

# DOSP Project 4 Part 2:

## Twitter Clone and WebSocket Interface

### TEAM:

Anisha Ashok Wadhwani (UFID: 66406383)

Mohit Parmanand Budhwani (UFID: 44369932)

### PROJECT DESCRIPTION

In part 1, we implemented a twitter client and simulator to create users and simulate the functionalities of twitter. In this project, we extent the client and create a WebSocket interface. We have implemented a User Interface which allows users to do the following functionalities :

- ➔ Register/Create their profile
- ➔ Tweet
- ➔ Subscribe to other users by their username
- ➔ Search tweets by Hashtags
- ➔ Search tweets by Mention
- ➔ Search tweets of a subscriber/user

### STEPS TO EXECUTE:

- ➔ Download the zip file
- ➔ Run the command : make run
- ➔ Open the link : localhost:8080/registerpage.html

### PROGRAM FLOW :

Open url -> localhost:8080/registerpage.html

Enter Details and click on register

<<Program redirects to next page>>

<<HTTP request sent to create websocket connection>>

<<Server responds and websocket connection established>>

User can now subscribe, tweet, retweet, search etc...

<<User 1 subscribes to user 2>>

<<User 2 tweets "First tweet">>

User1's feed is automatically updated with its subscribed user's tweets

## IMPLEMENTATION:

Below are different technologies and the description of modules and functions that we have implemented in the project

### Technologies and concepts:

Websocket –

Websocket connection allows the server to send messages to the client irrespective of client sending a request. HTTP on the other hand only works with request response pairs. The server cannot send any messages to the client without a request.

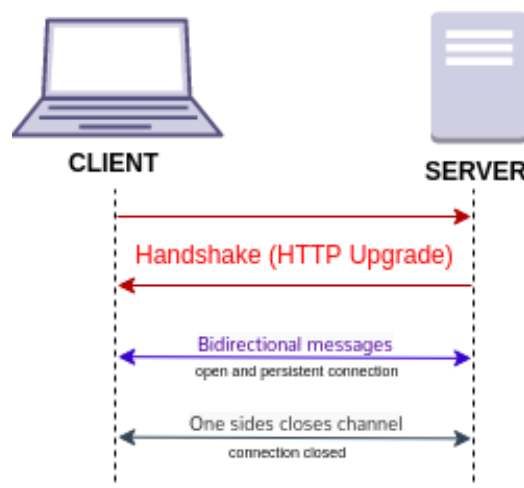


Figure 1 : Websocket connection

Cowboy =

Cowboy is server used for Erlang. It allows to create a whole web application using Erlang.

#### Build Application using cowboy:

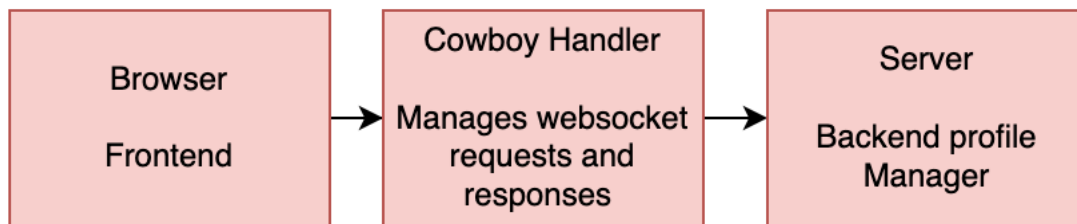
➔ Download erlang.mk

```
wget https://erlang.mk/erlang.mk
```

➔ Run :

```
make -f erlang.mk bootstrap bootstrap-rel  
make run
```

## Different modules of the project:



## Mapping of frontend HTML pages and handler

```
start(_Type, _Args) ->

    ServerPID = twitterserver:start_server(),

    Dispatch = cowboy_router:compile([
        %% {HostMatch, list({PathMatch, Handler module name (erl), InitialState})}
        {'_', [
            {"[/[...]", cowboy_static, {priv_dir, twitter, "assets"}}]} % html pages path ;
    ]),

    DispatchWs = cowboy_router:compile([
        {'_', [
            {""/homepage.html", feed_handler, [{serverPID,ServerPID}]}
        ]}
    ]),

    {ok, _} = cowboy:start_clear(
        my_http_listener,
        [{port, 8080}],
        #{env => #{dispatch => Dispatch}}
    ),

    {ok, _} = cowboy:start_clear(
        ws,
        [{port, 8889}],
        #{env => #{dispatch => DispatchWs}}
    ),

    twitter_sup:start_link().

stop(_State) ->
    ok.
```

Setting port and mapping routes and handler to the cowboy server

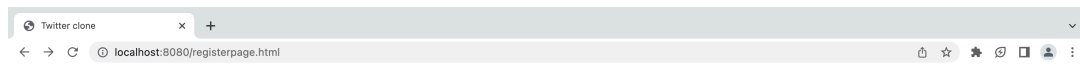
## Handler file

```
1  -module(feed_handler).
2  -behavior(cowboy_websocket).
3
4  -export([init/2]).
5  -export([websocket_init/1]).
6  -export([websocket_handle/2]).
7  -export([websocket_info/2]).
8
9
10 % State : serverPID, profile
11
12 > init(Req, State) -> ...
27 > websocket_init(State) -> ...
35 > websocket_handle({text, Data}, State) -> ...
108 > websocket_info({updateFeed,Data}, State) -> ...
118 > websocket_info({searchBegin,{Query,Keyword}}, State) -> ...
124 > websocket_info({searchEnd,_Data}, State) -> ...
127 > websocket_info({searchResult,{_Type,_Keyword,Tweets}}, State) -> ...
137 > websocket_info({allTweets,{UsernameSearch,Tweets}}, State) -> ...
```

Main handler functions

## SCREENSHOTS:

### Opening Page:



#### REGISTER ACCOUNT

Username:

Email ID:

Password:

### Entering data :

#### REGISTER ACCOUNT

Username:

Email ID:

Password:

### Homepage:

**Hello anishaw**

**Live Feed**

**Search results**

Subscribe to user:

Hello mohit

Tweet here.....

Tweet

Live Feed

localhost:8080 says  
User subscribed

OK

anishaw

Subscribe

Search box....

Hashtag

Mention

Subscriber

Search results

Tweet:

Hello anishaw

Tweet here.....

Tweet

Live Feed

anishaw

My first #tweet !!!!!

Retweet

Subscribe to User...

Subscribe

Search box....

Hashtag

Mention

Subscriber

Search results

Live feed and Retweet

Hello mohit

Tweet here.....

Tweet

Live Feed

anishaw

My first #tweet !!!!!

Retweet

mohit

[Retweet] anishaw:My first #tweet !!!!!

Retweet

Subscribe to User...

Subscribe

Search box....

Hashtag

Mention

Subscriber

Search results

## Search for tweets by user : anishaw

Hello user3

Tweet here.....

Tweet

Subscribe to User...

Subscribe

Live Feed

anishaw

Hashtag

Mention

Subscriber

Search results

localhost:8080/homepage.html?uname=user3&email=user%40example.in&password=user

Hello user3

Tweet here.....

Tweet

Subscribe to User...

Subscribe

Live Feed

Search box....

Hashtag

Mention

Subscriber

Search results

anishaw

My first #tweet !!!!!

anishaw

my second tweet #yay #update

## Search by hashtag

Hello mohit

Tweet here.....

Tweet

Subscribe to User...

Subscribe

Live Feed

anishaw

My first #tweet !!!!!

Retweet

mohit

[Retweet] anishaw:My first #tweet !!!!!

Retweet

anishaw

my second tweet #yay #update

Retweet

#update

Hashtag

Mention

Subscriber

Search results

## Hello mohit

Tweet here.....

Tweet

Subscribe to User...

Subscribe

### Live Feed

*anishaw*

My first #tweet !!!!!

Retweet

Search box....

Hashtag

Mention

Subscriber

*mohit*

[Retweet] anishaw:My first #tweet !!!!!

Retweet

### Search results

*anishaw*

my second tweet #yay #update

Retweet

*anishaw*

my second tweet  
#yay #update

## Search by Mention

## Hello user3

In a party with @anishaw @mohit

Tweet

Subscribe to User...

Subscribe

### Live Feed

Search box....

Hashtag

Mention

Subscriber

## Hello anishaw

Tweet here.....

Tweet

Subscribe to User...

Subscribe

### Live Feed

*anishaw*

My first #tweet !!!!!

Retweet

@anishaw

Hashtag

Mention

Subscriber

*anishaw*

my second tweet #yay #update

Retweet

### Search results

*user3*

In a party with @anishaw @mohit

Retweet