



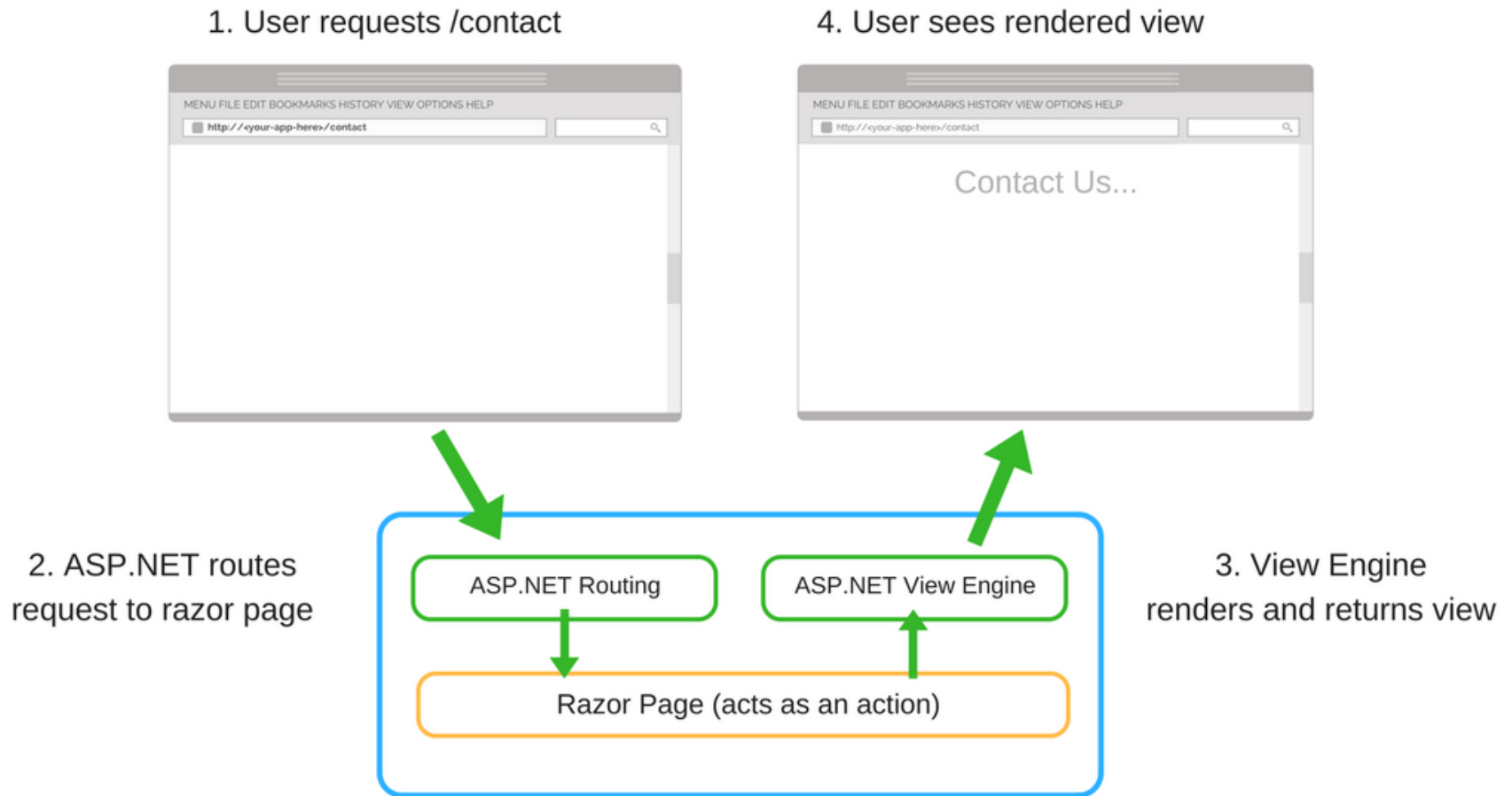
LẬP TRÌNH C# 4

BÀI 6: RAZOR VIEW - TAG HELPERS (P1)

- ◎ Razor View
- ◎ Tag Helpers



- ❑ Razor giúp dễ dàng nhúng C# code vào Html và cung cấp khả năng tạo ra response động.
- ❑ Razor View Engine là View Engine mặc định của ASP.NET Core. Nó lấy mã Razor trong file View và chuyển sang HTML response
- ❑ Cú pháp Razor thường ngắn hơn và đơn giản hơn, cũng dễ học như C# hoặc VB. Visual Studio IntelliSense cũng hỗ trợ cú pháp Razor.
- ❑ Razor Pages là tập tin có phần mở rộng cshtml



❑ Razor sử dụng ký tự @ để chuyển HTML sang C#, có 2 cách khai báo dùng razor:

- ❖ Razor code blocks @{.....}: Khối lệnh Razor bắt đầu bởi @ và nằm trong cặp ngoặc nhọn. Một khối Razor có thể sử dụng để thao tác Model, khai báo biến, đặt thuộc tính của View. Tuy nhiên nó không nên sử dụng cho việc xử lý logic. Ví dụ file Index.cshtml:

```
<!-- Single statement block -->
@{ var myMessage = "Hello Fpoly"; }
<!-- Multi-statement block -->
@{
    var greeting = "Welcome to Fpoly's Razor";
    var weekDay = DateTime.Now.DayOfWeek;
    var greetingMessage = greeting + " Here in Fpoly it is: " + weekDay;
}
```

❑ Dùng Razor Code Expressions

❖ Implicit Razor Expressions:

- Implicit Razor Expressions không cho phép dấu cách vì dấu cách dùng để kết thúc biểu thức.
- Các biểu thức được xử lý bởi Razor View Engine và kết quả được thêm ngay vào vị trí nó đặt

```
<!-- Single statement block -->  
@{ var myMessage = "Hello Fpoly"; }  
<!-- Multi-statement block -->
```

```
@{  
    var greeting = "Welcome to Fpoly's Razor";  
    var weekDay = DateTime.Now.DayOfWeek;  
    var greetingMessage = greeting + " Here in Fpoly it is: " + weekDay;  
}
```

```
<h3>Code Expression</h3>  
<p>@greeting</p>  
<p>@DateTime.Now</p>  
<p>Today is : @weekDay thank you </p>
```

Code Expression

Welcome to Fpoly's Razor

11/03/2020 3:19:21 PM

Today is : Wednesday thank you

❑ Dùng Razor Code Expressions

❖ Explicit Razor Expressions:

- Explicit Razor Expressions bắt đầu bằng dấu @ và theo sau bởi cặp ().
- Bất cứ nội dung nào trong cặp ngoặc đơn này đều được xử lý bởi Razor và tạo ra output.

```
<!-- Single statement block -->
```

```
@{ var myMessage = "Hello Fpoly"; }
```

```
<!-- Multi-statement block -->
```

```
@{
```

```
var greeting = "Welcome to Fpoly's Razor";
```

```
var weekDay = DateTime.Now.DayOfWeek;
```

```
var greetingMessage = greeting + " Here in Fpoly it is: " + weekDay;
```

```
}
```

```
</h3>Code Expression</h3>
```

```
<p>greeting : @(greeting)</p>
```

Code Expression

greeting : Welcome to Fpoly's Razor

- ❑ Trong razor các biến được khai báo sử dụng từ khóa var hoặc sử dụng kiểu dữ liệu C#
- ❑ Các kiểu như int, float, decimal, bool, DateTime hay string có thể được dùng để lưu trữ kiểu dữ liệu tương ứng.

```
<h3>Variables </h3>  
<!-- Storing a string -->  
@{ var message = "Welcome to our website"; }  
<!-- Storing a date -->  
@{ DateTime date = DateTime.Now; }  
<p>@message</p>  
<p>The current date is @date</p>
```


- ❑ Các chuỗi được bao trong một cặp nháy kép. Để sử dụng nháy kép trong chuỗi cần phải nhân đôi dấu nháy

```
@{ var helloWorld = @"Hello ""World"""; }  
<p>@helloWorld</p>
```

- ❑ In ký tự @ trong HTML bằng cách lặp 2 ký tự@

```
@{ var symbol = "You can print @ in html"; }  
<p>The @@symbol is: @symbol</p>
```

- ❑ Dấu @ trong email thì có thể chỉ đích danh nó và không cần phải double

```
<a href="mailto:admin@fpoly.com">admin@fpoly.com</a>
```

- ❑ Comment: sử dụng `@**@` để đặt comment
- ❑ HTML trong khối lệnh: bất cứ HTML Element nào trong khối lệnh Razor được nhận dạng như bình thường
- ❑ Dòng lệnh đơn: xuất ra giá trị mà không cần thẻ HTML, dùng bằng dấu `@`
- ❑ Đa dòng lệnh: đối với nhiều dòng bạn cần phải có thẻ `<text> </text>`

```
Using Html in code block razor
```

```
Hello Single line text
```

```
Hello from the multiline text
```

Lập trình C#4

```
@{  
    <p>Using Html in code block razor</p>  
    @*This is comment*@  
}  
@:  
Hello Single line text  
}  
@{  
    <text>Hello from the  
    multiline  
    text </text>  
}
```

- ❑ Kết hợp sử dụng câu lệnh điều kiện If else theo cú pháp razor với ký tự @

```
@{var price = 25;}  
<html>  
<body>  
    @if (price >= 30)  
    {  
        <p>The price is high.</p>  
    }  
    else if (price > 20 && price < 30)  
    {  
        <p>The price is OK.</p>  
    }  
    else  
    {  
        <p>The price is low.</p>  
    }  
</body>  
</html>
```

- ❑ Kết hợp sử dụng câu lệnh điều kiện Switch theo cú pháp razor với ký tự @

```
@{  
    var weekday = DateTime.Now.DayOfWeek;  
    var day = weekday.ToString();  
    var message = "";  
}  
<html>  
<body>  
    @switch (day)  
    {  
        case "Monday":  
            message = "This is the first weekday.";  
            break;  
        case "Thursday":  
            message = "Only one day before weekend.";  
            break;  
        case "Friday":  
            message = "Tomorrow is weekend!";  
            break;  
        default:  
            message = "Today is " + day;  
            break;  
    }  
    <p>@message</p>  
</body>  
</html>
```

□ Lặp foreach

```
@{ var numbers = Enumerable.Range(1, 10); }
```

```
@foreach (var number in numbers)  
{  
    @number  
}
```

□ For Loops

□ While Loops

```
<html>  
<body>  
@{  
    var i = 0;  
    while (i < 5)  
    {  
        i += 1;  
        <p>Line @i</p>  
    }  
}  
</body>  
</html>
```

```
<html>  
<body>  
    @for(var i = 10; i < 21; i++)  
        {<p>Line @i</p>}  
</body>  
</html>
```

□ Ví dụ loops và mảng

```
@{
    string[] members = { "HN", "Hcm", "ĐN", "TN", "CT" };
    int i = Array.IndexOf(members, "ĐN") + 1;
    int len = members.Length;
    string x = members[2 - 1];
}

<html>
<body>
    <h3>Members</h3>
    @foreach (var person in members)
    {
        <p>@person</p>
    }
    <p>The number of names in Members are @len</p>
    <p>The person at position 2 is @x</p>
    <p>Kai is now in position @i</p>
</body>
</html>
```

Members

HN

Hcm

ĐN

TN

CT

The number of names in Members are 5

The person at position 2 is Hcm

ĐN is now in position 3

❑ Ví dụ loops và danh sách đối tượng

- ❖ Tạo models:
 - ❖ Khai báo sử dụng model và
- Viết code razor:

`@using razor.Models`

`@{`
`var custList = new List<Customer>()`

`{`
`new Customer() { name = "FpolyHN", address = "Tòa nhà FPT Polytechnic`
`new Customer() { name = "FpolyHcm", address = "391A Nam Kỳ Khởi Nghĩa`
`};`
`}`

`<table>`
`<thead>`
`<tr><td>Name</td><td>Address</td></tr>`
`</thead>`

`@foreach (Customer custvar in custList)`
`{`

`<tr>`
`<td>@custvar.name</td>`
`<td>@custvar.address</td>`
`</tr>`
`}`

`</table>`

```
namespace razor.Models
{
    4 references
    public class Customer
    {
        3 references | 0 exceptions
        public string name { get; set; }
        3 references | 0 exceptions
        public string address { get; set; }
    }
}
```



DEMO

- Hiện thực các ví dụ





LẬP TRÌNH C# 4

BÀI 6: RAZOR VIEW - TAG HELPERS (P2)

- ❑ TagHelper là loại thẻ (tag) đặc biệt giúp viết phần tử HTML trong Razor sử dụng cú pháp thân thiện với HTML và tận dụng được các ưu điểm của việc xử lý phía server.
- ❑ Ví dụ, sử dụng thẻ Form Tag Helper, có thể tạo ra thẻ `<form>` trong html:

```
<form asp-action="create" asp-controller="home">
```

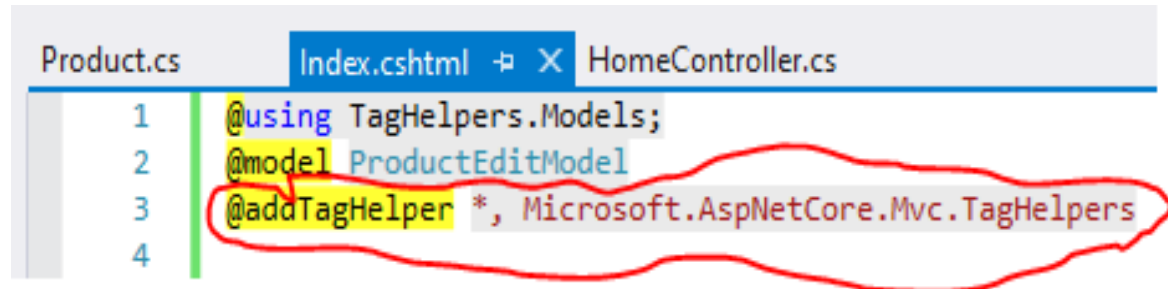


```
<form action="/home/create" method="post">
```

- ❑ Tag Helper sẽ giúp tạo ra view HTML dựa trên dữ liệu từ Model gắn vào nó
- ❑ Ví dụ
 - ❖ Label Tag Helper sẽ tạo ra tiêu đề dựa trên attribute Data Annotation trong View Model.
 - ❖ Input Tag Helper sẽ tạo ra id, name, type của phần tử HTML dựa trên kiểu dữ liệu của Model và thuộc tính Data Annotation.

```
<label asp-for="@Model.Name"></label>
```

- ❑ Tag Helper nằm trong thư viện Microsoft.AspNetCore.Mvc.TagHelpers
- ❑ Cần thêm khai báo @addTagHelper vào view muốn sử dụng



The screenshot shows a code editor with three tabs: Product.cs, Index.cshtml (selected), and HomeController.cs. The Index.cshtml file contains the following code:

```
1 @using TagHelpers.Models;  
2 @model ProductEditModel  
3 @addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers  
4
```

The line `@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers` is highlighted in yellow and circled in red.

- ❑ Thêm @addTagHelper vào một view nào đó chỉ có tác dụng trên view đó. Bạn có thể thêm @addTagHelper vào **_ViewImports.cshtml** để sử dụng Tag Helper trên toàn bộ các view của ứng dụng

- ❑ Có thể bỏ Tag Helper: @removeTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
- ❑ Thêm một số Tag Helper chỉ định thay vì dùng hết:

```
@addTagHelper "Microsoft.AspNetCore.Mvc.TagHelpers.InputTagHelper, Microsoft.AspNetCore.Mvc.TagHelpers"
```

- ❑ Sử dụng ký tự ! trước mỗi phần tử HTML có thể vô hiệu hóa tag helper cho phần tử đó

```
<!label asp-for="Name"></!label>
```

- ❑ Sử dụng @tagHelperPrefix để bật Tag Helper

```
@tagHelperPrefix th:
```

Giờ thì tiền tố **th**: phải được chỉ ra cho tất cả các tag helper trên view, để bật tag helper cho nó:

```
<th:label asp-for="Name"></th:label>    //Tag helper is enabled  
  
<label asp-for="Address"></label>        //Tag helper is disabled
```

- ❑ Ví dụ có ProductEditMode model như hình:
- ❑ Xây dựng controller HomeController Có action Index trả ProductEditModel về cho view

```
public class HomeController : Controller
{
    [HttpGet]
    0 references | 0 changes | 0 authors, 0 changes | 0 requests | 0 exceptions
    public IActionResult Index()
    {
        ProductEditModel model = new ProductEditModel();
        return View(model);
    }
}
```

```
public class Product
{
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public int ProductId { get; set; }
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public string Name { get; set; }

    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public Decimal Price { get; set; }
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public int Rating { get; set; }
}

4 references | 0 changes | 0 authors, 0 changes
public class ProductEditModel
{
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public int ProductId { get; set; }

    [Required(ErrorMessage = "Product Name is Required")]
    [Display(Name = "Product Name")]
    2 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public string Name { get; set; }

    2 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public Decimal Price { get; set; }
    2 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public int Rating { get; set; }
}
```

- ❑ Tạo view index sử dụng TagHelper hiển thị form cho phép người dùng nhập Name, Rating và Price

```

Index.cshtml*  HomeController.cs*
<!-- Khai báo dùng model-->
@using TagHelpers.Models;
@model ProductEditModel
<!-- khai báo dùng Taghelper-->
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers

@{
    ViewData["Title"] = "Index";
}

<h1>Vi du TagHelper</h1>

<form asp-controller="Home" asp-action="Index">

    <!--<label asp-for="Model.Name"></label-->
    <label asp-for="Name"></label>
    <input asp-for="Name" />

    <label asp-for="Rating"></label>
    <input asp-for="Rating" />
    <label asp-for="Price"></label>
    <input asp-for="Price" />
    <input type="submit" name="submit" />

</form>

```



Vi du TagHelper

Product Name Rating Price

- ❑ Form Tag Helper được bao bởi thẻ `<form>` và cung cấp một số thuộc tính phía server giúp tạo ra HTML
 - ❖ `asp-controller`: Chỉ ra tên Controller sử dụng
 - ❖ `asp-action`: Chỉ ra tên action method sử dụng
 - ❖ `asp-area`: Chỉ ra tên Area sử dụng

```
<form asp-controller="Home" asp-action="Create">
```

Đoạn code trên sẽ biên dịch ra HTML thuần là:

```
<form action="/Home/Create" method="post">  
  <input name="__RequestVerificationToken" type="hidden" value=  
    "CfDJ8PWn168XJhBDiaOWJw_ICJPDeJ9HYn8Hsawqaa6xDTM2hxBSxbKTmSHKzXIb  
    It_9hSD41N11rWkmdxpOsw7jdAOG4VeX5bfS26U7dGrwjtL3Jehi0Df16v07p580U  
    8B8l0Wi4EpmcgxTf3mM0Ezcj3s">  
</form>
```


Label Tag Helper

Label Tag Helper được áp dụng cho phần tử label. Nó có một thuộc tính là asp-for. Sử dụng như sau:

```
<label asp-for="@Model.Name"></label>
```

Nó sẽ dịch ra:

```
<label for="Name">Name</label>
```

Input Tag Helper

Tương tự, Input Tag Helper cũng được áp dụng cho phần tử input:

```
<input asp-for="Name" />
```

Nó sẽ dịch ra:

```
<input type="text" id="Name" name="Name" value="" />
```

Thuộc tính type, id và name tự động lấy từ tên và kiểu dữ liệu của trường đó trong Model.

□ Tìm hiểu taghelper tạo ra html

```

Index.cshtml*  HomeController.cs*
<!-- Khai báo dùng model-->
@using TagHelpers.Models;
@model ProductEditModel
<!-- khai báo dùng Taghelper-->
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers

@{
    ViewData["Title"] = "Index";
}

<h1>Vi du TagHelper</h1>

<form asp-controller="Home" asp-action="Index">

    <!--<label asp-for="Model.Name"></label>-->
    <label asp-for="Name"></label>
    <input asp-for="Name" />

    <label asp-for="Rating"></label>
    <input asp-for="Rating" />
    <label asp-for="Price"></label>
    <input asp-for="Price" />
    <input type="submit" name="submit" />

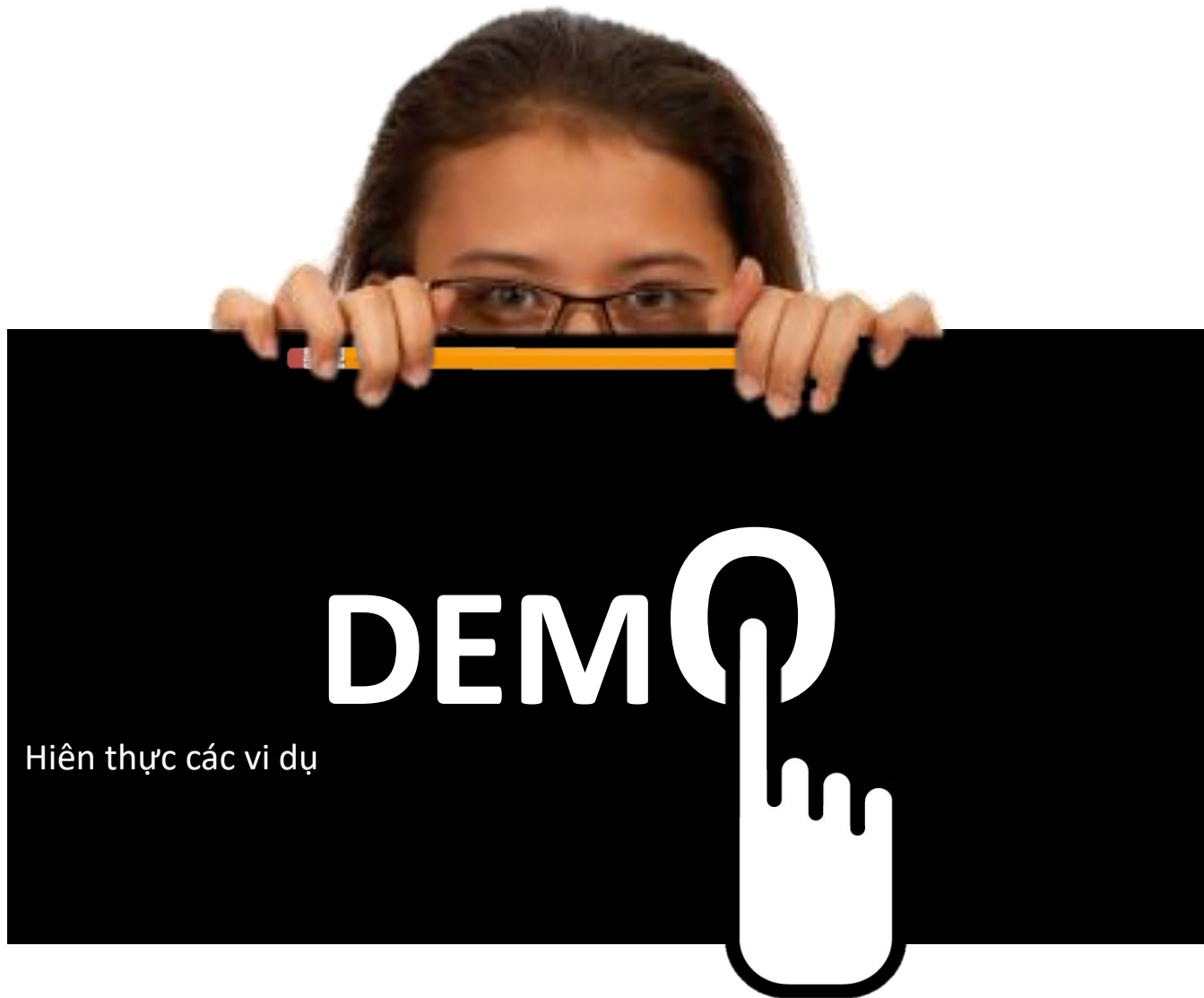
</form>
        
```

```

<form action="/" method="post">
    <!--<label asp-for="Model.Name"></label>-->
    <label for="Name">Product Name</label>
    <input type="text" data-val="true" data-val-required="Product
Name is Required" id="Name" name="Name" value>
    <label for="Rating">Rating</label>
    <input type="number" data-val="true" data-val-required="The
Rating field is required." id="Rating" name="Rating" value="0">
    <label for="Price">Price</label>
    <input type="text" data-val="true" data-val-number="The field
Price must be a number." data-val-required="The Price field is
required." id="Price" name="Price" value="0.00">
    <input type="submit" name="submit">
    <input name="__RequestVerificationToken" type="hidden" value=
"CfDJ8PWn168XJhBDiaOWJw_ICJ0gLFn5Ljn0LyyS0FCXCxTTLbqTt9YoABTE7HFU
Sab0-
uh8qeOPv1QiaUytHAJ41tAqBgBLhb_wiQZhyfOQB6Ti8suovCSDvz3nOYHmpafnWR
2iX4JKgHNs6juU4v7orqQ">
</form>
        
```

❑ Danh sách các Tag Helper có sẵn

TAG HELPER	TẠO RA	THUỘC TÍNH
Form Tag Helper	<form>	asp-action, asp-all-route-data, asp-area, asp-controller, asp-fragment, asp-host, asp-page, asp-page-handler, asp-protocol, asp-route, asp-route-
Anchor Tag Helpers	<a>	asp-action, asp-all-route-data, asp-area, asp-controller, asp-Fragment, asp-host, asp-page, asp-page-handler, asp-Protocol, asp-route, asp-route-
Cache Tag Helper	<cache>	enabled1, expires-after2, expires-on3, expires-sliding4, priority5, vary-by6
Environment Tag Helper	<environment>	names, include, exclude
Image Tag Helper		append-version
Input Tag Helper	<input>	for
Label Tag Helper	<label>	for
Link Tag Helper	<link>	href-include, href-exclude, fallback-href, fallback-href-include, fallback-href-exclude, fallback-test-class, fallback-test-value, fallback-test-property, fallback-test-value, append-version
Options Tag Helper	<select>	asp-for, asp-items
Partial Tag Helper	<partial>	name, model, for, view-data
Script Tag Helper	<script>	src-include, src-exclude, fallback-src, fallback-src-include, fallback-src-exclude, fallback-test, append-version
Select Tag Helper	<select>	for, items
Textarea Tag Helper	<textarea>	for
Validation Message Tag Helper		validation-for
Validation Summary Tag Helper	<div>	validation-summary



Tổng kết bài học

◎ Razor View

◎ Tag Helpers





KẾT THÚC