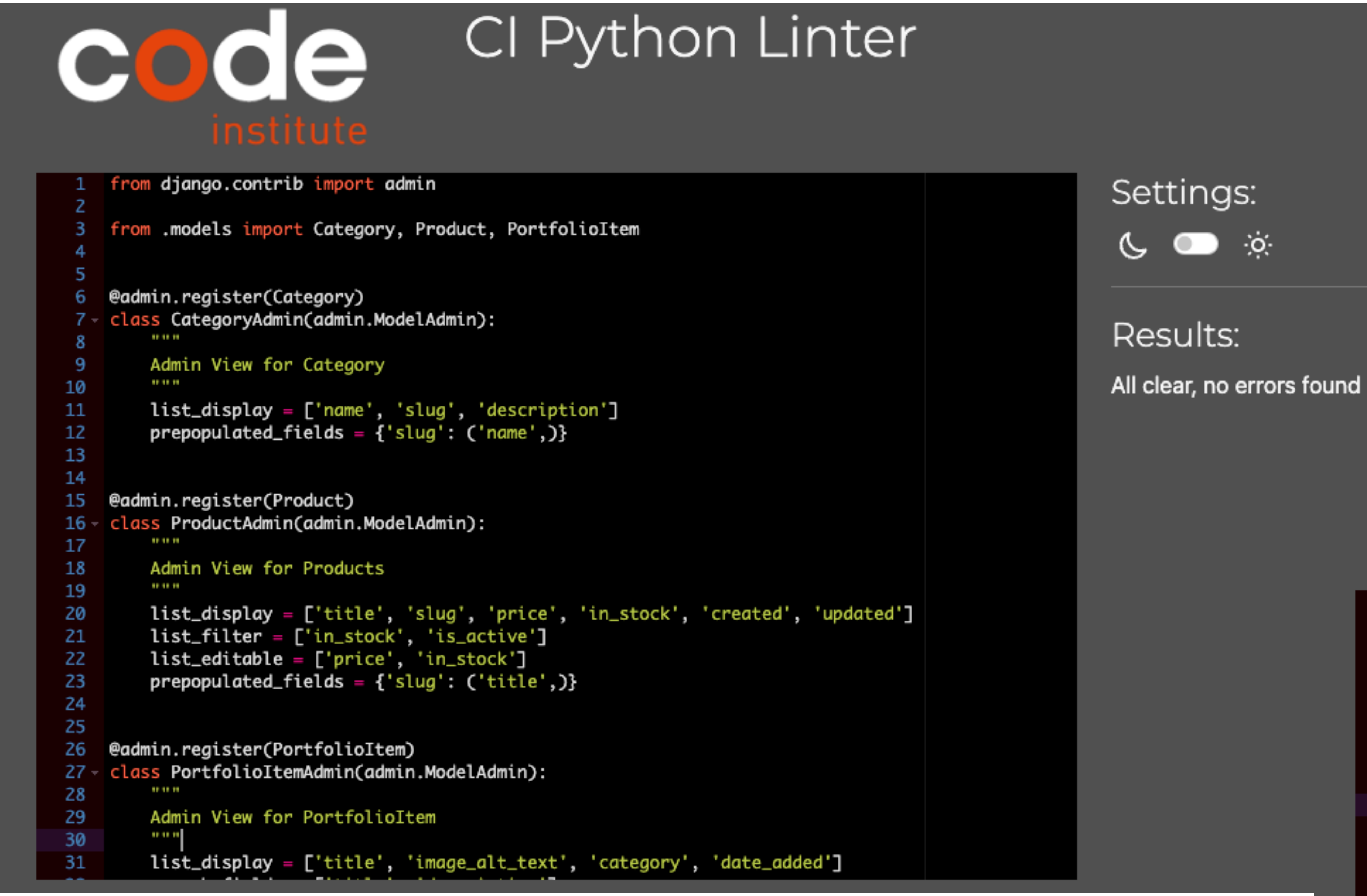
Store

admin.py



CI Python Linter

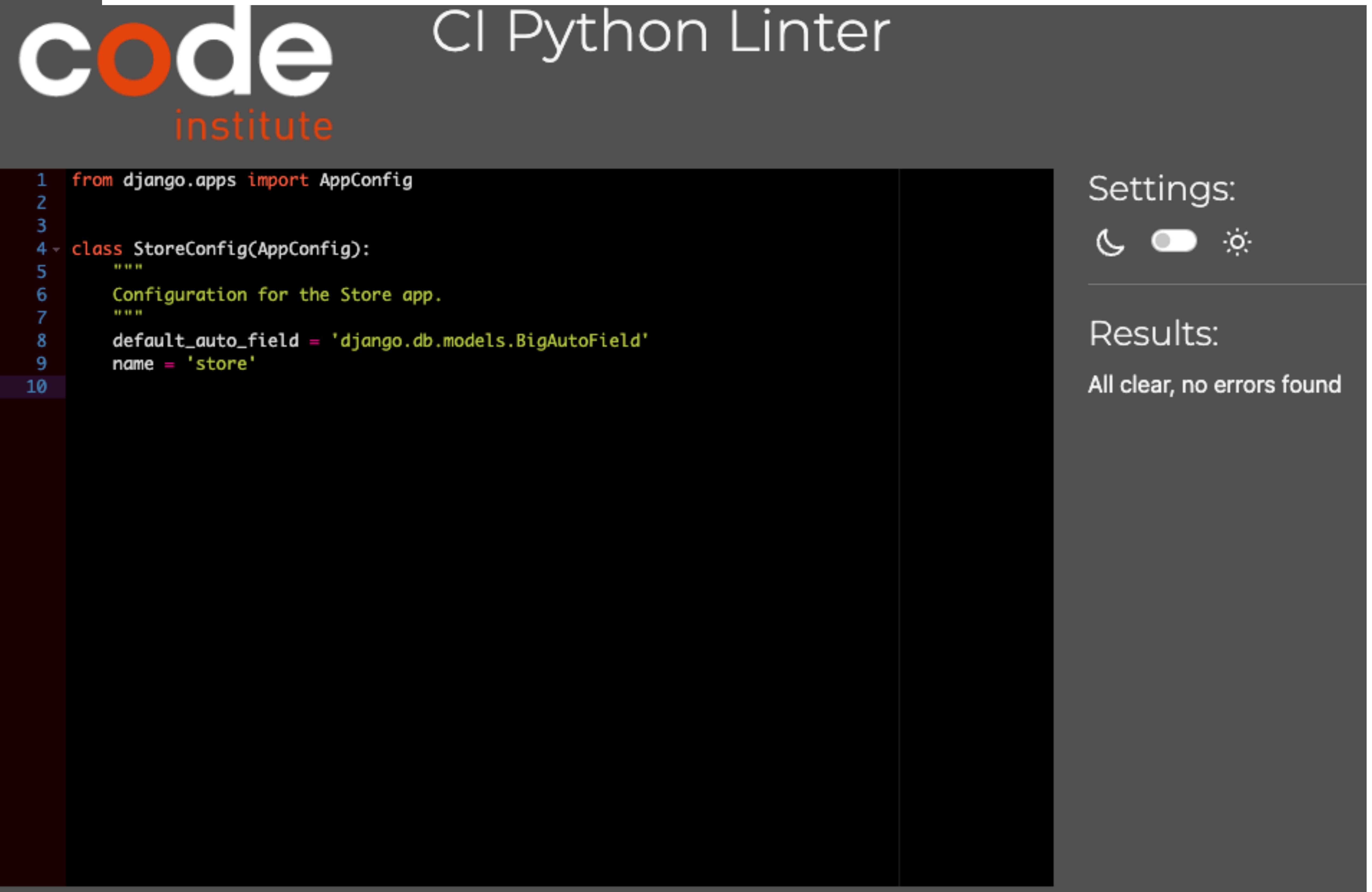
```python
from django.contrib import admin

from .models import Category, Product, PortfolioItem


@admin.register(Category)
class CategoryAdmin(admin.ModelAdmin):
    """
    Admin View for Category
    """
    list_display = ['name', 'slug', 'description']
    prepopulated_fields = {'slug': ('name',)}


@admin.register(Product)
class ProductAdmin(admin.ModelAdmin):
    """
    Admin View for Products
    """
    list_display = ['title', 'slug', 'price', 'in_stock', 'created', 'updated']
    list_filter = ['in_stock', 'is_active']
    list_editable = ['price', 'in_stock']
    prepopulated_fields = {'slug': ('title',)}


@admin.register(PortfolioItem)
class PortfolioItemAdmin(admin.ModelAdmin):
    """
    Admin View for PortfolioItem
    """
    list_display = ['title', 'image_alt_text', 'category', 'date_added']
```

Settings:

Results:

All clear, no errors found

apps.py



CI Python Linter

```python
from django.apps import AppConfig


class StoreConfig(AppConfig):
    """
    Configuration for the Store app.
    """
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'store'
```

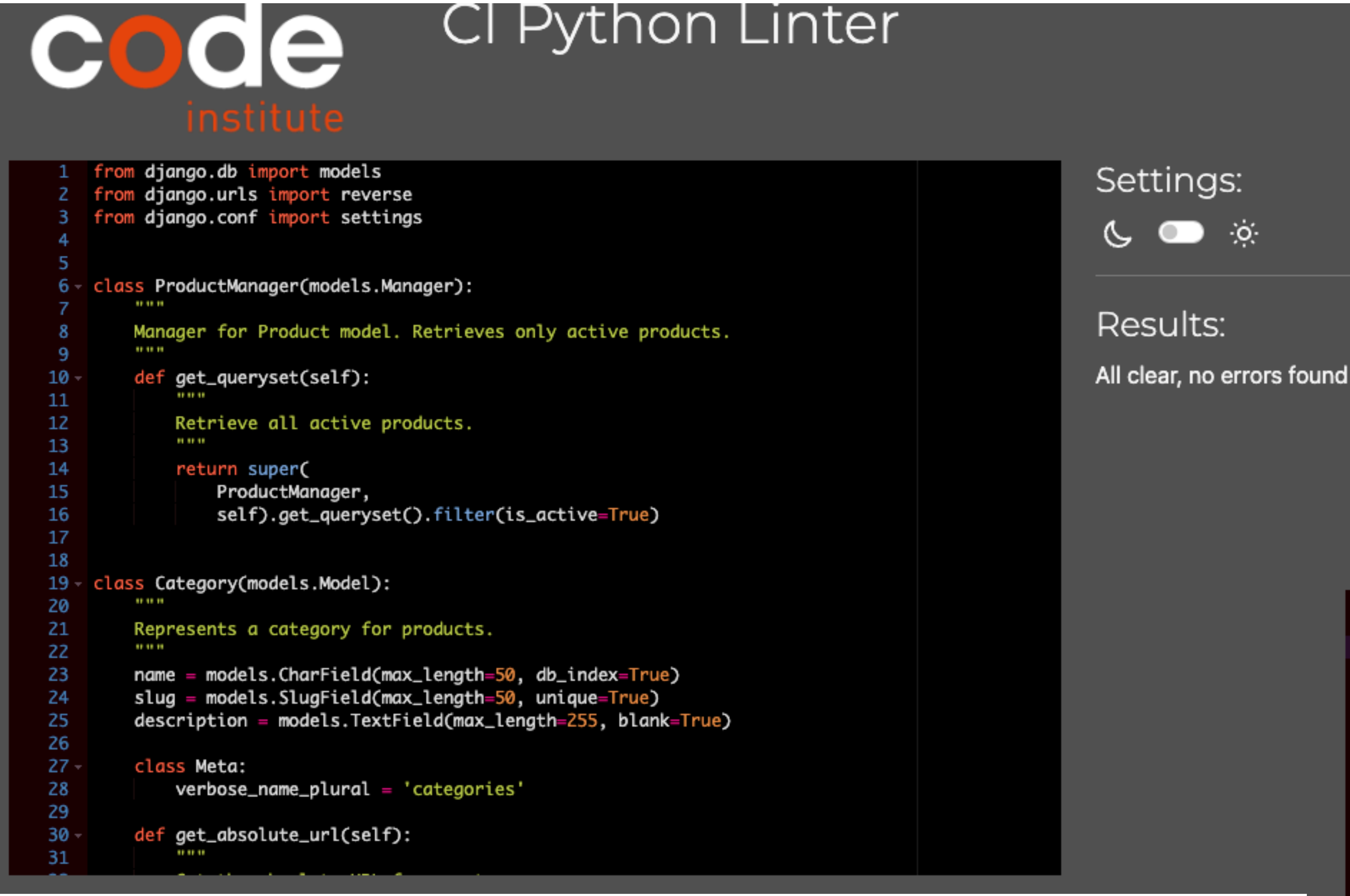
Settings:

Results:

All clear, no errors found

Store

Models.py



CI Python Linter

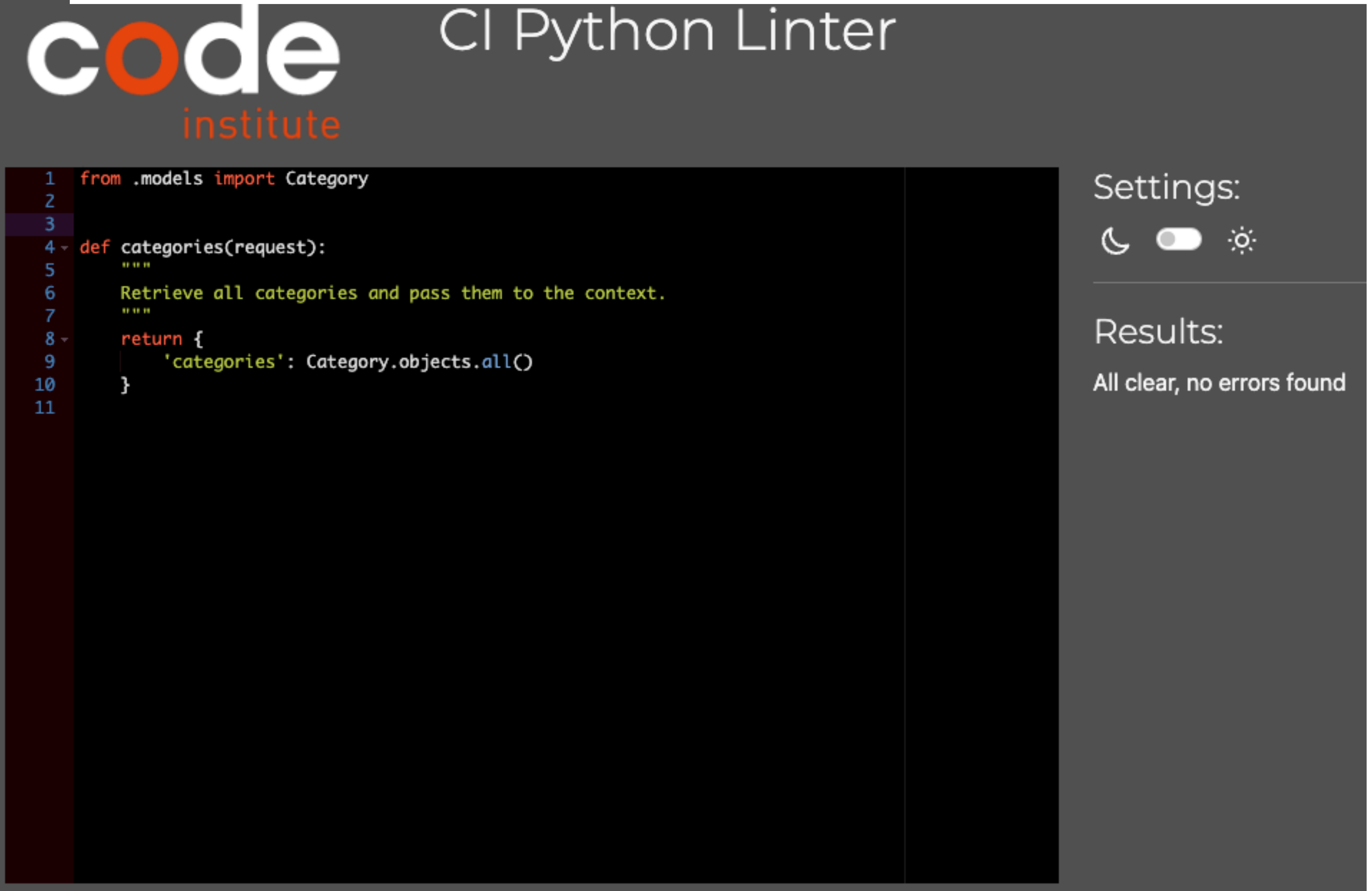
```python
from django.db import models
from django.urls import reverse
from django.conf import settings


class ProductManager(models.Manager):
    """
    Manager for Product model. Retrieves only active products.
    """
    def get_queryset(self):
        """
        Retrieve all active products.
        """
        return super(
            ProductManager,
            self).get_queryset().filter(is_active=True)


class Category(models.Model):
    """
    Represents a category for products.
    """
    name = models.CharField(max_length=50, db_index=True)
    slug = models.SlugField(max_length=50, unique=True)
    description = models.TextField(max_length=255, blank=True)

    class Meta:
        verbose_name_plural = 'categories'

    def get_absolute_url(self):
        """
```

Settings:

Results:

All clear, no errors found

context_processors.py



CI Python Linter

```python
from .models import Category


def categories(request):
    """
    Retrieve all categories and pass them to the context.
    """
    return {
        'categories': Category.objects.all()
    }
```

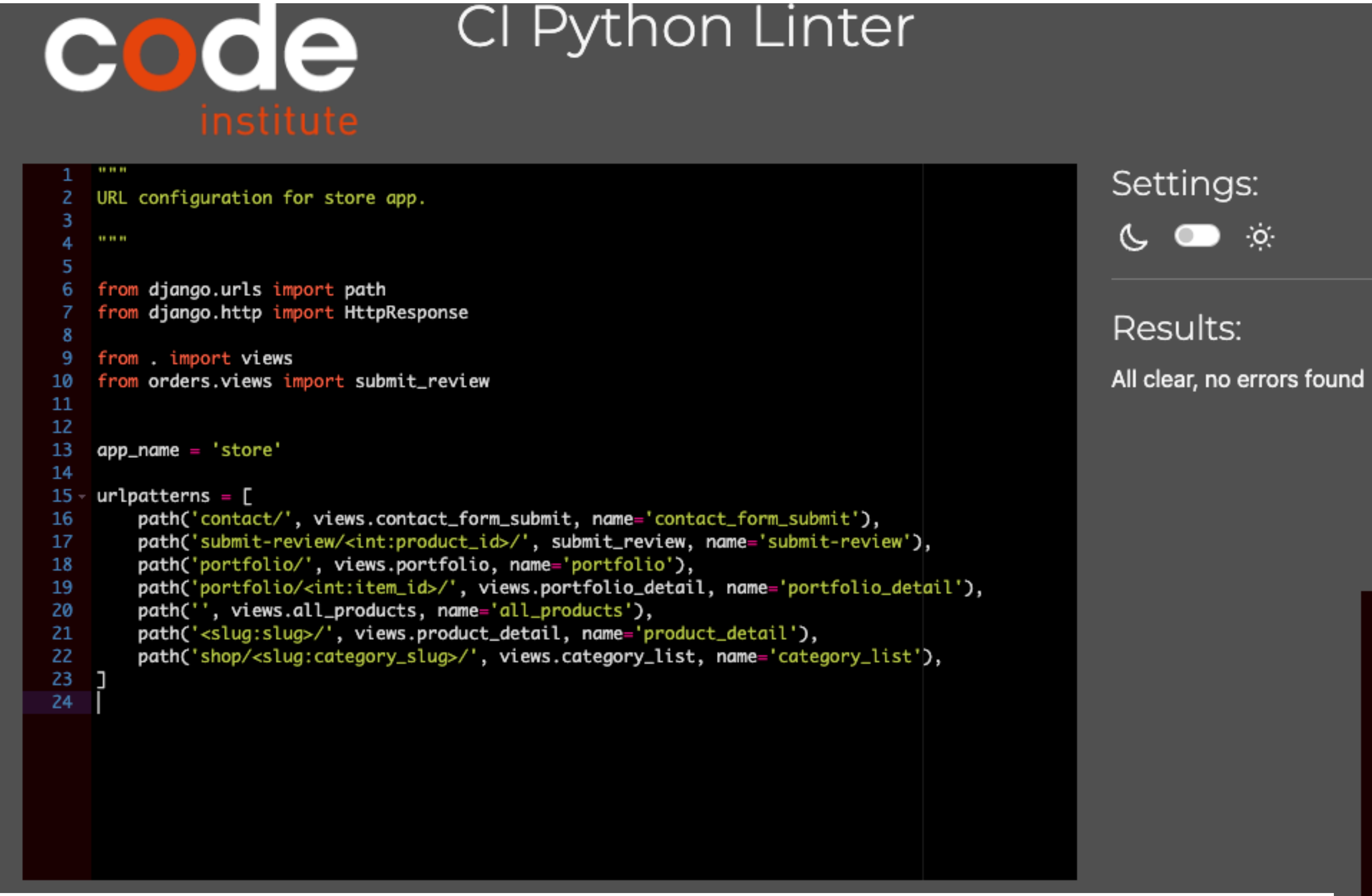
Settings:

Results:

All clear, no errors found

Store

Urls.py

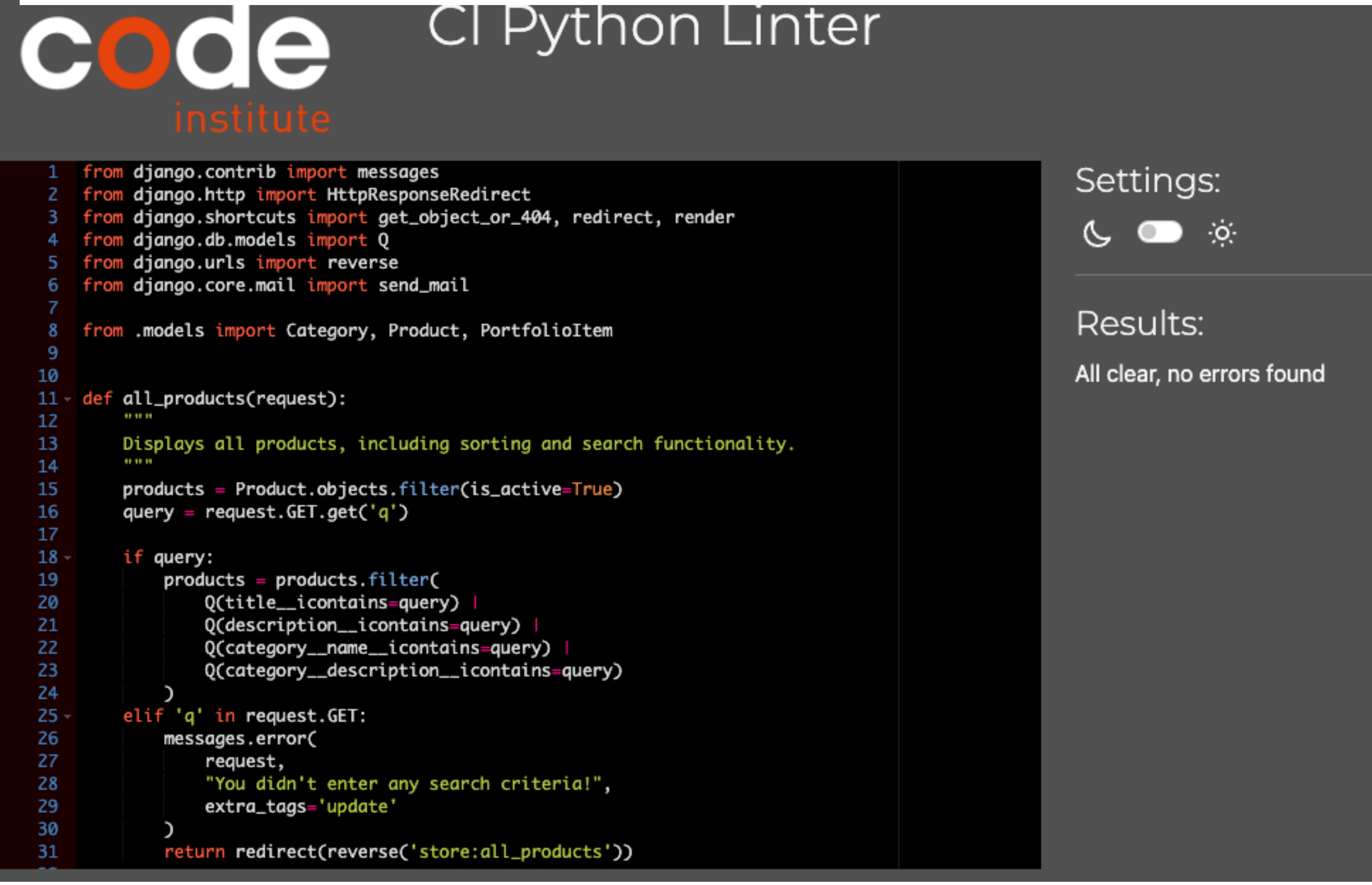


```python
"""
URL configuration for store app.

"""

from django.urls import path
from django.http import HttpResponse

from . import views
from orders.views import submit_review


app_name = 'store'

urlpatterns = [
    path('contact/', views.contact_form_submit, name='contact_form_submit'),
    path('submit-review/<int:product_id>/', submit_review, name='submit-review'),
    path('portfolio/', views.portfolio, name='portfolio'),
    path('portfolio/<int:item_id>/', views.portfolio_detail, name='portfolio_detail'),
    path('', views.all_products, name='all_products'),
    path('<slug:slug>/', views.product_detail, name='product_detail'),
    path('shop/<slug:category_slug>/', views.category_list, name='category_list'),
]

```

Results:

All clear, no errors found

views.py



```python
from django.contrib import messages
from django.http import HttpResponseRedirect
from django.shortcuts import get_object_or_404, redirect, render
from django.db.models import Q
from django.urls import reverse
from django.core.mail import send_mail

from .models import Category, Product, PortfolioItem


def all_products(request):
    """
    Displays all products, including sorting and search functionality.
    """
    products = Product.objects.filter(is_active=True)
    query = request.GET.get('q')

    if query:
        products = products.filter(
            Q(title__icontains=query) |
            Q(description__icontains=query) |
            Q(category__name__icontains=query) |
            Q(category__description__icontains=query)
        )
    elif 'q' in request.GET:
        messages.error(
            request,
            "You didn't enter any search criteria!",
            extra_tags='update'
        )
        return redirect(reverse('store:all_products'))
```

Results:

All clear, no errors found