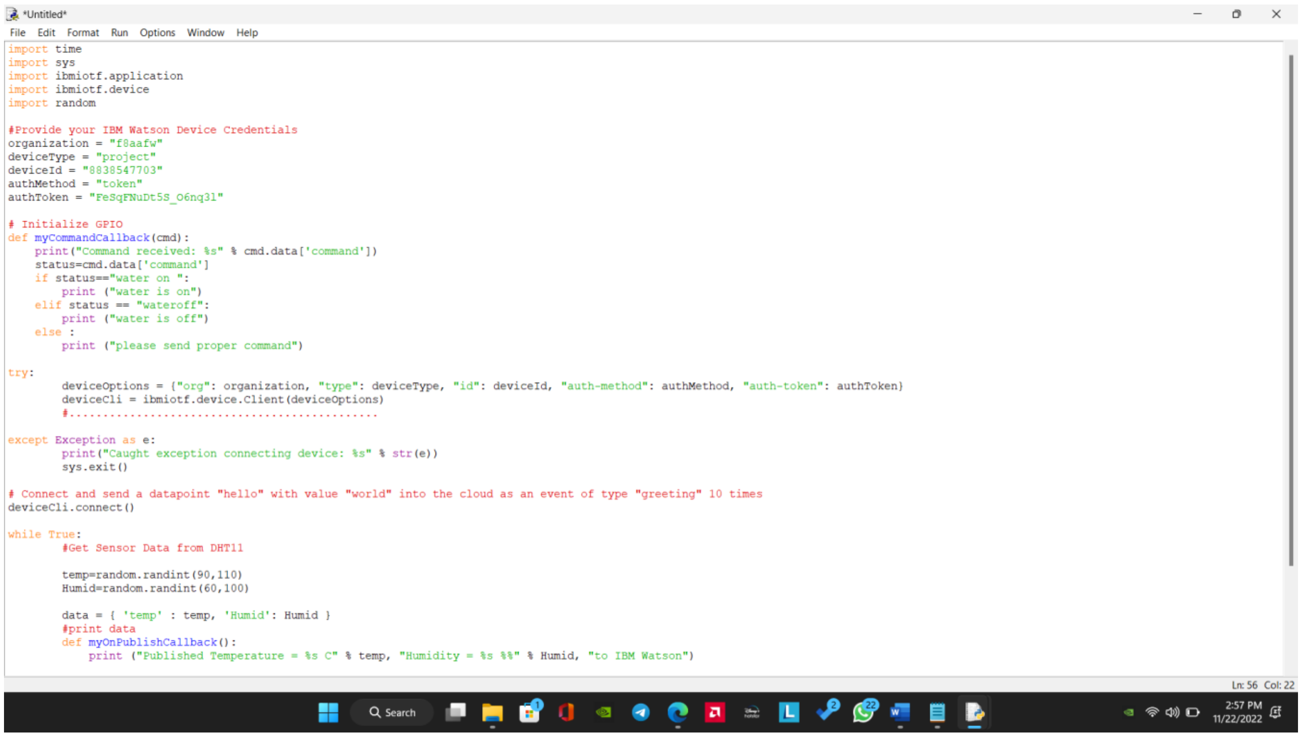
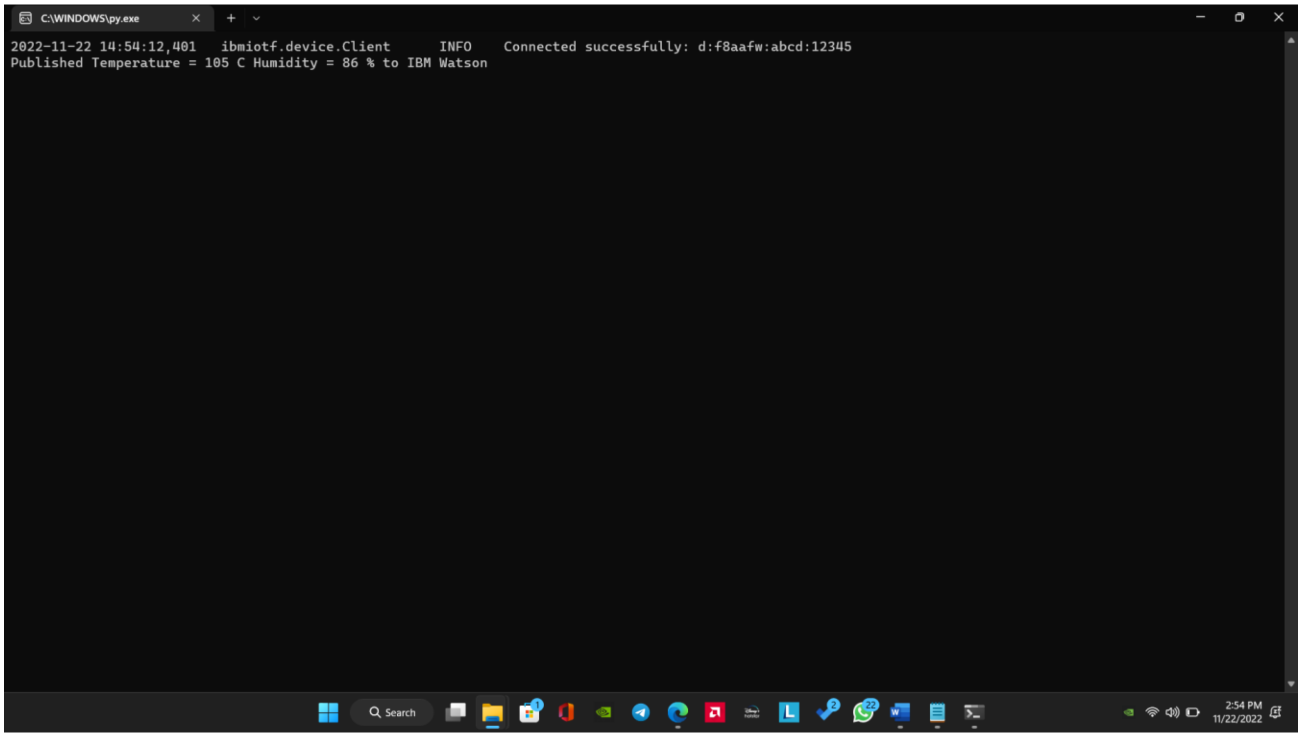
SMART FARMING – AN IOT ENABLED SMART FARMING APPLICATION
SPRINT -4
TEAMID: PNT2022TMID03145
PROJECT DEVELOPMENT PHASE

Receiving commands from IBM cloud

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random


#Provide your IBM Watson Device Credentials

organization = "f8aafw"

deviceType = "project"

deviceId = "8838547703"

authMethod = "token"

authToken = "FeSqFNuDt5S_O6nq3l"


# Initialize GPIO

def myCommandCallback(cmd):

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

   status=cmd.data['command']

   if status=="water on ":

      print ("water is on")

   elif status == "wateroff":
        print ("water 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)


        data = { 'temp' : temp, 'Humid': Humid }
        #print data
        def myOnPublishCallback():
          print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, 
"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 cloud
deviceCli.disconnect()
```

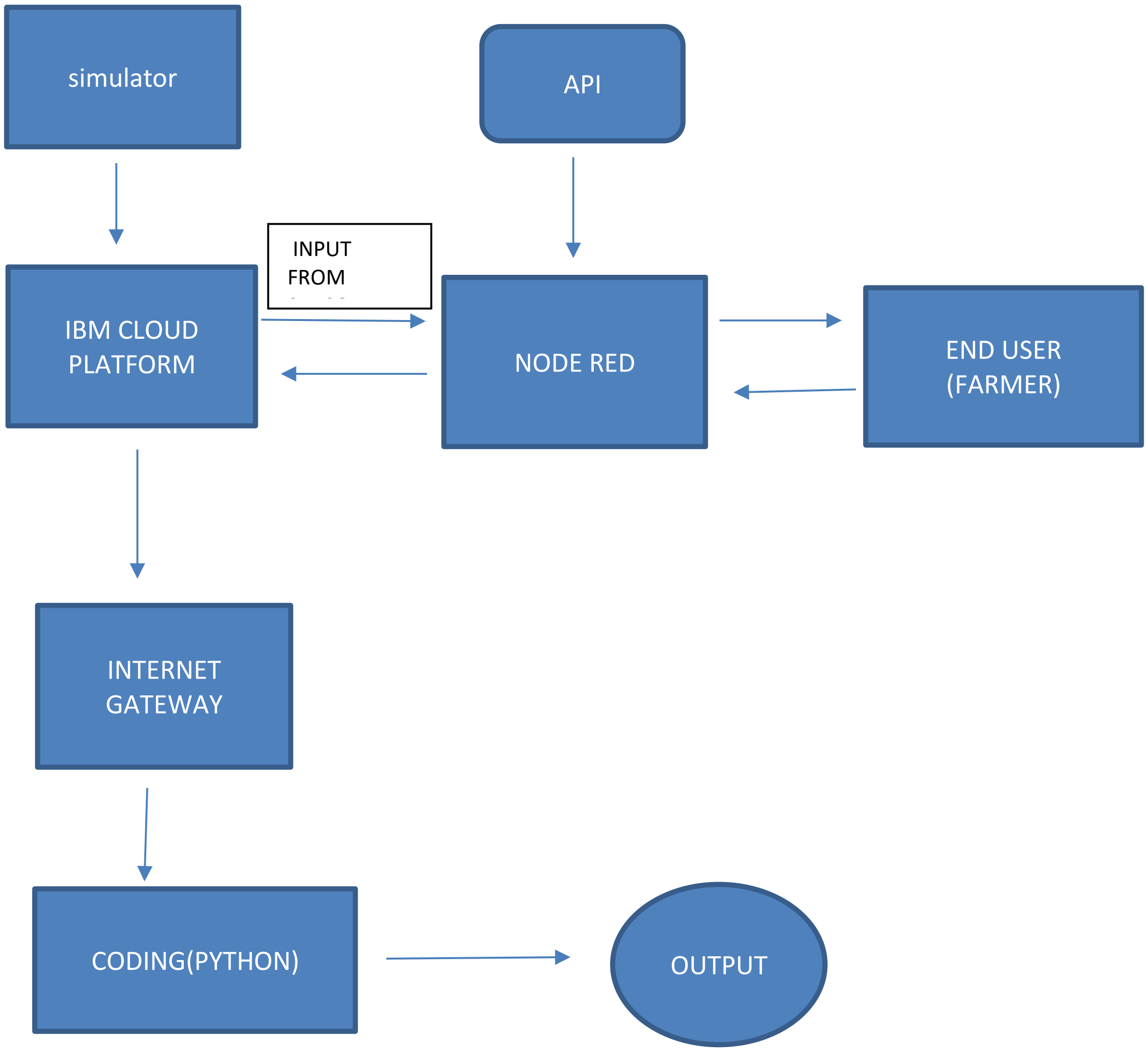


```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "f8aafw"
deviceType = "project"
deviceId = "8838547703"
authMethod = "token"
authToken = "FeSqFNuDt5S_O6nq3l"

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

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

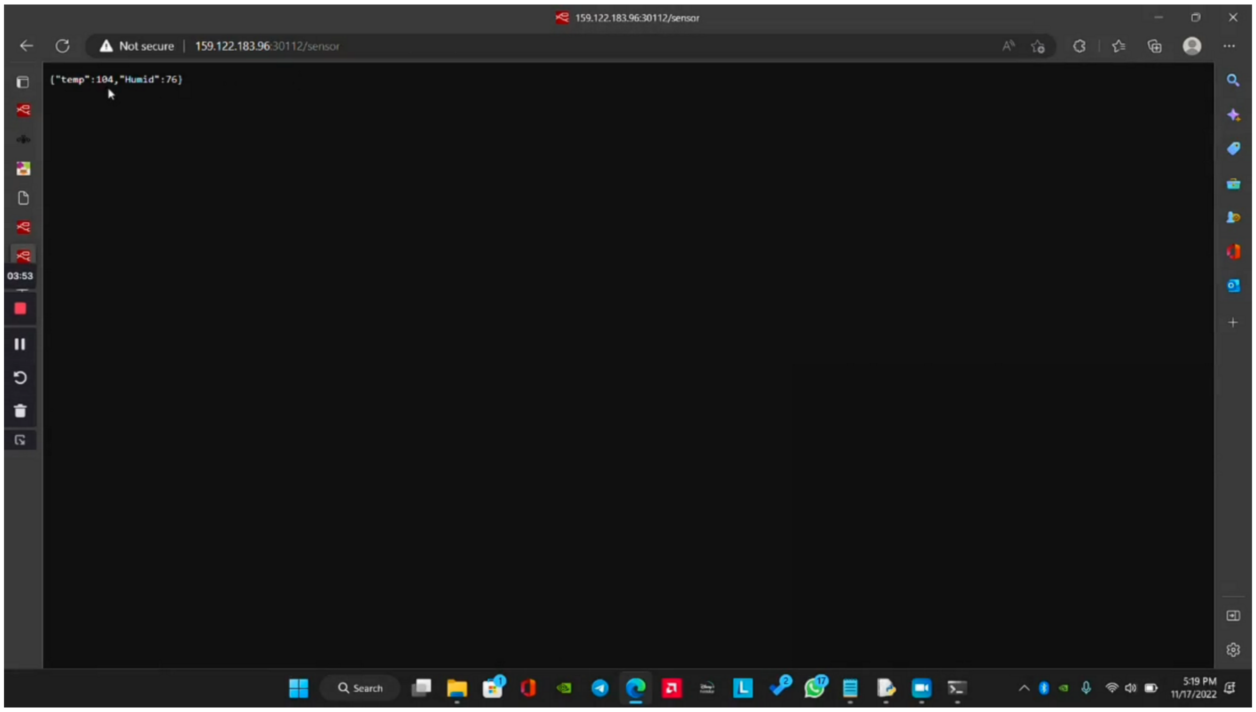


```
2022-11-22 14:54:12,401   ibmiotf.device.Client      INFO    Connected successfully: d:f8aafw:abcd:12345
Published Temperature = 105 C Humidity = 86 % to IBM Watson
```

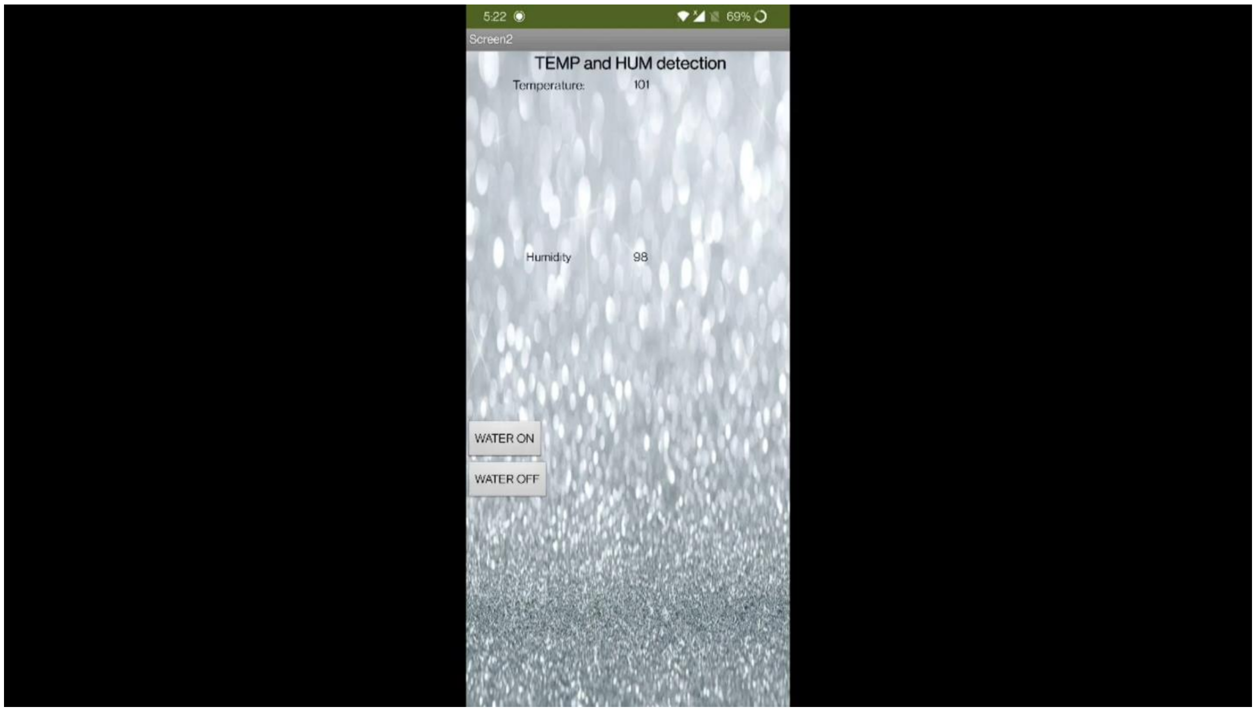
6. Flow Chart


simulator
API
INPUT FROM
IBM CLOUD PLATFORM
NODE RED
END USER (FARMER)
INTERNET GATEWAY
CODING(PYTHON)
OUTPUT

7.Observations & Results



```
{"temp":104,"Humid":76}
```



TEMP and HUM detection

Temperature: 101

Humidity 98

<u>8. Advantages:</u>

- Farms can be monitored and controlled remotely.
- Increase in convenience to farmers.
- Less labor cost.
- 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 Mobile App.

9.**Conclusion**

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.

10.Bibliography

IBM cloud reference: https://cloud.ibm.com/

IoT simulator : https://watson-iot-sensor-simulator.mybluemix.net/

OpenWeather : https://openweathermap.org/