Тестовое задание

Разработка API для работы с достижениями

Вам понадобятся:
- любая опенсорсная СУБД(предпочтительно - PostgreSQL) и опыт работы с ней;
- знание python;
- знание flask/fastapi;

Опционально, но будет плюсом:
- знание docker;
- знание nginx;
- знание nodejs;
- умение писать документацию к проекту.

Основная задача:

1. реализовать на python3 простой сервер (с помощью fastapi или flask) для общения с базой данных;
2. выложить результат работы на github/gitlab;
3. в ответном письме прислать ссылку на выполненную работу.

База данных должна иметь таблицы:
1. таблица пользователей;
   - необходимые поля:
     1. имя пользователя,
     2. выбранный пользователем язык (ru/en).
2. таблица достижений;
   - необходимые поля:
     1. имя достижения;
     2. количество баллов, ассоциирующихся с достижением (некоторое абстрактное число);
     3. описание.

Сервер должен выполнять следующие функции:
- предоставлять информацию о пользователе;
- предоставлять информацию о всех доступных достижениях;
- добавлять достижения;
- выдавать достижения пользователю с сохранением времени выдачи (сохранять связь пользователя с достижением и датой выдачи);
- предоставлять информацию о выданных пользователю достижениях на выбранном пользователем языке;
- предоставлять статистические данные системы:
  - пользователь с максимальным количеством достижений;
  - пользователь с максимальным количеством очков достижений;
  - пользователи с максимальной разностью очков достижений;
  - пользователи с минимальной разностью очков достижений;
  - пользователи, у которых достижения выдавались 7 дней подряд (есть достижения с датами выдачи соответсвующими).

Опциональные задачи:

1. составить docker compose файл для развертывания сервера и всех его компонентов(к примеру, СУБД);
2. одним из компонентов развертывания должен быть сервер nginx через который организует reverse proxy доступ к серверу;
3. сделать front-end для работы с сервером (интерфейс к одной или нескольким функциям, например для добавления достижения или вывода достижений пользователя по его имени);
4. показать умение работать с историей git;
5. показать навыки оформления проекта (комментарии к функциям, логирование, написание read.me);
6. показать умение составления документации (например, оформить описание внешних интерфейсов сервера);
7. проявить фантазию в именовании достижений и их описании;
8. показать умение тестировать разработанный код.