

Admin.py

code institute

CI Python Linter

```
45     queryset.update(approved=False)
46     for artwork in queryset:
47         artwork_denied.send(
48             sender=self.__class__, artwork=artwork, request=request
49         )
50
51     deny_artworks.short_description = "Deny selected artworks"
52
53     def delete_queryset(self, request, queryset):
54         """
55             This method is called when admin
56             tries to delete artworks in bulk
57             from the admin interface.
58         """
59         with transaction.atomic():
60             for artwork in queryset:
61                 Auction.objects.filter(artwork=artwork).delete()
62                 artwork.delete()
63
64     def delete_model(self, request, obj):
65         """
66             This method is called when an admin
67             tries to delete a single artwork instance.
68         """
69         with transaction.atomic():
70             Auction.objects.filter(artwork=obj).delete()
71             obj.delete()
72
73
74     admin.site.register(Artwork, ArtworkAdmin)
75 
```

Settings:

🌙 ⚡ ☀️

Results:

All clear, no errors found

Apps.py

code institute

CI Python Linter

```
1 import sys
2 from django.apps import AppConfig
3 from django.utils import timezone
4
5 """
6 Config class
7 """
8
9
10 class UptowngalleryConfig(AppConfig):
11     name = "uptowngallery"
12 
```

Settings:

🌙 ⚡ ☀️

Results:

All clear, no errors found

Forms.py

CI Python Linter

Settings:

Results:
All clear, no errors found

```
1 from django import forms
2 from .models import Artwork, Auction, Bids
3 import cloudinary.uploader
4 from decimal import Decimal, InvalidOperation
5 from allauth.account.forms import SignupForm
6 from django.contrib.auth import get_user_model
7 from .models import UserProfile
8 from django.core.exceptions import ValidationError
9
10
11 class ArtworkCreateForm(forms.ModelForm):
12     """
13     ArtworkCreateForm:
14     Django ModelForm for creating & validating
15     artwork forms for auctions.
16     Auction Duration Choices:
17     Offers 3, 5, or 7 days as auction duration options.
18     Fields:
19     Includes fields from Artwork model
20     except 'approved', 'auction_start'.
21     Custom Fields:
22     'auction_duration'
23     with dropdown choices.
24     Validation:
25     clean_reserve_price:
26     Validates and formats reserve
27     price as a positive decimal.
28     Initialization __init__:
29     Customizes 'reserve_price'
30     field with specific attributes.
31     Save Method:
32 
```

Models.py

CI Python Linter

Settings:

Results:
All clear, no errors found

```
1 from datetime import timedelta
2 from django.contrib.auth.models import User
3 from django.core.exceptions import ValidationError
4 from django.db import models
5 from django.db.models.signals import post_save
6 from django.dispatch import receiver
7 from django.utils import timezone
8 from cloudinary.models import CloudinaryField
9
10
11 class UserProfile(models.Model):
12     """
13     Represents a user profile,
14     extending the built-in User model.
15     Includes additional fields
16     like name and shipping address.
17     """
18
19     user = models.OneToOneField(
20         User,
21         on_delete=models.CASCADE,
22         blank=True,
23         related_name="profile",
24         verbose_name="User",
25         help_text="Format: Required, Unique",
26     )
27     name = models.CharField(
28         max_length=255,
29         null=True,
30         verbose_name="Name",
31         help_text="The name of the user.", 
```

Signals.py



CI Python Linter

```
1 import logging
2 from django.contrib.auth import get_user_model
3 from django.core.mail import send_mail
4 from django.db.models.signals import post_save
5 from django.dispatch import receiver, Signal
6 from allauth.account.signals import user_signed_up
7 from uptowngallery.models import UserProfile, Artwork
8
9 User = get_user_model()
10 logger = logging.getLogger(__name__)
11 artwork_approved = Signal()
12 artwork_denied = Signal()
13 auction_closed = Signal()
14 bid_placed = Signal()
15 profile_updated = Signal()
16
17 @receiver(user_signed_up)
18 def create_user_profile(sender, request, user, **kwargs):
19     """
20         Signal receiver to create a
21         UserProfile instance when a
22         new user signs up.
23         Logs the creation process
24         and handles any exceptions.
25     """
26     try:
27         logger.info("User created: Creating UserProfile")
28         profile, created = UserProfile.objects.get_or_create(user=user)
29
30         if created:
31             pass
```

Settings:



Results:

All clear, no errors found

Tests.py



CI Python Linter

```
1 from datetime import timedelta
2 from unittest.mock import patch
3 from django.contrib.admin.sites import AdminSite
4 from django.contrib.auth import get_user_model
5 from django.contrib.auth.models import (
6     User,
7 )
8 from django.contrib.messages import get_messages
9 from django.contrib.sessions.middleware import SessionMiddleware
10 from django.core import mail
11 from django.core.exceptions import ValidationError
12 from django.core.files.uploadedfile import SimpleUploadedFile
13 from django.test import TestCase, RequestFactory, Client
14 from django.test.utils import override_settings
15 from django.urls import reverse
16 from django.utils import timezone
17 from .admin import ArtworkAdmin
18 from .forms import CustomSignupForm, ArtworkCreateForm, BidForm
19 from .models import Artwork, UserProfile, Auction, Bids
20 from .signals import user_signed_up, auction_closed, bid_place
21
22
23 class UserProfileModelTest(TestCase):
24     def setup(self):
25         self.user = User.objects.create_user(
26             username="testuser", password="12345"
27         )
28         self.user_profile = UserProfile.objects.create(
29             user=self.user,
30             name="Test User",
31             shipping_address="123 Test St",
```

Settings:



Results:

All clear, no errors found

Urls.py App Level



CI Python Linter

```
1 from django.urls import path
2 from .views import (
3     LandingPageView,
4     ArtworkListView,
5     CreateArtworkView,
6     PendingArtworksView,
7     AuctionDetailView,
8     ProfileInfoView,
9     signup_view,
10    UpdateProfileView,
11    ActivityDashboardView,
12    AboutView,
13    SearchActiveAuctionArtworkView,
14    EditArtworkView,
15 )
16
17 urlpatterns = [
18     path("", LandingPageView.as_view(), name="home"),
19     path("artworks/", ArtworkListView.as_view(), name="artwork_list"),
20     path("create/", CreateArtworkView.as_view(), name="create_artwork"),
21     path(
22         "pending_artworks/",
23         PendingArtworksView.as_view(),
24         name="pending_artworks",
25     ),
26     path(
27         "auction_detail/<int:artwork_id>/<int:auction_id>/",
28         AuctionDetailView.as_view(),
29         name="auction_detail",
30     ),
31     path("signup/", signup_view, name="account_signup"),
32 ]
```

Settings:



Results:

All clear, no errors found

Urls.py Project Level



CI Python Linter

```
1 from django.contrib import admin
2 from django.urls import path, include
3 from django.conf import settings
4
5
6 urlpatterns = [
7     path("admin/", admin.site.urls),
8     path("", include("uptowngallery.urls"), name="uptowngallery_urls"),
9     path("accounts/", include("allauth.urls")),
10 ]
11
12 handler404 = "uptowngallery.views.handler404"
13 handler500 = "uptowngallery.views.handler500"
14 ]
```

Settings:



Results:

All clear, no errors found

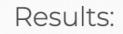
Views.py



CI Python Linter

```
1 from django.db import transaction
2 from django.db.models import Q, Max, Prefetch
3 from django.http import JsonResponse
4 from django.shortcuts import get_object_or_404, redirect, render
5 from django.urls import reverse_lazy
6 from django.utils import timezone
7 from django.core.paginator import Paginator, EmptyPage, PageNotAnInteger
8 from django.views import View
9 from django.views.generic import TemplateView
10 from django.views.generic.edit import CreateView
11 from django.contrib import messages
12 from django.contrib.auth.mixins import LoginRequiredMixin
13 from django.middleware.csrf import get_token
14 from .forms import (
15     ArtworkCreateForm,
16     CustomSignupForm,
17     BidForm,
18     ArtworkEditForm,
19 )
20 from .models import Artwork, Auction, Bids
21 from .signals import bid_placed, profile_updated
22
23
24 class LandingPageView(View):
25     """
26         Retrieve 10 most recent artworks to display
27     """
28
29     def get(self, request):
30         recent_artworks = Artwork.objects.filter(
31             approved=True
32     )
```

Settings:



Results:

All clear, no errors found