IoT Based Smart Crop Protection System for Agriculture

TEAM ID-PNT2022TMID15778

1.INTRODUCTION

1.1 PROJECT OVERVIEW

- 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 .
- 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 application.

1.2 PURPOSE

The purpose is to grant monitoring device for crop safety to animal outbreaks and environment circumstances. This supports to preserve stretch and cash by dipping the physical exertion, else obligatory if the cultivators themselves have to afford guard for their crops with their endless physical administration. Wildlife regularly wreck eminence crops, because of which annual manufacturing of vegetation reduces inflicting monetary victims to cultivators. Agriculturalist suicide is huge bother due to less harvest. This low harvest is duet the circumstance of two most significant purposes i.e. Crop wrecked via untamed animals and Crop wrecked by meteorological conditions . The ranchers will treasure these SMS containing location. The prime thing of this task is to furnish a greatreply to this distress . Each time either the wild animal or species are identified through PIR sensor which stimulates the web camera and givesrise to alert the buzzer in the locality, associates to the farmer direct to the cloud . When the moisture content is inferior to a terrifying level the sensor planted makes the water pumps to turn on . This ensures the complete safety of crops from animals also as from the weather conditions thus prevent the farmers .

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

IOT tendencies are often utilized in smart farming to boost the standard of agriculture . Farming the pillar of supports our country to the general commercial development. But our productivity is extremely low as associated to world standards. People from rural areas drift to an urban area for other worthwhile trades and they can't concentrate on agriculture .There are many disadvantages of the current traditional agricultural methods namely costlier and manual monitoring of the agriculture field.

Specifically, small-scale smart irrigation systems are utilized to provide the solution for dissimilar variety of plants in spite of getting the solution for moisture related issues Weather conditions like temperature, humidity and moisture are difficult to check manually frequently. Farmer suicide is turning into big problem due to low productiveness amongst farms . This low productiveness is due to the fact of two main reasons, Crop ruined bymeans of untamed weather conditions untamed animal attacks, small types of species, insects, some hazardous snakes and weather circumstances.

2.2 REFERENCES

- Krunal Mahajan1, Riya Parate2, Ekta Zade3, Shubham Khante4, Shishir Bagal5,” REVIEW PAPER ON SMART CROP PROTECTION SYSTEM”, International Research Journal of Engineering and Technology (IRJET),Volume: 08, issue 02 Feb 2021.
- Dr.M. Chandra, Mohan Reddy, Keerthi Raju KamakshiKodi, BabithaAnapalliMounikaPulla, “SMART CROP PROTECTION SYSTEM FROM LIVING OBJECTS AND FIRE USING ARDUINO”, Science, Technology and Development, Volume IX Issue IX, pg.no 261- 265, Sept 2020.
- Anjana, Sowmya, Charan Kumar, Monisha, Sahana, “Review on IoT in Agricultural Crop Protection and Power Generation”, International ResearchJournal of Engineering and Technology (IRJET) , Volume 06, Issue 11 ,Nov

2019.

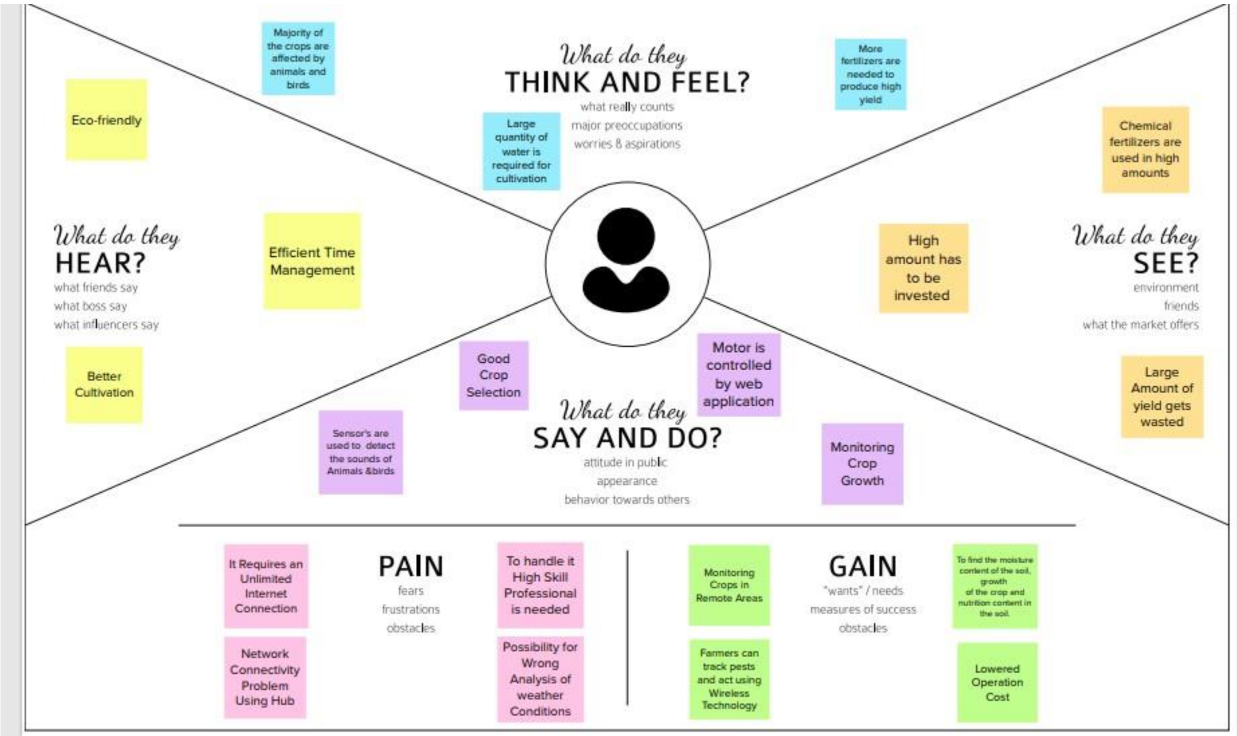
- G. NaveenBalaji, V. Nandhini, S. Mithra, N. Priya, R. Naveena, “IOT based smart crop monitoring in farmland”, Imperial Journal of Interdisciplinary Research (IJIR), Volume 04, Issue 01, Nov 2018.
- P.Rekha, T.Saranya, P.Preethi, L.Saraswathi, G.Shobana, “Smart AGRO Using ARDUINO and GSM”, International Journal of Emerging Technologiesin Engineering Research (IJETER) Volume 5, Issue 3, March 2017

2.3 PROBLEM STATEMENT DEFINITION

Within the existing system, electrical fencing is used to give up untamed animal assaults on agricultural vegetation which leads to the death of animals . The fundamental objective is to provide a fantastic answer to this problem, so that losses incurred will be minimized and farmers will have an accurate crop yield . This low productivity is because of the fact of two most important motives i.e. Crop destroyed via untamed animals and Cropdamaged by using nature object.The main objective of this assignment is to furnish a fantastic answer to this trouble, as a result with the purpose ofthe economic losses incurred through the support of our farmers are minimized to get truthful crop yield .This ensures complete security of vegetation from animals and defending the farmers loss. In the proposed system Raspberry Pi, PIR sensor, web camera, ultrasonic sensor, LDR sensor, temperature sensor, humidity sensor, moisture sensor, buzzer and monitor are used . This field of this effort remains towards withdraw to monitor the system for crop security conflicting to subconscious occurrences and meteorological conditions When the moisture content is below a critical level which is determined by the sensor planted in the fields, as the system is automated the water pumps are switched on . This ensures complete safety of crops from animals also as from the weather conditions thus prevent the farmers loss.

3.IDEATION & PROPOSED SOLUTION

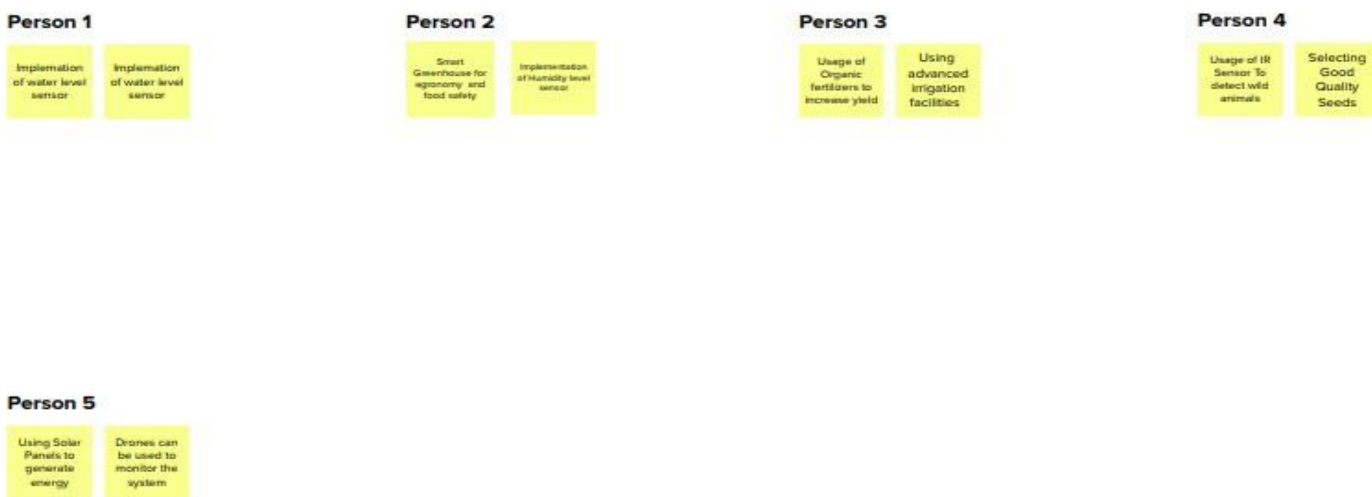
3.1 EMPATHY MAP CANVAS



Agriculture is the backbone of our country that contributes to 45% of the GDP that is responsible for the enhancement of country's economy. This IOT based Crop Protection System aims on building an integrated module for improving the efficiency of the present agricultural modules. A smart way of automating farming process can be called as Smart Agriculture. Precision agriculture is one of the most famous applications of loT in the agricultural sector and numerous organizations are leveraging this technique around the world. By implying an automated system, it possible to eliminate threats to the crops by reducing the human intervention.

The major emphasize will be on providing favourable atmosphere for plants. These agricultural automated systems will help in managing and maintain safe environment especially the agricultural areas. Environment real time. Monitoring is an important factor in smart farming. Graphical User Interface based software will be provided to control the hardware system and the system will be entirely isolated environment, equipped with sensors like temperature sensor, humidity sensor. The I controllers will be managed by a master station which will communicate with the human interactive software. This IOT based system will provide smart interface to the farmers and can increase the level of productionthan the current scenario.

3.2 IDEATION & BRAINSTORMING



- Implementation of water level sensor
- Using Solar panels to generate energy
- Usage of organic fertilizers to increase yield
- Usage of IR sensors to detect wild animals
- Selecting good quality seeds

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Protecting crops from insects , animals and other factors using pest sprayer , sound system and automatic drip irrigation.
2.	Idea / Solution description	Using moisture meter ,automatic sprayer of pesticide ,automatic DC motor and sensors are placed for protect crops.
3.	Novelty / Uniqueness	Water stagnation and scarcity is maintained every movement in field and growth of plants are monitored with mobile phone.
4.	Social Impact / Customer Satisfaction	Improved and high yield crops are obtained . Farmers work is reduced with automation.
5.	Business Model (Revenue Model)	This makes agriculture easier and profit is attained more by using this technique.
6.	Scalability of the Solution	This solution will gives high performance for proper maintenance.

3.4 PROBLEM SOLUTION FIT

Focus on J&P, tap into BE

1. CUSTOMER SEGMENT(S)

1. Farmers who need improved yield with smart automation will use this technique.
2. Gardeners also make this choice to improve their farm

6. CUSTOMER

What constraints prevent your customers from taking action or limit their choices

1. Pest control over the internal process.
2. Agricultural sector lack information of high adoption in IOT .
3. For security implementation of automation ,cost are not satisfied by farmers

5. AVAILABLE SOLUTIONS

or need to get the job done? What have they tried in the past? What pros & cons dothese solutions have? i.e. pen and paper is an alternative

1. Ask for customer needs and preferences
2. Offer a solution.
3. Understand the needs of farmer.
4. Pros:
   Wide spread to all.
   Increased profit.
5. Cons:

2. JOBS-TO-BE-DONE / PROBLEMS

Which jobs-to-be-done (or problems) do you address for your

Jobs to be done

1. Setting the apparatus and maintaining.
2. Proper monitoring for energy resource.

Problems

1. Environment and social impact of automation in agriculture- This cause reduction of human empowerment.
2. Distribution- Hard to reach in remote villages.
3. Cost – Setting the system in low budget is difficult.

9. PROBLEM ROOT CAUSE

What is the real reason that this problem exists? What is the back story behind the

1. Analyzing and giving solution.
2. The most common mistake people makes when equipment error or human error is to be identified.

7. BEHAVIOUR

i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering

1. Identify the troubles.
2. Understand the problems arising.
3. Make suitable choice of solutions.
4. Implement in field.
5. Monitor continuously.

Identify strong TR & EM

3. TRIGGERS

What triggers customers to act? i.e. seeing their neighbour installingsolar panels, reading about a

1. Through advertisements customers are triggered in automation.
2. Automation in agriculture are influenced by cinema ,government programs and by social platforms.

10. YOUR SOLUTION

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill inthe canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

1. Environment and social impact of automation in agriculture – make profit by innovative agriculture in smart way.
2. Distribution – make awareness in rural areas and make wider.
3. Cost – use cooling systems,high quality

8. CHANNELS of BEHAVIOUR

Extract online & offline CH of BE

8.1 ONLINE

What kind of actions do customers take online? Extract online channels from #7

This article highlights the potential of wireless sensors and IOT in agriculture, as well as challenges expected to be faced when integrating this technology

8.2 OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

1. This project will provide protection from animals

4. EMOTIONS: BEFORE / AFTER

How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

4. EMOTIONS: BEFORE / AFTER

How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Before

1. Crops were severely affected by extreme heat, heavy rainfall,animal grazing and other factors.

After

1. By this method,plants are protected from all factors that affect plants.

offline? Extract offline channels from #7 and use them for customer development.

1. This project will provide protection from animals through sound system.
2. Kills insects through automatic spray system.
3. We protect crops from excessive heat through bogie system.
4. Crop yield can be increased by monitoring

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

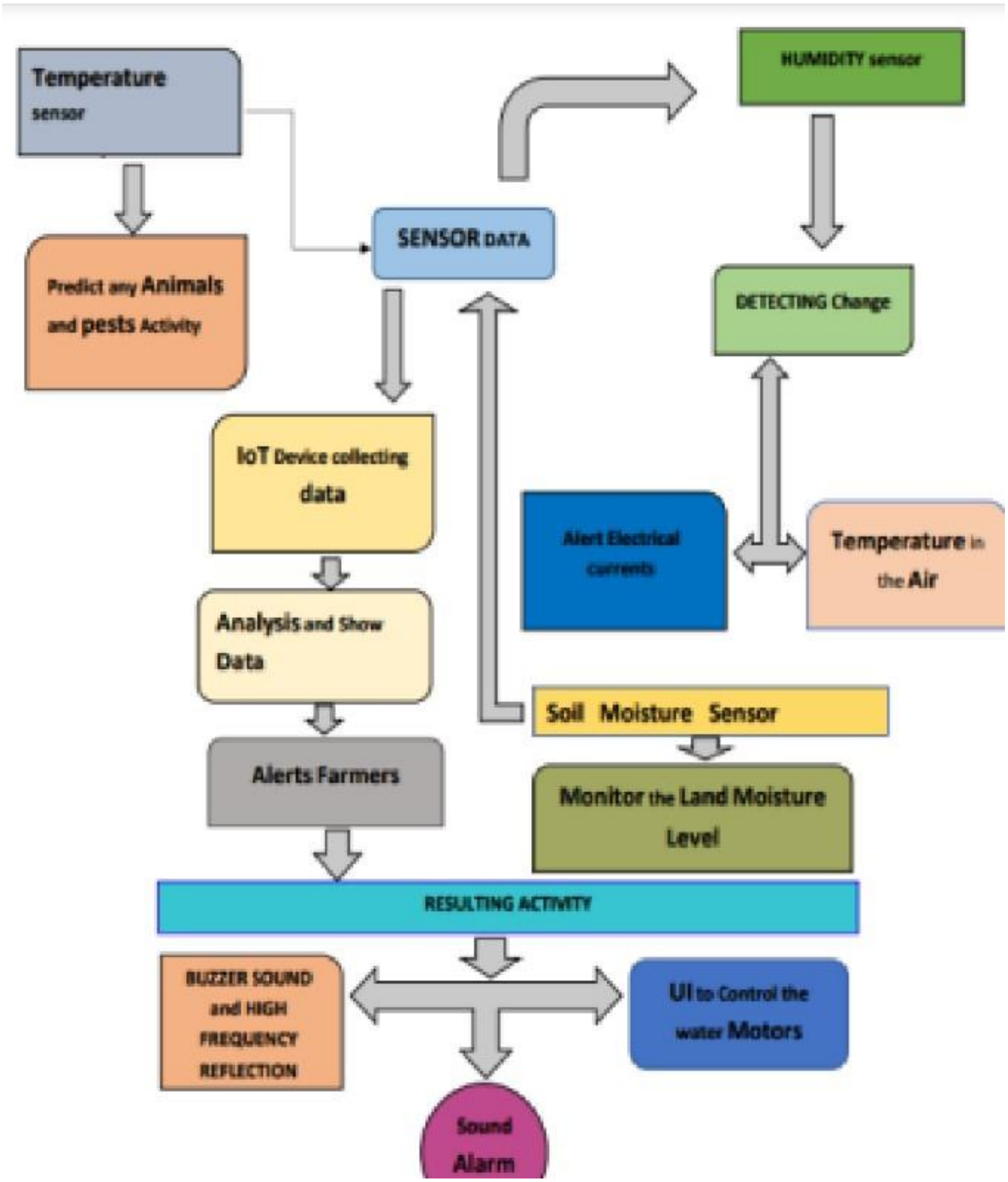
FR-NO	FUNCTIONAL REQUIREMENTS	SUB-REQUIREMENTS
FR-1	Fertilizing frame service	Documentation requirements and assisting information
FR-2	Economical service	Assisting information
FR-3	Technology assessment service	Selecting fertilizing features
FR-4	Feature assessment service	Updated technical information and machinery selection
FR-5	Information acquisition service	Assisting information about fertilizing rules
FR-6	Farm and field customizing service	Potential data acquisition service
FR-7	Field inspection	Spatial field information
FR-8	Field observation service	Analysed risks
FR-9	Assisting remote controlling	Inspecting and controlling fertilizing task
FR-10	Assisting "operational performance service"	Economical analysis of current technology

4.2 NON-FUNCTIONAL REQUIREMENT

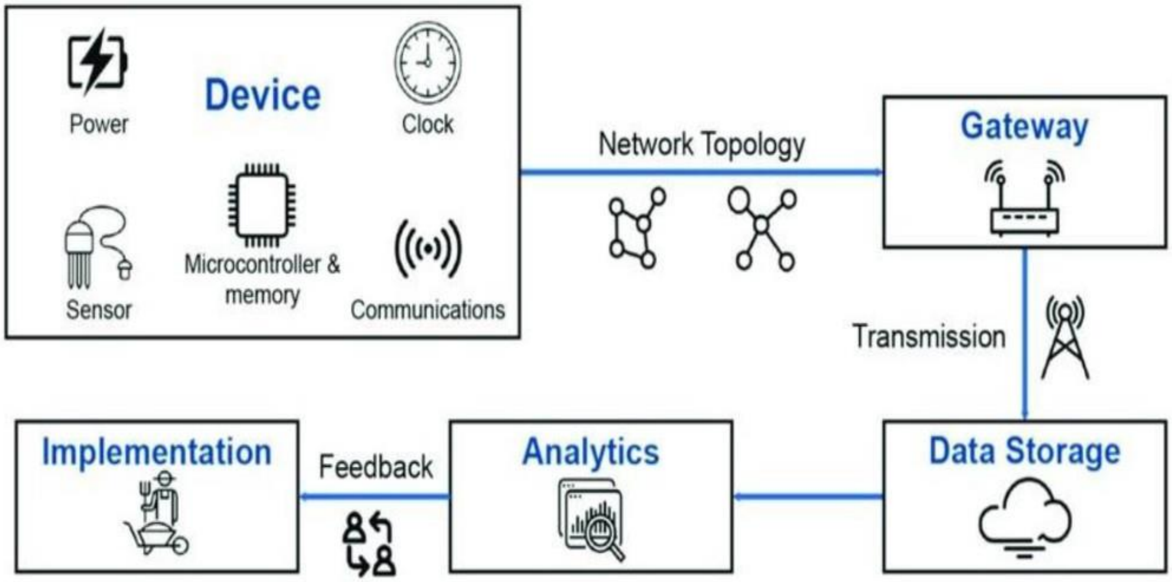
NRF.NO	NON FUNCTIONAL REQUIREMENTS	DESCRIPTION
NRF-1	Usability	To use new technologies and increase the quantity and quality
NRF-2	Security	Protect the field from animals.
NRF-3	Reliability	Increasing the demand for food with minimum resources
NRF-4	Performance	Maintain good yield and provide sustainable quantity
NRF-5	Availability	Agricultural fences are quite an effective wild animal protection
NRF-6	Scalability	The develop system will not harmful and injurious to animals as well as human beings.

5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS



5.2 SOLUTIONS & TECHNICAL ARCHITECTURE



a

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Farmer)	Maintaining Fields	USN-1	As a user, I can monitor the growth of crops and protect the crops against animals	I can maintain the fields with less labor	High	Sprint-1
	Analyzing Problems	USN-2	As a user, I collect the required information about the problems on agriculture fields	I can ask my field owner directly.	Low	Sprint-2
		USN-3	As a user, I can monitor the moisture level in soil and solve the problems by using Smart IOT System	I can take remedial action immediately	High	Sprint-1
Project Designers	Identifying the problem and provide solutions	USN-4	As a user, I can sense the water level and flame in the field using sensor and monitor using IOT	I can perform this actions via IoT.	Medium	Sprint-1
		USN-5	As a user, I can make services for Irrigation, pesticides, Fertilization, and Soil preparation	I can solve this problem using IOT	High	Sprint-1
			As a user, I can monitor the field against animal attacks using a camera interface module and appropriate actions can be taken	I can monitor the field continuously.	Medium	Sprint-2
Customer (Field Maintainer)	Problem solutions	USN-6	As a user, areas can be monitored from a remote place	Checking Process	Medium	Sprint-3
	Application	USN-7	As a user, I can respond to the problems in the fields immediately	Continuous monitoring and remedial actions.	Medium	Sprint-3
	Final Process	USN-8	This proposed smart IOT-based crop protection device is found to be cost-effective and efficient	I can take necessary action if required.	Medium	Sprint-4

6. PROJECT PLANNING & SCHEDULING

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	Jaya swetha, Kaviya sri, Lakshmipriya, Madhubala
Sprint-1		US-2	Configure the IBM Cloud services which are being used in completing this project.	4	Medium	Jaya swetha, Kaviya sri, Lakshmipriya, Madhubala
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	Jaya swetha, Kaviya sri, Lakshmipriya, Madhubala
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	Jaya swetha, Kaviya sri, Lakshmipriya, Madhubala
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	Jaya swetha, Kaviya sri, Lakshmipriya, Madhubala
Sprint-3		US-2	Create a Node-RED service.	10	High	Jaya swetha, Kaviya sri, Lakshmipriya, Madhubala
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	Jaya swetha, Kaviya sri, Lakshmipriya, Madhubala
Sprint-3		US-2	After developing python code, commands are received just print the statements which represent the control of the devices.	5	Medium	Jaya swetha, Kaviya sri, Lakshmipriya, Madhubala
Sprint-4		US-3	Publish Data to The IBM Cloud	8	High	Jaya swetha, Kaviya sri, Lakshmipriya, Madhubala
Sprint-4		US-1	Create Web UI in Node- Red	10	High	Jaya swetha, Kaviya sri, Lakshmipriya, Madhubala
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	Jaya swetha, Kaviya sri, Lakshmipriya, Madhubala

PROJECT TRACKER, VELOCITY & BURNDOWN CHART: (4 MARKS)

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

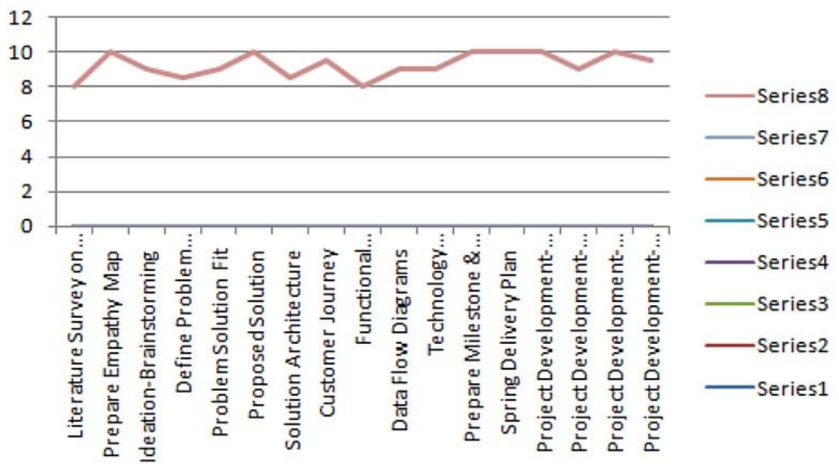
VELOCITY:

Imagine we have a 10-day sprint duration, and the velocity of the teamis 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

BURNDOWN CHART:

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burndown charts can be applied to any project containing measurable progress overtime



7. CODING & SOLUTIONING (EXPLAIN THE FEATURES ADDED IN THEPROJECT ALONG WITH THE CODE):

7.1 FEATURE 1

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep  5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
import cv2
import numpy as np
import wiot.sdk.device
import playsound
import random
import time
import datetime
import ibm_boto3
from ibm_botocore.client import Config, ClientError

#CloudantDB
from cloudant.client import Cloudant
from cloudant.error import CloudantException
from cloudant.result import Result, ResultByKey
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2_grpc
stub = service_pb2_grpc.V2Stub(clarifaiChannel.get.grpc_channel())
from clarifai_grpc.grpc.api import service_pb2, resource_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2

#This is how you authenticate
metadata = (('authorization', 'key 0620e202302b4508b90eab7efe7475e4'),)
COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID = "g5d4qO8EIgv4TWUCJj4hfEzgalqEjrDbE82AJDW1AOHo"
COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"
COS_RESOURCE_CRN = "crn:v1:bluemix:public:cloud-object-storage:global:a/c2fa2836eaf3434bbc8b5b58fefff3f0:62e450fd-4c82-4153-ba41-ccb53adb8111::"
clientdb = cloudant("apikey-W2njldnwtjO16V53LAVUCqPwc2aHTlmlj1xXvtdGKJBn", "88cc5f47c1a28afbfb8ad16161583f5a", url="https://d6c89f97-cf91-48b7-b14b-c99b2fe27c2f-bluemix.clouda
clientdb.connect()

#Create resource
cos = ibm_boto3.resource("s3",
                         ibm_api_key_id=COS_API_KEY_ID,
```

```
                        config=Config(signature_version="oauth"),
                        endpoint_url=COS_ENDPOINT
                        )
def - multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(item_name, bucket_name))
        #set 5 MB chunks
        part_size = 1024 * 1024 * 5
        #set threadhold to 15 MB
        file_threshold = 1024 * 1024 * 15
        #set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
            )
        #the upload_fileobj method will automatically execute a multi-part upload
        #in 5 MB chunks size
        with open(file_path, "rb") as file_data:
            cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
                )
        print("Transfer for {0} Complete!\n".format(item_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: {0}".format(e))

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)
    command=cmd.data['command']
    print(command)
    if(commamd=="lighton"):
        print('lighton')
    elif(command=="lightoff"):
        print('lightoff')
    elif(command=="motoron"):
        print('motoron')
```

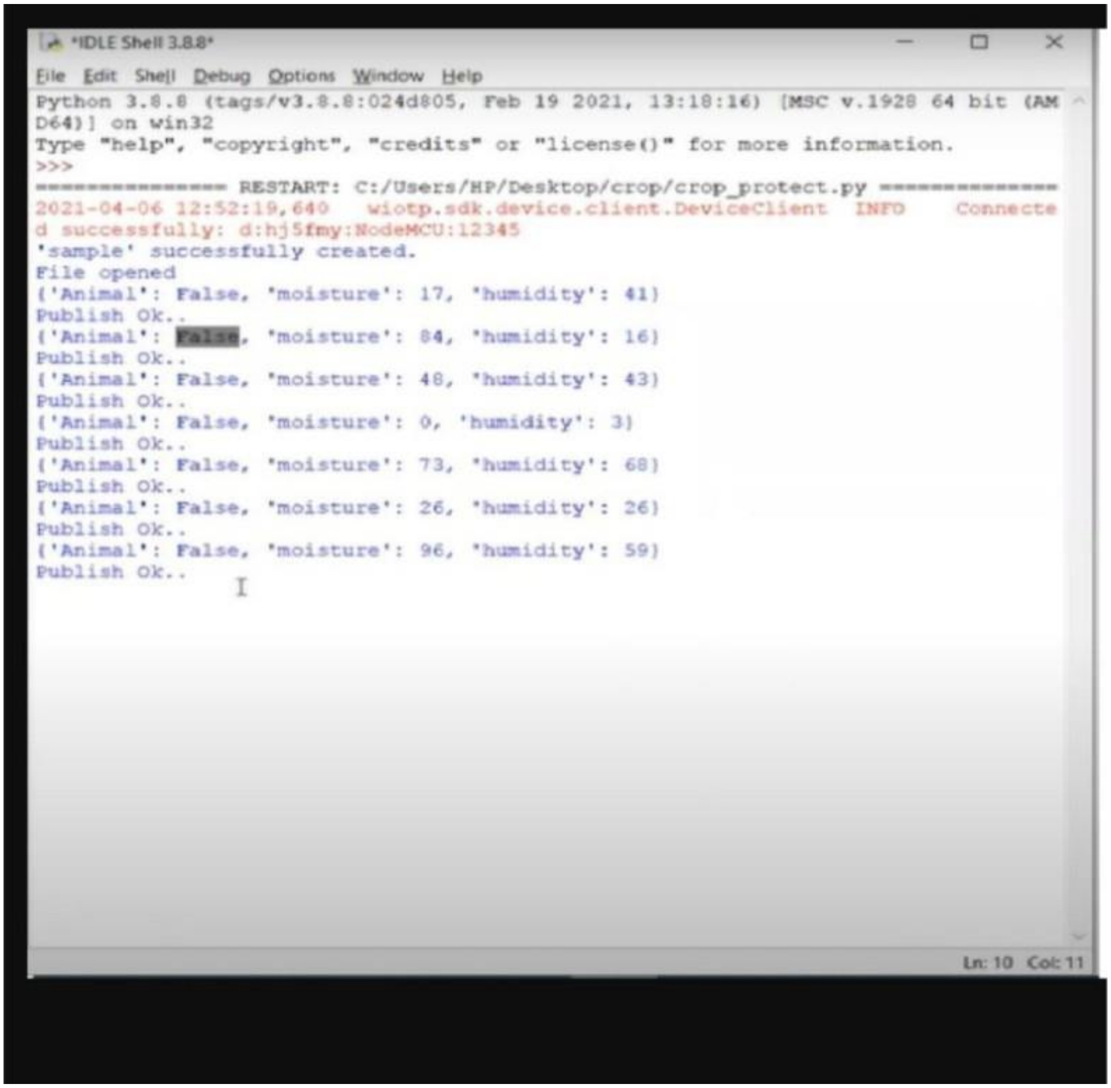
```
        print('motoron')
    elif(command=="motoroff"):
        print('motoroff')
myConfig = {
    "identity": {
        "orgId": "chytun",
        "typeId": "NodeMCU",
        "deviceId": "12345"
        },
    "auth": {
        "token": "12345678"
        }
    }
client = wiot.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

database_name = "sample"
my_database = clientdb.create_database(database_name)
if my_dtabase.exists():
    print(f"'(database_name)' successfully created.")
cap=cv2.VideoCapture("garden.mp4")
if(cap.isOpened()==True):
    print('File opened')
else:
    print('File not found')

while(cap.isOpened()):
    ret, frame = cap.read()
    gray = cv3.cvtColor(frame, cv2.COLOR_BGR@GRAY)
    imS= cv2.resize(frame, (960,540))
    cv2.inwrite('ex.jpg',imS)
    with open("ex.jpg", "rb") as f:
        file_bytes = f.read()
    #This is the model ID of a publicly available General model. You may use any other public or custom model ID.
    request = service_pb2.PostModeloutputsRequest(
        model_id='e9359dbe6ee44dbc8842ebe97247b201',
...         inputs=[resources_pb2.Input(data=resources_pb2.Data(image=resources_pb2.Image(base64=file_bytes))
```


```
...         inputs=[resources_pb2.Input(data=resources_pb2.Data(image=resources_pb2.Image(base64=file_bytes))
```


```
...                                     )])
...     response = stub.PostModelOutputs(request, metadata=metadata)
...     if response.status.code != status_code_pb2.SUCCESS:
...         raise Exception("Request failed, status code: " + str(response.status.code))
...     detect=False
...     for concept in response.outputs[0].data.concepts:
...         #print('%12s: %.f' % (concept.name, concept.value))
...         if(concept.value>0.98):
...             #print(concept.name)
...             if(concept.name=="animal"):
...                 print("Alert! Alert! animal detected")
...                 playsound.playsound('alert.mp3')
...                 picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
...                 cv2.inwrite(picname+'.jpg',frame)
...                 multi_part_upload('Dhakshesh', picname+'.jpg', picname+'.jpg')
...                 json_document={"link":COS_ENDPOINT+'/'+'Dhakshesh'+'/'+picname+'.jpg'}
...                 new_document = my_database.create_document(json_document)
...                 if new_document.exists():
...                     print(f"Document successfully created.")
...                 time.sleep(5)
...                 detect=True
...     moist=random.randint(0,100)
...     humidity=random.randint(0,100)
...     myData={'Animal':detect,'moisture':moist,'humidity':humidity}
...     print(myData)
```

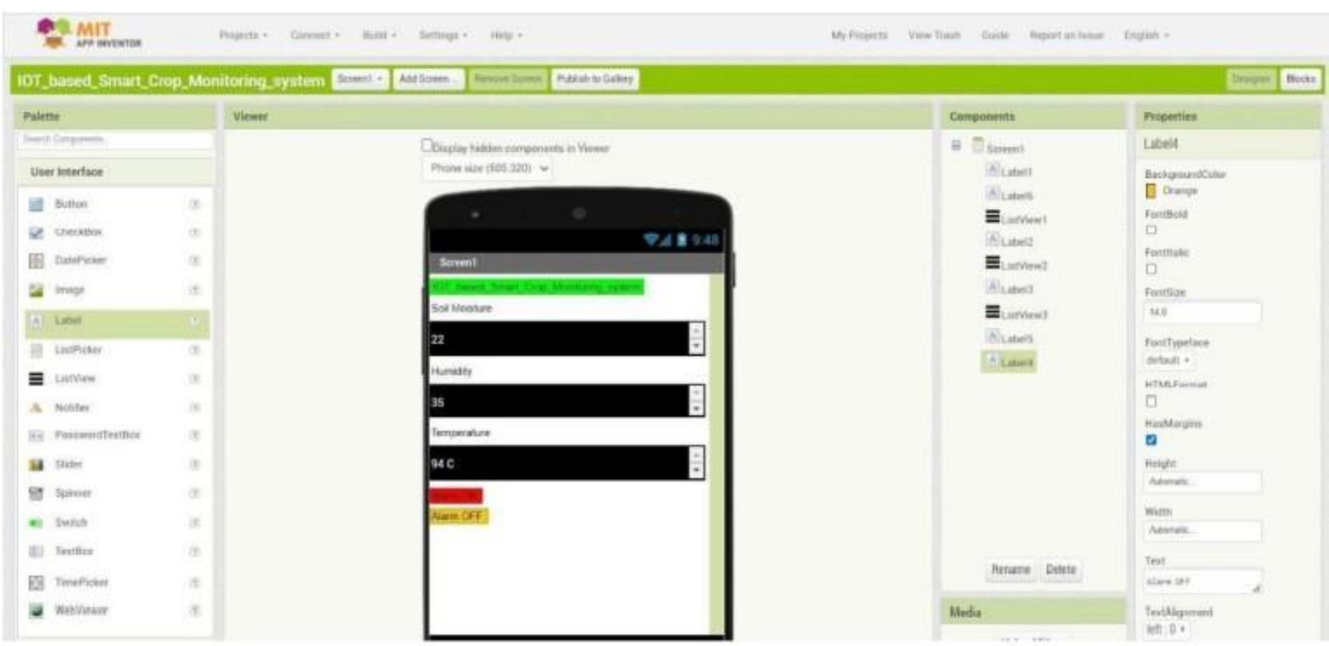



```
Python 3.8.8 (tags/v3.8.8:024d805, Feb 19 2021, 13:18:16) [MSC v.1928 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:/Users/HP/Desktop/crop/crop_protect.py ==============
2021-04-06 12:52:19,640   wiotp.sdk.device.client.DeviceClient  INFO    Connecte
d successfully: d:hj5fmy:NodeMCU:12345
'sample' successfully created.
File opened
{'Animal': False, 'moisture': 17, 'humidity': 41}
Publish Ok..
{'Animal': False, 'moisture': 84, 'humidity': 16}
Publish Ok..
{'Animal': False, 'moisture': 48, 'humidity': 43}
Publish Ok..
{'Animal': False, 'moisture': 0, 'humidity': 3}
Publish Ok..
{'Animal': False, 'moisture': 73, 'humidity': 68}
Publish Ok..
{'Animal': False, 'moisture': 26, 'humidity': 26}
Publish Ok..
{'Animal': False, 'moisture': 96, 'humidity': 59}
Publish Ok..
```

7.2 FEATURE 2

MIT APP INVENTOR TO DESIGN THE APP

CUSTOMIZING THE APP INTERFACE TO DISPLAY THE VALUES:



7.3 DATABASE SCHEMA

PYTHON CODE TO IBM

```
import time import sys
import
ibmiotf.application
import ibmiotf.device import random
#Provide your IBM Watson Device Credentials organization
= "wu5b55" deviceType = "crop1" deviceId =
"1234"authMethod = "token" authToken =
"1234567890"
# Initialize
GPIOtry:
        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()

# Connect and send a datapoint "hello" with value "world" into thecloud as an event of
type"greeting" 10 times
deviceCli.connect
()while True:

          #Get Sensor Data from DHT11


          temp=random.randint(0,100)
          Hum=random.randint(0,100)
          moisture=random.randint(0,100)
          data = { 'temperature' : temp, 'Humidity':
Hum,'Moisture':moisture }

#print data def myOnPublishCallback():

               print ("Temperature = " + str(temp)+" C Humidity = " + str(hum)+ " moisture
=" + str(moisture) + "to IBM Watson")
          success = deviceCli.publishEvent("IoTSensor", "json",
data,qos=0,on_publish=myOnPublishCallback) if not success:

               print("Not connected to
          IoTF")time.sleep(10)
```

deviceCli.commandCallback = myCommandCallback

Disconnect the device and application from the clouddeviceCli.disconnect()

8. TESTING

8.1.TEST CASES

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	11	4	2	2	19
Duplicate	1	1	2	0	4
External	2	3	0	1	6
Fixed	10	2	3	20	35
Not Reproduced	0	0	2	0	2
Skipped	0	0	2	1	3
Won't Fix	0	5	2	1	8
Totals	24	15	13	25	77

Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	1	4
Client Application	47	0	2	45

Security	3	0	0	3
Outsource Shipping	2	0	0	2
Exception Reporting	11	0	2	9
Final Report Output	5	0	0	5
Version Control	3	0	1	2

9. RESULTS

Thus the IOT based Smart Crop Protection has been build successfully with the help of MIT app, Node.Js, and node red. And theoutput has been tested and verified using MIT app.

The problem of crop vandalization by wild animals and fire hasbecome a major social problem in current time.

It requires urgent attention as no effective solution exists till date for this problem. Thus this project carries a great social relevance as it aims toaddress this problem. This project will help farmers in protecting their orchards and fields and save them from significant financial losses and willsave them from the unproductive efforts that they endure for the protectiontheir fields. This will also help them in achieving better crop yields thus leading to their economic wellbeing.

10. ADVANTAGES AND DISADVANTAGES

Advantage:

- Farmers can monitor the health of farm animals closely, even if they are physicallydistant.
- Smart farming systems reduce waste, improve productivity and enable management ofa greater number of resources through remote sensing.
- High reliance.
- Enhanced Security.

Disadvantage:

- Farms are located in remote areas and are far from access to the internet.
- A farmer needs to have access to crop data reliably at any time from any location, soconnection issues would cause an advanced monitoring system to be useless.

- High Cost
- Equipment needed to implement IoT in agriculture is expensive.

APPLICATIONS

- Monitoring the crop field with the help of sensors (light, humidity, temperature, soilmoisture, etc.)
- Automating the irrigation system
- Soil Moisture Monitoring (including conductivity and pH)

11. CONCLUSION

The aim of this project is to make the life and work of the farmer much easier. This can be achieved using the technique - Precision Farming, thisinvolves autonomous monitoring of crops and other environmental parameters which has an effect on the crop, these environmental conditions are:

1. Environmental Humidity
2. Environmental Temperature.
3. Soil Moisture.
4. Rain Sensing.

Above mentioned are some of the conditions monitored autonomously, threshold parameters for various crops are automatically set upon userinput of crop variety to be monitored. By this system one could achieve agood yield and better nutritional crops in their agricultural produce.

12. FUTURE SCOPE

Future scope of our project relies on the farmers and theirfeedbacks, in future we are planning to add the following features:

1. One device one farm - Cover the entire farm area with a single device.
2. Pest monitoring system.
3. Estimated yield calculator.
4. Estimated time of cultivation.
5. Individual cloud management dashboard.

13. APPENDX

SOURCECODE

MOTOR.PY

```
impor time

import sys

import  ibmiotf.application   # to install pip install
ibmiotf import ibmiotf.device

    # Provide your IBM Watson Device
    Credentials organization = "63004g"
    # replace the ORG ID
    deviceType = "MainDevice" # replace the
    Device type deviceId = "9344022806" # replace
    Device ID
    authMethod = "token"
    authToken = "a-63004g-86womzydrf" # Replace the authtoken

    def myCommandCallback(cmd): # function for

        Callback if cmd.data['command'] ==

        'motoron':

            print("MOTOR ON IS RECEIVED")

        elif cmd.data['command'] ==
            'motoroff': print("MOTOR
            OFF IS RECEIVED")
```

```
    if cmd.command ==

        "setInterval": if

        'interval' not in

        cmd.data:

            print("Error - command is missing required information: 'interval'")
        else:
            interval =
    cmd.data['interval'] elif
    cmd.command ==
    "print":
        if 'message' not in cmd.data:
            print("Error - command is missing required information: 'message'")

        else:
            output =
            cmd.data['message
            '] print(output)



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()

# Connect and send a datapoint "hello" with value "world" into the cloud as
an event of type "greeting" 10 times
deviceCli.connect()


while True:
    deviceCli.commandCallback = myCommandCallback


#   Disconnect   the   device   and   application   from
the  cloud deviceCli.disconnect()
```

SENSOR.PY

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random


#Provide your IBM Watson Device Credentials
organization = "63004g"
deviceType =
"MainDevice"
deviceId =
```

```
"9344022806"
authMethod = "token"
authToken =
"9944611970"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status ==
        "motoron":
        print ("motor
        is on")
    elif status ==
        "motoroff":
        print ("motor
        is off")
    else :
        print ("please send proper command")


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()

# Connect and send a datapoint "hello" with value "world" into the cloud as
an event of type "greeting" 10 times
deviceCli.connect()


while True:
    #Get Sensor Data from
    DHT11
    animal=random.uniform(0.
    1, 0.99)
    moisture=random.randint(0
    ,110)
    temperature=random.randin
    t(-20,125)
    Humid=random.randint(0,1
    00)
```

```
        data = {'animal':animal,'moisture': moisture, 'temperature' : temperature,
'Humid': Humid }
        #print data
        def myOnPublishCallback():
            print ("Published Soil Moisture = %s %%" %moisture,"Temperature =
%s C" % temperature, "Humidity = %s %%" % Humid,'animal = %s'%animal,
"to IBM Watson")
            if
                animal>
                0.98:
                print("
                Alert")
        success      =      deviceCli.publishEvent("IoTSensor",
"json",                              data,                    qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected
        to IoTF")
        time.sleep(10)

        deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

GitHub Link

https://github.com/IBM-EPBL/IBM-Project-15272-1659596184

PROJECT DEMO VIDEO LINK

https://drive.google.com/file/d/1Sx3sLVRmCvMvWVF3SWlCKvG44kJVd9ME/view?usp=drivesdk