**PROJECT REPORT**

**TWITTER ENGINE USING ELIXIR**

Team Members:

1. Karan Manghani (UFID: 7986-9199) Email: [karanmanghani@ufl.edu](mailto:karanmanghani@ufl.edu)
2. Yaswanth Bellam (UFID: 2461-6390) Email: [yaswanthbellam@ufl.edu](mailto:yaswanthbellam@ufl.edu)

**INTRODUCTION:**

In this project, we use Elixir to make a Twitter engine. The basic functionalities of the engine include creating and deleting users, publishing and distributing tweets, retweeting, hashtags, and mentions. To demonstrate the project, we generate a number of users and make them follow each other, tweet, retweet, mention each other and use hashtags. This functionality is the basis for the Twitter application to be created in project 4.2.

**WORKING:**

First, we start the server and create the ETS tables where the data will be stored. Since all the data is stored in the server, the clients can interact with the server whenever a task is to be performed. Once all the tables are up, we create users, and assign followers to them at random. Then the users start tweeting with the tweets containing hashtags and mentions. Each functionality is a separate Genserver process, so the whole system runs asynchronously.

The tables available are:

1. Users
2. Tweets
3. Followers
4. Hashtags
5. Mentions
6. News Feed

Hashtags and Mentions:

We are using a regular expression to find hashtags and mentions in any tweet. These are stored in respective tables so that they can be searched or manipulated easily.

**PERFORMANCE METRICS:**

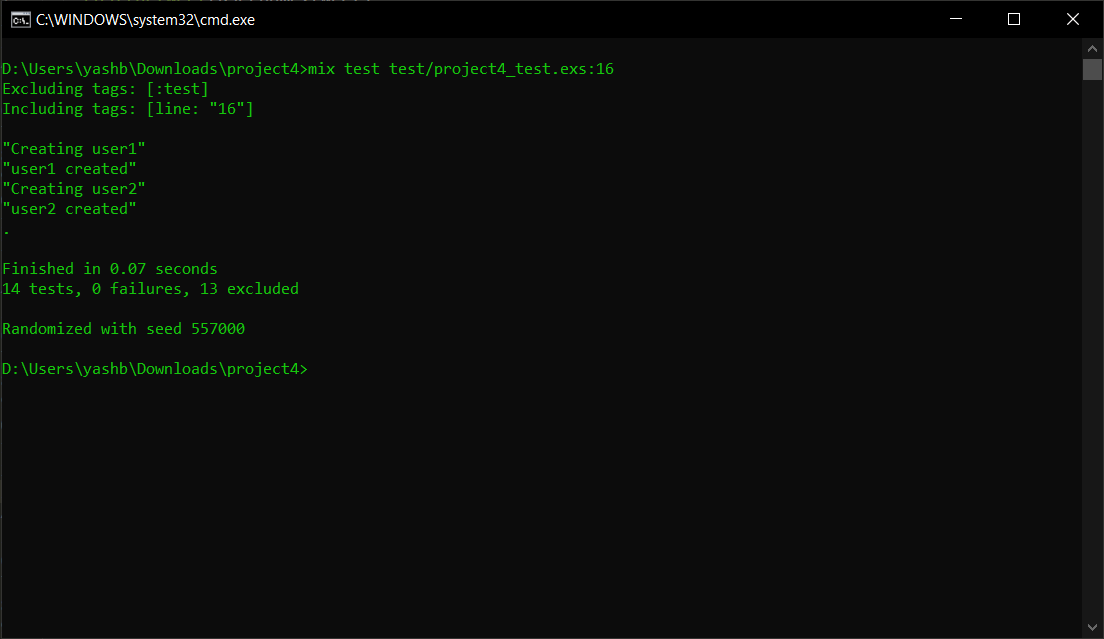
Since most of the functionality in our project is asynchronous, some operations take near constant time. So there isn’t much discrepancy in the time taken to create the tables or delete the accounts.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Number of users** | **Number of tweets** | **Time to create tables** | **Time to tweet (all users)** | **Time to delete account** | **Total time** |
| 10 | 1 | 2000 | 1063 | 1000 | 8093 |
| 100 | 10 | 2016 | 1078 | 1016 | 8156 |
| 500 | 50 | 2031 | 2578 | 1000 | 9609 |
| 750 | 75 | 2016 | 5937 | 1000 | 12953 |
| 1000 | 100 | 2032 | 13442 | 1000 | 20422 |
| 1500 | 150 | 2015 | 58063 | 1015 | 65109 |
| 2000 | 200 | 2031 | 164922 | 1000 | 171953 |

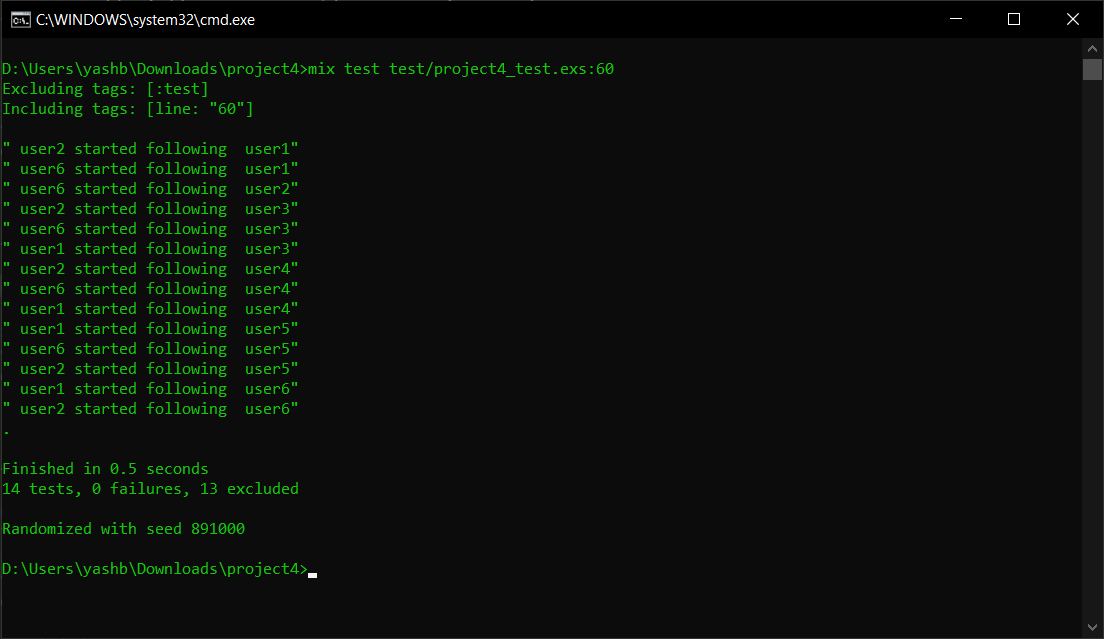
So, as we can see, the system scales very well with increase in load, i.e. both the number of users and the number of tweets per user.

**SAMPLE EXUNIT RESULTS:**

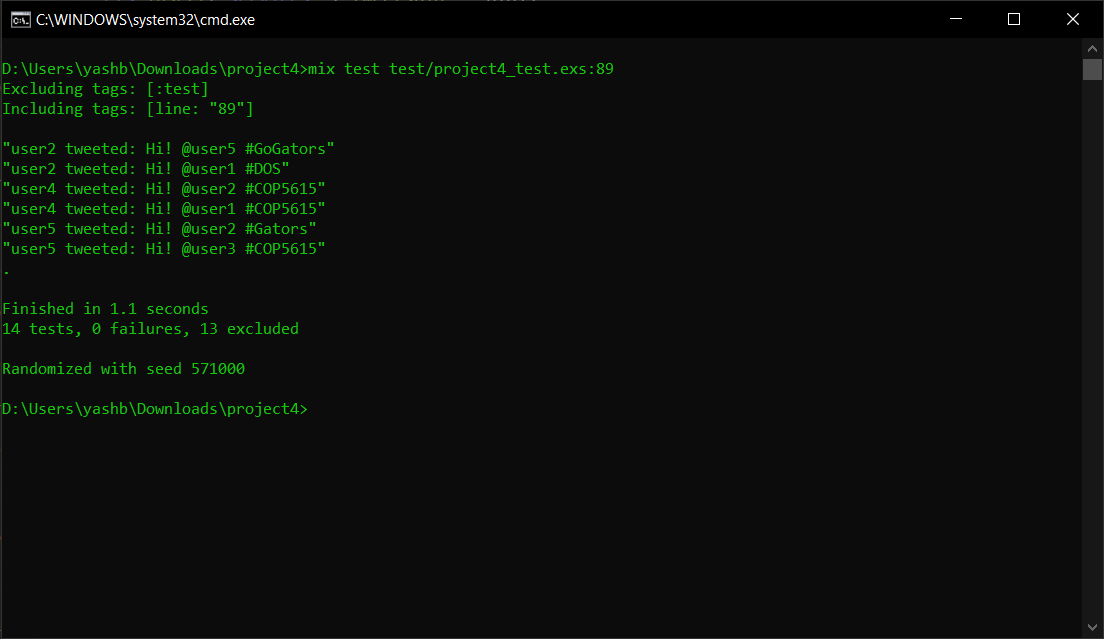
Testing user creation:



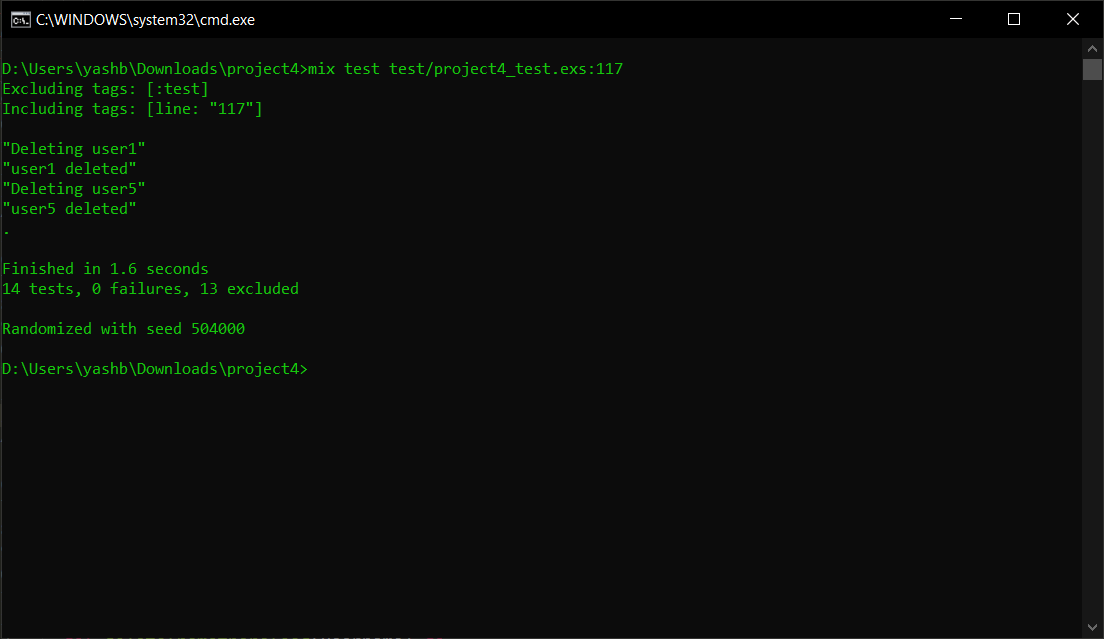
Testing followers:



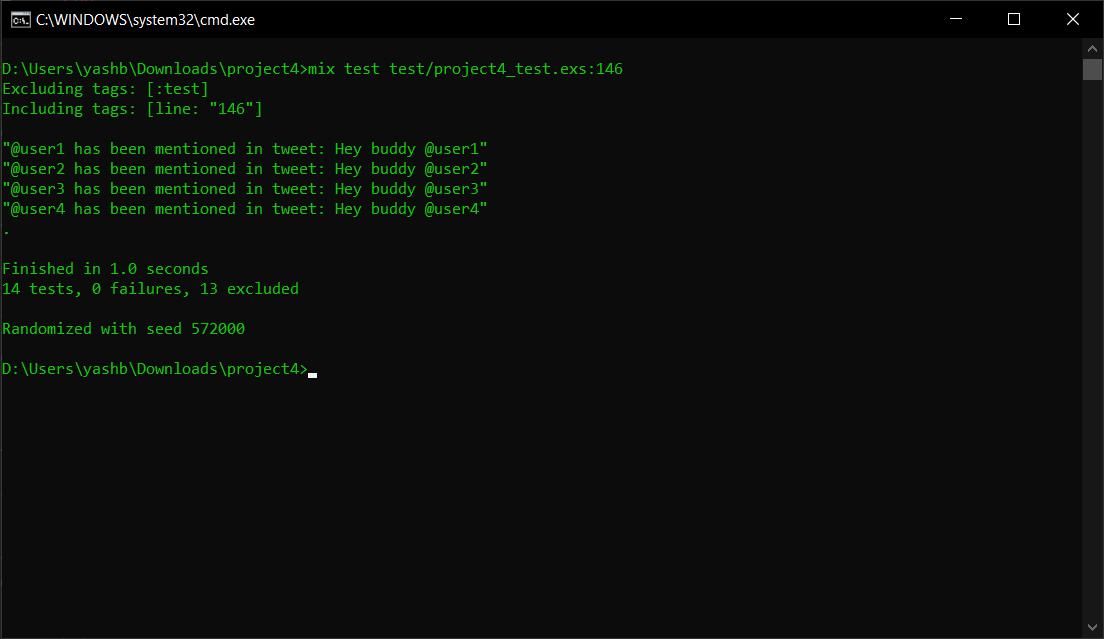
Testing tweets:



Testing delete account:



Testing mentions:



Testing hashtags:

