

Assignment 3

Purpose

- To give you exposure to
 - creating a program that
 - connects to a database
 - writes, reads and updates the data in the database
 - writing the interface between a database and a user
 - creating a client/server program that runs on the internet, serves up a page that runs on the client side (the users browser) and also connects to a back end database on the server side (your virtual machine)

Instructions

MAKE SURE YOU PUT ALL YOUR CODE IN A SUBFOLDER OF /var/www/html called *assignment3* AND MAKE SURE YOU SET THE PERMISSIONS AS FOLLOWS ON YOUR cs3319.gaul.csd.uwo.ca SITE:

```
cd ~  
cd /var/www/html  
mkdir assignment3  
chmod 755 assignment3
```

Put all your code in *assignment3* folder and subfolders of *assignment3*. Your URL will then be:
http://cs3319.gaul.csd.uwo.ca/vm??/assignment3

Make sure all your files in the *assignment3* directory have a permission of 644 (to do this, inside your *assignment3* folder do this command):

```
chmod 644 filename.php  
chmod 755 anysubdirectories that you make in assignment3 folder
```

Using the database you created for assignment 2, JavaScript, CSS, HTML, PHP and MySQL,, create a website that allows someone to update the customers, products and purchase tables.

- List all the information about all the customers in alphabetical order by last name. When a user selects a customer, display all of his/her products that he/she has purchased.
- List all the products in alphabetical order by description OR in order by price. Allow the user to decide if the order is ascending or descending for both the description and price.
- Insert a new purchase (prompt for necessary data) **Note: Send an error message if they try to give an invalid customer id number or invalid product number (or make your GUI so that it doesnt allow them to pick those).** If the user tries to let a customer purchase a product they already have purchased, instead just let them change the quantity that the customer will have purchased of that product. Don't allow the quantity to go lower, just higher by the amount they want now.
- Insert a new customer (prompt for necessary data) **Note: Send an error message if they try to insert an existing customer id number (or make your GUI so that it generated the new key for the user),**
- Update a customer's phone number, prompt for the customer id number, display the current phone number before prompting them for the new phone number. **Note: Send an error**

message if the user doesn't enter an existing customer id (or use your GUI to only allow them to pick from a list of current customers)

- Delete a customer (Prompt for the customer id to delete) Note: Send a warning message if they try to delete a non existing customer id number (or make your GUI so that they can only pick an existing customer from a list)
- List all the customer names who bought more than a given quantity of any product. Prompt the user for the quantity. Display the description of the product and quantity purchased also.
- List the description of any product that has never been purchased
- List the total number of purchases for a particular product and the product description and the total money made in sales for that product (cost * quantity). Prompt the user for the product id (Note: display an error message if the the product does not exist - or create the GUI in a way that the user cant pick a product that doesnt exist)
- Bonus (worth 2%): add an extra field to the customer's table called cusimage (you can do this right in mysql, not using php code, make it char(100)). Allow the user to click on one of the customers and if this field is null then let the user find an image online and add the url to the officials table AND display the image in your user interface. If the field is not null, display the image at the url..

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.

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 and you WILL receive NEGATIVE the worth of this assignment. Site any snippets of code (more than 20 lines) you get off the internet in your comment.

DO NOT COPY CODE, it is not worth it, it is better NOT to hand in the assignment and get 0 for the assignment THAN it is to copy from someone and get -15% towards your final grade, reported to the Dean and possible expulsion from Western!

Handing in your assignment:

You are required to submit the following via Owl:

- In the Owl submission text box put: the URL for your working application so the cs3319 teaching assistant can try it out.
- In the Owl 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.

Late Policy:

- 5% off for each day late, Saturday and Sunday count as one day. You can be up to THREE days late.