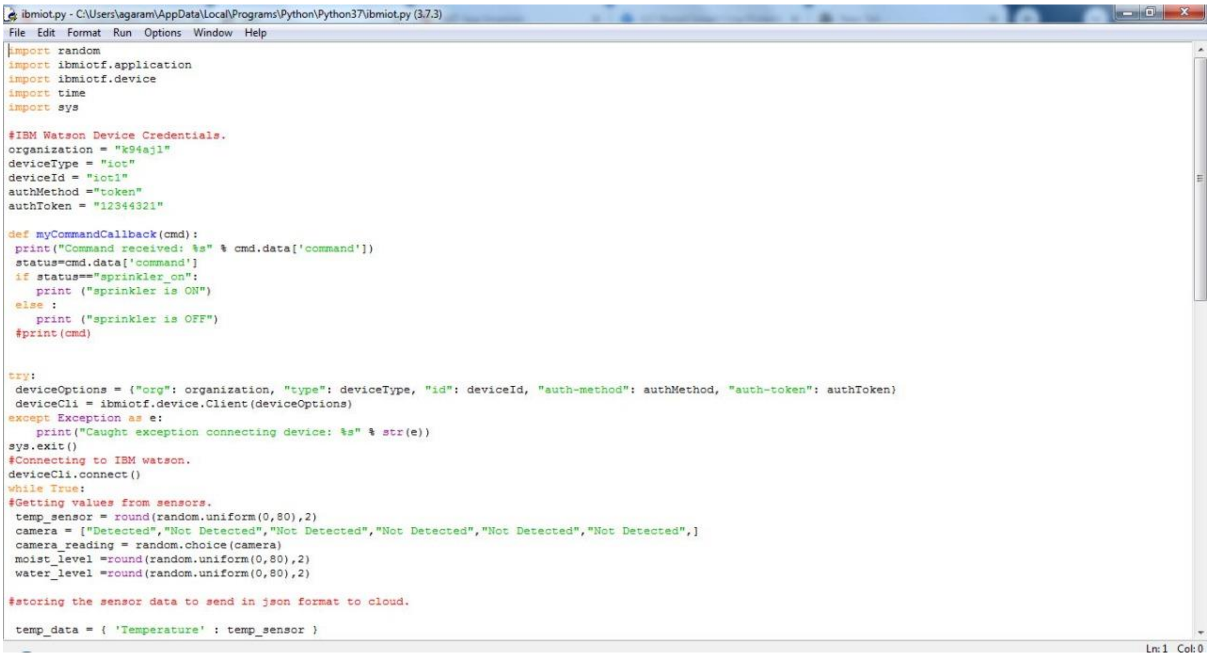
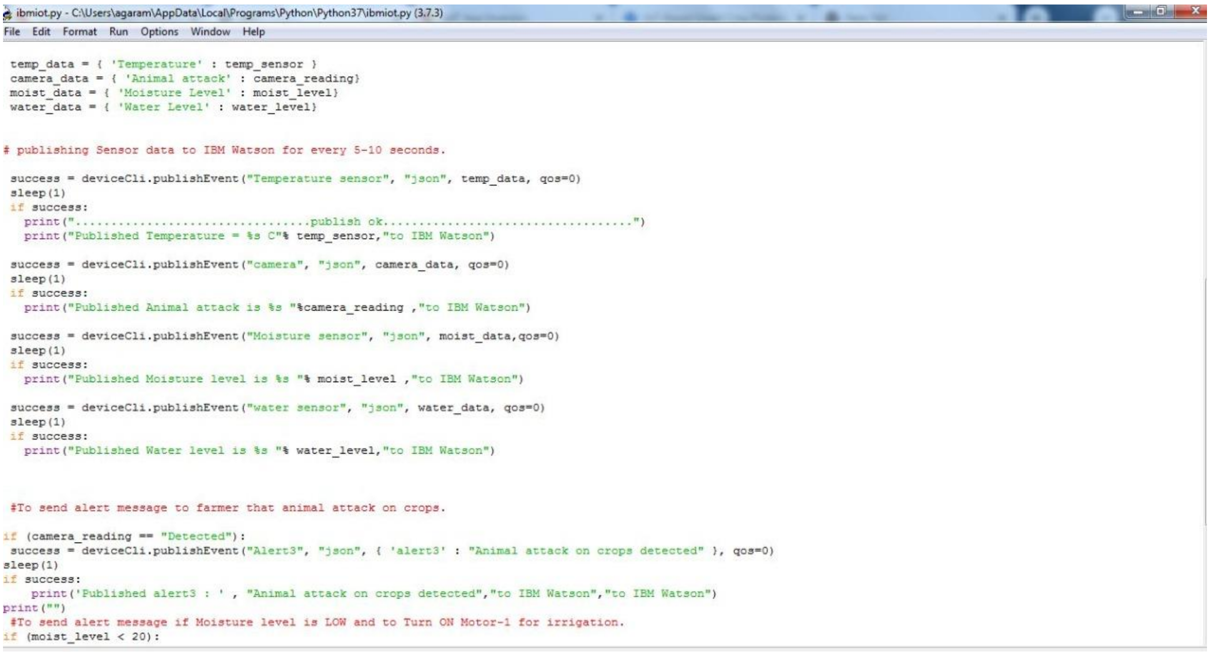
DEVELOP THE PYTHON SCRIPT

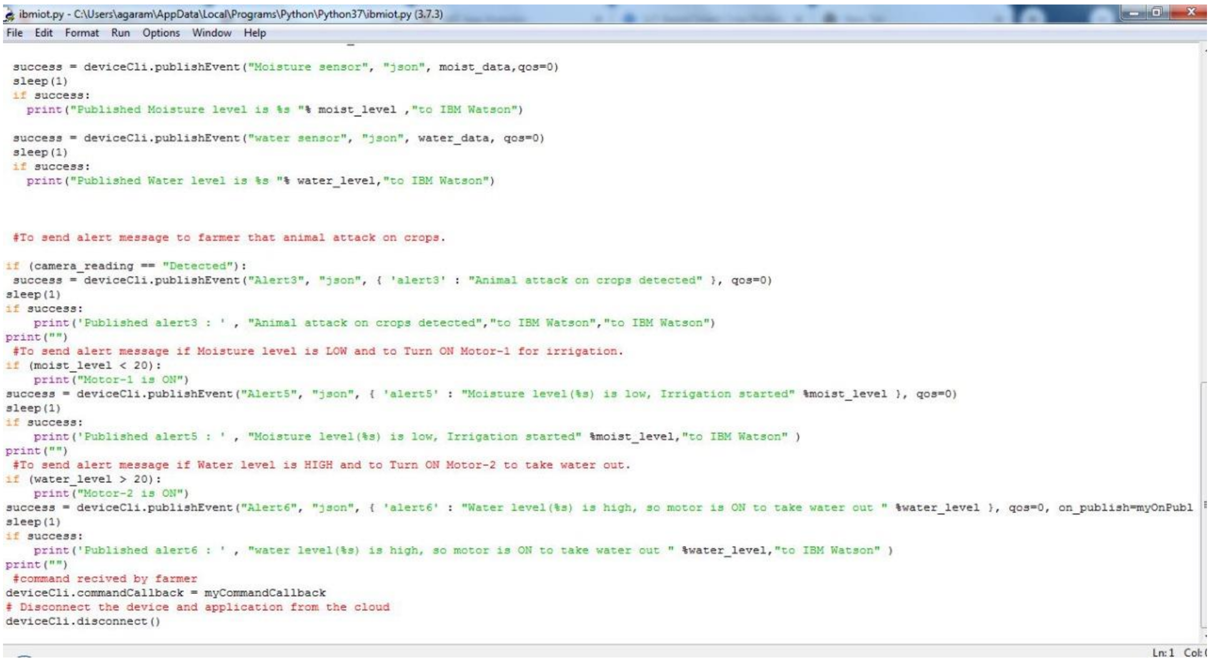
TEAM ID	PNT2022TMID34905
PROJECT TITLE	IOT BASED SMART CROP PROTECTION SYTEM FOR AGRICULTURE


ibmiot.py - C:\Users\agaram\AppData\Local\Programs\Python\Python37\ibmiot.py (3.7.3)
File Edit Format Run Options Window Help
Ln: 1 Col: 0

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

#IBM Watson Device Credentials.
organization = "k94ajl"
deviceType = "iot"
deviceId = "iot1"
authMethod ="token"
authToken = "12344321"

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","Not Detected",]
 camera_reading = random.choice(camera)
 moist_level =round(random.uniform(0,80),2)
 water_level =round(random.uniform(0,80),2)

#storing the sensor data to send in json format to cloud.

 temp_data = { 'Temperature' : temp_sensor }
```


ibmiot.py - C:\Users\agaram\AppData\Local\Programs\Python\Python37\ibmiot.py (3.7.3)
File Edit Format Run Options Window Help

```
 temp_data = { 'Temperature' : temp_sensor }
 camera_data = { 'Animal attack' : camera_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)
 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 is %s "%camera_reading ,"to IBM Watson")

 success = deviceCli.publishEvent("Moisture sensor", "json", moist_data,qos=0)
 sleep(1)
 if success:
   print("Published Moisture level is %s "% moist_level ,"to IBM Watson")

 success = deviceCli.publishEvent("water sensor", "json", water_data, qos=0)
 sleep(1)
 if success:
   print("Published Water level is %s "% water_level,"to IBM Watson")



 #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):
```



ibmiot.py - C:\Users\agaram\AppData\Local\Programs\Python\Python37\ibmiot.py (3.7.3)
File Edit Format Run Options Window Help
Ln: 1 Col:

```
 success = deviceCli.publishEvent("Moisture sensor", "json", moist_data,qos=0)
 sleep(1)
 if success:
   print("Published Moisture level is %s "% moist_level ,"to IBM Watson")

 success = deviceCli.publishEvent("water sensor", "json", water_data, qos=0)
 sleep(1)
 if success:
   print("Published Water level is %s "% water_level,"to IBM Watson")



 #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, on_publish=myOnPubl
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 recived by farmer
deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```