Тестовое задание

Стек обязательный к использованию:

- VM: Doker (желательно добавить CI/CD)

- PL: Python

- Packaging/Dependency management: Poetry

- Freamework: FastApi

- DB: Postgresql

Замметки:

- PK - primary key

- PR - private repository

1. прочитайте ТЗ, задайте вопросы при необходимости

2. выполните задание

3. желательно написать тесты на 3-4 вида апишек

4. **очень** желательно написать скрипт, который будет создавать 200-500 юзеров разных ролей, 200-500 заказов и 100-150 посещений

5. разместите код в PR в GitHub

6. пришлите:

a. ссылку на репозиторий

b. архитектуру бд (если таковая будет)

7. нам понадобится около недели на проведение ревью и сообщение результатов 8. если какие-то условия/требования не выполнены, но мы в вас заинтересованы, то можем попросить доработать код

ТЗ

Создать простое API для мобильного приложения, в котором полевой сотрудник заказчика будет выполнять визиты в магазины.

Создать сущности

1. "Работник", с полями:

○ имя – char 255

○ номер телефона – char 255 (username)

○ торговая точка – ForeignKey к "Торговая точка" (обязательно к заполнению)

2. "Заказчик", с полями:

○ имя – char 255

○ номер телефона – char 255

○ торговая точка – ForeignKey к "Торговая точка" (обязательно к

заполнению)

3. "Торговая точка"

○ название

○ сотрудники – ForeignKey к "Работник" (обязательно к заполнению) 4. "Заказ"

○ дата/время – создания

○ дата/время – окончания – после которой, заказ не актуален

○ куда – ForeignKey к "Торговая точка"

○ автор – ForeignKey к "Заказчик"

○ статус – char [‘started’, ‘ended’, ‘in process’, ‘awaiting’, ‘canceled’] ○ исполнитель – ForeignKey к "Работник"

5. "Посещение"

○ дата/время – создания

○ исполнитель – ForeignKey к "Работник"

○ заказ – OneToOne к "Заказ"

○ автор – ForeignKey к "Заказчик"

○ куда – ForeignKey к "Торговая точка"

Сделать методы

Для упрощения задания, вместо полноценной авторизации передавать в каждый запрос номер телефона Работника.

1. получить список Торговых точек привязанных к переданному номеру телефона ○ в ответе список Торговых точек (желательно с фильтрацией):

■ PK

■ Название

2. выполнить CRUD по сущности “Заказ” (автором может быть, только “Заказчик”) ○ сделать валидацию: “Заказчик” может создать заказ только на свою “торговую точку”

○ сделать валидацию: Исполнителем должен быть Работник привязанным к “Торговой точке”

○ желательно добавить фильтрацию списка по GET запросу

3. выполнить CRUD по сущности “Посещение” (автором может быть, только “Заказчик”)

○ сделать валидацию: Заказчик может создать “Посещение” только на свою “торговую точку” и если срок “Заказа” не истек

○ сделать валидацию: “Посещение” может создаваться, если ранее уже не было иной записи по PK “Заказа”

○ сделать валидацию: Исполнителем должен быть Работник привязанным к Заказу

○ желательно добавить фильтрацию списка

4. сделать отдельный PUT метод запроса для изменения статуса “Заказа”

Админка

● Работник, Заказчик, Торговая точка, Заказ, Посещение

○ создание

○ редактирование

○ удаление

○ поиск по имени Работника

○ поиск по номеру Работника

○ поиск по названию Торговой точки