Delivery Sprint-3

Industry-Specific Intelligent Fire Management System

TEAM ID: PNT2022TMID45189

Create a smart fire management system that includes a Gas sensor, Flame sensor and temperature sensors to detect any changes in the environment. Based on the temperature readings and if any Gases are present the exhaust fans are powered ON. If any flame is detected the sprinklers will be switched on automatically. Emergency alerts are notified to the authorities and Fire station.

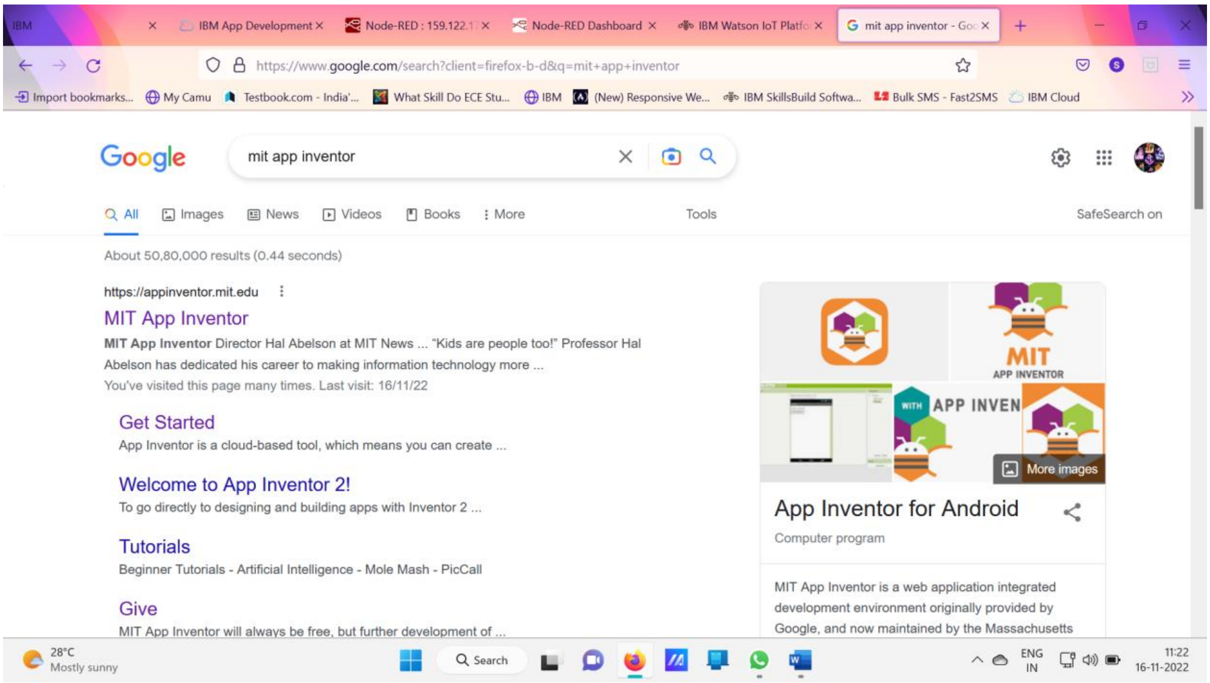
Sprint-3

MIT App Inventor, Dashboard (Application for your project using MIT App, Design the model and test the App)

STEPS

1) Create a account in the MIT App Inventor.
2) Then choose create apps and create a new project and name it.
3) Design the Designer and Blocks for your Requirement.
4) And connect with your MIT APP Companion in your phone (Install the MIT Companion using Playstore)
5) Finally run the program it shows the data to your mobile.

THE PROCESS



https://www.google.com/search?client=firefox-b-d&q=mit+app+inventor

Google

mit app inventor

About 50,80,000 results (0.44 seconds)

https://appinventor.mit.edu

MIT App Inventor

MIT App Inventor Director Hal Abelson at MIT News ... "Kids are people too!" Professor Hal Abelson has dedicated his career to making information technology more ...

You've visited this page many times. Last visit: 16/11/22

Get Started

App Inventor is a cloud-based tool, which means you can create ...

Welcome to App Inventor 2!

To go directly to designing and building apps with Inventor 2 ...

Tutorials

Beginner Tutorials - Artificial Intelligence - Mole Mash - PicCall

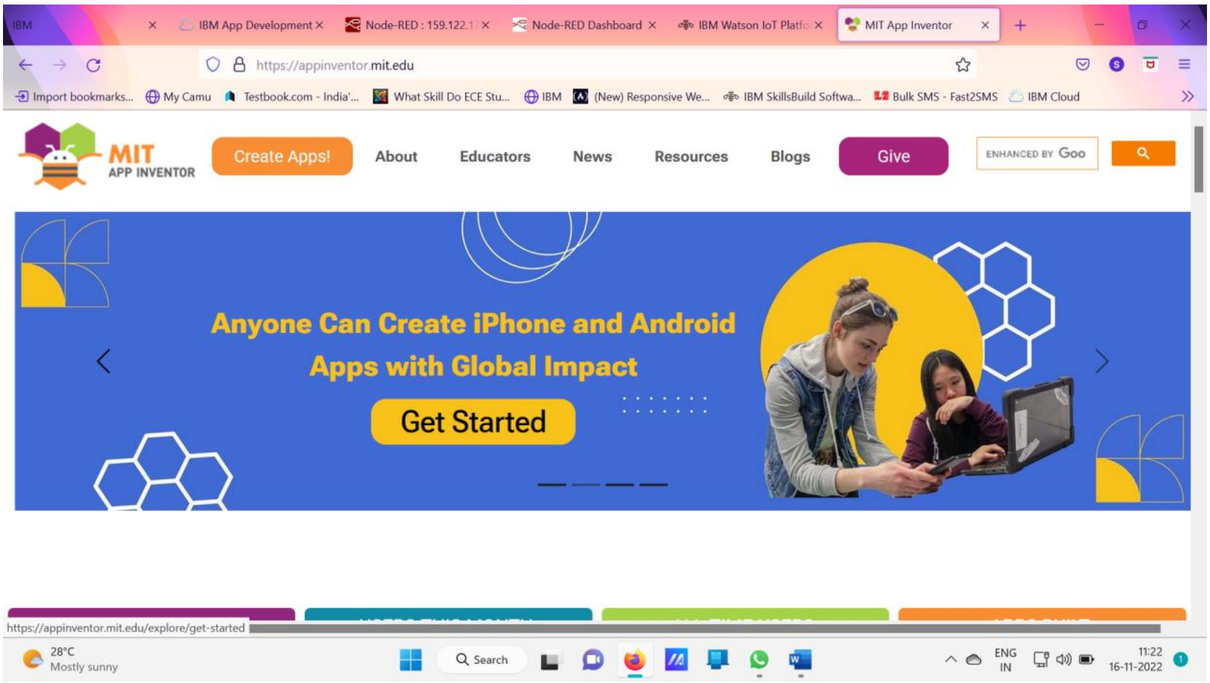
Give

MIT App Inventor will always be free, but further development of ...

App Inventor for Android

Computer program

MIT App Inventor is a web application integrated development environment originally provided by Google, and now maintained by the Massachusetts

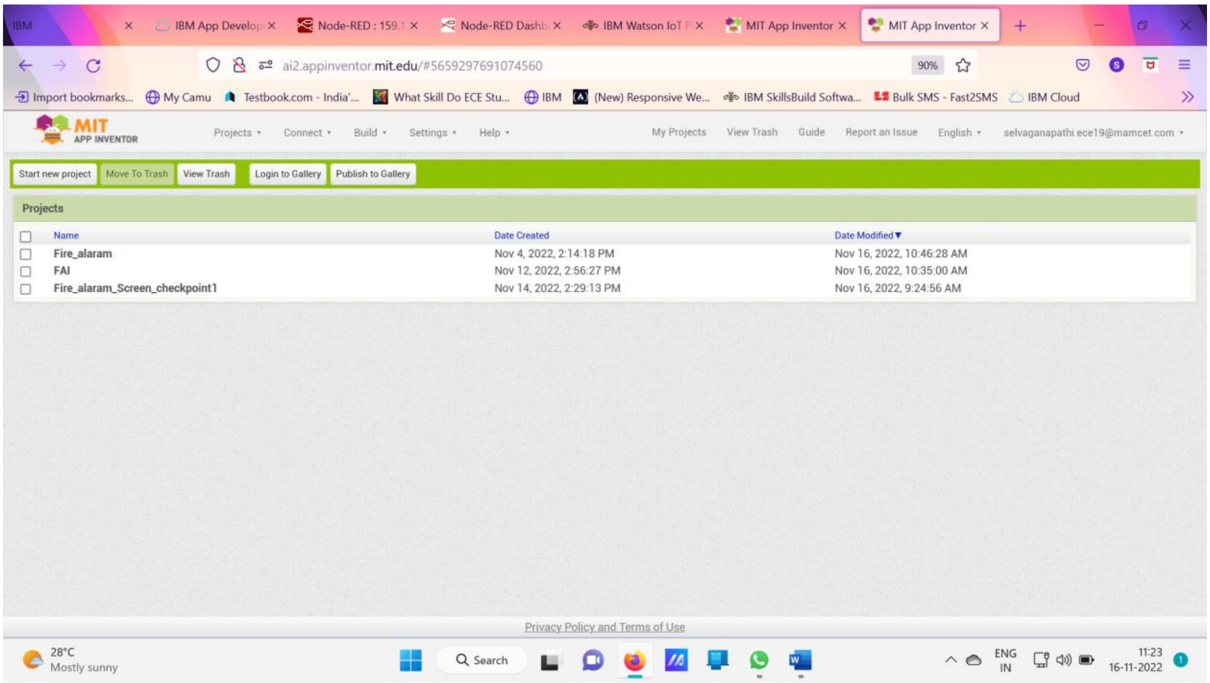


https://appinventor.mit.edu

Anyone Can Create iPhone and Android Apps with Global Impact

Get Started

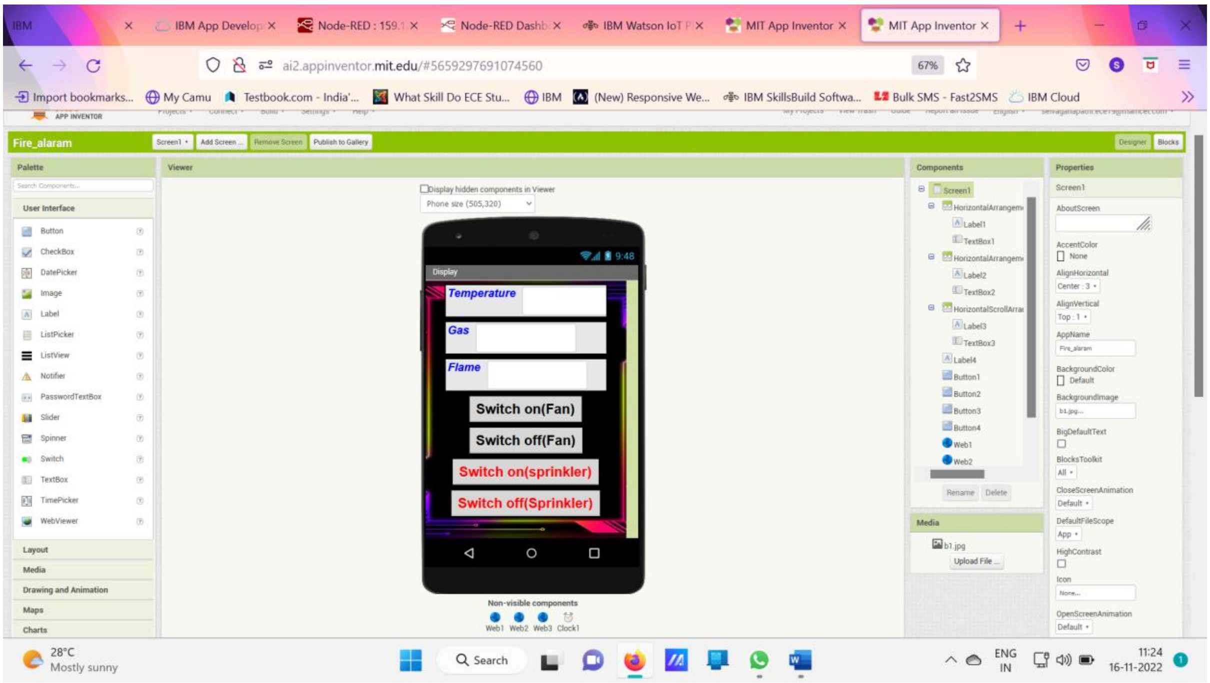
https://appinventor.mit.edu/explore/get-started



ai2.appinventor.mit.edu/#5659297691074560

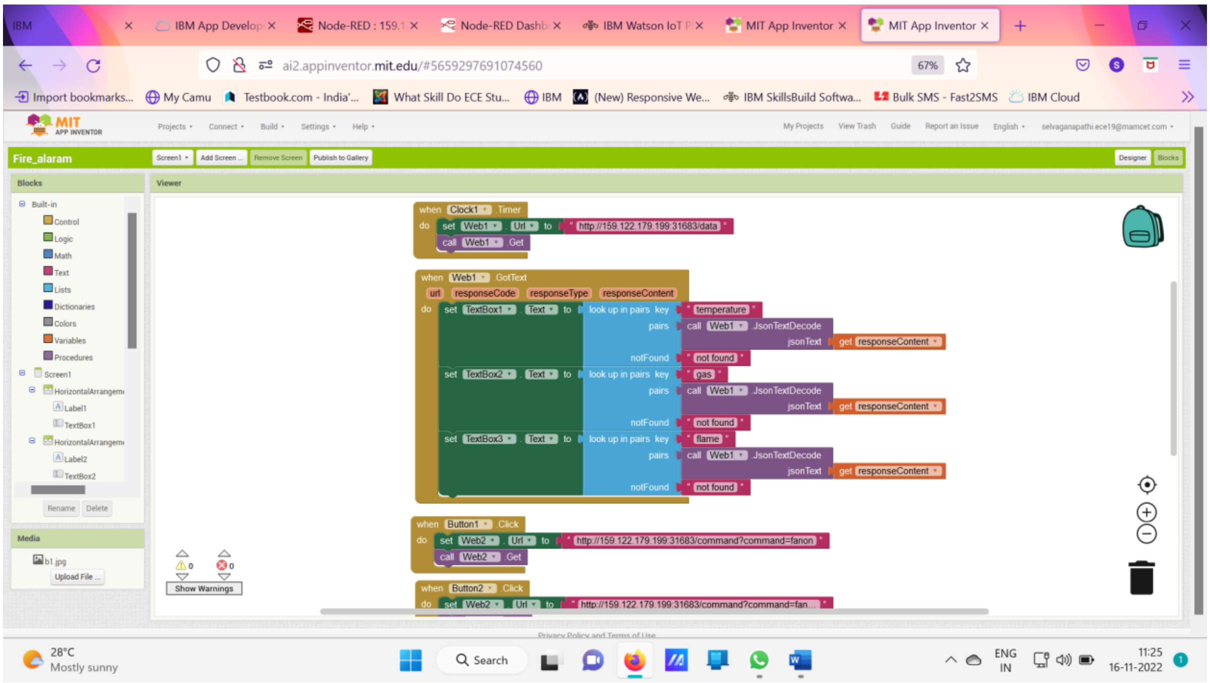
Projects

Name	Date Created	Date Modified ▼
Fire_alaram	Nov 4, 2022, 2:14:18 PM	Nov 16, 2022, 10:46:28 AM
FAI	Nov 12, 2022, 2:56:27 PM	Nov 16, 2022, 10:35:00 AM
Fire_alaram_Screen_checkpoint1	Nov 14, 2022, 2:29:13 PM	Nov 16, 2022, 9:24:56 AM

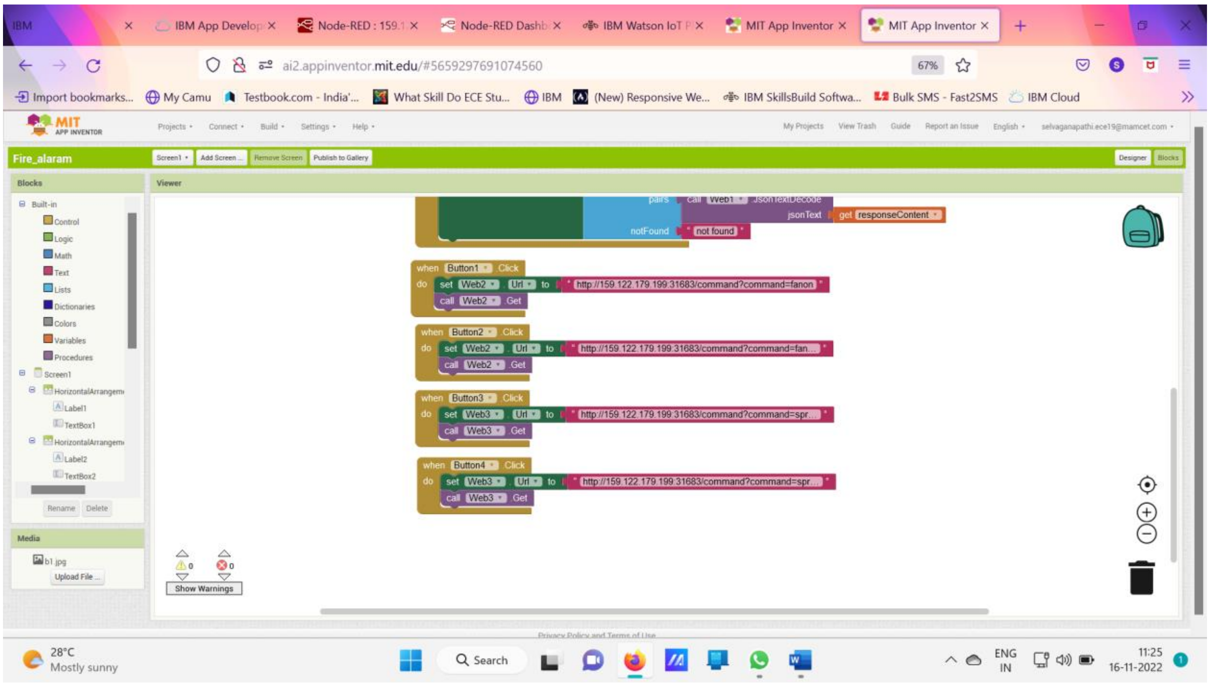


ai2.appinventor.mit.edu/#5659297691074560

Fire_alaram



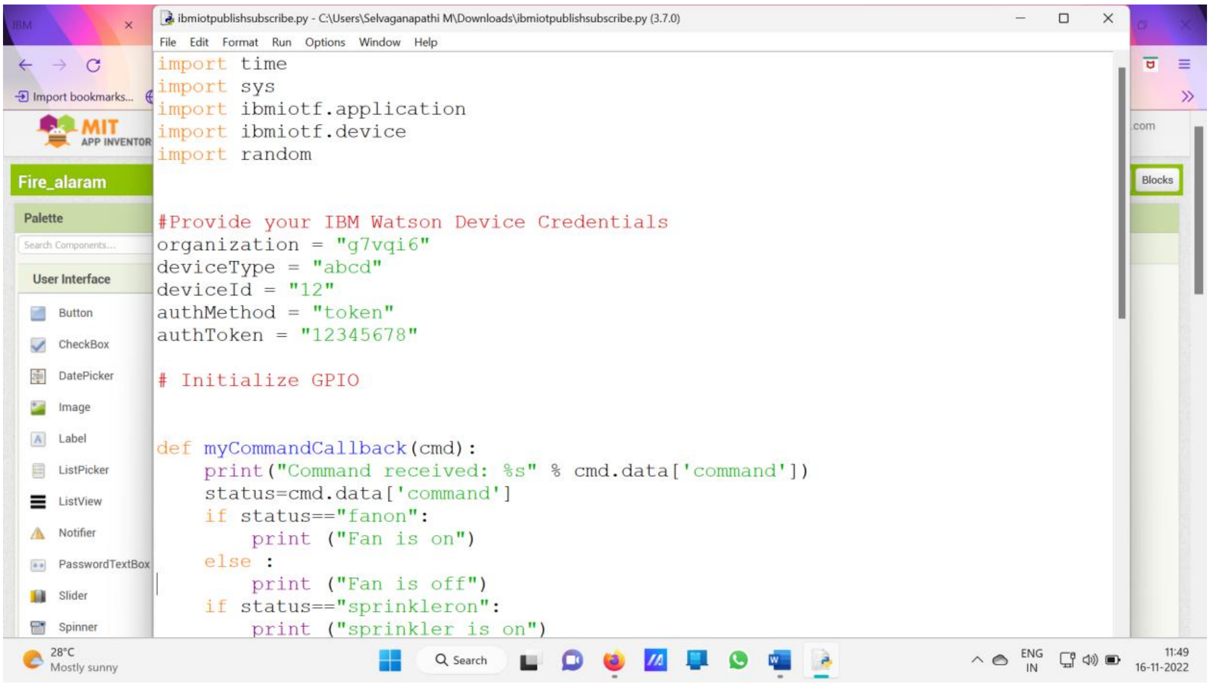
```
when Clock1 .Timer
do  set Web1 . Url to " http://159.122.179.199:31683/data "
    call Web1 .Get

when Web1 .GotText
 url  responseCode  responseType  responseContent
do  set TextBox1 . Text to  look up in pairs  key " temperature "
                                              pairs  call Web1 .JsonTextDecode
                                                          jsonText  get responseContent
                                           notFound " not found "
    set TextBox2 . Text to  look up in pairs  key " gas "
                                              pairs  call Web1 .JsonTextDecode
                                                          jsonText  get responseContent
                                           notFound " not found "
    set TextBox3 . Text to  look up in pairs  key " flame "
                                              pairs  call Web1 .JsonTextDecode
                                                          jsonText  get responseContent
                                           notFound " not found "

when Button1 .Click
do  set Web2 . Url to " http://159.122.179.199:31683/command?command=fanon "
    call Web2 .Get

when Button2 .Click
do  set Web2 . Url to " http://159.122.179.199:31683/command?command=fan... "
```



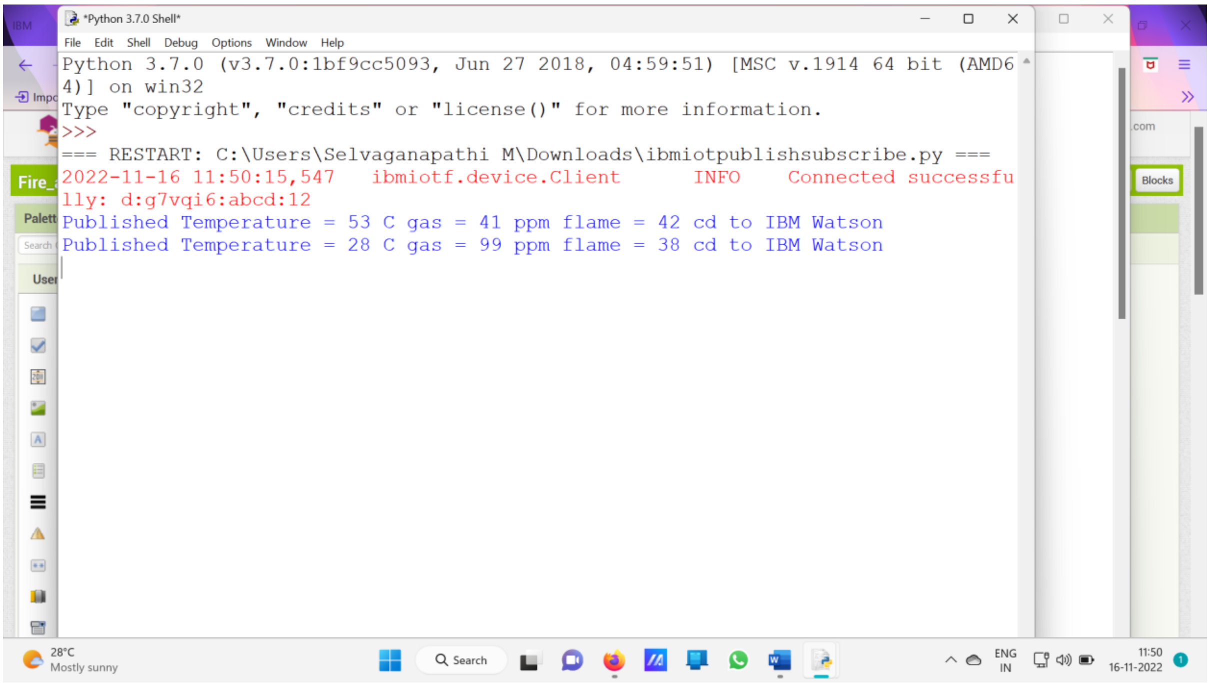
```
        pairs  call Web1 .JsonTextDecode
                    jsonText  get responseContent
     notFound " not found "

when Button1 .Click
do  set Web2 . Url to " http://159.122.179.199:31683/command?command=fanon "
    call Web2 .Get

when Button2 .Click
do  set Web2 . Url to " http://159.122.179.199:31683/command?command=fan... "
    call Web2 .Get

when Button3 .Click
do  set Web3 . Url to " http://159.122.179.199:31683/command?command=spr... "
    call Web3 .Get

when Button4 .Click
do  set Web3 . Url to " http://159.122.179.199:31683/command?command=spr... "
    call Web3 .Get
```

PYTHON PROGRAM

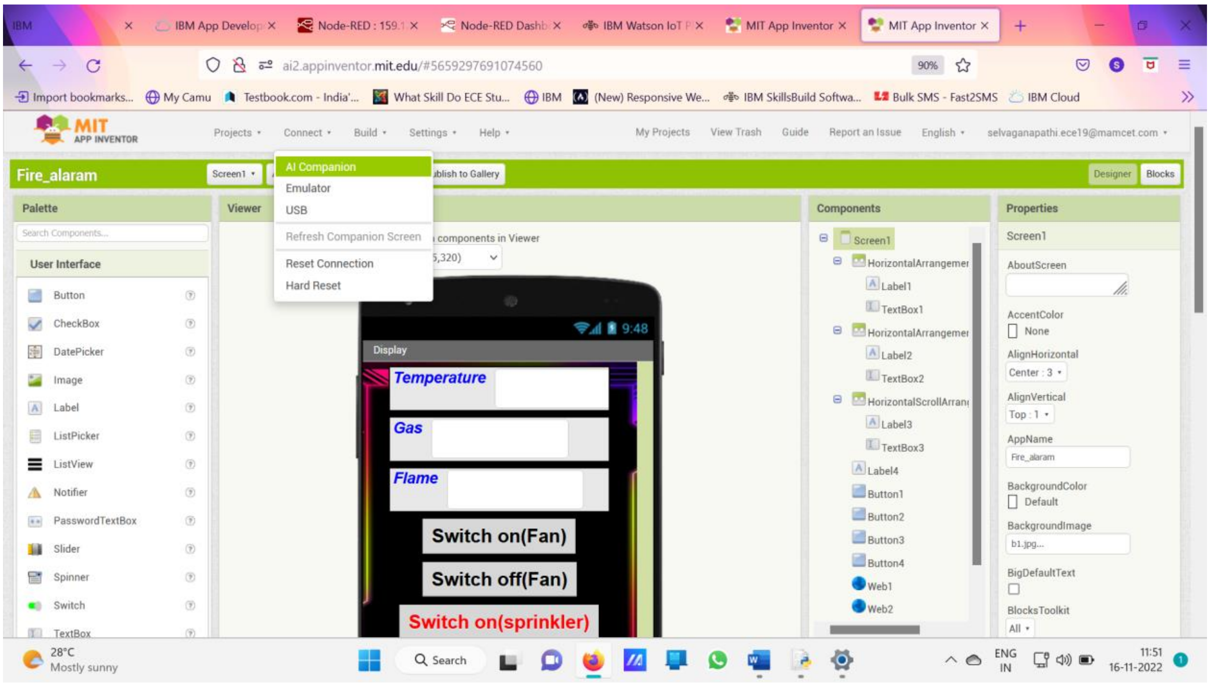
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random



#Provide your IBM Watson Device Credentials
organization = "g7vqi6"
deviceType = "abcd"
deviceId = "12"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO



def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="fanon":
        print ("Fan is on")
    else :
        print ("Fan is off")
    if status=="sprinkleron":
        print ("sprinkler is on")
    else :
        print ("sprinkler 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:
        #Get Sensor Data from DHT11

        temperature=random.randint(0,100)
        gas=str(random.randint(0,100))
        flame=str(random.randint(0,100))
        data = { 'temperature' : temperature, 'gas': gas, 'flame' : flame }
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temperature, "gas = %s ppm" % gas,"flame = %s cd" % flame, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(10)

        deviceCli.commandCallback = myCommandCallback

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

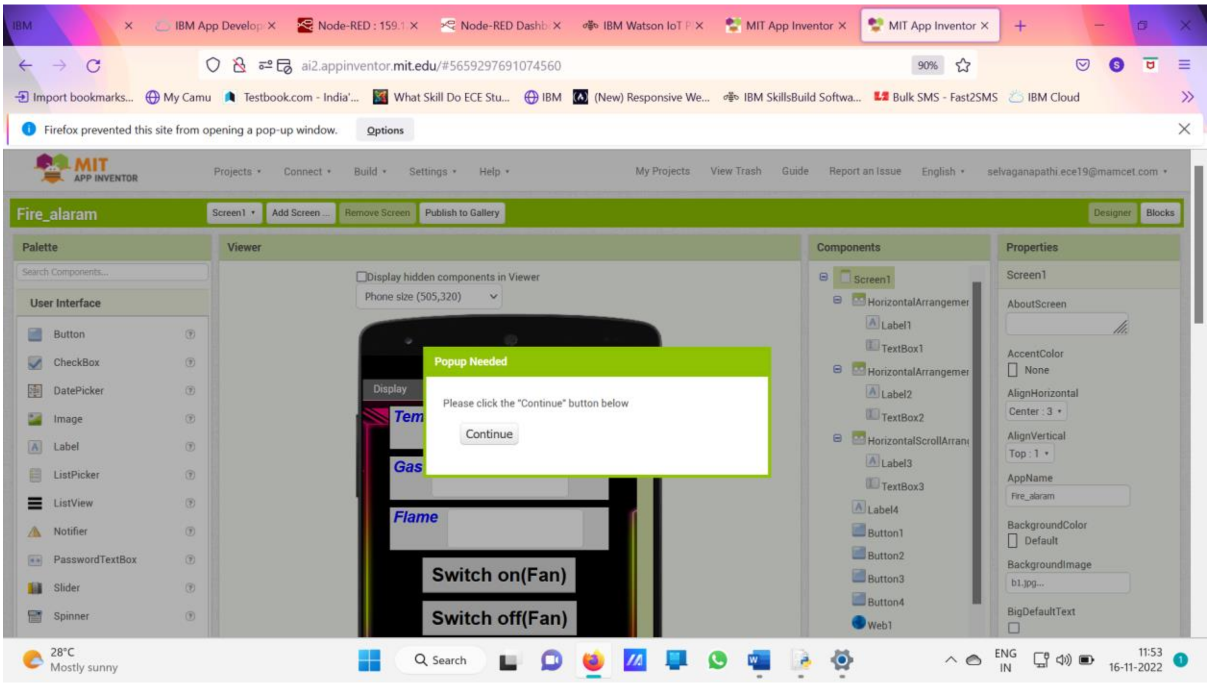


```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random


#Provide your IBM Watson Device Credentials
organization = "g7vqi6"
deviceType = "abcd"
deviceId = "12"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="fanon":
        print ("Fan is on")
    else :
        print ("Fan is off")
    if status=="sprinkleron":
        print ("sprinkler is on")
```



```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD6
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:\Users\Selvaganapathi M\Downloads\ibmiotpublishsubscribe.py ===
2022-11-16 11:50:15,547  ibmiotf.device.Client      INFO   Connected successfu
lly: d:g7vqi6:abcd:12
Published Temperature = 53 C gas = 41 ppm flame = 42 cd to IBM Watson
Published Temperature = 28 C gas = 99 ppm flame = 38 cd to IBM Watson
```





ai2.appinventor.mit.edu/#5659297691074560

Firefox prevented this site from opening a pop-up window.

Popup Needed

Please click the "Continue" button below

Continue

Screen1

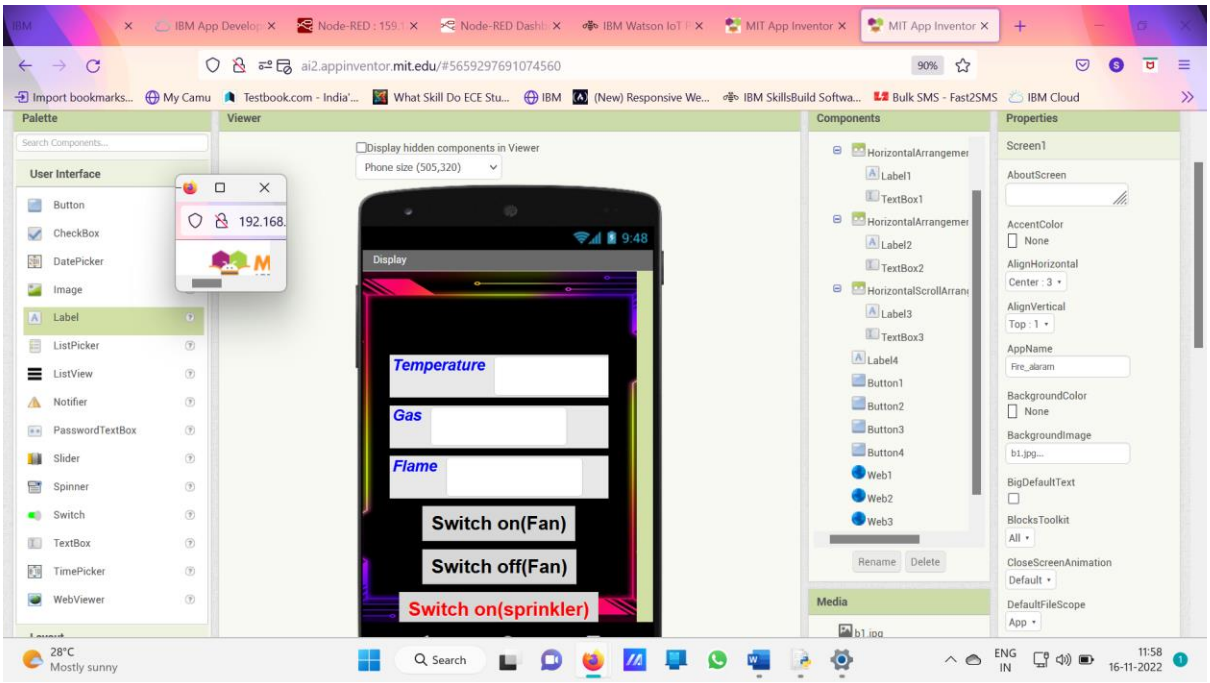
AppName: Fire_alaram

AlignHorizontal: Center : 3

AlignVertical: Top : 1

BackgroundColor: Default

BackgroundImage: b1.jpg...



ai2.appinventor.mit.edu/#5659297691074560

Screen1

AppName: Fire_alaram

AlignHorizontal: Center : 3

AlignVertical: Top : 1

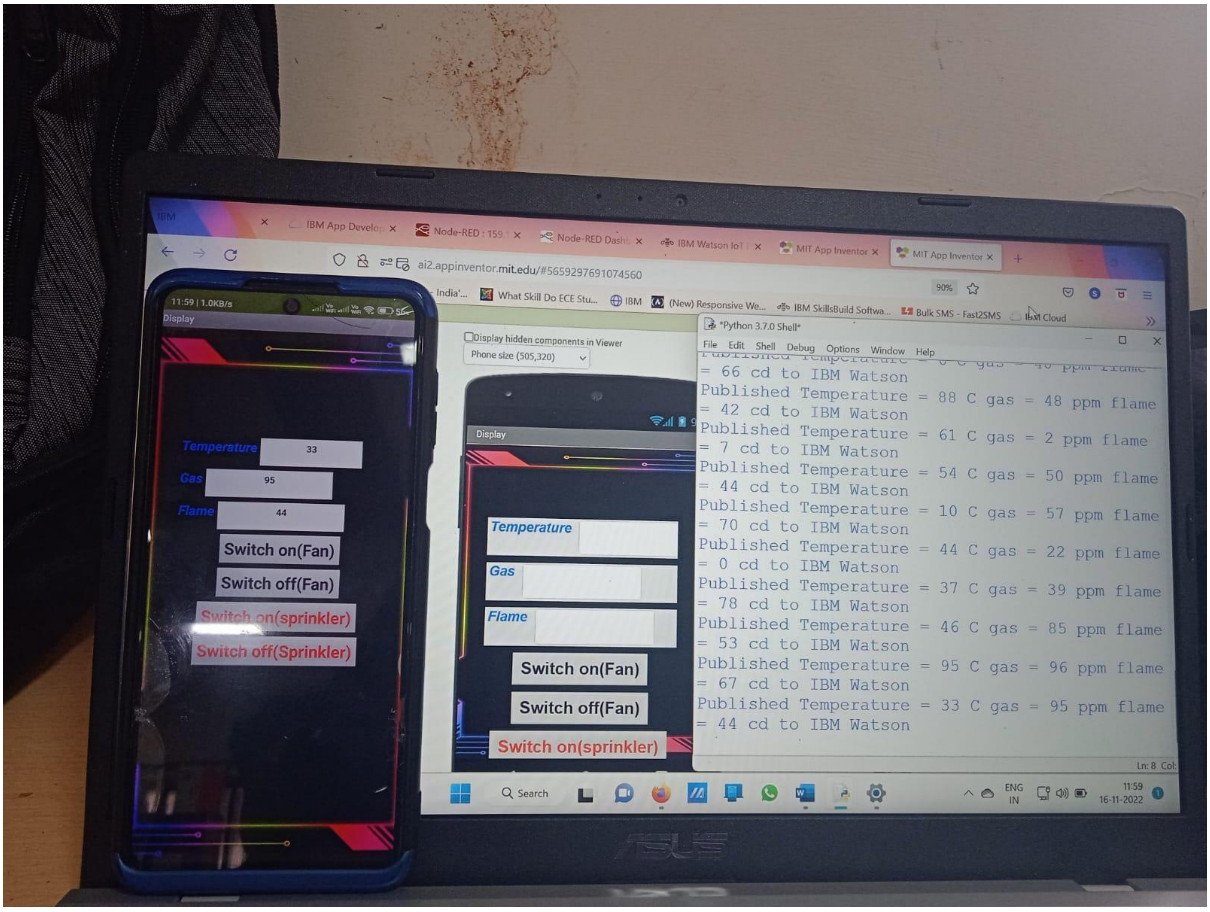
BackgroundColor: None

BackgroundImage: b1.jpg...

BlocksToolkit: All

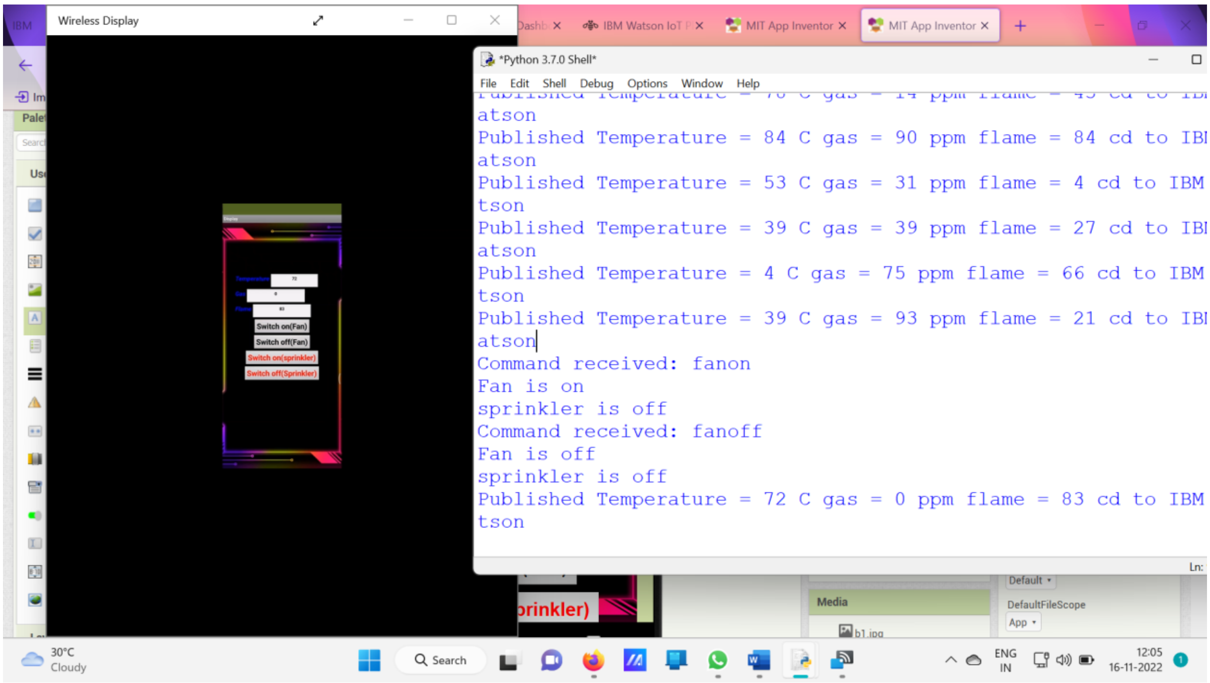
CloseScreenAnimation: Default

DefaultFileScope: App

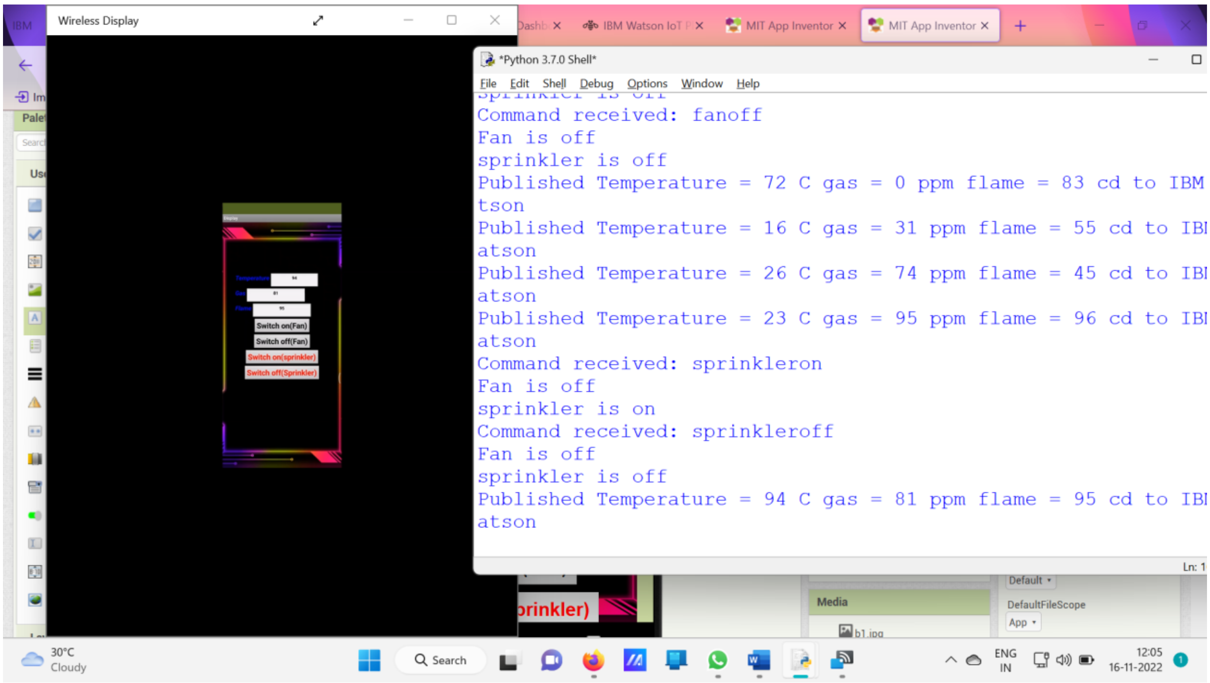


```
= 66 cd to IBM Watson
Published Temperature = 88 C gas = 48 ppm flame
= 42 cd to IBM Watson
Published Temperature = 61 C gas = 2 ppm flame
= 7 cd to IBM Watson
Published Temperature = 54 C gas = 50 ppm flame
= 44 cd to IBM Watson
Published Temperature = 10 C gas = 57 ppm flame
= 70 cd to IBM Watson
Published Temperature = 44 C gas = 22 ppm flame
= 0 cd to IBM Watson
Published Temperature = 37 C gas = 39 ppm flame
= 78 cd to IBM Watson
Published Temperature = 46 C gas = 85 ppm flame
= 53 cd to IBM Watson
Published Temperature = 95 C gas = 96 ppm flame
= 67 cd to IBM Watson
Published Temperature = 33 C gas = 95 ppm flame
= 44 cd to IBM Watson
```

Below is using the Wireless Display to Show the Output



```
atson
Published Temperature = 84 C gas = 90 ppm flame = 84 cd to IBI
atson
Published Temperature = 53 C gas = 31 ppm flame = 4 cd to IBM
tson
Published Temperature = 39 C gas = 39 ppm flame = 27 cd to IBI
atson
Published Temperature = 4 C gas = 75 ppm flame = 66 cd to IBM
tson
Published Temperature = 39 C gas = 93 ppm flame = 21 cd to IBI
atson
Command received: fanon
Fan is on
sprinkler is off
Command received: fanoff
Fan is off
sprinkler is off
Published Temperature = 72 C gas = 0 ppm flame = 83 cd to IBM
tson
```



```
sprinkler is off
Command received: fanoff
Fan is off
sprinkler is off
Published Temperature = 72 C gas = 0 ppm flame = 83 cd to IBM
tson
Published Temperature = 16 C gas = 31 ppm flame = 55 cd to IBI
atson
Published Temperature = 26 C gas = 74 ppm flame = 45 cd to IBI
atson
Published Temperature = 23 C gas = 95 ppm flame = 96 cd to IBI
atson
Command received: sprinkleron
Fan is off
sprinkler is on
Command received: sprinkleroff
Fan is off
sprinkler is off
Published Temperature = 94 C gas = 81 ppm flame = 95 cd to IBI
atson
```