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

Tentative Website Title: Paws Thy Stress

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).

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
- IcanhazdadJoke
- Quote of the day (FavQs) (db)
- OPENWeather (db)

Components:
Front End:

- Create a logo
- Generate meme images from API and processing them through an HTML page (getting the url)
    - 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 (getting the url)
    - Jinja, html, bootstrap
    - Grant user the option to ‘like’ a photo which will be saved in their account to allow for later viewing.
- 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)
- HTML/CSS Templates

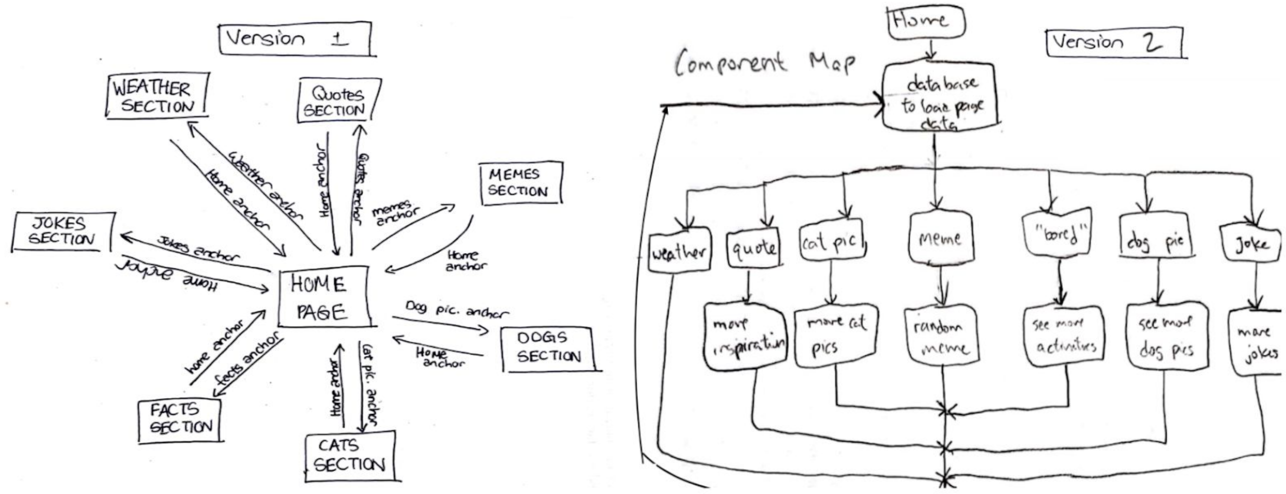
- ○ Jinja, html, bootstrap

Back End:

- Flask Route Functions to handle our APIs.
- Working with the database to generate random entries (returned as a dictionary)
- Working with the database to store in ‘favorite’ images

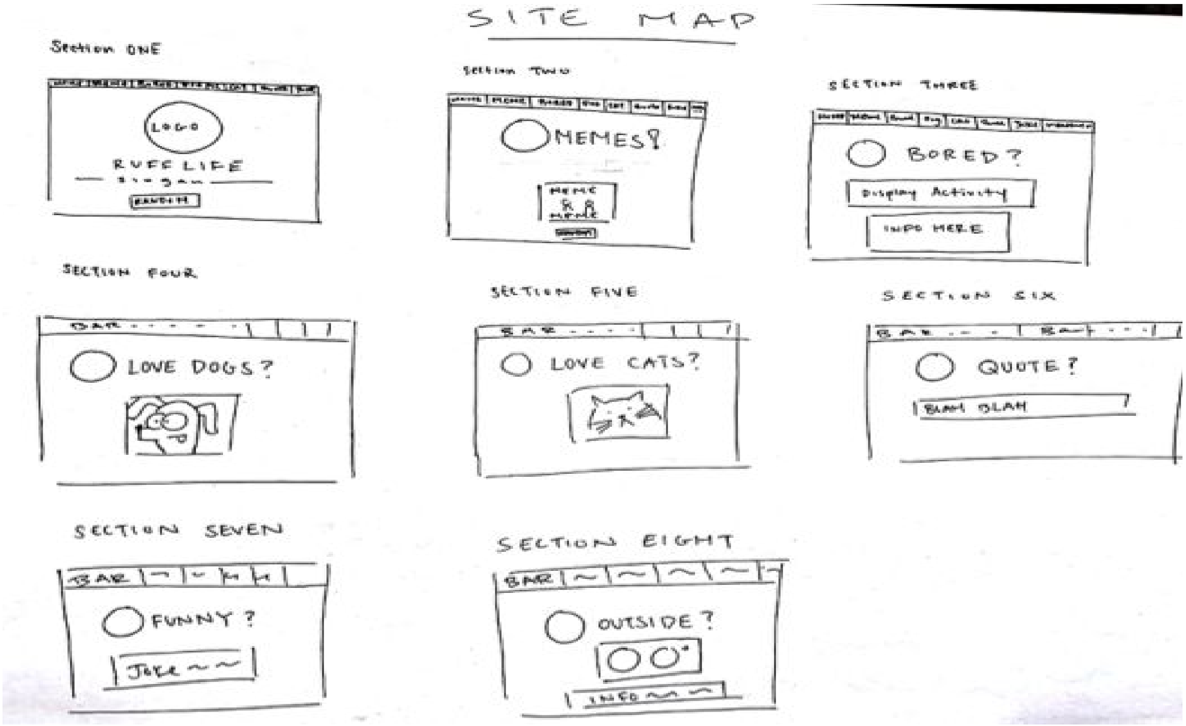
Back End: Bo
Front End: Kaitlin
Front/Back End: Aaron
Project Manager: Michelle

Component map:



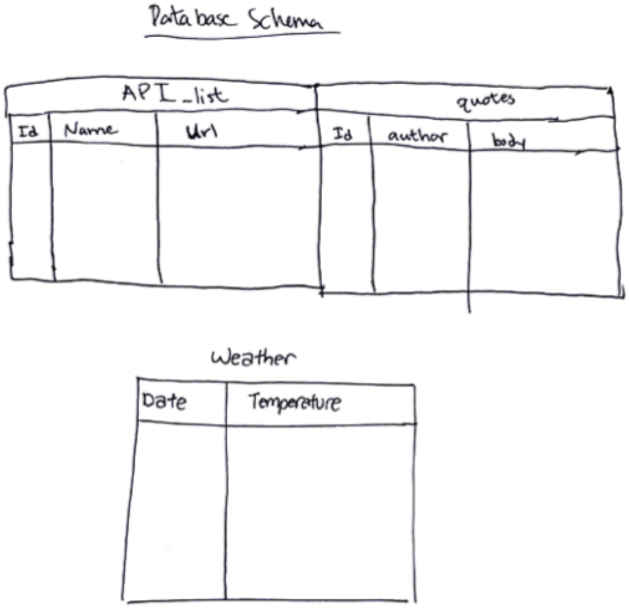
Initially, we have one webpage with multiple anchors that will lead to different sections. If we have time to expand, each anchor will correspond with their own webpage which will users to only get information related to one section.

Site Map:



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, but this will probably be an additional feature with dbs)·····



Database Schema

API_list			quotes		
Id	Name	Url	Id	author	body

Weather

Date	Temperature