Week 7 Practical Exercises

Note:

- Exercise 2 and 3 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 retrieve form inputs via post and get methods.
- write php scripts that correctly retrieve form inputs using the basic ideas of postback.
- implement a simple map using the Google Maps API and utilise features such as markers, labels and Info Windows in the map.

Suggested Resources:

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

Exercise 1:

- In the **practicals/prac2** folder of your TWA web site create a new subfolder named **week7**
- Upload the html and css files provided to you (see the zip file) into your TWA web site in the **practicals/prac2/week7** folder
- Create a file named **exercise1.php** in **practicals/prac2/week7** on your TWA web site.

A. Add the following code to **exercise1.php**:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Week 7 Exercise 1</title>
  </head>
  <body>
    <?php
       //obtain the firstname input from the $_GET array
       $namestr = $_GET["firstname"];
       //obtain the values for the other input devices here

    ?>
    <p>The following information was received from the form:</p>
    <p><strong>name = </strong> <?php echo "$namestr"; ?></p>
    <!--output the other form inputs here -->
  </body>
</html>
```

B. Test the php script by submitting data from exercise1.html to the php script. You will notice that not all of the input devices from the form are shown in the output of the php page. Fix this by making appropriate modifications to the php file so that all input devices are shown as part of the output.

> *What you should notice by completing this exercise is that it does not matter what type of input device is used on the form (text box, radio button, checkbox, selection list, etc) the method of obtaining the values is the same.*

Exercise 2

- This exercise utilises **exercise2.html** from the zip file**.** Upload this file to **practicals/prac2/week7** folder in your website.
- The forms in exercise 1 and 2 are very similar but with some important differences. These include:
  - the selection list size has been changed to "4"
  - multiple selections are now enabled in the selection list
  - the action of the form has been changed to exercise2.php
  - the form method has been changed
- create a file named **exercise2.php** in **practicals/prac2/week7** on your TWA web site
- **copy** your solution code from exercise1.php into exercise2.php

A. **modify** exercise2.php to correctly process the form from exercise2.html. **Hint**: Because of the changes made to the form (in particular, the change to the form method is critical) you will need to modify several aspects of the php script.
B. Ensure that your exercise2 script is capable of displaying all of the selections made from the selection list - it won't without appropriate modifications to both the html form and the php. **Hint**: Review lecture slides and examples for how to process input devices that can have multiple values.

Exercise 3

- This exercise utilises **exercise3.php** from the zip file. Upload this file to **practicals/prac2/week7** folder in your website.
- The differences between the forms in exercise2 and exercise3 are:
  - the form submits to itself (ie, the action is exercise3.php),
  - the php processing for retrieving and displaying the form values is in the same php file as the form instead of a separate php file. This is known as postback. Review lecture notes and examples where we covered the idea of postback.

Modify the code given in exercise3.php so that

A. **all** form values as submitted by the user are correctly displayed in the output section underneath the form. **Note**: The first form field has been done for you in the code.
B. the output section underneath the form only displays if the form has been submitted. **Hint**: review lecture notes and examples where we covered the idea of postback.
C. the *Undefined Index* notices do not display above the form (**do not move** this section of the code). **Hint**: your solution to part B will help you with part C.

Exercise 4

This is a simple exercise to explore the Google Maps API. It will teach you how to implement Google maps within your web pages and add simple features such as markers and labels.

A. Start by reading the Getting Started documentation at https://developers.google.com/maps/documentation/javascript/tutorial During this section you will need to register for your own API Key - the process is painless and it is free - details are given in the documentation.
B. Next, work through the Adding a map with a Marker tutorial at https://developers.google.com/maps/documentation/javascript/adding-a-google-map For this exercise add a marker that shows your favourite place to eat.
C. [Optional] Work through the Geolocation tutorial at https://developers.google.com/maps/documentation/javascript/geolocation
D. There are a number of useful sections in the left-hand menu on the Google Developers web site such as Styling a Map, Drawing on the Map. Work through some that are of interest to you. There are no direct exercises for this but it would good that you become familiar with how to add customised legends, various marker types, and information windows. eg, Try adding an Info Window to the map you created in part B above.