



Django İle Python Arka Yüz Yazılım Geliştirme

Teknik Bilimler
Meslek Yüksekokulu

Öğr. Gör. Furkan DURMUŞ



Bu Haftanın Ders Kazanımları



Statik Düzenleme

Anasayfa Son Yazılar

Migrations Oluşturma



Furkandurmus.com

-> Dersler

-> Django

-> Ders Uygulamaları

-> Blog Proje Kodları

Üzerinden projenizi güncelleyebilirsiniz...





```
require 'capybara/re Rails'
```

```
1 Capybara.javascript_driver = :webkit
```

```
2 Category.delete_all; Category.create
```

```
13 Shoulda::Matchers.configure do |config|
```

```
14 config.integrate do |with|
```

```
15 with.test_framework :rspec
```

```
16 with.library :rails
```

```
17 end
```

```
18 end
```

```
19 # Add additional requires below this line. When not
```

```
20 # Requires supporting ruby files with support/
```

```
21 # spec/support/ and its subdirectories. These files
```

```
22 # run as spec files by default. This will be
```

```
23 # in _spec.rb will both be required and run. It is recommended that you
```

Statik Düzenleme

Blog Statik Düzenleme

Son Yazılar



Django'ya Giriş

Ahmet Faruk DURSUN - 120 görüntülenme

Django framework'üne hızlı bir başlangıç.

[Yazının Devamı...](#)



Template Mantığı

Tuncay ALTUN - 85 görüntülenme

Django template sistemi nasıl çalışır?

[Yazının Devamı...](#)



Şuan tüm blogların görselleri aynı görsel.

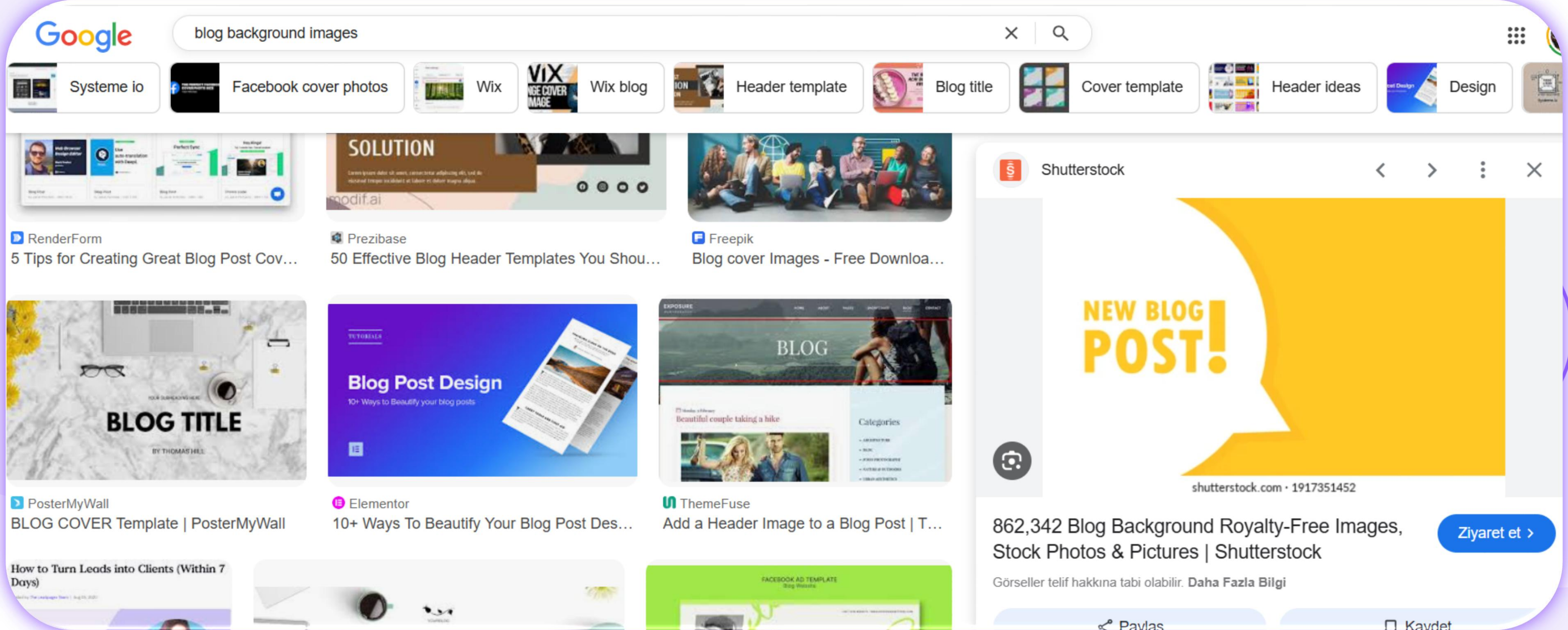
Blog Statik Düzenleme

```

<div class="card-body">
  <h5 class="card-title">{{ post.title }}</h5>
  <p class="text-muted">{{ post.author_name }} - {{ post.views }} görüntülenme</p>
  <p class="card-text">{{ post.summary }} </p>
  <a href="#" class="btn btn-primary"> Yazının Devamı...</a>
</div>
```

Anasayfa.html üzerinde tüm src'ler for döngüsünde aynı setleniyor.

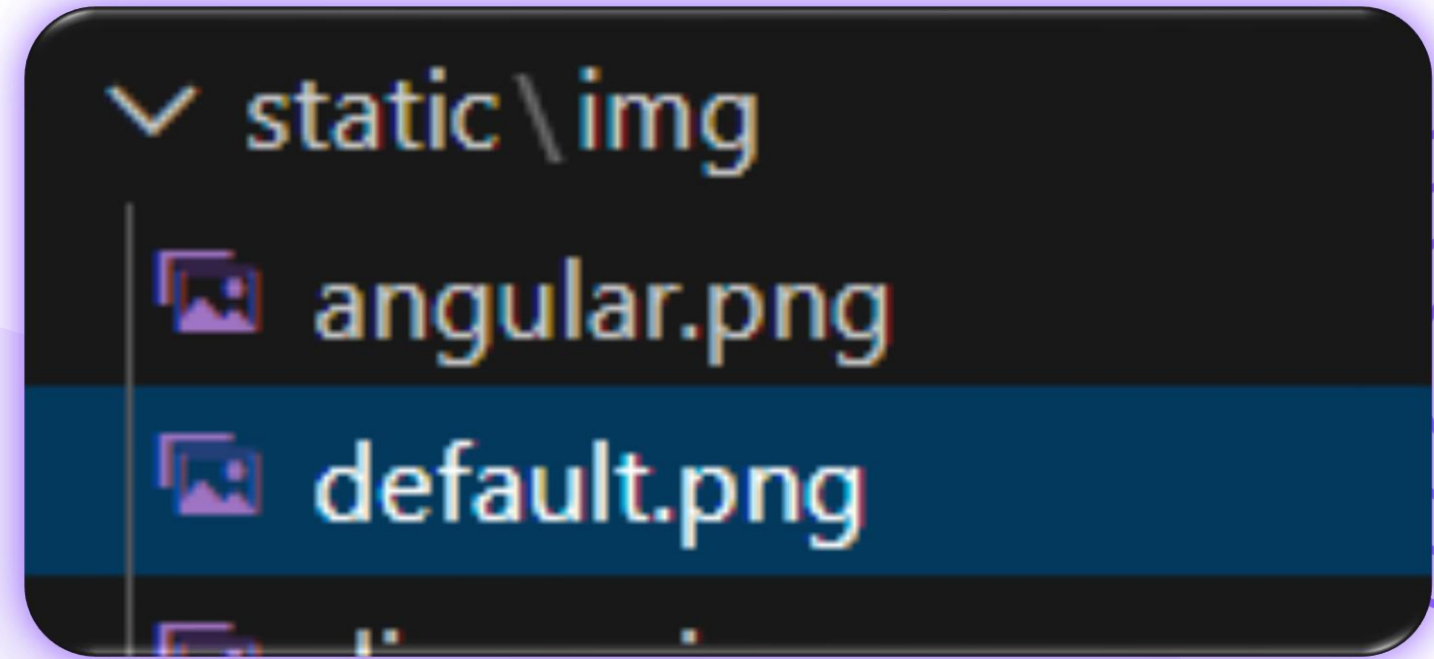
Blog Statik Düzenleme



İnternette 'Blog background Images' yazarak varsayılan bir blog görseli bulunuz.

Blog Statik Düzenleme

İndirdiğiniz görseli static/img altına 'default'
ismiyle kaydediniz.



Blog Statik Düzenleme

Blog - Models.py içerisindeki BlogPost Class'ına **static_url** field'ı ekleyiniz.

Default olarak az önce indirdiğiniz görselin dosya yolunu başına static yazmadan belirtiniz.

```
blog > models.py > ...
1  from django.db import models
2
3  class BlogPost(models.Model):
4      title = models.CharField(max_length=200)
5      summary = models.CharField(max_length=300)
6      content = models.TextField()
7      author_name = models.CharField(max_length=100)
8      views = models.IntegerField(default=0)
9      created_at = models.DateTimeField(auto_now_add=True)
10     is_published = models.BooleanField(default=True)
11
12     static_url = models.CharField(max_length=200, default='img/default.png')
13
14     def __str__(self):
15         return self.title
16
```

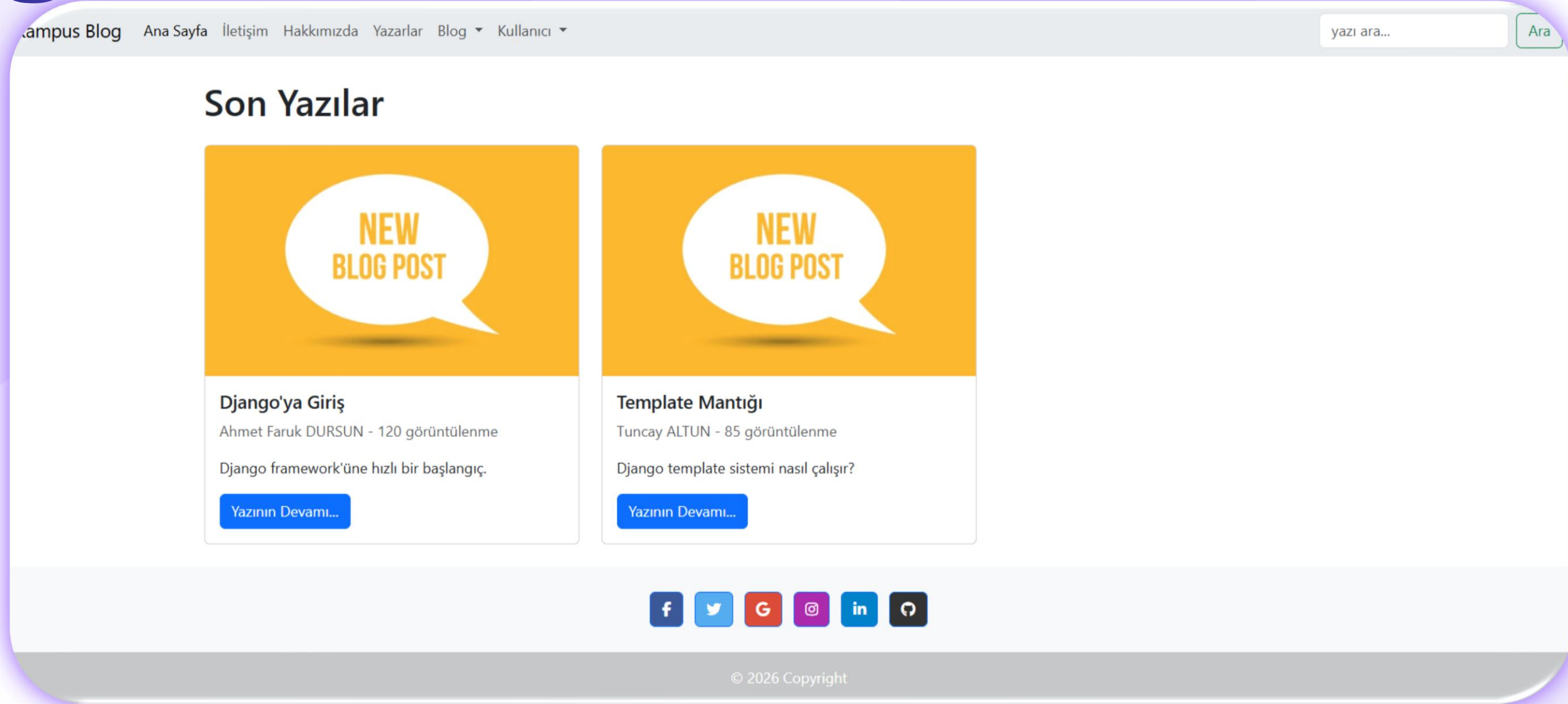


Blog Statik Düzenleme

```
{% for post in bloglar %}
<div class="col-md-4 mb-4">
  <div class="card h-100">
    
    <div class="card-body">
      <h5 class="card-title">{{ post.title }}</h5>
      <p class="text-muted">{{ post.author_name }} - {{ post.views }} görüntülenme</p>
      <p class="card-text">{{ post.summary }} </p>
      <a href="#" class="btn btn-primary"> Yazının Devamı...</a>
    </div>
  </div>
</div>
</div>
```

Anasayfa.html içersindeki img src kısmını üstteki gibi güncelleyiniz ve projeyi çalıştırınız.

Blog Statik Düzenleme



Şu ana kadar yapılan işlemleri yorumlayınız. Neden üstteki postlar yine aynı görsele sahip ?

Blog Statik Düzenleme

static \img

- angular.png
- default.png
- django.jpg
- html.jpg
- http.jpg
- node.png
- php.png
- python.png
- redis.png
- signalr.png
- > templates

```
def anasayfa(request):  
    bloglar = [  
        blog.models.BlogPost(  
            title="Django'ya Giriş",  
            summary="Django framework'üne hızlı bir başlangıç.",  
            content="Django, Python ile yazılmış güçlü bir web framework'tür...",  
            author_name="Ahmet Faruk DURSUN",  
            views=120,  
            static_url='img/django.jpg'  
        ),  
        blog.models.BlogPost(  
            title="Django'ya Giriş",  
            summary="Django framework'üne hızlı bir başlangıç.",  
            content="Django, Python ile yazılmış güçlü bir web framework'tür...",  
            author_name="Ahmet Faruk DURSUN",  
            views=120,  
            static_url='img/django.jpg'  
        ),  
        blog.models.BlogPost(  
            title="Django'ya Giriş",  
            summary="Django framework'üne hızlı bir başlangıç.",  
            content="Django, Python ile yazılmış güçlü bir web framework'tür...",  
            author_name="Ahmet Faruk DURSUN",  
            views=120,  
            static_url='img/django.jpg'  
        ),  
        blog.models.BlogPost(  
            title="Django'ya Giriş",  
            summary="Django framework'üne hızlı bir başlangıç.",  
            content="Django, Python ile yazılmış güçlü bir web framework'tür...",  
            author_name="Ahmet Faruk DURSUN",  
            views=120,  
            static_url='img/django.jpg'  
        ),  
        blog.models.BlogPost(  
            title="Django'ya Giriş",  
            summary="Django framework'üne hızlı bir başlangıç.",  
            content="Django, Python ile yazılmış güçlü bir web framework'tür...",  
            author_name="Ahmet Faruk DURSUN",  
            views=120,  
            static_url='img/django.jpg'  
        ),  
        blog.models.BlogPost(  
            title="Django'ya Giriş",  
            summary="Django framework'üne hızlı bir başlangıç.",  
            content="Django, Python ile yazılmış güçlü bir web framework'tür...",  
            author_name="Ahmet Faruk DURSUN",  
            views=120,  
            static_url='img/django.jpg'  
        )  
    ]  
    return render(request, "index.html", {"bloglar": bloglar})
```

Eğer core / views.py içerisindeki bloglara özel olarak static_url eklenmezse varsayılan değeri ekler.

Sizde 6 adet görsel bulunuz ve 6 blog yazısını ön yüze gönderiniz.

Blog Statik Düzenleme

Listenin sıralamasını kontrol edin.
views.py de son eklediğiniz eleman
ekranda nerde gözüküyor, nerede
olmalıydı ?

Son Yazılar



Django'ya Giriş

Ahmet Faruk DURSUN - 120 görüntülenme

Django framework'üne hızlı bir başlangıç.

[Yazının Devamı...](#)



angular ile Modern Web Geliştirme

Tuncay ALTUN - 95 görüntülenme

angular kütüphanesi ile dinamik kullanıcı arayüzleri oluşturma.

[Yazının Devamı...](#)

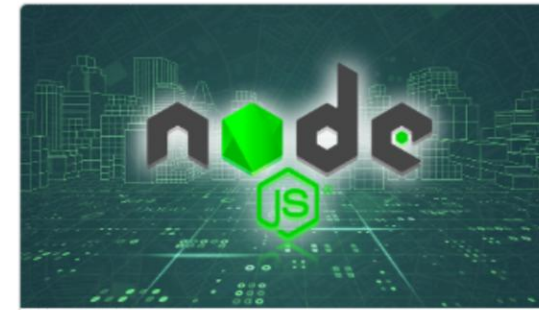


Veri Bilimi ve Python

Murat KAYA - 150 görüntülenme

Python'un veri bilimi alanındaki gücü.

[Yazının Devamı...](#)



Node.js ile Sunucu Tarafı Geliştirme

Ahmet Faruk DURSUN - 100 görüntülenme

Node.js kullanarak scalable ve yüksek performanslı sunucu tarafı uygulamalar geliştirme.

[Yazının Devamı...](#)



html5 ve CSS3 ile Modern Web Tasarımı

Elif GÖKÇE - 130 görüntülenme

Modern web tasarımı için html5 ve CSS3 teknikleri.

[Yazının Devamı...](#)



http protokolü ve web güvenliği

Ahmet Faruk DURSUN - 85 görüntülenme

Web güvenliği ve http protokolü hakkında bilgiler.

[Yazının Devamı...](#)

Blog Statik Düzenleme

```
sayilar = [1, 2, 3, 4, 5]
```

Python da listeyi tersine nasıl çeviririz ?



Blog Statik Düzenleme

```
sayilar = [1, 2, 3, 4, 5]
```

```
sayilar = [1, 2, 3, 4, 5]  
sayilar = sayilar[::-1]  
print(sayilar)
```

`[::-1]` ile bir liste tersine çevrilebilir.

Blog Statik Düzenleme

Son Yazılar



Django'ya Giriş
Ahmet Faruk DURSUN - 120 görüntülenme
Django framework'üne hızlı bir başlangıç.

Yazının Devamı...



angular ile Modern Web Geliştirme
Tuncay ALTUN - 95 görüntülenme
angular kütüphanesi ile dinamik kullanıcı arayüzleri oluşturma.

Yazının Devamı...



Veri Bilimi ve Python
Murat KAYA - 150 görüntülenme
Python'un veri bilimi alanındaki gücü.

Yazının Devamı...



Node.js ile Sunucu Tarafı Geliştirme
Ahmet Faruk DURSUN - 100 görüntülenme
Node.js kullanarak scalable ve yüksek performanslı sunucu tarafı uygulamalar geliştirme.

Yazının Devamı...



html5 ve CSS3 ile Modern Web Tasarımı
Elif GÖKÇE - 130 görüntülenme
Modern web tasarımı için html5 ve CSS3 teknikleri.

Yazının Devamı...



http protokolü ve web güvenliği
Ahmet Faruk DURSUN - 85 görüntülenme
Web güvenliği ve http protokolü hakkında bilgiler.

Yazının Devamı...

[::-1]



Son Yazılar



http protokolü ve web güvenliği
Ahmet Faruk DURSUN - 85 görüntülenme
Web güvenliği ve http protokolü hakkında bilgiler.

Yazının Devamı...



html5 ve CSS3 ile Modern Web Tasarımı
Elif GÖKÇE - 130 görüntülenme
Modern web tasarımı için html5 ve CSS3 teknikleri.

Yazının Devamı...



Node.js ile Sunucu Tarafı Geliştirme
Ahmet Faruk DURSUN - 100 görüntülenme
Node.js kullanarak scalable ve yüksek performanslı sunucu tarafı uygulamalar geliştirme.

Yazının Devamı...



Veri Bilimi ve Python



angular ile Modern Web Geliştirme
Tuncay ALTUN - 95 görüntülenme
angular kütüphanesi ile dinamik kullanıcı arayüzleri oluşturma.

Yazının Devamı...



Django'ya Giriş
Ahmet Faruk DURSUN - 120 görüntülenme
Django framework'üne hızlı bir başlangıç.

Yazının Devamı...

```
context = {  
    "bloglar": bloglar[::-1]  
}
```

Context'i güncelleyerek son eklenen nesnenin en başta gözükmesini sağlayınız.



```
require 'capybara/re Rails'
```

```
1 Capybara.javascript_driver = :webkit
```

```
2 Category.delete_all; Category.create
```

```
13 Shoulda::Matchers.configure do |config|
```

```
14 config.integrate do |with|
```

```
15 with.test_framework :rspec
```

```
16 with.library :rails
```

```
17 end
```

```
18 end
```

```
19 # Add additional requires below this line. 20
```

```
21 # Requires supporting ruby files with support/ support/ 22
```

```
23 # spec/support/ and its subdirectories. 24
```

```
25 # run as spec files by default. 26
```

“Son Yazılar” Düzenleme

'Son Yazılar' Düzenleme

Son Yazılar



http protokolü ve web güvenliği
Ahmet Faruk DURSUN - 85 görüntülenme

Web güvenliği ve http protokolü hakkında bilgiler.

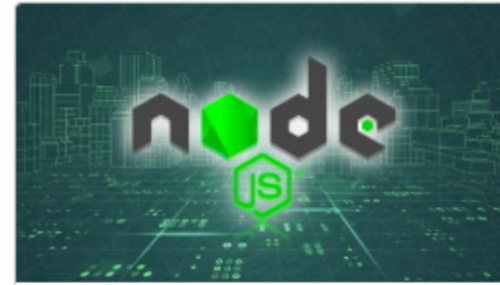
[Yazının Devamı...](#)



html5 ve CSS3 ile Modern Web Tasarımı
Elif GÖKÇE - 130 görüntülenme

Modern web tasarımı için html5 ve CSS3 teknikleri.

[Yazının Devamı...](#)



Node.js ile Sunucu Tarafı Geliştirme
Ahmet Faruk DURSUN - 100 görüntülenme

Node.js kullanarak scalable ve yüksek performanslı sunucu tarafı uygulamalar geliştirme.

[Yazının Devamı...](#)



Veri Bilimi ve Python
Murat KAYA - 150 görüntülenme

Python'un veri bilimi alanındaki gücü.

[Yazının Devamı...](#)



angular ile Modern Web Geliştirme
Tuncay ALTUN - 95 görüntülenme

angular kütüphanesi ile dinamik kullanıcı arayüzleri oluşturma.

[Yazının Devamı...](#)



Django'ya Giriş
Ahmet Faruk DURSUN - 120 görüntülenme

Django framework'üne hızlı bir başlangıç.

[Yazının Devamı...](#)

Son Yazılar



Redis ile Veritabanı Yönetimi
Mehmet ÖZTÜRK - 110 görüntülenme

Redis kullanarak hızlı ve etkili veritabanı yönetimi.

[Yazının Devamı...](#)



http protokolü ve web güvenliği
Ahmet Faruk DURSUN - 85 görüntülenme

Web güvenliği ve http protokolü hakkında bilgiler.

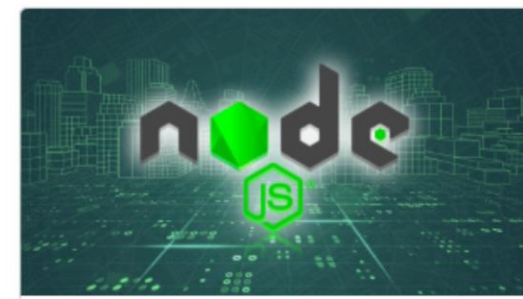
[Yazının Devamı...](#)



html5 ve CSS3 ile Modern Web Tasarımı
Elif GÖKÇE - 130 görüntülenme

Modern web tasarımı için html5 ve CSS3 teknikleri.

[Yazının Devamı...](#)



Node.js ile Sunucu Tarafı Geliştirme
Ahmet Faruk DURSUN - 100 görüntülenme

Node.js kullanarak scalable ve yüksek performanslı sunucu tarafı uygulamalar geliştirme.

[Yazının Devamı...](#)



Veri Bilimi ve Python
Murat KAYA - 150 görüntülenme

Python'un veri bilimi alanındaki gücü.

[Yazının Devamı...](#)



angular ile Modern Web Geliştirme
Tuncay ALTUN - 95 görüntülenme

angular kütüphanesi ile dinamik kullanıcı arayüzleri oluşturma.

[Yazının Devamı...](#)

Ana sayfadaki 'Son Yazılar' da sadece eklenen son 6 yazının gözükmesini sağlayınız.

‘Son Yazılar’ Düzenleme

```
context = {  
    "bloglar": bloglar[::-1][:6]  
}  
return render(request, "anasayfa.html", context)
```

```
bloglar = bloglar[::-1]  
bloglar = bloglar[:6]  
context = {  
    "bloglar": bloglar  
}
```

`[::-1]` ile listeyi ters çevirmiştik `[:6]` ile de ilk 6 elemanını alıyoruz. Böylece ters çevrilmiş listenin ilk 6 elemanı son eklenen 6 eleman olmuş oluyor.

Sizde üstteki güncellemeyi yapınız ve views'daki model'in sonuna bir eleman ekleyerek test ediniz.



```
require 'capybara/re Rails'
```

```
1 Capybara.javascript_driver = :webkit
```

```
2 Category.delete_all; Category.create
```

```
13 Shoulda::Matchers.configure do |config|
```

```
14 config.integrate do |with|
```

```
15 with.test_framework :rspec
```

```
16 with.library :rails
```

```
17 end
```

```
18 end
```

```
19 # Add additional requires below this line. 20
```

```
21 # Requires supporting ruby files with support/ support/ 22
```

```
23 # spec/support/ and its subdirectories. 24
```

```
25 # run as spec files by default. This will be required 26
```

Migrations

Migrations Oluşturma

Models.py

Migration, modelde yaptığın değişikliklerin veritabanına nasıl uygulanacağını anlatan dosyadır.

«» Kod

kampus_blog/

├── venv/

├── manage.py

├── config/

│ ├── __init__.py

│ ├── settings.py

│ ├── urls.py

│ ├── asgi.py

│ └── wsgi.py

└── core/

│ ├── admin.py

│ ├── apps.py

│ ├── models.py

│ ├── views.py

│ ├── urls.py

│ └── templates/

Migrations Oluşturma

BlogPost

Bu model Db de:

BlogPost → tablo olacak
title → sütun olacak
summary → sütun olacak
content → sütun olacak
views → sütun olacak
created_at → sütun olacak

```
from django.db import models
```

```
class BlogPost(models.Model):
```

```
    title = models.CharField(max_length=200)
```

```
    summary = models.CharField(max_length=300)
```

```
    content = models.TextField()
```

```
    author_name = models.CharField(max_length=100)
```

```
    views = models.IntegerField(default=0)
```

```
    created_at = models.DateTimeField(auto_now_add=True)
```

```
    is_published = models.BooleanField(default=True)
```

```
    static_url = models.CharField(max_length=200, default='img/default.png')
```

```
    def __str__(self):
```

```
        return self.title
```

Migrations Oluşturma

Django varsayılan olarak SQLite kullanır.

```
▼ DATABASES = {  
▼   'default': {  
     'ENGINE': 'django.db.backends.sqlite3',  
     'NAME': BASE_DIR / 'db.sqlite3',  
   }  
}
```



Şimdi yapmamız gereken benim modellerimi db ye taşımak. Bunun için projeyi durdurup terminali açıyorum.

```
○ (venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\kampus_blog>
```



Migrations Oluşturma

Amacım :

- 1- Önce projemde db de olmayan bir şey var mı onun farkını bulacağım.
- 2- Fark varsa Db güncelleyeceğim.



Migrations Oluşturma

Amacım :

1- Önce projemde db de olmayan bir şey var mı onun farkını bulacağım.

Terminalde : Bu kodu yazın ve enter'a basın.

python manage.py makemigrations



```
(venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\kampus_blog> python manage.py makemigrations
```

Migrations Oluşturma

```
• (venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\kampus_blog> python manage.py makemigrations
Migrations for 'blog':
  blog\migrations\0001_initial.py
  + Create model BlogPost
Migrations for 'core':
  core\migrations\0001_initial.py
  + Create model Author
```



Migrations Oluşturma

Makemigrations Ne yapar ?

- 1- INSTALLED_APPS içindeki app'leri kontrol eder,
- 2- her app'in models.py dosyasına bakar,
- 3- değişiklik var mı diye karşılaştırır.



Migrations Oluşturma

Makemigrations Ne yapar ?

- 1- INSTALLED_APPS içindeki app'leri kontrol eder,
- 2- her app'in models.py dosyasına bakar,
- 3- değişiklik var mı diye karşılaştırır.

makemigrations, Django'ya şunu söyler: Modellerime bak. Önceki haline göre değişiklik var mı? Varsa bunun migration dosyasını oluştur. Yani bu komut veritabanını değiştirmez. Sadece plan çıkarır.





Migrations Oluşturma

Amacım :

- 1- Önce projemde db de olmayan bir şey var mı onun farkını bulacağım.
- 2- Fark varsa **Db güncelleyeceğim.**



Migrations Oluşturma

Amacım :

2- Fark varsa Db güncelleyeceğim.

Migration dosyası oluştu ama veritabanına henüz uygulanmadı.

Terminalde : Bu kodu yazın ve enter'a basın.

python manage.py migrate

```
(venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\kampus_blog> python manage.py migrate
```

Migrations Olusturma

```
(venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\kampus_blog> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, blog, contenttypes, core, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying blog.0001_initial... OK
  Applying core.0001_initial... OK
  Applying sessions.0001_initial... OK
(venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\kampus_blog>
```



Migrations Oluşturma

Migrate Ne yapar ?

migrate, hazırlanan migration dosyalarını veritabanına uygular.

Yani:

makemigrations → plan hazırlar

migrate → planı uygular

migrate, artık gerçekten veritabanında tabloyu **oluşturur** veya **değiştirir**.

Bu komuttan sonra SQLite içinde blog_blogpost gibi bir tablo oluşur.





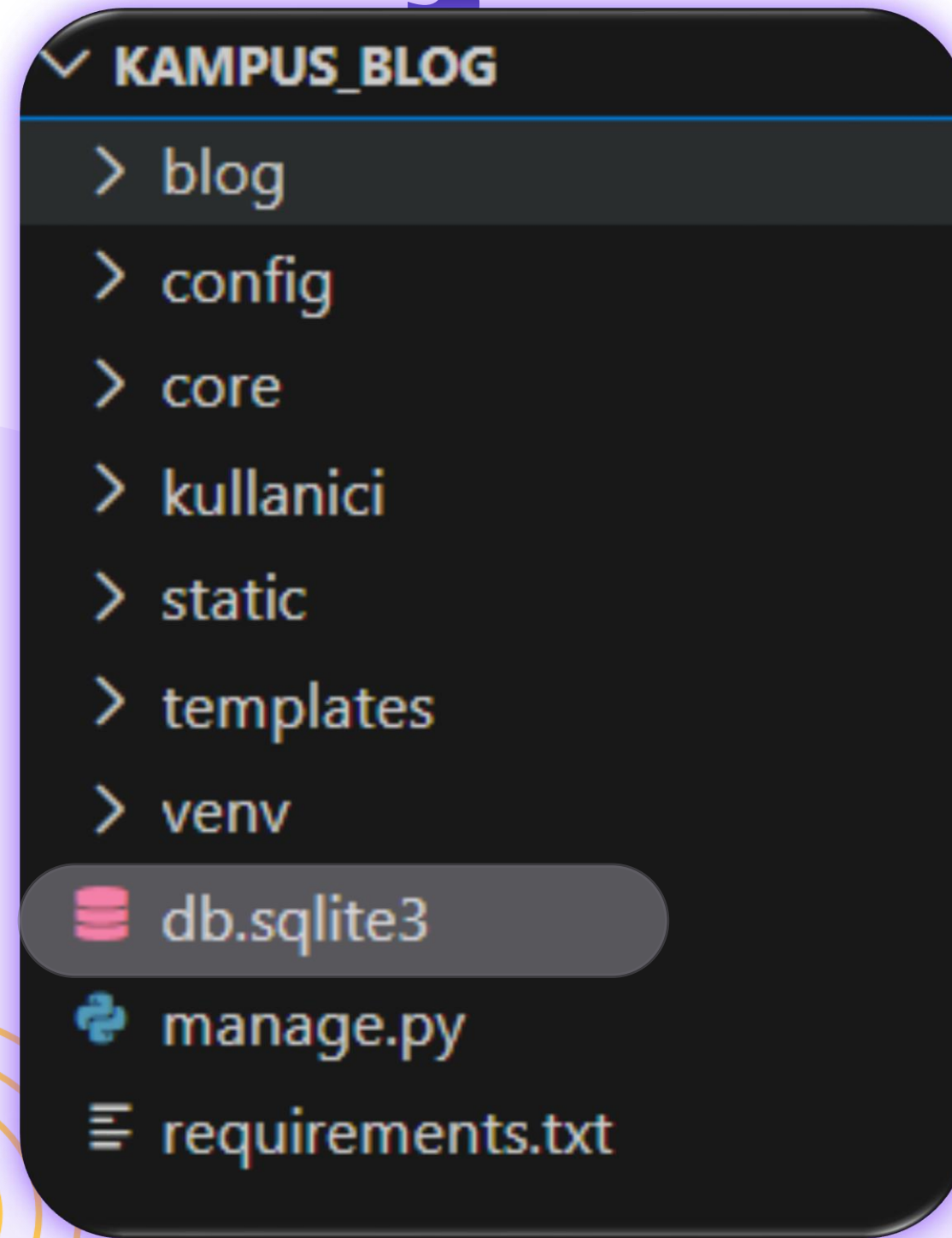
Migrations Oluşturma

Eklediğimiz tabloları kontrol etmek için SQLite Viewer Adındaki eklentiye kurabiliriz .

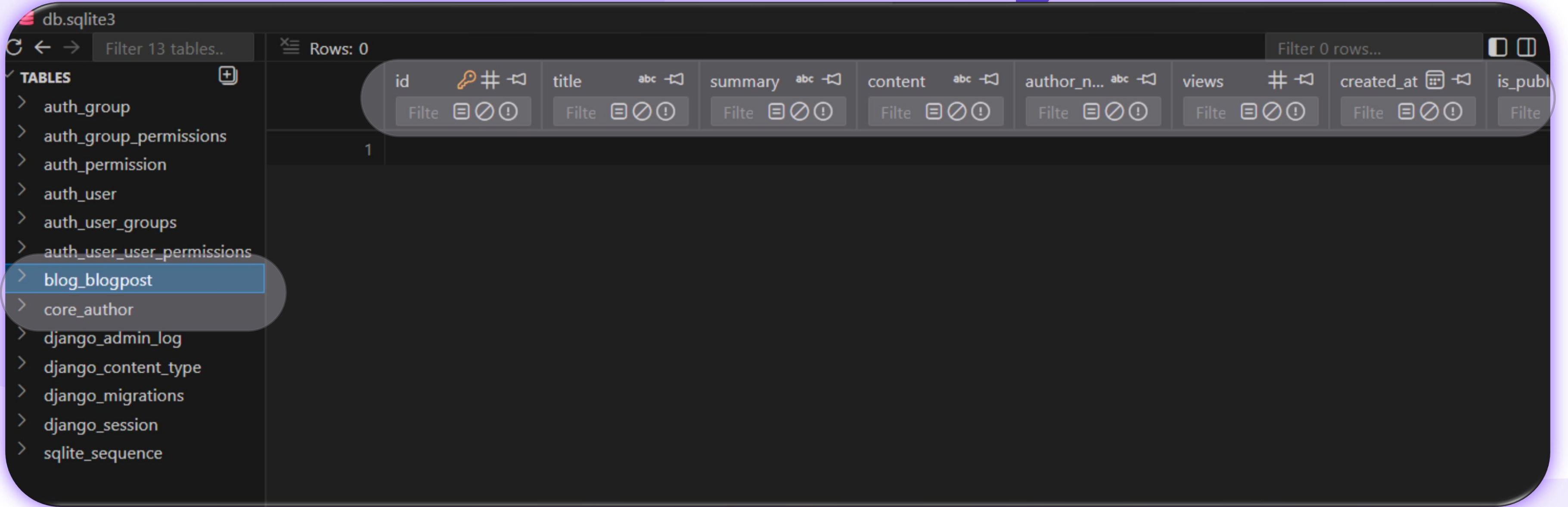


Migrations Oluşturma

Eklentiyi kurduktan sonra proje klasöründeki db.sqlite3'e tıklayınız.



Migrations Oluşturma



Gözükteğü üzere rows : 0. Şimdi buraya terminalden veri ekleyelim.

Migrations Oluşturma

Amacım :

1- Terminalden DB ye veri ekleme;

Terminalde : Bu kodu yazın ve enter'a basın.

python manage.py shell



```
(venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\kampus_blog> python manage.py shell
```

Migrations Oluşturma

```
(venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\kampus_blog> python manage.py shell  
14 objects imported automatically (use -v 2 for details).
```

```
Ctrl click to launch VS Code Native REPL
```

```
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct 7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)] on win32
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
(InteractiveConsole)
```

```
>>> █
```

Shell Django projesinin içinde Python kodu çalıştırabildiğimiz ortamdır.

Migrations Oluşturma

```
>>> from blog.models import BlogPost
```

Shell'de üstteki gibi blog model'ini çağırıp enter'a basın.

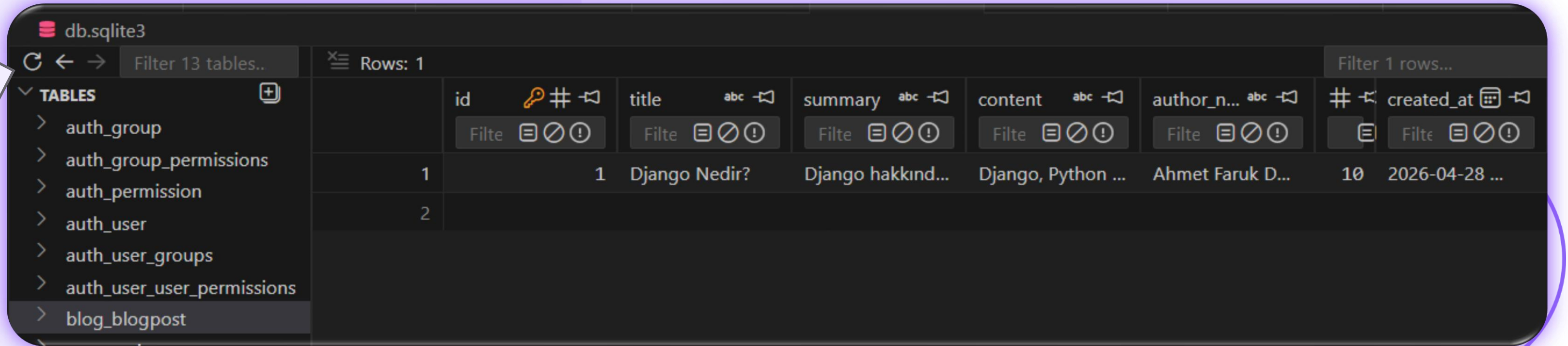
Migrations Oluşturma

```
>>> from blog.models import BlogPost
```

```
>>> BlogPost.objects.create(title="Django Nedir?",summary="Django hakkında kısa bilgi",  
content="Django, Python ile yazılmış güçlü bir web framework'tür.",author_name="Ahmet F  
aruk DURSUN",views=10, static_url="img/django.jpg")
```

Şimdi bir BlogPost create eden üstteki kodu yazınız.

Migrations Oluşturma



db.sqlite3

Filter 13 tables..

Rows: 1

Filter 1 rows..

	id	#	title	summary	content	author_n...	#	created_at
1	1		Django Nedir?	Django hakkınd...	Django, Python ...	Ahmet Faruk D...	10	2026-04-28 ...
2	2							

TABLES

- auth_group
- auth_group_permissions
- auth_permission
- auth_user
- auth_user_groups
- auth_user_user_permissions
- blog_blogpost

Db.sqlite3 açıp sol üstteki yenile butonuna tıkladığınızda blog_blogpost tablosuna verimizin düştüğünü görebiliriz.

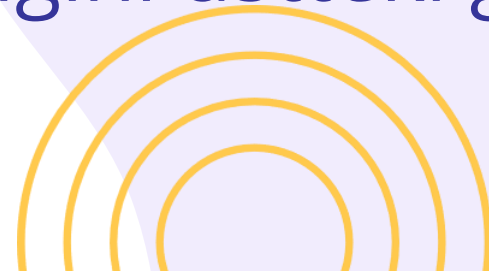
Migrations Oluşturma

```
def anasayfa(request):  
  
    bloglar = blog.models.BlogPost.objects.all()  
  
    context = {  
        "bloglar": bloglar  
    }  
  
    return render(request, "anasayfa.html", context)
```



Son olarak views.py tarafında son güncellememizi yapalım.

Mevcut anasayfa fonksiyonunu içeriğini üstteki gibi yapın, önceki kodları silebilirsiniz.

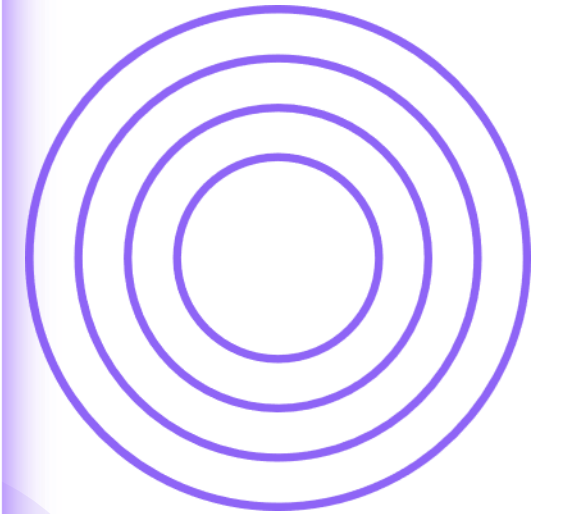
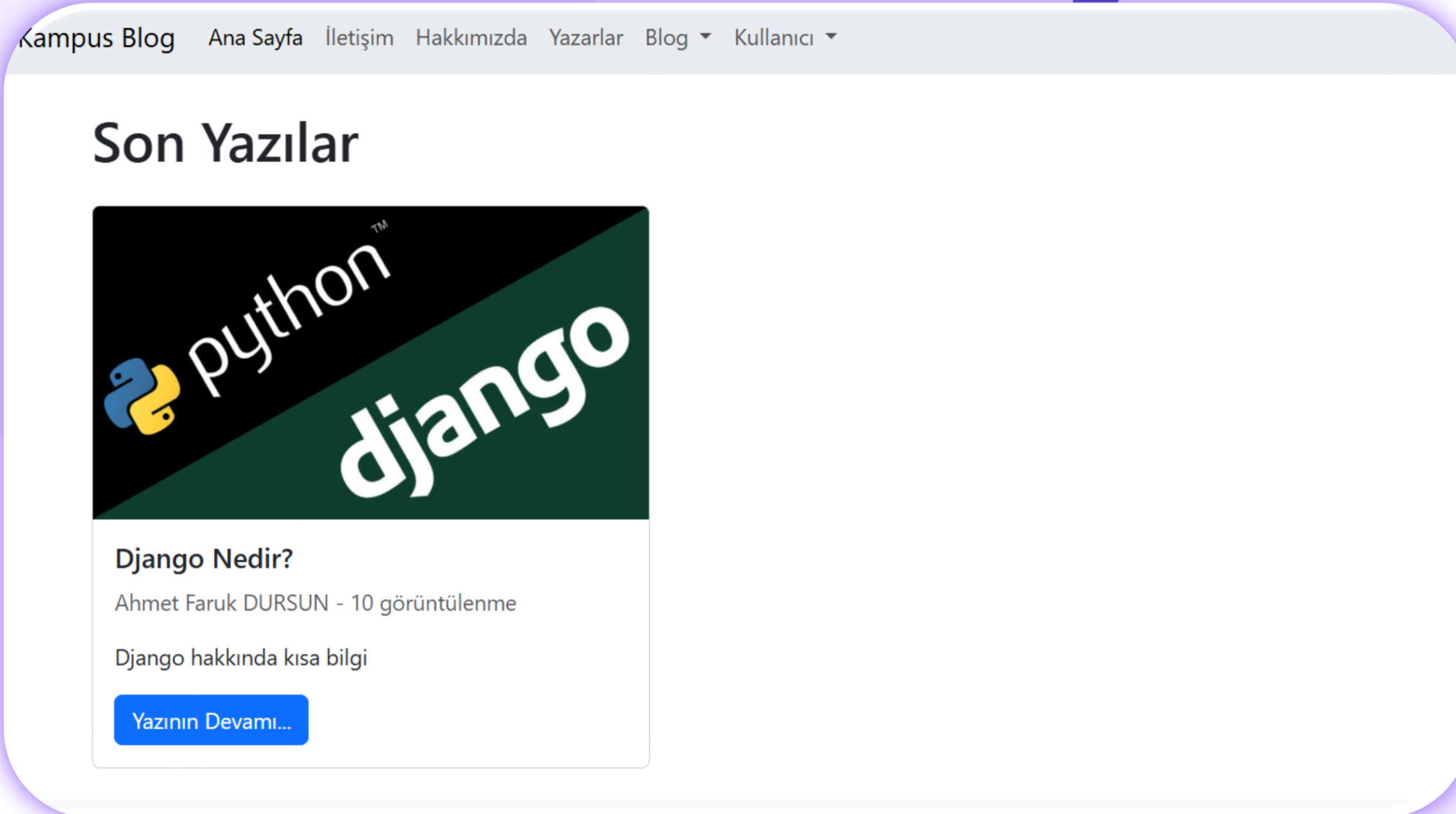


Migrations Oluşturma

Artık verilerimizi kaynak koda yazmıyoruz. Views.py de blogları db den ORM ile çekip ön yüze gönderiyoruz. Db ye yeni veri eklendiğinde ön yüze doğrudan yansıyacak dinamik yapıyı kurmuş olduk.

```
def anasayfa(request):  
  
    bloglar = blog.models.BlogPost.objects.all()  
  
    context = {  
        "bloglar": bloglar  
    }  
  
    return render(request, "anasayfa.html", context)
```

Migrations Oluşturma



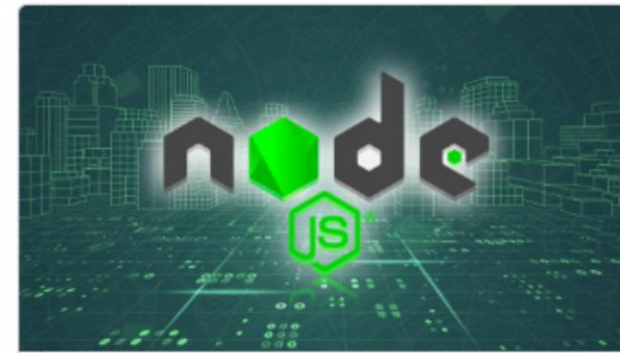
Migrations Oluşturma

```
(venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\kampus_blog> python manage.py shell
```

```
>>> from blog.models import BlogPost
```

```
>>> BlogPost.objects.create(title="Django Nedir?", summary="Django hakkında kısa bilgi",  
content="Django, Python ile yazılmış güçlü bir web framework'tür.", author_name="Ahmet Faruk DURSUN", views=10, static_url="img/django.jpg")
```

Shell'den yeni blog yazıları ekleyiniz.



Node.js ile Sunucu Tarafı Geliştirme

Ahmet Faruk DURSUN - 100 görüntülenme

Node.js kullanarak scalable ve yüksek performanslı sunucu tarafı uygulamalar geliştirme.

[Yazının Devamı...](#)

<html>

HTML5 ve CSS3 ile Modern Web Tasarımı

Elif GÖKÇE - 130 görüntülenme

Modern web tasarımı için HTML5 ve CSS3 teknikleri.

[Yazının Devamı...](#)

HTTP

HTTP Protokolü ve Web Güvenliği

Ahmet Faruk DURSUN - 85 görüntülenme

Web güvenliği ve HTTP protokolü hakkında bilgiler.

[Yazının Devamı...](#)

Son