SPRINT-3

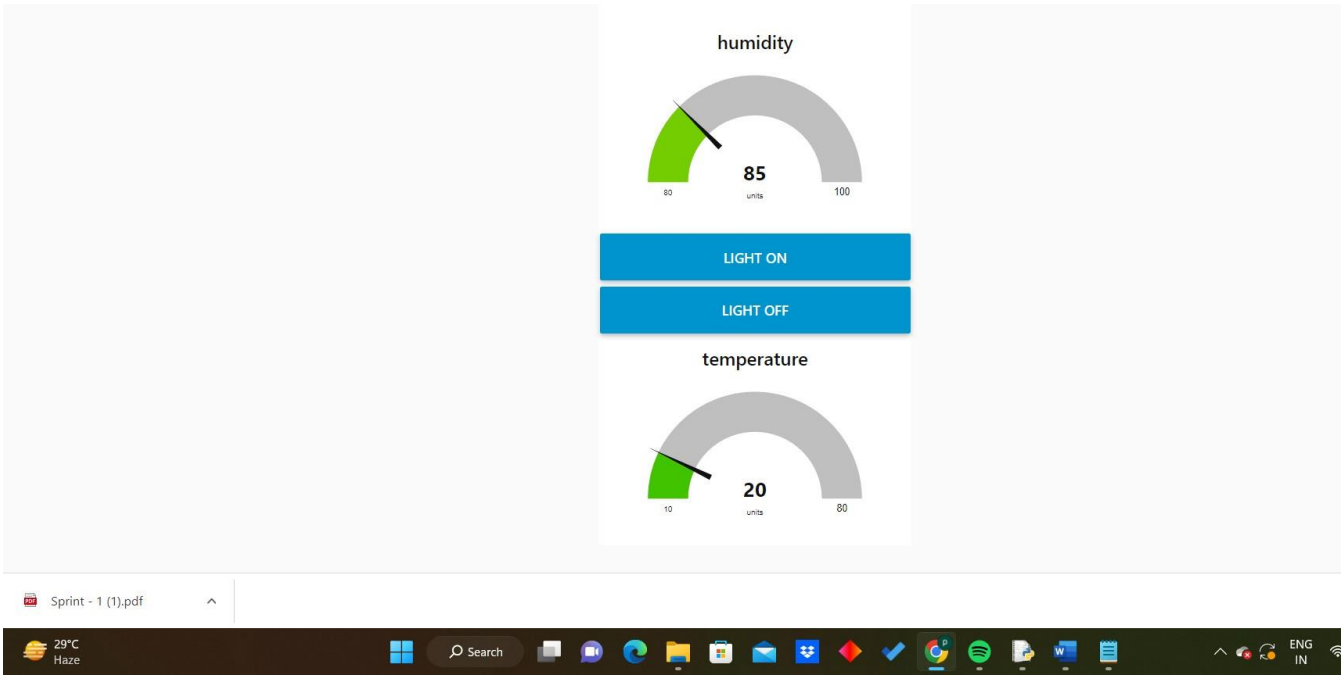
Team ID	PNT2022TMID14695
Project Name	Hazardous Area Monitoring for industrial Plant powered by IoT

Python code for the Temperature Alert and Humidity check

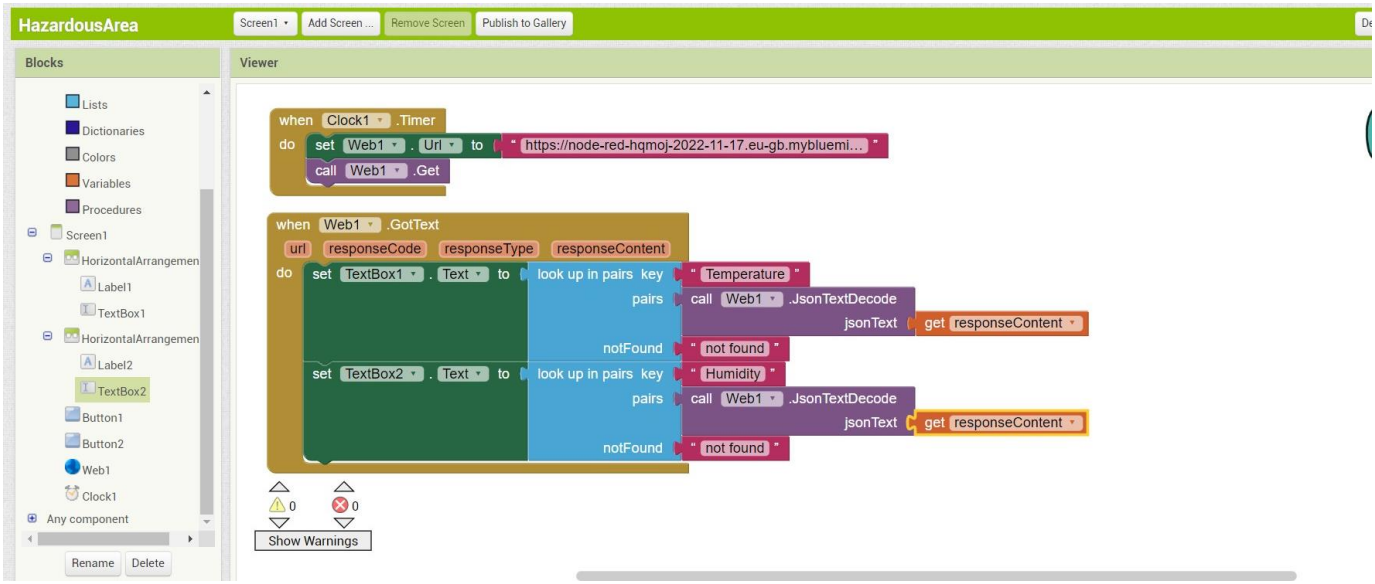
```
import time import sys

import ibmiotf.application

import ibmiotf.device import

random

# Initialize GPIO

#Provide your IBM Watson Device

Credentials organization = "ID3lhmfj"

deviceType = "efgh"

deviceId = " 56789"

authMethod = "use-token-auth"

authToken = "123456789"

def myCommandCallback(cmd):

print("Command received: %s" %

cmd.data['command'])

Status=cmd.data['command']

if Status=="Alert": print("Alert")

#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()
```

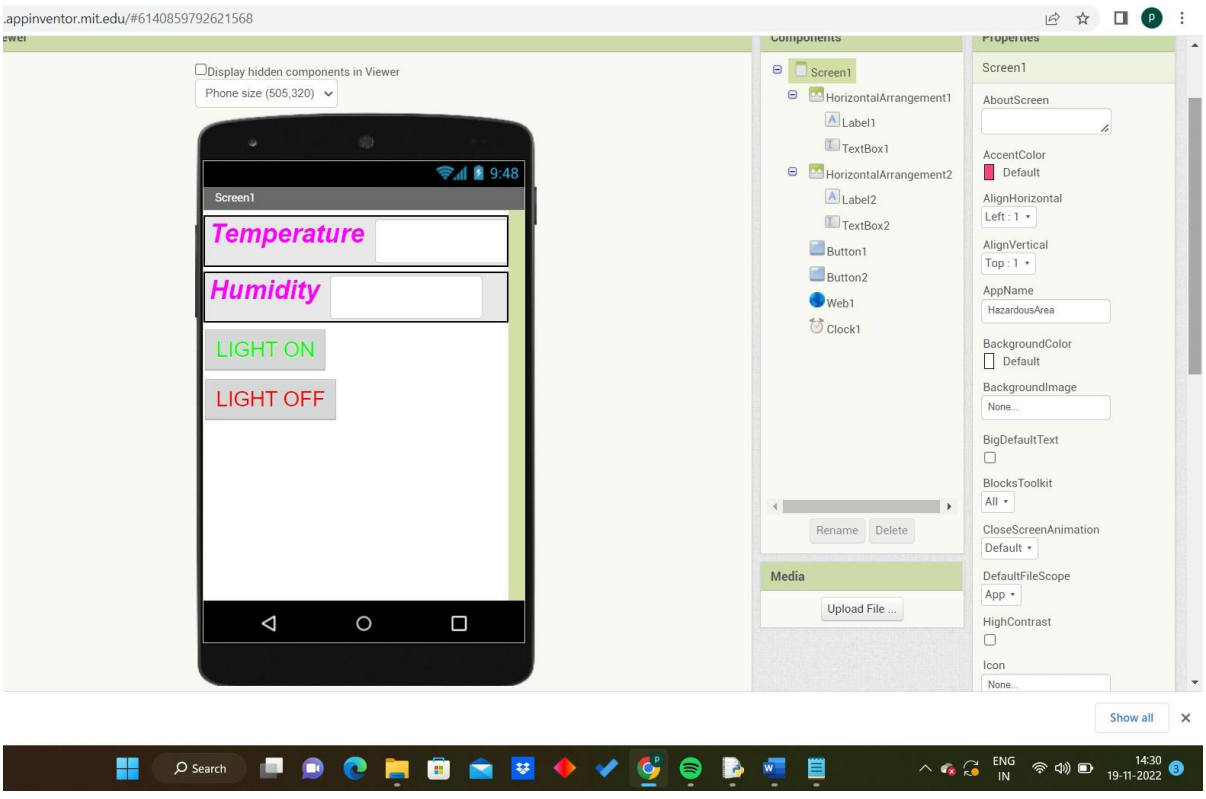
```
# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times

deviceCli.connect()

while True:

#Get Sensor Data from DHT11

temp =random.randint(0,100)

humid =random.randint(0,100)

oxygen =random.randint(0,100)


data = { 'temp' : temp, 'humidity': humid ,'oxygen': oxygen} data1 = {

'High temperature' : temp>60}

#print data

Def

myOnPublishCallback():

print ("Published Temperature = %s C" % temp, "humidity = %s %%" % humid,"alert", "to IBM Watson")
success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)


 if not success:

 print("Not connected to IoTF")

time.sleep(1)

deviceCli.commandCallback =

myCommandCallback


# Disconnect the device and application from the cloud

deviceCli.disconnect()
```

UI Dashboard


humidity
85
80
100
units
LIGHT ON
LIGHT OFF
temperature
20
10
80
units
Sprint - 1 (1).pdf
29°C
Haze
Search
ENG
IN

Design the application for the project using MIT App Inventor





LIGHT ON AND LIGHT OFF COMMAND