1. Project code

1-1 Link: https://github.com/szlee0508/FinalProject.git

1-2 README

**If you’d like to see further information, please go to final\_project file.**

1-2-1.API Application:

Yelp API & The Movie Database(TMD) API

Please put your API key in “secret.py” file, the format of API Key will be like

API\_KEY=””(This is for Yelp)

﻿client\_ID =””(This is for Yelp)

﻿API\_KEY\_M=””(This is for TMD)

﻿API\_token\_M=””(This is for TMD)

1-2-2. Code structure:

final\_proj.py

In the first part, I created an database and table for both Yelp and The Movie Database. There are six fields in Restaurant table, including Id, Name, Rating, Address, Phone and Res\_id; there are eight columns in Movie table, including Id, Title, Popularity, Release\_Date, Revenue, Run\_time, Vote\_count and Vote\_avg.

Secondly, user can choose which information they want to know, restaurants or movie. If they type restaurant, it will connect to Restaurant table in yelp\_db database. I applied cache and infoYelp() to retrieve data from Yelp based on a city and a category which entering by user. After typing a city and a category, it will return top 5 restaurant which contain name, phone, address and rating information based on their search word.

On the other hand, if they type movie for the first interactive step, it will connect to Movie table in movie\_db database. The user can enter the presentation of the data they want to see, the dataframe or bar graph, they can also choose how to order the data.

1-3 Python packages

Graphical user interface, text, application

Description automatically generated

2. Data Source

1. **Yelp (Basic Information)**

(<https://www.yelp.com/developers/documentation/v3/business_search>)

Base URL: <https://api.yelp.com/v3/businesses/search>

I accessed Yelp APIs to cache some basic information of the restaurants. The screenshots below are the code I used. The format of the file I cached from Yelp API is json. I applied **make\_request\_with\_cache** function to cache it and save it as JSON file. The first screenshot below is the function I define to scrape date and the second is the Yelp json file. There are six fields in my Yelp json file, including Id, Name, Rating, Address, Phone and Res\_id.

**2. The Movie Database** (https://www.themoviedb.org)

Base URL: ﻿ ﻿<https://api.themoviedb.org/3/discover/movie>

Detail URL: [https://api.themoviedb.org/3/movie/{movie\_id}?api\_key=<<api\_key](https://api.themoviedb.org/3/movie/%7bmovie_id%7d?api_key=%3c%3capi_key)>>

Firstly, I used the Base URL to grab the movie id, after that I applied this id in Detail URL. Since I want more detail of the movie like run time etc., I need to extract the data I needed from two URL. The code I used is display in the below screenshot. I also use **make\_request\_with\_cache** function to cache it. The format is also json. The third picture is the function I used to extract data from The Movie Database API. The fourth is the movie json file, there include Id, Title, Popularity, Release\_Date, Revenue, Run\_time, Vote\_count and Vote\_avg.

Text, chat or text message

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

3. Database

3-1. Database Schema

Yelp:

Text

Description automatically generated

Text

Description automatically generated

The Movie Database

Text

Description automatically generated

Text

Description automatically generated

3-2. Foreign Key

Since two of my data is unrelated, there are no foreign key between my tables.

3-3 Screenshot of my tables

Yelp:

Graphical user interface, table

Description automatically generated

The Movie Database:

Table

Description automatically generated

4. Interaction and Presentation Options

4-1 Description

The program has two main functions, including finding restaurants and finding movie by the order I choose. For all questions, once the user types unrelated word, it will return the error message. I used command line prompts and matplotlib to do the interaction and presentation.

1). Finding restaurant

* For the first part, the program displays an interactive command, which asks user to choose the information he/she wants to know.

**What are you looking for? (e.g. Enter Restaurant, Movie) or "exit":**

* Secondly, if the user chooses <restaurant> in step 1, the program displays an interactive command, which asks user to type in the city and category in interest. If the use types ‘exit’, then it will go back to the last step.

﻿**What city of the restaurant are you searching for**:

﻿**What kind of category are you looking for**:

* Finally, it will display the name, phone, address, rating of top5 restaurants based on the place and category.

2). Finding movie

* For the first part, the program displays an interactive command, which asks user to choose the information he/she wants to know.

**What are you looking for? (e.g. Enter Restaurant, Movie) or "exit":**

* Secondly, if the user chooses <movie> in step 1, the program displays an interactive command, which asks user to type which format they want to display the data. If the use types ‘exit’, then it will go back to the last step.

﻿**What way do you prefer to read movie data? (e.g graph or data):**

* Third, if the user chooses data in step2, the program displays an interactive command, which asks user how to order their data. On the other hand, if the user choose graph, it will also display a same question for asking how to sort their data. If the use types ‘exit’, then it will go back to the last step.

**Which way do you want to sort your data(e.g popularity, count, avg):**

* Finally, it will display the data of title, vote average scores (popularity scores, vote count scores) based on their choice, released data and run time of the 5 movies if they choose to display their information by data. Otherwise, it will display in a bar chart and sort the order by their choice.