

Basket

apps.py

codeinstitute

CI Python Linter

```
1 from django.apps import AppConfig
2
3
4 class StoreBasketConfig(AppConfig):
5     """
6     Configuration for the Store Basket app.
7     """
8     default_auto_field = 'django.db.models.BigAutoField'
9     name = 'store_basket'
10 |
```

Settings:

☒

Results:

All clear, no errors found

context_processors.py

codeinstitute

CI Python Linter

```
1 from .models import Basket
2
3
4 def basket(request):
5     """
6     Context processor for current user's shopping basket.
7     """
8     return {'basket': Basket(request)}
9 |
```

Settings:

☒

Results:

All clear, no errors found

Basket

Models.py

codeinstitute

CI Python Linter

```
1 from decimal import Decimal
2 from django.conf import settings
3
4 from store.models import Product
5
6
7 class Basket():
8     """
9     A base Basket class, providing some default behaviors that
10     can be inherited or overridden, as necessary.
11     """
12
13     def __init__(self, request):
14         """
15         Initializes the basket object.
16         """
17         self.session = request.session
18         basket = self.session.get(settings.BASKET_SESSION_ID)
19         if settings.BASKET_SESSION_ID not in request.session:
20             basket = self.session[settings.BASKET_SESSION_ID] = {}
21         self.basket = basket
22
23     def add(self, product, qty):
24         """
25         Adds or updates a product in the basket.
26         """
27         product_id = str(product.id)
28
29         if product_id in self.basket:
30             self.basket[product_id]['qty'] += qty
31         else:
```

Settings:

Results:

All clear, no errors found

urls.py

codeinstitute

CI Python Linter

```
1 """
2 URL configuration for store_basket app.
3
4 """
5
6 from django.urls import path
7 from django.http import HttpResponse
8 from . import views
9
10 app_name = 'store_basket'
11
12 urlpatterns = [
13     path('', views.basket_summary, name='basket_summary'),
14     path('add/', views.add_to_basket, name='add_to_basket'),
15     path('delete/', views.basket_delete, name='basket_delete'),
16     path('update/', views.basket_update, name='basket_update'),
17     path('clear/', views.clear_basket, name='clear_basket'),
18 ]
19
```

Settings:

Results:

All clear, no errors found

Basket

views.py

codeinstitute

CI Python Linter

```
1 from django.contrib import messages
2 from django.shortcuts import render, get_object_or_404
3 from django.http import JsonResponse, HttpResponseRedirect
4 from django.urls import reverse
5
6 from .models import Basket
7 from store.models import Product
8
9
10 def basket_summary(request):
11     """
12     Renders the summary page of the basket.
13     """
14     basket = Basket(request)
15     number_range = range(1, 101)
16     context = {
17         'number_range': number_range,
18         'basket': basket,
19     }
20     return render(request, 'basket/summary.html', context)
21
22
23 def add_to_basket(request):
24     """
25     Adds a product to the basket.
26     """
27     basket = Basket(request)
28     if request.POST.get('action') == 'post':
29         product_id = int(request.POST.get('productid'))
30         product_qty = int(request.POST.get('productqty'))
31         product = get_object_or_404(Product, id=product_id)
```

Settings:

☒

Results:

All clear, no errors found