TRƯỜNG ĐẠI HỌC NHA TRANG

PHÁT TRIỂN ỨNG DỤNG WEB

Biên soạn: ThS. Bùi Chí Thành

Chủ đề 1. TỔNG QUAN VỀ ASP.NET MVC

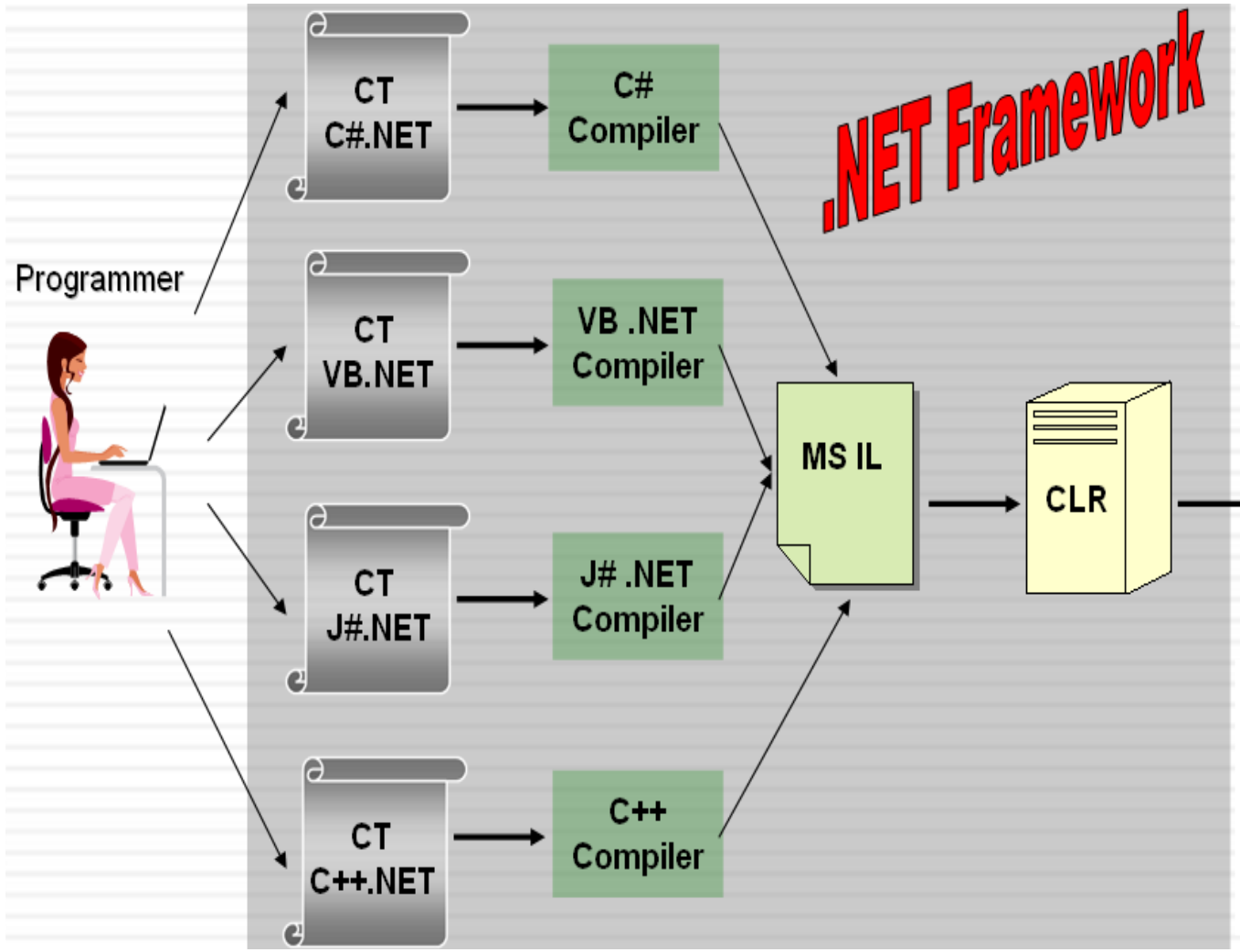
1. Giới thiệu về ASP.NET MVC
2. Môi trường và công cụ cài đặt
3. Triển khai ứng dụng ASP.NET MVC

1. Giới thiệu về ASP.NET MVC

1.1 Giới thiệu .NET

- .NET là một nền tảng mới cho phép phát triển các phần mềm nhanh hơn và đơn giản hơn được cung cấp bởi Microsoft năm 2002.
- .NET framework bao gồm tập các thư viện lập trình lớn, và những thư viện này hỗ trợ việc xây dựng các chương trình phần mềm như lập trình giao diện; truy cập, kết nối cơ sở dữ liệu; ứng dụng web; các giải thuật, cấu trúc dữ liệu; giao tiếp mạng... CLR cùng với bộ thư viện này là 2 thành phần chính của .NET framework.
- C#, Visual Basic .NET, C++, … là các ngôn ngữ có thể dùng để viết các ứng dụng .NET. Các ngôn ngữ này tuy khác nhau về cú pháp nhưng có cùng một kiến trúc.

Đặc điểm của ứng dụng .NET:



- ✓ Chạy trên nền .NET Framework
- ✓ Mã nguồn được biên dịch qua ngôn ngữ trung gian (MicroSoft Intermediate Language – MSIL)

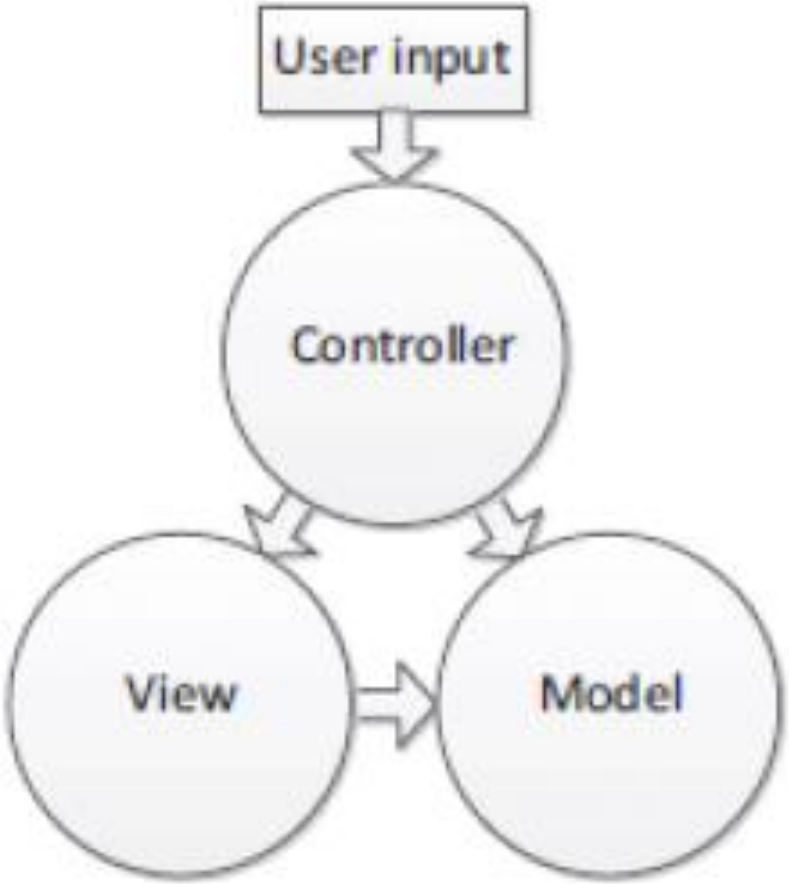
- ✓ MSIL được thông dịch qua mã máy lúc thực thi nhờ vào bộ thực thi ngôn ngữ chung (Common Language Runtime - CLR)

1.2. Giới thiệu ASP.NET MVC

- ASP.NET là một nền tảng ứng dụng web (web application framework) được phát triển và cung cấp bởi Microsoft, cho phép những người lập trình tạo ra những trang web động, những ứng dụng web và những dịch vụ web.
- Dựa trên ASP.NET, ASP.NET MVC cho phép các nhà phát triển phần mềm xây dựng một ứng dụng web dựa trên mẫu thiết kế MVC.
- MVC là một mẫu thiết kế (design pattern) chuẩn, được sử dụng nhằm chia ứng dụng thành ba thành phần chính: model, view và controller.
- Phiên bản ASP.NET MVC: ASP.NET MVC 5 (10/2013), ASP.NET MVC 5.2.7 (11/2017).

1.3. Kiến trúc ASP.NET MVC

- Controller:
  - Nhận yêu cầu từ user
  - Xử lý và xây dựng model phù hợp
  - Chuyển model cho view
- View:
  - Tiếp nhận Model từ Controller để sinh giao diện phù hợp
- Model:
  - Chứa dữ liệu chia sẻ chung giữa Controller và View



Đặc điểm MVC 5:

- MVC: Tách bạch các phần việc trong xử lý yêu cầu.
- Tự động nhận diện thiết bị: Tự lựa chọn view phù hợp.
- Razor: sinh giao diện.
- Kiểu dữ liệu động: ViewBag/DataView.
- Cải thiện Ajax: Jquery + Helper Ajax.
- Kiểm lỗi: lập trình 1 lần áp dụng cho cả 2 client và server.
- Web API: thư viện web dùng cho nhiều loại thiết bị.
- Action Filter: kiểm soát các Action.
- Dễ test: dễ dàng test các Action của các Controller.
- NuGet: quản lý các gói mở rộng.

2. MÔI TRƯỜNG VÀ CÔNG CỤ

1. Visual Studio.NET
2. SQL Server
3. Internet Information Server (IIS)
4. Web Browser: Google Chrome, Firefox, Microsoft Edge, …

MS Visual Studio

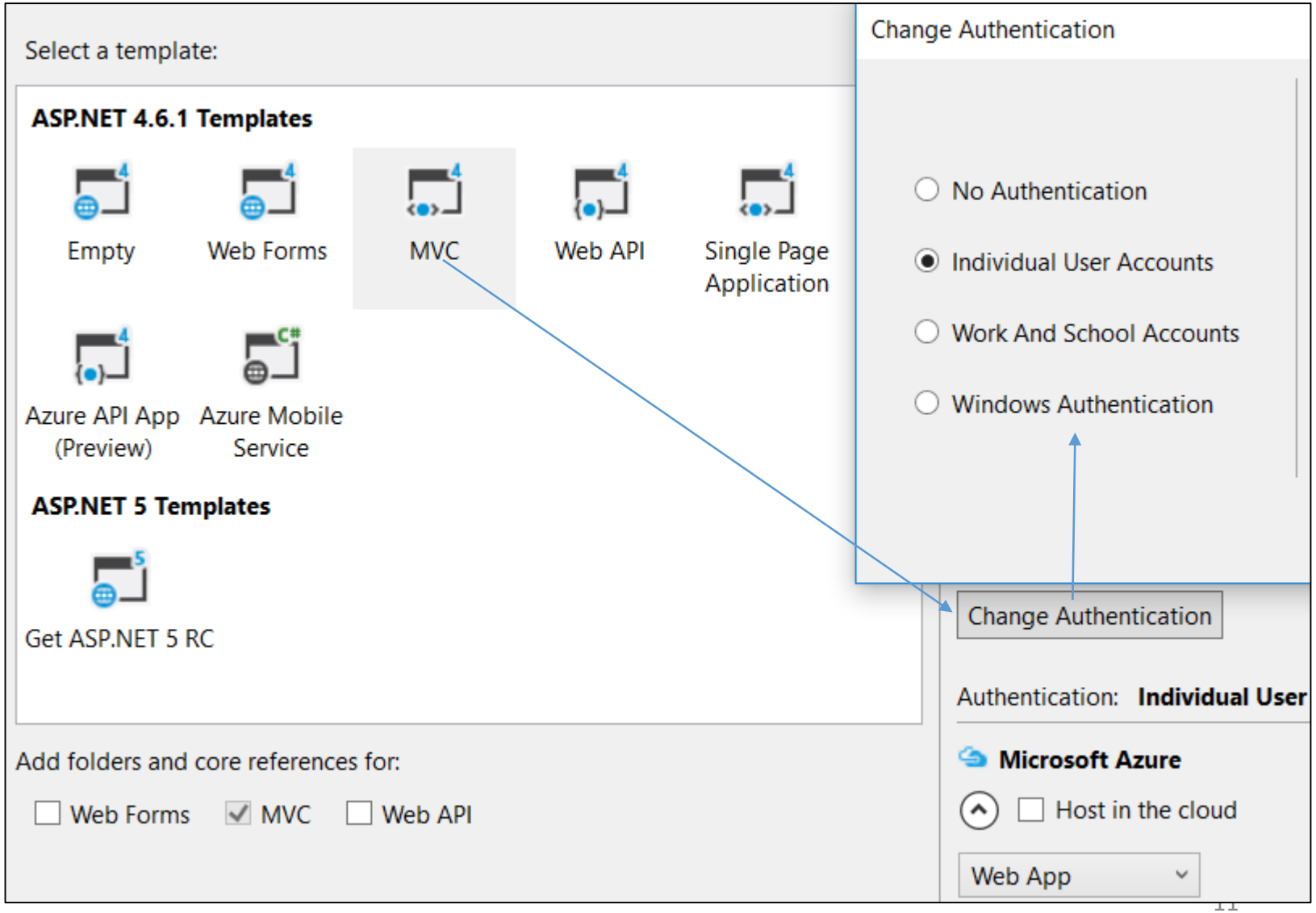
- Các phiên bản Visual Studio.NET phổ biến:
  - Visual Studio 2012
  - Visual Studio 2013
  - Visual Studio 2015
  - Visual Studio 2017
  - Visual Studio 2019
  - Visual Studio 2022
- Các ấn bản chính: Community/Express, Professional, Enterprise

Hệ quản trị CSDL Server

- Các phiên bản SQL Server phổ biến:
  - SQL Server 2012
  - SQL Server 2014
  - SQL Server 2016
  - SQL Server 2019
- Các ấn bản chính: Community/Express, Standard, Enterprise

3. TRIỂN KHAI ỨNG DỤNG ASP.NET MVC

Khởi động Visual Studio → File → Project → Visual C# → Web

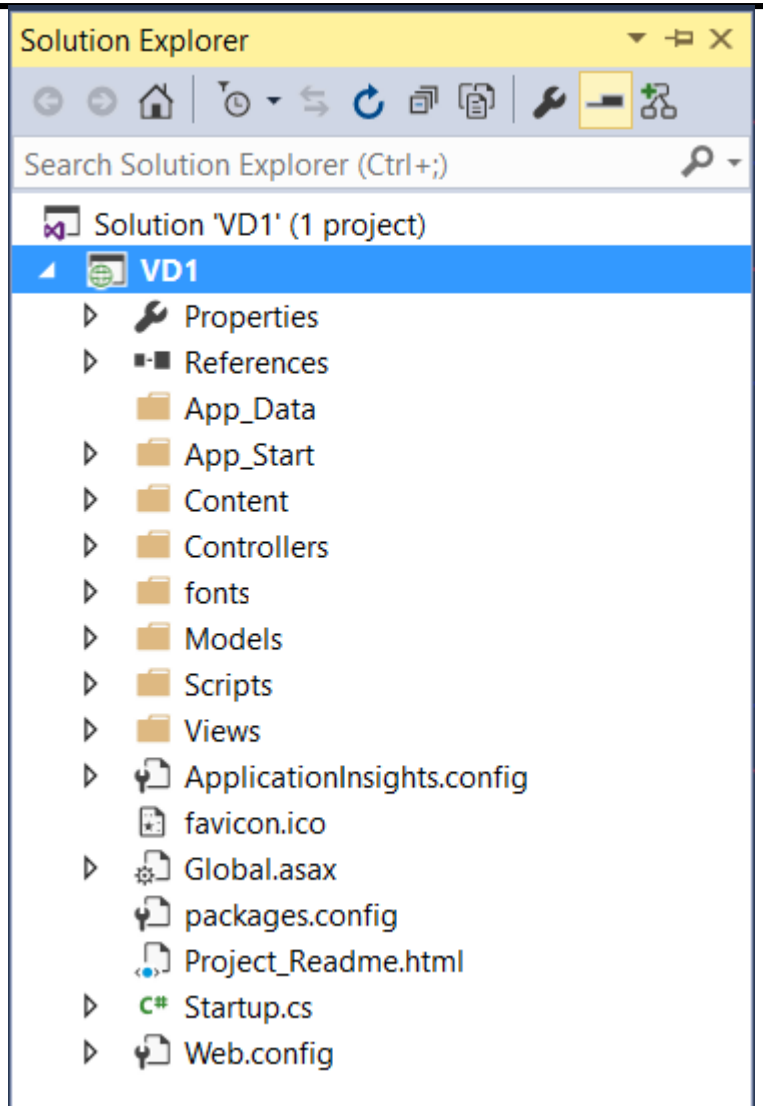


Sự thẩm định quyền (Authentication):

- Individual User Accounts: Dự án được tạo ra là loại dự án Internet bao gồm cả phần security
- No Authentication: Dự án được tạo ra là dự án Internet không bao gồm phần security.
- Windows Authentication: Dự án được tạo ra là loại dự án Intranet bao gồm cả phần security nhưng tài khoản được quản lý trên mạng nội bộ.
- Work anh School Accounts: Dự án được tạo ra là loại dự án sử dụng tài khoản từ cloud computing.
- Ngoài ra, có thể chọn Web API để bổ sung khả năng hỗ trợ xây dựng thư viện web.

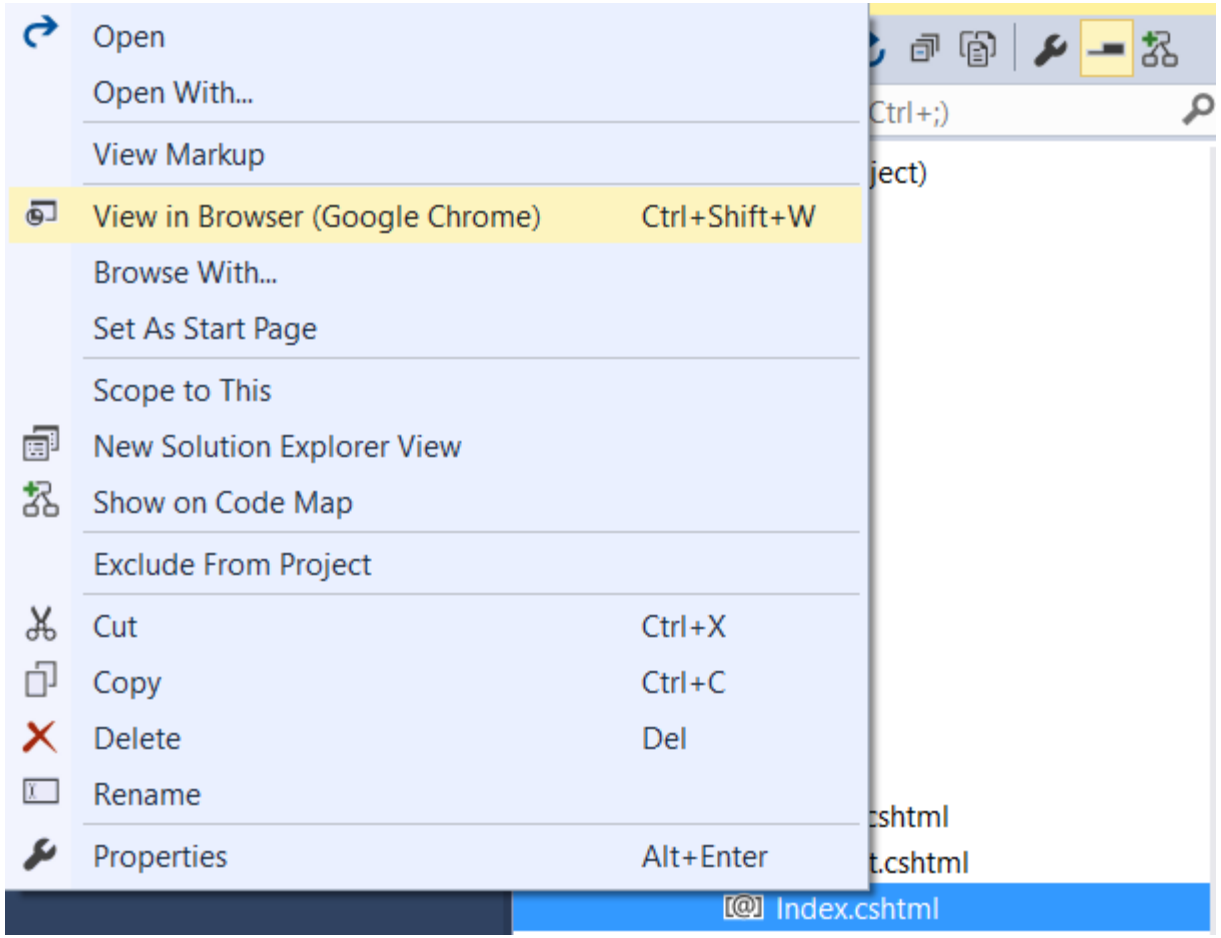
Cấu trúc ứng dụng ASP.NET MVC:

- 3 thư mục bắt buộc: Controllers, Models và Views chứa 3 thành phần chính của hệ thống.
- Ngoài ra, còn có các thư mục và tập tin như:
  - App_Data chứa các file dữ liệu;
  - App_Start chứa các file cấu hình hệ thống;
  - Content chứa các file định dạng hiện thị;
  - Scripts chứa các file xử lý bằng ngôn ngữ JavaScripts;



- File Global.asax dùng để liên kết các thiết lập, sự kiện cho toàn bộ ứng dụng;
- File packages.config chứa các thư viện sử dụng trong project;
- File web.config chứa thông tin cấu hình hệ thống.

- Chạy ứng dụng: F5 (biên dịch toàn bộ ứng dụng và chạy trang mặc định được thiết lập)



+ Hoặc chạy từng trang được chọn: Chọn trang → R_Click → View in Browser.

Cách thao tác Controllers, Models và Views:

1. Controllers: R_Click → Add → Controller …

→ Chọn: Empty; read/write action; Entity Framework; Web API 2 …., đặt tên (lưu ý, phải có chuỗi controller ở cuối).

```
public ActionResult IndexHomeUD2()
{
    return View();
}
```

2. Tạo Views, có 2 cách:

- Chọn View() → R_Click → Add View (Đặt tên View phải trùng tên với phương thức ActionResult)

- Views: R_Click → Add → View…(Đặt tên View phải trùng tên với phương thức ActionResult)
  - Thiết kế hiện thị nội dung trong trang cshtml
  - Hoặc xây dựng code trên đối tượng ViewBag trong Controller:

```
public ActionResult IndexHomeUD2()
{
    ViewBag.AString = "Hello, World";
    return View();
}
```

Gọi lại trong View:

```
<body>
    <div>
        @ViewBag.AString
    </div>
</body>
</html>
```

3. Tạo và sử dụng Models: R_Click → New Item → Code (Class) → Đặt tên: MessageModels

Code trong class MessageModels:

```
public class MessageModels
{
        2 references
    public string Welcome { set; get; }
}
```

Code trong IndexHomeUD2 Controllers:

```
public ActionResult IndexHomeUD2()
{
    var message = new Models.MessageModels();
    message.Welcome = "Chào mừng đến với ASP.NET MVC";
    return View(message);
}
```

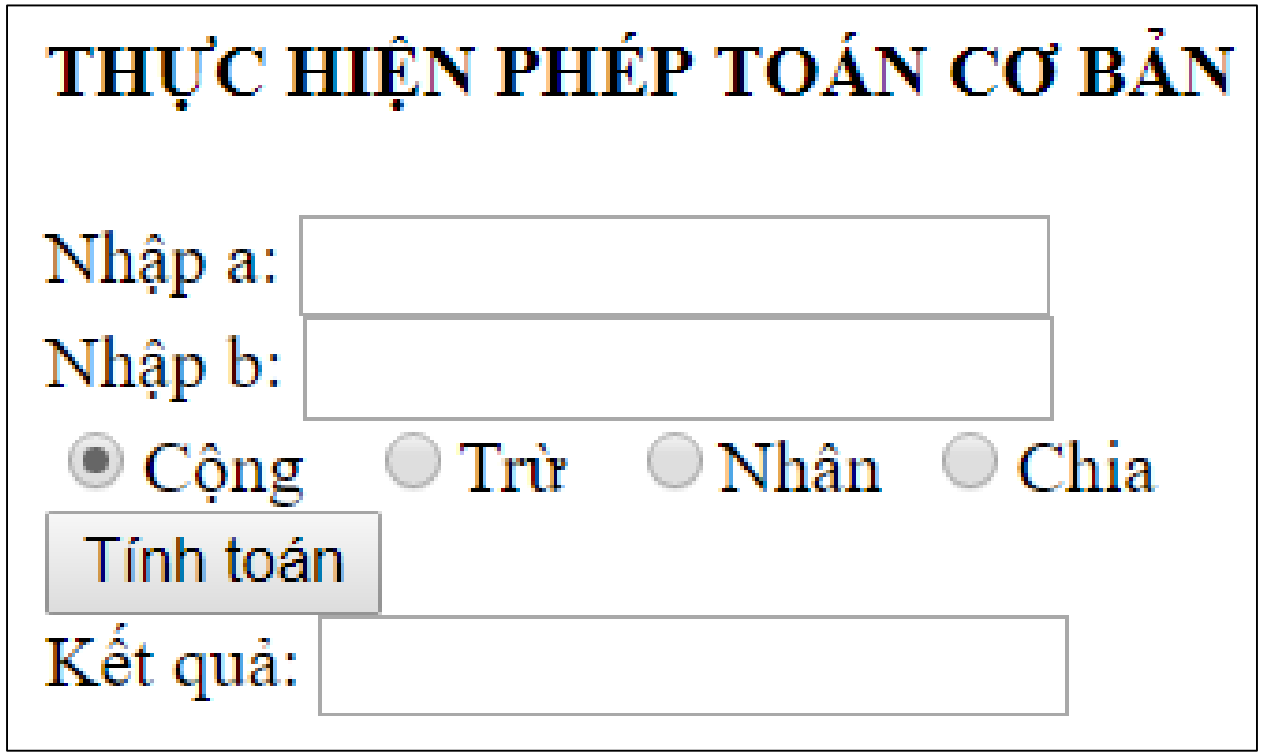
Chỉnh sửa code trong View:

```
@model UD2.Models.MessageModels
<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>IndexHomeUD2</title>
</head>
<body>
    <div>
        @Model.Welcome
    </div>
</body>
</html>
```

Bài tập 1:

- Tạo một Project có tên BaiTap1, chạy các view có trong project (các trang *.cshtml)
- Thêm vào ứng dụng có chức năng nhập vào 2 số thực, tính và in kết quả có giao diện như sau:



Tạo một Controller có tên CalculatorController, source gợi ý:

```
[HttpPost]
public ActionResult Index(double a,double b,string pt="+")
{
  switch (pt)
   {
       case "+": ViewBag.KQ = a + b;break;
       case "-": ViewBag.KQ = a - b; break;
       case "*": ViewBag.KQ = a * b; break;
       case "/":
           if (b == 0) ViewBag.KQ = "Không chia được cho 0";
           else ViewBag.KQ = a / b; break;
    }
    return View();
}
```

Tạo View, code View gợi ý:

```
@{
    ViewBag.Title = "Index";
    Layout = null;
}
<h4>THỰC HIỆN PHÉP TOÁN CƠ BẢN</h4>
<form name="f1" action="/Calculator/Index" method="post">
      <div>
         Nhập a: <input type="text" name="a" /> <br />
         Nhập b: <input type="text" name="b" />
      </div>
      <div>
        <input type="radio" checked name="pt" value="+" />Cộng  
        <input type="radio" name="pt" value="-" />Trừ  
        <input type="radio" name="pt" value="*" />Nhân 
        <input type="radio" name="pt" value="/" />Chia 
    </div>
    <div><input type="submit" value="Tính toán" /></div>
    <div>Kết quả: <input type="text" name= "kq" value=@ViewBag.KQ /></div>
</form>
```

Bài tập 2: Giống chức năng bài 1 nhưng truyền từ View sang Controller bằng Model:

- Tạo Model:

```
public class Calculatorcs
    {
        public double a { get; set; }
        public double b { get; set; }
        public string pt { get; set; }
    }
```

Tạo Controller, dùng: using Cal1.Models;

```
[HttpPost]
  public ActionResult Index(Calculatorcs cal)
  {
      switch (cal.pt)
      {
          case "+": ViewBag.KQ =cal. a + cal.b; break;
          case "-": ViewBag.KQ = cal.a - cal.b; break;
          case "*": ViewBag.KQ = cal.a * cal.b; break;
          case "/":
              if (cal.b == 0) ViewBag.KQ = "Không chia được cho 0";
              else ViewBag.KQ = cal.a / cal.b; break;
      }
      return View();
  }
```

Code View Index (dùng Razor sinh mã theo model):

```
@model Cal1.Models.Calculatorcs
@{
    ViewBag.Title = "Index";
}
@using (Html.BeginForm())
{
  <div>a= @Html.EditorFor(model => model.a) </div>
  <div>b= @Html.EditorFor(model => model.b) </div>
  <div>
    Cộng @Html.RadioButtonFor(model => model.pt,"+", new { @checked = "checked" })  
    Trừ @Html.RadioButtonFor(model => model.pt, "-")  
    Nhân @Html.RadioButtonFor(model => model.pt, "*")  
    Chia @Html.RadioButtonFor(model => model.pt, "/")  
  </div>
  <div><input type="submit" value="Tính toán"/></div>
  <div>Kết quả: <input type="text" name="kq" value=@ViewBag.KQ /></div>
}
```

Tạo một CSDL có tên UD2, gồm có 2 bảng dữ liệu như sau:

- Loại mặt hàng (**Mã loại mặt hàng**, Tên loại mặt hàng, Ghi chú)
- Mặt hàng (**Mã hàng**, Tên hàng, ĐVT, Đơn giá, Mã loại mặt hàng)
- Nhập vào mỗi bảng ít nhất 3 dòng dữ liệu.

Minh hoạ 1 project ASP.NET MVC có tương tác với CSDL