



Django İle Python Arka Yüz Yazılım Geliştirme

Teknik Bilimler
Meslek Yüksekokulu

Öğr. Gör. Furkan DURMUŞ



Bu Haftanın Ders Kazanımları



Django Proje Yapısı

Proje İskeleti

settings.py - urls.py

Proje Çalışması 2

Proje Çalışması 1



```
require 'capybara/reils'
```

```
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/ 22
```

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

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

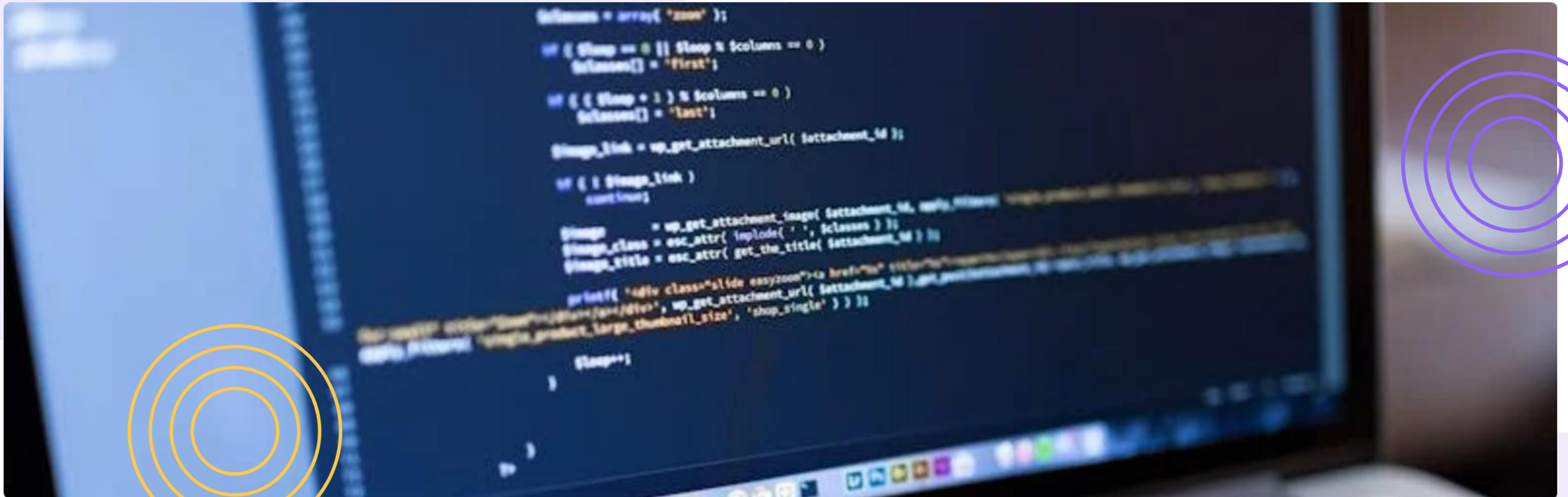
```
27 # in _spec.rb will both be required and 28
```

```
29 It is recommended that you configure 30
```

Django Proje Yapısı

Django Proje Yapısı

“Proje” ile “App” arasındaki fark nedir ?



Django Proje Yapısı

“Proje” ile “App” arasındaki fark nedir ?

<> 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/



Django Proje Yapısı

Proje oluşturma komutu :

django-admin startproject config .

(config ayarların olduğu klasör, config ismi zorunlu değil)

App oluşturma komutu :

python manage.py startapp app_adi



The image shows the Django logo, which is the word "django" in a white, lowercase, sans-serif font, centered on a dark green, semi-circular background.

Django Proje Yapısı

Kampus_blog

<> 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/

Django Proje Yapısı

Kampus_blog

Projenin bulunduğu ana klasör.

- GitHub'a atılacak yerdir.
- manage.py burada olur.
- Django'nun iç yapısı değil, bizim çalışma alanımızdır.

```
«» 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/
```

Django Proje Yapısı

venv



<> 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/

Django Proje Yapısı

venv

Projenin çalıştırıldığı sanal ortamdır.

- Paketler burada bulunur.
- Kod yazılmaz.
- GitHub'a yüklenmez.

```
«» 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/
```

Django Proje Yapısı

manage.py



<> Kod

kampus_blog/

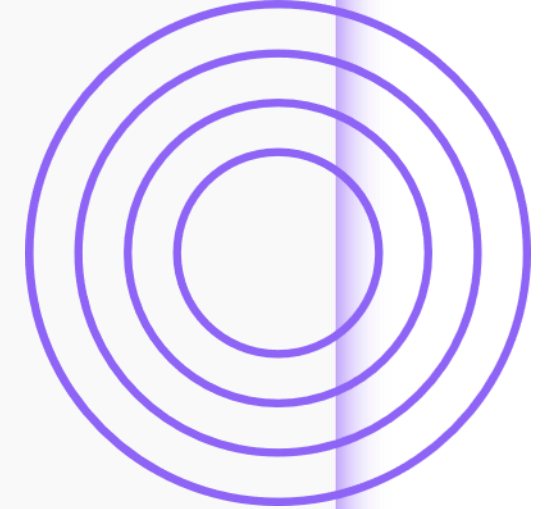
— venv/
— manage.py

— config/

— __init__.py
— settings.py
— urls.py
— asgi.py
— wsgi.py

— core/

— admin.py
— apps.py
— models.py
— views.py
— urls.py
— templates/



Django Proje Yapısı

manage.py ←

python **manage.py** runserver

python **manage.py** startapp blog

- Django komutlarını çalıştırır.
- settings dosyasını yükler.
- Projeyi ayağa kaldırır.

«» 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/

Django Proje Yapısı

config/



<> Kod

kampus_blog/

├── venv/

├── manage.py

├── config/

│ ├── __init__.py

│ ├── settings.py

│ ├── urls.py

│ ├── asgi.py

│ └── wsgi.py

└── core/

│ ├── admin.py

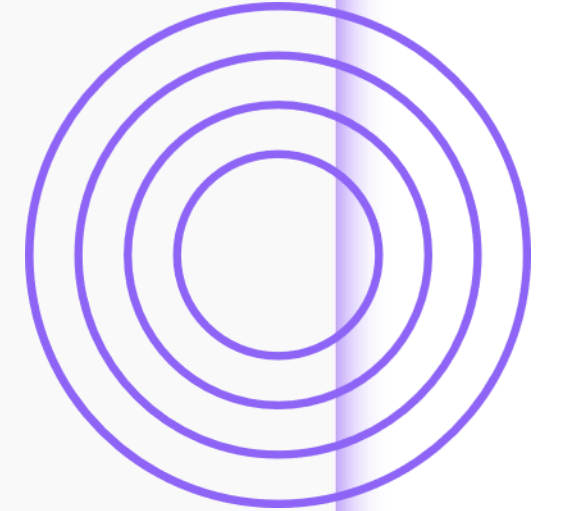
│ ├── apps.py

│ ├── models.py

│ ├── views.py

│ ├── urls.py

│ └── templates/



Django Proje Yapısı

config/ ←

- Bu klasör Django'nun "kontrol merkezi"dir.
- startproject komutu ile oluşur.
- Projenin merkezi ayar ve yönlendirme klasörüdür.
- Config ismi yaygın kullanımdır, isimlendirme özelleştirilebilir.

```
«» 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/
```

Django Proje Yapısı

init.py

- İçinde bulunduğu klasörü Python paketi yapar.
- İçerik yazılmaz.
- Sistem dosyasıdır.

Yani o klasörün Python tarafından import edilebilir olmasını sağlar.

<> 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/

Django Proje Yapısı

asgi.py

- Asenkron yapı için kullanılır.
- Websocket sistemler için gereklidir.

«» 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/

Django Proje Yapısı

wsgi.py

- Sunucuya dağıtım (deployment) için kullanılır.

<> 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

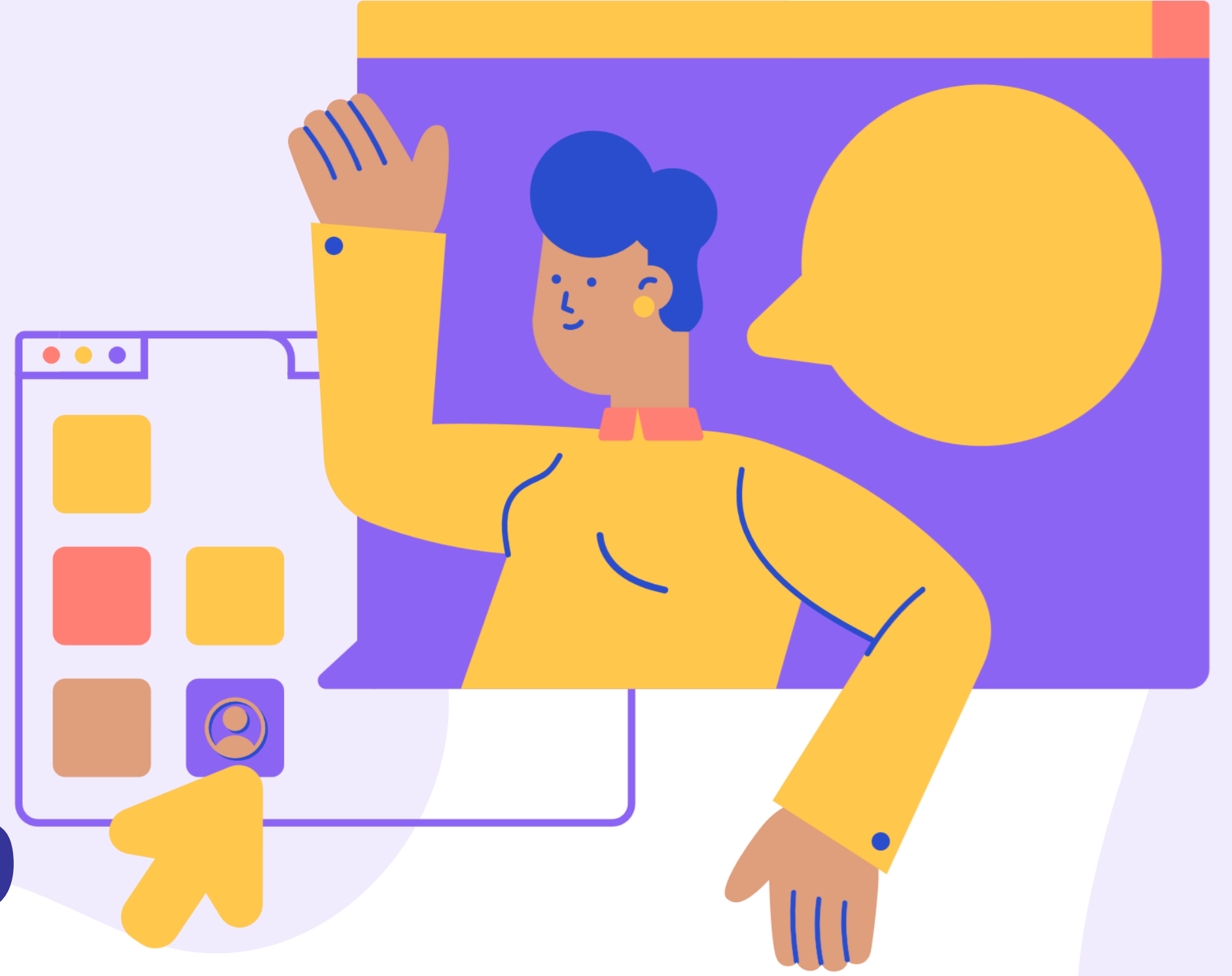
└── templates/



```
require 'capybara/re Rails'  
  
1  
2 Capybara.javascript_driver = :webkit  
3  
4 Category.delete_all; Category.create  
5  
6 Shoulda::Matchers.configure do |config|  
7   config.integrate do |with|  
8     with.test_framework :rspec  
9     with.library :rails  
10  end  
11 end  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25
```

Settings.py

Şuana kadar
settings.py
içerisinde hangi
işlemleri yaptık ?



Django Proje Yapısı

Settings.py

Django projesinin merkezi ayar dosyasıdır.

- Güvenlik
- Uygulamalar
- Veritabanı
- Dil
- Saat
- Static dosyalar

gibi tüm sistem ayarları bulunur.

«» 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/

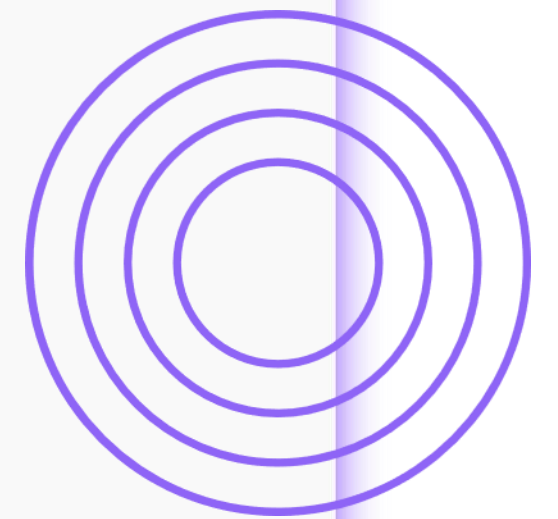
Django Proje Yapısı

Settings.py



```
«» 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/
```



```
from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent
```



Django Proje Yapısı

Settings.py

```
from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent
```

BASE_DIR: Projenin ana klasörünü temsil eder.

Django dosya yollarını buna göre hesaplar.

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/

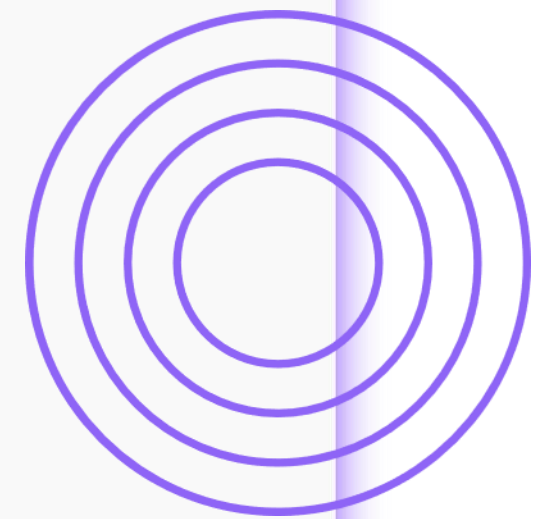
Django Proje Yapısı

Settings.py



```
22 # SECURITY WARNING: keep the secret key used in production secret!  
23 SECRET_KEY = 'django-insecure-t_6xatl5_5^!ar5'  
24
```

```
<> 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/
```



Django Proje Yapısı

Settings.py

```
22 # SECURITY WARNING: keep the secret key used in production secret!  
23 SECRET_KEY = 'django-insecure-t_6xatl5_5^!ar5'  
24
```

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/

- Güvenlik anahtarıdır.
- Şifreleme işlemlerinde kullanılır.
- Asla paylaşılmaz.

Django Proje Yapısı

Settings.py

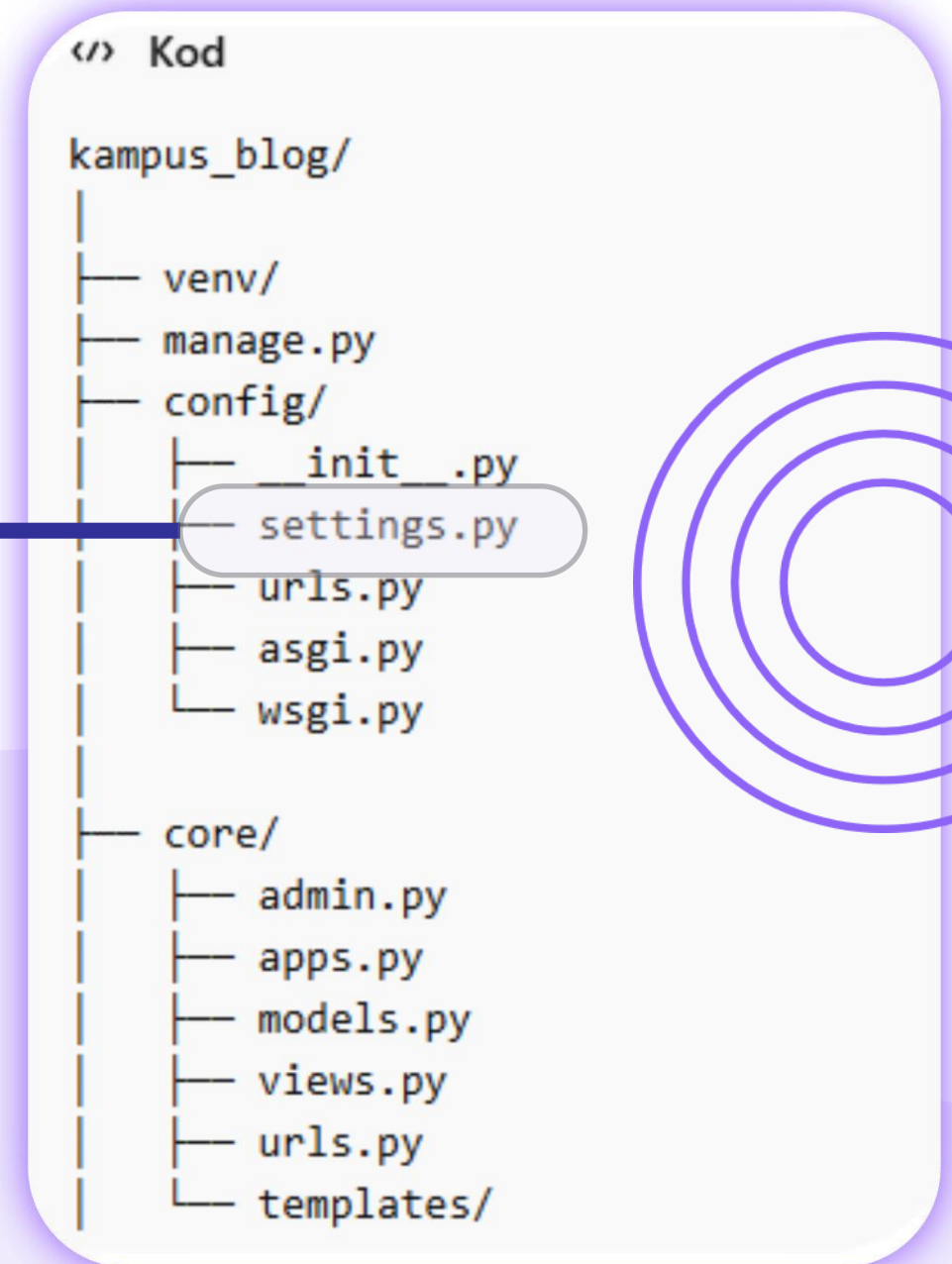
```
24  
25 # SECURITY WARNING:  
26 DEBUG = True  
27
```

```
<> 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/
```

Django Proje Yapısı

Settings.py

```
24  
25 # SECURITY WARNING:  
26 DEBUG = True  
27
```

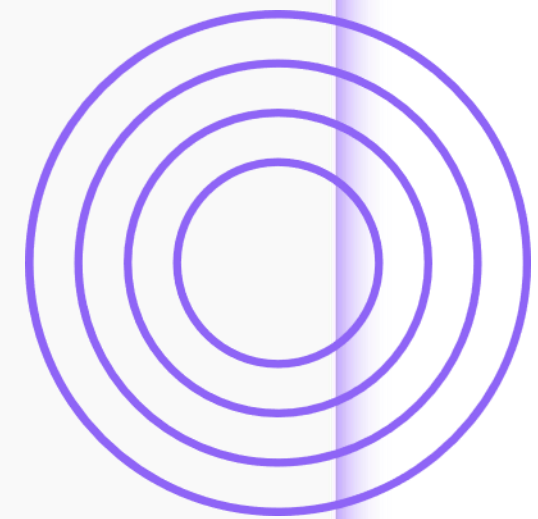
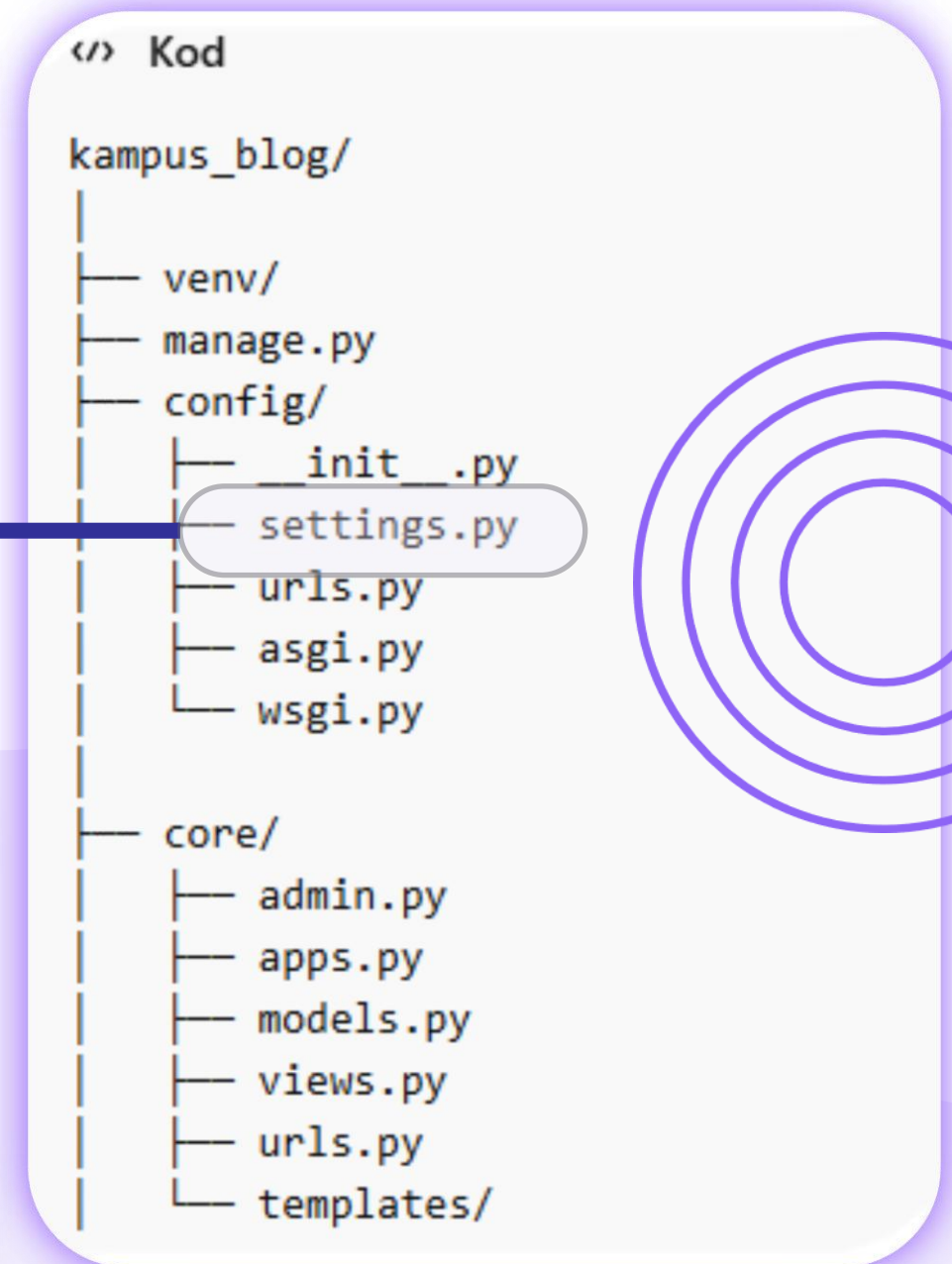


- Projeyi başlatın.
- Serverda web sitesini açın.
- 12.0.0.1:8000/test sayfasını açın.

Django Proje Yapısı

Settings.py

```
24  
25 # SECURITY WARNING:  
26 DEBUG = True  
27
```



- Projeyi başlatın.
- Serverda web sitesini açın.
- 12.0.0.1:8000/test sayfasını açın.



Django Proje Yapısı

Settings.py

```
DEBUG = False  
ALLOWED_HOSTS = ['*']
```

```
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/
```

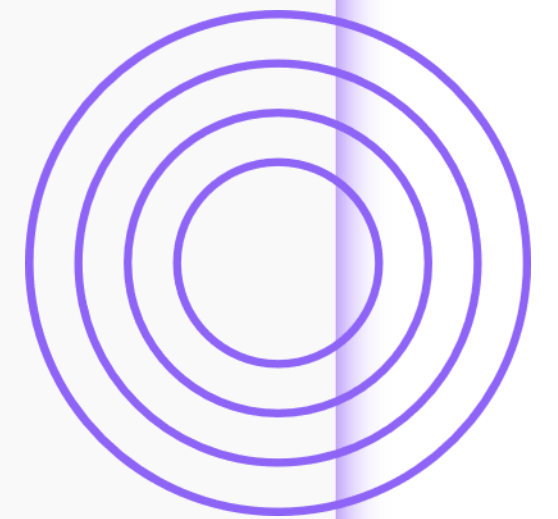
- Debug False yap.
- Allowed host içine '*' ekle.
- Projeyi tekrar başlat.
- /test sayfasını tekrar aç

Django Proje Yapısı

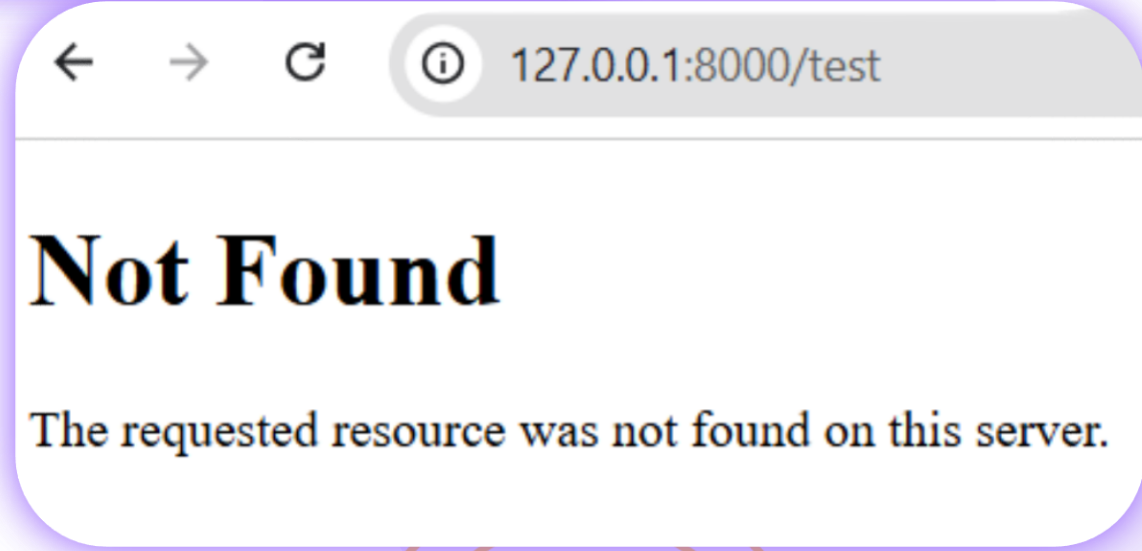
Settings.py

```
DEBUG = False  
ALLOWED_HOSTS = ['*']
```

```
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/
```



- Debug False yap.
- Allowed host içine '*' ekle.
- Projeyi tekrar başlat.
- /test sayfasını tekrar aç
- **Projeyi ilk haline getirmeyi unutmayınız!**



Django Proje Yapısı

Settings.py

```
24  
25 # SECURITY WARNING:  
26 DEBUG = True  
27
```

Debug modu, uygulamanın geliştirme aşamasında çalıştığını belirten moddur.

- Kod hatalarını ekranda açıkça gösterir.
- Geliştirme sırasında sorunları kolay bulmayı sağlar.
- Kod değişince auto reload olur

«» 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/

Django Proje Yapısı

Settings.py

```
ALLOWED_HOSTS = []
```

Hangi domainlerden erişime izin verileceğini belirler.

Debug False olduğunda allowed_host dolu olmalıdır.

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

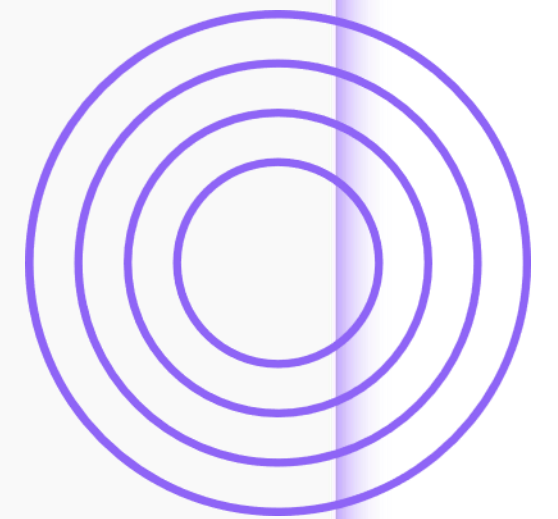
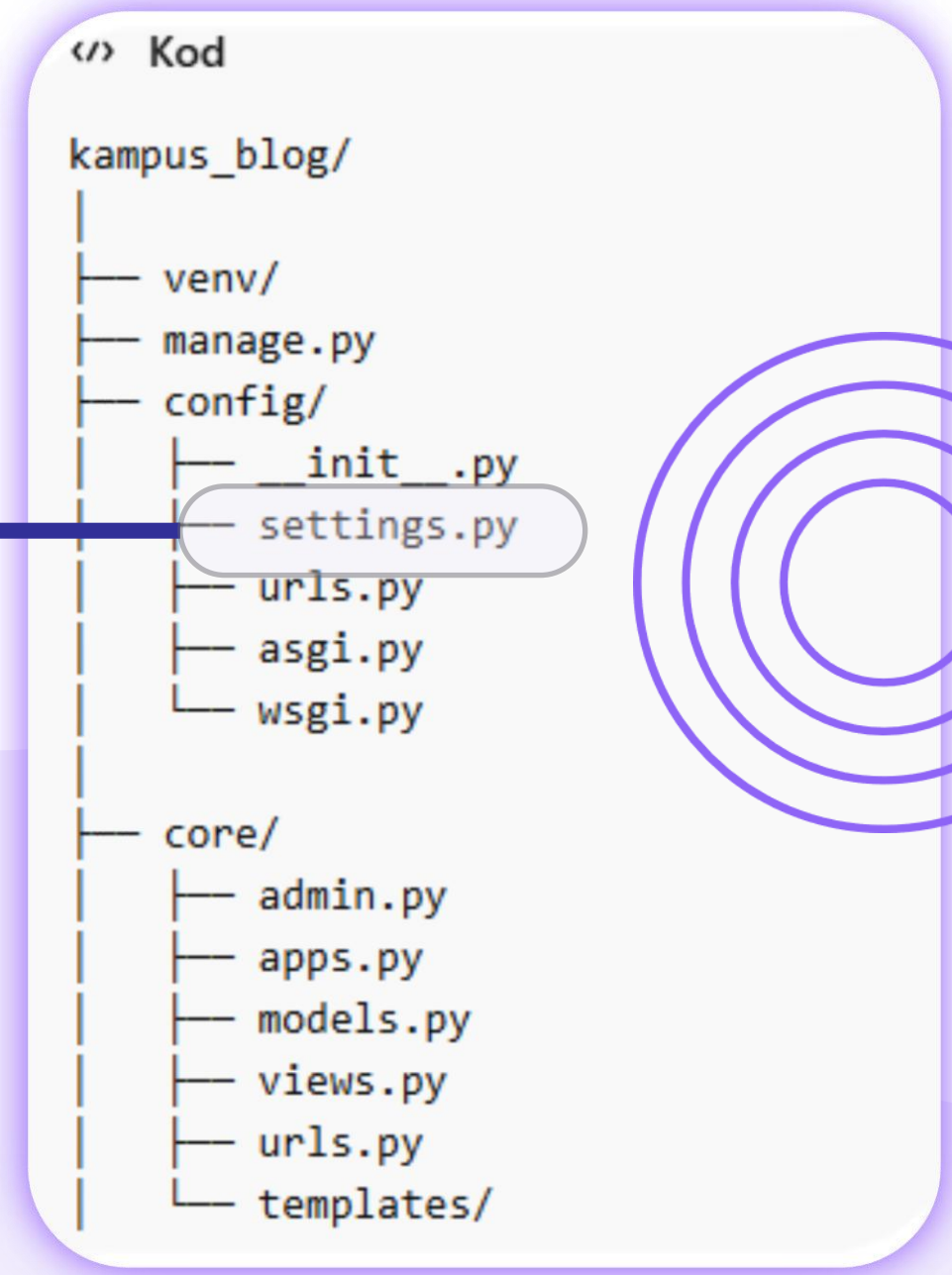
— templates/

Django Proje Yapısı

Settings.py



```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'core'  
]
```



Django Proje Yapısı

Settings.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'core'  
]
```

Projede aktif olan uygulamaları(app'leri) gösterir.
Django açılırken bu listedeki app'leri yükler.

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/
```

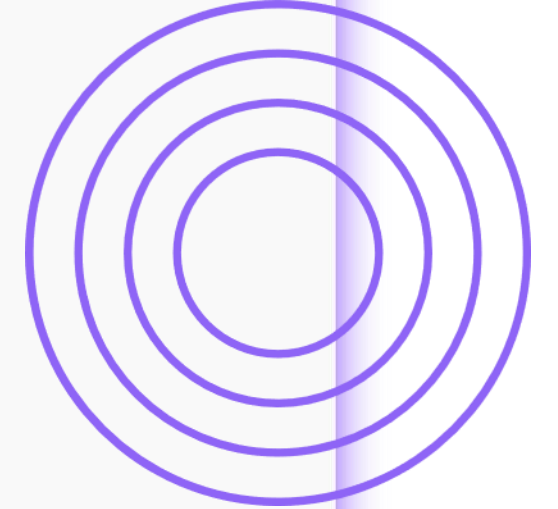
Django Proje Yapısı

Settings.py

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

Projenin veritabanını belirler.

```
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/
```



Django Proje Yapısı

Settings.py



```
LANGUAGE_CODE = 'tr-tr'
```

```
TIME_ZONE = 'UTC'
```

<> Kod

kampus_blog/

├── venv/

├── manage.py

├── config/

│ ├── init .py

│ └── settings.py

├── urls.py

├── asgi.py

└── wsgi.py

├── core/

│ ├── admin.py

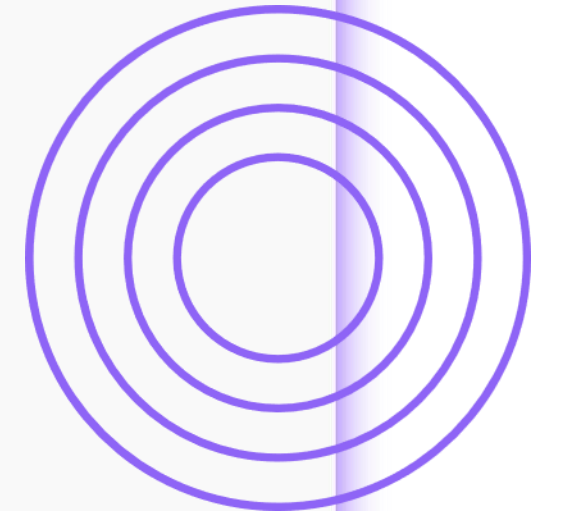
│ ├── apps.py

│ ├── models.py

│ ├── views.py

│ ├── urls.py

└── templates/





```
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/ and its subdirectories. 22
```

```
23 # spec/support/ and its subdirectories. These files will be loaded 24
```

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

urls.py

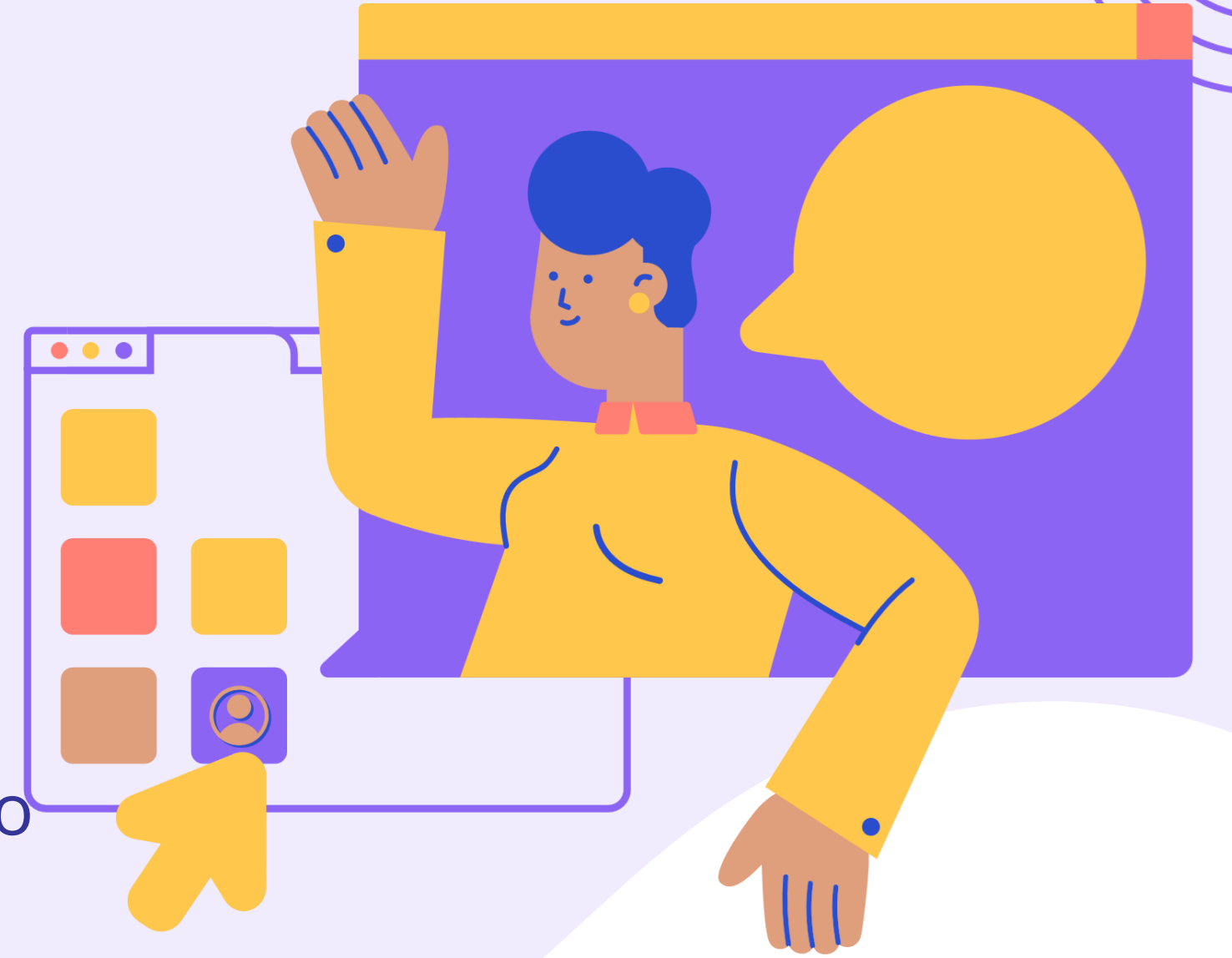


url nedir ?



url nedir ?

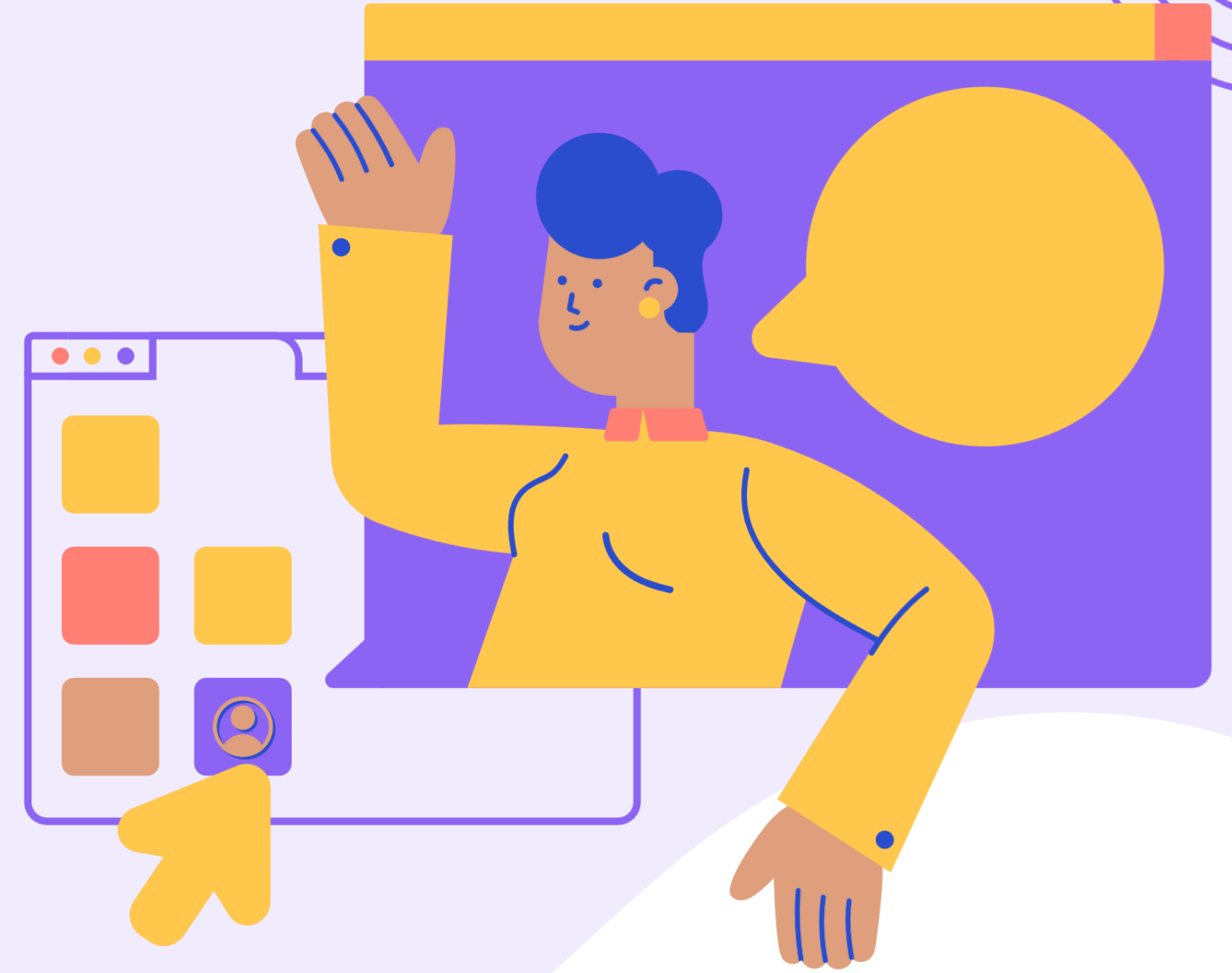
Tarayıcının "nereye gideceğini" söyleyen adres.
internet üzerindeki bir kaynağın (web sayfası, resim, video
veya dosya) tam "adresidir".





url nedir ?

Django'nun görevi girilen URL isteğini doğru view'a yönlendirmektir.



Django Proje Yapısı

urls.py

Tarayıcı
↓
config/urls.py
↓
(app varsa) include()
↓
app/urls.py
↓
View

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/
```

Django Proje Yapısı

urls.py

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', include('core.urls')),  
]
```

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/
```

Bu dosya: Projenin ana trafik merkezidir.

Django Proje Yapısı

urls.py



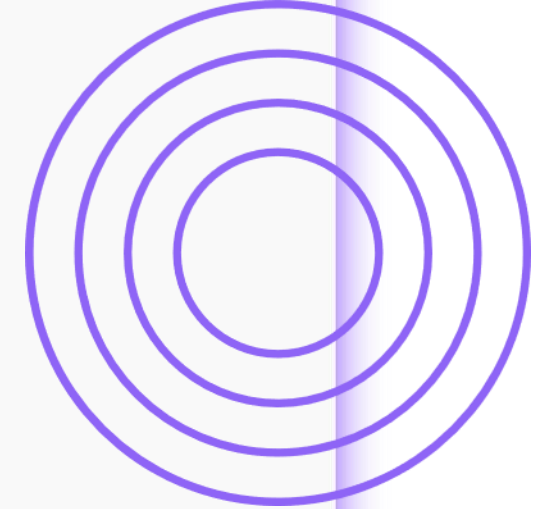
Furkandurmus.com/kayit/yenikayit

Config içerisindeki url yapısı

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', include('core.urls')),  
    path('kayit/', include('kayit.urls')),  
]
```

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/
```



Django Proje Yapısı

urls.py

Furkandurmus.com/kayit/yenikayit

Config içerisindeki url yapısı

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', include('core.urls')),  
    path('kayit/', include('kayit.urls')),  
]
```

kayit APP ime gidecek

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/
```

Django Proje Yapısı

urls.py

Furkandurmus.com/kayit/yenikayit

Config içerisindeki url yapısı

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', include('core.urls')),  
    path('kayit/', include('kayit.urls')),  
]
```

Kayit APP içerisindeki url yapısı

```
urlpatterns = [  
    path('', views.anasayfa, name='anasayfa'),  
    path('yenikayit/', views.iletisim, name='yenikayit'),  
    path('kayitsil/', views.kayitsil, name='kayitsil'),  
    path('kayitguncelle/', views.kayitguncelle, name='kayitguncelle'),  
]
```

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/
```

Kayit app içerisindeki urllerden eşleşen url'e gidecek.

Django Proje Yapısı

urls.py

```
urlpatterns = [  
    path('', views.anasayfa, name='anasayfa'),  
    path('iletisim/', views.iletisim, name='iletisim'),  
]
```

Applerin içerisindeki name kısmı ilgili url'in açtığı sayfanın çağırılacağı addır. Doğru isimlendirme önemlidir.

Front tarafında bir sayfadan başka bir sayfaya geçmek istediğimizde bu isimlendirmeleri kullanacağız.

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/
```



Proje Çalışması

```
require 'capybara/reils'
```

```
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 # Requires supporting ruby files with support/ and its subdirectories. This can be used to
```

```
21 # spec/support/ and its subdirectories. This can be used to
```

```
22 # run as spec files by default. This can be used to
```

```
23 # in _spec.rb will both be required and run as spec files by default. This can be used to
```

```
24 It is recommended that you use the following configuration:
```



Kampus_blog

```
anasayfa.html x iletisim.html urls.py config urls.py core
core > templates > anasayfa.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 |   <title>Kampüs blog</title>
5 </head>
6 <body>
7 |   <h1>Kampüs Blog Sitesine Hoş Geldiniz..</h1>
8 |   <a href="{% url 'iletisim' %}"><button>İletişim</button></a>
9 </body>
10 </html>
11
12 |
13
14
```

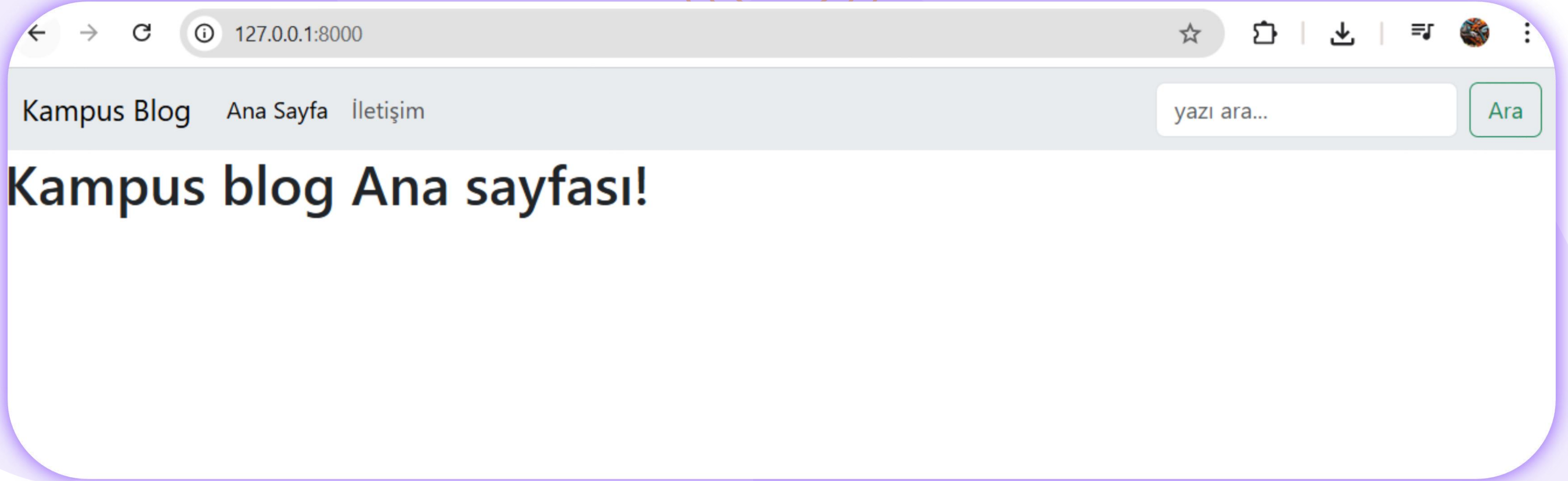
Üstteki kodu yorumlayınız.

Kampus_blog

Kendi projenizde anasayfadan iletişim sayfasına, iletişim sayfasından ana sayfaya geçecek basit bir buton ekleyiniz.



Kampus_blog



Bootstrap kullanarak her iki sayfa için navbar tasarımı yapınız ve sayfa geçişlerini Django template tag { } ile 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 # Requires supporting ruby files with support/
```

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

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

```
23 # in _spec.rb will both be required and
```

```
24 # It is recommended that you configure
```

Proje iskeleti

Kampus_blog

Yandaki yapıyı projenizde kurunuz.

```
├── core/      ← Genel sayfalar App i
│   ├── views.py
│   ├── urls.py
│   └── templates/
│       ├── anasayfa.html
│       ├── hakkimizda.html
│       └── iletisim.html
├── blog/      ← Blog App i
│   ├── models.py
│   ├── views.py
│   ├── urls.py
│   └── templates/
│       ├── blog_list.html
│       ├── blog_detay.html
│       ├── blog_ekle.html
│       ├── blog_duzenle.html
│       └── blog_sil.html
└── kullanıcı/ ← Kullanıcı App i
    ├── views.py
    ├── urls.py
    └── templates/
        ├── girisyap.html
        ├── kayitol.html
        ├── profil.html
        └── cikisyap.html
```

Son