# Distributed Operating Systems

## Project 4 – Part 1

## Implementation of Twitter Server supporting REST API and WebSocket's

**Deepthi Byneedi – 36871955**

**Sanjay Reddy Banda – 58782239**

### How to Run:

Server: Navigate to directory which has 'Project4_Part2.fsproj' document and run 'dotnet run' command.

Client: open 'client.html' in any browser

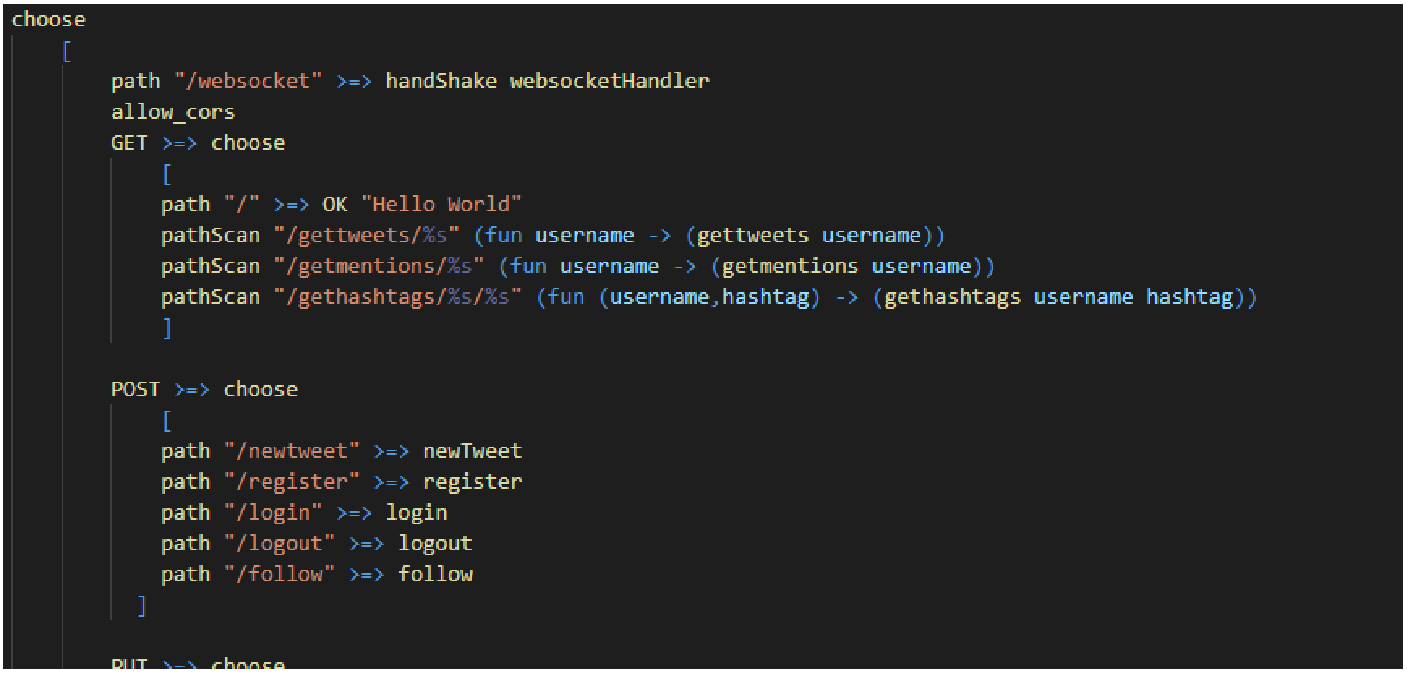
### Implementation:

### Rest API:

1. We are able to achieve rest services using suave.io package.
2. Server will accept '/newtweet', '/register', '/login', '/logout' and '/follow' requests as a POST requests.
3. Server will accept '/gettweets/%s', '/getmentions/%s', '/gethashtags' requests as a GET requests.
4. Once server accepts the request then it will delegate work to the other functions and actors accordingly as per the request.

### WebSocket:

1. Whenever a client successfully logs in to the server a WebSocket connection is initiated by the client with "<serveraddr>/websocket" URL.
2. Once the server accepts the request. Client and server will exchange handshake messages and client will send its username as "UserName:username" by WebSocket to the server.
3. After receiving message from client on WebSocket server will store the WebSocket reference with the client id.
4. Whenever actor needs to push a live update to the client, it picks the necessary WebSocket address and puts message to client on WebSocket.
5. On client side when it receives a message from server via WebSocket it displays to the user in the live feed tab.



```
choose
    [
        path "/websocket" >=> handShake websocketHandler
        allow_cors
        GET >=> choose
            [
            path "/" >=> OK "Hello World"
            pathScan "/gettweets/%s" (fun username -> (gettweets username))
            pathScan "/getmentions/%s" (fun username -> (getmentions username))
            pathScan "/gethashtags/%s/%s" (fun (username,hashtag) -> (gethashtags username hashtag))
            ]

        POST >=> choose
            [
            path "/newtweet" >=> newTweet
            path "/register" >=> register
            path "/login" >=> login
            path "/logout" >=> logout
            path "/follow" >=> follow
          ]

        PUT >=> choose
```

## Rest API’s Request and Response formats:

| Request | Request Body | Response to the requests |
|---|---|---|
| POST<br>‘/register’ | {<br>UserName: <username>,<br>Password: <password><br>} | {<br>Comment: <Successful/Unsuccessful message>,<br>Content: [],<br>status:1,<br>Error:<true/false><br>} |
| POST<br>‘/login’ | {<br>UserName: <username>,<br>Password: <password><br>} | {<br>Comment: <Successful/Wrong password/Not Registered>,<br>Content: [],<br>status:<0/1/2>,<br>Error:<true/false><br>} |
| POST<br>‘/logout’ | {<br>UserName: <username><br>} | {<br>Comment: <Successful/Not logged in/Not Registered>,<br>Content: [],<br>status:<0/1>,<br>Error:<true/false><br>} |
| POST<br>‘/newtweet’ | {<br>Tweet: <tweet>,<br>UserName: <username><br>} | {<br>Comment: <Successful/Not logged in/Not Registered>,<br>Content: [],<br>status:<0/1/2>,<br>Error:<true/false><br>} |
| POST<br>‘/follow’ | {<br>UserName:<username>,<br>Following:<follow><br>} | {<br>Comment: <Successful/Not logged in/Not Registered/ follower do not exist/ already following>,<br>Content: [],<br>status:<0/1/2>,<br>Error:<true/false><br>} |
| GET<br>‘/gettweets/<uname>’ | | {<br>Comment: <Successful/Not logged in/Not Registered>,<br>Content: [<tweet1>, <tweet2>,…],<br>status:<0/1/2>,<br>Error:<true/false><br>} |
| GET<br>‘/gethashtags/<uname>/<htag>’ | | {<br>Comment: <Successful/Not logged in/Not Registered>,<br>Content: [<tweet_with_htag1>, < tweet_with_htag2>,…],<br>status:<0/1/2>,<br>Error:<true/false><br>} |
| GET<br>‘/getmentions/<uname>’ | | {<br>Comment: <Successful/Not logged in/Not Registered>,<br>Content: [<mention_in_tweet1>, < mention_in_tweet2>,…],<br>status:<0/1/2>,<br>Error:<true/false><br>} |