



# **LẬP TRÌNH C# 4**

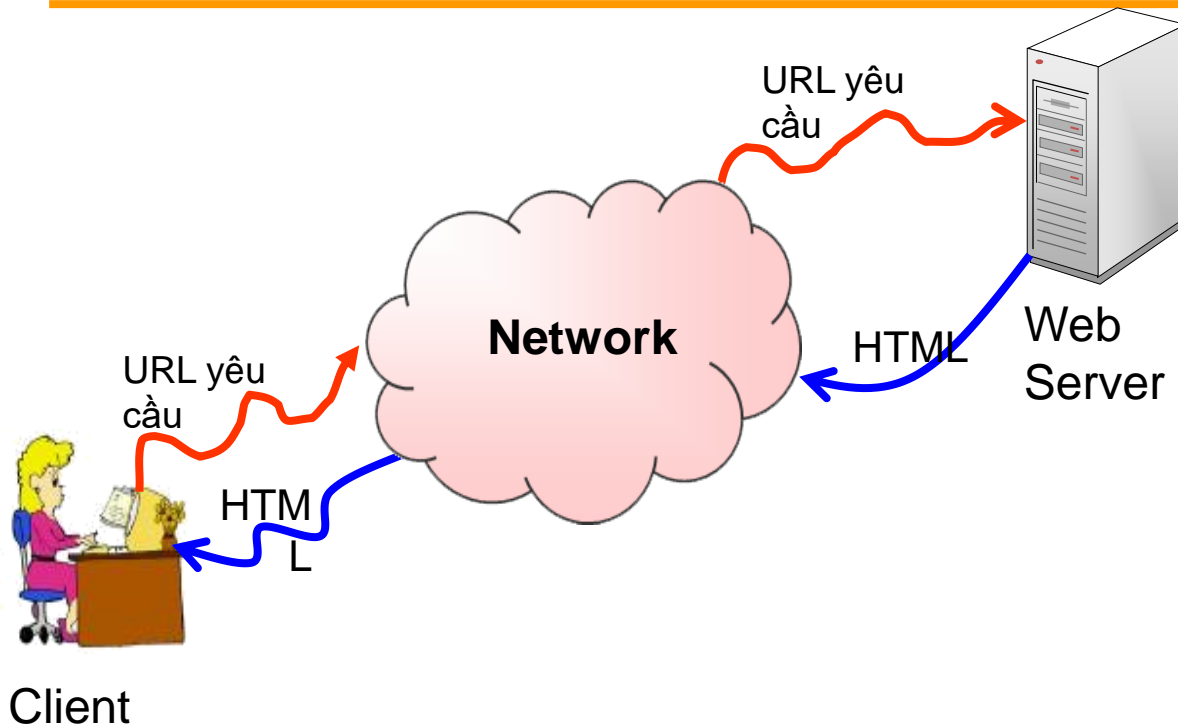
## **BÀI 1: TỔNG QUAN ASP.NET CORE**

- ⊙ Cơ bản về lập trình Web
- ⊙ .NET Core và .Net Framework
- ⊙ ASP.NET Core
- ⊙ Dự án ASP.NET Core từ Visual Studio

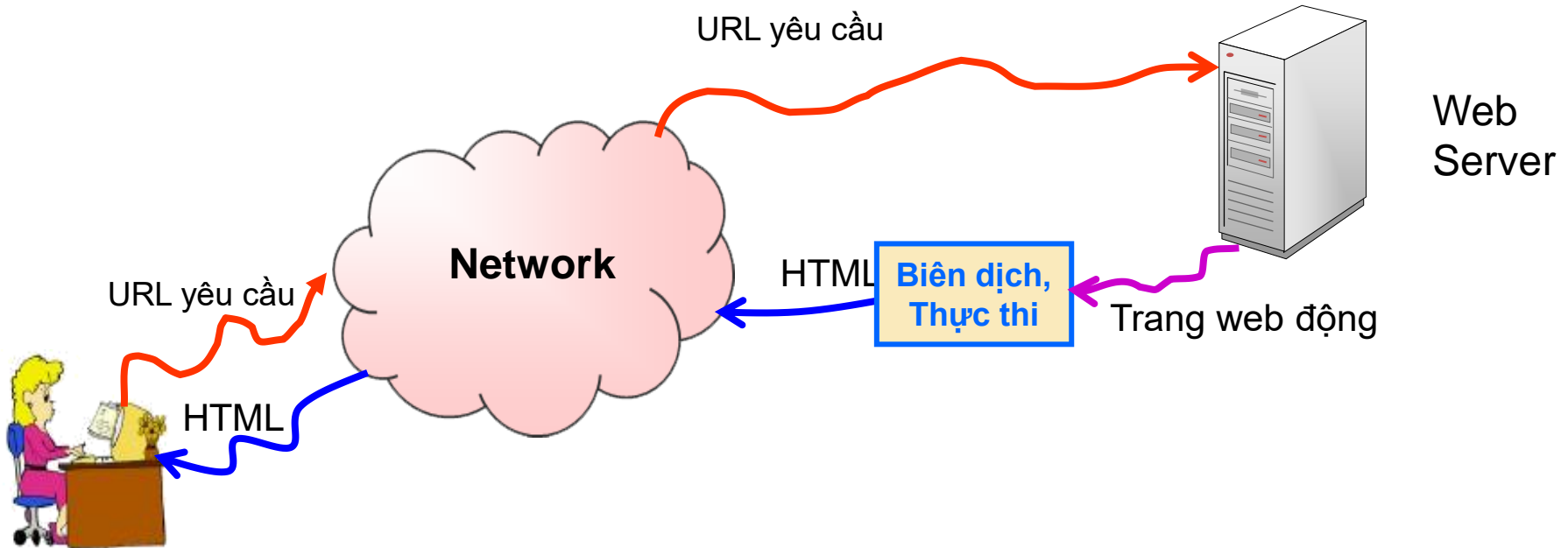


## ❑ Một số thuật ngữ

- ❖ **Web page:** Trang web , tập tin văn bản có nội dung gồm: content + presentation data được viết trong markup language.
- ❖ **Web Browser:** Phần mềm duyệt web, Internet Explorer, FireFox, Chrome,... Web browser biết cách kết nối với server để lấy về một web page theo yêu cầu của user. Web browser biết cách thông dịch các thẻ đánh dấu (presentation data) trong web page để thể hiện nội dung ra màn hình.
- ❖ **Web server:** Phần mềm có khả năng đáp ứng yêu cầu lấy thông tin từ Web browser
- ❖ **Database:** Cơ sở dữ liệu lưu trữ nhờ một RDBMS như SQL Server, Oracle.
- ❖ **Protocol:** Giao thức, cách thức truyền thông trên mạng, một tập các quy định về tổ chức vật lý của một đơn vị thông tin được truyền trên mạng.



- ☐ Mọi người sử dụng nhận được kết quả giống nhau
- ☐ Trang web được viết bằng HTML, chỉ thay đổi khi có sự thay đổi của người xây dựng
- ☐ Khả năng tương tác yếu
- ☐ Webserver hoạt động giống 1 file server

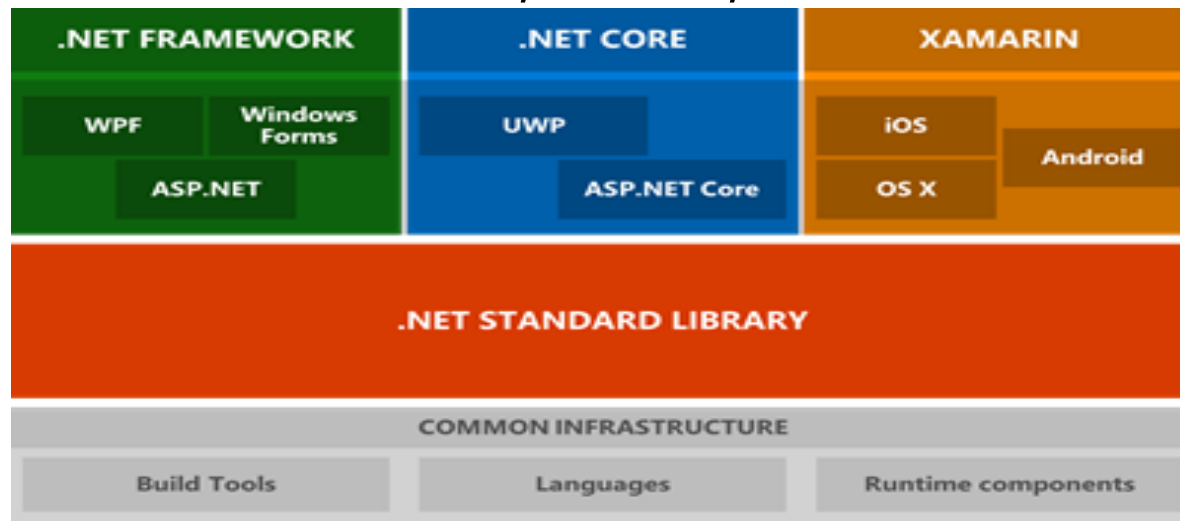


Client

- ❑ Mỗi người sử dụng có thể nhận được nội dung khác nhau phụ thuộc vào kết quả chạy chương trình
- ❑ Trang web viết bằng HTML + Ngôn ngữ lập trình phía server. Có thể được thay đổi bởi người sử dụng
- ❑ Khả năng tương tác mạnh

- ☐ Asp.Net web form
- ☐ Asp.Net MVC
- ☐ Asp.Net Core
- ☐ PHP – Personal Home Page
- ☐ Java Servlet
- ☐ JSP – Java Server Page
- ☐ Python

- ❑ .NET Framework chỉ hoạt động trên Windows. Những nền tảng ứng dụng như WPF, Winforms, ASP.NET(1-4) hoạt động dựa trên .NET Framework.
- ❑ .Net Core là một open-source mới và framework đa nền tảng (cross-platform) cho việc xây dựng những ứng dụng hiện tại dựa trên kết nối đám mây, giống như web apps, IoT và backend cho mobile. Do là một framework đa nền tảng nên .NET hỗ trợ hầu hết các hệ điều hành như Windows, Linux, macOS.



## ☐ Sử dụng .NET Core khi ?

- ❖ Cần đa nền tảng (Cross-platform)
- ❖ Targeting microservices
- ❖ Sử dụng Docker containers.
- ❖ Cần xây dựng ứng dụng Machine learning với kiến thức C#
- ❖ Cần cho các hệ thống hiệu suất cao và có thể mở rộng
- ❖ Các phiên bản cần chạy đa luồng khi xử lý

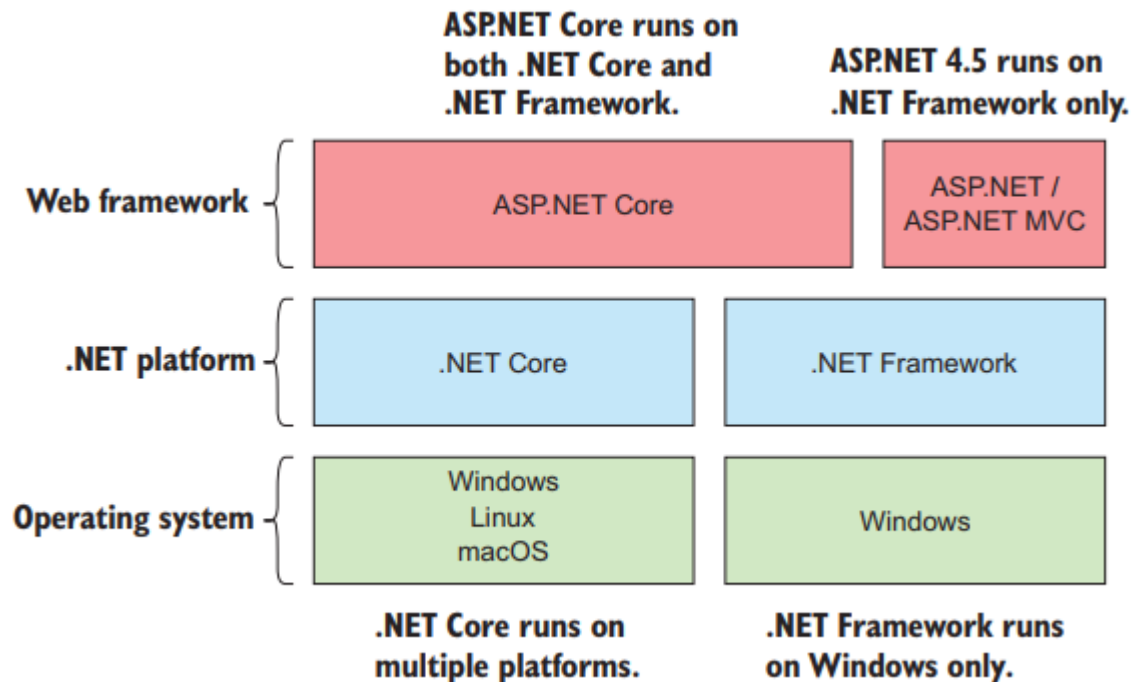
## ☐ Sử dụng .Net framework khi ?

- ❖ Nếu ứng dụng của bạn sử dụng .NET và nuget bên thứ 3 mà không hỗ trợ .NET Core.
- ❖ Nếu ứng dụng của bạn sử dụng công nghệ .NET và không hỗ trợ .NET Core.
- ❖ Nếu ứng dụng của bạn sử dụng platform và không hỗ trợ .NET Core.

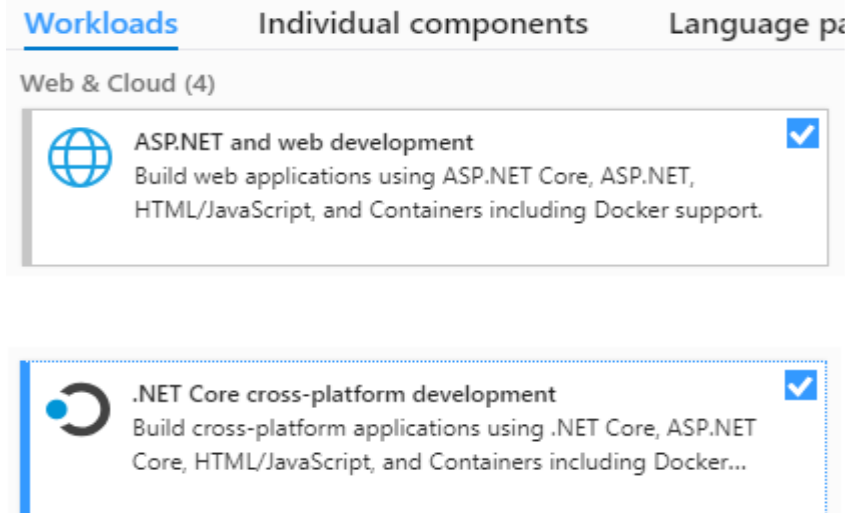


- ❑ ASP.NET ra đời từ 2002 với vai trò là một bộ phận của .NET Framework 1.0 nhằm thay thế cho ASP
- ❑ Microsoft xây dựng một số mô hình lập trình khác nhau để hỗ trợ lập trình viên: (1) Web Forms, (2) MVC.
- ❑ ASP.NET gắn chặt với dịch vụ hosting của Windows sử dụng IIS (Internet Information Service). Điều này khiến ASP.NET không thể mở rộng ra hoạt động trên các hệ điều hành khác – vốn là một yêu cầu rất quan trọng hiện nay.
- ❑ Không nên nhầm lẫn giữa ASP.NET và ASP.NET Core. Chúng là hai framework hoàn toàn khác nhau

- ❑ là framework của Microsoft hỗ trợ phát triển các ứng dụng web. Phiên bản đầu tiên ASP.NET Core phát hành tháng 6 năm 2016.
- ❑ ASP.NET Core có thể hoạt động trên .NET Framework



- ❑ Để phát triển ứng dụng ASP.NET Core bạn cần các công cụ sau:
  - ❖ Bộ ASP.NET Core SDK(2.x hoặc 3.x)  
(<https://www.microsoft.com/net/download/core>)
  - ❖ Một môi trường phát triển ứng dụng tích hợp như Visual Studio hoặc một code editor như Visual Studio Code
- ❑ Cài đặt ASP.NET Core SDK với Visual Studio 2019

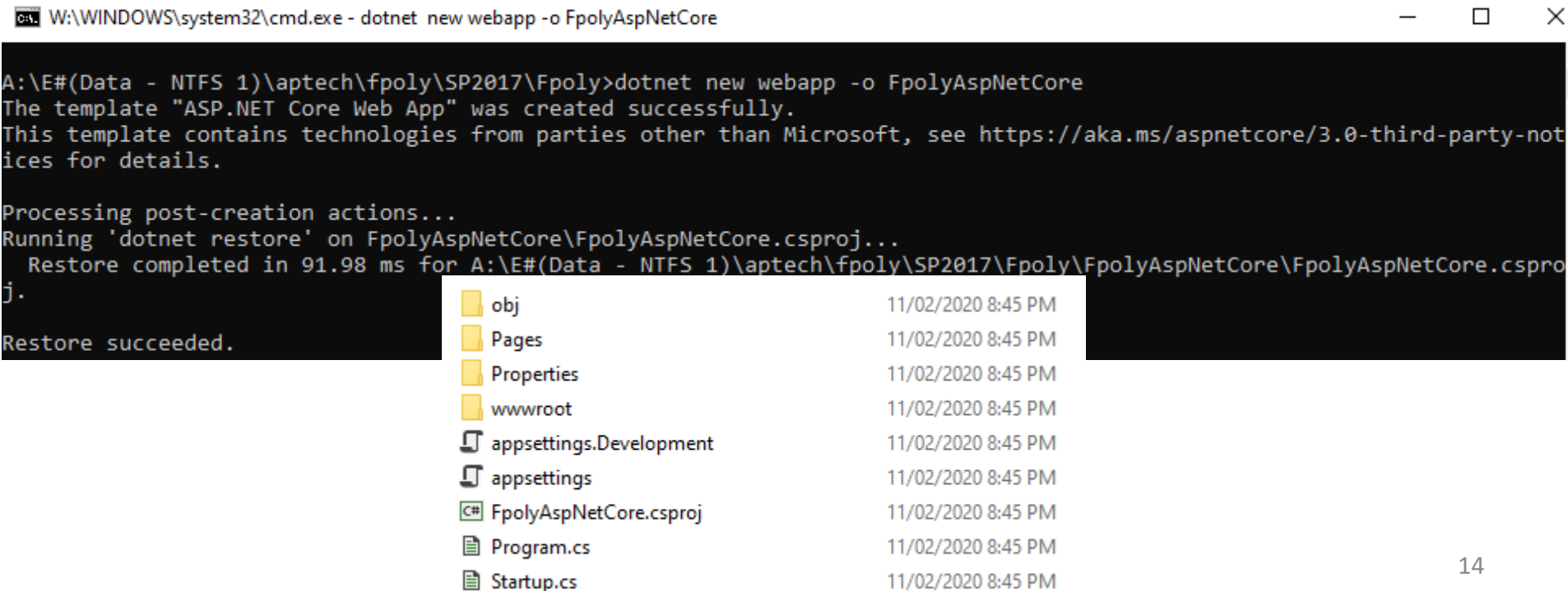


- ❑ Dotnet CLI là giao diện dòng lệnh trong ASP.NET Core
- ❑ Được cài đặt như một phần của .NET Core
- ❑ Cú pháp .NET CLI gồm 3 phần: từ khóa chính, động từ, và các tham số: `dotnet [verb] [arguments]`
  - ❖ Tên từ khóa chính là "dotnet"
  - ❖ Động từ là lệnh mà bạn muốn thực thi. Lệnh thực hiện một hành động
  - ❖ Các tham số sẽ được pass vào để cung cấp thêm thông tin cho hành động
- ❑ Các lệnh Dotnet CLI: <https://docs.microsoft.com/en-us/dotnet/core/tools/>

## Một số câu lệnh phổ biến thường dùng cho Dotnet CLI

new	Tạo mới project, file cấu hình hay solution.	vstest	Thực thi unit test từ file cụ thể
restore	Tải về các thành phần phụ thuộc trong project.	pack	Đóng gói code vào NuGet package.
build	Biên dịch dự án với các thành phần liên quan	clean	Làm sạch output của dự án.
publish	Đóng gói toàn bộ ứng dụng ra một thư mục.	sln	Chỉnh sửa file .NET Core solution
run	Chạy source code mà không cần lệnh biên dịch.	help	Xem thêm trợ giúp
test	Thực thi unit test	store	Lưu trữ các assembly cụ thể

- ❑ Có thể sử dụng CLI từ Command Prompt hoặc PowerShell của windows:
  - ❖ Chuyển tới thư mục nơi sẽ tạo project
  - ❖ Gõ lệnh `dotnet new ProjectName`, enter chờ tạo project
- ❑ Ví dụ tạo dự án Asp.net core tên FpolyAspNetCore



```
W:\WINDOWS\system32\cmd.exe - dotnet new webapp -o FpolyAspNetCore

A:\E#(Data - NTFS 1)\aptech\fpoly\SP2017\Fpoly>dotnet new webapp -o FpolyAspNetCore
The template "ASP.NET Core Web App" was created successfully.
This template contains technologies from parties other than Microsoft, see https://aka.ms/aspnetcore/3.0-third-party-notices for details.

Processing post-creation actions...
Running 'dotnet restore' on FpolyAspNetCore\FpolyAspNetCore.csproj...
  Restore completed in 91.98 ms for A:\E#(Data - NTFS 1)\aptech\fpoly\SP2017\Fpoly\FpolyAspNetCore\FpolyAspNetCore.csproj.
Restore succeeded.
```

obj	11/02/2020 8:45 PM
Pages	11/02/2020 8:45 PM
Properties	11/02/2020 8:45 PM
wwwroot	11/02/2020 8:45 PM
appsettings.Development	11/02/2020 8:45 PM
appsettings	11/02/2020 8:45 PM
FpolyAspNetCore.csproj	11/02/2020 8:45 PM
Program.cs	11/02/2020 8:45 PM
Startup.cs	11/02/2020 8:45 PM

- ❑ Chạy ứng dụng sử dụng dotnet run
  - ❖ Đặt thư mục chứa project là thư mục làm việc
  - ❖ Gõ lệnh dotnet run và ấn Enter
  - ❖ ASP.NET Core sẽ hoạt động như một Console App, tự lắng nghe truy vấn ở cổng 5000
  - ❖ Mở trình duyệt <http://localhost:5000> xem kết quả

```
A:\E#(Data - NTFS 1)\aptech\fpoly\SP2017\Fpoly\FpolyAspNetCore>dotnet run
info: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[0]
      User profile is available. Using 'W:\Users\Thepv\AppData\Local\ASP.N
Windows DPAPI to encrypt keys at rest.
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
      Now li
```

← → ↻ 🔒 localhost:5001

FpolyAspNetCore Home Privacy

# Welcome



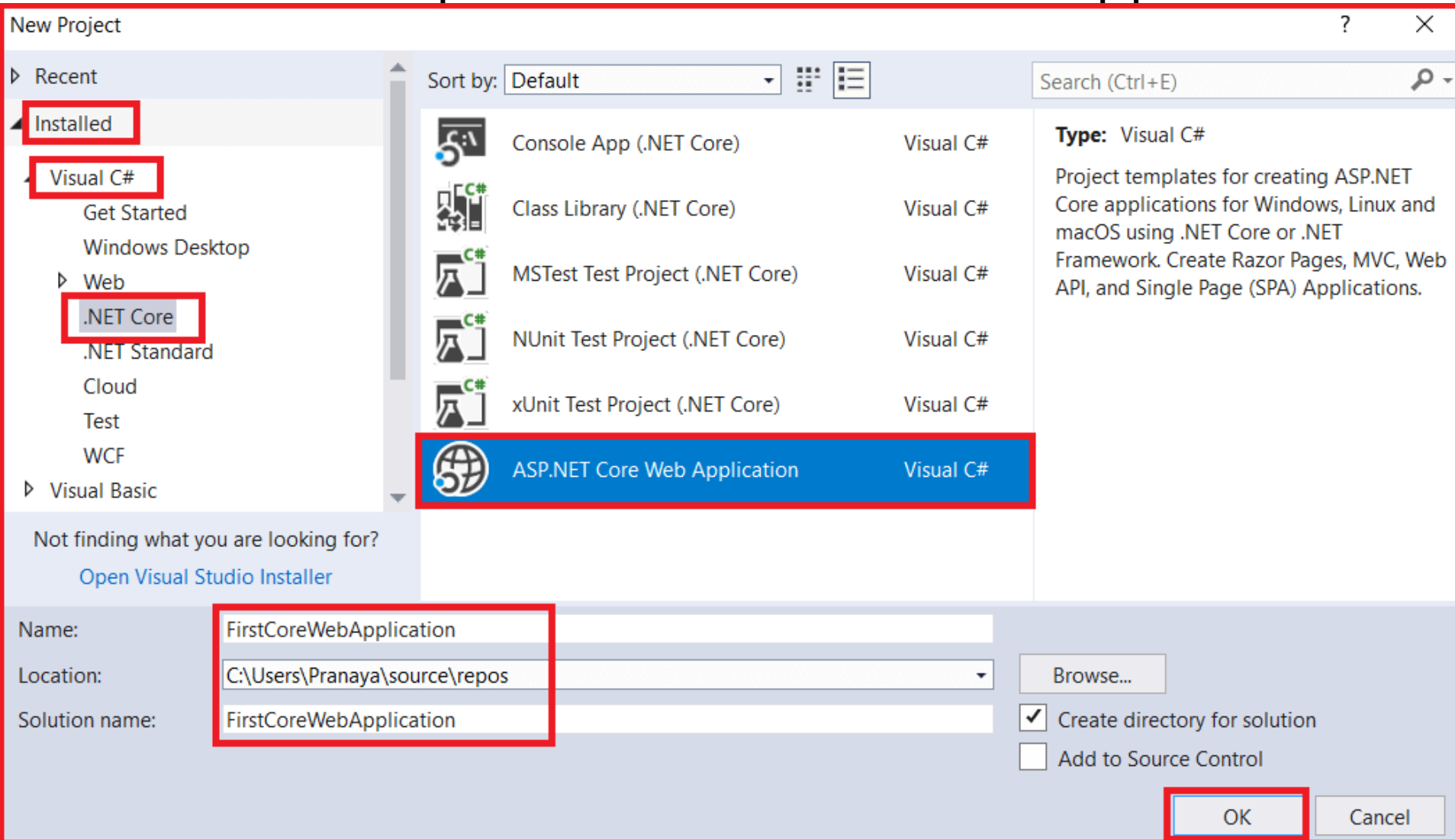




# **LẬP TRÌNH C# 4**

## **BÀI 1: TỔNG QUAN ASP.NET CORE(P2)**

## ❑ Lựa chọn template ASP.NET Core Web Application











- ❑ Lựa chọn phiên bản Asp.net core và loại ứng dụng web cần tạo
  - ❖ Empty: template trống, bạn cần tự mình cài đặt các gói cần thiết
  - ❖ API: template cho RESTful HTTP service, chỉ có Model và Controller
  - ❖ Web Application: còn gọi là Razor Pages, chỉ có View
  - ❖ Web Application (Model-View-Controller): template cho ứng dụng theo kiến trúc MVC
  - ❖ Angular/React.js: ứng dụng đơn trang (SPA) sử dụng JavaScript framework tương ứng

## Lựa chọn ASP.NET Core 2.2 và empty template

New ASP.NET Core Web Application - FirstCoreWebApplication

.NET Core ASP.NET Core 2.2 [Learn more](#)

 Empty  API  Web Application  Web Application (Model-View-Controller)  Razor Class Library

 Angular  React.js  React.js and Redux

[Get additional project templates](#)

☐ Enable Docker Support (Requires [Docker for Windows](#))

OS: Windows

☐ Configure for HTTPS **Uncheck this checkbox**

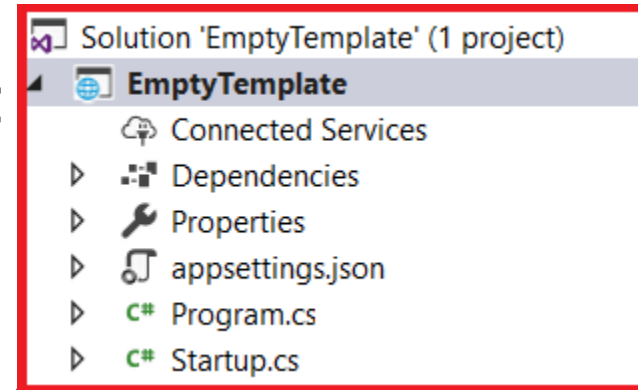
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.  
[Learn more](#)

Author: Microsoft  
Source: SDK 2.2.102

Authentication: **No Authentication**  
[Change Authentication](#)

**OK** Cancel

- ❑ Tạo thành công project có cấu trúc
- ❑ Dịch và chạy ứng dụng ASP.NET Core từ Visual Studio:

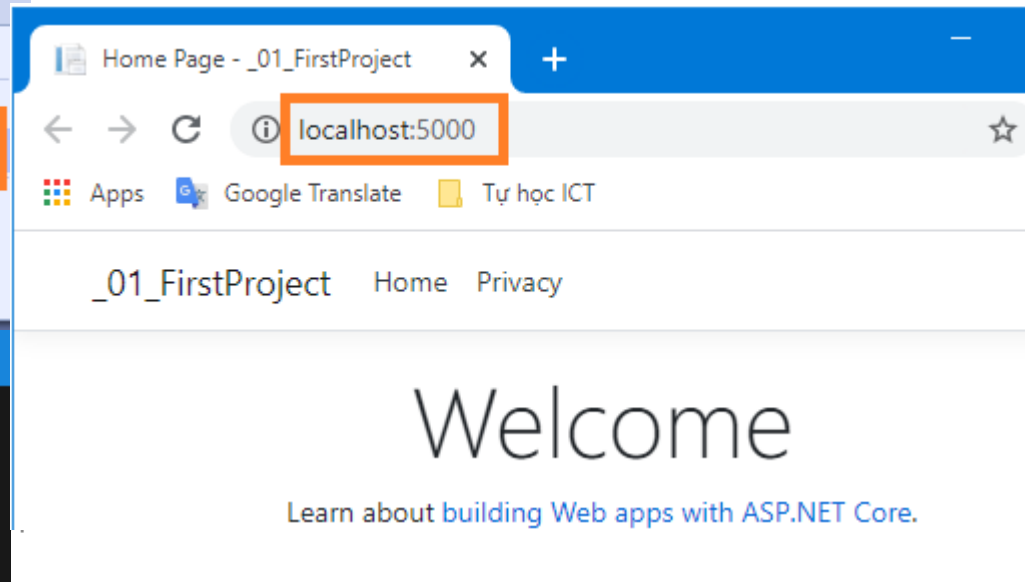
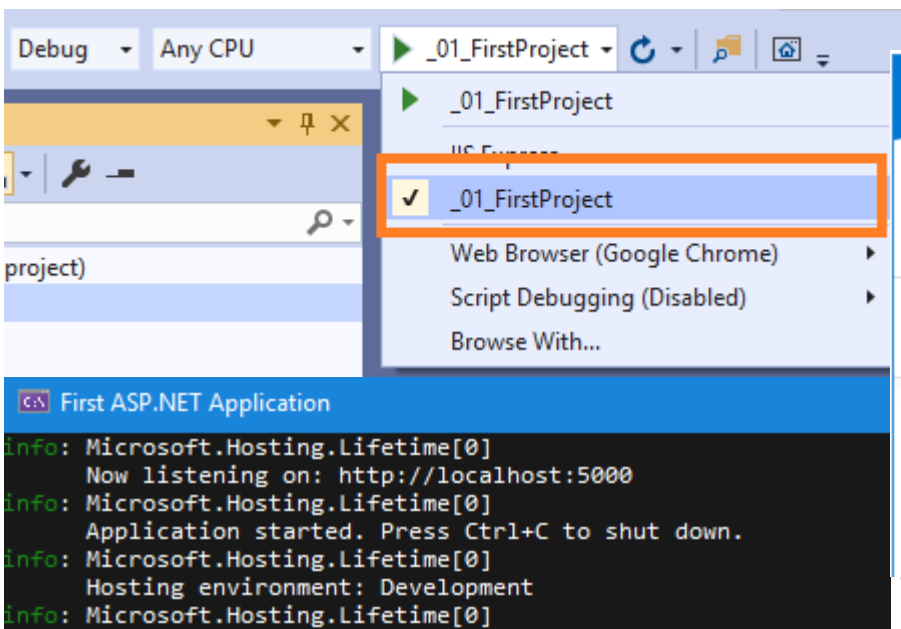


- ❖ Chạy ứng dụng với IIS Express: **IIS** là viết tắt của từ (**Internet Information Services**), được đính kèm với các phiên bản của Windows. IIS chính là các dịch vụ dành cho máy chủ chạy trên nền hệ điều hành Window nhằm cung cấp và phân tán các thông tin lên mạng, nó bao gồm nhiều dịch vụ khác nhau như Web Server, FTP Server...

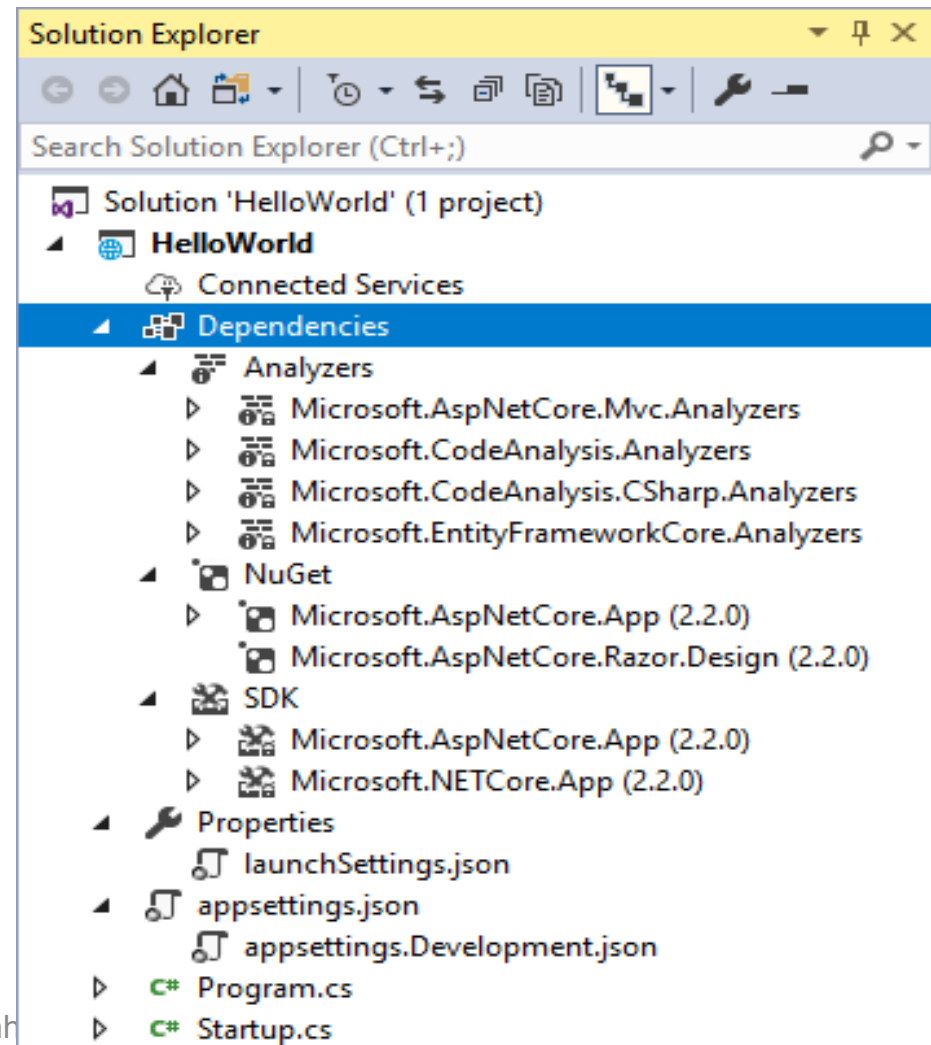


## ❑ Dịch và chạy ứng dụng ASP.NET Core từ Visual Studio:

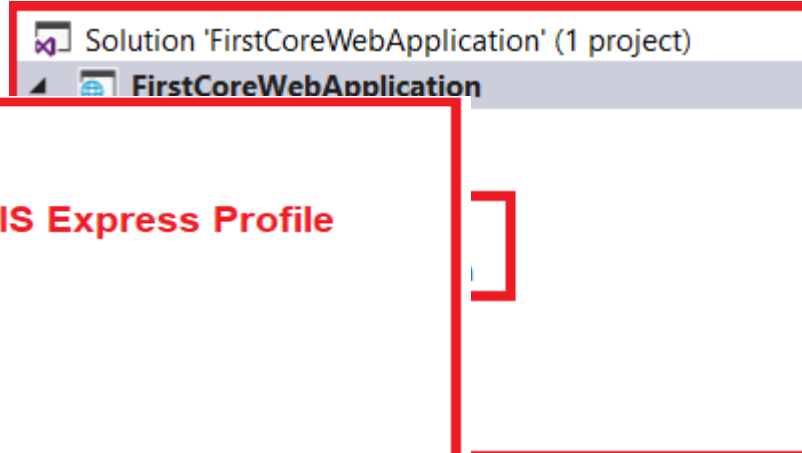
- ❖ Chạy ứng dụng ở dạng Console App: ứng dụng ASP.NET Core hoạt động như một web server hoàn toàn độc lập: nó có thể tự tiếp nhận và xử lý truy vấn HTTP từ trình duyệt. Không cần IIS/IIS Express.
- ❖ chương trình ASP.NET Core mặc định sẽ lắng nghe ở cổng 5000



- ❑ Một solution ASP.NET Core sử dụng Empty template có cấu trúc
- ❑ Solution chứa 3 folder là Dependencies, Properties và wwwroot.
- ❑ 2 file rất quan trọng là Program.cs và Startup.cs.



- ❑ Thư mục Properties có file **launchSettings.json** chứa tất cả thông tin cụ thể cài đặt project để có thể chạy được ứng dụng.



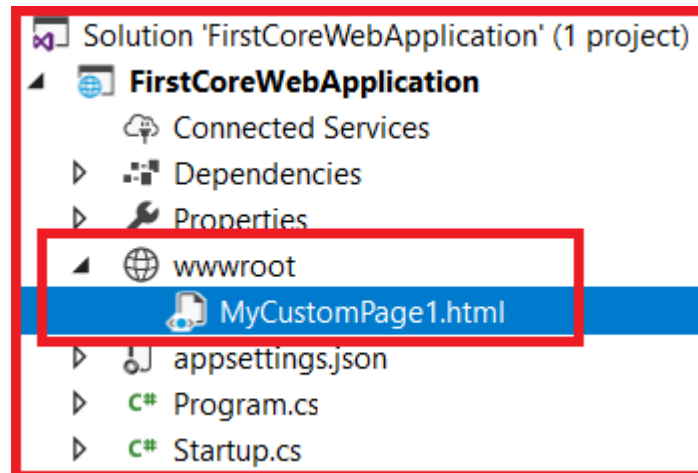
```
"profiles": {  
  "IIS Express": {  
    "commandName": "IISExpress",  
    "launchBrowser": true,  
    "environmentVariables": {  
      "ASPNETCORE_ENVIRONMENT": "Development"  
    }  
  },  
  "FirstCoreWebApplication": {  
    "commandName": "Project",  
    "launchBrowser": true,  
    "applicationUrl": "http://localhost:5000",  
    "environmentVariables": {  
      "ASPNETCORE_ENVIRONMENT": "Development"  
    }  
  }  
}
```

IIS Express Profile

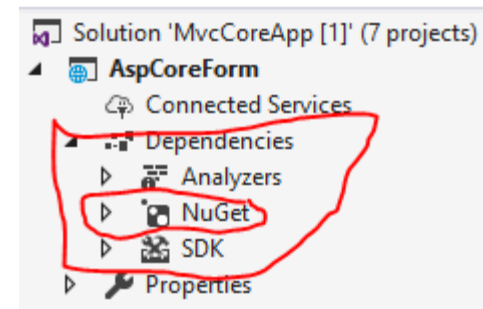
FirstCoreWebApplication  
Profile. This is the name  
of your application



- ❑ File tĩnh như HTML, CSS, Javascript hay hình ảnh sẽ đặt trong này hoặc các thư mục con của nó.
- ❑ Chỉ đặt các file mà bạn muốn publish ra internet vào thư mục này



- ❑ chứa toàn bộ các thành phần phụ thuộc dùng cho ứng dụng, chúng ta gọi là Dependencies.
- ❑ Visual Studio sử dụng Nuget Packages để quản lý tất cả các dependencies phía server.
- ❑ Với Client tức là các thư viện JS thì chúng ta sử dụng Client-side Libraries
- ❑ Trong thư mục **Dependencies**, chúng ta có thư mục Nuget chứa các Nuget Packages đang sử dụng và thư mục SDK chứa **Microsoft.NETCore.App**. Đây là .NET Core Runtime sẽ sử dụng trong project.

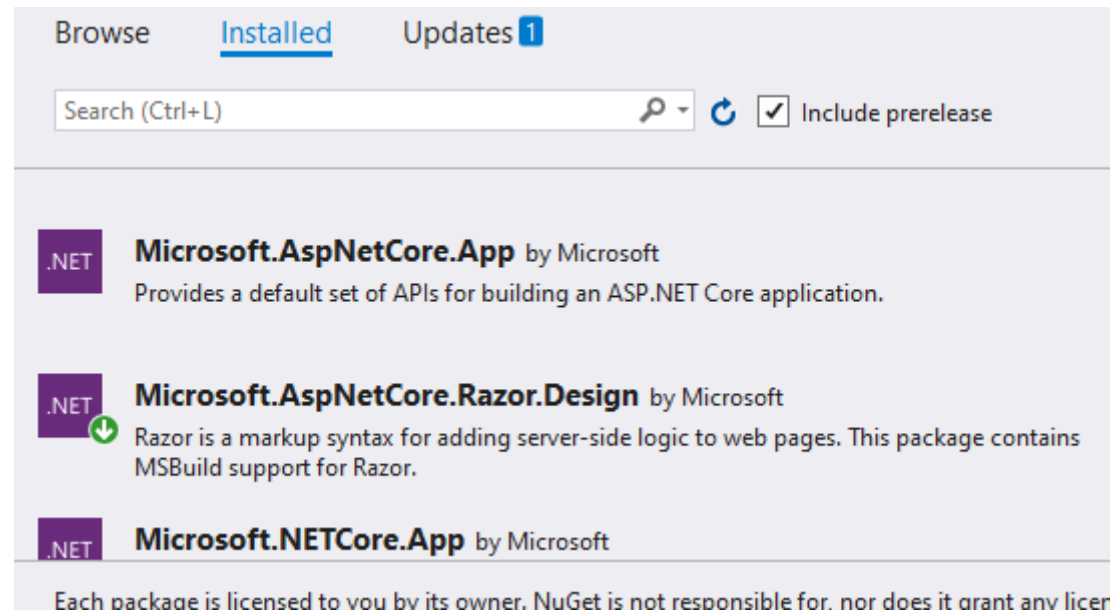
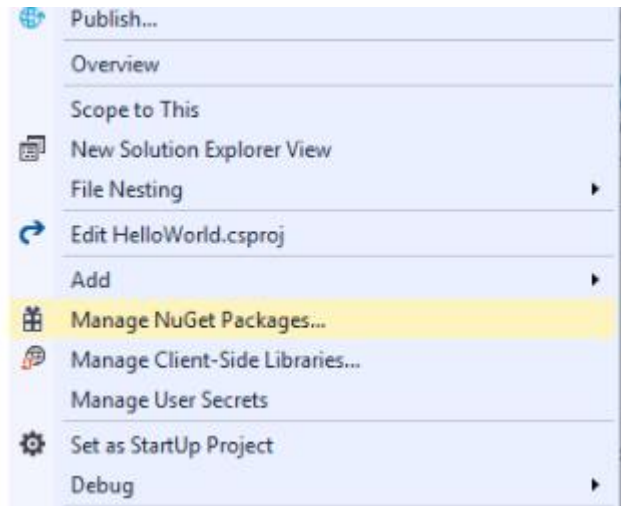


- ❑ Có 3 cách để tải Nuget Package vào project:
  - ❖ Chỉnh sửa file .csproj, ví dụ thêm Microsoft.ApplicationInsights



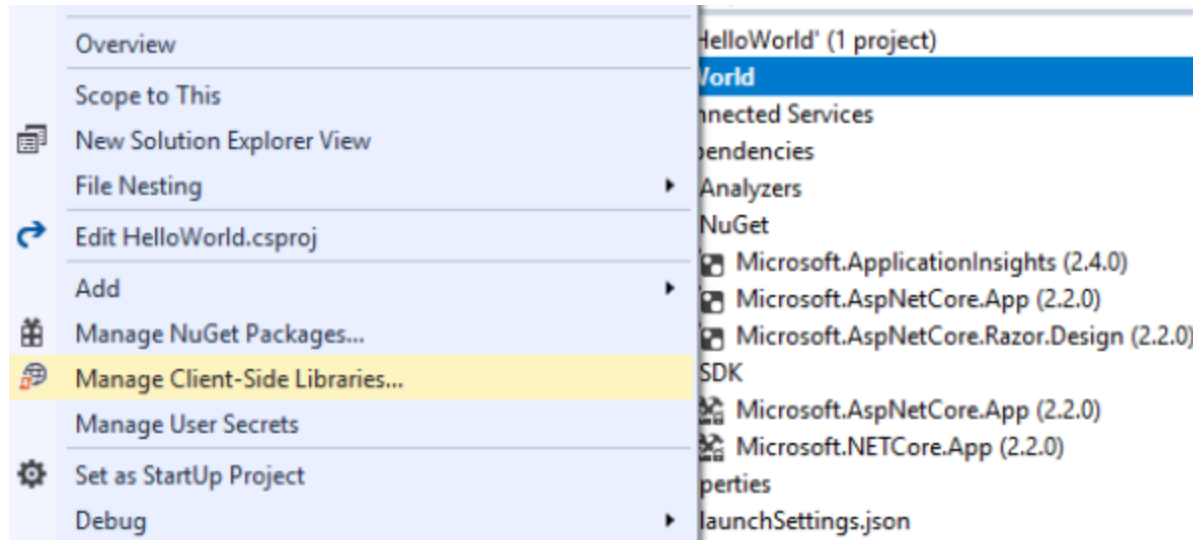
**Nuget Package Manager** sẽ tự download và cài đặt gói này về

- ❑ Có 3 cách để tải Nuget Package vào project:
  - ❖ Sử dụng Nuget Package Manager



## ❑ Có 3 cách để tải Nuget Package vào project:

- ❖ Sử dụng Bower Packages: Chuột phải vào project và chọn **Manage Client-side Libraries** nó sẽ tạo ra file libman.json nơi bạn có thể thêm hay xóa các thư viện phía client. Nó sẽ tự tạo folder theo thuộc tính destination của từng thư viện

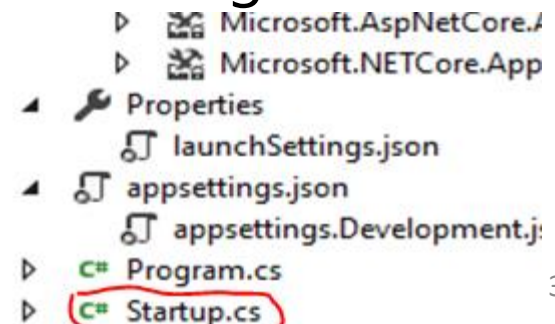


- ❑ Ứng dụng vừa tạo ra có hàm Main và đây là điểm khởi đầu của ứng dụng (entry point)

```
public class Program
{
    public static void Main(string[] args)
    {
        BuildWebHost(args).Run(); ← 3. Run it

        public static IWebHost BuildWebHost(string[] args) => ← 1. Create a Host
        {
            WebHost.CreateDefaultBuilder(args)
                .UseStartup<Startup>()
                .Build(); ← 2. Build it
        }
    }
}
```

- ❖ `.UseStartup<Startup>()`: WebHost sẽ gọi phương thức `UseStartup` trước khi build lên host, Host sẽ tìm kiếm phần cài đặt từ 2 phương thức `Configure` và `ConfigureServices` trong file **Startup.cs**



- ❑ Startup.cs là một class có hai chức năng chính:
- ❖ Cấu hình request pipeline để xử lý các request gửi đến ứng dụng
  - ❖ Cấu hình các services cho phần dependency injection

```
public IConfiguration Configuration { get; }

// This method gets called by the runtime. Use this method to add services
public void ConfigureServices(IServiceCollection services)
{
    services.AddRazorPages();
}

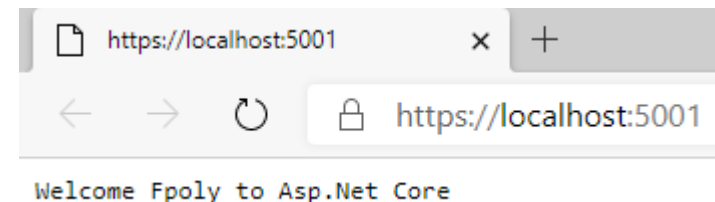
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
}
```

❑ Trong phương thức `Configure` thay đổi nội dung như:

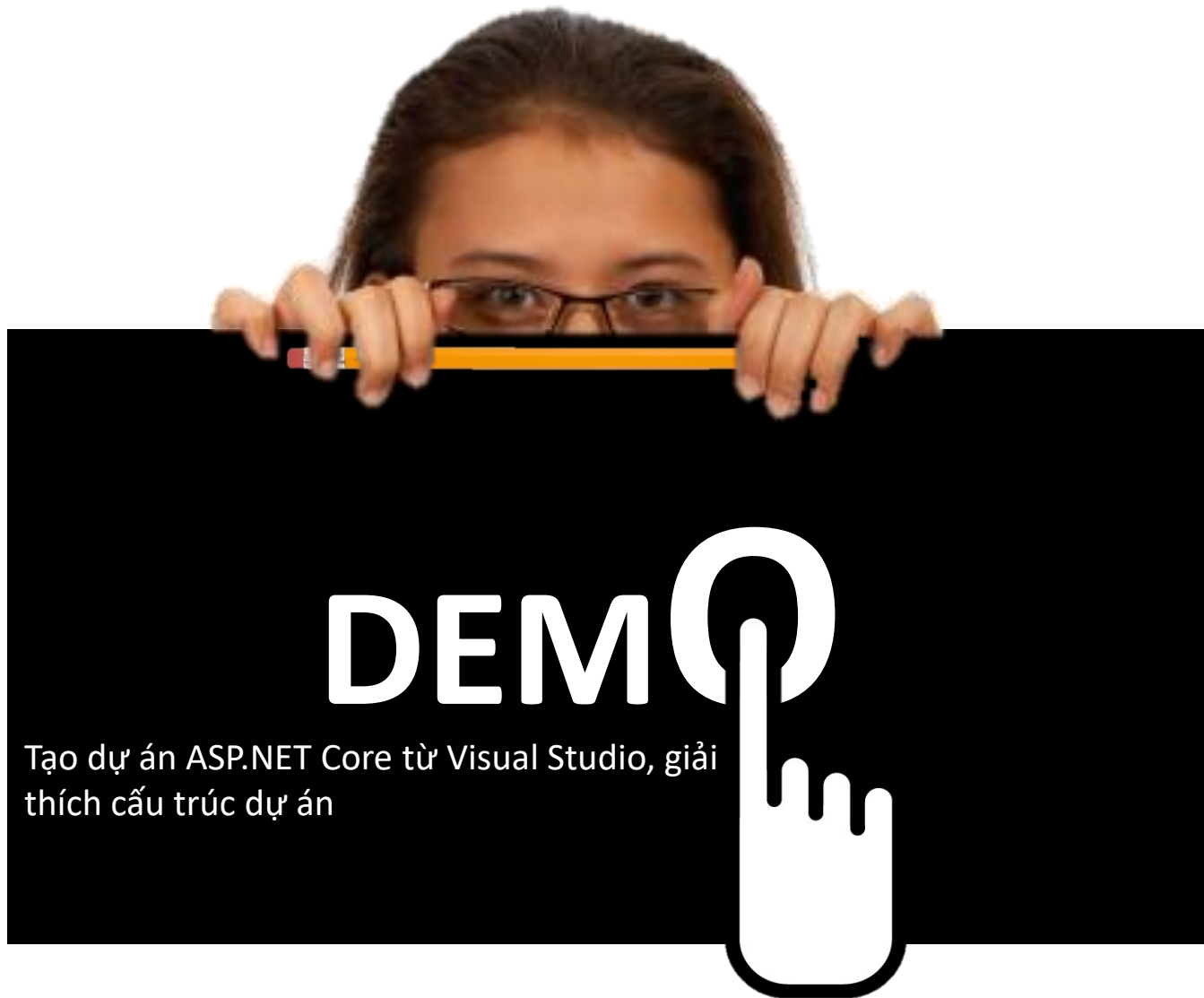
```
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }

    app.Run(async (context) =>
    {
        await context.Response.WriteAsync("Welcome Fpoly to Asp.Net Core");
    });
}
```

- ❖ Phương thức **app.Run** để lấy thể hiện của **HttpContext**
- ❖ `await context.Response.WriteAsync`: trả về một chuỗi theo cơ chế bất đồng bộ







## Tổng kết bài học

- ◎ Cơ bản về lập trình Web
- ◎ .NET Core và .Net Framework
- ◎ ASP.NET Core
- ◎ Dự án ASP.NET Core từ Visual Studio





**KẾT THÚC**