**Submission**

Submit a link to your final work on Canvas along with brief answers to the following questions:

1. Regarding Leaflet's map.locate method (documentation [here](https://leafletjs.com/reference-1.6.0.html#locate-options)), if we don't specify any options when we call the method, will the map recenter if geolocation is found? (Hint: what is the default parameter for the setView option?)

The only way I was able to figure this out was to remove my options within this method and test it. The documentation only said that the setView will set to world view if geolocation failed, not if it’s successful with no parameter. That really threw me for a loop. It’s good to know the parameters of setView.

1. Answers to the following questions about the Geolocation API documentation (<https://www.w3.org/TR/geolocation-API/>):

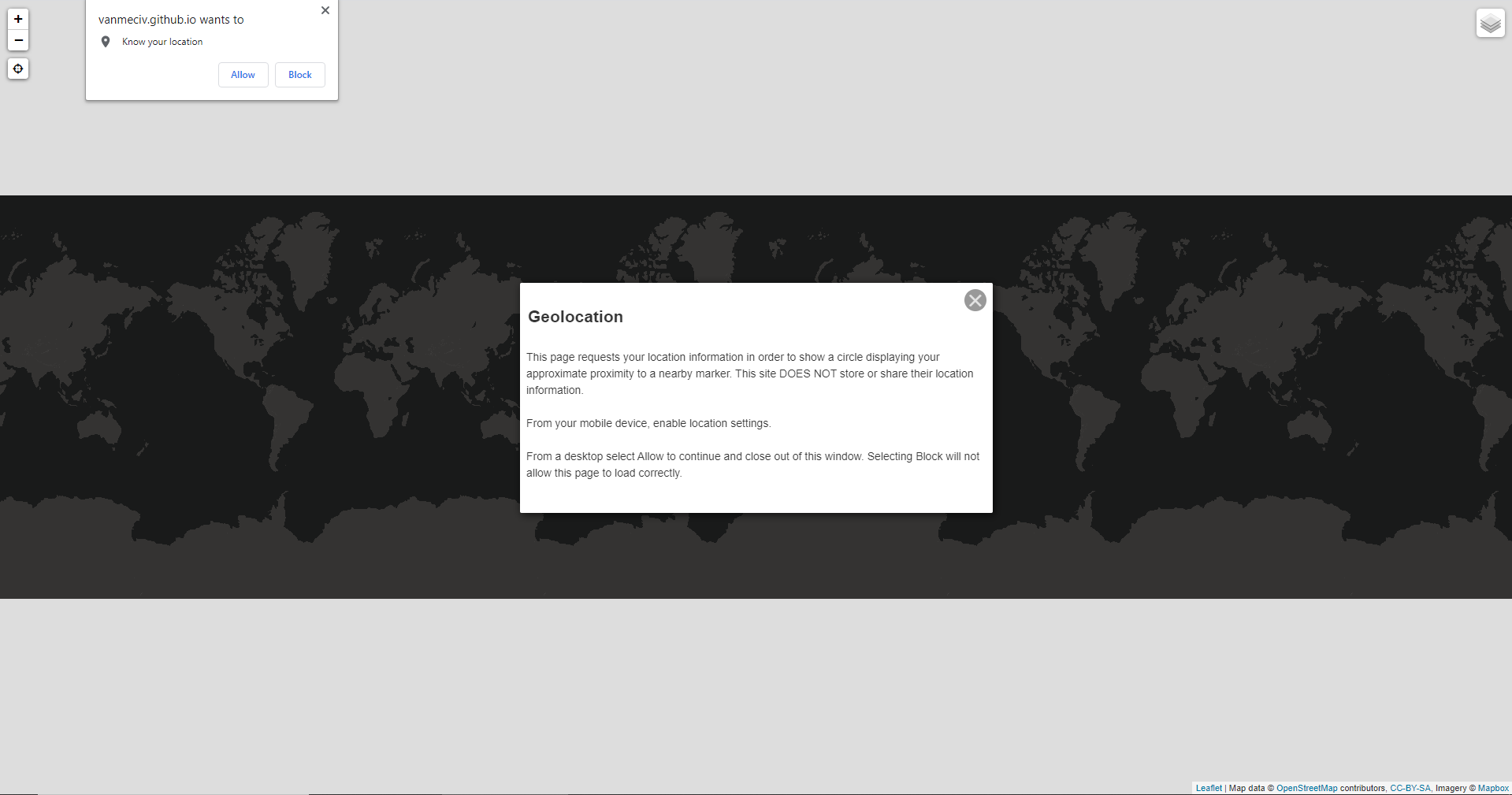
a. What is the confidence level on the accuracy value returned by the API?

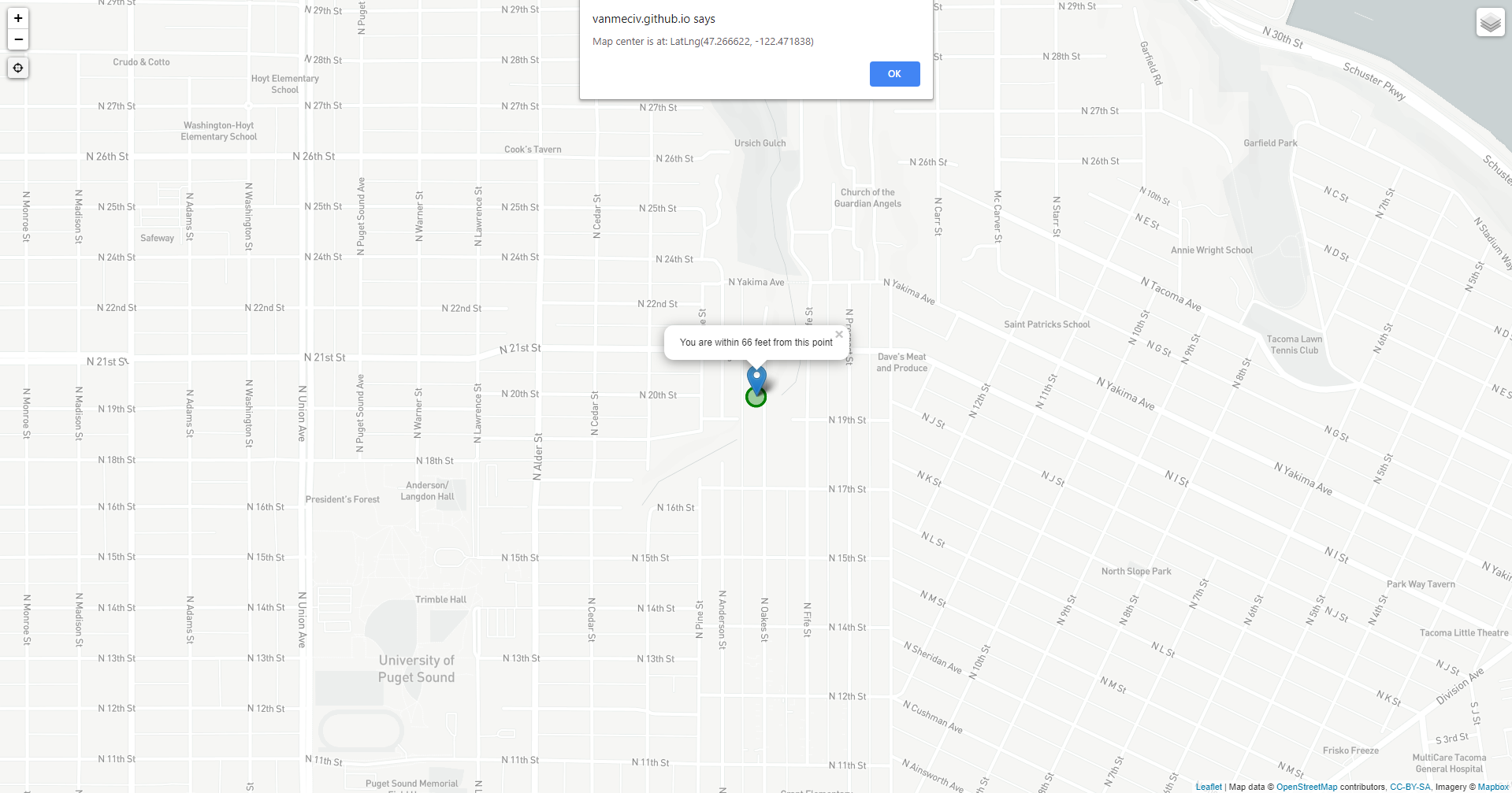
“The [accuracy](https://www.w3.org/TR/geolocation-API/#accuracy) and [altitudeAccuracy](https://www.w3.org/TR/geolocation-API/" \l "altitude-accuracy) values returned by an implementation should correspond to a 95% confidence level.”

b. In this lab, we haven't access data on the heading (or direction of travel) of the device, but using the Geolocation API, we could. If the Geolocation API returned a heading reading of 135, what direction would the device be facing?

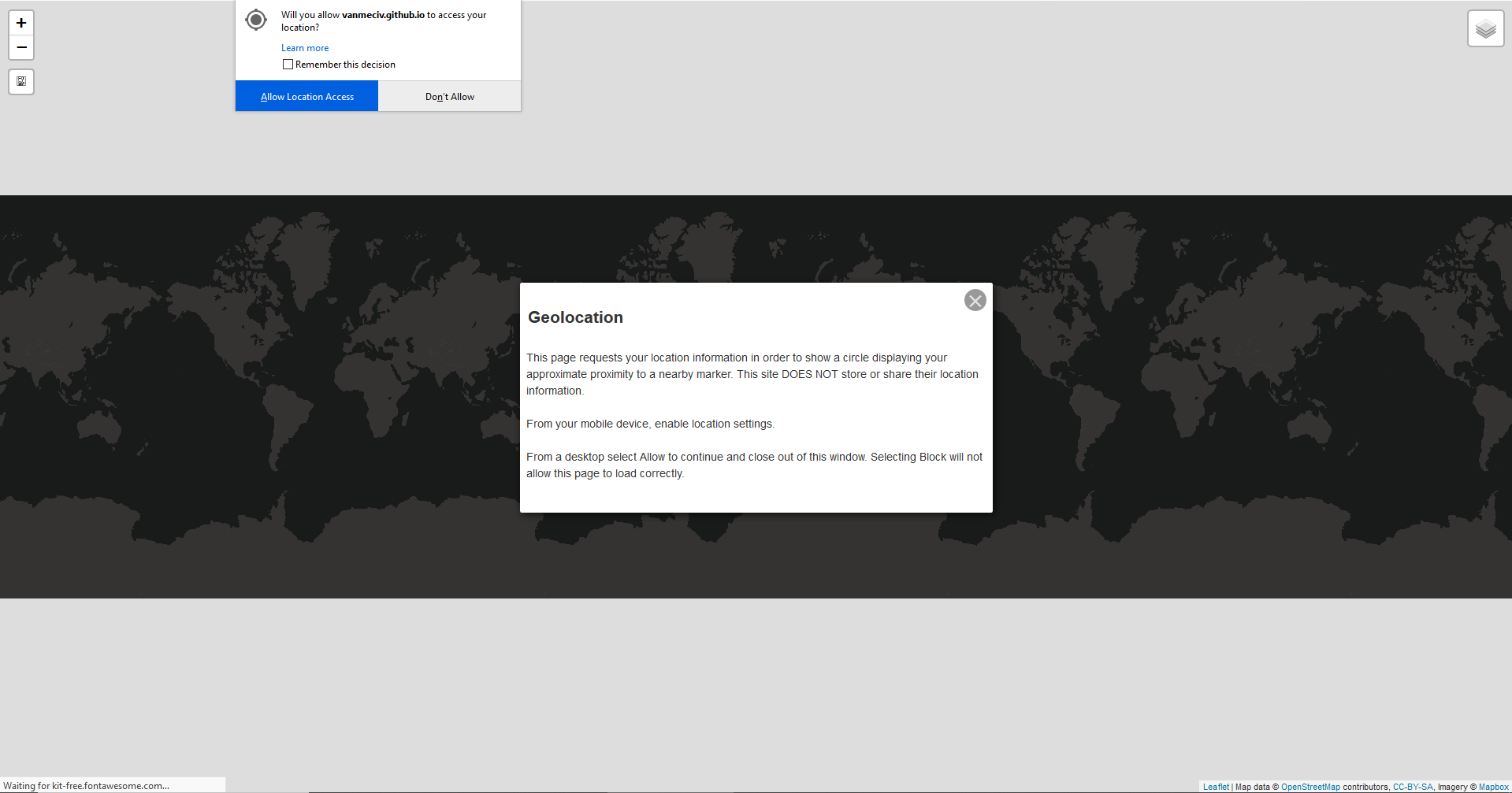
That’s a tricky question, because the heading of 135 or 135° tells us that that the device is moving SE, but that does not tell us the direction the device is facing. For this, we would need other sensors such as an accelerometer.

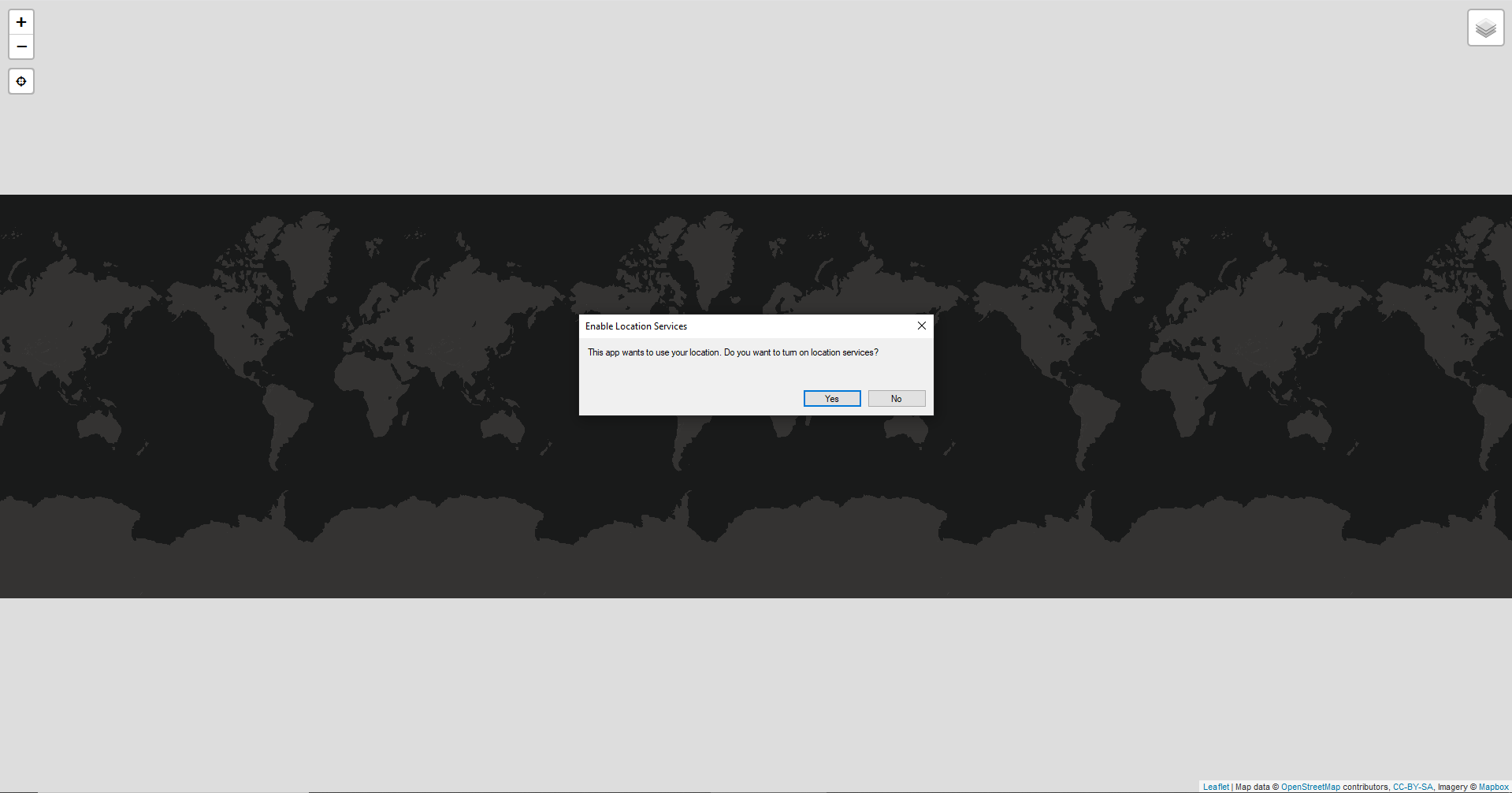
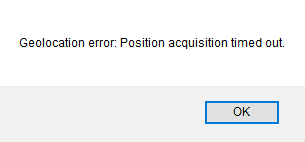
1. Once you've uploaded your final map to GitHub Pages or your UW server space, test the map on two different browsers on your computer and two different browsers on your mobile device. List the browsers you ran your test with and briefly describe the differences in user experience among the various options. Did you run into any bugs while testing? If so, explain.
   1. **Chrome Desktop:**

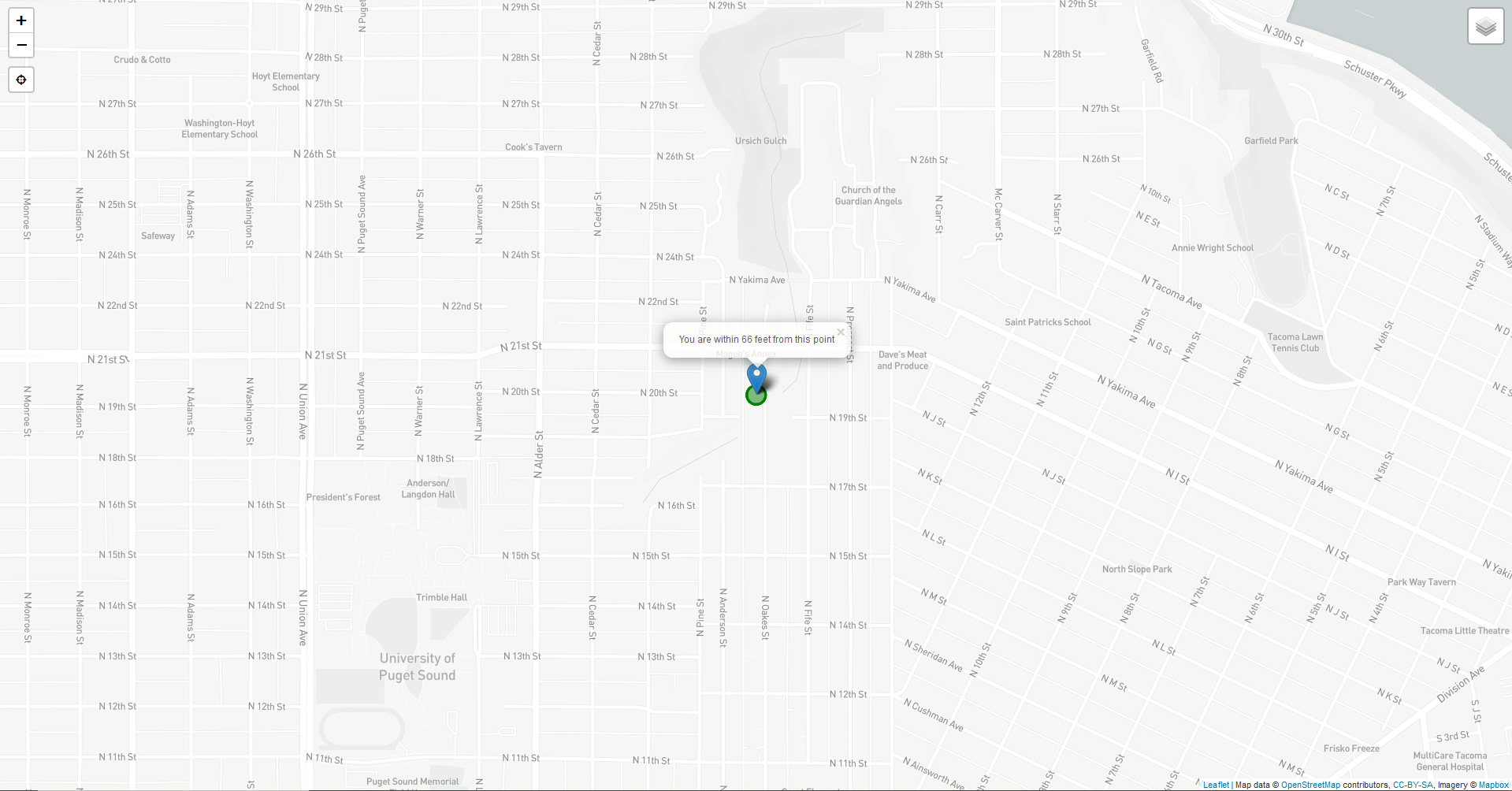




* 1. **Firefox Desktop:**

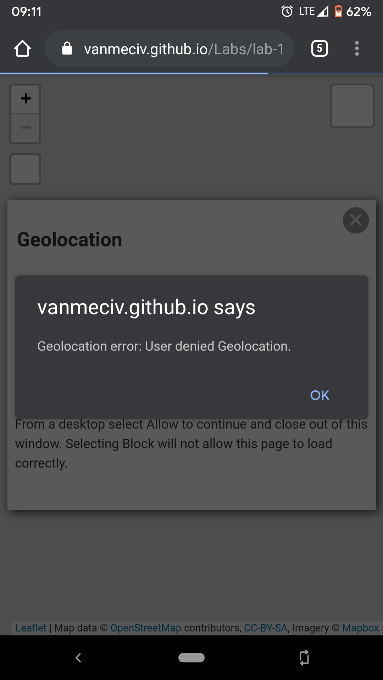


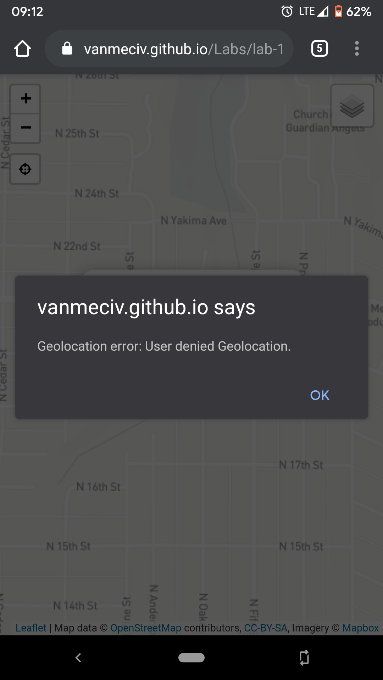
 

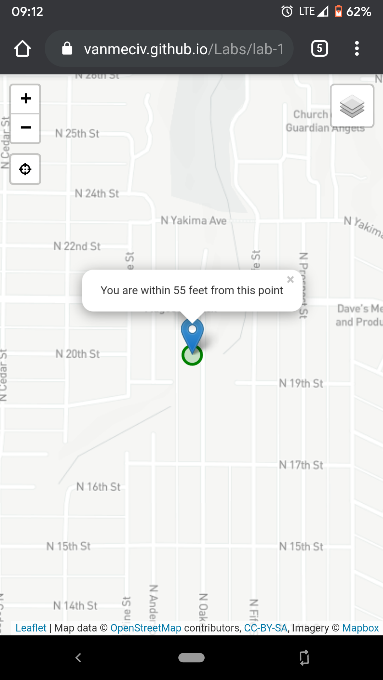


* 1. **Chrome Mobile:**

When first navigating to the web address with phone location settings toggled off:

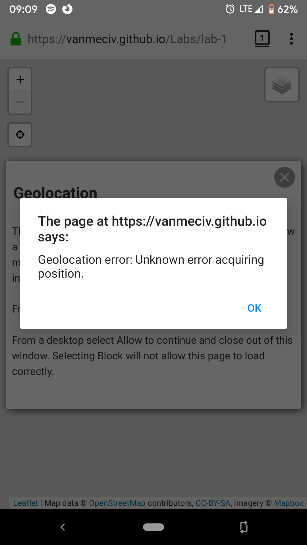
An error alert box is displayed

After the device location settings are toggled on, the browser still requests location with the same alert box.

Once both location settings are allowed, the browser displays the user’s location.

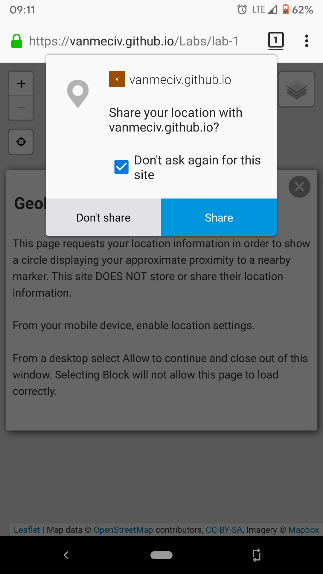
* 1. **Firefox Mobile:**

When first navigating to the web address with phone location settings toggled off:

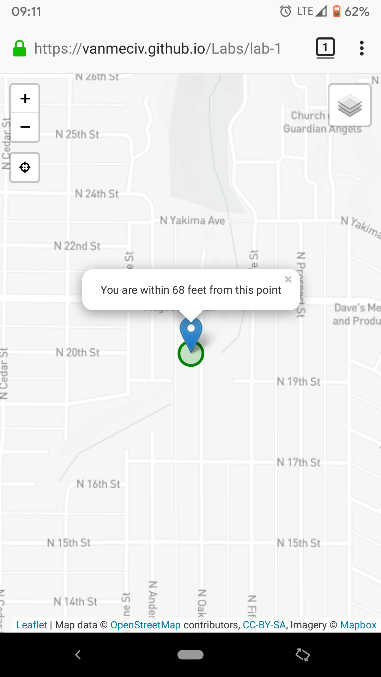


Firefox displays a geolocation error alert box.

Once the device location settings are toggled on: The browser displays a request for location settings.



The browser displays a request for location settings.

Once both location settings are set, the webpage will display the user’s location centered on the page.

I didn’t run into any bugs, per se, but I did notice that Chrome displayed the same error for two different issues, which I believe is something you covered in week 2 class slides with error handling. It would be a good argument for using navigator.geolocation.error as opposed to the onLocationError(e) alert function.