

BÀI 3: SỬ DỤNG LAYOUT, RAZOR, HTML HELPER

Nội dung bài học:

1. Layout của website.....	2
1.1. Layout mặc định khi tạo web mvc 5	2
1.2. Tự làm 1 layout cho website	7
2. Razor View	11
2.1. Razor là gì?	11
2.2. Các ví dụ:.....	11
3. HTML Helpers in ASP.NET MVC	14
3.1. Standard HTML helper trong Asp.NET MVC5	14
3.2. Cách sử dụng	14
3.3. Strongly-Typed HTML Helper.....	17
3.4. Templated HTML Helpers.....	17
4. Sử dụng Standard HTML helper thiết kế form	18
Bài tập 1. Sử dụng Razor, viết code thực hiện yêu cầu như hình sau:.....	26
Bài tập 2. Sử dụng HTML Helpers, thiết kế form theo mẫu:.....	26

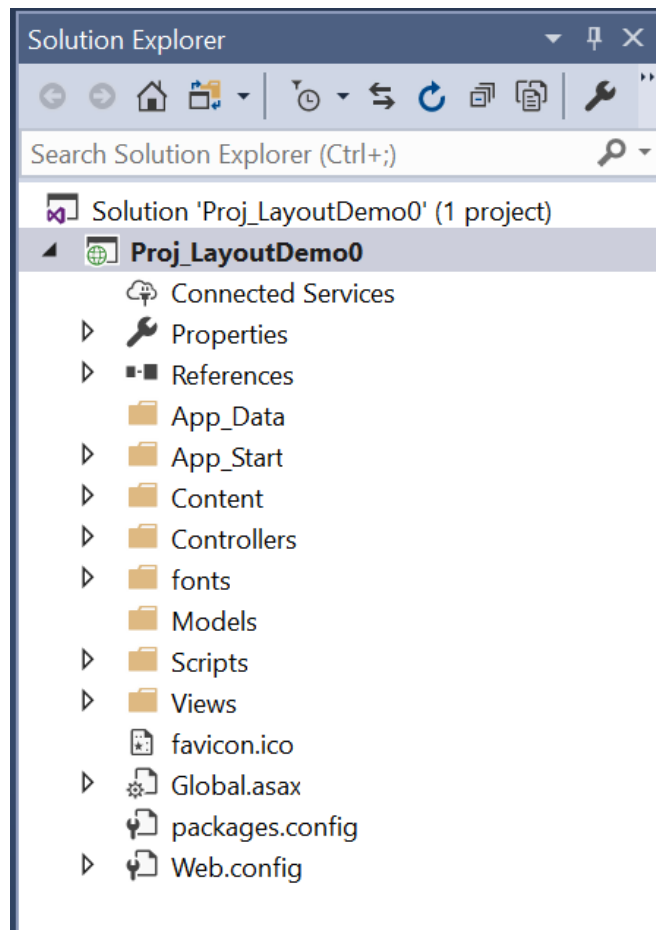
1. Layout của website

1.1. Layout mặc định khi tạo web mvc 5

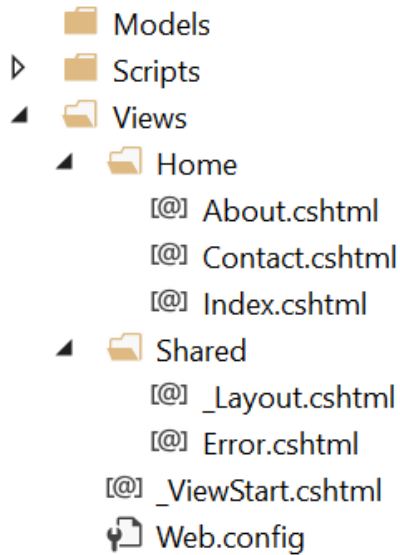
Trong một website, chúng ta thấy các page luôn giống nhau ở những vị trí như header, footer, body ... Thông thường, chỉ những nội dung ở giữa trang là thay đổi.

Vậy làm thế nào để không phải thiết kế lặp lại những phần chung của website. **ASP.Net MVC** 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.

Khi tạo mới Project MVC 5, Vision Studio đã tạo sẵn cho chúng ta cấu trúc thư mục:

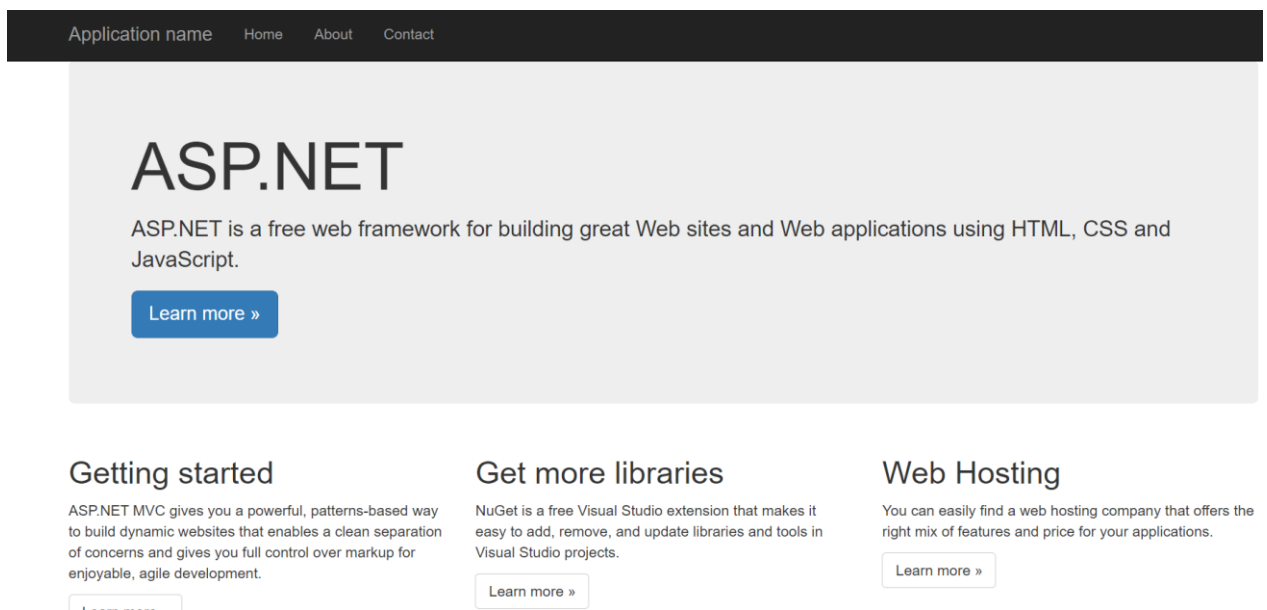


Quan sát các thư mục và file trong thư mục Views



File Index.cshtml trong thư mục Home được đặt mặc định là trang chạy đầu tiên của website.

Khi chạy chương trình, màn hình xuất hiện:



Click vào About:

Application name Home About Contact

About.

Your application description page.

Use this area to provide additional information.

© 2021 - My ASP.NET Application

Click vào Contact

Application name Home About Contact

Contact.

Your contact page.

One Microsoft Way
Redmond, WA 98052-6399
P: 425.555.0100

Support: Support@example.com
Marketing: Marketing@example.com

© 2021 - My ASP.NET Application

Chúng ta nhận thấy phần header, link, footer là giống nhau giữa các trang.

VS đã tạo sẵn cho chúng ta file `_Layout.cshtml`, chính là file chứa giao diện của website.

Nội dung file `Layout.cshtml`:

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>@ViewBag.Title - My ASP.NET Application</title>
7      @Styles.Render("~/Content/css")
8      @Scripts.Render("~/bundles/modernizr")
9  </head>
10 <body>
11     <div class="navbar navbar-inverse navbar-fixed-top">
12         <div class="container">
13             <div class="navbar-header">
14                 <button type="button" class="navbar-toggle"
15                     data-toggle="collapse" data-target=".navbar-collapse">
16                     <span class="icon-bar"></span>
17                     <span class="icon-bar"></span>
18                     <span class="icon-bar"></span>
19                 </button>
20                 @Html.ActionLink("Application name", "Index", "Home",
21                     new { area = "" }, new { @class = "navbar-brand" })
22             </div>
23
24             <div class="navbar-collapse collapse">
25                 <ul class="nav navbar-nav">
26                     <li>@Html.ActionLink("Home", "Index", "Home")</li>
27                     <li>@Html.ActionLink("About", "About", "Home")</li>
28                     <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
29                 </ul>
30             </div>
31         </div>
32     <div class="container body-content">
33         @RenderBody()
34         <hr />
35         <footer>
36             <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
37         </footer>
38     </div>
39
40     @Scripts.Render("~/bundles/jquery")
41     @Scripts.Render("~/bundles/bootstrap")
42     @RenderSection("scripts", required: false)
43 </body>
44 </html>
45

```

Trong đó, chúng ta quan tâm tới các thẻ sau:

a. @Style.Render

```
@Styles.Render("~/Content/css")
@Scripts.Render("~/bundles/modernizr")
```

@Styles.Render () là liên kết tới đến các cấu hình gói. Chúng ta có thể xem định nghĩa gói trong YourWebsite/packages.config. Nội dung file config này như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<packages>
  <package id="Antlr" version="3.5.0.2" targetFramework="net461" />
  <package id="bootstrap" version="3.3.7" targetFramework="net461" />
  <package id="jQuery" version="3.3.1" targetFramework="net461" />
  <package id="jQuery.Validation" version="1.17.0" targetFramework="net461" />
  <package id="Microsoft.AspNet.Mvc" version="5.2.4" targetFramework="net461" />
  <package id="Microsoft.AspNet.Razor" version="3.2.4" targetFramework="net461" />
  <package id="Microsoft.AspNet.Web.Optimization" version="1.1.3"
targetFramework="net461" />
  <package id="Microsoft.AspNet.WebPages" version="3.2.4" targetFramework="net461" />
  <package id="Microsoft.CodeDom.Providers.DotNetCompilerPlatform" version="2.0.0"
targetFramework="net461" />
  <package id="Microsoft.jQuery.Unobtrusive.Validation" version="3.2.4"
targetFramework="net461" />
  <package id="Microsoft.Web.Infrastructure" version="1.0.0.0"
targetFramework="net461" />
  <package id="Modernizr" version="2.8.3" targetFramework="net461" />
  <package id="Newtonsoft.Json" version="11.0.1" targetFramework="net461" />
  <package id="WebGrease" version="1.6.0" targetFramework="net461" />
</packages>
```

b. ActionLink

```
<li>@Html.ActionLink("Home", "Index", "Home")</li>
<li>@Html.ActionLink("About", "About", "Home")</li>
<li>@Html.ActionLink("Contact", "Contact", "Home")</li>
```

Là link tới các action và view.

@ActionLink("Text Link","Action Name","Controller Name")

c. @RenderBody()

@RenderBody được sử dụng để đưa nội dung từ trang con vào thiết kế trang chính. Nếu không có phần được đặt tên trong trang con, nội dung sẽ được hiển thị trong phần RenderBody.

d. @Scripts.Render

Là gói tất cả javascript hoặc style sheet thành một file không có định dạng (còn được gọi là rút gọn) để tiết kiệm băng thông và số lượng request một trang.

File About.cshtml có nội dung:

```

1  @{
2      ViewBag.Title = "About";
3  }
4  <h2>@ViewBag.Title.</h2>
5  <h3>@ViewBag.Message</h3>
6
7  <p>Use this area to provide additional information.</p>
8

```

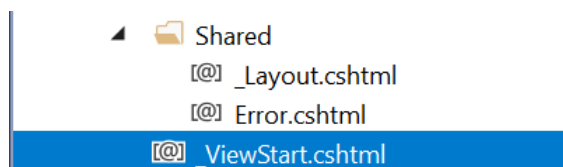
File Contact.cshtml có nội dung:

```

1  @{
2      ViewBag.Title = "Contact";
3  }
4  <h2>@ViewBag.Title.</h2>
5  <h3>@ViewBag.Message</h3>
6
7  <address>
8      One Microsoft Way<br />
9      Redmond, WA 98052-6399<br />
10     <abbr title="Phone">P:</abbr>
11     425.555.0100
12 </address>
13
14 <address>
15     <strong>Support:</strong> <a href="mailto:Support@example.com">Support@example.com</a><br />
16     <strong>Marketing:</strong> <a href="mailto:Marketing@example.com">Marketing@example.com</a>
17 </address>

```

Như vậy chúng ta thấy trong 2 file này không hề có đoạn code liên kết tới trang _Layout.cshtml; nhưng chúng ta vẫn thấy nội dung hiển thị vẫn đầy đủ header, link, footer.



File _ViewStart.cshtml chứa mặc định Layout được áp dụng cho view. Nội dung của file này là:

```

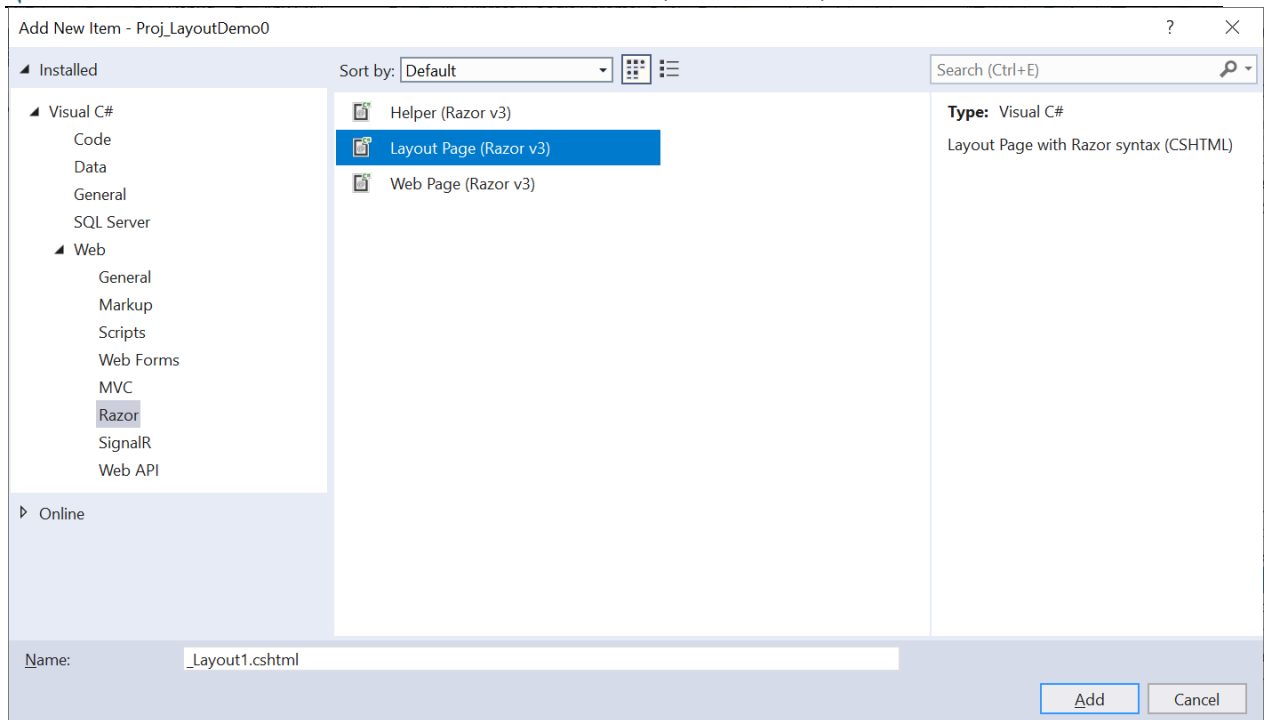
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}

```

1.2. Tự làm 1 layout cho website

Tạo một Layout mới, đặt tên là _Layout1.cshtml như hình: *Click chuột phải View/*

Shared, chọn Add/New Item



Xuất hiện đoạn code:

```
<!DOCTYPE html>
<html>
<head>
    <title>@Page.Title</title>
    @RenderSection("head", required: false)
</head>
<body>
    @RenderBody()
</body>
</html>
```

Chúng ta chỉnh sửa một chút:

```
<!DOCTYPE html>
<html>
<head>
    <title>@Page.Title</title>
    @RenderSection("head", required: false)
</head>
<body>
    <h1>Hanoi University of Industry</h1>
    @Html.ActionLink("Kỹ thuật lập trình", "Kythuat", "fit") <br />
    @Html.ActionLink("Hệ thống thông tin", "Hethong", "fit") <br />
    <hr />
    @RenderBody()
    <hr />
    <h4>Faculty of Information Technology</h4>
</body>
</html>
```

Tạo action fitController.cs

```
namespace Proj_LayoutDemo0.Controllers
{
    public class fitController : Controller
    {
        // GET: fit
        public ActionResult Kythuat()
```

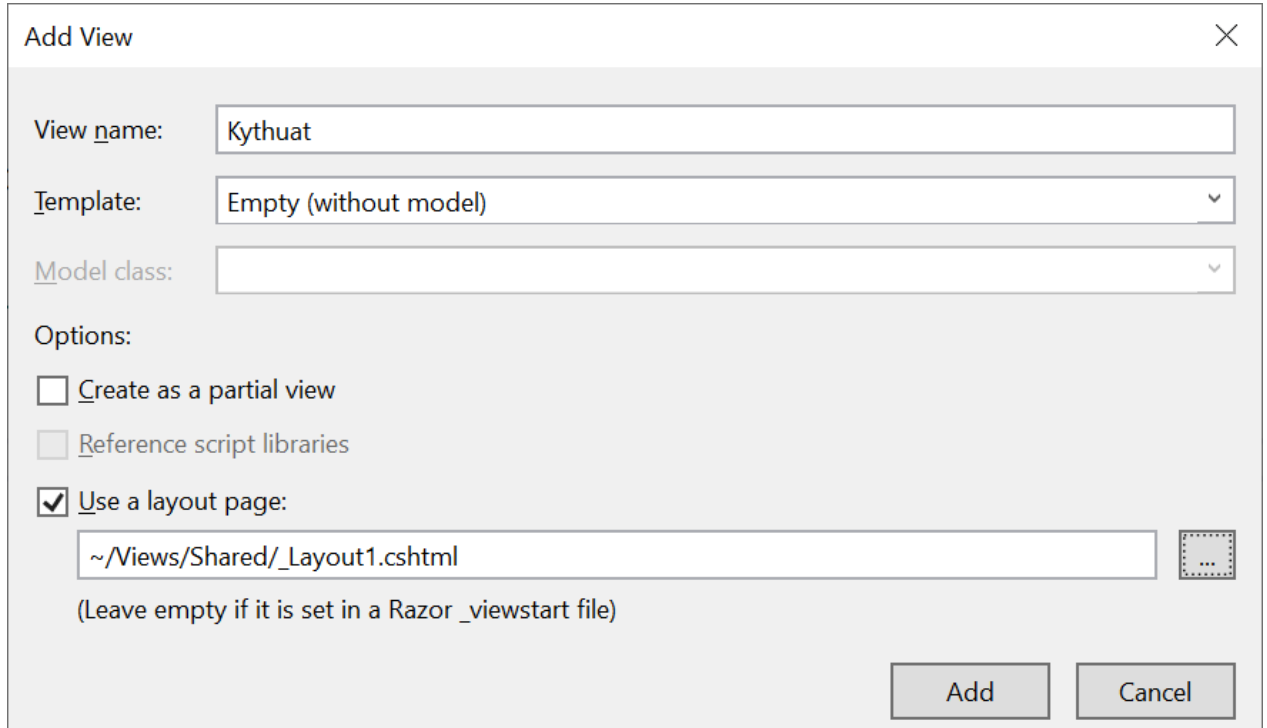


```

    {
        return View();
    }
    public ActionResult Hethong()
    {
        return View();
    }
}

```

Tạo view Kythuat.cshtml, sử dụng _Layout1.cshtml



Code của file Kythuat.cshtml như sau:

```

@{
    ViewBag.Title = "Kythuat";
    Layout = "~/Views/Shared/_Layout1.cshtml";
}

```

<h2>Ky thuat</h2>

Là ngành học Kỹ thuật lập trình

➔ Dòng Layout = "~/Views/Shared/_Layout1.cshtml"; cho biết sử dụng _Layout1.cshtml

Tiếp tục tạo view Hethong.cshtml, sử dụng _Layout1.cshtml

Add View
 ✕

View name:

Template:

Model class:

Options:

☐ Create as a partial view
 ☐ Reference script libraries
 ☒ Use a layout page:

...

(Leave empty if it is set in a Razor _viewstart file)

Add
Cancel

Code:

```

@{
    ViewBag.Title = "Hethong";
    Layout = "~/Views/Shared/_Layout1.cshtml";
}

<h2>He thong</h2>
Là ngành học hệ thống thông tin.
  
```

Chạy view Kythuat.cshtml:

Hanoi University of Industry

[Kỹ thuật lập trình](#)
[Hệ thống thông tin](#)

Ky thuat

Là ngành học Kỹ thuật lập trình

Falculty of Information Technology

Click vào link Hệ thống thông tin:

Hanoi University of Industry

[Kỹ thuật lập trình](#)
[Hệ thống thông tin](#)

He thong

Là ngành học hệ thống thông tin.

Faculty of Information Technology

⇒ Giao diện thống nhất, cùng sử dụng _Layout1.cshtml.

2. Razor View

2.1. Razor là gì?

- Razor là một cú pháp đánh dấu cho phép bạn nhúng mã dựa trên máy chủ vào các trang web bằng C# và VB.Net. Razor không phải là một ngôn ngữ lập trình. Nó là một ngôn ngữ đánh dấu phía máy chủ.
- Razor là ngôn ngữ ngắn gọn, rõ ràng và hữu ích cho phép tạo ra các giao diện ứng dụng ASP.NET MVC
- Cú pháp Razor: @{ <khối lệnh> }

Trong đó <khối lệnh> là mã C# hoặc VB.NET trộn HTML

2.2. Các ví dụ:

• Khối lệnh đơn

```
@{ var total = 7; }  
@{ var myMessage = "Hello World"; }
```

• Biểu thức nội tuyến

```
<p>The value of your account is: @total </p>  
<p>The value of myMessage is: @myMessage</p>
```

• Khối nhiều dòng lệnh

```
@{  
    var greeting = "Welcome to our site!";  
    var weekDay = DateTime.Now.DayOfWeek;  
    var greetingMessage = greeting + " Today is: " + weekDay;  
}  
<p>The greeting is: @greetingMessage</p>
```

• Lệnh If... else

```
@if(DateTime.IsLeapYear(DateTime.Now.Year) )
{
    @DateTime.Now.Year @:is a leap year.
}
else {
    @DateTime.Now.Year @:is not a leap year.
}
```

• Lệnh lặp for

```
@for (int i = 0; i < 5; i++) {
    @i.ToString() <br />
}
```

• Lệnh lặp foreach

```
<ul>
    @foreach (var myItem in Request.ServerVariables)
    {
        <li>@myItem</li>
    }
</ul>
```

• Lệnh lặp while

```
@{
    var countNum = 0;
    while (countNum < 50)
    {
        countNum += 1;
        <p>Line: @countNum: </p>
    }
}
```

• Model: Sử dụng @model để lấy ra đối tượng model tại vị trí bất kỳ trong view

```
@model Student

<h2>Student Detail:</h2>
<ul>
    <li>Student Id: @Model.StudentId</li>
    <li>Student Name: @Model.StudentName</li>
    <li>Age: @Model.Age</li>
</ul>
```

• Ví dụ

+ Trong HomeController.cs, tạo thêm 1 action Index2:

```
public ActionResult Index2()
{
    return View();
}
```

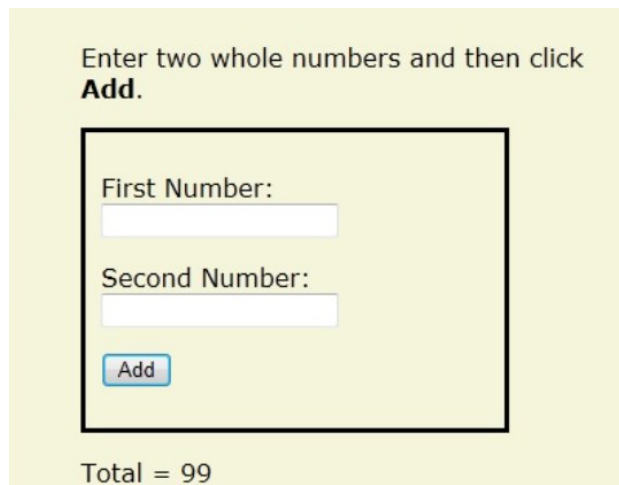
+ Tạo View **Index2.cshhtml** tương ứng:

```
@{
    var total = 0;
    var totalMessage = "";
    if(IsPost) {

        // Retrieve the numbers that the user entered.
        var num1 = Request["text1"];
        var num2 = Request["text2"];

        // Convert the entered strings into integers numbers and add.
        total = num1.AsInt() + num2.AsInt();
        totalMessage = "Total = " + total;
    }
}

<!DOCTYPE html>
<html lang="en">
<head>
<title>Add Numbers</title>
<meta charset="utf-8" />
<style type="text/css">
    body {background-color: beige; font-family: Verdana, Arial;
        margin: 50px; }
    form {padding: 10px; border-style: solid; width: 250px;}
</style>
</head>
<body>
<p>Enter two whole numbers and then click <strong>Add</strong>.</p>
<form action="" method="post">
    <p><label for="text1">First Number:</label>
        <input type="text" name="text1" />
    </p>
    <p><label for="text2">Second Number:</label>
        <input type="text" name="text2" />
    </p>
    <p><input type="submit" value="Add" /></p>
</form>
<p>@totalMessage</p>
</body>
</html>
```



Enter two whole numbers and then click **Add**.

First Number:

Second Number:

Total = 99

3. HTML Helpers in ASP.NET MVC

3.1. Standard HTML helper trong Asp.NET MVC5

Helper là thành phần sinh giao diện Web phù hợp, buộc dữ liệu với Model để duy trì thông tin trên các giao diện đó. Helper giúp đơn giản việc viết mã giao diện.

Standard HTML helpers được sử dụng để hiển thị các loại điều khiển HTML phổ biến nhất như: Label, TextBox, Password, TextArea, CheckBox, RadioButton, DropDownList, Listbox, Display, Editor and ActionLink v.v...

Standard HTML helpers luôn bắt đầu bằng @HTML. Chúng là một đối tượng của lớp HTML helpers. Ký hiệu @ được sử dụng để truy cập mã phía máy chủ. Phương thức mở rộng của class HTML help có một số phương thức được nạp chồng.

Ví dụ: @ Html.ActionLink ("Create Employee", "Create", "Employee")

sẽ tạo thẻ liên kết:

```
<a href="/Employee/Create"> Create Empolyee </a>
```

Danh sách các Standard HTML Helpers trong ASP.NET MVC 5

1. @Html.Label
2. @Html.TextBox
3. @Html.Password
4. @Html.TextArea
5. @Html.CheckBox
6. @Html.RadioButton
7. @Html.DropDownList
8. @Html.ListBox
9. @Html.Hidden
10. @Html.Display
11. @Html.Editor
12. @Html.ActionLink
13. @Html.BeginForm

3.2. Cách sử dụng

@Html.Label()

```
@Html.Label("User Name ")
```

Output:

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

@Html.TextBox

```
@Html.TextBox("txtUserName")
```

Output:

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

@Html.Hidden

```
@Html.Hidden("EmployeeId")
```

Output:

```
<input id="EmployeeId" name="EmployeeId" type="hidden" value="" />
```

@Html.Password()

```
@Html.Password("Password")
```

Output:

```
<input id="Password" name="Password" type="password" value="" />
```

@Html.TextArea()

```
@Html.TextArea("Address")
```

Output

```
<textarea cols="20" id="Address" name="Address" rows="2"/>
```

@ Html.CheckBox():

```
@Html.CheckBox("Cricket", true)
```

```
@Html.CheckBox("Dancing")
```

Output:

```
<input checked="checked" id="Cricket" name="Cricket" type="checkbox" value="true">
```

```
<input id="Dancing" name="Dancing" type="checkbox" value="false">
```

@Html.RadioButton()

```
@Html.RadioButton("Gender", "Male", true) Male
```

```
@Html.RadioButton("Gender", "Female", false) Female
```

Output:

```
<input checked="checked" id="male" name="Gender" type="radio" value="Male">
<input id="female" name="Gender" type="radio" value="Female">
```

@Html.DropDownList()

```
@{
    IEnumerable<string> strList = new List<string> { "BCA", "BCS", "MCA", "MCS"
    };
}
@Html.DropDownList("ddlCourse", new SelectList(strList, strList.FirstOrDefault(), "--Select Course--"))
```

Output:

```
<select id="ddlCourse" name="ddlCourse">
<option value="">--Select Course--</option>
<option selected="selected">BCA</option>
<option>BCS</option>
<option>MCA</option>
<option>MCS</option>
</select>
```

@Html.ListBox()

```
@Html.ListBox("Select Skills",new List<SelectListItem> {
    new SelectListItem{Text= "C#",Value="1"},
    new SelectListItem{ Text="ASP.NET",Value="2" },
    new SelectListItem{ Text="ASP.NET Core",Value="3" },
    new SelectListItem{ Text="Azure",Value="4" }
})
```

Output:

```
<select id="Select_Skills" multiple="multiple" name="Select Skills">
<option value="1">C#</option>
<option value="2">ASP.NET</option>
<option value="3">ASP.NET Core</option>
<option value="4">Azure</option>
</select>
```

@Html.ActionLink()

```
@Html.ActionLink("Go to About",Home, About)
```

Output:

```
<a href="/Home/About">Go to About</a>
```


@Html.BeginForm()

```
@using (Html.BeginForm("Index", "Home", FormMethod.Post))  
{  
    // Code here..  
}
```

Output:

```
<form action="Home/Index" method="post">  
</form>
```

3.3. Strongly-Typed HTML Helper

Danh sách các strongly-Typed HTML Helper

```
@Html.HiddenFor()  
@Html.LabelFor()  
@Html.TextBoxFor()  
@Html.RadioButtonFor()  
@Html.DropDownListFor()  
@Html.CheckBoxFor()  
@Html.TextAreaFor()  
@Html.PasswordFor()  
@Html.ListBoxFor()
```

3.4. Templated HTML Helpers**Hiển thị:**

```
@Html.Display()  
@Html.DisplayFor()  
@Html.DisplayName()  
@Html.DisplayNameFor()  
@Html.DisplayText()  
@Html.DisplayTextFor()  
@Html.DisplayModelFor()
```

Nhập, xuất:

```
@Html.Editor()  
@Html.EditorFor()  
@Html.EditorForModel()
```

4. Sử dụng Standard HTML helper thiết kế form

Student Registration Form

Student Name	<input type="text"/>
Gender	<input type="radio"/> Male <input type="radio"/> Female
Email	<input type="text"/>
Address	<input type="text" value="Hanoi"/> ▼
Subjects	<input type="checkbox"/> Java Core <input type="checkbox"/> SQL Server <input type="checkbox"/> PHP
Username	<input type="text"/>
Password	<input type="text"/>
Comment	<input type="text"/>
<input type="button" value="Regis"/>	<input type="button" value="Cancel"/>

Tạo file **StudentRegistrationController.cs** như sau:

```
public class StudentRegistrationController : Controller
{
    // GET: StudentRegistration
    public ActionResult Index()
    {
        return View();
    }
}
```

Tạo view cho phương thức Index()

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
</head>
<body>
    <div>
        <h2>Student Registration Form</h2>
        @using (Html.BeginForm("Regis", "StudentRegistration"))
        {
```

```

<table border="1" style="border-collapse:collapse" cellpadding="5" >
  <tr>
    <td>@Html.Label("Student Name")</td>
    <td>@Html.TextBox("name")</td>
  </tr>
  <tr>
    <td>@Html.Label("Gender")</td>
    <td>
      @Html.RadioButton("gender", "Male") Male
      @Html.RadioButton("gender", "Female") Female
    </td>
  </tr>
  <tr>
    <td>@Html.Label("Email")</td>
    <td>@Html.TextBox("email")</td>
  </tr>
  <tr>
    <td>@Html.Label("Address")</td>
    <td>
      @{
        IEnumerable<string> strList = new List<string>
        { "Hanoi", "Danang", "tpHCM" };
      }
      @Html.DropDownList("addr",
        new SelectList(strList, strList.FirstOrDefault()), "--Select--")
    </td>
  </tr>
  <tr>
    <td>@Html.Label("Subjects")</td>
    <td>
      @Html.CheckBox("Java core")Java Core
      @Html.CheckBox("SQL Server")SQL Server
      @Html.CheckBox("PHP")PHP
    </td>
  </tr>
  <tr>
    <td>@Html.Label("Username")</td>
    <td>@Html.TextBox("username")</td>
  </tr>
  <tr>
    <td>@Html.Label("Password")</td>
    <td>@Html.Password("password")</td>
  </tr>
  <tr>
    <td>@Html.Label("Comment")</td>
    <td>@Html.TextArea("comment", "")
  </tr>
  <tr>
    <td><input type="submit" value="Regis" /></td>
    <td><input type="reset" value="Cancel" /></td>
  </tr>
</table>
}
</div>
</body>
</html>

```

Chạy xem kết quả: <https://localhost:44349/StudentRegistration/Index>

Code HTML được sinh ra tương ứng như sau: (Click chuột phải vào Web Page, chọn View Source)

```

<!DOCTYPE html>

<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>Index</title>
</head>
<body>
  <div>
    <h2>Student Registration Form</h2>
    <form action="/StudentRegistration/Regis" method="post">
      <table border="1" style="border-collapse:collapse" cellpadding="5">
        <tr>
          <td><label for="Student_Name">Student Name</label></td>
          <td><input id="name" name="name" type="text" value="" /></td>
        </tr>
        <tr>
          <td><label for="Gender">Gender</label></td>
          <td>
            <input id="gender" name="gender" type="radio" value="Male" /> Male
            <input id="gender" name="gender" type="radio" value="Female" /> Female
          </td>
        </tr>
        <tr>
          <td><label for="Email">Email</label></td>
          <td><input id="email" name="email" type="text" value="" /></td>
        </tr>
        <tr>
          <td><label for="Address">Address</label></td>
          <td>
            <select id="addr" name="addr">
              <option value="">--Select--</option>
              <option selected="selected">Hanoi</option>
              <option>Danang</option>
              <option>tpHCM</option>
            </select>
          </td>
        </tr>
        <tr>
          <td><label for="Subjects">Subjects</label></td>
          <td>
            <input id="Java_core" name="Java core" type="checkbox"
            value="true" /><input name="Java core" type="hidden" value="false" />Java Core
            <input id="SQL_Server" name="SQL Server" type="checkbox"
            value="true" /><input name="SQL Server" type="hidden" value="false" />SQL Server
            <input id="PHP" name="PHP" type="checkbox" value="true"
            /><input name="PHP" type="hidden" value="false" />PHP
          </td>
        </tr>
        <tr>
          <td><label for="Username">Username</label></td>
          <td><input id="username" name="username" type="text" value="" /></td>
        </tr>
        <tr>
          <td><label for="Password">Password</label></td>
          <td><input id="password" name="password" type="password" /></td>
        </tr>
        <tr>
          <td><label for="Comment">Comment</label></td>
          <td>

```

```

        <textarea cols="20" id="comment" name="comment" rows="2"></textarea>
      </tr>
      <tr>
        <td><input type="submit" value="Regis" /></td>
        <td><input type="reset" value="Cancel" /></td>
      </tr>
    </table>
  </form>
</div>
</body>
</html>

```

Tạo file **Student.cs** ở thư mục Models có nội dung sau:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Proj_LayoutDemo.Models
{
    public class Student
    {
        public string name{ get; set; }
        public string gender { get; set; }
        public string email { get; set; }
        public string addr { get; set; }
        public string subjects { get; set; }
        public string username { get; set; }
        public string password { get; set; }
        public string comment { get; set; }
    }
}

```

Quay trở lại file **StudentRegistrationController.cs**, thêm phương thức

```

public ActionResult Regis(Student s)
{
    return View(s);
}

```

Biến **Student s** sẽ tự động lấy thông tin từ form chuyển sang controller và chuyển tiếp tới view **Regis.cshtml**.

Tạo view Regis và lựa chọn như hình:

Add View
 ✕

View name:

Template:

Model class:

Options:

☐ Create as a partial view
 ☐ Reference script libraries
 ☐ Use a layout page: ...

(Leave empty if it is set in a Razor `_viewstart` file)

Add
Cancel

MVC tự sinh code như sau: (Regis.cshtml)

```

@model Proj_LayoutDemo.Models.Student

@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Regis_ver2</title>
</head>
<body>
    <div>
        <h4>Student</h4>
        <hr />
        <dl class="dl-horizontal">
            <dt>
                @Html.DisplayNameFor(model => model.name)
            </dt>
            <dd>
                @Html.DisplayFor(model => model.name)
            </dd>
            <dt>
                @Html.DisplayNameFor(model => model.gender)
            </dt>
            <dd>
                @Html.DisplayFor(model => model.gender)
            </dd>
            <dt>
                @Html.DisplayNameFor(model => model.email)
            </dt>
            <dd>
                @Html.DisplayFor(model => model.email)
            </dd>
        </dl>
    </div>
  
```

```

<dt>
    @Html.DisplayNameFor(model => model.addr)
</dt>
<dd>
    @Html.DisplayFor(model => model.addr)
</dd>
<dt>
    @Html.DisplayNameFor(model => model.subjects)
</dt>
<dd>
    @Html.DisplayFor(model => model.subjects)
</dd>
<dt>
    @Html.DisplayNameFor(model => model.username)
</dt>
<dd>
    @Html.DisplayFor(model => model.username)
</dd>
<dt>
    @Html.DisplayNameFor(model => model.password)
</dt>
<dd>
    @Html.DisplayFor(model => model.password)
</dd>
<dt>
    @Html.DisplayNameFor(model => model.comment)
</dt>
<dd>
    @Html.DisplayFor(model => model.comment)
</dd>
</dl>
</div>
<p>
    @Html.ActionLink("Edit", "Edit", new { /* id = Model.PrimaryKey */ }) |
    @Html.ActionLink("Back to List", "Index")
</p>
</body>
</html>

```

Chạy file Views/StudentRegistration/index.cshtml

Student Registration Form

Student Name	<input type="text" value="Mai Lan"/>
Gender	<input type="radio"/> Male <input checked="" type="radio"/> Female
Email	<input type="text" value="lanmai.haui@gmail.com"/>
Address	<input type="text" value="Hanoi"/> <input type="button" value="v"/>
Subjects	<input checked="" type="checkbox"/> Java Core <input checked="" type="checkbox"/> SQL Server <input checked="" type="checkbox"/> PHP
Username	<input type="text" value="mailan"/>
Password	<input type="password" value="....."/>
Comment	<input type="text" value="Nothing"/>
<input type="button" value="Regis"/>	<input type="button" value="Cancel"/>

Kết quả nhận được là :

Student	
name	Mai Lan
gender	Female
email	lanmai.haui@gmail.com
addr	Hanoi
subjects	
username	mailan
password	123456
comment	Nothing
Edit Back to List	

Chúng ta thấy mục **subject** không nhận được giá trị bởi đây là các **check box** có tên khác nhau bên trang view.

Muốn nhận được giá trị của checkbox, ta cần thay đổi code trong file controller: Tạo thêm ActionResult **Regis2**, truyền tham số là **FormCollection**:

```
public ActionResult Regis2(FormCollection f)
{
    Student s = new Student();
    s.name = f["name"];
    s.gender = f["gender"];
    s.email = f["email"];
    s.addr = f["addr"];

    string temp = "";

    if (f["Java core"] == "true,false")
        temp = "Java core";
    if (f["SQL Server"] == "true,false")
        temp += " " + "SQL Server";
    if (f["PHP"] == "true,false")
        temp += " " + "PHP";

    s.subjects = temp;

    s.username = f["username"];
    s.password = f["password"];
}
```



```
s.comment = f["comment"];

return View(s);
}
```

Trong file **Index.cshtml**, đổi action ở đầu form thành Regis2

```
@using (Html.BeginForm("Regis2", "StudentRegistration"))
```

Đổi tên **Regis.cshtml** thành **Regis2.cshtml**

Chạy xem kết quả:

<https://localhost:44349/StudentRegistration/Index>

Kết quả thực hiện như sau:

Student	
name	Mai Lan
gender	Female
email	mailan.haui@gmail.com
addr	Hanoi
subjects	Java core SQL Server PHP
username	mailan
password	123456
comment	Nothing
Edit Back to List	

Như vậy giá trị của các check box đã nhận được và hiển thị ra màn hình.

Bài tập

Bài tập 1. Sử dụng Razor, viết code thực hiện yêu cầu như hình sau:

Nhập số thứ 1	<input type="text"/>
Nhập số thứ 2	<input type="text"/>
Kết quả	<input type="text"/>
<input type="button" value="Cộng"/> <input type="button" value="Trừ"/> <input type="button" value="Nhân"/> <input type="button" value="Chia"/>	

Bài tập 2. Sử dụng HTML Helpers, thiết kế form theo mẫu:

Registration Form	
First Name	<input type="text"/>
	<input type="text"/>
Email	<input type="text"/>
Password	<input type="text"/>
City	<input type="text" value="Hanoi"/> ▼
Gender	<input type="radio"/> Male <input type="radio"/> Female
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

Khi click vào Save, sẽ hiển thị các thông tin vừa nhập.