

Safety Gadget for Child Safety Monitoring and Notification

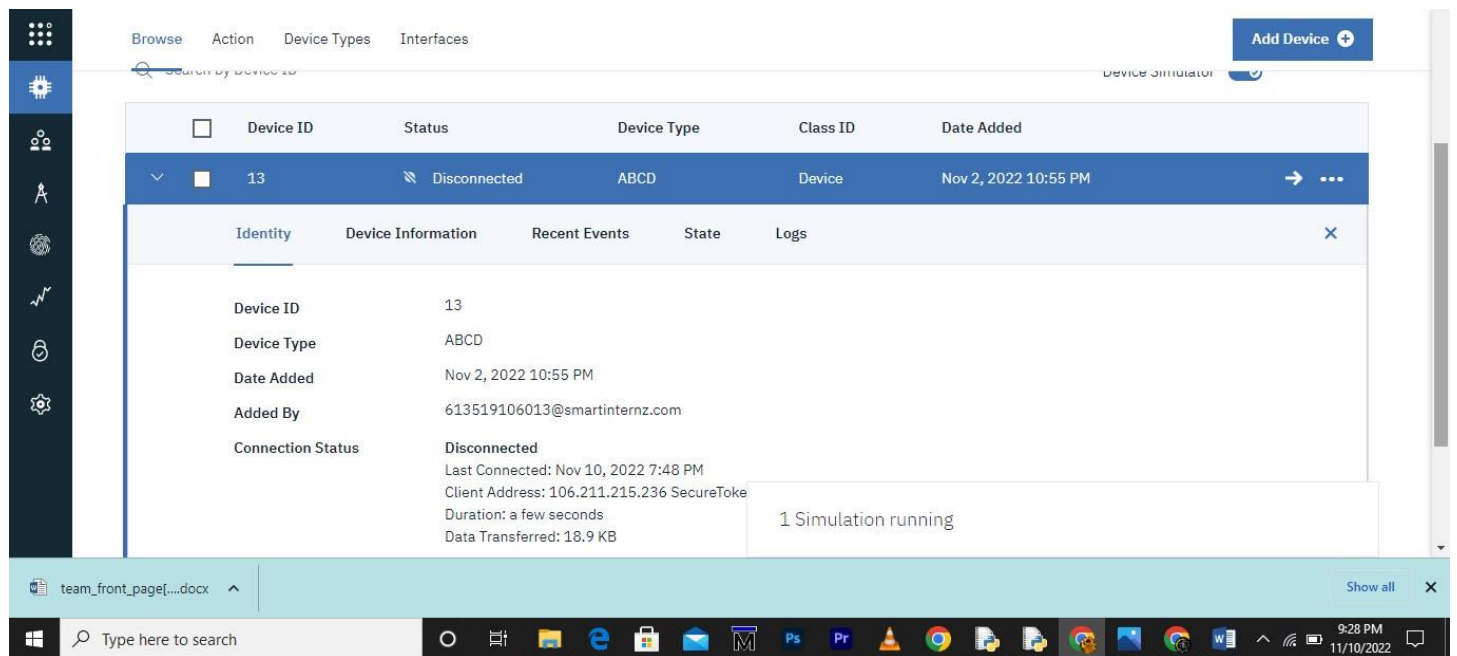
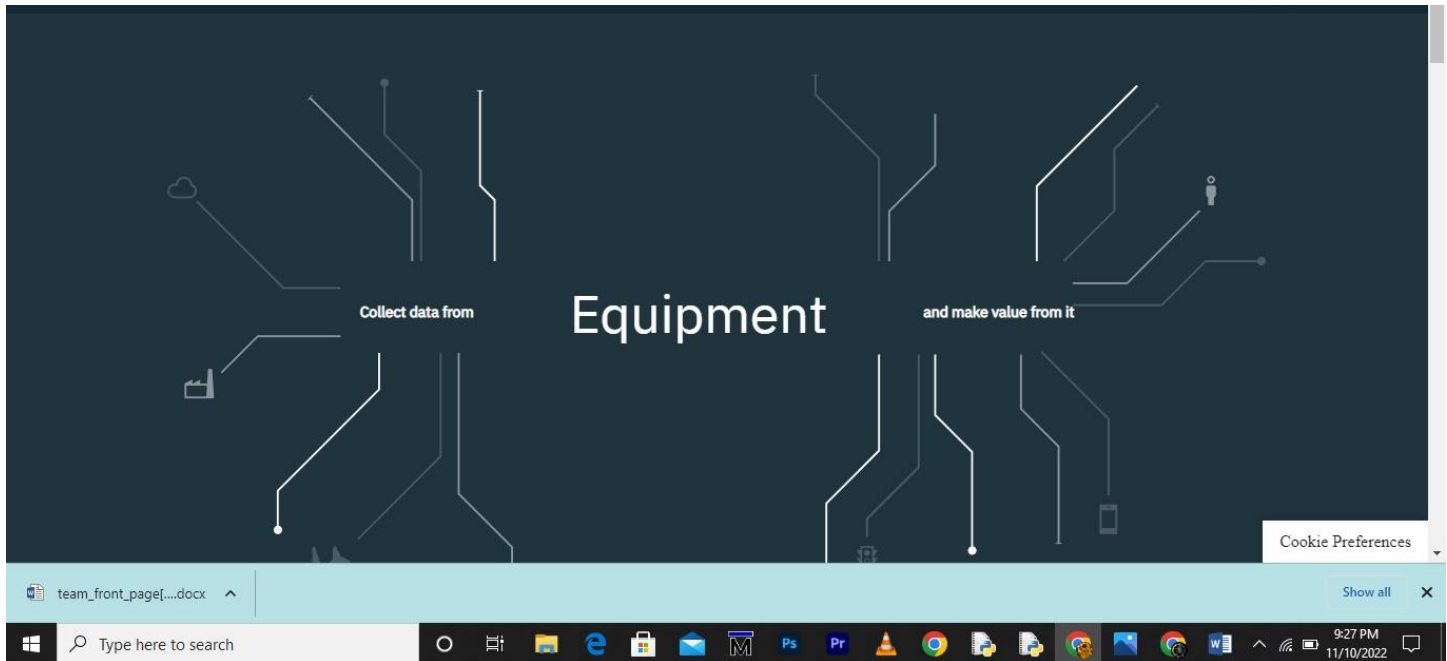
IBM NALAIYATHIRAN

Project Development –Delivery of Sprint 1

Creating and Connecting IBM cloud for Project and Python Code

| | |
|--------------------|--|
| TITLE | IOT based child safety gadget for child safety monitoring and notification |
| DOMAIN NAME | INTERNET OF THINGS |
| TEAM ID | PNT2022TMID39048 |

Creating IBM Cloud Service and creating the device:



Creating Python Code:

```
import time
```

```
import sys
```

```

import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "zwx6lb" deviceType = "ABCD"
deviceId = "13" authMethod = "token"
authToken = "12345678"

#api key {a-illza1-mbdxqo6z0s}
#api token {zSYzISuAWF&F_x7GkT}

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
print("power      on      ")
print("checking connection to
waston  iot...")  time.sleep(2)
deviceCli.connect()

print("dear user ... welcome to IBM-IOT ")

print("i can provide your children live location and temperature ") print()

```

```
name=str(input("enter your child name:")) while
True:
```

```
    temperature=random.randint(20,50)#random temperature for your child
latitude=random.uniform(10.781377,10.78643)#random latitude for your child
longitude=random.uniform(79.129113,79.134014)#random longitude for your child
a="Child inside the geofence"    b=" Child outside the geofence"    c="High
temperature"    d="Low temperature"    x={'your_child_Zone':a}
y={'your_child_Zone':b}    z={'temp_condition':c}    w={'temp_condition':d}
```

```
    data = { 'temp' : temperature, 'lat': latitude, 'lon':longitude, 'name':name }
    #print data    def
myOnPublishCallback():
print ("Published Temperature =
%s C" % temperature, "latitude
= %s %" % latitude,
"longitude = %s %" % longitude, "to IBM Watson")
print("\n")
    success = deviceCli.publishEvent("IoTSensorgpsdata", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if latitude>=10.78200 and latitude<=10.786000 and longitude >=79.130000 and longitude
<=79.133000:
```

```
deviceCli.publishEvent("IoTSensorgpsdata","json",data=x,qos=0,on_publish=myOnPublishCallb
ack)
```

```
    print(x)
```

```
print("\n")    else:
```

```
deviceCli.publishEvent("IoTSensorgpsdata","json",data=y,qos=0,on_publish=myOnPublishCallb
ack)
```

```
    print(y)
```

```
print("\n")
```

```
    if (temperature>35):
```

```
deviceCli.publishEvent("IoTSensorgpsdata","json",data=z,qos=0,on_publish=myOnPublishCallb
ack)
```

```
    print(c)
```

```
print("\n")    else:
```

```
deviceCli.publishEvent("IoTSensorgpsdata","json",data=w,qos=0,on_publish=myOnPublishCall
back)
```

```
    print(d)
```

```
    print("\n")
```

```
    if not success:        print("Not
connected to IoTTF")
```

```
print("\n")    time.sleep(3)
```

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

Connecting IBM Watson and python Code:

The screenshot displays two windows from a Windows operating system. The left window, titled "Python 3.7.4 Shell*", contains Python code for a program that checks a child's location relative to a geofence and reports their temperature. The code uses a loop to repeatedly check the status and print it. The right window shows a JSON object definition for authentication, specifically defining the "method" as "authMethod" and "auth-token" as "authToken".

```

ibmonlinenodered.py - C:\Users\kutta\Desktop\IBM-D\ibmonlinenodered.py (3.7.4)
File Edit Shell Debug Options Window Help
Python 3.7.4 Shell*
check wheather your child is Inside the geofence or Outside geofence
#Pr
org
dev:
dev: {'your_child_zone': 'Outside the geofence'}
aut:
aut: {'temp_status': 'High temperature'}
#ap: Published Temperature = 43 C latitude = 12.130 longitude = 78.198 to IBM Watson
#ap:

try: check wheather your child is Inside the geofence or Outside geofence

except: {'your_child_zone': 'Outside the geofence'}

{'temp_status': 'High temperature'}
Published Temperature = 39 C latitude = 12.131 longitude = 78.195 to IBM Watson

# Co
prin
prin
time
dev:
prin
prin
prin
name
While Published Temperature = 36 C latitude = 12.130 longitude = 78.197 to IBM Watson

check wheather your child is Inside the geofence or Outside geofence

{'your_child_zone': 'Inside the geofence'}

{'temp_status': 'High temperature'}

-method": authMethod, "auth-token": authToken}

type "greeting" 10 times
Ln: 6 Col: 22

```

The screenshot shows the AWS IoT console interface. At the top, there are tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. On the right, there is an 'Add Device' button. The main content area displays details for a device named 'ABCD', which is 'Connected'. The 'Recent Events' tab is selected, showing a list of events. The events are displayed in a table with columns: Event, Value, Format, and Last Received. The events show a stream of data from an IoT sensor, including temperature and geofence status. A simulation is running, as indicated by the text '1 Simulation running'.

| Event | Value | Format | Last Received |
|----------------|---|--------|-------------------|
| IoTSensorgp... | {"temp_status":"High temperature"} | json | a few seconds ago |
| IoTSensorgp... | {"your_child_zone":"Outside the geofence"} | json | a few seconds ago |
| IoTSensorgp... | {"temp":50,"lat":12.132819998043411,"lon":78... | json | a few seconds ago |
| IoTSensorgp... | {"temp_status":"Low temperature"} | | |
| IoTSensorgp... | {"your_child_zone":"Outside the geofence"} | | |

1 Simulation running