**MVC**

**# .Net ve Asp.Net ve .Net.Framework ve .Net.Core nedir?**

**.Net :** Microsoftundevoloperlar için geliştirdiği genel yapıya denebilir (Desktop da winform , Web de asp.net , cloud da azure gibi)

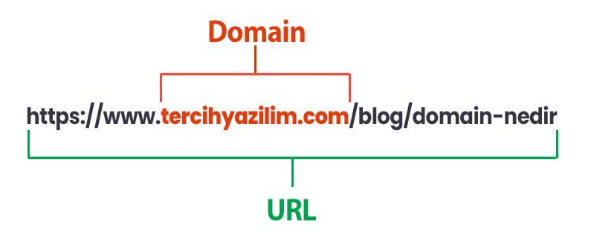
**Asp.Net**: ise .net yapısının web durumunda kullanılması için oluşturulmuş mimaridir Asp.Net mimarisi

**.Net FrameWork**: ise Winform Asp.Net gibi yapılarda yanlızca windows ekosistemine yönelik çözümler

**.Net Core:** Çeşitli işletim sistemlerini destler ve modüler olarak kullanılabilir

.net 5 den sonra tekrardan tek çatı altına girerek devam edilmiştir buradaki duruma .

**# Domain,Url, Hosting kavramları nelerdir ?**

**Domain :** Kişilerin web sitelerini  ziyaret etmek için tarayıcı url kısmına yazdıkları [web site](https://www.tercihyazilim.com/blog/web-site-nedir) adresleridir.

**Hosting** : Domain kısmında olan her türlü verinin sunucu ya da server gibi cihazlarda barınma durumunda kullanılan terim şeklinde ifade edile bilir ayrıca hostinglere ait depolanan web uygulamalarında ip le rile karıştırılmadan çalışma sağlanmaktadır

**Url:** Internet tarayıcısına yazılan adresdeki gidilmesi gereken yerin tamamaıdır

**# Http nedir ?**

Http yapısı bir protocol olup request ve response arasında client-server arasında ilişkiyi sağlayan model dir .Aradaki ilişki Get, Head, Post,Put,Delete,Trace,Options,Connect,Patch şeklinde olmaktadır.

**Get**: Sunucudan verileri listelemek için kullanılır

**Post**: Sunucuya verileri göndermek için kullanılır

**Put**: Var olan bir veriyi değiştirmek istediğinde kullanılmaktadır

**Patch**: Bir verinin bütün kısmı olmayıp bir kısmının güncellenmesi sağlanıyorsa kullanılır .

**Delete** : Var olan veriyi silmek için kullanılır

**# Server Nedir ? Hangi türleri bulunmaktadır?**

Web Uygulamaları yapılandırma ve geliştirmeleri bittikden sonra , online olarak diğer userların başvurabileceği temel yapılandırmalar olarak görülebilir , bunların bazı türleri şu şekildedir ;

**IIS**: Internet Information Services (IIS), esnek güvenli ve configure edilebilir bir web sunucusu olup asp.net dahil bir çok web uygulamasında kullanılmaktadır.

**Kestral** : Asp.Net Core uygulamalarında dahili olarak gelen bir web sunucusudur. IIS olmadan herhangi bir Microsoft sunucusu olmadığından Linux ya da farklı tabanlı sunucularda çalışabilmektedir

**Apache ve Nginix** : Linux gibi ortamlarda Asp.net Core uygulamalrında kestrel yapısı ile senkronize haberleşe bilen yapılardır

**Docker** : Yeni nesil sanallaştırma platformu olup , microservice yapılandırmalarında elimizdeki birden fazla servis tarafını ayağa kaldırmakda ciddi kullanılmaktadır .

**# MVC yapısı ve terimleri ile ilgili kısa açıklama yapabilir misiniz ?**

**MVC** yapısı bir framework yaklaşımı olduğu gibi aslında bir tasarım deseni olarak da kullanılabilmektedir

**Model** : Projenin iş mantığının tasarlandığı bölümdür, veritabanı modelleri başta olmak üzere görüntü verebilecek ve işlem yapılabilinecek modelleri burada tanımlarsınız. Bu modelleri projenizde size fazlasıyla yardımı olurlar. Doğrulama (Validation) işlemleri bu bölümde yapılabilinir.(Entity Framework , Entity Models , Ado.Net, Repository)

**View** : Projenin arayüzü veya kullanıcı arabimi de denilebilir (User Interface – UI), Bu bölümde HTML dosyaları üzerinde işlemler gerçekleştirilir.

**Control** : Projenin akışını, Model ile View arasındaki etkileşimleri kontrol eder. View dan gelen istekler Model yarımıyla yorumlanıp ne yapılacağı konusunda değerlendirilir. Fonksiyonel işlemlerin yapıldığı katmandır.Gelen isteklerin karşılanması durumuna bakıcaktır, verilerin görselleştirip , görselleştirmeme konusunda da controller katmanı aslında görevlidir

**Razor**: Asp.Net Mvc 3 teknolojisiyle birlikte gelen görüntüleme biçimidir

**ViewBag**: Verilerimizi Controllerdan View kısmına aktarmak için kullanırız.Runtime içerisinde oluşan dinamik bir nesnedir.

**ViewData**: Verilerimizi Controller dan View kısmına aktarmak için kullanırız.

**PartialView**: Sayfaların istediğimiz kısmına rahatça ekleyebileceğimiz, özel işlevleri yerine getirebilen, ancak kendi başına çalışmayan, bir sayfanın içerisinde entegre halde çalışan kısımlardır

# **MVC Pipeline asp.net core için nasıl çalışmaktadır** ?

