HANNAH BOADIWAA LORMENYO

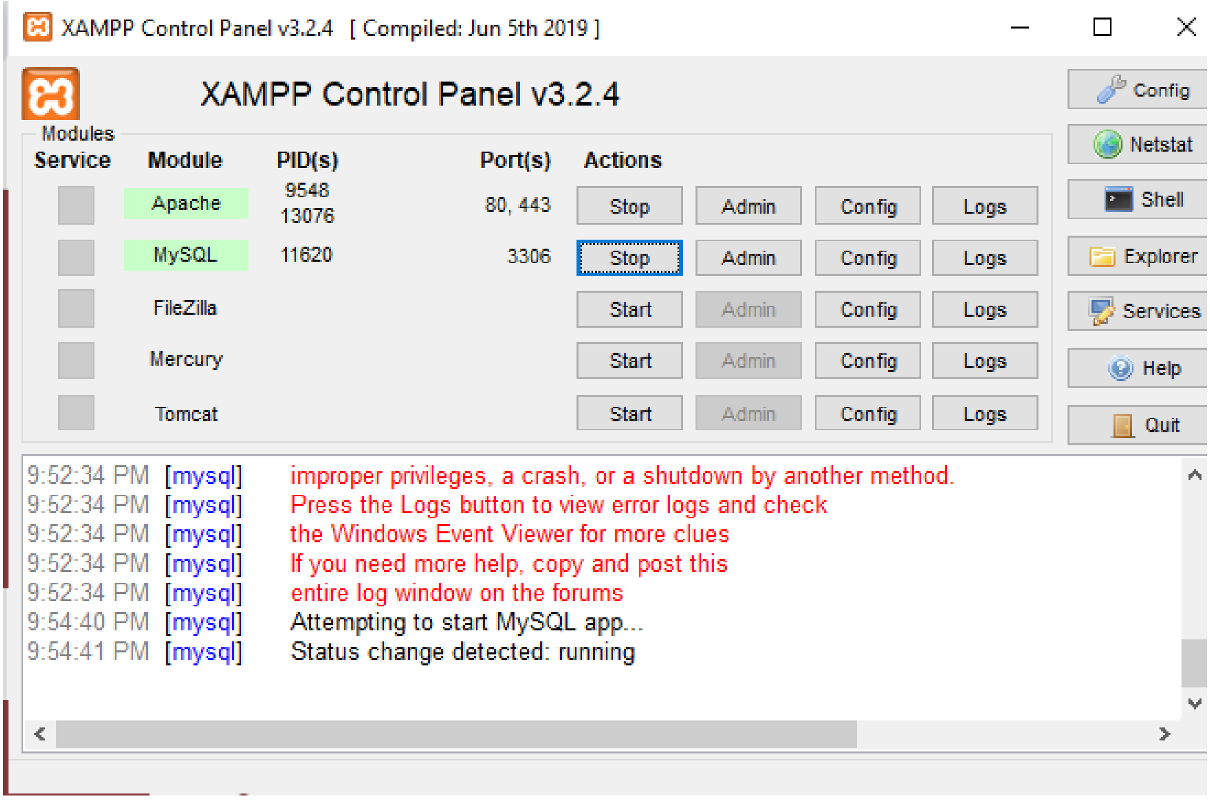
INTERNET OF THINGS: LAB 2

The main goal of this lab is to create a database and a php backend to interact with the database.

The steps taken in completing this task has been outlined below.

CREATING THE DATABASE

Apache and MySQL server were started on the XAMPP control panel. Afterwards, the localhost url was entered in the browser to show the Apache dashboard. From the Apache dashboard, I clicked on the phpMyAdmin link in the navigation bar which redirected me to the phpMyAdmin page.



XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]

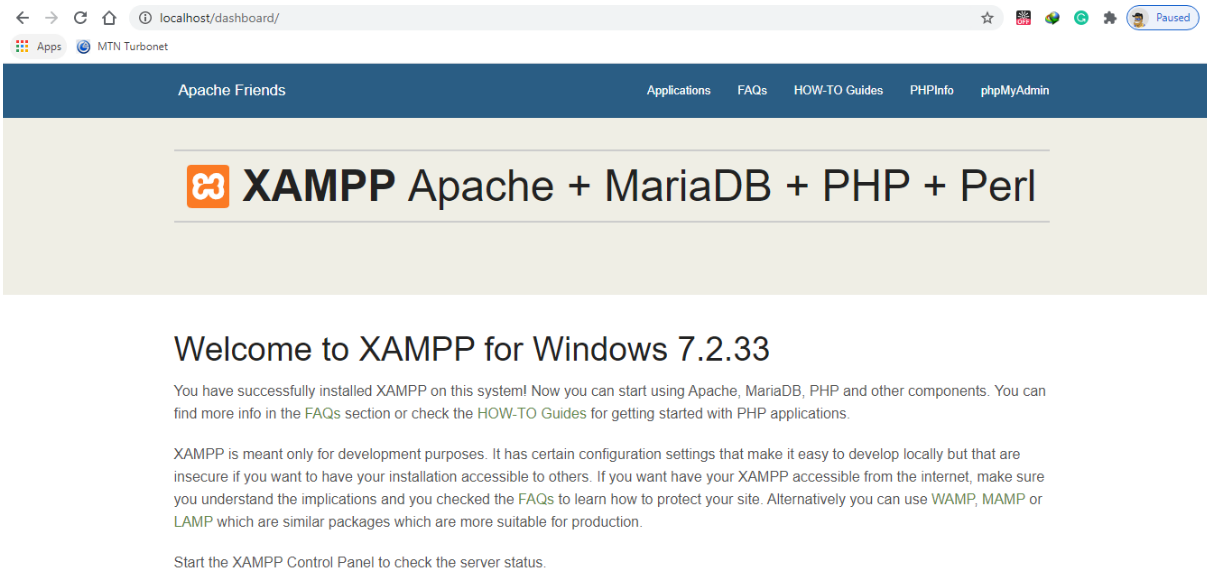
XAMPP Control Panel v3.2.4

Modules

Service	Module	PID(s)	Port(s)	Actions
	Apache	9548 13076	80, 443	Stop Admin Config Logs
	MySQL	11620	3306	Stop Admin Config Logs
	FileZilla			Start Admin Config Logs
	Mercury			Start Admin Config Logs
	Tomcat			Start Admin Config Logs

```
9:52:34 PM  [mysql]     improper privileges, a crash, or a shutdown by another method.
9:52:34 PM  [mysql]     Press the Logs button to view error logs and check
9:52:34 PM  [mysql]     the Windows Event Viewer for more clues
9:52:34 PM  [mysql]     If you need more help, copy and post this
9:52:34 PM  [mysql]     entire log window on the forums
9:54:40 PM  [mysql]     Attempting to start MySQL app...
9:54:41 PM  [mysql]     Status change detected: running
```

Figure 1: Starting the Apache server and MySQL server



Welcome to XAMPP for Windows 7.2.33

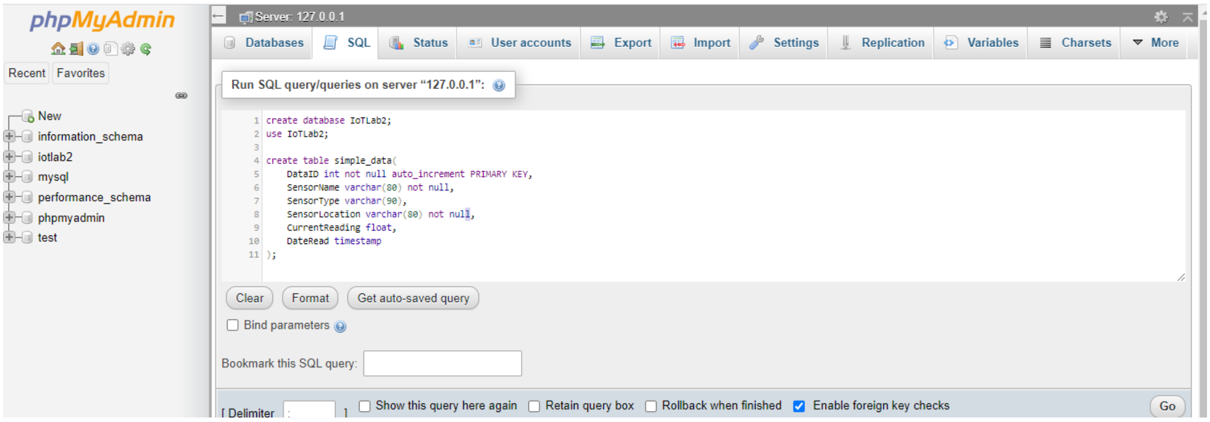
You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others. If you want have your XAMPP accessible from the internet, make sure you understand the implications and you checked the FAQs to learn how to protect your site. Alternatively you can use WAMP, MAMP or LAMP which are similar packages which are more suitable for production.

Start the XAMPP Control Panel to check the server status.

Figure 2: The Apache dashboard

On this page, I started by creating the **IoTLab2** database and the **simple_data** database using an SQL script in the SQL tab of the workspace.



Run SQL query/queries on server “127.0.0.1”:

```
create database IoTLab2;
use IoTLab2;

create table simple_data(
    DataID int not null auto_increment PRIMARY KEY,
    SensorName varchar(80) not null,
    SensorType varchar(90),
    SensorLocation varchar(80) not null,
    CurrentReading float,
    DateRead timestamp
);
```

□ Bind parameters

Bookmark this SQL query:

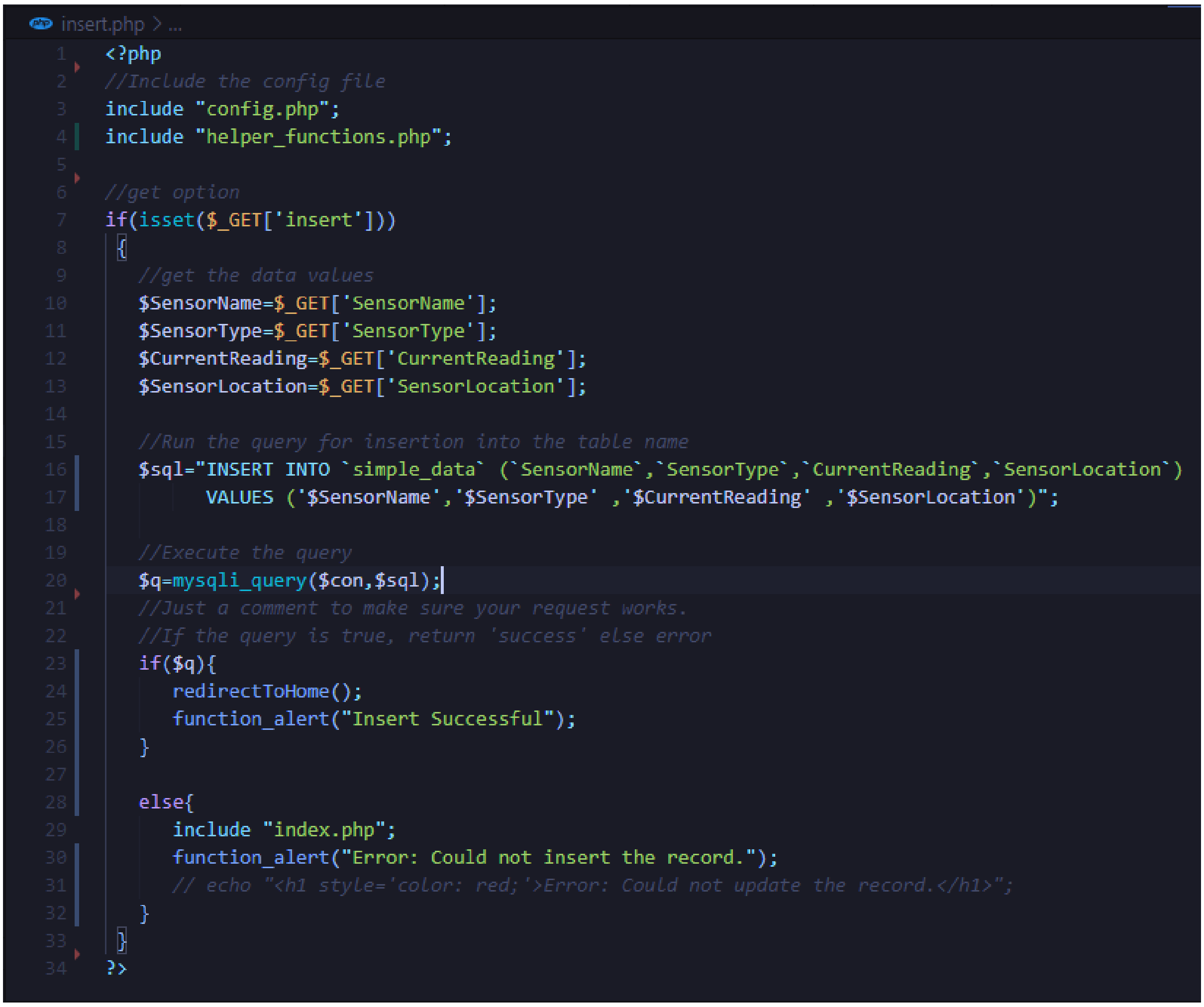
[Delimiter ;] □ Show this query here again □ Retain query box □ Rollback when finished ☑ Enable foreign key checks

Figure 3: PHPmyAdmin workspace

CREATING THE BACKEND

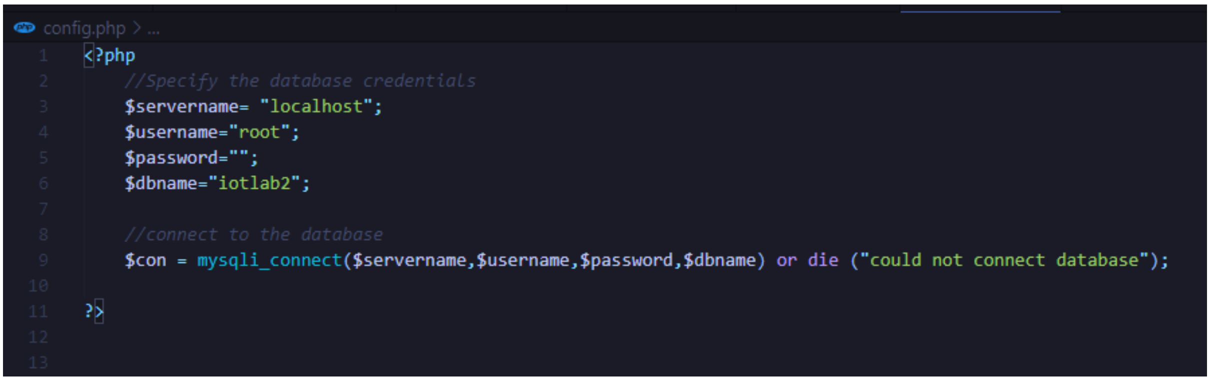
After creating the tables, I created a folder in **htdocs** folders located in the XAMP root folder. I named this folder lab2. I opened the folder in Visual Studio Code, then I created four php files; **insert.php**, **helper_functions.php**, **delete.php**, **update.php**, **config.php,** and **listings.php**.

I also used a bootstrap template in creating a great graphical user interface for users. Screenshots of the various pages can be seen below:



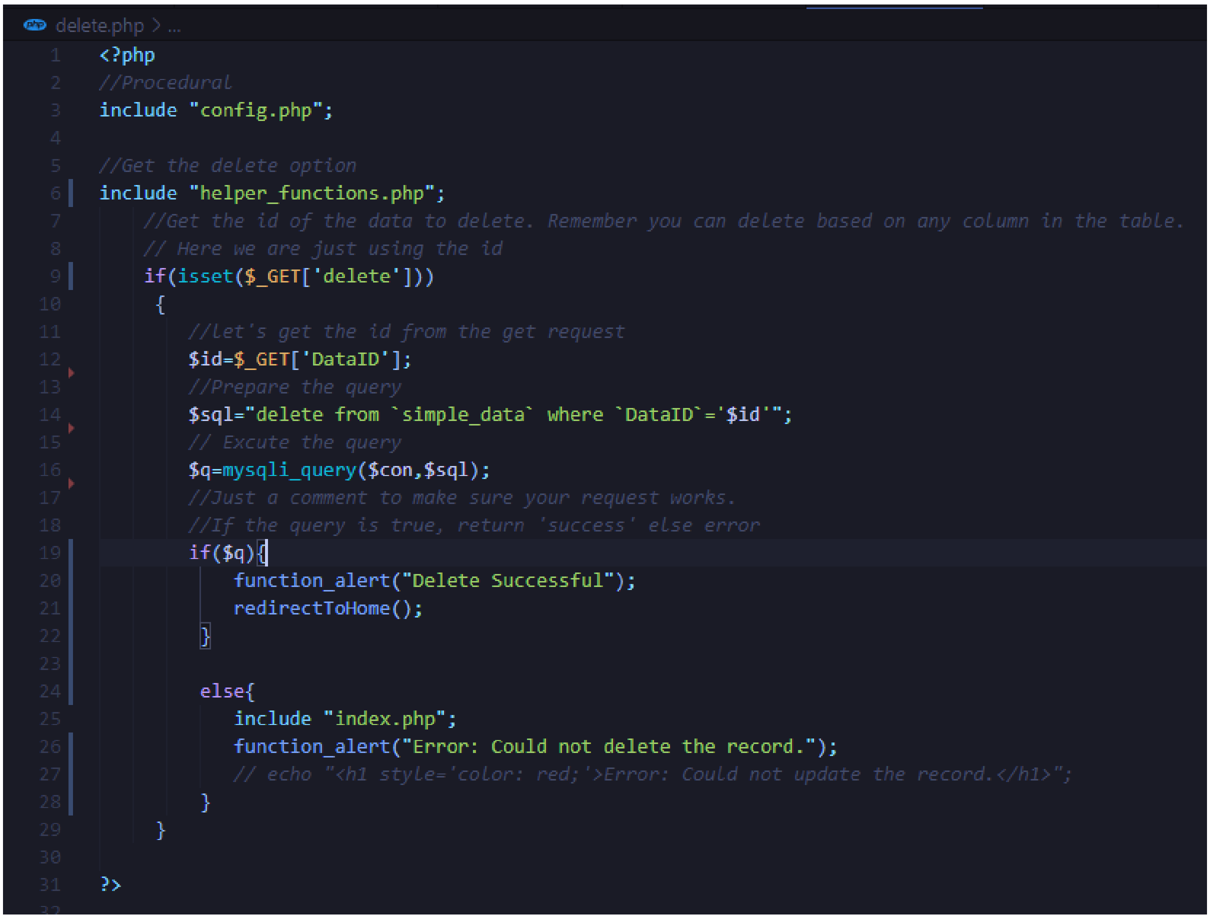
```php
<?php
//Include the config file
include "config.php";
include "helper_functions.php";

//get option
if(isset($_GET['insert']))
{
    //get the data values
    $SensorName=$_GET['SensorName'];
    $SensorType=$_GET['SensorType'];
    $CurrentReading=$_GET['CurrentReading'];
    $SensorLocation=$_GET['SensorLocation'];

    //Run the query for insertion into the table name
    $sql="INSERT INTO `simple_data` (`SensorName`,`SensorType`,`CurrentReading`,`SensorLocation`)
          VALUES ('$SensorName','$SensorType' ,'$CurrentReading' ,'$SensorLocation')";

    //Execute the query
    $q=mysqli_query($con,$sql);
    //Just a comment to make sure your request works.
    //If the query is true, return 'success' else error
    if($q){
        redirectToHome();
        function_alert("Insert Successful");
    }

    else{
        include "index.php";
        function_alert("Error: Could not insert the record.");
        // echo "<h1 style='color: red;'>Error: Could not update the record.</h1>";
    }
}
?>
```

Figure 4: Code for Insert



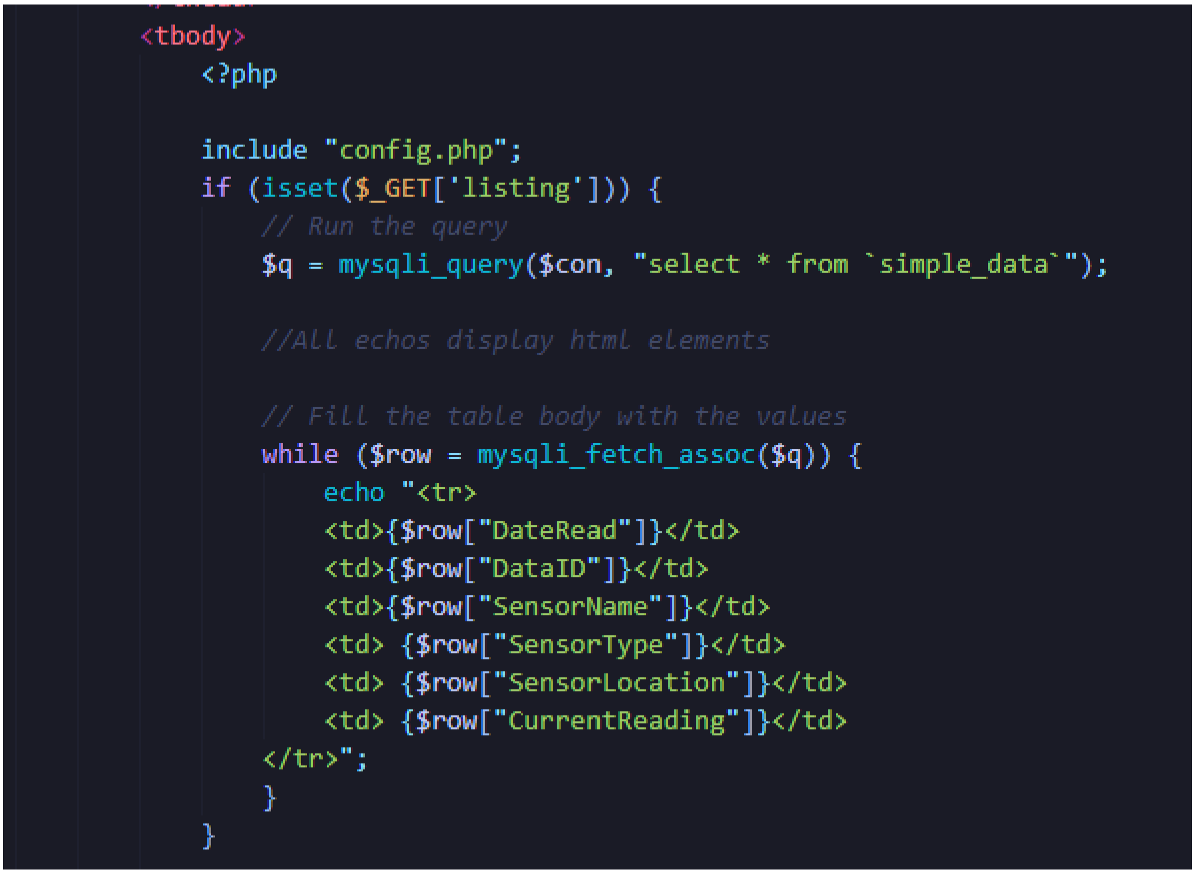
```php
<?php
    //Specify the database credentials
    $servername= "localhost";
    $username="root";
    $password="";
    $dbname="iotlab2";

    //connect to the database
    $con = mysqli_connect($servername,$username,$password,$dbname) or die ("could not connect database");

?>
```

Figure 5:Code for Connection to the database



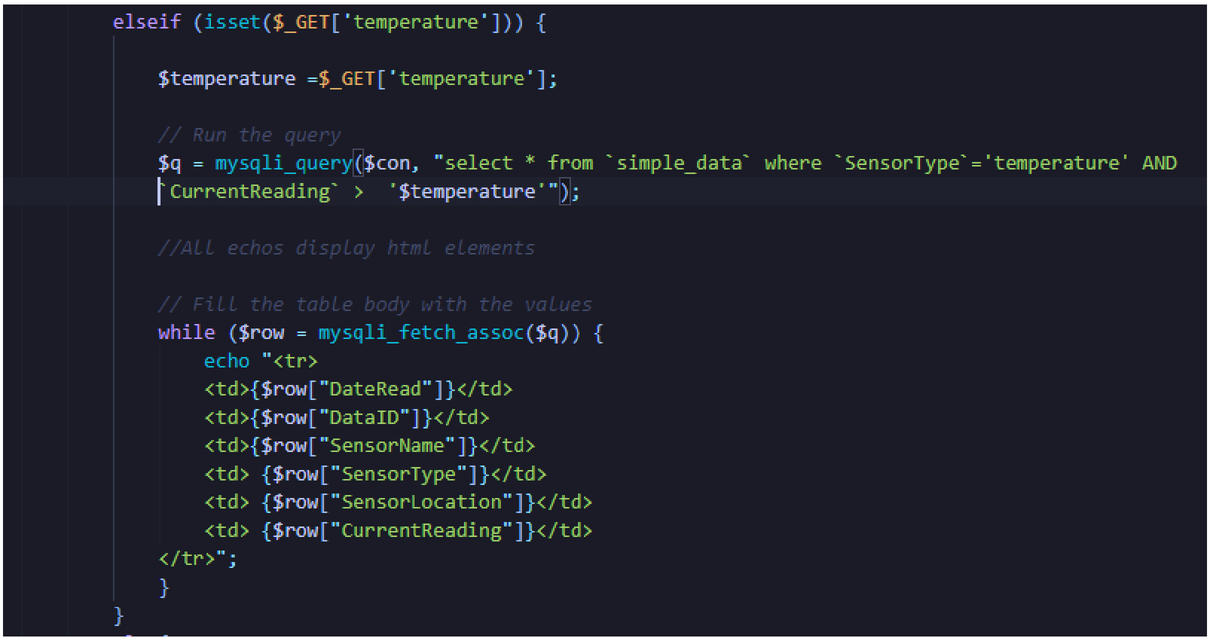
```php
<?php
//Procedural
include "config.php";

//Get the delete option
include "helper_functions.php";
    //Get the id of the data to delete. Remember you can delete based on any column in the table.
    // Here we are just using the id
    if(isset($_GET['delete']))
     {
        //let's get the id from the get request
        $id=$_GET['DataID'];
        //Prepare the query
        $sql="delete from `simple_data` where `DataID`='$id'";
        // Excute the query
        $q=mysqli_query($con,$sql);
        //Just a comment to make sure your request works.
        //If the query is true, return 'success' else error
        if($q){
            function_alert("Delete Successful");
            redirectToHome();
         }

         else{
            include "index.php";
            function_alert("Error: Could not delete the record.");
            // echo "<h1 style='color: red;'>Error: Could not update the record.</h1>";
         }
     }

?>
```

Figure 6: Code for delete



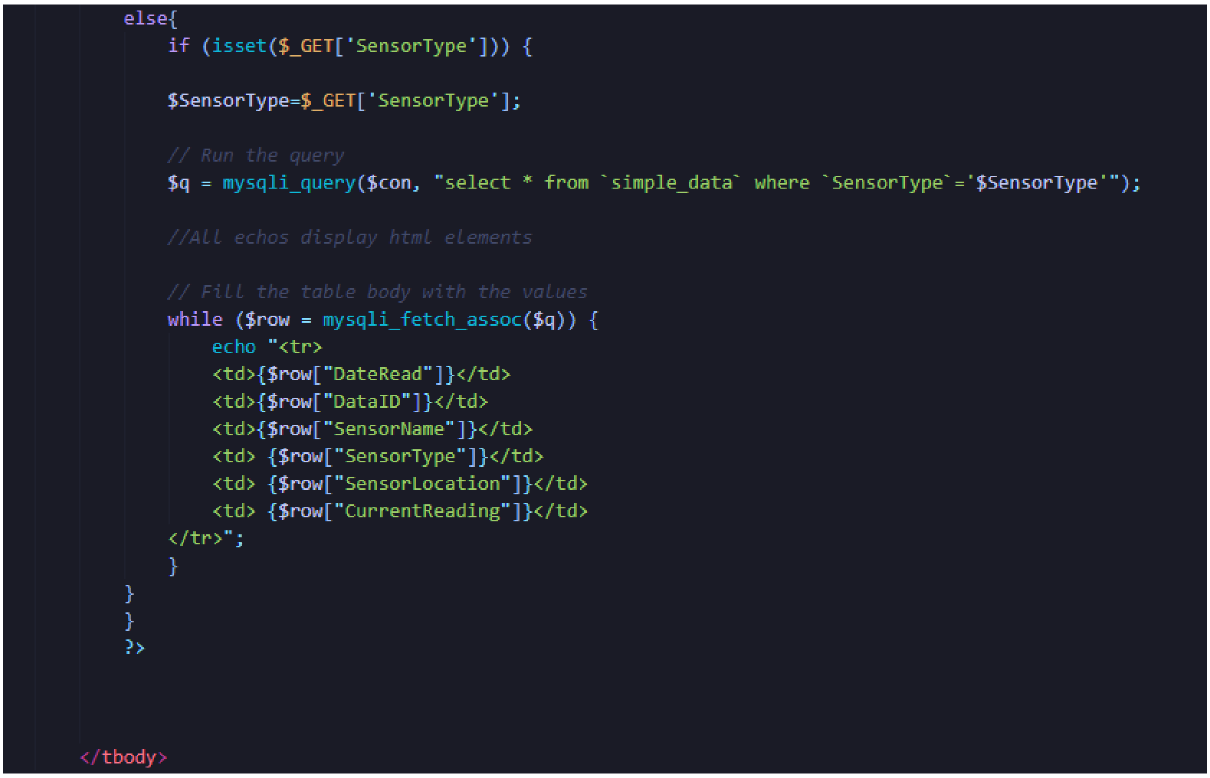
```
<tbody>
    <?php

    include "config.php";
    if (isset($_GET['listing'])) {
        // Run the query
        $q = mysqli_query($con, "select * from `simple_data`");

        //All echos display html elements

        // Fill the table body with the values
        while ($row = mysqli_fetch_assoc($q)) {
            echo "<tr>
            <td>{$row["DateRead"]}</td>
            <td>{$row["DataID"]}</td>
            <td>{$row["SensorName"]}</td>
            <td> {$row["SensorType"]}</td>
            <td> {$row["SensorLocation"]}</td>
            <td> {$row["CurrentReading"]}</td>
        </tr>";
        }
    }
```

Figure 7: Code for all data listing



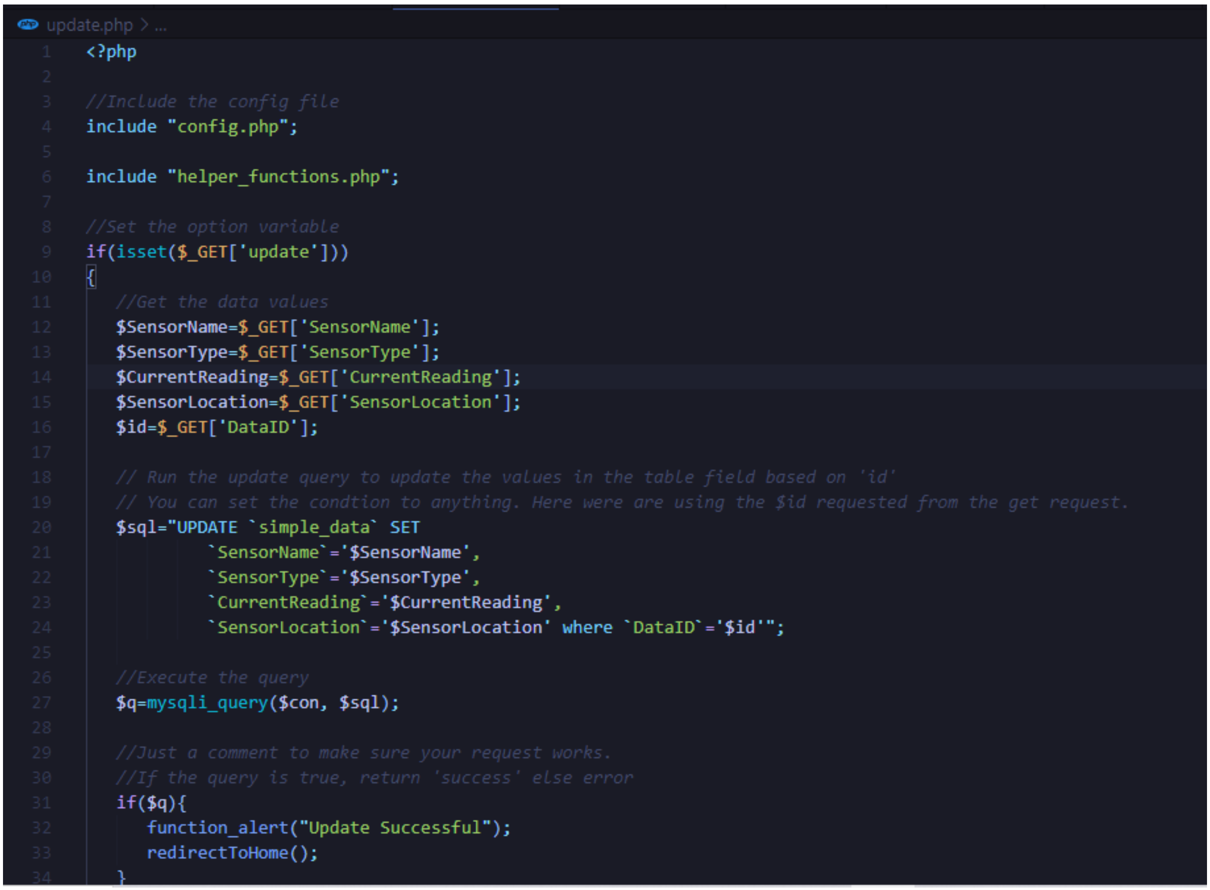
```
elseif (isset($_GET['temperature'])) {

    $temperature =$_GET['temperature'];

    // Run the query
    $q = mysqli_query($con, "select * from `simple_data` where `SensorType`='temperature' AND
    `CurrentReading` >  '$temperature'");

    //All echos display html elements

    // Fill the table body with the values
    while ($row = mysqli_fetch_assoc($q)) {
        echo "<tr>
        <td>{$row["DateRead"]}</td>
        <td>{$row["DataID"]}</td>
        <td>{$row["SensorName"]}</td>
        <td> {$row["SensorType"]}</td>
        <td> {$row["SensorLocation"]}</td>
        <td> {$row["CurrentReading"]}</td>
    </tr>";
    }
}
```

Figure 8: Code for Temperature listing



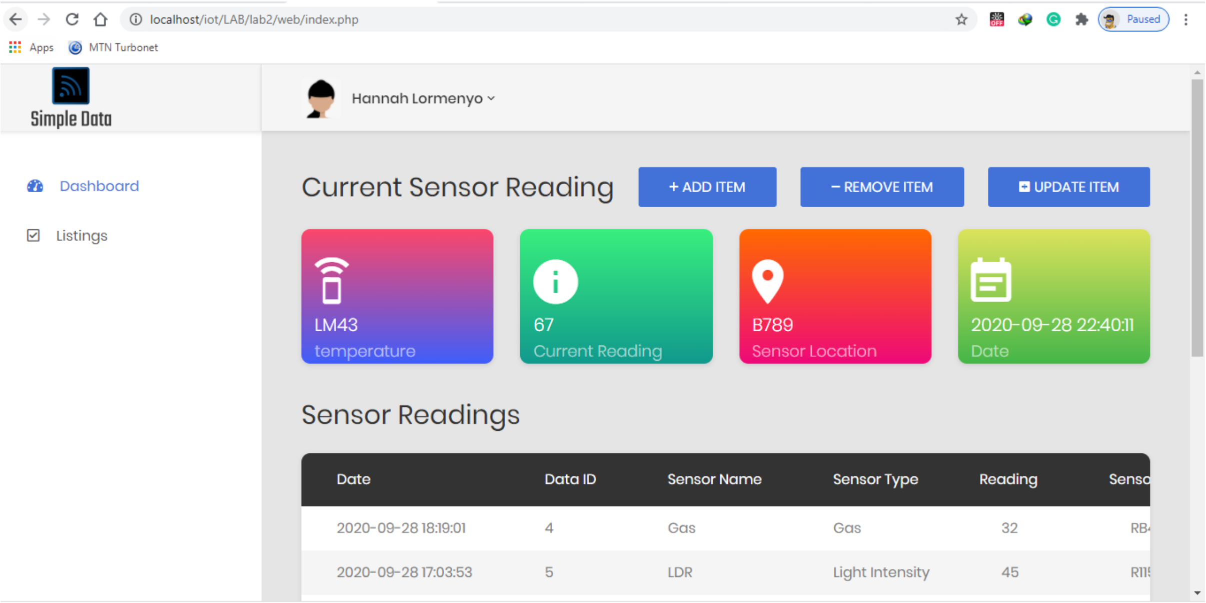
```
    else{
        if (isset($_GET['SensorType'])) {

        $SensorType=$_GET['SensorType'];

        // Run the query
        $q = mysqli_query($con, "select * from `simple_data` where `SensorType`='$SensorType'");

        //All echos display html elements

        // Fill the table body with the values
        while ($row = mysqli_fetch_assoc($q)) {
            echo "<tr>
            <td>{$row["DateRead"]}</td>
            <td>{$row["DataID"]}</td>
            <td>{$row["SensorName"]}</td>
            <td> {$row["SensorType"]}</td>
            <td> {$row["SensorLocation"]}</td>
            <td> {$row["CurrentReading"]}</td>
        </tr>";
        }
    }
    }
    ?>


</tbody>
```

Figure 9: Code for Sensor Listing



```
<?php

//Include the config file
include "config.php";

include "helper_functions.php";

//Set the option variable
if(isset($_GET['update']))
{
    //Get the data values
    $SensorName=$_GET['SensorName'];
    $SensorType=$_GET['SensorType'];
    $CurrentReading=$_GET['CurrentReading'];
    $SensorLocation=$_GET['SensorLocation'];
    $id=$_GET['DataID'];

    // Run the update query to update the values in the table field based on 'id'
    // You can set the condtion to anything. Here were are using the $id requested from the get request.
    $sql="UPDATE `simple_data` SET
            `SensorName`='$SensorName',
            `SensorType`='$SensorType',
            `CurrentReading`='$CurrentReading',
            `SensorLocation`='$SensorLocation' where `DataID`='$id'";

    //Execute the query
    $q=mysqli_query($con, $sql);

    //Just a comment to make sure your request works.
    //If the query is true, return 'success' else error
    if($q){
        function_alert("Update Successful");
        redirectToHome();
    }
```

Figure 10: Code for Update



Current Sensor Reading

LM43
temperature

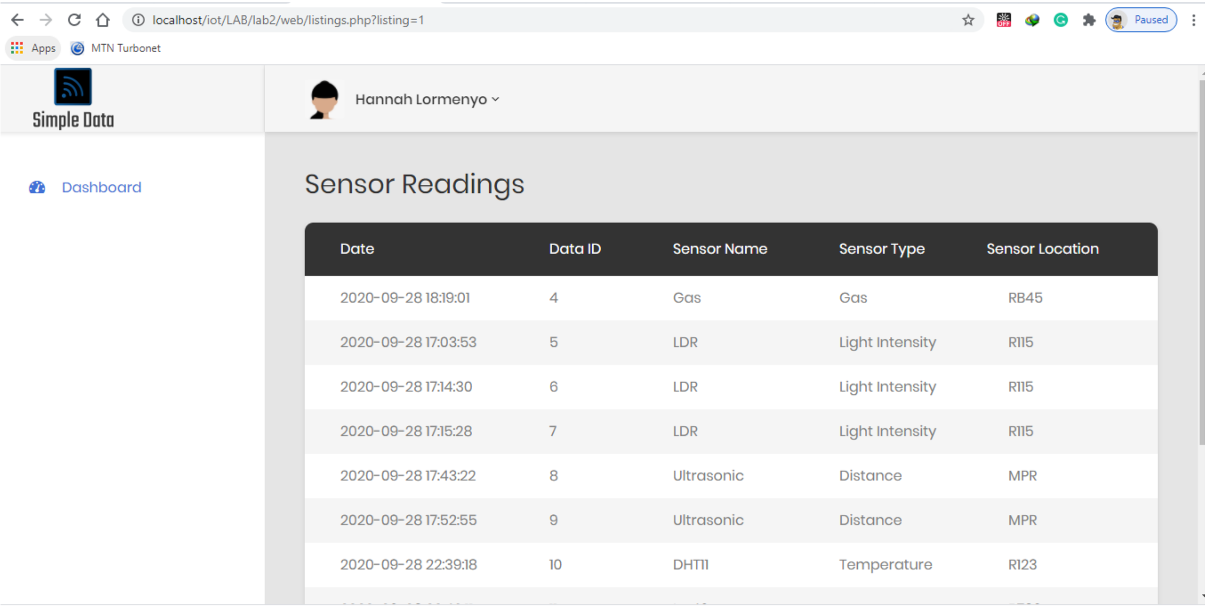
67
Current Reading

B789
Sensor Location

2020-09-28 22:40:11
Date

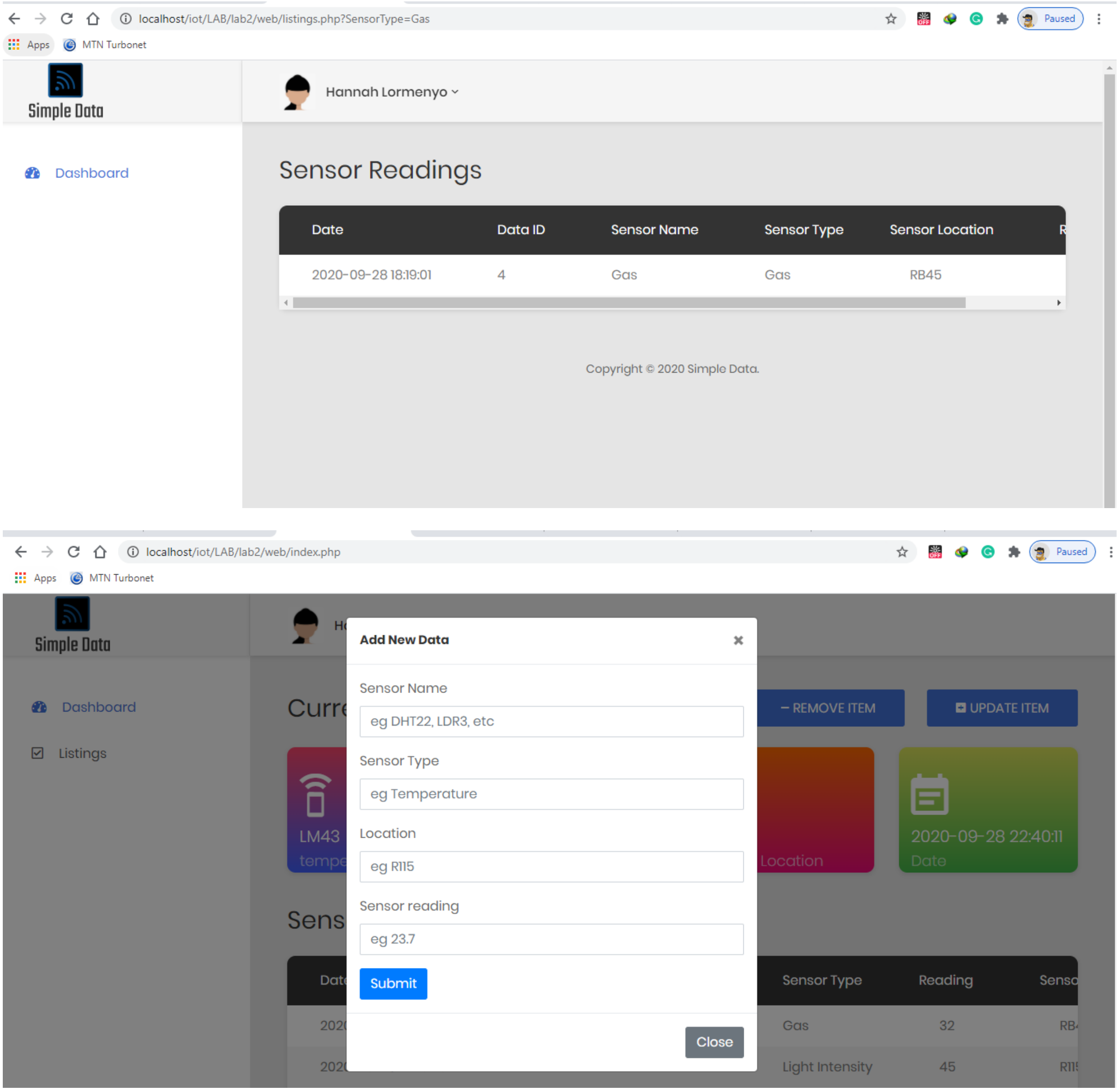
Sensor Readings

Date	Data ID	Sensor Name	Sensor Type	Reading	Senso
2020-09-28 18:19:01	4	Gas	Gas	32	RB4
2020-09-28 17:03:53	5	LDR	Light Intensity	45	R11



Sensor Readings

Date	Data ID	Sensor Name	Sensor Type	Sensor Location
2020-09-28 18:19:01	4	Gas	Gas	RB45
2020-09-28 17:03:53	5	LDR	Light Intensity	R115
2020-09-28 17:14:30	6	LDR	Light Intensity	R115
2020-09-28 17:15:28	7	LDR	Light Intensity	R115
2020-09-28 17:43:22	8	Ultrasonic	Distance	MPR
2020-09-28 17:52:55	9	Ultrasonic	Distance	MPR
2020-09-28 22:39:18	10	DHT11	Temperature	R123


localhost/iot/LAB/lab2/web/listings.php?SensorType=Gas

Simple Data

Hannah Lormenyo

Dashboard

Sensor Readings

Date	Data ID	Sensor Name	Sensor Type	Sensor Location	R
2020-09-28 18:19:01	4	Gas	Gas	RB45	

Copyright © 2020 Simple Data.

localhost/iot/LAB/lab2/web/index.php

Add New Data

Sensor Name

eg DHT22, LDR3, etc

Sensor Type

eg Temperature

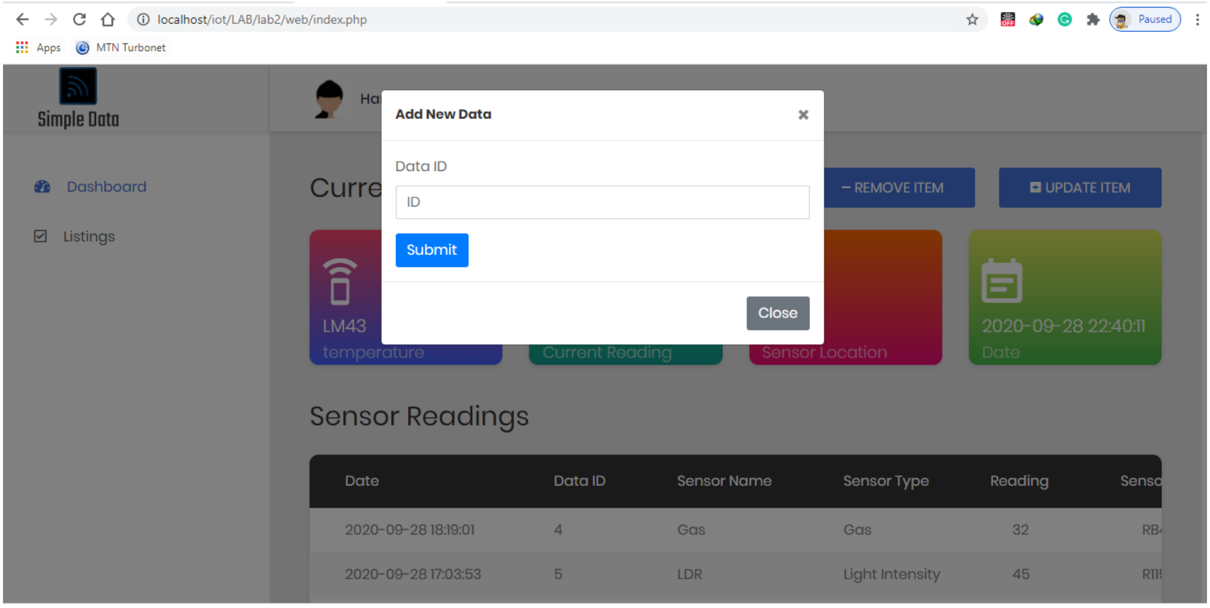
Location

eg R115

Sensor reading

eg 23.7

Figure 11: Inserting new data



localhost/iot/LAB/lab2/web/index.php

Add New Data

Data ID

ID

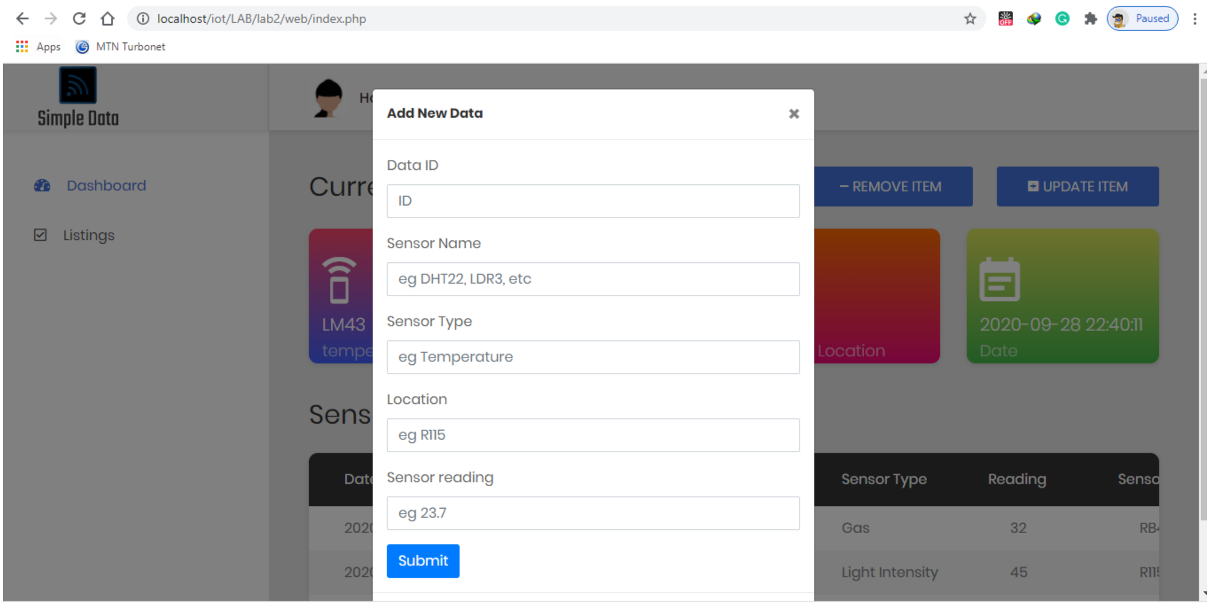
Submit

Close

Sensor Readings

Date	Data ID	Sensor Name	Sensor Type	Reading	Senso
2020-09-28 18:19:01	4	Gas	Gas	32	RB
2020-09-28 17:03:53	5	LDR	Light Intensity	45	R11

Figure 12: Deleting by ID



localhost/iot/LAB/lab2/web/index.php

Add New Data

Data ID

ID

Sensor Name

eg DHT22, LDR3, etc

Sensor Type

eg Temperature

Location

eg R115

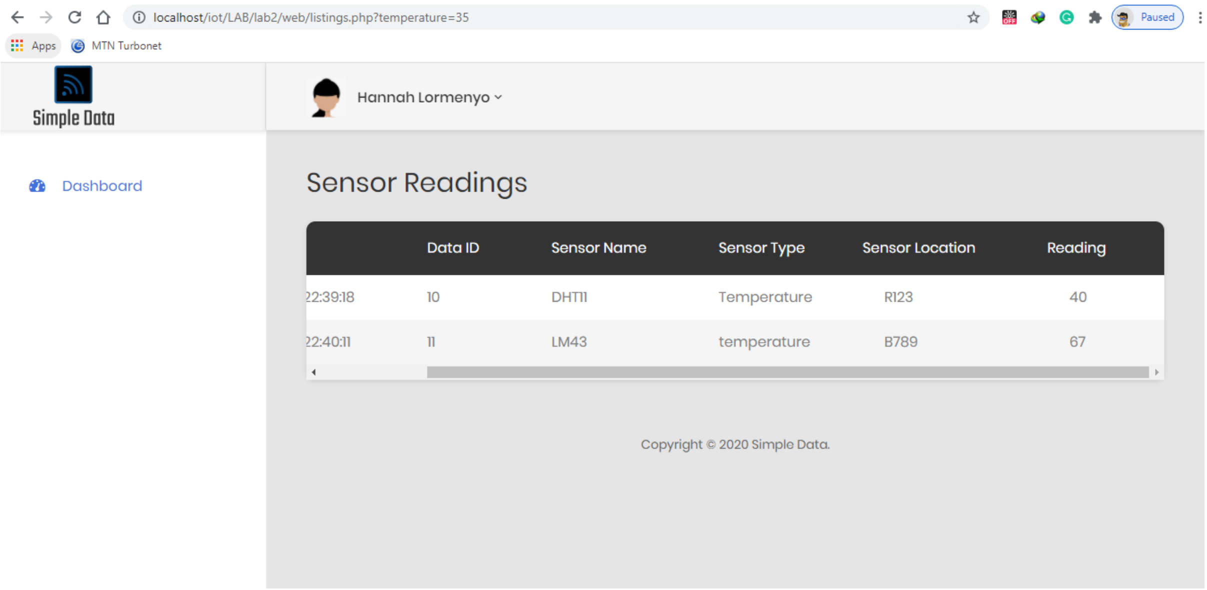
Sensor reading

eg 23.7

Submit

Date	Sensor Type	Reading	Senso
202	Gas	32	RB
202	Light Intensity	45	R11

Figure 13: Updating existing data



Sensor Readings

	Data ID	Sensor Name	Sensor Type	Sensor Location	Reading
22:39:18	10	DHT11	Temperature	R123	40
22:40:11	11	LM43	temperature	B789	67

Copyright © 2020 Simple Data.

CONCLUSION

Through this lab, I have learnt how to use XAMPP, PHPMyAdmin, Visual Studio Code and its extensions. I also learnt how to create a database and a backend to interact with the database.