About App – admin.py

CI Python Linter

```
1   from django.contrib import admin
2   from django_summernote.admin import SummernoteModelAdmin
3   from .models import About, CollaborateRequest
4
5
6   @admin.register(About)
7   class AboutAdmin(SummernoteModelAdmin):
8       """
9       Adds rich-text editing of content in admin
10      """
11      summernote_fields = ('content',)
12
13
14  @admin.register(CollaborateRequest)
15  class CollaborateRequestAdmin(admin.ModelAdmin):
16      """
17      Lists message and read fields for display in admin
18      """
19      list_display = ('message', 'read',)
20
```

Settings:

🌙 ⚪ ☀

Results:

All clear, no errors found

About App – apps.py

CI Python Linter

```python
from django.apps import AppConfig


class AboutConfig(AppConfig):
    """
    Provides primary key type for about app
    """
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'about'
```

Settings:

Results:

All clear, no errors found

About App – forms.py

CI Python Linter

```python
1   from django import forms
2   from .models import CollaborateRequest
3
4
5   class CollaborateForm(forms.ModelForm):
6       """
7       Form for users to request collaboration
8       """
9       class Meta:
10          model = CollaborateRequest
11          fields = ('name', 'email', 'message')
12
```

Settings:

Results:

All clear, no errors found

About App – models.py

About App – urls.py

About App – views.py

CI Python Linter

```python
1   from django.shortcuts import render
2   from django.contrib import messages
3   from .models import About
4   from .forms import CollaborateForm
5
6
7   def about_me(request):
8       """
9       Renders the most recent information on the website owner
10      and allows for collaboration requests.
11      Displays an individual instance of :model:`about.About`.
12      **Context**
13      ``about``
14          The most recent instance of :model:`about.About`.
15      ``collaborate_form``
16          An instance of :form:`about.CollaborateForm`.
17      **Template:**
18      :template:`about/about.html`
19      """
20      if request.method == "POST":
21          collaborate_form = CollaborateForm(data=request.POST)
22          if collaborate_form.is_valid():
23              collaborate_form.save()
24              messages.add_message(
25                  request, messages.SUCCESS,
26                  """Collaboration request received!
27                  I will respond to you within 3 working days.""")
28
29      about = About.objects.all().order_by('-updated_on').first()
30      collaborate_form = CollaborateForm()
31
32      return render(
33          request,
34          "about/about.html",
```

Settings:

🌙 ⬜ ☀️

Results:

All clear, no errors found

Blog App – admin.py

CI Python Linter

```python
 1  from django.contrib import admin
 2  from django_summernote.admin import SummernoteModelAdmin
 3  from .models import Review, Recipe, Comment
 4
 5
 6  @admin.register(Review)
 7  class ReviewAdmin(SummernoteModelAdmin):
 8      """
 9      Lists fields for display in admin, fields for search,
10      field filters, fields to prepopulate and rich-text editor.
11      """
12      list_display = ('title', 'slug', 'author', 'status', 'created_on')
13      search_fields = ['title', 'content']
14      list_filter = ('status', 'author', 'created_on',)
15      prepopulated_fields = {'slug': ('title',)}
16      summernote_fields = ('content',)
17
18
19  @admin.register(Recipe)
20  class RecipeAdmin(SummernoteModelAdmin):
21      """
22      Lists fields for display in admin, fields for search,
23      field filters, fields to prepopulate and rich-text editor.
24      """
25      list_display = ('title', 'slug', 'type', 'author', 'status', 'created_on')
26      search_fields = ['title', 'ingredients', 'instructions']
27      list_filter = ('status', 'author', 'type', 'created_on',)
28      prepopulated_fields = {'slug': ('title',)}
29      summernote_fields = ('ingredients', 'instructions')
30
31
32  admin.site.register(Comment)
33
```

Settings:

🌙 ⬤ ☀

Results:

All clear, no errors found

Blog App – apps.py

CI Python Linter

```
1  from django.apps import AppConfig
2
3
4  class BlogConfig(AppConfig):
5      """
6      Provides primary key type for blog app
7      """
8      default_auto_field = 'django.db.models.BigAutoField'
9      name = 'blog'
10
```

Settings:

🌙 ⬤ ☀️

Results:

All clear, no errors found

Blog App – forms.py

CI Python Linter

```python
1  from django import forms
2  from .models import Comment
3
4
5  class CommentForm(forms.ModelForm):
6      """
7      Form for users to comment on a post
8      """
9      class Meta:
10         model = Comment
11         fields = ('body',)
12
```

Settings:

🌙 ⬤ ☀

Results:

All clear, no errors found

Blog App – models.py

CI Python Linter

```python
from django.db import models
from django.contrib.auth.models import User
from cloudinary.models import CloudinaryField

STATUS = ((0, "Draft"), (1, "Published"))
RATING = (
    (5, "Excellent - 5 Stars"), (4, "Good - 4 Stars"), (3, "Ok - 3 Stars"),
    (2, "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"))


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:

All clear, no errors found

Blog App – urls.py

CI Python Linter

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

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'),
    path(
        'cocktail_recipes/<slug:slug>/',
        views.cocktail_detail, name='cocktail_detail'),
    path(
        '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'),
    path(
        'cocktail_recipes/<slug:slug>/delete_comment/<int:comment_id>',
        views.cocktail_comment_delete, name='cocktail_comment_delete'),
]
```

Settings:

Results:

All clear, no errors found

Blog App – views.py

CI Python Linter

```python
1  from django.shortcuts import render, get_object_or_404, reverse
2  from django.views import generic
3  from django.contrib import messages
4  from django.http import HttpResponseRedirect
5  from .models import Review, Recipe, Comment
6  from .forms import CommentForm
7
8
9  class ReviewList(generic.ListView):
10     queryset = Review.objects.filter(status=1)
11     template_name = "blog/index.html"
12     paginate_by = 6
13
14
15 class FoodList(generic.ListView):
16     queryset = Recipe.objects.filter(status=1, type=0)
17     template_name = "blog/food.html"
18     paginate_by = 6
19
20
21 class CocktailList(generic.ListView):
22     queryset = Recipe.objects.filter(status=1, type=1)
23     template_name = "blog/cocktail.html"
24     paginate_by = 6
25
26
27 def review_detail(request, slug):
28     """
29     Displays an individual instance of :model:`blog.Review`.
30     **Context**
31     ``review``
32         An instance of :model:`blog.Review`.
33     **Template:**
34     :template:`blog/review_detail.html`
```

Settings:

🌙 ⬤ ☀

Results:

All clear, no errors found