UE17CS490B – Capstone Project Phase – 2

SEMESTER - VIII

END SEMESTER ASSESSMENT

Project Title :Monitoring and Controlling the Crops in Warehouse in IOT
Project ID :PW21RBS01
Project Guide :Prof. Rachana B S
Project Team :1279_2403_2436

Outline

- Abstract
- Requirements and Design
- Methodology / Approach
- Design Description
- Modules and Implementation Details
- Project Demonstration
- Test Plan and Strategy
- Results and Discussion
- Lessons Learnt
- Conclusion and Future Work
- Team Roles and Responsibilities.
- References

Abstract

PROBLEM STATEMENT: To provide efficient decision support system using wireless sensor network which handle different activities of Warehouse and gives useful information related to warehouse to farmer and food spoilage detection using MQ-135 for food storage activities.

INTRODUCTION:

Our Project goal is to provide Automation to daily Food Storage Warehouse activities such as controlling Temperature, Humidity, Motion detection etc..,

The above mentioned activities are done using Sensors which in turn are connected to Microcontroller through which data is collected and stored in cloud for further usage in future.

Food Spoilage Detection is done through by capturing the Co_2 of the food which is stored in warehouse and sends text message to the user.

Scope

A. Provide Automation

B. Reduce Manual Work

C. Provide Better Food

D. Reduce Food Misfortunes

E. Increase Food Availability

- Benefits: The vital advantage of this cycle is that it permits the stakeholders to comprehend the present status of the undergoing Increase in Food Availability.
- Usage: Producers, Dealers, Traders, Distributors.

Fig 1: Crop

Requirements

Hardware Requirements
Raspberry Pi 3 B+
DHT22 Sensor
PIR Sensor
Fire Sensor
MQ-135 Sensor
MCP-3008
Bulb
Fan
Buzzer
5V Relay

Software Requirements
IBM Cloud

Constraints

Constraints
Revenue and Affordability
Power Supply
Communication Range
Harsh Device Environment
Quality and Availability of Data
Quality of Wireless Link

Assumptions

Assumptions
Gracefully merchandise into the commercial center by having the option to store products when flexibly surpasses request and afterward delivering them when request surpasses.
Keeping up reliable stock levels causes costs to remain stable, making it simpler for organizations to estimate creation, benefit and misfortune.

Risks

- Technical Failure and Resultant Damage
- Data Security and Privacy
- Context-Awarness (meta data)
- Food Wastage

Fig 2: Security

Methodology

- Methodology tries to explore technological solutions to enhance food availability.
- Improving the smart warehouse system or food storage system based on embedded electronics, IoT and wireless sensor networks.
- Maintain the food products to survive for long time than its original time.

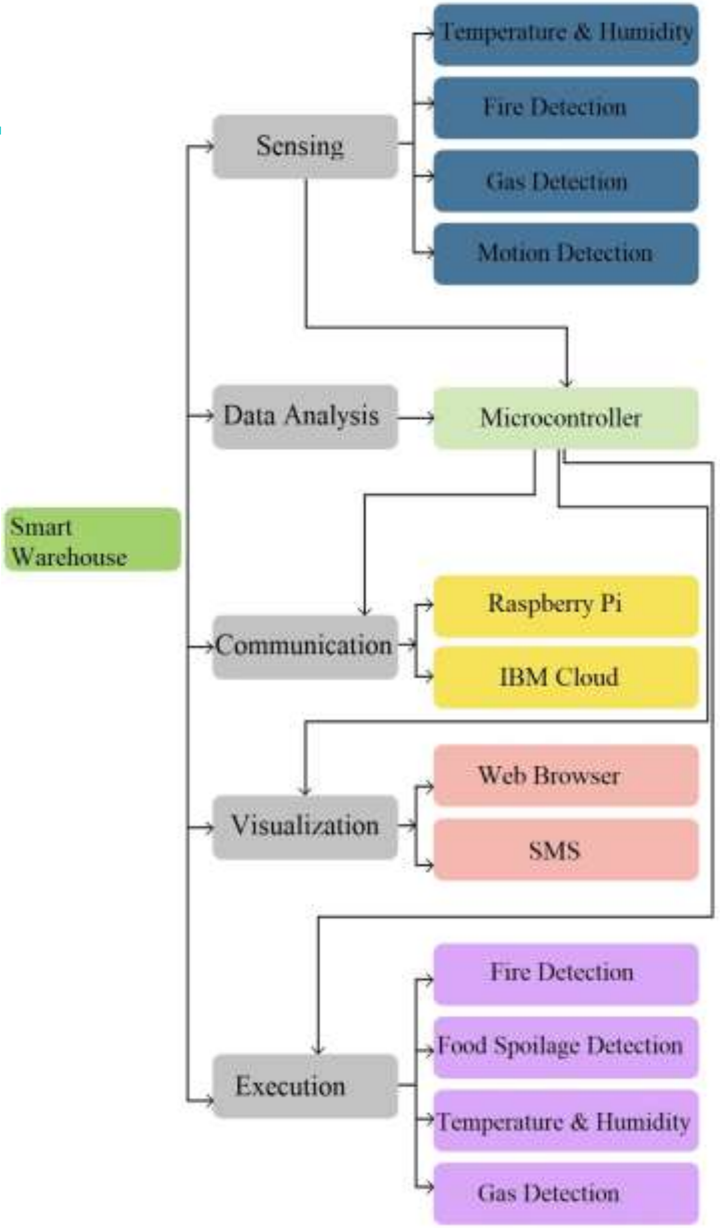


Fig 3: State Diagram

Design Approach

Design Goals:

- **Physical Implementation :** a)Execution of Sensors, Actuators and Micro Controllers b) Implementation of Network Equipment.
- **System Framework :**

  Subsystems:

  a) Sensing: Collection of Data (environment) through Sensors.

  b) Data Communication: I) Communication between the central and wireless sensor nodes.

  II) Sensor Data is collected into sheets, various Databases, Cloud Storage.

  c) Visualization : visualization, processing and manipulation of data.

  d) Data Analysis: Data is analyzed using different algorithms.
- Through this the system is capable of decision making and execution based on manipulating sensor data.
- **Circuit Design:**
- **Wireless Sensor Nodes:** Observing and Recording the states of being of the climate and putting together the gathered information at a central location..
- **Central Node:** Consists of Sheets, Various Databases, Cloud storage to collect data.

Implementation Details

Technologies Used:

- IOT
- Cloudant
- IBM CLOUD
- Node Red

FLOW CHART:

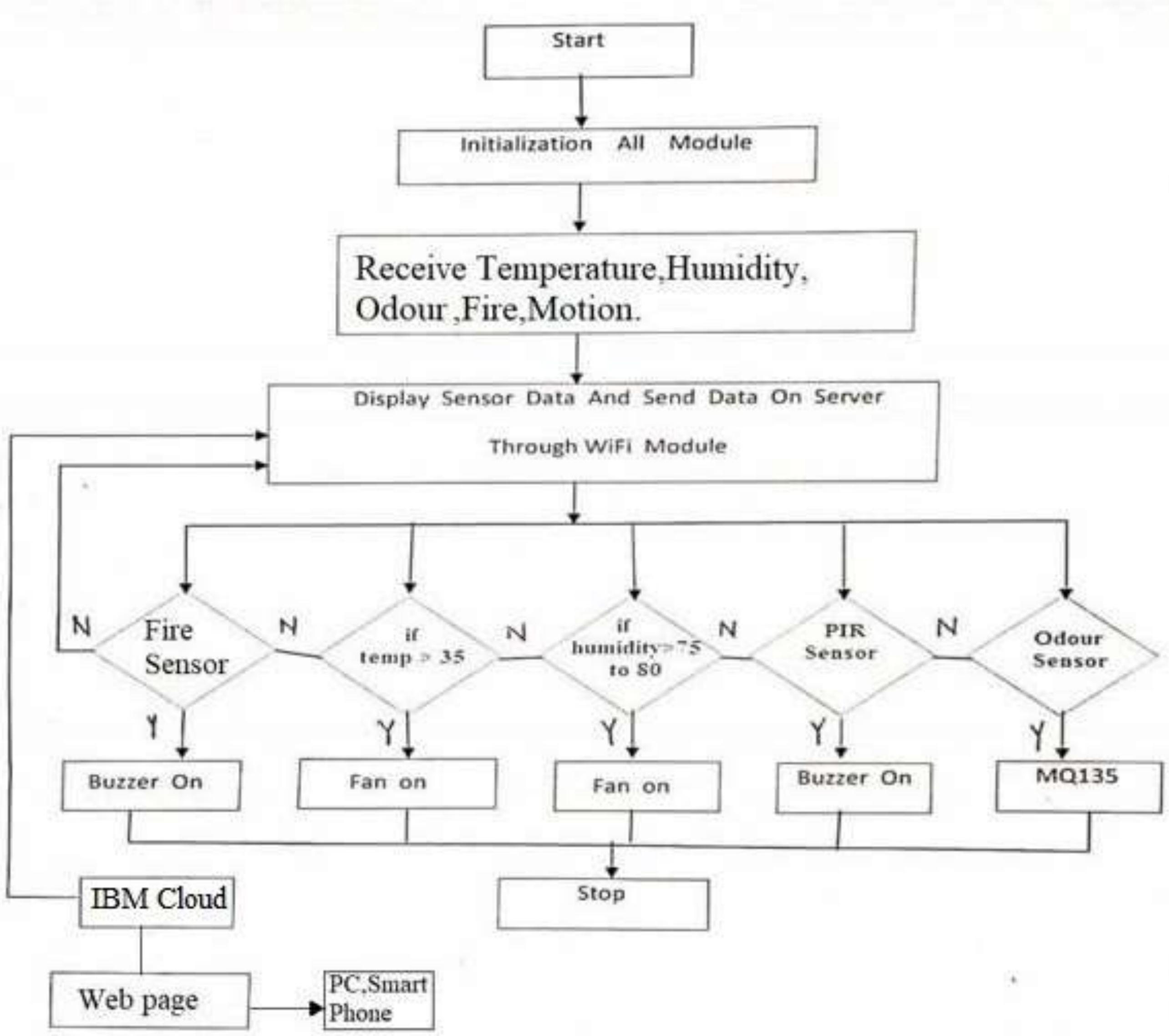


Fig 4:Flow Chart

Design Description

CLASS DIAGRAM

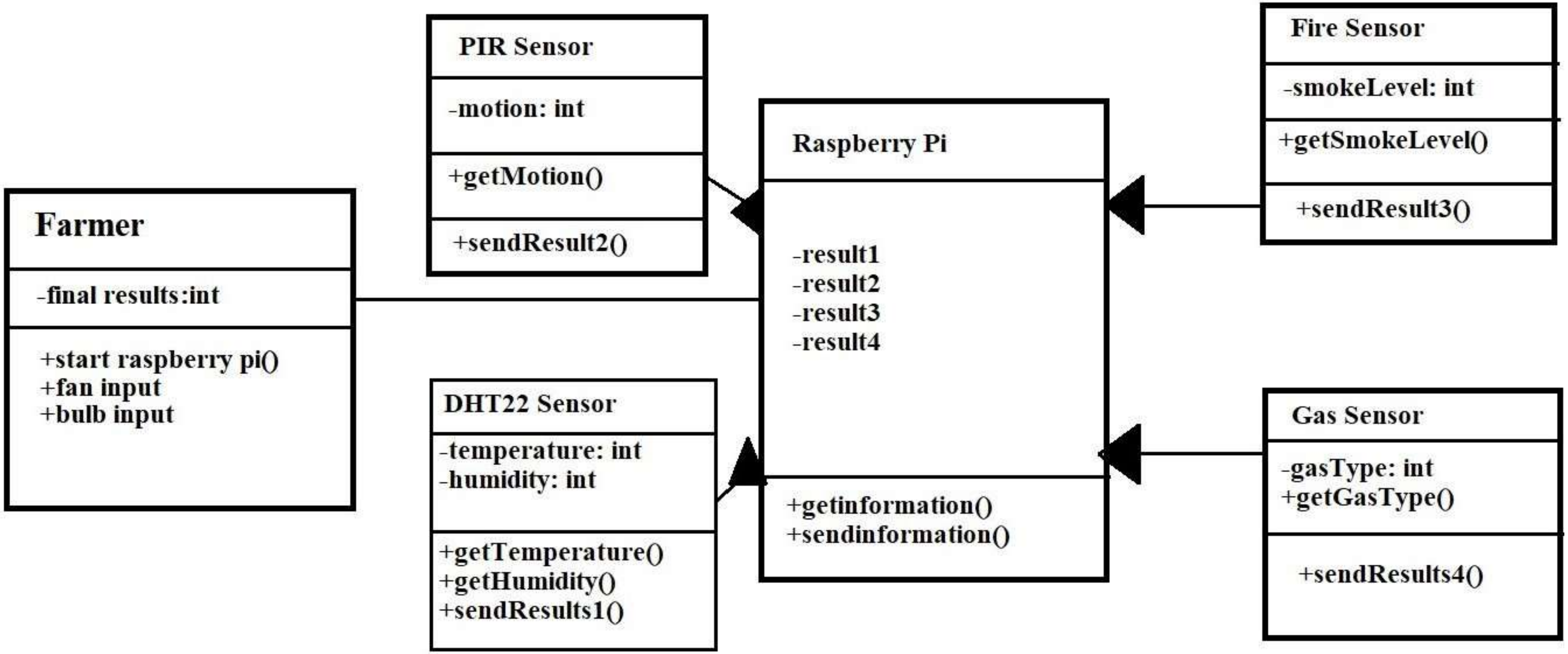


Fig 5: Class Diagram

Use Case Diagram

- To Model Functionalities of the System

  We Used Different:

  1)Roles-Clients,developer

  2)Entities-Raspberry Pi,Sensors

  3)Associated Use Cases

- The User Class Diagram here represents different functions, Services and actions performed in the system.

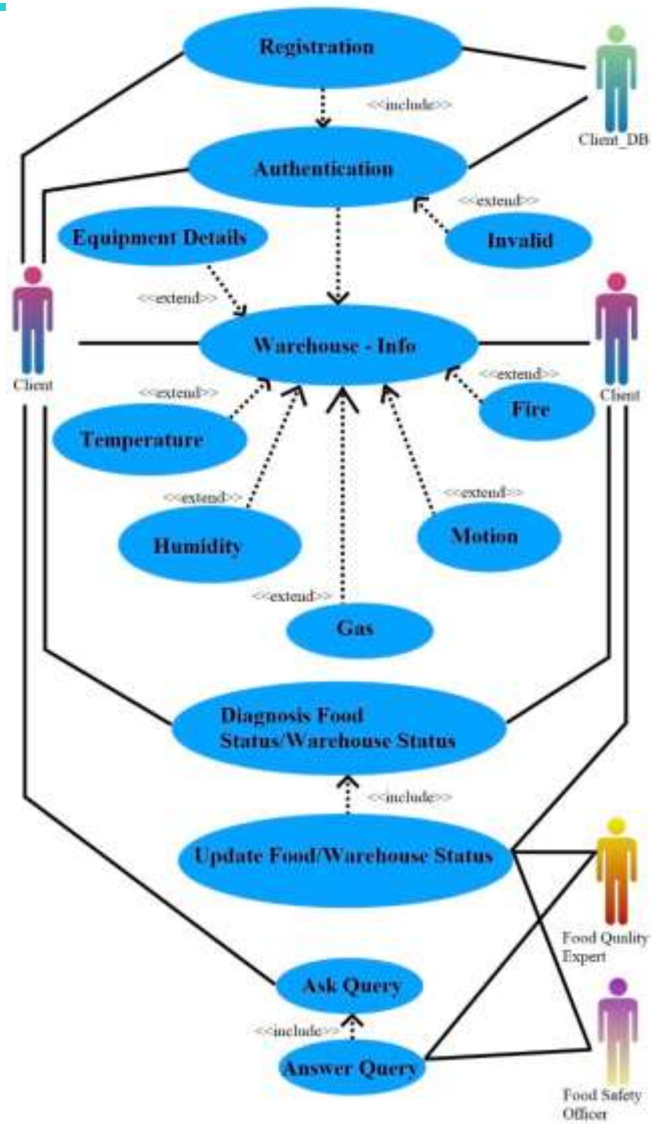


Fig 6:Use Case Diagram

Sequence Diagram

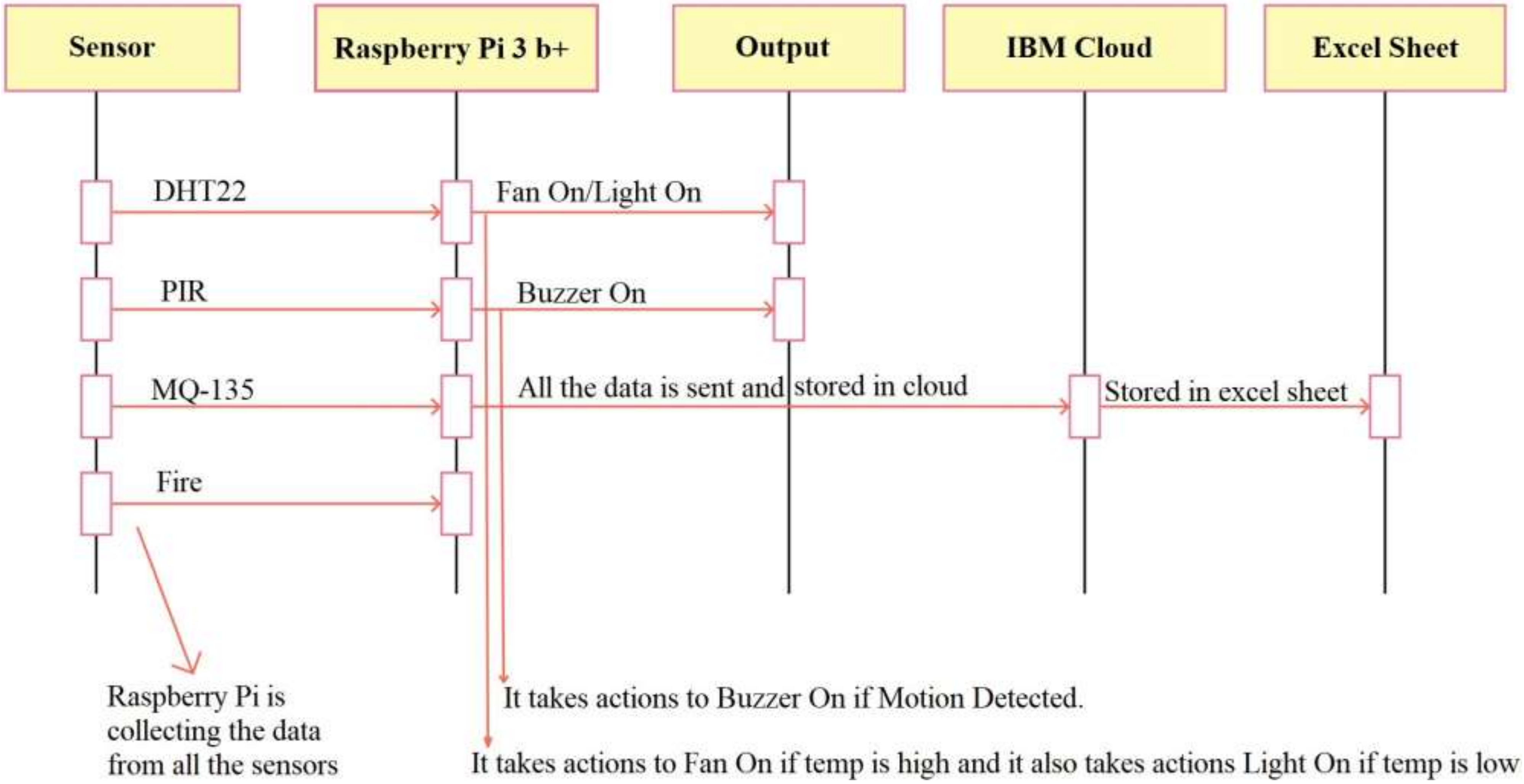


Fig 7:Sequence Diagram

Modules and Implementation Details

Module Name

- Device Module -
  - Raspberry Pi
  - DHT22
  - PIR
  - MQ135
  - Fire
- Cloud-IBM
- Web Application-Node Js

Module 1

- Device Module

Description

- All sensors are connected to Raspberry Pi which is the device module.

Module 2

- IBM Cloud

Description

- All the data which is collected by the sensors and sent to raspberry pi will be displayed and stored in IBM Cloud

Module 3

- Web Application

Description

- All the data displayed in IBM cloud will also be displayed in the website in charts, gauge etc.

Modules and Implementation Details

- **PIR Motion Detection Sensor**

```
GPIO.setup(23, GPIO.IN)
   sleep(0.5) #Wait 5 seconds and read again
    if GPIO.input(23):#pir
     M='detected'
     GPIO.output(24, True)#buzzer
     print("Motion Detected...")
     time.sleep(1)
    else:
      M='Not Detected'
      GPIO.output(24, False)
    q=requests.get('https://www.fast2sms.com/dev/bulkV2?authorization=CZajTYBFxqW9thEI
    uMd3QKlceHgikzSyRo0JL7b6UPVn4fpvsDJd6pvkVMt71eGDcnKIRBqQHzgNo5L9&me
    ssage= Alert!!Motion is Detected&language=english&route=q&numbers=7993397777')
    print(q)
```

Modules and Implementation Details

- **DHT22 Sensor**

```
import Adafruit_DHT
DHT_SENSOR = Adafruit_DHT.DHT22
DHT_PIN = 4
GPIO.setup(16,GPIO.OUT)#fan
GPIO.setup(27,GPIO.OUT)#Light
T=0
H=0
while True:
  H, T = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)
  sensorValue = _range(channel_0.value, 0, 60000, 0, 1023)#gas sensor
  if H is not None and T is not None:
    print("Temp={0:0.1f}*C  Humidity={1:0.1f}%".format(T, H))
  else:
    print("Failed to retrieve data from humidity sensor")
  if T>30:
    GPIO.output(16,True)
    GPIO.output(27,False)#light
    print("Fan ON")
  else:
    GPIO.output(27,True)
    GPIO.output(16,False)#fan
```

Modules and Implementation Details

- Fire Sensor

```
channel=25
GPIO.setup(channel,GPIO.IN)#flame
flame=0
def callback(channel):
   print("channel",channel)
   global flame
   flame=1
   print("flame detected")
   r=
requests.get('https://www.fast2sms.com/dev/bulkV2?authorization=CZajTYBFxqW9thEIuM
d3QKlceHgikzSyRo0JL7b6UPVn4fpvsDJd6pvkVMt71eGDcnKIRBqQHzgNo5L9&message
=Alert!!Fire Detected in Warehousee&language=english&route=q&numbers=7993397777')
   print(r)
GPIO.add_event_detect(channel, GPIO.BOTH, bouncetime=300)  # let us know when the pin
goes HIGH or LOW
GPIO.add_event_callback(channel, callback)  # assign function to GPIO PIN, Run function
on change
```

Modules and Implementation Details

- MQ-135

```
import adafruit_mcp3xxx.mcp3008 as MCP
from adafruit_mcp3xxx.analog_in import AnalogIn
  # Create the SPI bus
  spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)

  # Create the cs (chip select)
  cs = digitalio.DigitalInOut(board.D5)

  # Create the mcp object
  mcp = MCP.MCP3008(spi, cs)
  while True:
    H, T = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)
    sensorValue = _range(channel_0.value, 0, 60000, 0, 1023)#gas sensor
  if (T>40 and sensorValue>150):
      print("Spoil Alert!!!")

  q=requests.get('https://www.fast2sms.com/dev/bulkV2?authorization=CZajTYBFxqW9thEIu
  Md3QKlceHgikzSyRo0JL7b6UPVn4fpvsDJd6pvkVMt71eGDcnKIRBqQHzgNo5L9&messa
  ge=Food Spoil Alert&language=english&route=q&numbers=7993397777')
      print(q)
```

Modules and Implementation Details

- **MCP-3008**

```
from spidev import SpiDev
class MCP3008:
  def __init__(self, bus = 0, device = 0):
    self.bus, self.device = bus, device
    self.spi = SpiDev()
    self.open()
    self.spi.max_speed_hz = 1000000 # 1MHz
  def open(self):
    self.spi.open(self.bus, self.device)
    self.spi.max_speed_hz = 1000000 # 1MHz
  def read(self, channel = 0):
    cmd1 = 4 | 2 | (( channel & 4) >> 2)
    cmd2 = (channel & 3) << 6
    adc = self.spi.xfer2([cmd1, cmd2, 0])
    data = ((adc[1] & 15) << 8) + adc[2]
    return data
  def close(self):
    self.spi.close()
```

Modules and Implementation Details

- **IBM CLOUD**

```
deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method":
authMethod, "auth-token": authToken}

deviceCli = ibmiotf.device.Client(deviceOptions)
```

- **Call Back**

```
def myOnPublishCallback():
print ("Published Temperature = %s C" % T, "Humidity = %s %%" % H, "to IBM Watson")


success = deviceCli.publishEvent("iotproject", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoTF")
    time.sleep(1)
    flame=0
    deviceCli.commandCallback = myCommandCallback


data = { 'Temperature' : T, 'Humidity': H ,'Motion':M,'Flame':flame ,'Gas_Sensor':sensorValue}
print (data)
```

Demonstration

Test Plan

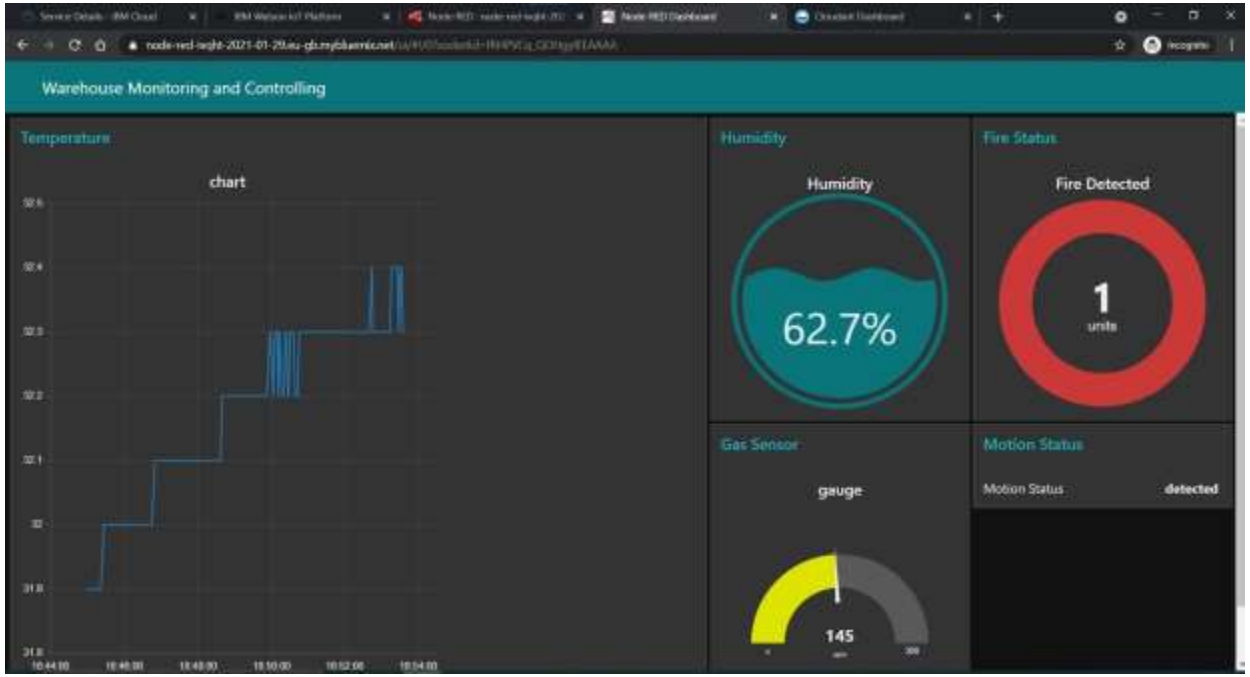
FUNCTIONS TESTED	**TEST RESULT**
Collect data	The sensors collected data and sent it to the Raspberry Pi
Transmit data	Data collected was transmitted to Cloudant
Retrieve data	After capturing data, we were able to view it on Google Sheets
Units Tested	
Temperature & Humidity sensor	This is highly sensitive area; during testing it was discovered that it is hard to determine the actual temperature of a Crop.
Fire Sensor	Fire detection is tested
Webpage	Tested the functionality and its connectivity to the database and web page
Notification	Tested whether the user's gets Alerts or not
Cloudant	Data was able to be stored and retrieved as required
Non - Functional Testing -Security:	To provide safe login for the user. Users can remotely monitor using .

Results and Discussion

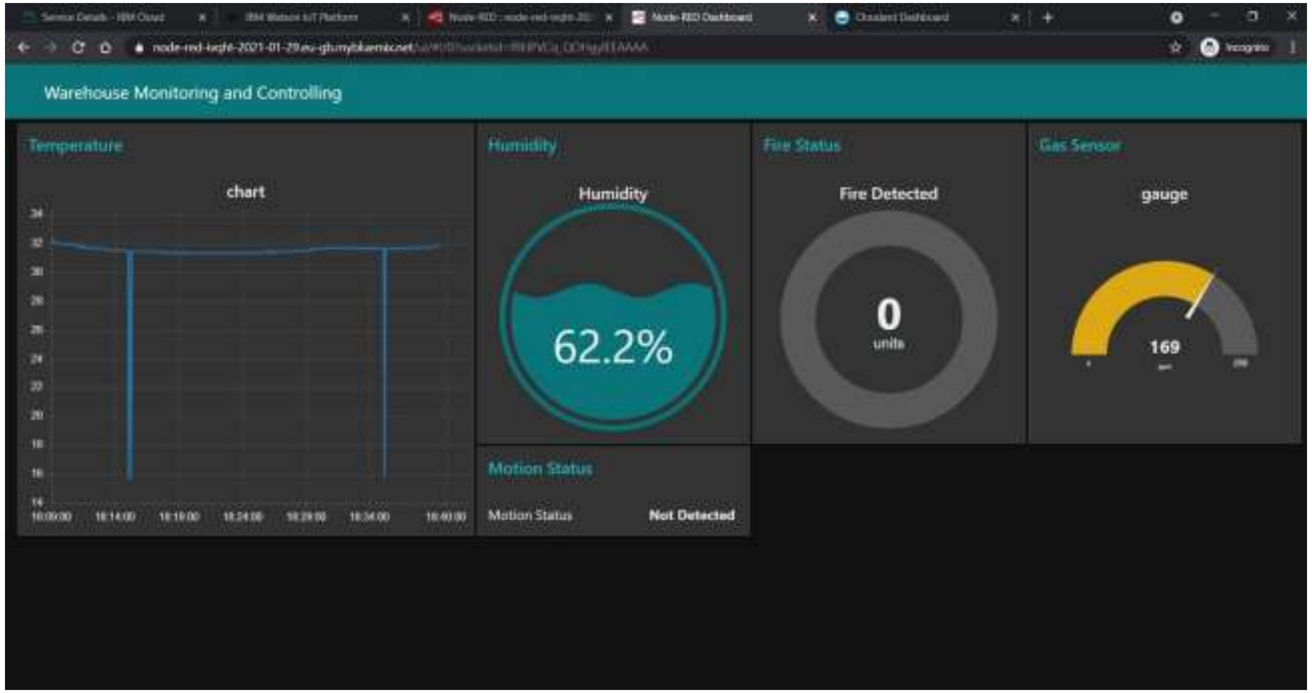
Crop Monitoring:
In this module it monitors the complete crop with the help of sensors and the data is visualized in cloudant and stored in sheets.

Monitoring and Controlling System:

Day 1:

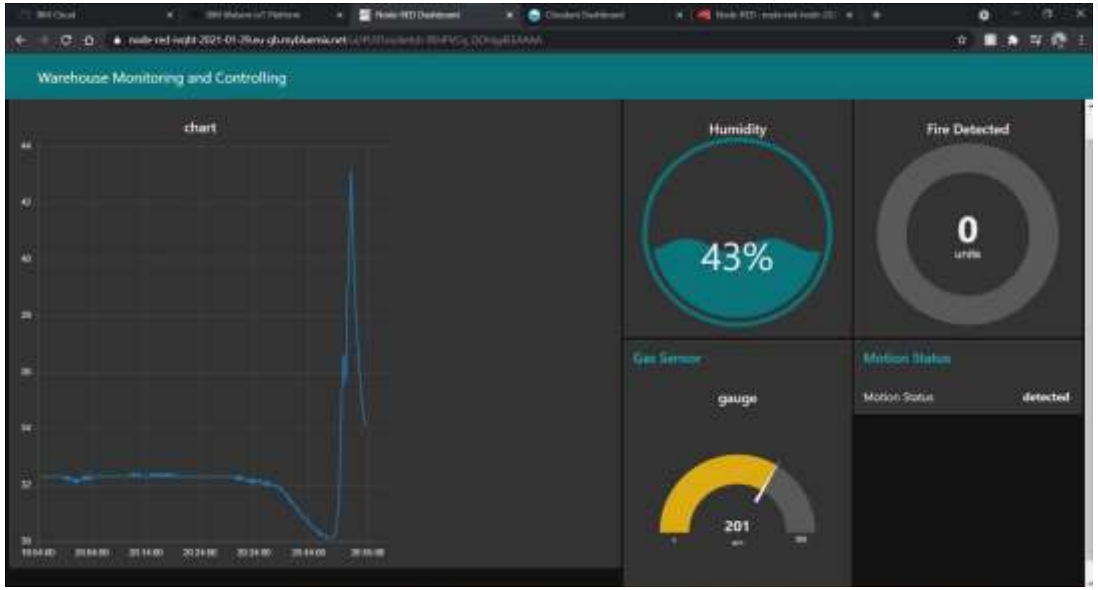


Results and Discussion



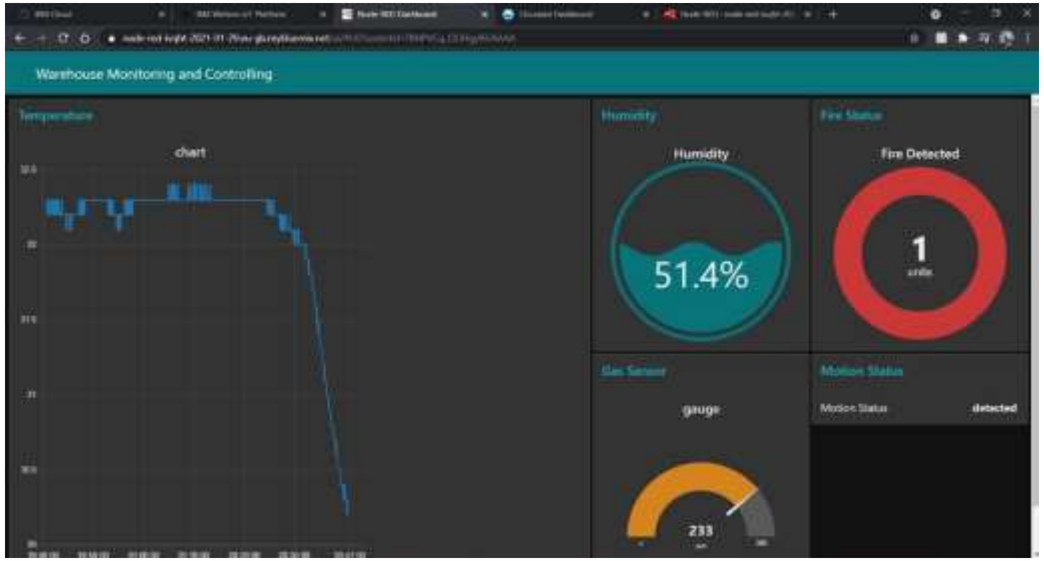
Results and Discussion

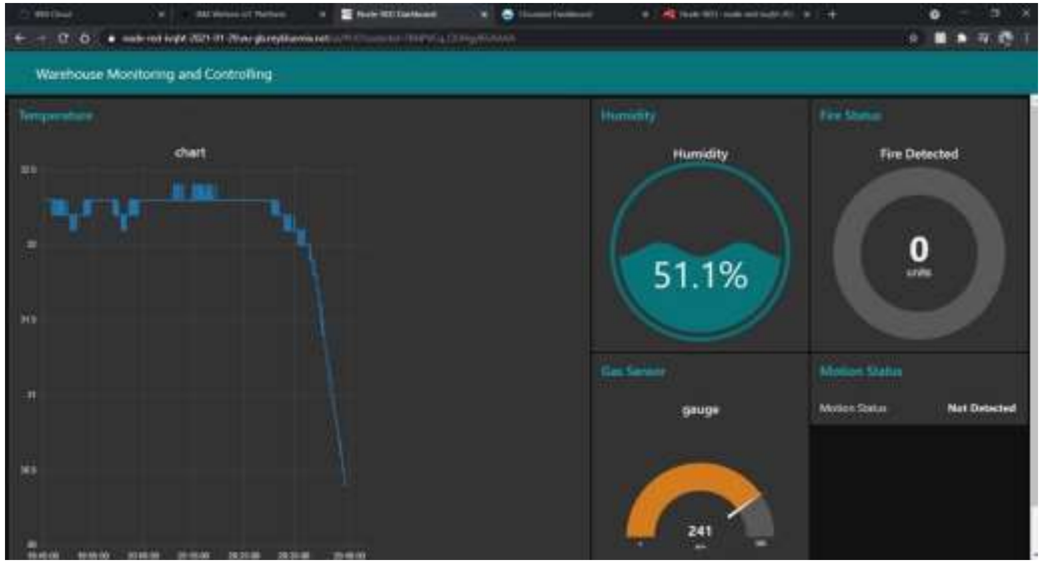
Day 2:



Results and Discussion

Day 3:



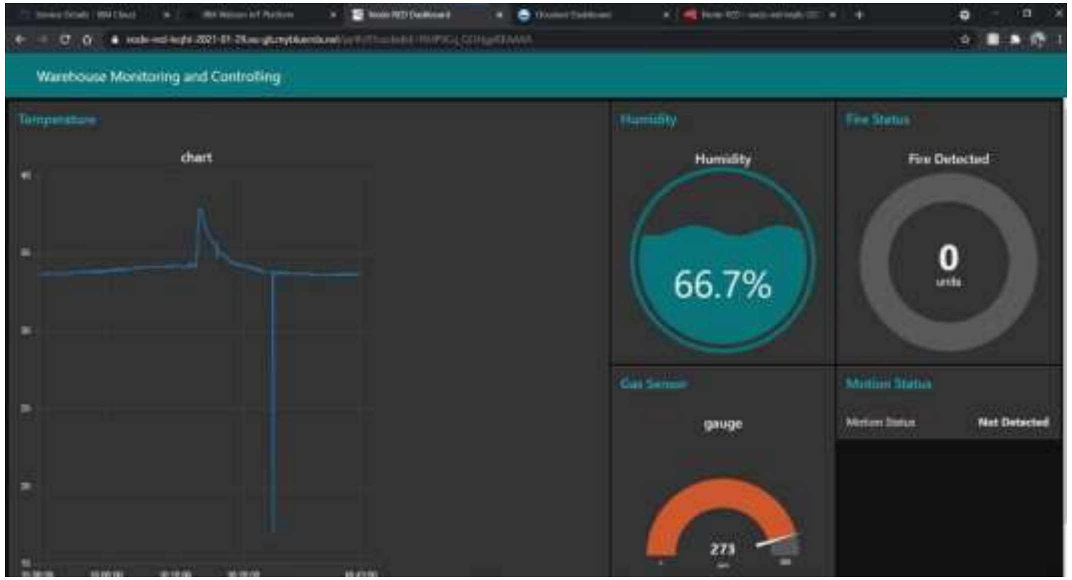


Results and Discussion

Day 4:

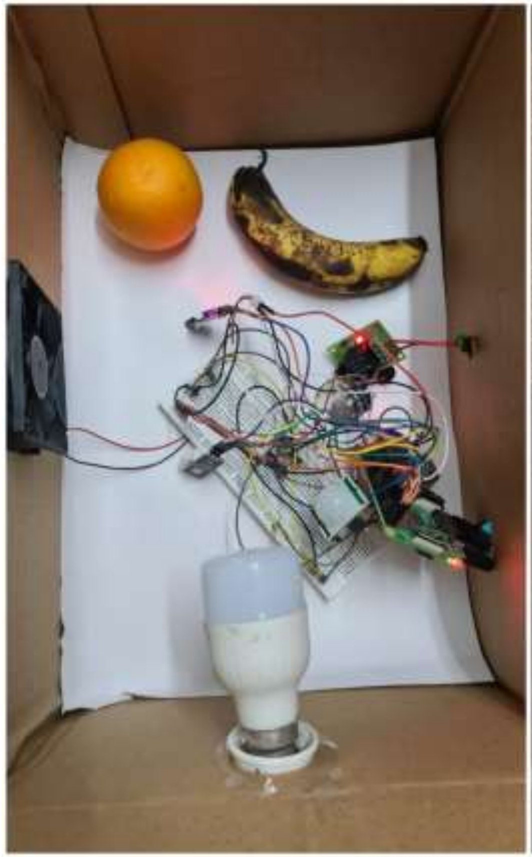
Results and Discussion

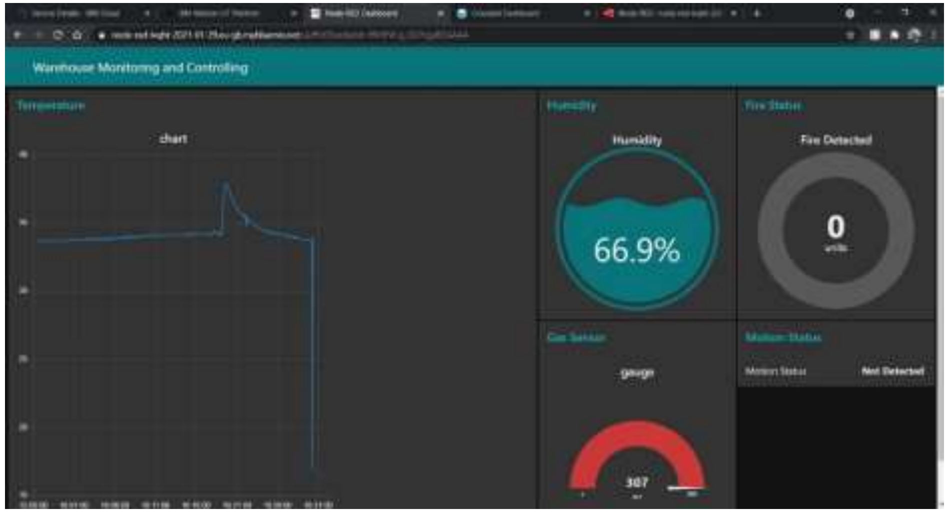
Day 5:



Results and Discussion

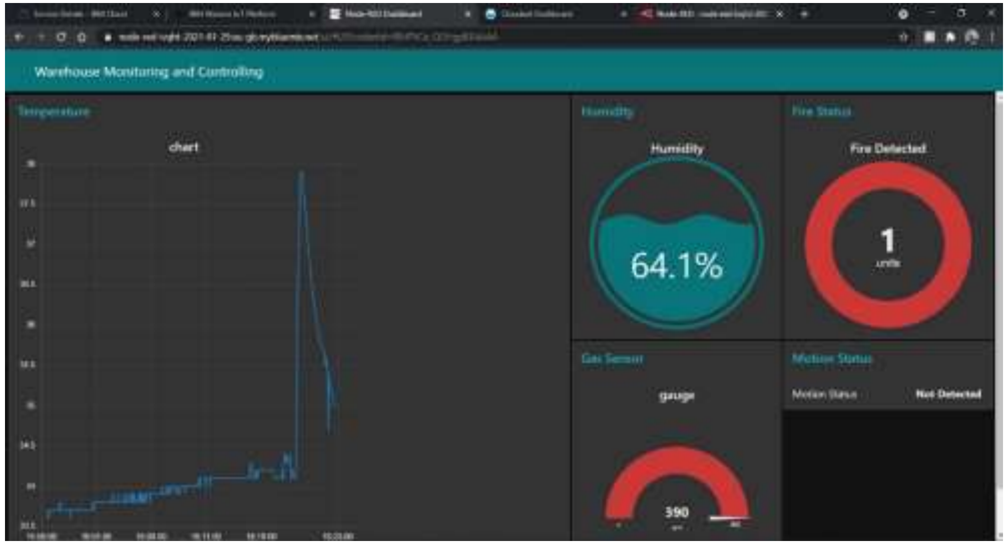
Day 6:





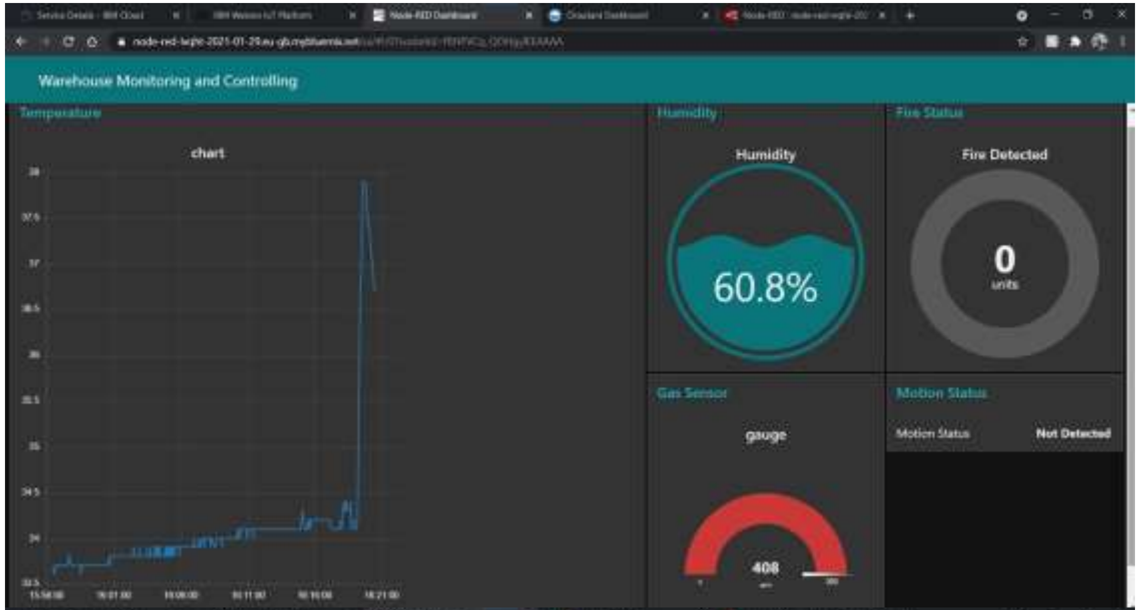
Results and Discussion

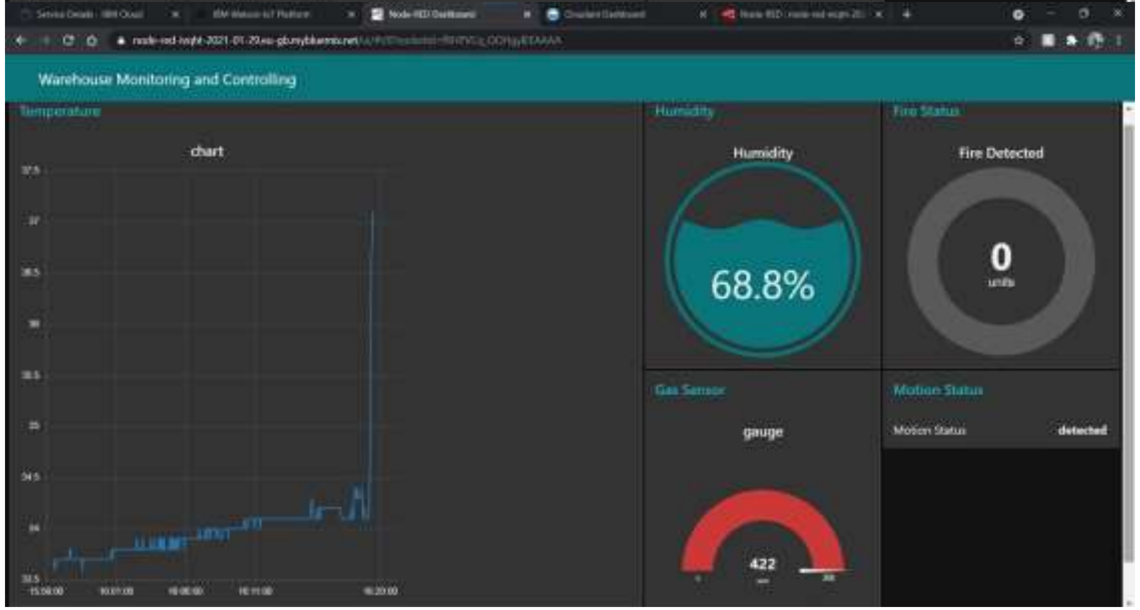
Day 7:



Results and Discussion

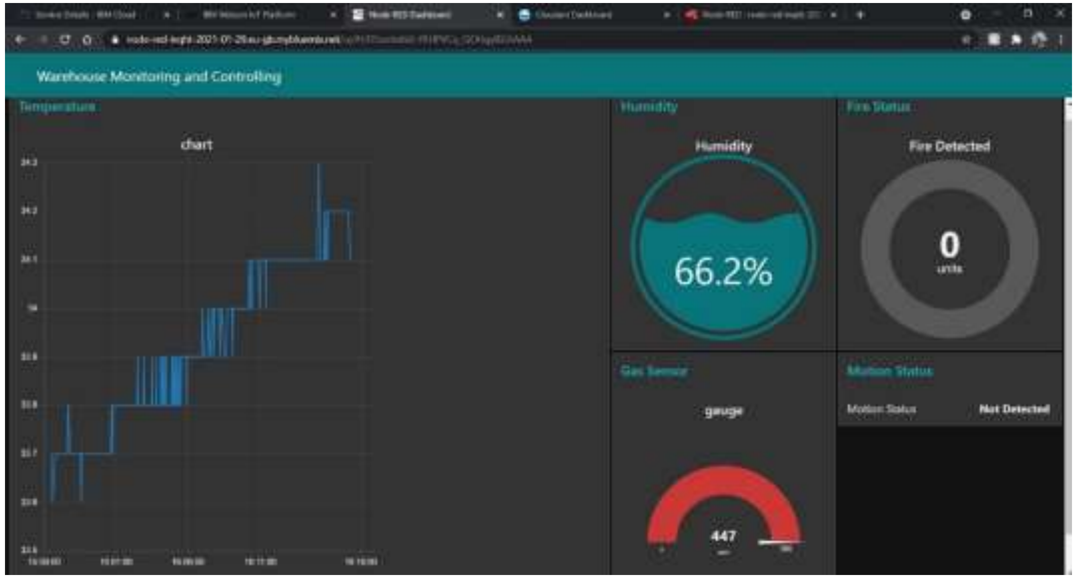
Day 8:

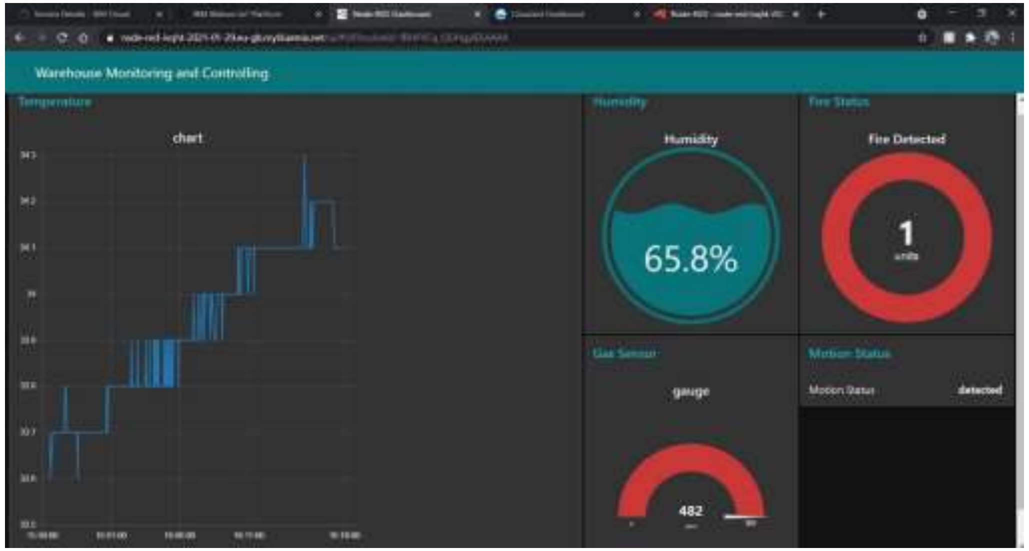




Results and Discussion

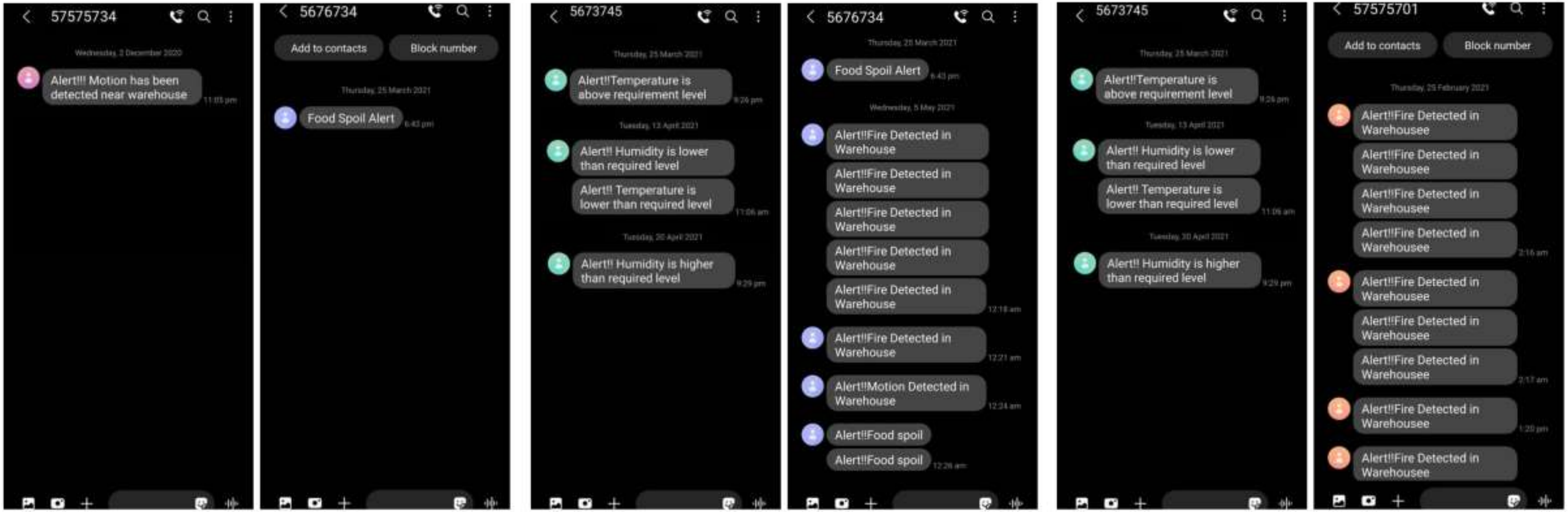
Day 9:





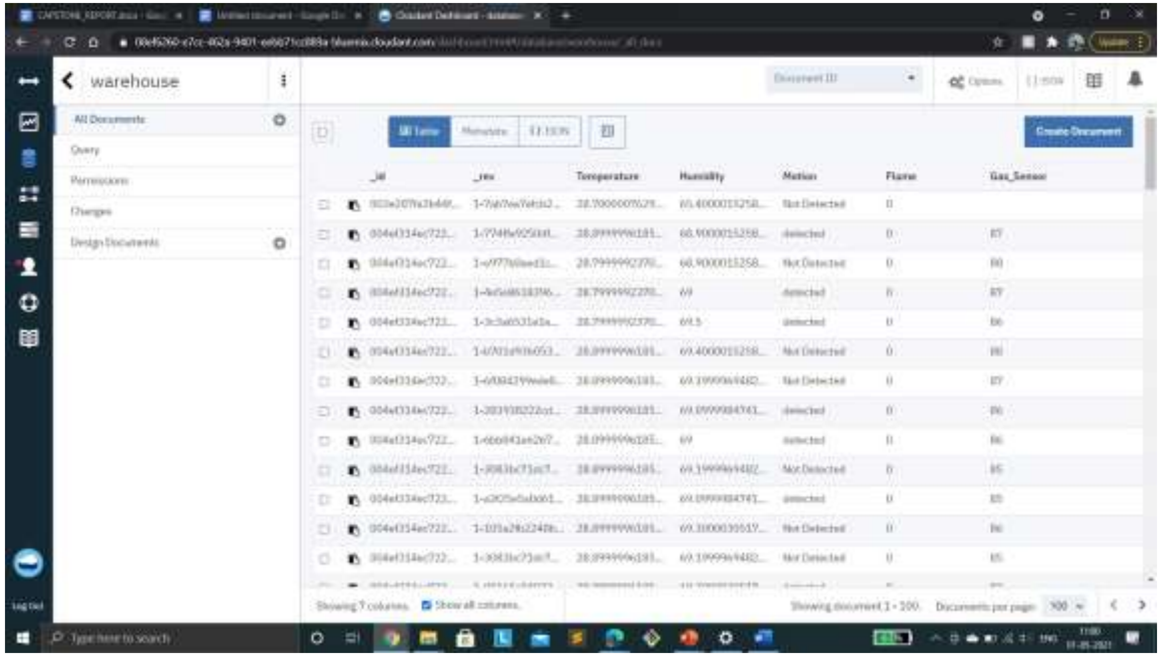
Results and Discussion

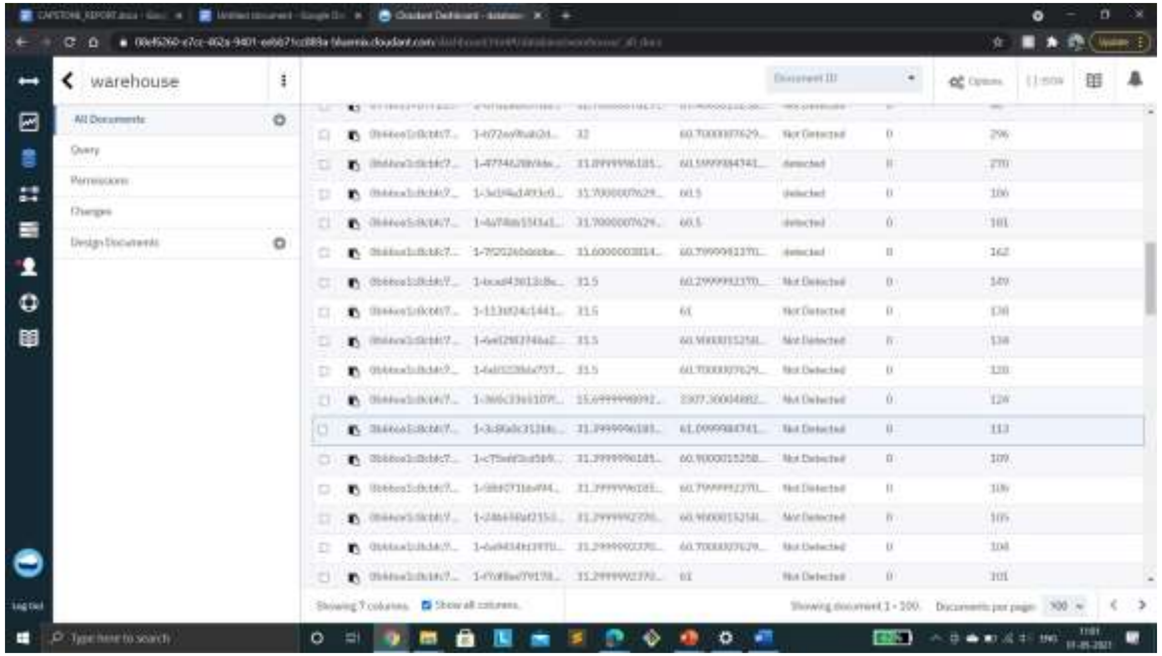
Messages :



Results and Discussion

Database:





Results and Discussion

Sheets:

Temperature(C	Humidity(%)	Gas	Motion	Fire	Fan	Light
31.5	61.09999847	127	Detected	0	ON	OFF
31.90000038	61.40000153	127	Detected	0	ON	OFF
31.5	61.20000024	127	Detected	Detected	ON	OFF
31.5	61.8	127	Not Detected	Detected	ON	OFF
31.5	61.8	127	Not Detected	Detected	ON	OFF
31.5	61.8	127	Not Detected	Detected	ON	OFF
31.5	61.8	125	Not Detected	Detected	ON	OFF
31.70000076	61.8	125	Not Detected	Detected	ON	OFF
31.70000076	61.8	125	Not Detected	Detected	ON	OFF
31.70000076	61.6	125	Not Detected	0	ON	OFF
31.70000076	61.09999847	125	Not Detected	0	ON	OFF
31.5	61.09999847	127	Not Detected	0	ON	OFF
31.5	61.09999847	127	Not Detected	0	ON	OFF
31.5	61.09999847	127	Detected	0	ON	OFF
31.5	61.09999847	127	Detected	0	ON	OFF
31.5	61.6	127	Detected	0	ON	OFF
31.8	61.6	127	Detected	0	ON	OFF
15.8	61.6	120	Detected	Detected	OFF	ON
15.8	61.6	120	Detected	Detected	OFF	ON
15.8	61.6	126	Detected	Detected	OFF	ON
15.6	61.6	126	Detected	Detected	OFF	ON
16.4	61.6	126	Detected	Detected	OFF	ON
16.4	61.6	126	Detected	Detected	OFF	ON
16.4	61.6	120	Detected	Detected	OFF	ON
16.4	45.7	120	Detected	0	OFF	ON
16.4	45.7	120	Not Detected	0	OFF	ON
16.4	45.7	120	Not Detected	0	OFF	ON
40.2	45.7	120	Not Detected	0	OFF	ON

Temperature(C	Humidity(%)	Gas	Motion	Fire	Fan	Light
52.20000076	61.09999847	200	Detected	0	ON	OFF
52.20000076	61.40000153	230	Detected	0	ON	OFF
52.20000076	61.20000024	230	Detected	Detected	ON	OFF
52.20000076	61.8	230	Not Detected	Detected	ON	OFF
52.20000076	61.8	200	Not Detected	Detected	ON	OFF
51.29999924	61.8	250	Not Detected	Detected	ON	OFF
51.29999924	61.8	250	Not Detected	Detected	ON	OFF
51.29999924	61.8	250	Not Detected	Detected	ON	OFF
51.29999924	61.8	250	Not Detected	Detected	ON	OFF
51.29999924	61.8	250	Not Detected	0	ON	OFF
51.29999924	61.09999847	210	Not Detected	0	ON	OFF
48.20000076	61.09999847	210	Not Detected	0	ON	OFF
42.20000076	61.09999847	210	Not Detected	0	ON	OFF
42.20000076	61.09999847	210	Detected	0	ON	OFF
41.20000076	61.09999847	210	Detected	0	ON	OFF
40.20000076	61.8	210	Detected	0	ON	OFF
40.20000076	61.8	250	Detected	0	ON	OFF
39.70999924	61.6	250	Detected	Detected	OFF	ON
39.70999924	61.6	260	Detected	Detected	OFF	ON
39.70999924	61.6	260	Detected	Detected	OFF	ON
38.70999924	61.6	250	Detected	Detected	OFF	ON
38.70999924	61.6	250	Detected	Detected	OFF	ON
39.70999924	61.6	200	Detected	Detected	OFF	ON
39.70999924	61.6	200	Detected	Detected	OFF	ON
37.70000076	45.7	250	Detected	0	OFF	ON
37.70000076	45.7	200	Not Detected	0	OFF	ON
37.70000076	45.7	200	Not Detected	0	OFF	ON
37.70000076	45.7	230	Not Detected	0	OFF	ON

Schedule


Capstone Phase-2
Task
January
February
March
April
w1 w2 w3 w4 w1 w2 w3 w4 w1 w2 w3 w4 w1 w2 w3 w4
Literature Survey
Implementation
Testing
Demonstration in real time
Hard and Soft copy of report

Lessons Learnt

- Current issues that warehouse owners or farmers facing in storing crops.
- How to implement new technologies.
- Integration between hardware components and software.
- Integration of different technologies(Cloud, IBM..)
- Executing a Project as a team.

Conclusion

- There are no limitations on how many connections and sensors can be set up in an IOT smart warehouse system.
- The discussed model and IOT system design explored in this project can be implemented in a small setting like a storage house in the backyard and even on a large-scale warehouse.
- All the components and design can be scaled down to meet the user’s needs. However, it should be noted that the smart warehouse(as an application of IOT Technology) has not been given much attention and sufficient research.
- This basically happened because this kind of warehouse is costly and it's available only in highly developed areas.
- But given more attention especially from countries with much agricultural exports,smart warehouses have the potential to revolutionize the way warehouses operate and hence to boost the food availability and food market around the world.

Future Work

- USSD functionality to implement an offline action control mechanism for warehouse management.
- We can use ML Algorithm.
- We can add more sensors likeMQ-2,IR Sensor,LDR Sensor etc..
- We can add a heater instead of light.
- We can add LCD Display

Team Roles and Responsibilities

TEAM MEMBERS	ROLES AND RESPONSIBILITIES
ARUMILLI MEGHANA	Monitoring, Documentation, Research on Cloud Platforms and Algorithms and Technologies, Slides for Presentation, IEEE(Literature Survey, Second Paper)
CHAITRA VISHANU DEVANGA	Monitoring, Research on Cloud Platforms and Technologies, Documentation, IEEE(Literature Survey,Second Paper)
ANUSHA T N	Monitoring , Documentation,IBM Cloud, Algorithms and Slides for Presentation, IEEE(Literature Survey,Second Paper)

References

[1.]Kavya P, Pallavi K N, Shwetha M N, Swetha K, Mrs. Jayasri B S ,“Use of Smart Sensor &IoT to Monitor the Preservation of Food Grains at Warehouse”;2017,IJRTI

[2.]T. N. Anil Kumar, Bevinahal Lal swamy,Y. Raghavendra, S.G.Usharani, S. Usharani ,“Intelligent Food and Grain Storage Management System for the Warehouse and Cold Storage";2018;IJRESM

[3.]Sazia Parvin; Amjad Gawanmeh; Sitalakshmi Venkatraman ,“Optimised Sensor Based Smart System for Efficient Monitoring of Grain Storage”;2018,IEEE Xplore

[4.]Asif Bin Karim, Md Zahid Hassan, Md Masum Akanda, Avijit Mallik ,“Monitoring food storage humidity and temperature data using IoT”;2018,Med Crave

[5.]Ajay Doltade, Ankita Kadam, Sayali Honmore, Sanjeev Wagh ,“Intelligent Grain Storage Management System based on IoT”;2018,IJSR

[6.]K.Mohanraj, S.Vijayalakshmi, N.Balaji, R.Chithra kannan, R.Karthikeyan, “Smart Warehouse Monitoring Using IOT”,2019, IJEAT

[7.]Dr.R.M. Rewatkar , Mr. Akash V. Gulhane , Ms. Ashwini S. Mungale, Ms. Nikita M. Dhengare , Ms. Pooja P. Thakare , Ms.Prajakta D. Sabane “IOT Based Smart Real Time Agriculture Warehouse Monitoring And Control System Using Raspberry Pi”, 2019,IJRECE

[8.]Automated Granary Monitoring And Controlling System Suitable For The Sub-Saharan Region, Temesgen Belay Tedla, Joe Joe L. Bovas, Yonas Berhane, Maxim Nikolaevich Davydkin, Shaji James P.;2019,IJSTR

References

[9.]Prediction and Monitoring of stored food grains health using IoT Enable Nodes,Aseem Sindwani; Akhilesh Kumar; Chitra Gautam; Gaurav Purohit; Pramod Tanwar;2020,IEEE Xplore

[10.]IOT Instrumented Food and Grain Warehouse Traceability System for Farmers, Susmita Banerjee, Anil Kumar Saini, Himanshu Nigam, Vijay;2020,IEEE Xplore.

[11.]IOT Based Food Monitoring System in Warehouses, Shivani Bhandari , Pooja Gangola , Shivani Verma , Surekha K Si;2020,IRJET

[12.]IOT Monitoring System for Grain Storage, Ravi Kishore Kodali; Jeswin John; Lakshmi Boppana;2020,IEEE Xplore

Thank You