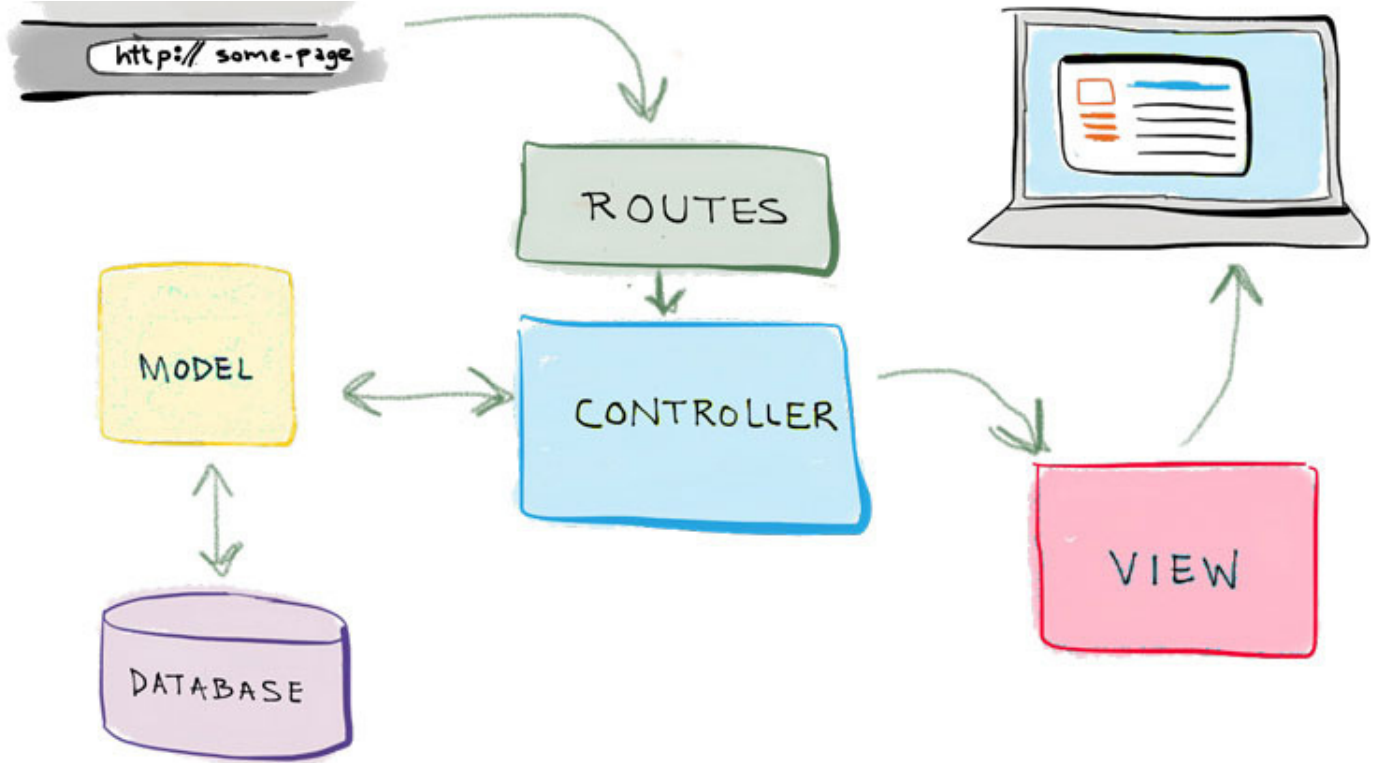


## MVC nedir?

MVC (*Model-View-Controller*), yazdığımız uygulamanın iş mantığı ile (business logic) kullanıcı arayüzünü birbirinden ayırıştıran, uygulamanın farklı amaçlara hizmet eden kısımlarının birbirine girmesini engelleyen yazılım mimarisidir.



## MODEL

Uygulamada kullanılan verileri temsil eder ve verilerin işleme **mantığının** saklandığı kısımdır. (Verilerin validasyonu burada yapılır)  
*Genelde* verilerin veritabanı (veya XML gibi benzer bir yere) kaydedilmesi ve kayıtlı yerden alınması işlemleri yine burada olabilir.  
Örneğin bir müşteri nesnesi, müşteri bilgilerini veritabanından alacak, işleyecek ve veriyi veritabanına geri güncelleyecek veya verileri işlemek için kullanacaktır.

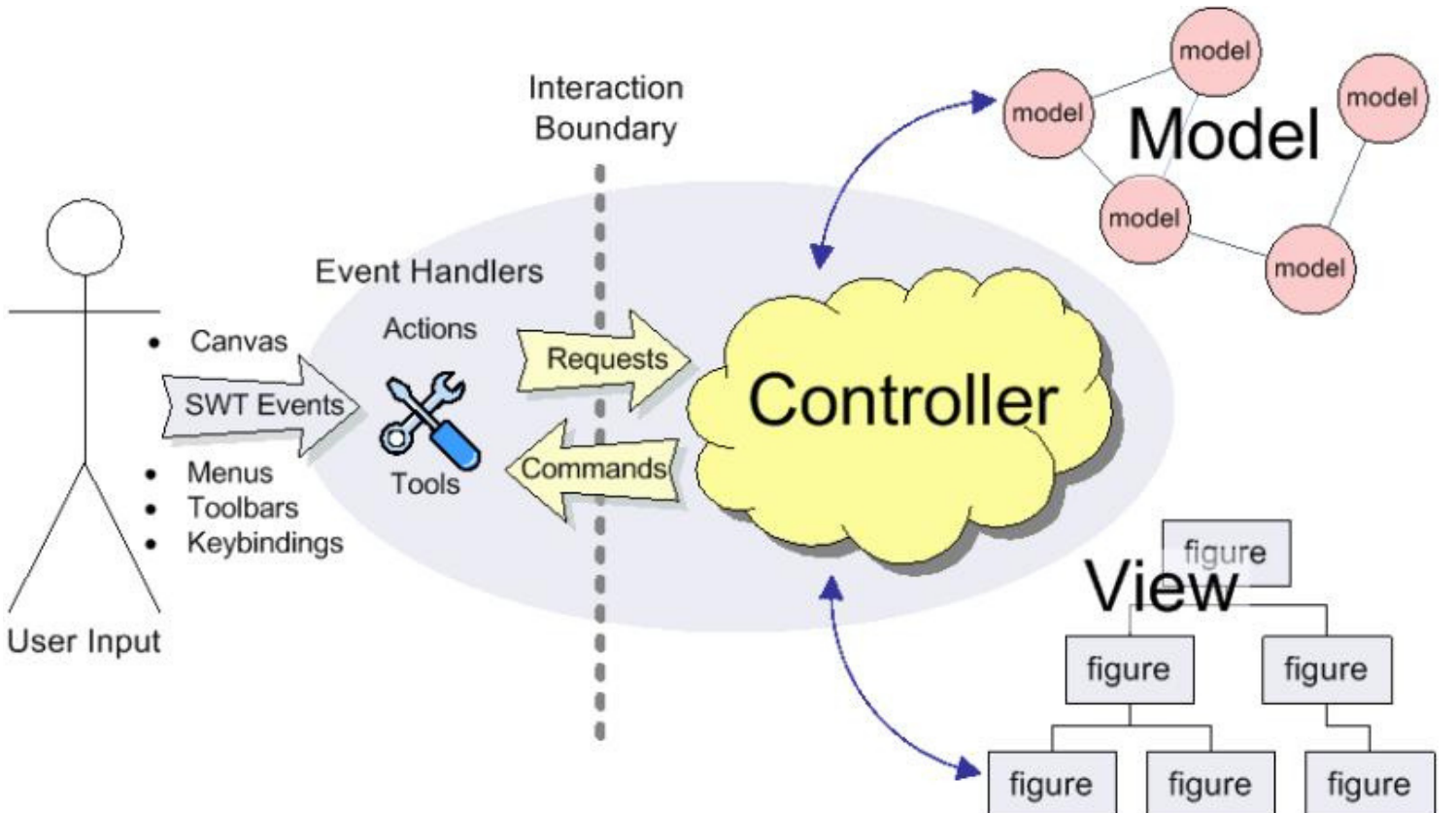
## VIEV

View bileşeni, uygulamanın tüm UI (User Interface - Kullanıcı Arayüzü) mantığı için kullanılır. Örneğin Müşteri görünümü, son kullanıcının etkileşimde bulunduğu metin kutuları, açılır menüler vb. Gibi tüm UI bileşenlerini içerecektir.

## CONTROLLER

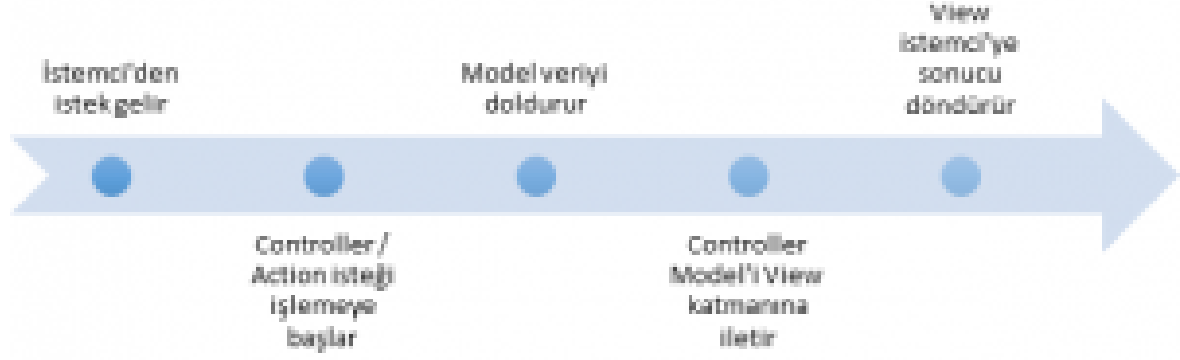
Controller'lar tüm iş mantığını ve gelen istekleri işlemek, Model bileşenini kullanarak verileri işlemek ve son çıktıyı oluşturmak için View'lar ile etkileşim kurmak için Model ve View bileşenleri arasında bir arayüz görevi görür.

Kullanıcıların View üzerinden gerçekleştirdiği işlemlerle alınan veriyi Model'e taşır, Model'den aldığı veriyi View üzerinden kullanıcıya gösterir.



## ROUTES

Routes (Routing) kısaca projenin URL yapısının kurguladığı yerdir. Anlaşılabilirlik ve SEO açısından uygun bir proje için buradan gerekli ayarlar yapılır, örnek vermek gerekirse "www.siteadi.com/yazidetay?id=58" gibi karmaşık bir kullanım yerine "www.siteadi.com/yazilar/mvc-nedir" şeklinde anlaşılır bir yönlendirme yapmayı sağlar.



### MVC'nin Avantajları Nelerdir?

**Proje Geliştirmeyi Hızlandırır:** Projede bulunan katmanlar birbirinden ayrı olduğu için eş zamanlı çalışma yapılabilir ve bu katmanlar kolayca birleştirilir.

**Düzenli, Anlaşılır ve Bağımsız:** En büyük avantajlarından biri de tasarım ile kodun birbirine karışmaması. .NET'te tasarım sayfasının arkasına (Code-Behind) kodlama işlemi yapıyorduk, MVC ile ise Modelleme yapılır ve Controller'a metotlar yazılır. Yani kodlar ile tasarım birbirinden bağımsız olur, böylece sayfa tasarımını değiştirmek gerektiğinde kod işleyişini değiştirmeye gerek kalmayacak. Vakitten kazanıyor, karmaşadan kurtuluyor ve az kod ile çok iş yapıyoruz.

**Performans:** .NET'te bulunan ViewState, durumları muhafaza etme mekanizması client-server arası transferi büyük boyutlara ulaştırıyor ve bu durum oldukça büyük performans kaybına yol açıyor, bu özellik MVC'de yerini ViewBag, TempData, ViewData taşıyıcılarına bırakmıştır. Bu sayede Controller - View arasında bulunan veri alışverişi daha basite indirgenmiş, oldukça hızlı bir yapıya kavuşturulmuştur.

**Test Etmek Kolaydır:** Katmanların birbirinden ayrı olması ve Test Driven Development işlemini kolaylaştıran yapısı ile test edilebilir bir proje geliştirilebilir.