

Certification Page

This page must be the first page of your uploaded document.

Your assignment will not be graded without this page (completed with your full name in the area provided) as the first page of your uploaded document.

I, Ulvi Bajarani, certify that the work I am uploading represents efforts of my team member and mine per the following task-contribution table, and is not copied from anyone else or any other resource (such as Internet).

Team Task/Subtask Table

	<i>Tasks for this quiz</i>	Percent contributed	Percent contributed	Total percent
#	<i>Subtask</i> Description	Ulvi Bajarani	Gwendolyn Poulos	
1	Design & create DB tables in PaaS	50%	50%	100%
2	Create DB tables in PaaS	50%	50%	100%
3	Populate DB tables with "example" data.	50%	50%	100%
4	Implement Android App screen (labels & SwitchCompats) to show status of the sensors & actuators	50%	50%	100%
5	Implement sensors (SW1, SW2, . . .) and actuators (LED1, LED2, . . .) on breadboard and test their functionality with RPi <i>locally</i>	50%	50%	100%
6	Interface & test the functionality of sensors and actuators between DB tables & RPi	50%	50%	100%
7	<i>Simulate</i> the functionality of <i>SwitchCompats</i> between DB tables & Android using RPi.	50%	50%	100%
8	Program Android App to interact with DB tables in PaaS	50%	50%	100%
9	Test interaction between Android App and RPi through DB tables in PaaS	50%	50%	100%
10	Design website	50%	50%	100%
11	Interface website components with the DB tables	50%	50%	100%
12	Collect real data and test the overall functionality of the project	50%	50%	100%

[phpMyAdmin](#)

The diagram illustrates a database schema structure. At the top level, there are two schemas: 'information_schema' and 'u551001383_team'. The 'u551001383_team' schema is expanded to show its contents. It contains a table named 'New' (indicated by a green checkmark icon). Below 'New' are several views, each represented by a blue checkmark icon: 'ActiveAlarms', 'Alarms', 'CannedMessages', 'device', 'SensorsAndActuators', 'TransactionalLogs', 'Users', and 'webuser'.

← Server: 127.0.0.1:3306 » Database: u551001383_team » Table: CannedMessages

Browse Structure SQL Search Insert Export Import Operations Triggers

Showing rows 0 - 14 (15 total, Query took 0.0003 seconds.)

SELECT * FROM `CannedMessages`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	MessageID	MessageType	MessageDescription
<input type="checkbox"/>	1	ALARM	Systolic BP HIGH
<input type="checkbox"/>	10	SYSTEM	User login - failure
<input type="checkbox"/>	11	SYSTEM	User logout
<input type="checkbox"/>	12	ADVISE	Reboot system
<input type="checkbox"/>	13	OPER	IV infusion of pressor
<input type="checkbox"/>	14	OPER	IV infusion of X
<input type="checkbox"/>	15	OPER	IV infusion of W
<input type="checkbox"/>	2	ALARM	Diastolic BP LOW
<input type="checkbox"/>	3	ALARM	BLOOD SUGAR HIGH
<input type="checkbox"/>	4	ALARM	BLOOD SUGAR LOW
<input type="checkbox"/>	5	CNST	Diastolic & Systolic BP simultaneously out of range
<input type="checkbox"/>	6	CNST	?
<input type="checkbox"/>	7	SYSTEM	System up
<input type="checkbox"/>	8	SYSTEM	System shutting down
<input type="checkbox"/>	9	SYSTEM	User login - success

Console

phpMyAdmin



Recent

Favorites

information_schema

u551001383_team

New

ActiveAlarms

Alarms

CannedMessages

device

SensorsAndActuators

TransactionalLogs

Users

webuser

Server: 127.0.0.1:3306 » Database: u551001383_team » Table: SensorsAndActuators

Browse

Structure

SQL

Search

Insert

Export

Import

Operations

Triggers

Showing rows 0 - 7 (8 total, Query took 0.0004 seconds.)

SELECT * FROM `SensorsAndActuators`

 Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh] Show all | Number of rows: 25

Filter rows: Search this table

Sort by key:

None

+ Options

	← T →	DevID	DevType	FunctionDescription	Control	Device Status
<input type="checkbox"/>	Edit Copy Delete	LED1	ACTUATOR	Administering Norepinephrine	ANDROID	0
<input type="checkbox"/>	Edit Copy Delete	LED2	ACTUATOR	Administering Nitroglycerin	ANDROID	0
<input type="checkbox"/>	Edit Copy Delete	LED3	ACTUATOR	IV infusion of W to decrease blood sugar	ANDROID	0
<input type="checkbox"/>	Edit Copy Delete	LED4	ACTUATOR	IV infusion of Z to increase blood sugar	ANDROID	0
<input type="checkbox"/>	Edit Copy Delete	SW1	SENSOR	sense systolic BP > 120 mm Hg	RPI	0
<input type="checkbox"/>	Edit Copy Delete	SW2	SENSOR	sense diastolic BP < 60 mm Hg	RPI	1
<input type="checkbox"/>	Edit Copy Delete	SW3	SENSOR	sense blood sugar > 120 mg/dL	RPI	0
<input type="checkbox"/>	Edit Copy Delete	SW4	SENSOR	sense blood sugar < 50 mg/dL	RPI	0

 Check all | With selected: Edit Copy Delete Export Show all | Number of rows: 25

Filter rows: Search this table

Sort by key:

None

Console results operations

phpMyAdmin

Server: 127.0.0.1:3306 » Database: u551001383_team » Table: TransactionalLogs

Browse Structure SQL Search Insert Export Import Operations Triggers

Showing rows 0 - 4 (5 total, Query took 0.0005 seconds.)

SELECT * FROM `TransactionalLogs`

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

	logID	TimestampInfo	MessageID	DataInformation
<input type="checkbox"/> Edit Copy Delete	2020-06-28 15:57:01	9	Test	
<input type="checkbox"/> Edit Copy Delete	1	2020-06-11 12:23:16	7	Some
<input type="checkbox"/> Edit Copy Delete	15	2020-06-26 11:50:11	9	Test2
<input type="checkbox"/> Edit Copy Delete	2	2020-06-12 10:8:37	9	ben
<input type="checkbox"/> Edit Copy Delete	3	2020-06-14 10:8:37	11	George

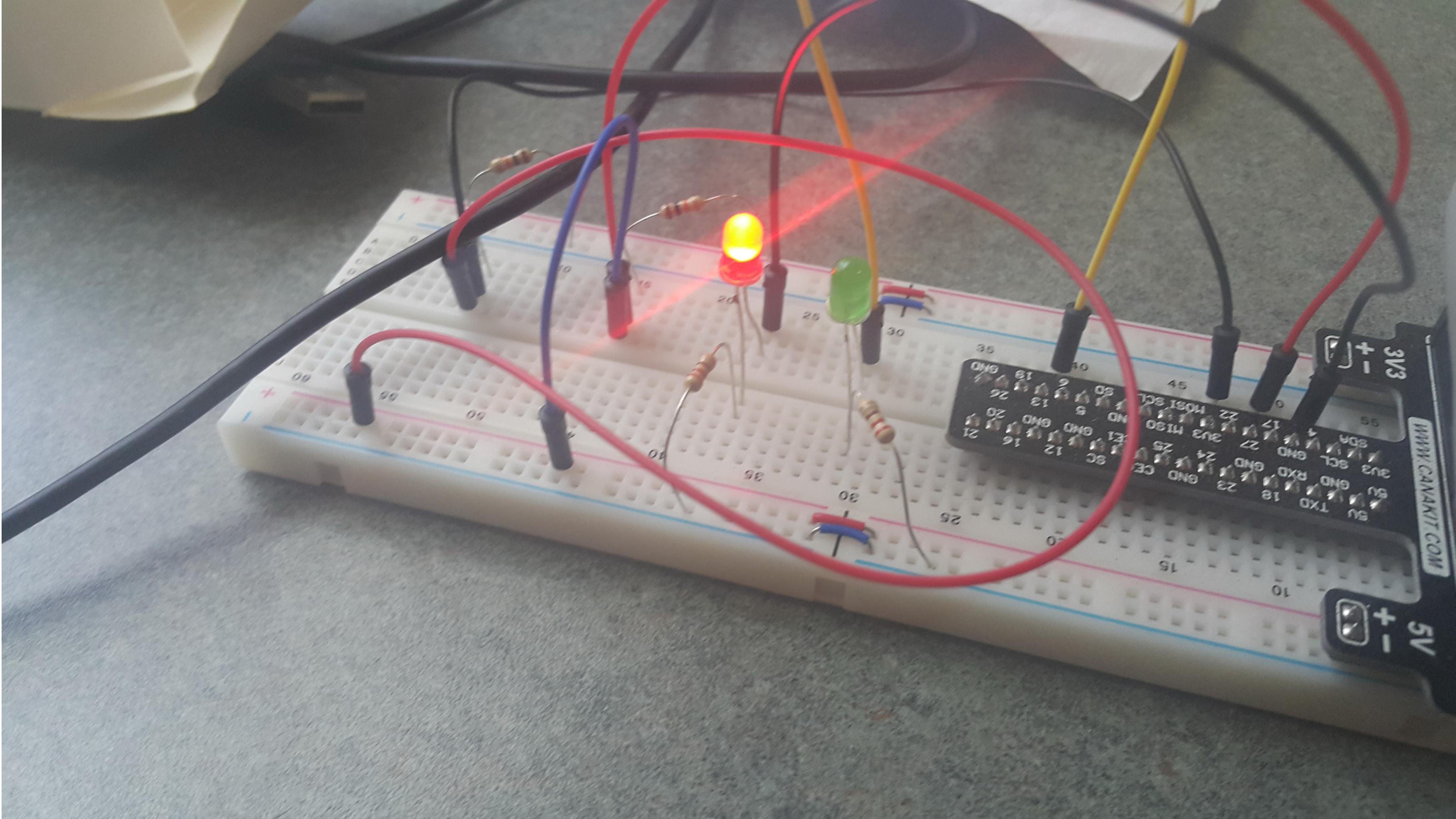
Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Check all With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

Query results operations

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)

Console



powerpoint diakov - No. | Transfers - Bitport.io | Facebook | Aslan Ismayilov - Posts | MySQL Databases | hPa | auth-db136.hostinger.com | team2project3342.online/Ch | +

BoxLayout

Monitoring System

Thu Jul 2 14:01:07 2020

Systolic BP > 120mmHg:

Diastolic BP < 60mmHg:

Administering Norepinephrine:

Administering Nitroglycerin:

Alarm1 On since: acknowledge

Alarm2 On since: acknowledge

Dispense Norepinephrin Dispense Auto

Database: u551001383_team » Table: SensorsAndActuators

SQL Search Insert Export Import Operations Triggers

```
SET `DeviceStatus` = '1' WHERE `SensorsAndActuators`.`DevID` = 'SW2';
```

[Edit inline] [Edit] [Create PHP code]

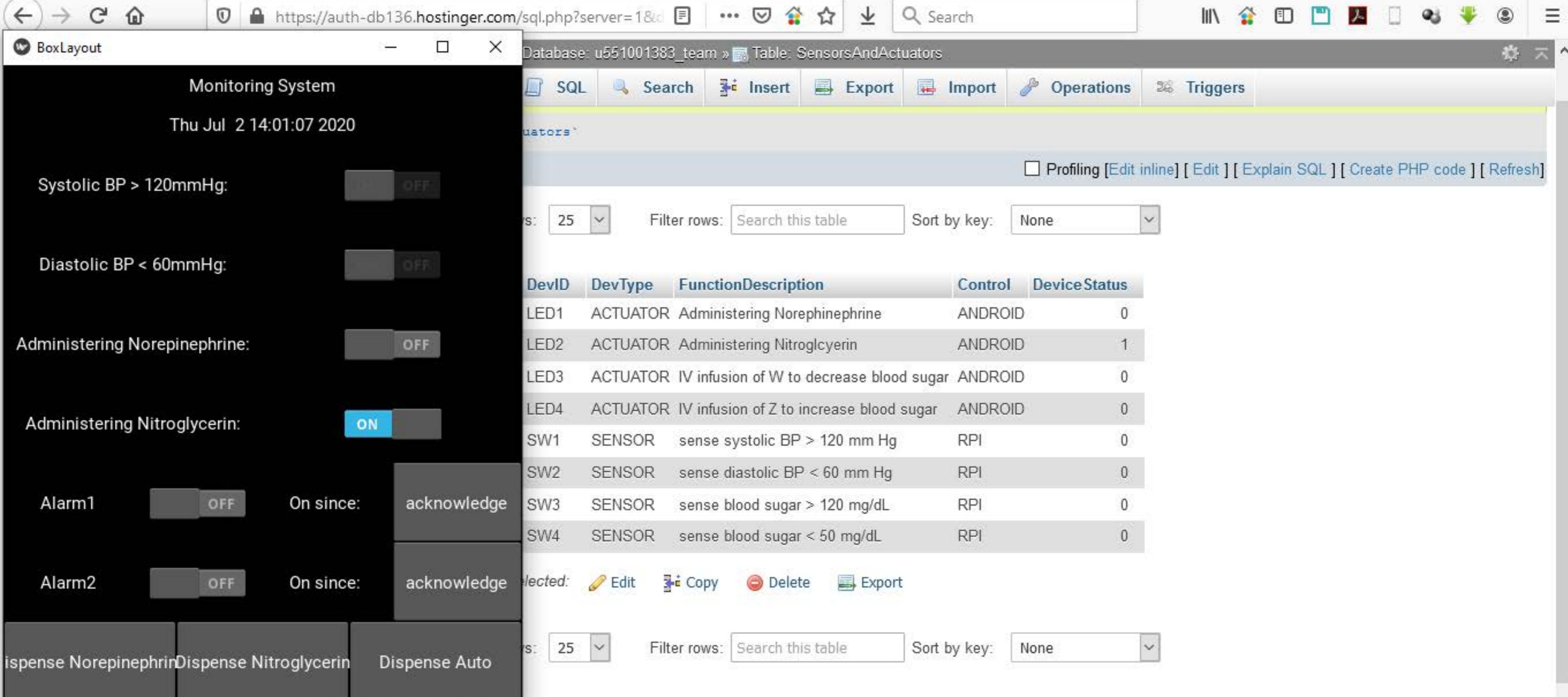
s: 25 Filter rows: Search this table Sort by key: None

DevID	DevType	FunctionDescription	Control	Device Status
LED1	ACTUATOR	Administering Norepinephrine	ANDROID	0
LED2	ACTUATOR	Administering Nitroglycerin	ANDROID	0
LED3	ACTUATOR	IV infusion of W to decrease blood sugar	ANDROID	0
LED4	ACTUATOR	IV infusion of Z to increase blood sugar	ANDROID	0
SW1	SENSOR	sense systolic BP > 120 mm Hg	RPI	0
SW2	SENSOR	sense diastolic BP < 60 mm Hg	RPI	1
SW3	SENSOR	sense blood sugar > 120 mg/dL	RPI	0
SW4	SENSOR	sense blood sugar < 50 mg/dL	RPI	0

Selected:

s: 25 Filter rows: Search this table Sort by key: None

Print



Monitoring System

Login

Username:

Password:

Submit



Monitoring System

Login

Username:

Password:

Login Credentials verified

Transferring data from team2project3342.online...



15:57
02/07/2020 ENG

Welcome

Last update: 07/02/20 03:57:49pm

Status of Sensors and Acutators

Device Name Status

LED1	1
LED2	0
LED3	0
LED4	0
SW1	0
SW2	0
SW3	0
SW4	0

Alarm Status

Alarm Description	Since	Acknowledged
3 BLOOD SUGAR HIGH	2020-03-11 14:25:21	0

[Logout](#)

The link to the Timestamp chart

powerpoint diakov - No. | Transfers - Bitport.io | Facebook | Aslan Ismayilov - Posts | MySQL Databases | hPar | auth-db136.hostinger.co | Welcome

Welcome

Last update: 07/02/20 04:04:25pm

Monitoring System

Thu Jul 2 15:57:31 2020

Systolic BP > 120mmHg:

Diastolic BP < 60mmHg:

Administering Norepinephrine:

Administering Nitroglycerin:

Alarm1 On since: acknowledge

Alarm2 On since: acknowledge

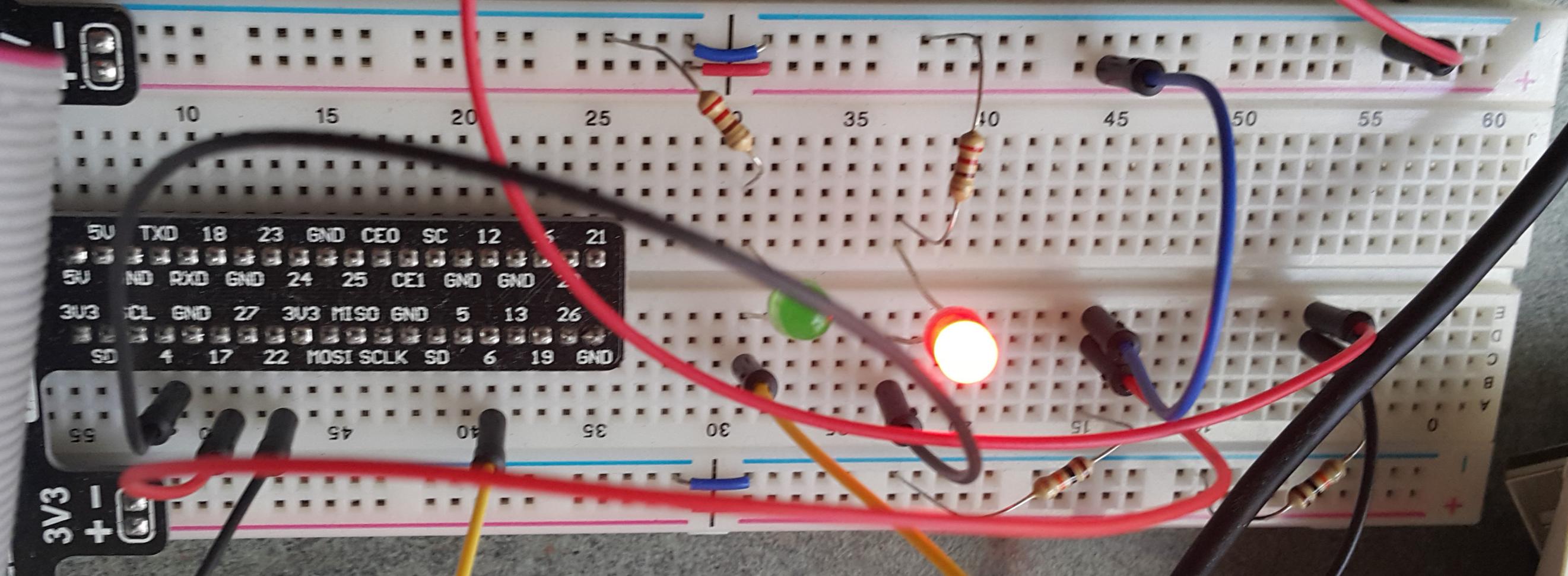
[Logout](#)

[The link to the Timestamp chart](#)

Dispense Norepinephrin Dispense Nitroglycerin Dispense Auto

VNC 192.168.137.224 (raspberrypi) - VNC Viewer
pi@raspberrypi: ~

```
The status of LED2 is 0
The status of SW1 is 1
The status of SW2 is 0
=====Server Response at 2020-07-02 22:04:09.427639
+++++Server request successful:
The status of LED1 is 1
The status of LED2 is 0
The status of SW1 is 1
The status of SW2 is 0
=====Server Response at 2020-07-02 22:04:14.765731
+++++Server request successful:
The status of LED1 is 1
The status of LED2 is 0
The status of SW1 is 1
The status of SW2 is 0
=====Server Response at 2020-07-02 22:04:20.771936
+++++Server request successful:
The status of LED1 is 1
The status of LED2 is 0
The status of SW1 is 1
The status of SW2 is 0
=====Server Response at 2020-07-02 22:04:26.182188
+++++Server request successful:
The status of LED1 is 1
The status of LED2 is 0
The status of SW1 is 1
The status of SW2 is 0
```



The code of SQL tables:

```
CREATE TABLE IF NOT EXISTS SensorsAndActuators(
    DevID VARCHAR(10),
    DevType ENUM('SENSOR', 'ACTUATOR'),
    FunctionDescription VARCHAR(255),
    Control ENUM('RPI', 'ANDROID'),
    DeviceStatus INT NOT NULL,
    PRIMARY KEY(DevID)
);
```

```
CREATE TABLE IF NOT EXISTS Alarms(
    AlarmID VARCHAR(10),
    Alarm ENUM('ALARM'),
    MessageID VARCHAR(255),
    PRIMARY KEY(AlarmID)
);
```

```
CREATE TABLE IF NOT EXISTS ActiveAlarms(
    AlarmID VARCHAR(10),
    SinceTS VARCHAR(255),
    AcknowledgementStatus INT NOT NULL,
    PRIMARY KEY(AlarmID)
);
```

```
CREATE TABLE IF NOT EXISTS CannedMessages(
    MessageID VARCHAR(10),
    MessageType VARCHAR(255),
    MessageDescription VARCHAR(255),
    PRIMARY KEY(MessageID)
);
```

```
CREATE TABLE IF NOT EXISTS TransactionalLogs
(
    logID VARCHAR(10),
    TimestampInfo VARCHAR(255),
    MessageID VARCHAR(255),
    DataInformation VARCHAR(255),
    PRIMARY KEY(logID)
);
```

```
CREATE TABLE IF NOT EXISTS Users(
    pname VARCHAR(30) NOT NULL,
    password VARCHAR(255) NOT NULL,
    PRIMARY KEY(pname)
);
```

```

/* Inserting */

INSERT INTO SensorsAndActuators VALUES ('SW1', 'SENSOR', 'sense systolic BP > 140 mm Hg', 'RPI', 0);
INSERT INTO SensorsAndActuators VALUES ('SW2', 'SENSOR', 'sense diastolic BP < 30 mm Hg', 'RPI', 0);
INSERT INTO SensorsAndActuators VALUES ('SW3', 'SENSOR', 'sense blood sugar > 120 mg/dL', 'RPI', 0);
INSERT INTO SensorsAndActuators VALUES ('SW4', 'SENSOR', 'sense blood sugar < 50 mg/dL', 'RPI', 0);
INSERT INTO SensorsAndActuators VALUES ('LED1', 'ACTUATOR', 'IV infusion of pressor to increase BP', 'ANDROID', 0);
INSERT INTO SensorsAndActuators VALUES ('LED2', 'ACTUATOR', 'IV infusion of Y to decrease BP', 'ANDROID', 0);
INSERT INTO SensorsAndActuators VALUES ('LED3', 'ACTUATOR', 'IV infusion of W to decrease blood sugar', 'ANDROID', 0);
INSERT INTO SensorsAndActuators VALUES ('LED4', 'ACTUATOR', 'IV infusion of Z to increase blood sugar', 'ANDROID', 0);

INSERT INTO Alarms VALUES ('1', 'ALARM', '1');
INSERT INTO Alarms VALUES ('2', 'ALARM', '2');
INSERT INTO Alarms VALUES ('3', 'ALARM', '3');
INSERT INTO Alarms VALUES ('4', 'ALARM', '4');

INSERT INTO ActiveAlarms VALUES ('3', '2020-03-11 14:25:21', '0');
INSERT INTO ActiveAlarms VALUES ('1', '2020-03-11 13:17:23', '1');

INSERT INTO CannedMessages VALUES ('1', 'ALARM', 'Systolic BP HIGH');
INSERT INTO CannedMessages VALUES ('2', 'ALARM', 'Diastolic BP LOW');
INSERT INTO CannedMessages VALUES ('3', 'ALARM', 'BLOOD SUGAR HIGH');
INSERT INTO CannedMessages VALUES ('4', 'ALARM', 'BLOOD SUGAR LOW');
INSERT INTO CannedMessages VALUES ('5', 'CNST', 'Diastolic & Systolic BP simultaneously out of range');
INSERT INTO CannedMessages VALUES ('6', 'CNST', '?');
INSERT INTO CannedMessages VALUES ('7', 'SYSTEM', 'System up');
INSERT INTO CannedMessages VALUES ('8', 'SYSTEM', 'System shutting down');
INSERT INTO CannedMessages VALUES ('9', 'SYSTEM', 'User login - success');
INSERT INTO CannedMessages VALUES ('10', 'SYSTEM', 'User login - failure');
INSERT INTO CannedMessages VALUES ('11', 'SYSTEM', 'User logout');
INSERT INTO CannedMessages VALUES ('12', 'ADVISE', 'Reboot system');
INSERT INTO CannedMessages VALUES ('13', 'OPER', 'IV infusion of pressor');
INSERT INTO CannedMessages VALUES ('14', 'OPER', 'IV infusion of X');
INSERT INTO CannedMessages VALUES ('15', 'OPER', 'IV infusion of W');

INSERT INTO TransactionalLogs VALUES ('1', '2020-03-11 12:23:16', '7', "");
INSERT INTO TransactionalLogs VALUES ('2', '2020-03-12 10:8:37', '9', "ben");
INSERT INTO TransactionalLogs VALUES ('3', '2020-03-12 10:8:37', '11', "ben");

```

The code of sync_rpi_data.php

```
<?php
require_once __DIR__ . '/../required/db_connect.php';
$input = file_get_contents("php://input");
$error = 0;
$out_json = array();
$out_json['success'] = 1; //assume success
$SW1_status = 0;
$SW2_status = 0;
$LED1_status = 0;
$LED2_status = 0;
if ($input)
{
    $json = json_decode($input, true); //check if it json input
    if (json_last_error() == JSON_ERROR_NONE)
    {
        if (isset($json["username"]) && isset($json["password"]) && isset($json["SW1"]) &&
            isset($json["SW2"]) && isset($json["LED1"]) && isset($json["LED2"]))
        {
            $in_username = $json["username"];
            $in_password = $json["password"]; //if the expected fields are not null, get them
            $in_SW1 = $json["SW1"];
            $in_SW2 = $json["SW2"];
            $in_LED1 = $json["LED1"];
            $in_LED2 = $json["LED2"];
            if ($stmt = $mysqli->prepare("SELECT password FROM Users WHERE pname = ? LIMIT
1"))
            {
                $stmt->bind_param('s', $in_username);
                $stmt->execute();
                $stmt->store_result(); //store_result to get num_rows etc.
                $stmt->bind_result($db_password); //get the hashed password
                $stmt->fetch();
                if ($stmt->num_rows == 1)
                { //if user exists, verify the password
                    if (password_verify($in_password, $db_password))
                    {
                        $stmt->close();
                    }
                    if ($stmt = $mysqli->prepare("UPDATE SensorsAndActuators SET DeviceStatus=? WHERE DevID = 'SW1'"))
                    { //update SW1
                        $stmt->bind_param('i', $in_SW1);
                        $stmt->execute();
                    }
                    else
                    {
                        $error = 1;
                    }
                }
            }
        }
    }
}
```

```

$stmt->close();
if (!$error && ($stmt = $mysqli->prepare("SELECT DeviceStatus FROM
SensorsAndActuators WHERE DevID = 'SW1'")))
{
    //read SW1
    $stmt->execute();
    $stmt->bind_result($SW1_status);
    $stmt->fetch();
}
else
{
    $error = 2;
}
$stmt->close();
if ($stmt = $mysqli->prepare("UPDATE SensorsAndActuators SET DeviceStatus=?"
WHERE DevID = 'SW2'"))
{
    //update SW1
    $stmt->bind_param('i', $in_SW2);
    $stmt->execute();
}
else
{
    $error = 1;
}
$stmt->close();
if (!$error && ($stmt = $mysqli->prepare("SELECT DeviceStatus FROM
SensorsAndActuators WHERE DevID = 'SW2'")))
{
    //read SW1
    $stmt->execute();
    $stmt->bind_result($SW2_status);
    $stmt->fetch();
}
else
{
    $error = 2;
}
$stmt->close();
if (!$error && ($stmt = $mysqli->prepare("SELECT DeviceStatus FROM
SensorsAndActuators WHERE DevID = 'LED1'")))
{
    //read LED1
    $stmt->execute();
    $stmt->bind_result($LED1_status);
    $stmt->fetch();
}
else
{
    $error = 3;
}
$stmt->close();

```

```

if (!$error && ($stmt = $mysqli->prepare("SELECT DeviceStatus FROM
SensorsAndActuators WHERE DevID = 'LED2'")))
{
    //read LED1
    $stmt->execute();
    $stmt->bind_result($LED2_status);
    $stmt->fetch();
}
else
{
    $error = 3;
}
$stmt->close();
}
else
{
    $error = 4;
}
}
else
{
    $error = 5;
}
}
else
{
    $error = 6;
}
}
else
{
    $error = 7;
}
}
else
{
    $error = 8;
}
}
else
{
    $error = 9;
}
}
if ($error)
{
    $out_json['success'] = 0; //flag failure
}
$out_json['SW1'] = $SW1_status;
$out_json['SW2'] = $SW2_status;

```

```

$out_json['LED1'] = $LED1_status;
$out_json['LED2'] = $LED2_status;
$out_json['error'] = $error; //provide error (if any) number for debugging
echo json_encode($out_json); //encode the data in json format

```

?>

The code of sync_android_data.php

```

<?php
require_once __DIR__ . '/../required/db_connect.php';
$input = file_get_contents("php://input");
$error = 0;
$out_json = array();
$out_json['success'] = 1; //assume success
$SW1_status = 0;
$SW2_status = 0;
$LED1_status = 0;
$LED2_status = 0;
$check=0;
if ($input)
{
    $json = json_decode($input, true); //check if it json input
    if (json_last_error() == JSON_ERROR_NONE)
    {
        if (isset($json["username"]) && isset($json["password"]) && isset($json["SW1"]) &&
            isset($json["LED1"]) && isset($json["SW2"]) && isset($json["LED2"]))
        {
            $in_username = $json["username"];
            $in_password = $json["password"]; //if the expected fields are not null, get them
            $in_SW1 = $json["SW1"];
            $in_SW2 = $json["SW2"];
            $in_LED1 = $json["LED1"];
            $in_LED2 = $json["LED2"];
            if ($stmt = $mysqli->prepare("SELECT password FROM Users WHERE pname = ? LIMIT
1"))
            {
                $stmt->bind_param('s', $in_username);
                $stmt->execute();
                $stmt->store_result(); //store_result to get num_rows etc.
                $stmt->bind_result($db_password); //get the hashed password
                $stmt->fetch();
                if ($stmt->num_rows == 1)
                { //if user exists, verify the password
                    if (password_verify($in_password, $db_password))
                    {
                        $stmt->close();
                        if ($stmt = $mysqli->prepare("UPDATE SensorsAndActuators set DeviceStatus=?"
where DevID = 'LED1')) { //update LED1

```

```

        $stmt->bind_param('i', $in_LED1); $stmt-
>execute();
        if($in_LED1==1){
            $check=1;
        }
    }
    else {$error=1;}
    $stmt->close();
    if($check==1){
        if ($stmt = $mysqli->prepare("INSERT INTO
TransactionalLogs(TimestampInfo,MessageID,DataInformation) VALUES(?, ?, ?)")) { //Insert into log
            $id='13';
            date_default_timezone_set("America/Chicago");
            $date=date("Y-m-d H:i:s");
            $stmt->bind_param('sss', $date, $id,
$in_username);
            $stmt->execute();
        }
        $stmt->close();
        $check=0;
    }
    if ($stmt = $mysqli->prepare("UPDATE
SensorsAndActuators set DeviceStatus=? where DevID = 'LED2'")) { //update LED1
        $stmt->bind_param('i', $in_LED2);
        $stmt->execute();
        if($in_LED2==1){
            $check=1;
        }
    }
    else {$error=1;}
    $stmt->close();
    if($check==1){
        if ($stmt = $mysqli->prepare("INSERT INTO
TransactionalLogs(TimestampInfo,MessageID,DataInformation) VALUES(?, ?, ?)")) { //Insert into log
            $id='14';
            date_default_timezone_set("America/Chicago");
            $date=date("Y-m-d H:i:s");
            $stmt->bind_param('sss', $date, $id,
$in_username);
            $stmt->execute();
        }
        $stmt->close();
        $check=0;
    }
if (!$error && ($stmt = $mysqli->prepare("SELECT DeviceStatus FROM
SensorsAndActuators WHERE DevID = 'SW1'")))

```

```

{ //read SW1
    $stmt->execute();
    $stmt->bind_result($SW1_status);
    $stmt->fetch();
}
else
{
    $error = 2;
}
$stmt->close();
if (!$error && ($stmt = $mysqli->prepare("SELECT DeviceStatus FROM
SensorsAndActuators WHERE DevID = 'SW2'")))
{ //read SW1
    $stmt->execute();
    $stmt->bind_result($SW2_status);
    $stmt->fetch();
}
else
{
    $error = 2;
}
$stmt->close();
if (!$error && ($stmt = $mysqli->prepare("SELECT DeviceStatus FROM
SensorsAndActuators WHERE DevID = 'LED1'")))
{ //read LED1
    $stmt->execute();
    $stmt->bind_result($LED1_status);
    $stmt->fetch();
}
else
{
    $error = 3;
}
$stmt->close();
if (!$error && ($stmt = $mysqli->prepare("SELECT DeviceStatus FROM
SensorsAndActuators WHERE DevID = 'LED2'")))
{ //read LED2
    $stmt->execute();
    $stmt->bind_result($LED2_status);
    $stmt->fetch();
}
else
{
    $error = 3;
}
$stmt->close();
}
else
{

```

```

        $error = 4;
    }
}
else
{
    $error = 5;
}
}
else
{
    $error = 6;
}
}
else
{
    $error = 7;
}
}
else
{
    $error = 8;
}
}
else
{
    $error = 9;
}
if ($error)
{
    $out_json['success'] = 0; //flag failure
}
$out_json['SW1'] = $SW1_status;
$out_json['SW2'] = $SW2_status;
$out_json['LED1'] = $LED1_status;
$out_json['LED2'] = $LED2_status;
$out_json['error'] = $error; //provide error (if any) number for debugging
echo json_encode($out_json); //encode the data in json format
?>
```

the code of index.php

```
<?php
require_once __DIR__ . '/../required/db_connect.php';
?>
<html>
    <head>
        <title>Team Project</title>
```

```

<link rel="stylesheet" type="text/css" href="prjct.css"/>
</head>
<body>
<div id="Header">
    <h1>Monitoring System</h1>
</div>
<div id="body">
    <h3>Login</h3>
    <form id="form1" method="post" action="">
        <div>
            <label for="usernm">Username: </label>
            <input type="text" name="usernm" id="usernm" required><br><br>
        </div>
        <div>
            <label for="psswrd">Password: </label>
            <input type="password" name="psswrd" id="psswrd" required><br><br>
        </div>
        <button id="sub" type="Submit">Submit</button>
    </form>
</div>
<div id="check">
    <?php
    if(isset($_POST['usernm']) and isset($_POST['psswrd'])){
        $uname=$_POST['usernm'];
        $passwrd=$_POST['psswrd'];

        if($query = $mysqli->prepare("SELECT * FROM Users WHERE pname='".$uname."'")){
            $query->execute();
            $query->bind_result($username,$password);
            $count=0;
            while($query->fetch()){
                $count=$count+1;
            }
            if ($count == 1){
                if(password_verify($passwrd,$password)){
                    $query->close();
                    $id='9';
                    date_default_timezone_set("America/Chicago");
                    $date=date("Y-m-d H:i:s");
                    $stmt=$mysqli->prepare("INSERT INTO
TransactionalLogs(TimestampInfo,MessageID,DataInformation) VALUES(?, ?, ?)");
                    $stmt->bind_param('sss',$date,$id,$uname);
                    $stmt->execute();
                    $stmt->close();
                    echo "<script type='text/javascript'>alert('Login Credentials verified')</script>";
                    echo "<script> window.location.assign('welcome.php');</script>";
                }
            }
        }
    }
    //echo "Login Credentials verified";

```

```

echo "<script type='text/javascript'>alert('Login Credentials do not match')</script>";
}

}
else{
echo "<script type='text/javascript'>alert('Invalid Login Credentials')</script>";
//echo "Invalid Login Credentials";
}
}
else{
echo "Error in if 2";
}
}
?>
</div>
</body>
</html>

```

the code of welcome.php

```

<?php
require_once __DIR__ . '/../required/db_connect.php';
?>
<html>
<header>
<title>Welcome</title>
<meta charset="UTF-8"/>
</header>
<body>
<div id="header">
<h1>Welcome</h1>
</div>
<div id="body">
<div id="status"></div>
<script type="text/javascript" src="jquery.js"></script>
<script src="https://cdn.jsdelivr.net/npm/chart.js@2.8.0"></script>
<script type="text/javascript">
$(document).ready(function() {
    setInterval(function() {
        $('#status').load('DBdevice.php')
    }, 3000);
});
</script>
<form id="form1" method="post" action="">
<input type="hidden" id="out" value="1">
<button type="submit">Logout</button>
</form>
<div id=check>
<?php

```

```

if(isset($_POST['out'])){
    $uname='ben';
    $id='10';
    date_default_timezone_set("America/Chicago");
    $date=date("Y-m-d H:i:s");
    $stmt=$mysqli->prepare("INSERT INTO
TransactionalLogs(TimestampInfo,MessageID,DataInformation) VALUES(?, ?, ?)");
    $stmt->bind_param('sss',$date,$id,$uname);
    $stmt->execute();
    $stmt->close();
    echo "<script type='text/javascript'>alert('Logging Out')</script>";
    echo "<script> window.location.assign('index.php');</script>";
}
?>
</div>
<div id="chartLink"><a href="Chart.php">The link to the Timestamp chart</div>
</div>
</body>
</html>

```

the code of data.php

```

<?php
require_once __DIR__ . '/../required/db_connect.php';

$stmt = "SELECT MessageID, TimestampInfo FROM TransactionalLogs ORDER BY
TimestampInfo";
$result = mysqli_query($mysqli,$stmt);
$data = array();
foreach ($result as $row){
    $data[] = $row;
}
mysqli_close($mysqli);
echo json_encode($data);
?>

```

the code of DBDevice.php

```

<?php
require_once __DIR__ . '/../required/db_connect.php';
?>
<?php
date_default_timezone_set("America/Chicago");
echo "Last update: " . date('m/d/y') . " " . date('h:i:sa') . "<br>";
echo "<h3>Status of Sensors and Acutators</h3>";
if ($stmt=$mysqli->prepare("SELECT * FROM SensorsAndActuators LIMIT 100")) {
    $stmt->execute();
    $stmt->bind_result($devid,$devtype,$devfun,$ctrl,$status);
    echo "<table><tr><td>Device Name</td><td>Status</td></tr>";
}

```

```

while ($stmt->fetch()) {
    echo "<tr><td>$devid</td><td>$status</td></tr>";
}
echo "</table>";
$stmt->close();
}
else {
    echo "error";
    $mysqli->close();
}
echo "<hr>";
echo "<h3>Alarm Status</h3>";
if ($stmt=$mysqli->prepare("SELECT a1.AlarmID, a1.SinceTS,
a1.AcknowledgementStatus,c1.MessageDescription FROM ActiveAlarms a1 LEFT JOIN Alarms a2
ON a1.AlarmID=a2.AlarmID LEFT JOIN CannedMessages c1 ON a2.MessageID=c1.MessageID")) {
    $stmt->execute();
    $stmt->bind_result($alarmid,$since,$acknowl,$description);
    echo "<table><tr><td>Alarm</td><td>Description</td><td>Since</td><td>Acknowledged</td></tr>";
    while ($stmt->fetch()) {
        if ($acknowl!=1){
            echo "<tr><td>$alarmid</td><td>$description</td><td>$since</td><td>$acknowl</td></tr>";
        }
    }
    echo "</table>";
    $stmt->close();
}
else {
    echo "error";
    $mysqli->close();
}
echo "<hr>";
?>

```

The code of Chart.php

```

<html>
    <!--https://phppot.com/php/creating-dynamic-data-graph-using-php-and-chart-js/-->
    <div id="charts">
        <canvas id="chart1"></canvas>
    </div>
    <script src='https://cdn.jsdelivr.net/npm/chart.js@2.8.0'></script>
    <script src='jquery.js'></script>
    <script>
        $(document).ready(function (){
            showGraph();
        });
    function showGraph(){

```

```

{
    $.post('data.php',
    function(data){
        console.log(data);
        data=JSON.parse(data);
        var time=[];
        var mid=[];

        for (var i in data){
            time.push(data[i].TimestampInfo);
            mid.push(data[i].MessageID);
        }

        var chartdata={
            labels: time,
            datasets:[{
                label: 'LED Monitoring',
                borderColor: '#F778A1',
                data: mid
            }]
        };
        var graphTarget = $("#chart1");

        var linegraph = new Chart (graphTarget,{
            type: 'bar',
            data: chartdata,
            options:{ 
                scales:{
                    xAxes:[{
                        scaleLabel:{
                            display: true,
                            fontColor:'#6C2DC7',
                            labelString: 'Time Stamp'
                        },
                        ticks: {
                            fontColor:'#6C2DC7'
                        }
                    }],
                    yAxes:[{
                        scaleLabel:{
                            display: true,
                            fontColor:'#6C2DC7',
                            labelString: 'MessageID'
                        },
                        ticks: {
                            fontColor:'#6C2DC7',
                            suggestedMin:12,
                            suggestedMax:15
                        }
                    }]
                }
            }
        );
    }
}

```

```

        }
    }
});

}

</script>
</html>

```

the code of p2_t8.py

```

#!/usr/bin/python
import requests #import JSONRequests library
import time #import time library for sleep function
import datetime #import datetime library for timestamp
import RPi.GPIO as GPIO #import GPIO library
GPIO.setmode(GPIO.BCM) #set the pins according to BCM scheme
GPIO.setup(4,GPIO.OUT) #configure BCM Pin #4 as OUTPUT
GPIO.setup(6,GPIO.OUT) #configure BCM Pin #6 as OUTPUT
GPIO.setup(17,GPIO.IN) #configure BCM Pin #17 as INPUT
GPIO.setup(22,GPIO.IN) #configure BCM Pin #17 as INPUT
i=0; n=10; delay=5 #limit number of tries to 5 (initially set it to 1 for debugging)
while i<n:
    LED1=GPIO.input(4) #read what BCM Pin #4 is set to (LED1)
    LED2=GPIO.input(6) #read what BCM Pin #6 is set to (LED2)
    SW1=GPIO.input(17) #read the status of BCM Pin #17 (SW1)
    SW2=GPIO.input(22) #read the status of BCM Pin #22 (SW2)
    data = {'username': 'Test', 'password': 'Tester3', 'SW1': SW1, 'SW2': SW2, 'LED1': LED1,
'LED2': LED2}
    res = requests.post("https://team2project3342.online/scripts/sync_rpi_data.php", json=data)
    #in case of errors (especially, syntax) , you may want to print res.text and comment out the
statements below
    r = res.json()
    ts = datetime.datetime.now() #get the time stamp
    print "=====Server Response at " + str(ts) + "====="
    if r['success']==1:
        print "+++++Server request successful: "
        if LED1!=r['LED1']:
            print "Changing LED status as requested by the server"
            if r['LED1']==1:
                GPIO.output(4,GPIO.HIGH)
            else: GPIO.output(4,GPIO.LOW)
        if LED2!=r['LED2']:
            print "Changing LED status as requested by the server"
            if r['LED2']==1:
                GPIO.output(6,GPIO.HIGH)
            else: GPIO.output(6,GPIO.LOW)

```

```

        print "The status of LED1 is " + str(r['LED1'])
        print "The status of LED2 is " + str(r['LED2'])
        print "The status of SW1 is " + str(r['SW1'])
        print "The status of SW2 is " + str(r['SW2'])

    else:
        print ">>>> Server request failed - Error #" + str(r['error'])
        time.sleep(delay) #wait for delay seconds before sending another request
        i+=1
GPIO.cleanup()

```

the code of Projectv4.py

```

import time
import kivy
from kivy.app import App
from kivy.uix.button import Button
from kivy.uix.boxlayout import BoxLayout
from kivy.uix.label import Label
from kivy.uix.switch import Switch
from kivy.clock import Clock
from functools import partial
import requests

class BoxLayoutApp(App):

    #def __init__(self, **kwargs):
    #    super(BoxLayoutApp, self).__init__(**kwargs)
    def build(self):
        self.superBox = BoxLayout(orientation ='vertical')

        self.VB1 = BoxLayout(orientation ='vertical')

        self.lbl1 = Label(text="Monitoring System")
        self.VB1.add_widget(self.lbl1)

        localtime = time.asctime(time.localtime(time.time()))
        self.lbl2 = Label(text=localtime)
        self.VB1.add_widget(self.lbl2)

        self.HB1 = BoxLayout(orientation ='horizontal')

        self.lbl3 = Label(text="Systolic BP > 120mmHg:")
        self.HB1.add_widget(self.lbl3)

        self.sw1=Switch(active=False,
                       disabled=True)
        self.HB1.add_widget(self.sw1)
        #self.sw1.bind(active=switch_callback1)

```

```
self.HB2 = BoxLayout(orientation ='horizontal')
self.lbl4 = Label(text="Diastolic BP < 60mmHg:")
self.HB2.add_widget(self.lbl4)

self.sw2=Switch(active=False,
                disabled=True)
self.HB2.add_widget(self.sw2)
#self.sw2.bind(active=switch_callback2)

self.HB3 = BoxLayout(orientation ='horizontal')
self.lbl5 = Label(text="Administering Norepinephrine:")
self.HB3.add_widget(self.lbl5)

self.led1=Switch(active=False,
                 disabled=False)
self.HB3.add_widget(self.led1)
self.led1.bind(active=switch_callback3)

self.HB4 = BoxLayout(orientation ='horizontal')
self.lbl6 = Label(text="Administering Nitroglycerin:")
self.HB4.add_widget(self.lbl6)

self.led2=Switch(active=False,
                 disabled=False)
self.HB4.add_widget(self.led2)
self.led2.bind(active=switch_callback4)

self.HB5 = BoxLayout(orientation ='horizontal')
self.lbl7 = Label(text="Alarm1")
self.HB5.add_widget(self.lbl7)

self.settings5=Switch(active=False)
self.HB5.add_widget(self.settings5)
self.settings5.bind(active=self.switch_callback5)

self.lbl71=Label(text="On since: ")
self.HB5.add_widget(self.lbl71)
self.btn5 = Button(text="acknowledge")
self.btn5.bind(on_press=self.pressed2)
self.HB5.add_widget(self.btn5)

self.HB6 = BoxLayout(orientation ='horizontal')
self.lbl8 = Label(text="Alarm2")
self.HB6.add_widget(self.lbl8)

self.settings6=Switch(active=False)
self.HB6.add_widget(self.settings6)
self.settings6.bind(active=self.switch_callback6)
```

```

self.lbl81=Label(text="On since: ")
self.HB6.add_widget(self.lbl81)

self.btn6 = Button(text="acknowledge")
self.btn6.bind(on_press=self.pressed3)
self.HB6.add_widget(self.btn6)

self.HB7 = BoxLayout(orientation ='horizontal')
self.btn7 = Button(text="Dispense Norepinephrine")
self.btn7.bind(on_press=self.pressed1)
self.HB7.add_widget(self.btn7)
self.btn8 = Button(text="Dispense Nitroglycerin")
self.btn8.bind(on_press=self.pressed)
self.HB7.add_widget(self.btn8)
self.btn9 = Button(text="Dispense Auto")
self.btn9.bind(on_press=self.presseda)
self.HB7.add_widget(self.btn9)

self.superBox.add_widget(self.VB1)
self.superBox.add_widget(self.HB1)
self.superBox.add_widget(self.HB2)
self.superBox.add_widget(self.HB3)
self.superBox.add_widget(self.HB4)
self.superBox.add_widget(self.HB5)
self.superBox.add_widget(self.HB6)
self.superBox.add_widget(self.HB7)

#schedule the JSONrequest function to trigger every 5 seconds to read/write database
event = Clock.schedule_interval(partial(self.JSONrequest),5)

return self.superBox

def JSONrequest(self, *largs):
    if(self.sw1.active==True):
        SW1=1
    else:
        SW1=0
    if(self.sw2.active==True):
        SW2=1
    else:
        SW2=0
    if(self.led1.active==True):
        LED1=1
    else:
        LED1=0
    if(self.led2.active==True):
        LED2=1
    else:
        LED2=0

```

```

data={'username':'Test', 'password':'Tester3','SW1':SW1,'LED1':LED1,'SW2':SW2,'LED2':LED2}
res=requests.post("https://team2project3342.online/scripts/sync_android_data.php",json=data)
print(res)
r=res.json()
if(SW1!=r['SW1']):
    print("Changing SW1 status to the value in the database.")
    if self.sw1.active==True:
        self.sw1.active=False
        self.led1.active=False
    else:
        self.sw1.active=True
if(SW2!=r['SW2']):
    print("Changing SW2 status to the value in the database.")
    if self.sw2.active==True:
        self.sw2.active=False
        self.led2.active=False
    else:
        self.sw2.active=True
return

def pressed(self,instance):
    print ("you picked " + instance.text)
    self.led2.active=True

def pressed1(self,instance):
    print ("you picked " + instance.text)
    self.led1.active=True

def presseda(self,instance):
    print ("you picked " + instance.text)
    if self.sw1.active==True:
        self.led1.active=True
    else:
        self.led1.active=False
    if self.sw2.active==True:
        self.led2.active=True
    else:
        self.led2.active=False

def pressed2(self,instance):
    print ("you acknowleged Alarm1")
    self.settings5.active=False
    print("turning alarm1 off")
def pressed3(self,instance):
    print ("you acknowleged Alarm2")
    self.settings6.active=False
    print("turning alarm1 off")
def switch_callback5(self, switchObject, switchValue):
    print('Value of Alarm1: ', switchValue)

```

```

if self.settings5.active==True:
    ltm = time.asctime(time.localtime(time.time()))
    self.lbl71.text+=ltm
else:
    self.lbl71.text = "On since: "
def switch_callback6(self, switchObject, switchValue):
    print('Value of Alarm2: ', switchValue)
    if self.settings6.active==True:
        ltm = time.asctime(time.localtime(time.time()))
        self.lbl81.text+=ltm
    else:
        self.lbl81.text = "On since: "

def switch_callback1(switchObject, switchValue):
    print('Value of SW1: ', switchValue)
def switch_callback2(switchObject, switchValue):
    print('Value of SW2: ', switchValue)
def switch_callback3(switchObject, switchValue):
    print('Value of LED1: ', switchValue)
def switch_callback4(switchObject, switchValue):
    print('Value of LED2: ', switchValue)

#class myApp(App):
#    def build(self):
#        return BoxLayout()

if __name__=="__main__":
    myApp=BoxLayoutApp()
    myApp.run()
    # myApp().run()

```