Final Project

Project code

- **Github repo link:** https://github.com/Huadous/final-project
- **README:** This program uses the API provided by Yelp Fusion. The **API verification key** is very easy to apply, please apply through this link: https://www.yelp.com/fusion. Sign up for an account and manage an app to gain the API Key. Then, you need to create a file named `secrets.py` and copy this variable `API_KEY = 'your_api_key'` into the file to support the program.

  The entire program is built within the **Flask app**, and the 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 types. After selecting the state and city, statistical information about different categories will be displayed. Then you can choose your favorite category based on them. At this time, the information of the restaurants in that category will be displayed in a table, or you can click the **Map button** to display their locations. In the table, you can click the "**Click here**" **buttons** to enter the detailed information page of the restaurant.
- **Required Python packages:** requests, plotyly, folium, pandas, flask and bs4.

Data sources

Data source 1:(4 points)Multiple related data files(one>1000 records).(source1,2,3) ***Data source 2:***(4 points)Web API requires API key✣.(source 4) ***Data source 3:***(8 points)Crawling and scraping multiple pages✣.(source 5)

Categories information

- **Origin:** Documentation Download **Format:** JSON$(> 1000\text{ records})[\approx 1500]\{192\text{ used}\}$
- **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$(< 1000\text{ records})[\approx 250]\{250\text{ used}\}$
- **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 Download **Format:** CSV$(>1000\text{ records})[\approx 28000]\{28399\text{ used}\}$
- **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$(>1000\text{ records})[\approx \infty]\{[50,1000]\text{ for each type will be used}\}$

  `GET https://api.yelp.com/v3/businesses/search` (Each request can only get up to 50 results and get up to 1000 results using multiple queries and combinations of the "limit" and "offset")

- **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. 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$[\approx \infty]\{[50,1000]\text{ for each type will be used}\}$
- **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 & satety measures the restaurant has implemented.

Database

Database schema

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

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

ISO 3166-1 alpha-2 codes are two-letter country codes. This form is suitable for filtering which categories are available in which countries (although my program is only used in the United States).

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

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.Because location is needed when searching, all the location is associated with the search record table.

```
CREATE TABLE IF NOT EXISTS us_states(
"city"  TEXT, # name of the city
"city_ascii"  TEXT, # ascii of the name of the city
"state_id"  TEXT, # alpha2 of the state (`'NY'` for New York)
"state_name"  TEXT, # name of the state (New York)
"county_fips" TEXT,
"county_name" TEXT,
"lat" REAL,
"lng" REAL,
"population"  INTEGER,
"density" INTEGER,
"source"  TEXT,
"military"  INTEGER,
"incorporated"  INTEGER,
"timezone"  TEXT,
"ranking" INTEGER,
"zips"  TEXT,
"id"  TEXT, # unique id for each city
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.

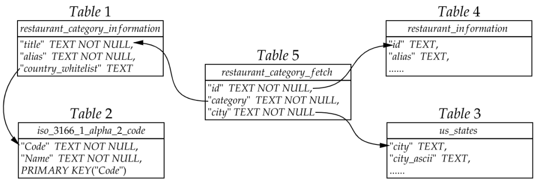
```
CREATE TABLE IF NOT EXISTS restaurant_information(
"id"  TEXT, # id of each restaurant
"alias" TEXT, # alias of the restaurant
"name"  TEXT, # name of the restaurant
"image_url" TEXT, # url of the image of the restaurant
"is_closed" INTEGER, # whether is closed
"url" TEXT, # url in yelp
"review_count"  INTEGER,
"categories"  TEXT,
"rating"  REAL,
"coordinates_latitude"  REAL,
"coordinates_longitude" REAL,
"transactions"  TEXT,
"price" TEXT,
"location"  TEXT,
"phone" TEXT,
"display_phone" TEXT,
PRIMARY KEY("id"));
```

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.

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

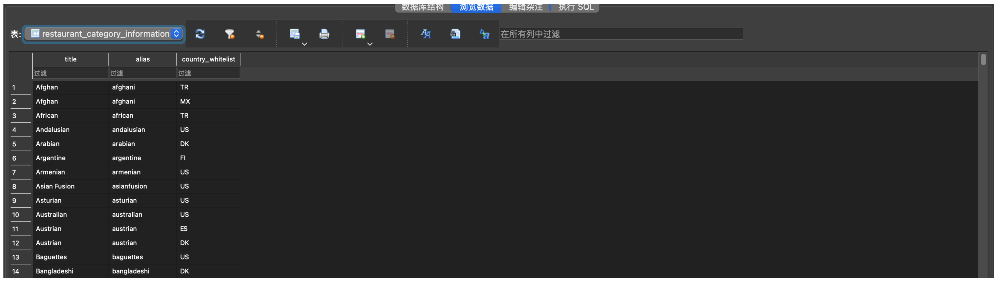
Foreign key-primary key relations

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



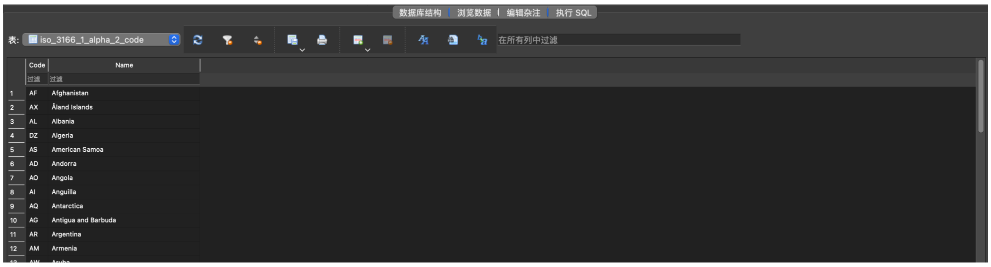
Screenshots of the data

- Table 1 : restaurant_category_information



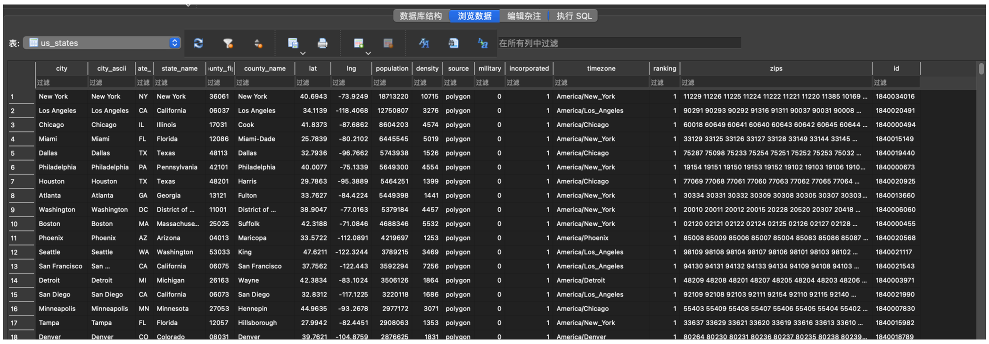
	title	alias	country_whitelist
	过滤	过滤	过滤
1	Afghan	afghani	TR
2	Afghan	afghani	MX
3	African	african	TR
4	Andalusian	andalusian	US
5	Arabian	arabian	DK
6	Argentine	argentine	FI
7	Armenian	armenian	US
8	Asian Fusion	asianfusion	US
9	Asturian	asturian	US
10	Australian	australian	US
11	Austrian	austrian	ES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

- Table 2 : iso_3166_1_alpha_2_code



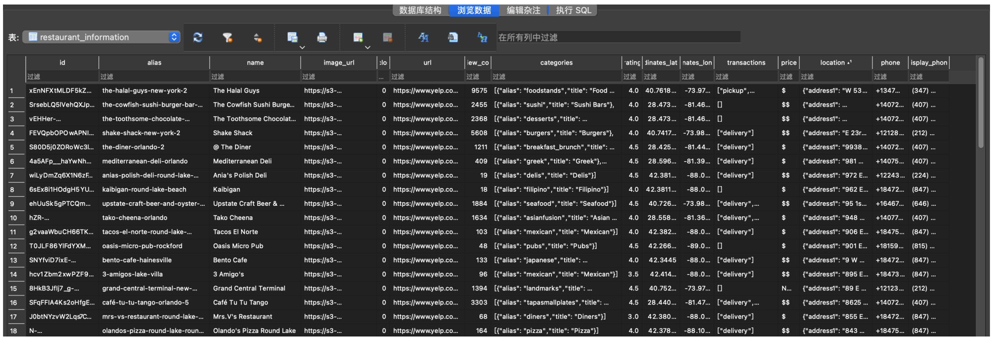
	Code	Name
	过滤	过滤
1	AF	Afghanistan
2	AX	Åland Islands
3	AL	Albania
4	DZ	Algeria
5	AS	American Samoa
6	AD	Andorra
7	AO	Angola
8	AI	Anguilla
9	AQ	Antarctica
10	AG	Antigua and Barbuda
11	AR	Argentina
12	AM	Armenia
13	AW	Aruba

- Table 3 : us_states



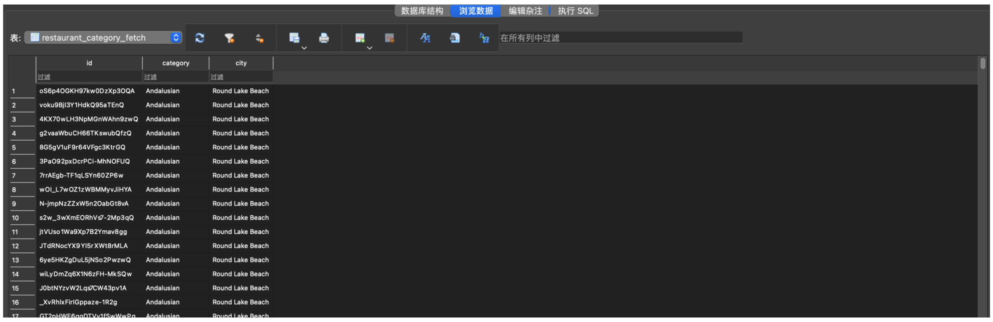
	city	city_ascii	ate_	state_name	unty_fi	county_name	lat	lng	population	density	source	military	incorporated	timezone	ranking	zips	id
	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤
1	New York	New York	NY	New York	36061	New York	40.6943	-73.9249	18713220	10715	polygon	0	1	America/New_York	1	11229 11226 11225 11224 11222 11221 11220 11385 10169 ...	1840034016
2	Los Angeles	Los Angeles	CA	California	06037	Los Angeles	34.1139	-118.4068	12750807	3276	polygon	0	1	America/Los_Angeles	1	90291 90293 90292 91316 91311 90037 90031 90008 ...	1840020491
3	Chicago	Chicago	IL	Illinois	17031	Cook	41.8373	-87.6862	8604203	4574	polygon	0	1	America/Chicago	1	60018 60649 60641 60640 60643 60642 60645 60644 ...	1840000494
4	Miami	Miami	FL	Florida	12086	Miami-Dade	25.7839	-80.2102	6445545	5019	polygon	0	1	America/New_York	1	33129 33125 33126 33127 33128 33149 33144 33145 ...	1840015149
5	Dallas	Dallas	TX	Texas	48113	Dallas	32.7936	-96.7662	5743938	1526	polygon	0	1	America/Chicago	1	75287 75098 75233 75254 75251 75252 75253 75032 ...	1840019440
6	Philadelphia	Philadelphia	PA	Pennsylvania	42101	Philadelphia	40.0077	-75.1339	5649300	4554	polygon	0	1	America/New_York	1	19154 19151 19150 19153 19152 19102 19103 19106 1910...	1840000673
7	Houston	Houston	TX	Texas	48201	Harris	29.7863	-95.3889	5464251	1399	polygon	0	1	America/Chicago	1	77069 77068 77061 77060 77063 77062 77065 77064 ...	1840020925
8	Atlanta	Atlanta	GA	Georgia	13121	Fulton	33.7627	-84.4224	5449398	1441	polygon	0	1	America/New_York	1	30334 30331 30332 30309 30308 30305 30307 30303...	1840013660
9	Washington	Washington	DC	District of ...	11001	District of ...	38.9047	-77.0163	5379184	4457	polygon	0	1	America/New_York	1	20010 20011 20012 20015 20228 20520 20307 20418 ...	1840006060
10	Boston	Boston	MA	Massachuse...	25025	Suffolk	42.3188	-71.0846	4688346	5532	polygon	0	1	America/New_York	1	02120 02121 02122 02124 02125 02126 02127 02128 ...	1840000455
11	Phoenix	Phoenix	AZ	Arizona	04013	Maricopa	33.5722	-112.0891	4219697	1253	polygon	0	1	America/Phoenix	1	85008 85009 85006 85007 85004 85083 85086 85087 ...	1840020568
12	Seattle	Seattle	WA	Washington	53033	King	47.6211	-122.3244	3789215	3469	polygon	0	1	America/Los_Angeles	1	98109 98108 98104 98107 98106 98101 98103 98102 ...	1840021117
13	San Francisco	San ...	CA	California	06075	San Francisco	37.7562	-122.443	3592294	7256	polygon	0	1	America/Los_Angeles	1	94130 94131 94132 94133 94134 94109 94108 94103 ...	1840021543
14	Detroit	Detroit	MI	Michigan	26163	Wayne	42.3834	-83.1024	3506126	1864	polygon	0	1	America/Detroit	1	48209 48208 48201 48207 48205 48204 48203 48206 ...	1840003971
15	San Diego	San Diego	CA	California	06073	San Diego	32.8312	-117.1225	3220118	1686	polygon	0	1	America/Los_Angeles	1	92109 92108 92103 92111 92154 92110 92115 92140 ...	1840021990
16	Minneapolis	Minneapolis	MN	Minnesota	27053	Hennepin	44.9635	-93.2678	2977172	3071	polygon	0	1	America/Chicago	1	55403 55409 55408 55407 55406 55405 55404 55402 ...	1840007830
17	Tampa	Tampa	FL	Florida	12057	Hillsborough	27.9942	-82.4451	2908063	1353	polygon	0	1	America/New_York	1	33637 33629 33621 33620 33619 33616 33613 33610 ...	1840015982
18	Denver	Denver	CO	Colorado	08031	Denver	39.7621	-104.8759	2876625	1831	polygon	0	1	America/Denver	1	80264 80230 80231 80236 80237 80235 80238 80239...	1840018789

- Table 4 : restaurant_information



	id	alias	name	image_url	:lo	url	iew_co	categories	ating	dinates_lat	nates_lon	transactions	price	location	phone	isplay_phon
	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	xEnNFXtMLDF5kZ...	the-halal-guys-new-york-2	The Halal Guys	https://s3-...	0	https://www.yelp.co...	9575	[{"alias": "foodstands","title": "Food ...	4.0	40.7618...	-73.97...	["pickup",...	$	{"address1": "W 53...	+1347...	(347) ...
2	SrsebLQ5lVehQXJp...	the-cowfish-sushi-burger-bar-...	The Cowfish Sushi Burge...	https://s3-...	0	https://www.yelp.co...	2455	[{"alias": "sushi","title": "Sushi Bars"},	4.0	28.473...	-81.46...	[]	$$	{"address1": ...	+14072...	(407) ...
3	vEHHer-...	the-toothsome-chocolate-...	The Toothsome Chocolat...	https://s3-...	0	https://www.yelp.co...	2368	[{"alias": "desserts","title": ...	4.0	28.473...	-81.46...	[]	$$	{"address1": ...	+14072...	(407) ...
4	FEVQpbOPOwAPNI...	shake-shack-new-york-2	Shake Shack	https://s3-...	0	https://www.yelp.co...	5608	[{"alias": "burgers","title": "Burgers"},	4.0	40.7417...	-73.98...	["delivery"]	$$	{"address1": "E 23r...	+12128...	(212) ...
5	S80D5j0ZORoWc3l...	the-diner-orlando-2	@ The Diner	https://s3-...	0	https://www.yelp.co...	1211	[{"alias": "breakfast_brunch","title": ...	4.5	28.425...	-81.44...	["delivery"]	$	{"address1": "9938...	+14072...	(407) ...
6	4a5AFp__haYwNh...	mediterranean-deli-orlando	Mediterranean Deli	https://s3-...	0	https://www.yelp.co...	409	[{"alias": "greek","title": "Greek"},...	4.5	28.596...	-81.39...	["delivery"]	$	{"address1": "981 ...	+14075...	(407) ...
7	wiLyDmZq6X1N6zF...	anias-polish-deli-round-lake-...	Ania's Polish Deli	https://s3-...	0	https://www.yelp.co...	19	[{"alias": "delis","title": "Delis"}]	4.5	42.381...	-88.0...	["delivery"]	$$	{"address1": "972 E...	+12243...	(224) ...
8	6sEx8i1HOdgH5YU...	kaibigan-round-lake-beach	Kaibigan	https://s3-...	0	https://www.yelp.co...	18	[{"alias": "filipino","title": "Filipino"}]	4.0	42.3811...	-88.0...	[]	$	{"address1": "962 E...	+18472...	(847) ...
9	ehUuSk5gPTCQm...	upstate-craft-beer-and-oyster-...	Upstate Craft Beer & ...	https://s3-...	0	https://www.yelp.co...	1884	[{"alias": "seafood","title": "Seafood"}]	4.5	40.726...	-73.98...	["delivery",...	$$	{"address1": "95 1s...	+16467...	(646) ...
10	hZR-...	tako-cheena-orlando	Tako Cheena	https://s3-...	0	https://www.yelp.co...	1634	[{"alias": "asianfusion","title": "Asian ...	4.0	28.558...	-81.36...	["delivery",...	$	{"address1": "948 ...	+14077...	(407) ...
11	g2vaaWbuCH66TK...	tacos-el-norte-round-lake-...	Tacos El Norte	https://s3-...	0	https://www.yelp.co...	103	[{"alias": "mexican","title": "Mexican"}]	4.0	42.382...	-88.0...	["delivery"]	$	{"address1": "906 E...	+18475...	(847) ...
12	T0JLF86YIFdYXM...	oasis-micro-pub-rockford	Oasis Micro Pub	https://s3-...	0	https://www.yelp.co...	48	[{"alias": "pubs","title": "Pubs"}]	4.5	42.266...	-89.0...	[]	$	{"address1": "901 E...	+18159...	(815) ...
13	SNYfviD7ixE-...	bento-cafe-hainesville	Bento Cafe	https://s3-...	0	https://www.yelp.co...	133	[{"alias": "japanese","title": ...	4.0	42.3445	-88.0...	["delivery"]	$$	{"address1": "9 W ...	+18472...	(847) ...
14	hcv1Zbm2xwPZF9...	3-amigos-lake-villa	3 Amigo's	https://s3-...	0	https://www.yelp.co...	96	[{"alias": "mexican","title": "Mexican"}]	3.5	42.414...	-88.0...	["delivery"]	$$	{"address1": "895 E...	+18473...	(847) ...
15	8HkB3Jflj7_g-...	grand-central-terminal-new-...	Grand Central Terminal	https://s3-...	0	https://www.yelp.co...	1394	[{"alias": "landmarks","title": ...	4.5	40.752...	-73.97...	[]	N...	{"address1": "89 E ...	+12123...	(212) ...
16	SFqFFIA4Ks2oHfgE...	café-tu-tu-tango-orlando-5	Café Tu Tu Tango	https://s3-...	0	https://www.yelp.co...	3303	[{"alias": "tapasmallplates","title": ...	4.5	28.440...	-81.47...	["delivery",...	$$	{"address1": "8625 ...	+14072...	(407) ...
17	J0btNYzvW2Lqs7C...	mrs-vs-restaurant-round-lake-...	Mrs.V's Restaurant	https://s3-...	0	https://www.yelp.co...	68	[{"alias": "diners","title": "Diners"}]	3.0	42.380...	-88.0...	["delivery"]	$	{"address1": "855 E...	+18472...	(847) ...
18	N-...	olandos-pizza-round-lake-roun...	Olando's Pizza Round Lake	https://s3-...	0	https://www.yelp.co...	164	[{"alias": "pizza","title": "Pizza"}]	4.0	42.378...	-88.10...	["delivery"]	$$	{"address1": "843 ...	+18475...	(847) ...

- Table 5 : restaurant_category_fetch



	id	category	city
	过滤	过滤	过滤
1	oS6p4OGKH97kw0DzXp3OQA	Andalusian	Round Lake Beach
2	voku9BjI3Y1HdkQ95aTEnQ	Andalusian	Round Lake Beach
3	4KX70wLH3NpMGnWAhn9zwQ	Andalusian	Round Lake Beach
4	g2vaaWbuCH66TKswubQfzQ	Andalusian	Round Lake Beach
5	8G5gV1uF9r64VFgc3KtrGQ	Andalusian	Round Lake Beach
6	3PaO92pxDcrPCi-MhNOFUQ	Andalusian	Round Lake Beach
7	7rrAEgb-TF1qLSYn60ZP6w	Andalusian	Round Lake Beach
8	wOI_L7wOZ1zWBMMyvJiHYA	Andalusian	Round Lake Beach
9	N-jmpNzZZxW5n2OabGt8vA	Andalusian	Round Lake Beach
10	s2w_3wXmEORhVs7-2Mp3qQ	Andalusian	Round Lake Beach
11	jtVUso1Wa9Xp7B2Ymav8gg	Andalusian	Round Lake Beach
12	JTdRNocYX9YI5rXWt8rMLA	Andalusian	Round Lake Beach
13	6ye5HKZgDuL5jNSo2PwzwQ	Andalusian	Round Lake Beach
14	wiLyDmZq6X1N6zFH-MkSQw	Andalusian	Round Lake Beach
15	J0btNYzvW2Lqs7CW43pv1A	Andalusian	Round Lake Beach
16	_XvRhlxFirlGppaze-1R2g	Andalusian	Round Lake Beach
17	GT2pHWF6qqDTVy1fSwWwPg	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 including 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

The entire program is built within the **Flask app**, and the 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 types. After selecting the state and city, statistical information about different categories will be displayed. Then you can choose your favorite category based on them. At this time, the information of the restaurants in that category will be displayed in a table, or you can click the **Map button** to display their locations. In the table, you can click the "**Click here**" **buttons** to enter the detailed information page of the restaurant.

Demo Link