**Привет! Мы в LogicLike создаем высоконагруженную EdTech-платформу, которая превращает развитие логики и мышления в увлекательную игру для миллионов детей по всему миру. Мы подготовили для вас небольшое тестовое задание, которое по нашей оценке, займет около 3-х часов.**

## **Задание:**

Мы хотим запустить публичную страницу, где любой желающий может посмотреть список идей для развития нашего продукта и проголосовать за самые понравившиеся. Чтобы избежать накруток, мы хотим ограничить голосование: **с одного IP-адреса можно отдать свой голос не более чем за 10 разных идей.**

**Бэкенд (**[**Node.js**](http://node.js) **+ Typescript + Express/Koa/Fastify)**

1. **Логика голосования:**
   * Сервер должен определить IP-адрес клиента. Учтите, что сервис может работать за reverse-proxy (например, Nginx), поэтому нужно корректно обрабатывать заголовок X-Forwarded-For.
   * Перед тем как засчитать голос, система должна проверить, не превысил ли данный IP-адрес лимит в 10 голосов.
   * Если лимит превышен или голос за эту идею уже был, сервер должен вернуть соответствующую ошибку 409 Conflict.
2. **База данных:**
   * Используйте PostgreSQL или MySQL.
   * Должен быть предоставлен seed-скрипт для наполнения БД.

**Фронтенд (React)**

1. **Отображение:**
   * При загрузке страница запрашивает и отображает список идей, отсортированный по количеству голосов.
   * У каждой идеи есть кнопка "Проголосовать", если с этого IP адреса за идею ещё не голосовали.
2. **Интерактивность:**
   * При нажатии на кнопку "Проголосовать" должен засчитываться голос за эту идею, если пользователь не исчерпал свой лимит голосов.

Код должен быть размещен в репозитории на одной из платформ: github/gitlab/bitbucket

**Что мы будем оценивать (что для нас важно):**

1. Архитектура бэкенда и работа с БД
2. Качество кода и структура проекта
3. Производительность кода
4. Управление состоянием на фронтенде(обработка ошибок, индикаторы загрузки…)
5. Простота и полнота инструкций для разворачивания проекта

**На что мы не будем смотреть (что не важно)**

1. Дизайн и стилизация. Достаточно чтобы интерфейс был понятен. Можно использовать любой UI/CSS фреймворк: Tailwind, bootstrap, bulma, RadixUI, shadcn…
2. Сложный стейт-менеджмент на фронтенде: Не нужно подключать Redux, MobX или другие библиотеки для управления стейтом, если в этом нет острой необходимости. Стандартных хуков React (useState, useEffect) будет более чем достаточно.
3. Тесты: Мы не ожидаем, что вы будете писать юнит- или e2e-тесты для тестового задания.
4. Кросс-браузерная совместимость: Достаточно, чтобы приложение работало в последней версии Google Chrome.

**Успехов!**

# Организационные вопросы

**Дедлайн:** 5 рабочих дней после получения ТЗ.

**Куда присылать ТЗ:** ТЗ прикрепи в этой [форме](https://docs.google.com/forms/d/e/1FAIpQLScmMBA91IMmkgrMLK_9wEIGoPnSSv0ASpAbLerHpJ0FHygavQ/viewform?usp=header)

**Обратная связь по заданию:**

С обратной связью мы вернемся в течение 7-ми рабочих дней после получения реализации.

**Приглашение на интервью:**

Кандидатов, которые успешно справятся с ТЗ, мы пригласим на техническое интервью к СТО.