RASPBERRY PI PROJECT

Erasmus+ Romanian students (Pitești)

Nicolescu Mihai-Robert
Marin Marian Puiu
Ene Remus Ionut
Nedelea Eugen Cristian
Oprea Alexandra Catalina
Year: 2
Lecturer: Irena Valova

Who has worked on what:

Nicolescu Mihai-Robert	- Setting up the RPI, Python script, Back-End, Documentation
Marin Marian Puiu	- Front-End, Back-End
Ene Remus Ionut	- Back-End
Nedelea Eugen Cristian	- Front-End
Oprea Alexandra Catalina	- Front-End

What we’ll talk about:

- What tech we’ve used
- Sensors
- Logic
- Website architecture
- Database
- Front - end
- Back - end:
    - Python
    - PHP sessions
    - Cookies
- Use cases
- Features

Tech we’ve used

Front-End:

- HTML
- CSS
- Bootstrap

Back-End:

- Apache2
- PHP
- MYSQL
- Python 3
- Javascript

Sensors - BH1750

This is a light sensor. It has a lux meter which can measure the light’s intensity without calculations.

We can say, for example, that when the light sensor reads more than 5lx we should turn on the lamp in the room because it’s dark. (for reference, 0lx means complete darkness).

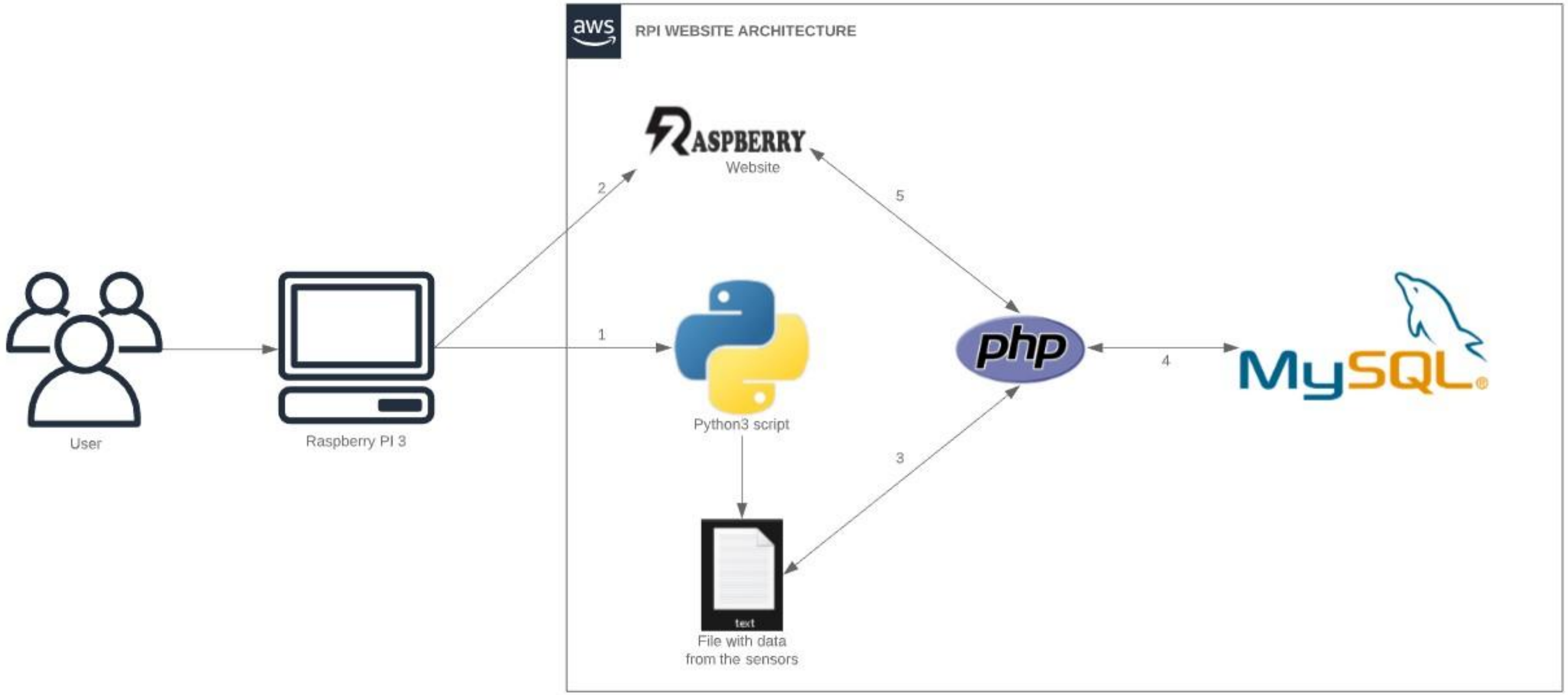
Sensors - BME280

The BME280 is as combined digital humidity, pressure and temperature sensor based on proven sensing principles.

Here’s an example of a reading from the sensor:

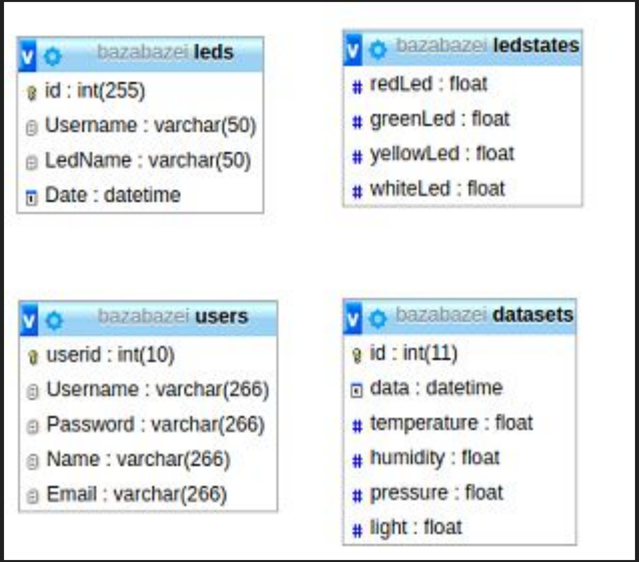
- humidity level 83.7%
- pressure level 999.1 hPa
- temperature 24.4C

The logic


RPI WEBSITE ARCHITECTURE
User
Raspberry PI 3
1
2
RASPBERRY
Website
Python3 script
File with data from the sensors
3
php
4
MySQL
5

The python script outputs the data we need every few seconds in a file(to avoid data repetition in the datasets table when using the site rapidly).
Then, using PHP we put that data in a table in the local database, and then we show the table on the page.
The website compares the most recently read data from the table with the bounds set by admin and decides if the LEDs should turn on or off.
Data is inserted into the table every time the page refreshes, so we told the page to refresh every 300 seconds (5 minutes).

Database



```
bazabazei leds
id : int(255)
Username : varchar(50)
LedName : varchar(50)
Date : datetime

bazabazei ledstates
redLed : float
greenLed : float
yellowLed : float
whiteLed : float

bazabazei users
userid : int(10)
Username : varchar(266)
Password : varchar(266)
Name : varchar(266)
Email : varchar(266)

bazabazei datasets
id : int(11)
data : datetime
temperature : float
humidity : float
pressure : float
light : float
```

PHP puts data in the MYSQL tables.

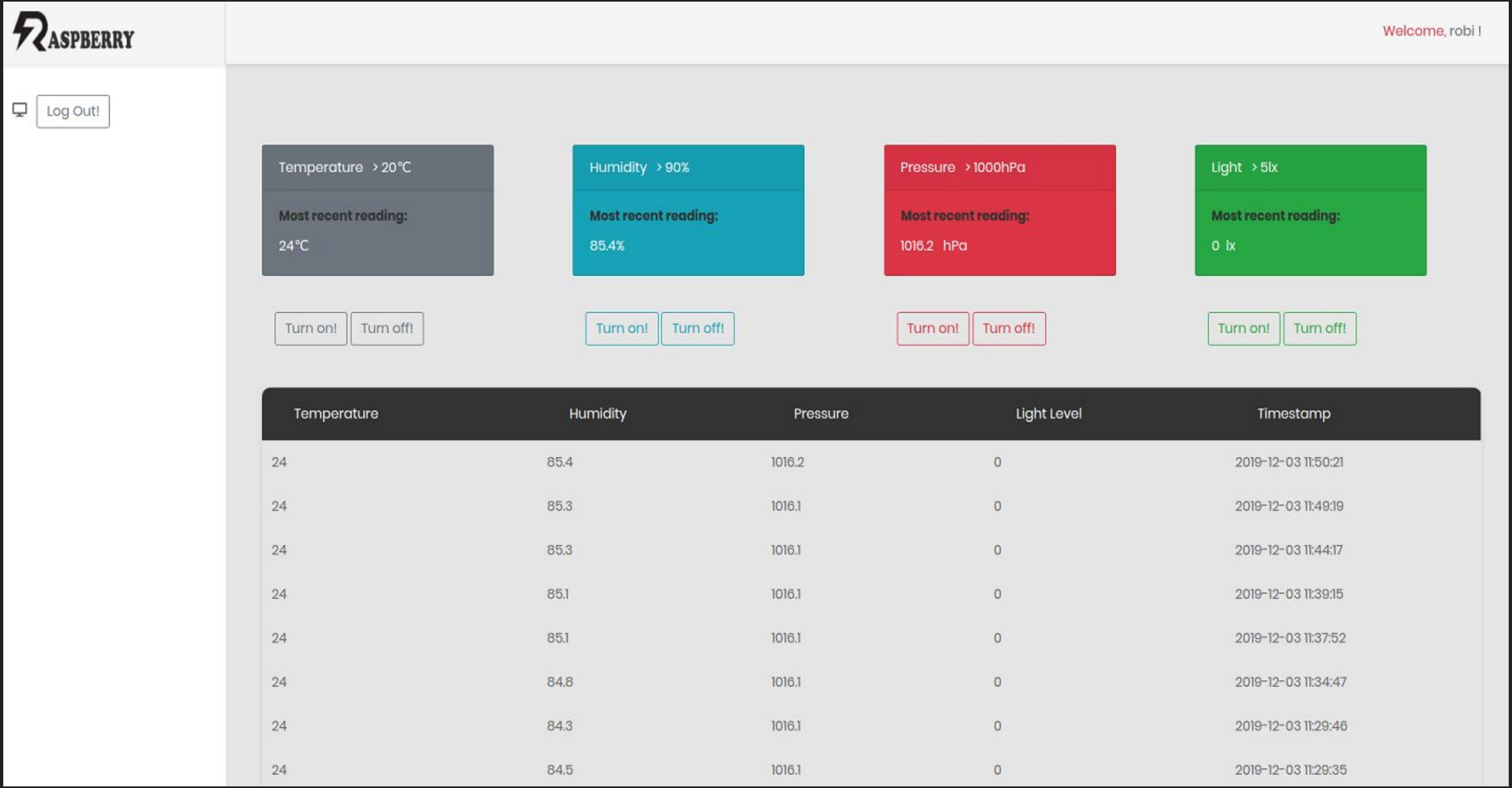
Because we didn’t have any complicated query to do, we’ve decided that we have no reason to connect the tables.

- Users has info about the users
- Datasets keeps track of the data read by the sensors
- Ledstates keeps track of the bounds set by the admin for the sensors
- Leds keeps track of who turned on/off a LED

Front-End

For the Front-End, we’ve decided to go with a simple and clean bootstrap template. Being easy to use, we didn’t have a lot of problems getting it to look exactly how we wanted.

We’ve also decided that the most relevant data to show on the page are the most recent 10 readings of the sensors.


RASPBERRY

Welcome, robi !

Log Out!

Temperature > 20°C

Most recent reading:

24°C

Humidity > 90%

Most recent reading:

85.4%

Pressure > 1000hPa

Most recent reading:

1016.2 hPa

Light > 5lx

Most recent reading:

0 lx

Temperature	Humidity	Pressure	Light Level	Timestamp
24	85.4	1016.2	0	2019-12-03 11:50:21
24	85.3	1016.1	0	2019-12-03 11:49:19
24	85.3	1016.1	0	2019-12-03 11:44:17
24	85.1	1016.1	0	2019-12-03 11:39:15
24	85.1	1016.1	0	2019-12-03 11:37:52
24	84.8	1016.1	0	2019-12-03 11:34:47
24	84.3	1016.1	0	2019-12-03 11:29:46
24	84.5	1016.1	0	2019-12-03 11:29:35

Back-End - general info

The server runs on apache2 combined with PHP/MYSQL. These were installed using the terminal.

Back-End - Using Python #1

Reading the data from BME280 was pretty easy (with the help of import adafruit_bme280), needing only 3 lines of code:

```
    i2c = busio.I2C(board.SCL, board.SDA)
    bme280 = adafruit_bme280.Adafruit_BME280_I2C(i2c)
    list = [bme280.temperature, bme280.humidity,
bme280.pressure]
```

Back-End - Using Python #2

Solution() puts the list of data into a string where the numbers are separated by a comma (e.g: 11.67,23.6,62.9,1010.6) and the lines of code afterwards put this string into a file every 3 seconds.

- 11.67 is the light level
- 23.6 is the temperature
- 62.9 is the humidity
- 1010.6 is the pressure

```
def solution():
    list = SensCheck()
    my_string = ','.join(map(str, list))
    return(my_string)
```

```
def fisier():
    f=open('date.txt',"w+")
    lista = solution()
    #print(lista)
    f.write(lista)
    f.close()

while (True):
    fisier()
    time.sleep(3)
```

Back-End - PHP

When someone registers, their info gets added to the users table.
Also, the user’s password is hashed in the database.

Back-End - PHP Register

```
<?php
require 'conectare.php';

if(!empty($_POST['Name'])&&
!empty($_POST['Password'])&&
!empty($_POST['Email'])&&
!empty($_POST['Username'])&&
    isset($_POST['Name'])&&
isset($_POST['Password'])&&
isset($_POST['Email'])&&
isset($_POST['Username']))
{

    $username =
strtolower($_POST['Username']);
    $password = $_POST['Password'];
    $email = $_POST['Email'];
    $name = $_POST['Name'];
    $password_hashed =
password_hash($password, PASSWORD_DEFAULT);
```

```
$sql = "SELECT Username from users where
Username='$username'";
    $result = mysqli_query($conectare, $sql);
    $check = mysqli_num_rows($result);

    if($check>0){

header("Location:indexsignup.php?Info=exista");
        die();
    } else{

    $sql = "INSERT INTO
users(Username,Password,Email,Name) VALUES
('$username' , '$password_hashed' , '$email' ,
'$name')";

    $result = mysqli_query($conectare, $sql);
    header("LOCATION:indexsingup.php?Info=ok");
    }
}
else {
    header("LOCATION:indexsingup.php?Info=eroare");
}
```

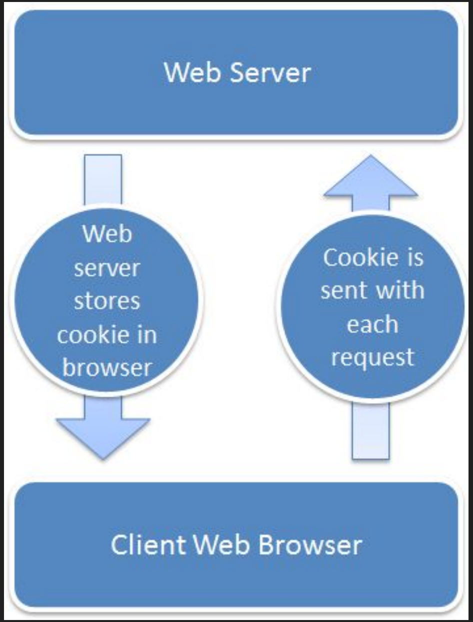
Back-End - PHP Sessions

When someone logs in, a session starts up containing the user’s name.
This is used at the top-right of the page to show who’s logged in.

And when you log out of the site, the session is destroyed.

Back-End - PHP Cookies

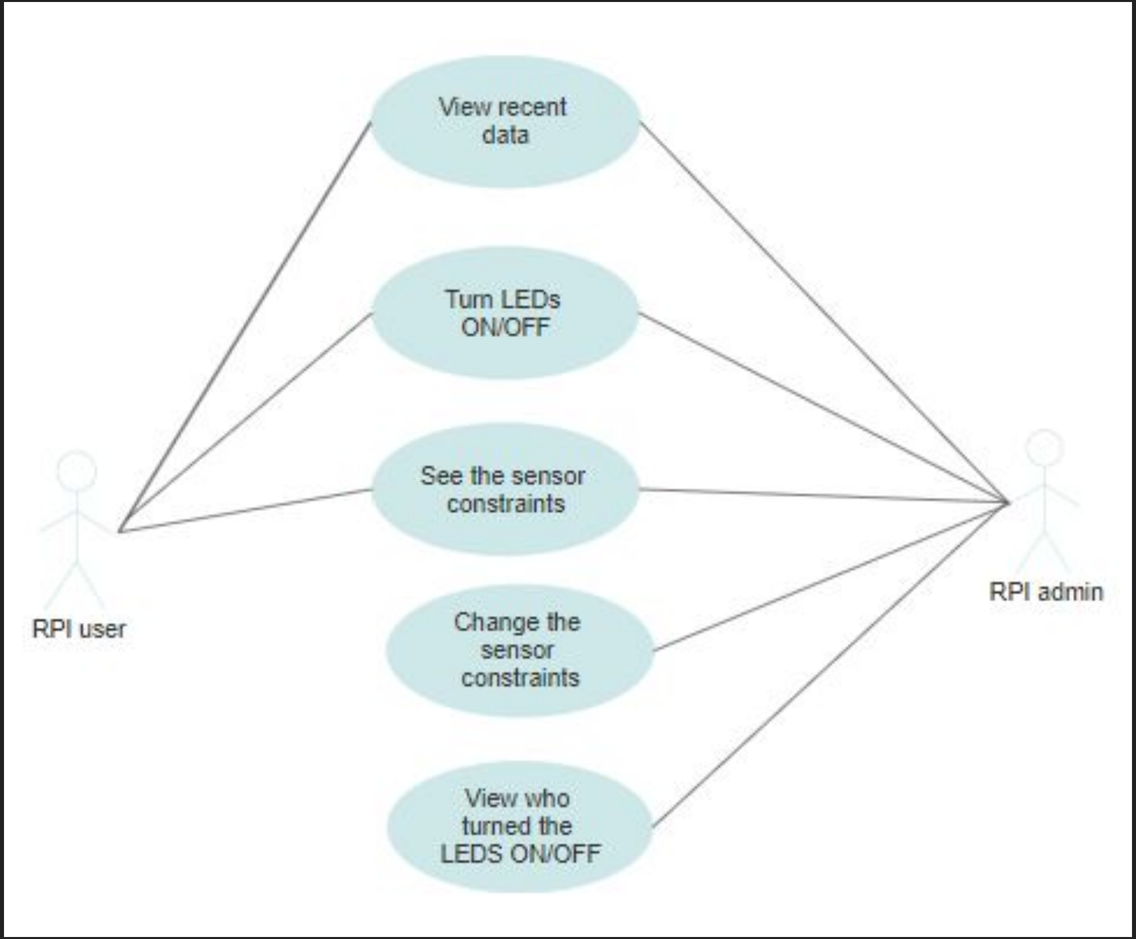
The most interesting thing about the back-end is that we’ve used cookies to be able to turn multiple leds on/off.
The process isn’t complicated at all, we basically have a cookie set for every led (it keeps track of its current state).
When you login, all the cookies are “unset”.
You can “set” them by pressing the turn on/off buttons. While the cookie is set, the specific led doesn’t turn on/off until you press the reset button.
Also, we’ve made the site in such a way that when you log out of the site, the leds are automatically unset.


Web Server
Web server stores cookie in browser
Cookie is sent with each request
Client Web Browser

Use cases

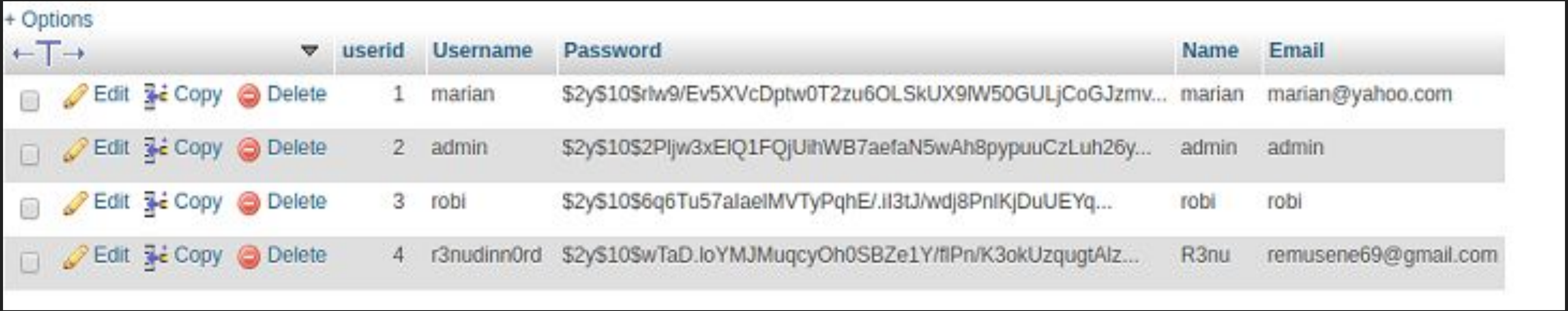
As seen here, there are some differences between what the normal user and the admin can do.

Mainly, the admin can change when the sensors turn on and see who turned the sensors on/off.


View recent data
Turn LEDs ON/OFF
See the sensor constraints
Change the sensor constraints
View who turned the LEDS ON/OFF
RPI user
RPI admin

Features - 1

The possibility of having separate admin and user account with hashed passwords

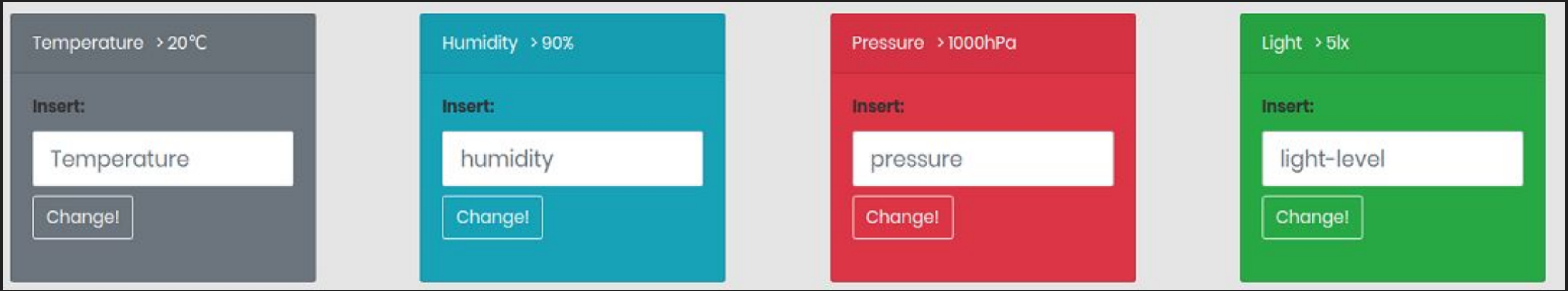


+ Options

←T→	userid	Username	Password	Name	Email
Edit Copy Delete	1	marian	$2y$10$rlw9/Ev5XVcDptw0T2zu6OLSkUX9lW50GULjCoGJzmv...	marian	marian@yahoo.com
Edit Copy Delete	2	admin	$2y$10$2Pljw3xElQ1FQjUihWB7aefaN5wAh8pypuuCzLuh26y...	admin	admin
Edit Copy Delete	3	robi	$2y$10$6q6Tu57alaelMVTyPqhE/.Il3tJ/wdj8PnIKjDuUEYq...	robi	robi
Edit Copy Delete	4	r3nudinn0rd	$2y$10$wTaD.loYMJMuqcyOh0SBZe1Y/flPn/K3okUzqugtAlz...	R3nu	remusene69@gmail.com

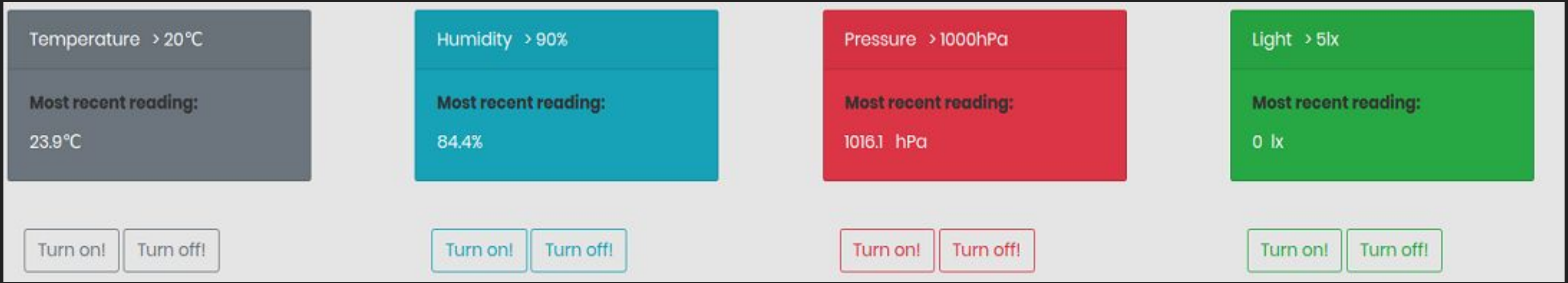
Features - 2

Change the bounds at which the sensors turn on (as an admin)



Features - 3

Turn the leds on/off until you reset them



Temperature > 20°C

Most recent reading:

23.9°C

Humidity > 90%

Most recent reading:

84.4%

Pressure > 1000hPa

Most recent reading:

1016.1 hPa

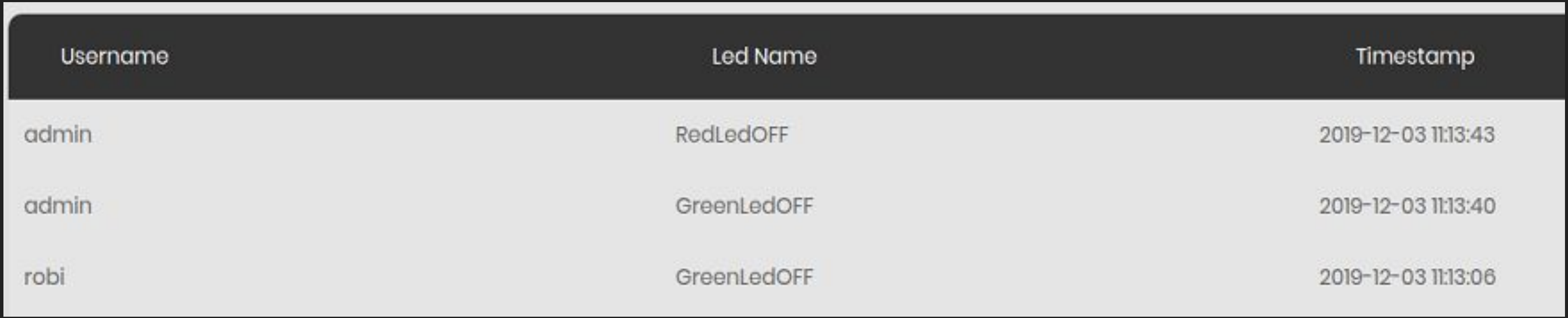
Light > 5lx

Most recent reading:

0 lx

Features - 4

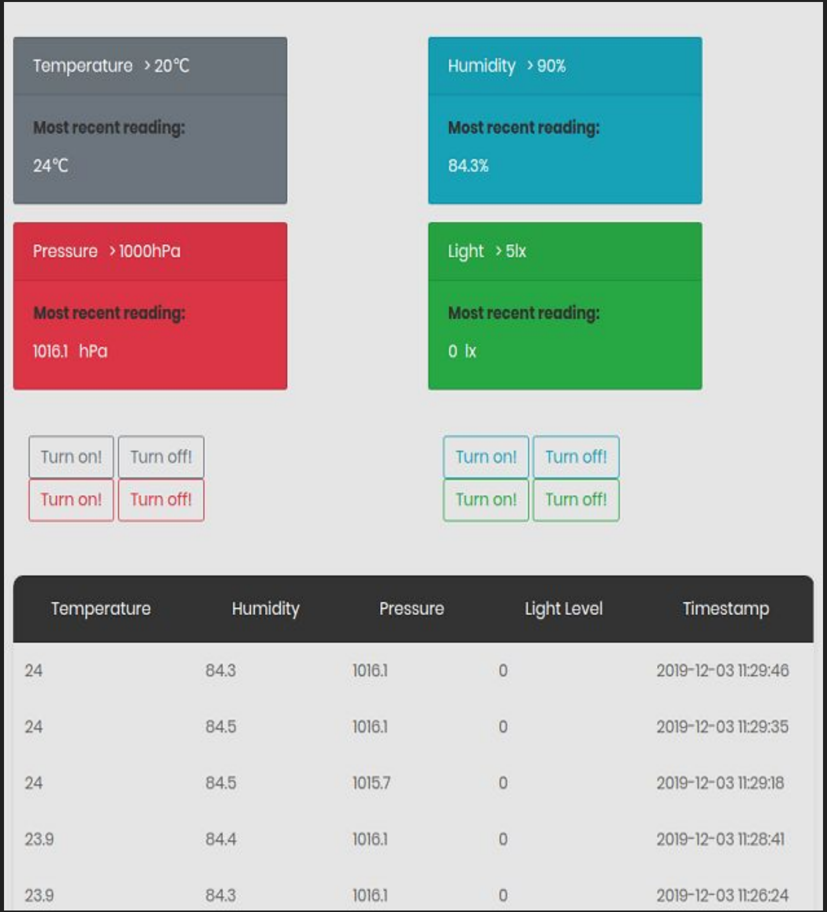
Know who turned leds on/off using PHP sessions



Username	Led Name	Timestamp
admin	RedLedOFF	2019-12-03 11:13:43
admin	GreenLedOFF	2019-12-03 11:13:40
robi	GreenLedOFF	2019-12-03 11:13:06

Features - 5

With the help of bootstrap, the website is mobile-ready



Temperature >20°C

Most recent reading:

24°C

Humidity >90%

Most recent reading:

84.3%

Pressure >1000hPa

Most recent reading:

1016.1 hPa

Light >5lx

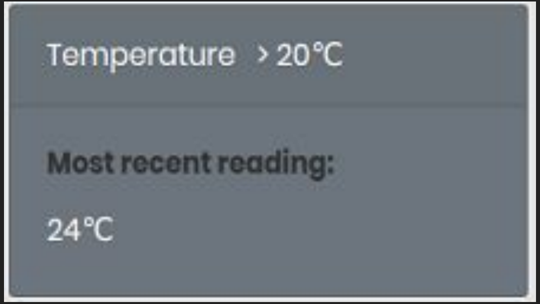
Most recent reading:

0 lx

Temperature	Humidity	Pressure	Light Level	Timestamp
24	84.3	1016.1	0	2019-12-03 11:29:46
24	84.5	1016.1	0	2019-12-03 11:29:35
24	84.5	1015.7	0	2019-12-03 11:29:18
23.9	84.4	1016.1	0	2019-12-03 11:28:41
23.9	84.3	1016.1	0	2019-12-03 11:26:24

Features - 6

See the current set bounds as any user (e.g. here the temperature has to be higher than 20°C to turn on).



Temperature > 20°C

Most recent reading:

24°C

Thank you for your attention!