Тестовое задание на позицию «[Junior JS developer (frontend)](https://hh.ru/vacancy/49877217)»

В тестовом задании не будет учитываться визуальное оформление, только качество самого кода. Работа будет проверяться под браузером Google Chrome версия 96 или новее.

**Цель задания**:

* оценить возможность разобраться в чужом коде
* оценить понимание основ js (замыкания, события, this);
* оценить понимание основ css (приоритеты селекторов, базовые псевдоклассы)
* читаемость кода;

Ограничения:

Нельзя использовать какие-либо библиотеки, кроме уже подключенных

* [jQuery](https://api.jquery.com/)
* [Ramda](https://ramdajs.com/)

**Дано**:

Прототип приложения на js. В модуле реализована загрузка и отображение данных в табличном виде.

По ссылке в Google Drive находятся 4 файла:

* module.js — класс модуля, загружает данные и отображает их в табличном виде — можно изменять файл как угодно;
* base\_table.js — класс для отображения однотипных данных в табличном виде. В начале файла есть документация, стоит максимально использовать существующие возможности таблицы;
* vendor.min.js — сторонний код необходимый для работы таблицы и модуля;
* index.html — страница модуля.

Для просмотра прототипа достаточно открыть index.html в браузере.

**Что сделать**:

1. Некоторые данные отсутствуют, в данных указано null.  
Сейчас в таблице на месте таких данных пропуск.  
  
Сделайте так, чтобы вместо пропусков отображался прочерк (-).  
  
2. Для значений в столбце "Уровень топлива" не указана единица измерения.  
  
Добавьте единицу измерения «л» к значениям в столбце (10л, 53.2л).  
Помните про пункт 1, если данных нет (null), должен отображаться прочерк.  
  
3. Добавьте возможность фильтрации отображаемых объектов на клиенте.

Над таблицей нужно расположить текстовое поле ввода.

При изменении текстового поля пользователем, через 0.3сек (задержку вызова нужно реализовать самостоятельно).  
  
Будет плюсом, если при нажатии enter будет происходить фильтрация объектов до истечения задержки.

Вызов фильтрации должен происходить не при каждом нажатии на клавиатуру, а только 1 раз, после последнего нажатия. Если событие происходит чаще, то нужно запускать таймер заново.

Фильтрация проходит по всем отображаемым полям и работает как поиск вхождения подстроки в отображаемом значении. В таблице должны быть только те строки, которые проходят фильтрацию.

4. Добавьте визуальное выделение шапки таблицы (например, полужирный шрифт, фон) и горизонтальные границы между строками.

5. Добавьте визуальную подсказу при наведении мышкой на строку (например, цветом).

6. Отсортируйте данные в таблице по убыванию скорости. Если у нескольких строк скорость одинаковая, то строки должны быть отсортированы по номеру БУ в алфавитном порядке, при этом строки без БУ должны быть внизу списка.

Для сортировки используйте this.\_storage.sort, данный метод работает аналогично сортировке в массивах.

// Task 6  
const table = ***document***.querySelector('table');  
const sort = ***document***.querySelector('#sort');  
  
const sortTable = function (index){  
 const tbody = table.querySelector('tbody');  
  
 const compare = function (row1, row2){  
 const rowsData1 = row1.cells[index].innerHTML;  
 const rowsData2 = row2.cells[index].innerHTML;  
  
 return rowsData1 - rowsData2;  
 // switch (name){  
 // case 'kek':  
 // return rowsData1 - rowsData2;  
 // break;  
 // case 'lol':  
 //  
 // break;  
 //  
 //  
 // }  
 }  
  
 let rows = [].slice.call(tbody.rows);  
  
 rows.sort(compare);  
  
 table.removeChild(tbody);  
  
 for(let i = 0; i < rows.length; i++){  
 tbody.appendChild(rows[i]);  
 }  
  
 table.appendChild(tbody);  
  
};  
sort.addEventListener('click', (e) => {  
 const thead = ***document***.querySelectorAll('thead');  
 for(let i = 0; i < thead.length; i++){  
 if(thead[i] === 3 ){  
 sortTable(3);  
 }  
 if(thead[i] === 1){  
 sortTable(1);  
 }  
  
 }  
  
  
});  
  
// table.addEventListener('click', (e) => {  
// const el = e.target;  
// if(el.nodeName === 'TH') return;  
//  
// const index = e.cellIndex  
//  
// sortTable(index);  
// });

*Желаем успехов в реализации и ждем Вас в нашей команде.*