📄 Assignments

[🔗 Link] [❓ Help] [⤢]

Assignment 3 - Fall 2023 - **Not Started**

Assignment Details

Title	Assignment 3 - Fall 2023
Student	Yulun Feng
Grade Scale	Points (max 100.00)

Instructions

Assignment 3

Purpose

- Upon completion of assignment 3 you will be able to:
  - Write a program using PHP and MySQL that connects to a backend database
  - Allow a user to make updates, insertions, deletions and query a database
  - Create a program that runs on the internet
  - Create a Graphical User Interface using HTML and CSS

Instructions:

Setting up the directories correctly for storing your code and being able to push to BitBucket:

If you did the entire workshop for Setting Up Your VM, you will likely have all the permissions set correctly and the directory set up and not have to this part but if you did not complete the workshop, make sure you complete the portion of the workshop called Using Git and Western's BitBucket Repository to Regularly Backup Your Work (around page 12 of here). By doing this workshop you will create a folder called *a3yourfavouritenanimal* where you can put all your code AND you can push and pull your code to our BitBucket system .

Put all your code in *a3yourfavanimal* folder and subfolders of *a3yourfavanimal*. Your URL will then be:
http://cs3319.gaul.csd.uwo.ca/vm???/a3yourfavouriteanimal

What your web application should be able to do:

Using the database created for assignment 2,CSS, HTML, PHP and MySQL (and JavaScript if you want, but you don't have to use JavaScript. It is possible to do the whole assignment without using JavaScript), create a website that allows someone to update and view the Teaching Assistant database information. Here are the tasks a user must be able to complete:

- List all the information except image about the teaching assistants. Allow the user to order the data by Last Name OR by degree type(you might want to use a radio button for this choice). For last name, allow the user to either order them in ascending or descending order (you could also use a radio button for this choice).
  - Allow the user to select a t.a and if they select a t.a., then display all the data about that t.a. and display the picture if the image field has a valid URL in it. If the image field is null just display the generic outline of a head.
  - Also show all the courses that the selected t.a. loves and the courses that the selected t.a. hates. If the t.a. doesnt have any courses that they love or hate, do not show anything (or put a message like "This t.a. has not picked courses that they love")
- Allow the user to select one of either Masters or PhD and then list all the ta information about tas studying for this degree.
- Insert a new teaching assistant. Prompt for the necessary data. The user may also enter the courses that this t.a. loves or hates. they must pick from existing courses, so you could give a list of courses to pick from (using a dropdown box or radio buttons). Note: if the user types in an existing Western User Id OR an existing student number, your program should output an error message and not let the new ta be added.
- Delete an existing ta. Either prompt for the ta Western User ID or you could display the list of tas and allow the user to pick the one they want to delete. If you decide to prompt for the Western User ID, make sure you remember to give an error message if the user tries to delete a non-existent ta. If the ta is assigned to a course offering, output a message that you cannot delete this ta. Also remember that any permanent deletions should always allow the user the chance to back out (e.g. "Are you sure you want to delete this person?").
- Modify a ta - Allow the user to change the image URL.
- Assign a ta to a course offering. Do not allow this if the relationship already exists (output a warning like "Ta is already assigned to this course offering"). Allow the user to select a ta and a course offering and then create the relationship. Prompt for the relationship.
- Allow the user to select a course and see the the all the course offering for the course. Show the course offering id number, the number of students that term, the term and the year it was offered for each course. Also allow the user to pick a start year and end year and just show the course offerings between (and including) those years.
- Allow the user to select a ta. and see all the course offerings that this t.a. has worked on. Make sure you show the course number and course name for the course offering as well as the term and the term and the year and the hours and if this t.a. loved this course or hated this course (maybe put a happy face or sad face next to it).
- Allow the user to select a course offering and display the course number and name and the first and last names and user ids of all the t.a.s who have worked on this course.
- WHEN YOU SUBMIT YOUR ASSIGNMENT, MAKE SURE YOU LEAVE YOUR PROGRAM FULL OF DATA so that the Peer Markers CAN TEST YOUR PROGRAM WELL and not run out of test data. You can use your scripts from assignment 2 to fill the database. Use THIS FILE: https://www.csd.uwo.ca/~lreid2/cs3319/assignments/assignment3/scriptscriptfall2023toloaddata.txt to fill up your database with some sample data.
- It is a bit easier to navigate the program if you have a main menu on your first screen with the available options, so you MUST have one page to start on with buttons/choices that take you to new pages to complete each of the tasks listed above. This starting page MUST be called **mainmenu.php**

It is a good habit to disconnect from a database once you have finished using it, make sure you program disconnects from the database.

Remember that PHP can get large and cluttered, your application will be marked partly on your structure, comments and modularity, don't put everything in one file, try using PHP *includes* and functions and separate files to break up the PHP code. Also put comments in your code at the top of each file to explain the purpose of each file and any complicated sections of code but don't include your name, just put something like :
Programmer Name: ?? (where ?? is the last 2 digits of your western student number)

NOTE: you **cannot** use any third party DAL frameworks that let you avoid writing SQL queries/statements. We want you get experience writing SQL with this assignment.
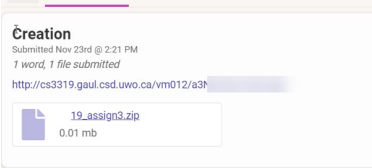
==REMEMBER: You are expected to write the interface YOURSELF, you can look at other sites for ideas/inspirations but do NOT copy and paste any code from any of your classmates (their HTML code or CSS code or JavaScript code or PHP code) or you could receive an academic penalty. Cite any snippets of code (more than 20 lines) you get off the internet in your comments. You may use the code we did in Flipped Classroom 4 and from the PHPWorkshop without citing it, I am fine with you using my code from those exercises and modifying it.==

==DO NOT COPY CODE AND DO NOT GIVE YOUR CODE TO YOUR FRIENDS --> it is not worth it! It is better NOT to hand in the assignment and get 0 for the assignment or hand something that works a little bit and get some marks THAN it is to copy from someone and get reported to the Dean and possible expulsion from Western!==

Handing in your assignment:

You are required to submit the following via Kritik.io:

- In the kritik.io submission textbox put the URL for your working application so the peer markers can try out your program, for example:
  http://cs3319.gaul.csd.uwo.ca/vm???/a3owl/ where ??? is your virtual machine number and a3owl is the name of YOUR folder.
- In the kritik.io attach/upload: all files (.php and .html and .js and .css and extra files) used to create your application. Please zip them together and just upload the .zip file. The .zip file MUST be named *last2digitsofyourstudentnumber_assignment3.zip*, eg **78_assignment3.zip**
- DO **NOT** PUT YOUR FILES IN A .RAR file, MACS cannot read .rar files, it MUST be a .zip file!
- Should look similar to this:

Creation
Submitted Nov 23rd @ 2:21 PM
1 word, 1 file submitted
http://cs3319.gaul.csd.uwo.ca/vm012/a3▓
📄 19_assign3.zip
0.01 mb

Late Policy:

- 5% off for each day late. You can be up to THREE days late.

GRADING IN KRITIK

Once the last day to hand in has passed you will be sent an email from kritik.io with 5 or 6 students whose code you must mark. You will get 3 days to do your peer marking. Then you will have 1 day to evaluate the feedback you were given. You will be able to see the rubric in kritik before you hand in the assignment so you should check it out to make sure your program will get a great grade.

This assignment does not accept online submissions. Contact your instructor for additional instructions.

[Back to list]