SYSC4504

Fundamentals of Web Development (Winter 2023)

Lab 5 – PHP and Databases (MySQL)

	Name	Email	Section	Responsibility
TA:	Amir Ghorab	amirhoseeinghorab@cmail.carleton.ca	L1O, L3O	***Res 1***
	Omar Imran	OmarImran@cmail.carleton.ca	L1O, L2O	***Res 1***
	Nihal Kulkarni	nihalkulkarni@cmail.carleton.ca	L3O	***Res 1***
	Kristen Beange	kristenbeange@cmail.carleton.ca	L2O	***Res 1 & Res 2***

TA Responsibilities:

Res 1. Lab Administration (answering questions and attendance)
Res 2. Grading submitted work

Marking scheme:

You can discuss your solution with other students; however, you have to submit your own work, you cannot submit the exact same work (solution) of other students.

The marking contains two components:

1. Lab attendance and it is graded as follows (**1 point**):
   - You must answer questions from the TA during the lab time to confirm your understanding of the topics covered
     - You will receive **Un-Satisfactory (U)** grade if you only finish questions 1 & 2 (0/1)
     - You will receive **Marginal (M)** grade if you only finish questions 1 to 3 (1/1)
     - You will receive **Satisfactory (S)** grade if you finish all the questions (1/1)
2. Your submitted work will be evaluated based on the completeness and functionality (**2 points**).
   - Refer to the "**File Submission**" section at the end of this document for more details.

Table of Contents

I. Introduction

In this lab, you will continue working on the Fan Page project you started in previous labs. The lab is divided into two parts; General MySQL examples and MySQL and PHP examples related to your Fan Page project.

NOTE: It is recommended to go over chapters 8, 9, 12 and 14 from the textbook before attempting the lab.

II. Objective

Apply the theory that was learnt during lectures about Databases (MySQL) and PHP in adding features to the website created in previous labs.

III. General MySQL exercises

In this exercise, you will go through SQL examples to practice what you have learnt during the Database lectures.

- Preparation

1. Inside your project's folder, create a folder named "MySQL".
2. Inside that folder, create a file named "firstname_lastname_lab05.sql"
3. Open the file using a text editor of your choice. If you installed notepad++ or BBEdit, you can use these.
4. **IMPORTANT:** Enter all your SQL commands in the sql file. Add a comment to identify which requirement each statement referring to.

- Accessing the MySQL server and database using phpMyAdmin

1. Open XAMPP
2. Similar to running the PHP Apache server, start MySQL Database server.
3. To access the MySQL database, open "http://localhost/phpmyadmin/"
4. You will be asked for your phpMyAdmin admin username and password
   - If you are using Windows, you would have been asked for creating a username and password while installing XAMPP. If you didn't change the username and password during the installation, the default username is "root" and the password is empty.
   - If you are using Windows and don't remember your username/password, check the instructions for MacOS below for details.
   - If you are using MacOS, you need an extra step:
     a. Open XAMPP
     b. Click on "Open Application Folder" under the Welcome Tab.
     c. Open "phpmyadmin" folder
     d. Open "config.inc.php" in a text editor

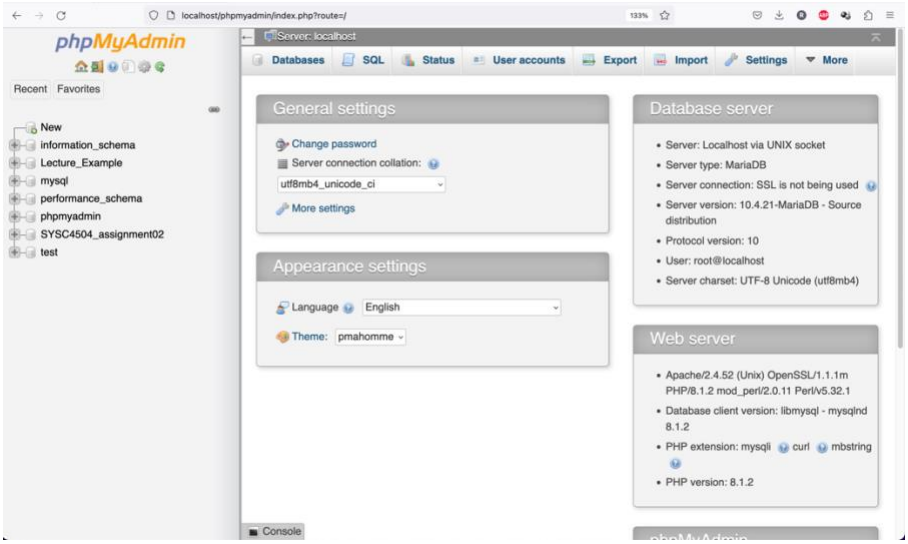
e. Find the comment "/* Authentication type */". Should be around line 27
f. Change the authorization type from **config** to **HTTP**

```
/* Authentication type */
$cfg['Servers'][$i]['auth_type'] = 'config';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = '';
```

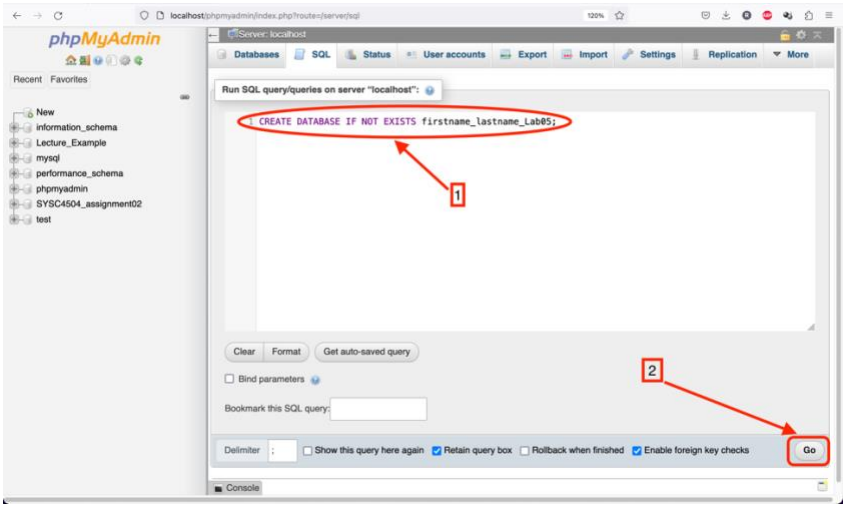
Change to HTTP

g. It is recommended to set the password and not keeping it empty

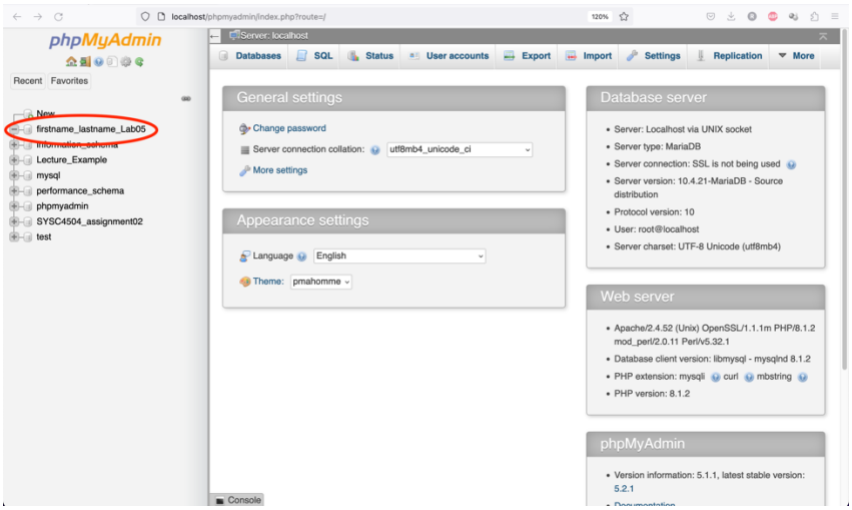
5. If you logged in to the server and had no issues, you should get something similar to the screenshot below



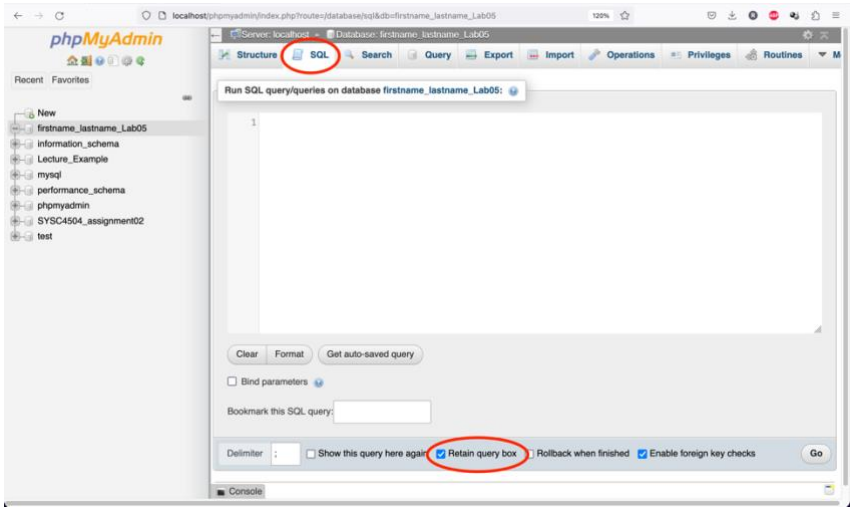
6. Click on "SQL" tab at the top of page.
7. Create a new database and name it "firstname_lastname_Lab05".
   - **Note:** You need to use the Create Database SQL instruction



8. Copy the SQL statement into the .sql file you created at the beginning of the lab.
9. The newly created database should appear on the left panel.
10. Click on the new database schema in the left panel



11. Click on SQL and make sure you select the option "Retain query box" otherwise you will have to click on SQL tab after every sql statement you run



- Creating General Tables
  - **NOTES:**
    - For each of the following examples, there are two questions. For examples 5-12, the minimum requirement for the lab is to answer the odd number of each of the questions (example: Q9, Q11, Q13, …).
    - It is recommended to finish all requirements.

1. Create Table Examples:

Q 1. Create a table named STUDENT_INFO with the following columns:

COLUMN NAME	COLUMN TYPE	PRIMARY KEY
ID	INT (10)	YES
NAME	VARCHAR (100)	NO
DOB	DATE	NO
INCOME	DECIMAL (10,2)	NO
COURSE_ID	INT (5)	NO

Q 2. Create a table named COURSE_INFO with the following columns:

COLUMN NAME	COLUMN TYPE	PRIMARY KEY
ID	INT (10)	YES
NAME	VARCHAR (100)	NO
LOCATION	VARCHAR (200)	NO
STARTDATE	DATE	NO
TYPE	VARCHAR (100)	NO

2. Alter Table Examples

Q 3. Alter the STUDENT_INFO table to add new column called ADDRESS with type of VARCHAR of 200 and default value of "NULL"

Q 4. Alter the COURSE_INFO table to extend the name column from 100 to 200

3. Rename Table Examples

Q 5. Rename the STUDENT_INFO table into STUDENT

Q 6. Rename the COURSE_INFO table into COURSE

4. Insert Examples

Q 7. Insert THREE new rows into STUDENT table with following information:

ID	NAME	DOB	INCOME	COURSE_ID	ADDRESS
1	JOHN	1998-05-01	1200	100	NORTH
2	MIKE	2000-08-15	1100.15	200	WEST
3	SAM	1997-11-01	500	100	SOUTH

Q 8. Insert FOUR new rows into COURSE table with following information:

ID	NAME	LOCATION	STARTDATE	TYPE
100	Fundamentals of Web Development	Azrieli Pavilion	2023-09-10	mandatory
300	Analytical Methods	Tory Building	2023-09-17	elective
500	Java Programming	Tory Building	2023-09-17	elective
700	C++ Programming	Patterson Hall	2023-09-10	elective

5. Update Examples

Q 9. Update income column for Student number 2 in STUDENT table so his income will be 1000 instead of 1100.15

Q 10. Update Location column for Course number 100 in COURSE table so the location will be Patterson Hall instead of Azrieli Pavilion

6. Delete Examples

Q 11. Delete Student Number 2 from STUDENT table

Q 12. Delete Course Number 300 from COURSE table

7. Select * Examples

Q 13. Select all columns from STUDENT table

Q 14. Select all columns from COURSE table

8. Select Specific Columns Examples

Q 15. Select Student name, DOB and income columns from STUDENT table
Q 16. Select Course ID, name and location columns from COURSE table

9. Select With Where Clause Examples

Q 17. Select Student ID, Name, DOB and income columns for all Students who have an income larger than 600 dollars
Q 18. Select all columns from COURSE table for courses which are elective and located in Patterson Hall.

10. Select With Join Examples

Q 19. Select Student ID, Name, DOB, Course Name from tables STUDENT, COURSE, (The two tables will be joined using COURSE_ID column)
Q 20. Select Student ID, Name, Course Name, Location and Startdate from tables STUDENT, COURSE where the student income is greater than 1000 (The two tables will be joined using COURSE_ID column)

11. Truncate Examples

Q 21. Truncate all data from STUDENT table then select * from the table
Q 22. Truncate all data from COURSE table then select * from the table

12. Drop Table Examples

Q 23. Drop table STUDENT table then select * from the table
Q 24. Drop table COURSE table then select * from the table

- Accessing the MySQL database using PHP – Part 1

Follow the instructions from Lab04 on setting up XAMPP and starting a PHP (apache) server.

1. In Lab 05's main project folder create a new file and name it "connection.php"
2. This file will retain your database information. Type the following script inside your file:

```php
<?php
    $server_name = 'localhost';
    $database_name = 'firstname_lastname_Lab05';
    $username = 'root';
    $password = '';
?>
```

3. To test your server's setup, create another .php file named "FirstConnection.php". Include the file created in the previous step into the file.
    - Refer to slide 22 from Topic 09 for an example
4. Create a new connection to the database using mysqli( ) class
5. Check if the connection was successful.
    - If **not successful**, output "Error: Couldn't connect." And display the error message
    - If **successful**, output "Connected Successfully"
6. Close the connection
7. Here is a screenshot of that code. It is recommended to try to solve it on your own before referring to the solution

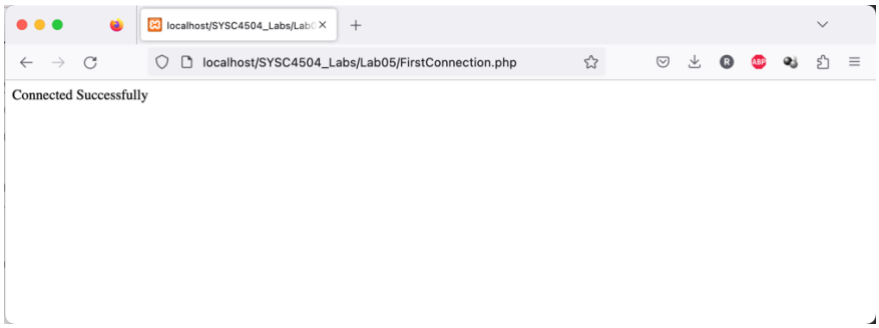
```php
<?php
    include("connection.php");      // server's information

    // create an object that connect to the Database
    $conn = new mysqli($server_name, $username, $password, $database_name);

    // check if the connection was successful
    if($conn->connect_error){
        die("Error: Couldn't connect. " . $conn -> connect_error);
    }
        echo "Connected Successfully";

    $conn -> close();   // close the connection to the database
?>
```

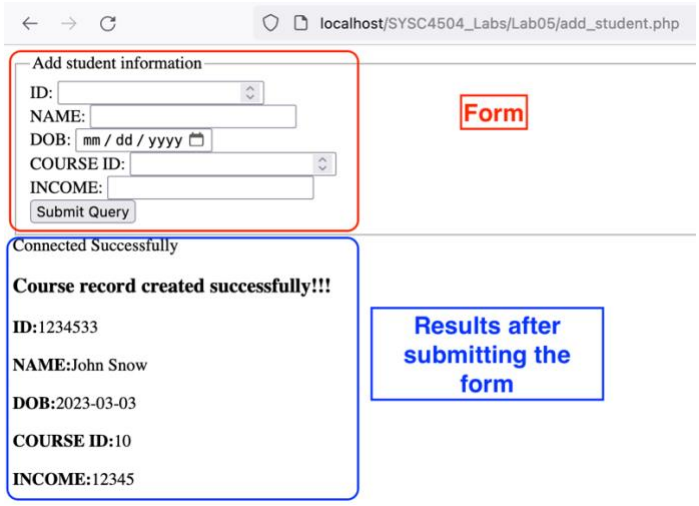
8. If the connection was successful and no had no issues, you should get something similar to the screenshot below:



- Accessing the MySQL database using PHP – Part 2

In the next exercises, we will work on accessing the tables we created earlier in our database

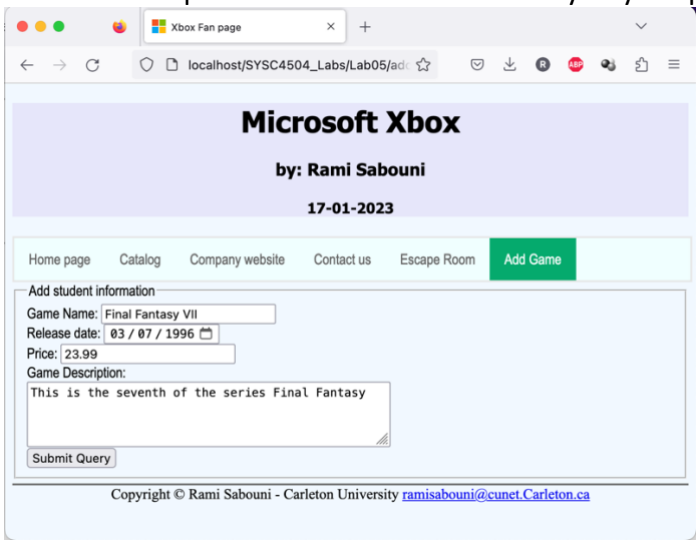
1. In the main folder, create another PHP script called "**add_student.php**" with **five** fields (ID, NAME, DOB, INCOME, COURSE_ID) and a **submit button**.
   - The form should use POST method
   - Refer to the "STUDENT_INFO" table for details on input type for each field
2. When the user submits the "add_student" form, the page should refresh and the information will be inserted into the database. The information from the database should be retrieved and displayed under the form.
   - **Notes:**
     - In order for the form to be submitted and not move to another page, you can keep the "action" of the form as empty "`action=""`".
     - You need to add a control statement after the form to check if the form has been submitted or not. You should check if the $_POST["submit"] is set or not.
3. To display the values retrieved from the database you need:
   a) Create a query (SELECT statement) and retrieve the ID submitted from the form.
      - Remember that the ID is stored in the $_POST["ID"] superglobal variable.
   b) Check if there were any results returned from the query
   c) Fetch the data retrieved using "fetch_assoc() method.
   d) Remember that the retrieved results are stored in an array and arrays in PHP have "key => value" format. (similar to dictionaries in Python)
   e) Echo each of the results to the page.
   f) Close the connection to the database



IV. Fan Page MySQL and PHP requirement

You will apply what we have learned in the previous exercises on the Fan Page you created in the previous labs.

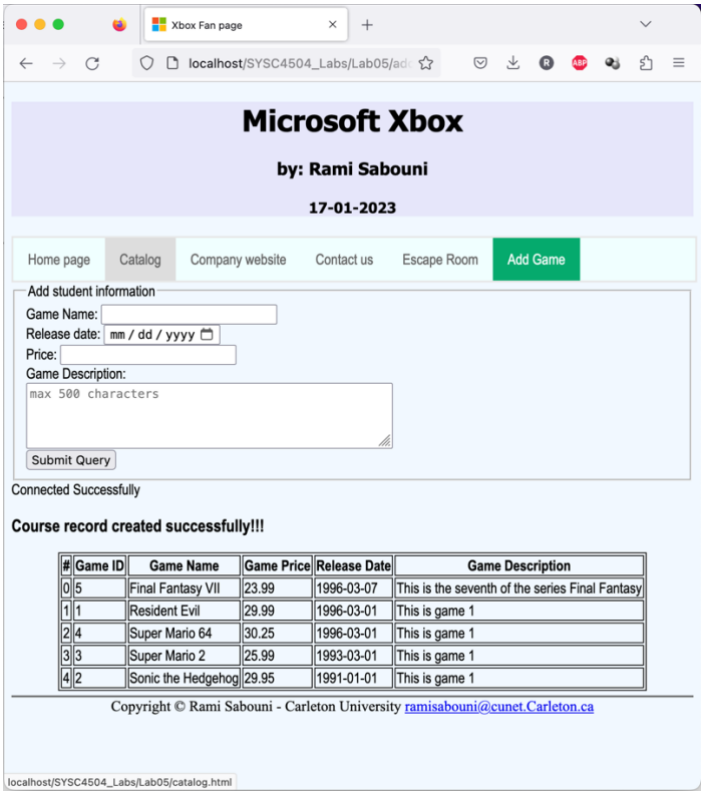
1. Make a copy of your index.html file and name it “add_game.php”. The file should be located in your project’s main folder (in the same level as your index.html).
2. Copy the provided form and replace the contents of the body of your page.



3. Create a table named game_details and add the following columns:

COLUMN NAME	COLUMN TYPE	PRIMARY KEY
game_ID	INT (10)	YES (auto increment)
game_name	VARCHAR (100)	NO
release_date	DATE	NO
game_price	DECIMAL (10,2)	NO
game_description	text	NO

4. Copy the SQL statement used to create the table into the .sql file created earlier.
5. When the user submits the “add_game” form, the page should refresh and the information will be inserted into the database.
6. The information from the database should be retrieved and displayed under the form. The data must be ordered by release date in descending order (DESC). To display the values retrieved from the database you need:



1. Create a query (SELECT statement) to retrieve all the data stored in the table.
2. Check if there were any results returned from the query
3. Fetch the data retrieved using “fetch_assoc() method.
   - Since we want to output all the records returned from the query, we can display them using a while loop
   - Every iteration of the loop, we will display one row of data
   - Remember that the retrieved results are stored in an array and arrays in PHP have “key => value” format. (similar to dictionaries in Python)
4. Close the connection to the database

V. File Submission

After you finish all the exercises:

1. Compress your project's folder and submit it to Brightspace.
   - **IMPORTANT:** The only allowed extension is: **.zip**
2. Once you are done, you must answer few questions from one of the TAs during the tutorial time to be "signed out"
3. Submit your work on Brightspace.
   - No email submissions are allowed
   - Missed deadline will result in an automatic **zero**