The State Universtiy of New York, Korea
Computer Science

CSE 305/532 **Handout 6: Problem Set 5** **April 30, 2018**

This problem set is due at **11:55pm on Thursday, May 10, 2018**. Don't go by the due date that you see on Blackboard because it is in EST. Go by the one given in this handout.

Be sure to include a comment at the top of each file submitted that gives **your name** and **email address**.

Submit all the files that make up your solutions as a **single zip or tar file** on **Blackboard**. Multiple submissions are allowed before the due date.

If you want to **pair program** with another student in our class, you are welcome to do so. In that case include two names in one copy that you submit. Use one of your names as you name files below.

Problem

In this problem you will build a simple yet complete web application using MySQL and PHP. Yes, you may use HTML, CSS, JavaScript, etc. in addition if you like. You may use something other than PHP if you like. In the description below replace PHP by X, where X is the language that you chose if it is not PHP. If you choose to use something other than PHP, please let me know in advance.

First, write a SQL script named (`ps5.sql`) that creates a database named `jedps5` if your name is John E. Doe, i.e., using your initials. We want to be able to run your script and PHP program(s) as we grade your work by first running your script and then your PHP program(s) which will be part of your web app. Your script will crate a database of the name given above, create tables and populate them. Populate each table with enough data so the queries that you execute will generate meaningful results.

The tables that you create and populate should implement the following database schema:

```
Products(maker, model, type)
PCs(model, speed, ram, hd, price)
Laptops(model, speed, ram, hd, screen, price)
Printers(model, color, type, price)
```

The `Products` relation gives the manufacturer, model number and type (PC, laptop, or printer) of various products. We assume for convenience that model numbers are unique over all manufacturers and product types, although that assumption is not realistic – a real database would include a code for the manufacturer as part of the model number. The PCs relation gives for each model number that is a PC the speed (of the processor, in gigahertz), the amount of ram, the amount of hd, and the price. The `Laptops` relation is similar, except that the screen size (in inches) is also included. The `Printers` relation records for each printer model whether the printer produces color output (true, if so), the printer type (laser or ink-jet, typically), and the price.

Your web application should at a minimum include the following capabilities:

(a) Your application should have a page that asks the user for a price and find the PC whose price is closest to the desired price. Print the maker, model number, and speed of the PC.

As for the user interface, the result page should have a link back to the page asking for a price so that the user can run this query as many times as s/he wants to. Or, if you like, you can group all of these (including the ones below) together onto a single page and add a link that just clears the output.

(b) Your application should have a page that asks the user for minimum values of the speed, RAM, hard-disk size, and screen size that s/he will accept. Find all the laptops that satisfy these requirements. Display their specifications (all attributes of `Laptops`) and their manufacturer. Again, a similar way of repeating as in part (a).

(c) Your application should have a page that asks the user for a manufacturer. Display the specifications of all products by that manufacturer. That is, display the model number, product-type, and all the attributes of whichever relation is appropriate for that type. Again, a similar way of repeating as in part (a).

(d) Your application should have a page that asks the user for a ”budget” (total price of a PC and printer), and a minimum speed of the PC. Find the cheapest ”system” (PC plus printer) that is within the budget and minimum speed, but make the printer a color printer if possible. Display the model numbers for the chosen system. Again, a similar way of repeating as in part (a).

(e) Your application should have a page that asks the user for a manufacturer, model number, speed, RAM, hard-disk size, and price of a new PC. Check that there is no PC with that model number. Display a warning if so, and also insert the information into tables `Products` and `PCs`. You don’t need to repeat this case unless you want to.

You should have a **readme.txt** file that describes how to run your application. Include what version of Windows or Mac OS X you are using; whatever else we would need to know to run your app on our machine. Include the url’s that we would have to use to access your pages too. If your instructions are not detailed enough and our grader can’t run it, it will be your responsibility. Just in case the grader can’t get it to run on her/his machine, include enough screen shots as you run yours on your machine so that we can see that at least it ran on your machine. We still want to be able to run yours on our machine though. If you deployed it to your machine that we can remotely access, describe how to access it there.

Submit all the files that make up your solution including screen shots as a single zip or tar file. Name your zip or tar file using your initials, e.g., **jedps5.zip** or **jedps5.tar** if your name is John E. Doe.