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

по дисциплине «Web-программирование»

Выполнил студент

группы БФИ1901

Кочеринский Н.В.

Проверил:

Москва, 2021

# **1 Задание на лабораторную работу**

1. Необходимо написать функцию фильтрации студентов, по средней оценке;

2. Изучить основы работы с веб-фреймворком Django;

3. Изучить интерфейс административного приложения Django;

4. Через интерфейс административного приложения создайть нового пользователя с правами суперпользователя;

5. Через интерфейс административного приложения создайть нового пользователя без прав суперпользователя;

6. Через интерфейс административного приложения «забанить» одного из пользователей (сделайте 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.

По плану лабораторной работы, необходимо создать программу, которая будет фильтровать студентов по среднему баллу, который будет вводить пользователь с клавиатуры. Программный код представлен на листинге 1. Результат работы программы представлен на рисунке 1.

Листинг 1 – Программный код.

groupmates = [  
 {  
 "name": "Агарков",  
 "surname": "Максим",  
 "exams": ["Информатика", "ЭЭиС", "Web"],  
 "marks": [4, 3, 5]  
 },  
  
 {  
 "name": "Смирнов",  
 "surname": "Феликс",  
 "exams": ["История", "АиГ", "КТП"],  
 "marks": [4, 4, 4]  
 },  
  
 {  
 "name": "Шацкий",  
 "surname": "Егор",  
 "exams": ["Философия", "ИС", "МЛИТА"],  
 "marks": [5, 5, 4]  
 },  
  
 {  
 "name": "Соцков",  
 "surname": "Игнатий",  
 "exams": ["Социология", "АГИЛА", "ВМ"],  
 "marks": [4, 5, 4]  
 },  
  
 {  
 "name": "Копытько",  
 "surname": "Сергей",  
 "exams": ["СТ", "СП", "ФизРа"],  
 "marks": [5, 5, 5]  
 },  
  
 {  
 "name": "Крутиков",  
 "surname": "Степан",  
 "exams": ["ЭДО", "ДМ", "АВС"],  
 "marks": [4, 3, 3]  
 }  
  
]  
  
  
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 срдений балл: ")  
 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)

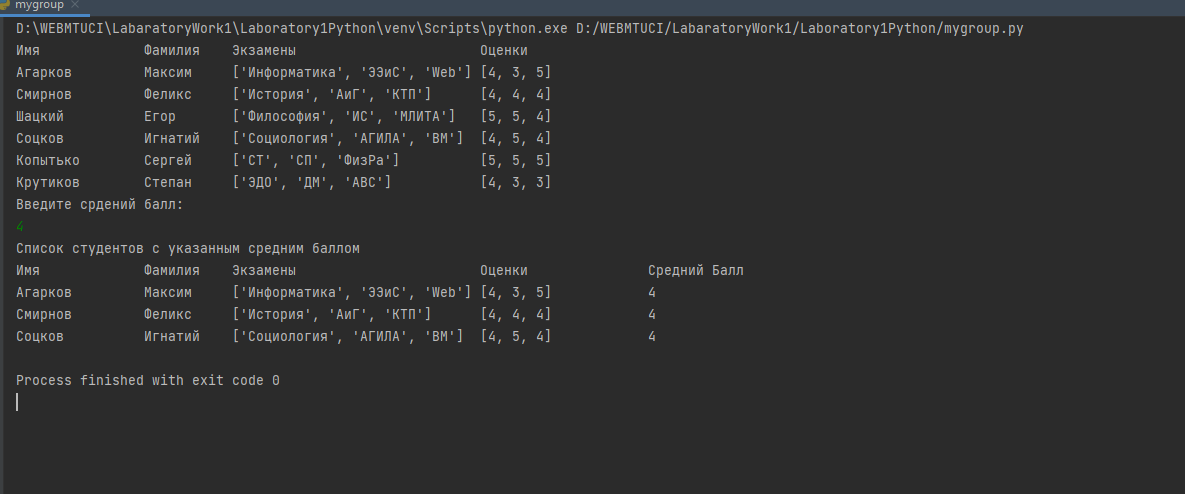


Рисунок 1 – Результат работы программы.

2.2 Начало работы с веб-фреймворком Django.

Для начала необходимо создать проект с веб-фреймворком, который автоматически создаст файлы, которые необходимо для выполнения лабораторной работы. Что бы 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/>. Результат этих действий представлен на рисунках 2 и 3.

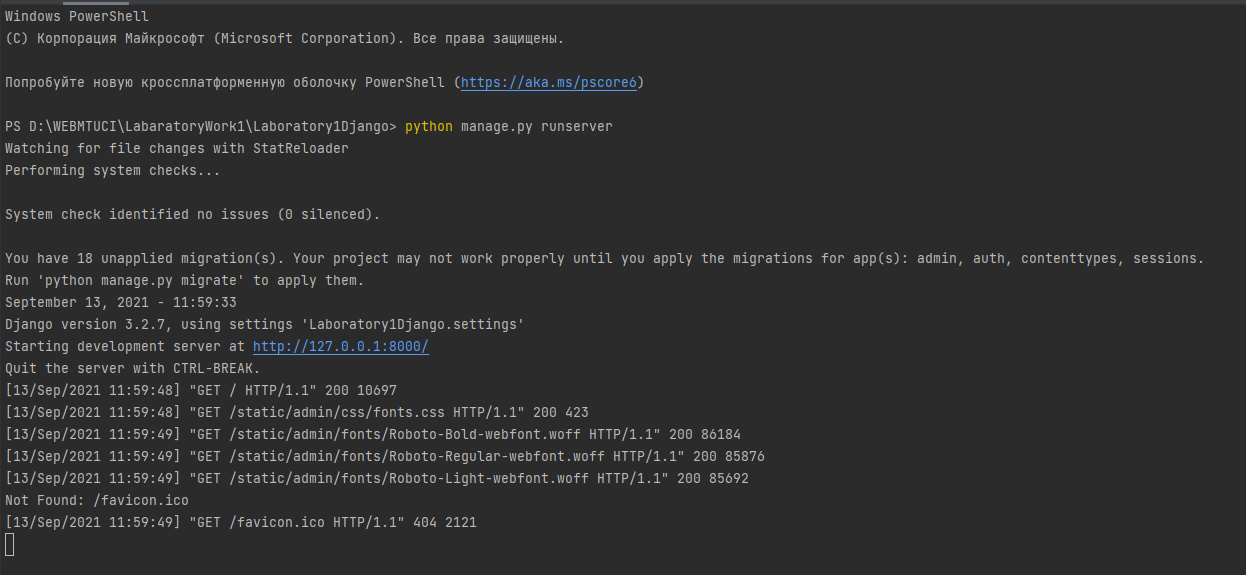


Рисунок 2 – Состояние командной строки после запуска локального сервера.

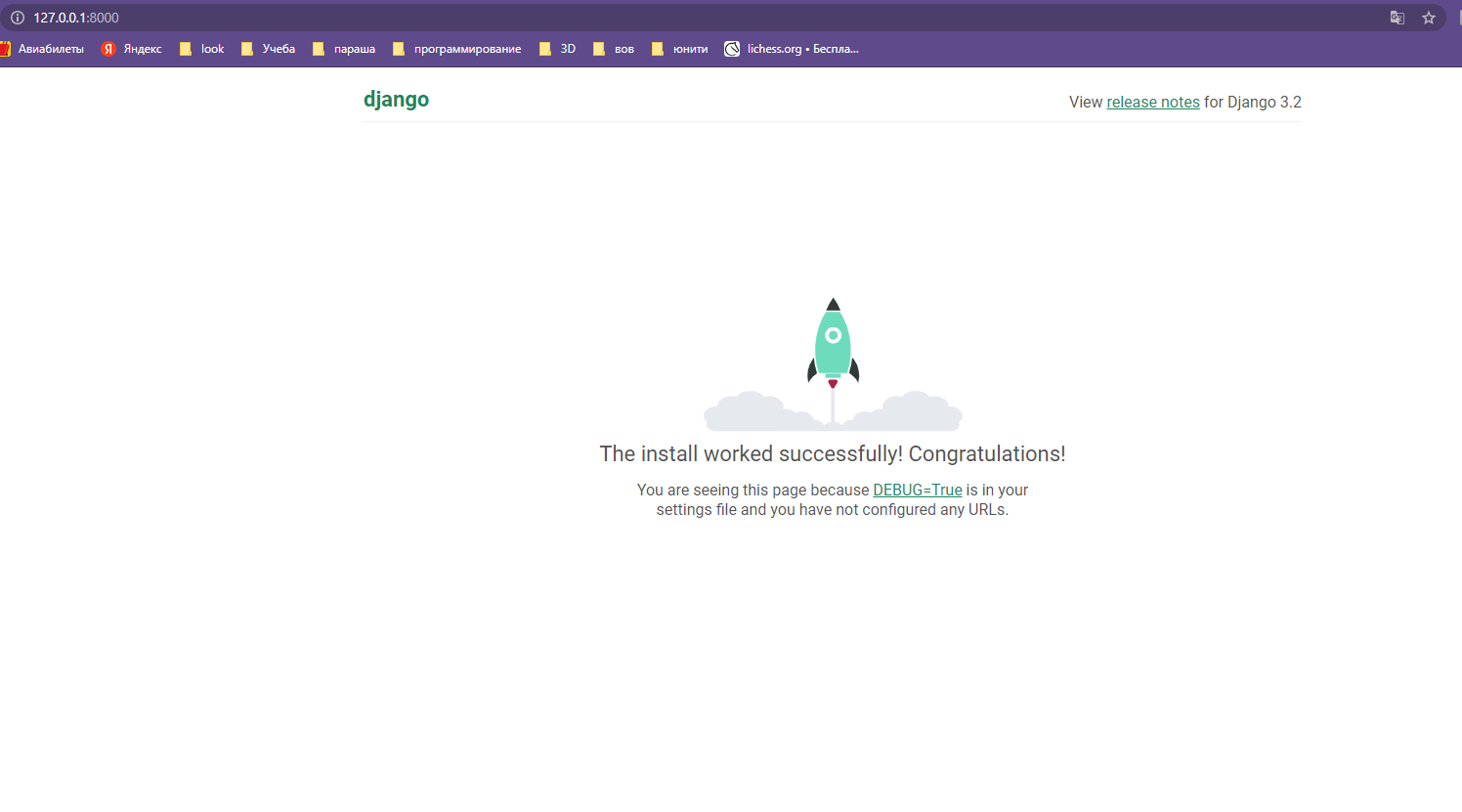


Рисунок 3 – Сообщение об успешном запуске сервера.

2.3 Работа с административным интерфейсом Django.

Далее, по плану лабораторной работы, необходимо В файле settings.py измените параметр базы данных. Переменной DATABASES есть внутренний словарь с ключами ENGINE и NAME. Для параметра NAME установите следующее значение:

'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.

Листинг 2 – Программный код файла setiings.py.

DATABASES = {  
 'default': {  
 'ENGINE': 'django.db.backends.sqlite3',  
 'NAME': os.path.join(BASE\_DIR, 'db\_nikita\_kocherinskiy'),  
 }  
}

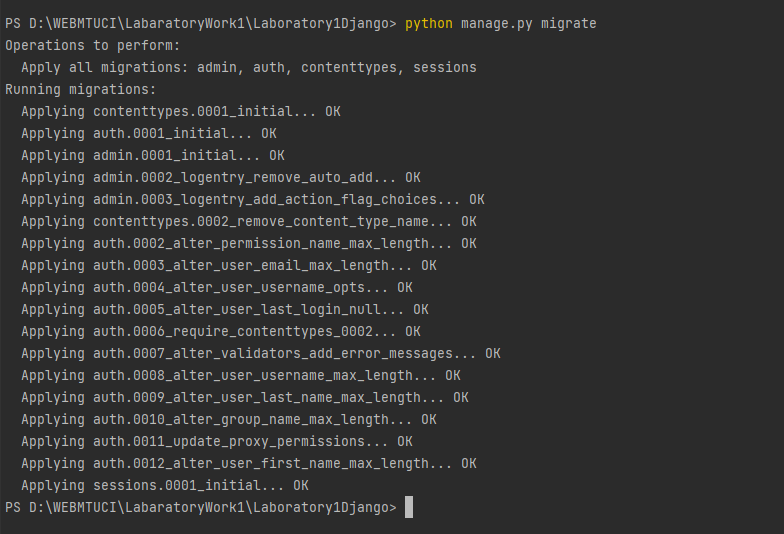


Рисунок 4 – Результат создания таблицы баз данных.

Далее необходимо создать суперпользователя с помощью командной строки, прописав в ней “python manage.py createsuperuser”. При необходимости ввести логин, email и пароль пользователя. Результат работы данной команды представлен на рисунке 5.

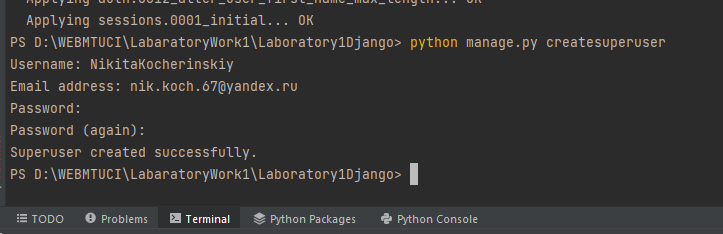


Рисунок 5 – Создание суперпользователя.

Далее необходимо перейти по адресу <http://127.0.0.1:8000/admin/>, где откроется окно куда необходимо ввести данные, которые были введены в командной строке. После ввода данный, откроется административная панель откуда осуществляется управление базами данных или их создание. На рисунке 6 представлен экран входа. На рисунке 7 представлена административная панель.

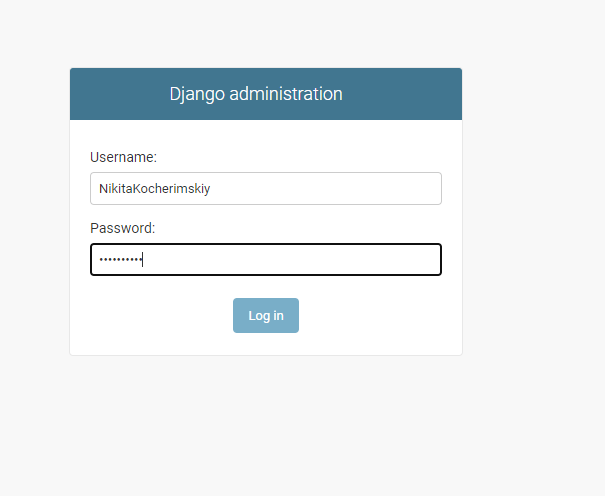


Рисунок 6 – Ввод данных

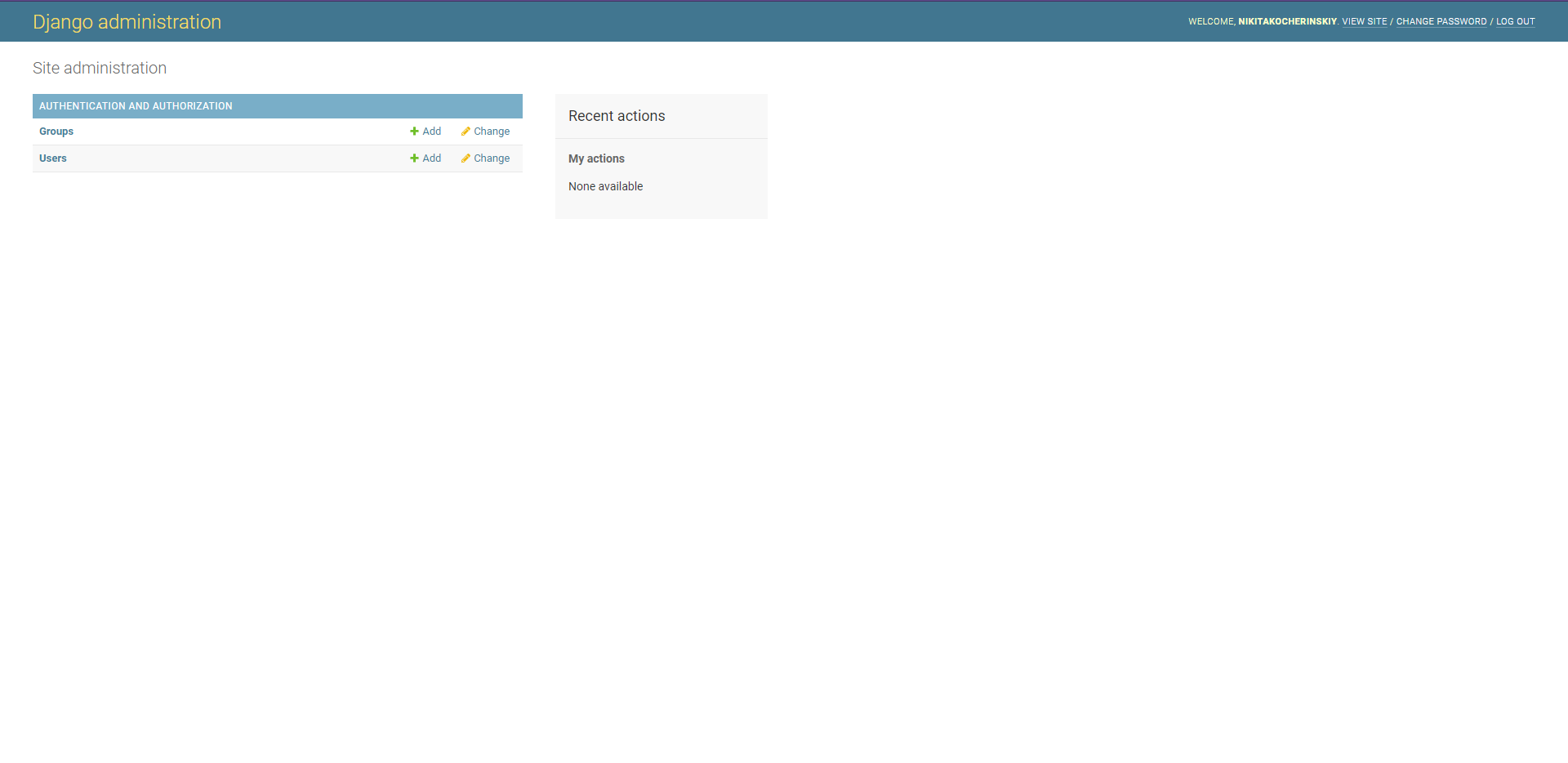


Рисунок 7 – Административная панель.

Далее необходимо создать нового пользователя с правами суперпользователя. Результат представлен на рисунках 8 и 9.

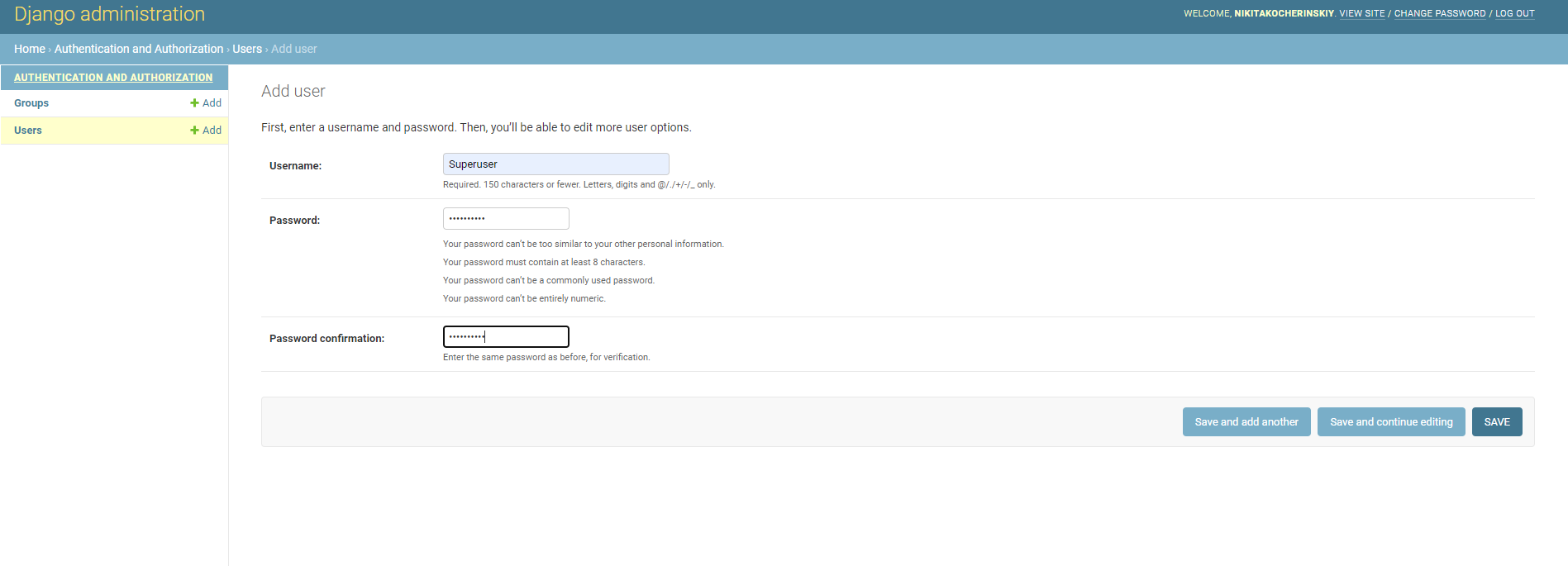


Рисунок 8 – Создание нового суперпользователя.

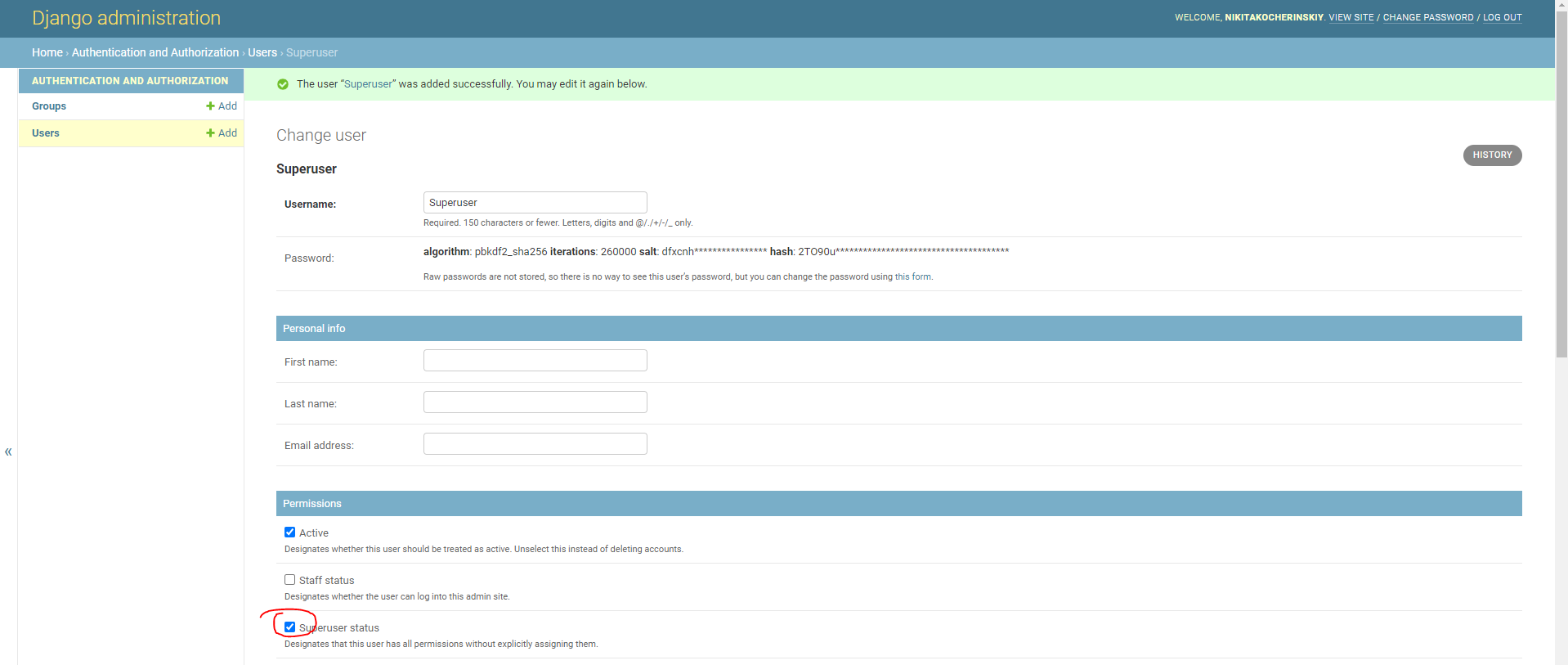


Рисунок 9 – Создание нового суперпользователя (2).

Затем, создать нового пользователя без прав суперпользователя. Результат представлен на рисунках 10 и 11.

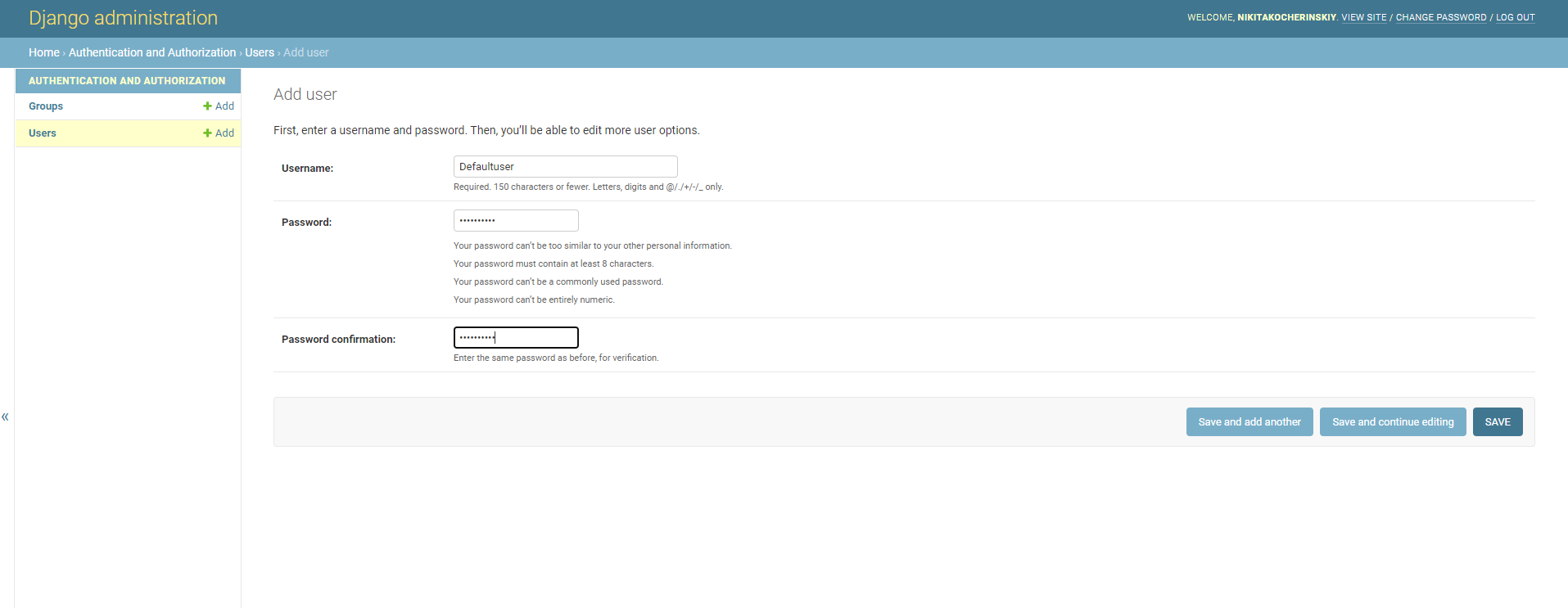


Рисунок 10 – Создание пользователя без прав суперпользователя.

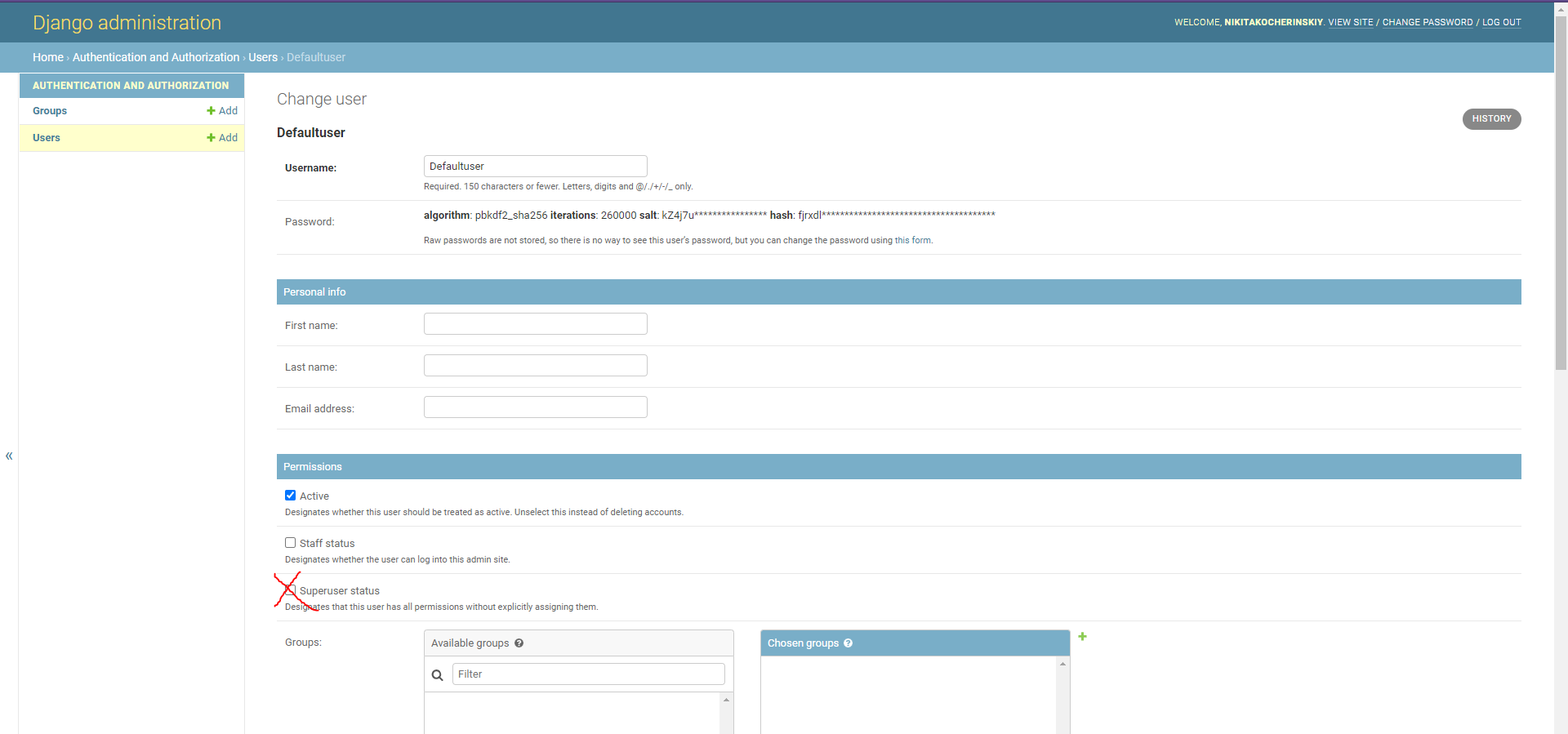


Рисунок 11 – Создание пользователя без прав суперпользователя.

Следом, необходимо «забанить» одного из пользователя, то есть сделать его неактивным. Результат представлен на рисунке 10.

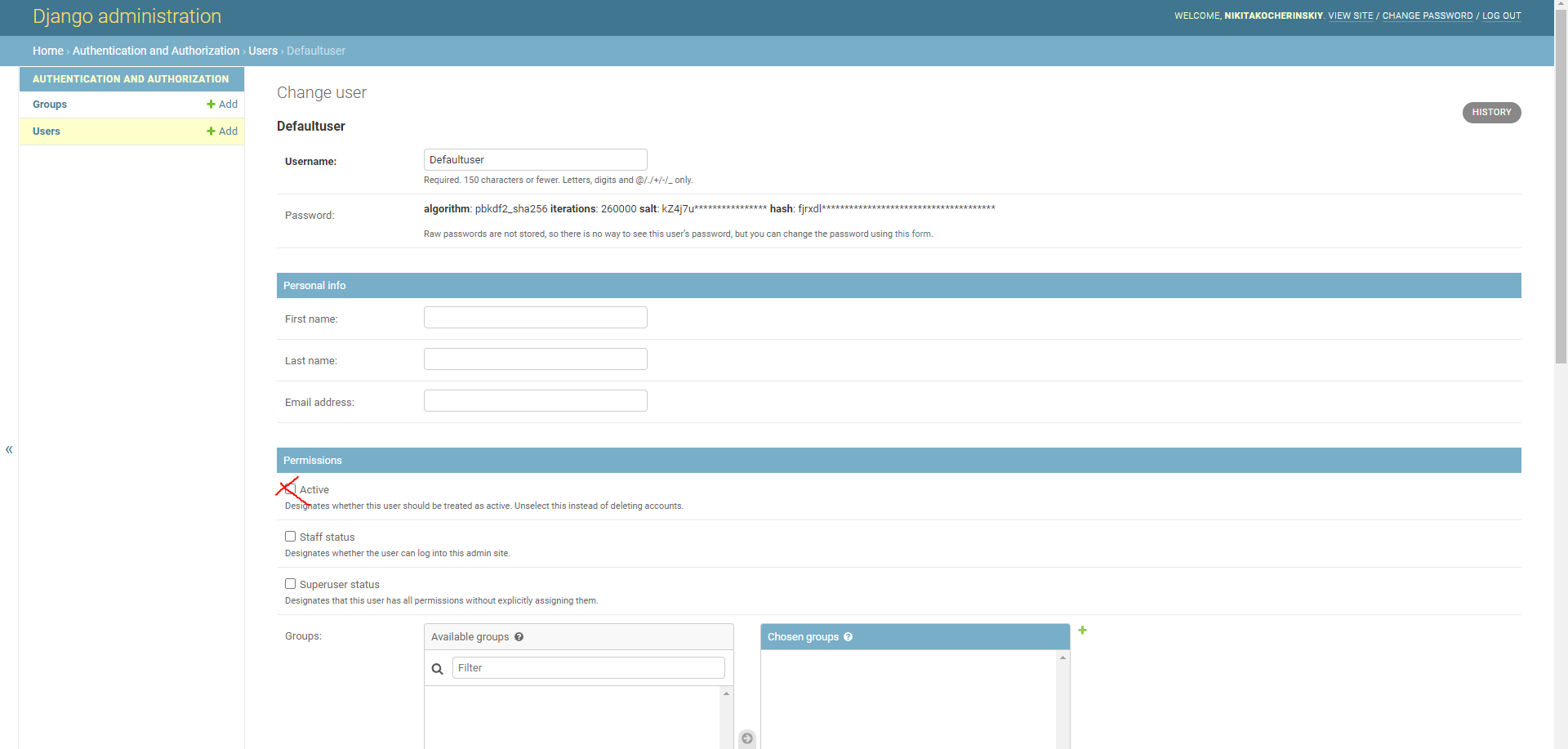
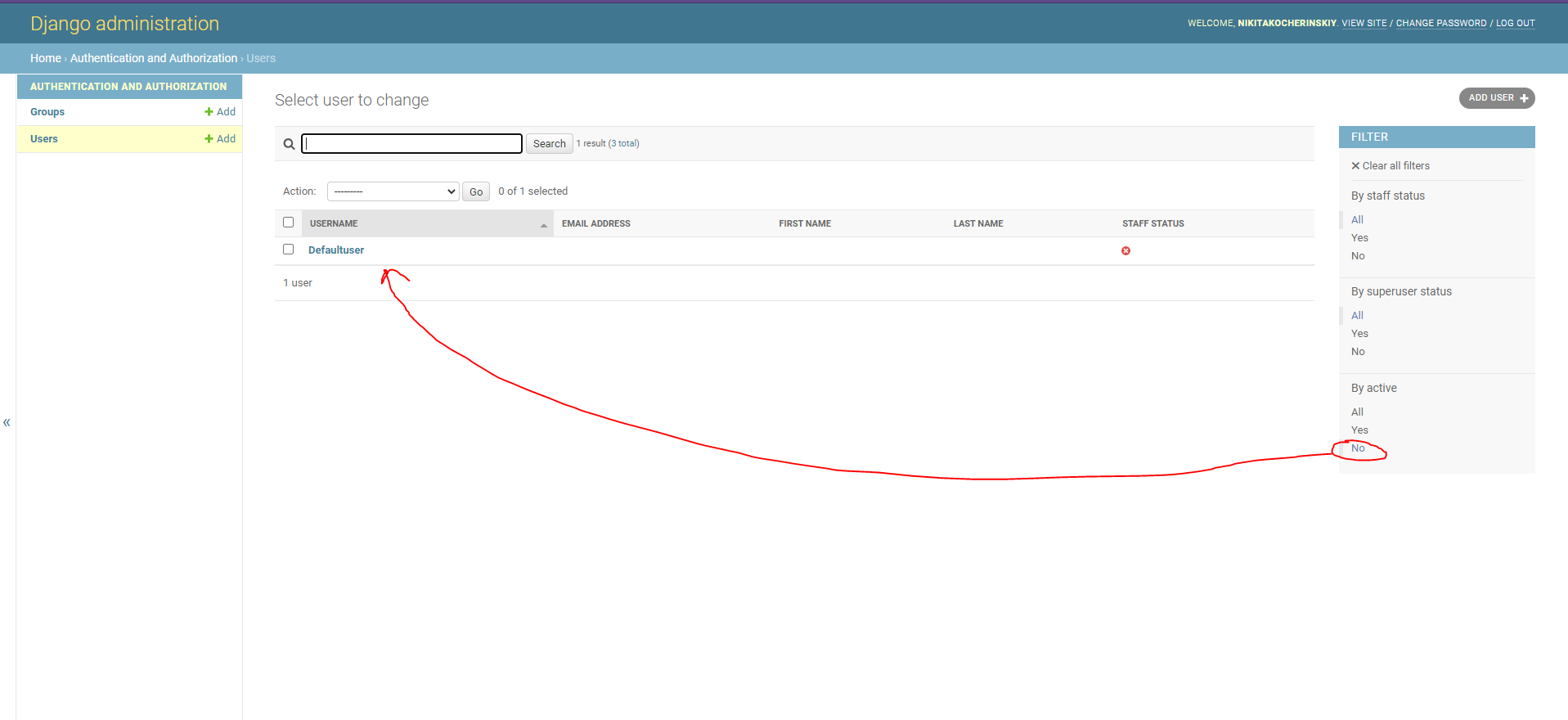


Рисунок 12 – «Бан» пользователя.

Что бы убедиться в том, что пользователь неактивен, выставил в фильтрах параметр, что бы показывались исключительно неактивные пользователи.



Для завершения лабораторной работы, необходимо загрузить её на любой гит репозиторий. Процесс загрузки представлен на рисунках 11 и 12.

Рисунок 13 – Загрузка лабораторной работы на гит репозиторий.

Рисунок 14 – Загрузка лабораторной работы на гит репозиторий (2).

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.