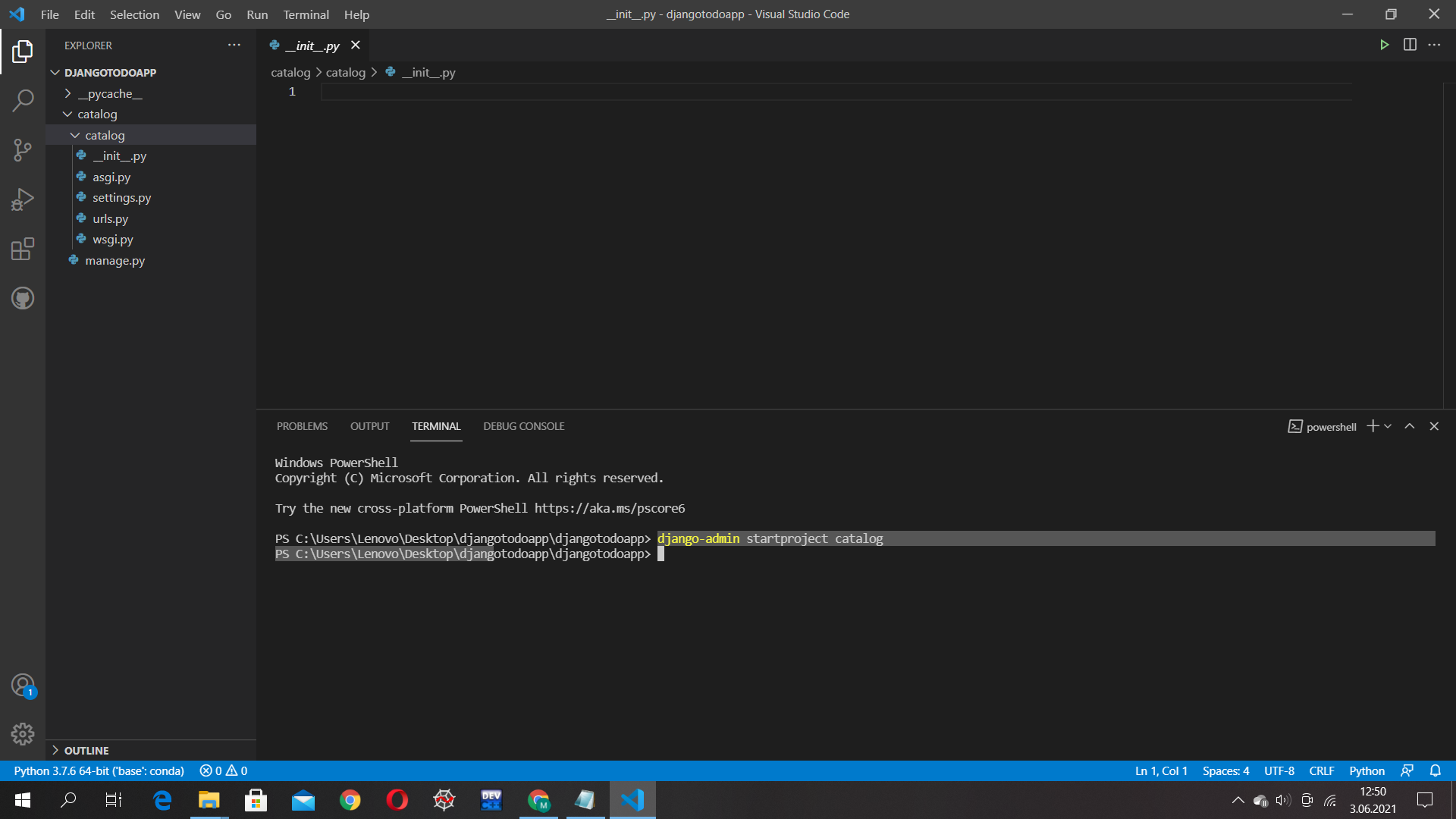
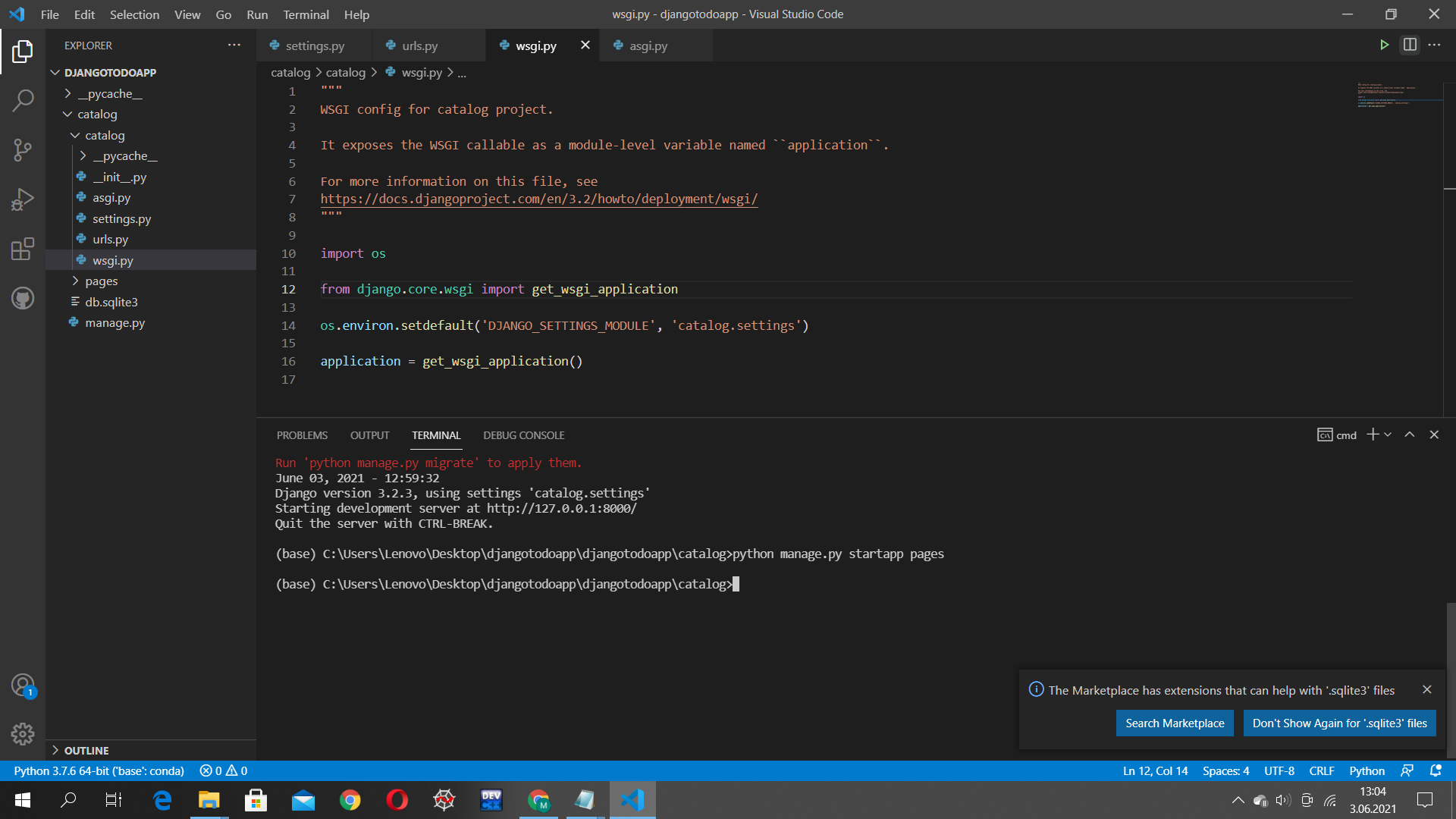
PROJE FİNAL DJANGO

MUSTAFA VAROL 190901001 WEB TEKNOLOJİLERİ

Catalog adlı proje oluşturmak icin kod

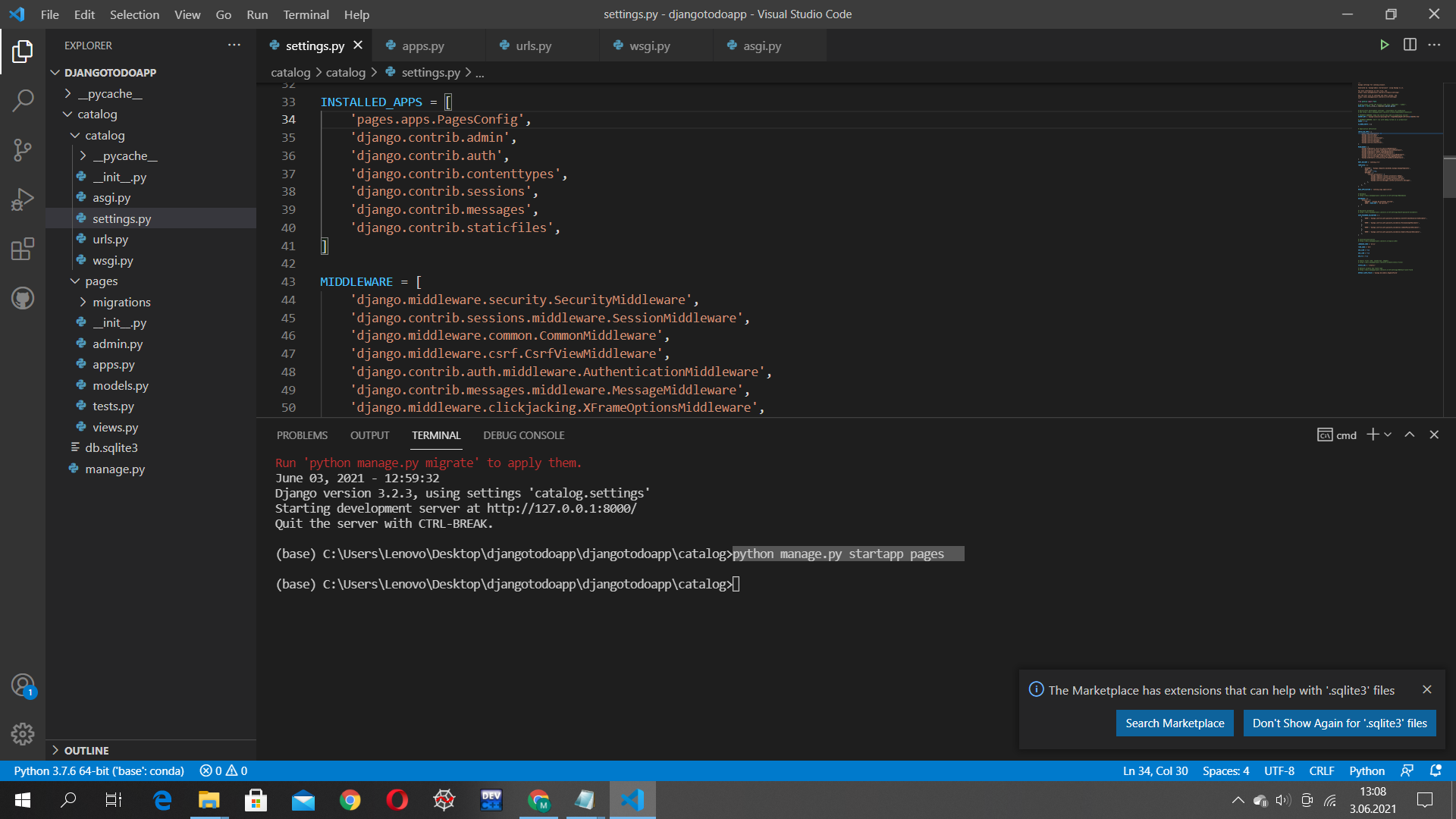


django-admin startproject catalog ile catolog adlı proje oluşturulmuş olur

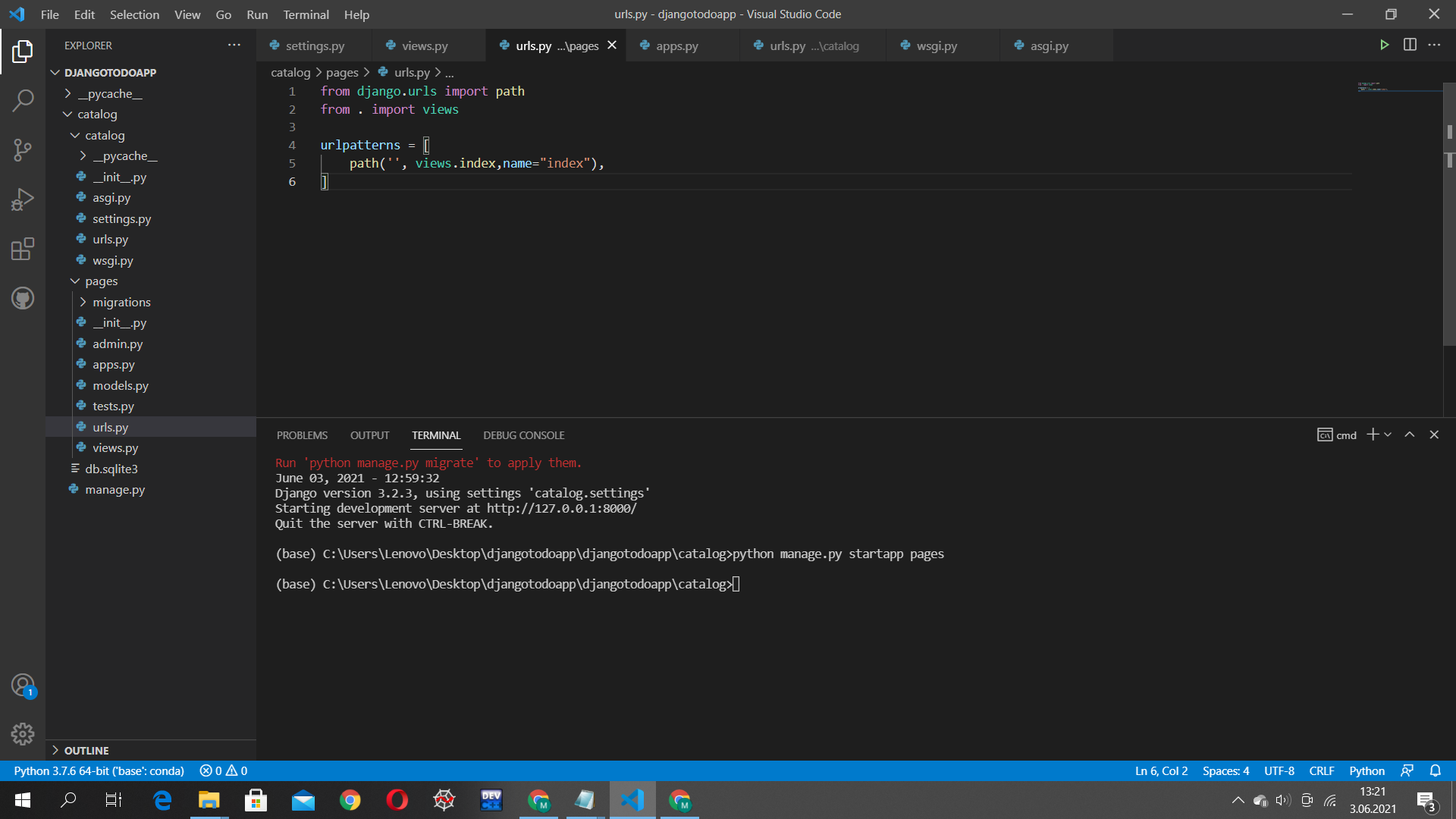


Esas katolog projesi altına pages adında bir

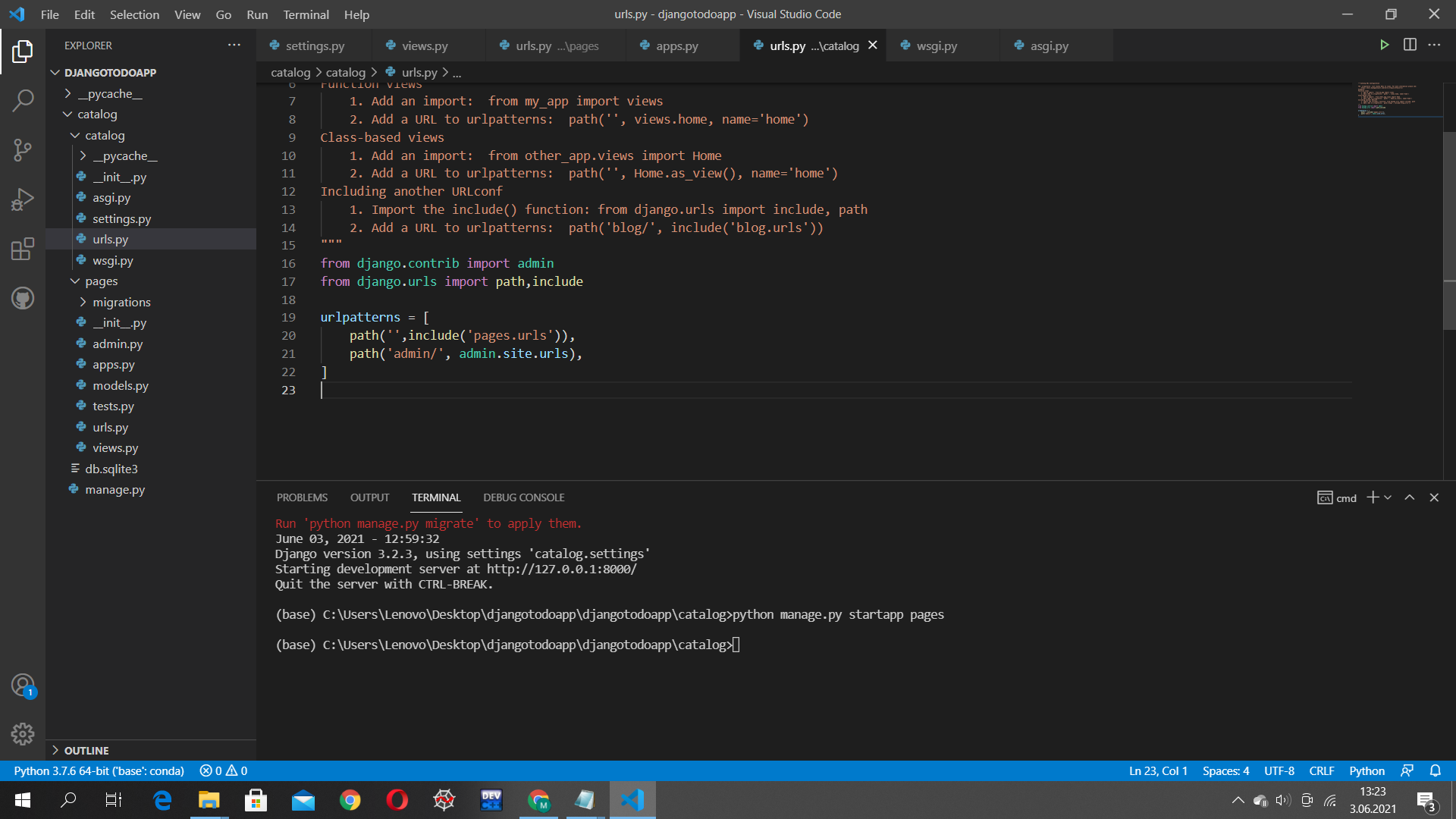
Proje daha ekledik



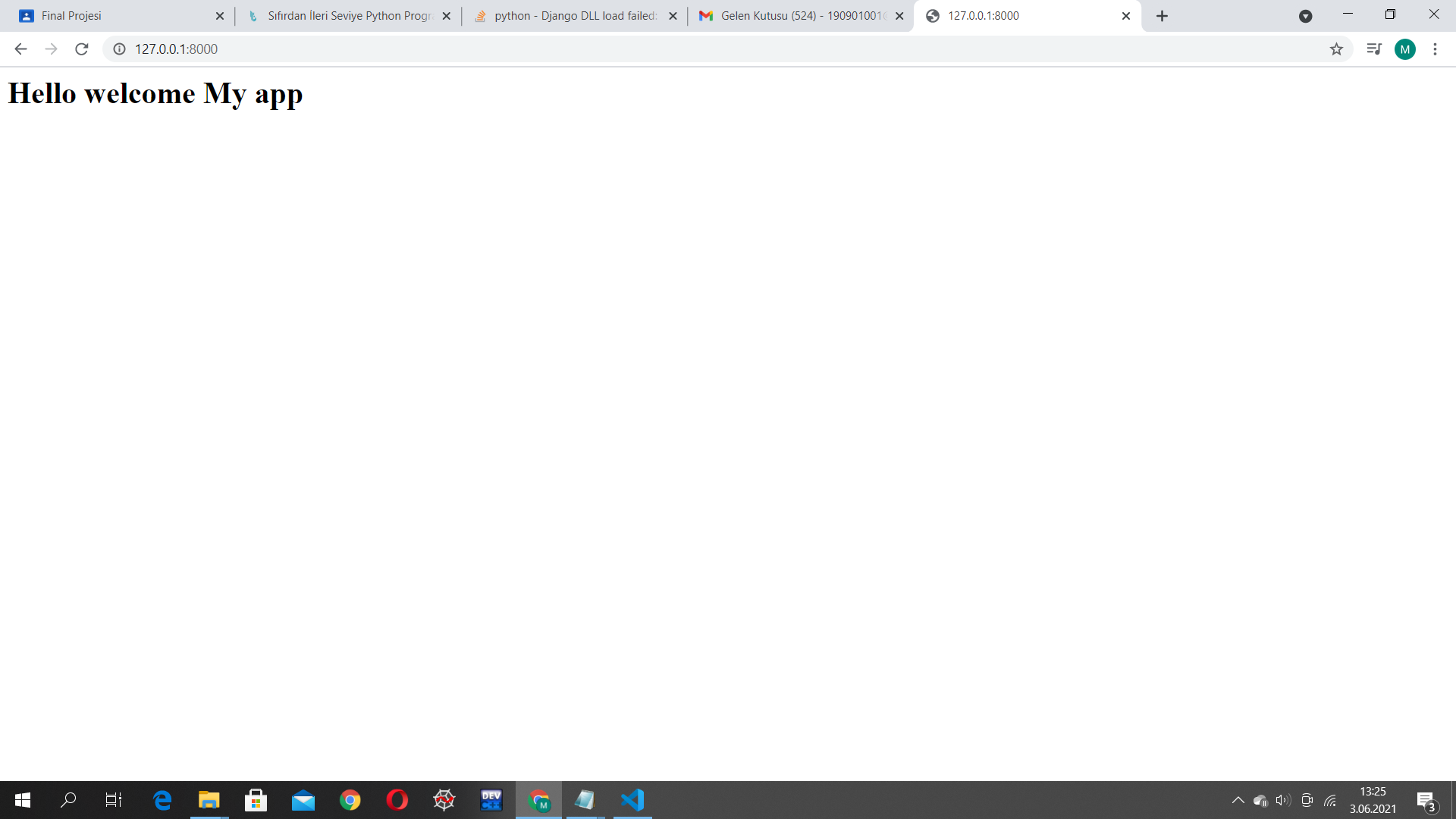
Settiing.py de installed apps kısmına eklenen projelerin confihleri eklenir şkeilde pagesConfig eklenmiştir



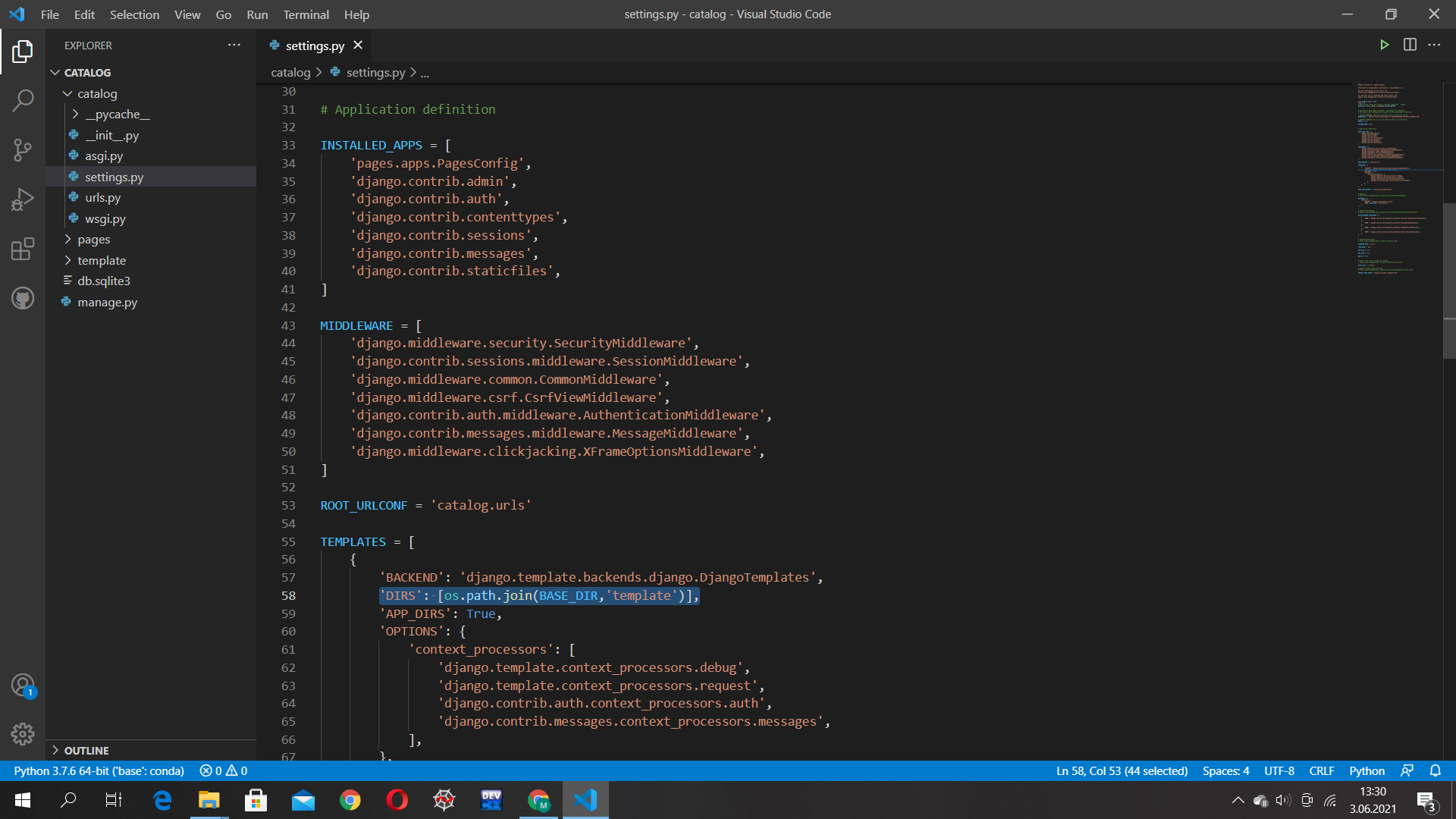
Pages in urls.py sine grekli pathler burda tanıtılır sekilde index eklenmiştir



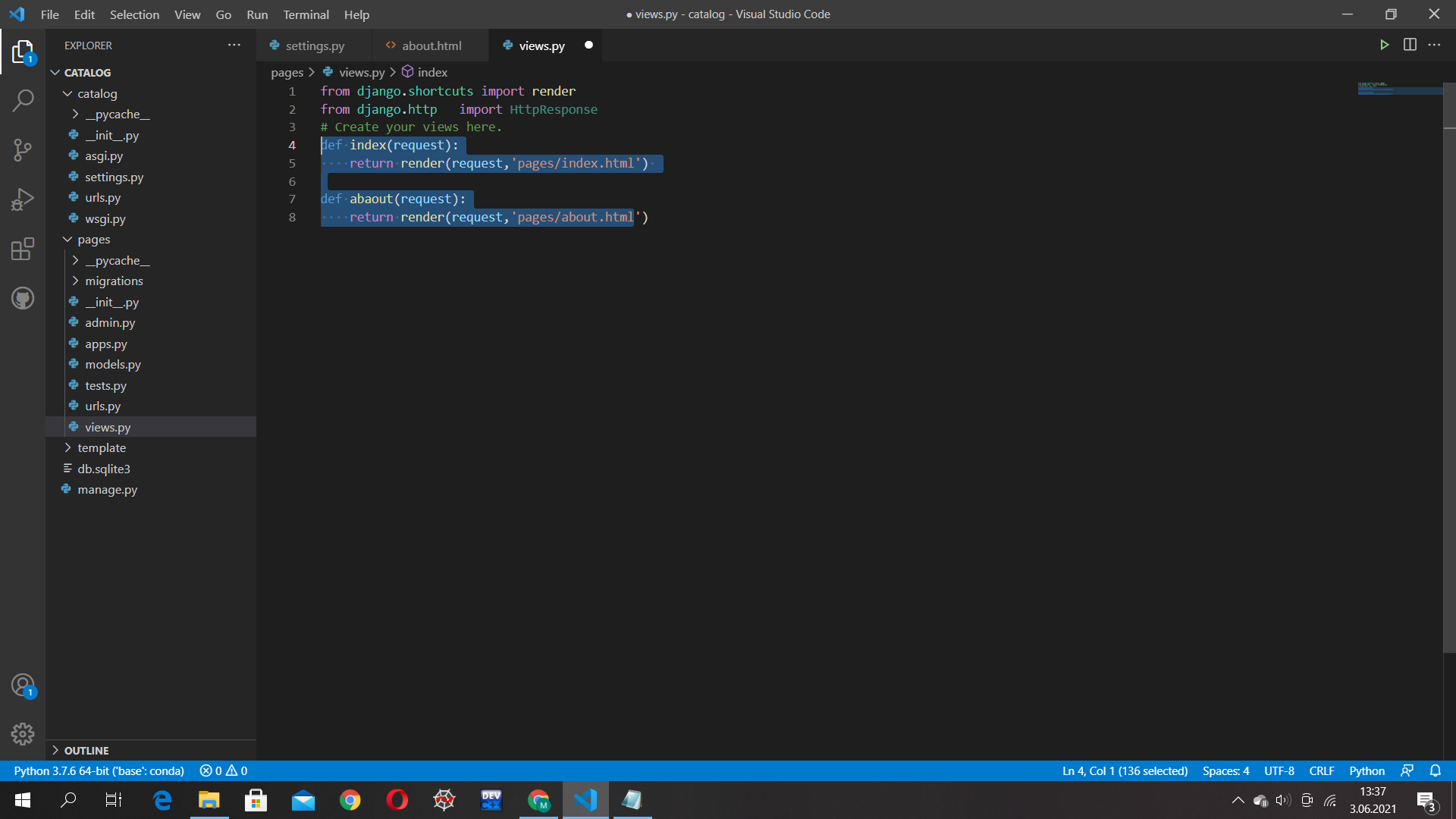
Pages urller katolog anadizindeki urls.py de pathlere tanıtılır



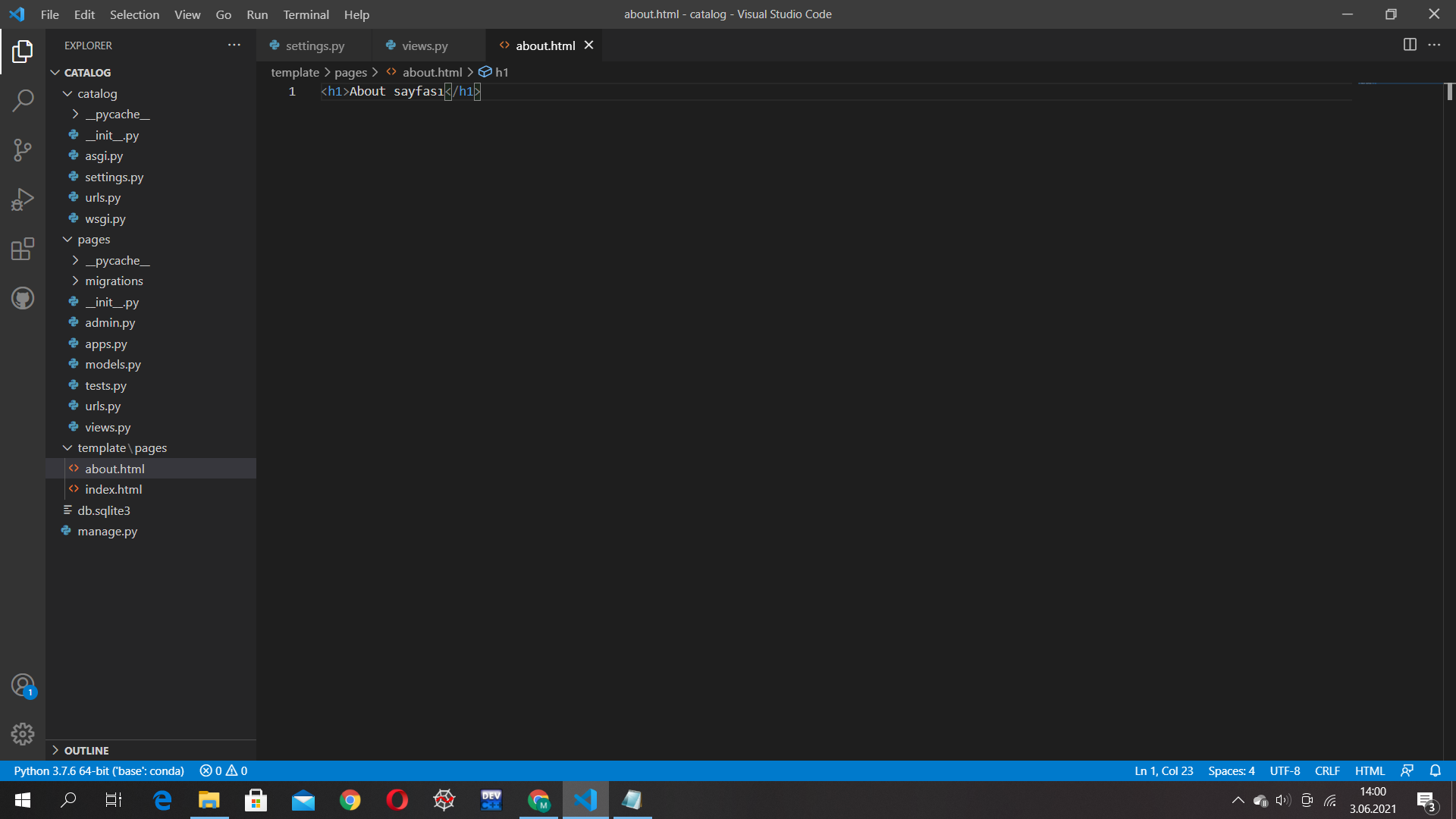
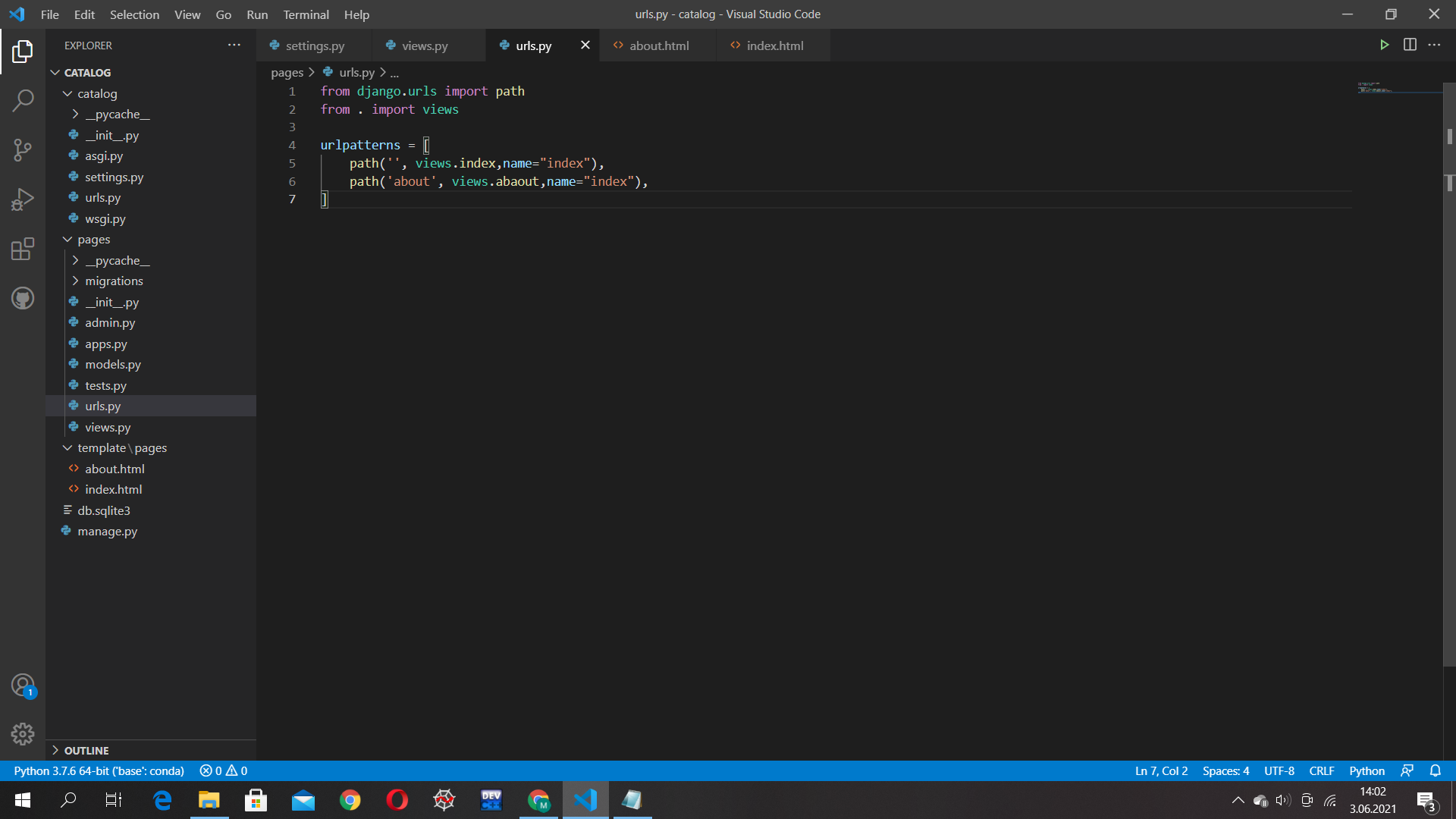
İlk uygulama görüntüsü



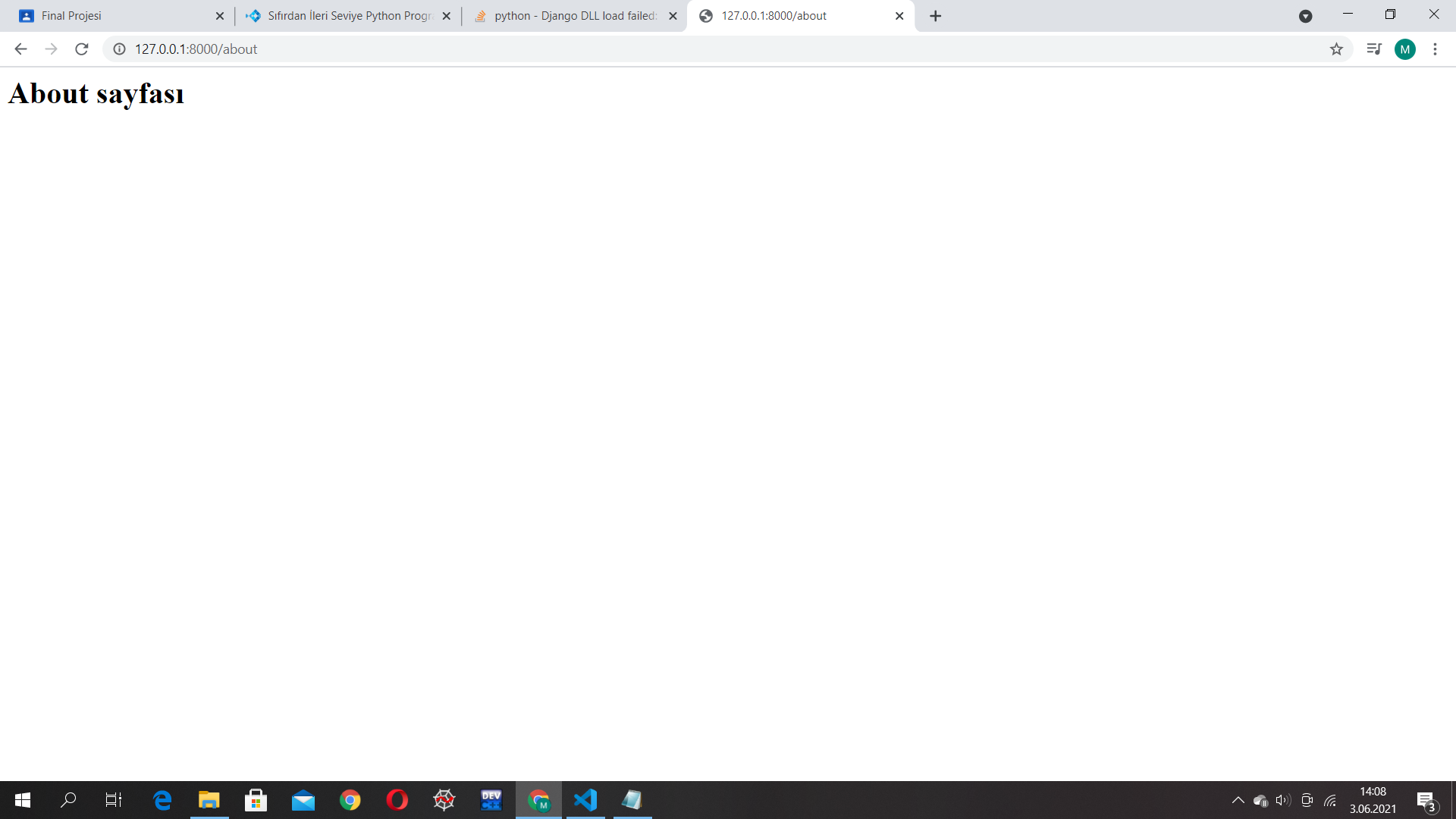
Settings.py de template klasörü tanıtılır templates içinde



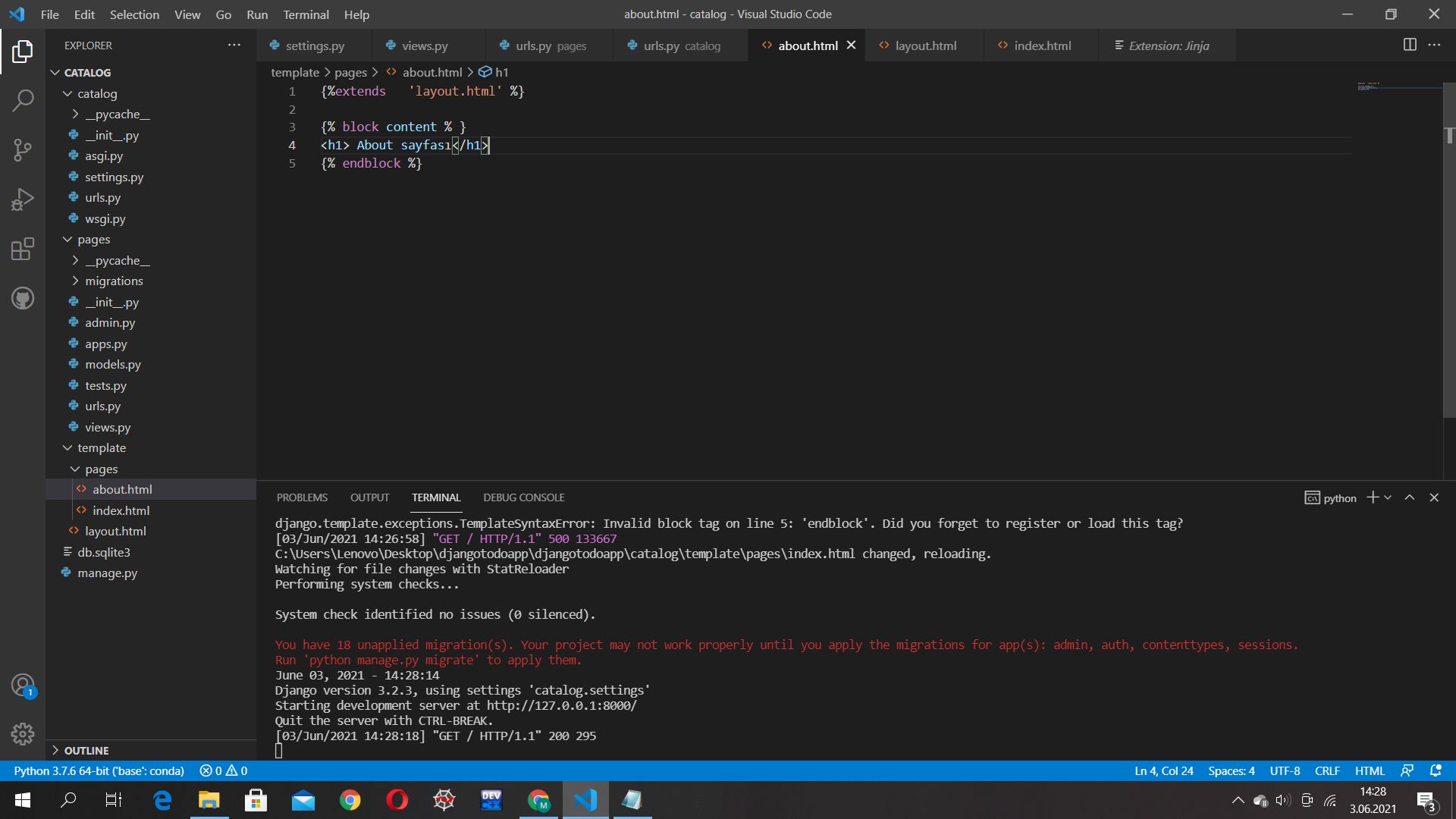
Kim için yaızlıyorsa htmlller onun viewsine eklenmelidir resimde pages için 2 tane html viewsa eklenmiştir.İndex sayfası ve about sayfası viewsa eklenerek index yazıldığında gidilecek sayfa belirtilmiştir



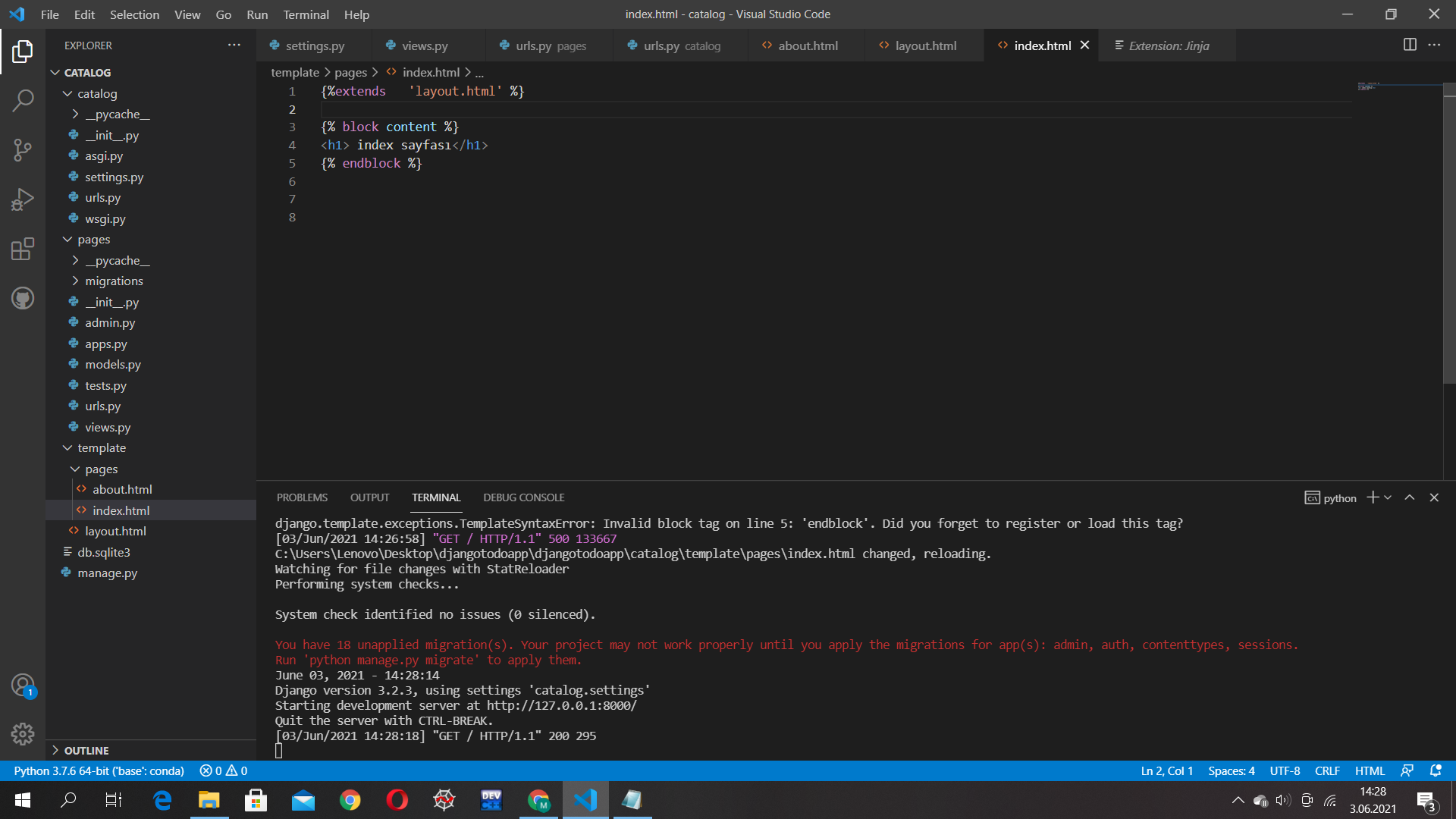
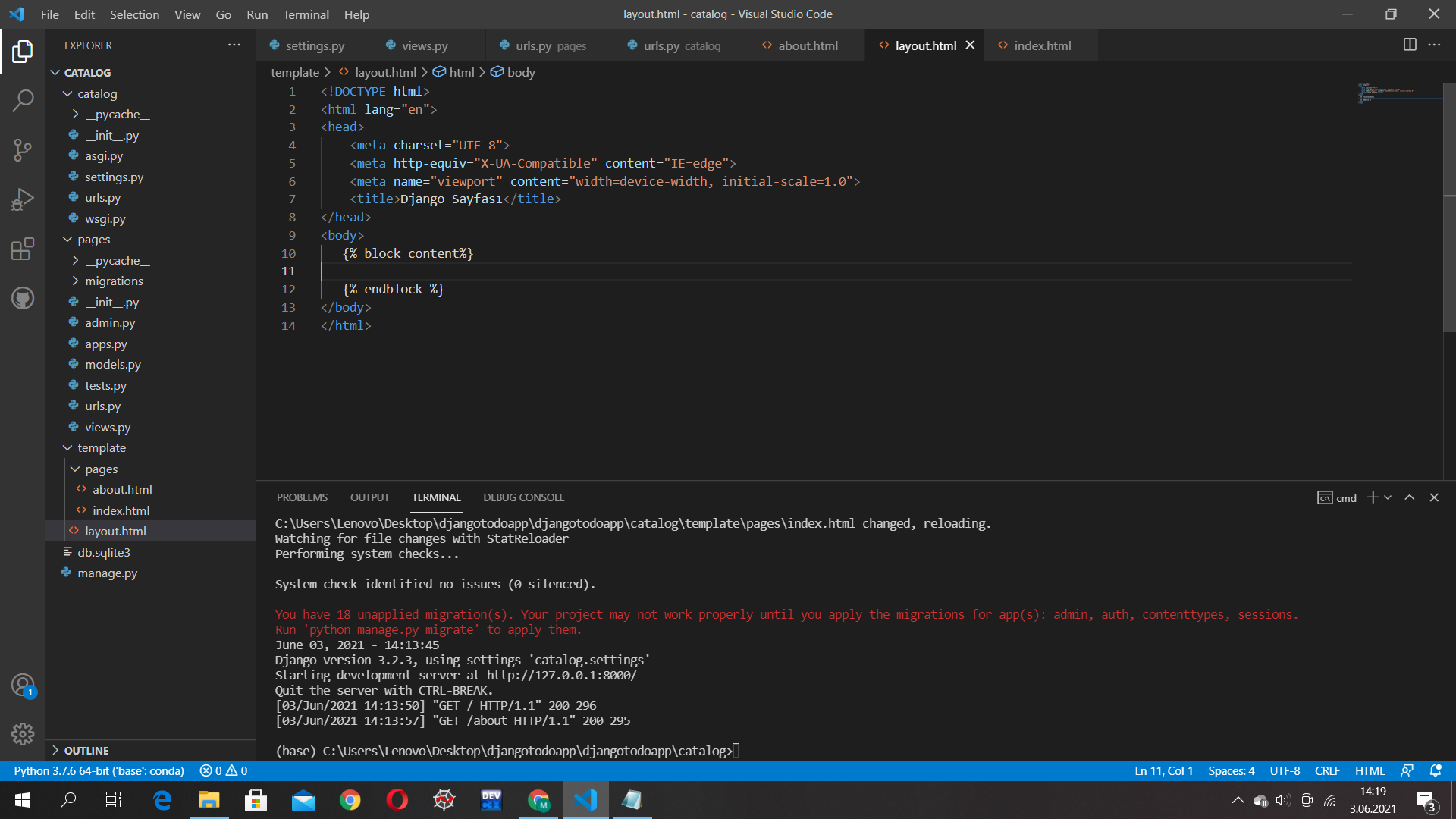
İndex ve about sayfaları deneme amaçlı kısa şeyler atılmıştır ve bunlar pagesin urls.pysinde eklenmiştir



Deneme sonuçları bu şekildedir.Görüldüğü gibi adres satırrına localhosttan sonra about yazılmıştır

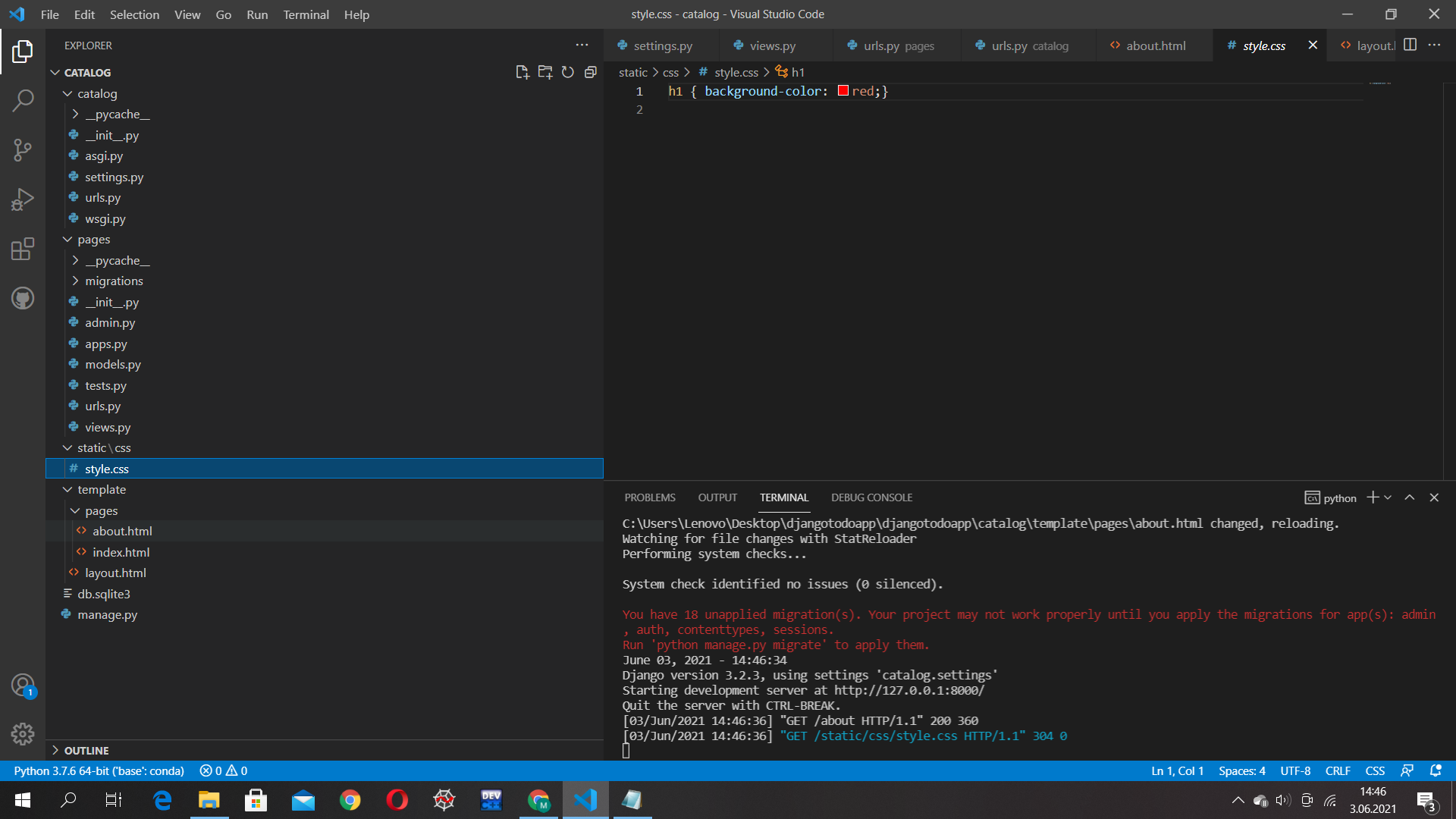


Bir layout.html sayfası yazılarak fazla koddan kurtulumuş olur . Cünkü layout sayfasında bir rende body oluşturulur ve buraya {%block content%} ler yazılan yerlerin içine ne gönderilirse onu alır



Burda gösterildiği gibi extends layout ile layout.html sayfasından türeyen index ve about htmller

Layout.html üzerinden kısa sekilde gönderim yaparlar . Sadece body kısmı değişir

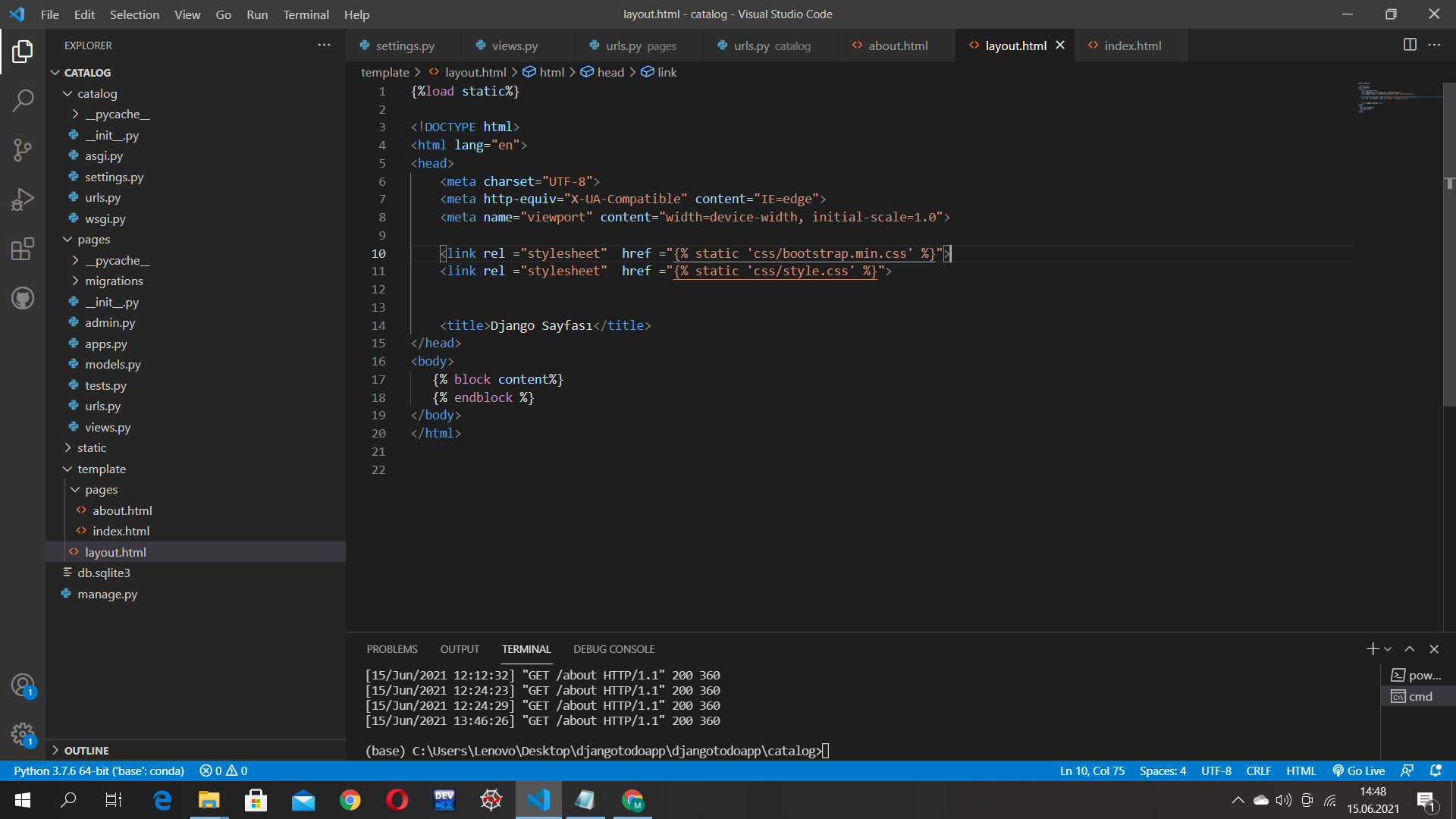


Static dosyası altına css dosyası ve js dosyası yazılır

Resimde css altına bir style.css dosyası deneme amaçlı oluşturuldu.

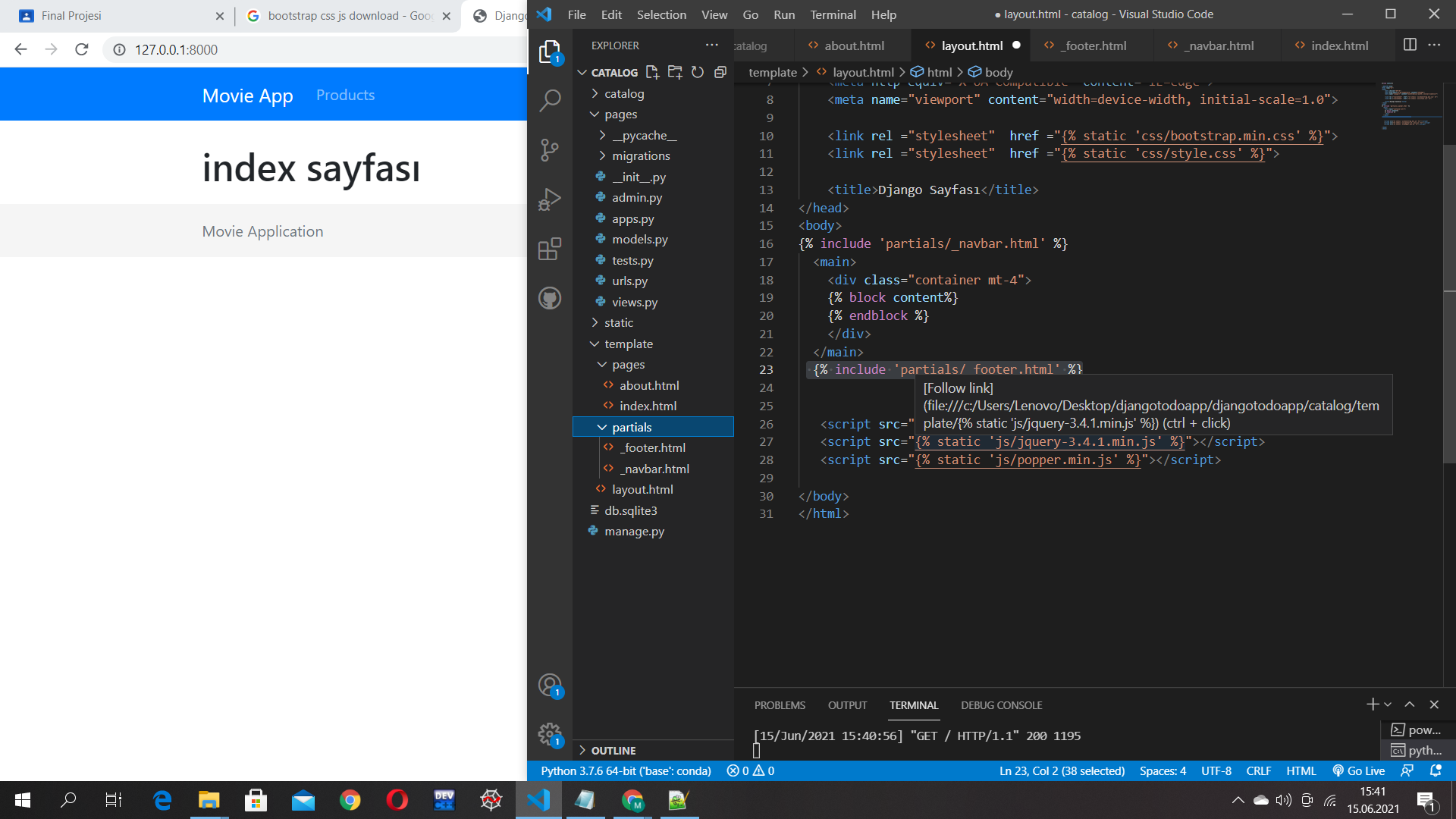


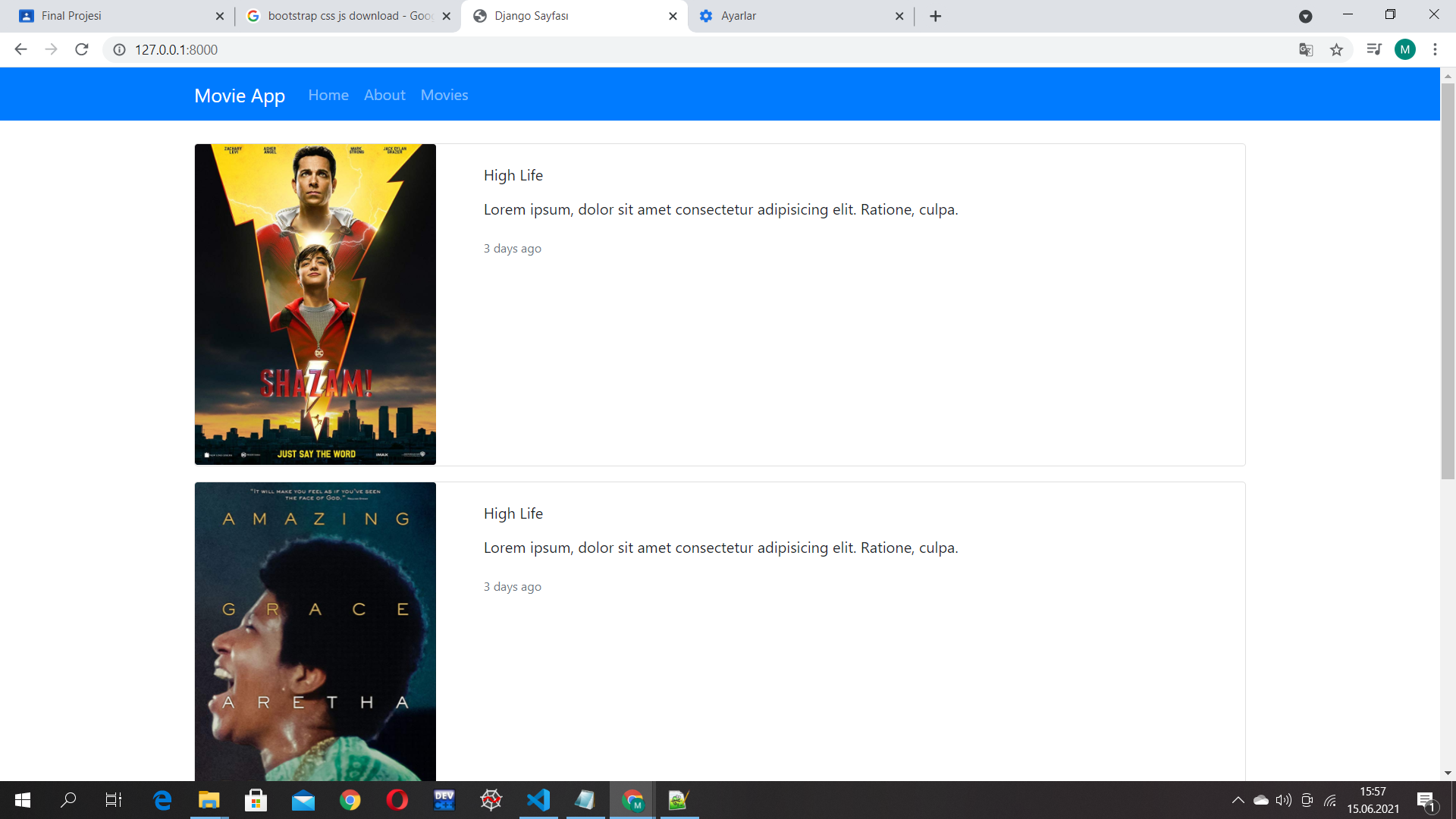
Css in calıştııgı network kısmındada test edildi.Layout.html kısmına yazılan css indeeex ve about dada calışıtğı gösterildi.



Layout.html le static kısmını eklenmesi ayrıca settings kısmına satic dir files kısmına ekleme yapılır

Bootsrap elemanları css js ve img ler eklendiktene sonra,arrtials altına navbar ve footer html sayfaları yapıldı. Kodlar bu şekilde daha az yer kaplama ve daha esnek oldular. Bu partial vieeewlarr layout htmlde {% include 'partials/\_footer.html' %} eklenir.

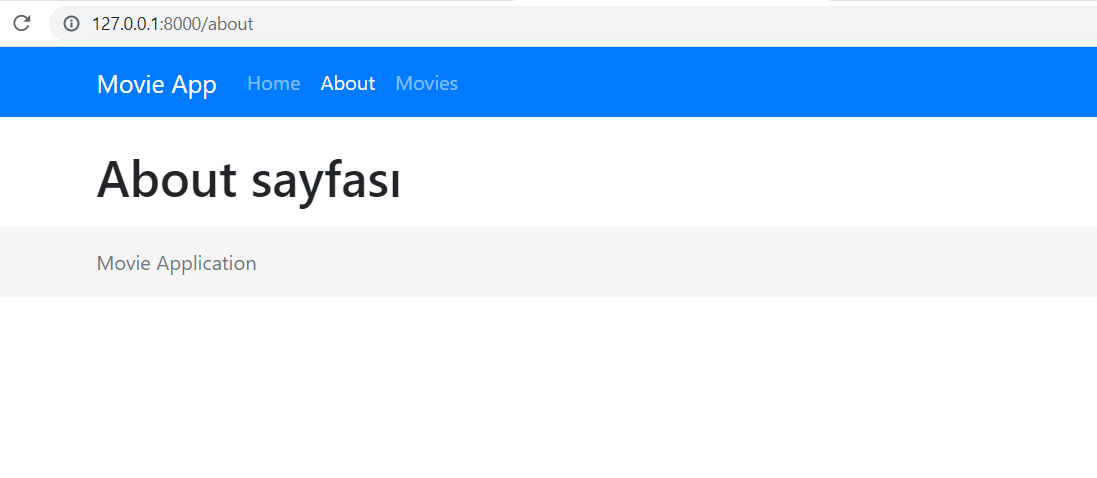




Filmlerin eklenmesi

Gerekli ortalamalar v.s ler bootsrap div classla içine yapıldı.Kod kütüphanesinde yazıldı

Movie projesi oluşturuldu urlleri pages gibi eklendi. İnex ,about ve detail htmller oluşturuldu.



Tıklanma yerleri aktif olmas için native-active div classlarrına eklendi

Ekstra python sorgusuyla hangi request gelmişsser onun aktif edilmesi sağlanıyor.

{% if 'about' in request.path %}

class="nav-item active"

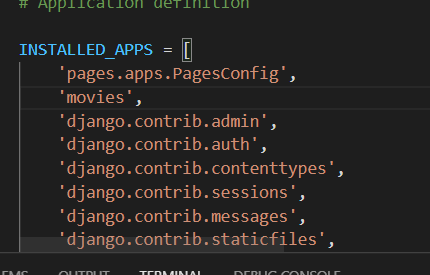
{% else %}

class="nav-item"

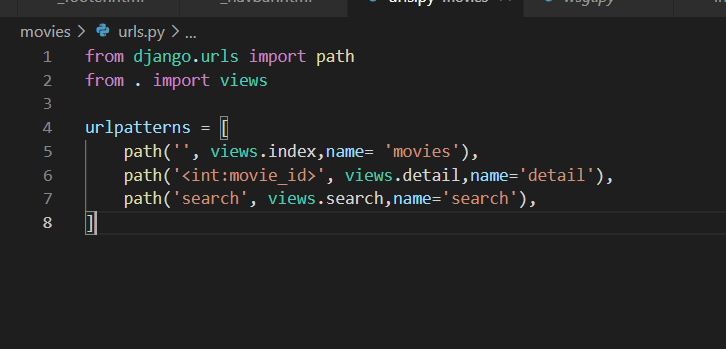
{% endif %}

>

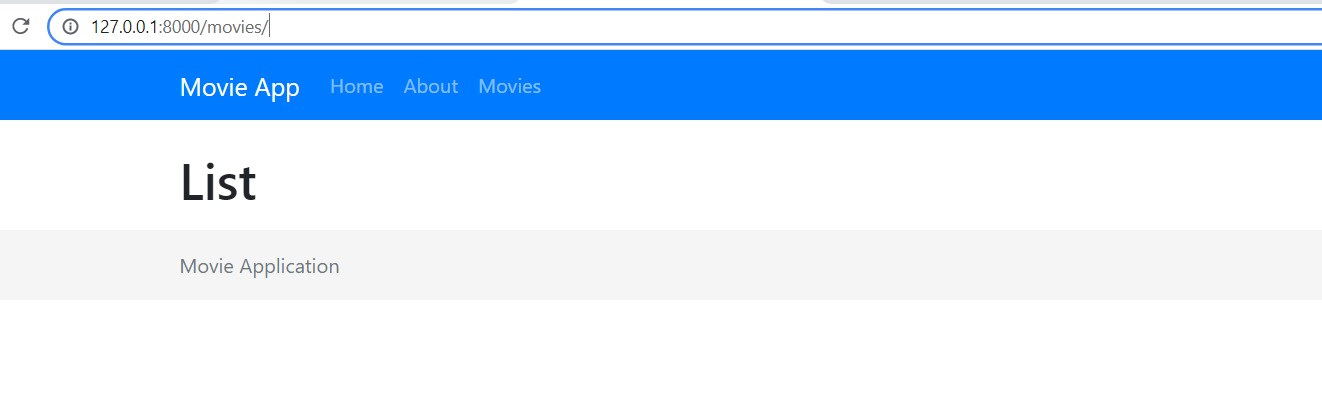
<a href="{% url 'about' %}" class="nav-link">About</a> bu sekilde li etiketlerinin içine atılır



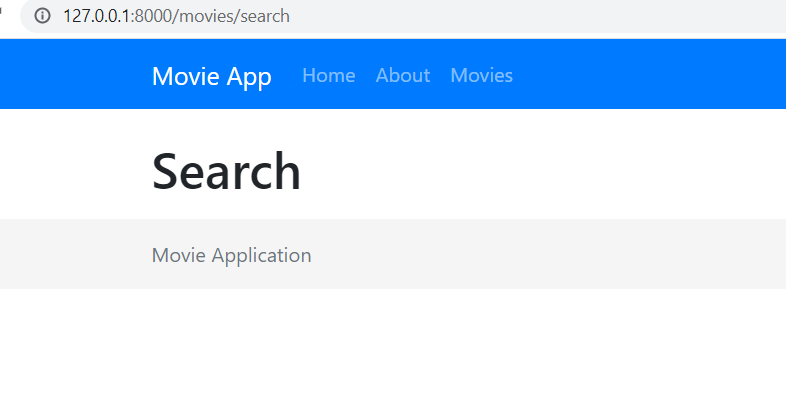
Setting.py kısmında sadece movies yazarakda config eklenebilir.



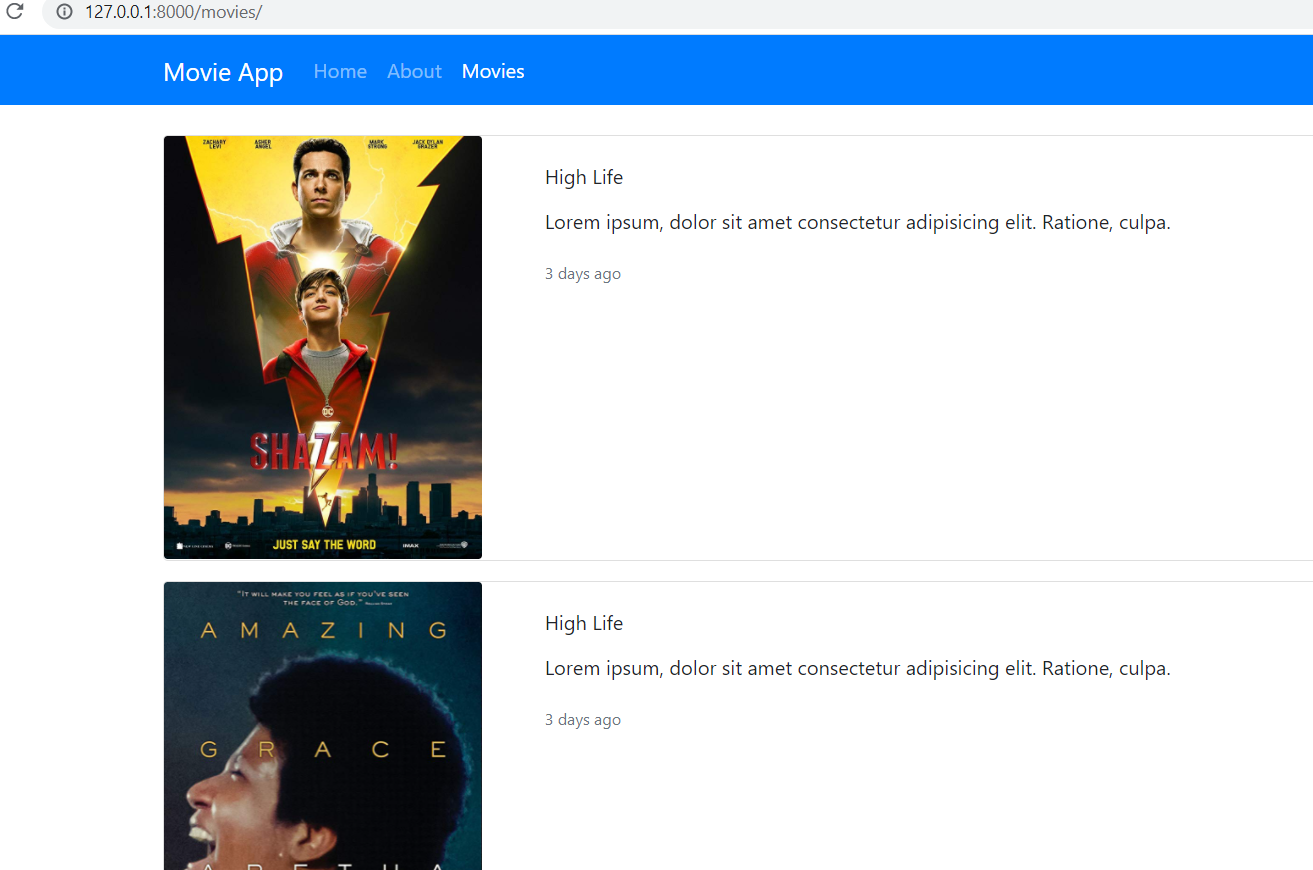
Moviesinde urlleri verilir



Movies index sayfası görünümü



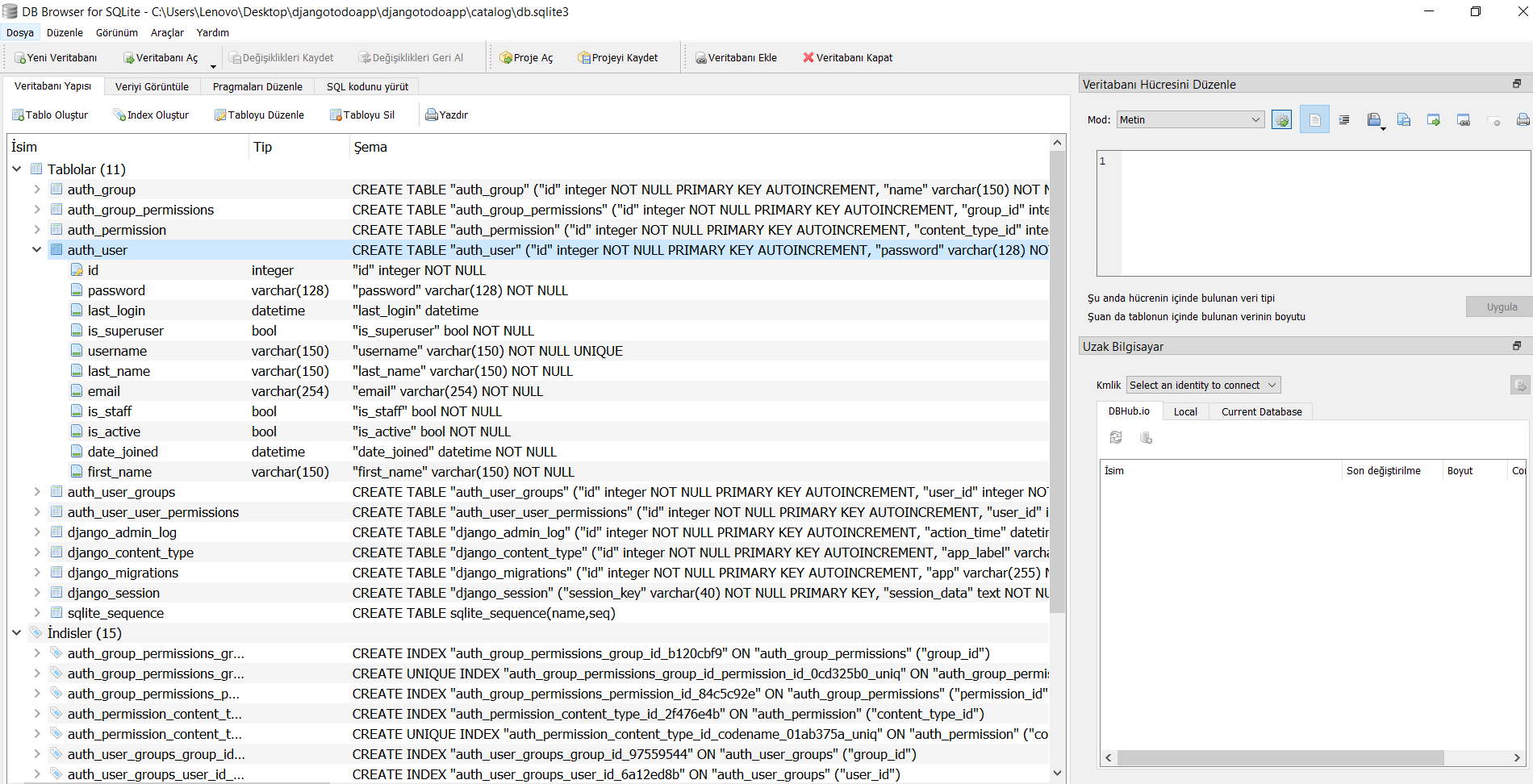
Moviesin search sayfası görüldüğü gibi adres staırına önce movies/search yazılır



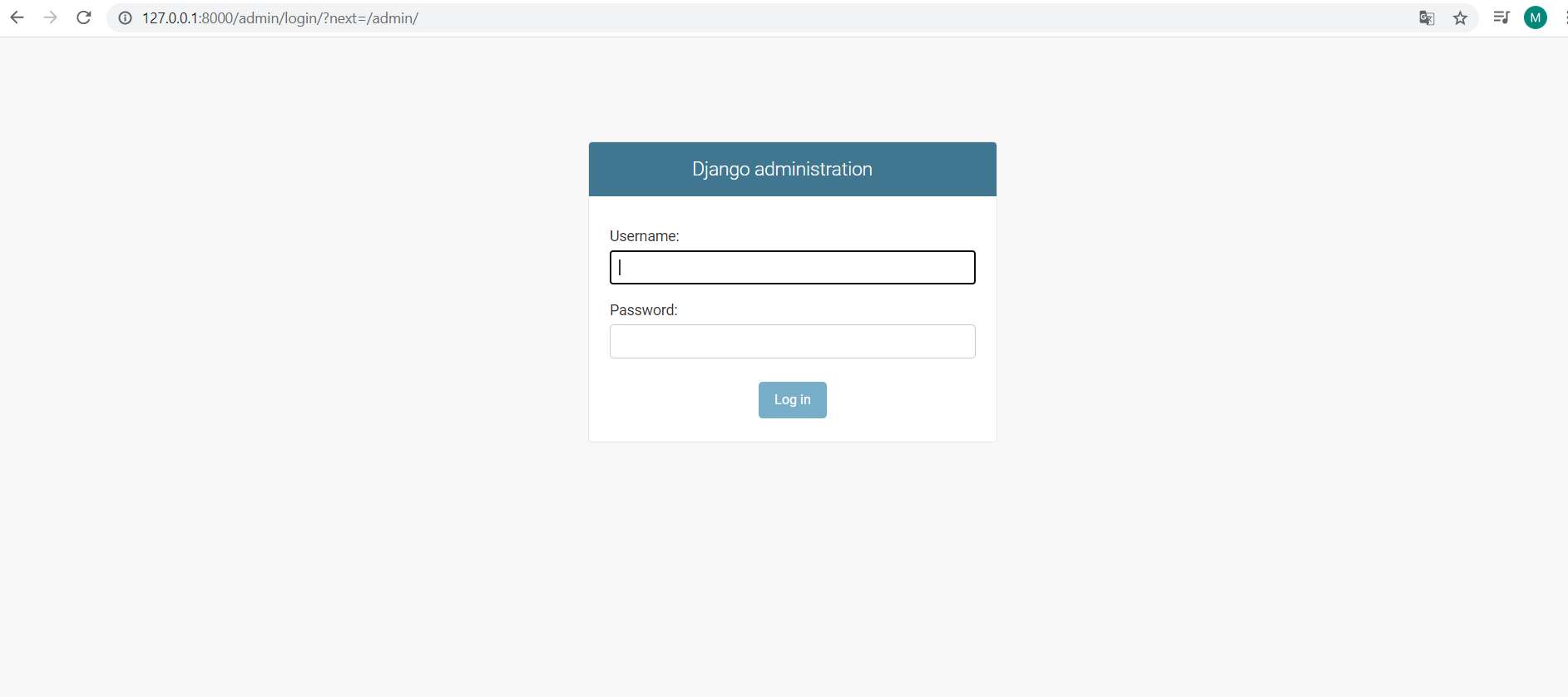
Movies kısmına filmler eklenir

Burada eklemede manuel kullanım vardır

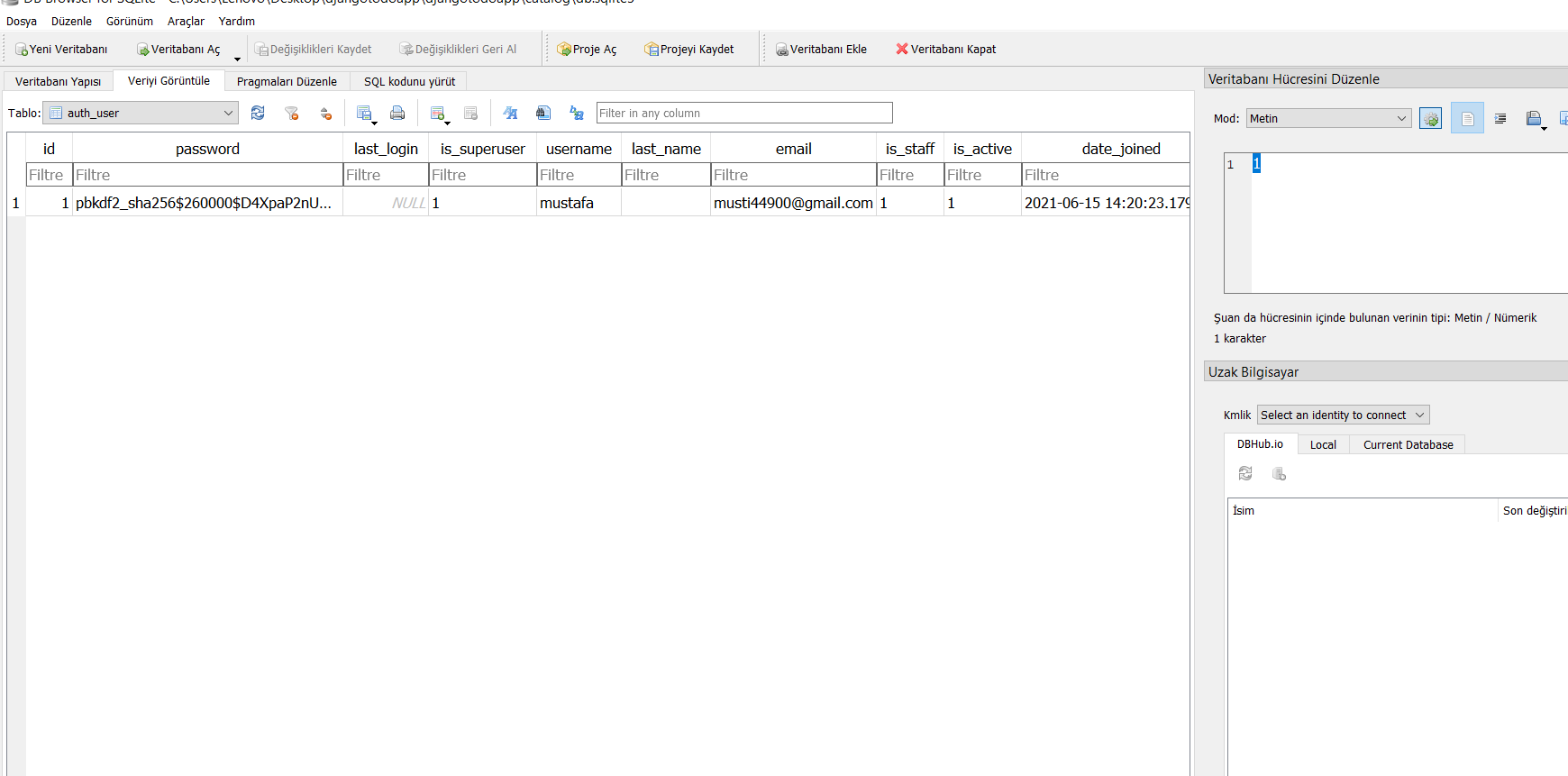
python manage.py migrate denilerek veritabanına aktarılır



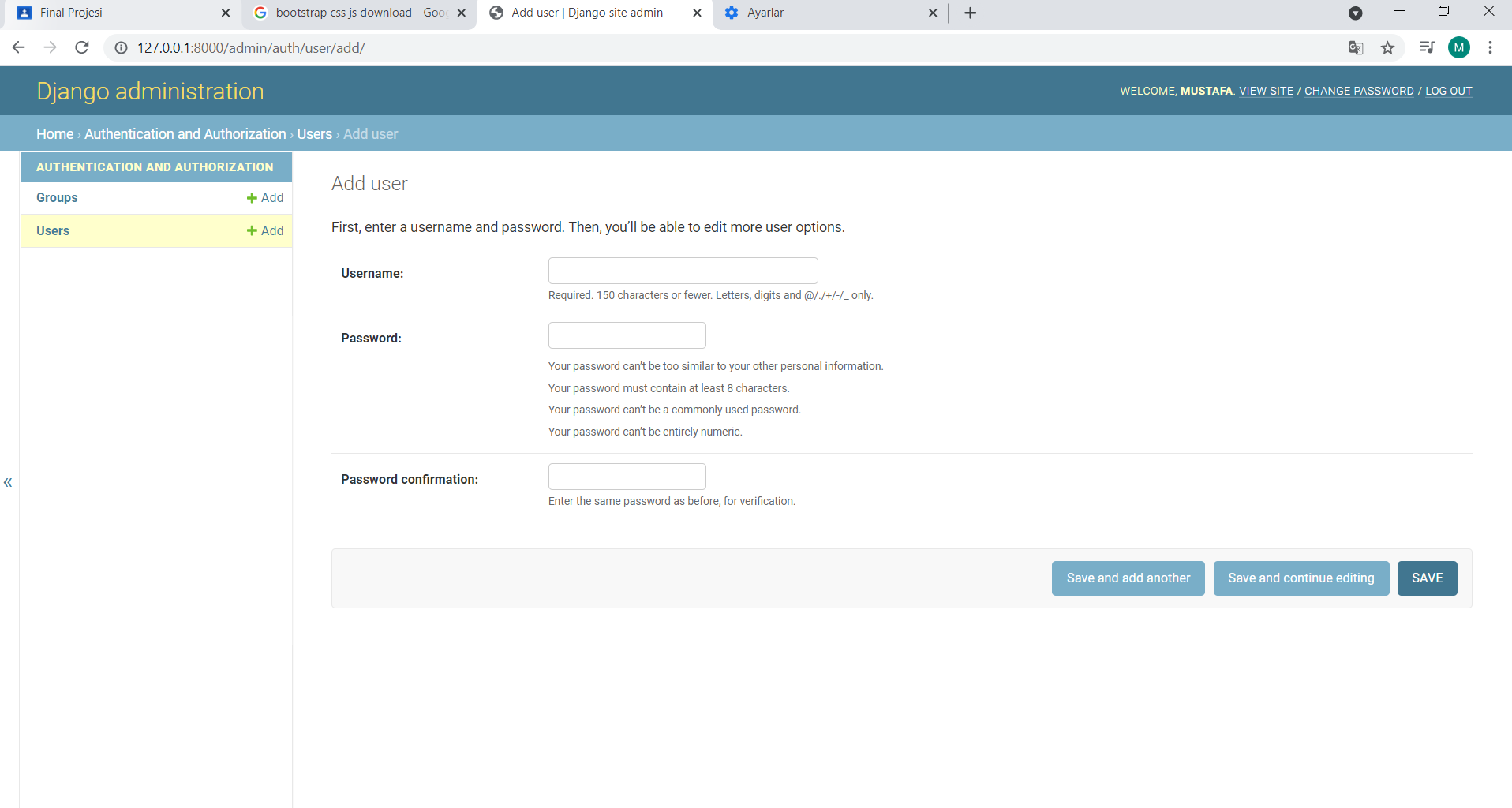
Veritabanı migrasyondan sonra bu şekildedir.



Admin paneline localhost:8080/admin denilerek girilir ve karşımıza bl bir giriş kısmı çıkar



python manage.py cretaesuperuser ile oluşturulan bir admin . tabi yetkilidir.



Burda user ekleme manulde yapılabilir.

Movies de models.py dosyasına kodlar yazılarak bir movies tablosu ismi ,açıklaması,resmi,tarihi olan bi tablo oluşturulabilir.

from django.db import models

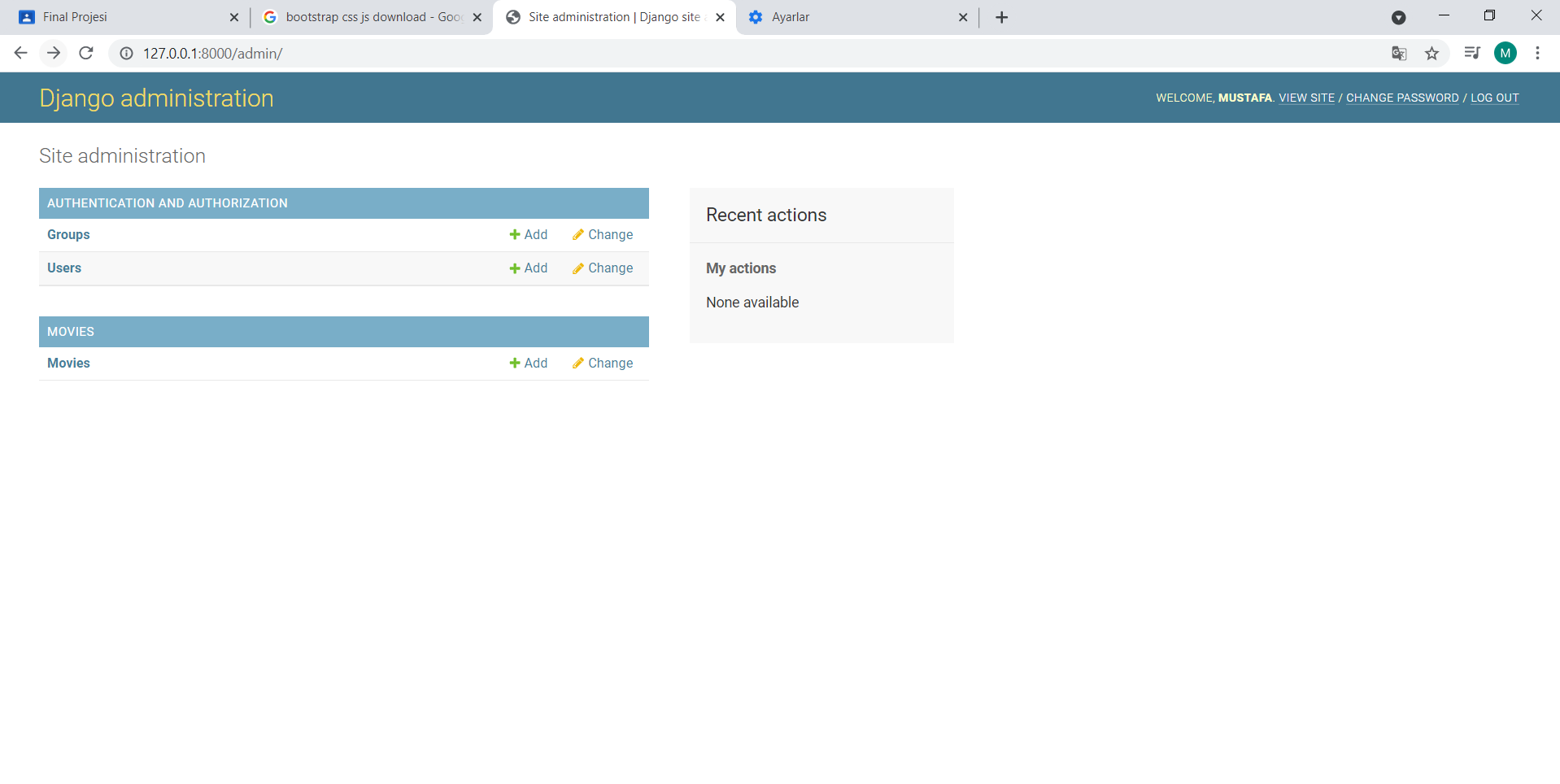
class Movie(models.Model):

name=models.CharField(max\_length=100)

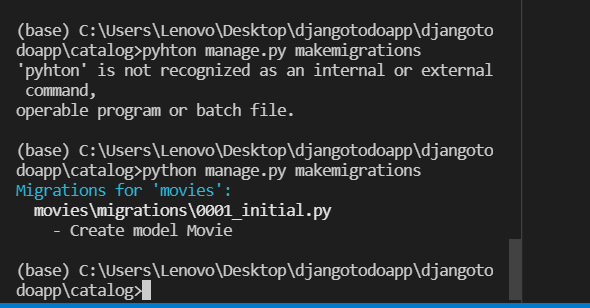
description=models.TextField()

image=models.CharField(max\_length=50)

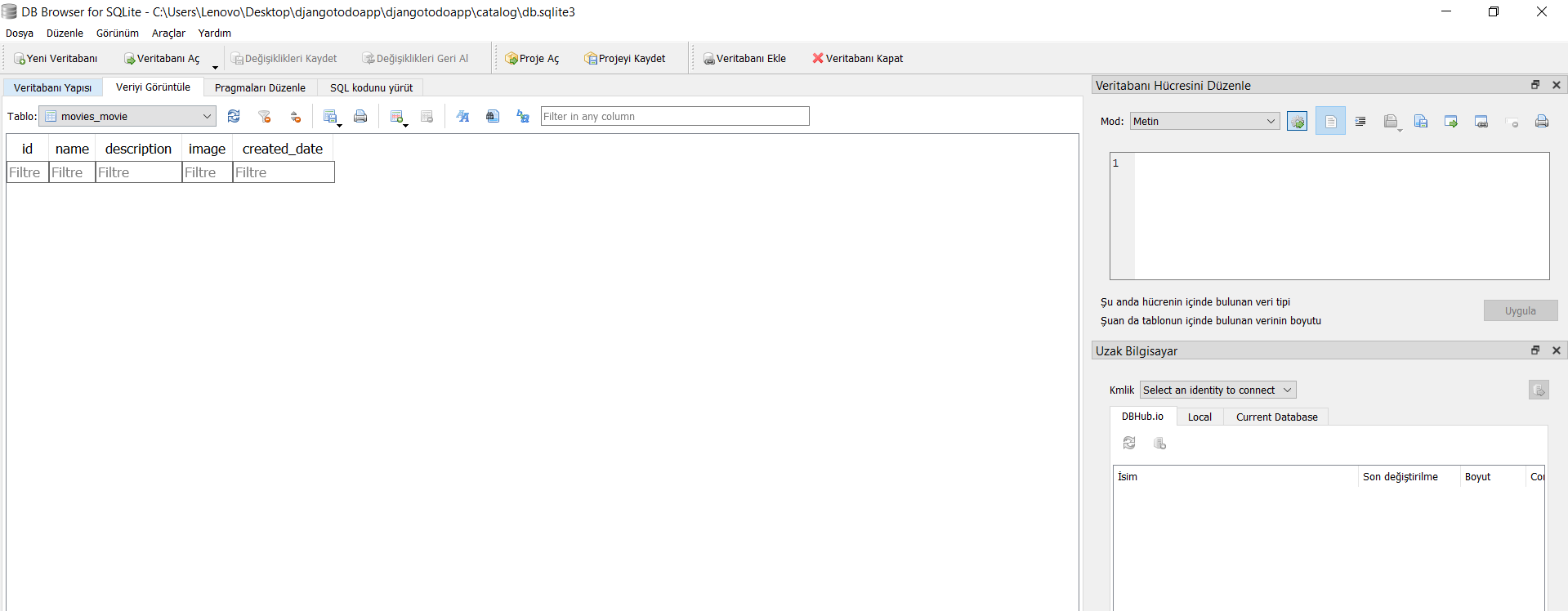
created\_date=models.DateTimeField(auto\_now\_add=True)



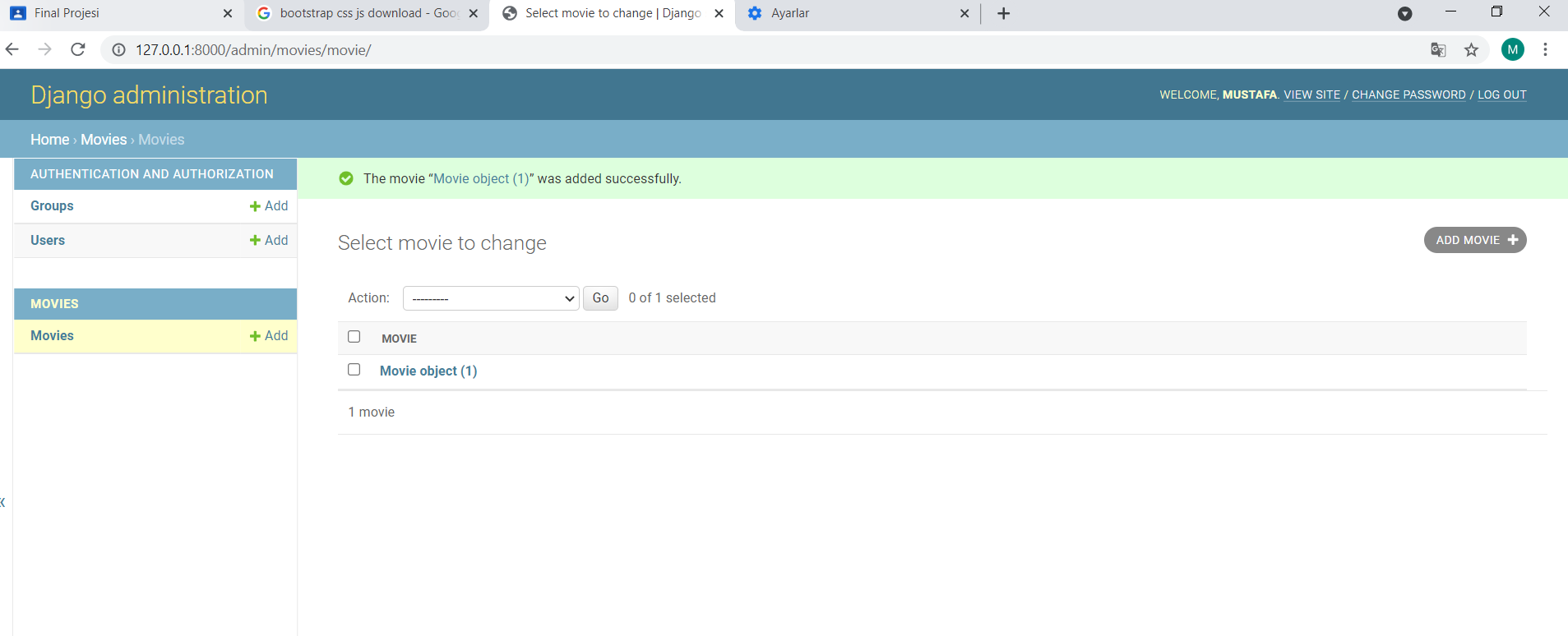
Eklenen model admin paneline bu şekilde yansır



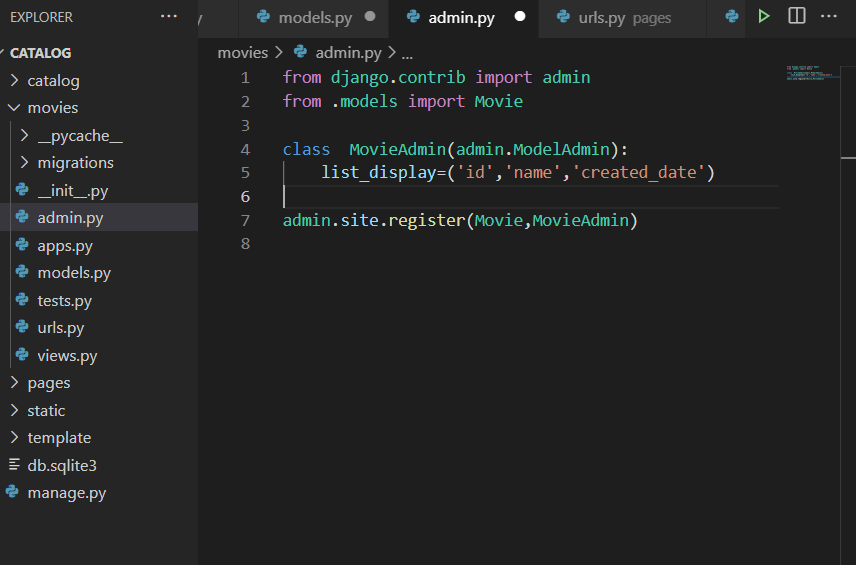
Makemigrations diyerek olan movie değişkliklerini aldık

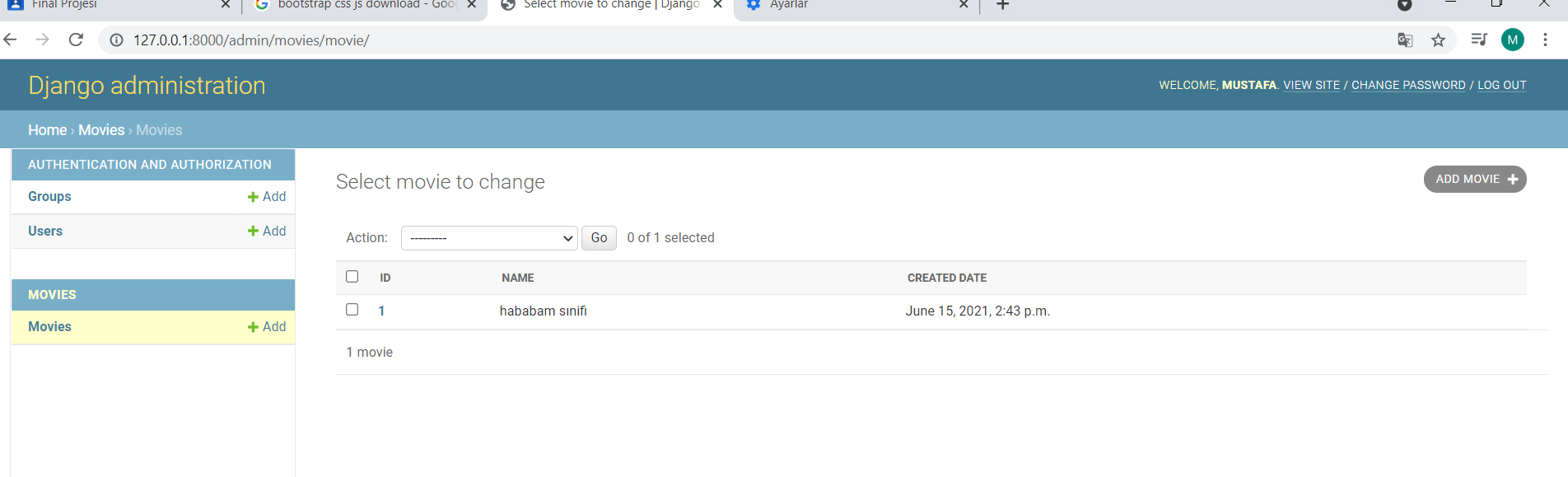


Veritabanında tablo bu şekilde görünüyor id kısmı otomatik gelir.

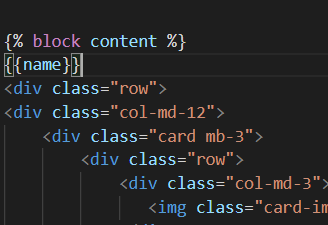
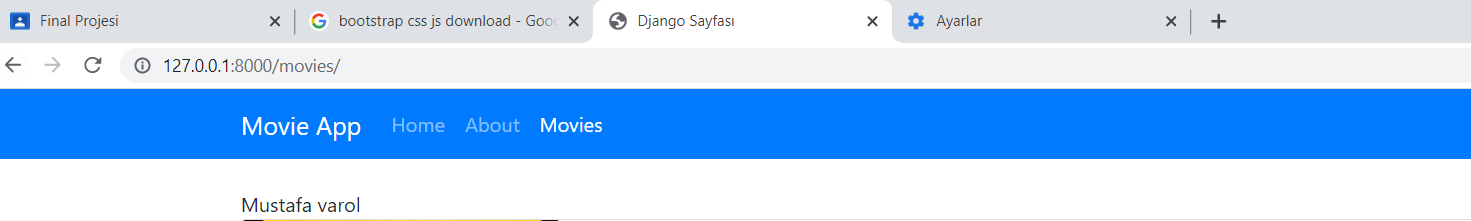


Admin panelinde manuıel film eklenebilir ve bunlar database e yansır.



Admin paneli nde movie kısmı bu resimde görüldüğü gibi özelleştirebilir. 

Bir paramatre gönderim yaparken örnek olarak isim gönderilmiştir.



context= {

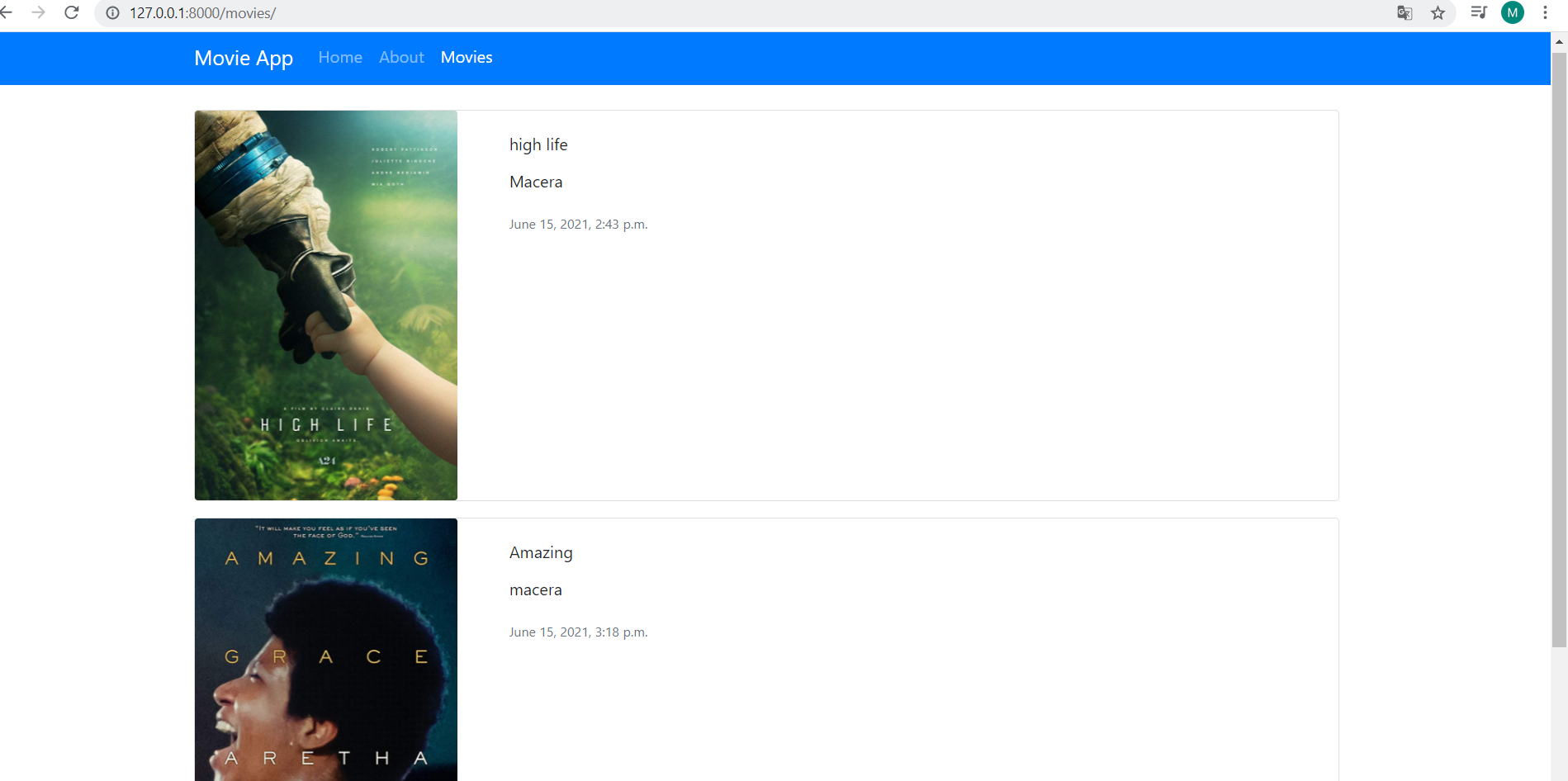
'name': 'Mustafa varol'

}

return render(request,'movies/list.html',context)

moviesin index html sayfasına bu yazılırsa name yukarda görüldüğü gibi alınabilir.

Bu denemenin amacı veritabanından bu şekilde bilgi almak ve sayfada göstermektir.



movies = Movie.objects.all()

context= {

'movies': movies

}

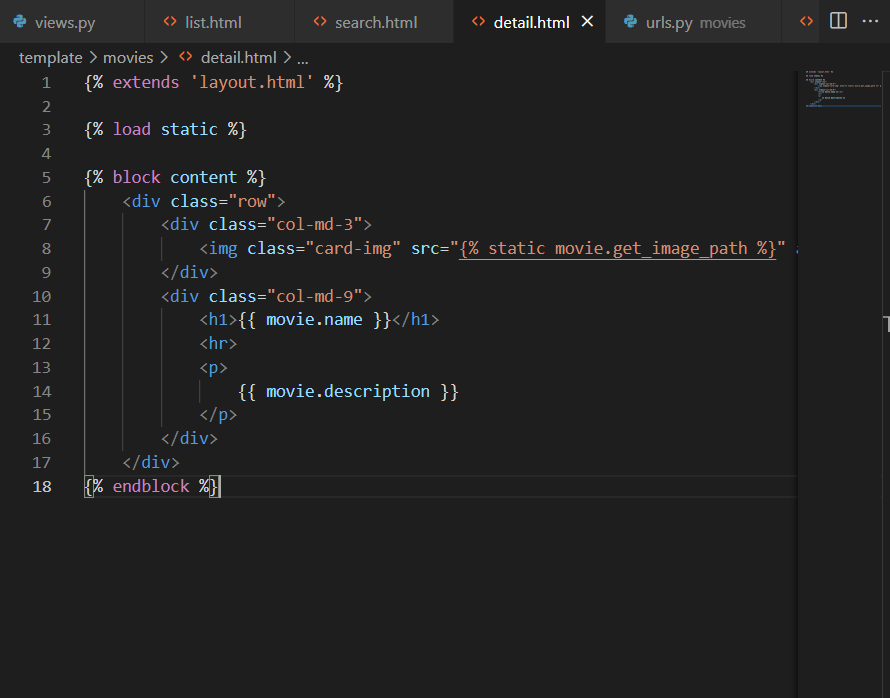
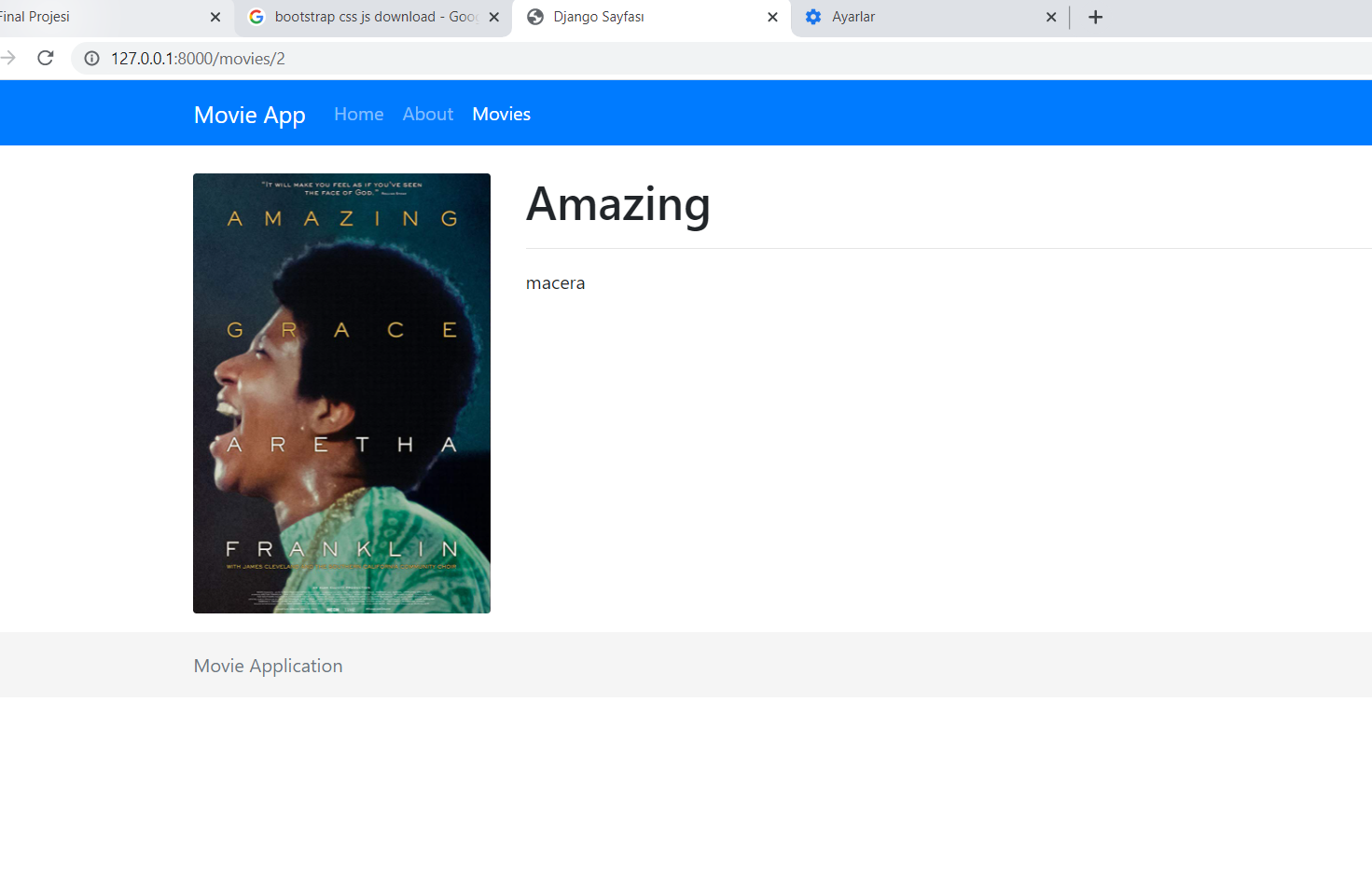
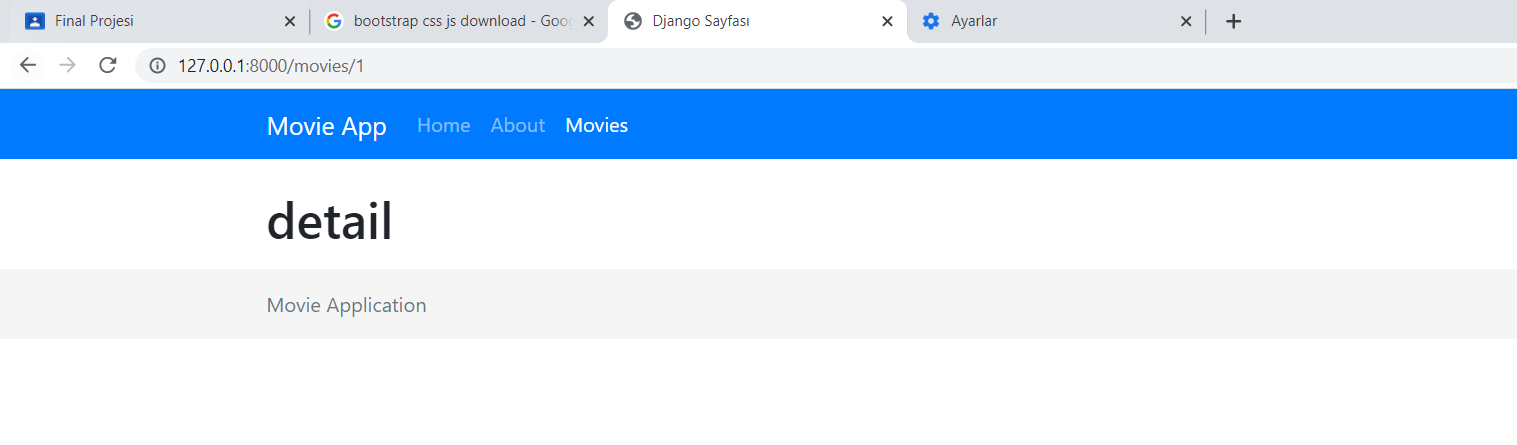
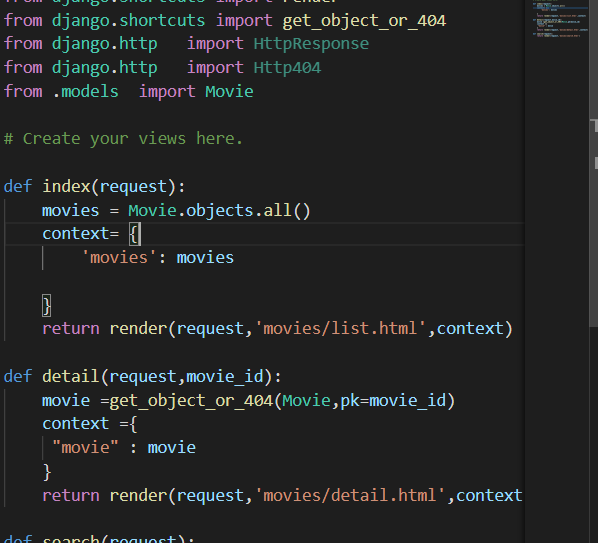
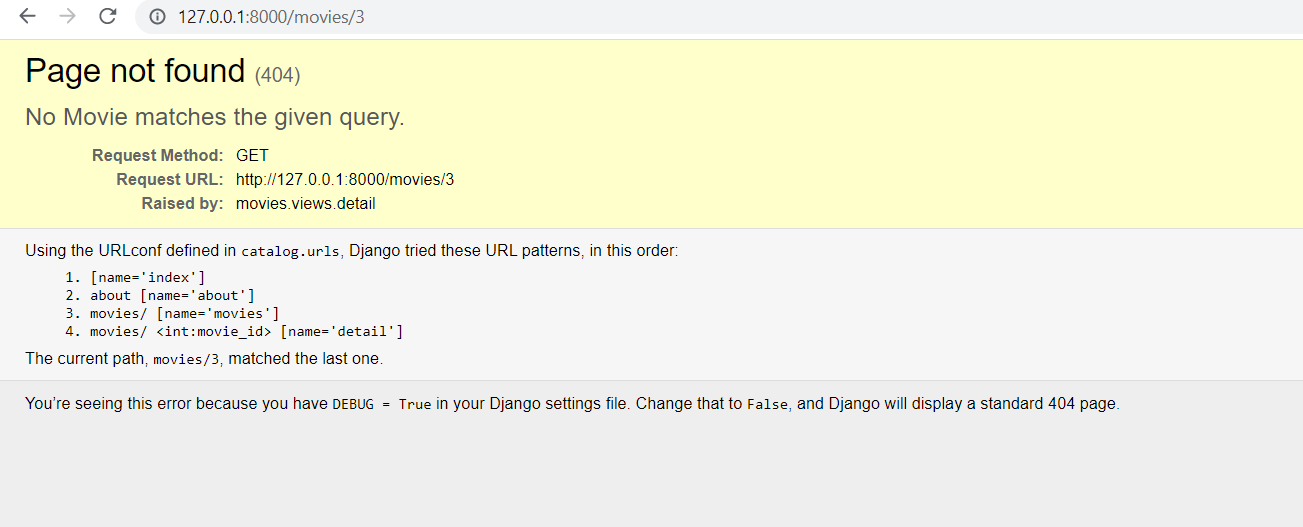
return render(request,'movies/list.html',movies) --- Bu şekilde bütün filmler alınır

AŞAĞIDAKİ İŞLEMLERDE detail e parametre olarak bir integer değer gönderileceği belirtilmiş.

Kod olarak da gelen integer değer veritabanında bir film id sine eşitse onun çıktısını deilse

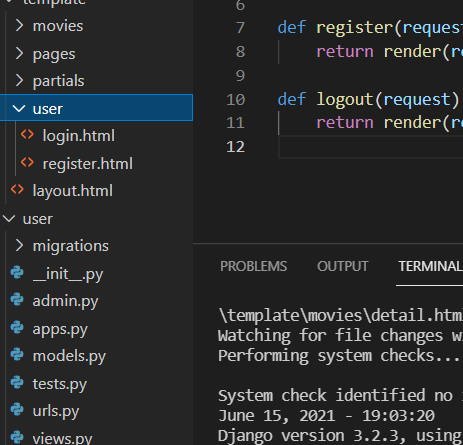
404 not found sayfasını getiriyor

Gerekli diyer açıklamalar kod kütüphanesinde yapılmştır

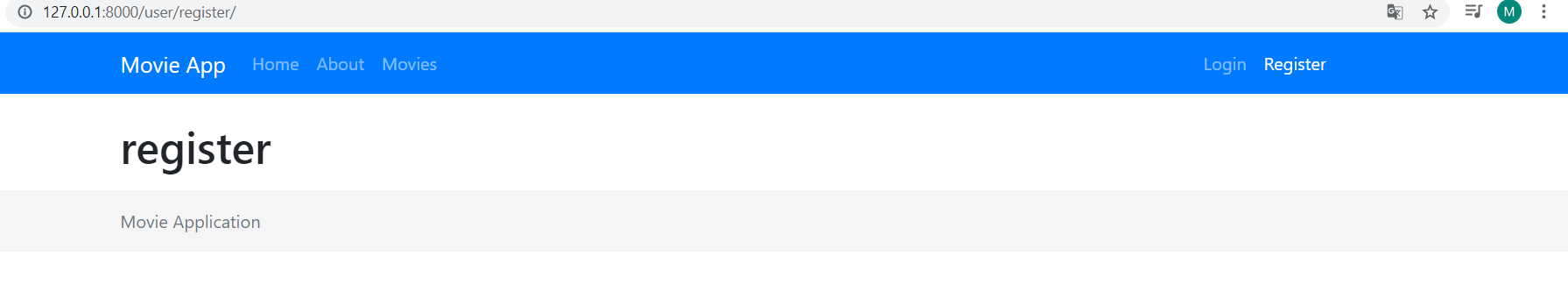


Movie ye ayrıca get image path adında bi method tanımlanmıştır resim yolunu otomatik almak için

USER işlemleri user projesi oluşturulur

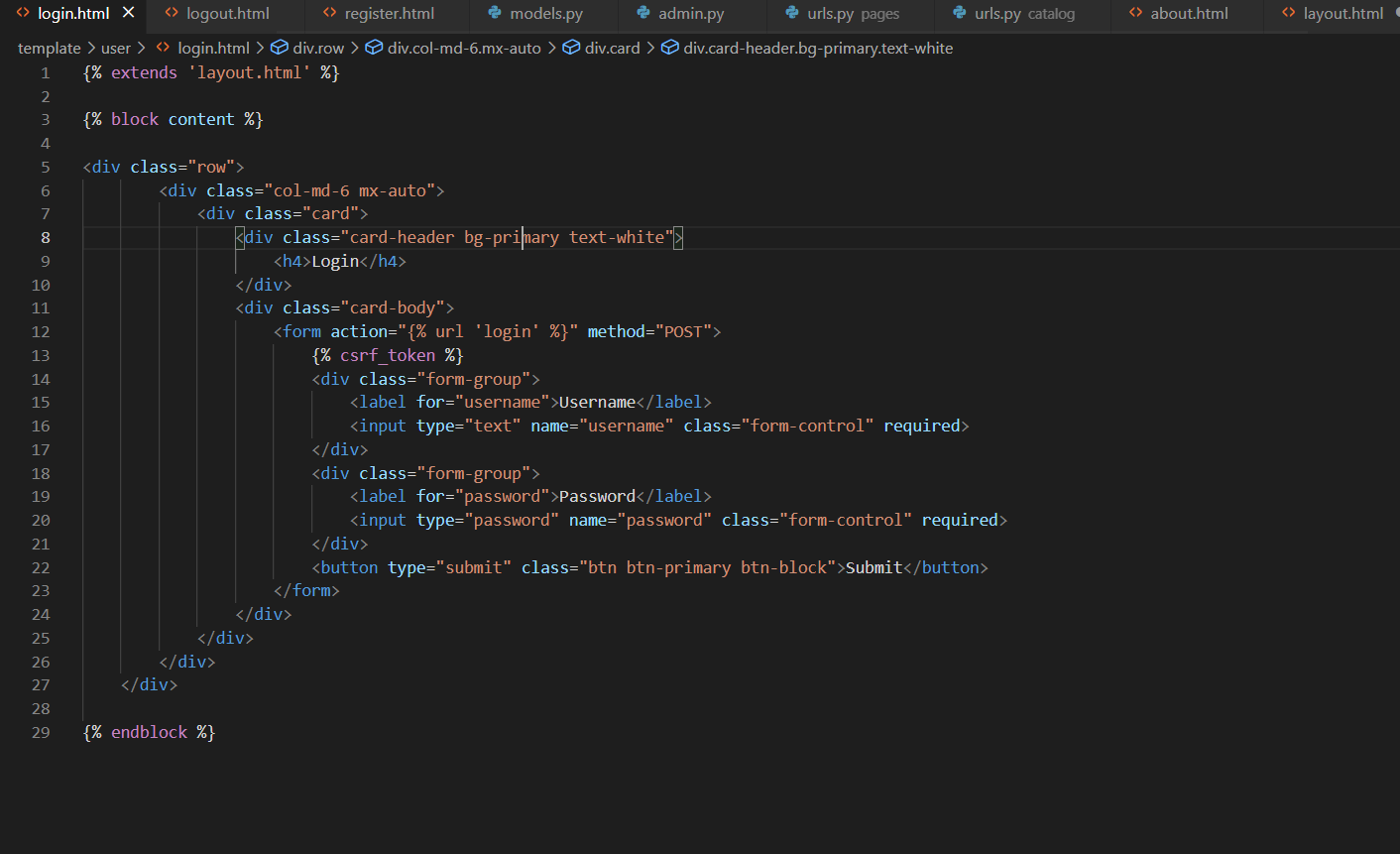


Login register sağ tarafa eklenir

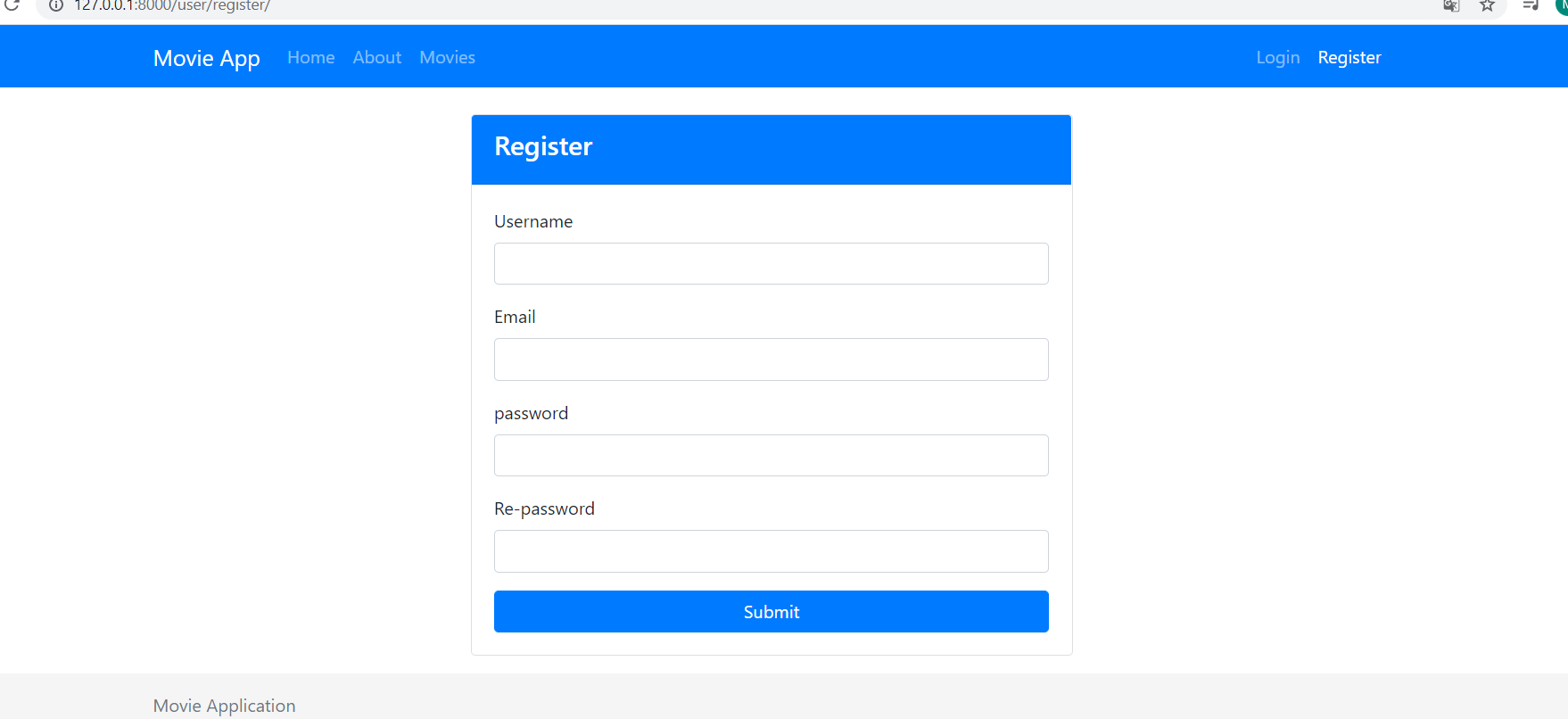


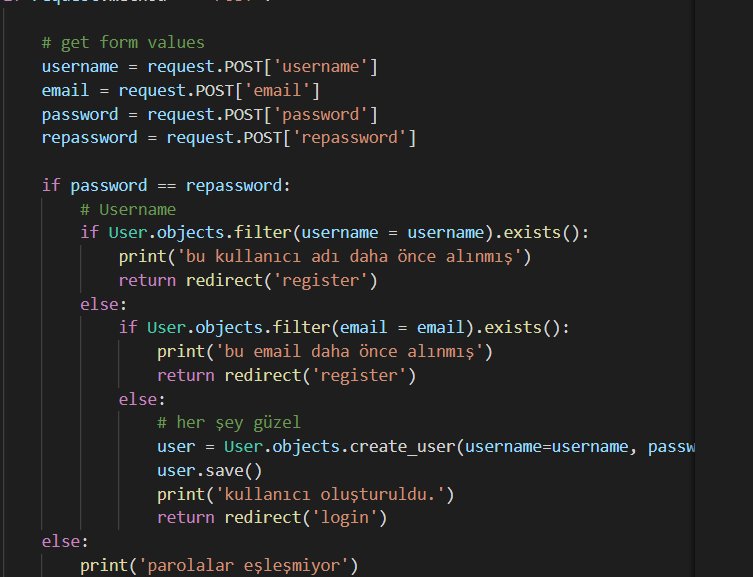
Bir register formu html kodlarla oluşturulur

Csrf\_token güvenlik doğrulaması eklenmelidir

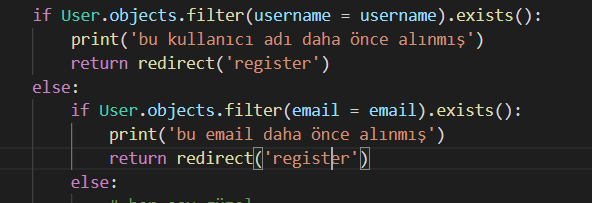


Form aşagıdakş şekilde görünürr





Post methodlarıyla gelen isim şifre mail gibi değerler karşılaştırmaları yapılır daha önce alınıp alnmadğna bakılır.



Bu şekilde kontrolleri yapılmış olur

HER ŞEY İÇİN TEŞEKKÜRLER HOCAM ☺☺☺