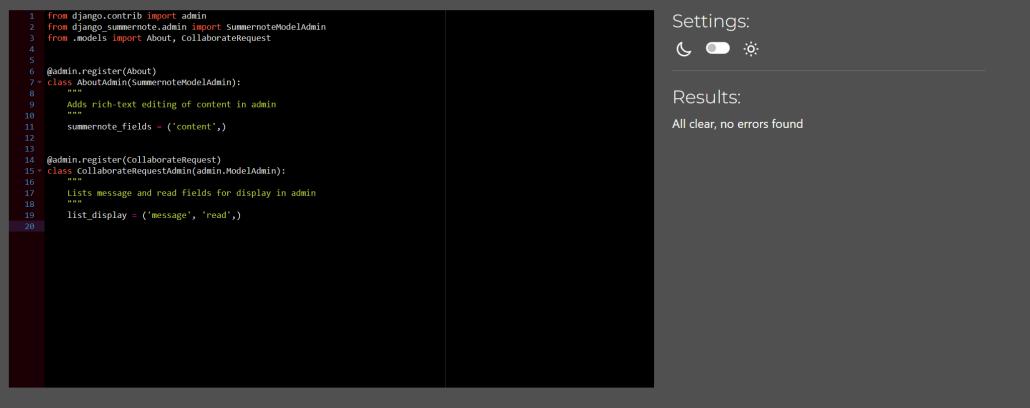
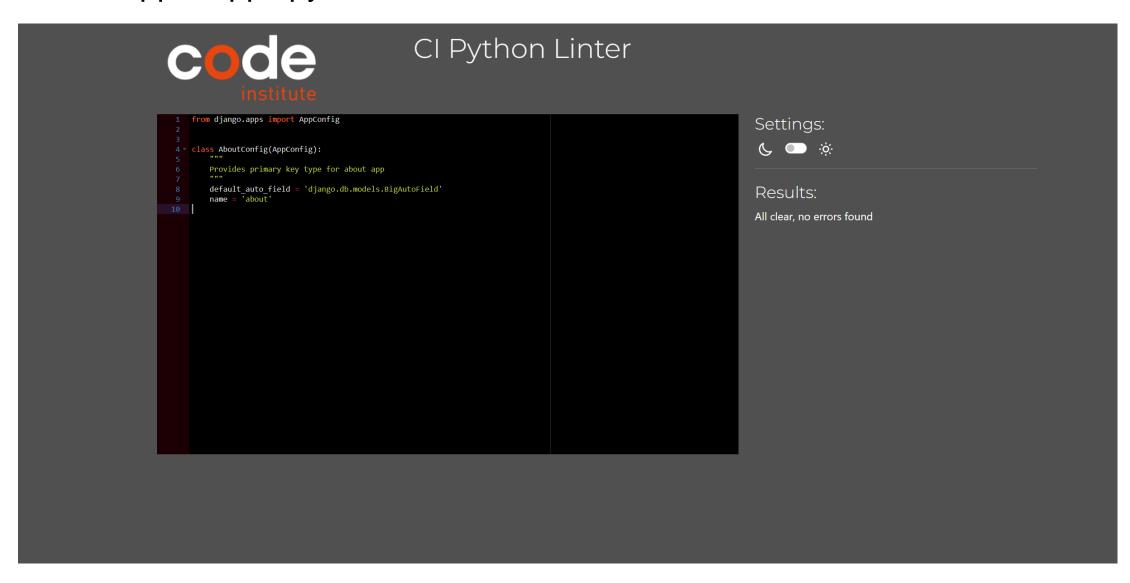
About App – admin.py



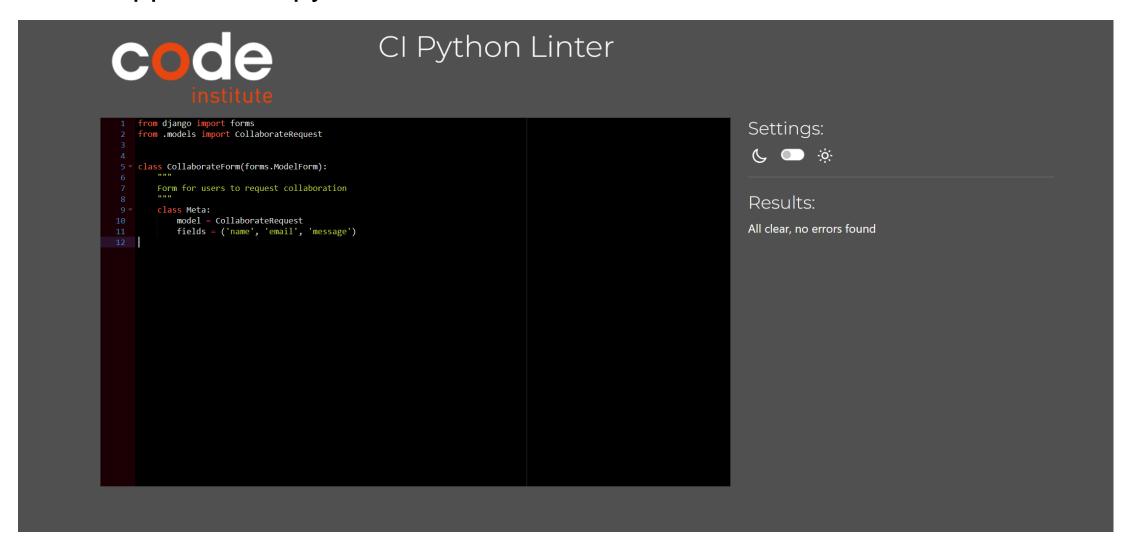
CI Python Linter



About App – apps.py



About App – forms.py



About App – models.py



CI Python Linter

```
1 from django.db import models
    from cloudinary.models import CloudinaryField
class About(models.Model):
       Stores a single about me text.
       title = models.CharField(max_length=200)
       updated_on = models.DateTimeField(auto_now=True)
       content = models.TextField()
        featured_image = CloudinaryField('image', default='placeholder')
       def __str__(self):
    return self.title
18  class CollaborateRequest(models.Model):
       Stores a single collaboration request message.
       name = models.CharField(max_length=200)
        email = models.EmailField()
        message = models.TextField()
       read = models.BooleanField(default=False)
       def __str__(self):
           return f"Collaboration request from {self.name}"
```

Settings:



Results:

About App – urls.py



About App – views.py

code

CI Python Linter

```
from django.shortcuts import render
                                                                                                                                  Settings:
   from django.contrib import messages
  from .models import About
from .forms import CollaborateForm
7 def about me(request):
                                                                                                                                  Results:
       Renders the most recent information on the website owner
       and allows for collaboration requests.
                                                                                                                                  All clear, no errors found
       Displays an individual instance of :model: about.About.
       **Context**
           The most recent instance of :model: about.About.
       ``collaborate form`
           An instance of :form: about.CollaborateForm .
       **Template:**
       :template: about/about.html
       if request.method == "POST":
           collaborate_form = CollaborateForm(data=request.POST)
           if collaborate form.is valid():
               collaborate form.save()
               messages.add message(
                   request, messages.SUCCESS,
                   """Collaboration request received!
                   I will respond to you within 3 working days.""")
       about = About.objects.all().order by('-updated on').first()
       collaborate form = CollaborateForm()
       return render(
           request,
           "about/about.html",
```

Blog App – admin.py



CI Python Linter

```
from django.contrib import admin
from django_summernote.admin import SummernoteModelAdmin
from .models import Review, Recipe, Comment

dadmin.register(Review)

class ReviewAdmin(SummernoteModelAdmin):

"""

tists fields for display in admin, fields for search,
field filters, fields to prepopulate and rich-text editor.

list display = ('title', 'slug', 'author', 'status', 'created_on')
search.fields = ['title', 'content']
list_filter = ('status', 'author', 'created_on',)
prepopulated fields = ('slug': ('title',))
summernote_fields = ('content',)

@admin.register(Recipe)
class RecipeAdmin(SummernoteModelAdmin):

"""

Lists fields for display in admin, fields for search,
field filters, fields to prepopulate and rich-text editor.

"""

list display = ('title', 'slug', 'type', 'author', 'status', 'created_on')
search fields = ['title', 'ingredients', 'instructions']
list_filter = ('status', 'author', 'type', 'created_on',)
prepopulated_fields = ('slug': ('title',))
summernote_fields = ('ingredients', 'instructions')

admin.site.register(comment)

admin.site.register(comment)
```

Settings:



Results:

Blog App – apps.py



Blog App – forms.py



Blog App – models.py



CI Python Linter

```
from django.db import models
from django.contrib.auth.models import User
 3 from cloudinary.models import CloudinaryField
 5 STATUS = ((0, "Draft"), (1, "Published"))
          (5, "Excellent - 5 Stars"), (4, "Good - 4 Stars"), (3, "Ok - 3 Stars"),
(3, "Could Be Better - 2 Stars"),
(1, "Poor - 1 Star"), (0, "Really Bad - 0 Stars"))

PRICE = ((0, "Low"), (1, "Medium"), (2, "High"))

TYPE = ((0, "Food"), (1, "Cocktail"))
14 class Review(models.Model):
          Stores a single restaurant review related to :model: auth.User.
         title = models.CharField(max_length=200, unique=True)
          slug = models.SlugField(max length=200, unique=True)
          author = models.ForeignKey(
              User, on_delete=models.CASCADE, related_name="blog_reviews"
          featured_image_1 = CloudinaryField('image', default='placeholder')
         featured_image_2 = CloudinaryField('image', default='placeholder')
         featured image 3 = CloudinaryField('image', default='placeholder')
         restaurant = models.CharField(max_length=200)
         content = models.TextField()
location = models.CharField(max_length=200)
         visited on = models.DateTimeField()
         rating = models.IntegerField(choices=RATING, default=5)
         price = models.IntegerField(choices=PRICE, default=0)
          created_on = models.DateTimeField(auto_now_add=True)
          status = models.IntegerField(choices=STATUS, default=0)
          updated_on = models.DateTimeField(auto_now=True)
```

Settings:



Results:

Blog App – urls.py



CI Python Linter

```
from django.urls import path
    from . import views
 4 urlpatterns = [
        path('', views.ReviewList.as_view(), name='home'),
        path('food recipes', views.FoodList.as view(), name='food'),
        path('cocktail_recipes', views.CocktailList.as_view(), name='cocktail'),
        path('<slug:slug>/', views.review_detail, name='review_detail'),
        path('food recipes/<slug:slug>/', views.food detail, name='food detail'),
            'cocktail_recipes/<slug:slug>/',
views.cocktail_detail, name='cocktail_detail'),
             'food_recipes/<slug:slug>/edit_comment/<int:comment_id>',
             views.food comment edit, name='food comment edit'),
        path(
             'cocktail_recipes/<slug:slug>/edit_comment/<int:comment_id>',
             views.cocktail comment edit, name='cocktail comment edit'),
        path(
             'food_recipes/<slug:slug>/delete_comment/<int:comment_id>',
             views.food comment delete, name='food comment delete'),
             'cocktail_recipes/<slug:slug>/delete_comment/<int:comment_id>',
             views.cocktail_comment_delete, name='cocktail_comment_delete'),
25
```

Settings:



Results:

Blog App – views.py

code

CI Python Linter

```
from django.shortcuts import render, get_object_or_404, reverse
                                                                                                                                              Settings:
 from django.views import generic from django.contrib import messages
4 from django.http import HttpResponseRedirect
5 from .models import Review, Recipe, Comment
6 from .forms import CommentForm
                                                                                                                                              Results:
 9  class ReviewList(generic.ListView):
         queryset = Review.objects.filter(status=1)
                                                                                                                                              All clear, no errors found
         template name = "blog/index.html"
         paginate by = 6
15 v class FoodList(generic.ListView):
         queryset = Recipe.objects.filter(status=1, type=0)
         template name = "blog/food.html"
         paginate by = 6
21 - class CocktailList(generic.ListView):
         queryset = Recipe.objects.filter(status=1, type=1)
         template name = "blog/cocktail.html"
         paginate_by = 6
27 def review_detail(request, slug):
         Displays an individual instance of :model: blog.Review.
         **Context**
             An instance of :model: blog.Review.
         :template:`blog/review_detail.html`
```