



Bài 07

Layout và Responsive Pages



- ◆ Consistent Look and Feel Using Layouts
- ◆ Tập tin _Layout.cshtml
- ◆ Tạo Custom Layout
- ◆ Xác định Layout cho View
- ◆ Sử dụng Style
- ◆ Sử dụng JavaScript
- ◆ Sử dụng JQuery
- ◆ Sử dụng Ajax Helper



- ◆ ASP.NET MVC Framework cho phép bạn sử dụng layouts để đơn giản hóa việc tạo ra consistent look and feel trong ứng dụng.
- ◆ Bạn có thể sử dụng Layout mà trong đó chỉ ra một template để sử dụng trong tất cả các views của ứng dụng, trong template này bạn có thể định nghĩa cấu trúc và các style chung cho các view tạo nên sự đồng nhất ở tất cả các view.



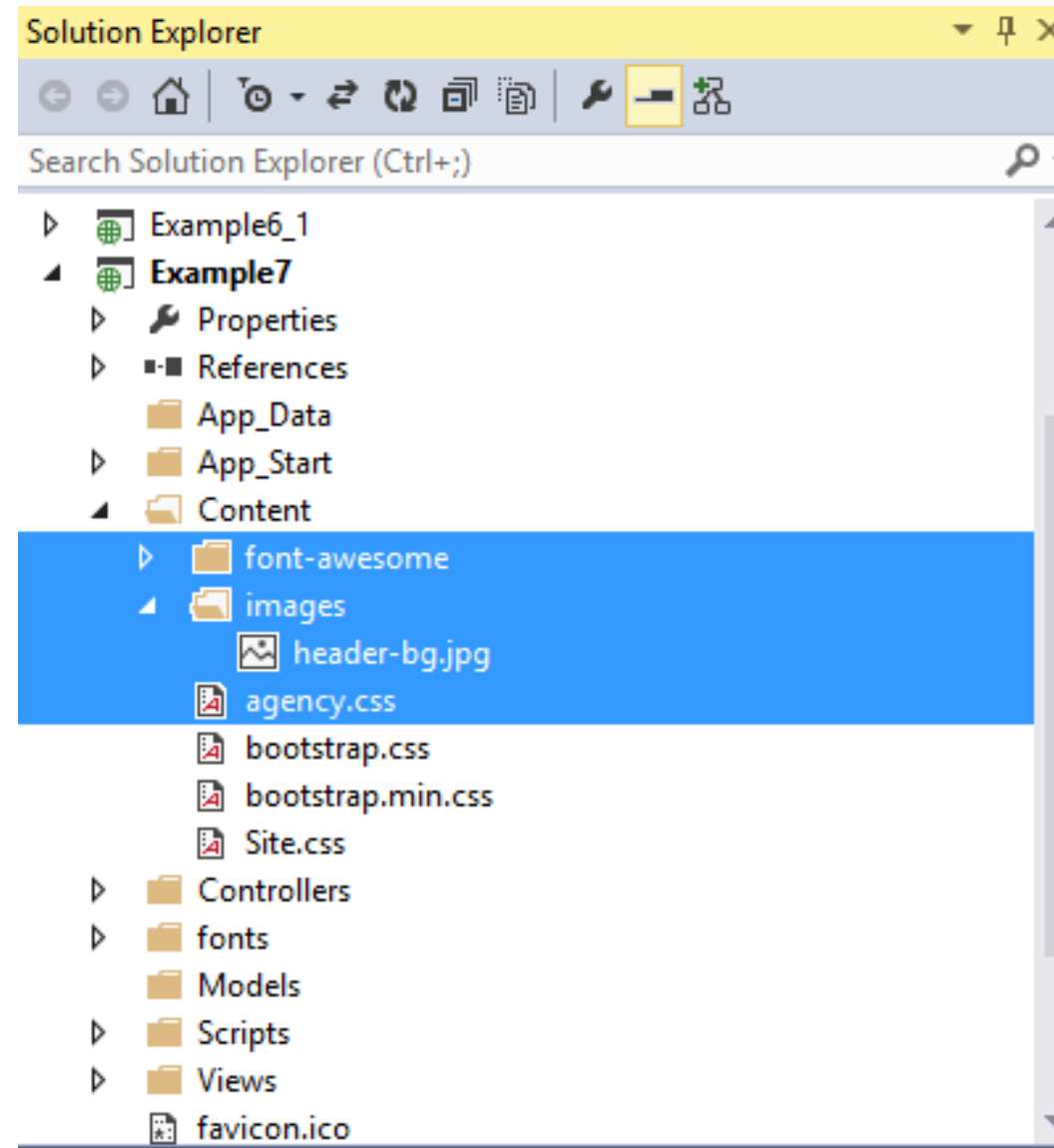
- ◆ Khi bạn tạo một ứng dụng Web ASP.NET MVC trong Visual Studio 2017, tệp tin _Layout.cshtml sẽ tự động thêm vào thư mục View/Shared.
- ◆ Tệp này trình bày layout chung sẽ được áp dụng cho các view trong ứng dụng.
- ◆ Nó chứa cấu trúc cơ bản của một view, trong quá trình phát triển ứng dụng bạn có thể thay đổi cấu trúc tệp này cho phù hợp với yêu cầu.

Chỉnh sửa tệp _Layout.cshtml

- ◆ Bước 1: vào địa chỉ sau download template bootstrap về:
<http://startbootstrap.com/template-overviews/agency/>
- ◆ Bước 2: Tìm các tệp tin trong thư mục download về và copy vào thư mục project như hình
- ◆ Bước 3: Sửa lại đường dẫn ảnh trong tệp agency.css cho phù hợp.
- ◆ Bước 4: Mở tệp Views/Shared/_Layout.cshtml và chỉnh sửa code như hình



Chỉnh sửa tệp _Layout.cshtml



Chỉnh sửa tệp _Layout.cshtml

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>@ViewBag.Title - My ASP.NET Application</title>
  @Styles.Render("~/Content/css")
  @Scripts.Render("~/bundles/modernizr")
  <!-- Custom CSS -->
  <link href="/Content/agency.css" rel="stylesheet">
  <!-- Custom Fonts -->
  <link href="/Content/font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">
  <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet" type="text/css">
  <link href='https://fonts.googleapis.com/css?family=Kaushan+Script' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Droid+Serif:400,700,400italic,700italic' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Roboto+Slab:400,100,300,700' rel='stylesheet' type='text/css'>
</head>
<body id="page-top" class="index">
  <!-- Navigation -->
  <nav class="navbar navbar-default navbar-fixed-top">
    <div class="container">
      <!-- Brand and toggle get grouped for better mobile display -->
      <div class="navbar-header page-scroll">
        <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1">
          <span class="sr-only">Toggle navigation</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand page-scroll" href="#page-top">ASP.NET MVC 5</a>
      </div>
```

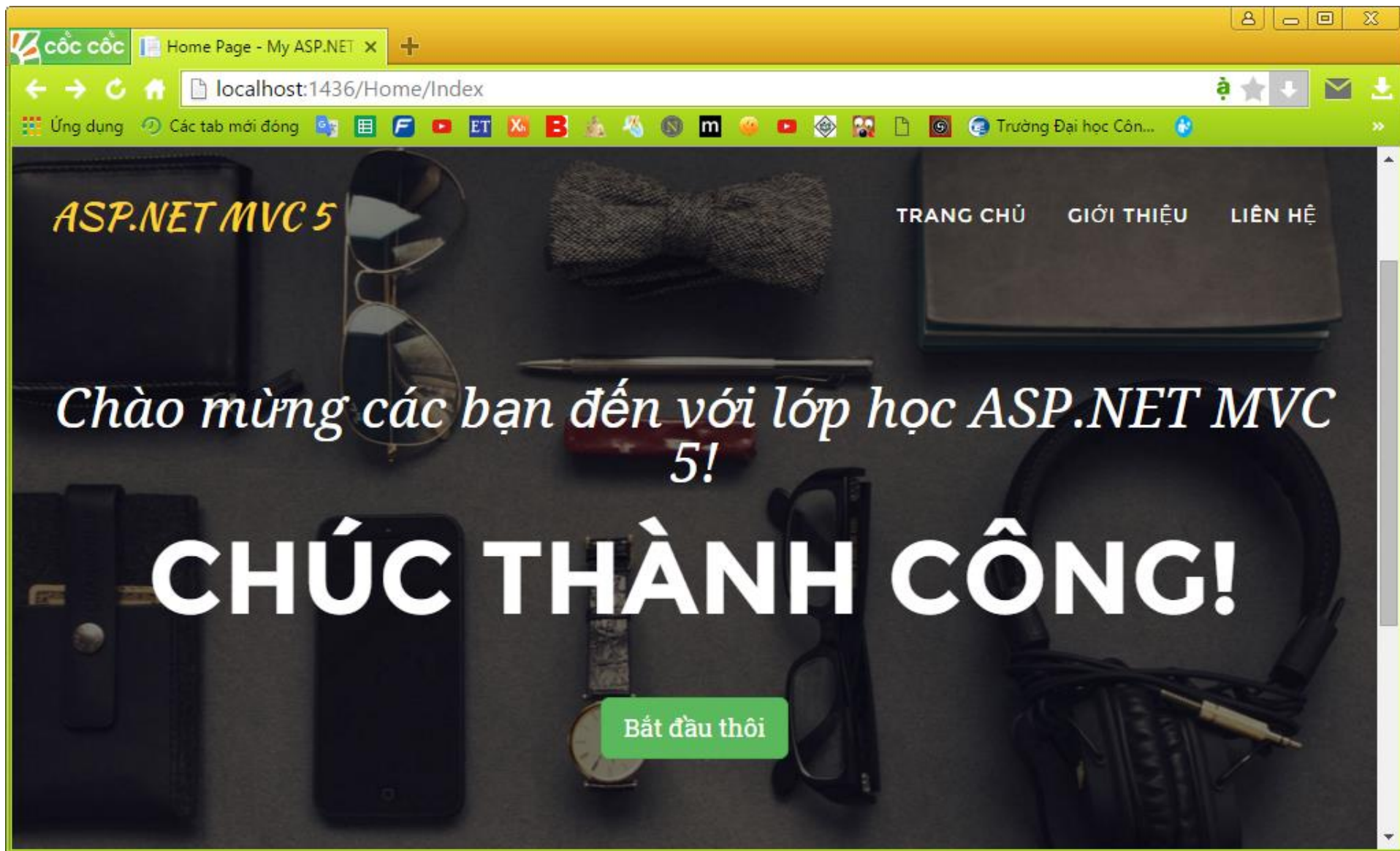
Chỉnh sửa tệp _Layout.cshtml

```
<!-- Collect the nav links, forms, and other content for toggling -->
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
  <ul class="nav navbar-nav navbar-right">
    <li class="hidden">
      <a href="#page-top"></a>
    </li>
    <li>
      <a class="page-scroll" href="/Home/Index">Trang chủ</a>
    </li>
    <li>
      <a class="page-scroll" href="/Home/About">Giới thiệu</a>
    </li>
    <li>
      <a class="page-scroll" href="/Home/Contact">Liên hệ</a>
    </li>
  </ul>
</div>
<!-- /.navbar-collapse -->
</div>
<!-- /.container-fluid -->
</nav>
<!-- Header -->
<header>
  <div class="container">
    <div class="intro-text">
      <div class="intro-lead-in">Chào mừng các bạn đến với lớp học ASP.NET MVC 5!</div>
      <div class="intro-heading">Chúc thành công!</div>
      <a href="#services" class="page-scroll btn btn-lg btn-success">Bắt đầu thôi</a>
    </div>
  </div>
</header>
```


Chỉnh sửa tệp _Layout.cshtml

```
<div class="container body-content">
    @RenderBody()
    <hr />
    <footer>
        <p>&copy; @DateTime.Now.Year - Phòng chuyên môn Bách Khoa Aptech</p>
    </footer>
</div>
@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/bootstrap")
@RenderSection("scripts", required: false)
</body>
</html>
```

Chỉnh sửa tệp _Layout.cshtml

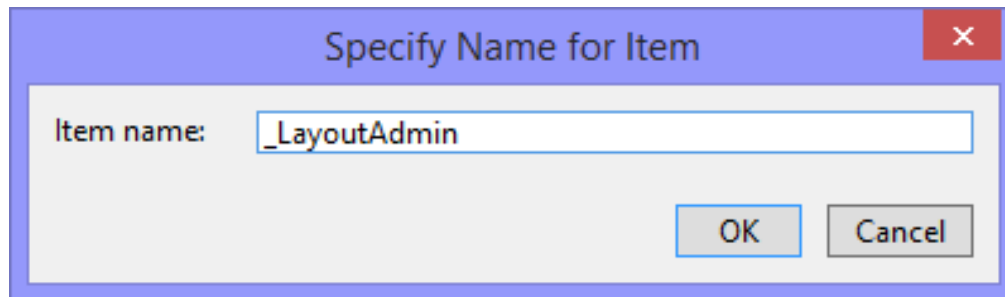


Chỉnh sửa tệp _Layout.cshtml

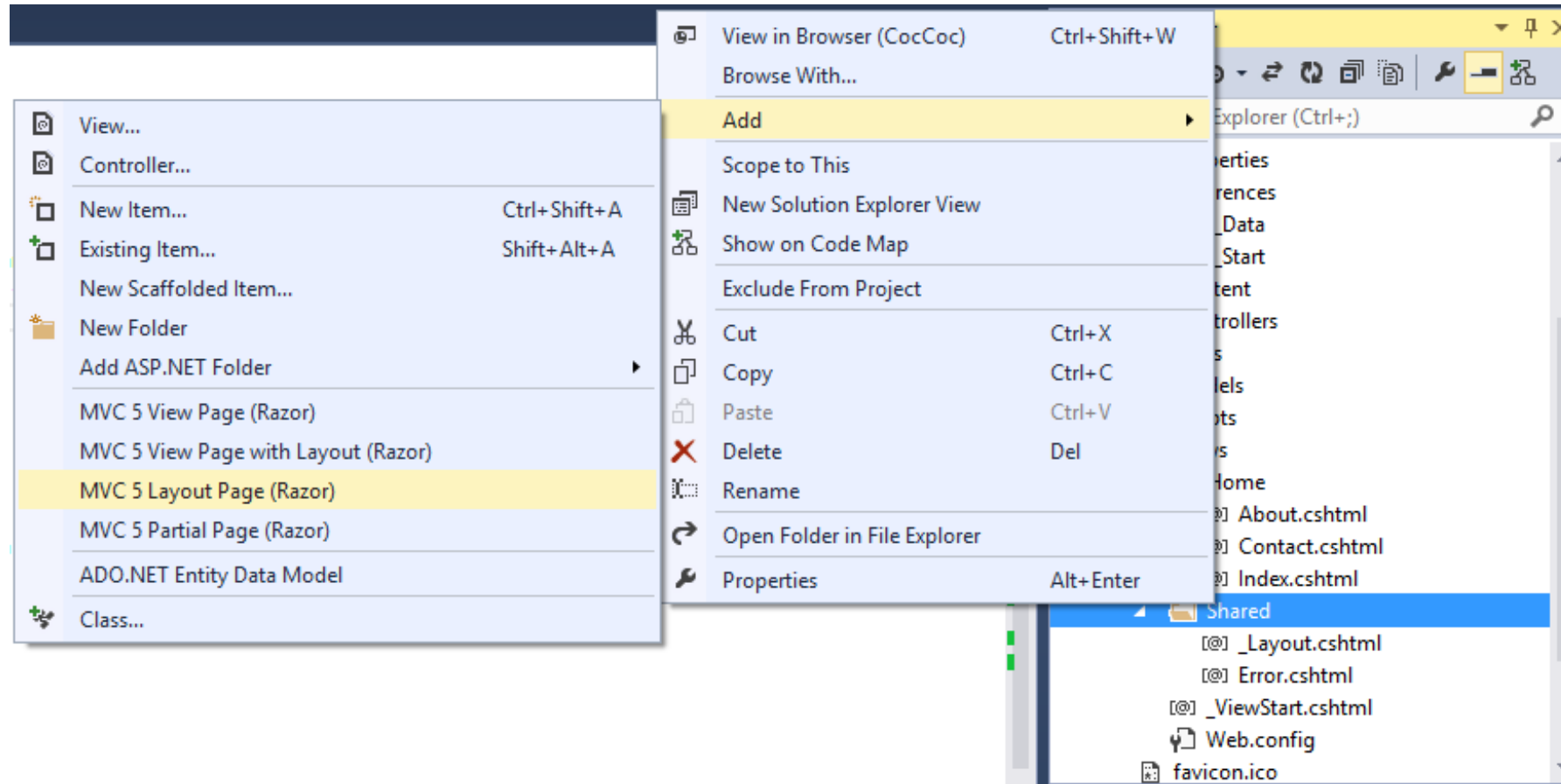


Tạo Custom Layout

- ◆ Tạo một Layout in Visual Studio 2013 tương tự như tạo một View. Mặc định tên layout bắt đầu với ký tự (_). Phần mở rộng của tệp layout là .cshtml.
- ◆ Bước 1: Kích chuột phải vào thư mục Views/Shared chọn add -> MVC 5 Layout Page (Razor)
- ◆ Bước 2: Nhập tên -> OK



Tạo Custom Layout



```
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>@ViewBag.Title</title>
</head>
<body>
    <div>
        @RenderBody()
    </div>
</body>
</html>
```



- ◆ Phương thức: `RenderBody()`: Cho phép bạn định nghĩa một Placeholder trong Layout để chèn nội dung chính của View, phương thức này chỉ được sử dụng 1 lần duy nhất trong View.
- ◆ `@RenderSection()`: Cho phép bạn hiển thị một đoạn được xác định trong view.



- ◆ Trong khi đang phát triển ứng dụng bạn có thể chỉ ra layout cho một View cụ thể hoặc cho tất cả các View.
- ◆ Cú pháp:
@{
Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
}
- ◆ Bạn cũng có thể chỉ ra Layout tất cả các View bằng cách vào tệp _ViewStart.cshtml và nhập dòng cú pháp như trên thay cho dòng mặc định.

[Xem Demo](#)

- ◆ Bên cạnh việc sử dụng layout, bạn cũng có thể sử dụng các style để áp dụng các định dạng đồng nhất qua các View.
- ◆ Style được sử dụng để định nghĩa 1 tập các tùy chọn định dạng, mà nó được sử dụng để định dạng các HTML Helpers trên một view hoặc nhiều view.



- ◆ Tệp tin .css trong ứng dụng ASP.NET Web dụng chứa đựng các định nghĩa style cần dùng trong ứng dụng, khi tạo ứng dụng ASP.NET MVC trong Visual Studio 2017, một tệp.css có tên Site.css được tự động tạo ra trong thư mục Content, tệp này chứa các định nghĩa style dùng cho trang mẫu mà asp.net tạo ra.

Xem code mẫu

Nhúng Styles vào trong Views

- ◆ Cú pháp:

- <link href="path of css file" rel="stylesheet" type="text/css" />

- ◆ Ví dụ

- <link href="~/Content/Site.css" rel="stylesheet" type="text/css" />



- ◆ Hiện giờ các thiết bị truy cập web rất đa dạng với các kích thước khác nhau, do đó các ứng dụng web phải đáp ứng việc thay đổi theo kích thước các thiết bị cho phù hợp.
- ◆ Cách thay đổi
 - ◆ Viewport meta tag
 - ◆ CSS media queries



- ◆ meta viewport nghĩa là một thẻ thiết lập cho trình duyệt hiển thị tương ứng với kích thước màn hình

```
<meta name="viewport" content="width=250">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```



- ◆ Để viết CSS tương ứng cho chiều rộng của trình duyệt, chúng ta sẽ sử dụng thẻ truy vấn @media trong CSS3

```
body {  
    background: #fff;  
    color: 333;  
}  
  
@media all and (max-width: 320px) {  
    body {  
        background: #e7e7e7;  
    }  
}
```

- ◆ Điều này có nghĩa là website của bạn mặc định sẽ có nền (background) màu trắng, nhưng khi thu nhỏ trình duyệt xuống còn từ 0px đến 320px thì nó sẽ có nền là màu xám với mã màu #e7e7e7.
- ◆ Trong đoạn trên, đoạn @media all and (max-width: 320px) nghĩa là thiết lập breakpoint cho toàn bộ thiết bị (all – media type) và có chiều rộng cố định tối đa là 320px (max-width – media features), 320px tương ứng với kích thước chiều rộng của màn hình iPhone. Và các đoạn CSS nằm bên trong query này sẽ được thực thi khi co màn trình duyệt lại đúng với kích thước từ 320px trở xuống.

- ◆ Ngoài ra, bạn cũng có thể thiết lập thêm nhiều điều kiện như:

```
@media only screen and (min-width: 320px) and (max-width: 860px) {...}
```

- ◆ Tức là query này sẽ chỉ áp dụng cho màn hình desktop, máy tính bảng, smartphone và có kích thước màn hình tối thiểu là 320px nhưng nhỏ hơn 860px.
- ◆ Về các giá trị của media type và media features trong @media query, bạn có thể tham khảo thêm tại W3Schools cho đầy đủ hơn nhưng thông thường ta cứ viết CSS cho media type là screen và media features chỉ quanh quẩn min-width, max-width mà thôi.



- ◆ JavaScript: là ngôn ngữ kịch bản chạy phía client cho phép bạn thao tác với các phần tử HTML và trình duyệt để tạo lên sự tương tác với ứng dụng web ngay tại trình duyệt mà không cần tương tác với Web Server.
- ◆ Mặc định khi tạo project ASP.NET MVC các thư viện javascript như bootstrap, jquery... được cài đặt sẵn trong thư mục Scripts
- ◆ Để sử dụng bạn cần đưa vào theo cú pháp sau

```
<script src= "@Url.Content("~/Scripts/Testscript.js")"
type="text/javascript">
</script>
```


- ◆ JQuery là một thư viện JavaScript tương thích với các trình duyệt được sử dụng để thực hiện các chức năng khác nhau như tìm kiếm, duyệt, thao tác với các phần tử HTML, điều khiển sự kiện, tạo các hiệu ứng... trên phía client.
- ◆ Đưa jquery vào sử dụng

```
<script type= "text/javascript"
src="@Url.Content("~/Scripts/jquery-1.7.1.js")">
</script>
```
- ◆ jQuery cung cấp hàm `document.ready()` để đảm bảo View của ứng dụng web được tải lên xong trước khi thực thi code JQuery

- ◆ Cú pháp

```
$(document).ready(function(){  
  // jQuery code...  
});
```

- ◆ Trong đó

`$`: bắt đầu code jquery.

`(document).ready()`: Kiểm tra sự sẵn sàng để thực hiện các hành động trên phần tử HTML.

- ◆ Cú pháp:
\$(selector).action()
trong đó
\$: Defines or access jQuery.
selector: Tìm phần tử HTML hoặc nhóm phần tử dựa trên name, id, class và attribute.
action(): chỉ ra hành động thực thi trên phần tử đó.
- ◆ Ví dụ:
\$("h2")
\$("#green")
\$(".red")



- ◆ Với jQuery, để điều khiển sự kiện bạn có thể sử dụng các hàm hoặc các phương thức có

Phương thức sự kiện	Mô tả
<code>\$(document).ready(function)</code>	Cho phép thực thi 1 lần khi tài liệu tải hoàn thành
<code>\$(selector).click(function)</code>	Cho phép thực thi một hàm khi kích chuột vào phần tử selector
<code>\$(selector).dblclick(function)</code>	kích double vào phần tử selector
<code>\$(selector).mouseover(function)</code>	Di chuột vào phần tử selector
<code>\$(selector).mouseout(function)</code>	Di chuột ra khỏi phần tử selector
<code>\$(selector).keydown(function)</code>	Nhấn 1 phím trên phần tử selector

- ◆ Hiệu ứng ẩn 1 phần tử
\$(selector).hide(speed)
trong đó
speed: là tốc độ có thể sử dụng một trong các giá trị: slow, fast
hoặc số milliseconds
- ◆ Hiệu ứng tăng giảm độ trong suốt của phần tử
\$(selector).fadeOut()
\$(selector).fadeIn()



- ◆ jQuery UI cung cấp một công nghệ cho phép bạn sử dụng để add, remove, hoặc toggle class cho các phần tử HTML.

```
$(document).ready(function () {  
    $("#button1").click(function () {  
  
        $("#div1").addClass("myclass", 550);  
  
    });  
    $("#button2").click(function () {  
  
        $("#div1").removeClass("myclass", 550);  
    });  
    $("#button3").click(function () {  
        $("#div1").toggleClass("myclass", 550);  
    });  
});
```

- ◆ Ajax là một công nghệ phát triển Web cho phép bạn gửi các request tới server trong chế độ background sử dụng code client, cho phép cập nhật lại view mà không cần tải lại toàn bộ trang, cải thiện hiệu suất ứng dụng của bạn.
- ◆ Ajax giao tiếp ở dạng bất đồng bộ tức là nó có gửi và nhận nhiều request và response từ server đồng thời, người dùng không phải mất thời gian chờ đợi kết quả mà vẫn có thể thực hiện các thao tác khác trên client.



- ◆ Ứng dụng ASP.NET MVC sử dụng một loạt các AJAX helper để tạo form và link tới các action trong controller.
- ◆ AJAX Helper giúp bạn không cần phải viết code javascript ajax tường minh mà nó tự sinh ra code phù hợp.
- ◆ Để thực hiện việc này bạn cần phải đảm bảo rằng tệp `jquery.unobtrusiveajax` script có mặt trong thư mục Scripts của ứng dụng.



- ◆ Phương thức `Ajax.ActionLink()` helper cho phép bạn tạo một thẻ anchor với khả năng bất đồng bộ

```
<div id="attachmentImage">
@Ajax.ActionLink("Attachment", "AttachmentImage",
new AjaxOptions
{
    HttpMethod = "Post",
    InsertionMode = InsertionMode.Replace,
    LoadingElementId = "loadingImage",
    UpdateTargetId = "attachmentImage"
})
</div>
<div id="loadingImage" style="display:none">
Loading attachment...
</div>
```

- ◆ `Ajax.BeginForm()`: tạo ra một form cho phép submit thông tin ở dạng bất đồng bộ

```
@using (Ajax.BeginForm("ProductSearch", "Product",  
    new AjaxOptions {  
        InsertionMode=InsertionMode.Replace,  
        HttpMethod="GET",  
        OnFailure="searchFailed",  
        LoadingElementId="ajax-loader",  
        UpdateTargetId="searchresults",  
    })  
{  
    <input type="text" name="q" />  
    <input type="Submit" value="search" />  
      
}
```

