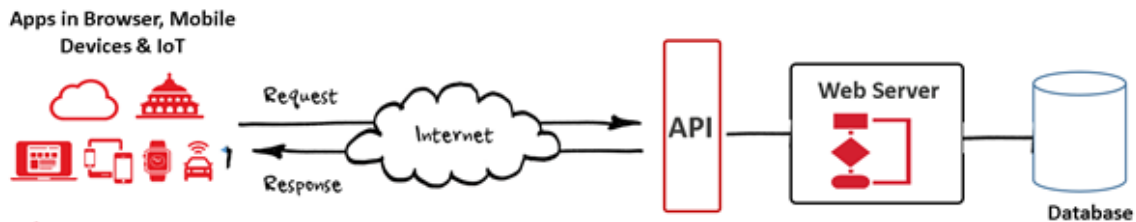


- Recapitulation of the topics from the lectures

We learned about the usage of API (Application Programming Interface) from lectures, which I found really challenging. Therefore I looked up other materials to understand the concept better. Afterall, I found the below images that explained the flow better for me from a perspective of a beginner, rather than other tutorials or articles.



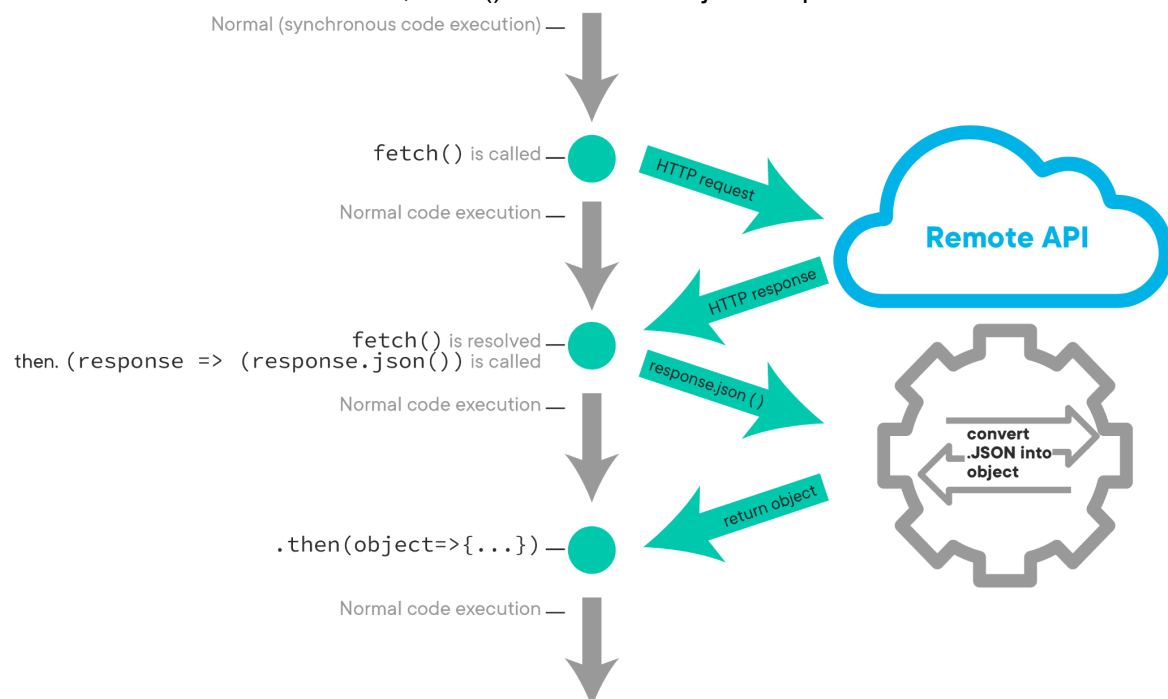
In short, API is a set of functions that allows applications to access data and interact with external software components, operating systems etc. As in the picture above, we have to make a “request” to send or receive the information that we need and this would be done by another programming language called JSON.

```

1  {
2    "string": "Hi",
3    "number": 2.5,
4    "boolean": true,
5    "null": null,
6    "object": { "name": "Kyle", "age": 24 },
7    "array": ["Hello", 5, false, null, { "key": "value", "number": 6 }],
8    "arrayOfObjects": [
9      { "name": "Jerry", "age": 28 },
10     { "name": "Sally", "age": 26 }
11   ]
12 }
13

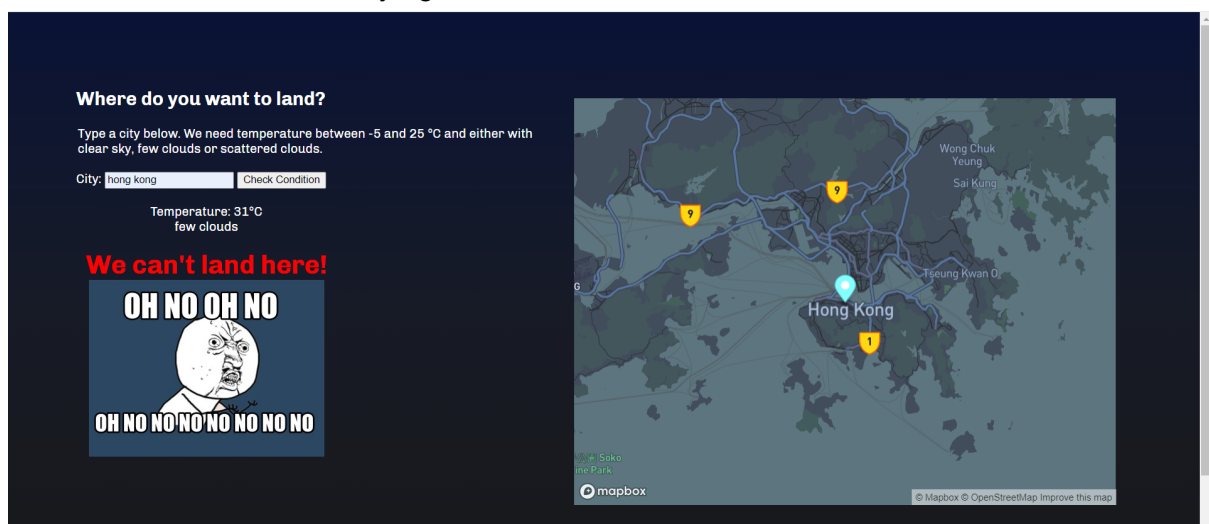
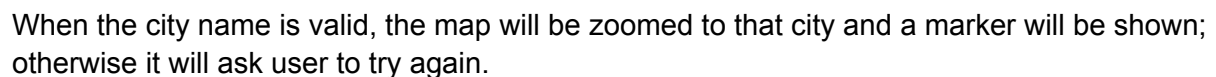
```

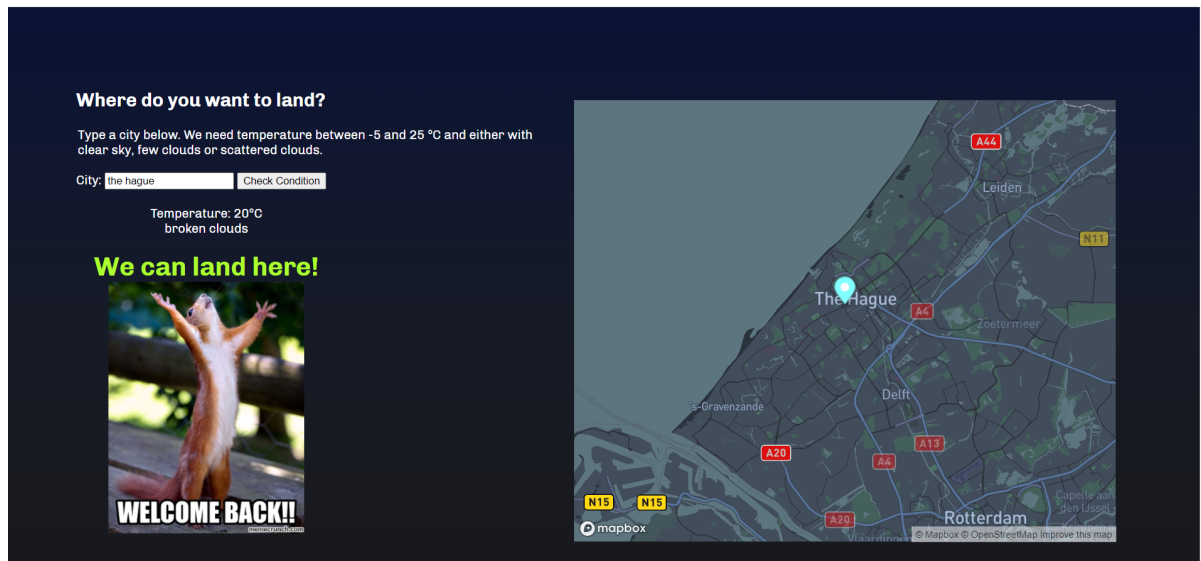
To fetch resources from JSON, `fetch()` will be used in javascript.



- About the assignment & what I tried to achieve

First of all, I customized the style for the map and initialized it for where user will see when the page is loaded. Then I set conditions of the temperature and cloudiness to decide a location for landing.

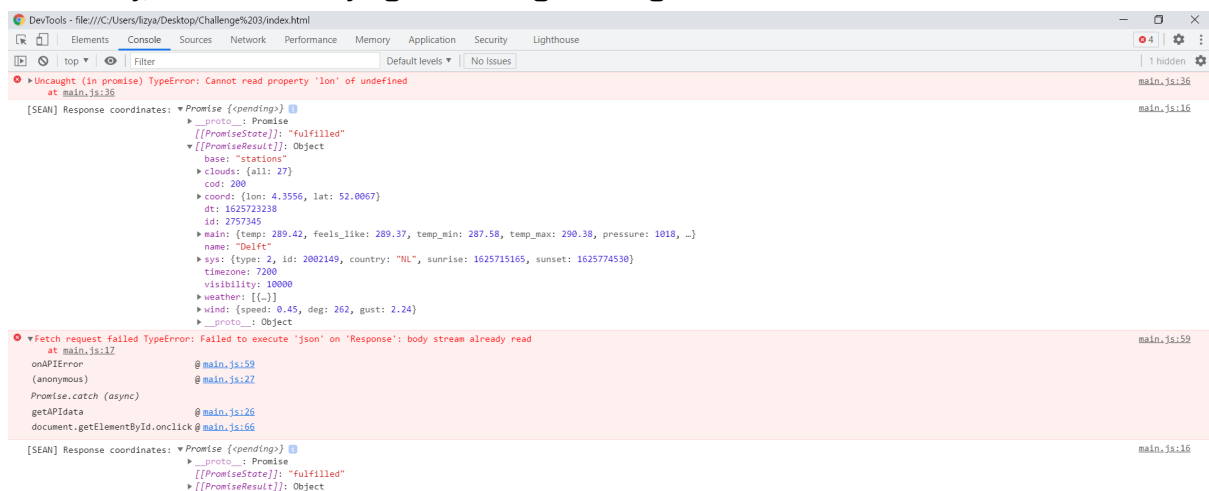




It actually took me quite a while to understand how to let the map fly to a new location based on the user's input. I found that in mapbox it has “flyto” which can execute it. However, with multiple trials, I realized it cannot respond to text (city name) but only the geographical coordinates. Therefore, I need to connect it to the result from openweather instead as the response itself isn't only the weather but also lat and lon. It still took a lot of time as I wasn't sure if I could successfully get the coordinates, but a good thing is that, I learned how to check by adding a console log. I know it has been mentioned before, but I didn't get to understand how it helps, or let's say even adding a console.log statement gave me some problems as well :D For example I didn't change it to .json before logging it.

```
.then(function(response) {
  if(!response.ok) throw Error(response.statusText);
  console.log("[SEAN] Response coordinates:", response.body());
  return response.json();
})
```

Eventually, it was a lot of trying and editing but I'm glad that it works.



It was still pretty frustrating as what I wanted to do was a lot more than that, but I cannot figure out how to do it yet. For example, I wanted to add an image and the covid situation of the corresponding country from other APIs. I tried it for a few days and decided to give up due to the deadline.