Django-allauth
Django 2.2

добавление собственных полей
с AbstractUser моделью

vk.com/python_django_ru

Устанавливаем Django-allauth: **pip install django-allauth**

settings.py/INSTALLED_APPS

```
...
 'django.contrib.sites',

'allauth',
'allauth.account',
'allauth.socialaccount',
...
```

settings.py

```
...
SITE_ID = 1
```

settings.py

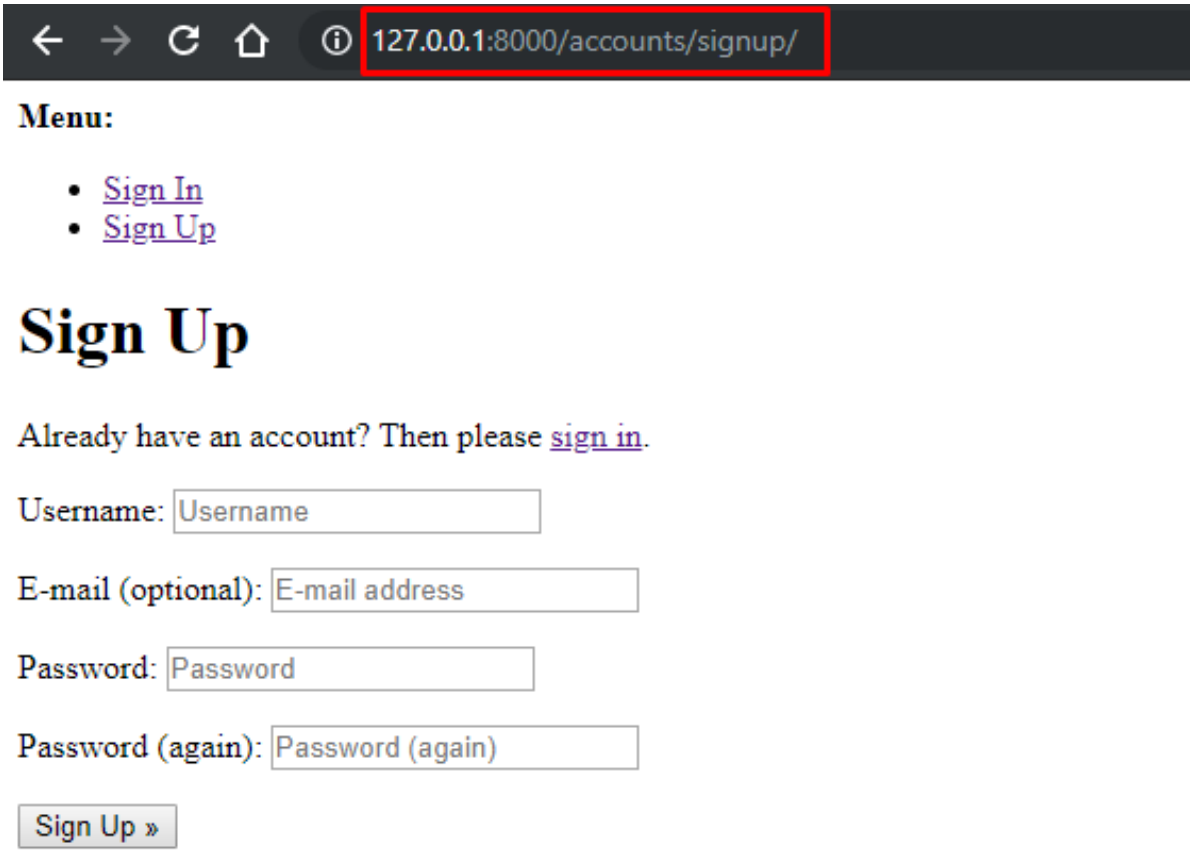
```
...
AUTHENTICATION_BACKENDS = (
    ...
    'django.contrib.auth.backends.ModelBackend',
    'allauth.account.auth_backends.AuthenticationBackend',
    ...
)
```

файл проекта urls.py

```
urlpatterns = [
    ...
    path('accounts/', include('allauth.urls')),
    ...
]
```

Выполнить миграцию: python manage.py migrate

Запускаем сервер переходим к http://127.0.0.1:8000/accounts/signup/



Останвливаем сервер и создаем новое приложение **accounts**:

python manage.py startapp accounts

Создаем модель:

accounts/models.py

```
from django.db import models
from django.contrib.auth.models import AbstractUser

class CustomUser(AbstractUser):
    phone = models.CharField(max_length=12)
```

Создаем файл **forms.py** в папке приложения **accounts**

accounts/forms.py

```
from allauth.account.forms import SignupForm
from django import forms
from .models import *

class SimpleSignupForm(SignupForm):
    phone = forms.CharField(max_length=12, label='Телефон')

    def save(self, request):
        user = super(SimpleSignupForm, self).save(request)
        user.phone = self.cleaned_data['phone']
        user.save()
        return user
```

В файле settings.py добавляем параметр ACCOUNT_FORMS с указанем созданной формы и параметр AUTH_USER_MODEL с указанием нашей кастомной модели.

settings.py

```
...

ACCOUNT_FORMS = {'signup': 'accounts.forms.SimpleSignupForm'}

AUTH_USER_MODEL = 'accounts.CustomUser'
```

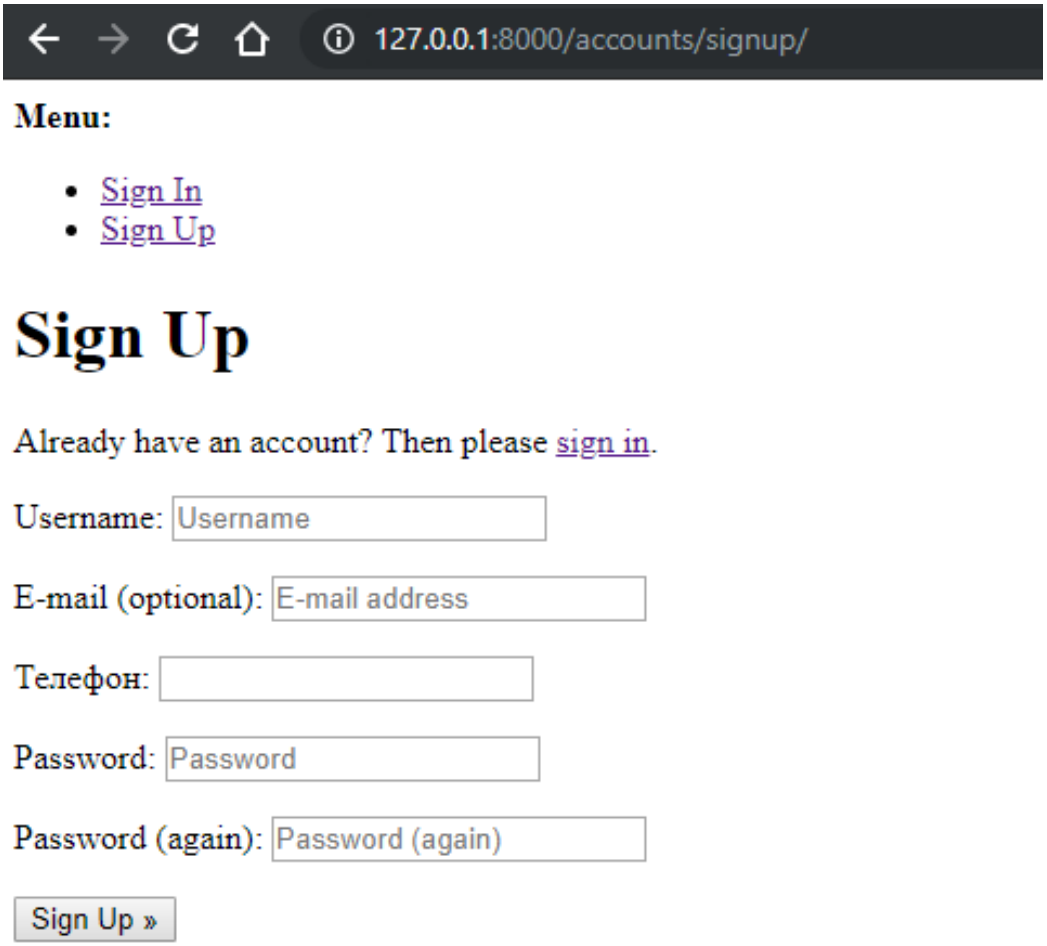
Добавляем приложение **accounts** в параметр **INSTALLED_APPS** файла **settings.py**

settings.py/INSTALLED_APPS

```
...
 'django.contrib.sites',

'accounts',

'allauth',
'allauth.account',
'allauth.socialaccount',
...
```

Теперь необходимо удалить созданную ранее базу данных и создать ее заново выполняем: **python manage.py makemigrations accounts,**
а затем **python manage.py migrate**

Запускаем сервер переходим к http://127.0.0.1:8000/accounts/signup/



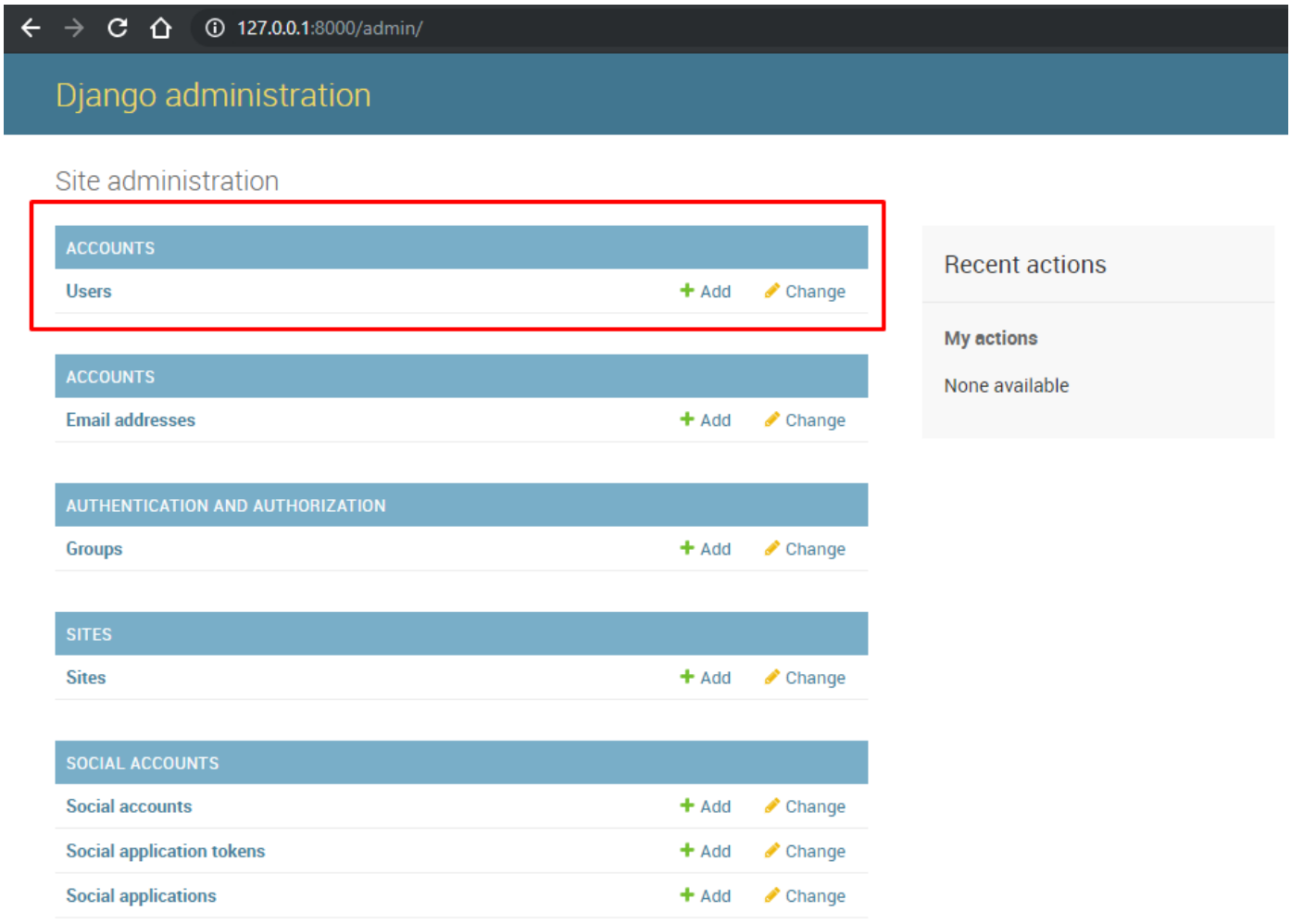
В файле admin.py приложения accounts добавим:

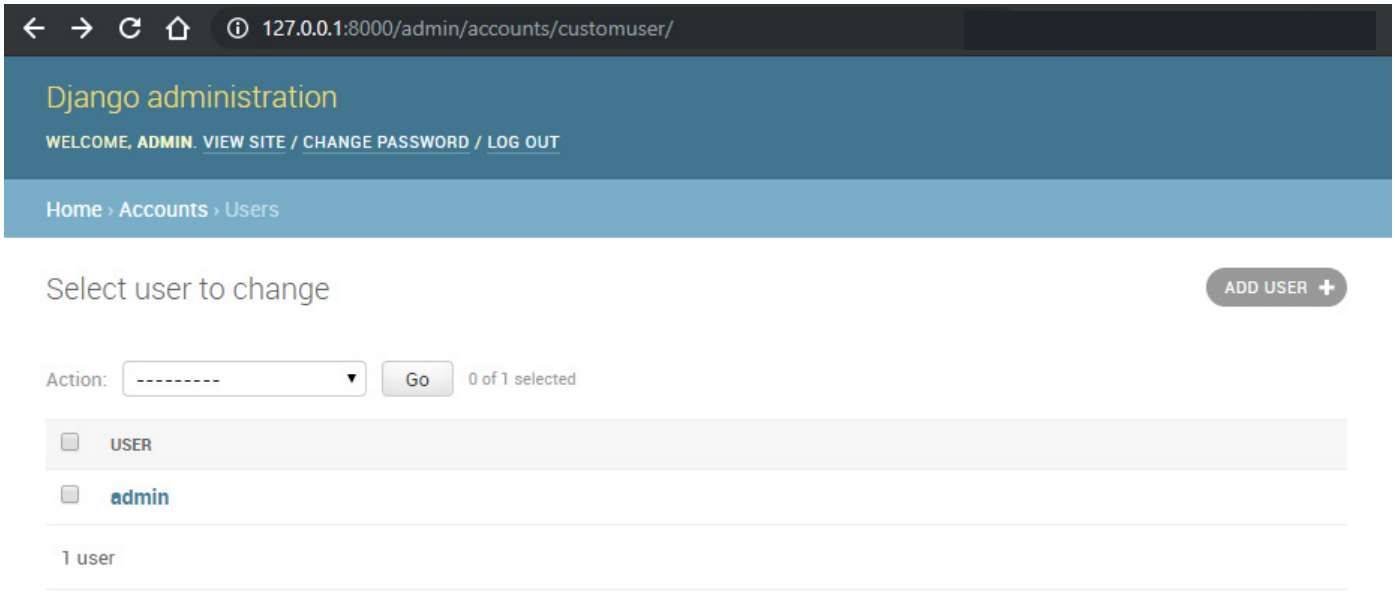
accounts/admin.py

```
from django.contrib import admin
from .models import *



admin.site.register(CustomUser)
```

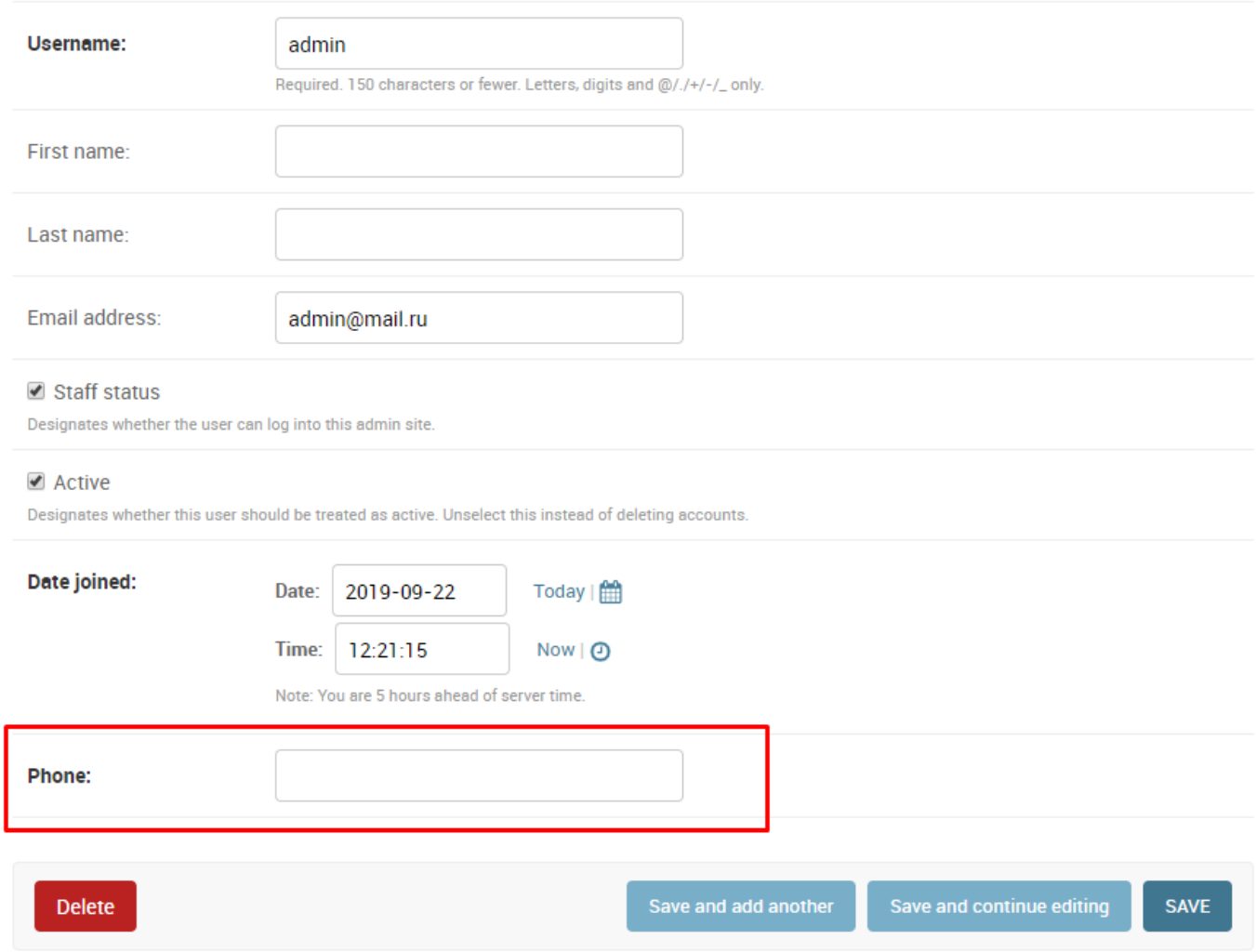
Создадим суперюзера **python manage.py createsuperuser**

Запустим сервер и перейдем в админку http://127.0.0.1:8000/admin/


127.0.0.1:8000/admin/
Django administration
Site administration
ACCOUNTS
Users
+ Add
Change
ACCOUNTS
Email addresses
+ Add
Change
AUTHENTICATION AND AUTHORIZATION
Groups
+ Add
Change
SITES
Sites
+ Add
Change
SOCIAL ACCOUNTS
Social accounts
+ Add
Change
Social application tokens
+ Add
Change
Social applications
+ Add
Change
Recent actions
My actions
None available


127.0.0.1:8000/admin/accounts/customuser/
Django administration
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Accounts › Users
Select user to change
ADD USER +
Action:

Go
0 of 1 selected
USER
admin
1 user



Сделаем регистрацию только по адресу эл. почты и обязательную ее проверку.
В нижней части файла settings.py добавим:

settings.py

```
...

ACCOUNT_USERNAME_REQUIRED = False
ACCOUNT_AUTHENTICATION_METHOD = 'email'
ACCOUNT_EMAIL_REQUIRED = True
ACCOUNT_UNIQUE_EMAIL = True
ACCOUNT_EMAIL_VERIFICATION = 'mandatory'

EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'
```

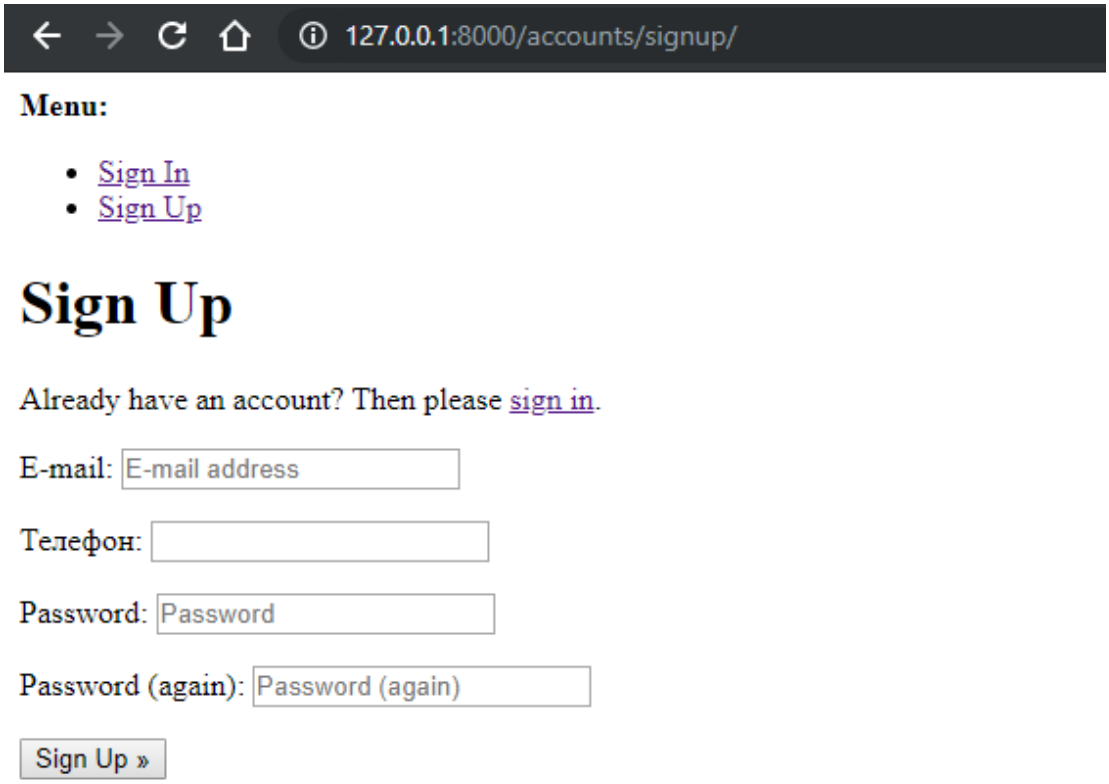
Если хотите использовать свой почтовый ящик для отправки приложением писем, добавьте следующие параметры в нижней части файла **settings.py** на примере почты **google.**

В настройках безопасности своего google аккаунта включите *Ненадежные приложения, у которых есть доступ к аккаунту* иначе работать не будет, также не работает если вкл двойная аутентификация.

settings.py

```
...

#EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'

EMAIL_USE_TLS = True
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 587
EMAIL_HOST_USER = DEFAULT_FROM_EMAIL = 'ваша эл почта@gmail.com'
EMAIL_HOST_PASSWORD = 'пароль вашей эл почты'
```

Закомментируйте параметр **EMAIL_BACKEND**



В шаблонах для отображения данных пользователя используйте **{% load account %} {{ user.email }}, {{ user.phone }} ...**