COMP-10260 – Assignment 2:

# Introduction:

Your task in this assignment is to enhance the script you created in Assignment 1 by utilizing two pieces of technology. Your application will now use a mySQL database for storing its product catalog. As well the shopping cart portion will operate as a JSON application:

Your application will perform all the required functions (See Assignment 1 if you’ve forgotten what the application is supposed to do) but adding and removing items from the shopping card will use the **fetch()** API to call a different **.php** file.



### fetch()

Your application will make calls to the **fetch()** method. The calls should set a HTTP header of **Content-Type: application/x-www-form-urlencoded** and should use the POST request method to send the data. The actual data you send must be formatted like this **field1=value1&field2=value2&field3=value3**. This will allow your server-side script to use **$\_POST** to pick up the data being passed to it. The data being returned by your server-side script should be in HTML form which your JavaScript can insert into your web page using the **innerHTML** property.

### session\_id()

Because you will be storing all the shopping cart data in the database. You will need to relate this to a particular user by also storing their session id. The session id is transmitted in the PHPSESSID cookie. The easiest way to get this information is to use the function session\_id(). For example:

**$user\_session\_id = session\_id();**

### databases

For this assignment you must create and use a MySQL database. The database needs to track both the session information (what’s in the user’s shopping cart) as well as the catalogue of products. You may not use flat/text files for storing any of this data.

Your database must have at least two tables – one for the catalogue data and one for the session data (the shopping cart contents). You can design it with more if you wish.

You may want to import the catalogue data from assignment 1 into your database. Canvas has a template for writing a script to do that. It’s called: **migrate\_data.php** – You will have to add your database password, location of your assignment 1 data and possible alter the table structure. This is just for your convenience. You don’t need to hand the import script in.

# Functional points:

In addition to all the functionality from Assignment 1 (except for using flat files to store your data). Your online store **MUST** do the following.

### Sessions

The session key is still generated by the PHP function **session\_start()**. This will create a file in **/var/lib/php/session** however no data is allowed to be stored there. Everything will be stored in a database table. If you need to get the session key, you can find it in **$\_COOKIE['PHPSESSID']**.

### Catalog:

You are required to store the catalog data in a database table and use this table to display the catalog data on the screen as well as to keep track of the quantity of items in your store.

### CATEGORIES

The list of categories needs to be dynamically generated from the database. So, if a new item was added to the catalog with a fourth category. It would appear on the left-hand side of the user-interface.

# Getting credit for Fixing Previous Functionality

Since this assignment builds on your previous work you have the opportunity to add or fix functionality which was missing from the code you handed in in Assignment 1. You can get credit for this by submitting a text file with your fixed elements along with your assignment. The file should be named CHANGELOG.txt and contain statements which document the problem with your Assignment 1 that you have fixed:

“Assignment 1 did not break the catalog up into pages. Fixed in this version.”

# What To Hand In

* UPLOAD YOUR SOLUTION TO CSUNIX IN A FOLDER /public\_html/private/10260/a2
  + *Do not do this at the last minute – test your solution as your functionality will be evaluated on this version. No resubmissions will be allowed to fix broken code after the due date.*
  + *Remember that many things that Windows allows you to be sloppy about (e.g. case sensitivity) will be treated strictly by Linux.*
  + *Password reset requests should be submitted ASAP and not at the last minute.*
* Append the extension “.TXT” to your PHP fileS containing your entire application, and upload these files to canvas. do not submit your html or css files as they are not being evaluated. do not rename zip files to .txt just to circumvent the upload restriction, you will be graded with a score of 0 for such submissions.
  + *If your file is called page1.php please submit page1.php.txt to Canvas.*
* an export of your database as a .sql file, with a .txt extension appended.
  + *Mydb.sql should become Mydb.sql.txt, for example.*
* A CHANGELOG.txt file if you are fixing functionality from the previous assignment. Any changes or improvements must be noted to be considered for re-evaluation.

# 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 and an academic integrity citation will be issued.