RICHFIELD

FACULTY OF INFORMATION TECHNOLOGY

INTERNET PROGRAMMING 621

1ST SEMESTER ASSIGNMENT

Name & Surname: JESHUA LEONARD PILLAY **ICAS No:** 401914544

Qualification: BSc I.T (2nd Year) **Semester:** 1 **Module Name:** INTERNET PROGRAMMING 621

Date Submitted: 06 / 05 / 2021

ASSESSMENT CRITERIA	MARK ALLOCATION	EXAMINER MARKS	MODERATOR MARKS
MARKS FOR CONTENT			
QUESTION ONE	50		
QUESTION TWO	50		
TOTAL MARKS	100		
Examiner's Comments:			
Moderator's Comments:			
Signature of Examiner:	Signature of Moderator:		

TABLE OF CONTENTS

QUESTION 1

1.1) WRITE A PHP SCRIPT USING NESTED FOR LOOP THAT CREATES A CHESS BOARD AS SHOWN BELOW. USE TABLE WIDTH="270PX" AND TAKE 30PX AS CELL HEIGHT AND WIDTH.

This question is a good way to begin our assignment. In order to have a good understanding of the entire assignment, we will break down the code explaining each part of the code and finally show the entire code along with the output at the end for each question.

The requirements for this program are as follows:

- Use ***nested for-loops*** to create a table.
- Table width must be ***270px***.
- Height and width of the table cells must be ***30px***.
- Add ***CSS*** for presentation purposes.

Within our ***head tags*** we have a ***title tag*** and an ***external CSS*** file which we will look at a later stage:

```
<head>

    <title>Question 1.1</title>

    <link rel = "stylesheet" href = "Q1_1.css">

</head>
```

Within our ***body tags*** we have a simple ***h1 tag***:

```

<h1>PHP CHESS BOARD</h1>

```

Below our ***h1 tag*** contains the code that will bring our program to life. We open our ***table tags*** which contains ***embedded CSS*** to space out the cells of the table. We then open our ***PHP tags***. Within our ***PHP tags***, we have a ***nested for-loop***. The ***outer for-loop*** will make our rows while the ***inner for-loop*** will create our columns. Within the ***inner for-loop*** we will create a ***variable*** name ***total*** which will be the addition of the ***rows*** and ***columns***. We will then create an ***if-statement*** with a condition that will check for all even numbered blocks. We do this by using the ***modulus operator***. For all the ***even*** numbered blocks, the colour will be ***black*** while all the ***odd*** numbered blocks, the colour will be ***white***. Below is the code:

```
<table cellspacing = 0px cellpadding = 0px border = 1px>

    <?php

        for($row = 1; $row <= 8; $row++) {
            echo "<tr>";
```

```
              for($col = 1; $col <= 8; $col++) {
                $total = $row + $col;
                if($total%2 == 0) {
                    echo "<td height = 30px width = 30px bgcolor = black></td>";
                } else {
                    echo "<td height = 30px width = 30px bgcolor = white></td>";
                }
            }
            echo "<tr>";
        }

    ?>

    </table>
```

Let us now look at the entire ***HTML code***:

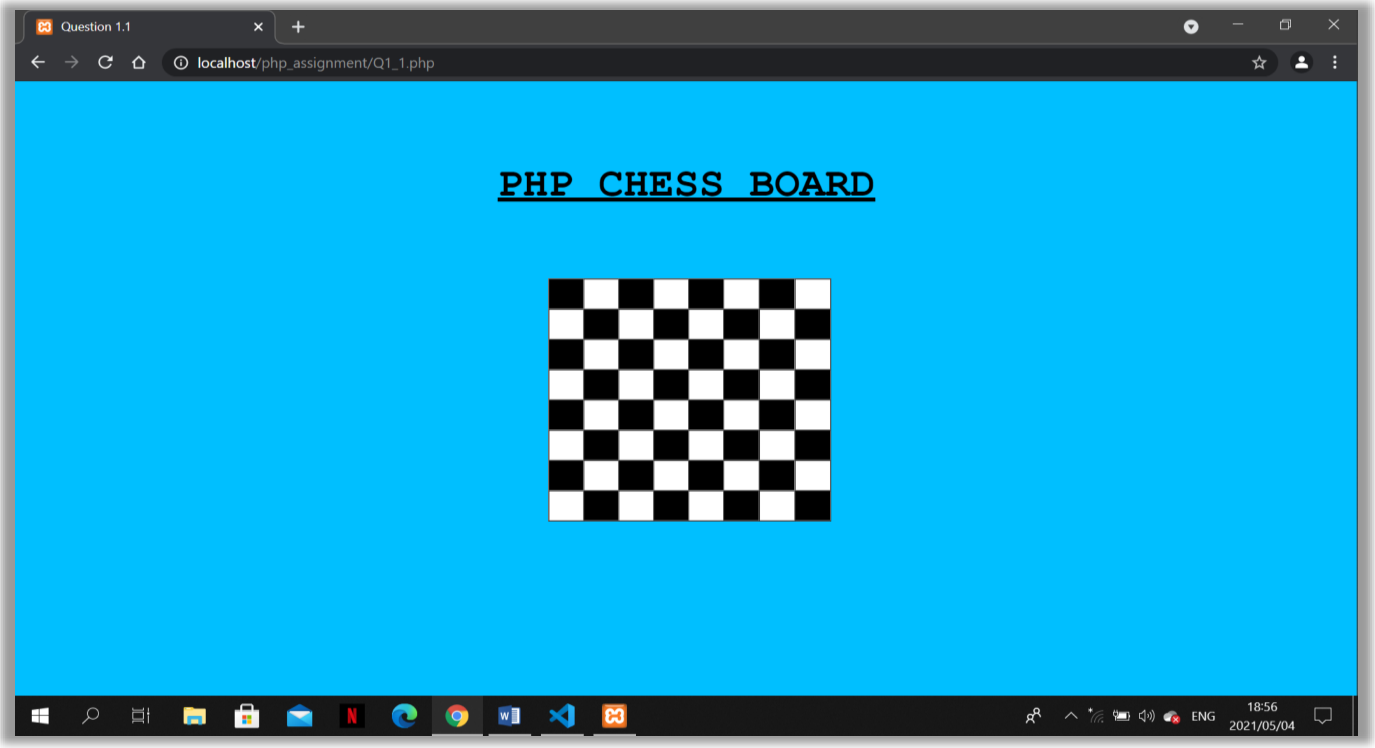
```
<html>

<head>

    <title>Question 1.1</title>

    <link rel = "stylesheet" href = "Q1_1.css">

</head>

<body>

    <h1>PHP CHESS BOARD</h1>

    <table cellspacing = 0px cellpadding = 0px border = 1px>

    <?php

        for($row = 1; $row <= 8; $row++) {
            echo "<tr>";
              for($col = 1; $col <= 8; $col++) {
                $total = $row + $col;
                if($total%2 == 0) {
                    echo "<td height = 30px width = 30px bgcolor = black></td>";
                } else {
                    echo "<td height = 30px width = 30px bgcolor = white></td>";
                }
            }
            echo "<tr>";
        }

    ?>

    </table>
```

```
</body>

</html>
```

Although our program is working, it does not look so attractive. Let us look at the external CSS file we mentioned earlier. Our CSS code will have styling for the ***body***, ***table*** and ***heading***. Below is the code for the external CSS file named ***Q1_1.css***:

```
body {
    background-color: deepskyblue;
}

table {
    width: 270px;
    margin-left: 500px;
    margin-top: 70px;
}

h1 {
    text-align: center;
    font-size: 40px;
    margin-top: 80px;
    font-family: 'Courier New', Courier, monospace;
    text-decoration: underline;
}
```

We have finally completed our program so let us now look at the output:



PHP CHESS BOARD

1.2) WRITE A PHP SCRIPT THAT CREATES THE FOLLOWING TABLE (USE FOR LOOPS).

This question is similar to the previous one we answered but it is much easier. The requirements for this program are as follows:

- Use for loops to create the table.
- Add ***CSS*** for presentation for presentation purposes.

Within our ***head tags*** we have a ***title tag*** and an ***external CSS file*** which we will look at a later stage. Below is the code:

```
<head>

    <link rel = "stylesheet" href = "Q1_2.css">

    <title>Question 1.2</title>

</head>
```

Within our ***body tags*** we have a simple ***h1 tag***:

```

<h1>Number Grid</h1>

```

Below our ***h1 tag*** is where the magic happens. We create a ***table*** using our ***table tags***. Within our table tags, we open our ***PHP tags***. Inside our PHP tags, we will have a ***nested for-loop***. The ***outer for-loop*** will make our rows while the ***inner for-loop*** will create our columns. Inside the nested for-loop, we will have a mathematical calculation by multiplying the rows and columns inside the ***table data tags***. This will print out our value for each cell in the table. Below is the code:

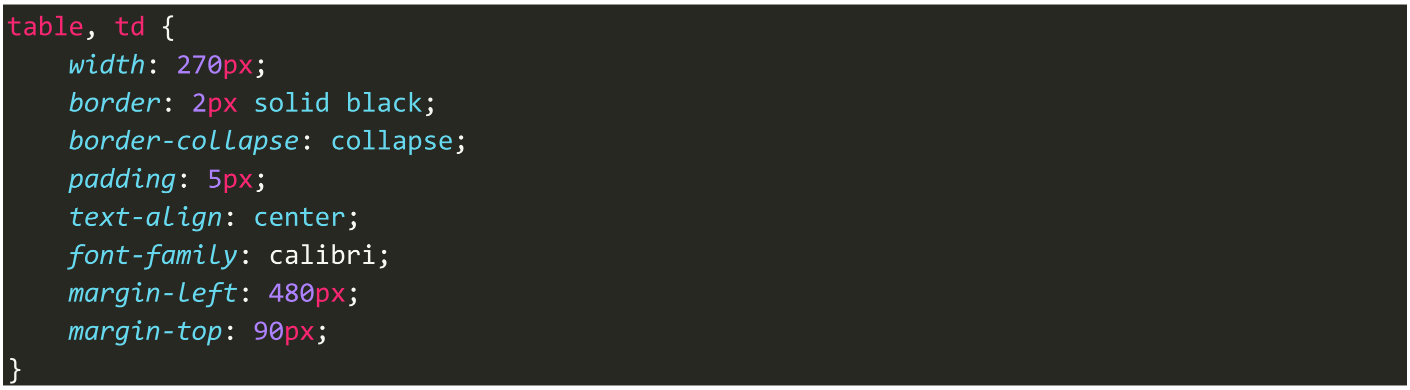
```
<table>

        <?php

            for($row = 1; $row < 11; $row++) {
                echo "<tr>";
                for($col = 1; $col < 11; $col++) {
                    echo("<td>" . $row * $col . "</td>");
                }
                echo "<tr>";
            }

        ?>

    </table>
```

Let us now look at the entire ***HTML code***:

```
<html>

<head>

    <link rel = "stylesheet" href = "Q1_2.css">

    <title>Question 1.2</title>

</head>

<body>

    <h1>Number Grid</h1>

    <table>

        <?php

            for($row = 1; $row < 11; $row++) {
                echo "<tr>";
                for($col = 1; $col < 11; $col++) {
                    echo("<td>" . $row * $col . "</td>");
                }
                echo "<tr>";
            }

        ?>

    </table>

</body>

</html>
```

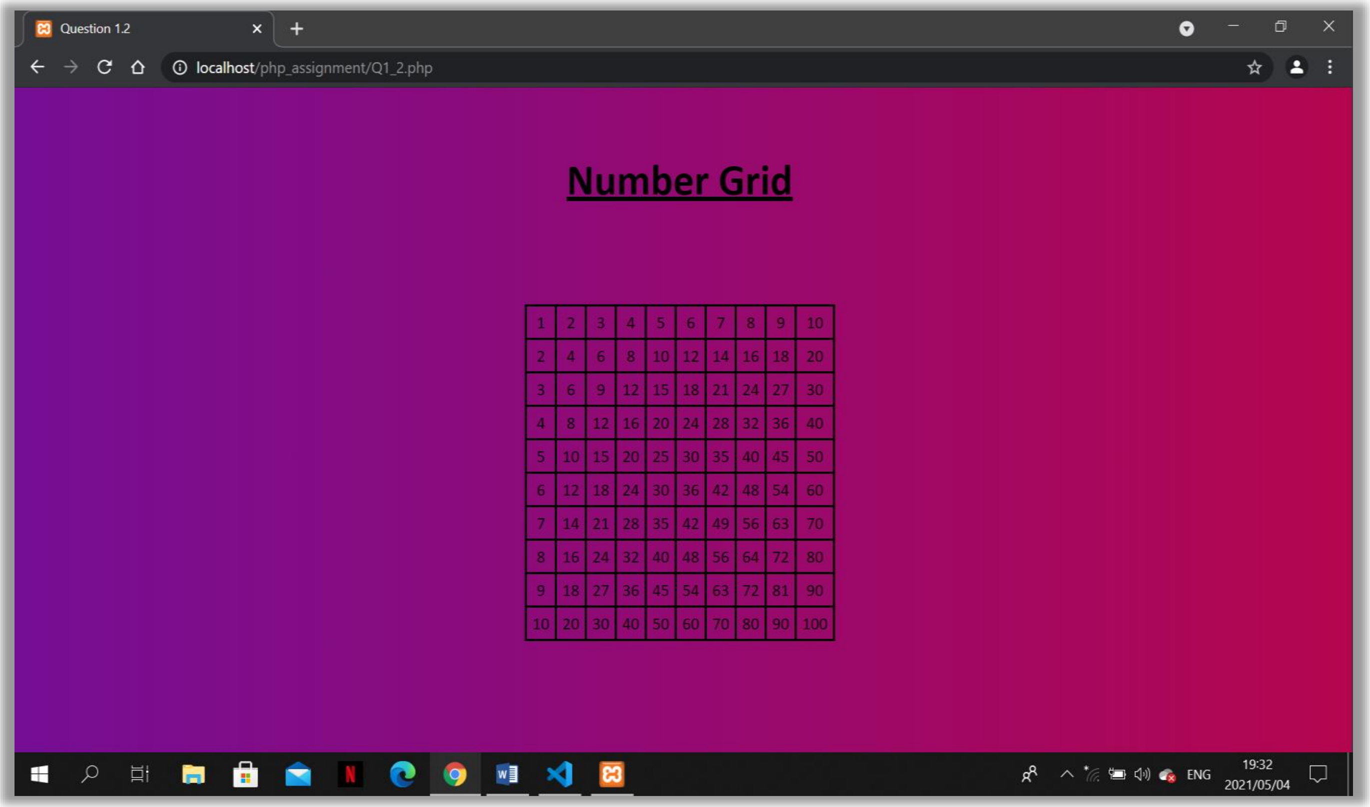
Our program is currently functional but it does not look appealing. Let us now look at the external CSS file named ***Q1_2.css*** that we mentioned earlier. We will have CSS for our ***body***, ***heading***, ***table*** and ***table cells***:

```
body {
    background: linear-gradient(to right, rgb(117, 14, 148), rgb(180, 6, 78));
}

h1 {
    margin-left: 520px;
    margin-top: 60px;
    text-decoration: underline;
    font-size: 40px;
    font-family: calibri;
}
```



```
table, td {
    width: 270px;
    border: 2px solid black;
    border-collapse: collapse;
    padding: 5px;
    text-align: center;
    font-family: calibri;
    margin-left: 480px;
    margin-top: 90px;
}
```

We have finally completed our program so let us now look at the output:



Number Grid

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

QUESTION 2

2.1) USING MULTI-DIMENSIONAL ARRAYS, CREATE AN ARRAY THAT WILL STORE 10 ITEMS OF STOCK (CAN BE ANY ITEM E.G. CARS,) WITH BRAND NAME, NUMBER OF STOCK IN STORE AND QUANTITY SOLD. NB: USE THE WHILE LOOP TO DISPLAY ARRAY ELEMENTS.

The requirements for this program are as follows:

- Use ***multi-dimensional arrays*** to store the data.
- Use ***while loop*** to display the items stored in the array.
- Add ***CSS*** for presentation purposes.

Within our ***head tags*** we will open up our ***style tags*** to have ***embedded CSS*** and outside our style tags we will have a ***title tag***. We will have CSS for our ***body***, ***heading***, ***table***, ***table headings*** and ***table cells***. The code for the CSS is shown below:

```
<style>

    body {
        background-color: black;
        color: white;
        font-family: calibri;
    }

    table, td, th {
        border: 2px solid white;
        border-collapse: collapse;
        padding: 10px;
        margin-left: 400px;
        margin-top: 30px;
    }

    h1 {
        margin-left: 455px;
        margin-top: 40px;
        font-size: 50px;
        text-decoration: underline;
    }

</style>

<title>Question 2.1_again</title>
```

Within our ***body tags*** we will have a simple ***h1 heading tag:***

```
<h1>Car Dealership</h1>
```

We will now open up our ***PHP tags***. Within our PHP tags, we will create a multi-dimensional array storing 10 arrays. In each array we will store a string value and 2 integer values. Below is the code:

```
$table = array(array("GTI", 15, 10), array("Polo", 20, 15), array("Golf", 25, 20),
               array("Audi", 30, 25), array("Toyota", 35, 30), array("Hyundai", 40, 35),
               array("Ferrari", 45, 40), array("TSI", 50, 45), array("Haval", 55, 50),
               array("Rangrover", 60, 55));
```

Below the multi-dimensional array we will print out ***HTML table tags*** containing the ***headings***:

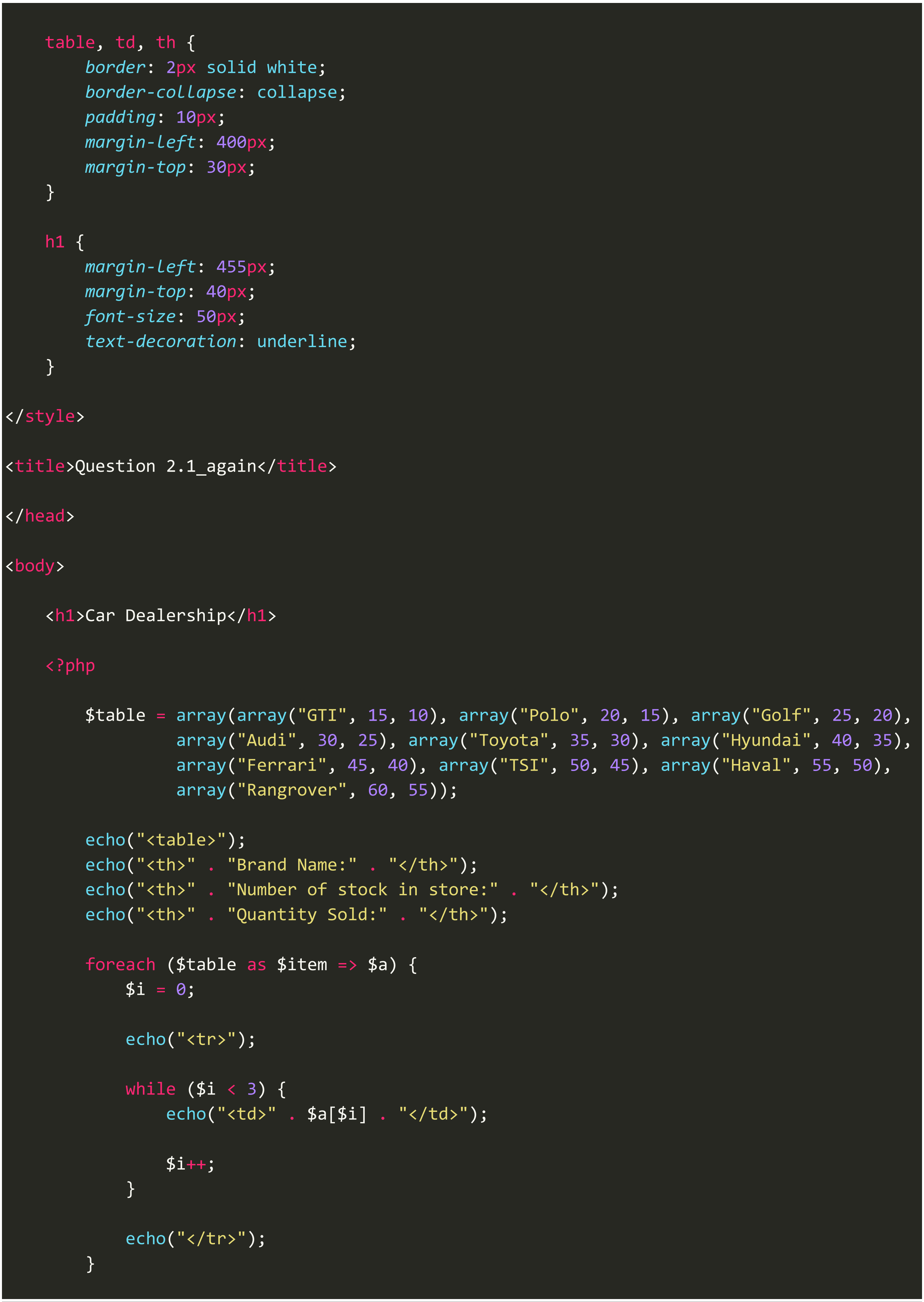
```
echo("<table>");
        echo("<th>" . "Brand Name:" . "</th>");
        echo("<th>" . "Number of stock in store:" . "</th>");
        echo("<th>" . "Quantity Sold:" . "</th>");
```

The next part of our code is crucial. We will create a ***foreach loop*** which will take the multi-dimensional arrays values and store it in variable ***$a***. We will then create a ***while loop*** to display the stored values. The embedded HTML code is used for displaying the results in a tabular format by organising the rows, columns and table cells. The code is shown below:

```
foreach ($table as $item => $a) {
            $i = 0;

            echo("<tr>");

            while ($i < 3) {
                echo("<td>" . $a[$i] . "</td>");

                $i++;
            }

            echo("</tr>");
        }

        echo("</table>");
```

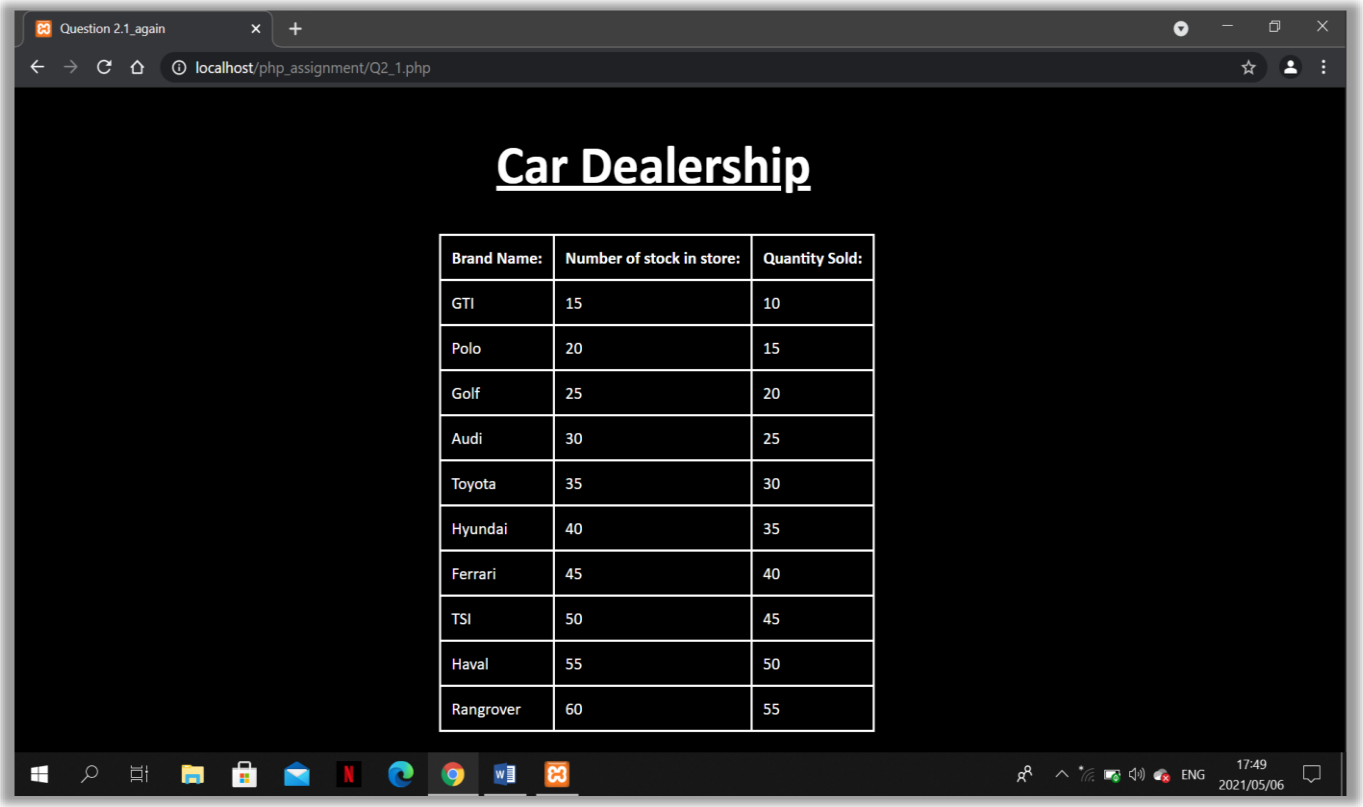
We have explained each part of the code so let us have a look at the entire code:

```
<html>

<head>

<style>

    body {
        background-color: black;
        color: white;
        font-family: calibri;
    }
```



```
    table, td, th {
        border: 2px solid white;
        border-collapse: collapse;
        padding: 10px;
        margin-left: 400px;
        margin-top: 30px;
    }

    h1 {
        margin-left: 455px;
        margin-top: 40px;
        font-size: 50px;
        text-decoration: underline;
    }

</style>

<title>Question 2.1_again</title>

</head>

<body>

    <h1>Car Dealership</h1>

    <?php

        $table = array(array("GTI", 15, 10), array("Polo", 20, 15), array("Golf", 25, 20),
                 array("Audi", 30, 25), array("Toyota", 35, 30), array("Hyundai", 40, 35),
                 array("Ferrari", 45, 40), array("TSI", 50, 45), array("Haval", 55, 50),
                 array("Rangrover", 60, 55));

        echo("<table>");
        echo("<th>" . "Brand Name:" . "</th>");
        echo("<th>" . "Number of stock in store:" . "</th>");
        echo("<th>" . "Quantity Sold:" . "</th>");

        foreach ($table as $item => $a) {
            $i = 0;

            echo("<tr>");

            while ($i < 3) {
                echo("<td>" . $a[$i] . "</td>");

                $i++;
            }

            echo("</tr>");
        }
        echo("</table>");

    ?>

</body>

</html>
```

Now that we have completed the code for our program, let us have a look at the output in our browser:



Car Dealership

Brand Name:	Number of stock in store:	Quantity Sold:
GTI	15	10
Polo	20	15
Golf	25	20
Audi	30	25
Toyota	35	30
Hyundai	40	35
Ferrari	45	40
TSI	50	45
Haval	55	50
Rangrover	60	55

We have come to the end of our question. We have been introduced to new concepts such as multi-dimensional arrays and while loops.

2.2) COMPARE AND CONTRAST THE IF/ELSEIF CONTROL STRUCTURE WITH THE SWITCH CONTROL STRUCTURED AND PROVIDE CODED EXAMPLES TO SUSTAIN YOUR ANSWER.

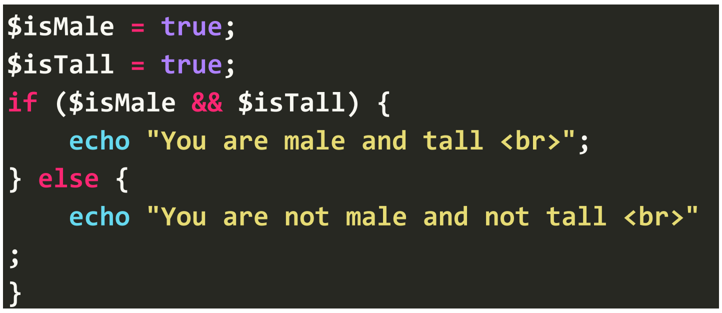
Comparison of:

IF/ELSEIF CONTROL STRUCTURE	***SWITCH CONTROL STRUCTURE***
If statements can be used for any occasion where comparisons need to be done. Each if statement checks a certain condition and when it is matched then it will execute the corresponding statement.	Switch control structures are good to use when the same value is compared against a set of values. In Switch control statements, since usually there are no conditions to evaluate, it can be bit efficient over if statements.

Contrast between:

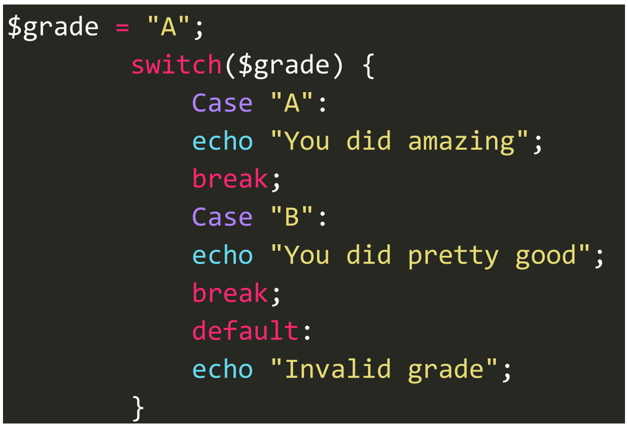
	IF/ELSEIF CONTROL STRUCTURE	**SWITCH CONTROL STRUCTURE**
Basic	The statement to be executed is decided by the output of the expression inside the if-statement.	The statement to be executed will be decided by the user.
Expression	If-else statement uses multiple statements for multiple choices.	Uses single expression for multiple choices.
Testing	Test for equality as well as for logical expression.	Test only for equality.
Evaluation	Evaluates integer, character, pointer or floating-point type or Boolean type.	Evaluates only character or integer value.
Sequence of execution	Either if statement will be executed or else statement is executed.	Switch statement execute one case after another till a break statement is appeared or the end of switch statement is reached.
Default execution	If the condition inside if statements is false, then by default the else statement is executed if created.	If condition does not match any of the cases then the default statement is executed if it is created.
Editing	Can become difficult to edit if-else statement, if the nested –else statement is used.	It is easy to edit switch cases as, they are recognized easily.

Example demonstrating the above points for if statements:



```
$isMale = true;
$isTall = true;
if ($isMale && $isTall) {
    echo "You are male and tall <br>";
} else {
    echo "You are not male and not tall <br>"
;
}
```

Example demonstrating the above Points for switch statements:



```
$grade = "A";
        switch($grade) {
            Case "A":
            echo "You did amazing";
            break;
            Case "B":
            echo "You did pretty good";
            break;
            default:
            echo "Invalid grade";
        }
```

2.3) USE LOOPS AND CONTROL STRUCTURES CREATE A PROGRAM THAT GRADES THE FOLLOWING LIST OF STUDENTS GIVEN THE GRADE TABLE BELOW:

We have arrived at our last question. The ***requirements*** for this program are as follows:

- Use loops and control structures.
- Display will be in a tabular format.
- Add ***CSS*** for presentation purposes.

Within our ***head tags*** we will open up our ***style tags*** to have ***embedded CSS***. We will have CSS for our ***body***, ***heading***, ***table*** and ***table*** cells. The code for the CSS is shown below:

```
<style>

    body{
        background-color: black;
        color: white;
    }

    h1 {
        font-family: calibri;
        font-size: 40px;
        text-decoration: underline;
        margin-left: 460px;
        margin-top: 100px;
    }

    table, td {
        border: 1.5px solid white;
        border-collapse: collapse;
        font-family: calibri;
        font-size: 20px;
        padding: 5px;
        margin-left: 400px;
        margin-top: 90px;
    }

</style>
```

We now move inside our ***body tags***. We will have a simple ***h1 tag***:

```
<h1>Student's Grades</h1>
```

We will now open up our ***PHP tags***. We will print out ***HTML table tags*** as follows:

```
echo("<table>");
    echo("<th>Name</th>");
    echo("<th>Mark</th>");
    echo("<th>Grade</th>");
```

We will now create an ***associative array*** storing the student’s names with their respective marks:

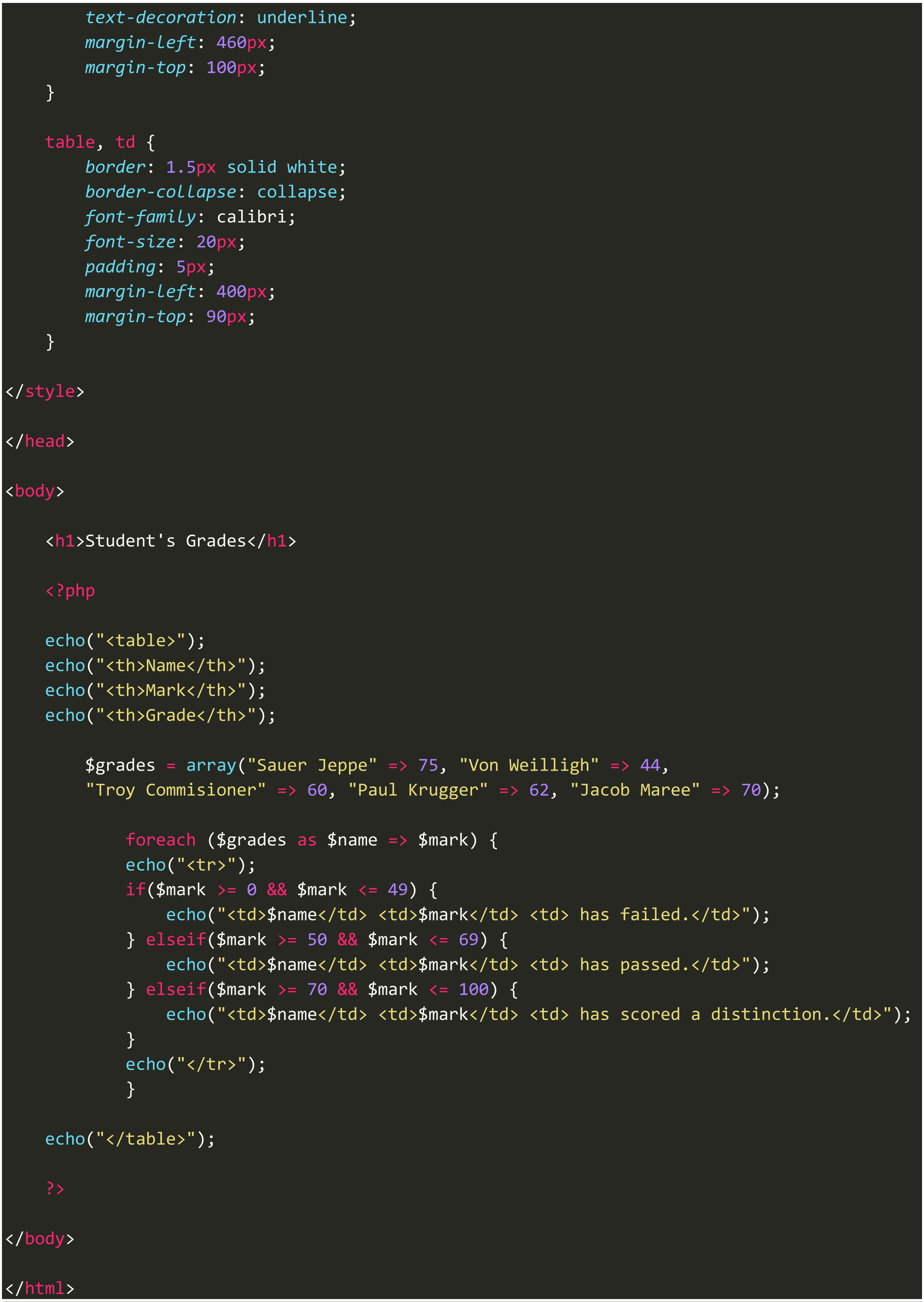
```
$grades = array("Sauer Jeppe" => 75, "Von Weilligh" => 44, "Troy Commisioner" => 60, "Paul Krugger" => 62, "Jacob Maree" => 70);
```

We will now have a ***foreach-loop*** and within our loop we will have an ***if-statement***. The if-statement will access the data in the associative array from the foreach-loop and match it with the appropriate conditional statement. Thereafter it will print out the results in a neat tabular format due to the embedded HTML tags. Below is the code:

```
foreach ($grades as $name => $mark) {
            echo("<tr>");
            if($mark >= 0 && $mark <= 49) {
                echo("<td>$name</td> <td>$mark</td> <td> has failed.</td>");
            } elseif($mark >= 50 && $mark <= 69) {
                echo("<td>$name</td> <td>$mark</td> <td> has passed.</td>");
            } elseif($mark >= 70 && $mark <= 100) {
                echo("<td>$name</td> <td>$mark</td> <td> has scored a distinction.</td>");
            }
            echo("</tr>");
            }

    echo("</table>");
```

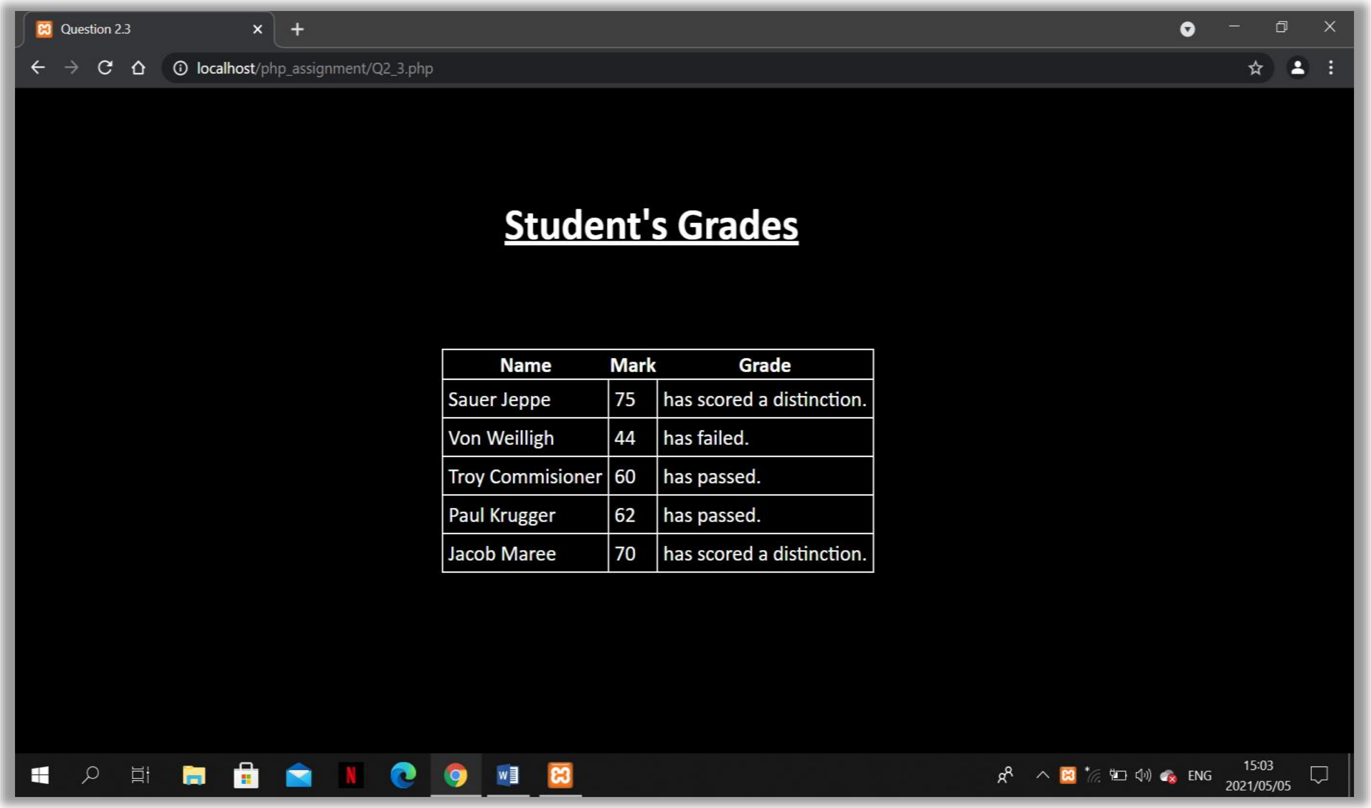
We have now completed explaining each part of the code so let us look at the entire code:

```
<html>

<head>

<title>Question 2.3</title>

<style>

    body{
        background-color: black;
        color: white;
    }

    h1 {
        font-family: calibri;
        font-size: 40px;
```



```
        text-decoration: underline;
        margin-left: 460px;
        margin-top: 100px;
    }

    table, td {
        border: 1.5px solid white;
        border-collapse: collapse;
        font-family: calibri;
        font-size: 20px;
        padding: 5px;
        margin-left: 400px;
        margin-top: 90px;
    }

</style>

</head>

<body>

    <h1>Student's Grades</h1>

    <?php

    echo("<table>");
    echo("<th>Name</th>");
    echo("<th>Mark</th>");
    echo("<th>Grade</th>");

        $grades = array("Sauer Jeppe" => 75, "Von Weilligh" => 44,
        "Troy Commisioner" => 60, "Paul Krugger" => 62, "Jacob Maree" => 70);

            foreach ($grades as $name => $mark) {
            echo("<tr>");
            if($mark >= 0 && $mark <= 49) {
                echo("<td>$name</td> <td>$mark</td> <td> has failed.</td>");
            } elseif($mark >= 50 && $mark <= 69) {
                echo("<td>$name</td> <td>$mark</td> <td> has passed.</td>");
            } elseif($mark >= 70 && $mark <= 100) {
                echo("<td>$name</td> <td>$mark</td> <td> has scored a distinction.</td>");
            }
            echo("</tr>");
            }

    echo("</table>");

    ?>

</body>

</html>
```

Let us now look at the output of our program:



Student's Grades

Name	Mark	Grade
Sauer Jeppe	75	has scored a distinction.
Von Weilligh	44	has failed.
Troy Commisioner	60	has passed.
Paul Krugger	62	has passed.
Jacob Maree	70	has scored a distinction.

We have finally come to the end of our assignment. This assignment has tested vital core concepts of PHP and in doing so, it has equipped us with valuable skills to make our websites dynamic.

REFERENCES

Bitoni, A. (2021, March 21). *PHP Arrays*. Retrieved from w3schools: https://www.w3schools.com/php/php_arrays.asp

Cooper, J. (2021, January 4). *PHP for Loop*. Retrieved from w3schools: https://www.w3schools.com/php/php_looping_for.asp

Doberman, R. (2020, December 23). *PHP for loop*. Retrieved from w3resource: https://www.w3resource.com/php-exercises/php-for-loop-exercise-4.php

Hamilton, G. (2021, January 5). *Difference Between if-else and switch*. Retrieved from tech differences: https://techdifferences.com/difference-between-if-else-and-switch.html#:~:text=The%20if%20statement%20evaluates%20integer,else%20block%20statement%20will%20execute.

Mitchell, J. (2020, April 24). *PHP Operators*. Retrieved from w3schools: https://www.w3schools.com/php/php_operators.asp

Norman, R. (2020, November 12). *PHP Multidimensional Arrays*. Retrieved from w2schools: https://www.w3schools.com/php/php_arrays_multidimensional.asp

Nowitski, L. (2021, February 11). *HP if...else...elseif Statements*. Retrieved from w3schools: https://www.w3schools.com/php/php_if_else.asp

Snyder, W. (2021, April 2). *PHP Variables*. Retrieved from w3schools: https://www.w3schools.com/php/php_variables.asp