

MDDN201 JOURNAL ENTRIES
EMILY ALICE SHAW
“People Zoo”
300378882

Choices - API group 25/9/2017

because I most likely will be using two API's (geolocation and twitterBots), I feel like the API and experimental group is more up my alley than the other two.

My concept for my design, is a creepy bot called People Zoo. People zoo does two main things.

1. The user can tweet to the bot, with their message and the celebrity they want the message to go to. Twitter bot them spams the celebrity with the message and the user, so that the chances of the celebrity seeing the tweet is more. I would like to do something like this, or something working around spam rules on twitter.
2. The second thing, is a geolocation API, where the last social media location of the celebrity is plotted on a visual map. You can search for a certain celebrity and find their last known location.

I do have some others ideas, Which I could develop if there is something 'experimental' which is relevant. My options are open, but I have an interest in API's and the technical side of things, so I would like to be put in this group.

SPA#3 19/10/2017

My project is called People Zoo. The meaning of this name is supposed to be a reflection of how in today social media, it is commonplace to share your entire life with the world. Social media in this sense, is a people zoo. This concept also extends to your everyday idol - The kim Kardashians of the world often put themselves out there on social media, leading to a cult following of crazy fans and people who follow their every move. My project, 'People Zoo' takes advantage of this, in the creepiest way possible, for your everyday joe to stalk, and show their appreciation of this social media phenomenon.

In my website, I make use of three API's, I used the Instagram API to pull the location of the latest social media check in of the celebrity. I did this based on the tag function, coupled with relevant hashtags, i.e. #starspotting. I then plotted the geotagged location of these, importing it into the google maps API. My second function is based on 'celebrity interactions'. I implemented a twitterbot, which when tweeted at with the correct hashtag and @celeb, will come up with a random poem to tweet at the celebrity, which is 'from' the original tweeter. I will have these two main functions as the core of my website.

I chose to work in the API group because I am very interested in the use of API's, hence why I am touching on three in this project. Being in this group I have learned the most in this course, especially becoming more familiar with not only API's but also being able to execute them through code, and terminal. I would say that this class has strengthened my coding abilities a lot, especially in javascript. I have also become familiar with node.js, and JSON.

Twitter bot deployment and API restrictions 19/10/2017

I had some issues with my twitter bot, I hit the Spam warning a couple of times when my bot went rogue. I accidentally did this twice.

1. tweeted swear words to Kylie Jenner, as my 'test' word was naughty (I have learned my lesson)
2. hit the maximum amount of @mentions in an hour

both of these were when I was on my lunch break and wasn't monitoring my bot as I still had it running in terminal. The first time I was banned for 48 hours, the second time I was suspended from 'writing activities' so I had to email twitter. A little funny, but definitely drew out the development process into a couple of days.

This was the email I got from them: At one point, when I was fiddling with the Instagram API, I also hit a point where I could not pull anything from the JSON file, this was because I had hit the maximum of 30 pulls per hour. I have learned now that these sort of restrictions I should research before I start playing with the API, as it is something that is completely avoidable.

Thanks for reaching out. Your app was restricted for sending a large volume of unsolicited and/or repeat @mentions. This is prohibited by the [Twitter Rules](#):

Spam: You may not use the Twitter service for the purpose of spamming anyone. What constitutes "spamming" will evolve as we respond to new tricks and tactics by spammers. Some of the factors that we take into account when determining what conduct is considered to be spamming are:

- if you send large numbers of duplicate replies or mentions;
- if you send large numbers of unsolicited replies or mentions;

Note that this behavior is not allowed regardless of whether or not it is automated. We have reactivated your app, but this behavior must stop. As an alternative to @mentions you may use hashtags.

Regards,

Twitter Platform Operations

Getting endpoints from the Instagram API 19/10/2017

The only thing I needed from the Instagram API was to grab locations, to do this I thought that maybe I would need to use the locations endpoints. This seemed a little more complex to implement, and did not seem particularly relevant to what exactly I wanted once I looked into it. The locations endpoints was more about searching via locations, i.e. 'recent images around me'.

It was after reading this that I formed the idea that I wanted to grab using hash tags, and relevant geolocation data from images that were attached to the hashtags. I was able to achieve this using just the basic and easiest hashtag endpoint. This grabs all the public information from the image it is attached to, including locations (long and lat), as well as tags.

Ok, so I had some success. With this, it was easy enough to get, the hardest part of this was setting up the account and getting the access token (as this was harder to find for some reason, and I had to jump through some hoops to get there.) The next hardest part was figuring out what exactly I had to do, as ya girl has never used anything API related before.

Scouring the Internet I found some relevant sources which helped me, this one here:

<http://dmolsen.com/2013/04/05/generating-access-tokens-for-instagram/>

was helpful for finding an access token. I stumbled through the rest of it to be honest.

I eventually was able to get my json link to work, here is the link for it:

<https://api.instagram.com/v1/tags/starspotting/media/recent?>

access_token=6217406642.348eb7c.bc9a06b20fae4fc1b2699141fc64f926&callback=callbackFunction

Ryan gave me advice to install a json viewer, so I could comprehend the output Instagram gave me better, which gave it clear indentations, where I was able to see the arrays better.

Putting location tags into google maps API

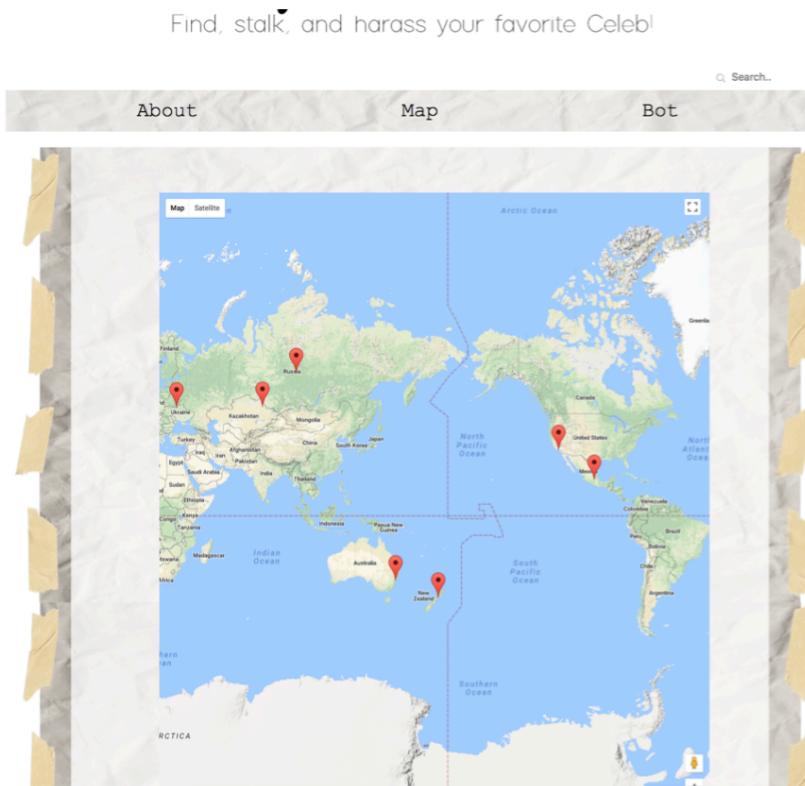
20/10/2017

The API I was currently using from project two, was just for show and it was only one point on the map. There were a couple of other options that I could use for multiple points, like marker clustering. I chose to do marker clustering which would enable me to have multiple points on the map, and when zoomed out there would be 'clusters' with a number, (although only when there were multiple people in one section it would do this).

Using the basic example provided on the developer website, I was able to cycle through the JSON file, to get to longitude and latitude sections of the array. I drew these out successfully and then plotted them all on the map, relatively free of troubles. The hard part, which I could not get myself, however was being able to cycle through to find the tags in the pictures. This is because the tags were located inside an array, which was INSIDE another array. I was not sure how I would get to these, as the coding experience I had mostly had was only Java, and I do not think that is possible in Java (or I have never encountered it, either way). I tried using a nested for loop, but I must have been doing it wrong, as it didn't work well.

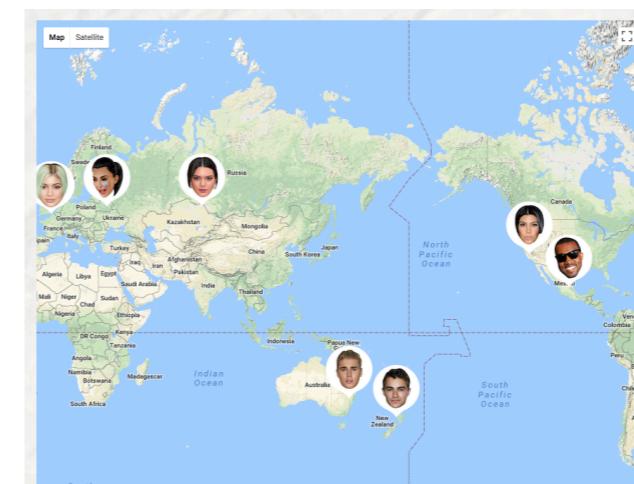
When I came to my tutorial, I got it working with help from the tutors, I also was able to separate my .js from the html file, as to simplify my code.

After this was done, this is what it looked like:



I was happy with this, although I didn't like the look of the markers, and how they didn't give off any information of who the person was. I chose to make customer markers, which is possible through the Google maps API. This is what it looked like once I was done.

I was very happy with this - If I have enough time after implementing my twitterbot fully, I will probably include infographs when you hover over the person and get relevant information, like, name, exact location, ect.



Twitter bot development in node.js 19/10/2017

I wasn't sure whether I wanted to work on a twitterbot or not, as I felt that The map function was probably enough. However, when I completed my map API in good time, I decided why not give it a go. I watched the tutorial videos from coding train (recommended by tutors) here is the first video in the series: https://www.youtube.com/watch?v=RF5_MPSNAtU

I had no experience with node.js to start off with, but an afternoon of binge watching this series left me in a really good/confident place to begin creating my bot. It wasn't too challengeing, if I hadn't been blocked by twitter so many times I probably would have been able to get it all done that night.

I ended up download the 'twit' package:

<https://www.npmjs.com/package/twit>

this was the base of my code for a while, while I was still going off videos from the coding train. I ended up needing different things so I downloaded some more packages using home brew.

A little bit about my bot:

my bot searches for tweets that use #peoplezoo, if you tweet using the hashtag, plus the name of the celebrity you want to 'contact', the bot will pick this up and quote your tweet, it tweets a string of random poems, based of an array holding strings of words. it tacks on at the end: -@yourtwitternamehere.

The idea of this is that the user can 'tweet love' to their favorite celebrity, the bot comes up with the love poem, and addresses the celebrity, saying it is from the origional user.

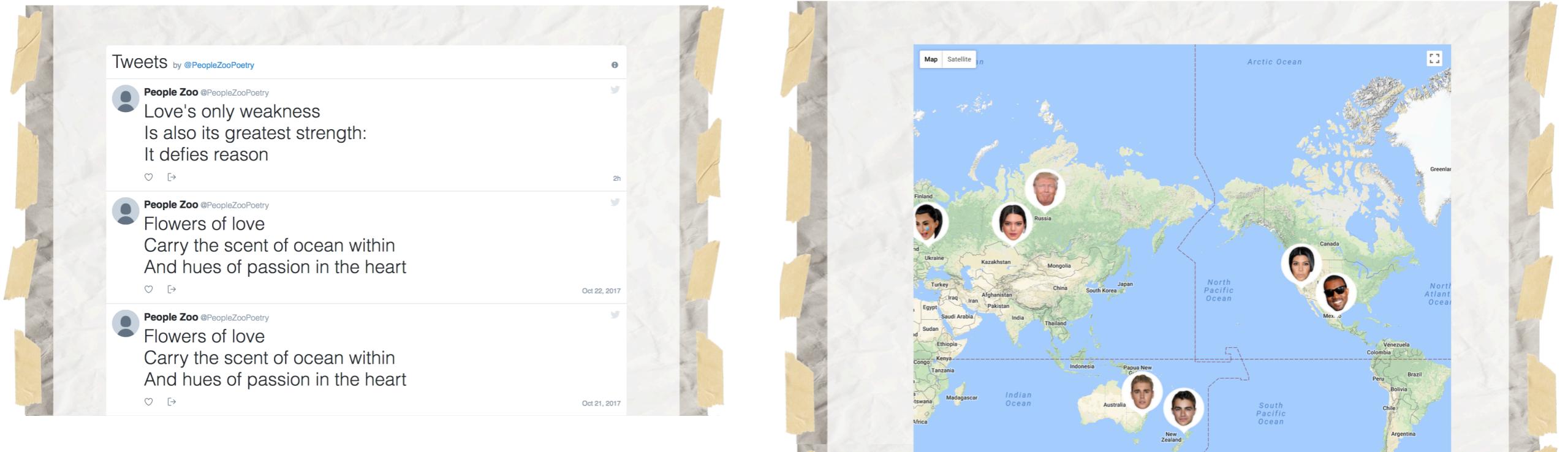
Twitter bot development to heroku 19/10/2017

My twitterbot works well, however I would like it to be live all the time, so I do not have to have terminal open. I followed the tutorial done by coding train on 'twitter deployment to heroku' so that my bot would be live all the time. this seems to have worked, although it takes some time for the bot to pick up the tweet. this is fine though, I have no problem with this.

rules I added to my bot:

there was an error which kept coming up, where the characters were over 140 (this is a common twitter rule). I ended up making a rule where, if the string was over 54 characters (as the maximum my poems would be is 86), the bot will spit out the poem without the @persons name, it then tweets the person back saying: "i'm sorry, your tweet was too big for me :(it was x characters too long.

SPA #4 23/10/2017



Find, stalk, and harass your favorite Celeb!

Search..



Final CSS Revisions 29/10/2017

So, with my maps done, and my twitterbot done, there was only a few things left to do for my website - which was CSS.
problems encountered:

- I had some issues with getting my font to work. I was not happy with web safe fonts, as my website did not look as good as I envisioned in courier new. I followed a tutorial w3schools about @font-face, and through this I was able to get the two royalty-free fonts I originally wanted to use.
- The second problem I faced, was that one of the fonts I wanted to use was not compatible with github. SO frustrating, as it worked with surge. I ended up changing my font - as this frustrated me greatly.

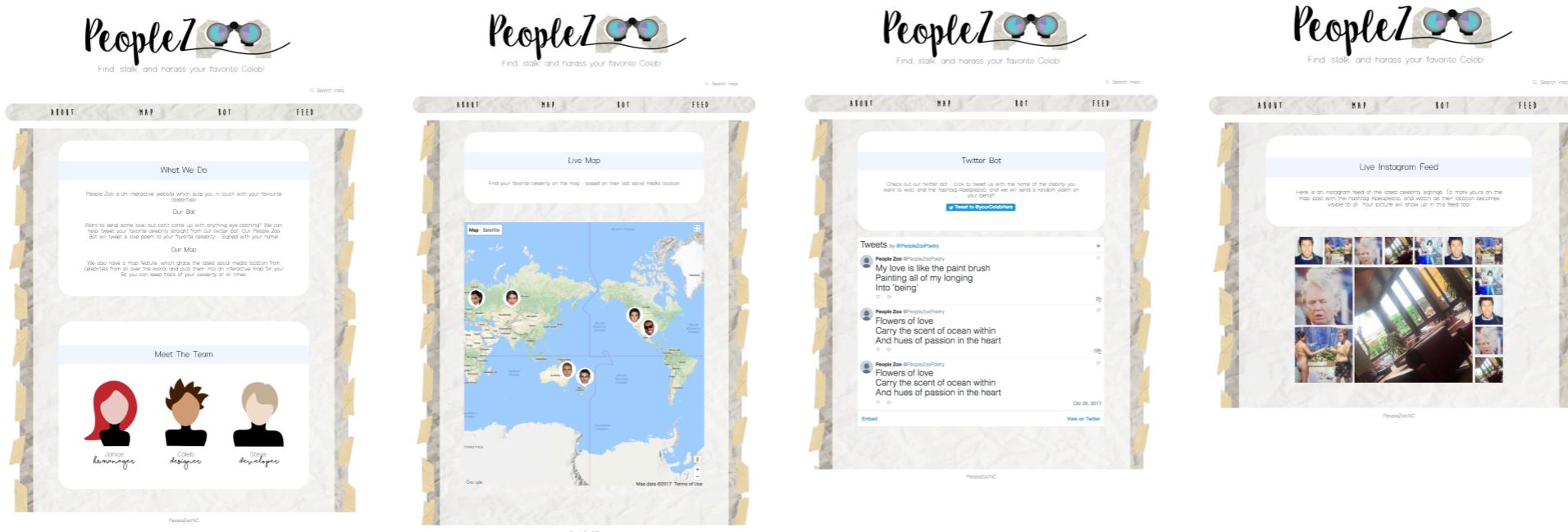
about my fonts: I downloaded the free versions from dafont.com the licenses are royalty, however if I was going to publish my website out of beta, I would purchase the full versions for commercial use.

other than these two things, I did not struggle with much else, that I could not fix.

things I achieved:

- general clean up of my code
- white boxes around text area
- revision on general text in my website
- instagram widget working perfectly, I used instash to do this, as it was complex to code myself at this time.

Finished product 29/10/2017



links used throughout this project 29/10/2017

node.js tutorial: https://www.youtube.com/watch?v=RF5_MPSNAtU

font-face: https://www.w3schools.com/cssref/css3_pr_font-face_rule.asp

comprehensive google maps API documentation: <https://developers.google.com/maps/>

comprehensive instagram sandbox developer documentation: <https://www.instagram.com/developer/sandbox/>

twitter api: <https://developer.twitter.com/en/docs>

instush: <http://www.instush.com>

fonts: www.dafont.com

for other css and html work, I used references from: www.w3schools.com