Week 10 Practical Exercises

Note:

- Exercises 4, 5, 6 will be assessed as part of the Practical Set 2 submission.
- Include HTML comments for your student ID, Name, and Practical Class Time at the top of each source file created.
- All files must be uploaded to your TWA web site before submission of Practical Set 2.

Objectives:

- write php scripts that correctly connect to a MySQL database
- write php scripts to view content of tables within the MySQL database
- write php scripts that incorporate form user input to query the database and view the returned data
- implement simple server-side form input validation
- write php scripts that correctly retrieve form inputs using the basic ideas of postback and query the database and incorporate simple server-side form input validation.

Suggested Resources:

- PHP Manual `https://www.php.net/manual/en/index.php`
- PHP tutorials `https://www.w3schools.com/php/`

Before you start:

1. Upload the html and php files provided to you (see the zip file) into your TWA web site in the `practicals/week10` folder
2. Upload the css file provided to you (see the zip file) into your TWA web site in the `practicals/css` folder

PHP Database Access

The following task is designed to help you work through how to connect to a MySQL database from your PHP scripts. You will use a MySQL database named `electrical` that is installed on the TWA server. The `electrical` database consists of **four** tables: `customer`, `purchase`, `product` and `staff`. Below gives the definition of these four tables, the underlined fields indicate the primary key of each table.

Table Name: `customer`

Field Name	Data type	Description
customerID	CHAR(6)	unique identifier for a customer
firstName	CHAR(30)	customer first name
lastName	CHAR(30)	customer last name
address	CHAR(150)	customer street address
suburb	CHAR(25)	customer suburb
state	CHAR(3)	customer state
postcode	CHAR(4)	customer postcode

Table Name: `purchase`

Field Name	Data type	Description
id	INT	unique identifier for an order placed by a customer
orderID	CHAR(7)	the order number
productCode	VARCHAR(10)	a product within an order
quantity	INT	how many of the product ordered
orderDate	DATETIME	when the order was placed by the customer
shippingDate	DATETIME	when the order was shipped to the customer
shipped	CHAR(1)	indicates if the order has been shipped
customerID	CHAR(6)	who the order is for
staffID	CHAR(7)	which staff member processed the order

Table Name: `product`

Field Name	Data type	Description
productCode	VARCHAR(10)	unique identifier for a product
name	VARCHAR(60)	product name
quantityInStock	INT	how many of the product are in stock
price	FLOAT	how much the product costs to the customer

Table Name: `staff`

Field Name	Data type	Description
staffID	CHAR(7)	Unique identifier for a staff member
staffName	VARCHAR(50)	Staff members name

To connect to the `electrical` database use the following **in your php script**

```
$dbConn = new mysqli("localhost", "TWA_student", "TWA_2024_Autumn", "electrical");
if($dbConn->connect_error) {
  die("Failed to connect to database " . $dbConn->connect_error);
}
```

Exercise 1:

A. Create a PHP file named `conn.php` in the `practicals/week10` folder of your TWA web site. Copy the **electrical** database connection details as given above into `conn.php` and save the file. Don't forget the php script tags!

B. Create a PHP file named `Exercise1.php` in the `practicals/week10` folder of your TWA web site. The purpose of this file will be to display all records of the `product` table from the `electrical` database. **Note** that all fields of the `product` table should be displayed. The following example code will be helpful (you will need to refer to the database table definitions above to determine the names of the fields that are not included in the example code below:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Week 8 Exercise 1</title>
    <link rel="stylesheet" href=" ../css/week10Styles.css">
  </head>
  <body>
  <?php
    require_once("conn.php");
    $sql = "SELECT * FROM product";
    $results = $dbConn->query($sql)
      or die ('Problem with query: ' . $dbConn->error);
  ?>

  <h1>Product table</h1>
  <table>
    <tr>
      <th>Product Code</th>
      <th>Name </th>
      <th>Quantity In Stock</th>
      <th>Price</th>
    </tr>
  <?php
    while ($row = $results->fetch_assoc()) { ?>
    <tr>
      <td><?php echo $row["productCode"]?></td>
      <td><?php echo $row["name"]?></td>
      <!-- output the other fields here from the $row array -->
    </tr>
  <?php }
    $dbConn->close(); ?>
  </table>
</body>
</html>
```

Exercise 2

The `ViewAllTables.php` script that has been provided to you gives you a view of the data that exists in the four tables of the `electrical` database from exercise 1 without needing to recode exercise 1 for each table. **Note**: you will need to have completed exercise 1A before the `ViewAllTables.php` file will work since line 5 of the `ViewAllTables.php` script includes `conn.php.`

- Run the `ViewAllTables.php` file to view the data for each table from the `electrical` database. You will need to refer to the output of this script to assist you with some of the following exercises.

Exercise 3

Write an SQL statement that extracts the `name`, `quantityInStock`, and `price` from the `product` table for the products that have **more than 10** in stock and sorts the results in ascending order of `quantityInStock`. Incorporate this SQL in a PHP page to **display the results in an HTML table**. Above the table display the heading: **Products with stock > 10**. Save the file as `Exercise3.php`. Upload it to your web site and test it. *Verify your output by comparing with the data in the product table as obtained from Exercise 2 above*.

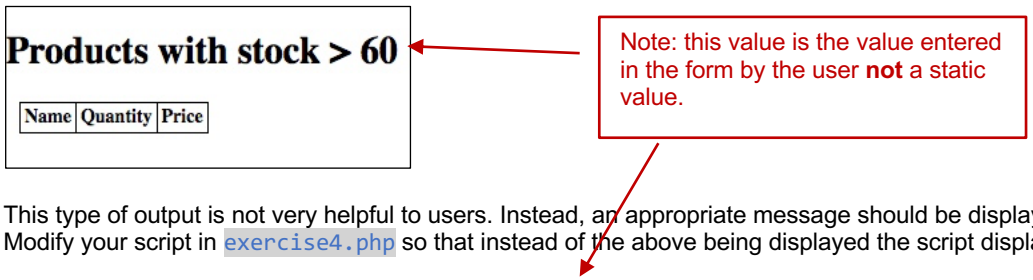
Note: you can use the code from Exercise 1 as a guide for producing the required php script for this exercise.

Exercise 4

- This exercise uses the file `exercise4.html` (found in the zip file for this practical) in the `practicals/week10` folder of your TWA website
- Create a PHP file named `Exercise4.php` in the `practicals/week10` folder of your TWA web site. This file will be the PHP script that processes the data submitted from `exercise4.html` (ie, this is the action URL of the form).

A. As a starting point, copy the code from `Exercise3.php` into `Exercise4.php`. Modify the code in `Exercise4.php` so that the sql query incorporates the value from the form into the **where clause** instead of the quantity value always being 10 as in exercise 3. Test the script by loading the form in the browser, entering a value in the form and submitting. *Verify your output by comparing with the data in the product table as obtained from Exercise 2.*

B. If you enter a value in the form that is 60 or larger (the ***current largest*** value in the quantity field) the PHP script will produce output that looks like:



This type of output is not very helpful to users. Instead, an appropriate message should be displayed.
Modify your script in `exercise4.php` so that instead of the above being displayed the script displays the message:

There are no products that have more than 60 in stock.

C. If you enter a value in the form that is not a number the PHP script will generate incorrect results. Test this by entering a word in the text box and submitting the form to observe the output that is generated. Instead of allowing the code to run the query in this situation we should validate that the value entered into the form is a number **before running the query**.

Modify your script in `exercise4.php` to validate that the value entered into the form is a number.

Note: the validation **MUST** be written in PHP, not JavaScript. That is, the validation of the user input occurs on the server not the client. The output generated by the script, when the value entered is not a number, should be the message:

The value entered for the quantity was not a number.

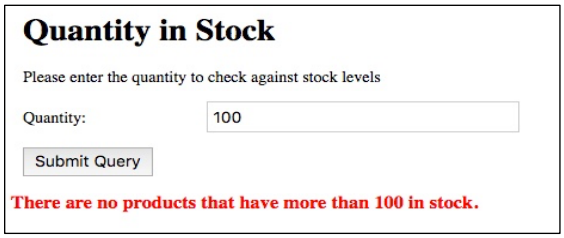
Exercise 5

- Create a PHP file named `Exercise5.php` in the `practicals/week10` folder of your TWA web site.
- Combine the code from `exercise4.html` and `exercise4.php` into `Exercise5.php` (this is similar to exercise 4 in week 9 practical exercises) so that the php script and the html form are in the same file.

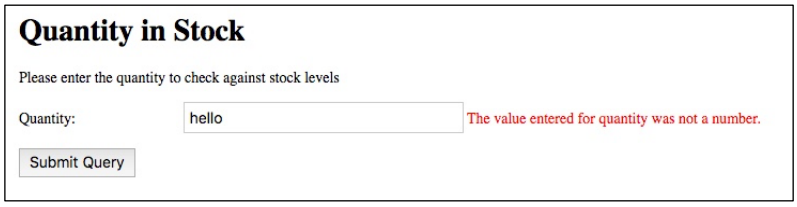
A. Modify the code so that
   i. The form uses postback (ie, the form action is the same file `exercise5.php`)
   ii. Only the form is displayed on first load of the page
   iii. The table of products is only displayed when there are records to display
   iv. The form **and** the messages as described in exercise 4 are displayed **when appropriate** in **appropriate** locations
   v. The value that the user enters in the form is maintained in the text box after form submission

The following screen dumps provide guidance for expected output for the above scenarios:

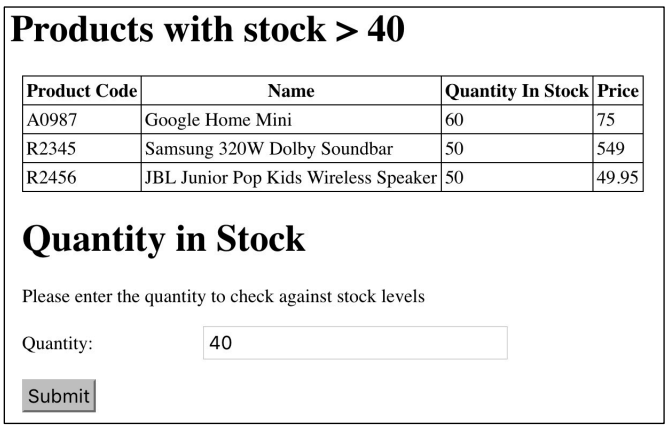
A(ii) only the form displayed on first page load



A(iii) table of products is only displayed when there are records to display
A(iv) The form **and** the messages as described in exercise 4 are displayed **when appropriate** in **appropriate** locations
A(v) The value that the user enters in the form is maintained in the text box after form submission



A(iii) table of products is only displayed when there are records to display
A(iv) The form **and** the messages as described in exercise 4 are displayed **when appropriate** in **appropriate** locations
A(v) The value that the user enters in the form is maintained in the text box after form submission



A(iii) table of products is only displayed when there are records to display
A(iv) The form **and** the messages as described in exercise 4 are displayed **when appropriate** in **appropriate** locations
A(v) The value that the user enters in the form is maintained in the text box after form submission

Exercise 6

- This exercise incorporates session variables to store information about the user that can be retrieved on subsequent php pages whenever needed. `Exercise6.html` in the `practicals/week10` folder of your TWA web site contains a form which will capture your name and favourite hobby. The form submits to `exercsie6.php`

  Note: Do NOT change the content of this file in any way.

A. Create a php page named `exercise6.php` in the `practicals/week10` folder. This is the action for the form in `exercise6.html`. The `exercise6.php` file should receive the data from the `exercise6.html` form and store the values into appropriate **session variables**. The only output on page `exercise6.php` will be a hypertext link which links to `exercise6a.php` (the second of two php files that you will create).

B. Create a second php page, `exercise6a.php`, in the `practicals/week10` folder. This file is to retrieve the session variables that were created by the `exercise6.php` page. After retrieving the session variables from the server, the page should display the values in an appropriate way.

`Exercise6.html` → `Exercise6.php` → `Exercise6a.php`

User fills in form & submits

User clicks link

The user details are displayed