1. **Introduction**

GAN Integrity Solutions aims to maintain a stimulating work place with dedicated professionals, where everyone understands the importance of high quality, maintainable code, and scalable solutions. The tasks given to candidates being considered for open positions are meant to ensure that the developers we hire are well suited for the work in our team. Because we are doing this, we can guarantee a very interesting and vibrant work environment, with smart, experienced and highly motivated people, who learn from each other and develop enterprise-grade solutions together, using the newest platforms and technology available.

We very much appreciate your interest in the company, and the time you spend on this. There are three parts to this task you will need to complete:

1. **Dashboard SPA in AngularJS** – deliver a working package
2. **Code Review** – write in-line response in this document
3. **Process Notes** – write in-line response in this document

When preparing your solution, keep in mind that it will be a significant element in our evaluation of you as a candidate for the position, and that the submitted material will undergo a thorough internal review, and will be checked for plagiarism. This is meant to show us your abilities and approach to a development task.

1. **Dashboard SPA in AngularJS**

Develop a simple Dashboard SPA that can add and remove widgets that shows charts & tables for email sent/received over days, weeks or months, email sent reply rate in percentage average.

Specifications:

* Use Bootstrap for building and designing the UI and use AngularJS for the SPA
* Use the fictive data from the email.json file attached
* **Optional** - AngularJS Directives to reuse elements where possible
* Possibility to add or remove widgets on the dashboard with content as show charts, tables for email sent/received over days, weeks or months, email sent reply rate in percentage average.

Notes:

* As a guideline, you should spend maximum 8 hours in total to develop the application

If you use any components you did not write yourself, other than standard AngularJS, please clearly state where and why it was used in the **Process Notes**. Hand in a solution you are pleased with, but also be sure to explain where and how it could be improved in the **Code Review**.

A copy of your solution should be provided in a RAR or ZIP file, together with a copy of this document, with the following sections filled in with your responses.

1. **Code Review**

**Code review Initial thoughts:**

It is a typical implementation of AngularJS that bind data and see dynamic data changes.

Not a difficult task but many detail need to be design well. The interesting part to me is to choose a library to show data in animation and chart graphic. Design one share template and several individual templates for panels is what definitely I must do. It also needs a directive that can choose template dynamically. The most twist mind part is to fill data correctly by horizontal or vertical. Moreover, a local storage can help user to load their preferences. Above thoughts are what II can see before I architect this small task.

**What has been presented:**

The page separate into two parts, data visualize panels and side bar. My aim is to make user can click check box to open and close data panels in left main area. Correspondingly, if user click “X” to close a panel window, the check box will be also unchecked. Panels use different way to show data, including bar chart, curve chart and table. The chars shows different aspects including year , month , day. Email sent , received , replied, received percentage and reply percentage.

I notice that due to the emails.json is fake data. The received and replied email could be higher than sent emails. So the percentage could be bigger than 100%. In chart panel, user can interact with graphics by hovering mouse to see details of data. Also in day chart, user can choose range of last days. The choices of switch panel on and off can be saved in local storage.

**Alternatives and improvements:**

The improvement should be optimizing data structure. It was not a good idea to reconstruct data and duplicate data. Here I duplicate data is because the data amount is not that big, and I feel clear and easy to use the reconstructed data for future use. Keep original data lightweight and less processing are important for scalable development.

The panel is in fixed size to show different amount of data, there is a need to change panel size for presenting data is a better way. Currently only switch panel on / off can be remembered in local storage. The range of day to view is also need to be stored. The way of saving panel statues is quite simple for now, that saving all panels data in to local storage. It should be only store useful data but not all data structure.

**Code style**

Keep to follow Angularjs Google Style, and Google JavaScript Style Guide , to structure folder and connect components. Keep more comments, especially at complicate logic and special reason for some detail loop and condition. Keep using restrict model and declare each variable in a local scope, that is better for debug and avoid human errors. Keep using indent for loop and conditions, try to use less line, and avoid duplicate code. keep code clear and simple .

1. **Process Notes**

As it is a small and fast task, I implement all controllers, directives , and services in one file: app.js.

Besides bootstrap , jQuery , angular, I also used a chart library called Angular-charts.js, which has a dependency called Chart.js. The Chart.js has a bug when hover mouse on Bar chart and Curve chart. I also debuged this public library to make it works. Moreover, for local storage purpose, I also need to include a ngStorage lib to save which panels are switched on. At last, I have my own “small tools lib” called smt.js that includes often-used functions crossing projects.

Starting development by set an environment can see page result and console trace. Chrome is my prefer working browser. I cut tasks in to small piece and create basic stricture is my priority, including make all basic folders and work files, include all fundamental libraries, write npm and bower files. Then fill code to implement each feature. Overall, I am working from big to small, fundamental to advance. So I am head to create the feature that switching panel on and off as a start.

Because panels looks pretty similar, they needs somehow dynamic template from one directive. Angular didn’t provide an official way to do this, but one classical way is to combine link, template, and ng-include together to reuse the template. Link can connect to scope and attribute after Parent template compiled. Template can load dynamic info by calling a method defined in link.

Both email data and widget are loading from external JSON file by $resource, and use ***get*** method to finish loading to a service, means that the data will be update after the url request done, therefor a watcher may be needed if I update date is controller later. The ***get*** method from $resource return not only includes respond data, but also other object list promise and functions. I did some filter detect in later code.

Basic data is loaded by a service array called widgetServices, which load both emails and widgets from JSON. Also some preprocessing works need to be done in services. This function generate new versions of data called dayData, YearData , and MonthData. And also create another version that transposes column and row, which called dayDataRev, monthDataRev, and yearDataRev, and also time labels Array for easy use in panels. I found the tasks are similar among these three-time categories. I write a function avoid duplicates codes which can reuse for three them all. A local scope object can help make all things automatically and dynamically.

When I implement local storage features, found that I need to solve first time loading and no cookie situation, such as in browser incognito model in different browsers. the page should be load in all cases. So I set up a ng-if to detect if a local-storage is available. Otherwise the page will just load data without storage. In this process I need to be ware that sync all interactions by bind together. Check box, click close button, and update local storage should be slicked on one parameter.

Last day cut is also an interesting feature to implement. I can get all days data, but 120 days are too much to showing together. I decide to keep last 15 days at beginning, and give an input box to allow user to change the range of view time. The labels and all series data are also need to be stick together.

Some code clean work still need to be done, reduce duplicate code. Or remove useless code.

At last , the code need to be run under a http server, as loading external json and html.