Overview:

My project was essentially going to be a comparing and contrasting of the implementation of the same web app using two separate languages and technologies. The project aimed to established a distinction between the usage of python and javascript to accomplish my goal of implementing a geolocation app. I established that python was much easier to write and read, but javascript had vastly more reliability, flexibility, and accuracy for this task. I also argue the establishment of the modern web language being javascript by illustrating issues that python has with the development of web pages, especially dynamic web pages and complicated routing. I did all of the programming and design after our group kind of fell apart, which is fine but it ended up being quite a bit of work just by myself. But it ended up being an enjoyable experience to work with new frameworks. Screenshots of my code are included and described in the powerpoint I presented, while also being fairly commented and are also included in the github repo.

Integration of course material:

As stated in the above paragraph, this project exists to show a distinction between two languages implementation of the same web app. I used some of the same strategies we used to compare programming languages and applied it to my web apps. I also found it to be incredibly important to create two separate web apps so I could directly compare their structure and implementations like we did in class. The verdict boils down to a direct comparison between python and javascript. Whereas javascript edged python out in terms of reliability and flexibility, being able to generate dynamic web pages and more extensive request payloads. Python though isn’t useless and still has its place on the server side, but is useless in comparison on the client side even if it is very easy to read and write. In the end it almost seemed like I was doing one of the in class assignments on a much grander scale, implementing the same concept in two separate languages and analyzing what made them distinctly different, I think it could be interesting to do more assignments on modern programming languages and their applications in the future.

Correctness:

I don’t have much to say in this regard, you can reference my presentation powerpoints to see videos of my web apps being operated. You may see a couple errors in the console for a few reasons. On the javascript side, the google maps api loads asynchronously in the webpage, and therefore is instantiated with “null” values technically, but is never ran without first getting your latitude and longitude. On the python side there is an unclosed filestream, but I didn’t find it to be impactful on the actual assignment. I also didn’t really show off being able to log the latitude and longitude into a sqlite3 database because in the end, they ended up being roughly the same and didn’t really influence my discussion between python and javascript.

Technical Complexity:

This assignment for me was an incredible pain in the rear, but I thought it was incredibly rewarding to do. One of the main concerns was that the idea itself wasn’t something that had very much support, and I had to read a lot of documentation on several different apis and technologies and stitch them all together. This was especially apparent on the node.js side of the project. I was fairly fluent in javascript being a TA for n201, but trying to get a node app working and installing all of the stuff needed to even get simple routes working was a nightmare. There isn’t an intense amount of code, but this is offset by hours and hours of reading documentation and figuring out how to get all of this stuff I’ve never used before working. One of the large hurdles I encountered was the google maps api which did not want to work with me, I also needed to input a debit card to use the free trial of the google maps api which was annoying. One of the even larger hurdles was the fact that python does not possess a similar supported method to draw google maps, unlike javascript. So I had to scour the internet for a python wrapper that is able to draw google maps, but it had an even larger problem with it not being updated to the current industry standard for google maps. Essentially the wrapper doesn’t support the direct insertion of an html key, so with the help of github, I managed to find a way to use the beautifulsoup html parser to insert my api key into the google map. The project itself in my opinion is super cool and is a great demonstration of the power of modern web programs and applications, while also being incredibly frustrating and time consuming to learn how to do. In the end, I really value the stuff I learned from the project, and I feel as though it was much more enjoyable than taking a regular final.