TwitMap Final Report

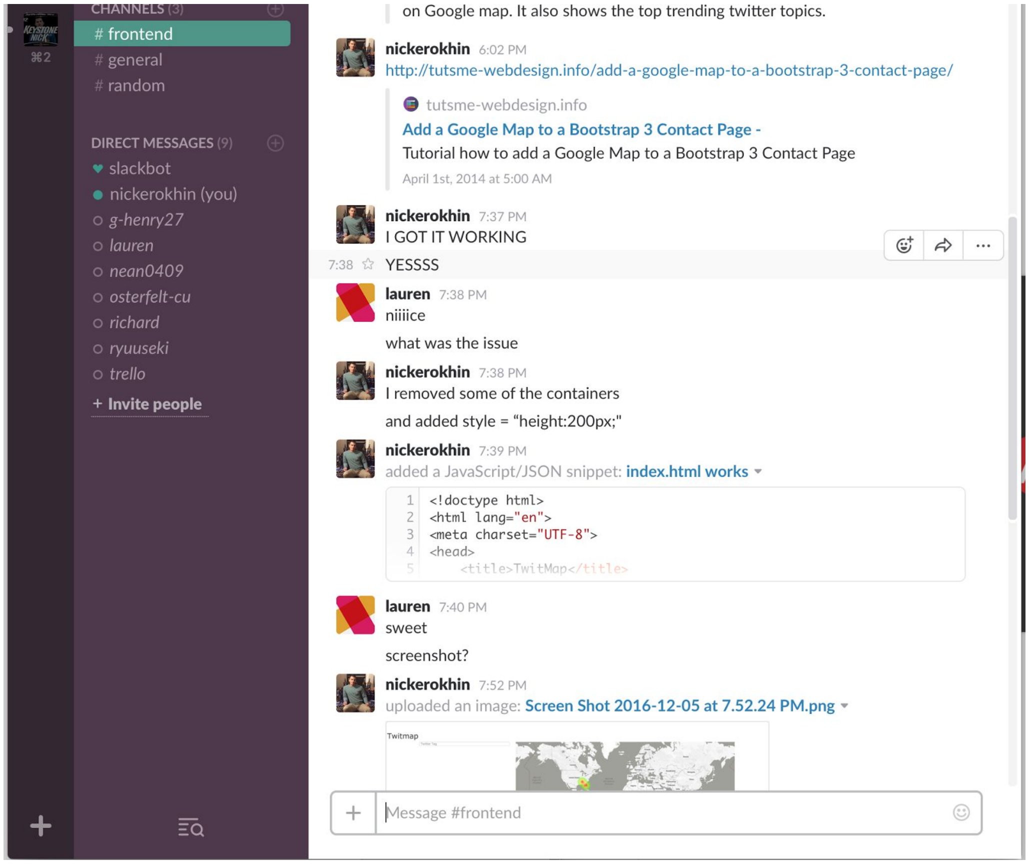
By Team 27: Nels Anderson, Richard Au, Nick Erokhin, Graham Henry, Nick Osterfelt, Lauren Raddatz
Website: https://arcane-castle-84668.herokuapp.com/
Github & Trello

Reflection

Productivity tools:

Slack

Slack was by far our most effective communication tool. We used to to organize meetings, plan for the future, and even “pair program” as shown above. Slack was particularly useful for us because of its ability to send code snippets to your group. We sent a lot of code through Slack, which is part of the reason we didn’t utilize Github to its fullest. Being able to make multiple channels in Slack also helped us stay on track for our respective roles.



on Google map. It also shows the top trending twitter topics.

nickerokhin 6:02 PM
http://tutsme-webdesign.info/add-a-google-map-to-a-bootstrap-3-contact-page/

tutsme-webdesign.info
Add a Google Map to a Bootstrap 3 Contact Page -
Tutorial how to add a Google Map to a Bootstrap 3 Contact Page
April 1st, 2014 at 5:00 AM

nickerokhin 7:37 PM
I GOT IT WORKING
7:38 YESSSS

lauren 7:38 PM
niiiice
what was the issue

nickerokhin 7:38 PM
I removed some of the containers
and added style = “height:200px;"

nickerokhin 7:39 PM
added a JavaScript/JSON snippet: index.html works

```
<!doctype html>
<html lang="en">
<meta charset="UTF-8">
<head>
    <title>TwitMap</title>
```

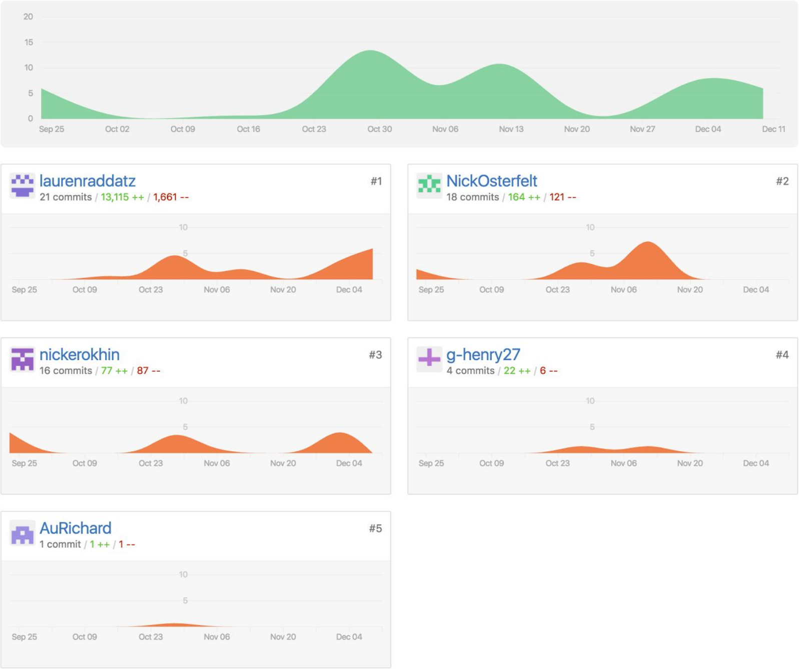
lauren 7:40 PM
sweet
screenshot?

nickerokhin 7:52 PM
uploaded an image: Screen Shot 2016-12-05 at 7.52.24 PM.png

Message #frontend

Github

Although at first we didn’t utilize Github to it’s fullest, we later realized how powerful of a tool it is. We at first did most of our work locally, only to push the entire directory later. This resulted in a lot of chaos with our versions until we learned how to properly manage pulls/pushes. But once we learned, our workflow drastically improved. We now know how powerful Github really is, and will certainly utilize more of it to manage our code in the future.



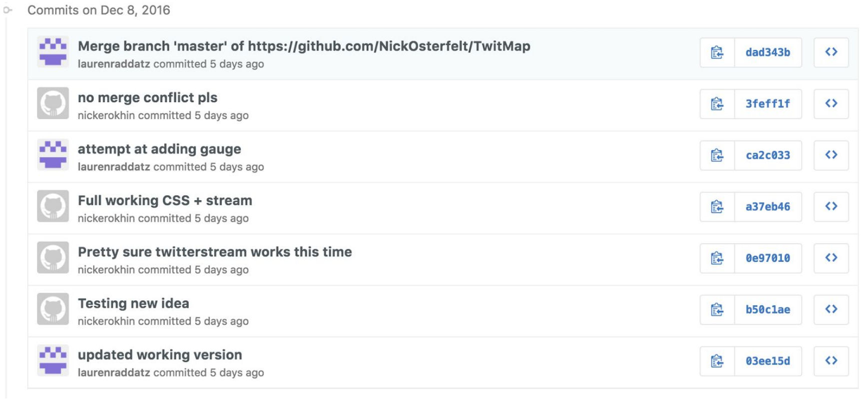
laurenraddatz #1
21 commits / 13,115 ++ / 1,661 --

NickOsterfelt #2
18 commits / 164 ++ / 121 --

nickerokhin #3
16 commits / 77 ++ / 87 --

g-henry27 #4
4 commits / 22 ++ / 6 --

AuRichard #5
1 commit / 1 ++ / 1 --



Commits on Dec 8, 2016

Merge branch 'master' of https://github.com/NickOsterfelt/TwitMap
laurenraddatz committed 5 days ago
dad343b

no merge conflict pls
nickerokhin committed 5 days ago
3feff1f

attempt at adding gauge
laurenraddatz committed 5 days ago
ca2c033

Full working CSS + stream
nickerokhin committed 5 days ago
a37eb46

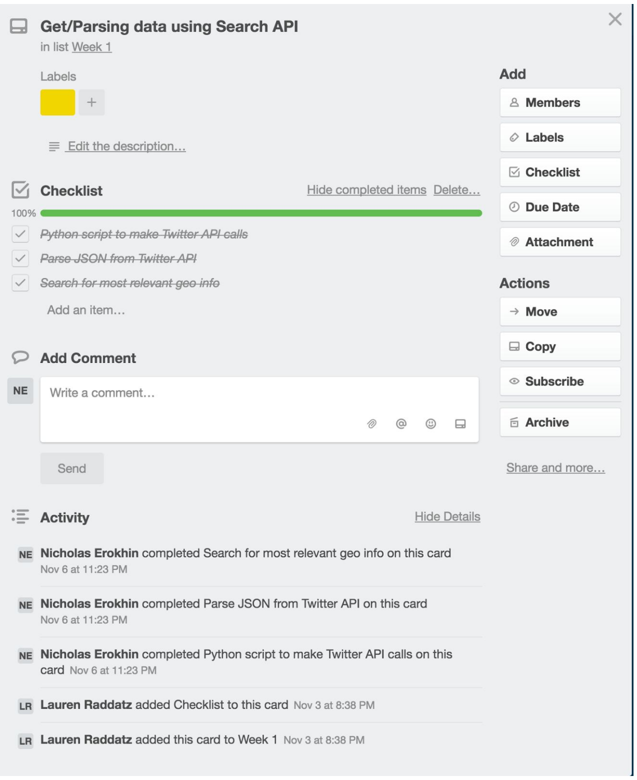
Pretty sure twitterstream works this time
nickerokhin committed 5 days ago
0e97010

Testing new idea
nickerokhin committed 5 days ago
b50c1ae

updated working version
laurenraddatz committed 5 days ago
03ee15d

Agile/Trello

Agile helped us stay on track and complete tasks in a reasonable amount of time. Trello/Agile were so effective for planning that the only time we lost momentum was due to not using either. To manage our sprints we used Trello, which provided us with an intuitive interface and task management system. Next time, we will account for all setbacks (midterms, breaks, etc.) in our calendars and plan around them. Below is an example of a 2 week sprint cycle, and the specifics within one of the tasks.



Get/Parsing data using Search API

in list Week 1

Labels

Checklist

100%

- ☑ ~~Python script to make Twitter API calls~~
- ☑ ~~Parse JSON from Twitter API~~
- ☑ ~~Search for most relevant geo info~~

Activity

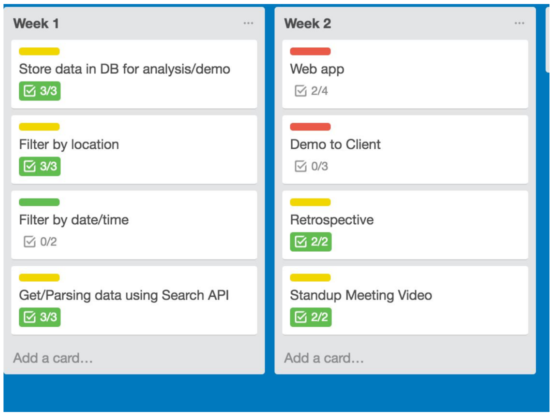
NE **Nicholas Erokhin** completed Search for most relevant geo info on this card Nov 6 at 11:23 PM

NE **Nicholas Erokhin** completed Parse JSON from Twitter API on this card Nov 6 at 11:23 PM

NE **Nicholas Erokhin** completed Python script to make Twitter API calls on this card Nov 6 at 11:23 PM

LR **Lauren Raddatz** added Checklist to this card Nov 3 at 8:38 PM

LR **Lauren Raddatz** added this card to Week 1 Nov 3 at 8:38 PM



Week 1

- Store data in DB for analysis/demo ☑ 3/3
- Filter by location ☑ 3/3
- Filter by date/time ☑ 0/2
- Get/Parsing data using Search API ☑ 3/3

Week 2

- Web app ☑ 2/4
- Demo to Client ☑ 0/3
- Retrospective ☑ 2/2
- Standup Meeting Video ☑ 2/2

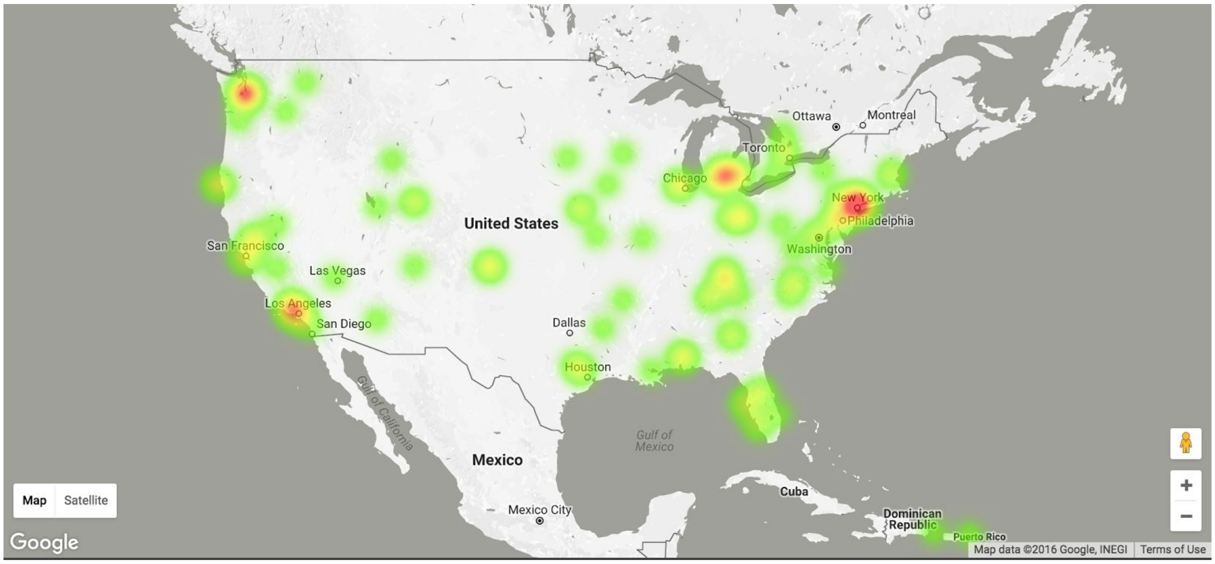
Front End tools:

HTML/Bootstrap/CSS/JavaScript/Photoshop

These were the standard front end/design tools that we used to complete our landing page. HTML/CSS are standard tools to build our page and make it look pretty. We used Bootstrap to neatly organize our page contents and make our website mobile friendly. The downside to Bootstrap was its very specific set of rules that prevented our map from working at first.

Google Maps Visualization API + Heatmap

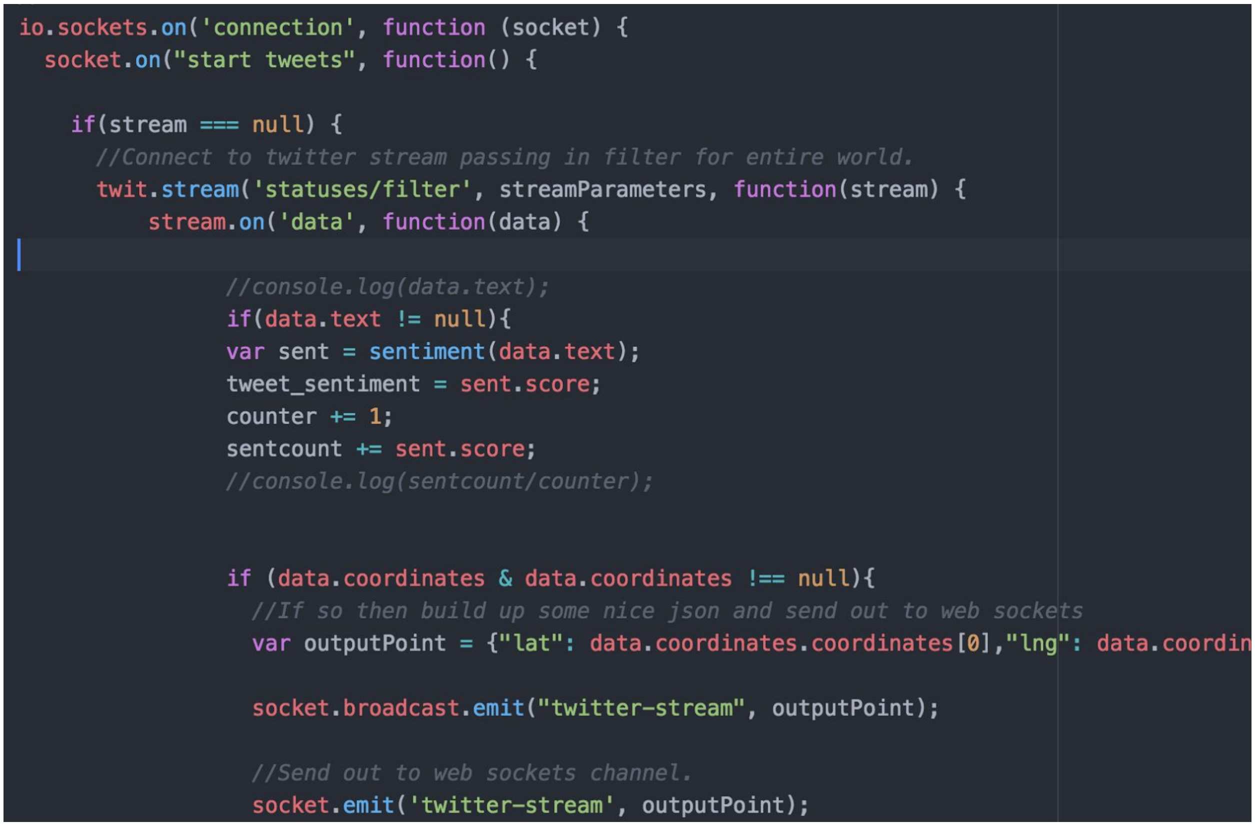
We chose this API to display our data because it is free. However, we had tremendous difficulty getting our Python to communicate with the map, which resulted in us switching to Node.js. Once we got past that, Google Maps was very easy to use.



Backend Tools:

Node.js + Socket.io + Express

After switching to Node, we had to learn quite a bit since none of us had really used it for a project before. This set us back a few weeks and certainly diminished the quality of our final product. However, Node proved immensely powerful with its asynchronicity and usability with Sockets. We used the Socket.io module to send data to the map from the stream, and we planned to use it to display sentiment data on a gauge, but we didn't have enough time. Below is an example of using sockets to send data to the channel "Twitter stream" which gets sent through a map through another function.



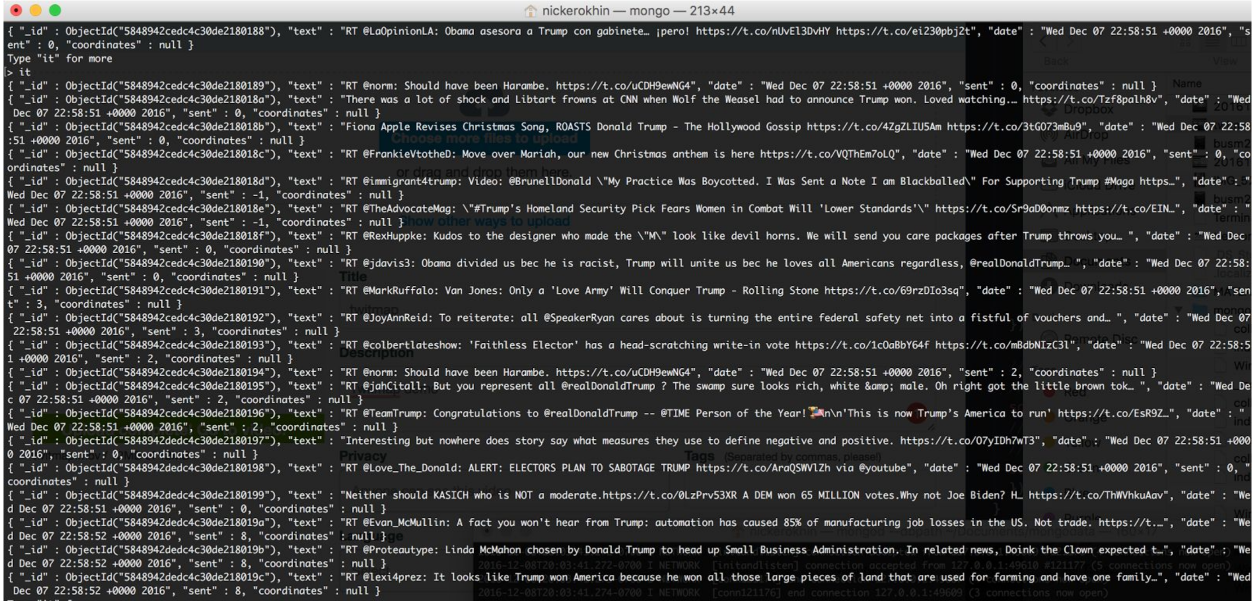
```
io.sockets.on('connection', function (socket) {
  socket.on("start tweets", function() {

    if(stream === null) {
      //Connect to twitter stream passing in filter for entire world.
      twit.stream('statuses/filter', streamParameters, function(stream) {
          stream.on('data', function(data) {

                //console.log(data.text);
                if(data.text != null){
                var sent = sentiment(data.text);
                tweet_sentiment = sent.score;
                counter += 1;
                sentcount += sent.score;
                //console.log(sentcount/counter);


                if (data.coordinates & data.coordinates !== null){
                  //If so then build up some nice json and send out to web sockets
                  var outputPoint = {"lat": data.coordinates.coordinates[0],"lng": data.coordin

                  socket.broadcast.emit("twitter-stream", outputPoint);

                  //Send out to web sockets channel.
                  socket.emit('twitter-stream', outputPoint);
```

Heroku

We first chose Heroku because it is free, and then learned that it was extremely easy to use, and worked perfectly with Git. We could push our current working repo directly to Heroku and our website worked right away. Unfortunately our product isn't scaled in the least, so more than 2 users crashes it.

MongoDB

We used MongoDB to store the relevant tweets captured by our stream. We primarily used MongoDB because it uses JSON which plays nice with Twitter and Node.js. However, it was far too forgiving. We were able to insert data using improper fields, something that mySQL would never allow. We stored the tweet with the ObjectID, Text, Date, Sentiment, and Coords.



```
{ "_id" : ObjectId("5848942cedc4c30de2180188"), "text" : "RT @LaOpinionLA: Obama asesora a Trump con gabinete… ¡pero! https://t.co/nUvEl3DvHY https://t.co/ei230pbj2t", "date" : "Wed Dec 07 22:58:51 +0000 2016", "s
ent" : 0, "coordinates" : null }
Type "it" for more
> it
{ "_id" : ObjectId("5848942cedc4c30de2180189"), "text" : "RT @norm: Should have been Harambe. https://t.co/uCDH9ewNG4", "date" : "Wed Dec 07 22:58:51 +0000 2016", "sent" : 0, "coordinates" : null }
{ "_id" : ObjectId("5848942cedc4c30de218018a"), "text" : "There was a lot of shock and Libtart frowns at CNN when Wolf the Weasel had to announce Trump won. Loved watching.… https://t.co/Tzf8palh8v", "date" : "Wed
 Dec 07 22:58:51 +0000 2016", "sent" : 0, "coordinates" : null }
{ "_id" : ObjectId("5848942cedc4c30de218018b"), "text" : "Fiona Apple Revises Christmas Song, ROASTS Donald Trump - The Hollywood Gossip https://t.co/4ZgZLIU5Am https://t.co/3tGO73mBu9", "date" : "Wed Dec 07 22:58
:51 +0000 2016", "sent" : 0, "coordinates" : null }
{ "_id" : ObjectId("5848942cedc4c30de218018c"), "text" : "RT @FrankieVtotheD: Move over Mariah, our new Christmas anthem is here https://t.co/VQThEm7oLQ", "date" : "Wed Dec 07 22:58:51 +0000 2016", "sent" : 0, "co
ordinates" : null }
{ "_id" : ObjectId("5848942cedc4c30de218018d"), "text" : "RT @immigrant4trump: Video: @BrunellDonald \"My Practice Was Boycotted. I Was Sent a Note I am Blackballed\" For Supporting Trump #Maga https…", "date" : "
Wed Dec 07 22:58:51 +0000 2016", "sent" : -1, "coordinates" : null }
{ "_id" : ObjectId("5848942cedc4c30de218018e"), "text" : "RT @TheAdvocateMag: \"#Trump's Homeland Security Pick Fears Women in Combat Will 'Lower Standards'\" https://t.co/Sr9aD0onmz https://t.co/EIN…", "date" : "
Wed Dec 07 22:58:51 +0000 2016", "sent" : -1, "coordinates" : null }
{ "_id" : ObjectId("5848942cedc4c30de218018f"), "text" : "RT @RexHuppke: Kudos to the designer who made the \"M\" look like devil horns. We will send you care packages after Trump throws you… ", "date" : "Wed Dec
07 22:58:51 +0000 2016", "sent" : 0, "coordinates" : null }
{ "_id" : ObjectId("5848942cedc4c30de2180190"), "text" : "RT @jdavis3: Obama divided us bec he is racist, Trump will unite us bec he loves all Americans regardless, @realDonaldTrump… ", "date" : "Wed Dec 07 22:58:
51 +0000 2016", "sent" : 0, "coordinates" : null }
{ "_id" : ObjectId("5848942cedc4c30de2180191"), "text" : "RT @MarkRuffalo: Van Jones: Only a 'Love Army' Will Conquer Trump - Rolling Stone https://t.co/69rzDIo3sq", "date" : "Wed Dec 07 22:58:51 +0000 2016", "sen
t" : 3, "coordinates" : null }
{ "_id" : ObjectId("5848942cedc4c30de2180192"), "text" : "RT @JoyAnnReid: To reiterate: all @SpeakerRyan cares about is turning the entire federal safety net into a fistful of vouchers and… ", "date" : "Wed Dec 07
 22:58:51 +0000 2016", "sent" : 3, "coordinates" : null }
{ "_id" : ObjectId("5848942cedc4c30de2180193"), "text" : "RT @colbertlateshow: 'Faithless Elector' has a head-scratching write-in vote https://t.co/1cOaBbY64f https://t.co/mBdbNIzC3l", "date" : "Wed Dec 07 22:58:5
1 +0000 2016", "sent" : 2, "coordinates" : null }
{ "_id" : ObjectId("5848942cedc4c30de2180194"), "text" : "RT @norm: Should have been Harambe. https://t.co/uCDH9ewNG4", "date" : "Wed Dec 07 22:58:51 +0000 2016", "sent" : 2, "coordinates" : null }
{ "_id" : ObjectId("5848942cedc4c30de2180195"), "text" : "RT @jahCitall: But you represent all @realDonaldTrump ? The swamp sure looks rich, white &amp; male. Oh right got the little brown tok… ", "date" : "Wed De
c 07 22:58:51 +0000 2016", "sent" : 2, "coordinates" : null }
{ "_id" : ObjectId("5848942cedc4c30de2180196"), "text" : "RT @TeamTrump: Congratulations to @realDonaldTrump -- @TIME Person of the Year! 🏆🇺🇸\n\n'This is now Trump's America to run' https://t.co/EsR9Z…", "date" : "
Wed Dec 07 22:58:51 +0000 2016", "sent" : 2, "coordinates" : null }
{ "_id" : ObjectId("5848942dedc4c30de2180197"), "text" : "Interesting but nowhere does story say what measures they use to define negative and positive. https://t.co/O7yIDh7wT3", "date" : "Wed Dec 07 22:58:51 +000
0 2016", "sent" : 0, "coordinates" : null }
{ "_id" : ObjectId("5848942dedc4c30de2180198"), "text" : "RT @Love_The_Donald: ALERT: ELECTORS PLAN TO SABOTAGE TRUMP https://t.co/AraQSWVlZh via @youtube", "date" : "Wed Dec 07 22:58:51 +0000 2016", "sent" : 0, "
coordinates" : null }
{ "_id" : ObjectId("5848942dedc4c30de2180199"), "text" : "Neither should KASICH who is NOT a moderate.https://t.co/0LzPrv53XR A DEM won 65 MILLION votes.Why not Joe Biden? H… https://t.co/ThWVhkuAav", "date" : "We
d Dec 07 22:58:51 +0000 2016", "sent" : 0, "coordinates" : null }
{ "_id" : ObjectId("5848942dedc4c30de218019a"), "text" : "RT @Evan_McMullin: A fact you won't hear from Trump: automation has caused 85% of manufacturing job losses in the US. Not trade. https://t.…", "date" : "We
d Dec 07 22:58:52 +0000 2016", "sent" : 8, "coordinates" : null }
{ "_id" : ObjectId("5848942dedc4c30de218019b"), "text" : "RT @Proteautype: Linda McMahon chosen by Donald Trump to head up Small Business Administration. In related news, Doink the Clown expected t…", "date" : "We
d Dec 07 22:58:52 +0000 2016", "sent" : 8, "coordinates" : null }
{ "_id" : ObjectId("5848942dedc4c30de218019c"), "text" : "RT @lexi4prez: It looks like Trump won America because he won all those large pieces of land that are used for farming and have one family…", "date" : "Wed
 Dec 07 22:58:52 +0000 2016", "sent" : 8, "coordinates" : null }
```

Report

Status

Overview

With the final view of TwitMap coming into sight, it is clear that not all of the goals initially set were accomplished; with that being said, TwitMap made large strides in terms of ingenuity, and the goals that did get accomplished are the foundations for an extremely powerful web application. At the core, TwitMap is a twitter analytics tool that collects tweets containing a search parameter, visually plots the locations of the collected tweets, and uses sentiment analysis to present the user with an average disposition toward the parameter. Considering that more than three-hundred-million people use Twitter, the potential of TwitMap can easily be seen with large scale marketing, and even possibly more obscure uses such as emergency response. Through implementing the foundation of collecting these searched tweets, there opens a door of unlimited potential for other forms of analysis on the massive amount of collected data.

Accomplishment of Goals

As unanimously decided in the initial phase of our project, the goals for TwitMap were set very high (based on the vision of the potential in our app). Although this led to some of the objectives not being met, this ambitious attitude was what ultimately drove our progress and made us strive to make our product designed for high scalability. A list of fundamental goals and non-essential goals (based on our original product vision) are listed below with their completion statuses:

Fundamental Goals:

Collecting and storing tweets (Completed)

The collecting and storing of tweets is the likely the most important component of our project considering every other component is reliant on the data collected from the twitter stream. This is accomplished using Node.js where a Twitter stream is initialized and tracking parameters are used to store relevant tweets in MongoDB.

Mapping geotagged tweets (Completed)

A core feature that is the basis of displaying the data in an interpretable manner. TwitMap uses the Google Maps JavaScript Visualization API where coordinates are sent through Socket.io and displayed on a heatmap that interprets the volume of tweets captured in each area.

Creation of user friendly Interface (In Progress)

An obvious component of any modern web application. The TwitMap website was created using the Bootstrap JavaScript framework, where the map is displayed and search boxes are present. However, there are many other ways that we had intended to represent the sentiment analysis data, and there are always changes that could be made to the website as more functionality of the app is added.

Non-essential Goals:

Sentiment Analysis **(Completed)**

The sentiment analysis component was designed with the intention of utilizing the Twitter data in a way that seemed meaningful. To accomplish this, TwitMap sends the text of tweets to the Node.js sentiment analysis api, which utilizes artificial intelligence to try to determine the disposition of the tweet toward its subject.

Prediction Algorithm **(In Progress)**

Our prediction algorithm unfortunately did not make it into the final release due to inability to decide on an effective means to form predictions. It is no surprise that forming predictions based on random expressed thoughts of millions of people is not a simple task, and would involve a huge amount statistical analysis to have any credibility. In the most basic form, a prediction algorithm was created that formed predictions based on the sentiment of comparative topics such as sports; however, this has not been added to the functional web app yet.

Creation of user profiles **(Removed)**

User profiles was a feature that survived as an ideally good component for the first two sprint cycles, but was eventually removed due to it being a large amount of work for relatively no functional gain. Also with our target market being businesses performing market analysis, user profiles seemed unnecessary.

Filter by time **(Removed)**

The idea of being able to see the spread of trends on a map over time is a very exciting idea. However when considering the implementation of this, we realised that the amount of data that would need to be held and collected for any reasonable time scale would be enormous in order to ensure that any user could search any parameter and see a timelapse of it. Unfortunately we did not have the capability to acquire or store this much data.

Overall Status:

Although TwitMap did not meet the high standards that were initially set, the fundamental components that give TwitMap its power came together, leaving unlimited room for improvement and potential uses. As can be seen above, with all of the fundamental goals being completed, the next step to further the development of TwitMap would be to add extra features and means to analyze the collected data.

Accomplishments

The most outstanding accomplishment during the development process is obviously the final result: a functioning TwitMap. Several college students with little to no professional development experience were able to conceive an idea from scratch and bring it to life. This required working with tools, frameworks and databases that none of us had any prior experience with. One of the challenges overcome during development was how to effectively identify the geolocation of many of the tweets. Many people, instead of providing their exact coordinates, provide a real-life location (which can range in size from a restaurant to a city). We had to choose a consistent way of representing such tweets on the heatmap, eventually choosing to translate such locations' coordinates into the lower left corner of its bounding box. Near the end of development, another large challenge was establishing TwitMap on a webhost, another significant accomplishment that none of us had attempted previously.

Aside from technical accomplishments, all of us felt like we communicated effectively as a team. Slack was constantly being utilized, to the point where no one ever felt out of the loop on where the project stood and what was needed from everyone. Not only did this reduce individual frustration, but it assisted in developing bonds between members much faster, as communication breeds familiarity. Moving forward into the workplace, this is an effective skill to possess; NACE ranks communication as the number one skill employees seek in college graduates.

Outstanding Issues

As mentioned before, the most significant issue encountered during TwitMap's development was the switching of frameworks halfway through. As a team, we had made significant progress in our Python code, but too late realized it did not interact with the other tools we needed to use to bring TwitMap to life. Thus, we had to switch from Python to Node, which we as a team had less experience in using. This cost us valuable time and led to us trying to frantically trying to make up for lost time.

Another issue that *still* remains is that sentiment analysis was not fully implemented to the extent initially imagined. Upon running the algorithm we chose to implement, the returned results are underwhelming at best, providing almost no definitive results. One possible explanation is that the algorithm is meant for more extensive pieces of text; 140 characters is not a large sample size, and may not provide the algorithm with enough information for it to definitively identify the underlying sentiment.

Finally, something that was not an issue for us per se, but could become an issue down the road, is the size of our storage (5mb). For a business trying to run analysis on a large-scale marketing campaign, this obviously will not hold enough data for them to make any significant decisions.

Future Plans

Moving forward, the most looming issue in TwitMap is the mediocre performance of the sentiment analysis. To truly accomplish the vision we put forth for ourselves, it is important for this to be functioning at as high a level as the rest of the project. Writing a high performance algorithm that can extract sentiment from text effectively is somewhat beyond the scope of our abilities; instead, we need to do more in-depth research on the options available to us before settling on one. Once one is chosen, we can then move forward with the goal of visualizing sentiment. As it stands now, TwitMap can only plot the real-time occurrences of a provided search term on a density map. A second option will be implemented, where positive and negative sentiment densities are plotted side by side in opposing colors. This requires tweaks to the UI implementation; as it stands now, the programmer has no control over the visualization color, nor the ability to plot different two distinct entities on the same map. This will require serious revisions to the code functionality.

Finally, in order to meet the needs of actual organizations, TwitMap will need to be revamped. Because this was originally written by broke college students, minimalist storage databases and Twitter API keys were utilized (because they were free). However, during testing on a single laptop, both were observed to fail after any significant amount of time. For the needs of a large organization, this will not do, and TwitMap’s database and access to Twitter will need to be expanded (likely requiring cold hard cash).