

**The Wake Link:** <https://corporalkruger.github.io/SketchAPIProject---The-Wake/weather.html>

**Aim**

The aim of this project was to build a simple responsive web app that loads dynamic data through at least one web-based GLAM API and displays the content in an interesting way.

**Development Process**

Initially, I had looked at a range of other examples of both movie libraries and book libraries which would fetching API information and displaying it to the user in a clear and concise way while fetching a range of information including authors, images, release dates, actors etc. During this process I was able to see the API information was being brought into the console, however I was not having success in displaying the information being fetched effectively. After a week of editing, and attempting both fetch() and axios, http requests, and inputs into the App, I decided to seek other Archive information in the way of Weather information, Covid-19 information with the addition of a more user focus element by including an Inspirational quote generator.

**The Wake – What is it?**

The Wake is, at glance an App/service which provides a user with a quick look to the start to the day. The three (3) elements to my App (Weather, Covid update and Inspirational quote) gives a user information that lets them plan, understand and inspire their day while being useful.

Weather element

After looking into a range of examples and interfaces in how to display weather to the user, I settled on simple yet informative design. The key information for the users included a searchable location (City), date, temperature, summary of the day “Cloudy / rainy” and the “high and low”. All examples below provide this basic information, however I was unable to achieve the effect of a background adapting to the unique to the time of day, or which would visually display the summary of the day through an image such as “rainy” = a raining image. I took my, API information from the following source: base: <https://api.openweathermap.org/data/2.5/>

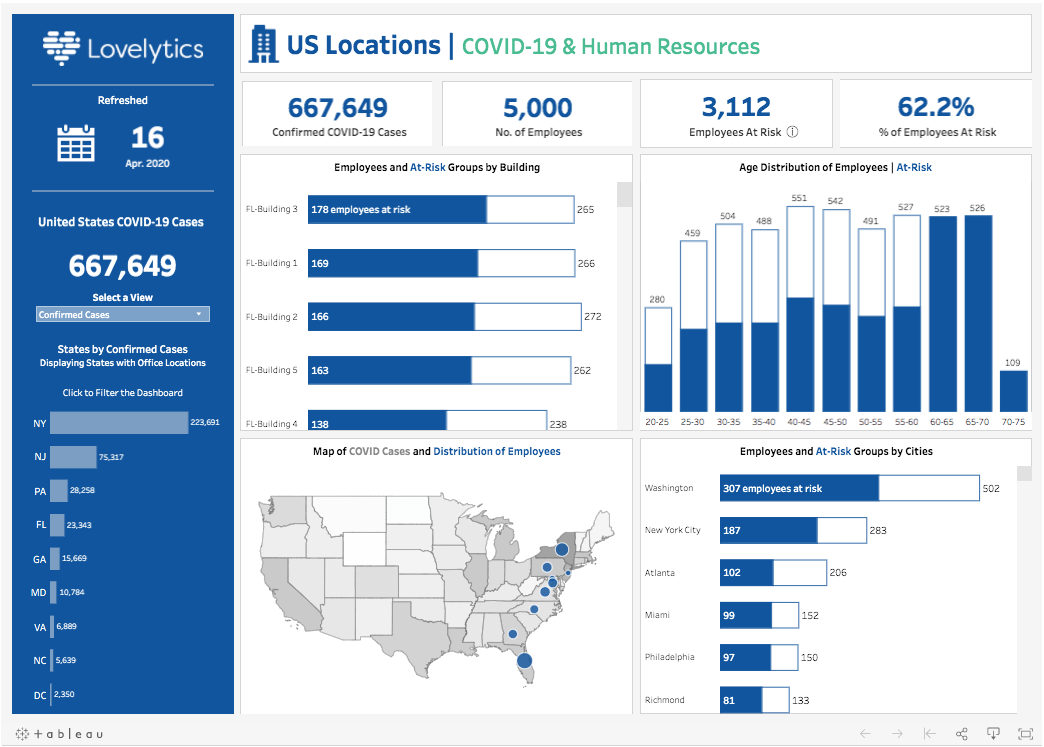
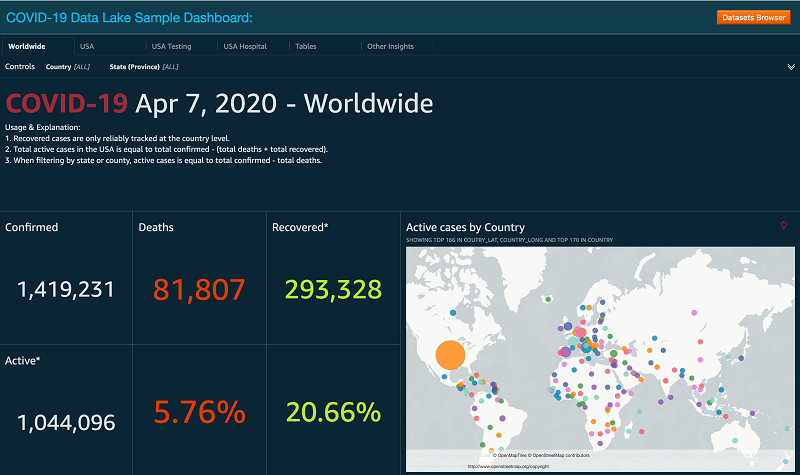
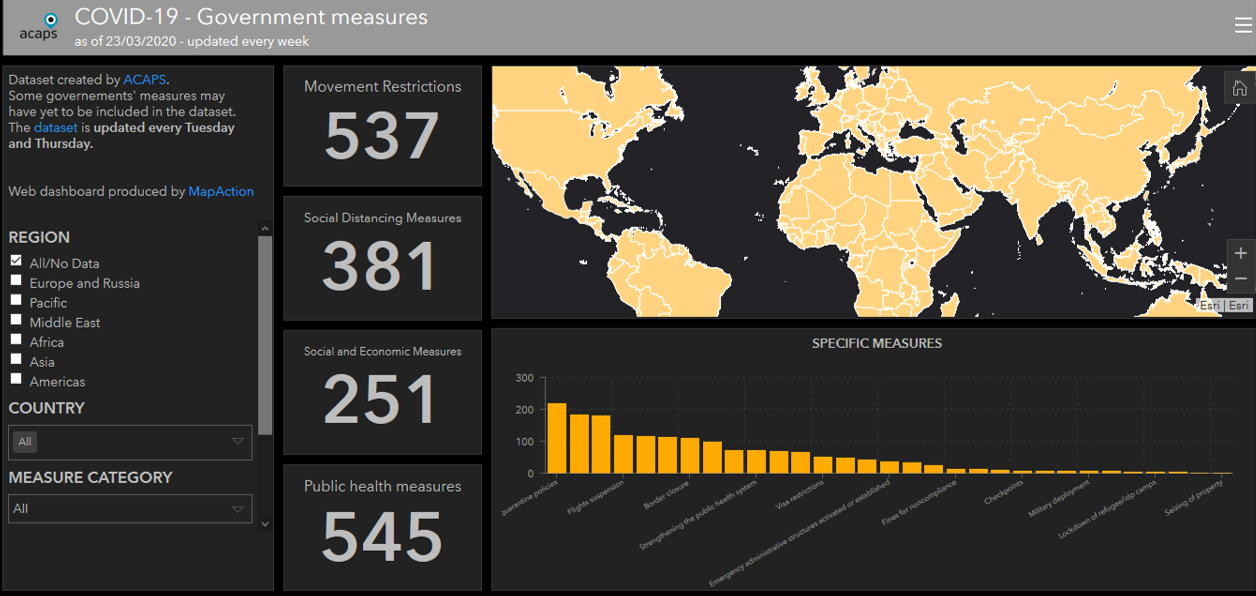
For the above information to be displayed in a concise way, I decided to add an app-wrapping convention as this allowed me to apply somewhat of a management layer to the page, which worked ideally on a mobile device. This also meant I did not require any changes to the original application



Covid-19 element

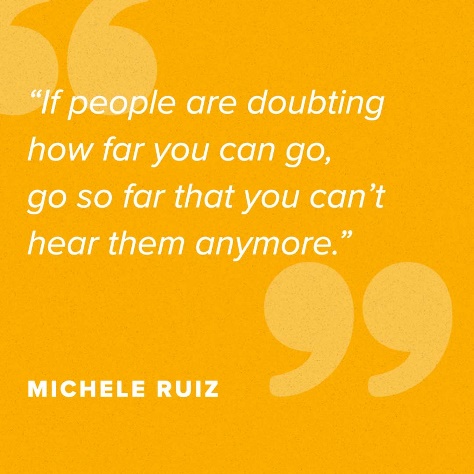
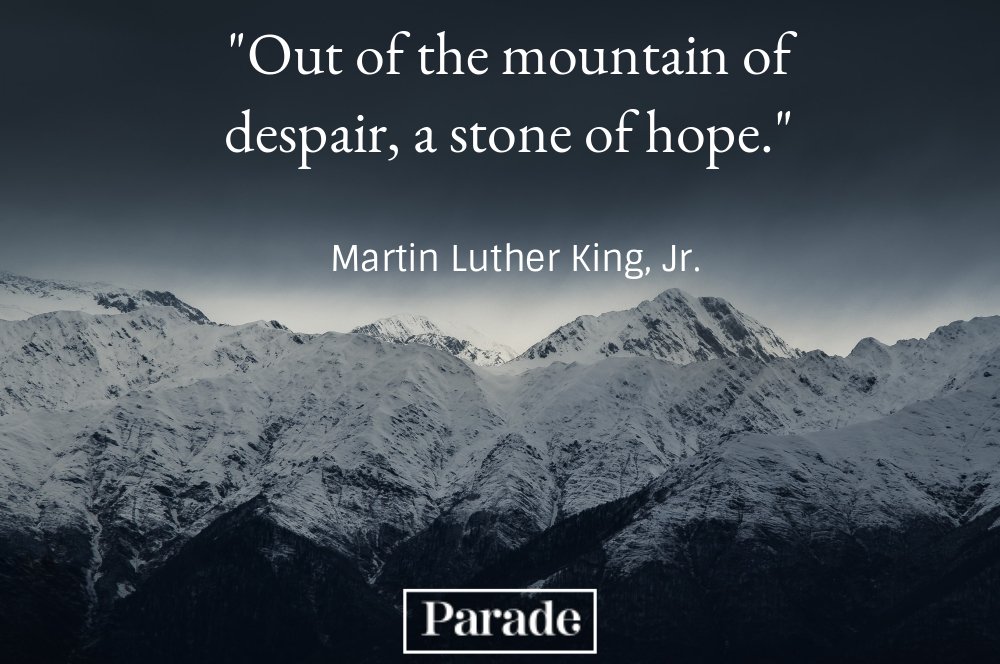
I wanted to add another piece of information which would update the user on the most important news and automatically and thought a snapshot on Covid-19 would be useful. There were plenty of examples of which some Universities were doing in tracking Covid-19. The examples below fetch a range of API data to populate and display on their UX/UI.

I did look to implement some graphics which would display cases worldwide using geo-location however the attempts at this did not result in a UI being usable. In the end, I was only able to display three lines of API with the use of <tr> tags and <td> tables. API was taken from var url = "https://api.covid19api.com/summary";

Inspirational element

In this, I was able to generate a “refresh” button that the user could interact with, and by using a math.random() return and randomising the numbers in the sourced API (fetch(`https://type.fit/api/quotes`);, the user has the ability to change the quote that suits them, making the experience a little more personalised.

The examples above were the general look and feel for inspirational quotes which made me choose the night’s sky as the hero image, while quote bold. The math.random code along with the ‘Tweet’ button allows to share an array of quotes, adding to the user experience.

**Learnings**

It is clear that fine tuning the ability to display API data from any source is a powerful tool and once in place and working has the ability to take away many manual requirements. Next time I will know that having one (1) main area of CSS for fetching API datasets will be easier to manage. Underestimating the UI/UX for displaying content was definitely noticeable, which I have learnt from.

**Overall**

In the end I was happy with eh end result however understand I did not meet the brief completely in terms of accessing a GLAM API completely. However, I felt the roundabout approach I achieved still gives a user a unique experience on each of the API’s which have been incorporated into the App. With my ability and strength leaning towards html and css, I made sure I used this in delivering a visually humbling website which is responsive on all devices, language was consistent through while ensuring tones and colours were also consistent throughout.

I did my best to ensure a user would never feel lost or misguided while using the App, with the home index.html) being attached to the logo at the top of the screen while the ‘hamburger’ dropdown also being available on all devices.