Spencer Griffin and Emely Cedano
Python Assignment 3
Reflection

Assignment 3: MBTA_Helper App and Webpage Design

In this assignment we were tasked to design an app: MBTA Helper, that essentially allowed users to enter the name of a place so that they could figure out the nearest stop next to them and whether or not it was wheelchair accessible. The app, designed in python through functions such as: get_nearest_station, get_lat_long, and get_near_stop, would allow the app to find the nearest stops and wheelchair accessibility within a specific radius to the place name that the user put in. To get this information, we required access to the MBTA API and Mapquest API so that it we were able to extract the latitude and longitude coordinates from Mapquest and all the stops and wheelchair accessible points that the MBTA website listed. On the other hand, we used HTML to create and display the information being coded in the MBTA_Helper app. This allowed us to render templates and input/result forms where we could specify what the webpage would display via the render templates and the types of inputs and results it could receive and display based on what place name the user input. Overall, this was a useful project as it showed us the combined power of Python and HTML and how they could be fused and intertwined.

Helpful Links used throughout Project:

- https://flask.palletsprojects.com/en/1.1.x/quickstart/
- https://api-v3.mbta.com/docs/swagger/index.html#/Stop/ApiWeb_StopController_index
- https://developer.mapquest.com/documentation/geocoding-api/address/get/
- https://github.com/mbta/api
- https://api-v3.mbta.com/

When it comes to the process, we believe the documents were honestly helpful but not enough to fully understand how to intertwine concepts together to exhibit what we wanted them to exhibit- this is more for the HTML portion of the coding. It was our first time using or tackling HTML so it was a bit difficult to get a handle on the mechanics of it. On the Python side, it was a bit hard at first to understand what parts of the API were necessary in order to retrieve the information we wanted, which as frustrating, but after some toggling we were able to understand how it worked. Additionally, for unit-testing and making sure the code worked, we had to do a deep dive into the JSON, making sure we understand the right parameters for how it was receiving place_name and what it considered as an appropriate name in order to return the close appropriate stations. Things as specific as a place_name= Arlington/Arlington st. was more accurate than simply inputting Arlington. We hope to use this moving forward in our jobs. Both of us are interested in data-analytics but my job (Emely) requires a lot of knowledge on all types of code (i.e. SQL, Python, R, HTML, C++ …) and this will help me be able to understand how two different languages speak to each other in order to be able to use both if needed. Spencer is also interested in Data-Analytics and so I'm sure this will help him in being able to explain both code types and their combined functionalities in interviews, work projects, etc. We both wish we knew more HTML before starting this project.

In terms of our team process we first started by divvying up the code where I did more of the HTML portion and Spencer and I worked on both the Python portion (more Spencer here). We then convened and pair-programmed the app and HTML templates as it was easier this way to rely on the other for help and questions, misunderstanding, and to understand how to build in the HTML into the Python and vice versa. We planned to do it this way and this is exactly how it turned out. We are both pretty communicative and considerate so we didn't run into any issues when it came to the amount of work one did vs. the other. When there was an issue or confusion, we asked each other and tried to help each other in that sense. What we would do differently is probably look at youtube tutorials to help us get the gist of how things run on HTML rather than spending so much time reading documents.