

BÀI 3: SỬ DỤNG LAYOUT, RAZOR VIEW ENGINE

Nội dung bài học:

1. Layout của website.....	1
1.1. Layout mặc định khi tạo web mvc 5	1
1.2. Tự làm 1 layout cho website	6
2. HTML Helpers in ASP.NET MVC	10
2.1. Standard HTML helper trong Asp.NET MVC5	10
2.2. Cách sử dụng	11
2.3. Strongly-Typed HTML Helper.....	15
2.4. Templated HTML Helpers.....	15
3. Sử dụng Standard HTML helper thiết kế form	16
4. Các lưu ý khi sử dụng Razor View	24
4.1. Sử dụng @ khi viết code:	24
4.2. Đặt các khối mã trong cặp dấu { }	25
4.3. Bên trong một khối, kết thúc mỗi câu lệnh mã bằng dấu chấm phẩy	25
4.4. Sử dụng các biến để lưu trữ các giá trị	25
4.5. Đặt các giá trị chuỗi ký tự trong dấu ngoặc kép	25
4.6. Code phân biệt chữ hoa chữ thường	25
4.7. Phần lớn code của bạn liên quan đến các đối tượng	26
4.8. Viết mã đưa ra quyết định và vòng lặp	26
4.9. Collection Objects (Arrays and Dictionaries)	28
4.10. Sử dụng try/catch.....	29

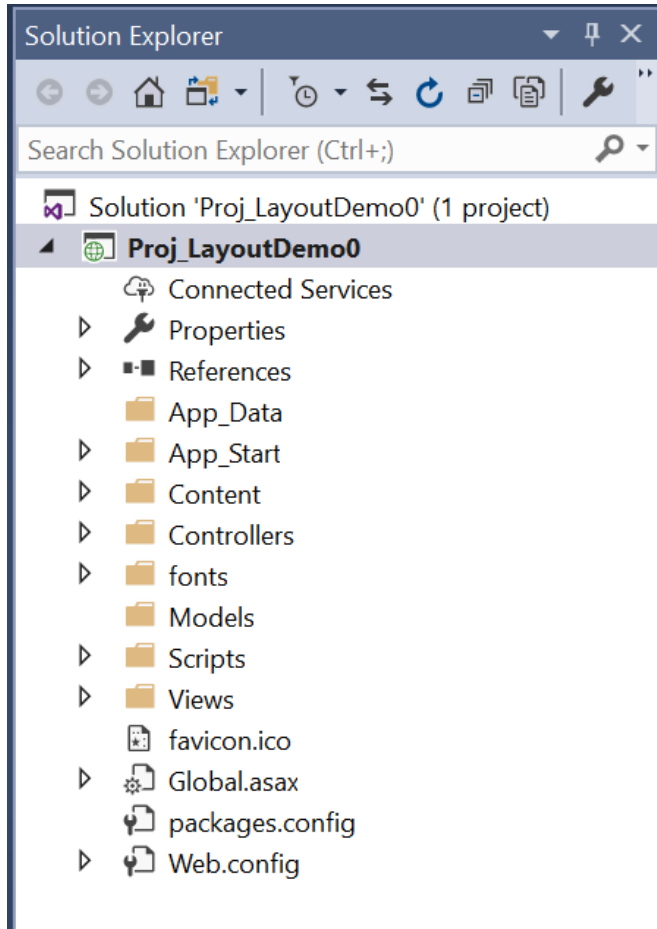
1. Layout của website

1.1. Layout mặc định khi tạo web mvc 5

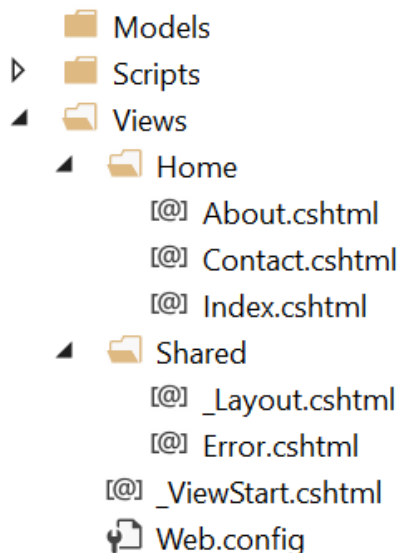
Trong một website, chúng ta thấy các page luôn giống nhau ở những vị trí như header, footer, body ... Thông thường, chỉ những nội dung ở giữa trang là thay đổi.

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:

ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.

[Learn more »](#)

Getting started

ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and gives you full control over markup for enjoyable, agile development.

[Learn more »](#)

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

[Learn more »](#)

Click vào About:

About.

Your application description page.

Use this area to provide additional information.

© 2021 - My ASP.NET Application

Click vào 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

```

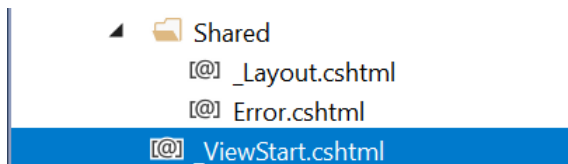
File Contact.cshtml có nội dung:

```

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:

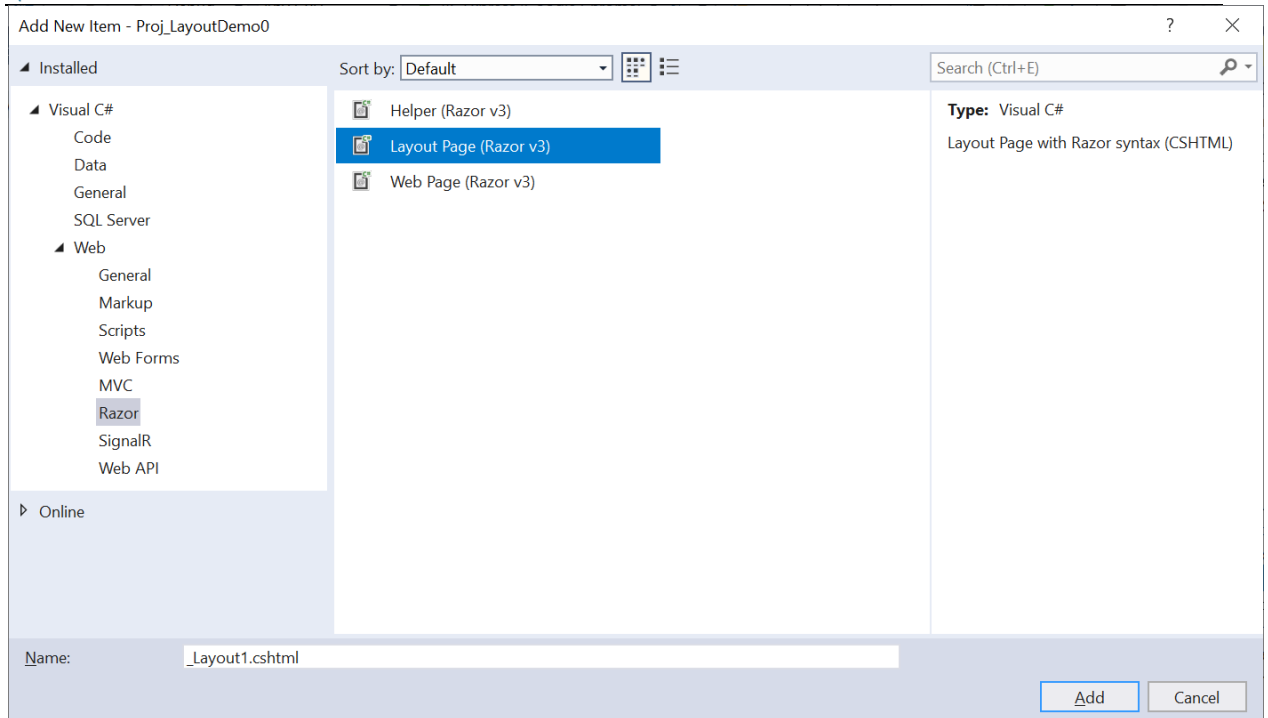
```

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

```

1.2. Tự làm 1 layout cho website

Tạo một Layout mới, đặt tên là _Layout.cshtml1 như hình:



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

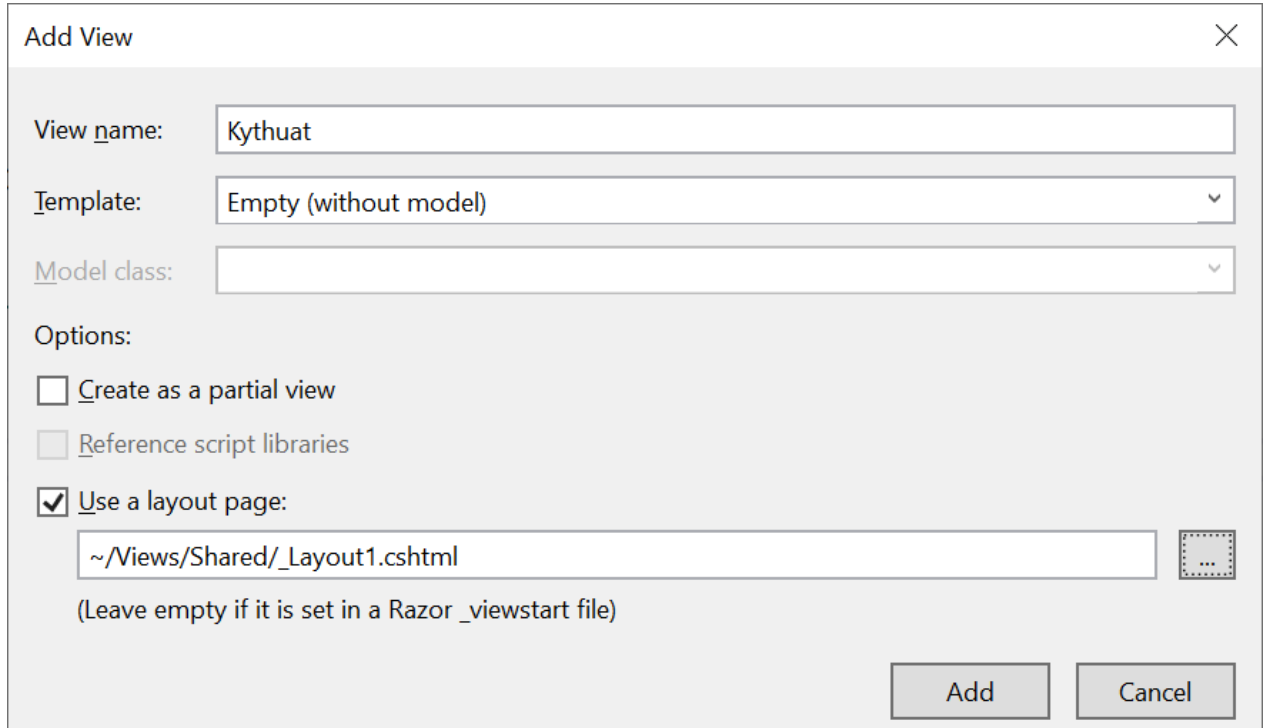
```
namespace Proj_LayoutDemo0.Controllers
{
    public class fitController : Controller
    {
        // GET: fit
        public ActionResult Kythuat()
```

```

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

```

Tạo view Kythuat.cshtml, sử dụng _Layout1.cshtml



Add View

View name: Kythuat

Template: Empty (without model)

Model class:

Options:

☐ Create as a partial view

☐ Reference script libraries

☒ Use a layout page:

~/Views/Shared/_Layout1.cshtml

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

Add Cancel

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

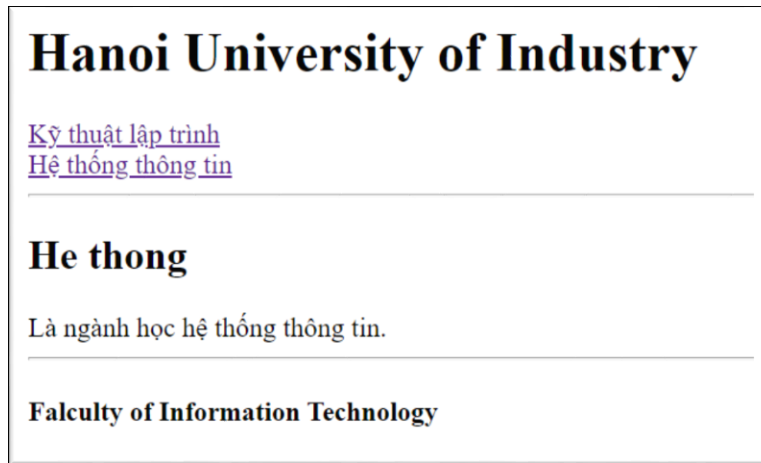
[Kỹ thuật lập trình](#)
[Hệ thống thông tin](#)

Ky thuat

Là ngành học Kỹ thuật lập trình

Falculty of Information Technology

Click vào link Hệ thống thông tin:



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

2. HTML Helpers in ASP.NET MVC

2.1. Standard HTML helper trong Asp.NET MVC5

Standard HTML helpers được sử dụng để hiển thị các loại điều khiển HTML phổ biến nhất 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**.

HTML là một thuộc tính của kiểu HtmlHelper được bao gồm trong lớp cơ sở của WebViewPage của chế độ xem Razor view. TextBox () hoặc TextBoxFor () ... là các phương thức mở rộng có trong class HtmlHelper. Class HtmlHelper tạo ra các phần tử HTML.

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

sẽ tạo thẻ liên kết:

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

2.2. Cách sử dụng

@Html.Label()

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

Output:

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

@Html.TextBox

```
Html.TextBox("txtUserName", "", new { @class = "form-control" })
```

Output:

```
<input class="form-control" 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", "", new { @class = "form-control" })
```

Output:

```
<input class="form-control" id="Password" name="Password" type="password" value="">
```

@Html.TextArea()

```
@Html.TextArea("Address", " ", new { @class = "form-control", id="IdAddress" })
```

Output

```
<textarea class="form-control"
cols="20"
id="IdAddress"
name="Address"
rows="2">
</textarea>
```

@ 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, new { id = "male" }) Male
```

```
@Html.RadioButton("Gender", "Female", false, new { id = "female" }) 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.Label("User Name ")
```

Output:

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

@Html.BeginForm()

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

Output:

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

*Ví dụ:

```
@{
    Layout = null;
}
<h2>Standard HTML helper trong Asp.NET MVC5</h2>

<h3>Label example</h3>
@Html.Label("firstName", "First Name")

<h3>Text Box Example</h3>
@Html.TextBox("txtFirstName", "", new { @class = "form-control",
placeholder = "First Name" })
```

```

<h3>Text Area Example</h3>
@Html.TextArea("address", new { @class = "form-control", rows = "5" })

<h3>password Example</h3>
@Html.Password("password", " ", new { @class = "form-control" })

<h3>Radio Button Example</h3>
@Html.RadioButton("MaritalStatus", "Married", new { id = "IsMarried" })
Married

<h3>Check Box Example</h3>
@Html.CheckBox("htmlSkill") HTML 5

<h3>List Box Example</h3>
@Html.ListBox("Skills", new List<SelectListItem> {
    new SelectListItem { Text="ASP.NET",Value="1"},
    new SelectListItem { Text="MVC",Value="2"},
    new SelectListItem { Text="SQL Server",Value="3"},
    new SelectListItem { Text="Angular",Value="4"},
    new SelectListItem { Text="Web API",Value="5"}
}, new { @class = "form-control" })

<h3>drop down List Example</h3>
@Html.DropDownList("Gender", new List<SelectListItem> {
    new SelectListItem {Text="Select Gender",Value="-1" },
    new SelectListItem {Text="Male",Value="1" },
    new SelectListItem {Text="Female", Value="2" }
}, new { @class = "custom-select" })

```

Kết quả render ra các thẻ HTML như sau:

```

<h3>password Example</h3>
<input class="form-control" id="password" name="password" type="password" value="
" />

<h3>Radio Button Example</h3>
<input id="IsMarried" name="MaritalStatus" type="radio" value="Married" /> Married

<h3>Check Box Example</h3>
<input id="htmlSkill" name="htmlSkill" type="checkbox" value="true" /><input
name="htmlSkill" type="hidden" value="false" /> HTML 5

<h3>List Box Example</h3>
<select class="form-control" id="Skills" multiple="multiple" name="Skills">
    <option value="1">ASP.NET</option>
    <option value="2">MVC</option>
    <option value="3">SQL Server</option>
    <option value="4">Angular</option>
    <option value="5">Web API</option>
</select>

<h3>drop down List Example</h3>
<select class="custom-select" id="Gender" name="Gender">
    <option value="-1">Select Gender</option>
    <option value="1">Male</option>
    <option value="2">Female</option>
</select>

```

2.3. 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()
```

2.4. 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()
```

Ví dụ:

```
public class HomeController : Controller
{
    public ActionResult Details()
    {
        //Here we are hardcoded the Employee Details
        //In Realtime you will get the data from any data source
        Employee employee = new Employee()
        {
            EmpId = 1,
            Name = "Rishabh Tyagi",
            Gender = "Male",
            city = city.Dehli,
            skills = skills.WebAPI,
        }
    }
}
```

```

    Address = "lajpat Nagar",
    AgreeTerm = true
  };
  ViewData["EmployeeData"] = employee;
  return View();
}
}

```

Trang view:

```

@{
    ViewBag.Title = "Details";
}
<fieldset>
    <legend>Employee Details</legend>
    @Html.Display("EmployeeData")
</fieldset>

```

3. 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>
            </table>
        }
    </div>
</body>
</html>
```

```

        <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", "")</td>
        </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>
            </table>
        </form>
    </div>

```

```

    </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>
    </td>
    </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;

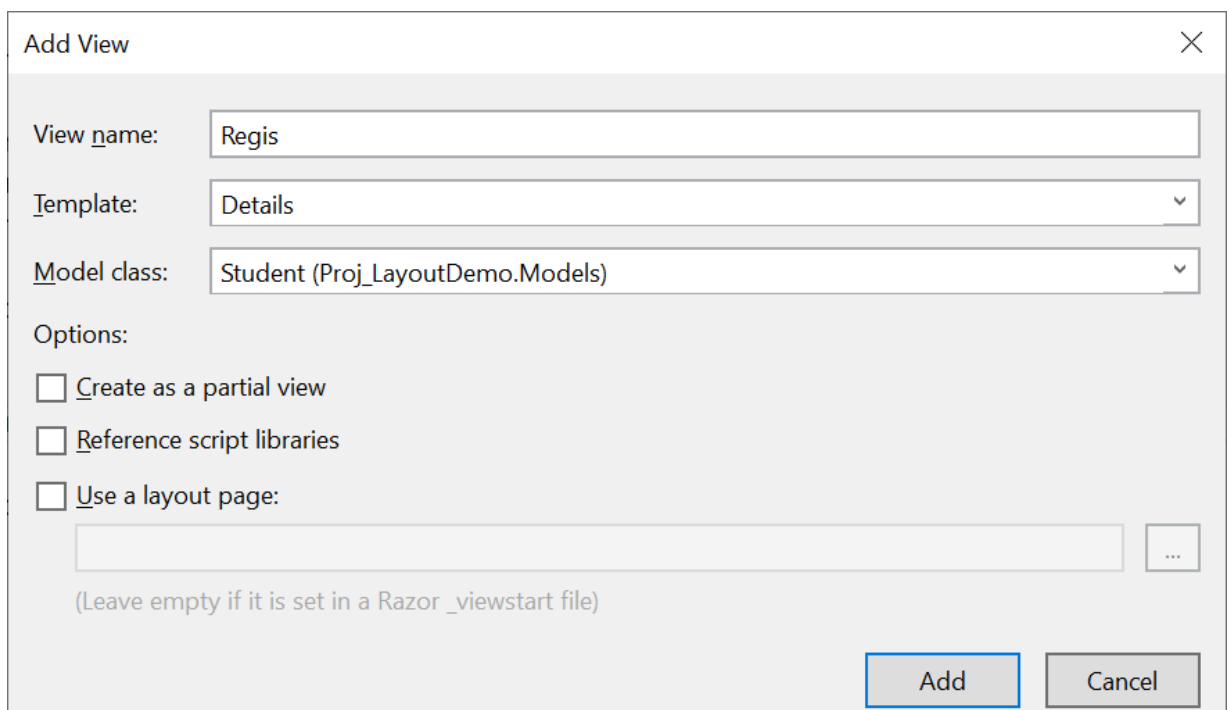
namespace Proj_LayoutDemo.Models
{
    public class Student
    {
        public string name{ get; set; }
        public string gender { get; set; }
        public string email { get; set; }
        public string addr { get; set; }
        public string subjects { get; set; }
        public string username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



MVC tự sinh code như sau:

```
@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>
            <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>
        </dl>
    </div>
</body>
</html>
```

```

        </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"/> ▼
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.

4. Các lưu ý khi sử dụng Razor View

4.1. Sử dụng @ khi viết code:

```
<!-- Single statement blocks -->
@{ var total = 7; }
@{ var myMessage = "Hello World"; }

<!-- Inline expressions -->
<p>The value of your account is: @total </p>
<p>The value of myMessage is: @myMessage</p>

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


4.2. Đặt các khối mã trong cặp dấu {}

```
<!-- Single statement block. -->
@{ var theMonth = DateTime.Now.Month; }sn
<p>The numeric value of the current month: @theMonth</p>

<!-- Multi-statement block. -->
@{
    var outsideTemp = 79;
    var weatherMessage = "Hello, it is " + outsideTemp + " degrees.";
}
<p>Today's weather: @weatherMessage</p>
```

4.3. Bên trong một khối, kết thúc mỗi câu lệnh mã bằng dấu chấm phẩy

```
<!-- Single-statement block -->
@{ var theMonth = DateTime.Now.Month; }

<!-- Multi-statement block -->
@{
    var outsideTemp = 79;
    var weatherMessage = "Hello, it is " +
        outsideTemp + " degrees.";
}

<!-- Inline expression, so no semicolon -->
<p>Today's weather: @weatherMessage</p>
```

4.4. Sử dụng các biến để lưu trữ các giá trị

```
<!-- Storing a string -->
@{ var welcomeMessage = "Welcome, new members!"; }
<p>@welcomeMessage</p>

<!-- Storing a date -->
@{ var year = DateTime.Now.Year; }

<!-- Displaying a variable -->
<p>Welcome to our new members who joined in @year!</p>
```

4.5. Đặt các giá trị chuỗi ký tự trong dấu ngoặc kép

```
@{ var myString = "This is a string literal"; }

//

<!-- Embedding a backslash in a string -->
@{ var myFilePath = @"C:\MyFolder\"; }
<p>The path is: @myFilePath</p>

//

<!-- Embedding double quotation marks in a string -->
@{ var myQuote = @"The person said: ""Hello, today is Monday."""; }
<p>@myQuote</p>
```

4.6. Code phân biệt chữ hoa chữ thường

```
@{
    var lastName = "Smith";
    var LastName = "Jones";
}
```

Lưu ý: Visual Basic không phân biệt chữ hoa hay chữ thường.

4.7. Phần lớn code của bạn liên quan đến các đối tượng

Chúng ta thường xuyên làm việc với đối tượng request, đối tượng này cung cấp cho chúng ta thông tin như giá trị của các textbox trên trang, loại trình duyệt đã thực hiện yêu cầu, URL của trang, danh tính người dùng, v.v.

Sau đây ví dụ cho thấy cách truy cập thuộc tính của đối tượng request và cách gọi phương thức MapPath của đối tượng request, cung cấp cho bạn đường dẫn tuyệt đối của trang trên máy chủ:

```
<table border="1">
<tr>
<td>Requested URL</td>
<td>Relative Path</td>
<td>Full Path</td>
<td>HTTP Request Type</td>
</tr>
<tr>
<td>@Request.Url</td>
<td>@Request.FilePath</td>
<td>@Request.MapPath(Request.FilePath)</td>
<td>@Request.RequestType</td>
</tr>
</table>
```

4.8. Viết mã đưa ra quyết định và vòng lặp

Câu lệnh if:

```
@{
var result = "";
if(IsPost)
{
result = "This page was posted using the Submit button.";
}
else
{
result = "This was the first request for this page.";
}
}

<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
<body>
<form method="POST" action="" >
<input type="Submit" name="Submit" value="Submit"/>
<p>@result</p>
</form>
</body>
</html>
```

Một ví dụ đơn giản:

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

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

+ Tạo View **Index2.cshtml** 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

Vòng lặp for:

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

Vòng lặp while:

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

4.9. Collection Objects (Arrays and Dictionaries)

Ví dụ 1:

```
@* Array block 1: Declaring a new array using braces. *@
@{
    <h3>Team Members</h3>
    string[] teamMembers = {"Matt", "Joanne", "Robert", "Nancy"};
    foreach (var person in teamMembers)
    {
        <p>@person</p>
    }
}
```

Ví dụ 2:

```
@{
    string[] teamMembers = {"Matt", "Joanne", "Robert", "Nancy"};
    <p>The number of names in the teamMembers array: @teamMembers.Length </p>
    <p>Robert is now in position: @Array.IndexOf(teamMembers, "Robert")</p>
    <p>The array item at position 2 (zero-based) is @teamMembers[2]</p>
    <h3>Current order of team members in the list</h3>
}
```

```

    foreach (var name in teamMembers)
    {
        <p>@name</p>
    }
    <h3>Reversed order of team members in the list< /h3>
    Array.Reverse(teamMembers);
    foreach (var reversedItem in teamMembers)
    {
        <p>@reversedItem</p>
    }
}

```

Ví dụ 3:

```

@{
    var myScores = new Dictionary<string, int>();
    myScores.Add("test1", 71);
    myScores.Add("test2", 82);
    myScores.Add("test3", 100);
    myScores.Add("test4", 59);
}
<p>My score on test 3 is: @myScores["test3"]%</p>
@{myScores["test4"] = 79;}
<p>My corrected score on test 4 is: @myScores["test4"]%</p>

```

4.10. Sử dụng try/catch

```

@{
    ViewBag.Title = "Index";
}

<h2>Try catch demo</h2>
@{
    var dataFilePath = "~/dataFile.txt";
    var fileContents = "";
    var physicalPath = Server.MapPath(dataFilePath);
    var userMessage = "Hello world, the time is " + DateTime.Now;
    var userErrMsg = "";
    var errMsg = "";

    if (IsPost)
    {
        //Khi người dùng click vào "Open File", sẽ mở file và đọc.
        try
        {
            //xuất hiện lỗi đầu tiên vì file bên dưới không tồn tại, sau đó chú
            thích lại lệnh này
            fileContents = File.ReadAllText(@"c:\batafile.txt");

            //tắt dòng trên đi và chạy lại
            fileContents = File.ReadAllText(physicalPath);
        }
        catch (FileNotFoundException ex)
        {
            errMsg = ex.Message;
            userErrMsg = "A file could not be opened, please contact your system
administrator.";
        }
        catch (DirectoryNotFoundException ex)
        {
            errMsg = ex.Message;
            userErrMsg = "A directory was not found, please contact your system
administrator.";
        }
    }
}

```

```
}  
  
}  
else  
{  
    //Lần đầu tiên chạy trang (get), tạo ra file text.  
    File.WriteAllText(physicalPath, userMessage);  
}  
  
}  
<form method="post" action="">  
    <input type="submit" name="submit" value="Open File"/>  
</form>  
<p> </p>  
<p>@physicalPath</p>  
<p> </p>  
<p>@fileContents</p>  
<p> </p>  
<p>@userErrMsg</p>
```