**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РФ**

**Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования «Московский технический университет связи и информатики»**

Кафедра математической кибернетики и информационных технологий

Лабораторная работа №1 по дисциплине «Web-программирование»:

«Начало работы с Python»

Выполнил студент

группы БФИ1901

Гусев Никита

Проверила:

Полянцева Ксения Андреевна

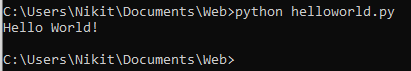
Москва 2021 г.

**Ход работы**

1. **Начало работы**

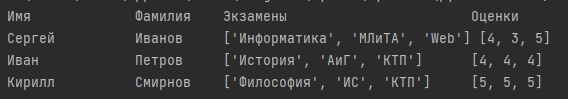
Создал директорию для лабораторных работ. В этой папке создал файл «helloworld.py». Используя редактор кода IDLE добавил в созданный файл код: **print("Hello world!").**

Запустил скрипт с помощью команды: python helloworld.py. Получил:

  
Рисунок 1 – Результат работы

1. **Задание №1**

Вам необходимо написать функцию фильтрации студентов по средней оценке, так, чтобы на экран выводился список студентов, средний балл которых выше заданного. Средний балл, по которому будет проводиться фильтрация, вводится пользователем с клавиатуры. Сформированный список студентов:

  
Рисунок 2 – Список студентов

Ниже представлен листинг функции фильтрации студентов по средней оценке:

groupmates = [

{

"name": "Александр",

"surname": "Иванов",

"exams": ["Информатика", "ЭЭиС", "Web"],

"marks": [4, 3, 5]

},

{

"name": "Иван",

"surname": "Петров",

"exams": ["История", "АиГ", "КТП"],

"marks": [4, 4, 4]

},

{

"name": "Кирилл",

"surname": "Смирнов",

"exams": ["Философия", "ИС", "КТП"],

"marks": [5, 5, 5]

}

]

def print\_students():

print(u"Имя".ljust(15), u"Фамилия".ljust(10), u"Экзамены".ljust(30), u"Оценки".ljust(20))

for student in groupmates:

print(student["name"].ljust(15), student["surname"].ljust(10), str(student["exams"]).ljust(30),

str(student["marks"]).ljust(20))

print\_students()

def count\_mark(groupmates):

print(u"Имя".ljust(15), u"Фамилия".ljust(8), u"Экзамены".ljust(8), u"Оценки".ljust(20))

for student in groupmates:

marks\_list = student['marks']

result = (sum(marks\_list) / len(marks\_list))

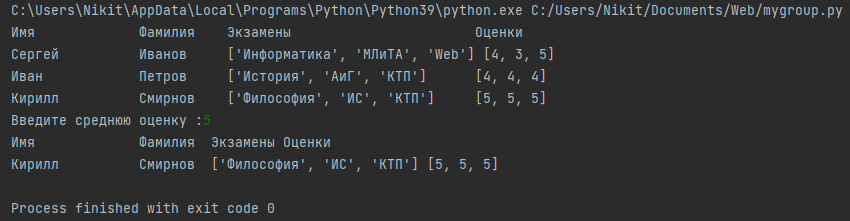
if result >= need:

print(student["name"].ljust(15), student["surname"].ljust(8), str(student["exams"]).ljust(8), str(student["marks"]).ljust(20))

need = int(input('Введите среднюю оценку: '))

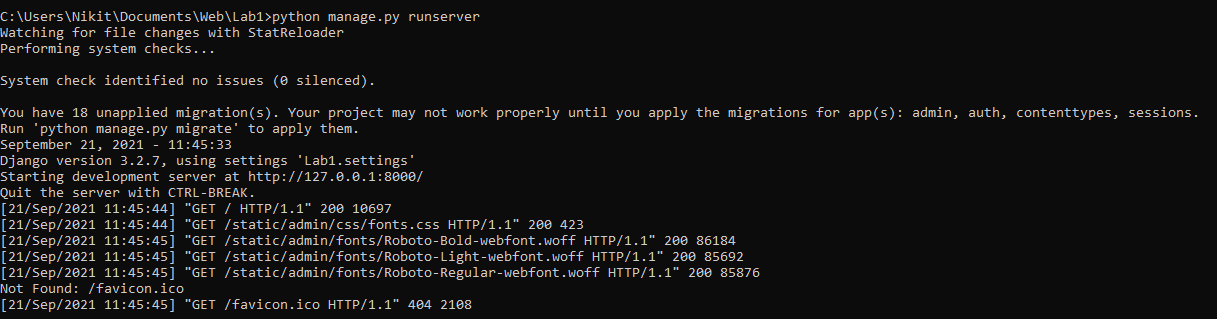
count\_mark(groupmates)

Результат работы показан на рисунке 3:

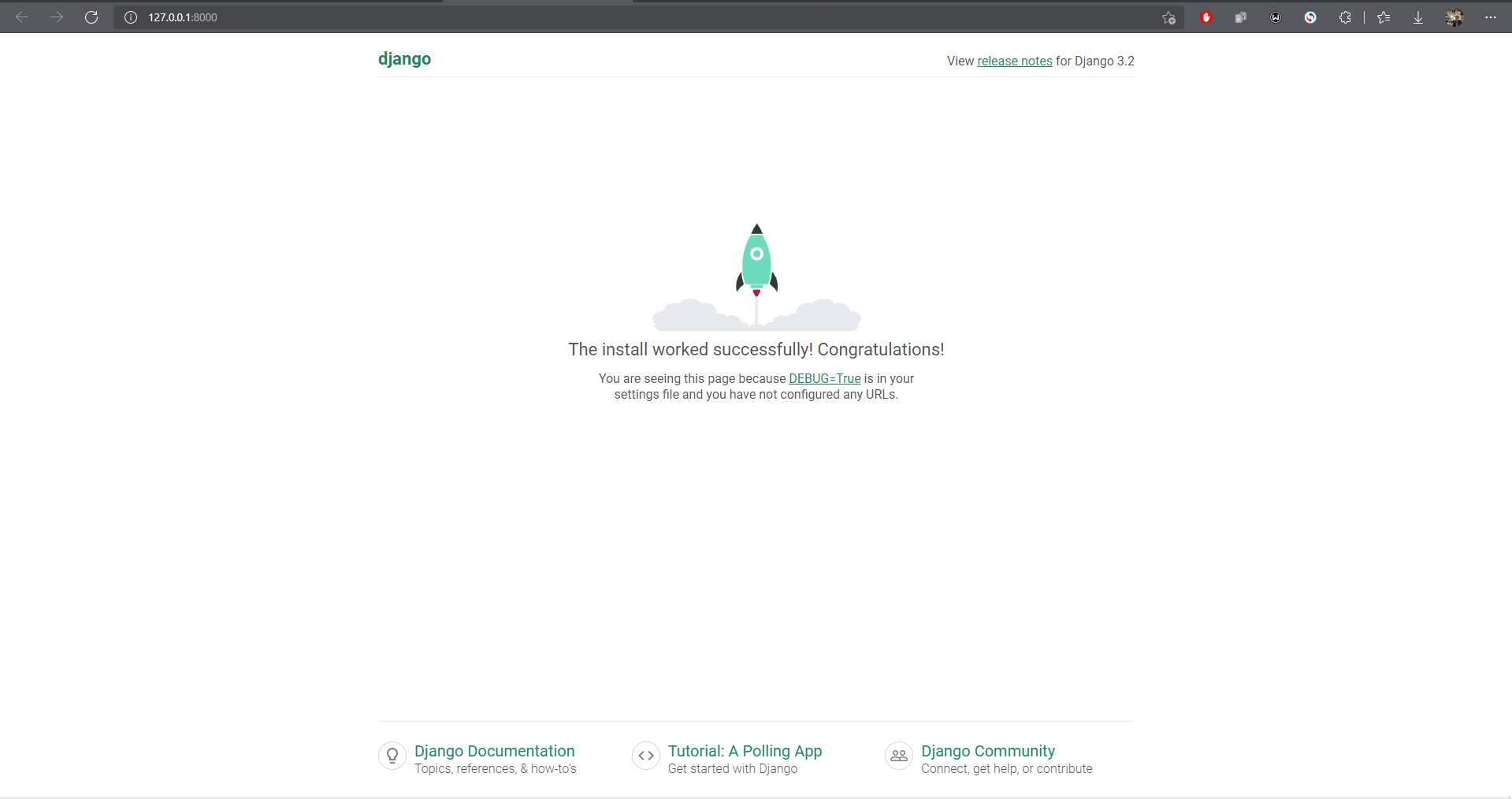
  
Рисунок 3 – Результат работы функции

1. **Задание №2. Начало работы в Django**

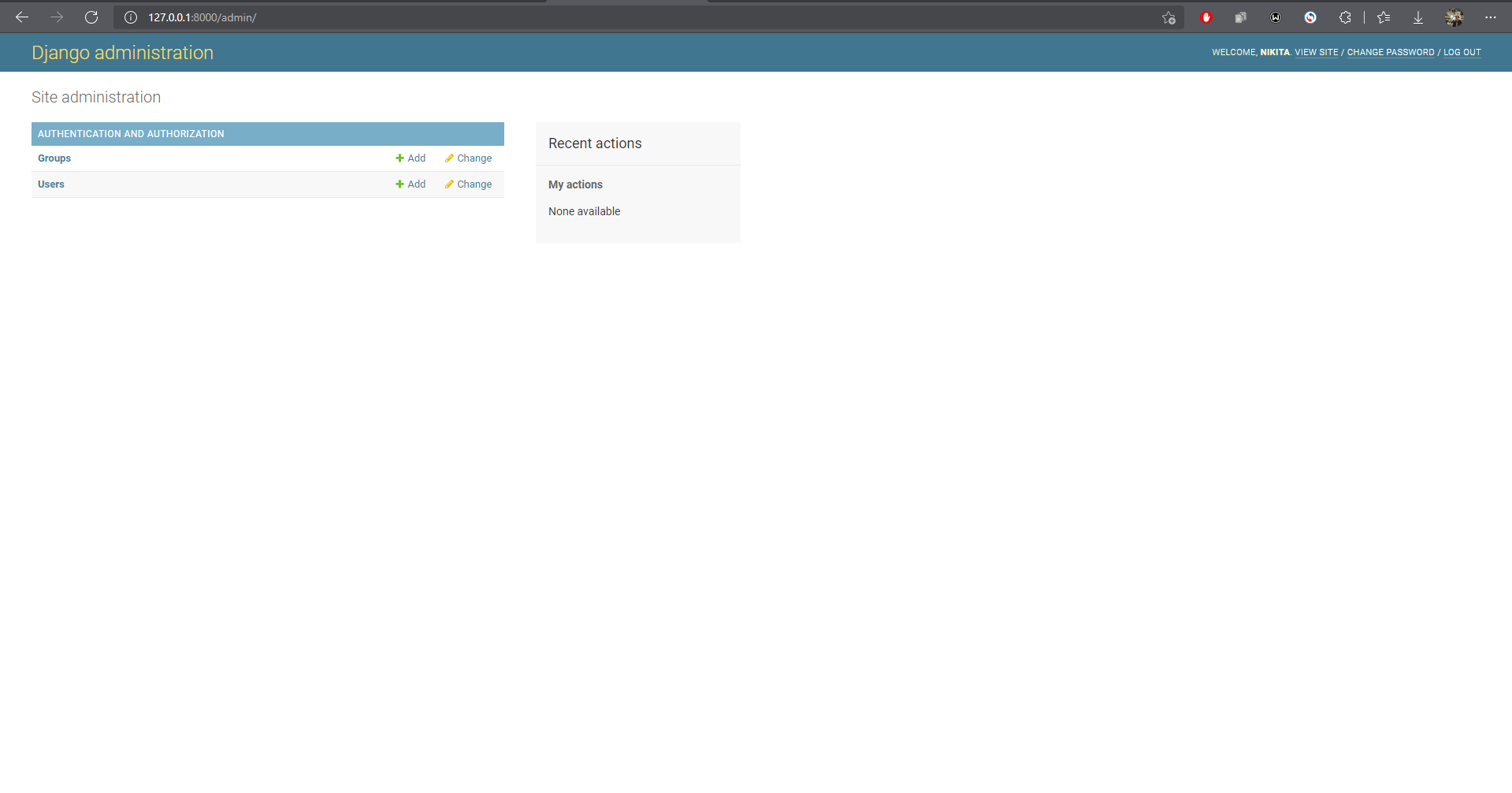
После установки и настройки фреймворка и создания проекта проверяем запуск сервера. На рисунке 4 показан успешный запуск сервера:

  
Рисунок 4 – Успешный запуск сервера

После перехода по адресу http://127.0.0.1:8000/ увидим, что установка сервера прошла успешно, как показано на рисунке 5:

  
Рисунок 5 – Подтверждение успешному запуску сервера

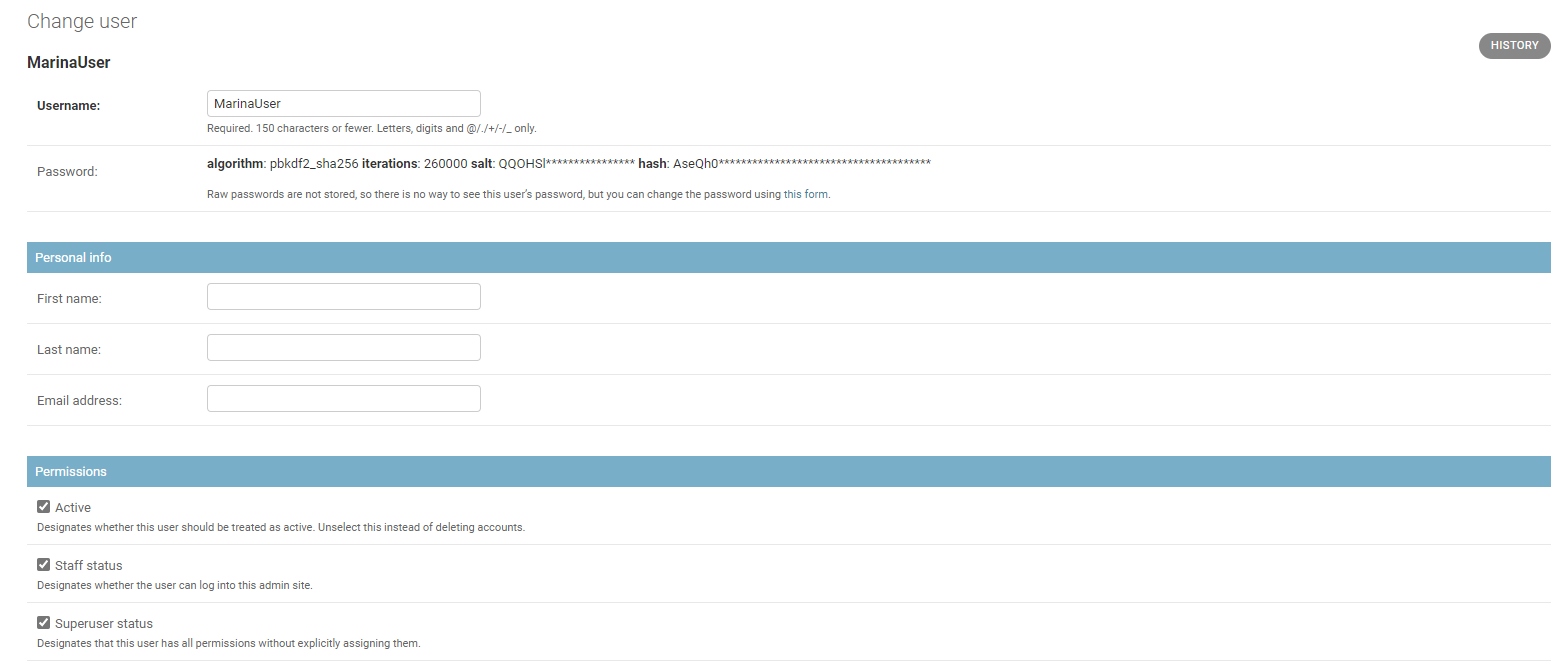
После создания базы данных, создания суперпользователя и перехода по адресу <http://127.0.0.1:8000/admin/>, появляется окно входа, куда необходимо ввести данные. После входа появляется главная страница административной панели, которая показана на рисунке 6:

  
Рисунок 6 – Главная панель администратора

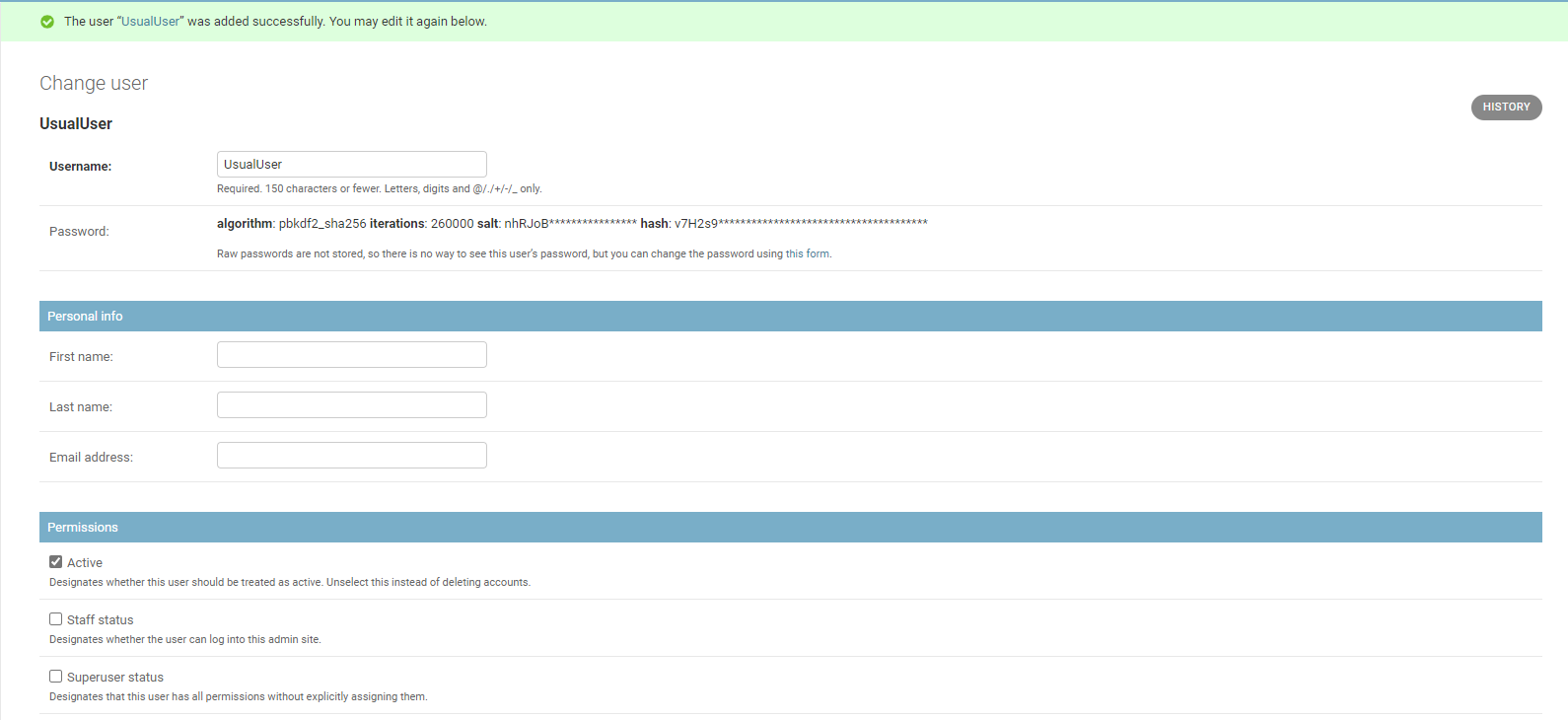
После выполнения шагов, описанных выше, необходимо выполнить следующие задания:

1. Изучите интерфейс административного приложения django;
2. Через интерфейс административного приложения создайте нового пользователя с правами суперпользователя;
3. Через интерфейс административного приложения создайте нового пользователя без прав суперпользователя;
4. Через интерфейс административного приложения «забаньте» одного из пользователей (сделайте пользователя «неактивным»);
5. Загрузите ваш проект на любой гит-репозиторий (GitHub, GitLab, Google Code, Bitbucket и т.п.).

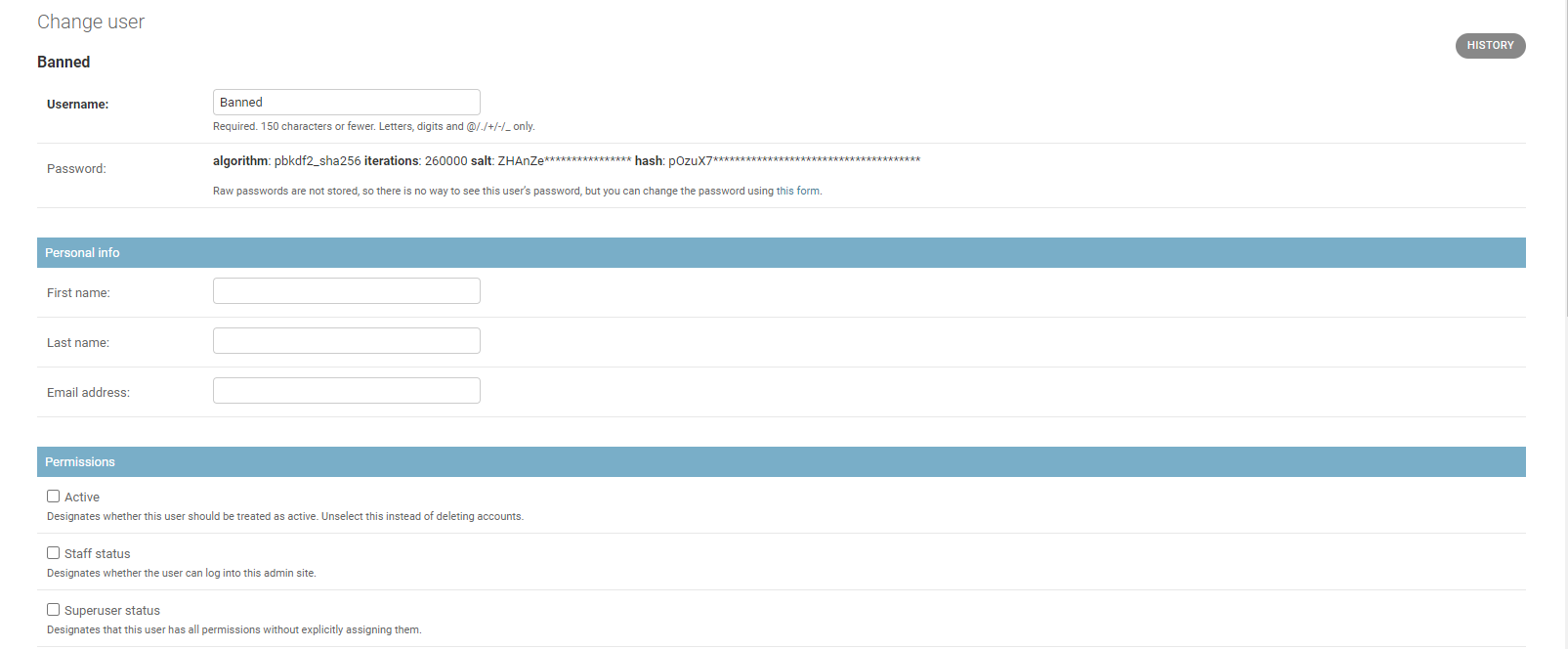
На рисунке 7 показано создание пользователя с правами суперпользователя:

  
Рисунок 7 – Создание суперпользователя

На рисунке 8 показано создание пользователя без прав суперпользователя:

Рисунок 8 - Созданный пользователь без прав суперпользователя

На рисунке 9 показан созданный пользователь, который был забанен, т.е. пункт «Active» был убран:

  
Рисунок 9 – Забаненный пользователь

# **Вывод**

В ходе лабораторной работы я научился работать с Python и Django.