

admin.py

admin.py - abc-bookshop - Git | x Select user profile to change | A x CI Python Linter x +

pep8ci.herokuapp.com

code institute

CI Python Linter

```
1 from django.contrib import admin
2 from .models import UserProfile
3 from wishlist.models import Wishlist
4
5
6 class UserProfileAdmin(admin.ModelAdmin):
7     list_display = ("user", "get_wishlist_products")
8
9     def get_wishlist_products(self, obj):
10         wishlist = Wishlist.objects.filter(user_profile=obj).first()
11         if wishlist:
12             return ", ".join(
13                 [product.title for product in wishlist.product.all()])
14         return "No products"
15
16     get_wishlist_products.short_description = "Wishlist Products"
17
18
19 admin.site.register(UserProfile, UserProfileAdmin)
20
```

Settings:

🌙 ☒ ☐ ☀️

Results:

All clear, no errors found

apps.py

apps.py - abc-bookshop - Gitp... x | 94ABCFE4A9254D0D87FD848A... x | CI Python Linter x +

← → ↻ pep8ci.herokuapp.com

code
institute

CI Python Linter

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

Settings:

🌙 ☒ ☐ ☀️

Results:

All clear, no errors found

urls.py

urls.py - abc-bookshop - Gitpod x 94ABCFE4A9254D0D87FD848A x CI Python Linter x +

← → ↻ pep8ci.herokuapp.com

code institute

CI Python Linter

```
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('', views.wishlist, name='wishlist'),
6     path('add/<int:product_id>/', views.add_to_wishlist,
7         name='add_to_wishlist'),
8     path('remove/<int:product_id>/',
9         views.remove_from_wishlist, name='remove_from_wishlist'),
10 ]
11 |
```

Settings:

🌙 ☒ ☐ ☀️

Results:

All clear, no errors found

views.py

The screenshot shows a web browser window with the URL `pep8ci.herokuapp.com`. The page title is "CI Python Linter" and the logo for "code institute" is visible. The main content area displays a Python code file named `views.py` with the following code:

```
26 def add_to_wishlist(request, product_id):
27     """ Add the chosen product into the wishlist """
28
29     product = get_object_or_404(Product, pk=product_id)
30     user_profile = UserProfile.objects.get(user=request.user)
31
32     wishlist, created = Wishlist.objects.get_or_create(user_profile=user_profile)
33     if product in wishlist.product.all():
34         messages.warning(request, f'You already have {product.title} in your wishlist.')
35     else:
36         wishlist.product.add(product)
37         messages.success(request, f'You have successfully added {product.title} to your wishlist.')
38
39     return redirect(reverse('wishlist'))
40
41
42 @login_required
43 def remove_from_wishlist(request, product_id):
44
45     """ Remove the chosen product from the wishlist """
46
47     product = get_object_or_404(Product, pk=product_id)
48     user_profile = UserProfile.objects.get(user=request.user)
49
50     try:
51         wishlist = Wishlist.objects.get(user_profile=user_profile)
52         if product in wishlist.product.all():
53             wishlist.product.remove(product)
54             messages.success(request, f'{product.title} was successfully removed from your wishlist
55             .')
56         else:
57             messages.warning(request, f'{product.title} is not in your wishlist.')
58     except Wishlist.DoesNotExist:
59         messages.error(request, 'You do not have a wishlist yet.')
60
61     return redirect(reverse('wishlist'))
```

On the right side of the interface, there is a "Settings:" section with a dark mode toggle switch and a "Results:" section that states "All clear, no errors found".

contexts.py

The screenshot shows a web browser window with the URL `pep8ci.herokuapp.com`. The page title is "CI Python Linter" and the logo for "code institute" is visible. The main content area is a dark-themed code editor displaying a Python file named `contexts.py`. The code defines a `wishlist_contents` function that checks if a user is authenticated and returns a dictionary with a `customer_wishlist` key. To the right of the code editor, there is a "Settings:" section with a dark mode toggle switch and a "Results:" section showing the message "All clear, no errors found".

```
1 from profiles.models import UserProfile
2 from .models import Wishlist
3
4
5 def wishlist_contents(request):
6     """Wishlist contents context processor"""
7
8     if request.user.is_authenticated:
9         wishlist_items = []
10        user_profile = UserProfile.objects.get(user=request.user)
11        wishlist_queryset = Wishlist.objects.filter(user_profile=user_profile)
12        # if there are items in the users wishlist
13        for item in wishlist_queryset:
14            wishlist_items.append(item.product)
15    else:
16        wishlist_items = []
17    return {
18        'customer_wishlist': wishlist_items,
19    }
20
```

Settings:

Results:

All clear, no errors found

models.py

The screenshot shows a web browser window with the URL `pep8ci.herokuapp.com`. The page title is "CI Python Linter" and the logo for "code institute" is visible. The main content area displays a Python code snippet from a file named `models.py`. The code defines a Django model class `Wishlist` that inherits from `models.Model`. It includes two database fields: `user_profile` (a `OneToOneField` to `UserProfile`) and `product` (a `ManyToManyField` to `Product`). A `__str__` method is also defined, returning a string representation of the model.

```
1 from django.db import models
2 from profiles.models import UserProfile
3 from products.models import Product
4
5
6 class Wishlist(models.Model):
7     """ A model to add favourite products into a wishlist """
8     user_profile = models.OneToOneField(
9         UserProfile,
10         on_delete=models.CASCADE,
11         related_name="customer_wishlist",
12         default=None
13     )
14     product = models.ManyToManyField(
15         Product, default=None
16     )
17
18     def __str__(self):
19         """Return the string representation of the model."""
20         return str(self.user_profile) + '\s wishlist'
21
```

On the right side of the interface, there is a "Settings:" section with a dark mode toggle switch (currently off) and a "Results:" section stating "All clear, no errors found".