

admin.py

admin.py - abc-bookshop - Git | CI Python Linter | ABC Bookshop | Shopping Bag

pep8ci.herokuapp.com

code institute

CI Python Linter

```
1 from django.contrib import admin
2 from .models import Enquiry
3
4
5 class EnquiryAdmin(admin.ModelAdmin):
6     """ Enquiry model display """
7     list_display = (
8         'enquiry', 'answer', 'status', 'created_on'
9     )
10
11     search_fields = ('enquiry', 'answer')
12     ordering = ('created_on',)
13
14
15 admin.site.register(Enquiry, EnquiryAdmin)
16
```

Settings:

☒ ☐ ☐

Results:

All clear, no errors found

apps.py

apps.py - abc-bookshop - Gitp... x CI Python Linter x ABC Bookshop | Shopping Bag x +

← → ↻ pep8ci.herokuapp.com

code
institute

CI Python Linter

```
1 from django.apps import AppConfig
2
3
4 class EnquiryConfig(AppConfig):
5     default_auto_field = 'django.db.models.BigAutoField'
6     name = 'enquiry'
7
```

Settings:

🌙 ☒ ☐ ☀️

Results:

All clear, no errors found

urls.py

urls.py - abc-bookshop - Gitpo...

CI Python Linter

ABC Bookshop | Shopping Bag

pep8ci.herokuapp.com




code

institute

CI Python Linter

```
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('', views.all_enquiries, name='enquiries'),
6     path('add/', views.add_enquiry, name='add_enquiry'),
7     path('edit/<int:enquiry_id>/', views.edit_enquiry, name='edit_enquiry'),
8     path(
9         'delete/<int:enquiry_id>/',
10        views.delete_enquiry,
11        name='delete_enquiry'),
12 ]
13 |
```

Settings:



Results:

All clear, no errors found

views.py

views.py - abc-bookshop - Gitp

CI Python Linter

ABC Bookshop | Books

← → ↻ pep8ci.herokuapp.com

code

institute

CI Python Linter

```
70         form.save()
71         messages.success(request, "Enquiry was edited successfully.")
72         return redirect(reverse("enquiries"))
73     else:
74         messages.error(
75             request, "Enquiry was not edited. Correct the form inputs.")
76 else:
77     form = EnquiryForm(instance=enquiry)
78
79     template = "enquiry/edit_enquiry.html"
80     context = {
81         "form": form,
82         "enquiry": enquiry,
83         "on_page": True,
84     }
85
86     return render(request, template, context)
87
88
89 @login_required
90 def delete_enquiry(request, enquiry_id):
91     """Delete an enquiry"""
92     if not request.user.is_superuser:
93         messages.error(request, "Sorry, only the admin can delete enquiries.")
94         return redirect(reverse("enquiries"))
95
96     enquiry = get_object_or_404(Enquiry, pk=enquiry_id)
97     enquiry.delete()
98     messages.success(request, "Enquiry was deleted successfully.")
99
100     context = {
101         "on_page": True,
102     }
103
104     return redirect(reverse("enquiries"), context)
105
```

Settings:

☐ ☒ ☐

Results:

All clear, no errors found

forms.py

forms.py - abc-bookshop - Gitp





CI Python Linter


ABC Bookshop | Books

+

← → ↺

pep8ci.herokuapp.com








CI Python Linter

```
1 from .models import Enquiry
2 from django import forms
3
4 STATUS = ((0, 'Draft'), (1, 'Published'))
5
6
7 class EnquiryForm(forms.ModelForm):
8     """ Form to create/edit an enquiry """
9     class Meta:
10         """ Enquiry form """
11         model = Enquiry
12         fields = '__all__'
13
```

Settings:



Results:

All clear, no errors found

models.py


models.py - abc-bookshop - Git x

CI Python Linter x

ABC Bookshop | Books x

+



← → ↻ pep8ci.herokuapp.com



CI Python Linter

```
1 from django.db import models
2
3
4 STATUS = ((0, "Draft"), (1, "Published"))
5
6
7 class Enquiry(models.Model):
8     """ Model for creating/managing enquiries """
9     enquiry = models.CharField(max_length=254)
10    answer = models.TextField()
11    created_on = models.DateTimeField(auto_now_add=True)
12    updated_on = models.DateTimeField(auto_now=True)
13    status = models.IntegerField(choices=STATUS, default=0)
14
15    class Meta:
16        """ Order Enquiries from newest to oldest """
17        ordering = ['created_on']
18        verbose_name_plural = 'Enquiries'
19
20    def __str__(self):
21        return self.enquiry
22
```

Settings:

 ☒ 

Results:

All clear, no errors found