2. List of features you will definitely implement in the application

* Overview
  1. Search Restaurant by name, zip code, rating
     1. By name, show location and a list of reviews and other attributes in a radar plot
     2. By zip code, show a list of all restaurants and if we select one, then show location and a list of reviews and other attributes in a radar plot
     3. By rating, show a list of all restaurants and if we select one, then show location and a list of reviews and other attributes in a radar plot
* Function For User
  1. Recommend places to travel and want to eat in style and variety. Return zip codes (with accompanying restaurant information) that contain at least one 4.5+ star-rated (separate) breakfast restaurant, lunch restaurant, and dinner restaurant. This will render results using pagination functionality that will be created using LIMIT.
  2. Query a single restaurant and add/remove attributes to customize what information is displayed on the screen. For example you want to see restaurants that have Bike Parking, Wifi, or “Trendy” ambiance. This is accomplished by joining the main Yelp data set against a list of sub-attributes. Recommend Top 20 lists (location, cuisine) order by rating
* Function For Restaurant Owner
  1. Cluster neighborhoods by certain attributes and recommend a niche market
  2. Show nearby high-rating restaurants
  3. Show newly created indicators for evaluating restaurants
     1. Revisiting rate: number of users who have visited the restaurant more than once
     2. Improvement score: (average star rating after the first visit / star rating when they first visited) for users who have visited more than once.
     3. Number of regular customers visited : number of users who visited more than 3 times for the last 12 months.
  4. Recent review and rating change in line graph

3. List of features you might implement in the application, given enough time

* Linking with a google map api, return postal code, longitude, latitude for connecting Google Maps API and create a slider for radius, which will return more restaurants in the query or less based on the option selected.
* Mapping with Covid dataset, show restaurants’ current policy toward COVID19

4. List of pages the application will have and a 1-2 sentence description of each page. We expect that the functionality of each page will be meaningfully different than the functionality of the other pages.

* Page1 : Overview
  1. Basic descriptions on all three pages and team members
  2. Description of each restaurant searched by name
* Page2 : Recommendation for User
  1. Recommend a location for a special journey with exceptional cuisines
  2. Check out an available restaurant in a certain location with key attributes for which a user is looking and recommend Top 20 lists (location, cuisine) order by rating
* Page3 : Recommendation for Restaurant Owner
  1. Find some niche locations to open up a new restaurant
  2. Show reviews of a near-by restaurants for a service improvement
  3. Show some newly created indicators to analyze the current status of a restaurant

7. Explanation of how you will clean and pre-process the data. This tutorial demonstrates how to do simple pre-processing in Python.

* Main tool : Python pandas library and Google Collaboratory for collaboration
* Through EDA(Explanatory Data Analysis), explore the given data set and draw ER diagram to determine which features we will use or not.
* Delete columns which won’t be used at all and make table light by adopting BCNF or 3NF normalization so that we can boost speed.
* Check if there are any columns with null values and then decide if we should drop those rows or replace them to default values.
* For number data, using max or min, check if there are unreasonable records.
* Change column name more clearly and less redundant.
* Unite values with the same entity into one. (ex: ‘South Korea’, ‘Republic of Korea’ -> ‘South Korea’)
* Unify related Boolean value columns into one enumerated type column if they are incompatible.

8. List of technologies you will use. You must use some kind of SQL database. We recommend using MySQL specifically because you will use MySQL in HW2, and we will provide guidance for setting up a MySQL database.

* Main DBMS
  1. MySQL : We will use MySQL database to control and query the given dataset for this final project.
  2. DataGrip : DataGrip will be used for browsing database and testing queries.
* Python : Python will be used for revising dataset and simple operations.
  1. Pandas : Pandas library will be used for mainly pre-processing and cleansing the given dataset.
  2. Google Colab : Google tools for sharing and edting ipynb files to share the process of pre-processing and cleansing the given dataset.
* JavaScript : JavaScript will be used for both server-side and client-side. JavaScript will be used as main language for both sides in pursuit of concurrency .
  1. Node.js : Server-sided back-end runtime library.
     1. NPM(Node Package Manager) : Version control and package management library for Node.js.
     2. Express.js : Simple and fast web framework library.
  2. React : Front-end library for making user interface.
  3. Google Maps JavaScript : Adding attributes of Maps in the webpage, which might not be used at all.