COMP-10260 – Assignment 1:

# Introduction:

Your task in this assignment is to choose a business and create a server-side script in PHP which gives users the experience of browsing through an online store.

Your online store will have a catalog of **at least fifteen products** that you will create. The products should be somewhat realistic. i.e., If your store is **Fast Fish Online Aquatic Superstore** then you should have items like “Aquarium, 3000L”, “Goldfish Food”, etc. Do not hand in something with placeholder names like “Product 1”, “Product 2”.

Each product will have six fields:

1. Product id: This is a unique identifier for the product. It can be a number or an alphanumeric code.
2. Product name: A relatively short readable name for the product.
3. Product description: A short paragraph describing a product.
4. Product Price: The cost of this product to the buyer.
5. Product Quantity: The number of items you have in your store.
6. Product Category: This should be a string representing the kind of item this is.

**You may use the text from other websites or sources for your product names and descriptions, but state your sources in a comment within the code.**

### Categories:

You will divide your products up into three categories. Each product must belong to a category. These can be anything reasonably realistic. i.e., FISH FOOD, AQUARIUMS, ACCESSORIES, …

If you have some items which you can’t come up with a single category to describe them. You can create a category called MISCELANEOUS and put just about anything in it.

### No databases

For Assignment 1, your store will store its catalog of products and any other data in arrays. To make these arrays persistent they will be written to disk using the **file\_storage.php** system available on Canvas. **You are not allowed to use a database of any kind.**

To use the **file\_storage.php** system. Your php script must be in the same directory as the file\_storage.php file and your php script must contain the line:

**include "file\_storage.php** **";**

After that you can write your data to a file with the following line:

**writeDataFile('somefilename.json',$data);**

Where **$data** is the array containing your product catalog.

To read the data back. Simply use:

**$data = readDataFile('somefilename.json');**

# Functional points:

Your online store **MUST** do the following:

### General

Your product catalog must contain **at least** fifteen items.

The items must be somewhat realistic (they may be copied from an existing online store -- but not from another student).

Each item must have a product\_id, product\_name, product\_description, product\_price, product\_quantity and product category. **At least one of your products must have a product\_quantity of zero.**

The system must create a shopping cart for each new user. A user is considered new if they do not transmit a PHPSESSID cookie.

When the user points their web browser to your script, they should see a user-interface similar (but NOT IDENTICAL) to the one below with the following features:

Graphical user interface

Description automatically generated

1. Your store name is always displayed.
2. Three product categories (plus the “everything” category) are always displayed.
3. The product catalog is displayed.
4. The user’s shopping cart is always displayed.

The use of BootStrap is welcome, but not required.

### Categories

As mentioned above you must sort your products into three categories. The user must be able to see all three categories on the main screen, each one should be selectable with the following behaviour:

* Selecting any one of these categories allows the user to view only items in that category.
* If there are more than seven items in a particular category, then controls should appear to allow the user to page through them.
* There should also be fourth category “everything” which allows the user to see all the products.

### Product catalog

The product catalog is shows information about the items in your store. No more than seven items can be displayed on the screen at a time. In order to see the other items in the catalog the user must be presented with controls when enable them to page forward and backward through all the items. The items should exhibit the following behaviour:

* The user should only be presented with the option to move forward or backward through the catalog if more records exist in the category being viewed.
* For example: If you are looking at AQUARIUMS and there are only five AQUARIUMS in your datafile. Then there should be no visible controls to move forward or backward.
* If you are viewing EVERYTHING and are looking at the first page of items. There should only be the option to move forward.
* If you are viewing EVERYTHING and are at the last page of items. There should only be an option to move backward.
* If the product has a quantity of zero. The number of items in stock should be replaced with the phrase “Out of Stock” written in red.

The catalog should display only display the **product id**, **product name, product price, and product quantity**. If an item has a quantity of “0” then the catalog should read “Out of Stock” instead of the number 0. Three items in your catalog should be out of stock (quantity 0).

When an item in the catalog is clicked on the system should display the **product\_id, product\_name, product\_description and product\_price** as well as provide the user with the ability to add the product to their shopping cart. It should look something like this:

Graphical user interface, text

Description automatically generated

As you can see, the complete information for the item clicked on is displayed. As well as a button marked "add to cart". Clicking on the “add to cart” button adds moves the item to your shopping cart. The button should only be visible if the item’s product\_quantity > 0. Otherwise, the area where the button would be should read “Out of Stock” (in red).

### Shopping cart

When an item is in your shopping cart the following rules should apply:

* When an item is placed in your shopping cart the product quantity for that item goes down by one.
* The product id, product name and product price for each item should appear in the cart.
* Each item in the cart should have a control beside it for removing the item from the cart.
* When the “remove item” control is clicked, the user should be asked “Are you sure?” before the item is removed.
* When the item is actually removed from your shopping cart. The product quantity should be increased by one.

A picture containing company name

Description automatically generated

# What To Hand In

You are required to submit your files **both** to Canvas and to CS Unix. The assignment is designed that you can work on it in XAMPP and then upload it when complete, but the work flow is up to you. I will grade your assignments once both submissions are complete.

For the purposes of due dates, the submission to Canvas is your proof of submission date/time.

I suggest that you upload your code to CS Unix first, verify that it is working, and then submit to Canvas. Both sets of file must be identical except for the file renaming specified below.

## CSUNIX SUBMISSION

**Please remember that all file names on CSUNIX are case sensitive, and will not react well to having spaces in the file name. For more information please see the CS Unix References module on Canvas.**

* In your public\_html/private directory on CSUNIX create a folder called 10260
* Inside your public\_html/private/10260 folder create a folder called a1
* Upload your solution files into this folder.

You can check your submission is working by visiting it in your web browser. If your student number is 123456789, you will go to <http://csunix.mohawkcollege.ca/~sa123456789/private/10260/a1>

Please email me with any password reset requests ***as early as possible***. I will try very hard to perform this within 1 business day, but if the problem is a configuration problem on the server we may need to involve IT.

## *Canvas Submission*

* Upload each file that you have created with the extension “.txt” added to the end of the file name. for example: myprogram.php should be renamed to myprogram.php.txt – this will enable me to preview your code using the built in canvas viewer
* Be sure to include all php, html and css files that you have created
* a copy of your catalog file
* you do not need to include any starter code that you have not modified (for example, file\_storage.php)

# Rules

1. What you hand in must be entirely your own original work. You are not allowed to use code from other sources including frameworks and libraries. Violating this is an automatic zero.
2. you must include a statement of authorship containing at least your name and your student number in a comment at the top of every file you submit. files without this identificaiton may be marked as a zero.