**Project Development Phase**

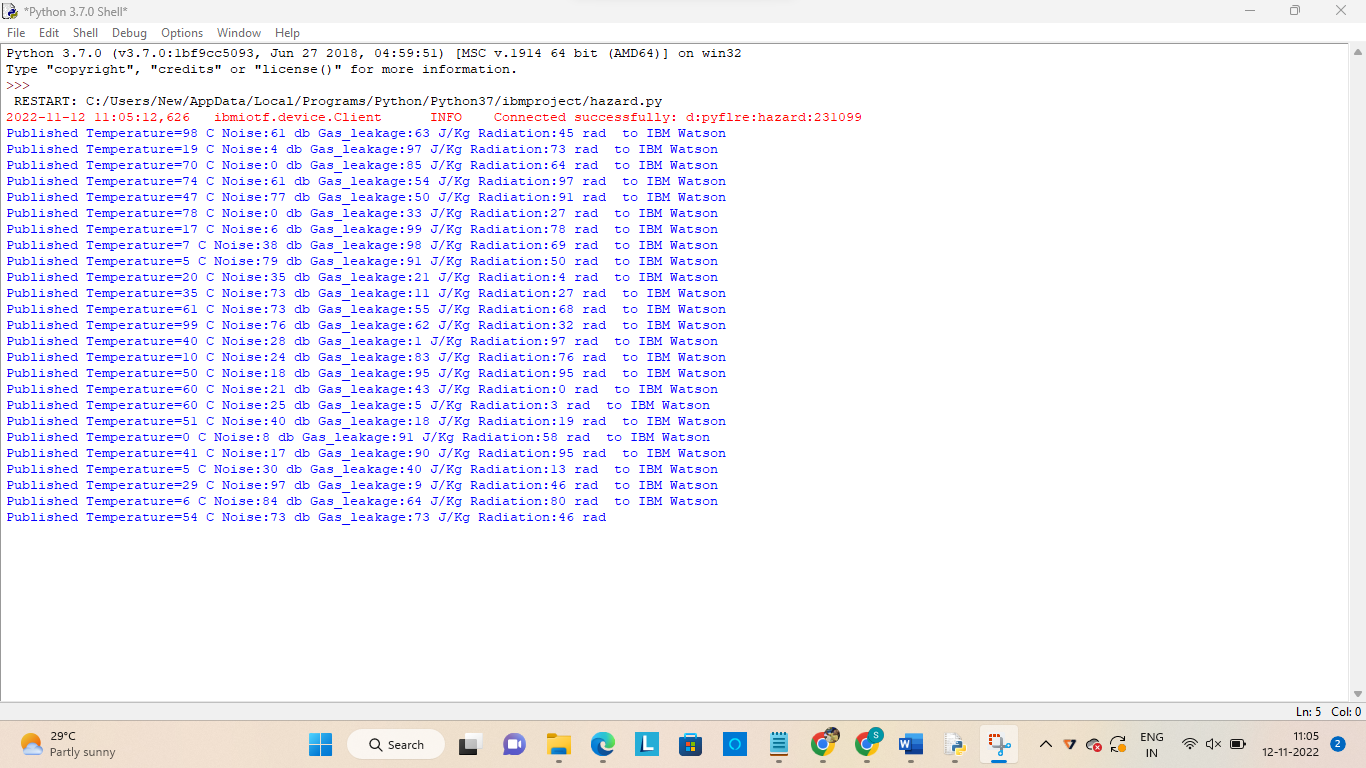
**Sprint – 3**

**Program:**

import time  
import sys  
import ibmiotf.application  
import ibmiotf.device  
import random  
  
organization="pyflre"  
deviceType="hazard"  
deviceId="231099"  
authMethod="token"  
authToken="zHP+8fjUb\*HmxvADd8"  
  
def myCommandCallback(cmd):  
print("Command received:%s" % cmd.data['command'])  
status=cmd.data['command']  
if status=="motoron":  
print("Motor is ON")  
else:  
print("Motor is OFF")  
  
  
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()  
  
deviceCli.connect()

while True:  
  
temp=random.randint(0,100)  
noise=random.randint(0,100)  
Gas=random.randint(0,100)  
radn=random.randint(0,100)  
data={'Temperature' :temp,'Noise':noise,'Gas\_leakage':Gas,'Radiation':radn}  
  
  
def myOnPublishCallback():  
print("Published Temperature=%s C" %temp,"Noise:%s db" %noise,"Gas\_leakage:%s J/Kg" %Gas,"Radiation:%s rad "%radn,"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  
  
deviceCli.disconnect()

**Output:**

****