

Final Project

Project code

- Github repo link: <https://github.com/Huadous/final-project>
- README

This program mainly uses the API provided by Yelp Fusion. The **API verification key** is also very easy to apply, please apply through this link: <https://www.yelp.com/fusion>. Sign up for an account and manage apps on the website which will contain an API Key. You need to create a file named `secrets.py` and copy this variable into the file `API_KEY = 'your_api_key'`.

The entire app is built within the **flask** framework, and the human-computer interaction is relatively simple. The main thing is to use the **three drop-down menus** to select the state, the city in state, and the available restaurant categories. After selecting the state and city, statistical information about different categories will be displayed. Then you can choose your favorite restaurant category based on them. At this time, the information of the restaurants in that category will be displayed in the form of a table, or you can click the **Map button** to display the location of the restaurants. In the table, you can click the **"Click here" button** to enter the detailed information page of the restaurant.

- Required Python packages

requests, plotly, folium, pandas, flask and bs4.

Data sources

Categories information

- Origin: [Documentation](#) [Download](#) Format: JSON(> 1000 records)[\approx 1500]{192 used}
- Data access and caching: downloaded directly without additional verification methods, I used cache.
- Summary of data: It contains information about categories and available countries. Then, this data can be used as a benchmark for restaurant category search. Because this file contains all the categories. What I need to do is to filter out the category of restaurants from all categories.
- Important fields: **"alias"**: alias of the child category, offer a different name for title. **"title"**: title of the child category and will be used to find different type of restaurant. **"parents"**: belongs to what parent category. **"country_whitelist"**: available countries (without this field means **TO ALL THE COUNTRIES**).

ISO 3166-1 alpha-2 code

- Origin: [Documentation](#) [Download](#) Format: JSON(< 1000 records)[\approx 250]{250 used}
- Data access and caching: Downloaded directly without additional verification methods, I used cache.
- Summary of data: Because in the previous category file, there are information about different restaurant categories in which countries provide search services. Therefore, it is necessary to use the abbreviations of the names of each country in this file to determine whether this category can be searched in the US.
- Important fields: **"Code"**: code of the country and improve its readability by providing the full name of the

country. **"Name"**: name of the country, which is better for human reading.

United States Cities Database

- **Origin:** [Documentation](#) **Format:** CSV(> 1000 records)[≈ 28000]{28399 used}
- **Data access and caching:** downloaded directly without additional verification methods. I used cache.
- **Summary of data:** The main usage of this data source is to provide an effective state-city relationship for the flask app. What's more, this data source have a very useful definition for city, city id, state and state id. I can use it as a mark for each city and state.
- **Important fields:** **"city"**: name of the city. **"city_ascii"**: ascii version. This is more general, and I will use it as each city's name. **"state_id"**: abbreviation for state. It is more convenient as a mark of the state. **"state_name"**: full state name. **"id"**: unique id for each city, which can be the primary key for each city in the database.

Using API key to get base information and do analysis

- **Origin:** [Documentation](#) **Format:** JSON(> 1000 records)[$\approx \infty$]{[50, 1000] for each type will be used}
`GET https://api.yelp.com/v3/businesses/search` (Each request can only get up to 50 results. you still can only get up to 1000 results using multiple queries and combinations of the "limit" and "offset" parameters)
- **Data access and caching** : The Yelp Fusion API uses private key to authenticate. I used cache.
- **Summary of data** : What I'm trying to get from this API is the data of different categories of restaurants. Each request can get up to 50 results. In order to make the flask app faster, I decided to let each type of category of the restaurant only gets 50 records at most to draw the average rating bar plot. There are approximately 200 types of restaurants available in yelp in the US. Then, each plot needs nearly 10000 records of restaurants(The restaurant may not be completely unique, because the restaurant may have more than one category)
- **Important fields in "businesses"**: **"categories"**: List of category title and alias pairs associated with this business. **"id"**: Unique Yelp ID of this business. Example: `'4kMBvIEWPxWkWKFN__8SxQ'`. **"name"**: Name of this business. **"rating"**: Rating for this business. **"coordinates"**: Coordinates of this business.

Crawling and scraping multiple pages in Yelp to gain information related covid-19

- **Origin:** <https://www.yelp.com/> **Format:** HTML[$\approx \infty$]{[50, 1000] for each type will be used}
- **Data access and caching:** By crawling and scraping. I used cache.
- **Summary of data:** This part is not fixed, each restaurant has its own services dealing with covid-19. There are some basic services provided by yelp. But, the user and the owner of the restaurant can change the information on the webpage. I will get all of them from the site and provide it in my flask app to the users.
- **Important attributes:** **"Updated Services"**: some basic services the restaurant can provide to the customer. **"Health & Safety Measures"**: what the health & safety measures the restaurant has implemented.

Database

Database schema

```
1 CREATE TABLE IF NOT EXISTS restaurant_category_information(  
2   "title" TEXT NOT NULL, # name of the category  
3   "alias" TEXT NOT NULL, # alias of the name  
4   "country_whitelist" TEXT # Which countries offer searches in this category);
```

It contains all the restaurant categories (not all the categories provided by yelp fusion)

```
1 CREATE TABLE IF NOT EXISTS iso_3166_1_alpha_2_code(  
2   "Code" TEXT NOT NULL, # ISO 3166-1 alpha-2 code  
3   "Name" TEXT NOT NULL, # English short name officially used by the ISO 3166  
4   PRIMARY KEY("Code"));
```

ISO 3166-1 alpha-2 codes are two-letter country codes defined in ISO 3166-1, part of the ISO 3166 standard published by the International Organization for Standardization (ISO), to represent countries, dependent territories, and special areas of geographical interest. This form is suitable for filtering which categories are available in which countries (although my program is only used in the United States).

```
1 CREATE TABLE IF NOT EXISTS us_states(  
2   "city" TEXT, # name of the city  
3   "city_ascii" TEXT, # ascii of the name of the city  
4   "state_id" TEXT, # alpha2 of the state (`'NY'` for New York)  
5   "state_name" TEXT, # name of the state (New York)  
6   "county_fips" TEXT,  
7   "county_name" TEXT,  
8   "lat" REAL,  
9   "lng" REAL,  
10  "population" INTEGER,  
11  "density" INTEGER,  
12  "source" TEXT,  
13  "military" INTEGER,  
14  "incorporated" INTEGER,  
15  "timezone" TEXT,  
16  "ranking" INTEGER,  
17  "zips" TEXT,  
18  "id" TEXT, # unique id for each city  
19  PRIMARY KEY("id"));
```

The purpose of this form is to help users select a certain city in a certain state, and then facilitate the flask application to continue to deal with subsequent statistical problems of different categories of restaurant ratings. You can see it from the pic in the *Interaction and Presentation Plans* part. Because location is needed when searching, all the location is associated with the search record table (restaurant_category_fetch).

```
1 CREATE TABLE IF NOT EXISTS restaurant_information(  
2   "id" TEXT, # id of each restaurant  
3   "alias" TEXT, # alias of the restaurant  
4   "name" TEXT, # name of the restaurant  
5   "image_url" TEXT, # url of the image of the restaurant  
6   "is_closed" INTEGER, # whether is closed  
7   "url" TEXT, # url in yelp  
8   "review_count" INTEGER,  
9   "categories" TEXT,  
10  "rating" REAL,  
11  "coordinates_latitude" REAL,  
12  "coordinates_longitude" REAL,  
13  "transactions" TEXT,  
14  "price" TEXT,  
15  "location" TEXT,  
16  "phone" TEXT,  
17  "display_phone" TEXT,  
18  PRIMARY KEY("id"));
```

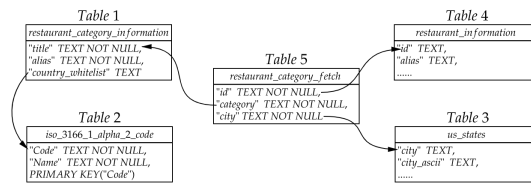
This table is used to record restaurant information. Through the search api provided by yelp fusion, information related to the search results can be collected. I will store the restaurant information obtained by each search in the database, because restaurant information is not frequently updated data.

```
1 CREATE TABLE IF NOT EXISTS restaurant_category_fetch (  
2   "id" TEXT NOT NULL, # id of restaurant(the same as restaurant_information.id)  
3   "category" TEXT NOT NULL, # searched category(the same as us_states.city_ascii)  
4   "city" TEXT NOT NULL # searched city(the same as  
   restaurant_category_information.title));
```

This is a table that records each search, which contains necessary information including location, category, and returned shop id. They are respectively associated with the three tables, please see the follow-up for details. This table may change in the future, or there is another table to filter the information of the categories of valid stores.

Foreign key-primary key relations

- ```
1 1. restaurant_category_information.country_whitelist = iso_3166_1_alpha_2_code.Code
2 2. restaurant_category_fetch.id = restaurant_information.id
3 3. restaurant_category_fetch.city = us_states.city_ascii
4 4. restaurant_category_fetch.category = restaurant_category_information.title
```



### Screenshots of the data

- Table 1 : restaurant\_category\_information

| id | title        | alias       | country_whitelist |
|----|--------------|-------------|-------------------|
| 1  | Afghan       | afghani     | TR                |
| 2  | Afghan       | afghani     | MX                |
| 3  | African      | african     | TR                |
| 4  | Andalusian   | andalusian  | US                |
| 5  | Arabian      | arabian     | DK                |
| 6  | Argentine    | argentine   | FI                |
| 7  | Armenian     | armenian    | US                |
| 8  | Asian Fusion | asianfusion | US                |
| 9  | Asturian     | asturian    | US                |
| 10 | Australian   | australian  | US                |
| 11 | Austrian     | austrian    | ES                |
| 12 | Austrian     | austrian    | DK                |
| 13 | Baguettes    | baguettes   | US                |
| 14 | Bangladeshi  | bangladeshi | DK                |

- Table 2 : iso\_3166\_1\_alpha\_2\_code

- Table 3 : us\_states

数据源结构

数据表

编辑备注

执行 SQL

表: us\_states

在下列列中过滤

|    | city          | city_ascii    | iso_3166_2 | state_name           | unit_fips | county_name          | lat     | lng       | population | density | source  | military | incorporated    | timezone              | ranking                                   | zip                                             | id         |
|----|---------------|---------------|------------|----------------------|-----------|----------------------|---------|-----------|------------|---------|---------|----------|-----------------|-----------------------|-------------------------------------------|-------------------------------------------------|------------|
|    | 城市            | 城市            | 城市         | 州名                   | 州名        | 县名                   | 纬度      | 经度        | 人口         | 人口      | 来源      | 军事       | 成立              | 时区                    | 排名                                        | 邮编                                              | 记录         |
| 1  | New York      | New York      | NY         | New York             | 36061     | New York             | 40.6843 | -73.9249  | 18713220   | 1075    | polygon | 0        | 1               | America/New_York      | 11228                                     | 11228 11228 11224 11222 11221 11220 11808 10989 | 1840030416 |
| 2  | Los Angeles   | Los Angeles   | CA         | California           | 06037     | Los Angeles          | 34.1139 | -118.4066 | 12758087   | 3274    | polygon | 0        | 1               | America/Los_Angeles   | 106291                                    | 90283 90292 91316 91311 90043 80008             | 1840020491 |
| 3  | Chicago       | Chicago       | IL         | Illinois             | 17031     | Cook                 | 41.8373 | -87.6602  | 8642103    | 4576    | polygon | 0        | 1               | America/Chicago       | 60048                                     | 60648 60641 60645 60643 60642 60648             | 1840020484 |
| 4  | Miami         | Miami         | FL         | Florida              | 12086     | Miami-Dade           | 25.7639 | -80.2102  | 5445455    | 5018    | polygon | 0        | 1               | America/New_York      | 33129                                     | 33129 33136 33077 33128 33149 33144 33145       | 1840015496 |
| 5  | Dallas        | Dallas        | TX         | Texas                | 48116     | Dallas               | 32.7813 | -96.7667  | 1256518    | 1450    | polygon | 0        | 1               | America/New_York      | 74359                                     | 75208 75008 75024 75021 75023 75025 75024       | 1840015490 |
| 6  | Philadelphia  | Philadelphia  | PA         | Pennsylvania         | 42103     | Philadelphia         | 40.2077 | -75.1339  | 5454300    | 556     | polygon | 0        | 1               | America/New_York      | 9154                                      | 19515 19106 19053 19115 19113 19033 19145       | 1840020073 |
| 7  | Houston       | Houston       | TX         | Texas                | 48201     | Harris               | 29.7663 | -95.3889  | 5464251    | 1399    | polygon | 0        | 1               | America/Chicago       | 77069                                     | 77068 77067 77060 77063 77062 77068 77064       | 1840019365 |
| 8  | Atlanta       | Atlanta       | GA         | Georgia              | 13121     | Fulton               | 33.7627 | -84.4224  | 5439398    | 1441    | polygon | 0        | 1               | America/New_York      | 130334                                    | 30331 30332 30339 30308 30308 30307 30303       | 1840020320 |
| 9  | Washington    | Washington    | DC         | District of Columbia | 11001     | Distric of Columbia  | 38.9047 | -77.0313  | 5791984    | 4457    | polygon | 0        | 1               | America/New_York      | 20101                                     | 20101 20102 20125 20228 20260 20307 20418       | 1840009660 |
| 10 | Boston        | Boston        | MA         | Massachusetts        | 25025     | District of Columbia | 42.3388 | -71.0846  | 4888346    | 5532    | polygon | 0        | 1               | America/New_York      | 02131                                     | 02112 02122 02124 02128 02126 02127 02128       | 1840004585 |
| 11 | Phoenix       | Phoenix       | AZ         | Arizona              | 05013     | Mariopola            | 33.7722 | -112.0219 | 1253       | polygon | 0       | 1        | America/Phoenix | 85004                 | 85004 85008 85007 85004 85005 85006 85007 | 1840020568                                      |            |
| 12 | Seattle       | Seattle       | WA         | Washington           | 52033     | King                 | 47.6121 | -122.3254 | 7387615    | 1265    | polygon | 0        | 1               | America/Seattle       | 98104                                     | 98008 98104 98107 98102 98101 98053 98103       | 1840020317 |
| 13 | San Francisco | San Francisco | CA         | California           | 06076     | San Francisco        | 37.6922 | -122.4243 | 3592924    | 2163    | polygon | 0        | 1               | America/San_Francisco | 94130                                     | 94131 94132 94133 94134 94109 94104 94103       | 1840021343 |
| 14 | Detroit       | Detroit       | MI         | Michigan             | 26163     | Wayne                | 42.3384 | -83.1024  | 3506126    | 1864    | polygon | 0        | 1               | America/Detroit       | 148209                                    | 48208 48201 48207 48205 48204 55405 48206       | 1840030937 |
| 15 | San Diego     | San Diego     | CA         | California           | 05013     | San Diego            | 32.8212 | -117.1025 | 3220118    | 1688    | polygon | 0        | 1               | America/Los_Angeles   | 92109                                     | 92108 92103 92113 92114 92114 92110 92115 92142 | 1840021990 |
| 16 | Minneapolis   | Minneapolis   | MN         | Minnesota            | 27025     | Hennepin             | 44.9635 | -93.2678  | 2977172    | 301     | polygon | 0        | 1               | America/Chicago       | 55403                                     | 55409 55408 55407 55406 55405 55404 55402       | 1840020730 |
| 17 | Tempe         | Tempe         | FL         | Florida              | 12057     | Hillsborough         | 27.8942 | -82.441   | 2500603    | 1363    | polygon | 0        | 1               | America/New_York      | 33637                                     | 33623 33621 33622 33619 33616 33613 33610       | 1840015892 |

- Table 4 : restaurant\_information

| 数据库结构 浏览数据 编辑备注 执行 SQL    |                 |                                  |                             |               |     |                        |                                                                                         |        |             |                   |       |                          |           |              |
|---------------------------|-----------------|----------------------------------|-----------------------------|---------------|-----|------------------------|-----------------------------------------------------------------------------------------|--------|-------------|-------------------|-------|--------------------------|-----------|--------------|
| 表: restaurant_information |                 |                                  |                             |               |     |                        |                                                                                         |        |             |                   |       |                          |           |              |
|                           | id              | alias                            | name                        | image_url     | url | review                 | categories                                                                              | rating | rating_text | transactions      | price | location                 | phone     | display_name |
| 过滤                        | 过滤              | 过滤                               | 过滤                          | 过滤            | 过滤  | 过滤                     | 过滤                                                                                      | 过滤     | 过滤          | 过滤                | 过滤    | 过滤                       | 过滤        | 过滤           |
| 1                         | xHNPKMLDF8K2... | the-halal-guys-new-york-2        | The Halal Guys              | https://i3... | 0   | https://www.yelp.co... | 9576 [{"alias": "hoodstanger", "title": "Food", "rating": 4.0, "text": "79.97..."}]     | 4.0    | 40.7616...  | ["pickup", "..."] | \$    | ["address": "W 53..."]   | +1347...  | (347) ...    |
| 2                         | Sh8eUQVh0XQp... | the-crawfish-sushi-burger-bar... | The Crawfish Sushi Burga... | https://i3... | 0   | https://www.yelp.co... | 2455 [{"alias": "sushi", "title": "Sushi Bar*", "rating": 4.0, "text": "81.46..."}]     | 4.0    | 28.473...   | ["delivery"]      | \$    | ["address": "..."]       | +14072... | (407) ...    |
| 3                         | v8Hw...         | the-toothsome-chocolate...       | The Toothsome Chocolat...   | https://i3... | 0   | https://www.yelp.co... | 2368 [{"alias": "desserts", "title": "...", "rating": 4.0, "text": "81.46..."}]         | 4.0    | 28.473...   | ["delivery"]      | \$    | ["address": "..."]       | +14072... | (407) ...    |
| 4                         | FEVQ6QPOwPN...  | shake-shack-new-york-2           | Shake Shack                 | https://i3... | 0   | https://www.yelp.co... | 5608 [{"alias": "burgers", "title": "Burgers", "rating": 4.0, "text": "73.98..."}]      | 4.0    | 40.7617...  | ["delivery"]      | \$    | ["address": "E 23r..."]  | +12128... | (212) ...    |
| 5                         | SB0G0Z0BwC3L... | the-diner-orlando-2              | @ The Diner                 | https://i3... | 0   | https://www.yelp.co... | 1211 [{"alias": "breakfast_brunch", "title": "...", "rating": 4.5, "text": "81.44..."}] | 4.5    | 28.425...   | ["delivery"]      | \$    | ["address": "9938..."]   | +14072... | (407) ...    |
| 6                         | 4a5AP...        | mediterranean-deli-orlando       | Mediterranean Deli          | https://i3... | 0   | https://www.yelp.co... | 409 [{"alias": "greek", "title": "Greek", "rating": 4.5, "text": "81.39..."}]           | 4.5    | 28.598...   | ["delivery"]      | \$    | ["address": "5981..."]   | +14076... | (407) ...    |
| 7                         | w4QZ0K0uN6E...  | alca's-pastal-deli-round-lake... | Alca's Pastal Deli          | https://i3... | 0   | https://www.yelp.co... | 19 [{"alias": "italian", "title": "Italian", "rating": 4.5, "text": "88.0..."}]         | 4.5    | 42.381...   | ["delivery"]      | \$    | ["address": "9972 E..."] | +12123... | (212) ...    |
| 8                         | 6a6a87H0u85U... | kalogen-round-lake-beach         | Kalogen                     | https://i3... | 0   | https://www.yelp.co... | 18 [{"alias": "pizza", "title": "Pizza", "rating": 4.0, "text": "88.0..."}]             | 4.0    | 42.381...   | ["delivery"]      | \$    | ["address": "9802 E..."] | +18472... | (847) ...    |
| 9                         | ehUa5g9TQm...   | uptate-craft-beer-and-cyster...  | Uptate Craft Beer & ...     | https://i3... | 0   | https://www.yelp.co... | 1844 [{"alias": "seafood", "title": "Seafood", "rating": 4.5, "text": "81.38..."}]      | 4.5    | 40.725...   | ["delivery"]      | \$    | ["address": "365 W..."]  | +18487... | (848) ...    |
| 10                        | H2R...          | toko-china-orlando               | Toko China                  | https://i3... | 0   | https://www.yelp.co... | 1634 [{"alias": "asianfusion", "title": "Asian", "rating": 4.0, "text": "81.36..."}]    | 4.0    | 28.558...   | ["delivery"]      | \$    | ["address": "948..."]    | +14077... | (407) ...    |
| 11                        | g2vaWuCH66TK... | tacos-el-norte-round-lake...     | Tacos El Norte              | https://i3... | 0   | https://www.yelp.co... | 103 [{"alias": "mexican", "title": "Mexican", "rating": 4.0, "text": "88.0..."}]        | 4.0    | 42.382...   | ["delivery"]      | \$    | ["address": "9006 E..."] | +18476... | (847) ...    |
| 12                        | T0JL86YRYXM...  | oasis-micro-pub-rockford         | Oasis Micro Pub             | https://i3... | 0   | https://www.yelp.co... | 48 [{"alias": "pubs", "title": "Pubs", "rating": 4.5, "text": "89.0..."}]               | 4.5    | 42.266...   | ["delivery"]      | \$    | ["address": "9901 E..."] | +18169... | (816) ...    |
| 13                        | SMYV07NE...     | benito-cafe-hainesville          | Benito Cafe                 | https://i3... | 0   | https://www.yelp.co... | 133 [{"alias": "japanese", "title": "...", "rating": 4.0, "text": "84.45..."}]          | 4.0    | 42.3445...  | ["delivery"]      | \$    | ["address": "9 W W..."]  | +18472... | (847) ...    |
| 14                        | h0Y2m2wPFZ9...  | 3-amigos-lake-villa              | 3 Amigos                    | https://i3... | 0   | https://www.yelp.co... | 96 [{"alias": "mexican", "title": "Mexican", "rating": 3.5, "text": "42.414..."}]       | 3.5    | 42.414...   | ["delivery"]      | \$    | ["address": "8996 E..."] | +18473... | (847) ...    |
| 15                        | 8H463JIT2...    | grand-central-terminal-new...    | Grand Central Terminal      | https://i3... | 0   | https://www.yelp.co... | 1384 [{"alias": "handmarks", "title": "...", "rating": 4.5, "text": "79.97..."}]        | 4.5    | 40.752...   | ["delivery"]      | N     | ["address": "59 E..."]   | +12123... | (212) ...    |
| 16                        | SH9FAMZ4H9E...  | cafe-to-go-orlando-5             | Cafe To Go Togo             | https://i3... | 0   | https://www.yelp.co... | 3393 [{"alias": "hawaiian", "title": "Hawa...", "rating": 4.5, "text": "88.44..."}]     | 4.5    | 28.446...   | ["delivery"]      | \$    | ["address": "9825 E..."] | +14072... | (407) ...    |
| 17                        | J06N7vW2LqU...  | mr-y's-restaurant-round-lake...  | Mrs. Y's Restaurant         | https://i3... | 0   | https://www.yelp.co... | 68 [{"alias": "diner", "title": "Diners", "rating": 3.5, "text": "42.380..."}]          | 3.5    | 42.380...   | ["delivery"]      | \$    | ["address": "865 E..."]  | +18472... | (847) ...    |
| 18                        | N...            | olando-pizza-round-lake-rou...   | Olando's Pizza Round Lake   | https://i3... | 0   | https://www.yelp.co... | 184 [{"alias": "pizza", "title": "Pizza", "rating": 4.0, "text": "42.378..."}]          | 4.0    | 42.378...   | ["delivery"]      | \$    | ["address": "843..."]    | +18479... | (847) ...    |

- Table 5 : restaurant\_category\_fetch

数据库结构

浏览数据

编辑备注

执行 SQL

表: restaurant\_category\_fetch

在所有列中过滤

|    | id                    | category   | city             |
|----|-----------------------|------------|------------------|
|    | 过滤                    |            | 过滤               |
| 1  | a5f6dQ0k987wD0xg30QA  | Andalusian | Round Lake Beach |
| 2  | vmk98Bj3Y1HhQ8aTEnQ   | Andalusian | Round Lake Beach |
| 3  | 4Kx70wLH3npM5hWAnRwQ  | Andalusian | Round Lake Beach |
| 4  | g2vaWuCH66TKswuQfQ    | Andalusian | Round Lake Beach |
| 5  | 8G6gVUgBr64Vg3K1vGQ   | Andalusian | Round Lake Beach |
| 6  | 3PaQ92pxDpC1-MNHQFQ   | Andalusian | Round Lake Beach |
| 7  | vA8gb-TTq3Y186ZP8w    | Andalusian | Round Lake Beach |
| 8  | w0L37wZ2W8Myj0H1v     | Andalusian | Round Lake Beach |
| 9  | HypmKZZv89g3QwD8B4    | Andalusian | Round Lake Beach |
| 10 | s2w_3wXm5C0RvY2_2Mq3Q | Andalusian | Round Lake Beach |
| 11 | jVUsu1W9g97B2Ymax8g   | Andalusian | Round Lake Beach |
| 12 | JT8RNoYK9Y1G-XW8vMLA  | Andalusian | Round Lake Beach |
| 13 | 6y8Hk2gBuL5N5o2PwrcQ  | Andalusian | Round Lake Beach |
| 14 | wLdM2o87N8rFH-M6SQw   | Andalusian | Round Lake Beach |
| 15 | J06N7vW2LqU3Cw43pr1A  | Andalusian | Round Lake Beach |
| 16 | JvR8hFrH0paze-IR2g    | Andalusian | Round Lake Beach |

# Interaction and Presentation Options

## Description

The program allows users to select specific states and cities, and view the average ratings of different restaurant types from the bar graph. You can first select a state and submit, and then select a city and submit. The program will generate bar graphs based on the state and city you provide. It contains two **bar graphs**, the first is the average rating, and the second is the statistics of each type of restaurant

Then, you can choose a specific type of food, the program will present useful information about the covid-19 (including updated services and health & safety measures) and specific restaurant information presented in a **table** to help you decide where to go. I will only show some useful information in the table (including name, image, url, and rating). The program will also provide a **plot of map** to facilitate you to choose a location closer to you. You can click the Map button to switch the table to the map.

Finally, you can choose one restaurant specifically, the program will offer you more details in **text**. This part will contain many details for each store. The URL is also provided if you want to make an order or see more details on the yelp webpage.

Besides the flask app, I also provide the **logging information** in the command line. You can know what exactly the flask app is doing from the command line. This includes several types: database, cache, yelp\_covid, yelp\_fusion, CSV, and flask. Each of them represents a specific python module and you can easily figure out the relationship.

## Technologies

- **Flask:** The whole program is running within a Flask App.
- **Plotly:** I use plotly to draw bar plot. There are some statistic data for covid 19 I can use.
- **Folium:** This is a map package, I use it to draw a map to show the locations of the restaurants.
- **Command line:** Only for logging, you can see what the flask app is doing from the command line.

## Brief instructions

## Demo Link