NYU TANDON SCHOOL OF ENGINEERING

Final Project Report

UCoN

University students' Social Network

Submitted by:

Anshul Sharma (as10950)
Mayank Lamba(ml5711)

Contents

1. Introduction

With the advent of globalization, the ecosystem of universities in the US has become more vibrant than ever. This provides us with all the more reason to have an application that can bring together people from different cultures and communities. Hence, UCoN: a University students' Social Network. It helps users connect with each other, interact, converse, and share experiences.

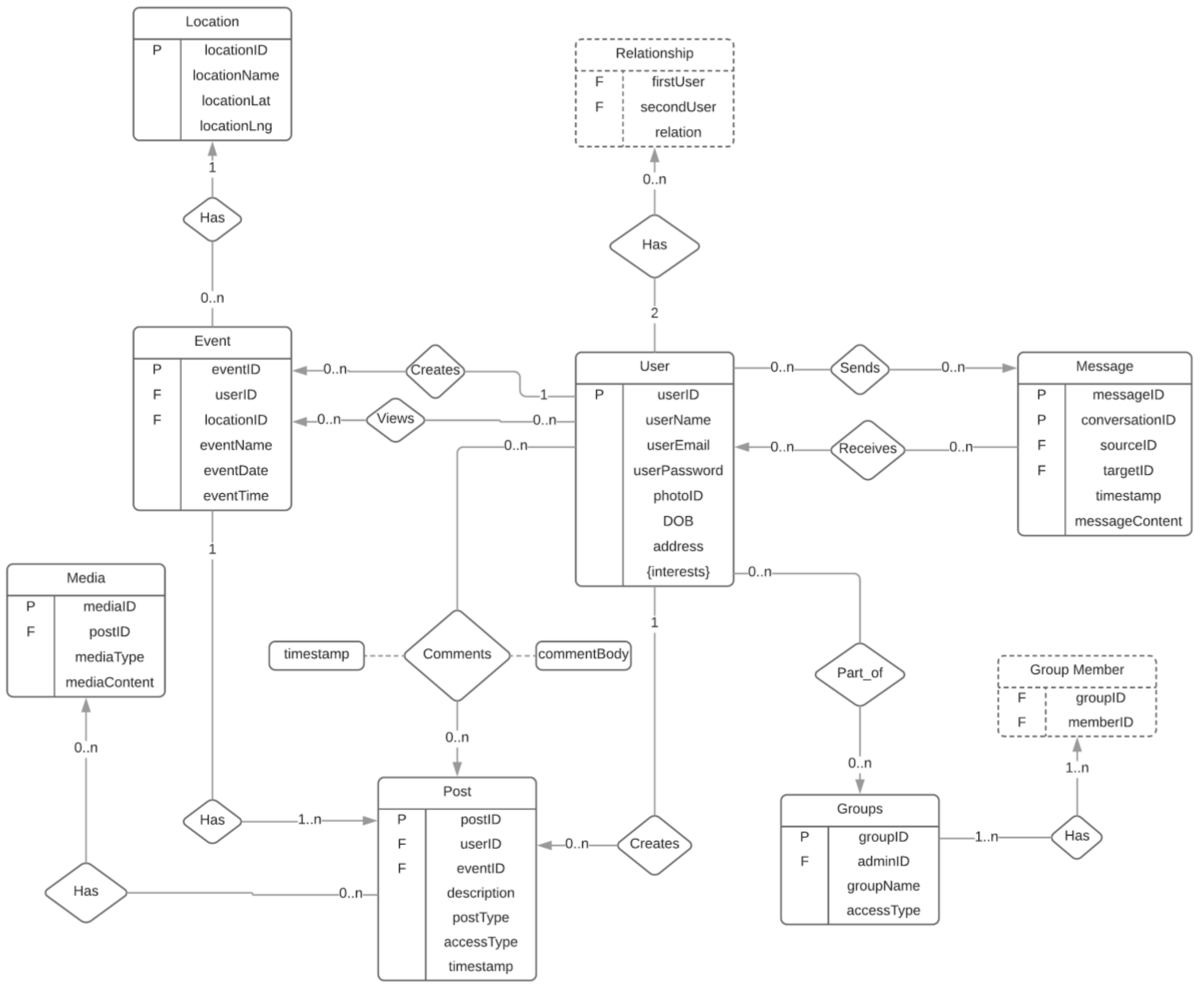
The key features of our system include the following:

- Students can sign up/log in/log out of the network using their email id.
- Students would be required to create a profile which would include entering their photo id, date of birth, interests, etc.
- An event (an event can be a project, collaborative meeting and other activities) can be created by a student which will be posted with various details like event name, location, description, etc.
- Students can also post things (text, photos, videos) which can then be commented upon by other students.
- Students can interact with each other by sending and receiving messages among themselves.
- To improve the interactive experience, students can form groups and do activities/messaging together.
- A student can have various relationships with one or more students. Those relationships can be classified as: friend, follower, none, blocked.
- A student can also accept/decline friend requests from other students

2. Database Design

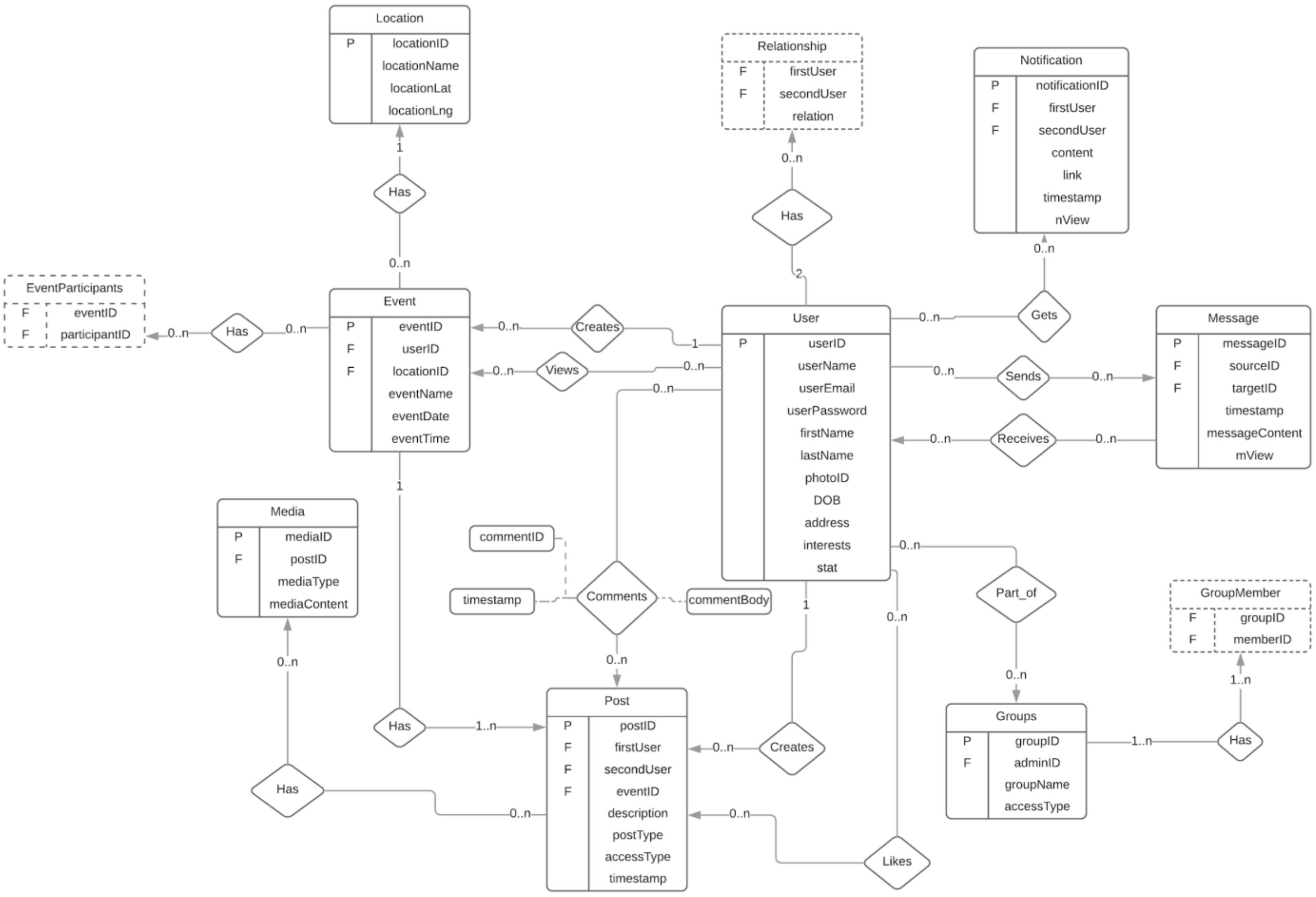
2.1 Entity Relationship Diagram:

This was our Entity relationship diagram before the start of the project.



Now, we have added some more functionality as we started building towards our projects:

We have added three new tables: likes, eventparticipants,and notification. We have also added the columns for the some of the tables. We have added the firstname, lastname and status of the user in the user table.



Note:2.2 Updated Relational Schema:

User(userID,userName,userEmail,userPassword,photoID,DOB,address,interests,firstname,lastname,stat)

Relationship(firstUser, secondUser, relation)

Message(messageID, sourceID, targetID, messageContent, timestamp,mView)

Location(locationID, locationName, locationLat, locationLng)

Comment(commentID, userID, postID, commentBody, timestamp)

Post(postID,firstUser,eventID,desciption, postType, accessType, timestamp**,**secondUser)

Media(mediaID, postID, mediaType, mediaContent)

Event(eventID, userID, eventName, locationID, eventDate, eventTime)

Groups(groupID, groupName, adminID,accessType)

GroupMember(groupID, memberID)

Likes(postID,likes,userID)

Eventparticipants(eventID,participantID)

Notification(notificationID, firstUser,secondUser,content,link,Timestamp,nView)

2.3 Explanation/Design Documentation

Following are the primary and foreign key constraints for the above schema:

Table	**Primary Key(s)**	**Foreign Key(s)**
User	userID	● userID -> Relationship (firstUser, secondUser) ● userID -> Message (sourceID, targetID) ● userID -> Group (adminID) ● userID -> Comment ● userID -> Post ● userID -> Event ● userID -> GroupMember (memberID)
Relationship	N/A	● firstUser -> User (userID) ● secondUser -> User (userID)
Message	conversationID, messageID	● sourceID -> User (userID) ● targetID -> User (userID)
Location	locationID	● locationID -> Event
Comment	commentID	● userID -> User ● postID -> Post
Post	postID	● postID -> Comment ● postID -> Media ● userID -> User ● eventID -> Event
Media	mediaID	● postID -> Post
Event	eventID	● locationID -> Location ● userID -> User ● eventID -> Post
Groups	groupID	● adminID -> User (userID) ● groupID -> GroupMember
GroupMember	N/A	● groupID -> Groups ● memberID -> User (userID)
Likes	N/A	● postID -> Post ● userID -> User
Eventparticipants	N/A	● eventID -> Event ● participantID -> User
Notification	notificationID	● firstUser -> User ● secondUser -> User

1. Abbreviations used in the above ER Diagram:-
    a. P : Primary Key
    b. F : Foreign Key
2. *Relationship*,*Group Member,Likes and Eventparticipants* are weak entities. The strong entity related to *Relationship* is *User* and *Group Member* is *Groups*.

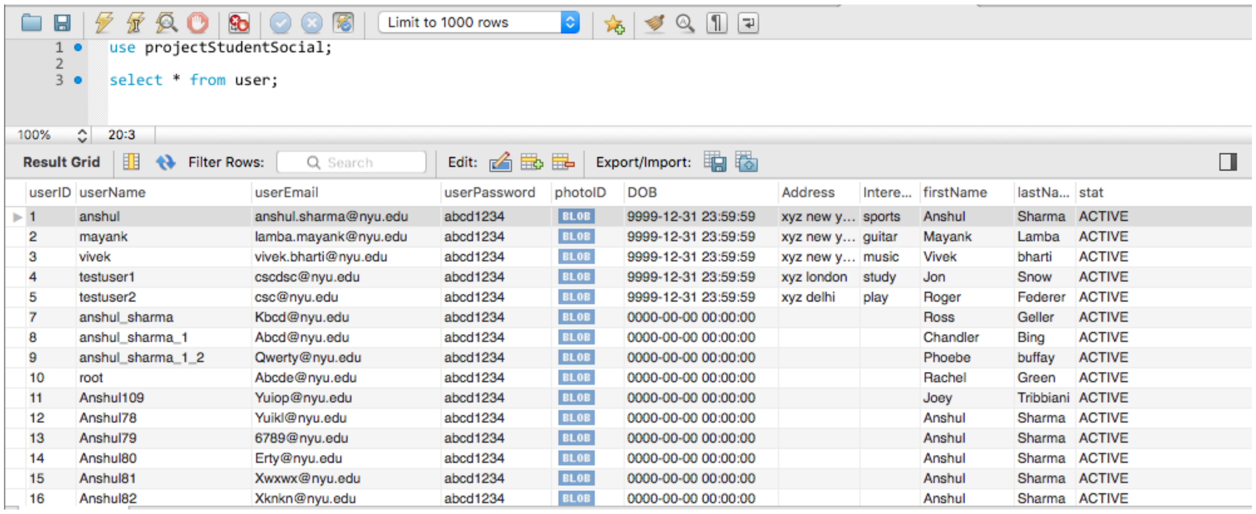
Following are the key points to be observed about our database design:

- The central entities in our database are: User(students), Post, Event, Group and Message.
- A User can do the following things:
  - Signup/Login/Logout
  - Can create a post on his/her profile or as part of an event
  - Search other User.
  - Update profile data.
  - Comment on posts
  - Like/ Unlike the posts.
  - View their own/someone else's profile
  - Create Event(s)
  - View Event(s)
  - Join Event(S)
  - Create Group(s)
  - View Groups(s)
  - Join Group(s)
  - Send/receive messages among friends with Conversation.
  - Get Location
  - Have relationships with other User(friend, follower, blocked, none)
- A Relationship(which is essentially a weak entity) is among two users.
- Messages can either be sent or received among users(referred to as sourceID, targetID respectively in the relational schema). They will all have a timestamp.
- A Post can:
  - Be created by users as a part of an event or even individually
  - Have media attached to them(photos, video)
  - Be commented upon by user(s)
  - Have an access type: private, public, friends or friends of friends(fof)
  - Will have a timestamp
- An Event:
  - Has at least one post related to it
  - Can have a location
  - Is created by a user
  - Can be viewed by many users depending upon the access type of event post
- A Location is represented by a location ID. It can have a name, and coordinates (latitude and longitude)
- All Media are related to at least one post. Its type can be either a photo or a video.
- A Comment by a user has its own timestamp, content and is recognized by comment ID. A comment can only be done on a post.
- Groups can:
  - Be created by a user which will become its administrator but is uniquely identified by a group ID.
  - Have multiple members entered into groupmember table.

- Have access type: private, public, friends, friends of friends

3. Database Tables:

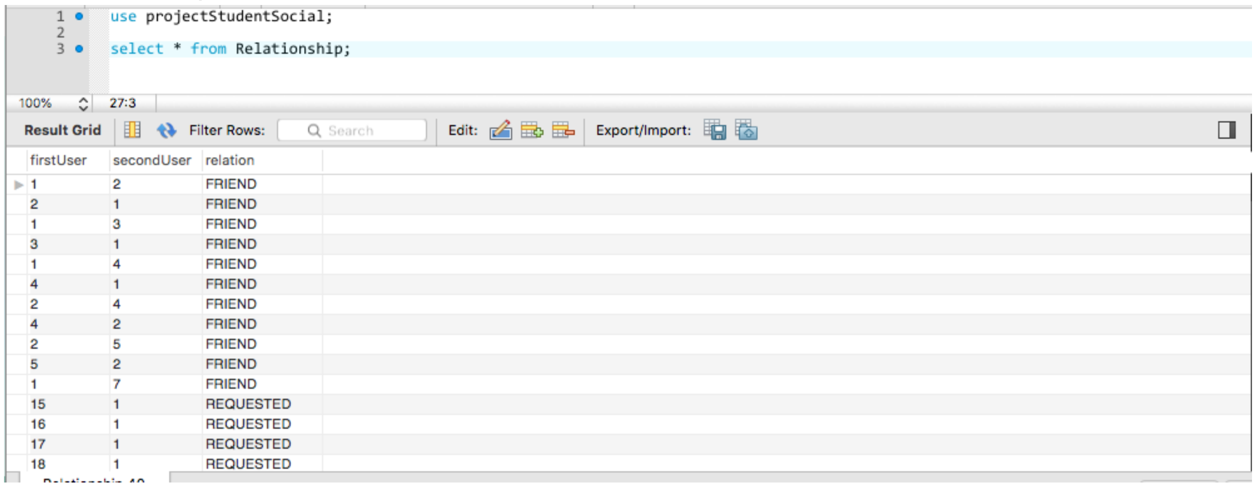
3.1 User Table:



```
use projectStudentSocial;

select * from user;
```

userID	userName	userEmail	userPassword	photoID	DOB	Address	Intere...	firstName	lastNa...	stat
1	anshul	anshul.sharma@nyu.edu	abcd1234	BLOB	9999-12-31 23:59:59	xyz new y...	sports	Anshul	Sharma	ACTIVE
2	mayank	lamba.mayank@nyu.edu	abcd1234	BLOB	9999-12-31 23:59:59	xyz new y...	guitar	Mayank	Lamba	ACTIVE
3	vivek	vivek.bharti@nyu.edu	abcd1234	BLOB	9999-12-31 23:59:59	xyz new y...	music	Vivek	bharti	ACTIVE
4	testuser1	cscdsc@nyu.edu	abcd1234	BLOB	9999-12-31 23:59:59	xyz london	study	Jon	Snow	ACTIVE
5	testuser2	csc@nyu.edu	abcd1234	BLOB	9999-12-31 23:59:59	xyz delhi	play	Roger	Federer	ACTIVE
7	anshul_sharma	Kbcd@nyu.edu	abcd1234	BLOB	0000-00-00 00:00:00			Ross	Geller	ACTIVE
8	anshul_sharma_1	Abcd@nyu.edu	abcd1234	BLOB	0000-00-00 00:00:00			Chandler	Bing	ACTIVE
9	anshul_sharma_1_2	Qwerty@nyu.edu	abcd1234	BLOB	0000-00-00 00:00:00			Phoebe	buffay	ACTIVE
10	root	Abcde@nyu.edu	abcd1234	BLOB	0000-00-00 00:00:00			Rachel	Green	ACTIVE
11	Anshul109	Yuiop@nyu.edu	abcd1234	BLOB	0000-00-00 00:00:00			Joey	Tribbiani	ACTIVE
12	Anshul78	Yuikl@nyu.edu	abcd1234	BLOB	0000-00-00 00:00:00			Anshul	Sharma	ACTIVE
13	Anshul79	6789@nyu.edu	abcd1234	BLOB	0000-00-00 00:00:00			Anshul	Sharma	ACTIVE
14	Anshul80	Erty@nyu.edu	abcd1234	BLOB	0000-00-00 00:00:00			Anshul	Sharma	ACTIVE
15	Anshul81	Xwxwx@nyu.edu	abcd1234	BLOB	0000-00-00 00:00:00			Anshul	Sharma	ACTIVE
16	Anshul82	Xknkn@nyu.edu	abcd1234	BLOB	0000-00-00 00:00:00			Anshul	Sharma	ACTIVE

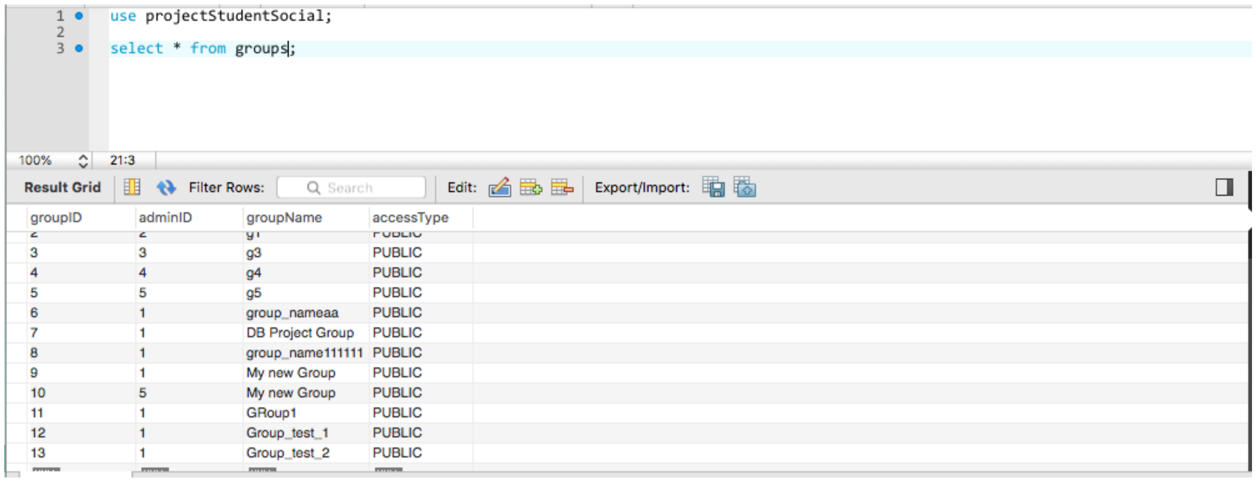
3.2 Relationship table:



```
use projectStudentSocial;

select * from Relationship;
```

firstUser	secondUser	relation
1	2	FRIEND
2	1	FRIEND
1	3	FRIEND
3	1	FRIEND
1	4	FRIEND
4	1	FRIEND
2	4	FRIEND
4	2	FRIEND
2	5	FRIEND
5	2	FRIEND
1	7	FRIEND
15	1	REQUESTED
16	1	REQUESTED
17	1	REQUESTED
18	1	REQUESTED

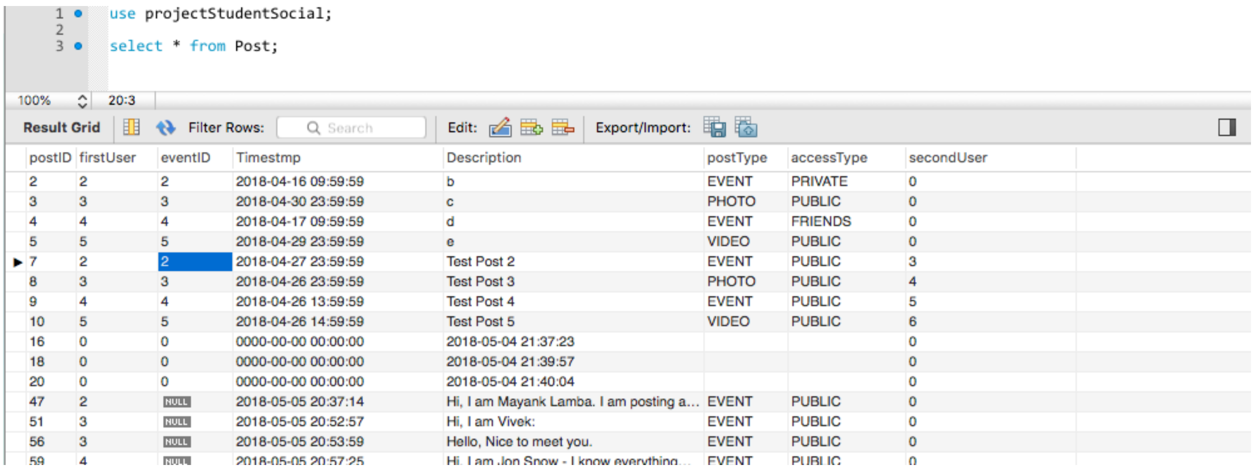
3.3 Groups Table:



```sql
use projectStudentSocial;

select * from groups;
```

groupID	adminID	groupName	accessType
2	2	g1	PUBLIC
3	3	g3	PUBLIC
4	4	g4	PUBLIC
5	5	g5	PUBLIC
6	1	group_nameaa	PUBLIC
7	1	DB Project Group	PUBLIC
8	1	group_name111111	PUBLIC
9	1	My new Group	PUBLIC
10	5	My new Group	PUBLIC
11	1	GRoup1	PUBLIC
12	1	Group_test_1	PUBLIC
13	1	Group_test_2	PUBLIC

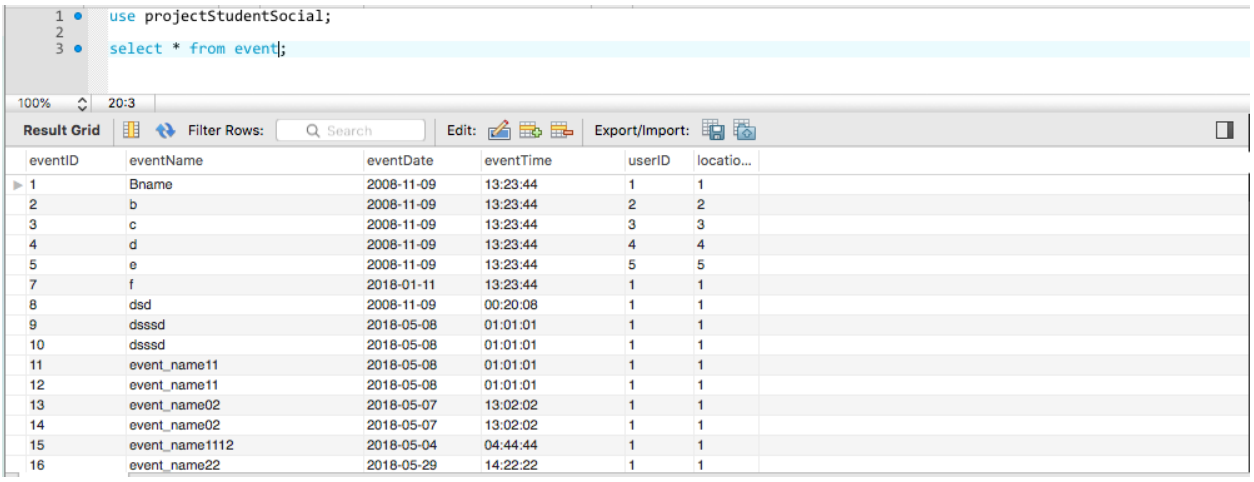
3.4 Post Table:



```sql
use projectStudentSocial;

select * from Post;
```

postID	firstUser	eventID	Timestmp	Description	postType	accessType	secondUser
2	2	2	2018-04-16 09:59:59	b	EVENT	PRIVATE	0
3	3	3	2018-04-30 23:59:59	c	PHOTO	PUBLIC	0
4	4	4	2018-04-17 09:59:59	d	EVENT	FRIENDS	0
5	5	5	2018-04-29 23:59:59	e	VIDEO	PUBLIC	0
7	2	2	2018-04-27 23:59:59	Test Post 2	EVENT	PUBLIC	3
8	3	3	2018-04-26 23:59:59	Test Post 3	PHOTO	PUBLIC	4
9	4	4	2018-04-26 13:59:59	Test Post 4	EVENT	PUBLIC	5
10	5	5	2018-04-26 14:59:59	Test Post 5	VIDEO	PUBLIC	6
16	0	0	0000-00-00 00:00:00	2018-05-04 21:37:23			0
18	0	0	0000-00-00 00:00:00	2018-05-04 21:39:57			0
20	0	0	0000-00-00 00:00:00	2018-05-04 21:40:04			0
47	2	NULL	2018-05-05 20:37:14	Hi, I am Mayank Lamba. I am posting a...	EVENT	PUBLIC	0
51	3	NULL	2018-05-05 20:52:57	Hi, I am Vivek:	EVENT	PUBLIC	0
56	3	NULL	2018-05-05 20:53:59	Hello, Nice to meet you.	EVENT	PUBLIC	0
59	4	NULL	2018-05-05 20:57:25	Hi, I am Jon Snow - I know everything...	EVENT	PUBLIC	0

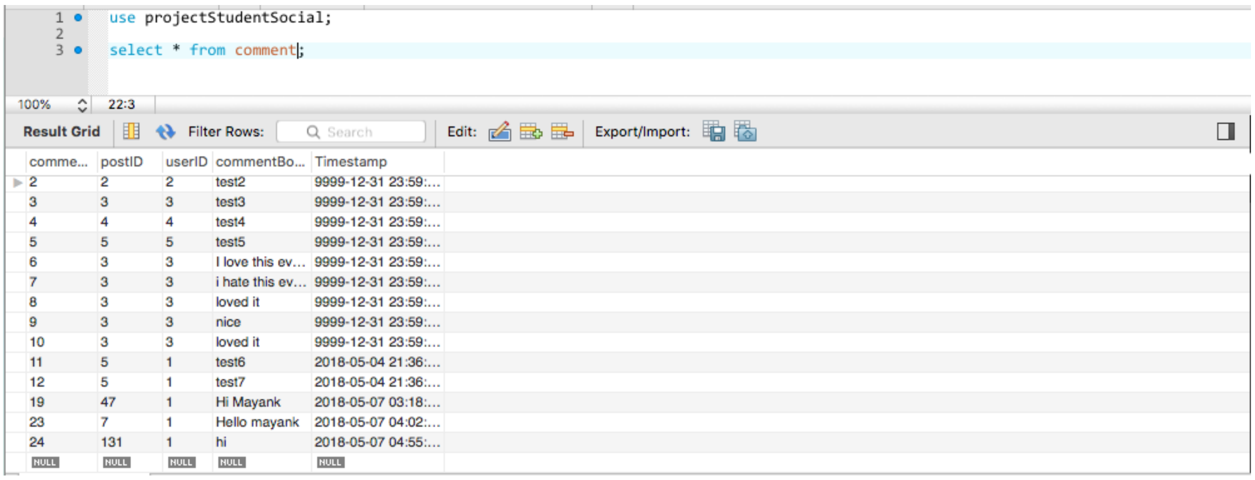
3.5 Event Table:



```sql
use projectStudentSocial;

select * from event;
```

eventID	eventName	eventDate	eventTime	userID	locatio...
1	Bname	2008-11-09	13:23:44	1	1
2	b	2008-11-09	13:23:44	2	2
3	c	2008-11-09	13:23:44	3	3
4	d	2008-11-09	13:23:44	4	4
5	e	2008-11-09	13:23:44	5	5
7	f	2018-01-11	13:23:44	1	1
8	dsd	2008-11-09	00:20:08	1	1
9	dsssd	2018-05-08	01:01:01	1	1
10	dsssd	2018-05-08	01:01:01	1	1
11	event_name11	2018-05-08	01:01:01	1	1
12	event_name11	2018-05-08	01:01:01	1	1
13	event_name02	2018-05-07	13:02:02	1	1
14	event_name02	2018-05-07	13:02:02	1	1
15	event_name1112	2018-05-04	04:44:44	1	1
16	event_name22	2018-05-29	14:22:22	1	1

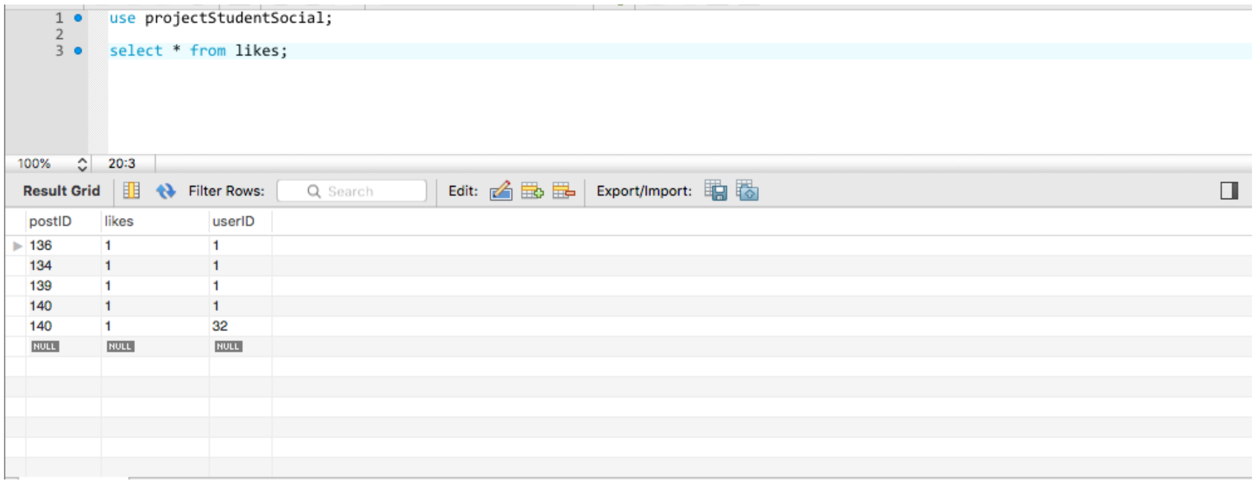
3.6 Comment table:



```
use projectStudentSocial;

select * from comment;
```

comme...	postID	userID	commentBo...	Timestamp
2	2	2	test2	9999-12-31 23:59:...
3	3	3	test3	9999-12-31 23:59:...
4	4	4	test4	9999-12-31 23:59:...
5	5	5	test5	9999-12-31 23:59:...
6	3	3	I love this ev...	9999-12-31 23:59:...
7	3	3	i hate this ev...	9999-12-31 23:59:...
8	3	3	loved it	9999-12-31 23:59:...
9	3	3	nice	9999-12-31 23:59:...
10	3	3	loved it	9999-12-31 23:59:...
11	5	1	test6	2018-05-04 21:36:...
12	5	1	test7	2018-05-04 21:36:...
19	47	1	Hi Mayank	2018-05-07 03:18:...
23	7	1	Hello mayank	2018-05-07 04:02:...
24	131	1	hi	2018-05-07 04:55:...
NULL	NULL	NULL	NULL	NULL

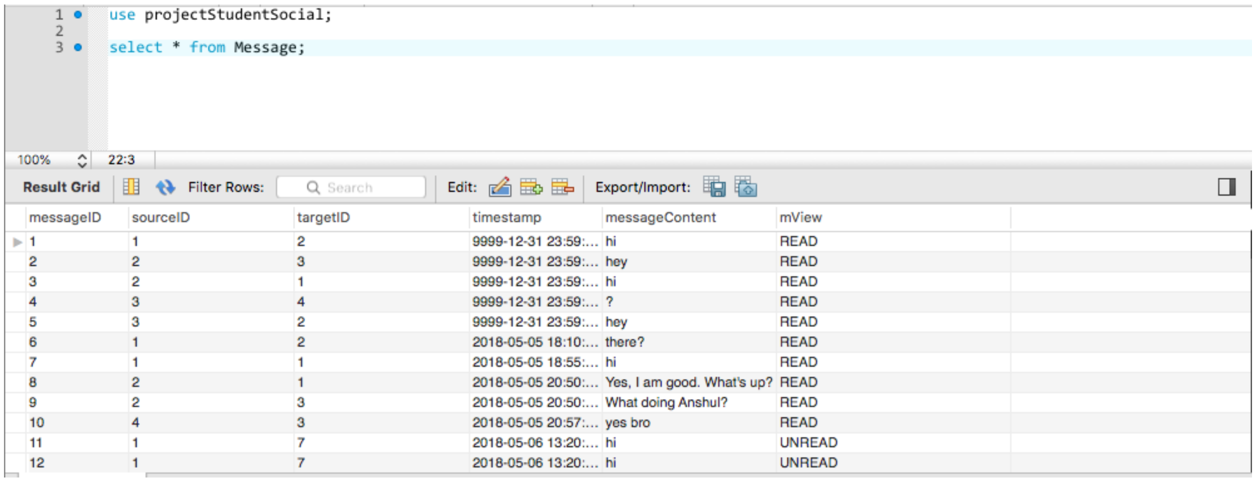
3.7 Likes Table:



```
use projectStudentSocial;

select * from likes;
```

postID	likes	userID
136	1	1
134	1	1
139	1	1
140	1	1
140	1	32
NULL	NULL	NULL

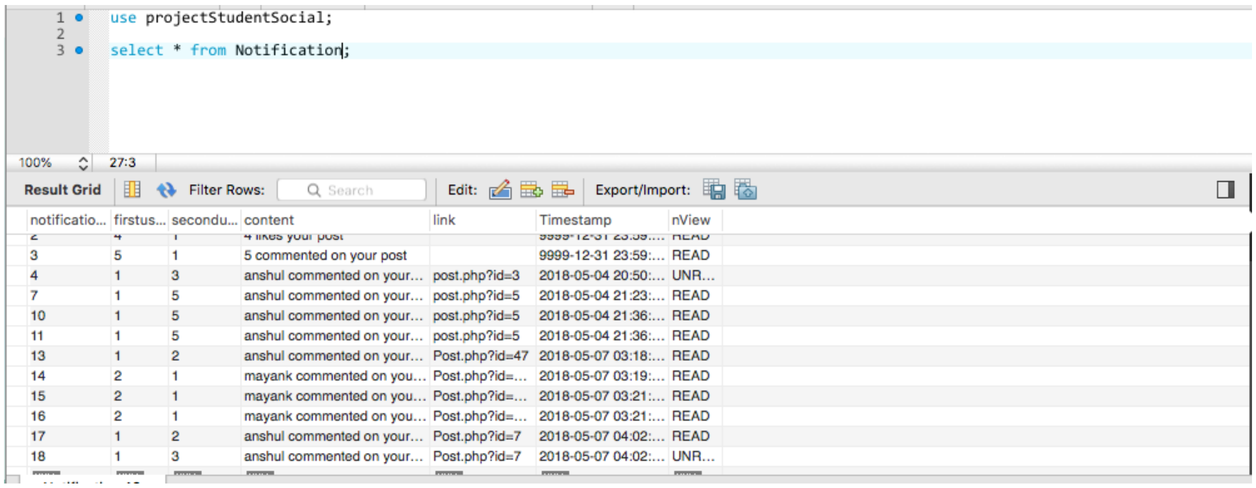
3.8 Message Table:



```
use projectStudentSocial;

select * from Message;
```

messageID	sourceID	targetID	timestamp	messageContent	mView
1	1	2	9999-12-31 23:59:...	hi	READ
2	2	3	9999-12-31 23:59:...	hey	READ
3	2	1	9999-12-31 23:59:...	hi	READ
4	3	4	9999-12-31 23:59:...	?	READ
5	3	2	9999-12-31 23:59:...	hey	READ
6	1	2	2018-05-05 18:10:...	there?	READ
7	1	1	2018-05-05 18:55:...	hi	READ
8	2	1	2018-05-05 20:50:...	Yes, I am good. What's up?	READ
9	2	3	2018-05-05 20:50:...	What doing Anshul?	READ
10	4	3	2018-05-05 20:57:...	yes bro	READ
11	1	7	2018-05-06 13:20:...	hi	UNREAD
12	1	7	2018-05-06 13:20:...	hi	UNREAD

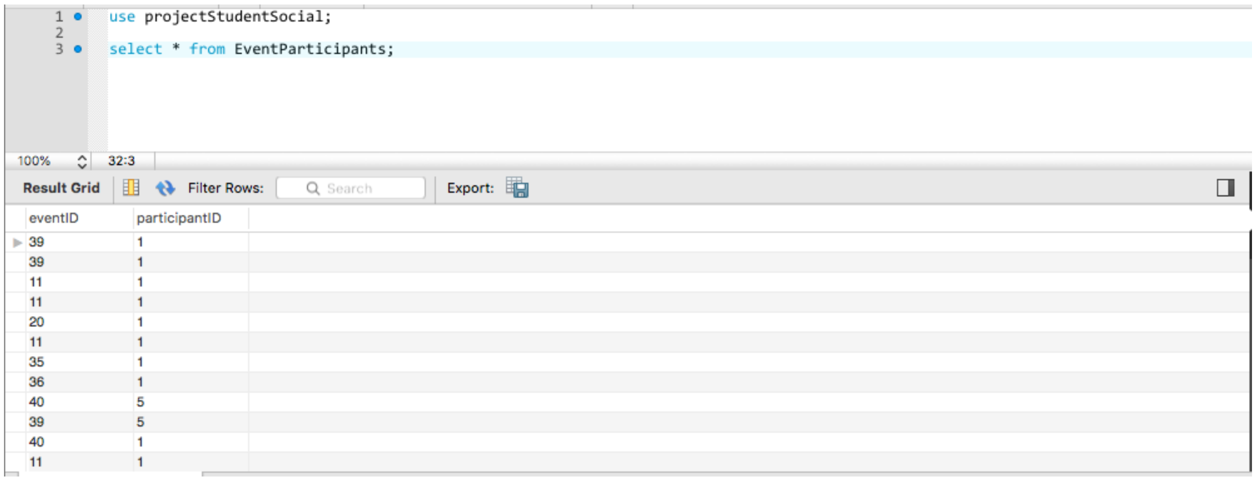
3.9 Notification table:



```
use projectStudentSocial;

select * from Notification;
```

notificatio...	firstus...	secondu...	content	link	Timestamp	nView
2	4	1	4 likes your post		9999-12-31 23:59:...	READ
3	5	1	5 commented on your post		9999-12-31 23:59:...	READ
4	1	3	anshul commented on your...	post.php?id=3	2018-05-04 20:50:...	UNR...
7	1	5	anshul commented on your...	post.php?id=5	2018-05-04 21:23:...	READ
10	1	5	anshul commented on your...	post.php?id=5	2018-05-04 21:36:...	READ
11	1	5	anshul commented on your...	post.php?id=5	2018-05-04 21:36:...	READ
13	1	2	anshul commented on your...	Post.php?id=47	2018-05-07 03:18:...	READ
14	2	1	mayank commented on you...	Post.php?id=...	2018-05-07 03:19:...	READ
15	2	1	mayank commented on you...	Post.php?id=...	2018-05-07 03:21:...	READ
16	2	1	mayank commented on you...	Post.php?id=...	2018-05-07 03:21:...	READ
17	1	2	anshul commented on your...	Post.php?id=7	2018-05-07 04:02:...	READ
18	1	3	anshul commented on your...	Post.php?id=7	2018-05-07 04:02:...	UNR...

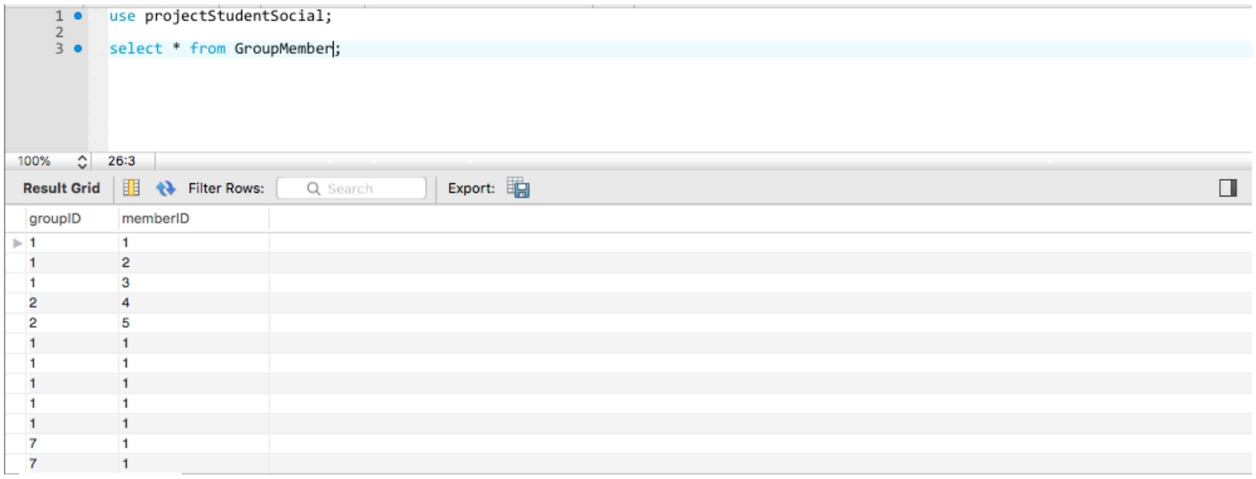
3.10 EventParticipants table:



```
use projectStudentSocial;

select * from EventParticipants;
```

eventID	participantID
39	1
39	1
11	1
11	1
20	1
11	1
35	1
36	1
40	5
39	5
40	1
11	1

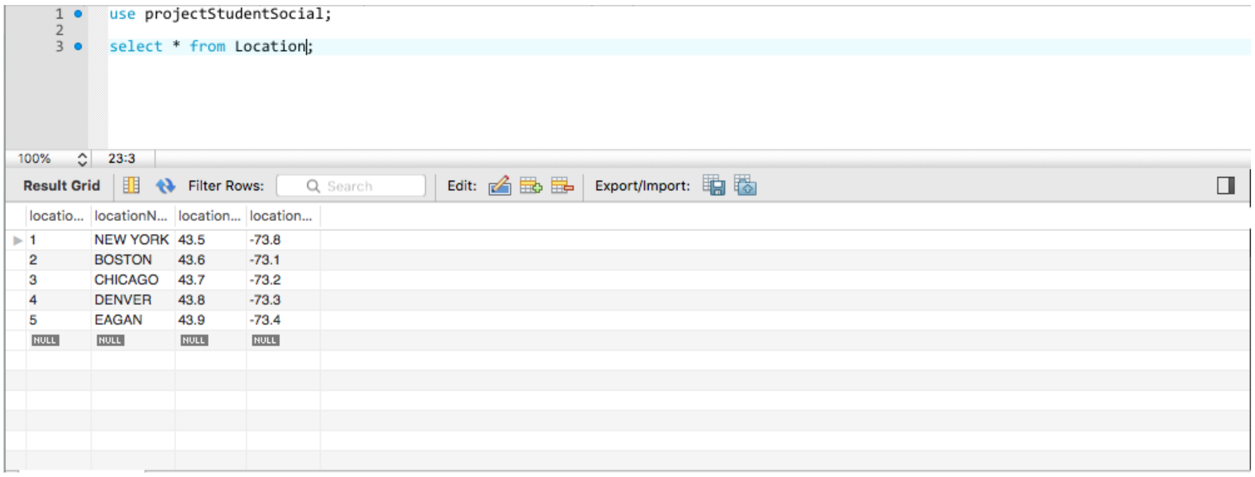
3.11 GroupMember table:



```
use projectStudentSocial;

select * from GroupMember;
```

groupID	memberID
1	1
1	2
1	3
2	4
2	5
1	1
1	1
1	1
1	1
1	1
7	1
7	1

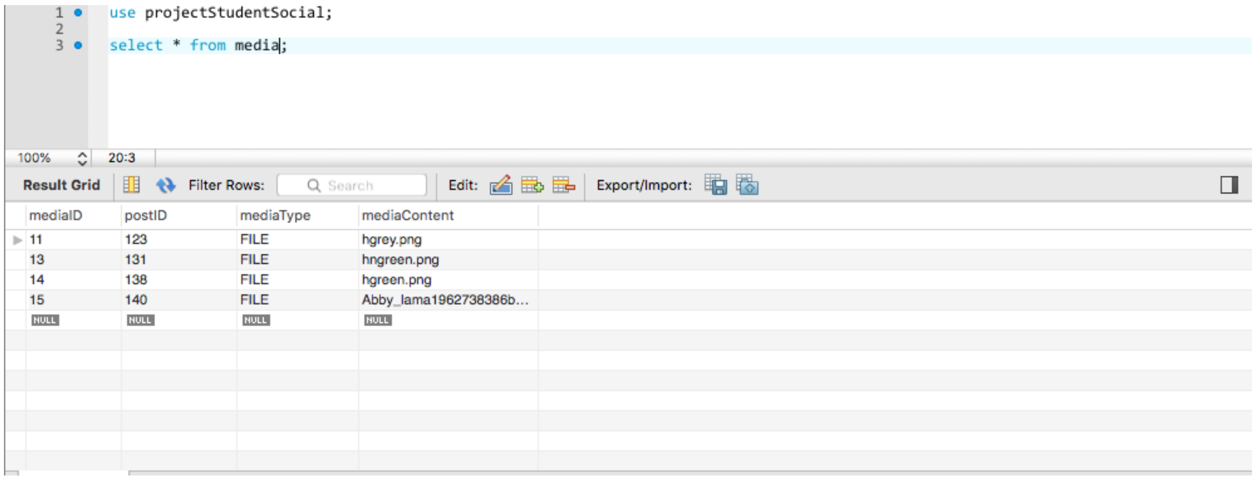
3.12 Location table:



```
use projectStudentSocial;

select * from Location;
```

locatio...	locationN...	location...	location...
1	NEW YORK	43.5	-73.8
2	BOSTON	43.6	-73.1
3	CHICAGO	43.7	-73.2
4	DENVER	43.8	-73.3
5	EAGAN	43.9	-73.4
NULL	NULL	NULL	NULL

3.13 Media Table:



```
use projectStudentSocial;

select * from media;
```

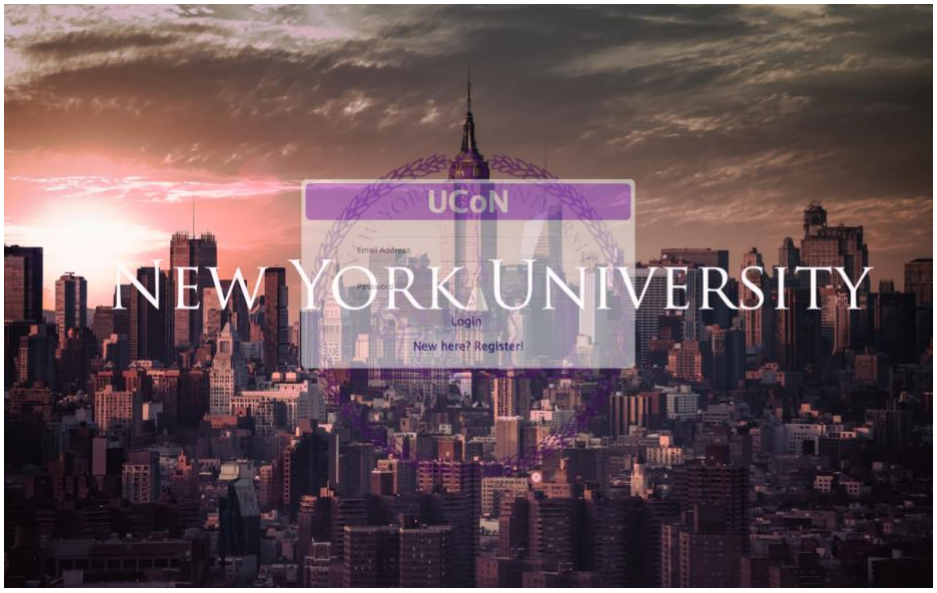
mediaID	postID	mediaType	mediaContent
11	123	FILE	hgrey.png
13	131	FILE	hngreen.png
14	138	FILE	hgreen.png
15	140	FILE	Abby_lama1962738386b...
NULL	NULL	NULL	NULL

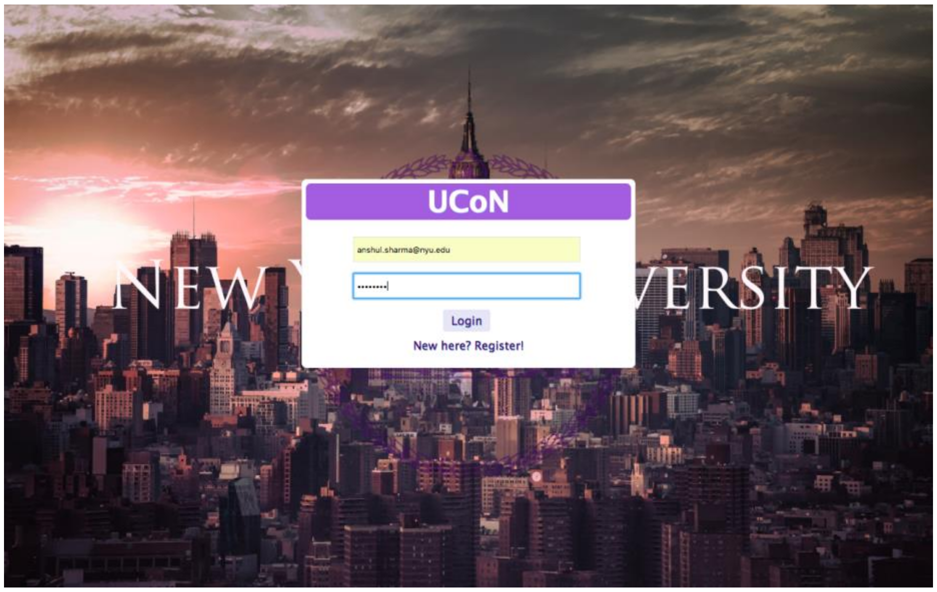
4. Implementation:

With the database created, we now have to implement the frontend for our university social network site. We have built our code using PHP, bootstrap, Ajax, Javascript. It contains all the basic features required by any social networking site. The users will be able to login to the website, where we are storing their data into session and using logged in user's id to fetch all the details and update the tables.

Login Page:

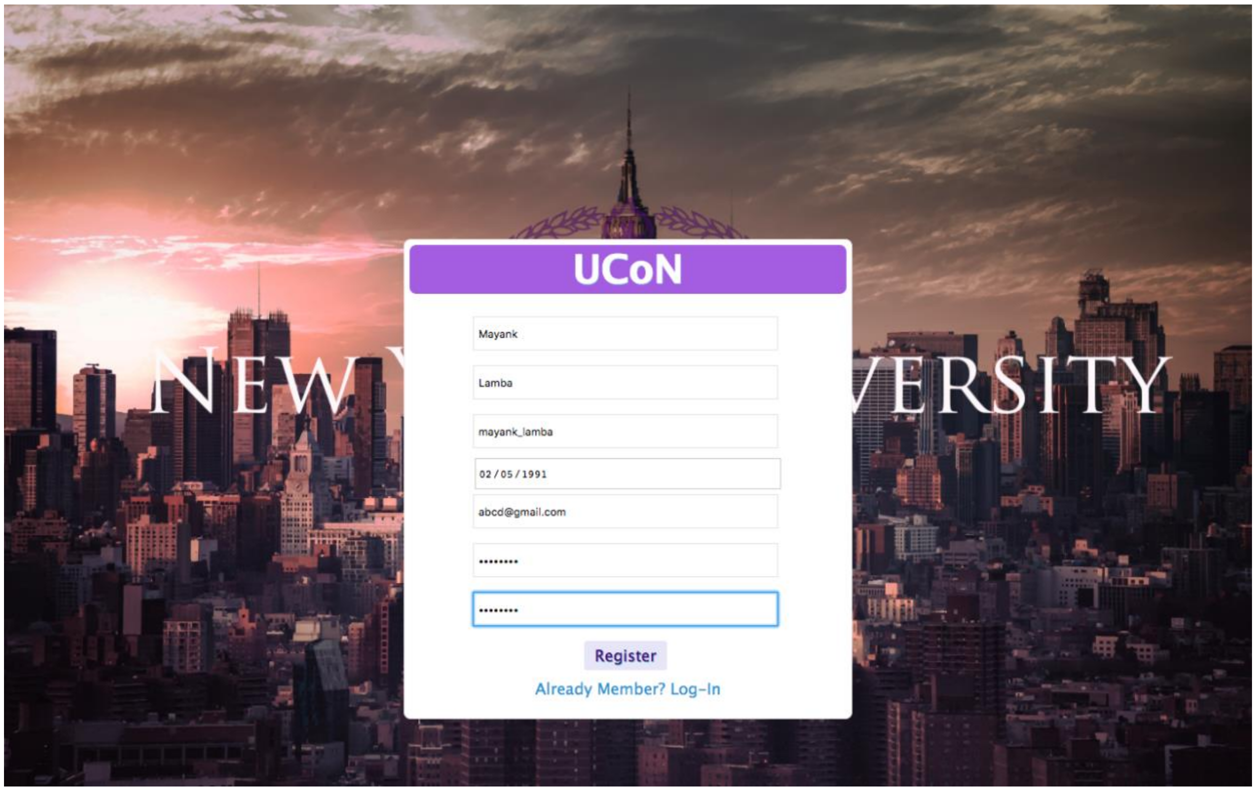
The first page displays the login window with blurred login box, which will appear properly when hovered upon. The Login page requires user to enter email ID and password to login to the system. After entering the data and clicking on login button, the user credentials will be checked and if there is match in our user table, session with user ID is set and user is logged in.





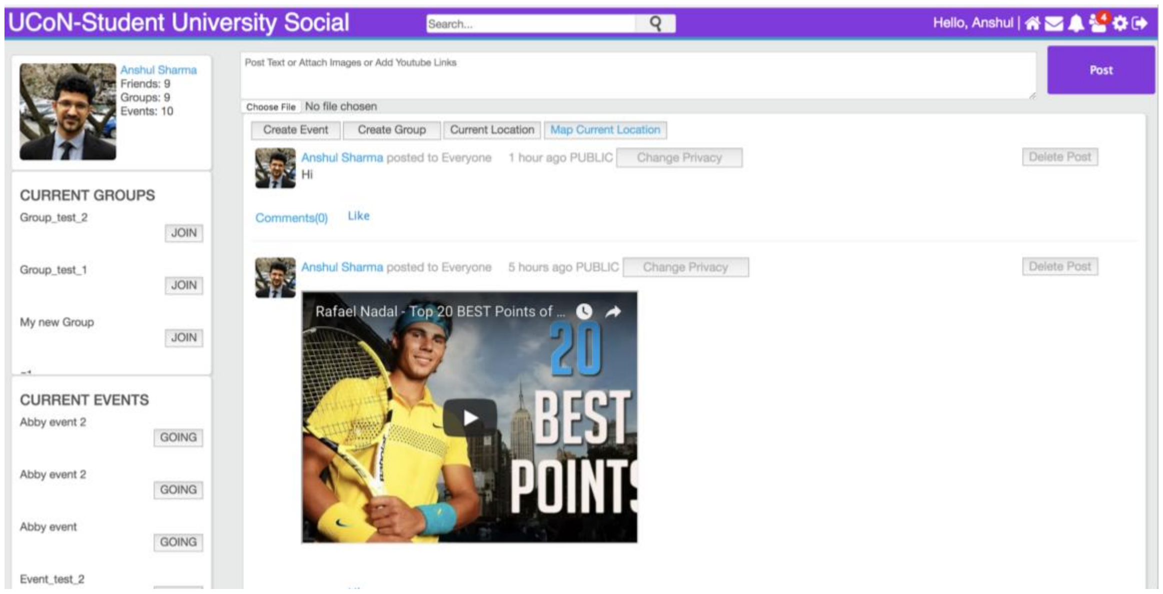
Registration Page:

We have implemented a fade in fade out model box, where user can easily switch between login and sign up page. The sign up page appears after the user clicked on Register button. It asks for basic details such as first name, last name, username, email id, date of birth and password. Before creating a user we are checking if the user email id is valid or not and if both the passwords match or not. If passwords do not match, a message appears regarding the invalid credentials. Also it checks whether the entered username and email is not already present in the table as they are both unique keys.



Mayank
Lamba
mayank_lamba
02 / 05 / 1991
abcd@gmail.com
••••••••
••••••••

Home Page/Timeline:
Our home page shows a variety of things. On the left side of the page, we have the user details including profile picture, total number of friends, total events participated and total number of groups joined. Below that, there are lists of events and groups which shows only those which the current have joined or taken part in. User can click on the join and going button which is written in front of each event and group. The selected group or the event will be automatically removed from the list as the user has joined it.
On the top bar, user can search for other users and can click on their profile to reach their profile page. On the right side, there are profile page button, home home button, message notifications, post notifications, friend request notification, update details button and log out button.
In the middle part, user can post using text and by also attaching an image to the post. The data here is linked between the media and the post table. The post will appear on the user wall and the home page as well with the timestamp when it was created. The post will display along the whole wall and are filtered by queried so that either public post are visible or the post by friend and self are visible. User can also delete the post through the post wall, but they can only delete the posted by them.
We also have three extra buttons on the top through which user can create the event, group or also know the current location of the user.

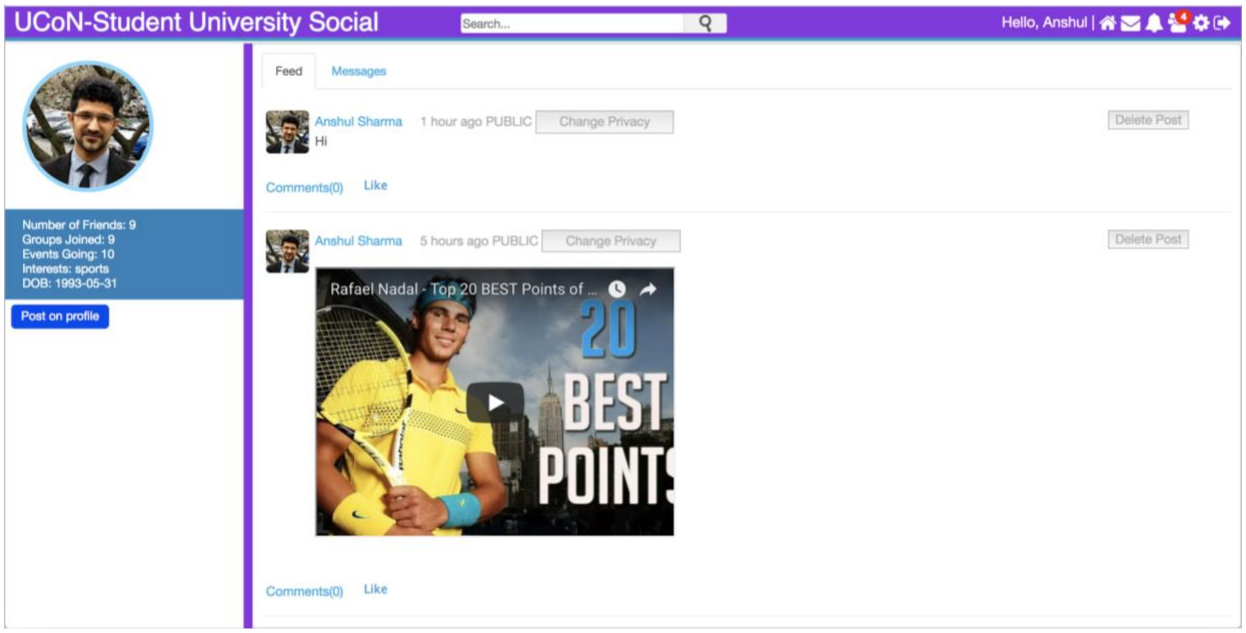


Profile Page:

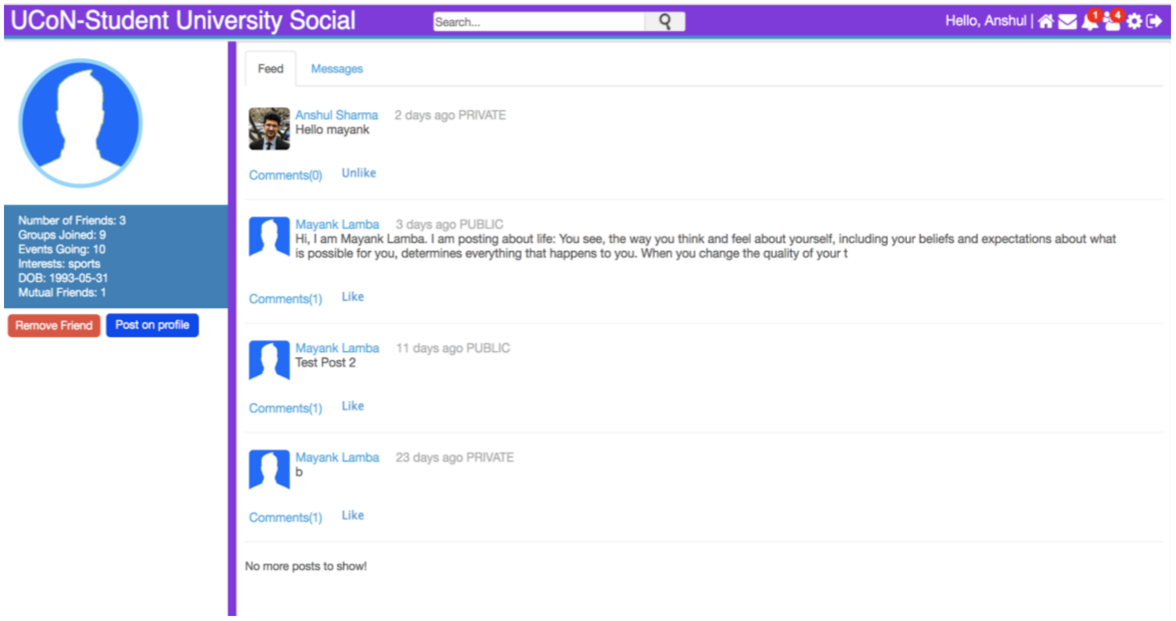
The profile page view is same as the home page view. On the left side, it will show the details of the user including friends,groups,events, date of birth, mutual friends and interests. On the center, it shows the post that is either posted on their profile or posted by them. They can also message the user by switching to the message box.

If the current user navigates to different user profile, there will be different options depending upon the the relationship between two users. If the both are currently friend (Identified by isFriend function) then there will be 'remove friend option'. If they are not friends 'Add friend option will be there '. If the request has already been sent or received, then it will either say to respond to request or request sent.

1. Profile of the user who is currently logged in



2. Profile of the user other than who is currently logged in:



UCoN-Student University Social

Search...

Hello, Anshul |

Feed | Messages

Anshul Sharma 2 days ago PRIVATE
Hello mayank

Comments(0) Unlike

Mayank Lamba 3 days ago PUBLIC
Hi, I am Mayank Lamba. I am posting about life: You see, the way you think and feel about yourself, including your beliefs and expectations about what is possible for you, determines everything that happens to you. When you change the quality of your t

Comments(1) Like

Mayank Lamba 11 days ago PUBLIC
Test Post 2

Comments(1) Like

Mayank Lamba 23 days ago PRIVATE
b

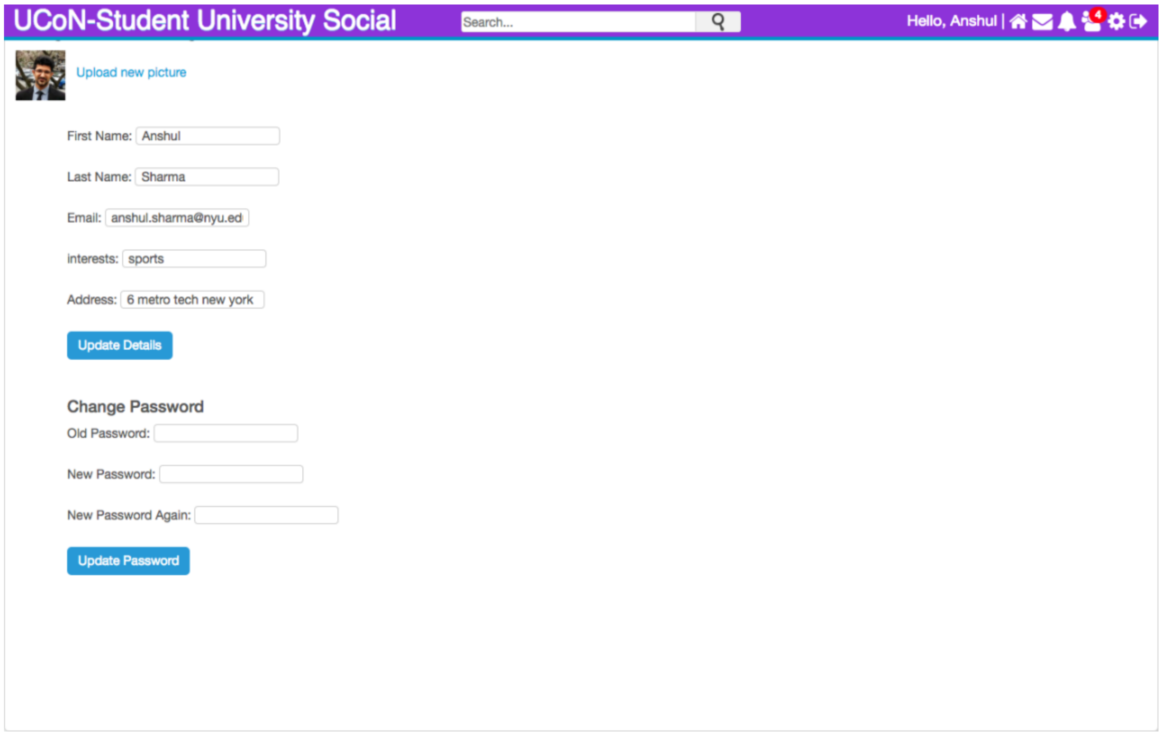
Comments(1) Like

No more posts to show!

Number of Friends: 3
Groups Joined: 9
Events Going: 10
Interests: sports
DOB: 1993-05-31
Mutual Friends: 1

Remove Friend | Post on profile

3. Update Details Page:
   We have also given an option for user to update their details and if they want to add their interest and their address. This will run a update query in the user table. Users can also update their profile picture if they want to.



UCoN-Student University Social

Search...

Hello, Anshul |

Upload new picture

First Name: Anshul

Last Name: Sharma

Email: anshul.sharma@nyu.ed

interests: sports

Address: 6 metro tech new york

Update Details

Change Password

Old Password:

New Password:

New Password Again:

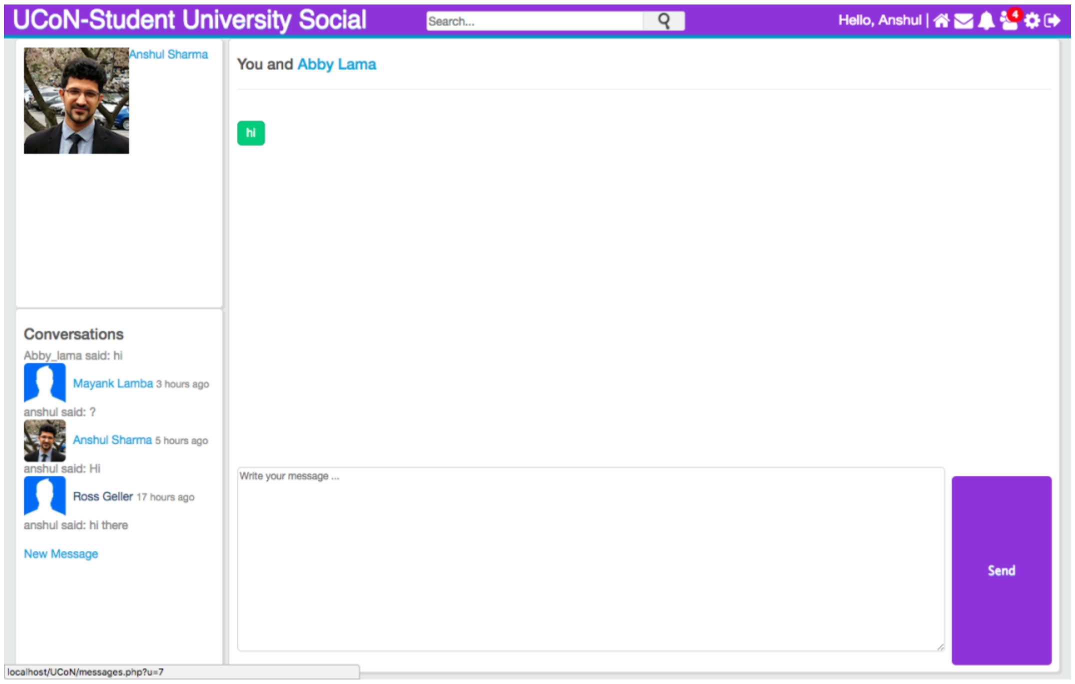
Update Password

Basic Features:

We have implemented some basic features as required by any social networking site and they are shown as follows:

1. Message Window:

   We can reach the message window by clicking on the messages box and also see the conversation between two users through green and blue colors.



UCoN-Student University Social

Hello, Anshul |

Anshul Sharma

You and Abby Lama

hi

Conversations

Abby_lama said: hi

Mayank Lamba 3 hours ago

anshul said: ?

Anshul Sharma 5 hours ago

anshul said: Hi

Ross Geller 17 hours ago

anshul said: hi there

New Message

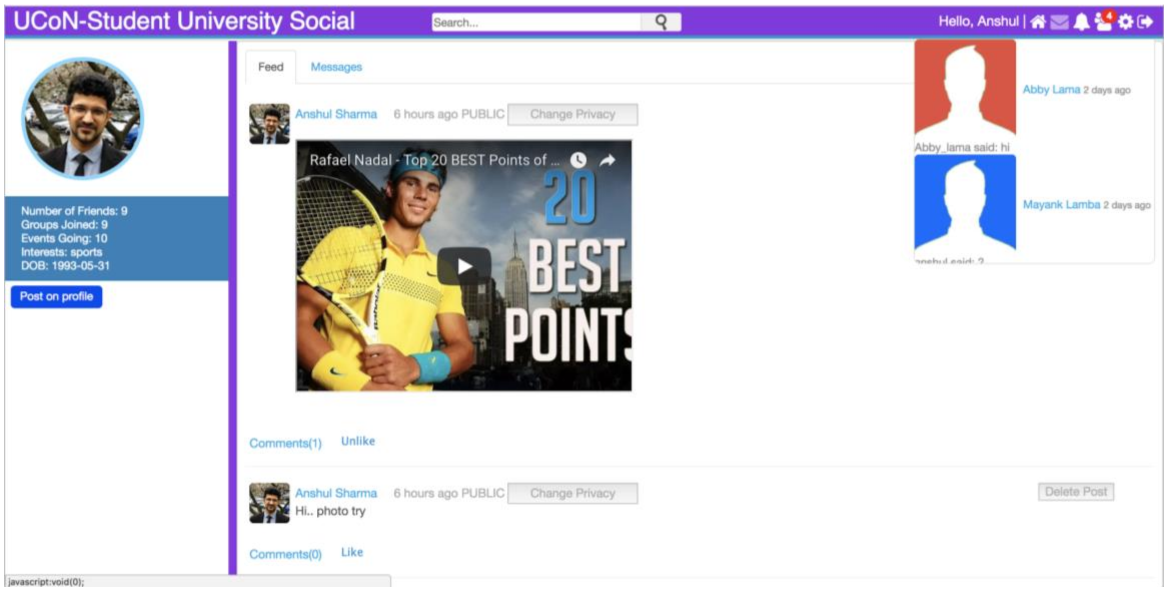
Write your message ...

Send

localhost/UCoN/messages.php?u=7

2. Message Notification:

   Message notification shows the message sent or received sorted by the latest in order.



UCoN-Student University Social

Hello, Anshul |

Number of Friends: 9
Groups Joined: 9
Events Going: 10
Interests: sports
DOB: 1993-05-31

Post on profile

Feed Messages

Anshul Sharma 6 hours ago PUBLIC Change Privacy

Rafael Nadal - Top 20 BEST Points of ...

Comments(1) Unlike

Anshul Sharma 6 hours ago PUBLIC Change Privacy Delete Post

Hi.. photo try

Comments(0) Like

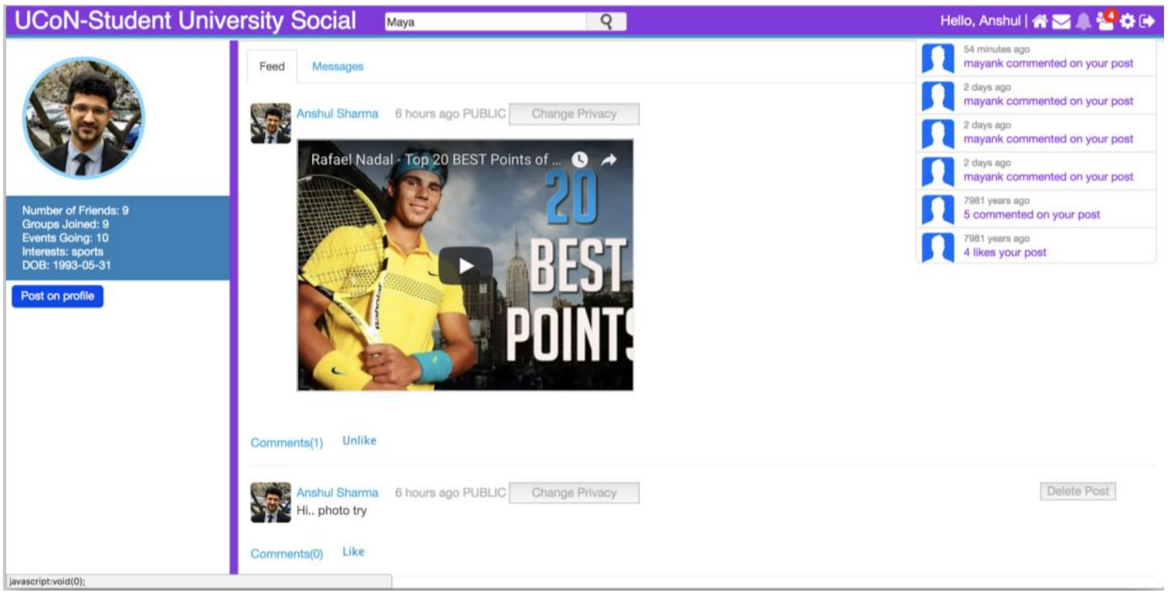
Abby Lama 2 days ago

Abby_lama said: hi

Mayank Lamba 2 days ago

javascript:void(0);

3. Post Notifications:
   Post notifications are also available on the option if the user commented or like the current user's post.


UCoN-Student University Social

Maya

Hello, Anshul |

54 minutes ago
mayank commented on your post

2 days ago
mayank commented on your post

2 days ago
mayank commented on your post

2 days ago
mayank commented on your post

7981 years ago
5 commented on your post

7981 years ago
4 likes your post

Number of Friends: 9
Groups Joined: 9
Events Going: 10
Interests: sports
DOB: 1993-05-31

Post on profile

Feed Messages

Anshul Sharma 6 hours ago PUBLIC Change Privacy

Rafael Nadal - Top 20 BEST Points of ...

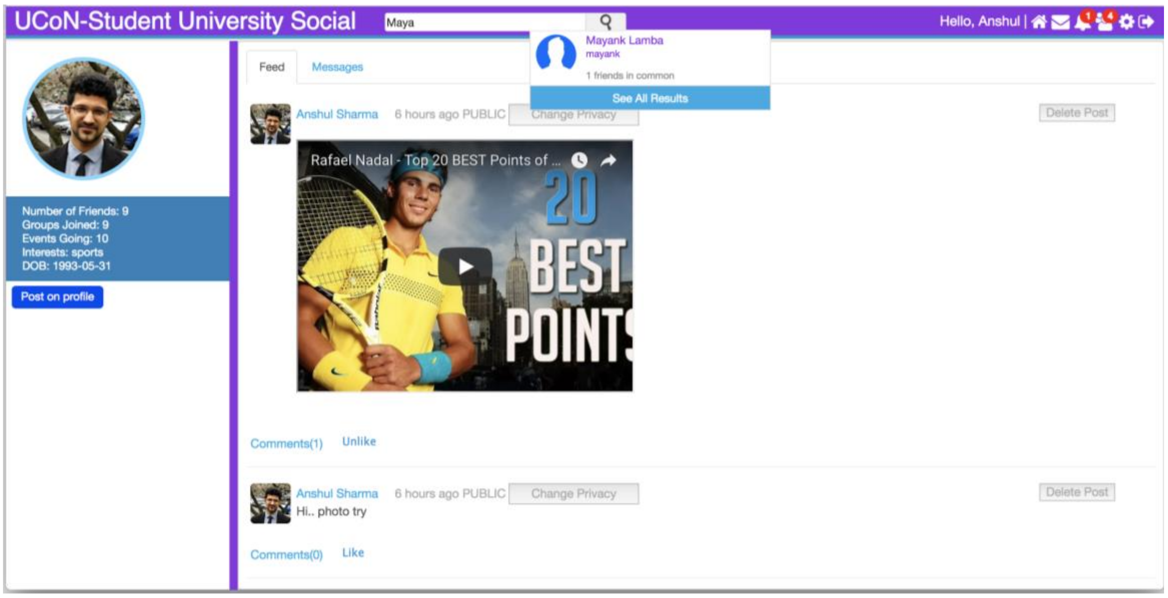
Comments(1) Unlike

Anshul Sharma 6 hours ago PUBLIC Change Privacy Delete Post
Hi.. photo try

Comments(0) Like

javascript:void(0);

4. Search:
   The user can also search the other users by typing the name in the textarea and then they can click on the user's name to see their profile.


UCoN-Student University Social

Maya

Mayank Lamba
mayank
1 friends in common

See All Results

Hello, Anshul |

Number of Friends: 9
Groups Joined: 9
Events Going: 10
Interests: sports
DOB: 1993-05-31

Post on profile

Feed Messages

Anshul Sharma 6 hours ago PUBLIC Change Privacy Delete Post

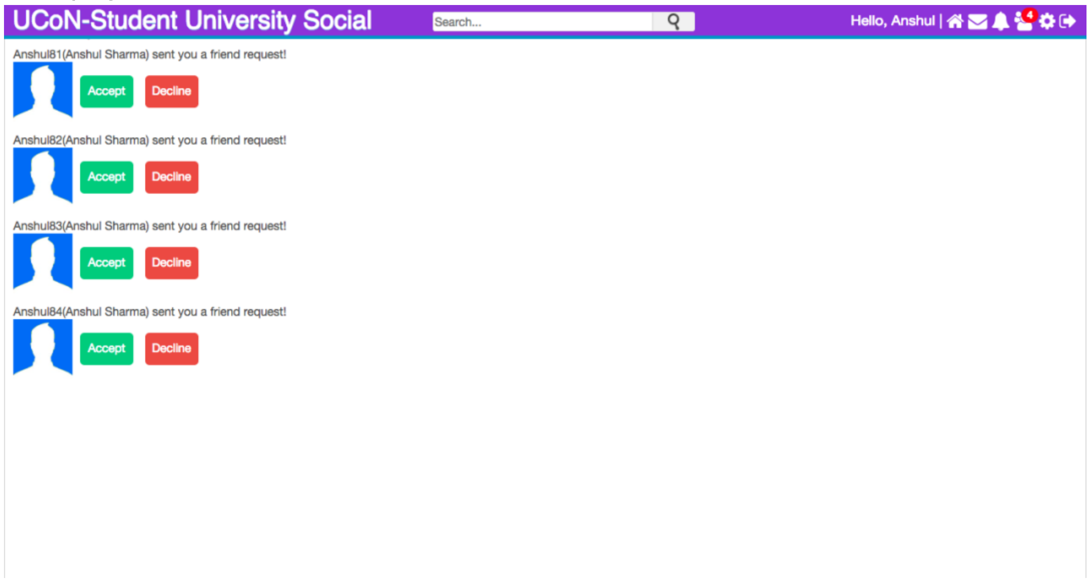
Rafael Nadal - Top 20 BEST Points of ...

Comments(1) Unlike

Anshul Sharma 6 hours ago PUBLIC Change Privacy Delete Post
Hi.. photo try

Comments(0) Like

5. Friend Requests:
   The friend request option navigates to new page showing the friend request which gives the user the option to accept and reject the request. This will remove the request from this page and add a record in the user table.



UCoN-Student University Social

Search...

Hello, Anshul |

Anshul81(Anshul Sharma) sent you a friend request!

Accept Decline

Anshul82(Anshul Sharma) sent you a friend request!

Accept Decline

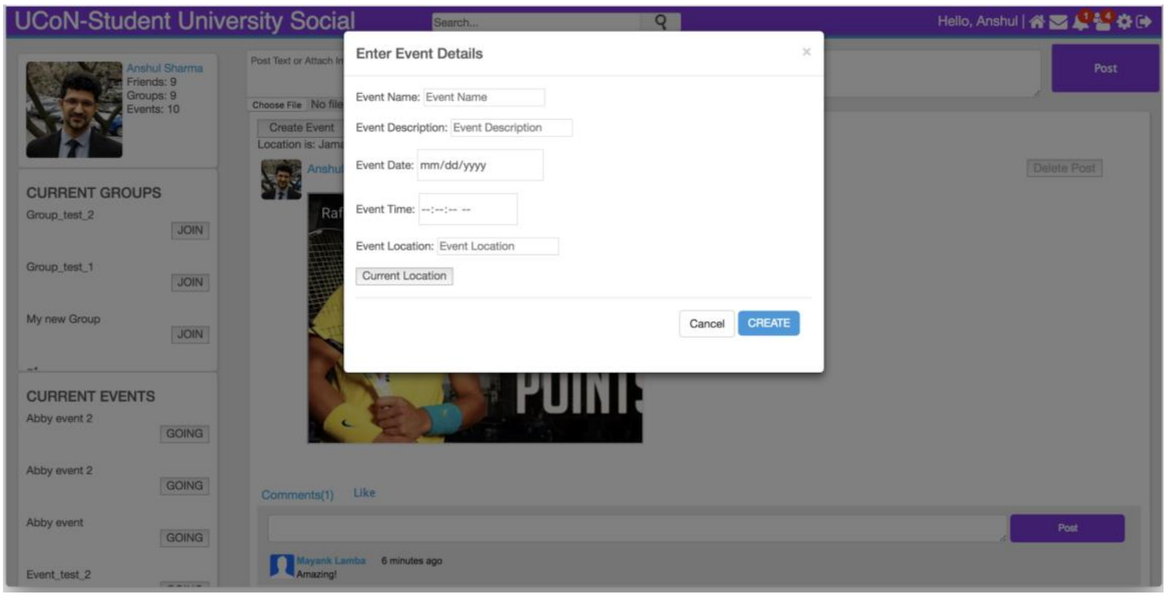
Anshul83(Anshul Sharma) sent you a friend request!

Accept Decline

Anshul84(Anshul Sharma) sent you a friend request!

Accept Decline

6. Create Event:
   Another option added is to create an event where user can go to. This adds an input to the event table with the description and the time and other details as shown. The user can select the option as to whether they will be going to an event or not.



UCoN-Student University Social

Search...

Hello, Anshul |

Anshul Sharma
Friends: 9
Groups: 9
Events: 10

Post Text or Attach Im

Choose File No file

Create Event

Location is: Jama

Post

Delete Post

CURRENT GROUPS

Group_test_2 JOIN

Group_test_1 JOIN

My new Group JOIN

CURRENT EVENTS

Abby event 2 GOING

Abby event 2 GOING

Abby event GOING

Event_test_2

Enter Event Details

Event Name: Event Name

Event Description: Event Description

Event Date: mm/dd/yyyy

Event Time: --:--:-- --

Event Location: Event Location

Current Location

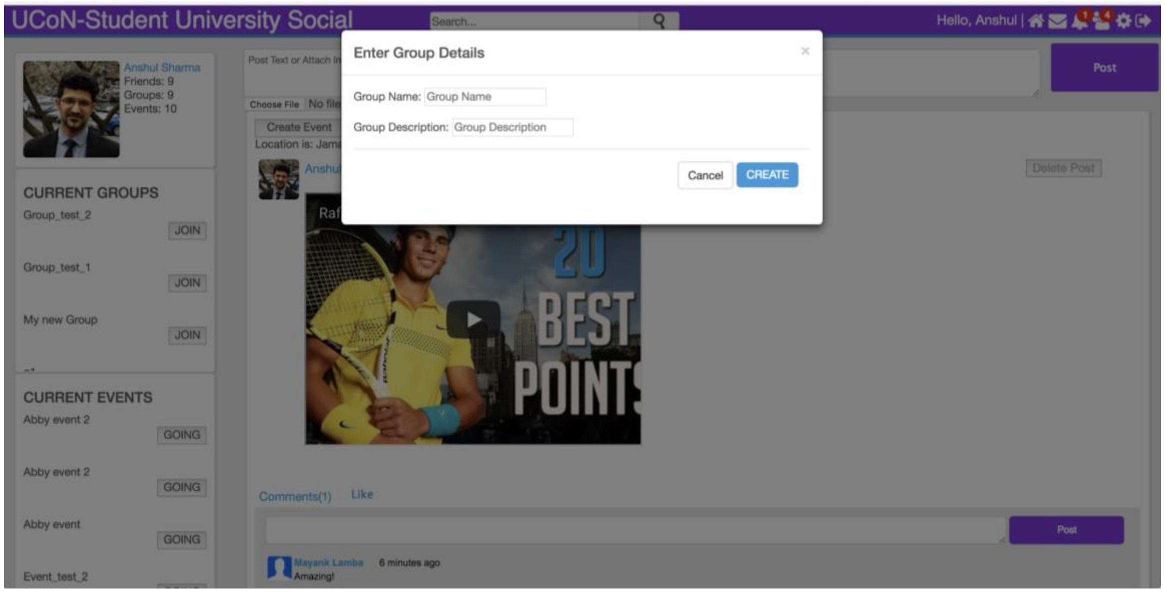
Cancel CREATE

Comments(1) Like

Post

Mayank Lamba 6 minutes ago
Amazing!

7. Creat Group:
   Users can also create a group and can join the group if they want to. They can add the group description and group name. And through the option on the home page, they can join the group as well.



Enter Group Details

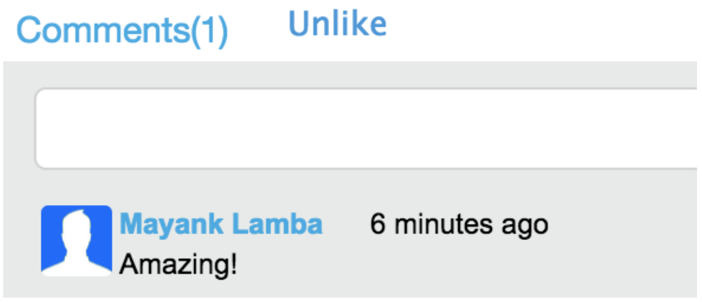
Group Name: Group Name

Group Description: Group Description

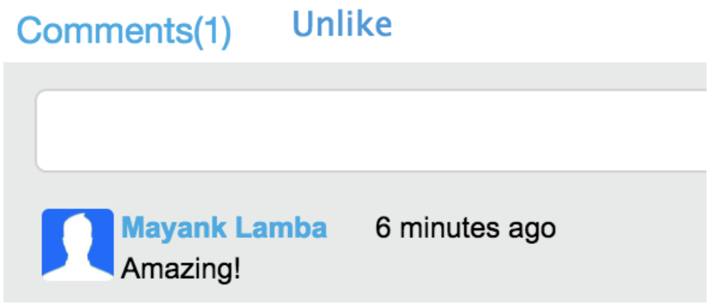
Cancel CREATE

Mayank Lamba 6 minutes ago
Amazing!

8. Like/Dislike:
   A user can like or unlike a post.

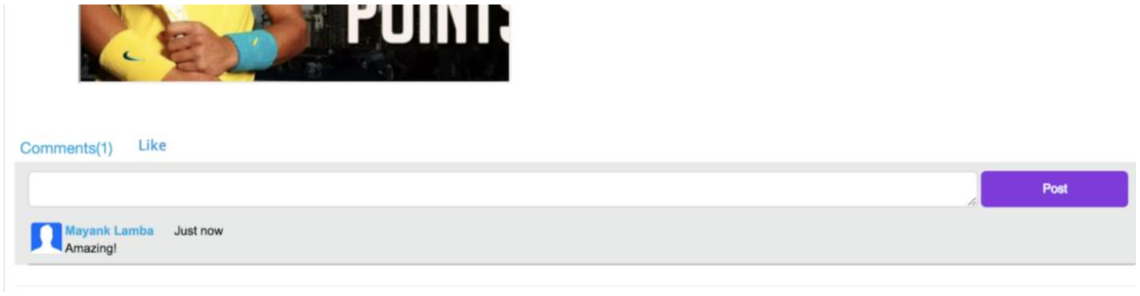


Mayank Lamba 6 minutes ago
Amazing!



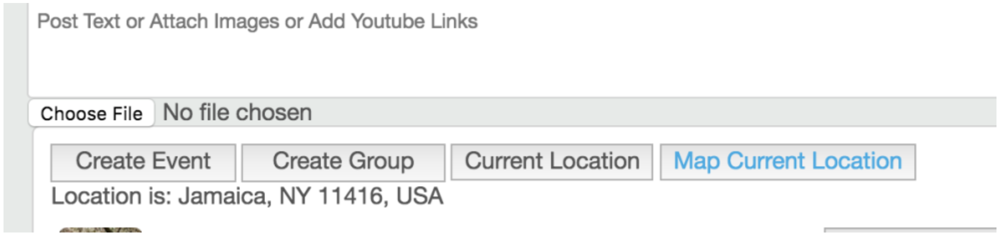
Mayank Lamba 6 minutes ago
Amazing!

9. Comment:
   A user can comment on a post.



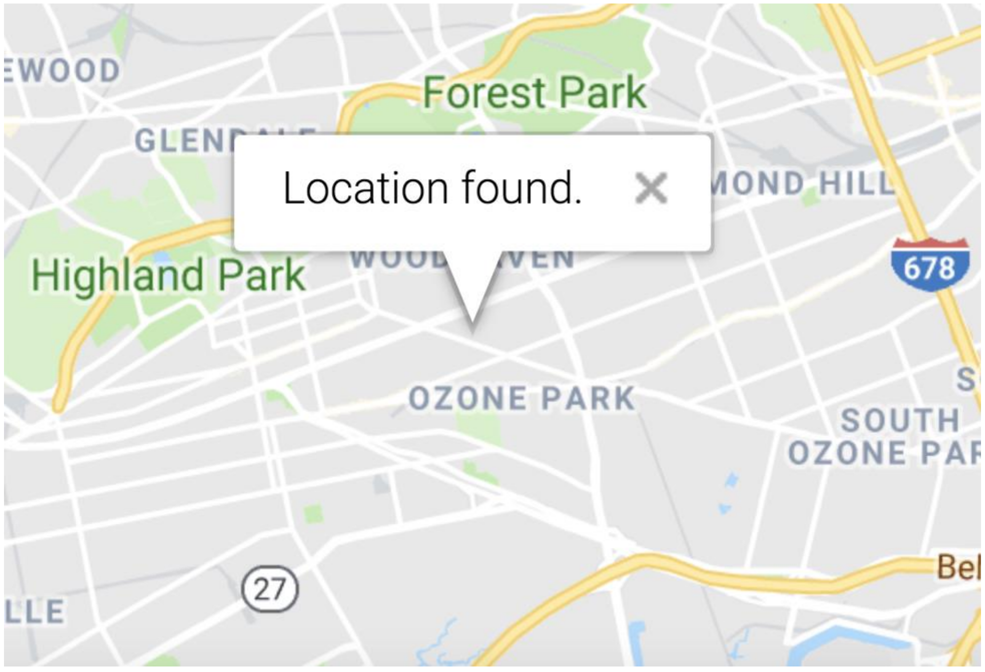
Mayank Lamba Just now
Amazing!

10. Location:
    A user has an option to check and add his/her location as well.
    When clicked on current location:

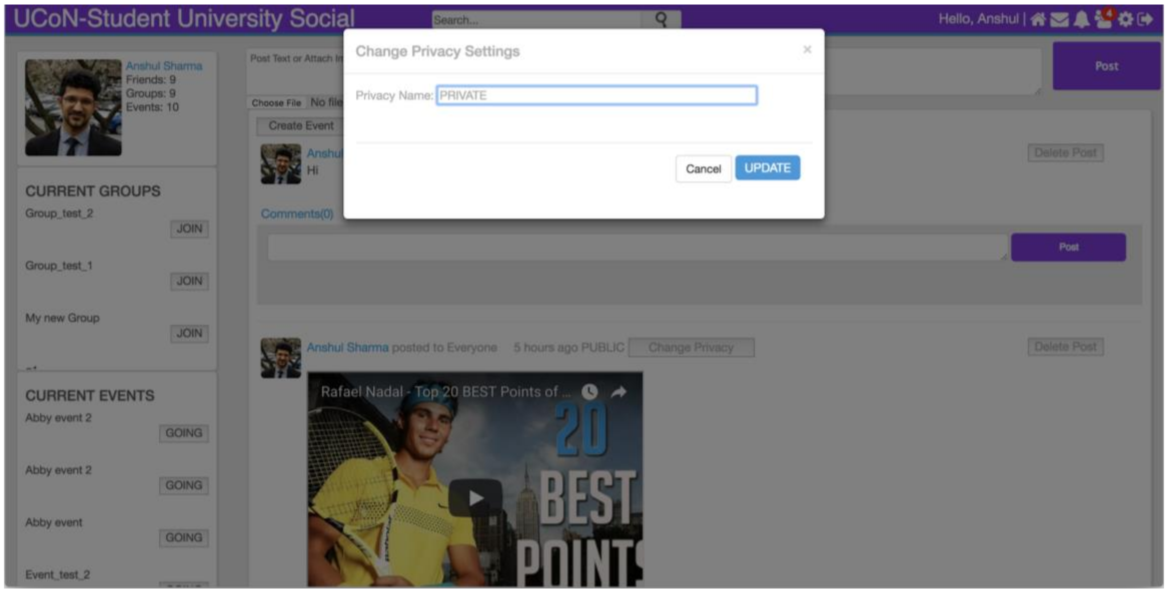


Location is: Jamaica, NY 11416, USA

When clicked on Map Current Location:



11. View Privileges:
    A user can change the view privileges of a post from public to private, friends, or friends of friends.



Change Privacy Settings

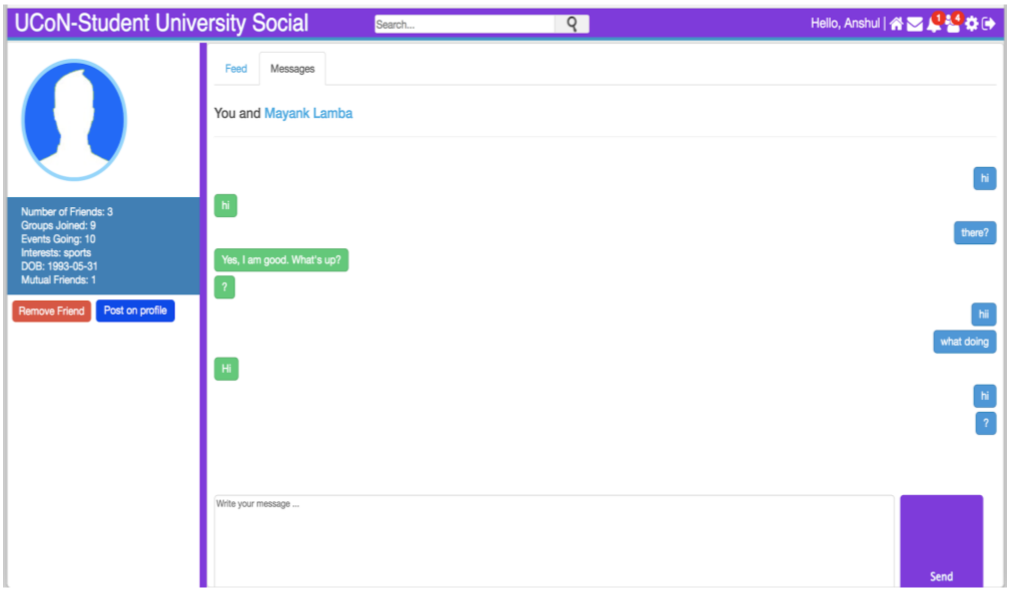
Privacy Name: PRIVATE

Extra Features:

1. Chat implementation:

   We have implemented the feature of chatting in a way that logged in users can easily have a conversation among each other.

   The difficulty for us was displaying the messages in the right order. We tried to use Ajax in order to do that in real time and hence preventing any problems with ordering.



Number of Friends: 3
Groups Joined: 9
Events Going: 10
Interests: sports
DOB: 1993-05-31
Mutual Friends: 1

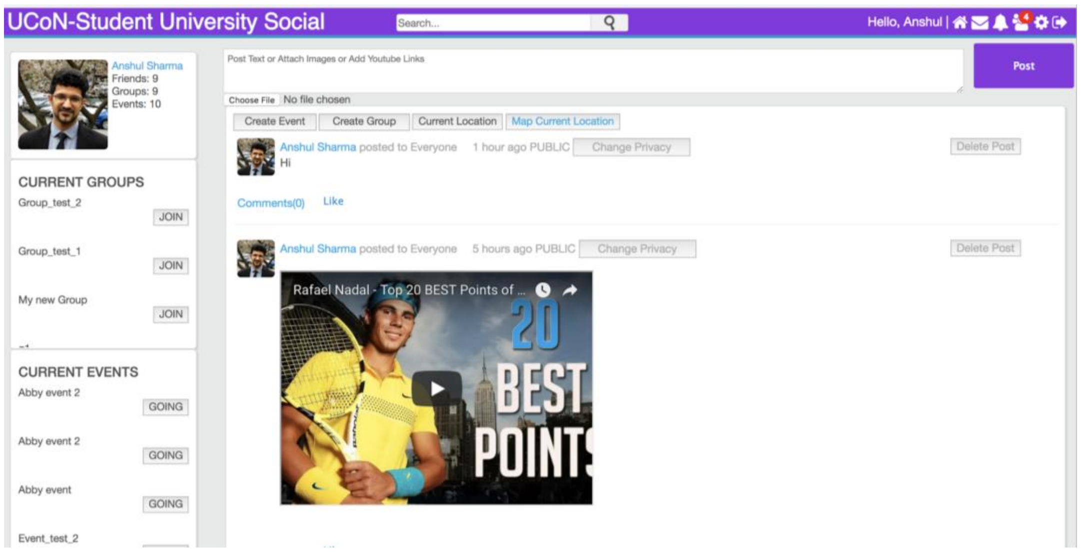
You and Mayank Lamba

hi
hi
there?
Yes, I am good. What's up?
?
hii
what doing
Hi
hi
?

2. Timeline:

   We have implemented a timeline for the logged in user in which we have basically displayed their posts, the posts that were attributed to them on their profile, or some public posts, along with addition information about them like their friend count, event count and group count.

   The difficulty we faced was that of ordering the posts and and showing the time. We did that by updating our sql queries as well as some if else statements for time. Also, it was a task to embed inline youtube links, which was resolved using iframes.



Anshul Sharma
Friends: 9
Groups: 9
Events: 10

CURRENT GROUPS
Group_test_2
Group_test_1
My new Group

CURRENT EVENTS
Abby event 2
Abby event 2
Abby event
Event_test_2

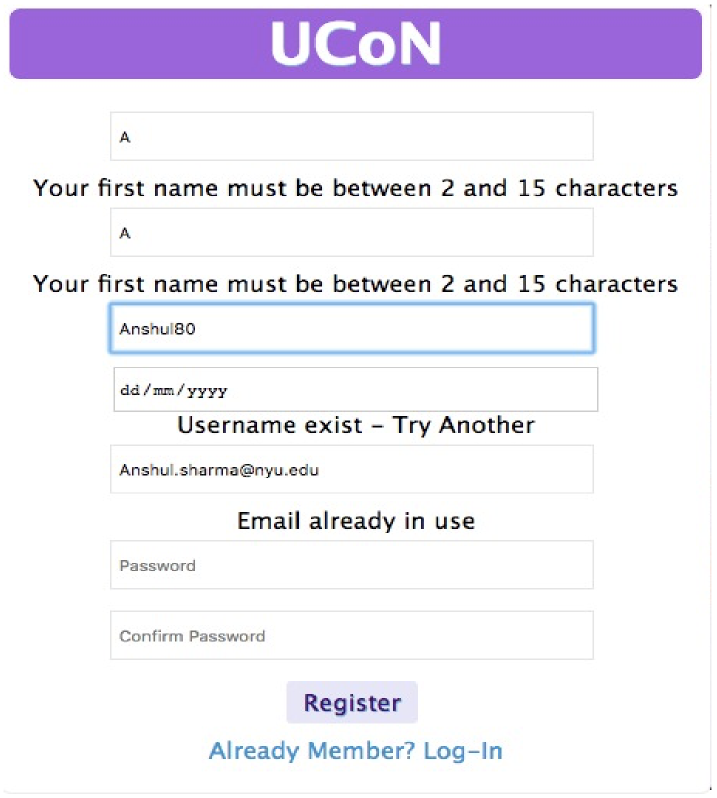
Anshul Sharma posted to Everyone 1 hour ago PUBLIC
Hi

Anshul Sharma posted to Everyone 5 hours ago PUBLIC

3. Input Checks:

We have applied various input checks while getting the data from a new user that is trying to register. They are as follows:

- Username can take in only alphanumeric values. There is a check on whether it already exists or not.
- Email should have '@' symbol present in it. There is a check on whether it already exists or not.
- Password should be in the range of 8-30 alphanumeric characters. While registering we make the user type the password again in order to match them and confirm.
- DOB cannot be a string and hence should be a calendar date.
- First and last names entered by the user should be 2-15 characters long.


UCoN

A

Your first name must be between 2 and 15 characters

A

Your first name must be between 2 and 15 characters

Anshul80

dd/mm/yyyy

Username exist – Try Another

Anshul.sharma@nyu.edu

Email already in use

Password

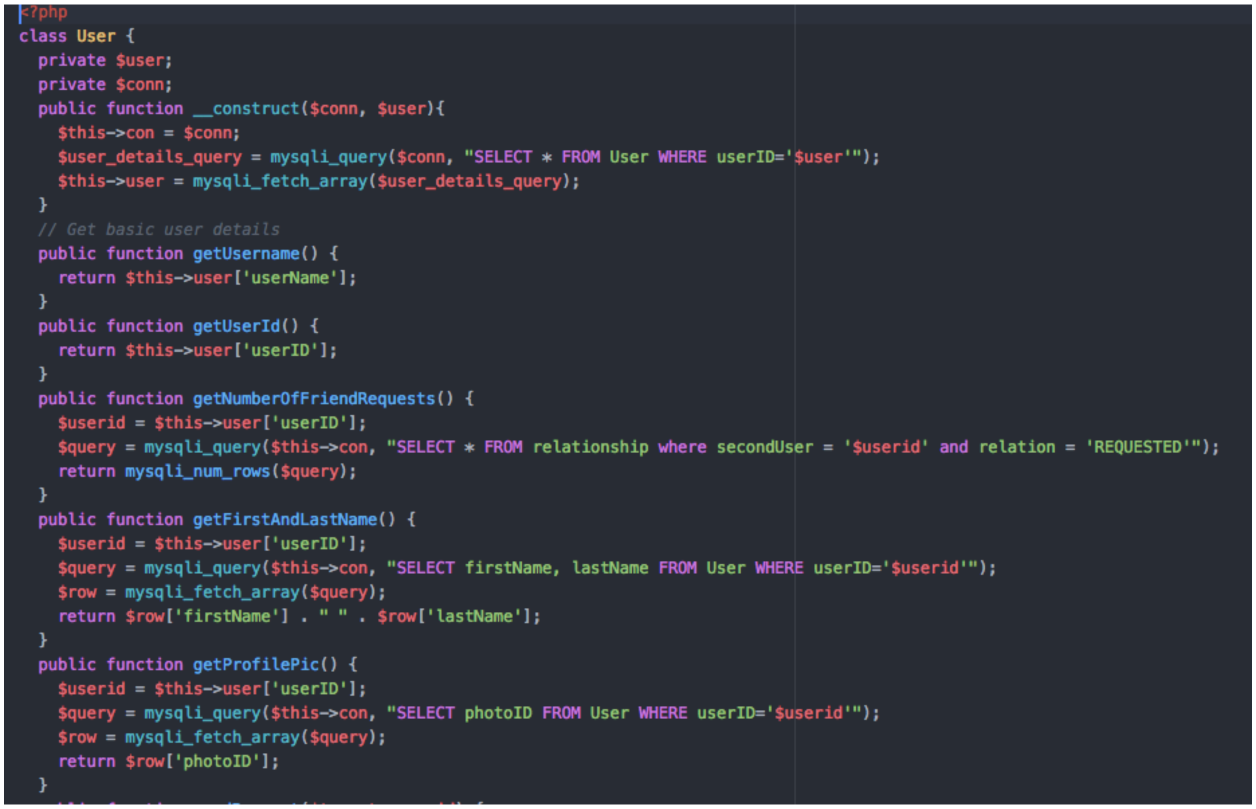
Confirm Password

Register

Already Member? Log-In

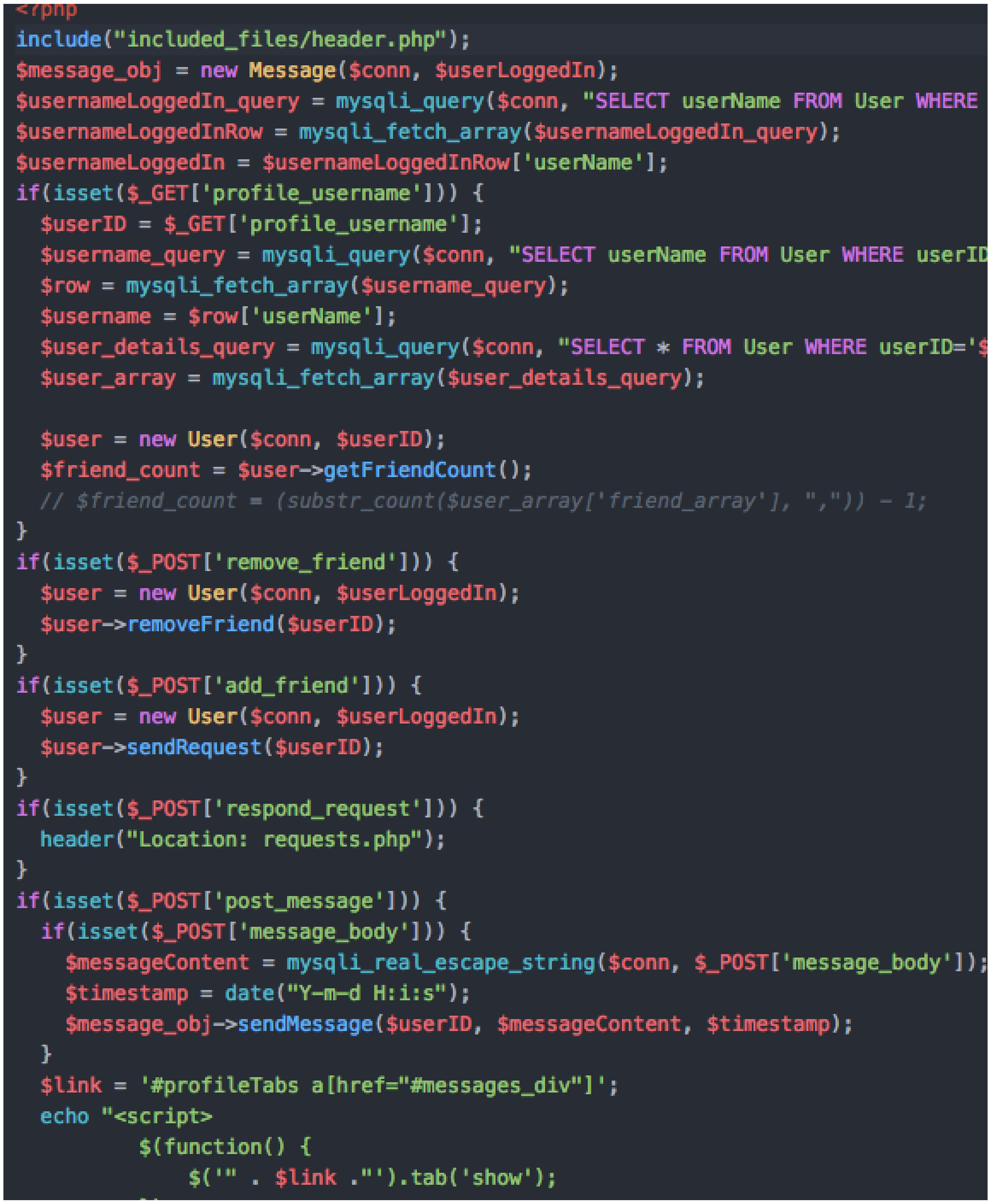
Code Snippets:

Get User Data for various purposes:



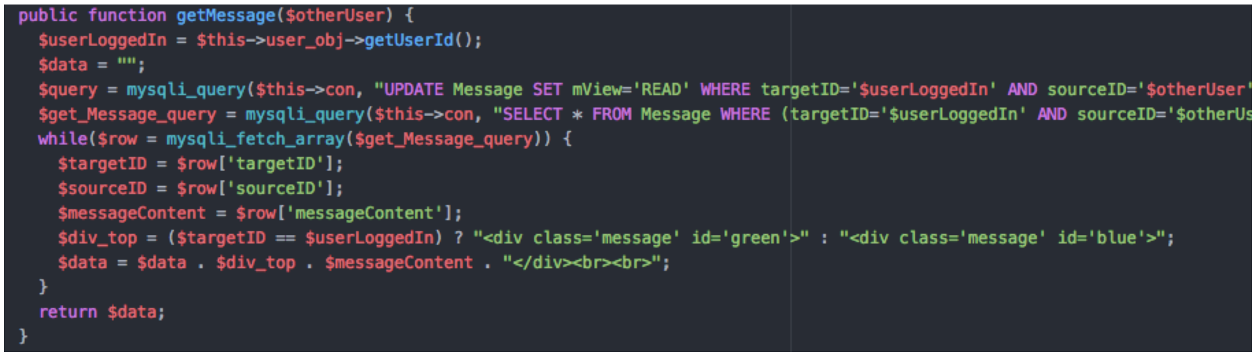
```php
<?php
class User {
  private $user;
  private $conn;
  public function __construct($conn, $user){
    $this->con = $conn;
    $user_details_query = mysqli_query($conn, "SELECT * FROM User WHERE userID='$user'");
    $this->user = mysqli_fetch_array($user_details_query);
  }
  // Get basic user details
  public function getUsername() {
    return $this->user['userName'];
  }
  public function getUserId() {
    return $this->user['userID'];
  }
  public function getNumberOfFriendRequests() {
    $userid = $this->user['userID'];
    $query = mysqli_query($this->con, "SELECT * FROM relationship where secondUser = '$userid' and relation = 'REQUESTED'");
    return mysqli_num_rows($query);
  }
  public function getFirstAndLastName() {
    $userid = $this->user['userID'];
    $query = mysqli_query($this->con, "SELECT firstName, lastName FROM User WHERE userID='$userid'");
    $row = mysqli_fetch_array($query);
    return $row['firstName'] . " " . $row['lastName'];
  }
  public function getProfilePic() {
    $userid = $this->user['userID'];
    $query = mysqli_query($this->con, "SELECT photoID FROM User WHERE userID='$userid'");
    $row = mysqli_fetch_array($query);
    return $row['photoID'];
  }
```

User Registration:



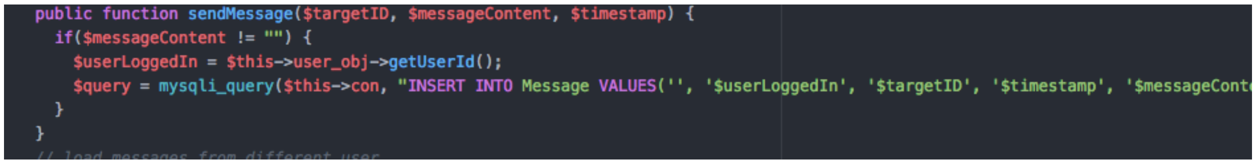
```
<?php
include("included_files/header.php");
$message_obj = new Message($conn, $userLoggedIn);
$usernameLoggedIn_query = mysqli_query($conn, "SELECT userName FROM User WHERE
$usernameLoggedInRow = mysqli_fetch_array($usernameLoggedIn_query);
$usernameLoggedIn = $usernameLoggedInRow['userName'];
if(isset($_GET['profile_username'])) {
  $userID = $_GET['profile_username'];
  $username_query = mysqli_query($conn, "SELECT userName FROM User WHERE userID
  $row = mysqli_fetch_array($username_query);
  $username = $row['userName'];
  $user_details_query = mysqli_query($conn, "SELECT * FROM User WHERE userID='$
  $user_array = mysqli_fetch_array($user_details_query);

  $user = new User($conn, $userID);
  $friend_count = $user->getFriendCount();
  // $friend_count = (substr_count($user_array['friend_array'], ",")) - 1;
}
if(isset($_POST['remove_friend'])) {
  $user = new User($conn, $userLoggedIn);
  $user->removeFriend($userID);
}
if(isset($_POST['add_friend'])) {
  $user = new User($conn, $userLoggedIn);
  $user->sendRequest($userID);
}
if(isset($_POST['respond_request'])) {
  header("Location: requests.php");
}
if(isset($_POST['post_message'])) {
  if(isset($_POST['message_body'])) {
    $messageContent = mysqli_real_escape_string($conn, $_POST['message_body']);
    $timestamp = date("Y-m-d H:i:s");
    $message_obj->sendMessage($userID, $messageContent, $timestamp);
  }
  $link = '#profileTabs a[href="#messages_div"]';
  echo "<script>
          $(function() {
              $('" . $link ."').tab('show');
```

Get Messages from other user:



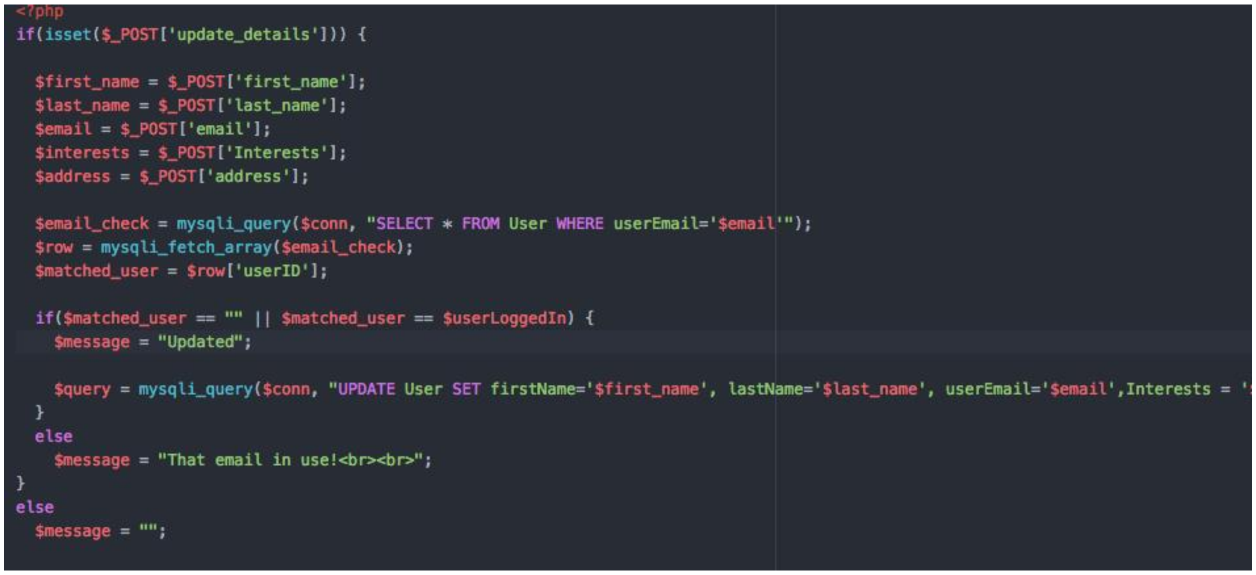
```php
public function getMessage($otherUser) {
  $userLoggedIn = $this->user_obj->getUserId();
  $data = "";
  $query = mysqli_query($this->con, "UPDATE Message SET mView='READ' WHERE targetID='$userLoggedIn' AND sourceID='$otherUser'
  $get_Message_query = mysqli_query($this->con, "SELECT * FROM Message WHERE (targetID='$userLoggedIn' AND sourceID='$otherUs
  while($row = mysqli_fetch_array($get_Message_query)) {
    $targetID = $row['targetID'];
    $sourceID = $row['sourceID'];
    $messageContent = $row['messageContent'];
    $div_top = ($targetID == $userLoggedIn) ? "<div class='message' id='green'>" : "<div class='message' id='blue'>";
    $data = $data . $div_top . $messageContent . "</div><br><br>";
  }
  return $data;
}
```

Send message to other user:



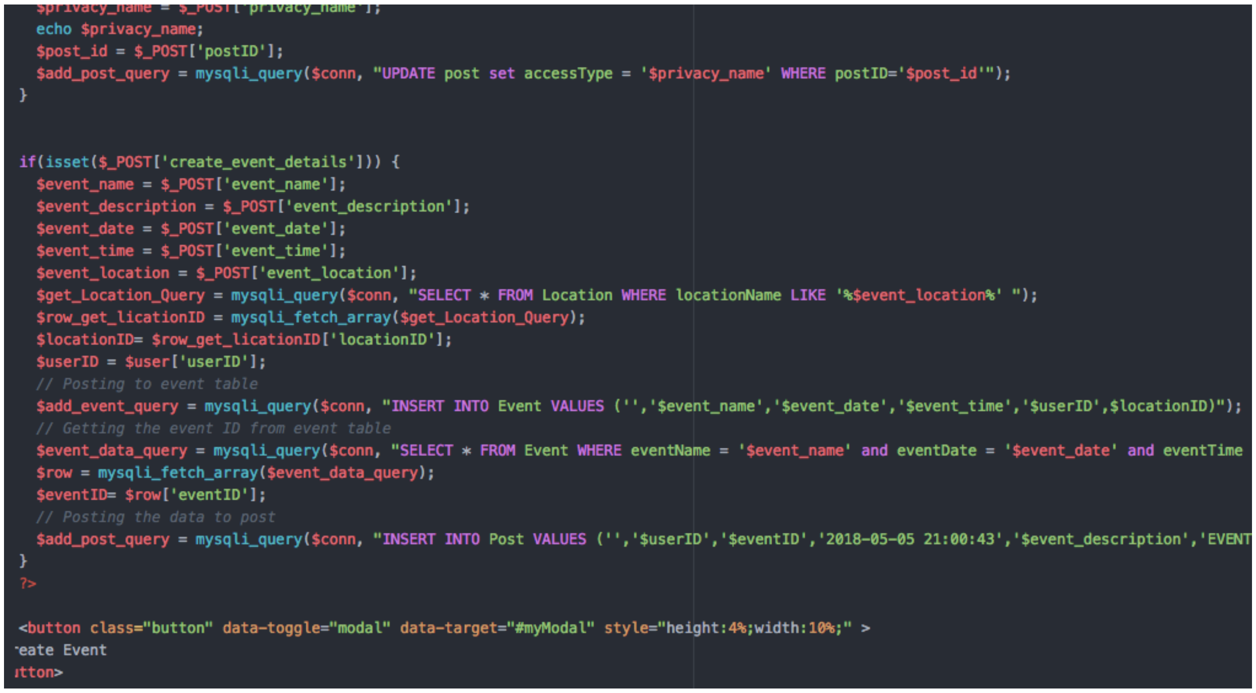
```php
  public function sendMessage($targetID, $messageContent, $timestamp) {
    if($messageContent != "") {
      $userLoggedIn = $this->user_obj->getUserId();
      $query = mysqli_query($this->con, "INSERT INTO Message VALUES('', '$userLoggedIn', '$targetID', '$timestamp', '$messageConte
    }
  }
  // load messages from different user
```

Update Profile:



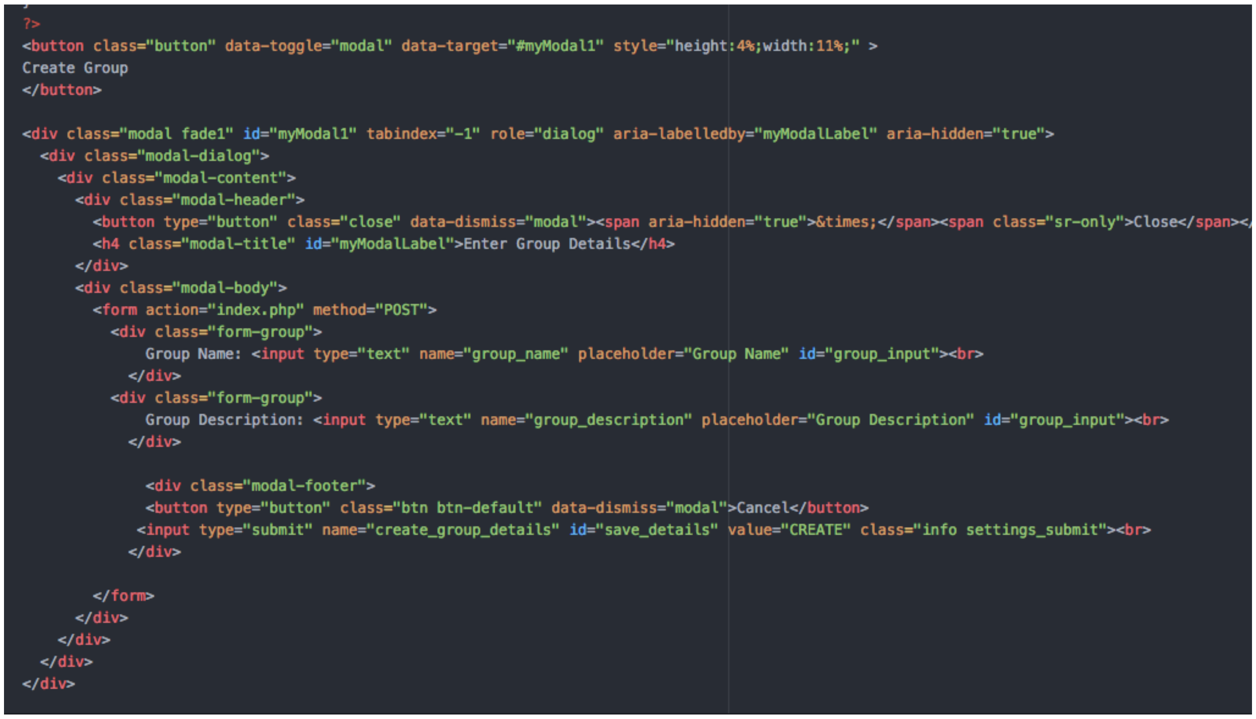
```php
<?php
if(isset($_POST['update_details'])) {

  $first_name = $_POST['first_name'];
  $last_name = $_POST['last_name'];
  $email = $_POST['email'];
  $interests = $_POST['Interests'];
  $address = $_POST['address'];

  $email_check = mysqli_query($conn, "SELECT * FROM User WHERE userEmail='$email'");
  $row = mysqli_fetch_array($email_check);
  $matched_user = $row['userID'];

  if($matched_user == "" || $matched_user == $userLoggedIn) {
    $message = "Updated";

    $query = mysqli_query($conn, "UPDATE User SET firstName='$first_name', lastName='$last_name', userEmail='$email',Interests = '
  }
  else
    $message = "That email in use!<br><br>";
}
else
  $message = "";
```

Create Events:



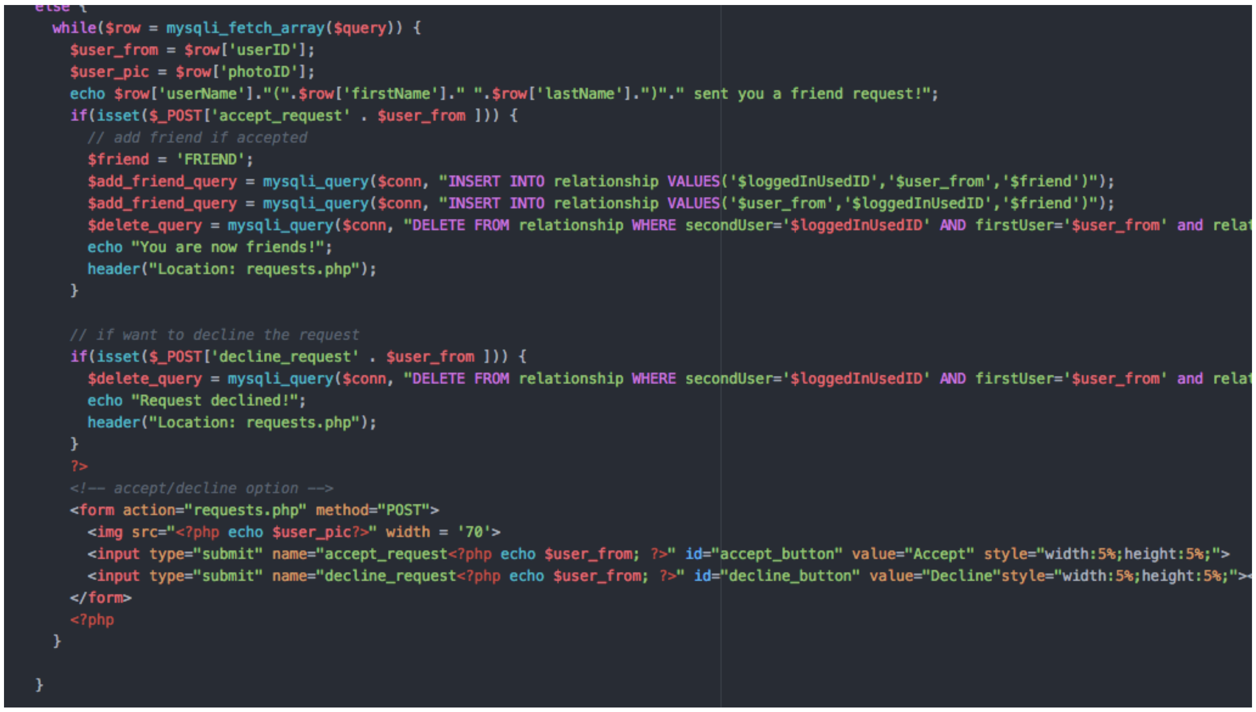
```
  $privacy_name = $_POST['privacy_name'];
  echo $privacy_name;
  $post_id = $_POST['postID'];
  $add_post_query = mysqli_query($conn, "UPDATE post set accessType = '$privacy_name' WHERE postID='$post_id'");
}


if(isset($_POST['create_event_details'])) {
  $event_name = $_POST['event_name'];
  $event_description = $_POST['event_description'];
  $event_date = $_POST['event_date'];
  $event_time = $_POST['event_time'];
  $event_location = $_POST['event_location'];
  $get_Location_Query = mysqli_query($conn, "SELECT * FROM Location WHERE locationName LIKE '%$event_location%' ");
  $row_get_licationID = mysqli_fetch_array($get_Location_Query);
  $locationID= $row_get_licationID['locationID'];
  $userID = $user['userID'];
  // Posting to event table
  $add_event_query = mysqli_query($conn, "INSERT INTO Event VALUES ('','$event_name','$event_date','$event_time','$userID',$locationID)");
  // Getting the event ID from event table
  $event_data_query = mysqli_query($conn, "SELECT * FROM Event WHERE eventName = '$event_name' and eventDate = '$event_date' and eventTime
  $row = mysqli_fetch_array($event_data_query);
  $eventID= $row['eventID'];
  // Posting the data to post
  $add_post_query = mysqli_query($conn, "INSERT INTO Post VALUES ('','$userID','$eventID','2018-05-05 21:00:43','$event_description','EVENT
}
?>

<button class="button" data-toggle="modal" data-target="#myModal" style="height:4%;width:10%;" >
eate Event
tton>
```

Create Groups:



```
 ?>
 <button class="button" data-toggle="modal" data-target="#myModal1" style="height:4%;width:11%;" >
 Create Group
 </button>

 <div class="modal fade1" id="myModal1" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
   <div class="modal-dialog">
     <div class="modal-content">
       <div class="modal-header">
         <button type="button" class="close" data-dismiss="modal"><span aria-hidden="true">&times;</span><span class="sr-only">Close</span></
         <h4 class="modal-title" id="myModalLabel">Enter Group Details</h4>
       </div>
       <div class="modal-body">
         <form action="index.php" method="POST">
           <div class="form-group">
               Group Name: <input type="text" name="group_name" placeholder="Group Name" id="group_input"><br>
             </div>
           <div class="form-group">
               Group Description: <input type="text" name="group_description" placeholder="Group Description" id="group_input"><br>
             </div>

               <div class="modal-footer">
               <button type="button" class="btn btn-default" data-dismiss="modal">Cancel</button>
              <input type="submit" name="create_group_details" id="save_details" value="CREATE" class="info settings_submit"><br>
             </div>


         </form>
       </div>
     </div>
   </div>
 </div>
```

Accept/Decline Request:



```
  else {
    while($row = mysqli_fetch_array($query)) {
      $user_from = $row['userID'];
      $user_pic = $row['photoID'];
      echo $row['userName']."(".$row['firstName']." ".$row['lastName'].")"." sent you a friend request!";
      if(isset($_POST['accept_request' . $user_from ])) {
        // add friend if accepted
        $friend = 'FRIEND';
        $add_friend_query = mysqli_query($conn, "INSERT INTO relationship VALUES('$loggedInUsedID','$user_from','$friend')");
        $add_friend_query = mysqli_query($conn, "INSERT INTO relationship VALUES('$user_from','$loggedInUsedID','$friend')");
        $delete_query = mysqli_query($conn, "DELETE FROM relationship WHERE secondUser='$loggedInUsedID' AND firstUser='$user_from' and relat
        echo "You are now friends!";
        header("Location: requests.php");
      }

      // if want to decline the request
      if(isset($_POST['decline_request' . $user_from ])) {
        $delete_query = mysqli_query($conn, "DELETE FROM relationship WHERE secondUser='$loggedInUsedID' AND firstUser='$user_from' and relat
        echo "Request declined!";
        header("Location: requests.php");
      }
      ?>
      <!-- accept/decline option -->
      <form action="requests.php" method="POST">
        <img src="<?php echo $user_pic?>" width = '70'>
        <input type="submit" name="accept_request<?php echo $user_from; ?>" id="accept_button" value="Accept" style="width:5%;height:5%;">
        <input type="submit" name="decline_request<?php echo $user_from; ?>" id="decline_button" value="Decline"style="width:5%;height:5%;">
      </form>
      <?php
    }

  }
```