Create djangoapi2 project

Then create Dockerfile and docker-compose.yml file

Change the database to postgres in settings.py

create accounts app

Create custom user mode

Register the user in settings.py

$python manage.py makemigrations accounts

$python manage.py migrate

$python manage.py createsuperuser

**ABSTRACT USER/AUTHENTICATION**

AbstractUser part of django.contrib.auth.models provides built in fields for authentication system

username

first\_name

last\_name

email

password

is\_staff (boolean field indicating whether user can access admin site)

is\_active (boolean, indicating whether users account is active)

is\_superuser (has all permissions)

last\_login (time stamp)

date\_joined

Methods are:

get\_full\_name()

get\_short\_name

has\_perm

has\_perms

get\_user\_permissions

get\_group\_permissions

get\_all\_permissions

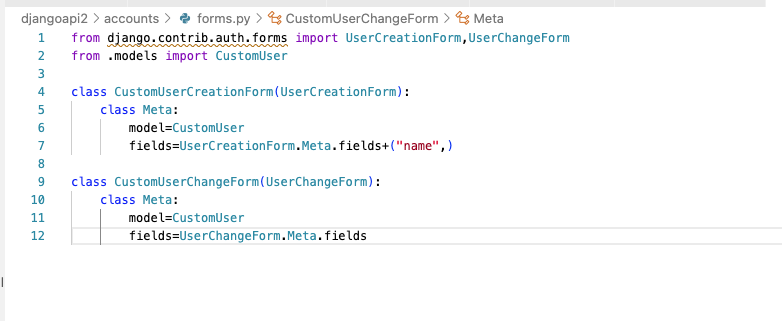
We can add additional fields to it.

Create a CustomUser model.

This model will be the basis for authentication.So register it in settings.py

AUTH\_USER\_MODEL= “accounts.CustomUser”

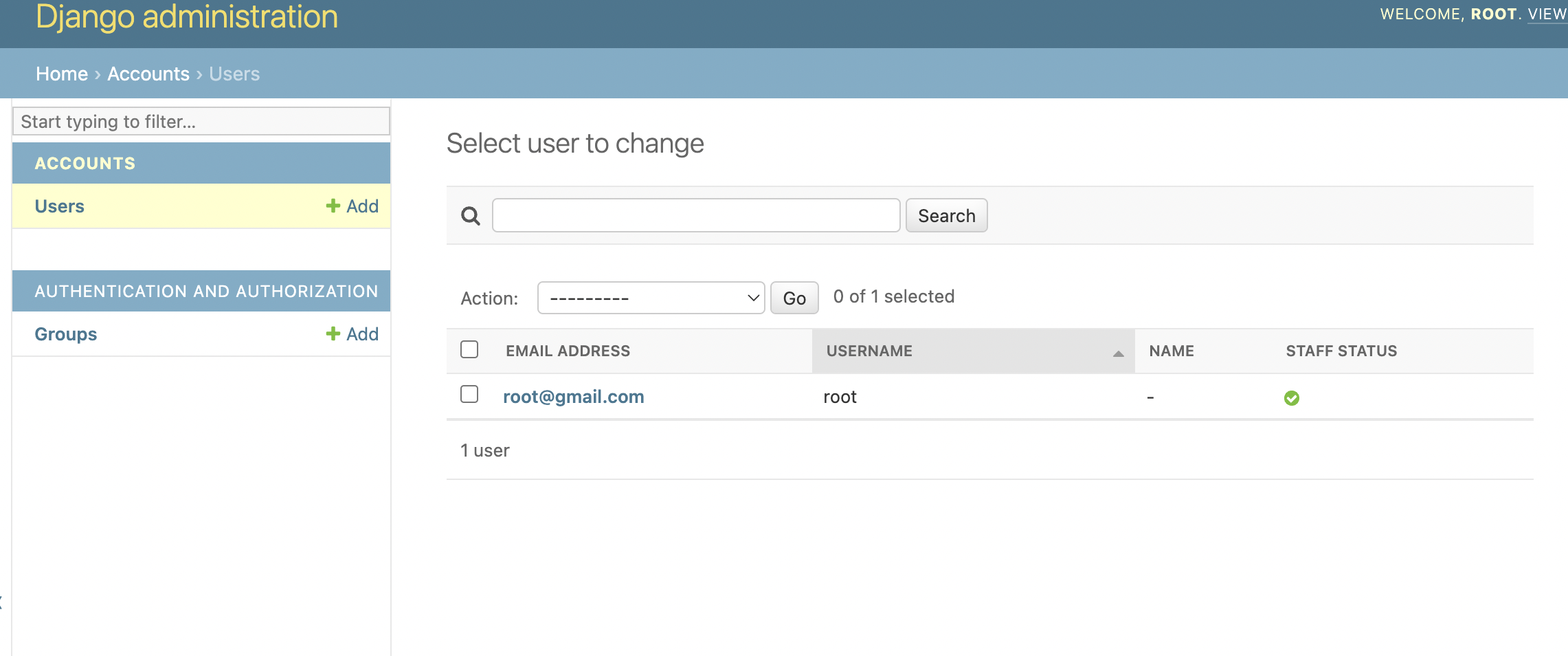
We need some users. Thats why we created above model. But these users must be seen on admin panel. They should be/can be managed over the admin panel. Hence, we create forms for creating and changing users. When the user model is added to our admin panel, we register these forms along.



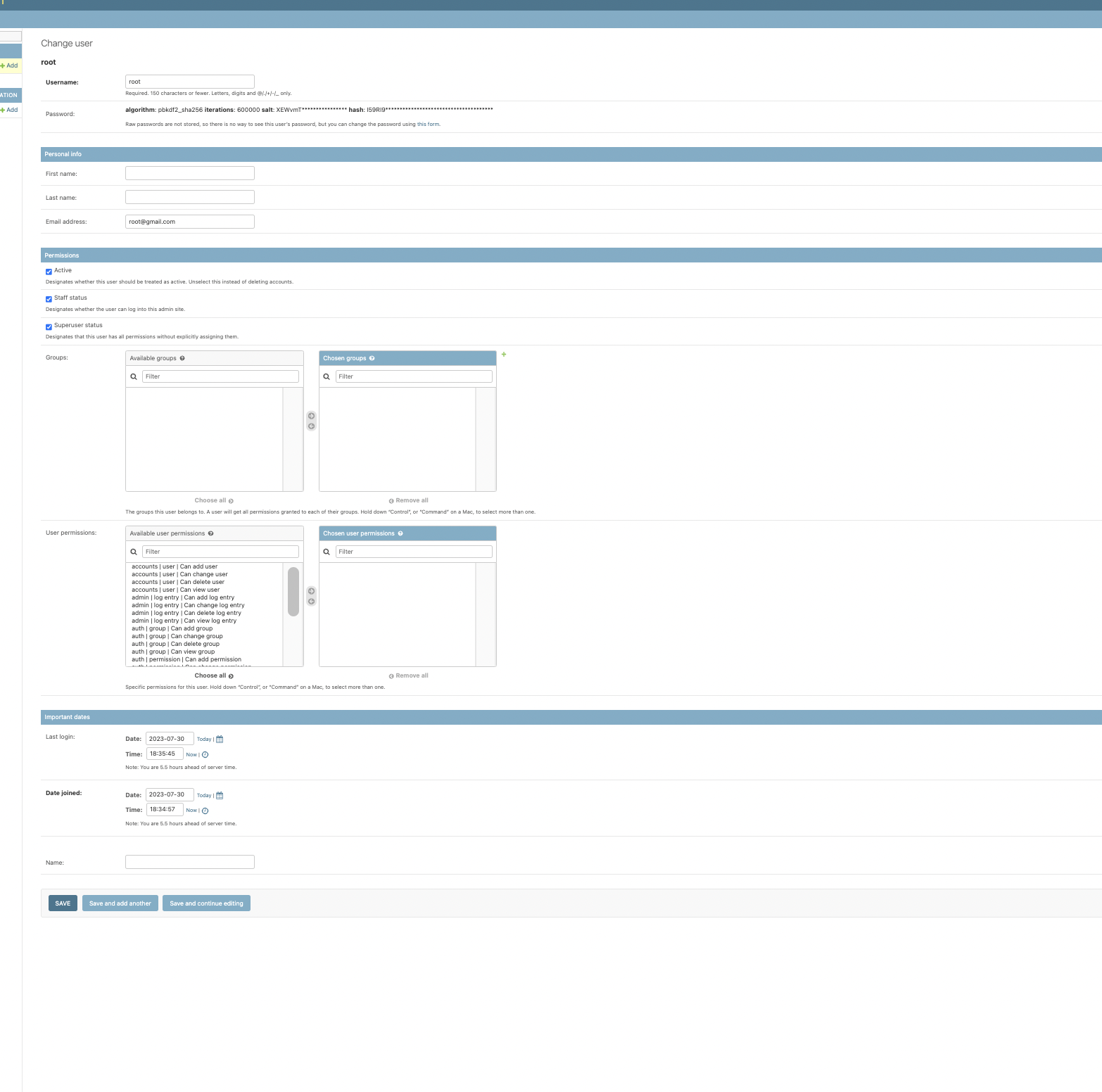
We create the above two forms. Now we will register them so that they can be used in admin interface.



Now go to admin panel. You can see Users table. Click on it. You can see the root user.



When you click on root user you can see the “name” too.



Also, when you want to add user, you can see the Name field too.

