apps.py

CI Python Linter

ABC Bookshop | Books

pep8ci.herokuapp.com

code
institute

CI Python Linter

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

Settings:

🌙 ⬤ ☀

Results:

All clear, no errors found

urls.py


```python
1  from django.urls import path
2  from . import views
3
4
5  urlpatterns = [
6      path('contact', views.contact, name="contact"),
7      path('thanks', views.thanks, name="thanks"),
8  ]
9
```

CI Python Linter

Settings:

Results:

All clear, no errors found

views.py

code institute

CI Python Linter

```
20            "email": form.cleaned_data["email_address"],
21            "message": form.cleaned_data["message"],
22        }
23        message = "\n".join(body.values())
24
25        try:
26            send_mail(
27                subject,
28                message,
29                "dimmando.travel@gmail.com",
30                ["dimmando.travel@gmail.com"],
31            )
32        except BadHeaderError:
33            return HttpResponse("Invalid header found.")
34        messages.success(request, "Your inquiry was sent successfully.")
35        return redirect("thanks")
36        else:
37            messages.warning(
38                request, "Wrong email. It should look like name@domain.com."
39            )
40    else:
41        form = ContactForm()
42
43    context = {
44        "form": form,
45        "on_page": True,
46    }
47
48    return render(request, "contact/contact.html", context)
49
50
51    def thanks(request):
52        """A view to return the thank you page"""
53
54        return render(request, "contact/thankyou.html")
55
```

Settings:

🌙 ⬜ ☀️

Results:

All clear, no errors found

forms.py

pep8ci.herokuapp.com

CI Python Linter

```python
from django import forms


class ContactForm(forms.Form):
    """
    Contact form for requests.
    All entries are validated. Names are checked if they contain
    numbers or special characters (front-end validation).
    """

    first_name = forms.CharField(
        max_length=20,
        required=True,
        widget=forms.TextInput(attrs={
            'pattern': r"^[ \u00c0-\u01ffa-zA-Z'\-]+$",
            'title': 'Please enter a valid first name (letters only).'
        })
    )
    last_name = forms.CharField(
        max_length=20,
        required=True,
        widget=forms.TextInput(attrs={
            'pattern': r"^[ \u00c0-\u01ffa-zA-Z'\-]+$",
            'title': 'Please enter a valid last name (letters only).'
        })
    )
    email_address = forms.EmailField(
        required=True,
        max_length=150,
    )

    message = forms.CharField(widget=forms.Textarea, max_length=2000)

    class Meta:
        fields = ['first_name', 'last_name', 'email_address', 'message']
```

Settings:

Results:

All clear, no errors found