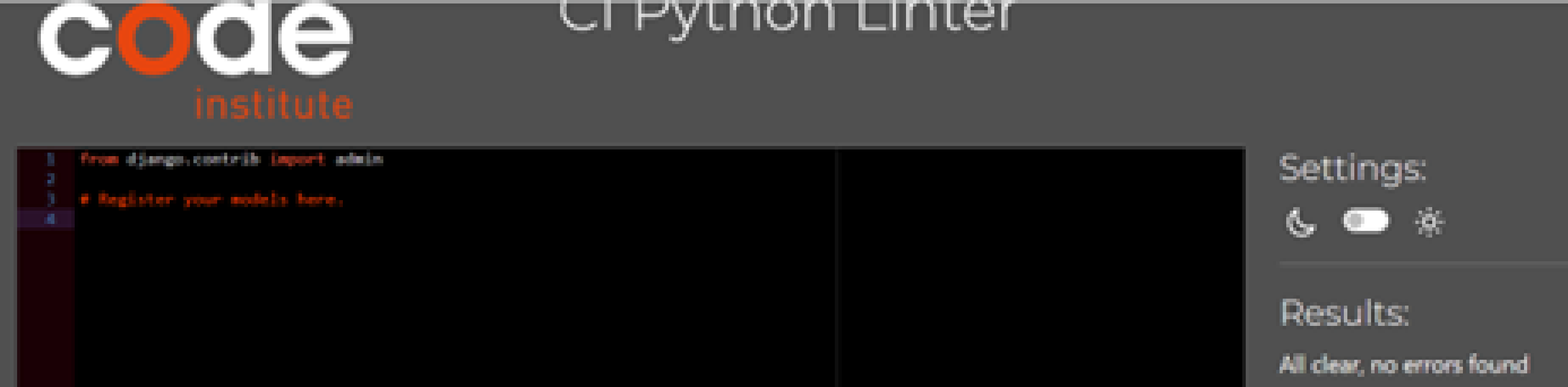


code institute

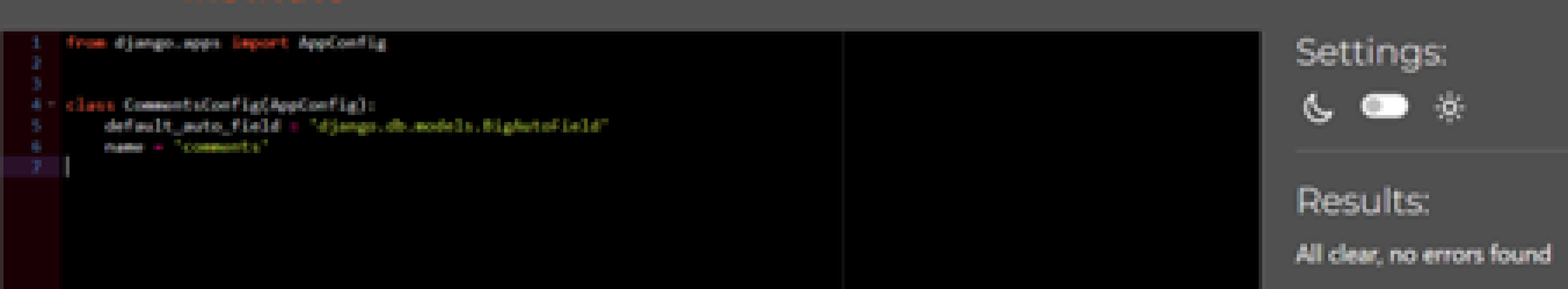
CI Python Linter

```
from django.contrib import admin

# Register your models here.

```

Settings:

Results:

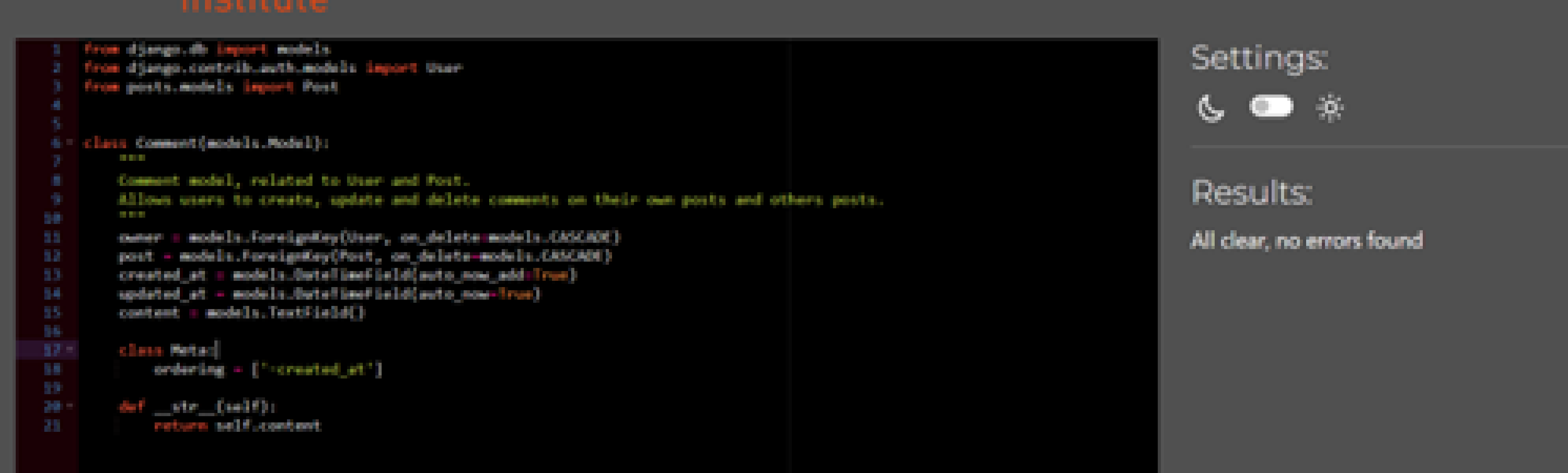
All clear, no errors found



```python
from django.apps import AppConfig


class CommentsConfig(AppConfig):
    default_auto_field = "django.db.models.BigAutoField"
    name = 'comments'
```

Results:

All clear, no errors found

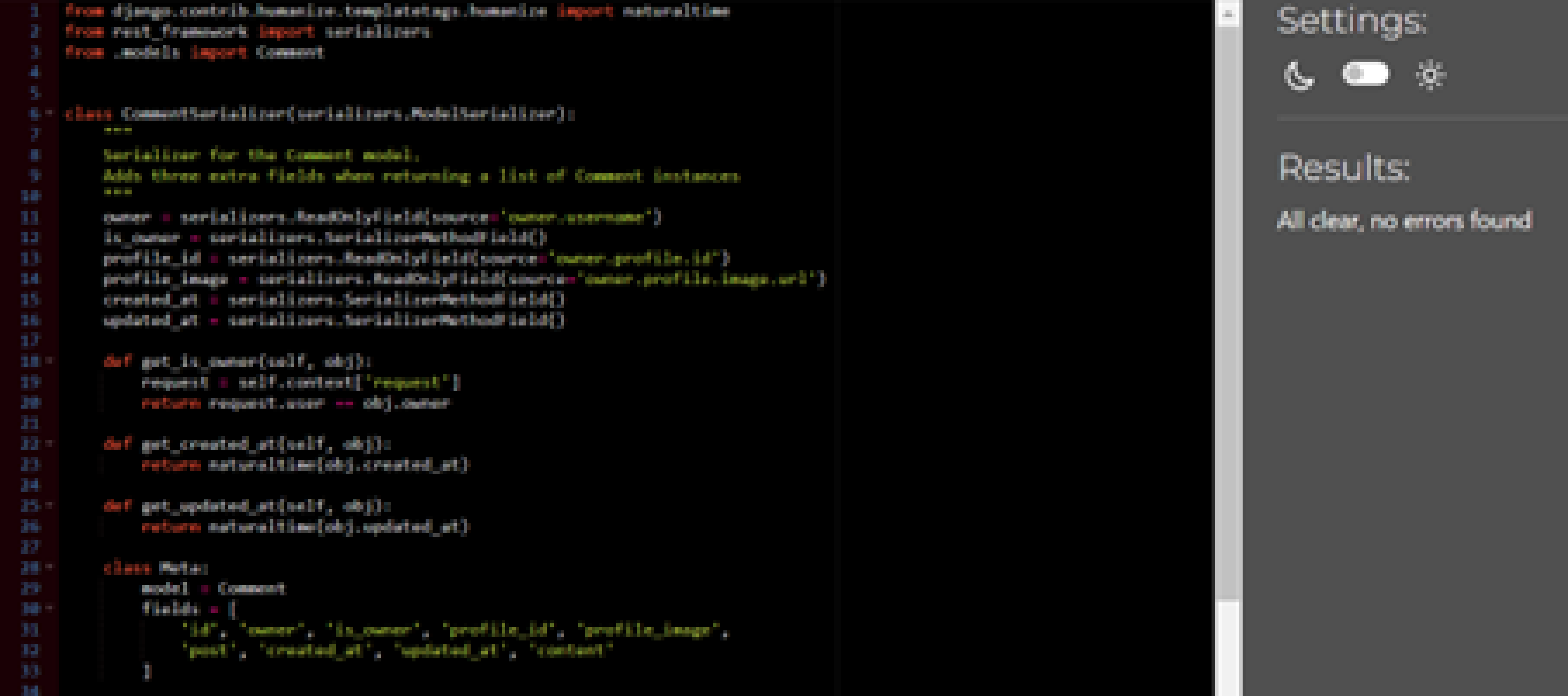


```
from django.db import models
from django.contrib.auth.models import User
from posts.models import Post


class Comment(models.Model):
    """
    Comment model, related to User and Post.
    Allows users to create, update and delete comments on their own posts and others posts.
    """
    owner = models.ForeignKey(User, on_delete=models.CASCADE)
    post = models.ForeignKey(Post, on_delete=models.CASCADE)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    content = models.TextField()

    class Meta:
        ordering = ['-created_at']

    def __str__(self):
        return self.content
```

Settings:

Results:

All clear, no errors found

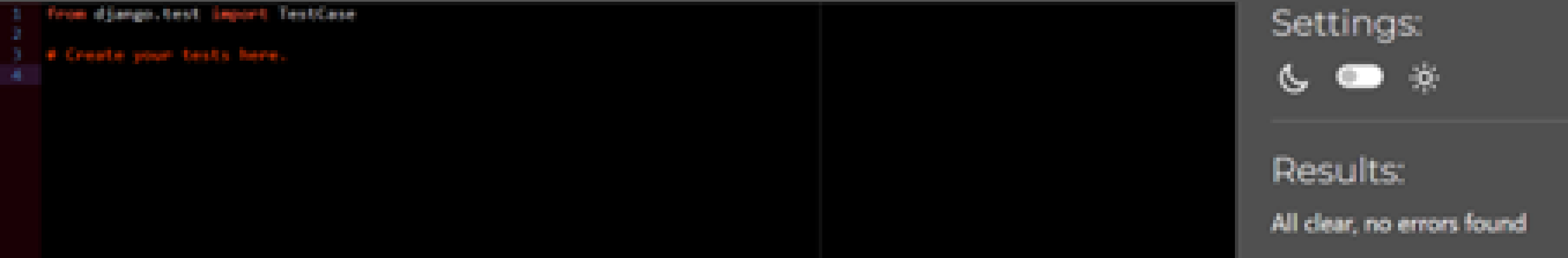


```
from django.contrib.humanize.templatetags.humanize import naturaltime
from rest_framework import serializers
from .models import Comment


class CommentSerializer(serializers.ModelSerializer):
    """
    Serializer for the Comment model.
    Adds three extra fields when returning a list of Comment instances
    """
    owner = serializers.ReadOnlyField(source='owner.username')
    is_owner = serializers.SerializerMethodField()
    profile_id = serializers.ReadOnlyField(source='owner.profile.id')
    profile_image = serializers.ReadOnlyField(source='owner.profile.image.url')
    created_at = serializers.SerializerMethodField()
    updated_at = serializers.SerializerMethodField()

    def get_is_owner(self, obj):
        request = self.context['request']
        return request.user == obj.owner

    def get_created_at(self, obj):
        return naturaltime(obj.created_at)

    def get_updated_at(self, obj):
        return naturaltime(obj.updated_at)

    class Meta:
        model = Comment
        fields = [
            'id', 'owner', 'is_owner', 'profile_id', 'profile_image',
            'post', 'created_at', 'updated_at', 'content'
        ]
```

Settings:

Results:

All clear, no errors found

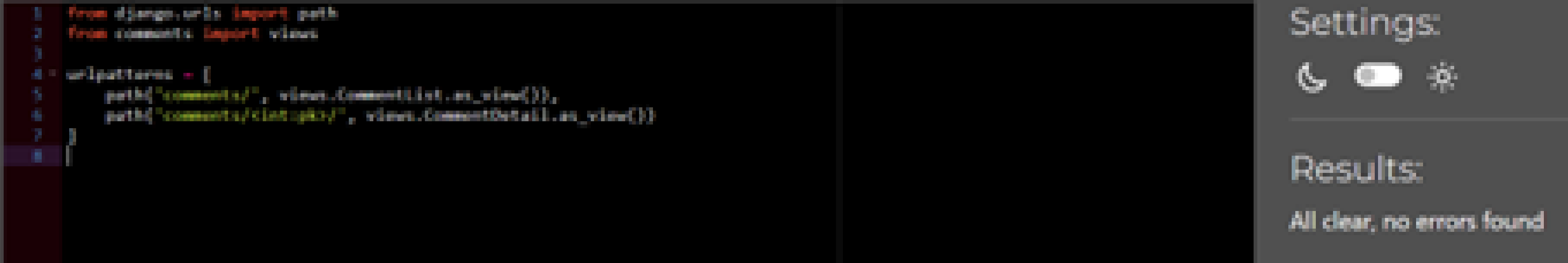


```
from django.test import TestCase

# Create your tests here.

```

Settings:

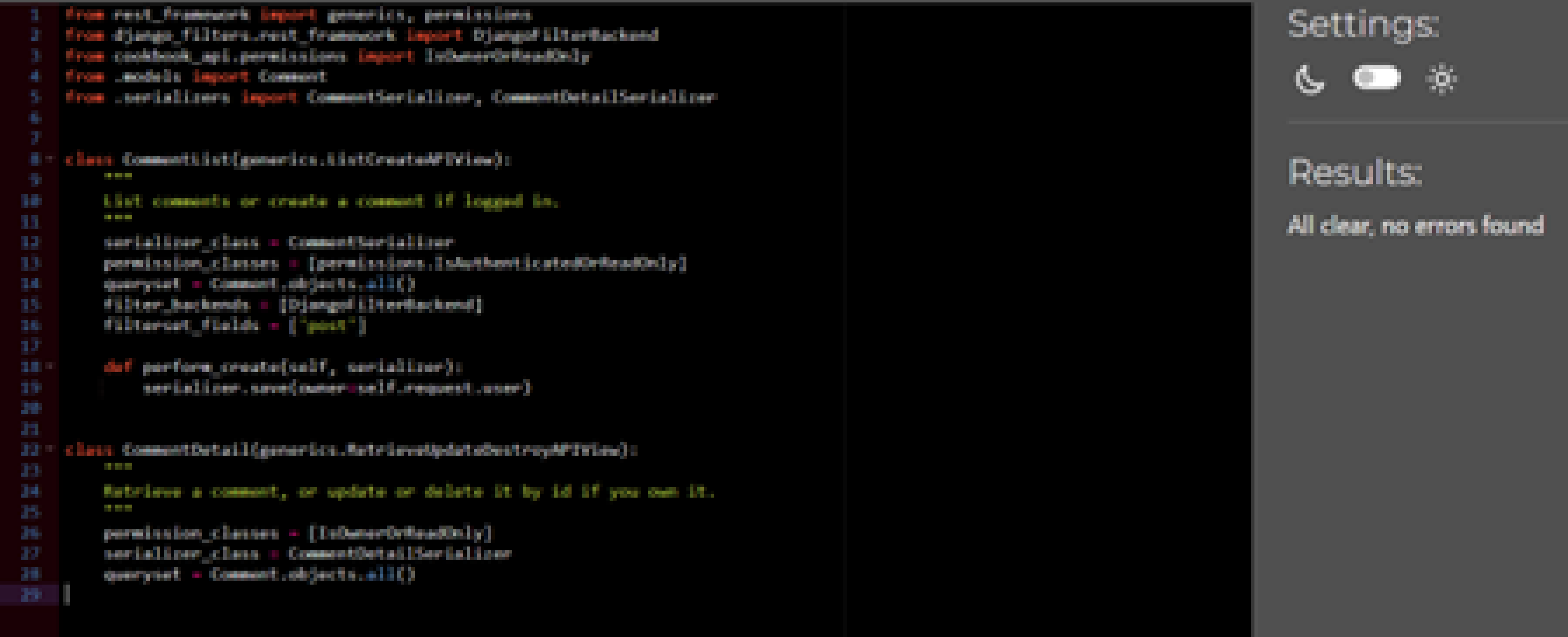
Results:

All clear, no errors found



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

urlpatterns = [
    path("comments/", views.CommentList.as_view()),
    path("comments/<int:pk>/", views.CommentDetail.as_view())
]
```

All clear, no errors found



```python
from rest_framework import generics, permissions
from django_filters.rest_framework import DjangoFilterBackend
from cookbook_api.permissions import IsOwnerOrReadOnly
from .models import Comment
from .serializers import CommentSerializer, CommentDetailSerializer


class CommentList(generics.ListCreateAPIView):
    """
    List comments or create a comment if logged in.
    """
    serializer_class = CommentSerializer
    permission_classes = [permissions.IsAuthenticatedOrReadOnly]
    queryset = Comment.objects.all()
    filter_backends = [DjangoFilterBackend]
    filterset_fields = ["post"]

    def perform_create(self, serializer):
        serializer.save(owner=self.request.user)


class CommentDetail(generics.RetrieveUpdateDestroyAPIView):
    """
    Retrieve a comment, or update or delete it by id if you own it.
    """
    permission_classes = [IsOwnerOrReadOnly]
    serializer_class = CommentDetailSerializer
    queryset = Comment.objects.all()

```

Settings:

Results:

All clear, no errors found