

MileStone 4

Revised List of Features

-Twitter Features

-Display recent tweets

-[FUNCTIONAL] Column of recent tweets on the search term will be displayed to the user, stylized in the same manner as twitter.

-[NON-FUNCTIONAL] Tweets will be embedded in cards. Tweets obtained through the Twitter Search API.

-Tweet Sentiment Score

-[FUNCTIONAL] Display graph with the distribution of positive, negative, and neutral tweets.

-[NON-FUNCTIONAL] Systems used: Twitter Search API, TextBlob, Pandas, Matplotlib.

-Popular locations

-[FUNCTIONAL] List the locations where the searched term has been tweeted about most in the last seven days.

-[NON-FUNCTIONAL] Systems used, Twitter Search API, Pandas, Matplotlib. Will log geotags for tweets and the calculate most frequent locations.

- Instagram Features

-Display Recent Posts

-[FUNCTIONAL] Column of popular recent posts containing the searched term.

-[NON-FUNCTIONAL] This will be achieved with the Instagram API. Posts will be formatted as they appear on the Instagram platform. Posts embedded in cards.

- Reddit Features

-Display Recent Posts

- [FUNCTIONAL] Column of popular recent posts containing the searched term.

- [NON-FUNCTIONAL] This will be achieved with the Reddit API. Posts will be formatted as they appear on the Reddit platform. Posts will be embedded in cards, placed into the Reddit column.

- Post Popularity Meter

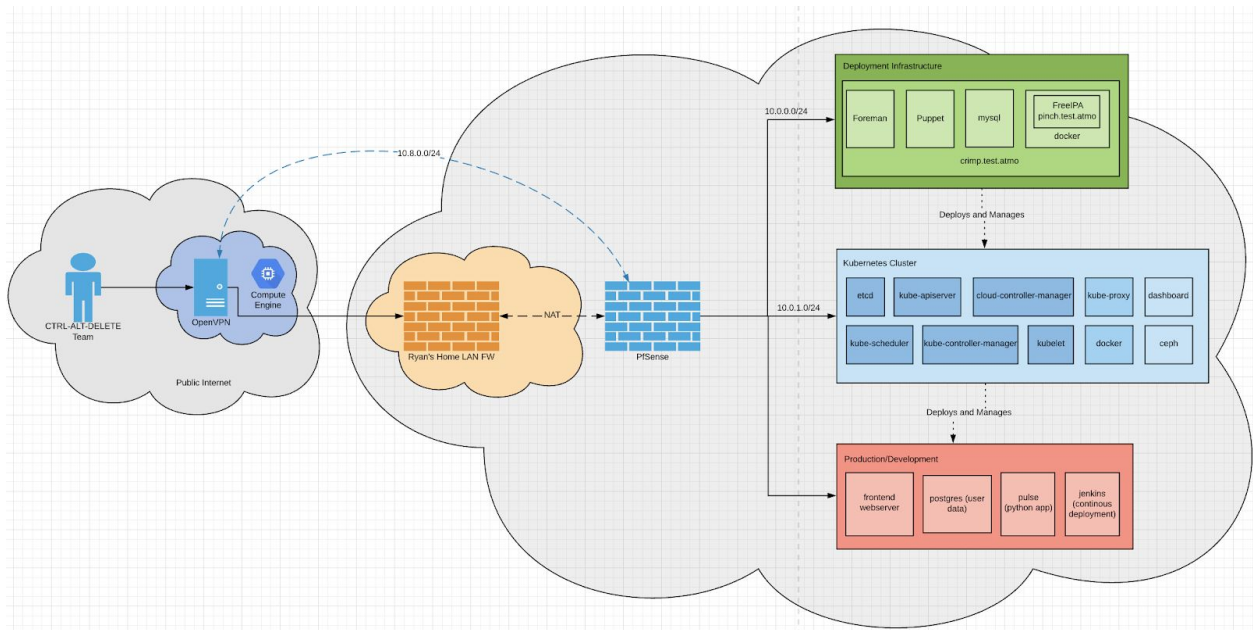
-[FUNCTIONAL] Include a colored meter on each post that represents upvote - downvote ratio

-[NON-FUNCTIONAL] The Reddit API will be used to find upvote and downvote count of each post that has been looked at, and a Bootstrap progress bar class will be used to visually convey vote ratio.

- YouTube Features

- Search Results
 - [FUNCTIONAL] Column that displays x most relevant video titles given a keyword
 - [NON-FUNCTIONAL] uses the search functionality in the Youtube Data API & html
- Video search popularity graph/chart
 - [FUNCTIONAL] Display the total number of views among the top x videos
 - [NON-FUNCTIONAL] Using Youtube Data API, Python, JS & HTML
- Website
 - Create user
 - [FUNCTIONAL] Allow for account creation.
 - [NON-FUNCTIONAL] Form built in HTML and JS.
 - Login
 - [FUNCTIONAL] Access user account. Will overlay the search screen. Taking style inspiration from the iCloud website.
 - [NON-FUNCTIONAL] Form built in HTML and JS.
 - Homepage
 - [FUNCTIONAL] Host the search screen and analytics.
 - [NON-FUNCTIONAL] Page built in HTML and JS.

Architecture Diagram



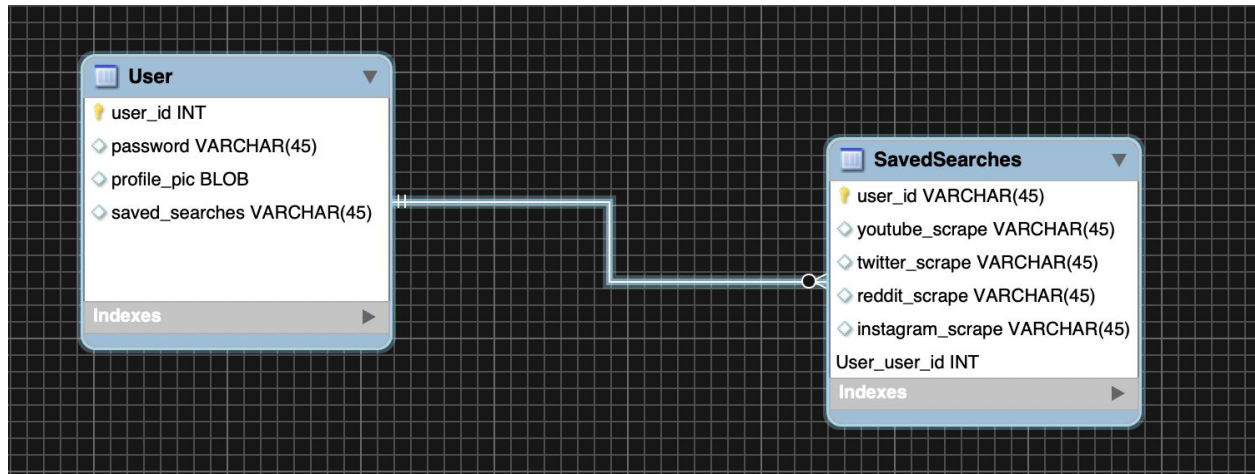
Web Service Design

The APIs that we will be collecting data and streaming to our website through json and jquery will be twitter, youtube, instagram, and reddit. Using these APIs we will scrape data from recent posts on each social media outlet to help create a census that will display on our website.

Database Design

Database Design

For our database, we are simply storing information about the user. Their login info and their saved searches from the website will be stored using postgres. The screenshot below is taken from mysql workbench, and can be used to create the DDL for creating the tables necessary in postgres. This is the info so far that will be stored, however, as we continue working on the website and figuring out and implementing the functionality further, i'm sure we will add/remove/change some of the tables/columns in our database.



Front-end Design

1. Six main pages: Home/Search, Search Results, About Pulse, Pulse Team, Create a User, User Account Page.
 - a. Home page- search bar centered, nav bar on left
 - b. Search Results - 4 cards with embedded results for Twitter, Reddit, YouTube, and analytics.
 - c. Register User - Form to collect information
 - d. Account Page - Display user data, past searches.
 - e. About Pulse - description of Pulse Product
 - f. Pulse Team - four cards, one for each team member, briefly describing ourselves
2. Login module pops up available from any page if the user has not yet logged in.
3. Create User/Account Page toggles depending of if user has logged in.

4. Navigational tab on left of page makes each page accessible from all areas of the site.

