**Frontend developer test**

This task required me to use Flickr API which was the public API. Upon observation Flickr API was not usual standard API. I will explain this later.

I chose to use Vue JavaScript framework to the create this Single Page Application (SPA). Vue.js is a library for and not limited to building interactive web interfaces. It provides data-reactive components with simple and flexible API.

For this task I have used Vue CLI 3 to create this application. In this project, I have used SCSS to create layout and styles and AOS JavaScript plugin/package for animation. These were used to speed up the process of development.

I have tried to obtain the relevant data from the API using the Axios, a promise based HTTP client. However, this wasn’t successful as Flickr public API has no CORs header and is in JSONP format. To resolve this issue, after lot of tests I installed JSONP package. This allowed me to call the API using the Vue created lifecycle hook.

 created () {

*// Call the api*

    jsonp('https://api.flickr.com/services/feeds/photos\_public.gne?format=json', { name: 'jsonFlickrFeed' }, (error, data) => {

*if* (error) {

        console.log('parsing failed', error)

      } *else* {

        console.log('success', data)

*// Upon success store the data in `items` and initialise AOS animation*

*this*.items = data.items

        AOS.init()

      }

    })

  }

Thereafter, the data was stored in Vue’s data variable called items. The data from the API was stored as an array.

I constructed a layout based on the requirements given. This was done using HTML and SCSS. (see project) In order to dynamically populated data to the SPA, I used v-for directive approach for mapping arrays to elements. This is usually used for a list of items but I used it to render data in certain elements of layout as shown below:

<section *class*="container">

        <div *class*="grp-cards" *ref*="content">

          <div *class*="card" *v-for*="(item, index) in items" :*key*="index" *data-aos*="fade-up">

            <div *class*="img\_bx">

              <img :*src*="item.media.m" *alt*="">

            </div>

            <div *class*="content">

              <div *class*="info">

                <p><a :*href*="item.media.m" *target*="\_blank">{{item.title}}</a> by <a *href*="#" *class*="author">{{item.author}}</a></p>

              </div>

              <div *class*="desc">

*<!-- {{item.description}} -->*

                <p>Content here has other elements within the description. For this reason I will not filter it for now</p>

              </div>

            </div>

          </div>

        </div>

      </section>

As you can see, data stored in items (array) is looped.

data: function () {

*return* {

      items: []

    }

  },

Vue’s v-for directive requires a special syntax in the form of item on items. Items is where the API data is store. Item is an alias for the array element being iterated on. Alias can be named anything. An example of pulling in the relevant data is as shown above: in order to retrieve author’s name, it can be accessed using the dot notation which will find the author property value in the object which is in the array.

I purposefully left out description and tags elements. The description property had multiple elements within it such as image and link. It will take bit more time to figure out how to retrieve the data in a meaningful way such as an objects in array.

I am pretty much in the early stages of learning Vue.js.