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-3

Node-RED Service

- the Node-RED flow to receive data from the IBM IoT platform.
- And also use Cloudant DB nodes to store the received sensor data in the cloudant DB.
- To create use dashboard nodes to visualize the data in graphical format.
- Create an HTTP API for communicating with Mobile applications.

Building Mobile App

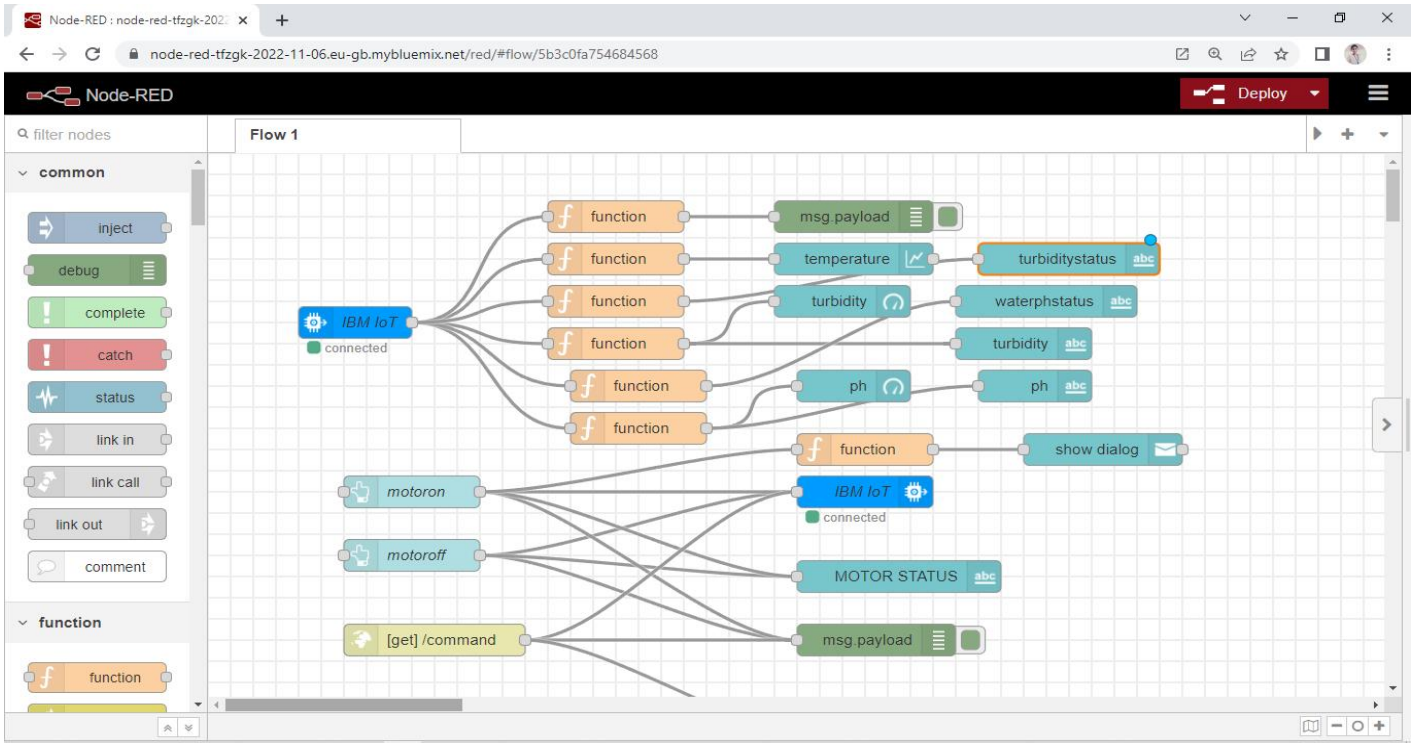
- we will build a basic mobile application to show the sensor data.
- Design UI to display the Water Turbidity, and pH values sensor values.
- the application to receive the data from the cloud.
- the mobile app to send commands to users using buttons.

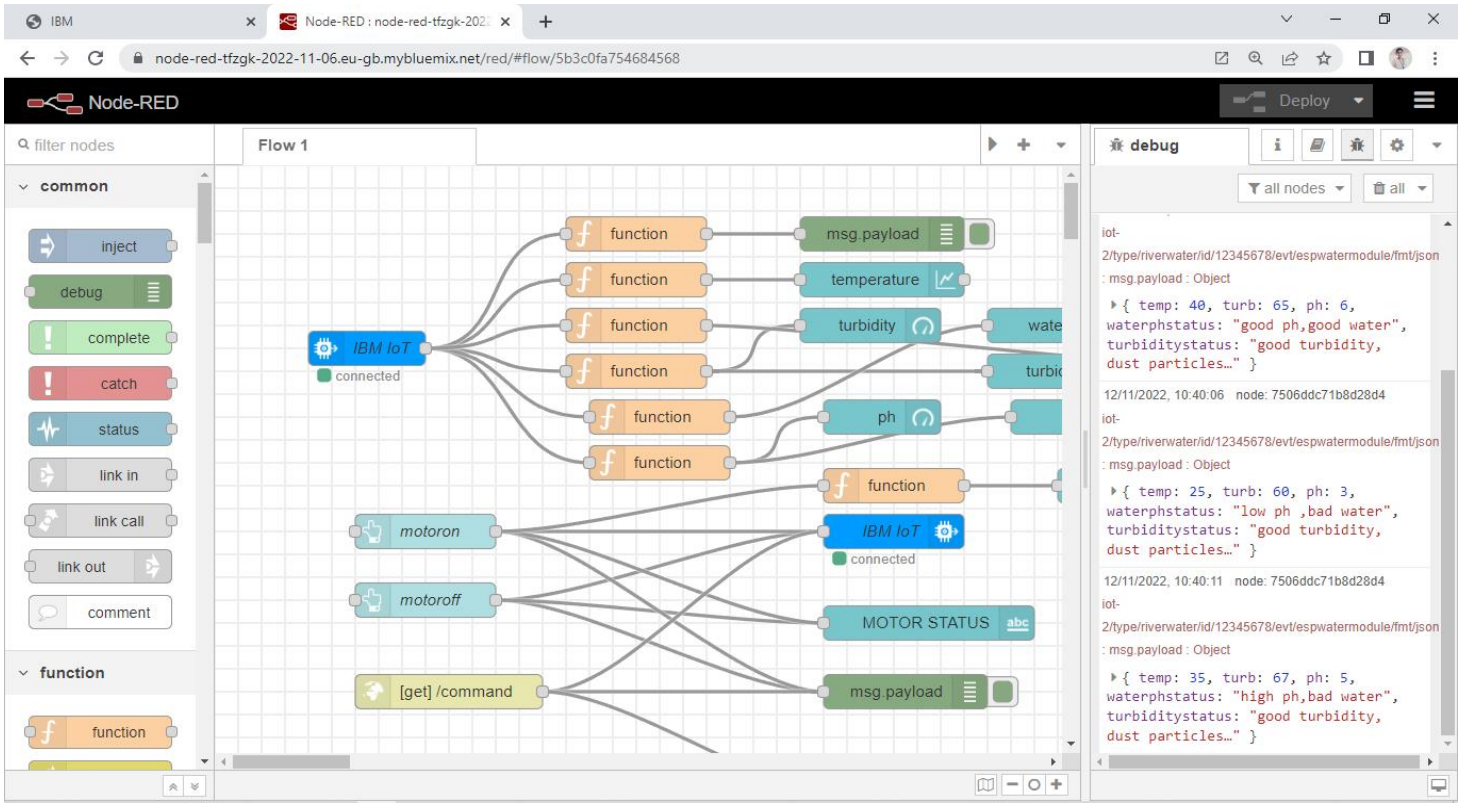
Python code test code:

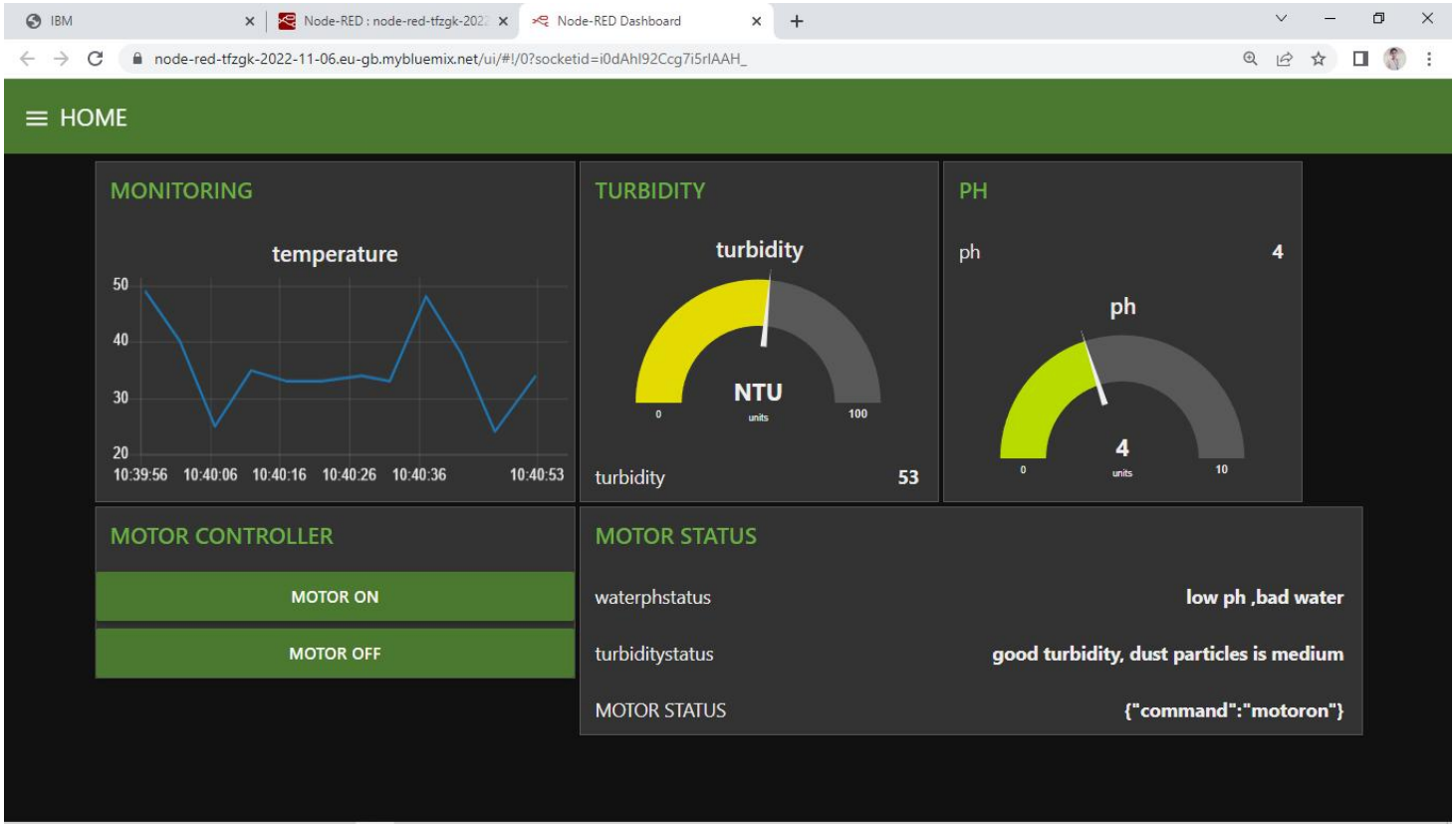
```
    import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "55i2ca"
deviceType = "riverwater"
deviceId = "12345678"
authMethod = "token"
authToken = "23452345"
```

```
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"
```

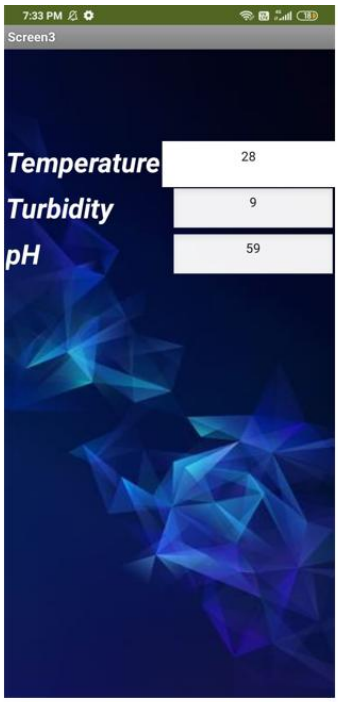
```
    if (turbidity<30):
        print("turbidity is low in water")
        turbiditystatus="low turbidity , dust particles is low"
elif( turbidity>30)and(waterph<7):
        print("normal turbidity in water")
        turbiditystatus="good turbidity, dust particles is medium "
    else:
        print("normal 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()
```


Node-RED : node-red-tfzgk-202
node-red-tfzgk-2022-11-06.eu-gb.mybluemix.net/red/#flow/5b3c0fa754684568
Node-RED
Deploy
filter nodes
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
function
function
function
function
function
function
function
msg.payload
temperature
turbidity
ph
turbiditystatus
waterphstatus
turbidity
ph
show dialog
IBM IoT
connected
motoron
motoroff
[get] /command
IBM IoT
connected
MOTOR STATUS
msg.payload


IBM
Node-RED : node-red-tfzgk-202
node-red-tfzgk-2022-11-06.eu-gb.mybluemix.net/red/#flow/5b3c0fa754684568
Node-RED
Deploy
filter nodes
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
function
function
function
function
function
function
function
msg.payload
temperature
turbidity
ph
IBM IoT
connected
motoron
motoroff
[get] /command
IBM IoT
connected
MOTOR STATUS
msg.payload
debug
all nodes
all
iot-2/type/riverwater/id/12345678/evt/espwatermodule/fmt/json : msg.payload : Object
{ temp: 40, turb: 65, ph: 6, waterphstatus: "good ph,good water", turbiditystatus: "good turbidity, dust particles…" }
12/11/2022, 10:40:06 node: 7506ddc71b8d28d4
iot-2/type/riverwater/id/12345678/evt/espwatermodule/fmt/json : msg.payload : Object
{ temp: 25, turb: 60, ph: 3, waterphstatus: "low ph ,bad water", turbiditystatus: "good turbidity, dust particles…" }
12/11/2022, 10:40:11 node: 7506ddc71b8d28d4
iot-2/type/riverwater/id/12345678/evt/espwatermodule/fmt/json : msg.payload : Object
{ temp: 35, turb: 67, ph: 5, waterphstatus: "high ph,bad water", turbiditystatus: "good turbidity, dust particles…" }


IBM
Node-RED : node-red-tfzgk-202
Node-RED Dashboard
node-red-tfzgk-2022-11-06.eu-gb.mybluemix.net/ui/#!/0?socketid=i0dAhI92Ccg7i5rIAAH_
HOME
MONITORING
temperature
50
40
30
20
10:39:56
10:40:06
10:40:16
10:40:26
10:40:36
10:40:53
TURBIDITY
turbidity
NTU
units
0
100
turbidity
53
PH
ph
4
ph
4
units
0
10
MOTOR CONTROLLER
MOTOR ON
MOTOR OFF
MOTOR STATUS
waterphstatus
low ph ,bad water
turbiditystatus
good turbidity, dust particles is medium
MOTOR STATUS
{"command":"motoron"}

MIT APP INVERTOR OUTPUT


7:33 PM
Screen3
Temperature
28
Turbidity
9
pH
59


7:33 PM
Screen4
MOTOR ON
MOTOR OFF