------------------ uygulama oluşturma ------------------

Projemiz uygulamalardan oluşmaktadır. Uygulama oluşturarak devam edelim.

Öncelikle bir tane learning adında uygulama oluşturalım.

**python manage.py startapp learning**

Uygulama oluşturduğumuzda Django otomatik olarak bazı klasör ve dosyaları oluşturmaktadır. Şimdi bu dosyalar ne işe yaradığına kısaca bakalım. İleriki konularda ayrıntılarına gireceğiz. Şimdilik ne işe yaradıklarını bilmemiz yeterli.

**\_\_init\_\_.py :** dosyası bulunduğu klasörün bir PAYTON modülü olduğunu göstermektedir.

**admin.py :** uygulama için django admin panelini özelleştirmeye yarayan dosyadır.

**apps.py :** uygulamanın configürasyon ayarları için kullanılan dosyadır.

**migrations/ :** models.py de yer alan veri yapılarımızın veritabanına taşınmasını sağlayan dosyadır.

**models.py :** veri yapılarımızın tanımlandığı dosyadır. Veritabanında kullanmak istediğimiz veri yapılarını bu dosyalar içerisinde tutarız.

**tests.py :** birim testlerinin yazıldığı dosyadır. Testler kodumuzun çalıştığını kontrol ederler. Bu sayede her değişiklikte manuel test yapmamıza gerek kalmaz.

**views.py :** Tarayıcıdan bir adrese girildiğinde projenin urls.py dosyasından bu dosyanın içerisindeki methodlara istek gönderilir. Bu methodların içerisindeki kodlar çalışarak tarayıcıya bir çıktı response edilir. Daha sonra tarayıcı o çıktıyı ekrana basar.

Dosyalar ile ilgili temel bilgiler bu şekilde olup, uygulama aşamasına geçildiğinde daha iyi anlaşılacaktır.

Bu dosyaları ve klasörleri kullanmak zorunda değilsiniz. Ben büyük projeler için daha kolay okunabilirlik olması nedeniyle view ve modeller için ayrı klasörler kullanıyorum. İki kullanım içinde örnekler yapacağım. Öncelikle klasörler kullanarak oluşturulan yapıyı inceleyelim.

Eğer uygulamalarınız büyükse views.py ve models.py dosyalarının içerisinde çok fazla kod oluşmakta

bu da dosyada yapılan düzenlemeleri ve okunabilirliği zorlaştırmaktadır.

Uygulamada 10 tane model ve her modelinde yaklaşık 30 satır olduğunu düşünürsek models.py dosyası

300 satırdan oluşacaktır. Bu da dosya içerisinde istediğimiz modeli bulmayı ve düzenlemeyi zorlaştıracaktır.

Ben bunun yerine models ve views klasör halinde tutup içerisine her model ve view için dosya oluşturuyorum. Models/ klasörüne baktığımızda product ve product\_detail için iki ayrı dosya oluşturdum ve içerisinde ilgili veri yapılarımı ekledim.

Bu videomuzda uygulama oluşturduk ve uygulama dosyalarını inceledik.

Bir sonraki videomuzda nasıl model oluşturacağımızı öğreneceğiz ve model özelliklerini inceleyeceğiz.

Bu videomuzda Django kurulumunu gerçekleştirdik ve proje ve uygulamanın temellerine değindik.