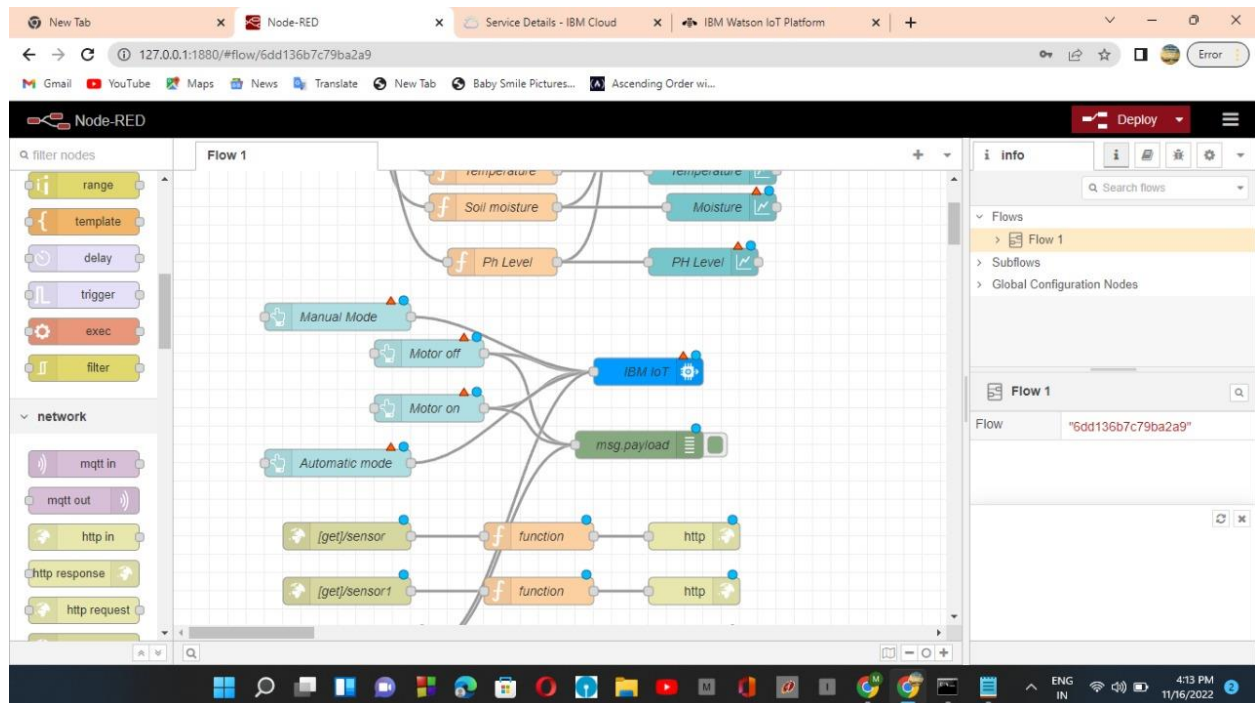


Sprint 2

Date	31 October 2022
Team ID	PNT2022TMID30750
Project Name	Project - SmartFarmer - IoT Enabled Smart Farming Application

- Created the device on IBM cloud platform and the node red platform to set the iot device workflow.



IBM Watson IoT Platform

Browse IBM Cloud Apps

Generate API Key

Browse API Keys

This table shows a summary of the API keys that have been added for the organization. It can be filtered, organized, and search on using different criteria. To get started, you can add API keys by clicking Generate API Key, or by using the API. For more information about adding API keys, see [API key connection](#).

Type the app description to search for

Key	Description	Role	Expires
6 results			
a-tuwf7p-8hwp70k3fu	-	Standard Application	-
a-tuwf7p-c8q76zifbz	-	Standard Application	-
a-tuwf7p-cas0rloblg	-	Standard Application	-

API Key Information Access Control/Permissions 0 Simulations running

```
smartform.py - C:/Users/ADMIN/Documents/smartform.py (3.11.0)
File Edit Format Run Options Window Help

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

#provide your IBM Watson Device Credentials
organization = "138693"
deviceType = "abod"
deviceId = "1234"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    Status=cmd.data['command']
    if status=="MotorOn":
        print ("Motor is On")
    elif status == "Motor Off":
        print ("Motor is Off")
    else :
        print ("please send proper command")
try:
    deviceOptions = {"org": organization, "type": deviceType, "id":deviceId}
    devicecli = ibmiotf.device.client(deviceOptions)
    #.....
except Exception as e:
    print("Caught exception connecting device: %s"%str(e))
    sys.exit()

# connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
devicecli.connect()

while True:
    #get sensor data from DHT11
    temp=random.randint(90,110)
    Humid=random.randint(60,100)

    data = { 'temp' : temp, 'Humid' : Humid }
```

Node-RED interface showing a flow named "Flow 7". The flow starts with an "IBM IoT" node (connected), which connects to three function nodes: "temp", "hum", and "moisture". These function nodes connect to three output nodes: "Temp", "HUM", and "Moisture". A "msg.payload" node is also connected to the output nodes. The interface includes a sidebar with various nodes (gauge, notification, template, button, dropdown, switch, slider, audio out, chart, ui control, numeric) and a dashboard section on the right with tabs for "Smart Home", "HALL AC", and "GARDEN".

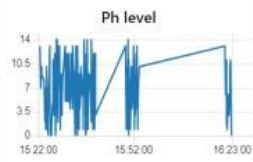
Smart Home dashboard interface. The dashboard displays two main sections: "HALL AC" and "GARDEN".

HALL AC:

- HUM:** A line graph showing humidity levels over time. The y-axis ranges from 60 to 100. The x-axis shows times 14:57:00, 15:01:00, and 15:05:00. The graph shows a peak in humidity around 15:01:00.
- Temp:** A line graph showing temperature levels over time. The y-axis ranges from 90 to 110. The x-axis shows times 14:57:00, 15:01:00, and 15:05:00. The graph shows a peak in temperature around 15:01:00.

GARDEN:

- Water moisture:** A gauge showing water moisture levels. The gauge is a semi-circle with a needle pointing to 93. The scale ranges from 0 to 100.



MOTOR OFF

MOTOR ON

MANUAL MODE

AUTOMATIC MODE

