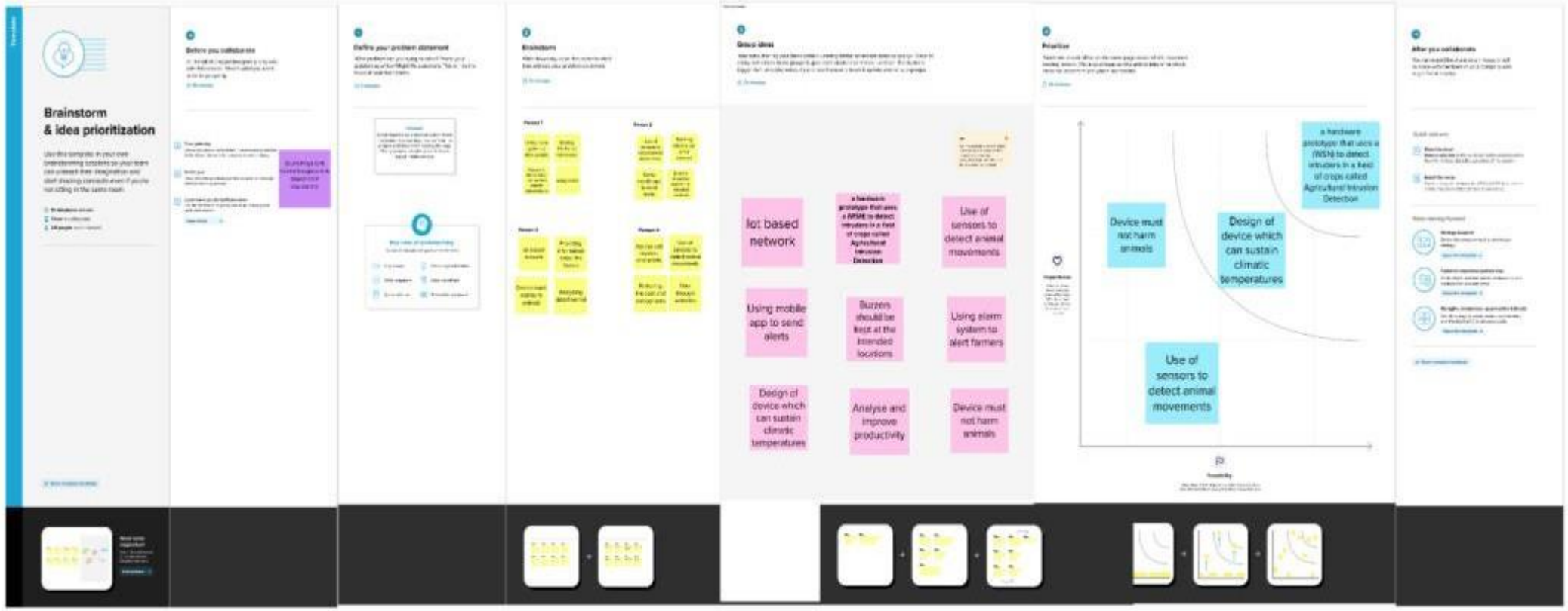
# IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

**MUTHAYAMMAL ENGINEERING COLLEGE**
**TEAM ID** : PNT2022TMID18899
**TEAM LEAD** :DHIVYA V
**TEAM MEMBERS**:
Dinesh M
DineshKumar N
Divyabharathi S

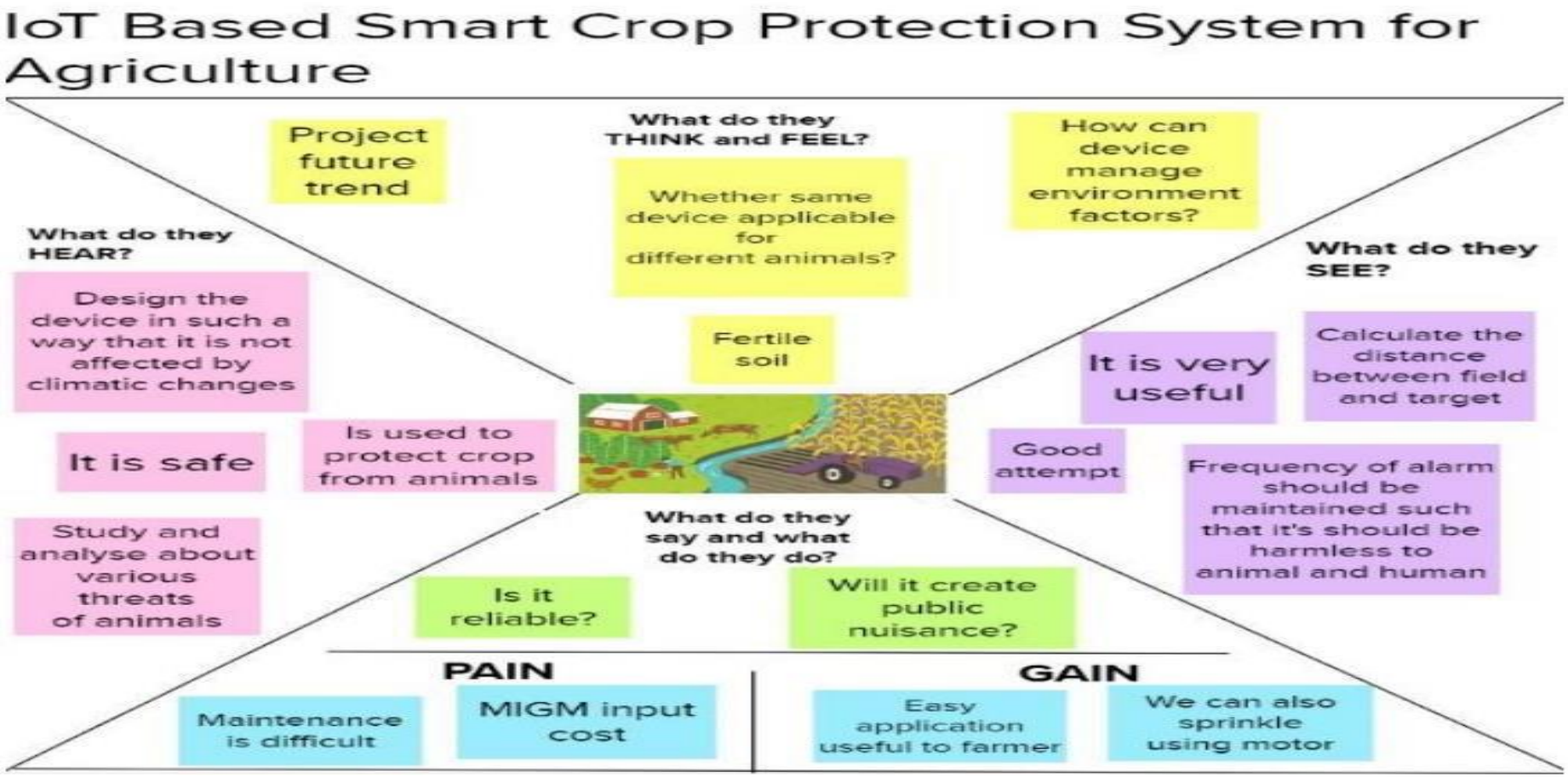
# PROJECT DESCRIPTION

- 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.

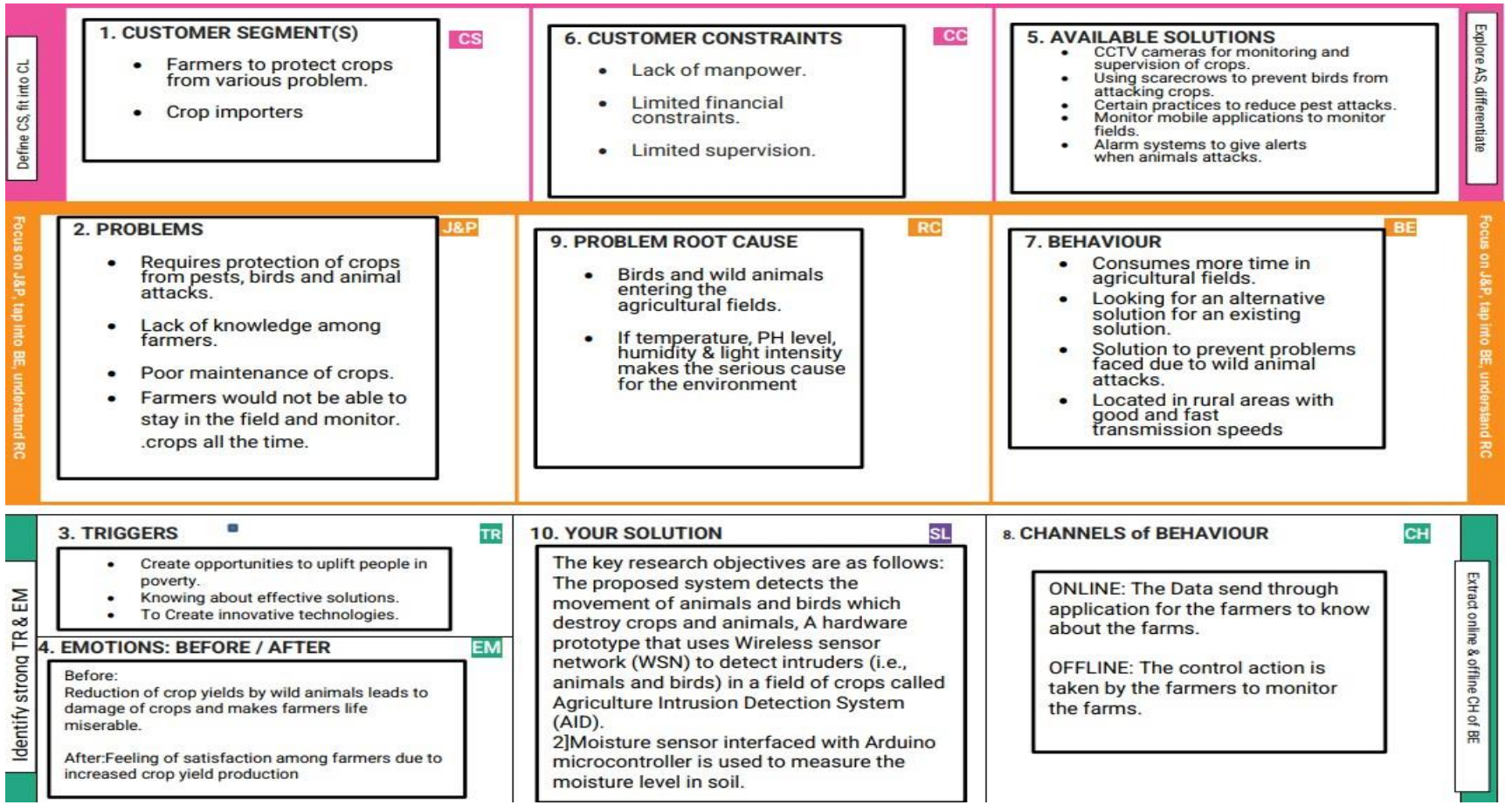
# IDEATION & BRAINSTROMING



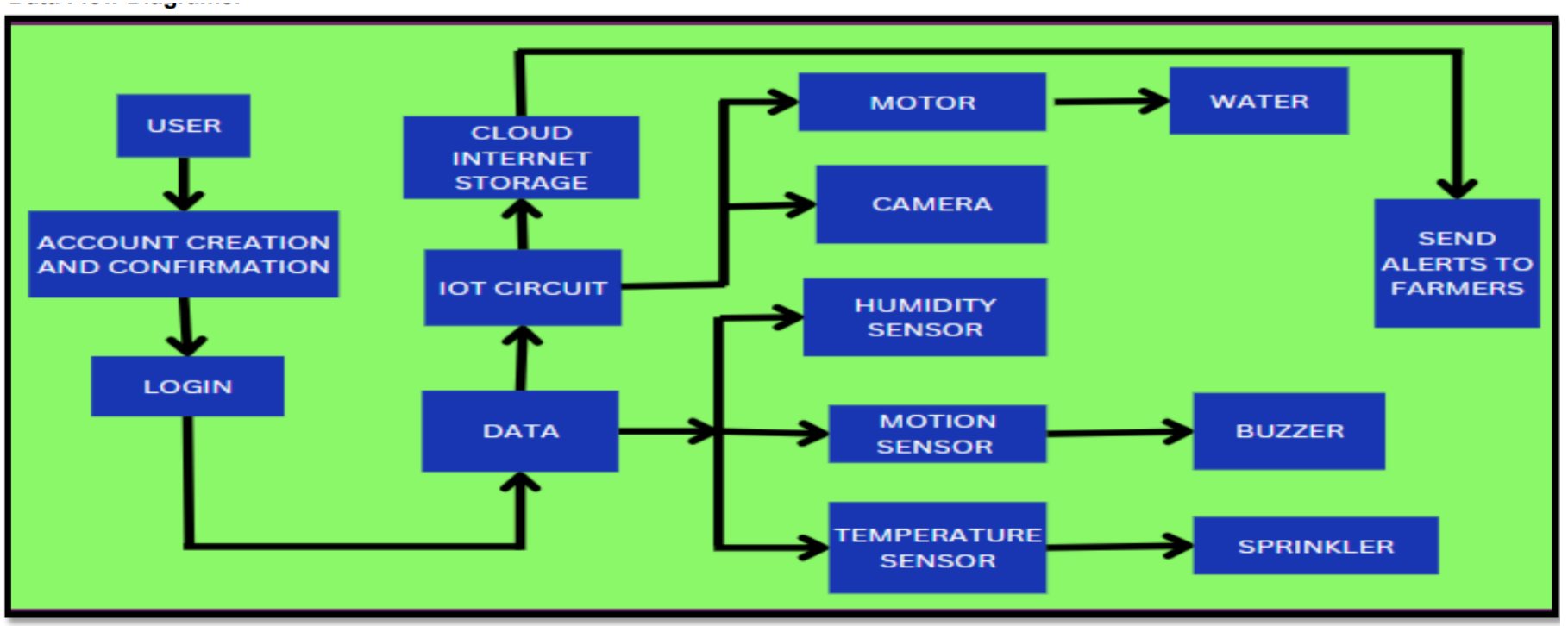
# EMPATHY MAP CANVAS



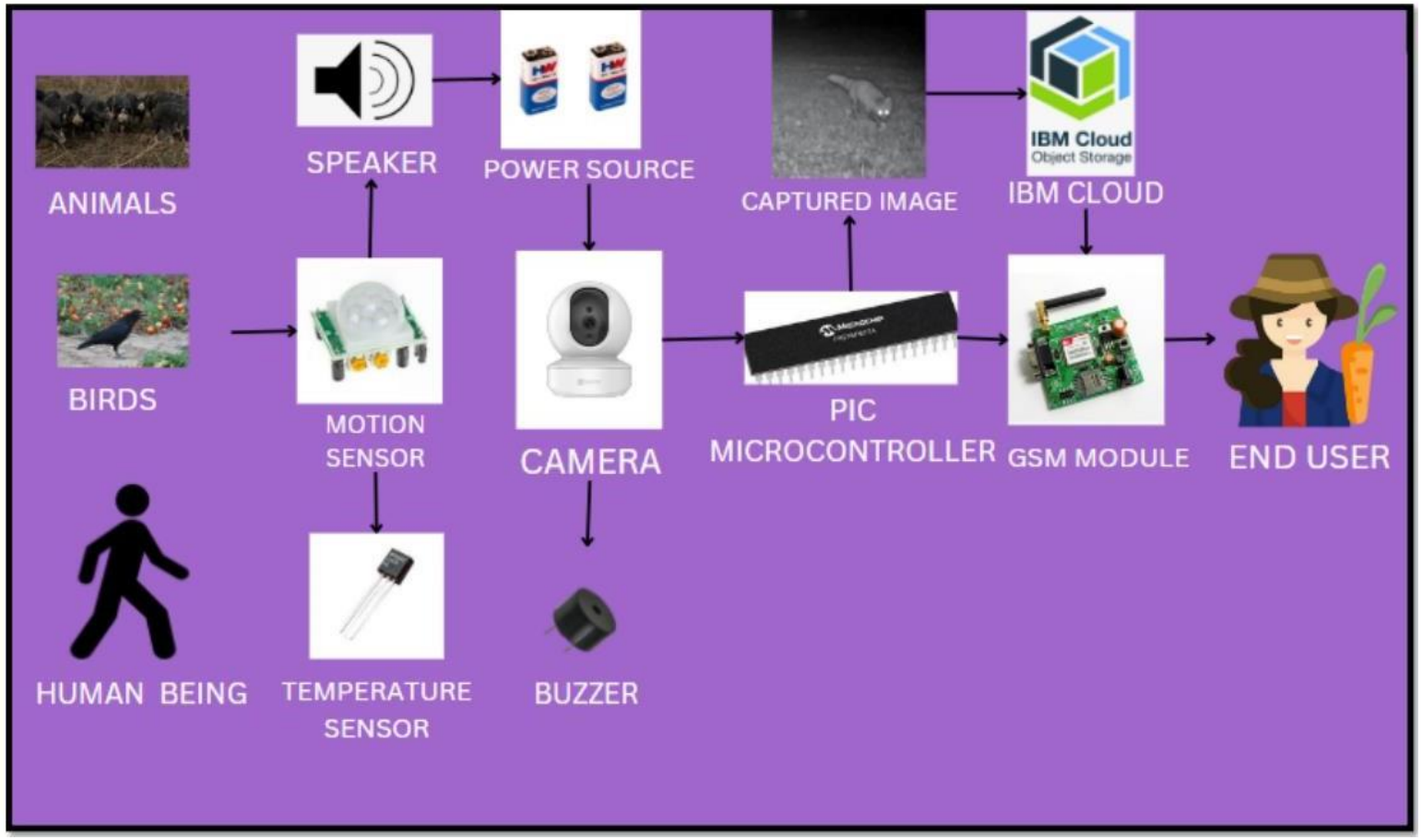
# PROPOSED SOLUTION FIT



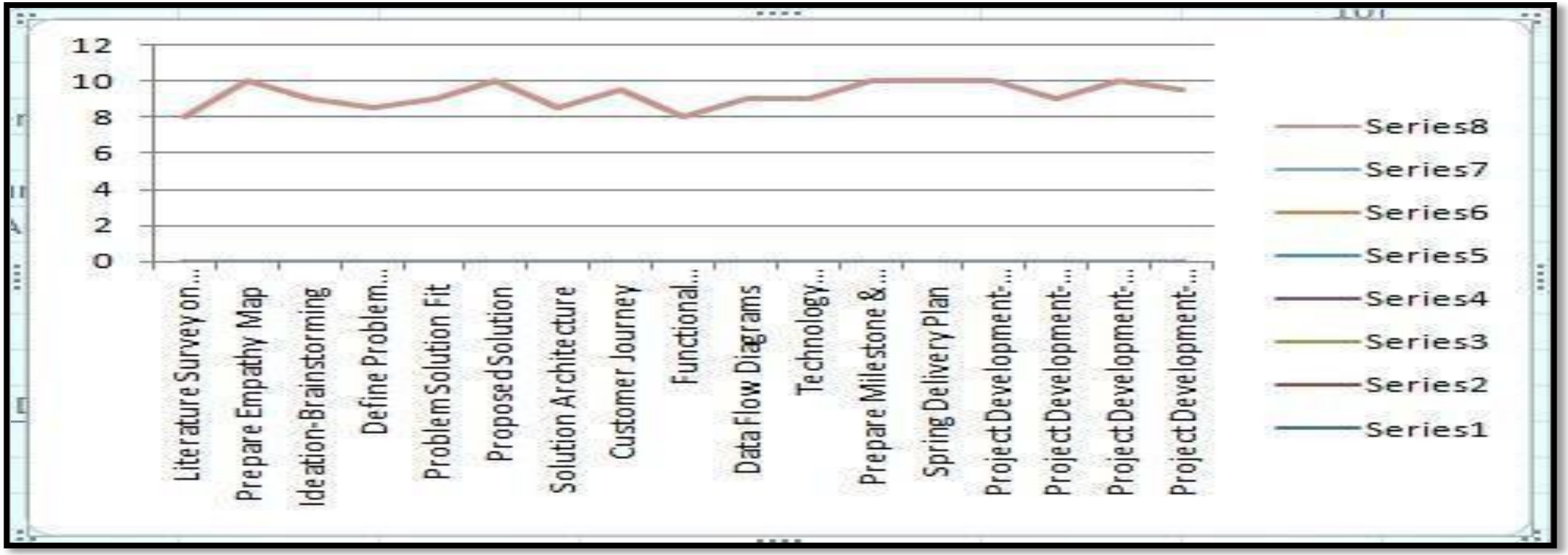
# DATA FLOW DIAGRAMS



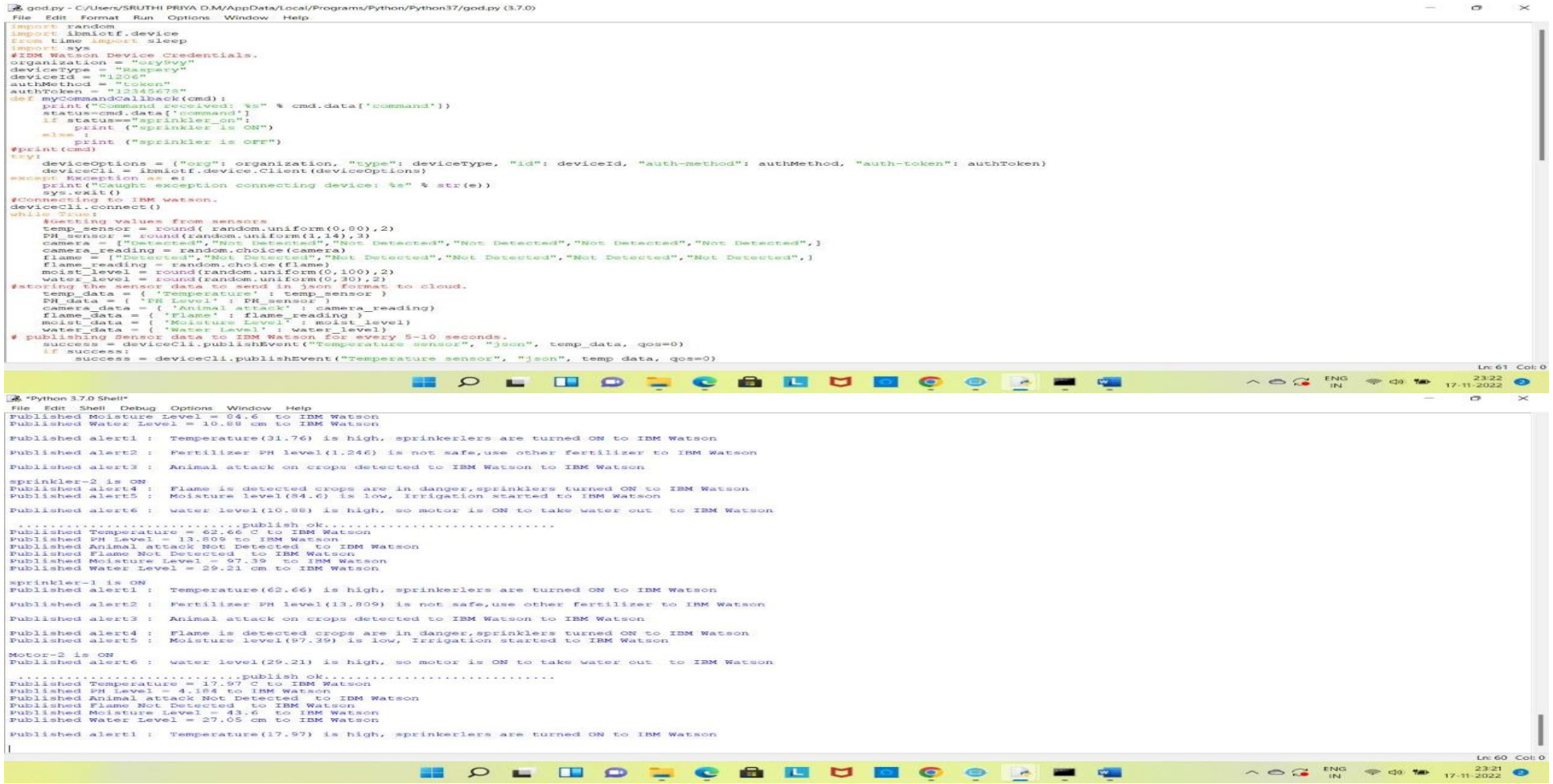
# SOLUTION & TECHNICAL ARCHITECTURE



# PROJECT PLANNING & SCHEDULING



# CODING & SOLUTIONING(FEATURE 1)

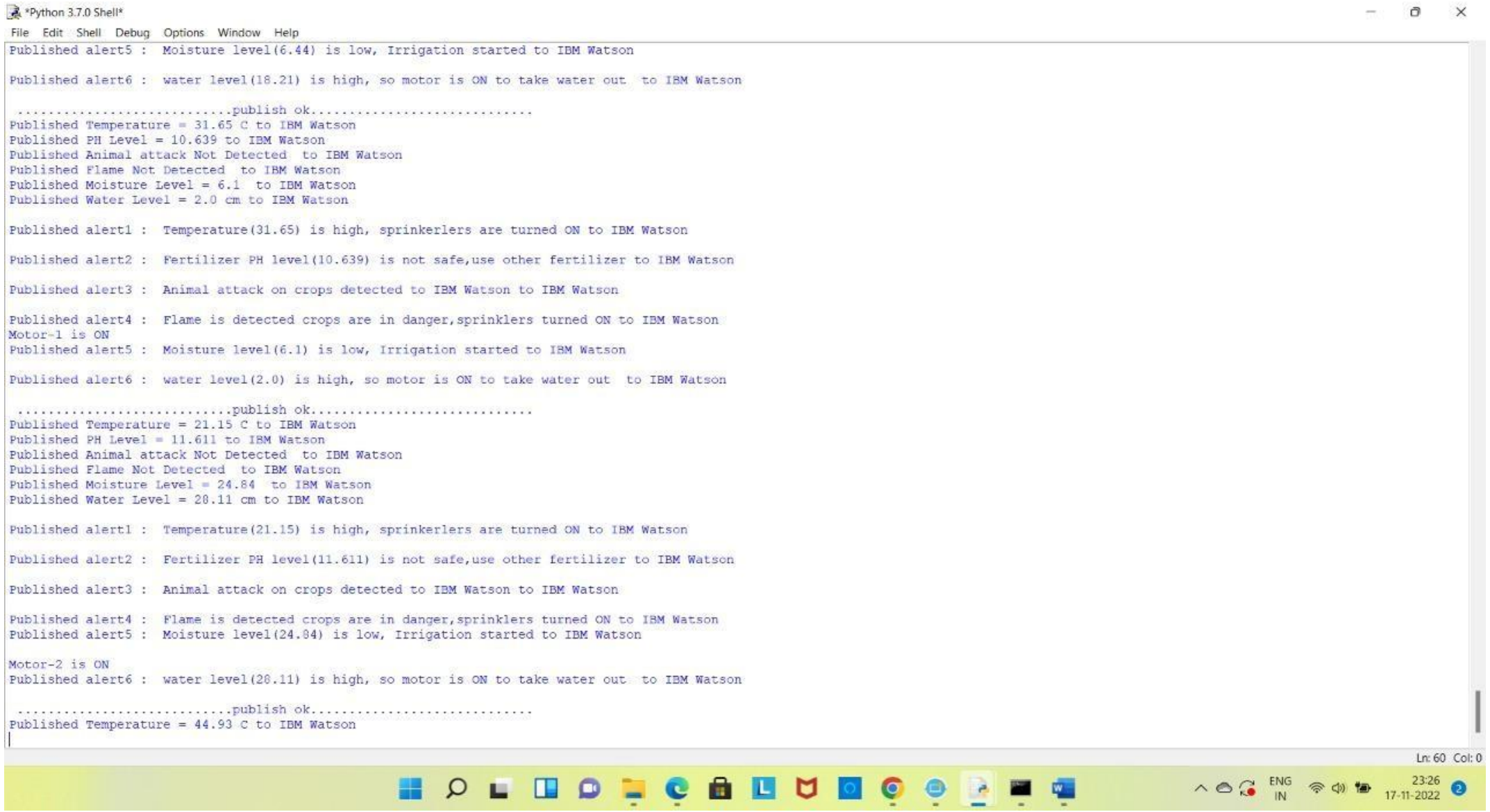


```
god.py - C:/Users/SRUTHI PRIYA D.M/AppData/Local/Programs/Python/Python37/god.py (3.7.0)
File  Edit  Format  Run  Options  Window  Help
import random
import ibmiotf.device
from time import sleep
import sys
#IBM Watson Device Credentials.
organization = "oiy9vy"
deviceType = "Raspery"
deviceId = "1206"
authMethod = "token"
authToken = "12345678"
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",]
    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)
    if success:
        success = deviceCli.publishEvent("Temperature sensor", "json", temp_data, qos=0)
```

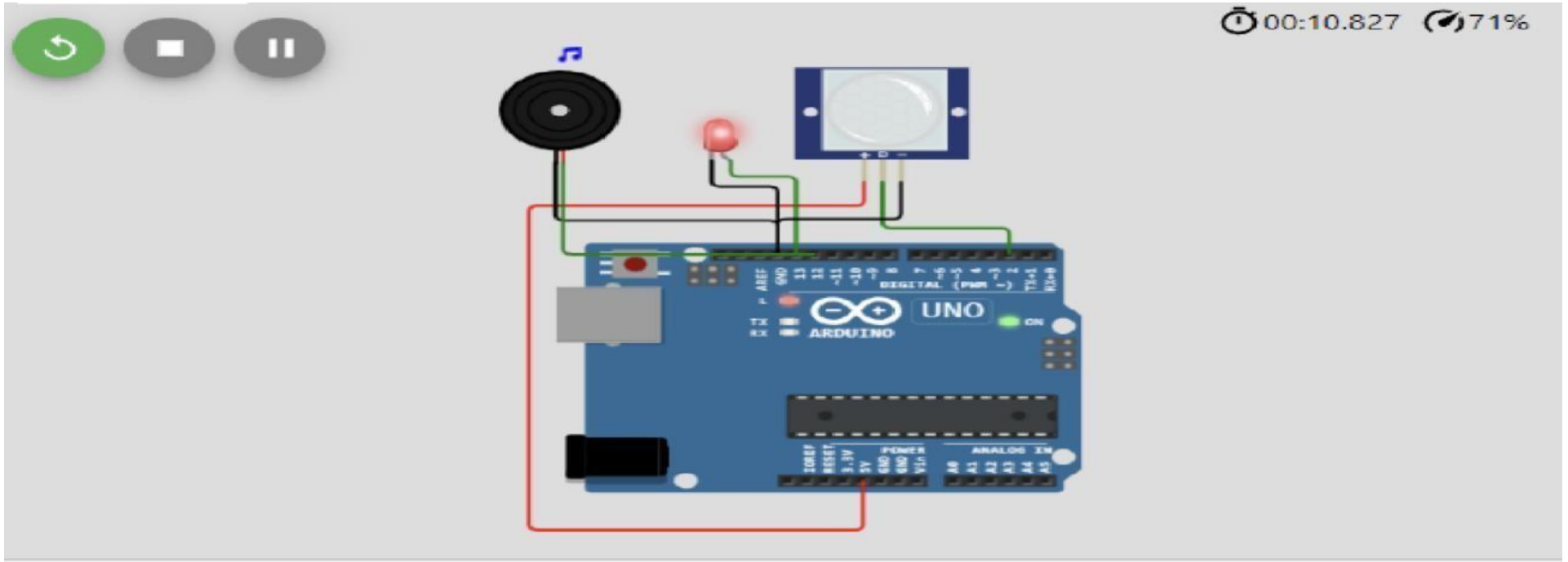
Ln: 61 Col: 0

```
*Python 3.7.0 Shell*
File  Edit  Shell  Debug  Options  Window  Help
Published Moisture Level = 84.6  to IBM Watson
Published Water Level = 10.88 cm to IBM Watson

Published alert1 :  Temperature(31.76) is high, sprinkerlers are turned ON to IBM Watson

Published alert2 :  Fertilizer PH level(1.246) is not safe,use other fertilizer to IBM Watson

Published alert3 :  Animal attack on crops detected to IBM Watson to IBM Watson

sprinkler-2 is ON
Published alert4 :  Flame is detected crops are in danger,sprinklers turned ON to IBM Watson
Published alert5 :  Moisture level(84.6) is low, Irrigation started to IBM Watson

Published alert6 :  water level(10.88) is high, so motor is ON to take water out  to IBM Watson

  ..............................publish ok..............................
Published Temperature = 62.66 C to IBM Watson
Published PH Level = 13.809 to IBM Watson
Published Animal attack Not Detected  to IBM Watson
Published Flame Not Detected  to IBM Watson
Published Moisture Level = 97.39  to IBM Watson
Published Water Level = 29.21 cm to IBM Watson

sprinkler-1 is ON
Published alert1 :  Temperature(62.66) is high, sprinkerlers are turned ON to IBM Watson

Published alert2 :  Fertilizer PH level(13.809) is not safe,use other fertilizer to IBM Watson

Published alert3 :  Animal attack on crops detected to IBM Watson to IBM Watson

Published alert4 :  Flame is detected crops are in danger,sprinklers turned ON to IBM Watson
Published alert5 :  Moisture level(97.39) is low, Irrigation started to IBM Watson

Motor-2 is ON
Published alert6 :  water level(29.21) is high, so motor is ON to take water out  to IBM Watson

  ..............................publish ok..............................
Published Temperature = 17.97 C to IBM Watson
Published PH Level = 4.184 to IBM Watson
Published Animal attack Not Detected  to IBM Watson
Published Flame Not Detected  to IBM Watson
Published Moisture Level = 43.6  to IBM Watson
Published Water Level = 27.05 cm to IBM Watson

Published alert1 :  Temperature(17.97) is high, sprinkerlers are turned ON to IBM Watson
```

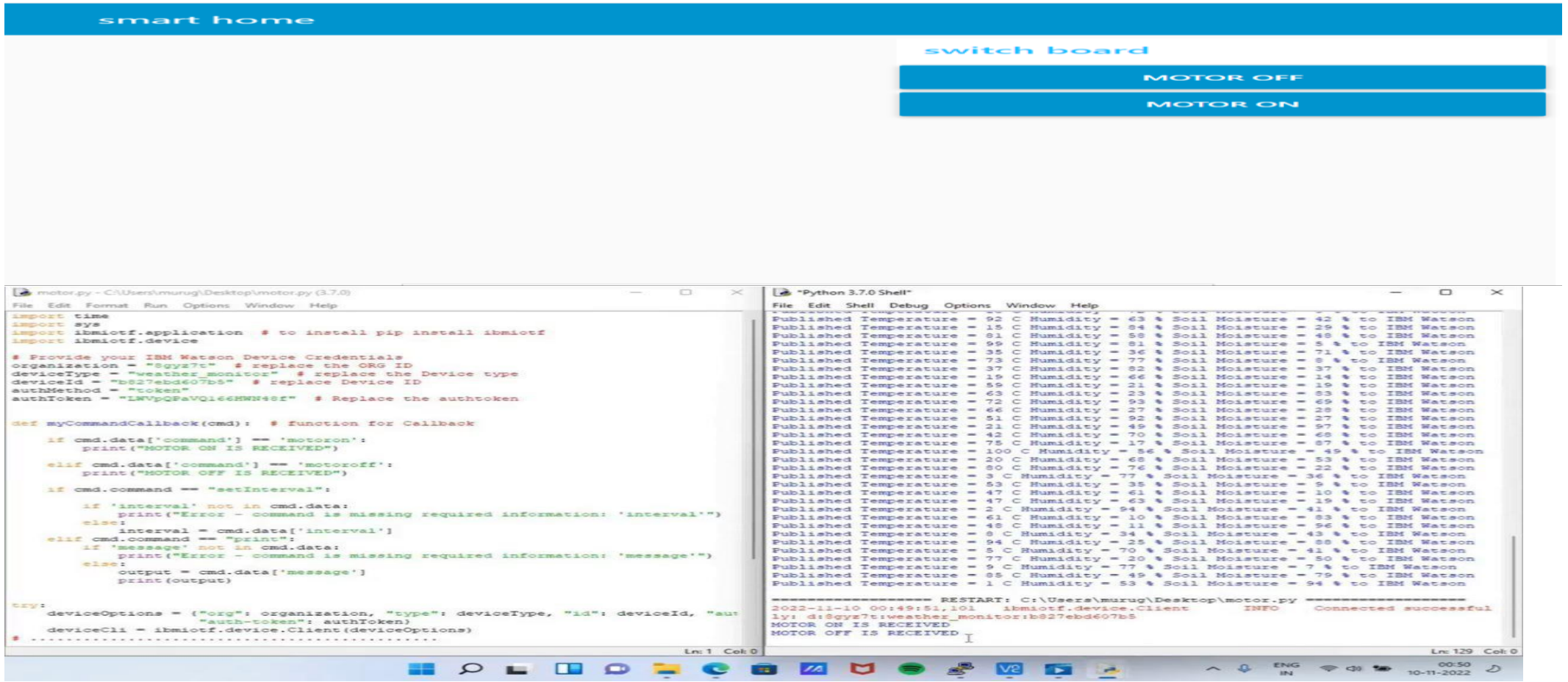
Ln: 60 Col: 0


*Python 3.7.0 Shell*
File Edit Shell Debug Options Window Help
Published alert5 : Moisture level(6.44) is low, Irrigation started to IBM Watson
Published alert6 : water level(18.21) is high, so motor is ON to take water out to IBM Watson
.....publish ok.....
Published Temperature = 31.65 C to IBM Watson
Published PH Level = 10.639 to IBM Watson
Published Animal attack Not Detected to IBM Watson
Published Flame Not Detected to IBM Watson
Published Moisture Level = 6.1 to IBM Watson
Published Water Level = 2.0 cm to IBM Watson
Published alert1 : Temperature(31.65) is high, sprinkerlers are turned ON to IBM Watson
Published alert2 : Fertilizer PH level(10.639) is not safe,use other fertilizer to IBM Watson
Published alert3 : Animal attack on crops detected to IBM Watson to IBM Watson
Published alert4 : Flame is detected crops are in danger,sprinklers turned ON to IBM Watson
Motor-1 is ON
Published alert5 : Moisture level(6.1) is low, Irrigation started to IBM Watson
Published alert6 : water level(2.0) is high, so motor is ON to take water out to IBM Watson
.....publish ok.....
Published Temperature = 21.15 C to IBM Watson
Published PH Level = 11.611 to IBM Watson
Published Animal attack Not Detected to IBM Watson
Published Flame Not Detected to IBM Watson
Published Moisture Level = 24.84 to IBM Watson
Published Water Level = 20.11 cm to IBM Watson
Published alert1 : Temperature(21.15) is high, sprinkerlers are turned ON to IBM Watson
Published alert2 : Fertilizer PH level(11.611) is not safe,use other fertilizer to IBM Watson
Published alert3 : Animal attack on crops detected to IBM Watson to IBM Watson
Published alert4 : Flame is detected crops are in danger,sprinklers turned ON to IBM Watson
Published alert5 : Moisture level(24.84) is low, Irrigation started to IBM Watson
Motor-2 is ON
Published alert6 : water level(20.11) is high, so motor is ON to take water out to IBM Watson
.....publish ok.....
Published Temperature = 44.93 C to IBM Watson
Ln: 60 Col: 0
ENG
IN
23:26
17-11-2022

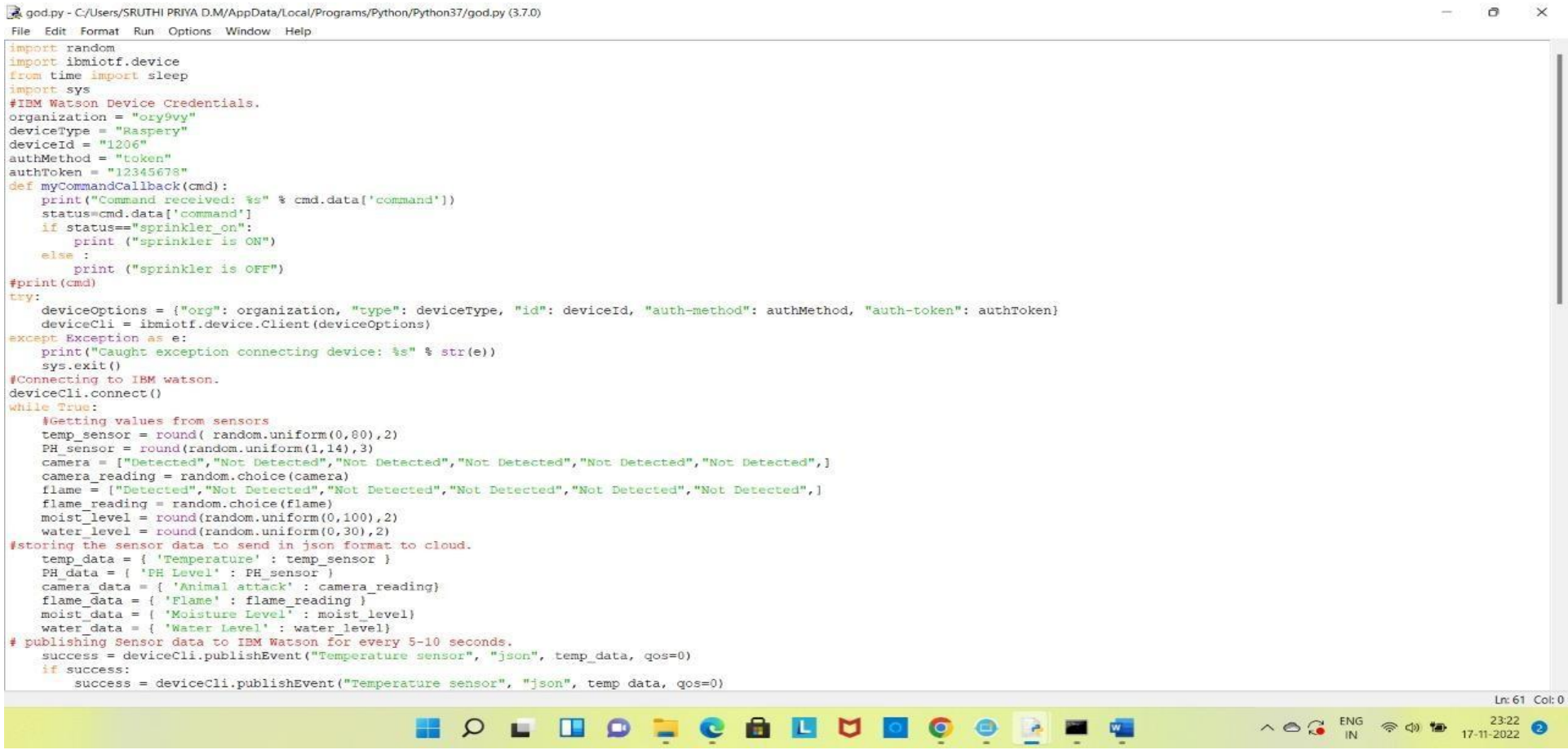
# CODING & SOLUTIONING(FEATURE 2)



# CODING & SOLUTIONING(FEATURE 3)



# PERFORMANCE METRICS



*Python 3.7.0 Shell*

File Edit Shell Debug Options Window Help

```
Published Moisture Level = 84.6  to IBM Watson
Published Water Level = 10.88 cm to IBM Watson

Published alert1 :  Temperature(31.76) is high, sprinkerlers are turned ON to IBM Watson

Published alert2 :  Fertilizer PH level(1.246) is not safe,use other fertilizer to IBM Watson

Published alert3 :  Animal attack on crops detected to IBM Watson to IBM Watson

sprinkler-2 is ON
Published alert4 :  Flame is detected crops are in danger,sprinklers turned ON to IBM Watson
Published alert5 :  Moisture level(84.6) is low, Irrigation started to IBM Watson

Published alert6 :  water level(10.88) is high, so motor is ON to take water out  to IBM Watson

 ............................publish ok............................
Published Temperature = 62.66 C to IBM Watson
Published PH Level = 13.809 to IBM Watson
Published Animal attack Not Detected  to IBM Watson
Published Flame Not Detected  to IBM Watson
Published Moisture Level = 97.39  to IBM Watson
Published Water Level = 29.21 cm to IBM Watson

sprinkler-1 is ON
Published alert1 :  Temperature(62.66) is high, sprinkerlers are turned ON to IBM Watson

Published alert2 :  Fertilizer PH level(13.809) is not safe,use other fertilizer to IBM Watson

Published alert3 :  Animal attack on crops detected to IBM Watson to IBM Watson

Published alert4 :  Flame is detected crops are in danger,sprinklers turned ON to IBM Watson
Published alert5 :  Moisture level(97.39) is low, Irrigation started to IBM Watson

Motor-2 is ON
Published alert6 :  water level(29.21) is high, so motor is ON to take water out  to IBM Watson

 ............................publish ok............................
Published Temperature = 17.97 C to IBM Watson
Published PH Level = 4.184 to IBM Watson
Published Animal attack Not Detected  to IBM Watson
Published Flame Not Detected  to IBM Watson
Published Moisture Level = 43.6  to IBM Watson
Published Water Level = 27.05 cm to IBM Watson

Published alert1 :  Temperature(17.97) is high, sprinkerlers are turned ON to IBM Watson
```

Ln: 60 Col: 0

ENG IN 23:21 17-11-2022

# SOURCE CODE

**FINAL CODE**

**TEAM ID** : PNT2022TMID18899

**PROJECT NAME** : IOT Based Smart crop protection for Agriculture

```
import time
import sys
import ibmiotf.application # to install pip install ibmiotf
import ibmiotf.device

#Provide your IBM Watson Device Credentials
organization = "hrodmj" #replace the ORG ID
deviceType = "NODEMCU1"#replace the Device type wi
deviceId = "12345"#replace Device ID
authMethod = "token"
authToken = "kp1234" #Replace the authtoken

def myCommandCallback(cmd): # function for Callback
    print("Command received: %s" % cmd.data)
    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)
        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()
```

# CONCLUSION

- This system focuses on developing devices and tool to manage, display and alert 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 implementan IoT system in order to help farmers to control and monitor their farms has been implemented successfully.


Thank
you