# users/models.py

from django.contrib.auth.models import AbstractUser

from django.db import models

class CustomUser(AbstractUser):

# Qo'shimcha maydonlar shuningdek first\_name, last\_name, email, etc.

# Agar siz boshqa ma'lumotlarni ham qo'shmoqchi bo'lsangiz, ularni ham qo'shing.

Pass

# users/serializers.py

from rest\_framework import serializers

from django.contrib.auth import get\_user\_model

User = get\_user\_model()

class UserSerializer(serializers.ModelSerializer):

class Meta:

model = User

fields = ('id', 'username', 'email', 'first\_name', 'last\_name')

# Yo'naltirilgan xussiyatlar

extra\_kwargs = {'password': {'write\_only': True}}

def create(self, validated\_data):

user = User.objects.create\_user(\*\*validated\_data)

return user

# users/views.py

from rest\_framework import viewsets

from django.contrib.auth import get\_user\_model

from .serializers import UserSerializer

User = get\_user\_model()

class UserViewSet(viewsets.ModelViewSet):

queryset = User.objects.all()

serializer\_class = UserSerializer

# users/urls.py

from django.urls import path, include

from rest\_framework.routers import DefaultRouter

from .views import UserViewSet

router = DefaultRouter()

router.register(r'users', UserViewSet)

urlpatterns = [

path('', include(router.urls)),

]

# settings.py

INSTALLED\_APPS = [

...

'rest\_framework',

'users', # Agar foydalanuvchi ilova bor bo'lsa

...

]

pip install djangorestframework\_simplejwt

# settings.py

REST\_FRAMEWORK = {

'DEFAULT\_AUTHENTICATION\_CLASSES': (

'rest\_framework\_simplejwt.authentication.JWTAuthentication',

# Agar xavfsizlik muhiti turlari

),

}

# users/views.py

from rest\_framework import generics, permissions

from rest\_framework.response import Response

from rest\_framework\_simplejwt.tokens import RefreshToken

from django.contrib.auth import get\_user\_model

from django.contrib.auth.models import User

from .serializers import UserSerializer

User = get\_user\_model()

class RegisterView(generics.CreateAPIView):

queryset = User.objects.all()

serializer\_class = UserSerializer

permission\_classes = [permissions.AllowAny]

class LoginView(generics.GenericAPIView):

serializer\_class = UserSerializer

def post(self, request, \*args, \*\*kwargs):

serializer = self.get\_serializer(data=request.data)

serializer.is\_valid(raise\_exception=True)

user = serializer.validated\_data

return Response({

"user": UserSerializer(user, context=self.get\_serializer\_context()).data,

"access\_token": str(RefreshToken.for\_user(user).access\_token),

})

class LogoutView(generics.GenericAPIView):

permission\_classes = [permissions.IsAuthenticated]

def post(self, request, \*args, \*\*kwargs):

try:

refresh\_token = request.data["refresh\_token"]

token = RefreshToken(refresh\_token)

token.blacklist()

return Response(status=204)

except Exception as e:

return Response(status=400)

# users/urls.py yoki global `urls.py` faylida

from django.urls import path

from .views import RegisterView, LoginView, LogoutView

urlpatterns = [

path('register/', RegisterView.as\_view(), name='register'),

path('login/', LoginView.as\_view(), name='login'),

path('logout/', LogoutView.as\_view(), name='logout'),

]