

SPRINT-3

Team ID	PNT2022TMID42646
<u>Project Name</u>	Smart farmer - IoT Enabled smart farming application.

Project Tracker

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	15	5 Days	26 Oct 2022	30 Oct 2022	15	30 Oct 2022
Sprint-2	15	7 Days	31 Oct 2022	06 Nov 2022		07 Nov 2022
Sprint-3	15	6 Days	07 Nov 2022	12 Nov 2022		13 Nov 2022
Sprint-4	15	6 Days	13 Nov 2022	18 Nov 2022		18 Nov 2022 – 19 Nov 2022

S.NO	Tools & Technology Used
1	Python 3.7.0
2	IBM Watson
3	Node-Red

PYTHON CODE

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

#Provide your IBM Watson Device Credentials
organization = "x0fxss" #replace the ORG ID
deviceType = "Testing"#replace the Device type
wi deviceId = "Testdevice1"#replace Device ID
authMethod = "token" authToken = "123456789"
#Replace the authtoken
# Initialize GPIO

#Receives Command from Node-red def
myCommandCallback(cmd):
    print ("Command received: %s" %
cmd.data['command'])__ status=cmd.data['command']
if status=="motoron":
    print ("motor is on")__
elif status == "motoroff" :
print ("motor is off") elif
status == "motor30" :
    print ("motor is on for 30 minutes")
    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()

# 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(0,100)
    Humid=random.randint(0,100)
    soilmoisture=random.randint(0,100)
    data = { 'temp' : temp, 'Humid': Humid, 'soilmoisture':
soilmoisture }
    #print data
    def
myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %" % Humid, "soilmoisture = %s %"
%soilmoisture, "to IBM Watson")
        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoT")
    time.sleep(5)
    deviceCli.commandCallback =
myCommandCallback

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

```

Output

```
*Python 3.7.0 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\charu\Downloads\ibmiotpublishsubscribe.py =====
2022-11-11 15:56:49,907 ibmiotf.device.Client INFO Connected successfully: d:x0fxss:Testing:Testdevice1
Published Temperature = 8 C Humidity = 44 % soilmoisture = 3 % to IBM Watson
Published Temperature = 13 C Humidity = 95 % soilmoisture = 43 % to IBM Watson
Published Temperature = 78 C Humidity = 83 % soilmoisture = 83 % to IBM Watson
Published Temperature = 100 C Humidity = 52 % soilmoisture = 60 % to IBM Watson
Published Temperature = 45 C Humidity = 93 % soilmoisture = 16 % to IBM Watson
Published Temperature = 53 C Humidity = 12 % soilmoisture = 59 % to IBM Watson
Published Temperature = 15 C Humidity = 49 % soilmoisture = 32 % to IBM Watson
Published Temperature = 37 C Humidity = 73 % soilmoisture = 25 % to IBM Watson
```

IBM Watson

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar labeled 'Search by Device ID' is present. The main content area shows a table of devices, with one device 'iot_device_1' selected. Below the table, a 'Recent Events' section displays a stream of data events. A notification at the bottom right indicates '1 Simulation running'.

IBM Watson IoT Platform

prathipathangaraj@gmail.com
ID: ma3ge3

Add Device +

Device Simulator ☒

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
iot_device_1	Connected	iot_device	Device	Nov 12, 2022 3:57 PM	

Identity Device Information **Recent Events** State Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event_1	{"temperature":12,"humidity":10,"soil_moisture"...	json	a few seconds ago
event_1	{"temperature":32,"humidity":30,"soil_moisture"...	json	a few seconds ago
event_1	{"temperature":47,"humidity":33,"soil_moisture"...	json	a few seconds ago
event_1	{"temperature":7,"humidity":33,"soil_moisture":...	json	a few seconds ago

1 Simulation running

Python Script.pdf Sprint 3 Report.pdf Screenshot_20221...jpg IBM Watson & N....docx App_develop.aia Show all

Node-Red

The screenshot displays the Node-RED web interface in a browser. The main workspace shows a flow named 'Flow 2' with the following components:

- smartfarmer_db** (Database) connected to three function nodes (**f**) for **Temperature**, **Humidity**, and **Soil Moisture**.
- Each function node is connected to a corresponding output node (**Temperature**, **Humidity**, **Soil Moisture**).
- MotorON** and **MotorOFF** (Control) nodes are connected to a **msg_payload** (Message) node.
- The **msg_payload** node is connected to an **IBM IoT** (IoT) node.
- Below, there are two **[get] /control** and **[get] /data** nodes, each connected to a **f** (function) node, which then connects to an **http** (HTTP) node.
- At the bottom, an **inject** node is connected to a **smartfarmer_db** node, which is connected to a **msg payload** node.

The right sidebar shows the **debug** console with a list of messages. The messages are all of type **msg_payload** and have a payload of **motoron**. The messages are timestamped from 11/12/2022, 4:37:38 PM to 11/12/2022, 4:38:01 PM.

Data are successfully received and displayed



