Sprint 3

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

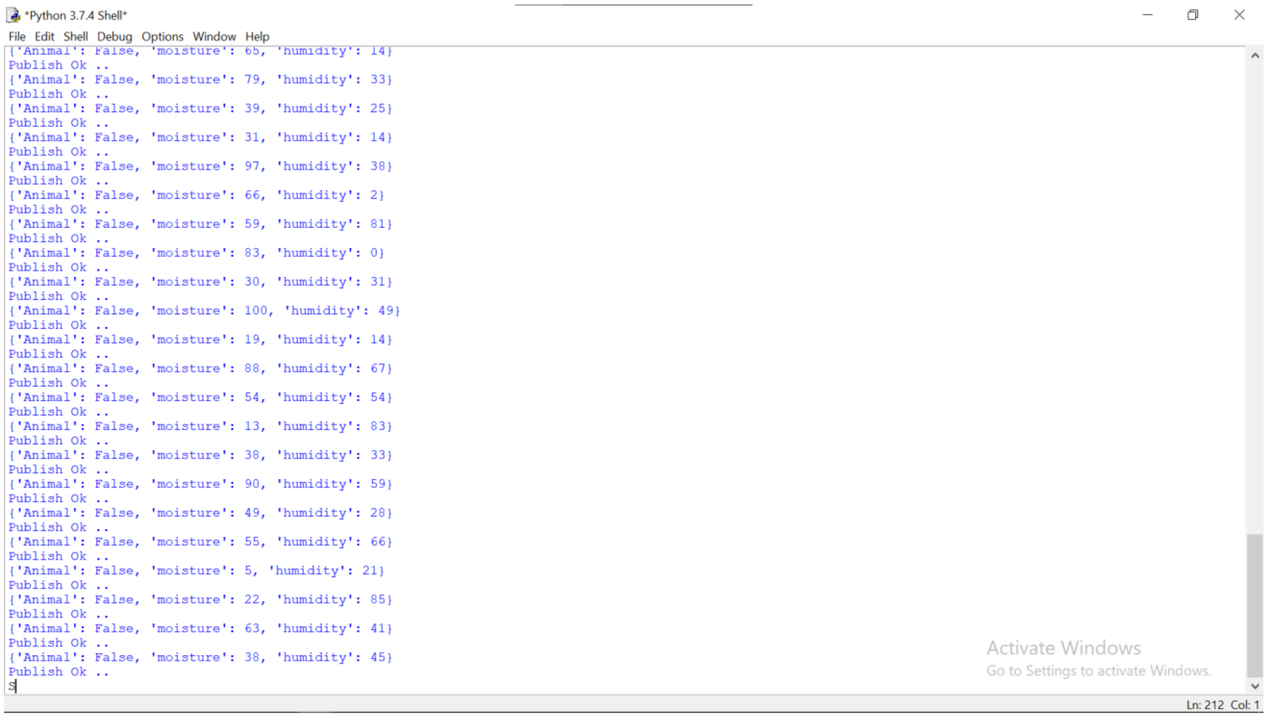
Develop a python script:

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

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, resources_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 avaiable at https://control.cloud-
object-storage.cloud.ibm.com/v2/endpoints

COS_API_KEY_ID = " f6Ap-ct18m07S9UZL7XPbAF7170ome
PLLUQOzqmnAzb5" # eg "W00YiRnLW4a3fTj MB-odB-
2ySfTrFBIQQ'Wanc -- P3byk"

COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"

COS_RESOURCE_CRN = "crn:v1:bluemix:public:cloudantnosqldb:eu-
gb:a/d43aa7d0631b400e9283084df08f9f60:502851d6-a240-4b22-8d4b-
3642ed2bc3a8::" # eg "crn:vl:bluemix:public:cloud-object-
storage:global:a/6b644a3fda97448b888c23eeef263ed6:199ab1e5-0d9d-
420f-8e4a-98d868c04368 ::"

clientdb = Cloudant("apikey-v2-
1wveoo6739lo7qj5cy7kqtpfsku8dumxlvp6dy62rwu2",
"64455b04f35e5d5f9b4fc25bb38904af", url = "https://apikey-v2-
1wveoo6739lo7qj5cy7kqtpfsku8dumxlvp6dy62rwu2:64455b04f35e5d5f
9b4fc25bb38904af@de3c99da-899c-43cb-9aa5-b6b3fdc4cc16-
bluemix.cloudantnosqldb.appdomain.cloud",

  username = "apikey-v2-
1wveoo6739lo7qj5cy7kqtpfsku8dumxlvp6dy62rwu2")
```

```
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
        #set threadhold 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"
}
}
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"¹ {database_name} ' successfully created.")
cap=cv2.VideoCapture('monkey.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: %.2f¹ % (concept.name, concept.value))
```

```
    if(concept.value>0.98):
      #print(concept.name)
      if(concept.name =="animal") :
        print("Alert! Alert! animal detected")
        playsound.playsound('alert.mp3')
       # 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'+'/'+picna
me+'.jpg'}
        new_document = my_database.create_document(json_document)
        if new_document.exists():
          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()
```


Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Activate Windows
Go to Settings to activate Windows.
Ln: 212 Col: 1

```
{'Animal': False, 'moisture': 65, 'humidity': 14}
Publish Ok ..
{'Animal': False, 'moisture': 79, 'humidity': 33}
Publish Ok ..
{'Animal': False, 'moisture': 39, 'humidity': 25}
Publish Ok ..
{'Animal': False, 'moisture': 31, 'humidity': 14}
Publish Ok ..
{'Animal': False, 'moisture': 97, 'humidity': 38}
Publish Ok ..
{'Animal': False, 'moisture': 66, 'humidity': 2}
Publish Ok ..
{'Animal': False, 'moisture': 59, 'humidity': 81}
Publish Ok ..
{'Animal': False, 'moisture': 83, 'humidity': 0}
Publish Ok ..
{'Animal': False, 'moisture': 30, 'humidity': 31}
Publish Ok ..
{'Animal': False, 'moisture': 100, 'humidity': 49}
Publish Ok ..
{'Animal': False, 'moisture': 19, 'humidity': 14}
Publish Ok ..
{'Animal': False, 'moisture': 88, 'humidity': 67}
Publish Ok ..
{'Animal': False, 'moisture': 54, 'humidity': 54}
Publish Ok ..
{'Animal': False, 'moisture': 13, 'humidity': 83}
Publish Ok ..
{'Animal': False, 'moisture': 38, 'humidity': 33}
Publish Ok ..
{'Animal': False, 'moisture': 90, 'humidity': 59}
Publish Ok ..
{'Animal': False, 'moisture': 49, 'humidity': 28}
Publish Ok ..
{'Animal': False, 'moisture': 55, 'humidity': 66}
Publish Ok ..
{'Animal': False, 'moisture': 5, 'humidity': 21}
Publish Ok ..
{'Animal': False, 'moisture': 22, 'humidity': 85}
Publish Ok ..
{'Animal': False, 'moisture': 63, 'humidity': 41}
Publish Ok ..
{'Animal': False, 'moisture': 38, 'humidity': 45}
Publish Ok ..
s
```