

Тестовое задание на позицию фронтенд разработчика

В этом тестовом задании Ваша задача разработать фронтенд часть одностраничного респонсив сайта, дизайн которого мы специально разработали для данного задания.

(!) Общая информация

1. Для разработки используйте **только** один из следующих фреймворков: Angular (version 6+) или React. Если один из этих фреймворков не будет использован – Ваша работа не будет принята к рассмотрению.
2. Требования для выполненного тестового задания:
 - a. Высылайте решение тестового задания на электронный адрес **hr@abz.agency** в виде ссылки на github/bitbucket репозиторий.
 - b. Также 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 выбранный Вами фреймворк. 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).

Важно! Тестовое задание будет принято к рассмотрению только в случае корректного оформления.

3. Мы рекомендуем Вам использовать расширение [“Pixel Perfect”](#) в Google Chrome. Мы будем использовать это расширение при проверке выполненного тестового задания.
4. Данное тестовое задание было специально разработано в качестве тестового задания для данной вакансии (прототипы, макеты, API, бэк-енд) и не имеет ничего общего с нашими коммерческими проектами. Ваши работы не будут использованы в коммерческих и (даже) некоммерческих целях.

Технические требования

- Angular (version 6+) / React;
- HTML5 / CSS3;
- Разметка должна быть pixel-perfect (десктоп: последняя версия Chrome / FF / Safari / Edge, мобильная версия: последняя версия Safari на iOS 11/12 и Chrome на Android 6/7/8);
- Минимальная ширина экрана 320px;
- Максимальная ширина экрана 2560px;
- Максимальная ширина контейнера 1440px;
- 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. Если в проект добавляются сторонние библиотеки css / javascript через менеджеры зависимостей npm / yarn (а не методом копи-паста), Вы получите бонусные баллы. Если Вы используете любой task runner (gulp / webpack), Вы также получите бонусные баллы.

Пункты задания

1. Вам необходимо реализовать респонзив (responsive) верстку HTML5/CSS3 в соответствии с макетами (mockups) <http://view.maquetter.com/ekwgbx/0201> - 0208 и стайл гайдом (style guide) <http://view.maquetter.com/ekwgbx/0101>. Все необходимые исходники Вы сможете [найти тут](#). При желании, Вы можете ознакомиться с прототипами ([wireframes](#)).

Все СТА (call to action) должны вести на регистрацию. Ссылки в хэдере должны вести на соответствующие блоки на странице. Все остальные ссылки на странице должны вести на "#". Обращаю Ваше внимание на тот факт, что хэдер

должен быть закреплен вверху страницы при скролле. Также обратите внимание что:

- 1.1. Все изображения на сайте должны поддерживать Retina экраны.
 - 1.2. Используйте CSS препроцессор (Sass/Less) или PostCSS.
 - 1.3. Организуйте Ваши CSS стили в читаемом виде (группируйте CSS по функциональности, пишите разъясняющие комментарии, разделяйте стили на несколько файлов по функциональности и используйте импорты чтобы соединить их в один файл).
2. Работа с REST API (GET). Тут Вы найдете [API документацию](#). Для проверки что все методы работают на стороне сервера можете использовать эту [ссылку](#).
- 2.1. Используйте параметр user_id=1 чтобы вывести пользователя в хэedere. Авторизации на сайте нет, Вам просто необходимо вывести полученные данные.
 - 2.2. Используя полученные данные юзеров реализуйте вывод блока "Our cheerful users" в соответствии с макетом. Кнопка "Show more" должна подгружать ещё 6 пользователей. Кнопка "Show more" должна быть скрыта, когда больше нет пользователей в БД (достигнута последняя страница результатов в АПИ). Пользователи отсортированы по дате регистрации (новые первые).
 - 2.3. Для отображения выпадающего списка в форме регистрации используйте соответствующий метод из API документации.
3. Работа с REST API (POST) – форма регистрации
- 3.1. Реализовать валидацию на фронт-енд части в соответствии с макетами и API документацией.
 - 3.2. Реализовать бизнес логику формы регистрации в соответствии с макетами и API документацией.

Важно! Каждую ночь список пользователей будет автоматически сброшен (на начальные 45 пользователей).

- 3.3. После успешной регистрации обновить список пользователей в блоке “Our cheerful users”. Если кнопка “Show more” была нажата, то свернуть все и отобразить только первых 6 пользователей. В результате новый пользователь будет выведен первым и Вы сможете проверить корректность работы пункта 3.2 не перезагружая страницу.
4. Самостоятельное тестирование (можно использовать BrowserStack)
 - 4.1. Проверьте Вашу работу в Chrome, Firefox, Edge, Safari (Windows)
 - 4.2. Проверьте Вашу работу в Chrome, Firefox, Safari (MacOS)
 - 4.3. Проверьте Вашу работу в Chrome, Safari (iOS)
Проверьте Вашу работу в Chrome (Android)
5. Оптимизация (бонусное задание на позицию джуниор и обязательное на позицию миддл). Минимизируйте и оптимизируйте css, js, изображения, etc. Для этого Вам необходимо развернуть Вашу работу на любом доступном Вам хостинге и домене.
 - 5.1. Проверьте Вашу работу используя Google Page Speed и убедитесь что Ваша работа в зеленой зоне.
 - 5.2. 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).
 - 5.3. Проверьте Вашу работу используя Webpagetest и убедитесь что показатели удовлетворительные.