**REAL-TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM**

**SPRINT-1**

**PYTHON SOURCE CODE:**

import ibmiotf.application

import ibmiotf.device

import time

import random

import sys

import requests

import json

import urllib.request

import urllib.parse

url="https://www.fast2sms.com/dev/bulkV2"

organization = "swz5ou"

deviceType = "abcd"

deviceId = "12"

authMethod = "token"

authToken = "12345678"

def sms(ph,temp,turbidity):

message='Water quality degraded PH value:'+str(ph)+'temperature value:'+str(temp)+'tubidity value:'+str(turbidity)

my\_data = {

'sender\_id': 'TXTIND',

'message': message,

'language': 'english',

'route': 'p',

'numbers': '9150661026, 6369521344,9840981094'

}

headers = {

'authorization': 'cjshq2uY05KWVOxSDndGMNyvAmR6rgzfUpI3Pe8JkE49ZXlBbwq2plfEB6IZ31CjywSchzNtRQkixoV0',

'Content-Type': "application/x-www-form-urlencoded",

'Cache-Control': "no-cache"

}

response = requests.request("POST",url,data=my\_data,headers=headers)

returned\_msg = json.loads(response.text)

print(returned\_msg['message'])

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 pH<6 or temperature >120 or turbidity > 500:

alert = 1

else:

alert = 0

data = {'pH': pH, 'turbid': turbidity, 'temp': temperature, 'alert':alert}

def myOnPublishCallback():

print("Published pH= %s" % pH, "Turbidity:%s" % turbidity, "Temperature:%s" % temperature)

success = deviceCli.publishEvent("water monitoring", "json", data, qos=0, on\_publish=myOnPublishCallback)

if not success:

print("Not Connected to ibmiot")

time.sleep(1)

deviceCli.disconnect()

**OUTPUT:**

