Django İle Python Arka Yüz Yazılım Geliştirme

Teknik Bilimler
Meslek Yüksekokulu

Öğr. Gör. Furkan DURMUŞ

Bu Haftanın Ders Kazanımları

- Blog Detay Sayfası
- 404 Sayfası
- Detay Sayfasına Geçiş

Furkandurmus.com
-> Dersler
-> Django
-> Ders Uygulamaları
-> Blog Proje Kodları

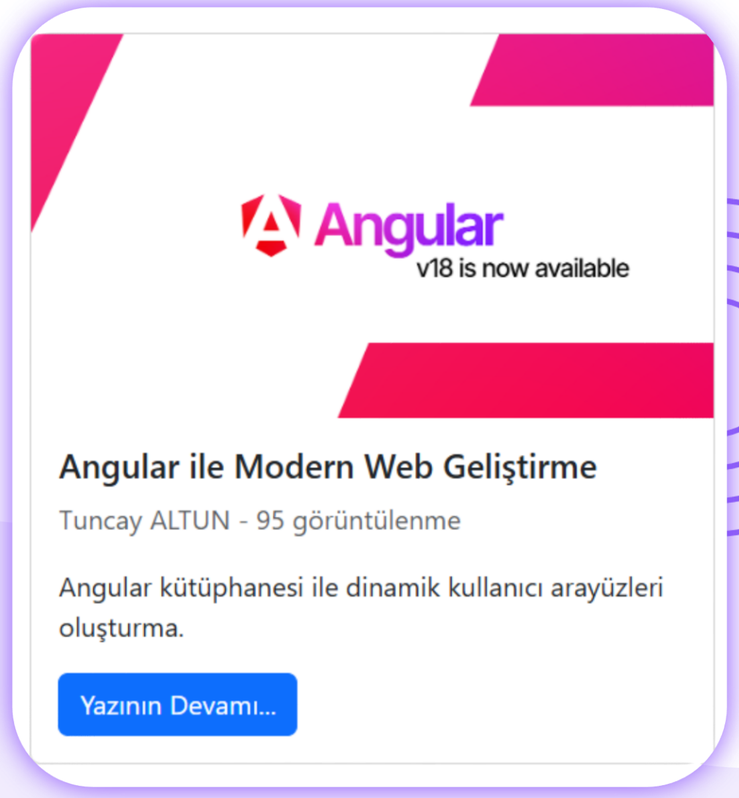
Üzerinden projenizi güncelleyebilirsiniz...


require 'capybara/rails'
Capybara.javascript_driver
Category.delete_all;
Shoulda::Matchers.configure
config.integrate
with.test_framework
with.library :rails
end
end
Add additional requires
Requires supporting
spec/support/ and its
run as spec files by default.
in _spec.rb will
It is recommended

Blog Detay

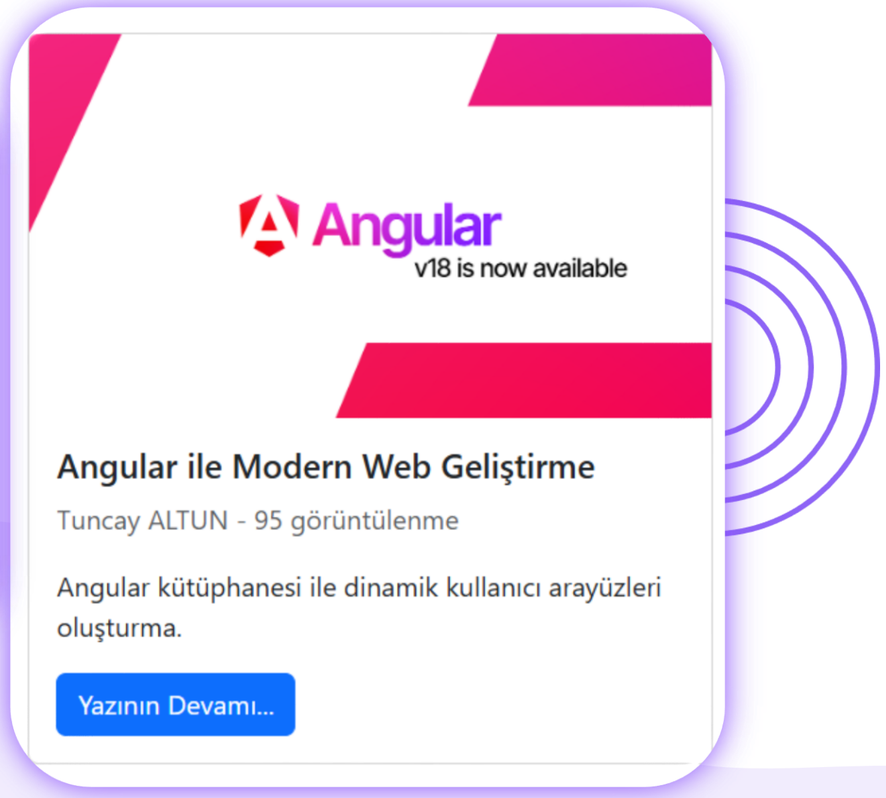
Blog Detay Sayfası

‘Yazının Devamı’ butonu aktif çalışıyor mu, Neden ?



Blog Detay Sayfası

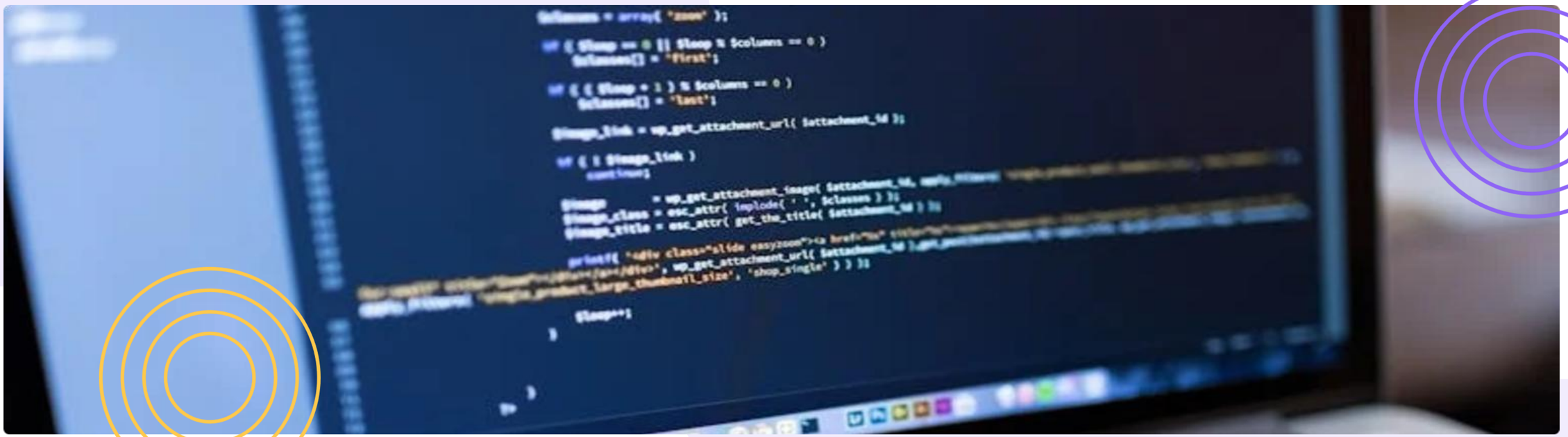
```
{% for post in bloglar %}
<div class="col-md-4 mb-4">
    <div class="card h-100">
        <img src="{% static post.static_url %}" class="card-img-top" alt="Blog">
        <div class="card-body">
            <h5 class="card-title">{{ post.title }}</h5>
            <p class="text-muted">{{ post.author_name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



Blog Detay Sayfası

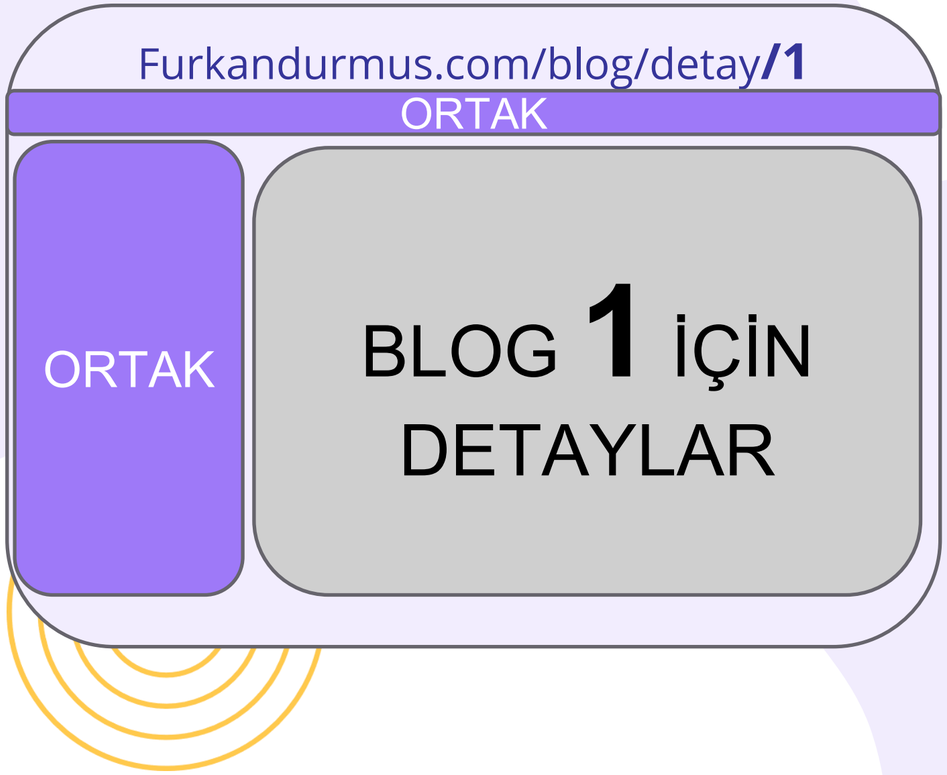
Anasayfa, Yazarlar, İletişim vs. her biri için .html sayfası oluşturduk.

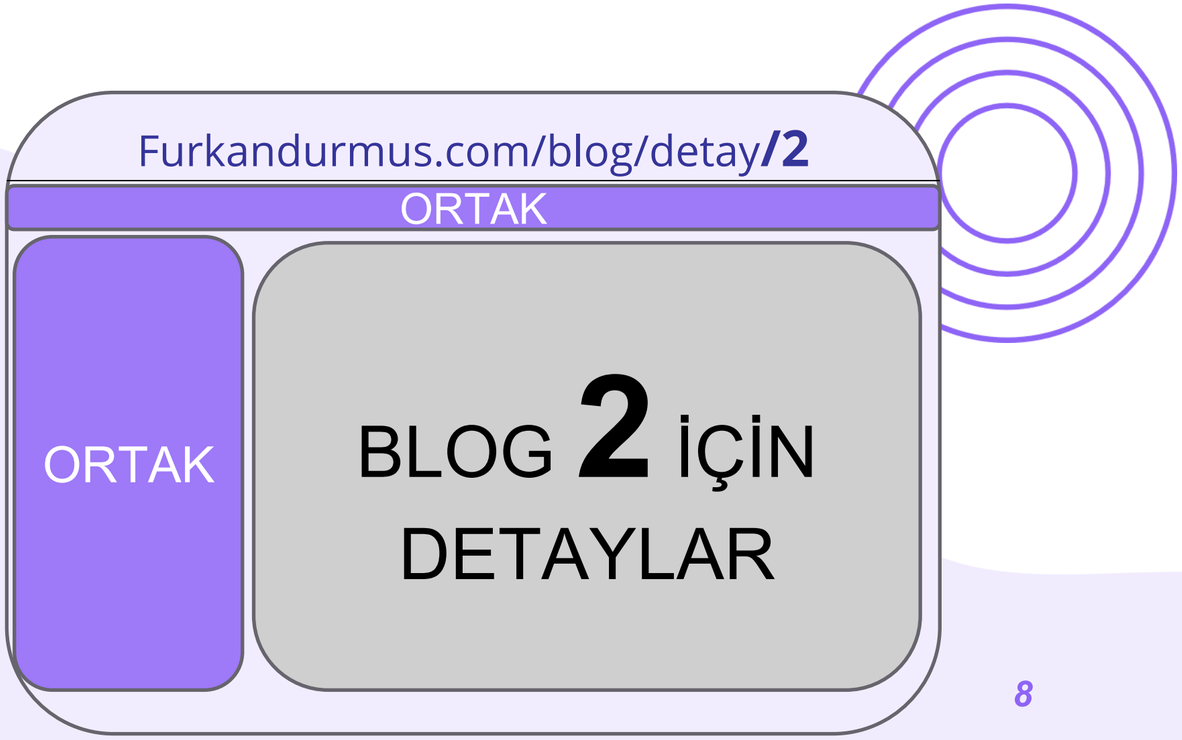
Her blog için bir html oluşturmamıza gerek var mı ?



Dinamik Url

Tek bir html sayfası dinamik bir yapı.





Dinamik Url Tanımlama

```
blog > urls.py > ...
1   from django.urls import path
2   from . import views
3
4   app_name = 'blog'
5
6   urlpatterns = [
7       path('', views.blog_anasayfa, name='blog_anasayfa'),
8       path('detay/<int:id>/', views.blog_detay, name='blog_detay'),
9       path('ekle/', views.blog_ekle, name='blog_ekle'),
10      path('duzenle/', views.blog_duzenle, name='blog_duzenle'),
11      path('sil/', views.blog_sil, name='blog_sil'),
12  ]
```

Blog -> urls.py İçerisindeki blog_detay url yolunu

path('detay/**<int:id>/**',views.blog_detay,name='blog_detay')

Olarak güncelleyiniz.

Dinamik Url Tanımlama

```
path('detay/<int:id>/', views.blog_detay, name='blog_detay'),
```

< > nedir ?

Bu işaret django url'inde bir değişken değerin geleceği manasına gelir.

Böylece kullanıcı her seferinde farklı bir değer gönderebilir.

Dinamik Url Tanımlama

```
path('detay/<int:id>/', views.blog_detay, name='blog_detay'),
```

< > nedir ?

Bu işaret django url'inde bir değişken değerin geleceği manasına gelir.

Böylece kullanıcı her seferinde farklı bir değer gönderebilir.

int nedir ?

Gelecek olan değişkenin tipini belirler.

Dinamik Url Tanımlama

```
path('detay/<int:id>/', views.blog_detay, name='blog_detay'),
```

< > nedir ?

Bu işaret django url'inde bir değişken değerin geleceği manasına gelir.

Böylece kullanıcı her seferinde farklı bir değer gönderebilir.

int nedir ?

Gelecek olan değişkenin tipini belirler.

id nedir ?

Gelecek olan değişkene verilen isim.

Dinamik Url Tanımlama

```
active{% endif %}" href="{% url 'blog:blog_sil' %}">Blog Sil</a></li>
<li><a class="dropdown-item {% if request.resolver_match.url_name == 'blog_detay' %}
active{% endif %}" href="{% url 'blog:blog_detay' %}">Blog Detay</a></li>
```

Templates / partials / navbar'da bulunan 37. satırdaki blog detay <li> dropdown'ının tüm satırını silin.

Sizce bu silme işlemini neden yaptık ?

Dinamik Url Tanımlama

```
active{% endif %}" href="{% url 'blog:blog_sil' %}">Blog Sil</a></li>
<li><a class="dropdown-item {% if request.resolver_match.url_name == 'blog_detay' %}
active{% endif %}" href="{% url 'blog:blog_detay' %}">Blog Detay</a></li>
```

Templates / partials / navbar'da bulunan 37. satırdaki blog detay dropdown'ını silin.
Sizce bu silme işlemini neden yaptık ?

Artık statik bir detay sayfamız olmayacak. Her defasında dinamik şekilde tıkladığımız bloğun id sine göre ilgili detay sayfasını açacak. Buradaki sabit bir sayfayı açmaya çalışacağı için hata verecek. Bu sebeple bu satırı kaldırdık.

Dinamik Url Tanımlama

```
blog > views.py > ...
1   from django.shortcuts import render
2   import blog.models
3
```

Blog modelleri ile işlem yapacağımız için önce blog / views.py ‘ye blog modellerini import etmeliyiz.

Dinamik Url Tanımlama

```
def blog_detay(request, id):
```

Şimdi **blog/views.py** içindeki **blog_detay** fonksiyonunu adım adım güncelleyelim.

Fonksiyon mevcutta request parametresi alıyordu, artık hem request hem **id** parametresi alacak. Çünkü urls.py den **id** gönderilecek.

Dinamik Url Tanımlama

```
def blog_detay(request, id):

    yazi = blog.models.BlogPost.objects.get(id=id)
```

Bir önceki dersimizde **all** ile tüm potları çekmiştik. Şuan benim tek bir post'a ihtiyacım var. Bu sebeple **get** fonksiyonu ile id'si benim gönderdiğim id'ye eşit olan **(id = id)** veriyi çekmesini sağlıyorum.

Dinamik Url Tanımlama

```
def blog_detay(request, id):

    yazi = blog.models.BlogPost.objects.get(id=id)

    context = {
    'yazi': yazi
    }
```

Database'den getirdiğimiz yazıyı önyüz'e göndermek için bir context içerisine alıyorum.

Dinamik Url Tanımlama

```
def blog_detay(request, id):

    yazi = blog.models.BlogPost.objects.get(id=id)

    context = {
    'yazi': yazi
    }
   return render(request, 'detay.html', context)
```

Blog detay html'ine context'i gönderecek şekilde fonksiyonu sonlandırıyorum.

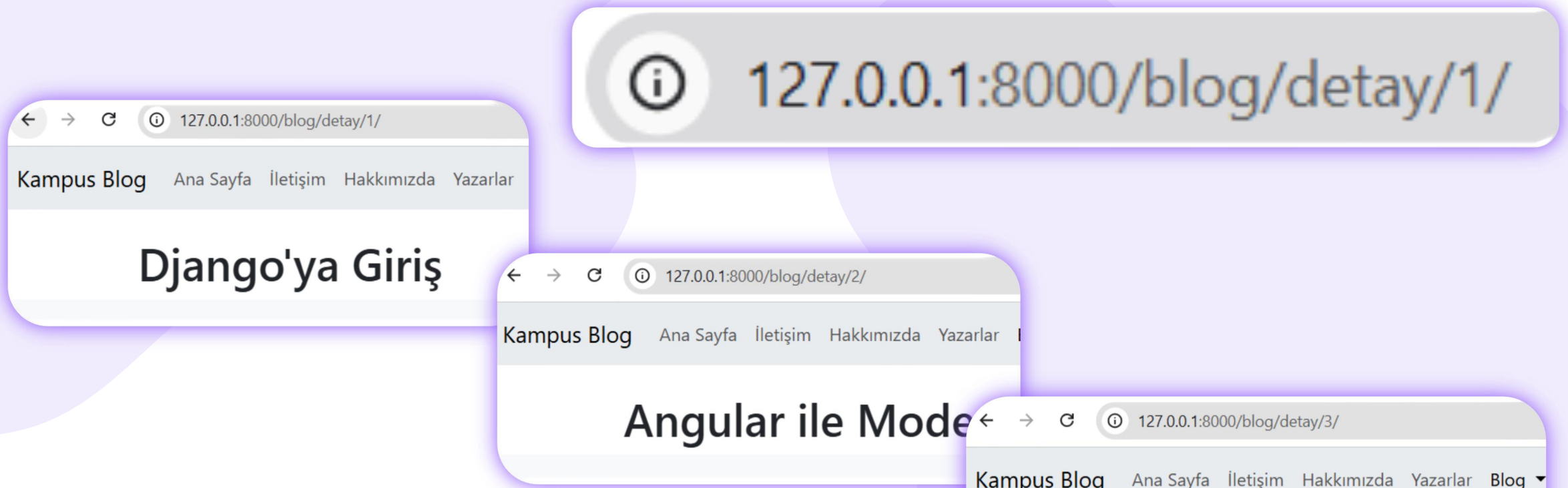
Dinamik Url Tanımlama

Tam tasarımı yapmadan önce test amaçlı blog/templates/detay.html'i yandaki gibi düzeltin ve projeyi çalıştırınız.

blog > templates > detay.html > ...

```
1   {% extends "base.html" %}
2
3   {% block baslik %}Detay{% endblock %}
4
5   {% block icerik %}
6
7   <h1>{{ yazi.title }}</h1>
8
9   {% endblock %}
10
11
```

Dinamik Url Tanımlama



url'den .../blog/detay/... sayfasını açtığınızda id değiştikçe sayfa içeriğinin de değiştiğini göreceksiniz.

Dinamik Url Tanımlama

Çalışma mantığı özet :

```
urlpatterns = [
    path('detay/<int:id>/', views.blog_detay, name='blog_detay'),
```

Dinamik Url Tanımlama

Çalışma mantığı özet :

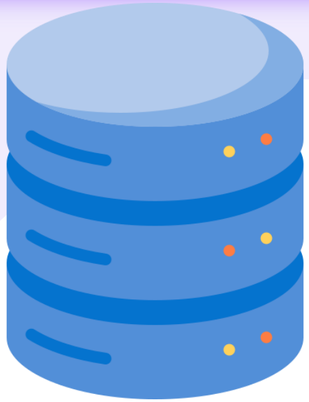
```
urlpatterns = [
    path('detay/<int:id>/', views.blog_detay, name='blog_detay'),
```

```
def blog_detay(request, id):

    yazi = blog.models.BlogPost.objects.get(id=id)

    context = {
    'yazi': yazi
    }
    return render(request, 'detay.html', context)
```

Dinamik Url Tanımlama

Çalışma mantığı özet :

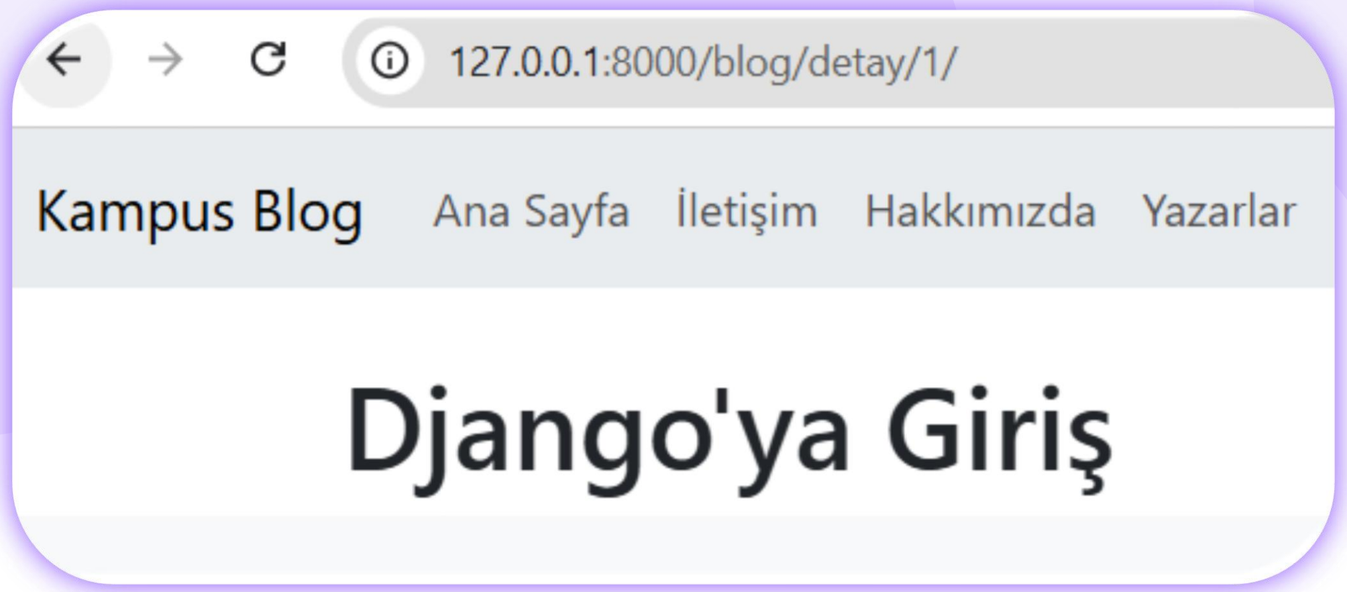
```
urlpatterns = [
    path('detay/<int:id>/', views.blog_detay, name='blog_detay'),
```



```
def blog_detay(request, id):

    yazi = blog.models.BlogPost.objects.get(id=id)

    context = {
    'yazi': yazi
    }
    return render(request, 'detay.html', context)
```

Dinamik Url Tanımlama

Çalışma mantığı özet :




```python
urlpatterns = [
    path('detay/<int:id>/', views.blog_detay, name='blog_detay'),
```

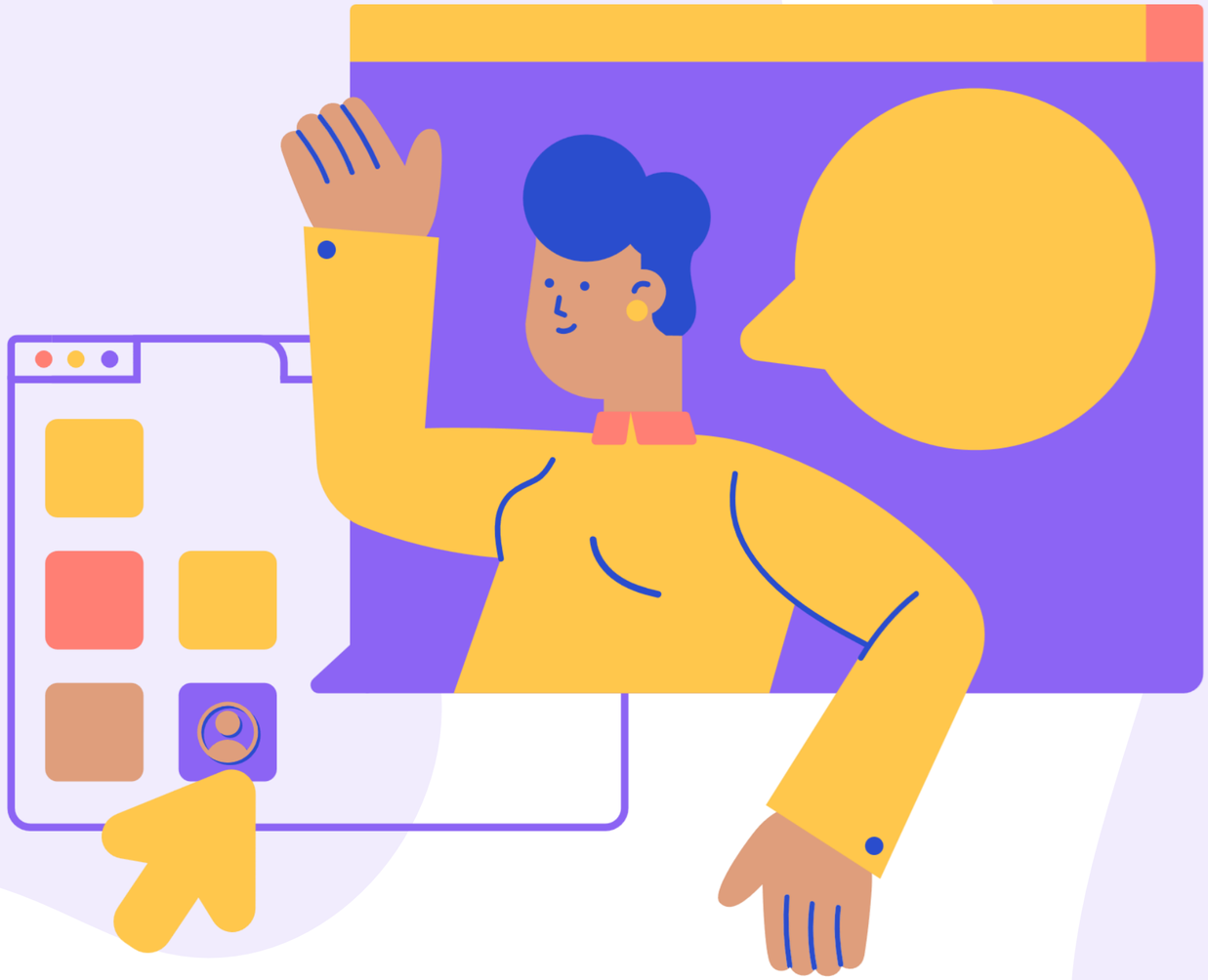
```python
def blog_detay(request, id):

    yazi = blog.models.BlogPost.objects.get(id=id)

    context = {
    'yazi': yazi
    }
    return render(request, 'detay.html', context)
```

blog > templates > detay.html > ...

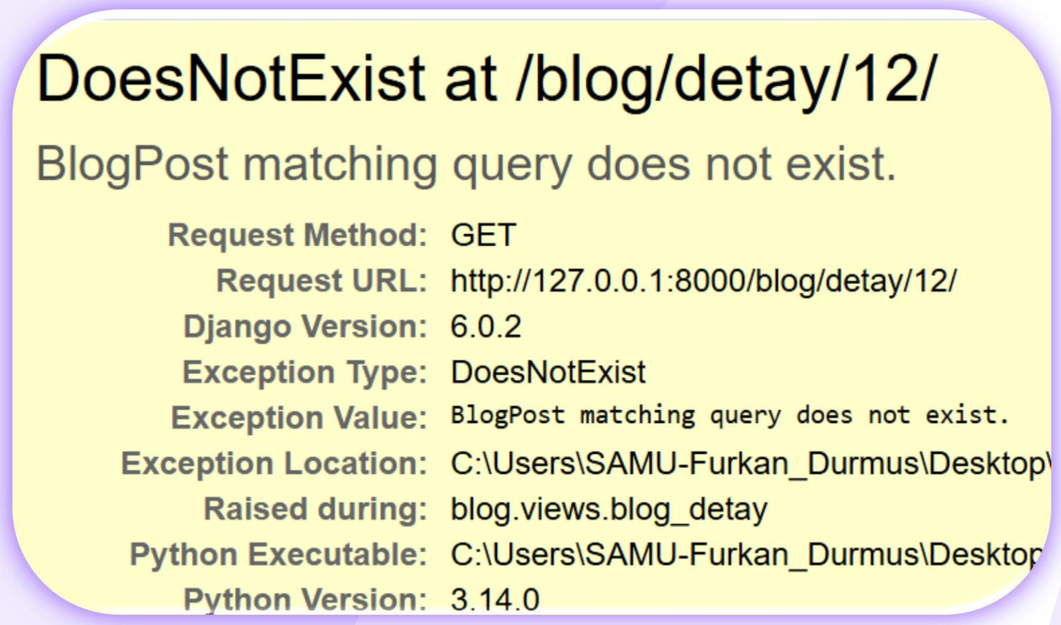
```
{% extends "base.html" %}

{% block baslik %}Detay{% endblock %}

{% block icerik %}

<h1>{{ yazi.title }}</h1>

{% endblock %}
```

127.0.0.1:8000/blog/detay/1/

Detay url'ine olmayan bir id girersek ne olur ?



Bunu nasıl önleriz ?




require 'capybara/rails'
Capybara.javascript_driver
Category.delete_all;
Shoulda::Matchers.
config.integrate
with.test_framework
with.library :rails
end
end
Add additional
Requires supporting
spec/support/ and its
run as spec files by
in _spec.rb will
It is

404 sayfası

404 Sayfası

```
def blog_detay(request, id):

    yazi = blog.models.BlogPost.objects.get(id=id)

    context = {
    'yazi': yazi
    }
  return render(request, 'detay.html', context)
```

İstisnai bir hata (**EXCEPTION**) alınma ihtimali olan satır neresi ?

404 Sayfası

```
def blog_detay(request, id):
    yazi = blog.models.BlogPost.objects.get(id=id)
    context = {
    'yazi': yazi
    }
    return render(request, 'detay.html', context)
```

O halde ilgili sayırı bir try except bloğu içine alıp, db'de yazı ararken hata alması durumunda 404 sayfasını açacak bir yapı kurulması gerekmektedir.

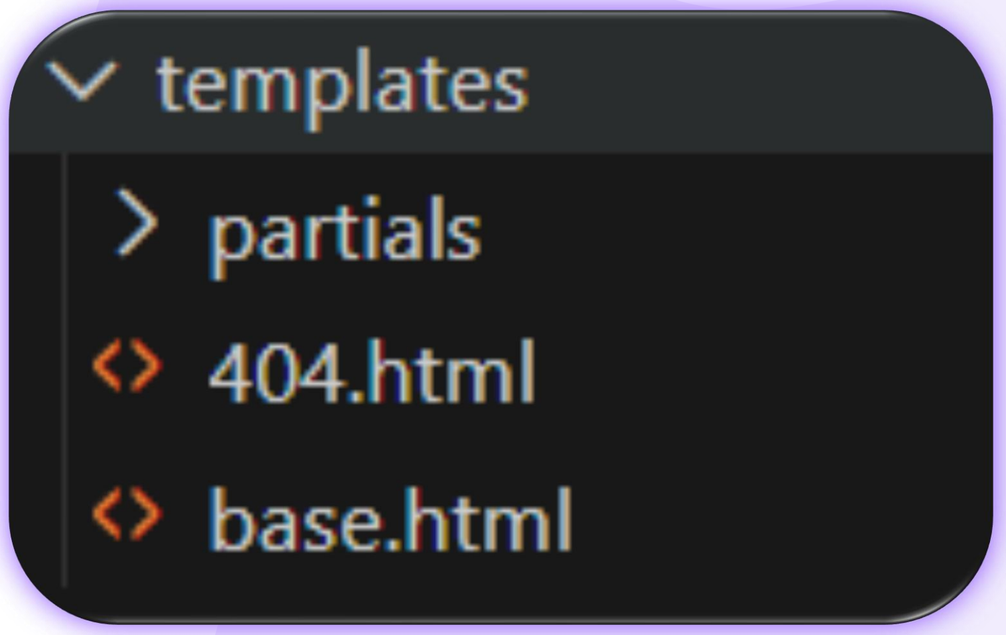
404 Sayfası

```python
def blog_detay(request, id):

    try:
        yazi = blog.models.BlogPost.objects.get(id=id)

    except:
        return render(request, "404.html", status=404)


    context = {
    'yazi': yazi
    }
    return render(request, 'detay.html', context)
```

Blog_detay fonksiyonunu üstteki gibi güncelleyin.

404 Sayfası

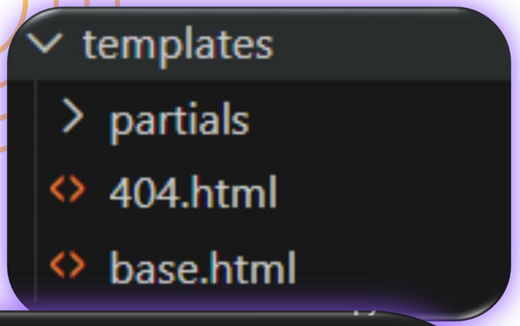
404.Html sayfasına yönlenmesi gerektiği belirtildi fakat henüz 404.html sayfamız yok. Bu sebeple Ana projedeki templates altına yani base.html'in yanına 404.html sayfası oluşturun.



404 Sayfası

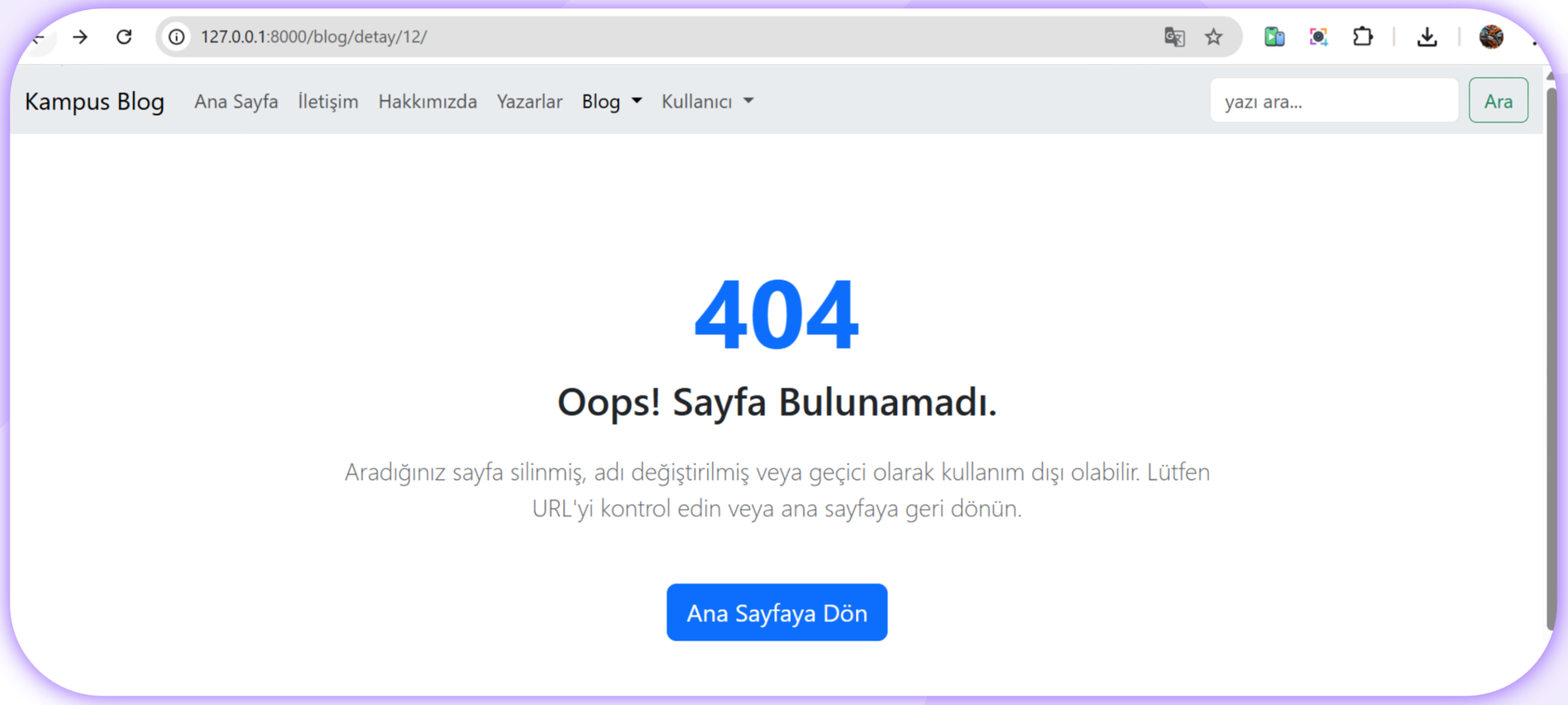
Sayfanın tasarımını yapay zeka araçlarına yaptırabilirsiniz.

- Sayfa tasarımının standart olması için base html'i extend etmeyi unutmayınız.
- 404 sayfa kodları 'block içerik' içinde olmalı.




```
{% extends "base.html" %}
{% load static %}
{% block baslik %}404 - Sayfa Bulunamadı{% endblock %}

{% block icerik %}
<div class="container mt-5 mb-5 text-center">
    <div class="row justify-content-center">
        <div class="col-md-8 py-5">
            <h1 class="display-1 fw-bold text-primary">404</h1>
            <h2 class="mb-4">Oops! Sayfa Bulunamadı.</h2>

            <p class="lead text-muted mb-5">
                Aradığınız sayfa silinmiş, adı değiştirilmiş veya geçici olarak kullanım dışı olabilir.
                Lütfen URL'yi kontrol edin veya ana sayfaya geri dönün.
            </p>

            <a href="/" class="btn btn-primary btn-lg">
                Ana Sayfaya Dön
            </a>
        </div>
    </div>
</div>
{% endblock %}
```

404 Sayfası



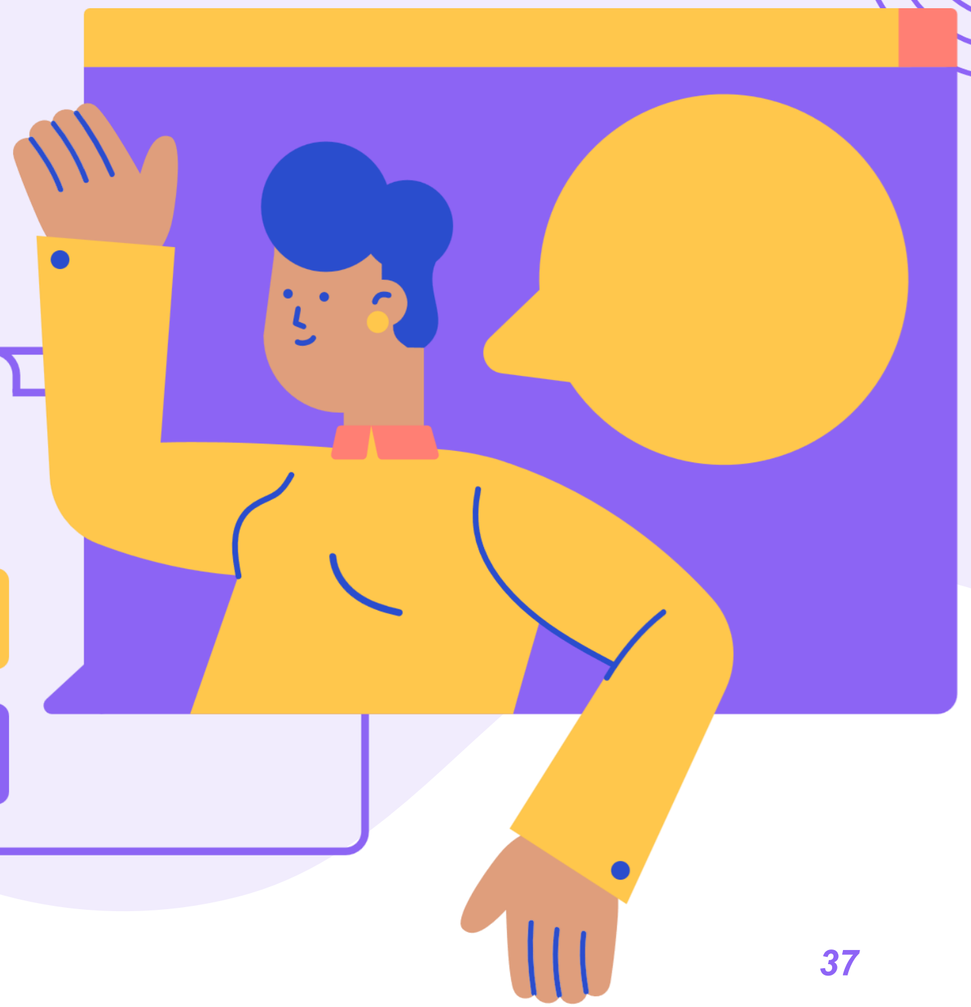
Sonuç

Detay Sayfasına Geçiş

Navbar'da dropdown url'leri (a href) nasıl tanımladık ?

Navbar url'leri

```
ndif %}" href="{% url 'blog:blog_anasayfa' %}

 %}" href="{% url 'blog:blog_ekle' %}">Yeni Blog

dif %}" href="{% url 'blog:blog_duzenle' %}">Blog

%}" href="{% url 'blog:blog_sil' %}">Blog Sil</a></
```

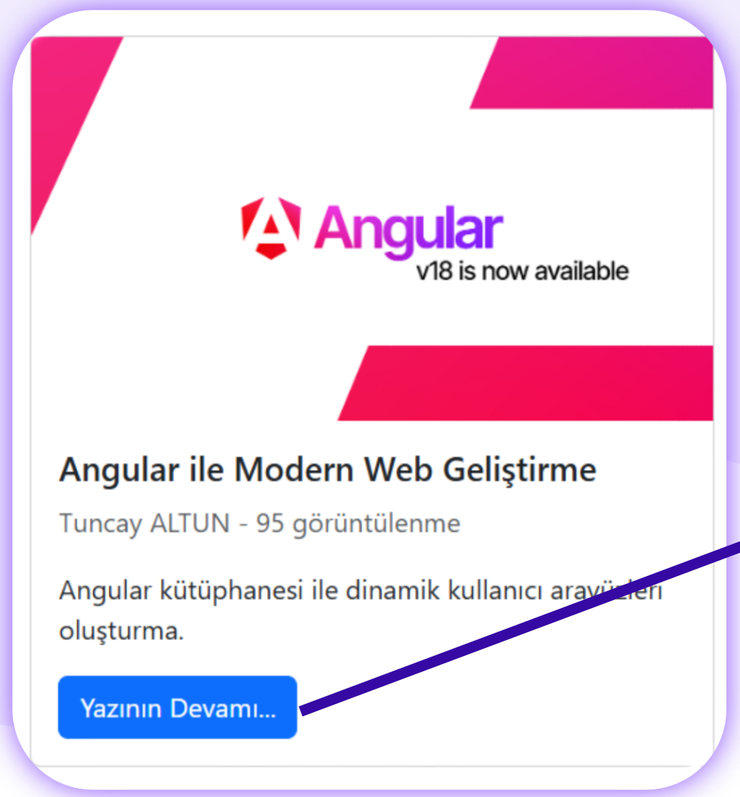


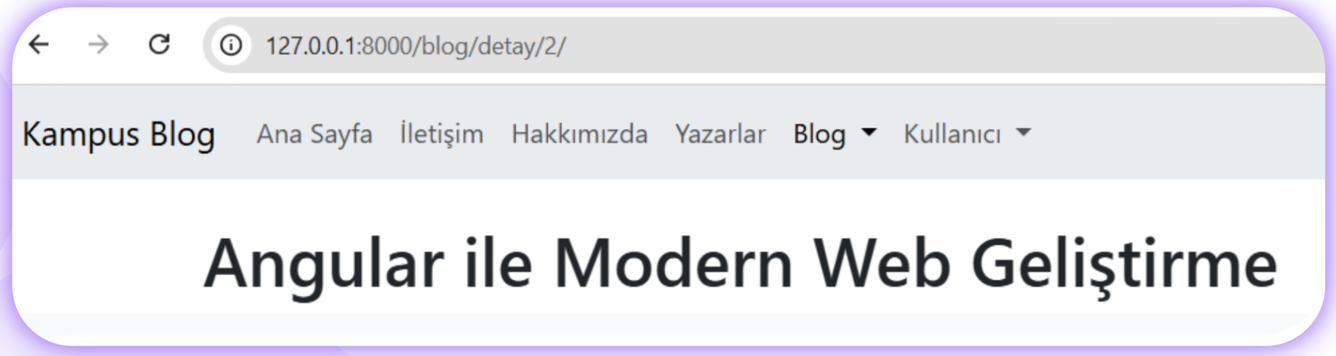
Ana Sayfa'dan Detay Sayfasına Geçiş

```
iv class="row">
    {% for post in bloglar %}
    <div class="col-md-4 mb-4">
        <div class="card h-100">
            <img src="{% static post.static_url %}" class="card-img-top" alt="Blog">
            <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="{% url 'blog:blog_detay' id=post.id %}" class="btn btn-primary"> Yazının Devamı...</a>
            </div>
        </div>
    </div>
    {% endfor %}
```

Anasayfa.html içerisindeki 'Yazının Devamı...' kısmını üstteki gibi güncelleyin ve projeyi çalıştırın.

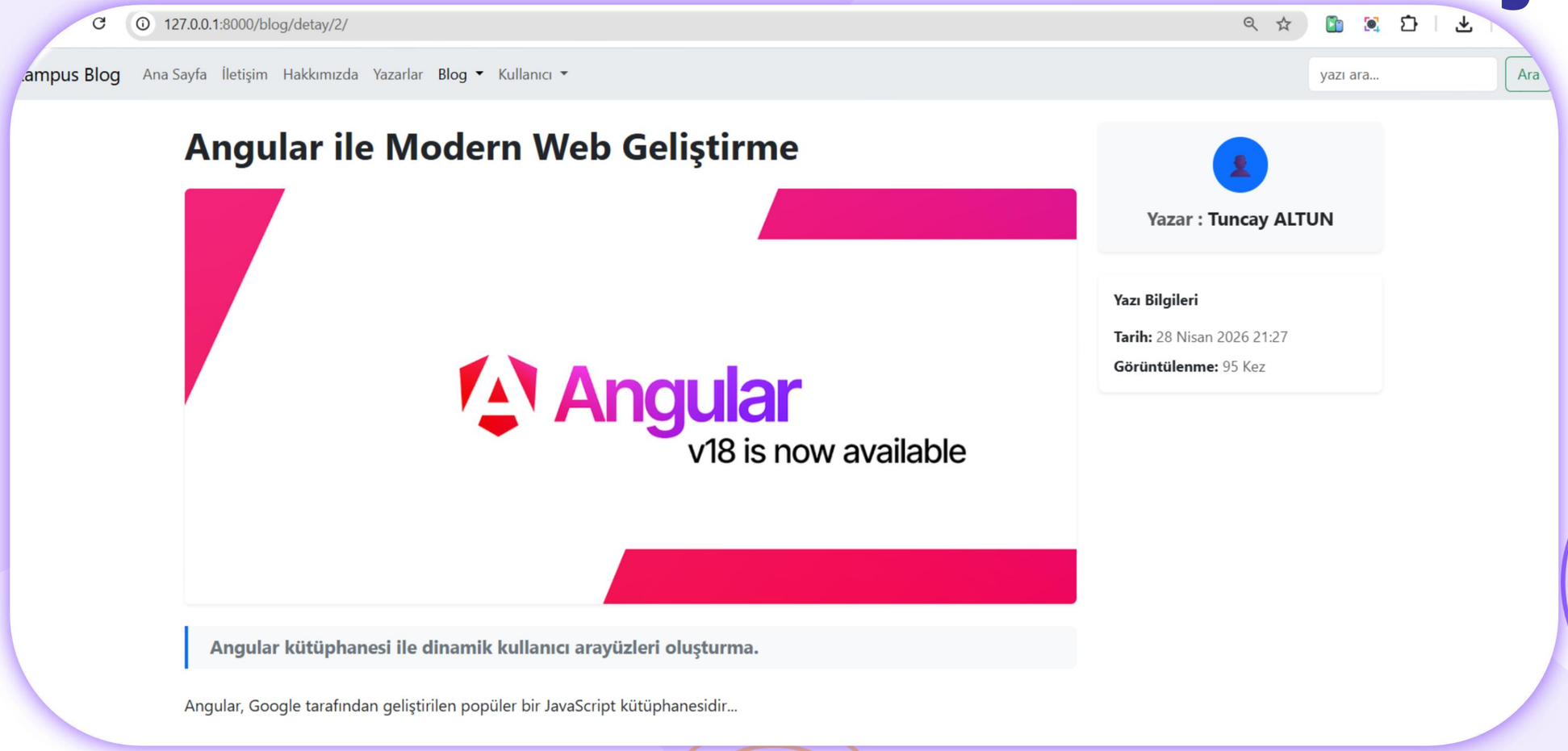
Ana Sayfa'dan Detay Sayfasına Geçiş





Artık 'Yazının Devamı' butonuna tıklandığında ilgili sayfayı açtığını göreceksiniz.

Ana Sayfa'dan Detay Sayfasına Geçiş



Detay Sayfası'nın tasarımını tamamlayınız.

Son