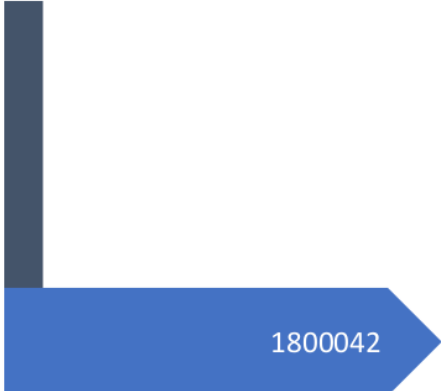


smrtcon1800042

by Kaylie Shoulder

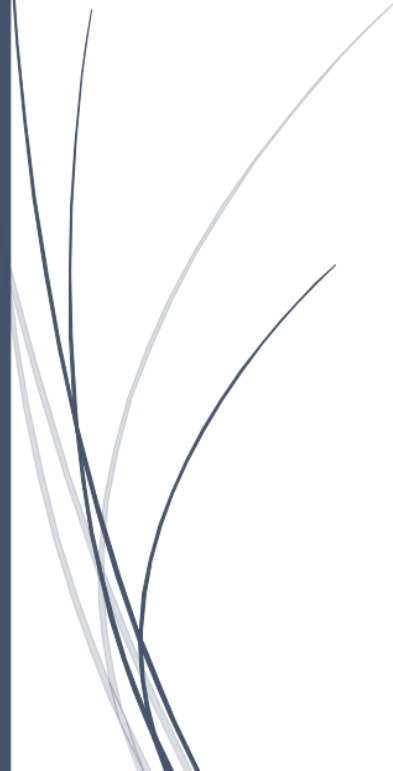
FILE	10677_KAYLIE_SHOULDER_SMRTCON1800042_236131_153258853.DOCX (5.05M)		
TIME SUBMITTED	06-MAR-2020 09:50AM (UTC+0000)	WORD COUNT	1025
SUBMISSION ID	121366919	CHARACTER COUNT	4768



1800042

Smart and Connected Devices

Web Based Assignment + Report



Kaylie Shoulder
UWTSD

Smart and Connected Devices Report

For web applications, especially those that include database management, PHP accompanies HTML and CSS in a very complimentary way. The aim of the assignment was to create a database which allowed three sensor's rainfall data to be collected and transferred into it – equip with good security measures, and a clear and concise way to display the recorded data.

Throughout the course of this particular report, all technicalities regarding the web aspect of this assignment will be analysed and discussed, touching upon why the application was approached in the way that it was.

The first page the user sees is the 'landing page' which is the page that allows users to sign in, or rejects those that don't have any correct login information.

Here is an image of the landing page:



Welcome to the Portal!

This is the login portal that will allow you to access the database information.

Username:

Password:

As you can see, the website welcomes the user and asks them to log in to access information. This page has been configured using validation, which was done in order to prevent those without permissions from accessing the information, and those who do have the rights to access the information, the permissions to sign in.

The result of the incorrect credentials being submitted:

Sorry, wrong details
[Try again](#)

The result of the correct credentials being submitted:

Thank you for logging in!
[Go to my secret page](#)

When the correct credentials are submitted, this screen displays and from there, the user can click on the link, and there they are able to access the data information.

Here is a screenshot of the code that was used in order to create this page:

```
<?php session_start();?>
<html>
<head>
</head>
<body>
<?if(isset($_SESSION["uname"])!= "")//logged in
<?
<echo "Welcome " . $_SESSION["uname"] . "</h1>"; //welcome the user with their own name
<echo "<a href='logout.php'>Log out</a>"; //when clicked, log out
<?
<?else//not logged in
<?
<h1>Welcome to the Portal! </h1>
<p> This is the login portal that will allow you to access the database information. </p>
<form action="checklogin.php" method="post">
Username: <input type="text" name="username"><br>
Password: <input type="password" name="password"><br><br>
<input type="submit" value="Submit">
</form>
<?>
</body>
</html>
```

As you can see, the code above emphasises on allowing or denying the user access to their information, using the method of a form to submit user input, thereon validating the information provided.

Once the user has signed in, from the picture above, you can see that the user can then forward themselves to a link which states 'Go to my secret page'.

The link forwards the user to a page which displays the data that was collected from all three text files separately – allowing data interpretation to be a much easier task. Here is an image of what the 'secret' page looks like:



As you can see from the two images attached above, once the user has signed in they are able to view the data on the table in the format of a polygon graph – this particular type of graph was chosen as it was the best suited for the set of data that was stored in the database. It is compared using the amount of rainfall and the date in which the rain occurred. The reason that the graphs were implemented in such a way was because it is much easier for the reader to interpret the data, it is clear and concise and you are able to clearly compare the differences between the three data sets that were uploaded onto the database.

Attached is the code which was used in order to allow this website to function

correctly:

```
<?
include_once ('phpMyGraph5.0.php'); // Link to library file phpMyGraph 5.0

$cfg ['title'] = 'Rainfall Graph'; //Configure, title, width and height
$cfg ['width'] = 700; //number of pixels horizontally
$cfg ['height'] = 300; //number of pixels vertically

include "conninfo.php";
$query="SELECT * FROM rainfalldata WHERE sensor = 1 ORDER BY 'thedata' " ;
$result=mysql_query($link,$query); //get data from database

$data=array();

while ($r=mysql_fetch_array($result))
{
    $data[$r["thedata"]]= $r["mydata"]; //put data into array
}

header ("Content-type: image/png");
$graph = new phpMyGraph();
$graph->parseVerticalPolygonGraph ($data, $cfg); //create graph
$cfg['average-line-color']='#FF0000'
?>
```

```
<?
include_once ('phpMyGraph5.0.php'); // Link to library file phpMyGraph 5.0

$cfg ['title'] = 'Rainfall Graph'; //Configure, title, width and height
$cfg ['width'] = 700; //number of pixels horizontally
$cfg ['height'] = 300; //number of pixels vertically

include "conninfo.php";
$query="SELECT * FROM rainfalldata WHERE sensor = 2 ORDER BY 'thedata' " ;
$result=mysql_query($link,$query); //get data from database

$data=array();

while ($r=mysql_fetch_array($result))
{
    $data[$r["thedata"]]= $r["mydata"]; //put data into array
}

header ("Content-type: image/png");
$graph = new phpMyGraph();
$graph->parseVerticalPolygonGraph ($data, $cfg); //create graph
$cfg['average-line-color']='#FF0000'
?>
```

```
<?
include_once ('phpMyGraph5.0.php'); // Link to library file phpMyGraph 5.0

$cfg ['title'] = 'Rainfall Graph'; //Configure, title, width and height
$cfg ['width'] = 700; //number of pixels horizontally
$cfg ['height'] = 300; //number of pixels vertically

include "conninfo.php";
$query="SELECT * FROM rainfalldata WHERE sensor = 3 ORDER BY 'thedata' " ;
$result=mysql_query($link,$query); //get data from database

$data=array();

while ($r=mysql_fetch_array($result))
{
    $data[$r["thedata"]]= $r["mydata"]; //put data into array
}

header ("Content-type: image/png");
$graph = new phpMyGraph();
$graph->parseVerticalPolygonGraph ($data, $cfg); //create graph
$cfg['average-line-color']='#FF0000'
?>
```

The three graphs use the same base code, changing only one thing every single time and that is the line that states '\$query=*SELECT * FROM rainfalldata WHERE sensor = ____ ORDER BY "thedata" ';

For each graph the value after 'sensor =' changes from 1, to 2 or 3 – allowing each sensor addition to be separated and have their own graph. The way in which the graph should display is stated first, then telling the database where to retrieve the

Kaylie Shoulder 1800042

information, onto the drawing/creation of the graph. The graphs that are drawn are transferred to display on the secret website using the following code:

```
</head>
<body>
<a href="landing.php">Logout of Portal </a> <br>
<a href="entersearch.php">Search Database Content </a> <br>
<a href="transferdata.php">Click here to transfer data </a> <br>
<a href="transferdata2.php"> Transfer datasheet 2 </a> <br>
<a href="transferdata3.php"> Transfer datasheet 3 </a> <br>
<h1>Rainfall Monitoring Graphing Page </h1>
<h1>Attached Below are the Graphs of the Rainfall Monitoring Process </h1>
<br><br>
The Data Below was taken from 3 rain sensors </br>
<p> The information you can see below allows you to see the maximum and
<p> Graph Highlighting the Data Collected from Sensor 1 </p>

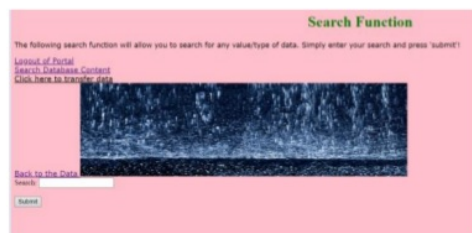
<p> Graph Highlighting the Data Collected From Sensor 2 </p>

<p> Graph Highlighting the Data Collected From Sensor 3 </p>

<br><br>
<a href="landing.php">Log out of Portal </a> <br>
</body>
</html>
</?>
```

This page essentially allows user navigation using the <a href tags, and then inserts the graphs using and so on.

The website also has a page which allows the user to search the database which unfortunately does not work as well as hoped. Here is an image of how the search page looks:



The user can enter something into the search bar and hit the 'search' button, which will then put out the results.



This is the result of searching '9', this is poorly organised and splurges out all the data that the database can find rather than refining it as it should have been done. Here is the code behind these pages:

```
<!-- Search Function -->
<p> The following search function will allow you to search for any value/type of data
<head> <link rel="stylesheet" href="stylesheet.css">
<a href="landing.php">Logout of Portal </a> <br>
<a href="entersearch.php">Search Database Content </a> <br>
<a href="transferfromsensor.php">Click here to transfer data </a> <br>
<a href="secret.php"> Back to the Data </a>

</head>
<body>
<form action="showdatabase.php" method="post">
Search: <input type="text" name="mysearch"><br><br>
<input type="submit">
</form>
</body>
</html>
```

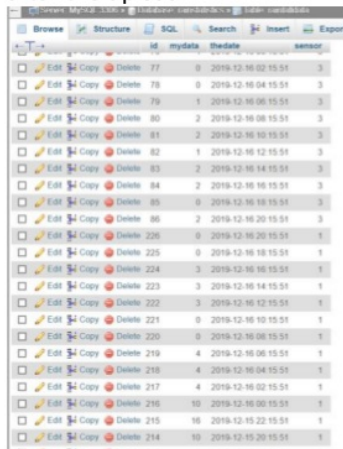
```
<?php
include "conninfo.php"; //make connection info part of this page
$query="SELECT * FROM rainfalldata ORDER BY sensor";
$result=mysqli_query($link,$query);
while($row=mysqli_fetch_array($result))
{
echo $row["id"]. " ";
echo $row["mydata"]. " ";
echo $row["thedata"]. " ";
echo $row["sensor"]. " ";
}
}
<html>
<a href="landing.php">Logout of Portal </a> <br>
<a href="entersearch.php">Search Database Content </a> <br>
<a href="transferfromsensor.php">Click here to transfer data </a> <br>
<a href="secret.php"> Back to the data </a>
</html>
```

This search method was used in the hopes of providing the user with an easy way to view any data, however it was poorly programmed and lacks the use of radio buttons, instead focusing on simply splurging the data onto the screen, but sorting it into the date that the data was obtained/uploaded rather than have any refined/filtered results.

As a part of the links, the user is able to click on 'Transfer Data Sheet' and this will then upload the data from the linked text file to the database. Unfortunately, at the moment the code only uploads new data and does not replace the previous. Here is what happens when this link is clicked:

Data has been transferred

And the database looks as follows after the upload:



	id	mydata	thedata	sensor
☐	77	0	2019-12-16 02:15:51	3
☐	78	0	2019-12-16 04:15:51	3
☐	79	1	2019-12-16 06:15:51	3
☐	80	2	2019-12-16 08:15:51	3
☐	81	2	2019-12-16 10:15:51	3
☐	82	1	2019-12-16 12:15:51	3
☐	83	2	2019-12-16 14:15:51	3
☐	84	2	2019-12-16 16:15:51	3
☐	85	0	2019-12-16 18:15:51	3
☐	86	2	2019-12-16 20:15:51	3
☐	226	0	2019-12-16 20:15:51	1
☐	225	0	2019-12-16 18:15:51	1
☐	224	3	2019-12-16 16:15:51	1
☐	223	3	2019-12-16 14:15:51	1
☐	222	3	2019-12-16 12:15:51	1
☐	221	0	2019-12-16 10:15:51	1
☐	220	0	2019-12-16 08:15:51	1
☐	219	4	2019-12-16 06:15:51	1
☐	218	4	2019-12-16 04:15:51	1
☐	217	4	2019-12-16 02:15:51	1
☐	216	10	2019-12-16 00:15:51	1
☐	215	10	2019-12-15 22:15:51	1
☐	214	10	2019-12-15 20:15:51	1

The code simply adds rather than deletes and adds:

```
<?php
include "conninfo.php"; //connect to database
$myfile="rainfall.txt"; //name of file

$filelink=fopen($myfile,"r") or die("Unable to open file!");
$data=fread($filelink,filesize($myfile)); //read entire file
fclose($filelink); //close the file

$incoming=array();
$incoming=explode(chr(13),$data);

$length=sizeof($incoming); //how big is the array?
$tempdata=array();

for($x=0;$x<$length;$x++)
{
    $incoming[$x]=trim($incoming[$x]); //clean off the linebreak
    $tempdata=explode(" ",$incoming[$x]); //split current line of
    $timeofreading=$tempdata[1]. " ". $tempdata[2]; //stick timestamp
    $rainfall=trim($tempdata[0]); //rainfall reading

    $query="INSERT INTO rainfalldata(mydata,thedata,sensor)
    VALUES($rainfall,$timeofreading,1)";

    $result=mysqli_query($link,$query);
    echo "Data has been transferred";
}
```

The code aims to connect to the database, and access/read the text file that has been called. It will detect the size of the file, add a timestamp and then add the information onto the database. As of yet, it does not delete prior information.

When the user logs out they cannot access any information, as you can see below:

You are not currently logged into the portal. Please sign in to access your information.

[Back to login page](#)

This is the code used to create the login/logout system:

```
<?session_start(); //start or access existing session
include "conninfo.php";
$un=$_POST["username"];
$pw=$_POST["password"];

$query="SELECT * FROM login WHERE username like '$un'
AND password like '$pw'";

$result=mysqli_query($link,$query);
$num=mysqli_num_rows($result); //how many rows found?

if($num==1) //log person in
{
    $_SESSION["uname"]=$un; //set session
    echo "Thank you for logging in!<br><br>";
    echo "<a href='\"secret.php\"'>Go to my secret page</a>";
}
else //don't log them in
{
    echo "Sorry, wrong details?>";
    <br>
    <a href="landing.php">Try again</a>
}
?>
```

This code checks the credentials and states whether they are right/wrong, and will reject/grant the user access to the information.

```
<?session_start();
$_SESSION["uname"]=""; //set to empty
session_destroy(); //remove session
header("location:landing.php");
?>
```

This is what happens when the user selects 'logout' – the session is terminated. If the user is not logged into the portal they won't be able to access any information on the database.

Kaylie Shoulder 1800042

To summarise, the way in which this application was approached was to allow the user to use the application easily and interpret the data in the easiest manner – whilst providing the application with the right means of security. In some ways, it has accomplished this aim, and in other ways has failed to meet this basic criteria.

FINAL GRADE

58/100

GENERAL COMMENTS

Instructor

Data management - You have transferred the data albeit in a copy and pasted way, it would be more efficient/expandable using a loop or function. It would also be better to do this in one file rather than have separate ones to make daily data transfer more efficient. You also need to delete from the file. Some of the fields you've chosen are a little long eg rainfall at 11 - 3 would be a more appropriate choice for rainfall!

5/10

Range of HTML/PHP coding demonstrated - Login is implemented well. There are searches implemented, it would be better to have searches where you can filter by more than one attribute at a time eg sensor X on date X rather than one search looking in every field. Presentation of data needs some break between them. You could also have provided some stats eg wettest day

14/25

Visualisation of data using graphs - Graphs are hardcoded and embedded as an image. For greater functionality you could allow the user to filter by using a HTML form to pass a parameter to the graph.10/15

Overall integration of website - A bit of a basic layout, but looks OK. You could use includes to make the production of the site more efficient rather than copying and pasting code to every page.

6 /10

Knowledge and understanding overall - A reasonable

understanding was demonstrated, but reflect on the points raised above

12/20

Report - All academic reports should have an Introduction and a Conclusion, and should be organised in numbered sections. You have gone through what you did, but you need more on why you did it - a critical appraisal. Was the approach you took successful, how would you change your approach if you did the project again, what features would you include in the future etc. You say in your conclusion that you have accomplished the aim in some ways and not others - give detail on this!

11/20

Overall - 58%

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6
