

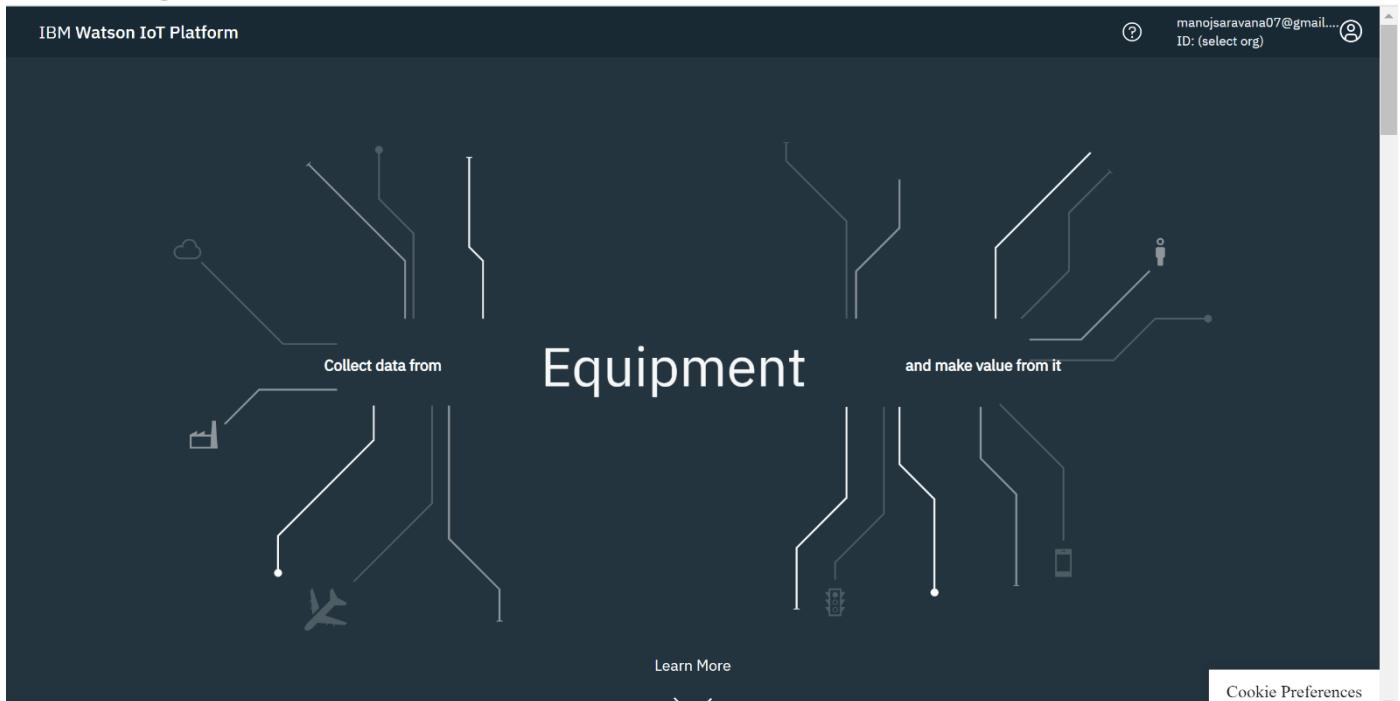
PANIMALAR ENGINEERING COLLEGE

IBM NALAIYATHIRAN

DELIVERY OF SPRINT 2

TEAM ID	PNT2022TMID01082
PROJECT NAME	IOT based safety gadget for child safety monitoring and notification
TEAM MEMBERS	MUKKESH B PREM KUMAR A RAGHUL S MANOJ KUMAR A S

Creating IBM Cloud Service and IBM WATSON IoT PLATFORM:



Creating and Connecting IBM cloud for Project and Python Code:

The screenshot shows the IBM Watson IoT Platform Device Catalog. On the left is a sidebar with various icons. At the top, there's a header bar with 'IBM Watson IoT Platform' and user information ('manojsaravana07@gmail.com', 'ID: i9i3ak'). Below the header is a navigation bar with 'Browse', 'Action', 'Device Types', 'Interfaces', and a blue 'Add Device' button. A search bar says 'Search by Device ID'. To the right of the search bar is a 'Device Simulator' toggle switch, which is off. The main area displays a table of devices. One row is selected, showing details for a device with ID '12345'. The device is 'Disconnected', belongs to 'NodePRMM' class, and was added on '12 Nov 2022 20:40'. The table has columns: Device ID, Status, Device Type, Class ID, Date Added, and three more columns for Identity, Device Information, Recent Events, State, and Logs. At the bottom, there are pagination controls: 'Items per page: 50', '1 of 1 page', and page numbers '1'.

	Device ID	Status	Device Type	Class ID	Date Added	
<input type="checkbox"/>	12345	Disconnected	NodePRMM	Device	12 Nov 2022 20:40	→ ...
Identity Device Information Recent Events State Logs X						
Device ID: 12345 Device Type: NodePRMM Date Added: 12 Nov 2022 20:40 Added By: manojsaravana07@gmail.com Connection Status: Disconnected						

```
import time import sys

import ibmiotf.application

import ibmiotf.device import

random

#Provide your IBM Watson Device Credentials

organization = "Opycss" deviceType =

"weather_Device1" deviceld = "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": deviceld, "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=myOnPublis
hCallback)

        print(x)

    print("\n") else:

deviceCli.publishEvent("IoTSensorgpsdata", "json", data=y, qos=0, on_publish=myOnPublis
hCallback)

    print(y) print("\n")

if (temperature>35):

deviceCli.publishEvent("IoTSensorgpsdata", "json", data=z, qos=0, on_publish=myOnPublis
hCallback)

    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()

```

Connecting IBM Watson and python Code:

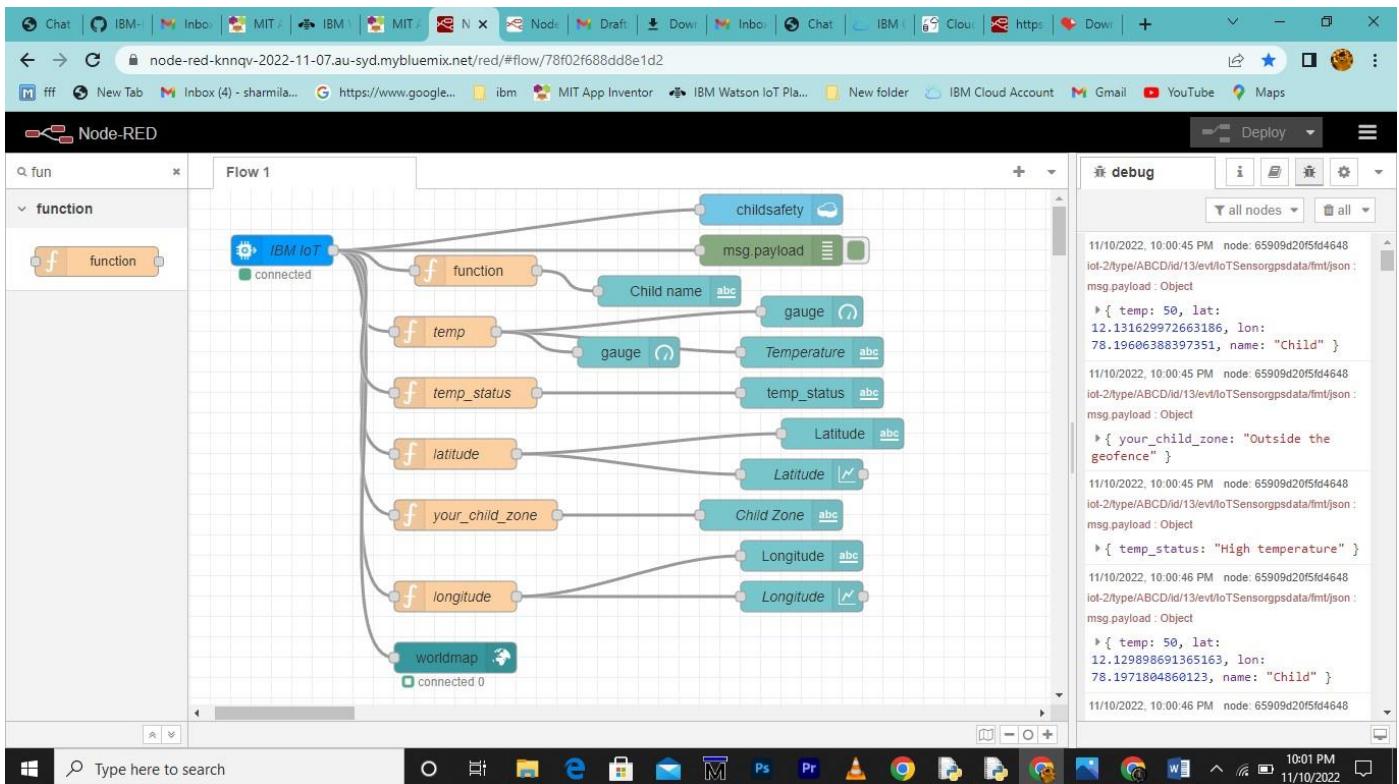
The screenshot shows a Windows desktop environment with two open windows. On the left is a code editor window titled 'Child Safety device.py - C:/Users/kutta/Desktop/IBM-Dr/Child Safety device.py (3.7.4)' containing Python code for connecting to IBM Watson. On the right is a terminal window titled 'Python 3.7.4 Shell' showing the execution of the script and its output.

```

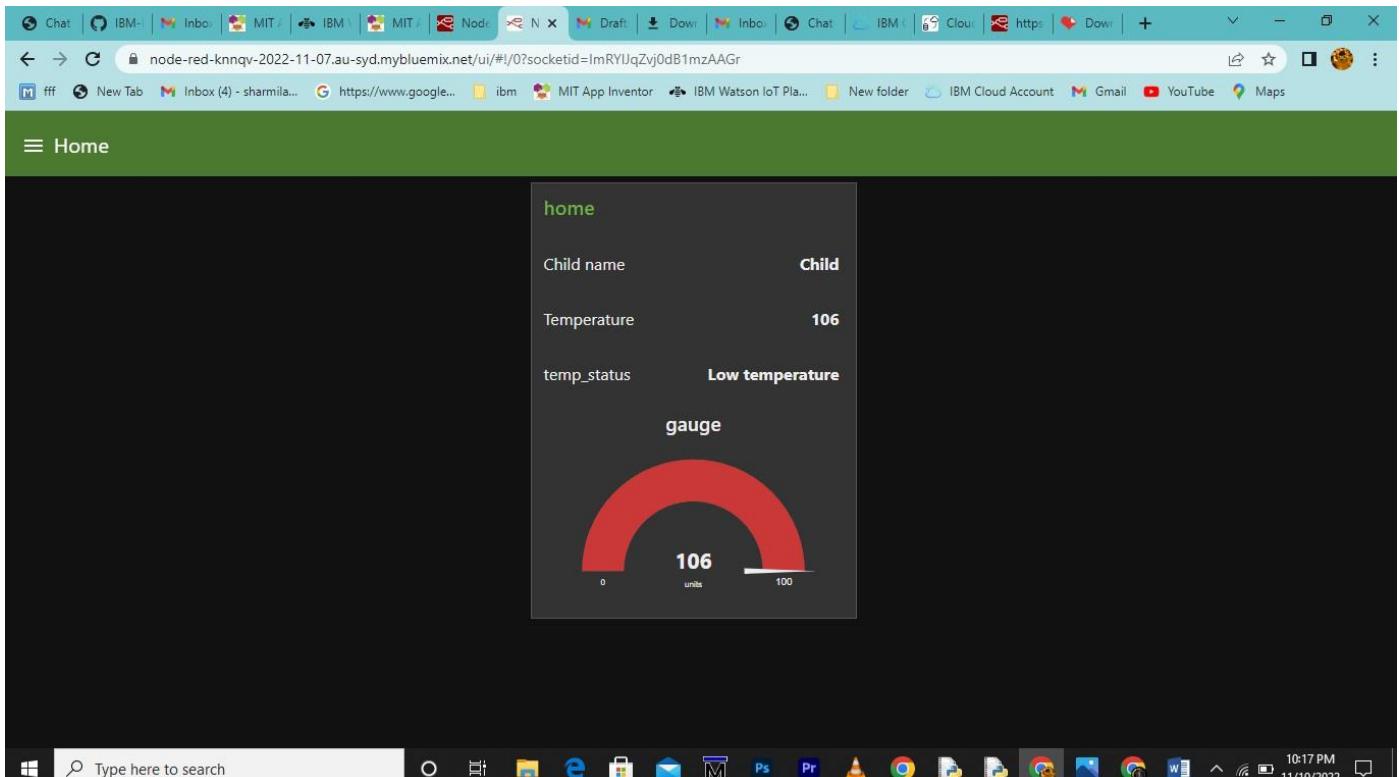
File Edit Format Run Options Window Help
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 ]
(AMD64) on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/kutta/Desktop/IBM-Dr/Child Safety device.py =====
power on
checking connection to waston iot...
2022-11-10 22:14:21,799 ibmiotf.device.Client      INFO  Connected successfully: dillzal@latlontem:613510
dear user ... welcome to IBM-IOT
i can provide your children live location and temperature

enter your child name:child
Published Temperature = 39 C latitude = 10.782749628132827 % longitude = 79.
867253162 % to IBM Watson
Published Temperature = 39 C latitude = 10.782669248109656 % longitude = 79.
1255540076 % to IBM Watson
Published Temperature = 43 C latitude = 10.781765104656792 % longitude = 79.
077864707 % to IBM Watson
Published Temperature = 30 C latitude = 10.786083936690018 % longitude = 79.
2366715787 % to IBM Watson
Published Temperature = 31 C latitude = 10.784810558975826 % longitude = 79.
01173559415 % to IBM Watson
Published Temperature = 45 C latitude = 10.785949922923024 % longitude = 79.
5563867668 % to IBM Watson
Published Temperature = 24 C latitude = 10.784168891438233 % longitude = 79.
9528906442 % to IBM Watson
Published Temperature = 23 C latitude = 10.786248060983958 % longitude = 79.
4368596464 % to IBM Watson
Published Temperature = 27 C latitude = 10.783808327214418 % longitude = 79.
951933729 % to IBM Watson
Published Temperature = 43 C latitude = 10.786340416981865 % longitude = 79.
7748803969 % to IBM Watson
Published Temperature = 49 C latitude = 10.786208956579015 % longitude = 79.
2192551409 % to IBM Watson
Published Temperature = 45 C latitude = 10.783690544907325 % longitude = 79.
504415061 % to IBM Watson

```



NODE-RED OUPUT:



≡ map

