

# PROJECT DEVELOPMENT PHASE

TEAM ID	PNT2022TMID7601
PROJECT TITLE	IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

## SPRINT 1

## SIMULATION CREATION:

```
iot.py - C:\Users\lagaram\Desktop\ECE 5\iot.py (3.7.3)
File Edit Format Run Options Window Help

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

#IBM Watson Device Credentials.
organization = "k94aj1"
deviceType = "iotpjtdevicetype"
deviceId = "11121315"
authMethod = "token"
authToken = ")_QozcIqy2G9Q5g9Pef"

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)
    camera = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected",]
    camera_reading = random.choice(camera)
    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 }
    camera_data = { 'Animal attack' : camera_reading}
    moist_data = { 'Moisture Level' : moist_level}
```

```
iot.py - C:\Users\agaram\Desktop\ECE 5\c\iot.py (3.7.3)
File Edit Format Run Options Window Help

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("camera", "json", camera_data, qos=0)
sleep(1)
if success:
    print ("Published Animal attack %s " % camera_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 ("")

#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 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)
```

```
iot.py - C:\Users\agaram\Desktop\ECE 5\c\iot.py (3.7.3)
File Edit Format Run Options Window Help

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 ("")

#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 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 received by farmer
deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

OUTPUT:

The screenshot shows the 'Recent Events' tab for a device named 'iotpitdevicetype\_1'. The device status is 'Connected'. Below the tabs, there is a message: 'The recent events listed show the live stream of data that is coming and going from this device.' A table displays five recent events, each containing temperature, humidity, and moisture data in JSON format.

Event	Value	Format	Last Received
event_1	{"Temperature":22,"Humidity":97,"Moisture":28}	json	a few seconds ago
event_1	{"Temperature":29,"Humidity":76,"Moisture":2}	json	a few seconds ago
event_1	{"Temperature":27,"Humidity":41,"Moisture":85}	json	a few seconds ago
event_1	{"Temperature":40,"Humidity":74,"Moisture":23}	json	a few seconds ago
event_1	{"Temperature":81,"Humidity":25,"Moisture":92}	json	a few seconds ago

At the bottom right, it indicates '2 Simulations running'.