Файл: C:\projects\constructor\_vo\_api\users\admin.py

from django.contrib import admin  
  
  
from .models import CustomUser  
class CustomUserAdmin(admin.ModelAdmin):  
 list\_display = ('email', 'first\_name', 'last\_name','middle\_name','role')  
  
  
admin.site.register(CustomUser, CustomUserAdmin)

---

Файл: C:\projects\constructor\_vo\_api\users\apps.py

from django.apps import AppConfig  
  
  
class UsersConfig(AppConfig):  
 default\_auto\_field = 'django.db.models.BigAutoField'  
 name = 'users'

---

Файл: C:\projects\constructor\_vo\_api\users\models.py

from django.contrib.auth.models import AbstractBaseUser, BaseUserManager, PermissionsMixin  
from django.db import models  
  
USER = 'user'  
MODERATOR = 'moderator'  
ADMIN = 'admin'  
  
POSSIBLE\_ROLES = [  
 (USER, USER),  
 (ADMIN, ADMIN)  
]  
  
class CustomUserManager(BaseUserManager):  
 def create\_user(self, email, password=None, \*\*extra\_fields):  
 if not email:  
 raise ValueError('The Email field must be set')  
 extra\_fields.setdefault('first\_name', '')  
 extra\_fields.setdefault('last\_name', '')  
 email = self.normalize\_email(email)  
 user = self.model(email=email, \*\*extra\_fields)  
 user.set\_password(password)  
 user.save(using=self.\_db)  
 return user  
  
 def create\_superuser(self, email, password=None, \*\*extra\_fields):  
 extra\_fields.setdefault('is\_staff', True)  
 extra\_fields.setdefault('is\_superuser', True)  
  
 return self.create\_user(email, password, \*\*extra\_fields)  
  
class CustomUser(AbstractBaseUser, PermissionsMixin):  
 username = models.CharField(max\_length=30, blank=True)  
 email = models.EmailField(unique=True)  
 first\_name = models.CharField(max\_length=30)  
 last\_name = models.CharField(max\_length=30)  
 middle\_name = models.CharField(max\_length=30, blank=True,default="")  
 date\_joined = models.DateTimeField(auto\_now\_add=True)  
 is\_active = models.BooleanField(default=True)  
 is\_staff = models.BooleanField(default=False)  
 role = models.CharField(  
 max\_length=10,  
 choices=POSSIBLE\_ROLES,  
 default=USER,  
 )  
  
 USERNAME\_FIELD = 'email'  
 REQUIRED\_FIELDS = ['first\_name','last\_name']  
  
 objects = CustomUserManager()  
  
 def get\_username(self):  
 return self.email  
  
 def \_\_str\_\_(self):  
 return self.email

---

Файл: C:\projects\constructor\_vo\_api\users\serializers.py

from dj\_rest\_auth.serializers import TokenSerializer  
from dj\_rest\_auth.registration.serializers import RegisterSerializer  
from rest\_framework.authtoken.models import Token  
from dj\_rest\_auth.serializers import UserDetailsSerializer  
from django.conf import settings  
from dj\_rest\_auth.serializers import PasswordResetSerializer as \_PasswordResetSerializer  
  
from rest\_framework import serializers  
from django.contrib.auth import get\_user\_model  
  
  
User = get\_user\_model()  
  
class CustomTokenSerializer(TokenSerializer):  
 token\_type = serializers.CharField(source="get\_token\_type")  
  
 def get\_token\_type(self, obj):  
 return "Bearer"  
  
 class Meta:  
 model = Token  
 fields = ('key', 'token\_type')  
  
class CustomRegisterSerializer(RegisterSerializer):  
 first\_name = serializers.CharField(required=True)  
 last\_name = serializers.CharField(required=True)  
 middle\_name = serializers.CharField()  
  
 # @transaction.atomic  
 def save(self, request):  
 user = super().save(request)  
 user.first\_name = self.data.get('first\_name')  
 user.last\_name = self.data.get('last\_name')  
 user.middle\_name = self.data.get('middle\_name')  
 user.save()  
 return user  
  
  
class CustomUserDetailsSerializer(UserDetailsSerializer):  
 id = serializers.IntegerField(source='pk')  
  
 class Meta(UserDetailsSerializer.Meta):  
 fields = ('id', 'email', 'first\_name', 'last\_name','middle\_name', 'role')  
  
  
class UserShortSerializer(serializers.ModelSerializer):  
 name = serializers.SerializerMethodField()  
  
 class Meta:  
 model = User  
 fields = ('id', 'name')  
  
 def get\_name(self, obj):  
 # Возвращает соединенные имя и фамилию  
 return f"{obj.first\_name} {obj.last\_name}"  
  
  
class PasswordResetSerializer(\_PasswordResetSerializer):  
  
 @property  
 def password\_reset\_form\_class(self):  
 return CustomResetPasswordForm  
  
 def get\_email\_options(self):  
  
 return {  
 'email\_template\_name': 'password/password\_reset'  
 }  
  
from allauth.account.adapter import get\_adapter  
from allauth.account.forms import ResetPasswordForm  
from allauth.account.utils import user\_pk\_to\_url\_str  
from django.conf import settings  
  
  
class CustomResetPasswordForm(ResetPasswordForm):  
 def save(self, request, \*\*kwargs):  
 email = self.cleaned\_data['email']  
 token\_generator = kwargs.get('token\_generator')  
 template = kwargs.get("email\_template\_name")  
 extra = kwargs.get("extra\_email\_context", {})  
 # client\_app = extra["client\_app"]  
 for user in self.users:  
 uid = user\_pk\_to\_url\_str(user)  
 token = token\_generator.make\_token(user)  
 reset\_url = f"{settings.FRONT\_END}/reset\_password/{uid}/{token}"  
 context = {"user": user, "request": request, "email": email, "reset\_url": reset\_url}  
 context.update(extra)  
 get\_adapter(request).send\_mail(template, email, context)  
 return email

---

Файл: C:\projects\constructor\_vo\_api\users\tests.py

from django.test import TestCase  
  
# Create your tests here.

---

Файл: C:\projects\constructor\_vo\_api\users\views.py

from django.shortcuts import render  
from rest\_framework import generics  
from django.contrib.auth import get\_user\_model  
from .serializers import UserShortSerializer  
User = get\_user\_model()  
  
class UserListView(generics.ListAPIView):  
 queryset = User.objects.all()  
 serializer\_class = UserShortSerializer

---

Файл: C:\projects\constructor\_vo\_api\users\\_\_init\_\_.py

---

Файл: C:\projects\constructor\_vo\_api\users\migrations\0001\_initial.py

# Generated by Django 4.1.5 on 2023-11-01 14:30  
  
from django.db import migrations, models  
  
  
class Migration(migrations.Migration):  
  
 initial = True  
  
 dependencies = [  
 ('auth', '0012\_alter\_user\_first\_name\_max\_length'),  
 ]  
  
 operations = [  
 migrations.CreateModel(  
 name='CustomUser',  
 fields=[  
 ('id', models.BigAutoField(auto\_created=True, primary\_key=True, serialize=False, verbose\_name='ID')),  
 ('password', models.CharField(max\_length=128, verbose\_name='password')),  
 ('last\_login', models.DateTimeField(blank=True, null=True, verbose\_name='last login')),  
 ('is\_superuser', models.BooleanField(default=False, help\_text='Designates that this user has all permissions without explicitly assigning them.', verbose\_name='superuser status')),  
 ('email', models.EmailField(max\_length=254, unique=True)),  
 ('first\_name', models.CharField(blank=True, max\_length=30)),  
 ('last\_name', models.CharField(blank=True, max\_length=30)),  
 ('date\_joined', models.DateTimeField(auto\_now\_add=True)),  
 ('is\_active', models.BooleanField(default=True)),  
 ('is\_staff', models.BooleanField(default=False)),  
 ('groups', models.ManyToManyField(blank=True, help\_text='The groups this user belongs to. A user will get all permissions granted to each of their groups.', related\_name='user\_set', related\_query\_name='user', to='auth.group', verbose\_name='groups')),  
 ('user\_permissions', models.ManyToManyField(blank=True, help\_text='Specific permissions for this user.', related\_name='user\_set', related\_query\_name='user', to='auth.permission', verbose\_name='user permissions')),  
 ],  
 options={  
 'abstract': False,  
 },  
 ),  
 ]

---

Файл: C:\projects\constructor\_vo\_api\users\migrations\0002\_customuser\_username.py

# Generated by Django 4.1.5 on 2023-11-01 15:18  
  
from django.db import migrations, models  
  
  
class Migration(migrations.Migration):  
  
 dependencies = [  
 ('users', '0001\_initial'),  
 ]  
  
 operations = [  
 migrations.AddField(  
 model\_name='customuser',  
 name='username',  
 field=models.CharField(blank=True, max\_length=30),  
 ),  
 ]

---

Файл: C:\projects\constructor\_vo\_api\users\migrations\0003\_alter\_customuser\_first\_name\_and\_more.py

# Generated by Django 4.1.5 on 2023-11-01 15:24  
  
from django.db import migrations, models  
  
  
class Migration(migrations.Migration):  
  
 dependencies = [  
 ('users', '0002\_customuser\_username'),  
 ]  
  
 operations = [  
 migrations.AlterField(  
 model\_name='customuser',  
 name='first\_name',  
 field=models.CharField(max\_length=30),  
 ),  
 migrations.AlterField(  
 model\_name='customuser',  
 name='last\_name',  
 field=models.CharField(max\_length=30),  
 ),  
 ]

---

Файл: C:\projects\constructor\_vo\_api\users\migrations\0004\_customuser\_role.py

# Generated by Django 4.1.5 on 2023-11-02 10:32  
  
from django.db import migrations, models  
  
  
class Migration(migrations.Migration):  
  
 dependencies = [  
 ('users', '0003\_alter\_customuser\_first\_name\_and\_more'),  
 ]  
  
 operations = [  
 migrations.AddField(  
 model\_name='customuser',  
 name='role',  
 field=models.CharField(choices=[('user', 'user'), ('admin', 'admin')], default='user', max\_length=10),  
 ),  
 ]

---

Файл: C:\projects\constructor\_vo\_api\users\migrations\0005\_customuser\_middle\_name.py

# Generated by Django 4.1.5 on 2023-12-18 10:42  
  
from django.db import migrations, models  
  
  
class Migration(migrations.Migration):  
  
 dependencies = [  
 ('users', '0004\_customuser\_role'),  
 ]  
  
 operations = [  
 migrations.AddField(  
 model\_name='customuser',  
 name='middle\_name',  
 field=models.CharField(max\_length=30, null=True),  
 ),  
 ]

---

Файл: C:\projects\constructor\_vo\_api\users\migrations\0006\_alter\_customuser\_middle\_name.py

# Generated by Django 4.1.5 on 2023-12-18 10:45  
  
from django.db import migrations, models  
  
  
class Migration(migrations.Migration):  
  
 dependencies = [  
 ('users', '0005\_customuser\_middle\_name'),  
 ]  
  
 operations = [  
 migrations.AlterField(  
 model\_name='customuser',  
 name='middle\_name',  
 field=models.CharField(blank=True, default='', max\_length=30),  
 ),  
 ]

---

Файл: C:\projects\constructor\_vo\_api\users\migrations\\_\_init\_\_.py

---