Project Development phase

Date	**09 November 2022**
Team ID	**PNT2022TMID41301**
Project Name	**Project – RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM**
Maximum Marks	**4 Marks**

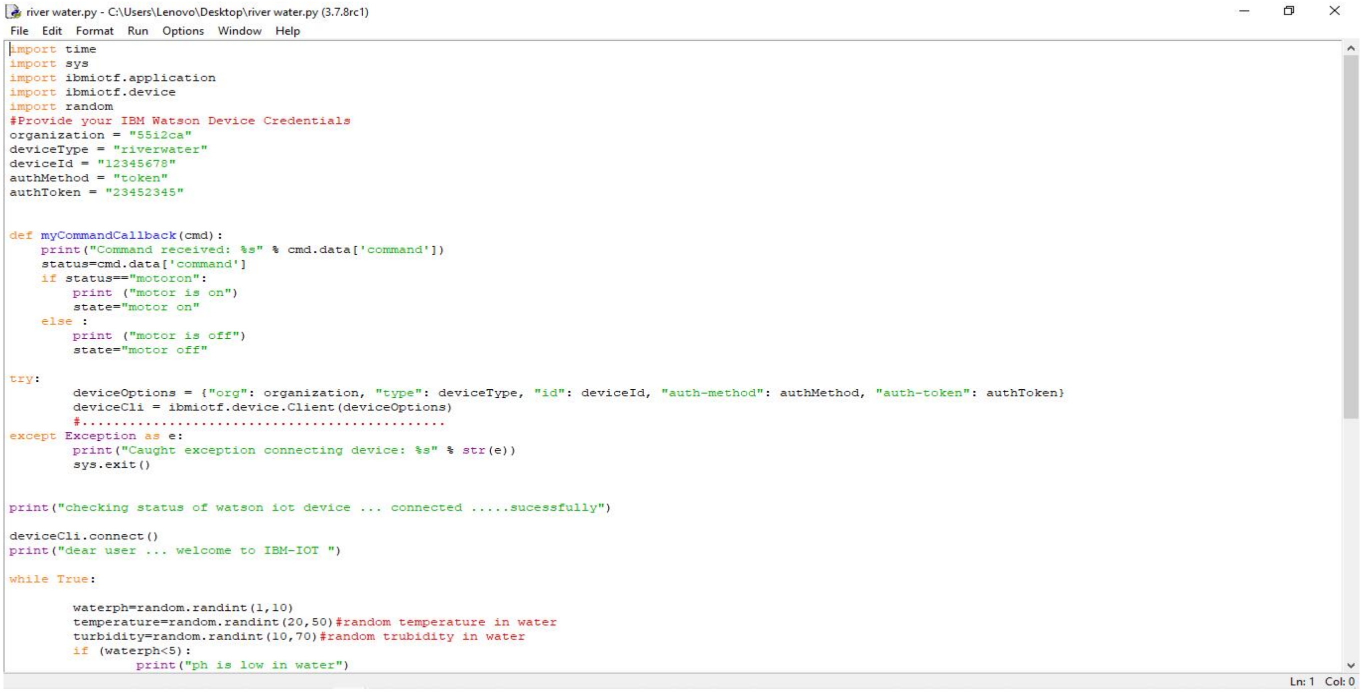
Delivering of Sprint-2

Python script:

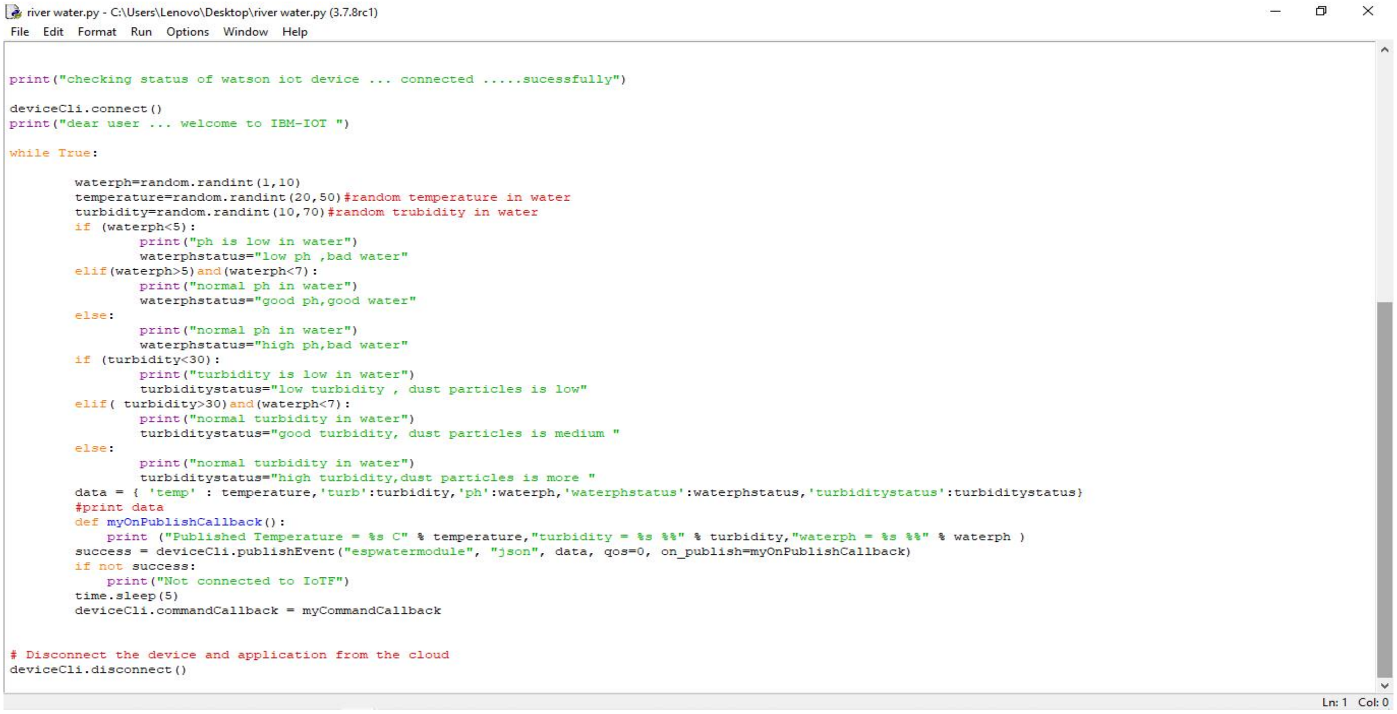
- We create a python code for motor status.

Python code (sending status of the motor):

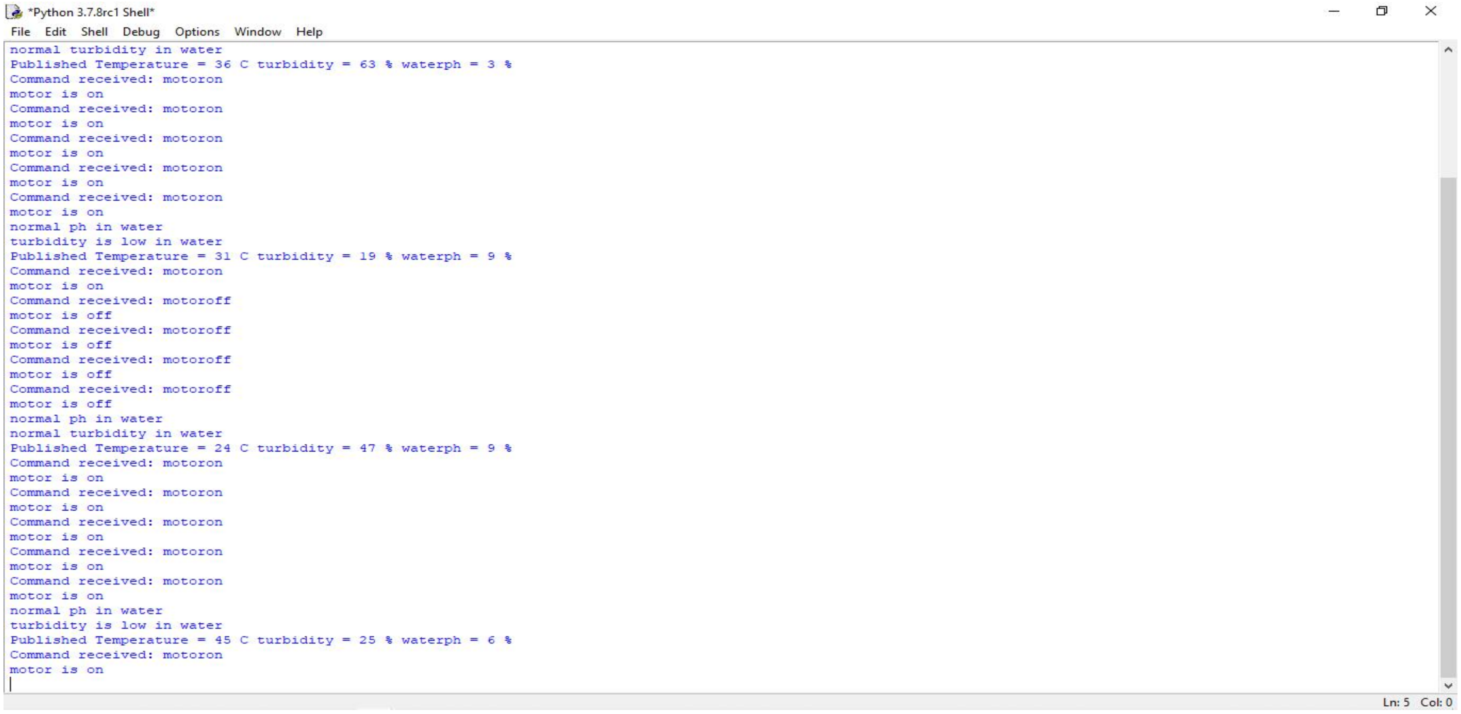
```
 import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "55i2ca"
deviceType = "riverwater"
deviceId = "12345678"
authMethod = "token"
authToken = "23452345"


def myCommandCallback(cmd):
   print("Command received: %s" % cmd.data['command'])
   status=cmd.data['command']
   if status=="motoron":
      print ("motor is on")
      state="motor on"
   else :
      print ("motor is off")
      state="motor 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()


print("checking status of watson iot device ... connected .....sucessfully")

deviceCli.connect()
print("dear user ... welcome to IBM-IOT ")

while True:

      waterph=random.randint(1,10)
      temperature=random.randint(20,50)#random temperature in water
      turbidity=random.randint(10,70)#random trubidity in water
      if (waterph<5):
          print("ph is low in water")
          waterphstatus="low ph ,bad water"
      elif(waterph>5)and(waterph<7):
          print("normal ph in water")
          waterphstatus="good ph,good water"
      else:
          print("normal ph in water")
          waterphstatus="high ph,bad water"
      if (turbidity<30):
          print("turbidity is low in water")
          turbiditystatus="low turbidity , dust particles is low"
      elif( turbidity>30)and(waterph<7):



 print("normal turbidity in water")
          turbiditystatus="good turbidity, dust particles is medium "
      else:
          print("normal turbidity in water")
          turbiditystatus="high turbidity,dust particles is more "
      data = { 'temp' :
temperature,'turb':turbidity,'ph':waterph,'waterphstatus':waterphstatus,'turbiditystatus':
turbiditystatus}

 #print data
      def myOnPublishCallback():
            print ("Published Temperature = %s C" % temperature,"turbidity = %s %%" % turbidity,"waterph = %s %%" % waterph )
        success = deviceCli.publishEvent("espwatermodule", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(5)
        deviceCli.commandCallback = myCommandCallback


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


river water.py - C:\Users\Lenovo\Desktop\river water.py (3.7.8rc1)
File Edit Format Run Options Window Help
Ln: 1 Col: 0

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "55i2ca"
deviceType = "riverwater"
deviceId = "12345678"
authMethod = "token"
authToken = "23452345"


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("motor is on")
        state="motor on"
    else :
        print ("motor is off")
        state="motor 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()


print("checking status of watson iot device ... connected .....sucessfully")

deviceCli.connect()
print("dear user ... welcome to IBM-IOT ")

while True:

        waterph=random.randint(1,10)
        temperature=random.randint(20,50)#random temperature in water
        turbidity=random.randint(10,70)#random trubidity in water
        if (waterph<5):
                print("ph is low in water")
```


river water.py - C:\Users\Lenovo\Desktop\river water.py (3.7.8rc1)
File Edit Format Run Options Window Help
Ln: 1 Col: 0

```
print("checking status of watson iot device ... connected .....sucessfully")

deviceCli.connect()
print("dear user ... welcome to IBM-IOT ")

while True:

        waterph=random.randint(1,10)
        temperature=random.randint(20,50)#random temperature in water
        turbidity=random.randint(10,70)#random trubidity in water
        if (waterph<5):
                print("ph is low in water")
                waterphstatus="low ph ,bad water"
        elif(waterph>5)and(waterph<7):
                print("normal ph in water")
                waterphstatus="good ph,good water"
        else:
                print("normal ph in water")
                waterphstatus="high ph,bad water"
        if (turbidity<30):
                print("turbidity is low in water")
                turbiditystatus="low turbidity , dust particles is low"
        elif( turbidity>30)and(waterph<7):
                print("normal turbidity in water")
                turbiditystatus="good turbidity, dust particles is medium "
        else:
                print("normal turbidity in water")
                turbiditystatus="high turbidity,dust particles is more "
        data = { 'temp' : temperature,'turb':turbidity,'ph':waterph,'waterphstatus':waterphstatus,'turbiditystatus':turbiditystatus}
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temperature,"turbidity = %s %%" % turbidity,"waterph = %s %%" % waterph )
        success = deviceCli.publishEvent("espwatermodule", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(5)
        deviceCli.commandCallback = myCommandCallback


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



```
normal turbidity in water
Published Temperature = 36 C turbidity = 63 % waterph = 3 %
Command received: motoron
motor is on
Command received: motoron
motor is on
Command received: motoron
motor is on
Command received: motoron
motor is on
Command received: motoron
motor is on
normal ph in water
turbidity is low in water
Published Temperature = 31 C turbidity = 19 % waterph = 9 %
Command received: motoron
motor is on
Command received: motoroff
motor is off
Command received: motoroff
motor is off
Command received: motoroff
motor is off
Command received: motoroff
motor is off
normal ph in water
normal turbidity in water
Published Temperature = 24 C turbidity = 47 % waterph = 9 %
Command received: motoron
motor is on
Command received: motoron
motor is on
Command received: motoron
motor is on
Command received: motoron
motor is on
Command received: motoron
motor is on
normal ph in water
turbidity is low in water
Published Temperature = 45 C turbidity = 25 % waterph = 6 %
Command received: motoron
motor is on
```