# Project 4 Part 2
# COP5612 – Fall 2017
# ReadMe

Team :-
Rohit Khobre - 99338795
Ritvik Ritvik - 62915992

## Introduction

The goal of this project is to implement the twitter engine to simulate the working of twitter system distributedly with large number of clients using websockets implemented using pheonix channels and GenSocket Client.

## Installing and Running

1. Run following commands from the directory which has mix.exs (twitterproject folder)

   **mix deps.get**
   **Mix ecto.create**
   **mix phx.server**

2. After printing of user details is completed, enter ‘ctrl + c’ to stop the execution.

## What is working

As output of this program, the following is performed

- Simulation of Twitter Engine.
- Use of WebSockets for communication between server and client. This is implemented using GenSocket client on client side and WebSocket API and Pheonix channels on server side.
- Register account (not displayed on screen)
- Send tweets with hashtags and mentions
- Follow a user
- Re-tweets
- Display of tweets when a specific hashtag is searched or for a particular mention
- Display of live tweets
- Display of profile for individual clients
- Display of timelines for each client
- Initially clients are registered with the server, but they are not active yet. When start_client is invoked, the client becomes a live connection and it starts it starts tweeting, following and other work and receives timeline from server.
- Simulated a Zipf distribution, so that the clients with more number of followers do have more number of tweets.

## Implementation Details:

1. We have used GenServer for twitter engine. This server is the single process which will keep the track of each & every user, tweet, subscription, retweet, follower, hashtag and mention.

2. This GenServer acts as a backend for the twitter simulation.
Gen server’s state is as following

[clientlist, clientNames, clientPassswds, follows , followsNames, tweets, tweetsid, alltweets]

3. We have used Phoenix Channels for communication between server and client. These channels internally implement WebSockets and transmit messages in JSON format.

4. We have used phoenix GenSocketClient library to implement clients.

5. Initially we create total number of users, and they are registered with the twitterengine topic at the server.

6. After this we add followers for each user randomly.

7. With this, users tweet and retweet randomly and for it genserver state changes, updating the details of users.

8. When username passed as mention, or hashtag is passed to the server, genserver returns the tweets containing corresponding mention or hashtag.

9. Client Simulator requests for the profile and Timeline for randomly selected users, and server returns their details to simulator.

In this simulation, we have considered above 3 tweets with hashtags and mentions. Out of these 3 tweets 1 tweet will be selected randomly and that tweet is broadcasted to the followers. Retweet is also based on the same logic.

## Performance

We tested the program with number of clients till 500. For different number of clients, program is giving CPU time to real time ratio around 3.