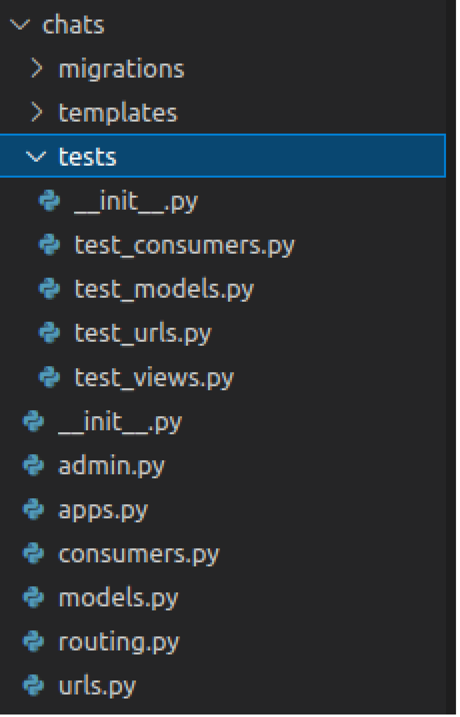
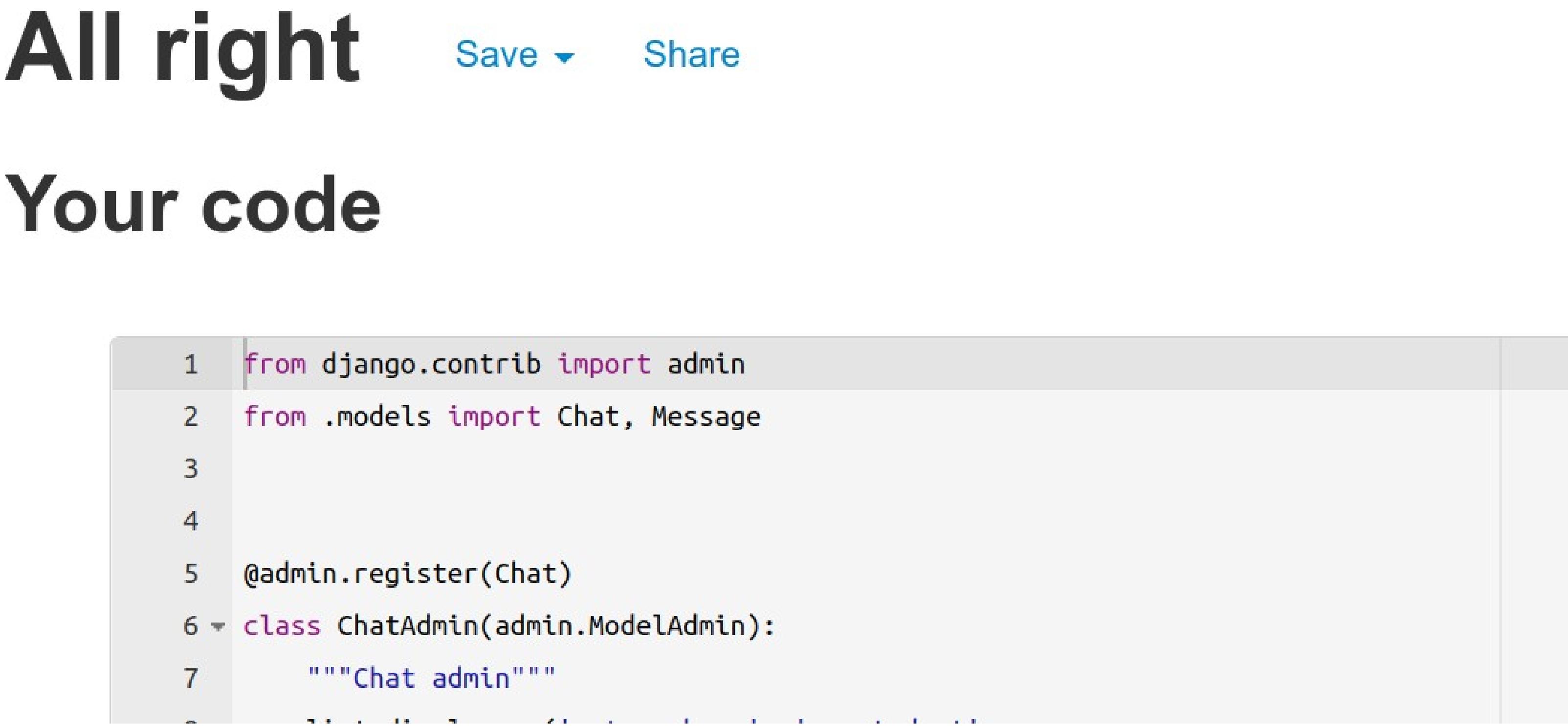
admin.py



```
chats
  migrations
  templates
  tests
    __init__.py
    test_consumers.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  consumers.py
  models.py
  routing.py
  urls.py
```

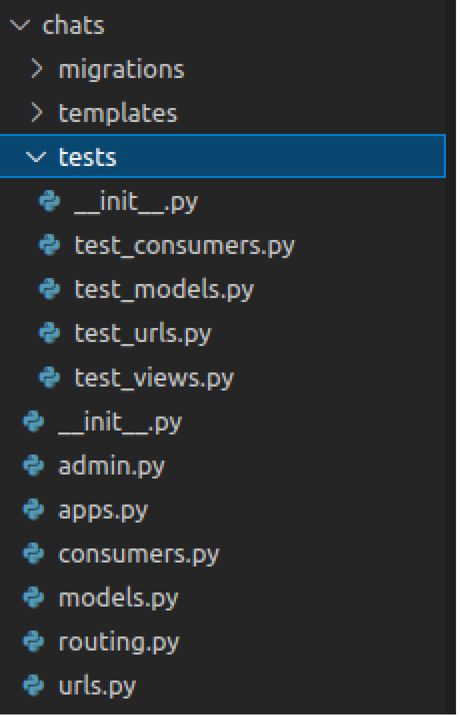


All right

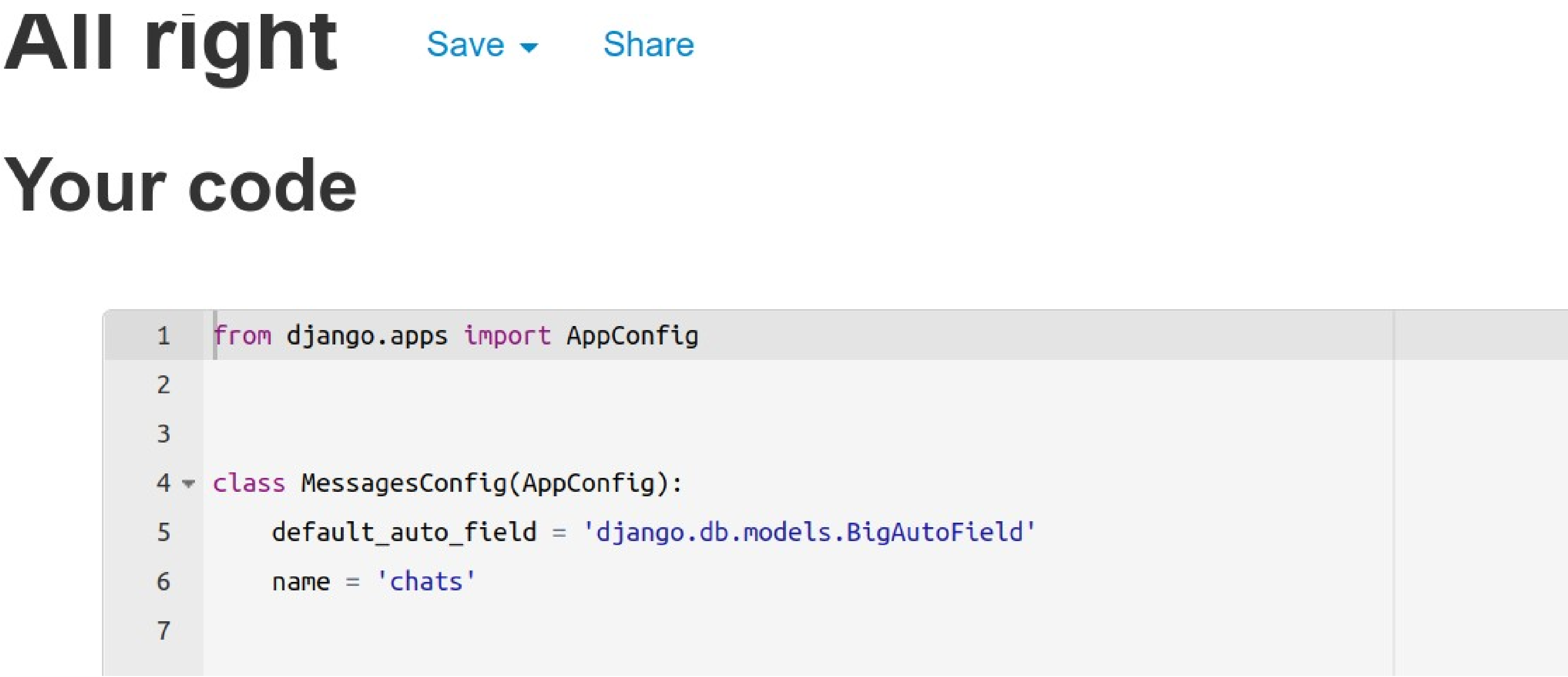
Your code

```python
from django.contrib import admin
from .models import Chat, Message


@admin.register(Chat)
class ChatAdmin(admin.ModelAdmin):
    """Chat admin"""
```

apps.py



```
chats
  migrations
  templates
  tests
    __init__.py
    test_consumers.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  consumers.py
  models.py
  routing.py
  urls.py
```

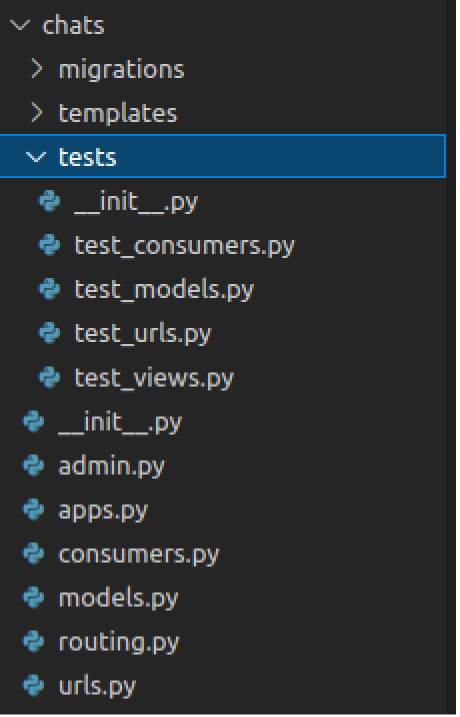


All right

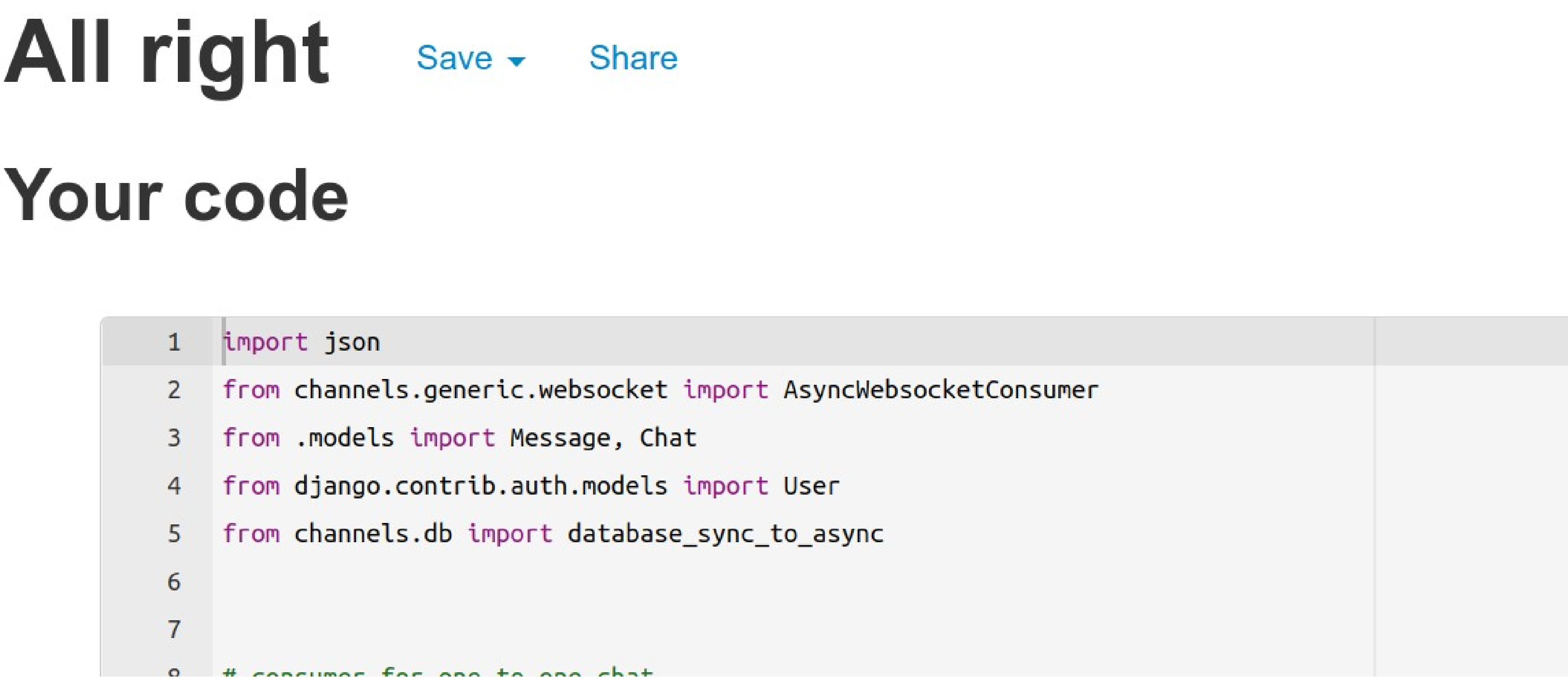
Your code

```python
from django.apps import AppConfig


class MessagesConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'chats'

```

consumers.py



```
chats
  migrations
  templates
  tests
    __init__.py
    test_consumers.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  consumers.py
  models.py
  routing.py
  urls.py
```

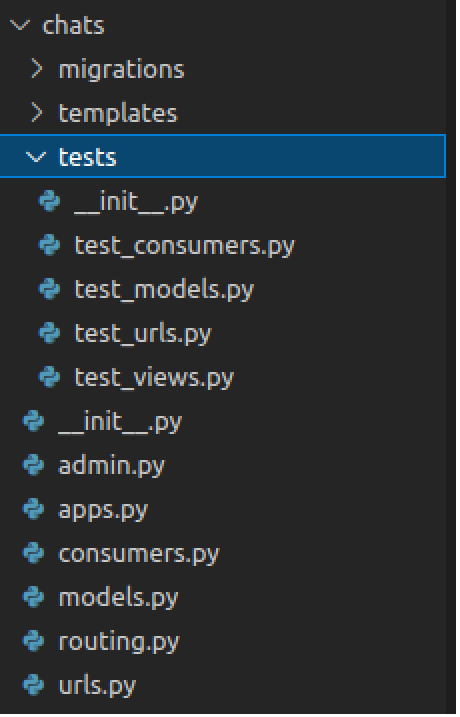


All right

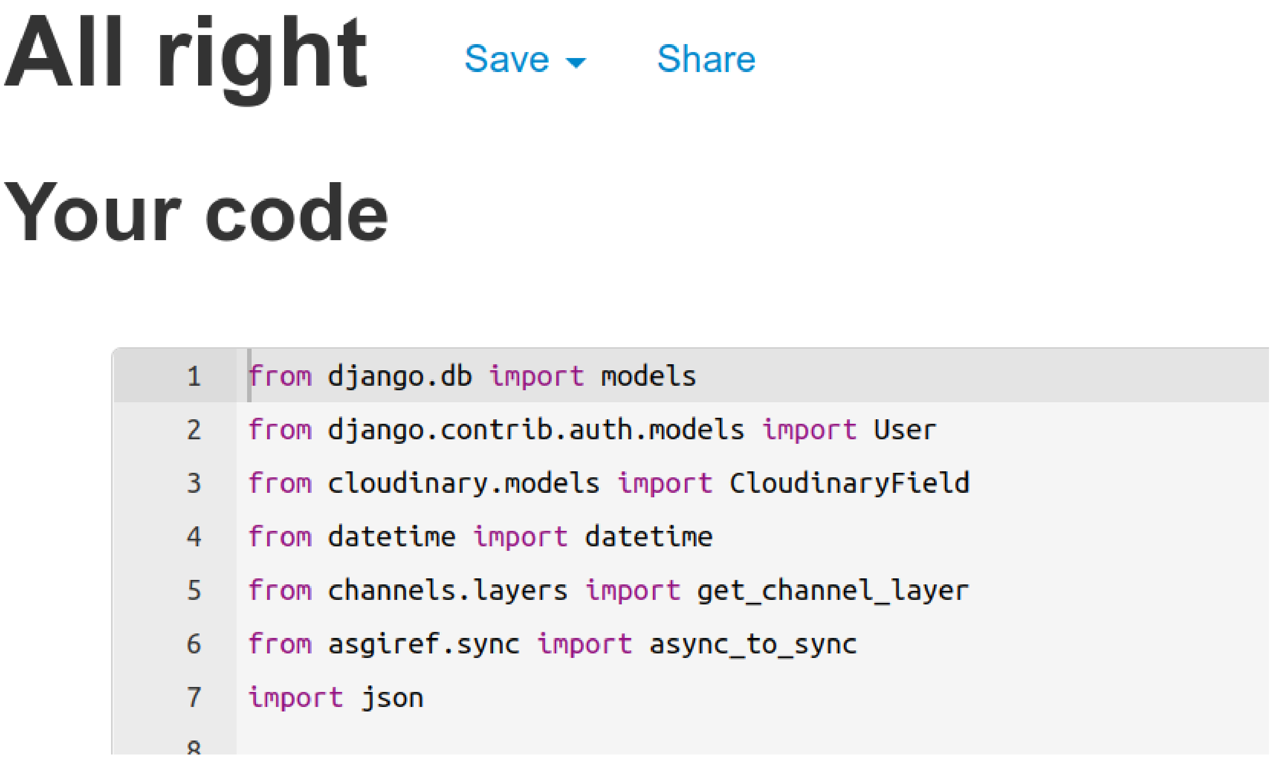
Your code

```python
import json
from channels.generic.websocket import AsyncWebsocketConsumer
from .models import Message, Chat
from django.contrib.auth.models import User
from channels.db import database_sync_to_async


# consumer for one to one chat
```

models.py



```
chats
  migrations
  templates
  tests
    __init__.py
    test_consumers.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  consumers.py
  models.py
  routing.py
  urls.py
```

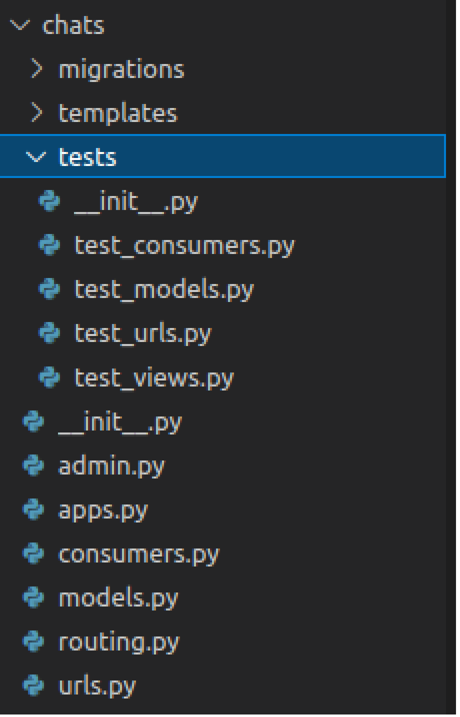


All right

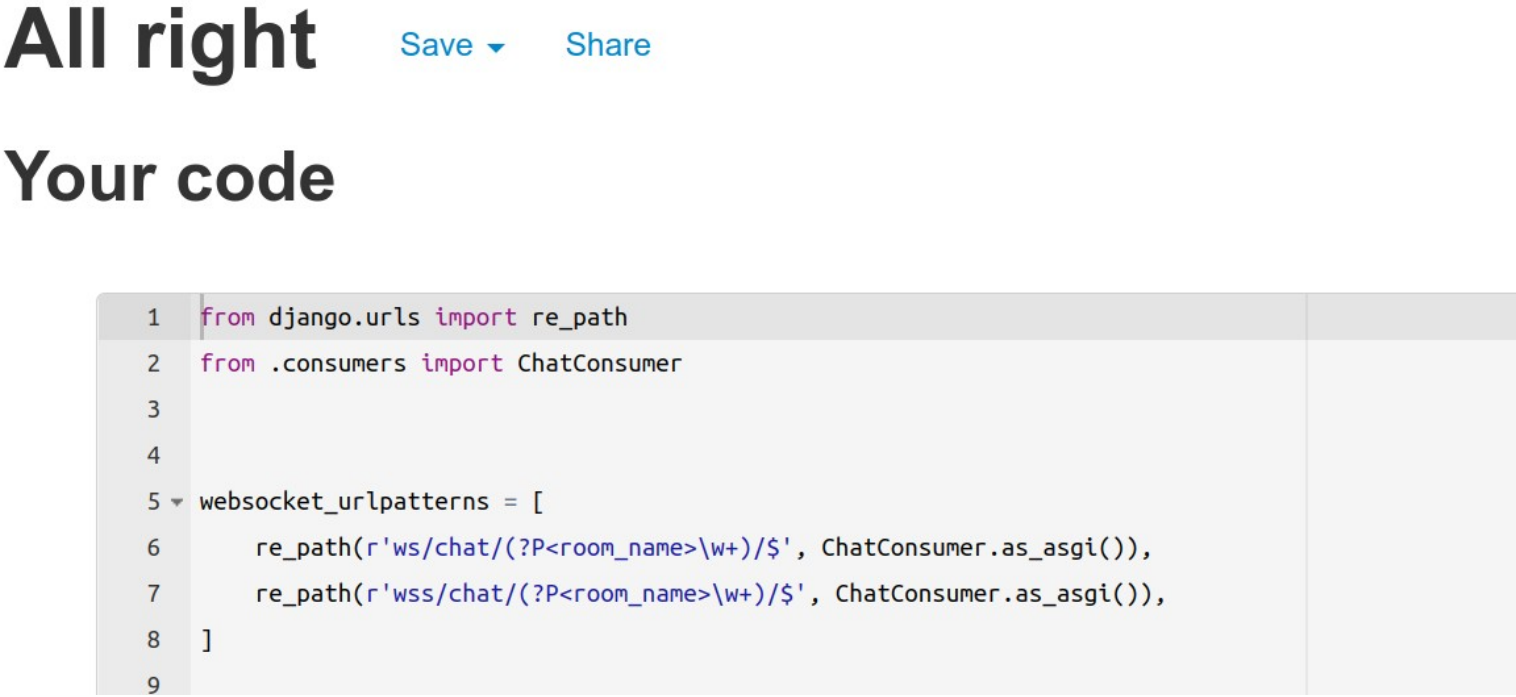
Your code

```python
from django.db import models
from django.contrib.auth.models import User
from cloudinary.models import CloudinaryField
from datetime import datetime
from channels.layers import get_channel_layer
from asgiref.sync import async_to_sync
import json
```

routing.py



```
chats
  migrations
  templates
  tests
    __init__.py
    test_consumers.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  consumers.py
  models.py
  routing.py
  urls.py
```

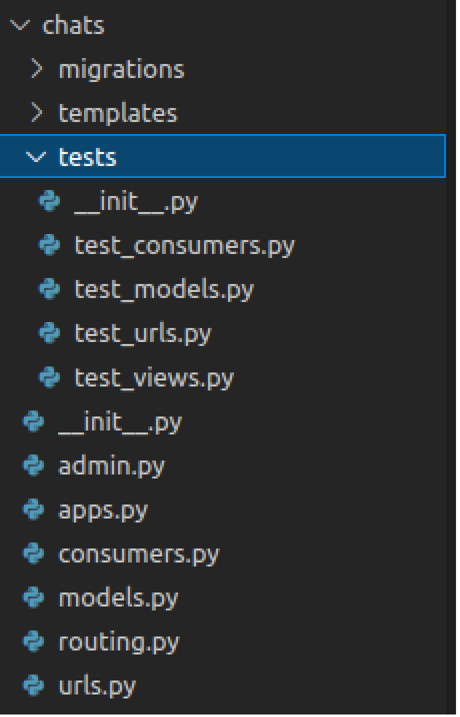


All right

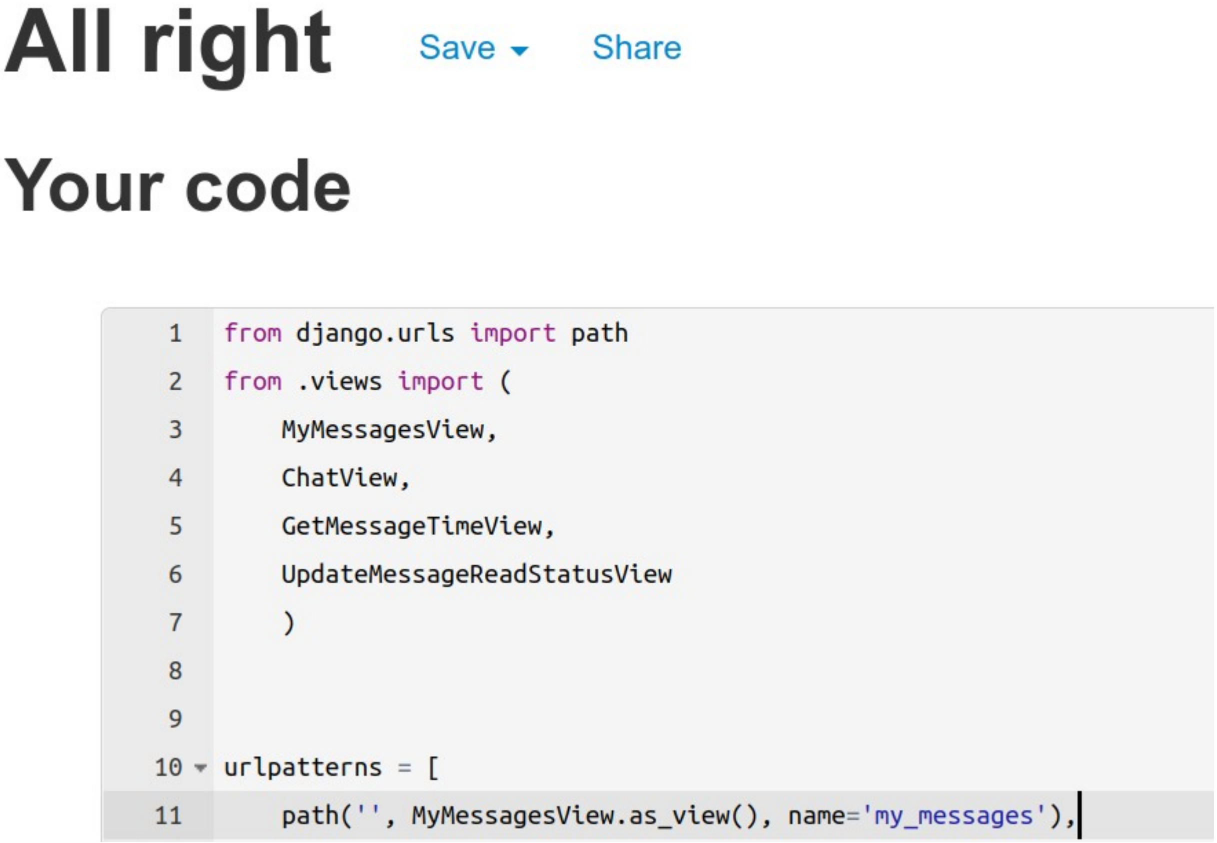
Your code

```
from django.urls import re_path
from .consumers import ChatConsumer


websocket_urlpatterns = [
    re_path(r'ws/chat/(?P<room_name>\w+)/$', ChatConsumer.as_asgi()),
    re_path(r'wss/chat/(?P<room_name>\w+)/$', ChatConsumer.as_asgi()),
]

```

urls.py



```
chats
  migrations
  templates
  tests
    __init__.py
    test_consumers.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  consumers.py
  models.py
  routing.py
  urls.py
```

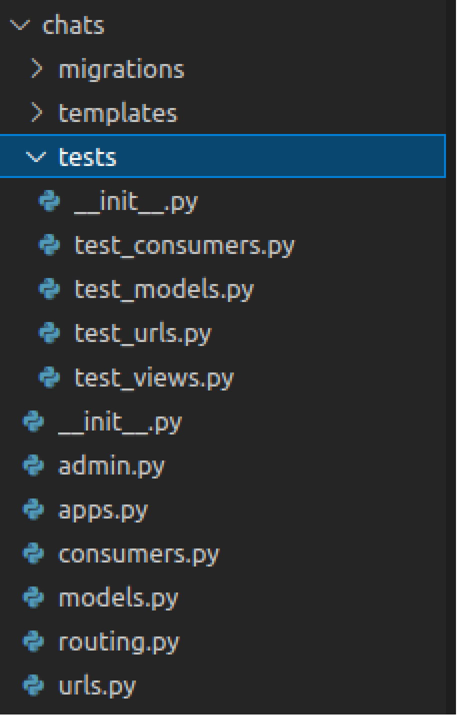


All right

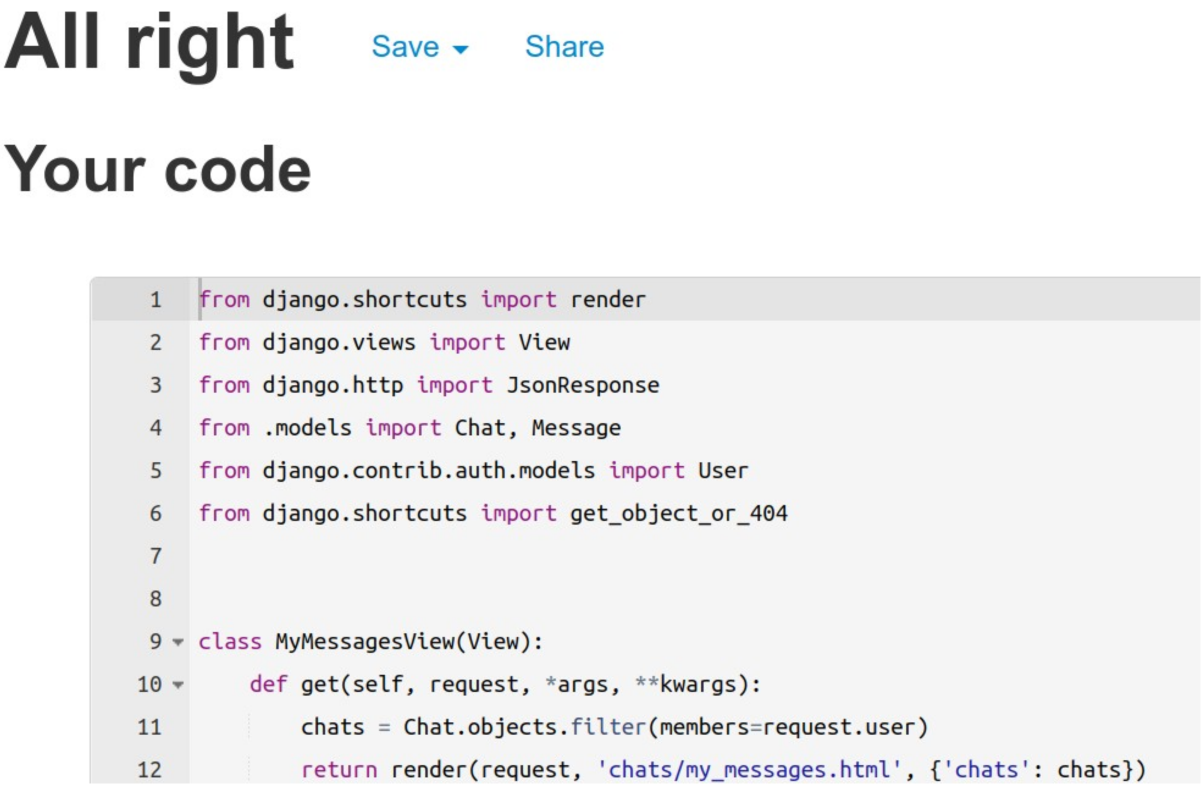
Your code

```python
from django.urls import path
from .views import (
    MyMessagesView,
    ChatView,
    GetMessageTimeView,
    UpdateMessageReadStatusView
    )


urlpatterns = [
    path('', MyMessagesView.as_view(), name='my_messages'),
```

views.py



```
chats
  migrations
  templates
  tests
    __init__.py
    test_consumers.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  consumers.py
  models.py
  routing.py
  urls.py
```

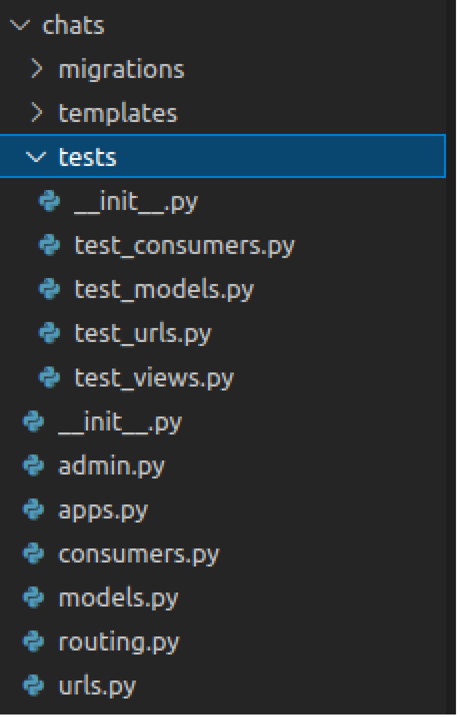


All right

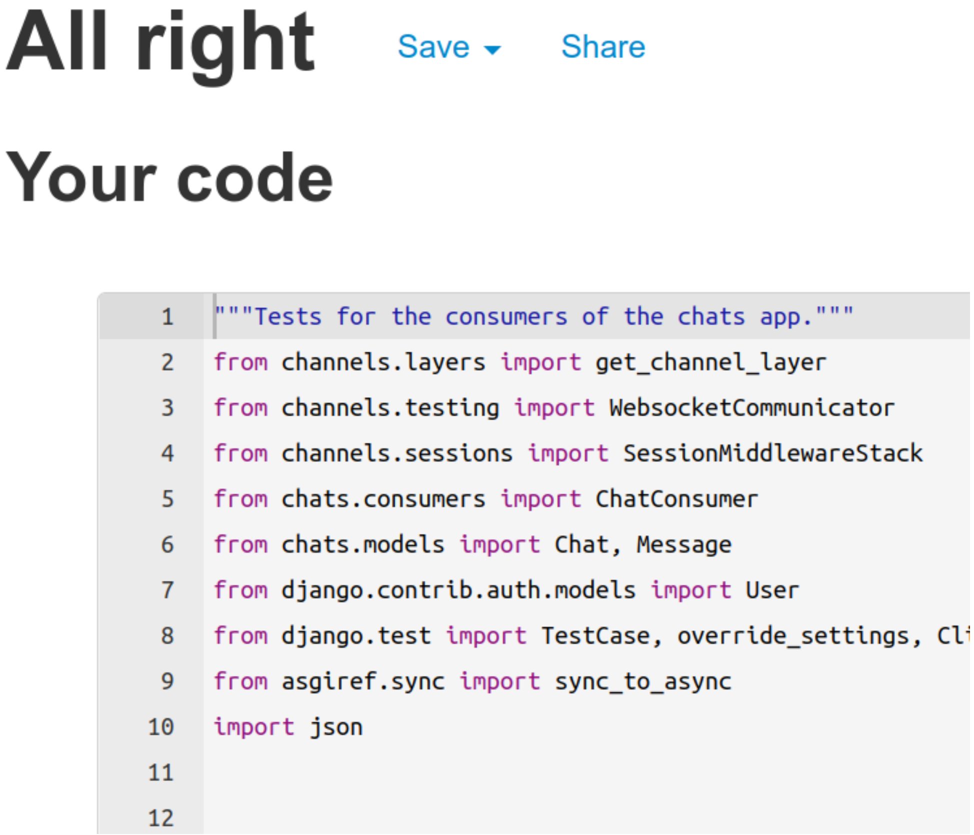
Your code

```python
from django.shortcuts import render
from django.views import View
from django.http import JsonResponse
from .models import Chat, Message
from django.contrib.auth.models import User
from django.shortcuts import get_object_or_404


class MyMessagesView(View):
    def get(self, request, *args, **kwargs):
        chats = Chat.objects.filter(members=request.user)
        return render(request, 'chats/my_messages.html', {'chats': chats})
```

tests/test_consumers.py



```
chats
  migrations
  templates
  tests
    __init__.py
    test_consumers.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  consumers.py
  models.py
  routing.py
  urls.py
```

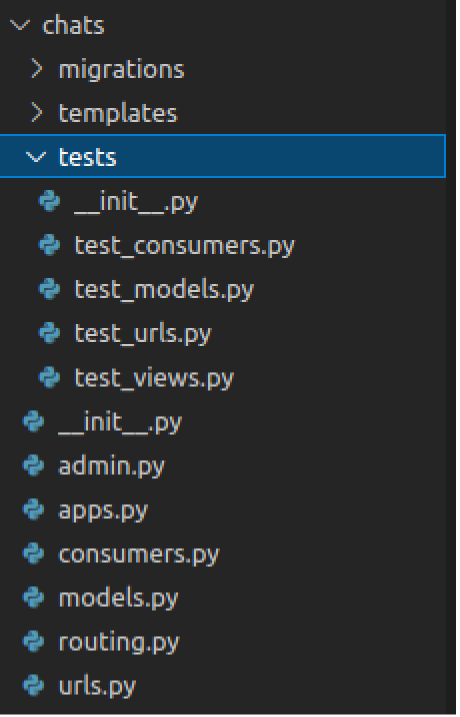


All right

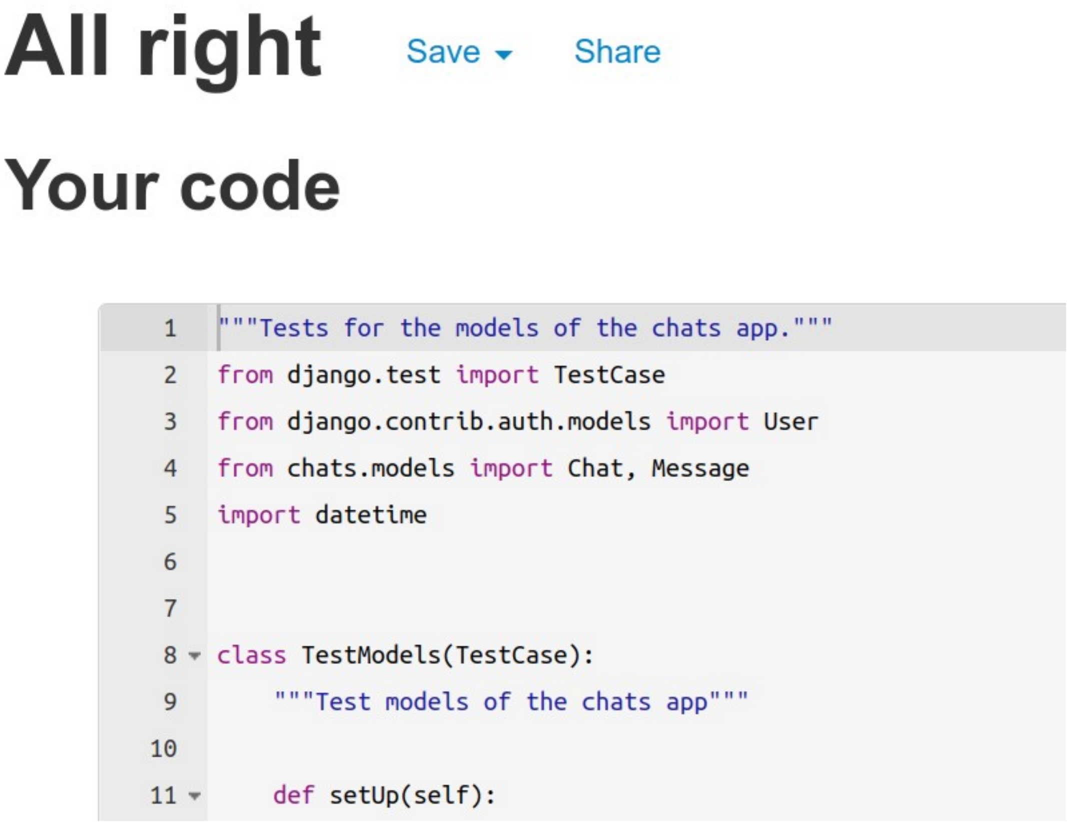
Your code

```python
"""Tests for the consumers of the chats app."""
from channels.layers import get_channel_layer
from channels.testing import WebsocketCommunicator
from channels.sessions import SessionMiddlewareStack
from chats.consumers import ChatConsumer
from chats.models import Chat, Message
from django.contrib.auth.models import User
from django.test import TestCase, override_settings, Cli
from asgiref.sync import sync_to_async
import json


```

tests/test_models.py



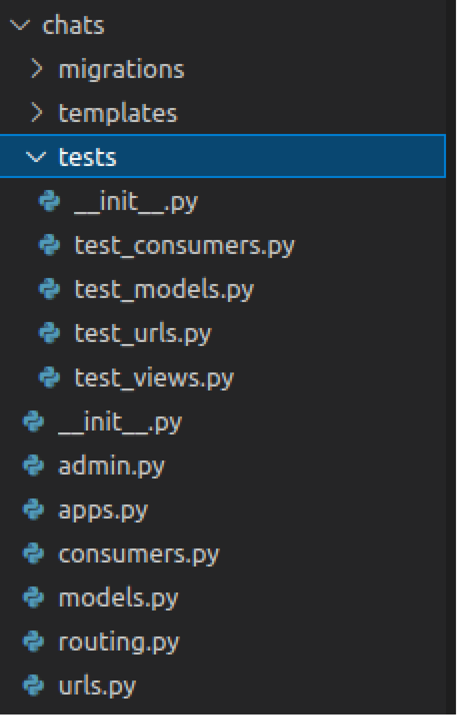
```
chats
  migrations
  templates
  tests
    __init__.py
    test_consumers.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  consumers.py
  models.py
  routing.py
  urls.py
```



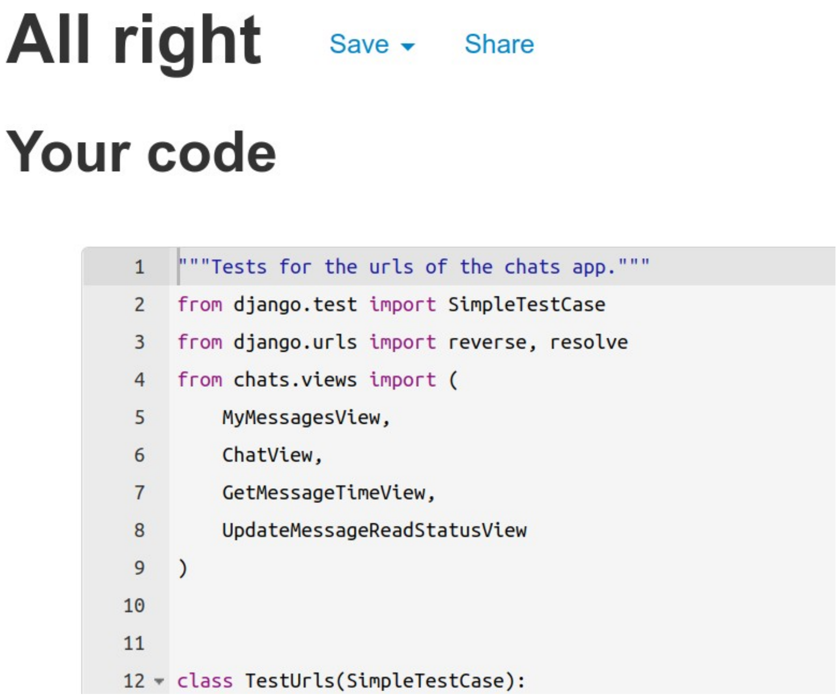
Your code

```python
"""Tests for the models of the chats app."""
from django.test import TestCase
from django.contrib.auth.models import User
from chats.models import Chat, Message
import datetime


class TestModels(TestCase):
    """Test models of the chats app"""

    def setUp(self):
```

tests/test_urls.py



```
chats
  migrations
  templates
  tests
    __init__.py
    test_consumers.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  consumers.py
  models.py
  routing.py
  urls.py
```

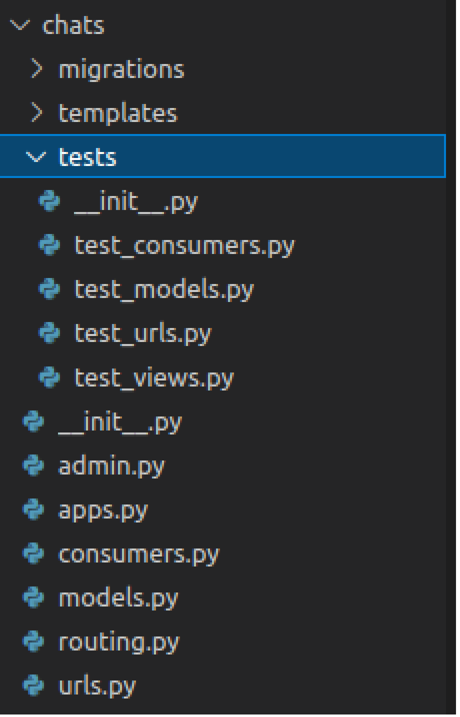


All right

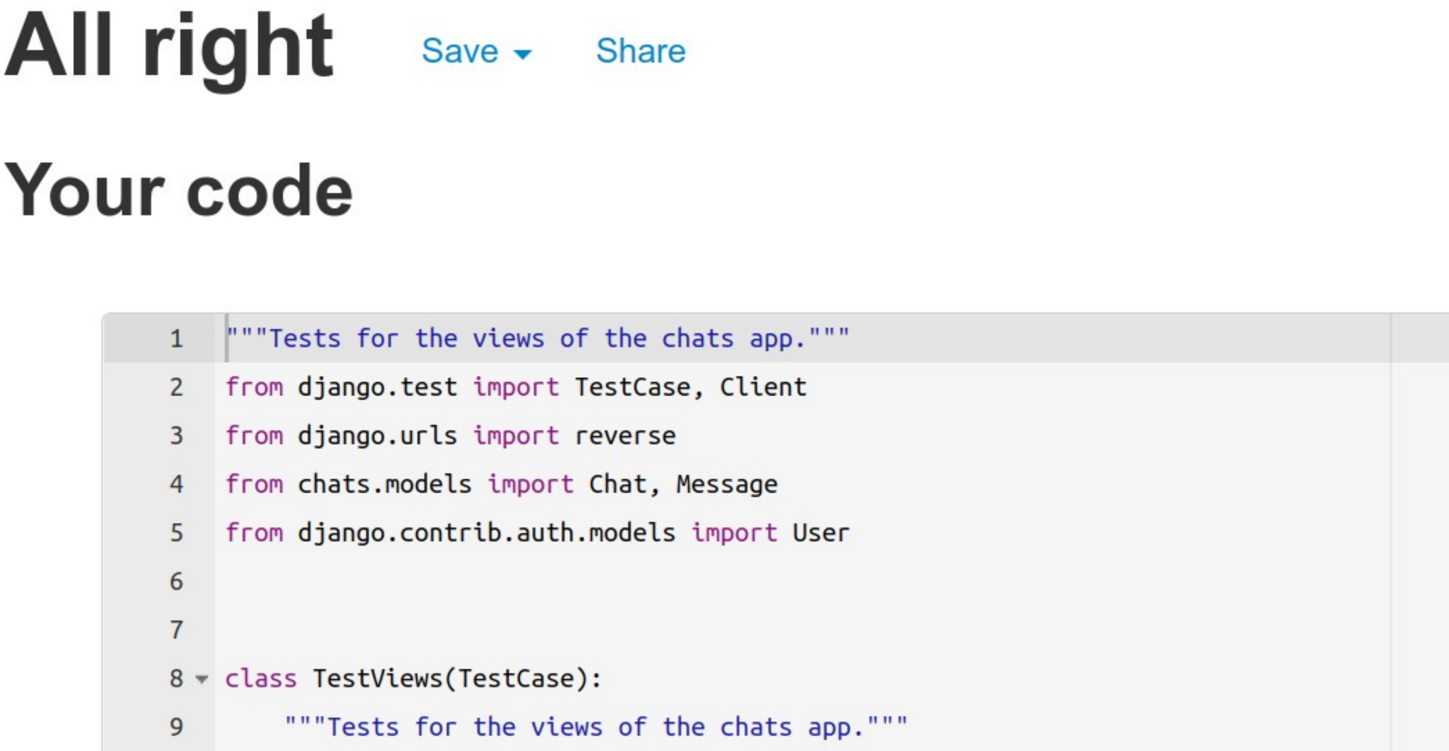
Your code

```python
"""Tests for the urls of the chats app."""
from django.test import SimpleTestCase
from django.urls import reverse, resolve
from chats.views import (
    MyMessagesView,
    ChatView,
    GetMessageTimeView,
    UpdateMessageReadStatusView
)


class TestUrls(SimpleTestCase):
```

tests/test_views.py



```
chats
  migrations
  templates
  tests
    __init__.py
    test_consumers.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  consumers.py
  models.py
  routing.py
  urls.py
```

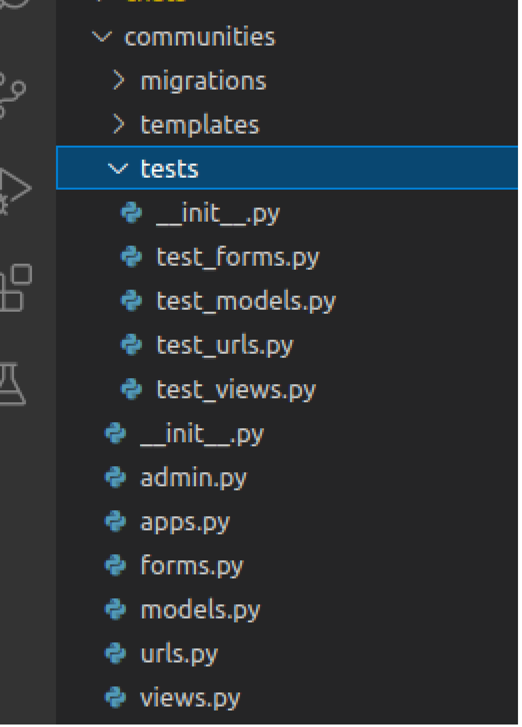


All right

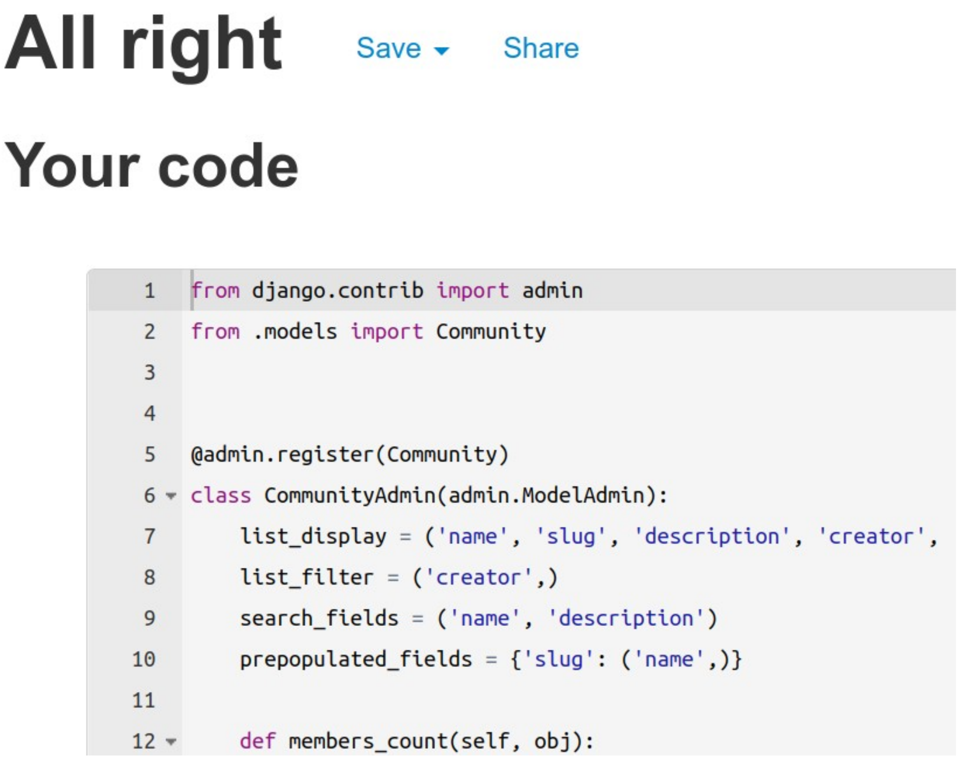
Your code

```python
"""Tests for the views of the chats app."""
from django.test import TestCase, Client
from django.urls import reverse
from chats.models import Chat, Message
from django.contrib.auth.models import User


class TestViews(TestCase):
    """Tests for the views of the chats app."""
```

admin.py



```
communities
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```

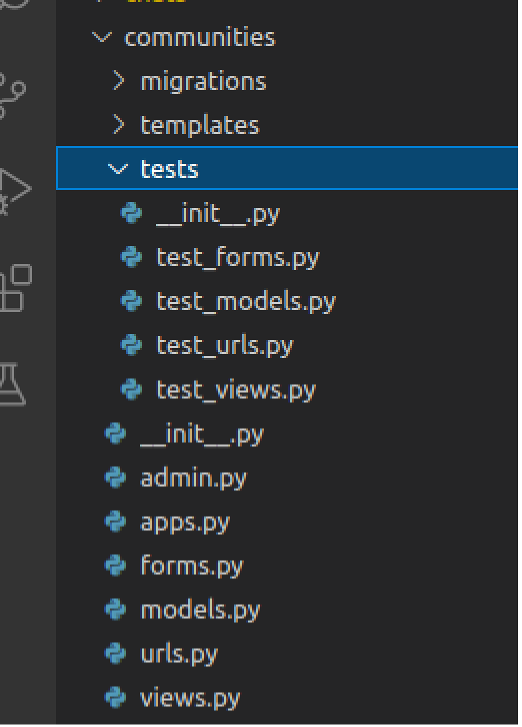


All right

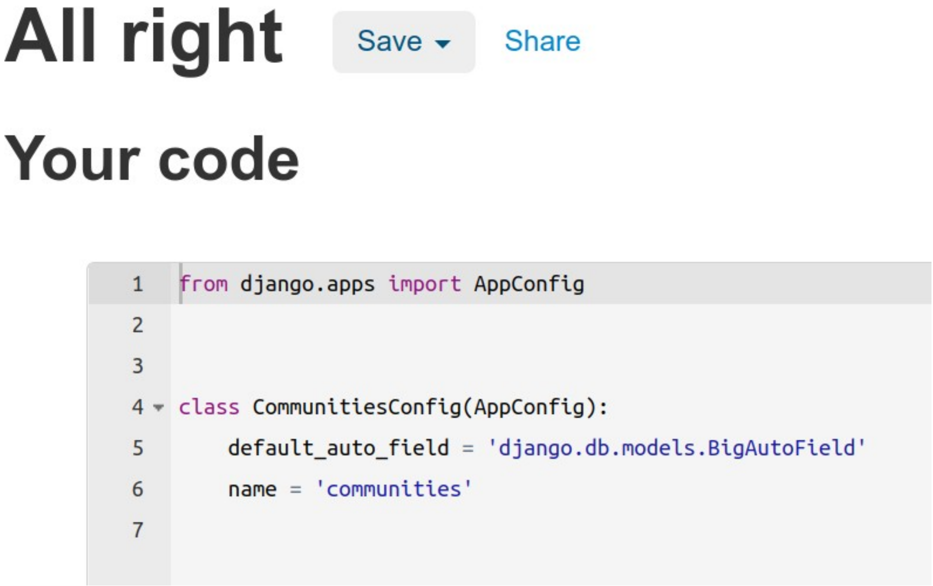
Your code

```python
from django.contrib import admin
from .models import Community


@admin.register(Community)
class CommunityAdmin(admin.ModelAdmin):
    list_display = ('name', 'slug', 'description', 'creator',
    list_filter = ('creator',)
    search_fields = ('name', 'description')
    prepopulated_fields = {'slug': ('name',)}

    def members_count(self, obj):
```

apps.py



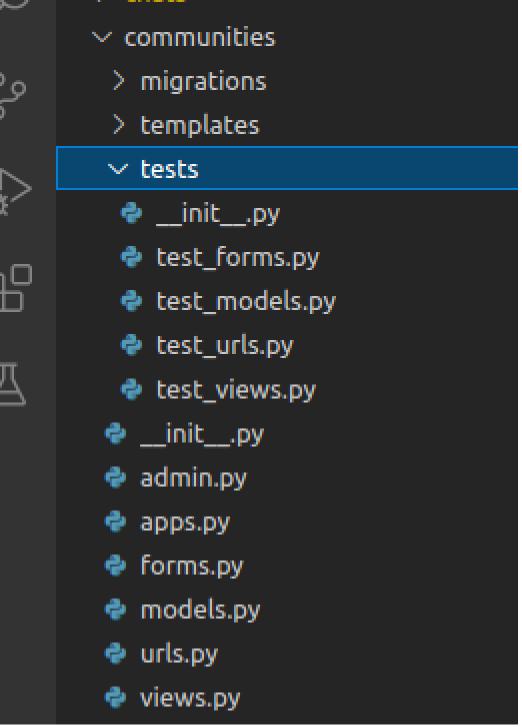
```
communities
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```



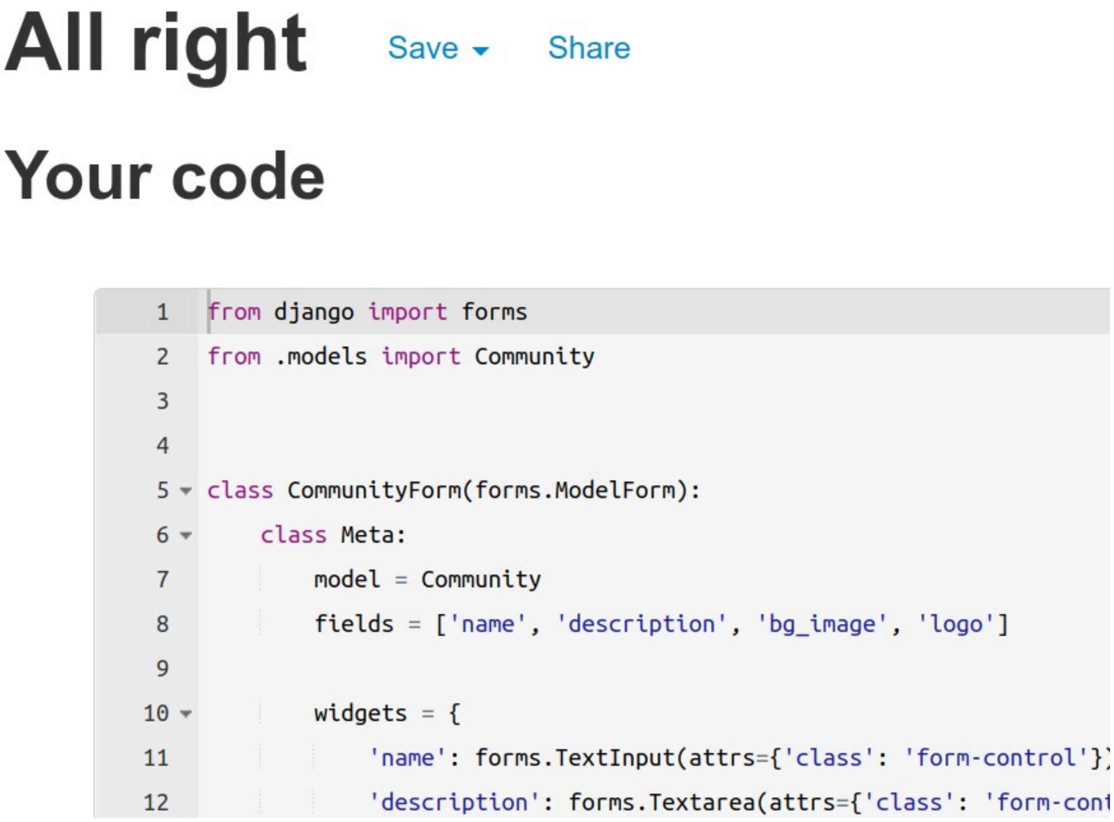
Your code

```python
from django.apps import AppConfig


class CommunitiesConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'communities'

```

forms.py



```
communities
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```

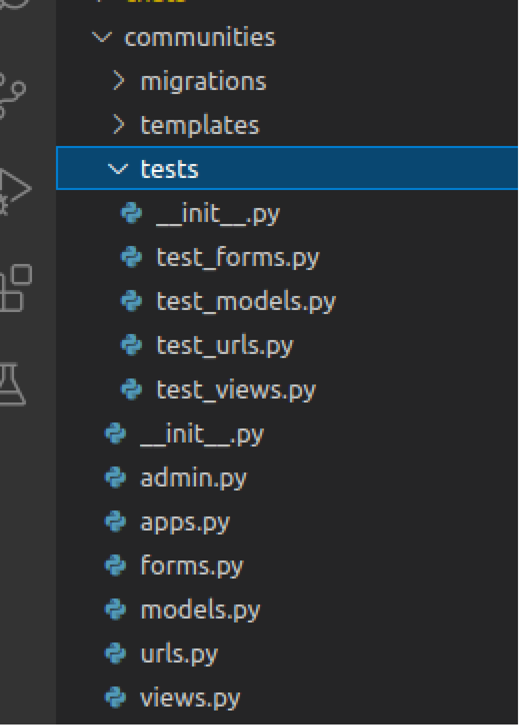


All right

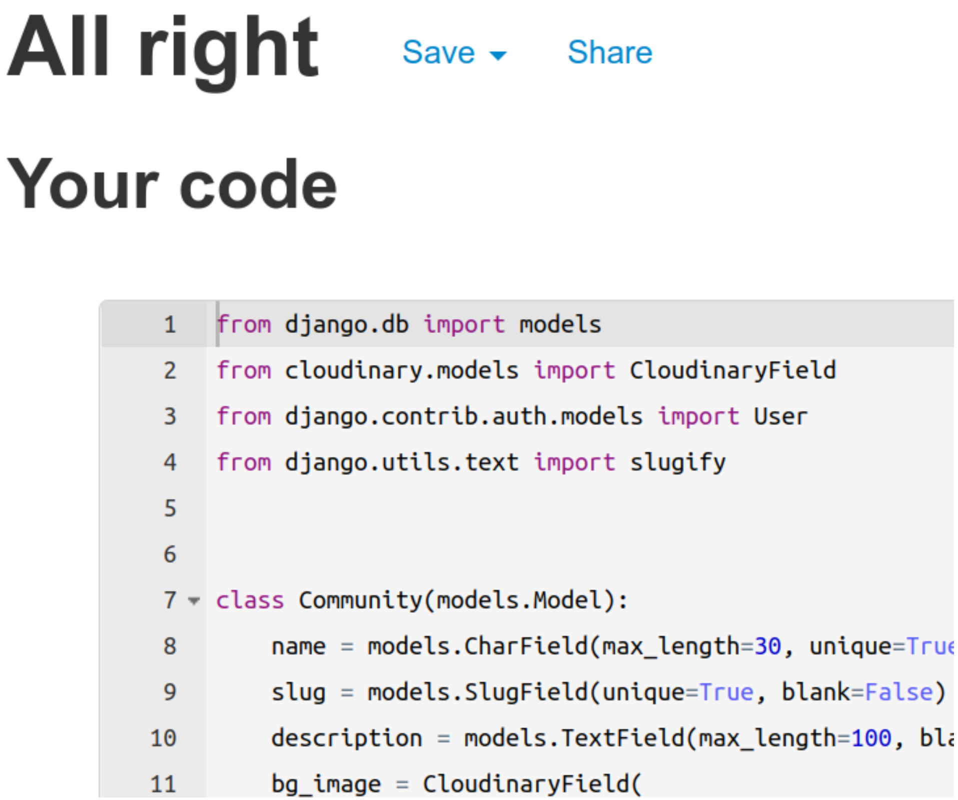
Your code

```python
from django import forms
from .models import Community


class CommunityForm(forms.ModelForm):
    class Meta:
        model = Community
        fields = ['name', 'description', 'bg_image', 'logo']

        widgets = {
            'name': forms.TextInput(attrs={'class': 'form-control'})
            'description': forms.Textarea(attrs={'class': 'form-cont
```

models.py



```
communities
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```

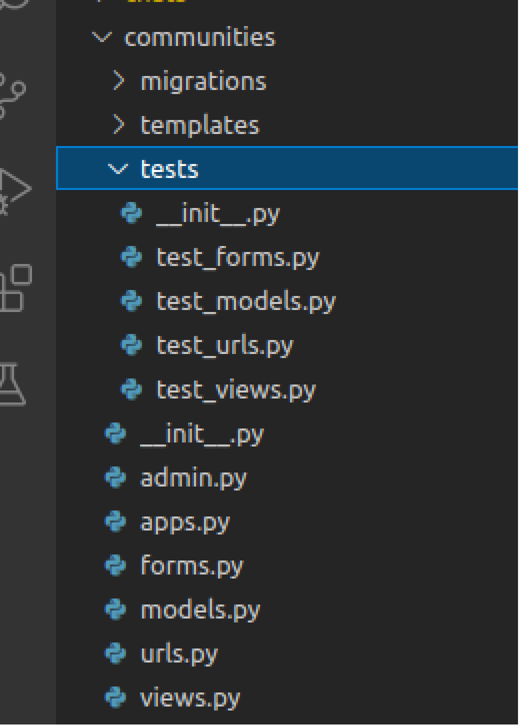


All right

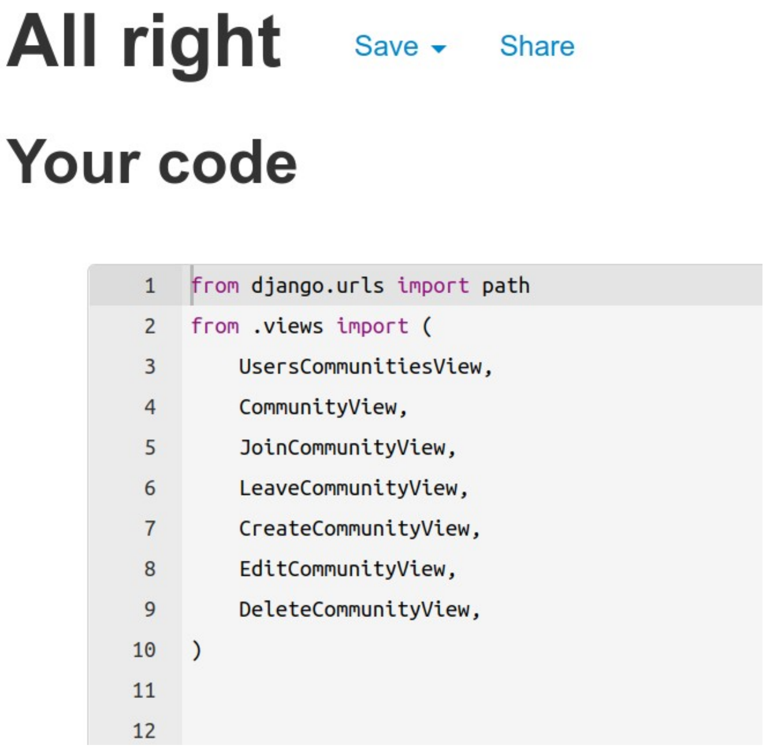
Your code

```python
from django.db import models
from cloudinary.models import CloudinaryField
from django.contrib.auth.models import User
from django.utils.text import slugify


class Community(models.Model):
    name = models.CharField(max_length=30, unique=True
    slug = models.SlugField(unique=True, blank=False)
    description = models.TextField(max_length=100, bla
    bg_image = CloudinaryField(
```

urls.py



```
communities
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```

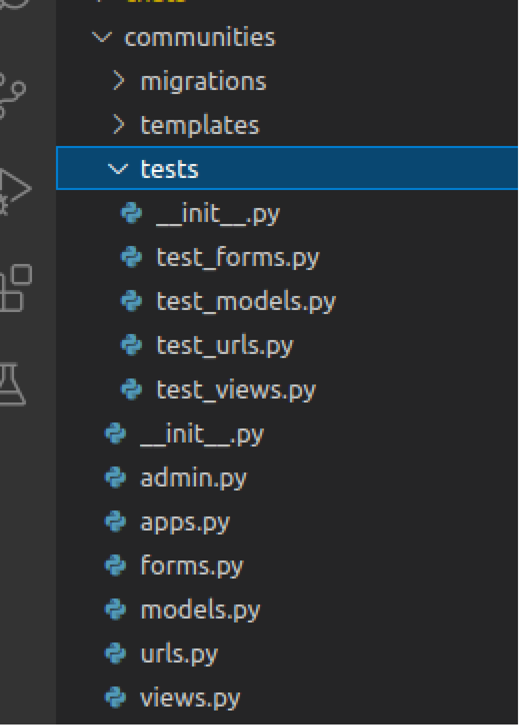


All right

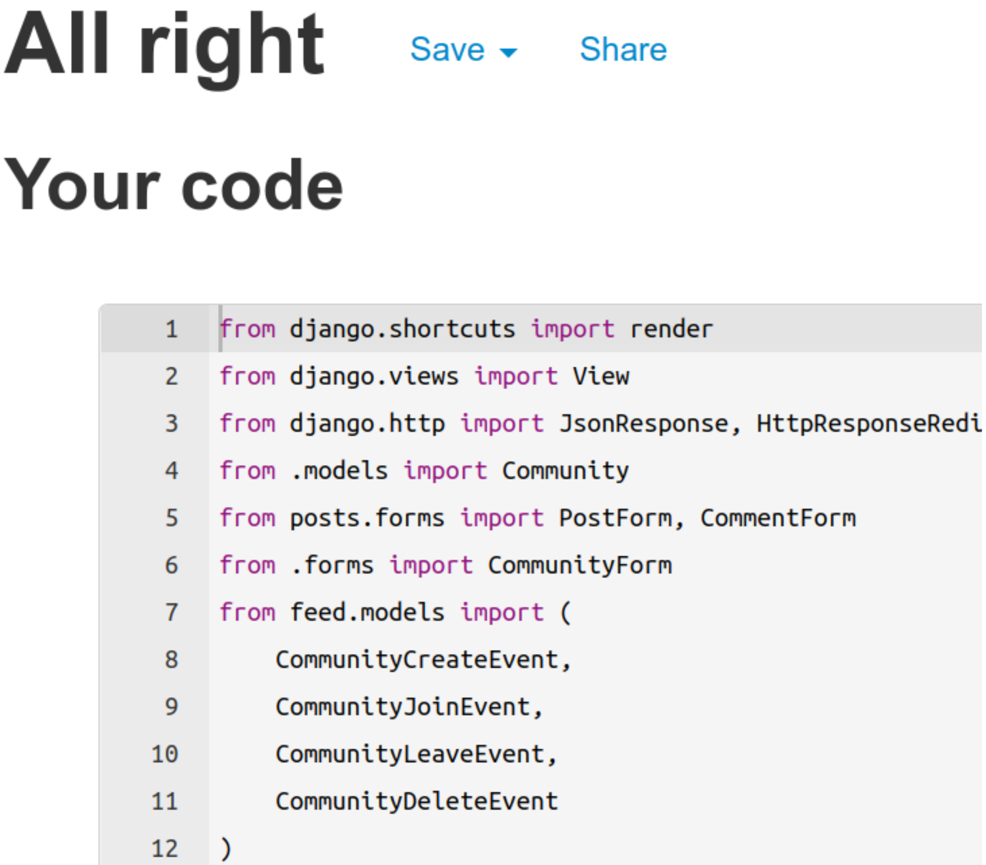
Your code

```python
from django.urls import path
from .views import (
    UsersCommunitiesView,
    CommunityView,
    JoinCommunityView,
    LeaveCommunityView,
    CreateCommunityView,
    EditCommunityView,
    DeleteCommunityView,
)


```

views.py



```
communities
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```

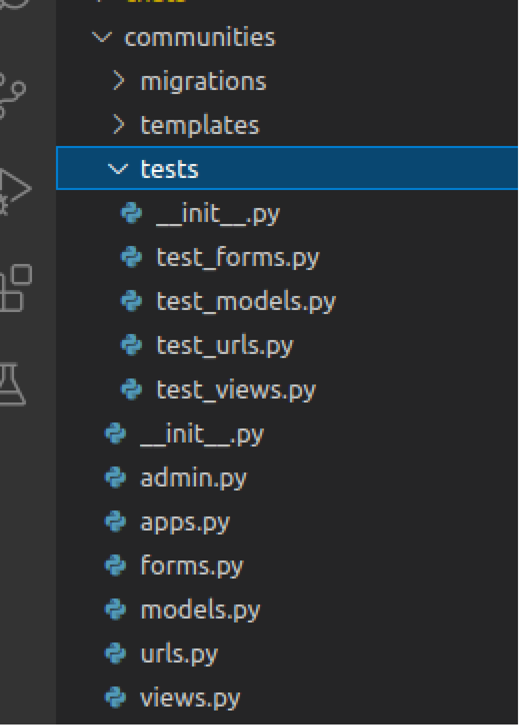


All right

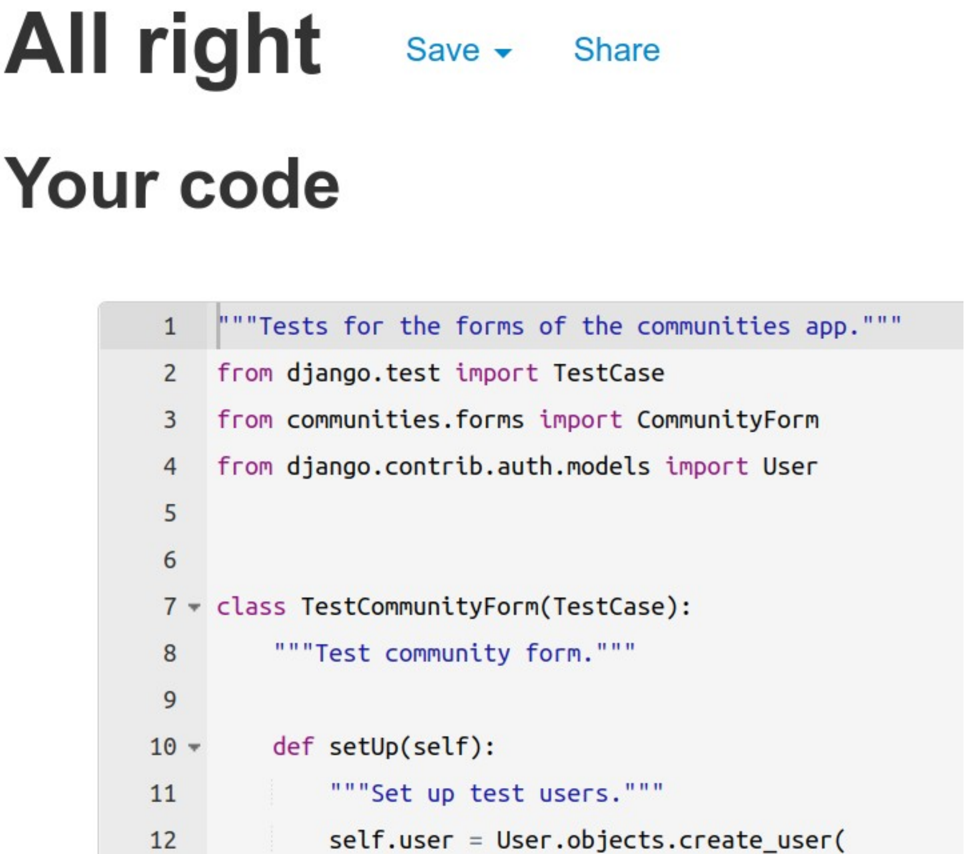
Your code

```python
from django.shortcuts import render
from django.views import View
from django.http import JsonResponse, HttpResponseRedi
from .models import Community
from posts.forms import PostForm, CommentForm
from .forms import CommunityForm
from feed.models import (
    CommunityCreateEvent,
    CommunityJoinEvent,
    CommunityLeaveEvent,
    CommunityDeleteEvent
)
```

tests/test_forms.py



```
communities
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```

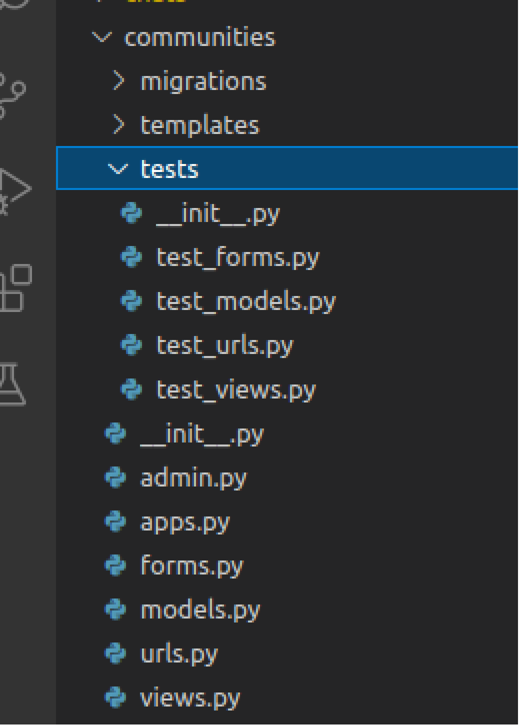


All right

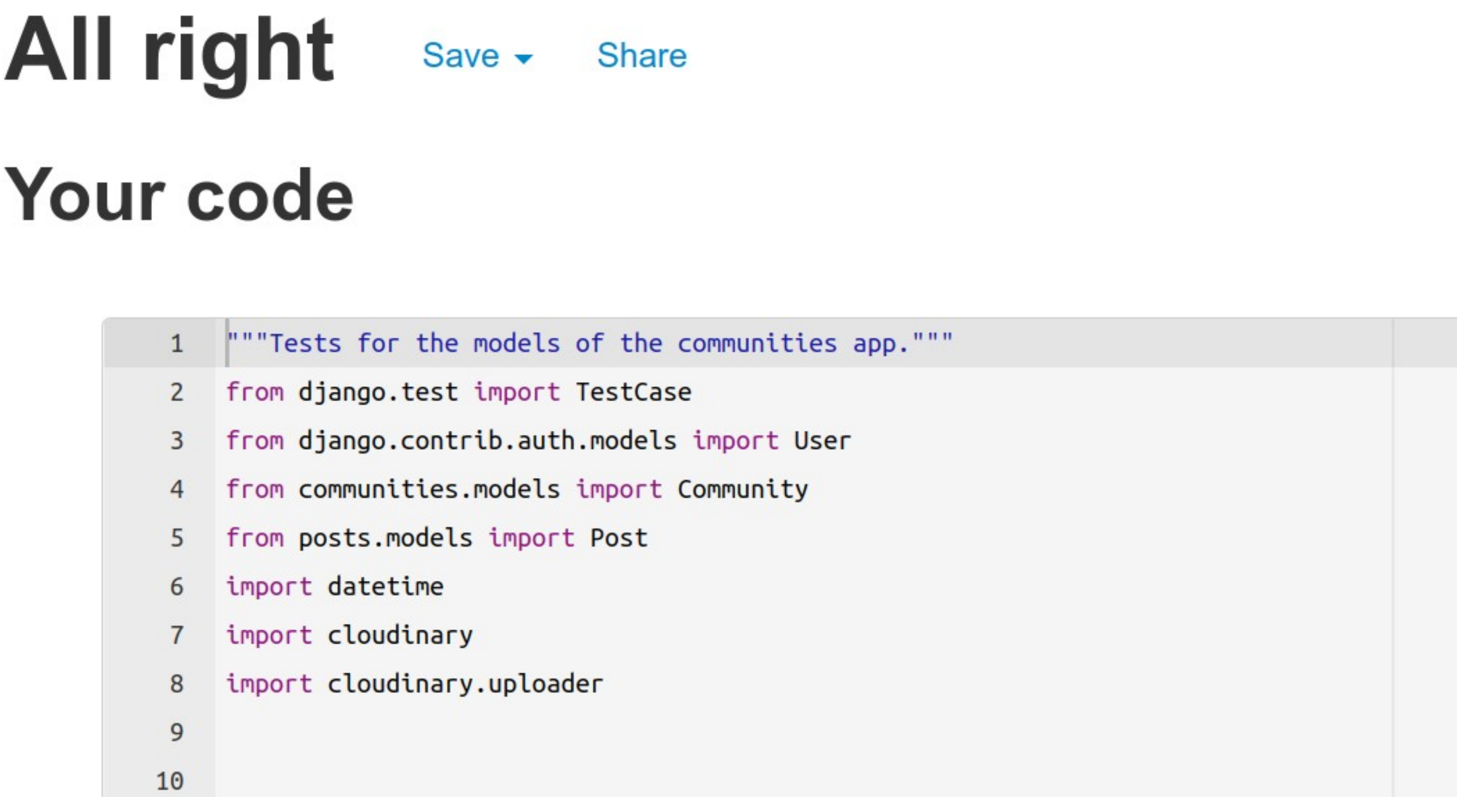
Your code

```python
"""Tests for the forms of the communities app."""
from django.test import TestCase
from communities.forms import CommunityForm
from django.contrib.auth.models import User


class TestCommunityForm(TestCase):
    """Test community form."""

    def setUp(self):
        """Set up test users."""
        self.user = User.objects.create_user(
```

tests/test_models.py



```
communities
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```

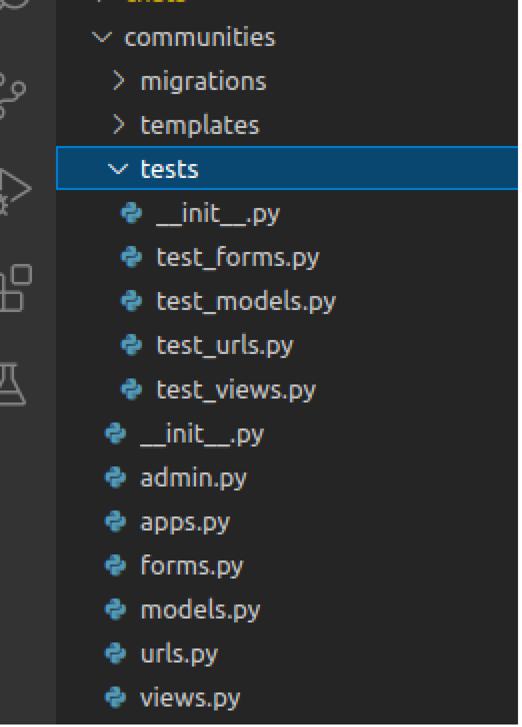


All right

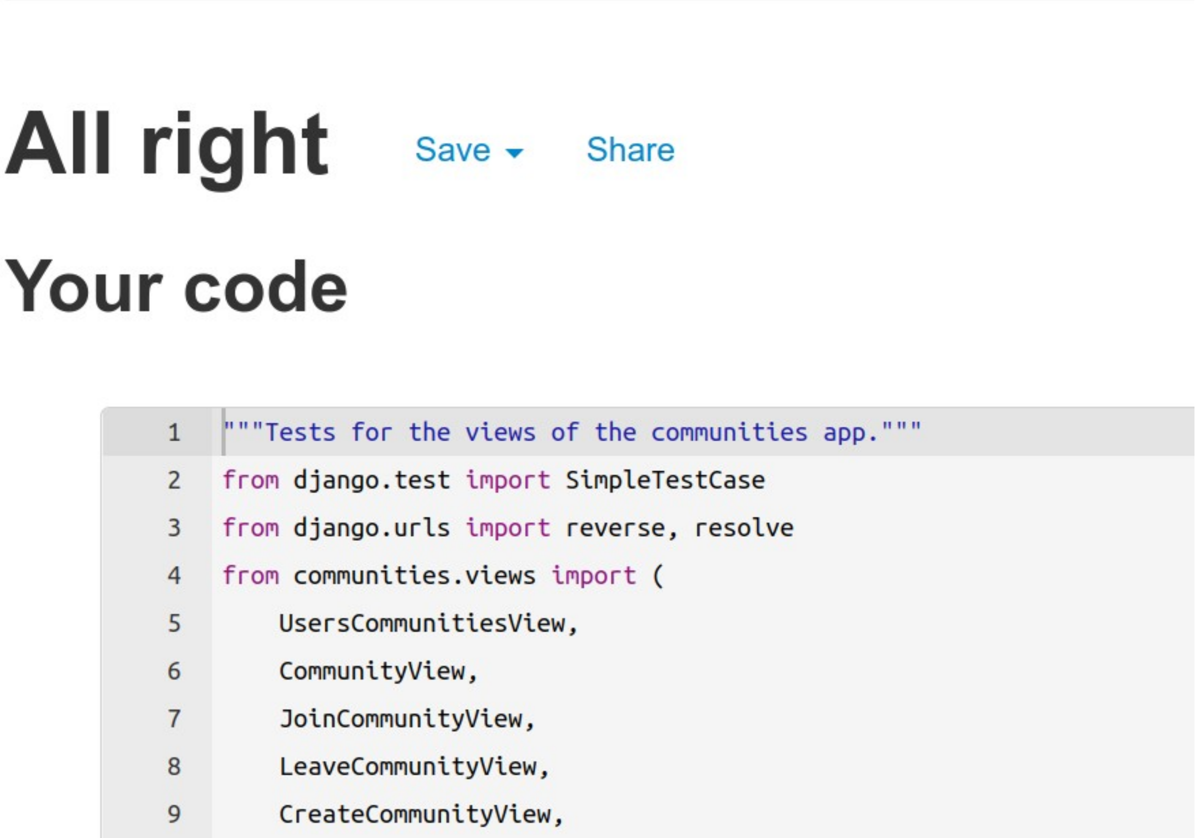
Your code

```python
"""Tests for the models of the communities app."""
from django.test import TestCase
from django.contrib.auth.models import User
from communities.models import Community
from posts.models import Post
import datetime
import cloudinary
import cloudinary.uploader


```

tests/test_urls.py



```
communities
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```

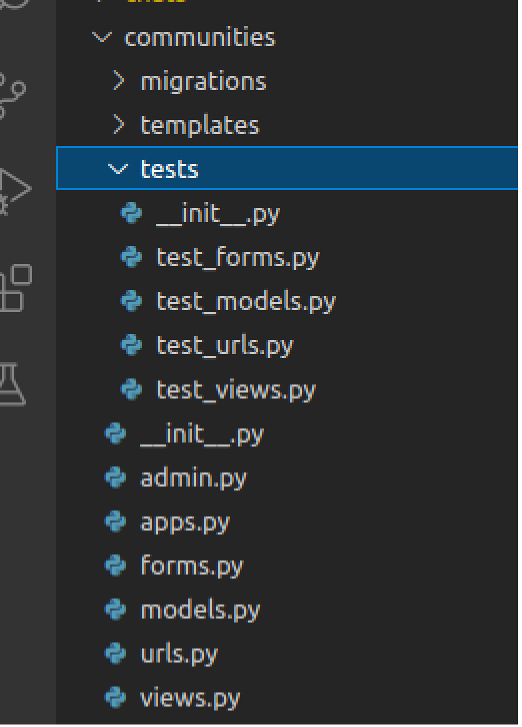


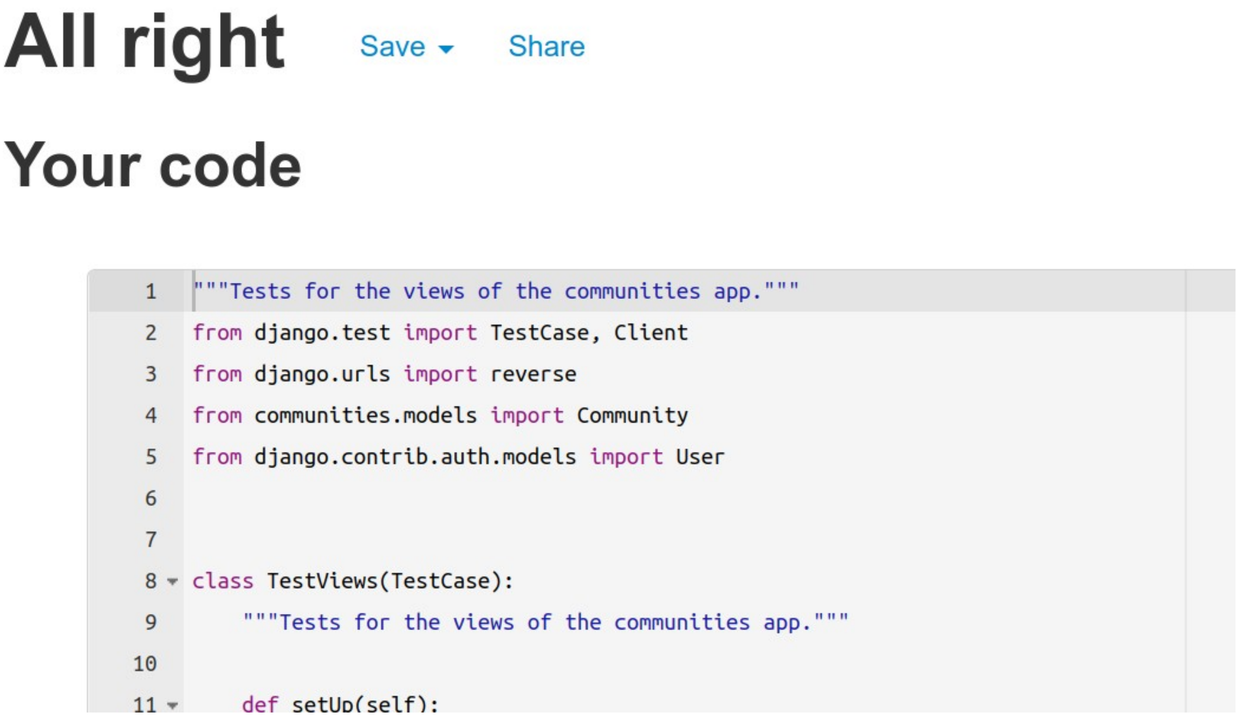
All right

Your code

```python
"""Tests for the views of the communities app."""
from django.test import SimpleTestCase
from django.urls import reverse, resolve
from communities.views import (
    UsersCommunitiesView,
    CommunityView,
    JoinCommunityView,
    LeaveCommunityView,
    CreateCommunityView,
```

tests/test_views.py

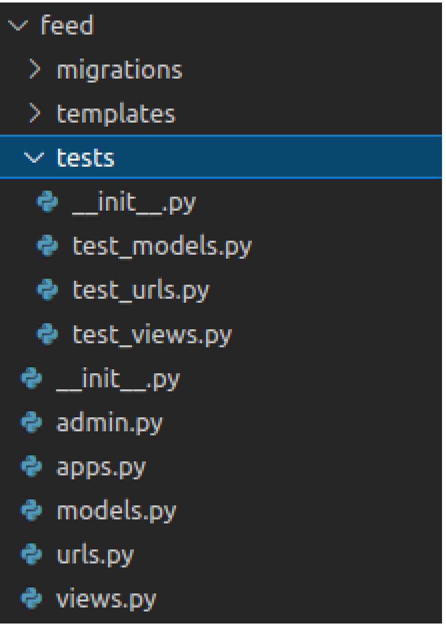




All right

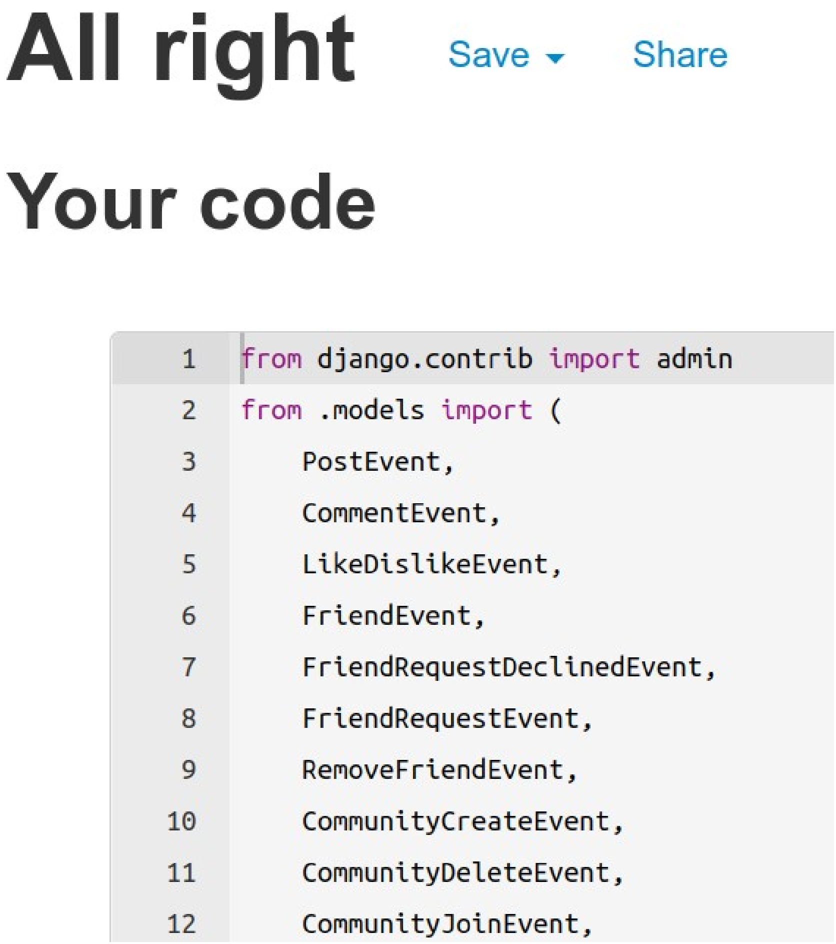
Your code

```python
"""Tests for the views of the communities app."""
from django.test import TestCase, Client
from django.urls import reverse
from communities.models import Community
from django.contrib.auth.models import User


class TestViews(TestCase):
    """Tests for the views of the communities app."""

    def setUp(self):
```



```
feed
  migrations
  templates
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

admin.py

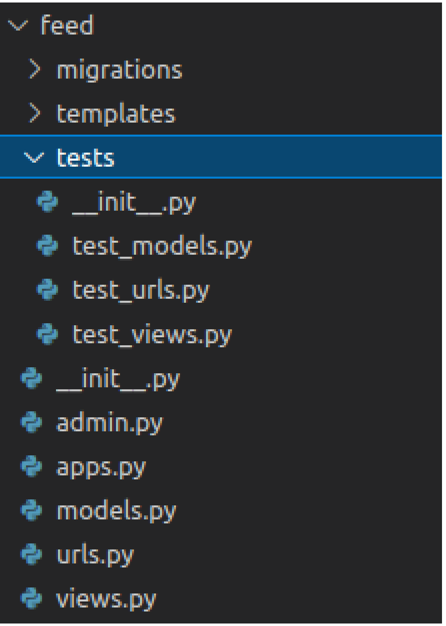


All right

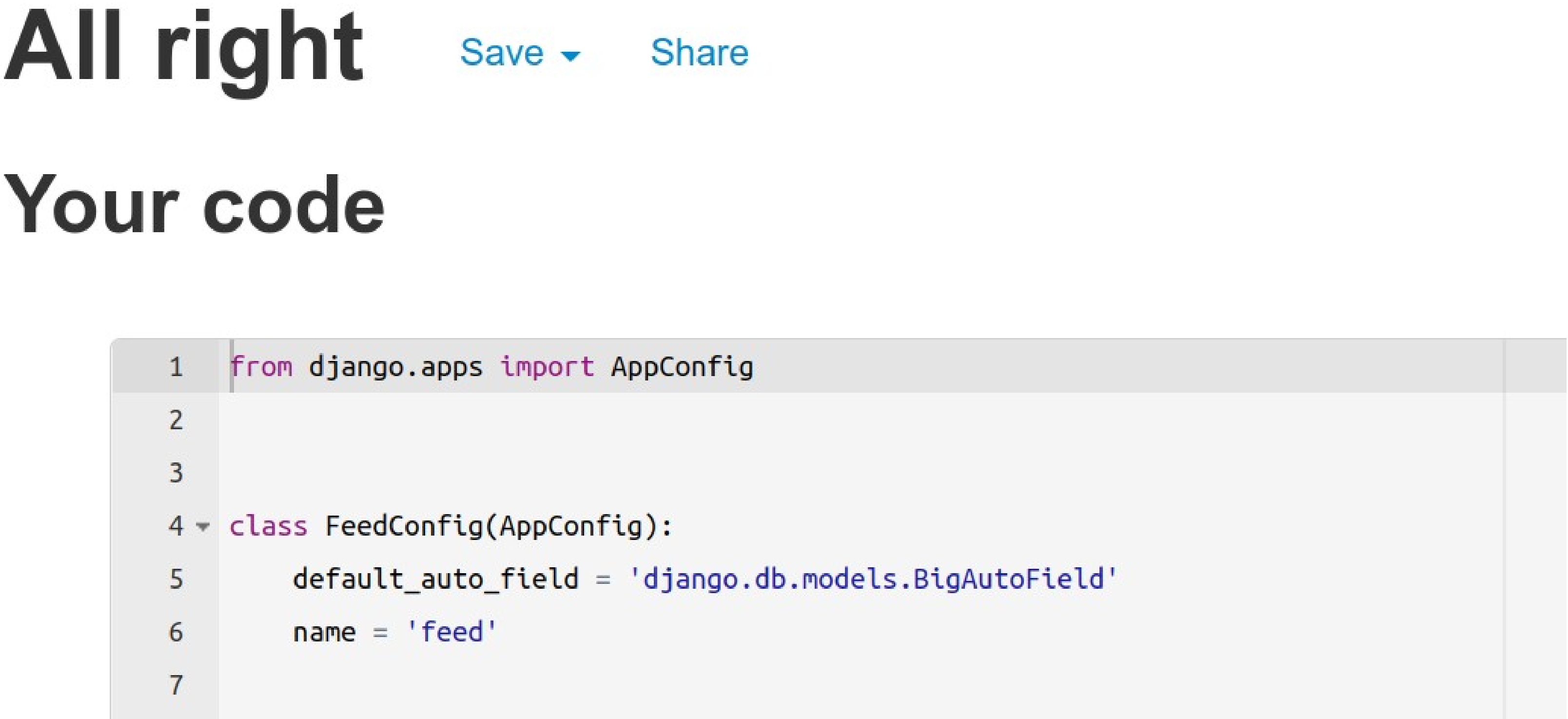
Your code

```python
from django.contrib import admin
from .models import (
    PostEvent,
    CommentEvent,
    LikeDislikeEvent,
    FriendEvent,
    FriendRequestDeclinedEvent,
    FriendRequestEvent,
    RemoveFriendEvent,
    CommunityCreateEvent,
    CommunityDeleteEvent,
    CommunityJoinEvent,
```

apps.py



```
feed
  migrations
  templates
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

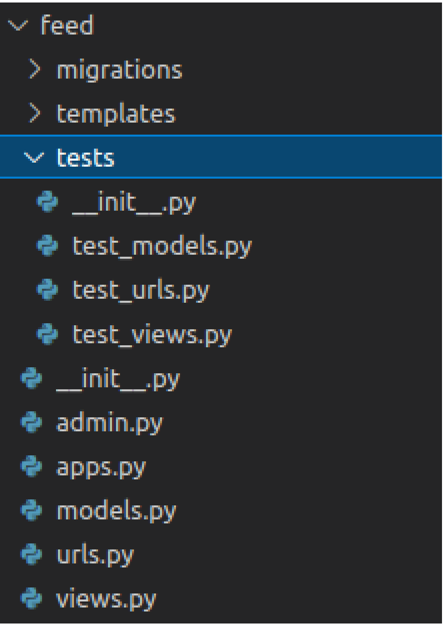


All right

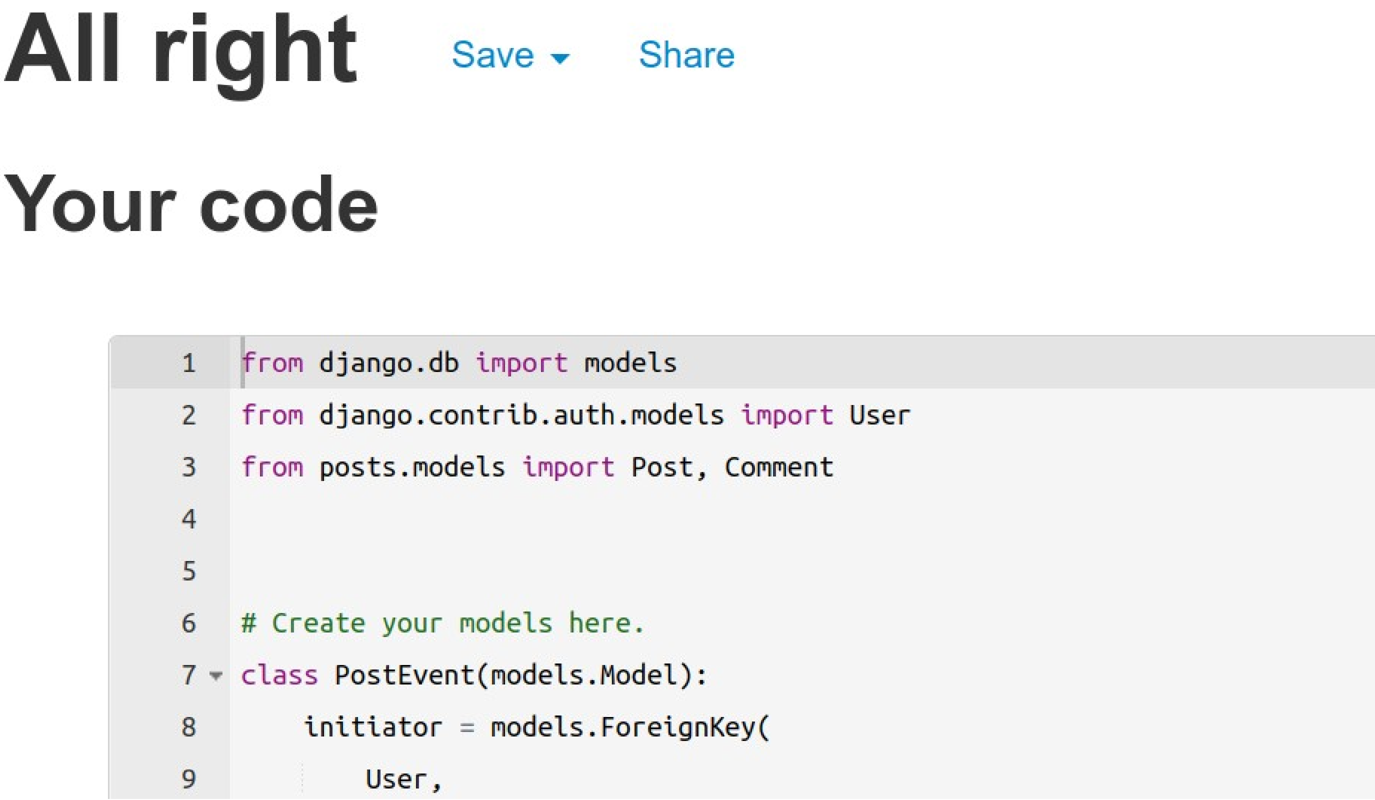
Your code

```python
from django.apps import AppConfig


class FeedConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'feed'

```

models.py



```
feed
  migrations
  templates
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

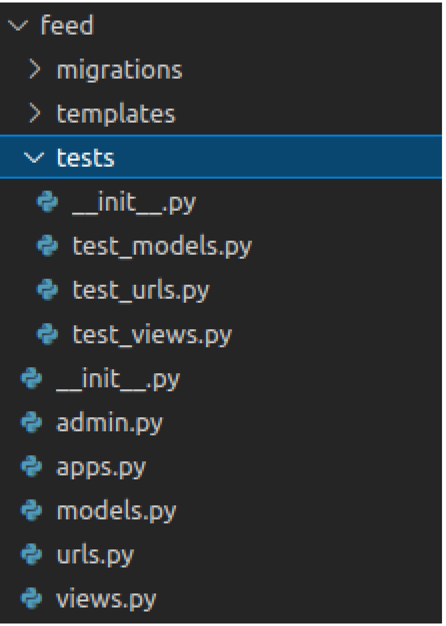


All right

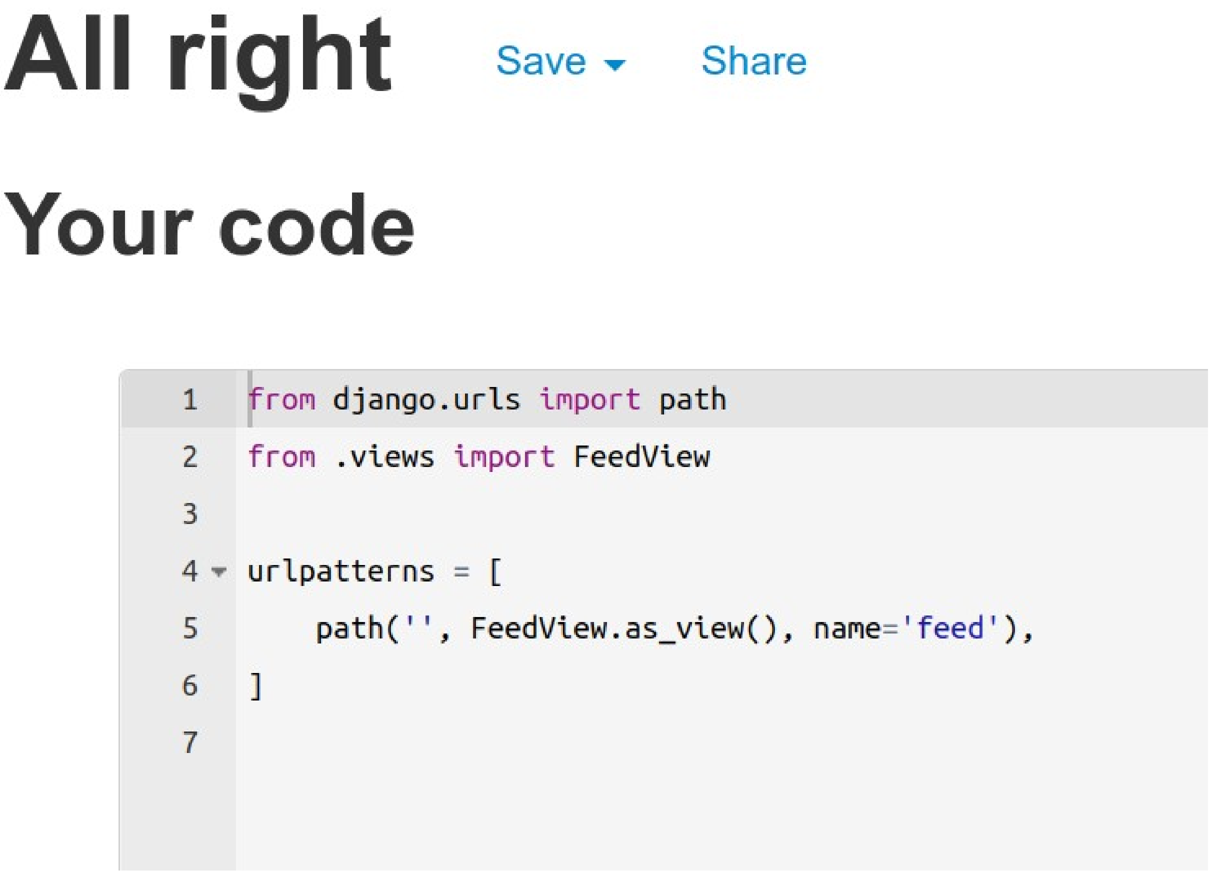
Your code

```python
from django.db import models
from django.contrib.auth.models import User
from posts.models import Post, Comment


# Create your models here.
class PostEvent(models.Model):
    initiator = models.ForeignKey(
        User,
```

urls.py



```
feed
  migrations
  templates
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

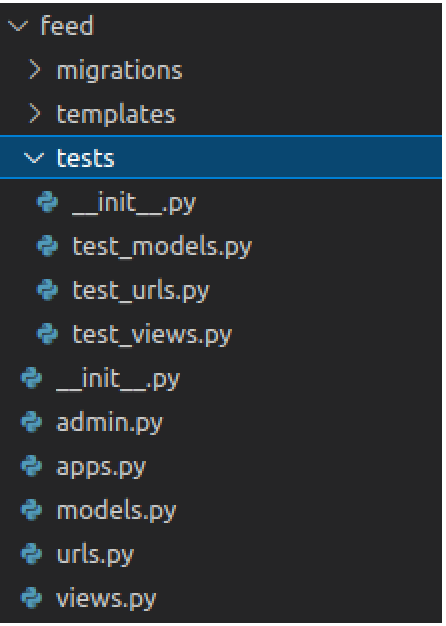


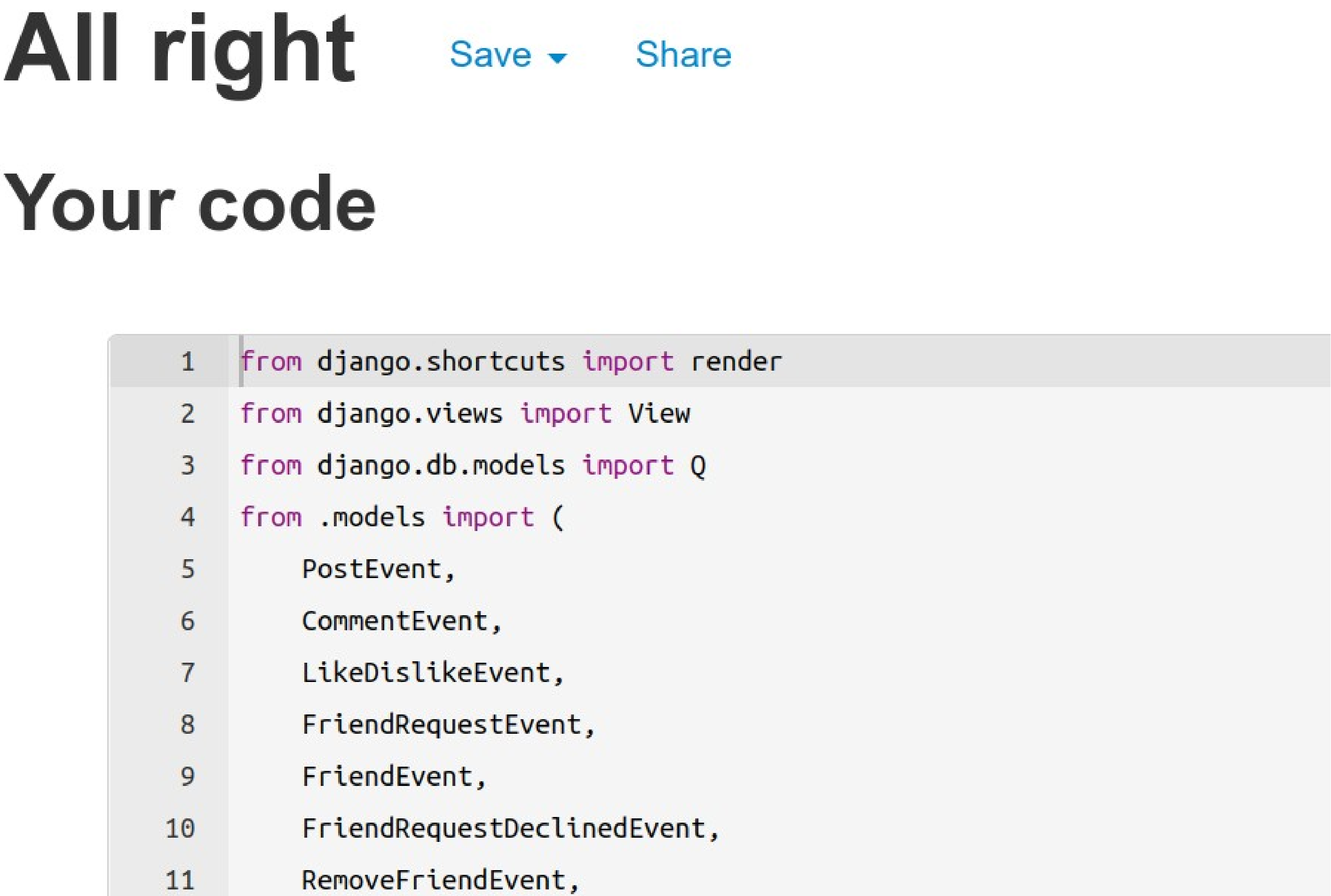
All right

Your code

```python
from django.urls import path
from .views import FeedView

urlpatterns = [
    path('', FeedView.as_view(), name='feed'),
]

```

views.py



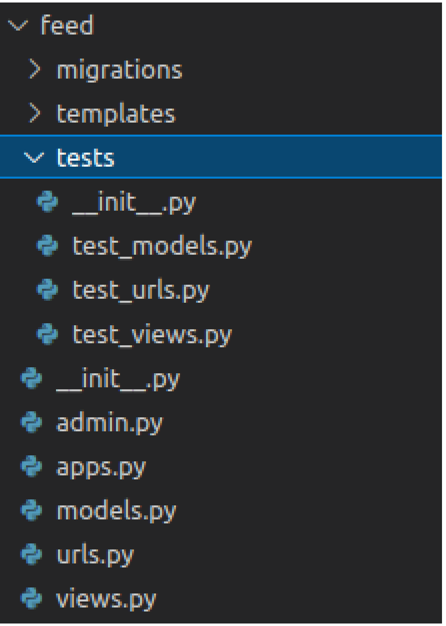


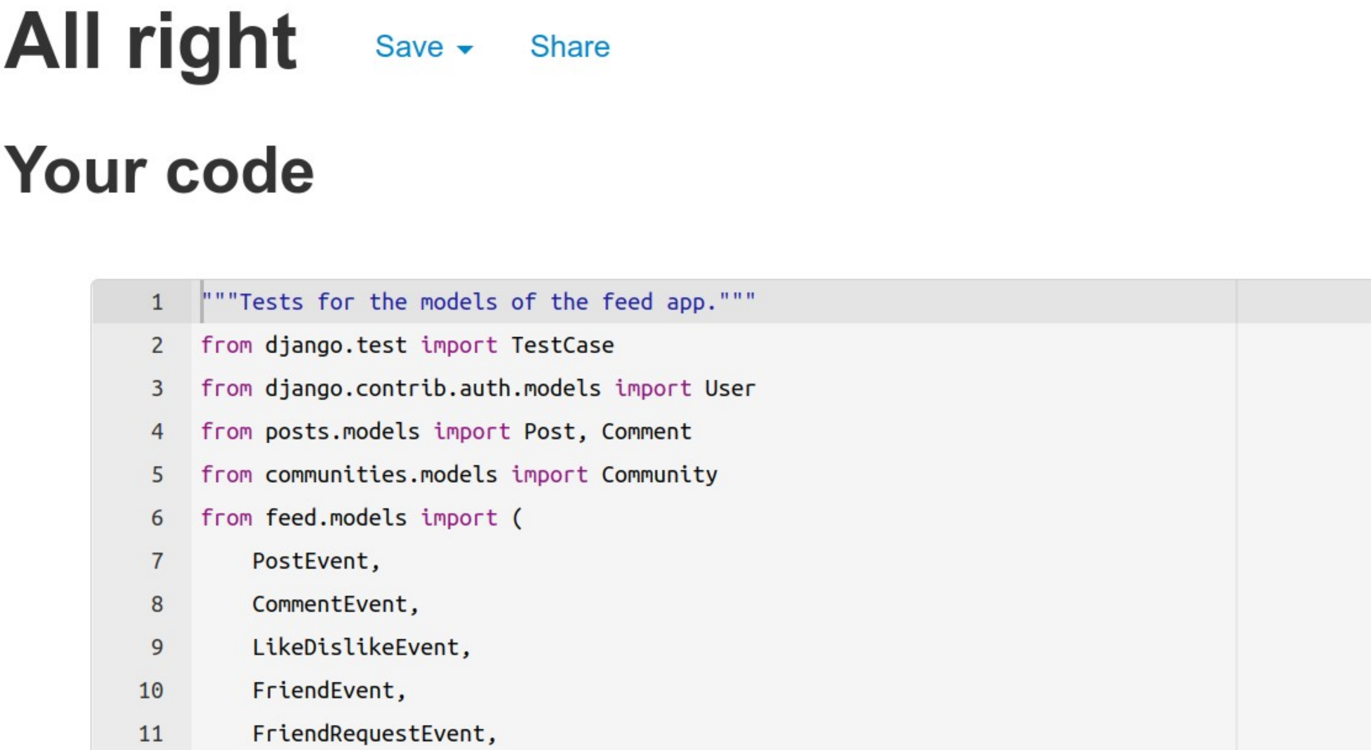
All right

Your code

```
from django.shortcuts import render
from django.views import View
from django.db.models import Q
from .models import (
    PostEvent,
    CommentEvent,
    LikeDislikeEvent,
    FriendRequestEvent,
    FriendEvent,
    FriendRequestDeclinedEvent,
    RemoveFriendEvent,
```

tests/test_models.py



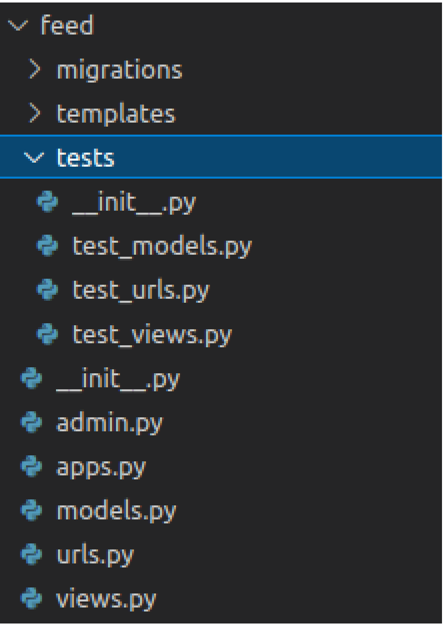


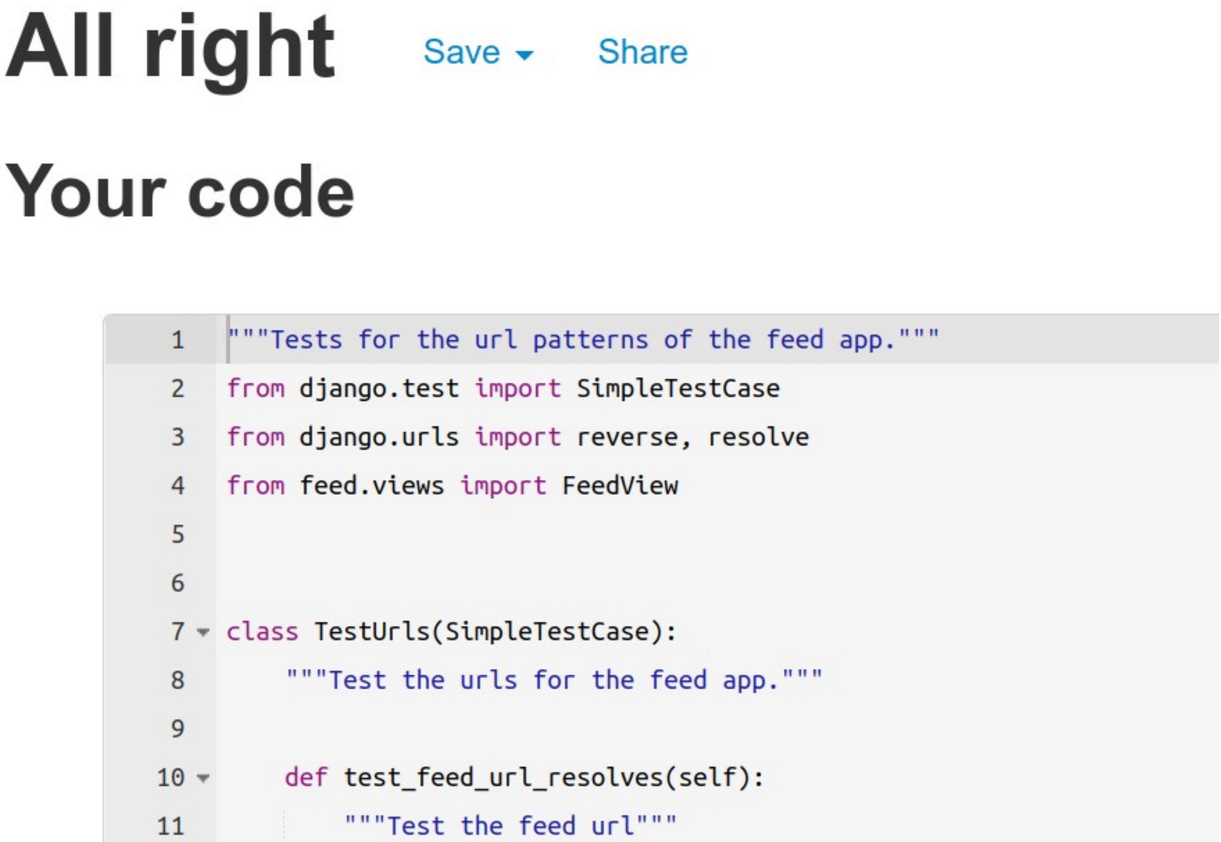
All right

Your code

```python
"""Tests for the models of the feed app."""
from django.test import TestCase
from django.contrib.auth.models import User
from posts.models import Post, Comment
from communities.models import Community
from feed.models import (
    PostEvent,
    CommentEvent,
    LikeDislikeEvent,
    FriendEvent,
    FriendRequestEvent,
```

tests/test_urls.py


```
feed
  migrations
  templates
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

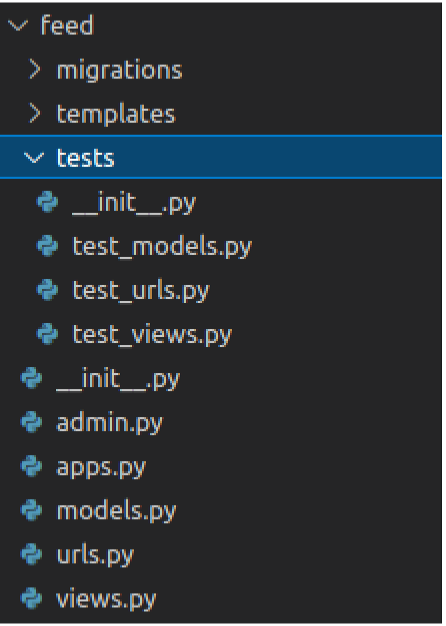


All right

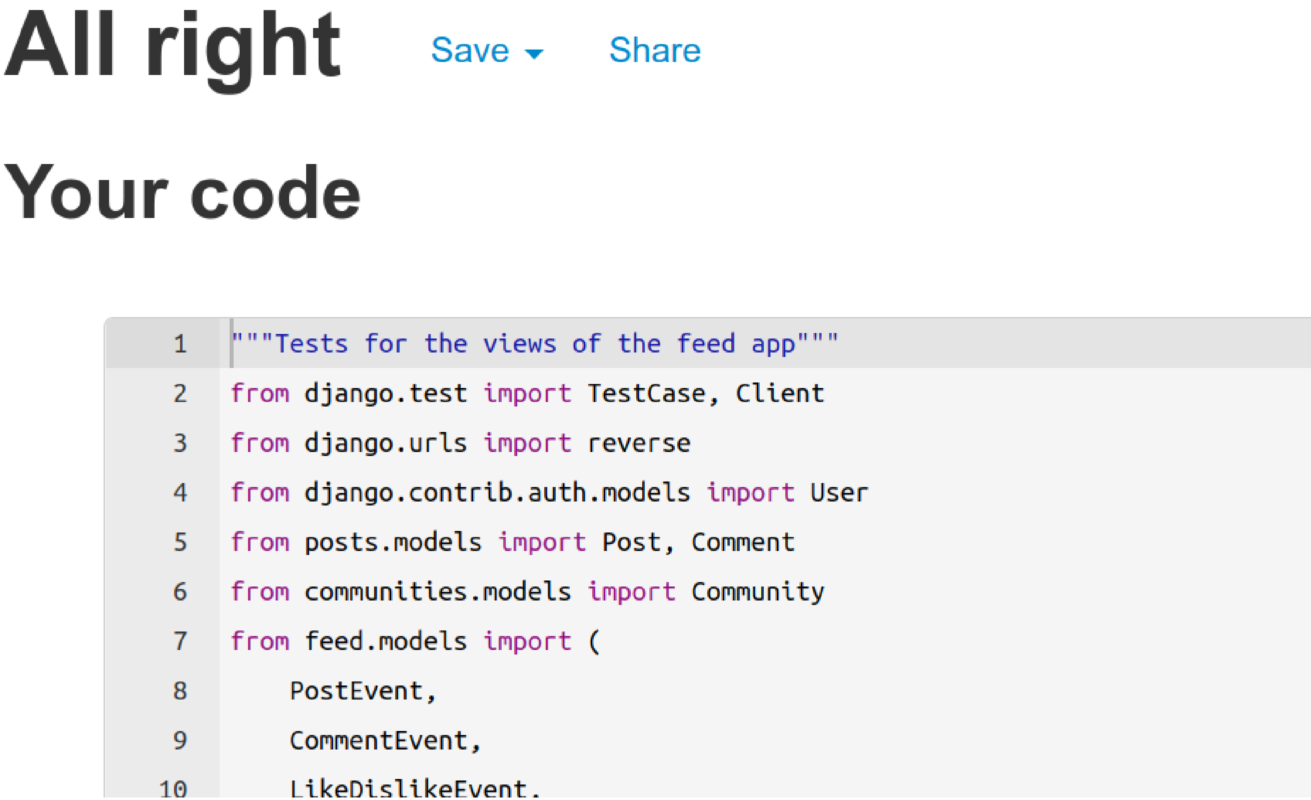
Your code

```python
"""Tests for the url patterns of the feed app."""
from django.test import SimpleTestCase
from django.urls import reverse, resolve
from feed.views import FeedView


class TestUrls(SimpleTestCase):
    """Test the urls for the feed app."""

    def test_feed_url_resolves(self):
        """Test the feed url"""
```

tests/test_views.py



```
feed
  migrations
  templates
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

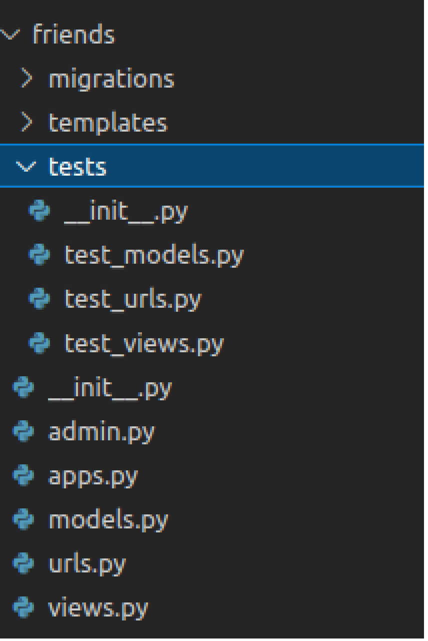


All right

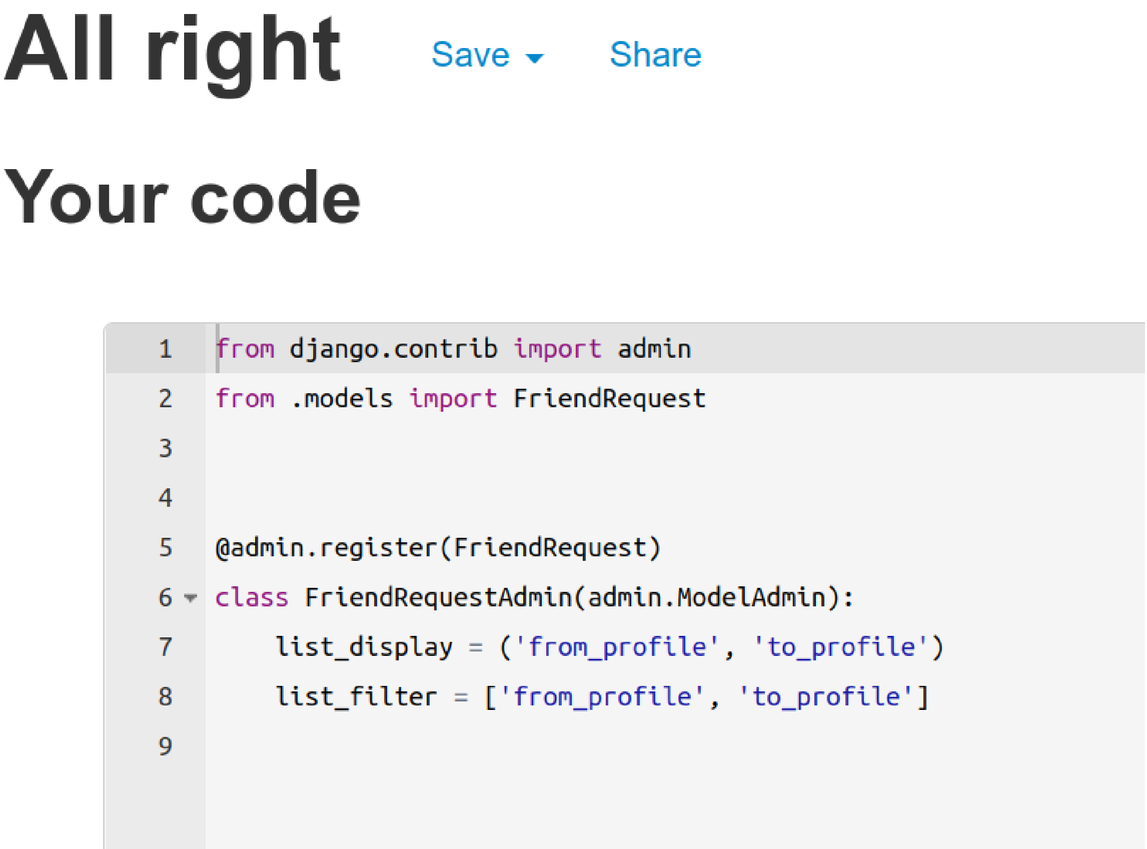
Your code

```python
"""Tests for the views of the feed app"""
from django.test import TestCase, Client
from django.urls import reverse
from django.contrib.auth.models import User
from posts.models import Post, Comment
from communities.models import Community
from feed.models import (
    PostEvent,
    CommentEvent,
    LikeDislikeEvent,
```

admin.py



```
friends
  migrations
  templates
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

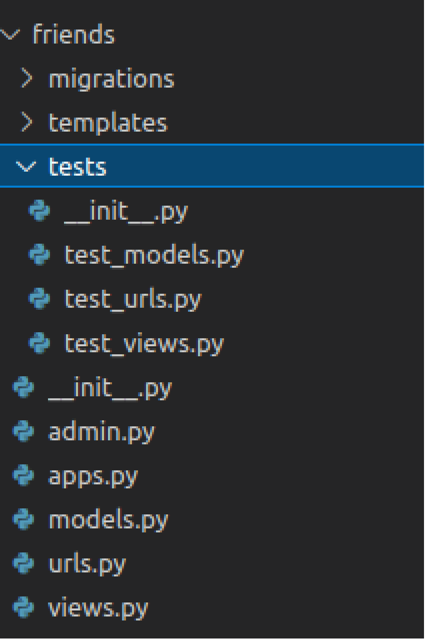


All right

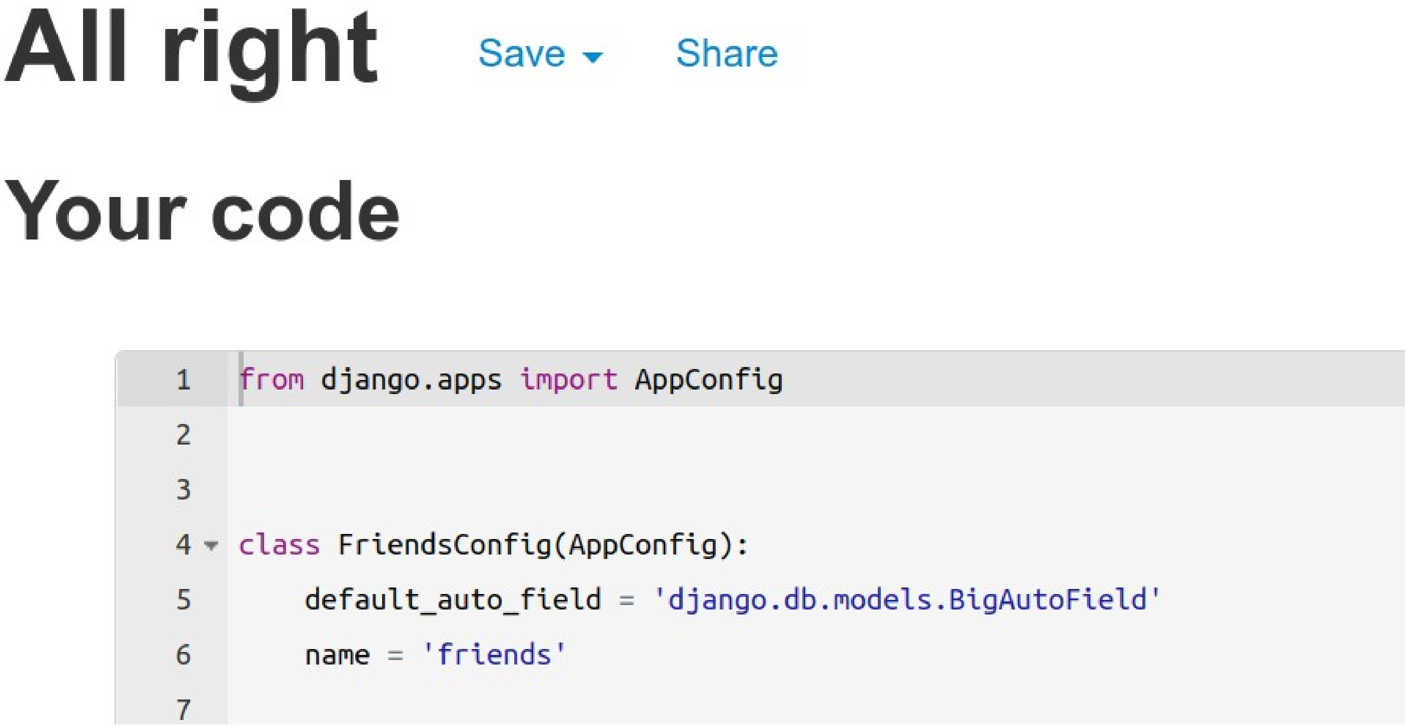
Your code

```python
from django.contrib import admin
from .models import FriendRequest


@admin.register(FriendRequest)
class FriendRequestAdmin(admin.ModelAdmin):
    list_display = ('from_profile', 'to_profile')
    list_filter = ['from_profile', 'to_profile']

```

apps.py



```
friends
  migrations
  templates
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

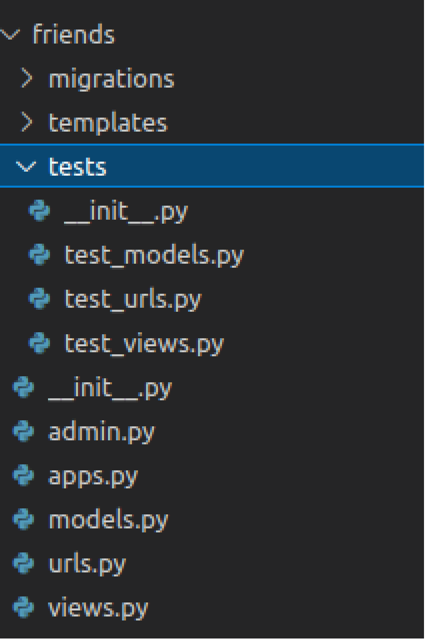


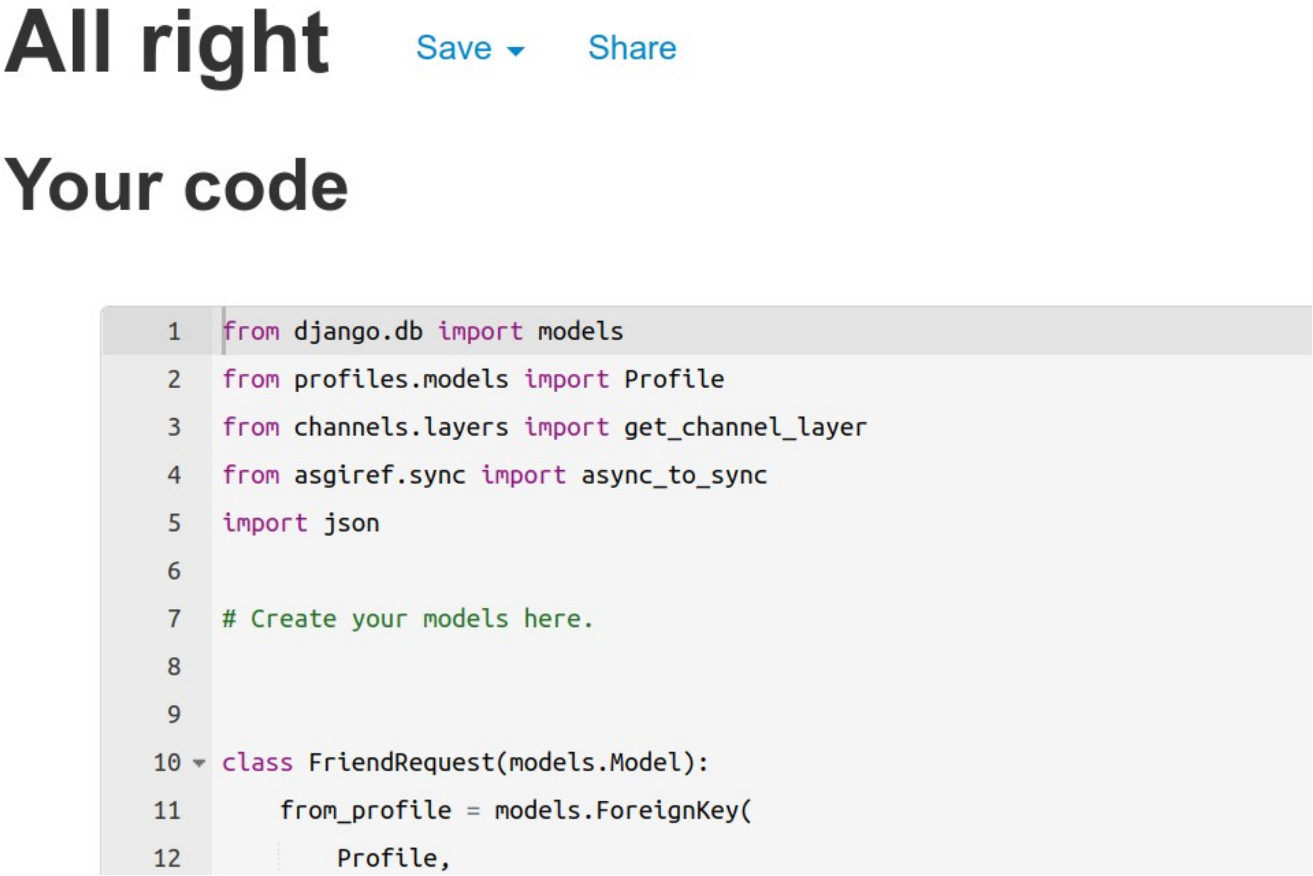
All right

Your code

```python
from django.apps import AppConfig


class FriendsConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'friends'

```

models.py



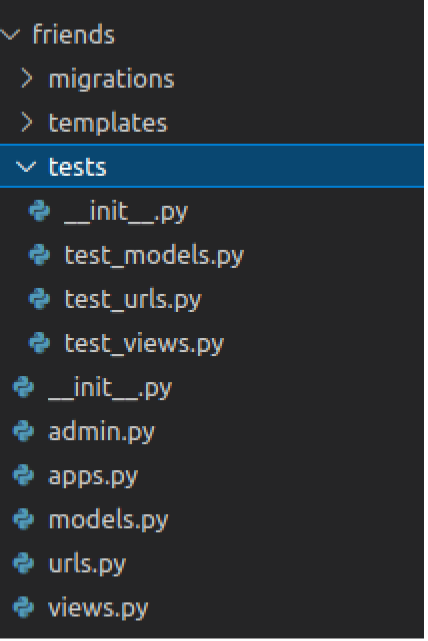


All right

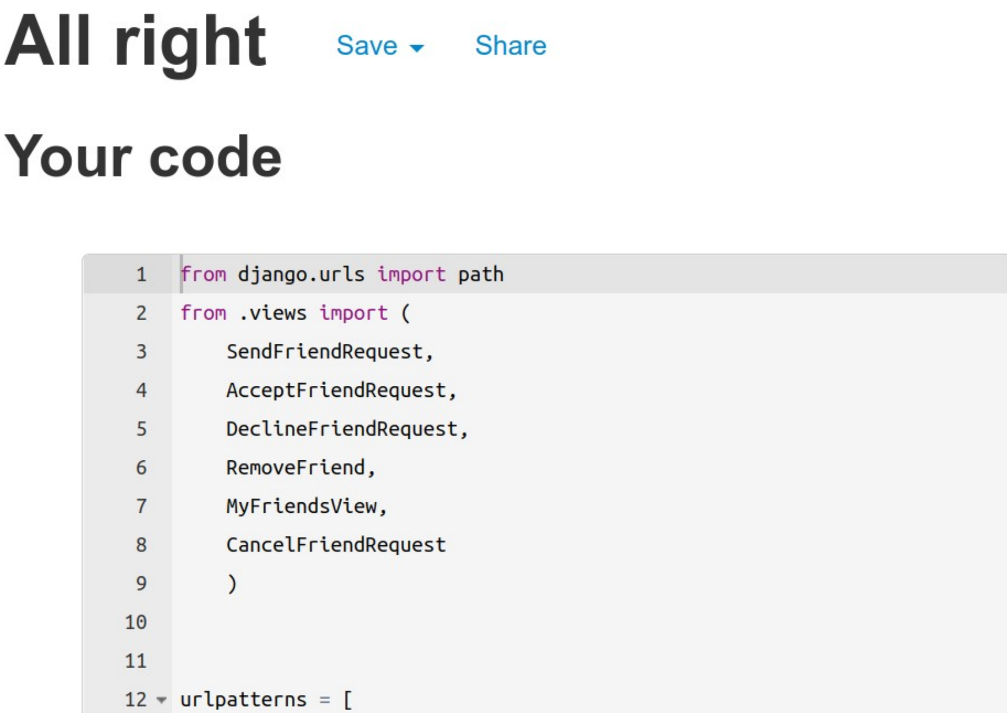
Your code

```
from django.db import models
from profiles.models import Profile
from channels.layers import get_channel_layer
from asgiref.sync import async_to_sync
import json

# Create your models here.


class FriendRequest(models.Model):
    from_profile = models.ForeignKey(
        Profile,
```

urls.py



```
friends
  migrations
  templates
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

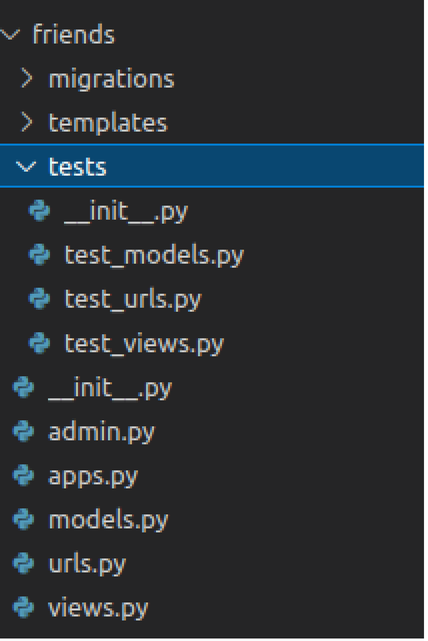


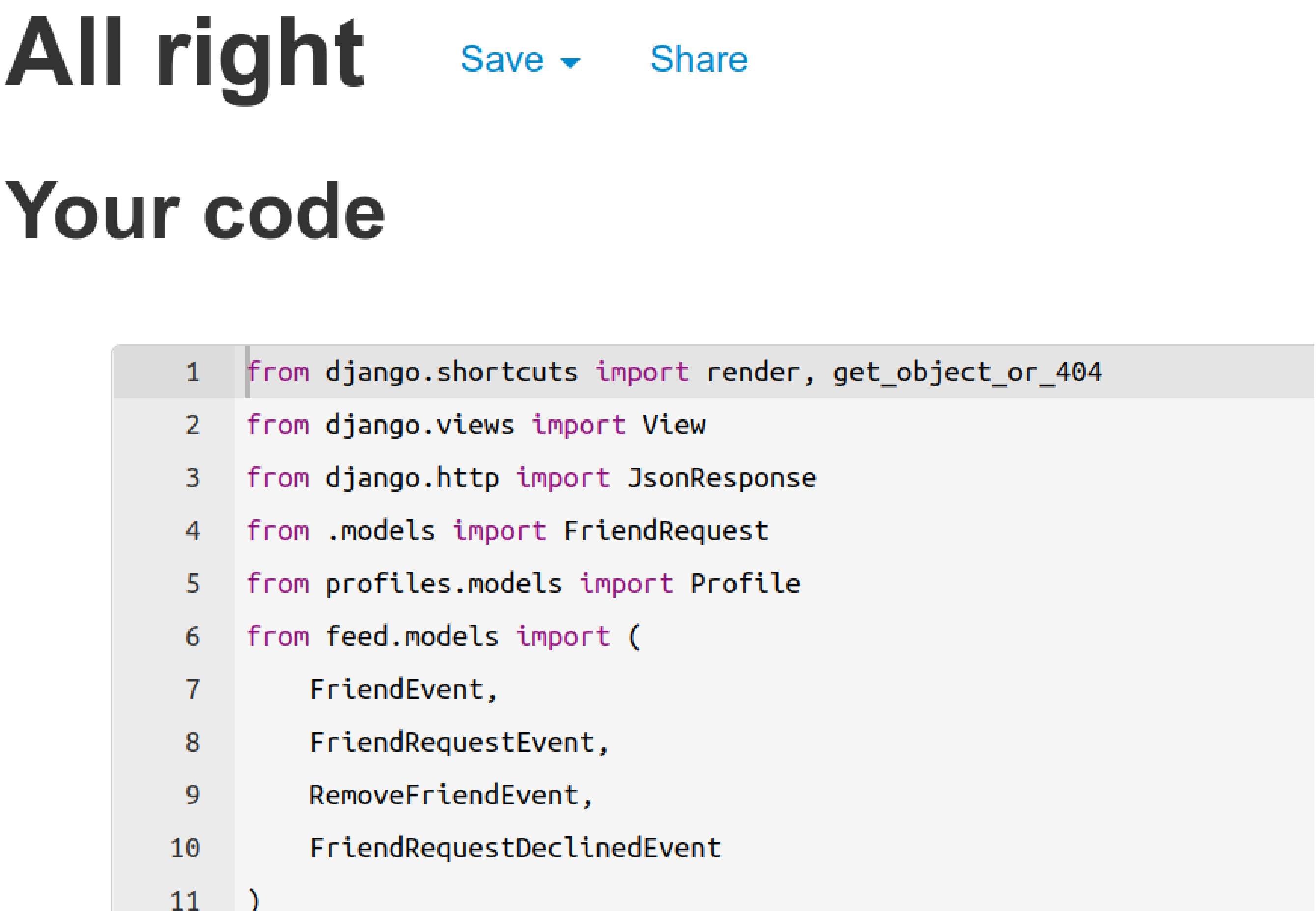
All right

Your code

```python
from django.urls import path
from .views import (
    SendFriendRequest,
    AcceptFriendRequest,
    DeclineFriendRequest,
    RemoveFriend,
    MyFriendsView,
    CancelFriendRequest
    )


urlpatterns = [
```

views.py



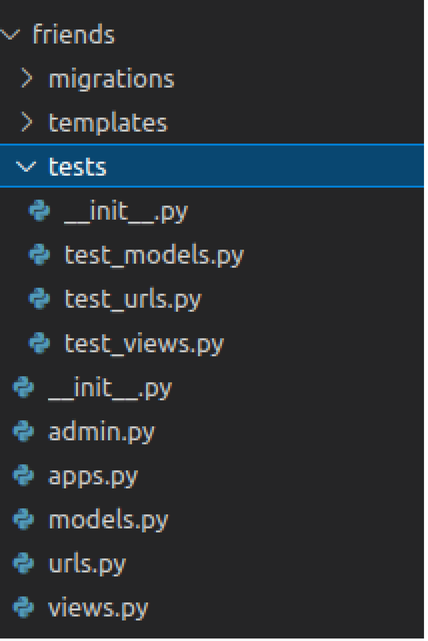


All right

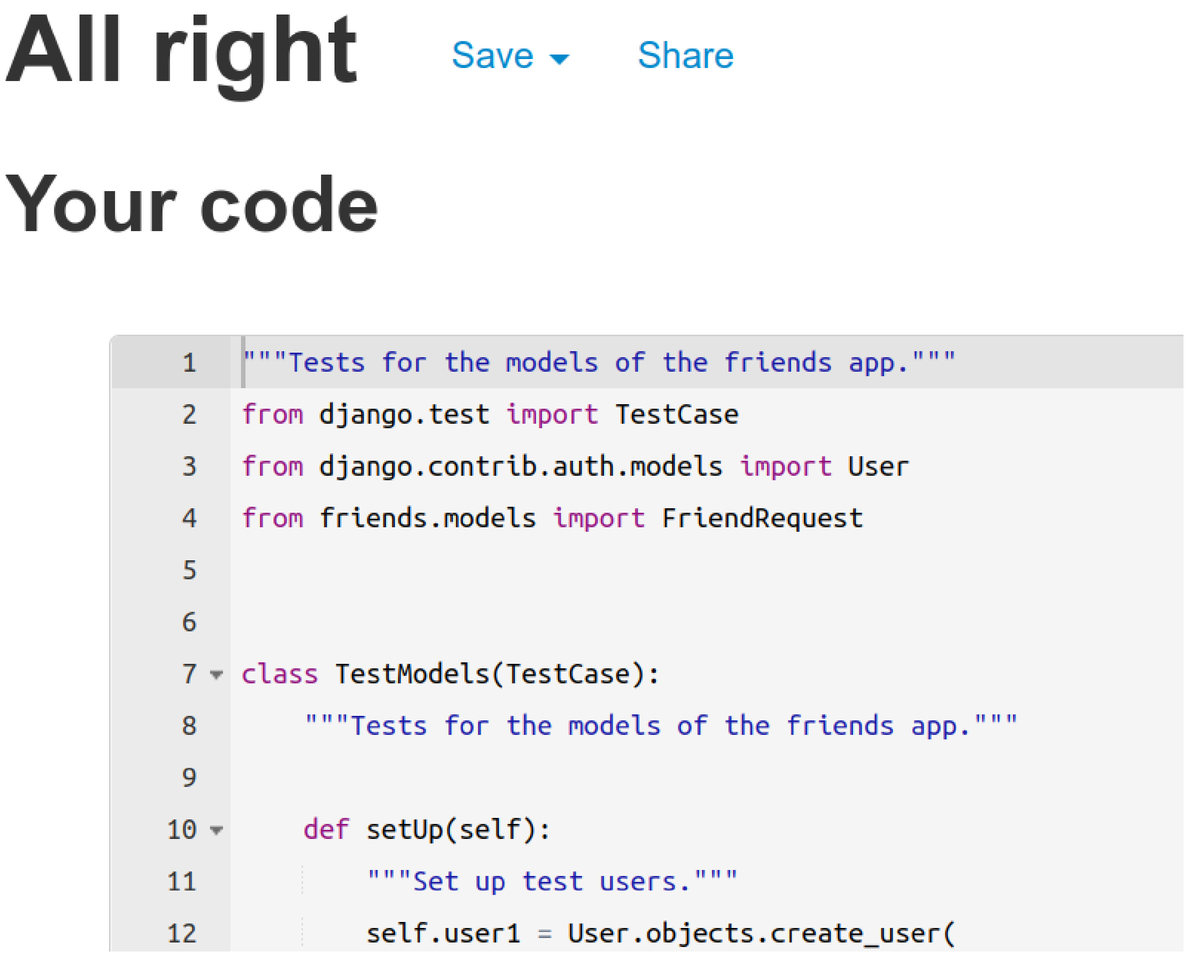
Your code

```
from django.shortcuts import render, get_object_or_404
from django.views import View
from django.http import JsonResponse
from .models import FriendRequest
from profiles.models import Profile
from feed.models import (
    FriendEvent,
    FriendRequestEvent,
    RemoveFriendEvent,
    FriendRequestDeclinedEvent
)
```

tests/test_models.py



```
friends
  migrations
  templates
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

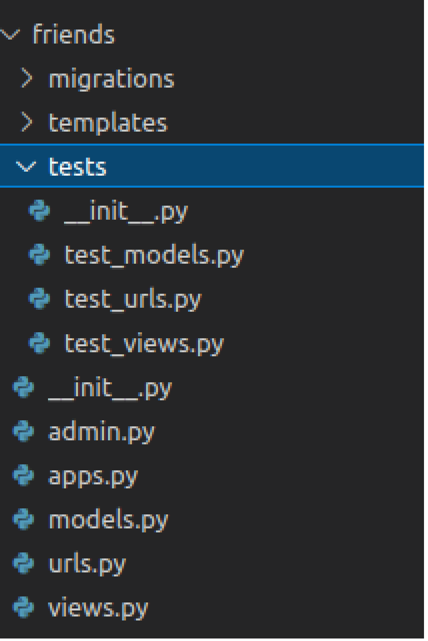


All right

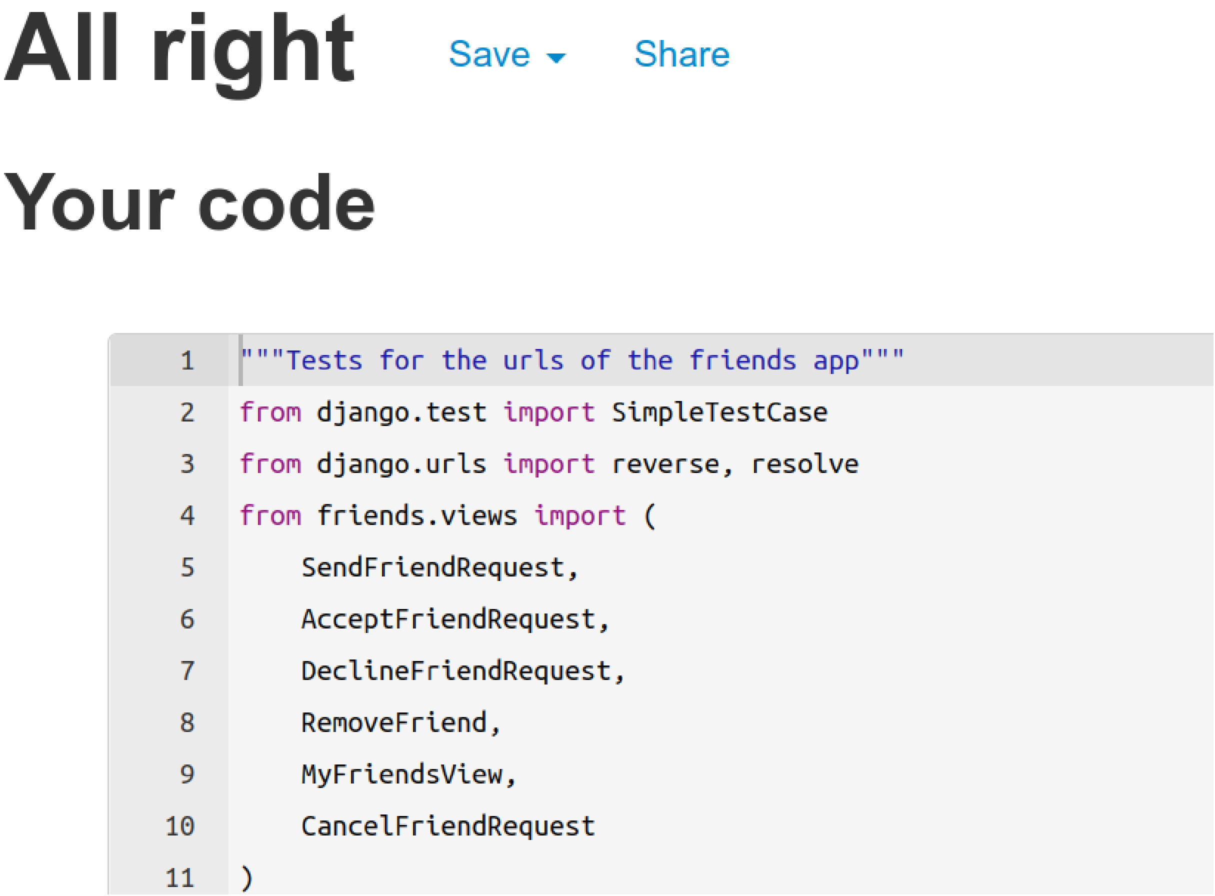
Your code

```python
"""Tests for the models of the friends app."""
from django.test import TestCase
from django.contrib.auth.models import User
from friends.models import FriendRequest


class TestModels(TestCase):
    """Tests for the models of the friends app."""

    def setUp(self):
        """Set up test users."""
        self.user1 = User.objects.create_user(
```

tests/test_urls.py



```
friends
  migrations
  templates
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

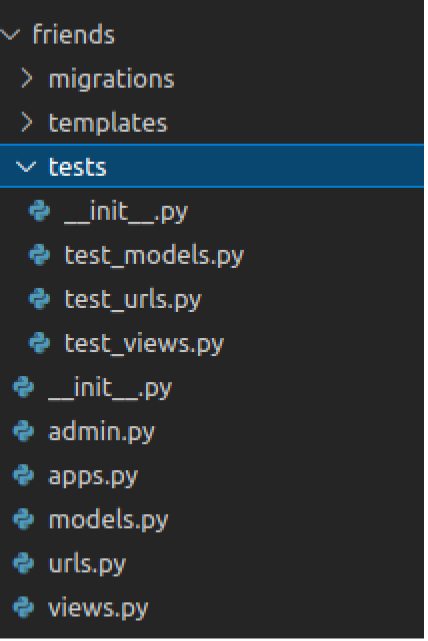


All right

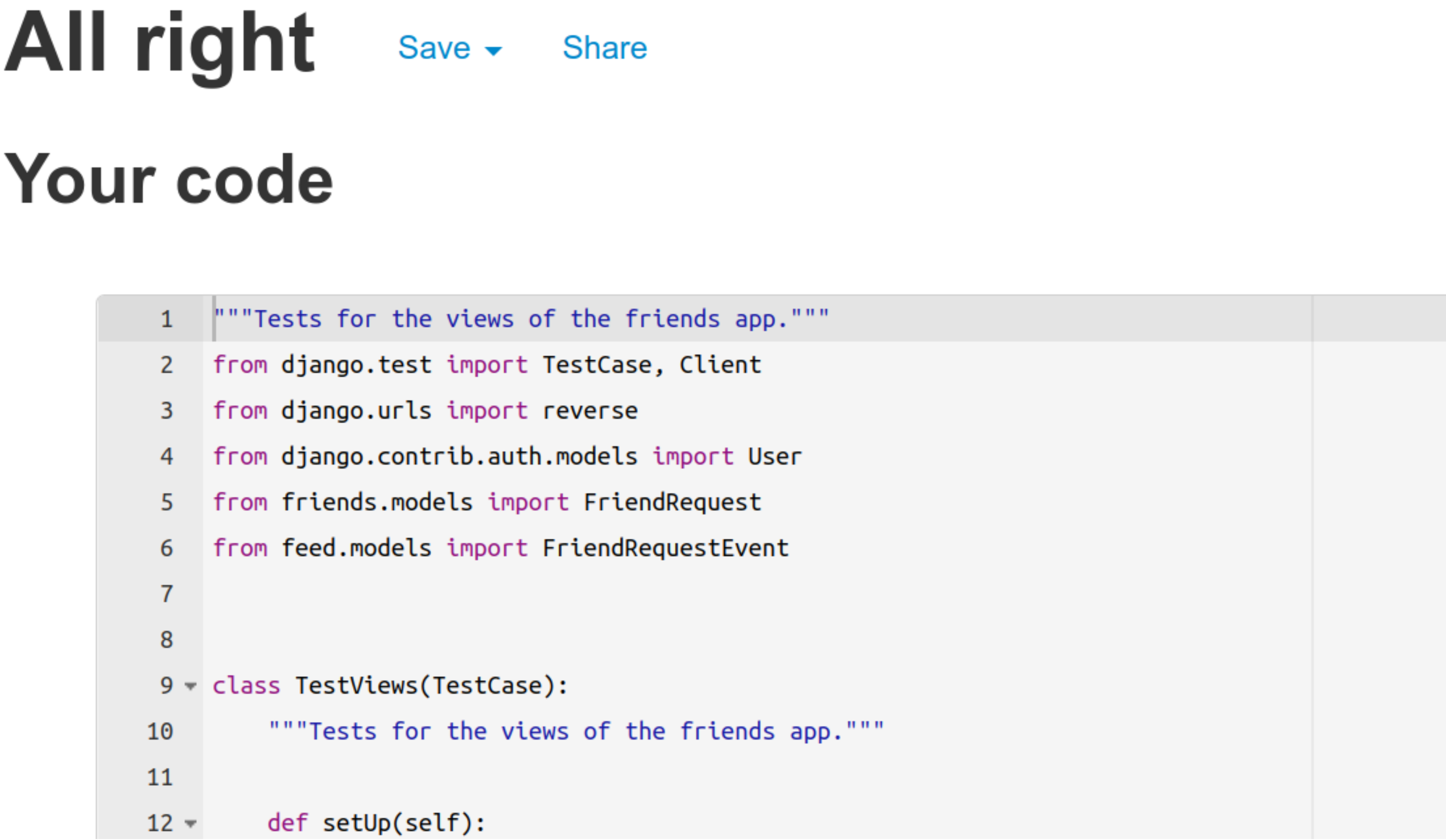
Your code

```python
"""Tests for the urls of the friends app"""
from django.test import SimpleTestCase
from django.urls import reverse, resolve
from friends.views import (
    SendFriendRequest,
    AcceptFriendRequest,
    DeclineFriendRequest,
    RemoveFriend,
    MyFriendsView,
    CancelFriendRequest
)
```

tests/test_views.py



```
friends
  migrations
  templates
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

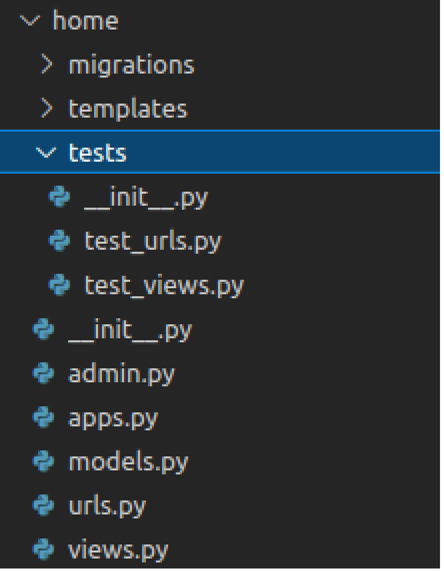


All right

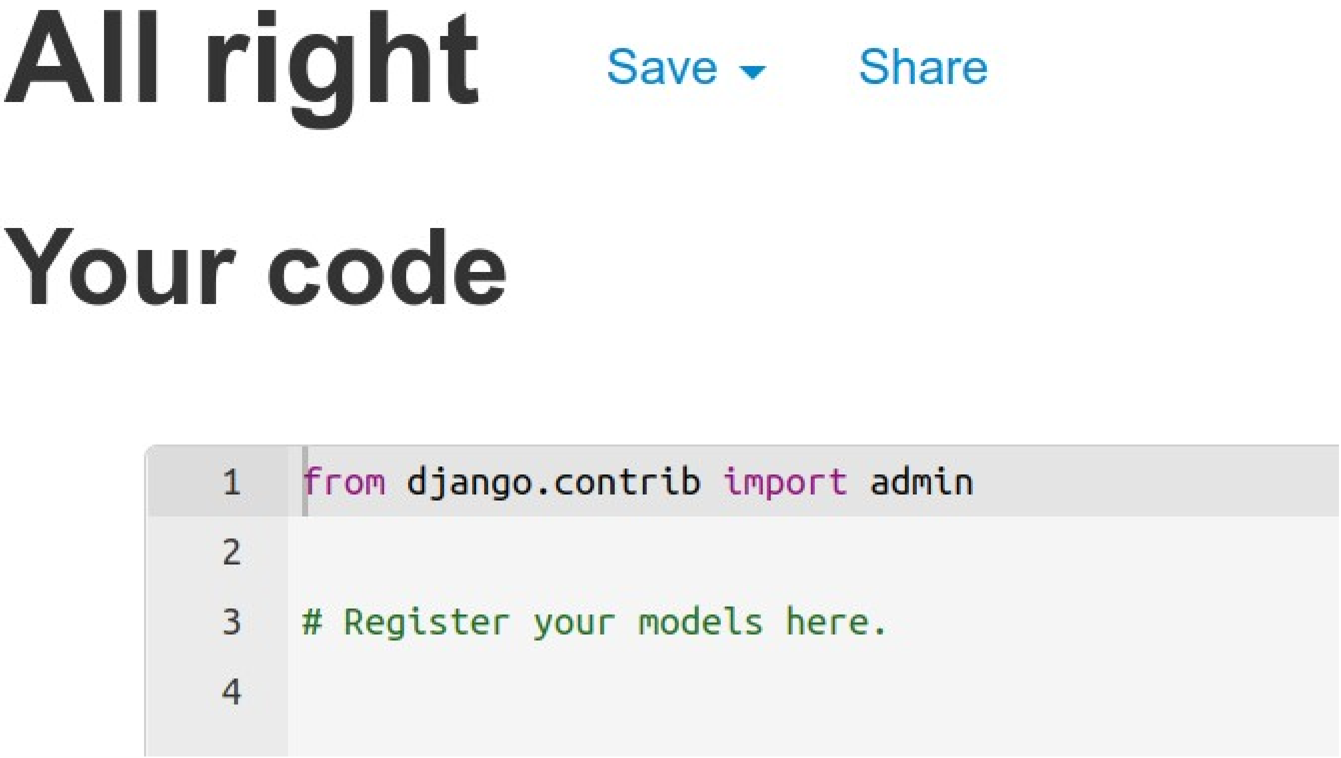
Your code

```python
"""Tests for the views of the friends app."""
from django.test import TestCase, Client
from django.urls import reverse
from django.contrib.auth.models import User
from friends.models import FriendRequest
from feed.models import FriendRequestEvent


class TestViews(TestCase):
    """Tests for the views of the friends app."""

    def setUp(self):
```

admin.py



```
home
  migrations
  templates
  tests
    __init__.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

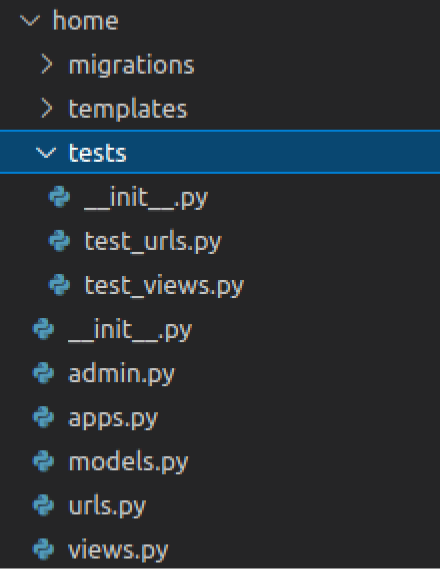


All right

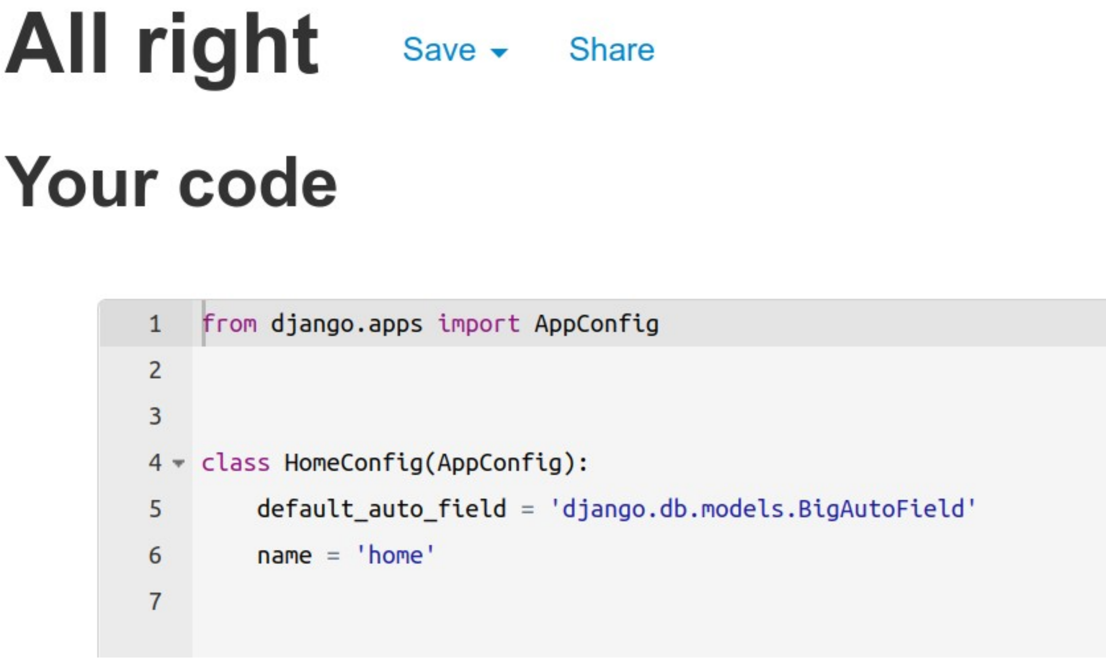
Your code

```python
from django.contrib import admin

# Register your models here.

```

apps.py



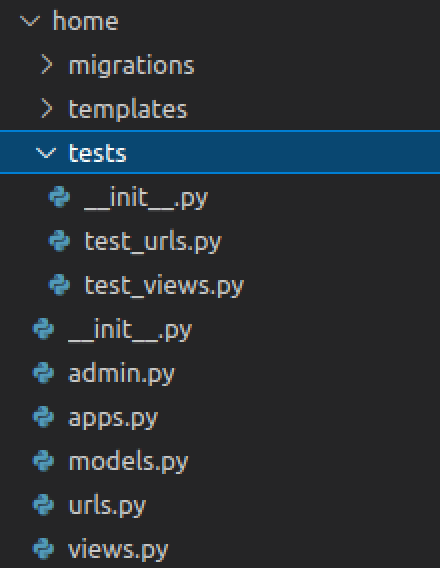
```
home
  migrations
  templates
  tests
    __init__.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```



All right

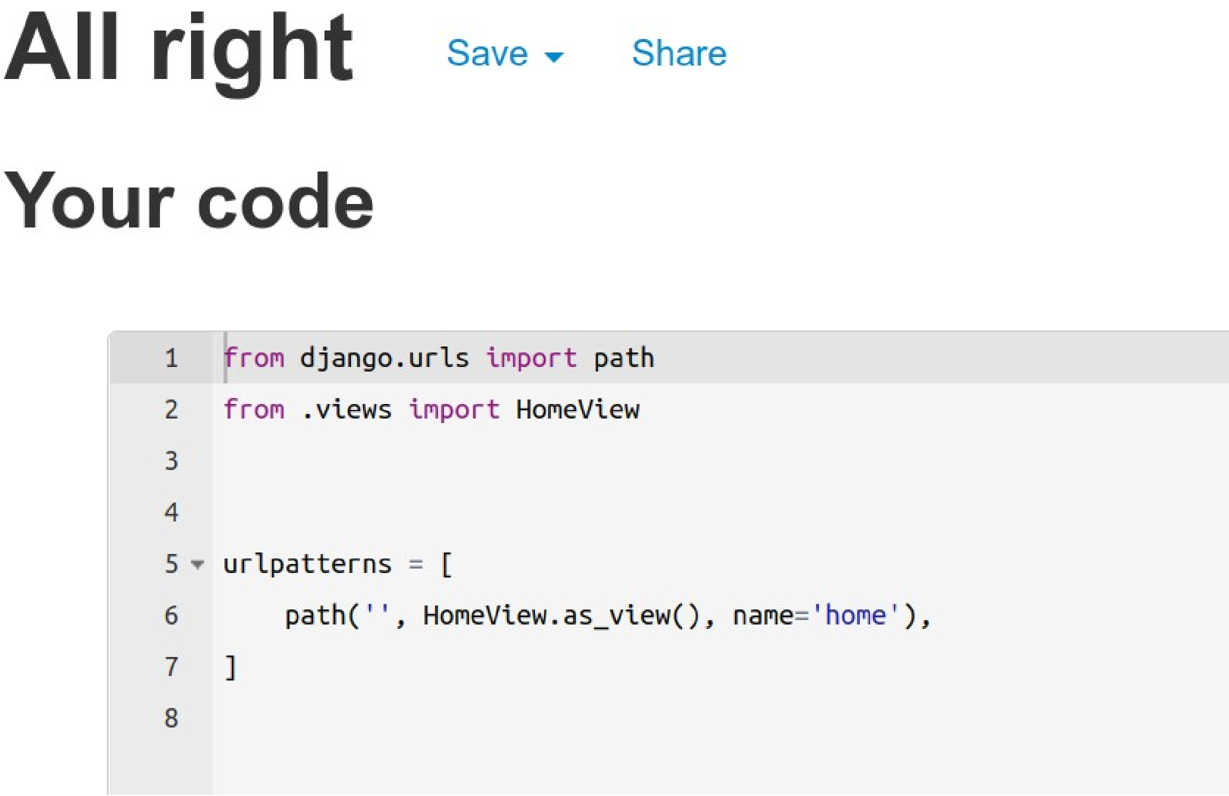
Your code

```python
from django.apps import AppConfig


class HomeConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'home'

```



```
home
  migrations
  templates
  tests
    __init__.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

urls.py

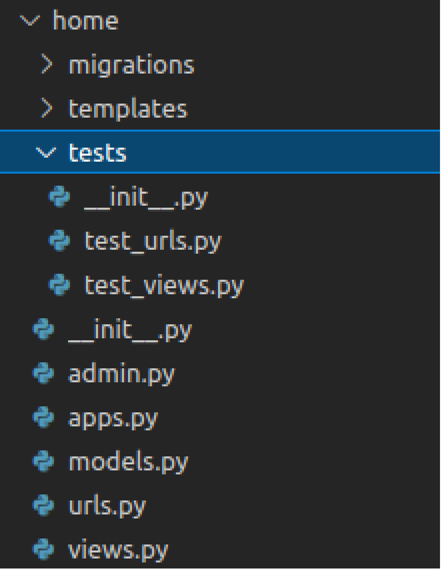


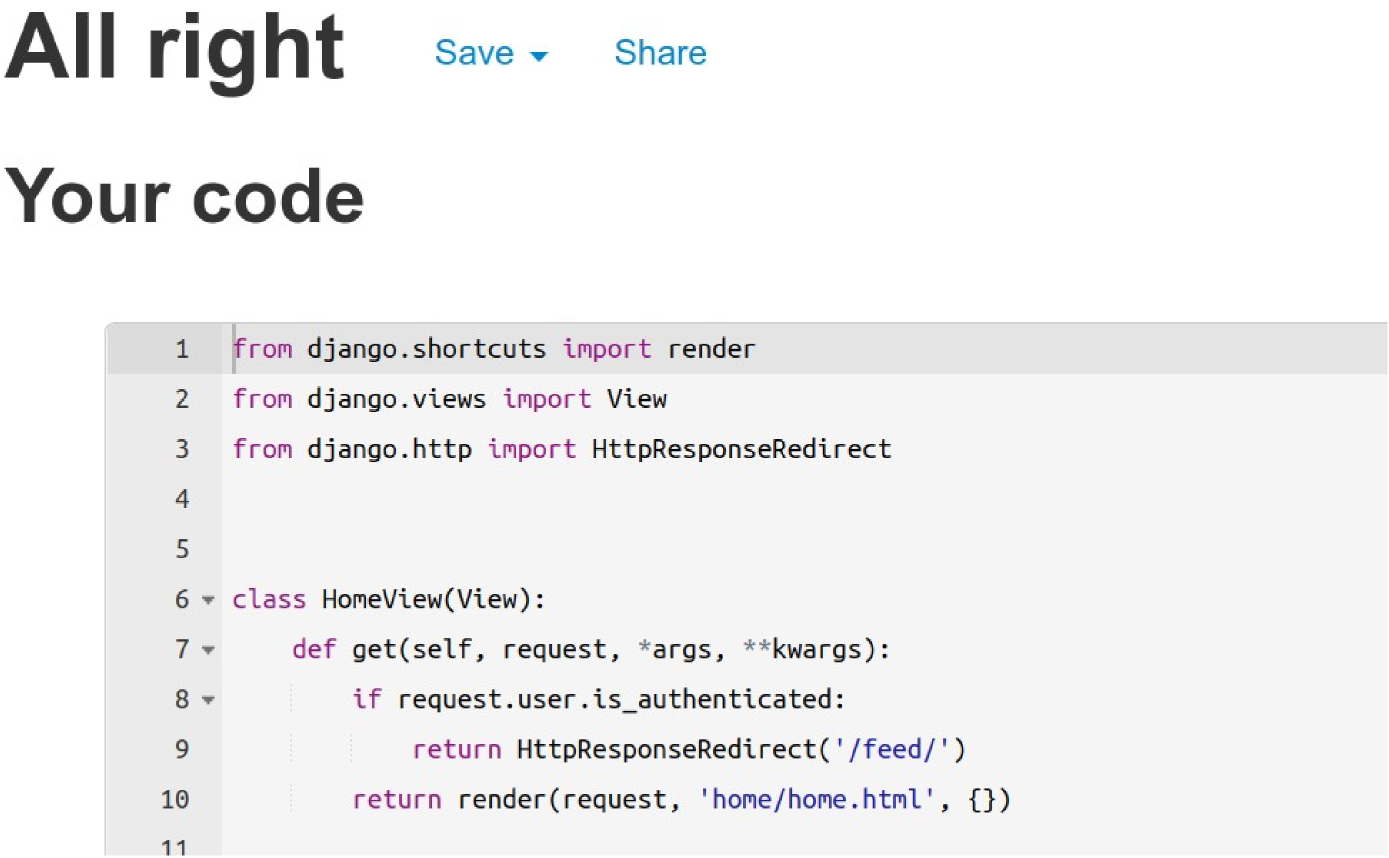
All right

Your code

```python
from django.urls import path
from .views import HomeView


urlpatterns = [
    path('', HomeView.as_view(), name='home'),
]

```

views.py



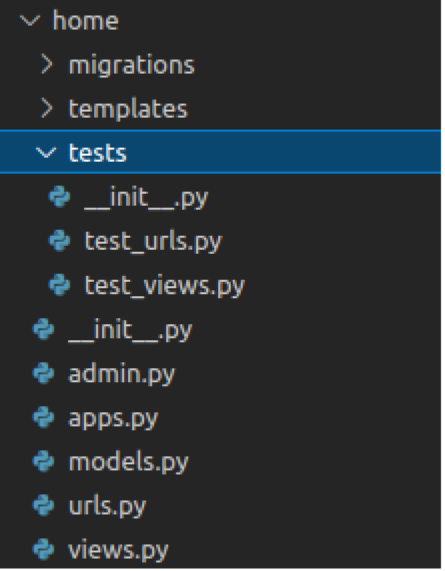


All right

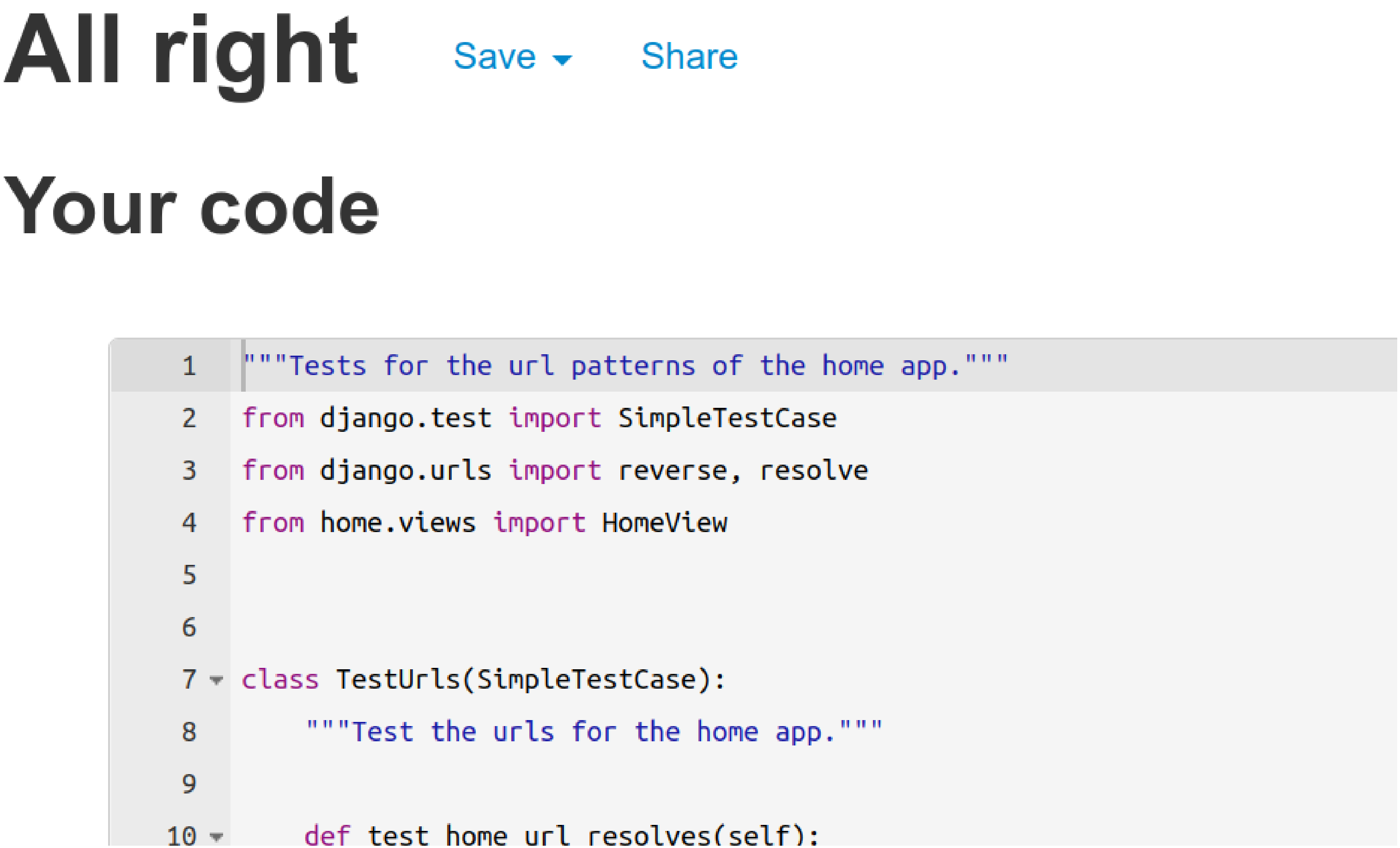
Your code

```python
from django.shortcuts import render
from django.views import View
from django.http import HttpResponseRedirect


class HomeView(View):
    def get(self, request, *args, **kwargs):
        if request.user.is_authenticated:
            return HttpResponseRedirect('/feed/')
        return render(request, 'home/home.html', {})

```

tests/test_urls.py



```
home
  migrations
  templates
  tests
    __init__.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

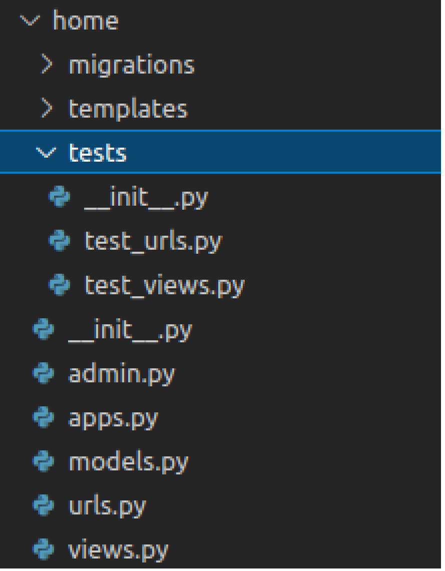


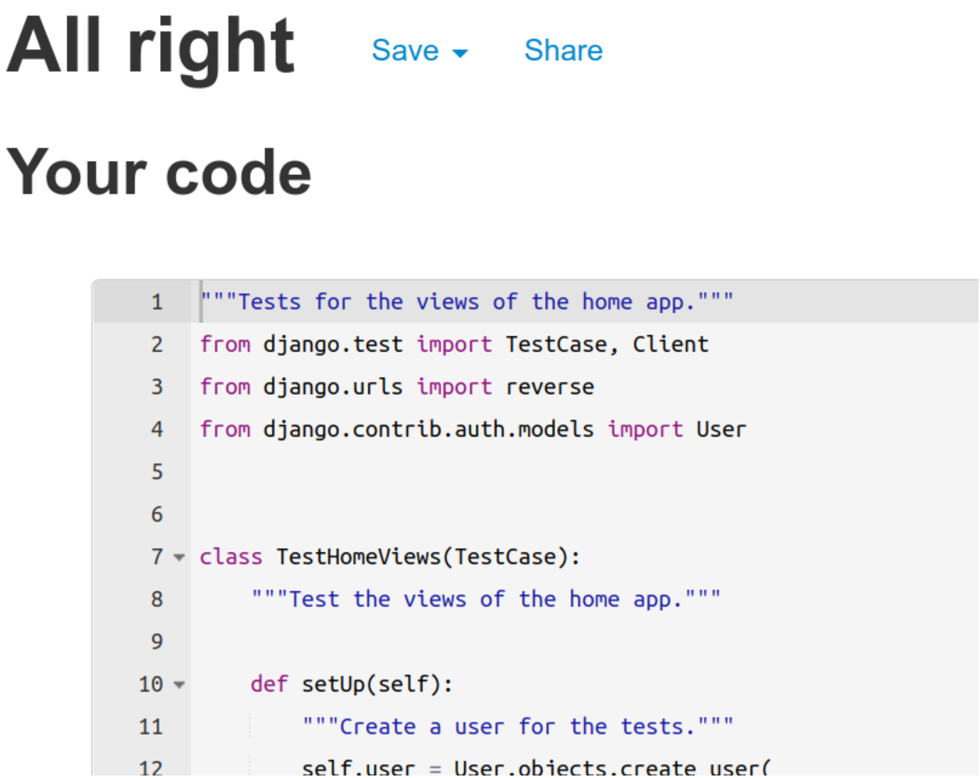
All right

Your code

```python
"""Tests for the url patterns of the home app."""
from django.test import SimpleTestCase
from django.urls import reverse, resolve
from home.views import HomeView


class TestUrls(SimpleTestCase):
    """Test the urls for the home app."""

    def test_home_url_resolves(self):
```

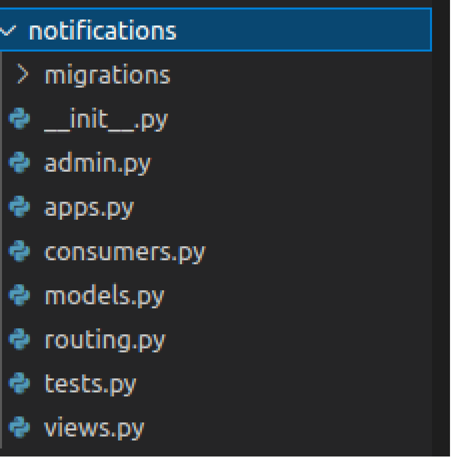
tests/test_views.py

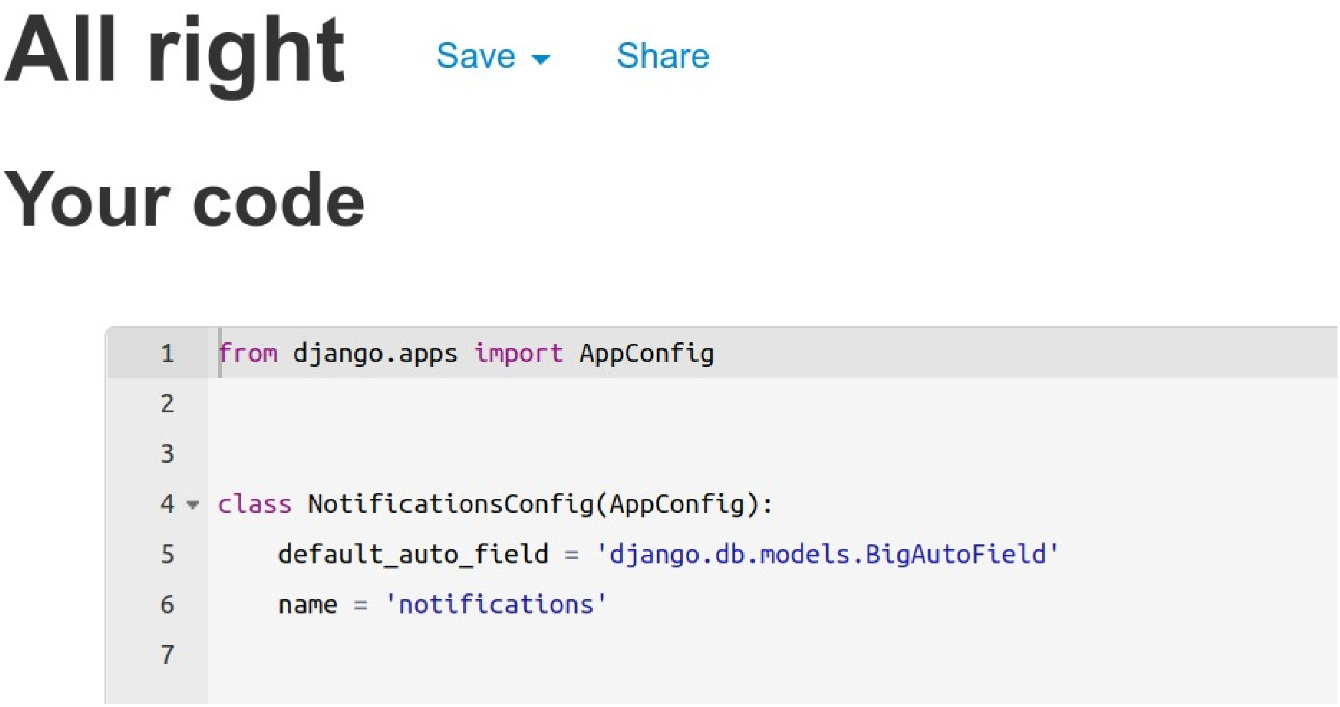




```python
"""Tests for the views of the home app."""
from django.test import TestCase, Client
from django.urls import reverse
from django.contrib.auth.models import User


class TestHomeViews(TestCase):
    """Test the views of the home app."""

    def setUp(self):
        """Create a user for the tests."""
        self.user = User.objects.create_user(
```

apps.py

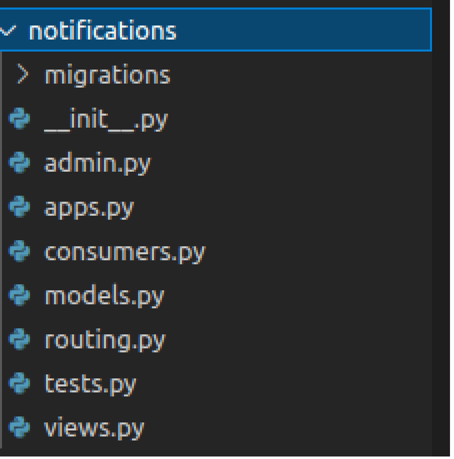


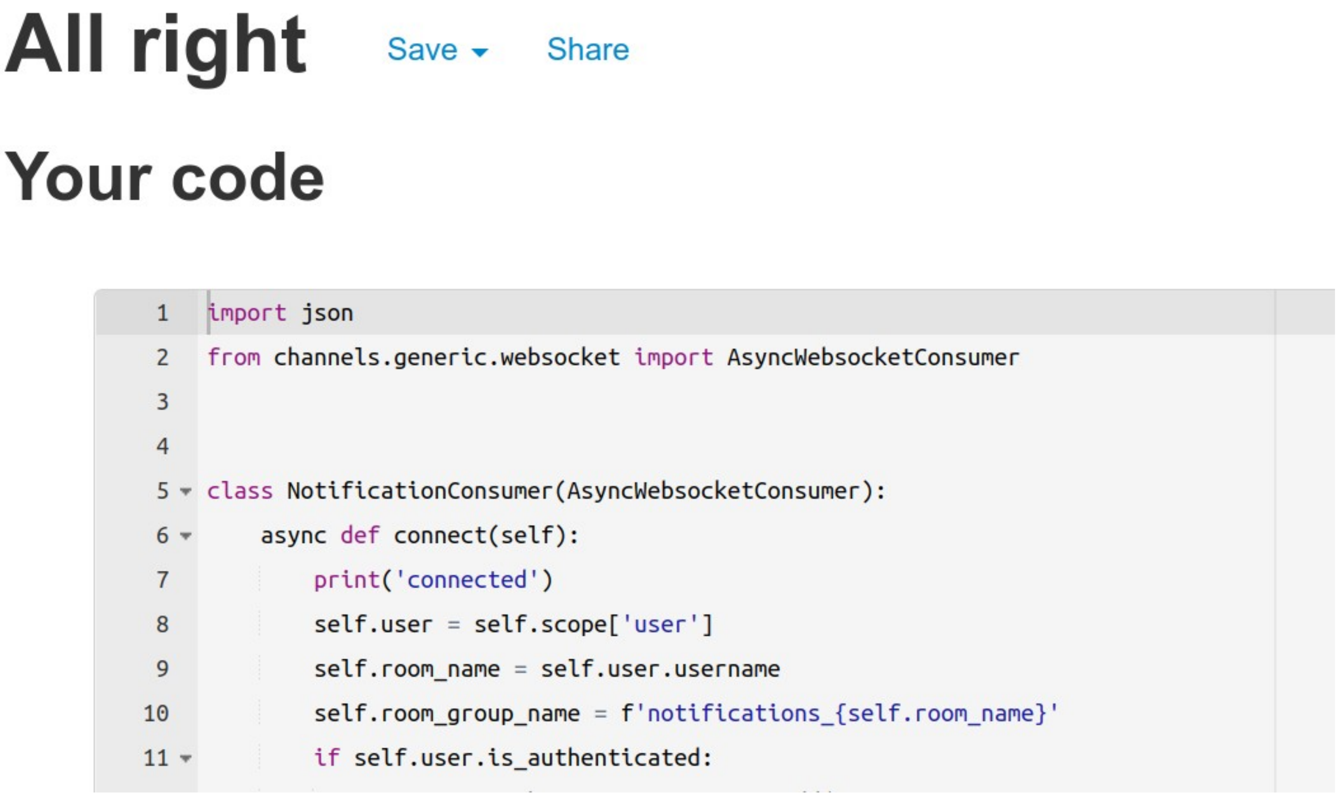


Your code

```python
from django.apps import AppConfig


class NotificationsConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'notifications'

```

consumers.py



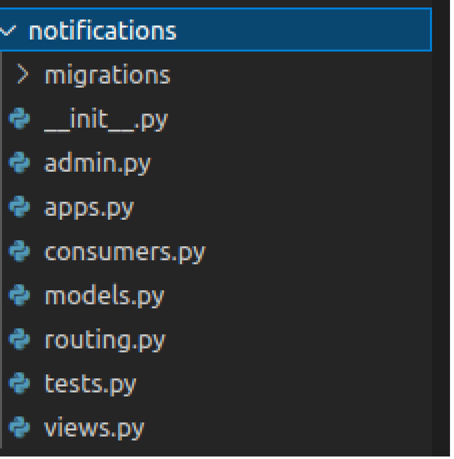


All right

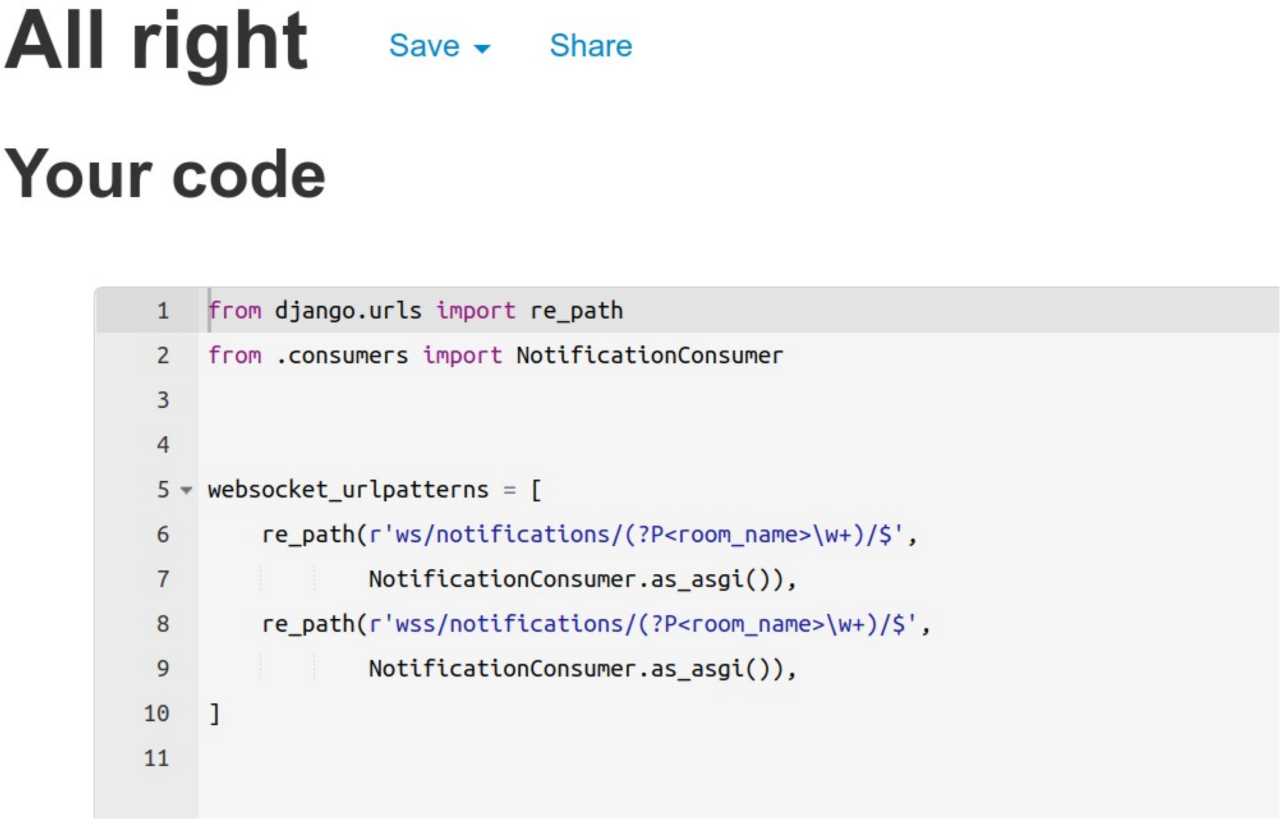
Your code

```
import json
from channels.generic.websocket import AsyncWebsocketConsumer


class NotificationConsumer(AsyncWebsocketConsumer):
    async def connect(self):
        print('connected')
        self.user = self.scope['user']
        self.room_name = self.user.username
        self.room_group_name = f'notifications_{self.room_name}'
        if self.user.is_authenticated:
```

routing.py



- notifications
  - migrations
  - __init__.py
  - admin.py
  - apps.py
  - consumers.py
  - models.py
  - routing.py
  - tests.py
  - views.py

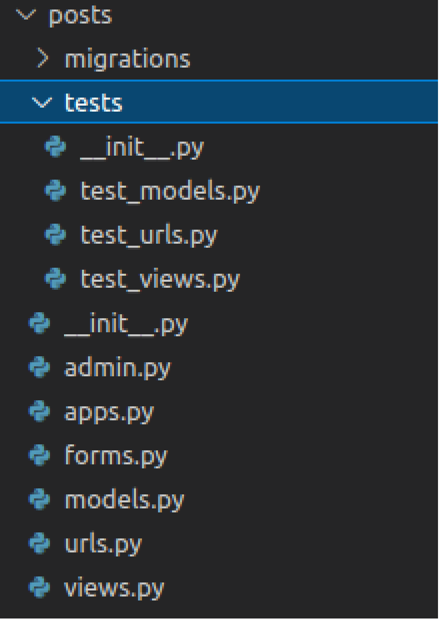


All right

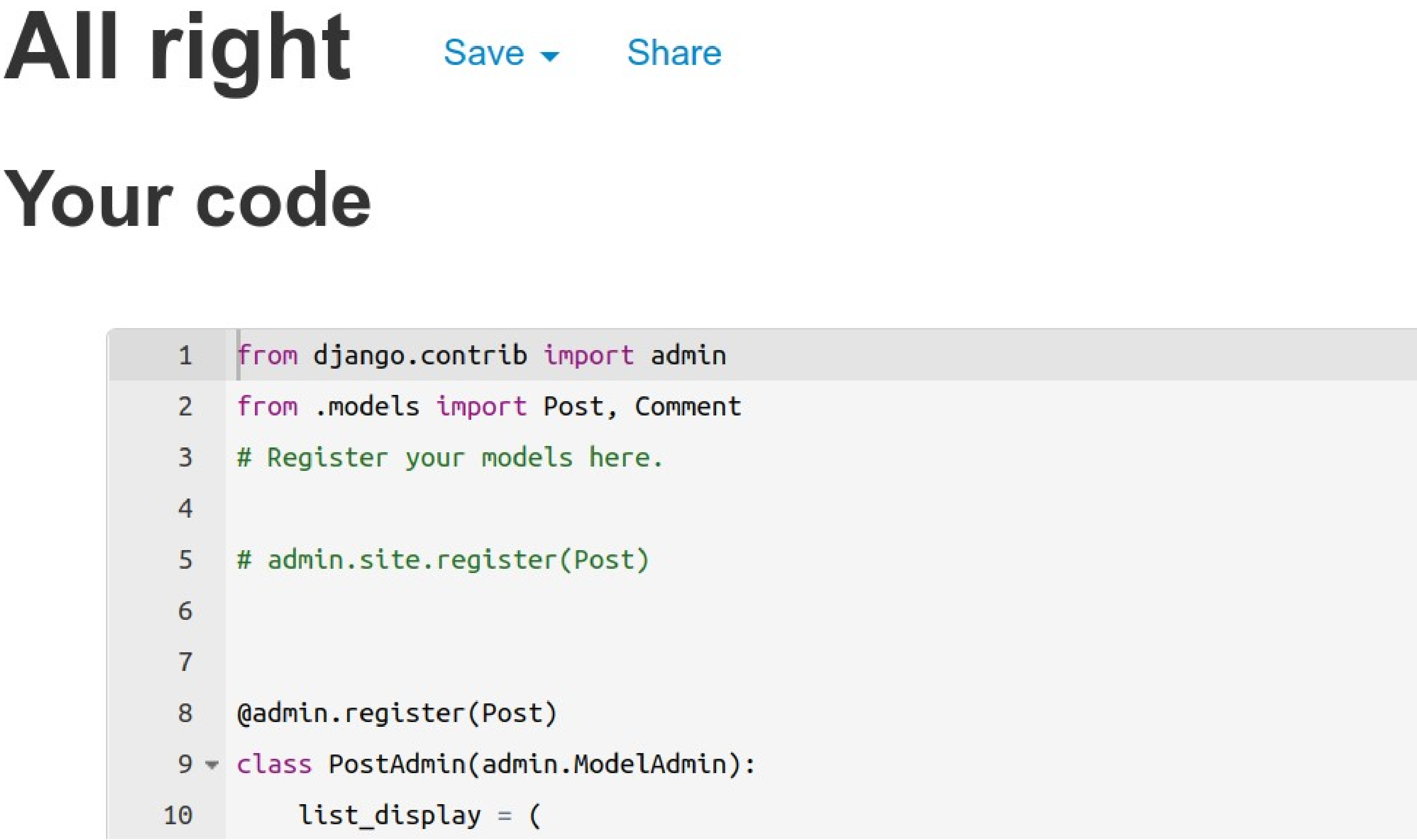
Your code

```python
from django.urls import re_path
from .consumers import NotificationConsumer


websocket_urlpatterns = [
    re_path(r'ws/notifications/(?P<room_name>\w+)/$',
            NotificationConsumer.as_asgi()),
    re_path(r'wss/notifications/(?P<room_name>\w+)/$',
            NotificationConsumer.as_asgi()),
]

```

admin.py



```
posts
  migrations
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```

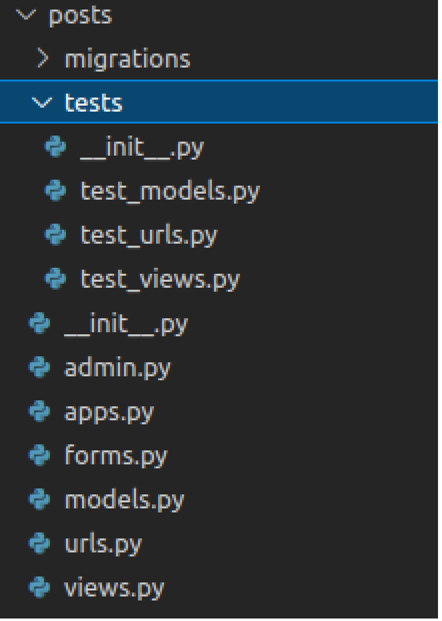


All right

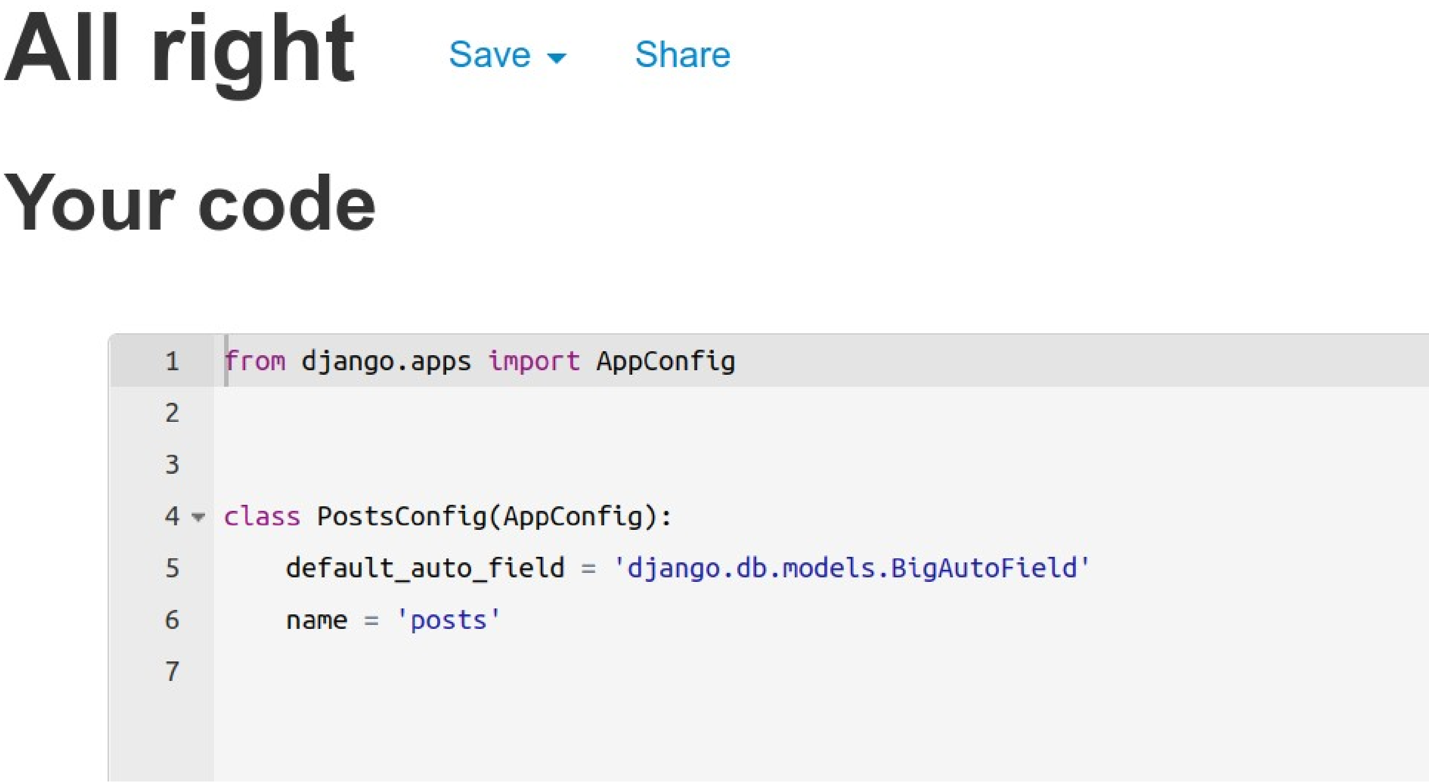
Your code

```python
from django.contrib import admin
from .models import Post, Comment
# Register your models here.

# admin.site.register(Post)



@admin.register(Post)
class PostAdmin(admin.ModelAdmin):
    list_display = (
```

apps.py



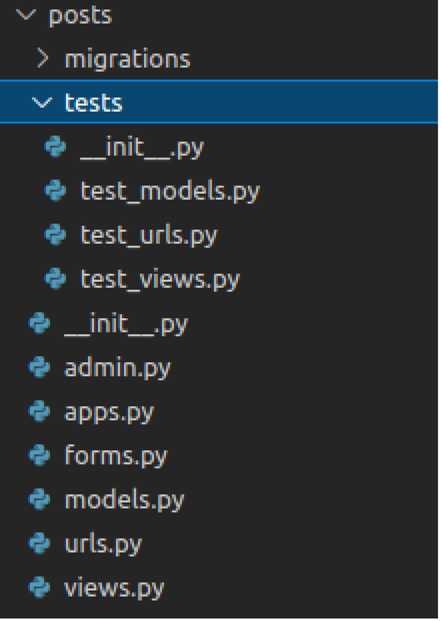
```
posts
  migrations
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```



All right

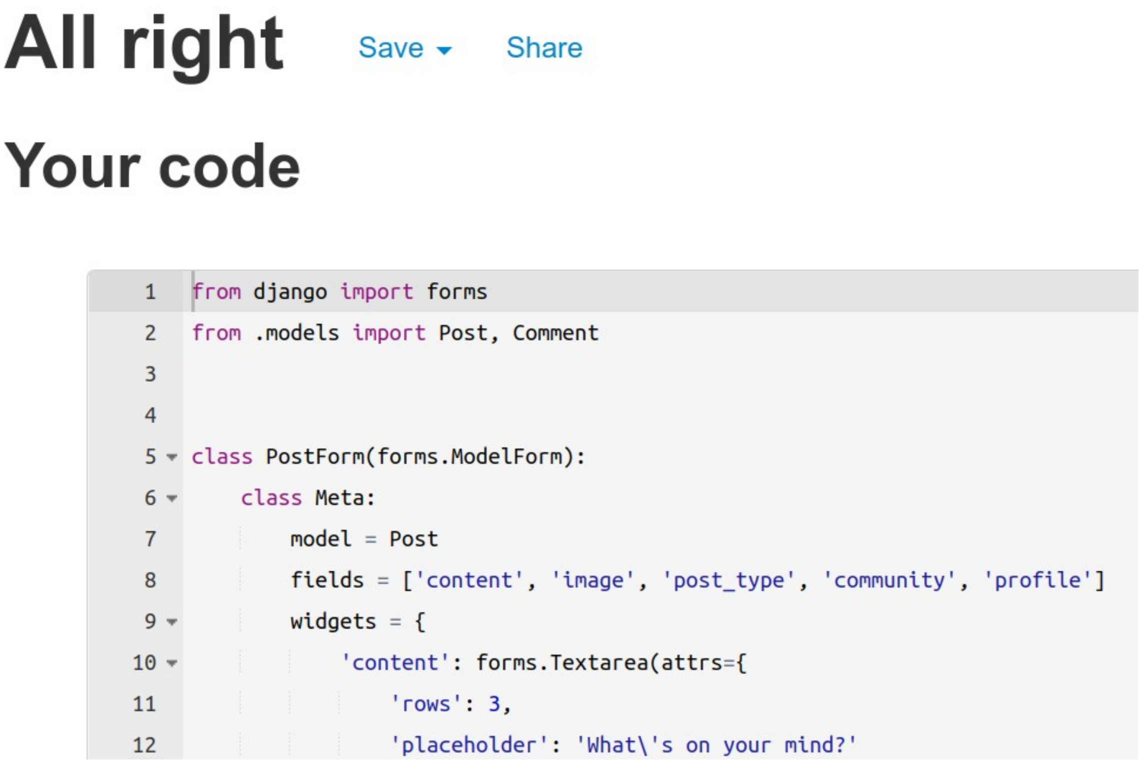
Your code

```python
from django.apps import AppConfig


class PostsConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'posts'

```



- posts
  - migrations
  - tests
    - __init__.py
    - test_models.py
    - test_urls.py
    - test_views.py
  - __init__.py
  - admin.py
  - apps.py
  - forms.py
  - models.py
  - urls.py
  - views.py

forms.py

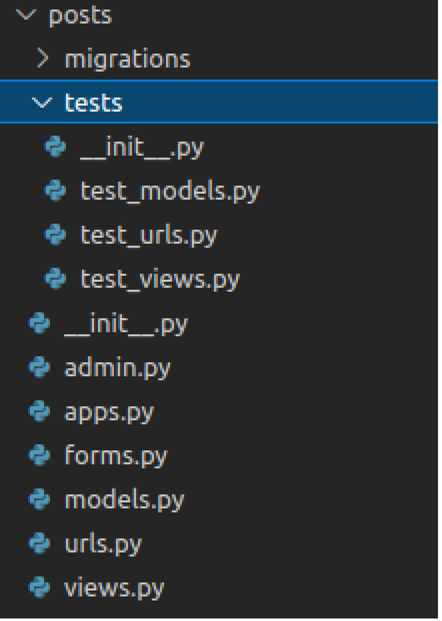


All right

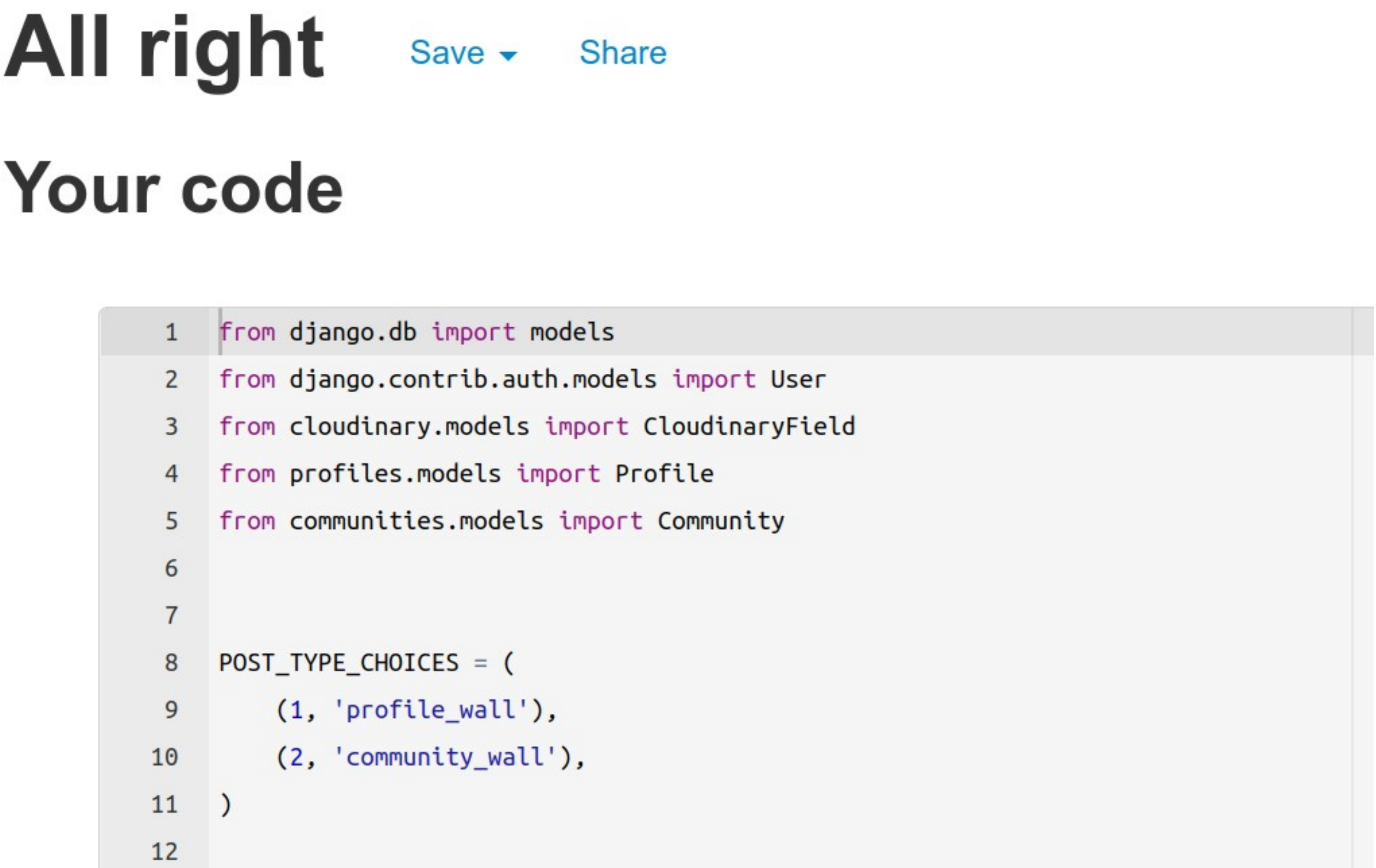
Your code

```python
from django import forms
from .models import Post, Comment


class PostForm(forms.ModelForm):
    class Meta:
        model = Post
        fields = ['content', 'image', 'post_type', 'community', 'profile']
        widgets = {
            'content': forms.Textarea(attrs={
                'rows': 3,
                'placeholder': 'What\'s on your mind?'
```

models.py



```
posts
  migrations
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```

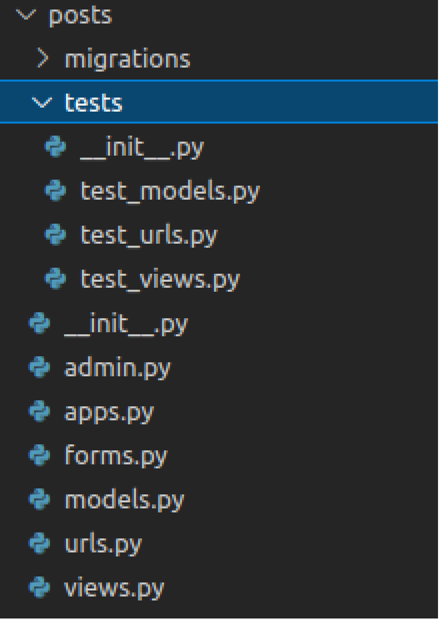


All right

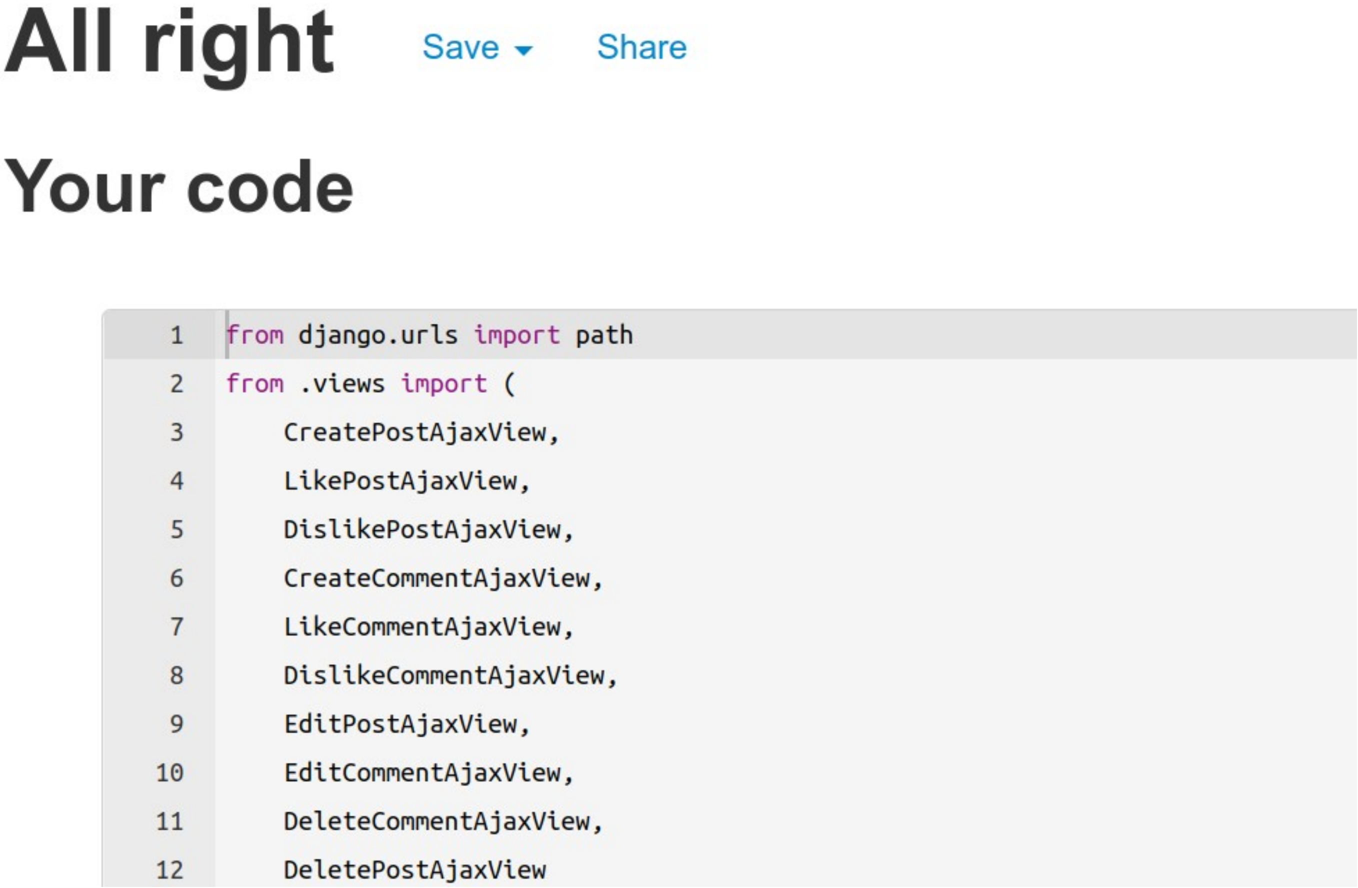
Your code

```python
from django.db import models
from django.contrib.auth.models import User
from cloudinary.models import CloudinaryField
from profiles.models import Profile
from communities.models import Community


POST_TYPE_CHOICES = (
    (1, 'profile_wall'),
    (2, 'community_wall'),
)

```

urls.py



```
posts
  migrations
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```

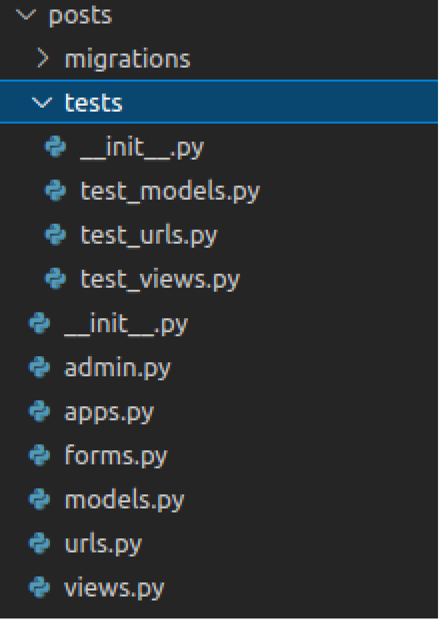


All right

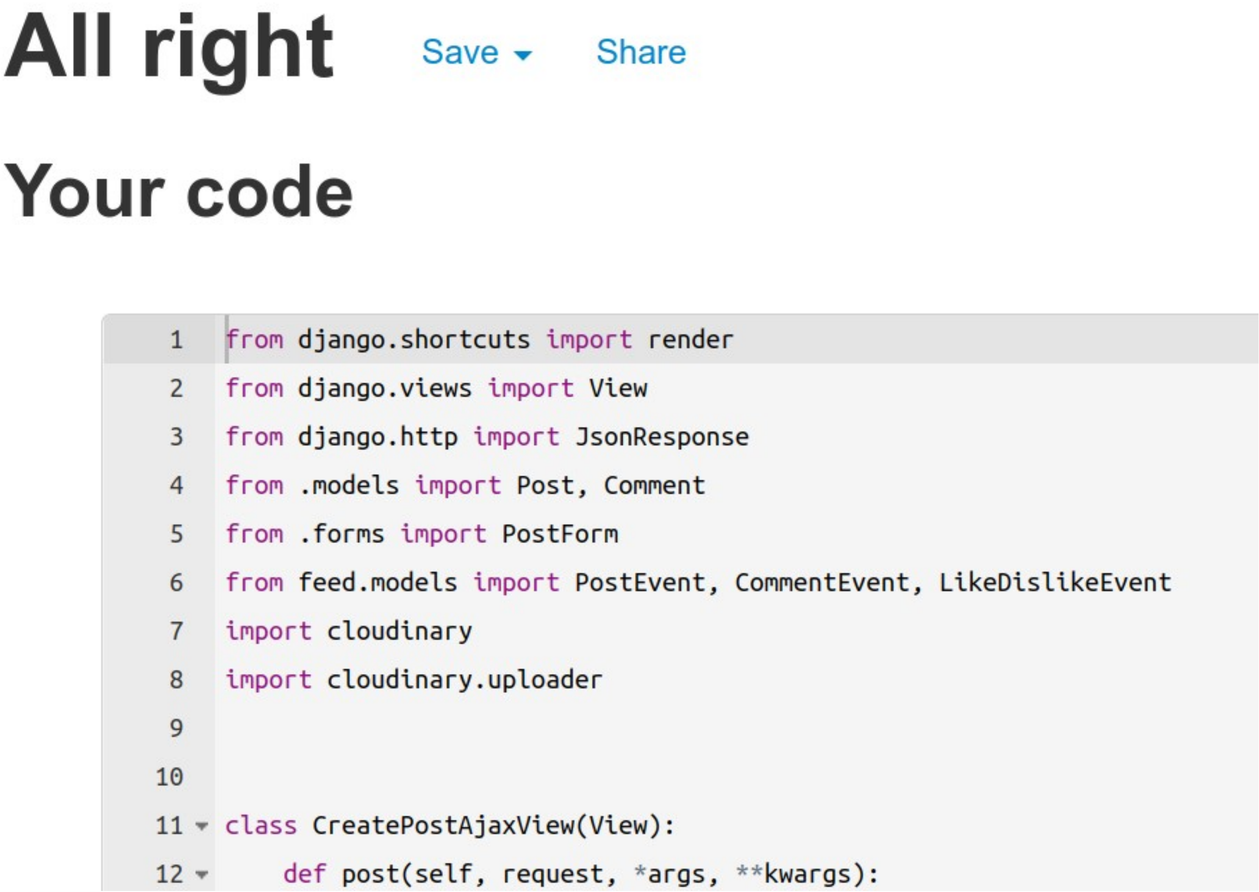
Your code

```python
from django.urls import path
from .views import (
    CreatePostAjaxView,
    LikePostAjaxView,
    DislikePostAjaxView,
    CreateCommentAjaxView,
    LikeCommentAjaxView,
    DislikeCommentAjaxView,
    EditPostAjaxView,
    EditCommentAjaxView,
    DeleteCommentAjaxView,
    DeletePostAjaxView
```

views.py



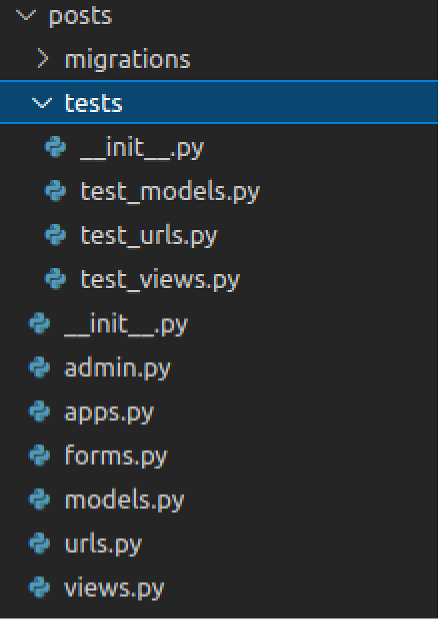
```
posts
  migrations
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```



All right

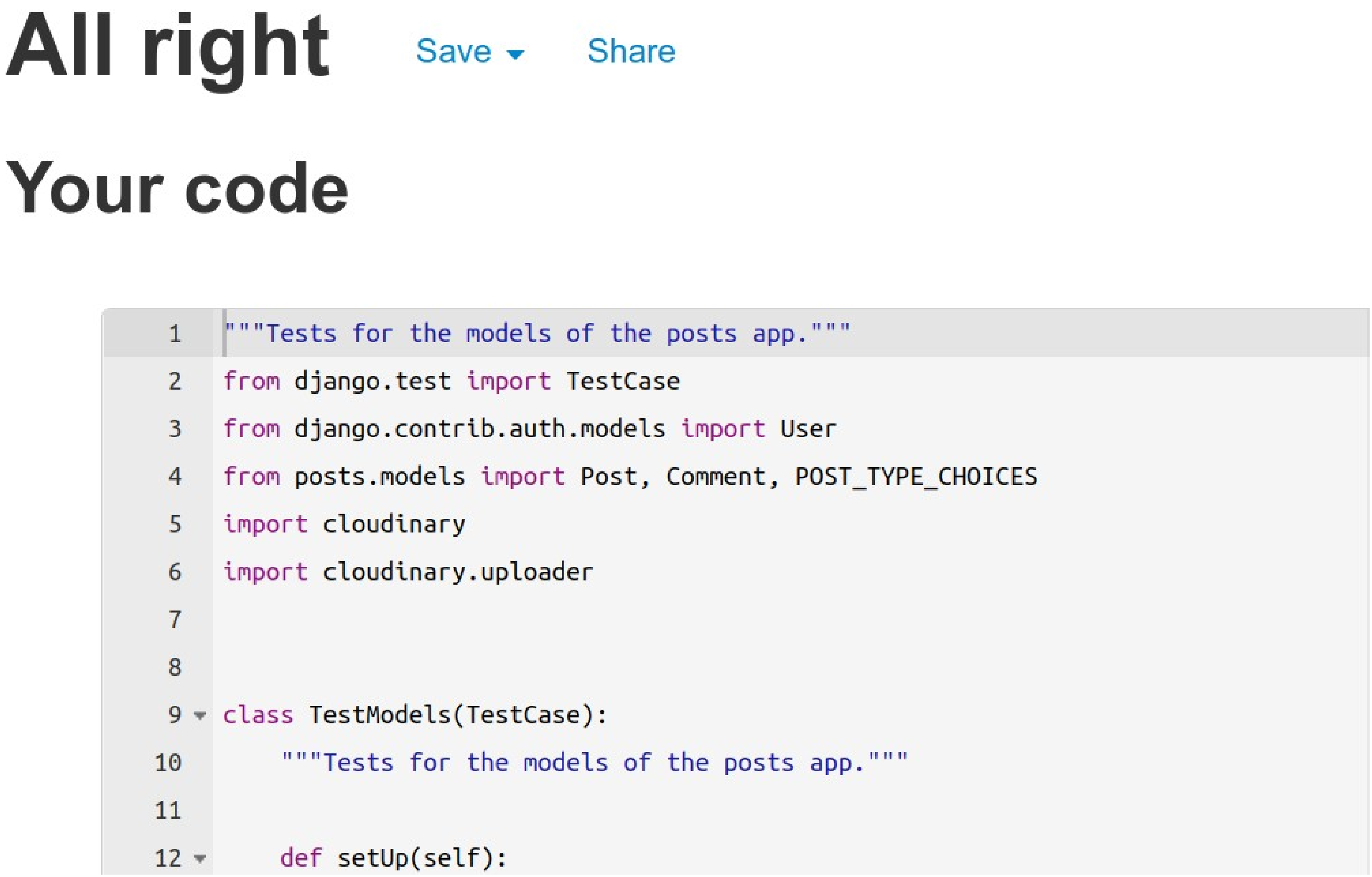
Your code

```python
from django.shortcuts import render
from django.views import View
from django.http import JsonResponse
from .models import Post, Comment
from .forms import PostForm
from feed.models import PostEvent, CommentEvent, LikeDislikeEvent
import cloudinary
import cloudinary.uploader


class CreatePostAjaxView(View):
    def post(self, request, *args, **kwargs):
```



```
posts
  migrations
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```

tests/test_models.py

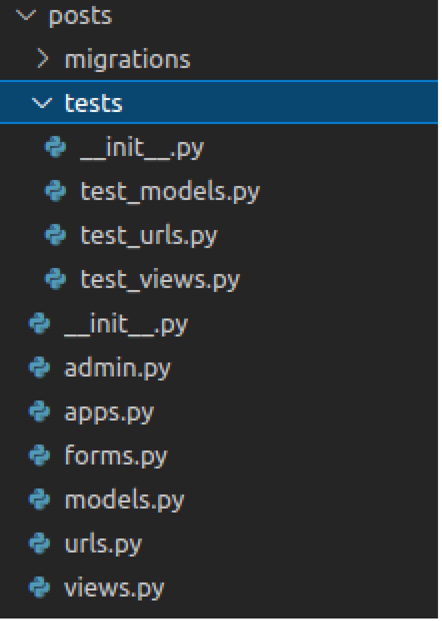


All right

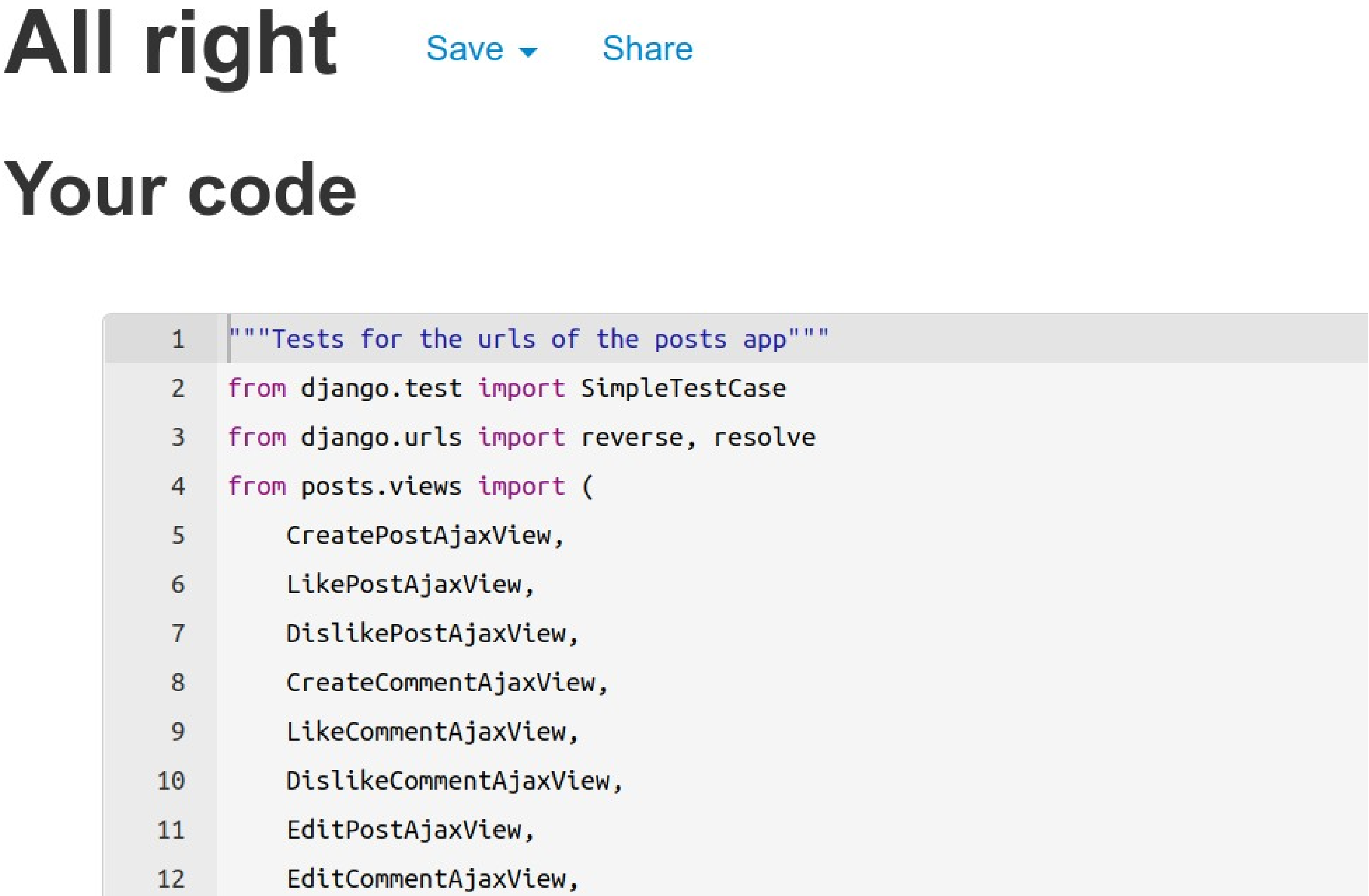
Your code

```python
"""Tests for the models of the posts app."""
from django.test import TestCase
from django.contrib.auth.models import User
from posts.models import Post, Comment, POST_TYPE_CHOICES
import cloudinary
import cloudinary.uploader


class TestModels(TestCase):
    """Tests for the models of the posts app."""

    def setUp(self):
```

tests/test_urls.py



```
posts
  migrations
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```

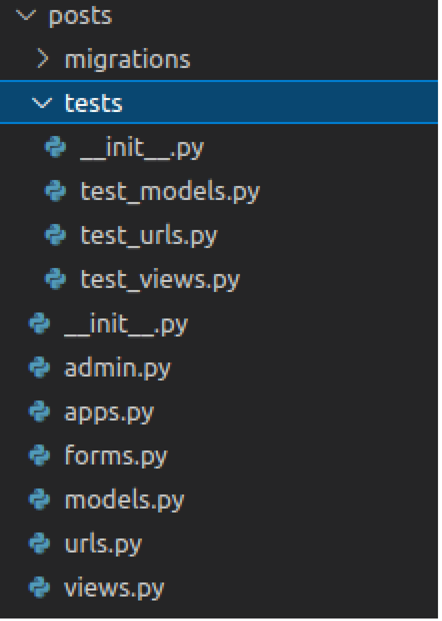


All right

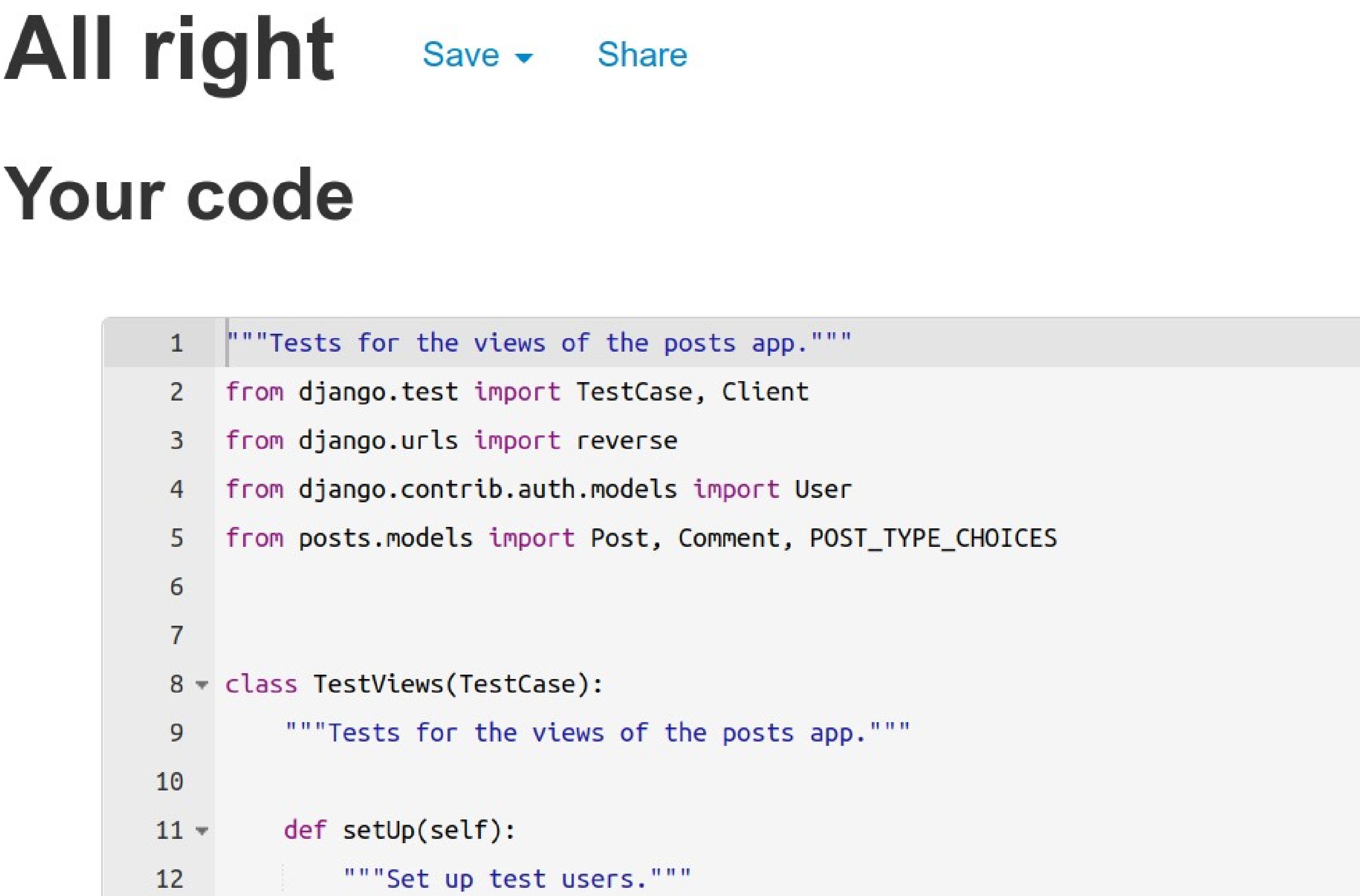
Your code

```python
"""Tests for the urls of the posts app"""
from django.test import SimpleTestCase
from django.urls import reverse, resolve
from posts.views import (
    CreatePostAjaxView,
    LikePostAjaxView,
    DislikePostAjaxView,
    CreateCommentAjaxView,
    LikeCommentAjaxView,
    DislikeCommentAjaxView,
    EditPostAjaxView,
    EditCommentAjaxView,
```

tests/test_views.py



```
posts
  migrations
  tests
    __init__.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  urls.py
  views.py
```

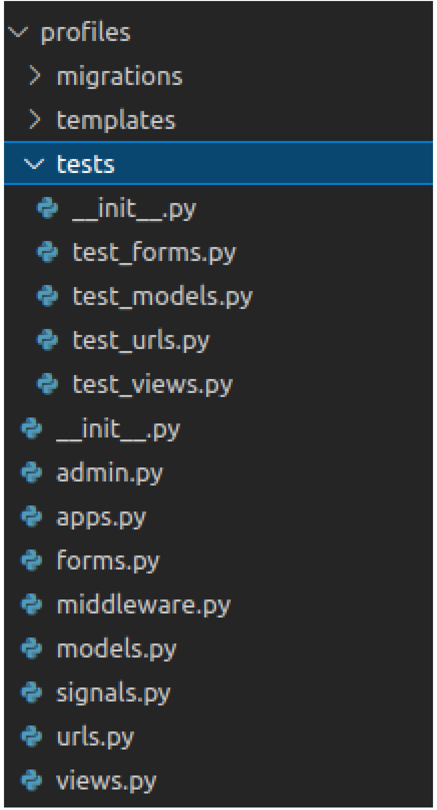


All right

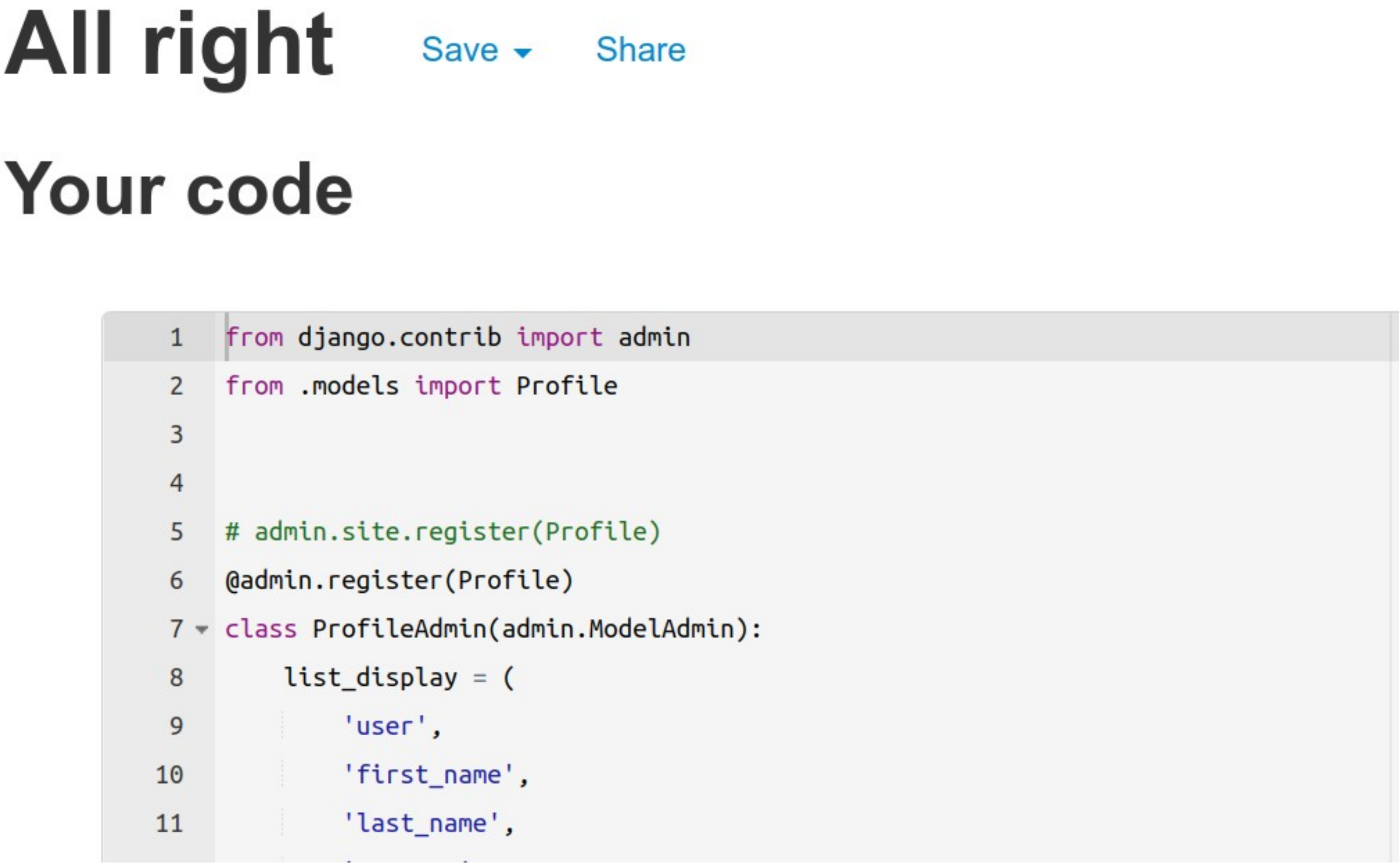
Your code

```python
"""Tests for the views of the posts app."""
from django.test import TestCase, Client
from django.urls import reverse
from django.contrib.auth.models import User
from posts.models import Post, Comment, POST_TYPE_CHOICES


class TestViews(TestCase):
    """Tests for the views of the posts app."""

    def setUp(self):
        """Set up test users."""
```

admin.py



```
profiles
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  middleware.py
  models.py
  signals.py
  urls.py
  views.py
```

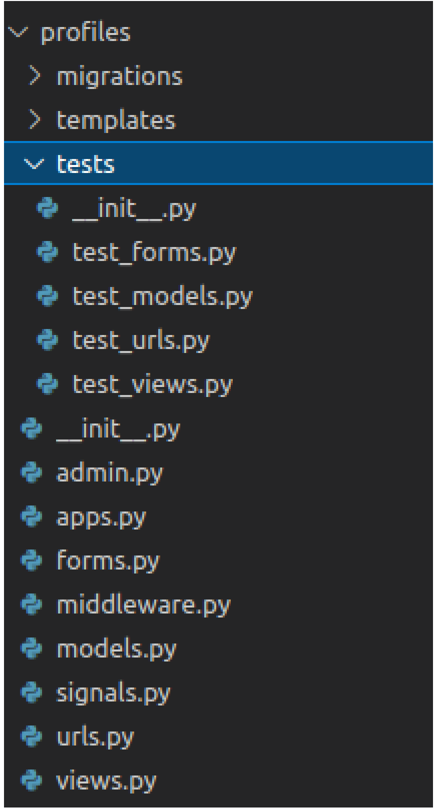


All right

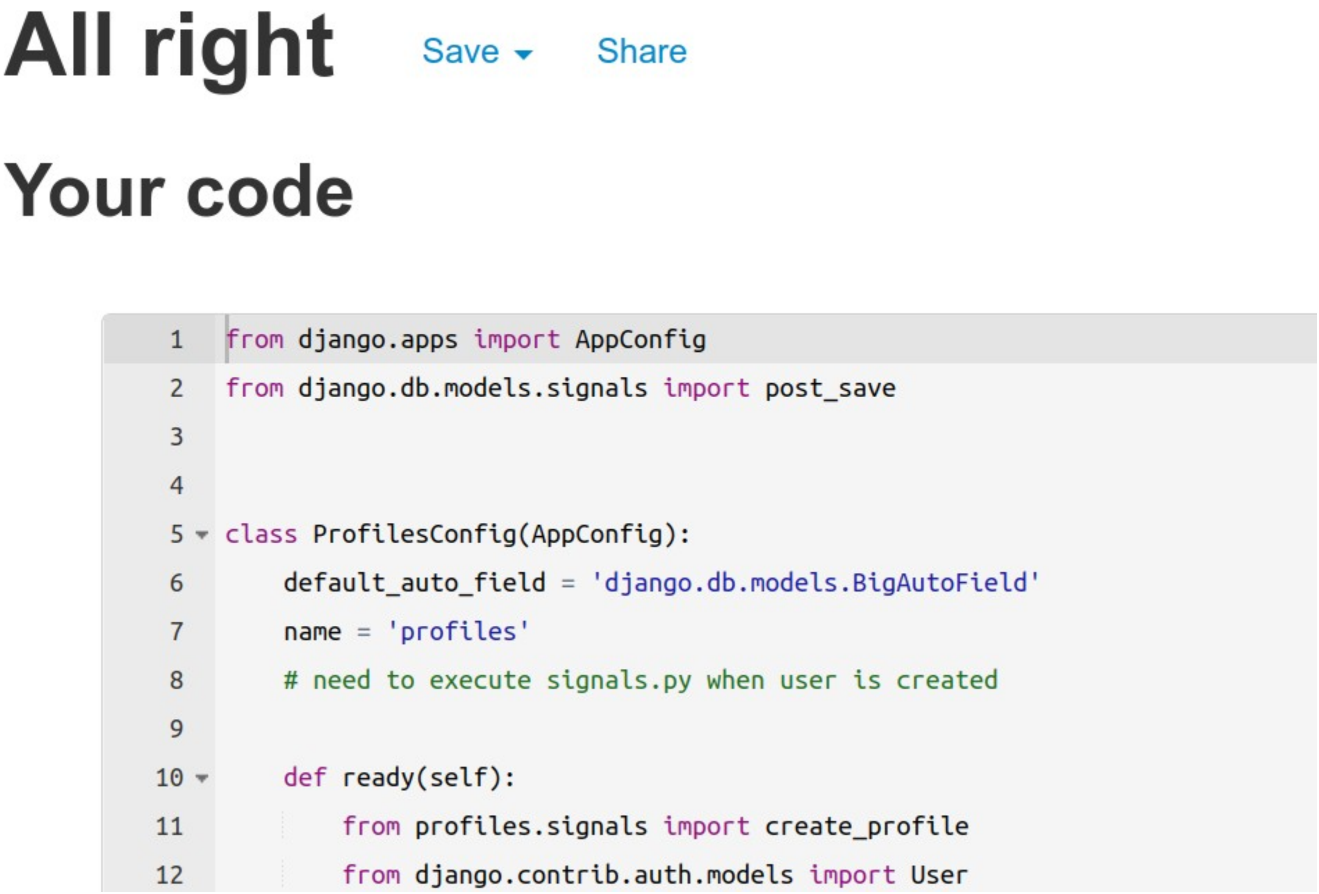
Your code

```python
from django.contrib import admin
from .models import Profile


# admin.site.register(Profile)
@admin.register(Profile)
class ProfileAdmin(admin.ModelAdmin):
    list_display = (
        'user',
        'first_name',
        'last_name',
```

apps.py



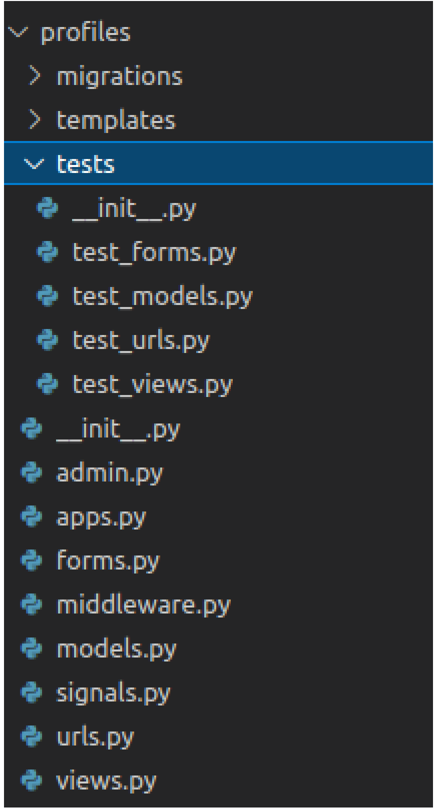
```
profiles
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  middleware.py
  models.py
  signals.py
  urls.py
  views.py
```



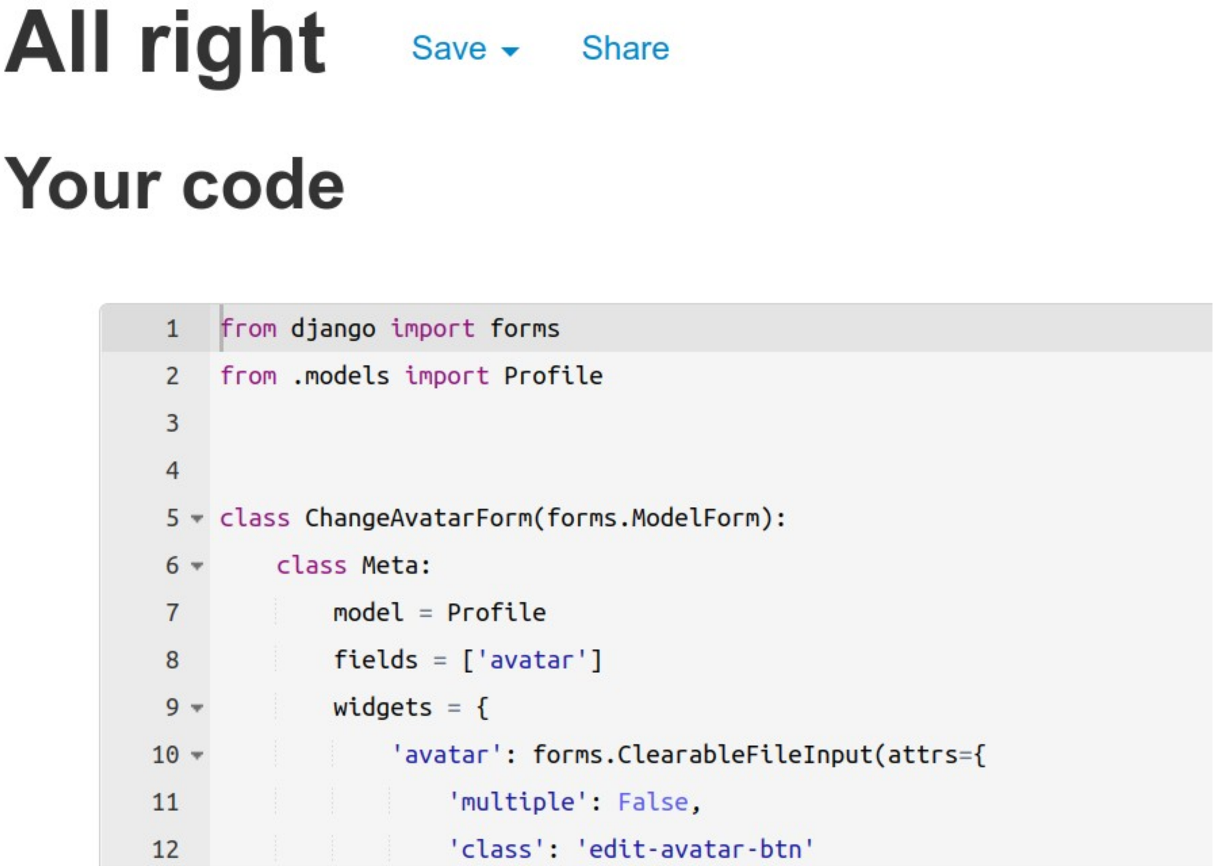
Your code

```python
from django.apps import AppConfig
from django.db.models.signals import post_save


class ProfilesConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'profiles'
    # need to execute signals.py when user is created

    def ready(self):
        from profiles.signals import create_profile
        from django.contrib.auth.models import User
```

forms.py



```
profiles
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  middleware.py
  models.py
  signals.py
  urls.py
  views.py
```

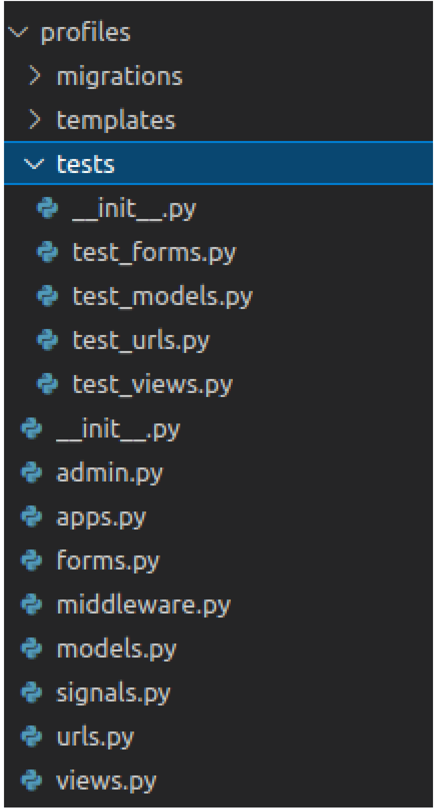


All right

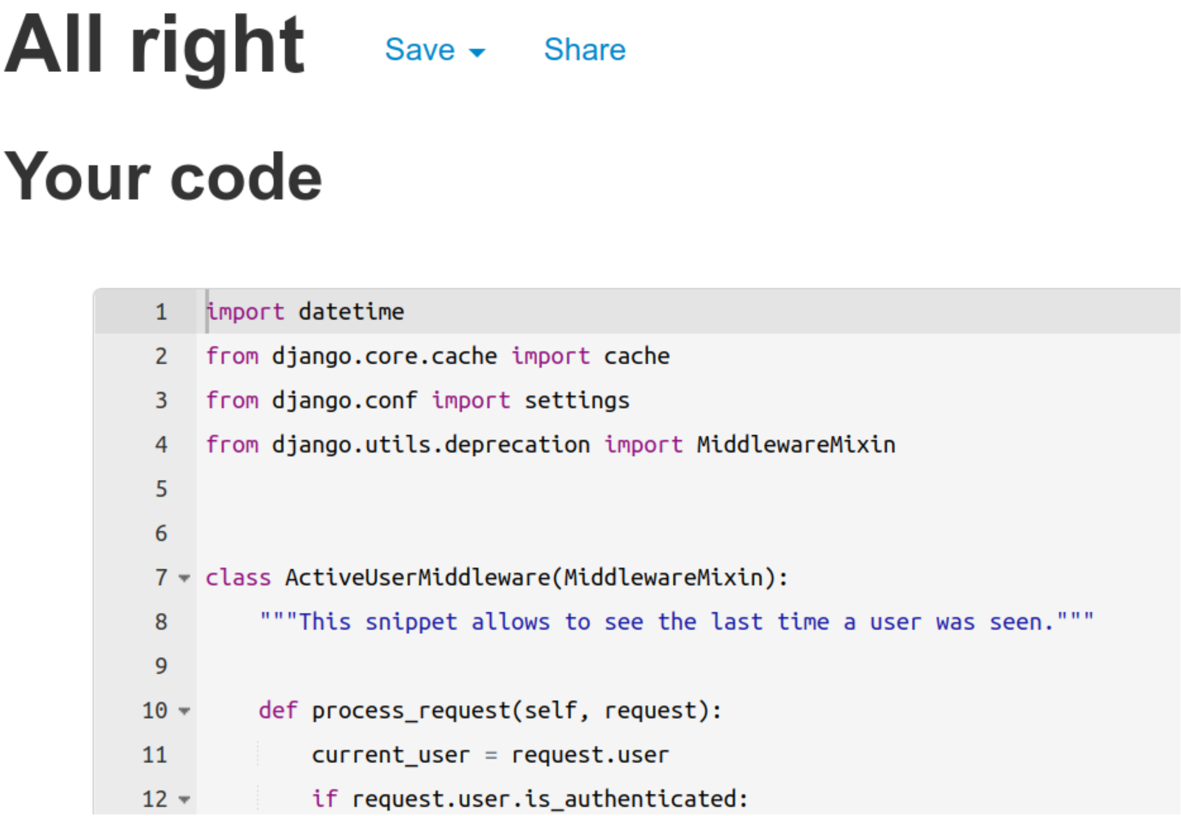
Your code

```python
from django import forms
from .models import Profile


class ChangeAvatarForm(forms.ModelForm):
    class Meta:
        model = Profile
        fields = ['avatar']
        widgets = {
            'avatar': forms.ClearableFileInput(attrs={
                'multiple': False,
                'class': 'edit-avatar-btn'
```

middleware.py



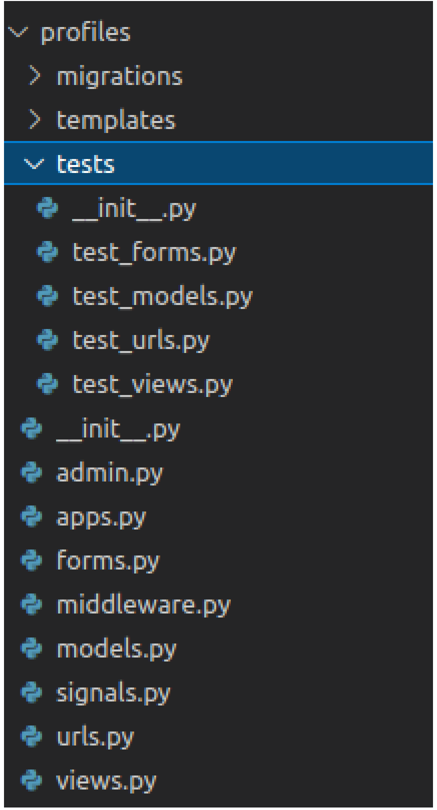
```
profiles
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  middleware.py
  models.py
  signals.py
  urls.py
  views.py
```



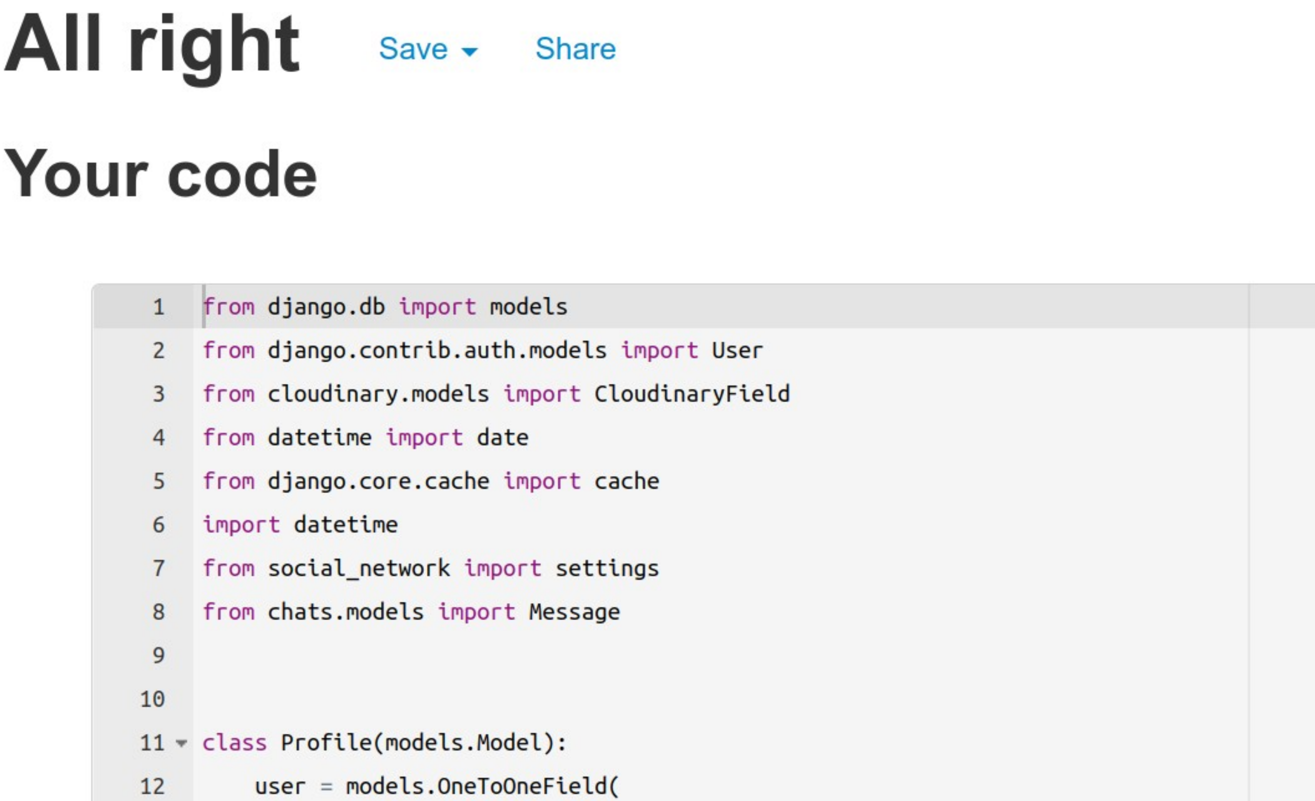
Your code

```python
import datetime
from django.core.cache import cache
from django.conf import settings
from django.utils.deprecation import MiddlewareMixin


class ActiveUserMiddleware(MiddlewareMixin):
    """This snippet allows to see the last time a user was seen."""

    def process_request(self, request):
        current_user = request.user
        if request.user.is_authenticated:
```

models.py



```
profiles
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  middleware.py
  models.py
  signals.py
  urls.py
  views.py
```

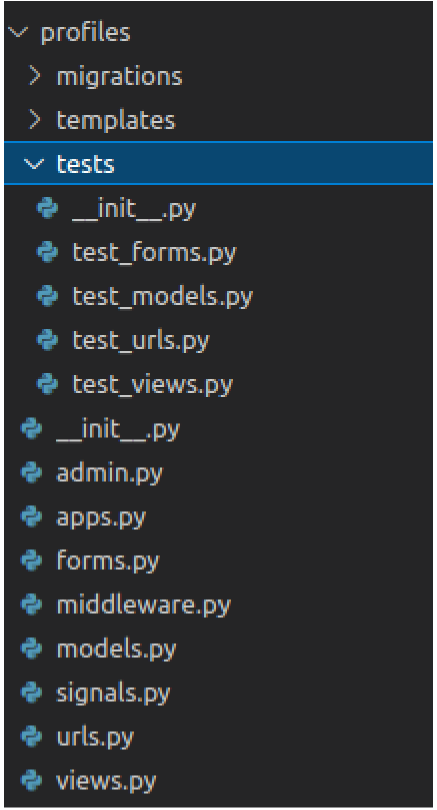


All right

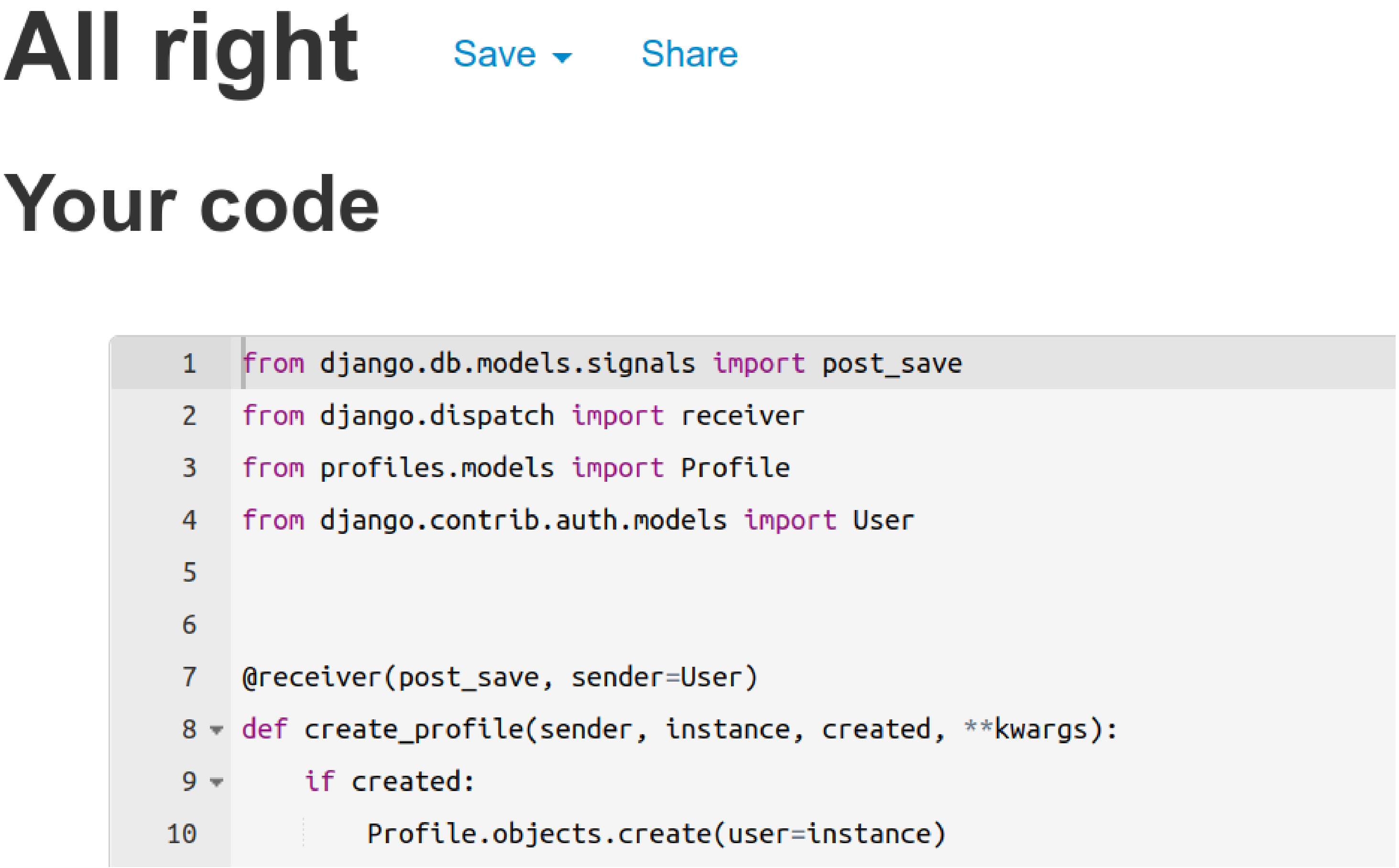
Your code

```python
from django.db import models
from django.contrib.auth.models import User
from cloudinary.models import CloudinaryField
from datetime import date
from django.core.cache import cache
import datetime
from social_network import settings
from chats.models import Message


class Profile(models.Model):
    user = models.OneToOneField(
```

signals.py



- profiles
  - migrations
  - templates
  - tests
    - __init__.py
    - test_forms.py
    - test_models.py
    - test_urls.py
    - test_views.py
  - __init__.py
  - admin.py
  - apps.py
  - forms.py
  - middleware.py
  - models.py
  - signals.py
  - urls.py
  - views.py

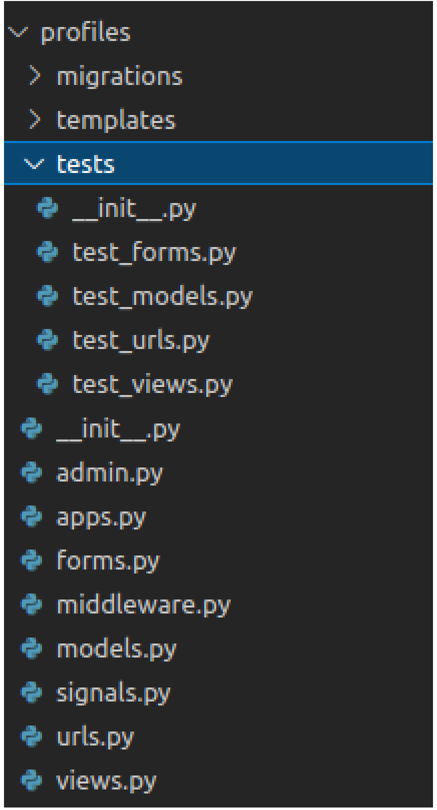


All right

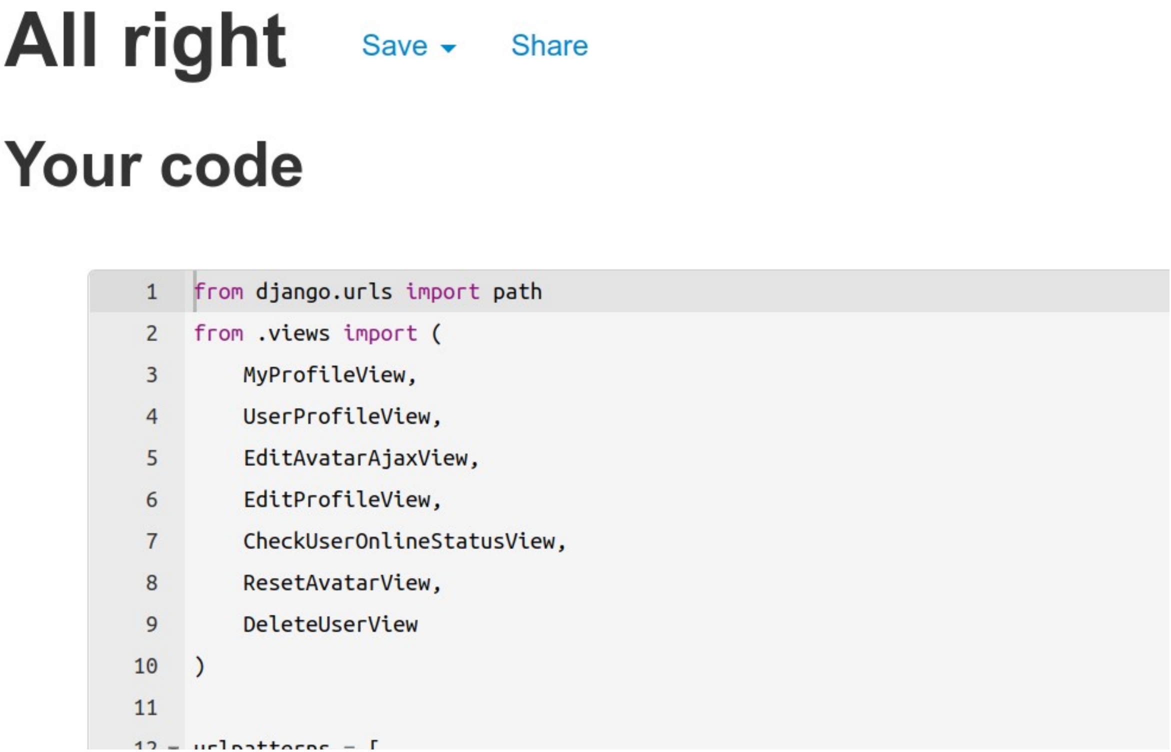
Your code

```
from django.db.models.signals import post_save
from django.dispatch import receiver
from profiles.models import Profile
from django.contrib.auth.models import User


@receiver(post_save, sender=User)
def create_profile(sender, instance, created, **kwargs):
    if created:
        Profile.objects.create(user=instance)
```

urls.py



```
profiles
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  middleware.py
  models.py
  signals.py
  urls.py
  views.py
```

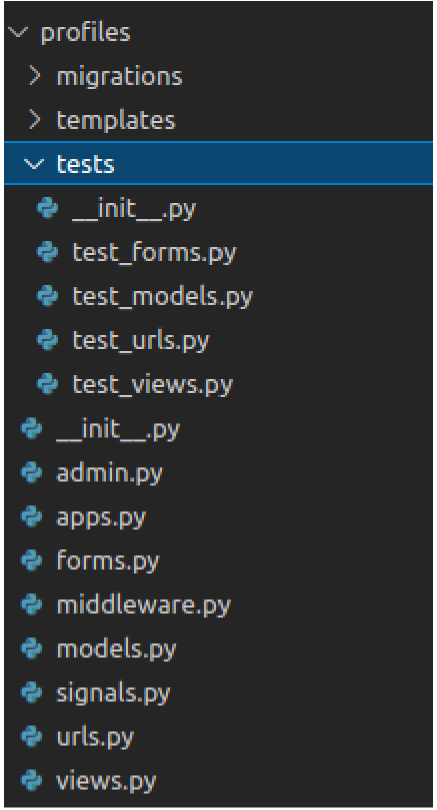


All right

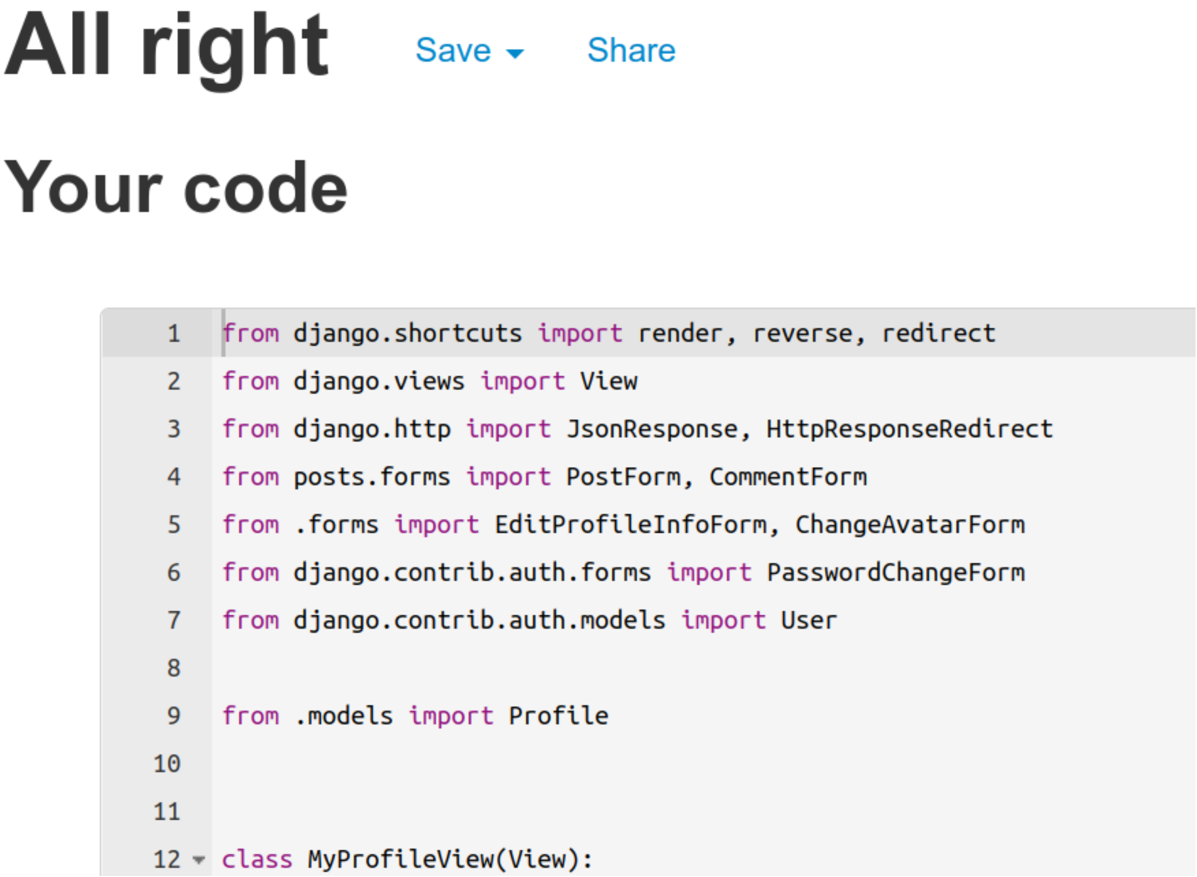
Your code

```python
from django.urls import path
from .views import (
    MyProfileView,
    UserProfileView,
    EditAvatarAjaxView,
    EditProfileView,
    CheckUserOnlineStatusView,
    ResetAvatarView,
    DeleteUserView
)

urlpatterns = [
```

views.py



```
profiles
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  middleware.py
  models.py
  signals.py
  urls.py
  views.py
```

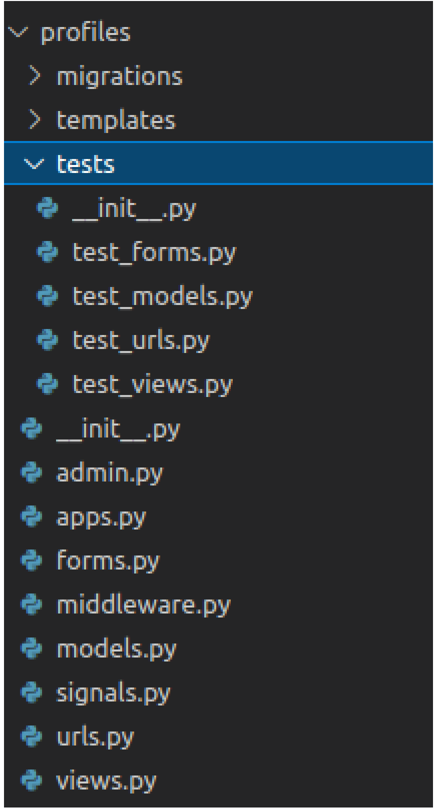


All right

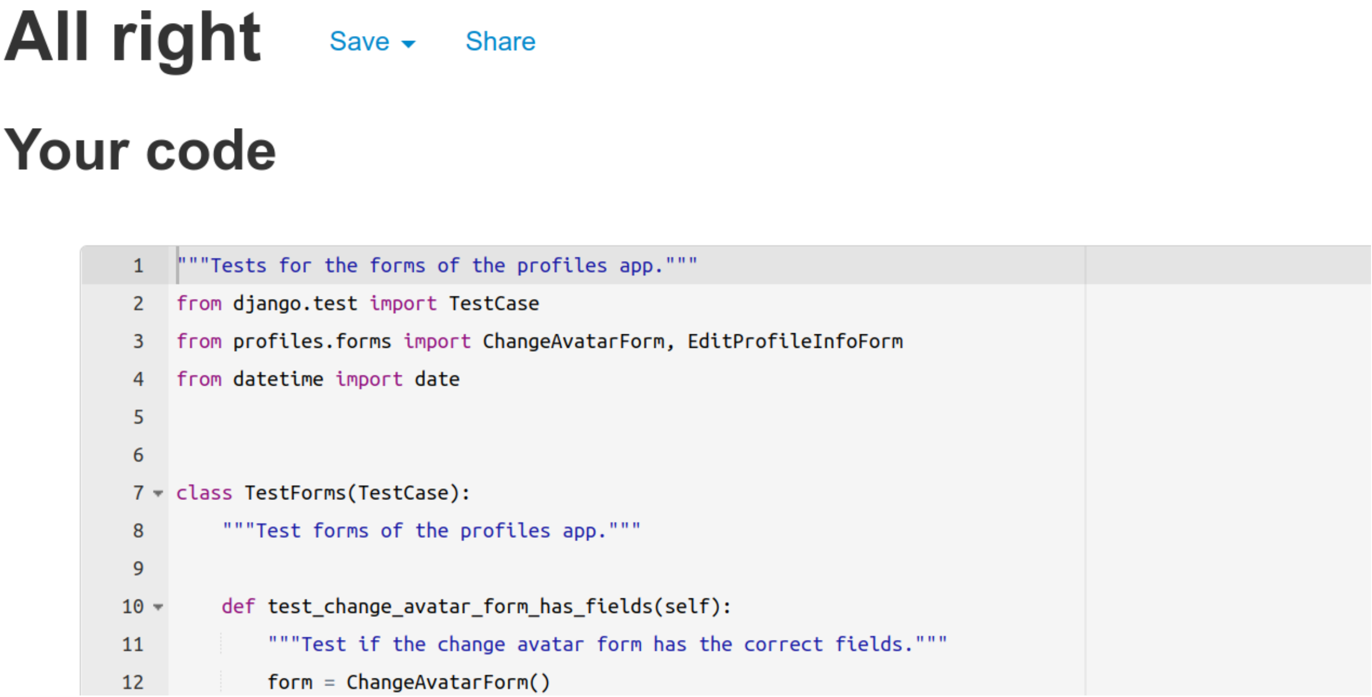
Your code

```python
from django.shortcuts import render, reverse, redirect
from django.views import View
from django.http import JsonResponse, HttpResponseRedirect
from posts.forms import PostForm, CommentForm
from .forms import EditProfileInfoForm, ChangeAvatarForm
from django.contrib.auth.forms import PasswordChangeForm
from django.contrib.auth.models import User

from .models import Profile


class MyProfileView(View):
```

tests/test_forms.py



- profiles
  - migrations
  - templates
  - tests
    - __init__.py
    - test_forms.py
    - test_models.py
    - test_urls.py
    - test_views.py
  - __init__.py
  - admin.py
  - apps.py
  - forms.py
  - middleware.py
  - models.py
  - signals.py
  - urls.py
  - views.py

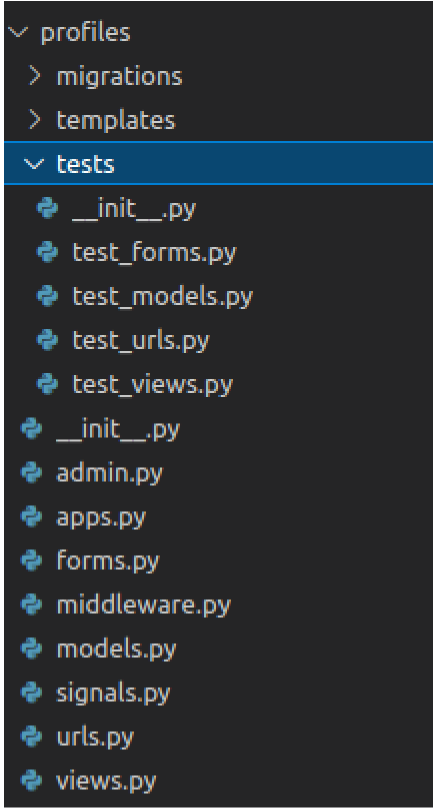


All right

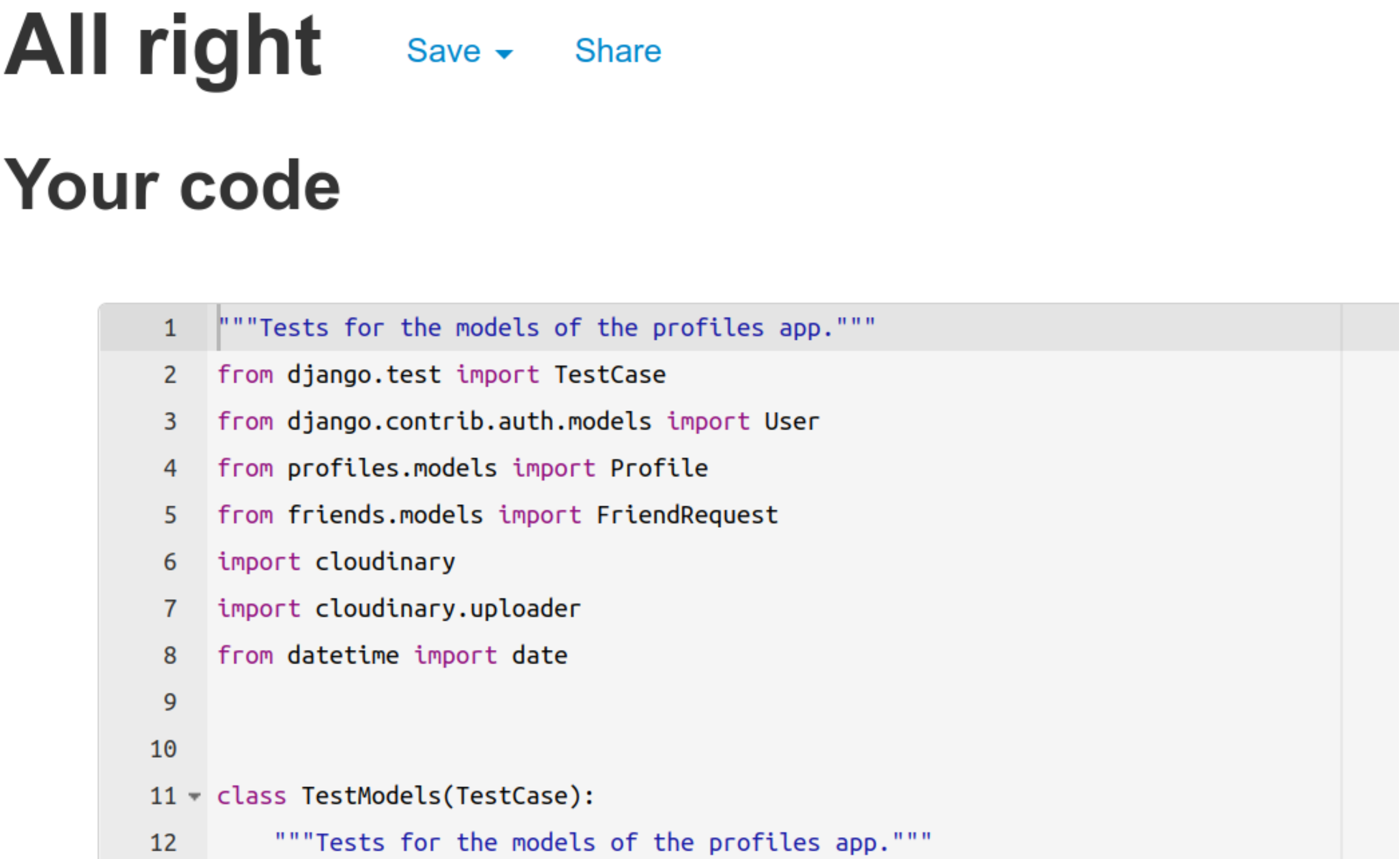
Your code

```python
"""Tests for the forms of the profiles app."""
from django.test import TestCase
from profiles.forms import ChangeAvatarForm, EditProfileInfoForm
from datetime import date


class TestForms(TestCase):
    """Test forms of the profiles app."""

    def test_change_avatar_form_has_fields(self):
        """Test if the change avatar form has the correct fields."""
        form = ChangeAvatarForm()
```

tests/test_models.py



```
profiles
  migrations
  templates
  tests
    __init__.py
    test_forms.py
    test_models.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  forms.py
  middleware.py
  models.py
  signals.py
  urls.py
  views.py
```

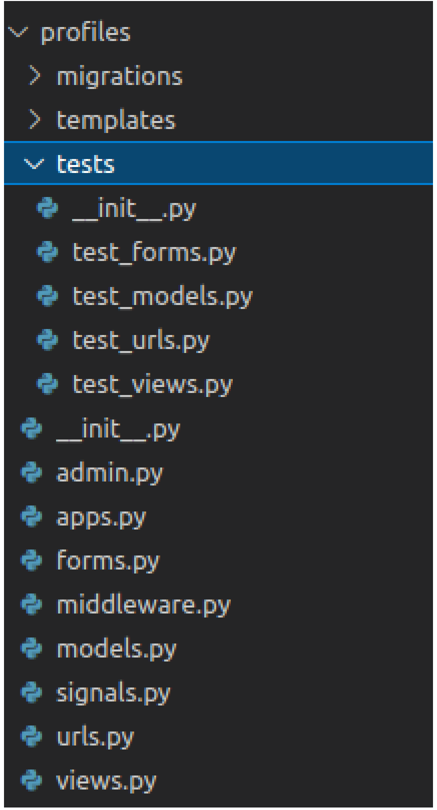


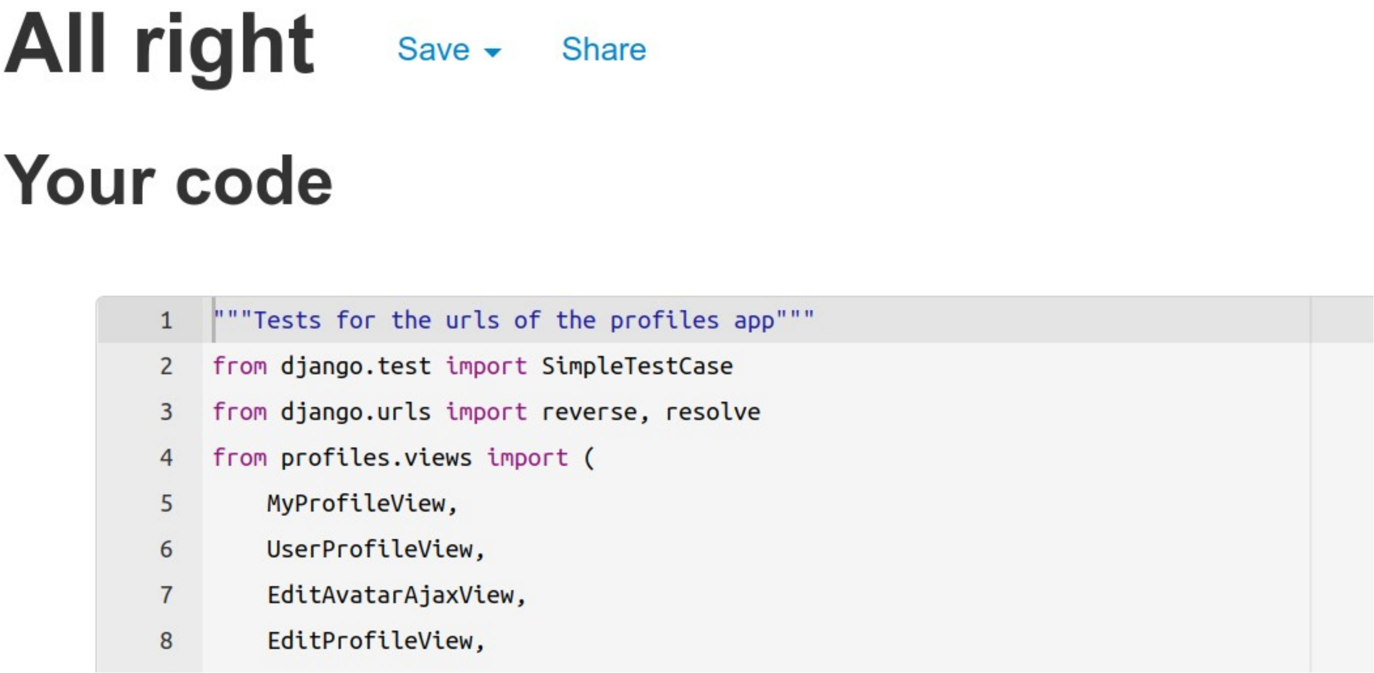
All right

Your code

```python
"""Tests for the models of the profiles app."""
from django.test import TestCase
from django.contrib.auth.models import User
from profiles.models import Profile
from friends.models import FriendRequest
import cloudinary
import cloudinary.uploader
from datetime import date


class TestModels(TestCase):
    """Tests for the models of the profiles app."""
```

tests/test_urls.py



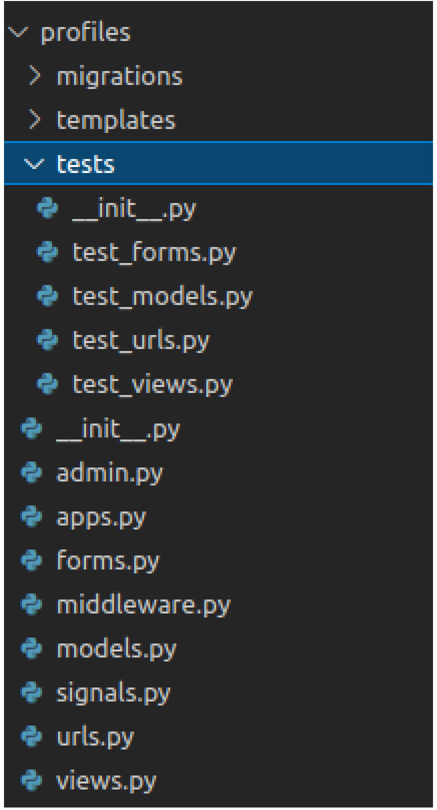


All right

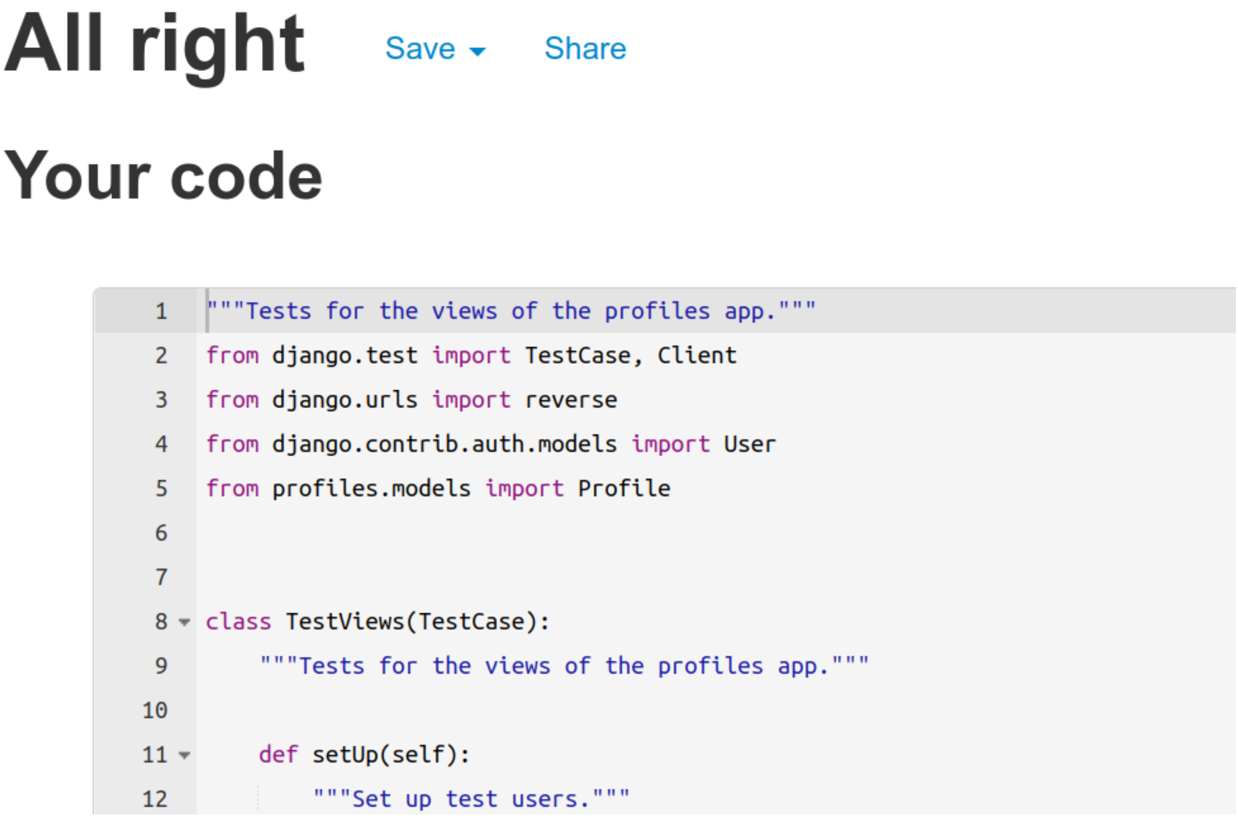
Your code

```python
"""Tests for the urls of the profiles app"""
from django.test import SimpleTestCase
from django.urls import reverse, resolve
from profiles.views import (
    MyProfileView,
    UserProfileView,
    EditAvatarAjaxView,
    EditProfileView,
```

tests/test_views.py



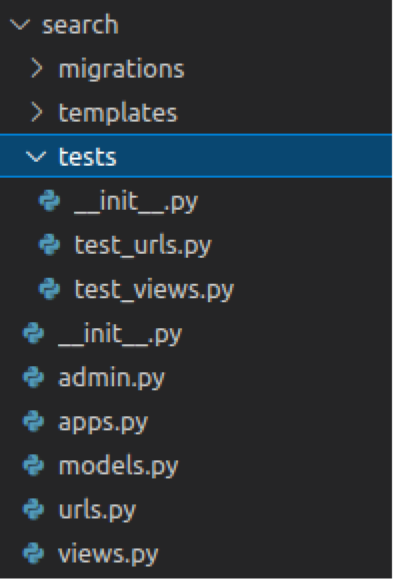
- profiles
  - migrations
  - templates
  - tests
    - __init__.py
    - test_forms.py
    - test_models.py
    - test_urls.py
    - test_views.py
  - __init__.py
  - admin.py
  - apps.py
  - forms.py
  - middleware.py
  - models.py
  - signals.py
  - urls.py
  - views.py



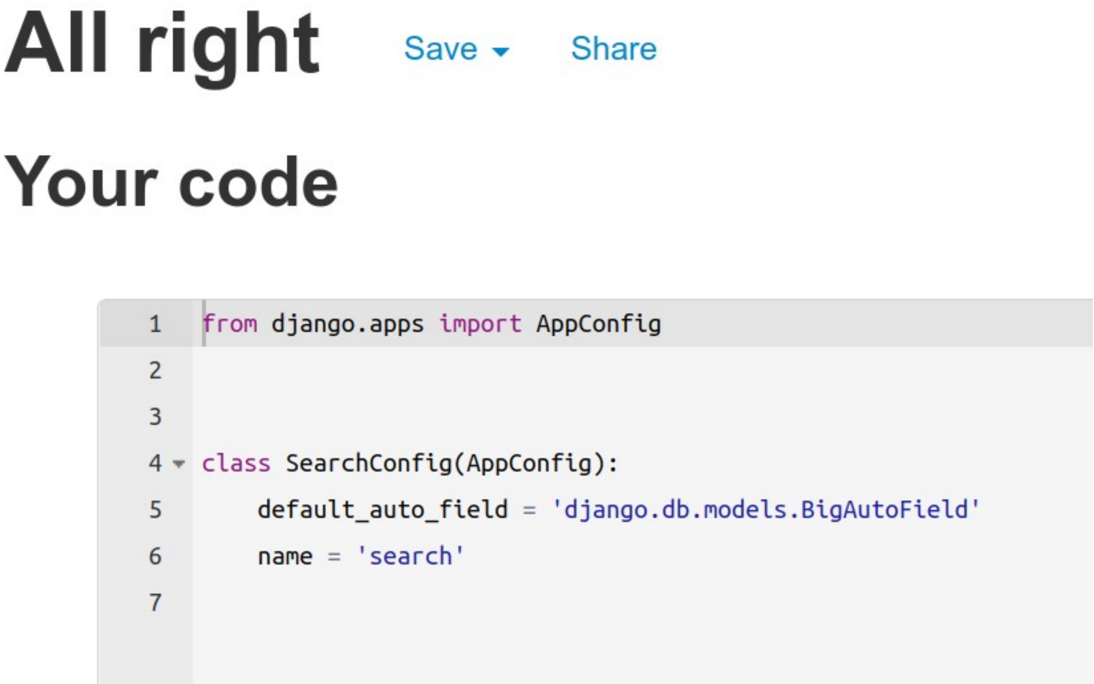
Your code

```python
"""Tests for the views of the profiles app."""
from django.test import TestCase, Client
from django.urls import reverse
from django.contrib.auth.models import User
from profiles.models import Profile


class TestViews(TestCase):
    """Tests for the views of the profiles app."""

    def setUp(self):
        """Set up test users."""
```

apps.py



```
search
  migrations
  templates
  tests
    __init__.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

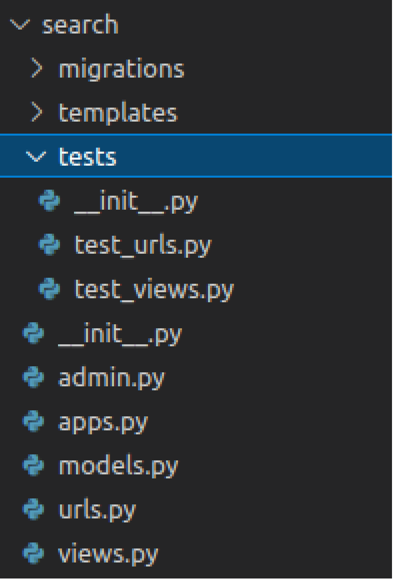


All right

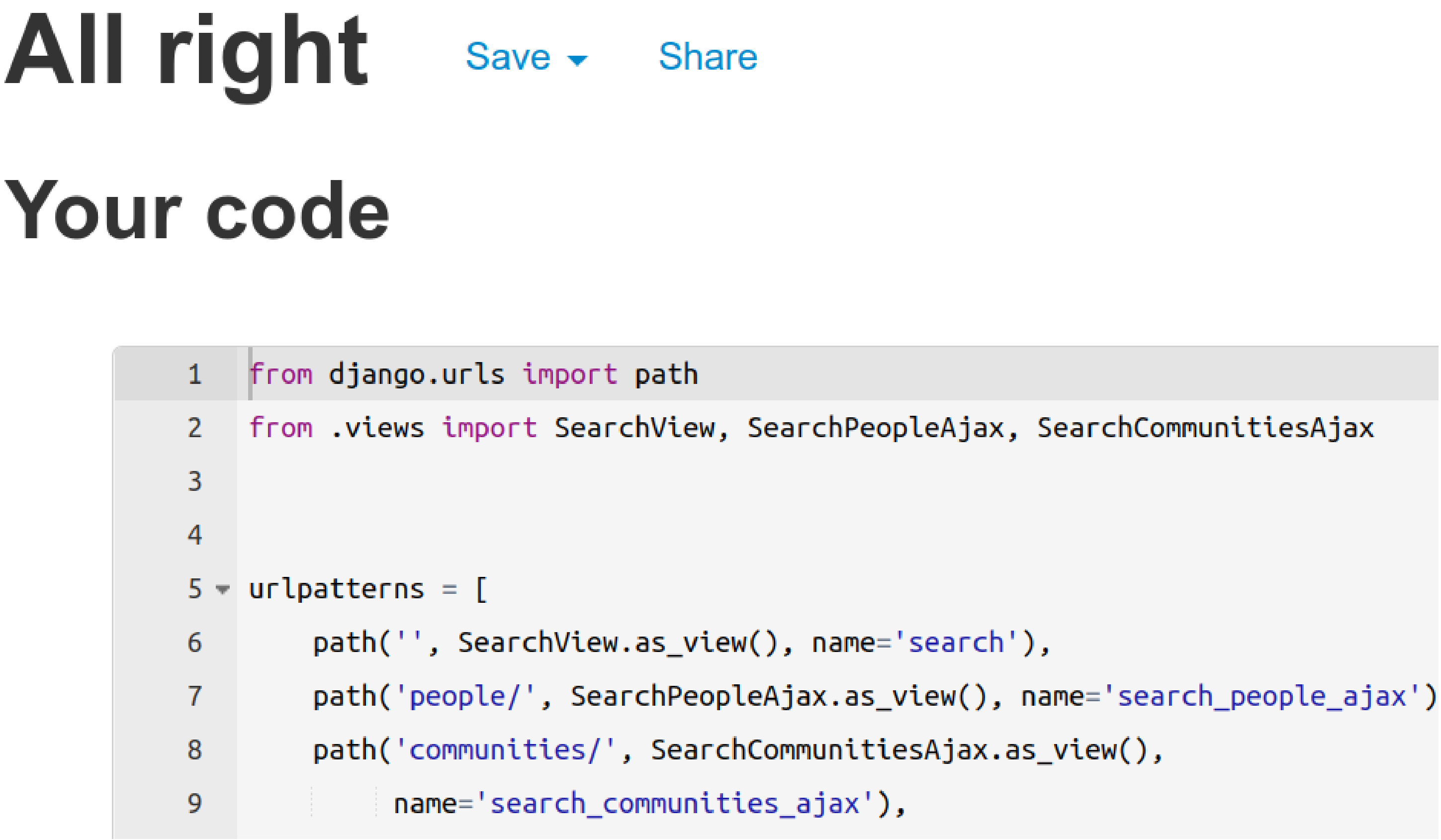
Your code

```
from django.apps import AppConfig


class SearchConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'search'

```

urls.py



```
search
  migrations
  templates
  tests
    __init__.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

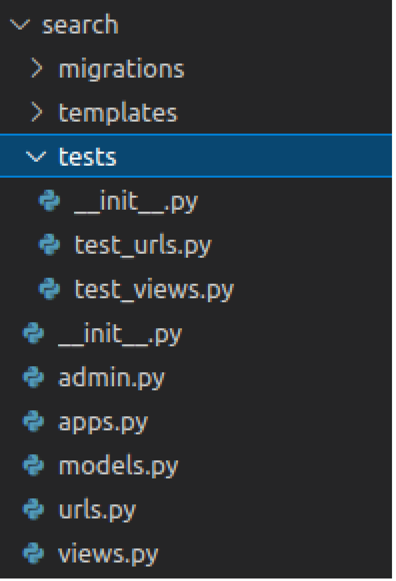


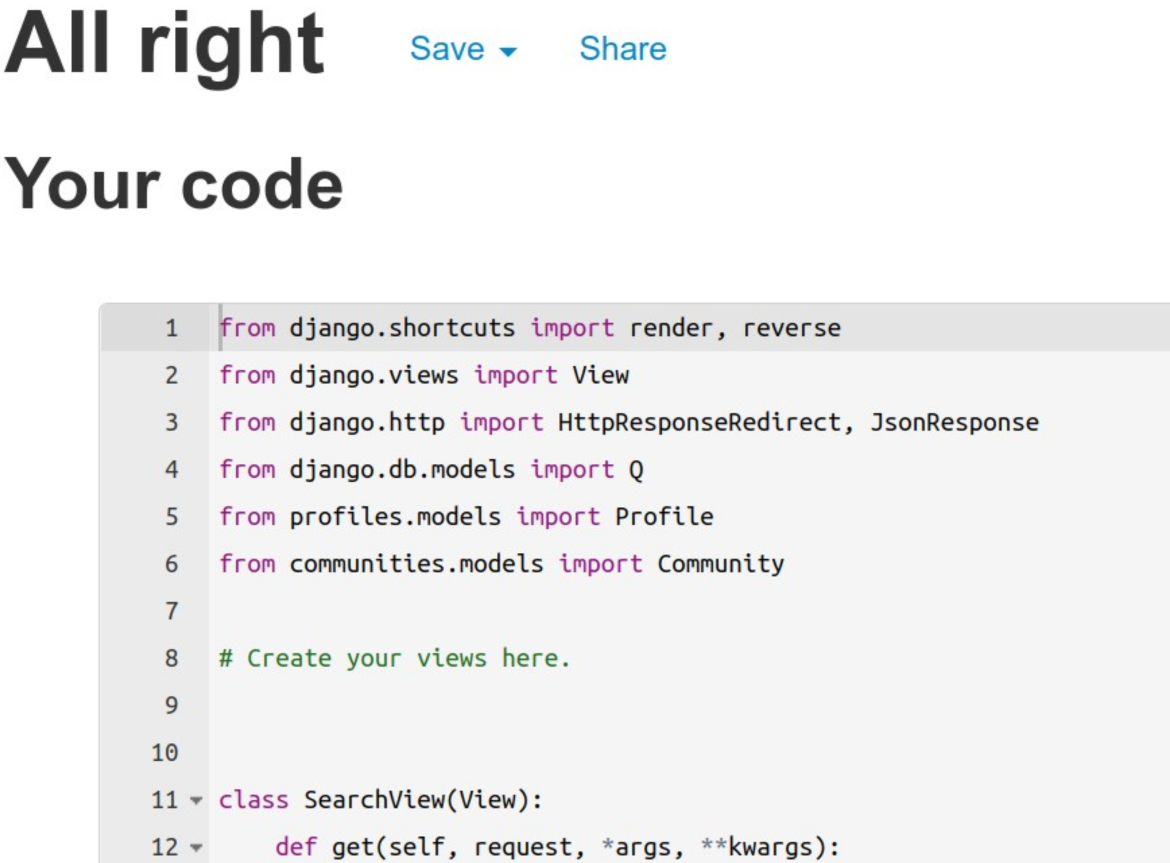
All right

Your code

```
from django.urls import path
from .views import SearchView, SearchPeopleAjax, SearchCommunitiesAjax


urlpatterns = [
    path('', SearchView.as_view(), name='search'),
    path('people/', SearchPeopleAjax.as_view(), name='search_people_ajax')
    path('communities/', SearchCommunitiesAjax.as_view(),
         name='search_communities_ajax'),
```

views.py



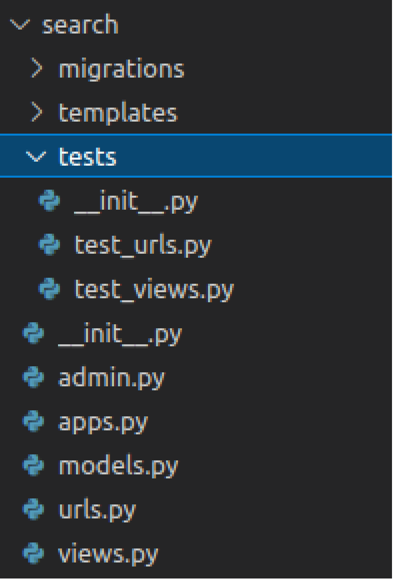


All right

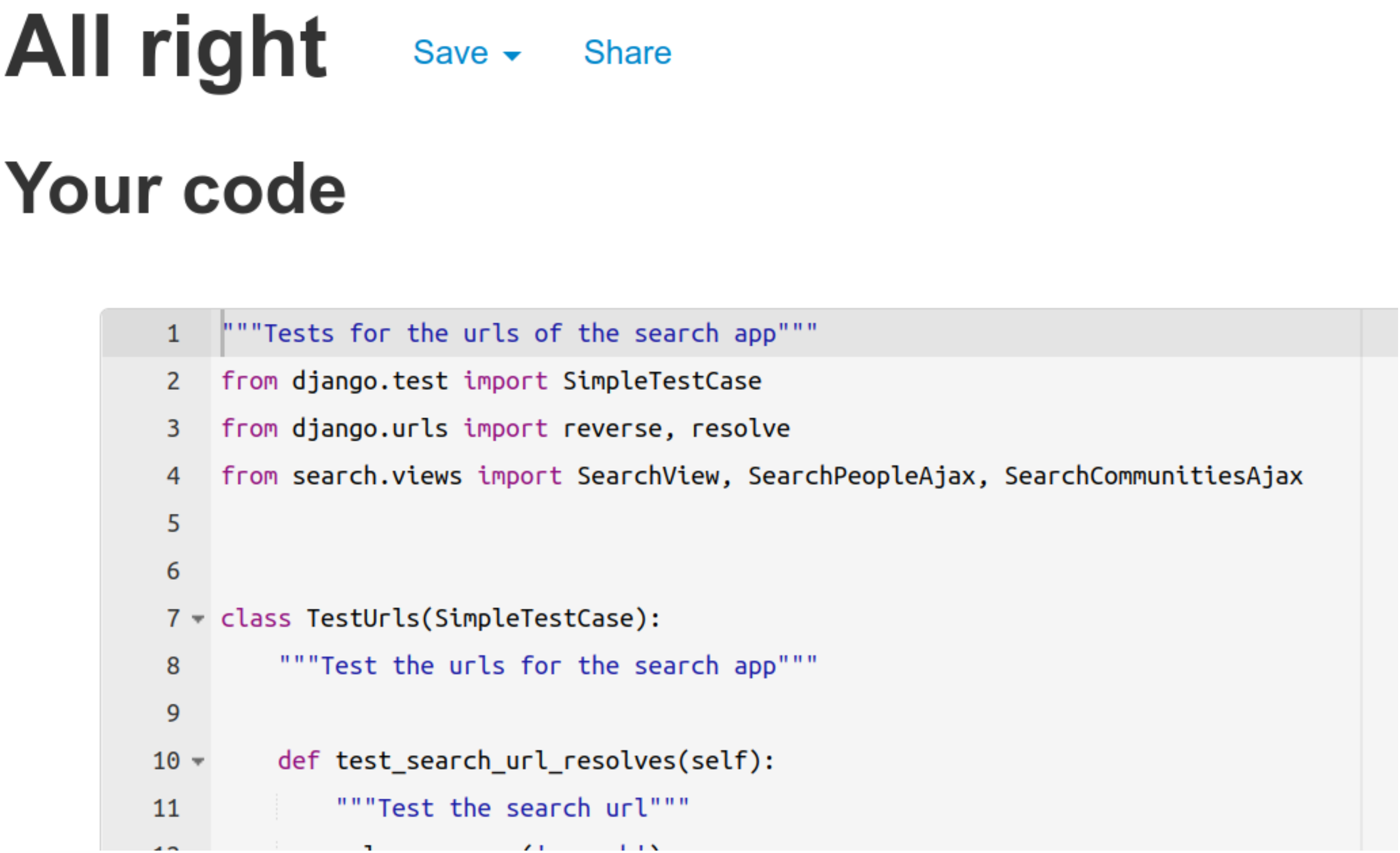
Your code

```python
from django.shortcuts import render, reverse
from django.views import View
from django.http import HttpResponseRedirect, JsonResponse
from django.db.models import Q
from profiles.models import Profile
from communities.models import Community

# Create your views here.


class SearchView(View):
    def get(self, request, *args, **kwargs):
```

tests/test_urls.py



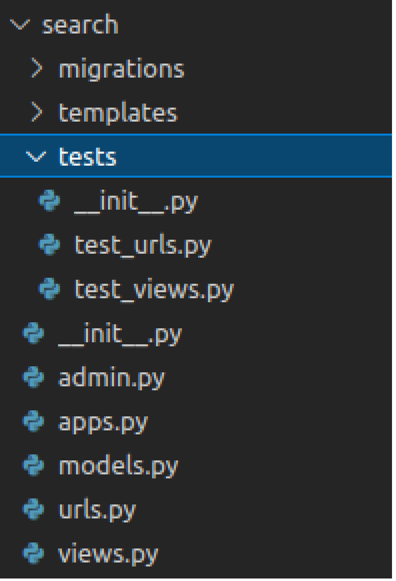
```
search
  migrations
  templates
  tests
    __init__.py
    test_urls.py
    test_views.py
  __init__.py
  admin.py
  apps.py
  models.py
  urls.py
  views.py
```

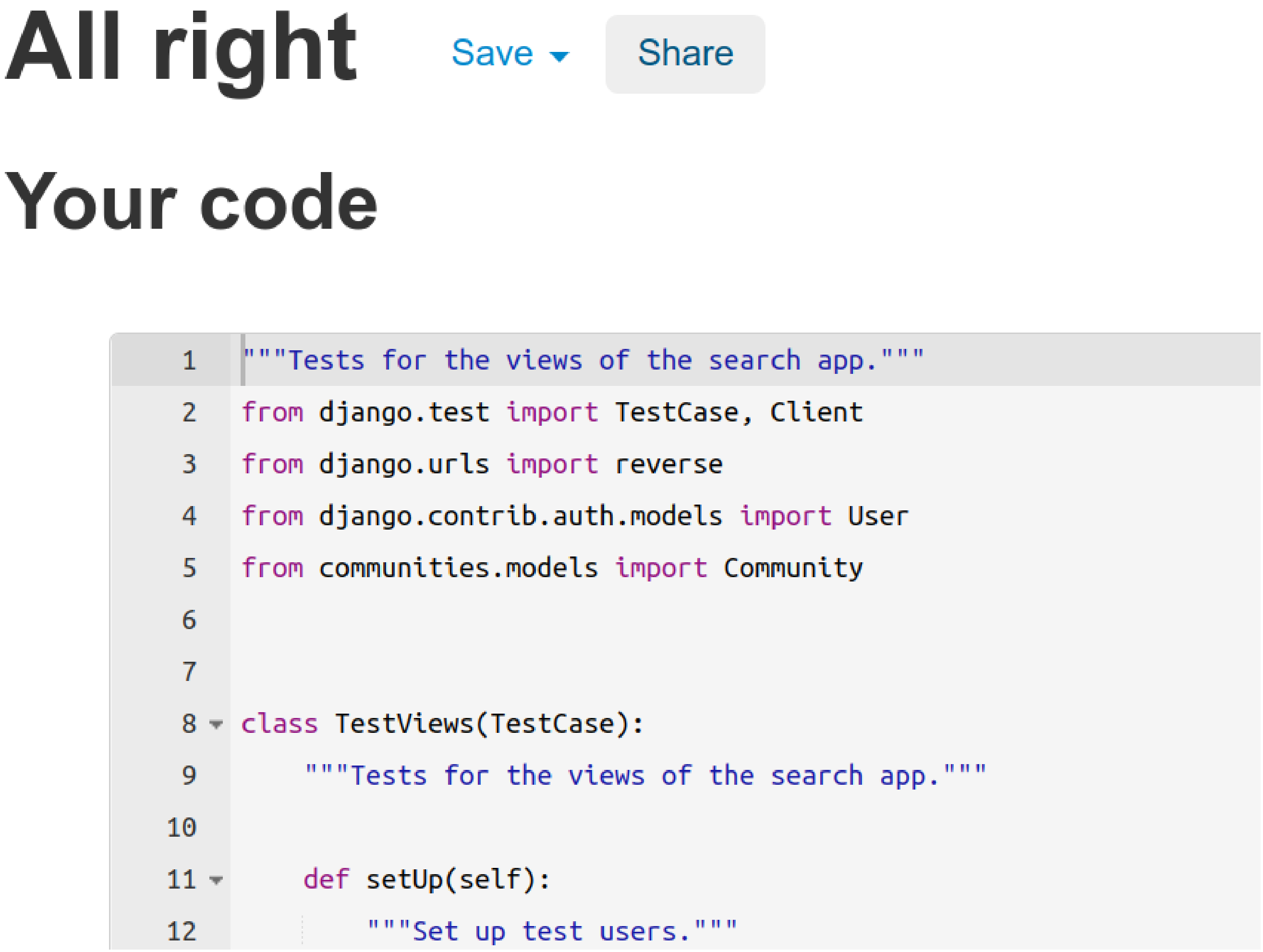


Your code

```python
"""Tests for the urls of the search app"""
from django.test import SimpleTestCase
from django.urls import reverse, resolve
from search.views import SearchView, SearchPeopleAjax, SearchCommunitiesAjax


class TestUrls(SimpleTestCase):
    """Test the urls for the search app"""

    def test_search_url_resolves(self):
        """Test the search url"""
```

tests/test_views.py





All right

Your code

```python
"""Tests for the views of the search app."""
from django.test import TestCase, Client
from django.urls import reverse
from django.contrib.auth.models import User
from communities.models import Community


class TestViews(TestCase):
    """Tests for the views of the search app."""

    def setUp(self):
        """Set up test users."""
```