

# CHỦ ĐỀ 1



# TỔNG QUAN VỀ ASP.NET MVC

GVGD: PHẠM THỊ KIM NGOAN

Email: [ngoanptk@ntu.edu.vn](mailto:ngoanptk@ntu.edu.vn)

# Nội dung



Giới thiệu ASP.NET



Mẫu kiến trúc MVC



Cách hoạt động của hệ thống web



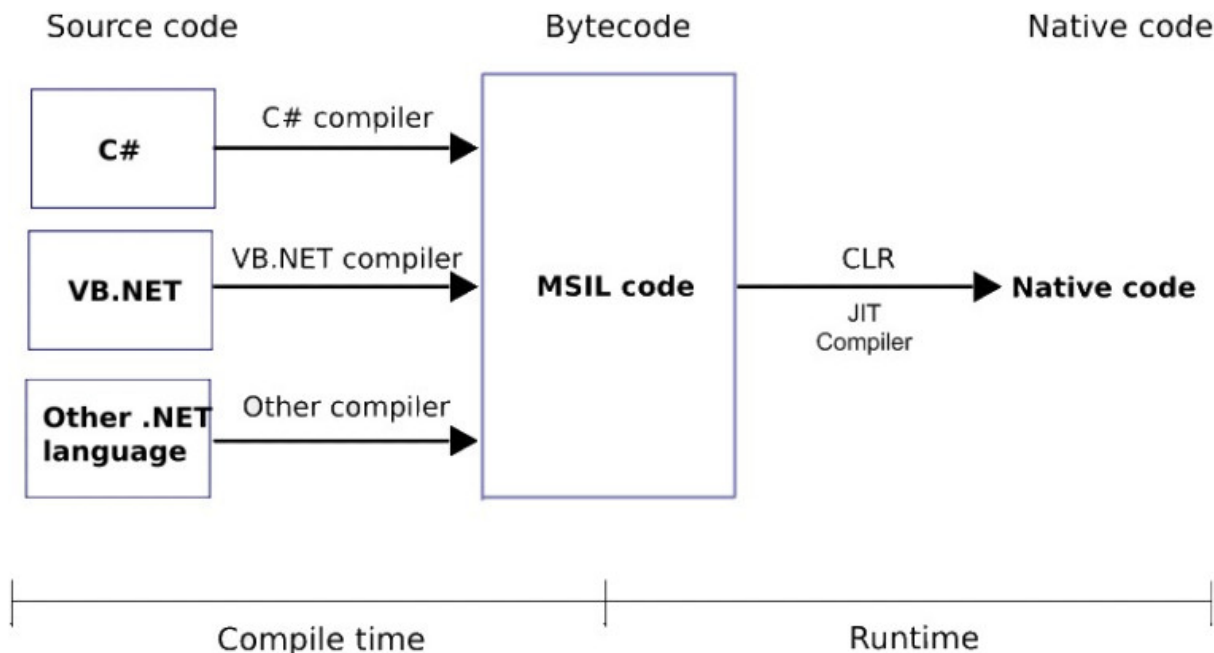
Cách tạo project ASP.NET MVC



Cấu trúc project ASP.NET MVC

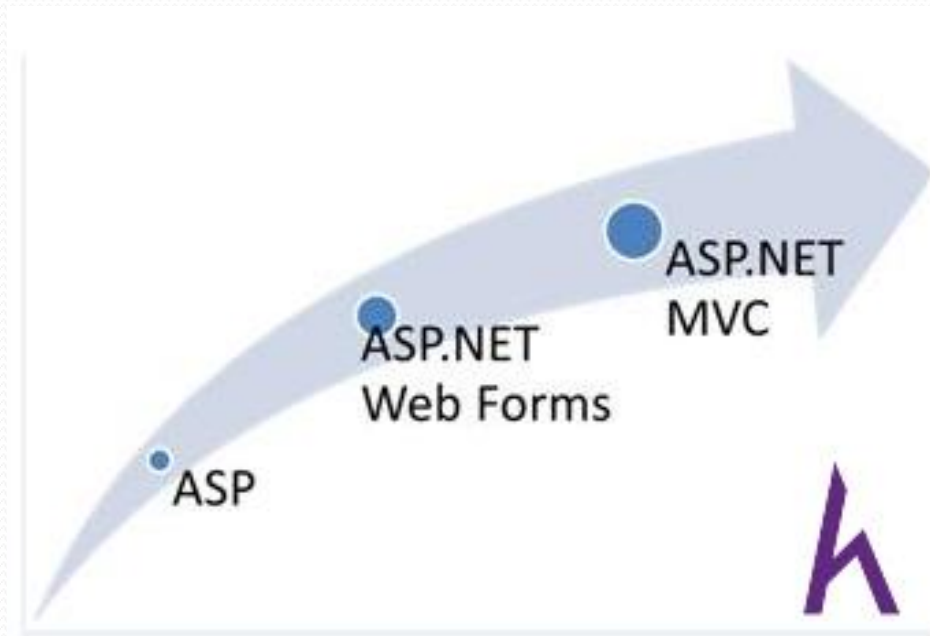
# 1. Giới thiệu ASP.NET

- ASP.NET (ASP - Active Server Pages, NET - Network Enabled Technologies) là một nền tảng dành cho phát ứng dụng web động.
- Các ứng dụng ASP.Net được viết bằng nhiều loại ngôn ngữ .Net khác nhau, phổ biến như C #, VB.Net và J #.



# 1. Giới thiệu ASP.NET

- ASP.NET gồm 2 model: Web Forms và MVC.



# 1. Giới thiệu ASP.NET

---

## ➤ Tại sao nên dùng MVC?

- Ứng dụng gồm ba thành phần chính tách biệt rõ ràng.
- Giao diện sử dụng công nghệ thiết kế web HTML, CSS.
- Hỗ trợ bất đồng bộ → nhanh hơn.
- Hỗ trợ tốt cho việc test từng phần.
- Thừa hưởng các tính năng của ASP.NET.
- ...

# 1. Giới thiệu ASP.NET

## ➤ Các tính năng của nền tảng ASP.NET MVC:

- Thừa hưởng tính minh bạch và cấu trúc rõ ràng của MVC.
- Là một nền tảng khả mở rộng (extensible) & khả nhúng (pluggable)
- Hỗ trợ đầy đủ các tính năng: caching, session, profile,...
- ...
- ASP.NET MVC 3: view engine mới là Razor View Engine.
- ASP.NET MVC 4: bổ sung ASP.NET Web API.
- ASP.NET MVC 5: hỗ trợ chứng thực qua các API khác (facebook, google+..), Bootstrap được thêm vào hỗ trợ phần thiết kế giao diện.

# 1. Môi trường cho ASP.NET

➤ Để phát triển ứng dụng ASP.NET cần các công cụ:

- Bộ .NET SDK

Link: <https://dotnet.microsoft.com/en-us/download>

- Môi trường phát triển ứng dụng tích hợp: Visual Studio, Visual Studio Code, ..

Link: <https://visualstudio.microsoft.com/downloads/>

<https://code.visualstudio.com>

- Hệ quản trị CSDL: SQLServer

Link: <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

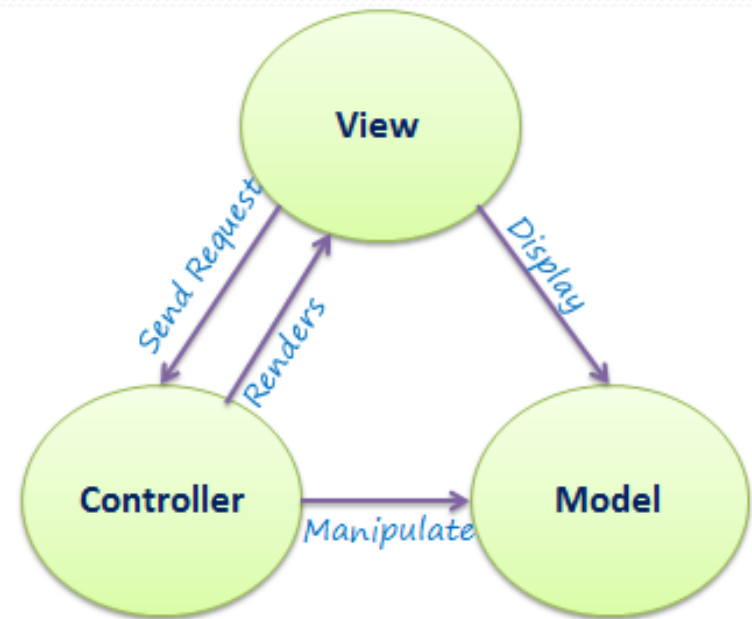
# Nội dung

- Giới thiệu ASP.NET
- Mẫu kiến trúc MVC
- Cách hoạt động của hệ thống web
- Cách tạo project ASP.NET MVC
- Cấu trúc project ASP.NET MVC



## 2. Kiến trúc MVC

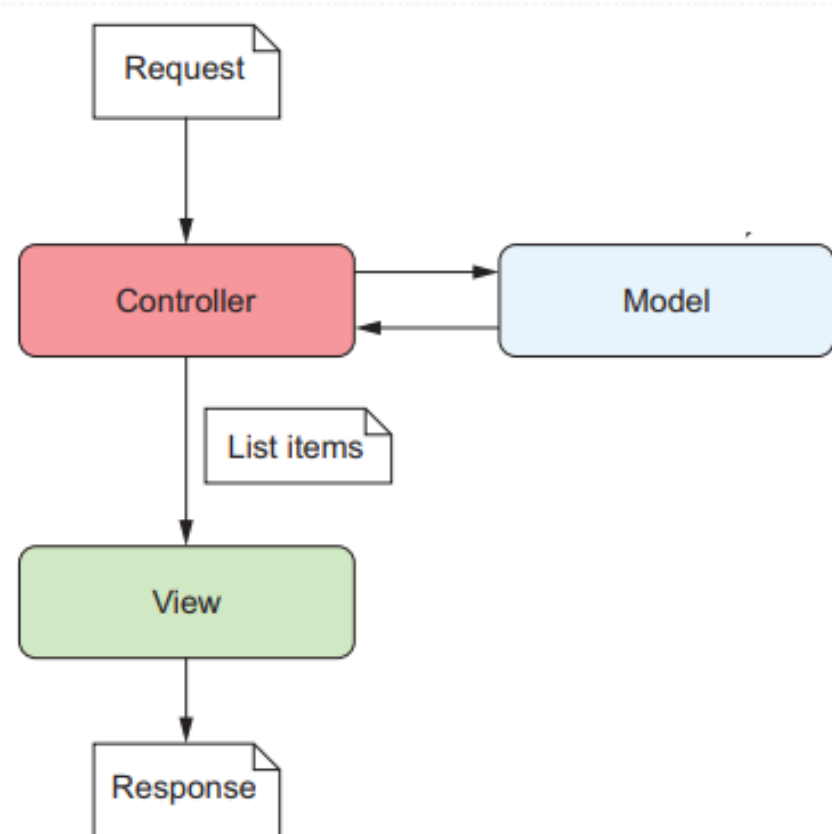
- MVC (Model – View – Controller) hướng tới phân tách việc biểu diễn dữ liệu khỏi quản lý và xử lý dữ liệu.
- MVC phân chia ứng dụng ra 3 thành phần:
  - Model – dữ liệu và trạng thái của ứng dụng.
  - View – khuôn mẫu dành cho hiển thị dữ liệu.
  - Controller – cập nhật model và lựa chọn view.



## 2. Kiến trúc MVC

### ➤ MVC:

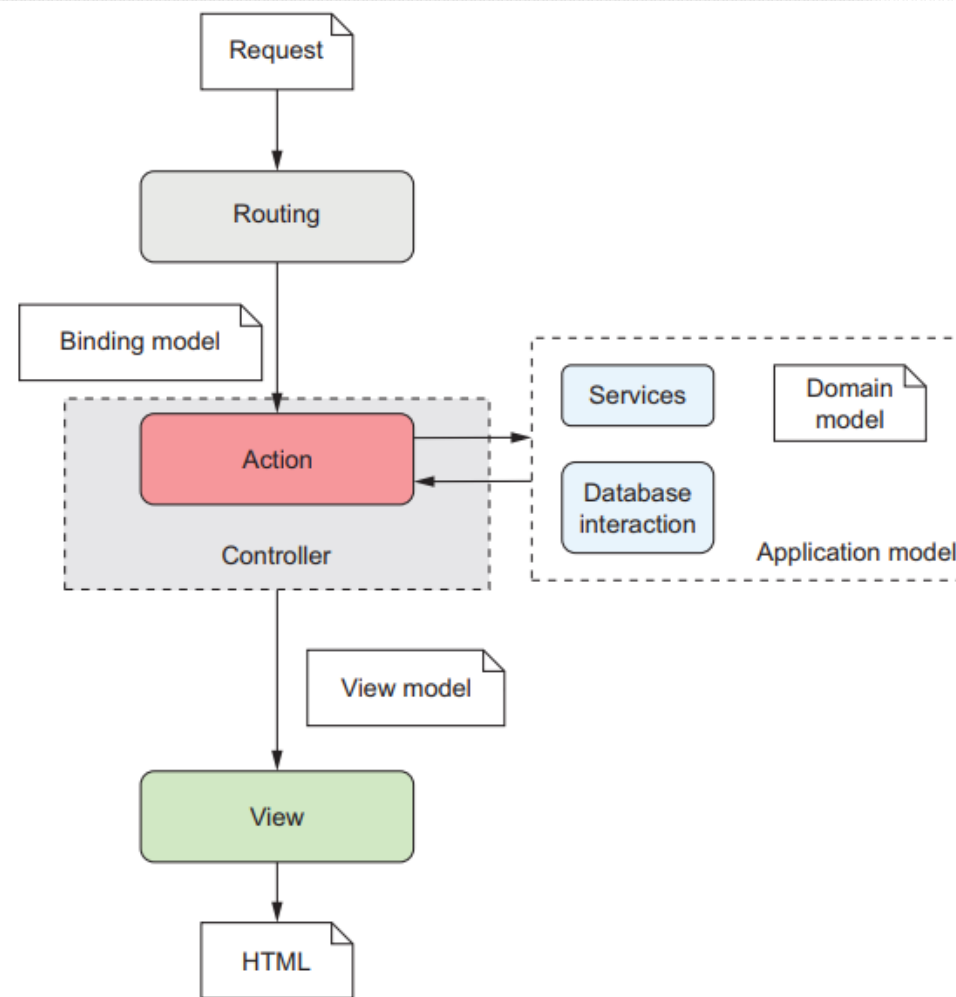
- Model: Chứa dữ liệu chia sẻ chung giữa Controller và View.
- View: Tiếp nhận Model từ Controller để sinh giao diện.
- Controller:
  - Nhận yêu cầu từ user.
  - Xử lý và xây dựng model phù hợp.
  - Chuyển model cho view.



## 2. Kiến trúc MVC

### ➤ MVC trong ASP.NET:

- Truy vấn HTTP được cơ chế Routing ánh xạ sang phương thức gọi là action.
- Controller: class chứa các action.
- View: các trang Razor chứa mã C# + HTML (Razor view)



## 2. Kiến trúc MVC

### ➤ Đặc điểm MVC 5:

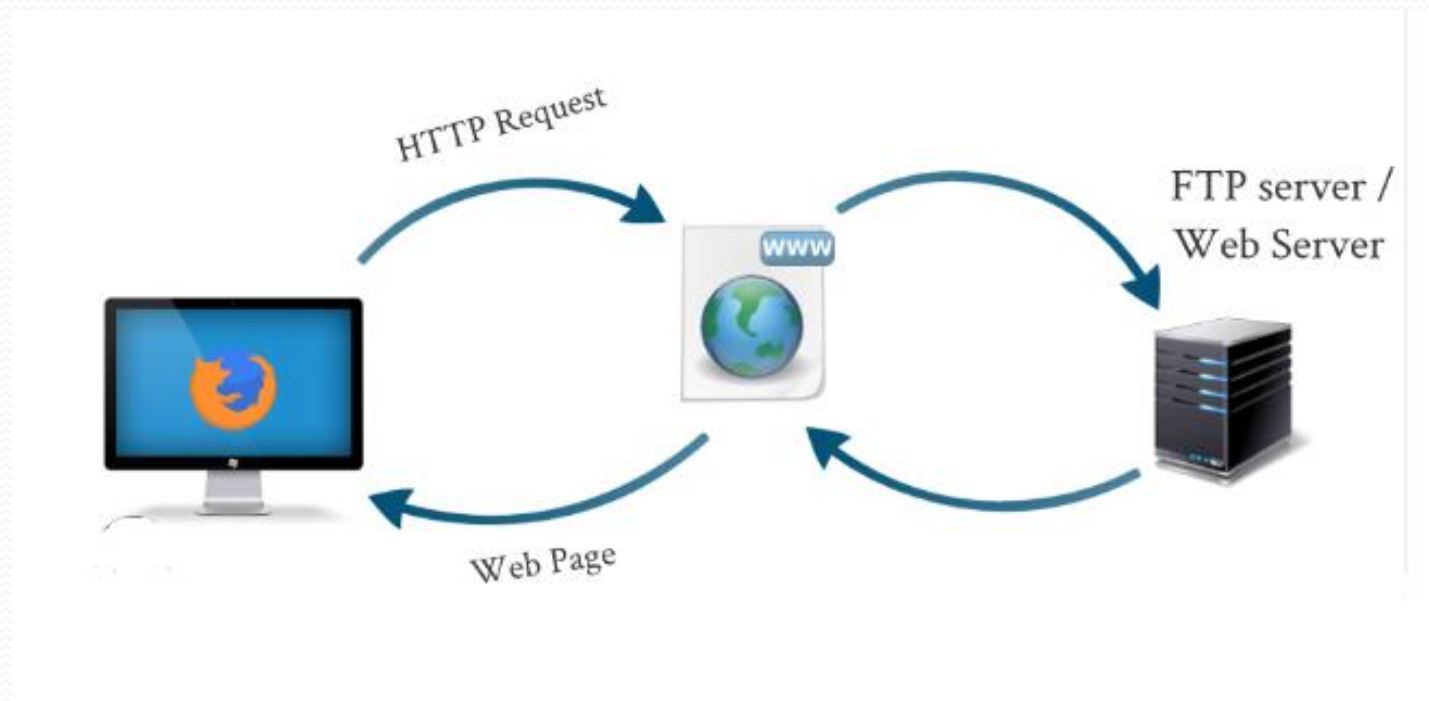
- Tách bạch các phần việc trong xử lý yêu cầu.
- Tự lựa chọn view phù hợp.
- Razor: sinh giao diện.
- Cải thiện Ajax: JQuery + Helper Ajax.
- Kiểm lỗi: lập trình 1 lần áp dụng cho cả 2 client và server.
- Web API: thư viện web dùng cho nhiều loại thiết bị.
- Action Filter: kiểm soát các Action.
- Dễ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

# Nội dung

- Giới thiệu ASP.NET
- Mẫu kiến trúc MVC
- Cách hoạt động của hệ thống web
- Cách tạo project ASP.NET MVC
- Cấu trúc project ASP.NET MVC

### 3. Cách hoạt động của của hệ thống web

- Hệ thống web thông thường gồm 2 loại chương trình:
  - Trình duyệt – chương trình chạy trên máy người dùng.
  - Web server - Chương trình chạy trên máy chủ



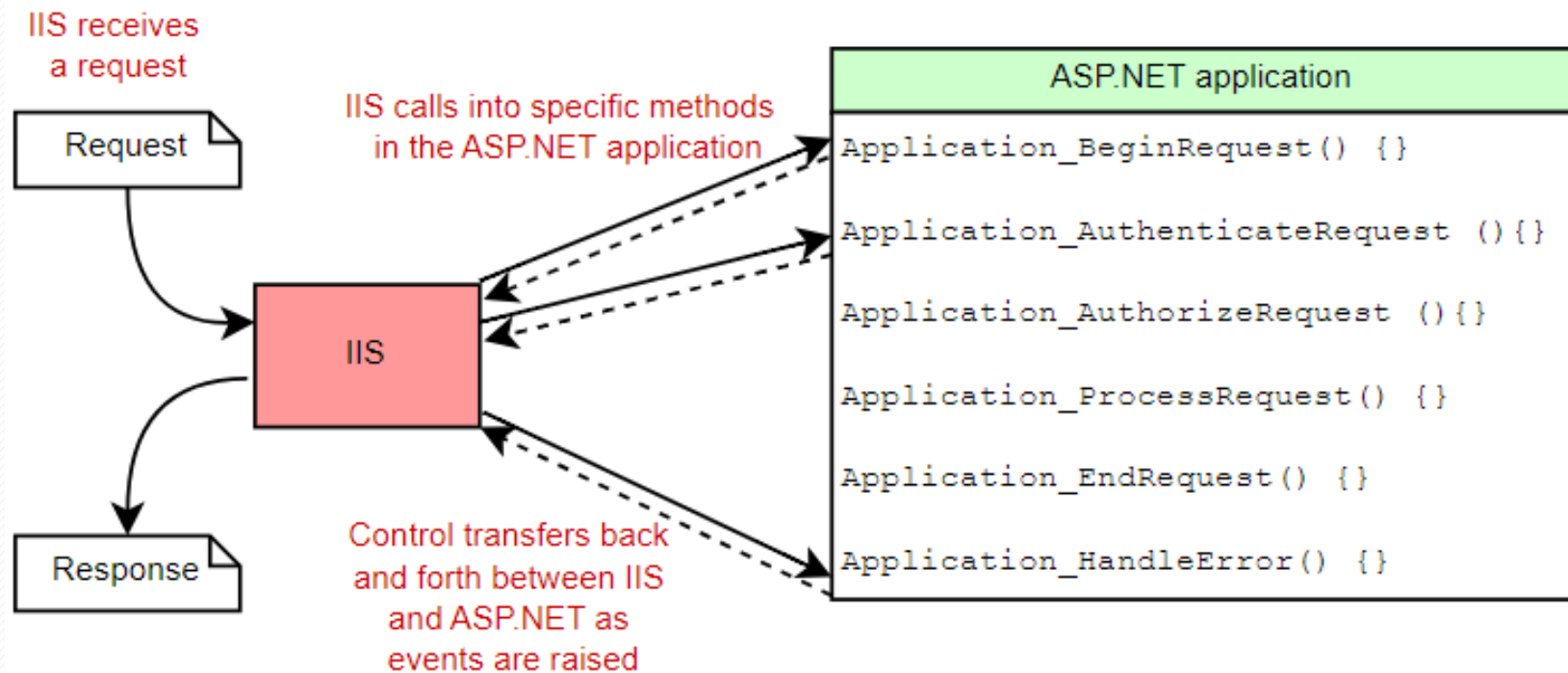
### 3. Cách hoạt động của hệ thống web

#### ➤ Hoạt động của hệ thống web:



# 3. Cách hoạt động của hệ thống web

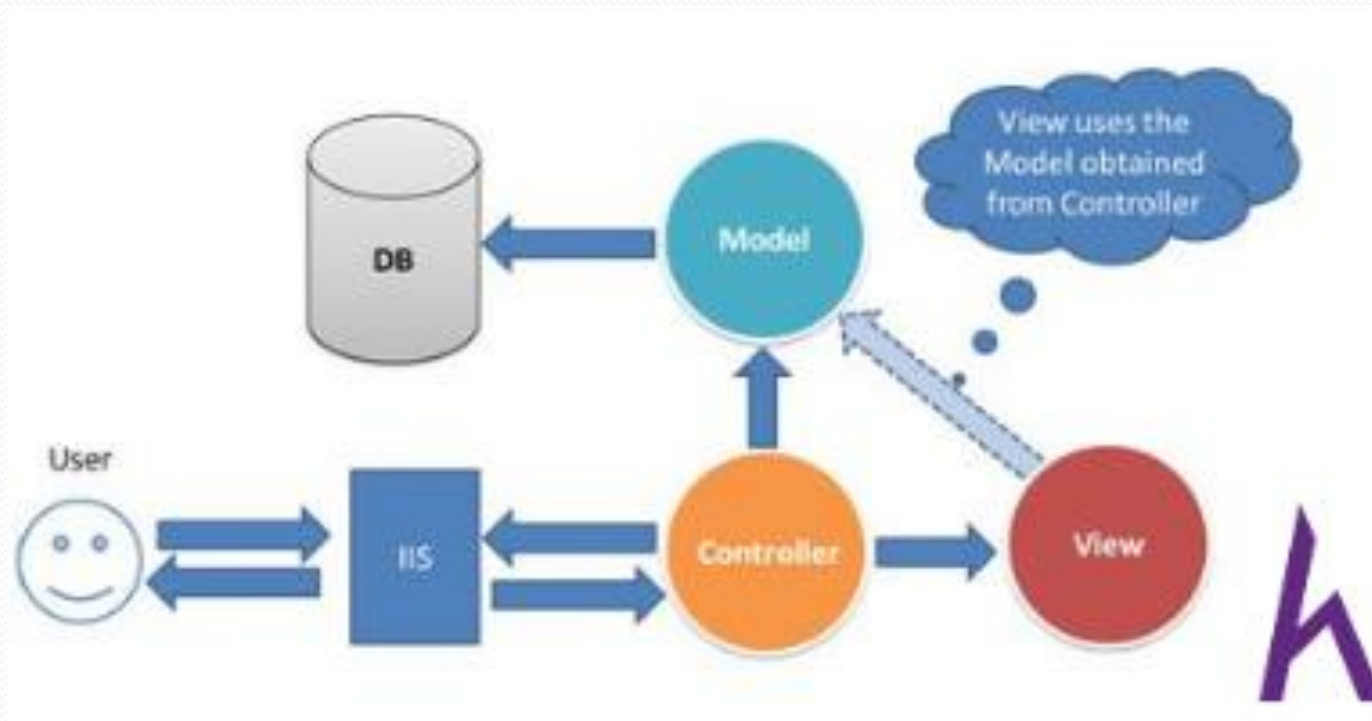
## ➤ Web ASP.NET:





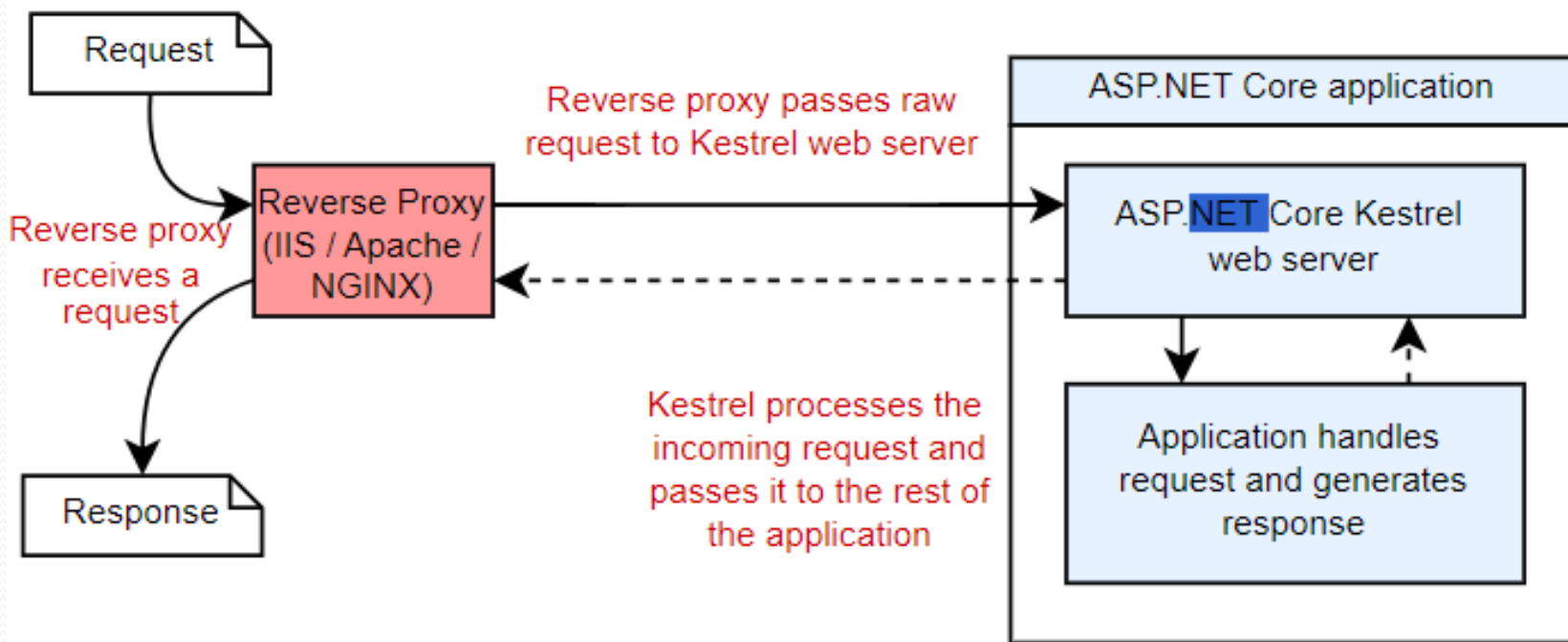
### 3. Cách hoạt động của hệ thống web

#### ➤ Web ASP.NET MVC:



# 3. Cách hoạt động của hệ thống web

## ➤ Web ASP.NET CORE:



# Nội dung

- Giới thiệu ASP.NET
- Mẫu kiến trúc MVC
- Cách hoạt động của hệ thống web
- Cách tạo project ASP.NET MVC
- Cấu trúc project ASP.NET MVC

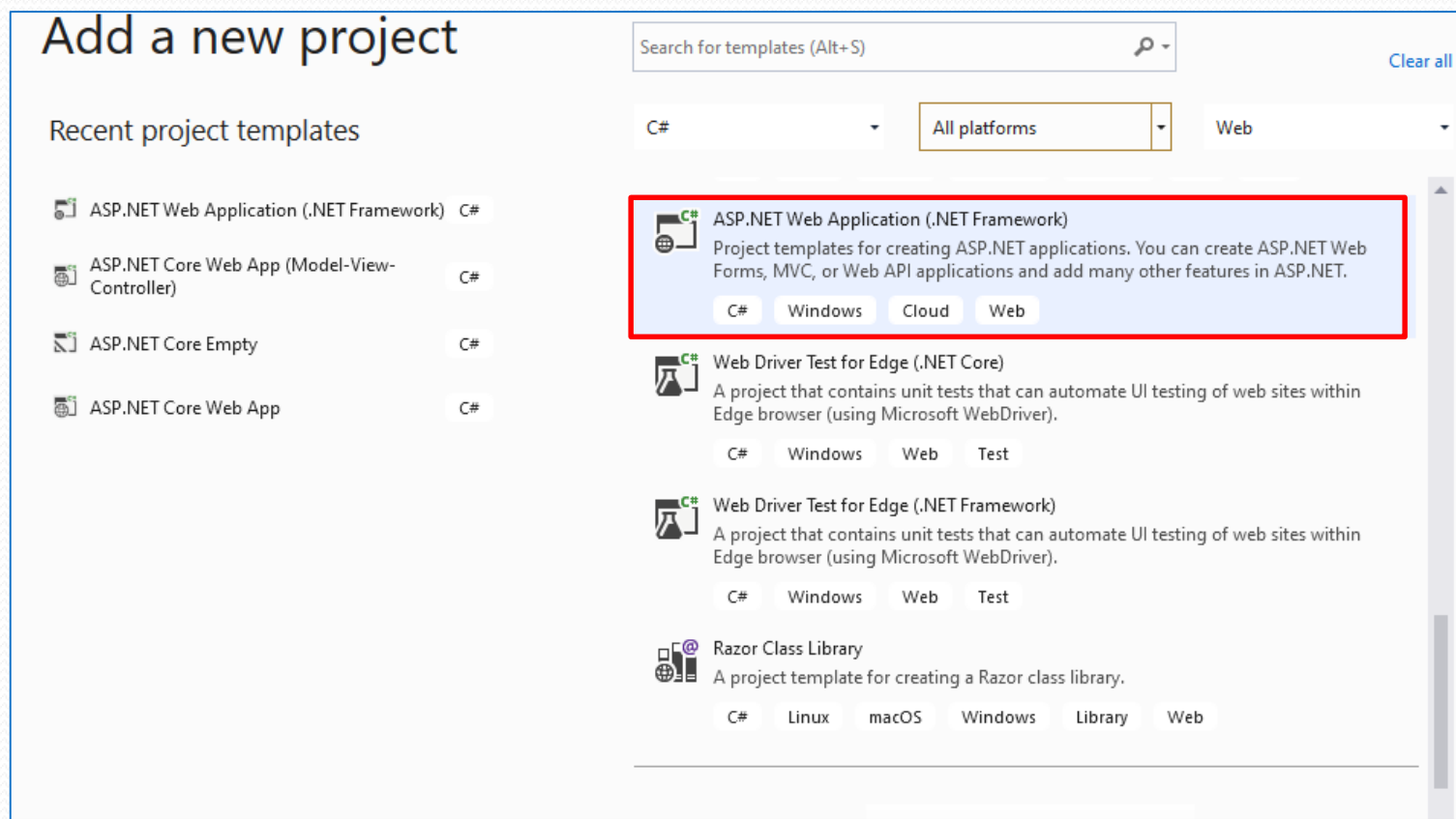
## 4. Cách tạo project ASP.NET MVC

---

- Có thể sử dụng 1 trong các công cụ sau:
  - Visual Studio (VS)
  - Visual Studio Code (VS Code)
  - CLI từ Command Prompt

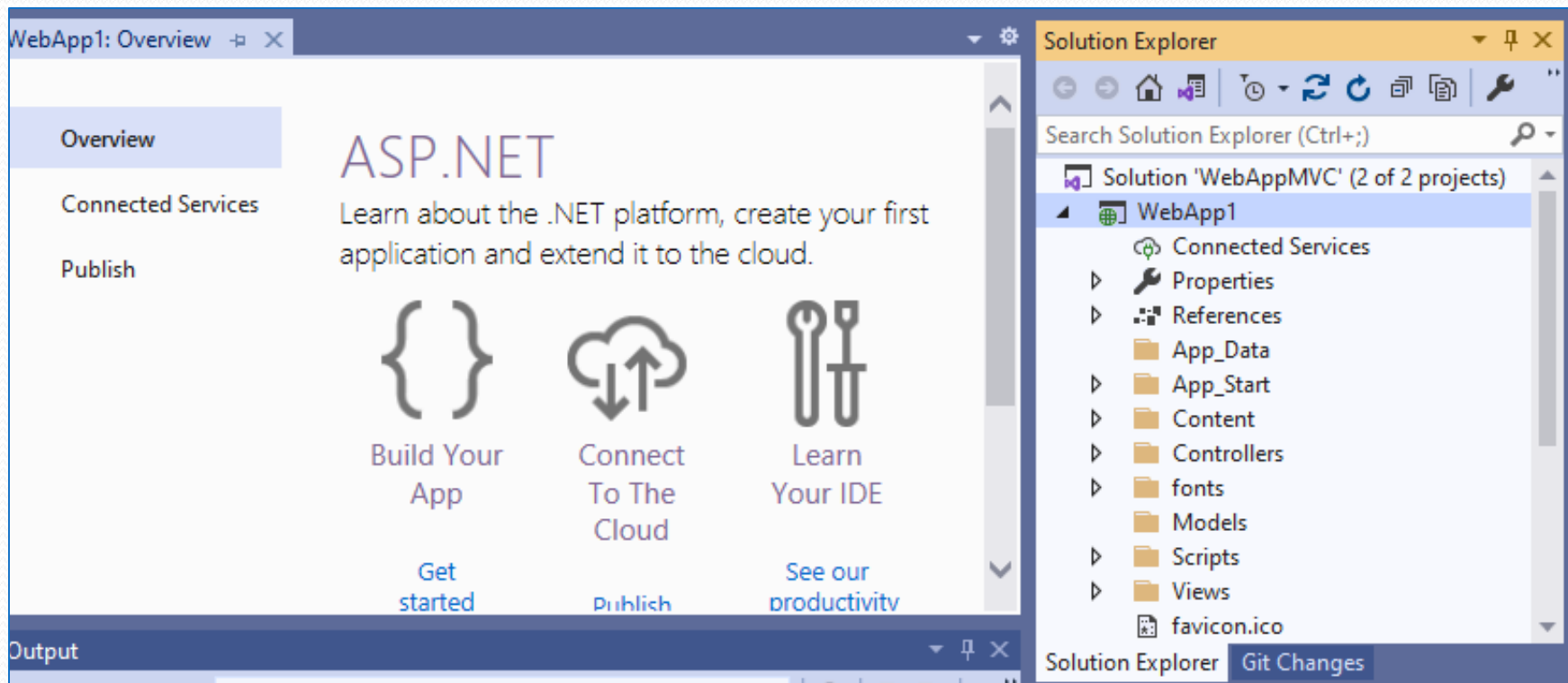
## 4. Tạo Project ASP.NET bằng VS 2019

- Khởi động VS 2019 → Create New Project → ASP.NET Web Application (.Net Framework) với tùy chọn C#, Windows/ All platform, Web.



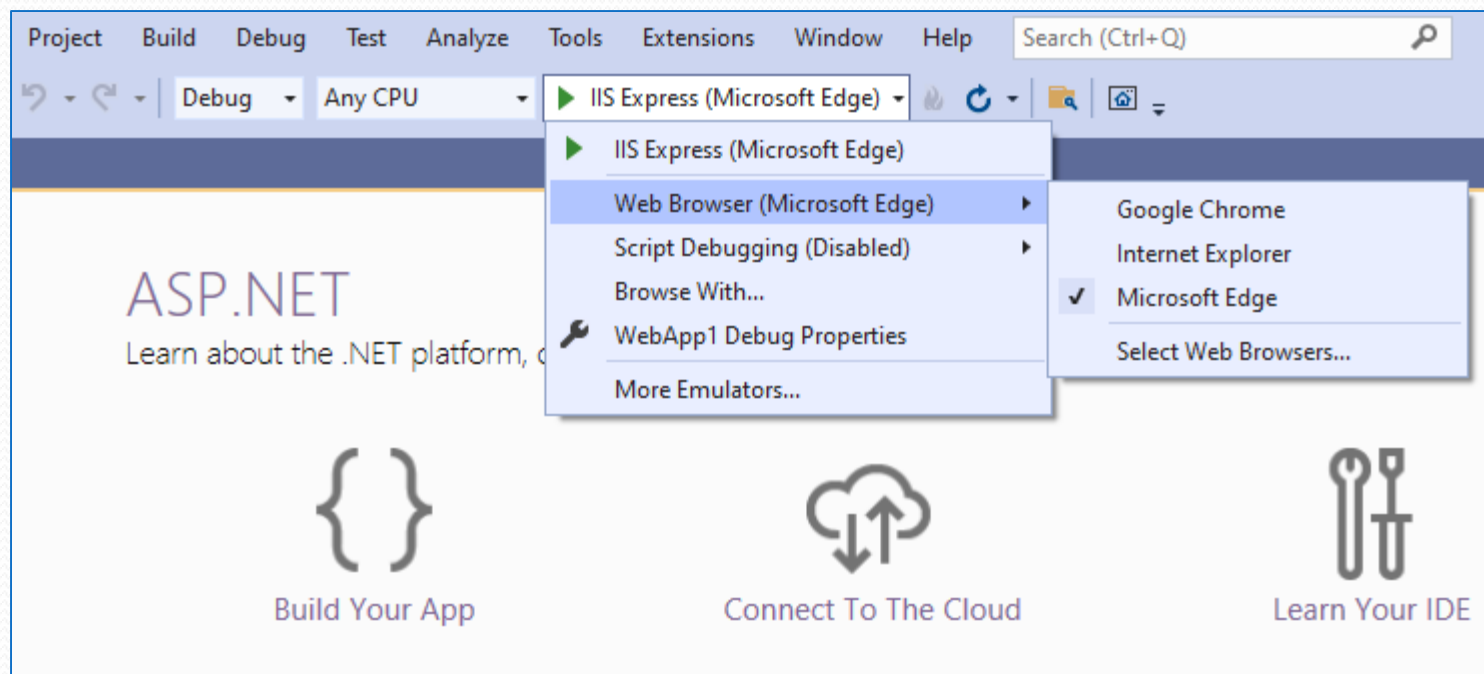
## 4. Tạo Project ASP.NET bằng VS 2019

- Đặt tên Project, vị trí lưu Project, version .Net Framework.
- Chọn MVC → Create



## 4. Tạo Project ASP.NET bằng VS 2019

- Chạy ứng dụng ASP.NET: IIS Express (IIS - Internet Information Server ) với Web Brower được chọn.



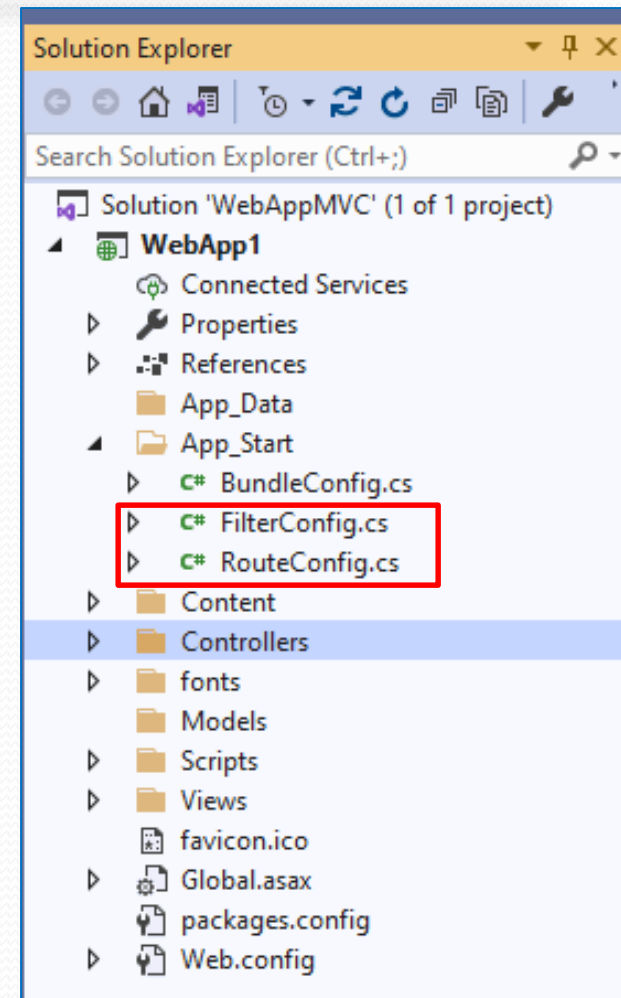
# Nội dung

- Giới thiệu ASP.NET
- Mẫu kiến trúc MVC
- Cách hoạt động của hệ thống web
- Cách tạo project ASP.NET MVC
- Cấu trúc project ASP.NET MVC



## 5. Cấu trúc project ASP.NET MVC

- **App\_Data**: thư mục chứa dữ liệu như: LocalDB, .mdf files, xml files.
- **App\_Start**: thư mục chứa các file cấu hình khởi động và biên dịch của project.
  - *FilterConfig.cs*: khai báo các filter sử dụng trước khi thực hiện 1 hành động.
  - *RouteConfig.cs*: định nghĩa các routes của web.



## 5. Cấu trúc project ASP.NET MVC

### ➤ App\_Start:

- *RouteConfig.cs*: định nghĩa các routes của web.

```
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 }
        );
    }
}
```

## 5. Cấu trúc project ASP.NET MVC

- **Content:** chứa các file :css, images và icons. Ứng dụng MVC 5 chứa các file mặc định: bootstrap.css, bootstrap.min.css và Site.css.
- **Controllers:** chứa controller xử lý yêu cầu và phản hồi cho người dùng. MVC yêu cầu tên của tất cả các tệp điều khiển kết thúc bằng "Controller".
- **Fonts:** chứa các tập tin font files của ứng dụng.
- **Models** chứa các file tương tác với CSDL
- **Scripts** : chứa các tập tin JavaScript/VBScript. MVC 5 chứa các tập tin mặc định :javascript bootstrap, jquery 1.10.
- **Views:** chứa các file HTML với đuôi là .cshtml.

## 5. Cấu trúc project ASP.NET MVC

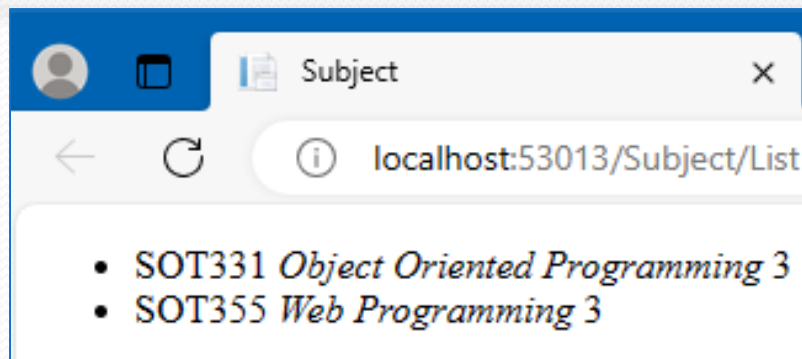
- **Global.asax:** cho phép viết code chạy để đáp ứng với các sự kiện ở mức ứng dụng.
- **Packages.config:** tập tin được NuGet quản lý để theo dõi những gói và phiên bản bạn đã cài đặt trong ứng dụng.
- **Web.config:** định nghĩa các cài đặt cho project.

## 5. Cấu trúc project ASP.NET MVC

- Tạo lớp **Controller** trong thư mục **Controllers** :  
R\_Click trên thư mục Controllers → Add → Controller ...  
→ Chọn: Empty; read/write action; Entity Framework;  
đặt tên (lưu ý, phải có chuỗi controller ở cuối).
- Tạo **View** trong thư mục **Views**:  
R\_Click trên thư mục chứa View → Add → View... (Đặt tên View phải trùng tên với phương thức ActionResult)
- Tạo và sử dụng **Models**:  
R\_Click → Add → Class → Đặt tên class

## 5. Cấu trúc project ASP.NET MVC

### ➤ Ví dụ

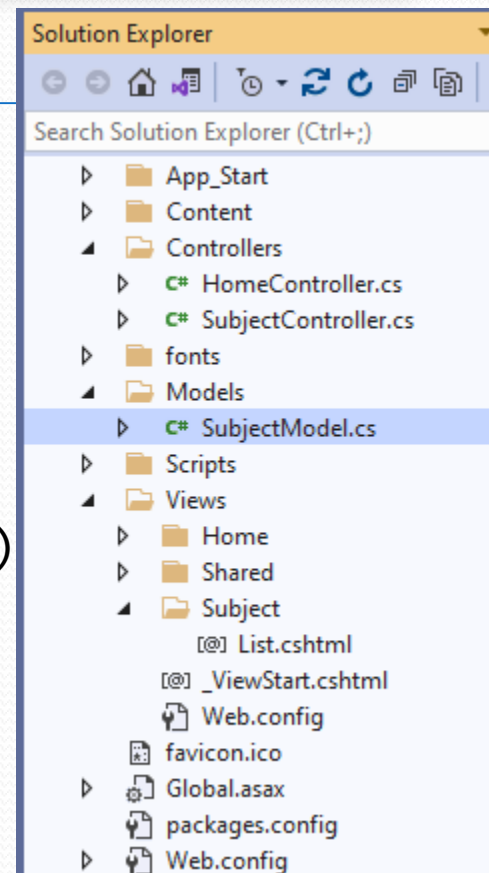


- Tạo Model: class SubjectModel trong thư mục Models.
- Tạo Controller trong thư mục Controllers:
  - Lớp SubjectController.
  - Phương thức List() trong lớp SubjectController.
- Tạo View:
  - Tạo thư mục Subject trong thư mục Views.
  - Tạo List.cshtml trong thư mục Subject.

## 5. Cấu trúc project ASP.NET MVC

### ➤ Tạo Model:

```
namespace WebApp1.Models
{
    public class SubjectModel
    {
        public string Id { get; set; }
        public string Name { get; set; }
        public byte Num { get; set; }
        public SubjectModel (string i, string s, byte n)
        {
            Id = i; Name = s; Num = n;
        }
    }
}
```



## 5. Cấu trúc project ASP.NET MVC

### ➤ Tạo Controller :

```
using WebApp1.Models;
namespace WebApp1.Controllers
{
    public class SubjectController : Controller
    {
        // GET: Subject
        public ActionResult List()
        {
            var model = new List<SubjectModel>()
            {
                new SubjectModel ("SOT331", "Object Oriented
Programming", 3),
                new SubjectModel("SOT355","Web Programming",3)
            };
            return View(model);
        }
    }
}
```



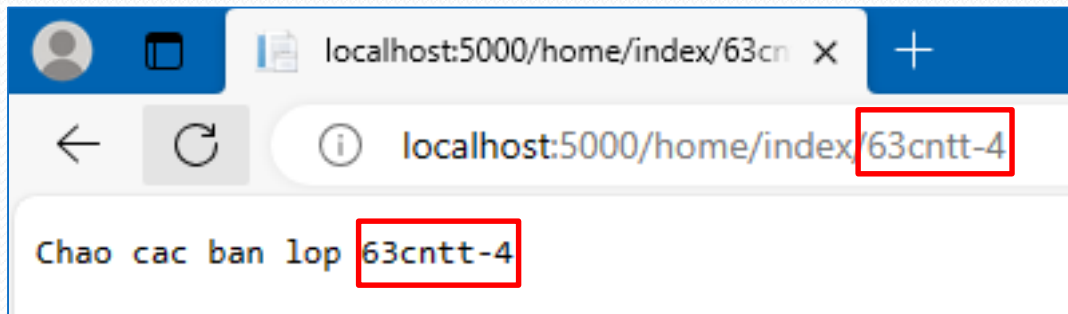
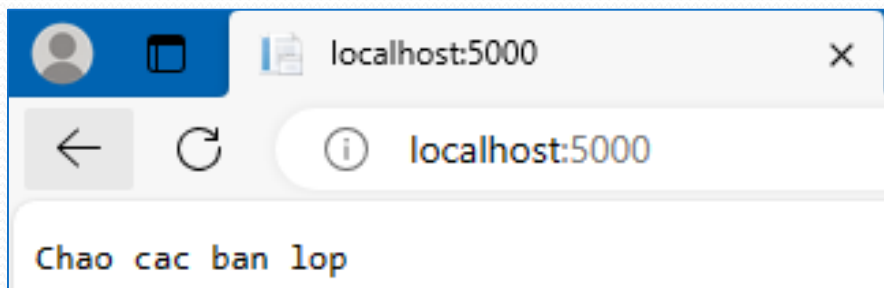
## 5. Cấu trúc project ASP.NET MVC

### ➤ Tạo View :

```
@model List<WebApp1.Models.SubjectModel>
@{
    Layout = null;
}
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Subject</title>
</head>
<body>
    <div>
        <ul>
            @foreach (var i in Model)
            {
                <li>@i.Id    <i>@i.Name    </i>    @i.Num</li>
            }
        </ul>
    </div>
</body>
</html>
```

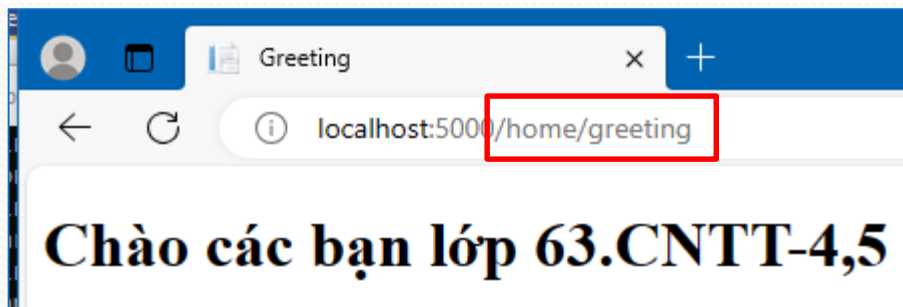
# Bài tập 1

- Tạo project tên Web\_Topic1 trong solution WebApp, khi chạy sẽ cho kết quả như sau: (sử dụng Controller)



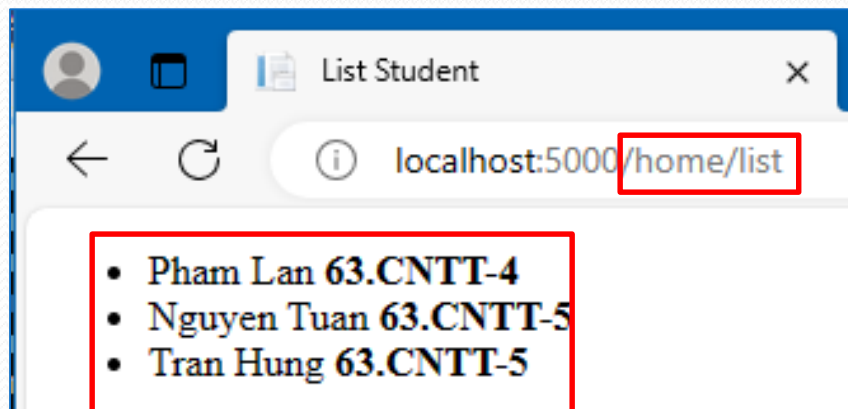
## Bài tập 2

- Thêm các thành phần View và phương thức Greeting() trong HomeController.cs của project Web\_Topic1, khi chạy sẽ cho kết quả như sau: (sử dụng Controller, View)



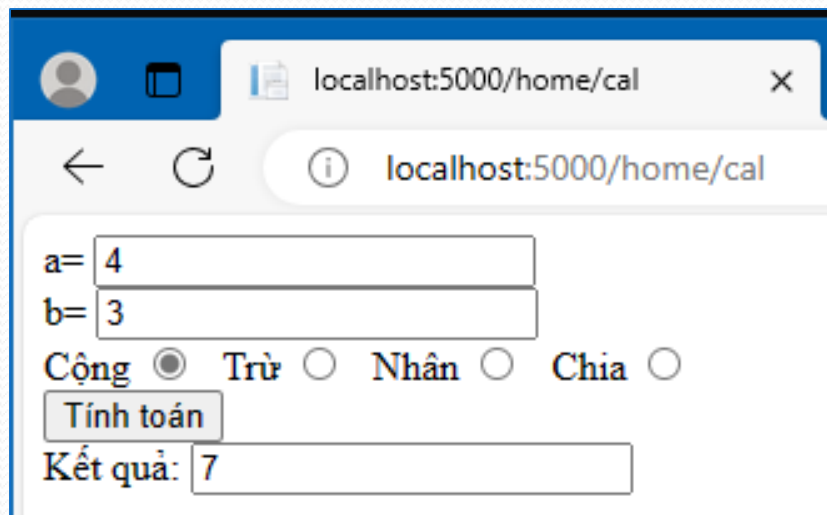
## Bài tập 3

- Thêm các thành phần View, Model và phương thức List() trong HomeController.cs của project Web\_Topic1, khi chạy sẽ cho kết quả như sau: (sử dụng Controller, View, Model)



## Bài tập 4

- Thêm các thành phần Model, View, Controller vào project Web\_Topic1, khi chạy sẽ có kết quả như sau:



localhost:5000/home/cal

← ↻ ⓘ localhost:5000/home/cal

a= 4

b= 3

Cộng ☒ Trừ ☐ Nhân ☐ Chia ☐

Tính toán

Kết quả: 7

# Kết thúc chủ đề 1

---

