Техническое задание

Часть 1 Backend JWT:

1) Необходимо создать простое Django (5 версия) приложение в связке с Django REST framework (DRF).
   https://www.django-rest-framework.org/tutorial/quickstart/#project-setup
2) Установить интеграцию с JWT (*simplejwt*). Добавить ендпойнты **/auth/token/**, **/auth/token/refresh/** для выдачи и обновления токена .
   https://django-rest-framework-simplejwt.readthedocs.io/en/latest/getting_started.html
3) Создать супер пользователя **medadmin** и несколько пользователей **user1**, ….
4) Сохранить скриншоты ответов через **Postman**, либо **Browsable API**.

Часть 2 Backend logic:

1) Создать новое приложение **medclinics** используя *django-admin startapp.*
2) Создать несколько модельки Appeal, Service, Diagnosis (Таблица 1) и установите приложение/миграции.
   Appeal: id (PK), created_at, patient_full_name (varchar), service_id (FK), diagnosis_id (FK) and reason (varchar).
   Diagnosis: id (PK), title (varchar), description (text), symptoms (text)
   Service: id (PK), title (varchar), description (text), price (number), category (varchar)
3) Добавить новую команду **populate_db** в *manage.py* для заполнения данных. Можно использовать **faker** для генерации имен, описании, названии и т.д.
4) Настроить сериализаторы **ModelViewSet** / **ModelSerializer** для моделей, зарегистрировать ендпойнты и добавить permission-ы для CRUD операции.
   https://www.django-rest-framework.org/tutorial/4-authentication-and-permissions/
5) Добавить кастомные модели в **Django Admin** панель.
6) Покрыть unit и integration тестами. **BONUS**: использовать coverage и линтеры.
7) Сохранить скриншоты админки с моделями, скриншоты ответов через **Postman/Browsable API** для каждого HTTP метода с JWT авторизацией.

Часть 3 Docker & Packaging:

1) Написать легкий Dockerfile для приложения
   https://betterstack.com/community/guides/scaling-python/dockerize-django/#step-4-creating-a-dockerfile-for-your-project
2) Написать docker-compose.yml файл с сервисами PostgreSQL и нашего Django приложения.
   https://betterstack.com/community/guides/scaling-python/dockerize-django/#step-10-deploying-the-django-application-with-docker-compose

Часть 4 Frontend:

1) Настроить Dash Plotly
   https://dash.plotly.com/dash-enterprise/application-structure/django-app
2) Создать страницу с визуализацией данных используя Dash.
3) Сохранить скриншот страницы

По завершению задания дать доступ к репозиторию на **GitHub** @yesseyev**.**