ECS417U Lab Sheet
Topic 9 - PHP Arrays, Forms and Sessions

For these exercises you will need to upload the .php files to your webprojects directory.

Exercise 1

Debug the following PHP document:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="utf-8">
   <link rel="stylesheet" href="exercise1.css" />
   <title>Week 10 - Exercise 1</title>
</head>

<body>

      <?php
            $count =1;
            While {$count <=100){
                  if ($count % 2)
                  print "<p> **** </p>"
                  else
                  print "<p> ++++++++</p>";
                  count ++;
                  }
      ?>


</body>

</html>
```

Exercise 2

Write two PHP functions which takes an array of numbers as an argument and return the average and median of the array.

Exercise 3

Write a PHP script to create an array named `employee_salary`. The keys of the array are names of employees and the values are their salaries. For this exercise, you will need to refer to the PHP manual (add link) to identity functions that you will need to use to:

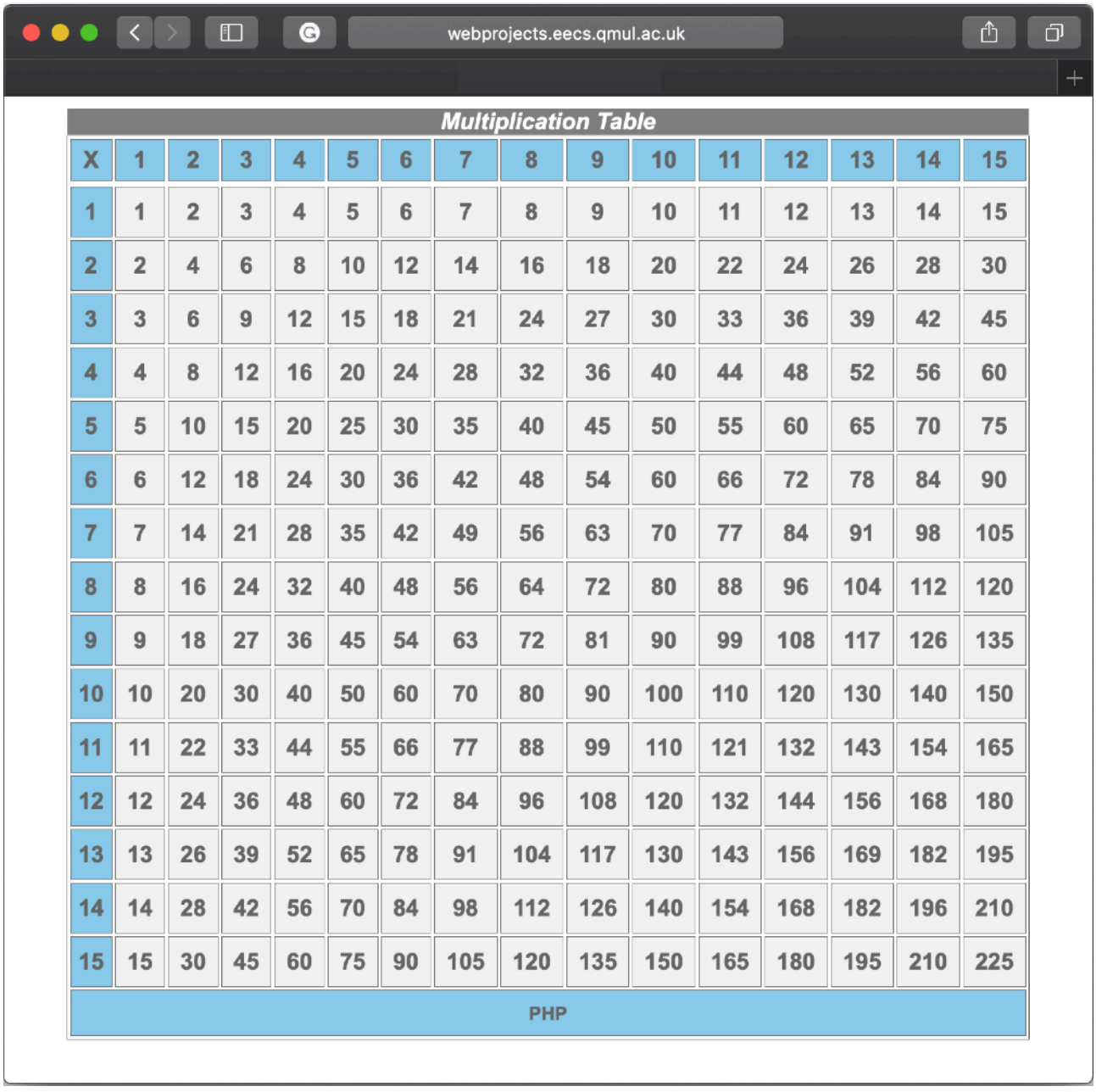
- Sort the array according to the salary of the employees.
- Sort the array by the names of the employees.

Use the `foreach` statement to output the keys and values of the array before and after sorting.

Exercise 4

Part 1

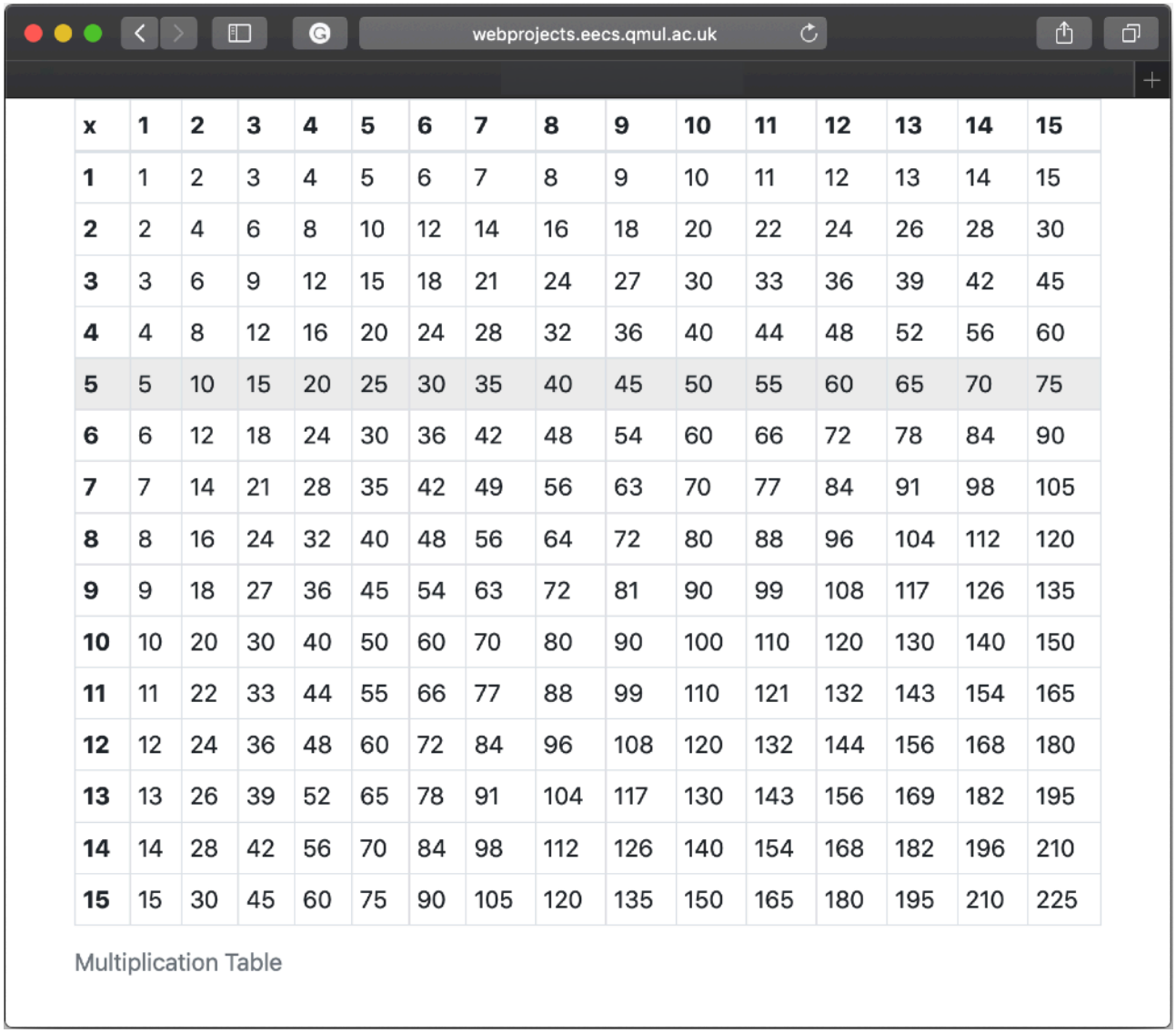
Create the multiplication table shown in figure 1 using PHP. Like the exercise in ***Topic 6***, you will need to use HTML5 elements such as `<caption>` , `<thead>` and `<tfoot>`. In addition to this, you will also need to create an external css file for styling the table.



X	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
5	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75
6	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90
7	7	14	21	28	35	42	49	56	63	70	77	84	91	98	105
8	8	16	24	32	40	48	56	64	72	80	88	96	104	112	120
9	9	18	27	36	45	54	63	72	81	90	99	108	117	126	135
10	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150
11	11	22	33	44	55	66	77	88	99	110	121	132	143	154	165
12	12	24	36	48	60	72	84	96	108	120	132	144	156	168	180
13	13	26	39	52	65	78	91	104	117	130	143	156	169	182	195
14	14	28	42	56	70	84	98	112	126	140	154	168	182	196	210
15	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225

FIGURE 1

Part 2
For this exercise, you will need to use Bootstrap classes to style the multiplication table as shown in figure 2.



x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
5	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75
6	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90
7	7	14	21	28	35	42	49	56	63	70	77	84	91	98	105
8	8	16	24	32	40	48	56	64	72	80	88	96	104	112	120
9	9	18	27	36	45	54	63	72	81	90	99	108	117	126	135
10	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150
11	11	22	33	44	55	66	77	88	99	110	121	132	143	154	165
12	12	24	36	48	60	72	84	96	108	120	132	144	156	168	180
13	13	26	39	52	65	78	91	104	117	130	143	156	169	182	195
14	14	28	42	56	70	84	98	112	126	140	154	168	182	196	210
15	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225

Multiplication Table

FIGURE 2

Exercise 5

Create a HTML5 document that includes a form to gather the following information:

- A text input type to collect the user's name

- Four checkboxes, one each for the following items:
  - Four 100-watt light bulbs for £2.39
  - Eight 100-watt light bulbs for £4.29
  - Four 100-watt long-life light bulbs for £3.95
  - Eight 100-watt long-life light bulbs for £7.49

- A text input type to collect the number of battery packs ordered by the user at the price of £10.42 each.

- A collection of three radio buttons that are labelled as follows:
    A. Visa
    B. Master Card
    C. American Express

The form action should be `exercise5.php` and method should be `post`.

Then write a PHP script (`exercise5.php`) that computes the total cost of the ordered light bulbs and battery packs after adding 20 percent VAT. The program must inform the buyer of exactly what was ordered in a table, and what credit card was chosen to make payment.

You will also need to create an external style sheet called reset.css that removes all the browser formatting and another style sheet that will specify the styling.

Exercise 6

Create a HTML5 document (exercise6.html) that includes a form to collect the following registration information (figure 3):

- First Name - use HTML validation (`required`)
- Surname - use HTML validation (`required`)
- Email Address - use HTML validation (`required`)
- Password
- Confirm Password

The form action should be exercise6.php and method should be `post`.

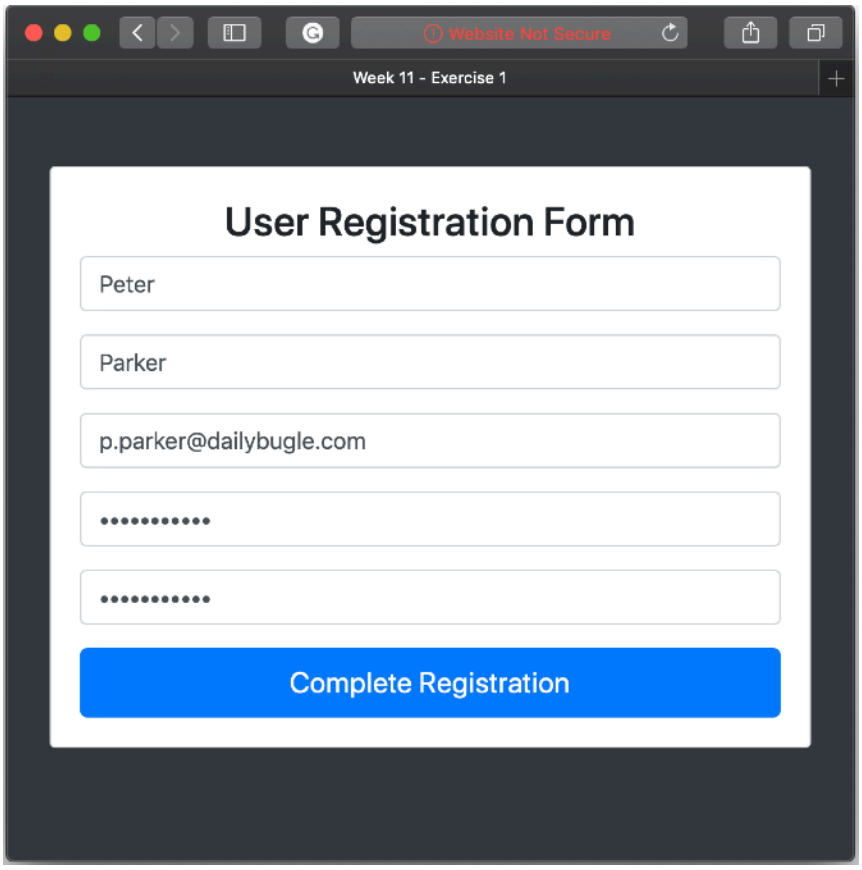


FIGURE 3

Create an external JavaScript file (exercise6.js), which carries out validation for ‘Password’ and ‘Confirm Password’ fields to ensure that both of these fields should be the same and not blank.

Then create a PHP script (exercise6.php), which receives the input from the form in exercise6.html. The objective is to insert the data from the form into a MySQL database.

Create a Database using phpMyAdmin (XAMPP)

Note: ensure that XAMPP is turned on.

1. Type the following URL: http://localhost:8080/phpmyadmin/

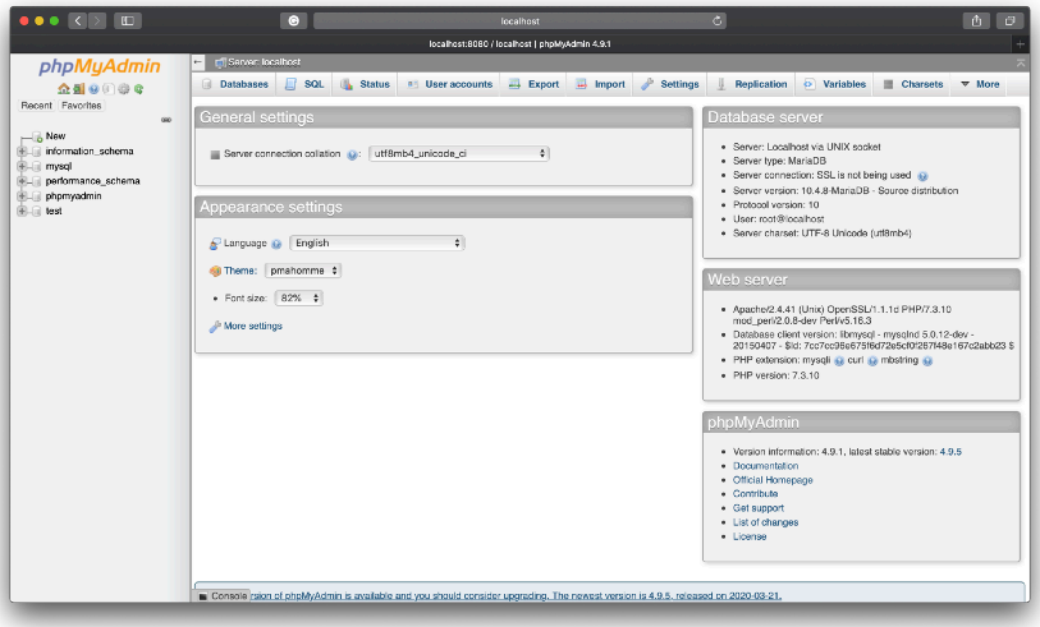


FIGURE 4

2. Click on ‘Databases’
3. Enter the name ‘ecs417’ for the database then click on the Create button. You will see that the database has been created in the left hand panel.

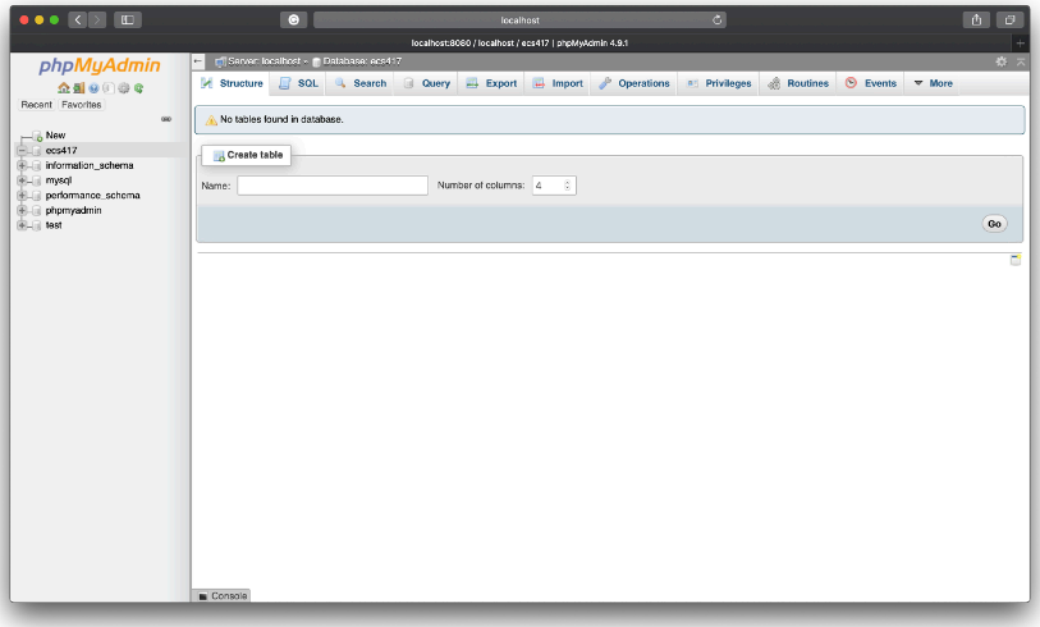


FIGURE 5

4. Once the database has been created, you will need to click on the 'SQL' link and paste the following query as shown in figure 6:

```
CREATE TABLE USERS (
    ID int NOT NULL AUTO_INCREMENT,
    firstName varchar(255),
    lastName varchar(255),
    email varchar(255),
    password varchar(255),
    PRIMARY KEY (ID)
);
```

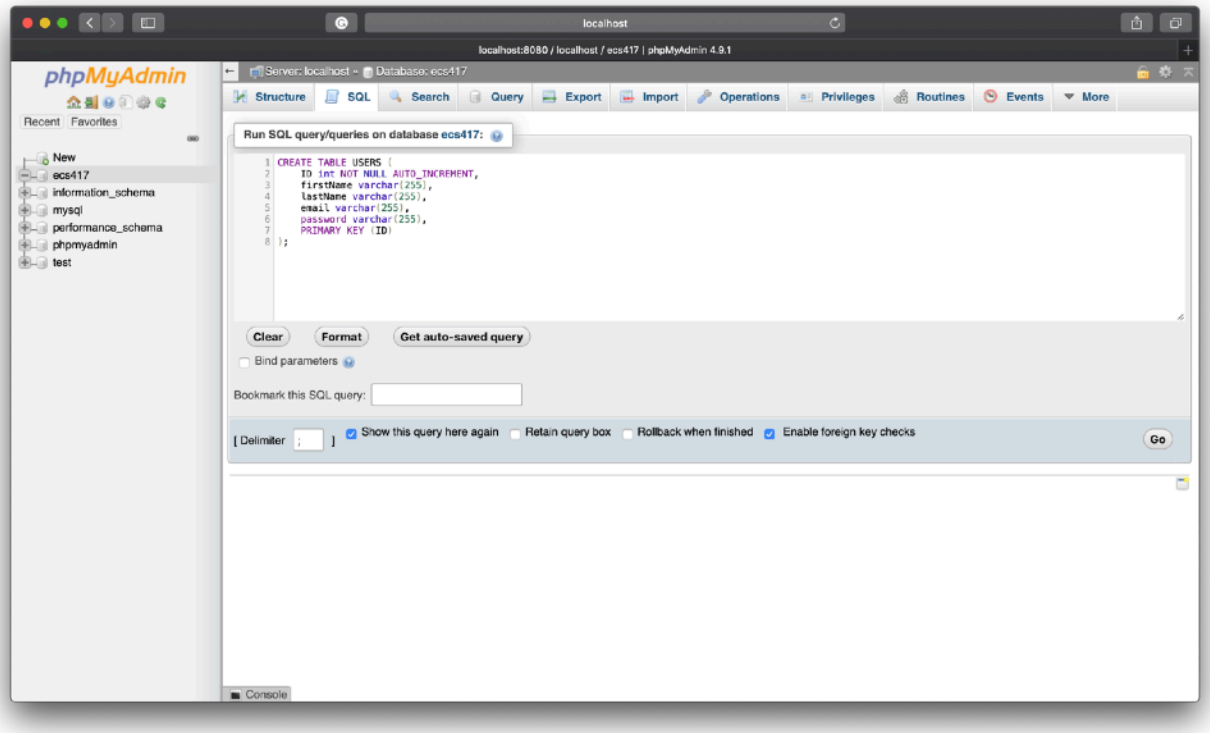


FIGURE 6

5. Then click 'Go'
6. As shown in figure 7, you should now be able to see your table 'USERS'.

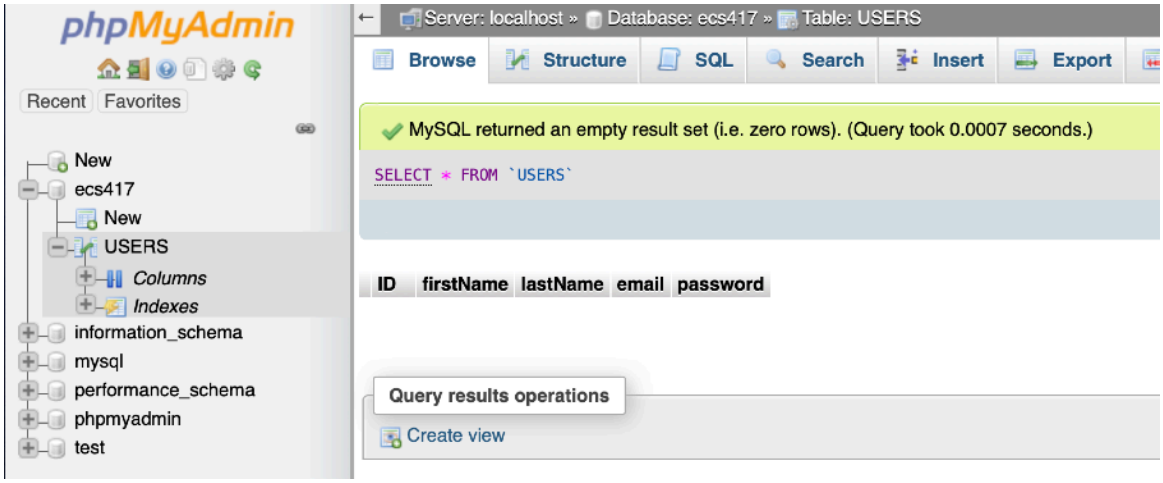


FIGURE 7

PHP Script - Inserting Data

In order to insert data using your PHP script, you will need to form a connection with the database using the following code:

```
$servername = "127.0.0.1";
$username = "root";
$password = "";
$dbname = "ecs417";

// Creates connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Checks connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

Below is example how you will insert data into the table you created in the previous step.

```
          if ($_SERVER['REQUEST_METHOD'] == 'POST'){

          $sql = "INSERT INTO USERS (firstName, lastName, email,
password) VALUES ('$fname', '$sname', '$email', '$pass1')";

          if ($conn->query($sql) === TRUE) {

          //YOUR CODE

          } else {
              echo "Error: " . $sql . "<br>" . $conn->error;
          }

          $conn->close();
        }
```

If the data has successfully been added into the table, then registration is completed and the output should be similar to figure 8.

Note: the styling of the registration pages has been done with Bootstrap classes.

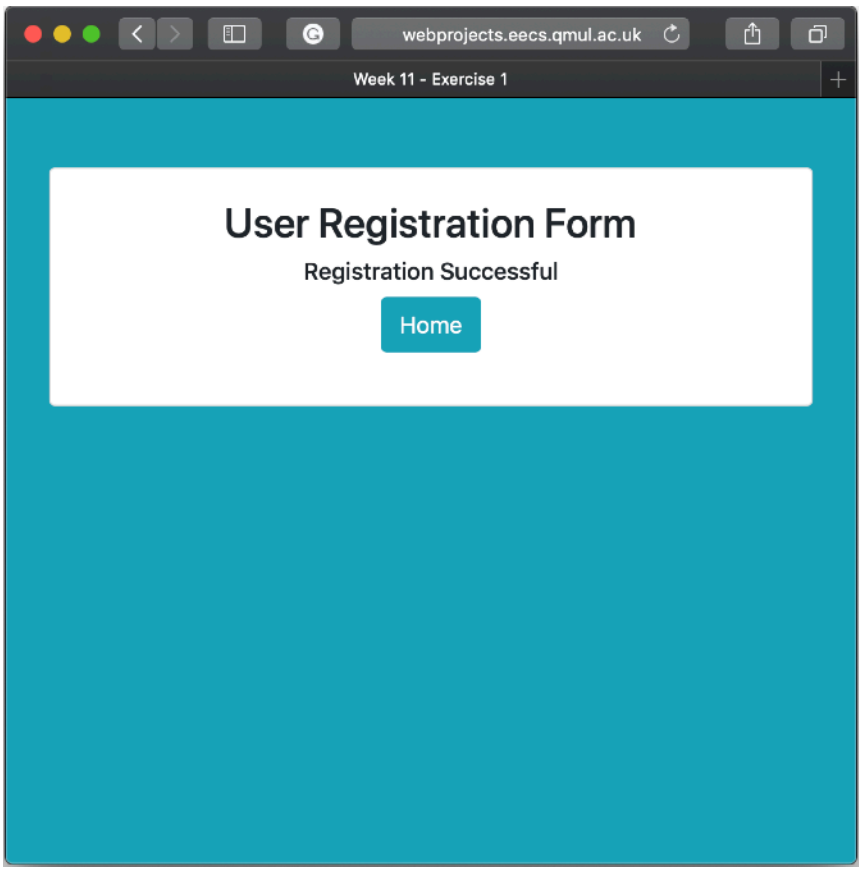


FIGURE 8

Exercise 7

For this exercise, you will create a login page (figure 9) with an authentication system that remembers a valid login using sessions and allows users to log out.

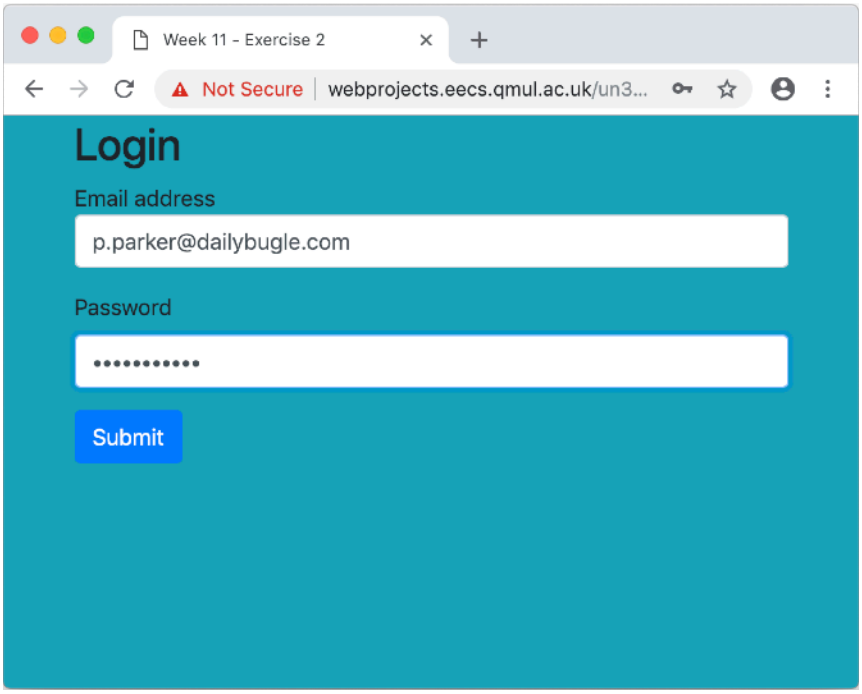


FIGURE 9

Part 1 - Authentication

7. Create a PHP script (exercise7.php) that includes a form to collect login information (figure 9). The form action should be `<?php echo $_SERVER['PHP_SELF'] ?>` and the method should `post`.

8. Use the SQL USERS table from the previous task to access the user credentials.

9. Your PHP script will need to handle the submitted form (figure 9) by validating the credentials against users in your database. If login is successful then display a welcome message (figure 10); otherwise, display an error message.

Note: the styling of the login page has been done with Bootstrap classes.

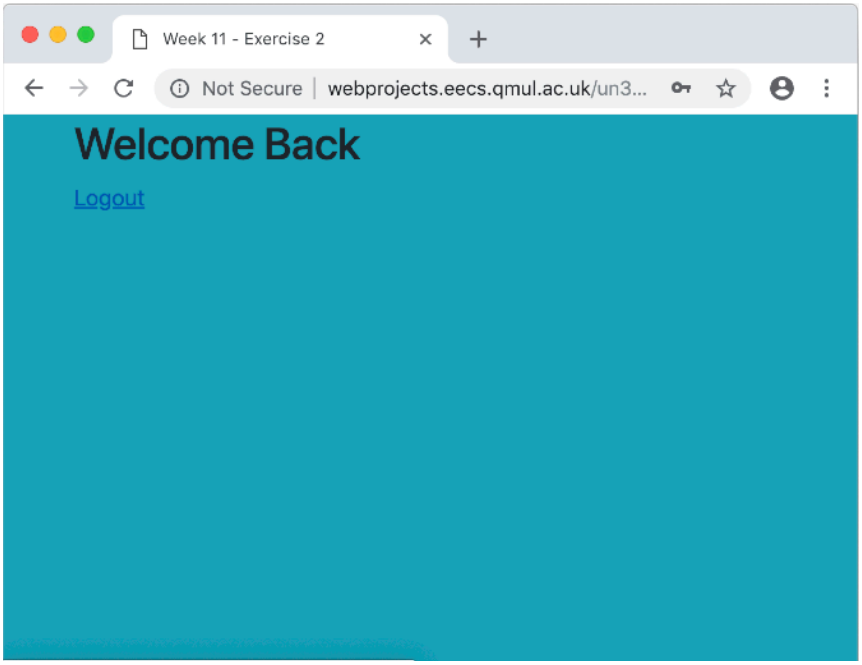


FIGURE 10

Part 2 - Session

1. Upon successful authentication, you will need to start a session. This will be done by adding the `session_start()` to your PHP script to add session functionality. This will be done by setting a new `$_SESSION['UserID']` variable to hold a value associated with the session.

2. Your PHP script should check the `$_SESSION` variable to see if a user ID has been set. If the session has not been set, then the script should display a login page as shown in figure 9, otherwise the output should be a welcome message with a log out link (to logout.php) as shown in figure 10.

3. Create another PHP script (logout.php), which will:
   - Call the `session_start()` function
   - Resets all the `$_SESSION` variables
   - Calls `session_destroy()`,
   - Redirects back to the page it was clicked from by using the `$_SERVER['REFERER']` value.