

Bài 2: Một dự án MVC mới và sử dụng nó để xây dựng một ứng dụng nhập dữ liệu MVC đơn giản

Mục tiêu: Sinh viên sau khi làm xong có thể hiểu được quy tắc làm việc của MVC

Biết cách tạo Controller, View, Model

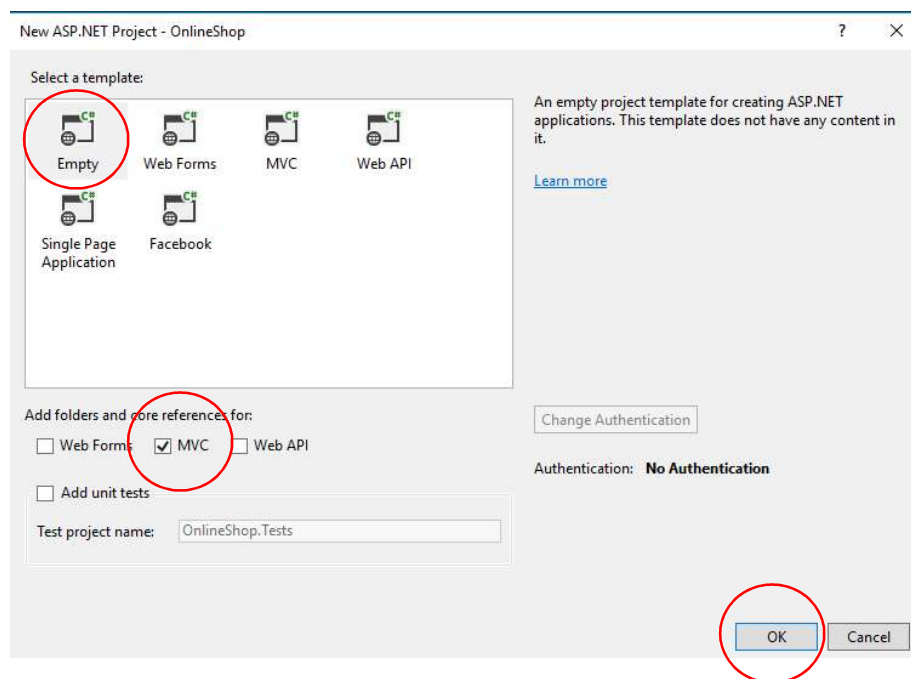
Có cái nhìn ban đầu về MVC Framework

Lý thuyết liên quan

Kỹ năng thực hành

Công việc 1: Tạo một Dự án MVC sử dụng template Empty của Visual.

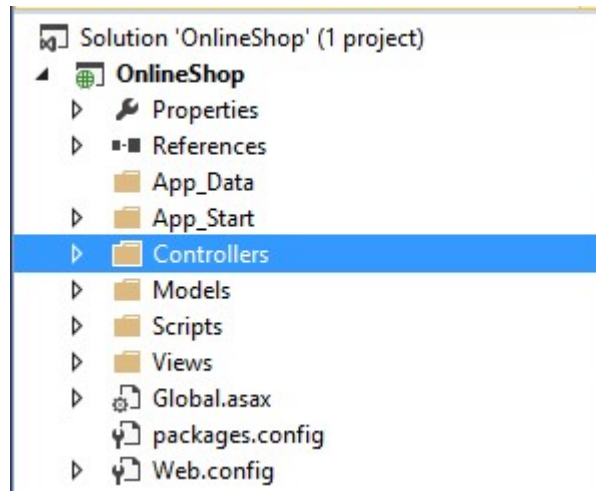
Bước 1: Tạo 1 dự án web có tên là OnlineShop



Chọn template Empty/Mô hình MVC

- Hãy chắc chắn bạn đã chọn phiên bản 4.5.1 của .NET Framework (hoặc hơn)
- Các tùy chọn mẫu dự án khác nhằm cung cấp cho bạn một khởi đầu hoàn chỉnh hơn cho dự án ASP.NET của bạn

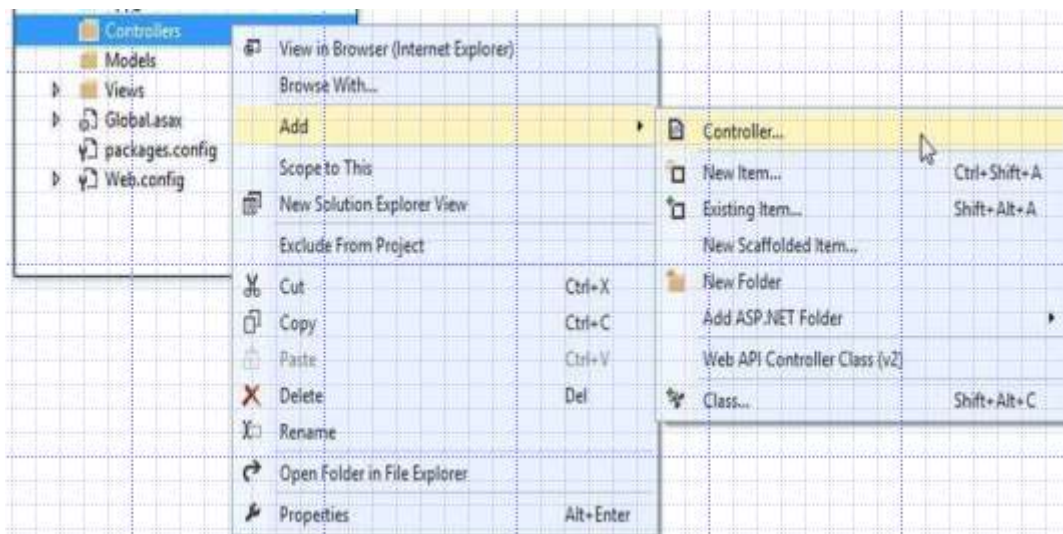
Bước 2: sau khi tạo dự án OnlineShop ta được:



Tại bước này nếu thực hiện debug chương trình, trình duyệt sẽ báo lỗi 404 (lý do do là dự án trống nên chưa có gì để hiển thị).

Công việc 2: Thêm một controller đầu tiên

Bước 1: Tại thư mục **Controller** R_click chọn Add/Controller
(Controller là gì? Hãy xem lại phần lý thuyết)



Bước 2: Chọn Scaffold / MVC5 Controller – Empty. Khi hộp thoại Add Controller xuất hiện hãy đặt tên Controller theo quy

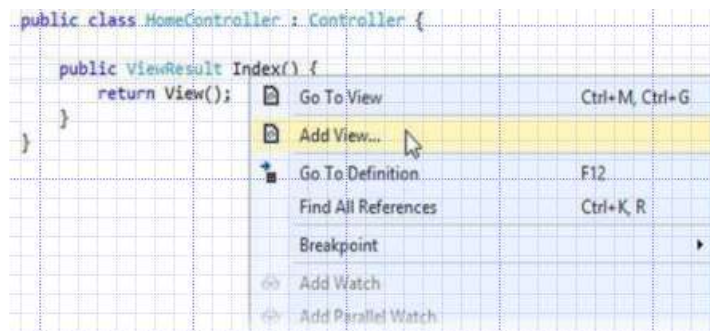
tắc:[Ten]Controller. Ví dụ ta sẽ đặt 1 controller là Home. [HomeController]. Lúc này vs sẽ tạo cho ta 1 file C# có tên là HomeController.cs

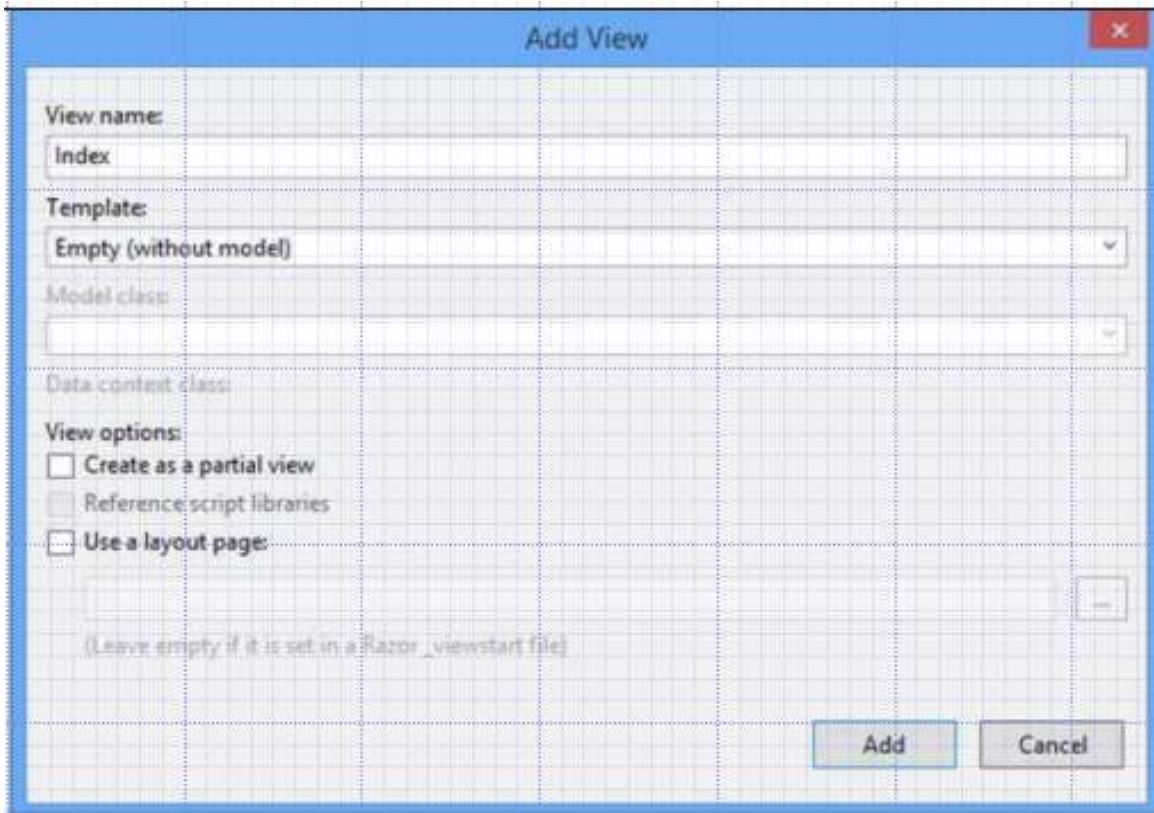
```
public class HomeController : Controller
{
    // GET: /Home/
    public ActionResult Index()
    {
        return View();
    }
}
```

Bước 2: thử biên dịch ứng dụng xem có vấn đề gì xảy ra

Công việc 3: tìm hiểu về URL và Tạo và tùy chỉnh 1 view.

Từ controller ban đầu để tạo view ta thực hiện nhấn R_click vào Action / chọn addview. MVC framework sẽ tạo 1 thư mục Home trong thư mục Views đồng thời tạo 1 file có tên là Index.cshtml (theo chuẩn razor).





//Phần code này theo cấu trúc Razor khai báo view này không có layout

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
</head>
<body>
    <div>
        Thêm thông tin gì đó ở đây để hiển thị lên trên trình duyệt
    </div>

</body>
</html>
```

Chú ý: khi trong controller không có trả về view mà thực hiện trả về một dữ liệu string, mvc sẽ hiểu rằng phải gán 1 chuỗi vào trình duyệt, khi ta trả về 1 view MVC sẽ tìm view đó theo đúng route để hiển thị view đó lên. Còn nếu trả về 1 dữ liệu

RdirectResult trình duyệt sẽ được chuyển hướng sang 1 URL khác...(xem trong phần Các phương thức Action)

Công việc 4: thêm vào Dynamic Output

Đây chính là 1 cách để truyền dữ liệu từ controller đến view bằng cách sử dụng ViewBag.

```
public ActionResult Index()
{
    int hour=DateTime.Now.Hour;

    ViewBag.Greeting = hour < 12 ? "lúc này là buổi sáng" : "Đã là buổi chiều rồi đó";

    return View();
}
```

Greeting : là 1 thuộc tính ta có thể đặt 1 tên bất kỳ.

Khi trả về View(). Thì nội dung file index.cs muốn hiển thị được thông tin của Greeting ta thực hiện

```
@ViewBag.Greeting
```

Công việc 5: Tạo 1 ứng dụng Data_Entry đơn giản

Cần lấy thông tin khách hàng và nội dung liên hệ của khách hàng đối với hệ thống onlineshop

Yêu cầu:

Trang chủ giới thiệu có link kẻ mở form liên hệ (index.cshtml);

Một form để lấy thông tin (Constact.cshtml)

Một trang để xác nhận và hiển thị lời cảm ơn (thanks.cshtml)

Bước 1: Thiết lập mô hình dữ liệu

Thêm lớp module

```
public class Feedback
{
    public int ID { get; set; }
    public string Name { get; set; }
    public string Phone { get; set; }
    public string Email { get; set; }
    public string Address { get; set; }
    public string Content { get; set; }
    public bool? Status { get; set; }
}
```

Trang chủ là Home/Index.cshtml(sử dụng để có link mở trang Feedback)

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
</head>
<body>
    <div>
        <h1> trình bày nội dung</h1>
    </div>
    @Html.ActionLink("Feedback", "FeedbackForm")
    //gồm 2 thuộc tính ActionLink("Nội dung hiển thị","Action trong Controller")

    <footer> Copyright 2017 by Admin</footer>
</body>
</html>
```

Trong file HomeController.cs

Ta viết thêm 1 Action

```
public ViewResult Feedback()  
{  
    return View();  
}
```

Tạo 1 View cho Action Feedback

```
@model OnlineShop.Models.FeedBack //chỉ định model sử dụng cho view này.  
{  
    Layout = null;  
}  
  
<!DOCTYPE html>  
  
<html>  
<head>  
    <meta name="viewport" content="width=device-width" />  
    <title>FeedBack</title>  
</head>  
<body>  
    @using (Html.BeginForm())  
    {  
        @Html.ValidationSummary()  
        <p>Name: @Html.TextBoxFor(x => x.Name) </p>  
        <p>Phone: @Html.TextBoxFor(x => x.Phone)</p>  
        <p>Email: @Html.TextBoxFor(x => x.Email)</p>  
        <p>Address: @Html.TextBoxFor(x => x.Address)</p>  
        <p>Content: @Html.TextAreaFor(x => x.Content)</p>  
  
        <p>  
            Status?  
            @Html.DropDownListFor(x => x.Status, new[] {  
                new SelectListItem() {Text = "Đồng ý", Value = bool.TrueString},  
                new SelectListItem() {Text = "Không đồng ý", Value = bool.FalseString},  
            }, "Choose an option")  
        </p>  
        <input type="submit" value="Submit" />  
    }  
  
    <div>  
        Thực hiện hiển thị form lấy thông tin trong phần này  
    </div>  
</body>  
</html>
```

Công việc 6: thực hiện thêm phương thức kiểm tra
Trong models thêm thuộc tính kiểm tra.

```
public int ID { get; set; }  
[Required(ErrorMessage = "Please enter your name")]
```

```
public string Name { get; set; }
[Required(ErrorMessage = "Please enter your's phone")]
public string Phone { get; set; }
[Required(ErrorMessage = "Please enter your's Email")]
public string Email { get; set; }
[Required(ErrorMessage = "Please enter your's Address")]
public string Address { get; set; }
[Required(ErrorMessage = "Please enter your's Content")]
public string Content { get; set; }
[Required(ErrorMessage = "Please enter your's Status")]
public bool? Status { get; set; }
```

Trong HomeController.cs sử dụng

```
public ViewResult Feedback(Feedback
feedback)
{
    if (ModelState.IsValid)
    {
        return View("Thanks", feedback);
    }
    else
    {
        return View();
    }
}
```

Trong view Feedback.cshtml

Thêm: @Html.ValidationSummary() vào cấu trúc

```
@using(Html.BeginForm){@Html.ValidationSummary() }
```

Hoàn tất dự án: chỉnh trang chỉ cho các nội dung

Tạo css cho dự án: Tạo 1 thư mục chứa css:Content/Style.css

Nội dung css để cho phần hiển thị lỗi

```
.field-validation-error {color: #f00;}
.field-validation-valid { display: none;}
.input-validation-error { border: 1px solid #f00; background-
color: #fee; }
.validation-summary-errors { font-weight: bold; color: #f00;}
.validation-summary-valid { display: none;}
```

Thêm vào tham chiếu ở phần header của file feedback.cshtml

```
<link rel="stylesheet" type="text/css" href=" ~ /Content/Styles.css" />
```

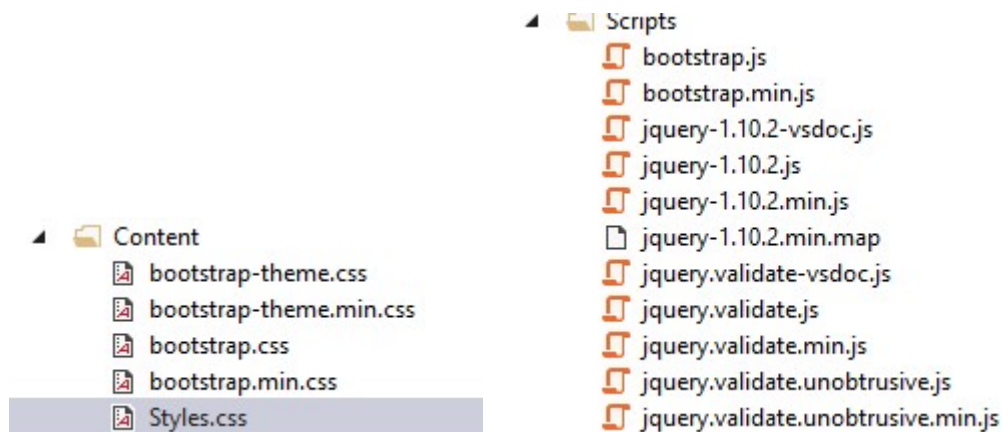

Thêm css cho phần nội dung

Sử dụng nuGet để cài đặt bootstrap cho dự án

Để cài đặt gói Bootstrap, chọn Library Package Manager/ Package Manager Console từ Tools của Visual Studio. Visual Studio sẽ mở dòng lệnh NuGet. Nhập lệnh và ấn Enter :

```
Install-Package -version 3.0.0 bootstrap
```

Sau khi cài project ta có thêm



Sử dụng bootstrap cho file index.cshtml

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <link href="~/Content/bootstrap.css"
        rel="stylesheet" />
    <link href="~/Content/bootstrap-theme.css"
        rel="stylesheet" />
    <title>Index</title>
    <style>
        .btn a {
            color: white;
            text-decoration: none;
        }

        body {
            background-color: #F1F1F1;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <div class="panel panel-default">
                    <div class="panel-heading">
                        <h3>Bootstrap</h3>
                    </div>
                    <div class="panel-body">
                        <div class="row">
                            <div class="col-md-6">
                                <div class="panel panel-primary">
                                    <div class="panel-heading">
                                        <h4>Panel 1</h4>
                                    </div>
                                    <div class="panel-body">
                                        <p>Content of Panel 1</p>
                                    </div>
                                </div>
                                <div class="panel panel-success">
                                    <div class="panel-heading">
                                        <h4>Panel 2</h4>
                                    </div>
                                    <div class="panel-body">
                                        <p>Content of Panel 2</p>
                                    </div>
                                </div>
                            </div>
                            <div class="col-md-6">
                                <div class="panel panel-warning">
                                    <div class="panel-heading">
                                        <h4>Panel 3</h4>
                                    </div>
                                    <div class="panel-body">
                                        <p>Content of Panel 3</p>
                                    </div>
                                </div>
                                <div class="panel panel-danger">
                                    <div class="panel-heading">
                                        <h4>Panel 4</h4>
                                    </div>
                                    <div class="panel-body">
                                        <p>Content of Panel 4</p>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

```
</style>
</head>
<body>
  <div class="text-center">
    <h2>Giới thiệu sản phẩm tại trang chủ</h2>
    <p>Bạn đã sẵn sàng làm việc với MVC</p>
    <div class="btn btn-success">
      @Html.ActionLink("FeedBack", "FeedBack")
    </div>
  </div>

</body>
</html>
```

Sử dụng bootstrap cho form feedback.cshtml

```
@model OnlineShop.Models.FeedBack
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <link href="~/Content/bootstrap.css"
        rel="stylesheet" />
  <link href="~/Content/bootstrap-theme.css"
        rel="stylesheet" />
  <title>FeedBack</title>
  <link rel="stylesheet" type="text/css" href="~/Content/Styles.css" />
</head>

<body>
  <div class="panel panel-success">
    <div class="panel-heading text-center">
      <h4>FeedBack</h4>
    </div>
    <div class="panel-body">
      @using (Html.BeginForm())
      {
        @Html.ValidationSummary()
        <div class="form-group">
          <label>Name:</label>
          @Html.TextBoxFor(x => x.Name, new { @class = "form-
control" })
        </div>
        <div class="form-group">
          <label> Phone:</label>
```

```

        @Html.TextBoxFor(x => x.Phone, new { @class = "form-control" })
    }
    </div>
    <div class="form-group">
        <label>Email:</label>
        @Html.TextBoxFor(x => x.Email, new { @class = "form-control" })
    }
    </div>
    <div class="form-group">
        <label>Address:</label>
        @Html.TextBoxFor(x => x.Address, new { @class = "form-control" })
    }
    </div>
    <div class="form-group">
        <label>Content:</label>
        @Html.TextAreaFor(x => x.Content, new { @class = "form-control" })
    }
    </div>
    <div class="form-group">
        <label>Status?</label>
        @Html.DropDownListFor(x => x.Status, new[] {
            new SelectListItem() {Text = "Đồng ý", Value = bool.TrueString},
            new SelectListItem() {Text = "Không đồng ý", Value = bool.FalseString}
        }, "Choose an option", new { @class = "form-control" })
    }
    </div>
    <div class="text-center">
        <input class="btn btn-success" type="submit" value="Submit feedback" />
    </div>
}
</div>
</div>
</body>
</html>

```

Thêm bootstrap cho file thanks.cshtml

```

@model OnlineShop.Models.FeedBack
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <link href="~/Content/bootstrap.css"
        rel="stylesheet" />
    <link href="~/Content/bootstrap-theme.css"
        rel="stylesheet" />
    <title>Thanks</title>

```

```
<style>
    body {
        background-color: #F1F1F1;
    }
</style>
</head>
<body>
    <div class="text-center">
        <h1>Thank you, @Model.Name!</h1>
        <div class="lead">
            @if (Model.Status == true)
            {
                @:Cảm ơn đã phản hồi thông tin về cho chúng tôi, chúng tôi sẽ l
iên hệ lại sau
            }
            else
            {
                @:Rất tiếc vì bạn không đồng ý nhận thông tin liên hệ từ chúng
tôi, tuy nhiên hy vọng có thể cùng bạn tham gia làm sau
            }
        </div>
    </div>
</body>
</html>
```

Công việc 7: sử dụng webMail Helper để gửi email

```
@{
    try
    {
        WebMail.SmtpServer = "smtp.gmail.com";
        WebMail.SmtpPort = 587;
        WebMail.EnableSsl = true;
        WebMail.UserName = "nmphuc1hu@gmail.com";
        WebMail.Password = "#####";
        WebMail.From = "nmphuc1hu@gmail.com";
        WebMail.Send("phuc@lhu.edu.vn", "feedback",
            Model.Name + " is " + ((Model.Status
            ?? false) ? "" : "not")
            + "attending");
    }
    catch (Exception ex)
    {
        @:<b>Sorry - we couldn't send the email to confirm your RSVP
. </b>
        @:ex.message;
    }
}
```