**Тестовое задание Python**

Задача:

1. Необходимо оценить в часах и реализовать небольшое Django-приложение, которое решает задачу по управлению заказами на автомобили поставщика.
2. Цель - понять потенциальные возможности кандидата в части изучения и применения новой для него технологии.
3. Этап предварительной оценки - обязательный. Помимо качества и объема реализации требований будет приниматься во внимание факт соблюдения сроков с учетом предварительной оценки. Ожидается (но не обязательно), что оценка не превышает 40 часов при 100% загрузке на выполнение.

Основные требования:

1. Создать модель для справочника "Цвета".

2. Создать модель для справочника "Марки автомобилей"

3. Создать модель для справочника "Модели автомобилей"

4. Создать модель для хранения заказов авто. Заказ должен включать в себя номер заказа, цвет, модель, количество, дату (по умолчанию текущая).

Допущения:

1. Поставщик в состоянии поставить любое количество авто любой марки/модели любого цвета– через админку реализовано.

Ограничения:

1. Все поля всех моделей обязательны к заполнению, кроме даты заказа. В случае не передачи даты заказа в запросе заказ создается с текущей датой.

Доп. требования:

1. Реализовать поддержку постраничного вывода списка заказов (объем страницы 10 элементов), реализовать сортировку списка заказов по количеству, реализовать фильтрацию списка заказов по марке авто

Нужно оптимизировать SQL запрос

5. С использованием библиотеки Django Rest Framework создать RestAPI для управления справочниками и заказами. API должно реализовать операции CRUD для моделей, а также чтение списков.

API для списка заказов должен возвращать элементы со след. атрибутами: дата заказа, цвет, марка авто, модель авто, количество.

Заказ должен включать в себя , цвет, модель, количество, дату (по умолчанию текущая). – поэтому POST и PUT в API только эти поля

Доп. требования:

2. Обеспечить пользовательское представление API в формате OpenApi (Swagger).

3.Реализовать API для получения след. информации:

3.1. список цветов с указанием количества заказанных авто каждого цвета (атрибуты элементов: цвет, количество),

3.2. список марок с указанием количества заказанных авто каждой марки (атрибуты элементов: марка, количество)

Окружение.

1. Python >3.9

2. Django >3.2

3. Postgres >13

4. Идеальный вариант: окружение должно подниматься в Docker

Предполагаемая модель данных

