МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И

МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

«МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

СВЯЗИ И ИНФОРМАТИКИ»

Кафедра «Сети связи и системы коммутации»

Лабораторная работа №1

«Начало работы с Python»

по дисциплине «Web-программирование»

Выполнил студент

группы БФИ1901

Шацкий Е.И.

Проверила:  
Полянцева К.А.

Москва, 2021

**Содержание**

[1 Задание на лабораторную работу 3](#_Toc82640250)

[2 Ход лабораторной работы. 3](#_Toc82640251)

[2.1 Создании программы фильтрации студентов. 3](#_Toc82640252)

[2.2 Начало работы с веб-фреймворком Django. 5](#_Toc82640253)

[2.3 Работа с административным интерфейсом Django. 7](#_Toc82640254)

[Список используемых источников 13](#_Toc82640255)

**1 Задание на лабораторную работу**

1. Необходимо написать функцию для фильтрации студентов по среднему арифметическому их экзаменационных оценок;

2. Изучить основы работы с веб-фреймворком Django;

3. Изучить интерфейс административного приложения Django;

4. Через интерфейс административного приложения создать нового пользователя с правами «суперпользователя»;

5. Через интерфейс административного приложения создать нового пользователя без прав «суперпользователя»;

6. Через интерфейс административного приложения «забанить» одного из пользователей (сделать пользователя «неактивным»);

7. Загрузить проект на любой гит-репозиторий (GitHub, GitLab, Google Code, Bitbucket и т.п.).

**2 Ход лабораторной работы.**

**2.1 Создании программы фильтрации студентов.**

В качестве первого задания, необходимо написать программу, которая будет показывать список студентов, соответствующих среднему баллу, введенному пользователем с клавиатуры. Программный код представлен на листинге 1. Результат работы программы представлен на рисунке 1.

Листинг 1 – программный код

groupmates = [  
 {  
 "name": "Жариков",  
 "surname": "Владислав",  
 "exams": ["Информатика", "ЭЭиС", "ООП"],  
 "marks": [4, 3, 5]  
 },  
 {  
 "name": "Иван",  
 "surname": "Штыков",  
 "exams": ["Экономика", "АиГ", "КТП"],  
 "marks": [4, 4, 4]  
 },  
 {  
 "name": "Феликс",  
 "surname": "Смирнов",  
 "exams": ["СИАОД", "ИС", "КТП"],  
 "marks": [5, 5, 3]  
 },  
 {  
 "name": "Максим",  
 "surname": "Агарков",  
 "exams": ["История", "АиГ", "Социология"],  
 "marks": [4, 4, 4]  
 },  
 {  
 "name": "Никита",  
 "surname": "Кокчеринский",  
 "exams": ["Философия", "ИС", "Право"],  
 "marks": [4, 4, 4]  
 }  
]  
  
def print\_students(students):  
 print(u"Имя".ljust(15), u"Фамилия".ljust(10), u"Экзамены".ljust(30), u"Оценки".ljust(20))  
  
 for student in students:  
 print(student["name"].ljust(15), student["surname"].ljust(10), str(student["exams"]).ljust(30),  
 str(student["marks"]).ljust(20))  
  
 print("Введите средний балл студента: ")  
 avgMark = input()  
 print("Список студентов с указанным средним баллом")  
 print(u"Имя".ljust(15), u"Фамилия".ljust(10), u"Экзамены".ljust(30), u"Оценки".ljust(20), u"Средний Балл".ljust(5))  
  
 for student in students:  
 avg = round(sum(student["marks"]) / len(student["marks"]))  
 if str(avgMark) == str(avg):  
 print(student["name"].ljust(15), student["surname"].ljust(10), str(student["exams"]).ljust(30),  
 str(student["marks"]).ljust(20), str(avg).ljust(5))  
  
  
print\_students(groupmates)

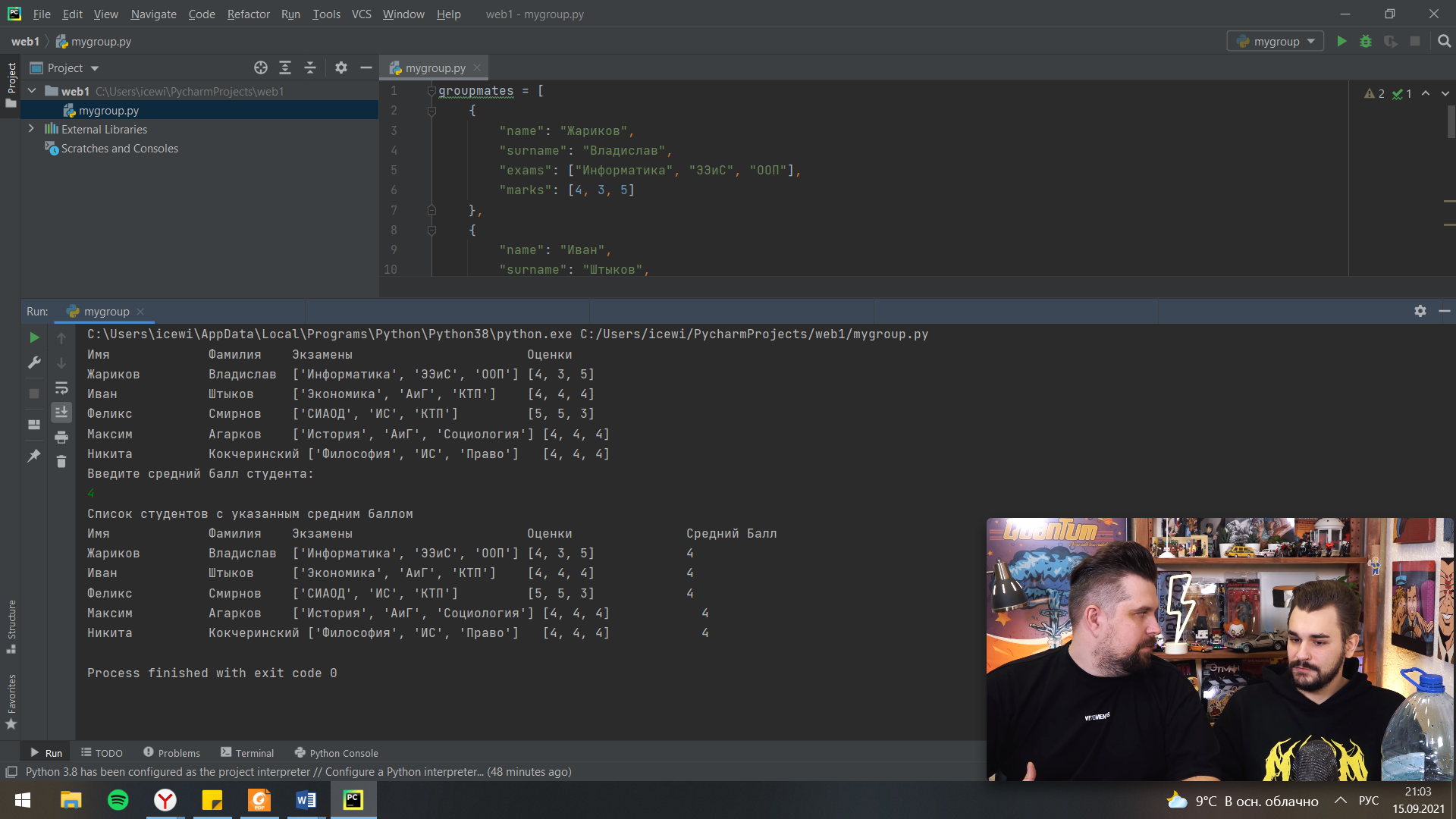


Рисунок 1 – Результат работы программы

**2.2 Начало работы с веб-фреймворком Django.**

Для начала необходимо создать проект с веб-фреймворком c помощью команды «Django-admin startproject webdjango1», который автоматически создаст файлы, которые необходимы для выполнения лабораторной работы. Для того, чтобы убедиться в том, что проект был создан корректно, необходимо ввести в командной строке команду “python manage.py runserver”, предварительно перейдя в директорию с проектом, а затем перейти по адресу <http://127.0.0.1:8000/> в браузере. Результат представлен на рисунках 2 и 3.

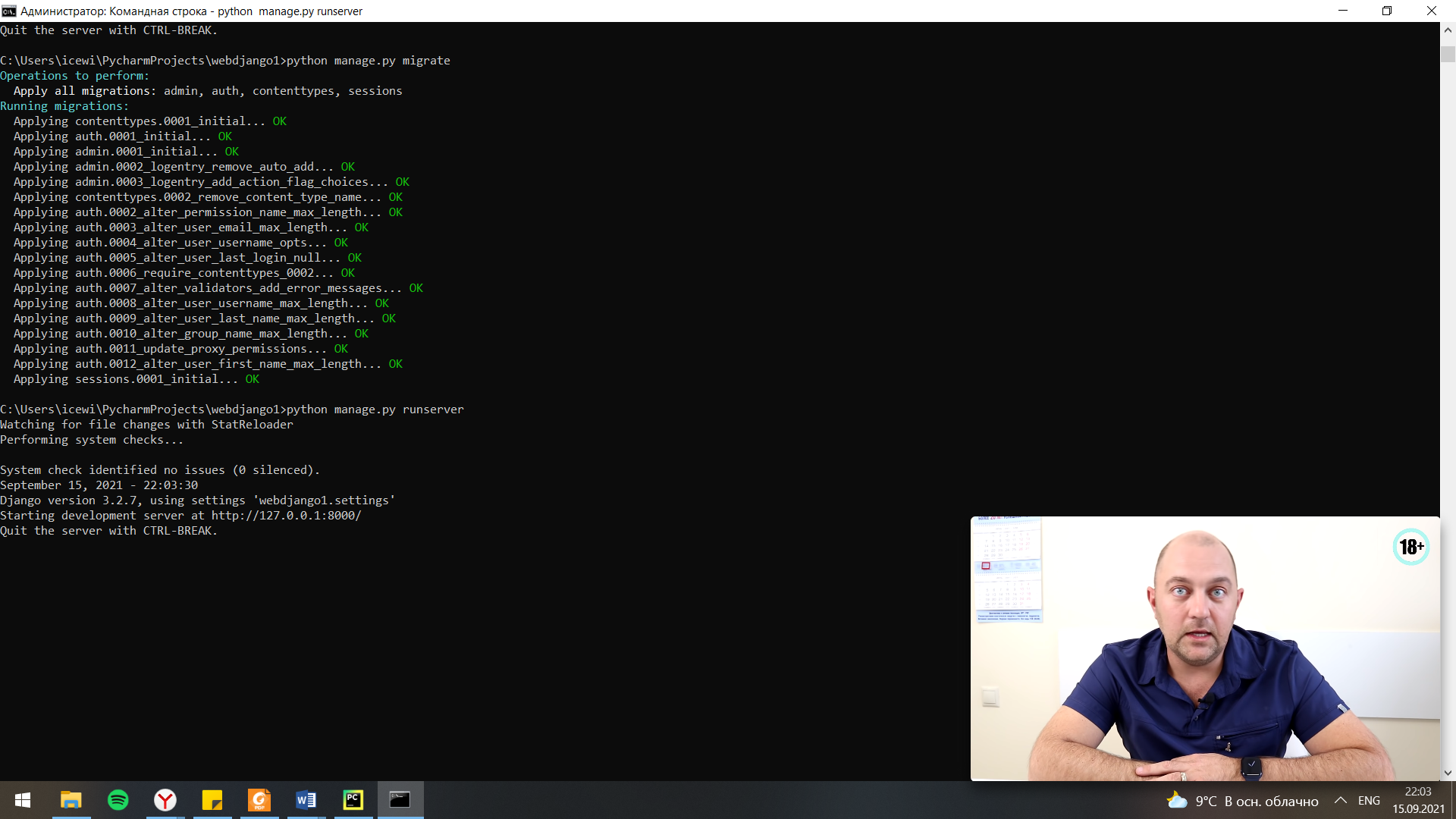


Рисунок 2 – Командная строка после запуска локального сервера

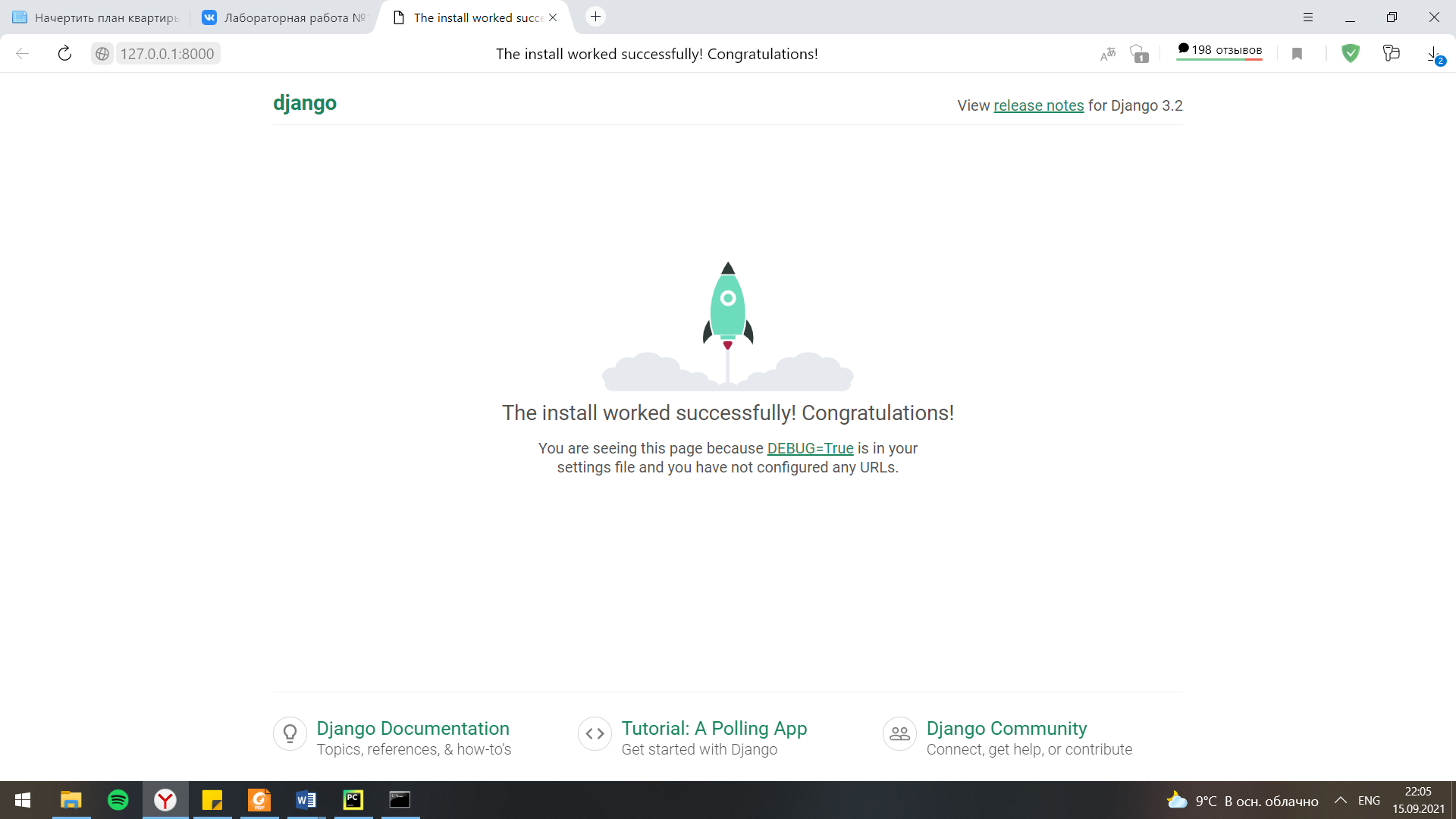


Рисунок 3 –Успешный запуск сервера

**2.3 Работа с административным интерфейсом Django.**

Далее, по плану лабораторной работы, необходимо в файле settings.py измените параметр базы данных. Для начала добавляем **import os** перед **from pathlib import Path** (Рисунок 4) это необходимо для успешного внесения изменений. В переменной DATABASES есть внутренний словарь с ключами ENGINE и NAME. Для параметра NAME устанавливаю следующее значение (рисунок 5):

**'NAME': os.path.join(BASE\_DIR, 'db\_project\_name')**

'db\_project\_name' – данный параметр является названием базы данных, так что он может быть любым.

Затем для завершения создания таблицы базы данных, необходимо в командной строке выполнить команду “python manage.py migrate”. На листинге 2 представлен программный код файла settings.py. На рисунке 4 представлен результат создания таблицы базы данных.

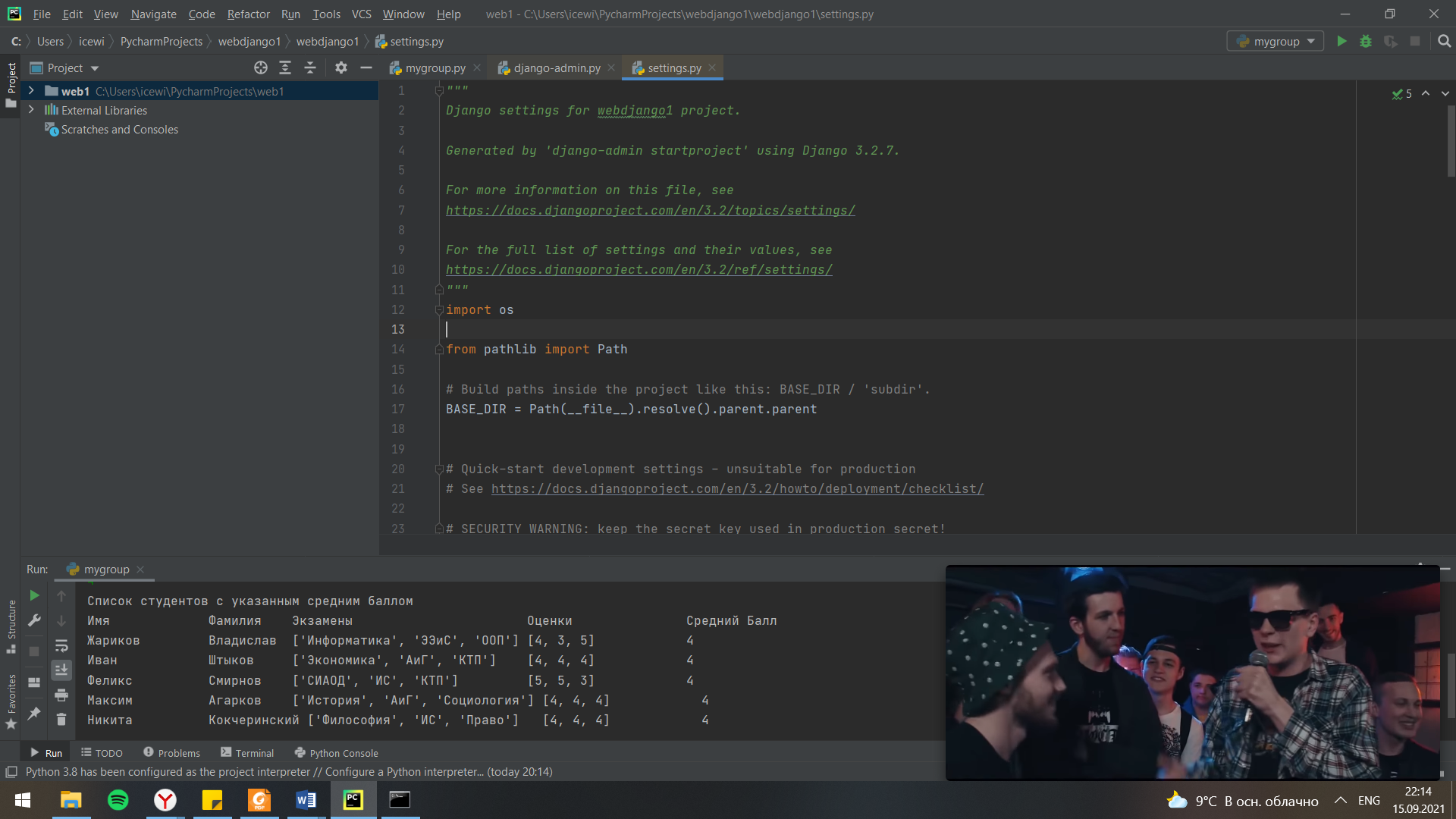


Рисунок 4 – добавление import os

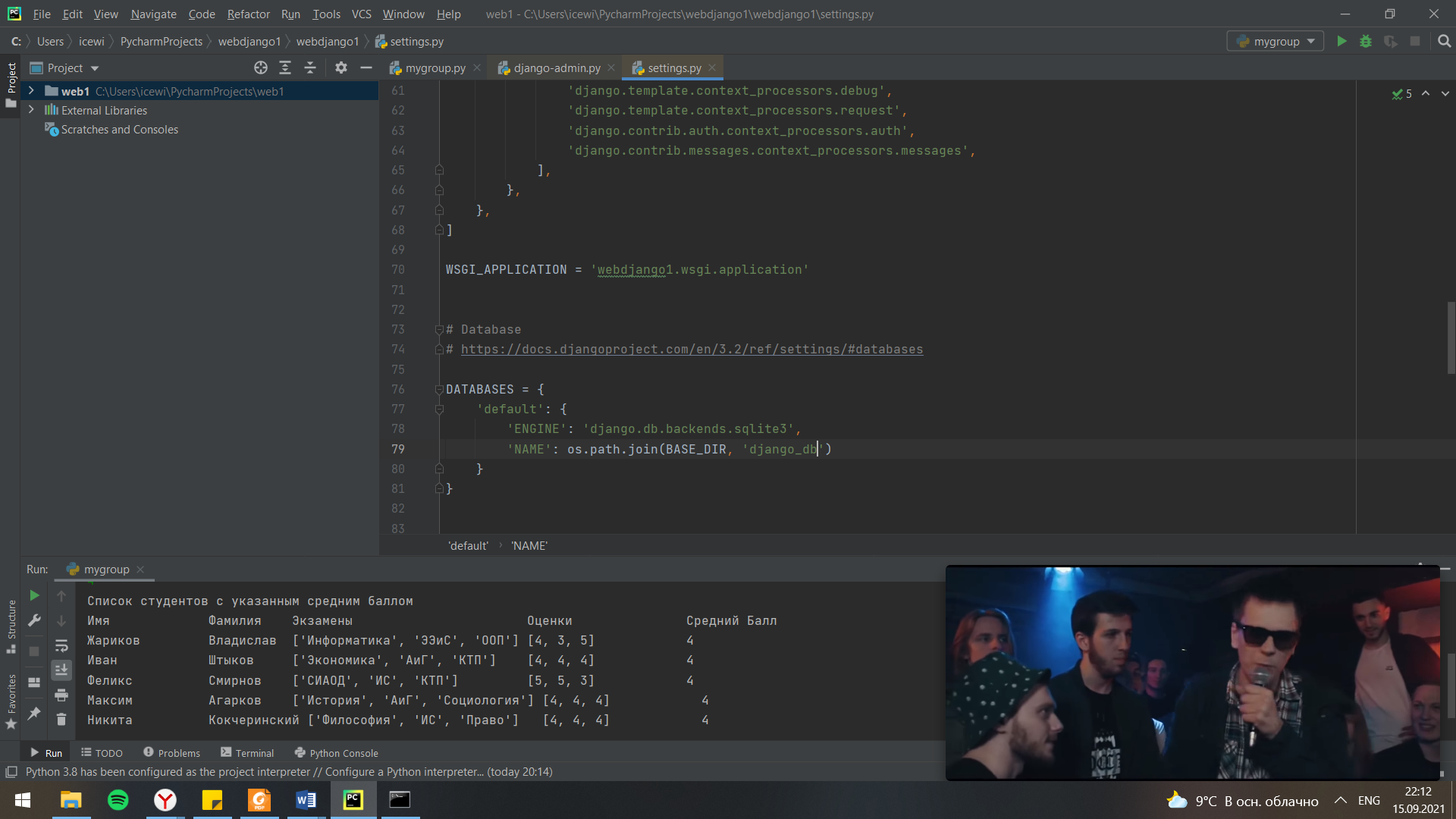


Рисунок 5 – изменение ключа NAME

Чтобы создать таблицы базы данных, необходимо перейти в директорию webdjango1 и через командную строку выполнить команду (рисунок 6)

**python manage.py migrate**

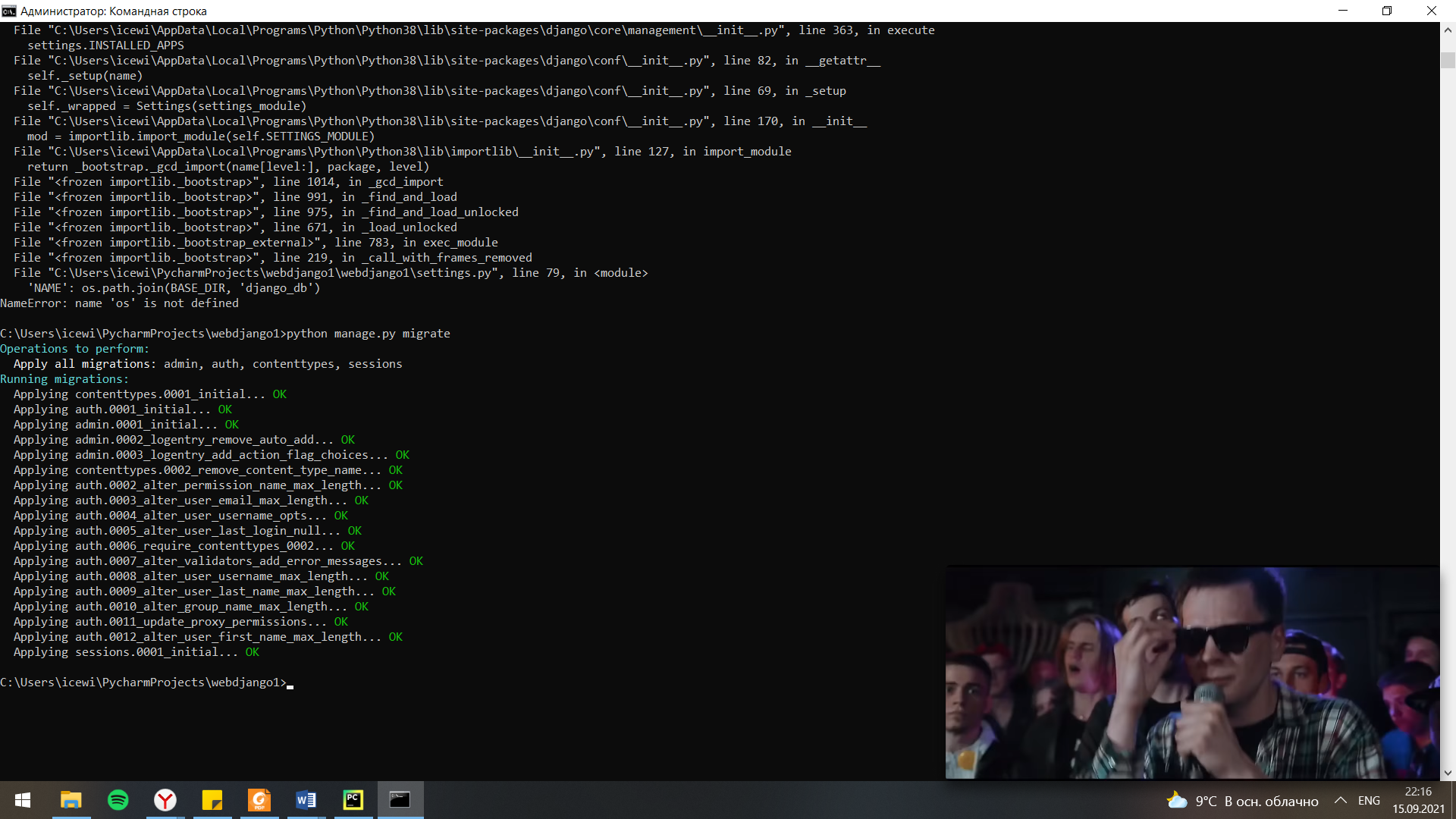


Рисунок 6 – Создание таблицы базы данных

Затем создам суперпользователя (при создании необходимо ввести логин, email и пароль пользователя) (рисунок 7):

**python manage.py createsuperuser**

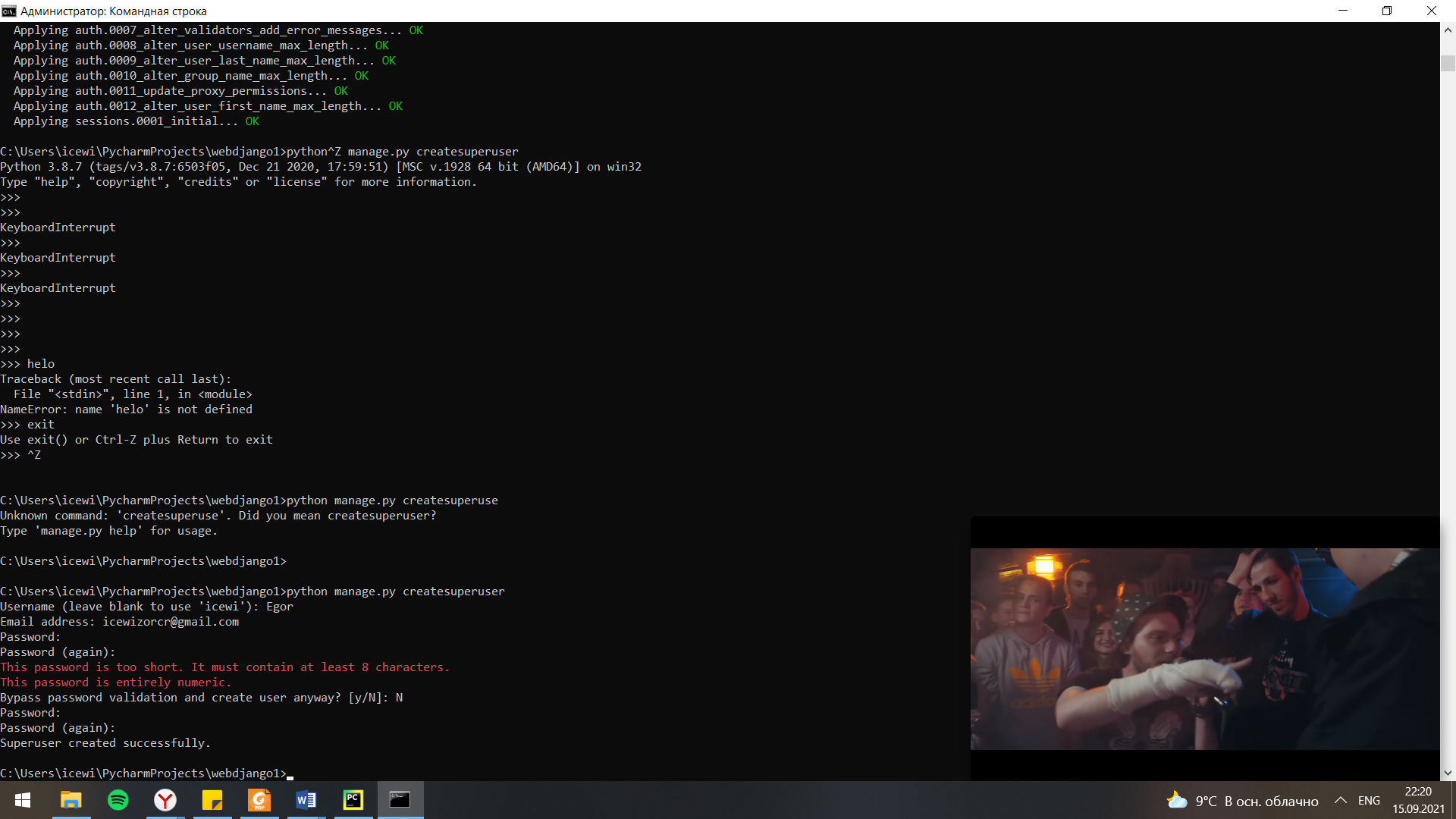


Рисунок 7 – создание суперпользователя

Теперь запускаю сервер командой:

**python manage.py runserver**

и в браузере пройти по адресу <http://127.0.0.1:8000/admin/> (рисунок 8).

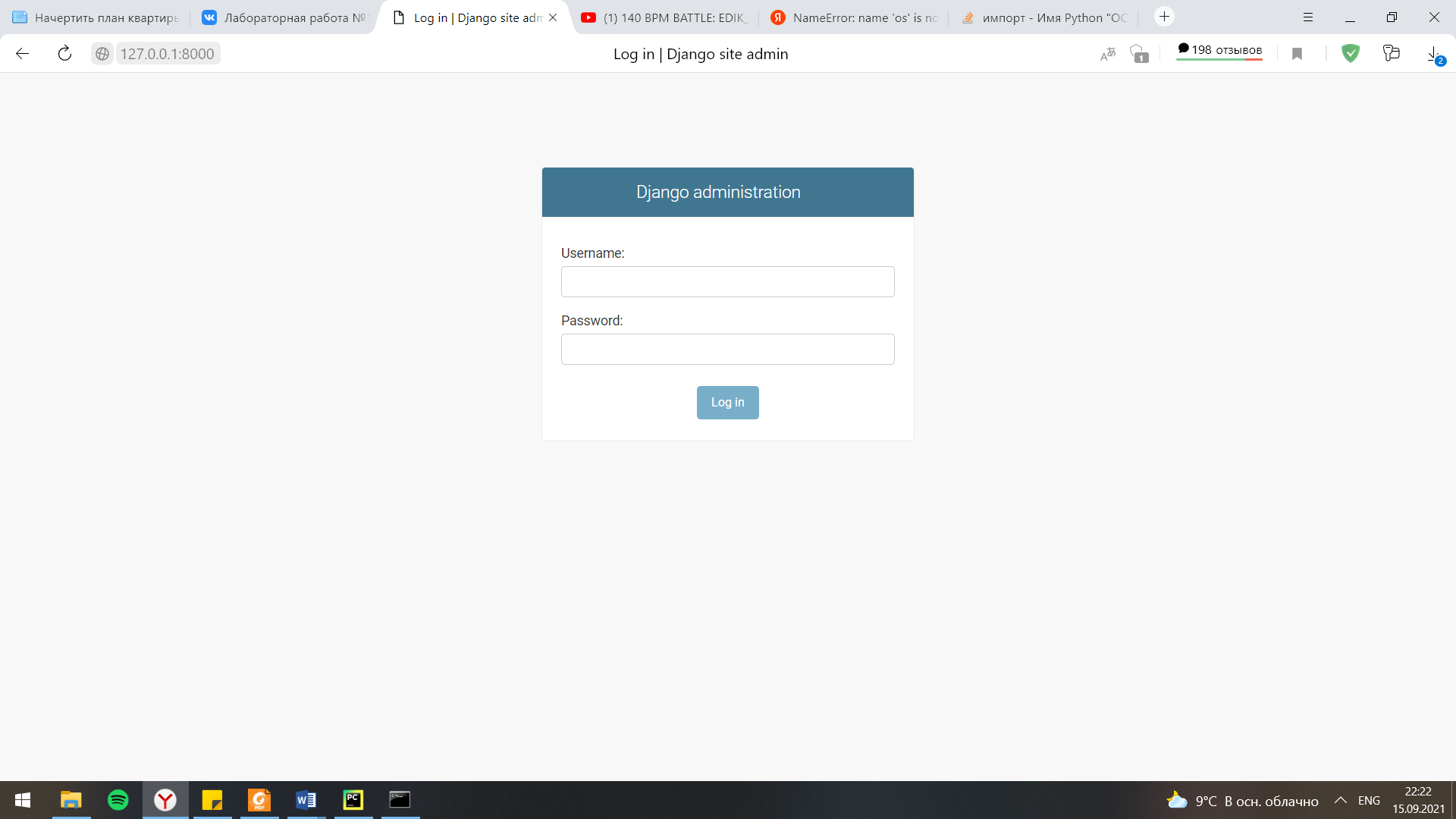


Рисунок 8 – вход в панель администратора

После этого должно появиться окно входа, куда я ввожу данные, которые указал при создании суперпользователя. Когда я успешно вошел в аккаунт суперюзера, то откроется главная страница административной панели (рисунок 9), через которую я могу управлять приложениями, редактируя существующие записи в базе данных или генерируя новые.

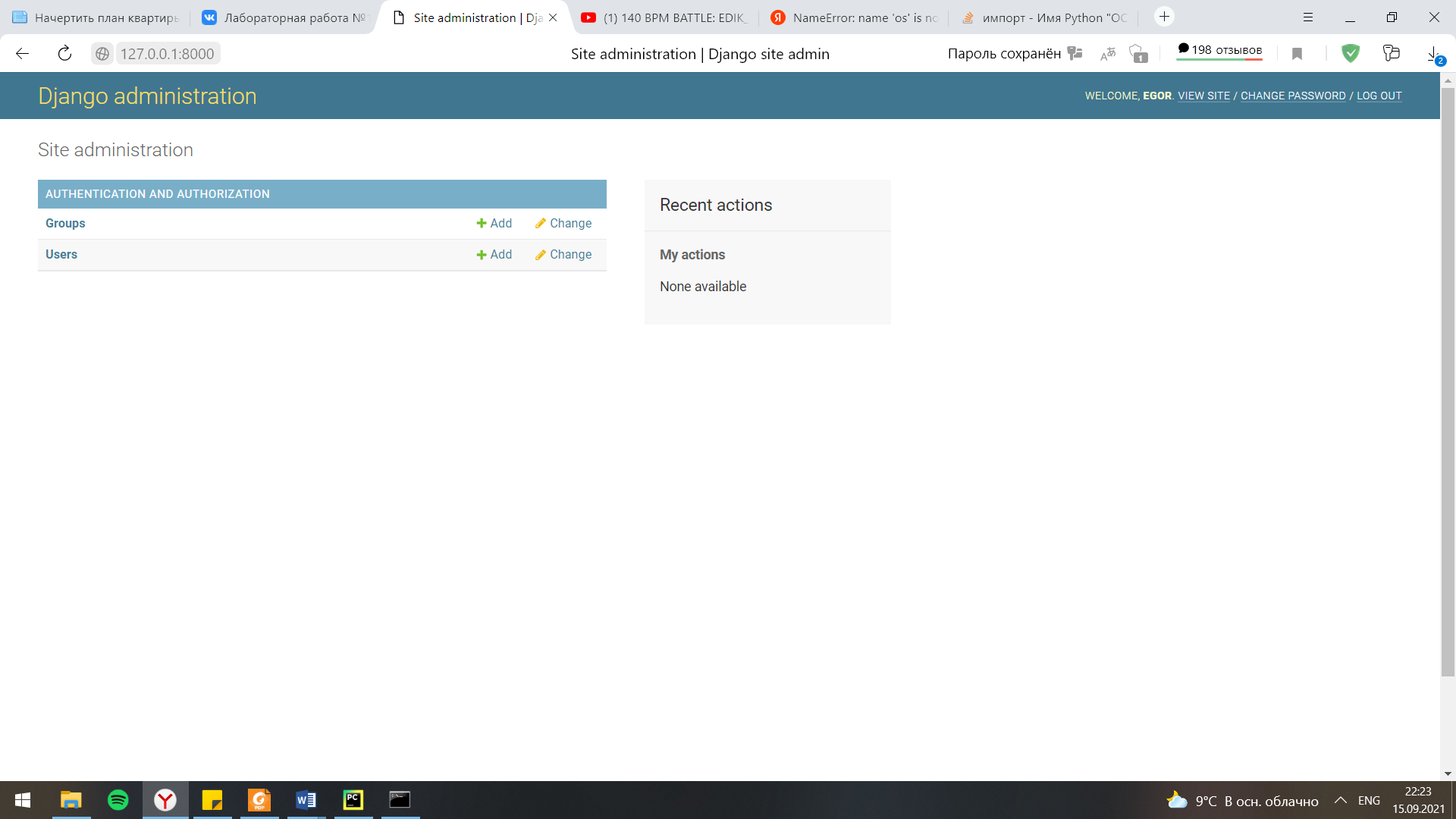


Рисунок 9 – Панель администратора

Через интерфейс административного приложения создаю нового пользователя с правами суперпользователя (Felix) (рисунок 10);

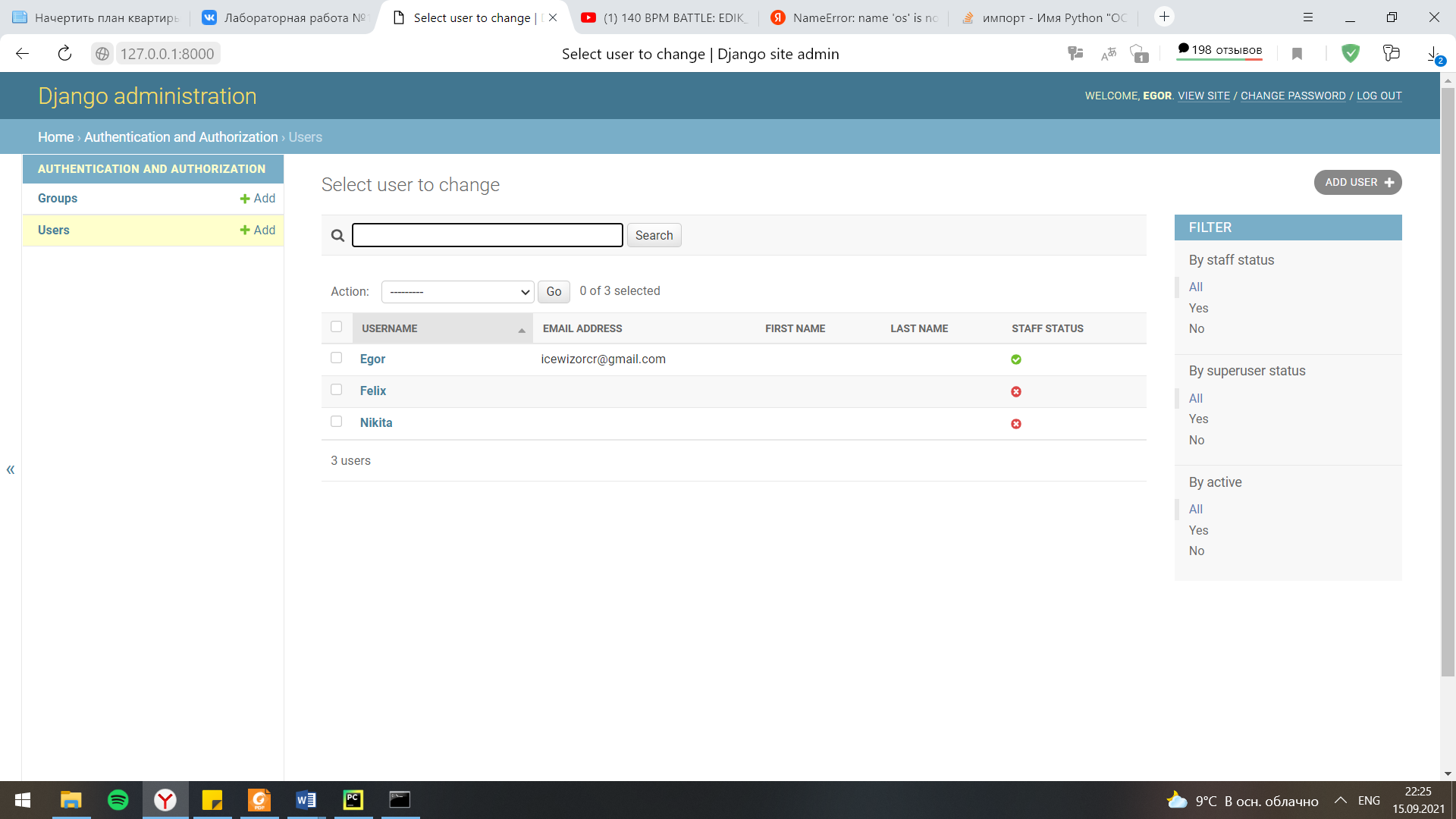


Рисунок 10 – Создание пользователей

Через интерфейс административного приложения создаю нового пользователя без прав суперпользователя (Nikita) (рисунок 11);

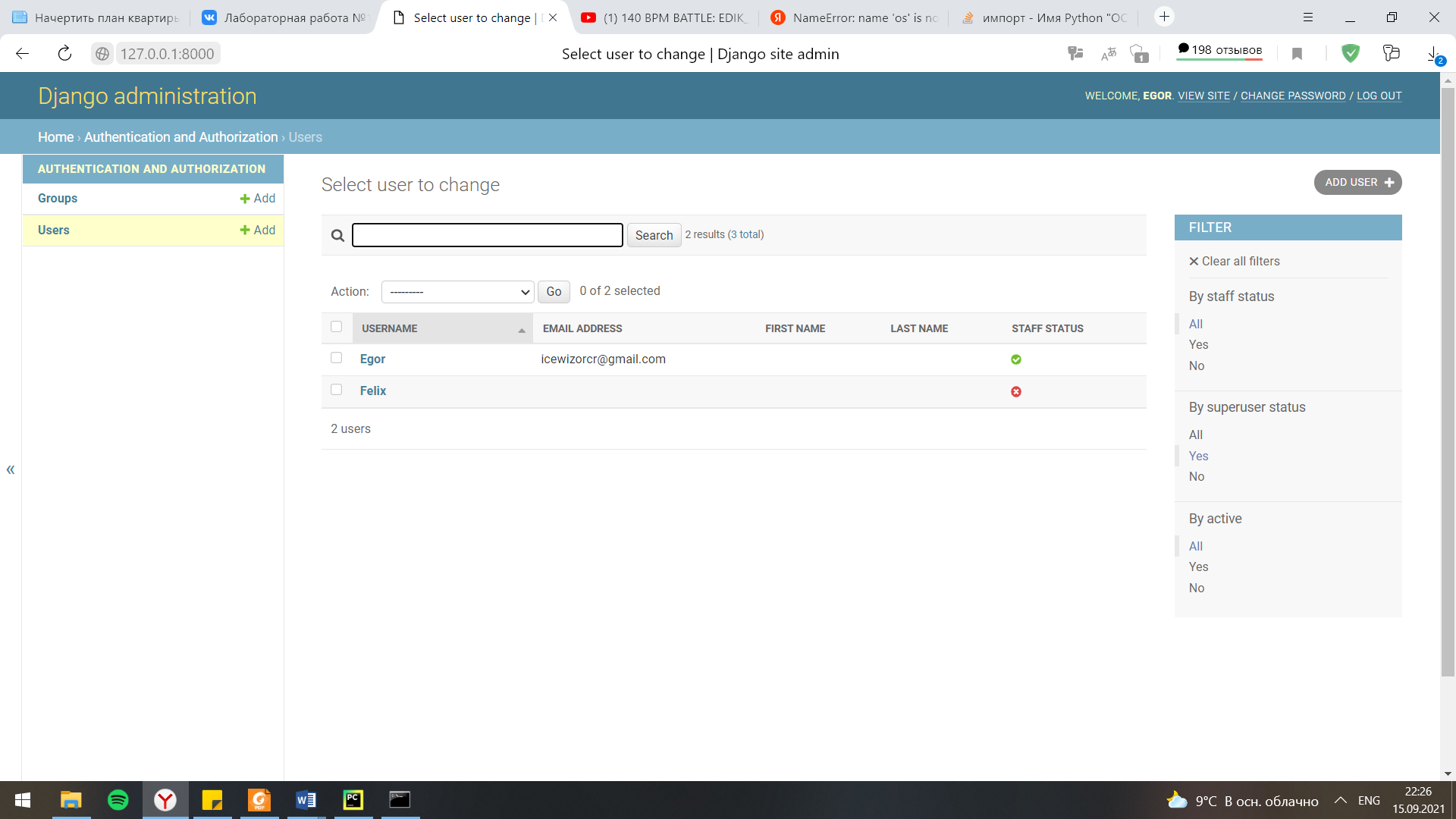


Рисунок 11 – пользователи с правом суперпользователя

Через интерфейс административного приложения «забаню» одного из пользователей (делаю пользователя «неактивным») (рисунок 12);

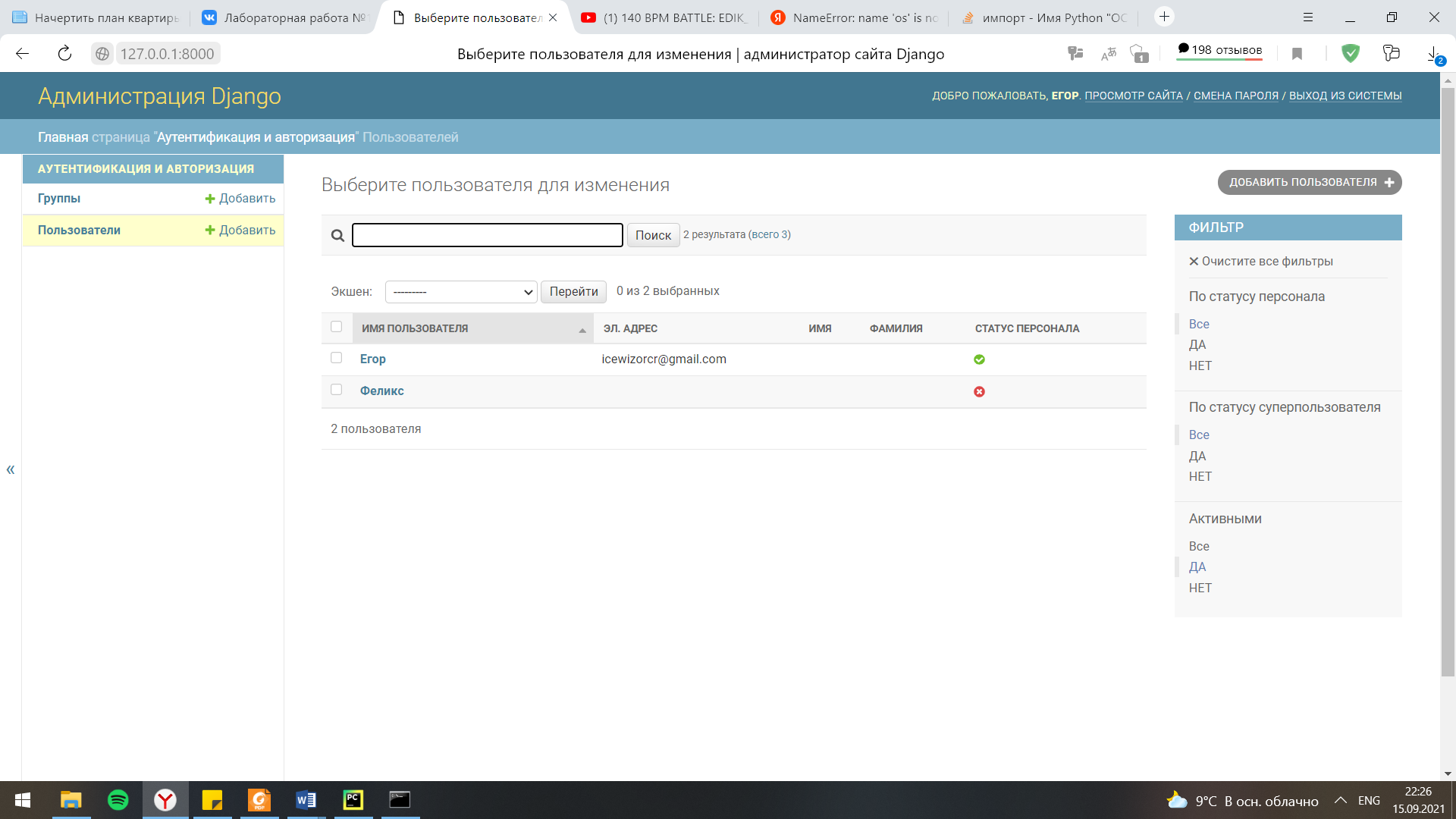


Рисунок 12 – активные пользователи

**Вывод:** в данной лабораторной работе, был реализован алгоритм фильтрации студентов по среднему баллы, а также были изучены основы работы в веб-фреймворком Django, как в командной строке, так и в административной панели.

**Список используемых источников**

1 ГОСТ 7.32-2017 СИБИД. Отчет о научно-исследовательской работе. Структура и правила оформления.

2 ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание.