The University of Georgia

06 Dec, 2017

**FINAL PROJECT REPORT**

**University Selection Using Semantic Analysis**

***Team Members:***

Rakesh Bhavsar

Prathamesh Dhanawade

Ayush Jaiswal

Shrutika Gamare

1. **INTRODUCTION**

A semantic web application which facilitates looking up information on universities and provides guidelines for users while visiting these universities.

It provides to select from the preferred weather statistics, information on crime rate and restaurants located near these universities

A semantic data model is developed initially to establish the relationship between the gathered datasets, ontology is created and the linked data is queried using SPARQL query language.

1. **LITERATURE SURVEY**

There are similar applications like Stupidsid, which is popular among students. It focuses on college reviews including information on infrastructure, location, faculty etc. US news website provides data about graduate schools, online programs, ranking and advice for finding the best schools. This also helps students to narrow down their college preferences based on factors like location, tuition fees, rankings etc. BigFuture provides information on career opportunities and financial costs incurred. It gives a step-by-step college plan right from exploring the career areas to paying for college. Though these applications provide good amount of information with respect to universities, our main focus was to provide data for a large set of user groups to help them know about secondary factors that could highly influence their choice of university selection. Thus, our application was developed keeping in mind the diverse user groups and guiding them throughout the planning process.

1. **HOW OUR APPLICATION IS UNIQUE**

Our application gets information on university surroundings by three search criterias namely weather, restaurants and crime rate. An interesting thing is that the application covers a wide range of user groups including researchers, scholars, university guests, exchange students, recruiters and of course students. Anyone developing a similar application in the future can reuse our data model and documentation to increase the functionality of their own application.

The web application has the following special features:

* Highly interactive UI for appropriate query responses.
* SPARQL endpoint.
* Application deployed on Amazon EC2 Instance.

1. **SYSTEM ARCHITECTURE**

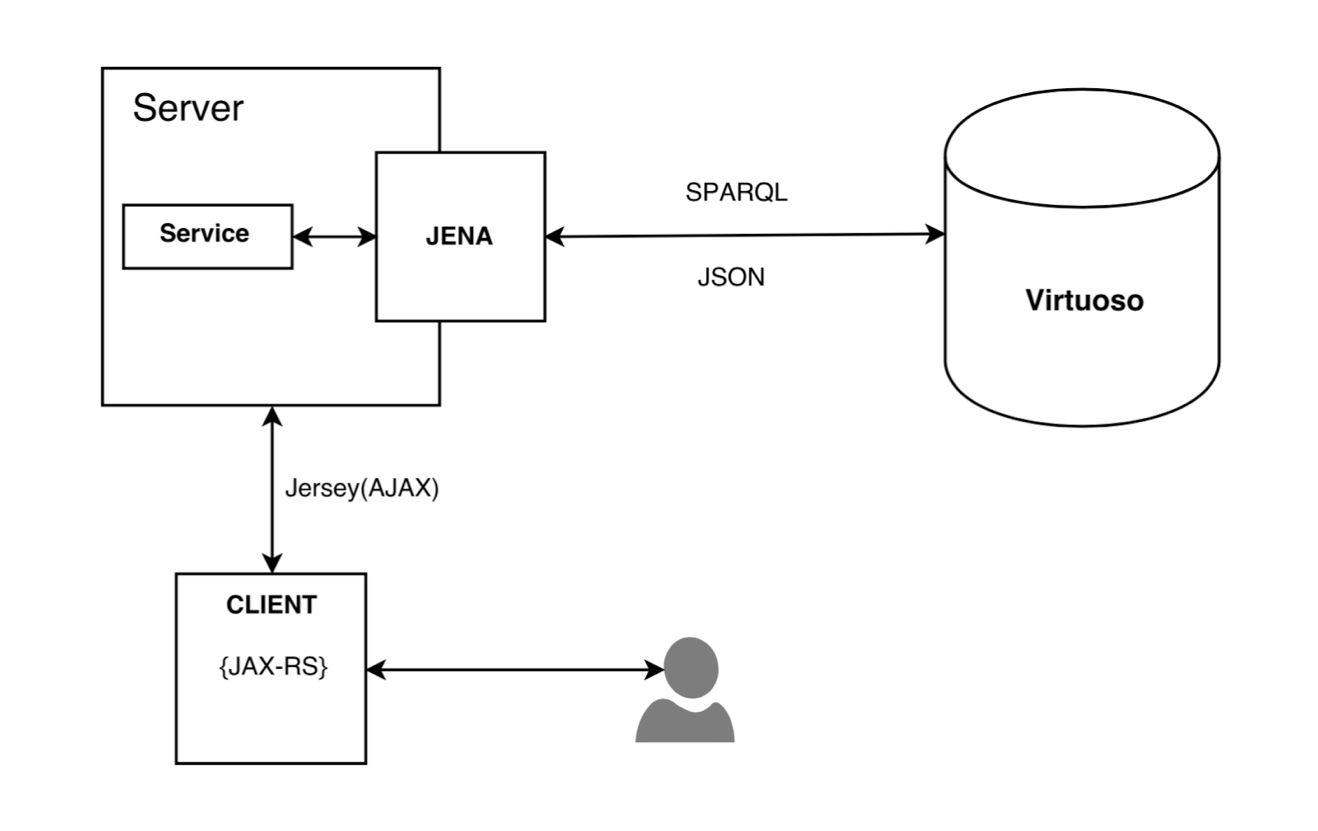


Figure: System Architecture

1. **FUNCTIONAL SPECIFICATION**

The user has functionality to search university based on various criteria.

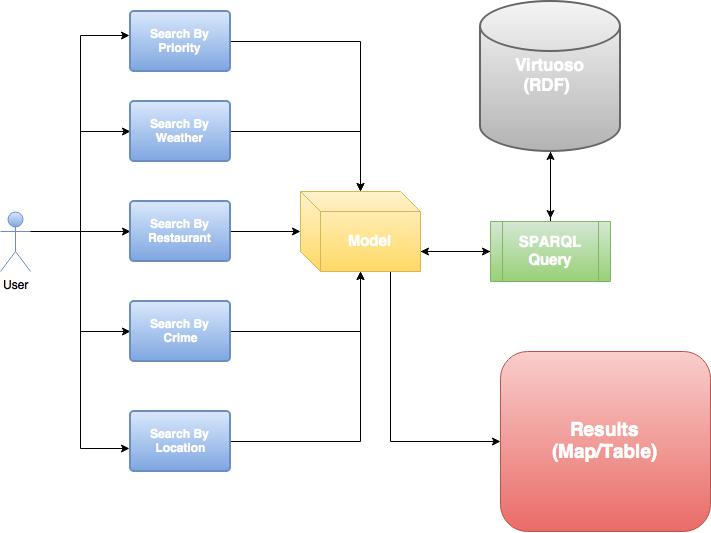
1. Search by Priority: The user has the priority based university search. The user selects priority by weather, restaurant and crime. The search results are populated on Google map based on the state selected.
2. Search by Weather:

This offers user to choose from three filters namely: Rain(cm), Average temperature (celsius), Humidity level(mmHg). The response is a set of colleges that meet these selected requirements.

1. Search by Restaurant:

This search allows to look up moderately priced to highly priced restaurants in a particular city.

1. Search by Crime: Crime rate accounts for major university selection factor. Search by Crime take input as total crime rate, university crime rate and public property crime rate. The university results are populated based on the input criteria.
2. Search by Location: Sometime users want university list based on city or state. The search by location provides this functionality.



1. **DATASET**

***University Dataset:***

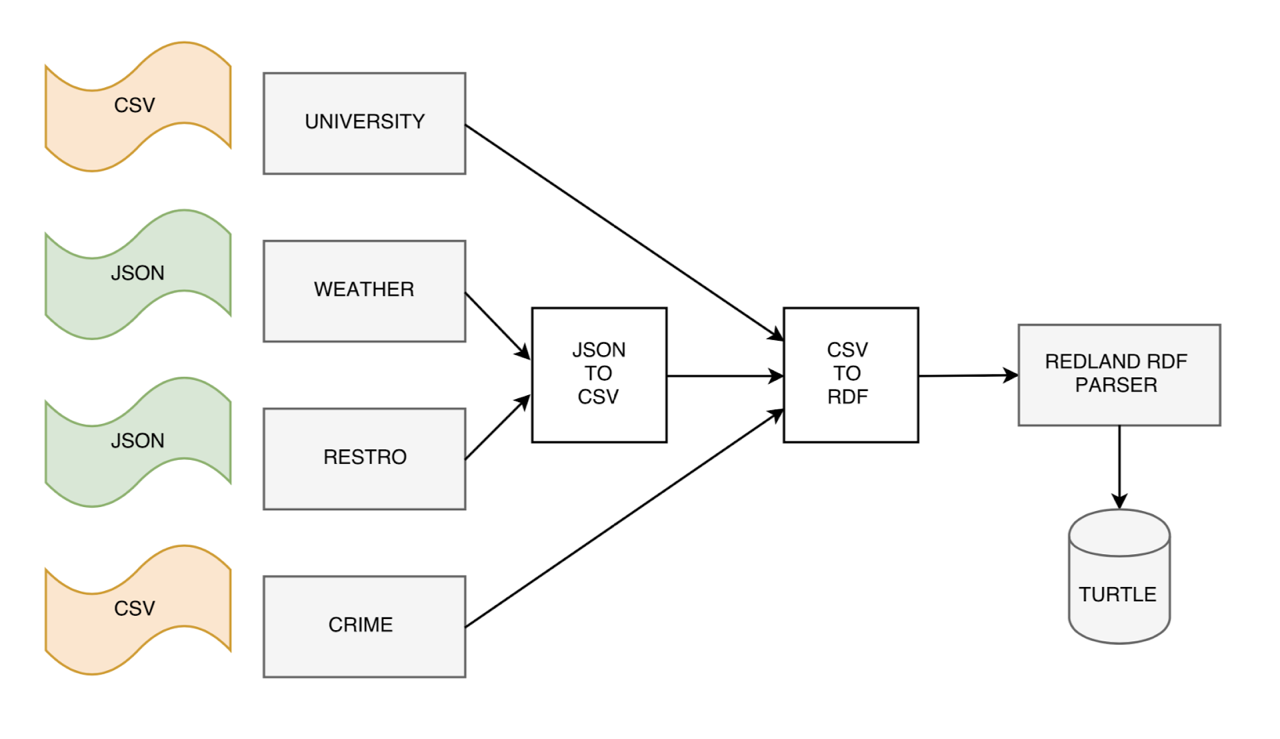
This dataset contains fields such as contact information, address of the university, latitude,longitude and university identity information. <https://catalog.data.gov/dataset>

***Crime Dataset:***

This dataset contains fields such as on-campus crime rate, crime reports, public property crime, residential hall crime etc. <https://ope.ed.gov/campussafety/#/customdata/search>

***OpenWeatherMap API***: This API provides all types of data associated with weather like minimum temperature, maximum temperature, average temperature, pressure, humidity, clouds, rains etc. <http://api.openweathermap.org/data/2.5/weather?q=BERKELEY,us&mode=json&APPID=1b31aec5119c555c4443daa73b9b576d>

***Yelp dataset:*** It contains information pertaining to restaurants near a school, information such as timing, price range, reviews, ratings and facilities such as delivery, wheelchair accessible, alcohol, wifi, takeout, accepts credit card etc. https://www.yelp.com/developers/documentation/v2/search\_api



1. **CHALLENGES**

Since, the data is miscellaneous (i.e.) from heterogeneous sources, there were many issues faced during the integration process.

Data conversion from JSON to CSV Karma Tool

Data conversion from CSV to RDF

http://levelup.networkedplanet.com/ Combine data into one file

Load data onto Virtuoso

1. **FUTURE SCOPE**

Our Project offers to serve a broader future scope as it can accommodate many such criterias that students seek in university decision making.

To mention a few, information on on-campus/off-campus housing options, information on types of assistantships or on-campus jobs available, facilities for international/domestic students, information on student bodies, alumni assistance, on-going research, courses offered, student reviewed hangout locations, campus orientation wrt job offerings/industries, all such data would be bought to a single point to help students, also researches to study the inflow of dynamics in university decision making by potential students.

1. **CONCLUSION**

Our main goal of the project was to incorporate all the Semantic Web concepts learnt as a part of Advanced Information Systems and this project fulfills to serve our desire to learn about Ontologies and Sparql queries. Also, it was through this project that we had a chance to learn about amazon web services and deploy our application on EC2.

1. **REFERENCES**

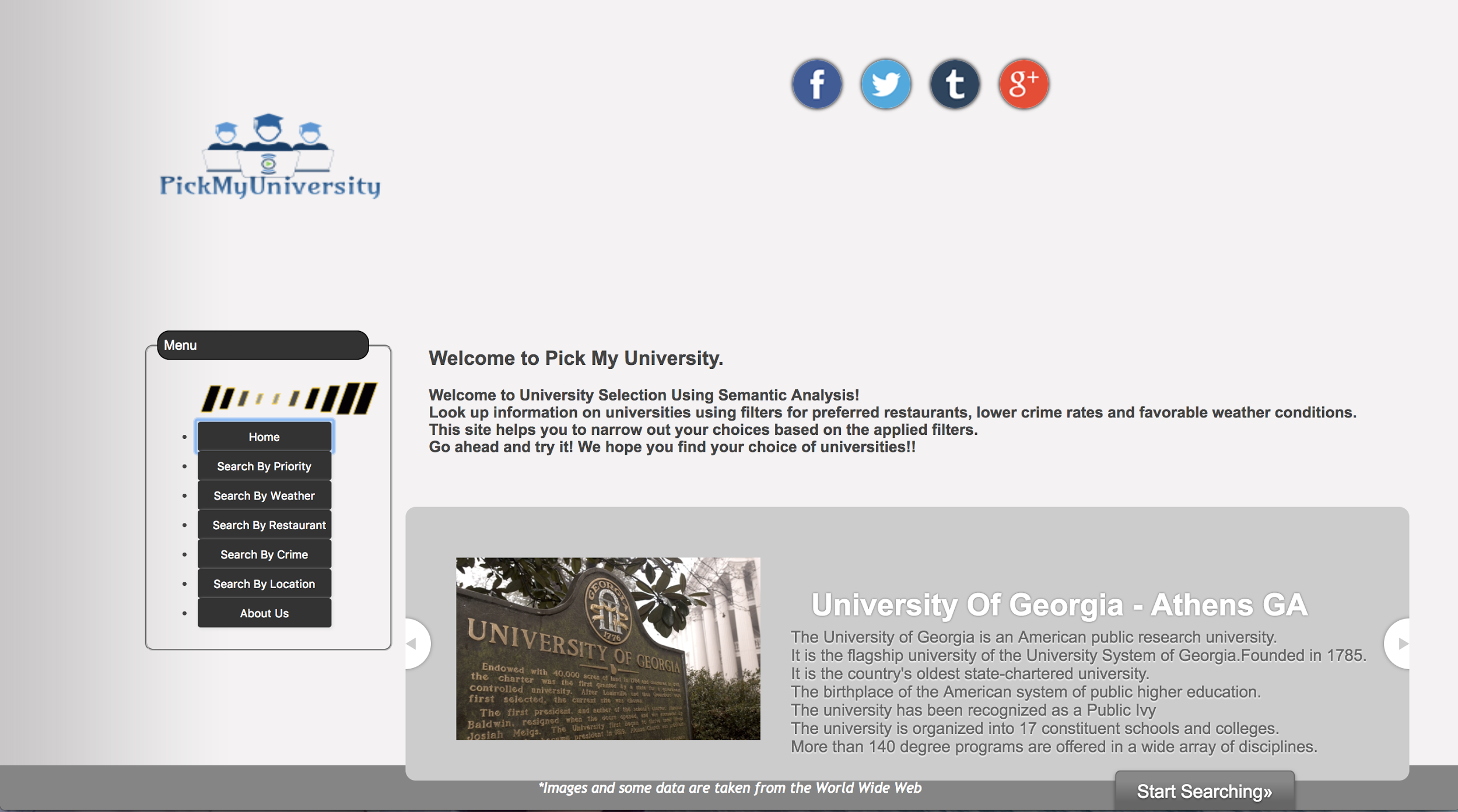
[1] Yelp API - Documentation: https://www.yelp.com/developers/documentation/v2/overview

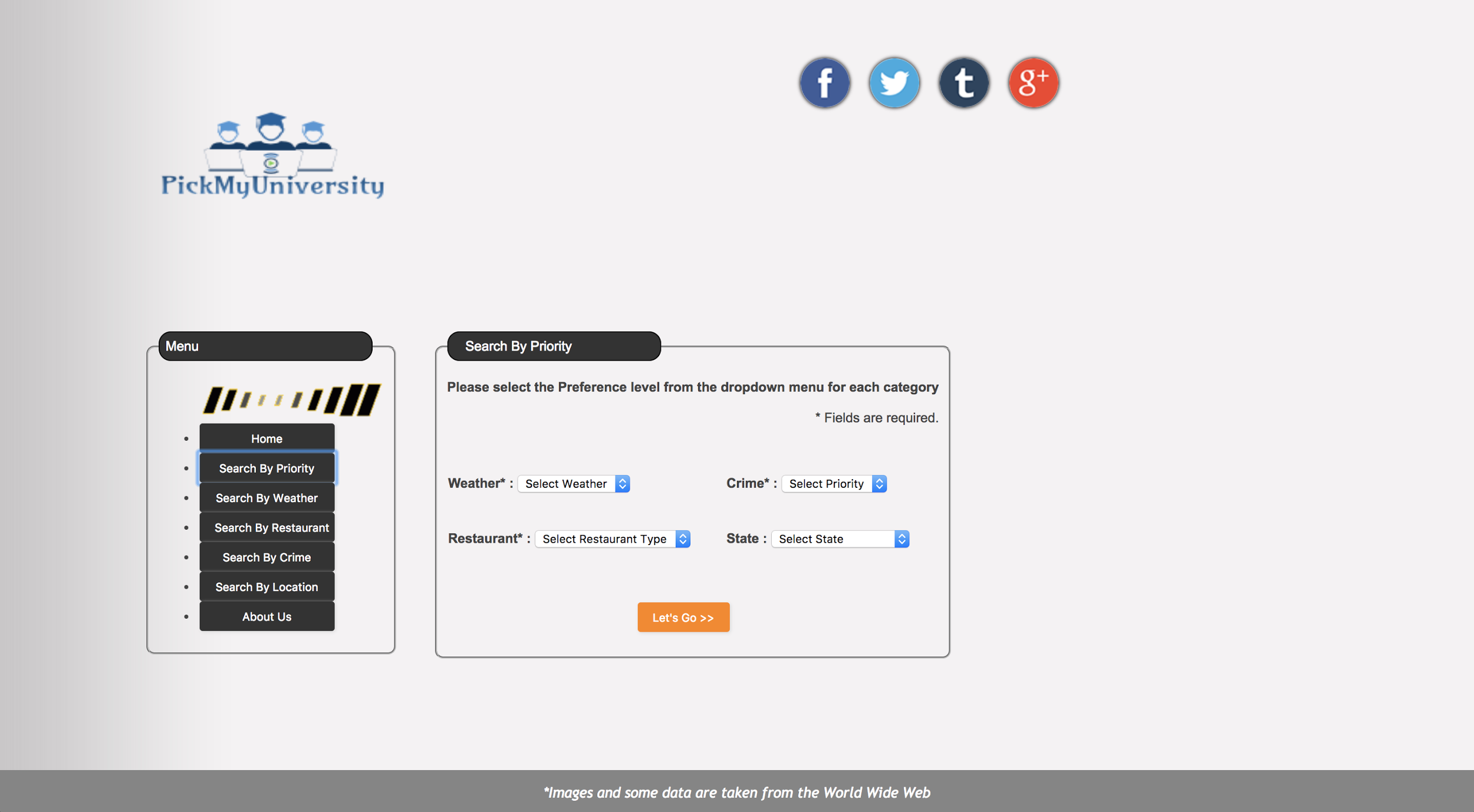
[2] Yelp API Code Samples : https://www.yelp.com/developers/documentation/v2/examples

[3] OpenWeatherMap API Documentation : https://openweathermap.org/current

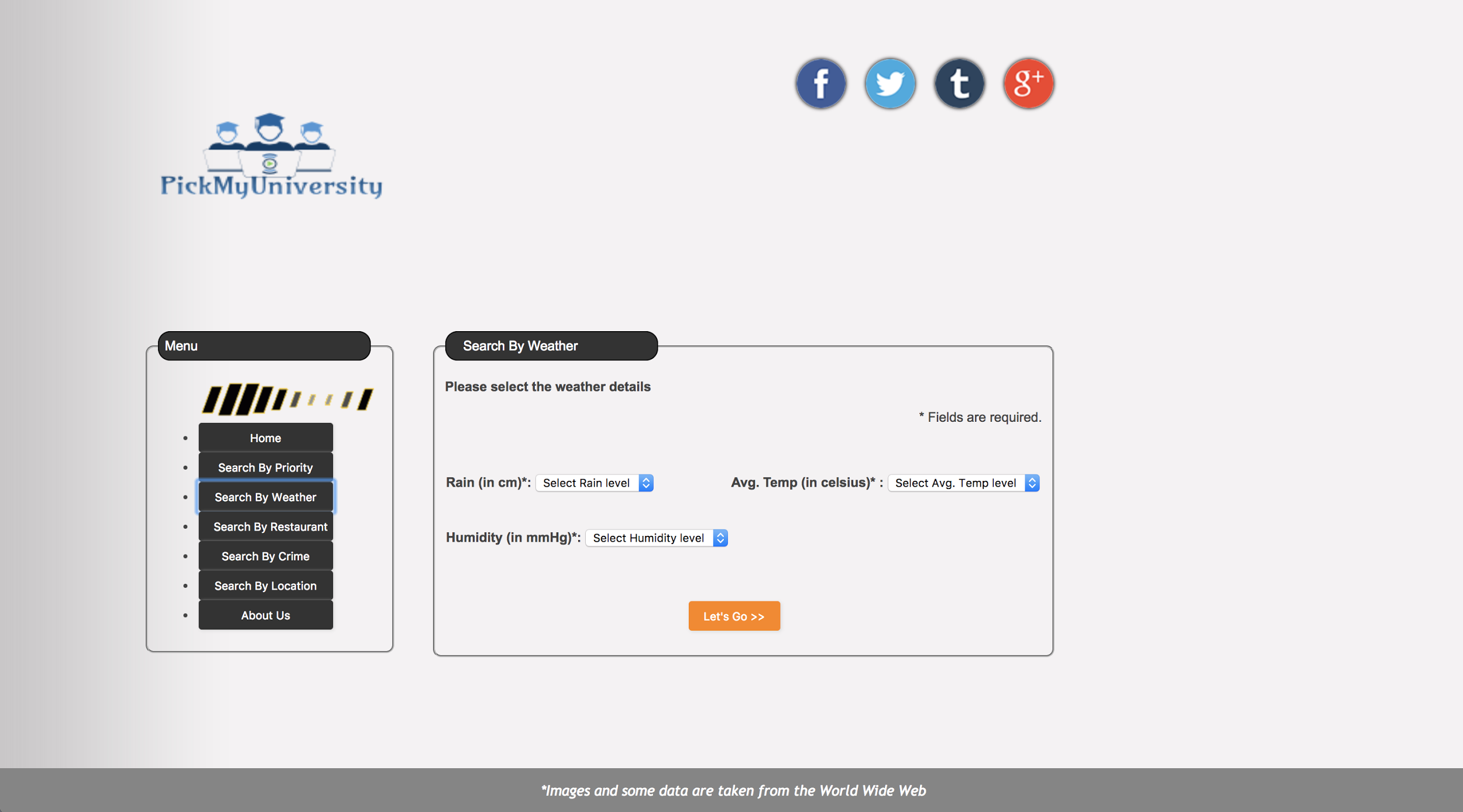
[4] OpenWeatherMap Code samples : https://openweathermap.org/examples

1. **USER INTERFACE SCREEN-SHOTS**

*****Figure 1: Home Page*

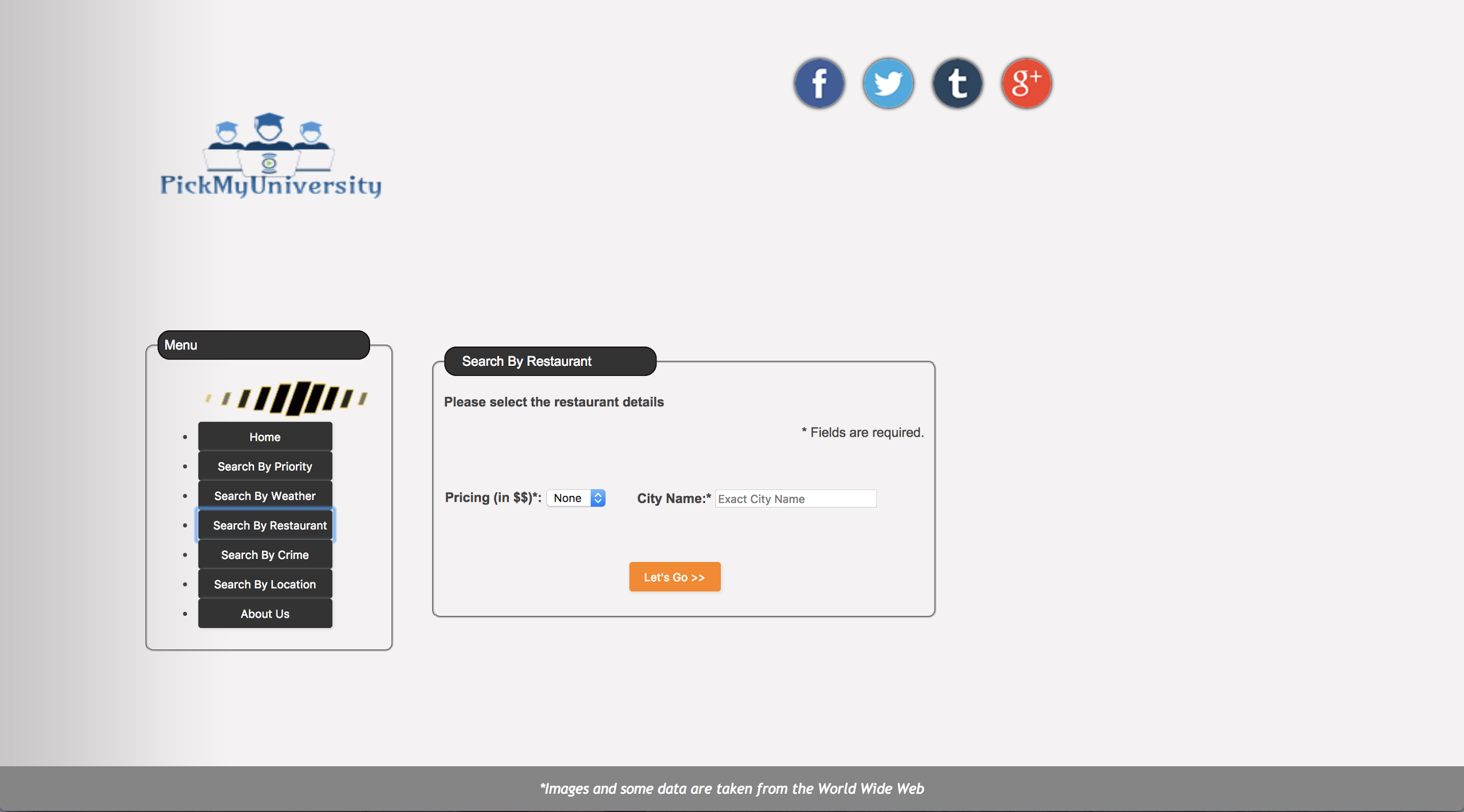
****

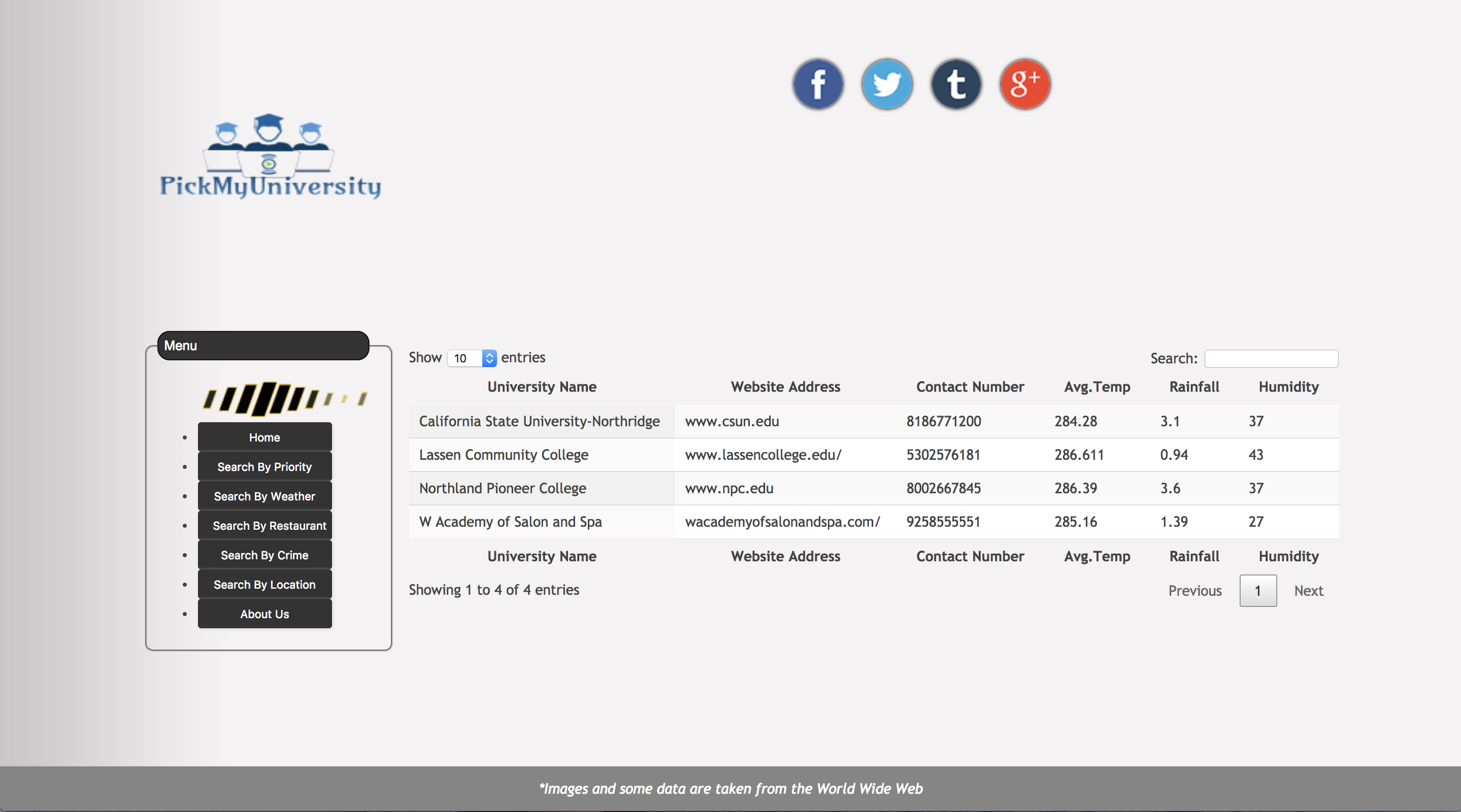
*Figure 2: Search by Priority*

****

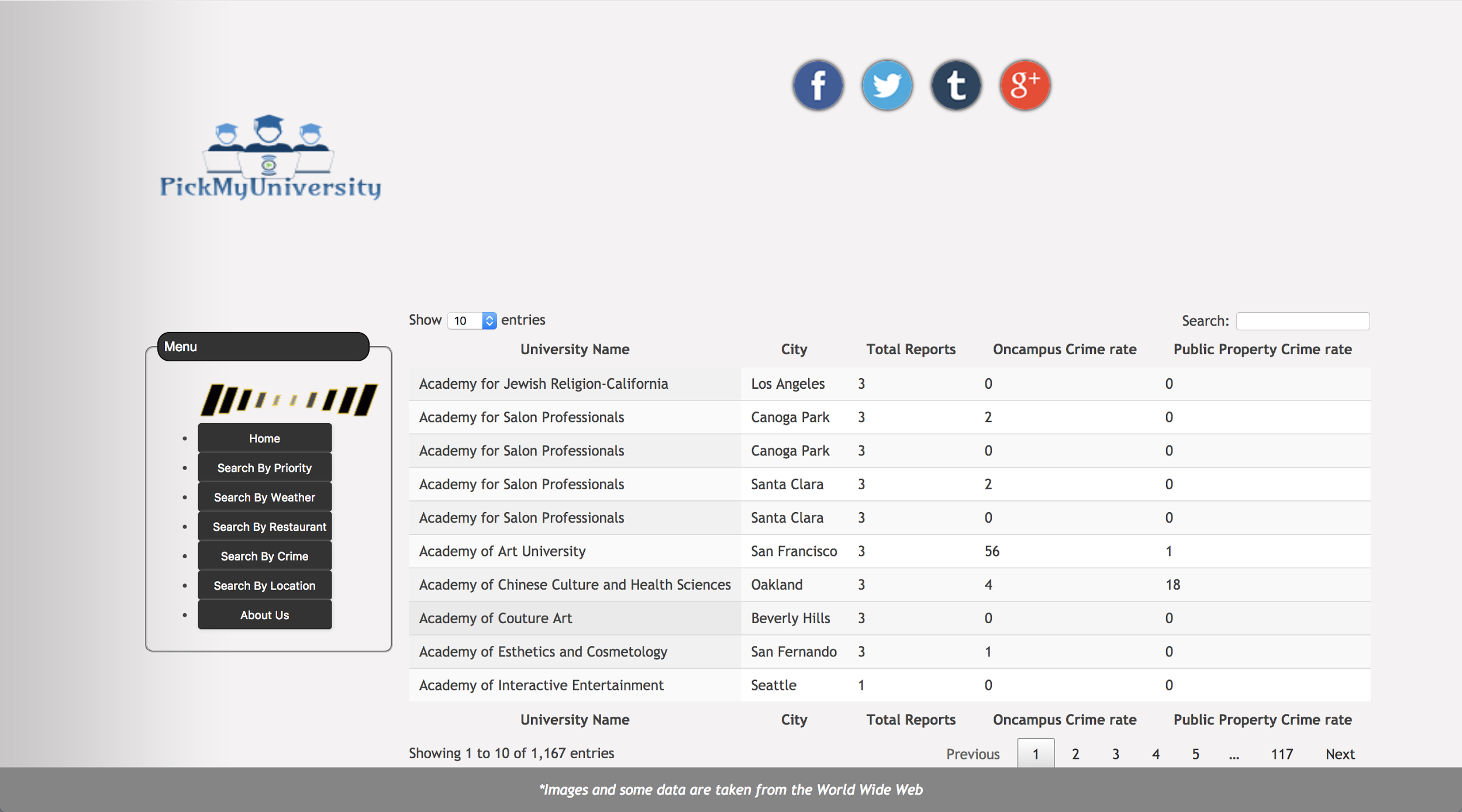
*Figure 3: Search by weather*****

*Figure 4: Search by Location*

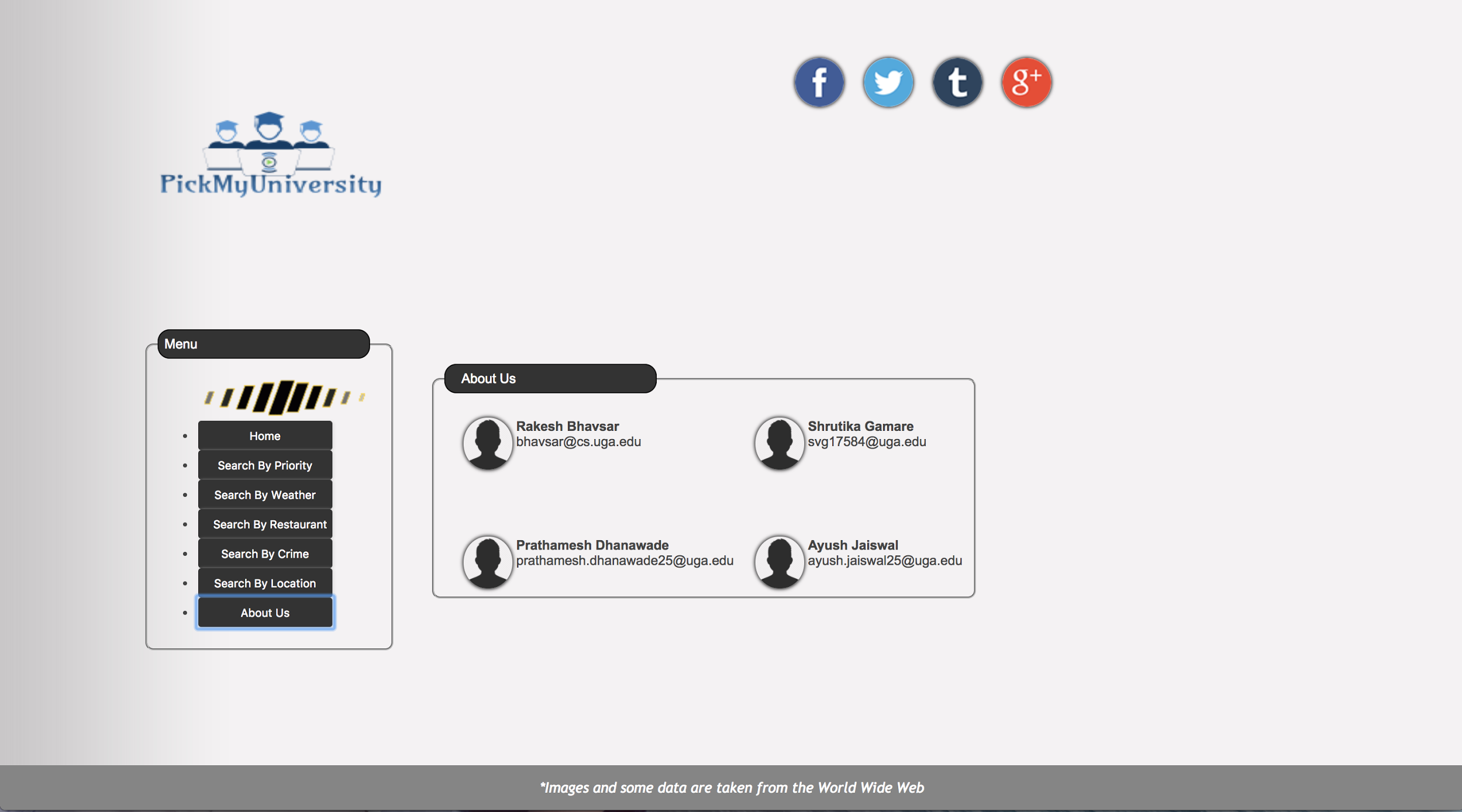
**** *Figure 4: Search by Restaurant*

****

*Figure 5: Results Page*

****

*Figure 6: Results Page*

****

*Figure 5: Team Details*