**Tìm hiểu về Layout Page, @RenderBody, @Html.ActionLink và ví dụ minh họa trong ASP.Net**

Dẫn nhập

Ở bài học trước chúng ta đã [tìm hiểu về View, Controller](https://www.howkteam.vn/redirect?Id=Y%2fHKZv6fbvNJnzXtx4Ic0rZRqPOVzQSxG20E6RGIOYEDxYgnx8PSLzpWDOlq39BXsbS6GsKC7kLvdlcpcH6vmU0C66GxrSDZCyqcKeqESvMecwukNHSVkRyGvtfBDet5hZ4eH0mYVOTjS3Nkff%2flhEgKs%2fvcDzc2a9FRGqCPdAZ7ye0RbG7WFFmkn09ZOp1GzbmkSLApdtOuayUNNDtL6c283N5FtuFuq4jR1TnPI5i%2b8S%2fbtxw6RdsNTy%2fVn51s)trong [**ASP.Net MVC**](https://www.howkteam.vn/redirect?Id=Y%2fHKZv6fbvNJnzXtx4Ic0rZRqPOVzQSxG20E6RGIOYEDxYgnx8PSLzpWDOlq39BXsbS6GsKC7kLvdlcpcH6vmZGa3t9pmvMkevYBlkLTQSs%3d) và [cách tạo 1 project ASP.Net MVC](https://www.howkteam.vn/redirect?Id=Y%2fHKZv6fbvNJnzXtx4Ic0rZRqPOVzQSxG20E6RGIOYEDxYgnx8PSLzpWDOlq39BXsbS6GsKC7kLvdlcpcH6vmU0C66GxrSDZCyqcKeqESvMecwukNHSVkRyGvtfBDet5hZ4eH0mYVOTjS3Nkff%2flhEgKs%2fvcDzc2a9FRGqCPdAZ7ye0RbG7WFFmkn09ZOp1GzbmkSLApdtOuayUNNDtL6c283N5FtuFuq4jR1TnPI5i%2b8S%2fbtxw6RdsNTy%2fVn51s) đơn giản với ví dụ là xuất câu “Hello Howkteam”. Tuy nhiên trên thực giao diện của 1 website không đơn giản như vậy.

Giao diện bao gồm rất nhiều thứ như màu sắc, bố cục … vậy trong bài học ngày hôm này mình sẽ hướng dẫn các bạn cách thiết kế giao diện cũng như tìm hiểu về 1 đối tượng rất quan trọng trong View đó là **Layout Page**. Nào! chúng ta cùng đi vào bài học luôn nhé!

Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về các phần:

* Đã đọc qua bài [**Tìm hiểu về View, Controller và cách xuất câu “Hello Howtem” trong ASP.Net MVC**](https://www.howkteam.vn/redirect?Id=Y%2fHKZv6fbvNJnzXtx4Ic0rZRqPOVzQSxG20E6RGIOYEDxYgnx8PSLzpWDOlq39BXsbS6GsKC7kLvdlcpcH6vmU0C66GxrSDZCyqcKeqESvMecwukNHSVkRyGvtfBDet5hZ4eH0mYVOTjS3Nkff%2flhEgKs%2fvcDzc2a9FRGqCPdAZ7ye0RbG7WFFmkn09ZOp1GzbmkSLApdtOuayUNNDtL6c283N5FtuFuq4jR1TnPI5i%2b8S%2fbtxw6RdsNTy%2fVn51s)**.**
* Kiến thức về HTML, CSS là 1 lợi thế trong việc xây dựng giao diện.

Trong bài học này, chúng ta sẽ cùng tìm hiểu các vấn đề:

* Tìm hiều về Layout Page, @RenderBody(), @Html.ActionLink()
* Ví dụ demo.

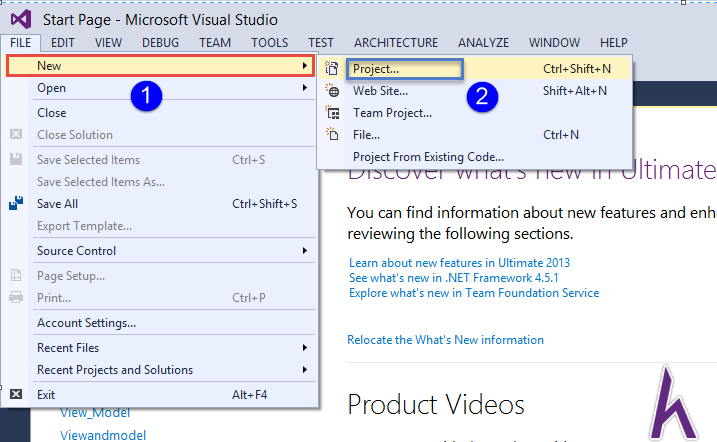
Tìm hiểu về Layout Page

Trong một website các bạn sẽ thấy một page lúc nào cũng có đủ các thành phần bao gồm: header, footer, body ... Nếu các bạn để ý sẽ thấy phần header và phần footer của các page thuộc cùng website đều giống nhau.

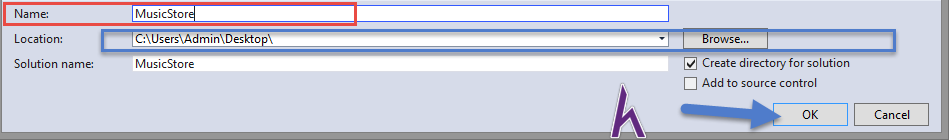
Tuy nhiên theo cách thiết kế thông thường, cứ mỗi khi tạo ra một page mới chúng ta phải tạo lại phần header, footer. Mà 2 phần này thì nội dung giống nhau ở tất cả các page, do đó sẽ làm mất rất nhiều thời gian cho người thiết kế.

Vì vậy để giải quyết tình trạng trên [**ASP.Net MVC**](https://www.howkteam.vn/redirect?Id=Y%2fHKZv6fbvNJnzXtx4Ic0rZRqPOVzQSxG20E6RGIOYEDxYgnx8PSLzpWDOlq39BXsbS6GsKC7kLvdlcpcH6vmZGa3t9pmvMkevYBlkLTQSs%3d) đã cho ra đời 1 khái niệm là **Layout Page**, với việc sử dụng Layout Page chúng ta chỉ cần tạo nội dung cho header và footer một lần, nếu một page bất kì muốn sử dụng thì chỉ cần thừa kế từ Layout Page chứa header và footer. Để hiểu rõ hơn thì chúng ta cùng đi qua ví dụ sau nhé!

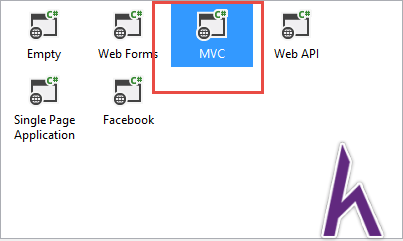
Đầu tiên các bạn tạo 1 project **ASP.Net MVC** có tên là MusicStore.



* Các bạn chọn File > New > Project

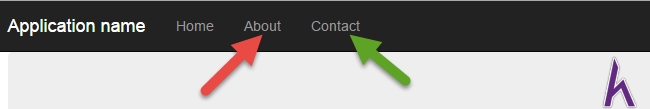


* Tại khung Name để tên project là MusicStore, sau đó chọn **Browse..**đề chọn nơi lưu trữ project sau đó các bạn nhấn **OK**.
* Sau thao tác trên, giao diện dưới hiện ra các bạn chọn MVC rồi nhấn **OK**tiếp là xong bước tạo project.

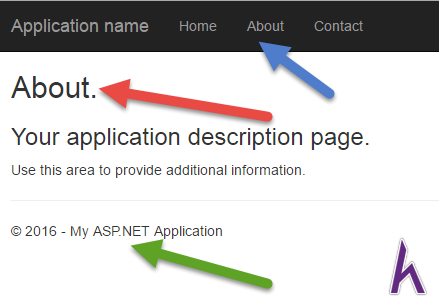


Chúng ta cùng xem thừa kế lại giao diện phần header và footer là sao nhé!

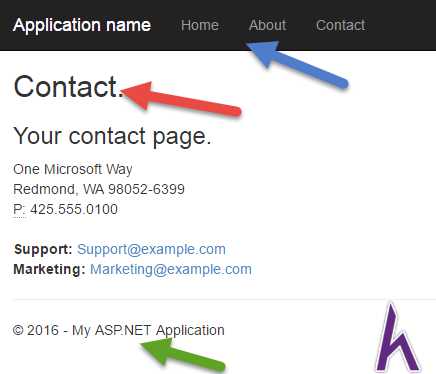
* Sau khi tạo project thành công, các bạn chạy Project lên, sau đó các bạn lần lượt truy cập vào 2 trang là About và Contact



* Các bạn chú ý sẽ thấy 2 trang này có phần footer và header giống nhau

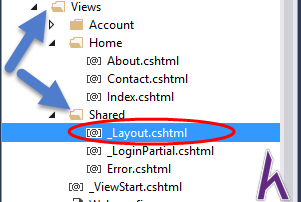


**Giao diện trang About**



**Giao diện trang Contact**

* Vì vậy, nếu mà khi tạo trang About hay trang Contact chúng ta tạo lại 2 phần header và footer thì sẽ mất thời gian, vì vậy Layout Page đã giúp chúng ta làm điều đó. Để hiểu rõ cơ chế sử dụng lại như thế nào thì các bạn mở folder View > Shared rồi mở nội dung file \_Layout.cshtml lên.



* Phần nội dung nằm trong thẻ sau chính là giao diện của phần header

<div class="navbar navbar-inverse navbar-fixed-top">…</div>



* Còn phần nội dung nằm trong thẻ <footer></footer> chính là nội dung của phần footer

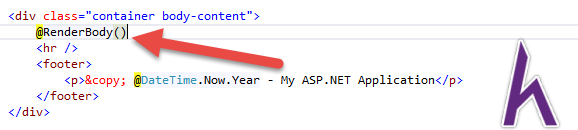
kteam

Vậy hai trang About và Contact là hai page khác biệt sẽ sử dụng chung hai phần footer và header trong \_Layout.cshtml. Khi các page được tạo ra mà có sử dụng footer và header giống nhau thì \_Layout.cshtml sẽ cung cấp cho các page đó.

Cách làm như thế nào thì tí nữa mình sẽ đi vào ví dụ cụ thể nhé!

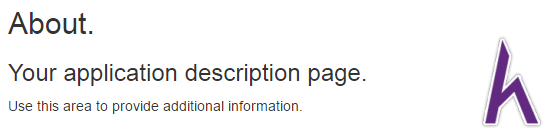
Tìm hiểu @RenderBody()

 Trong \_Layout.cshtml các bạn sẽ thấy 1 thuộc tính đó là **@RenderBody().**

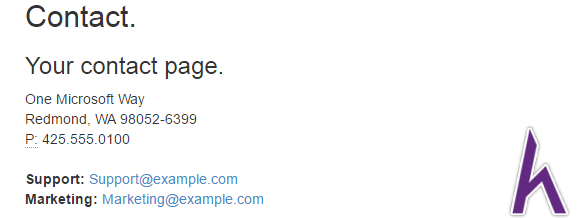


Tuy hai trang About và Contact có header và footer giống nhau, nhưng nội dung của hai trang này lại hoàn toàn khác nhau, vì vậy Layout Page cung cấp cho chúng ta 1 thuộc tính **@RenderBody(),**

Với thuộc tính này khi các bạn muốn hiển thị View nào lên thì **@RenderBody()**sẽ trả về nội dung đúng với View đó, tức là chúng ta muốn hiển thị trang About lên thì **@RenderBody()**sẽ trả về nội dung đúng với trang About và tương tự đối với trang Contact



**Nội dung trang About**



**Nội dung trang Contact**

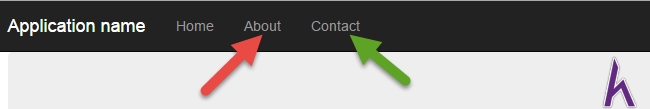
Qua hai kết quả trên thì **@RenderBody()** đã giúp cho hai trang About và Contact trả về nội dung mong muốn của các page.

Tìm hiểu về @Html.ActionLink

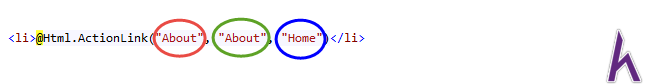
Các bạn thấy trong \_Layout.cshtml có 1 phương thức **@Html.ActionLink**, phương thức này là một trong các phương thức thuộc HTML Helper trong [ASP.Net MVC](https://www.howkteam.vn/redirect?Id=Y%2fHKZv6fbvNJnzXtx4Ic0rZRqPOVzQSxG20E6RGIOYEDxYgnx8PSLzpWDOlq39BXsbS6GsKC7kLvdlcpcH6vmZGa3t9pmvMkevYBlkLTQSs%3d).

Trong phạm vi bài này mình chỉ giới thiệu **@Html.ActionLink** còn các phương thức còn lại trong HTML Helper thì khi nào gặp cái nào, mình sẽ giới thiệu cái đó.

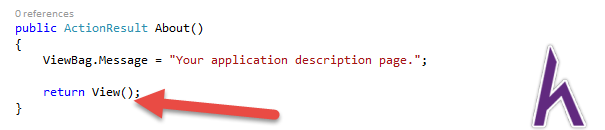
**@Html.ActionLink** là cách chúng ta tạo kết tới action trong 1 controller bất kì và thực thi action đó. Cụ thể như sau:



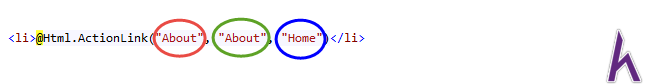
* Khi các bạn click vào About thì kết quả trả về là 1 trang hiện thị nội dung của Page About. Vậy có bạn nào đang thắc mắc là tại sao khi click vào chữ About nó lại chuyển qua trang About không? Sau đây mình sẽ giải thích lý do nhé.



* Khi các bạn click vào chữ About, thì action About của controller Home sẽ được thực thi thông qua phương thức **@Html.ActionLink**. Mà action About trong controller Home lại trả về View() About.



* Qua đó chúng ta đã có thể trả lời được lý do tại sao khi click vào About là sẽ có 1 giao diện trang About hiển thị lên.
* Khi chúng ta thực thi phương thức **@Html.ActionLink** thì các bạn lưu ý phải truyền đủ 3 tham số cho phương thức nhé.



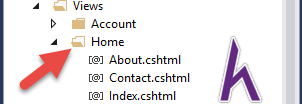
Mình sẽ lấy ví dụ của About:

* Tham số thứ nhất: About chính là tên hiển thị trên page
* Tham số thứ 2: About chính là tên của action mà mình muốn thực thi
* Tham số thứ 3: Home chình là controller chứa action ở mà tham số thứ 2 chúng ta khai báo

Ví dụ demo

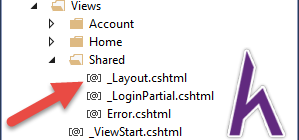
Phần đầu mình đã cùng các bạn tìm hiều về Layout Page, trong nội dung này chúng ta sẽ cùng nhau đi làm ví dụ về Layout Page nhé! Qua ví dụ này, các bạn sẽ sẽ hiểu rõ cách áp dụng Layout Page vào cho từng View hiển thị, biết cách nhúng CSS bên ngoài vào trong giao diện.

Ở phần trên chúng ta đã tạo thành công 1 project có có tên là MusicStore. Để tạo demo cho riêng mình thì các bạn nên xóa 1 vài thư mục có sẵn trong Project.



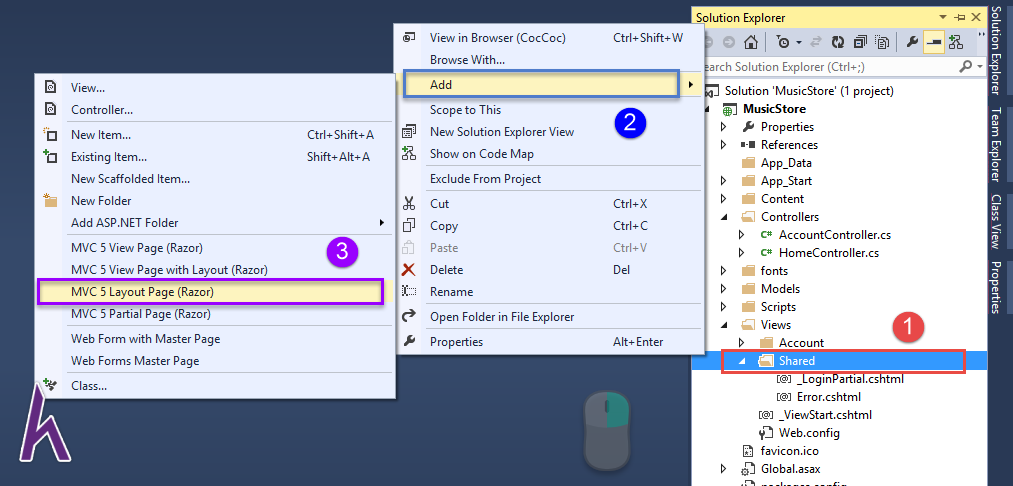
* Đầu tiên, các bạn xóa cho mình folder Home trong Views

Sau đó các bạn xóa tiếp cho mình folder \_Layout.cshtml trong thư mục Shared

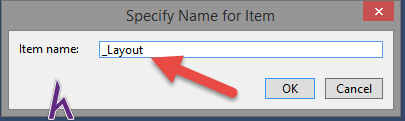


Lý do mà mình nói các bạn xóa đi là vì chúng ta sẽ tạo ra Layout Page theo ý của chúng ta, kể cả các trang thừa kề từ Layout Page thì mình cũng muốn làm khác đi theo cách riêng của mình.

Để tạo mới 1 Layout Page thì chúng ta vào folder Shared trong phần Views, sau đó các bạn làm theo như mô tả hình dưới đây.



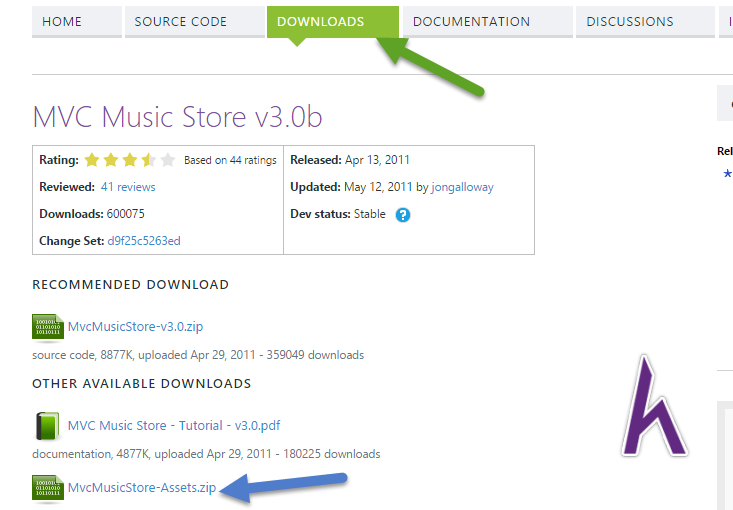
* Các bạn click phải chuột vào folder Shared > Add > MVC 5 Layout Page (Razor)



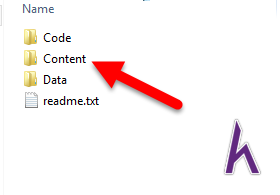
* Kết thúc bước chọn MVC 5 Layout Page (Razor) sẽ có 1 giao diện hiển thị như hình trên, ngay tại ô Item name các bạn thay đổi tên thành \_Layout nhé rồi nhấn **OK**để kết thúc.

Sau khi tạo xong được Layout Page thì chúng ta chưa tạo giao diện cho Layout Page vội, mà các bạn cần chuẩn bị cho mình 1 số file cần thiết như sau:

* Các bạn truy cập vào website [**http://mvcmusicstore.codeplex.com**](https://www.howkteam.vn/redirect?Id=uNmHDvPjq6kfn9XuwT%2fyIi8er%2fFak8Gt5m3bqvDsiDj4Os4m9JfYvy5b7lSjEIcW) > **Dowload** > tải file có tên [**MvcMusicStore-Assets.zip**](https://www.howkteam.vn/redirect?Id=uNmHDvPjq6kfn9XuwT%2fyIi8er%2fFak8Gt5m3bqvDsiDhWp%2bgay2AhQn23ucXZtEdartTv4VdsnTdAnMkYnTXg2w%3d%3d)

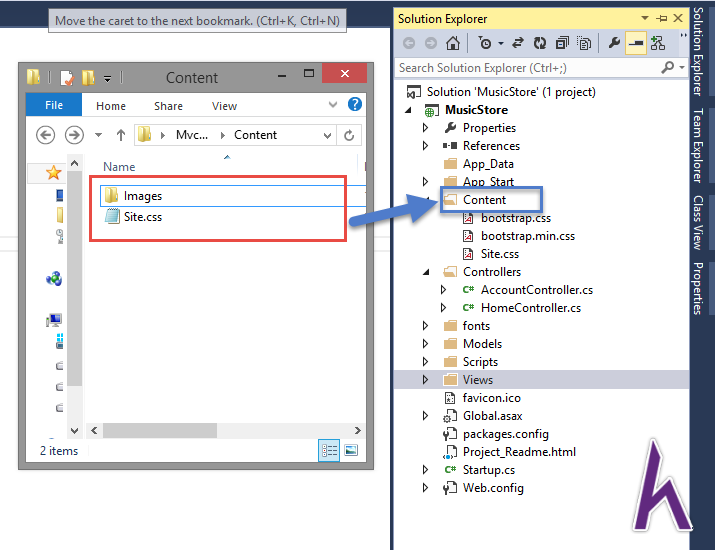


Download file [**MvcMusicStore-Assets.zip**](https://www.howkteam.vn/redirect?Id=uNmHDvPjq6kfn9XuwT%2fyIi8er%2fFak8Gt5m3bqvDsiDhWp%2bgay2AhQn23ucXZtEdartTv4VdsnTdAnMkYnTXg2w%3d%3d) thành công, các bạn giải nén file đó ra. Trong file giải nén các bạn sẽ thấy có 3 folder và 1 file txt thì các bạn truy cập vào folder có tên là **Content** cho mình nhé.

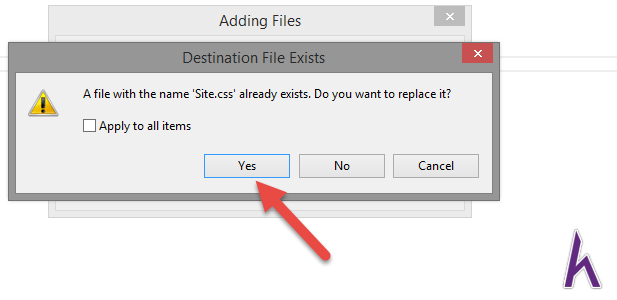


Lý do mà mình nói các bạn download file nén này về là vì chúng ta sẽ không sử dụng giao diện mặc định của ASP.Net MVC nữa mà chúng ta sẽ sử dụng 1 giao diện có sẵn khác với CSS riêng.

Sau khi truy cập vào folder Content các bạn copy tất các **thư mục** trong đó và bỏ vào thư mục **Content** trong project MusicStore của mình nhé.



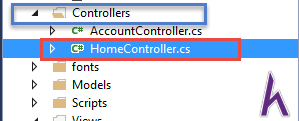
* Khi bạn **Paste** vào thư mục **Content** trong project thì sẽ có 1 bảng hội thoại xuất hiện lên như sau:



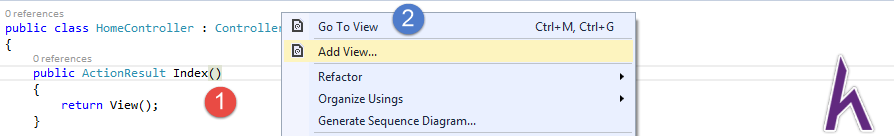
* Thông báo của bảng hội thoại này là bạn có muốn ghi đè file có tên **Site.css** đã có sẵn trong thực mục Content không? Thì các bạn chọn **Yes** cho mình nhé.

Vậy là các bạn đã chuẩn bị xong các tài nguyên cần thiết, bây giờ chúng ta bắt đầu vào sử dụng nó nhé.

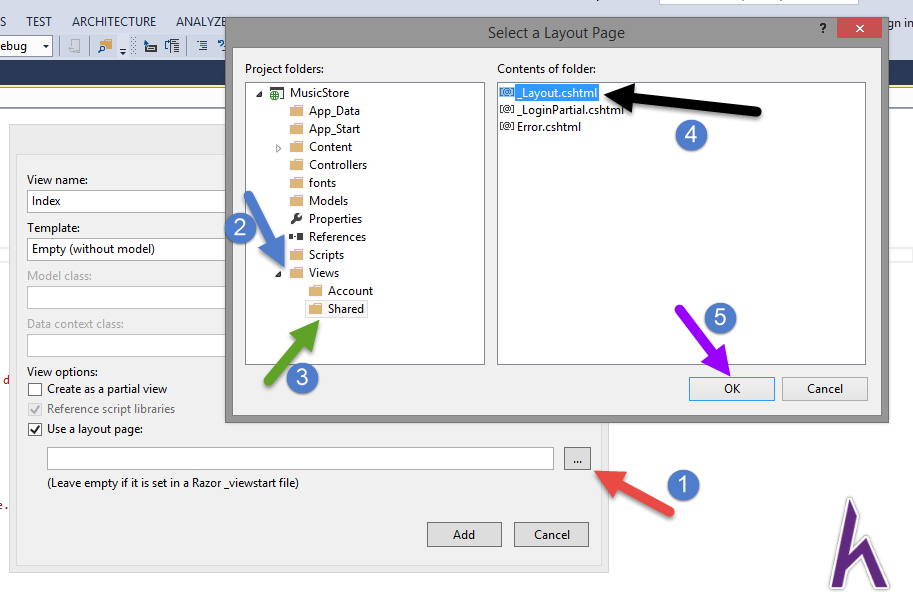
* Trước tiên các bạn cần tạo lại cho mình tầng View cho Action Index thuộc Controller **Home**



* Sau đó các bạn nhấp phải chuột tại **Index**> chọn **Add View**



* Các bạn làm theo thứ tự mà mình đánh số nhé



Sau khi nhấn **OK** thì các bạn chọn **Add**. Tại trang Index vừa tạo, các bạn chuyển chữ **Index** trong thẻ <h2> thành **This is my home page**  nhé!



* Vậy là các bạn đã tạo thành công View cho Action Index rồi, bây giờ các bạn quay trở lại thư mục **\_Layout.cshtml** và chỉnh sửa 1 vài cái nhé.

<!DOCTYPE html>

<**html**>

<**head**>

<**meta** name="viewport" content="width=device-width" />

<**title**>@ViewBag.Title</**title**>

<**link** href="/redirect?Id=LnRQ7pJ8KNXxXrMnBPvG1g%3d%3d"~/Content/Site.css")" rel="stylesheet" type="text/css" />

<**script** src="@Url.Content("~/Scripts/jquery-1.5.1.min.js")"

type="text/javascript"></**script**>

<**script** src="@Url.Content("~/Scripts/modernizr-1.7.min.js")"

type="text/javascript"></**script**>

</**head**>

<**body**>

<**div**>

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

<**h1**>

ASP.NET MVC MUSIC STORE

</**h1**>

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

<**li** class="first"><**a** href="/redirect?Id=Q%2fqjAzdLo8VFFKeFdWbiAA%3d%3d" id="current">Home</**a**></**li**>

<**li**><**a** href="/redirect?Id=0htwlDSmnekWTM6XP0g0Qg%3d%3d">Store</**a**></**li**>

</**ul**>

</**div**>

@RenderBody()

</**div**>

</**body**>

</**html**>

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

Trong đoạn code trên, các bạn copy những phần sau vào trong bài của các bạn. Phần này nằm trong thẻ <head></head>.

    <link href="@Url.Content("~/Content/Site.css")" rel="stylesheet" type="text/css" />

    <script src="@Url.Content("~/Scripts/jquery-1.5.1.min.js")"

            type="text/javascript"></script>

    <script src="@Url.Content("~/Scripts/modernizr-1.7.min.js")"

           type="text/javascript"></script>

        <div id="header">

            <h1>

                ASP.NET MVC MUSIC STORE

            </h1>

            <ul id="navlist">

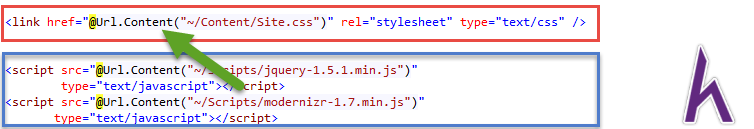
                <li class="first"><a href="/" id="current">Home</a></li>

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

            </ul>

        </div>

Từ khóa **@Url.Content** dùng để trỏ tới các thư mục css hoặc js hay bất kì file nào mà chúng ta muốn sử dụng. Phía trên mình có nói các bạn copy file Images và Site.css vào trong thư mục Content, để có thể sử dụng Site.css bạn chỉ cần viết theo cú pháp trong khung đỏ như hình dưới.  
Cũng như vậy, khung màu xanh các bạn cũng làm tương tự để chúng ta có thể sử dụng các file js (chú ý: file js ở đây là được cung cấp sẵn).

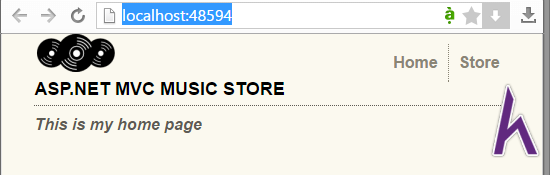


* File css: giúp chúng ta phối màu, tạo hoa văn cũng như làm cho website thêm sinh động.
* File js: giúp chúng ta tạo ra các hiệu cho website như tạo slider … nói chung các hiệu ứng động trong website.

Cuối cùng các bạn **chạy project** mình lên để kiểm tra sự thay đổi nhé

kteam

Sau khi chạy project, nếu bạn nào có màn hình chạy lên giống màn hình sau thì coi như các bạn đã hoàn thành bài hướng dẫn.



Nhìn vào kết quả đạt được, các bạn thấy sự khác biệt giữa file mặc định và file mà chúng ta tạo ra rồi nhé.

Kết luận

Qua bài học ngày hôm này, mình đã giúp các bạn hiểu rõ về LayoutPage, RenderBody, Html.ActionLink cũng như cách các bạn sử dụng giao diện bên ngoài.

Bài sau chúng ta sẽ đi vào **Tìm hiểu về PartialView, Model trong MVC, tham số và cách truyền tham số tới View sử dụng Model**

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên “**Luyện tập – Thử thách – Không ngại khó**”.