**University of Bahrain**

**Information Technology College**

**Computer Science Department**

**ITCS 333/473 – Assignment 2**

**Part 1: (SIMPLE)**

Write the necessary PHP code to calculate the total salary according to the following equation:

***Total Salary = Basic Salary + Martial Status Allowance + Extra Tasks Allowance if Any***

Examine the following HTML code that describes how the form will submit the required input data. Note the following requirements and details:

* **BD 50** is paid as an allowance for single status employee and **BD 100** is paid for married.
* Assume the number of extra tasks are **unknown**. Extra allowances are part of checkboxes values.
* The form is submitted to the **same program**.
* The output should follow the **sample screen** shown below using **ordered and unordered lists.**
* Your program **shouldn’t display extras** in case the user didn’t choose any of the extra tasks options.

<html><body>

<?php

//YOUR CODE SHOULD BE WRITTEN HERE

?>

<form method=**'post'**>

Basic Salary <input name=**'basic'** /><br />

Marital Status: <br />

<input type='radio' name=**'ms'** value='single' checked/> Single <br />

<input type='radio' name=**'ms'** value='married' /> Married <br />

Extra: <br />

<input type='checkbox' name="**extra**[]" value="200#Work on Weekend" /> Work on Weekend - BD 200<br />

..

..

<input type='checkbox' name="**extra**[]" value="100#Work Night Shift" /> Work Night Shift - BD 100<br />

<input type='checkbox' name="**extra**[]" value="400#Work Abroad" />

Work Abroad - BD 400<br />

<input type='submit' value='Calculate Salary' />

</form></body></html>

**Sample Screens**

|  |  |
| --- | --- |
| **The form Looks Like** | **Output: When the form is submitted** |
| **Graphical user interface, text, application  Description automatically generated** | **Text  Description automatically generated** |

**Part 2:**

Extend Part 1 by adding another question as “**Save this details “ with a checkbox**.

* If the checkbox is ticked and the Calculate Salary is clicked, the details (total salary, extras, date and time) should be stored in one cookie that expires in 1 week. This cookie should be able to store history of the salary calculations in one cookie.
* The cookie name should be “**historyDetails**” and it’s values should be an array structure of your choice.
* **Note** that the cookie value is always string, to store an array inside the cookie, use **json\_encode()** and then store it in the cookie. To convert the cookie value to it’s original array structure, use **json\_decode()**. Refer to php.net for examples.
* **Tip**: The array should be able to hold unknown number of salary calculations with it’s details. Example: you might use the following nested arrays (numeric and associative) to push new details in cookie value:

$details[‘Total’] = 1500

$details[‘Extras’][ ] = “Work on Weekend”; //use loop

$details[‘Extras’][ ] = “Work Abroad”;

$details[‘Date’] = “6 March 2021”; //use date function

$details[‘Time’] = “16:07:11”; //use date function

$cookieValue[ ] =$details;

setcookie(“historyDetails”, json\_encode($cookieValue), ……);

* Don’t overwrite, Be careful, before adding details, you need to retrieve all the previous details from the cookie and add the new calculations to the array and save it back to the cookie.

**Part 3:**

Add another submit button near Calculate Salary with a caption “View History”. Once clicked, all salary calculations details that were stored in a cookie should be displayed in an HTML table of your own design.