Project Development Phase

Sprint – 1

Date	17 November 2022
Team ID	PNT2022TMID12777
Project Name	Project – Real Time River Water Quality Monitoring and Control System

Python Code

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

#Provide the IBM Watson Device Credentials
organization = "m21300"
deviceType = "iot_project"
deviceID = "31052000"
authMethod = "use-token-auth"
authToken = "31050308"

def myCommandCallBack(cmd):
   print("Command received: %s" %cmd.data['command'])
   status = cmd.data['command']
   if status =='lighton':
      print("Light ON")
   elif status == 'lightoff':
      print("Light 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()
deviceCli.connect()
while True:
   pH = random.randint(0,100)
   conductivity = random.randint(0,100)
   temperature = random.randint(0,100)
   oxygen = random.randint(0,100)
   turbidity = random.randint(0,100)
   sulphate = random.randint(0,100)
   chloride = random.randint(0,100)
   data = {"temperature":
temperature,'pH':pH,'conductivity':conductivity,'oxygen':oxygen,'turbidity':turbidity,'sulphat
e':sulphate,'chloride':chloride}
   def myOnPublishCallBack():
      print("Published data",data,"to IBM Watson")
   success = deviceCli.publishEvent("event","json",data,0,myOnPublishCallback)
   if not success:
      print ("Not connected to IoTF")
   time.sleep(5)
   deviceCli.commandCallback = myCommandCallback
```

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

#Provide the IBM Watson Device Credentials
organization = "m21300"
deviceType = "iot_project"
deviceID = "31052000"
authMethod = "use-token-auth"
authToken = "31050308"

def myCommandCallBack(cmd):
    print("Command received: %s" %cmd.data['command'])
    status = cmd.data['command']
    if status =='lighton':
        print("Light ON")
    elif status == 'lightoff':
        print("Light 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()

deviceCli.connect()

while True:
    pH = random.randint(0,100)
    conductivity = random.randint(0,100)
    temperature = random.randint(0,100)
    oxygen = random.randint(0,100)
    turbidity = random.randint(0,100)
    sulphate = random.randint(0,100)
    chloride = random.randint(0,100)
    data = {"temperature": temperature,'pH': pH,'conductivity':conductivity,'oxygen':oxygen,'turbidity':turbidity,
            'sulphate':sulphate,'chloride':chloride}

    def myOnPublishCallBack():
        print("Published data",data,"to IBM Watson")

    success = deviceCli.publishEvent("event","json",data,0,myOnPublishCallback)
    if not success:
        print ("Not connected to IoTF")
```