**Data Grid Test by Fahim Chowdhury**

**Spreadsheet URL**

https://docs.google.com/spreadsheet/ccc?key=0ArMvT6nVqkMPdHBkazE2ZGZvVXQyYzFmQ2dSckRiWVE#gid=0

**Live Demo URL**

fahimchowdhury.com/demo/datagrid/

**Tested on the following Browsers**

-Chrome Version 29.0.1547.22

-IE 10.0.9200.16635

-Firefox 22.0

**Code was written is the following Languages**

-CSS

-HTML

-JavaScript

**JavaScript Libraries**

-Toolkit JS (https://github.com/fahimc/Toolkit-JS)

**Architecture and Structure**

-Model-View

-OOP

-Component design pattern

**Assumptions and Issues**

I really enjoy this test and if I wasn’t busy at work this would have been done to a level that I would be happy with. There was a few things that I missed out not for the lack of trying but because of time.

I wanted to showcase my level of JS so I used as much of how I code complex projects in here such as the architecture, design patterns and JS UI building.

I used ToolkitJS which is a JS framework written by myself. Its great of OOP and for implementing many/multiple design patterns. Please take a look at it here:

<https://github.com/fahimc/Toolkit-JS>

I would have also used Jasmine to test the JS Classes but again I didn’t have enough time.

Below is a list of things that I had issues with:

* **HTML**: I wasn't sure if you want to see my level of JS so instead of build most of the view in the HTML I've opted to build it in JS.
* **Background Image**: This doesn’t work well for taller browsers windows so I wasn’t sure if you wanted it to scale therefore I let it replicate.
* **Lack of time**: I would have implement much better and added a few things that I missed but I didn't have enough time due to my current workload.
* **Pixel perfect**: I didn't make it scalable depending on you browser window or optimised the view. I wasn't sure what was required.
* **Auto detect String and Number for sorting**: This need more thought as I rushed it I didn’t implement a generic data entry parser and I could have created a better data item class which can take in dynamic values and detect the type (String or Number). An Interface would have been ideal to use for the view class.
* **Hover State**: Missed the horizontal highlight, ran out of time. I would have given each cell a row and column index to track the current cell and on hover use JS to highlight the relevant column and row.
* **Adding new Columns**: Missed this, ran out of time. This would have made me rewrite a lot of what I have done to make the labels detached from the data. I was not clear on the data and label separation until I read this part. Not a complicate job but would have taken more thought in the architecture.