CAB432 REPORT ASSIGNMENT 2

Twitter application

Sean O'Connell

2018

Contents

[Introduction 2](#_Toc529135396)

[Description 2](#_Toc529135397)

[Scaling and Performance 3](#_Toc529135398)

[Limitations 3](#_Toc529135399)

[Extensions 3](#_Toc529135400)

[Appendix 4](#_Toc529135401)

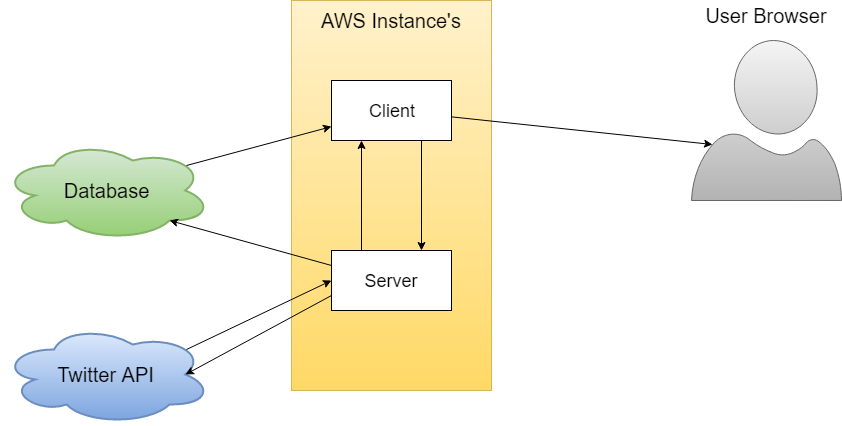
# Introduction

This application’s main purpose is to demonstrate the sentiment of selected live Twitter trends. The application uses the Twitter API and a natural language processing library to achieve this. The application shows this with a live line chart used from the google charting API. This causes load on the server equipment, which causes scaling on the AWS instances.

The application structure uses two express apps and an external javascript application that works in unison to push data to an offsite database and to retrieve that data from the database. They all provide the application with data transfer that causes load on the application.

The phases of the application caused a few stoppages along the development. The creation and implementation of the AWS instances with active load balancer. This caused problems as the applications doesn’t produce enough load to fully take advantage of this feature.

# Description



The application structure diagram above depicts the movement of data between the AWS instances and the off-site resources. The data movement between the application, the user and the off-site resources. There are two express apps, one serves as a client and the other serves as a restful server application. The client application uses a multiple node packages to get data to and from the database and the Twitter API. This allows the application the run and complete its purpose.

The purpose of the application was to demonstrate the sentiment of the tweets made in a specific trend. The application unfortunately doesn’t complete this task. However, this service was to be completed client side of the application. The movement of data to and from the Twitter API and to and from the database is completed on the server-side application.

Throughout development issues were constantly encountered, the most troubling issue was the completion and demonstration of the charts in a live manner. This caused multiple issues as it was troubling to demonstrate the full functionality of the application.

The implementation of the project is detailed as a diagram in the above the image. However, the specifics of the application are not demonstrated in the diagram. The server express application completes all the Twitter data retrievals and sentiment analysis. This application also adds this information to the off-site NoSQL database. The client express application works to retrieve the data from the database and package it into a clean and responsive web page. On the client side of the application the google charts library would display the tweet sentiment live on retrieval.

# Scaling and Performance

The application currently doesn’t scale correctly as the application doesn’t create enough load for application to correctly scale. However, the two EC2 instances running for the application do have a load balancer running on them. The load balancer scales the application based on network. This is measured based on the server’s number of TCP requests. This unfortunately didn’t work in full however this is only because the application was not 100% complete.

# Limitations

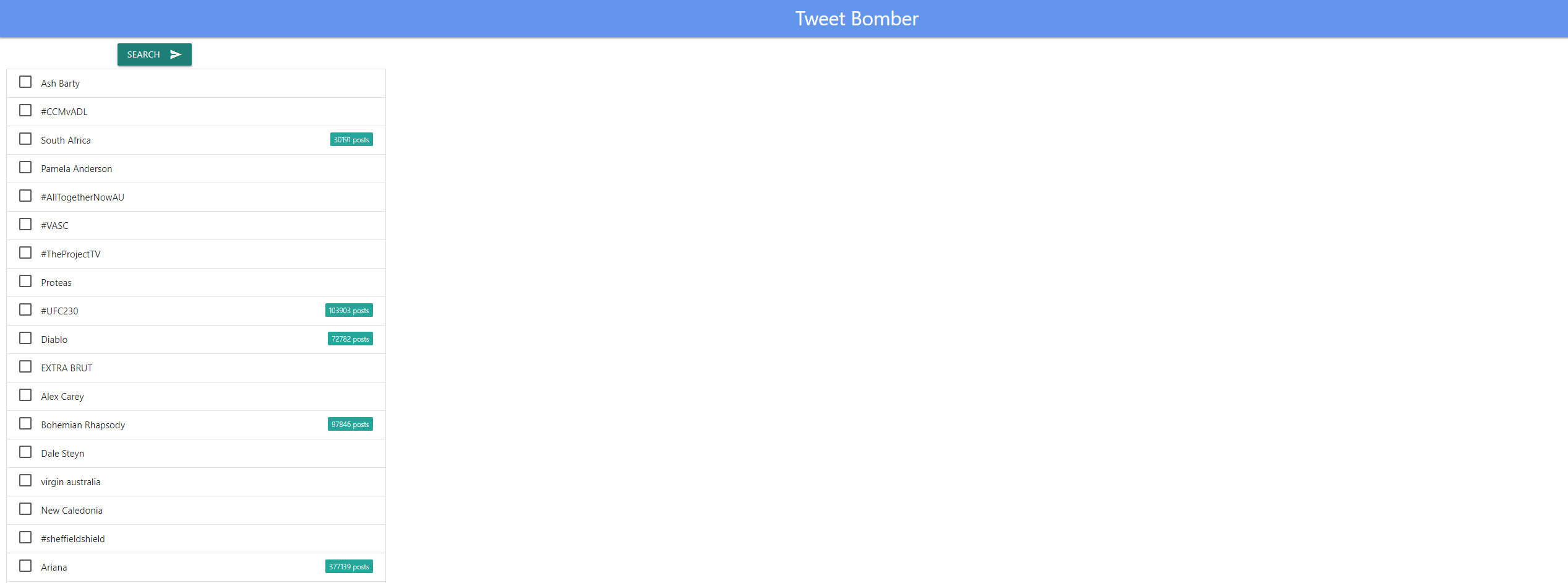
The application is limited is some regards. They are limited in more than one situation. These can include and are not limited to, the Twitter API too many requests and causing a 420-request error, the Twitter API pulling too many tweets and the API stopping from running, and finally the application can have connection errors that cause connections with the database and the application itself.

These limitations are caused by mainly third-party systems. However, they can be brought about due to user use and by poor implementation as the request may over consume the application’s resources.

# Extensions

The applications extensions are limitless. However, some key likelihoods of extension are the method in which the application displays the collected data. The current proposed method of a live google chart would demonstrate the intended purpose of sentiment. The key additions to this method would be live updating a table with the tweets made in the specifically selected trend. These extensions would improve the overall quality of the web application as the user would have more meaningful data presented to them.

# Appendix

The main page presents the functionality of the application as it presents the current trends for the application the get tweets from Twitter about. 

To implement on your own instances. These commands outline what is required to run. However, these require separate instances that work together.



