**We will create an ASP.NET core project from scratch.** As we build this project and progress through the course, we will discuss everything you need to know to build data driven web applications using this new ASP.NET Core web framework from Microsoft.  
  
  
**We will be discussing in detail**

* ASP.NET Core
* ASP.NET Core MVC
* ASP.NET Identity Core for security
* Entity Framework Core for data access

**What is ASP.NET Core and what are it's benefits**  
ASP.NET Core is a cross-platform, high-performance, open-source framework for building modern, cloud-based, Internet-connected applications.  
  
ASP.NET Core is a redesign of ASP.NET 4.x. For this reason it was initially, called ASP.NET 5, but later renamed to ASP.NET Core 1.0.  
  
ASP.NET Core has been redesigned from the ground up and offers the following benefits and features.  
  
**Cross Platform :**ASP.NET 4.x applications run only on windows platform, where as ASP.NET Core applications can be developed and run across different platforms like Windows, macOS, or Linux. ASP.NET 4.x applications can be hosted only on IIS, where as ASP.NET Core applications can be hosted on IIS, Apache, Docker, or even self-host in your own process. From a development standpoint, you can either use Visual Studio or Visual Studio Code for building .NET Core applications. You can also use third party editors like Sublime.  
  
**Unified Programming Model for MVC and Web API :** With ASP.NET core, we use the same unified programming model to create MVC style web applications and ASP.NET Web API's. In both the cases, the Controller that we create inherits from the same Controller base class and returns IActionResult. As the name implies IActionResult is an interface and it has got several implementations. ViewResult and JsonResult are just 2 examples of the built-in result types that implement IActionResult interface. So, in the case of a Web API, the controller returns a JsonResult and in the case of an MVC style web application it returns a ViewResult. If this does not make much sense at the moment, do not worry, it will be crystal clear as we progress through the course.  
  
**Dependency Injection :**Out of the box, ASP.NET Core has built-in support for dependency injection. If you are new to this powerful concept, please do not worry, we will discuss it in detail as we progress through this couse.  
  
**Testability :** With built-in dependency injection and the unified programming model for creating Web Applications and Web API's, unit testing ASP.NET Core applications is easy.  
  
**Open-source and community-focused :**ASP.NET Core is fully open source and is being actively developed by the .NET team in collaboration with a vast community of open source developers. So, ASP.NET core is continually evolving as the vast community behind it is suggesting ways to improve it and help fix bugs and problems. This means we have a more secure and better quality software.  
  
**Modular HTTP Request Pipeline :** ASP.NET Core Provides Modularity with Middleware Components. In ASP.NET Core, we compose the request and response pipeline using the middleware components. It includes a rich set of built-in middleware components. We can also write our own custom middleware components. As we progress through the course we will be discussing, what middleware components are and using them to compose request and response pipeline.  
  
**Course prerequisites:**

* Basic HTML, CSS and C#. You do not have to be an expert by any means. All you need is the basic knowledge.
* Prior MVC knowledge is helpful but not required. Even if you do not have any experience with MVC, that's OK. We will be discussing all the required MVC basics in this course.

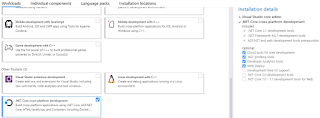
we will **install the required software and set up our machine for .NET Core development**.  
  
**There are 2 softwares that we need to install for .net core application development.**

* An Editor
* .NET Core SDK (Software Development Kit)

**Download and Install an editor for .NET Core Development**  
  
I have windows operating system, so, I will be using **Visual Studio**as the editor for .NET Core application development. You can use any editor of your choice. 

* Visual Studio
* Visual Studio Code
* Sublime
* Vim
* Atom
* Etc.

**Download and install Visual Studio**

* Visual Studio Community Edition is free
* As of this recording the latest version is Visual Studio 2017 and can be downloaded from <https://visualstudio.microsoft.com/>
* To develop .NET Core applications in Visual Studio 2017, please select .NET Core cross-platform development workload  
  [](https://4.bp.blogspot.com/-XzaLWj9hH28/XCiKMPpEe-I/AAAAAAAArkY/FrKGv27jMz8MBnSPiSPjYkc_h_e4FSL1ACLcBGAs/s1600/asp.net%2Bcore%2Bsoftware%2Brequirements.png)
* With this selection, .NET Core SDK 2.1 is installed
* You can verify this by creating a new .NET Core Application
* Notice, in the .NET Core Dropdownlist we only have ASP.NET Core 2.1
* As of this recording, the latest stable version of .NET Core is 2.2

**Download and install .NET Core SDK**

* Navigate to <https://dotnet.microsoft.com/download>
* Depending on the operating system you have, download and install the SDK.
* The SDK includes everything you need to build and run .NET core applications.
* The .NET Core Runtime is already included in the SDK. So, if you have installed the SDK, there is no need to install .NET Core Runtime
* The .NET Core Runtime includes just the resources required to run existing .NET Core applications.

After .NET Core SDK 2.2 is installed, create a new .NET Core Application. Notice, in the .NET Core Dropdownlist, we now have ASP.NET Core 2.2 in addition to ASP.NET Core 2.1.  
  
