

Практическое обучение

Чтобы научиться работать с сборкой и создавать сложные интерактивные элементы в корпоративном стиле, начинающим разработчикам предлагается пройти экзаменационное задание.

Описание задачи

Необходимо сверстать приложенный макет на нашей сборке, используя ES6 классы и компоненты:

- не требуется адаптив и pixel-perfect;
- требований к браузерной поддержке также нет (за исключением двух последних версий Chrome и Firefox);
- верстка должна лежать в созданном вами репозитории;

После выполнения работы проводится code review.

Алгоритм выполнения задачи

1. Создать приватный репозиторий в bitbucket, где установить Workspace с вашим именем, придумать название репозитория и клонировать его в свои проекты. Далее следует предоставить доступ к репозиторию пользователю ... (см. настройки репозитория);
2. Добавить в созданный репозиторий все файлы из нашей сборки;
3. Установить необходимые npm-зависимости, ознакомиться с макетом для работы, обсудить задачу и задать вопросы, если таковые имеются;
4. После выполнения работы залить готовую вёрстку на <https://test....ru/>, создав внутри свою уникальную директорию, выложить ссылки на репозиторий и тестовую страницу в задачу. Доступы к тестовой площадке взять из Стаффа

Макет для работы

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

(шрифты можно взять из репозитория)

Требования для каждого блока

1. *Бегунок* представляет собой ценовой диапазон от 0\$ до 100\$. Обе точки можно перетаскивать при помощи мыши. Подсказки следуют за перемещением точек, их содержимое также меняется в зависимости от текущего значения. Для реализации использовать **nouislider** и создать отдельный компонент.
2. *Слайдер* представляет собой блок с меняющимися только по клику на стрелки картинками. Справа от слайдера выводится динамическая информация о выбранном слайде и количестве текущих слайдов (подсчёт всех слайдов выполнить на JS). Для реализации использовать **swiper (sliders.js)**.
3. *Карта* представляет собой интерактивный элемент, для которого необходимо подключить соответствующее API. Метки не имеют балунов, однако их стиль меняется в зависимости от наведения мыши. Координаты меток (и центра карты) не принципиальны, однако всего их должно быть три. Для реализации использовать **Yandex.Maps API** и **инициализировать карту асинхронно через JS в отдельном компоненте**, а массив с координатами меток и данные для центра карты **положить в разметку**.
4. *Кнопка* представляет собой инструмент открытия модального окна. Попап загружается динамически, далее в него вставляются выбранные значения из бегунка и координаты активной метки карты. Для реализации использовать **fancybox (modals.js)**.
5. *Форма* представляет собой четыре поля: телефон, фамилию, комментарий и чекбокс для принятия условий. Комментарий, чекбокс условий и телефон являются обязательными полями. Телефон помимо обязательности должен иметь соответствующую маску ввода. **Фамилия должна быть исключительно на русском языке**. Реализовать валидацию формы **по образцу из сборки, используя data-attr** и при необходимости, добавив свой js (как в formValidation.js, так и для подключения масок ввода). Отправку успешной формы делать не требуется.