**Цель проекта:**

Создать API на основе Django и DRF для системы бронирования отелей, которая позволяет пользователям регистрироваться, просматривать отели и номера, оставлять отзывы, а также бронировать номера.

**Функционал API:**

1. **Пользователи:**

Регистрация и авторизация пользователей.

Разделение ролей:

Клиент (может бронировать и оставлять отзывы),

Владелец отеля (может добавлять отели и управлять номерами),

Администратор (полный доступ к данным).

1. **Отели и номера:**

Владелец отеля добавляет отели и номера.

У каждого отеля и номера могут быть разные статусы (номер может быть свободен, забронирован или занят).

Возможность прикрепления фотографий(1 или больше) для отелей и номеров.

1. **Бронирование:**

Клиенты могут бронировать номера (проверка статуса "свободен").

Отмена бронирования: статус номера меняется на "свободен".

Проверка уникальности бронирования для дат (нельзя забронировать номер на занятые даты).

1. **Отзывы:**

Клиенты могут оставлять отзывы об отелях (один отзыв на отель от одного пользователя).

Рейтинг отеля в диапазоне от 1 до 5.

Комментарии

**Проект разделён на две части для параллельной разработки.**

Разделение задач на двух разработчиков:

**Задачи для Разработчика 1 (Работа с пользователями и отелями):**

**Модели:**

User (пользователь с ролями).

Hotel (отель).

Room (номер).

Review (отзывы).

**Сериализаторы и эндпоинты:**

Регистрация и авторизация пользователей.

CRUD для отелей и номеров (только владельцы могут управлять).

Получение списка отелей и номеров для клиентов.

Получение отзывов для конкретного отеля.

**Права доступа (Permissions):**

Доступ к CRUD операциям для владельцев отелей.

Доступ к созданию отзывов только для клиентов.

**Задачи для Разработчика 2 (Бронирование и управление статусами):**

**Модели:**

Booking (бронирование).

Обновление статуса номера при создании и отмене бронирования.

**Сериализаторы и эндпоинты:**

Создание и отмена бронирования.

Проверка доступности номера перед бронированием.

Проверка уникальности дат при бронировании.

**Права доступа (Permissions):**

Бронирование доступно только для клиентов.

Владелец отеля не может бронировать свои номера.

**Требования к проекту:**

Python 3.x, Django 4.x, Django Rest Framework.

Использование PostgreSQL в качестве базы данных.

Подключение JWT аутентификации для пользователей.

Все данные (отели, номера, бронирования) должны быть доступны через REST API.

Код оформлен в соответствии с PEP8.

Использование GitHub:

Ветки для новых фич.

Комментарии и описание изменений в коммитах.

**Ожидаемый результат:**

API для пользователей, отелей, номеров и отзывов.

Возможность бронирования номеров и изменения их статуса.

Полноценная система ролей и прав доступа для разных типов пользователей.