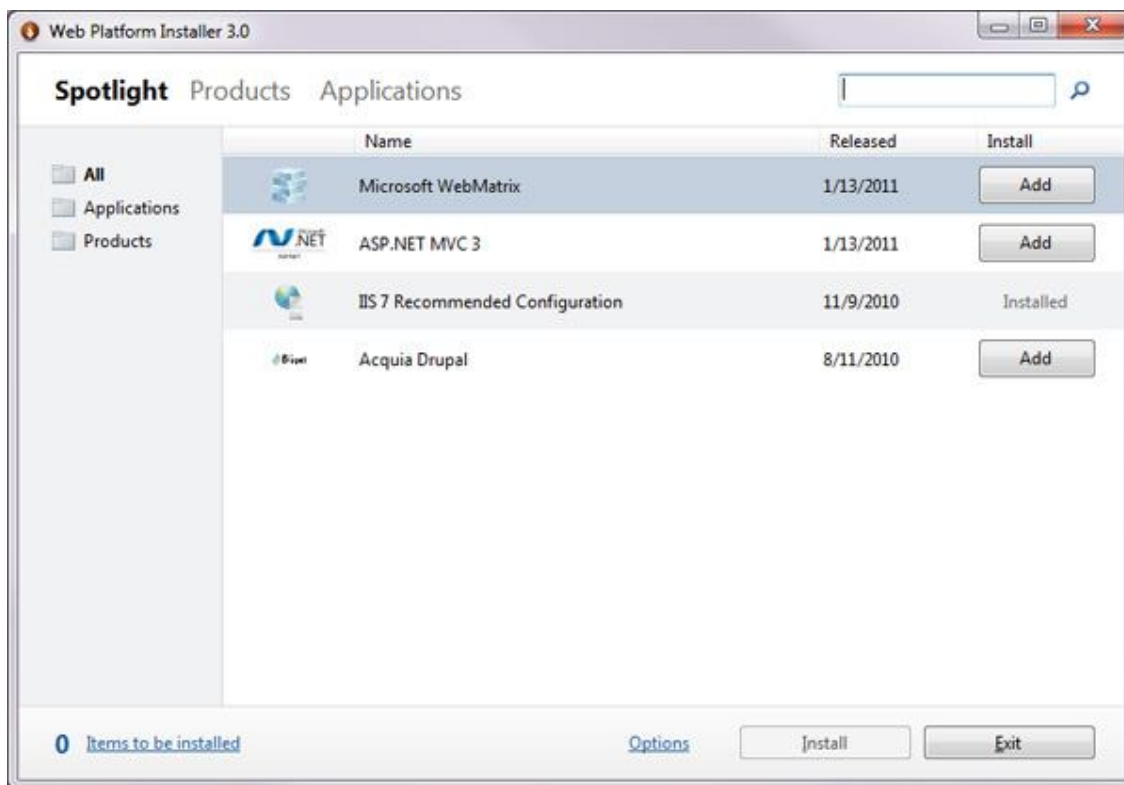


Tìm hiểu về WebMatrix (phần 1)

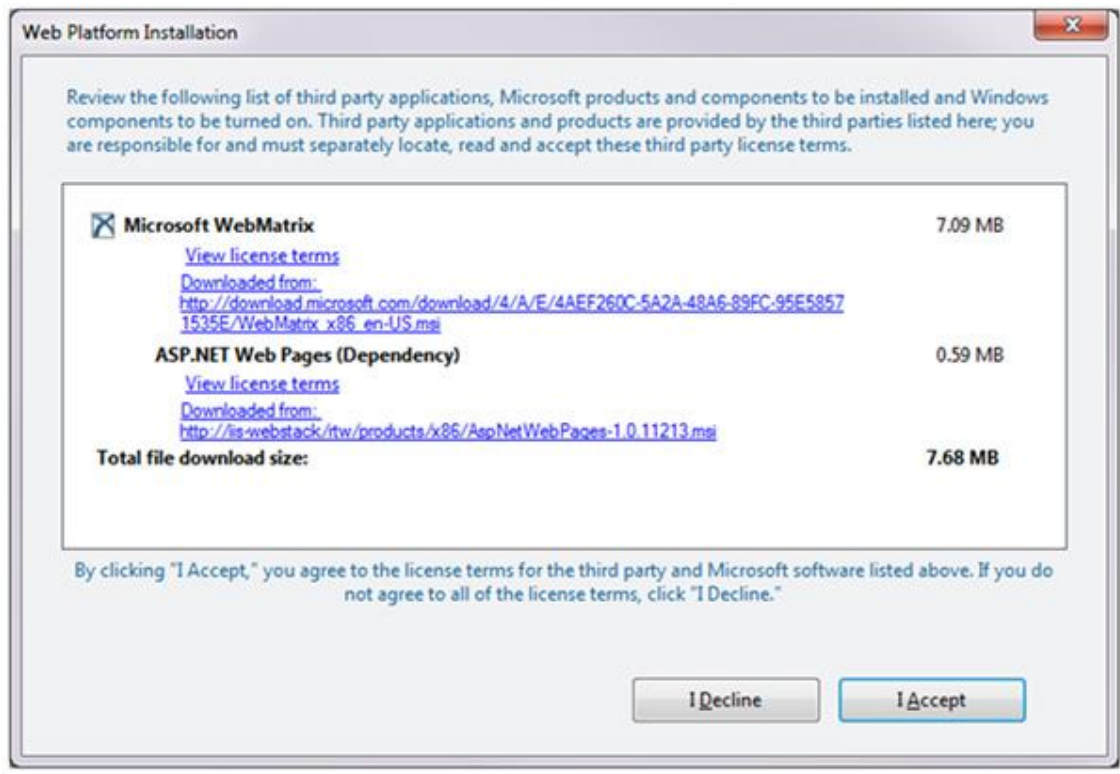
Trong loạt bài viết sau, chúng tôi sẽ giới thiệu một số bước cơ bản để xây dựng ứng dụng web đơn giản sử dụng **WebMatrix, CSS, HTML, HTML5, ASP.NET, SQL, Database...** Tại đây, chúng ta sẽ cùng nhau xây dựng 1 ứng dụng web cơ bản để quản lý danh sách những bộ phim ưa thích của bạn, bao gồm công đoạn tạo cơ sở dữ liệu, tạo và phân quyền cho các tài khoản, chỉnh sửa và xóa database... thuật ngữ *CRUD* – *Create, Retrieve, Update, Delete* chứa đựng đầy đủ các tiến trình trên, và trong phần sau, chúng ta sẽ bắt đầu ứng dụng với WebMatrix.

1. Bắt đầu với WebMatrix:

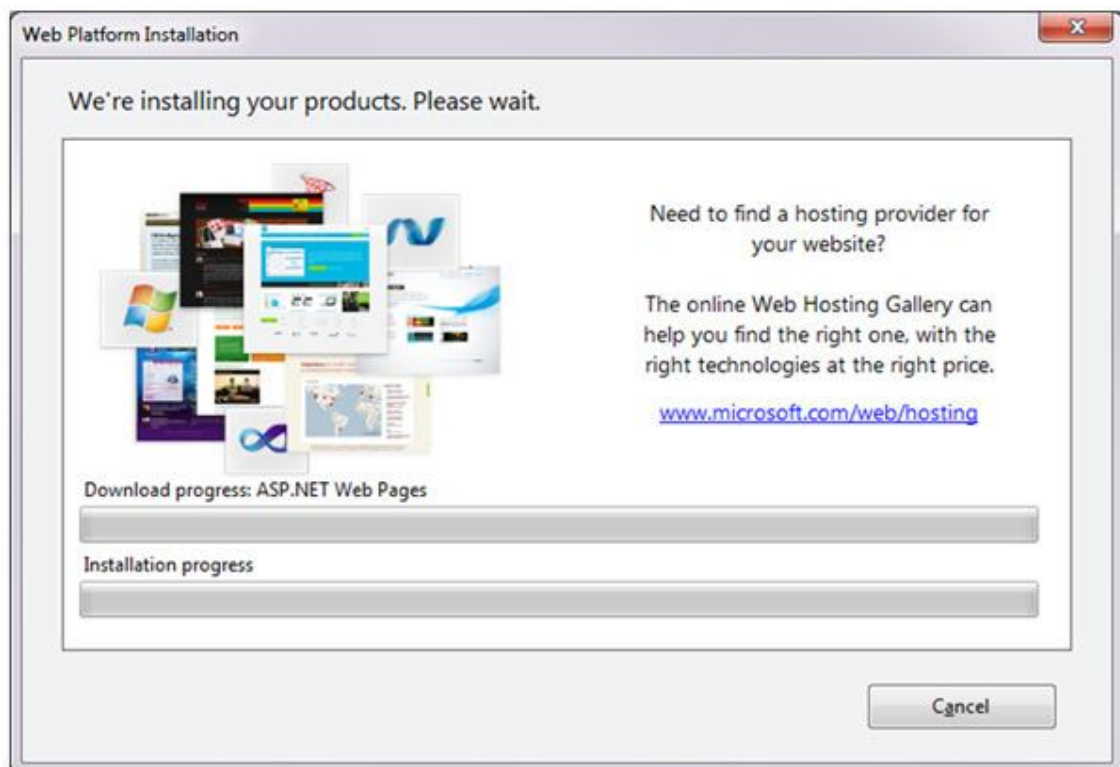
Microsoft WebMatrix là 1 công cụ miễn phí giúp người sử dụng dễ dàng tạo, thiết lập, tùy chỉnh và đẩy toàn bộ website của bạn lên Internet. Các bạn có thể bắt đầu với một số ứng dụng mã nguồn mở như *WordPress, Joomla, DotNetNuke* hoặc *Orchard*, và WebMatrix có thể nắm giữ, xử lý được tất cả nhiệm vụ như download, tùy chỉnh, cấu hình, thiết lập ứng dụng... hoặc người sử dụng có thể tự viết code dựa vào những template có sẵn... Cho dù chọn hình thức nào, WebMatrix vẫn cung cấp đầy đủ những yếu tố cần thiết để tạo 1 hệ thống **web server, database và framework**. Để bắt đầu, các bạn hãy truy cập vào [đây](#), download công cụ *Microsoft Web Platform Installer* – được sử dụng với tất cả các sản phẩm của Microsoft, bao gồm cả *WebMatrix*. Khi khởi động *Web Platform Installer*, các bạn sẽ nhìn thấy màn hình sau:



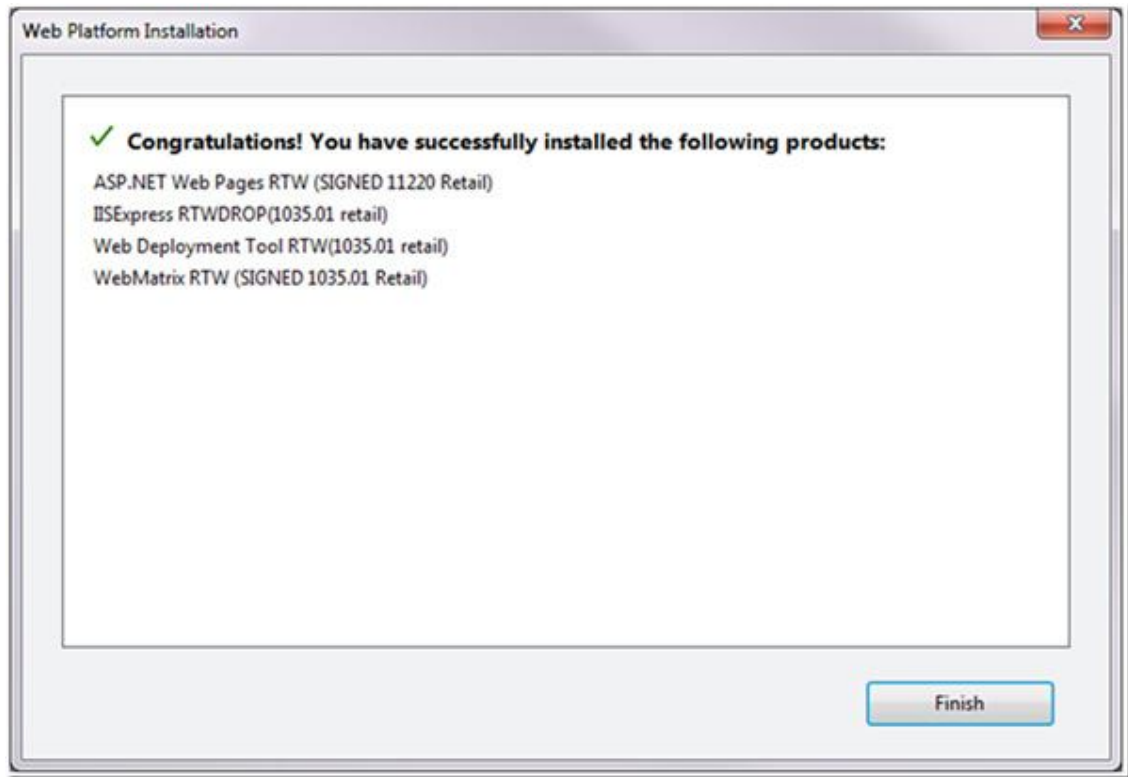
Để cài đặt, các bạn cần tìm được *WebMatrix* trong toàn bộ danh sách các sản phẩm của *Microsoft*, sau đó nhấn nút *Add* bên cạnh và *Install* ở phía dưới màn hình. Hệ thống sẽ hiển thị thông tin thỏa thuận và điều khoản sử dụng giữa nhà phát triển và người sử dụng, tương tự như hình bên dưới:



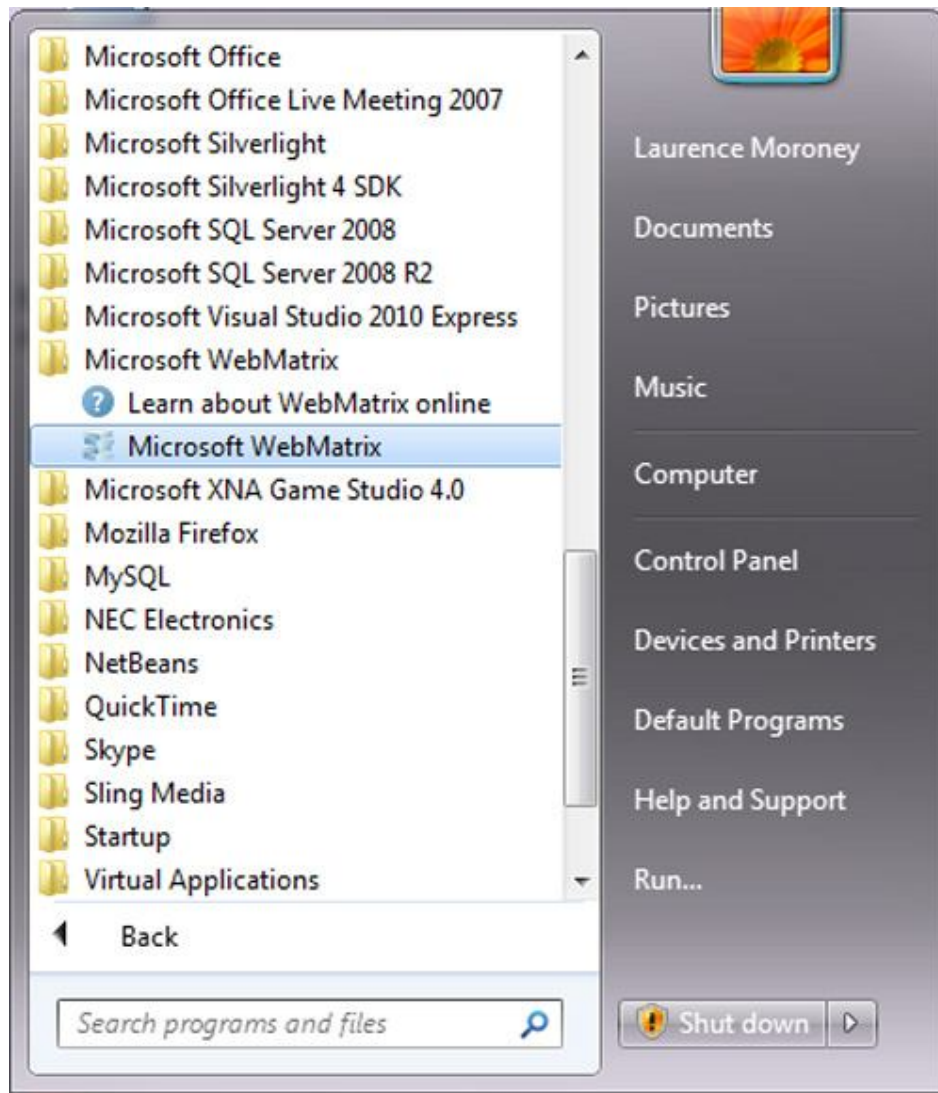
Sau khi đồng ý, chương trình sẽ tự động tải về những file cần thiết và bắt đầu cài đặt vào hệ thống:



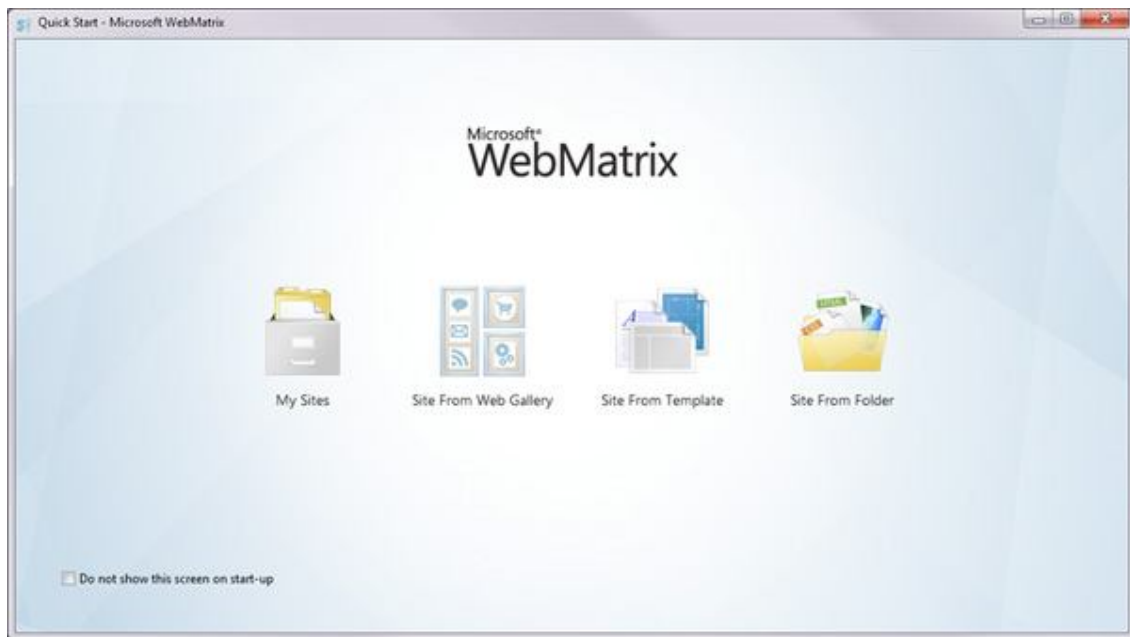
Khi hoàn tất công đoạn này, bạn sẽ nhìn thấy màn hình *Congratulations* hiển thị, cùng với chi tiết về các gói đã được tích hợp:



Mở *Start Menu*, các bạn sẽ thấy thư mục *Microsoft WebMatrix*:



Chọn tiếp *Microsoft WebMatrix*, màn hình *Welcome* ban đầu sẽ hiển thị:



Tại đây, các bạn có thể thấy những lựa chọn:

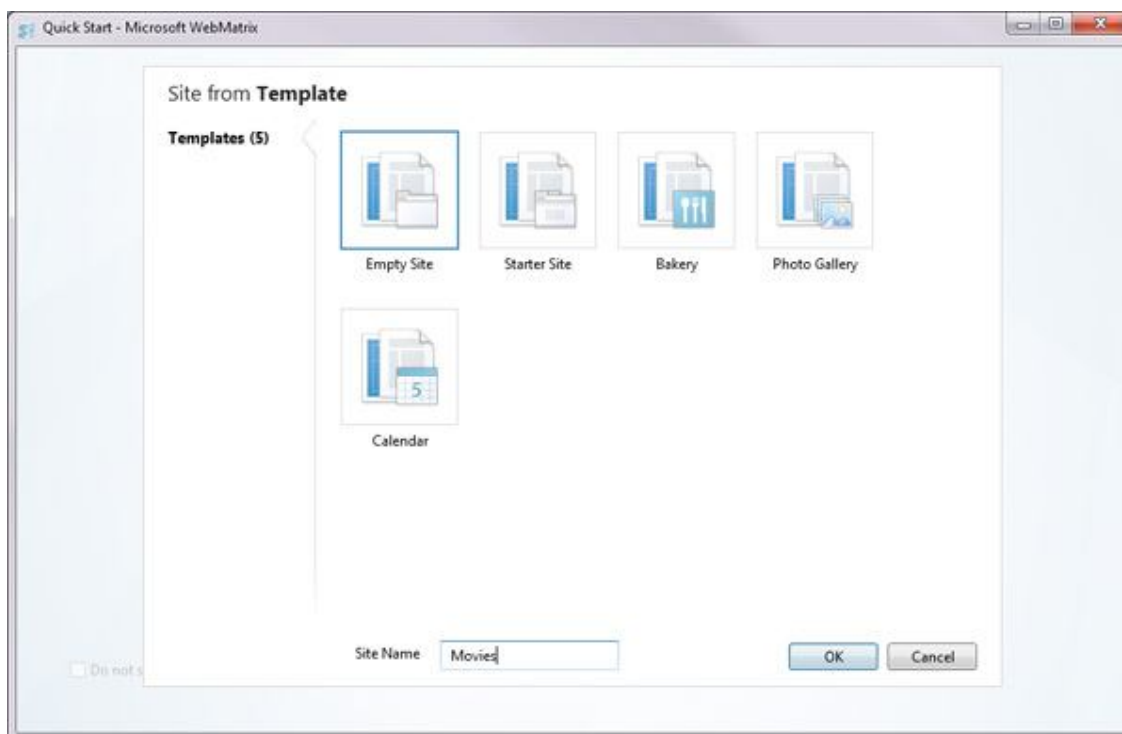
- **Site From Web Gallery:** Web Application Gallery (WAG) – có thể coi là 1 bộ sưu tập các ứng dụng web mã nguồn mở, được viết bằng nhiều công nghệ khác nhau, bao gồm PHP và ASP.NET.

- **Site From Template:** nếu bạn muốn tự mình xây dựng 1 site riêng, thay vì việc sử dụng mẫu từ WAG, đây là lựa chọn khá lý tưởng với 1 vài template đơn giản có sẵn.

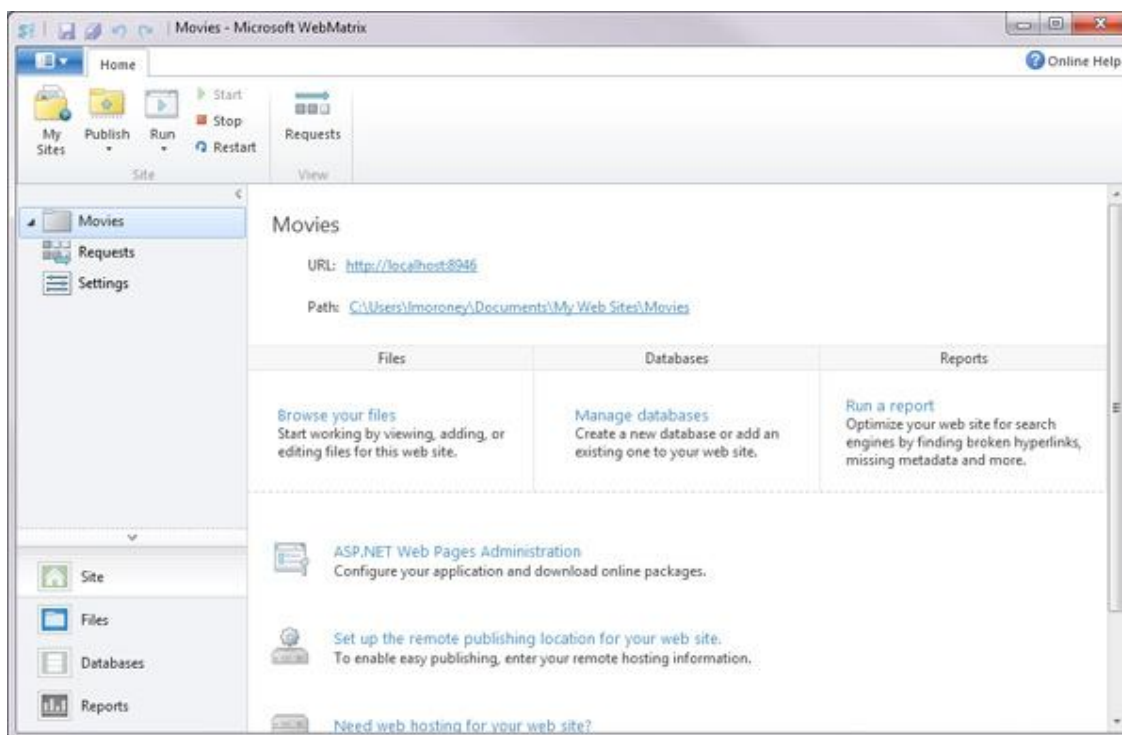
- **Site From Folder:** được chọn khi bạn có sẵn nhiều site đã làm trước đó, chỉ cần chỉnh sửa lại 1 chút thông tin và có thể triển khai nhanh chóng trên bất cứ webserver nào.

2. Những bước đầu tiên tạo website:

Khi chọn mục *Site From Template*, chương trình sẽ hiển thị hộp thoại như bên dưới, chọn phần *Empty Site* và đặt tên là *Movies*:

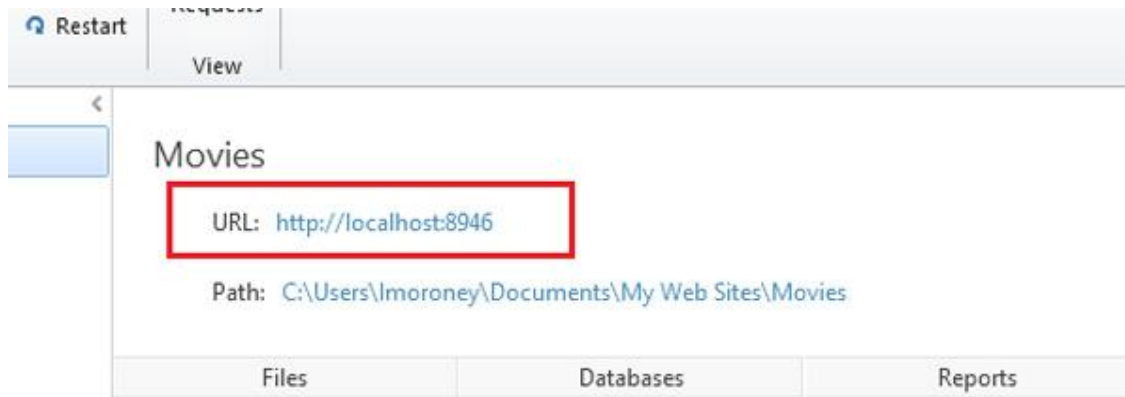


Nhấn *OK*, *WebMatrix* sẽ tạo mới 1 website hoàn toàn mới cho bạn, các yếu tố có liên quan đều được hiển thị đầy đủ tại màn hình *Editor* chính:



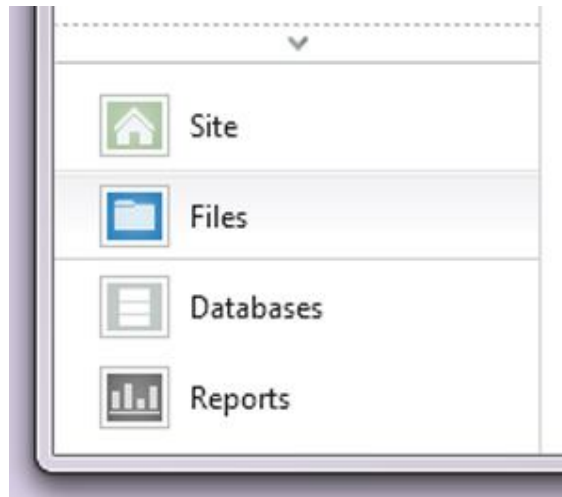
Trước khi tiến hành những bước tiếp theo, chúng ta cần hiểu rõ về những gì được đề cập tại đây. Điểm đầu tiên, *WebMatrix* là 1 chương trình tổng

hợp nhiều tính năng, không đơn giản chỉ là 1 công cụ tạo và chỉnh sửa code, được tích hợp với web server với tên gọi là *IIS Express*. Có thể coi web server là 1 phần đặc biệt của chương trình, có khả năng “lắng nghe” tất cả các yêu cầu về dữ liệu qua Internet, và phản hồi lại. Mỗi khi mở trình duyệt và gõ địa chỉ bất kỳ, ví dụ như <http://www.microsoft.com/>, nghĩa là bạn đang chuyển yêu cầu tới 1 web server của microsoft, và hệ thống sẽ trả lời bằng cách gửi về những thành phần như mã *HTML*, *JavaScript*, *CSS*, *ảnh...* và trình duyệt của bạn đảm nhận nhiệm vụ “nhúng” toàn bộ các thành phần đó trong 1 trang web:

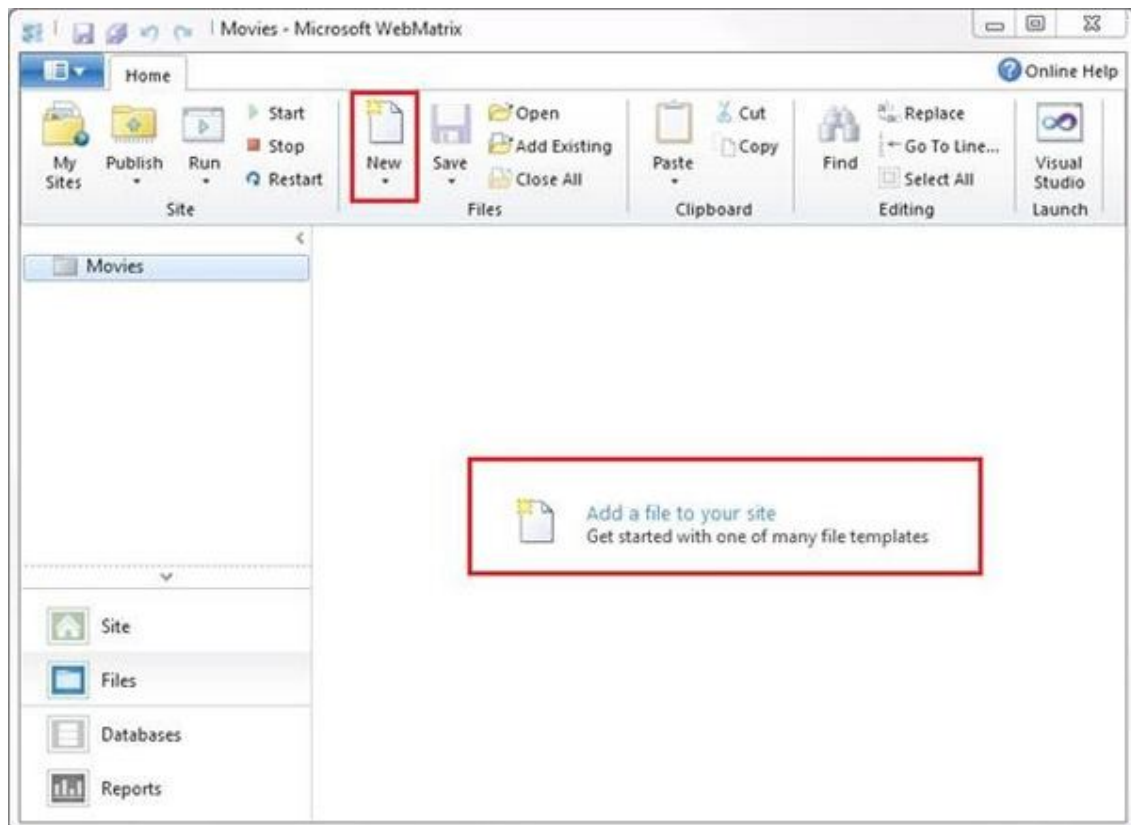


Việc có 1 server dựa trên WebMatrix sẽ giúp bạn dễ dàng hơn trong việc tạo, chỉnh sửa các thành phần như khi làm việc trực tiếp trên web server vậy. Khi nhìn vào màn hình chính, bên dưới phần tên site (ở đây là **Movies**), bạn sẽ thấy server đang “phục vụ” site qua địa chỉ **<http://localhost:8946/>**, nghĩa là host của server này là local (khi chúng ta làm trực tiếp trên máy tính). Tại thời điểm này, nếu cho chạy website thì bạn sẽ nhận được thông báo lỗi, đơn giản bởi vì chúng ta chưa tạo bất cứ nội dung gì cho trang web. Hãy tiếp tục với những bước sau đây.

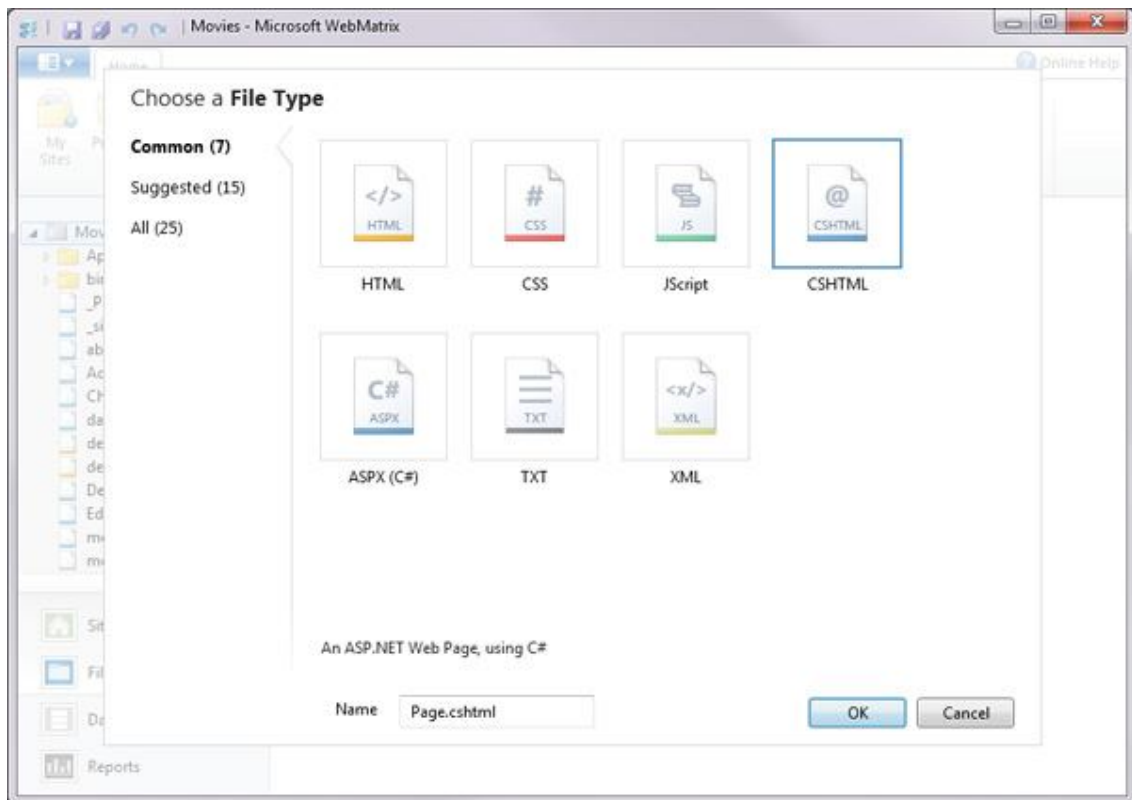
Bên cạnh đó, *WebMatrix* cho phép người sử dụng dễ dàng chuyển đổi qua lại giữa các *workspace* khác nhau bằng nút bấm tại cửa sổ bên trái:



WebMatrix sẽ mở cửa sổ làm việc *Files*, hoàn toàn trống rỗng vì chưa có bất kỳ nội dung gì. Nhấn nút *New* để tạo thành phần mới:



Tiếp đó, chọn từng loại file thường dùng tương ứng cần thêm vào website:



Tại đây, chúng ta chọn *HTML* và đặt tên là **default.html**:

```
default.html x
1 <!DOCTYPE html>
2
3 <html lang="en">
4   <head>
5     <meta charset="utf-8" />
6     <title></title>
7   </head>
8   <body>
9
10  </body>
11 </html>
12
```

1 file *HTML* (*HyperText Markup Language*) có thể coi là tổng hợp những

Chỉ dẫn trình duyệt làm thế nào để hiển thị 1 trang web một cách hoàn chỉnh. Thông thường, trong đây sẽ bao gồm thẻ Head, chứa những thông tin tổng quát về trang web, phần Body chứa nội dung chính. Nội dung bên trong được đánh dấu và phân biệt bằng các thẻ – tag, được bắt đầu với tên của tag trong dấu ngoặc đơn, ví dụ <body> và kết thúc bằng dấu gạch chéo, ví dụ </body>. Do đó, tất cả những gì ở bên trong thẻ tag sẽ được coi là phần thân của trang web, và hiển thị qua trình duyệt.

Chỉnh sửa lại mã của trang web sao cho giống như sau:

```
<!DOCTYPE html>

<html lang="en" >

<head>

<meta charset="utf-8" />

<title>My Favorite Movies</title>

</head>

<body>

<h1>A list of my Favorite Movies</h1>

<ol>

<li>It's a wonderful life</li>

<li>Lord of the Rings</li>

<li>The Fourth World</li>

<li>The Lion King</li>

</ol>

</body>

</html>
```

Có thể thấy, tại đây có một số tag cơ bản như <title>, <body>, <h1>, , với chức năng khác nhau. Tiếp theo, các bạn nhấn nút *Run* trên thanh công cụ của *WebMatrix*:



WebMatrix sẽ mở trang web vừa tạo bằng trình duyệt mặc định của hệ thống, và tất nhiên, site này chạy trực tiếp trên local:



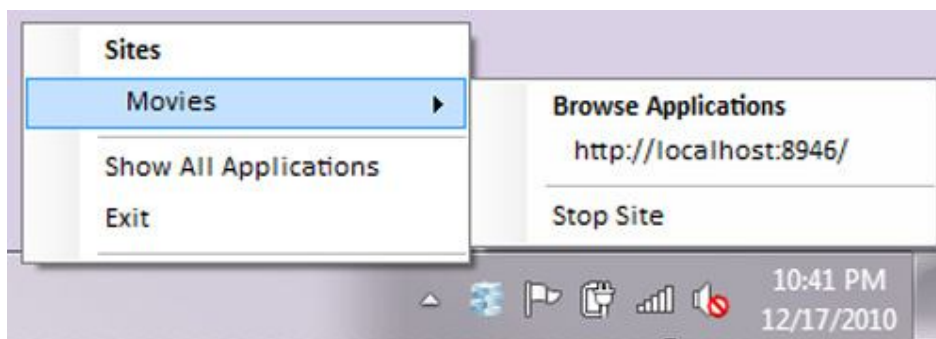
Các bạn có để ý đến địa chỉ trên trình duyệt? Nó không mở 1 file cụ thể nào, mà khởi động web server, và trở tới địa chỉ đó, yêu cầu web server tải file **default.html**:



Nhìn xuống phía dưới khay hệ thống, bạn sẽ thấy biểu tượng nhỏ xuất hiện, chỉ định rằng IIS Express – web server của chúng ta vẫn đang hoạt động:



Kích chuột phải vào biểu tượng đó, bạn sẽ thấy site Movies đang hoạt động:



Tại thanh tiêu đề của trình duyệt, các bạn thấy *My Favorite Movies*. Và để so sánh, chúng ta sẽ thử nghiệm trên các trình duyệt với nhau, như *Explorer*, *Chrome*, *Safari*, *FireFox* và *Opera*. Các đoạn text trong thẻ `<title>` tại phần `<head>` của HTML, và hiển thị khác nhau tùy theo từng trình duyệt:



Internet Explorer



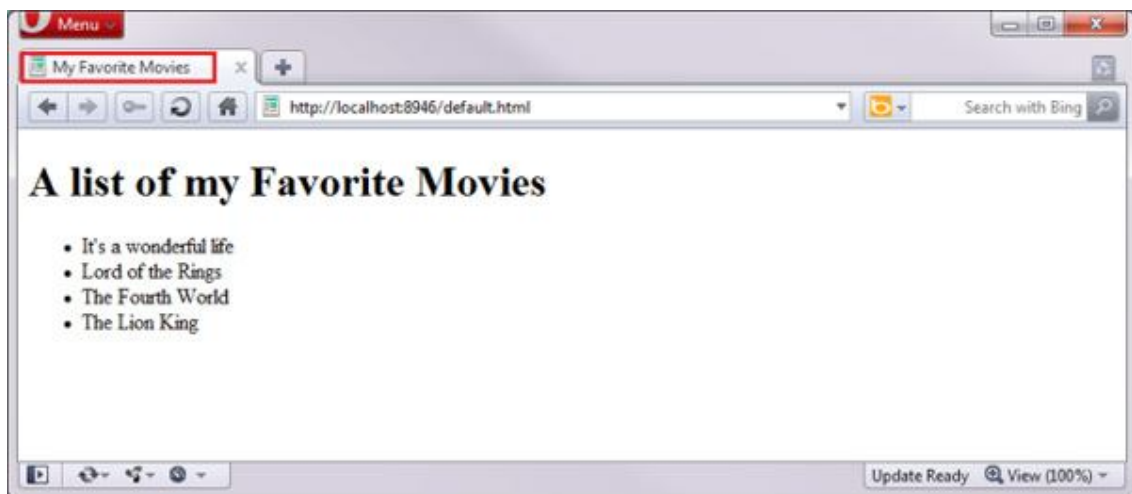
Google Chrome



Safari

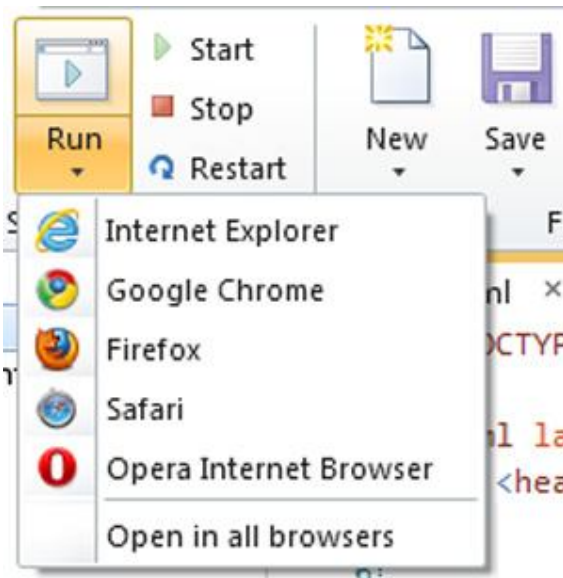


Mozilla Firefox



Opera

Để đảm bảo rằng web site của bạn hoạt động tốt trên tất cả các trình duyệt, vì vậy hãy nhớ kiểm tra kỹ sản phẩm của mình. Tiện ích này cũng đã được tích hợp sẵn trong *WebMatrix*:



Tiếp đến là phần body, tại đây chúng ta điền *A list of my Favorite Movies* bên trong thẻ `<h1>` - nghĩa là toàn bộ nội dung bên trong đó được hiển thị dưới dạng Heading 1:



Trong đây, chúng ta còn sử dụng thêm thẻ `` (**Unordered List**) và `` (**List Item**), trình duyệt sẽ tự động sắp xếp các đối tượng bên trong theo đúng chức năng.

Tìm hiểu về WebMatrix (phần 2)

3. Tìm hiểu về style:

Tiếp theo, chúng ta sẽ bắt đầu việc sắp xếp và chỉnh sửa giao diện bên ngoài sử dụng công nghệ quen thuộc *Cascading Style Sheets (CSS)*. Danh sách các bộ phim ưa thích được hiển thị trên trang web trong bài thử nghiệm của chúng ta như sau:



Sử dụng Divider:

Trong HTML, chúng ta có thể phân chia toàn bộ trang ra thành từng phần khác nhau sử dụng thẻ `<div>`. Tính năng này thực sự hữu ích đối với việc chỉnh sửa giao diện, và với mỗi phần khác nhau, người sử dụng hoàn toàn có thể bố trí và áp dụng các phong cách cho phù hợp với nhu cầu. Dưới đây là toàn bộ mã HTML của site ví dụ trên:

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<title>My Favorite Movies</title>

</head>

<body>
```

```
<h1>A list of my Favorite Movies</h1>
<ol>
<li>It's a wonderful life</li>
<li>Lord of the Rings</li>
<li>The Fourth World</li>
<li>The Lion King</li>
</ol>
</body>
</html>
```

Việc đầu tiên là cho danh sách các bộ phim vào bên trong thẻ <div>, như sau:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<title>My Favorite Movies</title>
</head>
<body>
<h1>A list of my Favorite Movies</h1>
<div id="movieslist">
<ul>
<li>It's a wonderful life</li>
<li>Lord of the Rings</li>
<li>The Fourth World</li>
```

```
<li>The Lion King</li>
</ul>
</div>
</body>
</html>
```

Giờ đây, bạn cũng có thể thấy rằng các tag `` cũng đã ở bên trong `<div>`. Nhưng khi xem lại trang web thì chúng ta chẳng thấy gì khác biệt so với lúc trước, Bởi vì thẻ `<div>` trong trường hợp này được áp dụng bên trong hệ thống.

Sử dụng Hyperlink:

Khái niệm *đường dẫn liên kết – hyperlink* tại đây có nghĩa là 1 đoạn nội dung văn bản hoặc 1 ký tự bất kỳ, khi nhấn vào đường dẫn đó, bạn sẽ được chuyển tới địa chỉ “nhúng” bên trong đó. Tính năng này được bắt đầu bằng thẻ `<a>`, toàn bộ nội dung bên trong `<a>` và `` sẽ xảy ra sự kiện nhấn chuột, khi người dùng di chuột qua đoạn text này, trình duyệt sẽ thay đổi chế độ và chuyển họ đến địa chỉ cụ thể đó, thông qua *HREF (Hyper-reference)*.

Thuộc tính được khởi tạo và định nghĩa bên trong thẻ tag đó, ví dụ như sau:

```
<tag
attribute="attributevalue">content</tag>
```

Do đó, để tạo hyperlink, các bạn sử dụng cú pháp sau:

```
<a href="http://www.microsoft.com">Click
Here</a>
```

Bên trong href không nhất thiết phải là địa chỉ web site, có thể là 1 đoạn mã JavaScript bất kỳ nào đó. Bên cạnh đó, còn có 1 cú pháp href đặc biệt thường được sử dụng để kiểm tra xem style hoặc hyperlink đó có hoạt động hay không bằng cách thêm # vào.

Do vậy, tại đây những đối tượng bên trong `` đều được chuyển thành hyperlink, bao quanh nội dung bên trong `<a>`, và thiết lập thuộc tính HREF thành #, giống như sau:

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<title>My Favorite Movies</title>

</head>

<body>

<h1>A list of my Favorite Movies</h1>

<div id="movieslist">

<ol>

<li><a href="#">It's a wonderful
life</a></li>

<li><a href="#">Lord of the Rings</a></li>

<li><a href="#">The Fourth World</a></li>

<li><a href="#">The Lion King</a></li>

</ol>

</div>

</html>
```

Khi chạy, trang web của chúng ta sẽ hiển thị như sau:



Danh sách các bộ phim trông giống như các hyperlink, được gạch chân và có màu xanh.

Thêm Header và Footer:

Tiếp theo, chúng ta sẽ thêm phần Header và Footer vào bên trong mã HTML, cụ thể là thẻ <header> và <footer> trong HTML5. Các bạn có thể tìm hiểu thêm về HTML5 tại [đây](#).

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<title>My Favorite Movies</title>
</head>
<body>
<header>
<h1>A list of my Favorite Movies</h1>
</header>
<div id="movieslist">
<ol>
```

```
<li><a href="#">It's a wonderful  
life</a></li>  
  
<li><a href="#">Lord of the Rings</a></li>  
  
<li><a href="#">The Fourth World</a></li>  
  
<li><a href="#">The Lion King</a></li>  
  
</ol>  
  
</div>  
  
<footer>  
  
This site was built using Microsoft  
WebMatrix.  
  
<a href="http://web.ms/webmatrix">Download  
it now.</a>  
  
</footer>  
  
</html>
```

Đối với phần header, ta sẽ cho `<h1>` đã tạo trước đó vào bên trong tag `<header>`. Còn với footer, chúng ta điền thêm 1 ít thông tin và gắn hyperlink. Ví dụ như sau:



Trong những phần trên, chúng ta đã tham khảo về những thẻ đánh dấu, thuộc tính chỉ định, các font chữ, kích cỡ, màu sắc, header và footer, cách thức hoạt động của từng thẻ đó, sự kiện xảy ra khi click chuột... Như

phần trên, chúng ta đã khởi tạo thẻ `<h1>` để biến A list of my Favorite Movies với kích cỡ lớn. Ví dụ tại đây, chúng ta sẽ thêm một số thuộc tính khác:

```
<h1 style="color:blue; font-size:32; font-family:Verdana; text-decoration:underline">A list of my Favorite Movies</h1>
```



Thông tin bên trong thẻ `<h1>` sau khi được chuyển thành màu xanh, font Verdana, kích cỡ 32 và kiểu chữ gạch chân

Bên cạnh cách trên để thay đổi vẻ bề ngoài của site, chúng ta còn có thể áp dụng thêm cách khác, đó là sử dụng Cascading Style Sheets như đã đề cập bên trên. Thay vì việc đặt tất cả các đoạn code trang trí vào trong thẻ `<h1>`, chúng ta sẽ chỉ định rõ ràng từng lớp thuộc tính như sau:

```
<h1 class="Title">A list of my Favorite Movies</h1>
```

Giờ đây, tag này đã có thêm 1 lớp – class mới, và chúng ta có thể chỉ ra cho trình duyệt sử dụng các kiểu style khác nhau để áp dụng vào class này. Quá trình này có thể được làm bằng mã CSS như sau:

```
.Title {  
  
font-size: xx-large;  
  
font-weight: normal;  
  
padding: 0px;
```

```
margin: 0px;  
}
```

Ngôn ngữ của style bao gồm danh sách các thuộc tính được phân chia bằng dấu phẩy (,) và chứa trong ngoặc đơn ({..}). Các mã này được đặt bên trong tag <style> trong phần header, ví dụ như sau:

```
<!DOCTYPE html>  
  
<html lang="en">  
  
<head>  
  
<meta charset="utf-8" />  
  
<title>My Favorite Movies</title>  
  
<style type="text/css">  
  
.Title {  
  
font-size: xx-large;  
  
font-weight: normal;  
  
padding: 0px;  
  
margin: 0px;  
  
}  
  
</style>  
  
</head>  
  
<body>  
  
<header>  
  
<h1 class="Title">A list of my Favorite  
Movies</h1>  
  
</header>  
  
<div id="movieslist">
```



```
<ol>

<li><a href="#">It's a wonderful
life</a></li>

<li><a href="#">Lord of the Rings</a></li>

<li><a href="#">The Fourth World</a></li>

<li><a href="#">The Lion King</a></li>

</ol>

</div>

<footer>

This site was built using Microsoft
WebMatrix.

<a href="http://web.ms/webmatrix">Download
it now.</a>

</footer>

</html>
```

Khi hoạt động, các bạn sẽ thấy sự thay đổi khi áp dụng style:



Lưu ý rằng thẻ `<h1>` có class `Title`, vì vậy khi thiết lập `.Title`, bạn có thể áp dụng toàn bộ style với các thành phần bên trong có tên cùng với class này. Mỗi khi muốn thiết lập 1 thành phần cố định nào đó, bạn có thể sử dụng 1 class bất kỳ hoặc đặt tên thành phần đó theo id, sau đó là thiết lập class riêng theo id đó. Khi nhìn vào mã HTML, bạn sẽ thấy danh sách các

movie được nắm giữ bởi <div> - có id là moviesList. Bạn có thể thiết lập style cho lớp này bằng cách gán # trong phần định nghĩa, như sau:

```
#movieslist{  
  
font-family: Geneva, Tahoma, sans-serif;  
  
}
```

Quá trình này sẽ khởi tạo style mới cho thẻ <div>, và bất cứ đối tượng nào bên trong thẻ div này sẽ được áp dụng style này. Tại đây, mặc dù chúng ta chưa chỉ định bất cứ style nào đối với , nhưng kết quả vẫn được hiển thị như dưới đây:



Nhưng ở chế độ mặc định, trình duyệt sẽ tự động chia nhỏ các đối tượng trong thành danh sách với thứ tự. Và nếu muốn, chúng ta hoàn toàn có thể loại bỏ bằng cách lập style khác. Đơn giản, vì những đối tượng bên trong div được gọi là movieslist, chúng ta có thể dễ dàng đánh dấu chúng để thay đổi style cho phù hợp. Cú pháp được sử dụng như sau:

```
#movieslist ol {  
  
list-style: none;  
  
margin: 0;  
  
padding: 0;  
  
border: none;  
  
}
```

Ví dụ, đối với mỗi trong #movieslist, các bạn hãy làm sao cho trông không giống như 1 danh sách bình thường (không có dấu phân cách đầu dòng, không căn lề...). Và sau đây là kết quả:

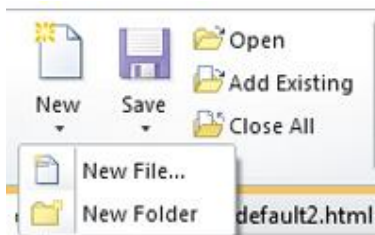


Các số thứ tự đầu dòng đã biến mất

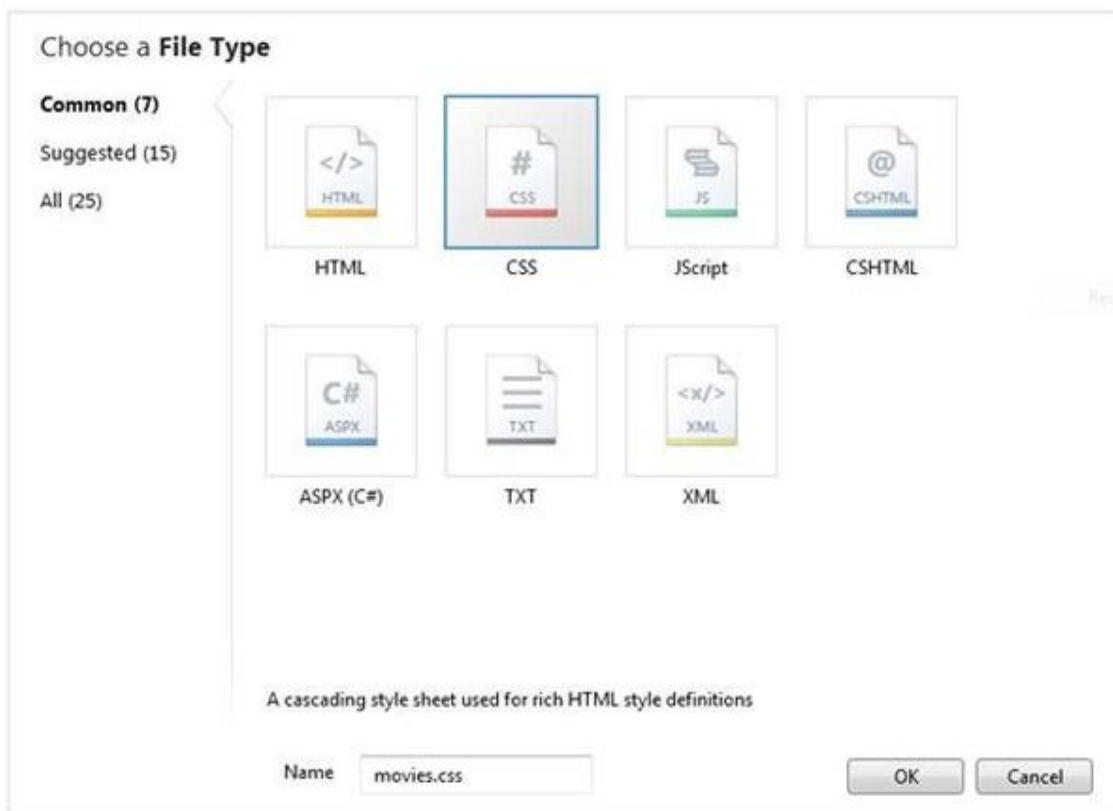
Các đoạn text của từng thành phần trong danh sách được xử lý bởi thẻ <a>, do vậy người sử dụng hoàn toàn có thể định nghĩa được giao diện hiển thị của từng tag <a> bên trong mỗi tag của #movieslist với cú pháp sau:

```
#movieslist li a {  
    font-size: large;  
  
    color: #000000;  
  
    display: block;  
  
    padding: 5px;  
  
}
```

Như vậy có thể thấy rằng, CSS là 1 trong những thành phần không thể thiếu trong bất kỳ trang web nào, nhưng web site của chúng ta không đơn thuần chỉ có 1 trang, vậy phải làm thế nào? Câu hỏi này khá thú vị, bởi vì CSS ở đây được tích hợp trong phần <head> sử dụng <script>. Lợi thế ở đây là <script> có thể sử dụng được các đoạn mã từ bất cứ nơi đâu, ở bên trong hoặc ngoài phạm vi của tag <script>. WebMatrix dễ dàng giúp chúng ta trong việc này, tại giao diện chính của chương trình, các bạn chọn *New > New File...* :



Cửa sổ *New Files* sẽ hiển thị, chọn tiếp *CSS* và đặt tên là **movies.css**:



Nhấn OK, các bạn sẽ có file *Movies.css* của riêng mình với thẻ `<body>` hoàn toàn trống rỗng như sau:

```
body {  
  
}
```

Thay thế bằng đoạn mã CSS dưới đây, với một số thay đổi về class để mang lại vẻ dễ nhìn hơn cho web site:

```
body {  
  
font-family: Tahoma, Verdana, Geneva, sans-serif;
```

```
width: 85%;
margin: 20px auto;
}
header {
padding: 10px;
border-bottom: 1px solid #e5e5e5;
}
header h1 {
font-size: xx-large;
font-weight: normal;
padding: 0px;
margin: 0px;
}
#movieslist{
margin: 20px 0;
}
#movieslist ul {
list-style: none;
margin: 0;
padding: 0;
}
#movieslist li a {
font-size: large;
```

```
color: #000000;

display: block;

padding: 5px;

}

#movieslist li a:hover {

border-left: 10px solid #94c9d4;

padding-left: 10px;

background-color: #e7f5f8;

text-decoration: none;

}
```

Quay trở lại mã HTML, xóa toàn bộ mã bên trong thẻ `<script>` và thay thế bằng:

```
<link rel="stylesheet" type="text/css"
href="movies.css" />
```

Đoạn mã trên có tác dụng yêu cầu trình duyệt tải CSS riêng biệt thay vì sử dụng mẫu có sẵn bên trong. Mã HTML của bạn lúc này sẽ trông giống như sau:

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<title>My Favorite Movies</title>

<link rel="stylesheet" type="text/css"
href="movies-html5.css" />

</head>

<body>
```

```
<header>

<h1>A list of my Favorite Movies</h1>

</header>

<div id="movieslist">

<ol>

<li><a href="#">It's a wonderful
life</a></li>

<li><a href="#">Lord of the Rings</a></li>

<li><a href="#">The Fourth World</a></li>

<li><a href="#">The Lion King</a></li>

</ol>

</div>

<footer>

This site was built using Microsoft
WebMatrix.

<a href="http://web.ms/webmatrix">Download
it now.</a>

</footer>

</html>
```

Phần Footer trông có vẻ không được “gọn gàng” cho lắm, hãy chỉnh sửa một vài thông tin trong CSS để sắp xếp lại, ví dụ như sau:

```
footer {

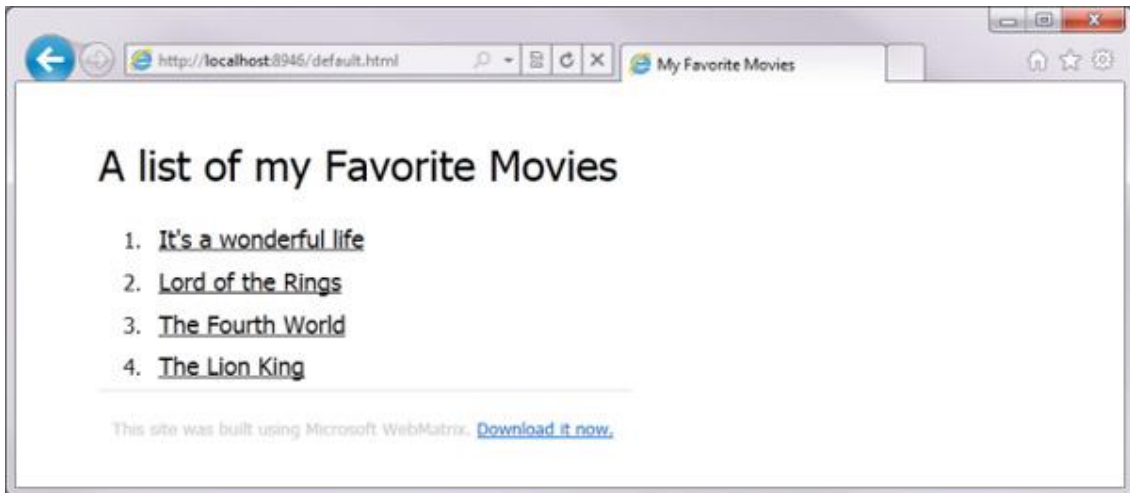
font-size: smaller;

color: #ccc;

text-align: center;
```

```
padding: 20px 10px 10px 10px;  
border-top: 1px solid #e5e5e5;  
}
```

Và dưới đây là kết quả sau khi chỉnh sửa:



Mỗi khi tới tag `<a>`, trình duyệt sẽ xử lý các hành động khác khi người dùng di chuột qua đó. Bạn có thể sử dụng CSS để thay đổi thuộc tính này dựa vào cú pháp: `#movieslist li a:hover{}`. Ví dụ như sau:

```
#movieslist li a:hover {  
  
border-left: 10px solid #94c9d4;  
}
```



```
padding-left: 10px;  
  
background-color: #e7f5f8;  
  
text-decoration: none;  
  
}
```

Đoạn mã trên sẽ gán thêm khoảng cách 10 pixel ở xung quanh bên trái và đổ màu vào phần nền để bôi đậm nội dung text. Và khi di chuột qua, bạn sẽ thấy sự thay đổi như sau:



Tìm hiểu về WebMatrix – phần 3

Trong phần tiếp theo của loạt bài kỹ thuật giới thiệu về WebMatrix, chúng ta sẽ cùng nhau tìm hiểu về những thành phần như: sử dụng Layout, dữ liệu, và trang có chức năng tạo thêm dữ liệu...

1. Sử dụng Layout:

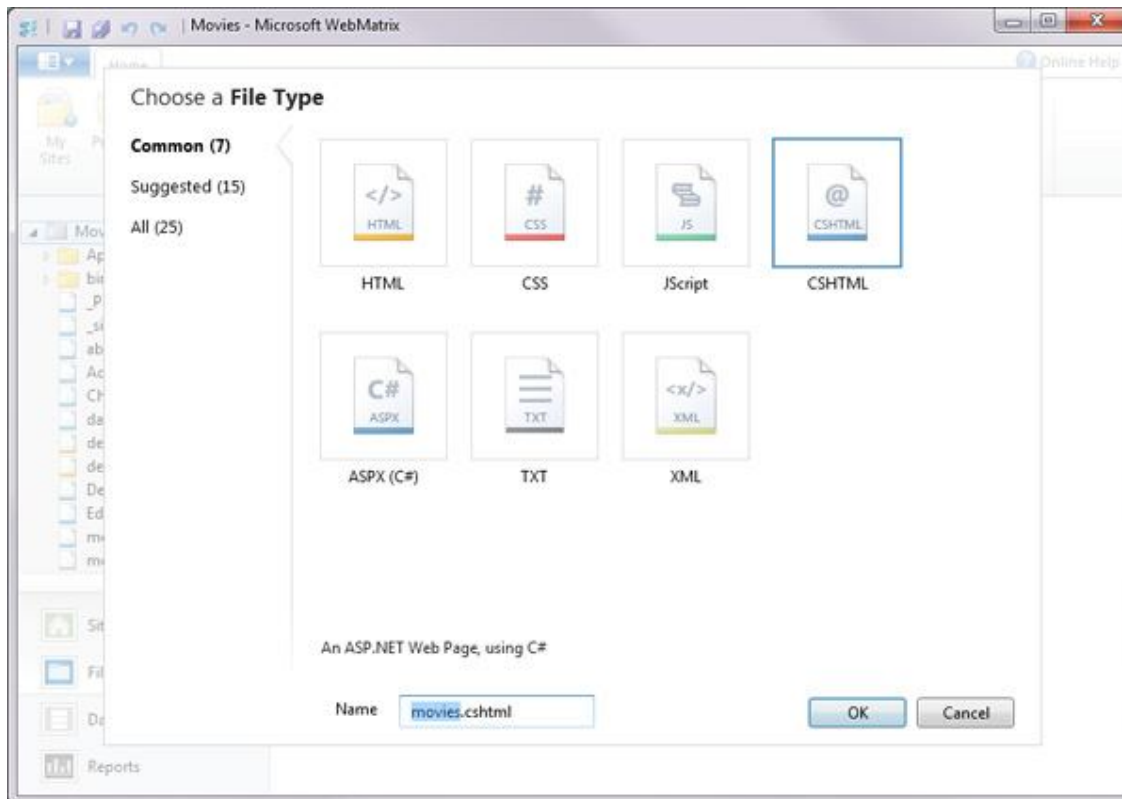
Trong phần [đầu](#), chúng ta đã tìm hiểu những bước cơ bản để tạo 1 trang web cơ bản bằng WebMatrix, và một số khái niệm cơ bản khác về CSS. Và tại đây, chúng ta sẽ chuyển sang bước tiếp theo, làm quen với khái niệm *server programming*, có thể mọi người đã quen với *client programming*, ví dụ như việc xây dựng chương trình hoạt động trên điện thoại, máy tính, hoặc ứng dụng JavaScript chạy bên trong trình duyệt. Điểm khác biệt quan trọng nhất với server programming là các mã nguồn không hoạt động trên thiết bị của client. Thay vào đó, các hành động từ phía người dùng sẽ gửi yêu cầu tới server, và nếu trang chứa thông tin yêu cầu đó được kích hoạt, server sẽ sử dụng các đoạn code và tạo mã HTML cùng với giá trị tương ứng gửi ngược lại về phía trình duyệt. Sau đó, trình duyệt sẽ xử lý và hiển thị kết quả với người dùng.

Đây là giai đoạn khá phức tạp, vì yêu cầu người dùng phải có một số kỹ năng nhất định về JavaScript, công nghệ *Rich Internet Application (RIA)* như Silverlight... *WebMatrix* cung cấp thêm cho người dùng cú pháp lệnh Razor để lập trình web, với khả năng hoạt động nhanh chóng, tính linh hoạt cao, các cơ chế layout engine đa dạng.

Tạo trang CSHTML sử dụng Razor:

Từ trước đến giờ, chúng ta chỉ làm quen với việc tạo trang HTML với phần mở rộng .HTM hoặc .HTML, đó đều là dạng trang web tĩnh, mỗi khi địa chỉ của chúng được gọi ra bởi trình duyệt, server chỉ làm nhiệm vụ chuyển toàn bộ nội dung và hiển thị, với cách làm này thì server hầu như không phải xử lý bất cứ yêu cầu nào. Có thể bạn đã quá quen với khái niệm trang web động, nghĩa là được server xây dựng dựa trên mã HTML cũng như loại mã ứng dụng trên server, để xác định cách thức tổng thể tạo trang web như thế nào. Các trang web động khá đa dạng và phong phú, và trong hầu hết phần còn lại của series chúng ta đều sử dụng chúng. Cụ thể, trong bài thử nghiệm của chúng ta, hệ thống cho phép người dùng lưu trữ dữ liệu về movie trong cơ sở dữ liệu, đồng thời server sẽ tạo nội dung hiển thị từ cơ sở dữ liệu nhập vào, thay vì phải điền trực tiếp nội dung của movie.

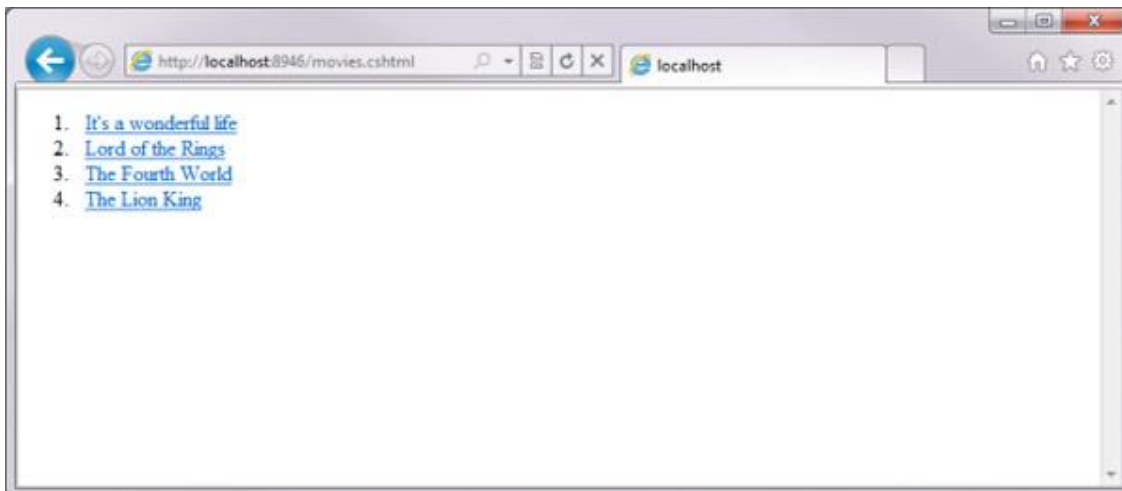
Với *WebMatrix*, các trang web động có phần mở rộng *.CSHTML* hoặc *.VBHTML*. Đây là những file *HTML* khá hiệu quả với phần lớn code bên trong viết bằng C# hoặc Visual Basic. Cụ thể, trong bài thử nghiệm của chúng ta, hãy tạo 1 file *CSHTML* mới và đặt tên là *movies.cshtml*:



WebMatrix sẽ tạo 1 trang trông như thường lệ, thay toàn bộ mã nguồn bằng đoạn code sau:

```
1: <div id="movieslist">
2: <ol>
3: <li><a href="#">It's a wonderful
   life</a></li>
4: <li><a href="#">Lord of the Rings</a></li>
5: <li><a href="#">The Fourth World</a></li>
6: <li><a href="#">The Lion King</a></li>
7: </ol>
8: </div>
```

Trông có vẻ hơi kỳ lạ, vì không có thẻ `<HTML>`, thẻ `<HEAD>` hoặc `<BODY>`, nhưng website vẫn hoạt động bình thường:



Phần Header và Footer:

Trang web bây giờ trông giống như phần trước đã làm trước kia, nhưng đến bước này chúng ta cần khởi tạo lại mục header trong HTML ở phía trước thẻ `<div>` có chứa danh sách movie. Các bạn không nên nhầm lẫn giữa 2 thẻ `<header>` và `<footer>` trong trang *default.html*.

Tạo 1 trang mới và đặt tên là *PageHeader.cshtml*, copy mọi thứ ngoài thẻ `<div>` của *moviesList* từ trang *default.html*. Toàn bộ đoạn mã của chúng ta sẽ trông giống như sau:

```
1: <!DOCTYPE html>
2: <html lang="en">
3: <head>
4: <meta charset="utf-8" />
5: <title>My Favorite Movies</title>
6: <link rel="stylesheet" type="text/css"
  href="movies-html5.css" />
7: </head>
8: <body>
```

```
9: <header>
10: <h1>A list of my Favorite Movies</h1>
11: </header>
```

Tương tự như vậy, hãy tạo 1 trang mới với tên *PageFooter.cshtml*, và copy mọi thứ bên dưới thẻ `<div>` trong phần `moviesList`:

```
1: <footer>
2: This site was built using Microsoft WebMatrix.
3: <a href="http://web.ms/webmatrix">Download it
  now.</a>
4: </footer>
5: </html>
```

Sử dụng Razor để thêm mục Header và Footer:

Giờ đây, bạn đã sẵn sàng để bắt đầu viết server code sử dụng Razor. WebMatrix phân biệt sự khác nhau giữa HTML và Razor bằng ký tự `@`, người sử dụng phải chèn thêm ký tự này vào tất cả các dòng code để chỉ định những hành động nào sẽ được server xử lý.

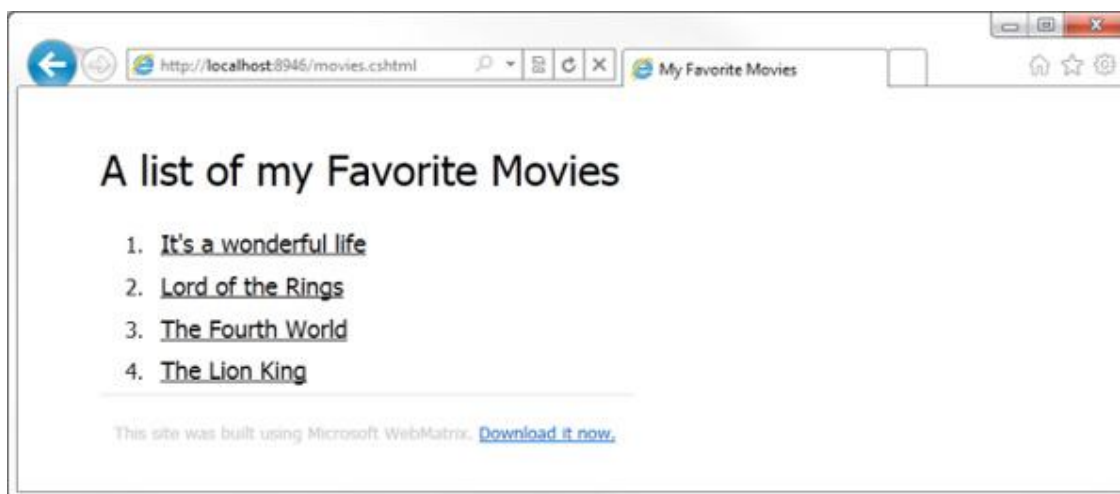
Ví dụ, `@RenderPage("pagename")` sẽ yêu cầu server tải mã HTML từ trang `pagename`. Do vậy, hãy sửa lại mã của trang `movies.cshtml` như sau:

```
1: @RenderPage("PageHeader.cshtml")
2: <div id="movieslist">
3: <ol>
4: <li><a href="#">It's a wonderful life</a></li>
5: <li><a href="#">Lord of the Rings</a></li>
6: <li><a href="#">The Fourth World</a></li>
7: <li><a href="#">The Lion King</a></li>
8: </ol>
```

```
9: </div>
```

```
10: @RenderPage( "PageFooter.cshtml" )
```

Khi chạy, website sẽ trông giống như sau:



Trông giống như trang HTML tĩnh trước kia, đơn giản bởi vì phần header và footer hoàn toàn giống nhau (bao gồm cả phần yêu cầu tải CSS), nhưng về bản chất thì quá trình này đơn giản hơn nhiều, vì bộ khung chính đã khá hoàn thiện, các trang mới sẽ sử dụng chung header, footer và style sheet.

Tạo trang Layout:

Khi làm theo phương pháp này, trước tiên chúng ta tạo phần cơ bản của trang web, sau đó sử dụng mã Razor để gọi phần header và footer trên toàn bộ site, đây là cách tiếp cận từ dưới lên. Bên cạnh đó, chúng ta còn một số cách làm khác, ví dụ như tạo trang layout trước để làm mẫu cho những phần tiếp theo, sau đó gán nội dung và chỉnh sửa giao diện. Đây là cách làm từ trên xuống.

Tại đây, chúng ta tiếp tục bằng cách tạo 1 trang mới *CSHTML* và đặt tên là *_siteLayout.cshtml*. Thay thế nội dung của trang với đoạn mã sau:

```
1: <!DOCTYPE html>
```

```
2: <html lang="en">
```

```
3: <head>
```

```
4: <meta charset="utf-8" />
```

```
5: <title>My Favorite Movies</title>
6: <link rel="stylesheet" type="text/css"
href="movies-html5.css" />
7: </head>
8: <body>
9: <header>
10: <h1>A list of my Favorite Movies</h1>
11: </header>
12: <div id="movieslist">
13: <ol>
14: <li><a href="#">It's a wonderful
life</a></li>
15: <li><a href="#">Lord of the
Rings</a></li>
16: <li><a href="#">The Fourth World</a></li>
17: <li><a href="#">The Lion King</a></li>
18: </ol>
19: </div>
20: <footer>
21: This site was built using Microsoft
WebMatrix.
22: <a
href="http://web.ms/webmatrix">Download it
now.</a>
23: </footer>
24: </html>
```

Sau đó, xóa bỏ đoạn mã `<div>` để gọi *movieslist* và thay thế bằng mã:

```
@RenderBody( )
```

Trang *_sitelayout.cshtml* sẽ trông giống như sau:

```
1: <!DOCTYPE html>
2: <html lang="en">
3: <head>
4: <meta charset="utf-8" />
5: <title>My Favorite Movies</title>
6: <link rel="stylesheet" type="text/css"
href="movies-html5.css" />
7: </head>
8: <body>
9: <header>
10: <h1>A list of my Favorite Movies</h1>
11: </header>
12: @RenderBody( )
13: <footer>
14: This site was built using Microsoft
WebMatrix.
15: <a
href="http://web.ms/webmatrix">Download it
now.</a>
16: </footer>
17: </html>
```

Khi hệ thống gửi yêu cầu đến trang *movies.cshtml* đã tạo trước đó, toàn bộ nội dung của trang lúc này sẽ nằm trong tag `<div>` và danh sách

. Về mặt kỹ thuật, chúng ta có thể suy luận theo cách sau, khi trình duyệt mở trang *CSHTML*, WebMatrix sẽ sử dụng trang layout để xác định nội dung sẽ được hiển thị theo cách nào, trước tiên là head, CSS và toàn bộ phần còn lại theo thứ tự. Nhưng trước khi áp dụng cách này, các bạn đừng quên bỏ lệnh *@RenderPage* tại trang *movies.cshtml*.

Thiết lập WebMatrix sử dụng trang layout:

Đến bước này, không ít bạn sẽ tự hỏi làm thế nào để sử dụng trang *_siteLayout.cshtml* thay vì *movies.cshtml*? Nếu bạn tạo 1 trang *_PageStart.cshtml*, thì trang này sẽ được gọi cho dù WebMatrix sử dụng CSTHTML hoặc VBHTML. Để tiếp tục, tạo trang *_SiteLayout.cshtml* và thay thế mã với đoạn code sau:

```
1: @{
2:     Layout = "~/_SiteLayout.cshtml";
3: }
```



Chúng ta hãy cùng nhìn lại từng bước WebMatrix thực hiện để chạy website:

- Trước tiên, trình duyệt gửi yêu cầu tới server về trang *movies.cshtml*
- Server kiểm tra thấy có sự tồn tại của file *_PageStart.cshtml* và kích hoạt nó. Chỉ có 1 dòng code, chỉ định rằng biến Layout – đại diện cho trang hiển thị bên ngoài là *SiteLayout.cshtml*

- Sau đó, server sẽ xử lý file *SiteLayout.cshtml* để tạo trang hiển thị, file này cung cấp phần head, hệ thống css, tag `<body>`... và tất cả những thứ cần thiết đối với trang HTML tĩnh
- Tiếp tục, server tìm kiếm các dòng lệnh bắt đầu bằng `@RenderBody()`, chỉ định toàn bộ nội dung của trang *movies.cshtml*
- Server sẽ xử lý toàn bộ phần mã còn lại của *SiteLayout.cshtml*, hoàn thiện thẻ `</body>`, `</html>` và footer
- Toàn bộ trang web hoàn chỉnh sẽ được trả về giao diện của trình duyệt.

Thêm trang mới:

Khi áp dụng theo cách này, chúng ta có thể thêm một trang mới bất kỳ một cách đơn giản và dễ dàng. Như ví dụ tại đây, các bạn tạo trang *about.cshtml* và thay bằng mã sau:

```
<h1>About me</h1>
```

```
<h2>I'm the author of this page. I should put something interesting here.</h2>
```

Đối với trang *About.cshtml* này, toàn bộ quá trình xử lý sẽ được lặp lại như trước, ngoại trừ khi trang *_siteLayout.cshtml* đọc thông tin từ phần `@RenderBody()`, kết quả sẽ hiển thị như sau:



Các bạn có thể dễ dàng nhận ra toàn bộ phần header và footer được giữ nguyên, không có gì thay đổi. Người sử dụng sẽ tiết kiệm được rất nhiều thời gian và công sức so với cách làm thủ công thông thường, mà không cần phải chỉnh sửa nhiều thông tin. Trong phần tiếp theo, chúng ta sẽ bắt

Simpo PDF Merge and Split Unregistered Version - <http://www.simpopdf.com>

tay vào quá trình tạo dữ liệu với ứng dụng bằng tính năng tích hợp sẵn trong *WebMatrix*. Chúc các bạn thành công!