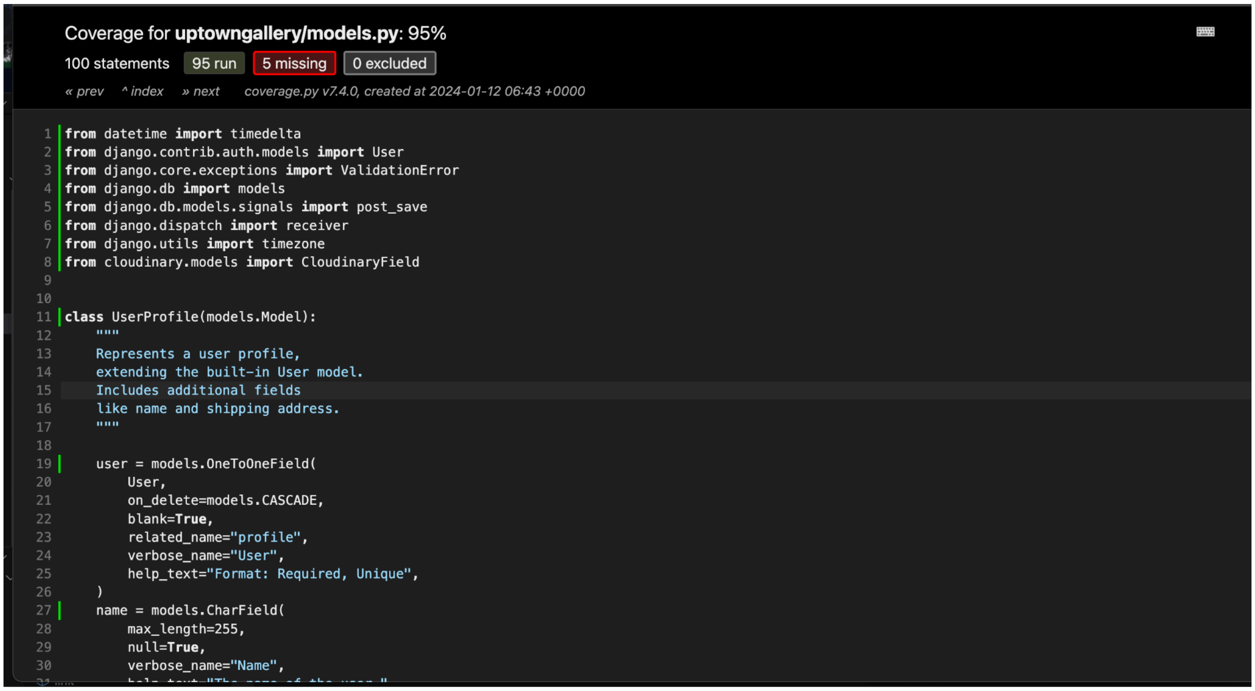


Coverage for **uptowngallery/models.py**: 95%

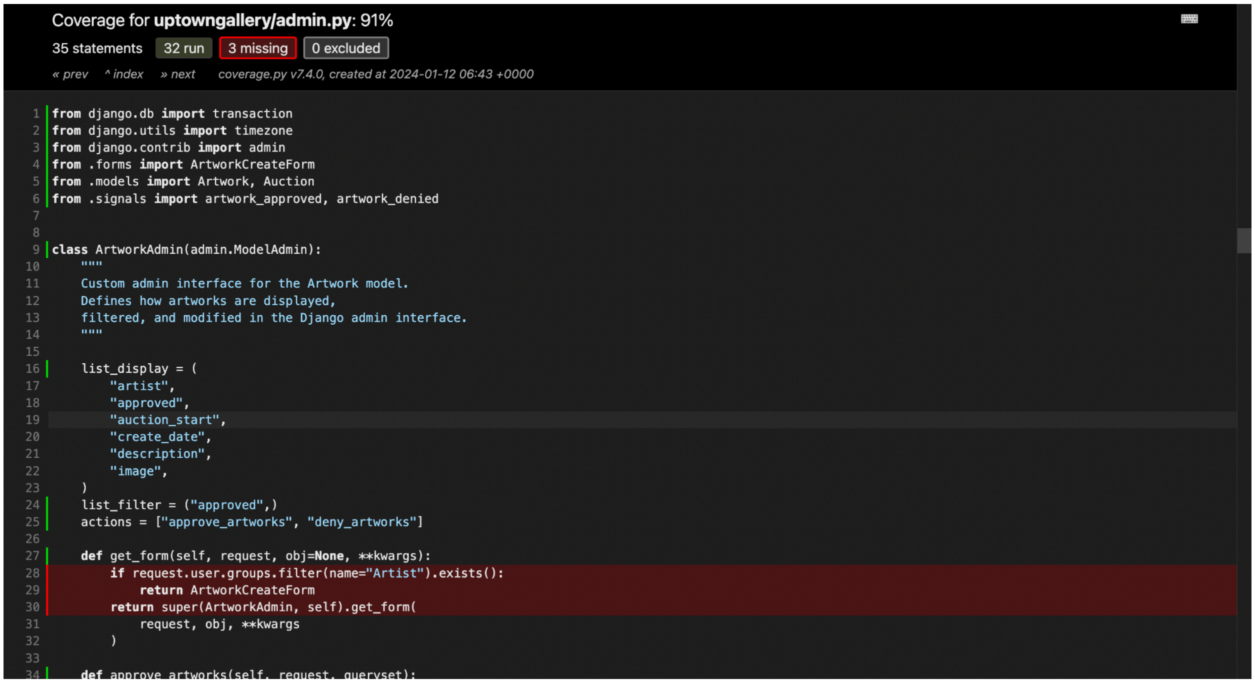
100 statements 95 run 5 missing 0 excluded

« prev ^ index » next coverage.py v7.4.0, created at 2024-01-12 06:43 +0000

```python
from datetime import timedelta
from django.contrib.auth.models import User
from django.core.exceptions import ValidationError
from django.db import models
from django.db.models.signals import post_save
from django.dispatch import receiver
from django.utils import timezone
from cloudinary.models import CloudinaryField


class UserProfile(models.Model):
    """
    Represents a user profile,
    extending the built-in User model.
    Includes additional fields
    like name and shipping address.
    """

    user = models.OneToOneField(
        User,
        on_delete=models.CASCADE,
        blank=True,
        related_name="profile",
        verbose_name="User",
        help_text="Format: Required, Unique",
    )
    name = models.CharField(
        max_length=255,
        null=True,
        verbose_name="Name",
        help_text="The name of the user.",
```



Coverage for **uptowngallery/admin.py**: 91%

35 statements 32 run 3 missing 0 excluded

« prev ^ index » next coverage.py v7.4.0, created at 2024-01-12 06:43 +0000

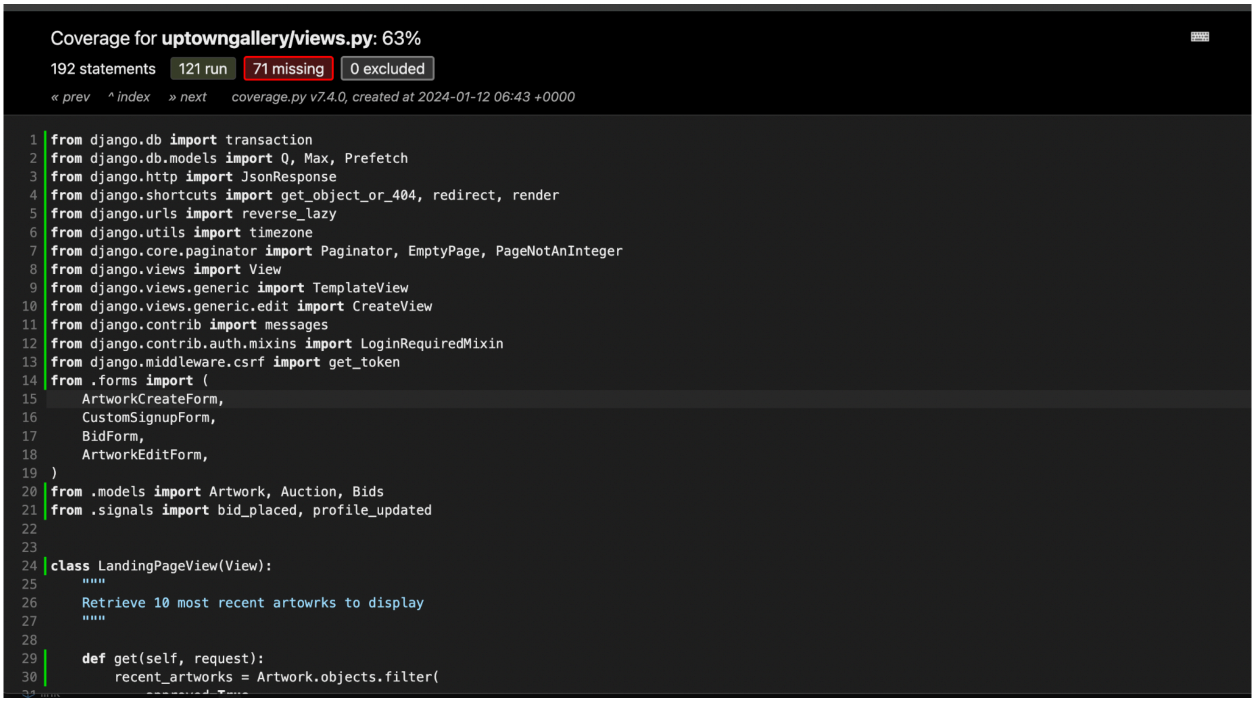
```python
from django.db import transaction
from django.utils import timezone
from django.contrib import admin
from .forms import ArtworkCreateForm
from .models import Artwork, Auction
from .signals import artwork_approved, artwork_denied


class ArtworkAdmin(admin.ModelAdmin):
    """
    Custom admin interface for the Artwork model.
    Defines how artworks are displayed,
    filtered, and modified in the Django admin interface.
    """

    list_display = (
        "artist",
        "approved",
        "auction_start",
        "create_date",
        "description",
        "image",
    )
    list_filter = ("approved",)
    actions = ["approve_artworks", "deny_artworks"]

    def get_form(self, request, obj=None, **kwargs):
        if request.user.groups.filter(name="Artist").exists():
            return ArtworkCreateForm
        return super(ArtworkAdmin, self).get_form(
            request, obj, **kwargs
        )

    def approve_artworks(self, request, queryset):
```



Coverage for **uptowngallery/views.py**: 63%

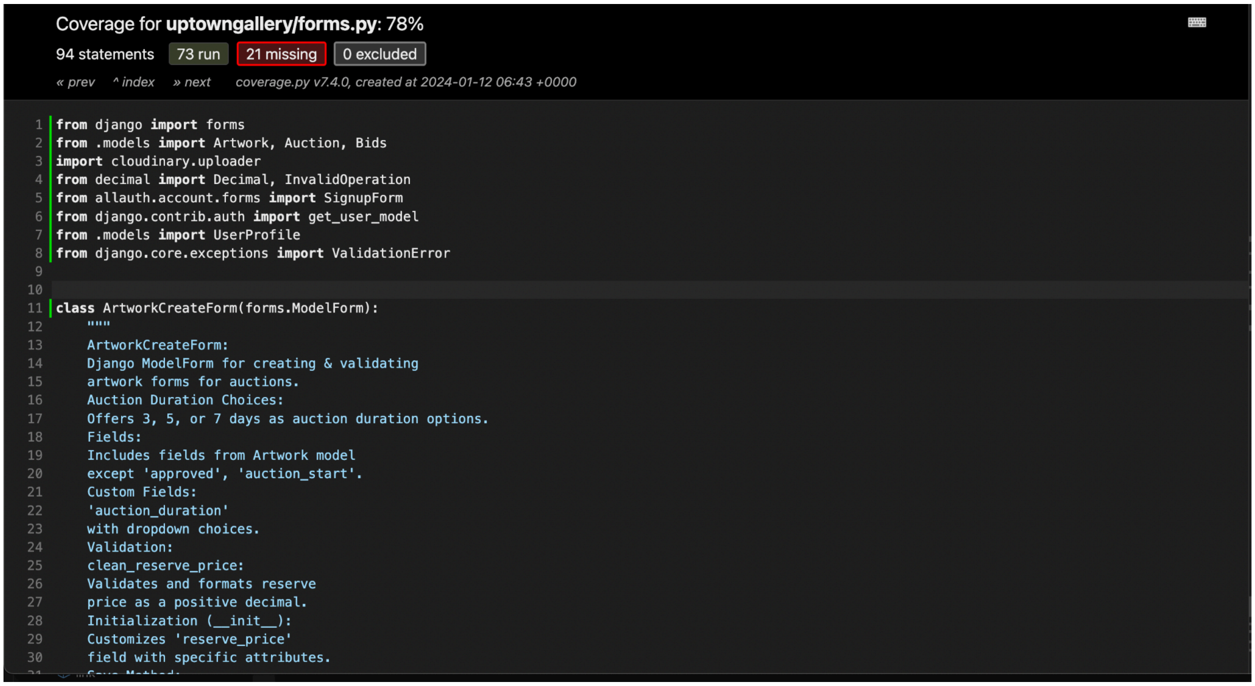
192 statements 121 run 71 missing 0 excluded

« prev ^ index » next *coverage.py v7.4.0, created at 2024-01-12 06:43 +0000*

```python
from django.db import transaction
from django.db.models import Q, Max, Prefetch
from django.http import JsonResponse
from django.shortcuts import get_object_or_404, redirect, render
from django.urls import reverse_lazy
from django.utils import timezone
from django.core.paginator import Paginator, EmptyPage, PageNotAnInteger
from django.views import View
from django.views.generic import TemplateView
from django.views.generic.edit import CreateView
from django.contrib import messages
from django.contrib.auth.mixins import LoginRequiredMixin
from django.middleware.csrf import get_token
from .forms import (
    ArtworkCreateForm,
    CustomSignupForm,
    BidForm,
    ArtworkEditForm,
)
from .models import Artwork, Auction, Bids
from .signals import bid_placed, profile_updated


class LandingPageView(View):
    """
    Retrieve 10 most recent artowrks to display
    """

    def get(self, request):
        recent_artworks = Artwork.objects.filter(
            approved=True
```



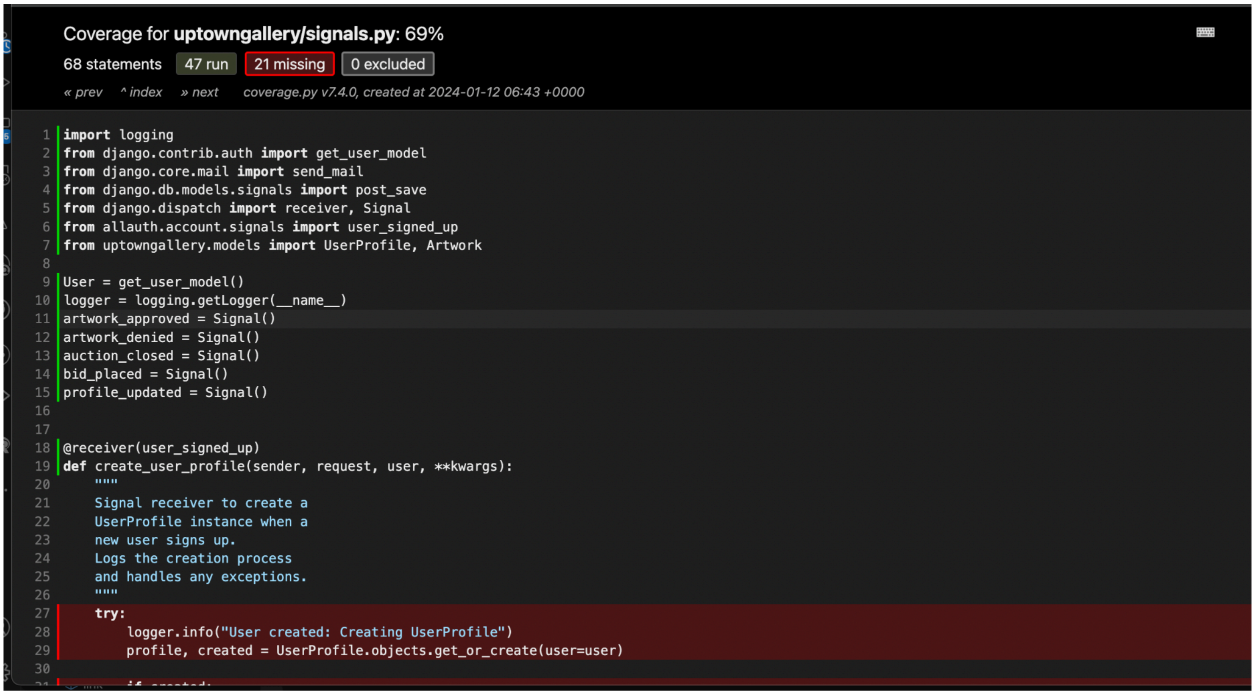
Coverage for **uptowngallery/forms.py**: 78%

94 statements | 73 run | 21 missing | 0 excluded

« prev ^ index » next *coverage.py v7.4.0, created at 2024-01-12 06:43 +0000*

```python
from django import forms
from .models import Artwork, Auction, Bids
import cloudinary.uploader
from decimal import Decimal, InvalidOperation
from allauth.account.forms import SignupForm
from django.contrib.auth import get_user_model
from .models import UserProfile
from django.core.exceptions import ValidationError


class ArtworkCreateForm(forms.ModelForm):
    """
    ArtworkCreateForm:
    Django ModelForm for creating & validating
    artwork forms for auctions.
    Auction Duration Choices:
    Offers 3, 5, or 7 days as auction duration options.
    Fields:
    Includes fields from Artwork model
    except 'approved', 'auction_start'.
    Custom Fields:
    'auction_duration'
    with dropdown choices.
    Validation:
    clean_reserve_price:
    Validates and formats reserve
    price as a positive decimal.
    Initialization (__init__):
    Customizes 'reserve_price'
    field with specific attributes.
    Save Method:
```



Coverage for **uptowngallery/signals.py**: 69%

68 statements | 47 run | 21 missing | 0 excluded

« prev ^ index » next *coverage.py v7.4.0, created at 2024-01-12 06:43 +0000*

```python
import logging
from django.contrib.auth import get_user_model
from django.core.mail import send_mail
from django.db.models.signals import post_save
from django.dispatch import receiver, Signal
from allauth.account.signals import user_signed_up
from uptowngallery.models import UserProfile, Artwork

User = get_user_model()
logger = logging.getLogger(__name__)
artwork_approved = Signal()
artwork_denied = Signal()
auction_closed = Signal()
bid_placed = Signal()
profile_updated = Signal()


@receiver(user_signed_up)
def create_user_profile(sender, request, user, **kwargs):
    """
    Signal receiver to create a
    UserProfile instance when a
    new user signs up.
    Logs the creation process
    and handles any exceptions.
    """
    try:
        logger.info("User created: Creating UserProfile")
        profile, created = UserProfile.objects.get_or_create(user=user)

        if created:
```