PROJECT 4: TWITTER CLONE

READ ME

1. Project Members –

   Aditya Chaudhary, 1278-6020

   Shruti Parihar, 9215-3237

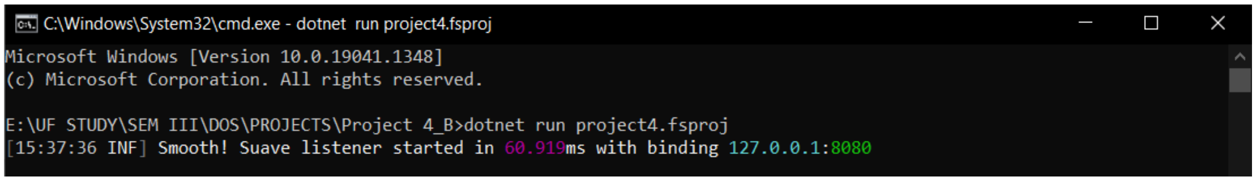
2. How to run the code –

   dotnet run project4.fs

   This runs the server and reserves port 8080 on our machine. We can access the port 8080 via the web browser.

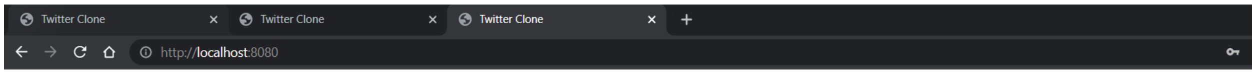
3. What is working –

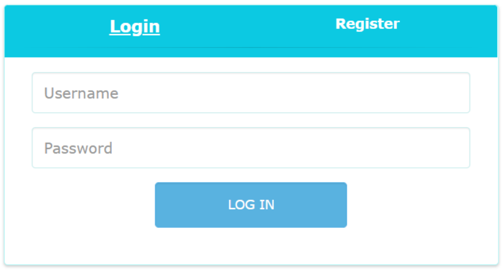
- The server connects to the User Interface.
- Implemented the WebSocket interface.
- Initialized the WebSocket interface.
- Client runs on the browser at localhost:8080.



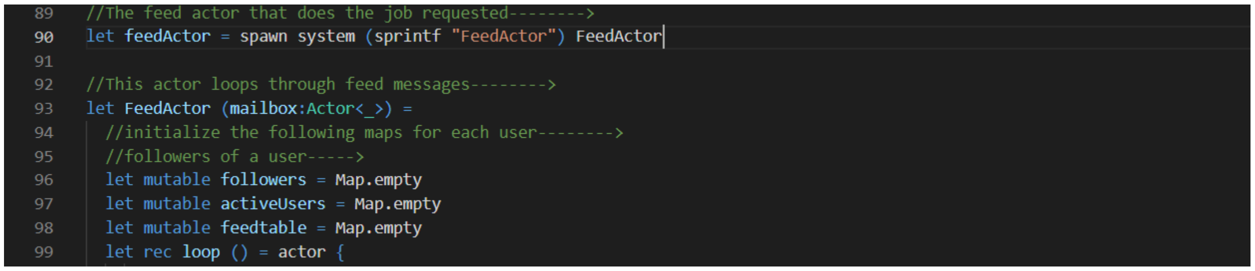
```
C:\Windows\System32\cmd.exe - dotnet  run project4.fsproj
Microsoft Windows [Version 10.0.19041.1348]
(c) Microsoft Corporation. All rights reserved.

E:\UF STUDY\SEM III\DOS\PROJECTS\Project 4_B>dotnet run project4.fsproj
[15:37:36 INF] Smooth! Suave listener started in 60.919ms with binding 127.0.0.1:8080
```

- Server instantiates the Login.html page once it is started.
- For multiple clients, multiple connections(new tabs) needed to be opened.

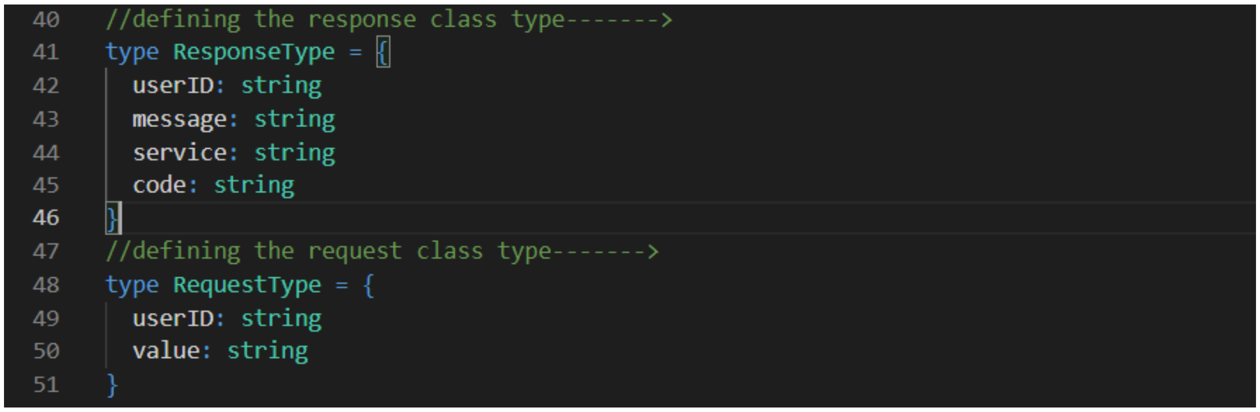




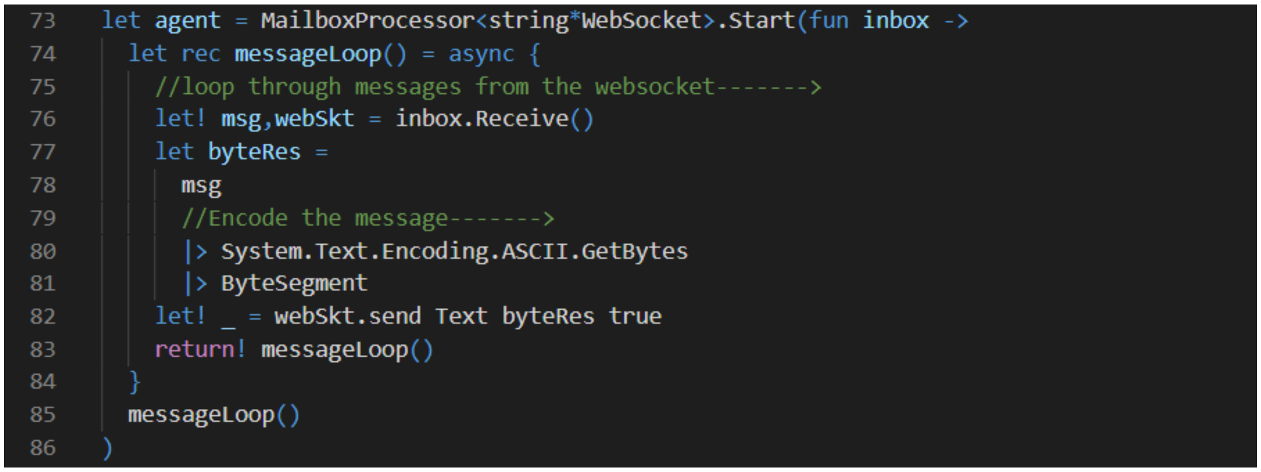
- The Client that can access the WebSocket.


```
//The feed actor that does the job requested-------->
let feedActor = spawn system (sprintf "FeedActor") FeedActor

//This actor loops through feed messages-------->
let FeedActor (mailbox:Actor<_>) =
  //initialize the following maps for each user-------->
  //followers of a user----->
  let mutable followers = Map.empty
  let mutable activeUsers = Map.empty
  let mutable feedtable = Map.empty
  let rec loop () = actor {
```

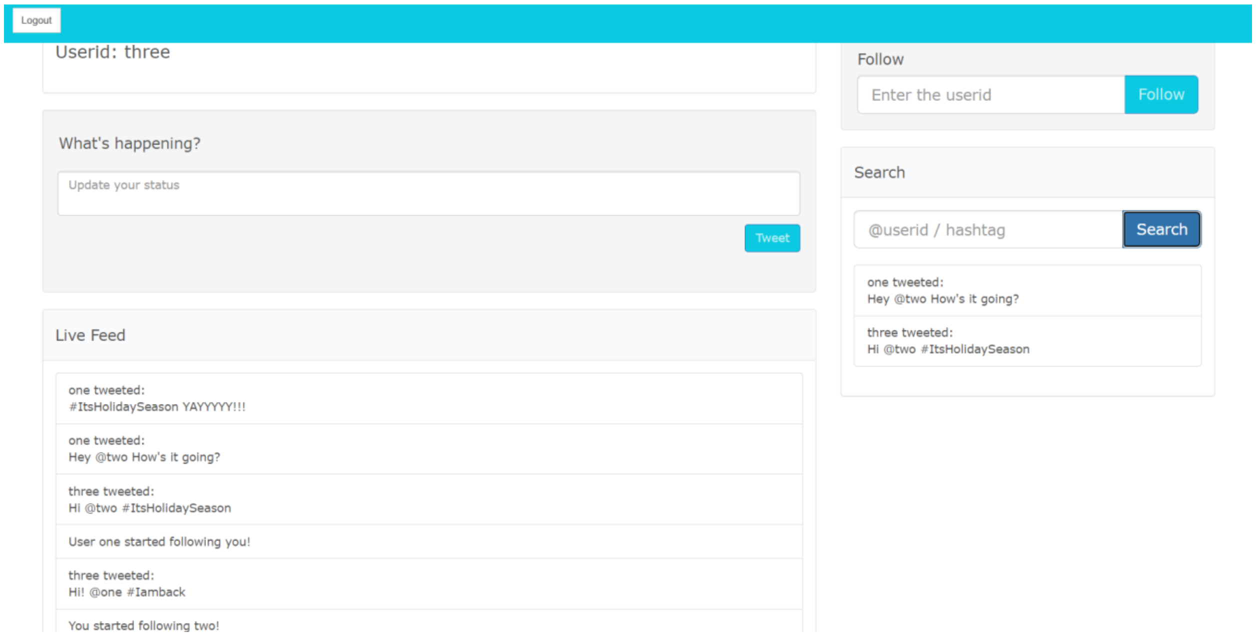
- We designed a JSON based API that represents all messages and their replies (including errors).


```
//defining the response class type------->
type ResponseType = {
  userID: string
  message: string
  service: string
  code: string
}
//defining the request class type------->
type RequestType = {
  userID: string
  value: string
}
```

- All the messages passed through the WebSocket are encoded.


```
let agent = MailboxProcessor<string*WebSocket>.Start(fun inbox ->
  let rec messageLoop() = async {
    //loop through messages from the websocket------->
    let! msg,webSkt = inbox.Receive()
    let byteRes =
      msg
      //Encode the message------->
      |> System.Text.Encoding.ASCII.GetBytes
      |> ByteSegment
    let! _ = webSkt.send Text byteRes true
    return! messageLoop()
  }
  messageLoop()
)
```

- Twitter engine successfully allows users to Tweet, Follow, Mention, Create Hashtag, Search Mention, Search Hashtag.


Logout

Userid: three

What's happening?

Update your status

Tweet

Live Feed

one tweeted:
#ItsHolidaySeason YAYYYYY!!!

one tweeted:
Hey @two How's it going?

three tweeted:
Hi @two #ItsHolidaySeason

User one started following you!

three tweeted:
Hi! @one #Iamback

You started following two!

Follow

Enter the userid

Follow

Search

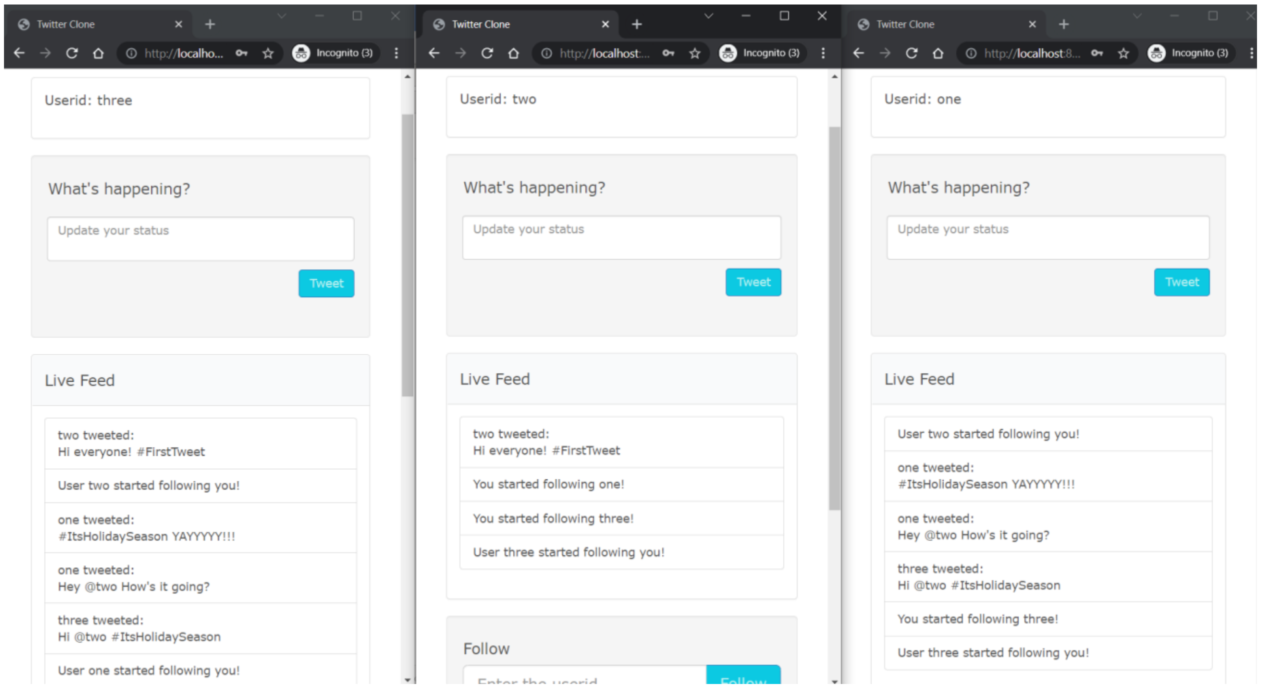
@userid / hashtag

Search

one tweeted:
Hey @two How's it going?

three tweeted:
Hi @two #ItsHolidaySeason

- Update the feeds of the Users affected by the event. This happens concurrently. This increases efficiency and reduces data inconsistency.


Twitter Clone

Userid: three

What's happening?

Update your status

Tweet

Live Feed

two tweeted:
Hi everyone! #FirstTweet

User two started following you!

one tweeted:
#ItsHolidaySeason YAYYYYY!!!

one tweeted:
Hey @two How's it going?

three tweeted:
Hi @two #ItsHolidaySeason

User one started following you!

Twitter Clone

Userid: two

What's happening?

Update your status

Tweet

Live Feed

two tweeted:
Hi everyone! #FirstTweet

You started following one!

You started following three!

User three started following you!

Follow

Twitter Clone

Userid: one

What's happening?

Update your status

Tweet

Live Feed

User two started following you!

one tweeted:
#ItsHolidaySeason YAYYYYY!!!

one tweeted:
Hey @two How's it going?

three tweeted:
Hi @two #ItsHolidaySeason

You started following three!

User three started following you!