

ruffLife- Michelle Tang, Bo Hui Lu, Aaron Li, Kaitlin Wan  
SoftDev6  
Project 1 - ArRESTed Development  
2018-11-21

#### Functionality:

Using RESTful APIs to help stressful users relax mentally by looking at various memes, aesthetically pleasing animals(just dogs and cats), or funny jokes. The user will laugh, smile, and appreciate the wonder of animals and our beautiful website. This will be a one-page functioning website (using anchors instead of pages). The user will log in and be able to like a picture/quote, which will save them in a database. They will then have access to their liked posts. Users will not have to log in to view the access the page but will need to log in for full functionality (viewing their own personal images/quotes). Thus, there will be an incentive for viewers to create an account.

#### APIs we plan to use:

Cat fact: <https://catfact.ninja/> (no quota)  
Dog API: <https://dog.ceo/dog-api/> (no quota)  
RandomCat (no quota)  
Geek Jokes: <https://github.com/sameerkumar18/geek-joke-api>  
Quote of the day (FavQs) (db)  
OPENWeather (db)

#### Components:

PM: Michelle

Front End: (Kaitlin & Aaron)

- Create a logo

- Generate meme images from API and processing them through an HTML page

  - Jinja, html, bootstrap

  - Grant user the option to 'like' a photo which will be saved in their account to allow for later viewing.

- Generate animal (ie cat/dog) images from API and process them through an HTML page

  - Jinja, html, bootstrap

  - Grant user the option to 'like' a photo which will be saved in their account to allow for later viewing.

  - For logged in users: they can see previously liked dog/cat pictures(incentive to make an account)

- Generate jokes from API and processing them through an HTML page

- Generate a quote from the API

  - Grant user the option to see other quotes from the author (and allows the user to limit the number of entries they are receiving)

- Generate a call from OPENWeather using a logged in user's entered zip code(or city)

  - Logged in users and save locations and click on links that show each location's weather.

- HTML/CSS Templates

  - Jinja, html, bootstrap

Users with accounts will have a complete, single page with all the sections mentioned above while users without accounts will have links to individual pages/sections and they can't look at previous liked dogs/cat pics.

## Back End: (Bo & Aaron)

- Flask route functions to handle the APIs, and rendering each page as appropriate
- Working with the database to generate random entries (returned as a dictionary)
- Working with the database to store in 'favorite' images
- Create sessions for users with accounts

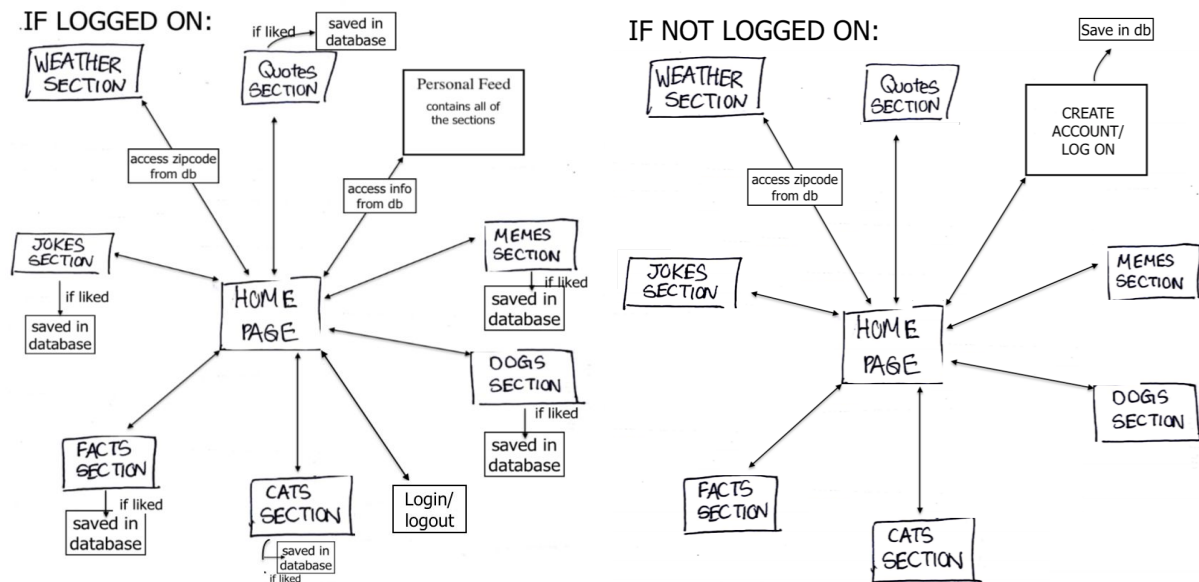
## Component map:

If not logged on:

Users will have viewing access to all of the pages: weather, quotes, jokes, facts, animals, and memes. Since they are not logged on, the weather page will reflect that of a randomized zipcode.

If logged on:

They will be given all of the abilities that those who are not logged on have. Users can like the content in specific sections. They will then be able to view all of it in a personal feed that will summarize all of their liked content. Their weather page will be generated by the specific zip code they entered when they initially created their account.



Database Schema: to store a local copy of all API data in order to minimize API calls!!!

- Facilitate logging in/out of users, as well as storing passwords(sessions) using dbs
- Certain aspects of the website will incorporate user data, such as types of memes they like/search for, jokes, etc (personalized website)
- With each username, there will be associated image links that correspond to that user

| USER     |         | API_LIST |         | QUOTES |         | WEATHER   |         |
|----------|---------|----------|---------|--------|---------|-----------|---------|
| USERNAME | TXT(PK) | ID       | INT(PK) | ID     | INT(PK) | DAY       | TXT(PK) |
| PASSWORD | TXT     | NAME     | TXT     | AUTHOR | TXT     | TEMP      | TXT     |
| L-ANIMAL | TXT     | URL      | TXT     | BODY   | TXT     | CONDITION | TXT     |
| L-MEME   | TXT     |          |         |        |         |           |         |

Site Map:

