Alvie Stoddard
April 12, 2020
Problem Solving | Li

MBTA Web App Project

Abstract/Project Overview:

For my project, my goal is to have a relatively good looking, simple website where the user would be greeted with a text box and three options, train, bus, or express-bus, to navigate to find the closes stop fitting their criterial(s). For the project, it will require six basic elements, an input/search box, three checkboxes, a submit button, and finally, Flask code for showing the data on the site. Since I already have experience with checkboxes even sightly beyond WebTech, I do not believe this will be that challenging. If this is as simple as I believe, I would also like to add in an accessibility checkbox as well.

One of the biggest changes I will have to make to my API is figuring out which type of stop a stop is (local bus, express-bus, or train) and then figure out if it is the closest stop and if not going back threw, ignoring that and doing the process over again. The data for stop type is contained in the json response so getting that result should not be too difficult.

Finally, the final requirement is returning the result back to the user. This is not difficult at all as my original project can already accomplish this. However, to add additional complexity and learn more I hope to only use one page in my final project as my current version uses two, one for if there is a result and one if there is not. This I think will not be too difficult to accomplish using some Javascript; however, I will have to figure out how to have the Javascript and Flask data work together.

Discussion/Project Reflection:

After completing this project, it was way more difficult than I expected. After re-reading my Project Overview, I was almost completely wrong about what was going to be difficult and what would be easy. It ended up being that the Checkboxes, as simple as they are probably took me the most time to figure out. Figuring out the behavior of checkboxes alone probably took between 5-7 hours and was the main reason this project was late. (Sorry about that.) What made this so complicated was that I couldn't figure out an error I was getting. I attempted to get the form result of the checkboxes with "find = request.form[variable]" (3 times, once per variable) but this would not work and even after reading the Flask documentation and doing some Stackoverflow research, I couldn't figure out how to get the results of multiple form inputs. Getting just one worked fine however getting multiple continually gave me a server error. After a lot of Googling and several hours, I finally figured out the best way to do it was to use a loop and two arrays. I settled on the following code:

```
checked = []
boxes = ["RapidTransit", "ExpressBus-Downtown", "LocalBus"]
for box in boxes:
    if box in request.form:
        checked.append(box)
```

This is simple code but coming up with it took a very long time and was only something I found after combining some code from a Stackoverflow article (https://stackoverflow.com/questions/53344797/how-create-an-array-with-checkboxes-in-flask) and applying that to what I already knew. After finally getting this to work, I had to now implement it on the backend which had originally believed would be the more difficult portion of the project.

This portion, however, ended up being a lot easier than I expected. I ended up swapping the order, getting the stops that fit the criteria first and then seeing which of those were the closest. Again, using multiple points did increase the difficulty but only slightly. In reality, it wasn't that much more difficult. One unexpected roadblock I hit was with my try/

except block. I was originally using one of these but ended up removing it because it was not providing any value and was making viewing the error more difficult. To get the type of stop it was, I needed to reach into the json data the problem was that for some of the stops, the type of stop (local bus, express-bus, or train) was missing. The json data spit it out anyway just without a “id”. In fact, the “id” data was just completely missing. I ended up having to define a variable that reached only far enough to see if that data returned None (which originally I was wrongly interpreting as a string) and then if it did not return None, I would reach back into the json data again and get the type of stop. Because some of the stops had no actual type (local bus, express-bus, or train) I assumed if the user did not choose an option or chose all three options they would be willing to take the undefined ones as well.

Finally, implementing the Flask data and putting it on just one file, index.html, was actually very simple. I chose to use an empty div and then show the text if a result existed. This was very simple.

From this project, I learned that while something may originally seem like it won’t be that difficult it actually can be way more difficult than expected. The amount of time it took me to understand the behavior of how Flask deals with checked-boxes really slowed me down and I found the documentation to be poor. If I were to continue working on this project, I would like to add accessibility as an option as doing it wouldn’t be that much additional code. I also think there are places in the API where could cut some code out or at least use a few fewer lines by using loops. I think when testing I took all circumstances into account, however, it is hard to tell. My process for unit testing could definitely be improved as every time I wanted to test it, I was continually typing some key locations such as "Boston Common,” “Reggie Lewis Center,” “Fenway Park,” etc. This was mostly done because I kept on thinking I was close to finished with the code but then would hit a problem. The first part of the project I found to be very easy and I think that is partly what caused me to be overly ambitious on the second part but I am proud of what I was able to create. I definitely learned a lot about how the Flask API works, how a problem that looks easy to start can be a lot more difficult, and more.