На момент сдачи реализовал javascript-компонент. Функционал: сортировка по столбцам (при нажатии на заголовок), фильтрация+ перефильтрация. Запускается с локального сервера.

Не успел: клиентская пагинация, построение таблицы из выбранных столбцов, кнопка, добавляющая новый блок(выбор набора данных + таблица).

Задание по верстке даже не брался, сори…

**Тестовое задание.** Задание рассчитано на 16 рабочих часов.

Если Вы готовы его выполнить, напишите комфортный для Вас срок сдачи с учетом Вашей загруженности. Задачу вы можете выполнить, используя любой интерфейс и удобные для Вас инструменты.

Необходимо разработать javascript-компонент для построения таблицы с дополнительными возможностями для пользователя.

**Функционал:**

- Клиентская пагинация: данные необходимо отображать постранично, максимум 50 элементов на страницу, необходимо предоставить пользовательскую навигацию для перехода по страницам.

- Сортировка по столбцам: при нажатии на название столбца, строки таблицы сортируются по возрастанию, при повторном клике - по убыванию.

- Фильтрация: компонент предоставляет текстовое поле, в которое пользователь может ввести текст и строки таблицы, данные которых не содержат подстроку, введённую пользователем, скрываются. Перефильтрация осуществляется на каждое изменение значения поля.

Для демонстрации работы компонента необходимо сделать страницу, при заходе на которую пользователю предлагается выбрать набор данных: маленький, большой или свой. При выборе набора данных он загружается с сервера и по данным строится таблица. Также необходимо сделать кнопку, которая добавляет ещё один такой же блок: выбор набора данных + таблица.

По клике на строку таблицы значения полей выводятся в дополнительном блоке под таблицей.

**Формат данных от сервера**

Сервер возвращает JSON-массив, первой строкой которого является hash структуры таблицы, где ключом является идентификатор поля, а значением - его название, а последующие строки являются массивов из значений полей в том же порядке, в каком они были перечислены в структуре.

Пример:

[

   { id: “Идентификатор”, name: “Название”, price: “Стоимость”, quantity: “Количество” },

   [ 1, “iPhone 5”, “400”, 5 ],

   [ 2, “XBOX”, “300”, 7 ],

   [ 2, “XBOX”, “300”, 7 ]

]

Замечания

* Особое внимание следует уделить скорости работы, зависание интерфейса при выполнении операций загрузки данных, фильтрации, сортировки недопустимо;
* Разрешённые фреймворки: Angular2 (предпочтительно), Backbone, самописный;
* Пишите код так, как бы вы его писали в работе - внутренности задания будут оцениваться даже тщательней, чем внешнее соответствие заданию;
* Код должен быть организован так, чтобы его можно было заново использовать;
* Помните про обработку ошибок!

**Верстка**

Необходимо сверстать страницу по макету. Макет страницы смотрите в приложении.

Страница должна занимать всю ширину окна браузера.

Верхний блок - основное меню сайта, высотой 35px. Пункты меню расположены горизонтально с отступом 10px, разделяясь вертикальной чертой.

Левый блок - дополнительное меню, оно может иметь несколько уровней вложенности, каждый последующий уровень должен иметь отступ на 10px больше предыдущего уровня. Ширина меню - 300px. Если меню превышает высоту страница - то должен показываться системный вертикальный скролл. Этот скролл должен прокручивать только меню.

Основной блок - наверху находятся “хлебные крошки”, высотой 35px. Под ними расположен блок с контентом.

Внутри блока с контентом должна находится галерея картинок 250x250 px.

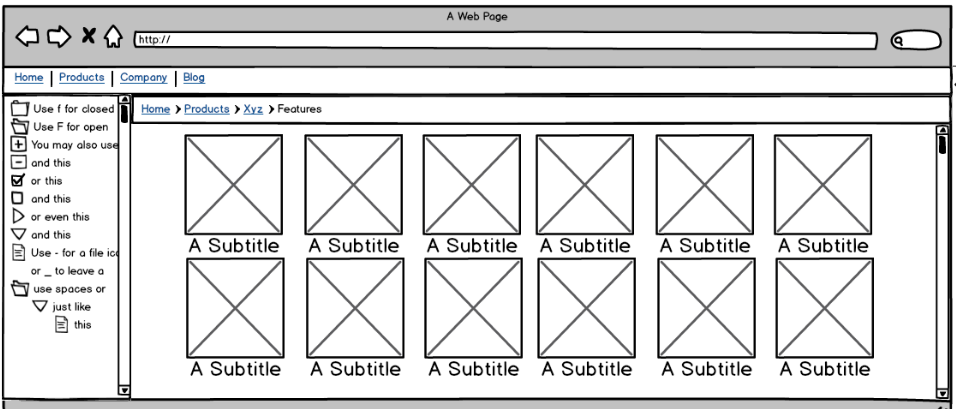
Блок контента должен также обладать при необходимости вертикальным скролом.

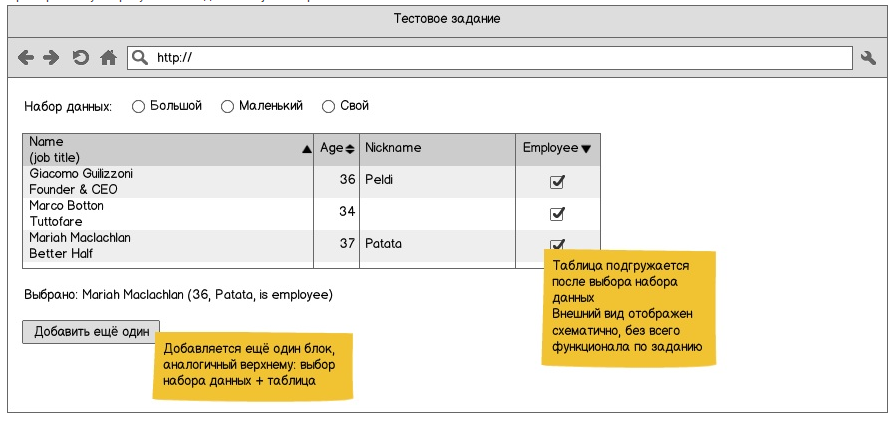
При изменении размеров окна браузера - блок контента должен растягиваться на всё доступное ему пространство, за вычетом левого меню, шапки и хлебных крошек. При прокрутке блока контента - остальные блоки должны быть неподвижны.

При вёрстке нельзя использовать таблицы, фреймы и JavaScript.

Опционально:

* должна быть возможность каждому элементу левого меню при помощи css класса задать иконку.
* предусмотреть вариант когда картинки в галерее имеют разную высоту.





Важно, чтобы в выполненной работе не было лишнего кода сторонних библиотек или другого мусора.