Project Development Phase

Sprint-2

Date	16 November 2022
Team ID	PNT2022TMID19640
Project Name	Project - Real time River water quality monitoring and control system

Installed necessary libraries in Python:

```
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys
from twilio.rest import Client
account_sid = 'AC18b4d7a136b9a07a181a837c23ad1358'

auth_token ='adc9782f6520041c84ac4930daad0625 '
client = Client(account_sid, auth_token)

organization = "wbp1fk"
deviceType = "ESP32"
deviceId = "sensor_data_1"
authMethod = "token"
authToken = "prototype_1"

pH = random.randint(1, 14)
turbidity = random.randint(1, 1000)
temperature = random.randint(0, 100)
info=""

def myCommandCallback(cmd):
    print("Command Received: %s" % cmd.data['command'])
```

Python program to connect IBM Watson Platform is shown below:

import ibmiotf.application

import ibmiotf.device

import time

import random

import sys

organization = "wbp1fk"

deviceType = "ESP32"

deviceId = "sensor_data_1"

authMethod = "token"

```
authToken = "prototype_1"


pH = random.randint(1, 14)
turbidity = random.randint(1, 1000)
temperature = random.randint(0, 100)
info=""
def myCommandCallback(cmd):
   print("Command Received: %s" % cmd.data['command'])
   print(cmd)
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:
   pH = random.randint(1, 14)
   turbidity = random.randint(1, 1000)
   temperature = random.randint(0, 100)
   if temperature>70 or pH<6 or pH>8 or turbidity>500:
      print("high")
      info="harmful to drink"
 else:
      info="capable to drinking"
   data = {'pH': pH, 'turbid': turbidity,'temp': temperature,'info':info}
   def myOnPublishCallback():
```

```
        print("Published pH= %s" % pH, "Turbidity:%s" % turbidity,
              "Temperature:%s" % temperature)
    success = deviceCli.publishEvent("demo", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not Connected to ibm iot")
    time.sleep(5)
    deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()
```

Python Console output is shown below:

Sensor data is uploaded to IBM watson Cloud is shown Below: