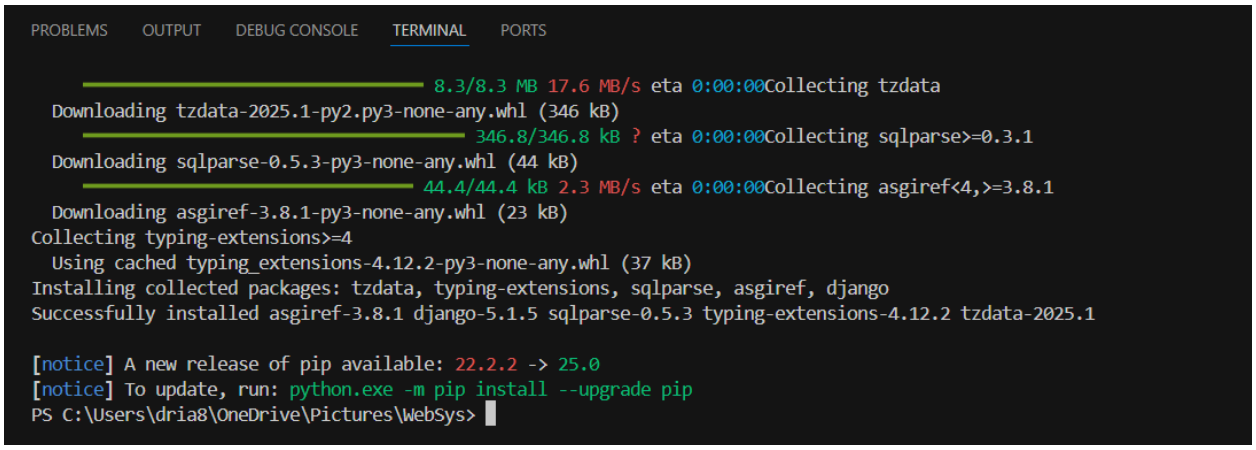
Technological Institute of the Philippines
Arlegui St, Quiapo, Manila City

College of Computer Studies

CIT 511 - Web Systems and Technologies 2

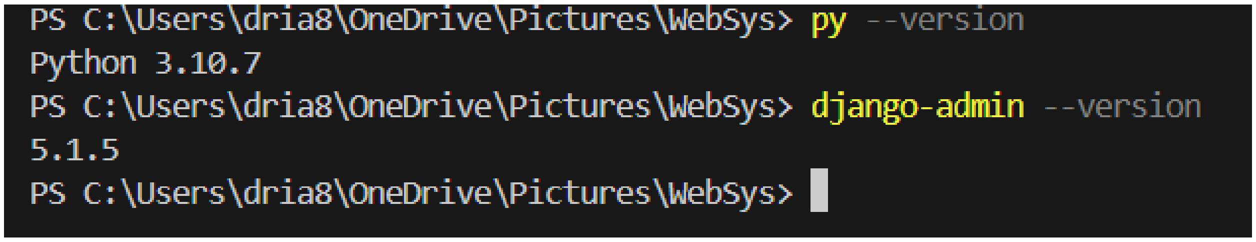
Midterm Period

Name: Dela Rosa, Rianne	Date:March 17, 2025
Program / Section: IT32S2	Instructor: Mr. Francis Carabuena
Lab Activity: Extending Blog App with Advanced Features	



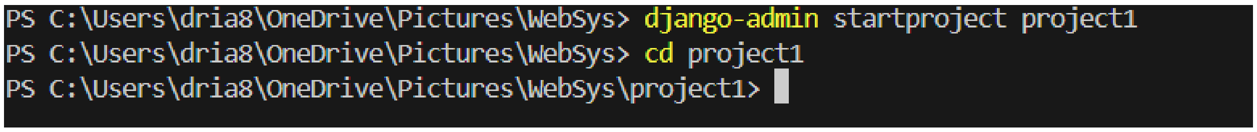
```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.3/8.3 MB 17.6 MB/s eta 0:00:00Collecting tzdata
  Downloading tzdata-2025.1-py2.py3-none-any.whl (346 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 346.8/346.8 kB ? eta 0:00:00Collecting sqlparse>=0.3.1
  Downloading sqlparse-0.5.3-py3-none-any.whl (44 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.4/44.4 kB 2.3 MB/s eta 0:00:00Collecting asgiref<4,>=3.8.1
  Downloading asgiref-3.8.1-py3-none-any.whl (23 kB)
Collecting typing-extensions>=4
  Using cached typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Installing collected packages: tzdata, typing-extensions, sqlparse, asgiref, django
Successfully installed asgiref-3.8.1 django-5.1.5 sqlparse-0.5.3 typing-extensions-4.12.2 tzdata-2025.1

[notice] A new release of pip available: 22.2.2 -> 25.0
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\dria8\OneDrive\Pictures\WebSys>
```

Django installation completed successfully.



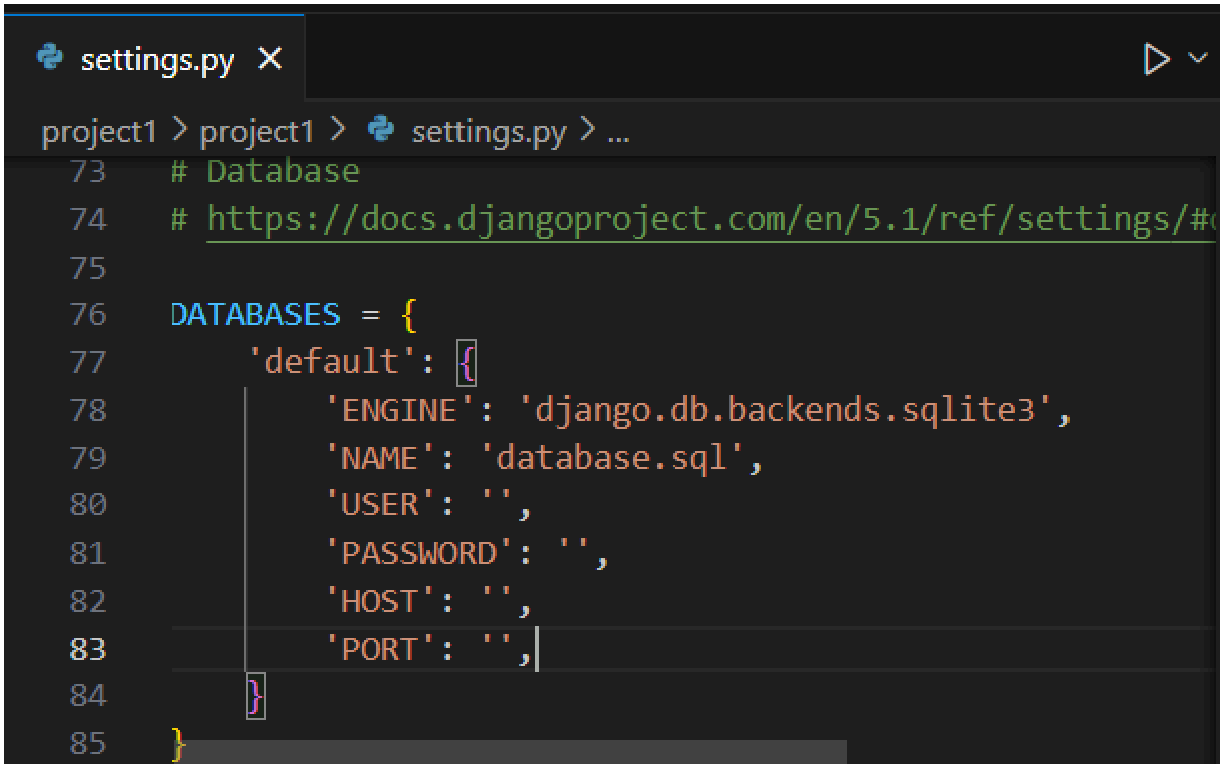
```
PS C:\Users\dria8\OneDrive\Pictures\WebSys> py --version
Python 3.10.7
PS C:\Users\dria8\OneDrive\Pictures\WebSys> django-admin --version
5.1.5
PS C:\Users\dria8\OneDrive\Pictures\WebSys>
```

Python and Django version verification.



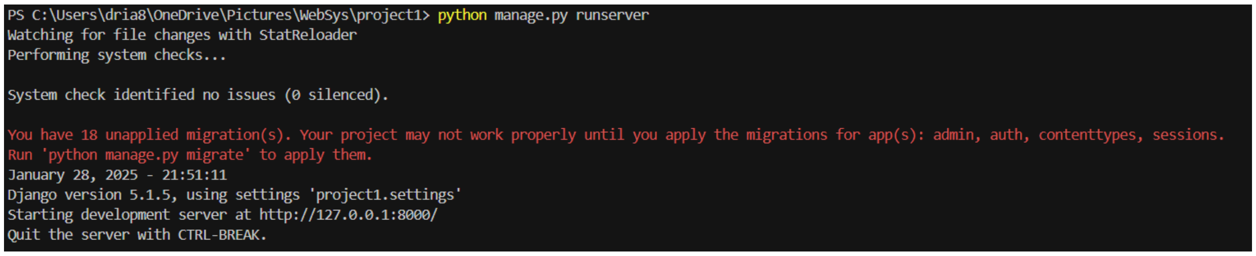
```
PS C:\Users\dria8\OneDrive\Pictures\WebSys> django-admin startproject project1
PS C:\Users\dria8\OneDrive\Pictures\WebSys> cd project1
PS C:\Users\dria8\OneDrive\Pictures\WebSys\project1>
```

New Django project created successfully.



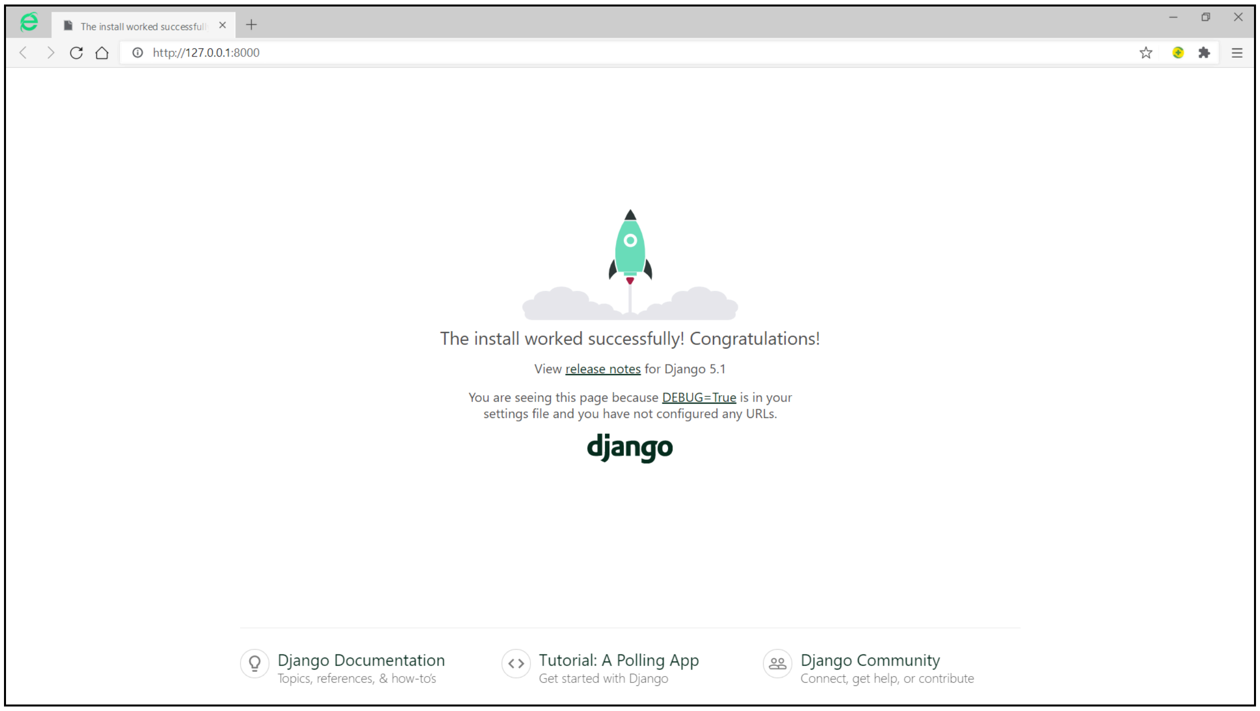
```python
# Database
# https://docs.djangoproject.com/en/5.1/ref/settings/#

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': 'database.sql',
        'USER': '',
        'PASSWORD': '',
        'HOST': '',
        'PORT': '',
    }
}
```

Database configuration updated in settings.py.



```
PS C:\Users\dria8\OneDrive\Pictures\WebSys\project1> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
January 28, 2025 - 21:51:11
Django version 5.1.5, using settings 'project1.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Django development server running successfully.

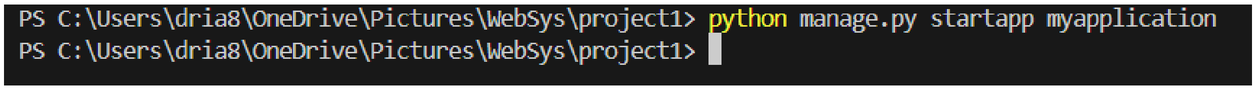


The install worked successfully! Congratulations!

View release notes for Django 5.1

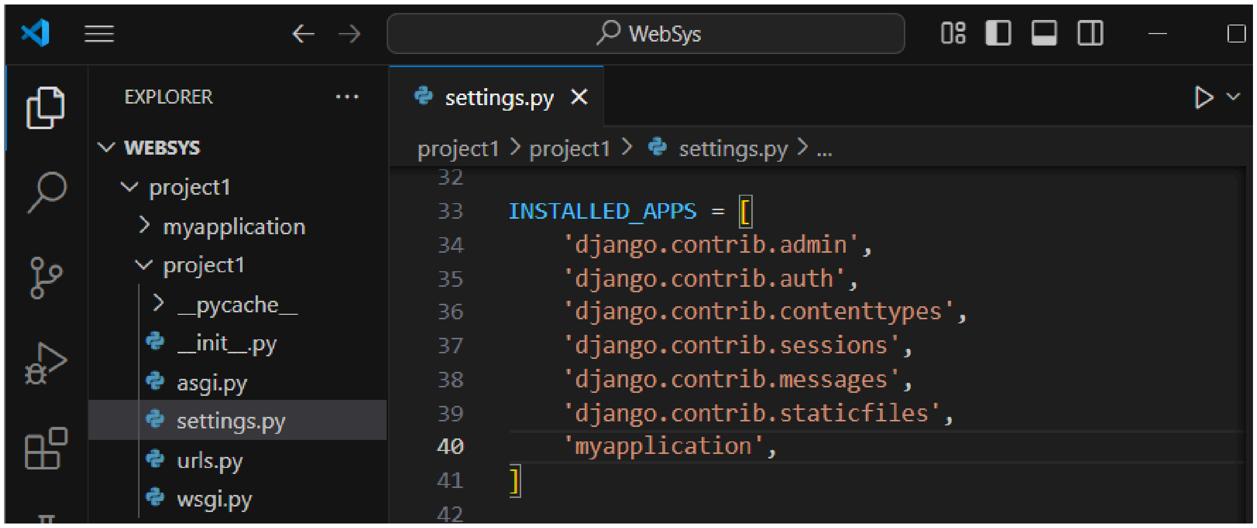
You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.

Django Installation Success Page.



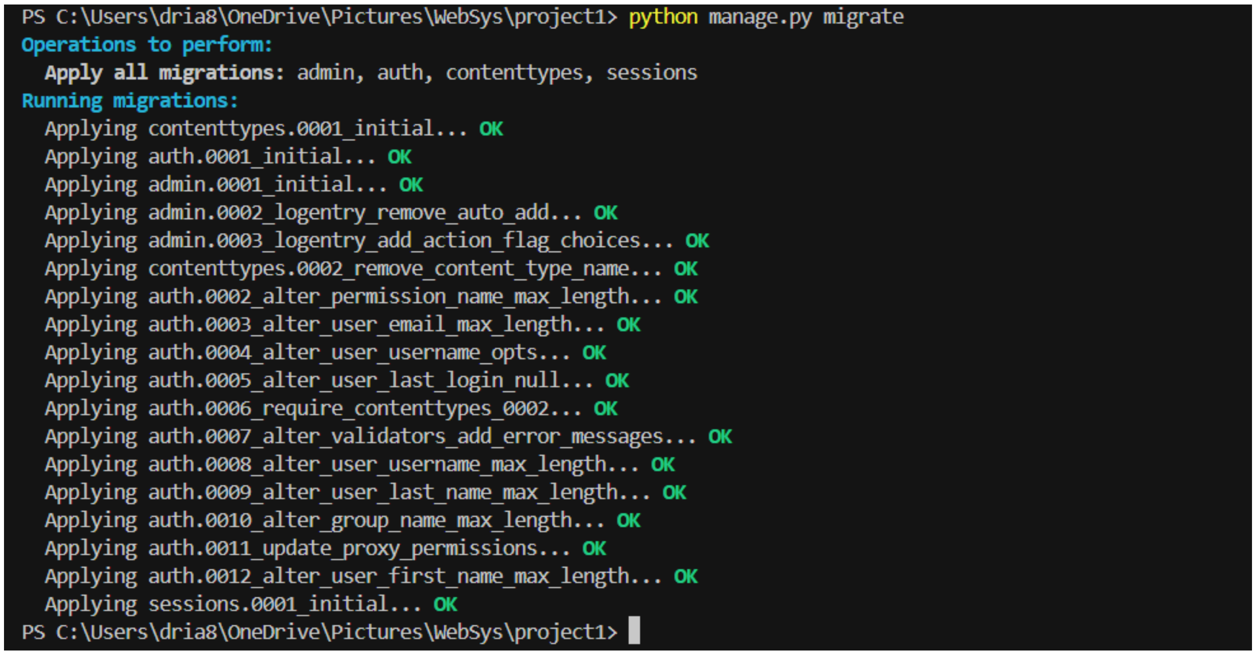
```
PS C:\Users\dria8\OneDrive\Pictures\WebSys\project1> python manage.py startapp myapplication
PS C:\Users\dria8\OneDrive\Pictures\WebSys\project1>
```

New application myapplication created successfully.



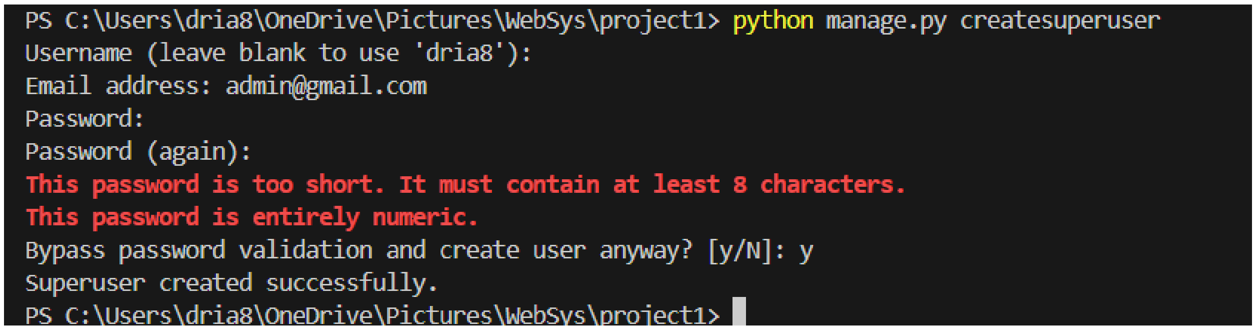
```

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

```

Application myapplication added to installed apps.



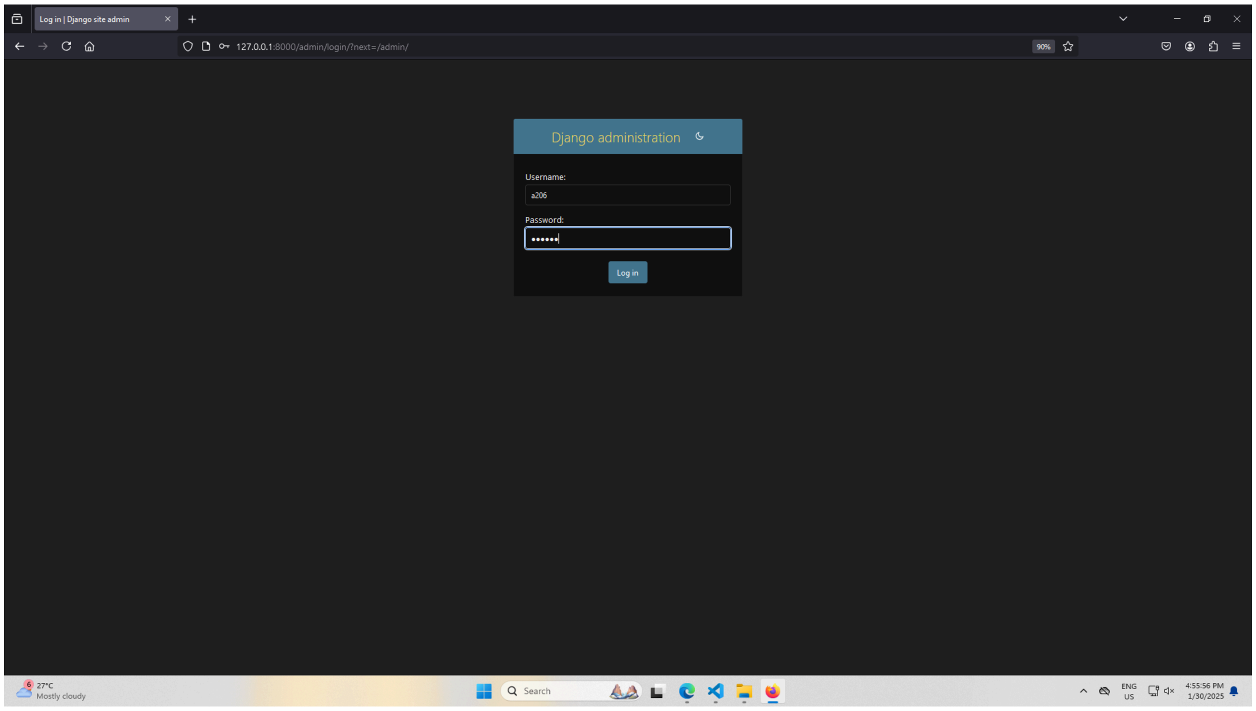
```
PS C:\Users\dria8\OneDrive\Pictures\WebSys\project1> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  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 sessions.0001_initial... OK
PS C:\Users\dria8\OneDrive\Pictures\WebSys\project1>
```

Database migrations completed successfully.

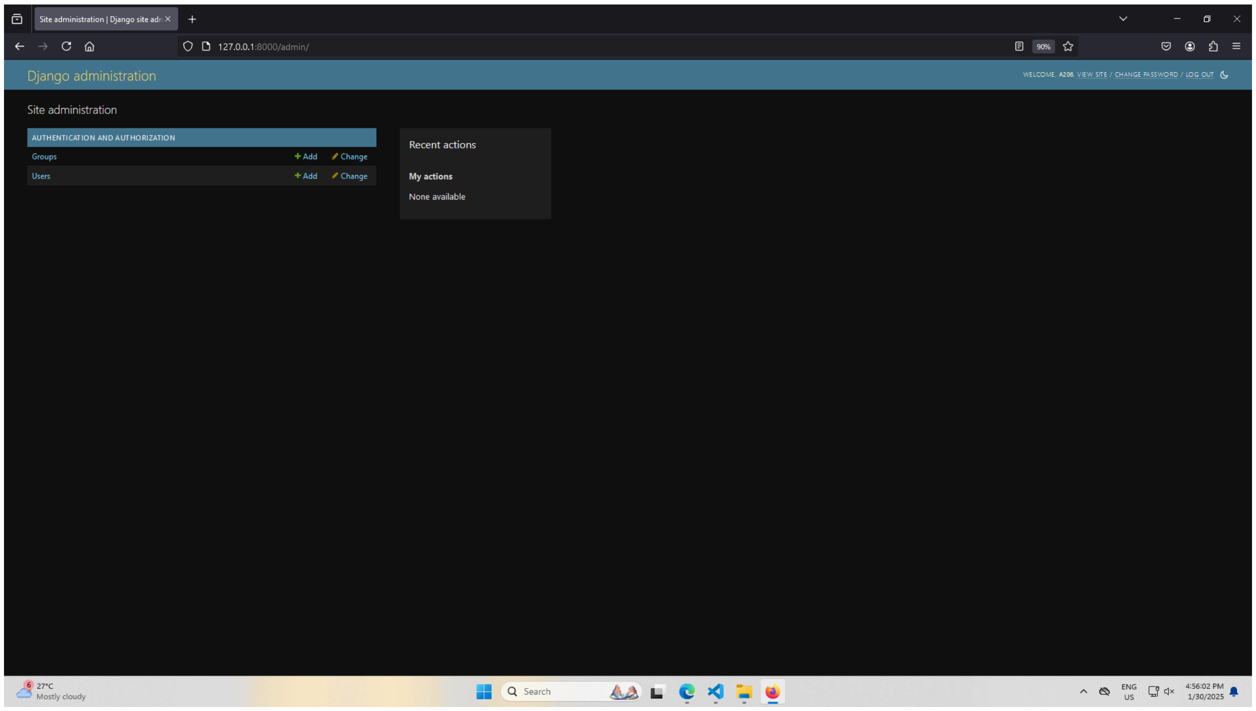


```
PS C:\Users\dria8\OneDrive\Pictures\WebSys\project1> python manage.py createsuperuser
Username (leave blank to use 'dria8'):
Email address: admin@gmail.com
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
PS C:\Users\dria8\OneDrive\Pictures\WebSys\project1>
```

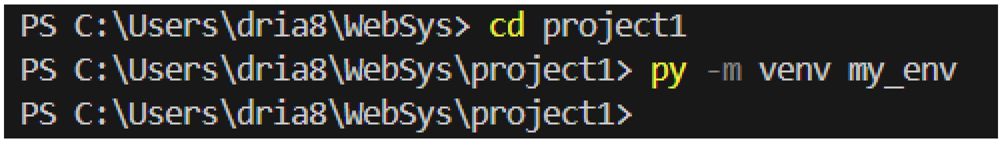
Superuser created successfully.


Log in | Django site admin
127.0.0.1:8000/admin/login/?next=/admin/
Django administration
Username:
a206
Password:
Log in

Django Admin Authentication Interface.

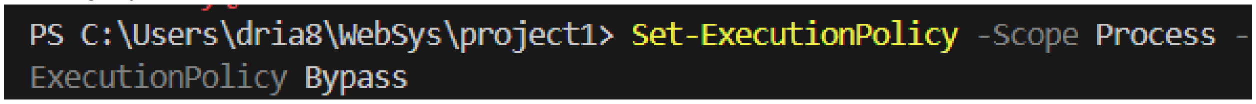


Django Admin Dashboard interface.



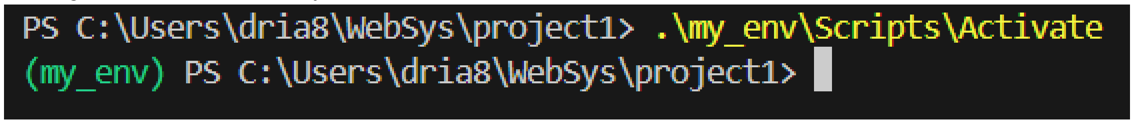
```
PS C:\Users\dria8\WebSys> cd project1
PS C:\Users\dria8\WebSys\project1> py -m venv my_env
PS C:\Users\dria8\WebSys\project1>
```

Creating my_env.



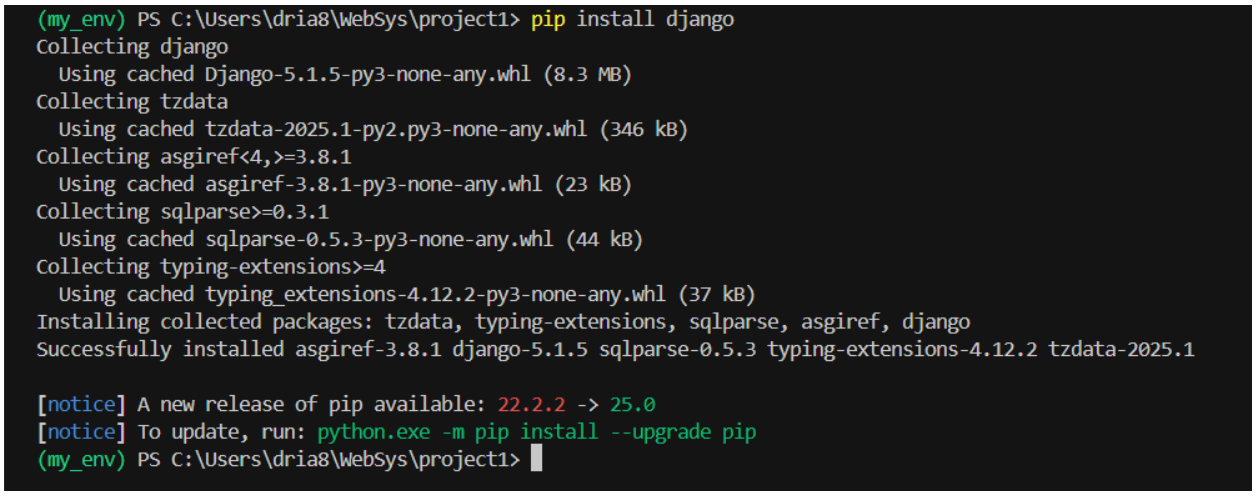
```
PS C:\Users\dria8\WebSys\project1> Set-ExecutionPolicy -Scope Process -
ExecutionPolicy Bypass
```

Setting the Execution Policy.



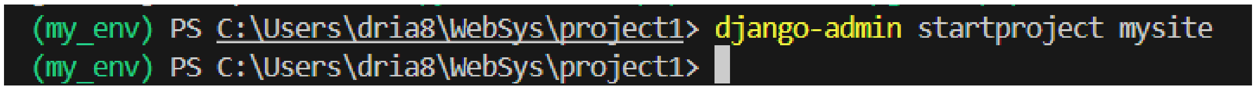
```
PS C:\Users\dria8\WebSys\project1> .\my_env\Scripts\Activate
(my_env) PS C:\Users\dria8\WebSys\project1>
```

Activating my_env



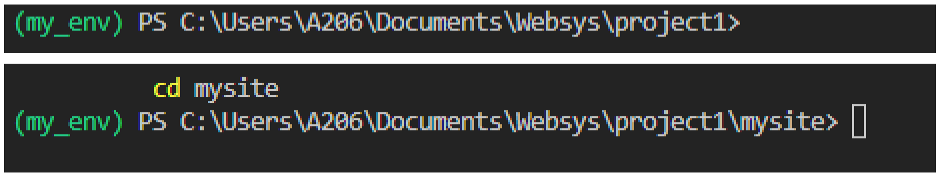
```
(my_env) PS C:\Users\dria8\WebSys\project1> pip install django
Collecting django
  Using cached Django-5.1.5-py3-none-any.whl (8.3 MB)
Collecting tzdata
  Using cached tzdata-2025.1-py2.py3-none-any.whl (346 kB)
Collecting asgiref<4,>=3.8.1
  Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)
Collecting sqlparse>=0.3.1
  Using cached sqlparse-0.5.3-py3-none-any.whl (44 kB)
Collecting typing-extensions>=4
  Using cached typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Installing collected packages: tzdata, typing-extensions, sqlparse, asgiref, django
Successfully installed asgiref-3.8.1 django-5.1.5 sqlparse-0.5.3 typing-extensions-4.12.2 tzdata-2025.1

[notice] A new release of pip available: 22.2.2 -> 25.0
[notice] To update, run: python.exe -m pip install --upgrade pip
(my_env) PS C:\Users\dria8\WebSys\project1>
```

Installing Django



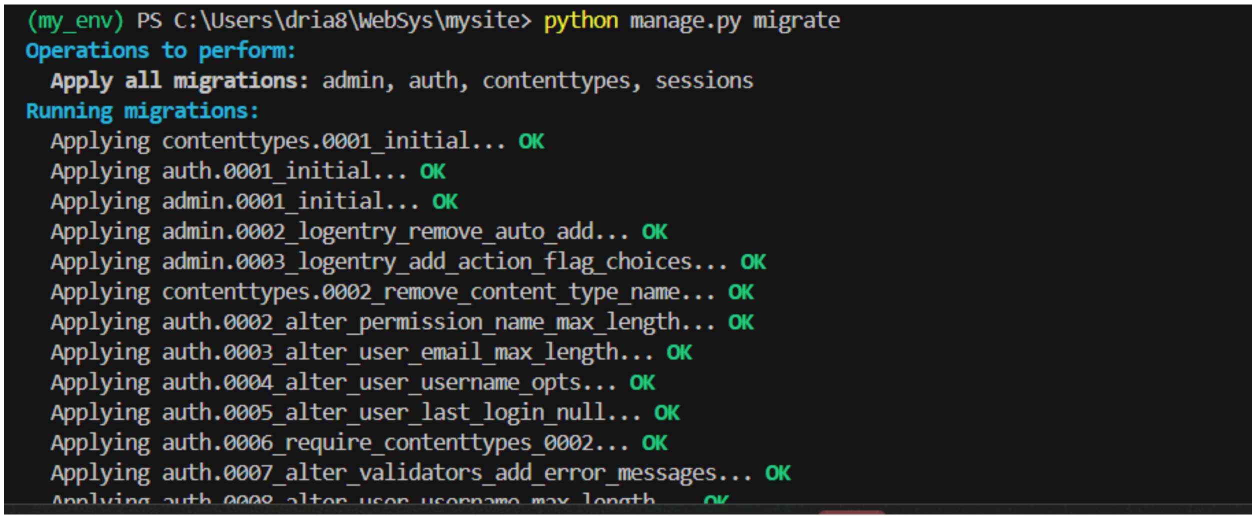
```
(my_env) PS C:\Users\dria8\WebSys\project1> django-admin startproject mysite
(my_env) PS C:\Users\dria8\WebSys\project1>
```

Creating mysite in env



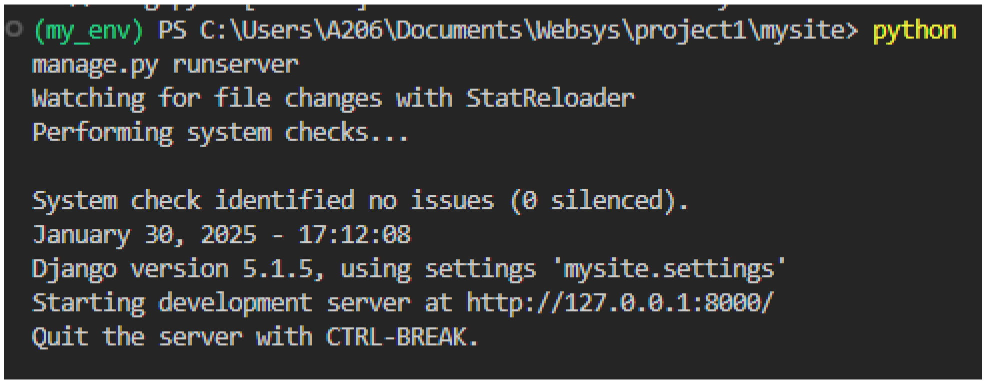
```
(my_env) PS C:\Users\A206\Documents\Websys\project1>

          cd mysite
(my_env) PS C:\Users\A206\Documents\Websys\project1\mysite>
```

Retrieving mysite



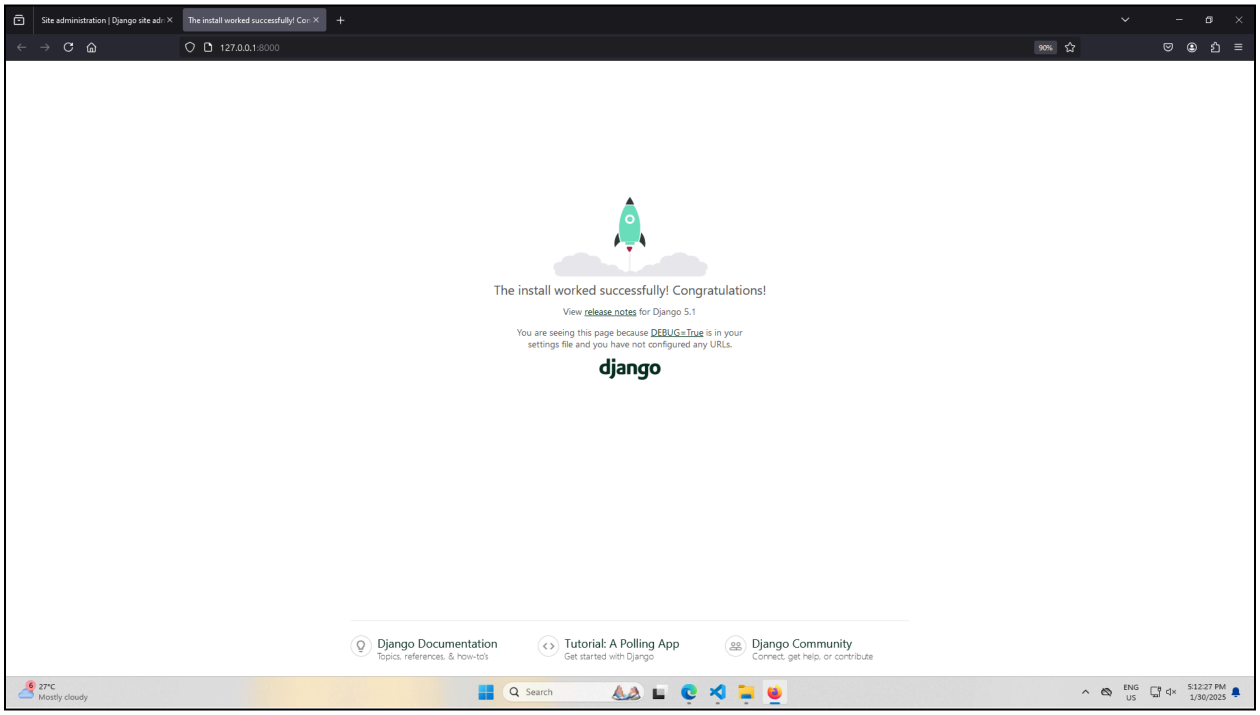
```
(my_env) PS C:\Users\dria8\WebSys\mysite> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... 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
```

Database migrations completed successfully.



```
(my_env) PS C:\Users\A206\Documents\Websys\project1\mysite> python
 manage.py runserver
 Watching for file changes with StatReloader
 Performing system checks...

 System check identified no issues (0 silenced).
 January 30, 2025 - 17:12:08
 Django version 5.1.5, using settings 'mysite.settings'
 Starting development server at http://127.0.0.1:8000/
 Quit the server with CTRL-BREAK.
```

Running Server


Site administration | Django site adı
The install worked successfully! Con
127.0.0.1:8000
90%

The install worked successfully! Congratulations!

View release notes for Django 5.1

You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.

django

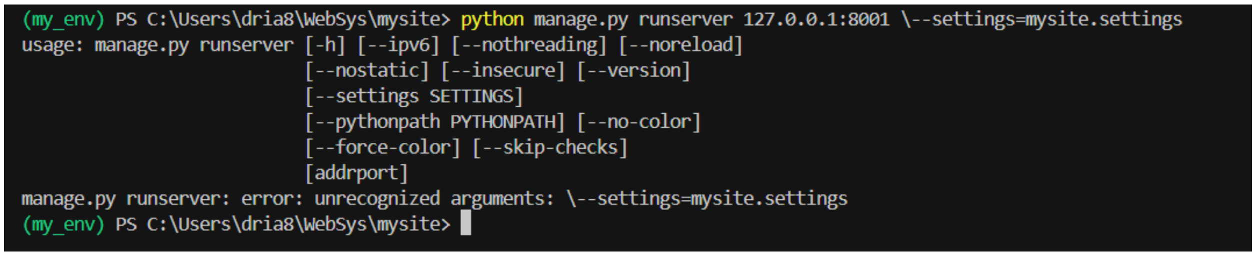
Django Documentation
Topics, references, & how-to's

Tutorial: A Polling App
Get started with Django

Django Community
Connect, get help, or contribute

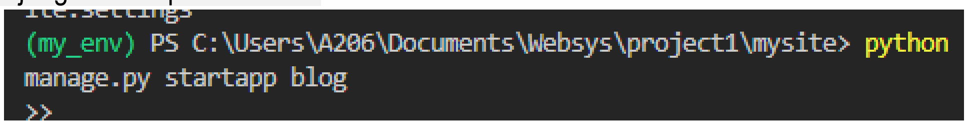
27°C
Mostly cloudy
Search
ENG
US
5:12:27 PM
1/30/2025

Django Installation Success Page.



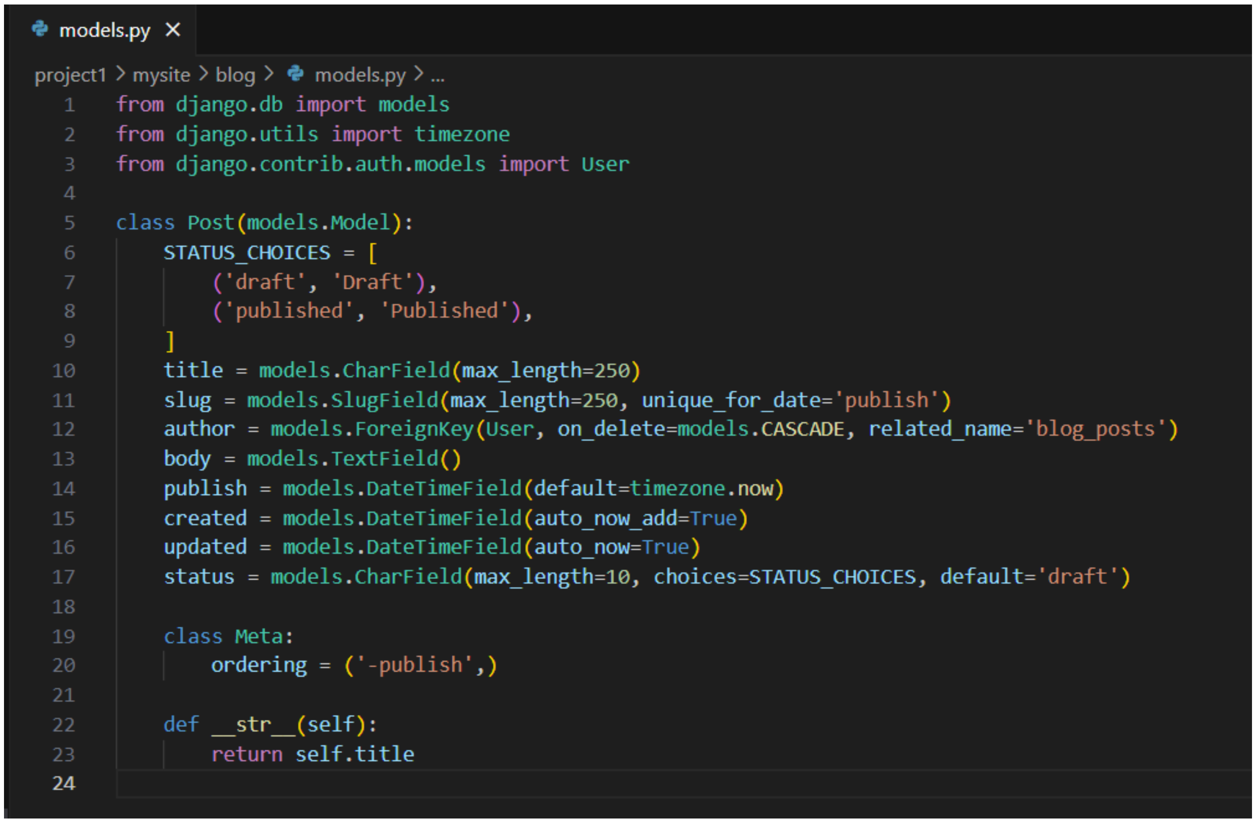
```
(my_env) PS C:\Users\dria8\WebSys\mysite> python manage.py runserver 127.0.0.1:8001 \--settings=mysite.settings
usage: manage.py runserver [-h] [--ipv6] [--nothreading] [--noreload]
                           [--nostatic] [--insecure] [--version]
                           [--settings SETTINGS]
                           [--pythonpath PYTHONPATH] [--no-color]
                           [--force-color] [--skip-checks]
                           [addrport]
manage.py runserver: error: unrecognized arguments: \--settings=mysite.settings
(my_env) PS C:\Users\dria8\WebSys\mysite>
```

Django development server



```
ite.settings
(my_env) PS C:\Users\A206\Documents\Websys\project1\mysite> python
manage.py startapp blog
>>
```

Creating a new blog application inside the Django project.

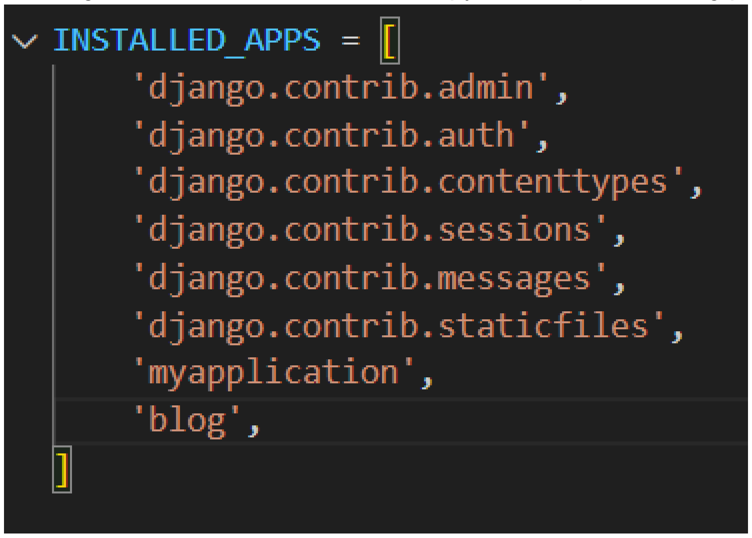


models.py ×

project1 > mysite > blog > models.py > ...

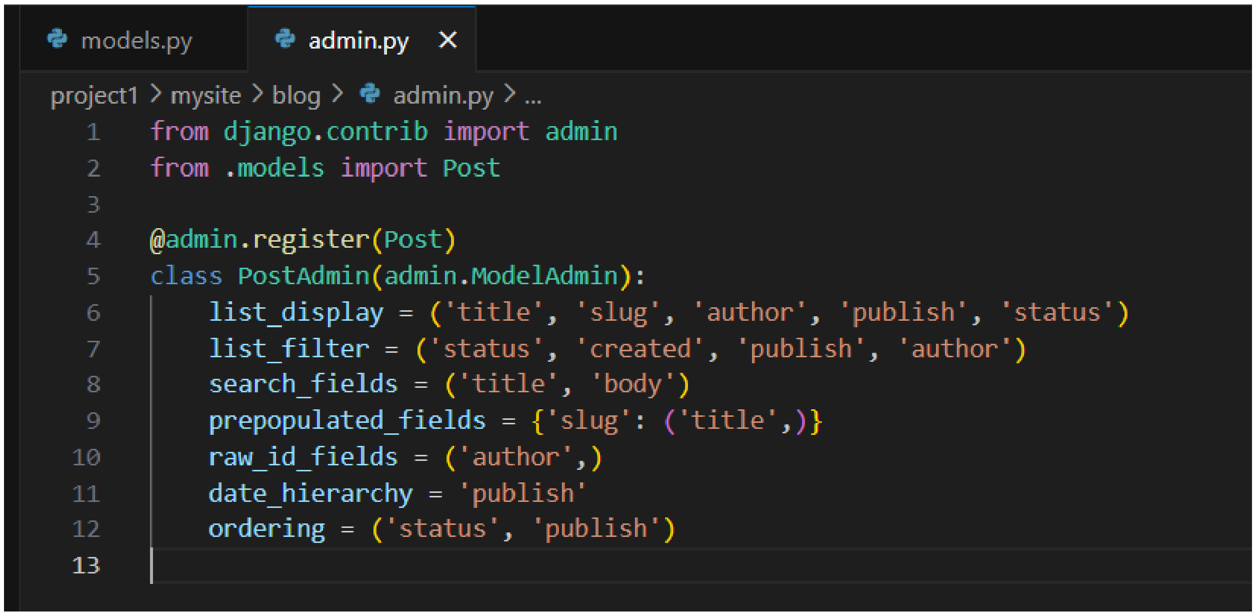
```python
from django.db import models
from django.utils import timezone
from django.contrib.auth.models import User

class Post(models.Model):
    STATUS_CHOICES = [
        ('draft', 'Draft'),
        ('published', 'Published'),
    ]
    title = models.CharField(max_length=250)
    slug = models.SlugField(max_length=250, unique_for_date='publish')
    author = models.ForeignKey(User, on_delete=models.CASCADE, related_name='blog_posts')
    body = models.TextField()
    publish = models.DateTimeField(default=timezone.now)
    created = models.DateTimeField(auto_now_add=True)
    updated = models.DateTimeField(auto_now=True)
    status = models.CharField(max_length=10, choices=STATUS_CHOICES, default='draft')

    class Meta:
        ordering = ('-publish',)

    def __str__(self):
        return self.title

```

Defining the Post model in the models.py file to represent blog posts.



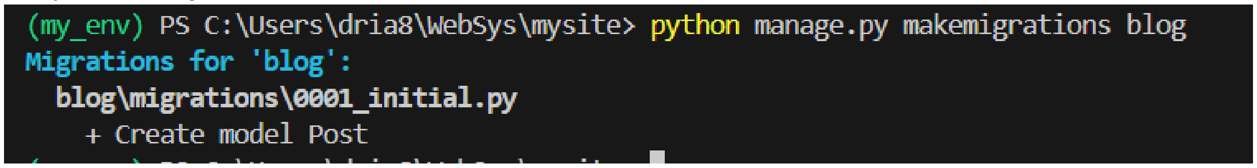
```python
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'myapplication',
    'blog',
]
```

Activating the blog application by adding it to the INSTALLED_APPS list.



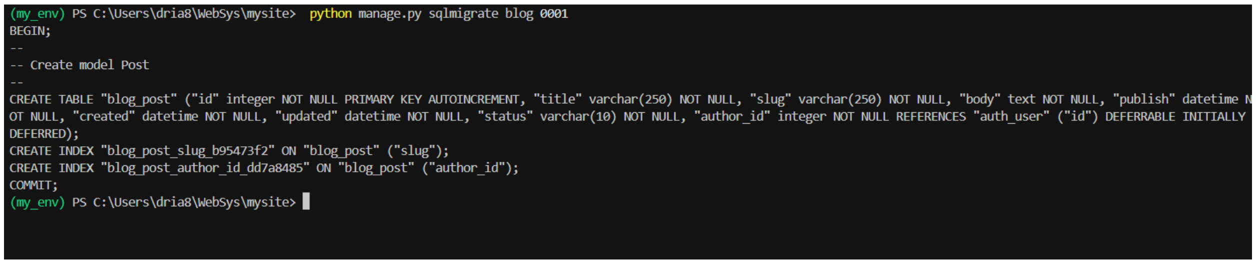
```python
from django.contrib import admin
from .models import Post

@admin.register(Post)
class PostAdmin(admin.ModelAdmin):
    list_display = ('title', 'slug', 'author', 'publish', 'status')
    list_filter = ('status', 'created', 'publish', 'author')
    search_fields = ('title', 'body')
    prepopulated_fields = {'slug': ('title',)}
    raw_id_fields = ('author',)
    date_hierarchy = 'publish'
    ordering = ('status', 'publish')

```

Verify Model Registration in Admin



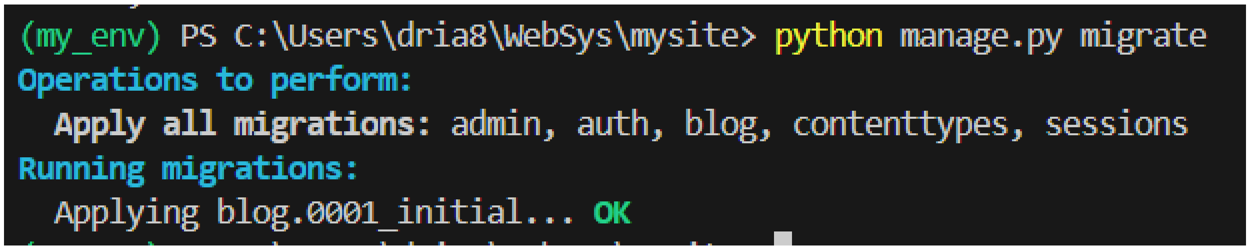
```
(my_env) PS C:\Users\dria8\WebSys\mysite> python manage.py makemigrations blog
Migrations for 'blog':
  blog\migrations\0001_initial.py
    + Create model Post
```

Generating migration files for the newly defined blog model.



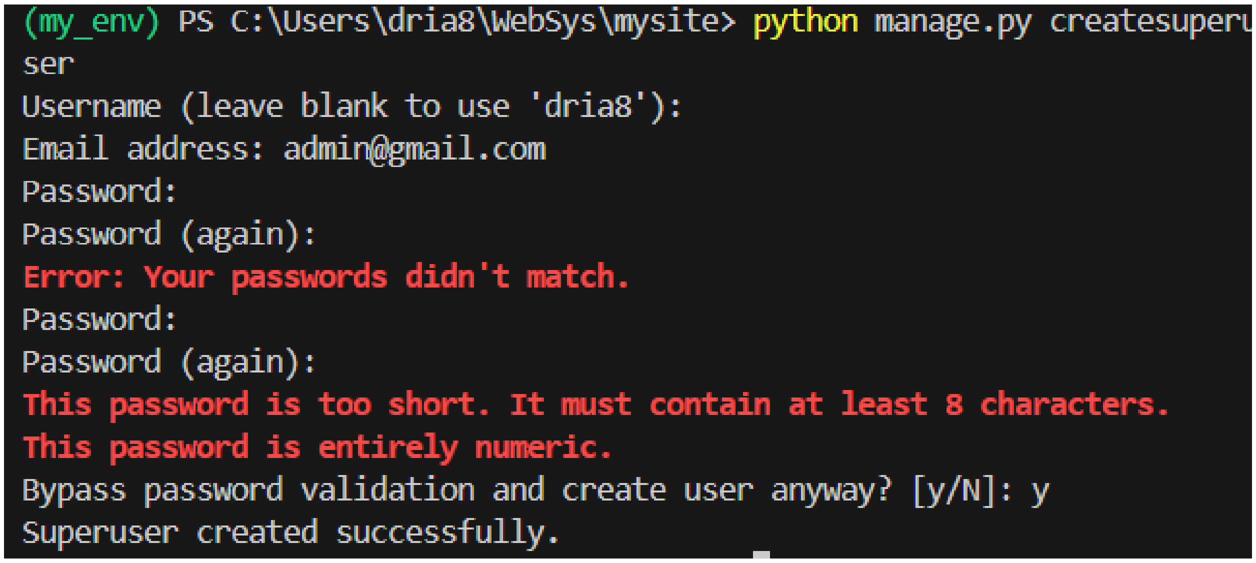
```
(my_env) PS C:\Users\dria8\WebSys\mysite>  python manage.py sqlmigrate blog 0001
BEGIN;
--
-- Create model Post
--
CREATE TABLE "blog_post" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "title" varchar(250) NOT NULL, "slug" varchar(250) NOT NULL, "body" text NOT NULL, "publish" datetime N
OT NULL, "created" datetime NOT NULL, "updated" datetime NOT NULL, "status" varchar(10) NOT NULL, "author_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY
DEFERRED);
CREATE INDEX "blog_post_slug_b95473f2" ON "blog_post" ("slug");
CREATE INDEX "blog_post_author_id_dd7a8485" ON "blog_post" ("author_id");
COMMIT;
(my_env) PS C:\Users\dria8\WebSys\mysite>
```

Applying migrations to update the database with the new blog model.



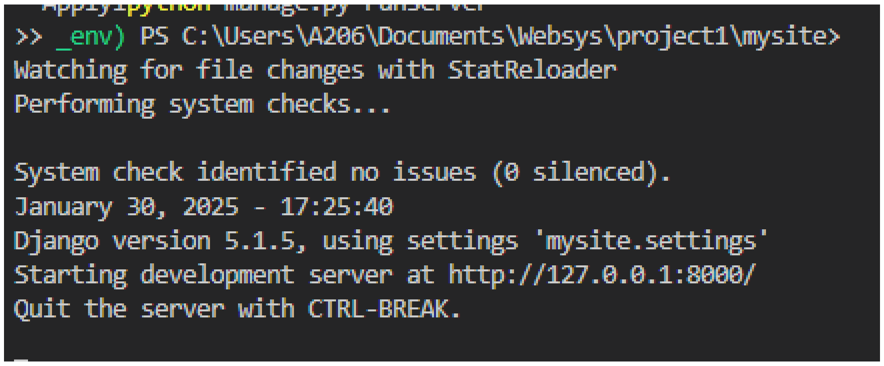
```
(my_env) PS C:\Users\dria8\WebSys\mysite> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, blog, contenttypes, sessions
Running migrations:
  Applying blog.0001_initial... OK
```

Migrate to sync the changes to the database



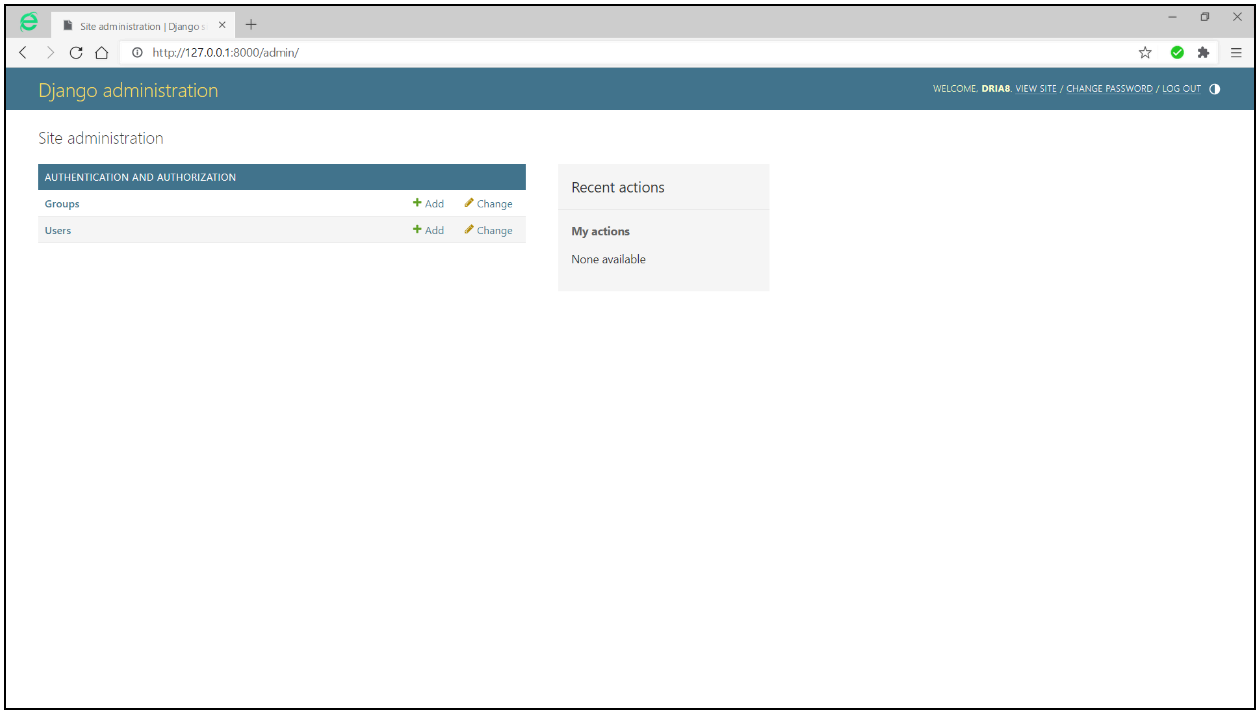
```
(my_env) PS C:\Users\dria8\WebSys\mysite> python manage.py createsuperu
ser
Username (leave blank to use 'dria8'):
Email address: admin@gmail.com
Password:
Password (again):
Error: Your passwords didn't match.
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

Creating SuperUser



```
  Apply1python manage.py runserver
>> _env) PS C:\Users\A206\Documents\Websys\project1\mysite>
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
January 30, 2025 - 17:25:40
Django version 5.1.5, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Running the development server after adding the blog application and database migrations.


Site administration | Django s
http://127.0.0.1:8000/admin/
Django administration
WELCOME, DRIA8. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

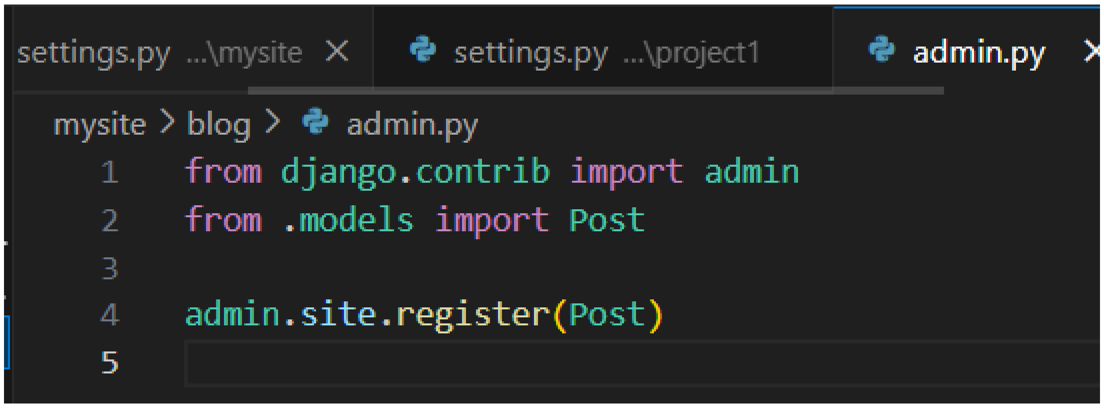
AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

Recent actions

My actions

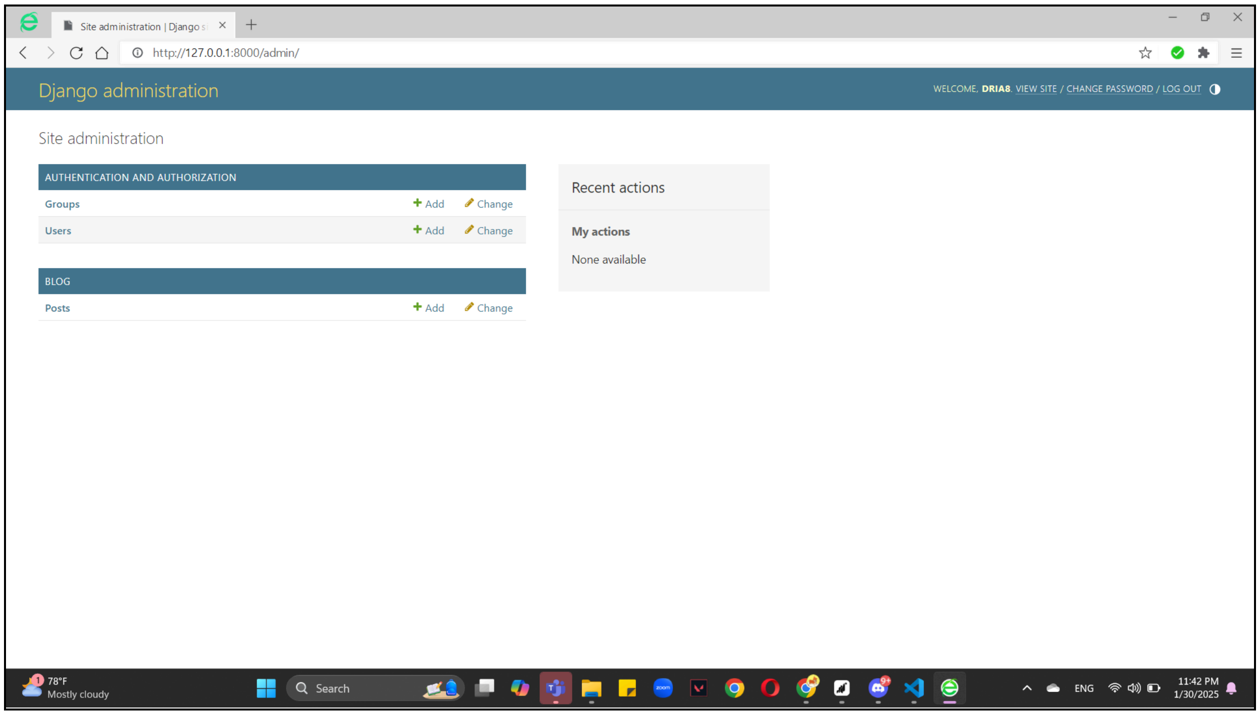
None available

Admin Page



```python
from django.contrib import admin
from .models import Post

admin.site.register(Post)

```

Adding the blog models that we created in our administration file.


Site administration | Django site admin

http://127.0.0.1:8000/admin/

Django administration

WELCOME, DRIA8. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

BLOG		
Posts	+ Add	Change

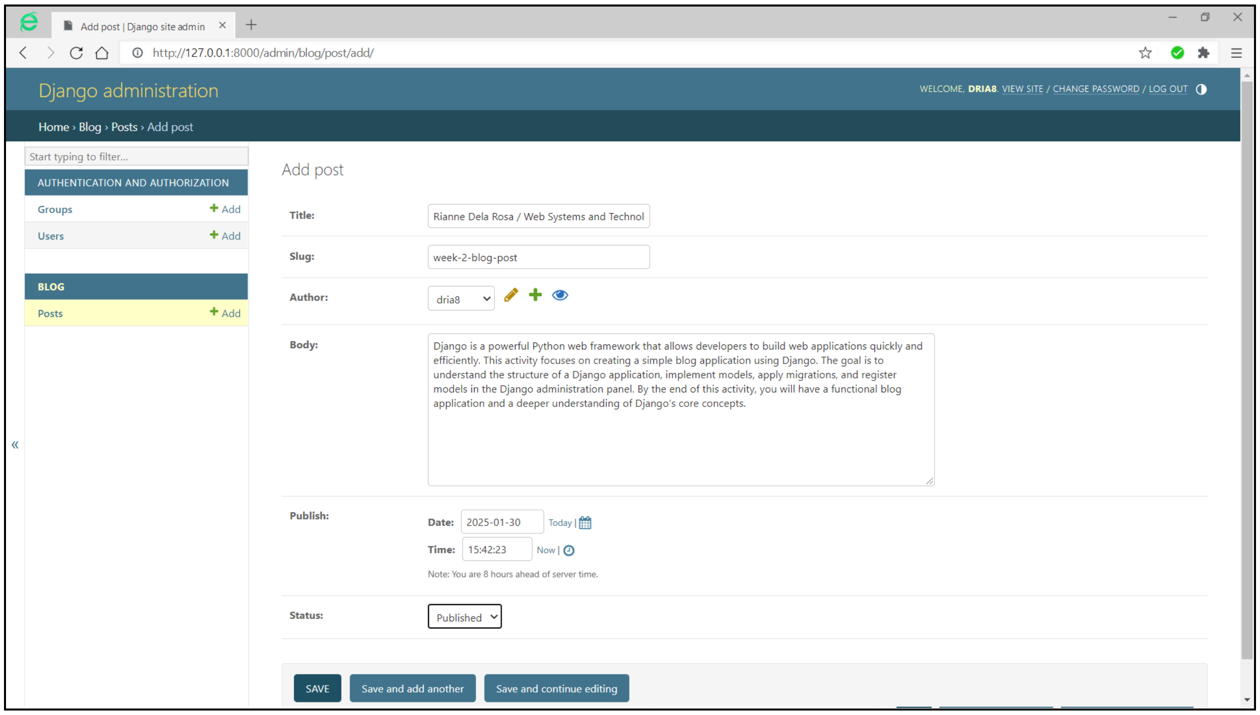
Recent actions

My actions

None available

78°F Mostly cloudy · Search · ENG · 11:42 PM 1/30/2025

Blog Section Displayed


Add post | Django site admin

http://127.0.0.1:8000/admin/blog/post/add/

Django administration

WELCOME, DRIA8. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Blog › Posts › Add post

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add

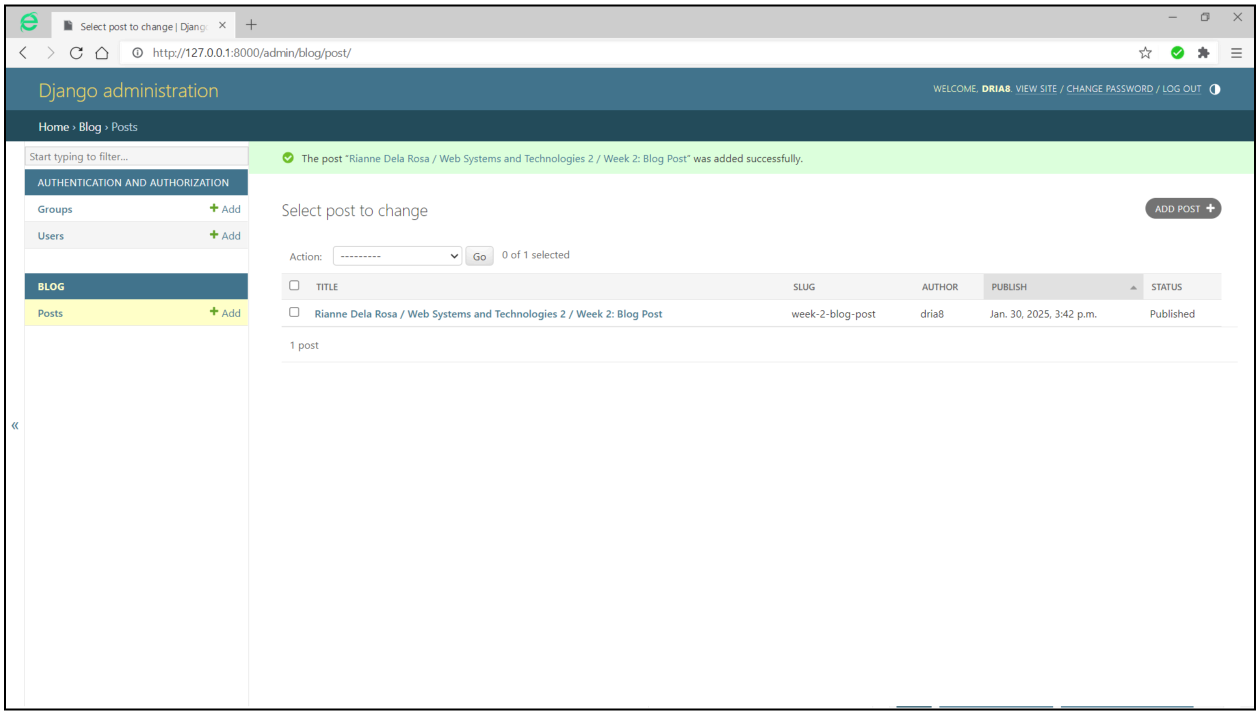
BLOG
Posts + Add

Add post

Title:	Rianne Dela Rosa / Web Systems and Technol
Slug:	week-2-blog-post
Author:	dria8
Body:	Django is a powerful Python web framework that allows developers to build web applications quickly and efficiently. This activity focuses on creating a simple blog application using Django. The goal is to understand the structure of a Django application, implement models, apply migrations, and register models in the Django administration panel. By the end of this activity, you will have a functional blog application and a deeper understanding of Django's core concepts.
Publish:	Date: 2025-01-30 Today | Time: 15:42:23 Now | Note: You are 8 hours ahead of server time.
Status:	Published

SAVE · Save and add another · Save and continue editing

Creating Blog


Select post to change | Django

http://127.0.0.1:8000/admin/blog/post/

Django administration

WELCOME, DRIA8. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Blog › Posts

The post "Rianne Dela Rosa / Web Systems and Technologies 2 / Week 2: Blog Post" was added successfully.

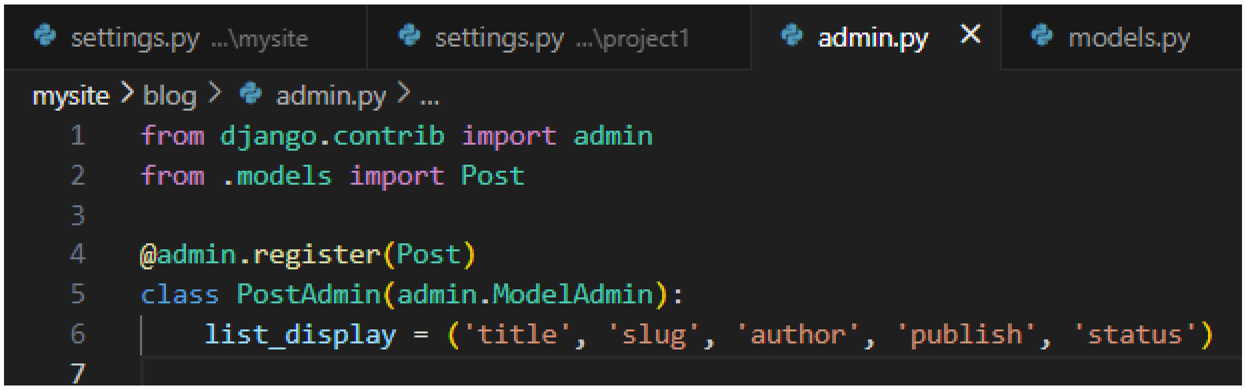
Select post to change

Action: --------- Go 0 of 1 selected

TITLE	SLUG	AUTHOR	PUBLISH	STATUS
Rianne Dela Rosa / Web Systems and Technologies 2 / Week 2: Blog Post	week-2-blog-post	dria8	Jan. 30, 2025, 3:42 p.m.	Published

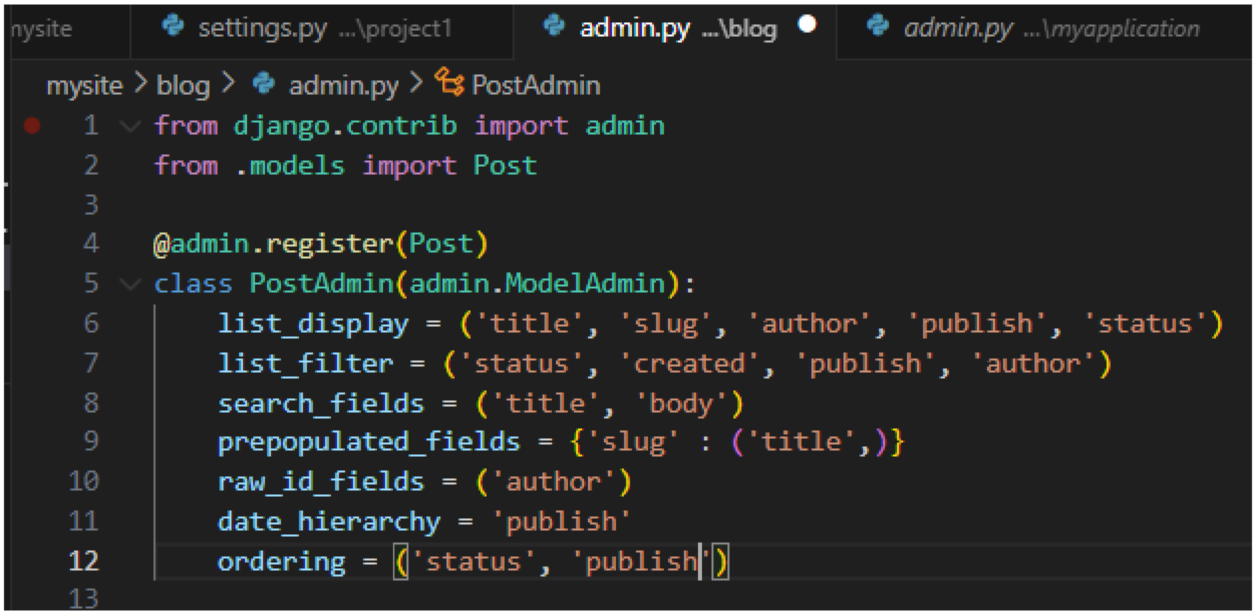
1 post

Blog Created



```python
from django.contrib import admin
from .models import Post

@admin.register(Post)
class PostAdmin(admin.ModelAdmin):
    list_display = ('title', 'slug', 'author', 'publish', 'status')
```

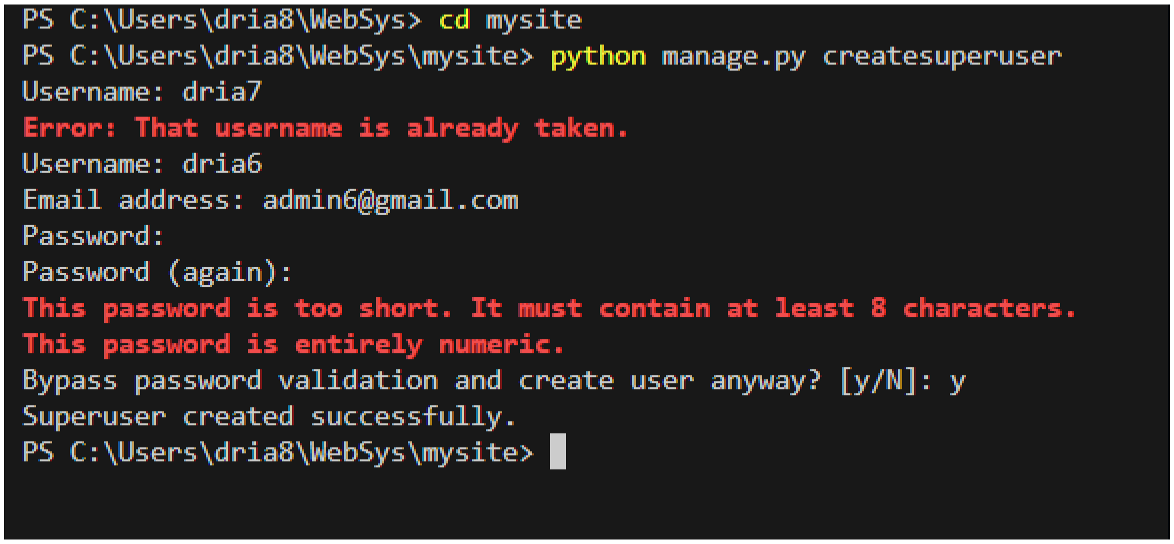
Customizing the way Models are Displayed



nysite
settings.py ...\project1
admin.py ...\blog
admin.py ...\myapplication
mysite > blog > admin.py > PostAdmin

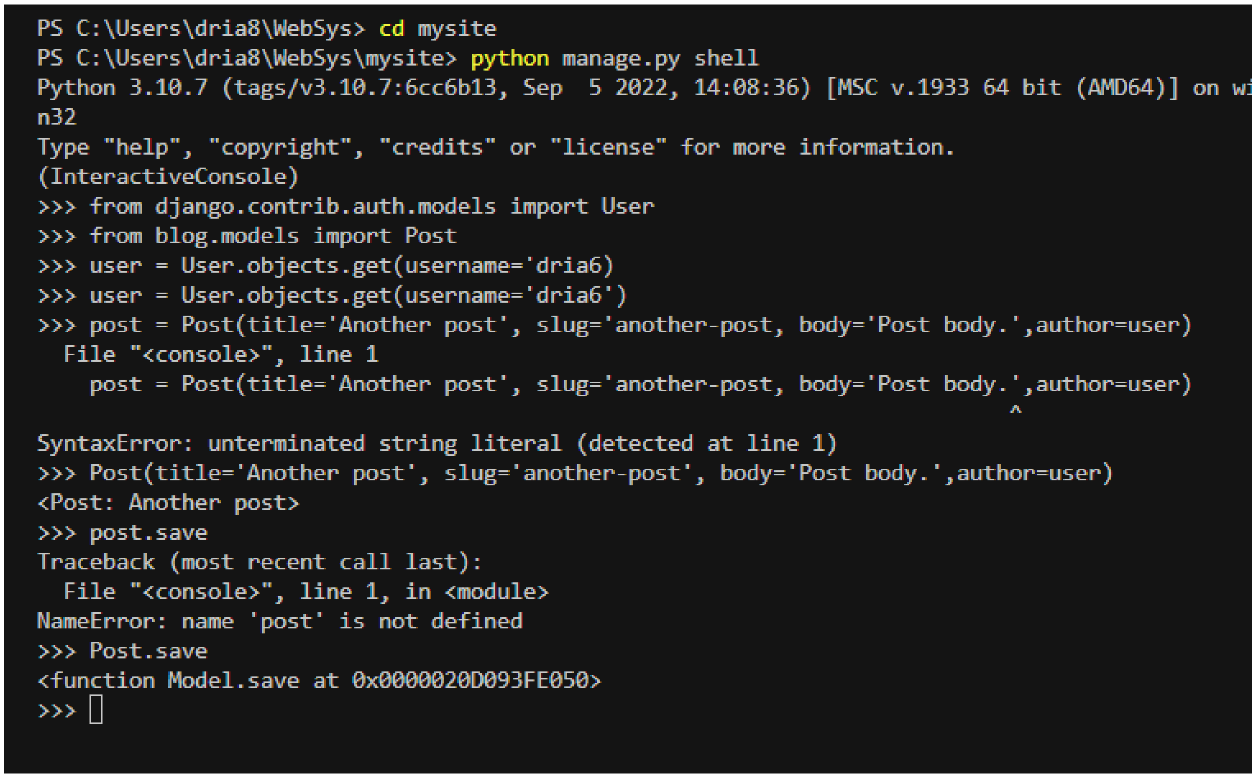
```python
from django.contrib import admin
from .models import Post

@admin.register(Post)
class PostAdmin(admin.ModelAdmin):
    list_display = ('title', 'slug', 'author', 'publish', 'status')
    list_filter = ('status', 'created', 'publish', 'author')
    search_fields = ('title', 'body')
    prepopulated_fields = {'slug' : ('title',)}
    raw_id_fields = ('author')
    date_hierarchy = 'publish'
    ordering = ('status', 'publish')

```

Customizing the way Models are Displayed



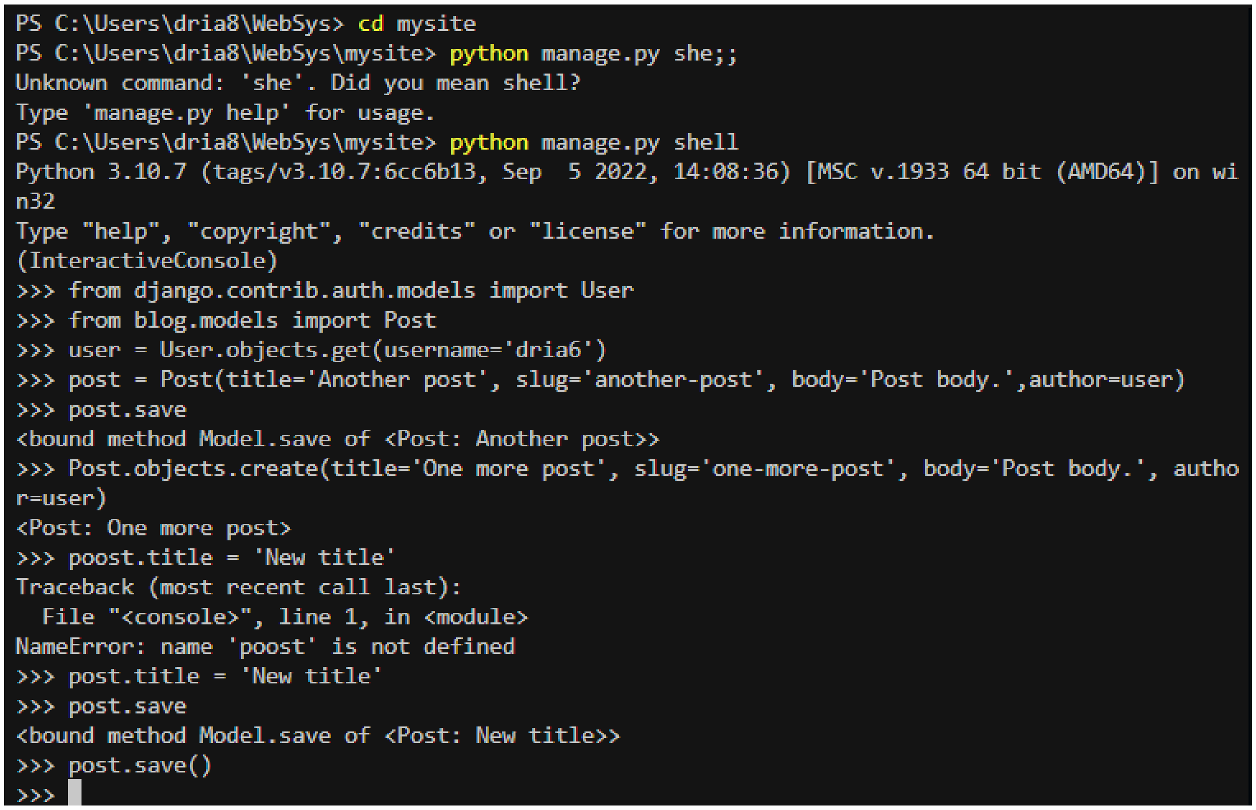
```
PS C:\Users\dria8\WebSys> cd mysite
PS C:\Users\dria8\WebSys\mysite> python manage.py createsuperuser
Username: dria7
Error: That username is already taken.
Username: dria6
Email address: admin6@gmail.com
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
PS C:\Users\dria8\WebSys\mysite>
```

Creating create super user



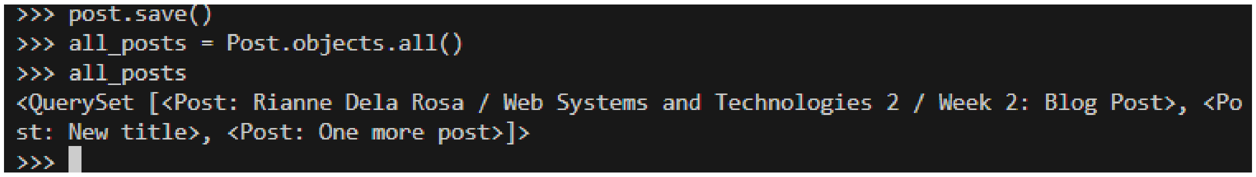
```
PS C:\Users\dria8\WebSys> cd mysite
PS C:\Users\dria8\WebSys\mysite> python manage.py shell
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep  5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on wi
n32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from django.contrib.auth.models import User
>>> from blog.models import Post
>>> user = User.objects.get(username='dria6)
>>> user = User.objects.get(username='dria6')
>>> post = Post(title='Another post', slug='another-post, body='Post body.',author=user)
  File "<console>", line 1
    post = Post(title='Another post', slug='another-post, body='Post body.',author=user)
                                                                          ^

SyntaxError: unterminated string literal (detected at line 1)
>>> Post(title='Another post', slug='another-post', body='Post body.',author=user)
<Post: Another post>
>>> post.save
Traceback (most recent call last):
  File "<console>", line 1, in <module>
NameError: name 'post' is not defined
>>> Post.save
<function Model.save at 0x0000020D093FE050>
>>> 
```

Creating Objects



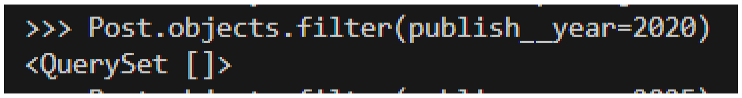
```
PS C:\Users\dria8\WebSys> cd mysite
PS C:\Users\dria8\WebSys\mysite> python manage.py she;;
Unknown command: 'she'. Did you mean shell?
Type 'manage.py help' for usage.
PS C:\Users\dria8\WebSys\mysite> python manage.py shell
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep  5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on wi
n32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from django.contrib.auth.models import User
>>> from blog.models import Post
>>> user = User.objects.get(username='dria6')
>>> post = Post(title='Another post', slug='another-post', body='Post body.',author=user)
>>> post.save
<bound method Model.save of <Post: Another post>>
>>> Post.objects.create(title='One more post', slug='one-more-post', body='Post body.', autho
r=user)
<Post: One more post>
>>> poost.title = 'New title'
Traceback (most recent call last):
  File "<console>", line 1, in <module>
NameError: name 'poost' is not defined
>>> post.title = 'New title'
>>> post.save
<bound method Model.save of <Post: New title>>
>>> post.save()
>>> 
```

Updating Objects

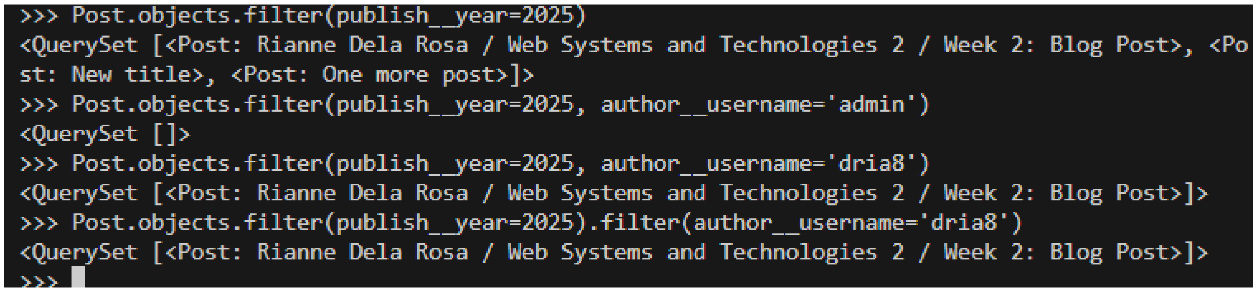


```
>>> post.save()
>>> all_posts = Post.objects.all()
>>> all_posts
<QuerySet [<Post: Rianne Dela Rosa / Web Systems and Technologies 2 / Week 2: Blog Post>, <Po
st: New title>, <Post: One more post>]>
>>>
```

Retrieving Objects

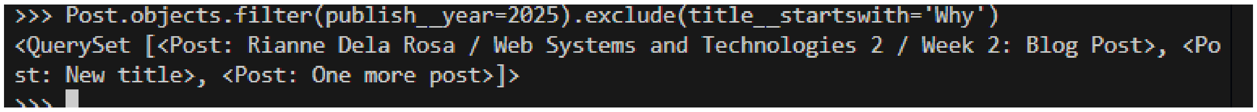


```
>>> Post.objects.filter(publish__year=2020)
<QuerySet []>
```



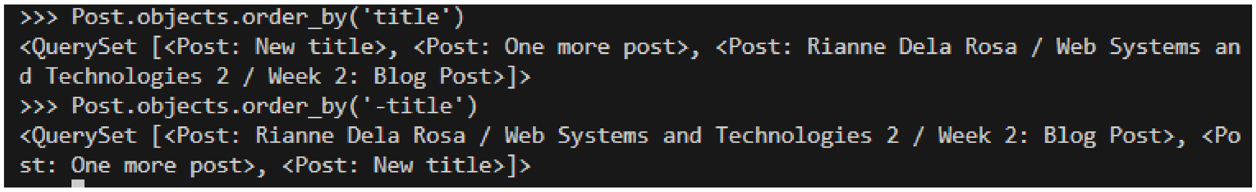
```
>>> Post.objects.filter(publish__year=2025)
<QuerySet [<Post: Rianne Dela Rosa / Web Systems and Technologies 2 / Week 2: Blog Post>, <Po
st: New title>, <Post: One more post>]>
>>> Post.objects.filter(publish__year=2025, author__username='admin')
<QuerySet []>
>>> Post.objects.filter(publish__year=2025, author__username='dria8')
<QuerySet [<Post: Rianne Dela Rosa / Web Systems and Technologies 2 / Week 2: Blog Post>]>
>>> Post.objects.filter(publish__year=2025).filter(author__username='dria8')
<QuerySet [<Post: Rianne Dela Rosa / Web Systems and Technologies 2 / Week 2: Blog Post>]>
>>>
```

Using the Filter( ) Method



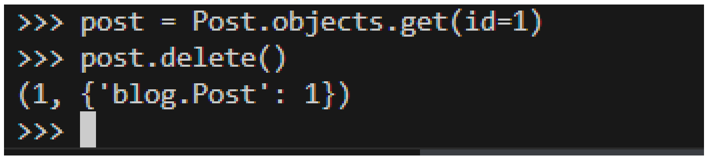
```
>>> Post.objects.filter(publish__year=2025).exclude(title__startswith='Why')
<QuerySet [<Post: Rianne Dela Rosa / Web Systems and Technologies 2 / Week 2: Blog Post>, <Po
st: New title>, <Post: One more post>]>
>>>
```

Using exclude( )



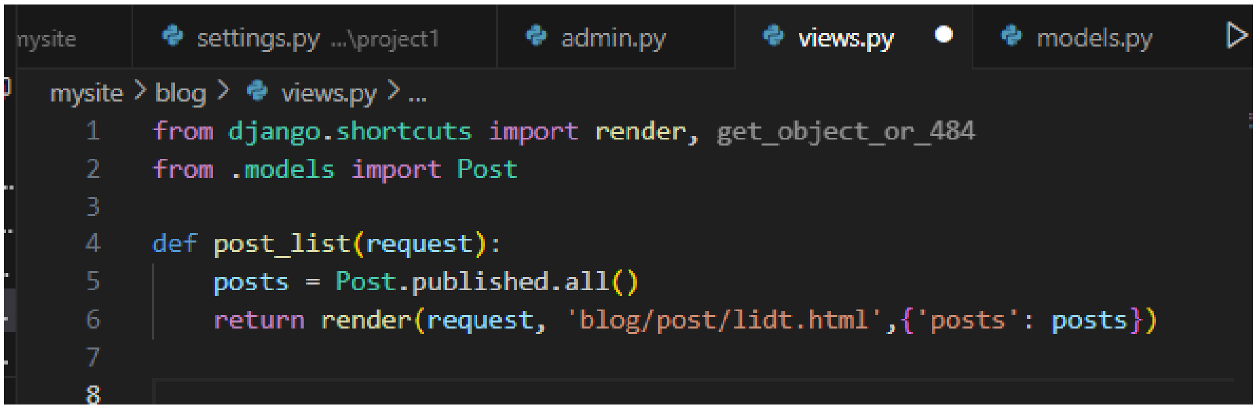
```
>>> Post.objects.order_by('title')
<QuerySet [<Post: New title>, <Post: One more post>, <Post: Rianne Dela Rosa / Web Systems an
d Technologies 2 / Week 2: Blog Post>]>
>>> Post.objects.order_by('-title')
<QuerySet [<Post: Rianne Dela Rosa / Web Systems and Technologies 2 / Week 2: Blog Post>, <Po
st: One more post>, <Post: New title>]>
```

Using order_by( )

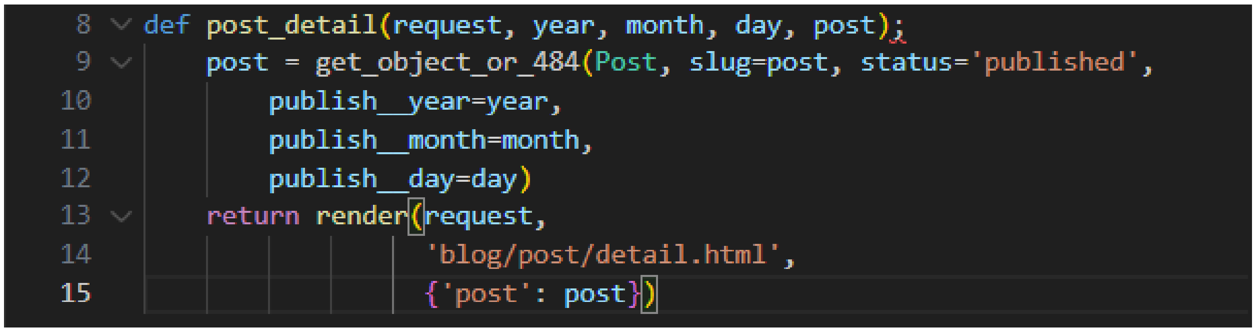


```
>>> post = Post.objects.get(id=1)
>>> post.delete()
(1, {'blog.Post': 1})
>>>
```

Deleting Objects( )

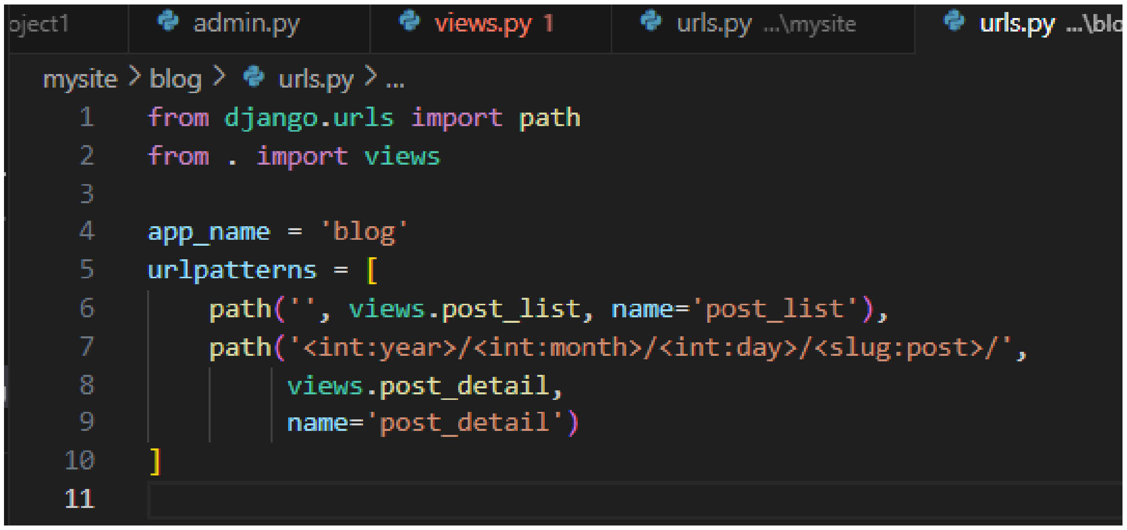


```python
from django.shortcuts import render, get_object_or_484
from .models import Post

def post_list(request):
    posts = Post.published.all()
    return render(request, 'blog/post/lidt.html',{'posts': posts})

```

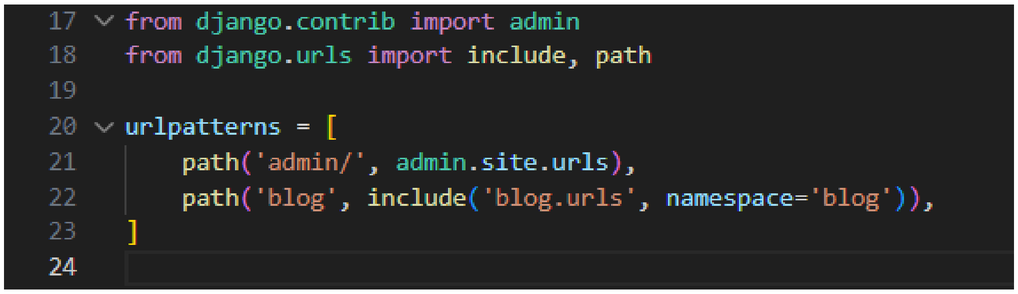


```python
def post_detail(request, year, month, day, post);
    post = get_object_or_484(Post, slug=post, status='published',
        publish__year=year,
        publish__month=month,
        publish__day=day)
    return render(request,
                  'blog/post/detail.html',
                  {'post': post})
```

Creating List and Detail Views

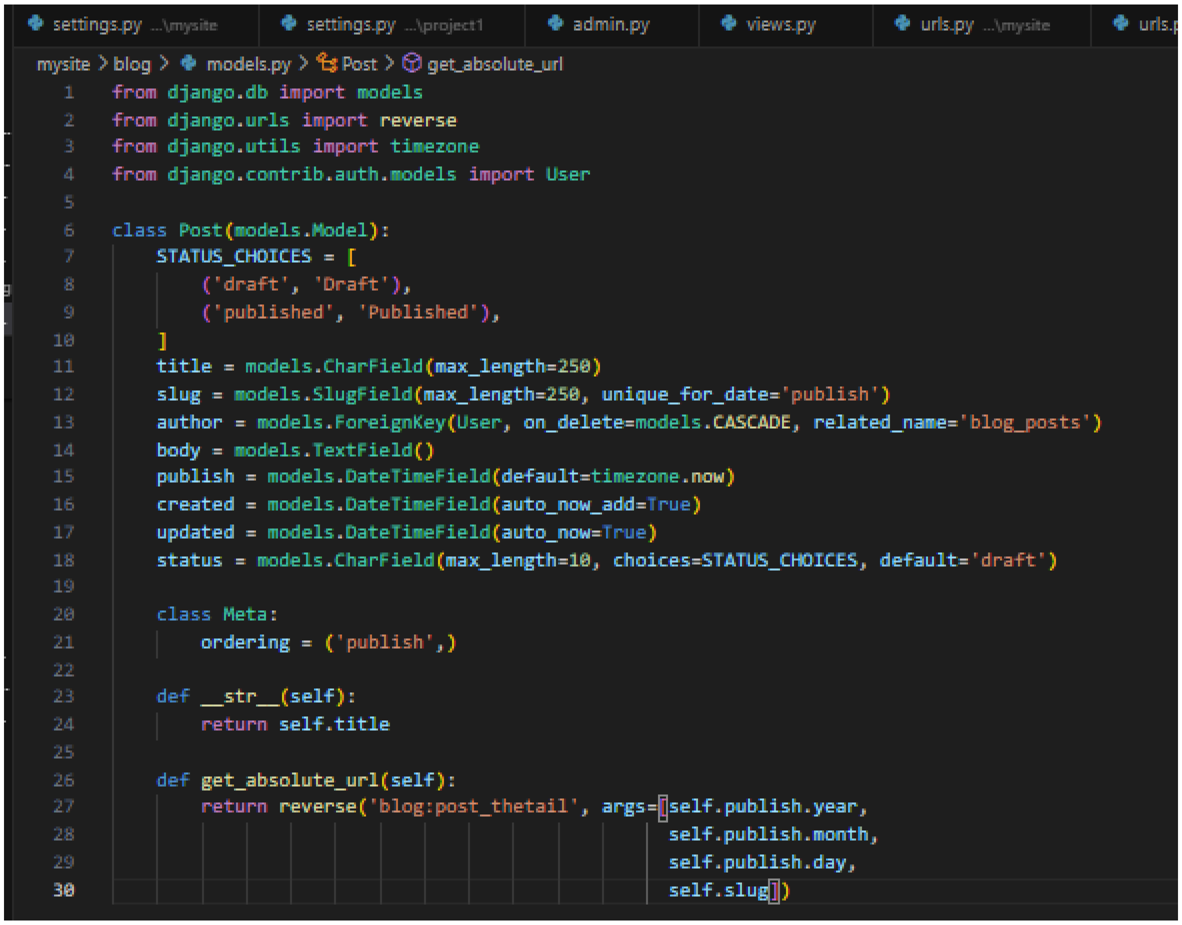


```python
from django.urls import path
from . import views

app_name = 'blog'
urlpatterns = [
    path('', views.post_list, name='post_list'),
    path('<int:year>/<int:month>/<int:day>/<slug:post>/',
         views.post_detail,
         name='post_detail')
]

```



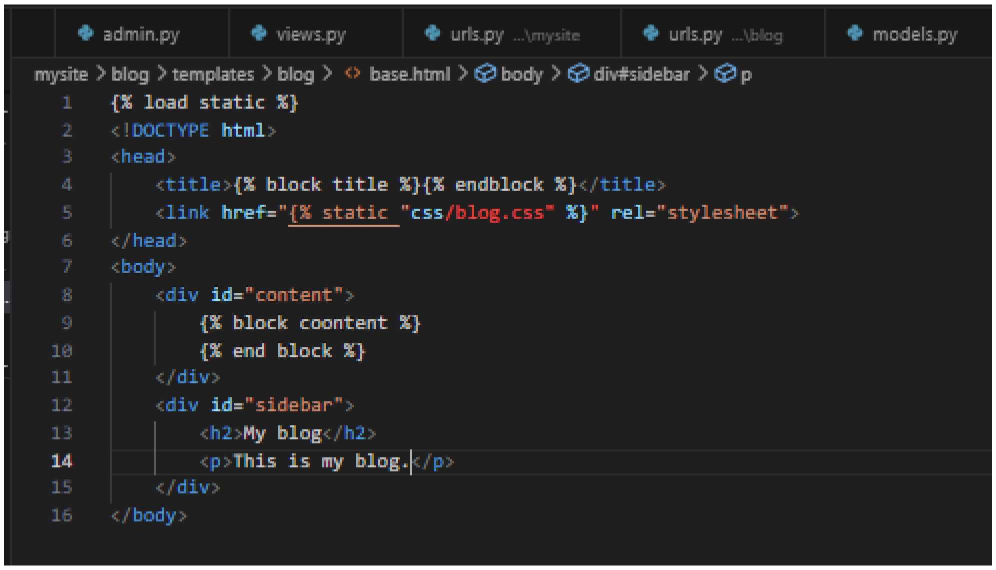
```python
from django.contrib import admin
from django.urls import include, path

urlpatterns = [
    path('admin/', admin.site.urls),
    path('blog', include('blog.urls', namespace='blog')),
]

```

Adding URL Patterns for your Views 28



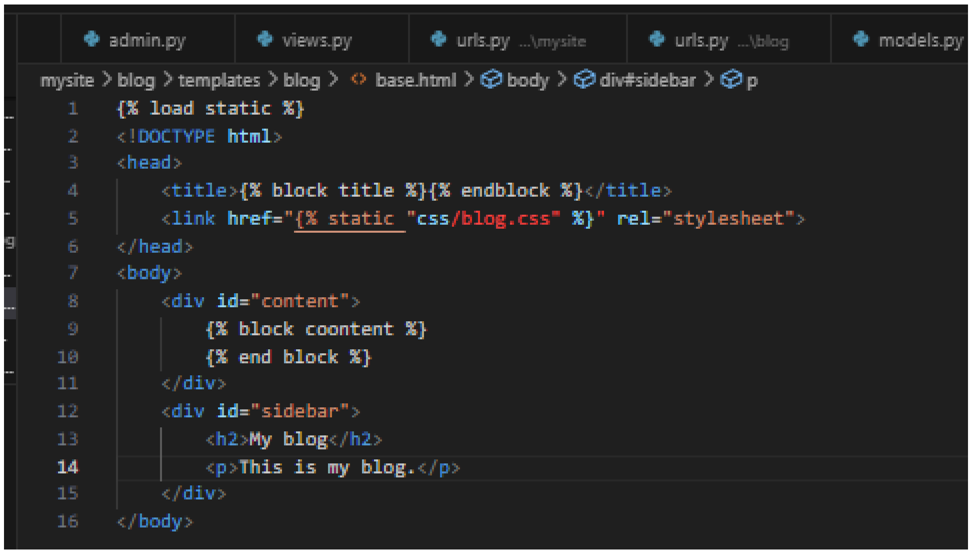
```python
from django.db import models
from django.urls import reverse
from django.utils import timezone
from django.contrib.auth.models import User

class Post(models.Model):
    STATUS_CHOICES = [
        ('draft', 'Draft'),
        ('published', 'Published'),
    ]
    title = models.CharField(max_length=250)
    slug = models.SlugField(max_length=250, unique_for_date='publish')
    author = models.ForeignKey(User, on_delete=models.CASCADE, related_name='blog_posts')
    body = models.TextField()
    publish = models.DateTimeField(default=timezone.now)
    created = models.DateTimeField(auto_now_add=True)
    updated = models.DateTimeField(auto_now=True)
    status = models.CharField(max_length=10, choices=STATUS_CHOICES, default='draft')

    class Meta:
        ordering = ('publish',)

    def __str__(self):
        return self.title

    def get_absolute_url(self):
        return reverse('blog:post_thetail', args=[self.publish.year,
                                                  self.publish.month,
                                                  self.publish.day,
                                                  self.slug])
```

Canonical URLs for Models


admin.py
views.py
urls.py ...\mysite
urls.py ...\blog
models.py
mysite > blog > templates > blog > base.html > body > div#sidebar > p

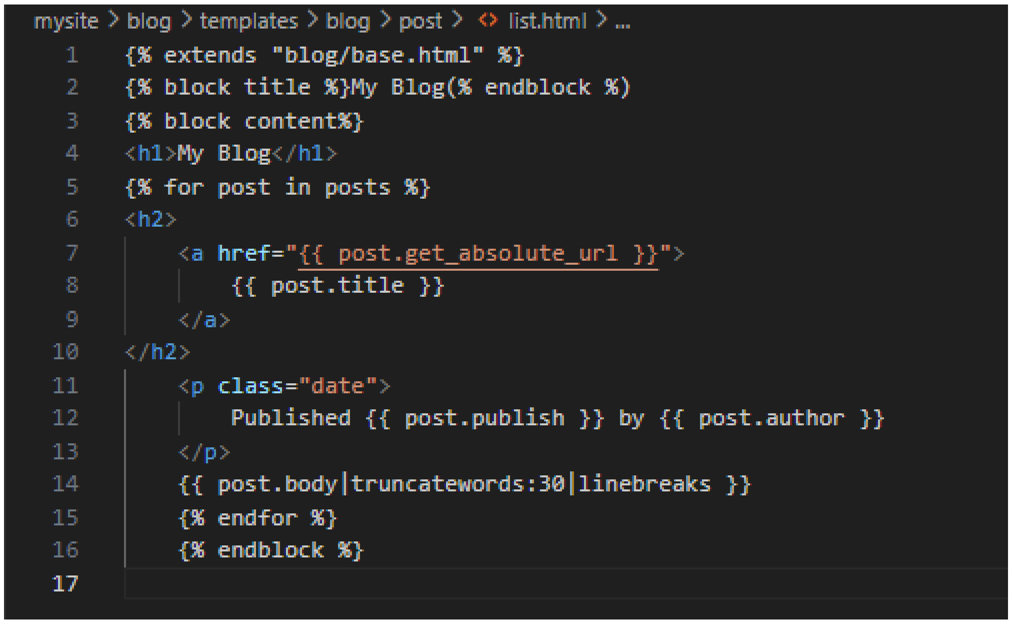
```html
{% load static %}
<!DOCTYPE html>
<head>
    <title>{% block title %}{% endblock %}</title>
    <link href="{% static "css/blog.css" %}" rel="stylesheet">
</head>
<body>
    <div id="content">
        {% block coontent %}
        {% end block %}
    </div>
    <div id="sidebar">
        <h2>My blog</h2>
        <p>This is my blog.</p>
    </div>
</body>
```

Creating Templates for Views


admin.py
views.py
urls.py ...\mysite
urls.py ...\blog
models.py
mysite > blog > templates > blog > base.html > body > div#sidebar > p

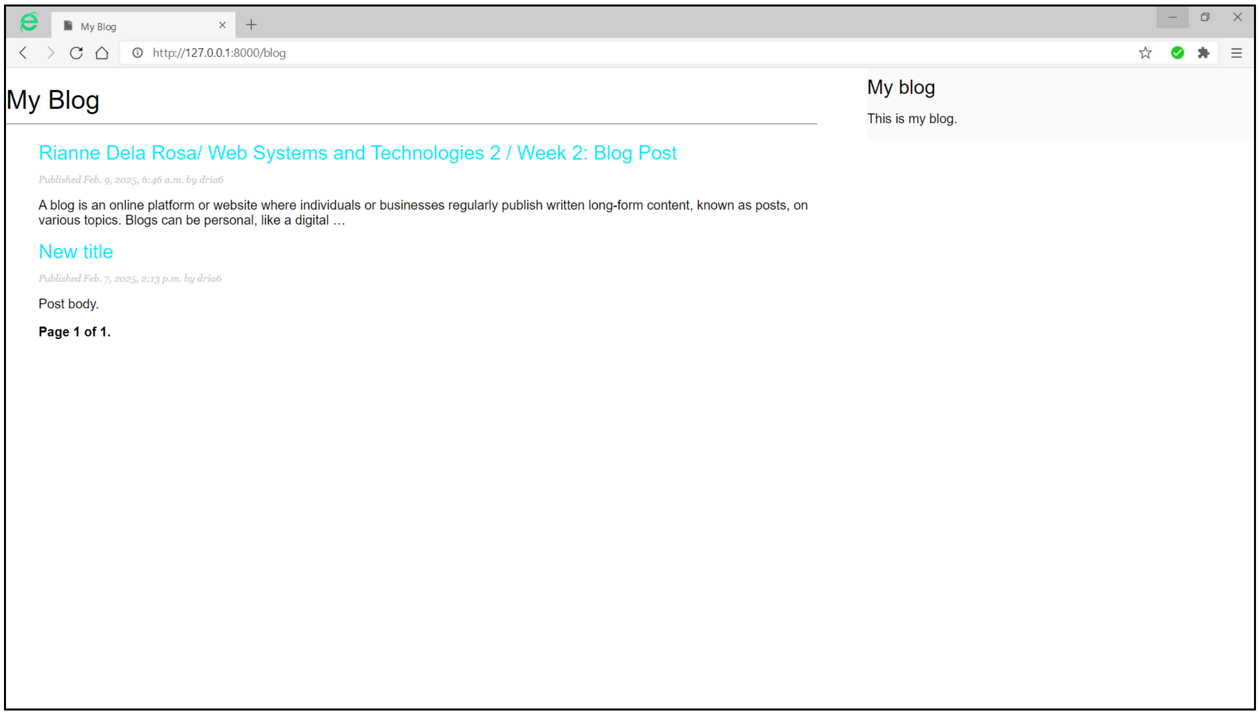
```
{% load static %}
<!DOCTYPE html>
<head>
    <title>{% block title %}{% endblock %}</title>
    <link href="{% static "css/blog.css" %}" rel="stylesheet">
</head>
<body>
    <div id="content">
        {% block coontent %}
        {% end block %}
    </div>
    <div id="sidebar">
        <h2>My blog</h2>
        <p>This is my blog.</p>
    </div>
</body>
```

Editing the base.html



```
{% extends "blog/base.html" %}
{% block title %}My Blog(% endblock %)
{% block content%}
<h1>My Blog</h1>
{% for post in posts %}
<h2>
    <a href="{{ post.get_absolute_url }}">
        {{ post.title }}
    </a>
</h2>
    <p class="date">
        Published {{ post.publish }} by {{ post.author }}
    </p>
    {{ post.body|truncatewords:30|linebreaks }}
    {% endfor %}
    {% endblock %}

```

Editing the list.html



My Blog

Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post

Published Feb. 9, 2025, 6:46 a.m. by dria6

A blog is an online platform or website where individuals or businesses regularly publish written long-form content, known as posts, on various topics. Blogs can be personal, like a digital …

New title

Published Feb. 7, 2025, 2:13 p.m. by dria6

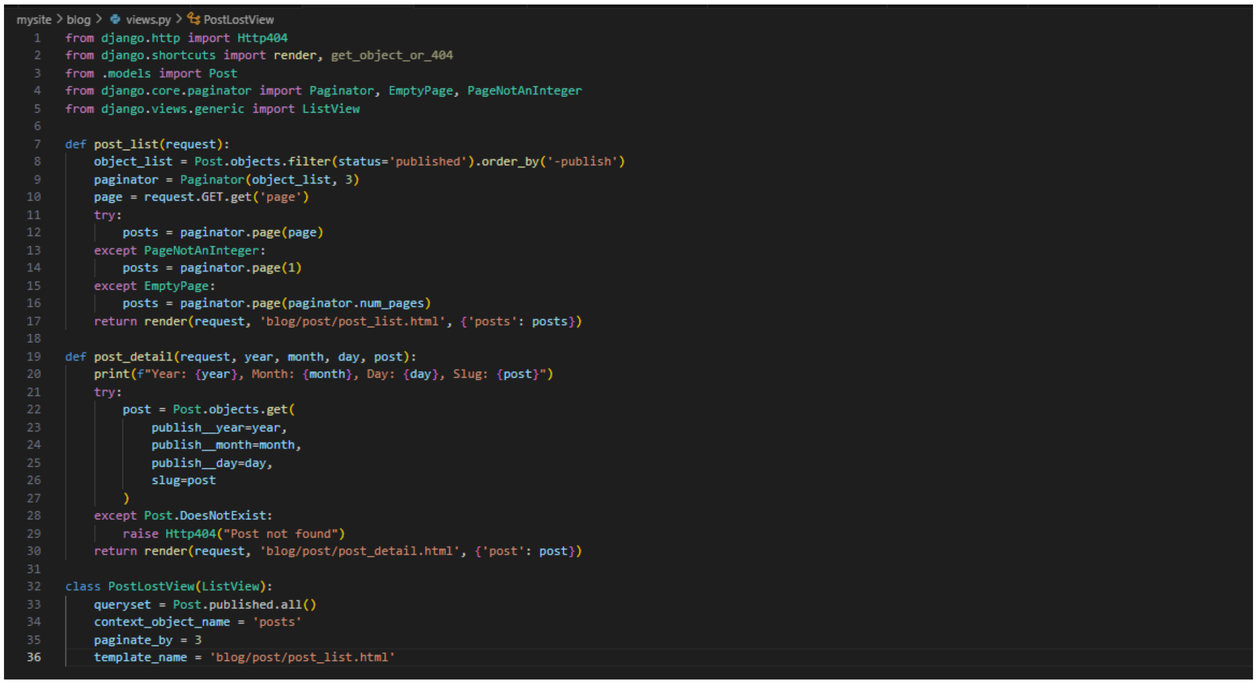
Post body.

Page 1 of 1.

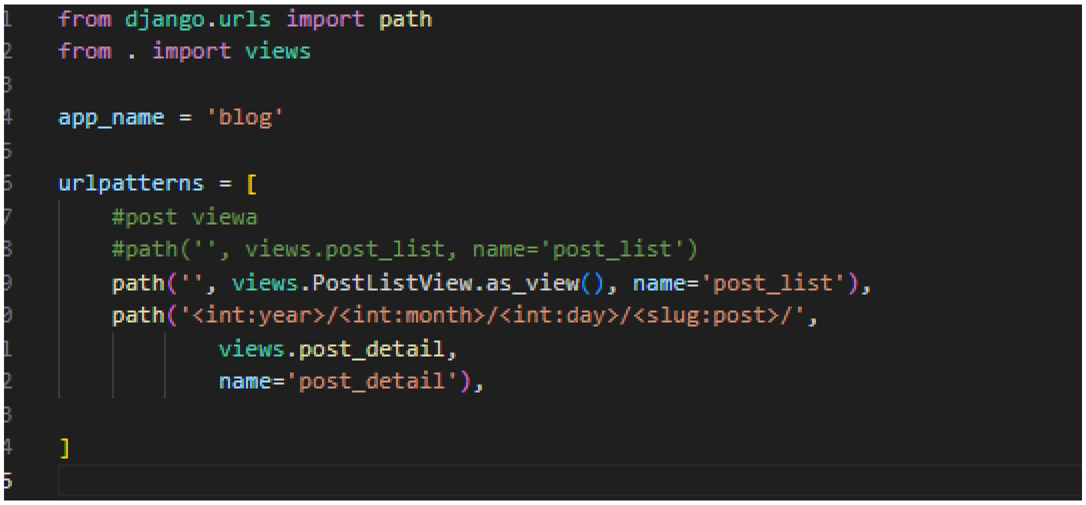
My blog

This is my blog.

Output for Views

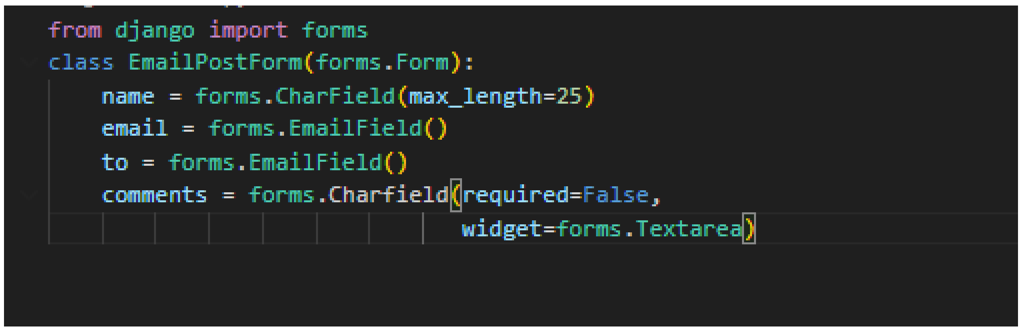


```python
from django.http import Http404
from django.shortcuts import render, get_object_or_404
from .models import Post
from django.core.paginator import Paginator, EmptyPage, PageNotAnInteger
from django.views.generic import ListView

def post_list(request):
    object_list = Post.objects.filter(status='published').order_by('-publish')
    paginator = Paginator(object_list, 3)
    page = request.GET.get('page')
    try:
        posts = paginator.page(page)
    except PageNotAnInteger:
        posts = paginator.page(1)
    except EmptyPage:
        posts = paginator.page(paginator.num_pages)
    return render(request, 'blog/post/post_list.html', {'posts': posts})

def post_detail(request, year, month, day, post):
    print(f"Year: {year}, Month: {month}, Day: {day}, Slug: {post}")
    try:
        post = Post.objects.get(
            publish__year=year,
            publish__month=month,
            publish__day=day,
            slug=post
        )
    except Post.DoesNotExist:
        raise Http404("Post not found")
    return render(request, 'blog/post/post_detail.html', {'post': post})

class PostLostView(ListView):
    queryset = Post.published.all()
    context_object_name = 'posts'
    paginate_by = 3
    template_name = 'blog/post/post_list.html'
```



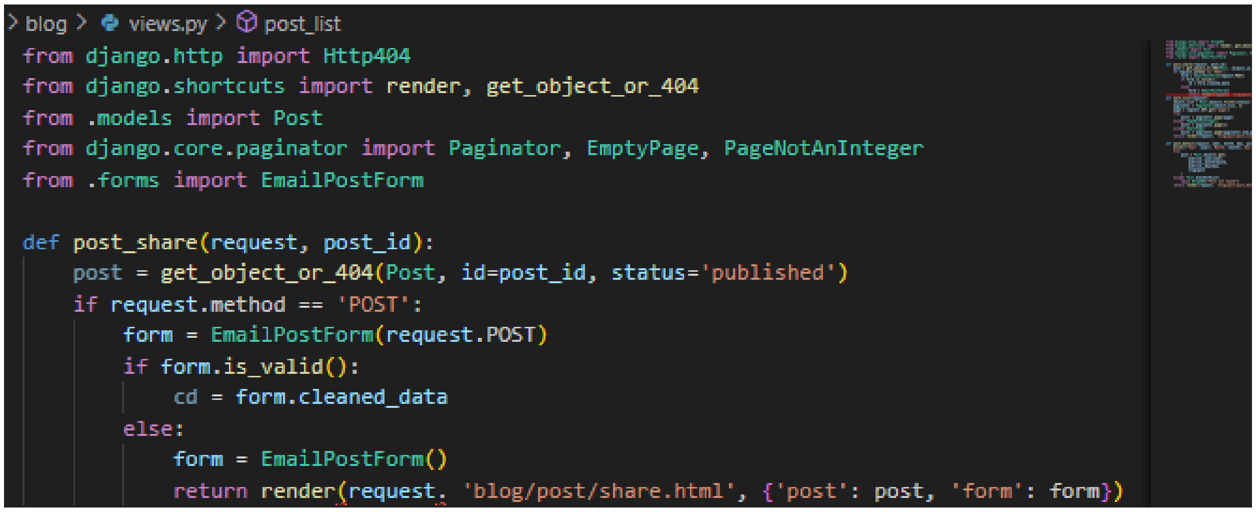
```python
from django.urls import path
from . import views

app_name = 'blog'

urlpatterns = [
    #post viewa
    #path('', views.post_list, name='post_list')
    path('', views.PostListView.as_view(), name='post_list'),
    path('<int:year>/<int:month>/<int:day>/<slug:post>/',
            views.post_detail,
            name='post_detail'),

]
```

Using Class-based Views



```python
from django import forms
class EmailPostForm(forms.Form):
    name = forms.CharField(max_length=25)
    email = forms.EmailField()
    to = forms.EmailField()
    comments = forms.Charfield(required=False,
                                widget=forms.Textarea)
```

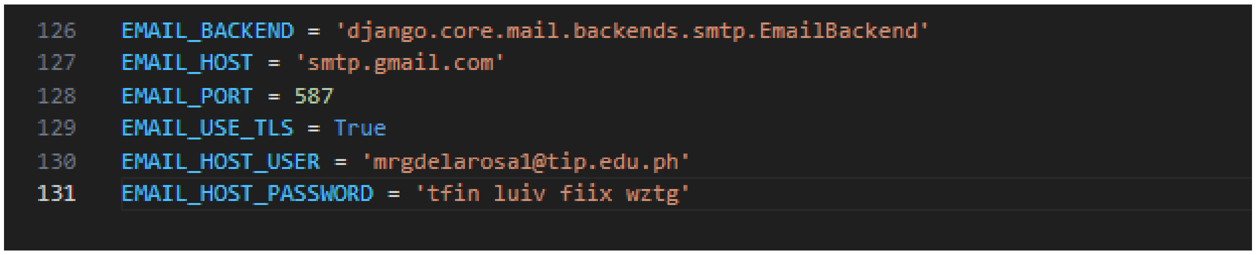
Creating forms.py



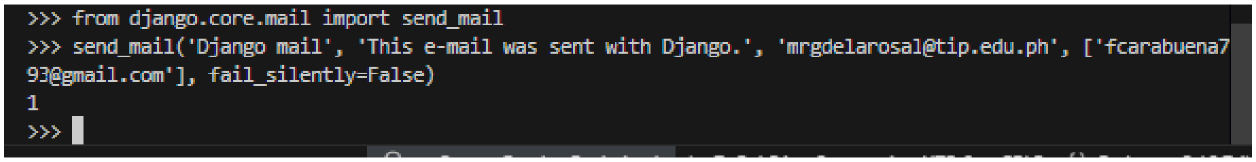
blog > views.py > post_list

```python
from django.http import Http404
from django.shortcuts import render, get_object_or_404
from .models import Post
from django.core.paginator import Paginator, EmptyPage, PageNotAnInteger
from .forms import EmailPostForm

def post_share(request, post_id):
    post = get_object_or_404(Post, id=post_id, status='published')
    if request.method == 'POST':
        form = EmailPostForm(request.POST)
        if form.is_valid():
            cd = form.cleaned_data
        else:
            form = EmailPostForm()
            return render(request, 'blog/post/share.html', {'post': post, 'form': form})
```

Handling Forms in Views

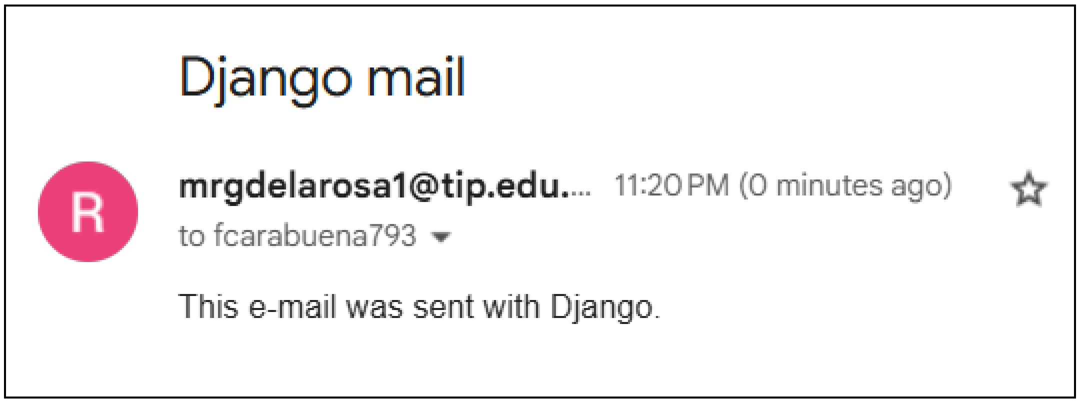


```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 587
EMAIL_USE_TLS = True
EMAIL_HOST_USER = 'mrgdelarosa1@tip.edu.ph'
EMAIL_HOST_PASSWORD = 'tfin luiv fiix wztg'
```



```
>>> from django.core.mail import send_mail
>>> send_mail('Django mail', 'This e-mail was sent with Django.', 'mrgdelarosa1@tip.edu.ph', ['fcarabuena7
93@gmail.com'], fail_silently=False)
1
>>> 
```

Sending Email with Django



Django mail

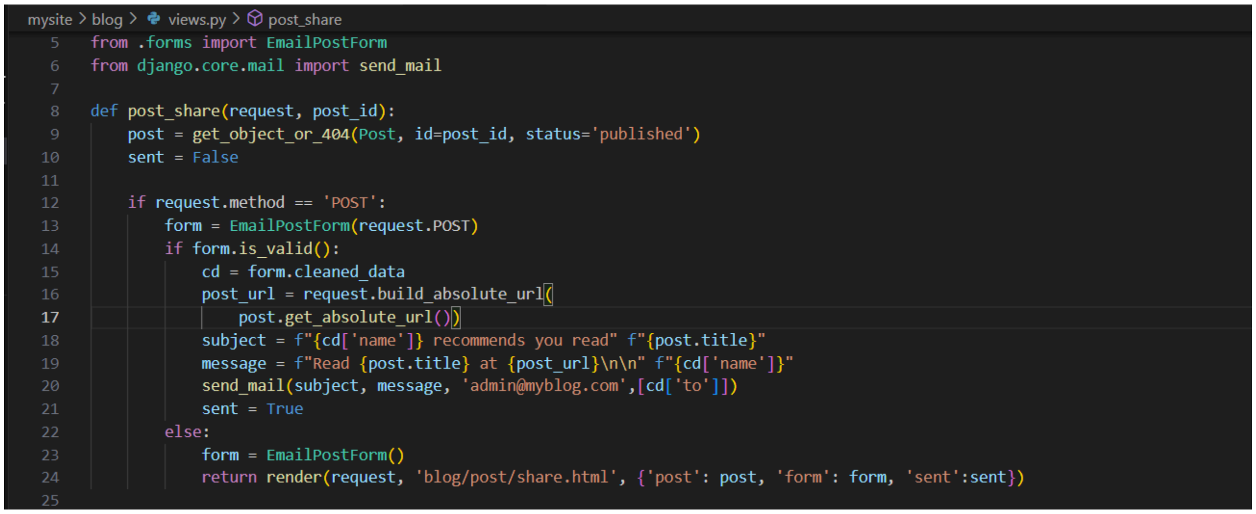
mrgdelarosa1@tip.edu.... 11:20 PM (0 minutes ago)

to fcarabuena793

This e-mail was sent with Django.

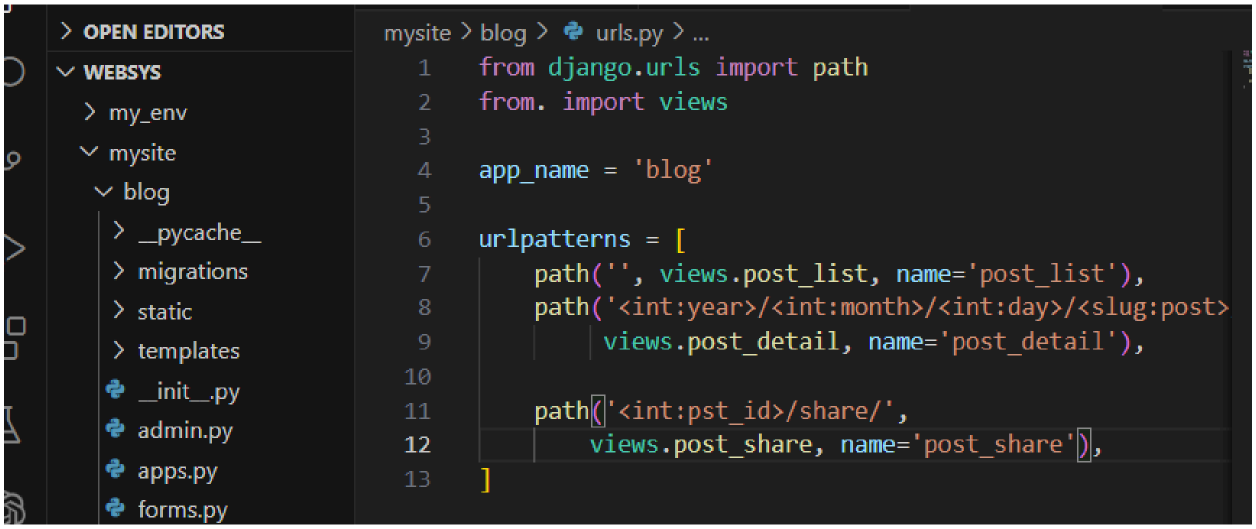
0.

Sent Email



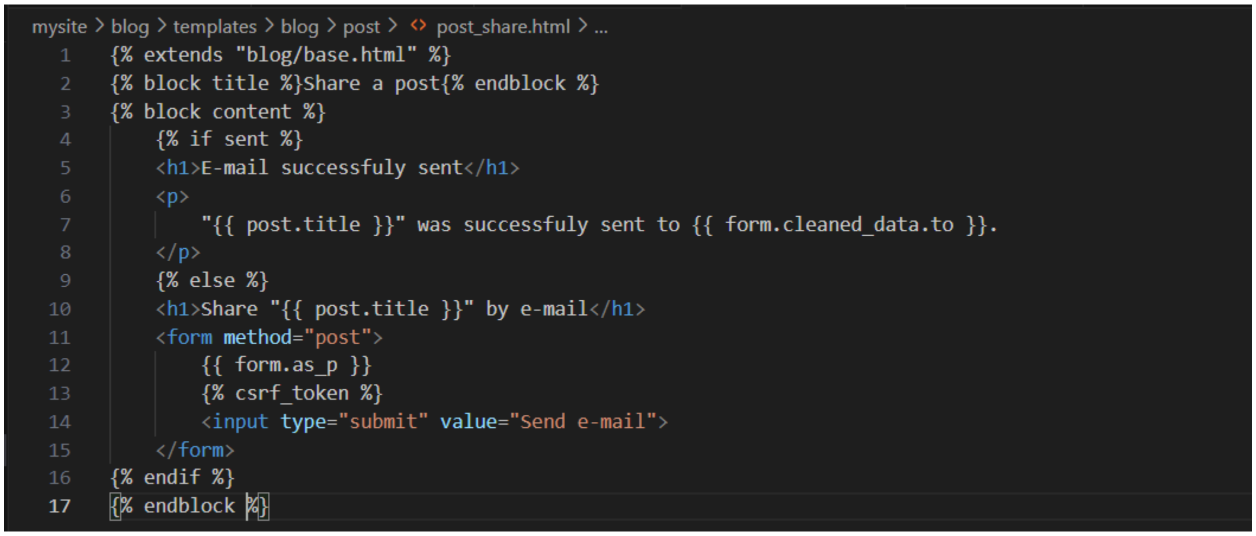
```
from .forms import EmailPostForm
from django.core.mail import send_mail

def post_share(request, post_id):
    post = get_object_or_404(Post, id=post_id, status='published')
    sent = False

    if request.method == 'POST':
        form = EmailPostForm(request.POST)
        if form.is_valid():
            cd = form.cleaned_data
            post_url = request.build_absolute_url(
                post.get_absolute_url())
            subject = f"{cd['name']} recommends you read" f"{post.title}"
            message = f"Read {post.title} at {post_url}\n\n" f"{cd['name']}"
            send_mail(subject, message, 'admin@myblog.com',[cd['to']])
            sent = True
        else:
            form = EmailPostForm()
            return render(request, 'blog/post/share.html', {'post': post, 'form': form, 'sent':sent})
```

Editing the post_share view in the views.py file of the blog application



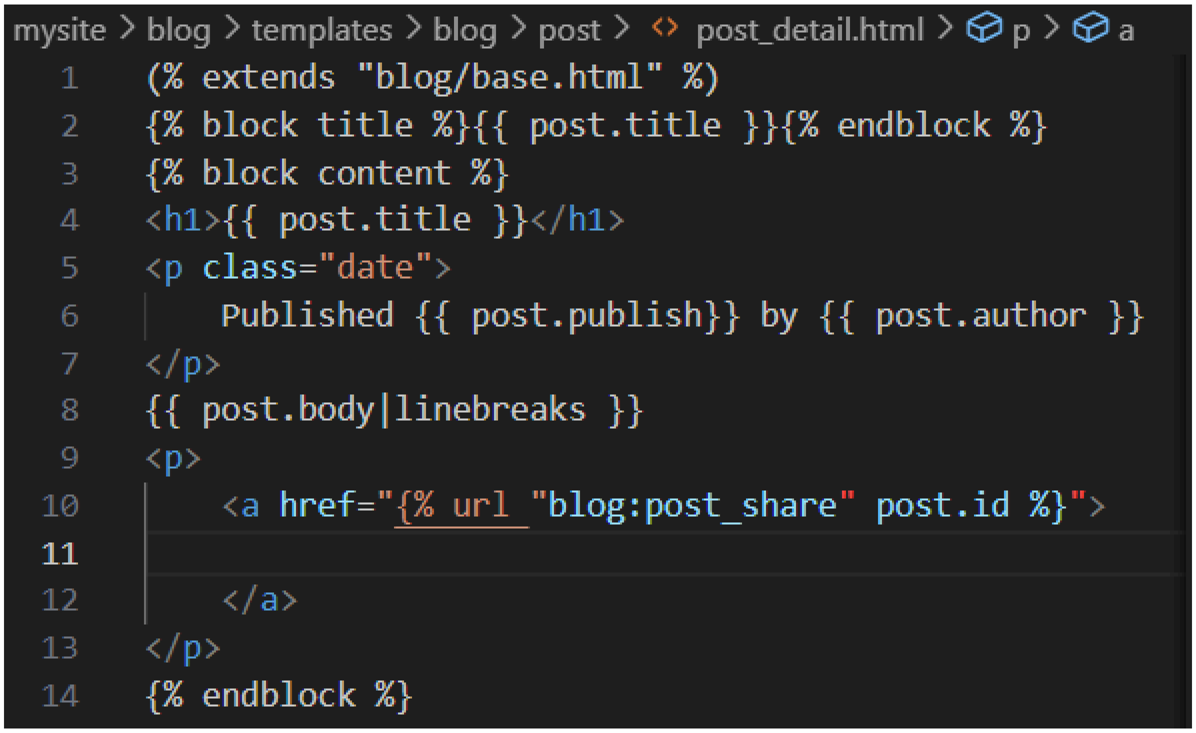
```python
from django.urls import path
from. import views

app_name = 'blog'

urlpatterns = [
    path('', views.post_list, name='post_list'),
    path('<int:year>/<int:month>/<int:day>/<slug:post>
         views.post_detail, name='post_detail'),

    path('<int:pst_id>/share/',
        views.post_share, name='post_share'),
]
```

Adding the post_share URL pattern


mysite > blog > templates > blog > post > post_share.html > ...

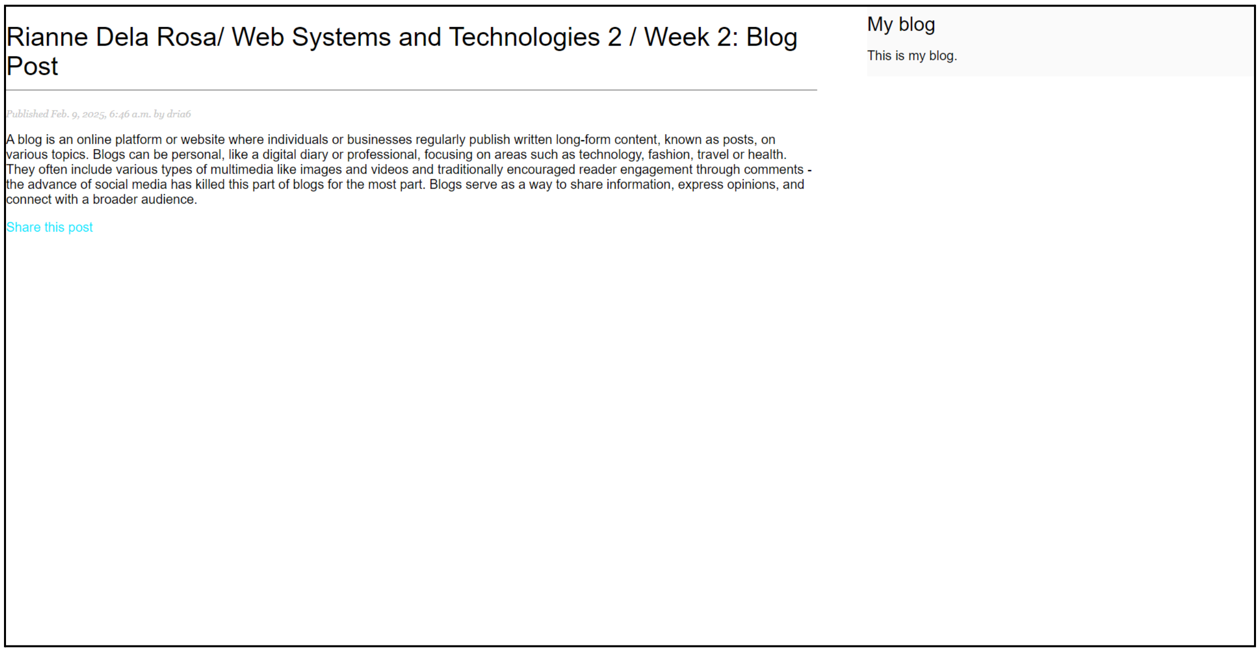
```html
{% extends "blog/base.html" %}
{% block title %}Share a post{% endblock %}
{% block content %}
    {% if sent %}
    <h1>E-mail successfuly sent</h1>
    <p>
        "{{ post.title }}" was successfuly sent to {{ form.cleaned_data.to }}.
    </p>
    {% else %}
    <h1>Share "{{ post.title }}" by e-mail</h1>
    <form method="post">
        {{ form.as_p }}
        {% csrf_token %}
        <input type="submit" value="Send e-mail">
    </form>
{% endif %}
{% endblock %}
```

Creating the of post_share


mysite > blog > templates > blog > post > post_detail.html > p > a

```
(% extends "blog/base.html" %)
{% block title %}{{ post.title }}{% endblock %}
{% block content %}
<h1>{{ post.title }}</h1>
<p class="date">
    Published {{ post.publish}} by {{ post.author }}
</p>
{{ post.body|linebreaks }}
<p>
    <a href="{% url "blog:post_share" post.id %}">

    </a>
</p>
{% endblock %}
```

Editing the blog/post/post_detail



Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post

Published Feb. 9, 2025, 6:46 a.m. by dria6

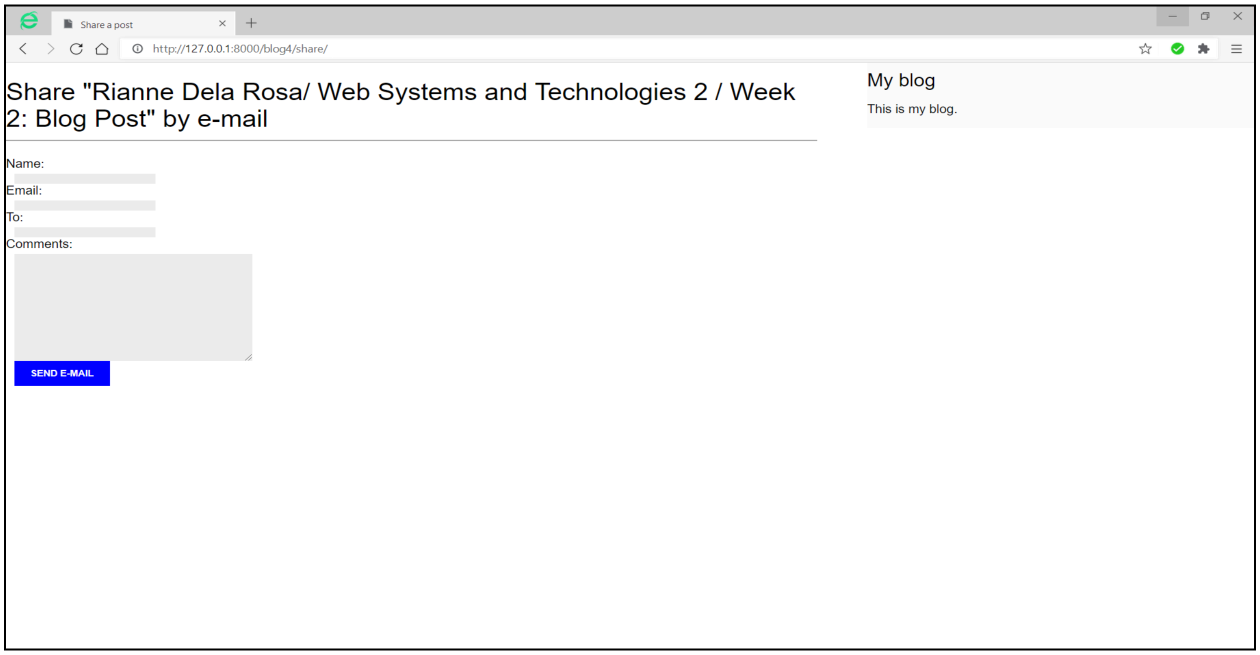
A blog is an online platform or website where individuals or businesses regularly publish written long-form content, known as posts, on various topics. Blogs can be personal, like a digital diary or professional, focusing on areas such as technology, fashion, travel or health. They often include various types of multimedia like images and videos and traditionally encouraged reader engagement through comments - the advance of social media has killed this part of blogs for the most part. Blogs serve as a way to share information, express opinions, and connect with a broader audience.

Share this post

My blog

This is my blog.

Blog Page



Share "Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post" by e-mail

Name:

Email:

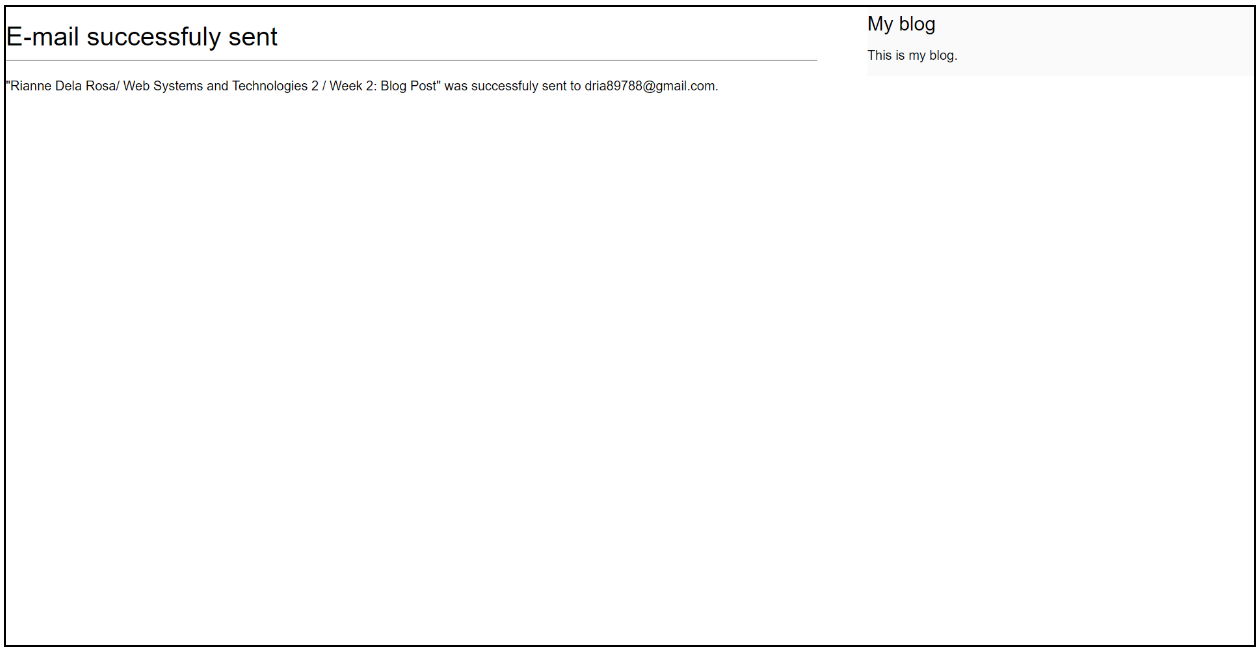
To:

Comments:

My blog

This is my blog.

Share page

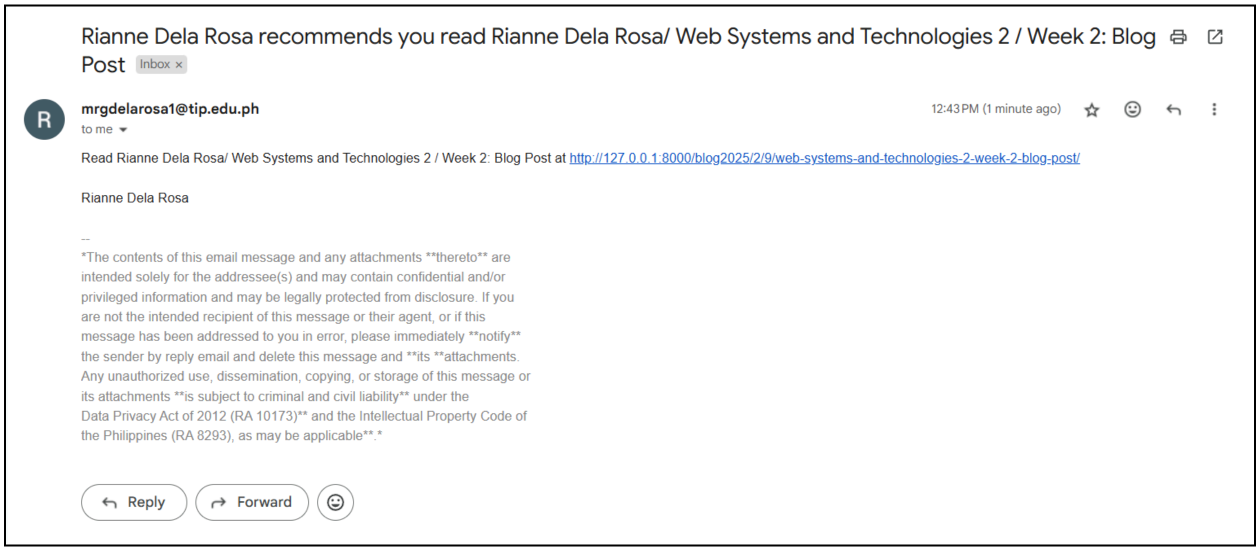


E-mail successfuly sent

"Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post" was successfuly sent to dria89788@gmail.com.

My blog

This is my blog.


Rianne Dela Rosa recommends you read Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post

mrgdelarosa1@tip.edu.ph
to me

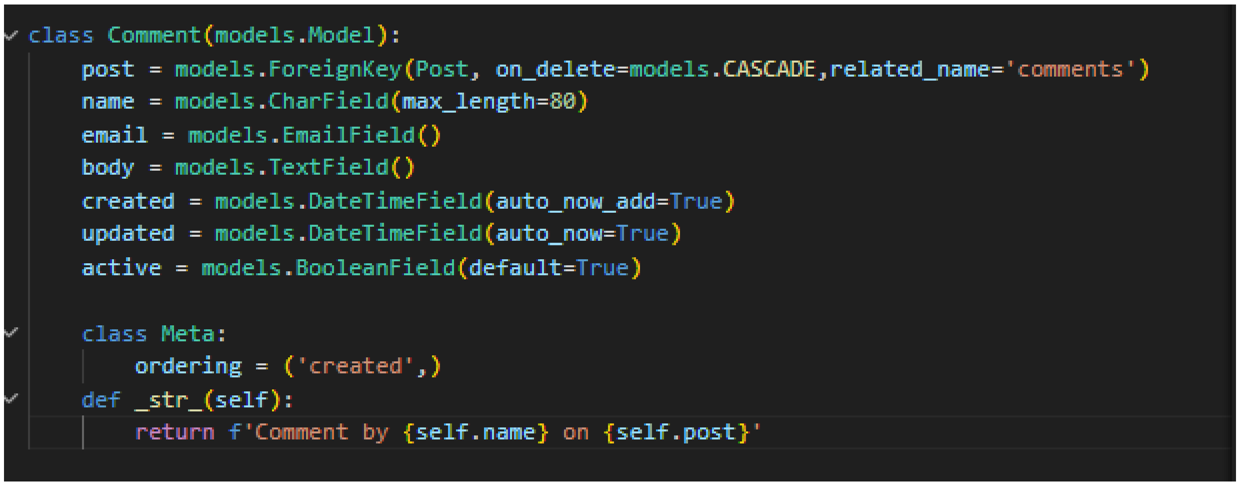
12:43 PM (1 minute ago)

Read Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post at http://127.0.0.1:8000/blog2025/2/9/web-systems-and-technologies-2-week-2-blog-post/

Rianne Dela Rosa

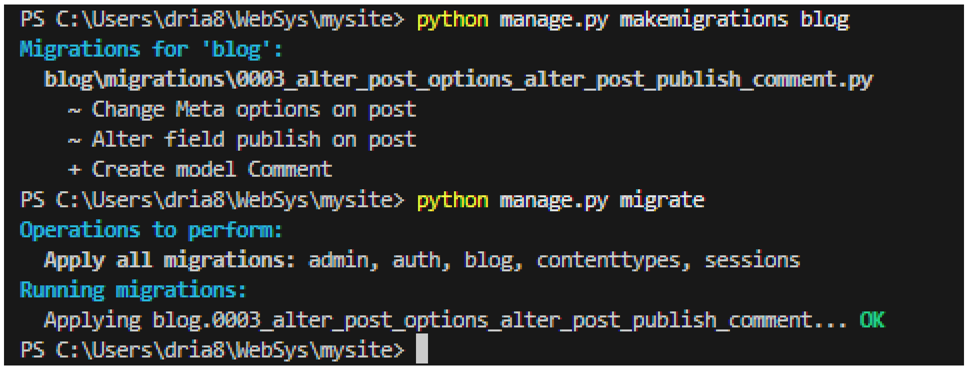
--
*The contents of this email message and any attachments **thereto** are intended solely for the addressee(s) and may contain confidential and/or privileged information and may be legally protected from disclosure. If you are not the intended recipient of this message or their agent, or if this message has been addressed to you in error, please immediately **notify** the sender by reply email and delete this message and **its **attachments. Any unauthorized use, dissemination, copying, or storage of this message or its attachments **is subject to criminal and civil liability** under the Data Privacy Act of 2012 (RA 10173)** and the Intellectual Property Code of the Philippines (RA 8293), as may be applicable**.*

Email successfully sent



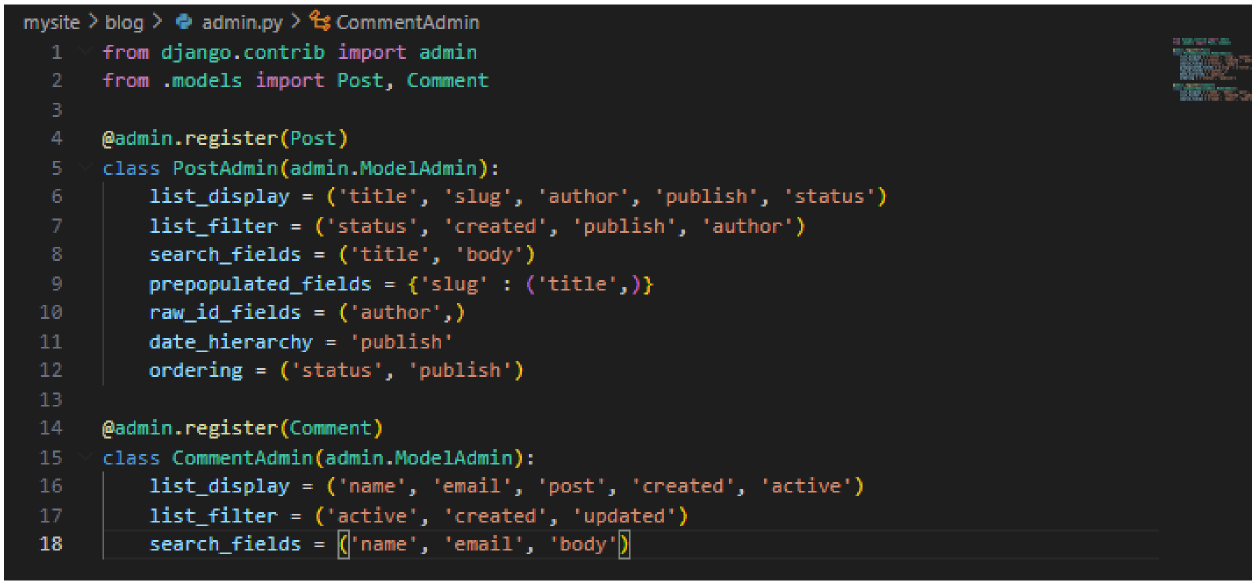
```python
class Comment(models.Model):
    post = models.ForeignKey(Post, on_delete=models.CASCADE,related_name='comments')
    name = models.CharField(max_length=80)
    email = models.EmailField()
    body = models.TextField()
    created = models.DateTimeField(auto_now_add=True)
    updated = models.DateTimeField(auto_now=True)
    active = models.BooleanField(default=True)

    class Meta:
        ordering = ('created',)
    def _str_(self):
        return f'Comment by {self.name} on {self.post}'
```

Editing models.py



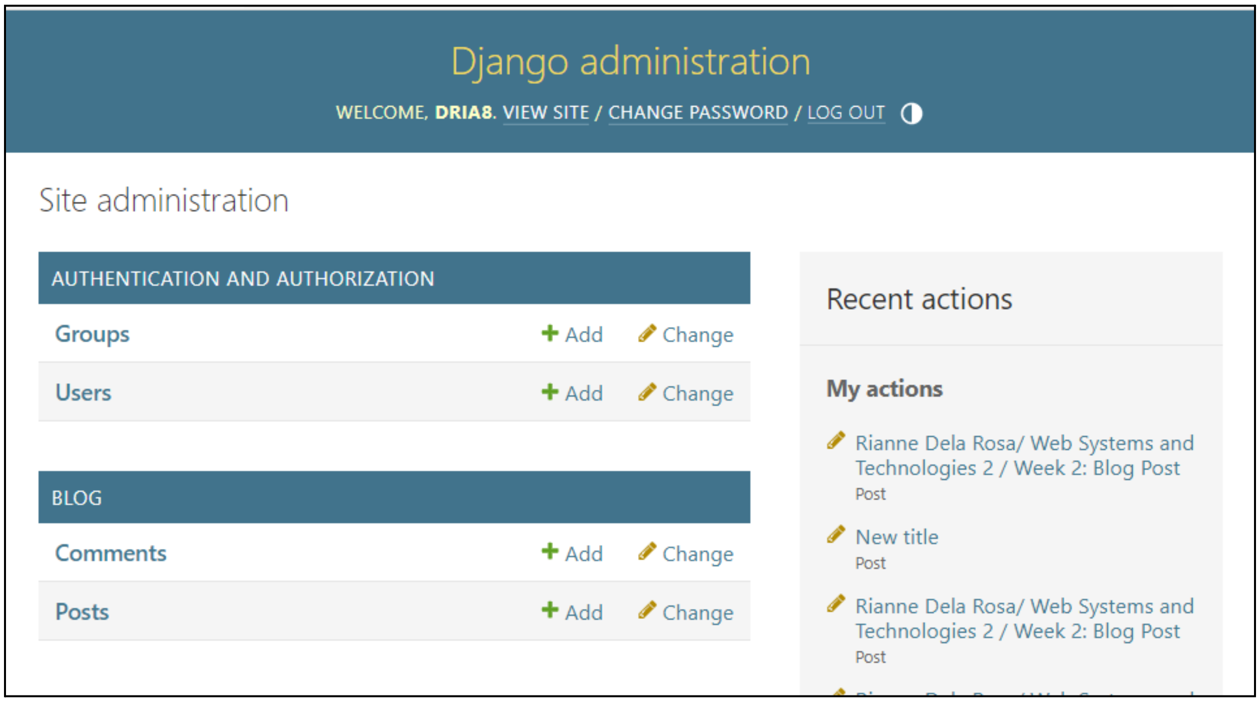
```
PS C:\Users\dria8\WebSys\mysite> python manage.py makemigrations blog
Migrations for 'blog':
  blog\migrations\0003_alter_post_options_alter_post_publish_comment.py
    ~ Change Meta options on post
    ~ Alter field publish on post
    + Create model Comment
PS C:\Users\dria8\WebSys\mysite> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, blog, contenttypes, sessions
Running migrations:
  Applying blog.0003_alter_post_options_alter_post_publish_comment... OK
PS C:\Users\dria8\WebSys\mysite> 
```

Creating migrations

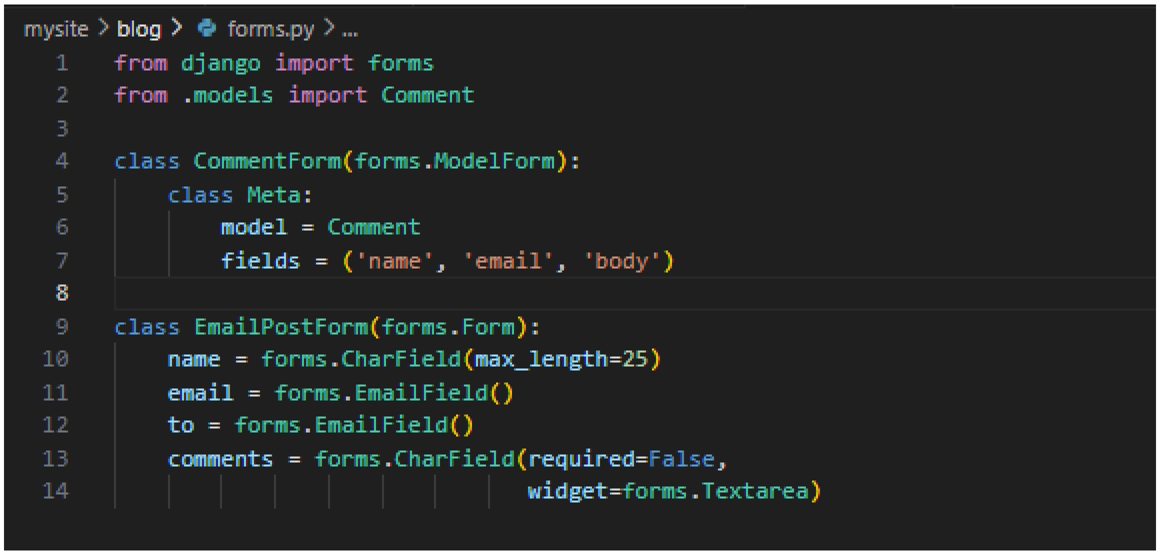


```python
from django.contrib import admin
from .models import Post, Comment

@admin.register(Post)
class PostAdmin(admin.ModelAdmin):
    list_display = ('title', 'slug', 'author', 'publish', 'status')
    list_filter = ('status', 'created', 'publish', 'author')
    search_fields = ('title', 'body')
    prepopulated_fields = {'slug' : ('title',)}
    raw_id_fields = ('author',)
    date_hierarchy = 'publish'
    ordering = ('status', 'publish')

@admin.register(Comment)
class CommentAdmin(admin.ModelAdmin):
    list_display = ('name', 'email', 'post', 'created', 'active')
    list_filter = ('active', 'created', 'updated')
    search_fields = ('name', 'email', 'body')
```

Editing admin.py

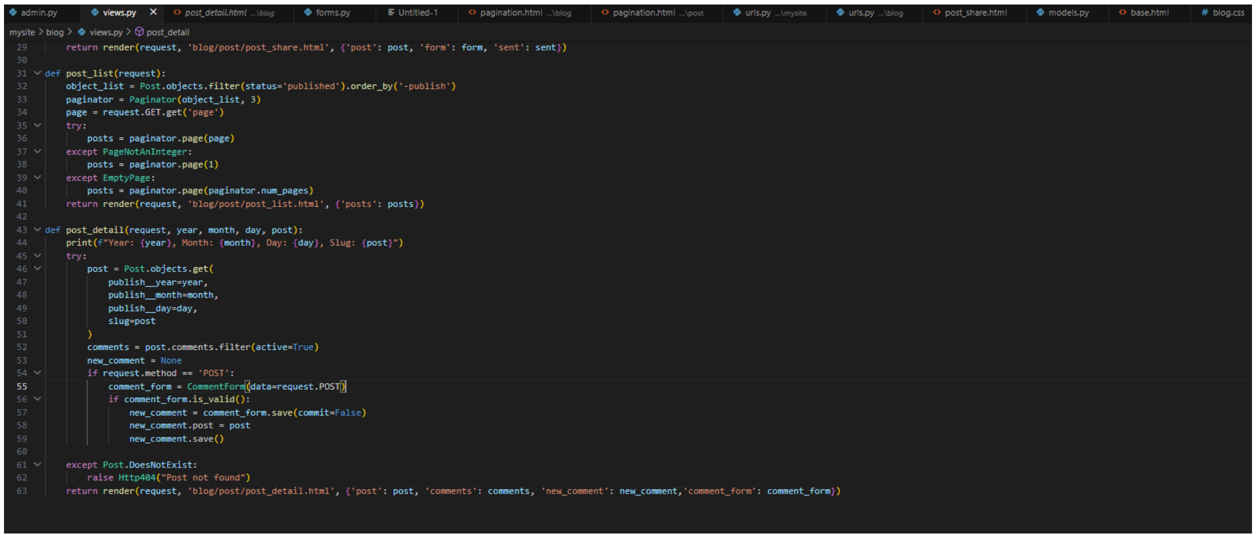

Django administration
WELCOME, DRIA8. VIEW SITE / CHANGE PASSWORD / LOG OUT
Site administration
AUTHENTICATION AND AUTHORIZATION
Groups + Add Change
Users + Add Change
BLOG
Comments + Add Change
Posts + Add Change
Recent actions
My actions
Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post
Post
New title
Post
Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post
Post

Comment in admin



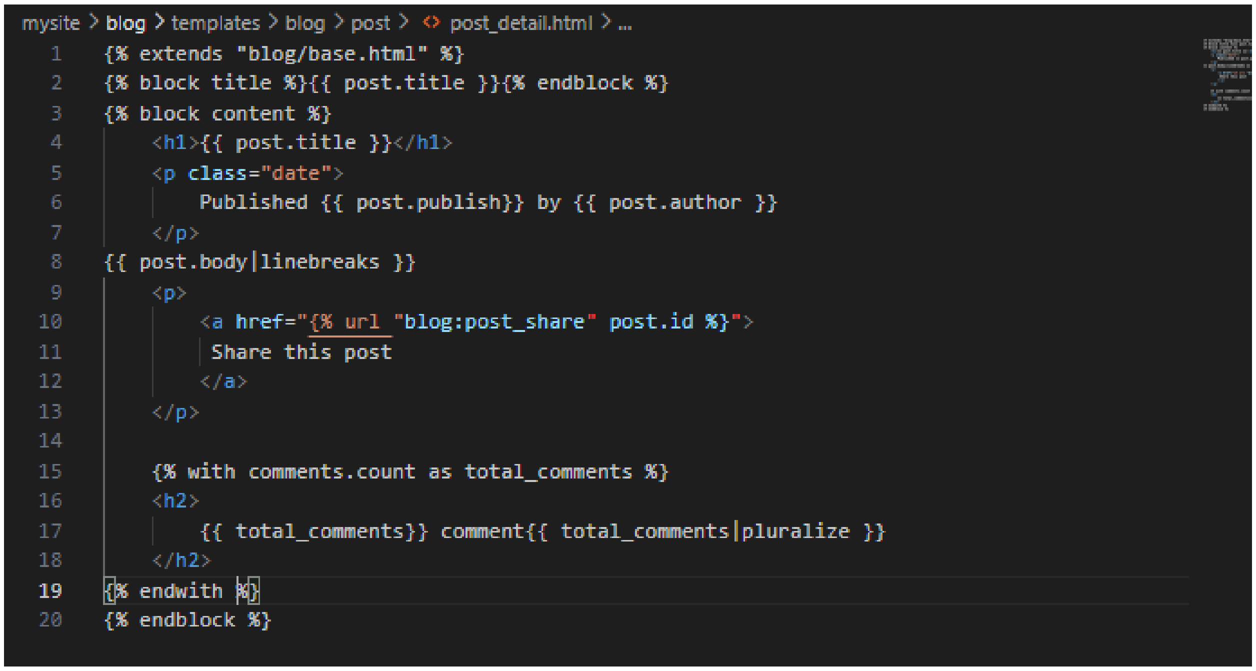
```python
from django import forms
from .models import Comment

class CommentForm(forms.ModelForm):
    class Meta:
        model = Comment
        fields = ('name', 'email', 'body')

class EmailPostForm(forms.Form):
    name = forms.CharField(max_length=25)
    email = forms.EmailField()
    to = forms.EmailField()
    comments = forms.CharField(required=False,
                               widget=forms.Textarea)
```

Creating forms from models



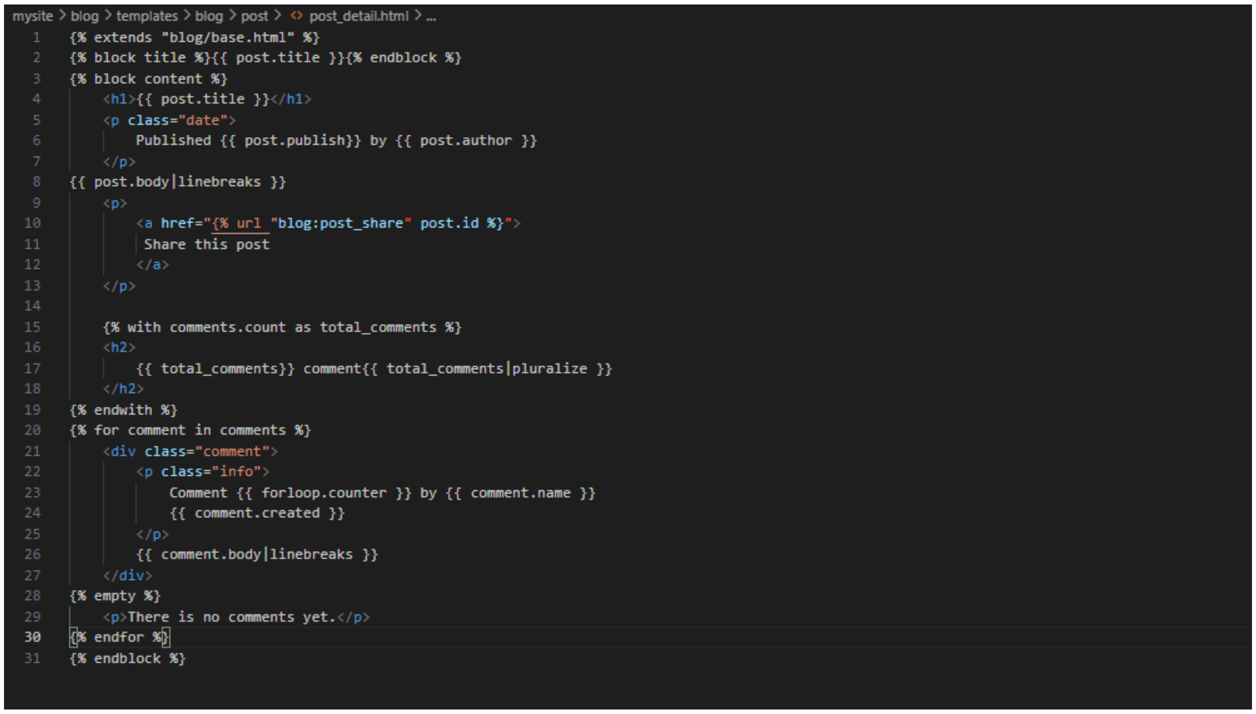
```python
    return render(request, 'blog/post/post_share.html', {'post': post, 'form': form, 'sent': sent})

def post_list(request):
    object_list = Post.objects.filter(status='published').order_by('-publish')
    paginator = Paginator(object_list, 3)
    page = request.GET.get('page')
    try:
        posts = paginator.page(page)
    except PageNotAnInteger:
        posts = paginator.page(1)
    except EmptyPage:
        posts = paginator.page(paginator.num_pages)
    return render(request, 'blog/post/post_list.html', {'posts': posts})

def post_detail(request, year, month, day, post):
    print(f"Year: {year}, Month: {month}, Day: {day}, Slug: {post}")
    try:
        post = Post.objects.get(
            publish__year=year,
            publish__month=month,
            publish__day=day,
            slug=post
        )
        comments = post.comments.filter(active=True)
        new_comment = None
        if request.method == 'POST':
            comment_form = CommentForm(data=request.POST)
            if comment_form.is_valid():
                new_comment = comment_form.save(commit=False)
                new_comment.post = post
                new_comment.save()

    except Post.DoesNotExist:
        raise Http404("Post not found")
    return render(request, 'blog/post/post_detail.html', {'post': post, 'comments': comments, 'new_comment': new_comment,'comment_form': comment_form})
```

Handling Modelforms in views



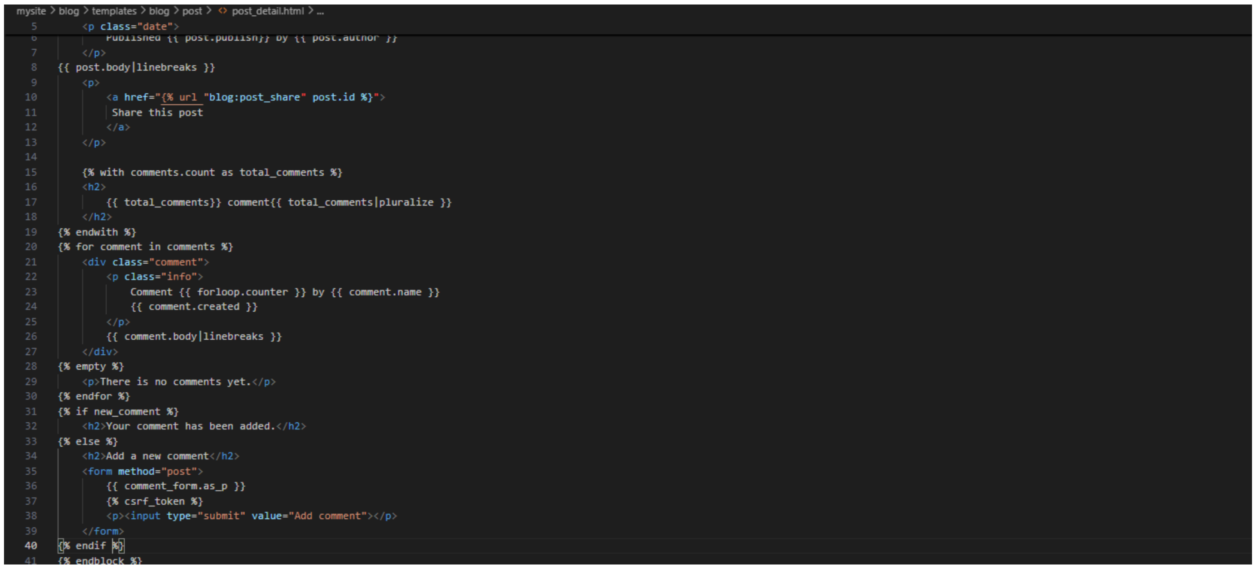
mysite > blog > templates > blog > post > post_detail.html > ...

```
{% extends "blog/base.html" %}
{% block title %}{{ post.title }}{% endblock %}
{% block content %}
    <h1>{{ post.title }}</h1>
    <p class="date">
        Published {{ post.publish}} by {{ post.author }}
    </p>
{{ post.body|linebreaks }}
    <p>
        <a href="{% url "blog:post_share" post.id %}">
         Share this post
        </a>
    </p>

    {% with comments.count as total_comments %}
    <h2>
        {{ total_comments}} comment{{ total_comments|pluralize }}
    </h2>
{% endwith %}
{% endblock %}
```



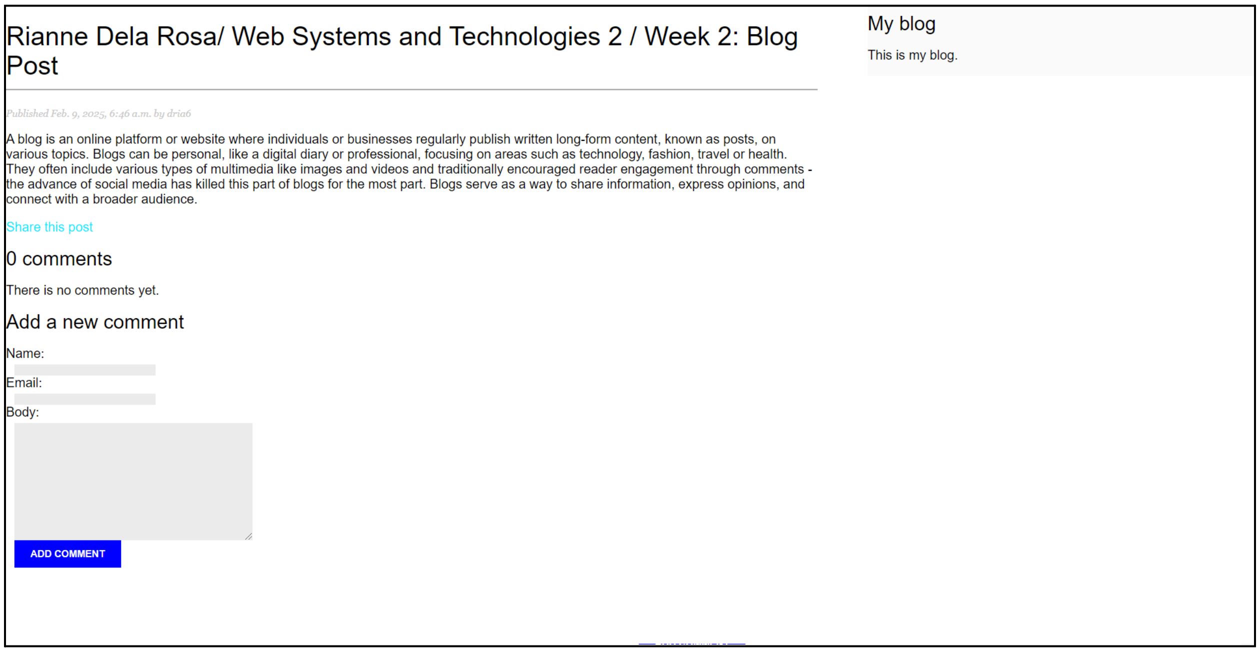
mysite > blog > templates > blog > post > post_detail.html > ...

```
{% extends "blog/base.html" %}
{% block title %}{{ post.title }}{% endblock %}
{% block content %}
    <h1>{{ post.title }}</h1>
    <p class="date">
        Published {{ post.publish}} by {{ post.author }}
    </p>
{{ post.body|linebreaks }}
    <p>
        <a href="{% url "blog:post_share" post.id %}">
         Share this post
        </a>
    </p>

    {% with comments.count as total_comments %}
    <h2>
        {{ total_comments}} comment{{ total_comments|pluralize }}
    </h2>
{% endwith %}
{% for comment in comments %}
    <div class="comment">
        <p class="info">
            Comment {{ forloop.counter }} by {{ comment.name }}
            {{ comment.created }}
        </p>
        {{ comment.body|linebreaks }}
    </div>
{% empty %}
    <p>There is no comments yet.</p>
{% endfor %}
{% endblock %}
```


mysite > blog > templates > blog > post > post_detail.html > ...

```
    <p class="date">
        Published {{ post.publish}} by {{ post.author }}
    </p>
{{ post.body|linebreaks }}
    <p>
        <a href="{% url "blog:post_share" post.id %}">
         Share this post
        </a>
    </p>

    {% with comments.count as total_comments %}
    <h2>
        {{ total_comments}} comment{{ total_comments|pluralize }}
    </h2>
{% endwith %}
{% for comment in comments %}
    <div class="comment">
        <p class="info">
            Comment {{ forloop.counter }} by {{ comment.name }}
            {{ comment.created }}
        </p>
        {{ comment.body|linebreaks }}
    </div>
{% empty %}
    <p>There is no comments yet.</p>
{% endfor %}
{% if new_comment %}
    <h2>Your comment has been added.</h2>
{% else %}
    <h2>Add a new comment</h2>
    <form method="post">
        {{ comment_form.as_p }}
        {% csrf_token %}
        <p><input type="submit" value="Add comment"></p>
    </form>
{% endif %}
{% endblock %}
```

Adding Comments to the Post Detail Template


Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post

Published Feb. 9, 2025, 6:46 a.m. by dria6

A blog is an online platform or website where individuals or businesses regularly publish written long-form content, known as posts, on various topics. Blogs can be personal, like a digital diary or professional, focusing on areas such as technology, fashion, travel or health. They often include various types of multimedia like images and videos and traditionally encouraged reader engagement through comments - the advance of social media has killed this part of blogs for the most part. Blogs serve as a way to share information, express opinions, and connect with a broader audience.

Share this post

0 comments

There is no comments yet.

Add a new comment

Name:

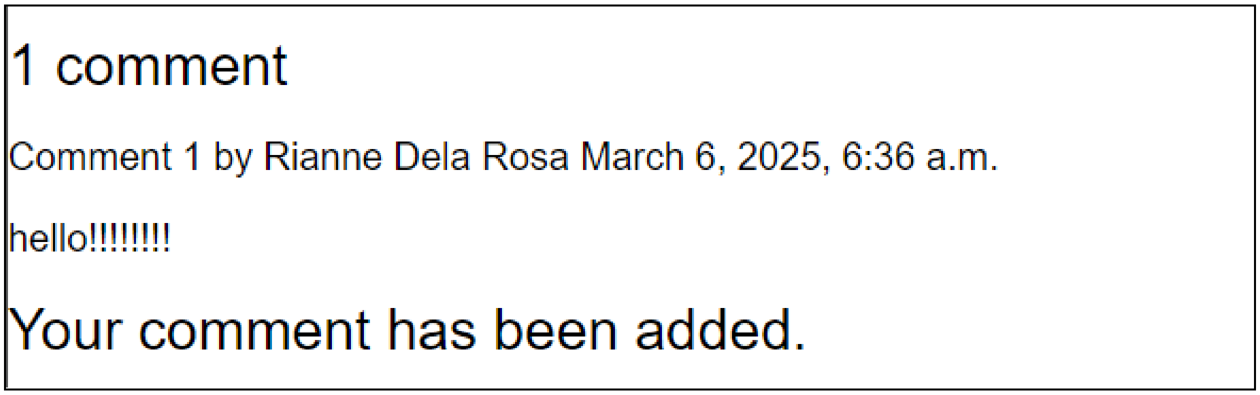
Email:

Body:

My blog

This is my blog.

Post/Comment page



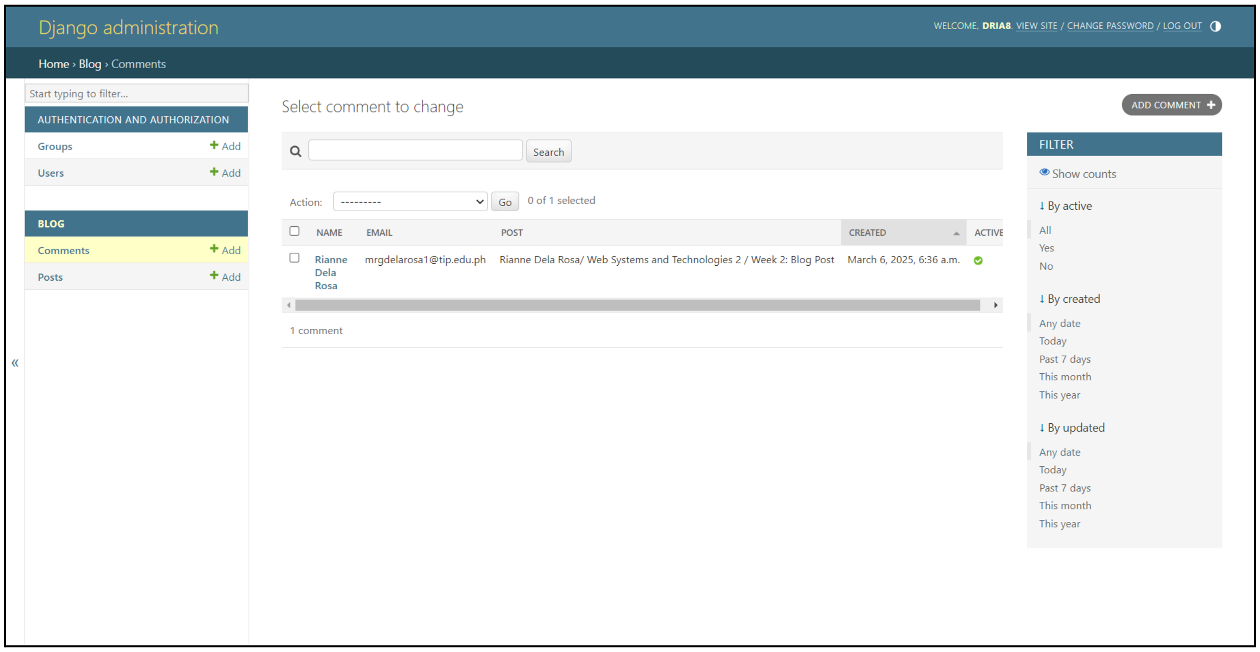
1 comment

Comment 1 by Rianne Dela Rosa March 6, 2025, 6:36 a.m.

hello!!!!!!!!

Your comment has been added.

Added comment



Django administration

WELCOME, **DRIA8**. VIEW SITE / CHANGE PASSWORD / LOG OUT

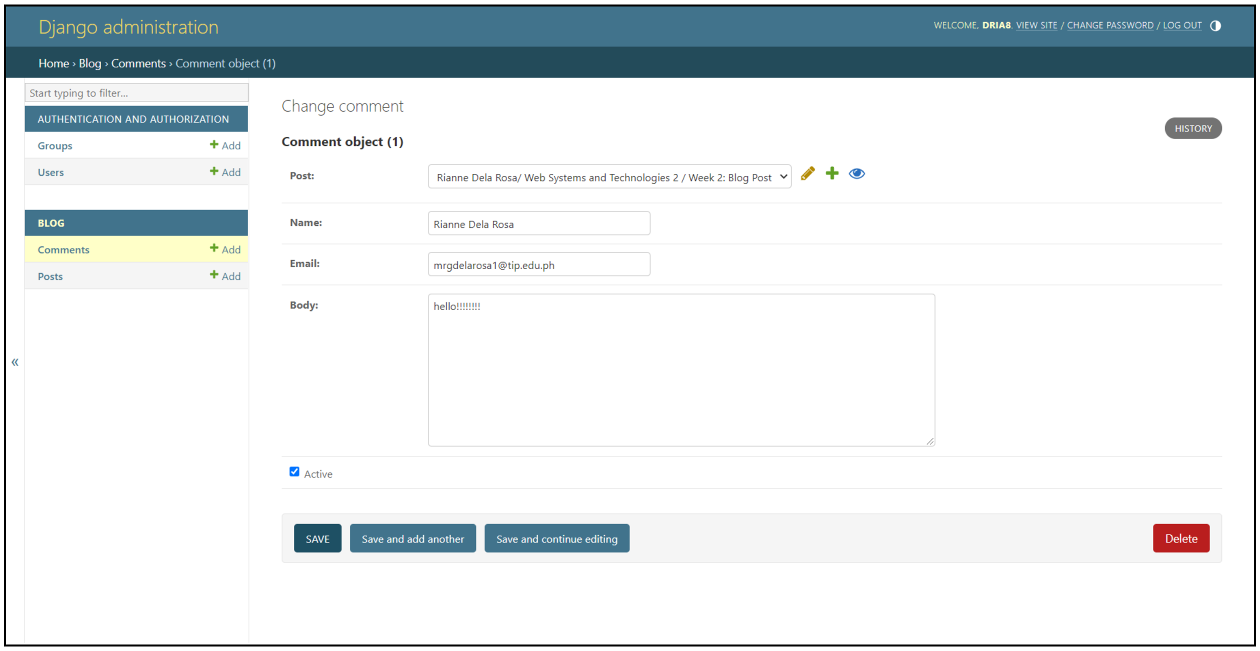
Home › Blog › Comments

Select comment to change

Action: --------- Go 0 of 1 selected

	NAME	EMAIL	POST	CREATED	ACTIVE
☐	Rianne Dela Rosa	mrgdelarosa1@tip.edu.ph	Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post	March 6, 2025, 6:36 a.m.	✓

1 comment


Django administration
WELCOME, DRIA8. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Blog › Comments › Comment object (1)
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups
Users
BLOG
Comments
Posts
Add
HISTORY
SAVE
Save and add another
Save and continue editing
Delete

Change comment

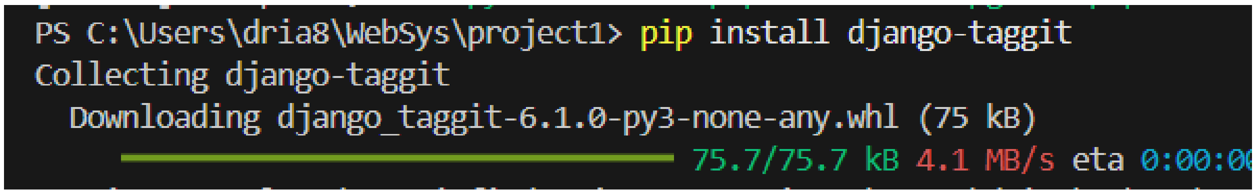
Comment object (1)

Post:	Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post
Name:	Rianne Dela Rosa
Email:	mrgdelarosa1@tip.edu.ph
Body:	hello!!!!!!!!

☑ Active

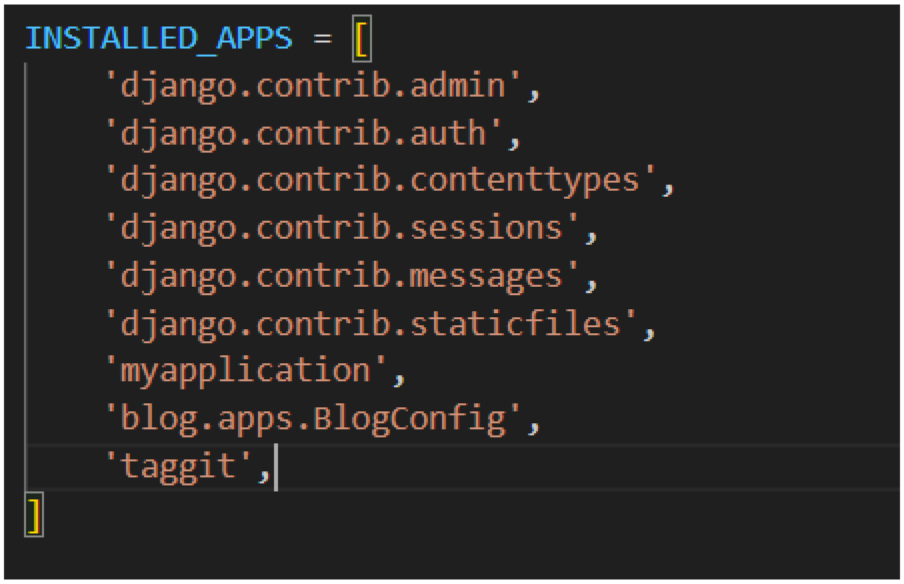
Added comment in admin

Tagging Functionality



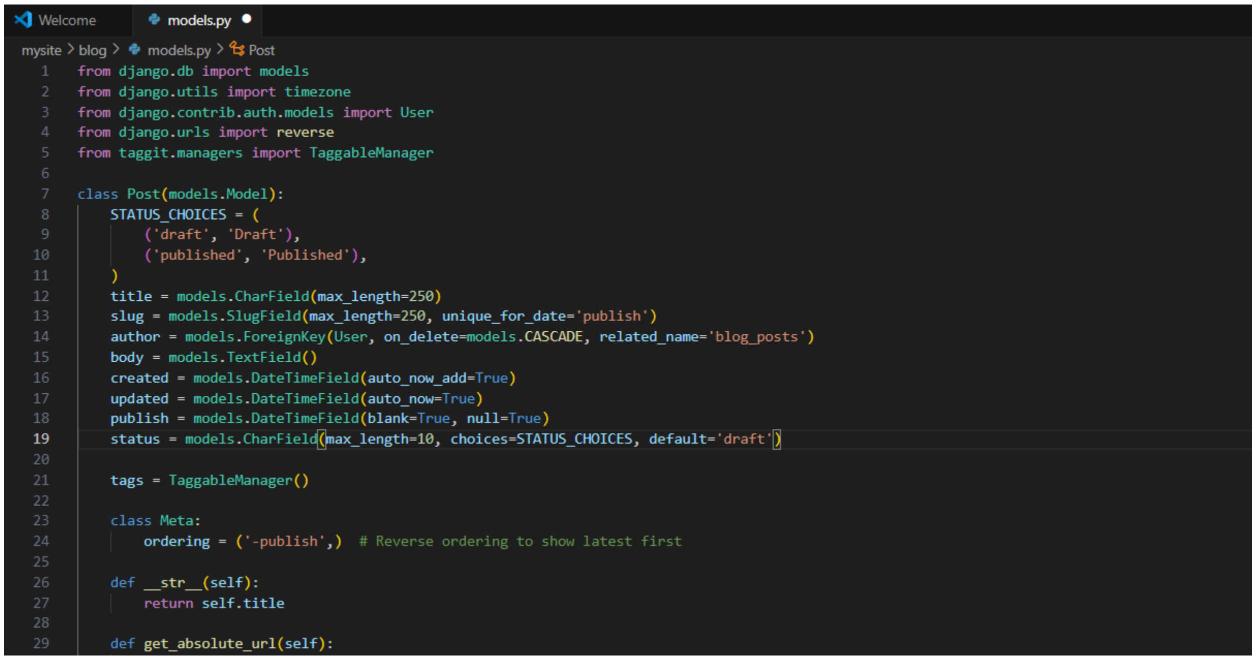
```
PS C:\Users\dria8\WebSys\project1> pip install django-taggit
Collecting django-taggit
  Downloading django_taggit-6.1.0-py3-none-any.whl (75 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 75.7/75.7 kB 4.1 MB/s eta 0:00:0
```

Installing django taggit



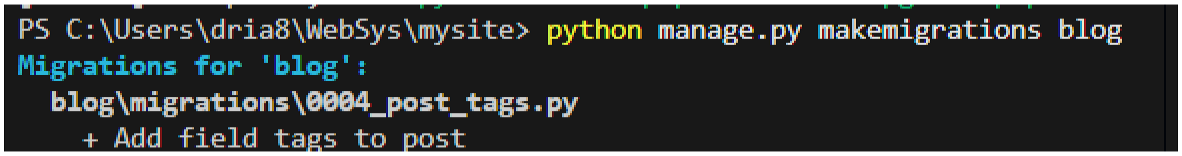
```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'myapplication',
    'blog.apps.BlogConfig',
    'taggit',
]
```

Adding installed django



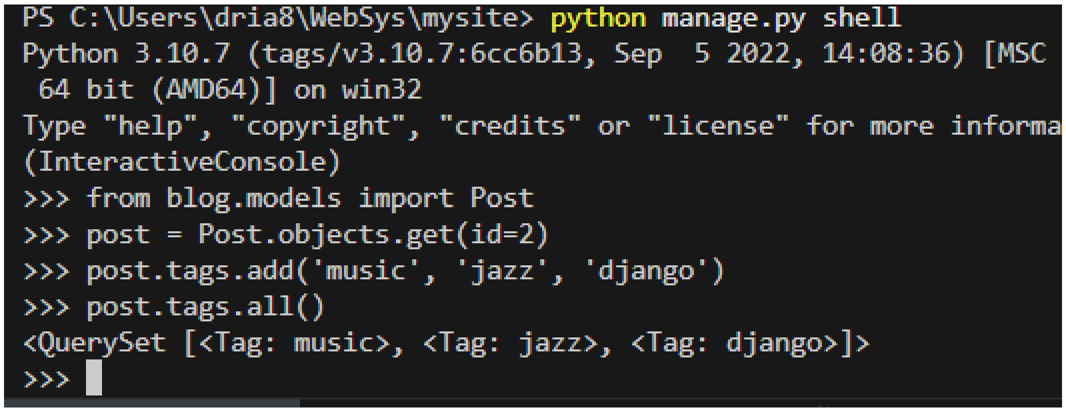
```
Welcome    models.py
mysite > blog > models.py > Post
from django.db import models
from django.utils import timezone
from django.contrib.auth.models import User
from django.urls import reverse
from taggit.managers import TaggableManager

class Post(models.Model):
    STATUS_CHOICES = (
        ('draft', 'Draft'),
        ('published', 'Published'),
    )
    title = models.CharField(max_length=250)
    slug = models.SlugField(max_length=250, unique_for_date='publish')
    author = models.ForeignKey(User, on_delete=models.CASCADE, related_name='blog_posts')
    body = models.TextField()
    created = models.DateTimeField(auto_now_add=True)
    updated = models.DateTimeField(auto_now=True)
    publish = models.DateTimeField(blank=True, null=True)
    status = models.CharField(max_length=10, choices=STATUS_CHOICES, default='draft')

    tags = TaggableManager()

    class Meta:
        ordering = ('-publish',)  # Reverse ordering to show latest first

    def __str__(self):
        return self.title

    def get_absolute_url(self):
```

Editing models.py to add tagging functionality



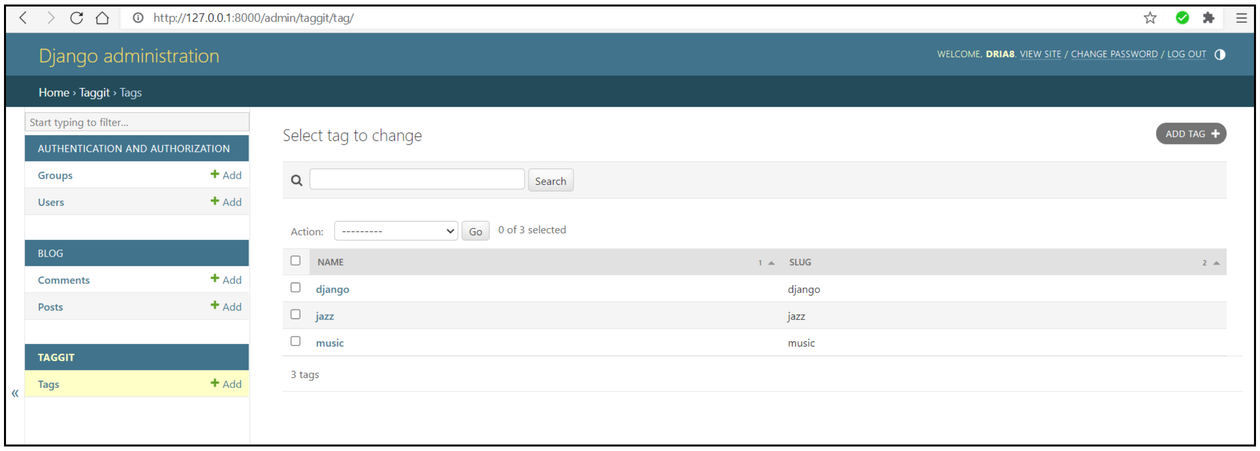
```
PS C:\Users\dria8\WebSys\mysite> python manage.py makemigrations blog
Migrations for 'blog':
  blog\migrations\0004_post_tags.py
    + Add field tags to post
```

Blog migration



```
PS C:\Users\dria8\WebSys\mysite> python manage.py shell
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep  5 2022, 14:08:36) [MSC
 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more informa
(InteractiveConsole)
>>> from blog.models import Post
>>> post = Post.objects.get(id=2)
>>> post.tags.add('music', 'jazz', 'django')
>>> post.tags.all()
<QuerySet [<Tag: music>, <Tag: jazz>, <Tag: django>]>
>>>
```

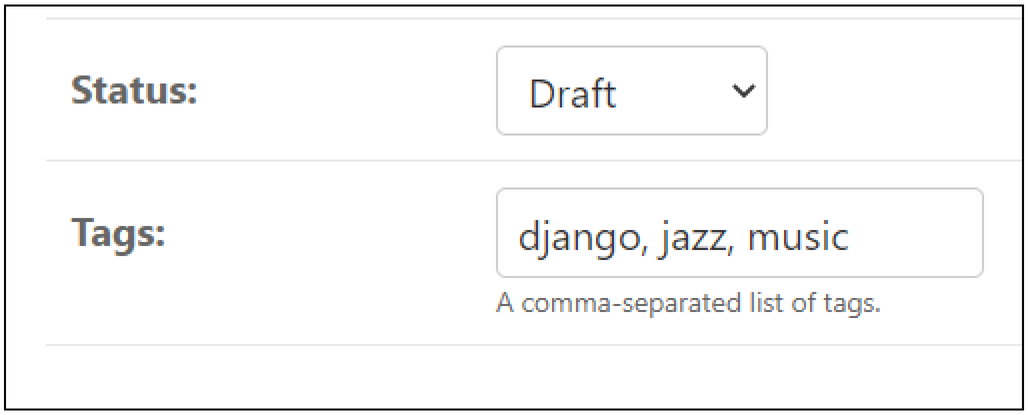
Retrieving a post


http://127.0.0.1:8000/admin/taggit/tag/
Django administration
WELCOME, DRIA8. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Taggit › Tags
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
BLOG
Comments + Add
Posts + Add
TAGGIT
Tags + Add
Select tag to change
ADD TAG +
Search
Action: --------- Go 0 of 3 selected

NAME	SLUG
django	django
jazz	jazz
music	music

3 tags

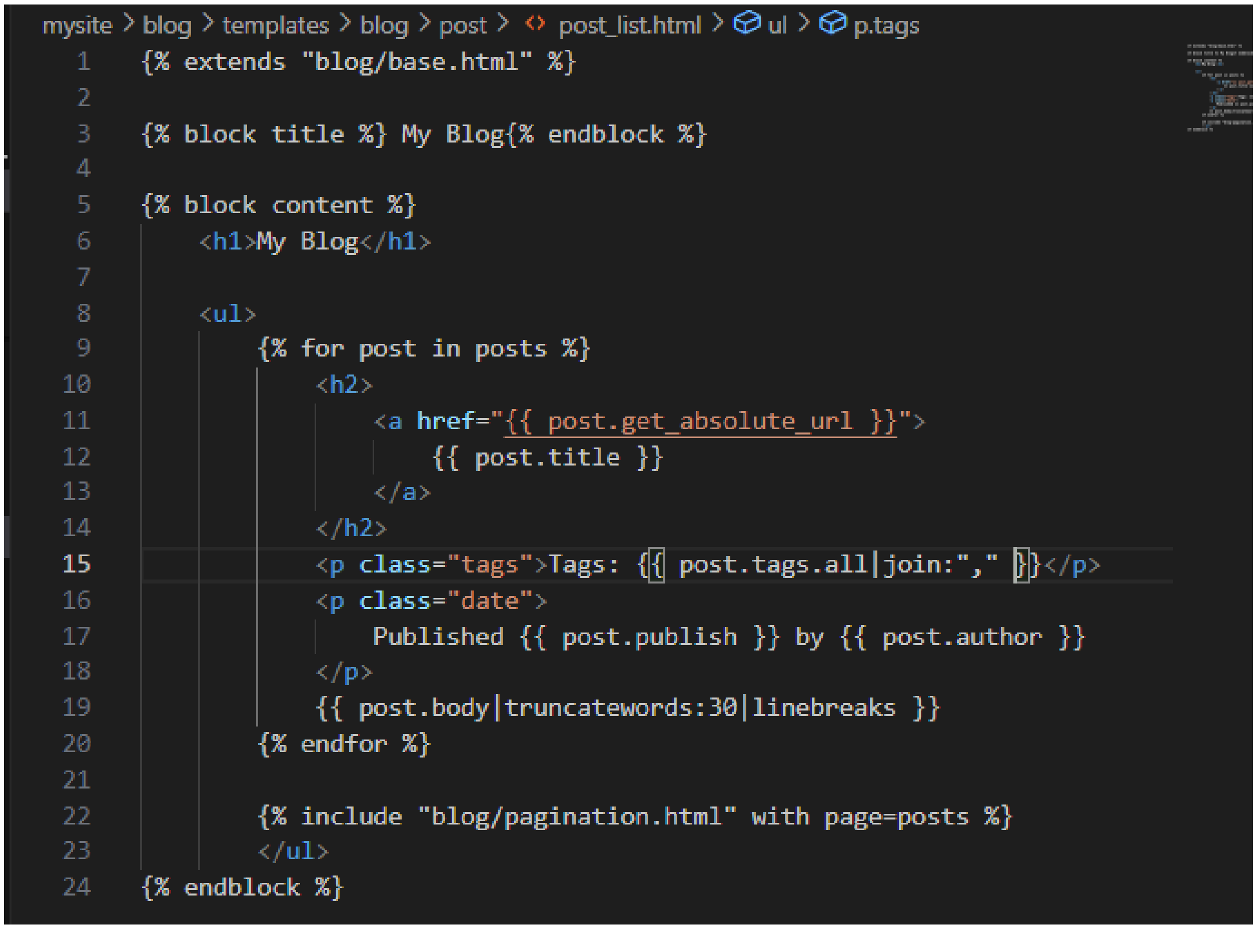
Administration page with the list of Tag objects of the taggit application



Status:	Draft
Tags:	django, jazz, music

A comma-separated list of tags.

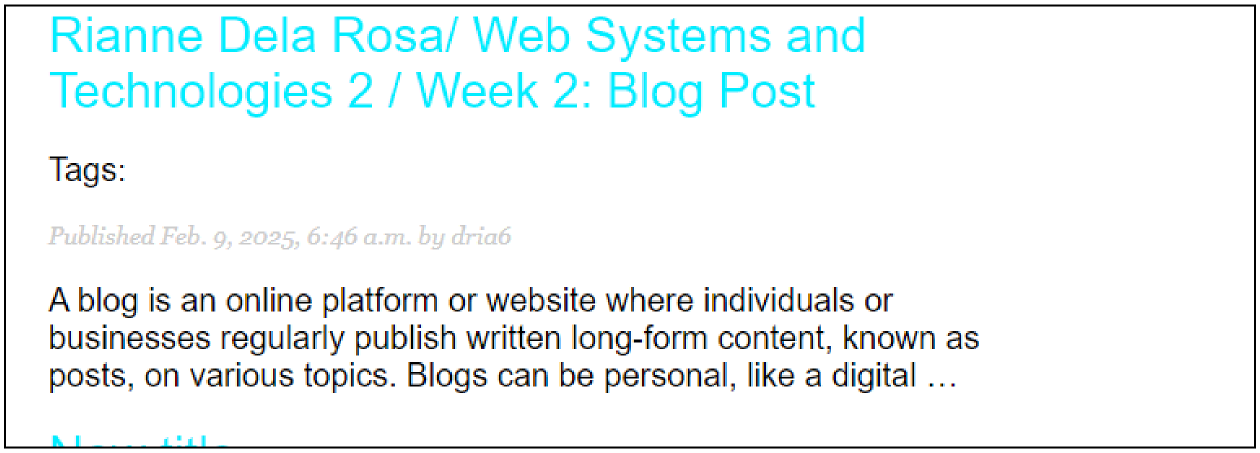
Posts now include a new Tags field, as follows, where you can easily edit tags



mysite > blog > templates > blog > post > post_list.html > ul > p.tags

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

{% block title %} My Blog{% endblock %}

{% block content %}
    <h1>My Blog</h1>

    <ul>
        {% for post in posts %}
            <h2>
                <a href="{{ post.get_absolute_url }}">
                    {{ post.title }}
                </a>
            </h2>
            <p class="tags">Tags: {{ post.tags.all|join:"," }}</p>
            <p class="date">
                Published {{ post.publish }} by {{ post.author }}
            </p>
            {{ post.body|truncatewords:30|linebreaks }}
        {% endfor %}

        {% include "blog/pagination.html" with page=posts %}
        </ul>
{% endblock %}
```

Editing blog posts to display tag

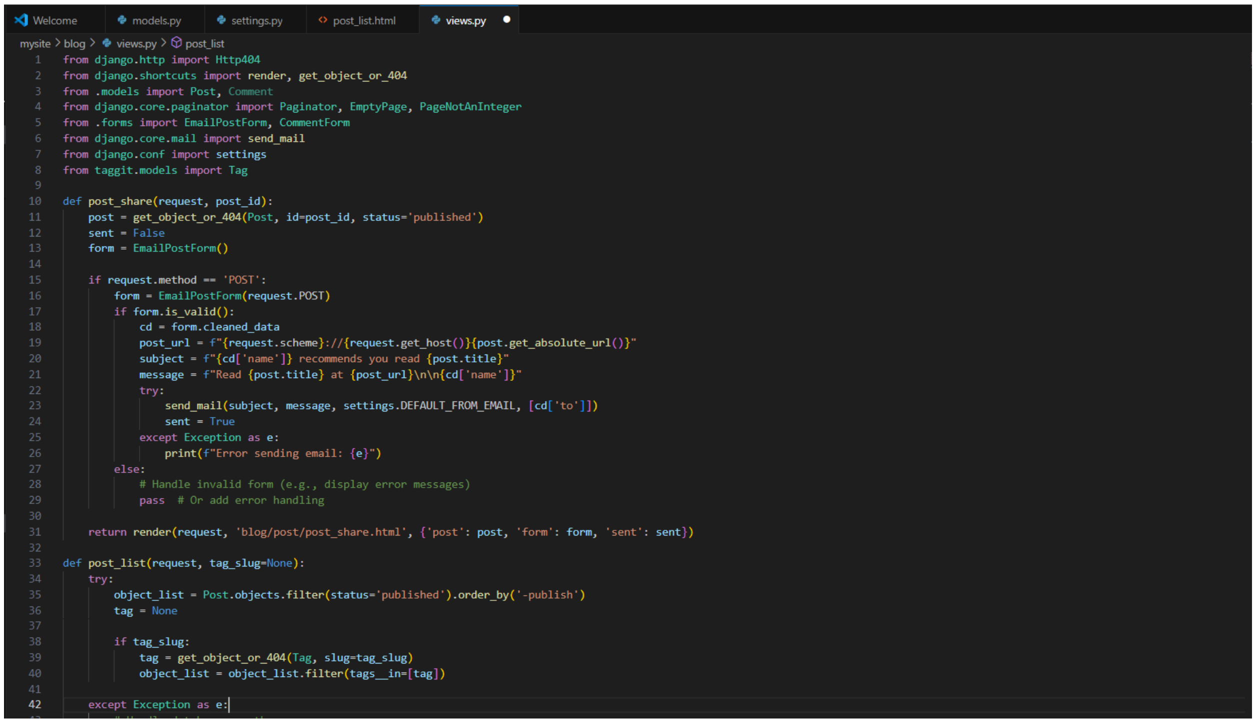


Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post

Tags:

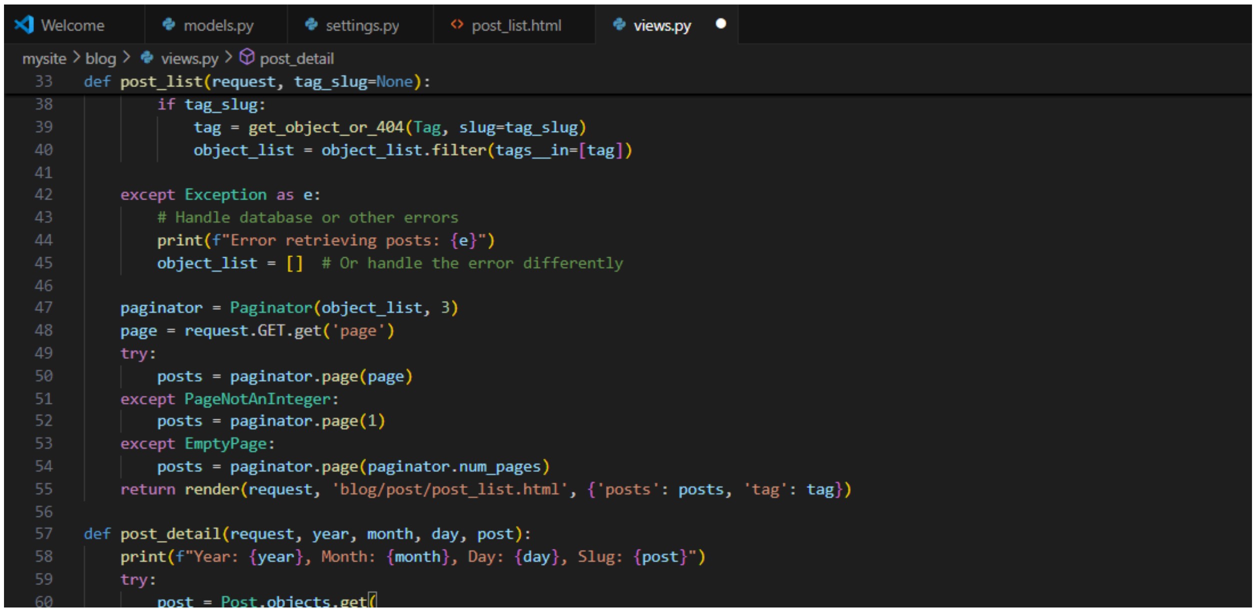
Published Feb. 9, 2025, 6:46 a.m. by dria6

A blog is an online platform or website where individuals or businesses regularly publish written long-form content, known as posts, on various topics. Blogs can be personal, like a digital …



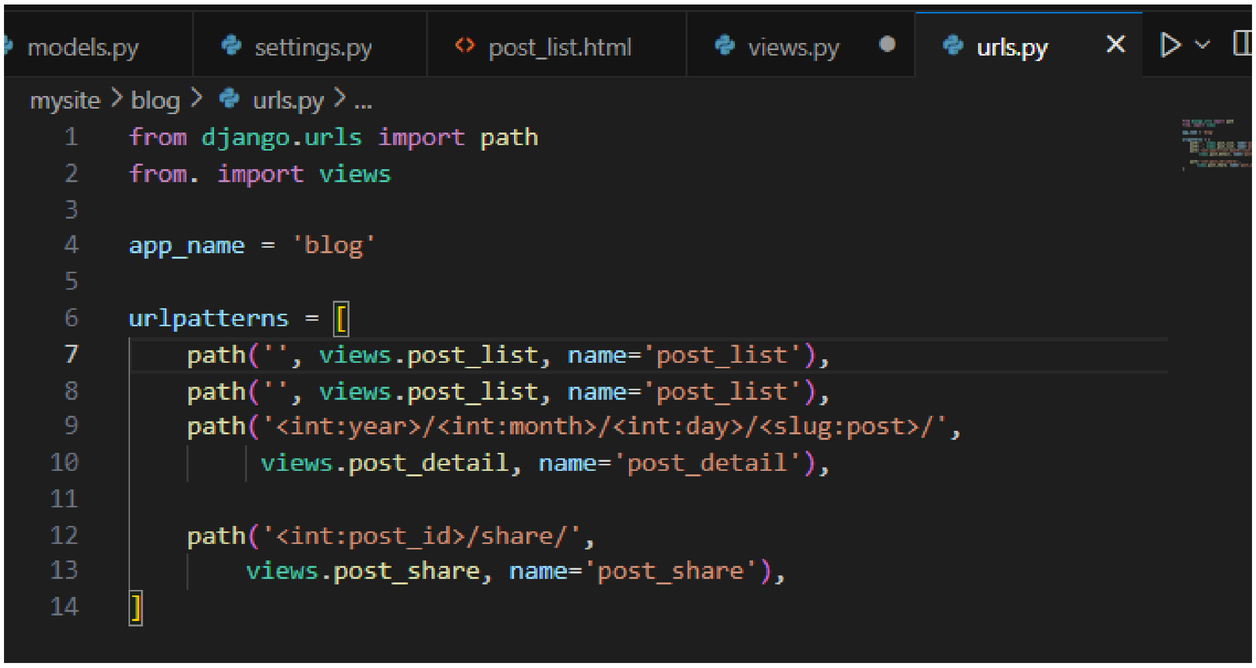
```python
from django.http import Http404
from django.shortcuts import render, get_object_or_404
from .models import Post, Comment
from django.core.paginator import Paginator, EmptyPage, PageNotAnInteger
from .forms import EmailPostForm, CommentForm
from django.core.mail import send_mail
from django.conf import settings
from taggit.models import Tag

def post_share(request, post_id):
    post = get_object_or_404(Post, id=post_id, status='published')
    sent = False
    form = EmailPostForm()

    if request.method == 'POST':
        form = EmailPostForm(request.POST)
        if form.is_valid():
            cd = form.cleaned_data
            post_url = f"{request.scheme}://{request.get_host()}{post.get_absolute_url()}"
            subject = f"{cd['name']} recommends you read {post.title}"
            message = f"Read {post.title} at {post_url}\n\n{cd['name']}"
            try:
                send_mail(subject, message, settings.DEFAULT_FROM_EMAIL, [cd['to']])
                sent = True
            except Exception as e:
                print(f"Error sending email: {e}")
        else:
            # Handle invalid form (e.g., display error messages)
            pass  # Or add error handling

    return render(request, 'blog/post/post_share.html', {'post': post, 'form': form, 'sent': sent})

def post_list(request, tag_slug=None):
    try:
        object_list = Post.objects.filter(status='published').order_by('-publish')
        tag = None

        if tag_slug:
            tag = get_object_or_404(Tag, slug=tag_slug)
            object_list = object_list.filter(tags__in=[tag])

    except Exception as e:
```

Edit the post_list view to let users list all posts tagged with a specific tag

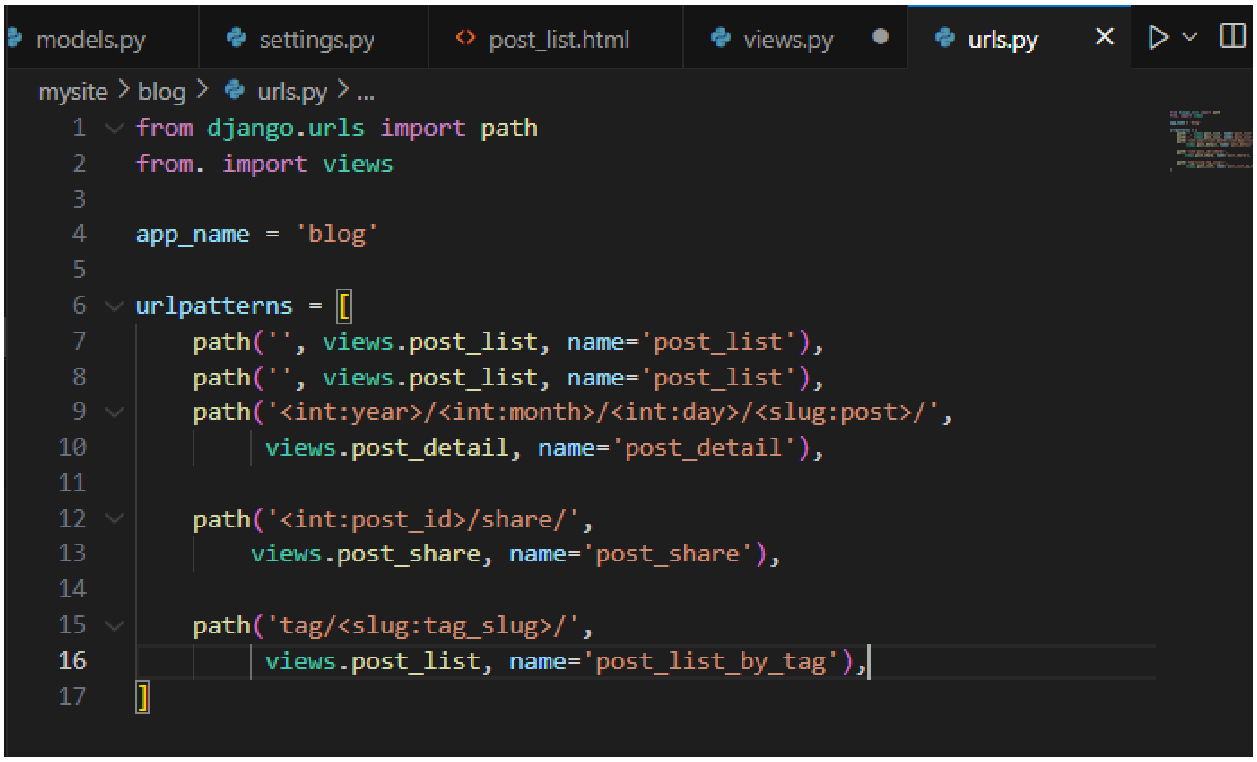


```python
def post_list(request, tag_slug=None):
        if tag_slug:
            tag = get_object_or_404(Tag, slug=tag_slug)
            object_list = object_list.filter(tags__in=[tag])

    except Exception as e:
        # Handle database or other errors
        print(f"Error retrieving posts: {e}")
        object_list = []  # Or handle the error differently

    paginator = Paginator(object_list, 3)
    page = request.GET.get('page')
    try:
        posts = paginator.page(page)
    except PageNotAnInteger:
        posts = paginator.page(1)
    except EmptyPage:
        posts = paginator.page(paginator.num_pages)
    return render(request, 'blog/post/post_list.html', {'posts': posts, 'tag': tag})

def post_detail(request, year, month, day, post):
    print(f"Year: {year}, Month: {month}, Day: {day}, Slug: {post}")
    try:
        post = Post.objects.get(
```

Modifying the render

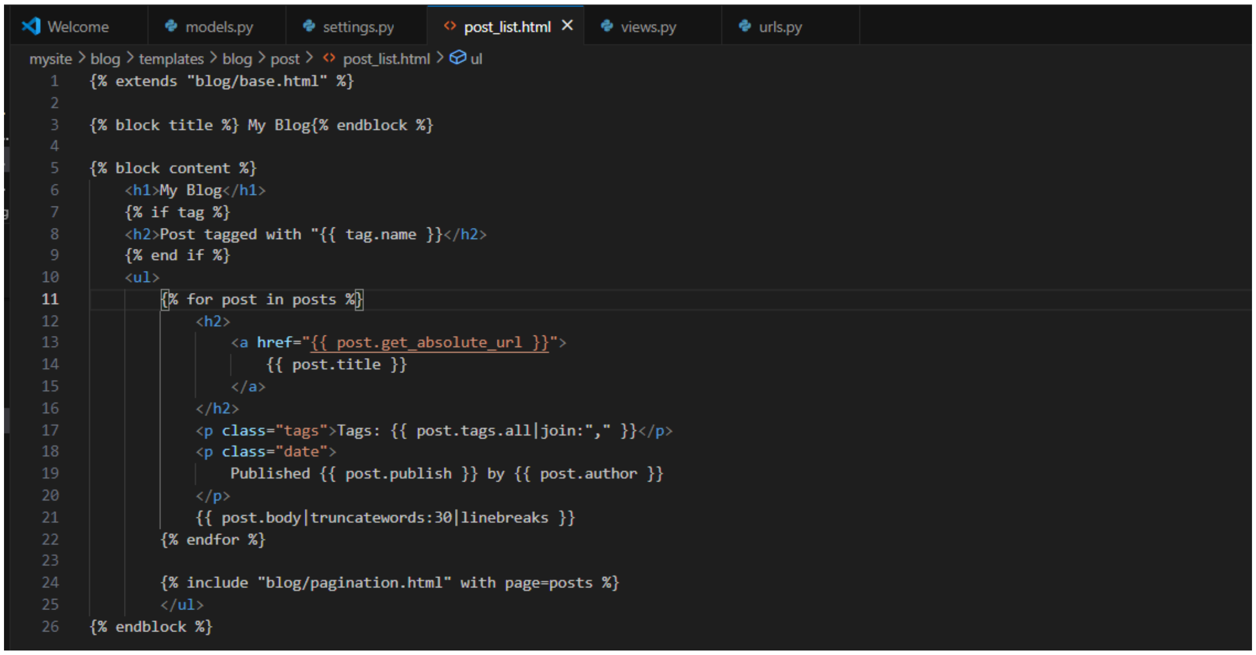


```python
from django.urls import path
from. import views

app_name = 'blog'

urlpatterns = [
    path('', views.post_list, name='post_list'),
    path('', views.post_list, name='post_list'),
    path('<int:year>/<int:month>/<int:day>/<slug:post>/',
         views.post_detail, name='post_detail'),

    path('<int:post_id>/share/',
        views.post_share, name='post_share'),
]
```

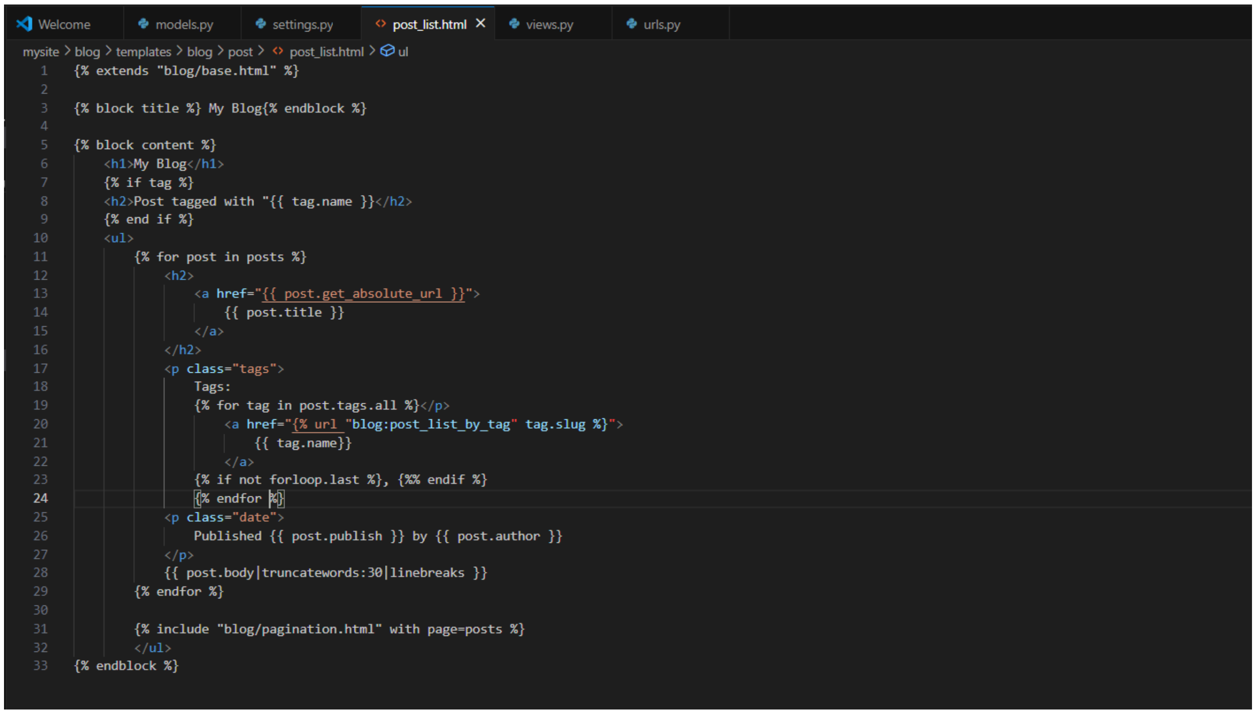


```python
from django.urls import path
from. import views

app_name = 'blog'

urlpatterns = [
    path('', views.post_list, name='post_list'),
    path('', views.post_list, name='post_list'),
    path('<int:year>/<int:month>/<int:day>/<slug:post>/',
         views.post_detail, name='post_detail'),

    path('<int:post_id>/share/',
        views.post_share, name='post_share'),

    path('tag/<slug:tag_slug>/',
         views.post_list, name='post_list_by_tag'),
]
```

Adding post_list tag

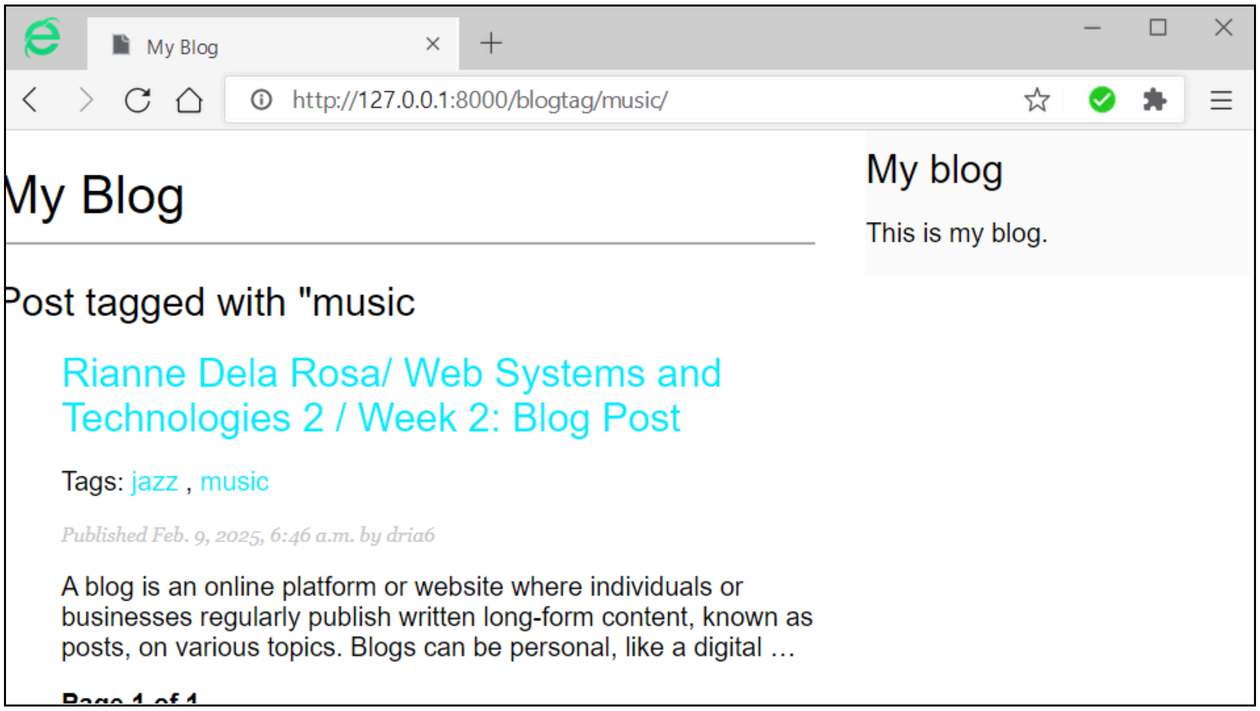

Welcome
models.py
settings.py
post_list.html
views.py
urls.py
mysite > blog > templates > blog > post > post_list.html > ul

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

{% block title %} My Blog{% endblock %}

{% block content %}
    <h1>My Blog</h1>
    {% if tag %}
    <h2>Post tagged with "{{ tag.name }}</h2>
    {% end if %}
    <ul>
        {% for post in posts %}
            <h2>
                <a href="{{ post.get_absolute_url }}">
                    {{ post.title }}
                </a>
            </h2>
            <p class="tags">Tags: {{ post.tags.all|join:"," }}</p>
            <p class="date">
                Published {{ post.publish }} by {{ post.author }}
            </p>
            {{ post.body|truncatewords:30|linebreaks }}
        {% endfor %}

        {% include "blog/pagination.html" with page=posts %}
        </ul>
{% endblock %}
```


Welcome
models.py
settings.py
post_list.html
views.py
urls.py
mysite > blog > templates > blog > post > post_list.html > ul

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

{% block title %} My Blog{% endblock %}

{% block content %}
    <h1>My Blog</h1>
    {% if tag %}
    <h2>Post tagged with "{{ tag.name }}</h2>
    {% end if %}
    <ul>
        {% for post in posts %}
            <h2>
                <a href="{{ post.get_absolute_url }}">
                    {{ post.title }}
                </a>
            </h2>
            <p class="tags">
                Tags:
                {% for tag in post.tags.all %}</p>
                    <a href="{% url "blog:post_list_by_tag" tag.slug %}">
                        {{ tag.name}}
                    </a>
                {% if not forloop.last %}, {%% endif %}
                {% endfor %}
            <p class="date">
                Published {{ post.publish }} by {{ post.author }}
            </p>
            {{ post.body|truncatewords:30|linebreaks }}
        {% endfor %}

        {% include "blog/pagination.html" with page=posts %}
        </ul>
{% endblock %}
```

Modifying post_list



My Blog

My blog

This is my blog.

Post tagged with "music

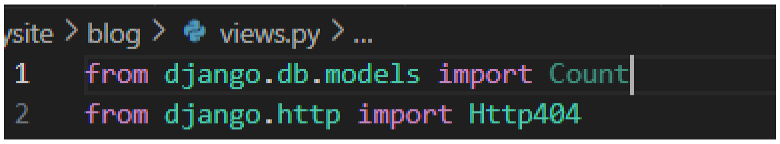
Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post

Tags: jazz , music

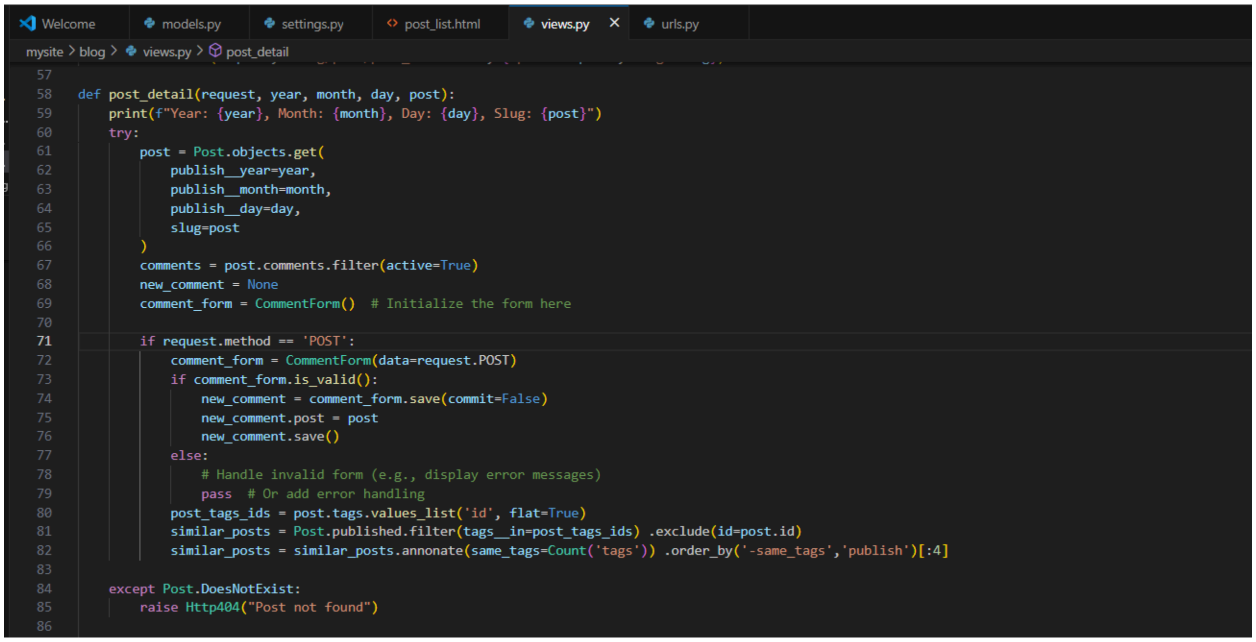
Published Feb. 9, 2025, 6:46 a.m. by dria6

A blog is an online platform or website where individuals or businesses regularly publish written long-form content, known as posts, on various topics. Blogs can be personal, like a digital …

List posts filtered by tag

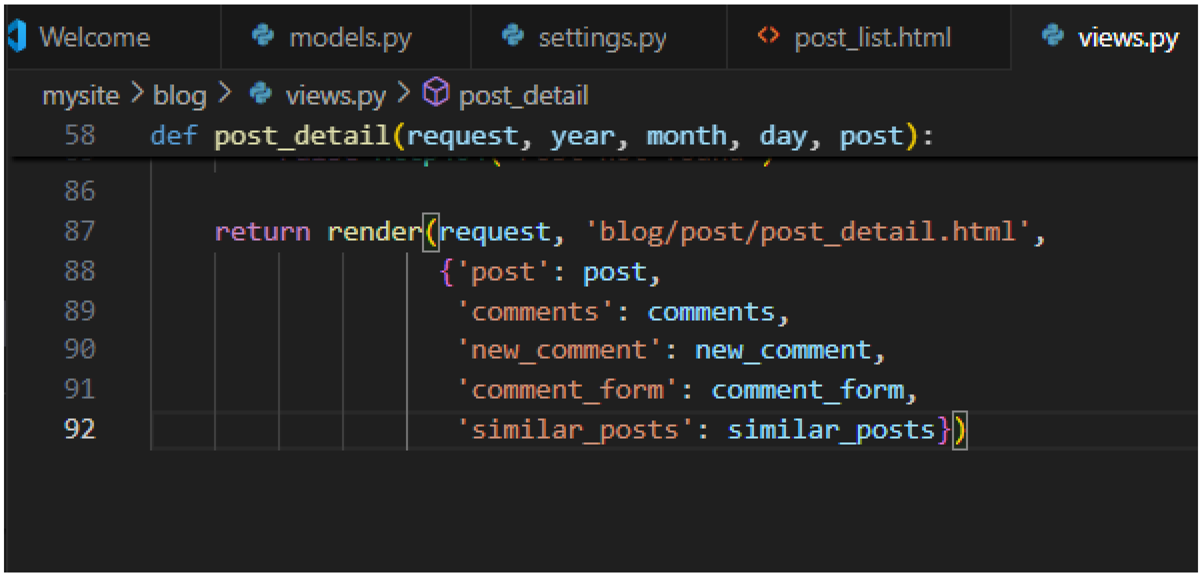


```python
from django.db.models import Count
from django.http import Http404
```



```python

def post_detail(request, year, month, day, post):
    print(f"Year: {year}, Month: {month}, Day: {day}, Slug: {post}")
    try:
        post = Post.objects.get(
            publish__year=year,
            publish__month=month,
            publish__day=day,
            slug=post
        )
        comments = post.comments.filter(active=True)
        new_comment = None
        comment_form = CommentForm()  # Initialize the form here

        if request.method == 'POST':
            comment_form = CommentForm(data=request.POST)
            if comment_form.is_valid():
                new_comment = comment_form.save(commit=False)
                new_comment.post = post
                new_comment.save()
            else:
                # Handle invalid form (e.g., display error messages)
                pass  # Or add error handling
            post_tags_ids = post.tags.values_list('id', flat=True)
            similar_posts = Post.published.filter(tags__in=post_tags_ids) .exclude(id=post.id)
            similar_posts = similar_posts.annonate(same_tags=Count('tags')) .order_by('-same_tags','publish')[:4]

    except Post.DoesNotExist:
        raise Http404("Post not found")

```

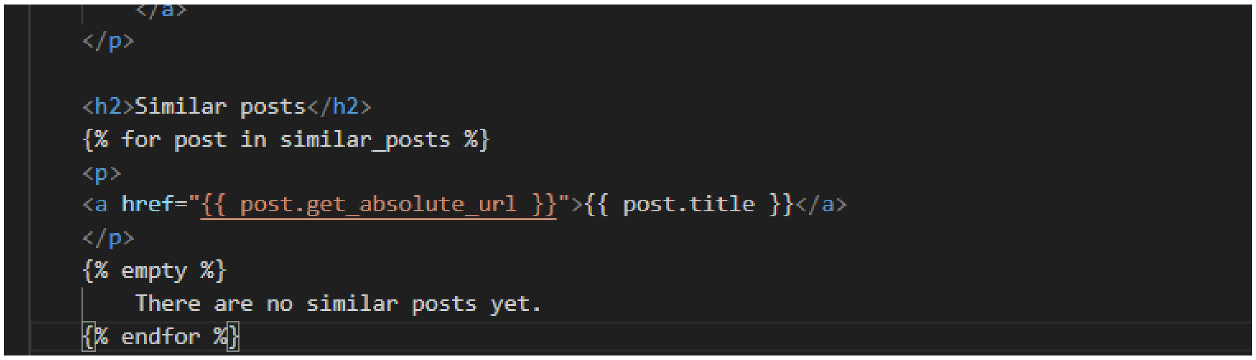
Adding function to perform aggregated counts of tags


Welcome | models.py | settings.py | post_list.html | views.py

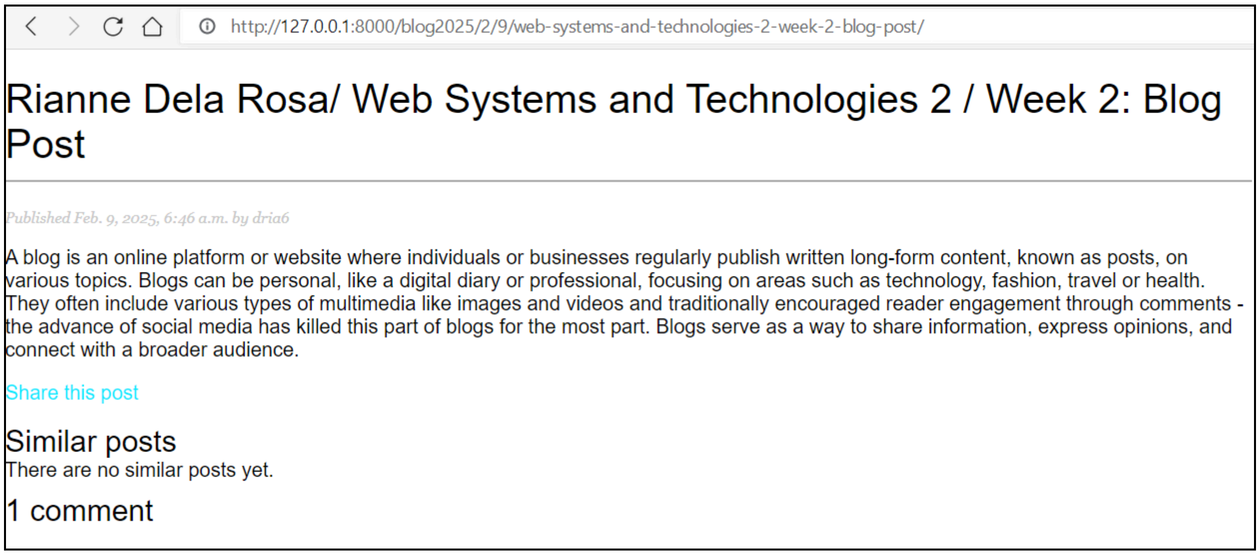
mysite > blog > views.py > post_detail

```
def post_detail(request, year, month, day, post):

    return render(request, 'blog/post/post_detail.html',
                  {'post': post,
                   'comments': comments,
                   'new_comment': new_comment,
                   'comment_form': comment_form,
                   'similar_posts': similar_posts})
```

Adding similar_posts


```
    </a>
</p>

<h2>Similar posts</h2>
{% for post in similar_posts %}
<p>
<a href="{{ post.get_absolute_url }}">{{ post.title }}</a>
</p>
{% empty %}
    There are no similar posts yet.
{% endfor %}
```

Editing post_detail


http://127.0.0.1:8000/blog2025/2/9/web-systems-and-technologies-2-week-2-blog-post/

Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post

Published Feb. 9, 2025, 6:46 a.m. by dria6

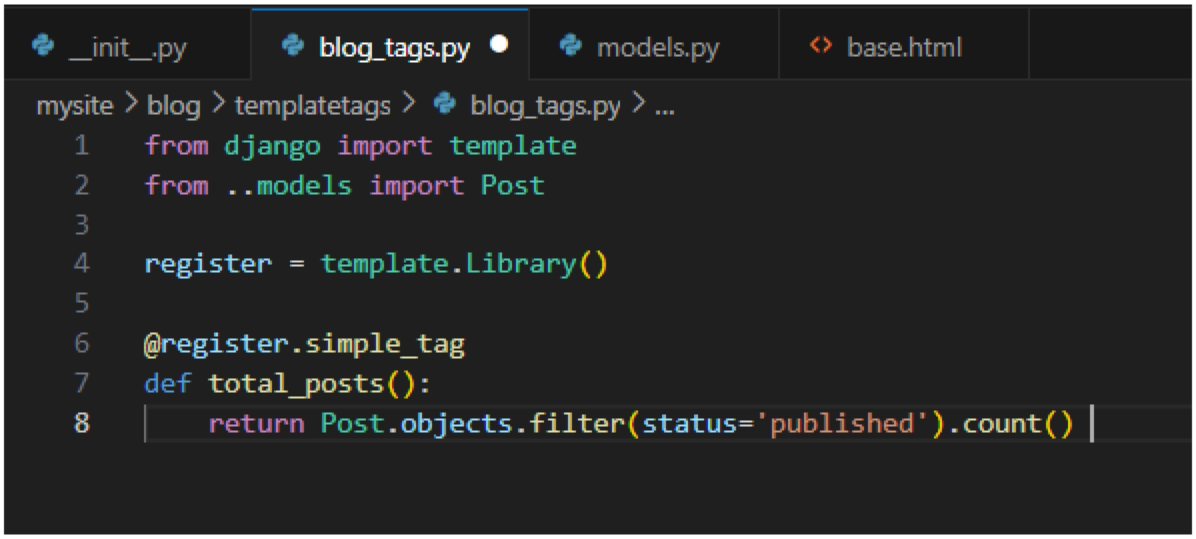
A blog is an online platform or website where individuals or businesses regularly publish written long-form content, known as posts, on various topics. Blogs can be personal, like a digital diary or professional, focusing on areas such as technology, fashion, travel or health. They often include various types of multimedia like images and videos and traditionally encouraged reader engagement through comments - the advance of social media has killed this part of blogs for the most part. Blogs serve as a way to share information, express opinions, and connect with a broader audience.

Share this post

Similar posts
There are no similar posts yet.

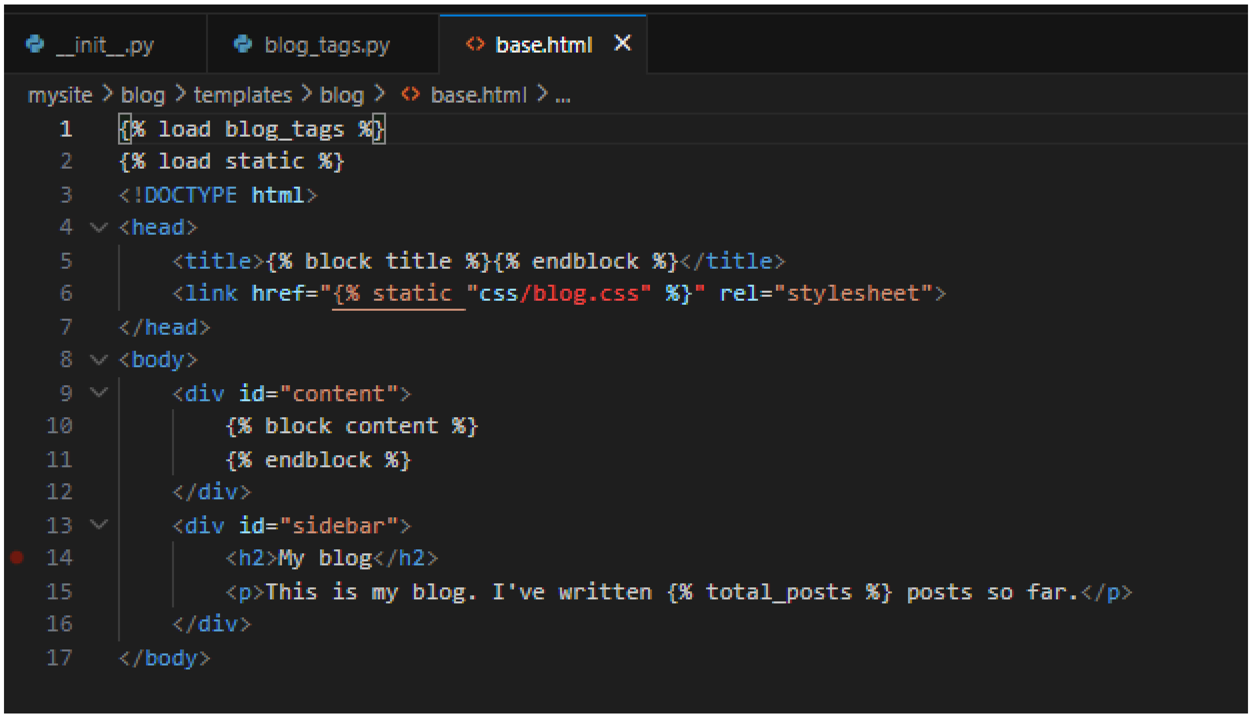
1 comment

Similar post page



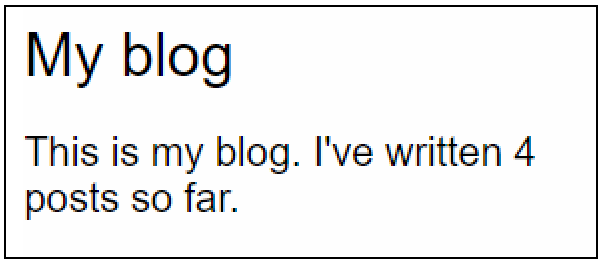
```python
from django import template
from ..models import Post

register = template.Library()

@register.simple_tag
def total_posts():
    return Post.objects.filter(status='published').count()
```

Custom Template Tags



```html
{% load blog_tags %}
{% load static %}
<!DOCTYPE html>
<head>
    <title>{% block title %}{% endblock %}</title>
    <link href="{% static "css/blog.css" %}" rel="stylesheet">
</head>
<body>
    <div id="content">
        {% block content %}
        {% endblock %}
    </div>
    <div id="sidebar">
        <h2>My blog</h2>
        <p>This is my blog. I've written {% total_posts %} posts so far.</p>
    </div>
</body>
```

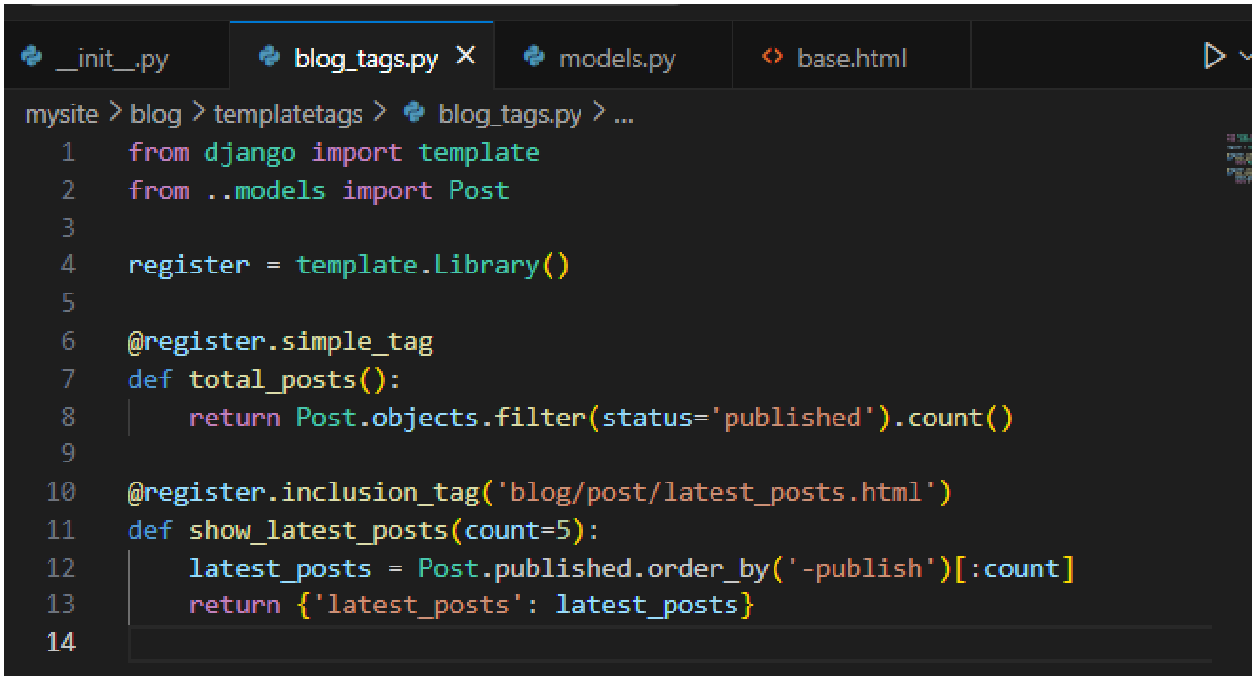
Adding {% load blog_tags %}



My blog

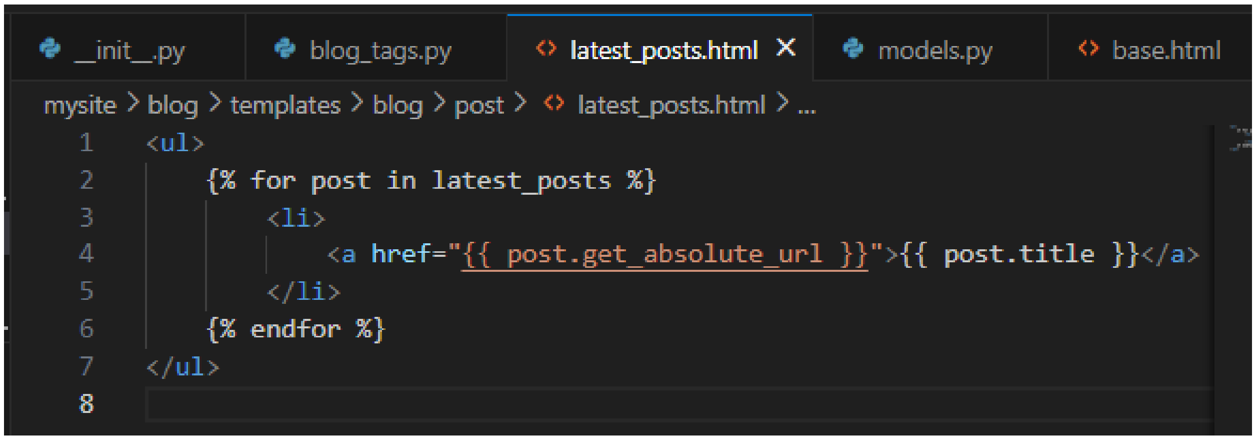
This is my blog. I've written 4 posts so far.

Total number of posts in the sidebar of the site



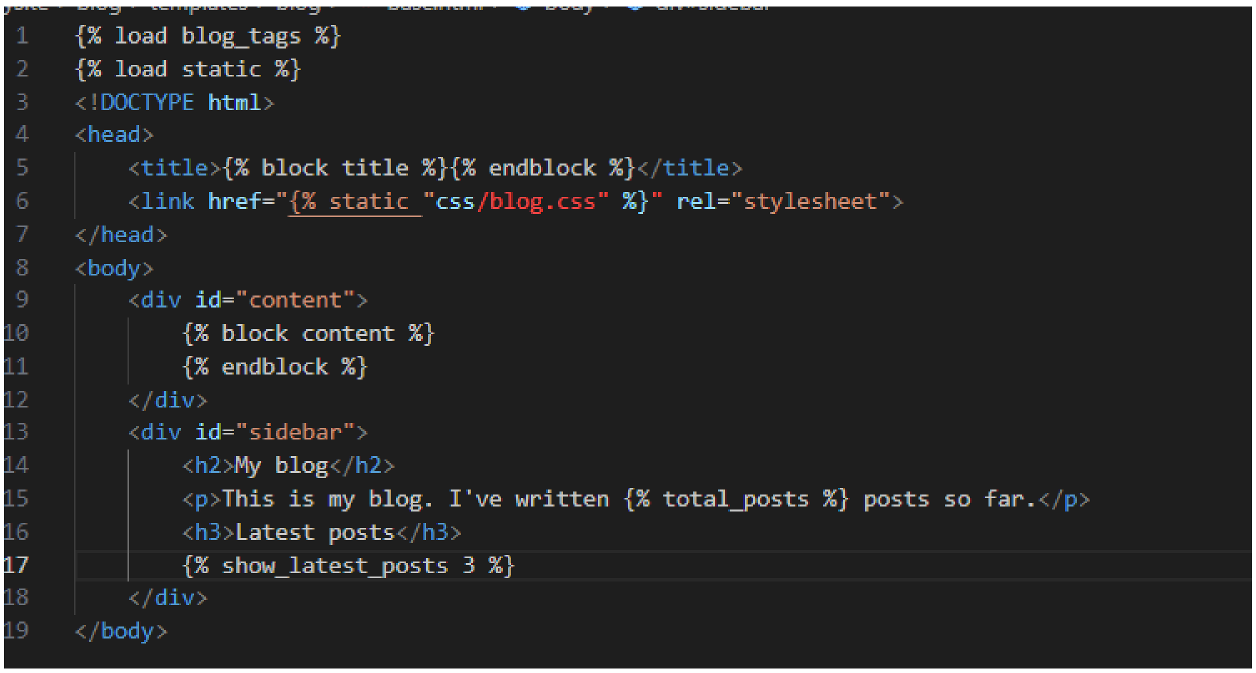
```python
from django import template
from ..models import Post

register = template.Library()

@register.simple_tag
def total_posts():
    return Post.objects.filter(status='published').count()

@register.inclusion_tag('blog/post/latest_posts.html')
def show_latest_posts(count=5):
    latest_posts = Post.published.order_by('-publish')[:count]
    return {'latest_posts': latest_posts}
```

Ediingt the blog_tags.py file


__init__.py
blog_tags.py
latest_posts.html
models.py
base.html
mysite > blog > templates > blog > post > latest_posts.html > ...

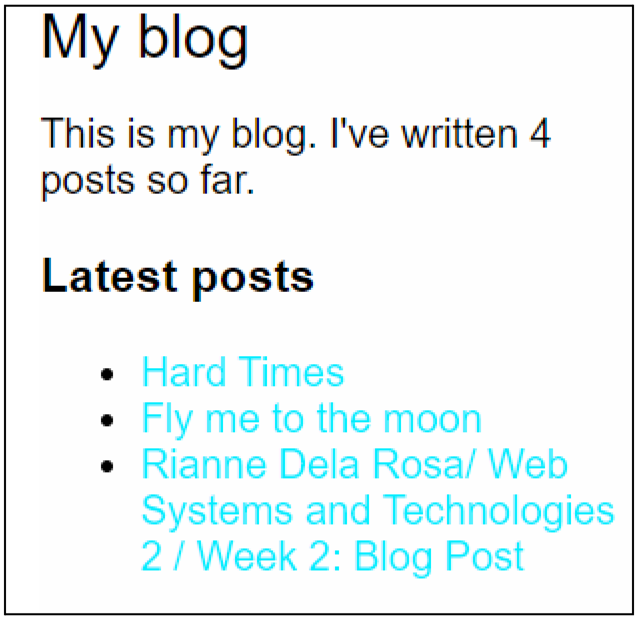
```html
<ul>
    {% for post in latest_posts %}
        <li>
            <a href="{{ post.get_absolute_url }}">{{ post.title }}</a>
        </li>
    {% endfor %}
</ul>
```

Creating a new template file under blog/post/ and name it latest_posts.html.



```
{% load blog_tags %}
{% load static %}
<!DOCTYPE html>
<head>
    <title>{% block title %}{% endblock %}</title>
    <link href="{% static "css/blog.css" %}" rel="stylesheet">
</head>
<body>
    <div id="content">
        {% block content %}
        {% endblock %}
    </div>
    <div id="sidebar">
        <h2>My blog</h2>
        <p>This is my blog. I've written {% total_posts %} posts so far.</p>
        <h3>Latest posts</h3>
        {% show_latest_posts 3 %}
    </div>
</body>
```

Editing the blog/base.html template



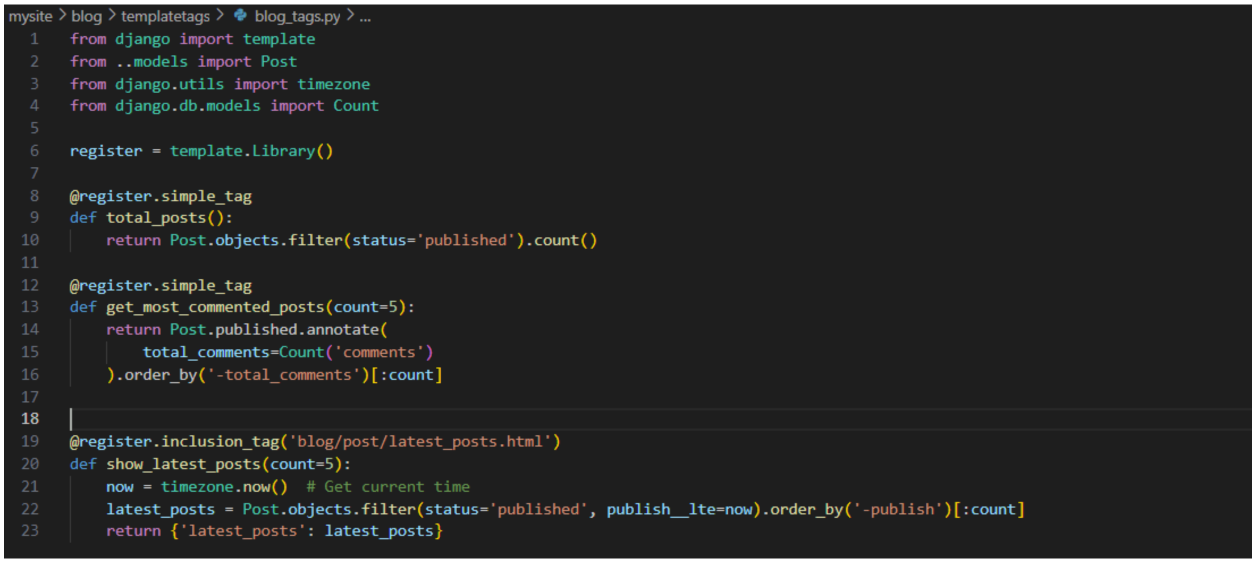
My blog

This is my blog. I've written 4 posts so far.

Latest posts

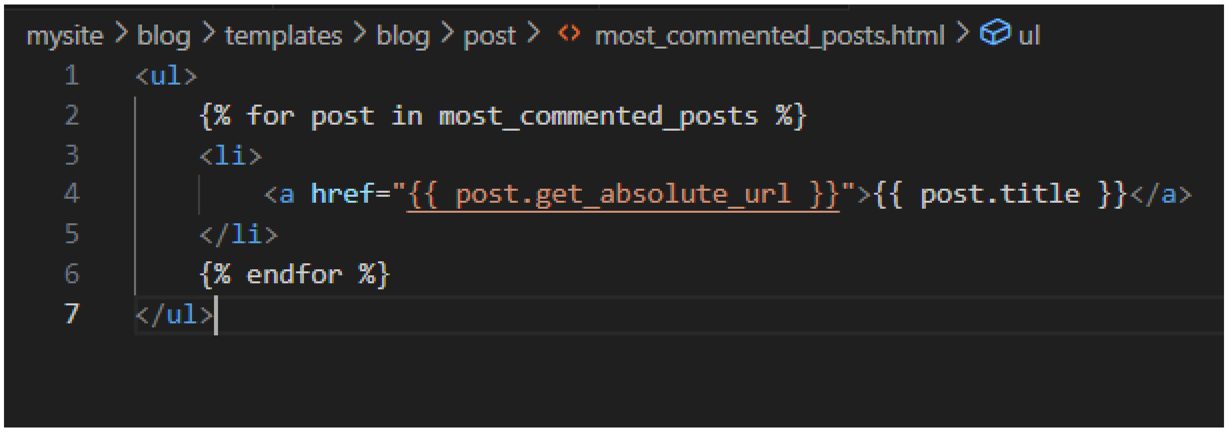
- Hard Times
- Fly me to the moon
- Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post

Sidebar latest post


mysite > blog > templatetags > blog_tags.py > ...

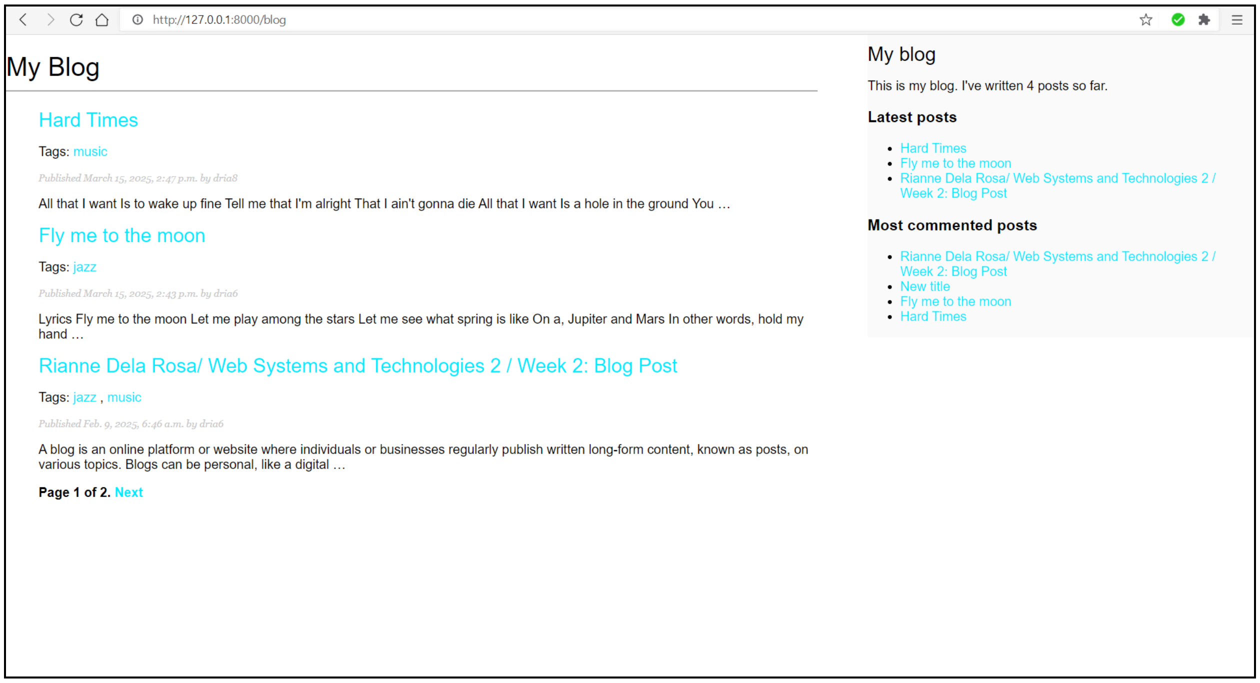
```python
from django import template
from ..models import Post
from django.utils import timezone
from django.db.models import Count

register = template.Library()

@register.simple_tag
def total_posts():
    return Post.objects.filter(status='published').count()

@register.simple_tag
def get_most_commented_posts(count=5):
    return Post.published.annotate(
        total_comments=Count('comments')
    ).order_by('-total_comments')[:count]

@register.inclusion_tag('blog/post/latest_posts.html')
def show_latest_posts(count=5):
    now = timezone.now()  # Get current time
    latest_posts = Post.objects.filter(status='published', publish__lte=now).order_by('-publish')[:count]
    return {'latest_posts': latest_posts}
```

Editing the blog_tags.py file


mysite > blog > templates > blog > post > most_commented_posts.html > ul

```html
<ul>
    {% for post in most_commented_posts %}
    <li>
        <a href="{{ post.get_absolute_url }}">{{ post.title }}</a>
    </li>
    {% endfor %}
</ul>
```

most_commented_posts.html



My Blog

Hard Times

Tags: music

Published March 15, 2025, 2:47 p.m. by dria8

All that I want Is to wake up fine Tell me that I'm alright That I ain't gonna die All that I want Is a hole in the ground You …

Fly me to the moon

Tags: jazz

Published March 15, 2025, 2:43 p.m. by dria6

Lyrics Fly me to the moon Let me play among the stars Let me see what spring is like On a, Jupiter and Mars In other words, hold my hand …

Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post

Tags: jazz , music

Published Feb. 9, 2025, 6:46 a.m. by dria6

A blog is an online platform or website where individuals or businesses regularly publish written long-form content, known as posts, on various topics. Blogs can be personal, like a digital …

Page 1 of 2. Next

My blog

This is my blog. I've written 4 posts so far.

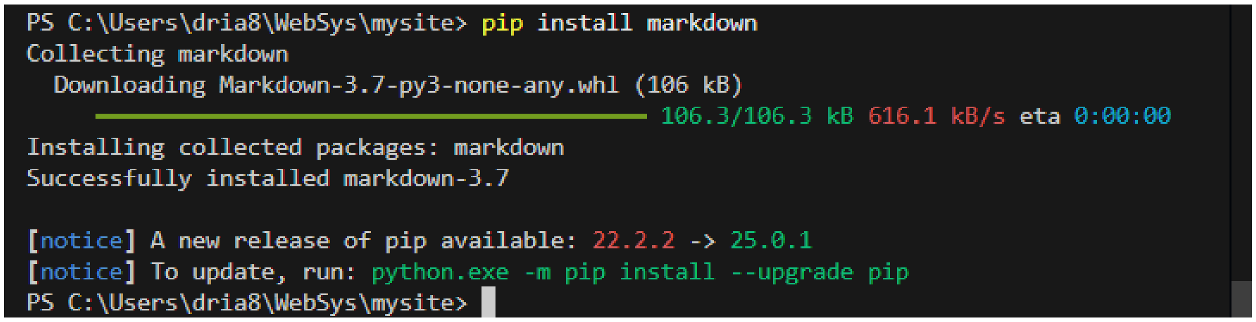
Latest posts

- Hard Times
- Fly me to the moon
- Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post

Most commented posts

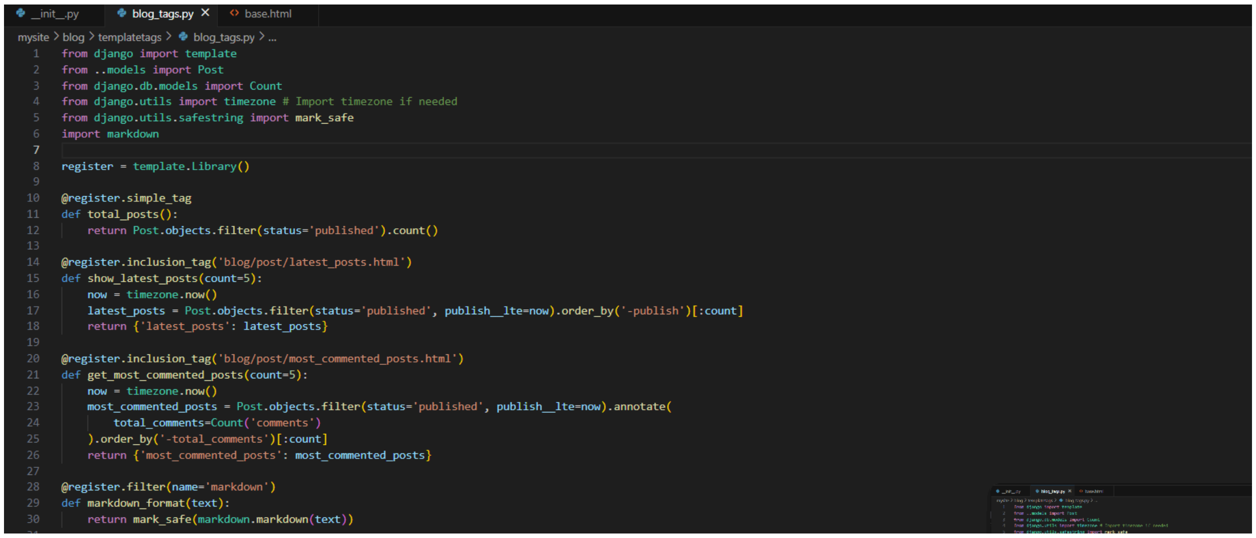
- Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog Post
- New title
- Fly me to the moon
- Hard Times

Final result of custom template tags and filters



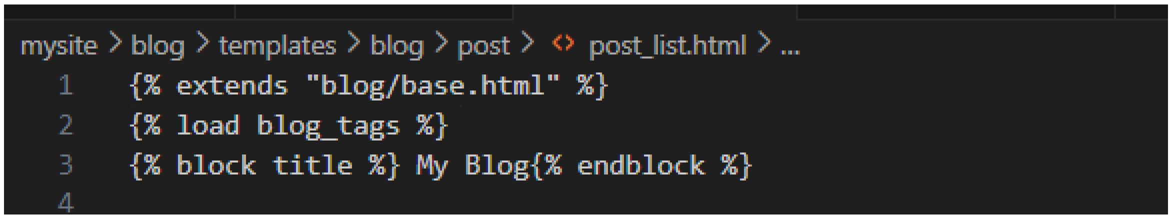
```
PS C:\Users\dria8\WebSys\mysite> pip install markdown
Collecting markdown
  Downloading Markdown-3.7-py3-none-any.whl (106 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 106.3/106.3 kB 616.1 kB/s eta 0:00:00
Installing collected packages: markdown
Successfully installed markdown-3.7

[notice] A new release of pip available: 22.2.2 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\dria8\WebSys\mysite>
```

Installing markdown



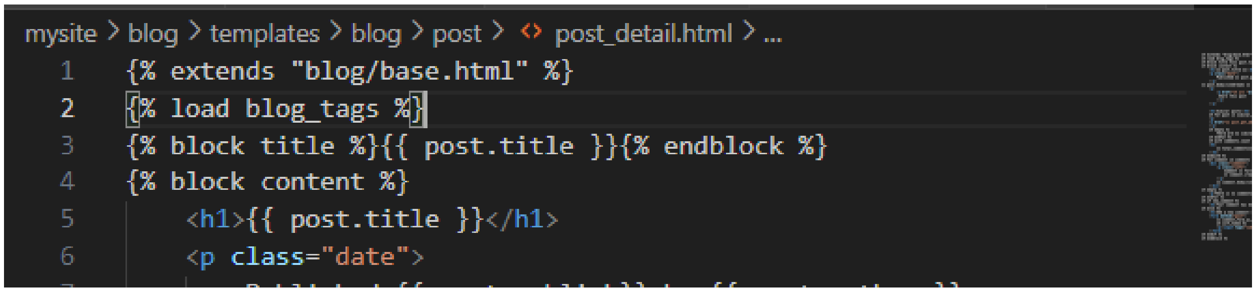
```python
from django import template
from ..models import Post
from django.db.models import Count
from django.utils import timezone # Import timezone if needed
from django.utils.safestring import mark_safe
import markdown

register = template.Library()

@register.simple_tag
def total_posts():
    return Post.objects.filter(status='published').count()

@register.inclusion_tag('blog/post/latest_posts.html')
def show_latest_posts(count=5):
    now = timezone.now()
    latest_posts = Post.objects.filter(status='published', publish__lte=now).order_by('-publish')[:count]
    return {'latest_posts': latest_posts}

@register.inclusion_tag('blog/post/most_commented_posts.html')
def get_most_commented_posts(count=5):
    now = timezone.now()
    most_commented_posts = Post.objects.filter(status='published', publish__lte=now).annotate(
        total_comments=Count('comments')
    ).order_by('-total_comments')[:count]
    return {'most_commented_posts': most_commented_posts}

@register.filter(name='markdown')
def markdown_format(text):
    return mark_safe(markdown.markdown(text))
```

Editing the blog_tags.py file



mysite > blog > templates > blog > post > post_list.html > ...

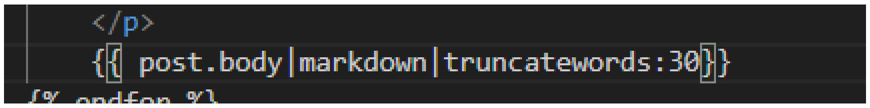
```
{% extends "blog/base.html" %}
{% load blog_tags %}
{% block title %} My Blog{% endblock %}
```



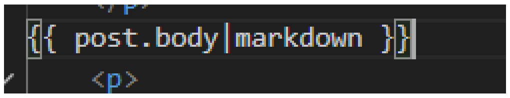
mysite > blog > templates > blog > post > post_detail.html > ...

```
{% extends "blog/base.html" %}
{% load blog_tags %}
{% block title %}{{ post.title }}{% endblock %}
{% block content %}
    <h1>{{ post.title }}</h1>
    <p class="date">
```

blog/post/list.html and blog/post/detail.html templates after the {% extends %} tag

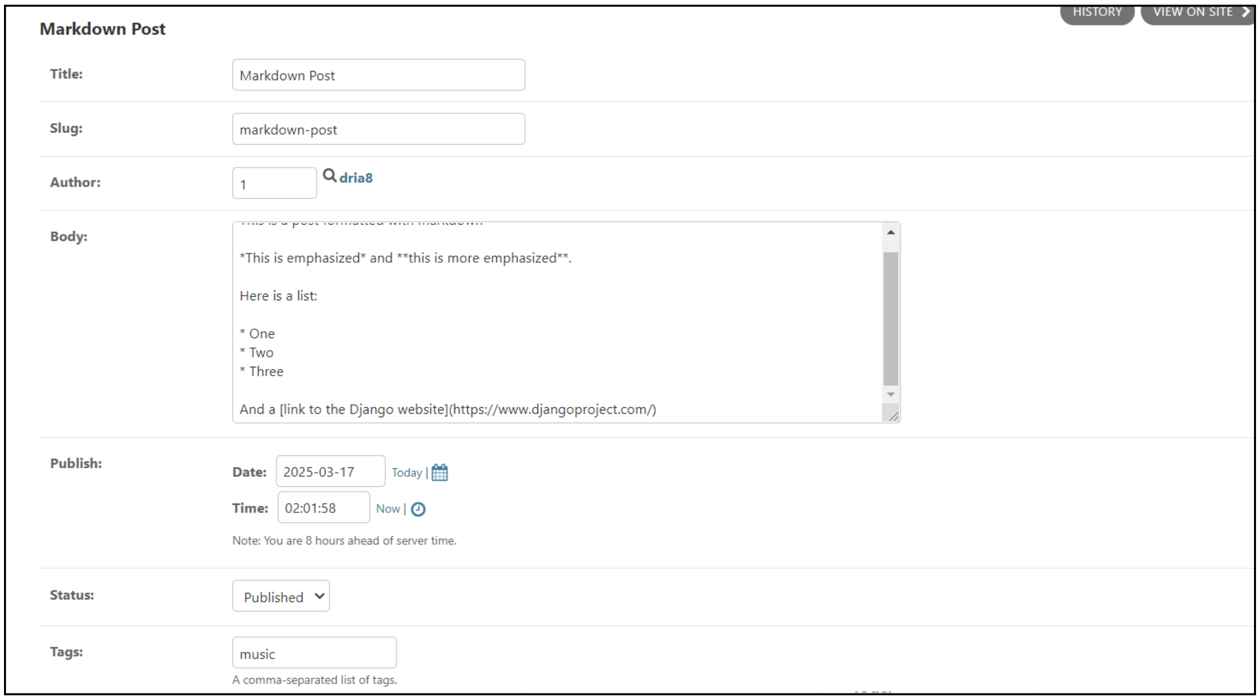


```
    </p>
    {{ post.body|markdown|truncatewords:30}}
```



```
{{ post.body|markdown }}
    <p>
```

Editing linebreaks



Markdown Post

Title: Markdown Post

Slug: markdown-post

Author: 1 dria8

Body:

This is emphasized and **this is more emphasized**.

Here is a list:

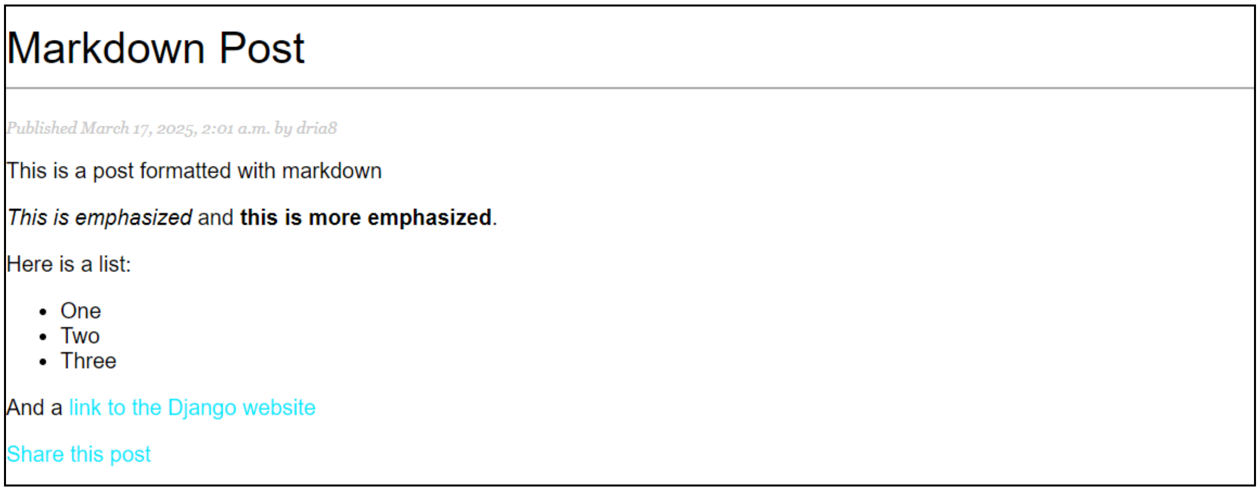
* One
* Two
* Three

And a [link to the Django website](https://www.djangoproject.com/)

Publish:
Date: 2025-03-17 Today |
Time: 02:01:58 Now |
Note: You are 8 hours ahead of server time.

Status: Published

Tags: music
A comma-separated list of tags.


Markdown Post

Published March 17, 2025, 2:01 a.m. by dria8

This is a post formatted with markdown

This is emphasized and **this is more emphasized**.

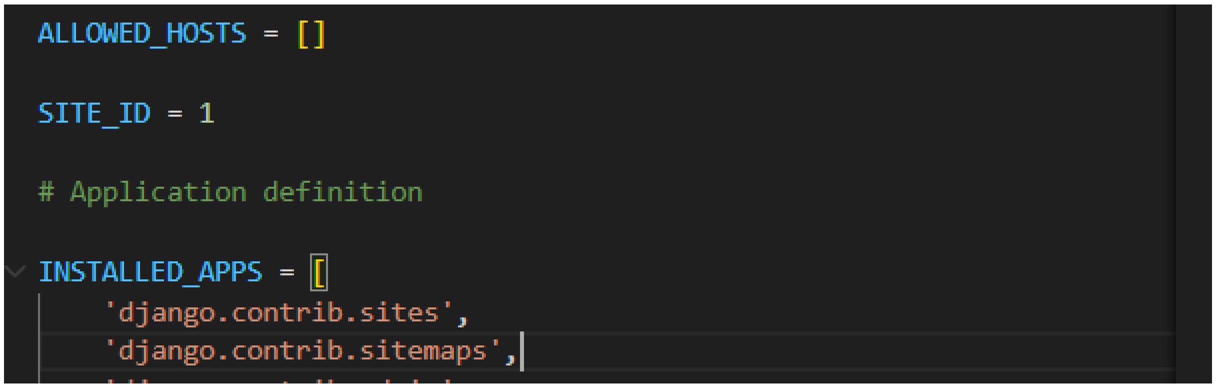
Here is a list:

- One
- Two
- Three

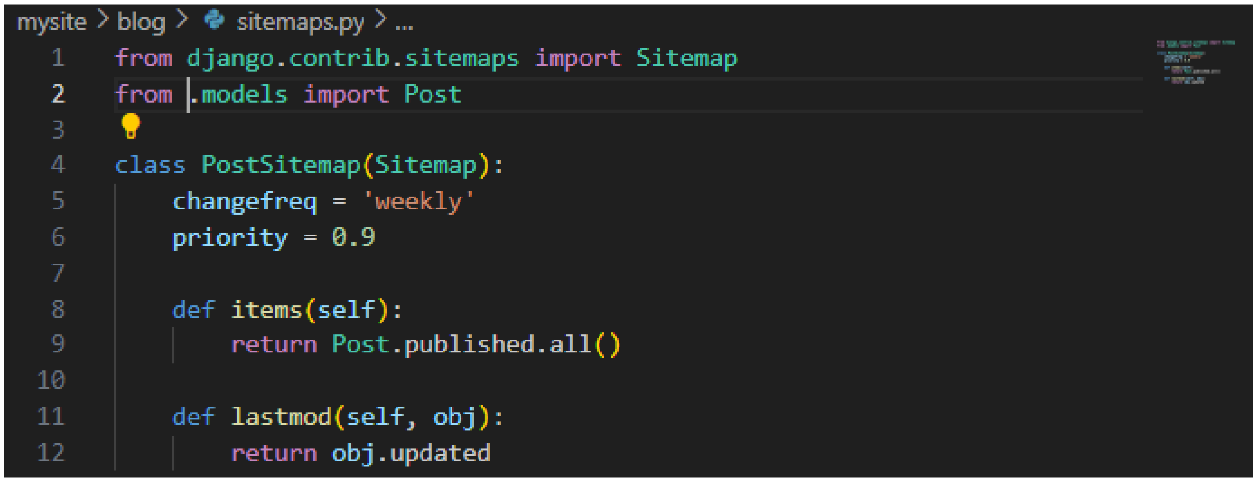
And a link to the Django website

Share this post

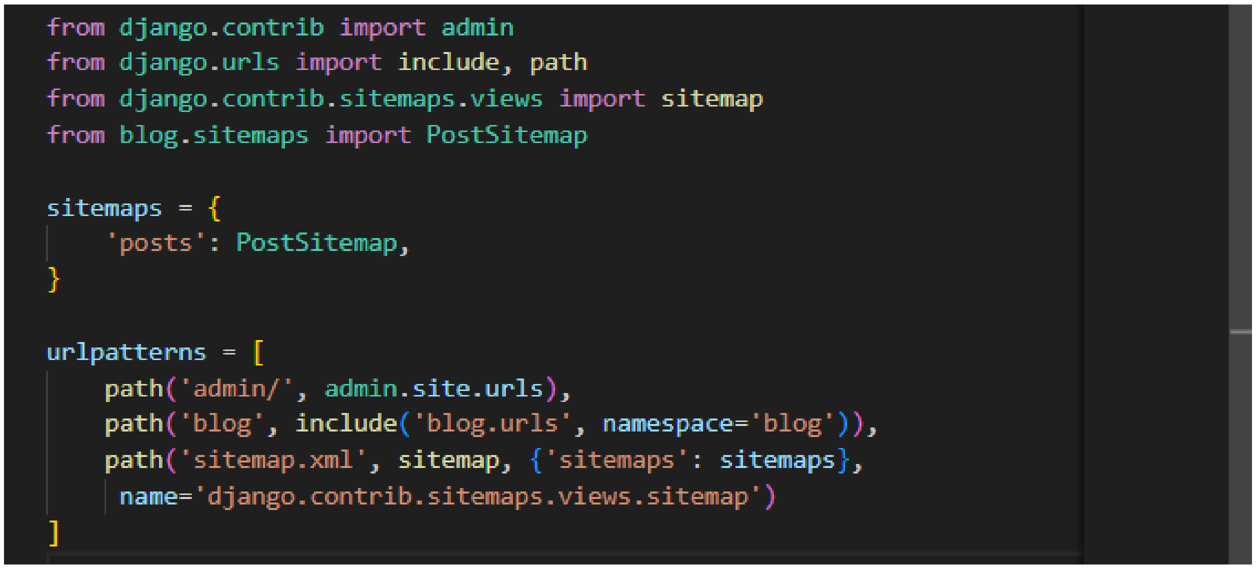
Markdown Post


```
ALLOWED_HOSTS = []

SITE_ID = 1

# Application definition

INSTALLED_APPS = [
    'django.contrib.sites',
    'django.contrib.sitemaps',
```

Editing the settings.py file of project and add django.contrib.sites and django.contrib.sitemaps to the INSTALLED_APPS setting.

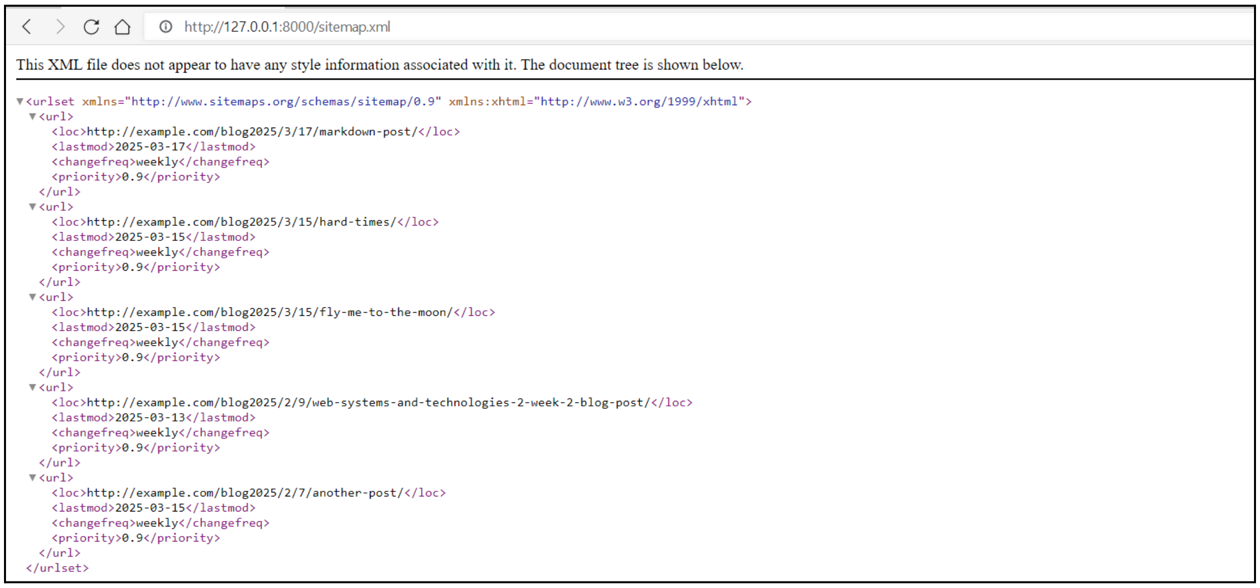

mysite > blog > sitemaps.py > ...
```
from django.contrib.sitemaps import Sitemap
from .models import Post

class PostSitemap(Sitemap):
    changefreq = 'weekly'
    priority = 0.9

    def items(self):
        return Post.published.all()

    def lastmod(self, obj):
        return obj.updated
```

Creating new file sitemaps.py.



```python
from django.contrib import admin
from django.urls import include, path
from django.contrib.sitemaps.views import sitemap
from blog.sitemaps import PostSitemap

sitemaps = {
    'posts': PostSitemap,
}

urlpatterns = [
    path('admin/', admin.site.urls),
    path('blog', include('blog.urls', namespace='blog')),
    path('sitemap.xml', sitemap, {'sitemaps': sitemaps},
     name='django.contrib.sitemaps.views.sitemap')
]
```

Adding your sitemap URL

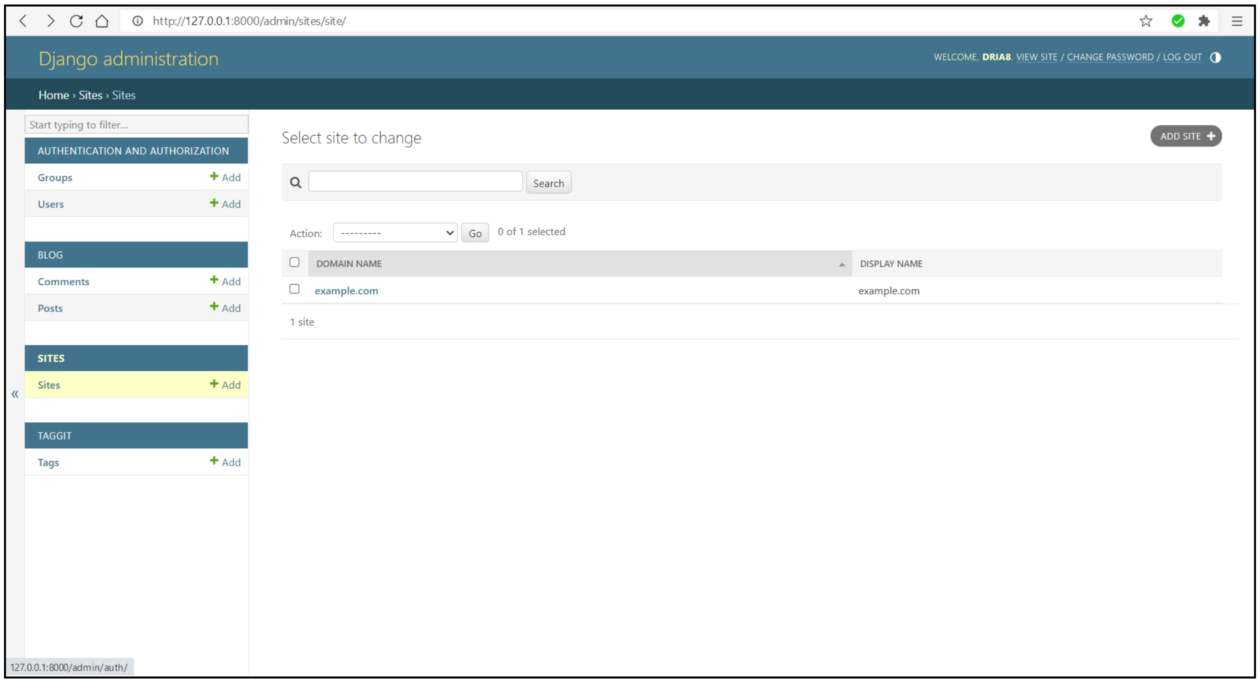


http://127.0.0.1:8000/sitemap.xml

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```xml
▼<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9" xmlns:xhtml="http://www.w3.org/1999/xhtml">
  ▼<url>
     <loc>http://example.com/blog2025/3/17/markdown-post/</loc>
     <lastmod>2025-03-17</lastmod>
     <changefreq>weekly</changefreq>
     <priority>0.9</priority>
   </url>
  ▼<url>
     <loc>http://example.com/blog2025/3/15/hard-times/</loc>
     <lastmod>2025-03-15</lastmod>
     <changefreq>weekly</changefreq>
     <priority>0.9</priority>
   </url>
  ▼<url>
     <loc>http://example.com/blog2025/3/15/fly-me-to-the-moon/</loc>
     <lastmod>2025-03-15</lastmod>
     <changefreq>weekly</changefreq>
     <priority>0.9</priority>
   </url>
  ▼<url>
     <loc>http://example.com/blog2025/2/9/web-systems-and-technologies-2-week-2-blog-post/</loc>
     <lastmod>2025-03-13</lastmod>
     <changefreq>weekly</changefreq>
     <priority>0.9</priority>
   </url>
  ▼<url>
     <loc>http://example.com/blog2025/2/7/another-post/</loc>
     <lastmod>2025-03-15</lastmod>
     <changefreq>weekly</changefreq>
     <priority>0.9</priority>
   </url>
 </urlset>
```

sitemap.xml



http://127.0.0.1:8000/admin/sites/site/

Django administration

WELCOME, DRIA8. VIEW SITE / CHANGE PASSWORD / LOG OUT

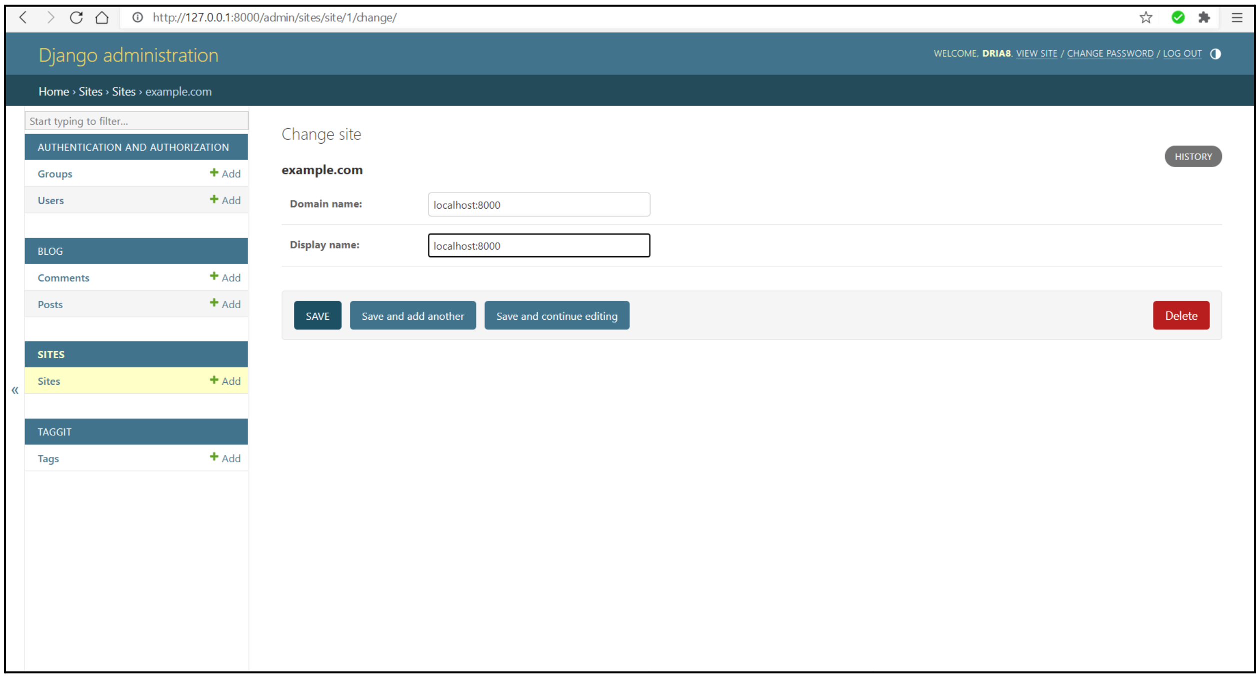
Home › Sites › Sites

Select site to change

Action: --------- Go 0 of 1 selected

DOMAIN NAME	DISPLAY NAME
example.com	example.com

1 site



http://127.0.0.1:8000/admin/sites/site/1/change/

Django administration

WELCOME, DRIA8. VIEW SITE / CHANGE PASSWORD / LOG OUT

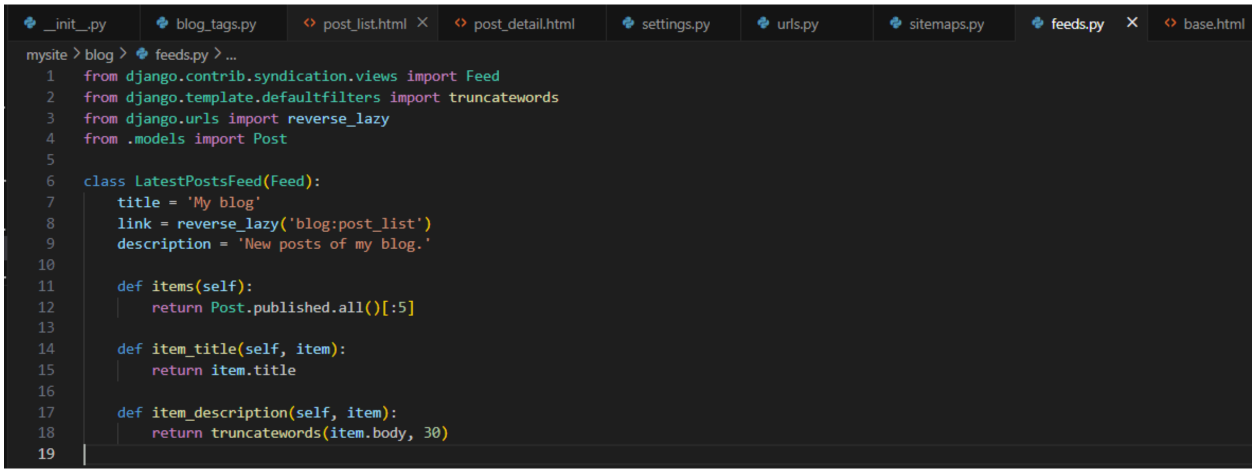
Home › Sites › Sites › example.com

Change site

example.com

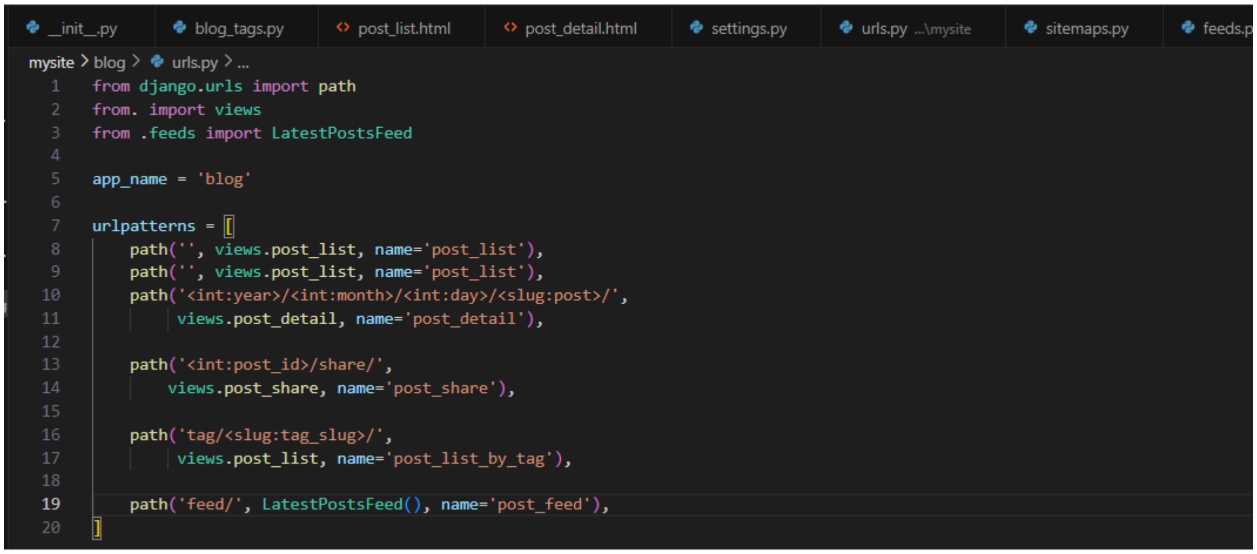
Domain name:	localhost:8000
Display name:	localhost:8000

Sitemap



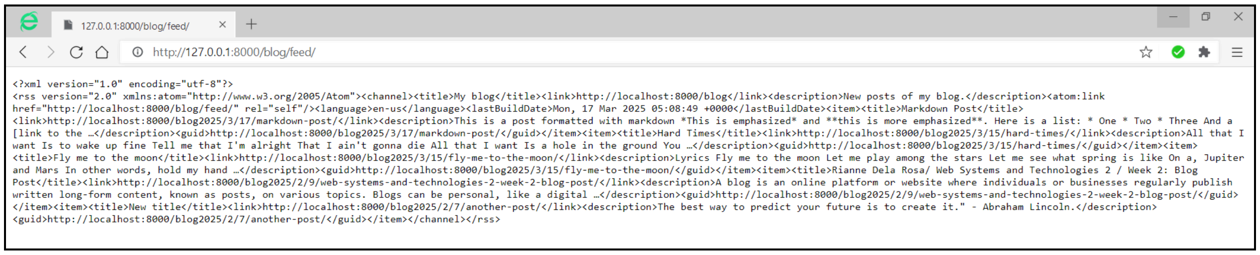
```python
from django.contrib.syndication.views import Feed
from django.template.defaultfilters import truncatewords
from django.urls import reverse_lazy
from .models import Post

class LatestPostsFeed(Feed):
    title = 'My blog'
    link = reverse_lazy('blog:post_list')
    description = 'New posts of my blog.'

    def items(self):
        return Post.published.all()[:5]

    def item_title(self, item):
        return item.title

    def item_description(self, item):
        return truncatewords(item.body, 30)

```

Creating new file feeds.py



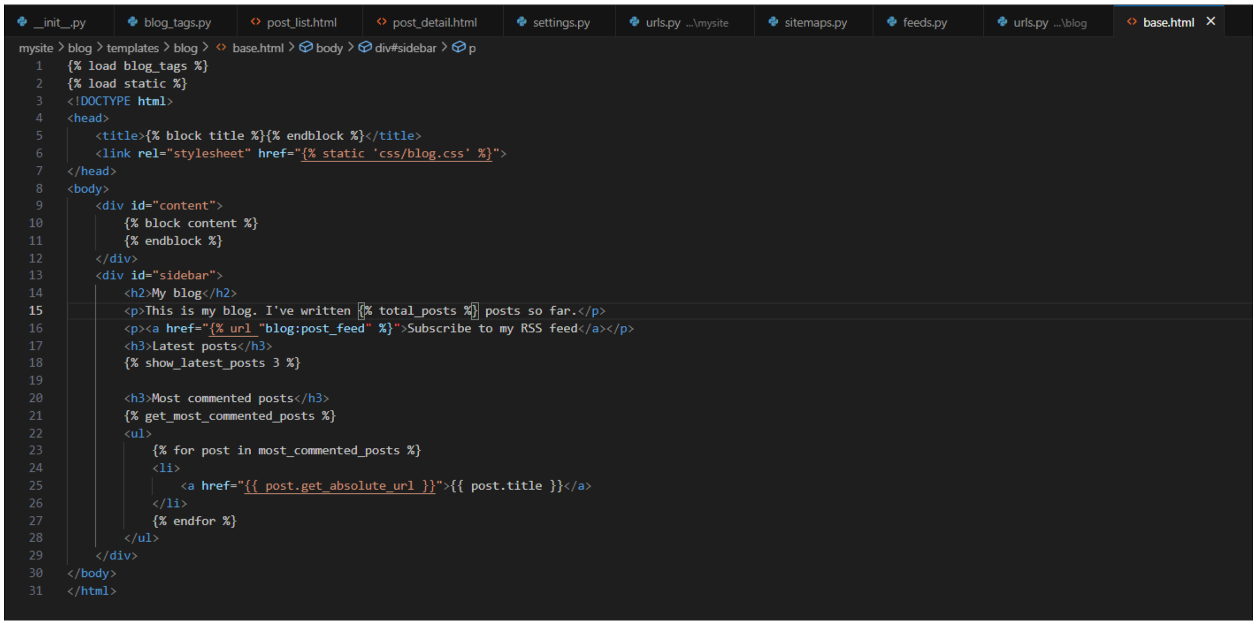
```python
from django.urls import path
from. import views
from .feeds import LatestPostsFeed

app_name = 'blog'

urlpatterns = [
    path('', views.post_list, name='post_list'),
    path('', views.post_list, name='post_list'),
    path('<int:year>/<int:month>/<int:day>/<slug:post>/',
         views.post_detail, name='post_detail'),

    path('<int:post_id>/share/',
        views.post_share, name='post_share'),

    path('tag/<slug:tag_slug>/',
         views.post_list, name='post_list_by_tag'),

    path('feed/', LatestPostsFeed(), name='post_feed'),
]
```

Editing the blog/urls.py file



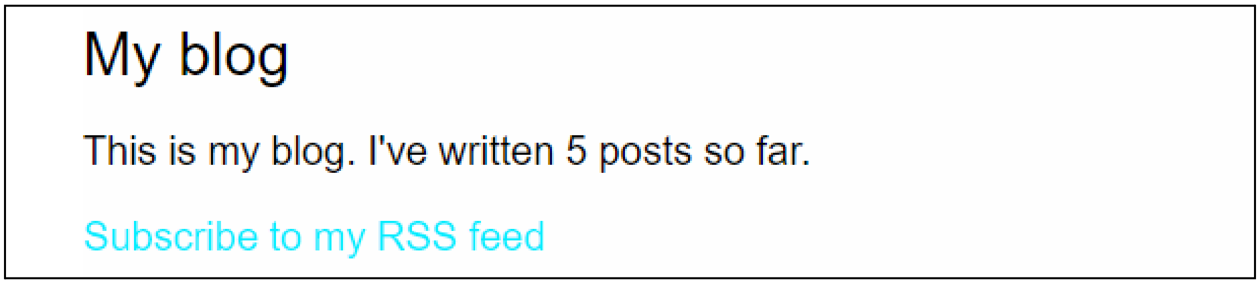
```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0" xmlns:atom="http://www.w3.org/2005/Atom"><channel><title>My blog</title><link>http://localhost:8000/blog</link><description>New posts of my blog.</description><atom:link
href="http://localhost:8000/blog/feed/" rel="self"/><language>en-us</language><lastBuildDate>Mon, 17 Mar 2025 05:08:49 +0000</lastBuildDate><item><title>Markdown Post</title>
<link>http://localhost:8000/blog2025/3/17/markdown-post/</link><description>This is a post formatted with markdown *This is emphasized* and **this is more emphasized**. Here is a list: * One * Two * Three And a
[link to the …</description><guid>http://localhost:8000/blog2025/3/17/markdown-post/</guid></item><item><title>Hard Times</title><link>http://localhost:8000/blog2025/3/15/hard-times/</link><description>All that I
want Is to wake up fine Tell me that I'm alright That I ain't gonna die All that I want Is a hole in the ground You …</description><guid>http://localhost:8000/blog2025/3/15/hard-times/</guid></item><item>
<title>Fly me to the moon</title><link>http://localhost:8000/blog2025/3/15/fly-me-to-the-moon/</link><description>Lyrics Fly me to the moon Let me play among the stars Let me see what spring is like On a, Jupiter
and Mars In other words, hold my hand …</description><guid>http://localhost:8000/blog2025/3/15/fly-me-to-the-moon/</guid></item><item><title>Rianne Dela Rosa/ Web Systems and Technologies 2 / Week 2: Blog
Post</title><link>http://localhost:8000/blog2025/2/9/web-systems-and-technologies-2-week-2-blog-post/</link><description>A blog is an online platform or website where individuals or businesses regularly publish
written long-form content, known as posts, on various topics. Blogs can be personal, like a digital …</description><guid>http://localhost:8000/blog2025/2/9/web-systems-and-technologies-2-week-2-blog-post/</guid>
</item><item><title>New title</title><link>http://localhost:8000/blog2025/2/7/another-post/</link><description>The best way to predict your future is to create it." - Abraham Lincoln.</description>
<guid>http://localhost:8000/blog2025/2/7/another-post/</guid></item></channel></rss>
```

Blog feed



```
{% load blog_tags %}
{% load static %}
<!DOCTYPE html>
<head>
    <title>{% block title %}{% endblock %}</title>
    <link rel="stylesheet" href="{% static 'css/blog.css' %}">
</head>
<body>
    <div id="content">
        {% block content %}
        {% endblock %}
    </div>
    <div id="sidebar">
        <h2>My blog</h2>
        <p>This is my blog. I've written {% total_posts %} posts so far.</p>
        <p><a href="{% url "blog:post_feed" %}">Subscribe to my RSS feed</a></p>
        <h3>Latest posts</h3>
        {% show_latest_posts 3 %}

        <h3>Most commented posts</h3>
        {% get_most_commented_posts %}
        <ul>
            {% for post in most_commented_posts %}
            <li>
                <a href="{{ post.get_absolute_url }}">{{ post.title }}</a>
            </li>
            {% endfor %}
        </ul>
    </div>
</body>
</html>
```

Editing base.html


My blog

This is my blog. I've written 5 posts so far.

Subscribe to my RSS feed

The new link taking to the blog's feed

Reflection

From my perspective, encountering the "AttributeError at /blog/feed/ type object 'Post' has no attribute 'published'" was a moment that demanded a shift in focus. Initially, I was operating under the assumption that the Post model possessed a custom manager or field named published, leading me to concentrate on verifying its existence and correct implementation. However, the error's persistence forced a re-evaluation, revealing a discrepancy between my assumed model structure and the reality. The challenge wasn't merely about fixing a line of code; it was about reconciling my understanding with the application's actual state. This process underscored the importance of not taking assumptions for granted, especially in complex systems like Django applications. It highlighted the necessity of thorough examination, even when dealing with seemingly straightforward errors. I learned that relying on assumptions, even if they seem logical, can lead to prolonged debugging sessions. Instead, systematically verifying each component, from the model definition to the feed generation logic, proved to be the more effective approach. This experience reinforced the value of meticulous code analysis and the need to maintain a clear understanding of the application's architecture. It also provided practical exposure to debugging RSS feed generation in Django, an area I hadn't explored in depth before. This hands-on experience will undoubtedly be beneficial in future projects, enabling me to approach similar challenges with greater confidence and efficiency.