Project Report

A. Environment Setup

This application requires installation of the following libraries:

1. pandas (Version: 2.1.3)
    install the library using pip in terminal:
    pip install pandas
2. folium (Version: 0.14.0)
    install the library using pip in terminal:
    pip install folium
3. streamlit-folium (Version: 0.16.0)
    install the library using pip in terminal:
    pip install streamlit-folium
4. requests-mock (Version: 1.11.0)
    install the library using pip in terminal:
    pip install requests-mock
5. streamlit (Version: 1.28.2)
    install the library using pip in terminal:
    pip install streamlit
6. requests (Version: 2.31.0)
    install the library using pip in terminal:
    pip install requests

After these libraries are installed, this application is supposed to be able to run successfully. To view this Streamlit app on a browser, run it with the following command:

streamlit run ./app.py

B. Code Highlights

1. One of the highlights of my code is the logic structure in the function map_render(). In this function, there are many nested and parallel if-statements deciding whether to display any certain features. I managed to stay organized and distinguish between different cases when different features take place.

```
39 ›    if "library" not in st.session_state:…
43      library_option = inp.display_selectbox("Where do you want to go?",
44                                             st.session_state["library"])
45      # By default, the map including all libraries and stations is displayed,
46      # because by default library_option is None
47      if library_option is None:
48 ›        if "station" not in st.session_state:…
52          out.display_map_with_all_libraries(st.session_state["station"],
53                                             st.session_state["library"])
54      # If the user selects from the selectbox, the first type of map
55      # including only one library and its nearby stations is displayed
56      else:
57          # If the user made their choice, more information about this
58          # specific library is given
59          add_button = inp.display_button("Add into want-to-go list")
60          # The user uses the add button to add any libraries
61          # into their fav list
62          if add_button is True:
63 ›            if library_option not in st.session_state["favourite list"]:…
66 ›            else:…
68          # Find nearby stations of the chosen library
69
70          library_option.find_nearby_stations(st.session_state["station"])
71
72 ›        if library_option.nearby_stations == []:…
75 ›        else:…
```

2. Another highlight is my methods in Library class and Station class. I managed to connect two classes with each other through coordinates.

In the class of Station, the calculate_distance_from_library() method calculates the distance between a station object and a library object by coordinates.

```
69      def calculate_distance_from_library(self, library) -> float:
70 ›        '''…
79          # In order to handle errors and prevent circular import, and
80          # since this method has one possible error, I choose not to
81          # use isinstance() to raise the AttributeError but try-except
82          try:
83              difference_lat_kilometer = (
84                  library.latitude - self.latitude) * CONVERSION_FACTOR
85              difference_lon_kilometer = (
86                  library.longitude - self.longitude) * CONVERSION_FACTOR
87              distance = (difference_lat_kilometer ** 2 +
88                          difference_lon_kilometer ** 2) ** (1/2)
89              return round(distance, 2)
```

In the class of Library, the method find_nearby_stations() calls the station method of calculating distance to find the stations close to the certain library.

```
77      def find_nearby_stations(self, station_list: list):
78 ›        '''…
87          # In order to prevent circular import (import Library in Station and
88          # import Station in Library), I choose not to use isinstance() to
89          # raise the AttributeError but try-except
90          try:
91 ›            if not isinstance(station_list, list):…
94              nearby_station_list = list()
95              for station in station_list:
96                  distance = station.calculate_distance_from_library(self)
97                  if distance <= THRESHOLD_KILOMETER:
98                      nearby_station_list.append(station)
99              self.nearby_stations = nearby_station_list
```

C. Next Steps

1. Data Display Page:
    a) enables users to search libraries or stations
    b) enables users to sort libraries with a certain neighbourhood (there is a key "geological area" in the original API which could be added as an attribute for the benefit of this feature)
    c) enables users to see a list of nearby stations in the library table
    d) displays maps with barely libraries or stations
2. Favourite List Page:
    a) displays all information (coordinates, address, url, and perhaps geological area) of the favourite libraries
    b) enables users to write reviews for libraries (and store these in a session state)
    c) displays a map with barely the favourite libraries (and makes the markers representing these favourite libraries distinctive in the Maps page when displaying the map with all libraries and stations)

D. Reflection

1. What did I learn from this project?

a) I think my biggest takeaway for this project is to allocate my time before working on this project. If I did not make the plan to finish my most challenging feature in the first week, I would not have enough time after Milestone 1 to refactor my code.

b) Speaking of refactoring, I learned from this project that it is an equally important part as writing codes. After implementing the MVC design pattern and organizing my files in such a hierarchy, I found that it is even easier for myself to find different functions.

c) I also learned that it is always important to build the skeleton first, especially when working on such a big (for me, for now, it indeed is) project. Also, it might be a good choice to finish the most difficult part in the beginning, then it will be much more easier to get the rest of things done. In this project, I decided to deal with the map feature first. I did not know how to build a map and did not even know that the folium library would help. Consequently, I felt stressed when I started working on this project. It was a relief for me when I managed to have a map of all libraries in the third day, then everything went smoothly in my eyes, because they are at least not as seemingly impossible as the map was.

2. What was the most challenging part?

The most challenging part for me is definitely finding the appropriate API. I faced various kinds of barriers in this process. I gave up on doing what I had planned to do in the very beginning (it was a favourite book list creator similar to Goodreads)because the API is out-dated and no longer functions. Then, I browsed different APIs, only to find that most of them were either too complex to be implemented in my project or too easy to dig into more and find their connections with other data sources. Fortunately, in the end I found my current APIs. It turns out that these two data sources are closely interconnected.

3. What was the most rewarding part?

The most rewarding part was that I learned how to create streamlit pages and folium maps through reading the tutorials. Words failed me when I finally managed to see the Vancouver map with several dots (which represent libraries at that time) displayed in front of my eyes. I found that compared with working with bare codes in progress, I am more interested in the outcome, how the result is to be displayed on my screen.

4. What would you do differently next time?

a) Next time working on such projects, I would implement MVC pattern in the beginning, which will surely save me much time refactoring.

b) I would deal with the models first. Only if I understand what my data on hand is capable to do could I figure out what features I can make with it.

c) I would write test classes as soon as I finish a class. This time, I choose to writing my test classes days after I finished my classes, which took me quite some time to refresh my memory.

d) By saying “descriptive names”, I think it means that each variable’s name needs to be distinctive and characteristic as well. I mistook some variable names with others in this project, which caused me lots of trouble.