Week 3 Assignment: Landing, Login, and Enrollment Pages Development

Joshua Hohendorf

The University of Arizona Global Campus

CST499: Capstone for Computer Software Technology

Prof. Joseph Rangitsch

August 12th, 2024

Week 3 Assignment: Landing, Login, and Enrollment Pages Development

Running a PHP file in XAMPP involves a straightforward process that combines setting up your development environment with ensuring your PHP scripts execute correctly. Start by installing XAMPP, a powerful tool that packages Apache, MySQL, PHP, and Perl together, simplifying setting up a local web server. Once installed, launch the XAMPP Control Panel and start the Apache module, which will power the web server needed to run PHP scripts.

Next, navigate to the XAMPP installation directory on your system. Within this directory, locate the htdocs folder. This folder acts as the web server's root directory, where you should place your PHP files. Create a new folder within htdocs to keep your project organized, then copy your PHP file into this newly created directory.

With your PHP file in place, open your preferred web browser. In the address bar, type localhost followed by the name of the folder you created within htdocs and the name of your PHP file. For example, if your file is named test.php and is located in a folder called myproject, you would enter localhost/myproject/test.php into the browser.

Upon pressing Enter, the browser sends a request to the Apache server to process the PHP script. The server executes the PHP code and returns the output, which the browser displays. If there are any errors in your script, they will be shown in the browser, allowing you to debug and refine your code as needed.

This approach ensures you can efficiently test and develop PHP scripts locally, leveraging XAMPP's robust capabilities. These steps will create a seamless workflow for developing and debugging PHP applications, ultimately enhancing your productivity and code quality.

Database Functions

Each database function is designed for a specific purpose. The executeQuery function handles simple database lookups without making any changes, as shown in Figure 1. The executeSelectQuery function retrieves data from the database, returning the requested information, which you can see in Figure 2. The final function checks if a registered user's email already exists in the database; if it does, the system prevents the creation of a duplicate account, as illustrated in Figure 3.

All functions use connection credentials and an SQL command, with the SQL command generated by the code. These credentials are securely stored in a protected folder and accessed via the require_once method, ensuring they remain separate and secure.

Figure 1

The executeQuery database function.

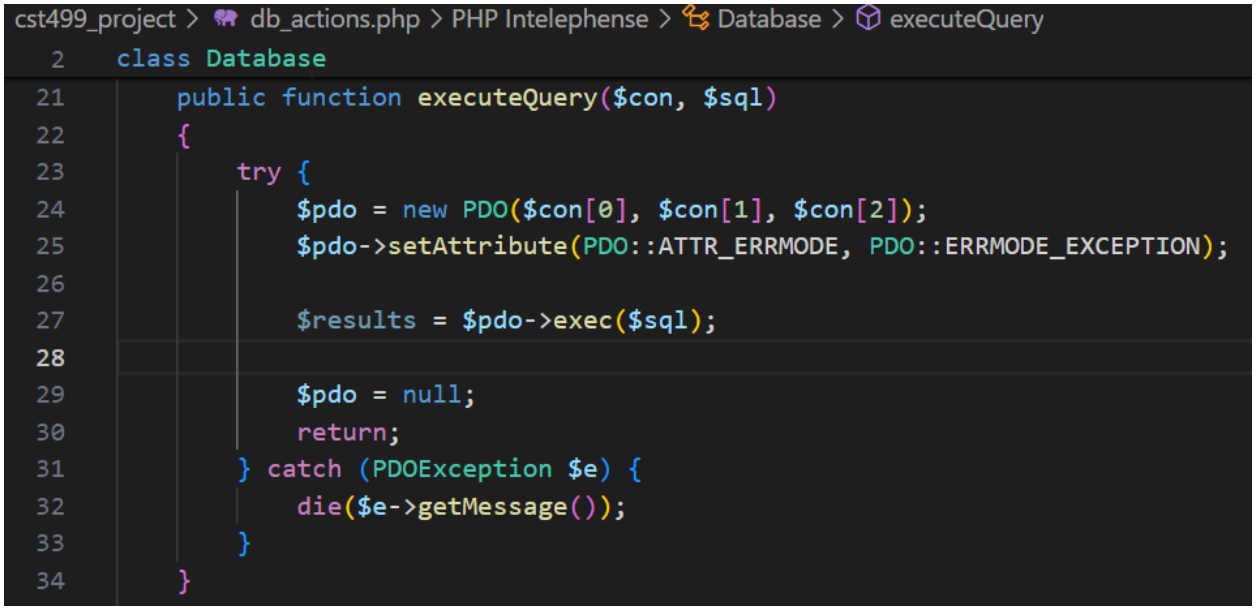


Figure 2

The executeSelectQuery database function.

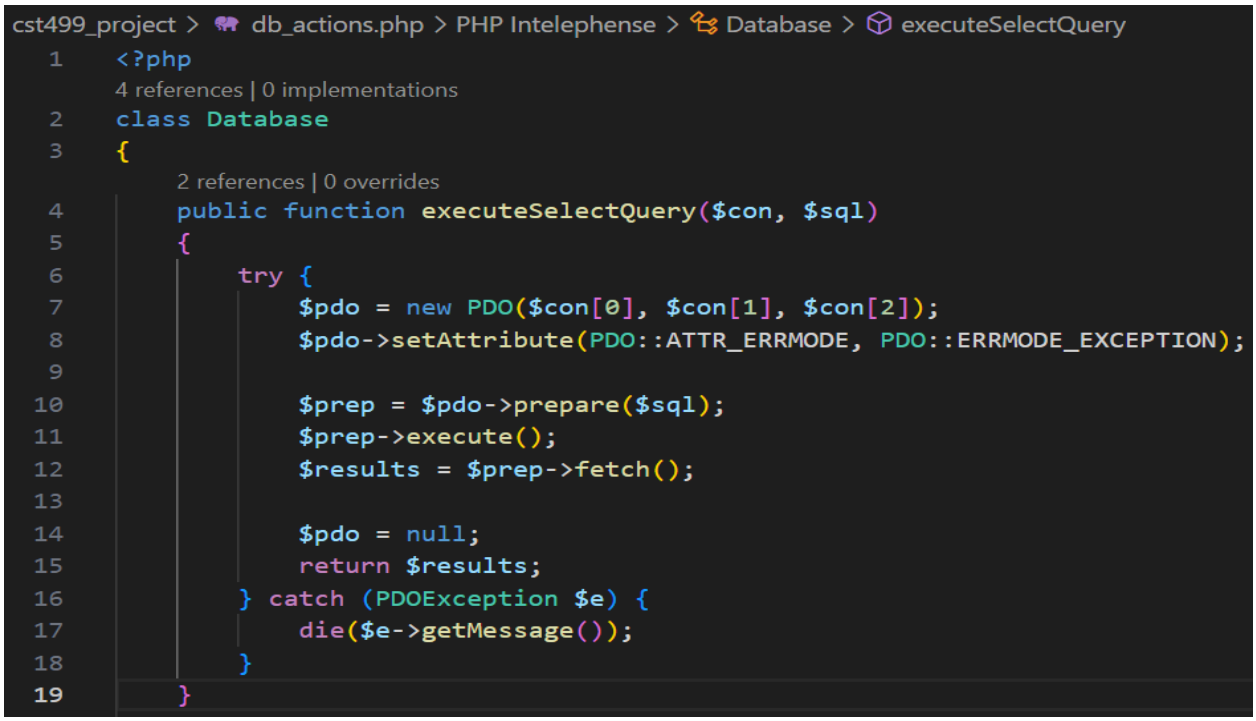
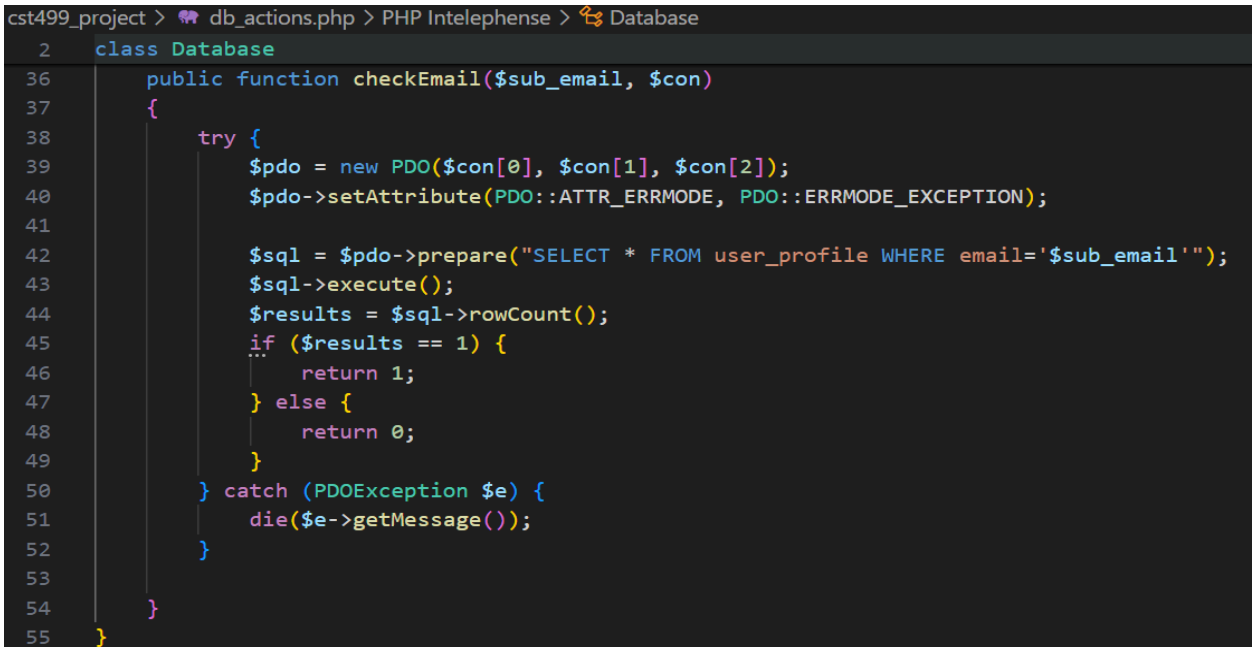


Figure 3

The checkEmail database function.



Creating the Registration Page

The form submits the entered information via POST, triggering the execution of insert_db.php. Figures 4 and 5 show the `registration.php` interface where users input their personal details and password. Figures 6 and 7 depict the insert_db.php process, which parses the submitted data, retrieves the database credentials, verifies the email address, and then displays the results.

Figure 4

The registration.php part 1 screenshot.

Figure 5

The registration.php part 2 screenshot.

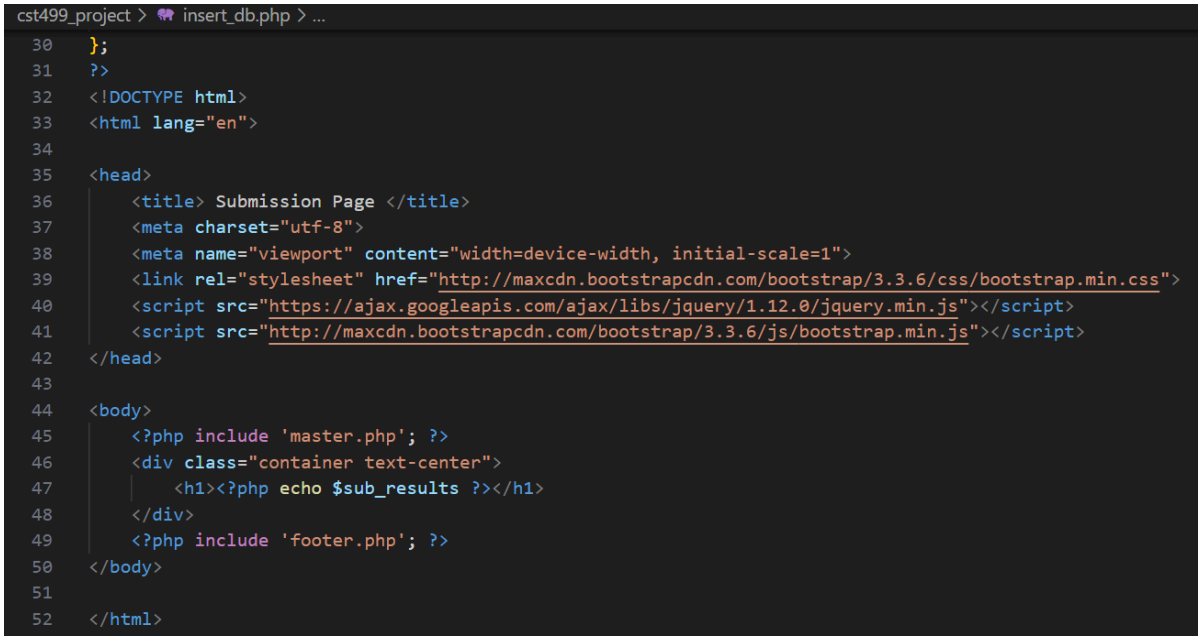
Figure 6

The insert db.php source code part 1 screenshot.

```
cst499_project > insert_db.php > ...
1   <?php
2   error_reporting(E_ALL ^ E_NOTICE);
3   include 'db_actions.php';
4   require_once ('protected/db_config.php');
5   $host = DBHOST;
6   $db = DBNAME;
7   $uname = DBUSER;
8   $db_pass = DBPASS;
9   $con = ["mysql:host=$host;dbname=$db", "$uname", "$db_pass"];
10
11  $db_action = new Database();
12
13  $fname = $_POST['fname'];
14  $lname = $_POST["lname"];
15  $email = $_POST["email"];
16  $pass = password_hash($_POST["password"],PASSWORD_DEFAULT);
17  $phone = $_POST["phone"];
18  $sub_results;
19
20  //check to see if email already in use
21  if ($db_action->checkEmail($email, $con) == 0) {
22      $sql = "INSERT INTO user_profile (`first_name`,`last_name`,`email`,`pass`,`phone`)
23      VALUES ('$fname', '$lname', '$email', '$pass', '$phone')";
24
25      $db_action->executeQuery($con, $sql);
26
27      $sub_results = 'The information has been saved successfully';
28  } else {
29      $sub_results = 'The email already exists. Please use a different email.';
```

Figure 7

The insert db.php source code part 2 screenshot.

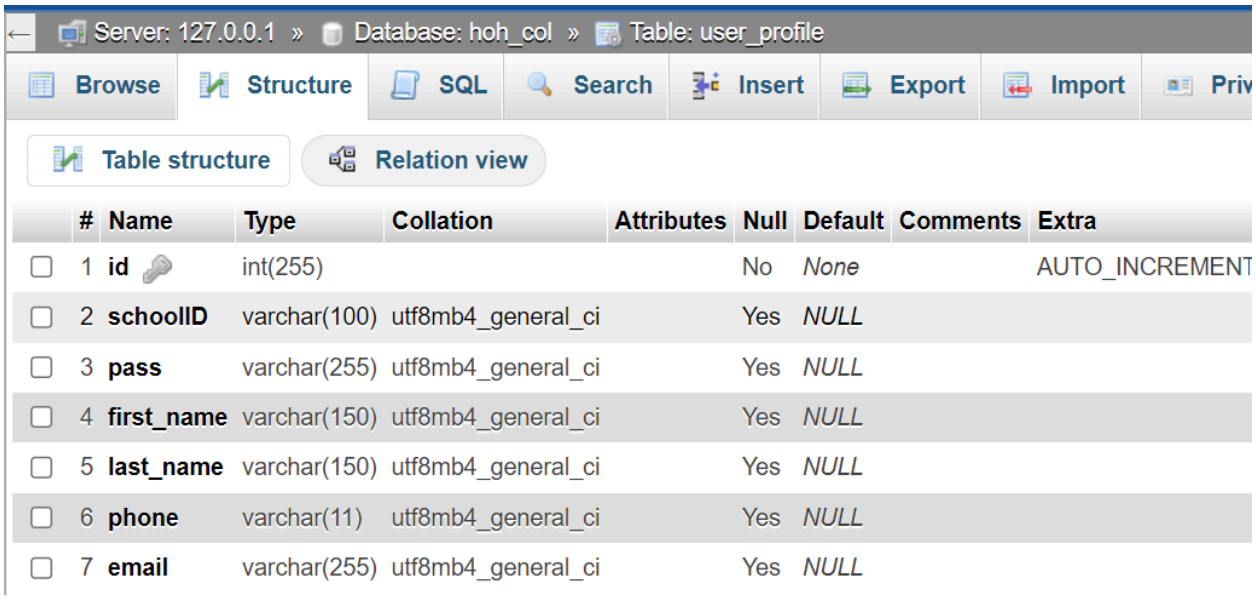


```
cst499_project > insert_db.php > ...
30  };
31  ?>
32  <!DOCTYPE html>
33  <html lang="en">
34
35  <head>
36      <title> Submission Page </title>
37      <meta charset="utf-8">
38      <meta name="viewport" content="width=device-width, initial-scale=1">
39      <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
40      <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
41      <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
42  </head>
43
44  <body>
45      <?php include 'master.php'; ?>
46      <div class="container text-center">
47          <h1><?php echo $sub_results ?></h1>
48      </div>
49      <?php include 'footer.php'; ?>
50  </body>
51
52  </html>
```

Project Screenshots

Figure 8

The table used for the project.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
☐	1	id	int(255)			No	None		AUTO_INCREMENT
☐	2	schoolID	varchar(100)	utf8mb4_general_ci		Yes	NULL		
☐	3	pass	varchar(255)	utf8mb4_general_ci		Yes	NULL		
☐	4	first_name	varchar(150)	utf8mb4_general_ci		Yes	NULL		
☐	5	last_name	varchar(150)	utf8mb4_general_ci		Yes	NULL		
☐	6	phone	varchar(11)	utf8mb4_general_ci		Yes	NULL		
☐	7	email	varchar(255)	utf8mb4_general_ci		Yes	NULL		

Figure 9

Test data from testing code for the project.

id	schoolID	pass	first_name	last_name	phone	email
1	NULL	$2y$10$rGoj8gNfMpBwmS7O/laKKekFBZBUeHS954WtpLsAxUO...	Test	Student	1234567	123@mail.com

Figure 10

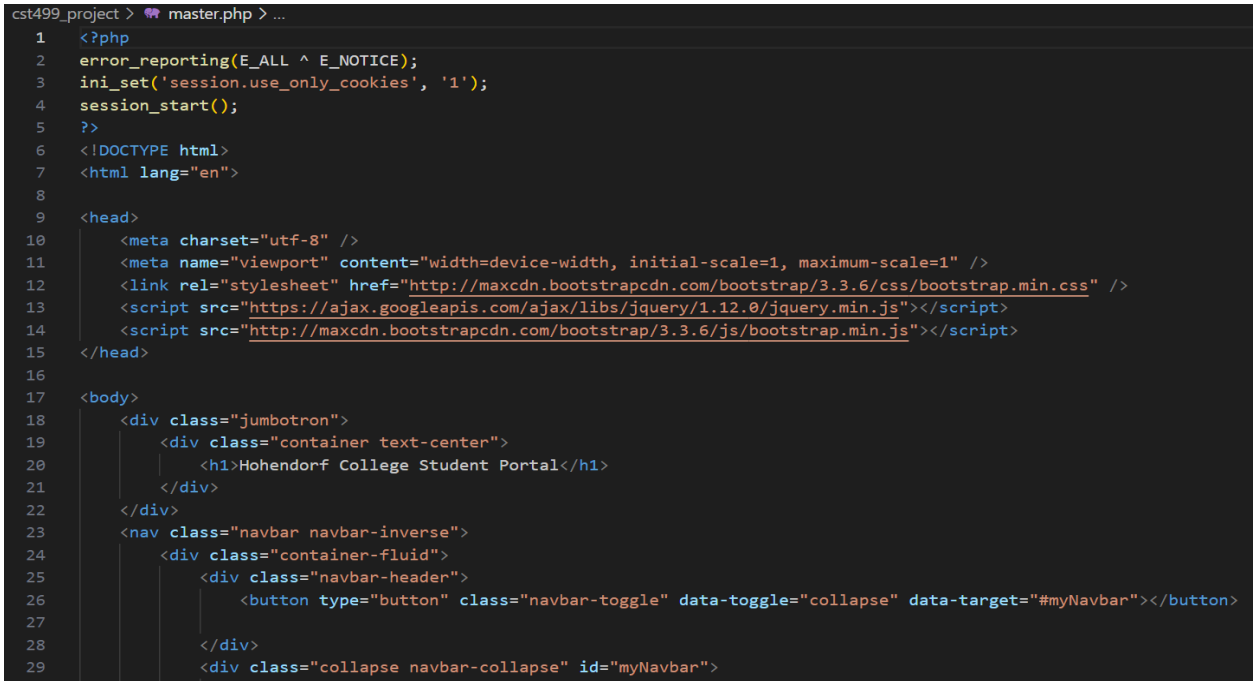
The index.php source code screenshot.

cst499_project > index.php > ...

```php
<?php
error_reporting(E_ALL ^ E_NOTICE);

if(isset($_GET['Logout'])){
    $_SESSION = array();
    $params = session_get_cookie_params();
    setcookie(session_name(), '', time() - 42000, $params["path"], $params["domain"], $params["secure"], $params["httponly"]);
    header('Location: index.php');
}
?>
<!DOCTYPE html>
<html lang="en">

<head>
    <title>Home Page</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
    <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
</head>

<body>
    <?php require 'master.php'; ?>
    <div class="container text-center">
        <h1>Welcome to the Hohendorf College Student Portal</h1>
    </div>
    <?php require_once 'footer.php'; ?>
</body>
```

Figure 11

The master.php source code part 1 screenshot.



```
cst499_project > master.php > ...
1   <?php
2   error_reporting(E_ALL ^ E_NOTICE);
3   ini_set('session.use_only_cookies', '1');
4   session_start();
5   ?>
6   <!DOCTYPE html>
7   <html lang="en">
8
9   <head>
10      <meta charset="utf-8" />
11      <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1" />
12      <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />
13      <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
14      <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
15  </head>
16
17  <body>
18      <div class="jumbotron">
19          <div class="container text-center">
20              <h1>Hohendorf College Student Portal</h1>
21          </div>
22      </div>
23      <nav class="navbar navbar-inverse">
24          <div class="container-fluid">
25              <div class="navbar-header">
26                  <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar"></button>
27
28              </div>
29              <div class="collapse navbar-collapse" id="myNavbar">
```

Figure 12

The master.php source code part 2 screenshot.

```
cst499_project > master.php > html
7   <html lang="en">
17  <body>
23      <nav class="navbar navbar-inverse">
24          <div class="container-fluid">
29              <div class="collapse navbar-collapse" id="myNavbar">
30                  <ul class="nav navbar-nav">
31                      <li class="active"><a href="index.php"><span class="glyphicon glyphicon-home"></span> Home</a></li>
32                      <li><a href="#"><span class="glyphicon glyphicon-exclamation-sign"></span> About Us</a></li>
33                      <li><a href="contact_us.php"><span class="glyphicon glyphicon-earphone"></span> Contact Us</a></li>
34                  </ul>
35                  <ul class="nav navbar-nav navbar-right">
36                      <?php
37                      if (isset($_SESSION['email'])) {
38                          echo '<li><a href="profile.php"><span class="glyphicon glyphicon-briefcase"></span> Profile</a></li>';
39                          echo '<li><a href="index.php?Logout=1"><span class="glyphicon glyphicon-off"></span> Logout</a></li>';
40                      } else {
41                          echo '<li><a href="login.php"><span class="glyphicon glyphicon-user"></span> Login</a></li>';
42                          echo '<li><a href="registration.php"><span class="glyphicon glyphicon-pencil"></span> Register</a></
                            li>';
43                      }
44                      ?>
45                  </ul>
46              </div>
47          </div>
48      </nav>+
49  </body>
50
51  </html>
```

Figure 13

The login.php source code screenshot.

References

Mikoluk, K. (2013, September 18). *XAMPP tutorial: How to use XAMPP to run your own web server*. Udemy. https://blog.udemy.com/xampp-tutorial

SiteGround. (n.d.). *phpMyAdmin create and populate tables tutorial*. https://www.siteground.com/tutorials/phpmyadmin/create-populate-tables/

Tsui, F., Karam, O., & Bernal, B. (2018). *Essentials of software engineering* (4th ed.). Jones & Bartlett Learning.