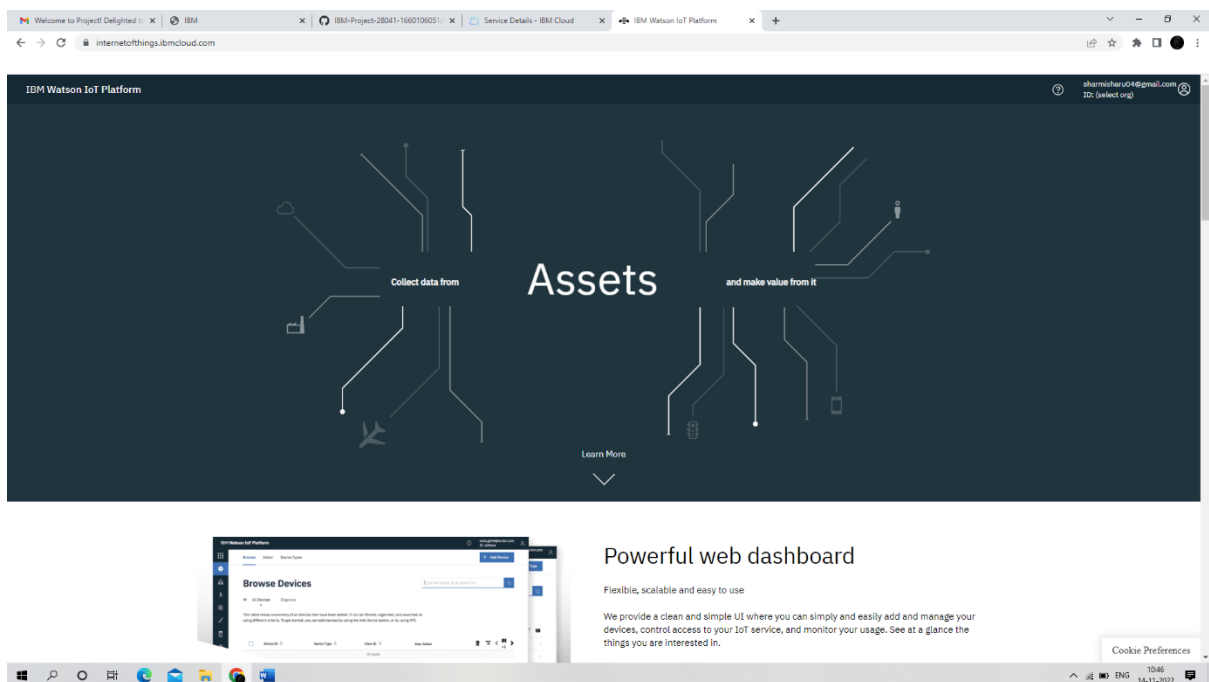


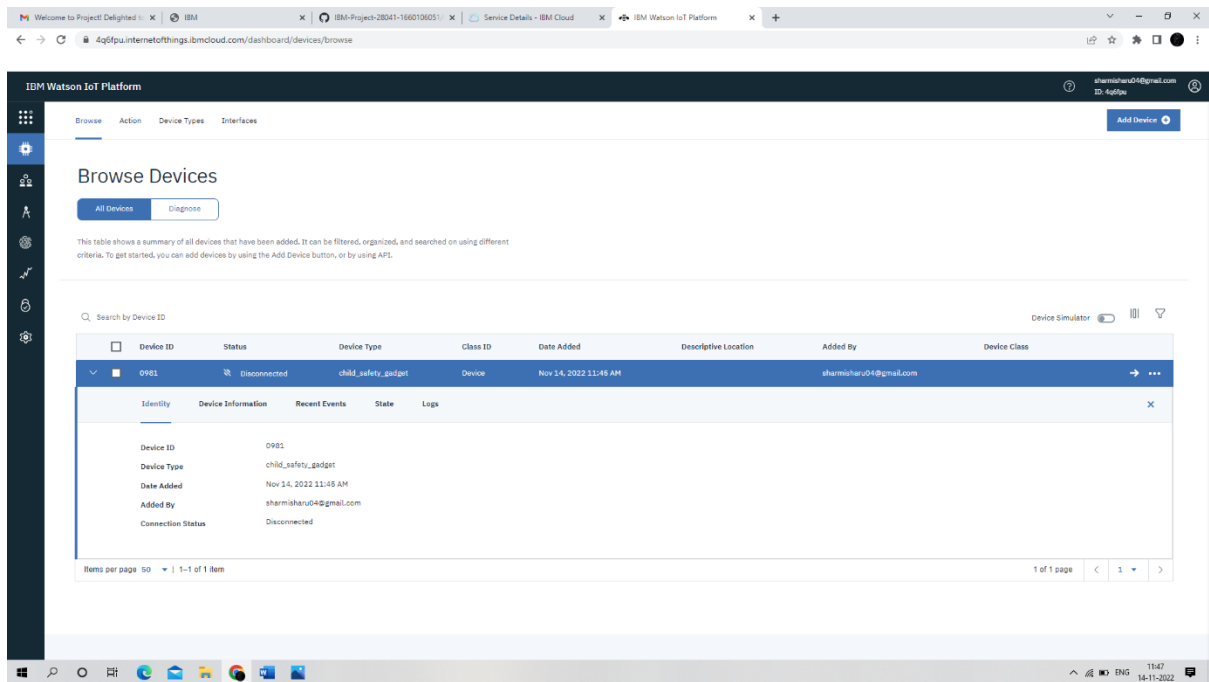
PROJECT DEVELOPMENT PHASE

Delivery of Sprint 2

Team ID	PNT2022TMID14424
Project Name	IoT Based Safety Gadget for Child Safety Monitoring & Notification

Creating IBM Cloud Service and IBM Watson IoT Platform :





Creating and Connecting IBM cloud for Project and Python Code :

```
import time
```

```
import sys
```

```
import ibmiotf.application
```

```
import ibmiotf.device
```

```
import random
```

```
#Provide your IBM Watson Device Credentials organization = "0pycss"
```

```
deviceType = "weather_Device1"
```

```
deviceId = "weather_deviceid" authMethod = "token"
```

```
authToken = "(j!jK*nvh9OKQD9!dJ"
```

```
#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
print("POWER ON ") print("CHECKING CONNECTION TO IBM WATSON...")
time.sleep(2) deviceCli.connect() print("dear user ... welcome to IBM-IOT ")
print("You can know your child's live location and temperature ")
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=myOnPublishCallback)

```

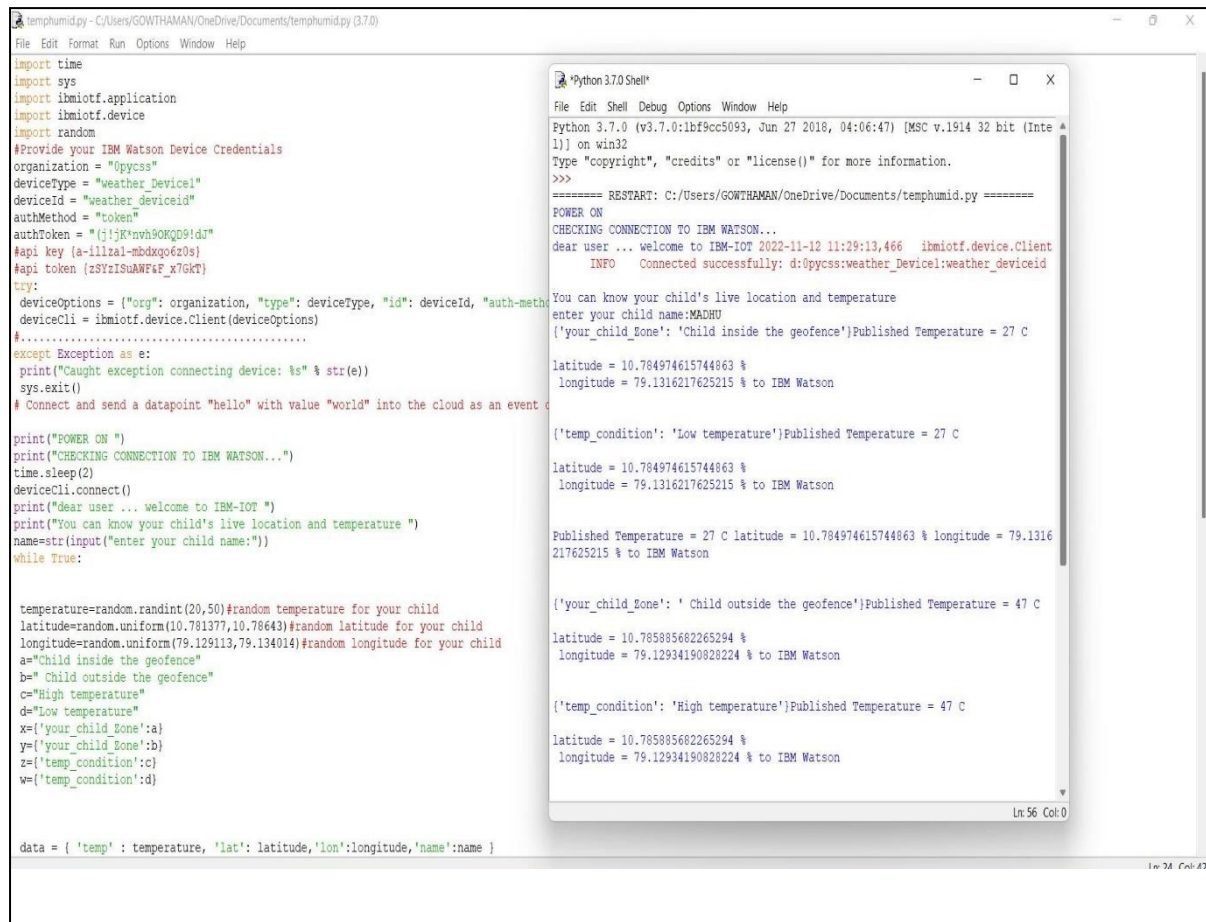
```

print(x) print("\n") else:
deviceCli.publishEvent("IoTSensorgpsdata","json",data=y,qos=0,on_publish=myOnPublishCallback)
print(y)
print("\n")
if (temperature>35):
deviceCli.publishEvent("IoTSensorgpsdata","json",data=z,qos=0,on_publish=myOnPublishCallback)
print(z)
print("\n")
else:
deviceCli.publishEvent("IoTSensorgpsdata","json",data=w,qos=0,on_publish=myOnPublishCallback)
print(w)
print("\n")
if not success:
print("Not connected to IoTF")
print("\n")
time.sleep(3)
# Disconnect the device and application from the cloud deviceCli.disconnect()
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=myOnPublishCallback)
print(x)
print("\n")

```

```
else:
    deviceCli.publishEvent("IoTSensorgpsdata","json",data=y,qos=0,on_publish=myOnPublishCallback)
    print(y)
    print("\n")
    if (temperature>35):
        deviceCli.publishEvent("IoTSensorgpsdata","json",data=z,qos=0,on_publish=myOnPublishCallback)
        print(z)
        print("\n")
    else:
        deviceCli.publishEvent("IoTSensorgpsdata","json",data=w,qos=0,on_publish=myOnPublishCallback)
        print(w)
        print("\n")
    if not success:
        print("Not connected to IoT")
        print("\n")
    time.sleep(3)
# Disconnect the device and application from the cloud deviceCli.disconnect()
```

Connecting IBM Watson and python Code :



The image shows a Python script in a text editor and its execution output in a terminal window. The script, named `temphumid.py`, is located at `C:/Users/GOWTHAMAN/OneDrive/Documents/temphumid.py (3.7.0)`. It imports `time`, `sys`, `ibmiotf.application`, `ibmiotf.device`, and `random`. It provides IBM Watson Device Credentials and uses the `ibmiotf.device.Client` to connect to the IBM Watson IoT Platform. The script sends a datapoint "hello" with value "world" into the cloud as an event. It also prints the temperature and location of a child.

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

#Provide your IBM Watson Device Credentials
organization = "0pycss"
deviceType = "weather_Device1"
deviceId = "weather_deviceid"
authMethod = "token"
authToken = "(j)K*rvh9OKQD9!dJ"
#api key (a-illza1-mbdxqo6z0s)
#api token (zSYzISuANFf_x7GkT)

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod}
    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

print("POWER ON ")
print("CHECKING CONNECTION TO IBM WATSON...")
time.sleep(2)
deviceCli.connect()
print("dear user ... welcome to IBM-IOT ")
print("You can know your child's live location and temperature ")
name = 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 }
```

The terminal output shows the following messages:

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/GOWTHAMAN/OneDrive/Documents/temphumid.py =====
POWER ON
CHECKING CONNECTION TO IBM WATSON...
dear user ... welcome to IBM-IOT 2022-11-12 11:29:13,466 ibmiotf.device.Client
INFO Connected successfully: d:0pycss:weather_Device1:weather_deviceid

You can know your child's live location and temperature
enter your child name: MADHU
('your_child_zone': 'Child inside the geofence') Published Temperature = 27 C

latitude = 10.784974615744863 %
longitude = 79.1316217625215 % to IBM Watson

('temp_condition': 'Low temperature') Published Temperature = 27 C

latitude = 10.784974615744863 %
longitude = 79.1316217625215 % to IBM Watson

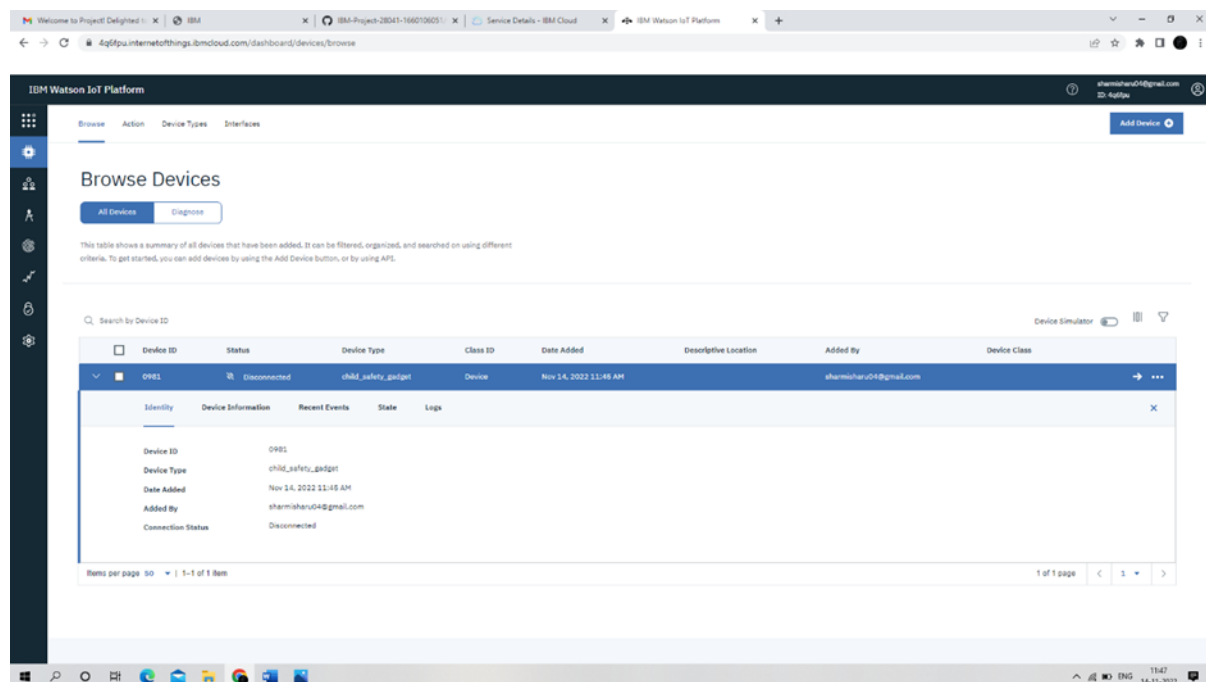
Published Temperature = 27 C latitude = 10.784974615744863 % longitude = 79.1316217625215 % to IBM Watson

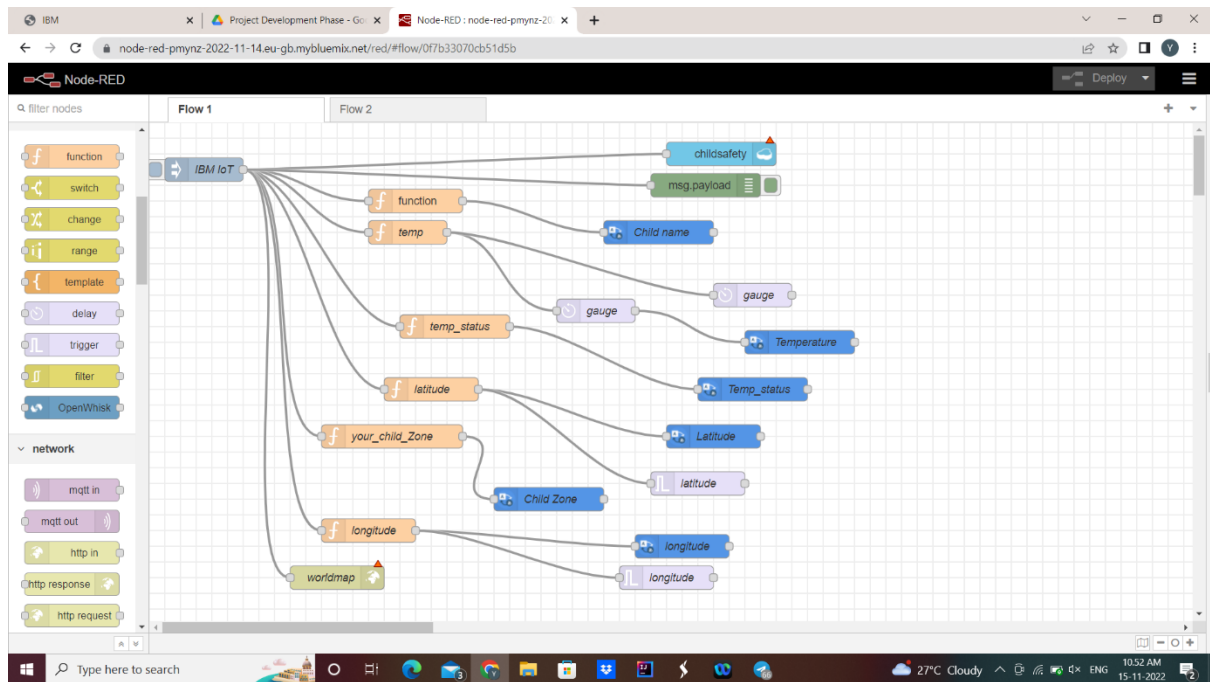
('your_child_zone': 'Child outside the geofence') Published Temperature = 47 C

latitude = 10.785885682265294 %
longitude = 79.12934190828224 % to IBM Watson

('temp_condition': 'High temperature') Published Temperature = 47 C

latitude = 10.785885682265294 %
longitude = 79.12934190828224 % to IBM Watson
```





Node-RED Output:

