

# READ ME

The goal of this project is to use the Phoenix web framework to build Twitter-Clone using the Web-Socket functionality. The problem statement is to implement a JSON based API that represents all messages and their replies, design an engine using Phoenix and multiple clients to implement the Web interface.

## Team:

Kirti Desai, UFID – 04541017, [desai.kirti@ufl.edu](mailto:desai.kirti@ufl.edu)

Pratik Loya, UFID – 31716903, [ploya@ufl.edu](mailto:ploya@ufl.edu)

## Procedure to run the file:

1. Go to Path “Desai\_Loya\project4.2”
2. Run command on terminal
  - a) `mix deps.get`
  - b) `mix deps.comple`
  - c) `mix phx.server`
3. Open this url <http://localhost:4000/> on your browser

## Implementation:

We have created a web socket listening at `//localhost:4000/`. Clients will be joining different channels through the server, where each channel is unique and represents a followed user's name. So, whenever a tweet event (tweet request) is sent from a user, that tweet is broadcasted to all the users present in the topic bound to that user. So, all the live users will be getting the tweet directly whereas for offline users the tweet would be saved at the server

## Functionalities:

1. A user can register, login and logout from the system.
2. Project is simulated initially with 100 users as clients with random followers and each client sends three tweets (random) which are then being sent to its followers.
3. A user can follow any other user by providing the user-id using “follow button”.
4. A user can send the tweet to its followers. The tweets can include hashtags or mention by any other users. The tweet will be send to all its followers depending

## READ ME

upon if the followers are connected or disconnected. If the user is disconnected then that follower will receive the tweet when it get logged in to the system

5. A user can retweet any tweet that it has received from its followers using retweet button.
6. In the query section, the user can search tweets with a specific hashtags or query tweets that has mentioned a particular user.
7. The user will receive tweets when it is mentioned irrespective of whether it is a follower or not of the user that is tweeting.
8. The Server stores the information of the user, the tweets send by the user, the follower list, current active/inactive users, newsfeed of the user.
9. A user can create a group and add members to chat with them, and this group would be replicated to every user's group section

### Client Simulation Distribution

The clients are distributed among various categories depending on the number of followers:

Category 1 User has 70% to 90% followers

Category 2 User has 50% to 70% followers

Category 3 User has 20% to 50% followers

Category 4 User has 5% to 20% followers

### BONUS:

#### 1. Authentication:

Password are hashed using **sha256** protocol and saved into the table. When a user tries to login, password's hash value is matched with the stored value and appropriate response is sent.

#### 2. Group Chat:

A user can create a group and add members to chat with them, and this group would be replicated to every user's group section

**Youtube Link:** <https://youtu.be/bk1L7wxJsA>