PROJECT REPORT

IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

TEAM ID: PNT2022TMID17561

ROLL NUMBER	NAME
713319EC009	ARSHAD PARVEZ G
713319EC056	LAVANYA M
713319EC052	KISHORE KUMAR K
713319EC062	MATHUPRAKAS R

CONTENT

1. **INTRODUCTION**
2. **LITERATURE SURVEY**
3. **IDEATION & PROPOSED SOLUTION**
   - 3.1 Empathy Map Canvas
   - 3.2 Ideation & Brainstorming
   - 3.3 Proposed Solution
   - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   - 4.1 Functional requirement
   - 4.2 Non-Functional requirement
5. **PROJECT DESIGN**
   - 5.1 Data Flow Diagrams
   - 5.2 Solution & Technical Architecture
   - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   - 6.1 Sprint Planning & Estimation
   - 6.2 Sprint Delivery Schedule
7. **CODING & SOLUTIONING**
   - 7.1 Feature 1
   - 7.2 Feature 2
   - 7.3 Feature 3
8. **RESULTS**
   - 8.1 Performance Metrics
9. **ADVANTAGES & DISADVANTAGES**
10. **CONCLUSION**
11. **FUTURE SCOPE**
12. **APPENDIX**
    - Source Code
    - GitHub & Project Demo Link

ABSTRACT

IOT Based Smart Crop-Protection for Agriculture monitoring is a system describes how to monitor crop field. It is developed by using sensors and according to the decision from a server based on sensed data, the irrigation and monitoring system is enhanced. Through wireless transmission the sensed data is forwarded to web server database. If the irrigation is automated, then the moisture and temperature fields are decreased below the potential range. The user can monitor and control the system remotely with the help of application which provides a web interface to user. By smart Agriculture monitoring system and one of the oldest ways in agriculture is the manual method of checking the parameters. In this method farmers by themselves verify all the parameter and calculate the reading. It aims at making agriculture smart using automation and IoT. The cloud computing devices are used at the end of the system that can create a whole computing system from sensors to tools that observe data from agriculture field. It proposes a novel methodology for smart farming by including a smart sensing system and smart irrigator system through wireless communication technology. This system is cheap at cost for installation. Here one can access and control the agriculture system in laptop, cell phone or a computer.

Chapter – 1

INTRODUCTION

A system using sensors that monitor different conditions of environment like humidity, temperature etc., the processor and GUI module is used. The field condition is sent to the farmer via mobile text messages. With thissystem Soil moisture, humidity and energy efficiency are managed. A system is proposed for intelligent agriculture monitoring system based on IOT technology.The main aim of this project is to help farmers to automate their farms by providing them with a Web App through which they can monitor the parametersof the field like Temperature, humidity etc. and control the equipment like watermotor and other devices remotely via internet without their actual presence in thefield.

Crops in the farms are many times devastated by the wild as well as domestic animals and low productivity of crops is one of the reasons for this. It is not possible to stay 24 hours in the farm to sentinel the crops. So to surmount this issue an automated perspicacious crop aegis system is proposed utilizing Internet of Things (IOT). The system consists of esp8266 (node MCU), soil moisture sensor, dihydrogen monoxide sensor, GPRS and GSM module, servo motor, dihydrogen monoxide pump, etc. to obtain the required output. As soon as any kineticism is detected the system will engender an alarm to be taken and the lights will glow up implemented at every corner of the farm. This will not harm any animal and the crops will stay forfended.

Chapter-2

LITERATURE SURVEY

S.No	Author and journal		Name of the topic	Features
	Naveenbalaji Gowthaman, V Nandhini IEEE		Advanced Crop Monitoring using Internet of Thingsbased Smart Intrusion & Prevention	❖ Agriculture is the backbone ofour country. To improve productivity in agricultural lands modern crop yielding methods have been used. ❖ A system by using the wirelesssensor networks to detect the intrusion of birds and animalsin the agricultural lands is discussed It is necessary to measure thecrop water requirements, temperature and humidity in agricultural lands which can
2	Alaa Adel Araby, Mohamed Ali Fahim, IEEE	2019	IOT Based Smart Crop Protection and Irrigation System	❖ Objective of this paper is to provide an automatic irrigationsystem thereby saving time, money & power of the farmer automatic environmental change detection

3	S Ayyasamy, S Eswaran, B Manikandan, S Nirmal kumar IEEE	2020	IoT based Agri Soil Maintenance Through Micro-Nutrients and Protection of Crops from Excess Water	❖ Smart irrigation with smart control and intelligent decisionmaking based on accurate realtime field data. ❖ Smart warehouse managementwhich includes temperature maintenance, humidity maintenance and theft detection
4	Muhammad Ayyaz, Ammad Udin, Ali nMansour	2019	Internet-of-Things (IoT)-Based Smart Agriculture	❖ Soil is stomach of plants, and its sampling is the first step of examination to obtain field-specific information, which is then further used to make various critical decisions at different stages. ❖ A fertilizer is a natural or chemical substance that can provide important nutrients for the growth and fertility of plants. ❖ Plants mainly need three key macronutrients: 1. nitrogen (N) for leaf growth; 2. phosphorus (P) for root, flowers, and fruit development; 3. potassium (K) for stem growth and water movement

5	Alaa Adel Araby, Mai Mohamed Abd Elhameed, Hassan Mostafa	2020	Smart IoT Monitoring System for Agriculture with Predictive Analysis	❖ Precision agriculture is a new concept in agriculture, it is defined as the farm management system using information technology to identify, analyze and manage the variability of fields to ensure profitability, sustainability, and protection of the environment. ❖ It is obvious that precision agriculture increases the efficiency that can be realized by understanding and dealing

- Farmers are to be present at farm for maintenance irrespective of the weather conditions.
- They must ensure that the crops are well irrigated, and the routine activities ofthe field must be monitored by them physically.
- To get good, cultivate or yield farmers need to stay in the field for longer timefor good yield
- Demand and supply are more if the field is to be monitored monotonously if itcovers vast area
- Anytime crops may prone to various calamities which leads to poor yield of thecrops

And these are some cons that is been faced by the farmers and which leads them to cultivate crops for less yield despite heavy work

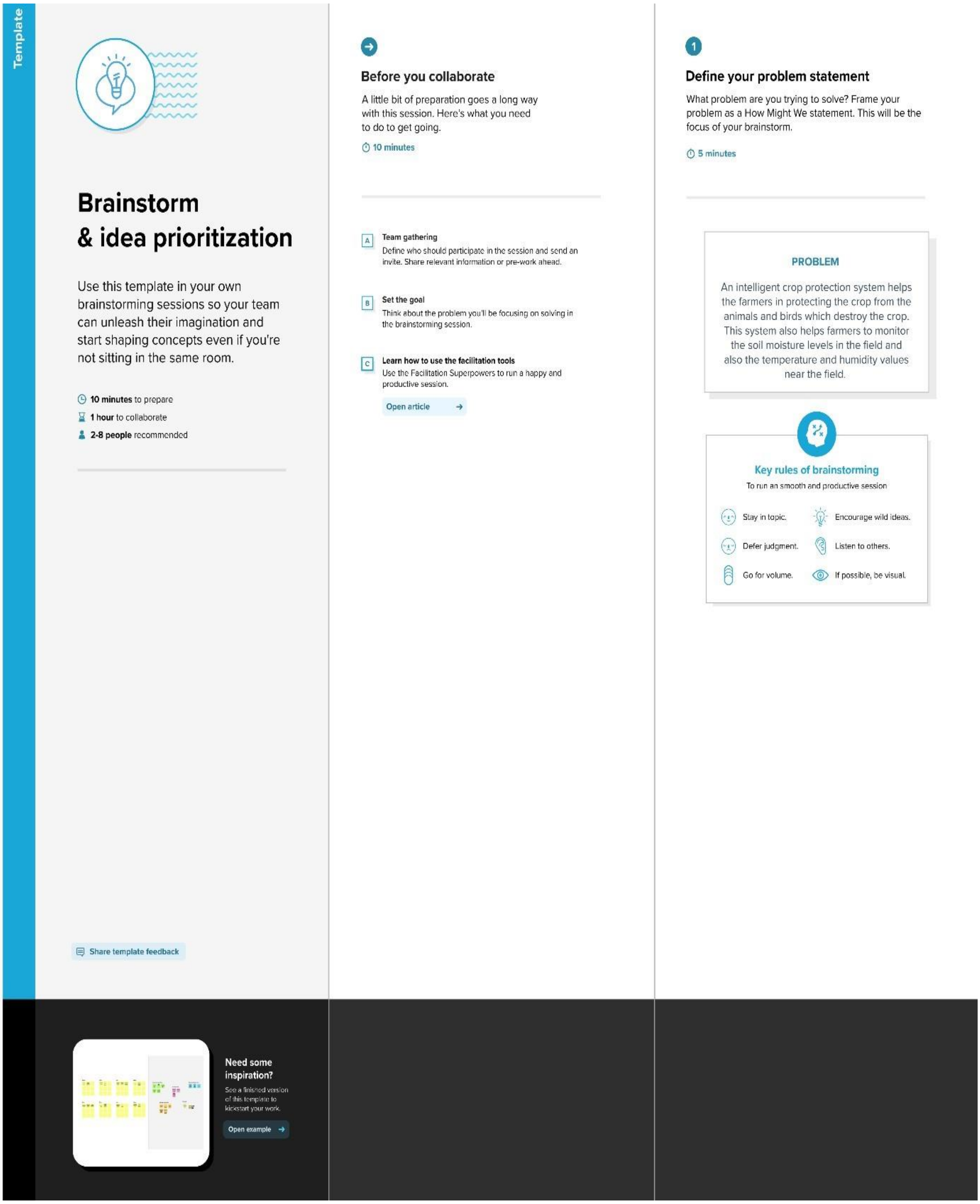
Chapter-3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:

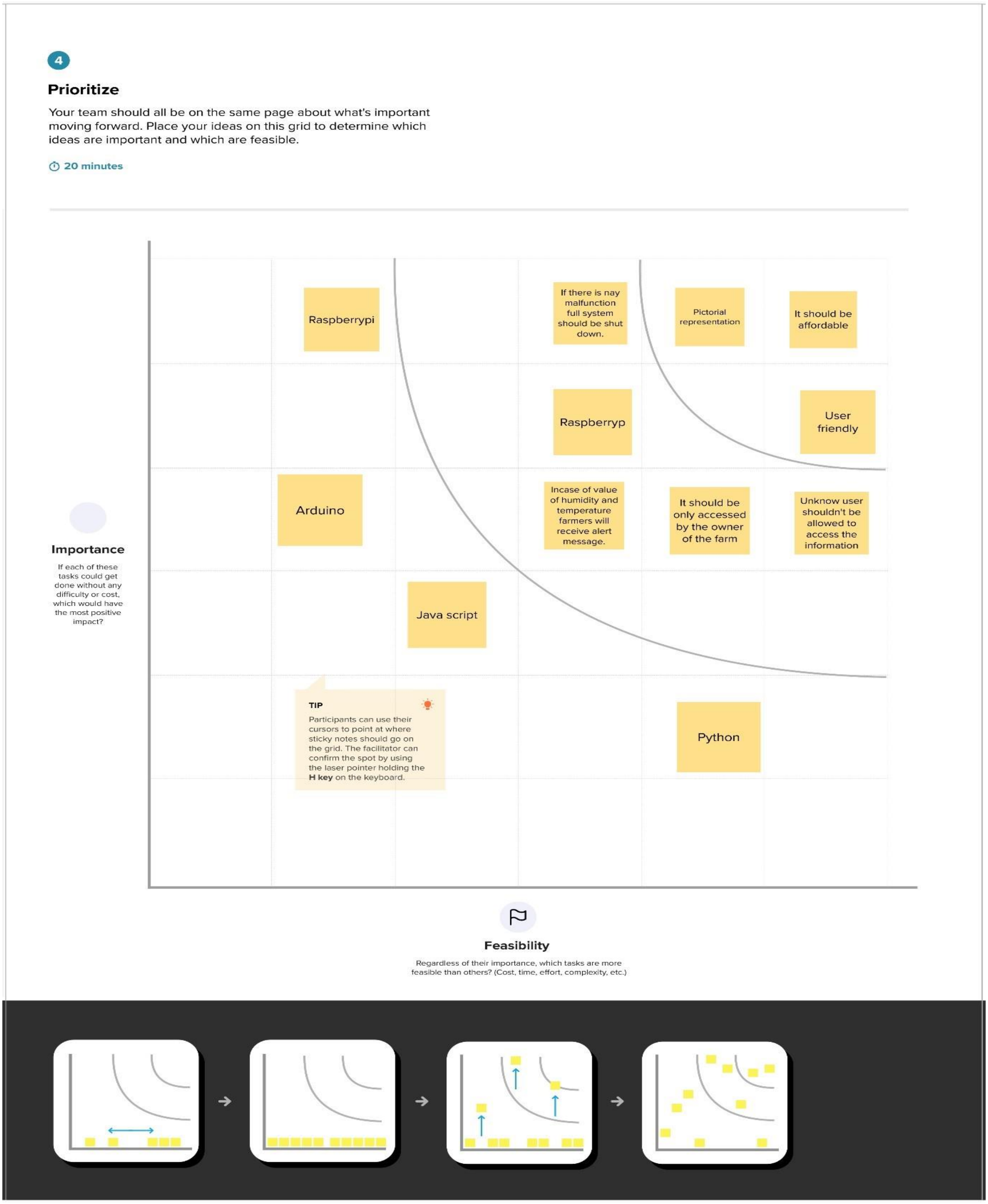
3.2 Ideation & Brainstorming:

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping

Step-3: Idea Prioritization



3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem tobe solved)	Farmers can protect their crops from animals and birds that might otherwise damage themwith the use of an intelligent crop protection system. This method aids farmers in keepingtrack of the temperature and humidity levels nearby the field, as well as the levels of soil moisture within the field. The mobile application can be used to control the field's motors and sprinklers.
2.	Idea / Solution description	**Main Solution:** ❖ Our project's primary goal is to create an intruder alert system for the farm in order to prevent animaland fire losses. ❖ Agricultural fences are a very successful technology for protectingwild animals. ❖ Utilize AGRVI's intelligent pest alarms to improve crop protection.
3.	Novelty / Uniqueness	With the use of GPS technology, this intelligentcrop protection system uses cloud DB data to save crop information and alerts the owner around-the-clock without their physical presence.

4.	Social Impact / CustomerSatisfaction	❖ Assists farmers in understanding key elements like water, topography, aspect,vegetation, and soil kinds. ❖ By integrating data along the supply chain, it also enables verification efforts,allowing manufacturing claims to be verified. ❖ Control of weeds and integrated management.
5.	Business Model (RevenueModel)	❖ Community based solution. ❖ Increase the proper products cost. ❖ Canvas a business model.
6.	Scalability of the Solution	Farmers can make significant profits with fully automated, limited resources thanks to smart agricultural systems. Animals and humans won'tbe harmed or injured by the developed system. Low cost fix, less need on electricity Simple fix that will work for the farming community.

3.4 Problem Solution fit:

1. CUSTOMER SEGMENT(S) CS

Farmer's ! Who's not near his field

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL

1)High adoption costs , security concerns.

2)Not aware of the implementation of IoT inagriculture.

5. AVAILABLE SOLUTIONS PLUSES & MINUSES AS

Monitor different parameters and mobile or web application make easily to farm thecrop field .

2. PROBLEMS / PAINS + ITS FREQUENCY PR

- It's difficult to monitor and control
- Ain't known if the applicationdoesn't work properly.

9. PROBLEM ROOT / CAUSE RC

1)If temperature ,PH level ,humidity & light intensity makes the serious cause for the environment.

2)Farmer affected by less productivity which will affect in their profit.

7. BEHAVIOR + ITS INTENSITY BE

Direct related: Tries to find a solution toprevent this problem

Indirect related: Located in rural where internet connectivity might not be strong enough to facilitate fast transmission speeds.

3. TRIGGERS TO ACT TR

Create opportunities to lift people out of poverty in developing nations. (Over60%)

4. EMOTIONS BEFORE / AFTER EM

BEFORE: Finances, Heavy work overload andconflict in relationship.

AFTER: It will easier to make more yield in

10. YOUR SOLUTION SL

"IoT based Smart crop protection systemfor agriculture" !!

It help farmers grow more food on lessland by protection crops from pests, diseases and weeds as well as raising productivity per hectare.

8. CHANNELS of BEHAVIOR CH

ONLINE: The Data send through applicationfor the farmers to know about the farms.

OFFLINE: The control action is taken bythe farmers to monitor the farms.

Chapter-4

REQUIREMENT ANALYSIS

4.1 Functional Requirement:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Safety of production	The Smart Protection System identifies this initiative as aiding farmers in preserving land. The IOT gadget is used to warn the farmer when birds visit the farm and an SD card module is used to store a specific sound to make the animals afraid.
FR-2	Real time monitoring	Farm crops are frequently destroyed by neighbourhood animals including buffalo, cows, goats, birds, etc. The farmer suffers significant losses as a result. Deforestation caused by overpopulation leads to a lack of food, water, and shelter in forested areas. Therefore, animal intrusion into residential areas is growing daily, endangering human life and property and leading to human-animal conflict.
FR-3	Eliminate man power	The device can be check the soil whether,it’s wet or dry after checking in the device can be sent the message to there respective owner. Alarm system has been set to avoid conflicts
FR-4	Fast communication	This system uses a motion sensor to detect wild animals approaching near the field and smoke sensor to detect the fire. In such a case the sensor signals the microcontroller to take action. The microcontroller now sounds an alarm to woo the animals away from the field as well as sends SMS to the farmer and makes call, so that farmer may know about the issue and come to the spot in case the animals don’t turn away by the alarm.
FR-5	Performance	Using IOT network the sensor sends an message to the user.
FR-6	Scalable Architecture	Justify the scalability of architecture.

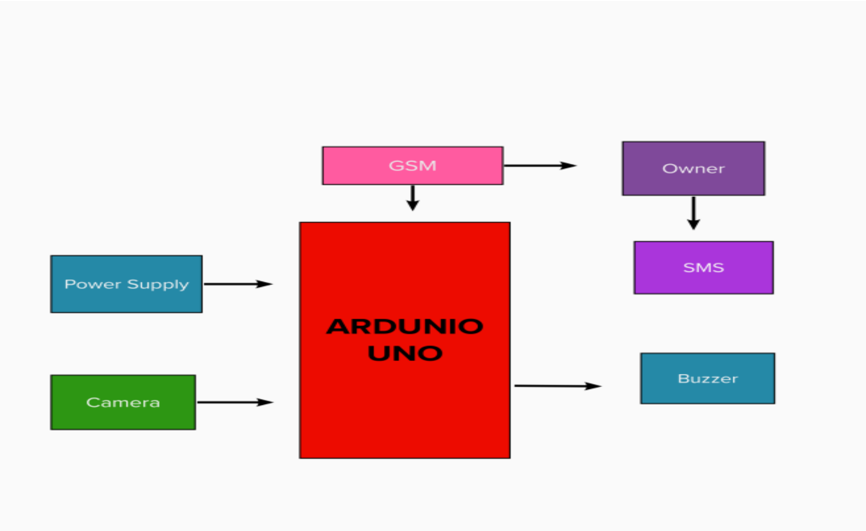
4.2 Non-Functional Requirement:

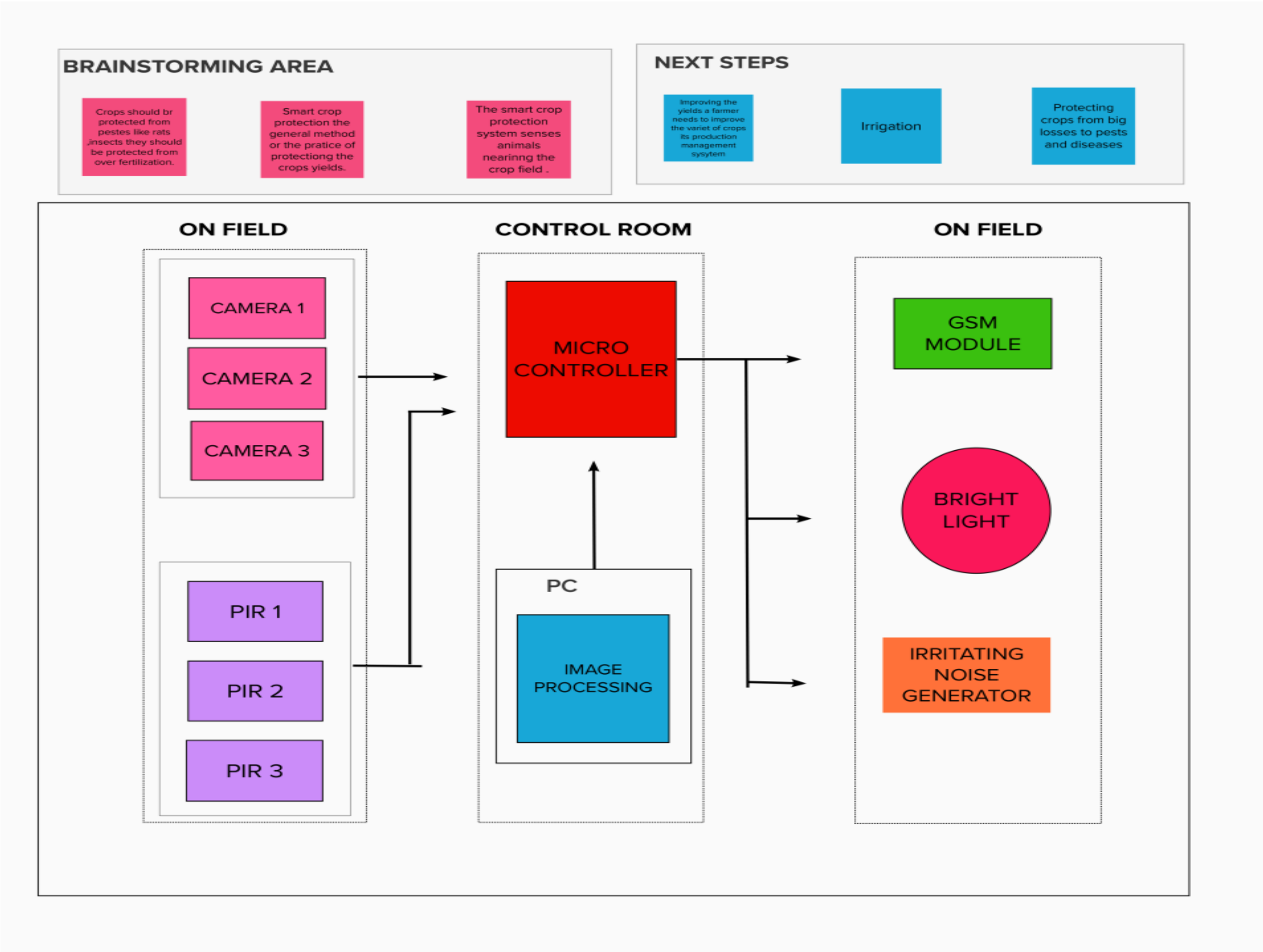
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Usability is a unique and significant perspective to examine user requirements, which may further enhance the design quality, according to IOT devices. Analysis of consumer product usability may help designers better understand users' prospective demands in gas leakage monitoring, behaviour, and experience in the design process where user experience is at the centre.
NFR-2	**Security**	It helps to prevent from material loss and human injuries
NFR-3	**Reliability**	Crop Protection System Using IOT to Prevent Bird and Wild Animal Attacks Using Arduino, a smart crop security device against wildlife Using Arduino, a smart crop protection system against fire and animals.
NFR-4	**Performance**	This device employs a motion sensor to find approaching wild animals close to the field and a smoke sensor to find a fire. The sensor instructs the microcontroller to operate in this situation.
NFR-5	**Availability**	This device employs a motion sensor to find approaching wild animals close to the field and a smoke sensor to find a fire. The sensor instructs the microcontroller to operate in this situation.

Chapter-5

PROJECT DESIGN

5.1 Data Flow Diagrams:

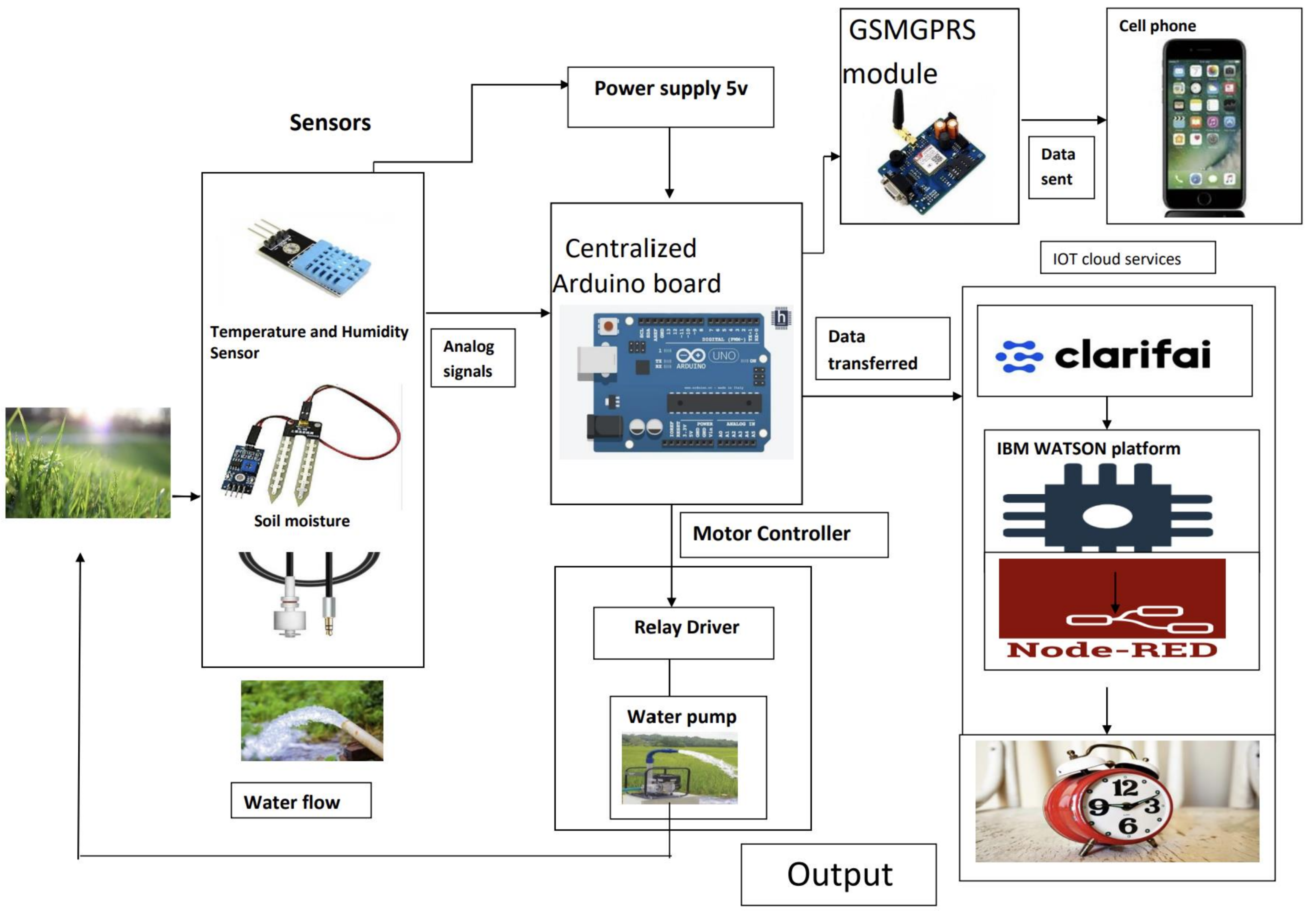




5.2 Solution & Technical Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.



Explanation for the Architecture Diagram:

- The device will detect the animals and birds using the Clarifai service.
- If any animal or bird is detected the image will be captured and stored in the IBM Cloud object storage.
- It also generates an alarm and avoid animals from destroying the crop.
- It also generates an alarm and avoid animals from destroying the crop.
- The image URL will be stored in the IBM Cloudant DB service.
- The device will also monitor the soil moisture levels, temperature, and humidity values and send them to the IBM IoT Platform.
- The image will be retrieved from Object storage and displayed in the web application.
- A web application is developed to visualize the soil moisture, temperature, and humidity values.
- Users can also control the motors through web applications

5.3 User Stories:

User type	Functional Requirement	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
User	Data collecting	USN-1	Smart farming based IOT	Sensing of Agriculture data and storing.	High	Sprint-4
		USN-2	As an user, I will inform the farmer to protect the crops.	I can inform the farmer.	Medium	Sprint-2
		USN-3	Extract data from source.	Management of data through expert and investigation method	High	Sprint-2
User 2	Login	USN-4	As an co-user, I cansend the alert message to the farmers.	I can alert farmers.	High	Sprint-1
Farmer	Login	USN-5	As a farmer, I will follow the route tothe crop which can avoid are detect animal intrusion.	I can reach the crops.	High	Sprint-2
Crop Protector		USN-6	A an crop protector.	In can protect the crop.	Medium	Sprint-2
Farmer	Login	USN-7	As a supervisior, I can Supervise the crop ar ensure the hygiene proces	I can manage all these process going good.	High	Spirit-1
Crop yielder	Register	USN-8	As a crop yielder,I can yield more crop.	I can register smart crop.	Medium	Spirit-3
Crop Monitor		USN-9	As a crop monitor,I check the quality of IOTdevice's quality.	I can check the IOT device.	Medium	Spirit-3

Chapter – 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1		US-1	Create the IBM Cloud services which are being used in this project.	6	High	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R
Sprint-1		US-2	Configure the IBM Cloud services which are being used in completing this project.	4	Medium	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R
Sprint-2		US-3	IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so create the IBM Watson IoT platform.	5	Medium	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R
Sprint-2		US-4	In order to connect the IoT device to the IBM cloud, create a device in the IBM Watson IoT platform and get the device credentials.	5	High	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R
Sprint-3		US-1	Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform.	10	High	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R
Sprint-3		US-2	Create a Node-RED service.	10	High	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R
Sprint-3		US-1	Develop a python script to publish random sensor data such as temperature, moisture, soil and humidity to the IBM IoT platform	7	High	Arshad Parvez G Lavanya Kishore Kumar K Mathuprakas R
Sprint-3		US-2	After developing python code, commands are received just print the statements which represent the control of the devices.	5	Medium	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R

Sprint-4		US-3	Publish Data to The IBM Cloud	8	High	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R
Sprint-4		US-1	Create Web UI in Node- Red	10	High	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R
Sprint-4		US-2	Configure the Node-RED flow to receive data from the IBM IoT platform and also use Cloudant DB nodes to store the received sensor data in the cloudant DB	10	High	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.2 Sprint Delivery Schedule:

TITLE	DESCRIPTION	DATE
Literature Survey on The Selected Project and Information Gathering	A Literature Survey is a compilation summary of research done previously in the given topic. Literature survey can be taken from books, research paper online or from any source.	19 September 2022
Prepare Empathy Map	Empathy Map is a visualization tool which can be used to get a better insight of the customer	19 September 2022
Ideation-Brainstorming	Brainstorming is a group problem solving session where ideas are shared, discussed and organized among the team members.	19 September 2022
Define Problem Statement	A Problem Statement is a concise description of the problem or issues a project seeks to address. The problem statement identifies the current state, the desired future state and any gaps between the two.	19 September 2022
Problem Solution Fit	This helps us to understand the thoughts of the customer their likes, behaviour, emotions etc.	12 October 2022
Proposed Solution	Proposed solution shows the current solution and it helps is going towards the desired result until it is achieved.	12 October 2022
Solution Architecture	Solution Architecture is a very complex process I.e it has a lot of sub-processes and branches. It helps in understanding the components and features to complete our project.	12 October 2022
Customer Journey	It helps us to analyse from the perspective of a customer, who uses our project.	15 October 2022
Functional Requirement	Here functional and nonfunctional requirements are briefed. It has specific features like usability, security, reliability, performance, availability and scalability.	15 October 2022
Data Flow Diagrams	Data Flow Diagram is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement.	15 October 2022
Technology Architecture	Technology Architecture is a more well defined version of solution architecture. It helps us analyze and understand various technologies that needs to be implemented in the project.	15 October 2022
Prepare Milestone & Activity List	It helps us to understand and evaluate our own progress and accuracy so far.	29 October 2022

Spring Delivery Plan	Sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved.	In Progress

Chapter-7

CODING & SOLUTIONING

7.1 Feature 1: Coding for Animals or pests enter into the field

```
 from pygame import
mixer class SoundPlayer:
      def ___init (self, sound_file):
          mixer.init(44100, -16, 2, 2048)
          self.sound
          =mixer.Sound(sound_file)
     def play(self):
          self.sound.pla
          y()
     import
     time
     class
     FPS:
       def ___init (self):
          self.frame_count
          = 0
          self.elapsed_time
       = 0 def start(self):
          self.start_time =
       time.time() def stop(self):
           self.stop_time =
           time.time()
           self.frame_count += 1
           self.elapsed_time += (self.stop_time-self.start_time)
       def count(self):
           return
     self.frame_count def
     elapsed(self):
          return
     self.elapsed_time def
     fps(self):
       if
          self.elapsed_time=
          =0:return 0
        else:
            return self.frame_count/self.elapsed_time
```

7.2 Feature 2: coding for moisture level checking

```
Import RPi.GPIO as GPIO
import time
channel=21
GPIO.setmode(GPIO.BCM)
GPIO.setup(channel,GPIO.IN)
def callback(channel):
        if GPIO.input(channel):
                print(“no water detected”)
        else:
                print(“water detected”)
GPIO.add_event_detect(channel,GPIO.BOTH,bouncetime=300)
GPIO.add_event_callback(channel,callback)
while True:
        time.sleep(1)
```

7.3 **Feature 3:** Detect The PH Level of Crops

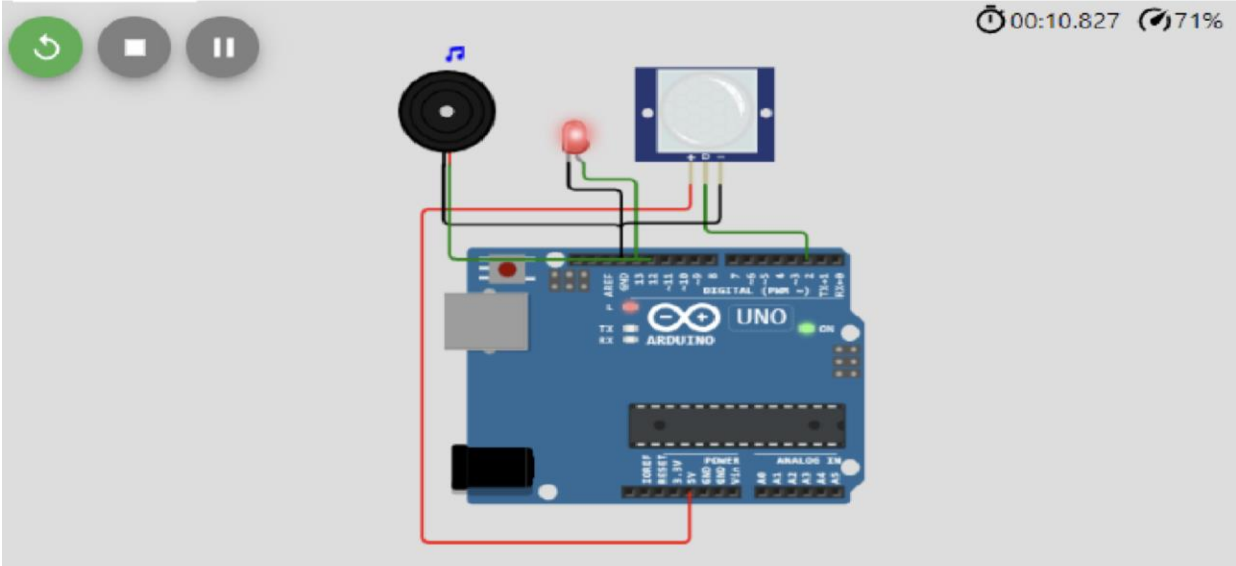
```
import io # used to create file streams

import fcntl # used to access I2C parameters like addresses

 import time # used for sleep delay and timestamps

class Ezo:

long_timeout = 1.5 # the timeout needed to query readings and

                    #calibrations

short_timeout = .5 # timeout for regular commands

default_bus = 1 # the default bus for I2C on the newer Raspberry Pis,

                  # certain older boards use bus 0

default_address = 99 # the default address for the pH sensor

def   init   (self, address=default_address, bus=default_bus):

# open two file streams, one for reading and one for writing

 # the specific I2C channel is selected with bus

# it is usually 1, except for older revisions where its 0

 # wb and rb indicate binary read and write

self.file_read = io.open("/dev/i2c-" + str(bus), "rb", buffering=0)

self.file_write = io.open("/dev/i2c-" + str(bus), "wb", buffering=0)


# initializes I2C to either a user specified or default address

 self.set_i2c_address(address)

def set_i2c_address(self, addr):

# set the I2C communications to the slave specified by the address

# The commands for I2C dev using the ioctl functions are specified in

 # the i2c-dev.h file from i2c-tools
```

```
        I2C_SLAVE = 0x703
        fcntl.ioctl(self.file_read, I2C_SLAVE, addr)
        fcntl.ioctl(self.file_write, I2C_SLAVE, addr)


        def write(self, string):


        # appends the null character and sends the string over I2C
        string += "\00"
        self.file_write.write(bytes(string, 'UTF-8'))


    def read(self, num_of_bytes=31):
        # reads a specified number of bytes from I2C,
         # then parses and displays the result
        res = self.file_read.read(num_of_bytes) # read from the board
        # remove the null characters to get the response
        response = [x for x in res if x != '\x00']
        if response[0] == 1: # if the response isnt an error
            # change MSB to 0 for all received characters except the first
            # and get a list of characters
            char_list = [chr(x & ~0x80) for x in list(response[1:])]
            # NOTE: having to change the MSB to 0 is a glitch in the
            # raspberry pi, and you shouldn't have to do this!
            # convert the char list to a string and returns it
             #return "Command succeeded " +
           return''.join(char_list)
        else:
```

```
            return "Error " + str(response[0])


    def query(self, string):
        # write a command to the board, wait the correct timeout,
        # and read the response
        self.write(string)


    # the read and calibration commands require a longer timeout
    if((string.upper().startswith("R")) or
        (string.upper().startswith("CAL"))):
        time.sleep(self.long_timeout)
    elif((string.upper().startswith("SLEEP"))):
        return "sleep mode"
    else:
        time.sleep(self.short_timeout)
    return self.read()
    def close(self):
        self.file_read.close()
        self.file_write.close()


        #ph = Ezo()
        #phvalue = ph.query('R')
         #ph1 = str(phvalue)
         #ph2 = round(phvalue)
         #print (ph.query('R'))
        #print (round(ph.query('R'),2))
```

Chapter-8

RESULTS




```
Motion detected!
Camera activated!
Pictures taken!
Buzzer activated!
```

Chapter – 9

ADVANTAGES AND DISADVANTAGES

Advantages:

- Farms can be monitored and controlled remotely.
- Increase in convenience to farmers.
- Less Manpower.
- Better standards of living.

Disadvantages:

- Lack of internet/connectivity issues.
- Added cost of internet and internet gateway infrastructure.
- Farmers wanted to adapt the use of WebApp.

Chapter – 10

CONCLUSION

This system focuses on developing devices and tool to manage, display andalert the users using the advantages of a wireless sensor network system. It aims at making agriculture smart using automation and IoT. The cloud computing devices are used at the end of the system that can create a whole computing system from sensors to tools that observe data from agriculture field. It proposes a novel methodology for smart farming by including a smart sensing system and smart irrigator system through wireless communication technology. Thus, the objective of the project to implement an IoT system in order to help farmers to control and monitor their farms has been implemented successfully.

Chapter – 11

FUTURE SCOPE

Agriculture domains encounters with many challenges starting from soil parameters, seed sowing, crop growth and its quality, weed handling, disease management till harvesting and storing crop. Artificial intelligence driven techniques along with other available tools and automation can address these challenges and proven the revolution in agriculture. Most popular AI application in agriculture is use of Robot and Drones, they perform almost all task like humans even at a faster rate with accuracy. From literature review it is clear that precision farming is probable by integrating sensors, cameras, data analytics, GPS and remote sensing. Image recognitions software's, IoT sensors can be used for disease recognition at primary stages and hence crop health can be supervised which increases superior quality production with minimum loss. Table 1 demonstrate the various applications in view of Smart Agriculture for improved evolution as well as superiority. Still there are several challenges associated with AI and IoT application in smart agriculture which is the promising future to be explored area for researchers. Some of major challenges are: • Awareness issues • Hardware implementation challenges • Cost of software and hardware • Network management • Energy management • Privacy issues • Security challenges • Interoperability of systems with the induction of Computer vision, Deep learning, Big data also agriculture sector has influenced a lot. Researchers can integrate IoT sensors along with smart systems and computational optimization algorithms to overcome the limitations/shortcomings. Smart Agriculture has a budding potential towards productivity, precision, optimization, adaptive resource management and intelligent food traceability. It will contribute to environment also in terms of efficient use of water, prevent disease contamination and precise use of pesticides.

Chapter – 12

APPENDIX

```
import random
import ibmiotf.device
from time import sleep
import sys
#IBM Watson Device Credentials.
organization = "op701j"
deviceType = "Lokesh"
deviceId = "Lokesh89"
authMethod = "token"
authToken = "1223334444"
def myCommandCallback(cmd):
 print("Command received: %s" % cmd.data['command'])
 status=cmd.data['command']
 if status=="sprinkler_on":
    print ("sprinkler is ON")
 else :
    print ("sprinkler is OFF")
 #print(cmd)




try:
 deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"auth-method": authMethod, "auth-token": authToken}
 deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
sys.exit()
#Connecting to IBM watson.
deviceCli.connect()
while True:
#Getting values from sensors.
 temp_sensor = round( random.uniform(0,80),2)
 PH_sensor = round(random.uniform(1,14),3)
```

```
 camera = ["Detected","Not Detected","Not Detected","Not Detected","Not
Detected","Not Detected",]
 camera_reading = random.choice(camera)
 flame = ["Detected","Not Detected","Not Detected","Not Detected","Not
Detected","Not Detected",]
 flame_reading = random.choice(flame)
 moist_level = round(random.uniform(0,100),2)
 water_level = round(random.uniform(0,30),2)


#storing the sensor data to send in json format to cloud.


 temp_data = { 'Temperature' : temp_sensor }
 PH_data = { 'PH Level' : PH_sensor }
 camera_data = { 'Animal attack' : camera_reading}
 flame_data = { 'Flame' : flame_reading }
 moist_data = { 'Moisture Level' : moist_level}
 water_data = { 'Water Level' : water_level}


# publishing Sensor data to IBM Watson for every 5-10 seconds.
 success = deviceCli.publishEvent("Temperature sensor", "json", temp_data,
qos=0)
 sleep(1)
 if success:
    print (" ...........................publish ok............................ ")
 print ("Published Temperature = %s C" % temp_sensor, "to IBM Watson")


 success = deviceCli.publishEvent("PH sensor", "json", PH_data, qos=0)
 sleep(1)
 if success:
    print ("Published PH Level = %s" % PH_sensor, "to IBM Watson")


 success = deviceCli.publishEvent("camera", "json", camera_data, qos=0)
 sleep(1)
 if success:
    print ("Published Animal attack %s " % camera_reading, "to IBM
Watson")
 success = deviceCli.publishEvent("Flame sensor", "json", flame_data,
qos=0)
 sleep(1)
 if success:
```

```
    print ("Published Flame %s " % flame_reading, "to IBM Watson")


 success = deviceCli.publishEvent("Moisture sensor", "json", moist_data,
qos=0)
 sleep(1)
 if success:
     print ("Published Moisture Level = %s " % moist_level, "to IBM
Watson")


 success = deviceCli.publishEvent("Water sensor", "json", water_data,
qos=0)
 sleep(1)
 if success:
    print ("Published Water Level = %s cm" % water_level, "to IBM Watson")
 print ("")
 #Automation to control sprinklers by present temperature an to send alert
message to IBM Watson.


 if (temp_sensor > 35):
    print("sprinkler-1 is ON")
 success = deviceCli.publishEvent("Alert1", "json",{ 'alert1' :
"Temperature(%s) is high, sprinkerlers are turned ON" %temp_sensor }
, qos=0)
 sleep(1)
 if success:
    print( 'Published alert1 : ', "Temperature(%s) is high, sprinkerlers are
turned ON" %temp_sensor,"to IBM Watson")
 print("")
else:
 print("sprinkler-1 is OFF")
 print("")


 #To send alert message if farmer uses the unsafe fertilizer to crops.


 if (PH_sensor > 7.5 or PH_sensor < 5.5):
    success = deviceCli.publishEvent("Alert2", "json",{ 'alert2' : "Fertilizer PH
level(%s) is not safe,use other fertilizer" %PH_sensor } ,
qos=0)
 sleep(1)
 if success:
```

```
    print('Published alert2 : ' , "Fertilizer PH level(%s) is not safe,use other fertilizer" %PH_sensor,"to IBM Watson")
 print("")


 #To send alert message to farmer that animal attack on crops.


 if (camera_reading == "Detected"):
    success = deviceCli.publishEvent("Alert3", "json", { 'alert3' : "Animal attack on crops detected" }, qos=0)
 sleep(1)
 if success:
    print('Published alert3 : ' , "Animal attack on crops detected","to IBM Watson","to IBM Watson")
 print("")
 #To send alert message if flame detected on crop land and turn ON the splinkers to take immediate action.


 if (flame_reading == "Detected"):
    print("sprinkler-2 is ON")
 success = deviceCli.publishEvent("Alert4", "json", { 'alert4' : "Flame is detected crops are in danger,sprinklers turned ON" }, qos=0)
 sleep(1)
 if success:
    print( 'Published alert4 : ' , "Flame is detected crops are in danger,sprinklers turned ON","to IBM Watson")


 #To send alert message if Moisture level is LOW and to Turn ON Motor-1 for irrigation.
 if (moist_level < 20):
    print("Motor-1 is ON")
 success = deviceCli.publishEvent("Alert5", "json", { 'alert5' : "Moisture level(%s) is low, Irrigation started" %moist_level }, qos=0)
 sleep(1)
 if success:
    print('Published alert5 : ' , "Moisture level(%s) is low, Irrigation started" %moist_level,"to IBM Watson" )
 print("")
 #To send alert message if Water level is HIGH and to Turn ON Motor-2 to take water out.
 if (water_level > 20):
    print("Motor-2 is ON")
```

```
 success = deviceCli.publishEvent("Alert6", "json", { 'alert6' : "Water
level(%s) is high, so motor is ON to take water out "
%water_level }, qos=0)
 sleep(1)
 if success:
    print('Published alert6 : ' , "water level(%s) is high, so motor is ON to take
water out " %water_level,"to IBM Watson" )
    print("")
 #command recived by farmer
deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```