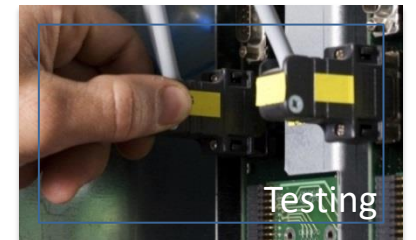
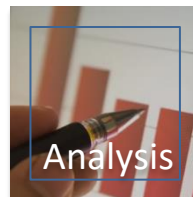


Basic ASP.NET MVC Project

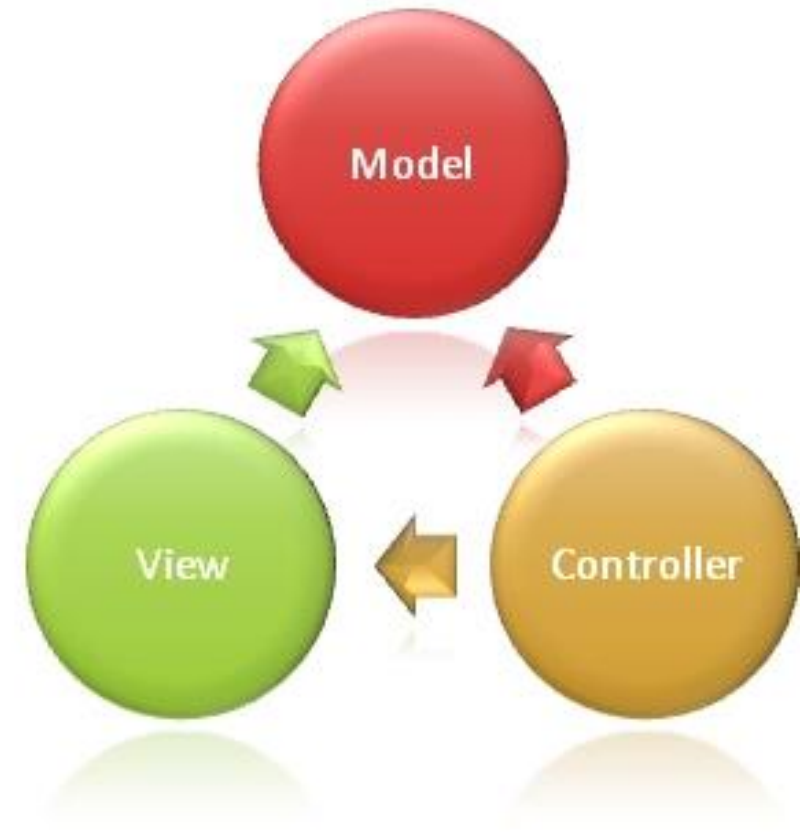


Topics

- ASP.NET MVC
- Routing
- Convention Over Configuration
- Layout

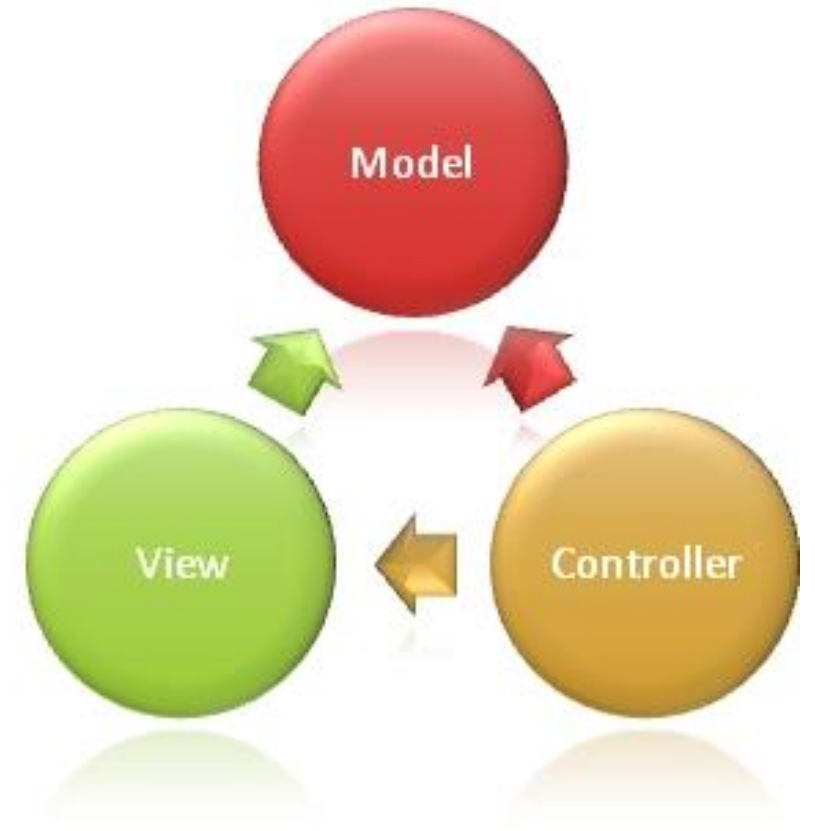
ASP.NET MVC based applications

- ASP.NET MVC based applications are made up of:
- **Models**: Composed of classes that model the data of the application. They are often referred to as POCO (Plain Old CLR Objects).
 - They are used to enforce any business logic.
- **Views**: These are template files that generate HTML to sent to the web browser.
 - Deals with displaying data from a model. Ideally, they should not contain any business logic.
- **Controllers**: These are classes that process incoming requests. They obtain data model, and return this data to a view for displaying as HTML.
 - For example, the controller handles query-string values, and passes these values to the model, which in turn queries the database by using the values.



MVC Advantages

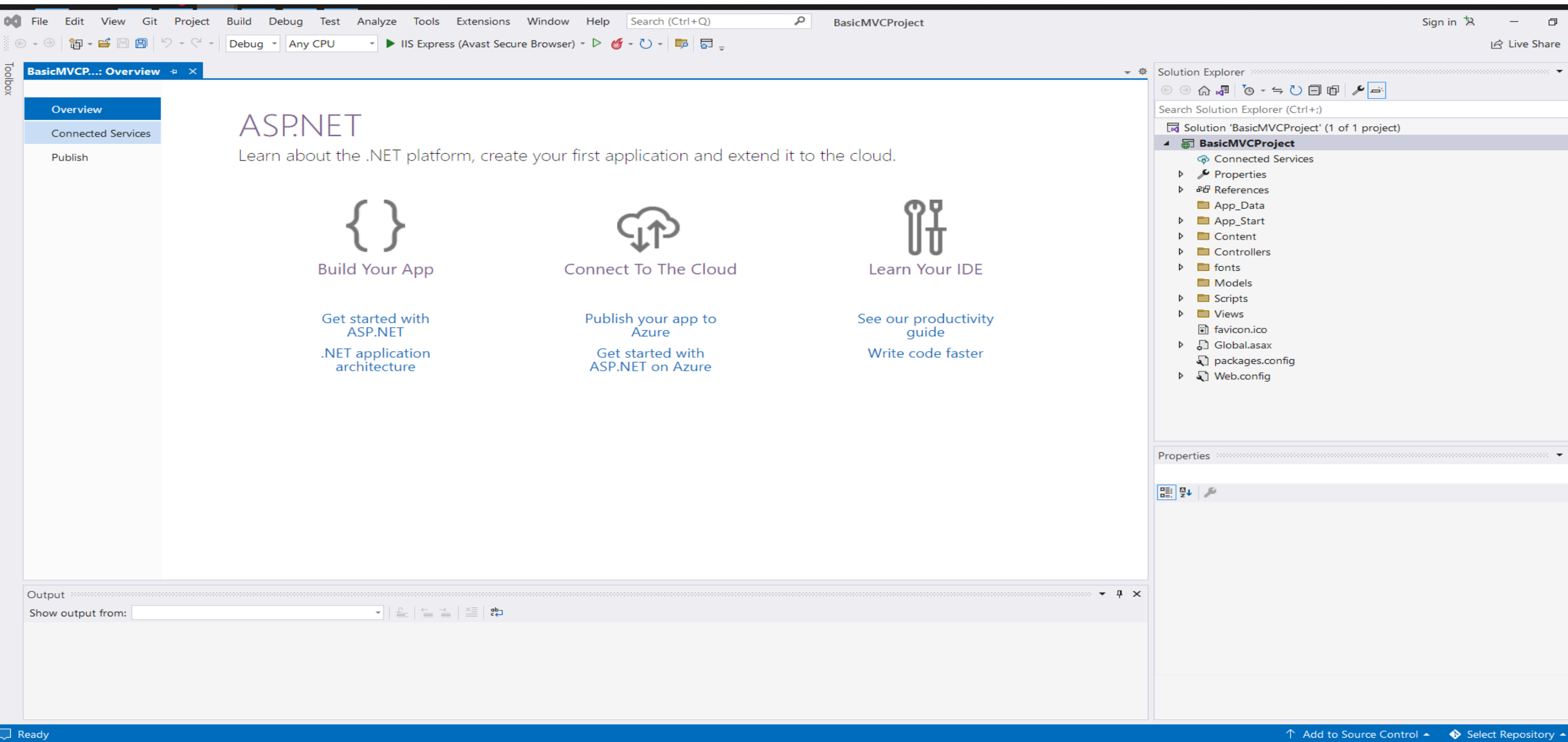
- Separation of concern
- Testable
- Maintainable



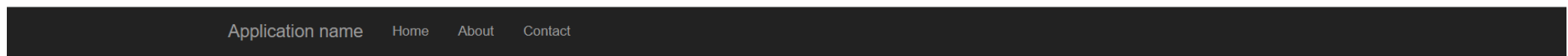
ASP.NET MVC Folder Structure

- **App_Data**
 - application data files like LocalDB, .mdf files, XML files, and other data related files
- **App_Start**
 - files that will be executed when the application starts including config files like BundleConfig.cs, FilterConfig.cs, RouteConfig.cs etc
- **Content**
 - static files like CSS files, images, and icons files
- **Controllers**
 - C# class files for the controllers
- **Fonts**
 - custom font files
- **Models**
 - model class files
- **Scripts**
 - JavaScript files
- **Views**
 - *.aspx, *.ascx and *.cshtml files
- **Views\Shared**
 - global Master Pages, Error Pages and User Controls used by multiple controllers
- **Web.config**
 - configurations
- **Packages.config**
 - packages and versions installed in the application

How the Home Page Works



How the Home Page Works



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 »](#)

Routing

- ASP.NET MVC uses ASP.NET routing to control how the application translates URLs. It also controls which controllers and methods the incoming HTTP requests target and hence which views to display.
- RouteConfig.cs file, which is located in the App_Start folder. The code contained in this file defines rules that the application uses to target the correct controller and method depending on the URL entered into the web browser.

App_Start\RouteConfig.cs

```
• using System.Web.Mvc;
• using System.Web.Routing;

• namespace BasicMVCProject
• {
•     public class RouteConfig
•     {
•         public static void RegisterRoutes(RouteCollection routes)
•         {
•             routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

•             routes.MapRoute(
•                 name: "Default",
•                 url: "{controller}/{action}/{id}",
•                 defaults: new { controller = "Home", action = "Index", id =
UrlParameter.Optional }
•             );
•         }
•     }
• }
```

Global.assx

```
using System.Web.Mvc;
using System.Web.Optimization;
using System.Web.Routing;

namespace BasicMVCProject
{
    public class MvcApplication : System.Web.HttpApplication
    {
        protected void Application_Start()
        {
            AreaRegistration.RegisterAllAreas();
            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);
        }
    }
}
```

How the Home Page Works?

- The home page shown above is displayed because the web site made a request to the URL /home/Index.
- If you add /home/index to the end of the URL, the page displayed will not be changed.
- The URL tells the MVC Web Application that a request has been made to the Index action method of the Home Controller.
- The method is found in the HomeController.cs file which contains a simple line of code `return View()` and this tells the website to open the corresponding view.

HomeController.cs

```
• using System.Web.Mvc;

• namespace BasicMVCProject.Controllers
• {
•     public class HomeController : Controller
•     {
•         public ActionResult Index()
•         {
•             return View();
•         }

•         public ActionResult About()
•         {
•             ViewBag.Message = "Your application description page.";

•             return View();
•         }

•         public ActionResult Contact()
•         {
•             ViewBag.Message = "Your contact page.";

•             return View();
•         }
•     }
• }
```

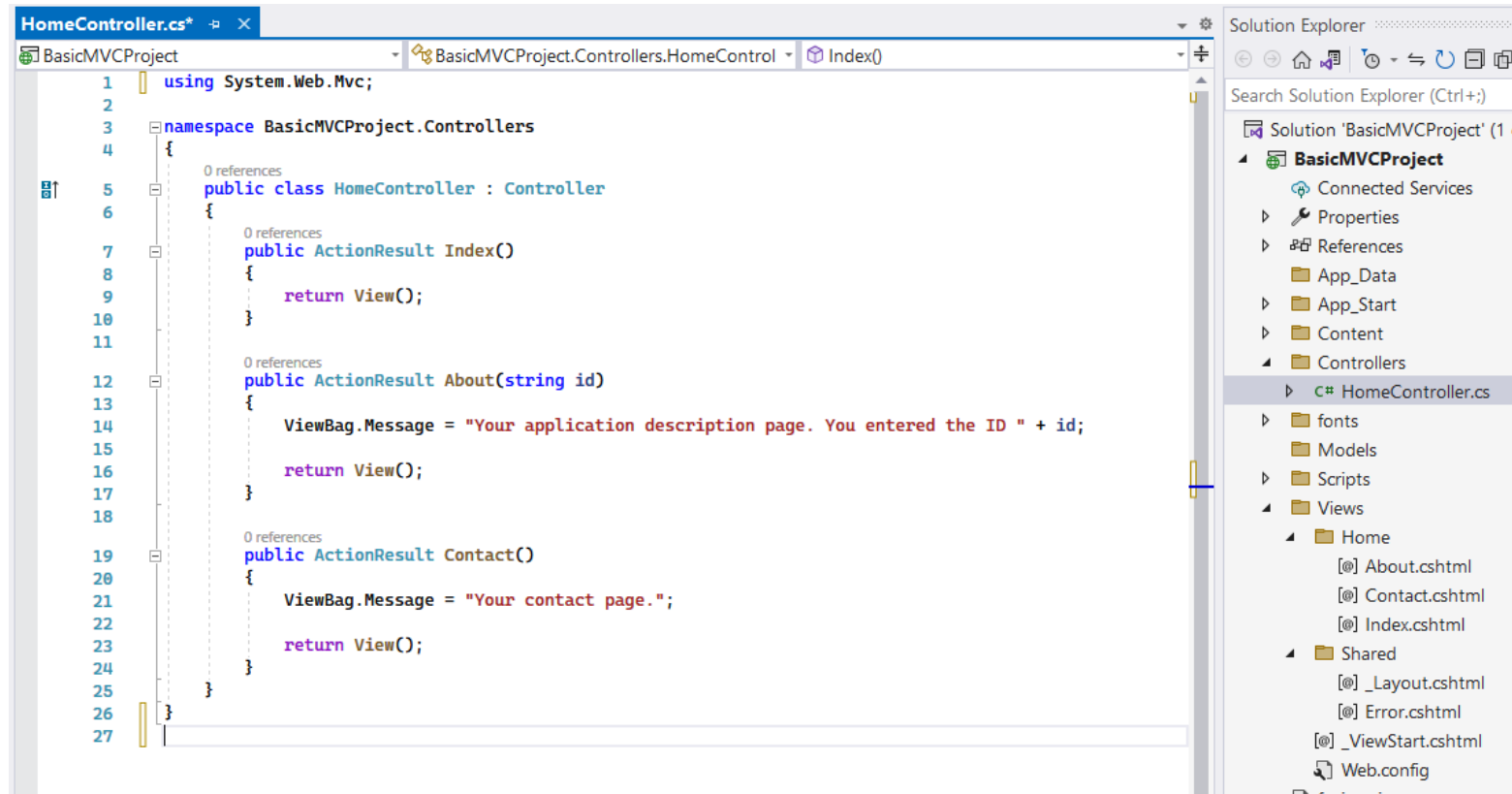
Home

- The corresponding view file is found under Views\Home\Index.cshtml.
- By convention, this is how ASP.NET MVC projects work with the default view for a controller method found in the **Views\ControllerName\MethodName.cshtml** path.
- The Views\Home\Index.cshtml file contains the HTML corresponding to the text shown in the home page.

HomeController.cs

URL /Home/Index. By convention, ASP.NET MVC projects work with the default view for a controller method found in the path:

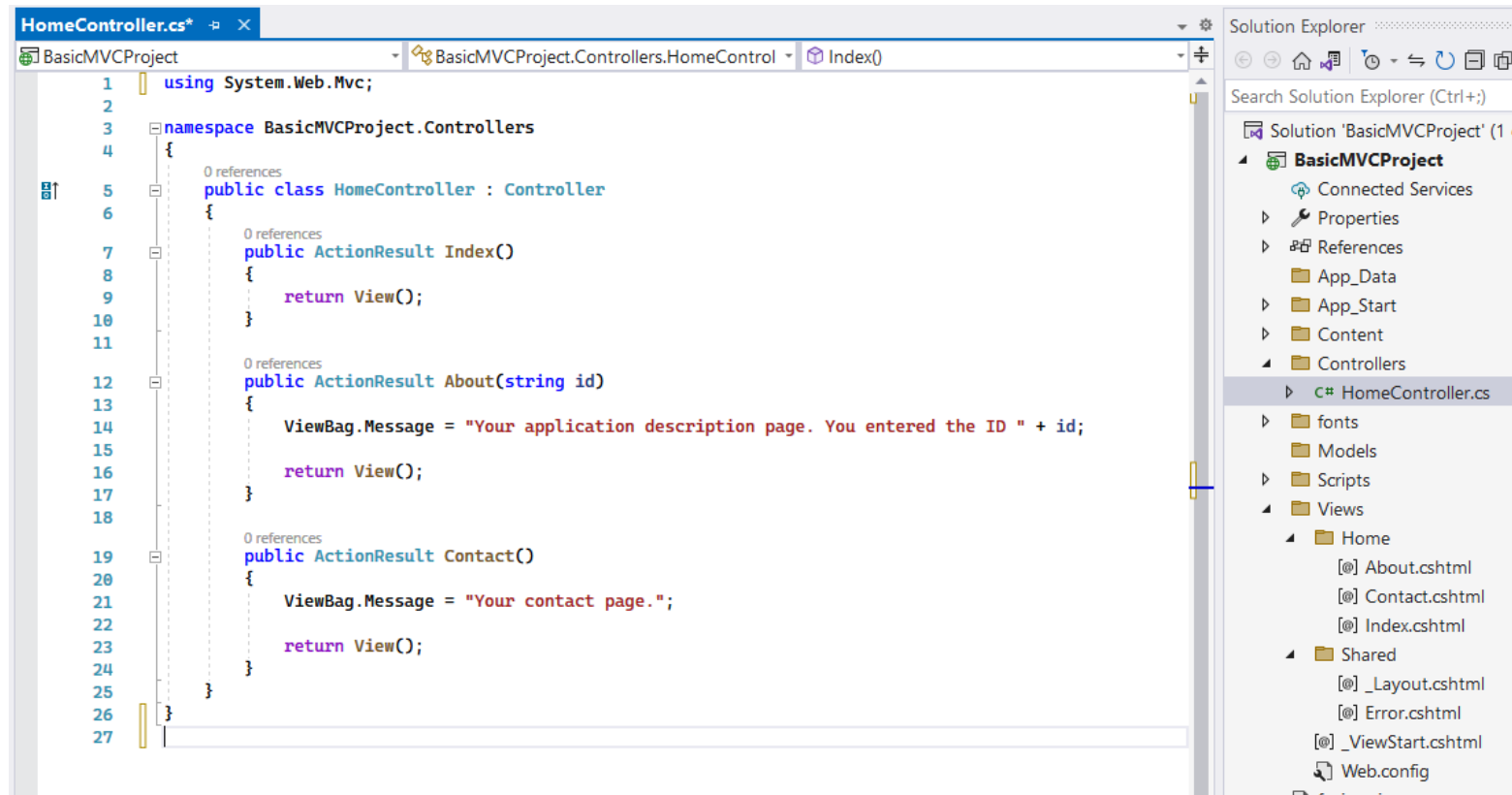
- Views\ControllerName\MethodName.cshtml
- Views\Home\Index.cshtml



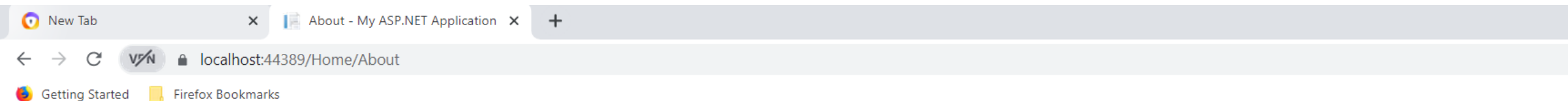
HomeController.cs

Views\ControllerName\MethodName.cshtml

- URL /Home/About =>Views\Home\ About.cshtml
- URL /Home/Contact =>Views\Home\Contact.cshtml



About



Application name

[Home](#)

[About](#)

[Contact](#)

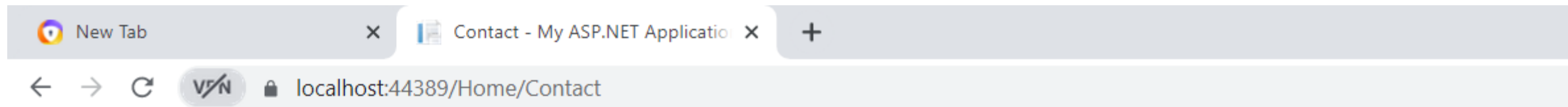
About.

Your application description page. You entered the ID

Use this area to provide additional information.

© 2022 - My ASP.NET Application

Contact



Getting Started Firefox Bookmarks

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

Convention Over Configuration

- ASP.NET MVC follows Convention Over Configuration principle for naming.
- Coding by Convention
- a software design paradigm
- don't repeat yourself (DRY) principles

https://localhost:44389/Home/About/5

localhost:44389/Home/About/5

Firefox Bookmarks

Application name Home About Contact

About.

Your application description page. You entered the ID 5

Use this area to provide additional information.

© 2022 - My ASP.NET Application

HomeController.cs

```
• using System.Web.Mvc;

• namespace BasicMVCProject.Controllers
• {
•     public class HomeController : Controller
•     {
•         public ActionResult Index()
•         {
•             return View();
•         }

•         public ActionResult About(string id)
•         {
•             ViewBag.Message = "Your application description page. You entered the ID " + id;

•             return View();
•         }

•         public ActionResult Contact()
•         {
•             ViewBag.Message = "Your contact page.";

•             return View();
•         }
•     }
• }
```

ViewBag

- ViewBag is a dynamic object.
- It is one way of passing data to a view.
- ViewBag is not strongly typed; i.e. it can contain any random data

```
using System.Web.Mvc;

namespace BasicMVCProject.Controllers
{
    0 references
    public class HomeController : Controller
    {
        0 references
        public ActionResult Index()
        {
            return View();
        }

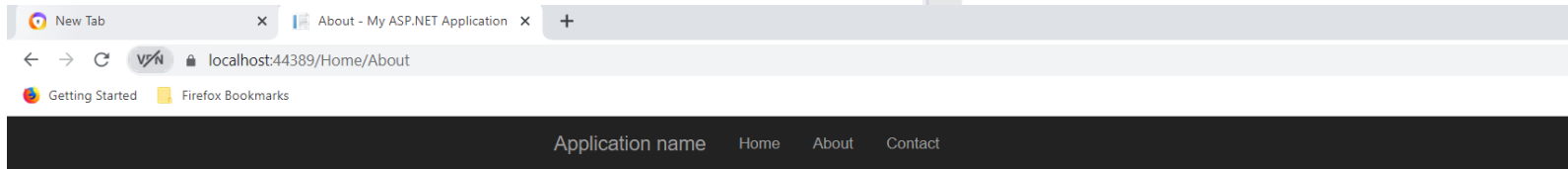
        0 references
        public ActionResult About(string id)
        {
            ViewBag.Message = "Your application description page. You entered the ID " + id;

            return View();
        }

        0 references
        public ActionResult Contact()
        {
            ViewBag.Message = "Your contact page.";

            return View();
        }
    }
}
```

```
About.cshtml
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
```

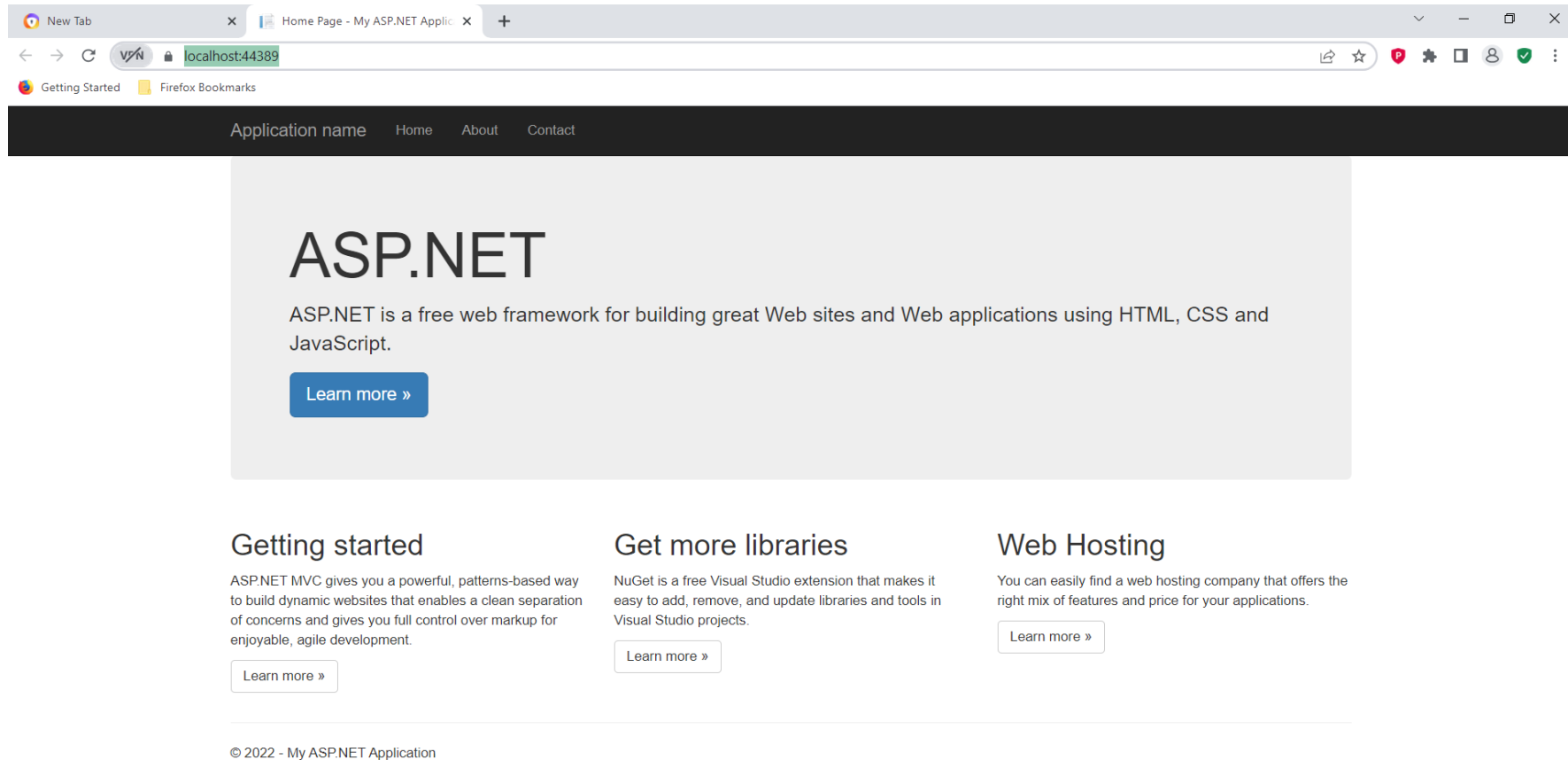


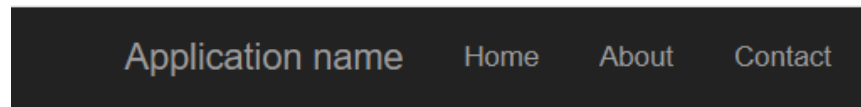
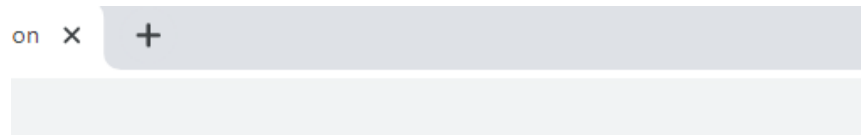
About.

Your application description page. You entered the ID

Use this area to provide additional information.

Home Page





About.

Your application description page.

Use this area to provide additional information.

© 2022 - My ASP.NET Application

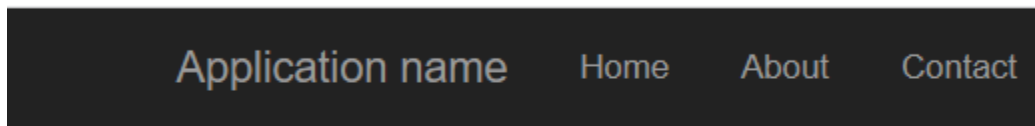
Layout Page



ASP.NET

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

[Learn more »](#)



Contact.

Your contact page.

One Microsoft Way
Redmond, WA 98052-6399

7 105 555 0100

The purpose of Layout Page

- You have noticed that when we navigate to different pages, the web site shows the same heading and navigation menu. This is by using a layout page. The layout page is contained in the file `Views\Shared_Layout.cshtml` as follows:
- It contains HTML elements containing the site navigation.
- The `RenderBody()` Code in the file is responsible for displaying the content of the view being requested.
- The website knows to use the `Views\Shared_Layout.cshtml` file as the main layout page because of the contents of the `_ViewStart.cshtml`, which specifies the layout page to use.

Views\Shared_Layout.cshtml

```
<!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>
  <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" title="more options">
          <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>
  <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>
```

Views_ViewStart.cshtml

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

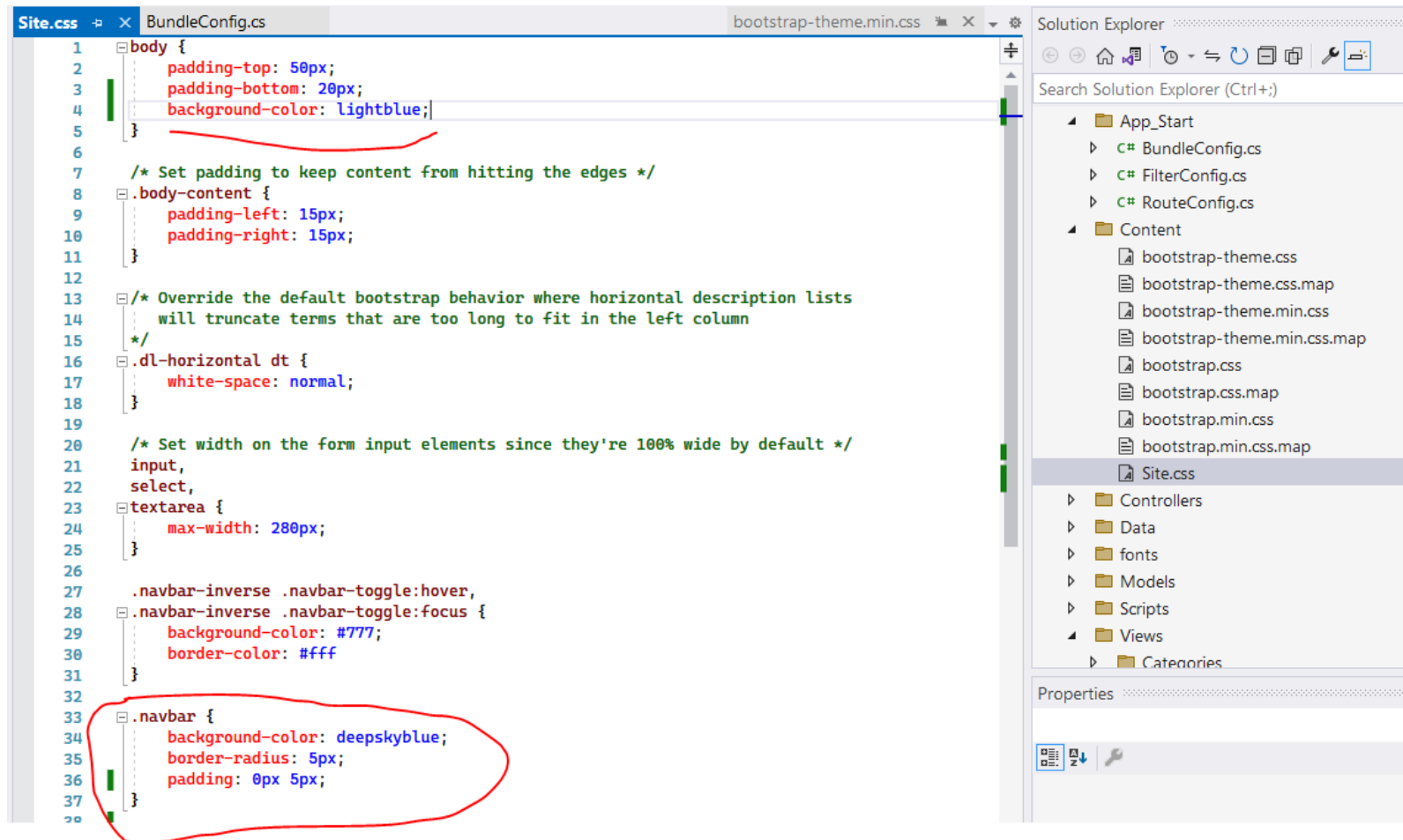
Razor Syntax (C#)

- Add code to a page using the @ character
- Enclose code blocks in braces @{ ... }
- Code statements end with a semicolon inside a block;
- Variables are declared with the var keyword, or the datatype (int, string, etc.)
- Use variables to store values
- Strings are enclosed with double quotation marks
- Code is case sensitive
- C# files have the extension .cshtml

Bootstrap

- HTML, CSS, and JavaScript framework
- For developing responsive, mobile-first websites
- Contains HTML, CSS and JavaScript-based design templates for
 - Typography
 - Forms
 - Buttons
 - Navigation
 - Other interface components.

Content: Site.css



The screenshot displays the Visual Studio IDE with the `Site.css` file open in the editor. The code defines styles for the body, body-content, dl-horizontal, and navbar. The `body` selector is highlighted with a red bracket, and the `navbar` selector is circled in red. The Solution Explorer on the right shows the project structure, including the `Content` folder which contains the `bootstrap-theme.min.css` and `Site.css` files.

```
1 body {  
2     padding-top: 50px;  
3     padding-bottom: 20px;  
4     background-color: lightblue;  
5 }  
6  
7 /* Set padding to keep content from hitting the edges */  
8 .body-content {  
9     padding-left: 15px;  
10    padding-right: 15px;  
11 }  
12  
13 /* Override the default bootstrap behavior where horizontal description lists  
14    will truncate terms that are too long to fit in the left column  
15 */  
16 .dl-horizontal dt {  
17     white-space: normal;  
18 }  
19  
20 /* Set width on the form input elements since they're 100% wide by default */  
21 input,  
22 select,  
23 textarea {  
24     max-width: 280px;  
25 }  
26  
27 .navbar-inverse .navbar-toggle:hover,  
28 .navbar-inverse .navbar-toggle:focus {  
29     background-color: #777;  
30     border-color: #fff  
31 }  
32  
33 .navbar {  
34     background-color: deepskyblue;  
35     border-radius: 5px;  
36     padding: 0px 5px;  
37 }  
38
```

Solution Explorer:

- App_Start
 - BundleConfig.cs
 - FilterConfig.cs
 - RouteConfig.cs
- Content
 - bootstrap-theme.css
 - bootstrap-theme.css.map
 - bootstrap-theme.min.css
 - bootstrap-theme.min.css.map
 - bootstrap.css
 - bootstrap.css.map
 - bootstrap.min.css
 - bootstrap.min.css.map
 - Site.css**
- Controllers
- Data
- fonts
- Models
- Scripts
- Views
 - Categories

BundleConfig: default

The screenshot displays the Visual Studio IDE with the `BundleConfig.cs` file open in the editor. The file is part of the `BasicMVCProject` solution. The code defines a `BundleConfig` class within the `BasicMVCProject` namespace, which implements the `RegisterBundles` method. The method configures bundles for jQuery, jQuery validation, Modernizr, and Bootstrap, and includes the site's CSS files.

```
1 using System.Web;
2 using System.Web.Optimization;
3
4 namespace BasicMVCProject
5 {
6     1 reference
7     public class BundleConfig
8     {
9         // For more information on bundling, visit https://go.microsoft.com/fwlink/?LinkId=301862
10        1 reference
11        public static void RegisterBundles(BundleCollection bundles)
12        {
13            bundles.Add(new ScriptBundle("~/bundles/jquery").Include(
14                "~/Scripts/jquery-{version}.js"));
15
16            bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(
17                "~/Scripts/jquery.validate*"));
18
19            // Use the development version of Modernizr to develop with and learn from. Then, when you're
20            // ready for production, use the build tool at https://modernizr.com to pick only the tests you need.
21            bundles.Add(new ScriptBundle("~/bundles/modernizr").Include(
22                "~/Scripts/modernizr-*"));
23
24            bundles.Add(new ScriptBundle("~/bundles/bootstrap").Include(
25                "~/Scripts/bootstrap.js"));
26
27            bundles.Add(new StyleBundle("~/Content/css").Include(
28                "~/Content/bootstrap.css",
29                "~/Content/site.css"));
30        }
31    }
32 }
```

The Solution Explorer on the right shows the project structure for `BasicMVCProject`, including files like `BundleConfig.cs`, `FilterConfig.cs`, `RouteConfig.cs`, and various CSS and JavaScript files in the `Content` folder.

BundleConfig: updated

The screenshot displays the Visual Studio IDE with the `BundleConfig.cs` file open in the editor. The file is part of the `BasicMVCProject` namespace and contains the `RegisterBundles` method. The code registers several bundles, including jQuery, jQuery validation, Modernizr, Bootstrap, and a CSS bundle. The Solution Explorer on the right shows the project structure, including the `Content` folder which contains the CSS files mentioned in the code.

```
using System.Web;
using System.Web.Optimization;

namespace BasicMVCProject
{
    1 reference
    public class BundleConfig
    {
        // For more information on bundling, visit https://go.microsoft.com/fwlink/?LinkId=301862
        1 reference
        public static void RegisterBundles(BundleCollection bundles)
        {
            bundles.Add(new ScriptBundle("~/bundles/jquery").Include(
                "~/Scripts/jquery-{version}.js"));

            bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(
                "~/Scripts/jquery.validate*"));

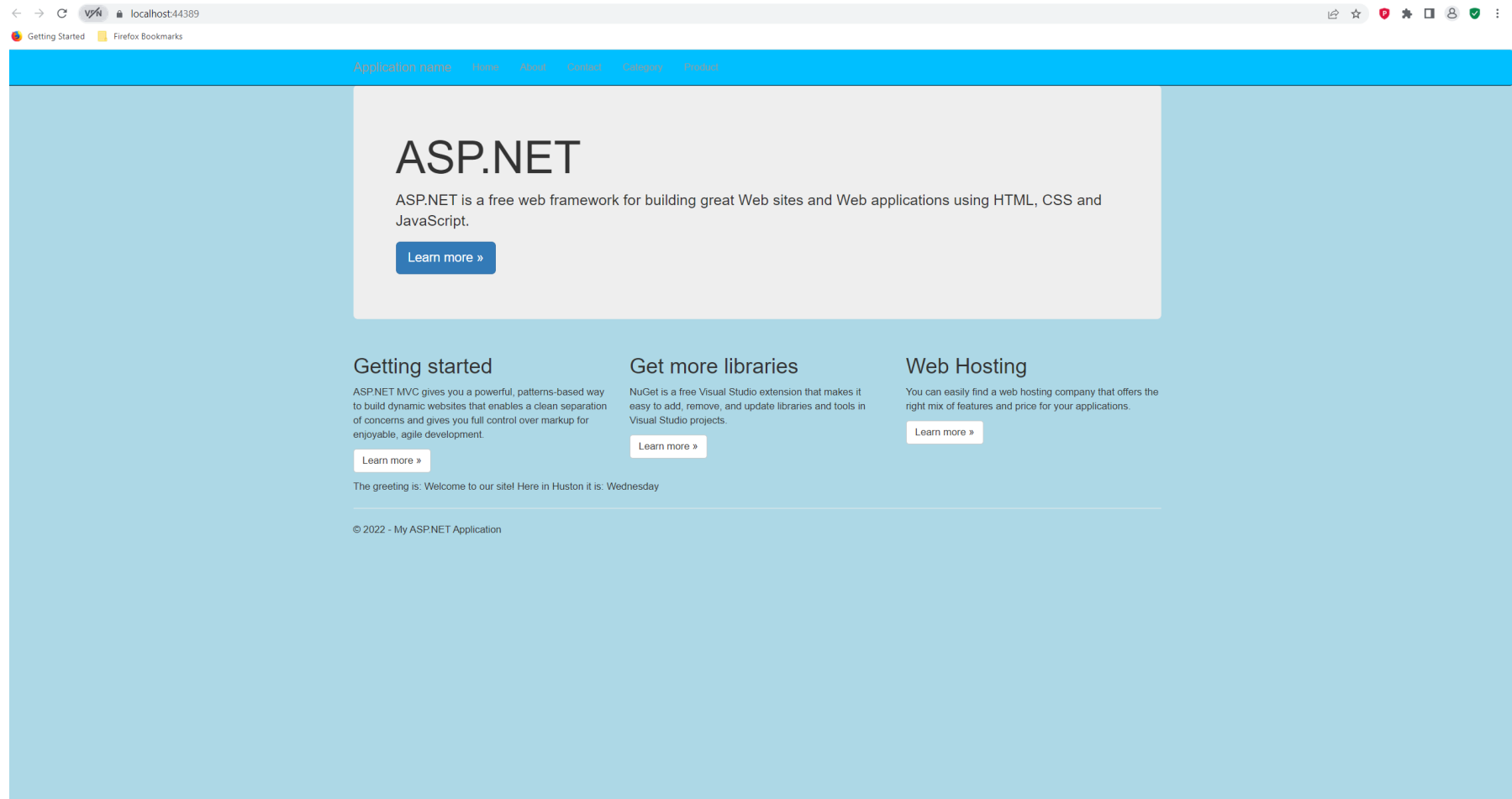
            // Use the development version of Modernizr to develop with and learn from. Then, when you're
            // ready for production, use the build tool at https://modernizr.com to pick only the tests you need.
            bundles.Add(new ScriptBundle("~/bundles/modernizr").Include(
                "~/Scripts/modernizr-*"));

            bundles.Add(new ScriptBundle("~/bundles/bootstrap").Include(
                "~/Scripts/bootstrap.js"));

            bundles.Add(new StyleBundle("~/Content/css").Include(
                "~/Content/bootstrap.css",
                "~/Content/Site.css"));
        }
    }
}
```

Solution Explorer (Ctrl+:)

- Solution 'BasicMVCProject' (1 of 1 project)
 - BasicMVCProject
 - Connected Services
 - Properties
 - References
 - App_Data
 - App_Start
 - BundleConfig.cs
 - FilterConfig.cs
 - RouteConfig.cs
 - Content
 - bootstrap-theme.css
 - bootstrap-theme.css.map
 - bootstrap-theme.min.css
 - bootstrap-theme.min.css.map
 - bootstrap.css
 - bootstrap.css.map
 - bootstrap.min.css
 - bootstrap.min.css.map
 - Site.css
 - Controllers



Add Email link



The screenshot shows the Visual Studio IDE with the `_Layout.cshtml` file open. The code is an ASP.NET Razor view layout. In the footer section (lines 41-48), an email link has been added to the address block. The Solution Explorer on the right shows the project structure, including folders for Controllers, Data, fonts, Models, Scripts, and Views, and files for Site.css, Web.config, and various configuration files.

```
25         <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
26         <li>@Html.ActionLink("Category", "Index", "Categories")</li>
27         <li>@Html.ActionLink("Product", "Index", "Products")</li>
28     </ul>
29 </div>
30 </div>
31 </div>
32 <div class="container body-content">
33     @RenderBody()
34     @{
35         var greeting = "Welcome to our site!";
36         var weekDay = DateTime.Now.DayOfWeek;
37         var greetingMessage = greeting + " Here in Huston it is: " + weekDay;
38     }
39     <p>The greeting is: @greetingMessage</p>
40     <hr />
41     <footer>
42         <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
43
44         <address>
45             Written by <a href="mailto:webmaster@massey.ac.nz">Tong Liu</a>.<br>
46             Visit us at: Massey.ac.nz<br>
47             New Zealand<br>
48         </address>
49
50     </footer>
51 </div>
52
53 @Scripts.Render("~/bundles/jquery")
54 @Scripts.Render("~/bundles/bootstrap")
55 @RenderSection("scripts", required: false)
56 </body>
57 </html>
58
```

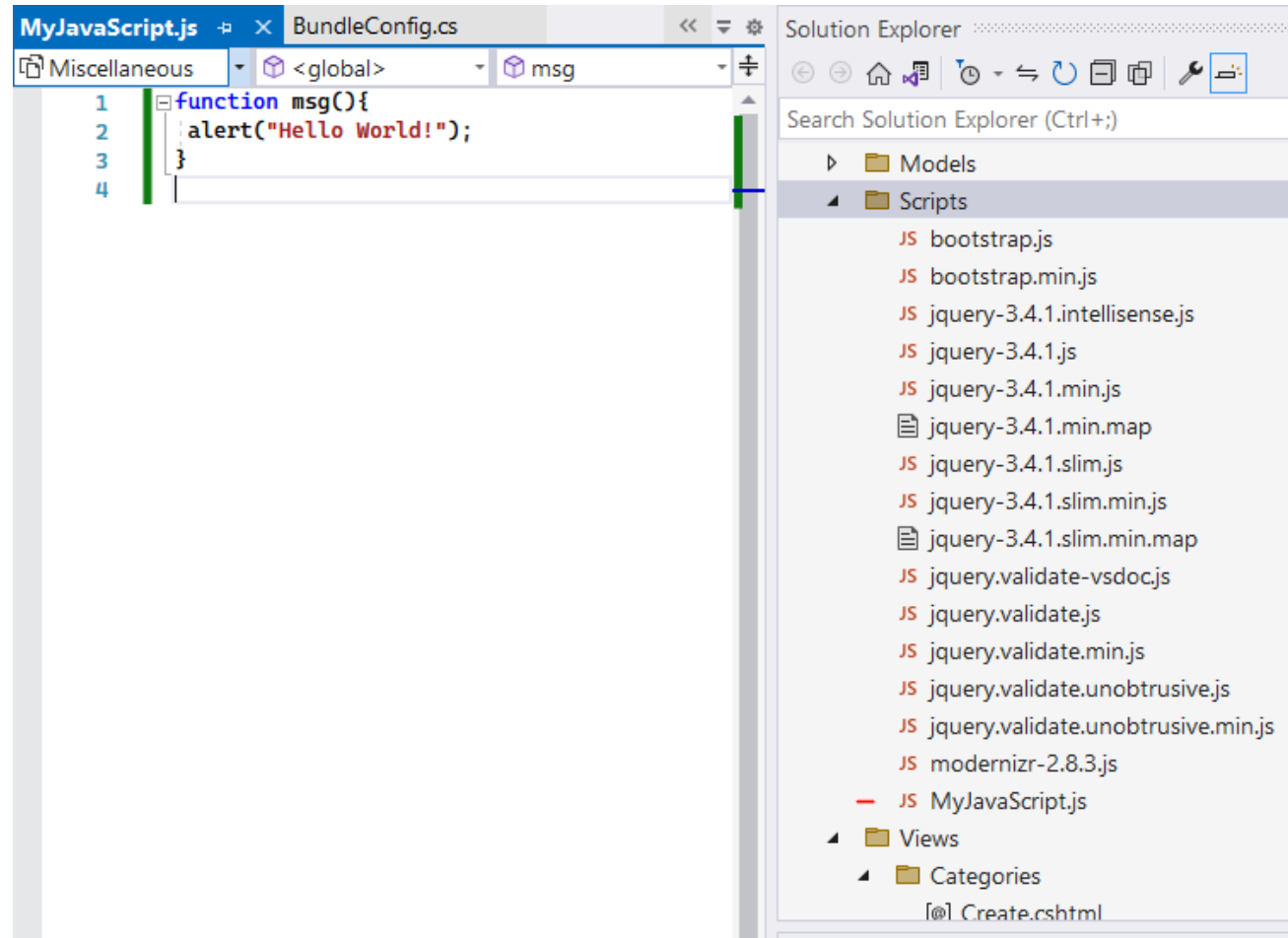
Solution Explorer:

- bootstrap.min.css.map
- Site.css
- Controllers
- Data
- fonts
- Models
- Scripts
- Views
 - Categories
 - Home
 - Products
 - Shared

Properties:

- [@] _Layout.cshtml
- [@] Error.cshtml
- [@] _ViewStart.cshtml
- Web.config
- favicon.ico
- Global.asax
- packages.config
- Web.config

JavaScript Files



Add JavaScript Files

The screenshot displays the Visual Studio IDE with the `BundleConfig.cs` file open in the editor. The file is part of the `BasicMVCProject` and contains the `RegisterBundles` method. A red line highlights the addition of `~/Scripts/MyJavaScript.js` to the bundle. The Solution Explorer on the right shows the project structure, including the `Scripts` folder which contains various JavaScript files, including `MyJavaScript.js`.

```
1 using System.Web;
2 using System.Web.Optimization;
3
4 namespace BasicMVCProject
5 {
6     1 reference
7     public class BundleConfig
8     {
9         // For more information on bundling, visit https://go.microsoft.com/fwlink/?LinkId=301862
10        1 reference
11        public static void RegisterBundles(BundleCollection bundles)
12        {
13            bundles.Add(new ScriptBundle("~/bundles/jquery").Include(
14                "~/Scripts/jquery-{version}.js"));
15
16            bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(
17                "~/Scripts/jquery.validate*"));
18
19            // Use the development version of Modernizr to develop with and learn from. Then, when you're
20            // ready for production, use the build tool at https://modernizr.com to pick only the tests you need
21            bundles.Add(new ScriptBundle("~/bundles/modernizr").Include(
22                "~/Scripts/modernizr-*"));
23
24            bundles.Add(new ScriptBundle("~/bundles/bootstrap").Include(
25                "~/Scripts/bootstrap.js", ~/Scripts/MyJavaScript.js);
26
27            bundles.Add(new StyleBundle("~/Content/css").Include(
28                "~/Content/bootstrap.css",
29                "~/Content/Site.css"));
30        }
31    }
32 }
```

Solution Explorer

- Models
- Scripts
 - bootstrap.js
 - bootstrap.min.js
 - jquery-3.4.1.intellisense.js
 - jquery-3.4.1.js
 - jquery-3.4.1.min.js
 - jquery-3.4.1.min.map
 - jquery-3.4.1.slim.js
 - jquery-3.4.1.slim.min.js
 - jquery-3.4.1.slim.min.map
 - jquery.validate-vsdoc.js
 - jquery.validate.js
 - jquery.validate.min.js
 - jquery.validate.unobtrusive.js
 - jquery.validate.unobtrusive.min.js
 - modernizr-2.8.3.js
 - MyJavaScript.js
- Views
 - Categories
 - [@] Create.cshtml

Properties

```
~\nz\web hosting\nz\
<p>You can easily find a web hosting company that offers the right mix of features and price for your app
<p><a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301867">Learn more &raquo;</a>
</div>
</div>
<div>
  <form>
    <input type="button" value="click" onclick="msg()" />
  </form>
</div>
<div>
  <p></p>
</div>
```

- [@] Index.cshtml
- ▲ Home
 - [@] About.cshtml
 - [@] Contact.cshtml
 - [@] Index.cshtml
- ▲ Products
 - [@] Create.cshtml
 - [@] Delete.cshtml
 - [@] Details.cshtml
 - [@] Edit.cshtml
 - [@] Index.cshtml

localhost:44389 says
Hello World!

OK

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 »](#)

click

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.

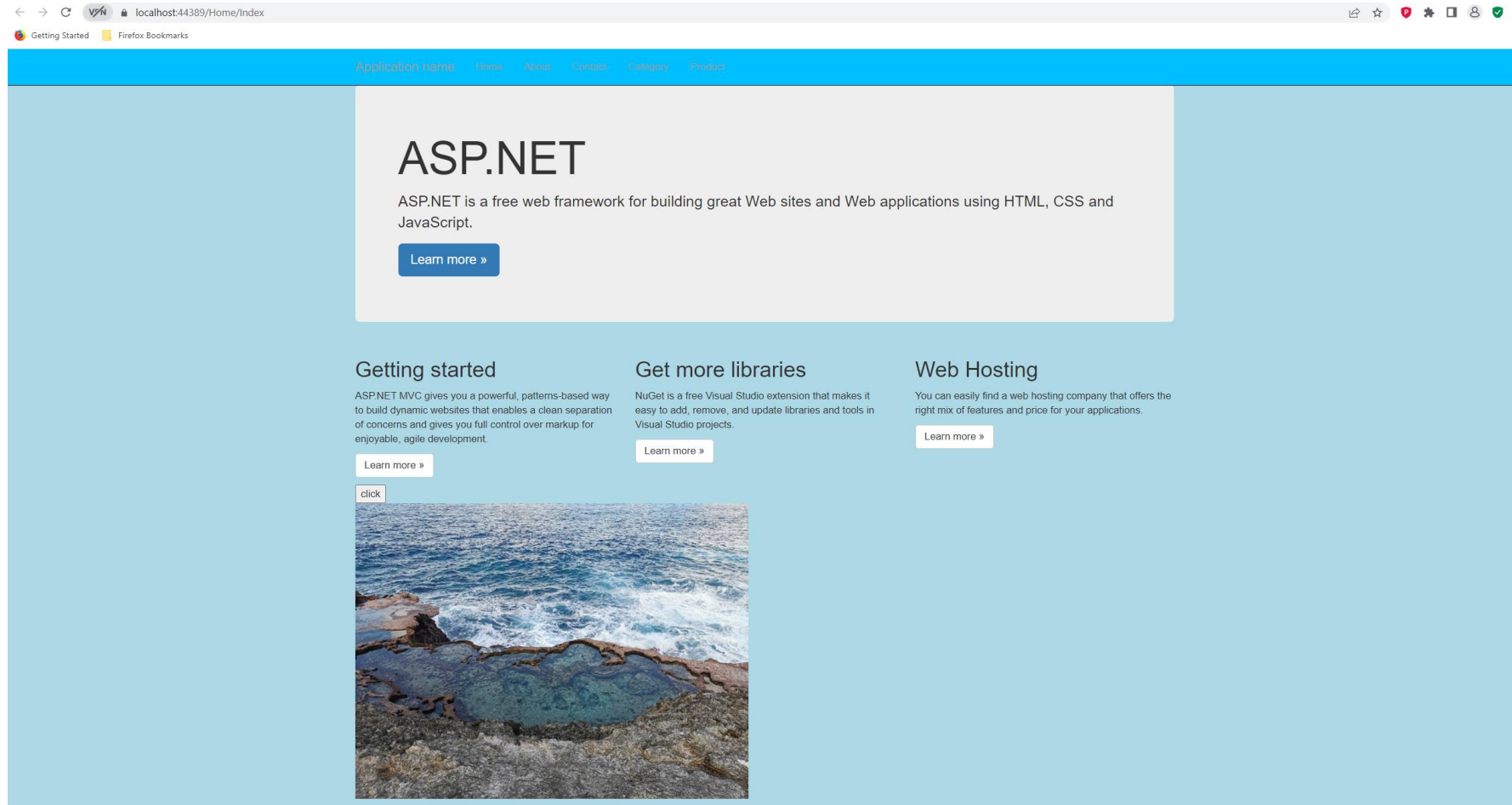
[Learn more »](#)

Web Hosting

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

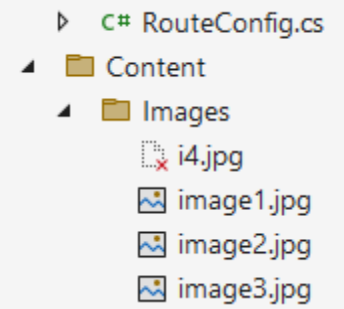
[Learn more »](#)

images

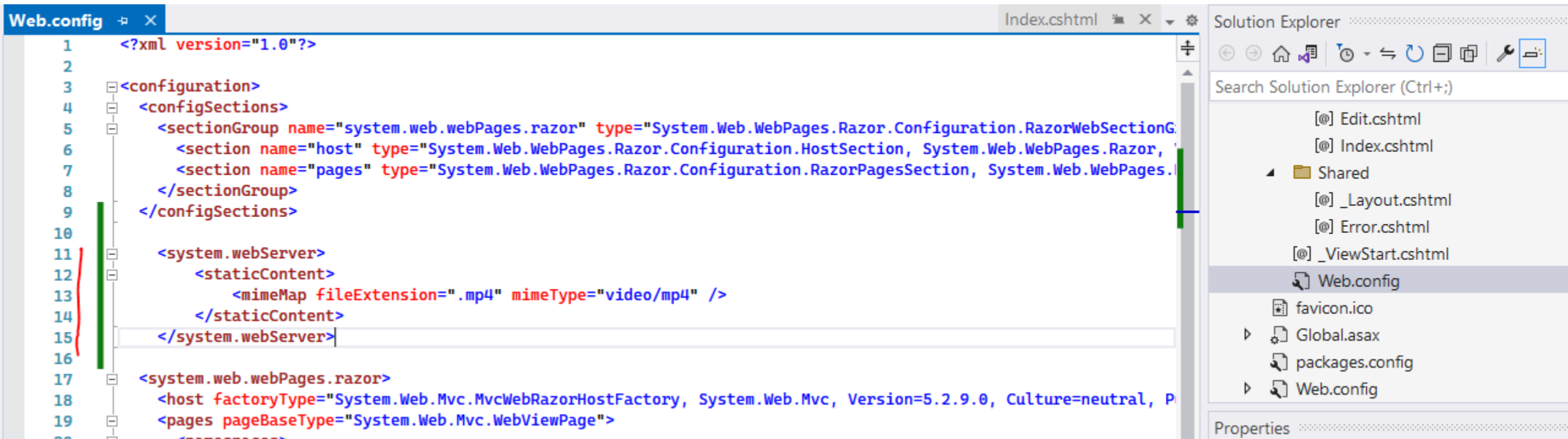


images

```
<div>
  <form>
    <input type="button" value="click" onclick="msg()" />
  </form>
</div>
<div>
  <p></p>
</div>
```



Web.config:Video



The image shows a screenshot of the Visual Studio IDE. The main window displays the `Web.config` file, which is an XML configuration file. The file is edited in the `Index.cshtml` tab. The code is as follows:

```
<?xml version="1.0"?>
<configuration>
  <configSections>
    <sectionGroup name="system.web.webPages.razor" type="System.Web.WebPages.Razor.Configuration.RazorWebSectionGroup, System.Web.WebPages.Razor, Version=3.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35">
      <section name="host" type="System.Web.WebPages.Razor.Configuration.HostSection, System.Web.WebPages.Razor, Version=3.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" required="true" />
      <section name="pages" type="System.Web.WebPages.Razor.Configuration.RazorPagesSection, System.Web.WebPages.Razor, Version=3.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
    </sectionGroup>
  </configSections>

  <system.webServer>
    <staticContent>
      <mimeMap fileExtension=".mp4" mimeType="video/mp4" />
    </staticContent>
  </system.webServer>

  <system.web.webPages.razor>
    <host factoryType="System.Web.Mvc.MvcWebRazorHostFactory, System.Web.Mvc, Version=5.2.9.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
    <pages pageBaseType="System.Web.Mvc.WebViewPage">
      <namespaces>
        <add namespace="System.Web.Mvc" />
        <add namespace="System.Web.Routing" />
      </namespaces>
    </pages>
  </system.web.webPages.razor>
</configuration>
```

The Solution Explorer on the right shows the project structure, including files like `Edit.cshtml`, `Index.cshtml`, `_Layout.cshtml`, `Error.cshtml`, `_ViewStart.cshtml`, `Web.config`, `favicon.ico`, `Global.asax`, `packages.config`, and `Web.config`.

Home/Index.cshtml




The image shows a Visual Studio IDE with the `Index.cshtml` file open in the editor. The file contains HTML code for a web page, including a heading, a paragraph, a button, an image, and a video player. The Solution Explorer on the right shows the project structure, including folders for `App_Data`, `App_Start`, `Content`, and `Images`, and files for `BundleConfig.cs`, `FilterConfig.cs`, `RouteConfig.cs`, and `Demo.mp4`.


```
25     </div>
26     <div class="col-md-4">
27         <h2>Web Hosting</h2>
28         <p>You can easily find a web hosting company that offers the right mix of features a
29         <p><a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301867">
30     </div>
31
32 </div>
33 <div>
34     <form>
35         <input type="button" value="click" onclick="msg()" />
36     </form>
37 </div>
38 <div>
39     <p></p>
40 </div>
41 <div>
42     <video width="320" height="240" controls="controls" autoplay="autoplay">
43         <source src="~/Content/Video/Demo.mp4" type="video/mp4" />
44         Your browser does not support the video tag.
45     </video>
46 </div>
47
```

Solution Explorer

- Properties
- References
- App_Data
- App_Start
 - BundleConfig.cs
 - FilterConfig.cs
 - RouteConfig.cs
- Content
 - Images
 - Video
 - Demo.mp4
- bootstrap-theme.css
- bootstrap-theme.css.map
- bootstrap-theme.min.css
- bootstrap-theme.min.css.map

Video

 localhost:44389/Home/Index


id  Firefox Bookmarks


Application nameHomeAboutContactCategoriesProduct

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 »

click





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 »

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

Learn more »

The greeting is: Welcome to our site! Here in Huston it is: Thursday

© 2022 - My ASP.NET Application
Written by [Tong Liu](#).
Visit us at: [Massey.ac.nz](#)
New Zealand

Summary

- ASP.NET MVC
- Routing
- Convention Over Configuration
- Layout