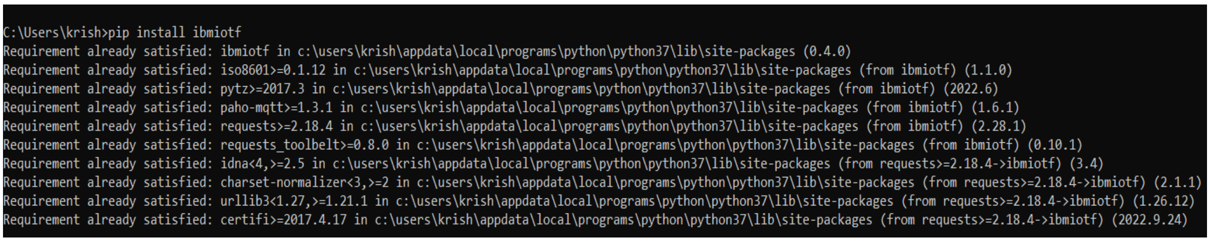
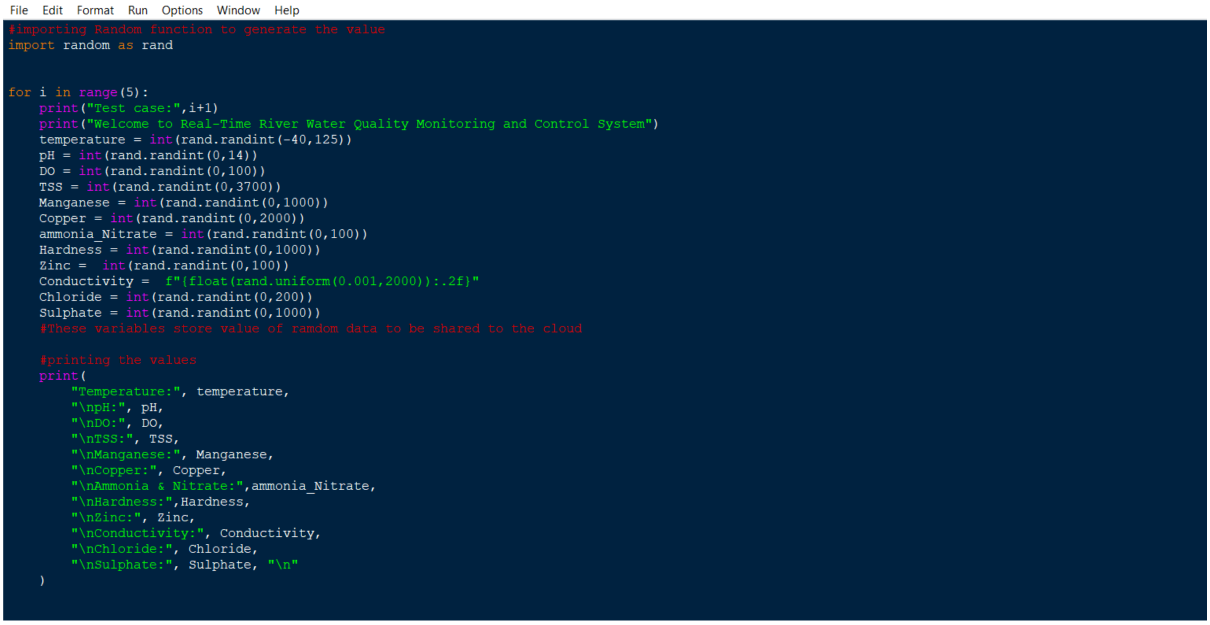
Develop the Python Script

Step 1: Install all the required modules like Random and ibmiotf for the python script



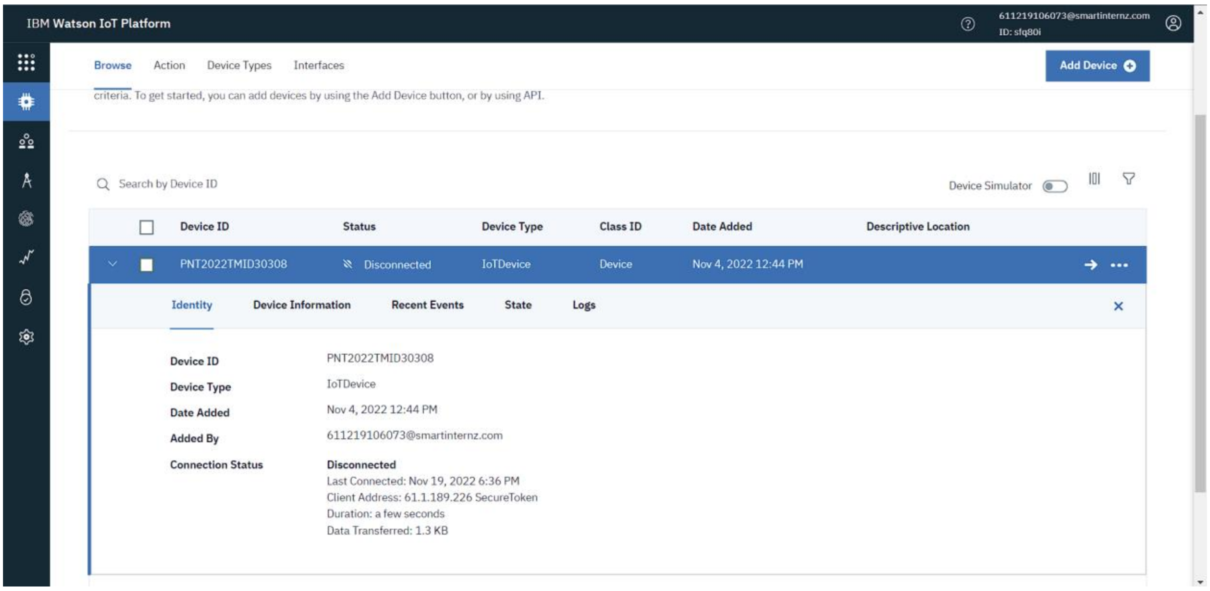
```
C:\Users\krish>pip install ibmiotf
Requirement already satisfied: ibmiotf in c:\users\krish\appdata\local\programs\python\python37\lib\site-packages (0.4.0)
Requirement already satisfied: iso8601>=0.1.12 in c:\users\krish\appdata\local\programs\python\python37\lib\site-packages (from ibmiotf) (1.1.0)
Requirement already satisfied: pytz>=2017.3 in c:\users\krish\appdata\local\programs\python\python37\lib\site-packages (from ibmiotf) (2022.6)
Requirement already satisfied: paho-mqtt>=1.3.1 in c:\users\krish\appdata\local\programs\python\python37\lib\site-packages (from ibmiotf) (1.6.1)
Requirement already satisfied: requests>=2.18.4 in c:\users\krish\appdata\local\programs\python\python37\lib\site-packages (from ibmiotf) (2.28.1)
Requirement already satisfied: requests_toolbelt>=0.8.0 in c:\users\krish\appdata\local\programs\python\python37\lib\site-packages (from ibmiotf) (0.10.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\krish\appdata\local\programs\python\python37\lib\site-packages (from requests>=2.18.4->ibmiotf) (3.4)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\krish\appdata\local\programs\python\python37\lib\site-packages (from requests>=2.18.4->ibmiotf) (2.1.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\krish\appdata\local\programs\python\python37\lib\site-packages (from requests>=2.18.4->ibmiotf) (1.26.12)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\krish\appdata\local\programs\python\python37\lib\site-packages (from requests>=2.18.4->ibmiotf) (2022.9.24)
```

Step 2: Import the required modules and then generate all the required data using random function



```python
#importing Random function to generate the value
import random as rand


for i in range(5):
    print("Test case:",i+1)
    print("Welcome to Real-Time River Water Quality Monitoring and Control System")
    temperature = int(rand.randint(-40,125))
    pH = int(rand.randint(0,14))
    DO = int(rand.randint(0,100))
    TSS = int(rand.randint(0,3700))
    Manganese = int(rand.randint(0,1000))
    Copper = int(rand.randint(0,2000))
    ammonia_Nitrate = int(rand.randint(0,100))
    Hardness = int(rand.randint(0,1000))
    Zinc =  int(rand.randint(0,100))
    Conductivity =  f"{float(rand.uniform(0.001,2000)):.2f}"
    Chloride = int(rand.randint(0,200))
    Sulphate = int(rand.randint(0,1000))
    #These variables store value of ramdom data to be shared to the cloud

    #printing the values
    print(
        "Temperature:", temperature,
        "\npH:", pH,
        "\nDO:", DO,
        "\nTSS:", TSS,
        "\nManganese:", Manganese,
        "\nCopper:", Copper,
        "\nAmmonia & Nitrate:",ammonia_Nitrate,
        "\nHardness:",Hardness,
        "\nZinc:", Zinc,
        "\nConductivity:", Conductivity,
        "\nChloride:", Chloride,
        "\nSulphate:", Sulphate, "\n"
    )
```

Step 3: Make the connection with the cloud platform using the device created in the IBM Watson Platform



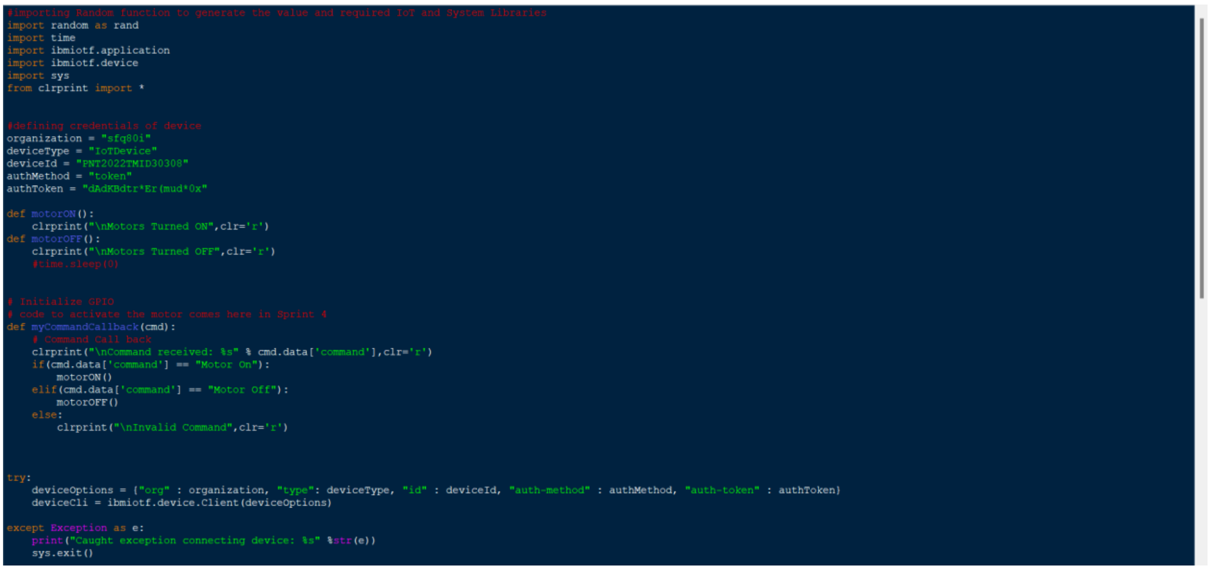
criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

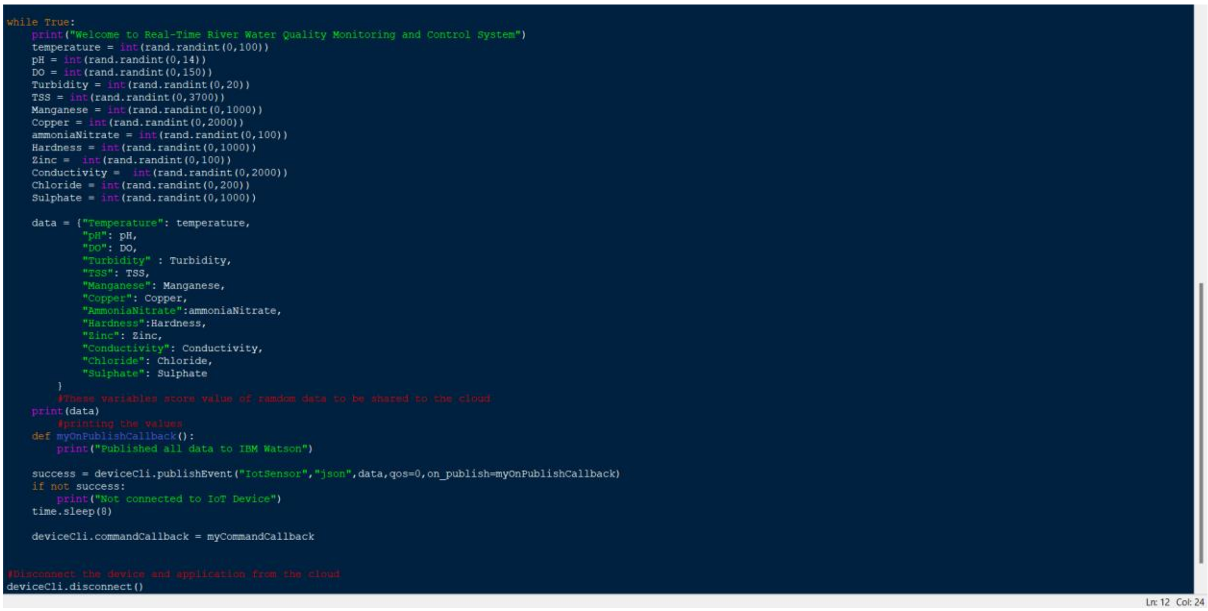
☐	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
☐	PNT2022TMID30308	Disconnected	IoTDevice	Device	Nov 4, 2022 12:44 PM	

Device ID	PNT2022TMID30308
Device Type	IoTDevice
Date Added	Nov 4, 2022 12:44 PM
Added By	611219106073@smartinternz.com
Connection Status	Disconnected Last Connected: Nov 19, 2022 6:36 PM Client Address: 61.1.189.226 SecureToken Duration: a few seconds Data Transferred: 1.3 KB

Step 4: Configure the code so that it is connected to the device

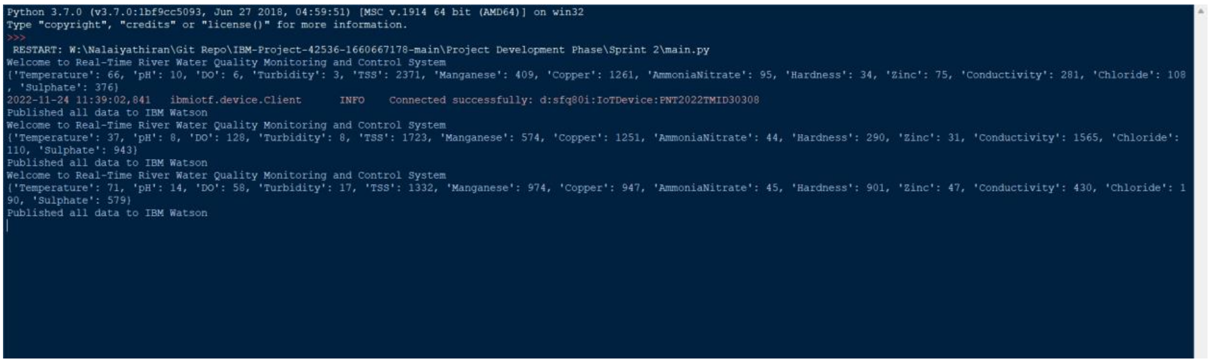


```python
#importing Random function to generate the value and required IoT and System Libraries
import random as rand
import time
import ibmiotf.application
import ibmiotf.device
import sys
from clrprint import *


#defining credentials of device
organization = "sfq80i"
deviceType = "IoTDevice"
deviceId = "PNT2022TMID30308"
authMethod = "token"
authToken = "dAdKBdtr*Er(mud*0x"

def motorON():
    clrprint("\nMotors Turned ON",clr='r')
def motorOFF():
    clrprint("\nMotors Turned OFF",clr='r')
    #time.sleep(0)


# Initialize GPIO
# code to activate the motor comes here in Sprint 4
def myCommandCallback(cmd):
    # Command Call back
    clrprint("\nCommand received: %s" % cmd.data['command'],clr='r')
    if(cmd.data['command'] == "Motor On"):
        motorON()
    elif(cmd.data['command'] == "Motor Off"):
        motorOFF()
    else:
        clrprint("\nInvalid Command",clr='r')



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()
```

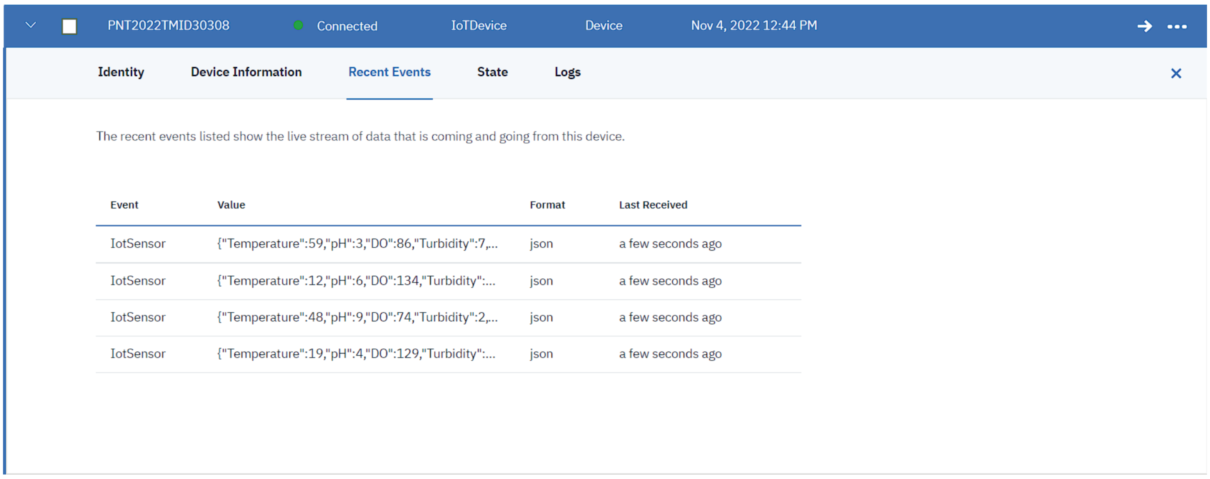


```
while True:
    print("Welcome to Real-Time River Water Quality Monitoring and Control System")
    temperature = int(rand.randint(0,100))
    pH = int(rand.randint(0,14))
    DO = int(rand.randint(0,150))
    Turbidity = int(rand.randint(0,20))
    TSS = int(rand.randint(0,3700))
    Manganese = int(rand.randint(0,1000))
    Copper = int(rand.randint(0,2000))
    ammoniaNitrate = int(rand.randint(0,100))
    Hardness = int(rand.randint(0,1000))
    Zinc =  int(rand.randint(0,100))
    Conductivity =  int(rand.randint(0,2000))
    Chloride = int(rand.randint(0,200))
    Sulphate = int(rand.randint(0,1000))

    data = {"Temperature": temperature,
            "pH": pH,
            "DO": DO,
            "Turbidity" : Turbidity,
            "TSS": TSS,
            "Manganese": Manganese,
            "Copper": Copper,
            "AmmoniaNitrate":ammoniaNitrate,
            "Hardness":Hardness,
            "Zinc": Zinc,
            "Conductivity": Conductivity,
            "Chloride": Chloride,
            "Sulphate": Sulphate
        }
        #These variables store value of ramdom data to be shared to the cloud
    print(data)
        #printing the values
    def myOnPublishCallback():
        print("Published all data to IBM Watson")

    success = deviceCli.publishEvent("IotSensor","json",data,qos=0,on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoT Device")
    time.sleep(8)

    deviceCli.commandCallback = myCommandCallback


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

Step 5: Execute the Python code to transfer the data to the device in cloud



```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
 RESTART: W:\Nalaiyathiran\Git Repo\IBM-Project-42536-1660667178-main\Project Development Phase\Sprint 2\main.py
Welcome to Real-Time River Water Quality Monitoring and Control System
{'Temperature': 66, 'pH': 10, 'DO': 6, 'Turbidity': 3, 'TSS': 2371, 'Manganese': 409, 'Copper': 1261, 'AmmoniaNitrate': 95, 'Hardness': 34, 'Zinc': 75, 'Conductivity': 281, 'Chloride': 108
, 'Sulphate': 376}
2022-11-24 11:39:02,841   ibmiotf.device.Client      INFO    Connected successfully: d:sfq80i:IoTDevice:PNT2022TMID30308
Published all data to IBM Watson
Welcome to Real-Time River Water Quality Monitoring and Control System
{'Temperature': 37, 'pH': 8, 'DO': 128, 'Turbidity': 8, 'TSS': 1723, 'Manganese': 574, 'Copper': 1251, 'AmmoniaNitrate': 44, 'Hardness': 290, 'Zinc': 31, 'Conductivity': 1565, 'Chloride':
110, 'Sulphate': 943}
Published all data to IBM Watson
Welcome to Real-Time River Water Quality Monitoring and Control System
{'Temperature': 71, 'pH': 14, 'DO': 58, 'Turbidity': 17, 'TSS': 1332, 'Manganese': 974, 'Copper': 947, 'AmmoniaNitrate': 45, 'Hardness': 901, 'Zinc': 47, 'Conductivity': 430, 'Chloride': 1
90, 'Sulphate': 579}
Published all data to IBM Watson
```



PNT2022TMID30308 ● Connected IoTDevice Device Nov 4, 2022 12:44 PM

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
IotSensor	{"Temperature":59,"pH":3,"DO":86,"Turbidity":7,...	json	a few seconds ago
IotSensor	{"Temperature":12,"pH":6,"DO":134,"Turbidity":...	json	a few seconds ago
IotSensor	{"Temperature":48,"pH":9,"DO":74,"Turbidity":2,...	json	a few seconds ago
IotSensor	{"Temperature":19,"pH":4,"DO":129,"Turbidity":...	json	a few seconds ago