Gworld- Social Network

A Report for the Evaluation 2&3 of Project No. 313

MANISH KUMAR

Admission No.: 16SCSE101446

Under the Supervision of

DR. MUKUND PRATAP SINGH

School of Computing Science and Engineering

Greater Noida, Uttar Pradesh

Winter 2019-2020

TABLE OF CONTENTS

Abstract

The purpose of Gworld Social Network is providing a social platform which will only for the university and collage students, where they can share their university moments, stay connected with friends. It will also a platform for the faculty member also which will stay connected with the students . Students can stay updated with whats happens in university. This platform will also provide the information and news of hackathon and others activity. This will also work as Emagazine of university, which show most famous student in the university, Most rated teacher in university etc.

Gworld social Network, as described above, can lead to error free, secure, reliable and fast social network. It can assist the user to concentrate on their other activities without missing any moments. Thus, it will help organization in better utilization of resources. The organization social activity can be maintained hassle free .

The aim is to digitalize social life of university by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that one will enjoy more their collage life.

Basically, the project manage the university social life in best way with more enjoyment and robustly.

It is also use for creating posting and viewing the others post, ideals, entries and blogs.

Introduction

The “Gworld Social Network” has been developed to override the problems prevailing in the share university activity. This software is supported to eliminate, and in some cases reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need for the organization to carry out operations in a smooth and effective manner. the project manage the university social life in best way with more enjoyment and robustly. It is also use for creating posting and viewing the others post, ideals, entries and blogs. Thus, by this all it proves it is user-friendly.

(i) Overall Description:

The main aim of this application is to provide a hassle-free accessing of the posted blogs, entries, topics etc. It also used for posting the blogs, editing the blogs, deleting the posted blogs etc. It is also used for viewing and posting the others one’s blogs/posts.

Every organization, whether big or small, has challenges to overcome and managing the information of Idea, Blogs, Entries, Content, Views. Every Online Blogging System has different Blogs needs therefore we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with will allow you to manage your workforce anytime, at all times. These systems will ultimately allow you to better manage resources.

(ii) Purpose :

The main purpose of the Gworld Social Network is to manage the details of Blogs, Idea, Topic, Entries, Views. It manages all the information about Blogs, Content, Views, Blogs. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the Blogs, Idea, Content, Topic. It tracks all the details about the Topics, Entries, Views.

Functionalities provided by Gworld are as follows:

- A sign-up process.
- A login form.
- Logout facility.
- Session control.
- User profile with upload thumbnail.
- Adding members as friend.
- Gworld social network also manage the Content details online for Entries details, Views details, Blogs.
- It tracks all the information of Idea, Content, Entries etc.
- Manage the information of Idea.
- Shows the information and description of the Blogs, Topics.
- Editing, adding, posting and updating of Records.

(iii) Motivation and Scope:

The gist of the idea is to digitalize the process of manually driven ideas, topics and blog sharing among the user, small or big organizations as well as small forums

And then to bake it into an web application to make it available for common people even to speed up process reducing further hassle and issues.

It may help collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Online Blogging System, social life of organization.

management & collection procedure will go on smoothly.

Our project aims is to socialized university environmen.

i.e. we have tried to computerize

various processes of social network.

- To utilize resources in an efficient manner by increasing their productivity through automation.
- It satisfies the user requirement.
- Be easy to understand by the user and operator
- Be easy to operate.

- Have a good user interface
- Be expandable

The proposed system has the following requirements:

- System needs store information about new entry of Blogs and posts.
- System needs to help the internal staff and students to keep connected with each other without any hindrance.
- System needs to maintain quantity record.
- System need to update and delete the record.
- System also needs a search area.
- It also needs a security system to prevent data.

Problem statement:

First we should know what is the work of a social network ?

The work of the social network is to meet the peoples which having same professional, cultural, similarity. It rather than this it also help to know the other culture of the different of peoples on the globe.

For example we can take LINKEDIN what is the work done by it . it is a social network which commonly focused on the corporate. It help to meet the professionals and the organizations.

I observed there is not any system purposed for the educational system on large scale. Which help students to keep connected with the university culture.

Proposed Model & & Existing system:

In this phase, a logical system is built which fulfills the given requirements. Design phase of software development deals with transforming the clients's requirements into a logically working system. Normally, design is performed in the following in the following two steps:

1. **Primary Design Phase:**

   In this phase, the system is designed at block level. The blocks are created on the basis of analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimizing the information flow between blocks. Thus, all activities which require more interaction are kept in one block.
2. **Secondary Design Phase:**

   In the secondary phase the detailed design of every block is performed. Which will provide the clear structure of the project.
3. **Final design phase:** In the final phase, the integration of all modules takes place after unit testing.
4. **Testing phase:** In this phase integrated module goes through testing process.

The general tasks involved in the design process are the following:

- Design various blocks for overall system processes.
- Design smaller, compact and workable modules in each block.
- Design various database structures.
- Specify details of programs to achieve desired functionality.
- Design the form of inputs, and outputs of the system.

- Perform documentation of the design.
- System reviews.

Implementation & Architecture:

User Interface Design:

User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system of logging into the system to the eventually presentation of desired inputs and outputs. The overall flow of screens and messages is called a dialogue.

The following steps are various guidelines for User Interface Design:

- The system user should always be aware of what to do next.
- The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.
- Message, instructions or information should be displayed long enough to allow the system user to read them.
- Use display attributes sparingly.
- Default values for fields and answered to be entered by the user should be speciied.
- A use should not be allowed to proceed without correcting an error.
- The system user should never get an operating system message or fatal error.
- Also priority has been provided to the User experience application.

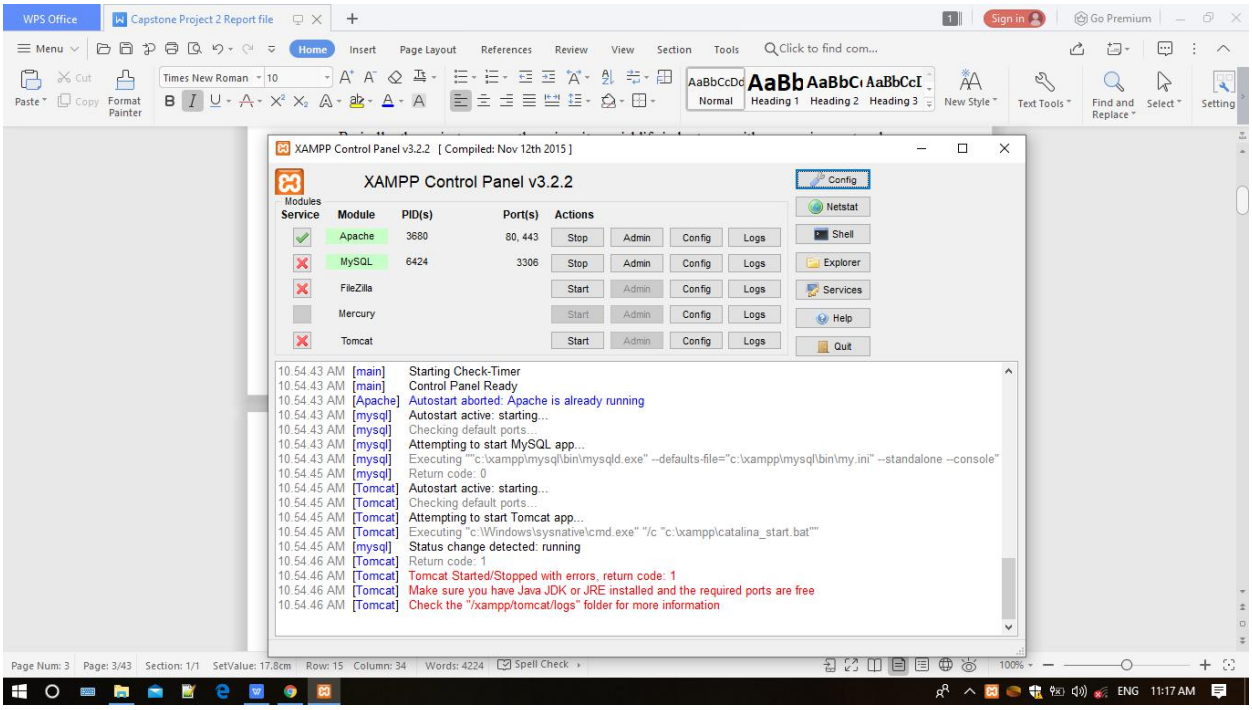
Architecture:

Details of the pages

Function.php

However, this file contains a little more than just the func- tions, as I have added the specifics of the database login here, rather than using another separate file. Thus, the first half-dozen lines of code identify the host, the name of the database, the username, and the password to use. It doesn't matter what you call the database, so long as it exists already.

Also make sure $dbuser and $dbpass are properly assigned to a MySQL username and password. The following two lines will open a connection to MySQL with correct values, and select the database. The last of the initial instructions sets the Social Networking site name by assigning the Robin's Nest value to the $appname variable. If you want to change your name, then this is the place to do it.



The Functions

The project uses five main functions:

createTable

Checks whether a table already exists and, if not, creates it

queryMysql

Issues a query to MySQL, outputting an error message if it fails destroySession

Destroys a PHP session and clears its data to log users out

sanitizeString

Removes potentially malicious code or tags from user input

showProfile - Displays a user's image and "about me" message if he has one.

```
<?php // Example 26-1: functions.php
 $dbhost  = 'localhost';    // Unlikely to require changing
 $dbname  = 'gworld';   // Modify these...
 $dbuser  = 'root';   // ...variables according  'robinsnest';
 $dbpass  = '';   // ...to your installation 'rnpassword';
 $appname = "GWORLD" ;

 $connection = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
 if ($connection->connect_error) die("Fatal Error");

 function createTable($name, $query)
 {
   queryMysql("CREATE TABLE IF NOT EXISTS $name($query)");
   echo "Table '$name' created or already exists.<br>";
 }

 function queryMysql($query)
```

```
{
  global $connection;
  $result = $connection->query($query);
  if (!$result) die("Fatal Error");
  return $result;
}

function destroySession()
{
  $_SESSION=array();

  if (session_id() != "" || isset($_COOKIE[session_name()]))
    setcookie(session_name(), '', time()-2592000, '/');

  session_destroy();
}

function sanitizeString($var)
{
  global $connection;
  $var = strip_tags($var);
  $var = htmlentities($var);
  if (get_magic_quotes_gpc())
    $var = stripslashes($var);
  return $connection->real_escape_string($var);
}

function showProfile($user)
{
```

```
  if (file_exists("$user.jpg"))
    echo "<img src='$user.jpg' style='float:left;'>";

  $result = queryMysql("SELECT * FROM profiles WHERE user='$user'");

  if ($result->num_rows)
  {
    $row = $result->fetch_array(MYSQLI_ASSOC);
    echo stripslashes($row['text']) . "<br style='clear:left;'><br>";
  }
  else echo "<p>Nothing to see here, yet</p><br>";
 }
?>
```

header.php

This is the file that the other files actually contain, which contains functions.php. This means that each file needs only a single require once. header.php starts by calling the function session_start.

Header.php

```
<?php // Example 26-2: header.php
 session_start();

echo <<<_INIT
<!DOCTYPE html>
<html>
 <head>
  <meta charset='utf-8'>
  <meta name='viewport' content='width=device-width, initial-scale=1'>
  <link rel='stylesheet' href='jquery.mobile-1.4.5.min.css'>
```

```
    <link rel='stylesheet' href='styles.css' type='text/css'>
    <script src='javascript.js'></script>
    <script src='jquery-2.2.4.min.js'></script>
    <script src='jquery.mobile-1.4.5.min.js'></script>
_INIT;
  require_once 'functions.php';
  $userstr = 'Welcome to Gworld Guys!';
  if (isset($_SESSION['user']))
  {
    $user     = $_SESSION['user'];
    $loggedin = TRUE;
    $userstr  = "Welcome : $user";
  }
  else $loggedin = FALSE;

echo <<<_MAIN
    <title>GWORLD: $userstr</title>
  </head>
  <body>
    <div data-role='page'  style="background-color:#DFF0D8;">
      <div data-role='header' style="background-color:#0030D7;">
        <div id='logo' style=><!-- <img id='glogo' src='logo.jpg' height=auto% width=5%> -->  Gworld</div>
        <div class='username'><font color='blue' style="background-color:white;">$userstr</font></div>
      </div>
      <div data-role='content'>
_MAIN;
  if ($loggedin)
  {
echo <<<_LOGGEDIN
```

```
    <div class='center'>
     <a data-role='button' data-inline='true' data-icon='home'
      data-transition="slide" href='members.php?view=$user'>Home</a>
     <a data-role='button' data-inline='true' data-icon='user'
      data-transition="slide" href='members.php'>Members</a>
     <a data-role='button' data-inline='true' data-icon='heart'
      data-transition="slide" href='friends.php'>Friends</a><br>
     <a data-role='button' data-inline='true' data-icon='mail'
      data-transition="slide" href='messages.php'>Messages</a>
     <a data-role='button' data-inline='true' data-icon='edit'
      data-transition="slide" href='profile.php'>Edit Profile</a>
     <a data-role='button' data-inline='true' data-icon='action'
      data-transition="slide" href='logout.php'>Log out</a>
    </div>
_LOGGEDIN;
 }
 else
 {
echo <<<_GUEST
_GUEST;
 }
?>
```

setup.php

If you want to build on this idea, you would most definitely need to add even more columns to those tables. If so, a MySQL DROP TABLE command may need to be issued before creating a table again.

setup.php

```
<!DOCTYPE html>
<html>
```

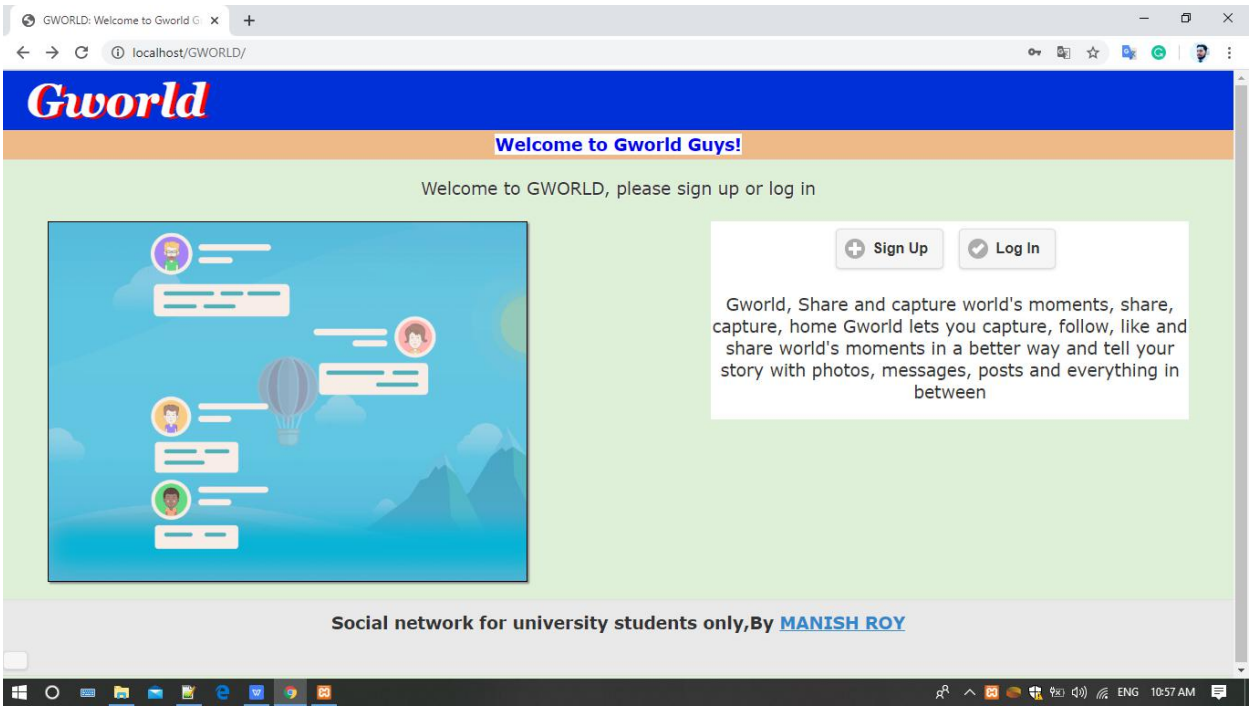
```
<head>
  <title>Setting up database</title>
</head>
<body>

  <h3>Setting up...</h3>

<?php // Example 26-3: setup.php
require_once 'functions.php';

createTable('members',
        'user VARCHAR(16),
        pass VARCHAR(16),
        INDEX(user(6))');

createTable('messages',
        'id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
        auth VARCHAR(16),
        recip VARCHAR(16),
        pm CHAR(1),
        time INT UNSIGNED,
        message VARCHAR(4096),
        INDEX(auth(6)),
        INDEX(recip(6))');

createTable('friends',
        'user VARCHAR(16),
```

```
        friend VARCHAR(16),
        INDEX(user(6)),
        INDEX(friend(6))');

createTable('profiles',
        'user VARCHAR(16),
        text VARCHAR(4096),
        INDEX(user(6))');
?>

  <br>...done.
 </body>
</html>
```

index.php

```
<?php // Example 26-4: index.php
 session_start();
 require_once 'header.php';

 echo "<div class='center'>Welcome to GWORLD,";

 if ($loggedin) echo " $user, you are logged in";
 else        echo ' please sign up or log in';

 echo <<<_END
    </div><br>
        <div style="width: 40%; float:left; height:40%; background:gray;  margin-left:30px;">
<img src="images/index_img/home1.gif" style="" width="100%" height="auto">
</div>
<div style="width: 40%; float:right; height:100%; background:white; margin-right:30px;">
```

```
<div class='center'>


        <a data-role='button' data-inline='true' data-icon='plus'

          data-transition="slide" href='signup.php'>Sign Up</a>

        <a data-role='button' data-inline='true' data-icon='check'

          data-transition="slide" href='login.php'>Log In</a>

     <!-- <p>Gworld, Share and capture world's moments, share, capture, home

  Gworld lets you capture, follow, like and share world's moments in a better way and tell your story with photos, messages, posts and everything in between</p>

 -->

                </div>


</div>

  </div>


   <div data-role="footer">

     <h4>Social network for university students only,By <a href='https/linkedin.com/in/manish-kumar-396291149'

     target='_blank'>MANISH ROY</i></h4>

   </div>

  </body>

</html>

_END;

?>
```

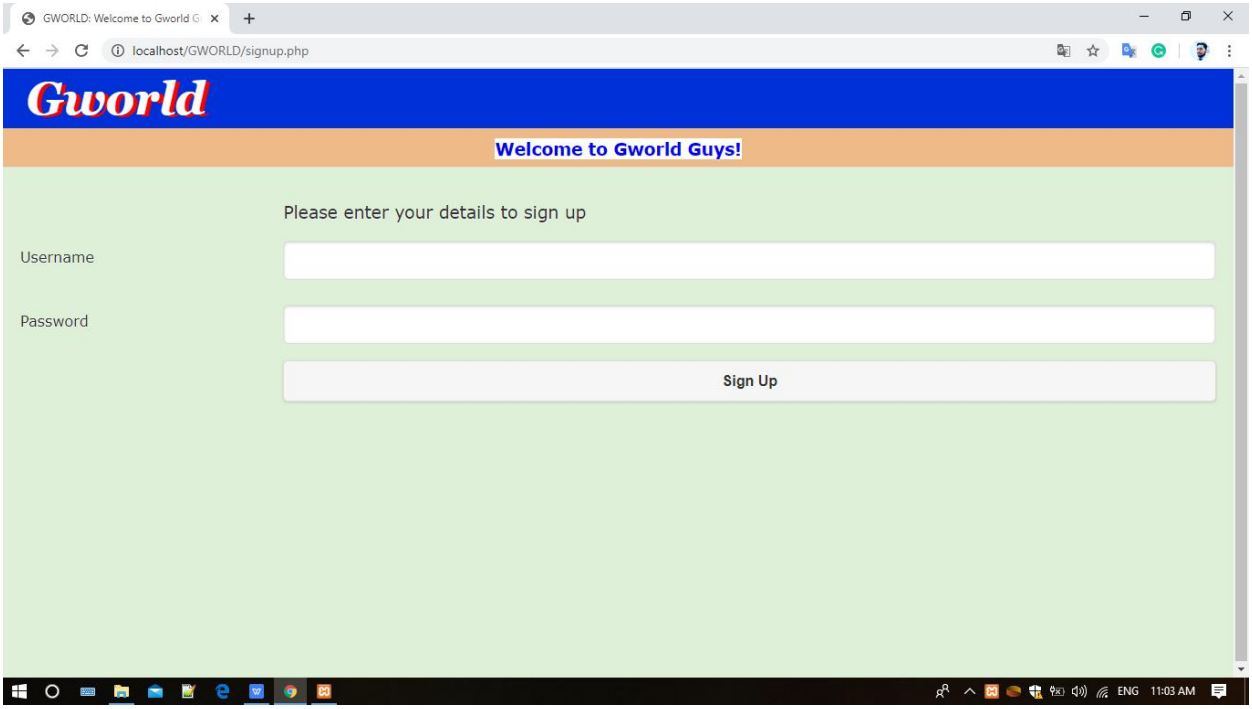


signup.php

We now need a module to allow users to join the new network, which is, signup.php. This is a slightly longer program but before you saw all its parts. Let's start by taking a look at the HTML end block. This is a simple form allowing input of a username and password. However, note the use of the empty span given the 'data' id.

In this program, this will be the destination of the Ajax call that checks if a desired username is available. For a detailed explanation of how this happens

works.



Checking for Username Availability

Now go back to program initialization and you will see a JavaScript block starting with the checkUser feature. This is called by the onBlur event JavaScript, when focus is removed from the form's username field. First it sets the contents of the span I mentioned (with the info I d) to an empty string, which will clear it in case it had a value previ- ously.

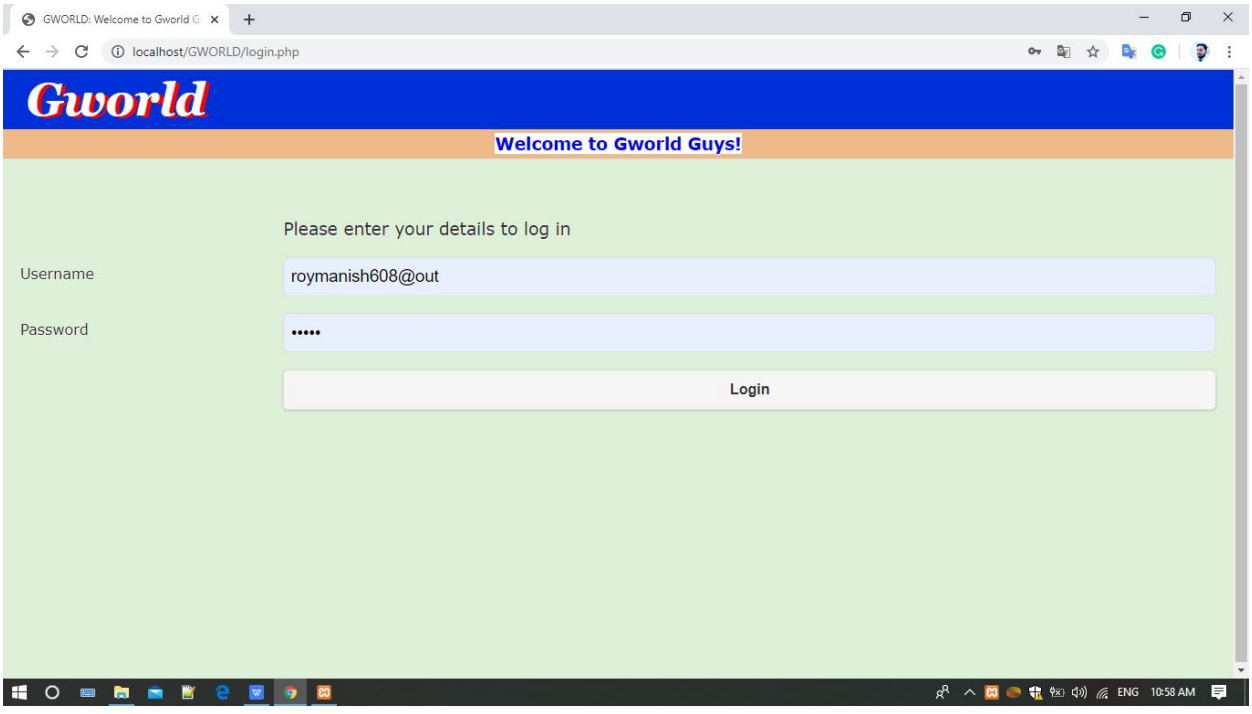
Logging

The user is then prompted to log in to Upon successful sign-up. At this point, a more fluid response might be to automatically log in to a newly created user, but as I don't want to overly complicate the code, I've kept the sign-up and login modules separate. However, if you wish, you can easily enforce this.

```
<?php // Example 26-7: login.php
 require_once 'header.php';
 $error = $user = $pass = "";

 if (isset($_POST['user']))
 {
   $user = sanitizeString($_POST['user']);
   $pass = sanitizeString($_POST['pass']);
```

```
  if ($user == "" || $pass == "")
    $error = 'Not all fields were entered';
  else
  {
    $result = queryMySQL("SELECT user,pass FROM members
      WHERE user='$user' AND pass='$pass'");

    if ($result->num_rows == 0)
    {
      $error = "Invalid login attempt";
    }
    else
    {
      $_SESSION['user'] = $user;
      $_SESSION['pass'] = $pass;
      die("<div class='center'>You are now logged in. Please
          <a data-transition='slide' href='members.php?view=$user'>click here</a>
          to continue.</div></div></body></html>");
    }
  }
 }

echo <<<_END
    <form method='post' action='login.php'>
      <div data-role='fieldcontain'>
        <label></label>
        <span class='error'>$error</span>
      </div>
      <div data-role='fieldcontain'>
        <label></label>
        Please enter your details to log in
      </div>
```

```
      <div data-role='fieldcontain'>
        <label>Username</label>
        <input type='text' maxlength='16' name='user' value='$user'>
      </div>
      <div data-role='fieldcontain'>
        <label>Password</label>
        <input type='password' maxlength='16' name='pass' value='$pass'>
      </div>
      <div data-role='fieldcontain'>
        <label></label>
        <input data-transition='slide' type='submit' value='Login'>
      </div>
    </form>
  </div>
 </body>
</html>
_END;
?>
```



checkuser.php

Here's to go with signup.php, which looks up a username in the database and returns a string indicating if it's already taken. Since it depends on the SanitizeString and queryMysql functions, the software contains the.php file functions first.

```
<?php // Example 26-6: checkuser.php

 require_once 'functions.php';


 if (isset($_POST['user']))

 {

  $user   = sanitizeString($_POST['user']);

  $result = queryMysql("SELECT * FROM members WHERE user='$user'");


  if ($result->num_rows)

   echo  "<span class='taken'> &#x2718; " .

        "The username '$user' is taken</span>";

  else

   echo "<span class='available'> &#x2714; " .

       "The username '$user' is available</span>";

 }

?>
```

profile.php

One of the first things new users may want to do after logging in and signing in is to create a profile that can be done through profile.php. I think you'll find some interesting code here, like uploading routines, redimensioning and sharpening images. Let's get started.

Adding the “About Me” Text

Then the text of the post variable is checked to see if any text has been posted to the program. If so, it will be sanitized, replacing all long white-space sequences (including returns and line feeds) with a single slot. This function includes a double security check to ensure that the user actually exists in the database and that no attempted hacking can be successful before this is inserted Text into the database, where the user's "about me" information would become. If no text has been posted, the database is queried to see if any text already exists to prepopulate the user's text area for editing.

Adding a Profile Image

Next we move on to the segment where the device variable $ FILES is tested to see if an image was uploaded. If so, a string variable called $saveto will be generated based on the username of the user followed by the.jpg extension. User Jill, for example, would cause $saveto to have Jill.jpg as value. This is the file where you save the uploaded image for use in the user's profile.

The uploaded image type is then scanned and accepted only if it is a jpeg, png, or gif image. Upon success, the $src variable will be filled with the uploaded image using one of the imagecreatefunctions according to the uploaded image type. The image is in a raw format which can be processed by PHP. If the picture is not of authorized form, the $typeok flag is set to FALSE,Impedes processing of the final section of the image upload code.

Processing the Image

First, we store the dimensions of the image in $w and $h using the following statement which is a easy way to assign values to different variables from an array: list($w, $h) = getimagesize($saveto);

Then, we calculate new dimensions using the value of $max (which is set to 100) which will result in a new image of the same ratio, but with no dimension greater than 100 pixels. This results in giving the new values required to the $tw and $th variables. If you want smaller or larger thumbnails just change the $max value accordingly.

Profile.php

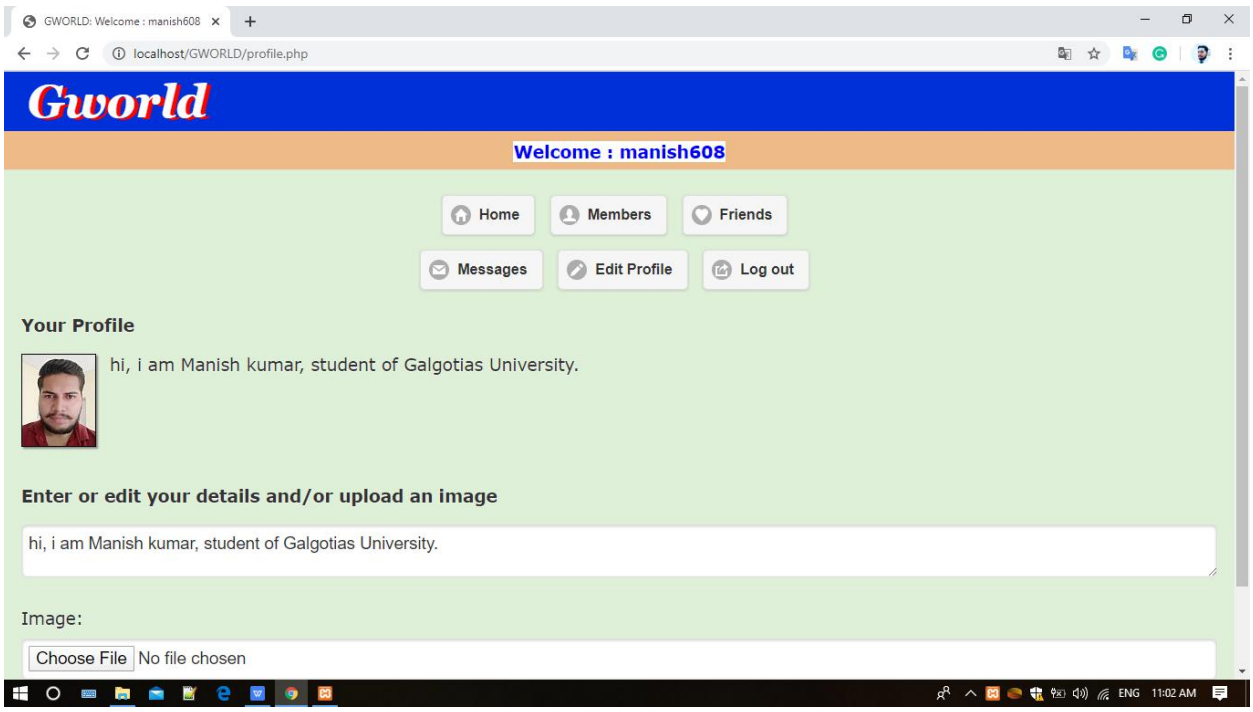
```
<?php // Example 26-8: profile.php

 require_once 'header.php';



 if (!$loggedin) die("</div></body></html>");



 echo "<h3>Your Profile</h3>";



 $result = queryMysql("SELECT * FROM profiles WHERE user='$user'");



 if (isset($_POST['text']))

 {
```

```
  $text = sanitizeString($_POST['text']);
  $text = preg_replace('/\s\s+/', ' ', $text);

  if ($result->num_rows)
      queryMysql("UPDATE profiles SET text='$text' where user='$user'");
  else queryMysql("INSERT INTO profiles VALUES('$user', '$text')");
}
else
{
  if ($result->num_rows)
  {
    $row  = $result->fetch_array(MYSQLI_ASSOC);
    $text = stripslashes($row['text']);
  }
  else $text = "";
}

$text = stripslashes(preg_replace('/\s\s+/', ' ', $text));

if (isset($_FILES['image']['name']))
{
  $saveto = "$user.jpg";
  move_uploaded_file($_FILES['image']['tmp_name'], $saveto);
```

```
$typeok = TRUE;


switch($_FILES['image']['type'])
{
  case "image/gif":   $src = imagecreatefromgif($saveto); break;
  case "image/jpeg":  // Both regular and progressive jpegs
  case "image/pjpeg": $src = imagecreatefromjpeg($saveto); break;
  case "image/png":   $src = imagecreatefrompng($saveto); break;
  default:            $typeok = FALSE; break;
}


if ($typeok)
{
  list($w, $h) = getimagesize($saveto);

  $max = 100;
  $tw  = $w;
  $th  = $h;

  if ($w > $h && $max < $w)
  {
    $th = $max / $w * $h;
    $tw = $max;
```

```
    }
    elseif ($h > $w && $max < $h)
    {
      $tw = $max / $h * $w;
      $th = $max;
    }
    elseif ($max < $w)
    {
      $tw = $th = $max;
    }

    $tmp = imagecreatetruecolor($tw, $th);
    imagecopyresampled($tmp, $src, 0, 0, 0, 0, $tw, $th, $w, $h);
    imageconvolution($tmp, array(array(-1, -1, -1),
      array(-1, 16, -1), array(-1, -1, -1)), 8, 0);
    imagejpeg($tmp, $saveto);
    imagedestroy($tmp);
    imagedestroy($src);
  }
}

showProfile($user);
```

```
echo <<<_END

    <form data-ajax='false' method='post'

     action='profile.php' enctype='multipart/form-data'>

    <h3>Enter or edit your details and/or upload an image</h3>

    <textarea name='text'>$text</textarea><br>

    Image: <input type='file' name='image' size='14'>

    <input type='submit' value='Save Profile'>

    </form>

  </div><br>

 </body>

</html>

_END;

?>
```



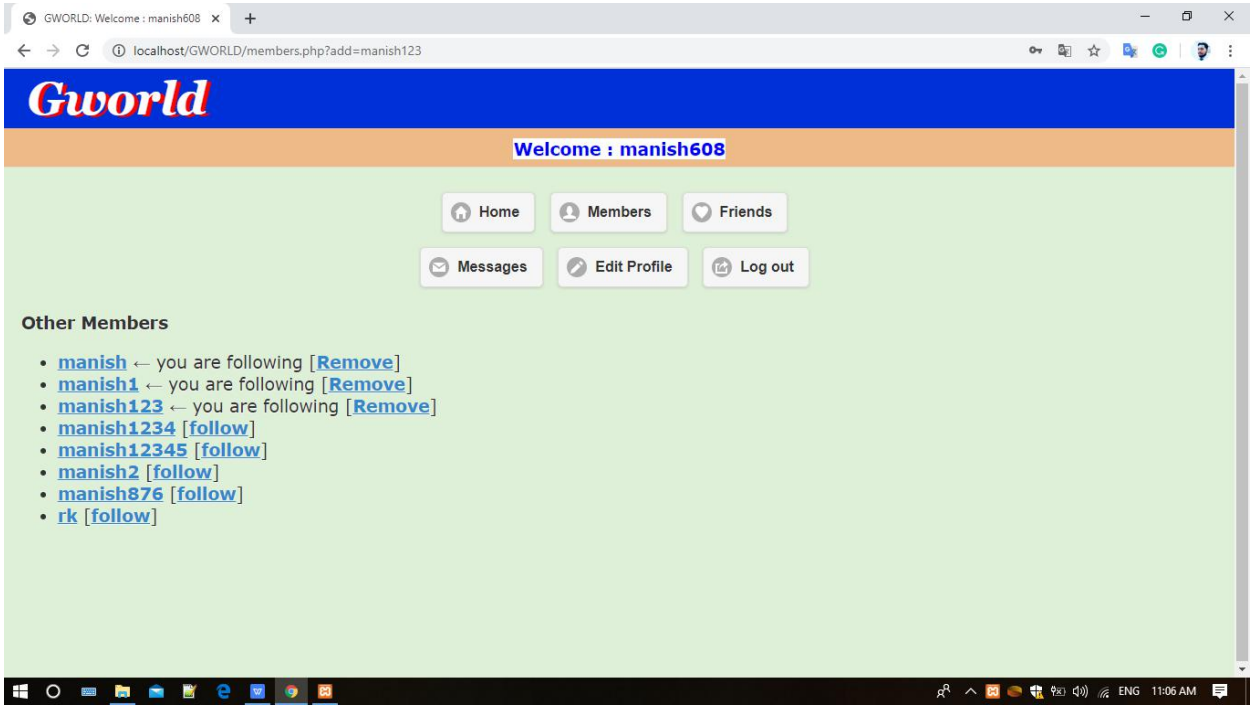
members.php

```
<?php // Example 26-9: members.php

 require_once 'header.php';



 if (!$loggedin) die("</div></body></html>");



 if (isset($_GET['view']))

 {

  $view = sanitizeString($_GET['view']);



  if ($view == $user) $name = "Your";

  else                $name = "$view's";



  echo "<h3>$name Profile</h3>";

  showProfile($view);

  echo "<a data-role='button' data-transition='slide'

        href='messages.php?view=$view'>View $name messages</a>";

  die("</div></body></html>");

 }



 if (isset($_GET['add']))
```

```
{
  $add = sanitizeString($_GET['add']);

  $result = queryMysql("SELECT * FROM friends WHERE user='$add' AND friend='$user'");
  if (!$result->num_rows)
    queryMysql("INSERT INTO friends VALUES ('$add', '$user')");
}
elseif (isset($_GET['remove']))
{
  $remove = sanitizeString($_GET['remove']);
  queryMysql("DELETE FROM friends WHERE user='$remove' AND friend='$user'");
}

$result = queryMysql("SELECT user FROM members ORDER BY user");
$num    = $result->num_rows;

echo "<h3>Other Members</h3><ul>";

for ($j = 0 ; $j < $num ; ++$j)
{
  $row = $result->fetch_array(MYSQLI_ASSOC);
```

```
if ($row['user'] == $user) continue;


echo "<li><a data-transition='slide' href='members.php?view=" .
  $row['user'] . "'>" . $row['user'] . "</a>";
$follow = "follow";


$result1 = queryMysql("SELECT * FROM friends WHERE
  user='" . $row['user'] . "' AND friend='$user'");
$t1      = $result1->num_rows;
$result1 = queryMysql("SELECT * FROM friends WHERE
  user='$user' AND friend='" . $row['user'] . "'");
$t2      = $result1->num_rows;


if (($t1 + $t2) > 1) echo " &harr; is your friend";
elseif ($t1)         echo " &larr; you are following";
elseif ($t2)       { echo " &rarr; is following you";
                     $follow = "Add"; }


if (!$t1) echo " [<a data-transition='slide'
  href='members.php?add=" . $row['user'] . "'>$follow</a>]";
else      echo " [<a data-transition='slide'
```

```
    href='members.php?remove=" . $row['user'] . "'>Remove</a>]";
  }
?>
   </ul></div>
  </body>
</html>
```



friends.php

The module showing friends and followers of a user is, friends.php. This interrogates the table of friends just like the program members.php but only for a single user. It then shows all the friends and followers of that user together with the people he follows

```
<?php // Example 26-10: friends.php

  require_once 'header.php';
```

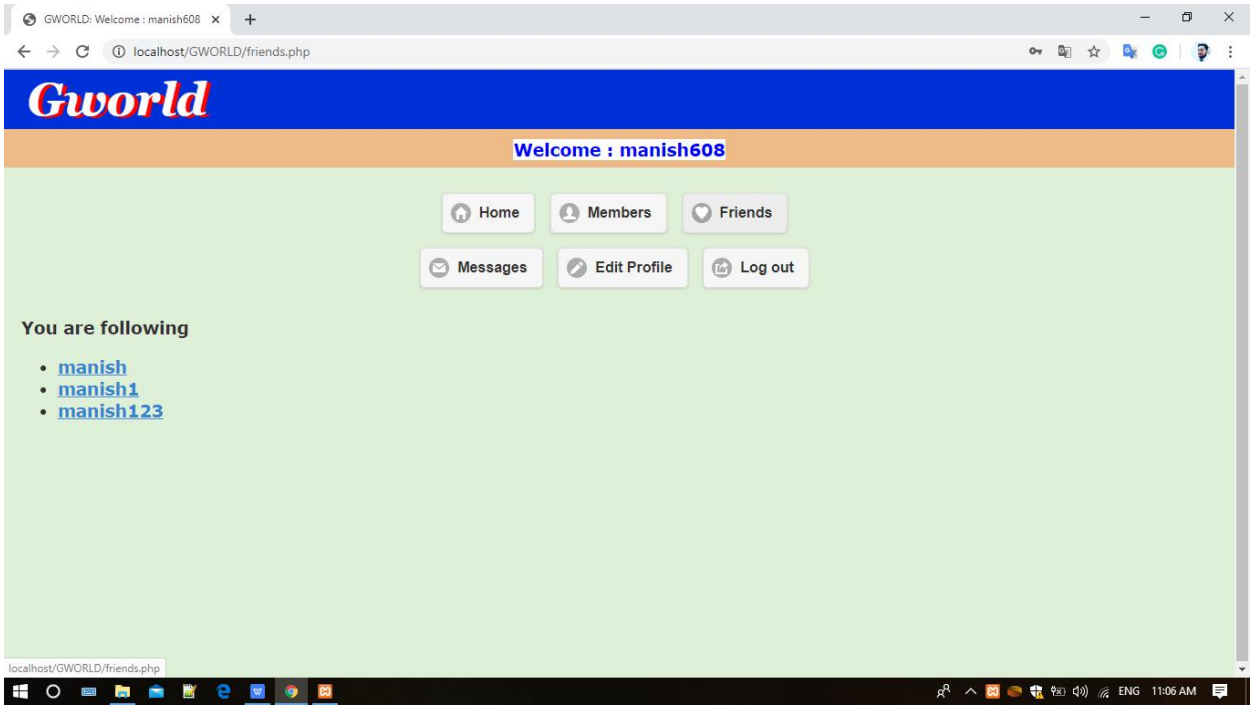
```
if (!$loggedin) die("</div></body></html>");


if (isset($_GET['view'])) $view = sanitizeString($_GET['view']);
else                      $view = $user;


if ($view == $user)
{
  $name1 = $name2 = "Your";
  $name3 =          "You are";
}
else
{
  $name1 = "<a data-transition='slide'
          href='members.php?view=$view'>$view</a>'s";
  $name2 = "$view's";
  $name3 = "$view is";
}


// Uncomment this line if you wish the user's profile to show here
// showProfile($view);
```

```
$followers = array();

$following = array();



$result = queryMysql("SELECT * FROM friends WHERE user='$view'");

$num    = $result->num_rows;



for ($j = 0 ; $j < $num ; ++$j)

{

  $row          = $result->fetch_array(MYSQLI_ASSOC);

  $followers[$j] = $row['friend'];

}



$result = queryMysql("SELECT * FROM friends WHERE friend='$view'");

$num    = $result->num_rows;



for ($j = 0 ; $j < $num ; ++$j)

{

    $row          = $result->fetch_array(MYSQLI_ASSOC);

    $following[$j] = $row['user'];

}
```

```
$mutual    = array_intersect($followers, $following);
$followers = array_diff($followers, $mutual);
$following = array_diff($following, $mutual);
$friends   = FALSE;

echo "<br>";

if (sizeof($mutual))
{
  echo "<span class='subhead'>$name2  friends</span><ul>";
  foreach($mutual as $friend)
    echo "<li><a data-transition='slide'
         href='members.php?view=$friend'>$friend</a>";
  echo "</ul>";
  $friends = TRUE;
}

if (sizeof($followers))
{
  echo "<span class='subhead'>$name2 followers</span><ul>";
```

```
  foreach($followers as $friend)
    echo "<li><a data-transition='slide'
          href='members.php?view=$friend'>$friend</a>";
  echo "</ul>";
  $friends = TRUE;
 }

 if (sizeof($following))
 {
  echo "<span class='subhead'>$name3 following</span><ul>";
  foreach($following as $friend)
    echo "<li><a data-transition='slide'
          href='members.php?view=$friend'>$friend</a>";
  echo "</ul>";
  $friends = TRUE;
 }

 if (!$friends) echo "<br>You don't have any friends yet.";
?>
   </div><br>
 </body>
```

</html>



Message.php

<?php // Example 26-11: messages.php

require_once 'header.php';

if (!$loggedin) die(" </body></html>");

if (isset($_GET['view'])) $view = sanitizeString($_GET['view']);

else $view = $user;

if (isset($_POST['text']))

{

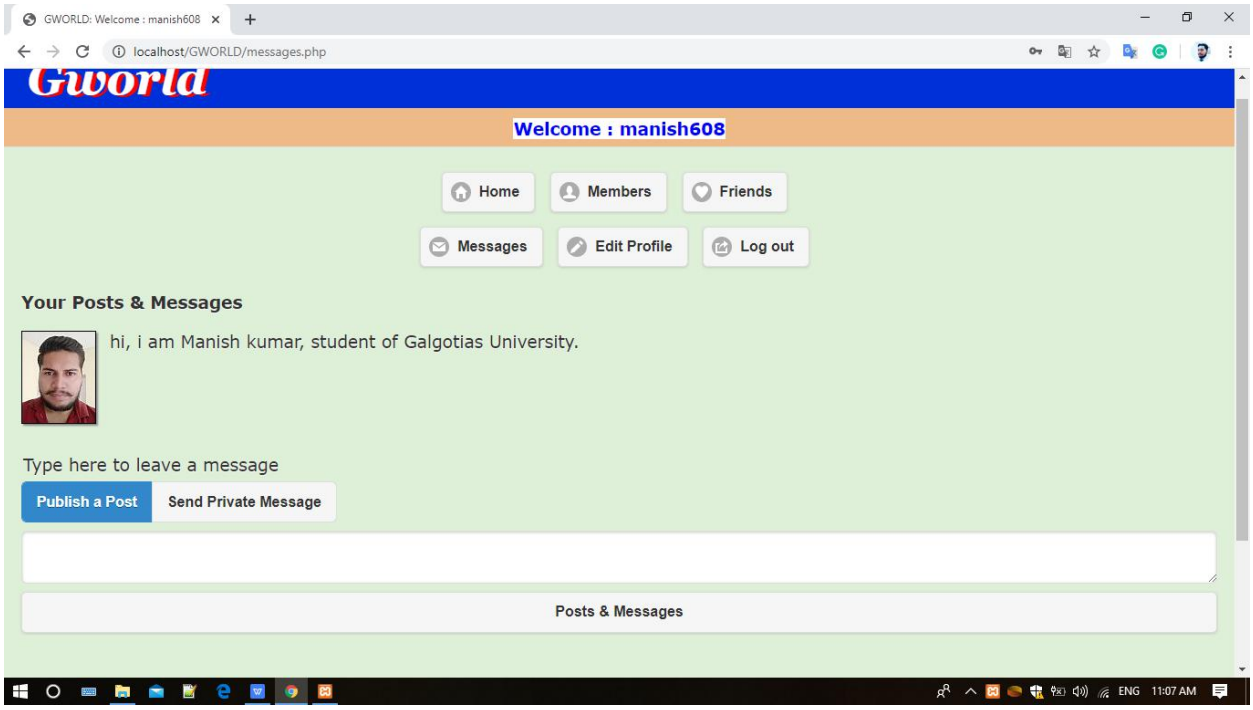
```
  $text = sanitizeString($_POST['text']);


  if ($text != "")
  {
    $pm   = substr(sanitizeString($_POST['pm']),0,1);
    $time = time();
    queryMysql("INSERT INTO messages VALUES(NULL, '$user',
      '$view', '$pm', $time, '$text')");
  }
}


if ($view != "")
{
  if ($view == $user) $name1 = $name2 = "Your";
  else
  {
    $name1 = "<a href='members.php?view=$view'>$view</a>'s";
    $name2 = "$view's";
  }


  echo "<h3>$name1 Posts & Messages</h3>";
```

```
  showProfile($view);


  echo <<<_END
    <form method='post' action='messages.php?view=$view'>
      <fieldset data-role="controlgroup" data-type="horizontal">
        <legend>Type here to leave a message</legend>
        <input type='radio' name='pm' id='public' value='0' checked='checked'>
        <label for="public">Publish a Post</label>
        <input type='radio' name='pm' id='private' value='1'>
        <label for="private">Send Private Message</label>
      </fieldset>
    <textarea name='text'></textarea>
    <input data-transition='slide' type='submit' value='Posts & Messages'>
  </form><br>
_END;


  date_default_timezone_set('UTC');


  if (isset($_GET['erase']))
  {
    $erase = sanitizeString($_GET['erase']);
```

```
  queryMysql("DELETE FROM messages WHERE id=$erase AND recip='$user'");
}

$query  = "SELECT * FROM messages WHERE recip='$view' ORDER BY time DESC";
$result = queryMysql($query);
$num    = $result->num_rows;

for ($j = 0 ; $j < $num ; ++$j)
{
  $row = $result->fetch_array(MYSQLI_ASSOC);

  if ($row['pm'] == 0 || $row['auth'] == $user || $row['recip'] == $user)
  {
    echo date('M jS \'y g:ia:', $row['time']);
    echo " <a href='messages.php?view=" . $row['auth'] .
         "'>" . $row['auth']. "</a> ";

    if ($row['pm'] == 0)
      echo "Public Post: &quot;" . $row['message'] . "&quot; ";
    else
      echo "Private Message: <span class='whisper'>&quot;" .
```

```
        $row['message']. "&amp;quot;</span> ";


      if ($row['recip'] == $user)
        echo "[<a href='messages.php?view=$view" .
          "&erase=" . $row['id'] . "'>erase</a>]";

      echo "<br>";
    }
  }
}

if (!$num)
  echo "<br><span class='info'>No messages yet</span><br><br>";

echo "<br><a data-role='button'
      href='messages.php?view=$view'>Refresh messages</a>";
?>

  </div><br>
</body>
```

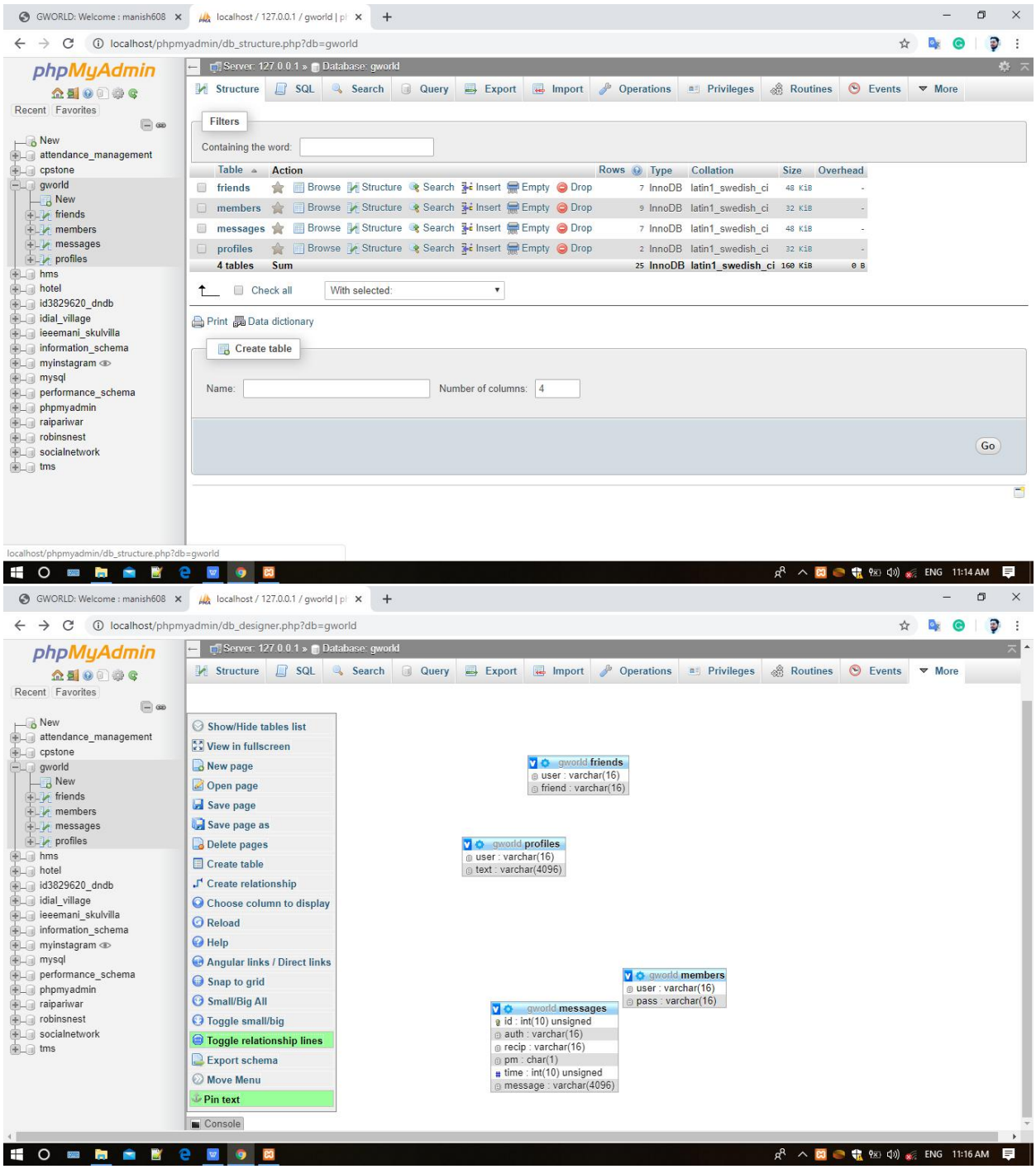
</html>



logout.php

The final ingredient in our logout.php social networking recipe, the logout page which closes a session and removes any associated data and cookies. The consequence of calling up this software, where the user is now asked to select a button to take her to the home page un-logged-in and delete the links from the top of the screen Of course, To do this, you might write a JavaScript or PHP redirect (probably a good idea if you want to keep your logo looking clean).

Database:



Conclusion & future enhancement: The project's future enhancement will contain some improved UI and features some rich features such as university blogs .

References:

- **PHP documents:** Urls https://docs.php.org
- **W3school : url:** https://w3school.com
- **PHP quick installation :** https://docs.php.org/en/7.0/intro/install/