------------------ proje oluşturma ------------------

Evet şimdi oluşturduğumuz projeyi PAYÇARM içine alalım. Pycharm -> Open -> Django Proje mizi seçiyoruz. Burada dikkat edelim Proje klasörünü değil (Project deği) Django Projesini (stocks) seçmemiz gerekmektedir.

Şimdi PAYÇARM için virtualenv ayarlarını yapmamız gerekiyor.Settings -> Project: stocks -> Project Interpreter den ayar simgesine tıklıyoruz. Proje klasörünün içindeki Scripts içerisinden python.exe seçiyoruz. Gördüğünüz üzere kurulu paketlerimiz buraya geldi. Daha önce terminalden Django projemizi oluşturmuştuk. Proje yapısına baktığımızda içerisine bir adet stocks adında klasör ve manage.py dosyası eklendi.

manage.py dosyası terminal ekranından komutlarımızı çalıştırmaya yarayacaktır. Bu dosya ile herhangi bir işlem yapmayacağız.

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

settings.py Django ayarlarının bulunduğu dosyadır. Bunun içerisinden genel Django ayarlarını yapıyoruz.

urls.py dosyasından browserden istek yapıldığında hangin kodun çalıştırılması gerekiyorsa oraya yönlendirme yapıyoruz.

wsgi.py – sunucumuzun ayarlarının yapıldığı dosyadır.

Şimdi web sunucumuz çalışıyormu kontrol edelim. Bunun için manage.py ile runserver komutunu çalıştırıyoruz.Django otomatik olarak kendini 8000 portunda çalıştırıyor.

**python manage.py runserver**

ctrl + c yaparak sunucuyu durdurabiliriz.

Eğer 8000 portunda başka bir şey çalıştırıyorsak portu elle vererek başka portta da çalıştırabiliriz.

**python manage.py runserver 8081**

Açılan karşılama sayfası gördüğünüz gibi İngilizce. Biraz önce Django ayarlarını settings.py den ayarlayabildiğimizi söylemiştik. Şimdi settings.py dosyasını biraz inceleyelim.

Burada **DEBUG = True** görüyoruz.

Bu sistem hataların ekrana basılmasını sağlamaktadır. Bu sayede hatalarımızı görebilir ve gerekli yerleri düzenleyebiliriz. Bu development yani geliştirme süreci bittiğinde False olarak değiştirilmelidir. Aksi taktirde güvenlik açığı oluşturacaktır.

**INSTALLED\_APPS** da hali hazırda kurulan uygulamalar gösterilmektedir. Biz paket indirdiğimizde veya uygulama yazdığımızda

Aktifleştirmek için bu alana eklememiz gerekmektedir.

**MIDDLEWARE**, ara katman anlamına gelmektedir. Ne işe yarar dersek gelen HTTP isteklerini filtrelemeye yarar. Örnek vermek gerekirse;

Tarayıcıdan bir url girmeye çalıştığınızda önce Middleware devreye girer kullanıcının giriş yapıp yapmadığına bakar. Eğer kullanıcı girişi başarısızsa giriş sayfasına yönlendirir, başarılı ise devam etmesine izin verir.

Kullanıcı bir form gönderdi. Formdaki alanları kontrol eder. Formdan gelen güvenlik ihlali sağlayacak veri varsa onları filtreler ve kalan veriyi işlemesi için bir sonraki aşamaya gönderir.

**ROOT\_URLCONF**  Django ya istek geldiğinde url maping için hang dosyanın kullanılacağını ayarlamaya yarar.

**TEMPLATES** : Template motoru ayar değişikliklerinin yapıldığı bölüm

**DATABASES** : Veritabanı ayar değişikliklerinin yapıldığı bölüm. Django birden fazla veritabanı ile işlem yapmamıza izin vermektedir.

**AUTH\_PASSWORD\_VALIDATORS** : Kullanılacak kullanıcı şifreleri için gerekli olan validasyon ayarları

**LANGUAGE\_CODE**: Django dil ayarı tr yaparak türkçeleştirebiliriz.

**TIME\_ZONE** : Zaman ayarı. Europe/Istanbul yaparak türkiye saatini ayarlayabiliriz.

**USE\_I18N, USE\_L10N, USE\_TZ** : Localizasyon ayarlarıdır.

**STATIC\_URL**: statik dosyaların konunlandıracağı yeri gösterir.