


Forms.py

CI Python Linter



pep8ci.herokuapp.com



CI Python Linter

```
17         "description",
18         "accommodation_type",
19         "city",
20         "price_per_night",
21         "booking_image",
22         "owner_email",
23         "owner_phone",
24     ]
25
26     widgets = {
27         "booking_image": forms.FileInput(
28             attrs={"class": "custom-file-input"}
29         ),
30     }
31
32     def __init__(self, *args, **kwargs):
33         super(BookForm, self).__init__(*args, **kwargs)
34
35         # Hides part of image field in crispy if not required during the update
36         if self.instance and self.instance.pk:
37             self.fields["booking_image"].required = False
38
39         self.helper = FormHelper()
40         self.helper.layout = Layout(
41             Field("title"),
42             Field(
43                 "description", maxlength="600", placeholder="Add some details"
44             ),
45             Field("accommodation_type"),
46             Field("city"),
47             Field("price_per_night", label="Price per night, &euro;"),
48             Field("booking_image"),
49             Field("owner_email"),
50             Field("owner_phone", placeholder="Defaults to Irish numbers"),
51         )
```

Settings:

☒ 


Results:

All clear, no errors found

Models.py

CI Python Linter



pep8ci.herokuapp.com



CI Python Linter

```
45 owner_phone = PhoneNumberField(region="IE")
46
47 created_on = models.DateTimeField(auto_now_add=True)
48 updated_on = models.DateTimeField(auto_now=True)
49
50 slug = AutoSlugField(populate_from="title", max_length=200, unique=True)
51
52 class Meta:
53     # order the posts on when they were created
54     # newest posts first
55     ordering = ["-created_on"]
56
57 def __str__(self):
58     # django docs say to define this, returns a string from the class
59     return self.title
60
61 def get_absolute_url(self):
62     """
63     Get absolute url to redirect the user to the page
64     for the property they just posted.
65     """
66     return reverse("book_detail", args=[str(self.slug)])
67
68 def image_tag(self):
69     """Create custom view in admin panel displaying image preview"""
70     if "placeholder" in self.booking_image.url:
71         placeholder_url = static("images/default.png")
72         return mark_safe(
73             f''
74         )
75     return mark_safe(
76         f''
77     )
78
79 image_tag.short_description = "Image Preview"
80
```

Settings:

 ☒ 


Results:

All clear, no errors found

Urls.py

CI Python Linter



pep8ci.herokuapp.com



CI Python Linter

```
1 from . import views
2 from django.urls import path
3 from django.http import HttpResponseRedirect
4
5
6 urlpatterns = [
7     path("", views.BookView.as_view(), name="home"),
8     path("about", views.about_view, name="about"),
9     path("<slug:slug>/", views.BookDetail.as_view(), name="book_detail"),
10    path("add-book", views.BookCreateView.as_view(), name="property_create"),
11    path("property-list", views.PropertyView.as_view(), name="property_list"),
12    path(
13        "<slug:slug>/edit", views.BookUpdateView.as_view(), name="book_update"
14    ),
15    path(
16        "<slug:slug>/delete", views.BookDeleteView.as_view(),
17        name="book_delete"),
18 ]
19 |
```

Settings:

 ☒ 


Results:

All clear, no errors found

views.py

CI Python Linter



pep8ci.herokuapp.com



CI Python Linter

```
1 from django.shortcuts import render, get_object_or_404
2 from django.views import generic, View
3 from django.views.generic.edit import CreateView, UpdateView, DeleteView
4 from django.contrib.auth.mixins import LoginRequiredMixin, UserPassesTestMixin
5 from django.contrib import messages
6 from django.urls import reverse
7 from .models import BookPost
8 from .forms import BookForm
9 from django.db.models import Q
10
11
12 def about_view(request):
13     """ Method to display About page """
14     return render(request, 'about.html')
15
16
17 class BookView(generic.ListView):
18     """
19     View from generic for viewing paginated all real estate scope on homepage
20     """
21     model = BookPost
22     queryset = BookPost.objects.order_by('-created_on')
23     template_name = 'index.html'
24     context_object_name = 'book_list'
25     paginate_by = 6
26
27 def get_queryset(self, **kwargs):
28     """ Search method """
29     query = self.request.GET.get("q")
30     if query:
31         book_list = self.model.objects.filter(
32             Q(title__icontains=query)
33             | Q(description__icontains=query)
34             | Q(accommodation_type__icontains=query)
35             | Q(city__icontains=query),
```

Settings:

 ☒ 


Results:

All clear, no errors found

admin.py

CI Python Linter



pep8ci.herokuapp.com



CI Python Linter

```
1 from django.contrib import admin
2 from .models import BookPost
3
4
5 @admin.register(BookPost)
6 class PostAdmin(admin.ModelAdmin):
7     """
8     Custom admin panel with table view of property posts and picture preview
9     """
10
11     list_display = (
12         "image_tag",
13         "post_owner",
14         "title",
15         "slug",
16         "accommodation_type",
17         "city",
18         "price_per_night",
19         "created_on",
20     )
21     readonly_fields = ("image_tag",)
22     search_fields = ["title", "content"]
23     list_filter = (
24         "city",
25         "created_on",
26     )
27
28
29 """ Custom header naming """
30 admin.site.site_title = "Sweet Home Accommodation"
31 admin.site.index_title = "Administration panel"
32 admin.site.site_header = "Sweet Home Accommodation administrator panel"
33 |
```

Settings:

☒ 

Results:

All clear, no errors found