I have made the whole project using jQuery because it is the library where I have my strongest skills.

Building the source files

The project needs a running local server (node, xampp, etc) to work properly because there is an Ajax call that has to be performed to grab the sales data.

All the CSS is built from SASS files that are in the *sass folder*. All the source files for the JavaScript are in the *js folder* while the built file is in the *dist folder*. To build them again run Gulp.

If you don’t have Gulp installed, in a node environment run a node modules installer first (npm or yarn) and then Gulp.

Entry Point

I have created a stand-alone plugin that can be called directly from an external entry point (i.e. another plugin or into the HTML, etc.) and that accept a starting date (divided in day, month and year) and an end date (again divided in day, month and year). Be careful that the month is calculated like JavaScript does starting by 0 (zero).

$(document).displaySalesTable({

startDay: 1,

startMonth: 10,

startYear: 2016,

endDay: 15,

endMonth: 1,

endYear: 2017

});

If you are not passing any value to that plugin, it takes as default the date of yesterday (as start) and today (as end).

Ajax call to get the data and Mustache.js

The easy part of this exercise was to retrieve the data from an Ajax call that simply read the "sales.json" file and show each object that is inside that file, into a table.

To show the table rows I have used also *Mustache.js* a very powerful template library. Using the template inside the HTML file, Mustache can render the rows very easily with the correct data.

Sorting the Objects by date

To sort these objects by date there is a specific function inside the plugin that do the job by calling the sort array method from JavaScript.

Once the array is sorted, the plugin shows only the rows that are inside the range specified in the entry point into the HTML file (if nothing is passed it is using the default values).

5 most sellable items

The hardest part for me was to filter the most 5 sellable item and show the graph about them.

To find the most sellable items of the range selected, during the function that filters the item between the date range, there is also a call to another function that basically is creating a new object of objects where for each product code it count the quantity of the items.

This quantity is used furthermore to sort all these items by quantity and save only the 5 most sellable in a new array that is finally passed to the graph library.

Graph

To show the graph of the most 5 sellable items, I have used another library ZingChart (<https://www.zingchart.com/>), that I have found easier to learn that D3.js to create a pie chart and with a lot of customization.

The colors for the slices are random, so do not be surprised if you will see some differences each time you reload the page or if you may do not see a slice because the random color can match the background in some rare cases.

This library has some very nice features like the ones where you can customize the texts, the overlays to each slice and the animations when you click the slices.

Conclusion

I know that it is not the fastest plugin and maybe it can be improved a lot, but with the less time that I had this week this is the best that I was able to find to complete this test.