

Automatic garage door opener

Final Project



**PYTHON CODE-**

#Vehicle Detection

import cv2

#IBM IOT connectioon

import wiotp.sdk.device

#IBM COS

import ibm\_boto3

#IBM Cloudant

from ibm\_botocore.client import Config, ClientError

from ibmcloudant.cloudant\_v1 import CloudantV1

from ibmcloudant import CouchDbSessionAuthenticator

from ibm\_cloud\_sdk\_core.authenticators import BasicAuthenticator

#Clarifai

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

#Miscellaneous

import time

import random

import datetime

#IOT device connection

myConfig = {

"identity": {

"orgId": "j8rgpm",

"typeId": "First\_Device",

"deviceId":"123"

},

"auth": {

"token": "First\_Device\_123"

}

}

# Constants for IBM COS values

COS\_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud"

COS\_API\_KEY\_ID = "78NLlHvfEUxheNWr4jehqUGifUPewWVaPrk3N2HXaMKy"

COS\_INSTANCE\_CRN = "crn:v1:bluemix:public:cloud-object-storage:global:a/abaf1723c9c84a148e04f11a73d2442a:880edc7f-ce14-4df9-8545-5c592c737b87::"

# Create resource

cos = ibm\_boto3.resource("s3",

ibm\_api\_key\_id=COS\_API\_KEY\_ID,

ibm\_service\_instance\_id=COS\_INSTANCE\_CRN,

config=Config(signature\_version="oauth"),

endpoint\_url=COS\_ENDPOINT

)

authenticator = BasicAuthenticator('apikey-v2-10xypw9nevga82oqb993uwcfdqgl748fcoiznj3jfrzn', '3fb54d29a2d64e40f7a0657467d33a27')

service = CloudantV1(authenticator=authenticator)

service.set\_service\_url('https://apikey-v2-10xypw9nevga82oqb993uwcfdqgl748fcoiznj3jfrzn:3fb54d29a2d64e40f7a0657467d33a27@c6a728a6-a633-4d59-a8e9-c06a612f8176-bluemix.cloudantnosqldb.appdomain.cloud')

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)

client.connect()

#Image upload to COS bucket

bucket = "rohanbucket"

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!\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))

video = cv2.VideoCapture('cars.avi')

# Clarifai Authentication

metadata = (('authorization', 'Key f9c413355966419bbd1b0e0125a9a5bc'),)

#Initial Conditions

print("Door is Closed")

print("Light is Off")

def myCommandCallback(cmd):

m=cmd.data['command']

print(m,"\n")

while True:

distance=random.randint(5,500) # random distance values from 5 to 500 meters

check,frame=video.read()

frame = cv2.resize(frame, (600,400))

if (distance<=15): #picture is sent to clarifai when distance is less than 15 meters

myData={'distance':distance}

print(myData)

client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)

picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M-%S")

path="D:\\rohan\\Personal\\Courses\\IOT\\Final Project\\pics\\" +picname + ".jpg"

cv2.imwrite(picname+".jpg",frame) # Images are stored locally in a folder

time.sleep(1)

with open(path, "rb") as f:

file\_bytes = f.read()

request = service\_pb2.PostModelOutputsRequest(

# This is the model ID of a publicly available General model. You may use any other public or custom model ID.

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))

a= []

for concept in response.outputs[0].data.concepts:

if(concept.value > 0.8):

a.append(concept.name)

t=1

for i in a:

if(i == "car" or i == "vehicle" or i=="bike"):

print("Vehicle is detected\n")

print("Door is open\nLight is On")

#if vehicle is detected document is uploaded to cloudant database

multi\_part\_upload(bucket, picname+'.jpg', picname+'.jpg')

json\_document={"link":COS\_ENDPOINT+'/'+bucket+'/'+picname+'.jpg'}

response = service.post\_document(db='sample', document=json\_document).get\_result()

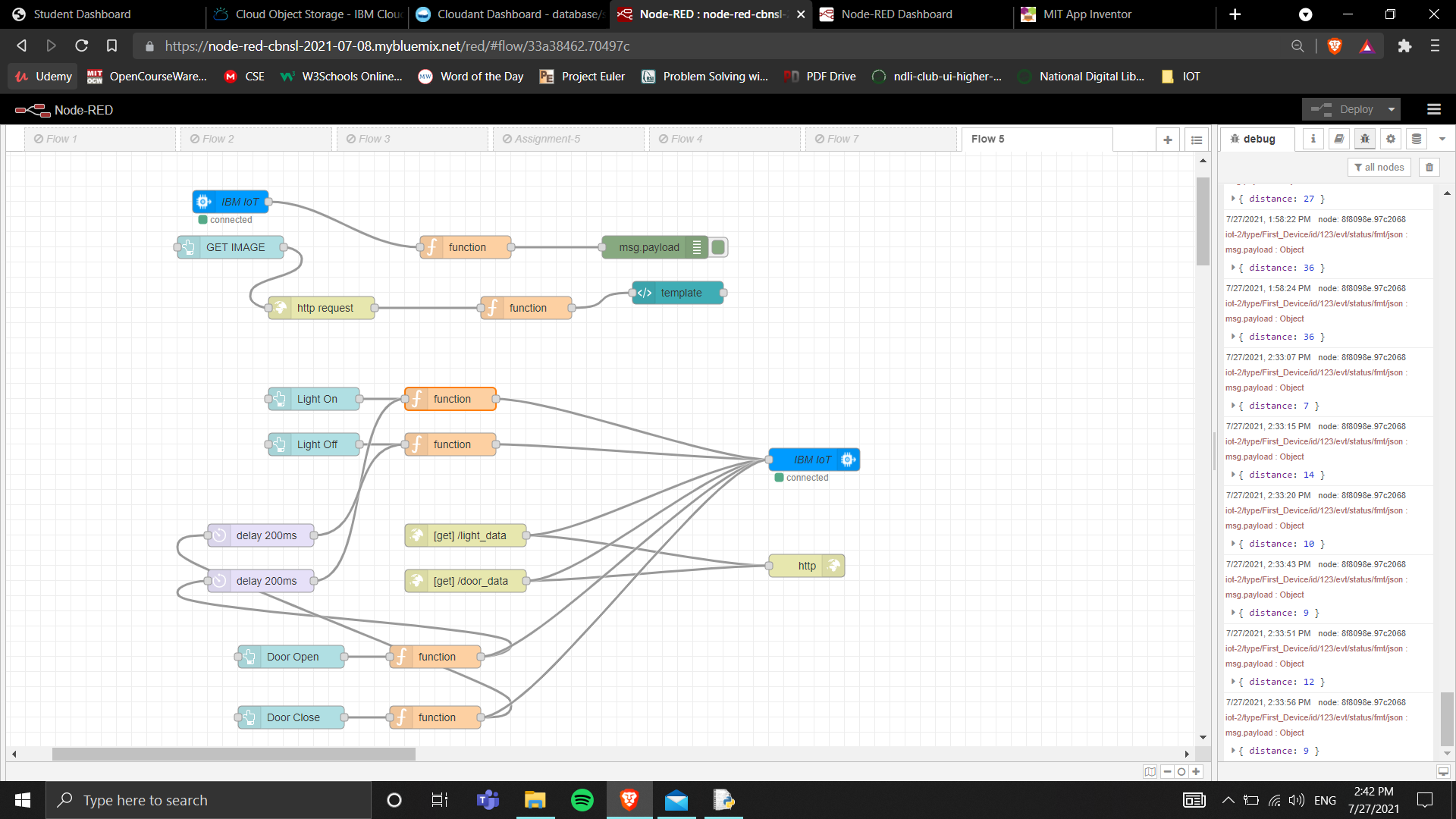
break

Key=cv2.waitKey(1)

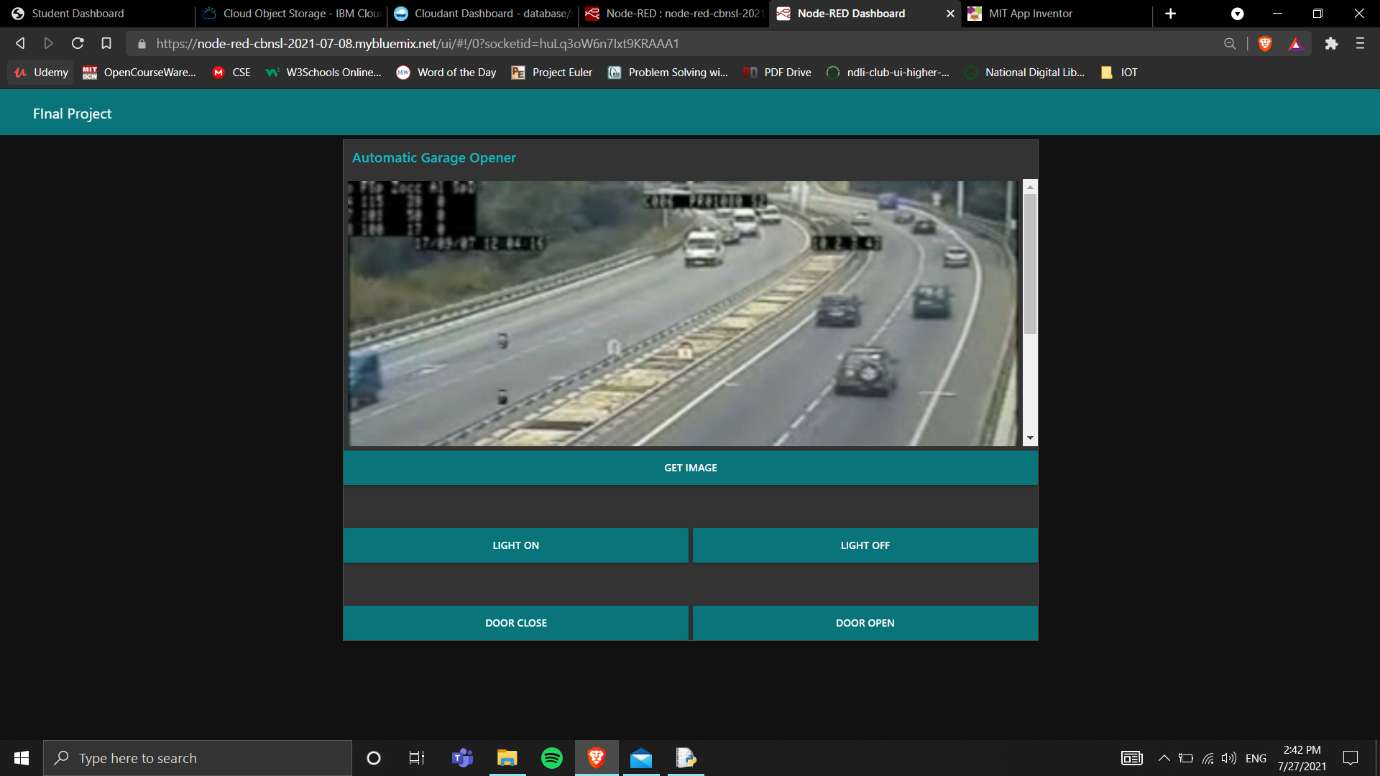
client.commandCallback = myCommandCallback

client.disconnect()

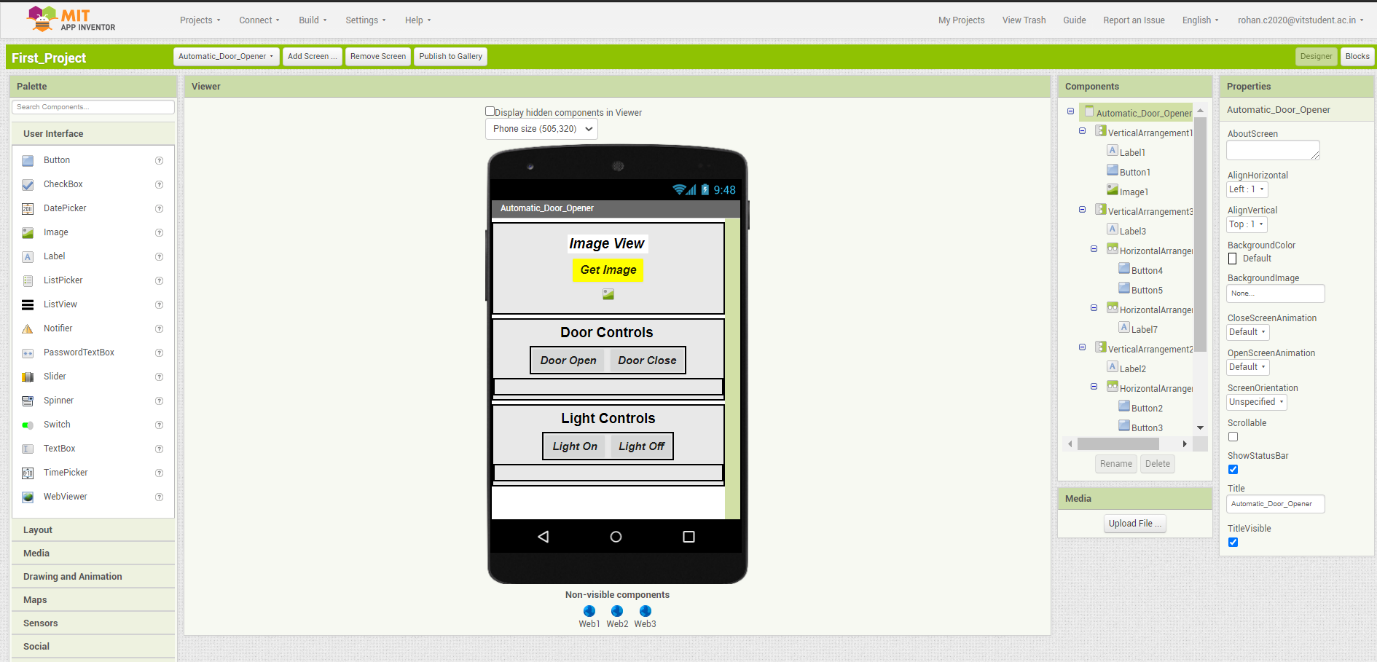
**NODE RED-**

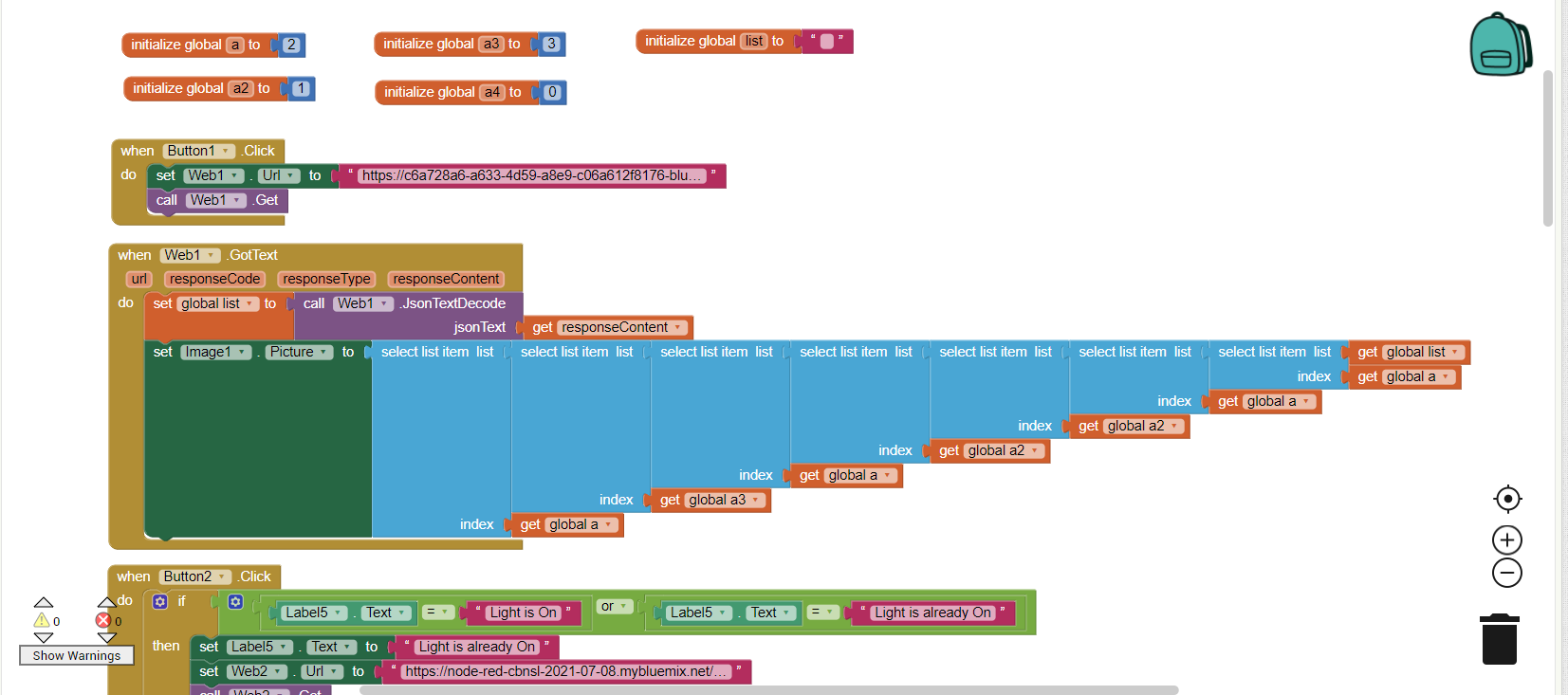


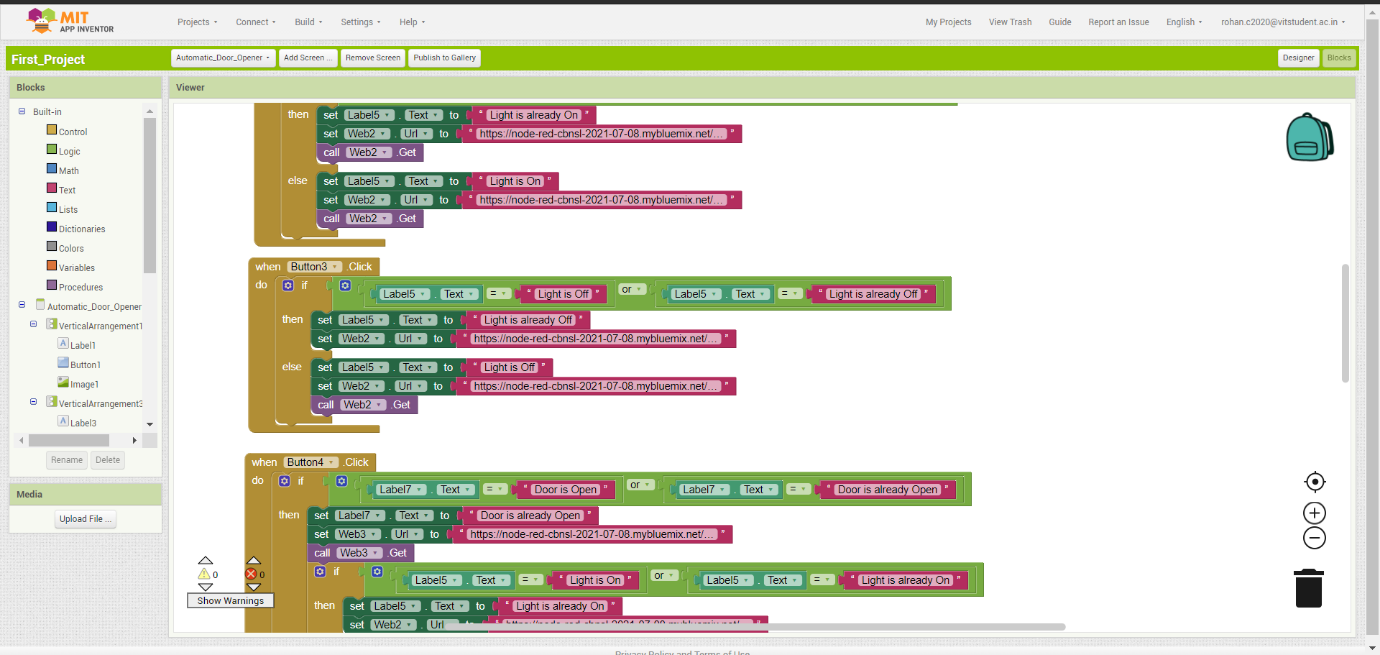
**WEB UI-**

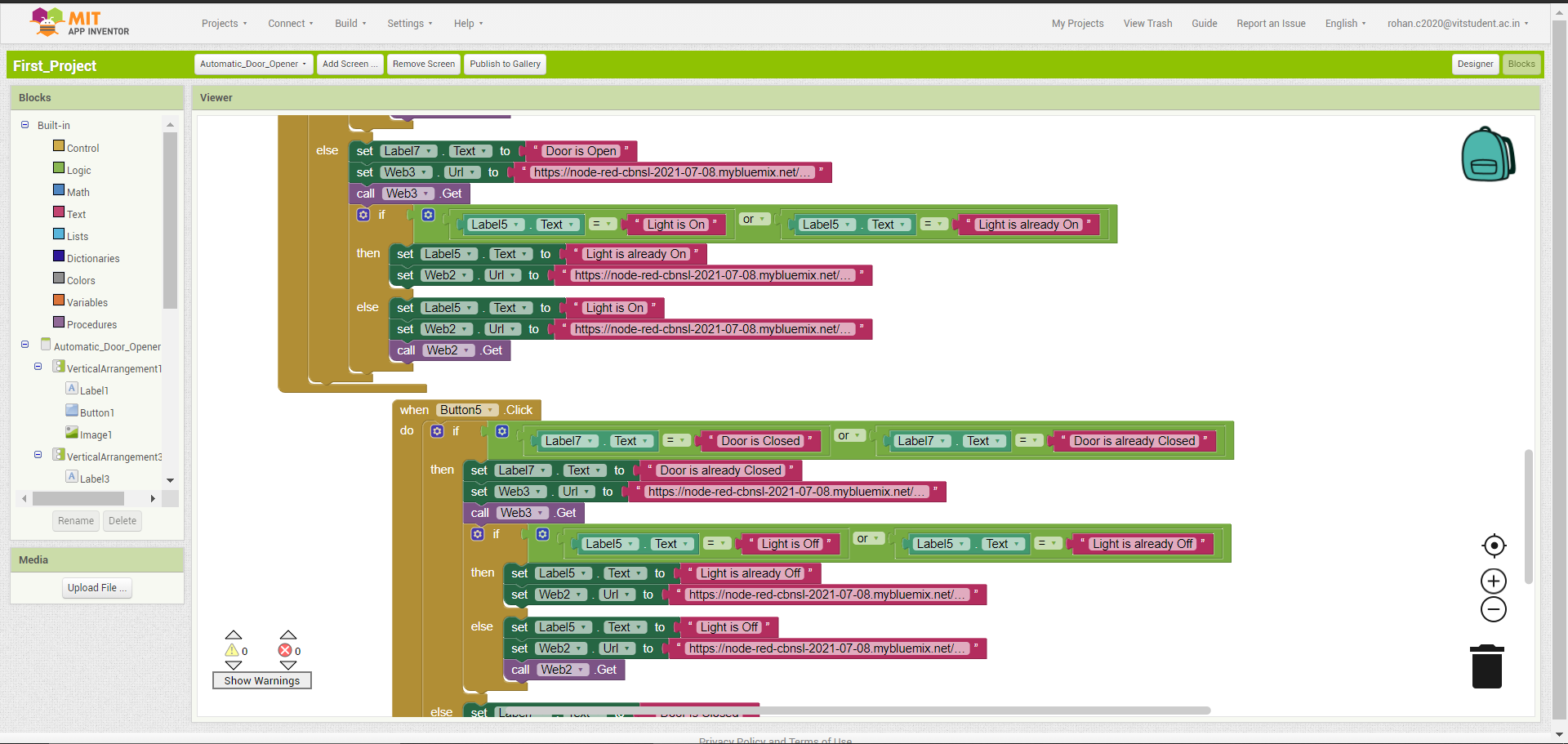


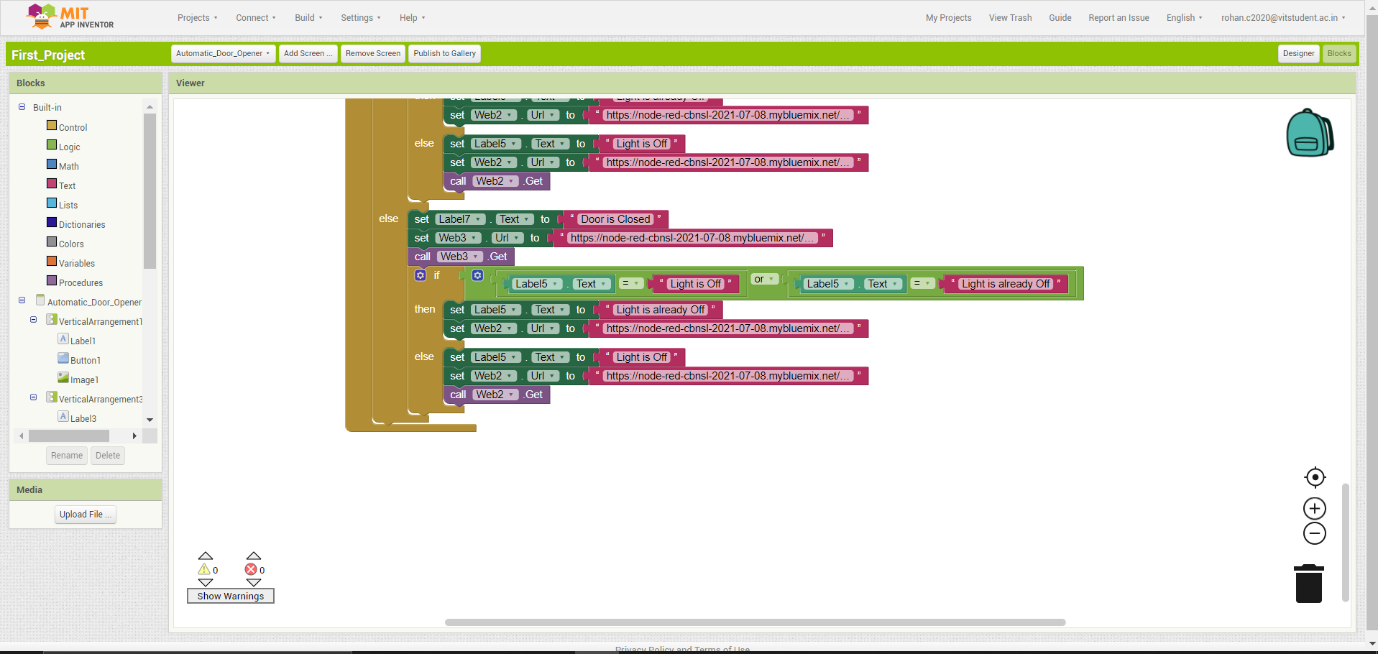
**MOBILE APP-**



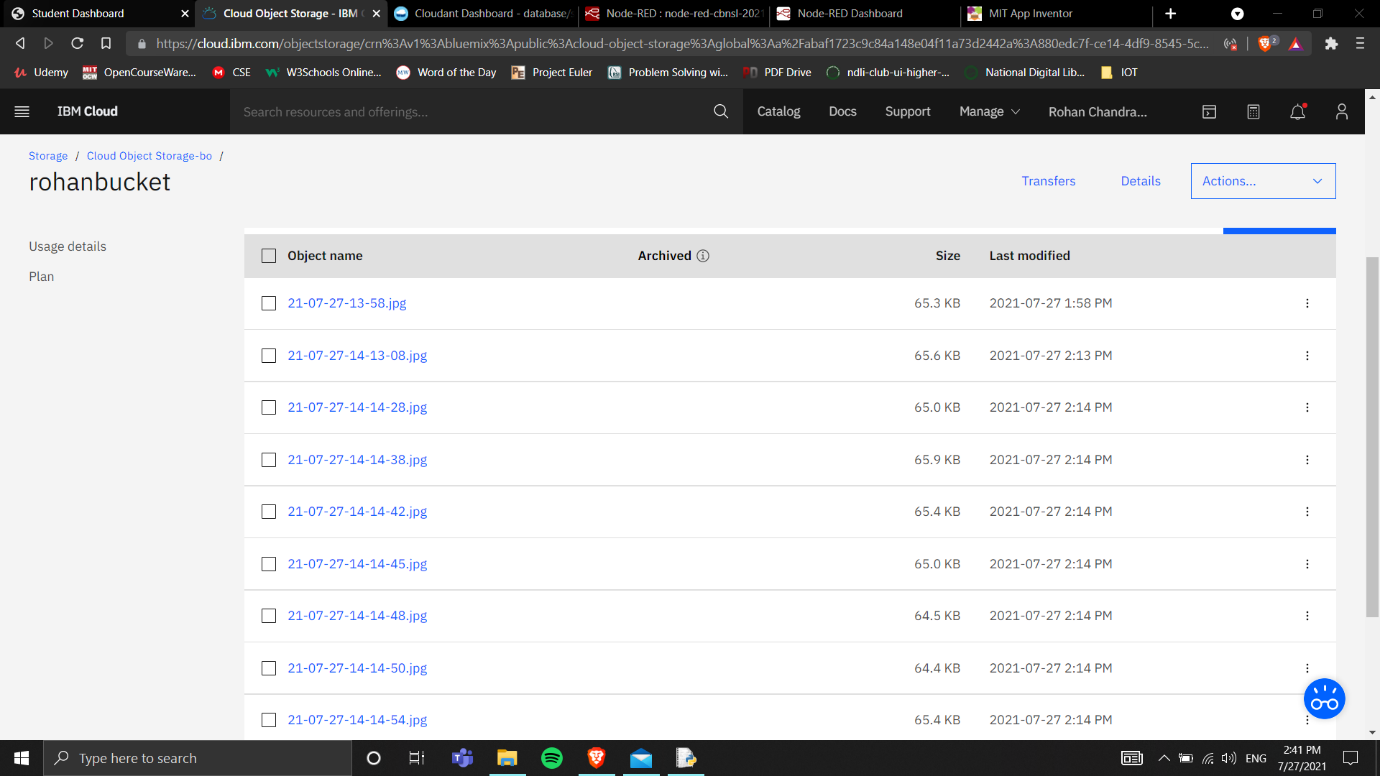


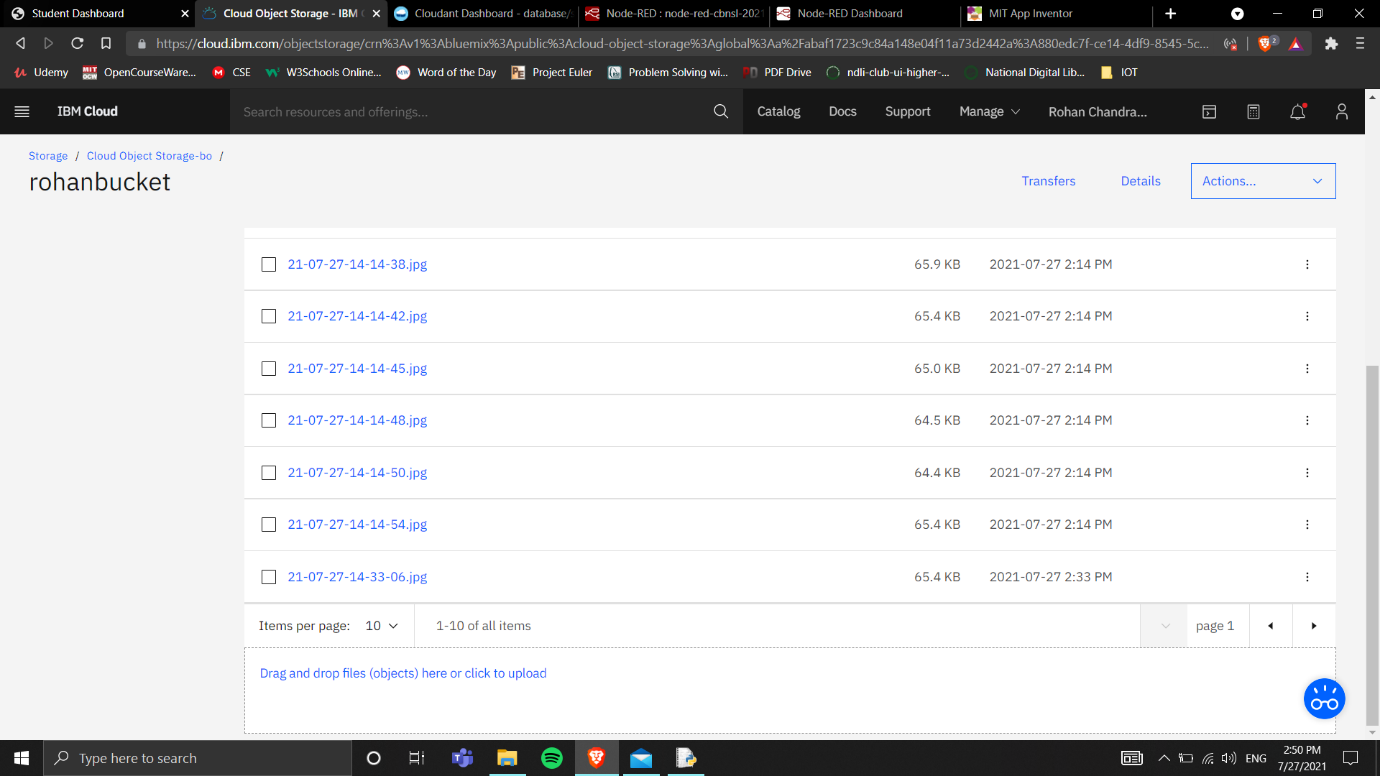




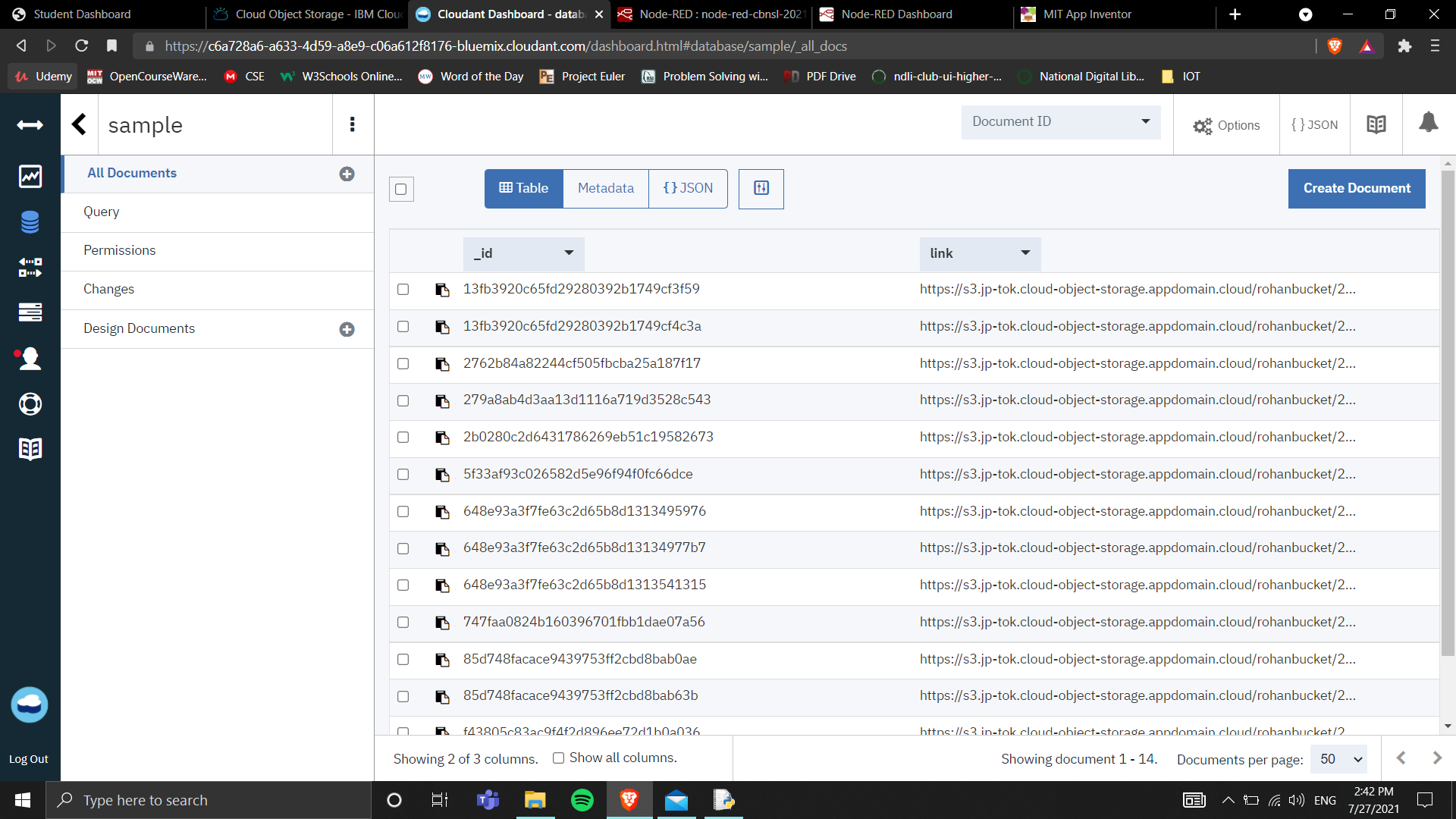


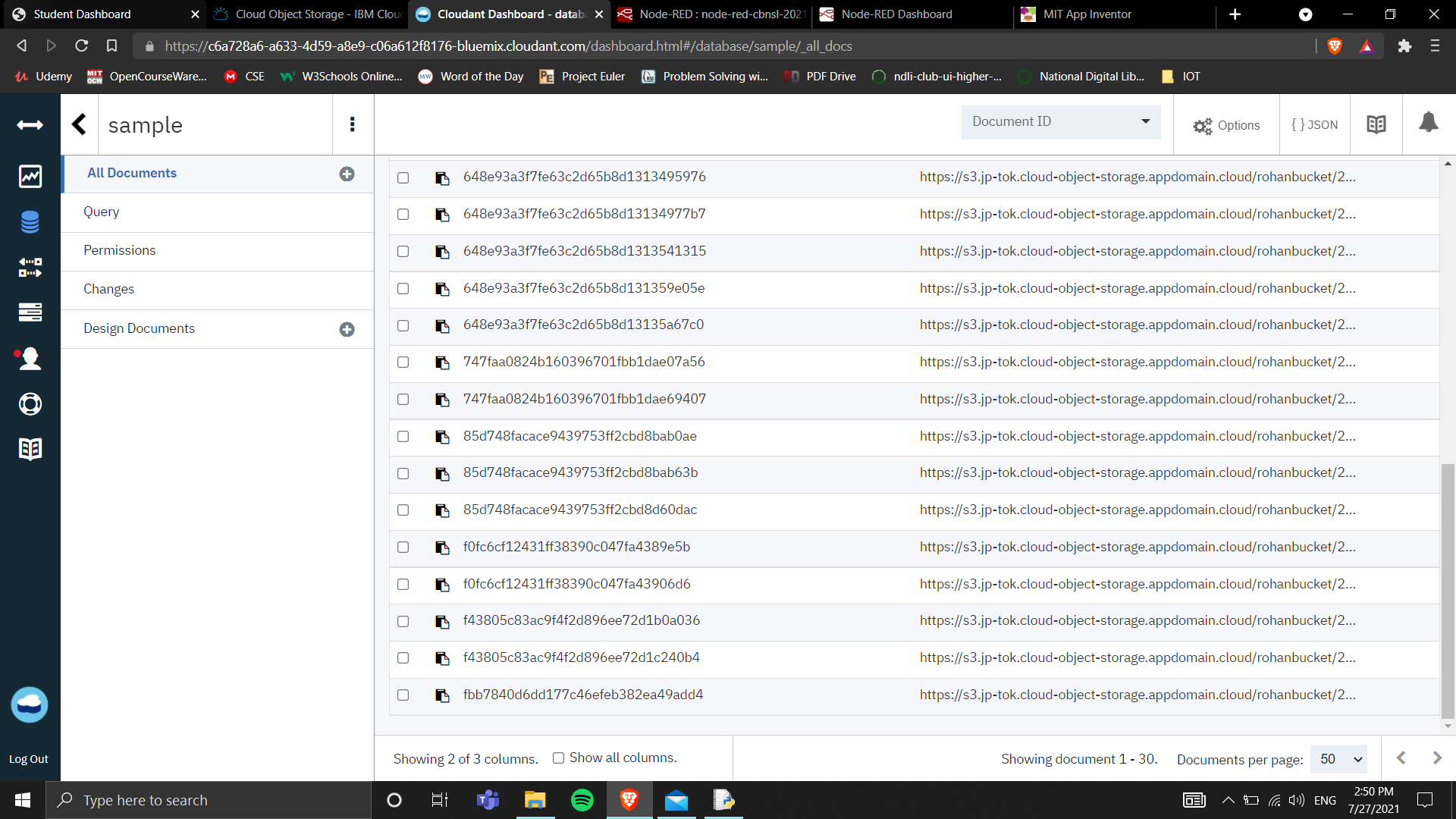
**CLOUD OBJECT STORAGE-**





**CLOUDANT DATABASE-**





**PYTHON SHELL OUTPUT-**



