

ruffLife- Michelle Tang, Bo Hui Lu, Aaron Li, Kaitlin Wan
SoftDev6
Project 1 - ArRESTed Development
2018-12-7

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). Logged in users also get a prettier page. 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 Fact: <http://kinduff.com/dog-api/> (no quota)

Dog API: <https://dog.ceo/dog-api/> (no quota)

RandomCat: <https://aws.random.cat/> (no quota)

Fun Jokes: <http://randomuselessfact.appspot.com/> (no quota)

GIPHY API: <https://developers.giphy.com/docs/> (NEEDS API KEY, 10000/day quota)

Quote of the day (FavQs):<https://favqs.com/api/qotd> (no quota)

Components:

PM: Michelle

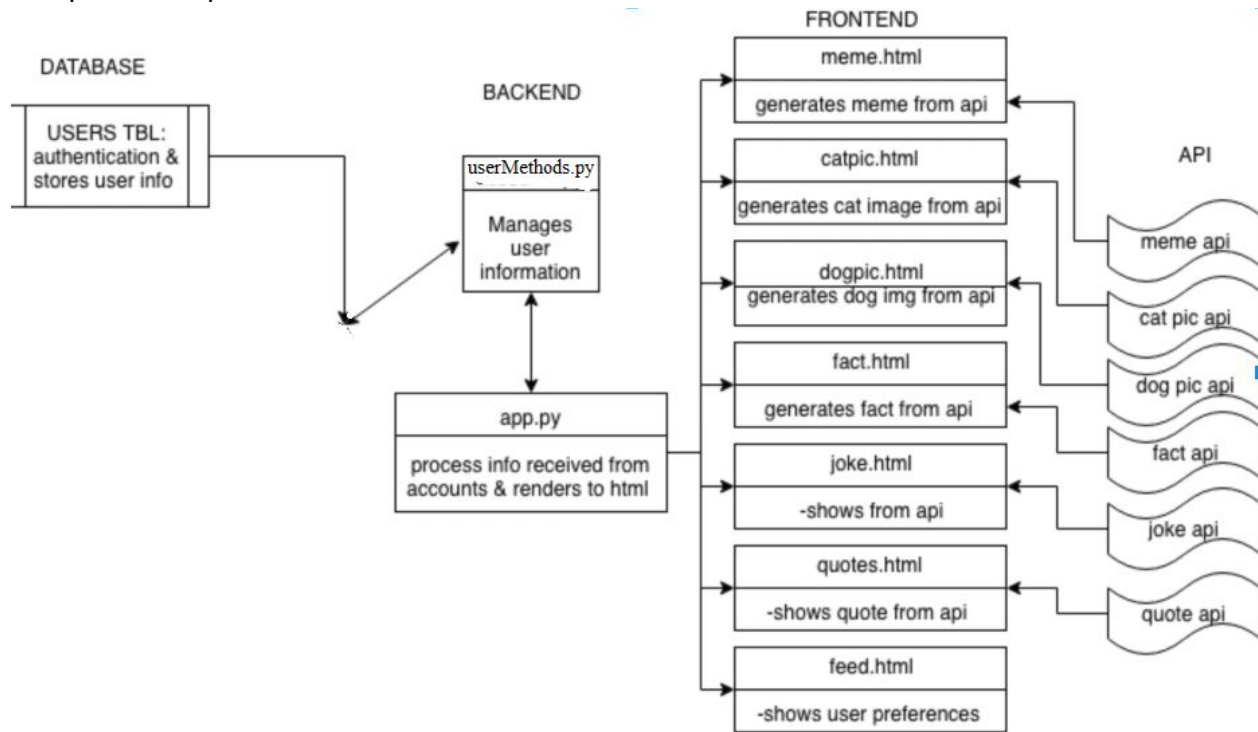
Front End: (Kaitlin & Aaron)

- 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)
- 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, which will be displayed in a personal feed
- Create sessions for users with accounts

Component Map:



Site map:

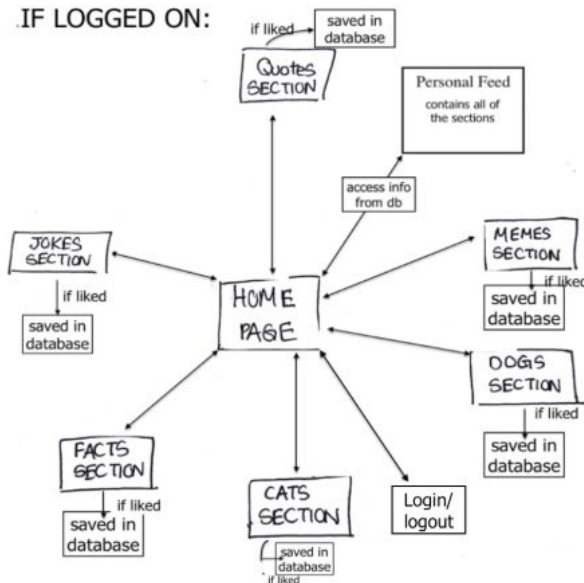
If not logged on:

Users will have viewing access to all of the pages: quotes, jokes, facts, animals, and memes.

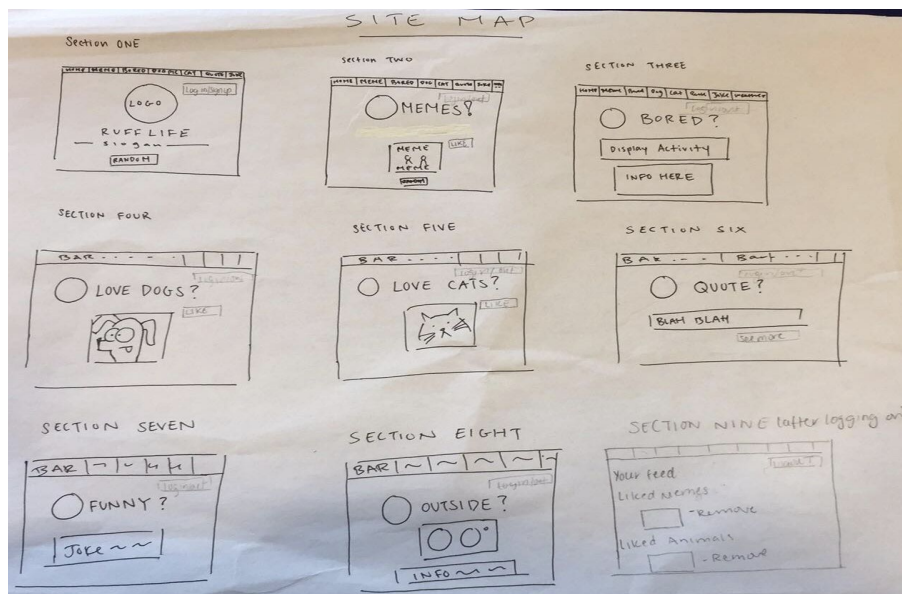
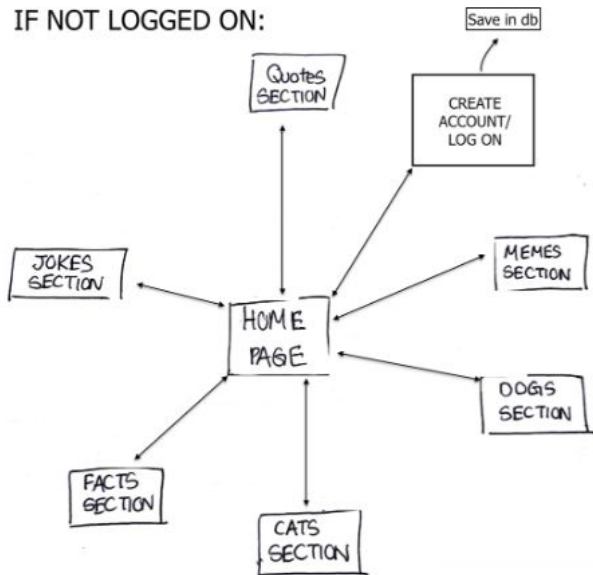
If logged on:

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.

IF LOGGED ON:



IF NOT LOGGED ON:



Database Schema:

- 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, animals pics, and quotes (personalized website)
- With each username, there will be associated image links that correspond to that user

USERINFO

username	TEXT
pass	TEXT
liked_image	TEXT
liked_words	TEXT