SYSC4504

Fundamentals of Web Development (Winter 2024)

Assignment 2

SYSCX – Part 2

	Name	Email	Responsibility
TA:	Josh Blondin	joshblondin@cmail.carleton.ca	**Marking**

Table of Contents

I. Introduction

In assignment 01, you started working on creating multiple pages (client-side) of a local version of X (Twitter) for SYSC students. In this assignment, you will work on creating the database (server-side) that will store all the data that is sent from your previously created pages.

II. Objective

Apply the theory that was learned during lectures about PHP, JavaScript and MySQL in creating the server-side part of the website.

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

III. Project Preparation

This assignment, will build up on your previous solution:

1. Download your solution from assignment 01.
2. Extract the solution and move the folder into SYSC4504_Labs folder you created in Lab 04.
3. Rename A01's folder to "`firstname_lastname_A02`". Replace `firstname` and `lastname` with your information.
   - **IMPORTANT:** the folder name should follow this format. Failure to name your folder as above will result in a zero on this assignment.
4. Change the extension of all the html files into .php. This change will allow your Apache server to recognize and run the php scripts in your pages.
   - Example: "`index.html`" becomes "`index.php`"
5. Open the folder in VSCode as usual and you can start implementing your project.

IV. Project Implementation

The Project is divided into three parts:

1. Creating your database
2. Sending data to the database
3. Displaying the retrieved information from the database

Below is the explanation for each part:

1. Creating your database (1.5 points)

In this part you need to create your database that will be used to store the data sent from the website.

a) On your server, create a database and name it "`firstname_lastname_syscx`". Replace `firstname` and `lastname` with your information.
   - **IMPORTANT:** the database name should follow this format. Failure to name your folder as above will result in a zero on this assignment.

b) Create five tables in the database:

1. **`users_info`**
   a. `student_id`
      - primary key
      - Length 10 digits
      - auto increment, starting at 100100
   b. `student_email`
      - 150 characters
   c. `first_name`
      - 150 characters

d. `last_name`
   - 150 characters

e. `dob`
   - Date

2. **`users_program`**

a. `student_id`
   - Primary/Foreign key

b. `Program`
   - 50 characters

3. **`users_avatar`**

a. `student_id`
   - Primary/Foreign key

b. `avatar`
   - Length 1 digit

4. **`users_address`**

a. `student_id`
   - Primary/Foreign key

b. `street_number`
   - Length 5 digits

c. `street_name`
   - 150 characters

d. `city`
   - 30 characters

e. `province`
   - 2 characters

f. `postal_code`
   - 7 characters

5. **`users_posts`**

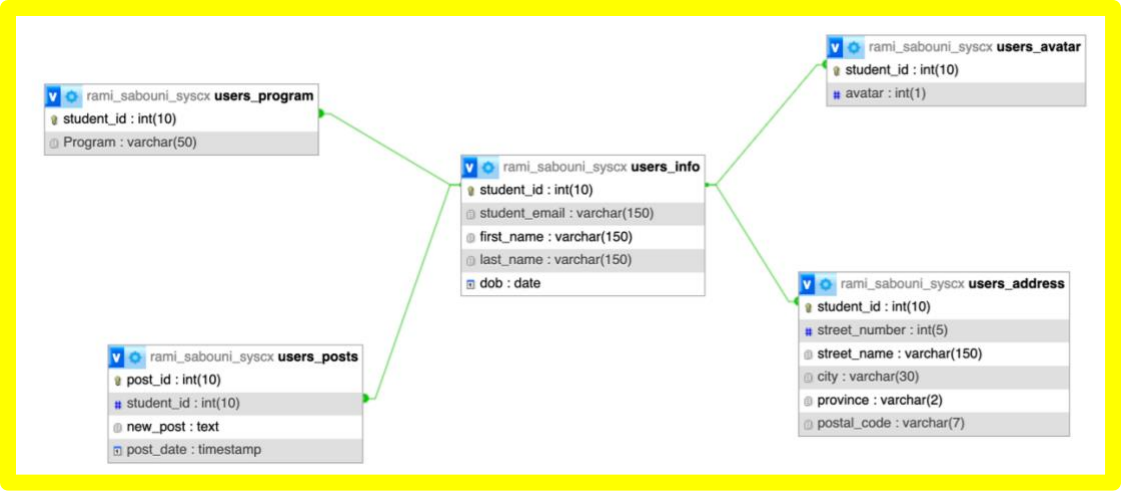
a. `post_id`
   - auto increment
   - primary key

b. `student_id`
   - Foreign key

c. `new_post`
   - Text (1000 characters)

d. `post_date (timestamp)`

c) The database structure should look similar to the screenshot below:



2. Sending data to the database (3 points)

After creating your database, you want to populate the database with data. In this part you will take the user input from the website and insert it into the database. For this assignment, you are expected to work on it in the following order:

a) `register.php` file

The information entered by the user will be added to `users_info` and `users_program`. For `users_avatar` and `users_address`, since the form didn't have any fields related to these tables, you will create a new record with the `student_Id` that was auto generated from the `users_info` table and set the string fields to NULL and the numbers to zeros.

b) `profile.php` file

The information entered by the user will be added to `user_info`, `users_program`, `users_avatar` and `users_address`.

c) `index.php` file

- The information entered by the user will be added to `users_posts`.

3. Displaying the retrieved information from the database (1.5 points)

For this assignment, you will create simple retrieval of data from the database. In Assignment 03, you will add conditions and full login system.

a) When the user registers (using `register.php`), their information is sent to the database and `profile.php` will be called. The user's information that was entered when registering will be displayed as the current values in all the `profile.php` form fields

- **Note:** 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=""`".
- **Hint:** this can be achieved by setting the value using JavaScript. Refer exercise 3 from Lab 04 for details on how to set "`input`" field's value.

b) When the user submits the profile form, the page should refresh, and the information will be updated to the database and the new values will be shown in the form.

- **Hint:** If you implement requirement (a) correctly, you should be able to build up on it so the `profile.php` page behaves based on which page/script requested it.

c) On `index.php`, when the user submits a post, the page should refresh and the last 5 posts by the user will be displayed on the `index.php` file.

V. Marking Scheme

The TA in charge of marking, will import your database into their MySQL server. They should be able to test that you correctly performed all required tasks.

- **IMPORTANT:** The marking will be done using a test harness. If your folder, file, field, or database names don't match what is listed above, there will be deduction and it might result in receiving a zero.

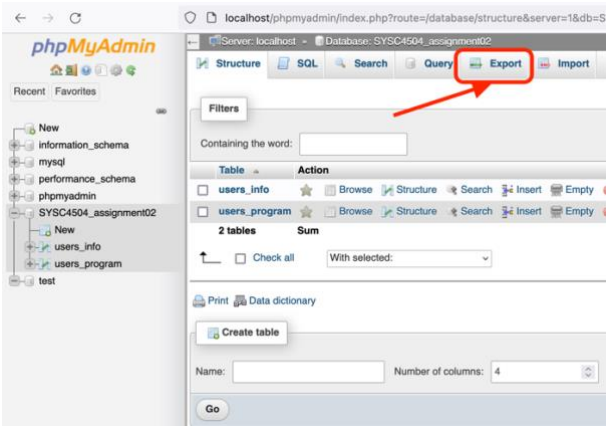
Mark division is noted next to each of the requirements listed above. The marks for each requirement will be based on:

- Completeness of your submission
- Correct solution to the problem
- Adhering to the submission requirements

VI. File Submission

After you finish all the requirements:

1. Export the database that was created as a "`.sql`" file

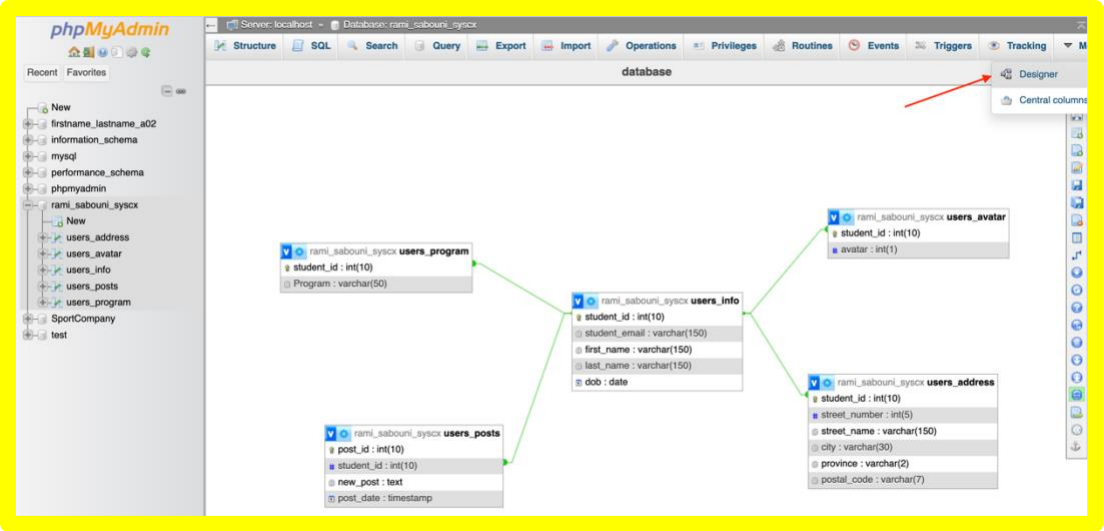


2. Name the file "`firstname_lastname_a02.sql`". Replace `firstname` and `lastname` with your information.
3. Open the created .sql file and add the following statements at the top.

```
CREATE DATABASE IF NOT EXISTS firstname_lastname_syscx;

USE firstname_lastname_syscx;
```

4. Add the .sql file to your main assignment project folder. Should be in the same level as your `index.php`, `profile.php` and `register.php`.
5. Once you finish creating the database and the five required tables, click on the "`Designer`" tab and take a screenshot of similar to the one provided in the instructions above. Name the screenshot "`firstname_lastname_database`". The accepted extensions are `.png` or `.jpg`.
   - **Note:** Depending on the width of your screen, you might find `Designer` tab under `More` drop down menu



6. Add the screenshot to the assignment's folder.
7. Compress your project's folder that you created at the beginning of the assignment.
   - **IMPORTANT:** The only allowed extension is: **.zip**
8. Submit your work on Brightspace.
   - No email submissions are allowed
   - Missed deadline will result in an automatic **zero**