

Время выполнения задания оценивается в 4-8 человеко-часов (для специалиста уровня Junior).

Максимальный срок выполнения - до 23.10.2023.

Результат выполнения задания должен быть выложен соискателем в публичный репозиторий github и помимо кода проекта содержать подробные инструкции по сборке и запуску. Ссылку на проект необходимо направить на почту: hr@bewise.ai в тексте письма указать ваше ФИО и ссылку на ваше резюме

Задачи:

1. С помощью Docker (предпочтительно - docker-compose) развернуть образ с любой опенсорсной СУБД (предпочтительно - PostgreSQL). Предоставить все необходимые скрипты и конфигурационные (docker/compose) файлы для развертывания СУБД, а также инструкции для подключения к ней. Необходимо обеспечить сохранность данных при рестарте контейнера (то есть - использовать volume-ы для хранения файлов СУБД на хост-машине).
2. Реализовать на Python3 простой веб сервис (с помощью FastAPI или Flask, например), выполняющий следующие функции:
 - В сервисе должно быть реализовано REST API, принимающее на вход POST запросы с содержимым вида {"questions_num": integer} ;
 - После получения запроса сервис, в свою очередь, запрашивает с публичного API (англоязычные вопросы для викторин) <https://jservice.io/api/random?count=1> указанное в полученном запросе количество вопросов.
 - Далее, полученные ответы должны сохраняться в базе данных из п. 1, причем сохранена должна быть как минимум следующая информация (название колонок и типы данных можете выбрать сами, также можете добавлять свои колонки): 1. ID вопроса, 2. Текст вопроса, 3. Текст ответа, 4. - Дата создания вопроса. В случае, если в БД имеется такой же вопрос, к публичному API с викторинами должны выполняться дополнительные запросы до тех пор, пока не будет получен уникальный вопрос для викторины.
 - Ответом на запрос из п.2.а должен быть предыдущий сохранённый вопрос для викторины. В случае его отсутствия - пустой объект.
3. В репозитории с заданием должны быть предоставлены инструкции по сборке докер-образа с сервисом из п. 2., его настройке и запуску. А также пример запроса к POST API сервиса.
4. Желательно, если при выполнении задания вы будете использовать docker-compose, SQLAlchemy, пользоваться аннотацией типов.