## Implementation

### Description of new skills and/or knowledge researched and developed

### PHP

##### isset() Function

When programming my solution, I realized that if a session variable did not exist, the page would give an error as I had included conditions that would include a session variable when it was defined. For example, for my session variable 'userError', I would want an error message echoed on the Register page if the session variable was set to true. However, as the ‘userError’ would be defined in the newUser.php code - which would be executed when the form on the Register page was submitted - I would have an error on the Register page until the newUser.php code was ran and ‘userError’ was defined.

On W3Schools, I found out about the isset() function, which would check if a variable was empty or not - thus, it would check if the variable was defined or not. I found this function extremely useful as it helped me meet many of the requirements in my project - such as echoing the personalised message on the home page and redirecting users to error pages based on whether they were logged in or not (using the session variable ‘currentUser’ to dictate which user was using the account).

##### for and foreach loops

I included “For” and “Foreach” loops in my PHP code for calcuateScore.php to improve the efficiency of my code.

The PHP “For” loop operates similarly to the “For” loops I have been taught in the computing science course. The “For” loop requires parameters for a start point and end point but allows for a value to be included that will be increased with each iteration of the loop - this was called the increment counter. I found that the increment counter was helpful as it saved me from declaring a counter that would have to be increased within the loop itself - as I required a counter so that I could check that for each radio button name (from Q1 to Q10), there was a value so I could add to my PHP array.

The PHP “Foreach” loop would loop through for each element of an array. This was helpful as I wanted to check each item in the array $answers[] to see if they were null or not null (so I could check whether to execute the process for increasing the score for a correct answer as if the current item in the $answers[] array was null, I could skip the block of code to do via an IF statement), thus helping me fulfil my requirements.

I used W3Schools to help me implement this.

##### Arrays

I needed a PHP array to store all the input retrieved from the form “quiz”. The array helped make my code more efficient as I did not have to have a separate procedure for each input to check whether it was null or not and whether the input was equal to the answer from the question table or not.

I used W3Schools to help me implement this.

##### stristr() Function

I needed to compare each item in the $answers[] array to the variable $rightAns - which would hold the value of the answer for a certain question ID (based on the iteration of the loop). I realized that when comparing these two that they weren’t equal to each other - this could be due to white space in one of the values which would make the two strings unequal. After research, I found out about the stristr() function on the PHP documentation. This function would see if a string (called the ‘haystack’) would include a specified substring (called the ‘needle’) and then return the string searched on (haystack). Thus, I found if I used this function with $rightAns as the haystack and the current $answers[] item as the needle and compared the result of the stristr() function to the $rightAns - as $rightAns be returned if the needle appeared in the haystack and $rightAns is equal to $rightAns. Thus, I can conclude that this function was helpful to me as it helped me fulfil this requirement.

##### ctype\_alnum() Function

I needed to validate the inputs from the forms on the Register.php and Login.php pages to assess if they only included numbers and/or letters. In the PHP documentation, I found out about the ctype\_alnum(), which would check whether or not the parameter being passed in it was alphanumeric. The function helped me meet the requirements for validating the form input.

### Javascript

##### XMLHttpRequest

I needed a timer using PHP so I could store the time in which the user completes the quiz in a session variable and use the value of this session variable in my procedure for calculating the score. I would need a PHP script to be executed every 1000ms so that the session variable of time would decrease and act as a countdown. Through watching a video by Amit Andipara (on Youtube), I was able to find out how XMLHttpRequests worked and implement one in the context of my own code. Using W3Schools, I found out that XMLHttpRequests work by getting and processing data from a server - even after the page has loaded. Thus, using XMLHttpRequests, I was able to run the Response.php code every 1000ms so that I could decrease the value of the session variable time without having to reload the Quiz page, thus helping me to fulfil the requirements of my code.

##### Intervals

As I was incorporating time events on my Quiz page, I required a function that would run every second so I could make a countdown timer. On W3Schools, I found an example of a countdown that used Javascript Intervals. On further research, I found that intervals would run a function at specified intervals (in milliseconds). I used the setInterval() functions to set up my PHP timer (which would mean I could call my XMLHttpRequests every second/1000ms) and my Javascript timer that would display the time the user had left to complete the quiz. I used the clearInterval() functions to stop the timer when the form was submitted or after my countdown variables(var count and var timeLeft) had timed out - so it wouldn’t decrease any further with the function running on their intervals. This helped me fulfil my requirements.

#### SQL

##### CASE Statement

I needed to do an INSERT INTO statement with conditions from another table. On Stack Overflow, I found out that to do this I would need to use a CASE statement. On further research on W3Schools, I found that a CASE statement would operate like an IF statement - as if the conditions set out by the CASE statement were met, then certain values would be returned. So, I could use a CASE statement to evaluate that for the leaderboardRank I was inserting, it was corresponding to the account in use. This helped me fulfil my requirements.

### Ongoing Test Log

|  |  |  |
| --- | --- | --- |
| What was Tested | Issues Encountered | Resolution |
| Trying to register an account that shares the same username as another account | 1. Would not perform SQL query but would not notify the user 2. When implementing the error message, an error would appear if the session variable holding the userError was undefined | 1. Use a query to validate whether or not a username was taken or not. Use the result of this query in an IF statement to send an error message in a session variable if applicable 2. Use isset() function in a conditional statement to see if the session variable existed before executing the code to echo an error message |
| Trying to register with NULL values | 1. Would allow the user to register with NULL values | 1. Use ctype\_alnum() function to validate the form input and redirect back to the register/login page with an error message when applicable |
| Trying to register when already using an account | 1. Would allow a logged-in user to make/login into another account - after they done this, the account in use would be the newly made one | 1. Used the isset() function to see if an account was in use when a user was trying to access the login/register page - if so, redirect to an error page |
| Trying to call the questions and radio buttons for the form | 1. Using AJAX beforehand, whenever a question was loaded up, the inputs for the last question would be omitted. Thus, values would only be collected for one question | 1. Used PHP only instead to get all the questions and then echo the question and their respective radio buttons when the corresponding button was clicked - AJAX only works for updating details but not for storing the values for each question |
| Trying to add a new user to the leaderboard | 1. Would not perform the INSERT INTO statement as a WHERE clause cannot be used in the statement’s syntax | 1. Used CASE statement as subqueries in the values so that I could return the $score and $timecompleted as values only if the leaderboardRank corresponded to the current account |
| Trying to execute the response.php code every interval | 1. Would only call the code once, even though the code was called within an interval so it should be executed every 1000 milliseconds | 1. Used XMLHttpRequest so I could run the PHP code without having the page reloaded every time the response.php code was executed |

### Resources

1. <https://www.w3schools.com/>
   1. <https://www.w3schools.com/php/func_var_isset.asp>
   2. <https://www.w3schools.com/php/php_looping_for.asp>
   3. <https://www.w3schools.com/php/php_looping_foreach.asp>
   4. <https://www.w3schools.com/php/php_arrays.asp>
   5. <https://www.w3schools.com/xml/xml_http.asp>
   6. <https://www.w3schools.com/howto/howto_js_countdown.asp>
   7. <https://www.w3schools.com/jsref/met_win_setinterval.asp>
   8. <https://www.w3schools.com/sql/sql_case.asp>
2. <https://www.php.net/docs.php>
   1. <https://www.php.net/manual/en/function.stristr.php>
   2. <https://www.php.net/manual/en/function.ctype-alnum.php>
3. <https://stackoverflow.com/>
   1. <https://stackoverflow.com/questions/19304464/inserting-from-another-table-with-conditions-for-a-column>
4. <https://www.youtube.com/channel/UCH0VXb3oFVaV2_lpZesl_6Q>
   1. <https://www.youtube.com/watch?v=MabjuyWrDI4>

### Evidence

See next page(s)

##### 