M4

2.5 Code Review

Our coding style is enforced by our code editor but we also make sure that our code is indented, commented, and space out for ease of understanding.

<?php

*// Trang Leminh: Data base connection is clear to me although I think it would be good to have a separate database.php so that you can just include() every time you need it*

*// connecting to the DB*

session\_start();

$conn = new mysqli('localhost', 'root', 'root','cen4010\_fa21\_g07');

if ($conn->connect\_error) {

        die("Connection failed: " . $conn->connect\_error);

}

echo "Connected successfully";

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>LikU</title>

</head>

<body>

<?php

*// Trang Leminh: I believe when you session\_start() the session either starts/or resumes the current session. Session variables are also saved during this head. Good to set variables here*

echo $\_SESSION['id'];

echo "Session variables are set.";

$sql = "SELECT id, user\_id, caption,img FROM posts ";

$result = $conn->query($sql);

if(isset($\_POST["submit"])&& $\_POST['randcheck']==$\_SESSION['rand']){

        $name = $\_FILES['image']['name'];

        $imgData = addslashes(file\_get\_contents($\_FILES['image']['tmp\_name']));

        $upload = "INSERT INTO posts (user\_id, caption, img, like1, like2, like3, like\_total) VALUES('$\_SESSION[id]','$\_POST[caption]','$imgData ',0,0,0,0 )";

        $hope = $conn->query($upload);

        if($hope)

        {

                echo '<script type="text/javascript"> alert("Data Inserted Seccessfully!"); </script>';    *// alert message*

        }

        else

        {

                echo '<script type="text/javascript"> alert("Error Uploading Data!"); </script>';    *// when error occur*

        }

}

?>

// updating post!!

<?php

**function** test($conn){

        $sql1 = "UPDATE posts SET like1=like1 + 1 WHERE id = 1 ";

        $conn->query($sql1);

        if ($conn->query($sql1) === TRUE)

{

                echo "Record updated successfully";

        }

else

{

                echo "Error updating record: " . $conn->error;

        }

}

?>

<?php if ($result->num\_rows > 0) {?>

// output data of each row

<?php while($row = $result->fetch\_assoc()) {?>

*<!-- Trang Leminh: looks good; fetching every instance for users, displaying them with the following information -->*

                <br><a href="profile.php?name=<?php echo $row['user\_id'] ?>">user's profile page</a>

                <br><?php echo '<img src="data:image/jpeg;base64,'.base64\_encode( $row['img'] ).'"/>';?>

                <br><?php echo htmlspecialchars($row['caption']); ?>

                <br><button onclick = "">1st like</button>

                <button>2nd like</button>

                <button>3rd like</button>

        <?php }?>

<?php } else {?>

<?php echo "0 results";?>

<?php } ?>

<form method="post" enctype="multipart/form-data">

        <?php

        $rand=rand();

        $\_SESSION['rand']=$rand;?>

        <input type="hidden" value="<?php echo $rand; ?>" name="randcheck" />

*<!-- Trang Leminh: Clear table formatting and easy to understand the input fields and association with the backend code -->*

        <table>

                <tr>

                    <td>Select Image</td>

                    <td><input type="file" name="image" Required></td>

                </tr>

                <tr>

                <td>Enter Caption</td>

                    <td><input type="text" name="caption" placeholder="Enter Caption" Required></td>

                </tr>

                <tr>

                   <td colspan="2"><input type="submit" name="submit" value="Upload"></td>

                </tr>

        </table>

</form>

<br><a href="profile.php?name=<?php echo $\_SESSION['id'] ?>">view your profile</a>

<br><a href="logout.php">Logout</a>

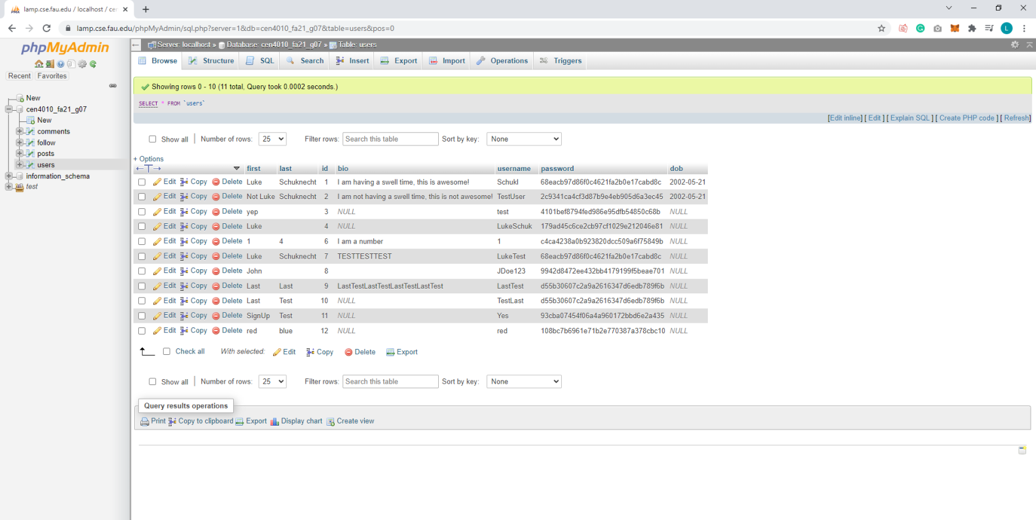
</body>

</html>

* 1. Self-check

List of assets are password, first and last name, post, comments

Password is encrypted:



List of what is being validated: ‘select Image’ – required, ‘Enter Caption’ – required, validate data form with trim, htmlspecialchars, and striplashes

2.7 Self-check: Adherence to original Non-functional specs

**Responsiveness**:DONE

**Response Time**: DONE

**Test Requirements:** DONE

**Training**: DONE

**Browser Compatibility**: DONE

**Computer and OS Compatibility**: DONE

**Login/Password System:** DONE

**Resource Utilization:** DONE

**Platform Compatibility:** DONE

**Coding Standards**: DONE

**Naming Conventions**: DONE

**Accessible Times:** DONE

**Downtime Impact**: DONE

**Rating:** DONE