Winter 2021 SI507 Final Project uniquename: ptkuo, UMID: 02569460

A. **Project code**

- **GitHub link**: *https://github.com/ptkuo0322/FinalProj_SI507.git*
- **README**:

# Project Overview

This is the final project for SI507 at University of Michigan for Winter2021. It is a project related to python, SQL, web crawling, and user interactive program. According to the users' input of any type of food or name of restaurants or location of restaurants, this program will search results along with different features, such as ratings, reviews, address…etc. The results will be represented in the following form:

- Command line prompt
- YouTube videos
- Figures (scatter charts, bar charts, and radar charts)
- Google map (static/dynamic map)

In addition, simultaneously, users are allowed to construct and expand their own SQL database, which allows them to either find or compare their desired restaurants and food at their ease.

# Required API

This program requires four kinds of API keys

1. Yelp fusion API
   a. Sign up at *shorturl.at/jHM24*
   b. Create new App, obtain personal private API Key and authenticate API calls with the API key.
2. YouTube API
   a. Sing up at *shorturl.at/oxBOS*
   b. Create new Credentials for API keys and Enable APIs and services for YouTube Data API v3
3. Google static map API and Google Directions API
   a. Sing up at *shorturl.at/oxBOS*
   b. Enabling Billing payment (free trial for three months) #Remember to cancel after the trial
   c. Create new Credentials for API keys and Enable APIs and services for Maps Static API and Direction API

# How to Run the Program

There are two following ways to interact with this program:

1. Instant search on Yelp for specific type of food or restaurant
   a. Users are required to type in the FoodName/RestaurantName (optional) and Location(required).
   b. Users would see the search results (name/phone/address/rating/...) in command line prompt/scattering plot.
   c. Users could decide to whether to open the relative videos on YouTube, mark the relative location of each restaurant on static google map and navigate themselves to chosen restaurant on google map.
   d. After finishing, program will be back to choose option and not to stopped until users enter "exit"
2. Search the Restaurants existing in the database based on Reviews/Rating/Price
   a. Users are required enter the state of USA in abbreviation.
   b. Users are required to decide the filtering conditions, such as rating/review price/order...etc

c. Users could see the result in command line prompt, bar plots and radar charts.
d. After finishing, program will be back to choose option and not to stopped until users enter "exit"

# Optional DB Browser for SQLite

It is additional tools for user to check the data in SQL database.

1. Downlink: *shorturl.at/xLRU1*
2. Documentation link for tutorial: *shorturl.at/dlqtQ*

- **Required Python Packages**
  1. sqlite3
  2. requests
  3. pandas
  4. plotly
  5. googleapiclient.discovery

B. Data Source

- **YELP fusion API**
  - Original URL: https://api.yelp.com/v3/businesses/search
  - Documentation: shorturl.at/loHP7
  - Data format: JSON
  - How to access the data
    - Ask user to key in the **RESTARUANT/FOOD** and **LOCATION** as the information.
    - Pass the user's input with the API_key as the query term.
    - Use HTTP. Request to retrieve the data from Yelp Fusion website.
    - Parsed the searched result and save as json file (for cache) and csv file (for SQL database).
  - Summary of data
    - # records available > 2500 (estimated)
    - # records retrieved > 1000 (estimated)
    - Name/phone/address/price will be retrieved to give the direct information about the restaurant. ReviewCount/rating will show the popularity. Latitude/longitude will show the relative location and the direction on maps. State shows where the restaurant locates. RefId is the element used to identify the information in the SQL database.
  - Evidence of caching
    - shorturl.at/yCHZ4
    - shorturl.at/lpvHO
- **YouTube API**
  - Original URL: https://www.googleapis.com/youtube/v3/videos
  - Documentation: shorturl.at/kAIUY
  - Data format: JSON
  - How to access the data
    - Ask the users which searched restaurant they are interested in?
    - Pull out the restaurant name from the cache created from YELP query step and pass it as the query term to search on YouTube.
    - Parse the searched result and save as json file and to show the related videos on web browser.
  - Summary of data
    - # records available > 2500 (estimated)

        - # records retrieved > 1000 (estimated)
        - Several central component of URL links of restaurants will be retrieved here. It will be first re-constructed in a specific format and use to search the videos on YouTube to give a glimpse of the restaurant.
    - Evidence of caching
        - shorturl.at/yCHZ4 (This one uses the cache created in Yelp query step)
        - shorturl.at/lpvHO (This one uses the cache created in Yelp query step)
- **Google static map API**
    - Original URL: https://maps.googleapis.com/maps/api/staticmap
    - Documentation: shorturl.at/nE368
    - Data format: JSON
    - How to access the data
        - Read the cache file created by the YELP query step to get the latitude/longitude.
        - Re-construct the information and use as query key for static map API.
        - Pass to static map API to get the location-marked static map on web browser.
    - Summary of data
        - # records available > 2500 (estimated)
        - # records retrieved > 1000 (estimated)
        - The latitude and longitude attributes in the cache created by YELP query step will used to mark the relative locations of restaurants. Users are able to know the distance between each two of them. (for better arranging their time and schedule)
    - Evidence of caching
        - shorturl.at/yCHZ4 (This one uses the cache created in Yelp query step)
        - shorturl.at/lpvHO (This one uses the cache created in Yelp query step)
- **Google Directions API**
    - Original URL: https://www.google.com/maps/search/?api=1
    - Documentation: shorturl.at/kKLWX
    - Data format: JSON
    - How to access the data
        - Read the cache file created by the YELP query step to get the latitude/longitude.
        - Re-construct the information and use as query key for map API.
        - Pass to map API and launch google map direction on the web browser.
    - Summary of data
        - # records available > 2500 (estimated)
        - # records retrieved > 1000 (estimated)
        - The latitude and longitude attributes in the cache created by YELP search step will used to show the direction on google map. Users are able to know how to get there and additional information on google maps.
    - Evidence of caching
        - shorturl.at/yCHZ4 (This one uses the cache created in Yelp query step)
        - shorturl.at/lpvHO (This one uses the cache created in Yelp query step)

C. Database

- **Database schema**
    - Basically, there are two tables, FOOD_RESULT and RESTARUANT_RESULT in this database.

- Fields [type] are RestaurantName [text], PhoneNumber [text], ReviewCount [real], Rating[real] , Price [text], Address [text], Longitude[text], Latitude[text], UrlLink[text], State[text], RefId [text][Primary Key].
- The difference between two table would be the number of search results since one of the tables is the search result of only specifying the location.

- **Foreign key-Primary key**
  - The filed RefId is used as "logical foreign key" – "logical primary key" within two table. In order to increase the efficiency as mentioned in the slack, I did not additionally specify new key pairs.
- **Screenshot of the Table in Database**
  - shorturl.at/zDO29 (Evidence of database schema shows here)
  - shorturl.at/mwHIY (Evidence of the table, FOOD_RESULT)
  - shorturl.at/lxzU5 (Evidence of the table, RESTARUANT_ RESULT)

D. **Interaction and Presentation Options**

- Step 1: The program will present the brief instruction of the search options for users and they need to decide which one do they want to use.
  - Option1: Instant search on Yelp for specific type of food or restaurant
  - Option 2: Search the Restaurants existing in the database based on Reviews/Rating/Price
- Option 1
  - Step 2: users are asked to type in two information: food/restaurant<optional>, and location<required>.
  - Step 3: After pass the information through Yelp API, the search results along with their features will be presented either in formatted text description *[command line prompt]* or figure illustrations *[scattering charts, plotly]*.
  - Step 4: Users will be asked whether to see the related videos of their desired restaurants on YouTube. If yes, the program will show the videos on the web browser *[video format, webbrowser]*. If no, they can just skip the step. (videos might not related to the restaurants since that the same name might have been used by others)
  - Step 5: The program will show 10 restaurants in a group and presented by formatted text description *[command line prompt]*, and then users will be asked whether to see their relative locations? If yes, then it will show the restaurants with different marks in the same static google map *[Figure format, webbrowser and static map API]*. If no, then, they can skip this group and move to next group until the end of this step.
  - Step 6: Next, the program would as the users whether to launch google map direction to their desired restaurant. If yes, after pick up the name of the restaurant, it will launch the navigation on google map *[web page format, webbrowser and Google Directions API]*. If no, it will skip the step.
  - Step 7: The program will go back to the beginning step. Users can decide whether to choose Option1 or exit. It will not be stopped until users enter "exit".
- Option 2
  - Step 2: Users are asked to decide the filtering condition, such as filter the database by rating and sort the result by price.
  - Step 3: Based on the users' filtering condition, the program will keep asking the desired standard scale of their choice, such as 1-5 for rating scale.
  - Step 4: Users will be asked to enter the sorted order, either in ascending order or descending order.
  - Step 5: Users will be asked to enter how many results do they want, range from 1-50.

- Step 6: Users will be asked which state of USA of the restaurants do they want to search. Users need to enter the abbreviation of the state in USA.
- Step 7: the result will be presented in either formatted text description *[command line prompt]* or figure illustrations *[radar charts and bar charts, plotly]*.
- Step 8: The program will go back to the beginning step. Users can decide whether to choose Option1 or exit. It will not be stopped until users enter "exit".

Note: Interactive and presentation technologies are denoted by braces [Presented format, python package] with Italic Type in color green

E. Demo Link

https://drive.google.com/file/d/1-wiBm9GqF-Dg5IEHdssGRNYvD5apy8w4/view?usp=sharing