Для запуска приложения требуется фреймворк Django и, в дополнение к нему, rest\_framework, а также БД postgresql.

Запуск осуществляется с помощью ввода в командную строку команды *manage.py runserver*. Необходимо будет также импортировать базу данных, приложенную к проекту на github. Пароль администратора бд (hotel\_admin): *hoteladmin*.

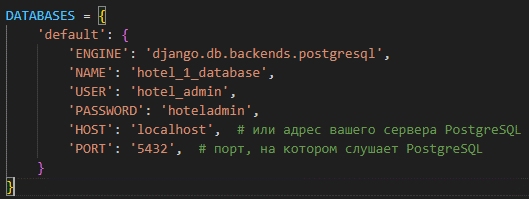


Рисунок 1 – подключение бд в проекте.

После запуска сервера для просмотра списка комнат можно перейти к пути rooms/

**Для сортировки и фильтрации используются url параметры, список доступных параметров:**  
min\_price (минимальная цена за комнату - float)

max\_price (максимальная цена за комнату - float)

min\_capacity (минимальная вместимость комнаты)

sort\_by (сортировка по возрастанию (asc) или по убыванию (desc) цены)

check\_in\_date (дата въезда в формате YYYY-MM-DD)

check\_out\_date (дата выезда в формате YYYY-MM-DD)

**Пример:**

http://127.0.0.1:8000/rooms/?min\_price=2500&max\_price=3300&min\_capacity=2&check\_in\_date=2023-10-30&check\_out\_date=2023-11-3&sort\_by=desc



Рисунок 2 - rooms/

Для перехода к бронированию нужно перейти к **reservations/** , однако пользователь должен быть заранее авторизован. Авторизация происходит в **login/**, для того чтобы успешно авторизоваться нужно сначала пройти регистрацию по url **register/** .

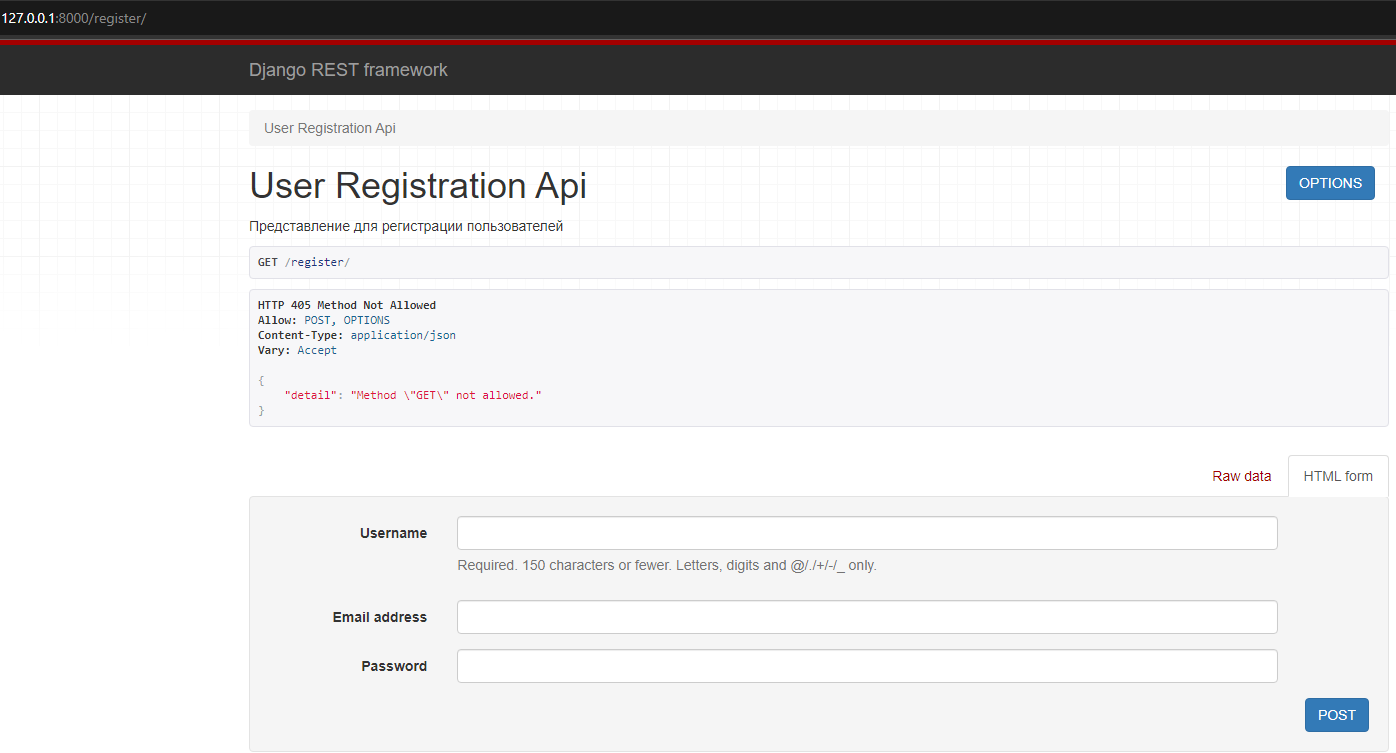


Рисунок 3 – register/

Для проверки был создан один пользователь с данными: login – SomeUser1 password – SomePassword1



Рисунок 4 – login/

После авторизации пользователя перекидывает к **accounts/profile/** , где он может посмотреть и отменить свои брони.

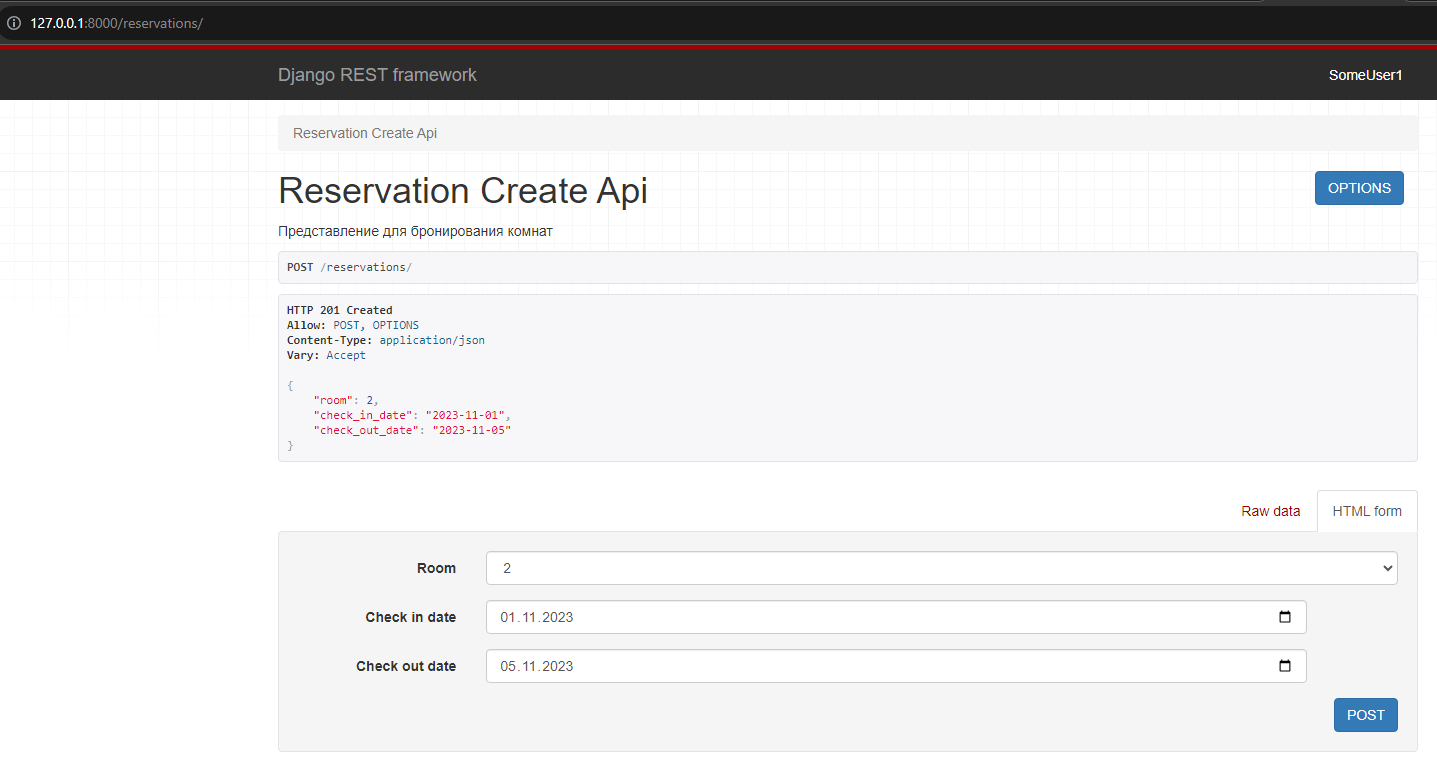


Рисунок 5 – reservation/

После создания брони в личном кабинете можно будет увидеть её и отменить.

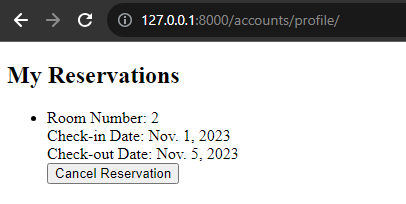


Рисунок 6 – accounts/profile/

В проекте также был создан суперпользователь с учётными данными: username – hotel\_administrator, password – hoteladmin. Суперпользователь может добавлять и удалять комнаты, пользователей, бронирования.

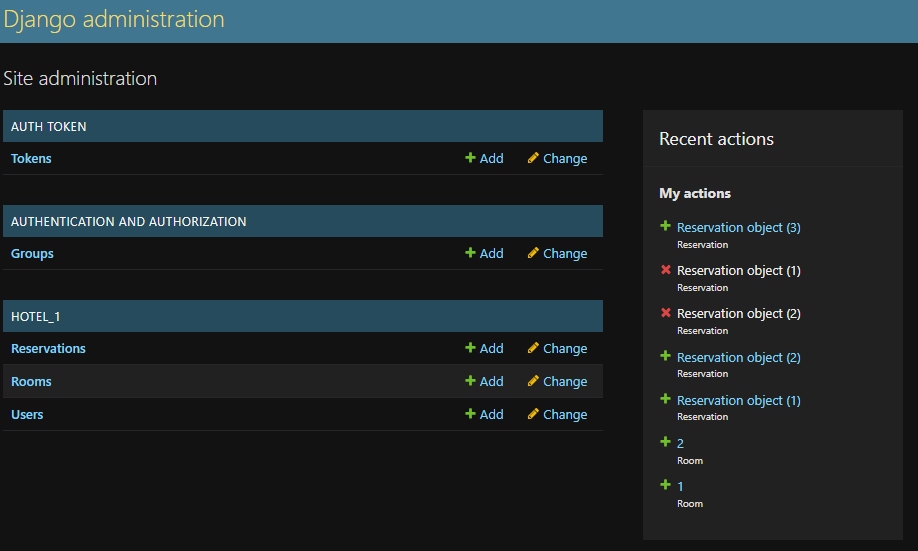


Рисунок 7 – админ-панель.