Quantitative

Completeness: Did you complete the required task?

YES, I was able to show the list of the Team Members with the desired Wireframe

Readability: Is it easy to read your code at a glance and understand how it works?

I put comments for people to understand it.

Robustness: Will your code still work given unexpected circumstances?

The circumstance that I check is if it will have values more than 6, the members will show only 6 cards at a time and the rest of the cards will be scrolled to the right.

Algorithm Design: Is your solution coded in as few possible steps?

It might be possible since my code is not using high quality coding technique

Compatibility: Can this code be reused?

Yes, as long as the API structure is the same.

Qualitative

Code Factoring: Does your code avoid redundancies?

No, CreateElement can be refactored for sure. I'm just not sure how to implement it with the current template.

Scalability: Will your code work given a larger amount of size?

It depends, for sure it will work for a lot of data and you can see all of it by scrolling. But getting all the data at the same time might slow down the browser or if the user is using mobile, its not a good idea. I'm thinking of only getting parts of the API, rather than getting all at the same time.

Documentation: Do you explain how your code works?

Yes, the comments on the code states why I used or go with that approach.

Test Cases: Did you test your code in different circumstances? How?

Yes. I wish I can actually get a lot of data on the API, but since I wasn't able to get on the API, I use the template in HTML and copy and paste the code if it will scroll or not. Or at least how many values I can take to show in the browser.

Debugging: Are you using debugging techniques?

I am using console.log to determined the kind of data I'm getting from the API. I also try to get a specific data by same method.

Semantic

UI/UX considerations: How will a user interact with your code?

Only a Developer can change and understand my code. But for the website interaction, the user will be able to scroll left or right if data is more than 6 values.

Data Structure: How is it built?

Once the website is loaded, the function getData() will call the API. The data on the API that gathered will be send thru a call back function to the get the Template for the data and then show it to the HTML.

Maintainability: Would it be easy to adjust your code?

Yes. I separate parts by parts the codes that the developer should look at if he/she wants to change the cards of members.

Directive: Can you describe exactly how you would approach the task?

First, I check if I can get the data on the API so that I know that I can retrieve the data. Next, I create a HTML template where I will be setting my data. After that, I build the CSS and see if I was able to produce the wireframe that I was looking for. Once I have the template for HTML and CSS, I will now focus on JavaScript. Since I have the template already, I was able to create the HTML components from the JavaScript to the HTML without having second thought. After that, I adjust some CSS, or HTML or JavaScript to show the proper execution of the code.

Scheduling: Can you describe

It took me almost 5 hours to finish this task. The template and the research for adjusting and testing took a while since it is new and it will come from scratch.