Final Project Checkpoint

GitHub Link:

Link to GitHub repo of final project: https://github.com/sherryxj1104/SI507-final-project.git
Demo Video Link: https://youtu.be/TaBGjH80jCA
For further information, please check the README.md.

Data Source:

Yelp Fusion API:

- API URL: https://api.yelp.com/v3/businesses/search
- Documentation URL: https://www.yelp.com/developers/documentation/v3/business_search
- Format: JSON
- Summar of data:
    - # record available: It will have the first 20 records available for users to go over and find the best.
    - # record retrieved: This API retrieved the first 20 records when I search Ann Arbor.
    - Description of records: In the API returned value, it will have many attributes for users to review, I will list some important and useful attributes below:
        - Name: this is the name of the business in the area where the user search.
        - Is_closed: this provides whether the business is currently closed or not.
        - Categories: this is the category to which this business belongs.
        - Rating: this is the rating of the business that users search for.
        - Price: it will provide the approximate price of this business shown as the number of the dollar sign.
        - Location: this can help users to find the business. It has the full address, city, zipcode, counter, state, and country.
        - Phone: this provides the phone numbers for users to contact the business.
- To access the data in this API, I need to use an API_key. I think it needs to use caching to store some data in order to have an advanced search.
- Evidence of caching:

Open in Notebook Editor

```
{'businesses': [{'id': 'uTMqhmpgfpDMLN3W3YvMeQ',
  'alias': 'frita-batidos-ann-arbor',
  'name': 'Frita Batidos',
  'image_url': 'https://s3-media3.fl.yelpcdn.com/bphoto/Dxn825cYU1eFEoUXClZ_Dg/o.jpg',
  'is_closed': False,
  'url': 'https://www.yelp.com/biz/frita-batidos-ann-arbor?adjust_creative=Fh4wM2R-Bluj5wKfL4THXw&utm_campaign=yelp_api_v3&utm_medium=api_v3_bu
  'review_count': 1937,
  'categories': [{'alias': 'cuban', 'title': 'Cuban'},
   {'alias': 'burgers', 'title': 'Burgers'}],
  'rating': 4.5,
  'coordinates': {'latitude': 42.2803651, 'longitude': -83.7491532},
  'transactions': ['pickup', 'delivery'],
  'price': '$$',
  'location': {'address1': '117 W Washington St',
   'address2': '',
   'address3': '',
   'city': 'Ann Arbor',
   'zip_code': '48104',
   'country': 'US',
   'state': 'MI',
   'display_address': ['117 W Washington St', 'Ann Arbor, MI 48104']},
  'phone': '+17347612882',
  'display_phone': '(734) 761-2882',
  'distance': 783.8571139726386},
 {'id': 'fQ8c9S6jitKS5RT6S-ziGA',
  'alias': 'zingermans-delicatessen-ann-arbor-2',
  'name': "Zingerman's Delicatessen",
  'image_url': 'https://s3-media1.fl.yelpcdn.com/bphoto/IzbO8i-Tcq6khNRrZSGU4A/o.jpg',
  'is_closed': False,
  'url': 'https://www.yelp.com/biz/zingermans-delicatessen-ann-arbor-2?adjust_creative=Fh4wM2R-Bluj5wKfL4THXw&utm_campaign=yelp_api_v3&utm_medi
  'review_count': 2275,
  'categories': [{'alias': 'delis', 'title': 'Delis'},
   {'alias': 'salad', 'title': 'Salad'},
   {'alias': 'sandwiches', 'title': 'Sandwiches'}],
  'rating': 4.0,
  'coordinates': {'latitude': 42.2844740041972,
   'longitude': -83.7452775306885},
```


Booking API:

The website, rapidapi.com, provides several free APIs. So, I will directly use the API it provided instead of applying API on different websites.
First of all, this search engine will use the user's input to find the "dest_id" by using the API https://booking-com.p.rapidapi.com/v1/hotels/locations which is also provided in the rapidapi.com. After getting the "dest_id", it will begin the hotel search. In the hotel search, it will ask users several questions to narrow down the search result. Here are the questions:

```
"checkout_date":"xxx","units":"metric","dest_id":"xxx","dest_type":"city","locale":"en-gb","adults_number":"xxx","order_by":"popularity","filter_by_currency":"USD","checkin_date":"xxx","room_number":"xxx"
```

- API URL: https://booking-com.p.rapidapi.com/v1/hotels/search
- Documentation URL: https://docs.rapidapi.com/
- Format: JSON
- Summar of data:
    - # record available: It will return all possible result if the hotel meets the requirements.
    - # record retrieved: Take Ann Arbor as an example, it will have 28 results.
    - Description of records: In the API returned value, it will have many attributes for users to review and choose, I will list some important and useful attributes below:
        - Hotel_name: This is the name of the hotel.
        - Review_score: This is the score of previous reviews that users can check.

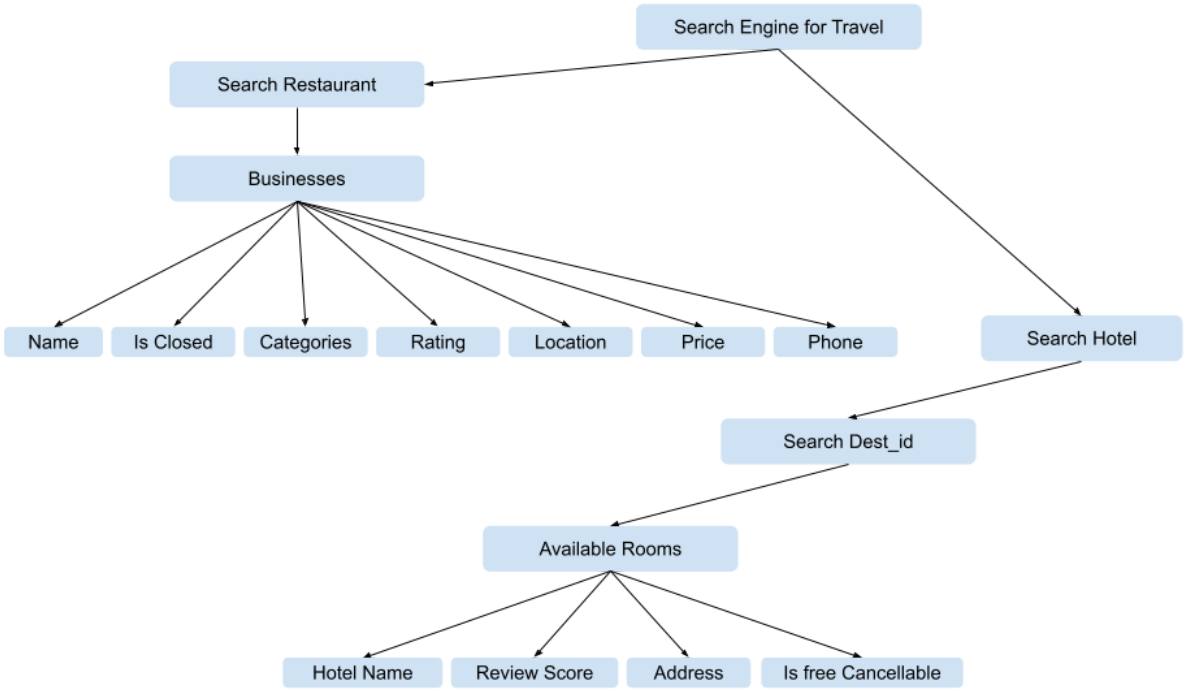
        - Address: This is the full address that the user can search for.
        - Is_free_cancellable: It shows as “0” or “1”. 0 means this room is not available for free canceling. 1 means this room is available for free canceling.
- To access the data in this API, I need to use an API_key. I think it needs to use caching to store some data in order to have an advanced search.
- Evidence of caching:


```
{"primary_count":28,"count":28,"room_distribution":[{"children":[],"adults":"2"}],"map_bounding_box":
{"sw_long":-83.841098,"sw_lat":42.2246145573659,"ne_lat":42.3078239141229,"ne_long":-83.6133746802807},"total_
count_with_filters":28,"unfiltered_count":46,"extended_count":0,"unfiltered_primary_count":46,"search_radius":
0.0,"sort":[{"id":"distance","name":"Distance from city centre"},{"id":"popularity","name":"Popularity"},
{"name":"Stars and other ratings (5 to 0)","id":"class_descending"},{"id":"class_ascending","name":"Stars and
other ratings (0 to 5)\n"},{"name":"Guest review score","id":"bayesian_review_score"},{"name":"Price (low to
high)","id":"price"}],"result":[{"wishlist_count":0,"address":"2455 Carpenter
Road","cc_required":1,"is_smart_deal":0,"preferred":1,"city_trans":"Ann Arbor","cc1":"us","hotel_name":"Ann
Arbor Regent Hotel and Suites","accommodation_type_name":"Hotel","timezone":"America/Detroit","checkout":
{"from":"","until":"12:00"},"class_is_estimated":0,"city_name_en":"Ann Arbor
(Michigan)","main_photo_id":27739245,"checkin":
{"from":"15:00","until":""},"districts":"","default_wishlist_name":"Ann
Arbor","in_best_district":0,"min_total_price":220.89,"urgency_message":"Only 3 left at this price on
Booking.com","mobile_discount_percentage":0,"native_ads_tracking":"","soldout":0,"is_mobile_deal":0,"preferred
_plus":0,"review_score":8.6,"bwallet":{"hotel_eligibility":0},"currency_code":"USD","distances":
[{"icon_name":"bui_geo_pin","icon_set":null,"text":"4.6 km from centre"},{"text":"Cleanliness
9.0","icon_set":null,"icon_name":"bui_clean"}],"extended":0,"native_ad_id":"","district_id":0,"selected_review
_topic":null,"is_wholesaler_candidate":1,"distance_to_cc":"4.60","url":"https://www.booking.com/hotel/us/ann-
arbor-regent-and-
suites.html","children_not_allowed":null,"main_photo_url":"https://cf.bstatic.com/xdata/images/hotel/square60/
27739245.jpg?k=9f41c5394402bcfc0e76aecce84cd46c484e35e5e0fc2db5e82f5731d4d02f69&o=","country_trans":"United
States","is_genius_deal":0,"hotel_has_vb_boost":0,"review_score_word":"Fabulous","hotel_name_trans":"Ann Arbor
Regent Hotel and Suites","currencycode":"USD","composite_price_breakdown":
{"included_taxes_and_charges_amount":{"value":21.89,"currency":"USD"},"items":[{"item_amount":
{"currency":"USD","value":21.89},"base":
{"percentage":11.0,"kind":"percentage"},"name":"Tax","kind":"charge","details":"11 %
Tax","inclusion_type":"included"}],"benefits":[],"excluded_amount":
```


Data Structure:

I will use a python function in the python file to create a tree structure from stored data. A JSON file named *hotel.json* will store the data retrieved from Booking API. A JSON file named *restaurant.json* will store the data retrieved from Yelp API. I will use some functions to combine

these two large JSON files together like the tree shown below:



This is the data structure of the restaurant.json file:


```
{'businesses': [{'id': 'uTMqhmpgfpDMLN3W3YvMeQ',
   'alias': 'frita-batidos-ann-arbor',
   'name': 'Frita Batidos',
   'image_url': 'https://s3-media3.fl.yelpcdn.com/bphoto/Dxn825cYU1eFEoUXClZ_Dg/o.jpg',
   'is_closed': False,
   'url': 'https://www.yelp.com/biz/frita-batidos-ann-arbor?adjust_creative=Fh4wM2R-
Bluj5wKfL4THXw&utm_campaign=yelp_api_v3&utm_medium=api_v3_business_search&utm_source=Fh4wM2R-Bluj5wKfL4THXw',
   'review_count': 1937,
   'categories': [{'alias': 'cuban', 'title': 'Cuban'},
    {'alias': 'burgers', 'title': 'Burgers'}],
   'rating': 4.5,
   'coordinates': {'latitude': 42.2803651, 'longitude': -83.7491532},
   'transactions': ['delivery', 'pickup'],
   'price': '$$',
   'location': {'address1': '117 W Washington St',
    'address2': '',
    'address3': '',
    'city': 'Ann Arbor',
    'zip_code': '48104',
    'country': 'US',
    'state': 'MI',
    'display_address': ['117 W Washington St', 'Ann Arbor, MI 48104']},
   'phone': '+17347612882',
   'display_phone': '(734) 761-2882',
   'distance': 783.8571139726386},
  {'id': 'fQ8c9S6jitKS5RT6S-ziGA',
...
   'display_phone': '(734) 369-2602',
   'distance': 336.4804016653341}],
 'total': 1000,
 'region': {'center': {'longitude': -83.74122619628906,
   'latitude': 42.2767546461982}}}
```


This is the data structure of the hotel.json file:


```
▼ { 11 items
    "primary_count" : 7
    "count" : 7
    ▸ "room_distribution" : [...] 1 item
    ▸ "map_bounding_box" : {...} 4 items
    "total_count_with_filters" : 16
    "unfiltered_count" : 46
    "extended_count" : 9
    "unfiltered_primary_count" : 46
    "search_radius" : 25
    ▸ "sort" : [...] 6 items
    ▼ "result" : [ 16 items
        ▼ 0 : { 78 items
            "hotel_name_trans" : "Sheraton Ann Arbor Hotel"
            "children_not_allowed" : 0
            "distance_to_cc" : "3.10"
            "main_photo_url" :
            "https://cf.bstatic.com/xdata/images/hotel/square60/327330452.jpg?
            k=f07f8070fc1c9c0961ef688d89b4b3a99c5d4965a22f9ceb1ea2e797fd418aa4&o="
            "selected_review_topic" : NULL
            "currency_code" : "USD"
            "preferred" : 1
            "timezone" : "America/Detroit"
            "city_trans" : "Ann Arbor"
            "native_ads_cpc" : 0
```

Interaction and Presentation Plan

The program will use the command line to ask users in order to refine the search result. In the command line, the program will ask users several question in order to refine the search results. Here is the list of questions and answer conditions:

- Do you want to find a restaurant or a hotel to stay in?
    - If restaurant, the program will goes into the tree with the restaurant API
    - If Hotel, the program will go into the tree with the hotel API
- In restaurant API:
    - Which city do you want to search for the restaurant? (text input)
        i. Return a list of restaurants' names
    - Is your expected price lower than $30?
        i. If yes, Is your expected price lower than $10?
            1. If yes, return a list of restaurants with prices lower than $10.
            2. If no, return a list of restaurants with prices between $10 to $20.
        ii. If no, Is your expected price lower than $60?
            1. If yes, return a list of restaurants with prices between $30 to $60.
            2. If no, return a list of restaurants with prices above $60.
    - Which restaurant do you want to check for further detail? (number input)
        i. Based on the user choice, return the detail of the restaurant, such as location, rating, price, phone number, category, etc.
    - Do you want to search for another restaurant, hotel, or exit?
        i. If yes, return to the restaurant branch.
        ii. If no, return to the hotel brach.
        iii. If Bye, break the program.
- In Hotel API:
    - Which city you want to travel and stay? (text input)
        i. Based on the user input, the string will goes to the "search location" API in order to get the dest_id for further search, but users will not know that.
    - What is your check-in date? (text input)
    - What is your check-out date? (text input)
    - How many people travel with you (please enter the total number of travelers)? (number input)
    - How many rooms do you want?
    - Based on these question, the program will return a list of available and possible result for users.