**HƯỚNG DẪN LẬP TRÌNH WEB VỚI ASP.NET MVC**

**( Phần 2 - Tạo trang chủ với ASP.Net MVC)**

1. Nhiệm vụ

Trong phần trước chúng ta đã làm quen với ASP.NET MVC và tạo ra một trang đăng nhập hệ thống có kết nối với CSDL cơ bản rồi. Trong chương này chúng ta đi sâu hơn vào việc xây dựng một trang Home với một layout dựng sẵn, đỗ dữ liệu từ Model ra View. Để thực hiện các nội dung này yêu cầu các bạn cần nhớ lại cách tạo giao diện với HTML, CSS và cách lấy dữ liệu từ ADO Entity Model và Linq.

1. Xây dựng master page

Master page là gì? Đó là thành phần chung của các giao diện. Chúng được lặp lại trong các trang khác nhau. Chúng định layout cho một Website.

Ví dụ:



Phần tô viền màu đỏ được gọi là Master page phần màu xanh phía dưới là nội dung trang và nó thay đổi cho từng trang khác nhau duy chỉ có phần đỏ là không đổi cho các trang.

Trong phần này chúng ta tiến hành xây dựng master page cho trang chủ. Bản chất việc dựng master page là viết mã HTML sau cho trong mã đó phân định ra các vùng rõ ràng trên giao diện ( định layout) trong đó có một vùng hiển thị nội dung các trang kế thừa master page. Để dựng Master page, trong thư lục Views bạn tạo mới thư mục Shares. Sau đó chuột phải vào thư mục Share chọn Add, MVC Layout page with razor và đặt tên là masterpage.

Bây giờ tôi muốn có một Dòng chữ Luxury Watches trên nền trắng năm ở giửa phía trên và phía bên dưới là một Menu màu đen chữ trắng. Tôi tiến hành làm như sau.

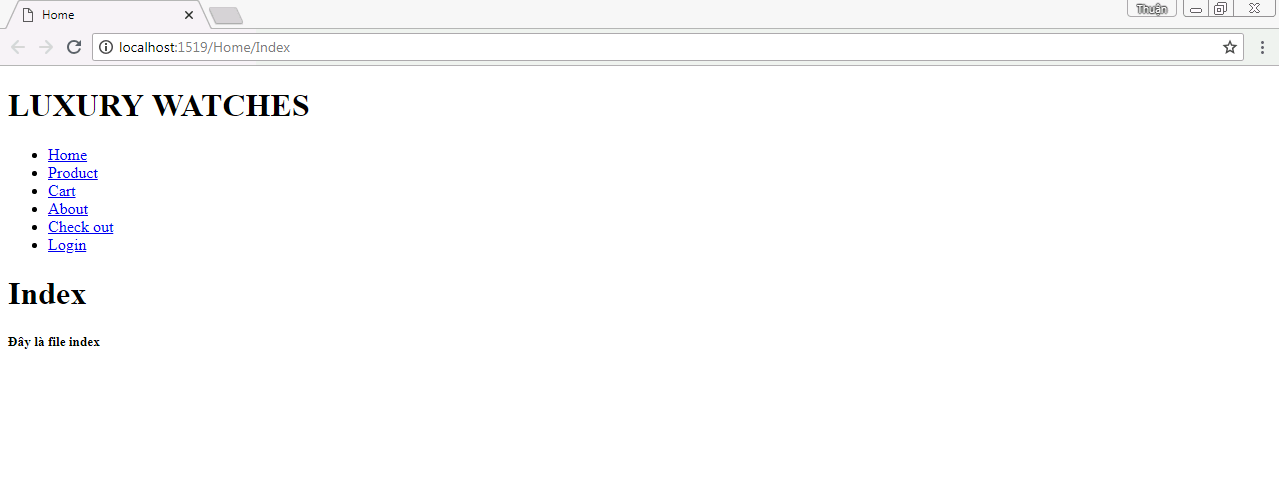
Bước 1: Chỉnh sửa trang masterpage.cshtml như sau:

|  |
| --- |
| HTML |
| <!DOCTYPE html>  <html>  <head>  <meta name="viewport" content="width=device-width" />  <title>@ViewBag.Title</title>  </head>  <body>  <div class="head">  <h1>LUXURY WATCHES</h1>  </div>  <div class="nav">  <ul class="menu">  <li class="menu-item"><a href="#">Home</a></li>  <li class="menu-item"><a href="#">Product</a></li>  <li class="menu-item"><a href="#">Cart</a></li>  <li class="menu-item"><a href="#">About</a></li>  <li class="menu-item"><a href="#">Check out</a></li>  <li class="menu-item"><a href="#">Login</a></li>  </ul>  </div>  <div>  @RenderBody()  </div>  </body>  </html> |

Bước 2: Thay đổi file Index.cshtml của Home như sau:

|  |
| --- |
| HTML |
| @{  ViewBag.Title = "Home";  Layout = "~/Views/Shares/masterpage.cshtml";  }  <div>  <h1>Index</h1>  <h5>Đây là file index</h5>  </div> |

Giờ hãy chạy trang Home lên và xem kết quả.



Chính dòng code thứ 3 trong Index đã định nghĩa rằng trang này sẽ kế thừa trang masterpage và nội dung của nó sẽ được vẽ bởi hàm RenderBody trong masterpage.

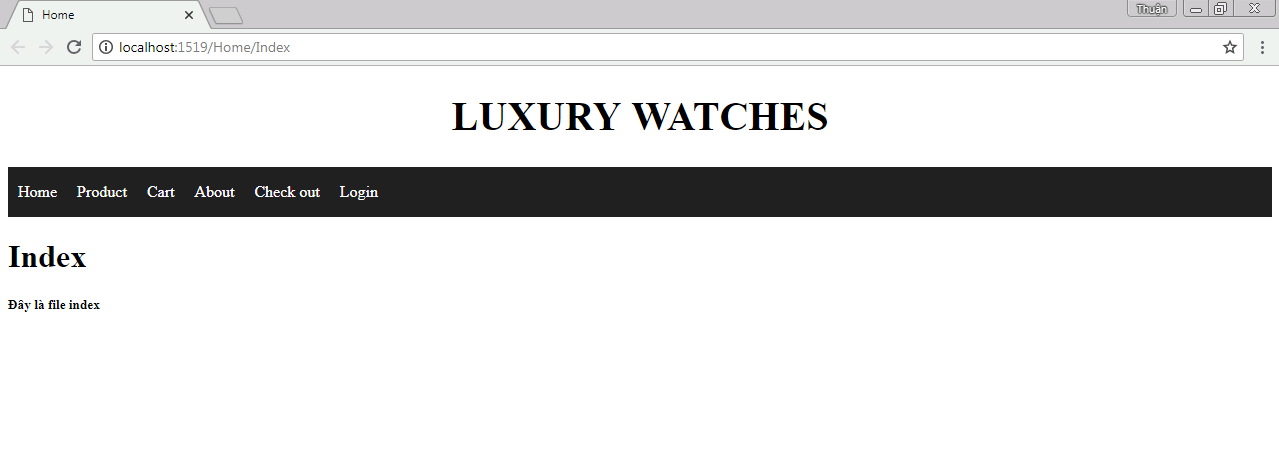
Tuy đã nhúng được trang Home vào masterpage rồi nhưng chúng ta chưa đạt được yêu cầu về mặt giao diện. Để đạt được yêu cầu ta tiến hành thêm file CSS có tên masterpage.css với nội dung như sau:

|  |
| --- |
| CSS |
| .head{  width:100%;  }  .head h1{  font-size:40px;  text-align:center;  }  .nav{  background:#222;  }  .nav .menu{  list-style:none;  display:inline-flex;  padding:0px;  align-items:center;  }  .nav .menu .menu-item a{  color:#fff;  text-decoration:none;  padding:5px 10px;  }  .nav .menu .menu-item a:hover{  color:#ffd800;  text-decoration:underline;  } |

Không quên nhúng mã css này vào trang masterpage với đoạn mã:

<link href="~/Contents/masterpage.css" type="text/css" rel="stylesheet"/>

Giờ thử chạy lại ứng dụng và ta được kết quả:



Không quá đẹp nhưng có thể chấp nhận được rồi. Bạn có thể thêm phần màu sắc hơn cho masterpage tùy ý bạn.

* Trong phần này bạn đã biết cách tạo ra một master page, kế thừa nó cho các view. Trong phần tiếp theo sẽ quan trọng hơn khi ta tiến hành đỗ dữ liệu từ CSDL lên View. Một điều cần nhắc lại là một Website có thể có nhiều masterpage hoặc không cần bất cứ một master page nào.

1. Xây dựng Model cho trang chủ

Việc đầu tiên là cần xác định trang xây dựng sẽ chứa những thông tin gì. Và trang chủ website này bao gồm hai danh sách. Tạm thời tôi muốn một danh sách phía trên bao gồm 4 sản phẩm mới nhất ( theo Id) và một danh sách phía dưới gồm 4 sản phẩm rẽ nhất. Mỗi sản phẩm bao gồm hình ảnh, Tên sản phẩm, giá tiền. Vì vậy Trước hết cần tạo HomeViewModel với hai danh sách SANPHAM một là NewProducts, một là SaveProducts.

|  |
| --- |
| C# |
| using WebsiteBanDongHo.Models.Context;  namespace WebsiteBanDongHo.Models.ViewModel  {  public class HomeViewModel  {  public List<SANPHAM> NewProducts { get; set; }  public List<SANPHAM> SaveProducts { get; set; }  }  } |

Tiếp theo chúng ta cũng tạo một Service để lấy dữ liệu về.

|  |
| --- |
| C# |
| using WebsiteBanDongHo.Models.Context;  namespace WebsiteBanDongHo.Models.Service  {  interface HomeService  {  List<SANPHAM> GetNewProducts(int number);  List<SANPHAM> GetSaveProducts(int number);  }  } |

Cuối cùng tạo một Class implement HomeService.

|  |
| --- |
| C# |
| using System.Collections.Generic;  using System.Linq;  using WebsiteBanDongHo.Models.Context;  using WebsiteBanDongHo.Models.Service;  namespace WebsiteBanDongHo.Models.Implement  {  public class HomeImplement : HomeService  {  // Phương thức Lấy về danh sách sản phẩm mới nhất  // @number là số lượng sản phẩm lấy về  public List<SANPHAM> GetNewProducts(int number)  {  List<SANPHAM> Result = new List<SANPHAM>();  // Mã Linq lấy về danh sách sản phẩm sắp xếp theo Mã  IEnumerable<SANPHAM> sanpham = null;  using (var db = new BANDONGHOEntities())  {  sanpham = from sp in db.SANPHAMs  orderby sp.MASP descending  select sp;  // Cắt ra @number phần tử  if (sanpham != null)  {  Result = sanpham.Take(number).ToList();  }  }    return Result;    }  // Phương thức Lấy về danh sách sản phẩm rẻ nhất  // @number là số lượng sản phẩm lấy về  public List<SANPHAM> GetSaveProducts(int number)  {  List<SANPHAM> Result = new List<SANPHAM>();  // Mã Linq lấy về danh sách sản phẩm sắp xếp theo Giá  IEnumerable<SANPHAM> sanpham = null;  using (var db = new BANDONGHOEntities())  {  sanpham = (from sp in db.SANPHAMs  orderby sp.DONGIA descending  select sp);  // Cắt ra @number phần tử  if (sanpham != null)  {  Result = sanpham.Take(number).ToList();  }  }    return Result;  }  }  } |

Mọi thứ ở Model đã chuẩn bị đầy đủ giờ ta tiếp tục việc gọi service trên Home Controller.

1. Xây dựng HomeController

Nhiệm vụ quan trọng của HomeController là lấy dữ liệu từ Model gửi lên View thông qua ViewModel.

|  |
| --- |
| C# |
| using System.Web.Mvc;  using WebsiteBanDongHo.Models.Implement;  using WebsiteBanDongHo.Models.Service;  using WebsiteBanDongHo.Models.ViewModel;  namespace WebsiteBanDongHo.Controllers  {  public class HomeController : Controller  {  // Số lượng sản phẩm mới lấy ra  const int NUMBER\_NEW\_PRODUCTS = 4;  // Số lượng sản phẩm rẽ lấy ra  const int NUMBER\_SAVE\_PRODUCTS = 4;  // Khai báo và khởi tạo ViewModel  HomeViewModel ViewModel = new HomeViewModel();  HomeService Service;  // GET: Home  public ActionResult Index()  {  // Khởi tạo service  Service = new HomeImplement();  // Gán giá trị ViewModel  ViewModel.NewProducts = Service.GetNewProducts(NUMBER\_NEW\_PRODUCTS);  ViewModel.SaveProducts = Service.GetSaveProducts(NUMBER\_SAVE\_PRODUCTS);  // Gửi ViewModel ra View  return View(ViewModel);  }  }  } |

1. Xây dựng trang Home View.

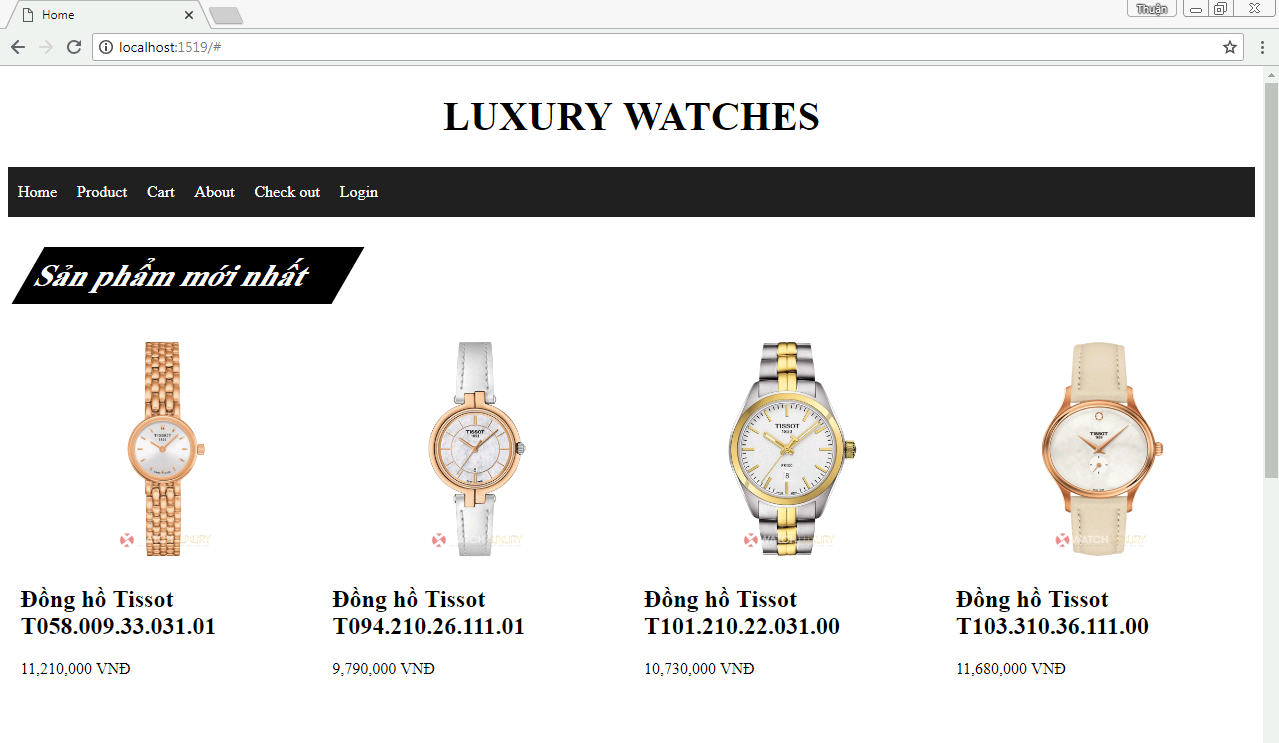
Dữ liệu đã được chuyển sang View rồi giờ làm cách nào để View đón nhận được và hiển thị chúng ra màn hình. ASP.Net sử dụng một đối tượng là Model để lấy ra các dữ liệu được gửi từ Controller. Cách dùng như sau:

|  |
| --- |
| HTML |
| @model WebsiteBanDongHo.Models.ViewModel.HomeViewModel  @{  ViewBag.Title = "Home";  Layout = "~/Views/Shares/masterpage.cshtml";  }  <div>  <div class="container">  <h1>Sản phẩm mới nhất</h1>  <div>  @foreach (var item in Model.NewProducts)  {  <div class="item">  <a href="#">  <div class="thumbnail">  <img src="@Url.Content("~/Contents/images/HINHLON/" + item.HINHLON)" alt="" />  </div>  </a>  <div class="name-product"><h2>@item.TENSP</h2></div>  <div><p>@String.Format("{0:0,0}", item.DONGIA) VNĐ</p></div>  </div>  }  <div style="clear:both"></div>  </div>  </div>  <div class="container">  <h1>Sản phẩm giá rẻ</h1>  <div>  @foreach (var item in Model.SaveProducts)  {  <div class="item">  <a href="#">  <div class="thumbnail">  <img src="@Url.Content("~/Contents/images/HINHLON/" + item.HINHLON)" alt="" />  </div>  </a>  <div class="name-product"><h2>@item.TENSP</h2></div>  <div><p>@String.Format("{0:0,0}", item.DONGIA) VNĐ</p></div>  </div>  }  <div style="clear:both"></div>  </div>  </div>  </div> |

Hãy chú ý những vị trí có “@” đó là những nơi ta chèn mã C# vào mã HTML. Đối tượng Model được khai báo qua câu lệnh @model WebsiteBanDongHo.Models.ViewModel.HomeViewModel. Vì vậy khi bạn bật chức năng nhắc nhở code ( Ctr + Space) bạn sẽ thấy các thuộc tính của ViewModel xuất hiện. Lệnh @Url.Content("~/Contents/images/HINHLON/" + item.HINHLON) giúp bạn lấy được đường dẫn đến hình ảnh. Trong dự án này tôi lưu thư mục hình ảnh trong thư mục Contents. Thư mục images sẽ được cung cấp trong dự án trên github. Giờ thử chạy mã bạn sẽ thấy Trang Home đã load được dữ liệu lên. Tuy chưa được đẹp nhưng đó là tất cả những gì tôi cần. Giờ ta có thể dùng CSS để làm cho trang Home đẹp hơn đôi chút. Mã CSS như sau:

|  |
| --- |
| CSS |
| .container{  padding: 10px 0;  }  .container h1{  padding: 10px;  color:#fff;  background:#000;  width:300px;  margin:20px;  transform:skew(-30deg,0deg);    }  .item{  width:23%;  float:left;  padding:1%;  }  .item .thumbnail{  width:100%;  }  .item .thumbnail img{  width:100%;  } |

Kết quả:



1. Tổng kết

Trong phần này tôi đã đi sâu vào cách tạo ra một layout cho một trang web, lấy dữ liệu lên từ Model, gửi dữ liệu ra View. Trong phần tiếp theo sẽ là các thành phần qua trọng trong một Website như Session hay Ajax.