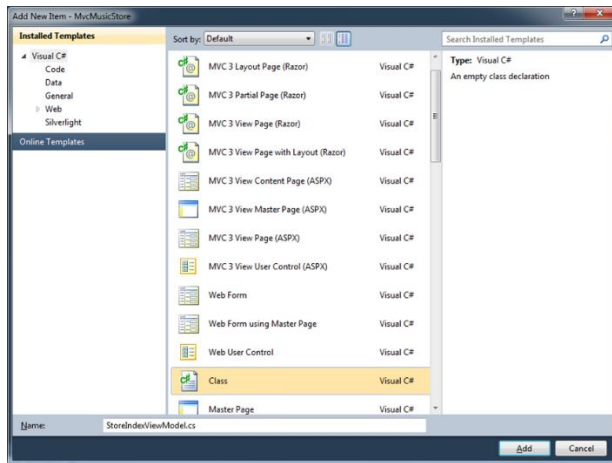
1. Start | All Programs | Microsoft Visual Studio 2010 Express | Microsoft Visual Web Developer 2010 Express.
2. Tạo một thư mục **ViewModels**.



3. Tạo một lớp trong ViewModels



4. Đặt tên là StoreIndexViewModel.



Bước 2 – Thêm những thuộc tính vào lớp ViewModel

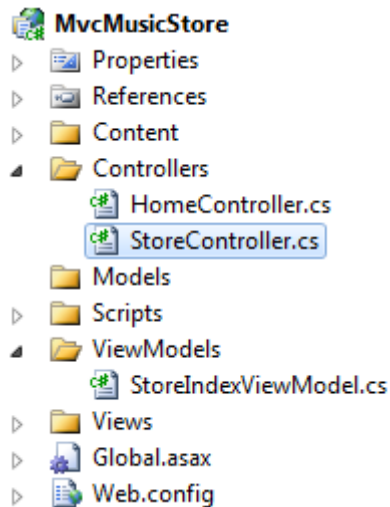
Ta sẽ thêm 2 thuộc tính

- Một là NumberOfGenres
- Hai là list Genres

```
public class StoreIndexViewModel
{
    public int NumberOfGenres { get; set; }
    public List<string> Genres { get; set; }
}
```

Bước 3 – Cập nhật lại StoreController để sử dụng StoreIndexViewModel

1. Mở lớp StoreController.



2. Để sử dụng StoreIndexViewModel ta thêm vào namespace.

```
using MvcMusicStore.ViewModels;
```

3. Thay thế phương thức index() lại như sau:

```
public ActionResult Index()
{
    // Create a list of genres
    var genres = new List<string> {"Rock", "Jazz", "Country", "Pop", "Disco"};

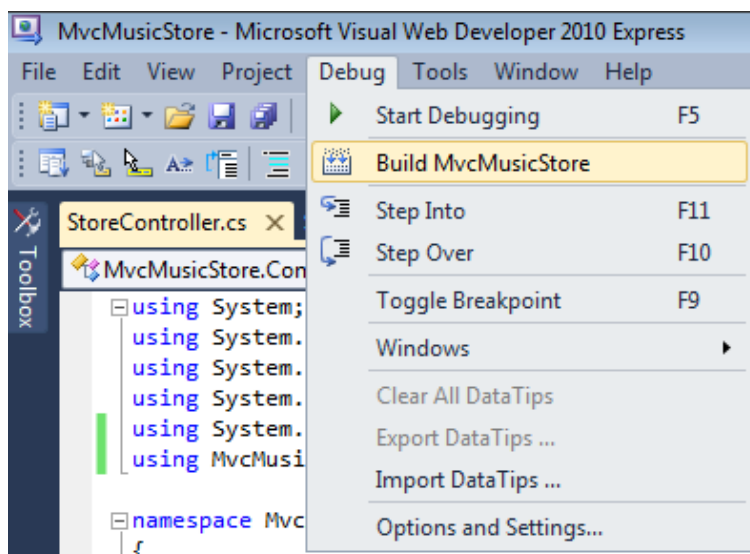
    // Create our view model
    var viewModel = new StoreIndexViewModel {
        NumberOfGenres = genres.Count(),
        Genres = genres
    };

    return View(viewModel);
}
```

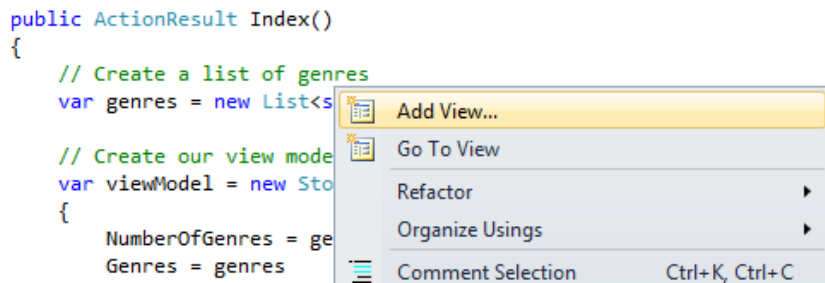
Bước 4 – Tạo một View sử dụng StoreIndexViewModel

Trong bước này, ta sẽ tạo một trang View sử dụng đối tượng **StoreIndexViewModel** để hiển thị danh sách thể loại.

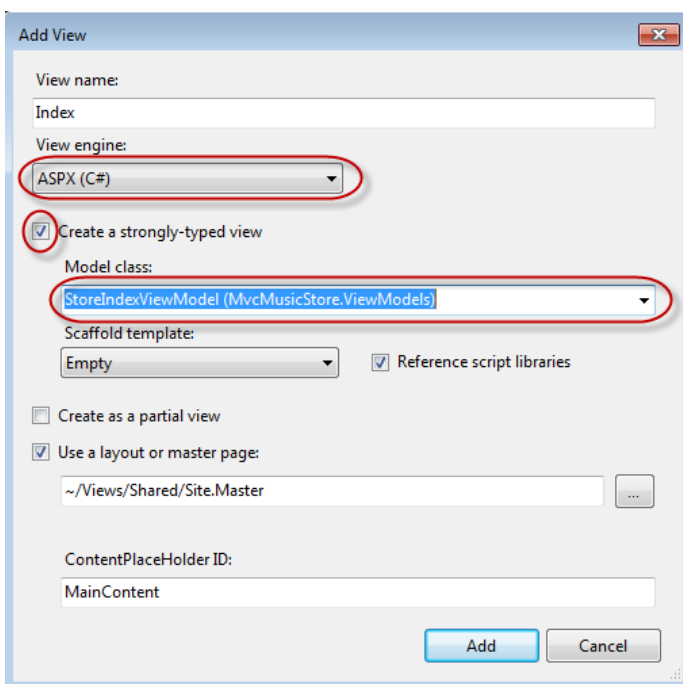
1. Ta biên dịch lại chương trình trước khi tạo một trang View mới.



- Chúng ta thêm vào một View mới.



- 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 danh sách.



Bước 5 – Cập nhật lại template View

- Trong file index.aspx ta thay thế lại như sau:

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

    <h3> Browse Genres</h3>

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

</asp:Content>
```

- Ta tạo một tag để thể hiện danh sách các thể loại (gerne).

```
<%@ 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>

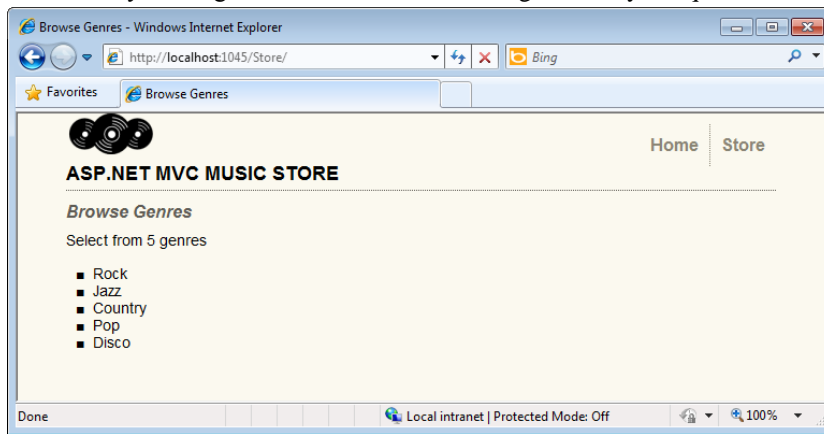
    <p>Select from <%= Model.NumberOfGenres %> genres</p>

    <ul>
        <% foreach (string genreName in Model.Genres) { %>
            <li>
                <%= genreName %>
            </li>
        <% } %>
    </ul>

</asp:Content>

```

3. Nhấn F5 để chạy chương trình với URL **/Store** chúng ta sẽ thấy kết quả như bên dưới.



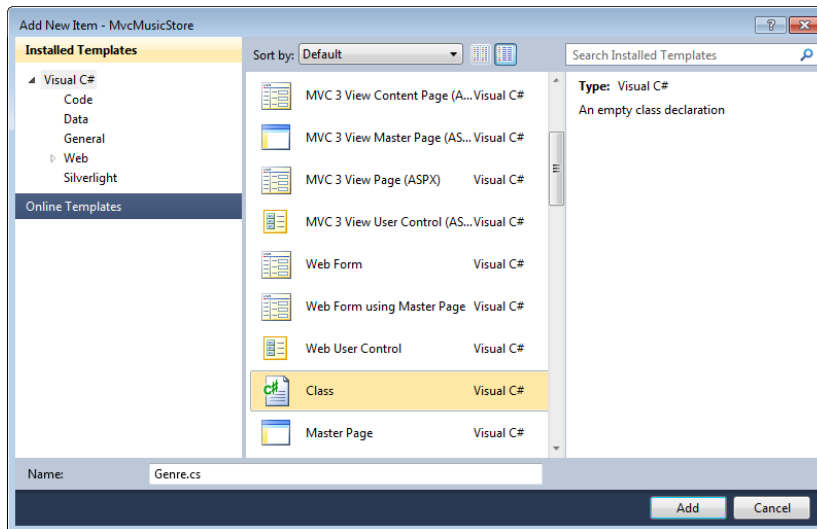
4. Đóng trình duyệt.

1.2.6 Sử dụng những tham số trong View

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

Bước 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. Nhấp phải vào thư mục **Models** và lựa chọn thêm một lớp mới, đặt tên là **Genre**.



3. Thêm một thuộc tính Name trong lớp **Genre**.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace MvcMusicStore.Models
{
    public class Genre
    {
        public string Name { get; set; }
    }
}
```

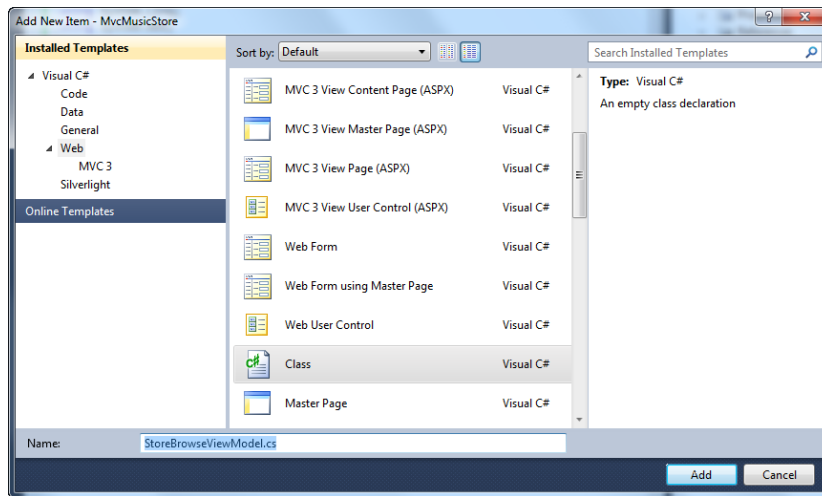
4. Tương tự, tạo thêm một lớp Album và thêm 2 thuộc tính Title và Genre.

```
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; }
    }
}
```

Bước 2 – Thêm một StoreBrowseViewModel

1. Bổ sung một lớp **StoreBrowseViewModel**. Nhấp phải chuột vào thư mục **ViewModels**, chọn **Add\Class**, đặt tên là **StoreBrowseViewModel**



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using MvcMusicStore.Models;

namespace MvcMusicStore.ViewModels
{
    public class StoreBrowseViewModel
    {
    }
}
```

2. Thêm 2 thuộc tính Genre và List Albums

```
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; }
    }
}
```

Bước 3 – Sử dụng ViewModel bên trong một StoreController

1. Chúng ta phải bổ sung thêm **using MvcMusicStore.Models** trong **StoreController** để sử dụng .

```
using System;
using System.Collections.Generic;
```

```
using System.Linq;
using System.Web;
using System.Web.Mvc;
```

```
using MvcMusicStore.ViewModels;
using MvcMusicStore.Models;
```

2. Thay thế phương thức Browse(). Chúng ta sẽ tạo một Genre và 2 Album mới.

```
//
// 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);
}
```

3. Trong phương thức **Details** chúng ta sẽ tạo một đối tượng **Album** và trả về View.

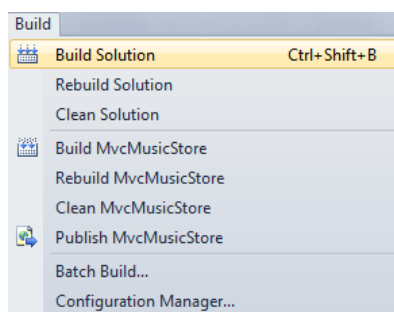
```
//
// GET: /Store/Details/5

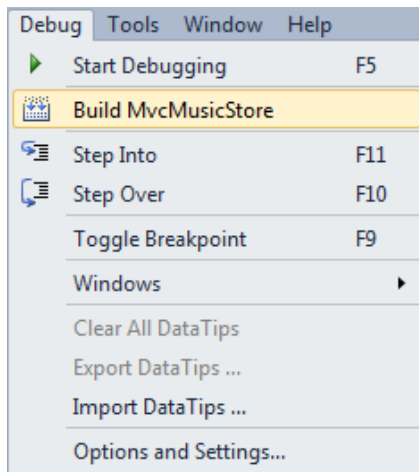
public ActionResult Details(int id)
{
    var album = new Album { Title = "Sample Album" };

    return View(album);
}
```

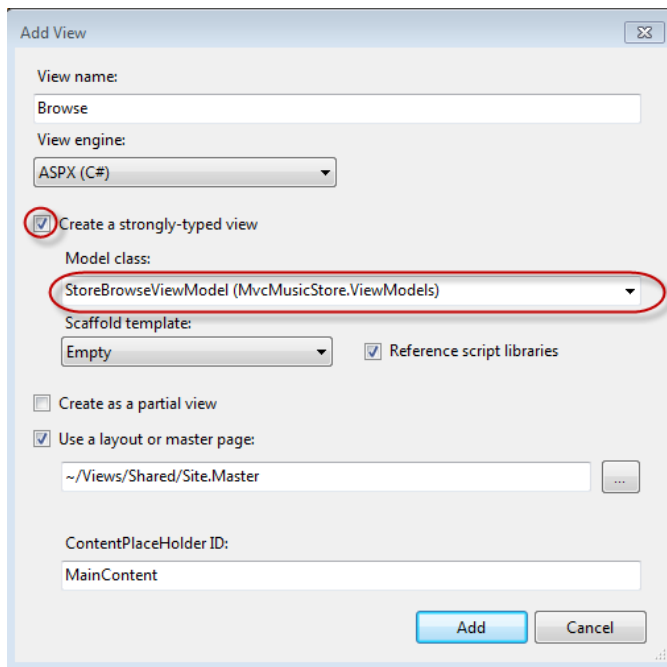
Bước 4 – Thêm một View mới cho phương thức Browse

1. Biên dịch lại chương trình trước khi thêm View mới.





2. Nhấp chuột phải vào phương thức **Browse** trong **StoreController** và nhấn **Add View**.



3. Thay đổi File Browse.aspx lại như sau.

```
<%@ 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>

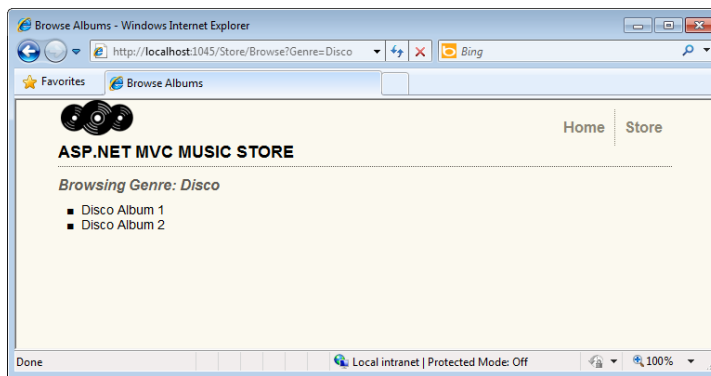
<% foreach (var album in Model.Albums)
{ %>
    <li><%: album.Title %></li>
<% } %>
</ul>

</asp:Content>

```

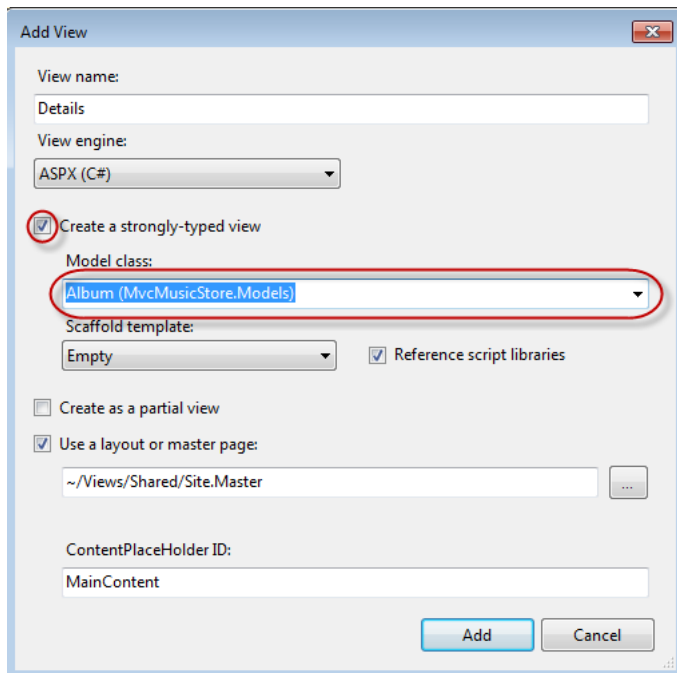
Bước 5 – Chạy chương trình

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



Bước 6 – Trình diễn thông tin về một Album

1. Nhấp chuột phải trong phương thức **Details** trong **StoreController** và nhấp **Add View**



2. Hiệu chỉnh lại lại Details.aspx để hiện ra một Albums.

```

<%@ 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">

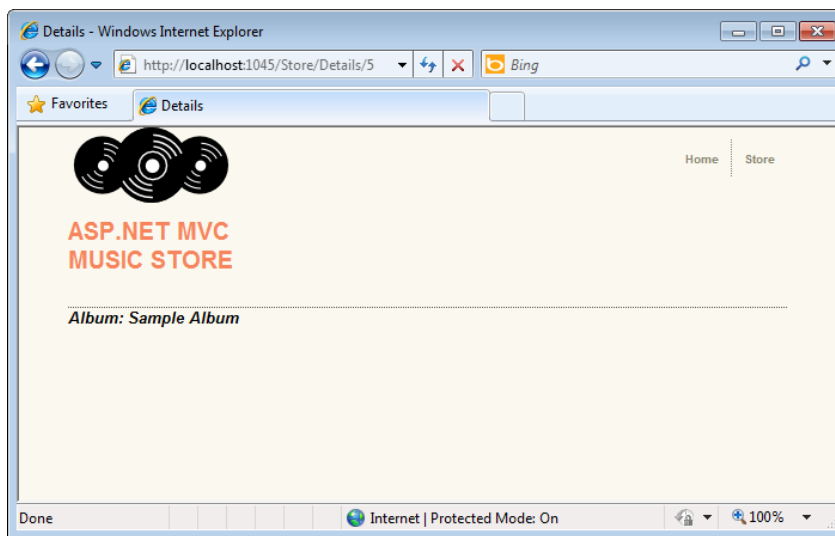
    <h2>Album: <%= Model.Title %></h2>

</asp:Content>

```

Bước 7 – Chạy ứng dụng

1. Nhấn F5 để chạy ứng dụng.
2. Thay đổi URL thành `/Store/Details/5` để hiển thị thông tin Albums có Id là 5.



Bước 8 – Thêm liên kết giữa các trang.

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

```

<%@ 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)
        { %>

```

```

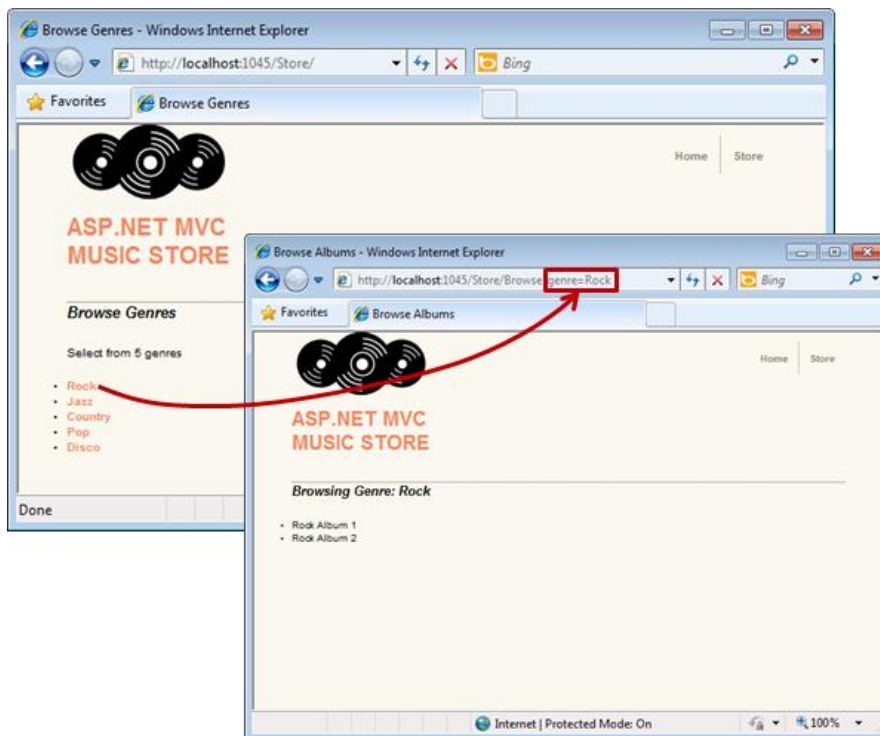
</li>
<%= Html.ActionLink(genreName, "Browse", new { genre = genreName }, null) %>
</li>
<% } %>
</ul>

</asp:Content>

```

Bước 9 – Chạy ứng dụng.

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



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

Trong bước 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.

```

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" };
}

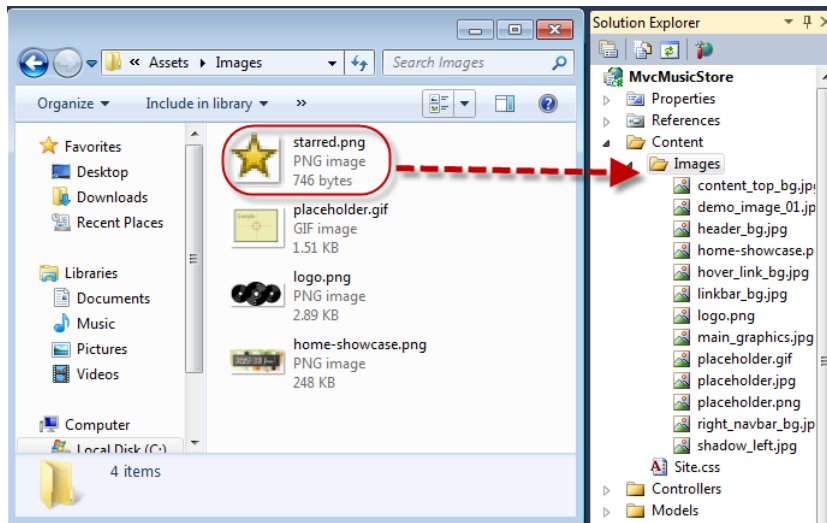
```

```

    return View(viewModel);
}

```

2. Tạo một icon ngôi sao cho Album bằng cách thêm hình ngôi từ máy của bạn vào thư mục Content \Images.



Hình – Thêm một ảnh ngôi sao vào trong Solution Explorer

```

<%@ 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)) { %>
                    
                <% } %>
            </li>
        <% } %>
    </ul>
    <br/>
    <h5>  Starred
    genres 20% Off! </h5>

</asp:Content>

```

Bước 11 – Chạy ứng dụng

