

TEAM ID	PNT2022TMID04587
PROJECT NAME	Industry - specific intelligent fire management system

PYTHON SCRIPT

```

ibmcodepy - E:\ibmcodepy (3.7.4)
File Edit Format Run Options Window Help

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

organization = "fwj3oq1"
deviceType = "RaspiMCU"
deviceId = "12345"
authMethod = "token"
authToken = "123456789"

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status = cmd.data['command']
    if status == "sprinkleron":
        print("sprinkler is on")
    elif status == "sprinkleroff":
        print("sprinkler is off")
    elif status == "fanon":
        print("fan is on")
    elif status == "fanoff":
        print("fan is off")
    else:
        print("Please send proper command")
    print(cmd)

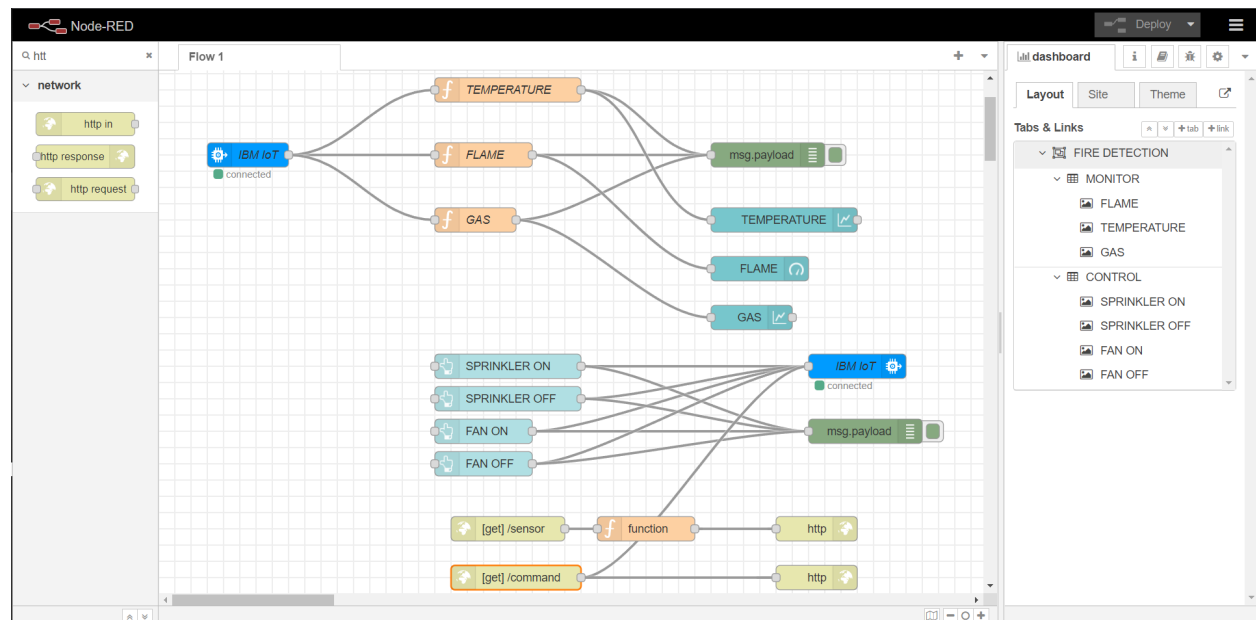
'''def myCommandCallback2(cmd):
    print("Command received: %s" % cmd.data['command'])
    status = cmd.data['command']
    if status == "fanon":
        print("fan is on")
    else:
        print("fan is off")
    print(cmd)'''

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 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()

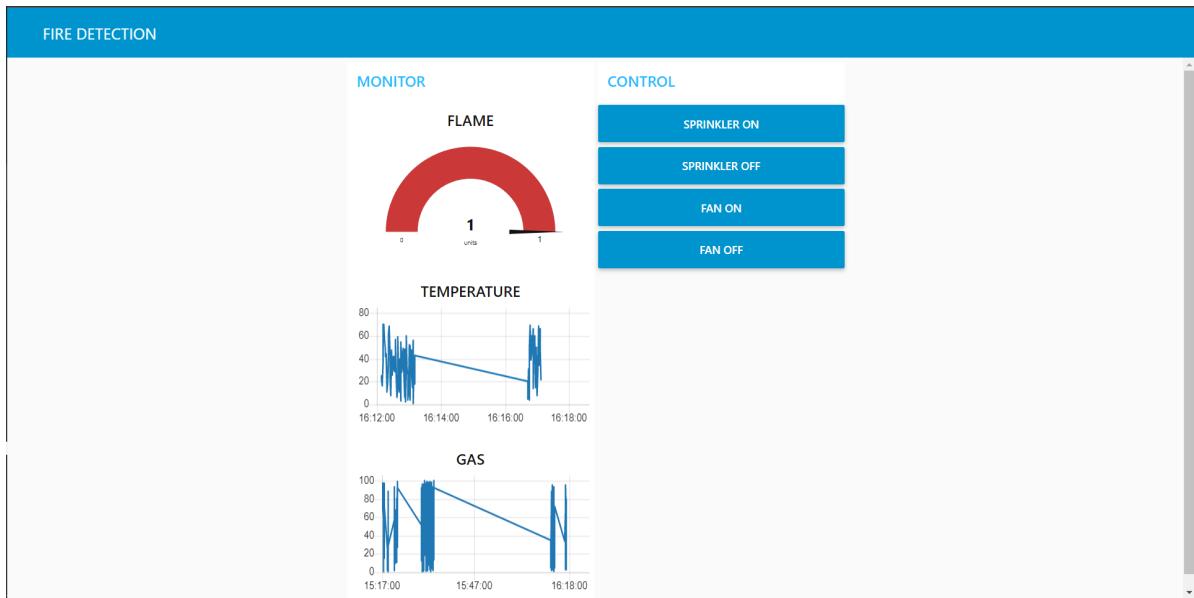
while True:

```

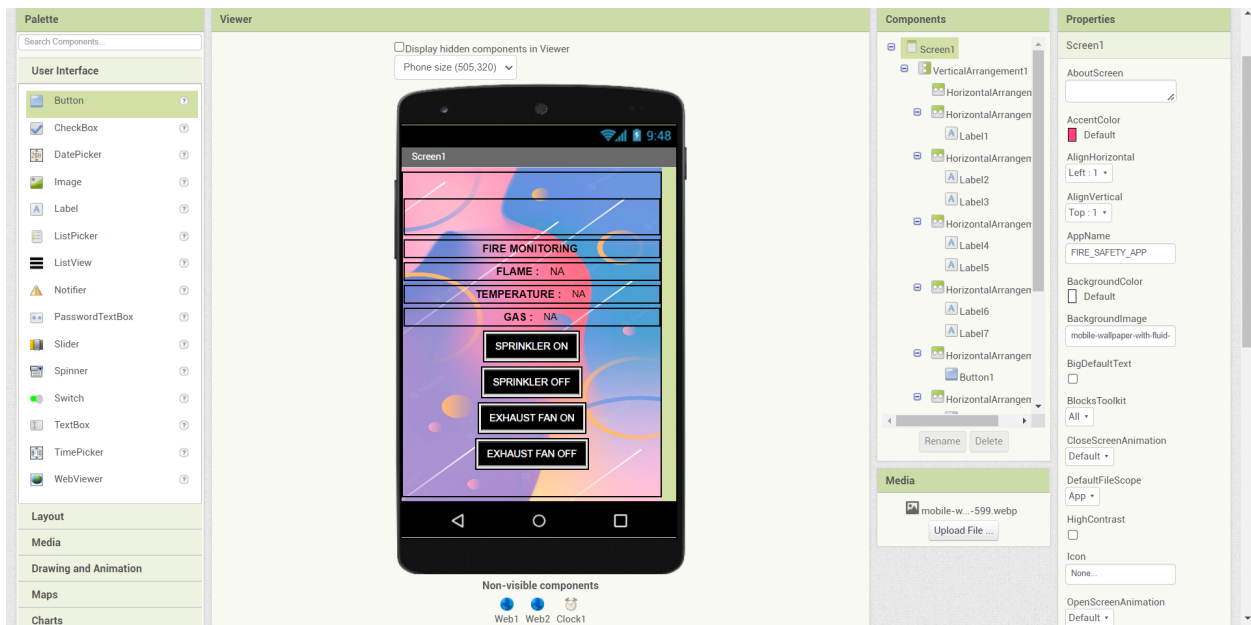
NODE-RED COMMANDS

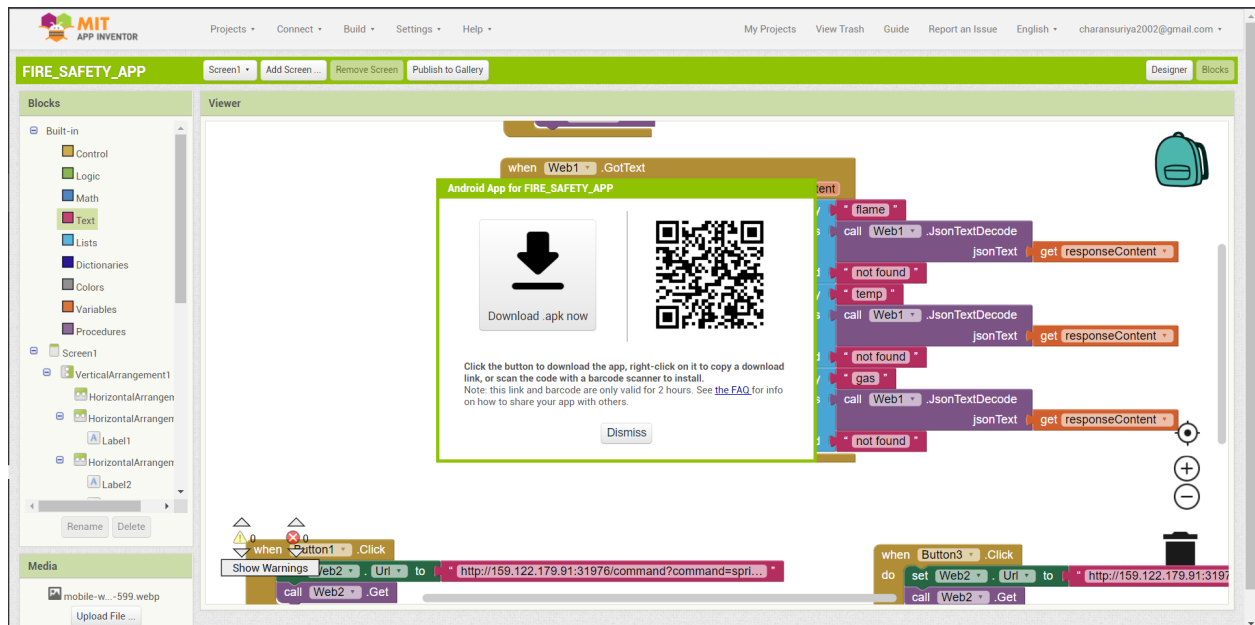
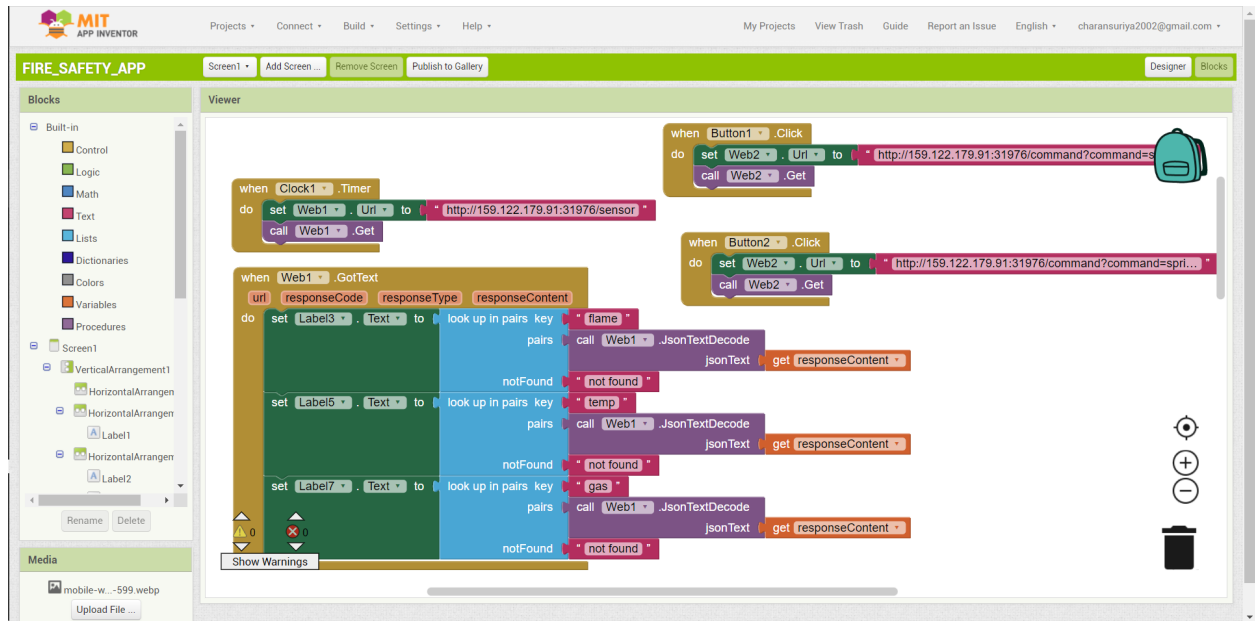


NODE-RED DASHBOARD



MIT APP BUILDER





4:23 PM | 5.1KB/s



Screen1

FIRE MONITORING

FLAME : 1

TEMPERATURE : 21

GAS : 81

SPRINKLER ON

SPRINKLER OFF

EXHAUST FAN ON

EXHAUST FAN OFF