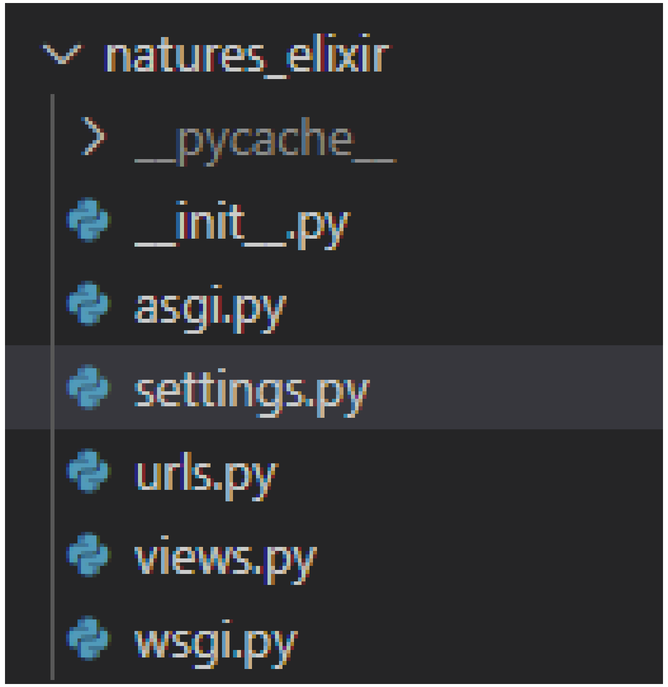
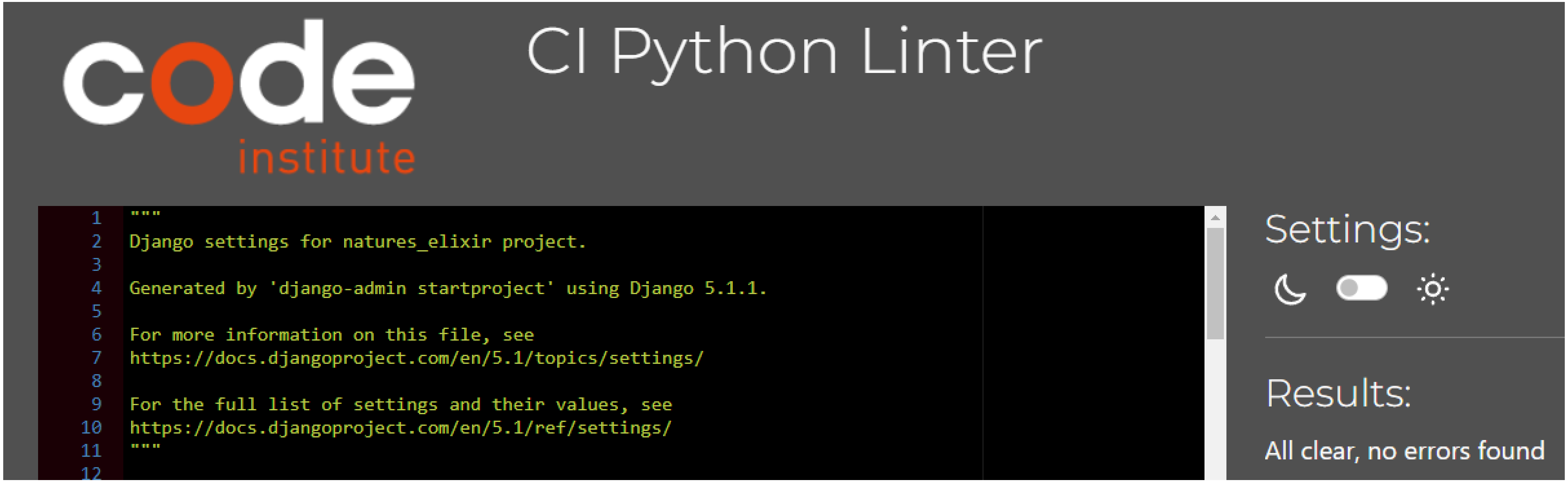
natures-elixir



```
natures_elixir
  __pycache__
  __init__.py
  asgi.py
  settings.py
  urls.py
  views.py
  wsgi.py
```

settings.py

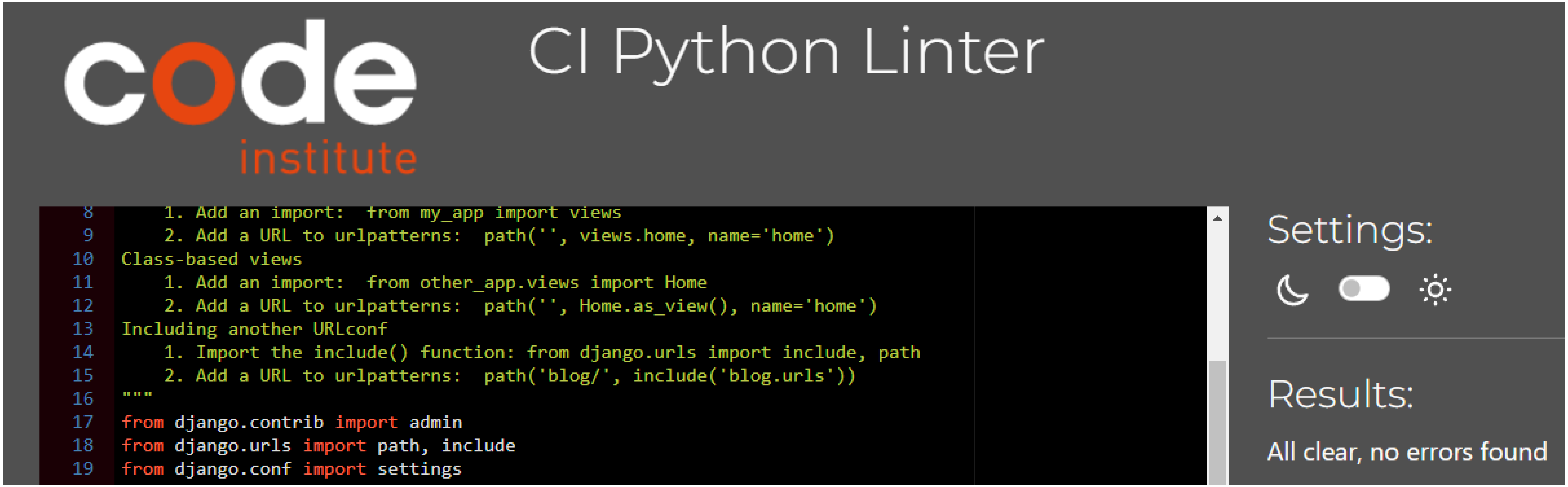


```
"""
Django settings for natures_elixir project.

Generated by 'django-admin startproject' using Django 5.1.1.

For more information on this file, see
https://docs.djangoproject.com/en/5.1/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/5.1/ref/settings/
"""
```

Results:

All clear, no errors found

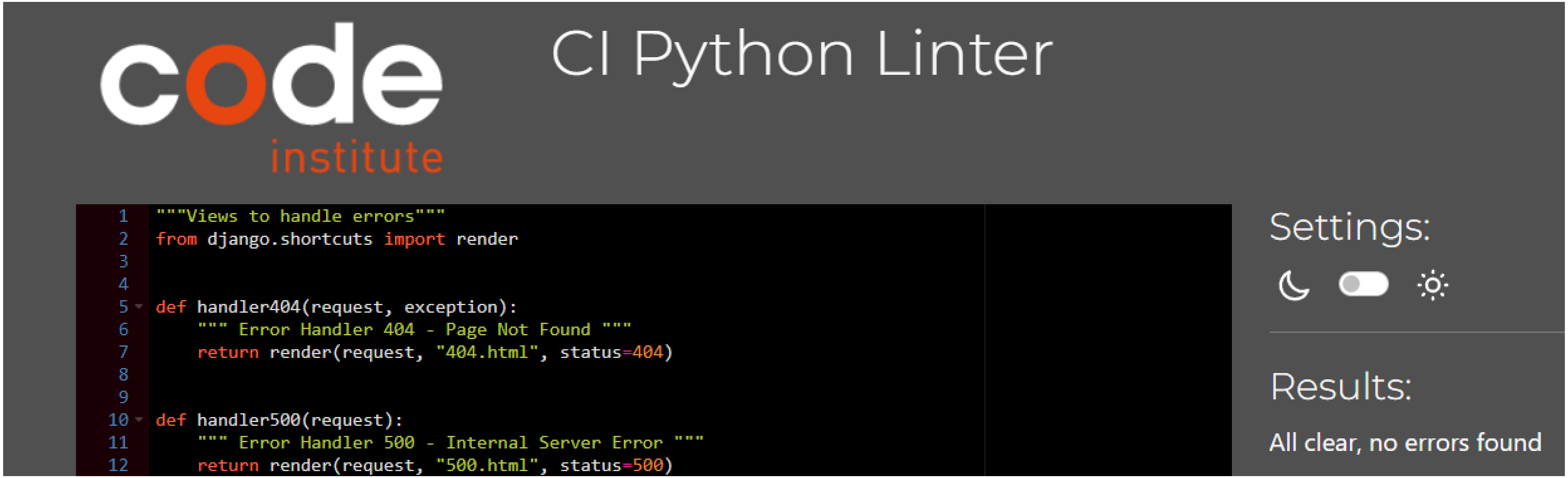
urls.py



```
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include
from django.conf import settings
```

All clear, no errors found

views.py

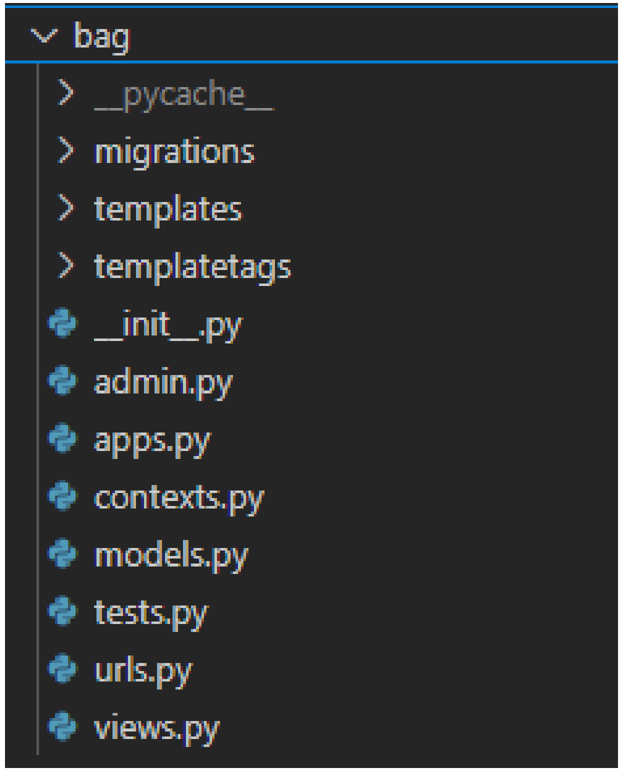


```python
"""Views to handle errors"""
from django.shortcuts import render


def handler404(request, exception):
    """ Error Handler 404 - Page Not Found """
    return render(request, "404.html", status=404)


def handler500(request):
    """ Error Handler 500 - Internal Server Error """
    return render(request, "500.html", status=500)
```

Results:

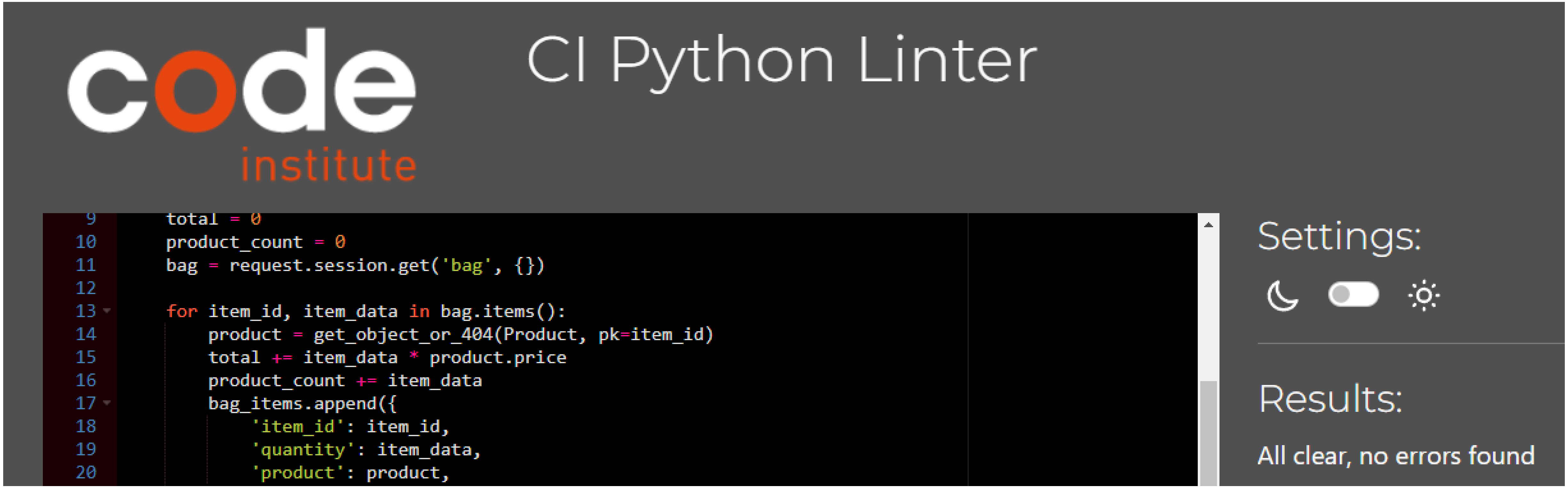
All clear, no errors found

bag app



```
bag
  __pycache__
  migrations
  templates
  templatetags
  __init__.py
  admin.py
  apps.py
  contexts.py
  models.py
  tests.py
  urls.py
  views.py
```

contexts.py



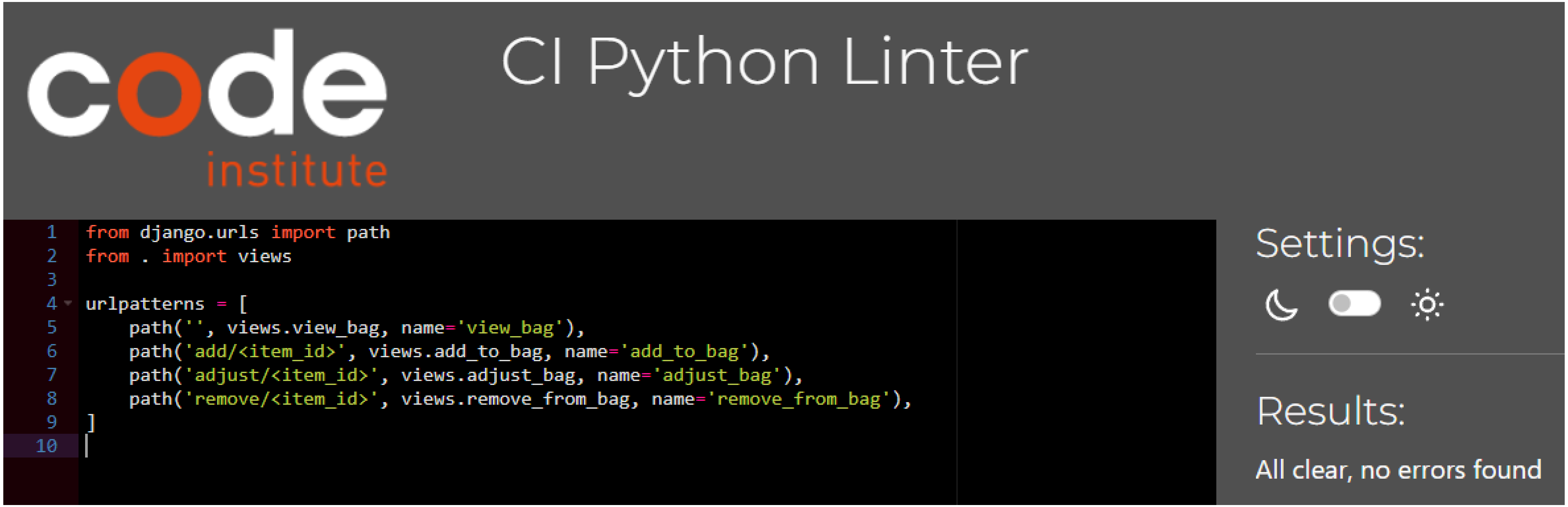
```
    total = 0
    product_count = 0
    bag = request.session.get('bag', {})

    for item_id, item_data in bag.items():
        product = get_object_or_404(Product, pk=item_id)
        total += item_data * product.price
        product_count += item_data
        bag_items.append({
            'item_id': item_id,
            'quantity': item_data,
            'product': product,
```

Settings:

Results:

All clear, no errors found

urls.py

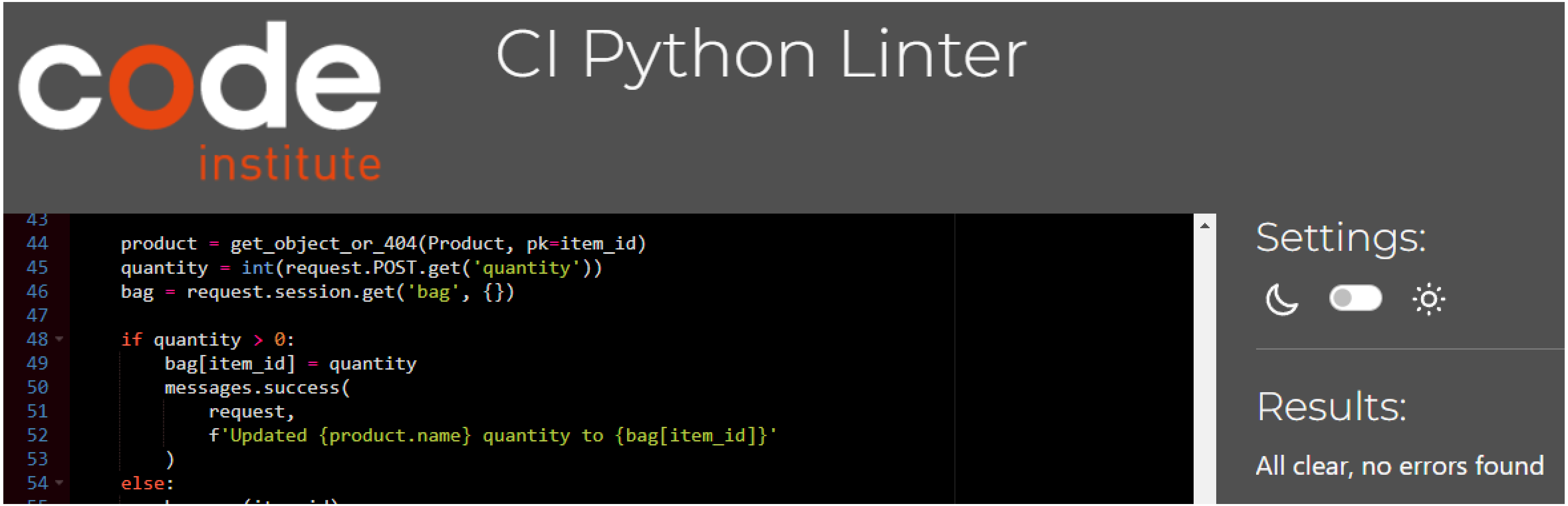


```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.view_bag, name='view_bag'),
    path('add/<item_id>', views.add_to_bag, name='add_to_bag'),
    path('adjust/<item_id>', views.adjust_bag, name='adjust_bag'),
    path('remove/<item_id>', views.remove_from_bag, name='remove_from_bag'),
]
```

Results:

All clear, no errors found

views.py



CI Python Linter

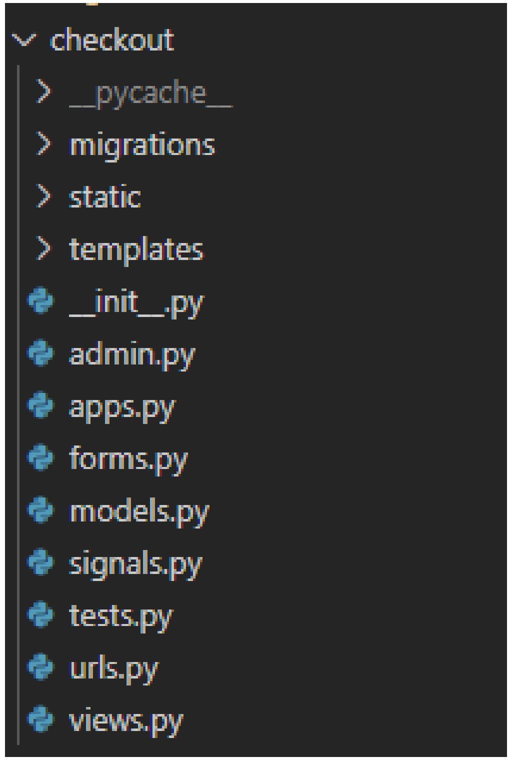
```
    product = get_object_or_404(Product, pk=item_id)
    quantity = int(request.POST.get('quantity'))
    bag = request.session.get('bag', {})

    if quantity > 0:
        bag[item_id] = quantity
        messages.success(
            request,
            f'Updated {product.name} quantity to {bag[item_id]}'
        )
    else:
```

Settings:

Results:

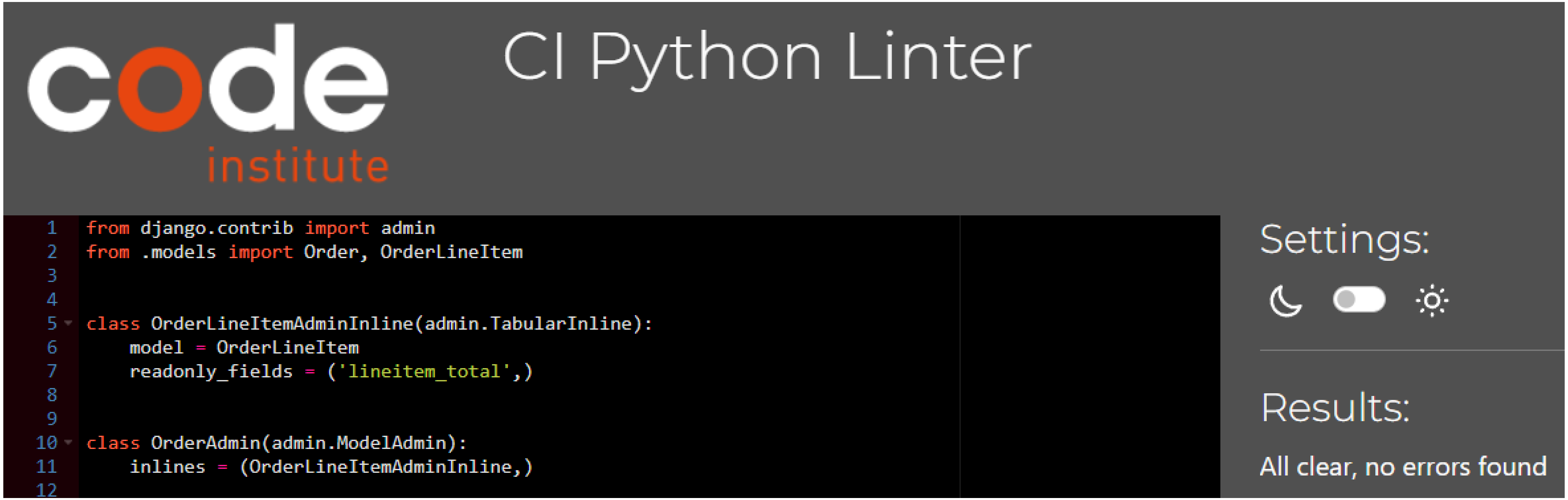
All clear, no errors found

checkout app



```
checkout
  __pycache__
  migrations
  static
  templates
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  signals.py
  tests.py
  urls.py
  views.py
```

admin.py

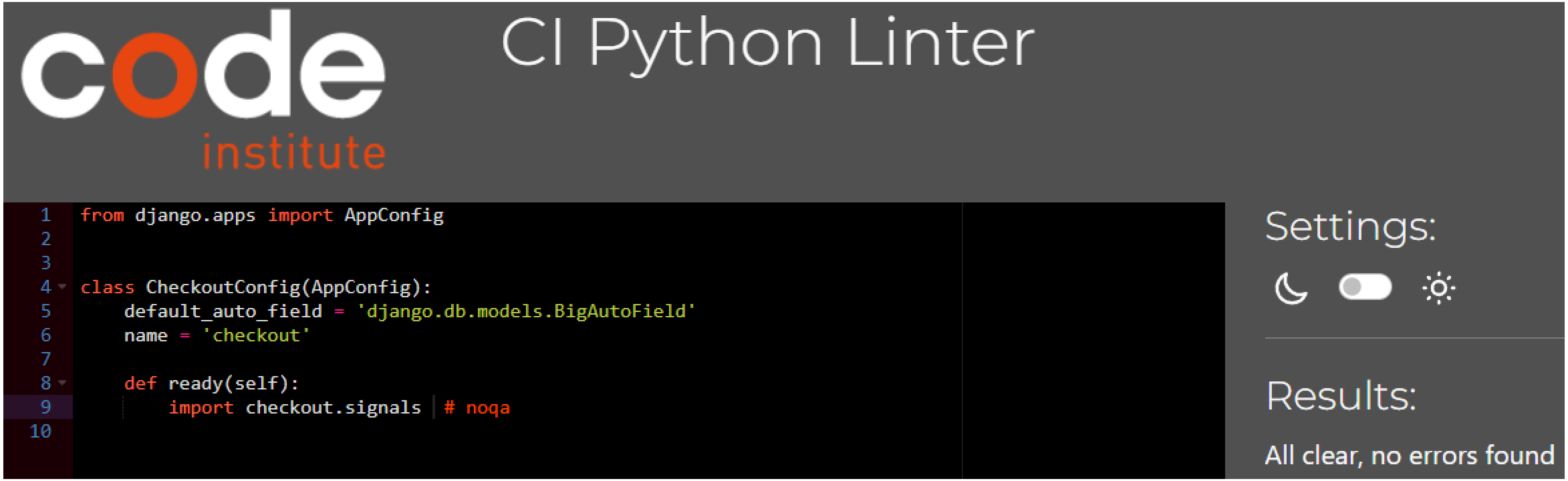


```python
from django.contrib import admin
from .models import Order, OrderLineItem


class OrderLineItemAdminInline(admin.TabularInline):
    model = OrderLineItem
    readonly_fields = ('lineitem_total',)


class OrderAdmin(admin.ModelAdmin):
    inlines = (OrderLineItemAdminInline,)

```

Results:

All clear, no errors found

apps.py

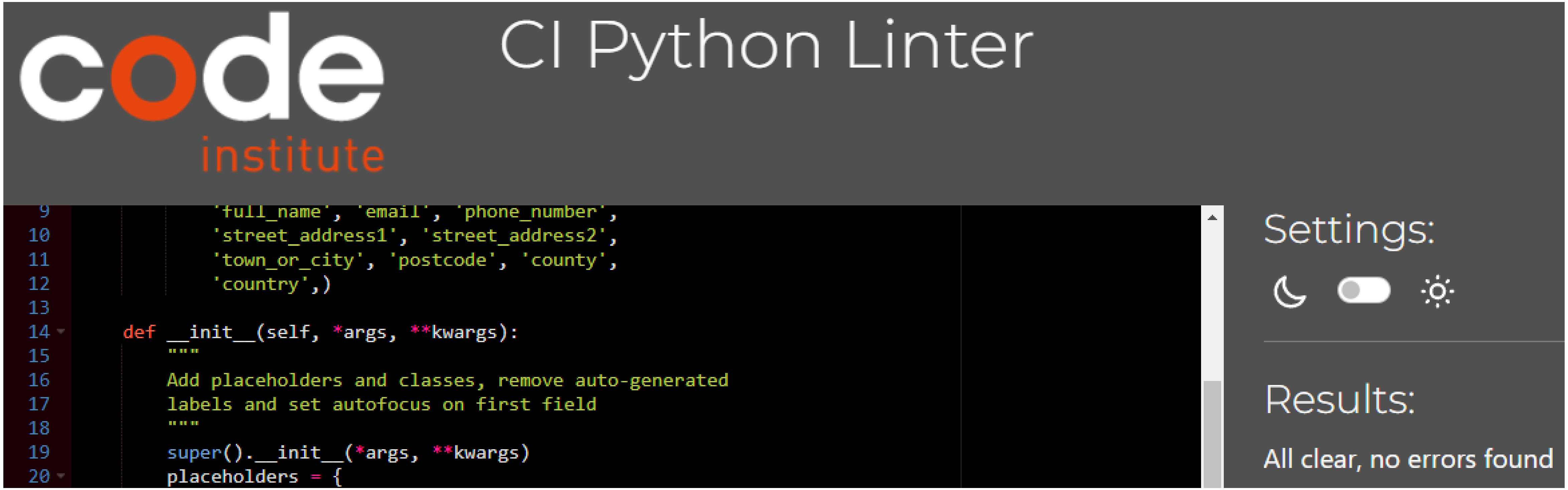


```python
from django.apps import AppConfig


class CheckoutConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'checkout'

    def ready(self):
        import checkout.signals  # noqa

```

Results:

All clear, no errors found

forms.py



CI Python Linter

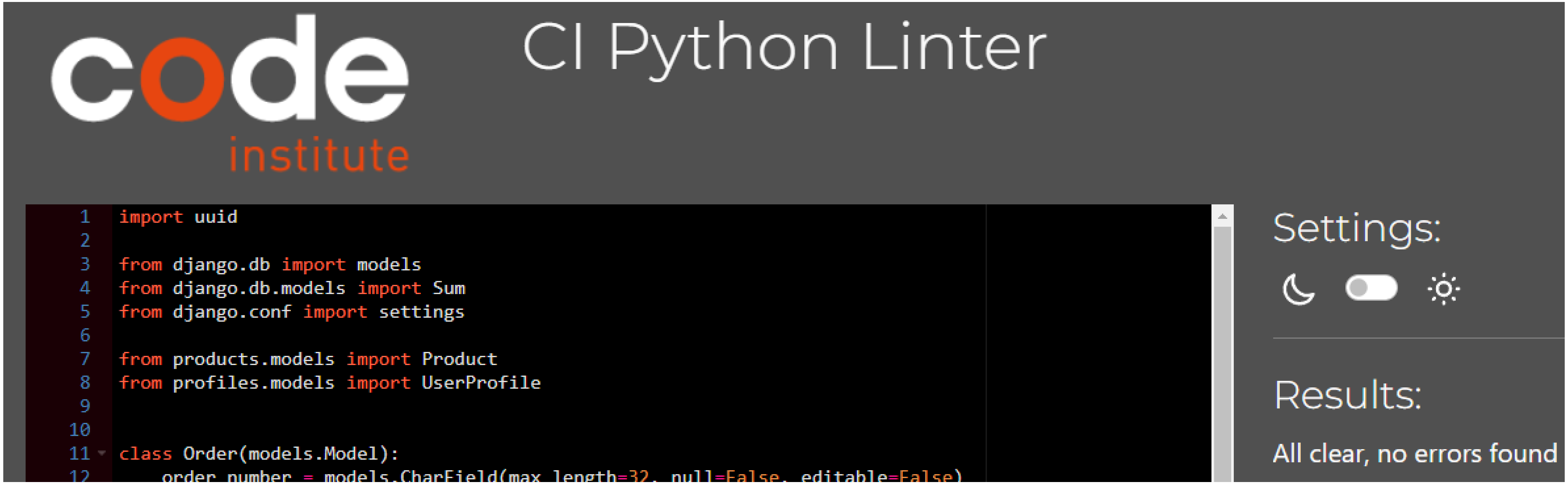
```
            'full_name', 'email', 'phone_number',
            'street_address1', 'street_address2',
            'town_or_city', 'postcode', 'county',
            'country',)

    def __init__(self, *args, **kwargs):
        """
        Add placeholders and classes, remove auto-generated
        labels and set autofocus on first field
        """
        super().__init__(*args, **kwargs)
        placeholders = {
```

Settings:

Results:

All clear, no errors found

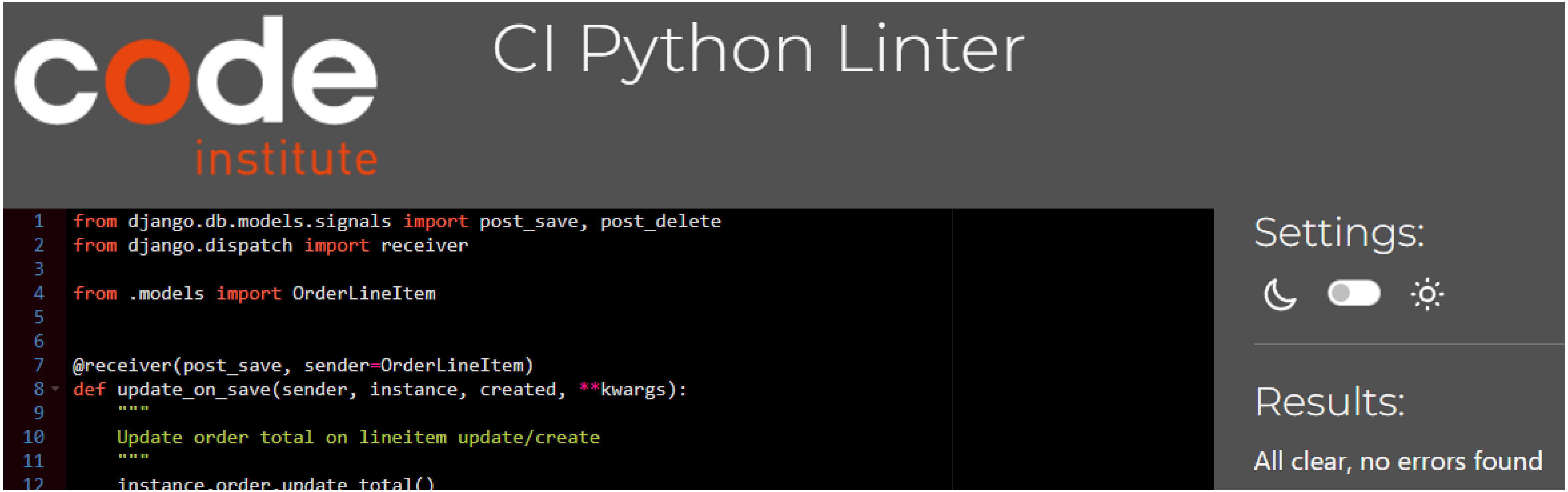
models.py



```python
import uuid

from django.db import models
from django.db.models import Sum
from django.conf import settings

from products.models import Product
from profiles.models import UserProfile


class Order(models.Model):
    order_number = models.CharField(max_length=32, null=False, editable=False)
```

All clear, no errors found

signals.py

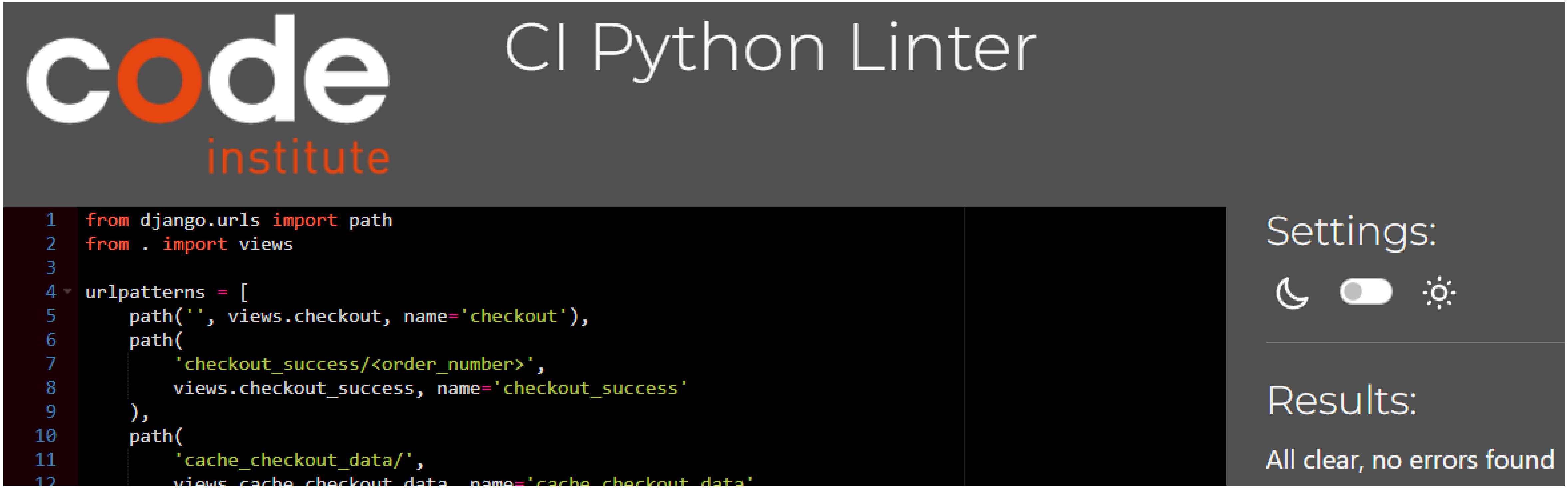


```
from django.db.models.signals import post_save, post_delete
from django.dispatch import receiver

from .models import OrderLineItem


@receiver(post_save, sender=OrderLineItem)
def update_on_save(sender, instance, created, **kwargs):
    """
    Update order total on lineitem update/create
    """
    instance.order.update_total()
```

Results:

All clear, no errors found

urls.py

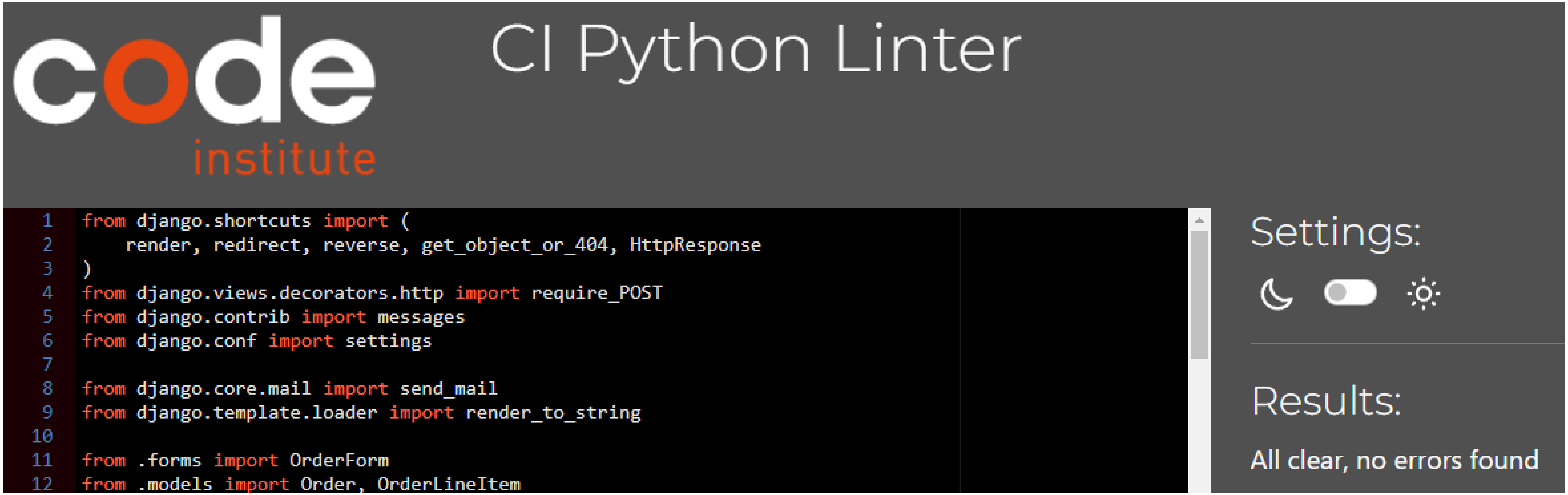


```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.checkout, name='checkout'),
    path(
        'checkout_success/<order_number>',
        views.checkout_success, name='checkout_success'
    ),
    path(
        'cache_checkout_data/',
        views.cache_checkout_data, name='cache_checkout_data'
```

Results:

All clear, no errors found

views.py

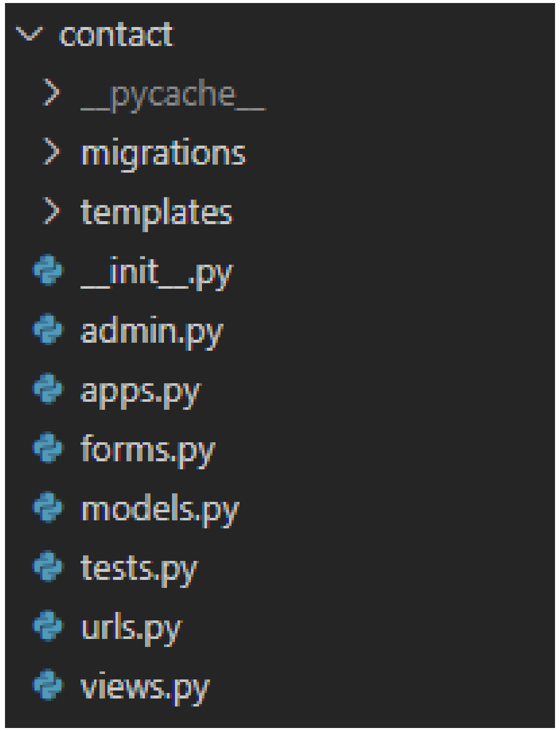


```
from django.shortcuts import (
    render, redirect, reverse, get_object_or_404, HttpResponse
)
from django.views.decorators.http import require_POST
from django.contrib import messages
from django.conf import settings

from django.core.mail import send_mail
from django.template.loader import render_to_string

from .forms import OrderForm
from .models import Order, OrderLineItem
```

Results:

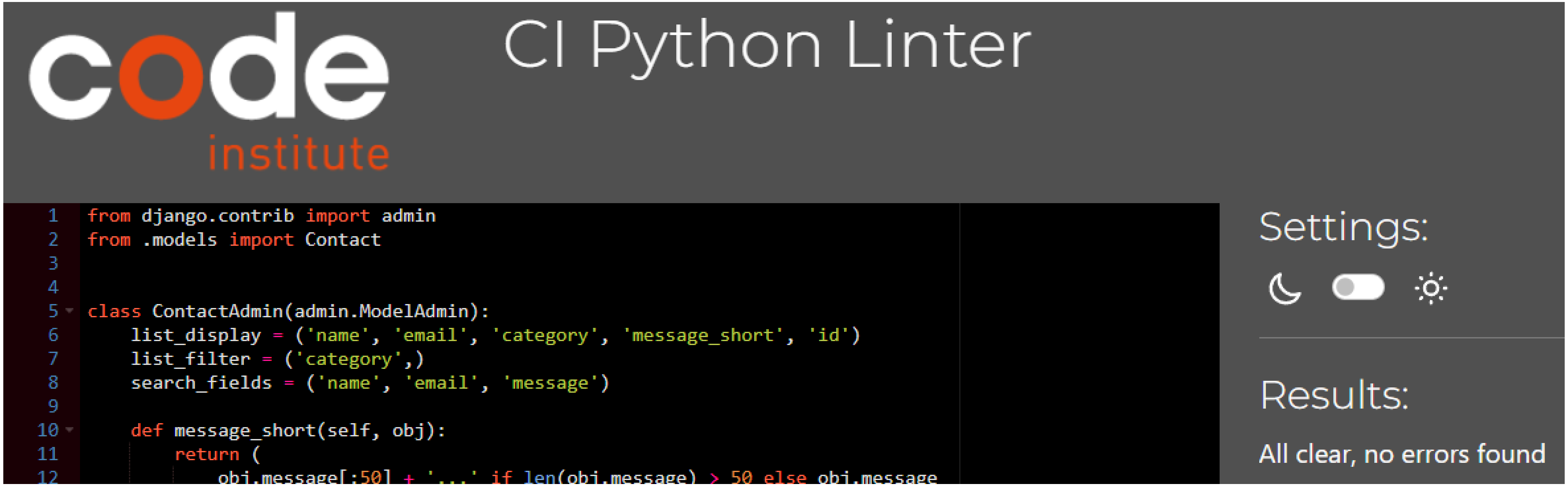
All clear, no errors found

contact app



```
contact
  __pycache__
  migrations
  templates
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  tests.py
  urls.py
  views.py
```

admin.py



CI Python Linter

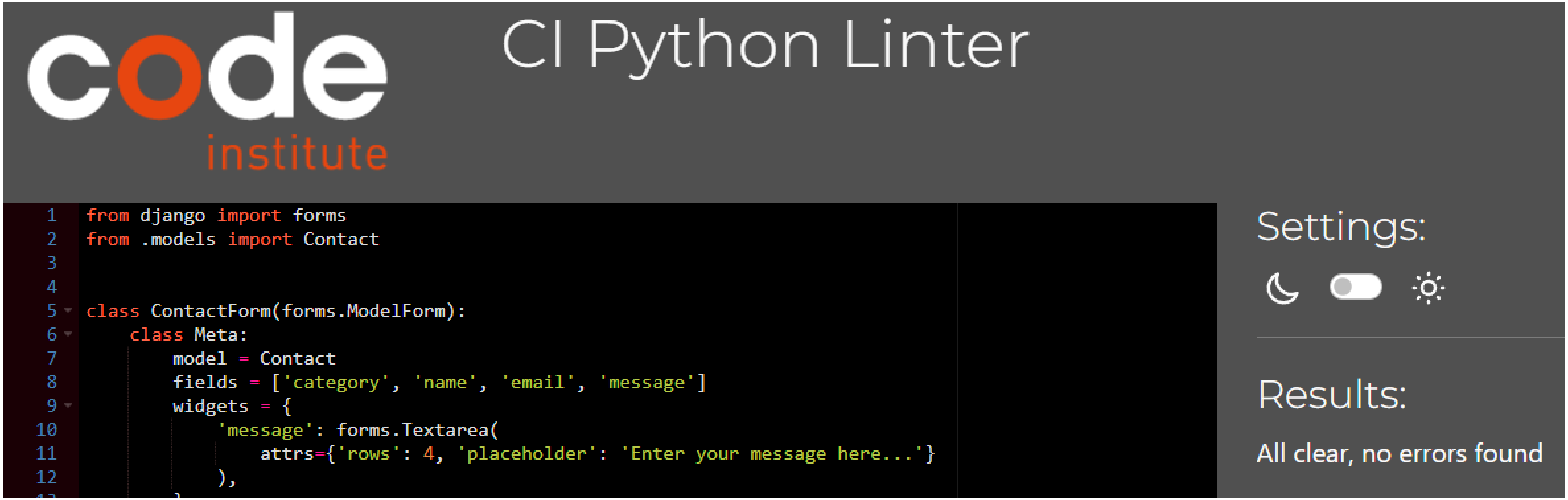
```
from django.contrib import admin
from .models import Contact


class ContactAdmin(admin.ModelAdmin):
    list_display = ('name', 'email', 'category', 'message_short', 'id')
    list_filter = ('category',)
    search_fields = ('name', 'email', 'message')

    def message_short(self, obj):
        return (
            obj.message[:50] + '...' if len(obj.message) > 50 else obj.message
```

Settings:

Results:

All clear, no errors found

forms.py

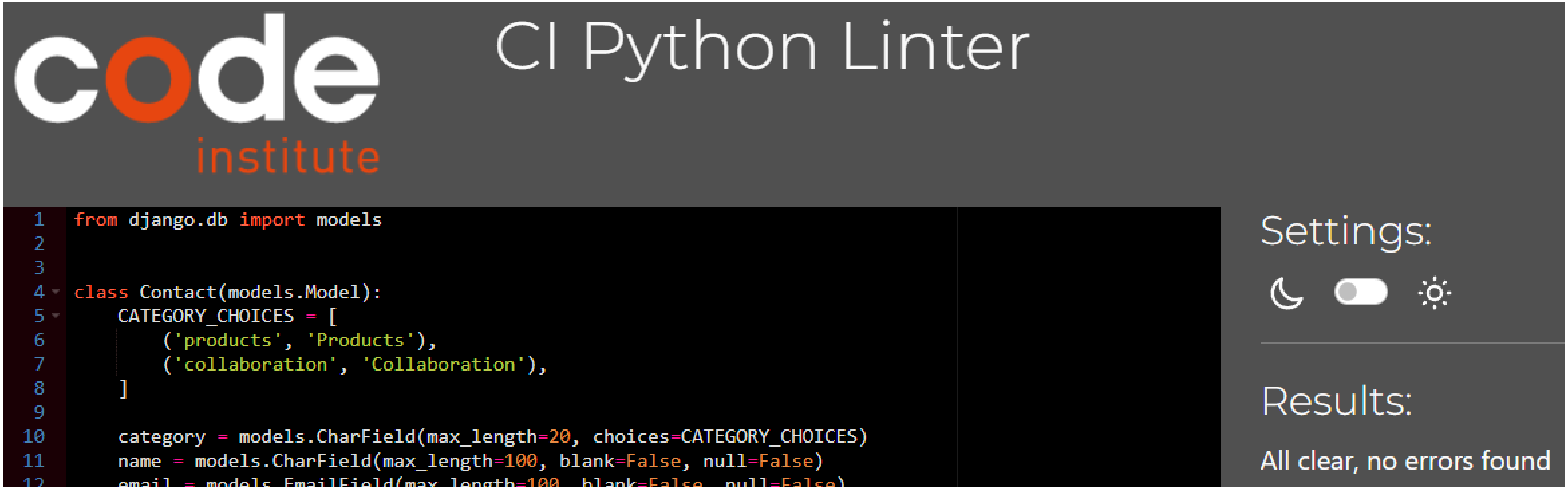


```
from django import forms
from .models import Contact


class ContactForm(forms.ModelForm):
    class Meta:
        model = Contact
        fields = ['category', 'name', 'email', 'message']
        widgets = {
            'message': forms.Textarea(
                attrs={'rows': 4, 'placeholder': 'Enter your message here...'}
            ),
        }
```

Results:

All clear, no errors found

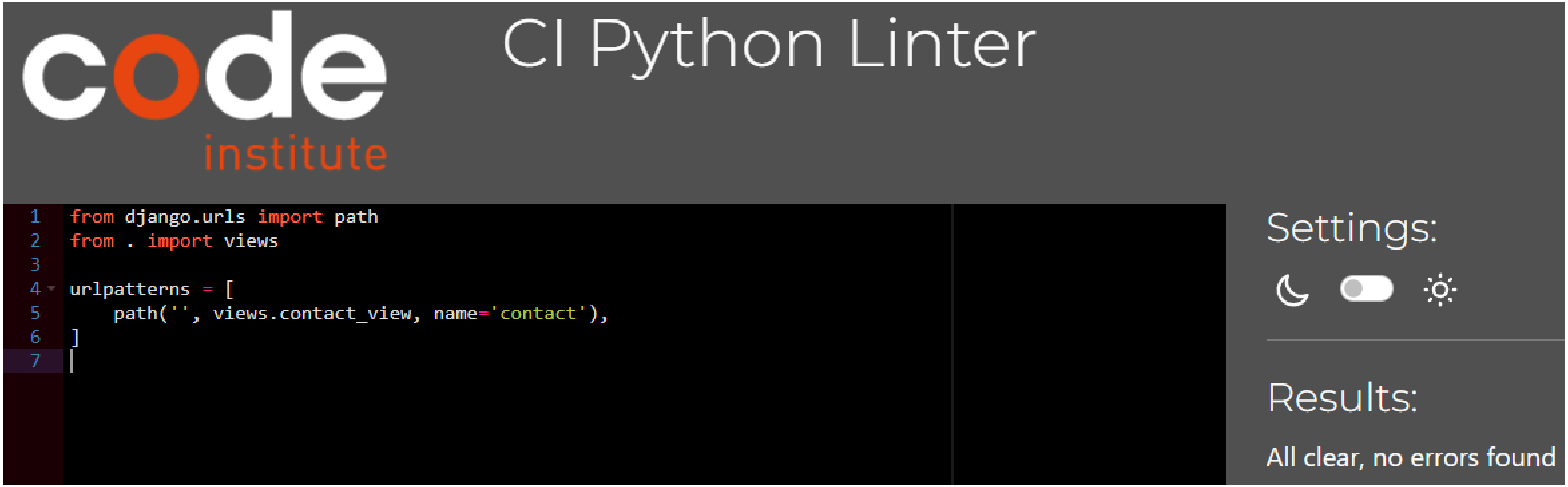
models.py



```
from django.db import models


class Contact(models.Model):
    CATEGORY_CHOICES = [
        ('products', 'Products'),
        ('collaboration', 'Collaboration'),
    ]

    category = models.CharField(max_length=20, choices=CATEGORY_CHOICES)
    name = models.CharField(max_length=100, blank=False, null=False)
    email = models.EmailField(max_length=100, blank=False, null=False)
```

All clear, no errors found

urls.py

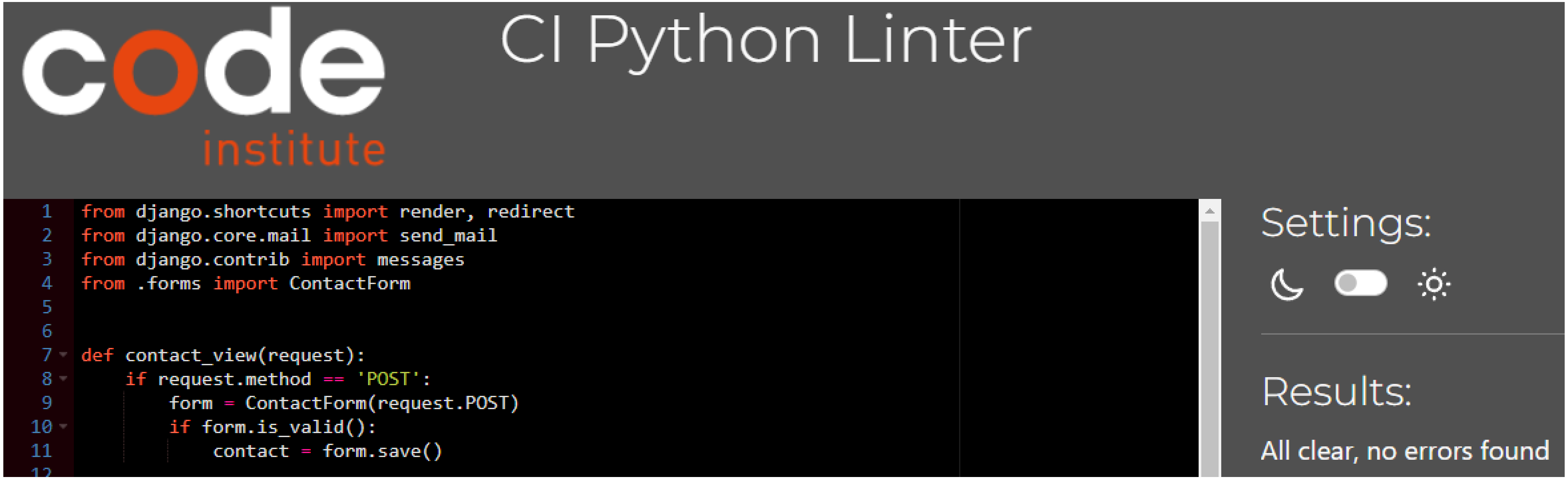


```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.contact_view, name='contact'),
]
```

Results:

All clear, no errors found

views.py



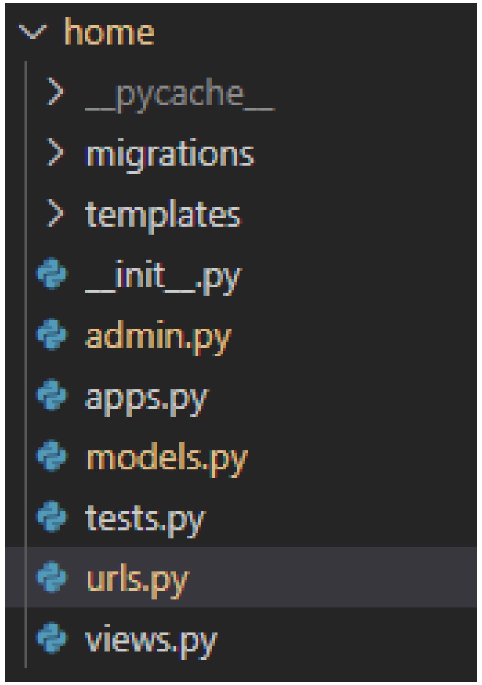
```
from django.shortcuts import render, redirect
from django.core.mail import send_mail
from django.contrib import messages
from .forms import ContactForm


def contact_view(request):
    if request.method == 'POST':
        form = ContactForm(request.POST)
        if form.is_valid():
            contact = form.save()
```

Settings:

Results:

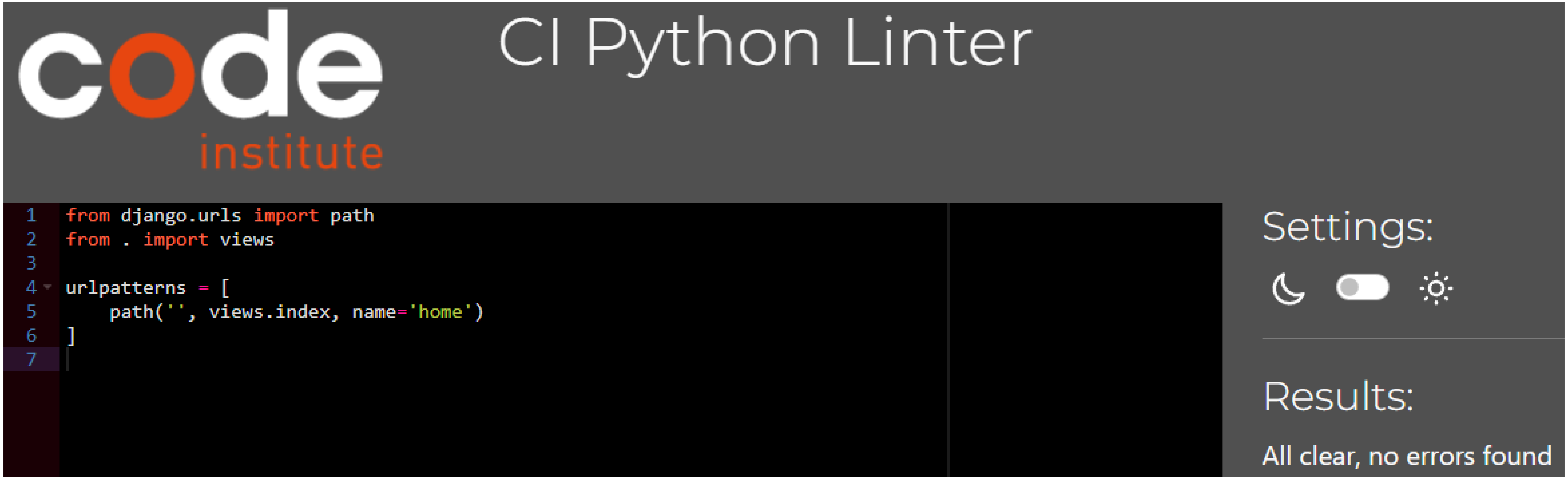
All clear, no errors found

home app



home
__pycache__
migrations
templates
__init__.py
admin.py
apps.py
models.py
tests.py
urls.py
views.py

urls.py

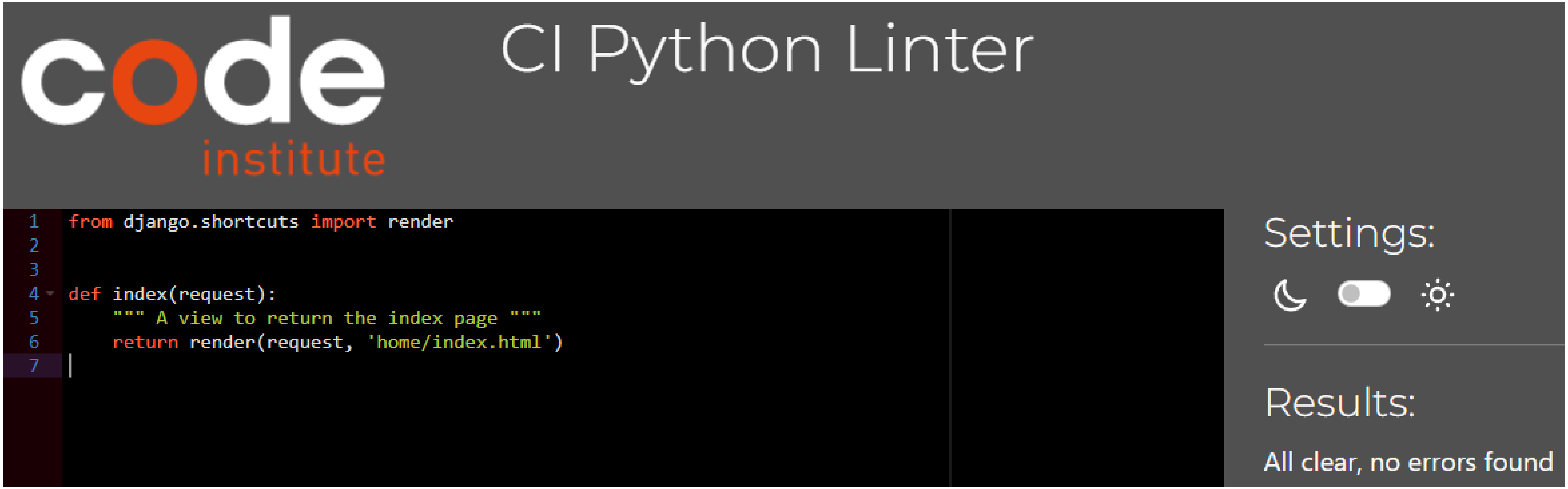


```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index, name='home')
]
```

Results:

All clear, no errors found

views.py

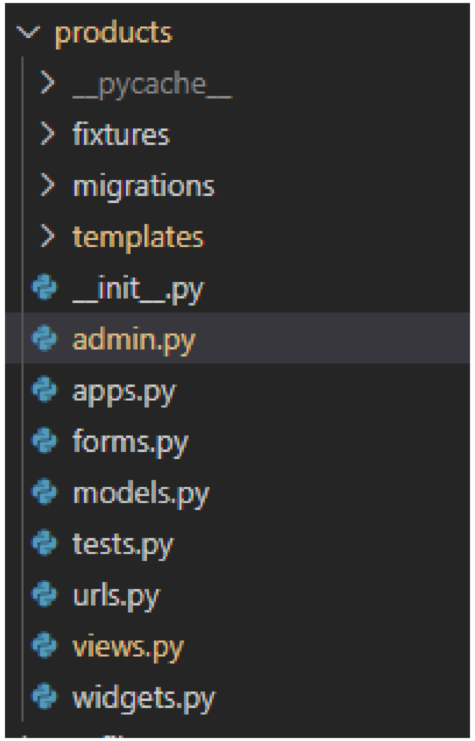


```python
from django.shortcuts import render


def index(request):
    """ A view to return the index page """
    return render(request, 'home/index.html')

```

Results:

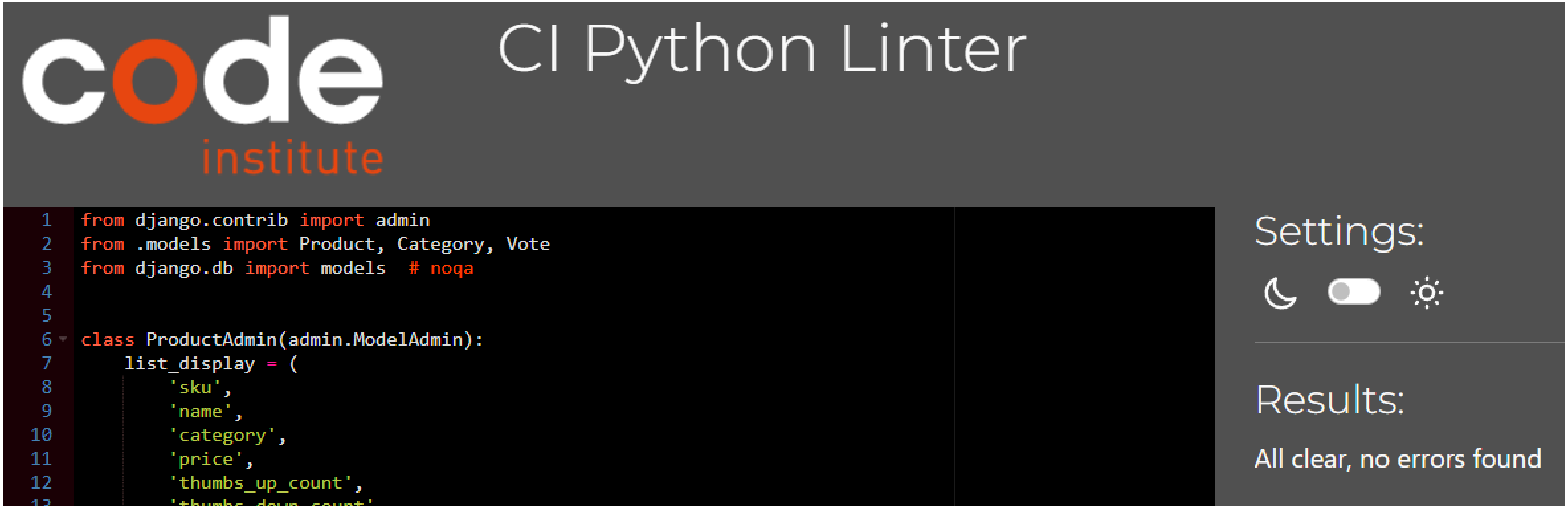
All clear, no errors found

products app



```
v products
  > __pycache__
  > fixtures
  > migrations
  > templates
    __init__.py
    admin.py
    apps.py
    forms.py
    models.py
    tests.py
    urls.py
    views.py
    widgets.py
```

admin.py



CI Python Linter

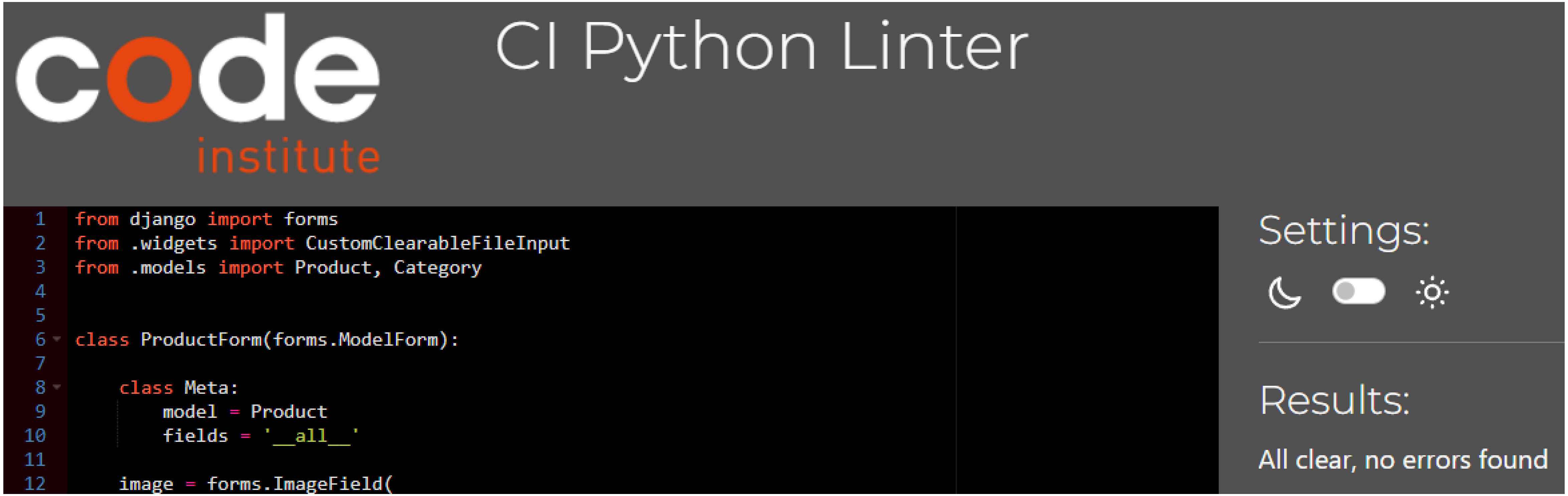
```
from django.contrib import admin
from .models import Product, Category, Vote
from django.db import models  # noqa


class ProductAdmin(admin.ModelAdmin):
    list_display = (
        'sku',
        'name',
        'category',
        'price',
        'thumbs_up_count',
        'thumbs_down_count'
```

Settings:

Results:

All clear, no errors found

forms.py

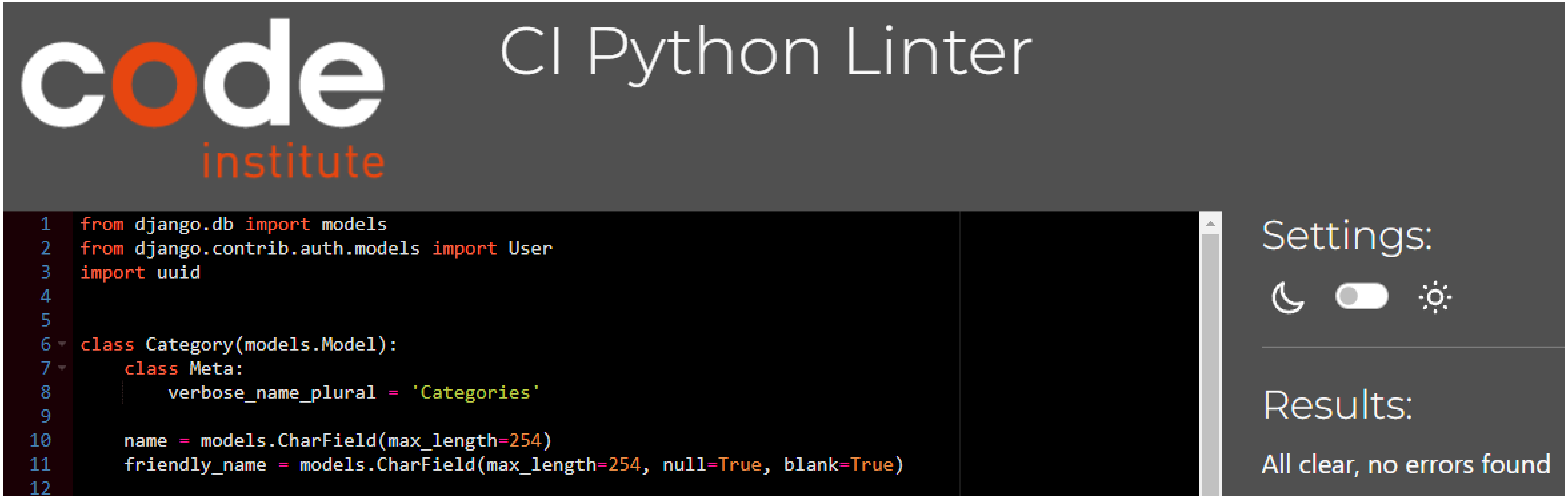


```
from django import forms
from .widgets import CustomClearableFileInput
from .models import Product, Category


class ProductForm(forms.ModelForm):

    class Meta:
        model = Product
        fields = '__all__'

    image = forms.ImageField(
```

Results:

All clear, no errors found

models.py

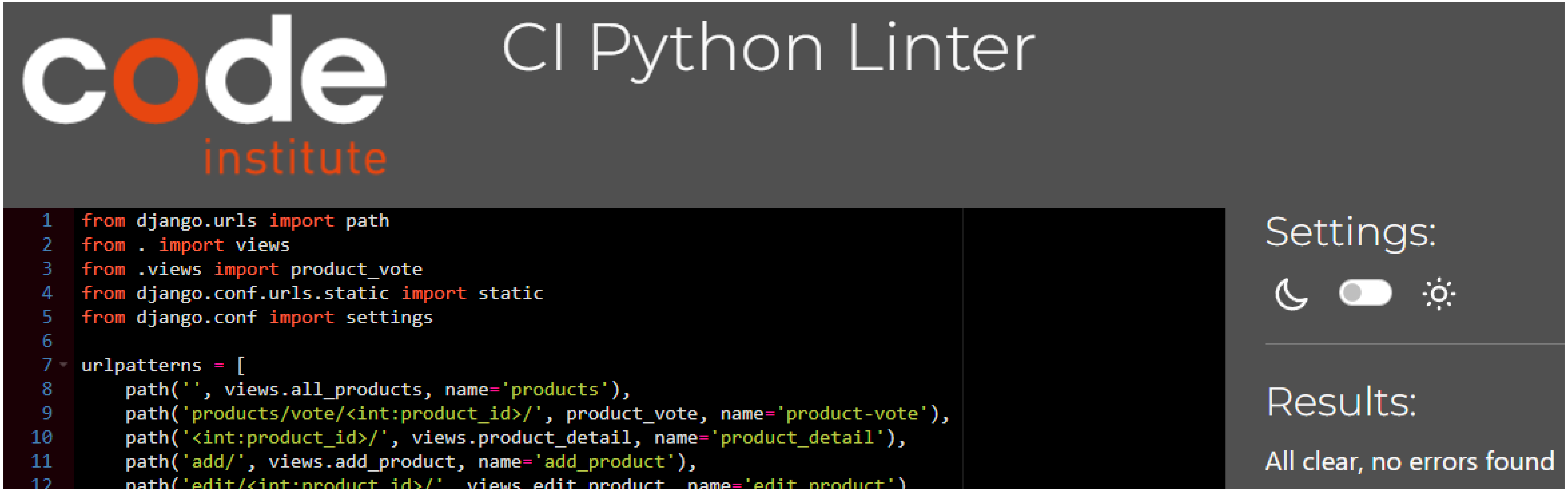


```
from django.db import models
from django.contrib.auth.models import User
import uuid


class Category(models.Model):
    class Meta:
        verbose_name_plural = 'Categories'

    name = models.CharField(max_length=254)
    friendly_name = models.CharField(max_length=254, null=True, blank=True)

```

Results:

All clear, no errors found

urls.py



CI Python Linter

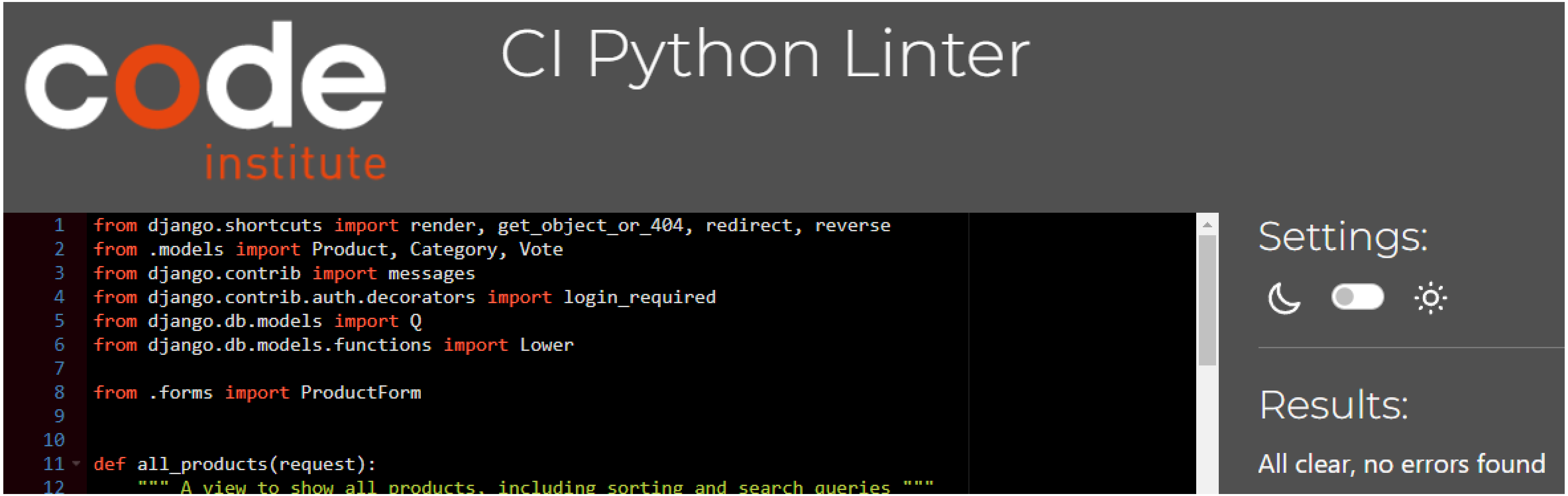
```
from django.urls import path
from . import views
from .views import product_vote
from django.conf.urls.static import static
from django.conf import settings

urlpatterns = [
    path('', views.all_products, name='products'),
    path('products/vote/<int:product_id>/', product_vote, name='product-vote'),
    path('<int:product_id>/', views.product_detail, name='product_detail'),
    path('add/', views.add_product, name='add_product'),
    path('edit/<int:product_id>/', views.edit_product, name='edit_product'),
```

Settings:

Results:

All clear, no errors found

views.py



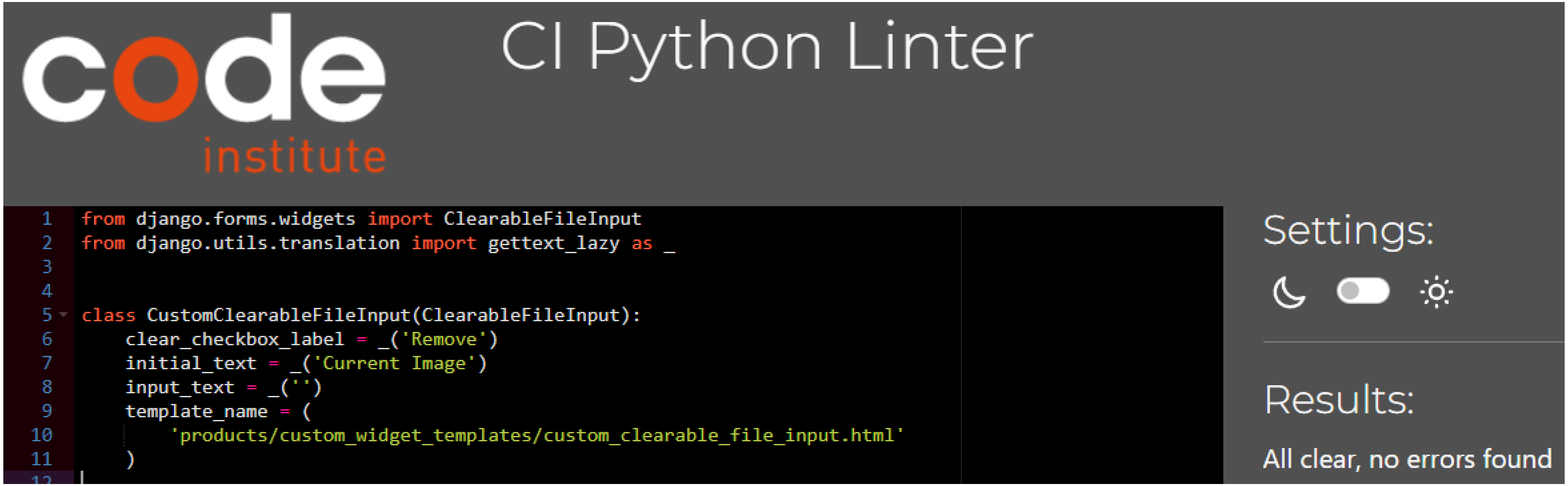
```
from django.shortcuts import render, get_object_or_404, redirect, reverse
from .models import Product, Category, Vote
from django.contrib import messages
from django.contrib.auth.decorators import login_required
from django.db.models import Q
from django.db.models.functions import Lower

from .forms import ProductForm


def all_products(request):
    """ A view to show all products, including sorting and search queries """
```

Settings:

Results:

All clear, no errors found

widgets.py

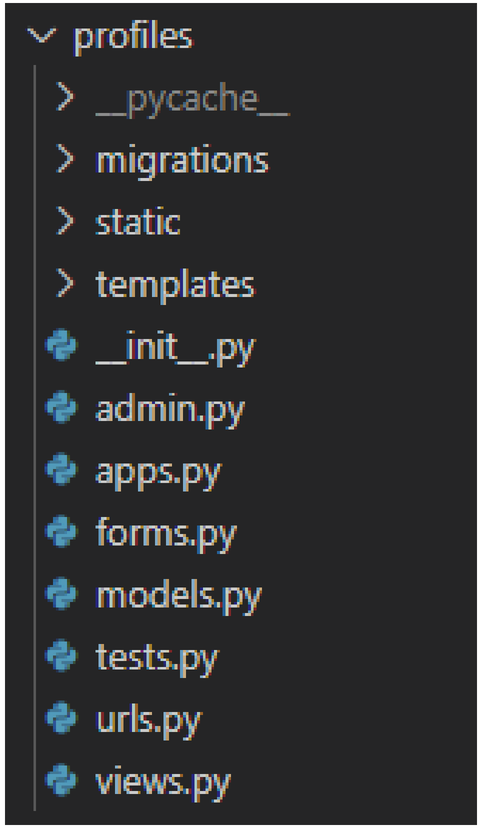


```
from django.forms.widgets import ClearableFileInput
from django.utils.translation import gettext_lazy as _


class CustomClearableFileInput(ClearableFileInput):
    clear_checkbox_label = _('Remove')
    initial_text = _('Current Image')
    input_text = _('')
    template_name = (
        'products/custom_widget_templates/custom_clearable_file_input.html'
    )
```

Results:

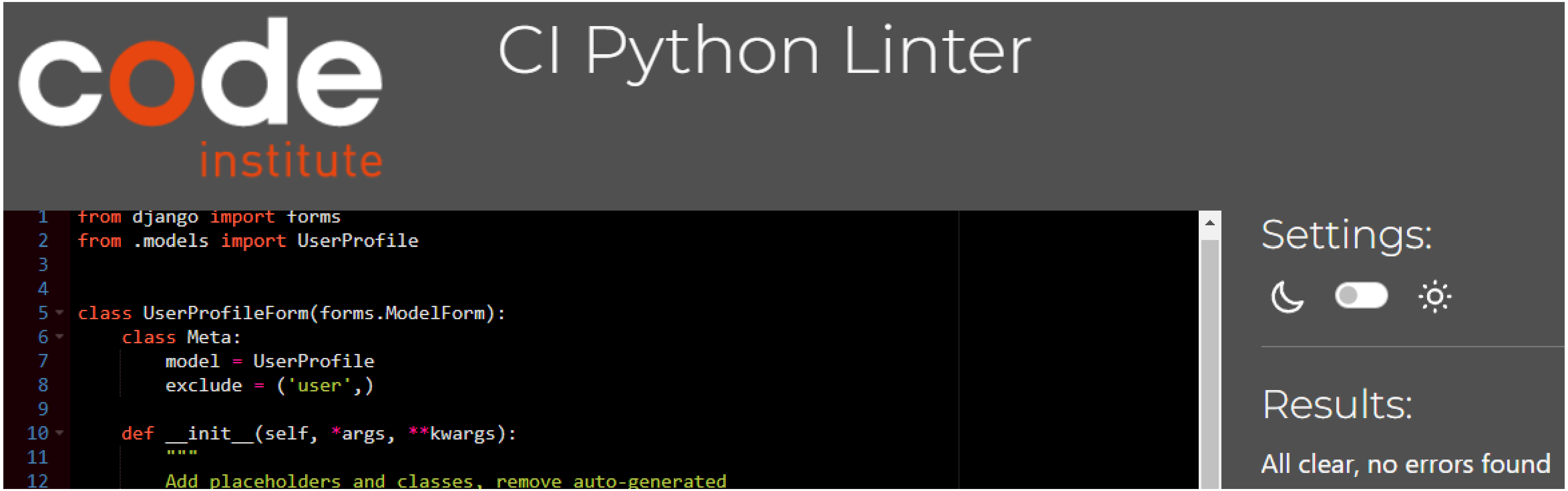
All clear, no errors found

profiles app



```
profiles
  __pycache__
  migrations
  static
  templates
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  tests.py
  urls.py
  views.py
```

forms.py



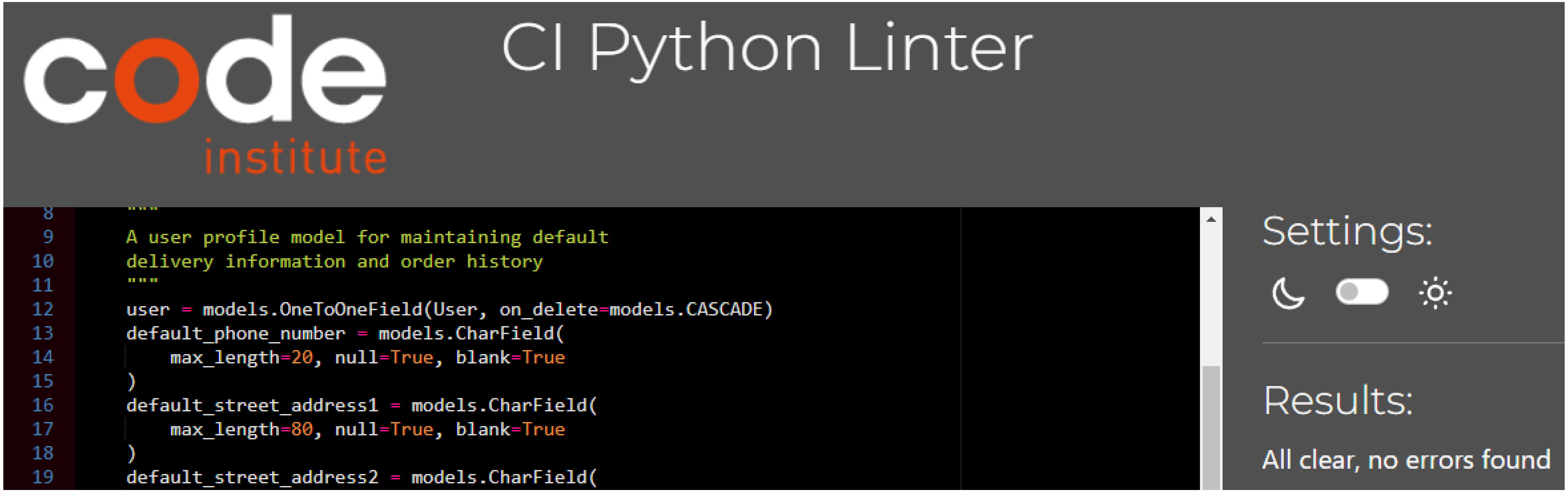
```
from django import forms
from .models import UserProfile


class UserProfileForm(forms.ModelForm):
    class Meta:
        model = UserProfile
        exclude = ('user',)

    def __init__(self, *args, **kwargs):
        """
        Add placeholders and classes, remove auto-generated
```

Settings:

Results:

All clear, no errors found

models.py



CI Python Linter

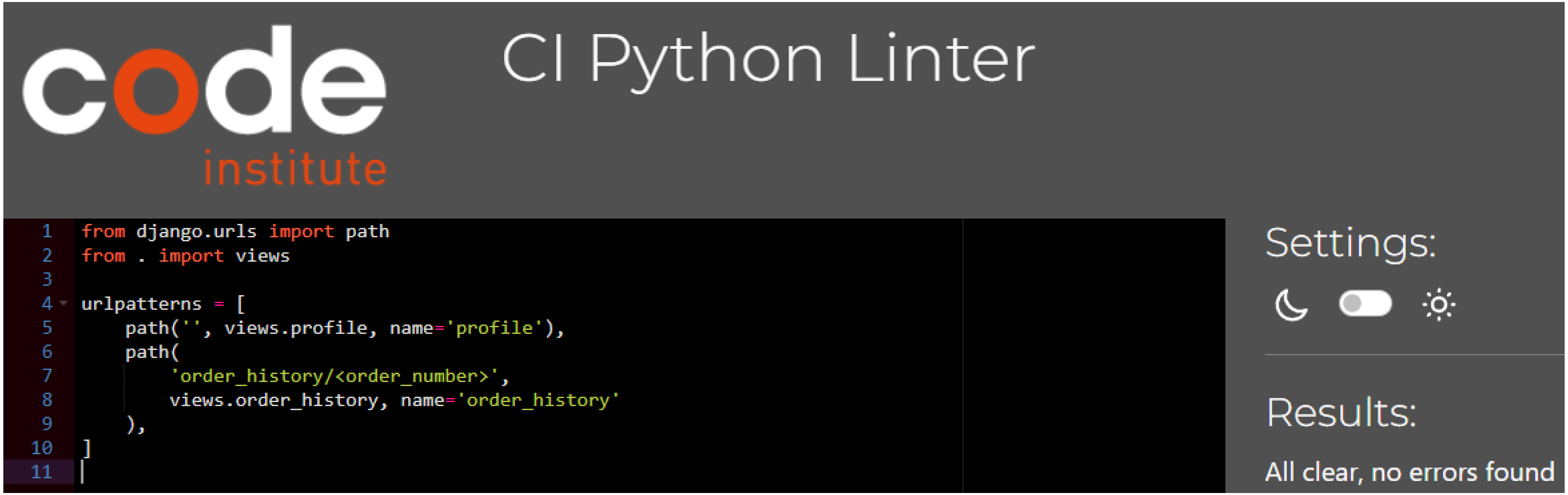
```
    """
    A user profile model for maintaining default
    delivery information and order history
    """
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    default_phone_number = models.CharField(
        max_length=20, null=True, blank=True
    )
    default_street_address1 = models.CharField(
        max_length=80, null=True, blank=True
    )
    default_street_address2 = models.CharField(
```

Settings:

Results:

All clear, no errors found

urls.py

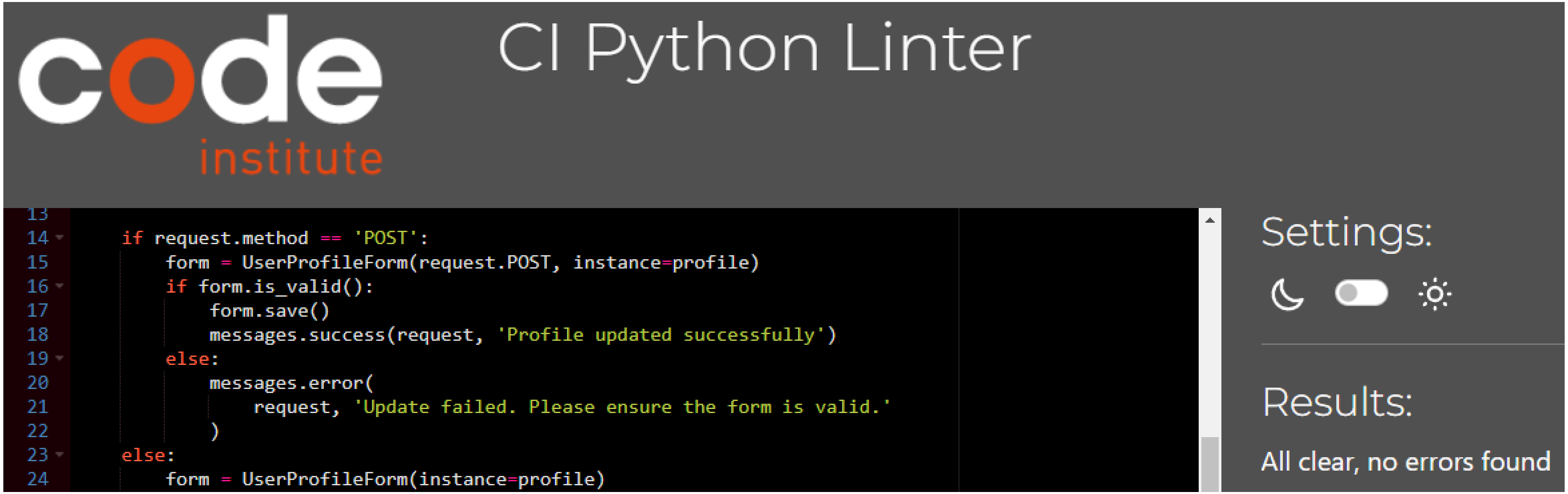


```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.profile, name='profile'),
    path(
        'order_history/<order_number>',
        views.order_history, name='order_history'
    ),
]
```

Results:

All clear, no errors found

views.py



CI Python Linter

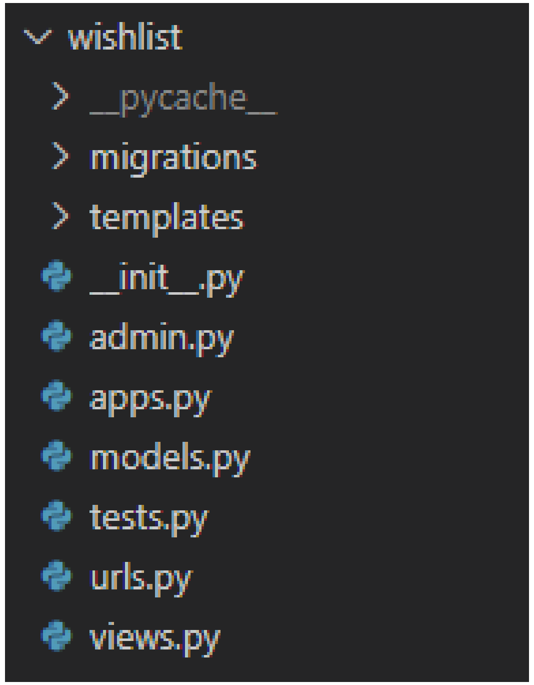
```
    if request.method == 'POST':
        form = UserProfileForm(request.POST, instance=profile)
        if form.is_valid():
            form.save()
            messages.success(request, 'Profile updated successfully')
        else:
            messages.error(
                request, 'Update failed. Please ensure the form is valid.'
            )
    else:
        form = UserProfileForm(instance=profile)
```

Settings:

Results:

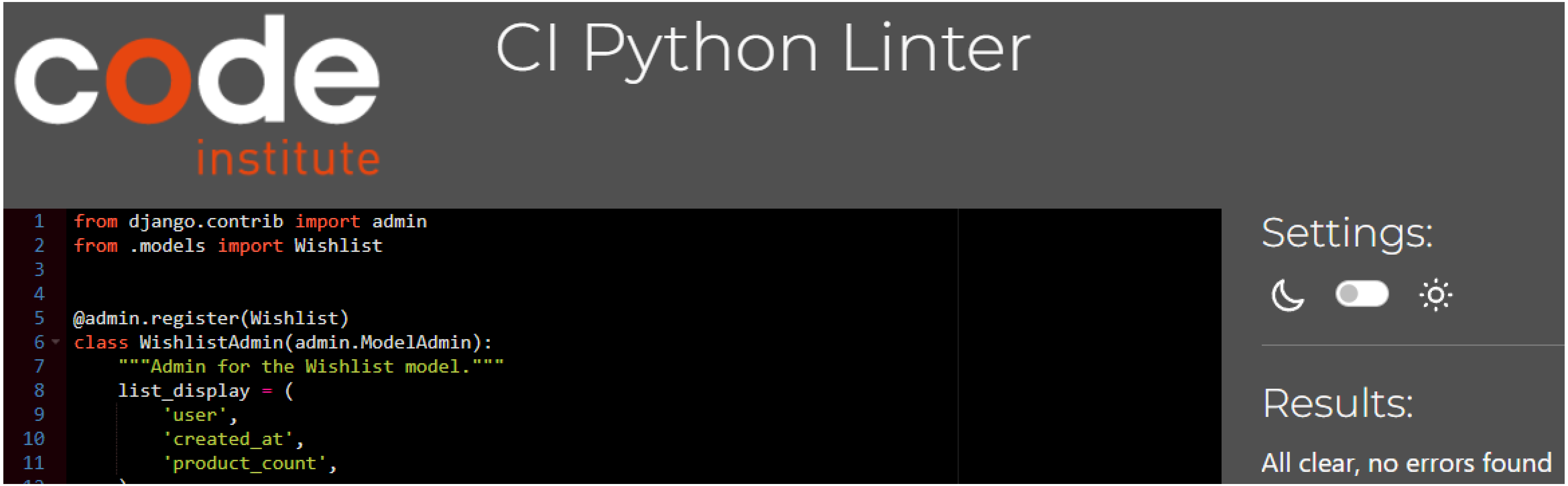
All clear, no errors found

wishlist app



```
∨ wishlist
  > __pycache__
  > migrations
  > templates
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    urls.py
    views.py
```

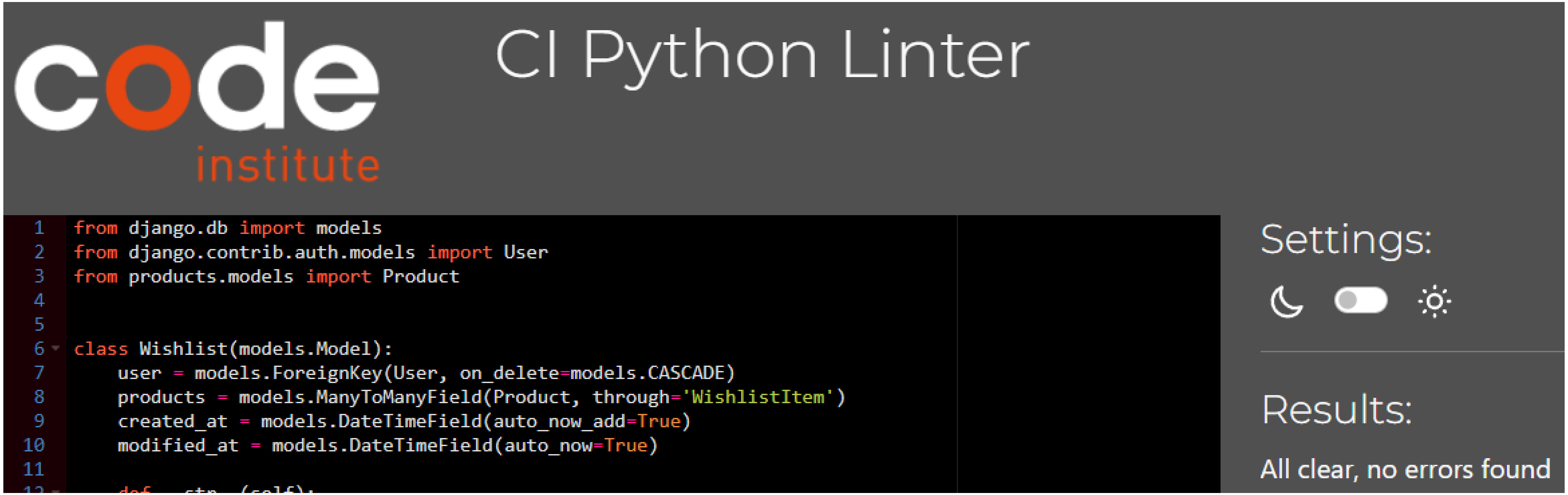
admin.py



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


@admin.register(Wishlist)
class WishlistAdmin(admin.ModelAdmin):
    """Admin for the Wishlist model."""
    list_display = (
        'user',
        'created_at',
        'product_count',
    )
```

All clear, no errors found

models.py



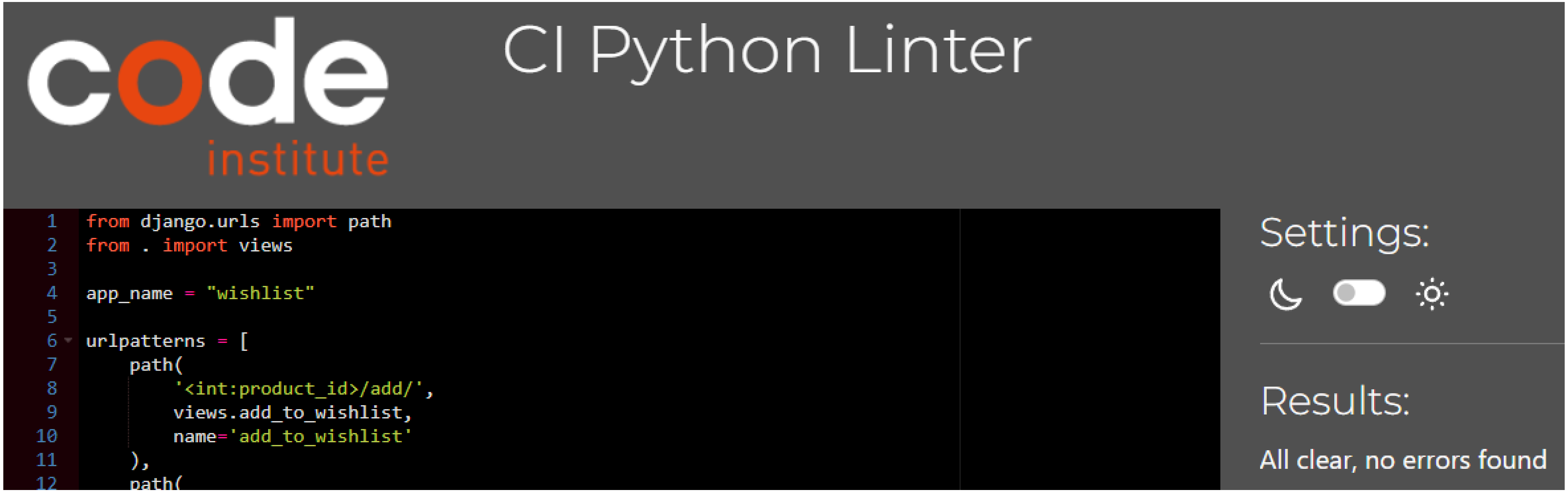
```python
from django.db import models
from django.contrib.auth.models import User
from products.models import Product


class Wishlist(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    products = models.ManyToManyField(Product, through='WishlistItem')
    created_at = models.DateTimeField(auto_now_add=True)
    modified_at = models.DateTimeField(auto_now=True)

    def __str__(self):
```

Settings:

Results:

All clear, no errors found

urls.py

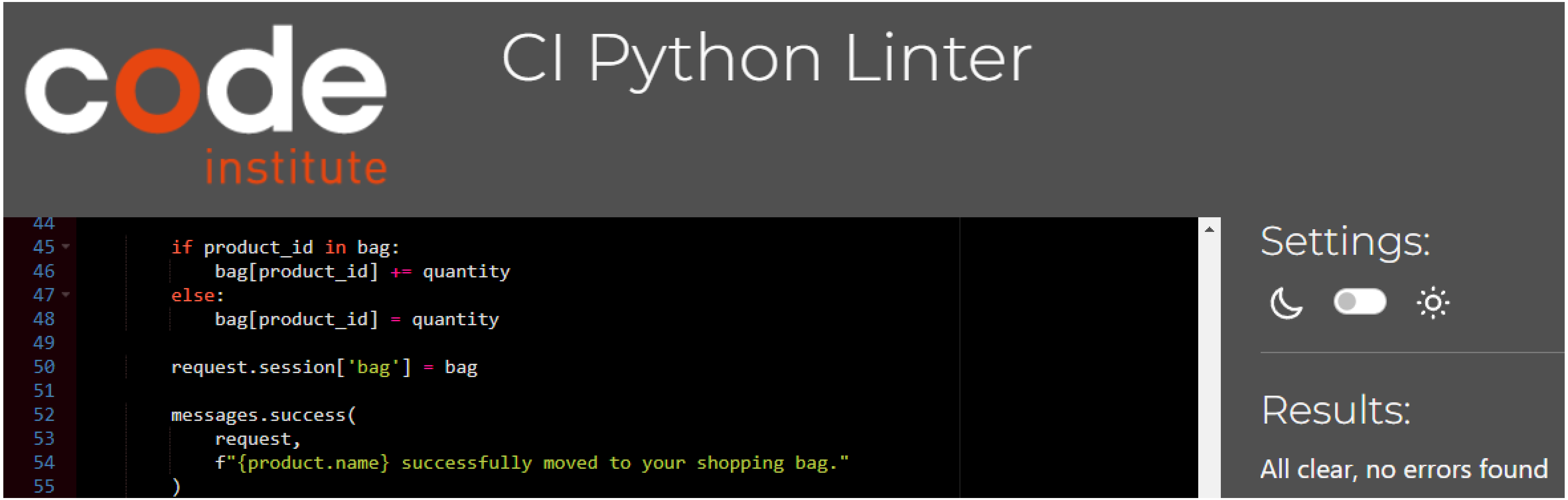


```python
from django.urls import path
from . import views

app_name = "wishlist"

urlpatterns = [
    path(
        '<int:product_id>/add/',
        views.add_to_wishlist,
        name='add_to_wishlist'
    ),
    path(
```

Results:

All clear, no errors found

views.py



```python
        if product_id in bag:
            bag[product_id] += quantity
        else:
            bag[product_id] = quantity

        request.session['bag'] = bag

        messages.success(
            request,
            f"{product.name} successfully moved to your shopping bag."
        )
```

Results:

All clear, no errors found