**PROJECT-4.2**

**(Distributed Operating System)**

**Twitter Simulator**

**Team Members:**

**Dikshant Gupta : 68401523**

**Lavish Mehta : 79818557**

**Video Link (YouTube):** https://youtu.be/M6Nxu65EZvg

To run the application, do the following:

1. Unzip the Gupta-Mehta folder.

2. Run the project by using: **mix phx.server**

3. Open localhost:4000 in the browser to check with the UI.

We’ve successfully simulated 100 users, also from the previous part we used zipf distribution.

Register new users:

1. To create new user open the simulator and on the first page signup the new user with name and password.

Authentication:

2. The authentication is performed during the login function of the simulator. Also if the password is wrong the user cannot login into his account.

After successful login user can perform:

3. Follow: Just write the name of valid user you want to follow and you are added to that user's followers list and that user is added to your following list on the server side maintained in the ets table. Whatever tweets a user sends are going to be visible to its followers.

4 Tweeting: You can directly send tweets just by writing tweets in the text area and it will also be stored in the backend table.

5. Query : It gives all the tweets which have the hashtag specified by the user. The query must be prefixed with ‘#’. Write the tweet with the # and it will display the information.

6. News Feed: It gives the data with time and date for the user’s followed clients and their updated. Also the tweets mentioning anywhere the user.

Code working flow:

1. JSON based API - We used Phoenix channels to create an API wrapper over our previous code from Part 1. A code was written for all the existing functionalities we had in the previous part. To show the connectivity we used Phoenix web sockets. Front end and Backend are connected and accessed using sockets and channels.

2. Channels uses Websockets to communicate to clients. It is written in channel file. While socket functionalities are defined in socket.js file.

3. In the JavaScript part of the client, we have used sockets to connect to the server's phoenix channel. Also apps,js is used to collect the values from client UI and push them to channels for processing.