ASHESI

ASHESI UNIVERSITY

THE DESIGN OF A LOWCOST REMOTE HEALTH MONITORING SYSTEM

CAPSTONE

BSc. Computer Engineering

Joshua Kasirye

2019

ASHESI UNIVERSITY

THE DESIGN OF A LOWCOST REMOTE HEALTH MONITORING SYSTEM

APPLIED

CAPSTONE PROJECT

Capstone Project submitted to the Department of Engineering, Ashesi University

in partial fulfillment of the requirements for the award of Bachelor of Science degree in

Computer Engineering

Joshua Kasirye

2019

DECLARATION

I hereby declare that this capstone is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……………………………………………………………………………………………

Candidate's Name:

……………………………………………………………………………………………

Date:

……………………………………………………………………………………………

I hereby declare that the preparation and presentation of this capstone were supervised in accordance with the guidelines on supervision of capstone laid down by Ashesi University.

Supervisor's Signature:

……………………………………………………………………………………………

Supervisor's Name:

Acknowledgments

To my supervisor, friends, doctors, Engineering faculty and all other people whose encouragement and academic advice helped me undertake this project, I thank you all for the support and constant feedback you provided during this project. This project wouldn't have been a success without your constant feedback, insights, and guidance. Thank you very much.

Table of Contents

List of Figures

Abstract

Accessing remote health care from public hospitals in Ghana has become more difficult and expensive recently due to the increasing number of patients that surpass the low number of doctors in the country. Many people have lost their lives waiting in long vital checking lines just to take their vitals in some hospitals. However, with the help of the Internet of Things (IoT), this report provides a design of a low-cost remote health monitoring system for patients with chronic diseases who need frequent health monitoring. This system was designed and fabricated using a Wi-Fi module on a microcontroller, a temperature sensor, and a web application. It provides a more convenient and cost-effective remote transmission of health vitals to public hospital administrators. During testing, a patient logged on to the web application and took the vitals using the system's temperature sensor. The new temperature readings were received by the hospital administrator successfully at very low costs with total convenience which made the system fully functional as intended. The system will be further developed by adding more medical sensors to make it able to capture other more vitals. As better technologies will be integrated into the proceeding versions of this system, in the long run, it will improve on the quality of remote health care provided in the country.

Chapter 1 : Introduction

1.1 Background

People who live in remote harsh environments experience the worst in accessing public health and personalized medication. The situation worsens for post-operative patients and other patients with chronic diseases like blood pressure, and stroke because they are expected to take their health vitals at the hospital almost daily for detecting any abnormalities. According to the 2017 Ghana epidemiological report [1], Kumasi public hospitals had an adult (50 -59 years) mortality rate of 19.3 percent mostly due to non-communicable diseases. This report highlights the threat posed by noncommunicable diseases to average Ghanaians if not well monitored and treated in time. Good human health monitoring refers to the use of health care sensors to accurately detect and analyze patients' primary vitals (body temperature, blood pressure, and heart rate) to eliminate any unhealthy threats on the patient's life [2].

1.2 Problem Definition

Owing to the inaccessibility of public health services in remote areas, there is a need for an effective and efficient medical monitoring system that is very reliable. A system that will work towards improving ease of access to patients' medication and health monitoring regardless of the patients' location. With this system set up in place and fully functional, it will improve the public health care system in Ghana's most public hospitals.

1.3 The Objectives of the Project work

This project is focused around design and development of an accurate, reliable, convenient and affordable remote health monitoring system. A system that will work towards improving on easy access to patients' health vitals at the public hospital. The system should increase the efficiency at the hospital given the low number of medical doctors in most public hospitals in Ghana [3]. In so doing, the system will be reducing transport and time costs incurred by patients in constantly taking the health vitals at distant hospitals and improving on the health monitoring of post-operative and chronic diseases patients. In case of any emergency on the patients' side, the system should provide all the relevant information about the patient's vitals history for immediate analysis and medication. And lastly, this system should be very affordable to an average Ghanaian cheaper to use.

1.4 Expected Outcomes of the Project Work

At the end of this project, a fully functional prototype of this system having a health vitals (temperature and or heart rate) sensing device supported with a software (web application) interface for both the hospitals' administrators and patients should be fabricated and tested. The system will not share patients' personal information or health data to any other third party unless requested by the person through the patient.

1.5 Motivation for the project topic

This amazing project was proposed by my Networks and Data communication lecturer, Mr. Gatsi Francis. He had this idea for a while hoping to realize it soon. Fortunately, I have similar interests in improving healthcare in public hospitals, so, he entrusted me with his idea for realization. I also have a special interest in using technology to leverage the power of big data to improve health care performance in public hospitals in general and Internet of Things (IoT) in particular. This project is to help me hone the skills I have gathered so far during my stay at Ashesi and learn more about IoT.

Chapter 2 : Related Work

Remote health monitoring is one of the trending technologies used in modern medical systems to improve personalized health monitoring. Marzencki's research paper shows developed technologies that provide a central communication support hub, for both stationary and wearable patient sensors, mostly for people in, remote, harsh environments with challenges associated with accessing personalized medication and public health [4]. His paper also suggested the use of a body sensor network to accurately capture the health vitals and transmit them remotely using the TCP/IP protocol and Wireless Sensor Networks (WSNs). Their proposed architecture of the system was fully autonomous and monitored the patients' health vital, the environment data and activity data to provide a more accurate assessment of the patient health status. This paper has provided more relevant information on how to improve the accuracy of the data being read from the patient's body by the used of several sensors to read for the same health vital.

The limitation with this architecture is the cost that might be incurred if several sensors, routers for WSNs were to be purchased. But to borrow some features from it, the system that is self-configured and autonomous will improve on the reliability of the low-cost remote health monitoring system this report is addressing. Ogawa's paper described the relevance of constant accurate monitoring of the patients' vitals in reducing the need for medical services [5]. This concept takes into consideration of the fact that if a system of this kind is developed, it should be able to retrieve the vitals irrespective of the location of the patient, it suggests taking more accurate vital data whenever or in times of convenience of the patient, it could be during bath or sleeping. This paper informs more about how more portable the remote system sensing device should be to in order to create a state of convenience whenever a patient is to take his/her health vitals.

Looking at both papers, [4] and [5], they highlight the importance of remote health monitoring irrespective of the location of the patient and the time of day at which the data is being read. They also spot the relevance of using wireless communication systems to transfer vital information to the doctors, because of its effectiveness in terms of higher data transmission speeds and low costs. In the modern market, products like Samsung gear, polar and apple watches use some of these technologies in providing personalized health monitoring. People using these devices can share their heart rate data and body temperature with their personal doctors for feedback and assessment.

Figure 2.1: Polar heart rate monitor and apple watch.

These are wonderful devices, but in a country like Ghana where the ratio of doctors to patients is one to eight thousand [3], this could not be the ideal solution but rather the rare solution because of the shortage of doctors. Besides that, research from some cardiologist shows that sensor reading from these watches are less accurate to what it expected by the doctors using the actual medical instruments [6]. Doctors point out that this is so simply because when data is obtained when the arms are in motion more noise is picked up as part of the signal, and if the device is not so good at filtering noise, it may affect the final value being measured.

Currently, patients in some public hospitals in Ghana can send their blood pressure and other health vitals to their doctors via WhatsApp messenger mobile application. This has been working until the number of patients increased, and doctors cannot monitor them all via their WhatsApp application. Doctors also mentioned that sometimes data gets lost when the patient's or doctor's phone gets missing. From five interviews I conducted with doctors from the West Africa International Health Summit (WAIHS) in February this year, the doctors emphasized the need of a platform to enable storing patient health vitals conveniently for future reference.

Bhatti's paper explores a similar idea to what this project is addressing, but for a real-time ECG system for home-bed monitored patients [7]. The paper describes the use of Arduino micro-controller and different heart rate and temperature sensors to sense and transmit data to distant hospitals in Realtime. This is the closest project similar to what my project is working on. The system was developed successfully to improve on the accuracy of data being transmitted from patients on beds (patients who are either hospitalized or cannot do their daily activities) in remote areas. The ECG readings were real time and required more power to keep the system running all the time. The system was so complex that it could only be operated by an expert which is not the case in the project this report is representing. The system also required the use of an oscilloscope which is quite expensive hence making the system more expensive.

Remote health monitoring is quite a wide problem space. This project seeks to combine the relevance of constant accurate monitoring of patient's vitals which reduces the need of medical services at hospitals, with wireless transmission of health vitals data to the hospitals subscribed to by the patients with total convenience. Considering the most pressing challenges like patient health monitoring in Ghana's public hospitals, patients incur transport and time costs when they visit hospitals from long distances and have to wait in long queues for just their vitals to be taken, and then book a meeting with the doctor for the following day or week. Wireless transmission of accurate vital data of patients to hospitals is necessary because it saves time and improves on the accuracy of the data being received since it was recorded at the patient's convenient time and place.

With this remote health monitoring system, patients will be taking measurements at their convenient times either at work or home and then send their vitals to the data server of the hospital they subscribed to using their mobile phone as a gateway. At this point, the health personnel at the hospital monitors the trends of the received data periodically, from which recommendations and feedback are given. Accuracy is improved when the vitals are measured while the person is calm in a place of convenience. This system will be relatively cheaper and accessible by using relatively cheaper micro-controllers and sensors that are also easily replaceable when faulty.

This system will increase the revenue of the hospital by selling these devices. The hospital will also have instant access to patients' data and vitals, thereby ensuring immediate health care in case of an emergency. This will eliminate the lengthy clinical decision often made, hence improving efficiency at the hospital. Patients (post-operative and chronic diseases patients) get an added advantage of reduction on the transport cost of frequent visits to hospitals for vitals measurements.

Chapter 3 : Design

3.1 Review of Existing Designs

After intensive research about systems providing the same functionalities like the one this project is developing, to the best of my knowledge, none like it exists in Ghana so far. Devices that have some similar functionalities include the modern device mentioned in chapter 2 Figure 2.1. These devices are portable and designed for specific functions and reasons. The polar device was designed for sportsmen. It is worn around the chest so that it can get more accurate data. It has a silicon rubber housing to improve on its durability.

On the other hand, the apple watch heart rate app was also mostly designed for sportsmen; people who engage in exercises regularly. It is worn as a watch hence the device has more value than the polar device. But the polar device is more accurate than the apple watch due to the position of the sensor. The common feature or technology that is used by these two devices is the use of the PPG (Photo Plethysmography) sensors. These sensors use light-based technology to sense the rate of blood flow as controlled by the heart's pumping action at special parts of the body like the chest, fingertips, and ears.

These devices are very expensive for an average Ghanaian, and their reported data is often shared with the person’s personalized doctor, which is another problem since an average person cannot afford personalized health considering the low number of doctors in the country.

3.2 System Design Objectives

After a requirement analysis from both patients and literature review, an accurate, reliable, portable and low-cost system will be preferable. The sensing device should use low energy and low-cost sensors and micro-controller. The sensors to be used should be more accurate. Lastly, the sensing device should be simple to operate and understood by the operator. Both the patients and hospital administrators should have a web application where they login and take vitals or get a report from the previous measurements and share their vitals report with the administrators. Therefore, a mobile gateway will be needed to enable connectivity between the sensing device and the remote server.

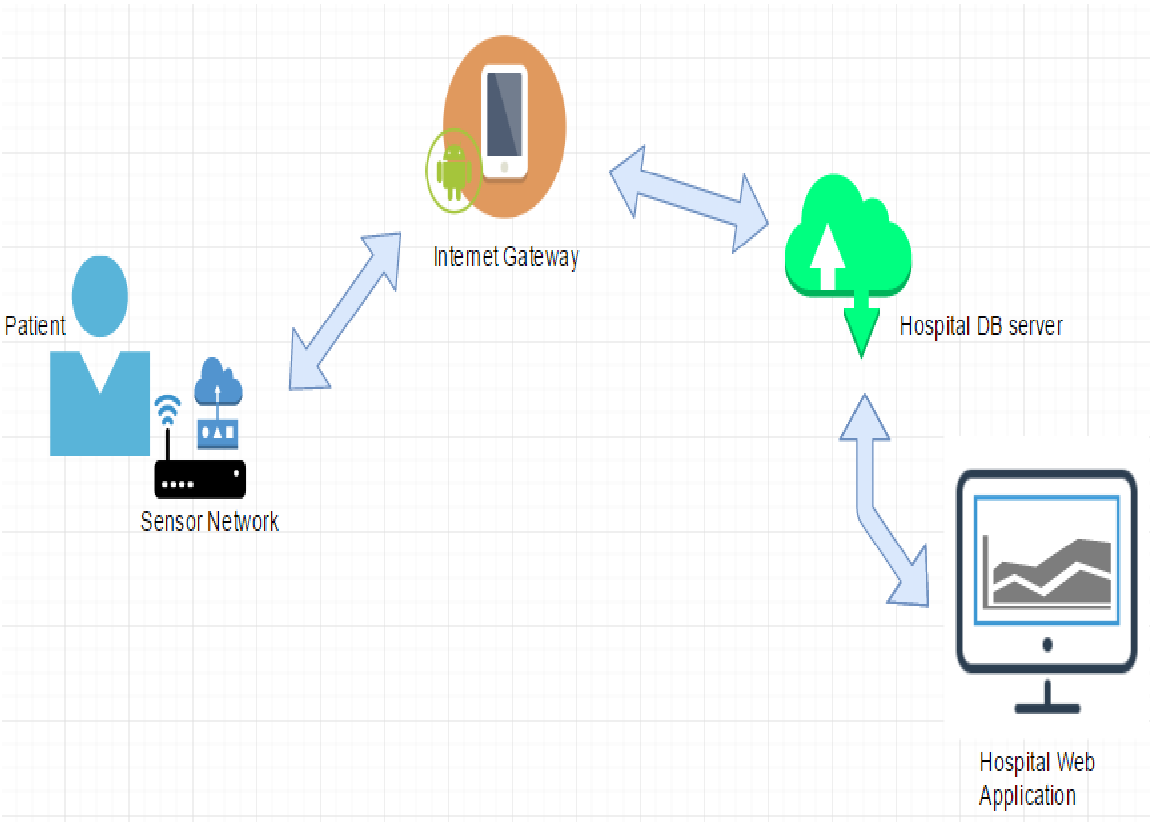


Figure 3.1: Systems functional diagram.

Figure 3.1 Shows the functional diagram of the proposed design objectives. There will be two-way communication between the systems sensing device and the mobile gateway with the help of an internet connection between the two. The sensed information will be transferred to the hospital database and later displayed on the web application accessed by both users through the mobile phone as a gateway.

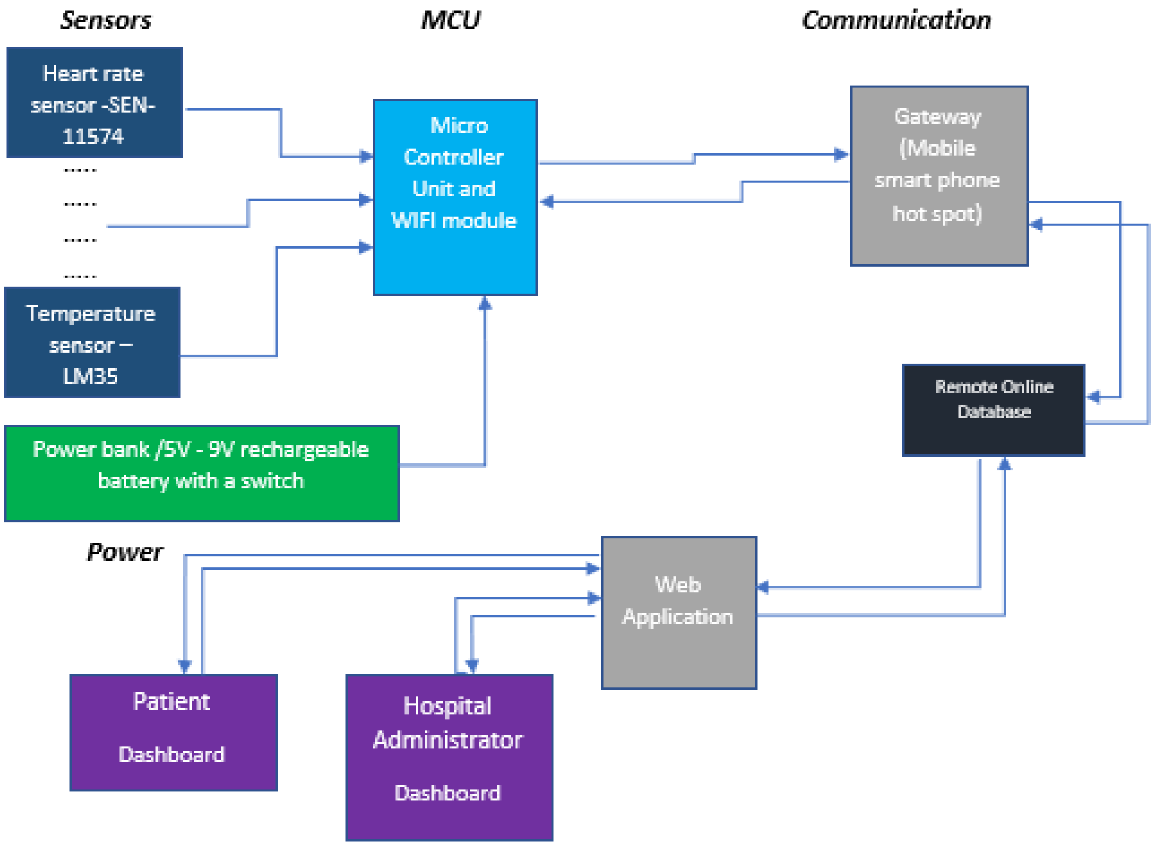


Figure 3.2: System's block diagram.

Figure 3.2 provides a proposed block diagram of what is expected to be the setup of the different components that will be used in developing the monitoring system. It is in three sections, the Sensing section that will account for the different sensors that will be used to get the patient’s vitals. The other section is the micro-controller or signal processing section were the sensed data would be processed into meaningful information for transmission. The micro-controller will use a Wi-Fi module to provide a secure connection between the mobile phone (gateway) and the device for easy vital data transmission to the hospital servers or web application servers. The last section is a power supply unit that will be used to provide all the necessary power to the sensing device for at least two successful transmissions.

3.3 Design Decisions

In this section, the Pugh matrix tool was used to make more informed and well thought out decisions on the sensor device design, micro-controller design, and power design, based on secondary research data from the different manufacturers' data sheets. A Pugh matrix involves using the selection criteria (Often user requirements), and secondary research data to evaluate a choice of material or tool using a scoring matrix. The selection criteria were obtained from a list of user requirements reduced to four most important requirements stated in section 3.2 above. Using the weight of each criterion, every positive means better than, negative means worse than and zero for equal weight among choices.

3.3.1 Sensors Pugh Matrix

Table 3.1: Sensors Pugh matrix

		Pulse sensors		**Temperature sensors**	
Criteria	**Weight**	**SEN - 11574**	**Electronic Stethoscope**	**LM 35**	**TMP35A**
1. Accuracy	5	0	0	-5	+5
2. Reliability	4	+4	4	0	0
3. Portability	3	+3	-3	0	0
4. Cost	2	0	0	+2	-2
TOTAL		**+7**	+1	-3	**+3**

While filling in the scoring matrix for the different pulse sensor choices, factors like; sensor input and output voltages, range of temperatures of operation, and user-friendliness while using it, were also considered. The secondary research data obtained from the different sensors' data sheets, was used to fill in the pulse sensors Pugh matrix. A conclusion was made that the SEN-11574 temperature sensor was more suitable to work with. Simply because of the wider temperature range of operation from -40 to +85 degrees Celsius, lower input current and voltage required for operation, the availability of enough information and resources about it, and finally a higher score on the Pugh matrix.

For the temperature sensor choices, factors like sensor input, and output voltages and range of temperatures measured were also considered while scoring the Pugh matrix. After filling the temperature sensor Pugh matrix, a conclusion was made that the TMP35A temperature sensor was more suitable to work with. Simply because of the wider temperature ranges measured ranging from -60 to +160 degrees Celsius, lower input current and voltage required for operation, the availability of enough information and resources about it, and finally a higher score on the Pugh matrix.

3.3.2 Micro-controller Unit Pugh Matrix

Table 3.2: Microcontroller unit Pugh matrix

Criteria	**Weight**	**Esp32**	**Arduino Uno**	**Raspberry Pi**
1. Accuracy	5	0	0	+5
2. Reliability	4	0	0	0
3. Portability	3	+3	-3	-3
4. Cost	2	+2	-2	-2
TOTAL		**+5**	-5	0

From the MCU Pugh matrix table above, the ESP32 micro-controller was chosen because of its accuracy, lower purchase cost and high processing power considering the computation to be done in this project. It is relatively easy to work with as compared to the Raspberry Pi, and it has most of the sensors and communication modules information as open source online. An extra advantage with the esp32 is the fact that it has a Bluetooth low energy module and a WIFI module part of it. This makes the use of a mobile phone as a gateway very possible and simple with the help of an application.

3.3.3 Power requirements for each component

Power Source Requirements.

Table 3.3: Power requirements for each component

Components	Voltage (V)	Current (mA)
1. ESP32	3.3 – 5.0	40
2. SEN-11574	3.0 – 5.0	4
3. TMP35A	5.0	0.1
4. HC-06	5.0	30 - 40
5. LEDs	3.5	30
TOTAL	**5V**	**115**

From the power requirement table above, the total voltage which will be required by the system is 5 to 9 volts.

Owing to the portability and reliability requirements, it is more feasible to use a rechargeable battery whose voltage and current value will depend on the total voltage to be supplied to the system.

3.3.4 Web application technologies

For the web application design, technologies like HTML 5, JavaScript, MySQL, PHP, and Bootstrap are the ideal technologies to use for a good looking, and effective web application as explained below. HTML5 is a user-friendly language in programming web pages; it also has more features like allowing scripting in HTML files as compared to the previous versions. JavaScript will be used for all animations that may be needed in the web application because it is also user-friendly and easy to understand if you have prior knowledge in C programming. For the client application and servers communication; which will happen among the ESP32 server and its client application, the web application server and its client application, MySQL language in PHP will be suitable for this because they are very user-friendly high-level languages. They can also be used to ease communication between different server types. Bootstrap will provide several templates with unique and purpose-specific dashboards for the web application.

Chapter 4 : Implementation

4.1 Sensing device

For better development of the system, one vital data was used for implementation and testing, and this was on the assumption that is the system fully functions well with one health vital, it means it can work with the integration of other sensors. So, the temperature sensor, TMP35 was connected to the esp8266 micro-controller as shown in Figure 4.1 below. ***NOTE***: It was difficult to ship in the esp32 in time before the close of this project, so the next best option which is the esp8266 was used to develop the system.

Two LEDs (Red and Green) were also connected to pins D2 and D5 of the micro-controller each protected by a 500 Ohm resistor from the 3.3V voltage supply by the micro-controller.

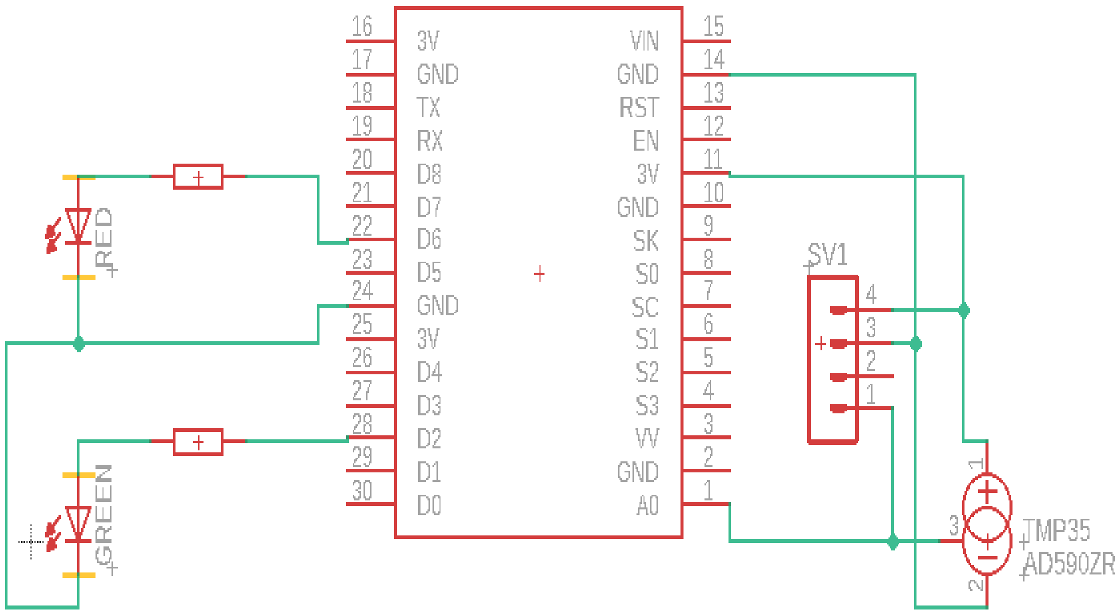


Figure 4.1: Schematic Diagram for the sensing device

By use of Arduino software on the micro-controller, the red LED was programmed to go on whenever the microthe controller was connected to a power source and when there is no Internet connection between the micro-controller and the mobile phone gateway. When it gets an Internet connection, the micro-controller’s inbuilt blue LED was programmed to come on, and then the red LED goes off. The green LED was programmed to come on continuously whenever the patient is taking measurements so that it helps the patients know that the device is measuring. It was also programmed to blinks three times after taking measurements to notify the user that the data was transmitted successfully to the database. The green LED goes off when the measurement is done. However, in case a wrong username or wrong password was provided before taking the measurement, after taking the readings, the red LED blinks for 3 seconds to notify the user to enter correct login detail before taking another measurement before transmission. Providing of correct login credential during taking measurements was made necessary because it provides an opportunity for any credible user to take his/her measurement with the same device, without necessarily purchasing his/her own device. This is to improve on the versatility and suitability of the device. The micro-controller was be powered by a 5V rechargeable power bank via a USB cable.

To improve on the accuracy of the temperature values being measured, the microcontroller was programmed to read ten temperature measurements, one, every second. After this, it computes the average of all the values and prepares for transmission. The formula for temperature conversions is found in the Arduino code section of the appendix section of this report. The temperature value read from the serial port (A0) conversion depends on the maximum voltage that can be read at that ADC pin (A0) of the micro-controller, which in this case is 3.3V. Also, to determine the voltage being read from the A0 pin, the value being read in bits is divided by 1024 and then multiplied by the total voltage that can be read at that pin.

After verifying that the above circuit in Figure 4.1 works, a casing for the device was developed using SolidWorks and 3D printed to make the sensing device look more presentable and protect the components direct contact with charged surfaces.

4.2 Data transmission

Since the esp8266 micro-controller has an inbuilt server service system, it required a client application on it to enable the patient to take the measurements when connected to the internet. In the Arduino code section in the appendix section, the *handleRoot* function shows the development of the client application written using HTML5 presentation technology. This client application was designed to provide a page where the user enters his/her username and password before taking the measurements.

After the mobile user hotspots the micro-controller, using his/her mobile phone, through providing the *ssid* name and *password* of the hotspot, the inbuilt blue LED of the micro-controller comes on to verify internet connectivity. At this point, the user is allowed to enter his/her login credentials and then clicks on the take temperature button to start taking the measurements using the sensing device's temperature sensor. After the user takes the vitals, an HTTP client on the microcontroller's server was used to access the remote database using the website's HTTP link address through the mobile phone(gateway). This HTTP client allows the sending of requests to the remote database to add newly measured temperature data using the user's login credentials. If the credentials are verified and correct, then the temperature is updated.

The microcontroller's server can only be accessed through an IP address that is obtained after connecting it to the mobile hotspot. It is this IP address that was used to access the client application on the server to enable the patient to take measurements and then transmit them to the remote database.

4.3 Web application

The design of the web application required several web technologies to get it running and hosted remotely. The files used to create the web application can be accessed through the information provided int the appendix of this report. The transmission of the sensor data from the sensing device to the remote database required hosting the web application on a remote server because a locally hosted web application can only be access by clients on the same server as the web application. But in this case, a remote global server was needed to host the application so that any sensing device connected to the Internet can always access the web application files remotely. In the development of the web application, presentation technologies like HTML5 and Bootstrap were used. To display graphs of the patient's previous measurements up to date, JavaScript was used in developing the graphs mostly. To store a patient's health vitals and enable logging into the web application, a remote database (Maria DB) was created using PHP and MySQL to store user information in the patient's table.

At the server side, two tables were created, one named *table_kh1* and *table_kh2* for patients and administrators in Korle Bu Hospital respectively. The web application is to be used by both the patients and hospital administrators but each with different dashboards upon login. The patients will be signed up by the hospital administrators after diagnosis. Every user's password was encrypted before it was stored in the database for security purposes. And every login session was to be destroyed whenever the user logs out from his/her dashboard for better security and privacy. Furthermore, privacy is very key in this application, so it is only the hospital administrator that can view all the patients' data after searching by name. Below is the flow chart for when the patient is interacting with the application.

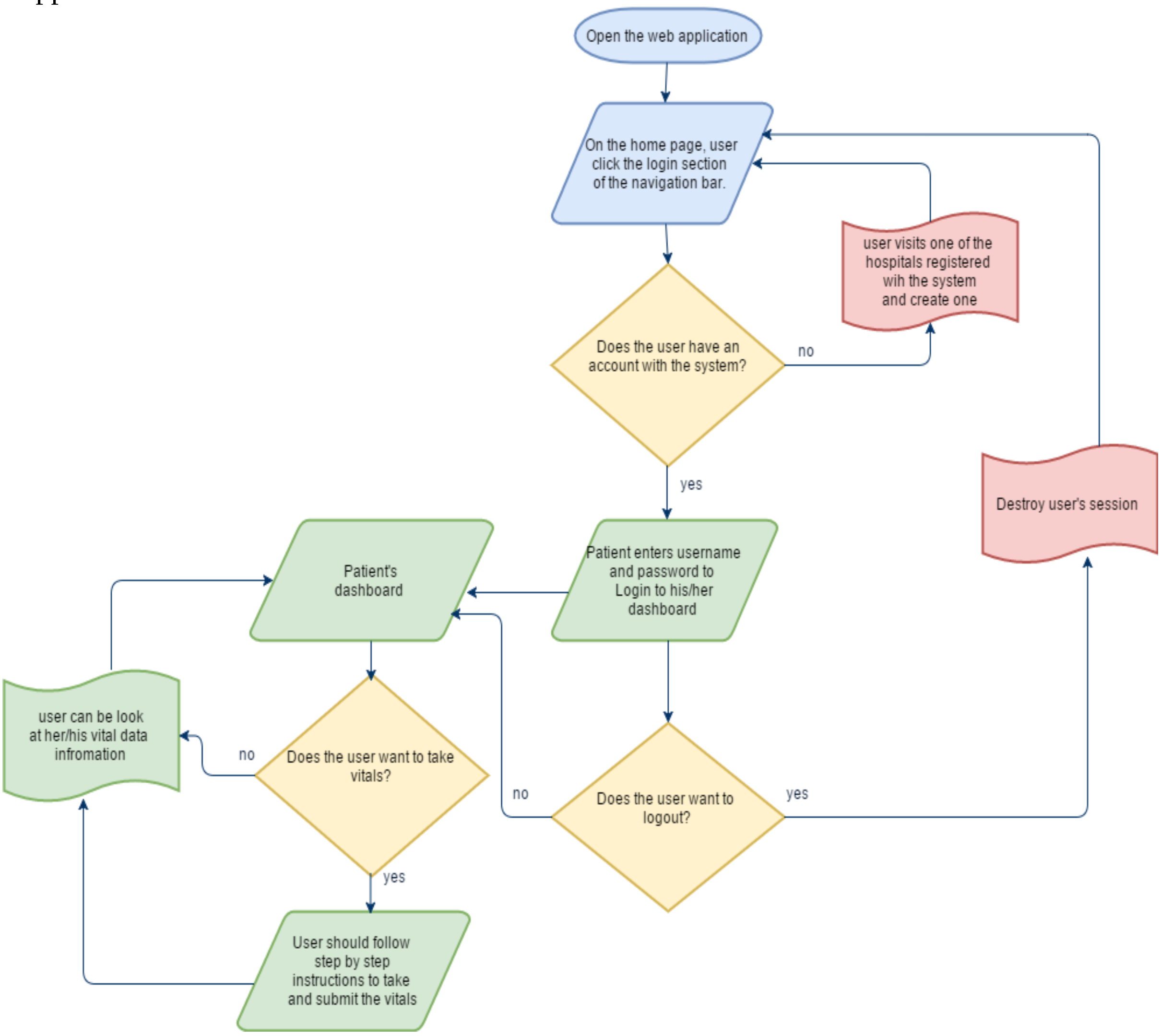


Figure 4.2: Web application flow chart for the patient

The hospital administrator uses the same web application except that he must use different login credentials. After logging in, the administrator's dashboard looks different from the patients' one with different features. The administrators' dashboards have their names, and they can find patients by names to get information about the patient's vitals history.

Chapter 5 : Results and Analysis

5.1 Web application

The web application was successfully hosted remotely on a free web application hosting services website, and accessible to both the hospital administrator and patients using this link: https://rehmsys1.000webhostapp.com/ from anywhere. The home page shown in Figure 5.1 below, has a how-to page that helps the user to know the steps necessary in navigating through the application and how to use the device.

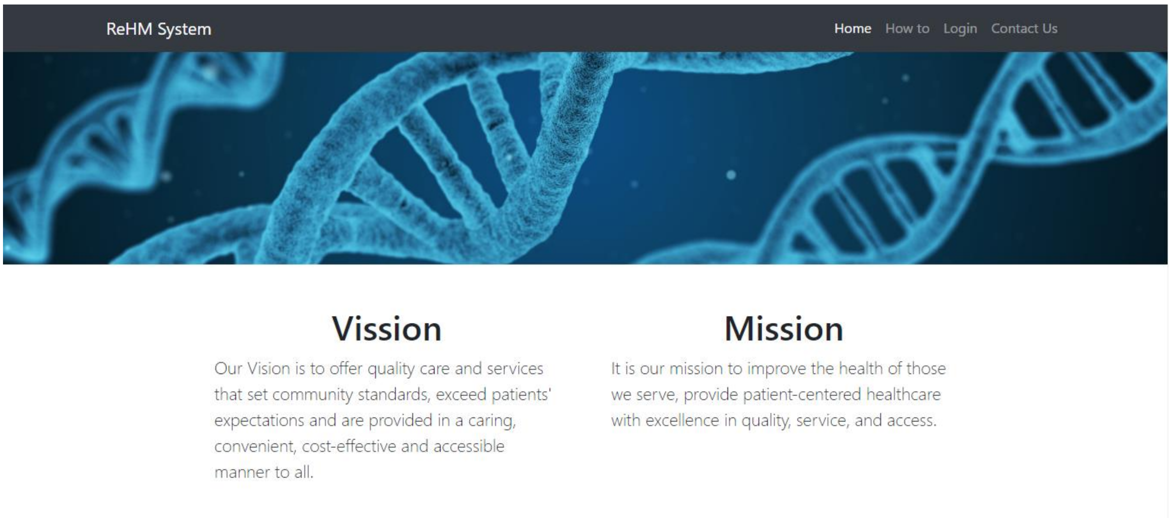


Figure 5.1: Home page of the web application.

It also has a contact us page for helping the user reach out to the developers. On the same page, there is a login page that the user enters his/her username, password and the category he/she falls in and then login. The passwords are encrypted to make the application more secure.

You might be wondering how users sign up. A hospital administrator can sign up another hospital administrator and most importantly signs up a new patient who wants to use the application. This is because before the patients start to use the system, it should be recommended to him/her by the hospital administrator.

After the hospital administrator logs in, he/she goes to the hospital administrator's dashboard, this dashboard has a report about the temperature and heartrate of actively monitored patients. On the same dashboard, the hospital administrator can see all the patients on the remote monitoring system and their most recent temperature and heart rate measurements as shown in Figure 5.2. Please note that the heart rate values were put in manually into the database due to the challenges faced in accessing and using the pulse sensor. The sensor was very noisy and very unreliable than expected from its data sheet.

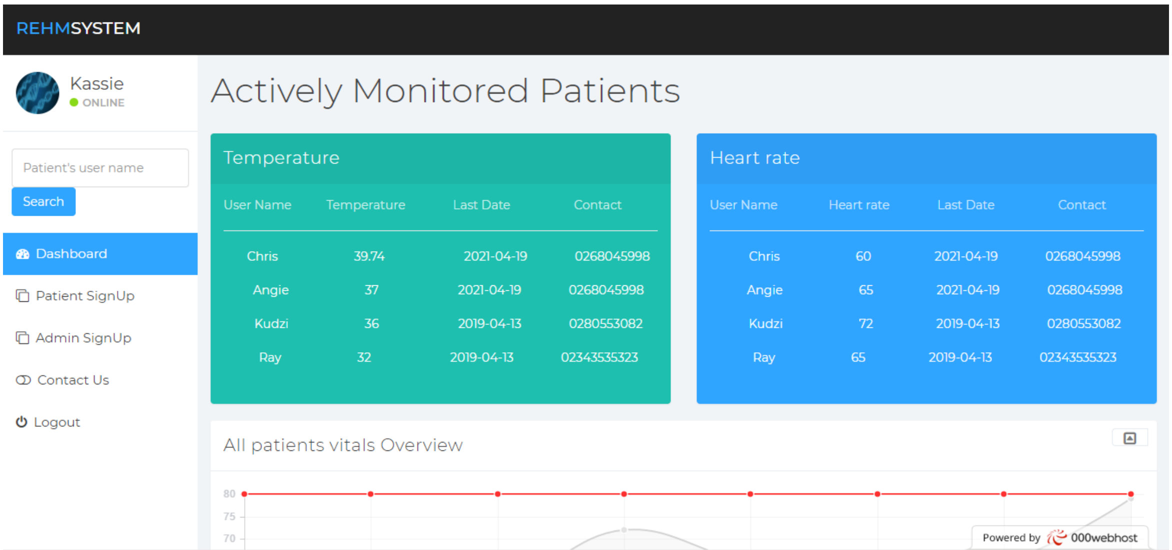


Figure 5.2: Administrator's (Kassie’s) dashboard.

The other feature provided by the administrator’s dashboard is that, the administrator can search for any patient on the system using his/her user name and a health vitals report about the patient monitoring history is generated and displayed with all the necessary information about the patient since he/she started taking measurements as shown in Figure 5.3 below.

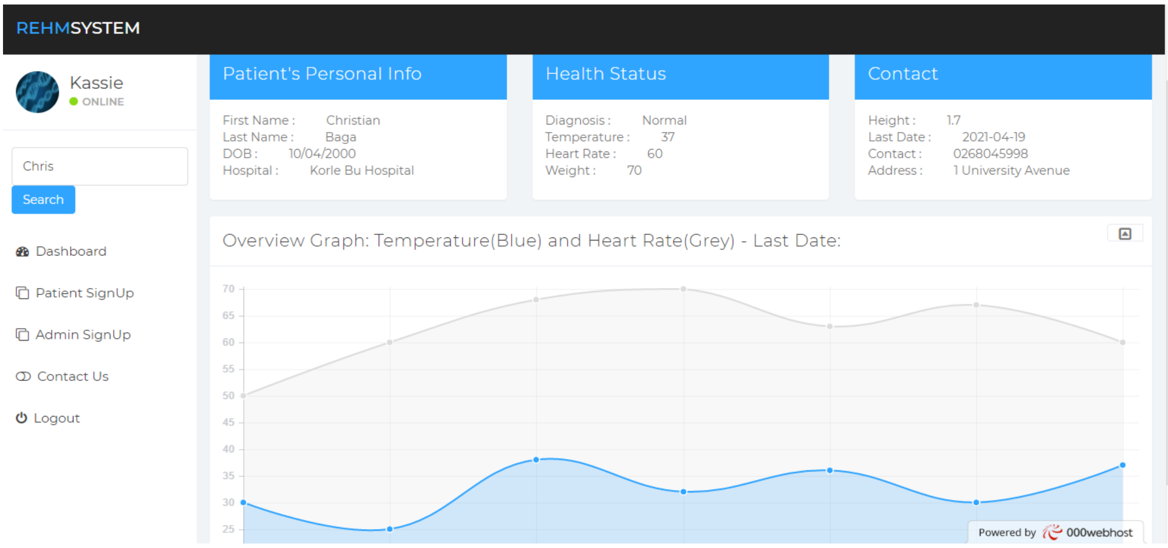


Figure 5.3: Administrator’s search results after searching for Chris.

Considering an already registered patient like Angie, if she logs into her dashboard, she will be able to see her most recent temperature, and heart rate measurement, in addition to her weight and blood pressure if she was measuring them all. A graph showing all her past measurements, and page where she can take new temperature measurements as shown in Figure 5.4 below.

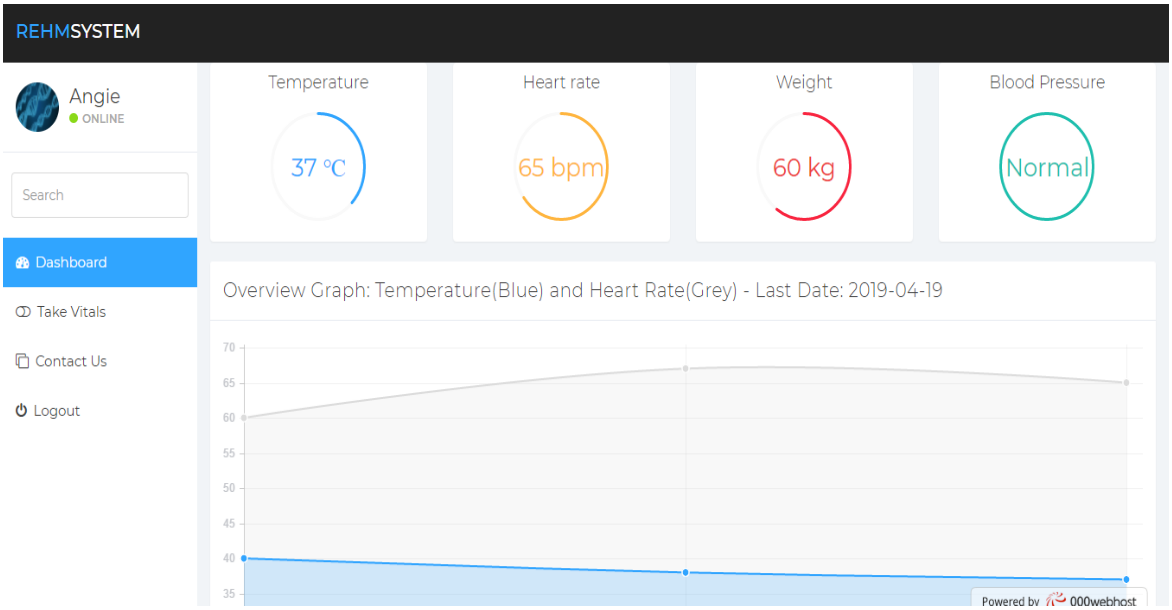


Figure 5.4:Patient's (Angie’s) Dash board.

If Angie clicks on take vital page as shown in Figure 5.5, she can click on take new temperature button that will take her to the vitals taking page on the sensing device.

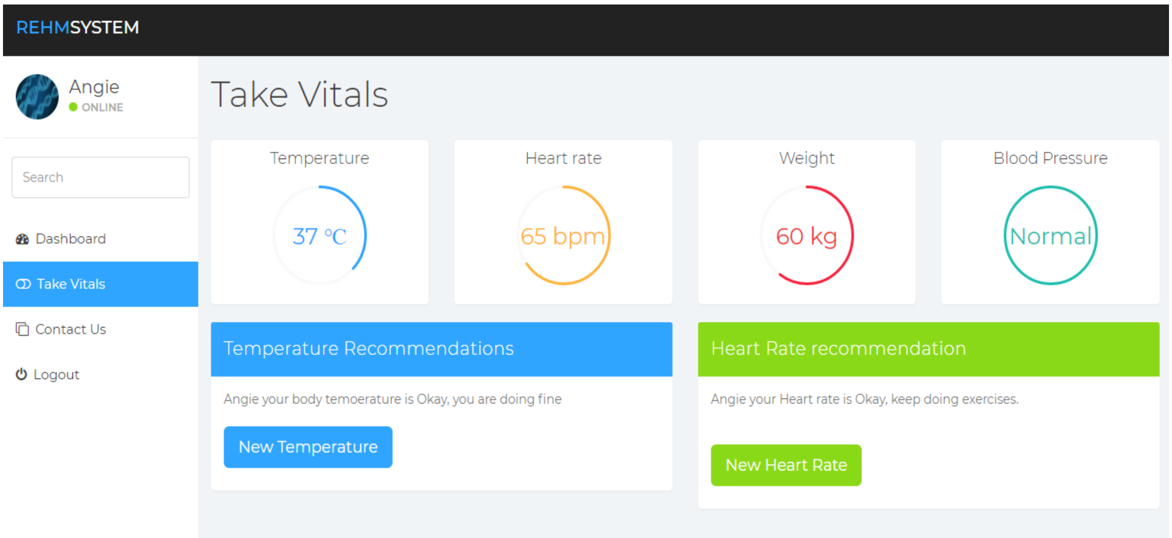


Figure 5.5: Patient's vitals taking page.

Both the administrators and the patients were able to log in and log out from the sessions and can always access the previously recorded temperature data anytime anywhere.

5.2 Sensing device

From the sensing device implementation above, the temperature sensor was able to provide accurate temperature reading whenever there was a change in the temperature of the environment. This was tested by providing warm rubbed hands around it, and different temperature values were recorded, and then an average of them was used as the temperature

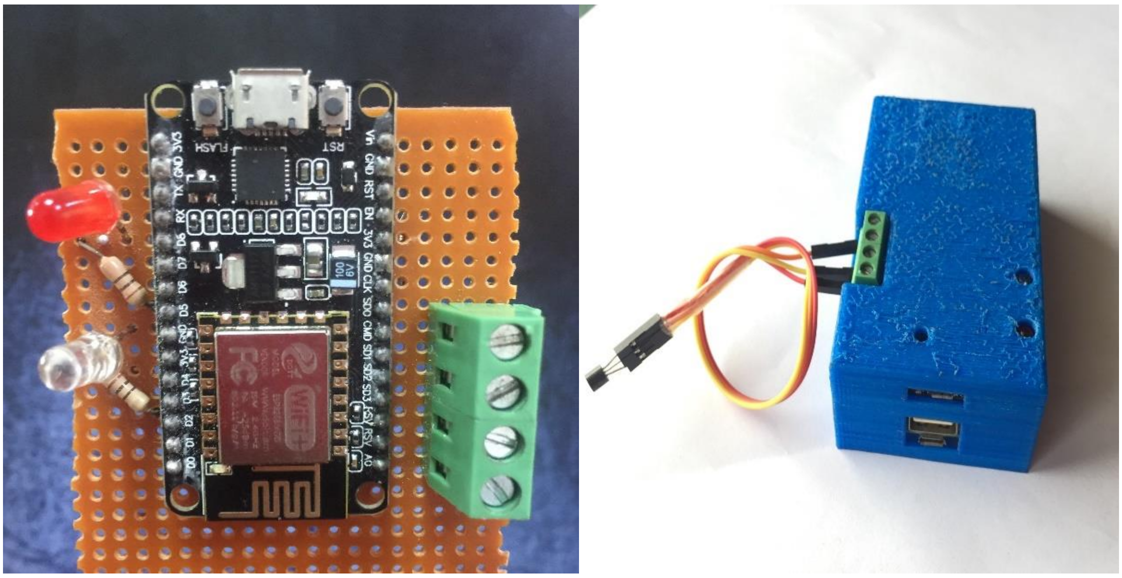


Figure 5.6: Sensing device designed and built Circuit.

being read at that point. A working circuit of the device was successfully built as shown in Figure 5.6.

Consider a patient like Chris who has logged into the system and wants to take his new temperature measurement. Below in Figure 5.7 is his dashboard before taking the measurement.

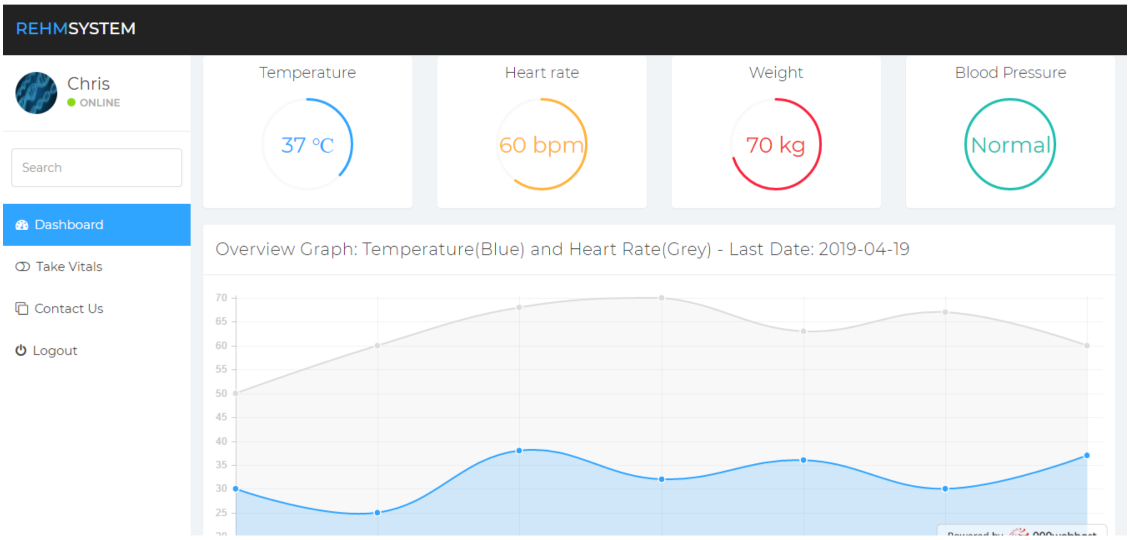
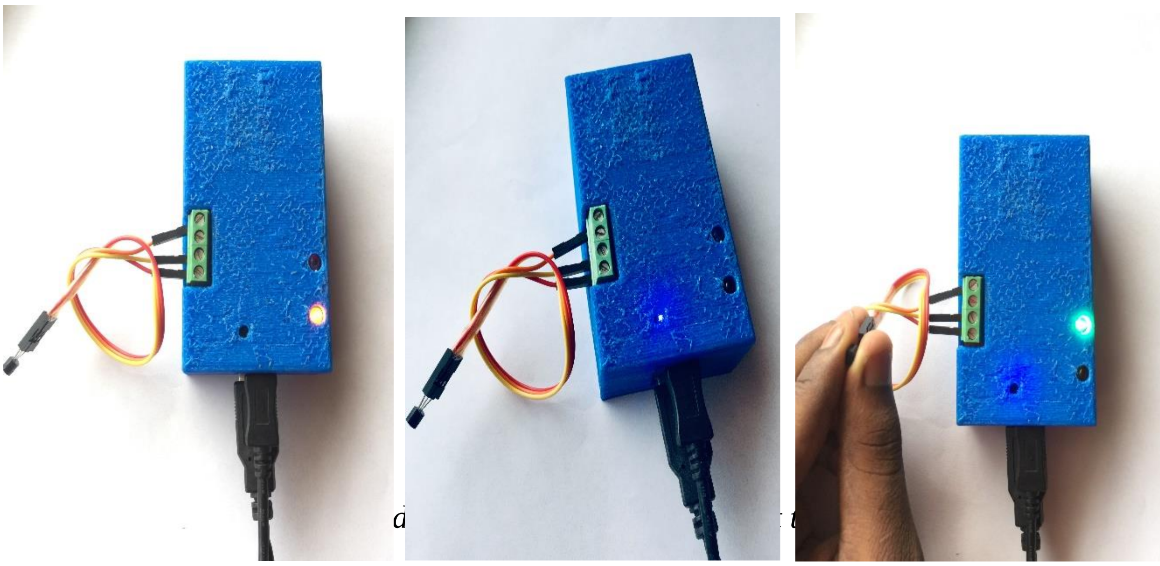


Figure 5.7: Chris's dashboard before new temperature measurement.

Chris switches on the device using the power source attached to the device as shown in Figure 5.6 A. The red light shows that the device is powered but not yet connected to the



Internet. When it connects to the Internet, the red light goes off, and the blue one comes on to confirm connectivity as shown in Figure 5.8 B.

The device takes on average 5 seconds to connect to the internet because it has got to search for the hotspot and then after requests for connectivity using the ssid name and password as shown in Figure 5.9.

```
Connecting to Joshua-HS
.....
WiFi connected
IP address:
192.168.137.110
```

Figure 5.9: Connecting to the hotspot Joshua-HS and after connecting to the hotspot.

After connecting the device to the hotspot, Chris clicked on the take new temperature button in the take vitals page. Using the IP address obtained from the connection in Figure 5.9 which is linked to the take new temperature button, Chris was directed to the vitals taking page shown in Figure 5.10 below. On this page, Chris entered his login credentials and clicked on the take temperature button.

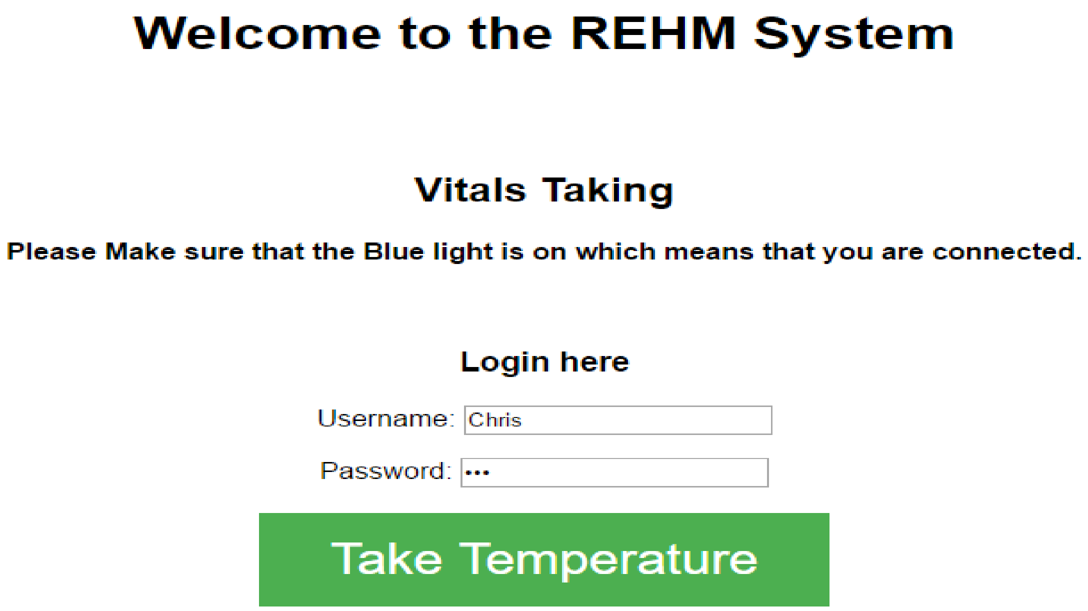


Figure 5.10: Vitals taking page found on the sensing device.

After clicking the button, the green LED on the device came on continuously until the measurement was done as shown in Figure 5.8 C. When the green light went off, to

```
User Name: Chris
Password: 123
35.24
36.76
36.98
37.20
36.33
37.85
37.85
38.06
38.06
38.06
Average: 37.24
http://rehmsys1.000webhostapp.com/add_temp.php?temp=37.24&username=Chris&password=123
[HTTP] GET...code: 200
Temperature successfuly saved!
```

Figure 5.11: Taking measurements and successful transmitting of the new temperature value to the remote database.

confirm that he entered the right credentials, the green light blinked three times to also confirm successful transmission of the new temperature value. Figure 5.11 below shows what was happening in the sensing device from the point Chris clicked the take temperature button.

For Chris to confirm that a new temperature was added, Chris first logged out and logged into his dashboard again, and the results in Figure 5.12 below were obtained.

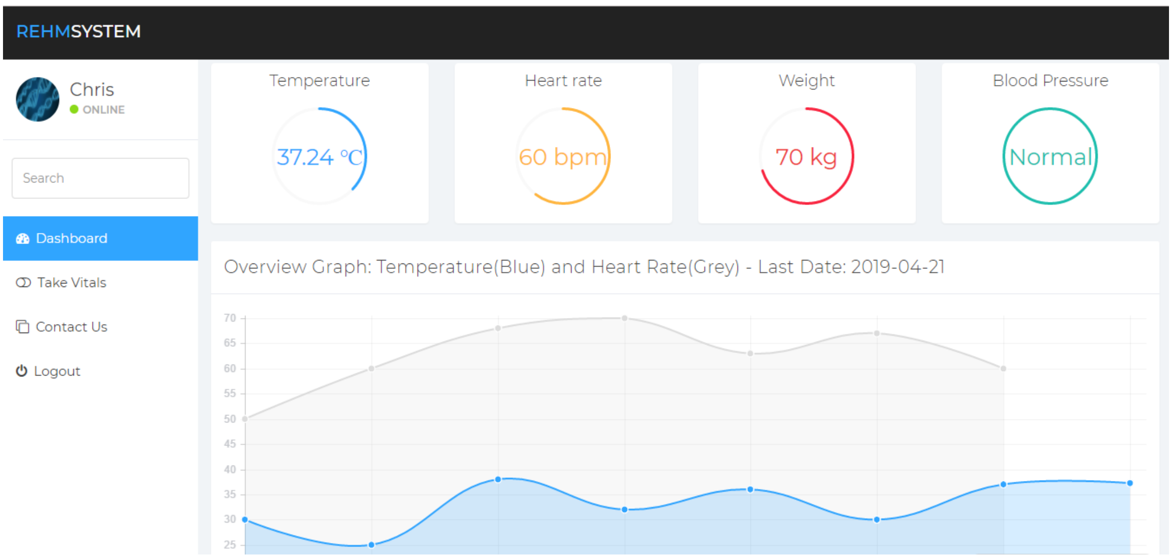


Figure 5.12: Dashboard after Chris took a new temperature measurement.

From Figure 5.12 above, the new temperature was displayed in the temperature section and also added it to the overview graph. The date at which he took the new temperature was also updated in the last date section.

5.3 Vital data transmission

After the temperature was measured by the patient, it was successfully transmitted to the database of the web application. This was possible because the patient able to connect the sensing device to the Internet, then entered the correct credentials and the device successfully used the HTTP protocol to communicate to the web application through the file, add_temp.php suing the gateway. This file required the temperature and login credentials from the device which the user had entered before taking the measurement. The http link used to allow this data transmission is found on line 2 in Figure 5.13 below, a more commented section of the code that activated the protocol is in the Arduino code in the appendix.

```
Average: 37.24
http://rehmsys1.000webhostapp.com/add_temp.php?temp=37.24&username=Chris&password=123
[HTTP] GET...code: 200
Temperature successfuly saved!
```

Figure 5.13: After a successful data transmission.

Chapter 6 : Conclusion

6.1 Discussion

The main objective of this project was to come up with a low cost, reliable, accurate and portable remote health monitoring system. A system that is accessible to both patients and hospital administrators remotely for easy communication of health vital data between the patients and their hospitals efficiently with minimal cost. The objective was successfully achieved, and a working prototype of the system is up and running. The complete design of this prototype costs 30 USD which is way cheaper than when the transport and time costs incurred by if they were to visit a hospital weekly. The accuracy section of the data transmitted was tested because of a lack of open source approved medical sensors. But it was confirmed that whatever was transmitted was received as it was sent. The system is very reliable as long as there is fast internet connectivity. On average it takes 2 minutes for a patient to take his/her vitals on a fast and stable internet connection. Lastly, the different codes and login details used for the project are attached in the appendix of this report in case anyone would want to try it out.

6.2 Limitations

This system can for now take temperature measurements only. Its efficiency in terms of time used in taking measurements is dependent on how fast the Internet connection is. This device's temperature readings cannot be used to diagnose any patient because the sensor accuracy was not tested in relation to the already existing ones.

6.3 Future works

This system is going to be worked upon until it is released as a product, but some things have to be modified or added. These include obtaining open source health sensors to test and improve on its accuracy and linking the information in its database with the health management information system. Lastly, the design can be improved on by adding a touch screen for the user to enter hotspot connection credential.

References

[1] J. Arthur, F. K. Awittor, F. Asiedu-Bekoe and D. O. Laryea, "Burden of Adolescent Morbidity and Mortality in Kumasi in 2013-2015," *Ghana Weekly Epidemiological Report,* vol. 2, no. 37, pp. 9-15, 2017.

[2] K. Byung-Hyun and Y. Jong-Gwan, "Human health monitoring technology," 18 May 2017. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10194/101940T/Human-health-monitoring-technology/10.1117/12.2262399.short?SSO=1. [Accessed December 2018].

[3] S. Asiedu-Addo, "Akufo-Addo alarmed at inadequate doctor-patient ratio," *GRAPHIC online,* 13 Aug 2017.

[4] M. Marzencki, P. Lin, T. Cho, J. Guo, B. Nga and B. Kaminska, "Remote Health, Activity, and Asset Monitoring," *IEEE 13th International Conference on e-Health Networking, Applications and Services,* pp. 98-101, 2011.

[5] M. Ogawa and T. Togawa, "The Concept of the Home health monitoring," *Department of Biomedical Instrumentation, Institute of Biomaterials and Bioengineering, Tokyo Medical and Dental University,* vol. 73, p. 71, 2010.

[6] R. Gonzalez, "The new ECG apple watch could do more harm than good," *WIRED,* 13 Sept 18.

[7] A. Bhatti, A. A. Siyal and A. Mehdi, "Development of Cost-Effective Tele-Monitoring System for Remote Area Patients," in *Internation Conference on Engineering and Emerging Technologies (ICEET)*, 2018.

[8] Death rate, crude (per 1,000 people). (n.d.). Retrieved from:

https://data.worldbank.org/indicator/SP.DYN.CDRT.IN?end=2016&locations=GH&start=2013

[9] S. K. Islam, A. Fathy, Y. Wang, M. Kuhn and M. Mahfouz, "Hassle-Free Vitals: BioWireleSS for a Patient-Centric Health-Care Paradigm," in IEEE Microwave Magazine, vol. 15, no. 7, pp. S25-S33, Nov.-Dec. 2014. doi: 10.1109/MMM.2014.2356148 Retrieved from:

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6954654&isnumber=6954637

APPENDIX

Web application Documentation

To access all the web application files used in this project, visit this link: https://www.000webhost.com/cpanel-login?from=panel and log in using the credential below:

Email: kasiryejoshua93@gmail.com

Password: KassieJJJ1

In the website manager section of the dashboard, chose the file manager, upload the page button to log into the web application files using the credentials below.

Website name: rehmsys1

Password: rehmsys1

After a successful login, all the files used to develop the web application will be found there.

To test the application as an Administrator, use this link: https://rehmsys1.000webhostapp.com/ to log in as an administrator using the credentials below

Username: Kassie

Password: 123

As a patient, use these credentials below

Username: Chris

Password: 123

Username: Angie

Password: Tabi

Arduino code

```
/*
   By: Joshua Kasirye
*/

/* Disclaimer
   Please the code below works for only esp8266 which I used, not esp32. It was difficult to ship in the
   esp32 before the end of this project, therefore the next best option was esp8266 which
   worked similarily like the esp32. Thank you*/


  //Including all the required libraries.
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266WebServer.h>
#include <ESP8266HTTPClient.h> //http library to enable http connection
#include <WiFiClient.h>


// WiFi network
const char* ssid     = "Joshua-HS"; //Hotspot name
const char* password = "KassieJJJ"; //Hotspot password


//initializing and declaring variables
volatile float val; //variable to keep the analog input from the sensor
int measuringLED = D2; //GREEN LED
int powerLED = D5; // RED LED
const int analog_inputPin = A0; //Analog pin on esp8266

// Set web servers at port number to 80
ESP8266WebServer server ( 80 );//This server name is for enabling using the html client app
WiFiServer server1(80); //This server name is for enabling using the http protocol


void setup() {
  //defining the pins, D2, D5 and A0
  pinMode(measuringLED, OUTPUT); //Green LED for measuriing
  pinMode(powerLED, OUTPUT); //Red LED for power and no connection
  pinMode(analog_inputPin, INPUT); //Pin for reading from the sensors
  digitalWrite(measuringLED, LOW); //Turning the Green LED off
  digitalWrite(powerLED, HIGH); //Turning the red LED on
  Serial.begin(115200); //serial monitor with baund frequency = 115200

  // Connecting to a WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password); // Requesting for internect connectivity

  while (WiFi.status() != WL_CONNECTED) { //Checking connecctivity
    delay(500); //waits for half a second before next request
```

```
  Serial.print(".");
  pinMode(LED_BUILTIN, LOW); //Blue LED goes off if not connected
 }

 Serial.println("");
 Serial.println("WiFi connected");
 Serial.println("IP address: ");
 Serial.println(WiFi.localIP());

 server.on ( "/", handleRoot ); //Getting user info from the html client input form
 server.on ("/save", handleSave); //Client appllication on ESP server and saving on server

 //Beginging both servers all at once for easy communication
 server.begin();
 server1.begin();

 pinMode(LED_BUILTIN, HIGH);//Blue LED comes on
 digitalWrite(powerLED, LOW); //Red LED goes off
}


void loop() {
 server.handleClient();// calling the main function that handles the client app.
}


char htmlResponse[3000]; //For storing the HTML response request

//Creating the HTML Vital taking page
void handleRoot() {
 snprintf ( htmlResponse, 3000,
      "<!DOCTYPE html>\
<html lang=\"en\">\
 <head>\
  <meta charset=\"utf-8\">\
  <meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">\
  <style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align:
center;}\
  .button { background-color: #4CAF50; border: none; color: white; padding: 16px 40px;\
  text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}\
  .button2 {background-color: #555555;}\
  </style>\
 </head>\
 <body>\
     <h1>Welcome to the REHM System</h1>\
     <br></br>\
     <h2> Vitals Taking</h2>\
     <h4>Please Make sure that the Blue light is on which means that you are connected.</h2>\
     <br>\
     <h3>Login here</h3>\
     <p>Username: <input type='text' name='user_name' id='user_name'required></p>\
     <p>Password: <input type='password' name='password' id='password' required></p>\
     <div>\
     <p><button class=\"button\" id=\"save_button\">Take Temperature</button></p>\
     </div>\
```

```
    <script src=\"https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js\"></script>\
    <script>\
      var UserName;\
      var Password;\
      $('#save_button').click(function(e){\
        e.preventDefault();\
        UserName = $('#user_name').val();\
        Password = $('#password').val();\
        $.get('/save?UserName=' + UserName + '&Password=' + Password , function(data){\
          console.log(data);\
        });\
      });\
    </script>\
  </body>\
</html>");

  server.send ( 200, "text/html", htmlResponse );// save input form data to server

}


void handleSave() {//function to retrieve the saved information from the server
  if (server.arg("UserName") != "") {
    Serial.println("User Name: " + server.arg("UserName")); //User name
  }
  if (server.arg("Password") != "") {
    Serial.println("Password: " + server.arg("Password")); //Password
  }

  digitalWrite(measuringLED, HIGH);//Green LED comes on.
  volatile float average = 0;
  volatile int count = 0; //variable to keep truck of added values
  for (int i = 0; i <= 10 - 1; i++) {
    val = analogRead(analog_inputPin); //Reading sensor data

    float cel = ((val * (5.0 / 1024)) - 1.375) / 0.0225; //obtaining temperature in Celcius
    if (cel > 25 and cel < 55 ) { //verrifying data being read by sensor
      average = average + cel;
      count = count + 1;
    }

    Serial.println(cel);
    delay(1000); //waiting after one second
  }
  average = average / count; // obtaining the average
  digitalWrite(measuringLED, LOW);

  Serial.println("Average: " + String(average));

  //initializing the http client as http
  HTTPClient http;
  //link to be used to send the measured average temperature to the remote data base.
  String url = "http://rehmsys1.000webhostapp.com/add_temp.php?temp=" + String(average) +
"&username=" + server.arg("UserName") + "&password=" + server.arg("Password");
```

```
  String payload = ""; //variable to store the response from the http request.
  Serial.println(url);
  http.begin(url); //sending the http request.


  //GET method
  int httpCode = http.GET(); //getting the response code
  if (httpCode > 0)
  {
    Serial.printf("[HTTP] GET...code: %d\n", httpCode);
    if (httpCode == HTTP_CODE_OK)
    {
      payload = http.getString(); // obtaing the string of the response from the database
      payload.trim(); // removing unwanted spaces from the string
    }
  }
  else
  {
    Serial.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(httpCode).c_str());
  }
  http.end();//closing the client

  delay(500);
  int len_payload = payload.length();//length of payload
  Serial.println(payload);
  Serial.println(len_payload);
  if (len_payload == 3 ) { //Transmission was successful
    for (int i = 0; i < 3; i++) {
      digitalWrite(measuringLED, HIGH);
      delay(1000);
      digitalWrite(measuringLED, LOW);
      delay(1000);
    }

  }
  else { //Transmission not successful
    for (int i = 0; i < 3; i++) {
      digitalWrite(powerLED, HIGH);
      delay(1000);
      digitalWrite(powerLED, LOW);
      delay(1000);
    }

  }
}
```

Add Temperature php code.

```
// BY Joshua Kasirye
<?php

    // define variables and set to empty values
    $servername = "remotemysql.com";
    $username = "y2O7u7sfAX";
    $password1 = "4DD9ReYRUX";
    $dbname = "y2O7u7sfAX";



    // Create connection
    $conn = mysqli_connect($servername, $username, $password1,$dbname);
    $myTable = "table_kh1";

    if (!$conn) {
        die("Connection failed: " . mysqli_connect_error());

    }
    else{
        $val = $_GET['temp'];
        $username1 = $_GET['username'];
        $password = $_GET['password'];

        $sql = "SELECT* FROM $myTable WHERE username = '$username1'";
        $result = $conn->query($sql);
        $row = $result->fetch_assoc();
        $database_password =" ";
        $database_password = $row["password"];

        if(password_verify($password,$database_password)){

            $date = date('y-m-d');
            $sql = "UPDATE `$myTable` SET `Temperature` = CONCAT(`Temperature`,',', $val),
`Last Date` = '$date'  WHERE `username` = '$username1'";
            if ($conn->query($sql) === TRUE) {
```

```
            $conn->close();
            echo "YES"; //Temperature successfuly saved
        }
        else{
            echo "did not add temp";
            echo "Error:" . $sql . "<br>" . $conn->error;
        }

    }else{

        echo "Wrong password";
        echo "Error:" . $sql . "<br>" . $conn->error;
    }
  }

?>
```