

PROJECT DEVELOPMENT PHASE

SPRINT-3 CODING

Date	09 November 2022
Team ID	PNT2022TMID49462
Project Name	IOT based Smart Crop Protection for Agriculture
Maximum Marks	8 Marks

Coding:

```
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
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
# Constants for IBM COS values
COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud" # Current list available at
https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints
COS_API_KEY_ID = "9Vp5U7vuntgNduJwVKZXDn7f4wKRPaDJT5a1kiDXapEP" # eg
"W00YiRnLW4a3fTjMB-odB-2ySfTrFBIQQWanc--P3byk"
COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"
COS_RESOURCE_CRN = "crn:v1:bluemix:public:cloud-object-
storage:global:a/9b399604cf904a88a997a81684b97184:24b5d2b1-5259-4c70-83ac-24552802d92e::" # eg
"crn:v1:bluemix:public:cloud-object-storage:global:a/3bf0d9003abfb5d29761c3e97696b71c:d6f04d83-6c4f-
4a62-a165-696756d63903::"

clientdb=Cloudant("apikey-v2-thovtw7l8mtpl3w17or63t37f6a4wivqvvgg4gkhi83",
"89c52b4f4fdd6459155676646ee45a3a", url="https://apikey-v2-
thovtw7l8mtpl3w17or63t37f6a4wivqvvgg4gkhi83:89c52b4f4fdd6459155676646ee45a3a@a6f9045f-ff4b-
4a32-a9a0-35efb443e91b-bluemix.cloudantnosqldb.appdomain.cloud")
clientdb.connect()

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))
        part_size = 1024 * 1024 * 5
        file_threshold = 1024 * 1024 * 15
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )
        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: {1}\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: %$" % 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": "4wq3lx",
        "typeId": "raspberrypi",
        "deviceId": "demo123"
    },
    "auth" : {
        "token": "mind1234"
    }
}

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()
        print("Alert! Alert! Animal detected")
        picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
        cv2.imwrite(picname+'.jpg',frame)
        multi_part_upload("karthi01", picname+'.jpg', picname+'.jpg')
        json_document={"link":COS_ENDPOINT+'/'+karthi01+'/'+picname+'.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()

```