# **Тестовое задание на вакансию “Frontend Developer”**

## О нас:

Dapplets Project - cтартап-криптоплатформа для приложений Augmented Web. Мы переносим идеи децентрализации и устойчивости на уровень визуального UI/UX традиционных сайтов.

Мы строим платформу Аугментированного веба, состоящую из браузерного плагина и децентрализованных приложений (дапплетов), основанных на крипто-технологиях.

Наша платформа создается в виде open-source framework для разработчиков со всего мира.

## В ваши задачи будет входить:

* разработка клиентской части веб-приложений с использованием React;
* вёрстка макетов из Figma;
* написание и отладка TypeScript/JavaScript кода, рефакторинг;
* взаимодействие с остальными участниками проекта (UI/UX, backend).

## Оплата тестового задания

В случае вашего трудоустройства, выполнение тестового задания будет оплачено из расчета 3-х рабочих дней.

# Тестовое задание:

Вы можете сделать тестовое задание частично.

1. Сверстать страничку по макету Figma (без прелоадера) с использованием фреймворка React и языка TypeScript:

Макет:   
<https://www.figma.com/file/YGXq1mzXTTc4Rfwt5vxgUo>

Презентация макета: <https://www.figma.com/proto/YGXq1mzXTTc4Rfwt5vxgUo>

* 1. Адаптивная вёрстка для desktop и mobile устройств
  2. Fuzzy and Live Search (живой фильтр)
     + debounce c окончательной фильтрацией на клиенте.
     + Сервер может вернуть “неподходящие” элементы, которые не попадают в фильтр. Их надо будет выкинуть и запросить еще пакет данных взамен выкинутых.
  3. Постраничный вывод (по мере скроллинга)
  4. Обработка серверных ошибок (вывод в область сообщений)
  5. Обработка ошибок скачивания картинок
  6. Состояние интерфейса сохранить в LocalStorage (installed/not installed)
  7. Drag & drop для сортировки элементов списка

1. Привязать бэкенд <https://dapplets-hiring-api.herokuapp.com/>

## Результат тестового задания

Результатом тестового задания будет:

1. Пояснительная записка, какие пункты из ТЗ сделаны;
2. Ссылка на опубликованный на Github проект;
3. Ссылка на Live Demo проекта на любом хостинге (например, Netlify).

Готовое тестовое задание присылайте в HeadHunter.

## Структура и функционал макета в Figma:

(другие элементы макета имплементировать не нужно)

1. Строка поиска + кнопка для направления сортировки
2. Строка сообщений состояния и ошибок (в самом верху).
3. Список плашек элементов спиcка (Items)
   1. UserGenerated Icon
   2. Текст названия (1 строчка)
   3. Текст описания (мультилайн)
   4. Статичный тэги (несколько штук)
   5. Гибридная кнопка (статус/действие) с 2мя состояниями и rollover.
      1. Если приложение не установлено, то отображается "INSTALL".
      2. Если приложение установлено, то отображается "INSTALLED" и при наведении курсора мыши надпись должна поменяться на "UNINSTALL". Этот статус сохраняется в localstorage, чтобы после перезагрузки страницы он не сбросился.

Никаких приложений, конечно же, устанавливать не надо. Это только фронтенд.

* 1. По клику на элемент списка, открывается Аккордеон (открытие плашки приложения).

1. Драг и Дроп элементов списка (ручная пересортировка). В макете есть “область захвата”.
2. Постраничная загрузка по мере скроллинга
   1. Loader с анимацией
3. Адаптивность под Mobile и Desktop
   1. Перемещение блоков на разных устройствах
   2. Скрытие лишнего функционала на мобильных устройствах

# Свойства Бэкенда

1. backend с HTTP API
   1. Возвращает ссылки на картинки, расположенные на другом сервере
      1. Картинки могут быть недоступными
      2. Могут быть тяжёлыми
      3. Сервер с картинками может долго отвечать
   2. Возвращает список приложений
      1. Серверная фильтрация\*
      2. Серверная сортировка\*
      3. Постраничная загрузка\*
   3. Возвращает список тегов
   4. Может вернуть слишком длинный title или description
   5. Может долго отвечать
   6. Может возвращать ошибки
   7. Может отвалиться и вернуть невалидный ответ
2. Спецификация на backend <https://dapplets-hiring-api.herokuapp.com/>