**ASSIGNMENT-3**

Develop a code to upload the water tank level and light intensity values to the IBM IoT platform and visualize them in the web application.

**Python code**

import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

import json

organization = "vikkhq"

deviceType = "iotdevice"

deviceId = "1001"

authMethod = "token"

authToken = "1234567890"

w=0

l=0

def myCommandCallback(cmd):

        print("Command received: %s" % cmd.data['command'])

        if cmd.data['command']=='lighton':

                print("LIGHT ON IS RECEIVED")

        elif cmd.data['command']=='lightoff':

                print("LIGHT OFF IS RECEIVED")

        if cmd.command == "setInterval":

                if 'interval' not in cmd.data:

                        print("Error - command is missing required information: 'interval'")

                else:

                  interval = cmd.data['interval']

        elif cmd.command == "print":

                if 'message' not in cmd.data:

                        print("Error - command is missing required information: 'message'")

                else:

                        print(cmd.data['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:

        w=70

        l=30

        data = {"d":{ 'waterlevel': w,'lightintensity': l}}

        #print data

        def myOnPublishCallback():

            print ("Published Waterlevel = %s %%"  %w, "Lightintensity = %s %%"  %l, "to IBM Watson")

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

        if not success:

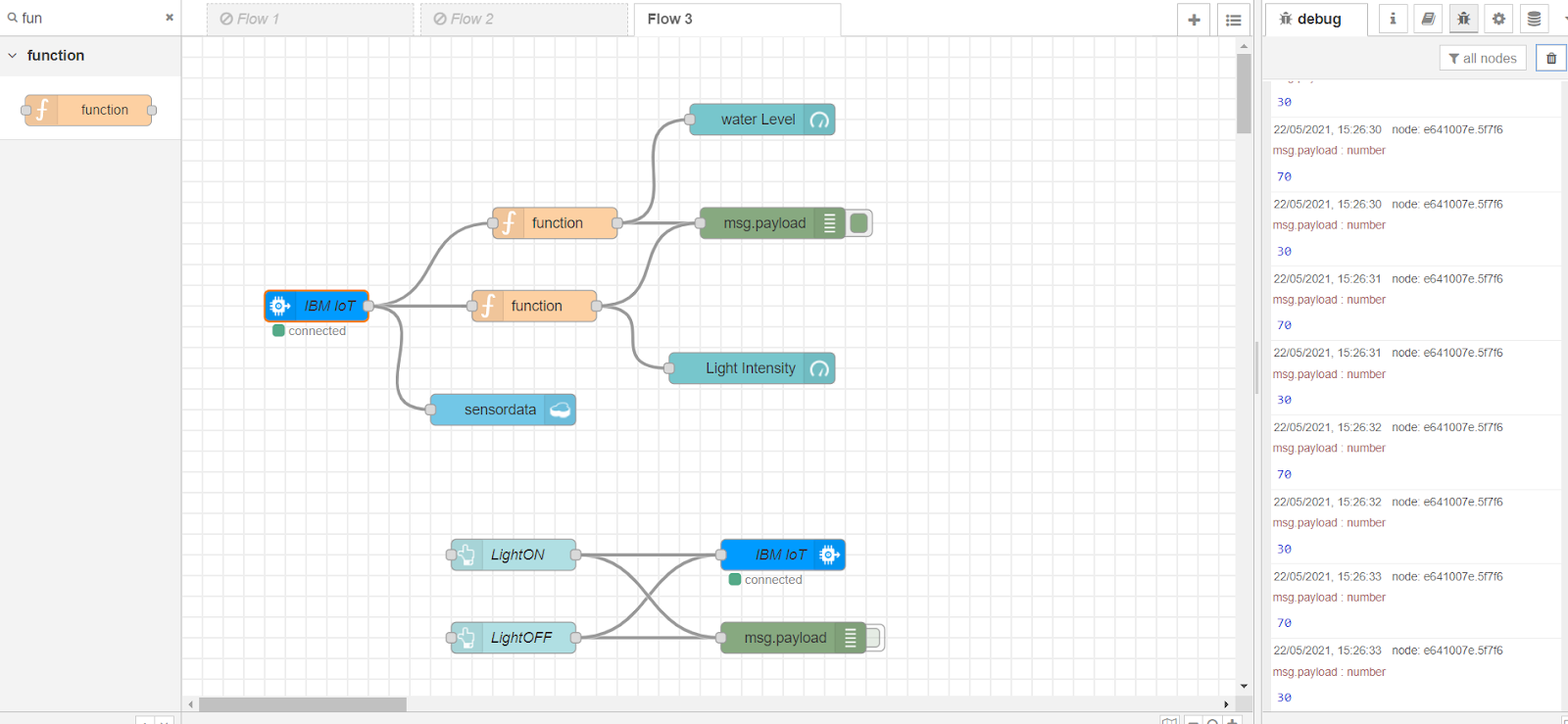
            print("Not connected to IoTF")

        time.sleep(1)

        deviceCli.commandCallback = myCommandCallback

deviceCli.disconnect()

**Node-RED blocks:**



**UI:**

