

## Project 4 Phase 1

Submitted By:-

Prakriti Vardhan (UF ID – 13325315, email- prakriti19@gmail.com)

Pratyoush Kumar Srivastava (UF ID – 79043951, email- pratyoush93@gmail.com )

### **How to run the project:**

1. First run Main.scala. It can take 2-5 minutes to run it. Meanwhile, there will be a few registration timeout errors that can be ignored. Once you get a message string "Ready!" you can proceed to step 2.
2. Now run, fbclient.scala. This is where the simulation take place. It will again take 2-5 minutes to run and will automatically start posting, viewing profiles etc.

### **What did we do:-**

Simulation of Facebook:-

For simulation of our Facebook we have used a statistics[1] for how certain percentage of people have certain number of friends.

So we are creating users with dummy details and a friend list. Users can later upload/update their profile details like name, dob, age and profile picture.

Users can add albums to their profile.

They can post on their friend's wall, view their wallposts. They can also delete posts, friends, profile and page post (admin only).

Following is the API for all the functionalities:-

#### **1. Add friends**

```
path(IntNumber / "user" / IntNumber / "addfriend"){ (userid, friendid) //userid wants to add friendid
```

#### **2. Delete friend**

```
path(IntNumber / "user" / IntNumber / "deletefriend"){ (userid, friendid)
```

#### **3. View profile**

```
path(IntNumber / "profile")
```

#### **4. Delete Profile**

```
path(IntNumber / "profile" / "deleteprofile")
```

## 5. Post on wall

```
path(IntNumber / "wallposts" / IntNumber / "addpost" ) { (userid, friendid) //friendid posts on
userid
```

## 6. Delete own wall Post

```
path(IntNumber / "wallposts" / "deletepost" / IntNumber) { (userid, postid) => //check if friend
is in friendlist
```

## 7. Update own profile.

```
path(IntNumber / "profile" / "update") { userid
parameter("name".?, "age".?, "dob".?){ (newName,newAge,newDob)
```

## 8. View Wallpost

```
path(IntNumber / "wallposts")
```

## 9. Subscribe of the page can post on it.

```
path(IntNumber / "page" / "addpost" / IntNumber){ (pageid, userid)
```

## 10. See subscribers of a page

```
path(IntNumber / "page" / "subscribers"){ pageid=>
```

## 11. Delete the post of a page

```
path(IntNumber / "page" / IntNumber / "user" / "deletepost" / IntNumber) { (pageid, userid, postid)
```

## 12. View Page

```
path(IntNumber / "page"){ pageid
```

## 13. Add/Update profile Pic

```
path(IntNumber / "getprofilepic") { id
```

## 14. create Album

```
path(IntNumber / "addprofilepic") { id
```

## 15. Show Album/ Friends album.

```
path(IntNumber / "album" / IntNumber ) { (userid, albumid)=> //user accesses his album
path(IntNumber / "user" / IntNumber / "showalbum" / IntNumber / "friend"){ (userid,
albumid, friendid) //friend accesses user's album
```

## 16. View own Album and Friend's albums.

```
path(IntNumber / "user" / IntNumber / "showalbum" / IntNumber / "friend"){ (userid, albumid,
friendid) //friend accesses user's album
```

## 17. View Friendlist

```
path(IntNumber / "friendlist") { id
```

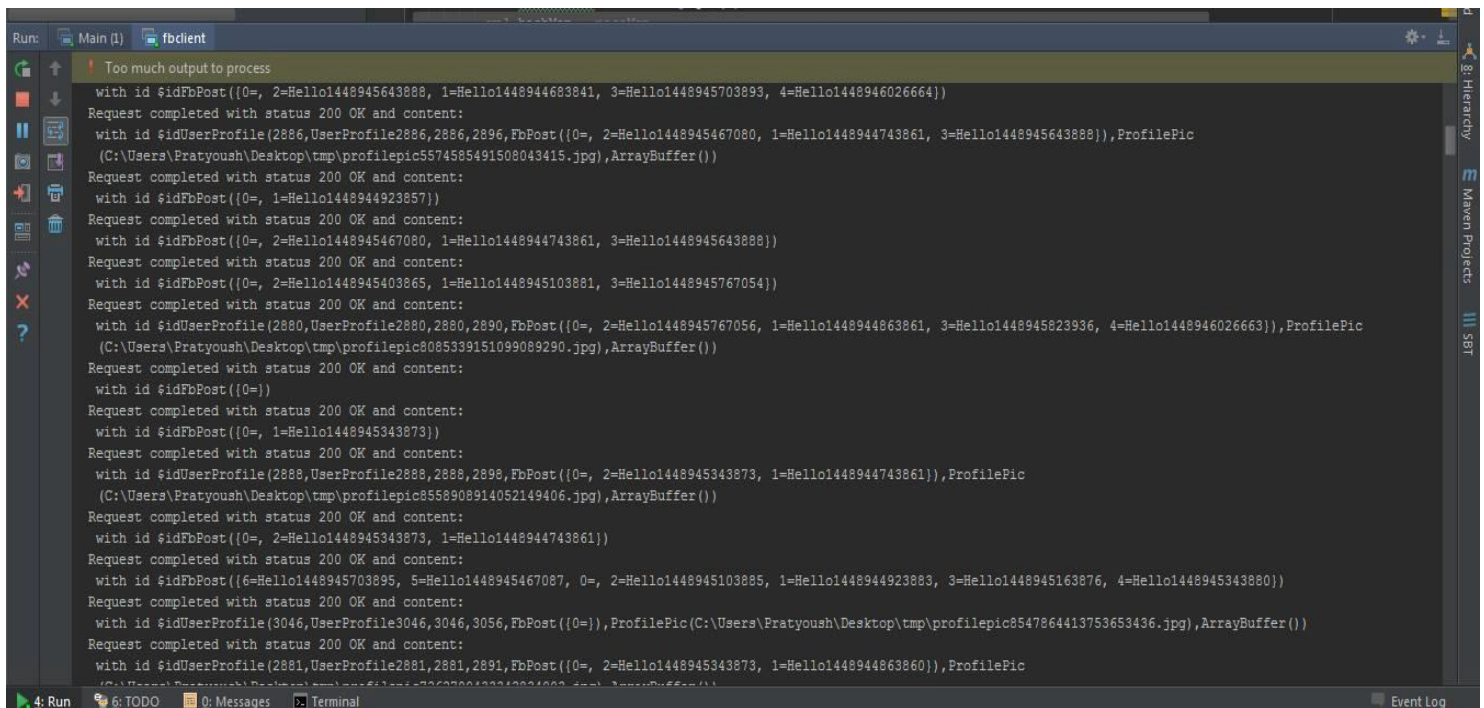
## 18. Allow Admin to delete post

```
path(IntNumber / "page" / IntNumber / "user" / "deletepost" / IntNumber) { (pageid, userid, postid)  
//check if friend is in friendlist
```

Sample URLs that can be directly used in the browser:-

1. <http://localhost:8080/60/profile>
2. <http://localhost:8080/5/friendlist>
3. <http://localhost:8080/21/profile/deleteprofile>
4. <http://localhost:8080/7/page>
5. <http://localhost:8080/60/wallposts>
6. <http://localhost:8080/60/getprofilepic>
7. <http://localhost:8080/60/album/1>

Sample Output:-



The screenshot shows a REST client interface with a list of requests and their corresponding responses. The responses are all 200 OK status codes. The requests include various endpoints for user profiles, friendlists, and posts, with some including profile pictures and user IDs.

```
Run: Main (1) fbclient  
Too much output to process  
with id $idFbPost({0=, 2=Hello1448945643888, 1=Hello1448944683841, 3=Hello1448945703893, 4=Hello1448946026664})  
Request completed with status 200 OK and content:  
with id $idUserProfile(2886,UserProfile2886,2886,2896,FbPost({0=, 2=Hello1448945467080, 1=Hello1448944743861, 3=Hello1448945643888}),ProfilePic  
(C:\Users\Pratyoush\Desktop\tmp\profilepic5574585491508043415.jpg),ArrayBuffer())  
Request completed with status 200 OK and content:  
with id $idFbPost({0=, 1=Hello1448944923857})  
Request completed with status 200 OK and content:  
with id $idFbPost({0=, 2=Hello1448945467080, 1=Hello1448944743861, 3=Hello1448945643888})  
Request completed with status 200 OK and content:  
with id $idFbPost({0=, 2=Hello1448945403865, 1=Hello1448945103881, 3=Hello1448945767054})  
Request completed with status 200 OK and content:  
with id $idUserProfile(2880,UserProfile2880,2880,2890,FbPost({0=, 2=Hello1448945767056, 1=Hello1448944863861, 3=Hello1448945823936, 4=Hello1448946026663}),ProfilePic  
(C:\Users\Pratyoush\Desktop\tmp\profilepic8085339151099089290.jpg),ArrayBuffer())  
Request completed with status 200 OK and content:  
with id $idFbPost({0=})  
Request completed with status 200 OK and content:  
with id $idFbPost({0=, 1=Hello1448945343873})  
Request completed with status 200 OK and content:  
with id $idUserProfile(2888,UserProfile2888,2888,2898,FbPost({0=, 2=Hello1448945343873, 1=Hello1448944743861}),ProfilePic  
(C:\Users\Pratyoush\Desktop\tmp\profilepic8558908914052149406.jpg),ArrayBuffer())  
Request completed with status 200 OK and content:  
with id $idFbPost({0=, 2=Hello1448945343873, 1=Hello1448944743861})  
Request completed with status 200 OK and content:  
with id $idFbPost({6=Hello1448945703895, 5=Hello1448945467087, 0=, 2=Hello1448945103885, 1=Hello1448944923883, 3=Hello1448945163876, 4=Hello1448945343880})  
Request completed with status 200 OK and content:  
with id $idUserProfile(3046,UserProfile3046,3046,3056,FbPost({0=}),ProfilePic(C:\Users\Pratyoush\Desktop\tmp\profilepic8547864413753653436.jpg),ArrayBuffer())  
Request completed with status 200 OK and content:  
with id $idUserProfile(2881,UserProfile2881,2881,2891,FbPost({0=, 2=Hello1448945343873, 1=Hello1448944863860}),ProfilePic  
(C:\Users\Pratyoush\Desktop\tmp\profilepic8547864413753653436.jpg),ArrayBuffer())
```

On the client side we have create users with high, medium and low activity.

High Activity Users are:-

1. Viewing their profile
2. Viewing their wallposts
3. Viewing friend's profile

4. Posting on wall
5. Viewing pages

Medium Activity Users are:-

1. Viewing their profile
2. Viewing their wallposts
3. Viewing friend's profile
4. Posing on wall

Low Activity users are:-

1. Viewing their profile
2. Viewing friend's profile

For the Post Method of Uploading Profile pic and Album we are using Chrome extension called "Postman" and we are able to upload the image to a temporary folder in our local system and associate it with the user who uploads it.

For album also we are using a POST request using Postman we are able to upload multiple images and associate it with an albumID which is being assigned to the user who is uploading it.

We have tested our project by simulating 5000 users. Out of these, 2400 are highly active users. 2100 are moderately active users. The remaining 500 users are low activity users.

References:-

[1] <https://www.facebook.com/notes/facebook-data-science/anatomy-of-facebook/10150388519243859>

[2] <https://www.youtube.com/watch?v=XPuOlWEvmw>

[3] <https://github.com/whataboutbob/spray-file-upload>

[4] <https://github.com/adamw/spray-tutorial>

[5] <https://developers.facebook.com/docs/graph-api/reference>

[6] <http://www.adweek.com/socialtimes/infographic-gwi-active-user-percentage-4q-2014/617078>