

Digital Business Application
Reflective report
Coursework 2
Week 7- Week 13

<http://daydream.lsbu.ac.uk/myweb/lamt3/>
<http://meno.lsbu.ac.uk/lamt3/>

Report Commissioned by:

Mike Child

Report Written by

Student ID: 3215402

Student Name: Timothy Lam

Date of Report: Friday 15th January 2015

Time of Report: 21:37

Table of Contents

| | |
|---|----|
| 1. Introduction | 1 |
| 2. Work Done | 1 |
| 2.1. Week 7 – PHP and MySQL..... | 1 |
| 2.1.1. Exercise 1 MySQL | 1 |
| 2.2. Week 8 – PHP and MySQL..... | 11 |
| 2.2.1. Exercise 2 PHPMyAdmin..... | 11 |
| 2.3. Week 9 – PHP and MySQL..... | 14 |
| 2.3.1. Exercise 2: PHPMyAdmin..... | 14 |
| 2.3.2. Extended Task: Exercise 2- PHPMyAdmin | 16 |
| 2.3.3. Exercise 3 Photo Album Server Version B and C..... | 16 |
| 2.3.3.1. Photo Album Sever Version B | 17 |
| 2.3.3.2. Photo Album Sever Version C | 19 |
| 2.3.4. Exercise 4: Photo Album Server Version D | 21 |
| 2.3.5. Exercise 5: PHP Session and Authentication | 24 |
| 2.3.6. Exercise 6: Photo Album Server Version E | 28 |
| 2.4. Week 10 – JavaScript and Ajax..... | 32 |
| 2.4.1. Exercise 1: Photo Album Client-Rendered Version A..... | 32 |
| 2.5. Week 11 - JavaScript and Ajax..... | 36 |
| 2.5.1. Exercise 2: Photo Album Client-Rendered Version B | 36 |
| 2.5.2. Exercise 3: Photo Album Client-Rendered Version C | 39 |
| 2.6. Week 12 - JavaScript and Ajax..... | 43 |
| 2.6.1. Exercise 4: Photo Album Client-Rendered Version D..... | 43 |
| 2.6.2. Exercise 5: Photo Album Client-Rendered Version E | 46 |
| 2.6.3. Exercise 6: Extended Task - Photo Album Server Version E | 53 |
| 2.7. Week 13 | 56 |
| 2.7.1. Extended task: Exercise 2: Books | 56 |
| 2.7.2. Additional Task..... | 58 |
| 3. Conclusion..... | 66 |

1. Introduction

Before I started this module I had some knowledge of Database, as I only knew what a database is used for which was to store data. However, I had never studied PHP and MySQL or JavaScript and Ajax before. In relation to completing the exercises, I managed to complete most of the initial exercises in the lessons, and those that I did not complete I finished at home. In relation to the extended tasks, I completed the extended tasks for Exercises 6: **Photo Album Sever Version E** and Extended task for Exercise 2: **PHP MyAdmin**, but did not manage to do the extended tasks for PHP and MySQL Exercise which was the following Exercise 1: **MySQL**, Exercise 3: **Photo Album Server Version B and C**, and Exercise 4: **Photo Album Server Version D**. What is more, the other extended task I did not complete was JavaScript and AJAX Exercise which was Exercise 1: **Photo Album Client-Rendered Version A**. The reason I did not complete all of the extended tasks was that some were too difficult and challenging. I did extra reading for Exercise 5: **Session and Authentication** and Exercise 6: **Photo Album Server Version E** in PHP and MySQL also Exercise 5 **Photo Album Client-Rendered Version E**, as I found these exercises the most challenging.

2. Work Done

2.1. Week 7 – PHP and MySQL

2.1.1. Exercise 1 MySQL

Meno: dbtest.php: <http://meno.lsbu.ac.uk/lamt3/dbtest.php>

Daydream: dbtest.php: <http://daydream.lsbu.ac.uk/myweb/lamt3/dbtest.php>

The Exercise was Exercise 1: **MySQL** where I had to create a copy of the following code and save it as **dbutils.php** in a **lib** folder.

```
<?php

function connect() {
    $username = '';
    $password = '';

    $mysqlhost = 'bcimsql.lsbu.ac.uk';
    $dbname = $username;
    $pdo = new PDO('mysql:host='.$mysqlhost.';dbname='.$dbname.';charset=utf8',
    $username, $password);
    if ( $pdo ) {
        // make errors throw exceptions
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        return $pdo;
    } else {
        die("Could not create PDO connection.");
    }
}

?>
```

The next thing was to copy another code and save it outside the **lib** folder and save the file called **dbtest.php**

```
<?php
    require_once( 'lib/dbutils.php' );
?>
<!DOCTYPE html>
<html>
<head>
<title>DB Test</title>
</head>
<body>
```

```

<?php

$pdo = connect();

if ( $pdo) {
    print "<h1>Successfully connected!</h1>\n";
} else {
    print "<h1>Did not connect.</h1>\n";
}

?>

</body>
</html>

```

Another thing I had to do is to copy another code and save it inside **dbutils.php** after the function

```

function tableExists( $pdo, $table) {
    $sql = "SHOW TABLES LIKE '$table'";
    return ( $pdo->query( $sql)->rowCount() > 0);
}

function createDemoTable( $pdo) {
    if ( !tableExists( $pdo, 'demotable')) {
        $sql = "CREATE TABLE IF NOT EXISTS `demotable` (
            `idnumber` int(11) NOT NULL auto_increment,
            `firstname` varchar(255) NOT NULL,
            `lastname` varchar(255) NOT NULL,
            `age` int(11) NOT NULL,
            PRIMARY KEY (`idnumber`)
        ) ";
        $pdo->exec( $sql);
        $sql = "INSERT INTO `demotable`
            (`firstname`, `lastname`, `age`)
            VALUES
            ('John','Smith',23),
            ('Sue','Davis',13),
            ('Peter','Young',45),
            ('Alice','Brown',56),
            ('Frank','Gray',22),
            ('Betty','Redwood',27),
            ('Jane','Corner',29),
            ('Bob','Church',19),
            ('Mary','Collins',67),
            ('Herbert','River',42),
            ('Lucy','Hardy',13),
            ('Charles','Winter',81),
            ('Samantha','Weather',37),
            ('Karl','Saunders',34),
            ('Angela','Brown',35)";
        $pdo->exec( $sql);
    }
}

function dumpTable( $pdo, $table) {
    $sql = "SELECT * FROM `".$table."`";
    $stmt = $pdo->query( $sql);
    // get all rows in an array
    $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
    print_r( $results);
}

```

Another thing I had to do is to add the following code to **dbtest.php** and upload one file and the lib folder containing **dbutils.php** and **dbtest.php** to the server **daydream.lsbu.ac.uk**

```

createDemoTable( $pdo);
print "<pre>\n";
dumpTable( $pdo, 'demotable');
print "</pre>\n";

```

After uploading the files, I had to test if the web page worked

Test Outcomes:

Successfully connected!

Array

```
(
  [0] => Array
    (
      [idnumber] => 1
      [firstname] => John
      [lastname] => Smith
      [age] => 23
    )

  [1] => Array
    (
      [idnumber] => 2
      [firstname] => Sue
      [lastname] => Davis
      [age] => 13
    )

  [2] => Array
    (
      [idnumber] => 3
      [firstname] => Peter
      [lastname] => Young
      [age] => 45
    )

  [3] => Array
    (
      [idnumber] => 4
      [firstname] => Alice
      [lastname] => Brown
      [age] => 56
    )

  [4] => Array
    (
      [idnumber] => 5
      [firstname] => Frank
      [lastname] => Gray
      [age] => 22
    )

  [5] => Array
    (
      [idnumber] => 6
```

```
    [firstname] => Betty
    [lastname] => Redwood
    [age] => 27
  )

[6] => Array
(
    [idnumber] => 7
    [firstname] => Jane
    [lastname] => Corner
    [age] => 29
)

[7] => Array
(
    [idnumber] => 8
    [firstname] => Bob
    [lastname] => Church
    [age] => 19
)

[8] => Array
(
    [idnumber] => 9
    [firstname] => Mary
    [lastname] => Collins
    [age] => 67
)

[9] => Array
(
    [idnumber] => 10
    [firstname] => Herbert
    [lastname] => River
    [age] => 42
)

[10] => Array
(
    [idnumber] => 11
    [firstname] => Lucy
    [lastname] => Hardy
    [age] => 13
)

[11] => Array
(
    [idnumber] => 12
    [firstname] => Charles
    [lastname] => Winter
    [age] => 81
)
```

```

    )

    [12] => Array
    (
        [idnumber] => 13
        [firstname] => Samantha
        [lastname] => Weather
        [age] => 37
    )

    [13] => Array
    (
        [idnumber] => 14
        [firstname] => Karl
        [lastname] => Saunders
        [age] => 34
    )

    [14] => Array
    (
        [idnumber] => 15
        [firstname] => Angela
        [lastname] => Brown
        [age] => 35
    )

)

```

Result:

| idnumber | firstname | lastname | age |
|----------|-----------|----------|-----|
| 1 | John | Smith | 23 |
| 2 | Sue | Davis | 13 |
| 3 | Peter | Young | 45 |
| 4 | Alice | Brown | 56 |
| 5 | Frank | Gray | 22 |
| 6 | Betty | Redwood | 27 |
| 7 | Jane | Corner | 29 |
| 8 | Bob | Church | 19 |
| 9 | Mary | Collins | 67 |
| 10 | Herbert | River | 42 |
| 11 | Lucy | Hardy | 13 |
| 12 | Charles | Winter | 81 |
| 13 | Samantha | Weather | 37 |
| 14 | Karl | Saunders | 34 |
| 15 | Angela | Brown | 35 |

The next thing I had to do is to insert a function to **dbutils.php** using the following code:

```

function htmlTable( $pdo, $table) {
    $sql = "DESCRIBE `".$table."`";
    $stmt = $pdo->query( $sql);
    print "<table>";
}

```

```

print "<tr>";
foreach( $stmt as $v) {
    print "<th>".$v['Field']."</th>";
}
print "</tr>";

$sql = "SELECT * FROM `".$table."`";
// specify only an associative array to be returned
$stmt = $pdo->query( $sql, PDO::FETCH_ASSOC);
foreach( $stmt as $row) {
    print "<tr>";
    foreach( $row as $v) {
        print "<td>".$v."</td>";
    }
    print "</tr>";
}
print "</table>";
}

```

Another thing is to add a style to **dbtest.php** to make a coloured table using the following code:

```

<style>
table {
    border-collapse: collapse;
}
tr:nth-of-type(odd) {
    background: #ffff99;
}
tr:nth-of-type(even) {
    background: #dddd88;
}
th {
    background: #ffffff;
    border-bottom: solid 2px black;
    text-align: left;
    padding: 4px 16px;
}
td {
    padding: 4px 16px;
}
</style>

```

The final thing was to call the function **htmltable**, so it prints out the table using the following code:

```

print"<h2>Result:</h2>";htmlTable( $pdo, 'demotable');

```

I managed to complete the exercise. The exercise was easy, but uploading the file to the server was difficult as I was getting an error and the web page would not load I figured out the problem was caused by an error which was `$mysqlhost = 'meno.lsbu.ac.uk'`; I had to change it to `$mysqlhost = 'localhost'`; to make the web page work. The important thing I have learnt is web pages can be uploaded to MySQL and can be viewed as a webpage. I have also learnt that there are a number of ways of uploading web pages on to different servers. I think the exercise is important and helped me understand how MySQL works and the different servers for uploading web pages.

The exercises helped me realise that by having an understanding of what the different servers can do enables you to understand that there are different ways of uploading webpages to the server. I did not do any extended tasks.

The screenshot below shows how the table description demo table matches the PHP code in **htmlTable**. By going on to putty and using the command line typing `$ mysql -u lamt3 -p`

lamt3 this will show up what tables are in the database on **meno**. For MySQL it matches the code as it uses a connect function to connect on **dbutils.php** and it also links to another PHP code called **dbtest.php** which on the **dbtest.php**. It prints out the **htmlTable** and uses the function **createDemotable**.

dbtest.php

```
<?php
    require_once( 'lib/dbutils.php');
?>
<!DOCTYPE html>
<html>
<head>
<title>DB Test</title>
<style>
    table {
        border-collapse: collapse;
    }
    tr:nth-of-type(odd) {
        background: #ffff99;
    }
    tr:nth-of-type(even) {
        background: #ddd888;
    }
    th {
        background: #ffffff;
        border-bottom: solid 2px black;
        text-align: left;
        padding: 4px 16px;
    }
    td {
        padding: 4px 16px;
    }
</style>

</head>
<body>

<?php

$pdo = connect();

if ( $pdo) {
    print "<h1>Successfully connected!</h1>\n";
} else {
    print "<h1>Did not connect.</h1>\n";
}

createDemoTable( $pdo);
print "<pre>\n";
dumpTable( $pdo, 'demotable');
print "</pre>\n";

print"<h2>Result:</h2>";
htmlTable( $pdo, 'demotable');

?>

</body>
</html>

dbutils.php
<?php

function connect() {
    $username = 'lamt3';
    $password = 'boiltow_';
    $mysqlhost = 'localhost';
    $dbname = $username;
    $pdo = new PDO('mysql:host='.$mysqlhost.';dbname='.$dbname.';charset=utf8',
$username, $password);
    if ( $pdo) {
        // make errors throw exceptions
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    }
}
```

```

        return $pdo;
    } else {
        die("Could not create PDO connection.");
    }
}

function tableExists( $pdo, $table) {
    $sql = "SHOW TABLES LIKE '$table'";
    return ( $pdo->query( $sql)->rowCount() > 0);
}

function createDemoTable( $pdo) {
    if ( !tableExists( $pdo, 'demotable')) {
        $sql = "CREATE TABLE IF NOT EXISTS `demotable` (
            `idnumber` int(11) NOT NULL auto_increment,
            `firstname` varchar(255) NOT NULL,
            `lastname` varchar(255) NOT NULL,
            `age` int(11) NOT NULL,
            PRIMARY KEY (`idnumber`)
        )";
        $pdo->exec( $sql);
        $sql = "INSERT INTO `demotable`
            (`firstname`, `lastname`, `age`)
        VALUES
            ('John','Smith',23),
            ('Sue','Davis',13),
            ('Peter','Young',45),
            ('Alice','Brown',56),
            ('Frank','Gray',22),
            ('Betty','Redwood',27),
            ('Jane','Corner',29),
            ('Bob','Church',19),
            ('Mary','Collins',67),
            ('Herbert','River',42),
            ('Lucy','Hardy',13),
            ('Charles','Winter',81),
            ('Samantha','Weather',37),
            ('Karl','Saunders',34),
            ('Angela','Brown',35)";
        $pdo->exec( $sql);
    }
}

function dumpTable( $pdo, $table) {
    $sql = "SELECT * FROM `".$table."`";
    $stmt = $pdo->query( $sql);
    // get all rows in an array
    $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
    print_r( $results);
}

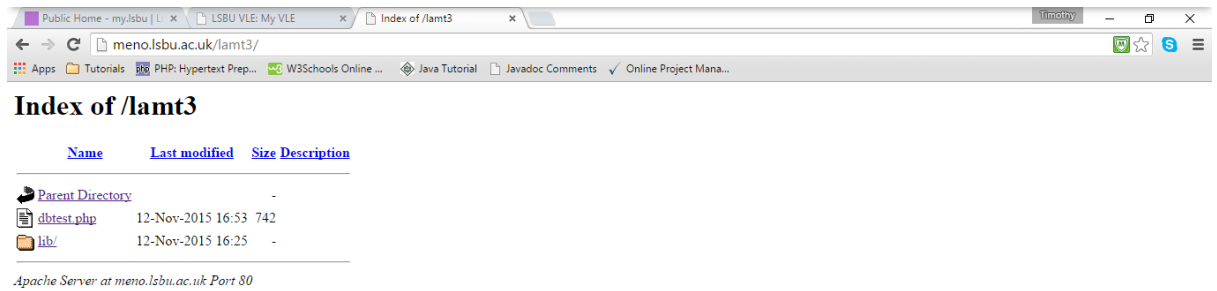
function htmlTable( $pdo, $table) {
    $sql = "DESCRIBE `".$table."`";
    $stmt = $pdo->query( $sql);
    print "<table>";
    print "<tr>";
    foreach( $stmt as $v) {
        print "<th>".$v['Field']. "</th>";
    }
    print "</tr>";

    $sql = "SELECT * FROM `".$table."`";
    // specify only an associative array to be returned
    $stmt = $pdo->query( $sql, PDO::FETCH_ASSOC);
    foreach( $stmt as $row) {
        print "<tr>";
        foreach( $row as $v) {
            print "<td>".$v. "</td>";
        }
        print "</tr>";
    }
    print "</table>";
}
?>

```

```
lamt3@meno:~  
login as: lamt3  
Warning : Unauthorised Access Prohibited  
  
lamt3@meno.lsbu.ac.uk's password:  
Last login: Fri Nov 13 00:12:36 2015 from 90.209.122.17  
#####  
#####  
Unauthorised access is prohibited  
  
IMPORTANT JULY 2015  
  
PHP has been updated from 5.1.6 to 5.6.4  
  
#####  
  
Disk quotas for user lamt3 (uid 142718):  
    Filesystem blocks   quota   limit   grace   files   quota   limit   grace  
    /dev/sdb1    120 100000 120000         20    4000    7500  
[lamt3@meno ~]$
```

```
lamt3@meno:~  
login as: lamt3  
Warning : Unauthorised Access Prohibited  
  
lamt3@meno.lsbu.ac.uk's password:  
Last login: Fri Nov 13 00:12:36 2015 from 90.209.122.17  
#####  
#####  
Unauthorised access is prohibited  
  
IMPORTANT JULY 2015  
  
PHP has been updated from 5.1.6 to 5.6.4  
  
#####  
  
Disk quotas for user lamt3 (uid 142718):  
    Filesystem blocks   quota   limit   grace   files   quota   limit   grace  
    /dev/sdb1    120 100000 120000         20    4000    7500  
[lamt3@meno ~]$ mysql -u lamt3 -p lamt3
```

2.2. Week 8 – PHP and MySQL

2.2.1. Exercise 2 PHPMYAdmin

Books.php: <http://daydream.lsbu.ac.uk/myweb/lamt3/books.php>

The Exercise was **Exercise 2: PHPMYAdmin** I had to log on to the website <https://daydream.lsbu.ac.uk/phpmyadmin/> and create a table called **books** and follow the specification that was given In addition I had to insert a function in **dbutils.php**:

```
function importTable( $pdo, $table, $file) {
    $data = file( $file, FILE_IGNORE_NEW_LINES);
    $numLines = count( $data);
    $columns = explode("\t",$data[0]);
    for( $i = 1; $i < $numLines; $i++) {
        $values = explode("\t", $data[$i]);
        $sql = "INSERT INTO ` $table ` SET ";
        for( $j = 0; $j < count( $columns); $j++) {
            if ( $j > 0) {
                $sql .= ", ";
            }
            $sql .= "`" . $columns[$j] . "`=";
            // permit specifying NULL values with the word NULL
            if ( $values[$j] == 'NULL') {
                $sql .= "NULL";
            } else {
                $sql .= "'" . $values[$j] . "'";
            }
        }
        $pdo->exec( $sql);
    }
    print ($numLines - 1)." rows imported into table $table.";
}
```

This will allow you to load an empty text data into database tables. The next thing I had to do is copy the following text and save the file called **books.txt** and upload the file **books.txt** to the server same place as the file **dbtest.php**

| bookid | title | publisher | price | format | instock | description | image |
|--------|-------|-----------|-------|--------|---------|-------------|-------|
|--------|-------|-----------|-------|--------|---------|-------------|-------|

| | | | | | | |
|---|-------------------------------------|---------------------|-------|-----------|---|---|
| 1 | War and Peace | New English Library | 12.99 | paperback | 1 | War and Peace is a vast epic centred on Napoleon's war with Russia. While it expresses Tolstoy's view that history is an inexorable process which man cannot influence, he peoples his great novel with a cast of over five hundred characters. Three of these, the artless and delightful Natasha Rostov, the world-weary Prince Andrew Bolkonsky and the idealistic Pierre Bezukhov illustrate Tolstoy's philosophy in this novel of unquestioned mastery. This translation is one which received Tolstoy's approval. warandpeace.jpg |
| 2 | Anna Karenina | New English Library | 11.99 | paperback | 1 | Anna Karenina is one of the most loved and memorable heroines of literature. Her overwhelming charm dominates a novel of unparalleled richness and density. Tolstoy considered this book to be his first real attempt at a novel form, and it addresses the very nature of society at all levels,- of destiny, death, human relationships and the irreconcilable contradictions of existence. It ends tragically, and there is much that evokes despair, yet set beside this is an abounding joy in life's many ephemeral pleasures, and a profusion of comic relief. annakarenina.jpg |
| 3 | Doctor Who and the Pyramids of Mars | Target | 7.99 | paperback | 1 | The mind of Sutekh the Destroyer is consumed with jealousy and hatred. Convinced that all living things are his mortal enemy, he is determined to annihilate all forms of life throughout the universe. Imprisoned at the heart of an Egyptian pyramid, the force of this maniacal evil has been paralysed for centuries. But now, after thousands upon thousands of years of long captivity, the moment of deliverance has arrived. Sutekh's vicious megalomania is about to be unleashed upon the world - unless the Doctor succeeds in outwitting a mind so powerful it can force him to his knees and torture him at a glance... drwho.jpg |

What is more I had to insert another code in **dbtest**:

```
importTable( $pdo, 'books', 'books.txt');
htmlTable( $pdo, 'books');
```

The next thing I had to do is to go on to **PHPmyAdmin** and insert a function to the table and add the following data to the table:

| | |
|-------------|---------------------------------|
| bookid | leave blank |
| title | Damiano |
| publisher | Bantam Books |
| price | 5.99 |
| format | paperback |
| instock | 0 |
| description | The story of Damiano Dalstrego. |
| image | damiano.jpg |

What is more, I had to insert the following code and save the file as **books.php**

```
<?php
require_once "lib/dbutils.php";
?>
<html>
<body>
<h2 align="center">Books</h2>
<table border="1" width="100%">
  <tr>
    <td width="10%" align="center"><big><b>Book id </b></big></td>
    <td width="30%" align="center"><big><b>Title</b></big></td>
    <td width="20%" align="center"><big><b>Publisher</b></big></td>
    <td width="20%" align="center"><big><b>Price</b></big></td>
    <td width="20%" align="center"><big><b>In stock</b></big></td>
  </tr>
</table>
<?php
  $pdo = connect();
  $results = $pdo->query("SELECT * FROM books", PDO::FETCH_ASSOC);
  foreach( $results as $row) {
    print("<tr>");
    print("<td>".$row['bookid']. "</td>");
    print("<td>".$row['title']. "</td>");
```

```

        print("<td>".$row['publisher']."</td>");
        print("<td>".$row['price']."</td>");
        print("<td>".$row['instock']."</td>");
        print("</tr>");
    };
?>
</table>
</body>
</html>

```

Furthermore, I had to insert the following code and save the file as **bookview.php**:

```

<?php
require_once "lib/dbutils.php";
?>
<html>
<body>
<?php
$pdo = connect();
$results = $pdo->query("SELECT * FROM books WHERE bookid=".$_REQUEST['bookid']);
foreach( $results as $row) {
    print("<tr>");
    print("<h1>".$row['title']."</h1>");
    print("<table><tr>");
    print("<td valign='top'><img src=''></td>");
    print("<td>".$row['description']."</td>");
    print("</tr></table>");
    print("<ul>");
    print("<li>Publisher: ".$row['publisher']."</li>");
    if ( $row['instock'] == 1) {
        print("<li>In stock: Yes</li>");
    } else {
        print("<li>In stock: No</li>");
    }
    print("</ul>");
    print("<p><big><strong>PRICE: &pound;".$row['price']."</strong></big></p>");
}
?>
<p><a href="books.php">Back to list</a></p>
</body>
</html>

```

Also I had to carry out the following task to make the **books.php** and **bookview.php**

- **Make the list in books.php; print out Yes or No dependent on whether the instock value is 1 or 0 - look at the code in bookview.php to see how to achieve this.**
- **Make the title of the book listed in books.php become a link to the bookview.php script which has the correct bookid number to display the full details about the book.**
- **Download the images in the archive on moodle and save them in your existing images folder. Then fix bookview.php so that it displays those images correctly.**
- **Replace the unnecessary book id column in the books.php list with a column which shows a small fixed-size image of the book which can be clicked to go to the details page.**

I did not manage to complete the whole exercise as I had missed out one part of the exercise which was:

```

importTable( $pdo, 'books', 'books.txt');
htmlTable( $pdo, 'books');

```

In **Exercise 2: PHPMyAdmin** I got up to where I had to carry out the following task for **books.php** and **bookview.php**, I did not complete the whole exercises for **Exercise 2: PHPMyAdmin** as I had some problems. The first thing I encountered is the error that I was getting when I used the following code:

```
importTable( $pdo, 'books', 'books.txt');  
htmlTable( $pdo, 'books');
```

The other task I did not complete was, **books.php** and **bookview.php**. The exercise was hard as I was getting an error and could not solve it. The exercise was difficult as I did not know what was causing the error when using the

```
importTable( $pdo, 'books', 'books.txt');  
htmlTable( $pdo, 'books');
```

I have learnt in **Exercise 2: PHPMyAdmin** how to create a table without using PHP code. In addition, this exercise showed me how to import text files to a table using PHP code. What is more, it showed me how to use **PHPMyAdmin** to insert data without using PHP Code. I think this exercise fits into the bigger picture as by knowing how a MySQL database and PHP work when creating and managing the website, it also allows you to understand that when creating a website using MySQL and PHP you would be able to view it on the website **PHPMyAdmin**. The exercise helped me understand how MySQL works. I did not do any extended task for **Exercise 2: Starting PHPMyAdmin**

2.3. Week 9 – PHP and MySQL

The Exercises was **Exercise 2: PHPMyAdmin**, **Exercise 3: Photo Album Server Version B and C**, **Exercise 4: Photo Album Server Version D**, **Exercise 5: PHP Session and authentication** and **Exercise 6: Photo Album Server Version E**

2.3.1. Exercise 2: PHPMyAdmin

Books.php: <http://daydream.lsbu.ac.uk/myweb/lamt3/books.php>

In **Exercise 2: PHPMyAdmin** I managed to fix some errors I was getting which was that I missed out a piece of important code:

```
importTable( $pdo, 'books', 'books.txt');  
htmlTable( $pdo, 'books');
```

There was an error in the document which produces an error on the webpage. The error was the apostrophes. This caused the website to produce an error. To solve this error I had to remove all the apostrophes in the text document books.txt. This solved the problem which imported the **books.txt** to the database on **PHPMyAdmin**, and this would allow me to complete the next task which was to disable the importable

```
//importTable( $pdo, 'books', 'books.txt');
```

What is more I added the following data to the database books in **PHPMyAdmin** using the following data:

| | |
|-----------|--------------|
| Bookid | leave blank |
| Title | Damiano |
| Publisher | Bantam Books |
| Price | 5.99 |
| Format | Paperback |

| | |
|-------------|---------------------------------|
| Instock | 0 |
| Description | The story of Damiano Dalstrego. |
| Image | damiano.jpg |

The final thing I had to do was to solve the task that was given which was to make the **book.php** link to **bookview.php** using the following task

- **Make the list in books.php print out Yes or No dependent on whether the instock value is 1 or 0 - look at the code in bookview.php to see how to achieve this.**
- **Make the title of the book listed in books.php be a link to the bookview.php script which has the correct bookid number to display the full details about the book.**
- **Download the images in the archive on moodle and save them in your existing images folder. Then fix bookview.php so that it displays those images correctly.**
- **Replace the unnecessary book id column in the books.php list with a column which shows a small fixed-size image of the book which can be clicked to go to the details page.**

I managed to complete the whole exercise and solve the task that was given to me.

This is the fully working code for **books.php** which links to **bookview.php**

Books.php

```
<?php
require_once "lib/dbutils.php";
?>
<html>
<body>
<h2 align="center">Books</h2>
<table border="1" width="100%">
  <tr>
    <td width="10%" align="center"><big><b>Book id </b></big></td>
    <td width="30%" align="center"><big><b>Title</b></big></td>
    <td width="20%" align="center"><big><b>Publisher</b></big></td>
    <td width="20%" align="center"><big><b>Price</b></big></td>
    <td width="20%" align="center"><big><b>In stock</b></big></td>
  </tr>
</table>
<?php
$pdo = connect();
$results = $pdo->query("SELECT * FROM books", PDO::FETCH_ASSOC);
foreach( $results as $row) {
  print("<tr>");
  print("<td><img
src='images/'. $row['image']. 'height=30%'>". $row['bookid']. "</td>");
  print("<td><a
href='bookview.php?bookid=". $row['bookid']. "'>". $row['title']. "</a></td>");
  print("<td>". $row['title']. "</td>");
  print("<td>". $row['publisher']. "</td>");
  print("<td>". $row['price']. "</td>");
  print("<td>". $row['instock']. "</td>");
  print("</tr>");
};
?>
</table>
</body>
</html>
```

bookview.php

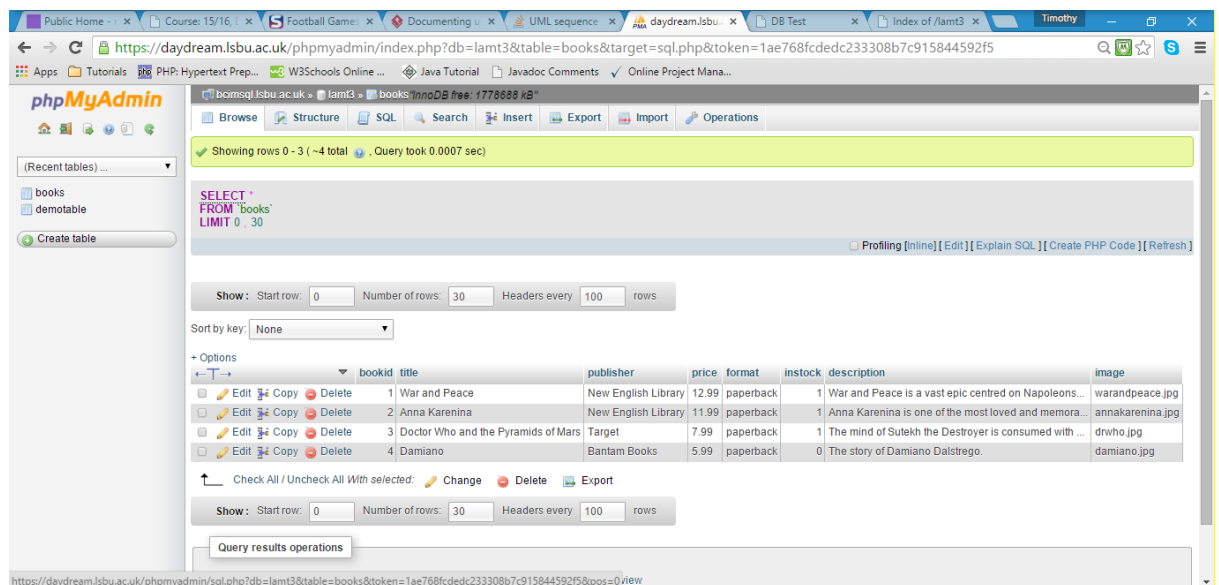
```
<?php
require_once ('lib/dbutils.php');
?>
```

```

<html>
<body>
<?php
$pdo = connect();
$results = $pdo->query("SELECT * FROM books WHERE bookid=".$_REQUEST['bookid']);
foreach( $results as $row) {
    print("<tr>");
    print("<h1>".$row['title']."</h1>");
    print("<table><tr>");
    print("<td valign='top'><img src='images/'".$row['image']."'></td>");
    print("<td>".$row['description']."</td>");
    print("</tr></table>");
    print("<ul>");
    print("<li>Publisher: ".$row['publisher']."</li>");
    if ( $row['instock'] == 1) {
        print("<li>In stock: Yes</li>");
    } else {
        print("<li>In stock: No</li>");
    }
    print("</ul>");
    print("<p><big><strong>PRICE: &pound;".$row['price']."</strong></big></p>");
}
?>
<p><a href="books.php">Back to list</a></p>
</body>
</html>

```

This is a screenshot of books table on **PHPMYAdmin** where it shows the books that are currently showing on the database



2.3.2. Extended Task: Exercise 2- PHPMyAdmin

The advantages of PHPMyAdmin is that it also showed how the database can be updated and deleted using MySQL and how MySQL is useful for showing if there is an item in stock or not. In addition, another advantage is that it you can also manage websites using MySQL by logging in to **PHPMYAdmin** and managing the database.

2.3.3. Exercise 3 Photo Album Server Version B and C

In Exercise 3: Photo Album Server Version B and C I had to download a folder called **photographs-db.zip** from moodle and extract the folder and upload two files containing **create-photograph-db.php** and **photographs.php** from the folder and upload both files to the server and view the file **create-photograph-db.php** on the webpage. This will create a table automatically on to the database **PHPMYadmin**. In addition I had to login to

PHPMyAdmin to check if the table has been created. What is more I had to create a line in **dbtest** which is:

```
htmlTable($pdo, 'photographs');
```

2.3.3.1. Photo Album Sever Version B

Photo Album Server Version B:

<http://daydream.lsbu.ac.uk/myweb/lamt3/photoalbum-server-version-b.php>

The next thing I had to do is to make a copy of **Photo Album Server Version A** and call it **Photo Album Server Version B** and add *require once* including statements at the top like **dbutils**.

```
<?php
require_once( 'lib/dbutils.php');
?>
```

Furthermore I had to delete all the lines containing coded array and replace the code to a query database as shown below:

```
$john['image'] = "b1.png";
$john['name'] = "John";
$john['description'] = "<p>This is John Lennon. He was one of the Beatles.</p><p>He was one of the two main song writers.</p>";

$paull['image'] = "b2.png";
$paull['name'] = "Paul";
$paull['description'] = "<p>This is Paul McCartney. He was also one of the Beatles.</p><p>He was the other main song writer.</p>";

$george['image'] = "b3.png";
$george['name'] = "George";
$george['description'] = "<p>This is George Harrison. He was also one of the Beatles.</p>";

$ringo['image'] = "b4.png";
$ringo['name'] = "Ringo";
$ringo['description'] = "<p>This is Ringo Starr. He was also one of the Beatles.</p>";

$rows[0] = $john;
$rows[1] = $paull;
$rows[2] = $george;
$rows[3] = $ringo;

$pdo = connect();

$sql = "SELECT * FROM `photographs` ORDER BY `photoid`";
$stmt = $pdo->query( $sql);
// get all rows in an array
$rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

The final things was to view **Photo Album Server Version B** in a web browser. This is the fully working code for **Photo Album Server Version B**:

```
<!DOCTYPE html>
<?php
require_once( 'lib/dbutils.php');
?>
<html>
<head>
<title>HTML Photo Album</title>
<meta name='viewport' content='width=device-width, initial-scale=1'>
<link rel='stylesheet' href='photoalbum.css'>
```

```

<link rel='stylesheet' href='http://www.w3schools.com/lib/w3.css'>
<script src='http://code.jquery.com/jquery-1.11.3.min.js'></script>
<link rel='stylesheet' href='lib/lightbox2/css/lightbox.css'>
</head>
<body>

<header class='w3-container w3-orange w3-center'>
  <h1>My Photo Album</h1>
  <h2>Your Name</h2>
  <p>Here are some photographs and descriptions.</p>
</header>

<?php

$pdo = connect();

$sql = "SELECT * FROM `photographs` ORDER BY `photoid`";
$stmt = $pdo->query( $sql);
// get all rows in an array
$rows = $stmt->fetchAll(PDO::FETCH_ASSOC);

$numberOfPhotos = count( $rows);

for( $i = 0; $i < $numberOfPhotos; $i++) {

  if ( $i % 3 == 0 ) {
    if ( $i > 0 ) { print "</div>\n"; }
    print "<div class='w3-row'>\n";
  }

  $row = $rows[$i];
  print "    <div class='w3-container w3-row w3-col 14'>\n";
  print "        <div class='w3-container w3-card-8 w3-margin-8'>\n";
  print "            <div class='w3-container w3-half w3-image'>\n";
  print "                <a class='example-image-link' data-lightbox='example-1'
href='images/" . $row['image'] . "'><img src='images/" . $row['image'] . "' class='example-
image w3-circle'></a>\n";
  print "                    <div class='w3-title w3-text-purple'>". $row['name'] . "</div>\n";
  print "                        </div>\n";
  print "                            <div class='w3-container w3-half'>\n";
  print $row['description'] . "\n";
  print "                                </div>\n";
  print "                                    </div>\n";
  print "                                        </div>\n";
}

print "</div>\n";

?>
<footer class='w3-container w3-orange w3-padding-jumbo'>
  <div class='w3-container w3-center'>
    <a href='mailto:childm@lsbu.ac.uk' class='link'>Contact me</a>
  </div>
</footer>
<script src='lib/lightbox2/js/jquery.js'></script>
<script src='lib/lightbox2/js/lightbox.js'></script>
</body>
</html>

```

2.3.3.2. Photo Album Sever Version C

Photo Album Server Version C:

<http://daydream.lsbu.ac.uk/myweb/lamt3/photoalbum-server-version-c.php>

The next task of the Exercises for **Exercise 3: Photo Album Server Version B and C** is deleting a picture from the album. The first thing I had to do was to create a new file called **photoalbum-common.php** and copy the following code into the file:

```
<?php

function deletePhotograph( $pdo, $deletionid) {
    $errorMessage = "";
    // retrieve name of image file so we can delete it
    $stmt = $pdo->prepare("SELECT `image` FROM `photographs` WHERE `photoid`=?" );
    $stmt->execute( array( $deletionid));
    $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
    if ( count( $rows) == 1) {
        // delete file
        unlink( "images/".$rows[0]['image']);
        // delete database record
        $stmt = $pdo->prepare("DELETE FROM `photographs` WHERE `photoid`=?" );
        $stmt->execute( array( $deletionid));
        $affected_rows = $stmt->rowCount();
    } else if (count( $rows) > 1) {
        $errorMessage .= "ID matches more than one record. ";
    } else {
        $errorMessage .= "ID not found: nothing to delete. ";
    }
    return $errorMessage;
}

?>
```

The next thing I had to do was to upload the file **photoalbum-common.php** to the server into the main folder and not the lib folder. In addition I had to save a copy of **photo-album-server-version-b** and call it **photo-album-server-version-c**. Additionally, I had to add the following line to **dbutils require_once** command to include the **photoalbum-common.php** file using the following code:

```
require_once("photoalbum-common.php");
```

This will allow the function to be available for the file **photo-album-server-version-c.php** Furthermore I had to insert the following code to **photo-album-server-version-c.php** after the **\$pdo = connect()** line:

```
deletePhotograph( $pdo, 1);
```

The next thing I had to do is replace the call of **deletphotograaph** and add the following code

```
if ( isset( $_GET['deletionid'])) {
    $errorMessage = deletePhotograph( $pdo, $_GET['deletionid']);
    if ( $errorMessage != "" ) {
        print "<div class='errorMessage'>$errorMessage</div>\n";
    } else {
        print "<div class='message'>Image deleted.</div>\n";
    }
}
```

The next thing I had to do is to insert the following code:

```
print " <div class='w3-container w3-row w3-col 14'>\n";
print " <div class='w3-container w3-card-8 w3-margin-8'>\n";
print " <div class='w3-container w3-half w3-image'>\n";
```

```

    print "          <a class='example-image-link' data-lightbox='example-1'
href='images/' . $row['image'] . "'><img src='images/' . $row['image'] . "' class='example-
image w3-circle'></a>\n";
    print "          <div class='w3-title w3-text-purple'>". $row['name'] . "</div>\n";
    print "          </div>\n";
    print "          <div class='w3-container w3-half'>\n";
    print $row['description'] . "\n";
    print "          <a href='?deletionid=" . $row['photoid'] . "' class='link'>(Delete)</a>";
    print "          </div>\n";
    print "          </div>\n";
    print "        </div>\n";

```

This will allow to every image to have a delete link. This is the full code of **photo album server vserion c**:

```

<!DOCTYPE html>
<?php
require_once( 'lib/dbutils.php' );
require_once("photoalbum-common.php");
?>
<html>
<head>
<title>HTML Photo Album</title>
<meta name='viewport' content='width=device-width, initial-scale=1'>
<link rel='stylesheet' href='photoalbum.css'>
<link rel='stylesheet' href='http://www.w3schools.com/lib/w3.css'>
<script src='http://code.jquery.com/jquery-1.11.3.min.js'></script>
<link rel='stylesheet' href='lib/lightbox2/css/lightbox.css'>
</head>
<body>

<header class='w3-container w3-orange w3-center'>
  <h1>My Photo Album</h1>
  <h2>Your Name</h2>
  <p>Here are some photographs and descriptions.</p>
</header>

<?php

$pdo = connect();

if ( isset( $_GET['deletionid'] ) ) {
  $errorMessage = deletePhotograph( $pdo, $_GET['deletionid'] );
  if ( $errorMessage != "" ) {
    print "<div class='errorMessage'>$errorMessage</div>\n";
  } else {
    print "<div class='message'>Image deleted.</div>\n";
  }
}

$sql = "SELECT * FROM `photographs` ORDER BY `photoid`";
$stmt = $pdo->query( $sql );
// get all rows in an array
$rows = $stmt->fetchAll(PDO::FETCH_ASSOC);

$numberOfPhotos = count( $rows );

for( $i = 0; $i < $numberOfPhotos; $i++ ) {

  if ( $i % 3 == 0 ) {
    if ( $i > 0 ) { print "</div>\n"; }
    print "<div class='w3-row'>\n";
  }

  $row = $rows[$i];
  print "    <div class='w3-container w3-row w3-col 14'>\n";
  print "      <div class='w3-container w3-card-8 w3-margin-8'>\n";
  print "        <div class='w3-container w3-half w3-image'>\n";
  print "          <a class='example-image-link' data-lightbox='example-1'
href='images/' . $row['image'] . "'><img src='images/' . $row['image'] . "' class='example-
image w3-circle'></a>\n";
  print "          <div class='w3-title w3-text-purple'>". $row['name'] . "</div>\n";
  print "          </div>\n";

```

```

        print "        <div class='w3-container w3-half'>\n";
        print $row['description']."\n";
        print "        <a href='?deletionid=".$row['photoid']."' class='link'>(Delete)</a>";
        print "        </div>\n";
        print "    </div>\n";
        print " </div>\n";
    }

    print "</div>\n";

?>
<footer class='w3-container w3-orange w3-padding-jumbo'>
    <div class='w3-container w3-center'>
        <a href='mailto:childm@lsbu.ac.uk' class='link'>Contact me</a>
    </div>
</footer>
<script src='lib/lightbox2/js/jquery.js'></script>
<script src='lib/lightbox2/js/lightbox.js'></script>
</body>
</html>

```

2.3.4. Exercise 4: Photo Album Server Version D

Photoalbum Server Version D: <http://daydream.lsbu.ac.uk/myweb/lamt3/photoalbum-server-version-d.php>

The first thing I had to do for **Exercise 4: Photo Album Server Version D**, is save a copy of **Photo Album Server Version C** and call it **Photo Album Server Version D**. What is more I had to insert the following html code after the end of the **PHP tag ?>**

```

<div class='w3-container w3-center'>
    <a href="#id01" class='link'>Add images</a>
</div>

<div id="id01" class="w3-modal">
    <div class="w3-modal-dialog">
        <div class="w3-modal-content w3-card-4">
            <header class="w3-container w3-teal">
                <a href="#" class="w3-closebtn">&times;</a>
                <h2>Upload photograph</h2>
            </header>
            <div class="w3-container">
                <form action='?' method='post' enctype='multipart/form-data' id='photoform'
class='w3-container'>
                    <div class="w3-group">
                        <input type='file' name='uploadfile' class='w3-input'>
                        <label class="w3-label">Image file</label>
                    </div>
                    <div class="w3-group">
                        <input type='text' name='name' class='w3-input' required>
                        <label class="w3-label">Name</label>
                    </div>
                    <div class="w3-group">
                        <textarea name='description' class='w3-input' required></textarea>
                        <label class="w3-label">Description</label>
                    </div>
                    <input type='hidden' name='formid' value='fileupload'>
                    <input type='submit' value='Upload' id='uploadbutton' class='w3-btn'>
                </form>
            </div>
            <footer class="w3-container w3-teal">
                <p></p>
            </footer>
        </div>
    </div>
</div>

```

In addition I had to insert the following function to **photoalbum-common.php**

```

function addPhotograph( $pdo, $name, $description, $user, $fileinfo) {
    // create DB entry to get an ID
    $stmt = $pdo->prepare("INSERT INTO photographs ( `name`, `description`, `user`)
VALUES (?, ?, ?)");
    $stmt->execute( array( $name, $description, $user));
    $photoid = $pdo->lastInsertId();

    // receive the files, I only have one
    $tmp_name = $fileinfo["tmp_name"];
    // create final image name from unique DB id
    $name = $photoid."_".$fileinfo["name"];
    // update the database with the final image name
    $stmt = $pdo->prepare("UPDATE photographs SET `image`=? WHERE `photoid`=?");
    $stmt->execute( array( $name, $photoid));
    // move the uploaded file to the final location
    $result = move_uploaded_file($tmp_name, "images/$name");
    return $photoid;
}

```

What is more I had to insert a code in **photoalbum-server version-d.php** after the following code. This will select the photographs from the database

```

    } else {
        print "<div class='message'>Image deleted.</div>\n";
    }
}

if ( isset( $_POST['formid']) && $_POST['formid'] == 'fileupload') {
    $photoid = addPhotograph( $pdo, $_POST['name'], $_POST['description'], 'self',
$_FILES['uploadfile']);
}

$sql = "SELECT * FROM `photographs` ORDER BY `photoid`";
$stmt = $pdo->query( $sql);

```

This is the full code of **Photoalbum -server-version-d.php**

```

<!DOCTYPE html>
<?php
require_once( 'lib/dbutils.php');
require_once("photoalbum-common.php");
?>
<html>
<head>
<title>HTML Photo Album</title>
<meta name='viewport' content='width=device-width, initial-scale=1'>
<link rel='stylesheet' href='photoalbum.css'>
<link rel='stylesheet' href='http://www.w3schools.com/lib/w3.css'>
<script src='http://code.jquery.com/jquery-1.11.3.min.js'></script>
<link rel='stylesheet' href='lib/lightbox2/css/lightbox.css'>
</head>
<body>

<header class='w3-container w3-orange w3-center'>
    <h1>My Photo Album</h1>
    <h2>Your Name</h2>
    <p>Here are some photographs and descriptions.</p>
</header>

<?php

$pdo = connect();

if ( isset( $_GET['deletionid'])) {
    $errorMessage = deletePhotograph( $pdo, $_GET['deletionid']);
    if ( $errorMessage != "") {
        print "<div class='errormessage'>$errorMessage</div>\n";
    } else {
        print "<div class='message'>Image deleted.</div>\n";
    }
}

if ( isset( $_POST['formid']) && $_POST['formid'] == 'fileupload') {
    $photoid = addPhotograph( $pdo, $_POST['name'], $_POST['description'], 'self',
$_FILES['uploadfile']);
}

```



```

$sql = "SELECT * FROM `photographs` ORDER BY `photoid`";
$stmt = $pdo->query( $sql);
// get all rows in an array
$rows = $stmt->fetchAll(PDO::FETCH_ASSOC);

$numberOfPhotos = count( $rows);

for( $i = 0; $i < $numberOfPhotos; $i++) {

    if ( $i % 3 == 0) {
        if ( $i > 0) { print "</div>\n"; }
        print "<div class='w3-row'>\n";
    }

    $row = $rows[$i];
    print "    <div class='w3-container w3-row w3-col 14'>\n";
    print "        <div class='w3-container w3-card-8 w3-margin-8'>\n";
    print "            <div class='w3-container w3-half w3-image'>\n";
    print "                <a class='example-image-link' data-lightbox='example-1'
href='images/' . $row['image'] . "'><img src='images/' . $row['image'] . "' class='example-
image w3-circle'></a>\n";
    print "                <div class='w3-title w3-text-purple'>". $row['name'] . "</div>\n";
    print "            </div>\n";
    print "            <div class='w3-container w3-half'>\n";
    print "                $row['description'] . "\n";
    print "                <a href='?deletionid=" . $row['photoid'] . "' class='link'>(Delete)</a>";
    print "            </div>\n";
    print "        </div>\n";
}

print "</div>\n";

?>
<div class='w3-container w3-center'>
    <a href="#id01" class='link'>Add images</a>
</div>

<div id="id01" class="w3-modal">
    <div class="w3-modal-dialog">
        <div class="w3-modal-content w3-card-4">
            <header class="w3-container w3-teal">
                <a href="#" class="w3-closebtn">&times;</a>
                <h2>Upload photograph</h2>
            </header>
            <div class="w3-container">
                <form action='?' method='post' enctype='multipart/form-data' id='photoform'
class='w3-container'>
                    <div class="w3-group">
                        <input type='file' name='uploadfile' class='w3-input'>
                        <label class="w3-label">Image file</label>
                    </div>
                    <div class="w3-group">
                        <input type='text' name='name' class='w3-input' required>
                        <label class="w3-label">Name</label>
                    </div>
                    <div class="w3-group">
                        <textarea name='description' class='w3-input' required></textarea>
                        <label class="w3-label">Description</label>
                    </div>
                    <input type='hidden' name='formid' value='fileupload'>
                    <input type='submit' value='Upload' id='uploadbutton' class='w3-btn'>
                </form>
            </div>
            <footer class="w3-container w3-teal">
                <p></p>
            </footer>
        </div>
    </div>
</div>

<footer class='w3-container w3-orange w3-padding-jumbo'>
    <div class='w3-container w3-center'>

```

```

        <a href='mailto:childm@lsbu.ac.uk' class='link'>Contact me</a>
    </div>
</footer>
<script src='lib/lightbox2/js/jquery.js'></script>
<script src='lib/lightbox2/js/lightbox.js'></script>
</body>
</html>

```

2.3.5.Exercise 5: PHP Session and Authentication

Poetry Form: <http://daydream.lsbu.ac.uk/myweb/lamt3/poetry-form.html>

Poetry: <http://daydream.lsbu.ac.uk/myweb/lamt3/poetry.php>

Login: <http://daydream.lsbu.ac.uk/myweb/lamt3/login.php>

The exercise for **Exercise 5 PHP Session and Authentication** was to insert the following code and save it as **poetry-form.html**

```

<!DOCTYPE html>
<html>
<head>
<title>Poetry Form</title>
</head>
<body>
<form action='poetry.php' method='post'>
    First name <input type='text' name='firstname'>
    Last name <input type='text' name='lastname'>
    <input type='submit'>
</form>
</body>
</html>

```

Furthermore, I had to insert another code and call it **poetry.php**

```

<html>
<head>
<title>Poetry</title>
</head>
<body>
<?php

print "<h1>A Poem for ".$_POST['firstname']." ".$_POST['lastname']. "</h1>\n";

$nouns = array( "cat", "dog", "chimney", "floor", "bicycle", "cheese", "tree",
"cloud", "sun", "moon", "balloon");
$verbs = array( "ran", "cried", "laughed", "loved", "ate", "stole", "bounced",
"shone", "walked", "awoke");
$adj = array( "brown", "big", "small", "shy", "happy", "old", "young", "brave",
"wise", "ancient", "orange", "kind");
$adv = array( "quickly", "slowly", "gracefully", "carefully", "enthusiastically",
"foolishly", "forgetfully");

print "<p>";
for( $i = 0; $i < 4; $i++) {
    print "The ".$adj[ rand(0, count($adj)- 1)]." ".$nouns[ rand(0, count($nouns)-
1)]." ".$adv[ rand(0, count($adv)- 1)]." ".$verbs[ rand(0, count($verbs)-
1)]. "<br>\n";
}
print "</p>\n";

print "<p><a href='?'>Another</a></p>\n";
?>
</body>
</html>

```

What is more I had to upload both files **poetry-form.html** and **poetry.php** to the server. In addition I had to change the following code in **poetry.php**

```

session_start();

```

```

if ( isset($_POST['firstname'])) {
    $_SESSION['firstname'] = $_POST['firstname'];
}
if ( isset($_POST['lastname'])) {
    $_SESSION['lastname'] = $_POST['lastname'];
}

print "<h1>A Poem for " . $_SESSION['firstname'] . " " . $_SESSION['lastname'] . "</h1>\n";

```

This will allow the script to remember the user name when users click on another link. This will store it on the server in the special **\$_Session array**. In addition I had to insert the following code and save the file as **create-user-db.php** and upload it to the server and view it on the web browser. This will automatically create a table in **PHPMyAdmin**

```

<?php
require_once("lib/dbutils.php");

$pdo = connect();

if ( !tableExists( $pdo, 'users')) {
    print "<p>Creating users table...";
    $sql = "CREATE TABLE IF NOT EXISTS `users` (
        `user` varchar(255) NOT NULL,
        `password` varchar(255) NOT NULL,
        `role` enum( 'normal','admin') NOT NULL,
        PRIMARY KEY (`user`)
    )";

    $pdo->exec( $sql);
    print "<p>Inserting default admin user admin/password... ";

    $sql = "INSERT INTO users SET `user`='admin', `password`='password',
`role`='admin'";
    $pdo->exec( $sql);

}
print "<p>Done";
?>

```

The next thing I had do is add two functions in **dbutils**

```

function setUpSession() {
    session_start();
    if ( !isset( $_SESSION['user'])) {
        $_SESSION['user'] = "anonymous";
    }
    if ( !isset( $_SESSION['role'])) {
        $_SESSION['role'] = "anonymous";
    }
}

/*
Session variables are set to anonymous IF THEY HAVE NO VALUE or if
a log in attempt is made and the credentials fail, or if a logout
attempt is made. If nothing is attempted and values are set, they
stay the same.
*/
function loginOrOut( $pdo, $usersTable) {
    setUpSession();

    $action = "none";
    if ( isset( $_POST['formid']) && $_POST['formid'] == 'login') {
        $stmt = $pdo->prepare("SELECT * FROM `{$usersTable}` WHERE user=? AND
password=?");
        $stmt->execute( array( $_POST['username'], $_POST['password']));
        $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
        if ( count( $rows) == 1) {
            $_SESSION['user'] = $rows[0]['user'];
            $_SESSION['role'] = $rows[0]['role'];
            $action = "login success";
        } else if ( count( $rows) == 0) {
            $action = "login failure";
            $_SESSION['user'] = "anonymous";
            $_SESSION['role'] = "anonymous";
        }
    }
}

```

```

    } else {
        die( "System error: ".count( $rows)." rows in database for user
        ".$_POST['username']);
    }
    } else if ( isset( $_REQUEST['logout'])) {
        $_SESSION['user'] = "anonymous";
        $_SESSION['role'] = "anonymous";
        $action = "logout";
    }

    return $action;
}

```

After that I inserted the following code at the beginning of **dbutils**

```

$pdo = connect();
loginOrOut( $pdo, 'users');

```

The final thing I had to do is to insert the following code and save the file called **login.php**

```

<?php
require_once("lib/dbutils.php");
?>
<!DOCTYPE html>
<html>
<head>
<title>Log in demo</title>
</head>
<body>
<?php
$pdo = connect();
loginOrOut( $pdo, 'users');

if ( $_SESSION['user'] == 'anonymous') {
?>
<h1>Please log in to use this site</h1>
<form action='' method='post'>
    Username <input type='text' name='username'>
    Password <input type='password' name='password'>
    <input type='hidden' name='formid' value='login'>
    <input type='submit'>
</form>
<?php
} else {
    print "<h1>You are logged in as ".$_SESSION['user']. "</h1>\n";

    $nouns = array( "cat", "dog", "chimney", "floor", "bicycle", "cheese", "tree",
    "cloud", "sun", "moon", "balloon");
    $verbs = array( "ran", "cried", "laughed", "loved", "ate", "stole", "bounced",
    "shone", "walked", "awoke");
    $adj = array( "brown", "big", "small", "shy", "happy", "old", "young", "brave",
    "wise", "ancient", "orange", "kind");
    $adv = array( "quickly", "slowly", "gracefully", "carefully", "enthusiastically",
    "foolishly", "forgetfully");

    print "<p>";
    for( $i = 0; $i < 4; $i++) {
        print "The ".$adj[ rand(0, count($adj)- 1)]. " ".$nouns[ rand(0, count($nouns)-
        1)]. " ".$adv[ rand(0, count($adv)- 1)]. " ".$verbs[ rand(0, count($verbs)-
        1)]. "<br>\n";
    }
    print "</p>\n";

    print "<p><a href='?'>Another</a></p>\n";

    print "<p><a href='?logout=true'>Log out</a></p>\n";
}
?>
</body>
</html>

```

This is the full code for **Exercise 5: PHP Session and Authentication**:

Poetry-form.html

```
<!DOCTYPE html>
<html>
<head>
<title>Poetry Form</title>
</head>
<body>
<form action='poetry.php' method='post'>
  First name <input type='text' name='firstname'>
  Last name <input type='text' name='lastname'>
  <input type='submit'>
</form>
</body>
</html>
```

Poetry.php

```
<html>
<head>
<title>Poetry</title>
</head>
<body>
<?php

session_start();

if ( isset($_POST['firstname'])) {
  $_SESSION['firstname'] = $_POST['firstname'];
}
if ( isset($_POST['lastname'])) {
  $_SESSION['lastname'] = $_POST['lastname'];
}

print "<h1>A Poem for " . $_SESSION['firstname'] . " " . $_SESSION['lastname'] . "</h1>\n";

$nouns = array( "cat", "dog", "chimney", "floor", "bicycle", "cheese", "tree",
"cloud", "sun", "moon", "balloon");
$verbs = array( "ran", "cried", "laughed", "loved", "ate", "stole", "bounced",
"shone", "walked", "awoke");
$adj = array( "brown", "big", "small", "shy", "happy", "old", "young", "brave",
"wise", "ancient", "orange", "kind");
$adv = array( "quickly", "slowly", "gracefully", "carefully", "enthusiastically",
"foolishly", "forgetfully");

print "<p>";
for( $i = 0; $i < 4; $i++) {
  print "The " . $adj[ rand(0, count($adj)- 1)] . " " . $nouns[ rand(0, count($nouns)-
1)] . " " . $adv[ rand(0, count($adv)- 1)] . " " . $verbs[ rand(0, count($verbs)-
1)] . "<br>\n";
}
print "</p>\n";

print "<p><a href='?'>Another</a></p>\n";
?>
</body>
</html>
```

Login.php

```
<?php
require_once("lib/dbutils.php");
?>
<!DOCTYPE html>
<html>
<head>
<title>Log in demo</title>
</head>
<body>
<?php
$pdo = connect();
loginOrOut( $pdo, 'users');

if ( $_SESSION['user'] == 'anonymous') {
```

```

?>
<h1>Please log in to use this site</h1>
<form action='' method='post'>
  Username <input type='text' name='username'>
  Password <input type='password' name='password'>
  <input type='hidden' name='formid' value='login'>
  <input type='submit'>
</form>
<?php
} else {
  print "<h1>You are logged in as ".$_SESSION['user']. "</h1>\n";

  $nouns = array( "cat", "dog", "chimney", "floor", "bicycle", "cheese", "tree",
"cloud", "sun", "moon", "balloon");
  $verbs = array( "ran", "cried", "laughed", "loved", "ate", "stole", "bounced",
"shone", "walked", "awoke");
  $adj = array( "brown", "big", "small", "shy", "happy", "old", "young", "brave",
"wise", "ancient", "orange", "kind");
  $adv = array( "quickly", "slowly", "gracefully", "carefully", "enthusiastically",
"foolishly", "forgetfully");

  print "<p>";
  for( $i = 0; $i < 4; $i++) {
    print "The ".$adj[ rand(0, count($adj)- 1)]. " ".$nouns[ rand(0, count($nouns)-
1)]. " ".$adv[ rand(0, count($adv)- 1)]. " ".$verbs[ rand(0, count($verbs)-
1)]. "<br>\n";
  }
  print "</p>\n";

  print "<p><a href='?'>Another</a></p>\n";

  print "<p><a href='?logout=true'>Log out</a></p>\n";
}
?>
</body>
</html>

```

2.3.6. Exercise 6: Photo Album Server Version E

Photo Album Server Version E: <http://daydream.lsbu.ac.uk/myweb/lamt3/photoalbum-server-version-e.php>

In **Exercise 6: Photo Album Server Version E** I had to save a copy of **photoalbum-server-version-d.php** and call it **photoalbum-server-version-e.php**. The next thing I had to do was to look at the code in **photoalbum-server-version-d.php**

```

if ( isset( $_GET['deletionid'])) {
  $errorMessage = deletePhotograph( $pdo, $_GET['deletionid']);
  if ( $errorMessage != "" ) {
    print "<div class='errorMessage'>$errorMessage</div>\n";
  } else {
    print "<div class='message'>Image deleted.</div>\n";
  }
}

if ( isset( $_POST['formid']) && $_POST['formid'] == 'fileupload') {
  $photoid = addPhotograph( $pdo, $_POST['name'], $_POST['description'], 'self',
$_FILES['uploadfile']);
}

and add a code loginorout and input the above code into the loginorout code:

loginOrOut( $pdo, 'users');

if ( $_SESSION['user'] != "anonymous") {

  ...code shown in box above goes here

}

```

This will check whether the user is logged in or not. What is more I had to upload **photoalbum-server-version-e** to the server and view it on a web browser.

Furthermore, I had to insert another code at the bottom of the script after the existing code for image upload just before the **<footer>** tag. This will make a link pop up

```
<div id='id02' class='w3-modal'>
  <div class='w3-modal-dialog'>
    <div class='w3-modal-content w3-card-4'>
      <header class='w3-container w3-teal'>
        <a href='#' class='w3-closebtn'>&times;</a>
        <h2>Log in</h2>
      </header>
      <div class='w3-container'>
        <form action='?' method='post' id='loginform' class='w3-container'>
          <div class='w3-group'>
            <input type='text' name='username' size='16' class='w3-input'>
            <label class='w3-label'>Username</label>
          </div>
          <div class='w3-group'>
            <input type='password' name='password' size='16' class='w3-input'>
            <label class='w3-label'>Password</label>
          </div>
          <input type='hidden' name='formid' value='login'>
          <input type='submit' value='Log in' id='loginbutton' class='w3-btn'>
        </form>
      </div>
      <footer class='w3-container w3-teal'>
        <p></p>
      </footer>
    </div>
  </div>
</div>
```

The next thing on the exercise is to add the following code at the top of the script after the **loginorout**. This will provide a link if the user is not already logged in, and a log out link if the user is already logged in

```
print "<div class='w3-container w3-center'>\n";
if ( $_SESSION['user'] == "anonymous") {
  print "  <a href='#id02' class='link'>Log in</a>\n";
} else {
  print "  User: " . $_SESSION['user'] . " Role: " . $_SESSION['role'] . " <a
href='?logout=true' class='link'>Log out</a>\n";
}
print "</div>\n";
```

This will allow the user to login to delete the photograph and if the user is not logged in the user will not be able to delete the photographs on the photo album.

This is the fully working code for **Photo Album Server Version E**:

```
<!DOCTYPE html>
<?php
require_once( 'lib/dbutils.php' );
require_once("photoalbum-common.php");
?>
<html>
<head>
<title>HTML Photo Album</title>
<meta name='viewport' content='width=device-width, initial-scale=1'>
<link rel='stylesheet' href='photoalbum.css'>
<link rel='stylesheet' href='http://www.w3schools.com/lib/w3.css'>
<script src='http://code.jquery.com/jquery-1.11.3.min.js'></script>
<link rel='stylesheet' href='lib/lightbox2/css/lightbox.css'>
</head>
<body>

<header class='w3-container w3-orange w3-center'>
```

```

<h1>My Photo Album</h1>
<h2>Your Name</h2>
<p>Here are some photographs and descriptions.</p>
</header>

<?php

$pdo = connect();
loginOrOut( $pdo, 'users');

if ( $_SESSION['user'] != "anonymous") {
    if ( isset( $_GET['deletionid'])) {
        $errorMessage = deletePhotograph( $pdo, $_GET['deletionid']);
        if ( $errorMessage != "" ) {
            print "<div class='errormessage'>$errorMessage</div>\n";
        } else {
            print "<div class='message'>Image deleted.</div>\n";
        }
    }

    if ( isset( $_POST['formid']) && $_POST['formid'] == 'fileupload') {
        $photoid = addPhotograph( $pdo, $_POST['name'], $_POST['description'],
'self', $_FILES['uploadfile']);
    }
}

print "<div class='w3-container w3-center'>\n";
if ( $_SESSION['user'] == "anonymous") {
    print "    <a href='#id02' class='link'>Log in</a>\n";
} else {
    print "    User: " . $_SESSION['user'] . " Role: " . $_SESSION['role'] . " <a
href='?logout=true' class='link'>Log out</a>\n";
}
print "</div>\n";

$sql = "SELECT * FROM `photographs` ORDER BY `photoid`";
$stmt = $pdo->query( $sql);
// get all rows in an array
$rows = $stmt->fetchAll(PDO::FETCH_ASSOC);

$numberOfPhotos = count( $rows);

for( $i = 0; $i < $numberOfPhotos; $i++) {

    if ( $i % 3 == 0) {
        if ( $i > 0) { print "</div>\n"; }
        print "<div class='w3-row'>\n";
    }

    $row = $rows[$i];
    print "    <div class='w3-container w3-row w3-col 14'>\n";
    print "        <div class='w3-container w3-card-8 w3-margin-8'>\n";
    print "            <div class='w3-container w3-half w3-image'>\n";
    print "                <a class='example-image-link' data-lightbox='example-1'
href='images/" . $row['image'] . "'><img src='images/" . $row['image'] . "' class='example-
image w3-circle'></a>\n";
    print "                <div class='w3-title w3-text-purple'>". $row['name'] . "</div>\n";
    print "            </div>\n";
    print "            <div class='w3-container w3-half'>\n";
    print $row['description'] . "\n";
    print "                <a href='?deletionid=" . $row['photoid'] . "' class='link'>(Delete)</a>\n";
    print "            </div>\n";
    print "        </div>\n";
}

print "</div>\n";

?>
<div class='w3-container w3-center'>
    <a href="#" id01" class='link'>Add images</a>
</div>

```



```

<div id="id01" class="w3-modal">
  <div class="w3-modal-dialog">
    <div class="w3-modal-content w3-card-4">
      <header class="w3-container w3-teal">
        <a href="#" class="w3-closebtn">&times;</a>
        <h2>Upload photograph</h2>
      </header>
      <div class="w3-container">
        <form action='?' method='post' enctype='multipart/form-data' id='photoform'
class='w3-container'>
          <div class="w3-group">
            <input type='file' name='uploadfile' class='w3-input'>
            <label class="w3-label">Image file</label>
          </div>
          <div class="w3-group">
            <input type='text' name='name' class='w3-input' required>
            <label class="w3-label">Name</label>
          </div>
          <div class="w3-group">
            <textarea name='description' class='w3-input' required></textarea>
            <label class="w3-label">Description</label>
          </div>
          <input type='hidden' name='formid' value='fileupload'>
          <input type='submit' value='Upload' id='uploadbutton' class='w3-btn'>
        </form>
      </div>
      <footer class="w3-container w3-teal">
        <p></p>
      </footer>
    </div>
  </div>
</div>
<div id='id02' class='w3-modal'>
  <div class='w3-modal-dialog'>
    <div class='w3-modal-content w3-card-4'>
      <header class='w3-container w3-teal'>
        <a href='#' class='w3-closebtn'>&times;</a>
        <h2>Log in</h2>
      </header>
      <div class='w3-container'>
        <form action='?' method='post' id='loginform' class='w3-container'>
          <div class='w3-group'>
            <input type='text' name='username' size='16' class='w3-input'>
            <label class='w3-label'>Username</label>
          </div>
          <div class='w3-group'>
            <input type='password' name='password' size='16' class='w3-input'>
            <label class='w3-label'>Password</label>
          </div>
          <input type='hidden' name='formid' value='login'>
          <input type='submit' value='Log in' id='loginbutton' class='w3-btn'>
        </form>
      </div>
      <footer class='w3-container w3-teal'>
        <p></p>
      </footer>
    </div>
  </div>
</div>

<footer class='w3-container w3-orange w3-padding-jumbo'>
  <div class='w3-container w3-center'>
    <a href='mailto:childm@lsbu.ac.uk' class='link'>Contact me</a>
  </div>
</footer>
<script src='lib/lightbox2/js/jquery.js'></script>
<script src='lib/lightbox2/js/lightbox.js'></script>
</body>
</html>

```

I managed to complete the whole exercise. I found all of the exercise easy; there were no difficulties, and I did not get stuck at any point.

I learned many things from doing this exercise. Firstly, I learnt how to create a login page. I also learnt how to create a delete link on **photoalbum-server-version-d.php**. In addition I also learnt how to create a login page on **photoalbum server version e**. This will allow the user to only delete the pictures on photo-album-server version-e I think these exercises fit into the bigger picture as they enable you to create a database. Also it will enable you to create a login using PHP and this will allow you to store details of which user is allowed to login and which user does not have permission to login.

In relation to Digital business Applications, I learnt that you can create a login page and store the username and password on the database. This will allow the user webpage to recognise the user once they are logged in. I also learnt that you can create a session using PHP. This will allow the user to login and have a certain amount of time before the webpage automatically logs out.

There were no extended task for Exercise 5: PHP session and authentication. In addition I did not do any of the extended tasks for the other exercises

2.4. Week 10 – JavaScript and Ajax

The Exercise was **Exercise 1: Photo Album Client Rendered Version A**

2.4.1. Exercise 1: Photo Album Client-Rendered Version A

Photoalbum Client Rendered Version A:

<http://daydream.lsbu.ac.uk/myweb/lamt3/photoalbum-client-rendered-version-a.html>

In **Exercise 1: Photo Album Client Rendered Version A** I had to create a file called **photoalbum-client-rendered-version-a.html** and insert the following code and also change the text in red to be my name and email address. This file **photoalbum-client-rendered-version-a.html** needed to be placed as the photo album script so that it can access the lightbox file and **photoalbum.css**:

```
<!DOCTYPE html>
<html>
<head>
<title>JSON Photo Album</title>
<meta name='viewport' content='width=device-width, initial-scale=1'>
<link rel='stylesheet' href='http://www.w3schools.com/lib/w3.css'>
<link rel='stylesheet' href='lib/lightbox2/css/lightbox.css'>
<link rel='stylesheet' href='photoalbum.css'>

<script src='http://code.jquery.com/jquery-1.11.3.min.js'></script>

</head>
<body>

<header class='w3-container w3-orange w3-center'>
  <h1>My Photo Album</h1>
  <h2>Your Name</h2>
  <p>Here are some photographs and descriptions.</p>
</header>

<div id='message'>

</div>

<div id='album'>

</div>

<footer class='w3-container w3-orange w3-padding-jumbo'>
  <div class='w3-container w3-center'>
    <a href='mailto:youremail@lsbu.ac.uk' class='link'>Contact me</a>
  </div>
```

```

</footer>

<script src='lib/lightbox2/js/lightbox.js'></script>
</body>
</html>

```

Additionally, I had to insert the following code inside the **<head>...</head>** tags:

```

<script>

    $(document).ready(
        function() {
            $('#album').append( "<div style='border: dashed magenta 8px;'> <h1>Here is
some content for the 'album' div</h1></div>");
        }
    );
</script>

```

The code above does the following: it iterates through the two nested loops counting up to nine in three groups. The next thing it does is for each group of three it prints out a div with the **w3-row-class** and creates a new div element to represent a row and appends it to the **"album" div**

The next thing I had to do is to view this code on a web browser and inspect the DOM(Document Object Model) to see the elements which have been added to the **album div**

What is more, I wanted to insert a whole grid of photographs so I needed a more complicated function so I had to insert the following code. This will replace all the code that I have already put in:

```

$(document).ready(
    function() {
        for (i = 0; i < 9; i += 3) {
            var rowDiv = $( "<div class='w3-row'>");
            $('#album').append( rowDiv);
            for( j = i; j < i + 3; j++ ) {
                if ( j < 9) {
                    rowDiv.append( "<div style='border: solid blue 2px;' class='w3-
third'>Place for photo</div>");
                }
            }
        }
    }
);

```

This code does the following: it converts the JSON string stored in the JSON variable into an actual array of the object and assigns them to the variable **photograph**. This is achieved using the **\$.parseJSON** function provided by the JQuery library. The next thing it does is it iterates through each element in the photo array using nested loops to handle them three at a time and for each row of three it prints out a **div** with the **w3-row-class** and creates a new **div** element to represent the row and appends the row to the **album div**. Also for each object it appends a div to the row with a blue border containing the value of the name field of the photo object.

The next thing I had to do is to view this on a web browser and inspect the DOM(Document Object Model) to see the elements which have been added to the **album div**

Furthermore I had to insert the following code inside the **<script>...</script>** tag:

```
var json = '[' +
  '{ "image": "b1.png", "name": "John", "description": "<p>This is John Lennon. He was one of the Beatles.</p><p>He was one of the two main song-writers.</p>" },' +
  '{ "image": "b2.png", "name": "Paul", "description": "<p>This is Paul McCartney. He was also one of the Beatles.</p><p>He was the other main song-writers.</p>" },' +
  '{ "image": "b3.png", "name": "George", "description": "<p>This is George Harrison. He was also one of the Beatles.</p>" },' +
  '{ "image": "b4.png", "name": "Ringo", "description": "<p>This is Ringo Starr. He was also one of the Beatles.</p>" }' +
  ']';
```

The next thing I had to do is to insert the following code. This will replace the `$(document).ready()` code block:

```
$(document).ready(
function() {
  var photos = $.parseJSON( json);
  for (i = 0; i < photos.length; i += 3) {
    var rowDiv = $( "<div class='w3-row'>" );
    $('#album').append( rowDiv);
    for( j = i; j < i + 3; j++ ) {
      if ( j < photos.length) {
        rowDiv.append( "<div style='border: solid blue 2px;' class='w3-third'>" +
photos[j].name + "</div>");
      }
    }
  }
});
```

This code is similar to the previous one. Additionally I had to insert the following function code:

```
function createPhotoHTML( photoInfo) {
  return " <div class='w3-container w3-row w3-col 14'>" +
    " <div class='w3-container w3-card-8 w3-margin-8'>" +
    " <div class='w3-container w3-half w3-image'>" +
    " <a class='example-image-link' data-lightbox='example-1' href='images/" + photoInfo.image + "'><img src='images/" + photoInfo.image + "' class='example-image w3-circle'></a>" +
    " <div class='w3-title w3-text-purple'>" + photoInfo.name + "</div>" +
    " </div>" +
    " <div class='w3-container w3-half'>" + photoInfo.description +
    " </div>" +
    " </div>";
}
```

The next thing I had to do is change the following code in `$(document).ready`

From:

```
rowDiv.append( "<div style='border: solid blue 2px;' class='w3-third'>" +
photos[j].name + "</div>");
```

To:

```
rowDiv.append( createPhotoHTML( photos[j]));
```

This code is similar which calls the function **createPhotoHTML** with the relevant photo object, the final thing I had to do is to upload the file on daydream

This is the fully working code for **Photo Album Client-Rendered Version A**

```
<!DOCTYPE html>
<html>
<head>
<title>JSON Photo Album</title>
```

```

<meta name='viewport' content='width=device-width, initial-scale=1'>
<link rel='stylesheet' href='http://www.w3schools.com/lib/w3.css'>
<link rel='stylesheet' href='lib/lightbox2/css/lightbox.css'>
<link rel='stylesheet' href='photoalbum.css'>

<script src='http://code.jquery.com/jquery-1.11.3.min.js'></script>

<script>

$(document).ready(
  function() {
    var photos = $.parseJSON( json);
    for (i = 0; i < photos.length; i += 3) {
      var rowDiv = $( "<div class='w3-row'>" );
      $('#album').append( rowDiv);
      for( j = i; j < i + 3; j++ ) {
        if ( j < photos.length) {
          rowDiv.append( createPhotoHTML( photos[j]));
        }
      }
    }
  }
);
var json = '[' +
  '{ "image": "b1.png", "name": "John", "description": "<p>This is John Lennon. He was one of the Beatles.</p><p>He was one of the two main song-writers.</p>" },' +
  '{ "image": "b2.png", "name": "Paul", "description": "<p>This is Paul McCartney. He was also one of the Beatles.</p><p>He was the other main song-writers.</p>" },' +
  '{ "image": "b3.png", "name": "George", "description": "<p>This is George Harrison. He was also one of the Beatles.</p>" },' +
  '{ "image": "b4.png", "name": "Ringo", "description": "<p>This is Ringo Starr. He was also one of the Beatles.</p>" }' +
  ']';

function createPhotoHTML( photoInfo) {
  return " <div class='w3-container w3-row w3-col 14'>" +
    " <div class='w3-container w3-card-8 w3-margin-8'>" +
    " <div class='w3-container w3-half w3-image'>" +
    " <a class='example-image-link' data-lightbox='example-1' href='images/" + photoInfo.image + "'><img src='images/" + photoInfo.image + "' class='example-image w3-circle'></a>" +
    " <div class='w3-title w3-text-purple'>" + photoInfo.name + "</div>" +
    " </div>" +
    " <div class='w3-container w3-half'>" + photoInfo.description +
    " </div>" +
    " </div>" +
    " </div>";
}

</script>

</head>
<body>

<header class='w3-container w3-orange w3-center'>
  <h1>My Photo Album</h1>
  <h2>Timothy Lam</h2>
  <p>Here are some photographs and descriptions.</p>
</header>

<div id='message'>

</div>

<div id='album'>

</div>

<footer class='w3-container w3-orange w3-padding-jumbo'>
  <div class='w3-container w3-center'>
    <a href='mailto:lamt3@lsbu.ac.uk' class='link'>Contact me</a>
  </div>
</footer>

```

```
<script src='lib/lightbox2/js/lightbox.js'></script>
</body>
</html>
```

I completed the whole of the activities on the exercises. The activities in the exercise given to me were easy which I did not have any difficulties on the exercises as I followed the instructions carefully

The important thing I have learnt is how to use Javascript on an html web page and where to use the Javascript. I have also learnt that Javascript is different to PHP and html as it does not use any content in the body. The Javascript uses a **<script>..</script>** tag which prints out the whole of the web pages but it uses PHP and CSS to make the web page run and work.

I think the exercise fits into the bigger picture as you can use PHP, HTML, CSS and Javascript together as it will enable the web page to work nicely. In addition it makes the website layout more neater and more attractive when users are browsing through the web site.

The exercise helped me understand about digital business application as many web pages use Javascript. This will allow the web page and layout to be attractive. Also the Javascript does not use any text as it uses functions. This will make the code neater and more readable for users. There is no extended task given in this exercise

2.5. Week 11 - JavaScript and Ajax

The Exercise was **Exercise 2: Photo Album Client-Rendered Version B** and **Exercise 3: Photo Album Client-Rendered Version C**

2.5.1. Exercise 2: Photo Album Client-Rendered Version B

<http://daydream.lsbu.ac.uk/myweb/lamt3/photoalbum-client-rendered-version-b.html>

In **Exercise 2 Photo Album Client-Rendered Version B** I had to make a copy of **photo-album-client-rendered-version-a.html** and call it **photo-album-client-server-version-b.html** and delete the JSON string declaration at the top

```
var json = '[' +
  '{ "image": "b1.png", "name": "John", "description": "<p>This is John Lennon. He was one of the Beatles.</p><p>He was one of the two main song-writers.</p>" },' +
  '{ "image": "b2.png", "name": "Paul", "description": "<p>This is Paul McCartney. He was also one of the Beatles.</p><p>He was the other main song-writers.</p>" },' +
  '{ "image": "b3.png", "name": "George", "description": "<p>This is George Harrison. He was also one of the Beatles.</p>" },' +
  '{ "image": "b4.png", "name": "Ringo", "description": "<p>This is Ringo Starr. He was also one of the Beatles.</p>" }' +
  ']' ;
DELETE THIS CODE
```

The next thing I had to do is keep the **createPhotoHTML**, but delete the existing function that displays the photos

```
$( document ).ready(
function() {
  var photos = $.parseJSON( json);
  for (i = 0; i < photos.length; i += 3) {
    var rowDiv = $( "<div class='w3-row'>" );
    $('#album').append( rowDiv);
    for( j = i; j < i + 3; j++ ) {
      if ( j < photos.length) {
        rowDiv.append( createPhotoHTML( photos[j]));
      }
    }
  }
});
```

```

    }
  }
}
);
DELETE THIS CODE

```

In addition I had to insert the following code which is a function that loads the photo data from the server and generates the html to display the photos and also a **\$(document).ready** function to trigger it.

```

function loadPhotos() {
    $.getJSON( "json.txt",
        function( photos) {
            $('#album').empty();
            for (i = 0; i < photos.length; i += 3) {
                var rowDiv = $( "<div class='w3-row'>");
                $('#album').append( rowDiv);
                for( j = i; j < i + 3; j++ ) {
                    if ( j < photos.length) {
                        rowDiv.append( createPhotoHTML( photos[j]));
                    }
                }
            }
        }
    );
}

$( document).ready(
    function() {
        loadPhotos();
    }
);

```

The script above **loadPhoto** uses the JQuery helper function to get JSON that is required for two parameters. The first is a URL which the JSON data object provides and the second parameter is where the object is passed through the function. This is where the **\$(document).ready** function loops through printing the photographs adding the photographs to the DOM(Document Object Model) The next thing I had to do is to insert the following code and save it as **json.txt** alongside the new photo album script. This will display two photographs based on the JSON data

```

[
  {
    "image": "b3.png",
    "name": "FAKED",
    "description": "<p>This is JSON data read from a plain text file.</p>"
  },
  {
    "image": "b4.png",
    "name": "JSON",
    "description": "<p>A better idea would be to read it from a dynamic process.</p>"
  }
]

```

What is more, the JSON is just reading from a static file which is not much use and I wanted the JSON to read from the PHP script which meant I had to insert the following code which has a similar function, as it converts a JSON data structure to **dbutils.php** as **dbutils.php** script has a function that reads from the table from the database

```

function jsonTable( $pdo, $table, $whereOrder = "" ) {
    $sql = "SELECT * FROM `".$table."`";
    if ( $whereOrder != "" ) {
        $sql .= " ".$whereOrder;
    }
}

```

```
// specify only an associative array to be returned
$stmt = $pdo->query( $sql, PDO::FETCH_ASSOC);
print json_encode( $stmt->fetchAll(PDO::FETCH_ASSOC));
}
```

The Final thing I had to do in this exercise was to insert the following code and save it as **json-photoalbum-photos.php**

```
<?php
header("content-type: application/json");
require_once("lib/dbutils.php");

$pdo = connect();

print jsonTable( $pdo, "photographs", "ORDER BY `photoid`");
?>
```

This Script above returns a JSON data that is required from another details and this is the final version of **photo-album-client-rendered-version-b.html** below

```
<!DOCTYPE html>
<html>
<head>
<title>JSON Photo Album</title>
<meta name='viewport' content='width=device-width, initial-scale=1'>
<link rel='stylesheet' href='http://www.w3schools.com/lib/w3.css'>
<link rel='stylesheet' href='lib/lightbox2/css/lightbox.css'>
<link rel='stylesheet' href='photoalbum.css'>

<script src='http://code.jquery.com/jquery-1.11.3.min.js'></script>

<script>

function createPhotoHTML( photoInfo) {
  return " <div class='w3-container w3-row w3-col 14'>" +
    " <div class='w3-container w3-card-8 w3-margin-8'>" +
    " <div class='w3-container w3-half w3-image'>" +
    " <a class='example-image-link' data-lightbox='example-1'
href='images/" + photoInfo.image + "'><img src='images/" + photoInfo.image + "'
class='example-image w3-circle'></a>" +
    " <div class='w3-title w3-text-purple'>" + photoInfo.name + "</div>"
+
    " </div>" +
    " <div class='w3-container w3-half'>" + photoInfo.description +
    " </div>" +
    " </div>";
}

function loadPhotos() {
  $.getJSON( "json-photoalbum-photos.php",
    function( photos) {
      $('#album').empty();
      for (i = 0; i < photos.length; i += 3) {
        var rowDiv = $( "<div class='w3-row'>");
        $('#album').append( rowDiv);
        for( j = i; j < i + 3; j++ ) {
          if ( j < photos.length) {
            rowDiv.append( createPhotoHTML( photos[j]));
          }
        }
      }
    }
  );
}

$( document).ready(
  function() {
    loadPhotos();
  }
);
```



```

</script>
</head>
<body>

<header class='w3-container w3-orange w3-center'>
  <h1>My Photo Album</h1>
  <h2>Timothy Lam</h2>
  <p>Here are some photographs and descriptions.</p>
</header>

<div id='message'>

</div>

<div id='album'>

</div>

<footer class='w3-container w3-orange w3-padding-jumbo'>
  <div class='w3-container w3-center'>
    <a href='mailto:lamt3@lsbu.ac.uk' class='link'>Contact me</a>
  </div>
</footer>

<script src='lib/lightbox2/js/lightbox.js'></script>
</body>
</html>

```

2.5.2. Exercise 3: Photo Album Client-Rendered Version C

<http://daydream.lsbu.ac.uk/myweb/lamt3/photoalbum-client-rendered-version-c.html>

<http://daydream.lsbu.ac.uk/myweb/lamt3/example.html>

In Exercise 3, **Photo Album Client-Rendered Version C**, I had to make a version of **photo-album-client-rendered-version-b.html** and call it **photo-album-client-rendered-version-c.html** and add the highlighted line to the existing function to **createHTML** function

```

function createPhotoHTML( photoInfo) {
  return " <div class='w3-container w3-row w3-col 14'>" +
    "   <div class='w3-container w3-card-8 w3-margin-8'>" +
    "     <div class='w3-container w3-half w3-image'>" +
    "       <a class='example-image-link' data-lightbox='example-1'
href='images/" + photoInfo.image + "'><img src='images/" + photoInfo.image + "'
class='example-image w3-circle'></a>" +
    "       <div class='w3-title w3-text-purple'>" + photoInfo.name + "</div>"
+
    "     </div>" +
    "     <div class='w3-container w3-half'>" + photoInfo.description +
    "       // next line should be conditional on whether logged in or not
    "       <span class='link deletelink' id='" + photoInfo.photoid +
    "'>(Delete)</span>" +
    "     </div>" +
    "   </div>" +
    " </div>";
}

```

This script above is a link which will be added beside each of the pictures In addition, I had to try the following example in a new file

```

<html>
<head>
<script src='http://code.jquery.com/jquery-1.11.3.min.js'></script>
<script>
$(document).ready(
  function() {
    $("h1").click( function( event) {

```

```

        alert( "You clicked: " + event.target.innerHTML);
    })
}
);
</script>
</head>
<body>
<h1>Click Me</h1>
<h1>Click me too!</h1>
</body>
</html>

```

What is more, I had to modify the **loadPhoto** function below and add a click handler to all of the delete links. This added a function which is similar to the example above

```

function loadPhotos() {
    $.getJSON( "json.txt",
        function( photos) {
            $('#album').empty();
            for (i = 0; i < photos.length; i += 3) {
                var rowDiv = $( "<div class='w3-row'>");
                $('#album').append( rowDiv);
                for( j = i; j < i + 3; j++ ) {
                    if ( j < photos.length) {
                        rowDiv.append( createPhotoHTML( photos[j]));
                    }
                }
            }
            // add listener to all deletelinks
            $('.deletelink').click(
                function( event) {
                    alert('Photo id: ' + event.target.id);
                }
            );
        }
    );
}

```

This code above added links which runs the function when the link is clicked on and the function knows the id number of the photo. What is more, I had to insert the following function

```

function deletePhoto( event) {
    $.post( "json-photoalbum-delete.php",
        "deletionid=" + event.target.id,
        function( data) {
            if ( data.result != "success") {
                alert( "ERROR: " + data.error);
            } else {
                loadPhotos();
            }
        }
    );
}

```

This code above will now remember the photographs that are stored on the server so it will have to ask the server to do the deletion Furthermore, I had to insert a PHP script and save it as **json-photoalbum-delete.php**

```

<?php
header("content-type: application/json");
require_once("lib/dbutils.php");
require_once("photoalbum-common.php");

$pdo = connect();

if ( isset( $_POST['deletionid'])) {
    $errorMessage = deletePhotograph( $pdo, $_POST['deletionid']);
} else {
    $errorMessage = "No id provided in post";
}

```

```

}

if ( $errorMessage == "" ) {
    print "{\"result\": \"success\"}";
} else {
    print "{";
    print "\"result\": \"failure\"";
    print ",";
    print "\"error\": \"$errorMessage\"";
    print "}";
}

?>

```

This code above adds a click handler to the delete link so instead of doing ***alert('Photo id: ' + event.target.id);*** it does ***deletePhoto(event);*** instead and below is the full code for **photo-album-client-rendered-version-c.html**

```

<!DOCTYPE html>
<html>
<head>
<title>JSON Photo Album</title>
<meta name='viewport' content='width=device-width, initial-scale=1'>
<link rel='stylesheet' href='http://www.w3schools.com/lib/w3.css'>
<link rel='stylesheet' href='lib/lightbox2/css/lightbox.css'>
<link rel='stylesheet' href='photoalbum.css'>

<script src='http://code.jquery.com/jquery-1.11.3.min.js'></script>

<script>

function createPhotoHTML( photoInfo) {
    return " <div class='w3-container w3-row w3-col 14'>" +
        " <div class='w3-container w3-card-8 w3-margin-8'>" +
        " <div class='w3-container w3-half w3-image'>" +
        " <a class='example-image-link' data-lightbox='example-1'
href='images/" + photoInfo.image + "'><img src='images/" + photoInfo.image + "'
class='example-image w3-circle'></a>" +
        " <div class='w3-tittle w3-text-purple'>" + photoInfo.name + "</div>"
+
        " </div>" +
        " <div class='w3-container w3-half'>" + photoInfo.description +
        " // next line should be conditional on whether logged in or not
        " <span class='link deletelink' id='" + photoInfo.photoid +
        "'>(Delete)</span>" +
        " </div>" +
        " </div>" +
        " </div>";
}

function loadPhotos() {
    $.getJSON( "json-photoalbum-photos.php",
        function( photos) {
            $('#album').empty();
            for (i = 0; i < photos.length; i += 3) {
                var rowDiv = $( "<div class='w3-row'>");
                $('#album').append( rowDiv);
                for( j = i; j < i + 3; j++) {
                    if ( j < photos.length) {
                        rowDiv.append( createPhotoHTML( photos[j]));
                    }
                }
            }

            // add listener to all deletelinks
            $('.deletelink').click(
                function( event) {
                    //alert('Photo id: ' + event.target.id);
                    deletePhoto(event);
                }
            );
        }
    );
}

}

```

```

function deletePhoto( event) {
    $.post( "json-photoalbum-delete.php",
        "deletionid=" + event.target.id,
        function( data) {
            if ( data.result != "success") {
                alert( "ERROR: " + data.error);
            } else {
                loadPhotos();
            }
        });
}

$( document).ready(
    function() {
        loadPhotos();
    }
);

</script>

</head>
<body>

<header class='w3-container w3-orange w3-center'>
    <h1>My Photo Album</h1>
    <h2>Timothy Lam</h2>
    <p>Here are some photographs and descriptions.</p>
</header>

<div id='message'>

</div>

<div id='album'>

</div>

<footer class='w3-container w3-orange w3-padding-jumbo'>
    <div class='w3-container w3-center'>
        <a href='mailto:lamt3@lsbu.ac.uk' class='link'>Contact me</a>
    </div>
</footer>

<script src='lib/lightbox2/js/lightbox.js'></script>
</body>
</html>

```

I managed to complete both exercises The Exercise was a bit easy as I read and followed the exercise carefully so I did not miss any important part. I had no difficulty with any part of the exercise. The important thing I have learnt is that you can create JavaScript web pages using a web page you have already created using PHP just like the Photo Album Client-Rendered Version B and Photo Album Client-Rendered Version C where I used the PHP web page which is Photo Album Client Version B and Photo Album Client Version C

I also learnt that the **json-photoalbum.php** works by firstly fetching the data from **dbutils.php** and then printing out the photographs where it matches the photo id and the **json-photoalbum.php** is linked to photo-album-client-rendered-version.b.html where it runs through the JavaScript function in order to load the web page and make the web page work. I think the **Exercise 2: Photo Album Client-Rendered Version B.html** and **Exercise 3: Photo Album Client-Rendered Version C** fits in to the bigger pictures as you can recreate a PHP web page but using JavaScript also JavaScript fetches data from PHP but the JavaScript works differently when loading the web page as JavaScript has to go through the function of JavaScript before loading the web page

The Exercise help me understand about Digital Business Application by knowing when to use JavaScript and how to use JavaScript it also helped me understand that JavaScript uses function to load the page and make the web page working. There was no extended task given in **Exercise 2: Photo Album Client-Rendered Version B** and **Exercise 3 Photo Album Client-Rendered Version C**

2.6. Week 12 - JavaScript and Ajax

The Exercise was **Exercise 4: Photo Album Client-Rendered Version D**, **Exercise 5: Photo Album Client-Rendered Version E** and **Exercise 6: Extended Task – Photo Album Server Version E**

2.6.1. Exercise 4: Photo Album Client-Rendered Version D

<http://daydream.lsbu.ac.uk/myweb/lamt3/photoalbum-client-rendered-version-d.html>

In **Exercise 4 Photo Album Client-Rendered Version D** I had to download a file from moodle called *ajaxupload.zip* and extract it and put the *Ajax* upload folder into the *lib* folder and upload it on to the web server

The next thing I had to do is to make a copy of **photoalbum-client-rendered-version-c.html** and call it **photoalbum-client-rendered-version-d.html**. Additionally I had to add the following line after the import JQuery library

```
<script src='lib/ajaxupload/jquery-fileupload.min.js'></script>
```

This code will make the **fileupload** function available to use. What is more I had to insert the **fileupload** function inside the function **\$(document).ready**

```
$( document).ready(
function() {
    loadPhotos();
    $('#photoform').fileUpload({
        beforeSubmit :
            function(uploadData){ console.log(uploadData); return true; },
        // access the data returned by the upload return false to stop the submit
    ajax call
        success :
            function(data, textStatus, jqXHR){ if ( data.result != "success") { alert(
"ERROR: " + data.error); }},
        // Callback for the submit success ajax call
        error :
            function(jqXHR, textStatus, errorThrown){ alert( "error");
console.log(jqXHR); },
        // Callback if an error happens with your upload call or the submit call
        complete :
            function(jqXHR, textStatus){ $('#photoform').get(0).reset();
location.href='#'; loadPhotos(); }
        // Callback on completion
    });
}
);
```

The code above does the following:

1. Clears all the fields of the upload form. (*\$('#photoform').get(0).reset();*)
2. Clears the value of the targeted link causing the form to be hidden again. (*location.href='#';*)
3. Runs *loadPhotos()* again, to rebuild the gallery with the newly uploaded image included.

Furthermore I had to create a file called **json-photalbum-upload.php** and insert the following script to the file

```

<?php
require_once("lib/dbutils.php");
require_once("photoalbum-common.php");
header("content-type: application/json");

// This is called twice, once with the form value file_upload_incoming set to the
// number of files,
// and the file data then again with values of the form fields. Note that the file
// data DOES NOT
// come in the normal format $_FILES['formfieldname'] but as a numbered array
$_FILES[0] etc.

$pdo = connect();

if ( isset( $_POST['file_upload_incoming'])) {
    // FIRST AJAX INVOCATION

    // create a database record with dummy name and description but pointing to the
    // uploaded photo
    $photoId = addPhotograph( $pdo, 'incoming', 'pending', 'self', $_FILES[0]);
    // send the photoId back to the javascript process on the client
    print "{\"photoId\": \"\".$photoId.\"\"}";
} else {
    // SECOND AJAX INVOCATION

    // now update the database with the other fields
    $stmt = $pdo->prepare("UPDATE photographs SET `name`=?, `description`=? WHERE
`photoId`=?");
    $stmt->execute( array( $_POST['name'], $_POST['description'], $_POST['photoId']));
    print "{\"completed photoId\": \"\".$_POST['photoId'].\"\"}";
}
}
?>

```

The code above includes the **dbutils.php** in order to access the database and **photoalbum-common.php** and in order to make use of the **addPhotograph** function we have already used the function for the upload using PHP. What is more that code above does the following: the script is called twice by the upload code; the first is with the file data and then the form field data. This is the full version of **Photo Album Client-Rendered Version D**

```

<!DOCTYPE html>
<html>
<head>
<title>JSON Photo Album</title>
<meta name='viewport' content='width=device-width, initial-scale=1'>
<link rel='stylesheet' href='http://www.w3schools.com/lib/w3.css'>
<link rel='stylesheet' href='lib/lightbox2/css/lightbox.css'>
<link rel='stylesheet' href='photoalbum.css'>

<script src='http://code.jquery.com/jquery-1.11.3.min.js'></script>
<script src='lib/ajaxupload/jquery-fileupload.min.js'></script>

<script>
function createPhotoHTML( photoInfo) {
    return " <div class='w3-container w3-row w3-col 14'>" +
        " <div class='w3-container w3-card-8 w3-margin-8'>" +
        " <div class='w3-container w3-half w3-image'>" +
        " <a class='example-image-link' data-lightbox='example-1'
href='images/" + photoInfo.image + "'><img src='images/" + photoInfo.image + "'
class='example-image w3-circle'></a>" +
        " <div class='w3-title w3-text-purple'>" + photoInfo.name + "</div>"
+
        " </div>" +
        " <div class='w3-container w3-half'>" + photoInfo.description +
// next line should be conditional on whether logged in or not
        " <span class='link deletelink' id='" + photoInfo.photoId +
">(Delete)</span>" +
        " </div>" +
        " </div>" +
        " </div>";
}

```

```

function loadPhotos() {
    $.getJSON( "json-photoalbum-photos.php",
        function( photos) {
            $('#album').empty();
            for (i = 0; i < photos.length; i += 3) {
                var rowDiv = $( "<div class='w3-row'>" );
                $('#album').append( rowDiv );
                for( j = i; j < i + 3; j++ ) {
                    if ( j < photos.length ) {
                        rowDiv.append( createPhotoHTML( photos[j]) );
                    }
                }
            }

            // add listener to all deletelinks
            $('.deletelink').click(
                function( event) {
                    //alert('Photo id: ' + event.target.id);
                    deletePhoto(event);
                }
            );
        }
    );
}

function deletePhoto( event) {
    $.post( "json-photoalbum-delete.php",
        "deletionid=" + event.target.id,
        function( data) {
            if ( data.result != "success") {
                alert( "ERROR: " + data.error );
            } else {
                loadPhotos();
            }
        }
    );
}

$( document ).ready(
    function() {
        loadPhotos();
        $('#photoform').fileUpload({
            beforeSubmit :
                function(uploadData){ console.log(uploadData); return true; },
            // access the data returned by the upload return false to stop the submit
            ajax call
                success :
                    function(data, textStatus, jqXHR){ },
                    // Callback for the submit success ajax call
                error :
                    function(jqXHR, textStatus, errorThrown){ alert( "error" );
                    console.log(jqXHR); },
                    // Callback if an error happens with your upload call or the submit call
                complete :
                    function(jqXHR, textStatus){ $('#photoform').get(0).reset();
                    location.href='#'; loadPhotos(); }
                    // Callback on completion
                });
    }
);

</script>

</head>
<body>

<header class='w3-container w3-orange w3-center'>
    <h1>My Photo Album</h1>
    <h2>Timothy Lam</h2>
    <p>Here are some photographs and descriptions.</p>
</header>

<div id='message'>

</div>

```

```

<div id='album'>

</div>
<div class='w3-container w3-center'>
  <a href="#id01" class='link'>Add images</a>
</div>

<div id="id01" class="w3-modal">
  <div class="w3-modal-dialog">
    <div class="w3-modal-content w3-card-4">
      <header class="w3-container w3-teal">
        <a href="#" class="w3-closebtn">&times;</a>
        <h2>Upload photograph</h2>
      </header>
      <div class="w3-container">
        <form action='json-photoalbum-upload.php' method='post'
enctype='multipart/form-data' id='photoform' class='w3-container'>
          <div class="w3-group">
            <input type='file' name='uploadfile' class='w3-input'>
            <label class="w3-label">Image file</label>
          </div>
          <div class="w3-group">
            <input type='text' name='name' class='w3-input' required>
            <label class="w3-label">Name</label>
          </div>
          <div class="w3-group">
            <textarea name='description' class='w3-input' required></textarea>
            <label class="w3-label">Description</label>
          </div>
          <input type='hidden' name='formid' value='fileupload'>

          <input type='submit' value='Upload' id='uploadbutton' class='w3-btn'>
        </form>
      </div>
      <footer class="w3-container w3-teal">
        <p></p>
      </footer>
    </div>
  </div>
</div>
<footer class='w3-container w3-orange w3-padding-jumbo'>
  <div class='w3-container w3-center'>
    <a href='mailto:lamt3@lsbu.ac.uk' class='link'>Contact me</a>
  </div>
</footer>

<script src='lib/lightbox2/js/lightbox.js'></script>
</body>
</html>

```

2.6.2. Exercise 5: Photo Album Client-Rendered Version E

Photo Album Client Rendered Version E:

<http://daydream.lsbu.ac.uk/myweb/lamt3/photoalbum-client-rendered-version-e.html>

In Exercise 5: **Photo Album Client-Rendered Version E** I had to restore the use of authentication to the site (corresponding to the PHP version e).

The first thing I had to do is to insert the following code in both files **json-photoalbum-delete.php** and **json-photoalbum-upload.php**. This code will prevent you from uploading and deleting the images.

```

/* For part e */
setUpSession();
if ( $_SESSION['user'] == "anonymous") {
  print "{";
  print "\"result\": \"failure\"";
  print ",";
  print "\"error\": \"not logged in\"";
  print "}";
  exit();
}

```


The next thing I had to do is to add the following code and call it **json-photoalbum-login.php**. This will enable you to login.

```
<?php
header("content-type: application/json");
require_once("lib/dbutils.php");

$pdo = connect();

$action = loginOrOut( $pdo, 'users');

print "{";
print "\"action\": \"".$action."\"";
print ",";
print "\"user\": \"".$_SESSION['user']."\"";
print ",";
print "\"role\": \"".$_SESSION['role']."\"";
print "}";

?>
```

The script above uses the existing **loginOrOut** function in **dbutils** to start a session. If the username and password are provided correctly it will then print out the JSON object containing the username and the role that has been assigned

The next thing I had to do is to make a copy of photoalbum-client-rendered-version-d.html and call it as photoalbum-client-rendered-version-e.html

This will add the HTML login form that was given in PHP and MySQL Exercise 6 to a new version of the page. The next thing I had to do is to add the following div between the photo album header and the unused message in the **div**

```
<header class='w3-container w3-orange w3-center'>
  <h1>My Photo Album</h1>
  <h2>Your Name</h2>
  <p>Here are some photographs and descriptions.</p>
</header>

<div class='w3-container w3-center' id='userinfo'>

</div>

<div id='message'>

</div>
```

This **div** will display log in and log out link

What is more I had to add the following script just after the opening **<script>** tag

```
var user = "anonymous";
var role = "anonymous";

function renderUserInformation() {
  $('#userinfo').empty();
  if ( user == "anonymous" ) {
    $('#userinfo').append( "<a href='#id02' class='link'>Log in</a>");
    $('#addlink').hide();
  } else {
    $('#userinfo').append( "User: " + user + " Role: " + role + " <a href='#' class='link' onclick='logout()'>Log out</a>");
    $('#addlink').show();
  }
}
```

```

function login() {
    $.post( "json-photoalbum-logon.php",
        $('#loginform').serialize(),
        function( data ) {
            user = data.user;
            role = data.role;
            renderUserInformation();
            loadPhotos();
            $('#loginform').get(0).reset();
            location.href='#';
            alert( data.action + " " + data.user + " " + data.role);
        }
    );
}

function logout() {
    $.getJSON( "json-photoalbum-logon.php?logout=true",
        function( data ) {
            user = data.user;
            role = data.role;
            renderUserInformation();
            loadPhotos();
            location.href='#';
            //alert( data.action + " " + data.user + " " + data.role);
        }
    );
}

```

This code above does the following: starts by declaring two global variables, `user` and `role`, and setting them both to **"anonymous"**. Once logged in, these values will be replaced by the username and a role provided by the server. It also does the following: the function **renderUserInformation()** fills in the **div** we added a moment ago. When run it does the following: First it discards any content of the **div** with **id #userinfo**. Then if the value of the global variable `user` is **"anonymous"**, it adds a link to that **div** with the text **"Log in"** and then hides the link with the **id #addlink**, which is the **"Add images"** link. Thus if no one is logged in, it displays a **log in** link and hides the **add images** link. The **"Log in"** link uses the w3css classes to show the log in form when it is clicked. If the value of `user` is not anonymous, it means someone is logged in. In that case the function adds a display of the username and role, and a link with the text **"Log out"**. It also shows the add image link. The log out link has an **onclick()** handler pointing to the **logout()** function.

The **logout()** function constructs an AJAX request using get JSON and sends it to **json-photoalbum-logon.php** with the appended value **logout=true**. This will cause it to end the session on the server and return a JSON object with `user` and `role` fields set to **"anonymous"**. These are copied into the global variables in the JavaScript. It then does the following:

1. Calls **renderUserInformation()** to replace the log out link with a log in link
2. Calls **loadPhotos()** - this is to allow the delete links beside the images to be removed, but the code to do that has not yet been added.
3. Sets **location.href** to **'#'** to clear any target thus hiding the log on form, although this is unnecessary since it isn't showing (I copied it from when I wrote the **login()** function where it is necessary!)

To log in the user must click the log in link. This simply causes the log in form to be displayed. When the log in form is submitted it needs to call the **login()** function. (It doesn't yet do this.)

The **login()** function is very similar to the **logout()** function, but this time it creates a POST request using **\$.post()**, using the JQuery **serialize** function to read all the fields from the form and add it to the request. The request is again sent to **json-photoalbum-logon.php** and a JSON object is received which contains the username and role. If the log on was successful these will be real values, if the log on details were not recognised, they will be anonymous. The function then updates the log in link to be a log out link, reloads the photos (to add delete links beside each image), clears the log in form and then hides it. Finally it displays an alert either confirming log in or notifying of failure

Furthermore I had to make the login function work. This will make a connection when the first page loads by adding the following code to the **\$(document).ready** function

```
// Callback on completion
});
$('#loginform').submit(
function( event) {
    event.preventDefault();
    login();
}
);
};
);
```

The code above will prevent the form from being submitted to the server by the web browser but instead will call the **login()** which is submitted via AJAX. Additionally I had to change that following code from

```
"      <span class='link deletelink' id='" + photoInfo.photoid + "'>(Delete)</span>"
+      +
```

To

```
(( user != "anonymous") ? "<span class='link deletelink' id='" + photoInfo.photoid + "'>(Delete)</span>" : "" ) +
```

This will print out the delete link if the user is login and if the user is not login then the delete link will not show as previously the **createPhotoHTML** constructs and returns the string all in one statement and the best thing to do is to use the ternary operator to replace part of the conditionally the ternary operator looks like this

condition ? value-if-true : value-if-false

For example,

```
Status = (age >= 18) ? "adult" : "child";
```

This is the complete full code of **Exercise 5: Photo Album Client-Rendered Version E**

```
<!DOCTYPE html>
<html>
<head>
<title>JSON Photo Album</title>
<meta name='viewport' content='width=device-width, initial-scale=1'>
<link rel='stylesheet' href='http://www.w3schools.com/lib/w3.css'>
<link rel='stylesheet' href='lib/lightbox2/css/lightbox.css'>
<link rel='stylesheet' href='photoalbum.css'>

<script src='http://code.jquery.com/jquery-1.11.3.min.js'></script>
```

```

<script src='lib/ajaxupload/jquery-fileupload.min.js'></script>

<script>
var user = "anonymous";
var role = "anonymous";

function createPhotoHTML( photoInfo) {
    return " <div class='w3-container w3-row w3-col 14'>" +
        " <div class='w3-container w3-card-8 w3-margin-8'>" +
        " <div class='w3-container w3-half w3-image'>" +
        " <a class='example-image-link' data-lightbox='example-1'
href='images/" + photoInfo.image + "'><img src='images/" + photoInfo.image + "'
class='example-image w3-circle'></a>" +
        " <div class='w3-tittle w3-text-purple'>" + photoInfo.name + "</div>"
+
        " </div>" +
        " <div class='w3-container w3-half'>" + photoInfo.description +
        // next line should be conditional on whether logged in or not
        (( user != "anonymous") ? "<span class='link deletelink' id='" +
photoInfo.photoid + "'>(Delete)</span>" : "") +
        " </div>" +
        " </div>" +
        " </div>";
}

function loadPhotos() {
    $.getJSON( "json-photoalbum-photos.php",
        function( photos) {
            $('#album').empty();
            for (i = 0; i < photos.length; i += 3) {
                var rowDiv = $( "<div class='w3-row'>");
                $('#album').append( rowDiv);
                for( j = i; j < i + 3; j++ ) {
                    if ( j < photos.length) {
                        rowDiv.append( createPhotoHTML( photos[j]));
                    }
                }
            }
            // add listener to all deletelinks
            $('.deletelink').click(
                function( event) {
                    //alert('Photo id: ' + event.target.id);
                    deletePhoto(event);
                }
            );
        }
    );
}

function deletePhoto( event) {
    $.post( "json-photoalbum-delete.php",
        "deletionid=" + event.target.id,
        function( data) {
            if ( data.result != "success") {
                alert( "ERROR: " + data.error);
            } else {
                loadPhotos();
            }
        }
    );
}

function renderUserInformation() {
    $('#userinfo').empty();
    if ( user == "anonymous") {
        $('#userinfo').append( "<a href='#id02' class='link'>Log in</a>");
        $('#addlink').hide( "<a href='#id01' class='link'>Add images</a>");
    } else {
        $('#userinfo').append( "User: " + user + " Role: " + role + " <a href='#"
class='link' onclick='logout()'>Log out</a>");
        $('#addlink').show( "<a href='#id01' class='link'>Add images</a>");
    }
}

function login() {
    $.post( "json-photoalbum-logon.php",
        $('#loginform').serialize(),
        function( data) {

```

```

        user = data.user;
        role = data.role;
        renderUserInformation();
        loadPhotos();
        $('#loginform').get(0).reset();
        location.href='#';
        alert( data.action + " " + data.user + " " + data.role);
    }
    );
}

function logout() {
    $.getJSON( "json-photoalbum-logon.php?logout=true",
        function( data) {
            user = data.user;
            role = data.role;
            renderUserInformation();
            loadPhotos();
            location.href='#';
            //alert( data.action + " " + data.user + " " + data.role);
        }
    );
}

$( document).ready(
    function() {
        loadPhotos();
        $('#photoform').fileUpload({
            beforeSubmit :
                function(uploadData){ console.log(uploadData); return true; },
            // access the data returned by the upload return false to stop the submit
ajax call
            success :
                function(data, textStatus, jqXHR){ },
            // Callback for the submit success ajax call
            error :
                function(jqXHR, textStatus, errorThrown){ alert( "error");
console.log(jqXHR); },
            // Callback if an error happens with your upload call or the submit call
            complete :
                function(jqXHR, textStatus){ $('#photoform').get(0).reset();
location.href='#'; loadPhotos(); }
            // Callback on completion
        });

        $('#loginform').submit(
            function( event) {
                event.preventDefault();

                login();
            }
        );
    }
);

</script>

</head>
<body>

<header class='w3-container w3-orange w3-center'>
    <h1>My Photo Album</h1>
    <h2>Timothy Lam</h2>
    <p>Here are some photographs and descriptions.</p>
</header>

<div class='w3-container w3-center' id='userinfo'>
<a href="#id02" class='link'>Log in</a>
</div>

<div id='message'>

</div>

```

```

<div id='album'>

</div>

<div class='w3-container w3-center' id='addlink'>
<a href="#id01" class='link'>Add images</a>
</div>

<div id="id01" class="w3-modal">
  <div class="w3-modal-dialog">
    <div class="w3-modal-content w3-card-4">
      <header class="w3-container w3-teal">
        <a href="#" class="w3-closebtn">&times;</a>
        <h2>Upload photograph</h2>
      </header>
      <div class="w3-container">
        <form action='json-photoalbum-upload.php' method='post'
enctype='multipart/form-data' id='photoform' class='w3-container'>
          <div class="w3-group">
            <input type='file' name='uploadfile' class='w3-input'>
            <label class="w3-label">Image file</label>
          </div>
          <div class="w3-group">
            <input type='text' name='name' class='w3-input' required>
            <label class="w3-label">Name</label>
          </div>
          <div class="w3-group">
            <textarea name='description' class='w3-input' required></textarea>
            <label class="w3-label">Description</label>
          </div>
          <input type='hidden' name='formid' value='fileupload'>

          <input type='submit' value='Upload' id='uploadbutton' class='w3-btn'>
        </form>
      </div>
      <footer class="w3-container w3-teal">
        <p></p>
      </footer>
    </div>
  </div>
</div>

<div id='id02' class='w3-modal'>
  <div class='w3-modal-dialog'>
    <div class='w3-modal-content w3-card-4'>
      <header class='w3-container w3-teal'>
        <a href='#' class='w3-closebtn'>&times;</a>
        <h2>Log in</h2>
      </header>
      <div class='w3-container'>
        <form action='json-photoalbum-logon.php' method='post' id='loginform'
class='w3-container'>
          <div class='w3-group'>
            <input type='text' name='username' size='16' class='w3-input'>
            <label class='w3-label'>Username</label>
          </div>
          <div class='w3-group'>
            <input type='password' name='password' size='16' class='w3-input'>
            <label class='w3-label'>Password</label>
          </div>
          <input type='hidden' name='formid' value='login'>
          <input type='submit' value='Log in' id='loginbutton' class='w3-btn'>
        </form>
      </div>
      <footer class='w3-container w3-teal'>
        <p></p>
      </footer>
    </div>
  </div>
</div>

<div id='id03' class='w3-modal'>
  <div class='w3-modal-dialog'>
    <div class='w3-modal-content w3-card-4'>
      <header class='w3-container w3-teal'>
        <a href='#' class='w3-closebtn'>&times;</a>
        <h2>Contact me</h2>
      </header>
      <div class='w3-container'>
        <form action='mailto:lamt3@lsbu.ac.uk' method='post' id='contactform'
class='w3-container'>
          <div class='w3-group'>
            <input type='text' name='name' size='16' class='w3-input'>
            <label class='w3-label'>Name</label>
          </div>
          <div class='w3-group'>
            <input type='text' name='email' size='16' class='w3-input'>
            <label class='w3-label'>Email</label>
          </div>
          <div class='w3-group'>
            <input type='text' name='subject' size='16' class='w3-input'>
            <label class='w3-label'>Subject</label>
          </div>
          <div class='w3-group'>
            <input type='text' name='message' size='16' class='w3-input'>
            <label class='w3-label'>Message</label>
          </div>
          <input type='hidden' name='formid' value='contact'>
          <input type='submit' value='Contact me' id='contactbutton' class='w3-btn'>
        </form>
      </div>
      <footer class='w3-container w3-teal'>
        <p></p>
      </footer>
    </div>
  </div>
</div>

<div id='id04' class='w3-modal'>
  <div class='w3-modal-dialog'>
    <div class='w3-modal-content w3-card-4'>
      <header class='w3-container w3-teal'>
        <a href='#' class='w3-closebtn'>&times;</a>
        <h2>About me</h2>
      </header>
      <div class='w3-container'>
        <div class='w3-group'>
          <div class='w3-container'>
            <div class='w3-center'>
              <div class='w3-container w3-center'>
                <a href='mailto:lamt3@lsbu.ac.uk' class='link'>Contact me</a>
              </div>
            </div>
          </div>
        </div>
      </div>
      <footer class='w3-container w3-teal'>
        <p></p>
      </footer>
    </div>
  </div>
</div>

```

```

</footer>

<script src='lib/lightbox2/js/lightbox.js'></script>
</body>
</html>

```

I managed to complete both of the exercises. In addition, the exercise was easy as I had no difficulties in completing the exercise without any error

The import thing I have learnt about both of the Exercise was that you can use what you have learnt in PHP and MySQL Exercise and apply it to the exercise JavaScript and Ajax but JavaScript and AJAX uses both HTML and PHP to make the web page work. What is more I have also learnt that **Exercise 5: Photo Album Client-Rendered Version E** in JavaScript and AJAX is that this exercise uses 2 webpages one web pages is a login form display the photo album and the second web page is to display the user who is logged on which display the photo album with the delete link as well as a upload link. Furthermore I have also learnt in **Exercise 4: Photo Album Client-Rendered Version D** JavaScript and AJAX that you can use one code that prevents the user deleting the photo album unless the user is log in

I think these Exercise fits into the bigger picture as many website use JavaScript and AJAX this would make the code more organised and make the web page more attractive.

The exercises have help me understand that JavaScript and Ajax is different to HTML, PHP and MySQL but creates a web page. In addition the exercise also help me understand how JavaScript and Ajax work differently when loading up the page which is different to PHP and MySQL as JavaScript and Ajax firstly loads up the HTML page and then goes through the JavaScript and Ajax code to make the webpage work.

2.6.3.Exercise 6: Extended Task - Photo Album Server Version E

Extended Task: Photo Album Server Version E

<http://daydream.lsbu.ac.uk/myweb/lamt3/photoalbum-server-version-e%20ext.php>

The extended task I have done was **Exercise 6: Photo Album Server Version E** where I had to add a conditional statement to the script so that the link to the add images and link to delete the images are not displayed unless the user is logged in to achieved this I add the following after the end of the PHP Script;

```

print "</div>\n";

print "<div class='w3-container w3-center'>";
if($_SESSION['user'] != "anonymous") {
    print "<a href='#id01' class='link'>Add images</a>";
}
print"</div>";

```

The next thing I had to do is to delete the div class which contains add images links as inside the PHP script it has already got an add images link with a conditional statement.

```

<div class='w3-container w3-center'>
    <a href="#id01" class='link'>Add images</a>
</div>

```

This the full working code of extended task for **Exercise 6: Photo Album Server Version E**

```

<!DOCTYPE html>
<?php
require_once( 'lib/dbutils.php');

```

```

require_once("photoalbum-common.php");
?>
<html>
<head>
<title>HTML Photo Album</title>
<meta name='viewport' content='width=device-width, initial-scale=1'>
<link rel='stylesheet' href='photoalbum.css'>
<link rel='stylesheet' href='http://www.w3schools.com/lib/w3.css'>
<script src='http://code.jquery.com/jquery-1.11.3.min.js'></script>
<link rel='stylesheet' href='lib/lightbox2/css/lightbox.css'>
</head>
<body>

<header class='w3-container w3-orange w3-center'>
  <h1>My Photo Album</h1>
  <h2>Your Name</h2>
  <p>Here are some photographs and descriptions.</p>
</header>

<?php

$pdo = connect();
loginOrOut( $pdo, 'users');

if ( $_SESSION['user'] != "anonymous") {
    if ( isset( $_GET['deletionid'])) {
        $errorMessage = deletePhotograph( $pdo, $_GET['deletionid']);
        if ( $errorMessage != "" ) {
            print "<div class='errorMessage'>$errorMessage</div>\n";
        } else {
            print "<div class='message'>Image deleted.</div>\n";
        }
    }

    if ( isset( $_POST['formid']) && $_POST['formid'] == 'fileupload') {
        $photoid = addPhotograph( $pdo, $_POST['name'], $_POST['description'],
'self', $_FILES['uploadfile']);
    }
}

print "<div class='w3-container w3-center'>\n";
if ( $_SESSION['user'] == "anonymous") {
    print "   <a href='#id02' class='link'>Log in</a>\n";
} else {
    print "   User: " . $_SESSION['user'] . " Role: " . $_SESSION['role'] . " <a
href='?logout=true' class='link'>Log out</a>\n";
}
print "</div>\n";

$sql = "SELECT * FROM `photographs` ORDER BY `photoid`";
$stmt = $pdo->query( $sql);
// get all rows in an array
$rows = $stmt->fetchAll(PDO::FETCH_ASSOC);

$numberOfPhotos = count( $rows);

for( $i = 0; $i < $numberOfPhotos; $i++) {

    if ( $i % 3 == 0 ) {
        if ( $i > 0 ) { print "</div>\n"; }
        print "<div class='w3-row'>\n";
    }

    $row = $rows[$i];
    print "   <div class='w3-container w3-row w3-col 14'>\n";
    print "       <div class='w3-container w3-card-8 w3-margin-8'>\n";
    print "           <div class='w3-container w3-half w3-image'>\n";
    print "               <a class='example-image-link' data-lightbox='example-1'
href='images/" . $row['image'] . "'><img src='images/" . $row['image'] . "' class='example-
image w3-circle'></a>\n";
    print "           <div class='w3-title w3-text-purple'>" . $row['name'] . "</div>\n";

```



```

print "      </div>\n";
print "      <div class='w3-container w3-half'>\n";
print $row['description']. "\n";
if ( $_SESSION['user'] != "anonymous") {
    print "      <a href='?deletionid=".$row['photoid']."'
class='link'>(Delete)</a>";
}
print "      </div>\n";
print "    </div>\n";
print "  </div>\n";
}

print "</div>\n";

print "<div class='w3-container w3-center'>";
if($_SESSION['user'] != "anonymous") {
    print "<a href='#id01' class='link'>Add images</a>";
}
print"</div>";

?>
<div id="id01" class="w3-modal">
  <div class="w3-modal-dialog">
    <div class="w3-modal-content w3-card-4">
      <header class="w3-container w3-teal">
        <a href="#" class="w3-closebtn">&times;</a>
        <h2>Upload photograph</h2>
      </header>
      <div class="w3-container">
        <form action='?' method='post' enctype='multipart/form-data' id='photoform'
class='w3-container'>
          <div class="w3-group">
            <input type='file' name='uploadfile' class='w3-input'>
            <label class="w3-label">Image file</label>
          </div>
          <div class="w3-group">
            <input type='text' name='name' class='w3-input' required>
            <label class="w3-label">Name</label>
          </div>
          <div class="w3-group">
            <textarea name='description' class='w3-input' required></textarea>
            <label class="w3-label">Description</label>
          </div>
          <input type='hidden' name='formid' value='fileupload'>
          <input type='submit' value='Upload' id='uploadbutton' class='w3-btn'>
        </form>
      </div>
      <footer class="w3-container w3-teal">
        <p></p>
      </footer>
    </div>
  </div>
</div>
<div id='id02' class='w3-modal'>
  <div class='w3-modal-dialog'>
    <div class='w3-modal-content w3-card-4'>
      <header class='w3-container w3-teal'>
        <a href='#' class='w3-closebtn'>&times;</a>
        <h2>Log in</h2>
      </header>
      <div class='w3-container'>
        <form action='?' method='post' id='loginform' class='w3-container'>
          <div class='w3-group'>
            <input type='text' name='username' size='16' class='w3-input'>
            <label class='w3-label'>Username</label>
          </div>
          <div class='w3-group'>
            <input type='password' name='password' size='16' class='w3-input'>
            <label class='w3-label'>Password</label>
          </div>
          <input type='hidden' name='formid' value='login'>
          <input type='submit' value='Log in' id='loginbutton' class='w3-btn'>
        </form>
      </div>
      <footer class='w3-container w3-teal'>
        <p></p>
      </div>
    </div>
  </div>
</div>

```

```

        </footer>
    </div>
</div>
</div>

<footer class='w3-container w3-orange w3-padding-jumbo'>
    <div class='w3-container w3-center'>
        <a href='mailto:childm@lsbu.ac.uk' class='link'>Contact me</a>
    </div>
</footer>
<script src='lib/lightbox2/js/jquery.js'></script>
<script src='lib/lightbox2/js/lightbox.js'></script>
</body>
</html>

```

There were no extended task given for JavaScript and AJAX Exercises: **Exercise 4: Photo Album Client-Rendered Version D** and **Exercise 5: Photo Album Client-Rendered Version E** I manage to complete the PHP and MySQL Extend task for **Exercise 6: Photo Album Server Version E**. What is more, I never had any difficulties doing the extended task for **Exercise 6: Photo Album Server Version E** in PHP and MySQL

I have learnt that from doing the extended task for **Exercise 6: Photo Album Server Version E** in PHP and MySQL is that it will allow the user to delete the photo and add images unless the user is logged in and if the user is not logged in then the add images and the delete images link will not show

I think the Extended task for **Exercise 6: Photo Album Server Version E** in PHP and MySQL has helped me understand that users will need to have a login access; for example username and password to give the user full access and full control like delete, add etc. and if the user does not have a login access then the user will have limited access and can only view the web page.

2.7. Week 13

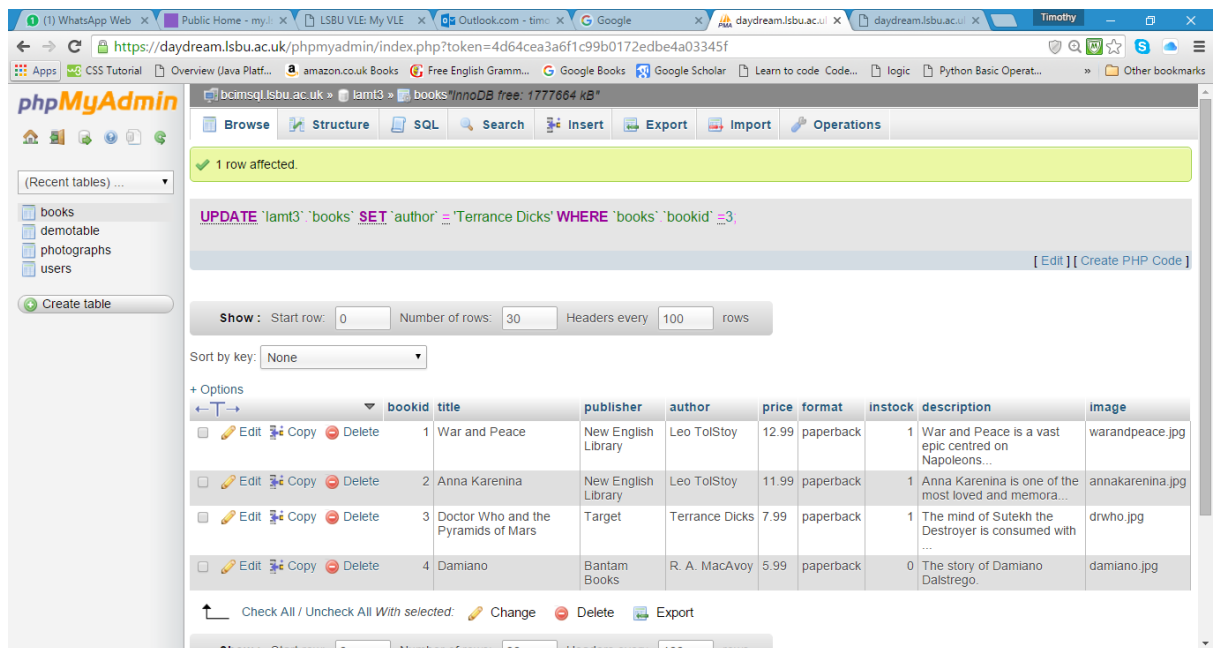
There were no exercises as I had to complete anything I did not complete and complete any extended task that I did not complete

2.7.1. Extended task: Exercise 2: Books

Books.php: <http://daydream.lsbu.ac.uk/myweb/lamt3/booksext.php>



The extended task I did was **Exercise 2: PHPMysqlAdmin** was to add an author fields to books table on **PHPMysqlAdmin**. This would allow the book table to match with the headings

Below I have added a field called author



This is a **books.php** where it shows the heading matching the fields

Books

| Book id | Book Title | Title | Publisher | Author | Price | In stock |
|--|---|-------------------------------------|---------------------|----------------|-------|----------|
|  1 | War and Peace | War and Peace | New English Library | Leo TolStoy | 12.99 | 1 |
|  2 | Anna Karenina | Anna Karenina | New English Library | Leo TolStoy | 11.99 | 1 |
|  3 | Doctor Who and the Pyramids of Mars | Doctor Who and the Pyramids of Mars | Target | Terrance Dicks | 7.99 | 1 |

This is the full code of extended task for **Exercise 2: PHPMyAdmin**

```
<?php
require_once "lib/dbutils.php";
?>
<html>
<body>
<h2 align="center">Books</h2>
<table border="1" width="100%">
  <tr>
    <td width="10%" align="center"><big><b>Book id </b></big></td>
    <td width="30%" align="center"><big><b>Book Title</b></big></td>
    <td width="20%" align="center"><big><b>Title</b></big></td>
    <td width="20%" align="centre"><big><b>Publisher</b></big></td>
    <td width="20%" align="centre"><big><b>Author</b></big></td>
    <td width="20%" align="center"><big><b>Price</b></big></td>
    <td width="20%" align="center"><big><b>In stock</b></big></td>
```

```

</tr>
<?php
    $pdo = connect();
    $results = $pdo->query("SELECT * FROM books", PDO::FETCH_ASSOC);
    foreach( $results as $row) {
        print("<tr>");
        print("<td><img src='images/'.".$row['image']."'height=30%'".".$row['bookid']."'."</td>");
        print("<td><a
href='bookview.php?bookid=".".$row['bookid']."'>".$row['title']."'."</a></td>");
        print("<td>".$row['title']."'."</td>");
        print("<td>".$row['publisher']."'."</td>");
        print("<td>".$row['author']."'."</td>");
        print("<td>".$row['price']."'."</td>");
        print("<td>".$row['instock']."'."</td>");

        print("</tr>");
    };
?>
</table>
</body>
</html>

```

This exercise was easy as I had to insert a field on PHPMYAdmin to match the heading this would make it easier for users to understand the webpage. I managed to complete the extended task for **Exercise 2: PHPMYAdmin** as I had no difficulties from doing the extended task for **Exercise 2: PHPMYAdmin**

I have learnt that you can use PHPMYAdmin as a database to store information. This information is useful as it makes it easier to understand what is in the database.

The exercise would help me understand things in Digital Business application as it will allow users to create a website which allows information like a database. This would allow them to use a website like a catalogue like other website use to store items so other customers can use their website to do online shopping.

2.7.2. Additional Task

Car Photo Album PHP and MySQL: <http://daydream.lsbu.ac.uk/myweb/lamt3/cars-photo.php>

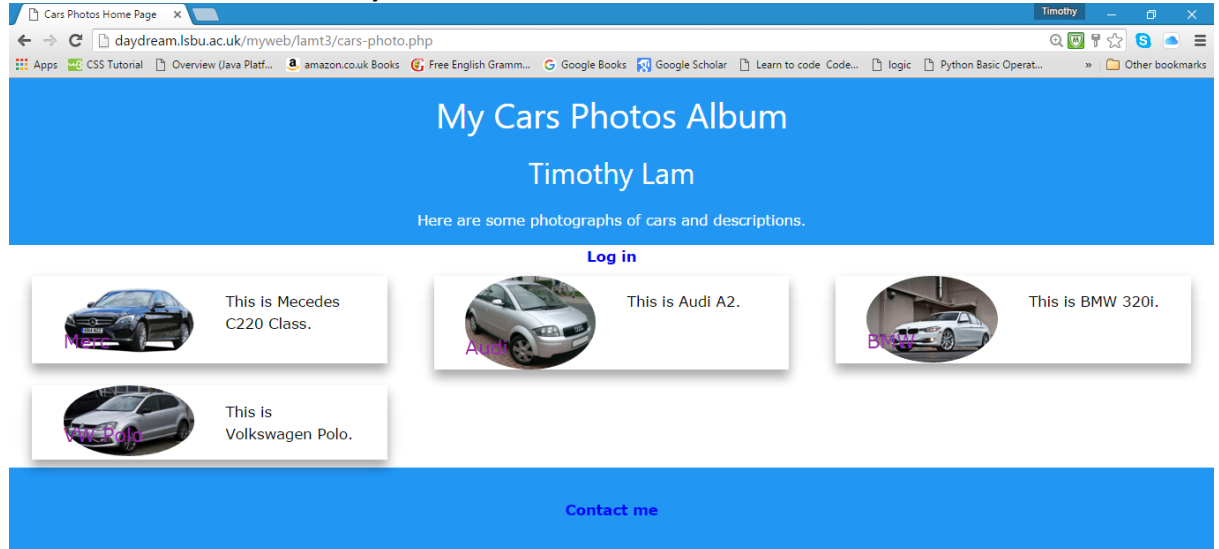
Car Photo Album JavaScript and Ajax: <http://daydream.lsbu.ac.uk/myweb/lamt3/cars-photo.html>

The extra work I did to investigate things I found interesting was to create my own version of Session and Authentication of what I have learnt and create my own photo album. I did not do any reading.

In addition, I have done more practice on creating my own photo album using the exercises I have applied

Below is a screenshot of the work I did for more practice on the task where firstly I created my own photo album using PHP & MySQL and the next one was using JavaScript and AJAX

Cars Photo Album PHP and MySQL



This the full code of Cars Photo Album PHP and MySQL

```
<!DOCTYPE html>
<?php
require_once( 'lib/dbutils.php' );
require_once("photoalbum-common.php");
?>
<html>
<head>
<title>Cars Photos Home Page</title>
<meta name='viewport' content='width=device-width, initial-scale=1'>
<link rel='stylesheet' href='photoalbum.css'>
<link rel='stylesheet' href='http://www.w3schools.com/lib/w3.css'>
<script src='http://code.jquery.com/jquery-1.11.3.min.js'></script>
<link rel='stylesheet' href='lib/lightbox2/css/lightbox.css'>
</head>
<body>
<header class='w3-container w3-blue w3-center'>
  <h1>My Cars Photos Album</h1>
  <h2>Timothy Lam</h2>
  <p>Here are some photographs of cars and descriptions.</p>
</header>

<?php
$pdo = connect();
loginOrOut( $pdo, 'users' );

if ( $_SESSION['user'] != "anonymous" ) {
  if ( isset( $_GET['deletionid'] ) ) {
    $errorMessage = deletePhotograph( $pdo, $_GET['deletionid'] );
    if ( $errorMessage != "" ) {
      print "<div class='errorMessage'>$errorMessage</div>\n";
    } else {
      print "<div class='message'>Image deleted.</div>\n";
    }
  }

  if ( isset( $_POST['formid'] ) && $_POST['formid'] == 'fileupload' ) {
    $photoid = addPhotograph( $pdo, $_POST['name'], $_POST['description'], 'self',
$_FILES['uploadfile'] );
  }
}

print "<div class='w3-container w3-center'>\n";
if ( $_SESSION['user'] == "anonymous" ) {
```

```

    print "    <a href='#id02' class='link'>Log in</a>\n";
} else {
    print "    User: ".$_SESSION['user']." Role: ".$_SESSION['role']." <a href='?logout=true'
class='link'>Log out</a>\n";
}
print "</div>\n";

$sql = "SELECT * FROM `carphotographs` ORDER BY `photoid`";
$stmt = $pdo->query( $sql);
// get all rows in an array
$rows = $stmt->fetchAll(PDO::FETCH_ASSOC);

$numberOfPhotos = count( $rows);

for( $i = 0; $i < $numberOfPhotos; $i++) {

    if ( $i % 3 == 0 ) {
        if ( $i > 0 ) { print "</div>\n"; }
        print "<div class='w3-row'>\n";
    }

    $row = $rows[$i];
    print "    <div class='w3-container w3-row w3-col 14'>\n";
    print "        <div class='w3-container w3-card-8 w3-margin-8'>\n";
    print "            <div class='w3-container w3-half w3-image'>\n";
    print "                <a class='example-image-link' data-lightbox='example-1'
href='images/' . $row['image'] . "'><img src='images/' . $row['image'] . "' class='example-image w3-
circle'></a>\n";
    print "                <div class='w3-title w3-text-purple'>".$row['name']. "</div>\n";
    print "            </div>\n";
    print "            <div class='w3-container w3-half'>\n";
    print $row['description']. "\n";
    if ( $_SESSION['user'] != "anonymous") {
        print "                <a href='?deletionid=".$row['photoid'] . "' class='link'>(Delete)</a>";
    }
    print "            </div>\n";
    print "        </div>\n";
    print "    </div>\n";

    print "</div>\n";

    print "<div class='w3-container w3-center'>";
    if($_SESSION['user'] != "anonymous") {
        print "    <a href='#id01' class='link'>Add images</a>";
    }
    print "</div>";

    ?>

<div id="id01" class="w3-modal">
    <div class="w3-modal-dialog">
        <div class="w3-modal-content w3-card-4">
            <header class="w3-container w3-teal">
                <a href="#" class="w3-closebtn">&times;</a>
                <h2>Upload photograph</h2>
            </header>
            <div class="w3-container">
                <form action='?' method='post' enctype='multipart/form-data' id='photoform'
class='w3-container'>
                    <div class="w3-group">
                        <input type='file' name='uploadfile' class='w3-input'>
                        <label class="w3-label">Image file</label>
                    </div>
                    <div class="w3-group">
                        <input type='text' name='name' class='w3-input' required>
                        <label class="w3-label">Name</label>
                    </div>
                    <div class="w3-group">
                        <textarea name='description' class='w3-input' required></textarea>
                        <label class="w3-label">Description</label>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>

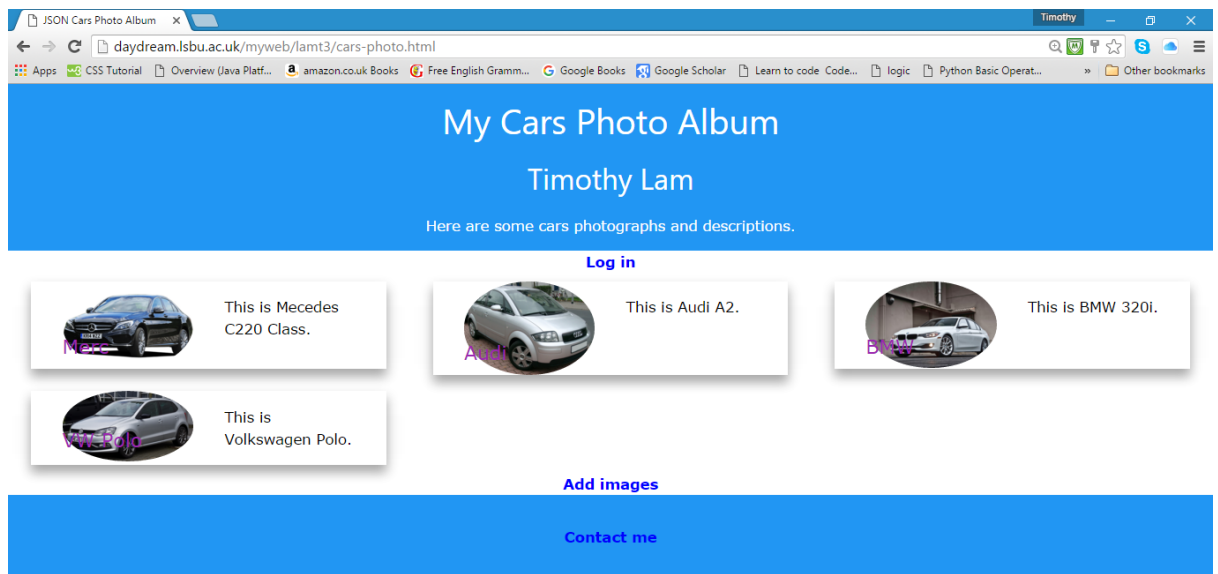
```

```

        </div>
        <input type='hidden' name='formid' value='fileupload'>
        <input type='submit' value='Upload' id='uploadbutton' class='w3-btn'>
    </form>
</div>
<div>
    <div class="w3-container w3-teal">
        <p></p>
    </div>
</div>
<div id='id02' class='w3-modal'>
    <div class='w3-modal-dialog'>
        <div class='w3-modal-content w3-card-4'>
            <header class='w3-container w3-teal'>
                <a href='#' class='w3-closebtn'>&times;</a>
                <h2>Log in</h2>
            </header>
            <div class='w3-container'>
                <form action='?' method='post' id='loginform' class='w3-container'>
                    <div class='w3-group'>
                        <input type='text' name='username' size='16' class='w3-input'>
                        <label class='w3-label'>Username</label>
                    </div>
                    <div class='w3-group'>
                        <input type='password' name='password' size='16' class='w3-input'>
                        <label class='w3-label'>Password</label>
                    </div>
                    <input type='hidden' name='formid' value='login'>
                    <input type='submit' value='Log in' id='loginbutton' class='w3-btn'>
                </form>
            </div>
            <div class='w3-container w3-teal'>
                <p></p>
            </div>
        </div>
    </div>
</div>
<div>
    <div class='w3-container w3-blue w3-padding-jumbo'>
        <div class='w3-container w3-center'>
            <a href='mailto:timothylam92@hotmail.co.uk' class='link'>Contact me</a>
        </div>
    </div>
</div>
<script src='lib/lightbox2/js/jquery.js'></script>
<script src='lib/lightbox2/js/lightbox.js'></script>
</body>
</html>

```

Cars Photo Album using JavaScript and AJAX



This the full code of Cars Photo Album JavaScript and AJAX

```
<!DOCTYPE html>
<html>
<head>
<title>JSON Cars Photo Album</title>
<meta name='viewport' content='width=device-width, initial-scale=1'>
<link rel='stylesheet' href='http://www.w3schools.com/lib/w3.css'>
<link rel='stylesheet' href='lib/lightbox2/css/lightbox.css'>
<link rel='stylesheet' href='photoalbum.css'>

<script src='http://code.jquery.com/jquery-1.11.3.min.js'></script>
<script src='lib/ajaxupload/jquery-fileupload.min.js'></script>

<script>
var user = "anonymous";
var role = "anonymous";

function createPhotoHTML( photoInfo) {
  return " <div class='w3-container w3-row w3-col 14'>" +
    "   <div class='w3-container w3-card-8 w3-margin-8'>" +
    "     <div class='w3-container w3-half w3-image'>" +
    "       <a class='example-image-link' data-lightbox='example-1' href='images/" +
photoInfo.image + "'><img src='images/" + photoInfo.image + "' class='example-image w3-
circle'></a>" +
    "     <div class='w3-title w3-text-purple'>" + photoInfo.name + "</div>" +
    "   </div>" +
    "   <div class='w3-container w3-half'>" + photoInfo.description +
    "     // next line should be conditional on whether logged in or not
    (( user != "anonymous") ? "<span class='link deletelink' id='" + photoInfo.photoid
+ "'>(Delete)</span>" : "" ) +
    "   </div>" +
    " </div>";
}

function loadPhotos() {
  $.getJSON( "json-carphotoalbum-photos.php",
    function( photos) {
      $('#album').empty();
      for (i = 0; i < photos.length; i += 3) {
        var rowDiv = $( "<div class='w3-row'>");
        $('#album').append( rowDiv);
        for( j = i; j < i + 3; j++ ) {
          if ( j < photos.length) {
            rowDiv.append( createPhotoHTML( photos[j]));
          }
        }
      }
    }
  )
  // add listener to all deletelinks
}
```



```

                                $('.deletelink').click(
                                function( event) {
                                    //alert('Photo id: ' + event.target.id);
                                    deletePhoto(event);
                                }
                                );
                            }
                        );
                    }
                }
            }
        }

        function deletePhoto( event) {
            $.post( "json-carphotoalbum-delete.php",
                "deletionid=" + event.target.id,
                function( data) {
                    if ( data.result != "success") {
                        alert( "ERROR: " + data.error);
                    } else {
                        loadPhotos();
                    }
                }
            );
        }

    }

    function renderUserInformation() {
        $('#userinfo').empty();
        if ( user == "anonymous") {
            $('#userinfo').append( "<a href='#id02' class='link'>Log in</a>");
            $('#addlink').hide( "<a href='#id01' class='link'>Add images</a>");
        } else {
            $('#userinfo').append( "User: " + user + " Role: " + role + " <a href='#' class='link'
onclick='logout()'>Log out</a>");
            $('#addlink').show( "<a href='#id01' class='link'>Add images</a>");
        }
    }

    function login() {
        $.post( "json-photoalbum-logon.php",
            $('#loginform').serialize(),
            function( data) {
                user = data.user;
                role = data.role;
                renderUserInformation();
                loadPhotos();
                $('#loginform').get(0).reset();
                location.href='#';
                alert( data.action + " " + data.user + " " + data.role);
            }
        );
    }

    function logout() {
        $.getJSON( "json-photoalbum-logon.php?logout=true",
            function( data) {
                user = data.user;
                role = data.role;
                renderUserInformation();
                loadPhotos();
                location.href='#';
                //alert( data.action + " " + data.user + " " + data.role);
            }
        );
    }

    $( document).ready(
        function() {
            loadPhotos();
            $('#photoform').fileUpload({
                beforeSubmit :
                    function(uploadData){ console.log(uploadData); return true; },
                // access the data returned by the upload return false to stop the submit ajax
call
                success :
                    function(data, textStatus, jqXHR){ },
                    // Callback for the submit success ajax call
                error :
                    function(jqXHR, textStatus, errorThrown){ alert( "error"); console.log(jqXHR); },
                    // Callback if an error happens with your upload call or the submit call
            });
        }
    );

```

```

        complete :
            function(jqXHR, textStatus){ $('#photoform').get(0).reset(); location.href='#';
loadPhotos(); }
        // Callback on completion
        });

        $('#loginform').submit(
            function( event) {
                event.preventDefault();

                login();
            }
        );
    }
);

</script>

</head>
<body>

<header class='w3-container w3-blue w3-center'>
    <h1>My Cars Photo Album</h1>
    <h2>Timothy Lam</h2>
    <p>Here are some cars photographs and descriptions.</p>
</header>

<div class='w3-container w3-center' id='userinfo'>
<a href="#id02" class='link'>Log in</a>
</div>

<div id='message'>

</div>

<div id='album'>

</div>

<div class='w3-container w3-center' id='addlink'>
<a href="#id01" class='link'>Add images</a>
</div>

<div id="id01" class="w3-modal">
    <div class="w3-modal-dialog">
        <div class="w3-modal-content w3-card-4">
            <header class="w3-container w3-teal">
                <a href="#" class="w3-closebtn">&times;</a>
                <h2>Upload photograph</h2>
            </header>
            <div class="w3-container">
                <form action='json-photoalbum-upload.php' method='post' enctype='multipart/form-
data' id='photoform' class='w3-container'>
                    <div class="w3-group">
                        <input type='file' name='uploadfile' class='w3-input'>
                        <label class="w3-label">Image file</label>
                    </div>
                    <div class="w3-group">
                        <input type='text' name='name' class='w3-input' required>
                        <label class="w3-label">Name</label>
                    </div>
                    <div class="w3-group">
                        <textarea name='description' class='w3-input' required></textarea>
                        <label class="w3-label">Description</label>
                    </div>
                    <input type='hidden' name='formid' value='fileupload'>

                    <input type='submit' value='Upload' id='uploadbutton' class='w3-btn'>
                </form>
            </div>
            <footer class="w3-container w3-teal">
                <p></p>
            </footer>
        </div>
    </div>
</div>

```

```

    </div>
</div>

<div id='id02' class='w3-modal'>
  <div class='w3-modal-dialog'>
    <div class='w3-modal-content w3-card-4'>
      <header class='w3-container w3-teal'>
        <a href='#' class='w3-closebtn'>&times;</a>
        <h2>Log in</h2>
      </header>
      <div class='w3-container'>
        <form action='json-photoalbum-logon.php' method='post' id='loginform' class='w3-
container'>
          <div class='w3-group'>
            <input type='text' name='username' size='16' class='w3-input'>
            <label class='w3-label'>Username</label>
          </div>
          <div class='w3-group'>
            <input type='password' name='password' size='16' class='w3-input'>
            <label class='w3-label'>Password</label>
          </div>
          <input type='hidden' name='formid' value='login'>
          <input type='submit' value='Log in' id='loginbutton' class='w3-btn'>
        </form>
      </div>
      <div class='w3-container w3-teal'>
        <p></p>
      </div>
    </div>
  </div>
</div>
<div>
<div class='w3-container w3-blue w3-padding-jumbo'>
  <div class='w3-container w3-center'>
    <a href='mailto:lamt3@lsbu.ac.uk' class='link'>Contact me</a>
  </div>
</div>
</div>
<script src='lib/lightbox2/js/lightbox.js'></script>
</body>
</html>

```

I learnt from doing extra work that this will help me build a secure website so the users must have log in details to have full control of the website and if the users haven't got log in details then they will have limited control when using the website

These exercise had help me understand that Session and Authentication are very useful for every website as it will protect other users from messing around with the website or other users' details on the website. What is more if the users have their own login detail then they can do what they want on they own the account. This would be safer as each user will have their own access to their own account and the users would not be able to access other users' accounts

3. Conclusion

The exercises which I found most challenging were the PHP and MySQL and the extended task. These exercises are the following: **Exercise 1: MySQL, Exercise 2: PHPMyAdmin, Exercise 3 Photo Album Server Version B and C, Exercise 4 Photo Album Server Version D and Exercise 6: Photo Album Server Version E**. Additionally another exercise I found challenging was JavaScript and Ajax Exercise Extended task. These are the following: **Exercise 1: Photo Album Client-Rendered Version A**. This is because some of the extended tasks on PHP and MySQL and extended tasks for JavaScript and AJAX I found too challenging. The parts that I found most interesting were the exercises PHP and MySQL this includes knowing how to use Log in session in one of the exercises which was **Exercise 5: Session and Authentication** and how to apply the Log in Session to **Exercise 6 Photo Album Server Version E**. Furthermore another interesting part I found on JavaScript and Ajax was applying the log in session from PHP and MySQL Exercises to JavaScript and AJAX Exercises. This would only allow users to have access to login to have control of the website. The reason for this is that it was the first time going over these areas as I only know the basics of HTML. The only things that I still do not understand fully is how to use MySQL to build a website and store a website as I never came across MySQL but I have heard about MySQL.