**Yap.com**

An online social networking website for getting together people from all across the world.

**INDEX :**

**Sr No. Topic Pg. No**

1. Introduction.............................................3
2. Breakdown Of The Project......................4
3. Software Used.........................................5
4. Features...................................................6
5. Database Schema.....................................8
6. Front-End Work......................................10
7. Back-End Work.......................................12
8. Marquee Feature : Chat Services.............16
9. References...............................................21

**Introduction :**

The website Yap.com is a social networking site that helps connect millions of users over a virtual social media. Our main aim through this open source technologies project is to create and implement a social networking website with various features to connect people from all walks of life on one single platform. All kinds of information can be easily accessed and shared over this network. Yap has all features which allow a user to socialize like in a real world scenario. Users need to register themselves. Yap provides each user with a unique profile page exclusive to the user on Yap which needs to be logged in on. This is also allows users to update their status, profile images etc. Friend requests can be sent and accepted to and by people from all over the world . Also, users can post various kinds of stuff on their walls which can be seen only by their friends on their respective walls. Each post can be liked as well. The marquee feature of Yap is the implementation of a chatroom or a chat service which permits.

Live chatting between any two friends. The website has been developed keeping in mind the above and the needs of the end-users enabling them to maintain and update their friendlist. Like any general registration based website users will also be notified about friend requests, new friends and incoming messages.

**Breakdown of The Project**

1. **Creating and Defining the problem statement:**

Here the problem statement will be defined and the website will be broken down into a series of tasks to be completed according to its features and requirements.

2. **Software Installation:**

All softwares required for this project and the development of Yap.com will be installed and made to work properly in this stage.

3. **Segregation of Work:**

Work will be segregated into units like front-end, back-end, incorporation of both and deployment of project.

4. **Front-End**

The designing of the website comesunder this section which defines how the outside world will see our Website.

5. **Back-End**

The internal and intricate working of the project comes under this stage where functions, working of the project hidden from the front users will be worked upon.

6**. Testing:**

Debugging the portability, persistence to complexity of database.

7. **Deployment:**

Finally delivering the website in a fully functional manner is the end step of the plan.

**Software Used :**

1. **XAMPP** is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.
2. **HTML**(HyperTextMarkupLanguage): It is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language) used to create [web pages](https://en.wikipedia.org/wiki/Web_page).  [Web browsers](https://en.wikipedia.org/wiki/Web_browser) can read HTML files and render them into visible or audible web pages. HTML describes the structure of a [website](https://en.wikipedia.org/wiki/Website) [semantically](https://en.wikipedia.org/wiki/Semantic) . It is used to develop and design webpage.
3. **CSS**(Cascading style sheet):It is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](https://en.wikipedia.org/wiki/Markup_language). CSS is designed primarily to enable [the separation of document content from document presentation](https://en.wikipedia.org/wiki/Separation_of_presentation_and_content), including aspects such as the [layout](https://en.wikipedia.org/wiki/Page_layout), [colors](https://en.wikipedia.org/wiki/Color), and [fonts](https://en.wikipedia.org/wiki/Typeface). It is used to enhance the webpages.
4. **JS**(Java Script):It is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [dynamic](https://en.wikipedia.org/wiki/Dynamic_programming_language), [untyped](https://en.wikipedia.org/wiki/Programming_language#Type_system), and [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) programming language. JavaScript is [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming) with [first-class functions](https://en.wikipedia.org/wiki/First-class_function), making it a [multi-paradigm](https://en.wikipedia.org/wiki/Multi-paradigm) language,supporting [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming),[[8]](https://en.wikipedia.org/wiki/JavaScript#cite_note-ECMA-262-8) [imperative](https://en.wikipedia.org/wiki/Imperative_programming), and [functional](https://en.wikipedia.org/wiki/Functional_programming) programming styles. It is used to make website more interactive.
5. **Bootstrap:**It is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software) collection of tools for creating [websites](https://en.wikipedia.org/wiki/Website) and [web applications](https://en.wikipedia.org/wiki/Web_application). It contains [HTML](https://en.wikipedia.org/wiki/HTML)- and [CSS](https://en.wikipedia.org/wiki/CSS)-based design templates for [typography](https://en.wikipedia.org/wiki/Typography), forms, buttons, navigation and other interface components, as well as optional [JavaScript](https://en.wikipedia.org/wiki/JavaScript) extensions. It aims to ease the [development](https://en.wikipedia.org/wiki/Web_development) of [dynamic websites](https://en.wikipedia.org/wiki/Dynamic_web_page) and [web applications](https://en.wikipedia.org/wiki/Web_application).It is used to make website platform independent to make it available for any resolution screen.
6. **PHP:**It is a [server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting) language designed for [web development](https://en.wikipedia.org/wiki/Web_development) but also used as a [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). PHP code may be embedded into [HTML](https://en.wikipedia.org/wiki/HTML) code, or it can be used in combination with various [web template systems](https://en.wikipedia.org/wiki/Web_template_system), web content management system and [web frameworks](https://en.wikipedia.org/wiki/Web_framework).It is used to do the server side coding.
7. **MySQL:**It is an [open-source](https://en.wikipedia.org/wiki/Open-source) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). The [SQL](https://en.wikipedia.org/wiki/SQL) abbreviation stands for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language). It is used to create databases.

**Features:**

* **Signup/ Login**  - A user must register on Yap.com by using providing some information to create his/her own unique profile and then login in order to gain access to profile pages exclusive to them thus giving them acces to their personal information and connecting them to the world through Yap.com .
* **Personal Info :** Every user will be associated with a profile picture, a cover picture and a status that are editable.
* **Search Engine:** A largely efficient search engine used to search for people based on keywords is provided by Yap.com. The search results automatically provide the user with the option of sending a friend request to people.
* **Wall :** Each user will be provided with a wall on which various stuff can be postedthat will be reflected on the walls of all friends of the user. Each post can be liked once by each user and can be deleted from their respective walls.
* **Notifications Bar :** A dropdown on the navigation bar of the profile page of Yap.com notifies all of its users of any friend requests received or of any recent additions to their friendlist.
* **Settings/Logout** : The settings and logout option like for any general registration based website allows the user to edit their profiles and logout of their respective Yap profiles.
* **Chat Room :** The best feature of this social networking site allows live chatting between friends registered on the site through text messages and images and also notifies each user of the number of unread messages and the number of unread messages from each friend.

**Database Schema:**

**Relational Database Schema :**

users(userid : integer primary key, Email : string, Password : string, Fname : string, Lname : string)

logs(id : integer primary key, username: string, msg : string, fromid : integer, toid : integer, readmsg : boolean)

posts(post\_id : integer primary key, friend\_id : integer, post : string, likes : integer)

**As users are created, entities are created dynamically for each user with emailid.**

friends\_emailid(friend\_id : integer primary key ,reqsent : Boolean, reqreceived : Boolean, friends : boolean )

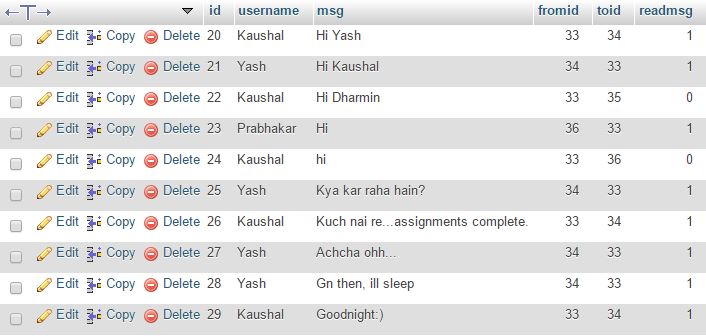
posts\_emailid(postid : integer foreign key, post: string, friend\_id : integer, liked : boolean, seen : boolean)

**Screenshots of tables :**

**users:**

****

**logs:**

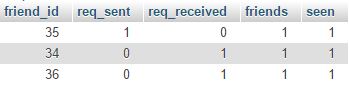
****

**posts:**

**posts.JPG**

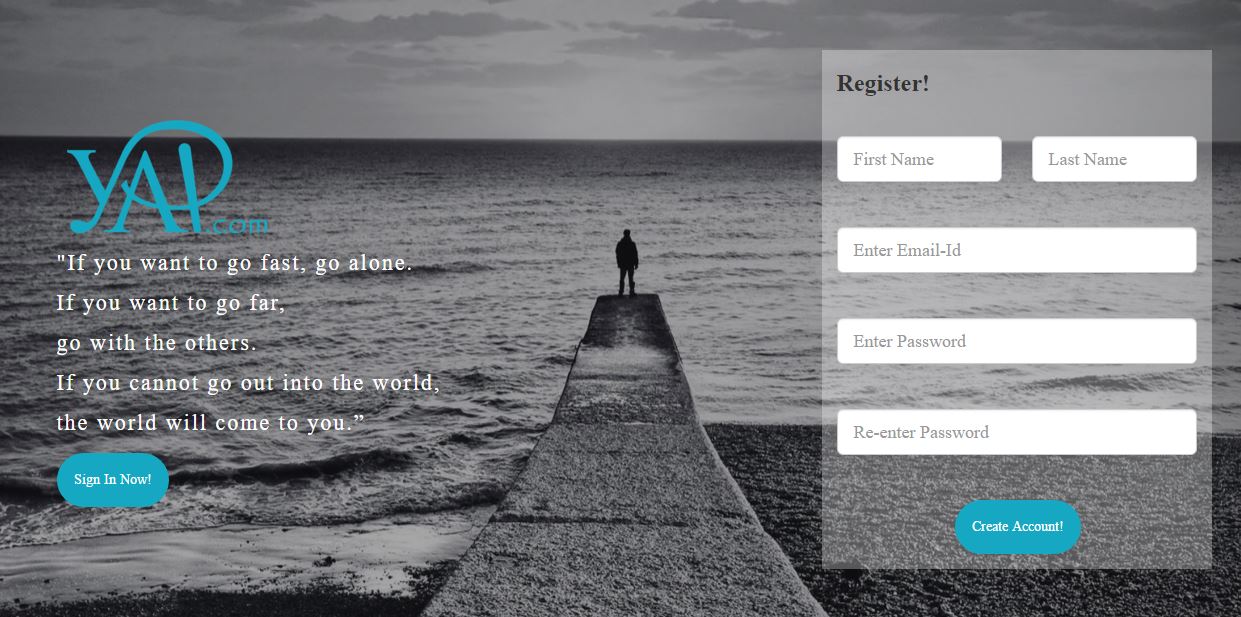
**For user\_id = 33, posts and friends tables respectively:**

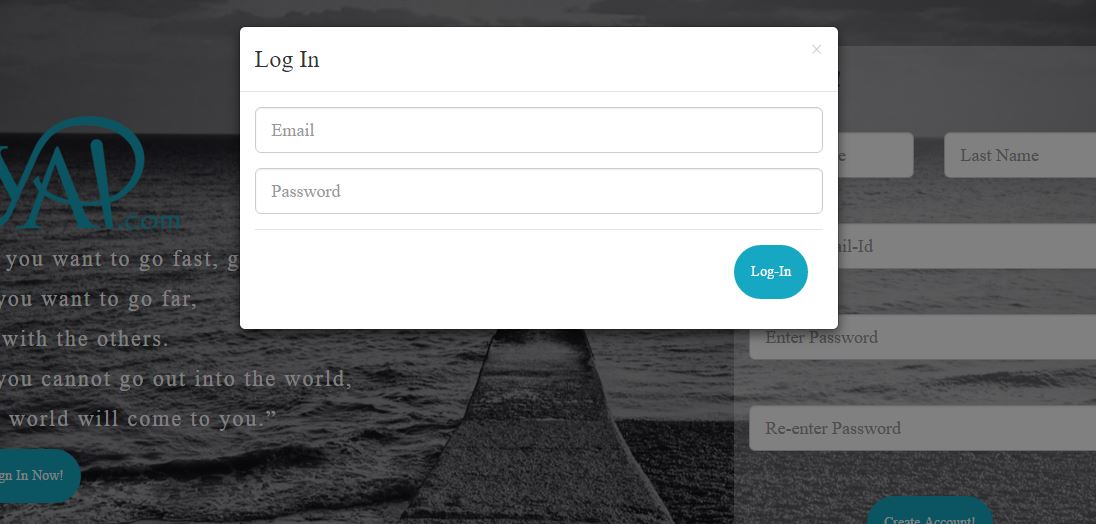
**post.JPG**

****

**Front-End**

The website has been designed using HTML and CSS . The styling has been done using CSS and the content has been added using HTML .The website has an attractive **homepage** with a register and a login option.





The homepage has:

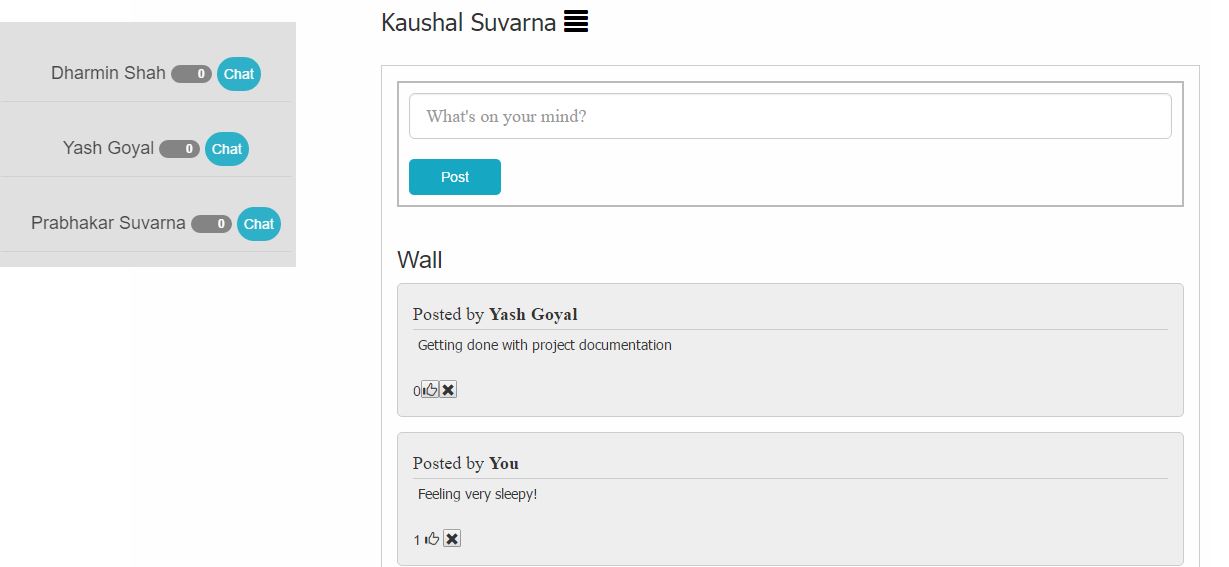
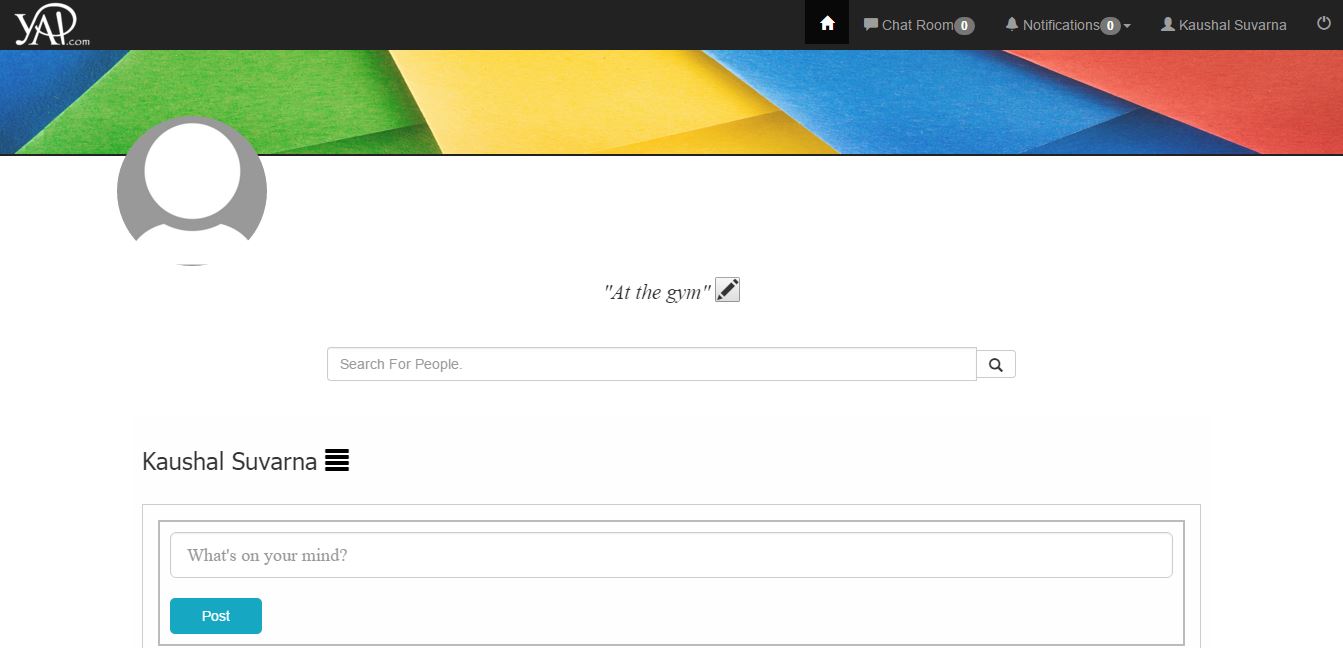
* A button to sign-in using already registered credentials.
* A registration form to register as a new user onto Yap.

The profile page consists of the editable status, profile picture and cover picture along with displaying the users name.

The navigation bar on top has links to the chatroom that displays the unread messages count, another notifications section for diplaying new friends and friend requests and a logout option.

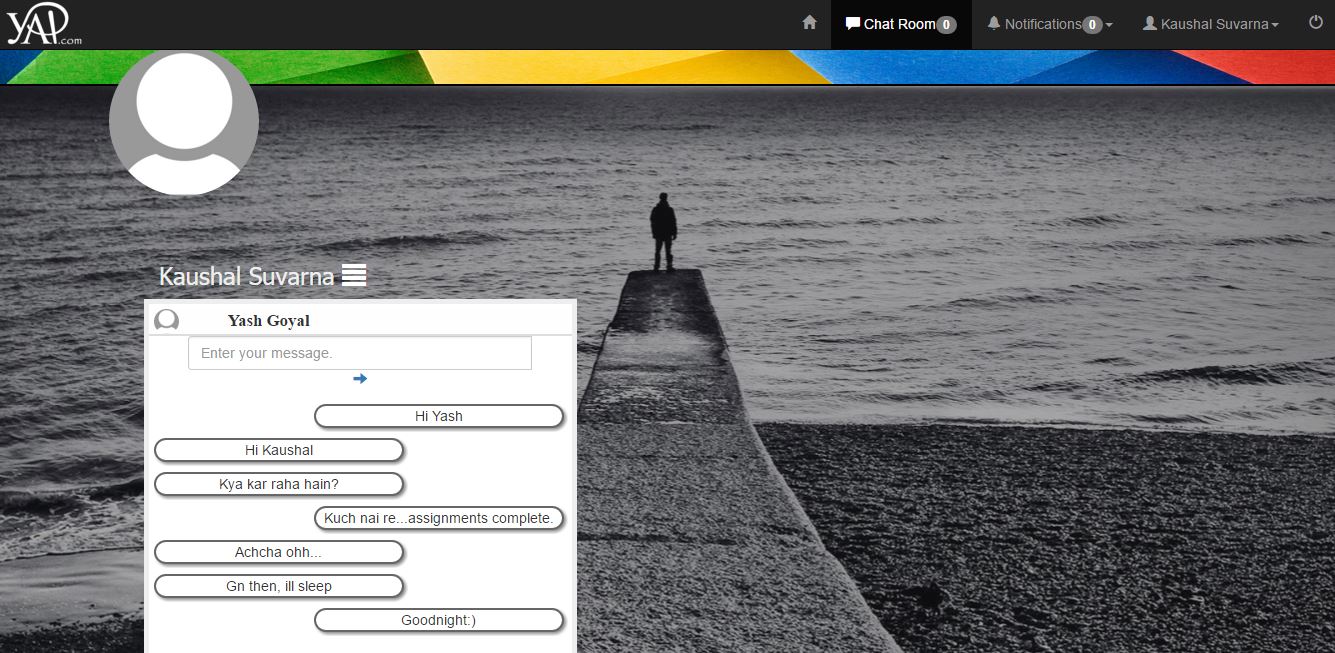
Below the profile picture is the search bar that allows users to search for people.

Next to the name is a button to display a toggleable sidebar to diplay the list of friends.

****

Below the name is the wall of the user where posts posted by the user and his/her friends are displayed.

Next is the chatroom whose screenshot displayed but will be discussed in further sections.

****

**Backend :**

Signup/Login :

Whenever a user signs up for the first time, his/her credentials are taken and inserted into the database as a new user into the users table. Along with this two new tables are dynamically created for each new registration, posts\_emailid and friends\_emaild where emailed stands for the particular users emailed. On logging in, the information is taken from the user and credentials are checked from the users table, If they match, page is redirected to the profile page exclusive to this particular user. The code is as follows:

<?php

$query1 = "SELECT `Id` FROM `users` WHERE `Email` = '$email'";

$query\_run1 = mysql\_query($query1);

if(mysql\_num\_rows($query\_run1)!=0){

echo ' Already existing user';

}

else{

$var = 'friends\_'.$email;

$query2 = "INSERT INTO users(Email,Password,Username,Fname,Lname) VALUES ('$email','$password','$fname','$fname','$lname')";

$query3 = "CREATE TABLE `ost`.`$var`( `friend\_id` INT NOT NULL , `req\_sent` BOOLEAN NOT NULL DEFAULT FALSE , `req\_received` BOOLEAN NOT NULL DEFAULT FALSE , `friends` BOOLEAN NULL DEFAULT NULL , `seen` BOOLEAN NOT NULL DEFAULT FALSE)";

$var = 'posts\_'.$email;

$query4 = "CREATE TABLE `ost`.`$var`( `post\_id` INT NOT NULL PRIMARY KEY, `post` VARCHAR(500) NOT NULL , `friend\_id` INT ,`liked` BOOLEAN DEFAULT FALSE, `seen` BOOLEAN DEFAULT FALSE)";

if($query\_run4 = mysql\_query($query4)){

echo ' Create table query run';

}

else{

echo mysql\_error();

}

if($query\_run3 = mysql\_query($query3)){

echo ' Create table query run';

}

else{

echo mysql\_error();

}

if($query\_run2 = mysql\_query($query2)){

echo ' Insert query run';

header('Location: index.php');

}

else{

echo mysql\_error();

}

?>

<?php

$query = "SELECT `Email`, `Password`, `Id`, `Fname`, `Lname` FROM `users` WHERE `Email` = '$email' AND `Password` = '$password'";

if($query\_run = mysql\_query($query)){

echo 'Query run';

$query\_num\_rows = mysql\_num\_rows($query\_run);

if($query\_num\_rows==0){

//alert('Sorry! Your Email-Id or Password is incorrect');

//header('Location: index.php');

echo '<script language="javascript">';

echo 'alert("Sorry! Your Login Credentials do not match.")';

echo '</script>';

header("Location:index.php");

}

else if($query\_num\_rows==1){

$user\_id = mysql\_result($query\_run,0,'Id');

$\_SESSION['user\_id'] = $user\_id;

if(!empty($\_SESSION['user\_id'])){

echo 'Logged in';

}

header('Location: Profile.php');

}

}

else{

echo mysql\_error();

echo 'Query not run';

}

?>

Logging in creates a new session for a user. Logging out simply closes that session and gets ready for a new session.

On sending a friend request, a new tuple will be inserted in the users friend table with friend\_id of the friend and req-sent = true. Similarly a new entry is inserted into the friends table of the friend with the friend\_id set to the userid of the user sending the request and req\_received set ti true.

if(isset($\_POST['friend\_id'])){

$friend = $\_POST['friend\_id'];

$you = $\_SESSION['user\_id'];

$to = 'friends\_'.getuserfield('Email',$friend);

$from = 'friends\_'.getuserfield('Email',$\_SESSION['user\_id']);

$query1 = "INSERT INTO `$from`(friend\_id,req\_sent) VALUES ('$friend','1')";

$query2 = "INSERT INTO `$to`(friend\_id,req\_received) VALUES ('$you','1')";

if($query\_run1 = mysql\_query($query1)){

echo ' Insert query run';

}

else{

echo mysql\_error();

}

if($query\_run2 = mysql\_query($query2)){

echo ' Friend request sent.';

}

else{

echo mysql\_error();

}

}

For any user to respond to any friend request, they have three options, accepting changes the friends value on both the friends table to true, ignore deletes the corresponding entry from both tables and block sets friends = false so that no friends request can be exchanged between the two in future.

if(isset($\_POST['accept'])){

$email = getuserfield('Email',$\_SESSION['user\_id']);

echo $email;

$id = $\_POST['friend\_id'];

echo $id;

echo 'id';

$query1 = "UPDATE `friends\_$email` set `friends` = 1 WHERE friend\_id= $id";

$from = getuserfield('Email',$\_POST['friend\_id']);

$query2 = "UPDATE `friends\_$from` set `friends` = 1 WHERE `friend\_id` = $\_SESSION[user\_id]";

echo 'gghredcp\n';

if($query\_run1 = mysql\_query($query1)){

echo 'gogogogog\n';

if($query\_run2 = mysql\_query($query2)){

header('Location: Profile.php');

}

else

echo mysql\_error();

}

else{

echo mysql\_error();

}

}

Now posting, does two things : inserts a new entry in the posts table with post = the string entered and then using the post id from this table for this particular post as foreign key first makes an entry into the post table for the user and then into the the posts table of all the friends of the user. The code for this is :

if(isset($\_POST['post'])){

$id = $\_SESSION['user\_id'];

$email = getuserfield('Email',$id);

$p = $\_POST['post'];

$query3 = "INSERT INTO `posts` (`post`, `friend\_id`) VALUES ('$p',$id)";

if($query\_run3 = mysql\_query($query3)){

$query4 = "SELECT MAX(`post\_id`) as max FROM `posts`";

if($query\_run4 = mysql\_query($query4)){

$row = mysql\_fetch\_array($query\_run4);

$postid = $row['max'];

}

}

else{

echo mysql\_error();

}

$query1 = "INSERT INTO `posts\_$email` (`post\_id`,`post`, `friend\_id`) VALUES ('$postid','$p',$id)";

if($query\_run1 = mysql\_query($query1)){

$query2 = "SELECT \* FROM `friends\_$email` WHERE friends = 1";

if($query\_run2 = mysql\_query($query2)){

if($query\_run2){

while($row = mysql\_fetch\_array($query\_run2)){

$friend\_id = $row['friend\_id'];

$friend = getuserfield('Email',$friend\_id);

$query3 = "INSERT INTO `posts\_$friend` (`post\_id`,`post`, `friend\_id`) VALUES ('$postid','$p',$id)";

if($query\_run3 = mysql\_query($query3)){

}

else{

echo mysql\_error();

}

}

header('Location: Profile.php');

}

}

else{

echo mysql\_error();

}

}

else{

echo mysql\_error();

}

On deleting a post, the particular post is simply removed from the posts table of that particular user. On liking a post, the liked attribute in the posts table of the user is set to true for that particular post and then the likes attribute is incremented by 1 for the particular post in the posts table. The setting of liked to true ensures that each user can like a particular post only once. The code for the above is :

if(isset($\_POST['like'])){

$email = getuserfield('Email',$\_SESSION['user\_id']);

$post\_id = $\_POST['post\_id'];

$query1 = "UPDATE `posts\_$email` set liked = 1 WHERE `post\_id` = $post\_id";

$query2 = "UPDATE `posts` set likes = likes + 1 WHERE `post\_id` = $post\_id";

if($query\_run1 = mysql\_query($query1)){

if($query\_run2 = mysql\_query($query2)){

header('Location: Profile.php');

}

else

echo mysql\_error();

}

else

echo mysql\_error();

}

else if(isset($\_POST['remove'])){

$email = getuserfield('Email',$\_SESSION['user\_id']);

$post\_id = $\_POST['post\_id'];

$query1 = "DELETE FROM `posts\_$email` WHERE `post\_id` = $post\_id";

if($query\_run1 = mysql\_query($query1)){

header('Location: Profile.php');

}

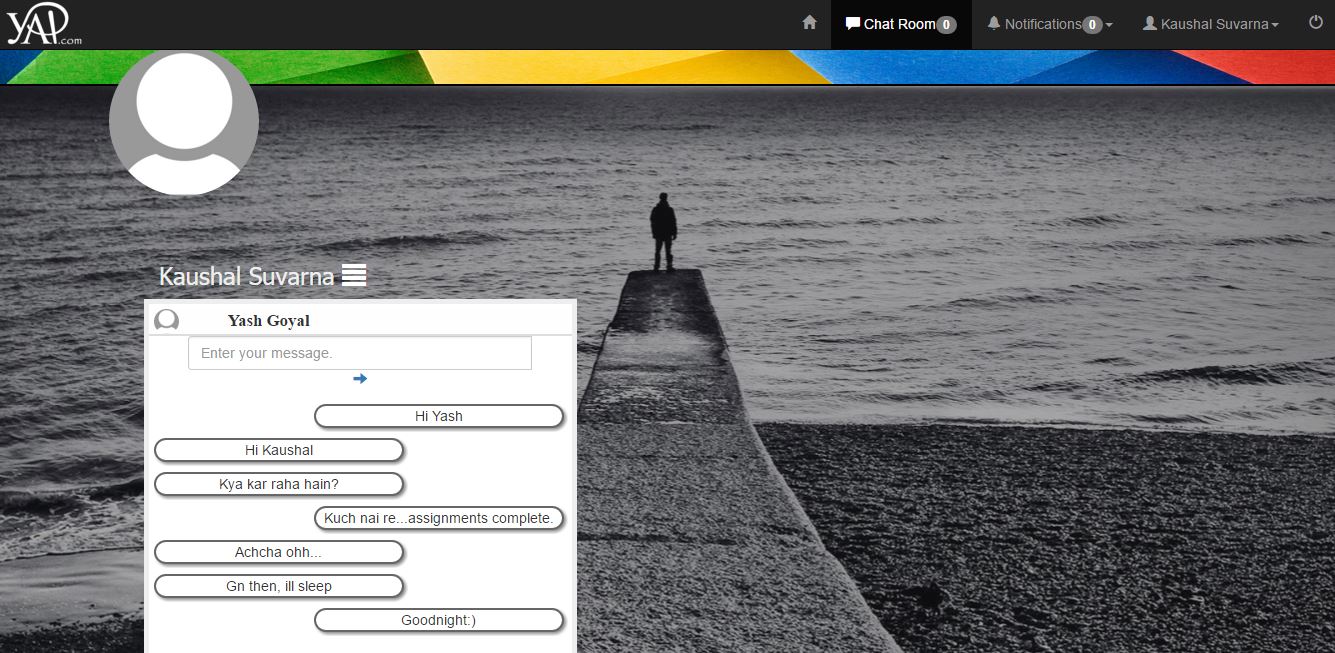
else

echo mysql\_error();

}

**The ChatRoom :**

**CHAT SERVICE**



The Chat service is a stand alone service which is successfully integrated

within other features of the project stated above.

This service enables two friends to chat with each other. The following softwares have been used for the same

 PHP

 HTML

 MySQL

 CSS

 Java Script

 XML

 AJAX

**OVERVIEW**

The basic overview of the chat service is based on the logic given

below:

1. When a user registers, he is generated with a unique ‘id’

(primary key).

2. This ‘id’ attribute is useful in successfully identifying the two

specific people chatting with each other

3. After retrieving the ‘id’ of both the users, use of the table ‘logs’

comes into picture.

4. The ‘logs’ table consists of attributes

‘id’,’username’,’msg’,’fromid’,’toid’,’readmsg’,’iname’ and

‘image’.

5. The ‘id’ of the logs table is the message id of each message(unique) and not same as the ‘id’ attribute of users table.

6. The data types of the attributes is given by DATA TYPES username VARCHAR(55)

fromid INT

readmsg SMALL INT(0 OR 1)

Image BLOB

id INT

msg TEXT

toid INT

iname VARCHAR(55)

7. The ‘msg’ attribute contains the information or message sent

by the user.

8. Fromid gives the ‘id’ of the sender.This is set when a user

logins.

9. The ‘toid’ attribute gives the ‘id’ of the receiver(to whom the

message is sent). This is set when the user selects the person

with whom the user is willing to chat

10. The ‘readmsg’ attribute is set to 0 by default.This indicates that

the message is unread by the recipient. If readmsg is 1 the message is read by the recipient.

11. The attributes ‘iname’ and ’image’ are usefull for sending

images over chat.

12. ‘iname’ is the name of the image and ‘image’ being the actual

image sent.

**WORKING**

The working of chat service can be described as follows:

 When the option of chatroom is selected the page to

 From this page the user can select the person to chat

 Also,the user can select person to chat with from the select the person with whom the user wants to chat opens up.with chat button next to his friend names in the list

 On selection ,the chatbox page opens up.This page shows a text box and a submit button and dispalys information “LOADING CHATLOGS..PLEASE WAIT “

 During this time the messages exchange between the specific users are loaded from the database

 Once the retrieval process is complete, the loading message on screen is replaced by the chats exchanged between the users.

 This updation of the page without reloading is done by javascript. This ensures that the users can be updated with the messages exchanged without refreshing the page everytime.

 The messages are updated using the file ‘logs.php’.

 This file is called every 20 ms.And if any new message is sent by the opposite user,are almost instantly loaded on the receivers screen ensuring fast receival of message.

setInterval(function(){$('#chatlogs').load('logs1.php');}, 2000);

 Note:Blank messages cannot be send.

 On typing a message and clicking the submit button, the file ‘insertion.php’ is called.

xmlhttp.open('GET','insertion.php?msg='+msg , true);

 In this file, the message is inserted into logs table using

proper ‘fromid’ and ’toid’ and setting defaut readmsg as 0.

 Since the sender has sent a message and the logs on his side need to be updated again to show the last message sent , thus the messages are are updated after the insertion takes place.

 The receiver is notified of unread message.

 If he reads the message them readmsg is set to 1.

MySQL Statements:

1. The entire tuple of a logged in user can be obtained using the SQL

statement

$result=mysql\_query("SELECT \* FROM users WHERE user\_id='$userid'");

$userRow=mysql\_fetch\_array($result);

Herewith, a variable is declared result which receives the tuple of the

logged in user.

The userRow variable fetches the entire array of the details present in

user table.

Thus any required information of the logged in user can be obtained by

using the fetched array.

Example: $uname = $userRow1['username'];

Thus retrieves the username of the user whose userid is determined at the

time of login.

2. Inserting the image in database:

$msg="\*img\*";

mysql\_query("INSERT INTO logs(username, msg, fromid ,toid ,iname,

image) VALUES('$uname', '$msg' ,'$userid', '$frndid', '$name',

'$image')");

This statement inserts the image selected in variable ‘$image’ in the

database.

3. Insertion of message in database

mysql\_query("INSERT INTO logs(`username`, `msg`,fromid ,toid)

VALUES('$uname', '$msg' ,'$userid', '$frndid')");

This statement inserts the message entered by user into the database.

4. Retreival of messages

$result1 = mysql\_query("SELECT \* FROM logs WHERE (fromid='$userid'

AND toid='$frndid') OR (fromid='$frndid' AND toid='$userid')");

while($extract = mysql\_fetch\_array($result1)){

echo $extract['msg'];

}

Here $result1 fetches all the message which are sent by the user to a

specific person or the messages he has received from that user.

This result is stored in array and used as looping conditions until all such

messages are retrieved.

These messages are printed.

5. Retrieval of images

The images are reterived from the database with the help of following

code:

echo '<img height="300" width="300"

src="data:image;base64,'.$extract['image'].' "> ';

The width and height of the image is fixed.

**References :**

* W3schools.com : Java Script , PHP , Bootstrap
* Stackoverflow.com : for features
* Google : To solve errors and queries.
* The newboston videos on youtube.com : for PHP, JavaScript, BootStrap &CSS.