

Lab 08: Working with Databases – PHP and MySQL

Task 1: Create database to store test scores

(a) Setup database:

- i. Create a new MySQL database named “lab8”.
- ii. Create two new tables named “students” and “scores” in “lab8” database to store the data for students and scores.
- iii. Insert some records into “students” table.
- iv. Create a new user with the username ('user1') and password ('user1abc'). Grant the user to have privilege to access the “lab8” database.

NOTE: Refer to the `task1.sql` file for the SQL statements to create database, create tables and insert new records into table.

(b) Create a new PHP file named “config.php” file and write the PHP codes with **PDO** statements to connect to MySQL database and handle connection errors. The database information is below:

- i. Database Host Name = 'localhost';
- ii. Database Name = 'lab8';
- iii. Username = 'user1';
- iv. Password = 'user1abc';

(c) Modify the HTML form that you have written in Lab07 – Task 1: “addScores.php”:

- i. Add a new <input> element with option values in a <datalist> to allow users to select a **Student ID** to enter the test scores.
- ii. Include the `config.php` file to connect to the database and handle connection errors.
- iii. Write the PHP codes with **PDO** statements to:
 - Retrieve all the Student IDs from the “students” table using **SELECT** SQL statement.
 - Write a while loop to print all the retrieved Student IDs as the <option value> in the <datalist>. Figure 1 shows the sample screenshots for the list of Student IDs in a <datalist> in “addScores.php”.

```
<!DOCTYPE html>
<html>
<body>
<h1>Enter Scores:</h1>
<form method="POST" action="grader.php" >
    <input list="students">
    <datalist id="students">
        //Write php codes to generate list from 'students' table.
        <option value="StudentID">
    </datalist>
    Score 1 <input type="text" name="score1"/><br/>
    Score 2 <input type="text" name="score2"/><br/>
    Score 3 <input type="text" name="score3"/><br/>
    Score 4 <input type="text" name="score4"/><br/>
    Score 5 <input type="text" name="score5"/><br/>
    Score 6 <input type="text" name="score6"/><br/><br/>
    <input type="submit">
</form>
</body>
</html>
```

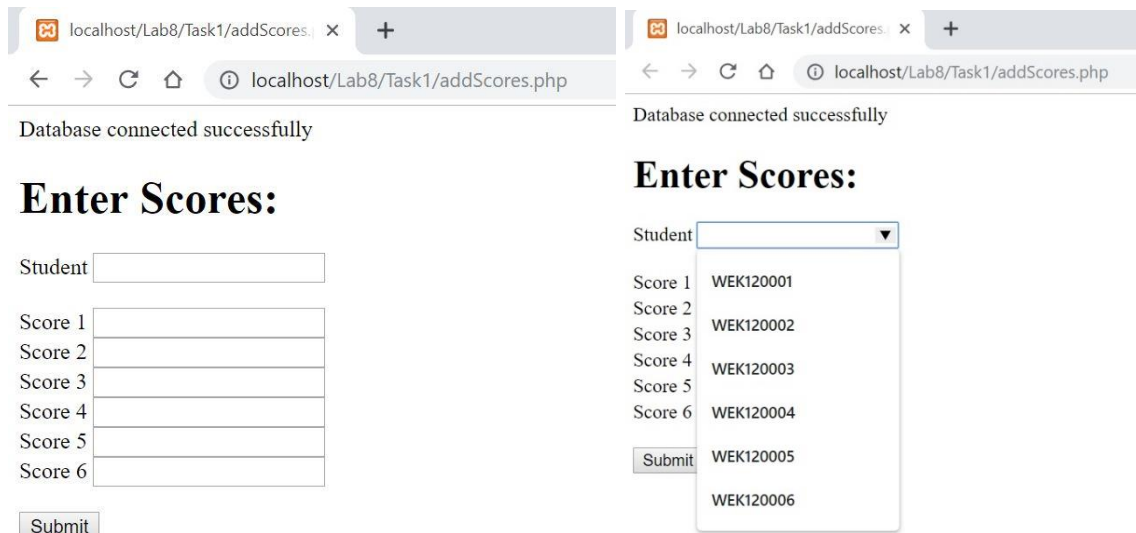


Figure 1: Screenshots for addScores.php

(d) Figure 2 shows the sample screenshot for “grader.php”. Modify the PHP program that you have written in Lab07 – Task 1: “grader.php” to do the following tasks:

- i. Define a new variable named `$studentID` to store the Student ID sent from the form with the POST method in the “addScores.php” page.
- ii. Include the `config.php` file to connect to the database and handle connection errors.
- iii. Write PHP codes with **PDO statements** to:
 - insert a new record into “scores” table using the **INSERT** SQL statement;
 - get ScoreID of last inserted record in the “scores” table and define a new variable named `$last_id` to store the id.
 - print a statement to show that a new record was created successfully;
 - free resources and closing database connection;
- iv. Provide a link to allow navigation to “View Score” and send the last inserted ScoreID (`$last_id`) to `viewScores.php` using the GET method:
`View Scores`
- v. Provide a link to allow navigation to “Add New Score”:
`Add New Scores`

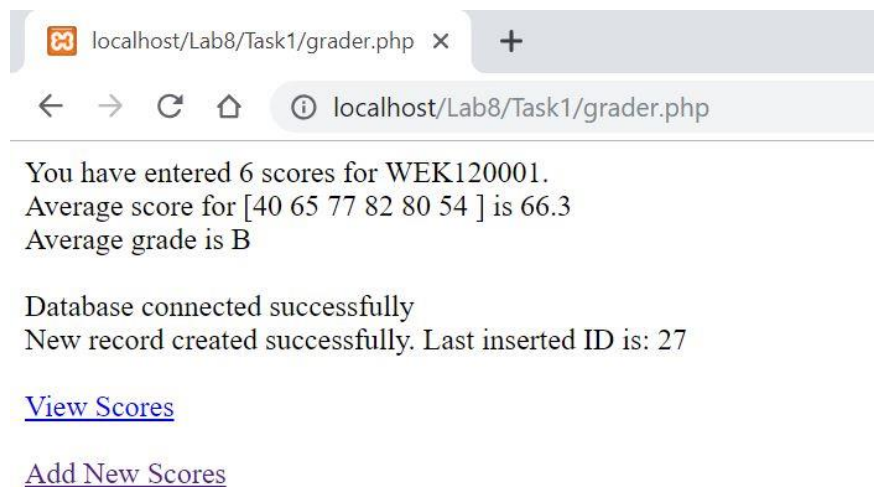


Figure 2: Screenshot for grader.php

(e) Create a new PHP file named “**viewScores.php**” to allow users to **view the scores and choose to update or delete** the record. Figure 3 shows the sample screenshot for “viewScores.php”. Write PHP codes to do the following tasks:

- i. Include the `config.php` file to connect to the database and handle connection errors.
- ii. Define a new variable named `$id` to store the **ScoreID** sent from the form with the POST method in the “addScores.php” page.
- i. Write PHP codes with **PDO statements** to:
 - Select all the data associated with this particular ScoreID in the “scores” table using the **SELECT** SQL statement.
 - Define variables to store all the data retrieved from the database.
 - Create an html form with the following parameters:
`<form method="post" action="editScores.php">` and inputs:
 - 6 text inputs for the 6 scores,
 - 2 hidden inputs for ScoreID and StudentID.
 - 2 submit buttons: **Update** and **Delete**,`<input type="submit" name="update" value="Update">`
`<input type="submit" name="delete" value="Delete">`
 - free resources and closing database connection.

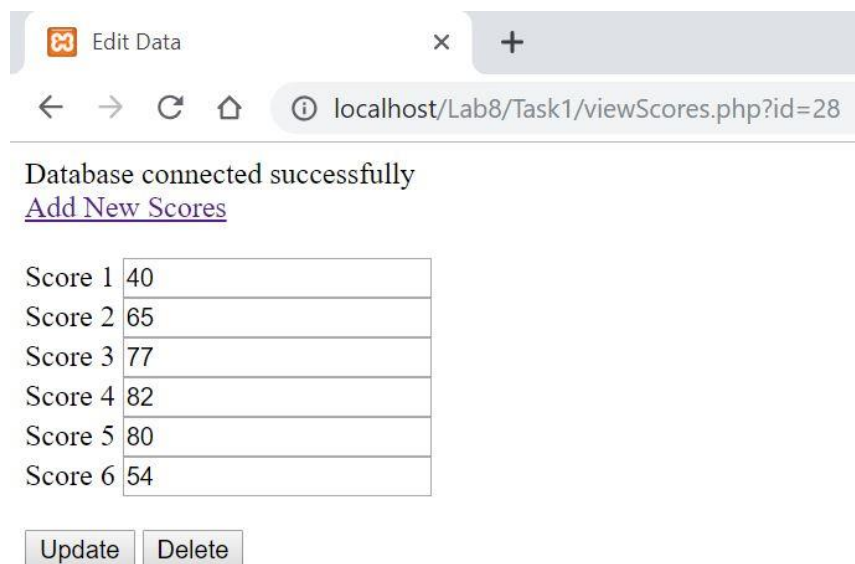


Figure 3: Screenshot for viewScores.php

(f) Create a new file named “**editScores.php**” to handle either the update or delete request from users. Write PHP codes with if-else statements to handle the two requests:

- i. **Update request** `if(isset($_POST['update']))`: Write PHP codes with **PDO statements** to get the ScoreID, StudentID and updated score values sent from the `viewScores.php` form and update the record in the database using the **UPDATE** SQL statement. Print a message to show whether the update is successful or failed.

Figure 4 shows the sample screenshot for “editScores.php” that handle the update request when the user click the **Update** button.

- ii. **Delete Request** `else if(isset($_POST['delete']))`: Write PHP codes with **PDO statements** to get the ScoreID sent from the viewScores.php form and delete the record in the database using the **DELETE** SQL statement. Print a message to show whether the update is successful or failed.

Figure 5 shows the sample screenshot for “editScores.php” that handle the delete request when the user clicks the **Delete** button.



Figure 4: Screenshot for editScores.php (Handle Update request)

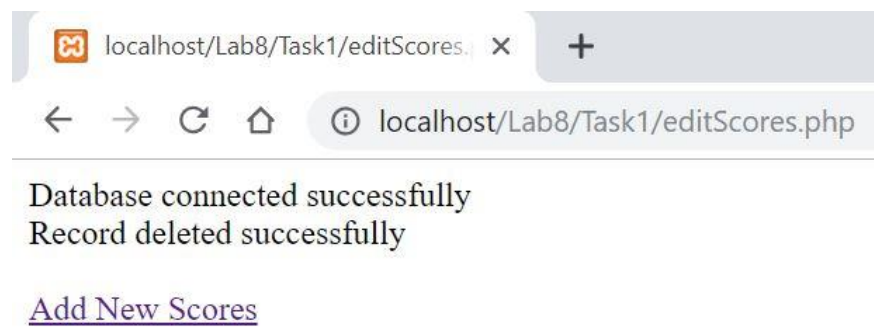


Figure 5: Screenshot for editScores.php (Handle Delete request)

References:

https://www.w3schools.com/tags/tag_datalist.asp

https://www.w3schools.com/php/php_mysql_insert_lastid.asp

Task 2: Using PHP in SE Club website:

NOTE: Create new folder and files

- ❖ Create a new folder on your hard drive or portable storage device (USB or SD card) called “**seclub7**”. Copy **all** the files from your Lab 2’s folder (seclub6) into the “**seclub7**” folder.
- ❖ Create a new php file named “**profile.php**”.
- ❖ Create a new php file named “**config.php**”.

a) Setup database:

- Create a new MySQL database named “seclub”.
- Create a new table named “members” in the “seclub” database to store the new member information.

b) Open the “config.php” file and write the PHP code using **MySQLi statements** to connect to database and handle connection errors. The database information is below:

- Database Host Name = 'localhost';
- Database Name = 'seclub';
- Username = 'user';
- Password = 'userpwd';

c) Open the signup.html file and modify the form action from “/sign-in-url” to “processSignup.php”.

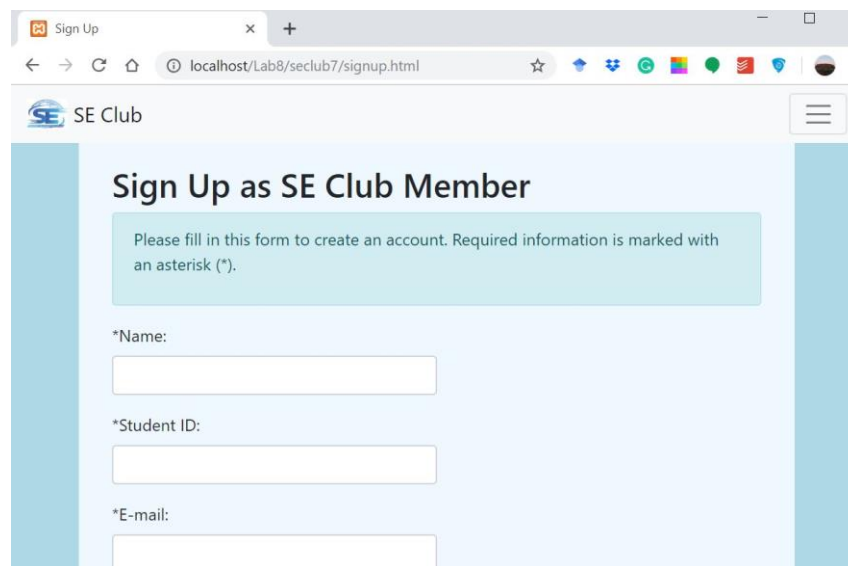
When the user fills out the form in “signup.html” and clicks the submit button, the form data is sent for processing to a PHP file named “processSignup.php”. The form data is sent with the HTTP POST method.

d) In the “processSignup.php” file, modify PHP codes to:

- i. Remove the codes using the PHP File functions (e.g. fopen, fwrite, fclose, file_put_contents).
- ii. Include the database connection file: `include_once("config.php");`
- iii. Write PHP codes with **MySQLi statements** to:
 - Check email and student ID to validate whether the member already exists. Print a message to show that the account already exists (e.g. Email or Student ID already exists). Provide links to allow navigation to “Sign Up” and “Log In” pages.
 - If it is a new member, insert the new record into “members” table using the **INSERT** SQL statement. Print a message to show that the registration is successful. Provide a link to allow navigation to “Log In” page.
 - Free resources and closing database connection;
- iv. **store the input data** (Name, Student ID, email, password, birthday, most favourite event), sent from the previous page, **Sign Up** form with the POST method **in the database, seclub’s member table**.
- v. Redirect the page to **login.php** page.

e) In the “login.php” file, write HTML and PHP codes to include a sign in form with 2 text inputs (email address and password) and 1 submit button named “Sign In”.

Figures 6, 7, 8 and 9 show the screenshots of the “signup.html”, “processSignup.php”, “login.php”.



A web browser window showing the 'Sign Up' page for the SE Club. The browser's address bar displays 'localhost/Lab8/seclub7/signup.html'. The page features a light blue header with the 'SE Club' logo and a hamburger menu icon. The main content area has a light blue background and contains the heading 'Sign Up as SE Club Member'. Below the heading is a light blue box with the text: 'Please fill in this form to create an account. Required information is marked with an asterisk (*)'. The form consists of three input fields, each preceded by an asterisk: '*Name:', '*Student ID:', and '*E-mail:'. The input fields are empty and have a light yellow border.

Figure 6: SE Club’s Sign Up page

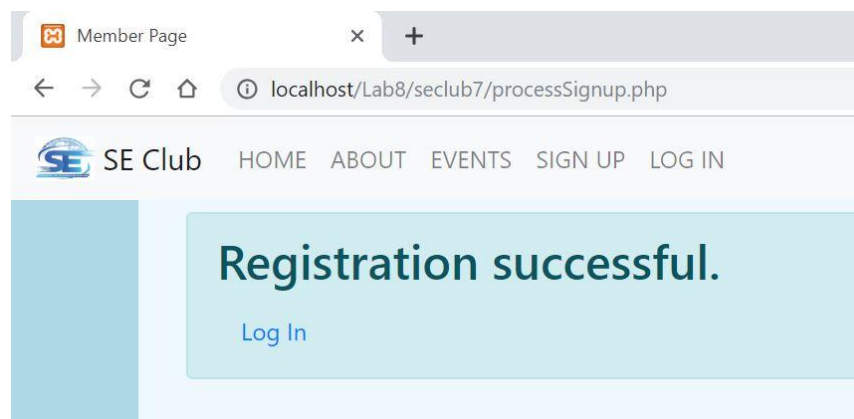


Figure 7: SE Club’s processSignup.php (Registration successful)

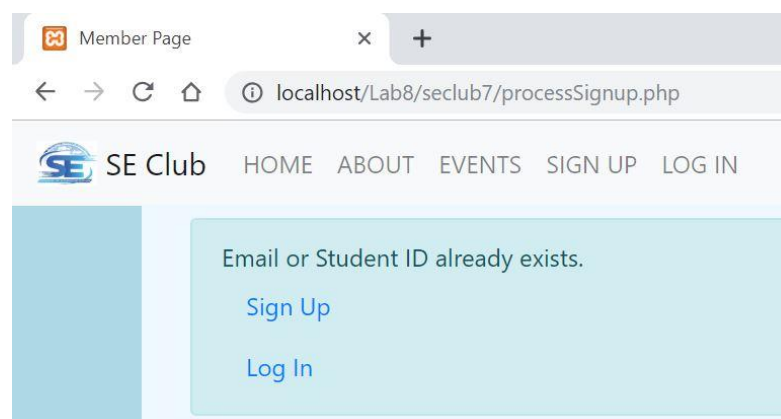
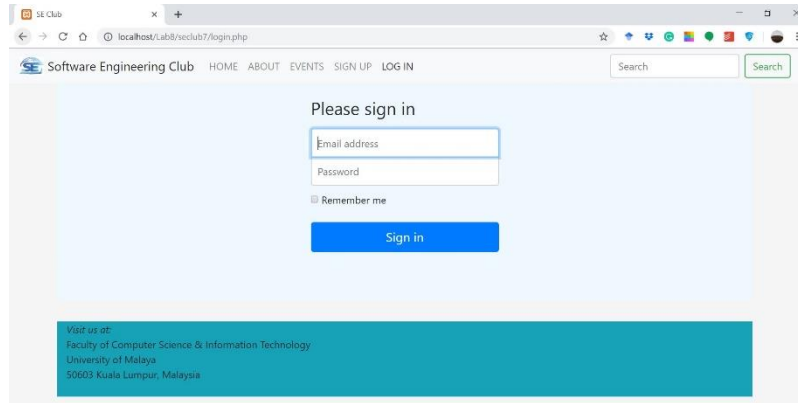


Figure 8: SE Club’s processSignup.php (Registration failed – account already exists)

WIF2003 Web Programming



The image shows a web browser window displaying the login page for the Software Engineering Club (SE Club). The browser's address bar shows the URL `localhost/lab8/seclub7/login.php`. The page has a light blue header with the club's logo and navigation links: HOME, ABOUT, EVENTS, SIGN UP, and LOG IN. A search bar is located on the right side of the header. The main content area is a light blue box with the heading "Please sign in". Below this heading are two input fields: "Email address" and "Password". There is a checkbox labeled "Remember me" and a blue "Sign in" button. At the bottom of the page, there is a teal footer with the text: "Visit us at: Faculty of Computer Science & Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia".

Software Engineering Club HOME ABOUT EVENTS SIGN UP LOG IN Search

Please sign in

Email address

Password

☐ Remember me

Sign in

Visit us at:
Faculty of Computer Science & Information Technology
University of Malaya
50603 Kuala Lumpur, Malaysia

Figure 9: SE Club's login.php