



Basket

views.py

```
1 from django.shortcuts import (
2     render, redirect, reverse, HttpResponseRedirect, get_object_or_404)
3 from django.contrib import messages
4
5 from products.models import Product
6
7
8 def basket(request):
9     """
10     A view to show the basket page for
11     the customer
12     """
13     return render(request, 'basket/basket.html')
14
15
16 def add_to_basket(request, item_id):
17     """
18     A view to add a quantity of products to
19     the shopping basket
20     """
21
22     product = get_object_or_404(Product, pk=item_id)
23
24     quantity = int(request.POST.get('quantity'))
```

Settings:

 ☒ 



Results:

All clear, no errors found

urls.py

```
1 from django.urls import path
2
3 from . import views
4
5 # URLs to add, update, and remove products from shopping basket
6
7 urlpatterns = [
8     path('', views.basket, name='basket'),
9     path('add/<item_id>', views.add_to_basket, name='add_to_basket'),
10    path('update/<item_id>', views.update_basket, name='update_basket'),
11    path('remove/<item_id>', views.delete_from_basket,
12         name='delete_from_basket'),
13 ]
14 |
```

Settings:

 ☒ 



Results:

All clear, no errors found

context.py

```
1 from decimal import Decimal
2 from django.conf import settings
3 from django.shortcuts import get_object_or_404
4
5 from products.models import Product
6
7
8 def basket_contents(request):
9     """
10     A view to show the contents of the basket
11     to the customer
12     """
13
14     basket_items = []
15     total = 0
16     product_count = 0
17     basket = request.session.get('basket', {})
18
```

Settings:

 ☒ 

Results:

All clear, no errors found

Checkout

admin.py

```
1 from django.contrib import admin
2 from .models import Order, OrderLineItem
3
4 # Add all model fields to the admin page
5 # Source: https://stackoverflow.com/questions/10543032/
6
7
8 class OrderLineItemAdminInline(admin.TabularInline):
9     # Admin for OrderLineItem model
10     model = OrderLineItem
11     readonly_fields = ('lineitem_total',)
12
13
14 @admin.register(Order)
```

Settings:

🌙 ☒ ☀️

Results:

All clear, no errors found

forms.py

```
1 from django.forms import CharField, TextInput
2 from django import forms
3 from .models import Order
4
5
6 class OrderForm(forms.ModelForm):
7     """ Form for customer to fill out their
8     personal details in checkout """
9     phone_number = CharField(
10         widget=TextInput(attrs={'type': 'number'}))
11
12 class Meta:
13     model = Order
14     fields = ('full_name', 'email', 'phone_number',
15             'street_address1', 'street_address2',
16             'town_or_city', 'postcode', 'country')
```

Settings:

🌙 ☒ ☀️

Results:

All clear, no errors found

models.py

```
1 from django.db import models
2 from django.db.models import Sum
3 from django.conf import settings
4
5 from django_countries.fields import CountryField
6
7 from products.models import Product
8 from profile.models import UserProfile
9
10 import uuid
11
12
13 class Order(models.Model):
14     """ Model for product orders """
15     order_number = models.CharField(
```

Settings:

🌙 ☒ ☀️

Results:

All clear, no errors found

signals.py

```
1 from django.db.models.signals import post_save, post_delete
2 from django.dispatch import receiver
3
4 from .models import OrderLineItem
5
6
7 @receiver(post_save, sender=OrderLineItem, weak=False)
8 def update_on_save(sender, instance, created, **kwargs):
9     """
10     Update order total on lineitem update/create
11     """
12     instance.order.update_total()
13
14
15 @receiver(post_delete, sender=OrderLineItem, weak=False)
16 def update_on_delete(sender, instance, **kwargs):
17     """
```

Settings:

🌙 ☒ ☀️

Results:

All clear, no errors found

urls.py - Fixed

```
1 from django.urls import path
2 from . import views
3 from .webhooks import webhook
4
5 # URLs to view checkout template, to view order history, and for webhooks
  connected to Stripe
6
7 urlpatterns = [
8     path('', views.checkout, name='checkout'),
9     path('checkout_success/<order_number>', views.checkout_success,
10         name='checkout_success'),
11     path('wh/', webhook, name='webhook'),
12     path('cache_checkout_data/', views.cache_checkout_data,
13         name='cache_checkout_data'),
14 ]
15
```

Settings:



Results:

5: E501 line too long (93 > 79 characters)

```
1 from django.urls import path
2 from . import views
3 from .webhooks import webhook
4
5 """URLS to view checkout template, to view order history,
6    and for webhooks connected to Stripe """
7
8 urlpatterns = [
9     path('', views.checkout, name='checkout'),
10    path('checkout_success/<order_number>', views.checkout_success,
11        name='checkout_success'),
12    path('wh/', webhook, name='webhook'),
13    path('cache_checkout_data/', views.cache_checkout_data,
14        name='cache_checkout_data'),
15 ]
16
```

Settings:



Results:

All clear, no errors found

views.py

```
1 from django.shortcuts import (
2     render, redirect, reverse, get_object_or_404, HttpResponseRedirect)
3 from django.views.decorators.http import require_POST
4 from django.contrib import messages
5 from django.conf import settings
6
7 from .forms import OrderForm
8 from products.models import Product
9 from .models import Order, OrderLineItem
10 from basket.context import basket_contents
11 from profile.forms import UserProfileForm
12 from profile.models import UserProfile
13
14 import stripe
15 import json
16
17
18 @require_POST
19 def cache_checkout_data(request):
20     """ STRIPE payment """
21     try:
```

Settings:



Results:

All clear, no errors found

webhook_handler.py

```
1 from django.http import HttpResponseRedirect
2 from django.core.mail import send_mail
3 from django.template.loader import render_to_string
4 from django.conf import settings
5
6 from .models import Order, OrderLineItem
7 from products.models import Product
8 from profile.models import UserProfile
9
10 import stripe
11 import json
12 import time
13
14
15 class StripeWH_Handler:
16     """ Handle Stripe webhooks """
17
```

Settings:



Results:

All clear, no errors found

webhooks.py

```
1 from django.conf import settings
2 from django.http import HttpResponse
3 from django.views.decorators.http import require_POST
4 from django.views.decorators.csrf import csrf_exempt
5
6 from checkout.webhook_handler import StripeWH_Handler
7
8 import stripe
9
10
11 @require_POST
12 @csrf_exempt
13 def webhook(request):
14     """Listen for webhooks from Stripe"""
15     # Setup
16     wh_secret = settings.STRIPE_WH_SECRET
17     stripe.api_key = settings.STRIPE_SECRET_KEY
18
```

Settings:



Results:

All clear, no errors found

Company

admin.py

```
1 from django.contrib import admin
2
3 from .models import Contact
4
5
6 @admin.register(Contact)
7 class ContactAdmin(admin.ModelAdmin):
8     """ Admin for contact model """
9     list_display = (
10         'email',
11         'subject',
12     )
13
```

Settings:



Results:

All clear, no errors found

forms.py

```
1 from django.forms import ModelForm
2 from .models import Contact
3
4
5 class ContactForm(ModelForm):
6     """ Form to contact the company """
7     class Meta:
8         model = Contact
9         fields = '__all__'
10
```

Settings:



Results:

All clear, no errors found

models.py

```
1 from django.db import models
2
3
4 class Contact(models.Model):
5     """ Model for contact form """
6     email = models.EmailField()
7     subject = models.CharField(max_length=255)
8     message = models.TextField()
9
10     def __str__(self):
11         return self.email
12
```

Settings:



Results:

All clear, no errors found

urls.py

```
1 from django.urls import path, include
2 from . import views
3
4 # URLs to view company info, faq, and contact templates
5
6 urlpatterns = [
7     path('', views.company_info, name='company_info'),
8     path('faq', views.company_faq, name='company_faq'),
9     path('contact', views.company_contact, name='company_contact'),
10 ]
11 |
```

Settings:



Results:

All clear, no errors found

views.py

```
1 from django.shortcuts import render
2 from django.conf import settings
3 from django.contrib import messages
4 from django.core.mail import send_mail
5
6 from .forms import ContactForm
7
8
9 def company_info(request):
10     """
11     View to display the template for
12     company information
13     """
14     return render(request, 'company_info/company_info.html')
15
16
17 def company_faq(request):
18     """
19     View to display the template for
20     company faq
21     """
22     return render(request, 'company_faq/company_faq.html')
```

Settings:



Results:

All clear, no errors found

Home

urls.py

```
1 from django.urls import path
2 from . import views
3
4 # URLs to view the home template
5
6 urlpatterns = [
7     path('', views.home, name='home'),
8 ]
9 |
```

Settings:



Results:

All clear, no errors found

views.py

```
1 from django.shortcuts import render
2
3
4 def home(request):
5     """
6     A view to return the home page
7     """
8
9     return render(request, 'home/index.html')
10 |
```

Settings:



Results:

All clear, no errors found

Knits&Knots

urls.py

```
1 from django.contrib import admin
2 from django.urls import path, include
3
4 from django.conf import settings
5 from django.conf.urls.static import static
6 from .views import handler404, handler500
7
8
9 urlpatterns = [
10     path('admin/', admin.site.urls),
11     path('accounts/', include('allauth.urls')),
12     path('', include('home.urls')),
13     path('company/', include('company.urls')),
14     path('products/', include('products.urls')),
15     path('basket/', include('basket.urls')),
16     path('checkout/', include('checkout.urls')),
17     path('wishlist/', include('wishlist.urls')),
18     path('profile/', include('profile.urls')),
19 ] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
20
21 handler404 = 'knits_knots.views.handler404'
22 handler500 = 'knits_knots.views.handler500'
23
24 |
```

Settings:



Results:

All clear, no errors found

views.py

```
1 from django.shortcuts import render
2
3
4 def handler404(request, exception):
5     """
6     A view for an error page if the user
7     tries to visit an unexisting page
8     """
9     return render(request, "errors/404.html", status=404)
10
11
12 def handler500(request):
13     """
14     A view for an error page if the user
15     gets an internal error
16     """
17     return render(request, "errors/500.html", status=500)
18
19 |
```

Settings:



Results:

All clear, no errors found

Products

admin.py

```
1 from django.contrib import admin
2 from .models import Category, Product, Tag, Videocall
3 from django_summernote.admin import SummernoteModelAdmin
4
5
6 @admin.register(Tag)
7 class TagAdmin(admin.ModelAdmin):
8     """ Admin for Tag model """
9     list_display = (
10         'name',
11         'slug',
12     )
13
14
15 @admin.register(Category)
16 class CategoryAdmin(admin.ModelAdmin):
17     """ Admin for category model """
18     list_display = (
```

Settings:



Results:

All clear, no errors found

forms.py

```
1 from django import forms
2 from .models import Product, Category, Videocall
3 from django.forms.widgets import NumberInput
4 from django.contrib.admin.widgets import AdminDateWidget
5 from django.contrib.admin import widgets
6
7
8 class ProductForm(forms.ModelForm):
9     """
10     A form for the superuser to add or
11     edit products on the webpage
12     """
13
14     class Meta:
15         model = Product
16         fields = ('name', 'description', 'price', 'category')
```

Settings:



Results:

All clear, no errors found

models.py

```
1 from django.db import models
2 from django.urls import reverse
3 from django.contrib.auth.models import User
4
5 import uuid
6 import datetime
7
8
9 class Category(models.Model):
10     """ Model for product categories """
11     name = models.CharField(
12         max_length=200,
13         db_index=True)
14
15
16 class Product(models.Model):
17     """ Model for products """
18     name = models.CharField(
19         max_length=200,
20         db_index=True)
21     description = models.TextField(
22         max_length=1000)
23     price = models.DecimalField(
24         max_digits=10,
25         decimal_places=2)
26     category = models.ForeignKey(
27         Category, on_delete=models.CASCADE)
28     created_at = models.DateTimeField(
29         auto_now_add=True)
30     updated_at = models.DateTimeField(
31         auto_now=True)
32
33
34 class Videocall(models.Model):
35     """ Model for videocalls """
36     product = models.ForeignKey(
37         Product, on_delete=models.CASCADE)
38     url = models.CharField(
39         max_length=200,
40         db_index=True)
41     created_at = models.DateTimeField(
42         auto_now_add=True)
43     updated_at = models.DateTimeField(
44         auto_now=True)
```

Settings:



Results:

All clear, no errors found

urls.py - Fixed

```
1 from django.urls import path, include
2 from products import views
3
4 """ URLS for users to view videocall template, products
5 and product details, and for superuser to edit and delete products """
6
7 urlpatterns = [
8     path('', views.product_all, name='product'),
9     path('videocall/', views.videocall, name='videocall'),
10    path('videocall_description/', views.videocall_description,
11         name='videocall_description'),
12    path('<int:product_id>/', views.product_detail, name='product_detail'),
13    path('add/', views.add_product, name='add_product'),
14    path('edit/<int:product_id>/', views.edit_product, name='edit_product'),
```

Settings:



Results:

4: W291 trailing whitespace

```
1 from django.urls import path, include
2 from products import views
3
4 """ URLS for users to view videocall template, products
5 and product details, and for superuser to edit and delete products """
6
7 urlpatterns = [
8     path('', views.product_all, name='product'),
9     path('videocall/', views.videocall, name='videocall'),
10    path('videocall_description/', views.videocall_description,
11         name='videocall_description'),
12    path('<int:product_id>/', views.product_detail, name='product_detail'),
13    path('add/', views.add_product, name='add_product'),
14    path('edit/<int:product_id>/', views.edit_product, name='edit_product'),
```

Settings:



Results:

All clear, no errors found

views.py

```
1 from django.shortcuts import render, redirect, reverse, get_object_or_404
2 from django.contrib.auth.decorators import login_required
3 from django.http import HttpResponseRedirect
4 from django.core.mail import send_mail
5 from django.contrib import messages
6 from django.conf import settings
7 from django.http import Http404
8 from django.db.models import Q
9
10 from .models import Product, Category, Tag, Videocall
11 from wishlist.models import Wishlist
12 from .forms import ProductForm, VideocallForm
13
14
15 def tag(request):
16     """
```

Settings:



Results:

All clear, no errors found

Profile

forms.py

```
1 from django.forms import CharField, TextInput
2 from django import forms
3 from .models import UserProfile
4
5
6 class UserProfileForm(forms.ModelForm):
7     """ Form for user to fill out their details """
8     default_phone_number = CharField(
9         widget=TextInput(attrs={'type': 'number', }))
10
11 class Meta:
12     model = UserProfile
13     exclude = ('user',)
14
15 def __init__(self, *args, **kwargs):
16     """
17     Add placeholders and classes, remove auto-generated
```

Settings:

Results:

All clear, no errors found

models.py

```
1 from django.db import models
2 from django.contrib.auth.models import User
3 from django.db.models.signals import post_save
4 from django.dispatch import receiver
5
6 from django_countries.fields import CountryField
7
8
9 class UserProfile(models.Model):
10     """
11     User model to save users delivery
12     information and order history
13     """
14     user = models.OneToOneField(
15         User,
16         on_delete=models.CASCADE)
17     default_full_name = models.CharField(
```

Settings:

Results:

All clear, no errors found

urls.py

```
1 from django.urls import path
2
3 from . import views
4 # URLs for users to view their order history
5
6 urlpatterns = [
7     path('', views.profile, name='profile'),
8     path('order_history/<order_number>', views.order_history,
9         name='order_history'),
10 ]
11 |
```

Settings:

Results:

All clear, no errors found

views.py

```
1 from django.shortcuts import render, get_object_or_404
2 from django.contrib.auth.decorators import login_required
3 from django.contrib import messages
4
5 from .models import UserProfile
6 from .forms import UserProfileForm
7 from checkout.models import Order
8
9
10 def profile(request):
11     """
12     A view to show the basket page for
13     the customer
14     """
15     return render(request, 'profile/profile.html')
```

Settings:

Results:

All clear, no errors found

Wishlist

urls.py

```
1 from django.urls import path
2
3 from . import views
4 # URLs for users to view their order history
5
6 urlpatterns = [
7     path('', views.profile, name='profile'),
8     path('order_history/<order_number>', views.order_history,
9         name='order_history'),
10 ]
11 |
```

Settings:



Results:

All clear, no errors found

views.py

```
1 from django.shortcuts import render, get_object_or_404
2 from django.contrib.auth.decorators import login_required
3 from django.contrib import messages
4
5 from .models import UserProfile
6 from .forms import UserProfileForm
7 from checkout.models import Order
8
9
10 def profile(request):
11     """
12     A view to show the basket page for
13     the customer
14     """
```

Settings:



Results:

All clear, no errors found