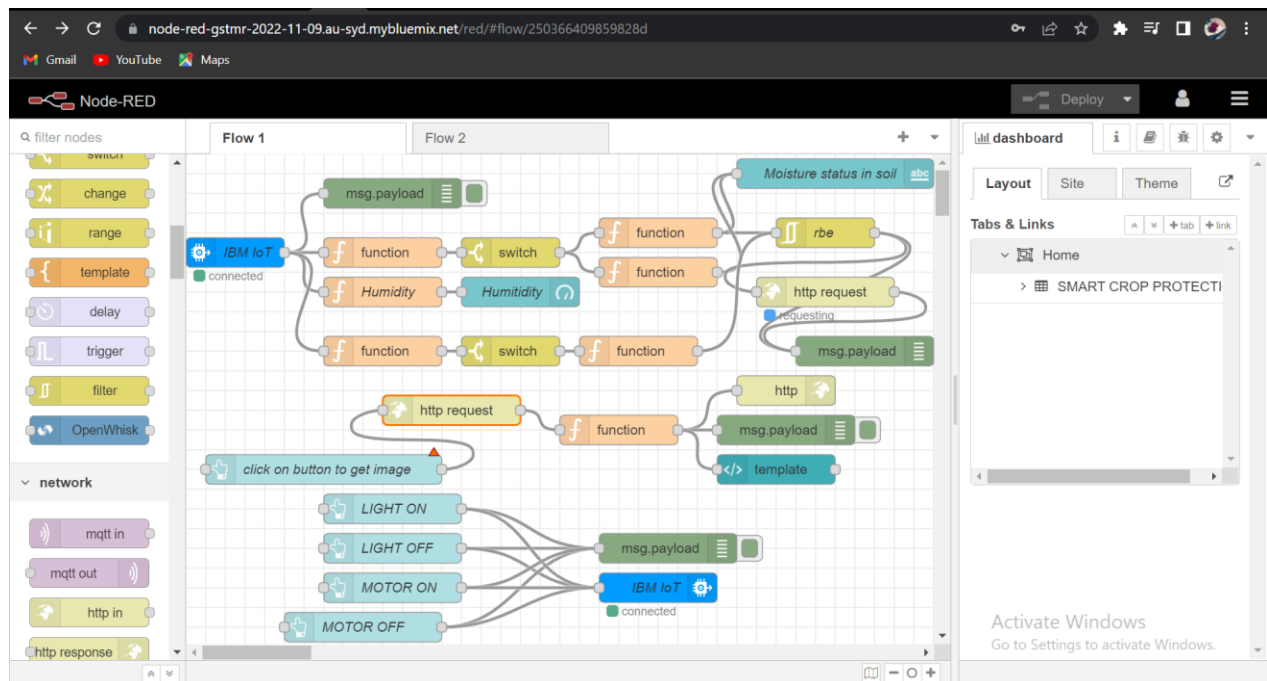


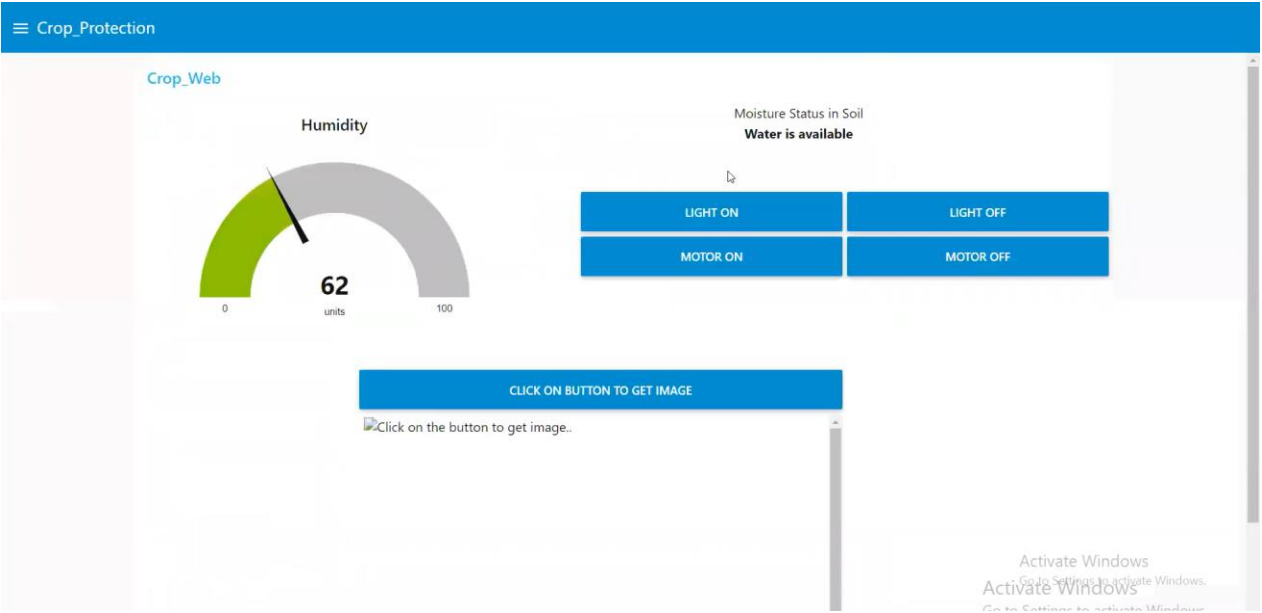
## Sprint 4

Team ID	PNT2022TMID13381
PROJECT NAME	IoT Based Smart Crop Protection System For Agriculture

### NODE-RED-CONNECTION:



OUTPUT:



# PYTHON CODE:

PY CODE.py - C:\python\Python37\PY CODE.py (3.7.4)

File Edit Format Run Options Window Help

```
import cv2
import numpy as np
import wiotp.sdk.device
import playsound
import random
import time
import datetime
import ibm_boto3
import winsound
import wave

from ibm_botocore.client import Config, ClientError
#CloudantDB
from cloudant.client import Cloudant
from cloudant.error import CloudantException
from cloudant.result import Result, ResultByKey
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2_grpc
stub = service_pb2_grpc.V2Stub(ClarifaiChannel.get_grpc_channel())
from clarifai_grpc.grpc.api import service_pb2, resource_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2
# This is how you authenticate.
metadata = (('authorization', 'Key bc885e5165d74ef48f42f6f6a2c9eb87'),)
COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud" # Current list available at https://control.cloud-object-storage.cloud.ibm.com/v2
COS_API_KEY_ID = "f6Ap-ct18m07S9UZL7XPbAF7l70ome PLLUQ0zqgmAzb5" # eg "W00YiRnLW4a3ftj MB-odB-2ySfTrFBIQQ'Wanc -- P3byk"
COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"
COS_RESOURCE_CRN = "crn:vl:bluemix:public:cloudantnosqldb:eu-gb:a/d43aa7d0631b400e9283084df08f9f60:502851d6-a240-4b22-8d4b-3642ed2bc3a8::" # eg "crn:vl:
clientdb = Cloudant("apikey-v2-lwveoo67391o7qj5cy7kqtpfsku8dumxlv6dy62rwu2", "64455b04f35e5d5f9b4fc25bb38904af", url = "https://apikey-v2-lwveoo67391o7
    username = "apikey-v2-lwveoo67391o7qj5cy7kqtpfsku8dumxlv6dy62rwu2")
clientdb.connect()
#Create resource
cos=ibm_boto3.resource("s3",
    ibm_api_key_id=COS_API_KEY_ID,
    ibm_service_instance_id=COS_RESOURCE_CRN,
    ibm_auth_endpoint=COS_AUTH_ENDPOINT,
    config=Config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)
def multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".
            format(item_name, bucket_name))
        #set 5 MB chunks
        part_size = 1024*1024 * 5
```

Activate Windows  
Go to Settings to activate Windows

PY CODE.py - C:\python\Python37\PY CODE.py (3.7.4)

File Edit Format Run Options Window Help

```
        format(item_name, bucket_name))
        #set 5 MB chunks
        part_size = 1024*1024 * 5
        #set threshold to 15 MB
        file_threshold = 1024 * 1024 * 15
        #set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )
        # the upload_fileobj method will automatically execute a multi-part upload
        # in 5 MB chunks for all files over 15 MB
        with open(file_path, "rb") as file_data:
            cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
            )
        print("Transfer for {0} Complete!\n".format(item_name))

    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: {0}".format(e))
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)
    command=cmd.data[ ' command']
    print(command)
    if(command == 'lighton'):
        print('lighton')
    elif(command == 'lightoff'):
        print('lightoff')
    elif(command == 'motoron'):
        print('motoron')
    elif(command == 'motoroff') :
        print('motoroff')

myConfig = {
    "identity": {
        "orgId": "blxckb",
        "typeId": "NodeMCU",
        "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}
```

Activate Windows  
Go to Settings to activate Windows.

Ln: 35 Col:

```

}
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
database_name = "sample"
my_database = clientdb.create_database(database_name)
if my_database.exists():
    print(f'{database_name} ' successfully created.")
cap=cv2.VideoCapture('garden.mp4')
if(cap.isOpened()== True) :
    print('File opened')
else:
    print('File not found')
while(cap.isOpened()):
    ret, frame=cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    ims = cv2.resize(frame, (960, 540))
    cv2.imwrite('ex.jpg',ims)
    with open("ex.jpg", "rb") as f:
        file_bytes = f.read()
    # This is the model ID of a publicly available General model. You may use any other public or custom model ID.
    request = service_pb2.PostModelOutputsRequest(
        model_id="aaa03c23b3724a16a56b629203edc62c",
        inputs=[resources_pb2.Input(data=resources_pb2.Data(image=resources_pb2.Image(base64=file_bytes))
        ))
    response = stub.PostModelOutputs(request, metadata=metadata)
    if response.status.code != status_code_pb2.SUCCESS:
        raise Exception("Request failed, status code: " + str(response.status.code))
    detect=False
    for concept in response.outputs[0].data.concepts:
        #print('%12s: %.2f: % (concept.name, concept.value))
        if(concept.value>0.98):
            #print(concept.name)
            if(concept.name=="animal") :
                print("Alert! Alert! animal detected")
                #playsound.playsound('alarm.wav')
                playsound.playsound('alert.mp3')
                picname=datetime.datetime.now().strftime("%Y-%m-%d-%H-%M")
                cv2.imwrite(picname+ '.jpg',frame)
                multi_part_upload('kiruthika2001', picname+ '.jpg', picname+ '.jpg')
                json_document={"link":COS_ENDPOINT+'/'+'kiruthika2001'+'/'+'picname+'.jpg'}
                new_document = my_database.create_document(json_document)
                if new_document.exists():
                    print(f'Document successfully created.")
                    time.sleep(5)

```

Activate Windows  
Go to Settings to activate Windows.

Ln: 35 Col: 0

```

        print(f"Document successfully created.")
        time.sleep(5)
        detect=True
        moist=random.randint(0,100)
        humidity=random.randint(0,100)
        myData= { 'Animal' : detect, 'moisture' :moist, 'humidity':humidity}
        print(myData)
        if(humidity!=None):
            client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
            print("Publish Ok .")
            client.commandCallback = myCommandCallback
            cv2.imshow('frame ', ims)
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
client.disconnect()
cap.release()
cv2.destroyAllWindows()

```

Activate Windows  
Go to Settings to activate Windows.

Ln: 35 Col: 0

# OUTPUT:

```
*Python 3.7.4 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\python\Python37\PY CODE.py =====
2022-11-18 23:59:40,332 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:blxckb:NodeMCU:12345
'sample' successfully created.
File opened
{'Animal': False, 'moisture': 0, 'humidity': 60}
Publish Ok ..
{'Animal': False, 'moisture': 100, 'humidity': 27}
Publish Ok ..
{'Animal': False, 'moisture': 45, 'humidity': 37}
Publish Ok ..
{'Animal': False, 'moisture': 93, 'humidity': 16}
Publish Ok ..
{'Animal': False, 'moisture': 86, 'humidity': 40}
Publish Ok ..
{'Animal': False, 'moisture': 87, 'humidity': 81}
Publish Ok ..
{'Animal': False, 'moisture': 47, 'humidity': 28}
Publish Ok ..
{'Animal': False, 'moisture': 24, 'humidity': 24}
Publish Ok ..
{'Animal': False, 'moisture': 6, 'humidity': 71}
Publish Ok ..
{'Animal': False, 'moisture': 18, 'humidity': 36}
Publish Ok ..
{'Animal': False, 'moisture': 61, 'humidity': 76}
Publish Ok ..
{'Animal': False, 'moisture': 94, 'humidity': 20}
Publish Ok ..
{'Animal': False, 'moisture': 23, 'humidity': 88}
Publish Ok ..
|
```

Activate Windows  
Go to Settings to activate Windows.

Ln: 34 Col: 0