Daniel James Rodgers

Assignment 3 (MBTA) Summary

Our assignment was to create a webapp that took a location input, and outputted the nearest MBTA stop. In order to achieve this, we utilised MapQuest and MBTA APIs in Python to find the nearest stop, and outputted that on a webapp built in Python and using HTML.

I made good use of the framework that Professor Li kindly provided, and the first step was to get and interpret a JSON object from a MapQuest API with the latitude and longitude for a given point. An initial problem was that the API didn't accept spaces in the input, which I fixed with:

```
place_name = place_name.replace(" ", "%20")
```

Then, the API kept returning streets in the wrong city, so I edited that line to include:

```
place_name = place_name.replace(" ", "%20") + ",%20Boston,%20MA"
```

Then, I returned the latitude/longitude values in a tuple. Next, I used a MBTA API to return the name and wheelchair accessibility of the nearest stop from the given coordinates. I then took that data and defined a function which returned it as a sentence:

```
f"The nearest station to {place_name} is {response_data[0]}, and it is
{accessible_info} via wheelchair"
```

I then built the app using Flask. First I created an index page in HTML which used the POST request method to allow the user to submit the location they were searching for, and redirect the user to the HTML page containing the desired output. Due to needing to retain information between pages, I stored the location in session["place"], which also necessitated the use of a secret key. If the search was unsuccessful, the user was directed to a pre-written HTML error page prompting them to try again. If the search was successful, the user was directed to a HTML page displaying the information from MapQuest/MBTA as formatted in the previously mentioned code.

I enjoyed this project very much, though as I am new to Flask/HTML I had to refer frequently to the Flask documentation for even basic tasks e.g. how to access form data from another page. Also, due to time constraints (entirely self-inflicted), I left some untidy code such as:

```
if latLng["lat"] == 42.52277 and latLng["lng"] == -70.91888:
        return "n/a", "n/a"
```

This code was necessary as after I added `",%20Boston,%20MA"` to the MapQuest query, even if I entered garbage input then MapQuest returned a default Boston latLng of (42.52277, -70.91888), and not the desired error. However, this is an extremely inelegant solution that needs rectifying, probably by revisiting the `",%20Boston,%20MA″` addition. Although I didn't explore it, this project has plenty of stretch goals, including listing multiple nearby stops, displaying the type of stop (e.g. bus, T), and making the webapp more interactive (e.g. a clickable map). Self-isolation may prove a great opportunity to further build on these ideas...

...after the exam/final project.