

MCSE 541: Web Computing and Mining

ASP.NET Core-Razor Views

Prof. Dr. Shamim Akhter

Razor

- ▶ A view engine in ASP.NET MVC
- ▶ Allows to write mix of HTML and server side code using C#
- ▶ C# syntax has .cshtml file extension
- ▶ Razor syntax has following Characteristics:

- ▶ **Compact:** enables to minimize number of characters and keystrokes required to write a code.
- ▶ **Easy to Learn:** familiar language C# or Visual Basic.
- ▶ **Intelligence:** supports statement completion within Visual Studio.

- ▶ Supports inline or multi-statement code block

- ▶ `<h2>@DateTime.Now.ToShortDateString()</h2>`

- ▶ **Output:** 08-09-2014

@{

var date = DateTime.Now.ToShortDateString();

var message = "Hello World";

}

`<h2>Today's date is: @date </h2>`

`<h3>@message</h3>`

Use @: or <text>/<text> to display texts within code block.

The C# Script Compilation

- ▶ Modern Browsers have Javascript Interpreters(Javascript engine).
 - ▶ So when the browser come across javascript code,each line is executed by the Javascript engine.
- ▶ But in case of Java or C++ or C# you convert the code into an intermediate form.
- ▶ Normally in ASP.NET MVC, the views are not compiled until they are requested by the browser.
 - ▶ To avoid this you can precompile the views. Precompile also helps to identify any errors upfront than at runtime.
- ▶ We need to Add the following lines in CSProj file:

```
"tools": { "Microsoft.AspNetCore.Mvc.Razor.ViewCompilation.Tools": {  
  "version": "1.1.0-preview4-final" } }
```



if-else condition

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

@ is razor syntax to access the server side variable.



for loop

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



Model

Use @model to use model object anywhere in the 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>
```



HTML Helpers

- HtmlHelper class generates html elements using the model class object in razor view.
- It binds the model object to html elements to display value of model properties into html elements and also assigns the value of the html elements to the model properties while submitting web form.

```
@model IEnumerable<MVC_BasicTutorials.Models.Student>
```

```
@{  
    ViewBag.Title = "Index";  
    Layout = "~/Views/Shared/_Layout.cshtml";  
}
```

```
<h2>Index</h2>
```

```
<p>  
    @Html.ActionLink("Create New", "Create")  
</p>  
<table class="table">  
    <tr>  
        <th>  
            @Html.DisplayNameFor(model => model.StudentName)  
        </th>  
        <th>  
            @Html.DisplayNameFor(model => model.Age)  
        </th>  
    </tr>
```

@Html.ActionLink("Create New", "Create") would generate anchor tag
`Create New.`

Extension method for
HtmlHelper

`<a asp-controller="Speaker" asp-action="Evaluations">Speaker Evaluations`

`Speaker Evaluations`

`<html> <body> <a asp-controller="Speaker" asp-action="Detail" asp-route-id="@Model.SpeakerId">SpeakerId: @Model.SpeakerId </body> </html>`

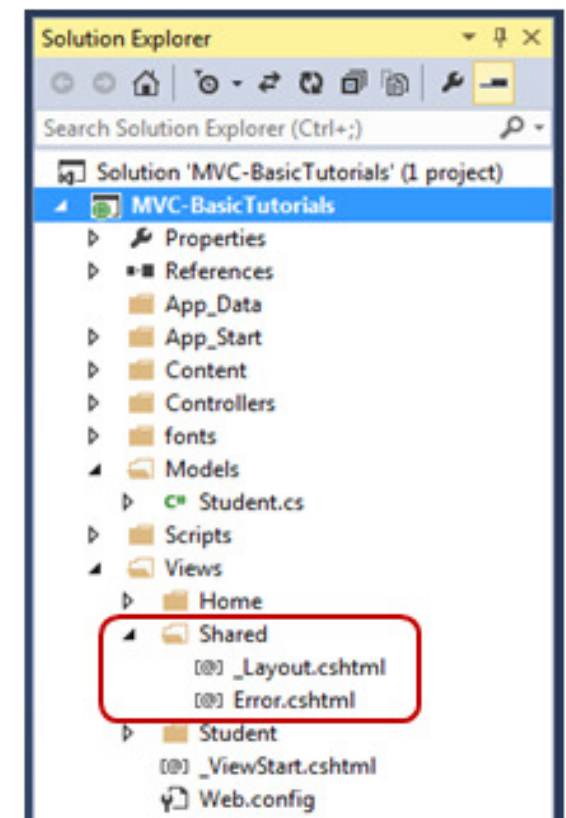
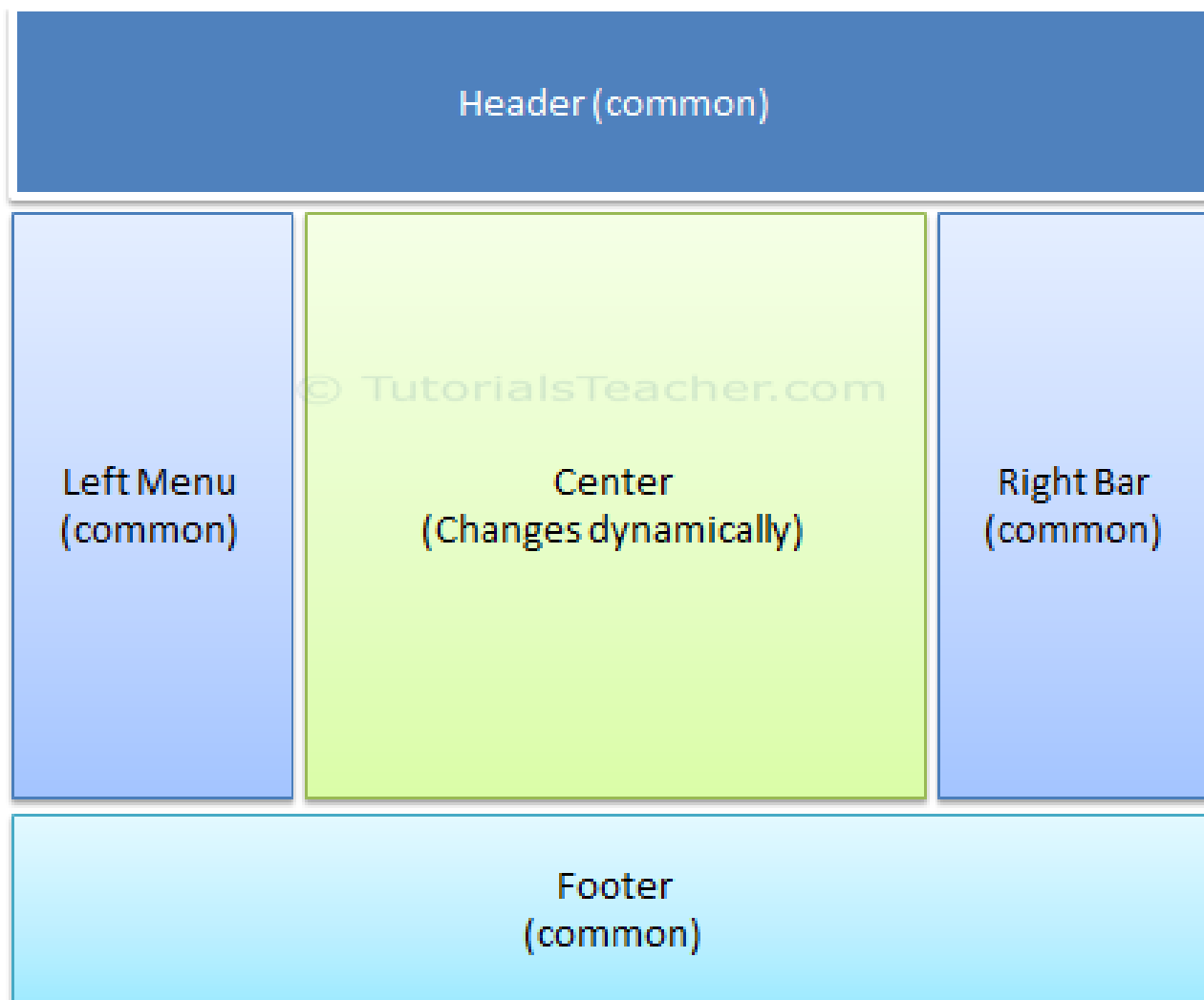
`SpeakerId: 12`

HTML Helper methods

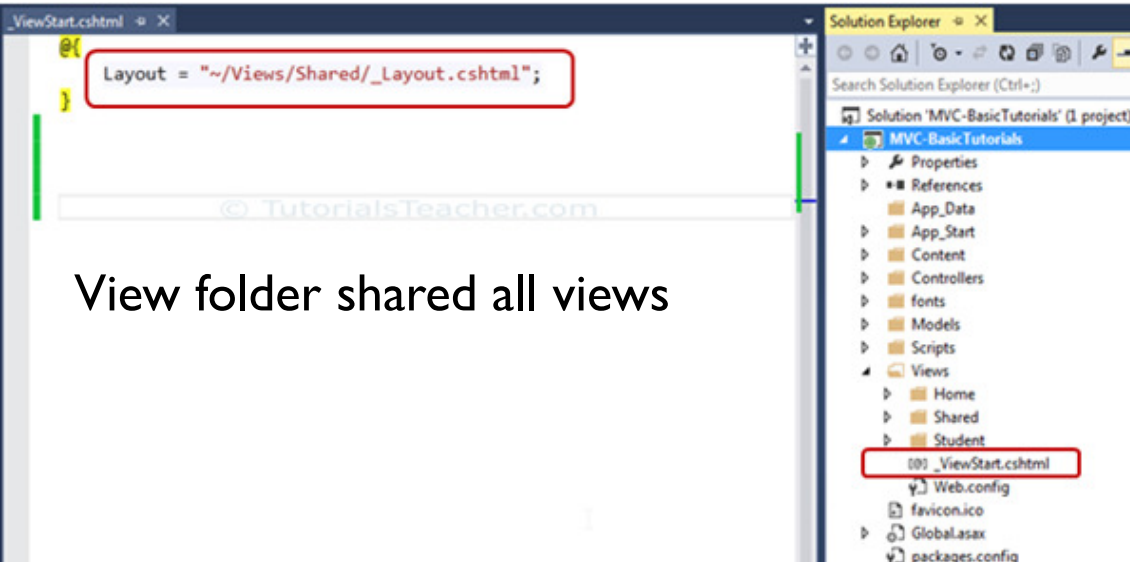
HtmlHelper	Strongly Typed HtmlHelpers	Html Control
Html.ActionLink		Anchor link
Html.TextBox	Html.TextBoxFor	Textbox
Html.TextArea	Html.TextAreaFor	TextArea
Html.CheckBox	Html.CheckBoxFor	Checkbox
Html.RadioButton	Html.RadioButtonFor	Radio button
Html.DropDownList	Html.DropDownListFor	Dropdown, combobox
Html.ListBox	Html.ListBoxFor	multi-select list box
Html.Hidden	Html.HiddenFor	Hidden field
Password	Html.PasswordFor	Password textbox
Html.Display	Html.DisplayFor	Html text
Html.Label	Html.LabelFor	Label
Html.Editor	Html.EditorFor	Generates Html controls based on data type of specified model property e.g. textbox for string property, numeric field for int, double or other numeric type.

Layout View

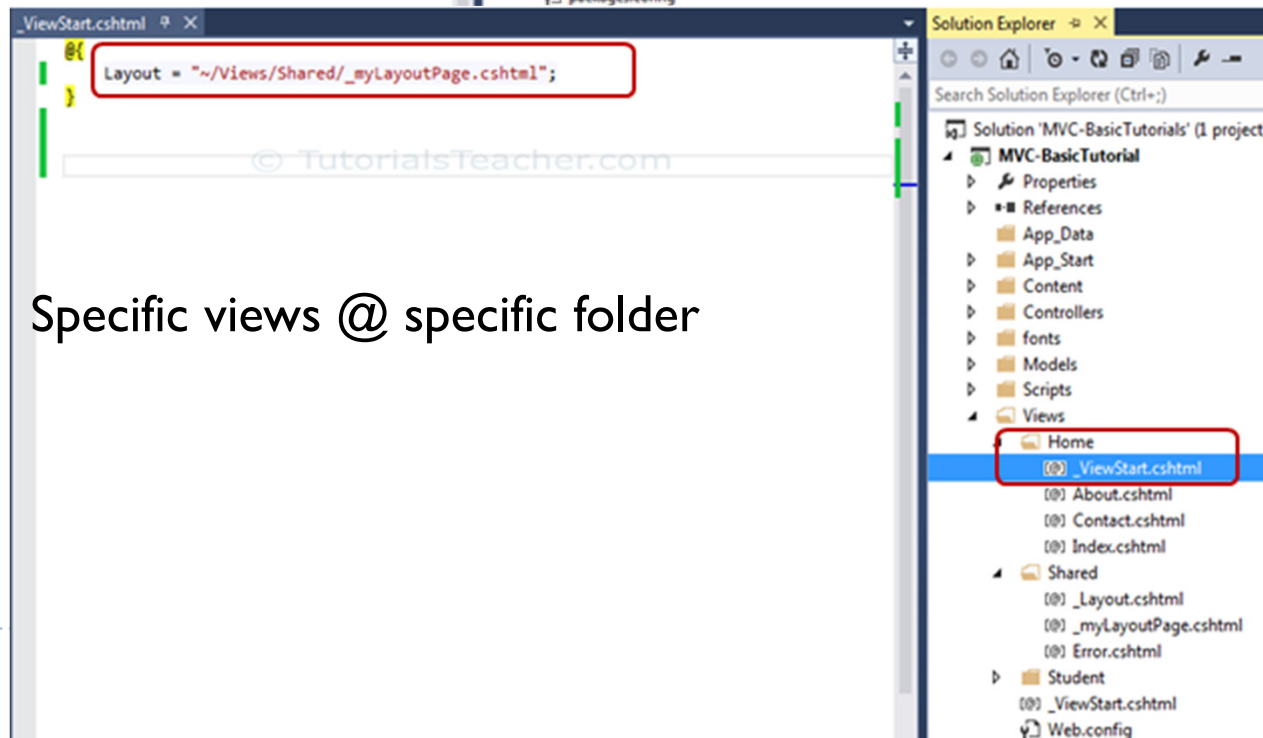
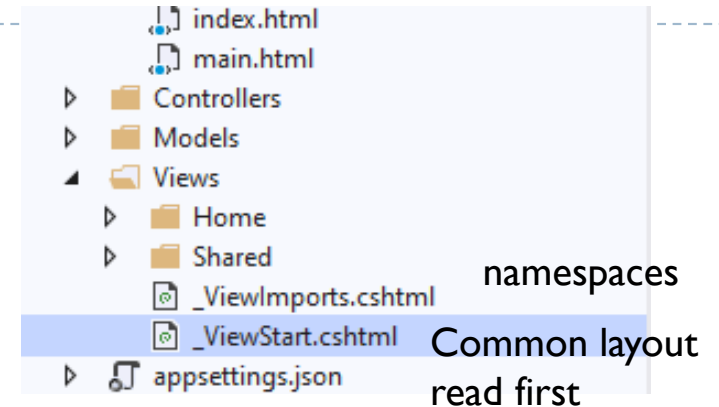
- ▶ Common parts in the UI such as the logo, header, left navigation bar, right bar or footer section.



Use Layout View



View folder shared all views



Specific views @ specific folder

Override default layout page

```
@{
    ViewBag.Title = "Home Page";
    Layout = "~/Views/Shared/_myLayoutPage.cshtml";
}

<div class="jumbotron">
    <h1>ASP.NET</h1>
    <p class="lead">ASP.NET is a free web framework for building great Web sites and Web applications</p>
    <p><a href="http://asp.net" class="btn btn-primary btn-lg">Learn more &raquo;</a></p>
</div>

<div class="row">
    <div class="col-md-4">
        <h2>Getting started</h2>
        <p>
            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.
        </p>
    </div>
</div>
```



Specify Layout Page in ActionResult Method

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View("Index", "_myLayoutPage");
    }

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

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

Rendering Methods

Method	Description
<code>RenderBody()</code>	Renders the portion of the child view that is not within a named section. Layout view must include <code>RenderBody()</code> method.
<code>RenderSection(string name)</code>	Renders a content of named section and specifies whether the section is required. <code>RenderSection()</code> is optional in Layout view.

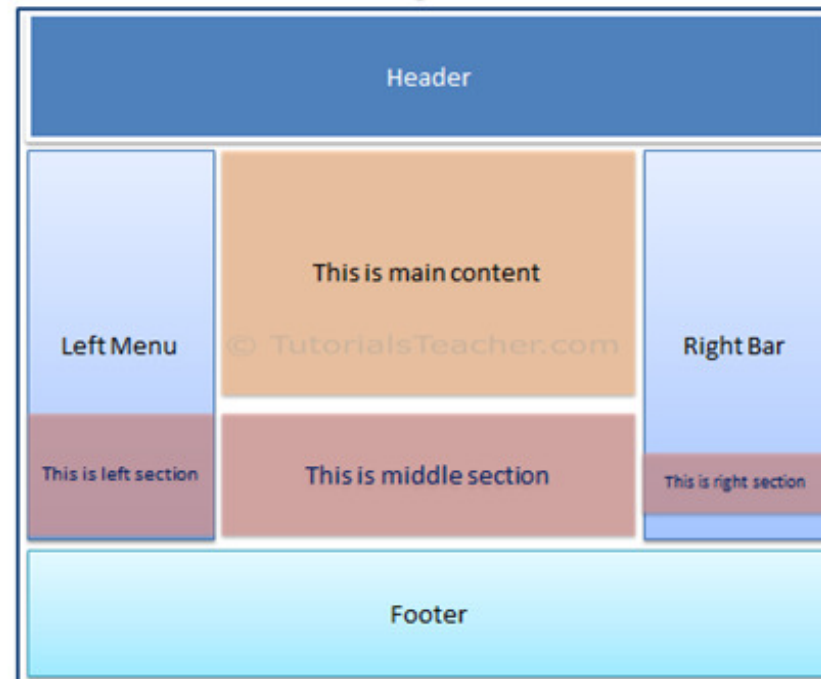
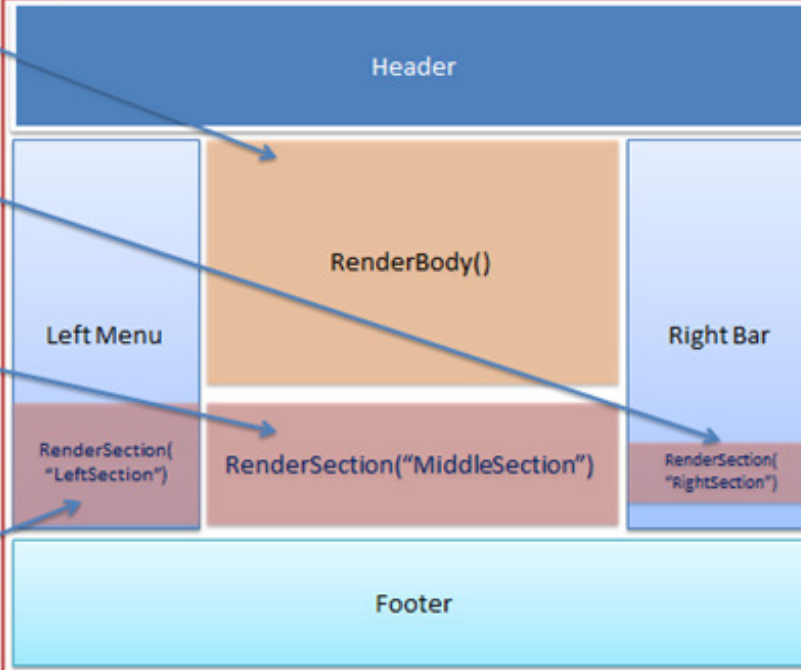


Index.cshtml

```
<div>
  This is main content
</div>

@section RightSection{
  <text>
    This is right section content
  </text>
}
@section MiddleSection{
  <text>
    This is middle section content
  </text>
}
@section LeftSection{
  <text>
    This is left section content
  </text>
}
```

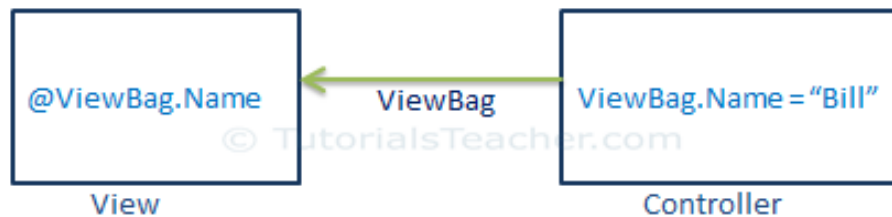
_Layout.cshtml



<http://localhost:1234/home/index>

ASP.NET MVC - ViewBag

- ▶ Transfers temporary data (which is not included in model) from the controller to the view.



@ is razor syntax to access the server side variable.

```
namespace MVC_BasicTutorials.Controllers
```

```
{
```

```
    public class StudentController : Controller
```

```
    {
```

```
        IList<Student> studentList = new List<Student>() {
```

```
            new Student(){ StudentID=1, StudentName="Steve", Age = 21 },
```

```
            new Student(){ StudentID=2, StudentName="Bill", Age = 25 },
```

```
            new Student(){ StudentID=3, StudentName="Ram", Age = 20 },
```

```
            new Student(){ StudentID=4, StudentName="Ron", Age = 31 },
```

```
            new Student(){ StudentID=5, StudentName="Rob", Age = 19 }
```

```
        };
```

```
        // GET: Student
```

```
        public ActionResult Index()
```

```
        {
```

```
            ViewBag.TotalStudents = studentList.Count();
```

```
            return View();
```

```
        }
```

```
    }
```

```
}
```

Server Code

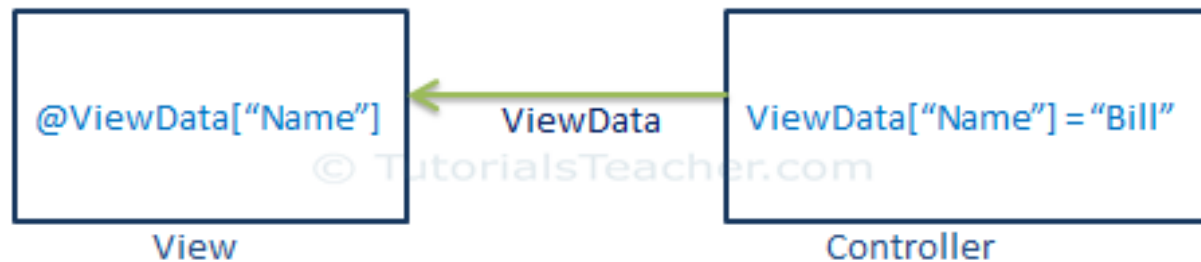
Script Code

```
<label>Total Students:</label> @ViewBag.TotalStudents
```

```
Total Students: 5
```

ASP.NET MVC - ViewData

- ▶ ViewData is similar to ViewBag. It is useful in transferring data from Controller to View.
- ▶ ViewData is a dictionary which can contain **key-value pairs** where each key must be string.



```
public ActionResult Index()
```

Server Code

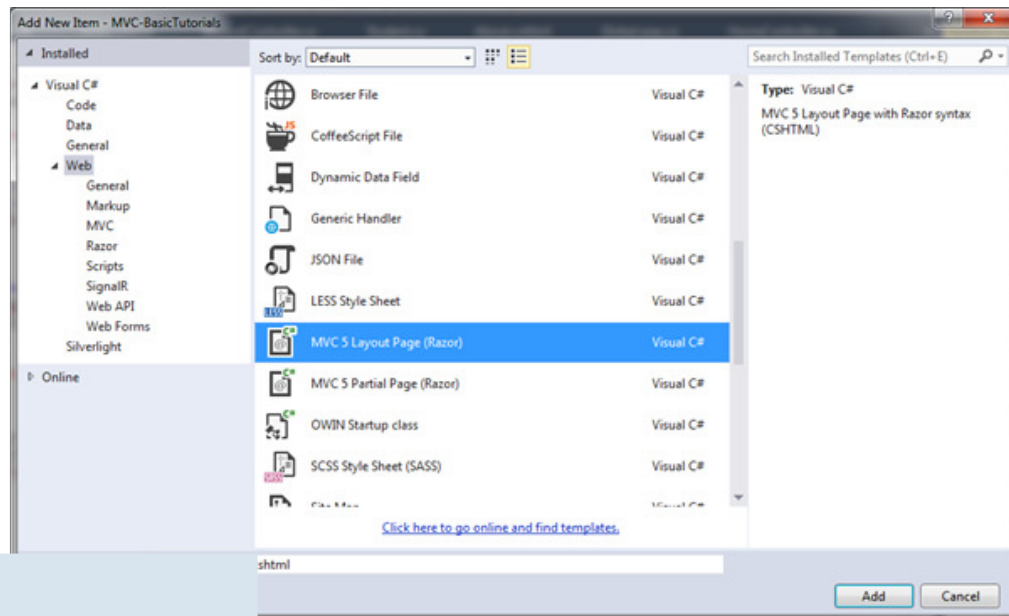
```
{  
    IList<Student> studentList = new List<Student>();  
    studentList.Add(new Student(){ StudentName = "Bill" });  
    studentList.Add(new Student(){ StudentName = "Steve" });  
    studentList.Add(new Student(){ StudentName = "Ram" });  
  
    ViewData["students"] = studentList;  
  
    return View();  
}
```

Script Code

```
<ul>  
@foreach (var std in ViewData["students"] as IList<Student>)  
{  
    <li>  
        @std.StudentName  
    </li>  
}
```


Create Layout View

- ▶ Right click on shared folder -> select Add -> click on **New Item..**



_myLayoutPage.cshtml:

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

Adding RenderSection at Layout View

```
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>@ViewBag.Title</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
</head>
<body>
    <div>
        @RenderBody()
    </div>
    <footer class="panel-footer">
        @RenderSection("footer", true)
    </footer>
</body>
</html>
```

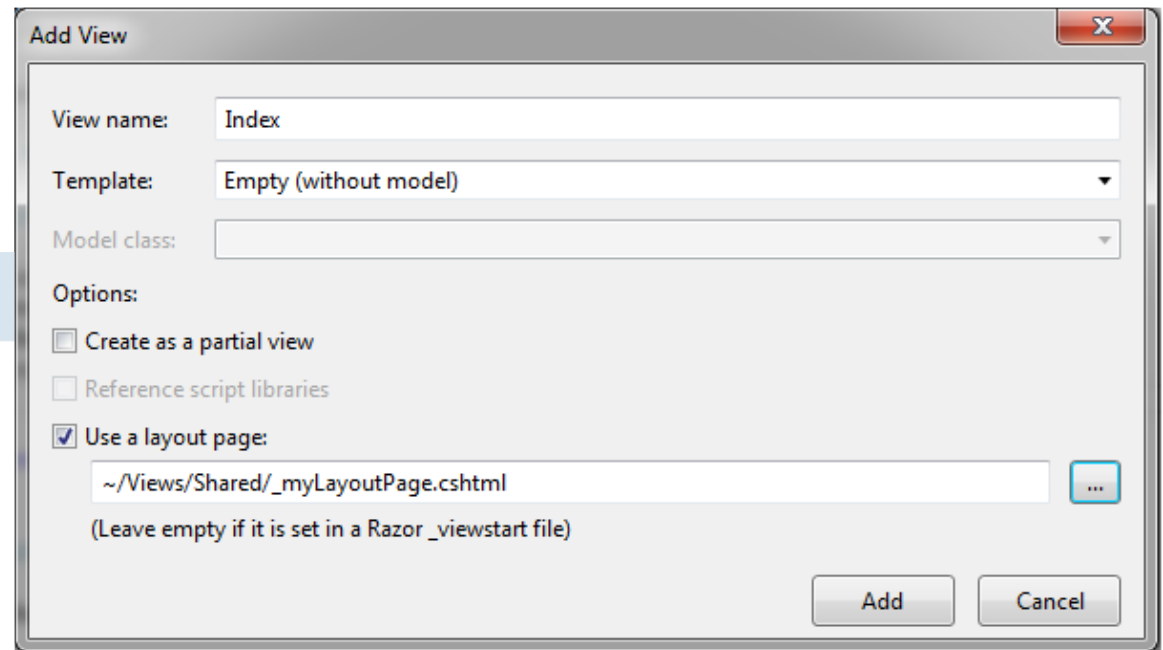


Use this _myLayoutPage.cshtml with the Index view of HomeController.

Index view:

```
@{
    ViewBag.Title = "Home Page";
    Layout = "~/Views/Shared/_myLayoutPage.cshtml";
}

<div class="jumbotron">
    <h2>Index</h2>
</div>
<div class="row">
    <div class="col-md-4">
        <p>This is body.</p>
    </div>
    @section footer{
        <p class="lead">
            This is footer section.
        </p>
    }
</div>
```



Add View

View name: Index

Template: Empty (without model)

Model class:

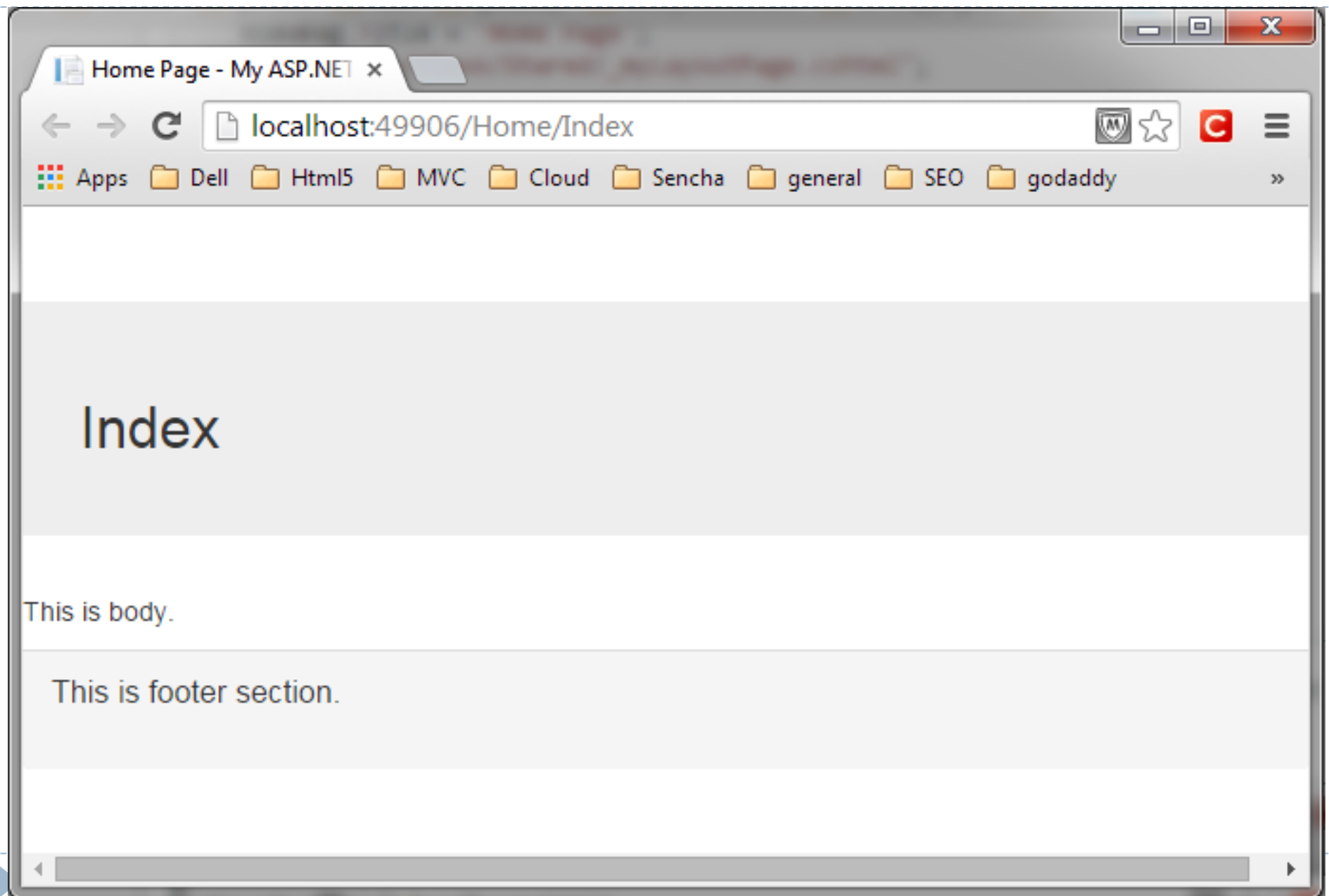
Options:

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

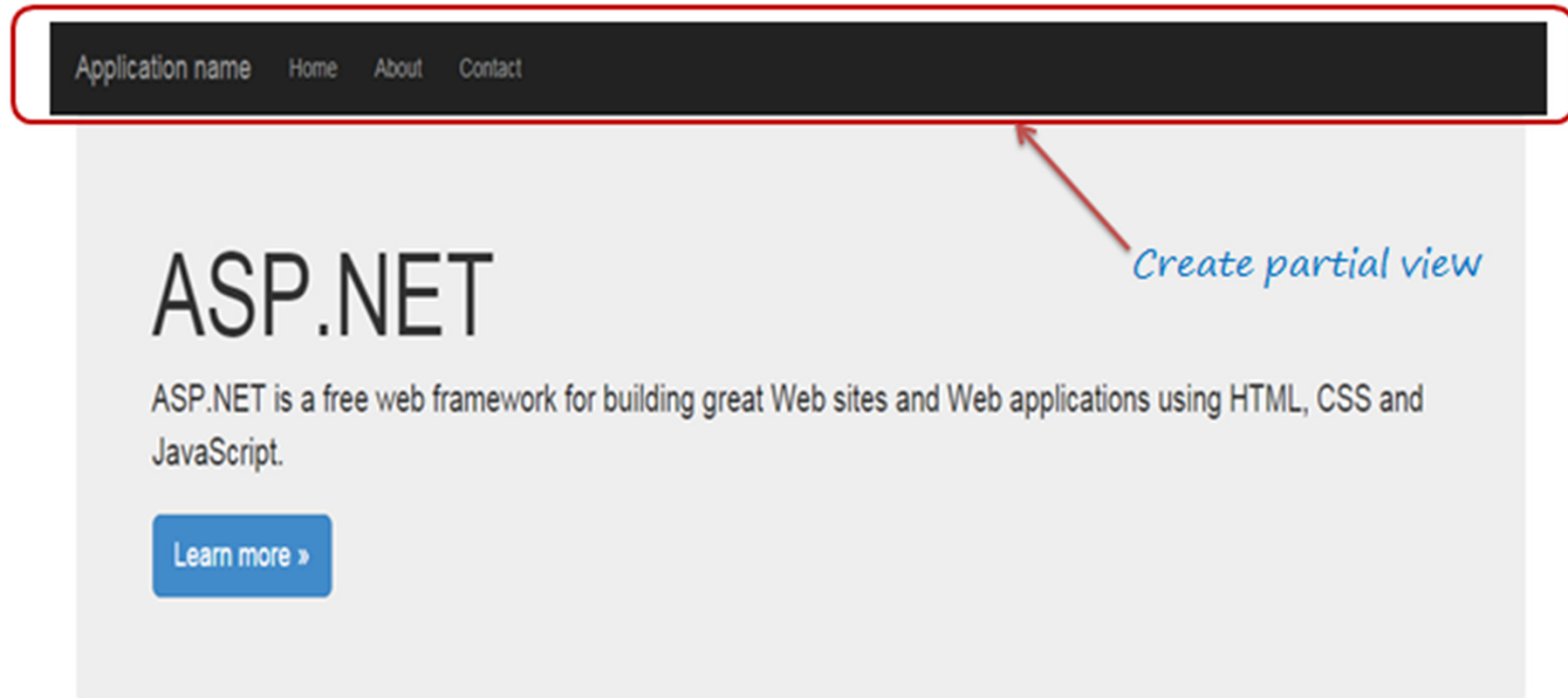
~/Views/Shared/_myLayoutPage.cshtml

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

Add Cancel



Partial View



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.

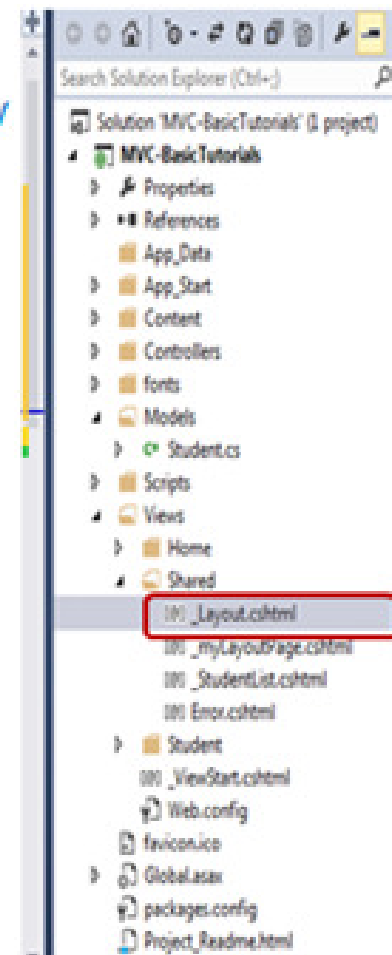
[Learn more »](#)



Partial View

```
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>@ViewBag.Title - My ASP.NET Application</title>
  @Styles.Render("~/Content/css")
  @Scripts.Render("~/bundles/modernizr")
</head>
<body>
  <div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
      <div class="navbar-header">
        <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        @Html.ActionLink("Application name", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })
      </div>
      <div class="navbar-collapse collapse">
        <ul class="nav navbar-nav">
          <li>@Html.ActionLink("Home", "Index", "Home")</li>
          <li>@Html.ActionLink("About", "About", "Home")</li>
          <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
        </ul>
      </div>
    </div>
  </div>
</body>
</html>
```

Move it to partial view



Create a New Partial View

- ▶ Right click on Shared folder -> select **Add** -> click on **View..**

Add View

View name: _HeaderNavBar

Template: Empty (without model)

Model class:

Options:

☒ Create as a partial view

☐ Reference script libraries

☒ Use a layout page:

~/_Views/Shared/_Layout.cshtml

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

Add Cancel

Example: Partial View _HeaderNavBar.cshtml

```
<div class="navbar navbar-inverse navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      @Html.ActionLink("Application name", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })
    </div>
    <div class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li>@Html.ActionLink("Home", "Index", "Home")</li>
        <li>@Html.ActionLink("About", "About", "Home")</li>
        <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
      </ul>
    </div>
  </div>
</div>
```


Example: Html.RenderPartial()

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - My ASP.NET Application</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
</head>
<body>
    @{
        Html.RenderPartial("_HeaderNavBar");
    }
    <div class="container body-content">
        @RenderBody()

        <hr />
        <footer>
            <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
        </footer>
    </div>
    @Scripts.Render("~/bundles/jquery")
    @Scripts.Render("~/bundles/bootstrap")
    @RenderSection("scripts", required: false)
</body>
</html>
```