

apps.py

apps.py - abc-bookshop - Gitp...

CI Python Linter

ABC Bookshop | Books

pep8ci.herokuapp.com

code

institute

CI Python Linter

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

Settings:

Results:

All clear, no errors found


urls.py

urls.py - abc-bookshop - Gitpod

CI Python Linter

ABC Bookshop | Books



pep8ci.herokuapp.com



CI Python Linter

```
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('', views.profile, name='profile'),
6     path('order_history/<order_number>',
7         views.order_history, name='order_history'
8     ),
9 ]
10
```

Settings:

 ☒ 

Results:

All clear, no errors found


views.py

views.py - abc-bookshop - Gitp

CI Python Linter

ABC Bookshop | Books




pep8ci.herokuapp.com



CI Python Linter

```
20         messages.success(request, 'Your profile was updated successfully.')
21     else:
22         messages.error(
23             request, 'Profile was not updated. Please check form inputs.')
24     else:
25         form = UserProfileForm(instance=profile)
26         orders = profile.orders.all()
27
28         template = 'profiles/profile.html'
29         context = {
30             'form': form,
31             'orders': orders,
32             'on_page': True,
33         }
34
35         return render(request, template, context)
36
37
38 def order_history(request, order_number):
39     """ Create order history """
40     order = get_object_or_404(Order, order_number=order_number)
41
42     messages.info(request, (
43         f'This is a past confirmation for order number {order_number}. '
44         'An email confirmation was sent on the order date.'
45     ))
46
47     template = 'checkout/checkout_success.html'
48     context = {
49         'order': order,
50         'from_profile': True,
51         'on_page': True,
52     }
53
54     return render(request, template, context)
55
```

Settings:



Results:

All clear, no errors found

forms.py

forms.py - abc-bookshop - Gitp

CI Python Linter

ABC Bookshop | Books

← → ↺ pep8ci.herokuapp.com

code

institute

CI Python Linter

```
11         'default_phone_number': 'Phone Number',
12         'default_postcode': 'Postal Code',
13         'default_town_or_city': 'Town or City',
14         'default_street_address1': 'Street Address 1',
15         'default_street_address2': 'Street Address 2',
16         'default_county': 'County, State or Locality',
17     }
18
19     def __init__(self, *args, **kwargs):
20         """
21         Add placeholders and classes, remove auto-generated
22         labels and set autofocus on first field
23         """
24         super().__init__(*args, **kwargs)
25
26         placeholders = {
27             'default_phone_number': 'Phone Number',
28             'default_postcode': 'Postal Code',
29             'default_town_or_city': 'Town or City',
30             'default_street_address1': 'Street Address 1',
31             'default_street_address2': 'Street Address 2',
32             'default_county': 'County, State or Locality',
33         }
34
35         self.fields['default_phone_number'].widget.attrs['autofocus'] = True
36         for field in self.fields:
37             if field != 'default_county':
38                 if self.fields[field].required:
39                     placeholder = f'{placeholders[field]} *'
40                 else:
41                     placeholder = placeholders[field]
42                 self.fields[field].widget.attrs['placeholder'] = placeholder
43                 self.fields[field].widget.attrs['class'] = (
44                     'border-black rounded-0 profile-form-input'
45                 )
46
```

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

code institute

CI Python Linter

```
17
18
19     default_street_address1 = models.CharField(
20         max_length=80, null=True, blank=True
21     )
22     default_street_address2 = models.CharField(
23         max_length=80, null=True, blank=True
24     )
25     default_town_or_city = models.CharField(
26         max_length=40, null=True, blank=True
27     )
28     default_postcode = models.CharField(
29         max_length=20, null=True, blank=True
30     )
31     default_county = models.CharField(
32         max_length=80, null=True, blank=True
33     )
34     default_country = CountryField(
35         blank_label='Country', null=True, blank=True
36     )
37
38     def __str__(self):
39         return self.user.username
40
41 @receiver(post_save, sender=User)
42 def create_or_update_user_profile(sender, instance, created, **kwargs):
43     """
44     Create or update the user profile and wishlist
45     """
46     if created:
47         user_profile = UserProfile.objects.create(user=instance)
48         Wishlist = apps.get_model('wishlist', 'Wishlist')
49         Wishlist.objects.create(user_profile=user_profile)
50     # Existing users: just save the profile
51     instance.userprofile.save()
52
```

Settings:

☒ ☐ ☐

Results:

All clear, no errors found