стажировка в Salt & Pepper no направлению frontend разработки

куратор: Валерия Зеленова valerie.z@snp.agency

# Пара вступительных слов

Твоя стажировка будет состоять из двух **последовательных** этапов - базовой части и обучения с ментором (куратором-разработчиком).

## Базовая стажировка

Первый этап стажировки включает в себя изучение основ HTML, CSS и JS.

Начинается он с прохождения курсов <u>HTML Academy</u>. Когда ты пройдешь 80% курсов, ты сможешь приступить к выполнению практического задания (его описание на стр.3). После его выполнения мы протестируем твою работу и дадим комментарии по коду и визуальному исполнению. Если ты уверен в своих силах, имеешь опыт в верстке и готов начать сразу с задания - напиши куратору об этом.

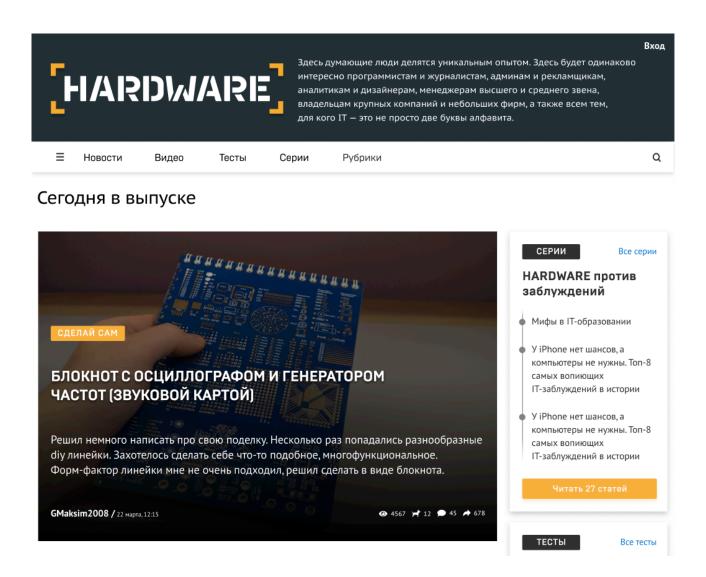
После сдачи первого задания переходи к изучению JS и, как будешь уверен в своих силах, приступай к выполнению второго. Мы его проверим и дадим рекомендации. Как только задание будет выполнено хорошо, начнем стажировку с ментором.

### Стажировка с ментором

Каждому стажеру мы выделяем куратора из числа разработчиков компании. Совместно с ним вы определяете индивидуальный план стажировки в рамках стека технологий, используемых в компании.

Если ты круто проходишь стажировку, тебе может быть предложено вступить в ряды компании на должность junior-разработчика.

# Первое тестовое задание «Hardware»



#### Задача

Твоя задача сверстать главную страницу сайта, ориентируясь на предоставленные материалы в сервисе для веб-дизайна Figma. Задание направлено на обучение верстке без использования JS. Поэтому функцию слайдера реализовывать не нужно, достаточно его сверстать.

Сайт на проверку нам должен попадать в состоянии, которое ты считаешь максимально хорошим и без багов. Помни, что качество работы мы оцениваем по качеству кода и визуальному исполнению.

# Требования

Страница должна быть сверстана в пяти вьюпортах согласно дизайнам:

- 1920px
- 1440px
- 1024px
- 768px
- 320px

**Ховеры** на сайте должны быть реализованы согласно последнему экрану. Анимация переходов между состояниями на твое усмотрение. Ховеры на блоках "Тесты" и "Видео" должны быть аналогичны ховерам на статьях в блоке "Сегодня в выпуске". Ховеров и активных состояний в мобильной и планшетной версии быть не должно.

**Шрифты**, используемые в дизайнах, можно найти по <u>ссылке</u>.

Верстка должна быть **Pixel Perfect.** Иными словами, каждый элемент сайта в заявленных вьюпортах должен быть один-в-один как на превью сайта. Для ускорения процесса рекомендуем использовать специальные плагины для браузера (пример для Chrome).

При разработке, в организации стилей, нужно придерживаться методологии БЭМ.

Выполненное задание необходимо скинуть нам ссылкой на проект в публичном **репозитории** (например, <a href="https://github.com">https://github.com</a>), а также размесить на любом **хостинге**.

Приветствуется использование **git**. С его основами можно ознакомиться по <u>ссылке</u>.

Также приветствуется использование **системы сборки** (webpack, gulp) и **препроцессоры** (sass, postcss) для повышения эффективности разработки.

Для удобства разработки рекомендуем использовать редакторы типа <u>Sublime Text</u> и <u>VS Code</u>.

# Второе тестовое задание «ToDo»

Это задание направлено на приобретение навыков написания кода на **JavaScript**. Для того, чтобы справиться с ним, тебе нужно изучить следующие ресурсы:

- <u>базовый курс по программированию на Hexlet</u>
- курс по JS на Codecademy
- обучение на ресурсе Learn.Javascript:
  - документ
  - события и интерфейсы
  - знакомство с остальными разделами учебника
  - современные возможности JS (ES-2015)

Более подробно изучить язык можно по учебнику тут.

### Задача

Твоя задача реализовать приложение «**Список дел**» («Todo list») аналогично тому, как это сделано <u>здесь</u>. Важно выполнить эту задачу на чистом JS (ES6/ES7), без фреймворков и библиотек. Верстка может быть не Pixel Perfect, но должна быть без багов на всех вьюпортах.

После выполнения проект необходимо развернуть на хостинге и предоставить нам на тестирование вместе со ссылкой на публичный репозиторий.

Для начала изучите теорию.

Затем обретите собственный стиль программирования.

И, наконец, забудьте об этой ерунде и просто пишите код.

**George Carrette**