SMART FARMER – IOT ENABLEDD SMART FARMING APPLICATION

DEVELOP A PYTHON CODE

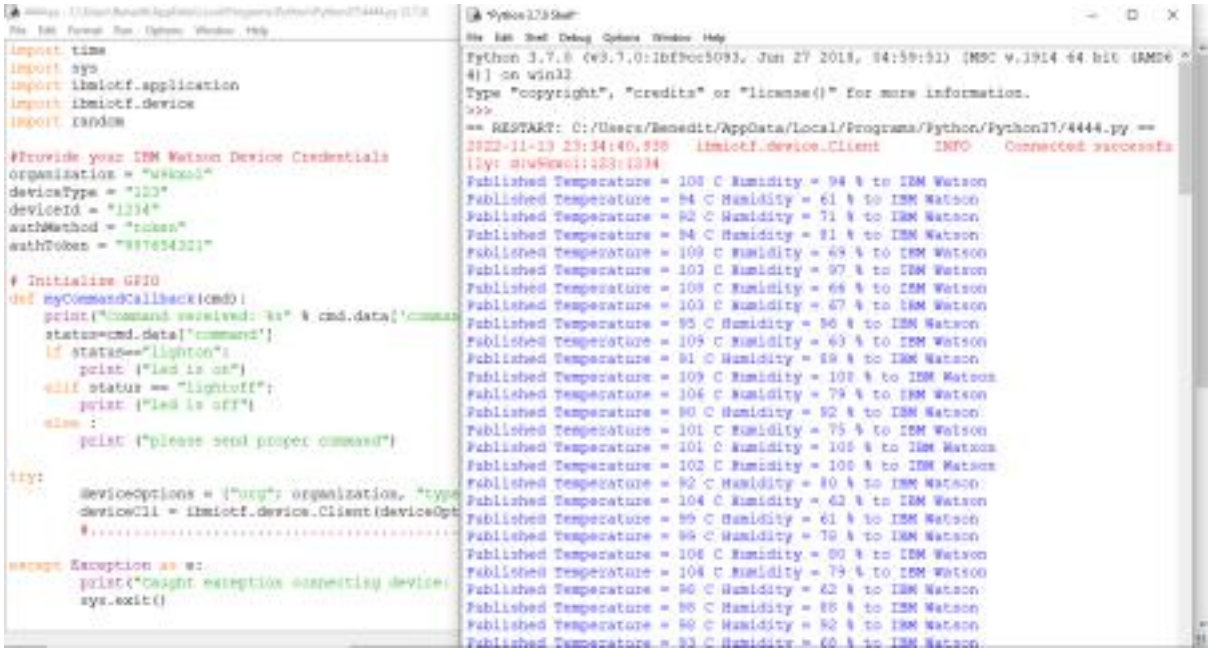
DATE	17 NOVEMBER 2022
TITLE	SMART FARMER – IOT ENABLEDD SMART FARMING APPLICATION
TEAM ID	PNT2022TMID33748
TEAM LEADER NAME	SUBIKA M
TEAM MEMBER NAME	PEMALATHA S SELENA CLARA M SNEHA L

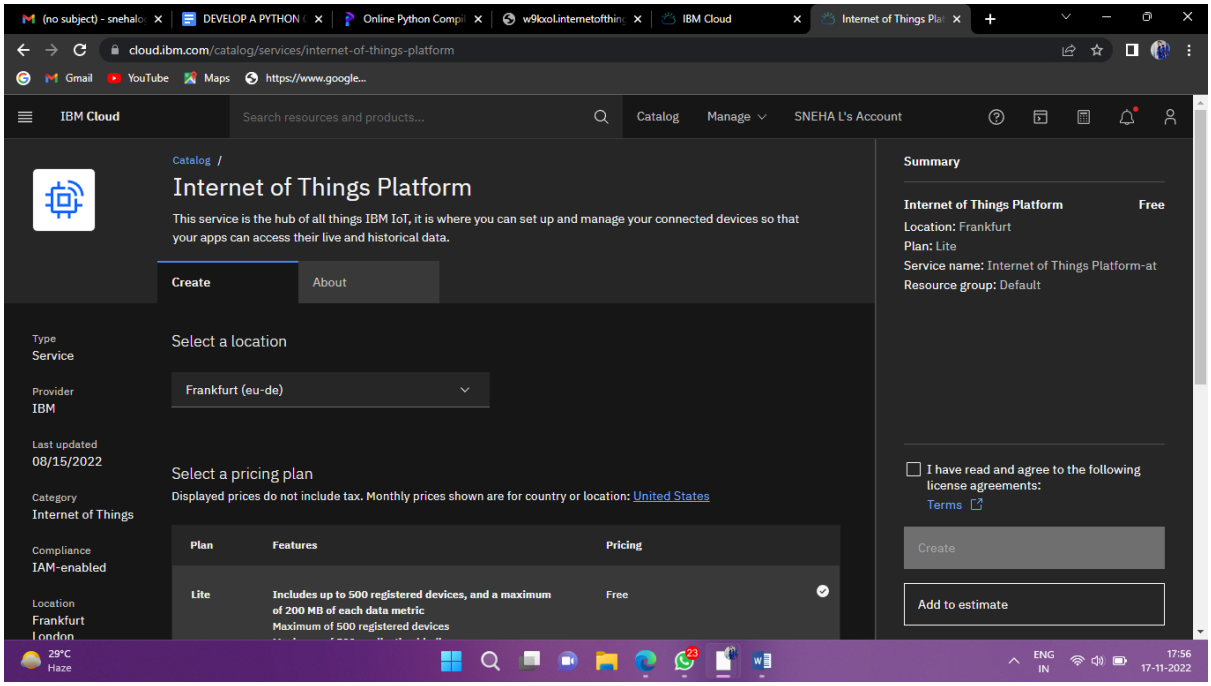
PYTHON CODE:

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

#Provide your IBM Watson Device Credentials
organization = "w9kxol"
deviceType = "123"
deviceId = "1234"
authMethod = "token"
authToken = "8925435346"

# Initialize GPIO
def myCommandCallback(cmd):
 print("Command received: %s" % cmd.data['command'])
 status=cmd.data['command']
 if status=="lighton":
 print ("led is on")
 elif status == "lightoff":
 print ("led is off")
```

```
 else :
 print ("please send proper command")

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(90,110)
 Humid=random.randint(60,100)

 data = { 'temp' : temp, 'Humid': Humid }
 #print data
 def myOnPublishCallback():
 print ("Published Temperature = %s C" % temp, "Humidity = %s %%" %
Humid, "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()
```

OUTPUT:


import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "w9kxol"
deviceType = "123"
deviceId = "1234"
authMethod = "token"
authToken = "9876543211"
# Initialize GPIO
def myCommandCallback(cmd):
print("Command received: %s" % cmd.data['command'])
status=cmd.data['command']
if status=="lighton":
print ("Led is on")
elif status == "lightoff":
print ("Led is off")
else :
print ("please send proper command")
try:
deviceOptions = {"org": organization, "type
deviceCli = ibmiotf.device.Client(deviceOpt
except Exception as e:
print("Caught exception connecting device:
sys.exit()
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/Benedit/AppData/Local/Programs/Python/Python37/4444.py ==
2022-11-13 23:34:40,930 ibmiotf.device.Client INFO Connected successfu
lly: d:w9kxol:123:1234
Published Temperature = 108 C Humidity = 94 % to IBM Watson
Published Temperature = 84 C Humidity = 61 % to IBM Watson
Published Temperature = 82 C Humidity = 71 % to IBM Watson
Published Temperature = 84 C Humidity = 81 % to IBM Watson
Published Temperature = 109 C Humidity = 69 % to IBM Watson
Published Temperature = 103 C Humidity = 97 % to IBM Watson
Published Temperature = 108 C Humidity = 66 % to IBM Watson
Published Temperature = 103 C Humidity = 67 % to IBM Watson
Published Temperature = 95 C Humidity = 96 % to IBM Watson
Published Temperature = 109 C Humidity = 63 % to IBM Watson
Published Temperature = 81 C Humidity = 89 % to IBM Watson
Published Temperature = 109 C Humidity = 102 % to IBM Watson
Published Temperature = 106 C Humidity = 79 % to IBM Watson
Published Temperature = 90 C Humidity = 92 % to IBM Watson
Published Temperature = 101 C Humidity = 75 % to IBM Watson
Published Temperature = 101 C Humidity = 105 % to IBM Watson
Published Temperature = 102 C Humidity = 100 % to IBM Watson
Published Temperature = 92 C Humidity = 80 % to IBM Watson
Published Temperature = 104 C Humidity = 62 % to IBM Watson
Published Temperature = 99 C Humidity = 61 % to IBM Watson
Published Temperature = 86 C Humidity = 70 % to IBM Watson
Published Temperature = 104 C Humidity = 80 % to IBM Watson
Published Temperature = 104 C Humidity = 79 % to IBM Watson
Published Temperature = 80 C Humidity = 62 % to IBM Watson
Published Temperature = 90 C Humidity = 85 % to IBM Watson
Published Temperature = 80 C Humidity = 92 % to IBM Watson
Published Temperature = 93 C Humidity = 60 % to IBM Watson


(no subject) - snehalo
DEVELOP A PYTHON
Online Python Compi
w9kxol.internetofthin
IBM Cloud
Internet of Things Pla
cloud.ibm.com/catalog/services/internet-of-things-platform
Gmail
YouTube
Maps
https://www.google...
IBM Cloud
Search resources and products...
Catalog
Manage
SNEHA L's Account
Catalog /
Internet of Things Platform
This service is the hub of all things IBM IoT, it is where you can set up and manage your connected devices so that your apps can access their live and historical data.
Create
About
Type
Service
Provider
IBM
Last updated
08/15/2022
Category
Internet of Things
Compliance
IAM-enabled
Location
Frankfurt
London
Select a location
Frankfurt (eu-de)
Select a pricing plan
Displayed prices do not include tax. Monthly prices shown are for country or location: United States
Plan
Features
Pricing
Lite
Includes up to 500 registered devices, and a maximum of 200 MB of each data metric
Maximum of 500 registered devices
Free
Summary
Internet of Things Platform
Free
Location: Frankfurt
Plan: Lite
Service name: Internet of Things Platform-at
Resource group: Default
I have read and agree to the following license agreements:
Terms
Create
Add to estimate
29°C
Haze
ENG
IN
17:56
17-11-2022