1. INTRODUCTION

    a. OVIERVIEW

This is the documentation of my IoT Project from my internship. The project is based on fire dtectors while also detection gas levels and mointoring temperature.

    b. PURPOSE

This document highlights the need and reason to warrant a production of IoT based fire systems for comercial use, although this document will only deal with simulations and the software side of the IoT application.

2. LITERATURE SUMMARY

    a. EXISTING PROBLEM

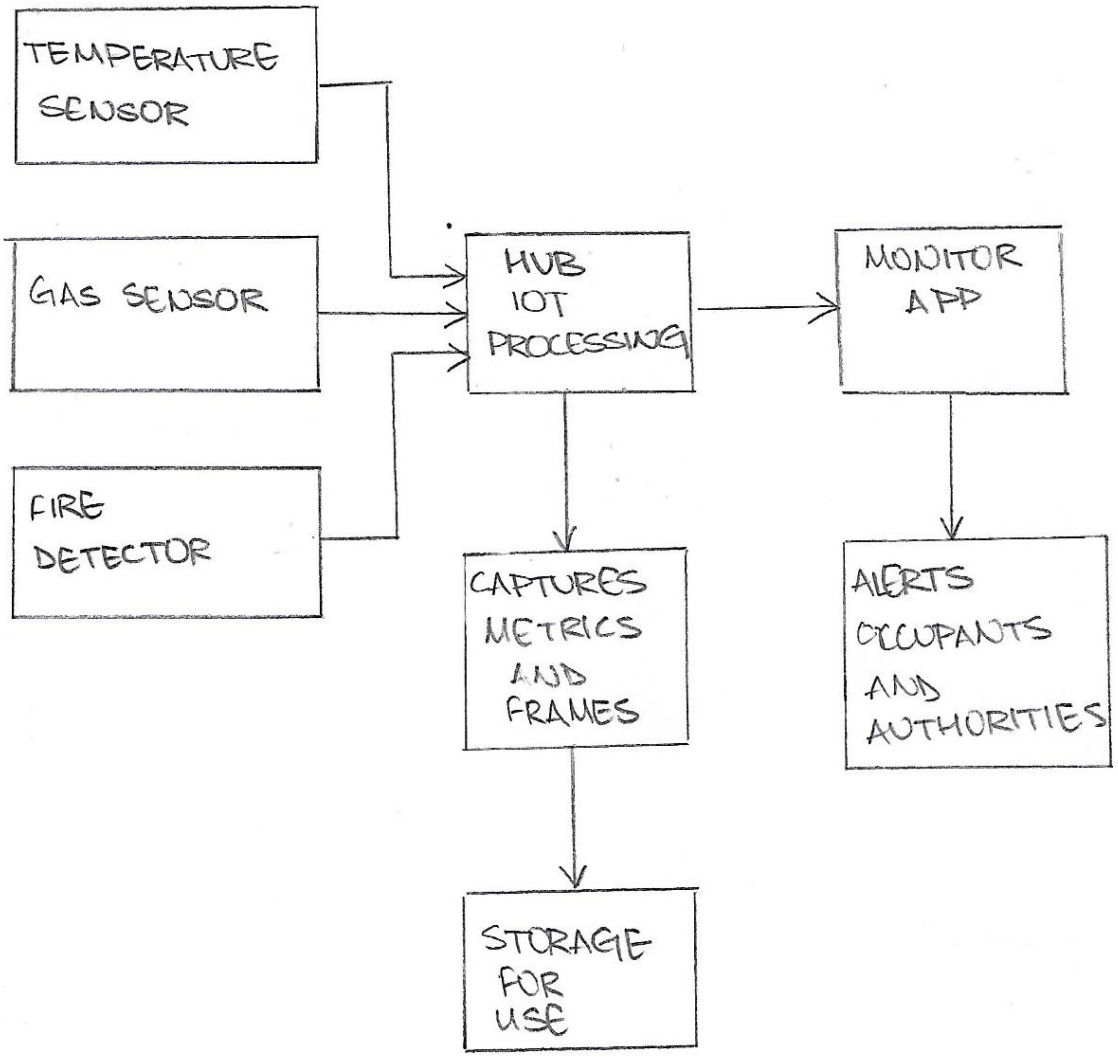
Many bulinding and resource are in present danger of succbumbing to flames due to various reasons, like short-circuting, which causes massive loss in resources and in many cases information and human lives.

    b. PROPOSED SOLUTION

This detector aims to timely detect and fire and alerting occupants to evacuate and the authorities to take necessary precautions. This is simply a prototype of teh device nad can only be applied in small-scale senarios at best.

3. THEORETICAL ANALYSIS

    a. BLOCK DIAGRAM



b. HARDWARE/SOFTWARE DESIGNING

For this prototype, python is used at the programming language. IBM Cloud and its services are used. As well as, MIT App Inventor for a mobile app to show the data.

4. EXPERIMENTAL INVESTIGATIONS

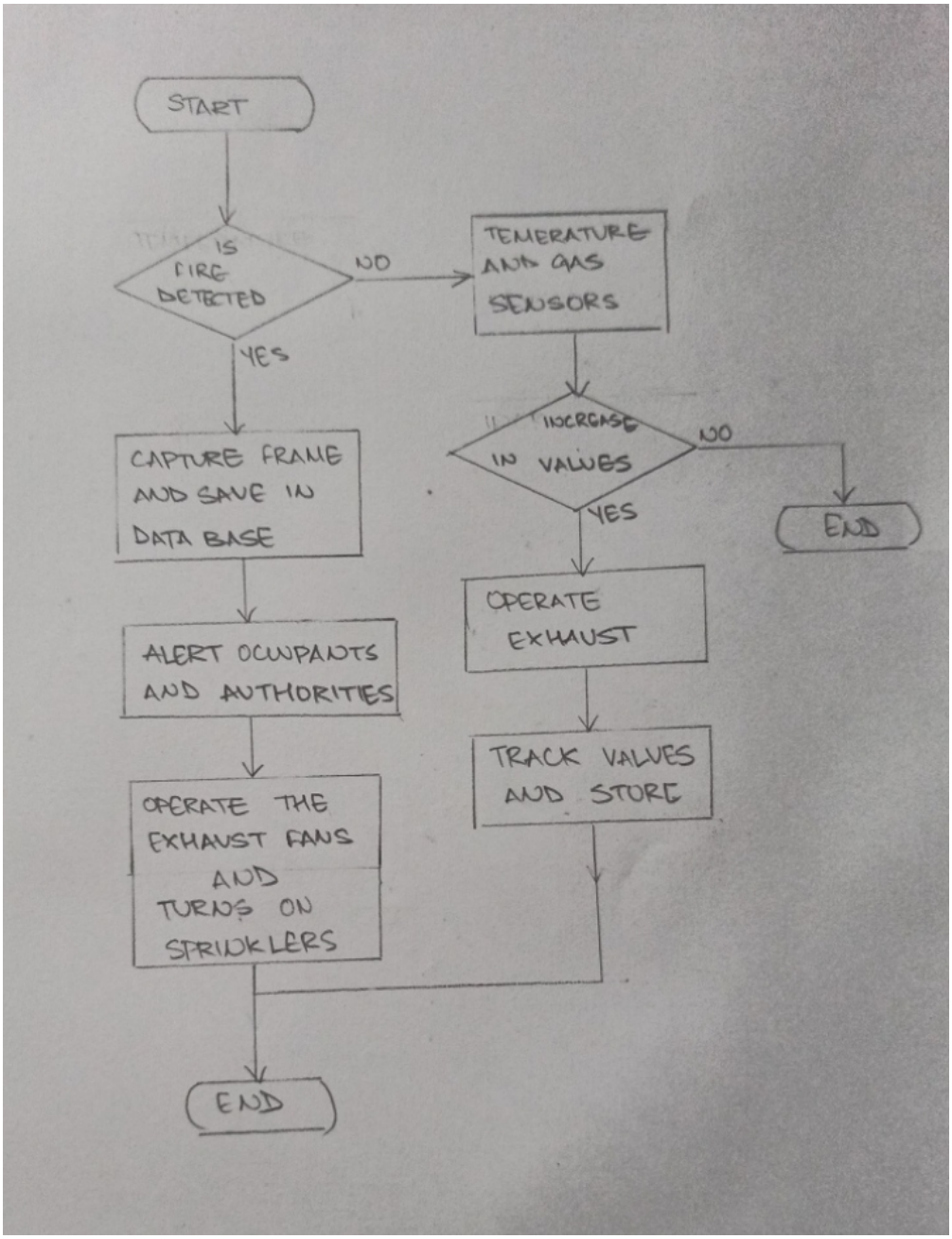
The python program send data to the IBM Platform.

the IBM Platform is connected to the NodeRed with processes data and send this data to the mobile app (MIT App Inventor) and displays this information. The infromation on the mobile app will be updated ever few seconds.

The IBM Platform also has fire detections software that processes incoming video and frames to process and detect fire, once the fire is detected, authorities and occupants

are alerted. The frames are store in a database for future use and investigation to the cause of the fire.

5. FLOWCHART



6. RESULT

This prototype is far from perfect, but as a basis for future development it surfices.

7. ADVANTAGES AND DISADVNATAGES

A. Advantages:

i. Can be used as a builing sturcture
ii. Opens upto more development
iii. Can be used commercially
iv. Many scopes of application

B. Disadvantages:

i. Many bugs
ii. Cannot, as of yet, be used commercially
iii. System is not very efficient

8. APPLICATIONS

a. Disiaster managemnt in ibararies
b. Schools
c. Small-scale firms
d. Rent-a-Offices
e. Domestic residences

9. CONCLUSION

This is one of the many basis of crisis management, both urban and rural, which has a huge scope of development.

10. FUTURE SCOPE

Crisis and disaster managemnt is one of the major sectors that can use this apploication and prototype. Aslo instustrial and mining industries need this for human safety and protection.

11. BIBLIOGRAPHY

1. https://www.hsa.ie/eng/Topics/Fire/Fire_Detection_and_Warning/
2. https://publicsafety.tufts.edu/firesafety/fire-alarm-response/
3. https://ehs.stanford.edu/reference/toxic-gas-alarm-systems
4. https://sunypoly.edu/sites/default/files/Research/Contractor%20Forms%20and%20Training/EHS-00049%20R8%20TGMS%20Evacuation%20Plan.pdf

APPENDIX

a. SOURCE CODE

1. Main Source:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#IBM Credentials
organization="unuahm"
deviceType="DeviceFire"
deviceId="F1"
authMethod="token"
authToken="Amiya_1Fireproject"


def myCommandCallback(cmd):
        print("Command received: %s." %cmd.data)
        print(type(cmd.data))
        i=cmd.data['command']
        if i=='lighton':
                print("Light is on.")
        elif i=='lightoff':
                print("Light is off.")
        elif i=='fanoff':
                print("Fan is switched off.")
        elif i=='fanon':
                print("Fan is switched on.")



try:
        deviceOptions={"org": organization, "type": deviceType, "id": deviceId, "auth-method":
authMethod, "auth-token": authToken}
        deviceCli=ibmiotf.device.Client(deviceOptions)

except Exception as e:
           print("Caught exception while connecting device: %s." %str(e))
```

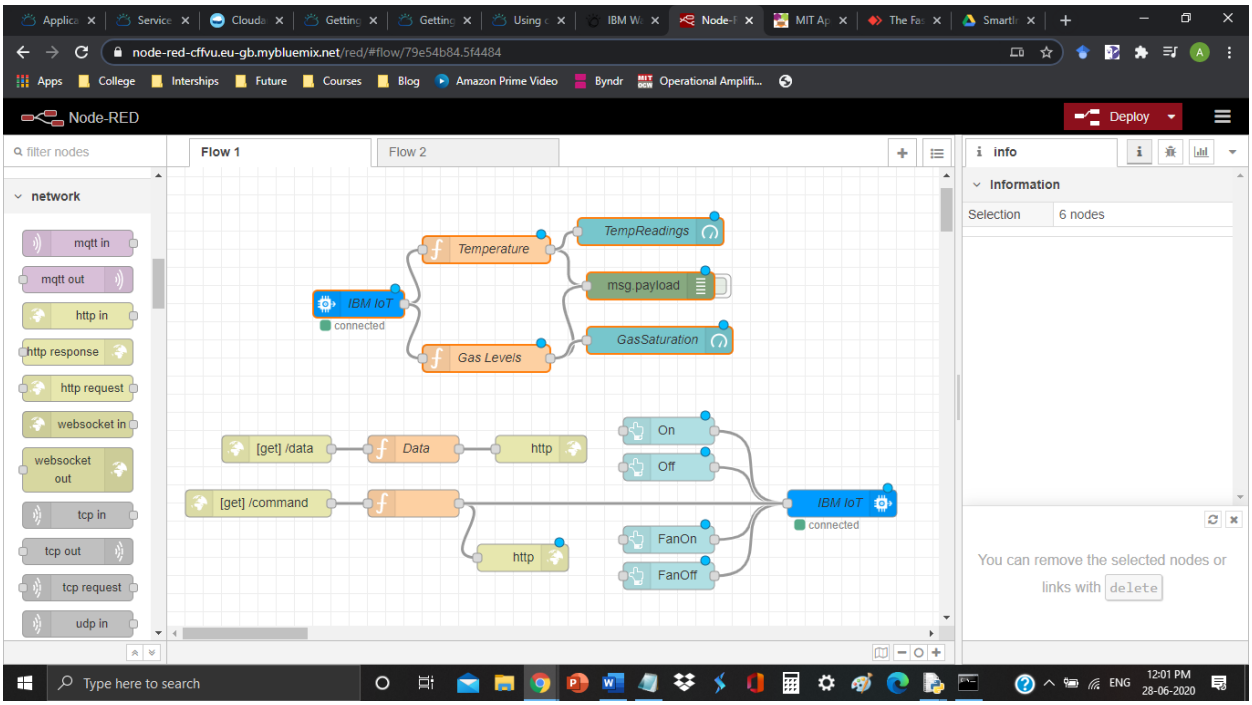
```
        sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()

while True:

    temp=random.randint(-20,110)
    gas=random.randint(1,100)
    #Send values to IBM
    data={'Temperature': temp, 'Gas': gas}
    #print
    def myOnPublishCallback():
      print ("Published Temperature = %s C" %temp, "Gas = %s %%" %gas, "to IBM Watson")

    success=deviceCli.publishEvent("CFFVU", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
      print("Not connected to IoT.")
    time.sleep(4)

deviceCli.commandCallback=myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

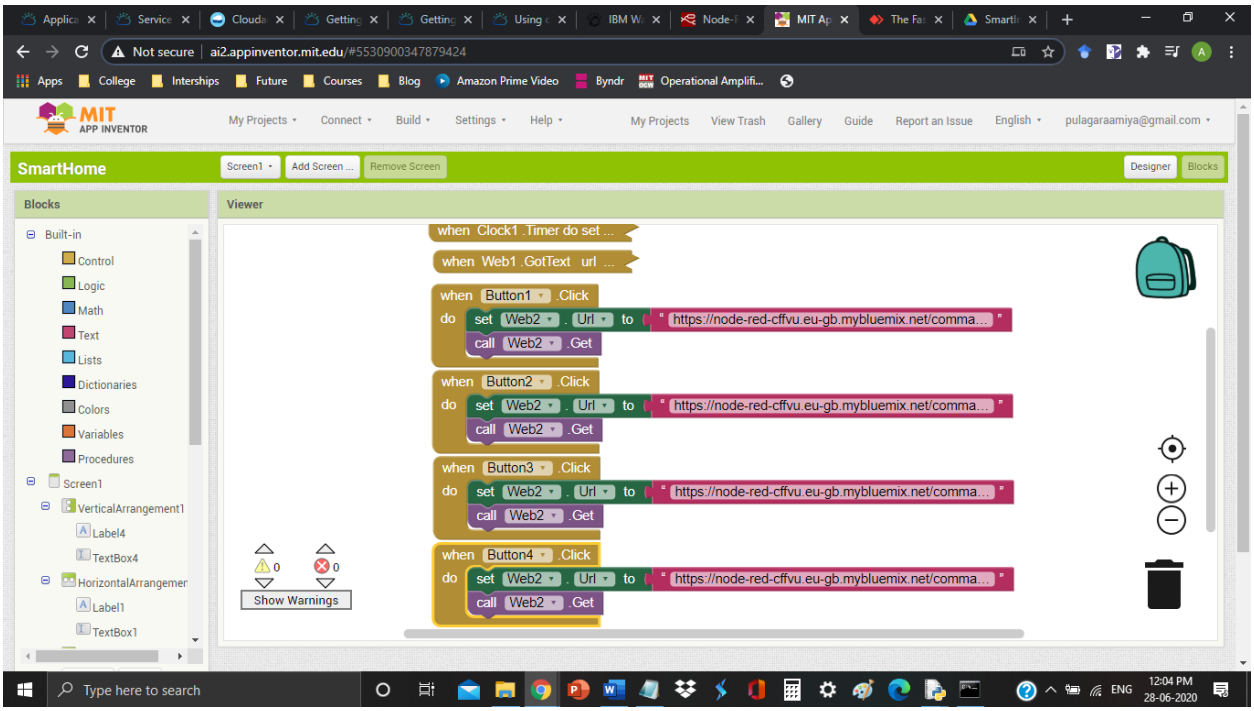
2. Image Recognition:

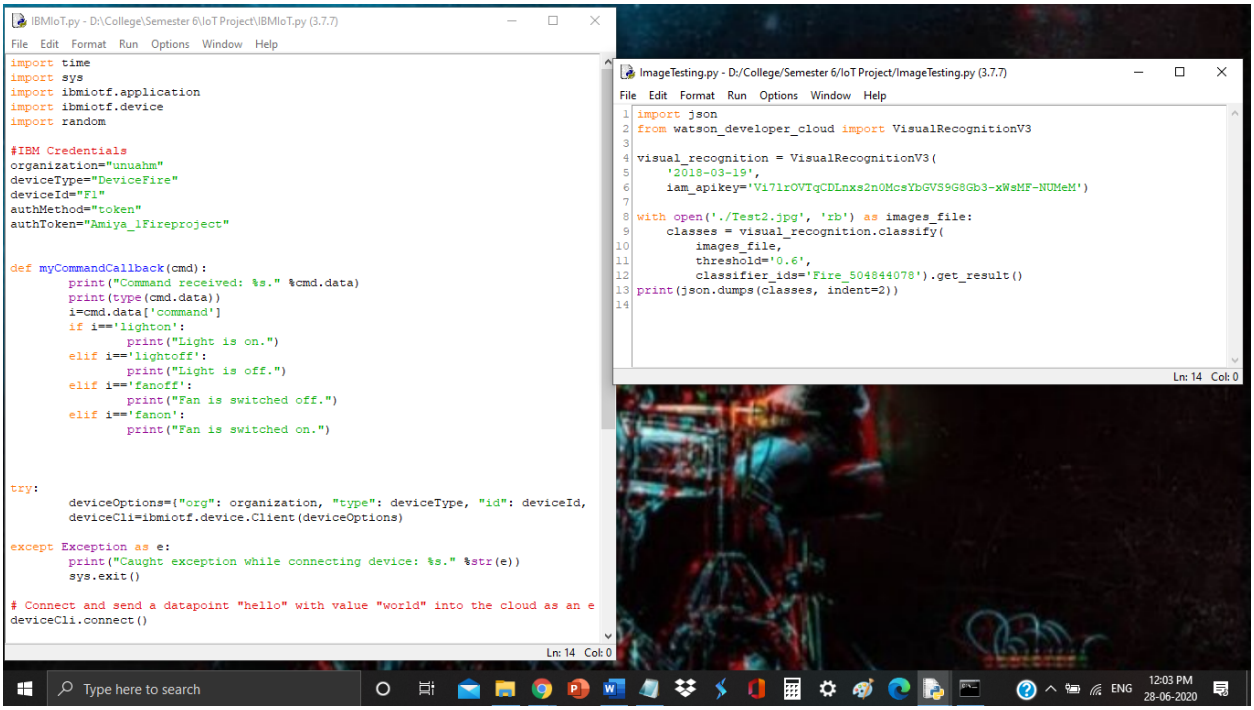
```
import json
from watson_developer_cloud import VisualRecognitionV3

visual_recognition = VisualRecognitionV3(
  '2018-03-19',
  iam_apikey='Vi7lrOVTqCDLnxs2n0McsYbGVS9G8Gb3-xWsMF-NUMeM')

with open('./Test2.jpg', 'rb') as images_file:
```

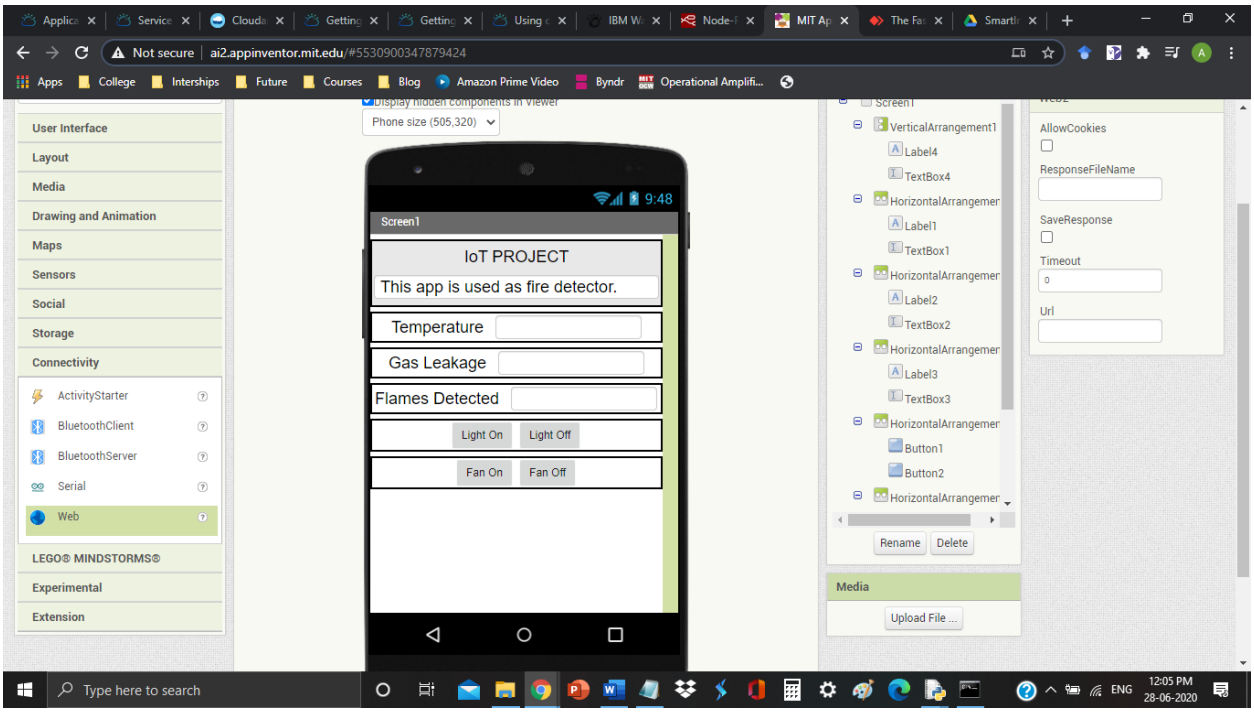
```
    classes = visual_recognition.classify(
        images_file,
        threshold='0.6',
            classifier_ids='Fire_504844078').get_result()
print(json.dumps(classes, indent=2))
```

PROOF




Not secure ai2.appinventor.mit.edu/#5530900347879424
MIT APP INVENTOR
My Projects Connect Build Settings Help My Projects View Trash Gallery Guide Report an Issue English pulagaraamiya@gmail.com
SmartHome
Screen1 Add Screen ... Remove Screen
Designer Blocks
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen1
VerticalArrangement1
Label4
TextBox4
HorizontalArrangemer
Label1
TextBox1
Show Warnings
when Clock1 .Timer do set ...
when Web1 .GotText url ...
when Button1 .Click
do set Web2 . Url to " https://node-red-cffvu.eu-gb.mybluemix.net/comma... "
call Web2 .Get
when Button2 .Click
do set Web2 . Url to " https://node-red-cffvu.eu-gb.mybluemix.net/comma... "
call Web2 .Get
when Button3 .Click
do set Web2 . Url to " https://node-red-cffvu.eu-gb.mybluemix.net/comma... "
call Web2 .Get
when Button4 .Click
do set Web2 . Url to " https://node-red-cffvu.eu-gb.mybluemix.net/comma... "
call Web2 .Get
Type here to search
ENG 12:04 PM 28-06-2020


IBMIoT.py - D:\College\Semester 6\IoT Project\IBMIoT.py (3.7.7)
File Edit Format Run Options Window Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#IBM Credentials
organization="unuahm"
deviceType="DeviceFire"
deviceId="F1"
authMethod="token"
authToken="Amiya_1Fireproject"
def myCommandCallback(cmd):
print("Command received: %s." %cmd.data)
print(type(cmd.data))
i=cmd.data['command']
if i=='lighton':
print("Light is on.")
elif i=='lightoff':
print("Light is off.")
elif i=='fanoff':
print("Fan is switched off.")
elif i=='fanon':
print("Fan is switched on.")
try:
deviceOptions={"org": organization, "type": deviceType, "id": deviceId,
deviceCli=ibmiotf.device.Client(deviceOptions)
except Exception as e:
print("Caught exception while connecting device: %s." %str(e))
sys.exit()
# Connect and send a datapoint "hello" with value "world" into the cloud as an e
deviceCli.connect()
Ln: 14 Col: 0
ImageTesting.py - D:/College/Semester 6/IoT Project/ImageTesting.py (3.7.7)
File Edit Format Run Options Window Help
import json
from watson_developer_cloud import VisualRecognitionV3
visual_recognition = VisualRecognitionV3(
'2018-03-19',
iam_apikey='Vi71rOVTqCDLnxs2n0McsYbGVS9G8Gb3-xWsMF-NUMeM')
with open('./Test2.jpg', 'rb') as images_file:
classes = visual_recognition.classify(
images_file,
threshold='0.6',
classifier_ids='Fire_504844078').get_result()
print(json.dumps(classes, indent=2))
Ln: 14 Col: 0
Type here to search
ENG 12:03 PM 28-06-2020


Not secure
ai2.appinventor.mit.edu/#5530900347879424
Apps
College
Interships
Future
Courses
Blog
Amazon Prime Video
Byndr
Operational Amplifi...
User Interface
Layout
Media
Drawing and Animation
Maps
Sensors
Social
Storage
Connectivity
ActivityStarter
BluetoothClient
BluetoothServer
Serial
Web
LEGO® MINDSTORMS®
Experimental
Extension
Phone size (505,320)
9:48
Screen1
IoT PROJECT
This app is used as fire detector.
Temperature
Gas Leakage
Flames Detected
Light On
Light Off
Fan On
Fan Off
VerticalArrangement1
Label4
TextBox4
HorizontalArrangemer
Label1
TextBox1
Label2
TextBox2
Label3
TextBox3
Button1
Button2
Rename
Delete
Media
Upload File ...
AllowCookies
ResponseFileName
SaveResponse
Timeout
Url
Type here to search
ENG
12:05 PM
28-06-2020