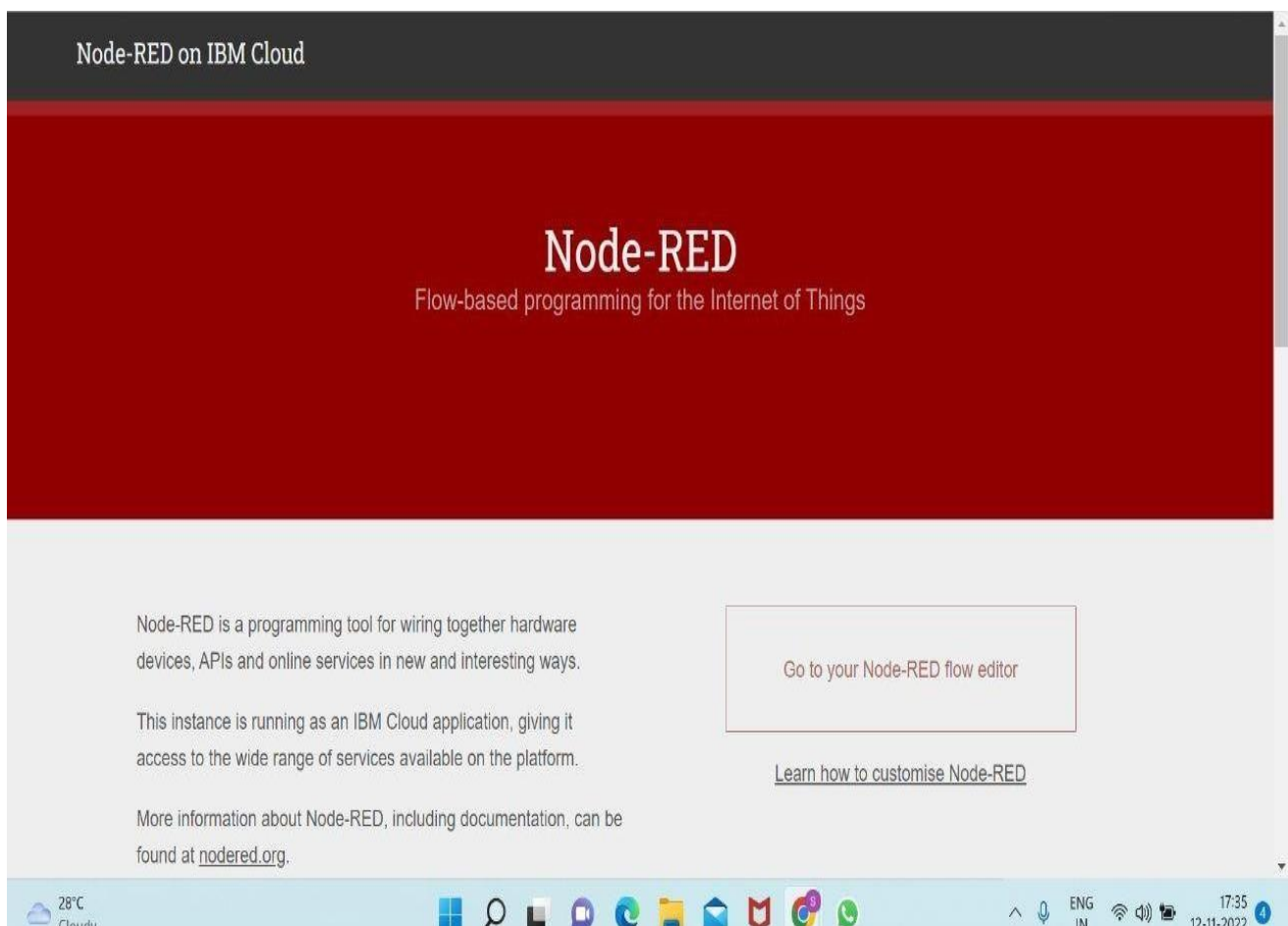
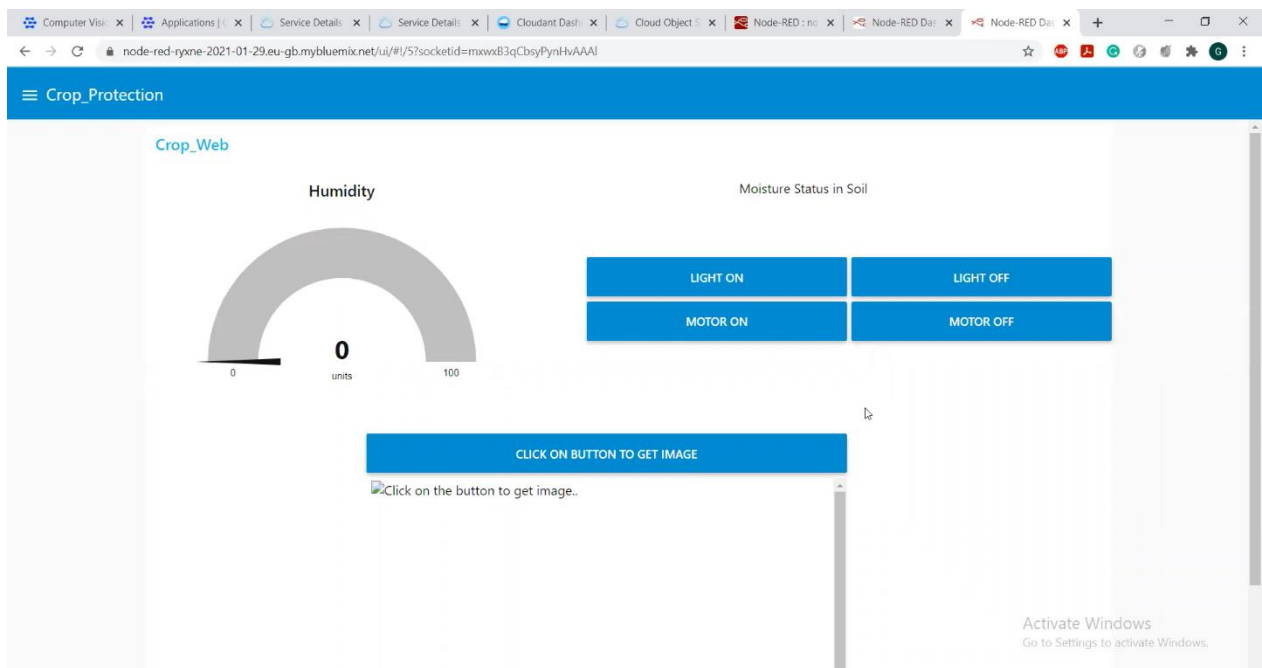
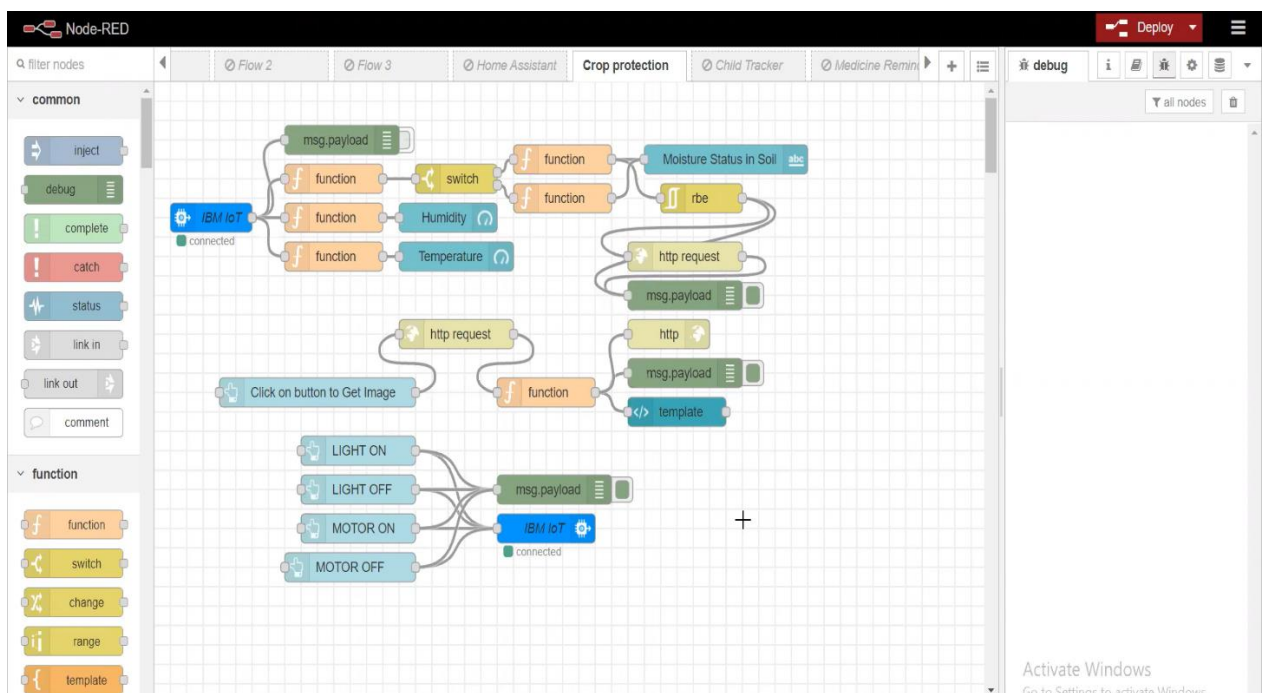


DEVELOP A WEB APPLICATION USING NODERED

TEAM ID	PNT2022TMID14113
TITLE	IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE





```

C:\Users\HP/Desktop/crop/ocrop/ocrop.py (3.8.8)
File Edit Format Run Options Window Help

import cv2
import numpy as np
import wiop.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"),})
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 = "f6ap-ctl8mo7S9u17XtBA7170omePLLUQ0zgnAzb5" # eg "W09YiRnLm4a3ffjMB-0dB-2ysftrfBIQQWanc--P3byk"
COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"
COS_RESOURCE_CRN = "crn:vl:bluemix:public:cloud-object-storage:global:a/6b644a3fda97448b898c23eeef263ed6:199able5-0d9d-420f-8e4a-98d868c04368:" # eg "crn:vl:bluemix:public:cloud-object-storage:global:a/6b644a3fda97448b898c23eeef263ed6:199able5-0d9d-420f-8e4a-98d868c04368:"
clientdb = Cloudant(
    endpoint=COS_AUTH_ENDPOINT,
    apikey=COS_API_KEY_ID,
    crn=COS_RESOURCE_CRN,
    config=Config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)

# 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 {} to bucket: {}".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_chunksiz=part_size
        )
        # the upload_fileobj method will automatically execute a multi-part upload
    except Exception as e:
        print(e)

```

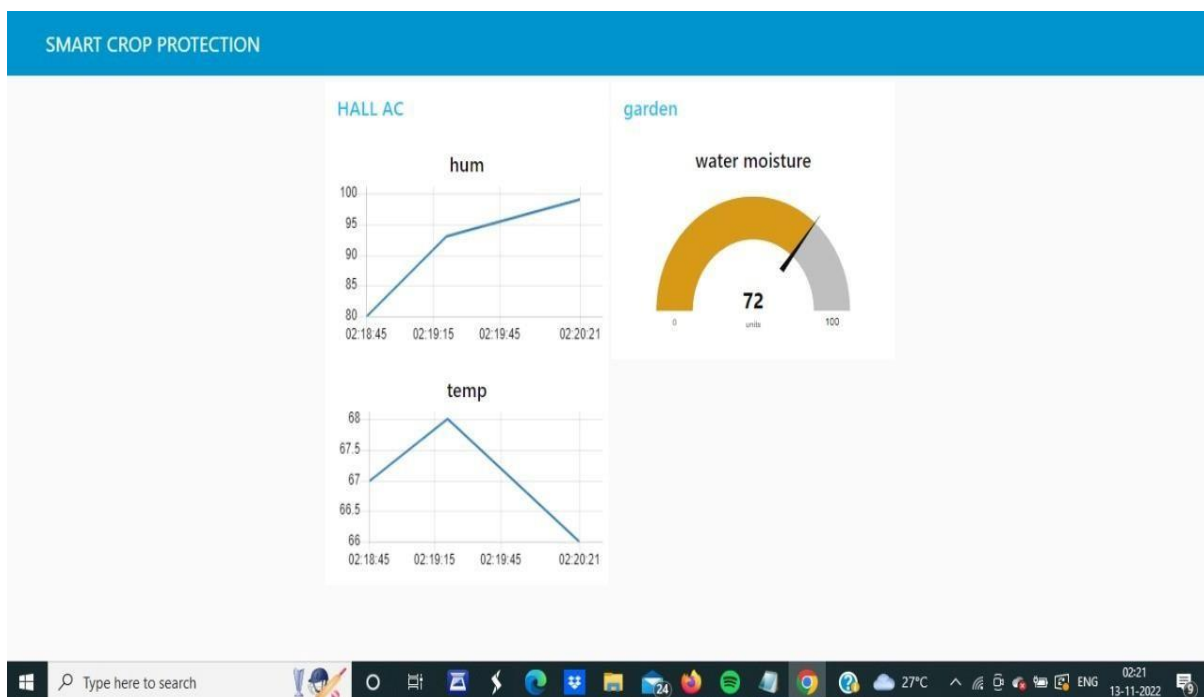
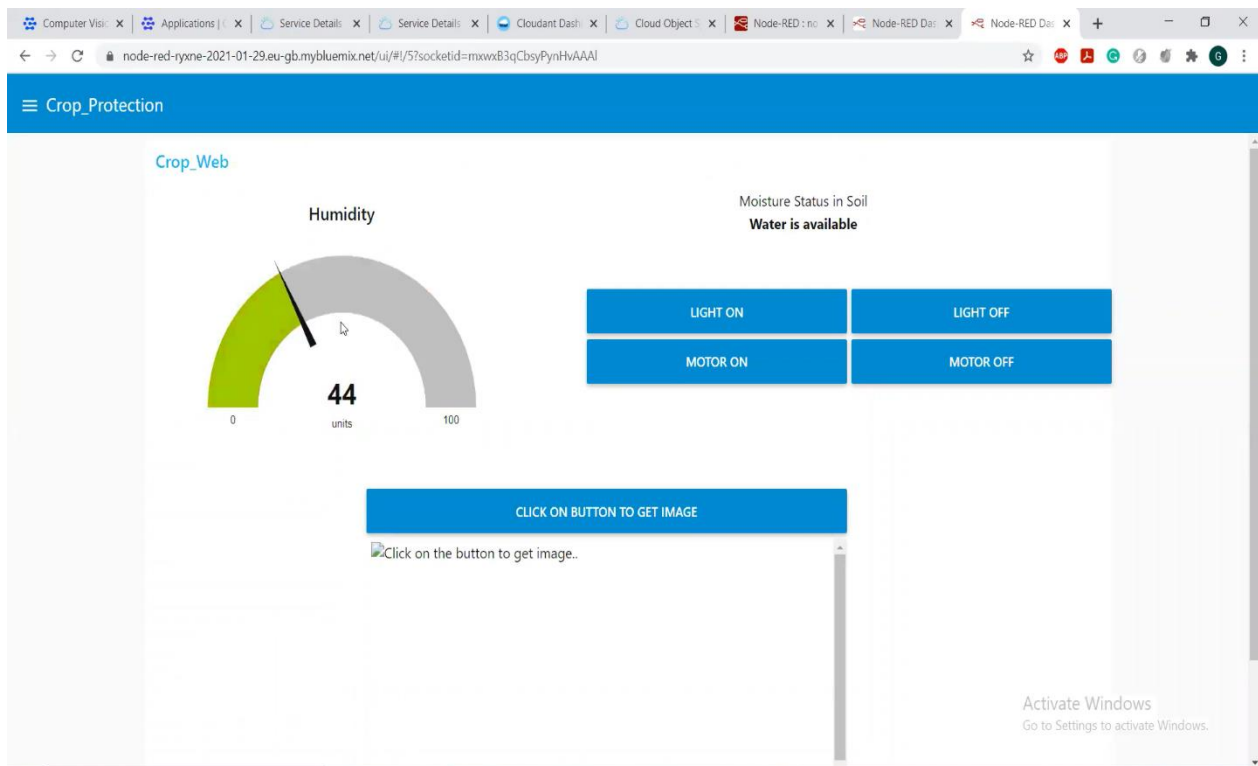
```
C:\Users\HP\Desktop/crop/crop_protect.py (3.8.8)
File Edit Format Run Options Window Help

Fileobj=file data,
Config=transfer_config
)
print("Transfer for {} Complete!\n".format
except ClientError as be:
print("CLIENT ERROR: (0)\n".format(be))
except Exception as e:
print("Unable to complete multi-part upload")

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": "hj5fmy",
"typeId": "NodeMCU",
"deviceId": "12345"
},
"auth": {
"token": "12345678"
}
}
client = wiotp.sdk.device.DeviceClient(config=myCo
client.connect()

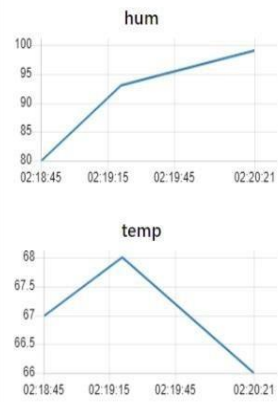
database_name = "sample"
my_database = clientdb.create_database(database_na
if my_database.exists():
print("'{database_name}' successfully created.
cap=cv2.VideoCapture('borden.mp4')
if(cap.isOpened()==True):
print('File opened')
else:
print('File not found')

while(cap.isOpened()):
ret, frame = cap.read()
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```



SMART CROP PROTECTION

HALL AC



garden

water moisture



SMART CROP PROTECTION

HALL AC



garden

water moisture

