Term Project Report, CSCI 4300

Group Name: Lean Squad

Group Members: Cesar Acosta, Komran Ghahremani, Joshua Bennett, Kevin McDaniel

**Polarize**

Our term project is about using the vast capabilities of the Twitter API to create a homepage that people can set in their browsers to see the content that we have created for them. We used the Twitter API to query various key phrases for popular opinions of the day. Our applications can be used for a serious issue, opinion on an event or even a political candidate. We constructed an algorithm to grab two polarizing opinions regarding a specific topic and then our application will do the job of presenting the polarizing opinions in aesthetic form. This is the reason we are naming our application *Polarize*. With almost every opinion there will be people on both sides of the argument and our goal with this application is to capture both sides of the argument and present them to you.

In terms of the original contributions, Josh was in charge of adding to the visual design of our pages through Bootstrap, Komran designed the initial Polarize page and worked with the PHP Insight API, Kevin was in charge of the backend portion which mainly involved PHP, JavaScript, and SQL, while Cesar worked with the Twitter API and parsed the data we received through JSON. This involved creating a Twitter account that was used for testing purposes and for signing up through the Twitter’s developers page in order to gain access to keys that were used for authenticating certain requests to Twitter’s server, in order to search for tweets, trends, and the location closest to the user’s latitude and longitude. Kevin developed the database, as well as the trends page, which displays current trends from the user’s location and sends them to the database, and the database search page, which searches for information previously entered into the database through all our users’ visits to the trends page.

Since our original contributions to the website we have made significant improvements to the site. The site was originally a single page but has now expanded to 4 pages, a Home page, a page that is used to display statistics regarding the current trends, a page which displays searches for trends pulled from the database, and the Polarize page itself. We have also made some CSS style improvements to the website to give it a better overall feel. We have implemented the use of bootstrap to add a navigation bar at the top of the page displaying the pages you can navigate to. We also stylized the search bar a bit to keep it from having a dull feeling. For the statistics page we display the location from where you made the request and then the current trends based on your current location are displayed. For each trend we display the trend name, the number of tweets that used that particular hashtag (trend) and the URL (direct link) to that trend on Twitter.com. We also attached a database to our application that is able to store the current trends now so that you are able to view later through the search page which trends our users have searched for, and at what date, location, and to what degree they were trending. One of the very interesting features that we were able to implement is the use of the PHP Insight API. A problem we faced when using polarize is the wrong outcome when it came to searching for whether someone loved or hated a certain topic. For example, if we were to input “Donald Trump” in the search bar a possible result on the HATE side of our application could yield a result such as “ ’ I hate Donald Trump’ is not something I would say, I love him”. In this example it is clear that the person is in favor of Donald Trump, instead of them disliking him. Previously our algorithm wouldn’t be able to distinguish this, but with the use of the PHP Insight API, we were able to decipher the true meaning of the tweet when the results came up. This greatly enhances our search results.

This application can have a broad impact depending on those who uses it. This application can be useful for certain companies in determining what topics are trending, what is being talked about and what is liked so that these companies can incorporate this data into their marketing strategy which can have a profound impact on their company, possibly allowing revenue to go up. This can be useful to many people as well as it presents information on the current trending topics, which can help people be up to date with the latest news or talked about topic.

In terms of the technical side of this program, we implemented MySQL to serve as the backend database with PHP used for communicating with the database and also serving as a handler of the Twitter API where we can make a certain number of queries within a span of time and the results would be displayed to the screen. Upon visiting the trends page, the displayed trends, which are based on the user’s location, are stored in the database, so that as more users access the website from around the world, our database will be filled with data via automatic crowdsourcing. Users can then use the search form on the trends page to search for trends from the database from specified locations, date ranges, and having at least a certain popularity level. HTML and advanced CSS, such as the use of bootstrap, allowed us to display our information in a more user friendly manner with great aesthetic features. JavaScript was also used in order to correctly display the information needed for the trends page. We used the PHP Insight API and made calls to it to evaluate positivity, negativity and neutrality scores identified within the text of each tweet, basing our results on these factors to ensure that our results were as accurate as we could make them.

For this project Cesar worked the initial algorithm with PHP to make a call to Twitter’s API and return the results of “Love” and “Hate” whenever a string was entered into the search box. He set the limit on the number of results and made sure that the results of “Love” would appear on the left side and the results of “Hate” would appear on the right side. He set up the basic interface that made it easy to make calls to Twitter’s API so that everyone could easily access it; this included setting up the access key, access token secret, consumer key and consumer key secret. Kevin worked on the backend database side of this project by also making calls to Twitter’s API in order to get trends and connecting them with the database so you can store the trends by date. He wrote the code to find the user’s location, and made the hidden form which sends through that information to the trends page as a POST value. He also worked on the display, writing the PHP that produces boxes for each result on all pages, and the Javascript which makes each half of the Polarize page match in size. Josh was in charge of the aesthetics of the Homepage and the overall look of the website. He designed the navbar, and made sure everything was prepared and used Bootstrap to give the website a professional feel. Komran initially worked on the CSS of our application in its early stages but he managed to also help out with the algorithm and worked with PHP Insight in order to ensure that the results of the query were as accurate as possible.

In conclusion we believe this project allowed us to gain the much needed experience necessary to be comfortable in building websites. No matter how much time we spend with our heads in our books, we won’t be able to gain nearly as much knowledge as we did working with this project. This project allowed us to tie in all that we learned throughout the semester such as HTML, CSS, JavaScript, PHP and MySQL. These are the necessary elements needed to build a robust webpage and it also helped in giving us a feel of what will be required in the real world. We also were able to gain experience working with APIs as there are a vast number of APIs online that are waiting to be taken advantage of in order to create very useful applications. We strongly believe our application is something that can be of great use or entertaining to use and we are grateful for everything we learned from it.