

PHP –Developing a Social Media Network with Image processing

Assignment Tasks 01 (Signup and Sign in):

Program for Signup Page – In order to sign up to use the social media site, the user must enter valid credentials, which are validated, and when valid, upon submission, these credentials are added to a Database. Below shows the entire program made for this task.

PHP validation of the signup form: (loginPage.php)

```
1 <?php
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <link rel="stylesheet" href="home.css"/>
6 <title></title>
7 </head>
8 <body>
9 <?php
10 global $nameErr;
11 $nameErr = "";
12 global $passErr;
13 $passErr = "";
14 global $fname;
15 $fnameErr = "";
16 global $emailErr;
17 $emailErr = "";
18
19 global $name, $pass, $email, $fname;
20
21 if ($_SERVER["REQUEST_METHOD"] == "POST") {
22     if (empty($_POST["name"])) {
23         $nameErr = "User Name is Required";
24     } else {
25         $name = test_input($_POST["name"]);
26
27         if (!preg_match("/^[a-zA-Z0-9]*$/", $_POST["name"])) {
28             $nameErr = "Only letters and numbers allowed";
29         }
30     }
31
32     if (empty($_POST["pass"])) {
33         $passErr = "Password is Required";
34     } else {
35         $pass = test_input($_POST["pass"]);
36         if (!preg_match("/^[a-zA-Z0-9]*$/", $_POST["pass"])) {
37             $passErr = "Only letters and numbers allowed";
38         }
39     }
40
41     if (empty($_POST["fname"])) {
42         $fnameErr = "User Name is Required";
43     } else {
44         $fname = test_input($_POST["fname"]);
45
46         if (!preg_match("/^[a-zA-Z- ']*$/", $_POST["fname"])) {
47             $fnameErr = "Only letters and whitespaces allowed";
48         }
49     }
50
51     if (empty($_POST["email"])) {
52         $emailErr = "Email is Required";
53     } else {
54         $email = test_input($_POST["email"]);
55         if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
56             $emailErr = "Invalid Format";
57         }
58     }
59 }
60
61 function test_input($data) {
62     $data = trim($data);
63     $data = stripslashes($data);
64     $data = htmlspecialchars($data);
65     return $data;
66 }
67
68 ?>
```

HTML form:

```

68
69
70 <div id="form">
71 <h3> Sign up: </h3>
72 <h5>Enter all the details below to create an account</h5>
73 <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>"
74 <p><span class="error">* Required Field</span></p>
75
76 User Name: <input type="text" name="name">
77 <span class="error"><?php echo $nameErr; ?></span>
78 <br><br>
79 Password: <input type="text" name="pass">
80 <span class="error"><?php echo $passErr; ?></span>
81 <br><br>
82 Full Name: <input type="text" name="fname">
83 <span class="error"><?php echo $fnameErr; ?></span>
84 <br><br>
85 E-Mail: <input type="text" name="email">
86 <span class="error"><?php echo $emailErr; ?></span>
87 <br><br>
88 <input type="submit" name="check" value="Verify">
89 <input type="submit" name="submit">
90 </form>
91 </div>
92

```

PHP inserting validated form info into database (with password hashing)

```

105 <?php
106 if (isset($_POST["submit"])) {
107     $name = $_POST["name"];
108     $pass = $_POST["pass"];
109     $fname = $_POST["fname"];
110     $email = $_POST["email"];
111     session_start();
112
113     $servername = "localhost:3311";
114     $username = "lorenas";
115     $password = "lorenas";
116     $dbname = 'social';
117     $login = 0;
118
119     try {
120         $conn = new PDO("mysql:host=$servername; dbname=$dbname", $username, $password);
121         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
122
123         $hashedpassword = hash("sha256", $_POST["pass"]);
124         $sql = "INSERT INTO `users` (`ID`, `UserName`, `Password`, `Name`, `Email`) "
125             . "VALUES (NULL, '$name', '$hashedpassword', '$fname', '$email')";
126
127         $conn->exec($sql);
128
129         echo "<h3>Thank you for signing up, continue to login page.</h3>";
130         echo "<h3><a href='signinPage.php'>Login Here</a><h3>";
131
132     } catch(PDOException $e) {
133         echo "Failed: " . $e->getMessage();
134     }
135     $conn = null;
136 }
137

```

Top navigation element – Same on every page:

```

<header>
<div class="navbar">
<a style="text" href="home.php">Home</a><br><br><br><br>
<a href="friend.php">Friends</a><br><br><br><br>
<a href="image.php">Images</a><br><br><br><br>
<a href="profile.php">Profile</a><br><br><br><br>
</div>
</header>

```

Database (From phpMyAdmin)

ID	UserName	Password	Name	Email
1	Lorena02	a581822c19a39b3917ee9c5b4351669442c65cfa958907e525...	Lorena Spallino	lorena1@gmail.com
2	Sofia04	86846e84f9bfb2053a4823193843d22a0be24a75d304d6f8a2...	Sofia M Spallino	sofia@yahoo.com

Complete Output:

The following screenshots display the output of the above program and displays how the user would use this aspect of the application. CSS was used to make the output look interesting.

Page to sign up:

After signing up with valid credentials:

When the user clicks submit, inputs are entered into the database, and the success message is shown.

Upon clicking the Login Here link, user is directed to the next page, in order to log into the application.

Thank you for signing up, continue to login page.

[Login Here](#)

Login Page Program – (signinPage.php). The below screenshots display the entire program for the logging in page, which users would use to log into the application with their respective accounts.

Form validation program – The form users must enter to log in, is first validated:

```

1 <html>
2 <head>
3 <meta charset="UTF-8">
4 <link rel="stylesheet" href="home.css"/>
5 <title></title>
6
7 </head>
8 <body>
9
10 <?php
11 global $nameErr;
12 $nameErr = "";
13 global $passErr;
14 $passErr = "";
15 global $fname;
16 $fnameErr = "";
17 global $emailErr;
18 $emailErr = "";
19
20 global $name, $pass, $email, $fname;
21
22 if ($_SERVER["REQUEST_METHOD"] == "POST") {
23     if (empty($_POST["name"])) {
24         $nameErr = "User Name is Required";
25     } else {
26         $name = test_input($_POST["name"]);
27         if (!preg_match("/^[a-zA-Z0-9]*$/", $_POST["name"])) {
28             $nameErr = "Only letters and numbers allowed";
29         }
30     }
31 }

```

```

31 |
32 |         if(empty($_POST["pass"])){
33 |             $priceErr = "Price is Required";
34 |         } else {
35 |             $pass = test_input($_POST["pass"]);
36 |             if(!preg_match("/^[a-zA-Z0-9]*$/", $_POST["pass"])) {
37 |                 $nameErr = "Only letters and numbers allowed";
38 |             }
39 |         }
40 |     }
41 |
42 | }
43 |
44 | function test_input($data) {
45 |     $data = trim($data);
46 |     $data = stripslashes($data);
47 |     $data = htmlspecialchars($data);
48 |     return $data;
49 | }
50 |
51 | }

```

HTML Form:

```

52 | <header>
53 |
54 | <div class="navbar">
55 |     <a style="text-decoration: none;" href="home.php">Home</a><br><br><br>
56 |     <a href="">Friends</a><br><br><br>
57 |     <a href="">Images</a><br><br><br>
58 |     <a href="">Profile</a><br><br><br>
59 |
60 | </div>
61 |
62 | </header>
63 | <h2 id="nav">What's Up <?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?></h2>
64 |
65 | <body>
66 |     <div id="form">
67 |         <h3> Sign in: </h3>
68 |         <h5>Don't have an Account? Sign up here</h5>
69 |         <a href="loginPage.php">Sign Up Here</a>
70 |         <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
71 |             <p><span class="error"><?php echo $nameErr;?></span> Required Field</p>
72 |
73 |             User Name: <input type="text" name="name">
74 |             <span class="error"><?php echo $nameErr;?></span>
75 |             <br><br>
76 |             Password: <input type="text" name="pass">
77 |             <span class="error"><?php echo $passErr;?></span>
78 |             <br><br>
79 |             <input type="submit" name="check" value="Verify">
80 |             <input type="submit" name="submit">
81 |
82 |         </form>
83 |     </div>

```

PHP checking valid login info from database: - if login is correct, sessions and cookies are started with user's credentials, and user is directed to the site page, if not there is an error message telling user to try again.

```

84 | <?php
85 |
86 |
87 |
88 | if (isset($_POST["submit"])) {
89 |     //echo " <h2> Inserting into DataBase:</h2>";
90 |
91 |     $name = $_POST["name"];
92 |     $pass = $_POST["pass"];
93 |
94 |
95 |     $servername = "localhost:3311";
96 |     $username = "lorenaS";
97 |     $password = "lorenaS";
98 |     $dbname = 'social';
99 |     $login = 0;
100 |
101 |     try {
102 |         $conn = new PDO("mysql:host=$servername; dbname=$dbname", $username, $password);
103 |
104 |         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
105 |         //echo "Made the connection";
106 |
107 |         //hashing the PW
108 |         $hashedpassword = hash("sha256", $_POST["pass"]);
109 |
110 |         $sql1 = "SELECT * FROM user WHERE UserName='$name' AND Password='$hashedpassword'";
111 |         $login = 0;
112 |         $q1 = $conn->query($sql1);
113 |
114 |         $q1->setFetchMode(PDO::FETCH_ASSOC);
115 |

```

```

$login = 1;
session_start();
//Setting Session Variables
$sfname = $row["Name"];
$email = $row["Email"];
$id = $row["ID"];

$_SESSION["username"] = "$fname";
$_SESSION["password"] = "$pass";
$_SESSION["fullname"] = "$sfname";
$_SESSION["email"] = "$email";
$_SESSION["userid"] = "$id";

}

}

if(isset($_POST["submit"]) && $login == 0){
    echo "<h1 class='error'>invalid username or password, please try again</h1>";
}

} catch(PDOException $e) {
    echo "Failed: " . $e->getMessage();
}

$conn = null;

}

?>
</body>
</html>

```

Output of login page:

The below screenshots show the entirety of the outputs of this page. This displays how the user can login if they have already created an account.

Example of invalid username or password (error message shown):

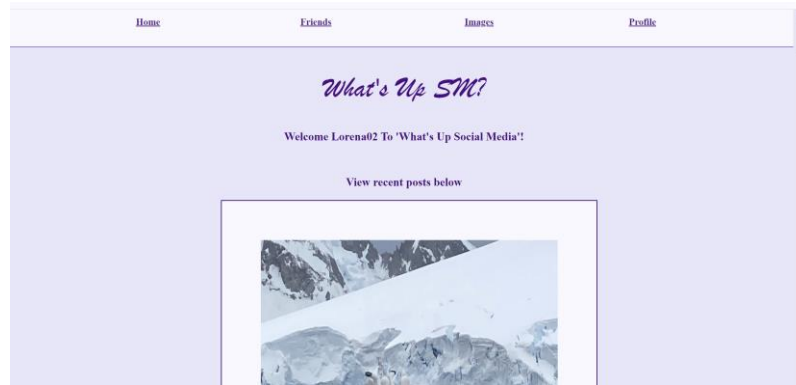
invalid username or password, please try again

Example of valid login info – if user's credentials are in the database, they will be successfully logged in:

Login Successful, Hello Lorena02

[Please continue to site page](#)

The link above directs user to home page, which looks like the screenshot below:



Assignment Tasks 02 (Edit User Profile data):

After logging in with username and password, user data is stored in Session variables and cookies: then the profile.php page programming looks like the following screenshots In order to display the user's data.

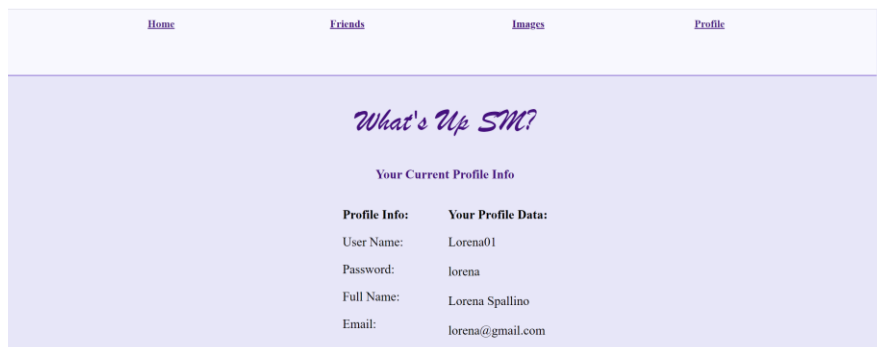
Profile.php Displaying user's data from the database, and using sessions and cookies enabled upon login:

```

1 <html>
2 <head>
3   <meta charset="UTF-8">
4   <title></title>
5   <link rel="stylesheet" href="home.css"/>
6 </head>
7 <body>
8   <header>
9     <div class="navbar">
10      <a style="text" href="home.php">Home</a><br><br><br><br>
11      <a href="products.htm">Friends</a><br><br><br><br>
12      <a href="cart.htm">Images</a><br><br><br><br>
13      <a href="profile.php">Profile</a><br><br><br><br>
14    </div>
15  </header>
16  <h2 id="nav">What's Up SM?</h2>
17  <?php
18    session_start();
19    echo "<h3 id='blue'>Your Current Profile Info</h3>";
20
21  ?>
22
23  <div id='div1'>
24    <h3>Profile Info: </h3>
25    <p> User Name: </p>
26    <p> Password: </p>
27    <p> Full Name: </p>
28    <p> Email: </p>
29  </div>
30
31  <div id='div2'>
32    <h3> Your Profile Data: </h3>
33    <?php
34      echo $_SESSION["username"] . "<br>";
35      echo "<br>";
36      echo $_SESSION["password"] . "<br>";
37      echo "<br>";
38      echo $_SESSION["fullname"] . "<br>";
39      echo "<br>";
40      echo $_SESSION["email"] . "<br>";
41    ?>

```

Output of this part of the profile.php page looks like the following screenshot (logged in as a user called Lorena01):



PHP form validation for the forms that allow the user to update their information. (No more required attributes, only data validation to ensure web security)

```

42 | <?php
43 | global $nameErr;
44 | $nameErr = "";
45 | global $passErr;
46 | $passErr = "";
47 | global $fname;
48 | $fnameErr = "";
49 | global $emailErr;
50 | $emailErr = "";
51 |
52 | global $name, $pass, $email, $fname;
53 |
54 | if ($_SERVER["REQUEST_METHOD"] == "POST") {
55 |     if (empty($_POST["name"])) {
56 |         $nameErr = "Name is required";
57 |     } else {
58 |         $name = test_input($_POST["name"]);
59 |         if (!preg_match("/^[a-zA-Z-0-9]*$/", $_POST["name"])) {
60 |             $nameErr = "Only letters and numbers allowed";
61 |         }
62 |     }
63 |
64 |     if (empty($_POST["pass"])) {
65 |         $passErr = "Password is required";
66 |     } else {
67 |         $pass = test_input($_POST["pass"]);
68 |         if (!preg_match("/^[a-zA-Z-0-9]*$/", $_POST["pass"])) {
69 |             $passErr = "Only letters and numbers allowed";
70 |         }
71 |     }
72 | }
73 |
74 | if (empty($_POST["fname"])) {
75 |     $fnameErr = "Full Name is required";
76 | } else {
77 |     $fname = test_input($_POST["fname"]);
78 |     if (!preg_match("/^[a-zA-Z- ]*$/", $_POST["fname"])) {
79 |         $fnameErr = "Only letters and whitespaces allowed";
80 |     }
81 | }
82 |
83 | if (empty($_POST["email"])) {
84 |     $emailErr = "Email is required";
85 | } else {
86 |     $email = test_input($_POST["email"]);
87 |     if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
88 |         $emailErr = "Invalid Email Format";
89 |     }
90 | }
91 |
92 | function test_input($data) {
93 |     $data = trim($data);
94 |     $data = stripslashes($data);
95 |     $data = htmlspecialchars($data);
96 |     return $data;
97 | }
98 |
99 |

```

HTML forms allowing users to update the different elements of their profile:

```

100 | </div>
101 | <div id='edit'>
102 | <h2> Edit your Profile Data </h2>
103 | <p> Fill out the respective forms below with the data you wish to update</p>
104 |
105 | <h3> To Edit Your User Name: </h3>
106 | <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
107 |
108 | User Name: <input type="text" name="name">
109 | <span class="error"><?php echo $nameErr; ?></span>
110 | <br><br>
111 | <input type="submit" name="submit1">
112 | </form>
113 |
114 | <h3> To Edit Your Password: </h3>
115 | <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
116 | Password: <input type="text" name="pass">
117 | <span class="error"><?php echo $passErr; ?></span>
118 | <br><br>
119 | <input type="submit" name="submit2">
120 | </form>
121 |
122 | <h3> To Edit Your Full Name on File: </h3>
123 | <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
124 | Full Name: <input type="text" name="fname">
125 | <span class="error"><?php echo $fnameErr; ?></span>
126 | <br><br>
127 | <input type="submit" name="submit3">
128 | </form>
129 |
130 | <h3> To Edit Your Email on File: </h3>
131 | <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
132 | E-Mail: <input type="text" name="email">
133 | <span class="error"><?php echo $emailErr; ?></span>
134 | <br><br>
135 | <input type="submit" name="submit4">
136 | </form>
137 | </div>

```

Program to update the username in the database, if the user fills out the first form:

```

139 <?php
140 if (isset($_POST["submit1"])) {
141     $newName = $_POST["name"];
142     $servername = "localhost:3311";
143     $username = "lorenas";
144     $password = "lorenas";
145     $dbname = 'social';
146     $login = 0;
147     $oldEmail = $_COOKIE["email"];
148
149     try {
150         $conn = new PDO("mysql:host=$servername; dbname=$dbname", $username, $password);
151         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
152         //echo "Made the connection";
153
154         //hashing the PW
155         $hashedpassword = hash("sha256", $_POST["pass"]);
156         $sql = "UPDATE user SET UserName = '$newName' WHERE Email='$oldEmail'";
157
158         $conn->exec($sql);
159
160         echo "ch3>Updated Successfully.</h3>";
161         echo "ch3>New User Name: $newName</h3>";
162
163         $_SESSION["username"] = $newName;
164     } catch (PDOException $e) {
165         echo "Failed: " . $e->getMessage();
166     }
167     $conn = null;
168 }
169
170

```

Program to update the password in the database, if the user fills out the second form:

```

171
172 if (isset($_POST["submit2"])) {
173     $newPass = $_POST["pass"];
174     $servername = "localhost:3311";
175     $username = "lorenas";
176     $password = "lorenas";
177     $dbname = 'social';
178     $login = 0;
179     $oldEmail = $_COOKIE["email"];
180
181     try {
182         $conn = new PDO("mysql:host=$servername; dbname=$dbname", $username, $password);
183         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
184         //echo "Made the connection";
185
186         //hashing the PW
187         $hashedNewPassword = hash("sha256", $_POST["pass"]);
188         $sql = "UPDATE user SET Password = '$hashedNewPassword' WHERE Email='$oldEmail'";
189
190         $conn->exec($sql);
191
192         echo "ch3>Updated Successfully.</h3>";
193         echo "ch3>New Password: $newPass</h3>";
194
195         $_SESSION["password"] = $newPass;
196     } catch (PDOException $e) {
197         echo "Failed: " . $e->getMessage();
198     }
199     $conn = null;
200 }
201
202

```

Program to update the name in the database, if the user fills out the third form:

```

203
204 if (isset($_POST["submit3"])) {
205     $newName = $_POST["name"];
206
207     $servername = "localhost:3311";
208     $username = "lorenas";
209     $password = "lorenas";
210     $dbname = 'social';
211     $login = 0;
212     $oldEmail = $_COOKIE["email"];
213
214     try {
215         $conn = new PDO("mysql:host=$servername; dbname=$dbname", $username, $password);
216         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
217
218         $sql = "UPDATE user SET Name = '$newName' WHERE Email='$oldEmail'";
219
220         $conn->exec($sql);
221
222         echo "ch3>Updated Successfully.</h3>";
223         echo "ch3>New Name: $newName</h3>";
224
225         $cookie_name = "fullname";
226         $cookie_value = $newName;
227         setcookie($cookie_name, $cookie_value, time()+446400, "/");
228     } catch (PDOException $e) {
229         echo "Failed: " . $e->getMessage();
230     }
231     $conn = null;
232 }
233
234

```

Program to update the email in the database, if the user fills out the fourth form:


```

237 if (isset($_POST["submit4"])) {
238     $newEmail = $_POST["email4"];
239     $servername = "localhost:3333";
240     $username = "username";
241     $password = "password";
242     $dbname = "localhost";
243     $login = 0;
244     $oldName = $_SESSION["username"];
245
246     try {
247         $conn = new PDO("mysql:host=$servername; dbname=$dbname", $username, $password);
248         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
249         $sql = "UPDATE user SET Email = '$newEmail' WHERE Email='$oldName'";
250         $conn->exec($sql);
251         echo "Updated Successfully</div>";
252         echo "Click New Email: $newEmail</div>";
253         $cookie_name = "email";
254         $cookie_value = "newEmail";
255         setcookie($cookie_name, $cookie_value, time()+60000, "/");
256     } catch (PDOException $e) {
257         echo "Failed: " . $e->getMessage();
258     }
259     $conn = null;
260 }

```

Output of the form and how the user can fill it out to update the data:

(I chose to make these separate forms so that the user can choose to update just one element (ie. The password) or all the elements but submitting all the forms)

Edit your Profile Data

Fill out the respective forms below with the data you wish to update

To Edit Your User Name:

User Name:

To Edit Your Password:

Password:

To Edit Your Full Name on File:

Full Name:

To Edit Your Email on File:

E-Mail:

Updating every element for this profile and submitting them all:

Edit your Profile Data

Fill out the respective forms below with the data you wish to update

To Edit Your User Name:

User Name:

To Edit Your Password:

Password:

To Edit Your Full Name on File:

Full Name:

To Edit Your Email on File:

E-Mail:

Success message at the bottom of the form:

Updated Successfully.

Now the edits are reflected in the displayed information at the top of the page, as well as in the database.

What's Up SM?

Your Current Profile Info

Profile Info:	Your Profile Data:
User Name:	Lorena02
Password:	lorena
Full Name:	Lorena B Spallino
Email:	lorena1@gmail.com

Assignment Tasks 03 (List all the users):

Program of the 'friend.php' page. This programming allows the user to see all other users if they wish:

```

1 <html>
2 <head>
3 <meta charset="UTF-8">
4 <title></title>
5 <link rel="stylesheet" href="home.css"/>
6 </head>
7 <body>
8 <header>
9 <div class="navbars">
10 <a style="text" href="home.php">Home</a><br><br><br>
11 <a href="friend.php">Friends</a><br><br><br>
12 <a href="cart.htm">Images</a><br><br><br>
13 <a href="profile.php">Profile</a><br><br><br>
14 </div>
15 </header>
16 <h2 id="h2">What's Up SM?</h2>
17 <body>
18 <h3>Here is where you can see a list of all users, and friends</h3>
19 <form method="post" action="friend.php" <?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>
20 <h3>Click the button below to show a list of all users</h3>
21 <input type="submit" name="users" value="Users" id="btn">
22 </form>
23 <div id="div3">
24 <?php
25
26 if($_SERVER["REQUEST_METHOD"] == "POST"){
27
28     echo "ch3 id='blue'>List of all users</h3>";
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59

```

```

29
30 $servername = "localhost:3311";
31 $username = "LorenaS";
32 $password = "LorenaS";
33 $dbname = "social";
34
35
36 try {
37     $conn = new PDO("mysql:host=$servername; dbname=$dbname", $username, $password);
38     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
39
40     $sql2 = "SELECT * FROM user";
41     $q = $conn->query($sql2);
42     $q->setFetchMode(PDO::FETCH_ASSOC);
43
44     // displaying the data
45     while($row = $q->fetch()){
46         echo "ch6 id='blue'> User Name: " . $row["UserName"] . "<br> </h3>";
47     }
48
49     catch(PDOException $e) {
50         echo "Failed: " . $e->getMessage();
51     }
52     $conn = null;
53 }
54
55 <?php
56 </div>
57 </body>
58 </html>
59

```

The output of the above program, before user choses to see the list of all users:

[Home](#)
[Friends](#)
[Images](#)
[Profile](#)

What's Up SM?

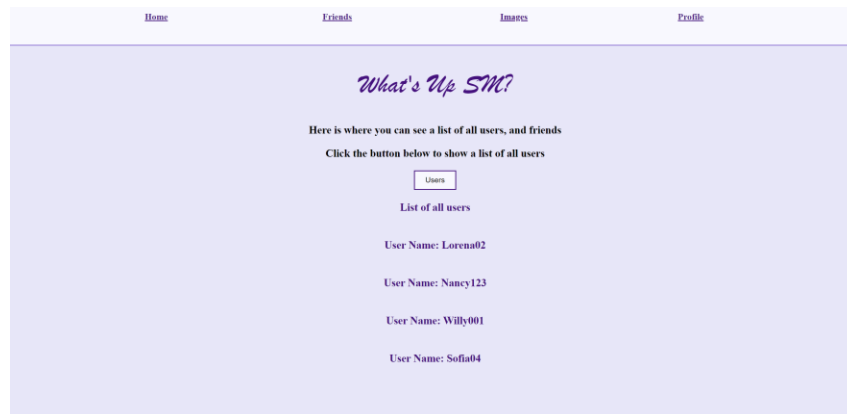
Here is where you can see a list of all users, and friends

Click the button below to show a list of all users

Users

Output (after user clicks the 'user; button): - Only showing the username to protect user's sensitive data (like full name and email)

Displaying a list of all users:



Assignment Tasks 04 (Send and accept Friend request):

Below shows the friend Table in database with it's appropriate columns.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 ID	int(3)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 UserName1	varchar(60)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3 UserName2	varchar(60)	utf8mb4_general_ci		No	None			Change Drop More

Below shows the php program in the friend.php page to send a friend request to a specific user. Using database, sessions and cookies.

```
31 <?php
32 if(isset($_POST["users"])){
33     $SESSION["friend1"] = $_SESSION["username"];
34     echo "<h3 id='blue'>List of all users</h3>";
35
36     $servername = "localhost:3311";
37     $username = "LorenaS";
38     $password = "LorenaS";
39     $dbname = 'social';
40
41     try {
42         $conn = new PDO("mysql:host=$servername; dbname=$dbname", $username, $password);
43         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
44
45         $sql2 = "SELECT * FROM users";
46         $q = $conn->query($sql2);
47
48         $users = $q->fetchAll(PDO::FETCH_ASSOC);
49         static $a=0;
50         foreach($users as $row){
51             $a++;
52             echo "<h6 id='blue'><br> User Name: " . $row["UserName"] . " <form method='POST' name='fr'><input type='submit' name='$a' value='Send Frie
53
54
55         }
56     } catch(PDOException $e) {
57         echo "Failed: " . $e->getMessage();
58     }
59 }
60 $conn = null;
```

```

62         static $x = 0;
63         while($x<100){
64             $x++;
65             if(isset($_POST[strval($x)])){
66                 $fid = strval($x);
67                 echo "<h3 class='error'>Friend Request Sent! </h3><br>";
68                 $_SESSION["friend2"] = $fid;
69             }
70         }
71     }
72
73     $cookieName = "friend";
74     $cookieValue = $_SESSION["friend1"];
75     setcookie($cookieName, $cookieValue, time()+846000, "/");
76
77

```

Program for accepting or denying a friend request:

```

80     <div id="$i" style="float:right;">
81     <?php
82
83     if(isset($_POST["friends"])){
84
85         if ($_SESSION["friend2"] == $_SESSION["userid"]) {
86
87         }
88         if ($_SESSION["friend2"] == $_SESSION["userid"]) {
89             echo "<h1 class='error'> ". $_COOKIE['friend'] . " Has sent you a friend request! You can accept or deny their request </h1>";
90
91         }
92     }
93     <form method='post' id='form1' style='margin-left:40%;'>
94         <input type='submit' name='accept' Value='Accept!'>
95         <input type='submit' name='deny' Value='Deny!'>
96     </form> <?php
97
98
99     $u1 = $_COOKIE['friend'];
100     $u2 = $_SESSION["username"];
101
102     if(isset($_POST["accept"])){
103
104         $u1 = $_COOKIE['friend'];
105         $u2 = $_SESSION["username"];
106
107         $servername = "localhost:3311";
108         $username = "LorenaS";
109         $password = "LorenaS";
110         $dbname = 'social';
111
112         try {
113             $conn = new PDO("mysql:host=$servername; dbname=$dbname", $username, $password);
114
115             $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
116             // echo "made the connection";
117
118             $sql = "INSERT INTO `friend` (`ID`, `UserName1`, `UserName2`) *
119                 . "VALUES (NULL, '$u1', '$u2')";
120
121             $conn->exec($sql);
122
123             echo "<h3>You have accepted $u1 friend request. You are now friends!</h3>";
124
125         } catch(PDOException $e) {
126             echo "Failed: " . $e->getMessage();
127         }
128     }
129     $conn = null;
130
131     if(isset($_POST["deny"])){
132
133         $u1 = $_COOKIE['friend'];
134         $u2 = $_SESSION["username"];
135         echo "<h3 class='error' style='margin-right:5%;margin-left:-25%;>You have denied $u1's friend request. You are not friends.</h3>";
136     }
137
138     ?>
139     <?php
140     if(isset($_POST["friends"])){
141         echo "<form method='post' id='fform' style='margin-top:20px;'>
142         <input type='submit' name='flist' Value='View Friend List' style='margin-left:155px;'>
143         </form>";

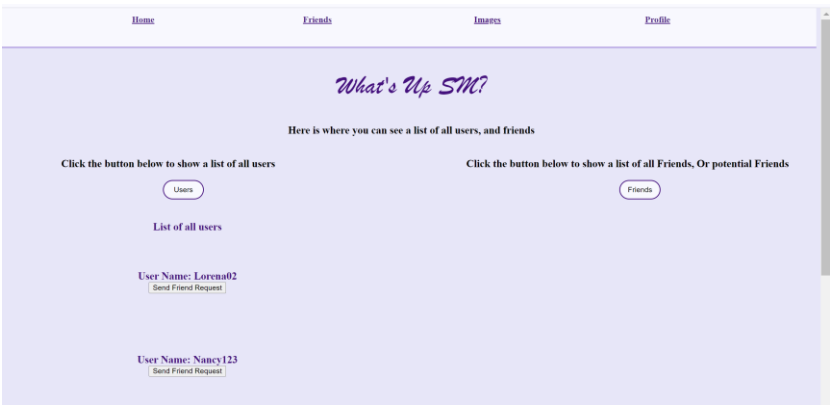
```

```

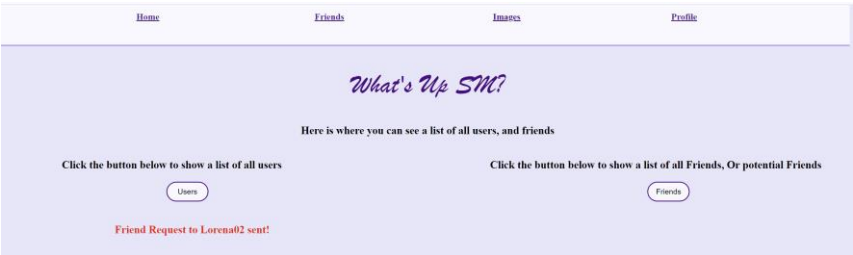
144     }
145     if(isset($_POST["flist"])){
146         header("Location: /PhTestpProject1/friend_r.php");
147         exit();
148     }
149
150     ?>
151 </div>
152 </div>
153 </body>

```

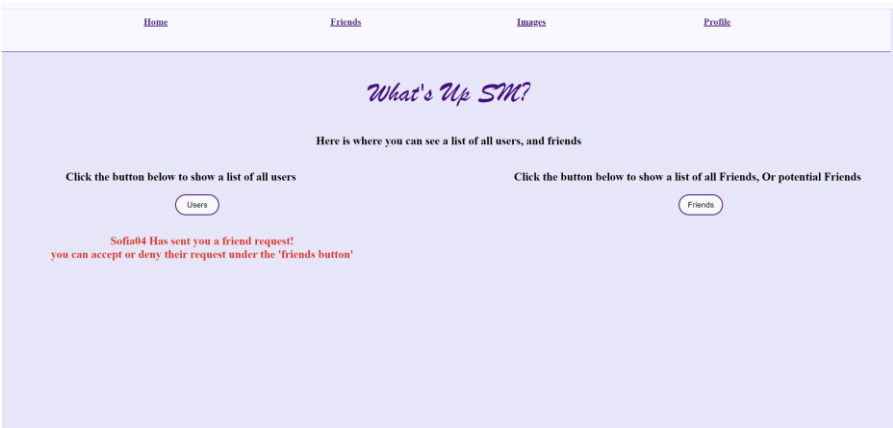
Output of requesting a friend:
(Currently logged in under username Sofia04):
Viewing the users:



Clicking on send friend request button for user Lorena02:



Now logged in as Lorena02:



Accepting the friend request:

[Home](#)
[Friends](#)
[Images](#)
[Profile](#)

What's Up SM?

Here is where you can see a list of all users, and friends

Click the button below to show a list of all users

Users

Click the button below to show a list of all Friends, Or potential Friends

Friends

Sofia04 Has sent you a friend request!
you can accept or deny their request under the 'Friends button'

Sofia04 Has sent you a friend request! You can accept or deny their request

Accept | Deny

You have accepted Sofia04 friend request. You are now friends!

Checking the Database:

Extra Options			
	ID	UserName1	UserName2
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Sofia04	Lorena02

OR - Denying the friend request:

You have denied Sofia04's friend request. You are not friends.

Assignment Tasks 05 (Display list of friends):

On the friend.php page, user can choose to see a list of all friends, if so, their friends will be displayed, from the friends database this shows all their current friends.

Program:

```

168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
<?php
if(isset($_POST["friends"])){
    echo "<form method='post' id='fform' style='margin-top:20px;'>
    <input type='submit' name='flist' Value='View Friend List' style='margin-left:255px;'>
    </form>";
}
if(isset($_POST["flist"])){
    $servername = "localhost:3311";
    $username = "Lorena02";
    $password = "Lorena02";
    $dbname = 'social';

    try {
        $conn = new PDO("mysql:host=$servername; dbname=$dbname", $username, $password);
        //echo "Made connection";
        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        $usernamef = $_SESSION["username"];
        $sql = "SELECT * FROM friend WHERE UserName2 = '$usernamef'";
        $q = $conn->query($sql);
        $q->setFetchMode(PDO::FETCH_ASSOC);

        // displaying the data

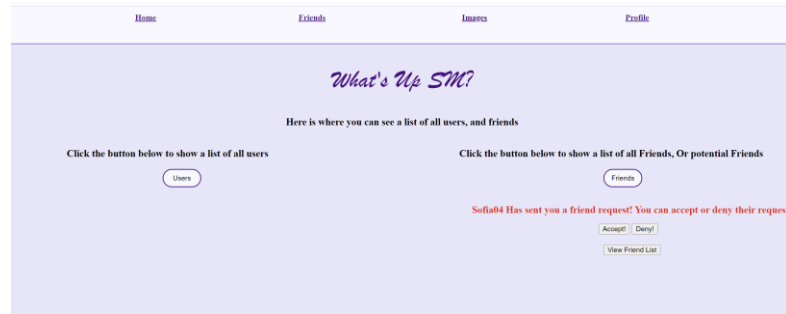
```

```

192
193
194 while($row = $q->fetch()){
195
196     echo "<div id='blue' style='text-align:right; margin-left:250px;'><div> Friend: " . $row["UserName"] . " " </div>";
197
198 }
199 catch(PDOException $e) {
200     echo "Failed: " . $e->getMessage();
201 }
202 }
203 $conn = null;
204
205 </div>
206 </div>
207 </body>
208 </html>

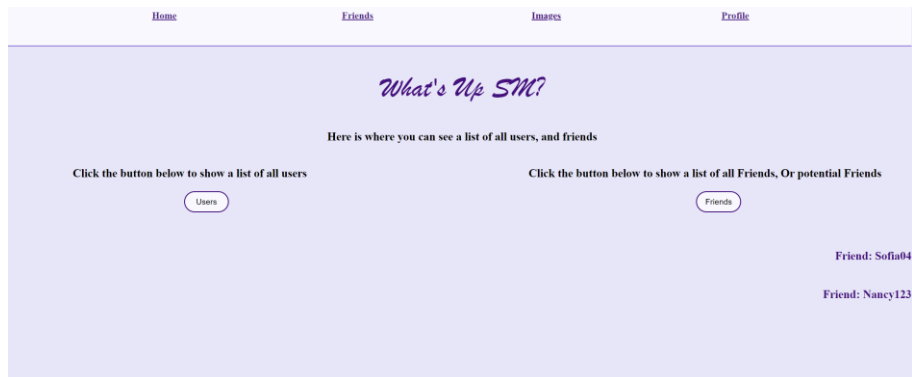
```

Output:



Clicking on View friend list button:

All of my friends are displayed on the right side of the screen, only their usernames to protect their sensitive data.



Assignment Tasks 06 (Share text):

Making a new table for messages in this database, called chat:

Server: 127.0.0.1:3311 > Database: social > Table: chat										
Browse Structure SQL Search Insert Export Import Privileges Operations										
Table structure Relation view										
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	
<input type="checkbox"/>	1 Friend1	varchar(60)	utf8mb4_general_ci		No	None			Change	Drop More
<input type="checkbox"/>	2 Friend2	varchar(60)	utf8mb4_general_ci		No	None			Change	Drop More
<input type="checkbox"/>	3 Message	varchar(200)	utf8mb4_general_ci		No	None			Change	Drop More

Program:

After clicking on view friend list button, will show the friends with a "send message" button next to it, to start the chat this button redirects to friend_r.php page, where users can view all their friends and optionally message each one individually.

```

145 <?php
146 if(isset($_POST["friends"])){
147     echo "<form method='post' id='fform' style='margin-top:20px;'>
148     <input type='submit' name='flist' value='View Friend List' style='margin-left:155px;'>
149     </form>";
150 }
151 if(isset($_POST["flist"])){
152     header("Location: /PhTestpProject1/friend_r.php");
153     exit();
154 }
155 }
156

```

Friend_r.php Program, lists all friends and buttons to message each one, uses sessions and cookies to identify friends, and pulls info from the respective database – if button is clicked, redirects to the chat page for that friend.

```

20 <?php
21 session_start();
22 $servername = "localhost:3311";
23 $username = "Lorenad";
24 $password = "Lorenad";
25 $dbname = 'social';
26
27
28 try {
29     $conn = new PDO("mysql:host=$servername; dbname=$dbname", $username, $password);
30     // echo "Made connection";
31     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
32
33     $usernamef = $_SESSION["username"];
34     $sql = "SELECT * FROM friend WHERE UserNme2 = '$usernamef' OR UserNme1 = '$usernamef'";
35     $q = $conn->query($sql);
36     // $q->setFetchMode(PDO::FETCH_ASSOC);
37
38     // displaying the data
39     $users = $q->fetchAll(PDO::FETCH_ASSOC);
40     static $b=0;
41     foreach($users as $row){
42         $b++;
43         echo "<div id='fb' style='text-align:center; margin-left:50px;'><div> Friends " . $row["UserName1"] . " " .
44         . "<form method='POST'><input type='submit' name='fb' value='Message Friend' id='btn1' style='margin-left:10px;'></form></div></div>";
45
46         $me = $row["UserName2"];
47         $you = $row["UserName1"];
48         $id = $row["ID"];
49
50
51         echo $me;
52         echo $you;
53         $_SESSION["you"] = $you;
54         $_SESSION["chatID"] = $id;
55         echo $_SESSION["chatID"];
56     }
57 } catch(PDOException $e) {
58     echo "Failed: " . $e->getMessage();
59 }
60
61 $conn = null;
62 static $t = 0;
63 while($t < 100){
64     $t++;
65     if(isset($_POST["strval($t)"])){
66         $servername = "localhost:3311";
67         $username = "Lorenad";
68         $password = "Lorenad";
69         $dbname = 'social';
70
71         try {
72             $conn = new PDO("mysql:host=$servername; dbname=$dbname", $username, $password);
73             // echo "Made connection";
74             $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
75
76             $cid1 = $_SESSION["chatID"];
77             $sql = "SELECT * FROM friend WHERE ID=$cid1";
78             $q = $conn->query($sql);
79             // $q->setFetchMode(PDO::FETCH_ASSOC);
80
81             // displaying the data
82             $users = $q->fetchAll(PDO::FETCH_ASSOC);
83             static $b=0;
84             foreach($users as $row){
85                 $b++;
86                 $me = $row["UserName2"];
87                 $you = $row["UserName1"];
88
89                 $_SESSION["you"] = $you;
90                 $_SESSION["me"] = $me;
91
92             }
93         } catch(PDOException $e) {
94             echo "Failed: " . $e->getMessage();
95         }
96
97         $conn = null;
98         header("Location: /PhTestpProject1/chat.php");
99         exit();
100     }
101 }
102
103 ?>
104 </body>
105 </html>
106

```


As already mentioned, user can choose the friend they would like to chat with and redirects to the chat page (Redirect to chat.php).

Program for chat.php – This is where users can text each other:

```
21 <?php
22 session_start();
23 $servername = "localhost:3311";
24 $username = "LorenaS";
25 $password = "LorenaS";
26 $dbname = 'social';
27
28 try {
29     $conn = new PDO("mysql:host=$servername; dbname=$dbname", $username, $password);
30     // echo "Made connection";
31     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
32
33     $username = $_SESSION["username"];
34     $sql = "SELECT * FROM friend WHERE UserName2 = '$username' OR UserName1 = '$username'";
35     $q = $conn->query($sql);
36
37     $users = $q->fetchAll(PDO::FETCH_ASSOC);
38     static $b=0;
39     foreach($users as $row) {
40
41         $me = $row["UserName2"];
42         $you = $row["UserName1"];
43         $cid = $row["ID"];
44         //echo $me;
45         //echo $you;
46         $_SESSION["me"] = $me;
47         $_SESSION["you"] = $you;
48         $_SESSION["chatID"] = $cid;
49         //echo $_SESSION["chatID"];
50         //echo $_SESSION["you"];
51     }
52 }
```

```
149 if(isset($_POST["send"])){
150     $servername = "localhost:3311";
151     $username = "LorenaS";
152     $password = "LorenaS";
153     $dbname = 'social';
154
155     try {
156         $conn = new PDO("mysql:host=$servername; dbname=$dbname", $username, $password);
157
158         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
159         // echo "made the connection";
160         $f1 = $_SESSION["you"];
161         $f2 = $_SESSION["me"];
162         $txt = $_POST["box"];
163
164         $sql = "INSERT INTO `chat` (`Friend1`, `Friend2`, `Message`) "
165             . "VALUES ('$f1', '$f2', '$txt')";
166
167         $conn->exec($sql);
168
169     } catch(PDOException $e) {
170         echo "Failed: " . $e->getMessage();
171     }
172 }
173
174 $conn = null;
175
176 $servername = "localhost:3311";
177 $username = "LorenaS";
178 $password = "LorenaS";
```

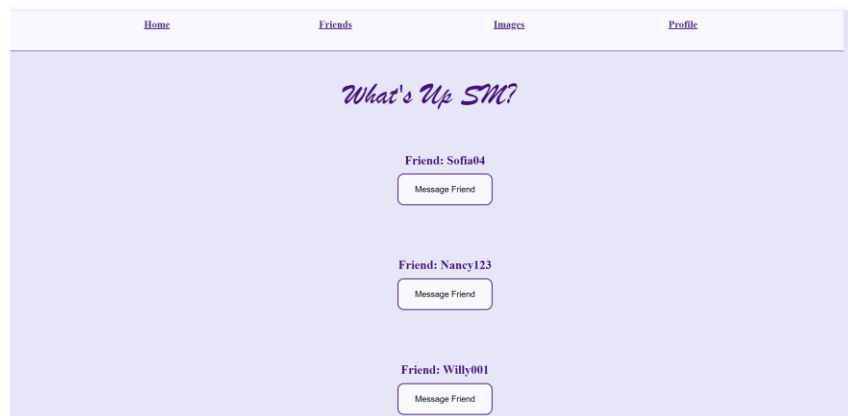
```
180 $dbname = 'social';
181
182 try {
183     $conn = new PDO("mysql:host=$servername; dbname=$dbname", $username, $password);
184     //echo "Made connection";
185     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
186     $f1 = $_SESSION["you"];
187     $f2 = $_SESSION["me"];
188
189     $sql = "SELECT * FROM chat WHERE Friend1 = '$f1' AND Friend2 = '$f2' OR Friend1 = '$f2' AND Friend2 = '$f1'";
190     $q = $conn->query($sql);
191     $q->setFetchMode(PDO::FETCH_ASSOC);
192
193     while($row = $q->fetch()) {
194
195         echo "<h3 id='blue'> <br> " . $row["Message"] . " "
196             . " <br> </h3>";
197     }
198 } catch(PDOException $e) {
199     echo "Failed: " . $e->getMessage();
200 }
201
202 $conn = null;
203
204
205 ?>
206 </div>
207
```

Output:

Navigate to the Messages page: (Logged in as Lorena02 – view friends button)

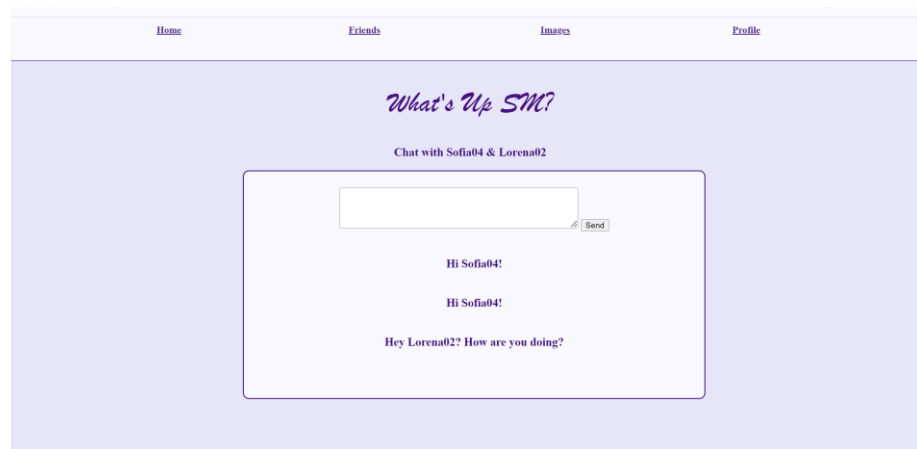


View friend list – redirect to friend_r.php:



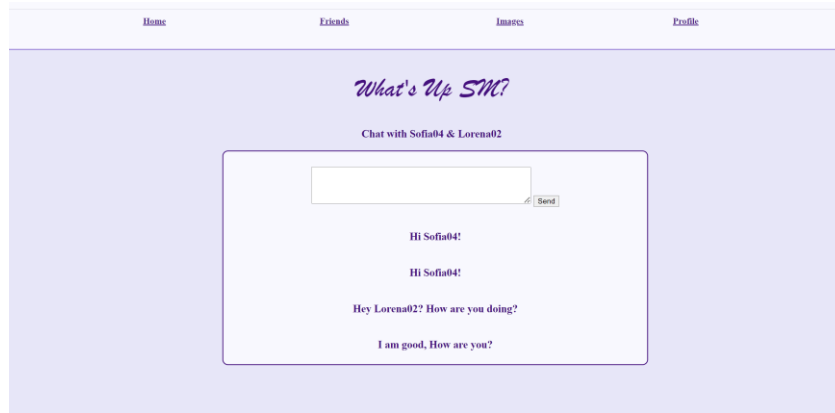
User clicks on Message friend and it takes them to that friends message platform:

Redirect to chat.php, here users can send messages, database is populated, and messages are displayed within the chat:

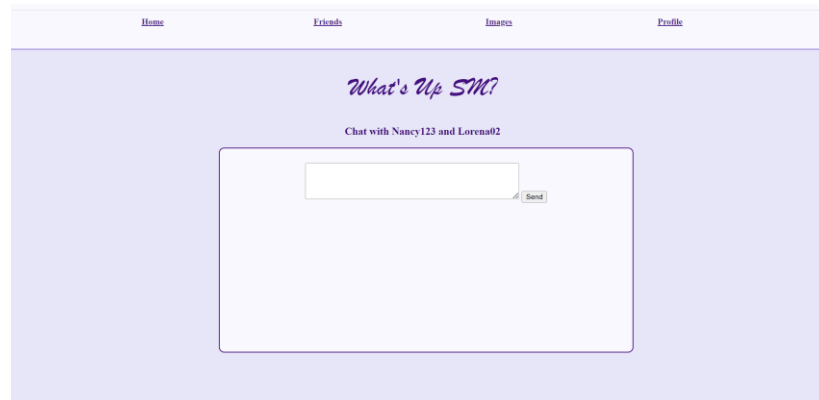


Both users can see previous messages sent by either user, and send new ones which appear at the bottom:

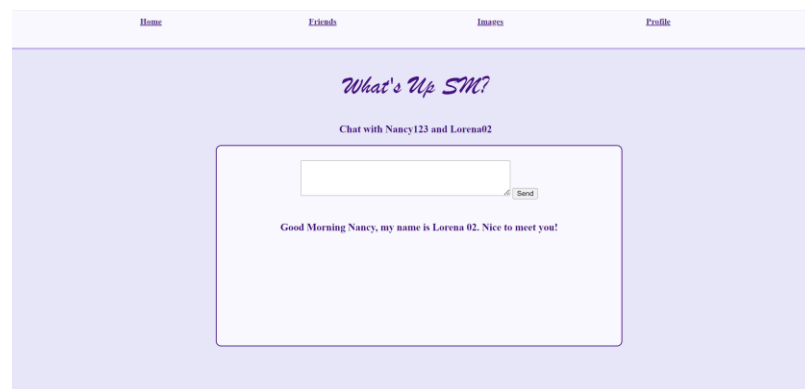
The program adds the text when the user clicks send, and adds the text to the database:



Opening a different chat – with a different friend:



Sending some text:



Assignment Tasks 07 (Create Photo Gallery):

Programming to display all the images in the gallery – all images are pulled from the uploads folder, and displayed in a visually appealing way for the user:

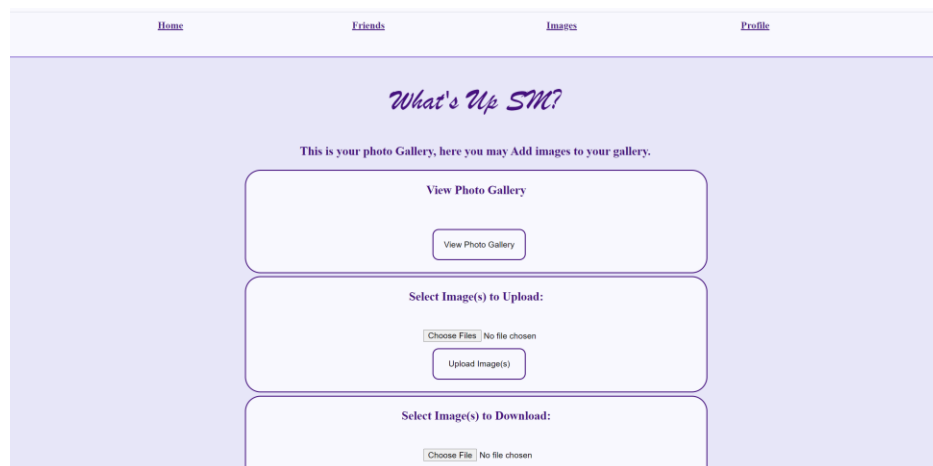
```

4      <link rel='stylesheet' href='home.css' />
5      <title></title>
6      </head>
7      <body>
8          <header>
9
10         <div class="navbar">
11             <a style='text' href="home.php">Home</a><br><br><br><br>
12             <a href="friend.php">Friends</a><br><br><br><br>
13             <a href="image.php">Images</a><br><br><br><br>
14             <a href="profile.php">Profile</a><br><br><br><br>
15
16         </div>
17     </header>
18     <h2 id='nav'>What's Up SM?</h2>
19     <h3 id="blue">View Your Image Gallery Below </h3>
20
21     <?php
22     function displayImg() {
23         $dirname = "uploads/";
24         $images = scandir($dirname);
25         $ignore = Array(".", "..");
26         foreach($images as $curimg) {
27             if (in_array($curimg, $ignore)) {
28                 echo "<img src='uploads/$curimg' /height='200' width='200' id='imgs'>";
29             }
30         }
31     }
32     displayImg();
33
34     ?>
35 </body>
36 </html>

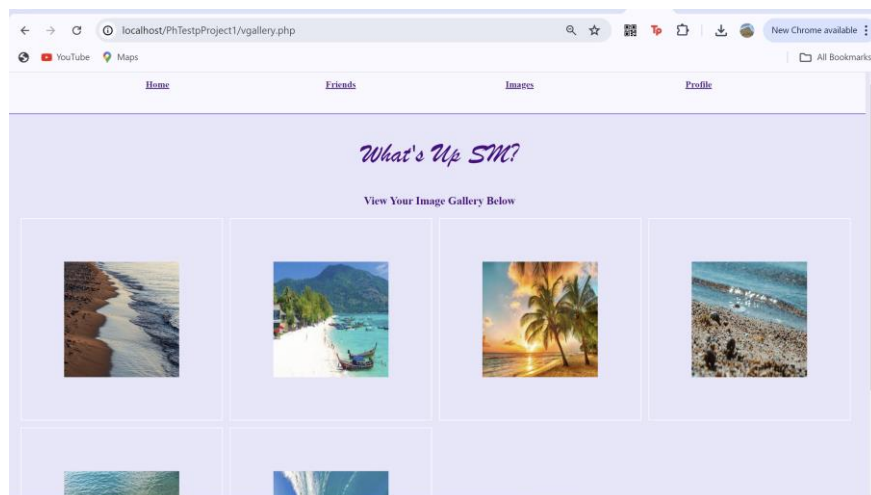
```

Output:

In the image.php file, user can click on view gallery, to see their photo gallery



Viewing the photo gallery



Assignment Tasks 08 (Upload/Download to Photo Gallery):

Image.php file includes the forms for uploading images and downloading images from gallery:

Program for image.php HTML forms:

```
7 <body>
8 <header>
9 <div class="navbar">
10 <a style="text" href="home.php">Home</a><br><br><br>
11 <a href="friend.php">Friends</a><br><br><br>
12 <a href="image.php">Images</a><br><br><br>
13 <a href="profile.php">Profile</a><br><br><br>
14 </div>
15 </header>
16 <h2 id="nav">What's Up SM?</h2>
17 <h3 id="blue">This is your photo Gallery, here you may Add images to your gallery.</h3>
18 <div class="imgdiv">
19 <form id="imgform" action="gallery.php" method="post">
20 <h3 id="blue">View Photo Gallery </h3><br>
21 <input type="submit" value="View Photo Gallery" name="gal" id="btn1">
22 </form>
23 </div>
24 <div class="imgdiv">
25 <form id="imgform" action="gallery.php" method="post" enctype="multipart/form-data">
26 <h3 id="blue">Select Image(s) to Upload: </h3><br>
27 <input type="file" name="fileToUpload" id="fileToUpload" accept="image/*" multiple>
28 <br>
29 <input type="submit" value="Upload Image(s)" name="submit" id="btn1">
30 </form>
31 </div>
32 <div class="imgdiv">
33 <form action="gallery.php" method="post" enctype="multipart/form-data">
34 <h3 id="blue">Select Image(s) to Download: </h3><br>
35 <input type="file" name="fileToDownload" id="fileToDownload">
36 <br><!-- comment -->
37 <input type="submit" value="Download File" name="fileToDownload" id="btn1">
38 </form>
39 </div>
```

Output:

The screenshot displays the web application's output. At the top, there is a navigation bar with links for Home, Friends, Images, and Profile. Below this, the main content area has a heading "What's Up SM?" and a subheading "This is your photo Gallery, here you may Add images to your gallery." The interface contains three main sections: "View Photo Gallery" with a "View Photo Gallery" button, "Select Image(s) to Upload:" with a "Choose Files" button (showing "No file chosen") and an "Upload Image(s)" button, and "Select Image(s) to Download:" with a "Choose File" button (showing "No file chosen").

Gallery.php handles the uploading of the images, and displays them upon successful upload.

Program for gallery.php:

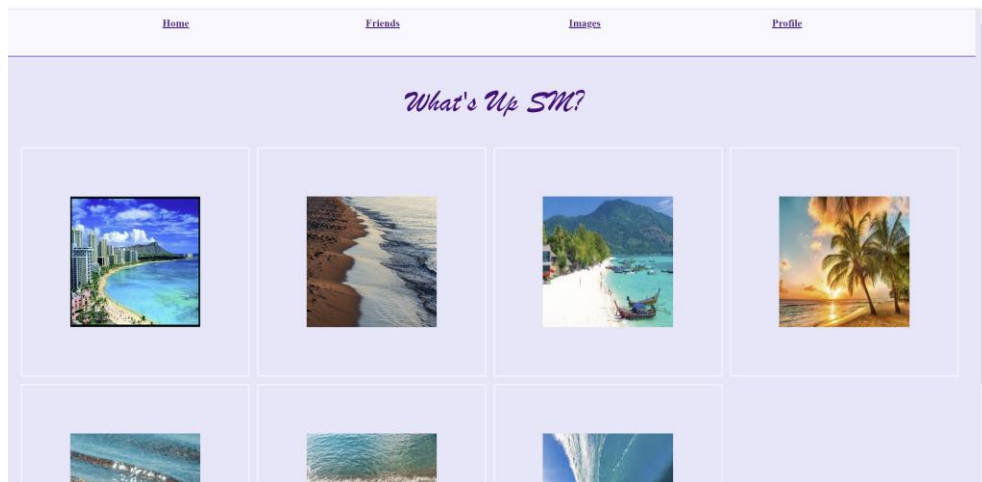
```
1 <html>
2 <head>
3 <meta charset="UTF-8">
4 <link rel="stylesheet" href="home.css"/>
5 <title></title>
6 </head>
7 <body>
8 <header>
9 <div class="navbar">
10 <a style="text" href="home.php">Home</a><br><br><br>
11 <a href="friend.php">Friends</a><br><br><br>
12 <a href="image.php">Images</a><br><br><br>
13 <a href="profile.php">Profile</a><br><br><br>
14 </div>
15 </header>
16 <h2 id="nav">What's Up SM?</h2>
17
18 <?php
19 $target_dir = "uploads/";
20 $target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
21 $uploadOk = 1;
22 $imageFileType = strtolower(pathinfo($target_file, PATHINFO_EXTENSION));
23
24 //check if image is real or fake
25 if (isset($_POST["submit"])) {
26     $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
27     if ($check !== false) {
28         //echo "File is an image " . $check["mime"] . ". ";
29         $uploadOk = 1;
30     }
31 }
```

```

33     } else {
34         echo "file is not an image";
35         $uploadOk=0;
36     }
37 }
38
39 //Check if file already exists
40 if (file_exists($target_file)){
41     echo "Sorry, file already exists.";
42     $uploadOk=0;
43 }
44
45 //check file size
46 if ($FILES["fileToUpload"]["size"] > 500000){
47     echo "File too big";
48     $uploadOk=0;
49 }
50
51 //Allow certain file formats
52 if ($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg" && $imageFileType != "gif") {
53     echo "File type not allowed";
54     $uploadOk=0;
55 }
56
57 //Check is uploadOk is set to 0 by error
58 if ($uploadOk == 0){
59     echo "File was not uploaded";
60     //If everything else is ok, try to upload the file
61 } else {
62     if (move_uploaded_file($FILES["fileToUpload"]["tmp_name"], $target_file)){
63
64         if (move_uploaded_file($FILES["fileToUpload"]["tmp_name"], $target_file)){
65             //echo "The file " . htmlspecialchars(basename($FILES["fileToUpload"]["name"])) . " has been uploaded";
66         } else{
67             echo "could not upload";
68         }
69
70         $image = imagecreatefromjpeg($target_file);
71
72         $rotate = imagerotate($image, 00, 00);
73
74         imagejpeg($rotate, "rotatedimg.jpg");
75
76         function displayImg($image){
77             $dirname = "uploads/";
78             $images = scandir($dirname);
79             $ignore = Array(".", "..");
80             foreach($images as $curlimg){
81                 if(!in_array($curlimg, $ignore)) {
82                     echo "<img src='uploads/$curlimg' />";
83                 }
84             }
85         }
86
87     }
88
89     displayImg($image);
90     imagedestroy($image);
91     ?>
92 }
93 </body>

```

After choosing files and uploading them the output looks as follows:



Downloading Images

Html form to select images to download:

```

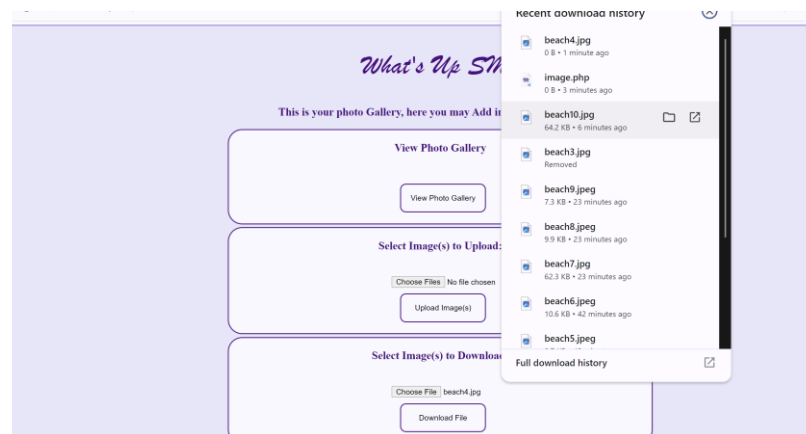
32 </div>
33 <div class="imgdiv">
34     <form action="downloadfile.php" method="post" enctype="multipart/form-data">
35         <h3 id="blue"> Select Image(s) to Download: </h3><br>
36         <input type="file" name="fileToDownload" id="fileToDownload">
37         <br><!-- comment -->
38         <input type="submit" value="Download File" name="fileToDownload" id="btn1">
39     </form>
40 </div>
41
42 </body>
43 </html>

```

Downloadfile.php – which handles the downloading of the files:

```
Source History
1 <?php
2 function downloadFile($filePath){
3     $fileName = basename($filePath);
4     $fileLength = filesize($filePath);
5
6     header("Content-Type: application/octet-stream");
7     header("Content-Disposition: attachment; filename = \"{$fileName}\"");
8     header("Content-Length: $fileLength");
9     header("Content-Description: File Transfer");
10    //header("Expires: 0");
11    //header("Cache-Control: must-revalidate");
12    //header("Pragma: private");
13    echo "inside calling. <br>";
14    ob_clean();
15    flush();
16    readfile($filePath);
17    flush();
18    readfile($fileName);
19 }
20
21 $fileSource = $_FILES["fileToDownload"]["name"];
22 echo $fileSource;
23 downloadFile($fileSource);
24 readfile($fileSource);
25
26 ?>
```

Output:



Assignment Tasks 09 (Share/upload multiple images):

The screenshots below show the program and output that allows the users to send images within the chat feature by uploading them. Users can view these images sent/received and optionally download the image – as seen in the next task.

Program for uploading images – within the chat feature (chat.php)

```
33 <form class="chat" method="post" enctype="multipart/form-data" style="width:50%;margin-left:20%;background:ghostwhite;border:1px solid indigo; border-ra
34 <h3 id="blue"> Select Image(s) to Upload: </h3><br>
35 <input type="file" name="fileToUpload" id="fileToUpload" accept="image/*" multiple>
36 <br>
37 <input type="submit" value="Upload Image(s)" name="submit1" id="btn1" style="margin-left:5%" >
38 </form>
39
40 <?php
41 if(isset($_POST["submit1"])){
42     $target_dir = "uploads/";
43     $target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
44     $uploadOk = 1;
45     $imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
46
47     //check if image is real or fake
48     if (isset($_POST["submit1"])){
49         $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
50         if($check !== false ) {
51             //echo "File is an image " . $check["mime"] . ". ";
52             $uploadOk=1;
53         } else {
54             echo "file is not an image";
55             $uploadOk=0;
56         }
57     }
58
59     //Check if file already exists
60     if(file_exists($target_file)){
61         echo "Sorry, file already exists.";
62         $uploadOk=0;
63     }
64 }
```

```

69
70 //Allow certain file formats
71 if ($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg" && $imageFileType != "gif") {
72     echo "File type not allowed";
73     $uploadOk=0;
74 }
75
76 //Check is uploadOk is set to 0 by error
77 if ($uploadOk == 0){
78     echo "File was not uploaded";
79     //If everything else is ok, try to upload the file
80 } else {
81     if(move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file)){
82         //echo "The file " . htmlspecialchars(basename($_FILES["fileToUpload"]["name"])) . " has been uploaded";
83     } else{
84         echo "could not upload";
85     }
86 }
87
88 $image = imagecreatefromjpeg($target_file);
89
90 $rotate = imagerotate($image, 00, 00);
91
92 imagejpeg($rotate, "rotatedimg.jpg");
93

```

Displaying and downloading the image in the chat:

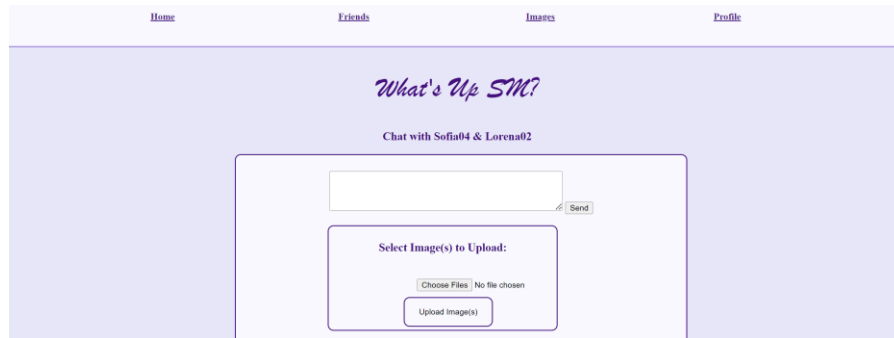
```

94
95 function displayImg($image){
96
97     echo "<h3> Click on image to download <br> </h3>";
98     echo "<a href='rotatedimg.jpg' download='rotatedimg.jpg'><img src='rotatedimg.jpg' alt='img' width='200' height='200'>";
99     </a>";
100
101     $fileToDownload = 'rotatedimg.gif';
102
103     if(isset($_POST["down"])){
104         echo "<a href='download' download='rotatedimg.jpg'>";
105     }
106 }
107
108
109
110 displayImg($image);
111 imagedestroy($image);
112

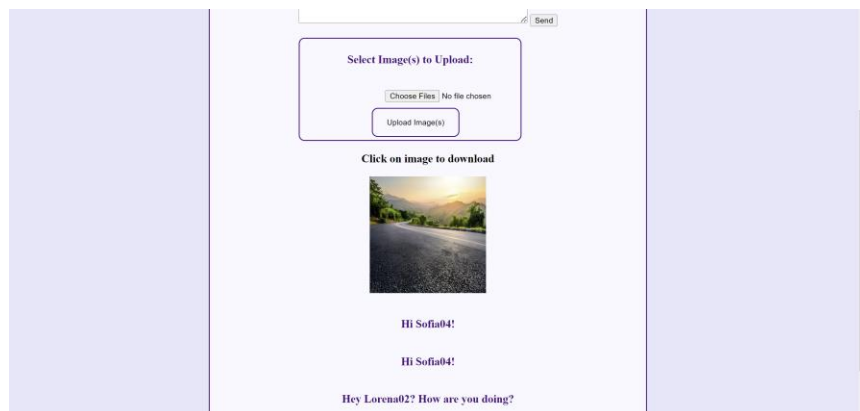
```

Output: Takes place in the chat feature between friends:

Now the chat feature has an additional form; for uploading images to the chat:



After selecting image(s) and uploading them, they are added to the chat, along with the text messages between friends:



Assignment Tasks 10 (Download shared image):

Allowing the user to download chat images. Program in chat.php:

```
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112
```

```
function displayImg($image) {  
  
    echo "<h3> Click on image to download <br> </h3>";  
    echo "<a href='rotatedimg.jpg' download='rotatedimg.jpg'><img src='rotatedimg.jpg' alt='img' width='200' height='200'>  
</a>";  
  
    $fileToDownload = 'rotatedimg.gif';  
  
    if(isset($_POST["down"])) {  
        echo "<a href='download' download='rotatedimg.jpg'>";  
    }  
}  
  
displayImg($image);  
imagedestroy($image);
```

Each time an image is 'sent'/uploaded to the chat, the user can click on the image to download it.

Output of downloading the recently sent image in the chat.

