IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE .

TEAM ID: PNT2022TMID15789

NALAIYA THIRAN PROJECT BASED LEARNING ON PROFESSIONAL READINESS FOR INNOVATION EMPLOYMENT AND ENTERPRENEURSHIP .

A PROJECT REPORT BY

KAMESH C	**– [19BEC4088]**
MANIKANDAN P	**– [19BEC4113]**
MANIVEL R	**– [19BEC4114]**
MOHAN P	**– [19BEC4119]**

BACHELOR OF ENGINEERING
IN

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
M.KUMARASAMY COLLEGE OF ENGINEERING

(Autonomous)

KARUR – 639113

ANNA UNIVERSITY CHENNAI 600025

INDEX

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

An intelligent crop protection system helps the farmers in protecting the crop from the animals and birds which destroy the crop. This system also helps farmers to monitor the soil moisture levels in the field and also the temperature and humidity values near the field. The motors and sprinklers in the field can be controlled using the mobile application.

2. LITERATURE SURVEY

2.1 Existing Problem

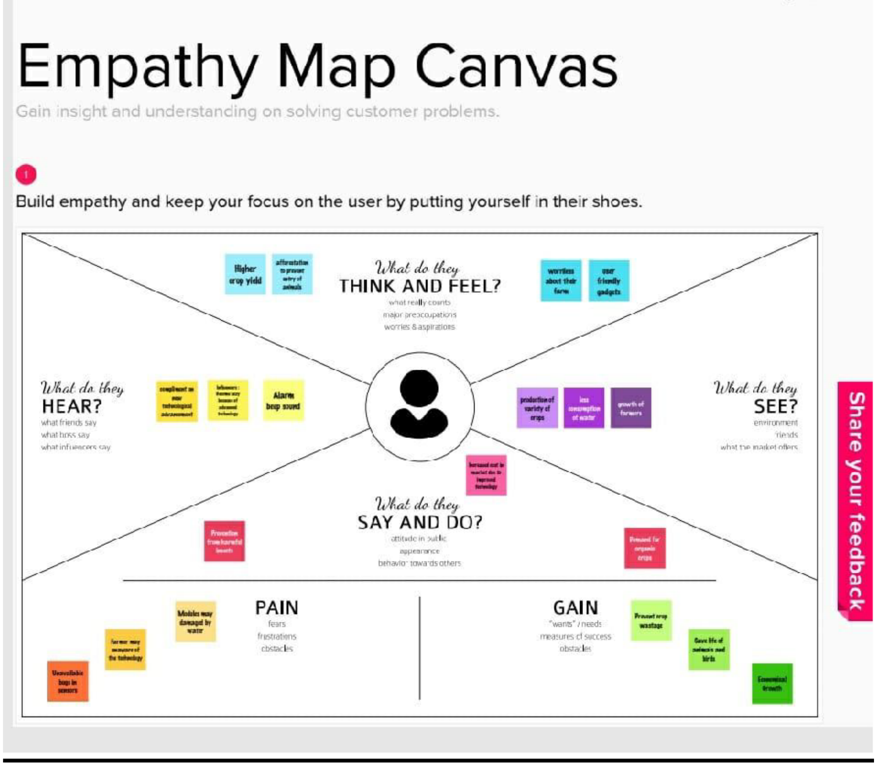
Most of the farmers are facing many problems nowadays due to many reasons. Our problem to solve is the invasion of various species such as birds and animals that harm the crops that are being cultivated. Various types of species such as birds and animals come to the cultivation field according to the crop that is being cultivated and also according to the season of cultivation. Some wild animals enter the field during night times when the field is near a forest region or when the farm cultivates some fruits and other crops that attract animals. Some animals cross the field in search of food and water and also the birds enter the field for food and they damage all the crops. When the animals enter the field they not only eat food but they also damage the entire field by walking upon the crops and also by spoiling the food crops. The birds, by entering the field they come to eat seeds of the crops and also they tend to drag the crops and ruin the entire field. Some birds enter the field to eat the insects and pests in the field.

2.2 Problem Statement Definition

Most of the farmers are facing many problems nowadays due to many reasons. Our problem to solve is the invasion of various species such as birds and animals that harm the crops that are being cultivated. Various types of species such as birds and animals come to the cultivation field according to the crop that is being cultivated and also according to the season of cultivation. Some wild animals enter the field during night times when the field is near a forest region or when the farm cultivates some fruits and other crops that attract animals.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



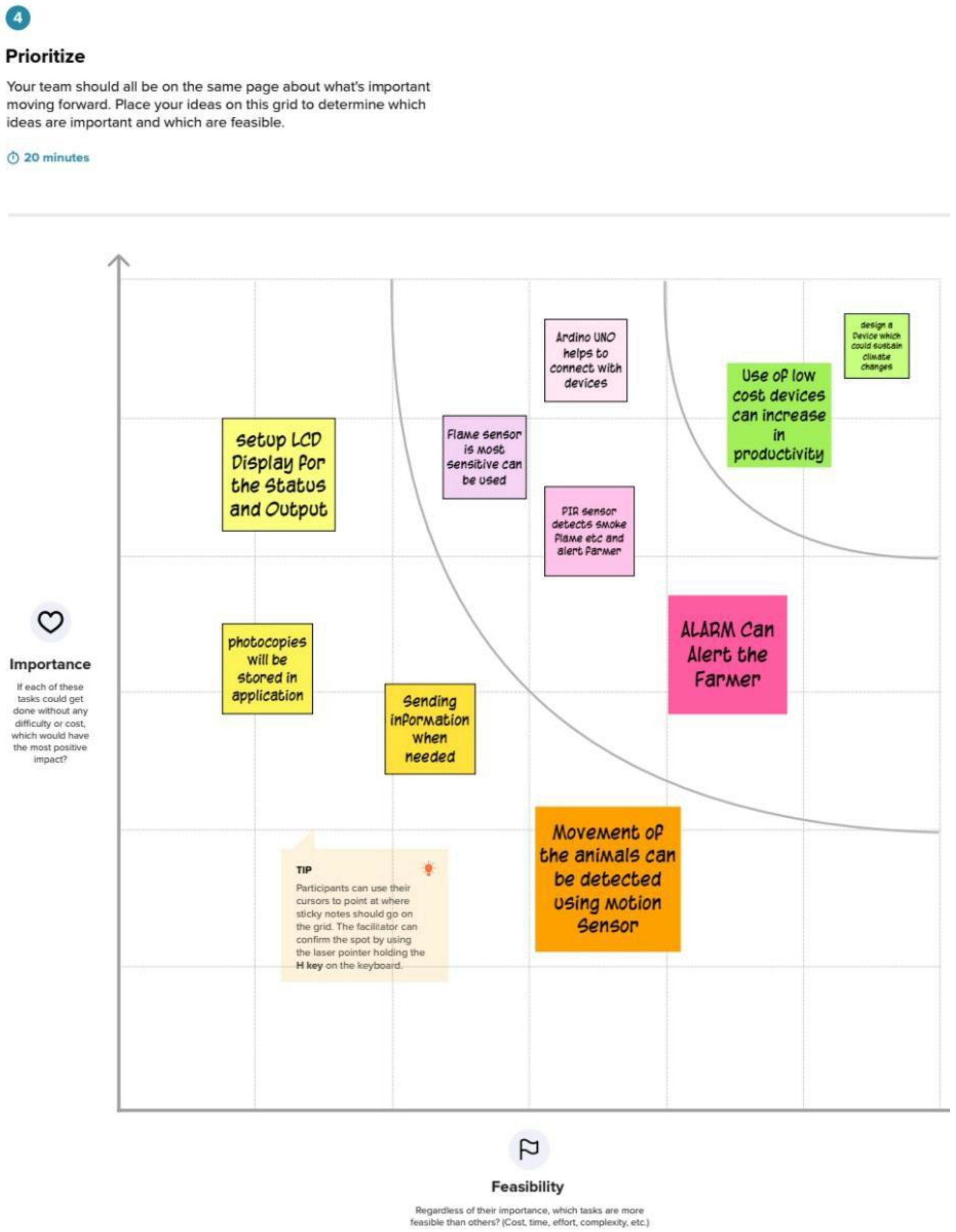
Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.

3.2 Ideation and Brainstorming



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Ardino UNO helps to connect with devices

design a Device which could sustain climate changes

Use of low cost devices can increase in productivity

Setup LCD Display for the Status and Output

Flame sensor is most sensitive can be used

PIR sensor detects smoke flame etc and alert farmer

ALARM Can Alert the Farmer

photocopies will be stored in application

Sending information when needed

Movement of the animals can be detected using motion Sensor

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

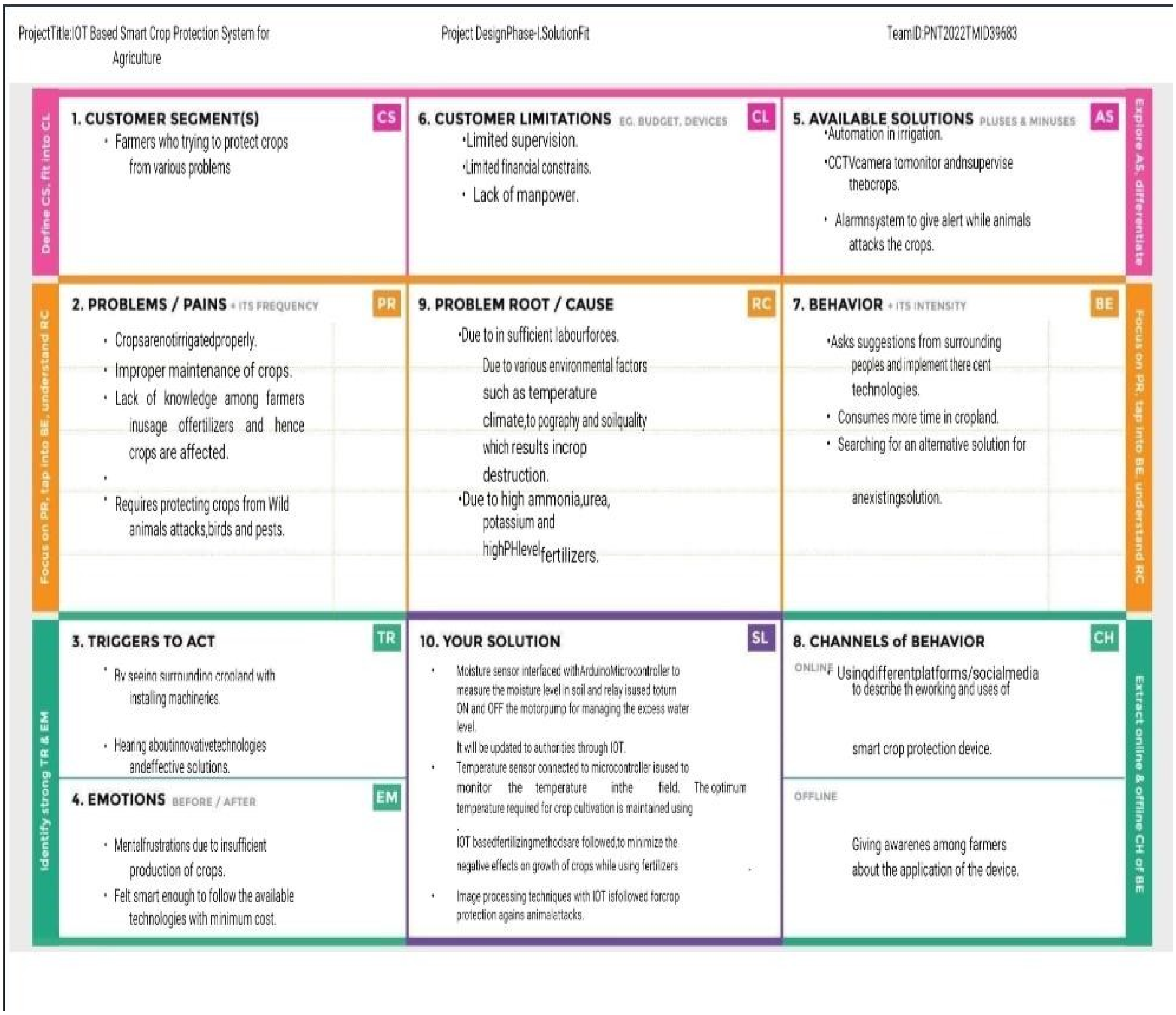
Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 Proposed Solution

S.No	Parameter	Description
1.	• Problem Statement (Problem to be solved)	• Crops are not irrigated properly due to insufficient labour forces. • Improper maintenance of crops against various environmental factors such as temperature climate, topography and soil quality which results in crop destruction. • Requires protecting crops from Wild animals attacks, birds and pests.
2.	• Idea / Solution description	• Moisture sensor is interfaced with Arduino Microcontroller to measure the moisture level in soil and relay is used to turn ON and OFF the motor pump for managing the excess water level. It will be updated to authorities through IOT. • Temperature sensor connected to microcontroller is used to monitor the temperature in the field. • Image processing techniques with IOT is followed for crop protection against animal attacks.
3.	Novelty / Uniqueness	• Automatic crop maintenance and protection using embedded and IOT technology.
4.	Social Impact / Customer Satisfaction	• This proposed system provides many facilities which helps the farmers to maintain the crop field without much loss.
5.	Business Model (Revenue Model)	• This prototype can be developed as product with minimum cost with high performance .
6.	Scalability of the Solution	• This can be developed to a scalable product by using sensors and transmitting the data through Wireless Sensor Network and Analysing the data in cloud and operations is performed using robots .

3.4 Problem Solution Fit



ProjectTitle:IOT Based Smart Crop Protection System for Agriculture

Project DesignPhase-I.SolutionFit

TeamID:PNT2022TMID39683

1. CUSTOMER SEGMENT(S) CS

- Farmers who trying to protect crops from various problems

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL

- •Limited supervision.
- •Limited financial constrains.
- Lack of manpower.

5. AVAILABLE SOLUTIONS PLUSES & MINUSES AS

- •Automation in irrigation.
- •CCTVcamera tomonitor andnsupervise thebcrops.
- Alarmnsystem to give alert while animals attacks the crops.

2. PROBLEMS / PAINS + ITS FREQUENCY PR

- Cropsarenotirrigatedproperly.
- Improper maintenance of crops.
- Lack of knowledge among farmers inusage offertilizers and hence crops are affected.
-
- Requires protecting crops from Wild animals attacks,birds and pests.

9. PROBLEM ROOT / CAUSE RC

•Due to in sufficient labourforces.

Due to various environmental factors such as temperature climate,to pography and soilquality which results incrop destruction.

•Due to high ammonia,urea, potassium and highPHlevelfertilizers.

7. BEHAVIOR + ITS INTENSITY BE

- •Asks suggestions from surrounding peoples and implement there cent technologies.
- Consumes more time in cropland.
- Searching for an alternative solution for anexistingsolution.

3. TRIGGERS TO ACT TR

- By seeing surrounding cropland with installing machineries.
- Hearing aboutinnovativetechnologies andeffective solutions.

4. EMOTIONS BEFORE / AFTER EM

- Mentalfrustrations due to insufficient production of crops.
- Felt smart enough to follow the available technologies with minimum cost.

10. YOUR SOLUTION SL

- Moisture sensor interfaced withArduinoMicrocontroller to measure the moisture level in soil and relay isused toturn ON and OFF the motorpump for managing the excess water level.

  It will be updated to authorties through IOT.
- Temperature sensor connected to microcontroller isused to monitor the temperature inthe field. The optimum temperature required for crop cultivation is maintained using .

  IOT basedfertilizingmethodsare followed,to minimize the negative effects on growth of crops while using fertilizers
- Image processing techniques with IOT isfollowed forcrop protection agains animalattacks.

8. CHANNELS of BEHAVIOR CH

ONLINE

- Usingdifferentplatforms/socialmedia to describe th eworking and uses of smart crop protection device.

OFFLINE

Giving awarenes among farmers about the application of the device.

Define CS, fit into CL
Explore AS, differentiate
Focus on PR, tap into BE, understand RC
Focus on PR, tap into BE, understand RC
Identify strong TR & EM
Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

Following are the functional requirements of the proposed solution.

- FR-1 User Registration,Registration through Form Registration through Gmail Registration through LinkedIN
- FR-2 User Confirmation ,Confirmation via Email Confirmation via OTP
- FR-3 Tracking Expense Helpful insights about money management
- FR-4 Alert Message Give alert mail if the amount exceeds the budget limit
- FR-5 Category This application shall allow users to add categories of their expenses

4.2 Non Functional requirement

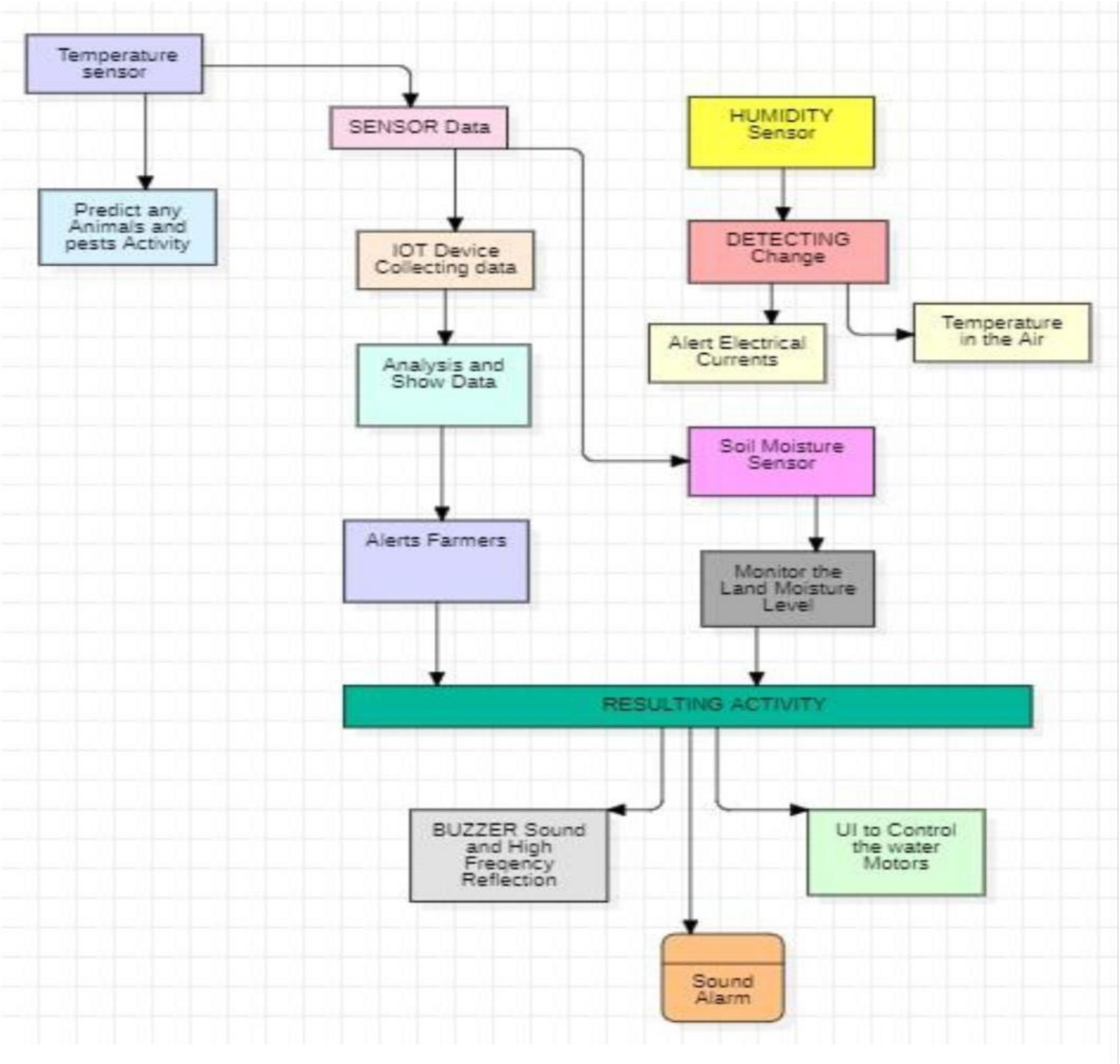
Following are the non-functional requirements of the proposed solution.

- NFR-1 Usability You will able to allocate money to different priorities and also help you to cut down on unnecessary spending
- NFR-2 Security More security of the customer data and bank account details.
- NFR-3 Reliability Used to manage his/her expense so that the user is the path of financial stability. It is categorized by week, month, and year and also helps to see more expenses made. Helps to define their own categories.
- NFR-4 Performance The types of expense are categories along with an option .Throughput of the system is increased due to light weight database support.
- NFR-5 Availability Able to track business expense and monitor important for maintaining healthy cash flow. NFR-6 Scalability The ability to appropriately handle increasing demands.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is store.

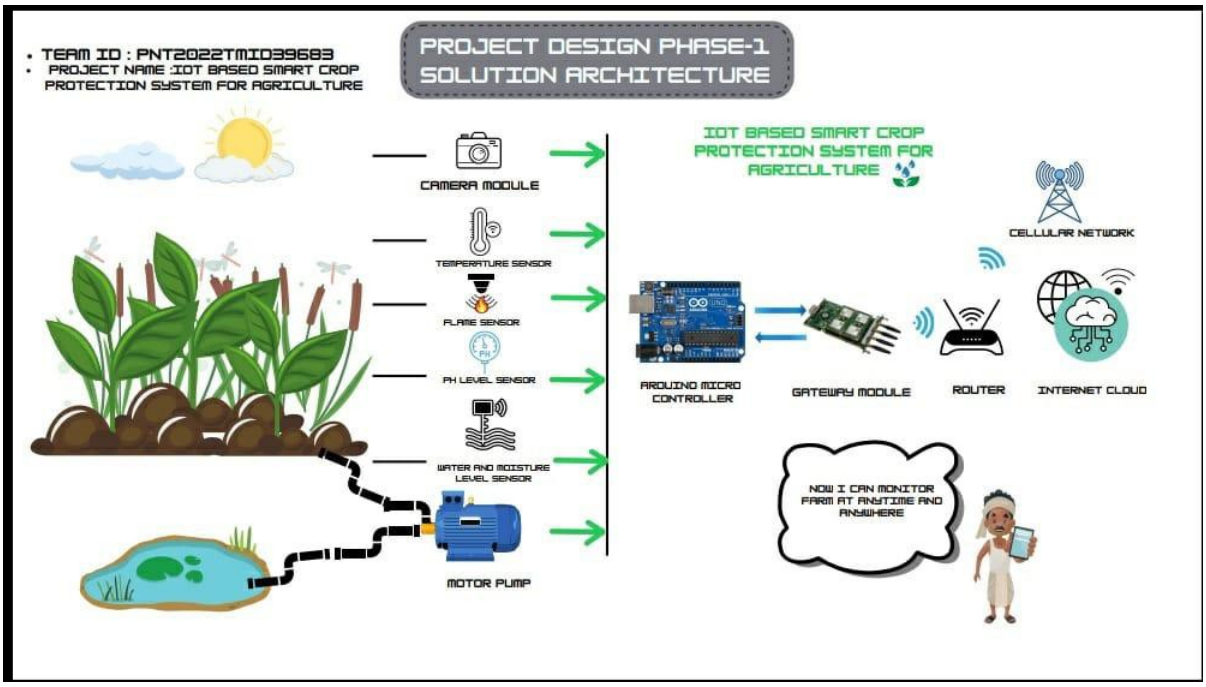

Temperature sensor
SENSOR Data
HUMIDITY Sensor
Predict any Animals and pests Activity
IOT Device Collecting data
DETECTING Change
Alert Electrical Currents
Temperature in the Air
Analysis and Show Data
Soil Moisture Sensor
Alerts Farmers
Monitor the Land Moisture Level
RESULTING ACTIVITY
BUZZER Sound and High Freqency Reflection
UI to Control the water Motors
Sound Alarm

5.2 Solution 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, behaviour, 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.

5.3 Solution Architecture Diagram:


• TEAM ID : PNT2022TMID39683
• PROJECT NAME :IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE
PROJECT DESIGN PHASE-1
SOLUTION ARCHITECTURE
CAMERA MODULE
TEMPERATURE SENSOR
FLAME SENSOR
PH LEVEL SENSOR
WATER AND MOISTURE LEVEL SENSOR
MOTOR PUMP
IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE
CELLULAR NETWORK
ARDUINO MICRO CONTROLLER
GATEWAY MODULE
ROUTER
INTERNET CLOUD
NOW I CAN MONITOR FARM AT ANYTIME AND ANYWHERE

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint1	Sensor Data(python script)	USN-1	The Data of sensor which are feed to the Raspberry pi .Here we are using python script to generate a random sensor data.	3	High	KAMESH C (Team leader)
Sprint1	Automation (python script)	USN-2	Some activities are made to automation to overcome insufficient of labour force in the field. Hence that also included in python script to implement automation	5	High	KAMESH C (Team leader)
Sprint2	IBM IOT platform	USN-3	To Send the raspberry pi data to IOT platform, we create an IBM IOT platform and connect the raspberry pi to the device created in IBM IOT.	5	High	MANIVEL R (Team Member)
Sprint3	Node RED service	USN-4	To access the IBM IOT platform from external application or from external UI Node red service is established.	5	High	KAMESH C MANIKANDAN P (Team Member)
Sprint3	API Key	USN-5	To protect the IBM IOT platform creating an API Key .		High	MOHAN P MANIVEL R (Team Member)
Sprint4	User Application	USN-6	To monitor and control the field sensors the User is provided with an User application created by MIT app inventor	8	High	KAMESH C MANIVEL R MOHAN P MANIKANDAN P (Team Member)

Project Tracker, Velocity & Burndown Chart :

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	8	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	5	6 Days	31 Oct 2022	05 Nov 2022	5	05 Nov 2022
Sprint-3	8	6 Days	07 Nov 2022	12 Nov 2022	8	12 Nov 2022
Sprint-4	8	6 Days	14 Nov 2022	19 Nov 2022	8	19 Nov 2022

7. Coding And Solutioning :

7.1 Features

Feature 1: Detect the Temperature

Feature 2: Detect the Humidity

Feature 3: Detect the Moisture

Feature 4: Detect the Animals

Codes:

PYTHON CODE TO IBM:

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

#Provide your IBM Watson Device Credentials
organization        =        "iritj7"
deviceType = "abcd"
```

```
deviceId = "12345"
authMethod = "token"
authToken = "12345678"


# Initialize GPIO
def myCommandCallback(cmd):
print("Command received: %s" % cmd.data['command'])
status=cmd.data['command']
if status=="lighton":
print ("led is on")
elif status == "lightoff":
print ("led 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

temp=random.randint(90,110)
Humid=random.randint(60,100)
Moist=random.randint(20,100)
Animal_dect=random.randint(1,20)

 data = { 'temp' : temp, 'Humid': Humid, 'Moist' : Moist, 'Animal_dect' :
Animal_dect }
 #print data
 def myOnPublishCallback():
```

```
 print ("Published Temperature = %s C" % temp, "Humidity = %s %%"
% Humid, "to IBM Watson", "Published Moisture= %s" % Moist, "Published
Animal detection = " , Animal_dect)

 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()
```

NODE RED CODE:

TEMPERATURE:

```
msg.payload=msg.payload.”temp”

return msg;
```

HUMIDITY:

```
msg.payload=msg.payload.”Humid”

return msg;
```

MOISTURE:

```
msg.payload=msg.payload.”Moist”

return msg;
```

ANIMAL DETECTION:

```
msg.payload=msg.payload.”Animal_dect”

return msg;
```

8. TESTING:

8.1 TESTING :

- PYTHON CODE TO IBM
- IoT SENSOR OUTPUT
- IBM CLOUD TO NODE RED OUTPUT

8.2 User Acceptance Testing:

8.1 Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

8.2 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	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

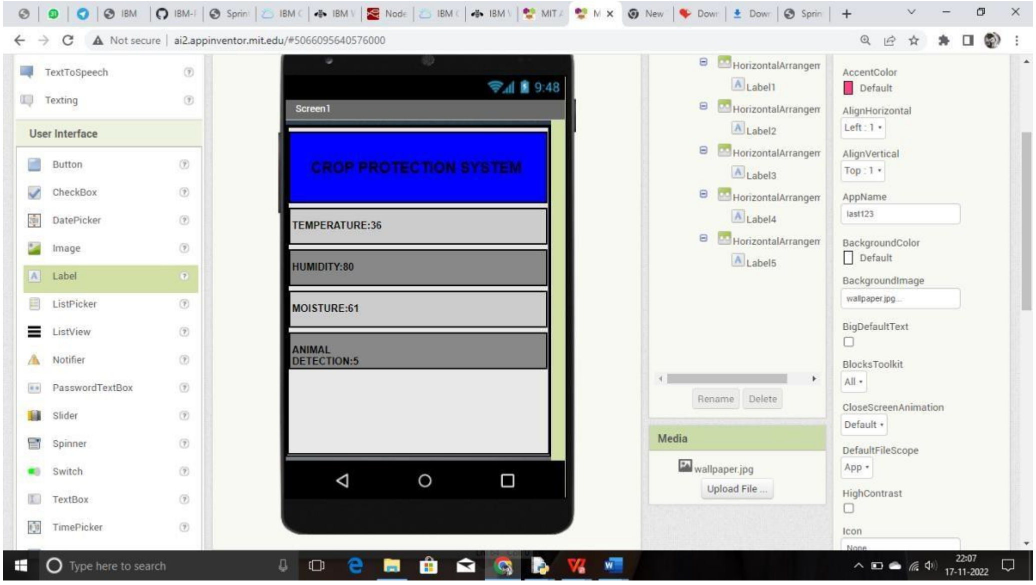
8.3 Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9 RESULT

MIT APP INVENTOR- TO DESIGN THE APP



ai2.appinventor.mit.edu/#5066095640576000

Screen1

CROP PROTECTION SYSTEM

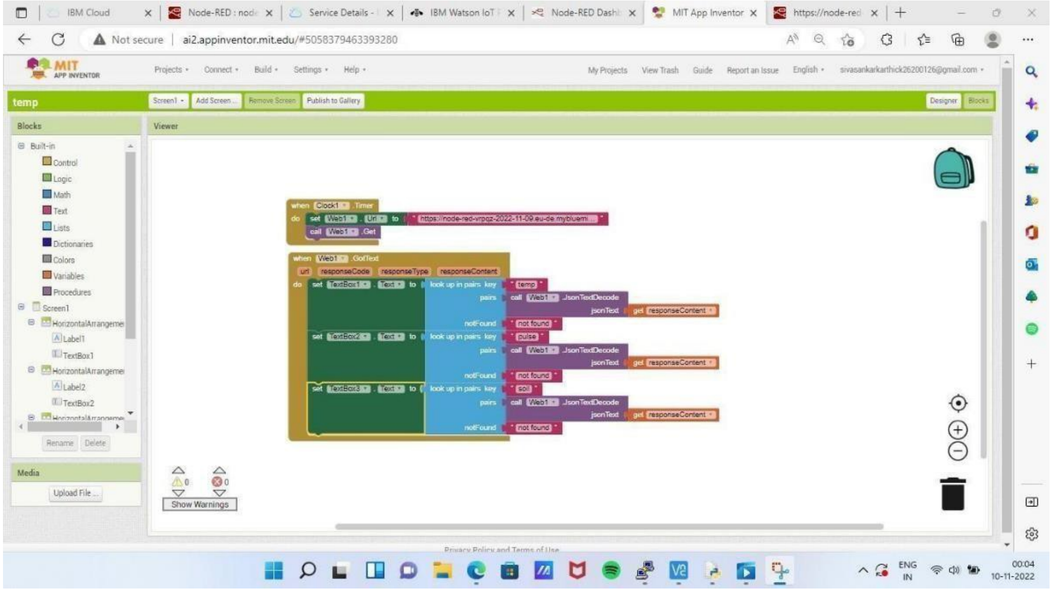
TEMPERATURE:36

HUMIDITY:80

MOISTURE:61

ANIMAL DETECTION:5

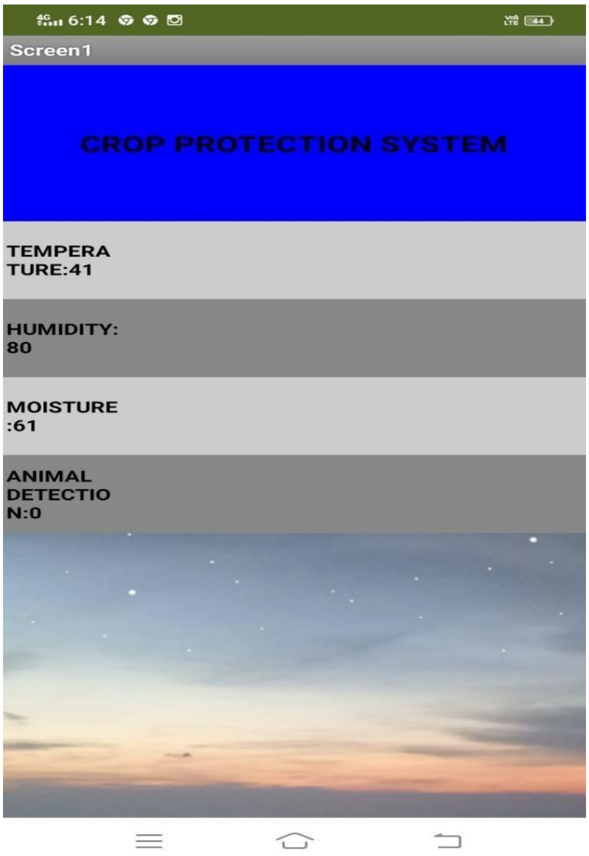
AccentColor: Default
AlignHorizontal: Left : 1
AlignVertical: Top : 1
AppName: last123
BackgroundColor: Default
BackgroundImage: wallpaper.jpg...
BigDefaultText: □
BlocksToolkit: All
CloseScreenAnimation: Default
DefaultFileScope: App
HighContrast: □
Icon: None



```
when Clock1 .Timer
do  set Web1 . Url to " https://node-red-vrpqz-2022-11-09.eu-de.mybluemi... "
    call Web1 .Get

when Web1 .GotText
 url  responseCode  responseType  responseContent
do  set TextBox1 . Text to  look up in pairs  key " temp "
                                              pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseContent
                                           notFound " not found "
    set TextBox2 . Text to  look up in pairs  key " pulse "
                                              pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseContent
                                           notFound " not found "
    set TextBox3 . Text to  look up in pairs  key " soil "
                                              pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseContent
                                           notFound " not found "
```

MIT AI2 COMPANION APP – TO DISPLAY THE OUTPUT VIA QR CODE



CROP PROTECTION SYSTEM

TEMPERA
TURE:41

HUMIDITY:
80

MOISTURE
:61

ANIMAL
DETECTIO
N:0

ADVANTAGES :

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

DISADVANTAGES:

- 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, so connection 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, soil moisture, etc.)
- Automating the irrigation system
- Soil Moisture Monitoring (including conductivity)

CONCLUSION:

The problem of crop vandalization by wild animals and fire has become 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 to address this problem. This project will help farmers in protecting their orchards and fields and save them from significant financial losses and will save them from the unproductive efforts that they endure for the protection their fields. This will also help them in achieving better crop yields thus leading to their economic well being.

FUTURE SCOPE:

Study and analysis of the developed Crop protection systems for its cost effectiveness with the development of Arduino based variable frequency Ultrasonic bird deterrent circuit. outline of the crop damage caused by a particular Wild animal if the behavioral features of the With the reduced cost in the smart phones.

APPENDIX:

SOURCE CODE

```
  import time importsys import ibmiotf.application # toinstallpip
  install ibmiotf importibmiotf.device


  # Provide your IBM Watson Device Credentials organization =
  "8gyz7t" # replace the ORG ID deviceType = "weather_monitor"
  #replace the Device type deviceId = "b827ebd607b5" # replace
  Device ID authMethod = "token"authToken =
  "LWVpQPaVQ166HWN48f" # Replace the authtoken


  def myCommandCallback(cmd): # function for Callbackif



     cm.data['command'] == 'motoron':



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

 RECEIVED")if cmd.command == "setInterval":


 else:

if 'interval' not in cmd.data:
   print("Error - command is missing requiredinformation: 'interval'")


   interval =

  cmd.data['interval']elif

  cmd.command == "print":
```

```
  if 'message' not in cmd.data:
              print("Error - commandis missing requiredinformation:
              'message'")else:output = cmd.data['message']
              print(output)

try:

      deviceOptions = {"org": organization, "type": deviceType, "id":
  deviceId,"authmethod": authMethod,
                    "auth-token": authToken}           deviceCli
=  ibmiotf.device.Client(deviceOptions)#
.............................................


exceptException 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 oftype
  "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
  sysimport
  ibmiotf.application
  importibmiotf.device
import random


  # Provide your IBM Watson Device Credentials organization =
  "8gyz7t" # replace the ORG ID deviceType = "weather_monitor"
  #replace the Device type deviceId = "b827ebd607b5" # replace
  Device ID authMethod = "token"authToken =
  "LWVpQPaVQ166HWN48f" # Replace the authtoken
```

```
def myCommandCallback(cmd):

        print("Command received: %s" %
    cmd.data['command'])print(cmd)

  try:
                deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
    "auth-method": authMethod, "auth-token":
    authToken} deviceCli =
    ibmiotf.device.Client(deviceOptions)
                #.............................................


  exceptException 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 oftype
    "greeting" 10 times
  deviceCli.connect()


  while True:
           temp=random.randint(0,1
    00)
    pulse=random.randint(0,100)
           soil=random.randint(0,10
           0)


           data = { 'temp' : temp, 'pulse': pulse
           ,'soil':soil}#print data            def
    myOnPublishCallback():
               print ("Published Temperature = %s C" % temp, "Humidity = %s %%"
    %pulse,"Soil Moisture = %s %%" % soil,"to IBM Watson")


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

           deviceCli.commandCallback = myCommandCallback


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

Node-RED FLOW :

```
[
{
"id":"625574ead9839b
34",
"type":"ibmiotout",
"z":"630c8601c5ac3295",
"authentication":"apiKey",
"apiKey":"ef745d48e395ccc0",
"outputType":"cmd",
"deviceId":"b827ebd607b5",
"deviceType":"weather_monitor",
"eventCommandType":"data",
"format":"json",
"data":"data",
"qos":0,
"name":"IBM
IoT",
"service":"registe
red","x":680,
"y":220,
"wires":[]
},
{
"id":"4cff18c3274cccc4","type":"ui_button",
"z":"630c8601c5ac3295",
"name":"",
"group":"716e956.00eed6c",
"order":2,
"width":"0",
"height":"0",
"passthru":false,
"label":"MotorO
N",
"tooltip":"",
"color":"",
"bgcolor":"",
"className":"",
"icon":"",
"payload":"{\"command\":\"motoron\"
}","payloadType":"str",
"topic":"motoron",
"topicType":
"s
tr","x":360,
"y":160, "wires":[["625574ead9839b34"]]},
{
"id":"659589baceb4e0b0",
"type":"ui_button",
"z":"630c8601c5ac3295","name":"",
"group":"716e956.00eed6c",
"order":3,
"width":"0",
"height":"0",
"passthru":true,
"label":"Motor
OFF",
"tooltip":"",
"color":"",
"bgcolor":"",
"className":"",
"icon":"",
"payload":"{\"command\":\"motoroff\"}
","payloadType":"str",
"topic":"motoroff",
"topicType":
"s
tr","x":350,

"y":220, "wires":[["625574ead9839b34"]]},
{"id":"ef745d48e395ccc0","type":"ibmiot",
"name":"weather_monitor","keepalive":"60",
"serverName":"",
"cleansession":true,
"appId":"",
"shared":false},
{"id":"716e956.00eed6c",
"type":"ui_group",
"name":"Form",
"tab":"7e62365e.b7e6b8
","order":1,
"disp":true,
"width":"6",
"collapse":f
alse},
{"id":"7e62365e.b7e6b8",
"type":"ui_tab",
"name":"contorl
",
"icon":"dashboar
d","order":1,
"disabled":false,
"hidden":false}
]


[
{
"id":"b42b5519fee73ee2", "type":"ibmiotin",
"z":"03acb6ae05a0c712",
"authentication":"apiKey",
"apiKey":"ef745d48e395ccc0",

"inputType":"evt",
"logicalInterface":"",
"ruleId":"",
"deviceId":"b827ebd607b5",
"applicationId":"",
"deviceType":"weather_monitor",
"eventType":"+",
"commandType":"",
"format":"json",
"name":"IBMIoT",
"service":"registered",
"allDevices":"",
"allApplications":"",
"allDeviceTypes":"",
"allLogicalInterfaces":
"","allEvents":true,
"allCommands":"",
"allFormats
":"",
"qos":0,
"x":270,
"y":180,
  "wires":[["50b13e02170d73fc","d7da6c2f5302ffaf","a949797028158f3f","a71f164bc3 78bcf1"]]
},
{
"id":"50b13e02170d73f
c",
"type":"function",
"z":"03acb6ae05a0c71
2","name":"Soil
Moisture",
  "func":"msg.payload = msg.payload.soil;\nglobal.set('s',msg.payload);\nreturn msg;",
  "outputs":1,
"noerr":
0,
"initializ
e":"",
"finalize":"",
"libs":[],

"x":490,
"y":120,
"wires":[["a949797028158f3f","ba98e701f55f04fe"]]
},
{
"id":"d7da6c2f5302ffaf","type":"function
","z":"03acb6ae05a0c712",
"name":"Humidity",
  "func":"msg.payload = msg.payload.pulse;\nglobal.set('p',msg.payload)\nreturn msg;",
  "outputs":1,
"noerr":
0,
"initializ
e ":"",
"finalize":"",
"l
i
bs
":
[
],
"
x
":
48
0,
"y":260, "wires":[["a949797028158f3f","70a5b076eeb80b70"]]
},
{
"id":"a949797028158f
3f",
"type":"debug",
"z":"03acb6ae05a0c71
2","name":"IBMo/p",
"active":true,
"tosidebar":true,
"console":false,
"tostatus":false,
"complete":"payload"
, "targetType":"msg",
"statusVal":"",
"statusType":"auto",
"x":780,
"y":180,
"wires":[]
},
{
"id":"70a5b076eeb80b70",
"type":"ui_gauge",
"z":"03acb6ae05a0c712",
"name":"",
"group":"f4cb8513b95c98a4",
"order":6,
"width":"0",
"height":"0",
"gtype":"gage",
"title":"Humidity",
"label":"Percentage(%)",
"format":"{{value}}
","min":0,
"max":"100",
"colors":["#00b500","#e6e600","#ca3838"],
"seg1":"","seg2":"",
"classNam
e
":"","x":86
0,
"y":260,
"wires":[]
},
{
"id":"a71f164bc378bcf1","type":"functi
on","z":"03acb6ae05a0c712",
"name":"Temperature",
  "func":"msg.payload=msg.payload.temp;\nglobal.set('t',msg.payload);\nreturn msg;","outputs":1,
"noerr":
0,
"initializ
e":"",
"finalize":"",
"l
i
bs
":
[
],
```

"
x
":
49
0,
"y":360
,

"wires":[["8e8b63b110c5ec2d","a949797028158f3f"]]
},
{
"id":"8e8b63b110c5ec2d",
"type":"ui_gauge",
"z":"03acb6ae05a0c712",
"name":"",
"group":"f4cb8513b95c98a4",
"order":11,
"width":"0",
"height":"0",
"gtype":"gage",
"title":"Temperature",
"label":"DegreeCelciu
s",
"format":"{{value}}",
"min":0,
"max":"100",
"colors":["#00b500","#e6e600","#ca3838"],"seg1":"",
"seg2":"",
"classNam
e":"",
"x":790,
"y":360,
"wires":[]
},
{
"id":"ba98e701f55f04fe",
"type":"ui_gauge",
"z":"03acb6ae05a0c712",
"name":"",
"group":"f4cb8513b95c98a4",
"order":1,

```
"width":"0",
"height":"0",
"gtype":"gage",

"title":"Soil
Moisture",
"label":"Percentage(%
)",
"format":"{{value}}
","min":0,
"max":"100",
"colors":["#00b500","#e6e600","#ca3838"],"seg1":"",
"seg2":"",
"classNam
e":"",
"x":790,
"y":120,
"wires":[]
},
{
"id":"a259673baf5f0f
98","type":"httpin",
"z":"03acb6ae05a0c7
12","name":"",
"url":"/sensor",
"method":"g
et",
"upload":fals
e,
"swaggerDo
c"
:"","x":37
0,"y":500,
"wires":[["18a8cdbf7943d27a"]]
},
{
"id":"18a8cdbf7943d27a","type":"functi
on","z":"03acb6ae05a0c712",
"name":"httpfunction",
"func":"msg.payload{\"pulse\":global.get('p'),\"temp\":global.get('t'),\"soil\":global.get( 's')};\nreturn
msg;",
"outputs":1,
"noerr":0,

"initialize":"",
"finalize":"
","li
bs
":[
],
"
x
":
63
0,
"y":500, "wires":[["5c7996d53a445412"]]
},
{
"id":"5c7996d53a4454
12",
"type":"httpresponse",
"z":"03acb6ae05a0c71
2","name":"",
"statusCode":"",
"header
s":{},
"x":870,
"y":500,
"wires":[]
},
{
"id":"ef745d48e395ccc0",
"type":"ibmiot",
"name":"weather_monitor",
"keepalive":"60",
"serverName":"",
"cleansession":true,
"appId":"",
"shared":false},
{
"id":"f4cb8513b95c98a4","type":"ui_gro
up","name":"monitor",
"tab":"1f4cb829.2fde
e8","order":2,
"disp":
true,
"widt
h
":"6",

"collapse":
false,
"classNam
e":""
},
{
"id":"1f4cb829.2fdee8",
"type":"ui_tab",
"name":"Home",
"icon":"dashboar
d","order":3,
"disabled":false,
"hidden":false }
```

The source code has been uploaded in git hub.

SOURCE CODE & DEMO VIDEO