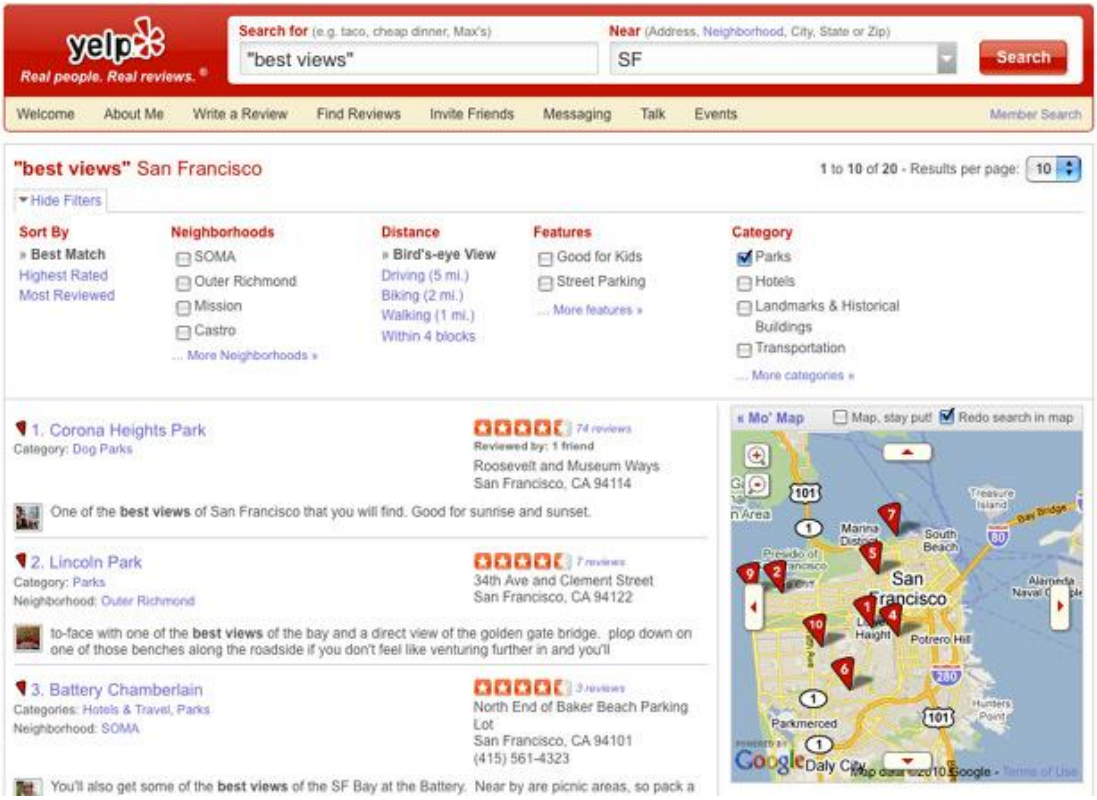
CS 4731-A Big Data (FALL 2016)

Business Search Engine based on Yelp and Panoramio

(The final comprehensive project)

Due： 11:59 pm, Monday, November 28, 2016

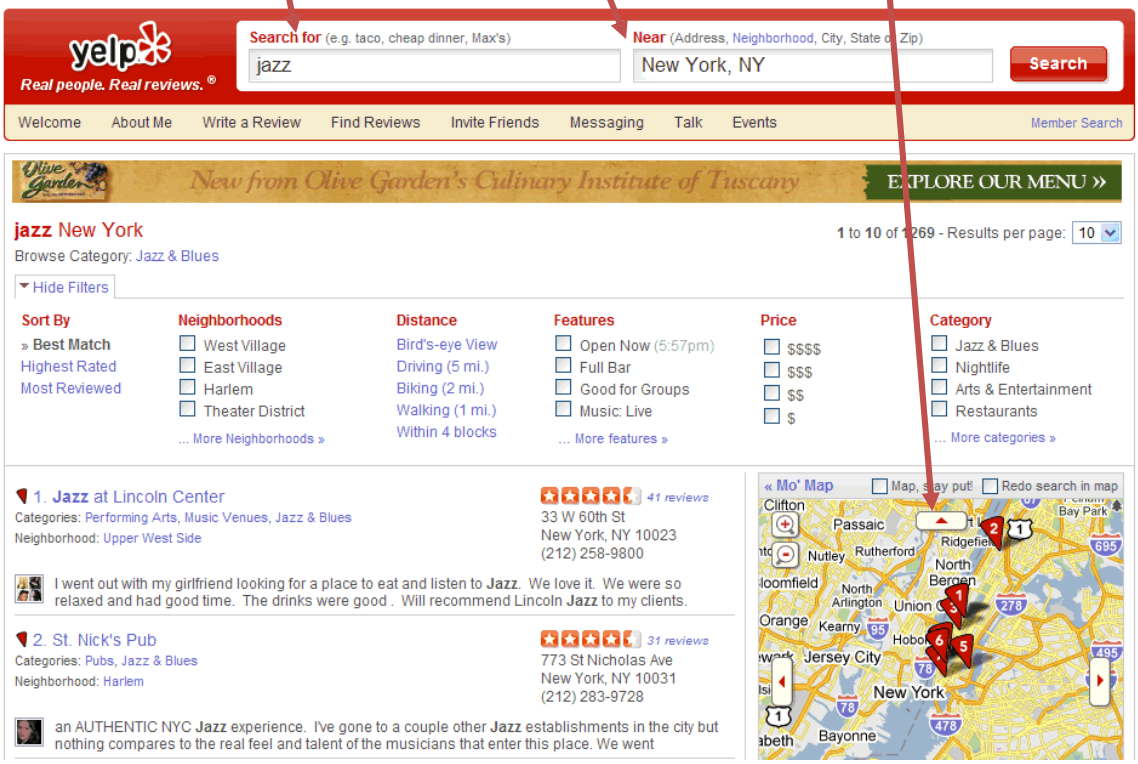


1. Project Requirements.

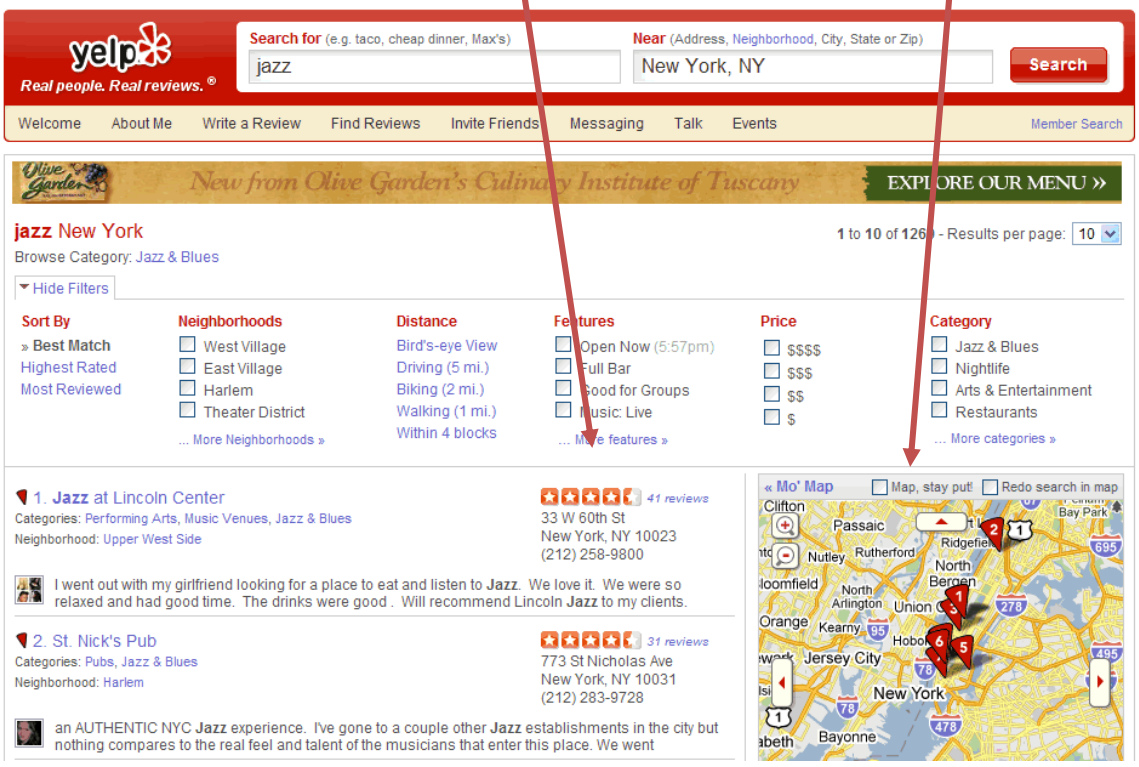
You are expected to use PHP, MongoDB/MySQL, Google Maps API, and Panomario API to design and implement your business search engine which provides the following features:

1. Given a keyword and a city name, your system should return a list of businesses whose name "contains" that keyword and whose location is within that city. For each business in the result list, you need to show (1) the business name, (2) the business address, and (3) the business star rating (0-5). Businesses should be sorted on star rating.

   Besides the textual business list, show a map with all the returned businesses. Each marker should have a pop-up info window that has its corresponding business name and the full address.



2. Proceed to provide a "Star Rating" filter to refine the search result above. If a user specifies the rating filter as 3 .0 stars, then a business is returned only if its stars rating is no less than 3.0. For each business in the result list, show (1) the business name, (2) business address, and (3) its star rating. For example, if the keyword is "jazz", the city is "New York", and the rating filter is set to "3.0", then only the businesses whose name contain "jazz", located in New York and holding a rating of no less than "3.0", will be returned. Businesses should be sorted on star rating. Show a map with all the returned businesses. Each marker should have a pop-up info window that has its corresponding business name and the full address.



3. For each returned business, provide a way for users to access its Google street view

For an example,

go to http://mypages.valdosta.edu/hachen/streetview.php?lat=40.7484&lng=-73.9857

4. For each returned business, provide a way for users to access the Panoramio photos taken within 1 mile from this business. You may assume that 1 mile equals 0.02 degree in latitude and 0.02 degree in longitude for simplicity.

Hint: you may pass the business latitude/longitude as GET parameter or define a form with hidden fields.

5. For each returned business, provide a way to direct user to the Twitter search page that shows all the tweets containing the name of the business as keyword. For example, for the business named "Pine Cone Restaurant", simply direct users to

https://twitter.com/search?q="Pine Cone Restaurant"

You do not have to use Twitter API for this feature.

6. For each returned business, provide a way for users to access all the reviews associated to that business

7. (Optional) For ADDITIONAL FEATURES, I will give you 1-10 extra points based on your implementation. Those features must be non-trivial. **Creativity is encouraged here.** The following are some possibilities:

- For any particular user, show all his/her posted reviews.
- Given a category and a city name, show all the businesses in that category in the city, sorted on star rating.

Do not forget to give your search engine a "GREAT" name ☺

2. Roadmap

1. Follow the instructions to download the Yelp dataset. Examine and understand the data format. To understand the JSON format of each entity, go to http://www.jsoneditoronline.org
2. Write PHP script to insert business and review data into your MongoDB/MySQL.
3. Use HTML/PHP/MongoDB/MySQL to implement your search engine.
4. Use HTML/JavaScript/JQuery to give your search engine a nice-looking front end.
5. Use Google Maps API to implement maps and Panoramio API to fetch photos from a particular region.

3. Our Yelp Dataset

- Stats about the dataset:
    - **2.2M** reviews and **591K** tips by **552K** users for **77K** businesses
    - **566K** business attributes, e.g., hours, parking availability, ambience.
    - Social network of **552K** users for a total of **3.5M** social edges.
    - Aggregated check-ins over time for each of the **77K** businesses
    - Cities:
        - U.K.: Edinburgh
        - Germany: Karlsruhe
        - Canada: Montreal and Waterloo
        - U.S.: Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, Las Vegas, Madison

Each file is composed of a single entity type, one JSON string per-line. For example, the following show the JSON format of each business

business

```
{
    'type': 'business',
    'business_id': (encrypted business id),
    'name': (business name),
    'neighborhoods': [(hood names)],
    'full_address': (localized address),
    'city': (city),
    'state': (state),
    'latitude': latitude,
    'longitude': longitude,
    'stars': (star rating, rounded to half-stars),
    'review_count': review count,
    'categories': [(localized category names)]
    'open': True / False (corresponds to closed, not business hours),
    'hours': {
        (day_of_week): {
            'open': (HH:MM),
            'close': (HH:MM)
        },
        ...
    },
    'attributes': {
        (attribute_name): (attribute_value),
        ...
    },
}
```

4. How to Download our Yelp Dataset

Go to the link:

https://1drv.ms/u/s!Au2bbpo1ICaZhDMgcAAZ0rxkkz3P

5. Managing Data at the Back End

You may use Robomongo GUI/phpMyAdmin to manage your data at the back end.

6. Front-end Website Implementation

Try to make your interface appealing aesthetically. You may want to use Bootstrap/ Weebly to generate a template to bootstrap your coding.

7. Publishing Your Search Engine (optional)

It is strongly encouraged that you test and debug your system on your localhost before you upload it to a public Web server.

8. Materials to Hand In

1. Your PowerPoint file for the presentation should include the following sections:

    1. Overview (GUI and functionality) of your search engine
    2. Techniques used in the project
    3. Show all the data in your MongoDB/MySQL
    4. Demo (for each feature you claim credit for, at least two sample queries)
    5. Major challenges faced and how you addressed them
    6. Work division over group members
    7. Reflection: what you learned from this project

2. All your source files of the system. Include all your code in a zip file.

All the materials above should be submitted on BlazeView before

11:59pm, Monday, November 28, 2016

NO late submissions will be accepted.

9. Grading Breakdown

Refer to my evaluation form on BlazeView.

10. Demo Date and Time

Monday, November 28, 2016 @ 2:00pm - 3:15pm

Wednesday, November 30, 2016 @ 2:00pm - 3:15pm

11. Teamwork

Each group should have no more than three members. All members will receive the same grade. Each team leader must email the instructor. By **Midnight Monday October 24, 2016**, each team leader must send the instructor (hachen@valdosta.edu) an email that describes the following:

1. Names of all the members in your group
2. The name of your business search engine.

If the instructor fail to hear from you by then, you will be assigned to a random group.

12. Help Sessions

During our meetings, I will check your progress and guide you through the project.

CS4731 Project Meetings (attendance is required):

Monday, November 14, 2016 2pm-3:15pm

Monday, November 21, 2016 2pm-3:15pm