Milestone 4 - Database Design

Software and Creation Scripts

We have chosen Postgres as our database system and it is hosted on Heroku so we can all work on it together. The name of the Heroku add-on is Heroku Postgres Hobby Dev because we will not have enough data or traffic to warrant upgrading to a paid version of the addon. We provision the database through Heroku's interface and then the rest of the work is done in the command line tool, psql. The following commands were run to create tables, specify data types and column names, and populate them with sample data.

Creating tables:

We created the tables in heroku psql so the commands used to make the tables were SQL commands.

CREATE TABLE posts (id serial, source text, title text, description text, author text, datetime timestamp not null, image text, likes int not null, dislikes int not null, link text);

CREATE TABLE users (id serial, email text not null, password text not null, first_name text not null, last_name text not null);

CREATE TABLE likes (id serial, user_id int not null, like_id int, dislike_id int);

Populating tables:
Early on to create sample data we would enter a few posts manual like such:

INSERT INTO posts (source, title, description, author, datetime, image, likes, dislikes, link) VALUES ('CNN', 'Breaking news: Trump to Duel Putin for Honor', 'A brief summary of recent events','Alan Paradise', '2018-11-15 00:00:00', 'imageLinkHere', 0, 0);

INSERT INTO users (email, password, first_name, last_name) VALUES ('bill@email.com','HASHEDPASSWORD','Alan','Paradise');

INSERT INTO likes (user_id, like_id, dislike_id) VALUES ('120492','420','233');

We did not want to have to manually enter too many lines of data so we wrote a python script to fill our posts table with the real data. We have yet to work much with the users and likes tables yet as our models for those are not as well defined. The script below fills our posts table with all the necessary data with one condition: Occasionally the data returned from the newsAPI has a field with a NULL value which does not work for our

table so we are currently excluding these posts however, we will need to find a way to solve this problem.

```
def insert_data():
    newsapi = NewsApiClient(api_key='8aac93d6a52b4e69b0d1fe72f526afe5')
    all_articles = newsapi.get_top_headlines(category="business",
                                             country='us',
                                             language='en',
                                             page=1)
    innercount = 0
    outercount = 0
    for j in all_articles['articles']:
        if j['title'] == None or j['source']['name'] == None or j['description'] == None or j['author'] == None or j['publishedAt'] == None or j['urlToImage']:
            continue
        outercount += 1
        test = False
        print("Outer Count: ", outercount)
        for i in Posts.objects.all().values():
            if i['link'] == j['url']:
                test = True
        if test == False:
            print("Link: ", i['link'], " URL: ", j['url'])
            innercount += 1
            print("Inner Count: ", innercount)
            p = Posts(source =j["source"]["name"],
            title=j['title'],
            description=j['description'],
            author="Adam Rosa",
            datetime = j['publishedAt'].split('T')[0] + " " + j['publishedAt'].split('T')[1].strip('Z'),
            image=j["urlToImage"],
            likes=0,
            dislikes=0,
            link=j['url'])
            p.save()
```

Data Model:

The diagram below shows our database and the relations between the tables. Arrows depict foreign keys and key icons depict the primary key of a table. The tables can be joined various different ways but each table has a distinct purpose and data set. Most columns do not accept NULL values because we are displaying the information and displaying NULL values will cause issues.