Home Monitoring System

Nishchayjeet Singh

T00637213

A report on research conducted under supervision of Dr Anthony Aighobahi. The project is completed in team

OM COMPUTER Science Lab
Thompson Rivers University, Kamloops

November 30, 2022

ABSTRACT

This report details the development of an IoT (Internet of Things) project using a Raspberry Pi and SenseHat to capture environment temperature, humidity, and pressure data. This data was then stored in AWS and pushed to Google Firebase. Furthermore, an IOS application was developed to register and log in new users, and to monitor the data captured from the SenseHat. The application was successful in registering and logging in users and monitoring the data gathered from the SenseHat. The data collected is valuable for monitoring the physical environment in various locations and situations such as our homes, and this report offers an insight into the development of the project. The results indicate that the system is capable of accurately monitoring data and displaying it in real-time, thus providing an efficient means of remotely monitoring the environment.

ACKNOWLEDGMENTS

I would like to express our sincere gratitude to our professor Dr. Anthony Aighobahi for his kind guidance, valuable advice, and support throughout the project. His expertise in the field of raspberry pi and SenseHat, AWS, and Google Firebase was of great help in the successful completion of our project. We are grateful to him for his patience and willingness to provide us with the necessary assistance in the project. We are thankful to him for his valuable comments and suggestions throughout the project. We are highly grateful to him for his continuous motivation and support throughout the project. His presence was of great help in completing the project.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

SUMMARY

0.1 THE PROBLEM

The desire to keep an indoor environment pleasant, and secure, and use as little energy as possible inspired the development of home monitoring systems as a solution to a problem that needed to be solved. Both the temperature and the humidity in a home may have a significant influence on the degree of comfort it provides as well as the amount of energy it uses. If there is not enough humidity in the air, it may lead to health problems such as dry skin and respiratory discomfort. On the other hand, too much humidity can contribute to the formation of mold and other problems. The improper temperature can also lead to living circumstances that are uncomfortable as well as an increase in the cost of the energy used. Monitoring systems for the house are helpful in ensuring that the levels of temperature and humidity in a home are maintained at levels that are satisfactory.

0.2 METHOD OF INVESTIGATION

There were quite a ways to investigate an IoT project that monitors environmental conditions using a Raspberry Pi and SenseHat.The following list includes how I followed the approach:

1. Determine the scope of the project.The scope of the project should be determined by considering the key objectives and the available resources. This includes the specific environmental data that needs to be collected, the types of devices and sensors used to collect the data, the platform used to store and analyze the data, and the software used to display the data in real time.

2. Research the devices and sensors used to collect the data.The devices and sensors used to collect the data should be thoroughly researched, including their features and capabilities. This includes the Raspberry Pi and SenseHat, as well as any other devices used to collect the data.

3. Research the platform used to store and analyze the data.The platform used to store and analyze the data should be researched, including its features, capabilities, and security measures. This includes the AWS and Google Firebase platforms used to store and analyze the data.

4. Research the software used to display the data in real-time.The software used to display the data in real-time should be thoroughly researched, including its features, capabilities, and security measures. This includes the iOS app used to display the data.

5. Test the system

1 INTRODUCTION

This document describes the process of creating an IoT project that monitors environmental conditions including temperature, humidity, and pressure using a Raspberry Pi and SenseHat. The SenseHat data can be seen and managed through an iOS app that allows users to sign up or log in, and view their sensor-collected data. The information was subsequently transferred from AWS to Google Firebase for storage. The overarching objective of the project was to collect environmental data from multiple sources, store it, and then present it in real-time via an iOS app. The results of the project were successful, both in registering and logging in users and in monitoring and gathering data from the SenseHat. We can use the information gathered to keep tabs on the state of the world in places as basic as our own houses. This report sheds light on the project's progression and final outcomes. It is hoped that in the future engineers and developers will be able to use this data to better understand how to create an IoT system for tracking environmental metrics. In total, it took 7 weeks to do the project. In this report, we'll go over the project's development, including the challenges we faced and the strategies we implemented to overcome them. The project's findings and conclusions will also be discussed in this report.

2 METHODS, ASSUMPTIONS, AND PROCEDURES

2.1 METHODS

2.1.1 Problems Discovered

I started by conducting research on the internet to find out what kinds of items already exist on the market that are comparable to the one I intended to develop. The next thing I did was investigate the components that are required to build a system of this kind. These components include sensors for temperature, humidity, and pressure, in addition to a CPU, cloud storage, and a mobile application. In contrast to that, I investigated the most effective methods for linking these components and investigated the many possibilities for cloud storage, such as Amazon Web Services and Google Firebase. In the end, I investigated various development frameworks and programming languages with the purpose of constructing a mobile application.Manjula et al. (2016)

Some Other challenges included were:

1. Power Supply: One of the major challenges in developing a home monitoring system using Raspberry Pi and SenseHAT is to provide a reliable and constant power supply. This is because Raspberry Pi and SenseHAT require a constant power supply to ensure their proper functioning.

2. Network Connectivity: Another challenge is to ensure that the system is connected to the internet in order to send and receive data across the cloud platform. This requires the integration of a reliable and stable network connection.

3. Programming: Programming the Raspberry Pi and SenseHAT to interact with the AWS and Google Firebase cloud platforms is also a major challenge. This requires knowledge of the programming language and the ability to integrate the hardware components of the system with the cloud platform.

4. Data Security: Ensuring the security of the data stored on the cloud platform is also a major challenge. This requires the use of secure authentication and encryption methods in order to protect the data from unauthorized access.

2.1.2 Tools And Measurement Systems

1. Raspberry Pi: This is the main device used to build the home monitoring system.
2. Sense HAT: This is the hardware used to sense various environmental parameters like temperature, humidity, and pressure.
3. Programming Language: A programming language such as Python is required to write the code to control the Sense HAT and Raspberry Pi.
4. Serial Port Monitor: This is used to monitor the serial communication between the Raspberry Pi and the Sense HAT.
5. Cloud Services: Cloud services such as AWS are required to host a web service to access the sensor data from the cloud.Guth et al. (2016)
6. Database: A database such as Google Firebase is used to store sensor data.
7. Charting Dashboard: A charting platform such as Adafruit is used to visualize the sensor data.
8. Mobile APP Development: Required programming and integration skills to develop IOS mobile user-friendly app.

2.2 ASSUMPTIONS

When I first began working on the project, there were a few things that needed to be assumed before the project could even begin. These items included the project's scope and its overall purpose. The timetable, as well as the money, in addition to the needed resources, for the completion of the project. A list of the necessary tools and technologies, as well as management of risk and the consequences of failing to produce. The developing project will be subjected to testing and validation, as well as post-project evaluations.

There some substantiated facts that were involved in the course of this project include:

1. That the Raspberry Pi is able to connect to the SenseHAT.
2. That the Raspberry Pi is able to connect to the AWS and Firebase databases.
3. That the data collected from the SenseHAT is compatible with the AWS and Firebase databases.
4. That the Raspberry Pi has enough processing power to store and manage the data.
5. That the Raspberry Pi is able to communicate with the mobile app.
6. That the mobile app is able to communicate with the Raspberry Pi.

7. That the mobile app is able to connect to and access the AWS and Firebase databases.

8. That the mobile app is able to interpret and display the data from the AWS and Firebase databases.

9. That the mobile app is able to send commands to the Raspberry Pi.

10. That the Raspberry Pi is able to interpret and respond to the commands sent by the mobile app.

2.3 PROCEDURES

To gain a better understanding of the problems involved in developing a home monitoring system using a Raspberry Pi and SenseHAT, it was important to become familiar with the hardware and software involved. It was also important to understand the data that can be collected and the applications that can be created with the data. Additionally, understanding the security requirements to ensure the system is secure from external threats is essential. Lastly, understanding the infrastructure of the system and the integration with other systems such as an Internet of Things (IoT) platform is important. By understanding these elements, it was possible to gain a better understanding of the problems that may be encountered when developing a home monitoring system using Raspberry Pi and SenseHAT.

3 RESULTS AND DISCUSSION

My experience with this project taught me how to configure a Raspberry Pi with a Sense HAT, write code to instruct the device to take readings from its surrounding environment, and save those readings in Amazon Web Services and Google Firebase respectively. I was also shown how to make an iOS app for mobile devices in order to monitor the data and how to make an Adafruit dashboard for the web in order to present the data in a graphical way. In addition, I was able to figure out how to connect the Raspberry Pi to the Internet in order for it to be able to transmit the information that it gathers to cloud storage services.Childs-Maidment (2017) In the end, I was able to acquire the knowledge necessary to solve any issues that came in the course of the project.

3.1 TABLE

Table 3.1. These Tables show the data generated from Sensehat which is sent to AWS and then to Firebase

Data Created	Humidity	Local Date	Local Time	Pressure	Temperature
Data1	23.2	November 29, 2022	19:17:14	961.5	33.9
Data2	23.1	November 29, 2022	19:27:54	961.4	33.7
Data3	23.3	November 29, 2022	19:28:46	961.2	33.5
Data 4	23.7	November 29, 2022	19:29:38	961.3	32.9

Firebase stores data in a JSON tree structure, which is essentially a hierarchical collection of keys and values. JSON Tree Structure is a hierarchical data structure that stores data in a parent-child relationship. It is typically used to represent a set of hierarchical data in an organized and easy-to-navigate manner. It is commonly used to display data in web applications and web services, such as a hierarchical directory structure or a family tree. Each element in the tree is represented as a node and can have any number of children.

3.2 FIGURES

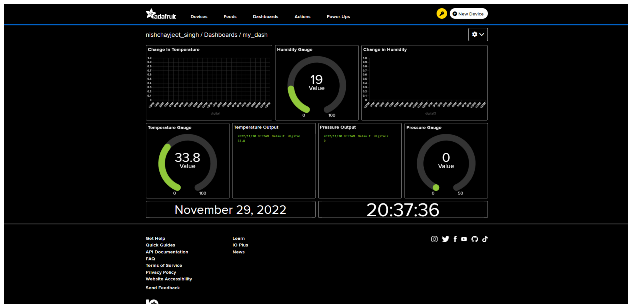


Figure 3.1. Adafruit Dashboard that displays Different Gauges for Temperature, Pressure, and Humidity and Graphs of each parameter With Current Date and Time

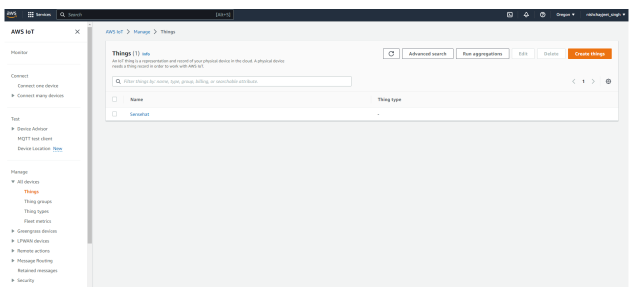


Figure 3.2. The Thing created in AWS that was required to push data AWS to Google Firebase

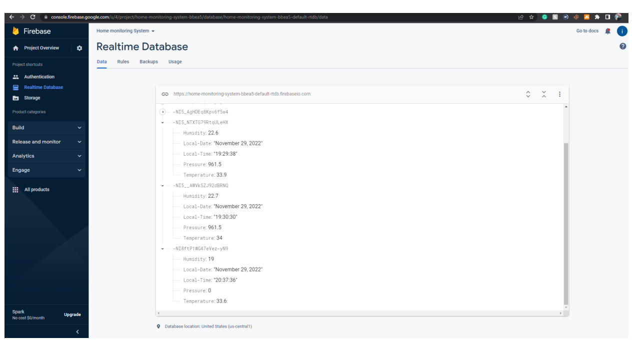
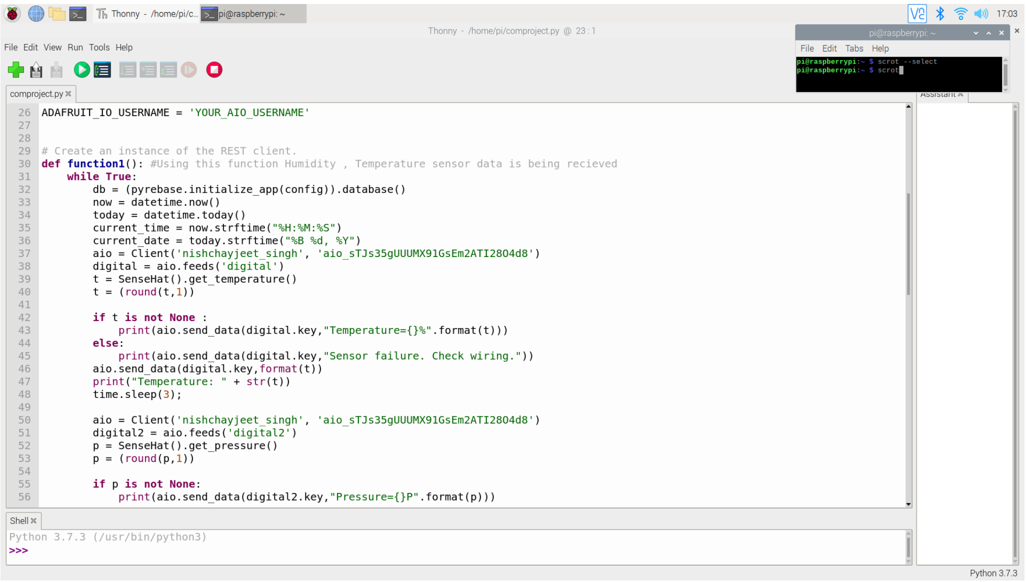


Figure 3.3. The Picture format of how data is being transferred from AWS To Google Firebase

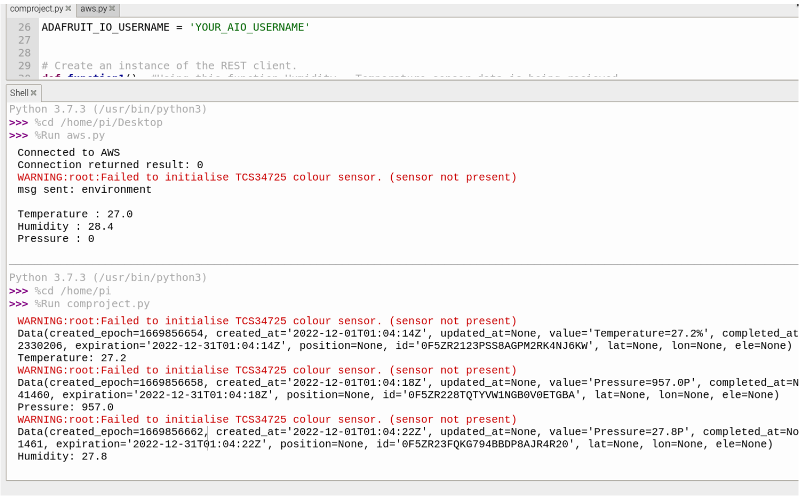
3.3 SOURCE CODE INTEGRATION

3.3.1 comproject.py

Comproject.py python file created in raspberry pi imports several libraries and the code pushes the data from raspberry pi to AWS. The code also imports Adafruit libraries which are responsible for working of the web-based Adafruit dashboard.

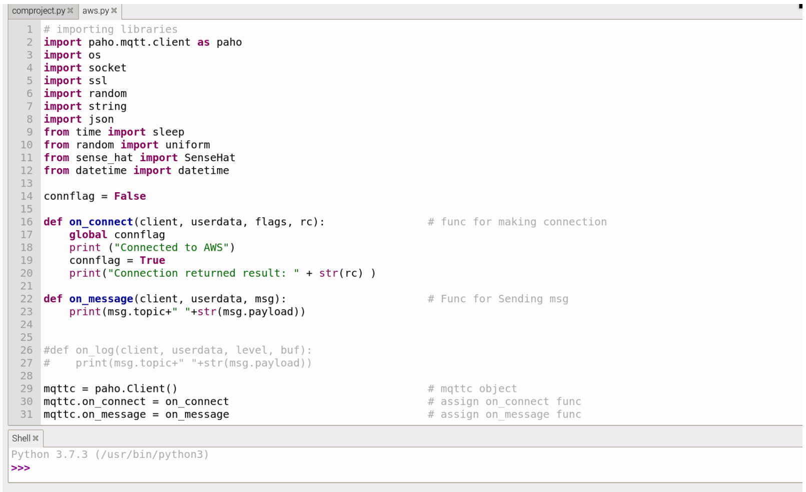


The following Picture Represents the output of the comproject.py code. Note: Although the code gives quite a warning, those warnings mention extra usage on the trial plan.



3.3.2 aws.py

aws.py python file created in raspberry pi imports several libraries and the code pushes the data from AWS to Google Firebase. The Data stored in AWS are in JSON Tree Format.



comproject.py aws.py

```
# importing libraries
import paho.mqtt.client as paho
import os
import socket
import ssl
import random
import string
import json
from time import sleep
from random import uniform
from sense_hat import SenseHat
from datetime import datetime

connflag = False

def on_connect(client, userdata, flags, rc):                # func for making connection
    global connflag
    print ("Connected to AWS")
    connflag = True
    print("Connection returned result: " + str(rc) )

def on_message(client, userdata, msg):                      # Func for Sending msg
    print(msg.topic+" "+str(msg.payload))


#def on_log(client, userdata, level, buf):
#    print(msg.topic+" "+str(msg.payload))

mqttc = paho.Client()                                       # mqttc object
mqttc.on_connect = on_connect                               # assign on_connect func
mqttc.on_message = on_message                               # assign on_message func
```

Shell

```
Python 3.7.3 (/usr/bin/python3)
>>>
```

The following Picture Represents the output of aws.py code.

comproject.py aws.py

```
#def on_log(client, userdata, level, buf):
#    print(msg.topic+" "+str(msg.payload))
```

Shell

```
Python 3.7.3 (/usr/bin/python3)
>>> %cd /home/pi/Desktop
>>> %Run aws.py
 Connected to AWS
 Connection returned result: 0
 WARNING:root:Failed to initialise TCS34725 colour sensor. (sensor not present)
 msg sent: environment

 Temperature : 27.0
 Humidity : 28.4
 Pressure : 0
```

3.3.3 Project RAW Stages

1. Set up the Raspberry Pi: Install the OS (Raspbian or Ubuntu) and configure the network. Install the necessary libraries and packages.

2. Connect the Sense Hat to the Raspberry Pi: Connect the Sense Hat to the GPIO pins of the Raspberry Pi. Install the necessary drivers and packages.

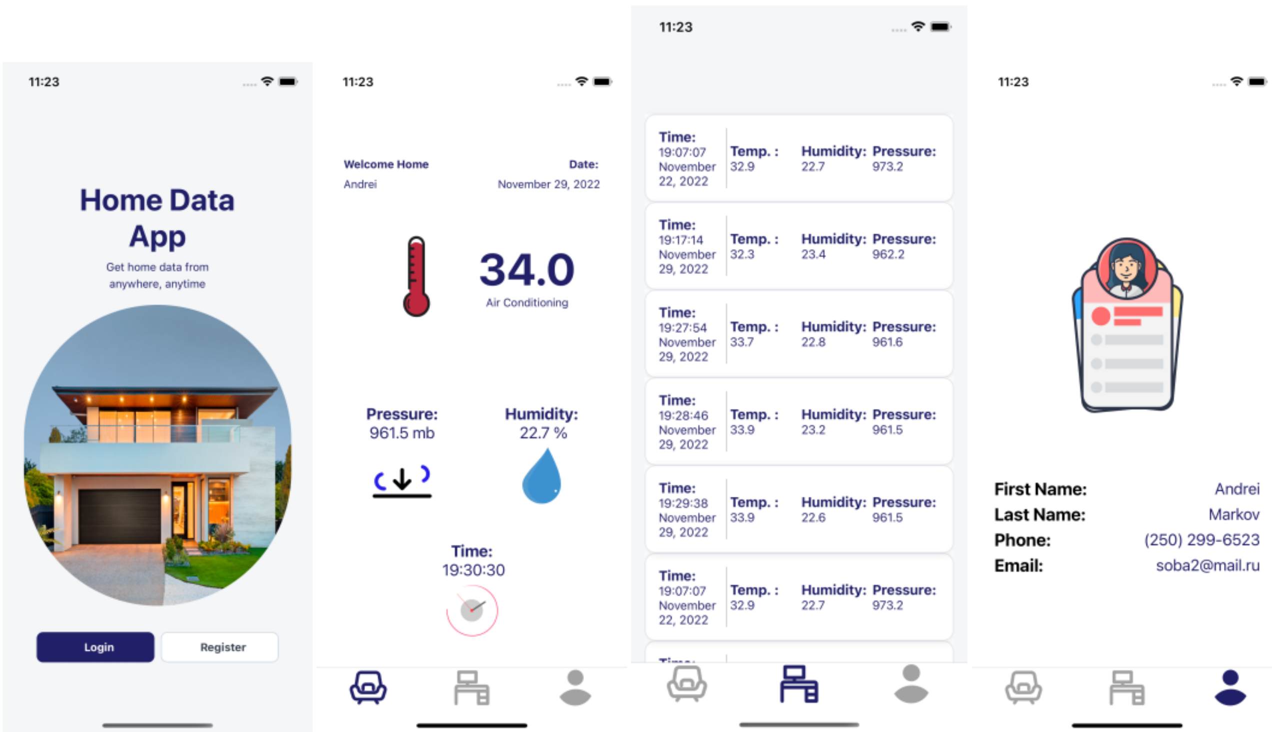
3. Configure the Sense Hat to collect data from the environment: Set up the sensors to monitor the environment. Set up the frequency of data acquisition and storage.

4. Set up the AWS and Google Firebase databases: Create an AWS and Google Firebase account. Configure the databases and set up the necessary security protocols.

5. Integrate the Sense Hat and the AWS and Google Firebase databases: Configure the Sense Hat to send the collected data to the AWS and Google Firebase databases. Set up the necessary authentication protocols.

6. Build the mobile iOS app to monitor the collected data: Create the app using iOS SDK. Connect the app to the AWS and Google Firebase databases. Design the user interface. Implement the necessary features.

3.4 APPLICATION BUILD



3.4.1 Application Description

The app is designed to allow users to monitor data collected from a Raspberry Pi and SenseHat device. The app collects data from Firebase and allows users to easily monitor the data from their mobile devices. The app also allows users to view a history of the data that has been captured. The app can be used to monitor temperature, humidity, pressure, and other environmental factors that can be detected by SenseHat. The user can also set up thresholds for each data type to track changes in the environment and receive notifications. In addition, the user can view a graph of historical data to monitor trends and changes over time. The app can be used to set alerts for specific readings, or even control the environment in a room. Overall, the iOS Home Monitor App is a powerful tool that allows users to monitor data collected from a Raspberry Pi and SenseHat device. Additionally, the user can view a history of the data that has been captured. Furthermore, the app will encrypt all data collected and stored in the Firebase database, ensuring that all user data is kept secure. Finally, users will be able to access their data from anywhere, as the app will be available on all iOS devices.David (2018)

CONCLUSION

In conclusion, the Raspberry Pi and SenseHAT have demonstrated that they are efficient and dependable option for monitoring the temperature, pressure, and humidity in the air or in the surrounding environment. The information that is obtained using the SenseHAT is uploaded to the cloud via AWS, and then the information is uploaded from the cloud to Google Firebase. An Adafruit Dashboard was constructed in order to view the live sensor data on the web, and an iOS app for mobile devices was designed in order to monitor the data as it was animated. The mobile app gives users the ability to see the history of data that was acquired in the past, as well as register for accounts and log in to those accounts. In general, this project has been effective in demonstrating the ability of Raspberry Pi with SenseHAT to provide a solution for monitoring air quality that is dependable and efficient in terms of cost. The project has the potential to be expanded so that more air quality metrics, such as dust, gases, and other contaminants, may be measured. Additionally, users will be able to define thresholds for certain characteristics if the mobile app is upgraded to include this functionality. In addition, there is the possibility that the project might be expanded into other application areas, such as the monitoring of the soil and water quality in agricultural applications. Users will be able to connect to a wide number of sensors and data sources using the Adafruit IO platform. This will enable users to monitor and view various kinds of data in real-time. Users have more control over their data and more visibility into it because to Adafruit. In addition, users are able to personalize their dashboards by selecting from a range of themes and graphical components, which enables them to generate effective representations of the sensor data that they have collected.

REFERENCES

Childs-Maidment, J. (2017). “Pyrebase a simple python wrapper for the firebase api.

David, T. (2018). “The comparison between swift and java programming language.” *Bachelor thesis, University of Ostrava*.

Guth, J., Breitenbücher, U., Falkenthal, M., Leymann, F., and Reinfurt, L. (2016). “Comparison of iot platform architectures: A field study based on a reference architecture.” *2016 Cloudification of the Internet of Things (CIoT)*, IEEE, 1–6.

Manjula, T., Sreenivasulu, U., and Hussain, S. J. (2016). “A dynamic raspberry pi sense hat multimodality alerting system by using aws iot.” *Indian Journal of Science and Technology*, 9, 39.