

Тестовое задание от 02.2020 на позицию фронтенд разработчика

(!) Общая информация

1. Для разработки используйте на выбор: React, Vue или Angular.
2. Высылайте выполненное тестовое задание на электронный адрес **hr@abz.agency**:
 - a. Высылайте решение тестового задания в виде ссылки на github/bitbucket репозиторий.
 - b. Разверните Вашу работу на любом хостинге (чтобы ее можно было посмотреть в браузере не разворачивая проект из исходного кода) и вышлите нам ссылку.
 - c. Укажите в письме список выполненных пунктов и выбранные вами инструменты. Если Вы сделали не все пункты тестового задания – пожалуйста, укажите причину по которой Вы их не выполнили (не хватило времени, не хватает опыта/знания, что-то ещё).
 - d. Обязательно приложите к письму максимально подробное и актуальное на дату отправки резюме.
 - e. Укажите сколько часов было потрачено на выполнение задания.
3. Мы рекомендуем Вам использовать расширение [“Pixel Perfect”](#) в Google Chrome. Мы будем использовать это расширение при проверке выполненного тестового задания.
4. Все материалы, используемые в данном тестовом задании **было специально разработаны** сотрудниками компании abz.agency для проверки знаний соискателей в рамках тестового задания для данной вакансии (прототипы, макеты, API, бэк-энд) и не имеют ничего общего с нашими коммерческими проектами. Ваши работы не будут использованы в коммерческих и (даже) некоммерческих целях. Вы также **не имеете права** использовать текст и материалы данного тестового задания, а также Ваше решение данного задания в коммерческих и некоммерческих целях (в том числе в портфолио, для составления курсов или в качестве тестового задания в другой компании).

Важно! Тестовое задание будет принято к рассмотрению только в случае если оно соблюдает требования, перечисленные выше.

(Т) Технические требования

- React / Vue / Angular (version 6+);
- HTML5 / CSS3;
- CSS препроцессор (Sass/Less) или PostCSS;
- Разметка должна быть pixel-perfect (desktop: последняя версия Chrome / FF / Safari / Edge, mobile/tablet: последняя версия Safari на iOS 12+ и Chrome на Android 7+);
- Минимальная ширина экрана 360px;
- Максимальная ширина экрана 2560px;
- Максимальная ширина основного контейнера 1170px;
- Разметка должна быть готова к переполнению содержимого, включая длину заголовка блока. Если текст длиннее чем может отображать блок - Вы должны его обрезать и показать знак «...» с помощью CSS или Javascript.

Важно! Мы будем обращать внимание, насколько чистый у Вас подход к написанию CSS и Javascript-кода. Вы можете использовать любые сторонние библиотеки CSS и Javascript без каких-либо ограничений. Сторонние библиотеки CSS / Javascript должны быть подключены через менеджеры зависимостей npm / yarn (а не методом копи-паста). Сборка проекта должна быть реализована с помощью одного из популярных на данный момент task runner'ов npm / gulp / webpack (а не с помощью bash скриптов или Makefile).

(<) Задание

1. Вам необходимо реализовать responsive верстку HTML5/CSS3 в соответствии с макетами (mockups) <http://view.maquetter.com/rkymlp/0201> - 0205 и стайл гайд (style guide) <http://view.maquetter.com/rkymlp/0101>. Все необходимые исходники Вы сможете [найти тут](#).
2. Все СТА (call to action) ссылки/кнопки должны вести на регистрацию. Обращаю Ваше внимание на тот факт, что хэдер должен быть закреплен сверху страницы при скролле. Также обратите внимание что:
 - a. Все изображения на сайте должны поддерживать Retina экраны (для моб версии).
 - b. Используйте CSS препроцессор (Sass/Less) или PostCSS.
 - c. Организуйте Ваши CSS стили в читаемом виде (группируйте CSS по функциональности, пишите разъясняющие комментарии, разделяйте стили на несколько файлов по функциональности и т.д.).

3. Работа с REST API (GET). Тут Вы найдете [API документацию](#).
 - a. Используя полученные данные юзеров реализуйте вывод блока “Our cheerful users” в соответствии с макетом. Кнопка “Show more” должна подгружать пользователей из API согласно макетам. Кнопка “Show more” должна быть скрыта, когда больше нет пользователей в БД (достигнута последняя страница результатов в API). Пользователи отсортированы по дате регистрации (новые первые).
 - b. Обратите внимание на mockups и style guide при работе с блоком юзеров. Задание предполагает ваше внимание к деталями вывода контента. Такие детали как количество блоков, “...”, tooltips, перенос строк и т.п.
 - c. Для отображения радиобаттонов на форме регистрации используйте GET /positions метод из API документации.
 4. Работа с REST API (POST) – форма регистрации
 - a. Реализовать валидацию на фронт-енд части в соответствии с макетами и API документацией.
 - b. Реализовать бизнес логику формы регистрации в соответствии с макетами и API документацией.
 - c. После успешной регистрации обновить список пользователей в блоке “Our cheerful users”. Если кнопка “Show more” была нажата (т.е. больше чем одна страница пользователей была загружена из API), то свернуть все и отобразить только первых 6 пользователей. В результате новый пользователь будет выведен первым и Вы сможете проверить корректность работы пункта 5b без перезагрузки страницы.
- Важно!** Каждую ночь список пользователей будет автоматически сброшен (на начальные 45 пользователей).
5. Самостоятельное тестирование в следующих браузерах (можно использовать BrowserStack):
 - a. Chrome, Firefox, Edge, Safari (Windows)
 - b. Chrome, Firefox, Safari (MacOS)
 - c. Chrome, Safari (iOS)
 - d. Chrome (Android)
 6. Оптимизация вебсайта (бонусное задание). Минимизируйте и оптимизируйте css, js, изображения, и т.д. Для этого Вам необходимо развернуть Вашу работу на любом доступном Вам хостинге и направить на него любой доступный Вам домен.
 - a. Проверьте Вашу работу используя Google Page Speed и убедитесь что Ваша работа в зеленой зоне.

- b. Проверьте Вашу работу используя Google Chrome Audit и убедитесь что Ваша работа в зеленой зоне для Performance, Best practices, SEO (mobile and desktop для 3G).
- c. Проверьте Вашу работу используя Webpagetest и убедитесь что показатели близки к AAAAAA.