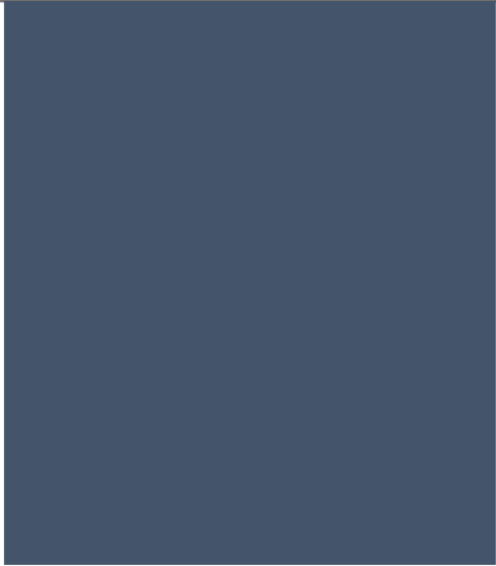


1908090

by Michael Simmonds

FILE	18297_MICHAEL_SIMMONDS_1908090_236131_955403200.DOCX (1.09M)		
TIME SUBMITTED	28-FEB-2020 02:03PM (UTC+0000)	WORD COUNT	1038
SUBMISSION ID	120909253	CHARACTER COUNT	4914



Smart and connected devices 1

1908090

Visualisation of IoT data



Contents

Visualisation of IoT data.....	0
Introduction	2
Security	3
Importing data	5
Viewing, searching and filtering data	6
Visualisations/Graphs	7
Averages.....	8
Test case diagram	0
Conclusion.....	1
Bibliography	3

Introduction

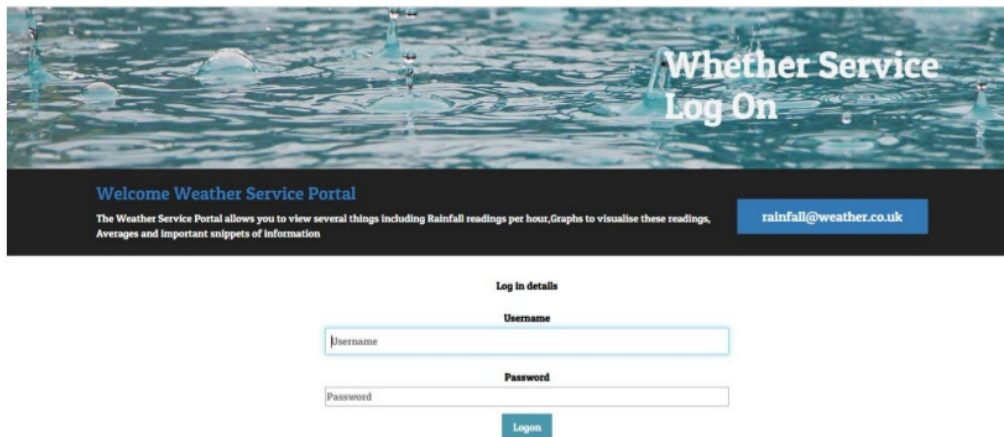
The developer was asked to create a web user-interface that a user is able to view and interact with data which has been read from three weather sensors. This data would need to be imported in to a MYSQL database using PHP.

In this report you will learn how the developer accomplished the build including what features the developer decided to implement, how the build went and also how the developer tested the web interface in terms of a test plan.

Security

The first feature the developer had to investigate was how the web-interface could be secured and how to allow only authorised personnel to view and interact with the weather data.

The developer chose to implement a log in system where each authorised personnel would have a username and password in order to access the interface.



The screenshot shows the 'Weather Service Log On' interface. At the top, there's a header with the text 'Whether Service Log On' over a background of water ripples. Below this, a dark blue banner contains the text 'Welcome Weather Service Portal' and a description: 'The Weather Service Portal allows you to view several things including Rainfall readings per hour, Graphs to visualise these readings, Averages and important snippets of information'. To the right of the banner is a blue button with the email 'rainfall@weather.co.uk'. Below the banner, the 'Log in details' section contains two input fields: 'Username' and 'Password'. The 'Username' field has a placeholder text 'Username' and the 'Password' field has a placeholder text 'Password'. A blue 'Login' button is positioned below the password field.

Session variables were used in order to secure the interface. Session variables are used to store individual client's information on the web server for later use, as a web server does not know which client's request to be respond because HTTP address does not maintain state. [1]

In this case the developer used a session variable to store the username of the user, when the user moves around the system each page checks that the session variable is still active to make sure that the user is still logged in.

If the user is trying to get to a page but is not logged in validation comes in to play and directs the user back to the log in screen.

When the logout button is clicked a PHP script is run and kills the session so the user will need to log in again.

Login Script

```
<?php
include 'conninfo.php';
session_start();

$myusername= $_POST["username"];
$mypassword= $_POST["password"];

$query = "SELECT * FROM login WHERE username='$myusername' AND password='$mypassword'";
$result=mysqli_query($link,$query);
$num=mysqli_num_rows($result);

if ($num == 1)
{
    $_SESSION['user_logged'] = $_POST['username'];
    unset($myusername);
    unset($mypassword);
    echo "<h3>Your password is correct</h3>";
    header('Location:index.php');
    die();
}

else {
    echo "<h3>The username / password combination entered is incorrect!</h3>";
    unset($myusername);
    unset($mypassword);
    header("Refresh: 1;URL=login.html");
}

$link = null;
?>
```

Session checking on each page

```
<?php

if(isset($_SESSION['user_logged'])) {
    //do nothing!
} else {
    echo "<h2>You are not currently logged in.</h2>";
    echo "<h2>We are redirecting you.</h2>";
    header("Refresh: 3;URL=login.php");
    die();
}

?>
```

Importing data

The data collected from each sensor needed to be imported in to a MYSQL database via a PHP script.

Each page of the interface the developer created has a button which when clicked will update the data within the MySQL database with the most up to date data from the text files generated by the weather sensor.

The PHP script opens the text file, reads in the data, closes the file, uses an array to store the data, then uses a for loop to split the data as well as putting it back together, then the data is input in to the table within database using an insert statement and then clears the text file to stop duplicate readings.

Latest Rainfall Data

On this page you will be able to view, search and filter the latest rain fall data collected by our high tech sensors. If you would like to make sure that this is the latest data please go to the update rainfall data page in import our latest files.

Update rainfall data

```
include "conninfo.php"; //connect to database
$myfile="rainfall1.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 data that has been read into 3
    $timeofreading=$tempdata[1]. " " . $tempdata[2]; //stick timestamp back together in one variable ($timeofreading)
    $rainfall=trim($tempdata[0]); //rainfall reading

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

    $result=mysqli_query($link,$query);
}

$filelink = fopen($myfile, "w") or die("Unable to open file!"); //w effectively wipes file
fclose($filelink); //close link to file

echo "Data has been transferred"; header("Refresh: 1;URL=rainfall_readings.php");
```

When the button is clicked, and the data transferred the user is alerted if the transfer has been successful. One feature that the developer wants to implement in the future is an alert to the user if there isn't any data within the text files to transfer.

Viewing, searching and filtering data

One of the prerequisites of the assignment was that the user needs to be able to view the data output by the weather sensors.

The interface that was created allows the user to view all data within a table, the user is also able to filter the data by clicking on the title of the column, this was achieved by implementing a java script.

If the user wants to narrow down the data before viewing it, the developer has implemented a search function where the user can pick which sensor and what date they want to view the data of.

This was implemented by using a dropdown and free text box where the value the user enters get put in to a PHP variable and then the variable is used in a SQL statement to narrow down the data.

Please choose from the following options in order to narrow down the data within the table below:

Please Select Sensor dd/mm/yyyy

ID	Rainfall	Date	Sensor
450	2	2019-12-16 20:15:51.282545	3
449	0	2019-12-16 18:15:51.282545	3
448	2	2019-12-16 16:15:51.282545	3
447	2	2019-12-16 14:15:51.282545	3
446	1	2019-12-16 12:15:51.282545	3
445	2	2019-12-16 10:15:51.282545	3
444	2	2019-12-16 08:15:51.282545	3

```
include "conninfo.php";
if(isset($_POST['sensor'])) {
$query = $_POST['sensor'];
$query="SELECT * FROM rainfallreadings WHERE sensor = '$query'";
$result=$link-> query($query);

if ($result-> num_rows >0) {
while($row = $result-> fetch_assoc()){
echo "<tr>";
echo "<td style='text-align:center'>".$row['id']. "</td>";
echo "<td style='text-align:center'>".$row['mydata']. "</td>";
echo "<td style='text-align:center'>".$row['thedata']. "</td>";
echo "<td style='text-align:center'>".$row['sensor']. "</td>";
}
echo "</table>";
}
else{
}
}
$link-> close();
```

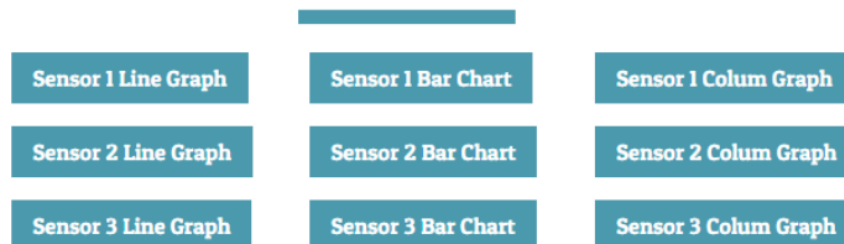
Visualisations/Graphs

The developer wanted to make the data visual in terms of graphs, when scripting the graphs, the developer used a library to help, there are many libraries available that produce graphs but the developer chooses phpMyGraph5.0.

To give the user a good choice of graphs to use, 3 graphs were produced for each sensor:

- Vertical Column Graph
- Line Graph
- Horizontal Column Graph

Nine buttons were placed on the visualisation page where the user can pick which one to choose, when the user selects one of these buttons the PHP script (below) which includes the phpMyGraph library is run and gives the user an up to date visual representation of the relevant data.



```
include_once ('phpMyGraph5.0.php'); // Link to library file phpMyGraph 5.0
include "conninfo.php";

$cfg ['title'] = 'Sensor 1 Bar Chart'; //Configure, title, width and height
$cfg ['width'] = 1250; //number of pixels horizontally
$cfg ['height'] = 300; //number of pixels vertically
$cfg['title-font-size'] = 20; //size of tittle

$query="SELECT mydata, thedate FROM rainfallreadings WHERE sensor = 1";
$result=mysqli_query($link,$query); //get data from database

$data=array();
$x=0;

while ($r=mysqli_fetch_array($result))
{
    $data[$x]=$r["mydata"];
    //put data into array
    $x++;
}

header ("Content-type: image/png");
$graph = new phpMyGraph();
$graph->parseVerticalColumnGraph ($data, $cfg); //create graph
header('Location:graph_result.php');
```

Averages

Having data at your fingertips is great but being able to view important up to date snippets of information giving the user what they were exactly looking for is even better.

Within the user interface there was a page created called averages which gives up to date snippets of information including:

- Highest rainfall in mm
- Date of the highest rainfall reading
- Average rainfall in mm
- Number of dry readings
- Lowest rainfall reading in mm
- Date of the lowest rainfall reading

The information above is all controlled via a dropdown list which the user can choose which sensor they would like the data for. The value of the drop-down list is then transferred into a PHP variable and then the variable is used in a SQL statement that relates to each snippets of information above.

Avarage Rainfall Data

On this page you will be able to view snipits of information from the latest rain fall data collected by our high tech sensors. If you would like to make sure that this is the latest data please go to the update rainfall data page in Import our latest files.

Update rainfall data

Please choose from the following sensor in order to display information:

Please Select Sensor

Submit Sensor

Highest rainfall (wettest day) for the selected sensor and the date it happened:

mm

Avarage rainfall to the left and how many days there was no rain fall to the right for the selected sensor:

mm

Lowest rainfall (driest day) for the selected sensor and the date it happened:

mm

```
include "conninfo.php";

if(isset($_POST['sensor'])){
    $query = $_POST['sensor'];

    $sql = ("SELECT COUNT(mydata) FROM `rainfallreadings` WHERE mydata = 0 and sensor ='$query'");
    $result = mysqli_query($link, $sql);

    if (mysqli_num_rows($result) > 0) {
        // output data of each row
        while($row = mysqli_fetch_assoc($result)) {
            $norain= $row["COUNT(mydata)"];
        }
    }
    else {
        echo "0 results";
    }
}

mysqli_close($link);
```

Test case diagram

Objective	Steps	Expected	Outcome
Import data	-Log in -Go to rainfall_readings -Click update data	Up to data data is import into the database	Pass
Search via date	-Log in -Go to rainfall_readings -Enter a search via a date	Only readings for selected date comeback	Fail – date picker is not working
View Graph	-Log in -Go to rainfall_graphs -click a graph	Graph displays	Pass

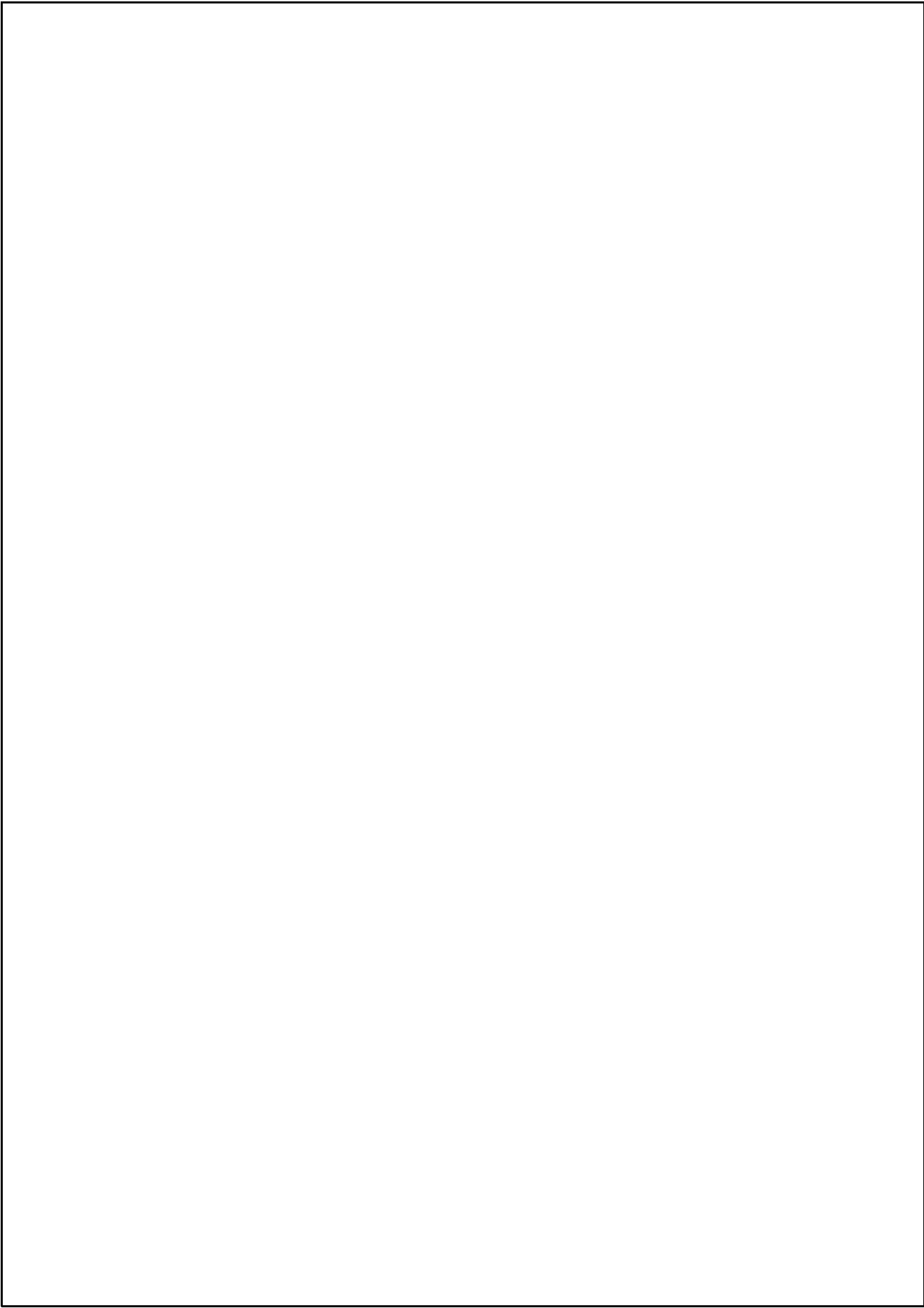
a

Conclusion

The developer feels that the user interface that was developed worked well and allows the user to follow all prerequisites including:

- Security including log in
- Viewing and interacting with the data collected
- Graph visual representation of the data

In future the developer could have added more validation in to the website in order to alert the user that there is nothing to be imported along with making sure that the date search works.



Bibliography

[1]"PHP Login Form with Sessions | FormGet", *FormGet*, 2020. [Online]. Available: <https://www.formget.com/login-form-in-php/>. [Accessed: 28- Feb- 2020].

FINAL GRADE

64/100

GENERAL COMMENTS

Instructor

Data management - Generally good, although some fields are a bit short - mydata at 2 and id at 5 are a bit on the short side. Also, you need to have an id within the login table. You have transferred the data albeit in a copy and pasted way, it would be more efficient/expandable using a loop or function. 6/10

Range of HTML/PHP coding demonstrated - 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 limiting to just sensor or just date. You did attempt this but it wasn't quite working. Some good stats included eg max, avg, lowest reading. Could also have had wettest overall day rather than wettest reading. Login is implemented well, but you need to stop URL-based attacks (ie the user gaining entry by guessing/remembering a URL that should be inaccessible to them).

17/25

Visualisation of data using graphs - Graphs are hardcoded but not 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. The axes need to show the date to allow the user to interpret the data 8/15

Overall integration of website - Looks good, although a few pages are unstyled. Watch your spelling too eg "weather" "column". The banner is a bit big - make it a bit smaller to cut down the need to scroll 7 /10

Knowledge and understanding overall - A good

understanding was demonstrated, but reflect on the points raised above

14/20

Report - Don't use "the developer" in an academic report - this is really "I" in disguise. You need to use the passive voice. For example, "The first feature the developer had to investigate" -> "The first feature which was investigated was..." Please number your sections also. Some good use of screenshots and code excerpts. 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.

12/20

Overall - 64%

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12
