**Our website to-do list** **Performance Analysis**

**Network Performance**

This test will cover how fast the site loads which includes loading resources and how fast execution for each function

**Test Environment**

1. Network throttling is set to simulated fast 3gGwith 4 times CPU slowdown

2. Cached has been disabled to simulate real user experience when first time visiting the site

3. Test is set in desktop environment

**Loading Result**

It takes 1.7 seconds before the site is fully loaded and become properly function. The cause probably are caused by:

* 8 scripts are loaded, this is considered as too much for such as small application
* 2 style sheets are loaded
* Some of the scripts that is written is cause a block render, initial analysis was probably caused by the fact the style sheets are loaded in the head of the document

Total downloaded resource is 47.4 Kb

**Possible Optimizations**

Below are several options on how to optimize the apps for future optimizations:

* Change server to HTTP2 rather than HTTP1. Because HTTP2 allow download 12 simultaneous connections rather than HTTP1 which only 6 simultaneous connections
* Minified the scripts as well as the style sheets which can make the files to become smaller, smaller files means faster download.
* Fetch the critical style inline in head and fetch the unnecessary using preload

**Competitor website analysis**

**Loading Result**

It takes 9.7 seconds before the site is fully loaded and become properly function. This is really bad as based on the google performance metrics, people will abandon website that is loaded more than 5s.

* 12 images are being downloaded with an size of 634Kb and itself took 4.04s to fully loaded
* The images that is served are not in next-gen formats
* 29 Scripts are being downloaded with a total of 672Kb and took 3.01s to fully loaded, one of the scripts are Jquery which is considered as overkill for such an small app
* 3 style sheets are being downloaded and lots of unsued css as well as the style sheet caused render blocking

Total downloaded resource is 3.4Mb

**Possible Optimizations**

Below are several options on how to optimize the apps for future optimizations:

* Move or defer parses blocking scripts such as loading script for ads, Facebook, and twitter, so it does not block HTML parse.
* Use only minified scripts, jquery-ui is been used with the un-minified version
* Keep only critical style inline in head and fetch the rest using preload
* Preload the fonts so that the app do not have to wait until the CSS is parsed before fetching them
* Use next-gen image format

**Runtime Performance**

The apps is rendering recalculating the style for each executions which can be optimized to just render the changes style