SPRINT 2

Date	13th November 2022
Team ID	PNT2022TMID34110
Project Name	SmartFarmer –IoT Enabled Smart Farming Application

Develop a python script to publish random sensor data :

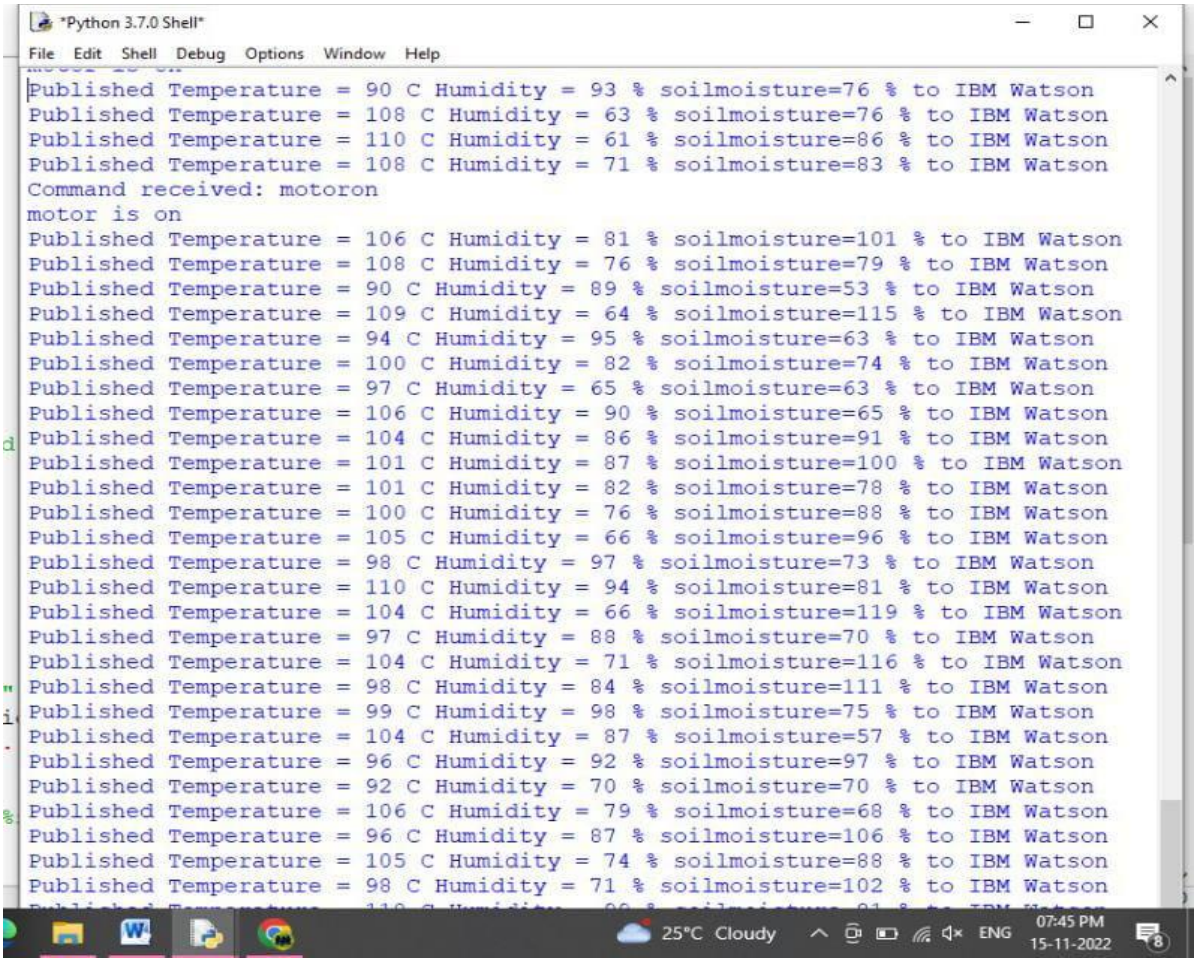
Program:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "6pjvs7"
deviceType = "Arshidevicetype"
deviceId = "Arshideviceid"
authMethod = "token"
authToken = "tGfGvVl-F2luRl2bsG"
# Initialize GPIO
def myCommandCallback(cmd):
```

```
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("motor is on")
    else :
        print ("motor is off")
    #print(cmd)
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)
```

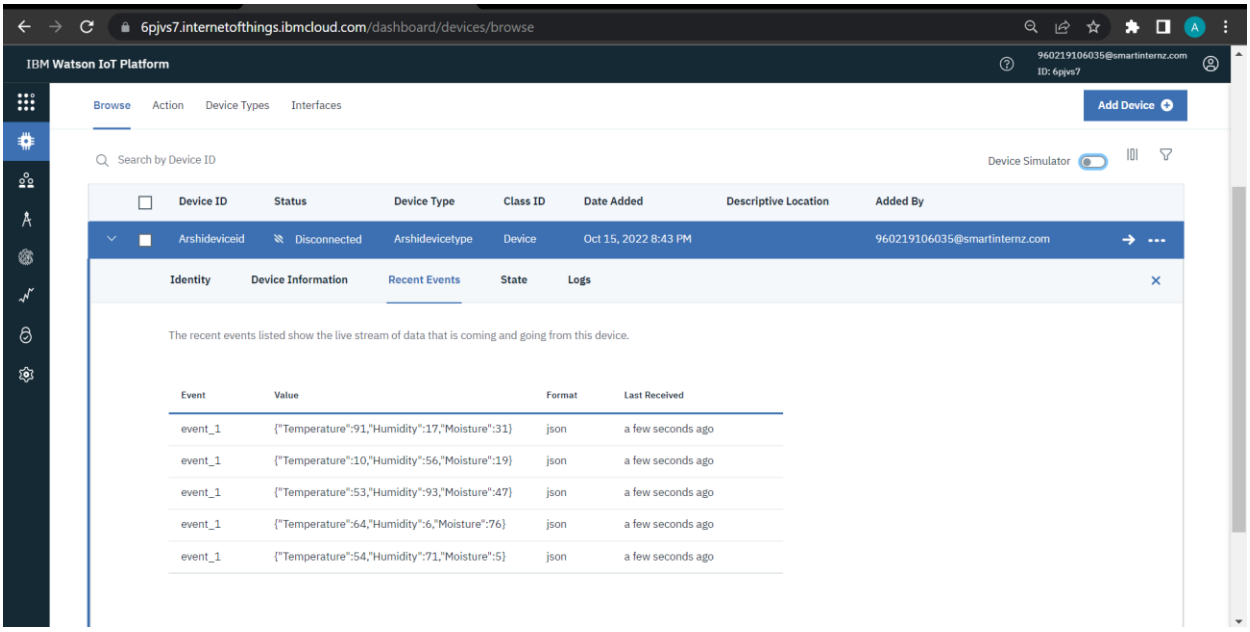
```
        Humid=random.randint(0,100)
        moist=random.randint(0,100)
        data = { 'Temperature' : temp, 'Humidity': Humid,
'Moisture':moist}
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp,
"Humidity = %s %%" % Humid, "to IBM Watson")
        success = deviceCli.publishEvent("IoTSensor", "json",
data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)
        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
Published Temperature = 90 C Humidity = 93 % soilmoisture=76 % to IBM Watson
Published Temperature = 108 C Humidity = 63 % soilmoisture=76 % to IBM Watson
Published Temperature = 110 C Humidity = 61 % soilmoisture=86 % to IBM Watson
Published Temperature = 108 C Humidity = 71 % soilmoisture=83 % to IBM Watson
Command received: motoron
motor is on
Published Temperature = 106 C Humidity = 81 % soilmoisture=101 % to IBM Watson
Published Temperature = 108 C Humidity = 76 % soilmoisture=79 % to IBM Watson
Published Temperature = 90 C Humidity = 89 % soilmoisture=53 % to IBM Watson
Published Temperature = 109 C Humidity = 64 % soilmoisture=115 % to IBM Watson
Published Temperature = 94 C Humidity = 95 % soilmoisture=63 % to IBM Watson
Published Temperature = 100 C Humidity = 82 % soilmoisture=74 % to IBM Watson
Published Temperature = 97 C Humidity = 65 % soilmoisture=63 % to IBM Watson
Published Temperature = 106 C Humidity = 90 % soilmoisture=65 % to IBM Watson
Published Temperature = 104 C Humidity = 86 % soilmoisture=91 % to IBM Watson
Published Temperature = 101 C Humidity = 87 % soilmoisture=100 % to IBM Watson
Published Temperature = 101 C Humidity = 82 % soilmoisture=78 % to IBM Watson
Published Temperature = 100 C Humidity = 76 % soilmoisture=88 % to IBM Watson
Published Temperature = 105 C Humidity = 66 % soilmoisture=96 % to IBM Watson
Published Temperature = 98 C Humidity = 97 % soilmoisture=73 % to IBM Watson
Published Temperature = 110 C Humidity = 94 % soilmoisture=81 % to IBM Watson
Published Temperature = 104 C Humidity = 66 % soilmoisture=119 % to IBM Watson
Published Temperature = 97 C Humidity = 88 % soilmoisture=70 % to IBM Watson
Published Temperature = 104 C Humidity = 71 % soilmoisture=116 % to IBM Watson
Published Temperature = 98 C Humidity = 84 % soilmoisture=111 % to IBM Watson
Published Temperature = 99 C Humidity = 98 % soilmoisture=75 % to IBM Watson
Published Temperature = 104 C Humidity = 87 % soilmoisture=57 % to IBM Watson
Published Temperature = 96 C Humidity = 92 % soilmoisture=97 % to IBM Watson
Published Temperature = 92 C Humidity = 70 % soilmoisture=70 % to IBM Watson
Published Temperature = 106 C Humidity = 79 % soilmoisture=68 % to IBM Watson
Published Temperature = 96 C Humidity = 87 % soilmoisture=106 % to IBM Watson
Published Temperature = 105 C Humidity = 74 % soilmoisture=88 % to IBM Watson
Published Temperature = 98 C Humidity = 71 % soilmoisture=102 % to IBM Watson
25°C Cloudy  ENG  07:45 PM 15-11-2022
```

Publish data to IBM cloud



Connect the circuit with the IBM Cloudant API integration

<u>Program:</u>

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN 15     // what pin we're connected to
#define DHTTYPE DHT22   // define type of sensor DHT 11
#define LED 2

DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of dht connected
```

```
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
//-------credentials of IBM Accounts------
#define ORG "6pjvs7"//IBM ORGANITION ID
#define DEVICE_TYPE "Arshidevicetype"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "Arshideviceid"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "tGfGvVl-F2luRl2bsG"     //Token
String data3;
float h, t;

//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id

//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by passing parameter like server id,portand wificredential
void setup()// configureing the ESP32
{
  Serial.begin(115200);
  dht.begin();
  pinMode(LED,OUTPUT);
  delay(10);
```

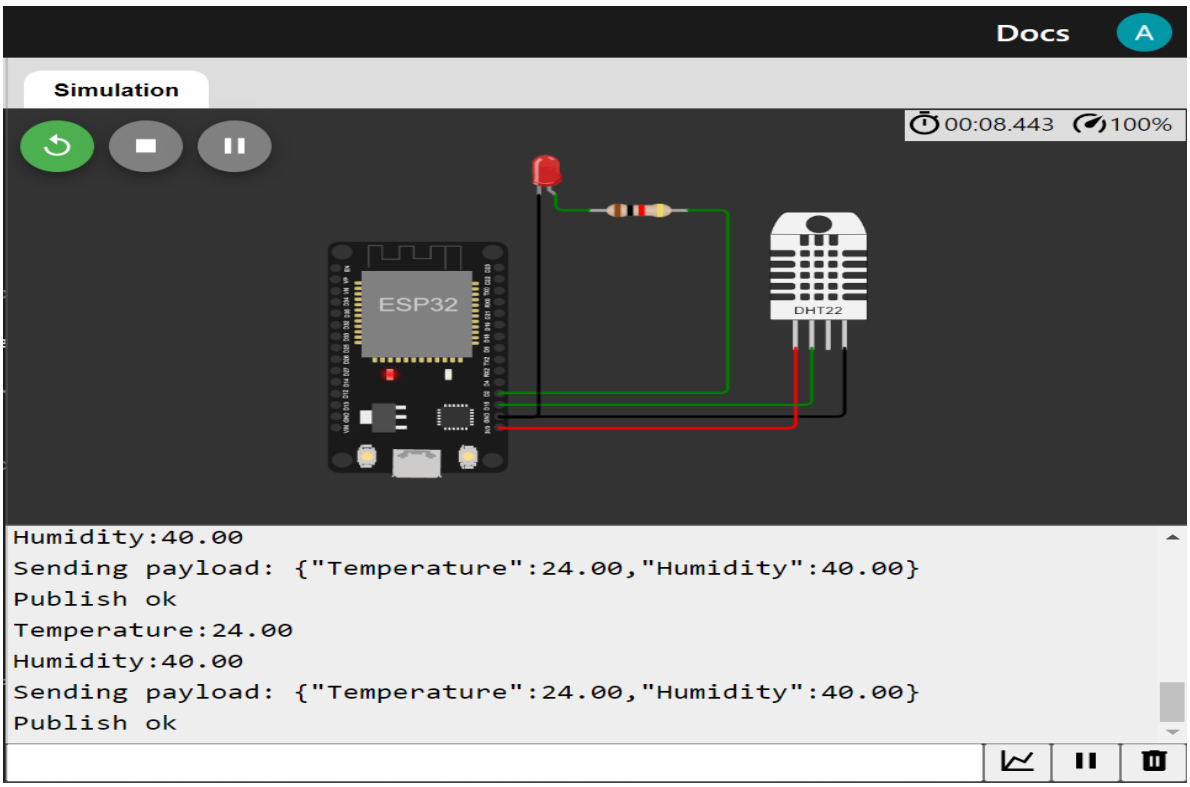
```
  Serial.println();
  wificonnect();
  mqttconnect();
}
void loop()// Recursive Function
{
  h = dht.readHumidity();
  t = dht.readTemperature();
  Serial.print("Temperature:");
  Serial.println(t);
  Serial.print("Humidity:");
  Serial.println(h);
  PublishData(t, h);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}
/*.................................retrieving to Cloud.............................*/
void PublishData(float temp, float humid) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"Temperature\":";
  payload += temp;
  payload += "," "\"Humidity\":";
  payload += humid;
  payload += "}";
```

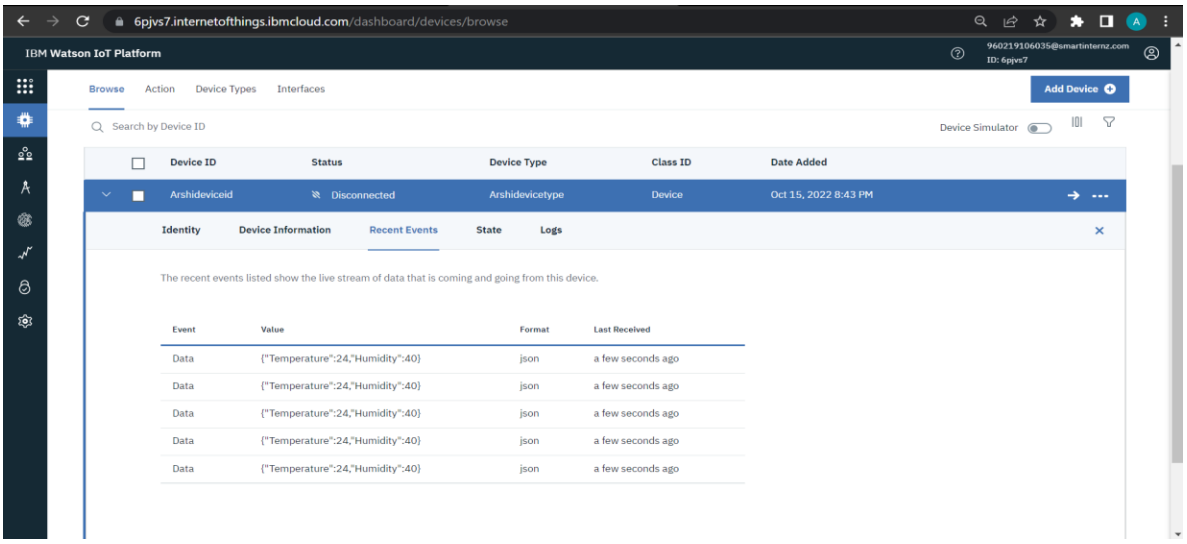
```
  Serial.print("Sending payload: ");
  Serial.println(payload);
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it will print publish ok in Serial monitor or else it will print publish failed
  } else {
    Serial.println("Publish failed");
  }
}
void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }
     initManagedDevice();
     Serial.println();
  }
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");
  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
```

```
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{
  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);
  if(data3=="lighton")
  {
Serial.println(data3);
digitalWrite(LED,HIGH);
  }
  else
```

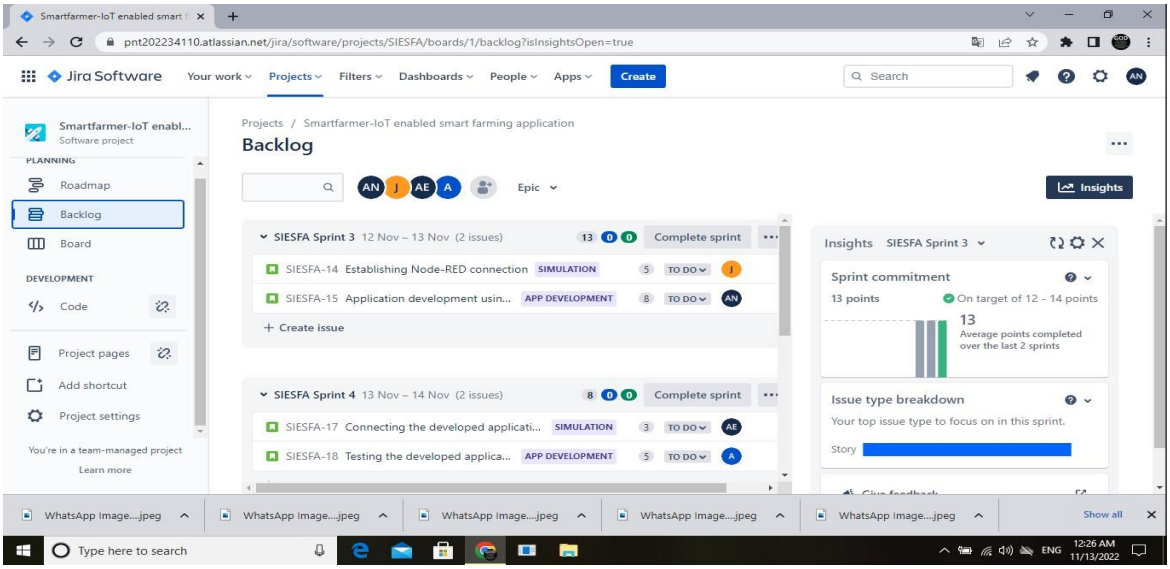
```
  {
Serial.println(data3);
digitalWrite(LED,LOW);
  }
data3="";
}
```

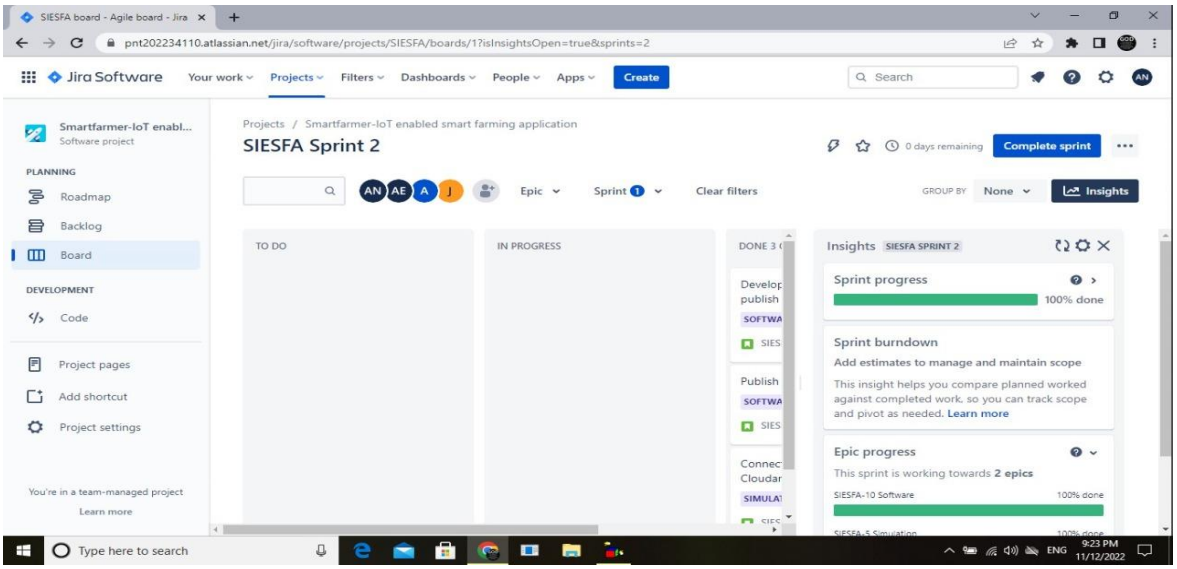
Circuit & Output:

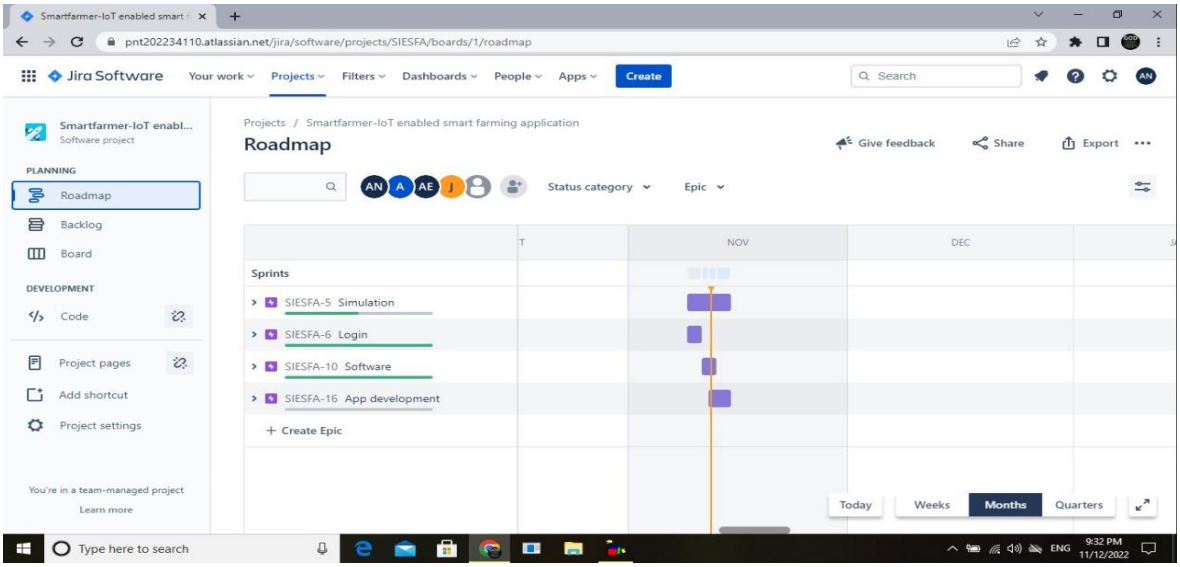



6pjvs7.internetofthings.ibmcloud.com/dashboard/devices/browse
IBM Watson IoT Platform
Browse
Action
Device Types
Interfaces
Add Device
Search by Device ID
Device Simulator
Device ID
Status
Device Type
Class ID
Date Added
Arshideviceid
Disconnected
Arshidevicetype
Device
Oct 15, 2022 8:43 PM
Identity
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
Data
{"Temperature":24,"Humidity":40}
json
a few seconds ago

REPORTS:


pnt202234110.atlassian.net/jira/software/projects/SIESFA/boards/1/backlog?isInsightsOpen=true
Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Smartfarmer-IoT enabl...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / Smartfarmer-IoT enabled smart farming application
Backlog
Epic
Insights
SIESFA Sprint 3 12 Nov – 13 Nov (2 issues)
Complete sprint
SIESFA-14 Establishing Node-RED connection SIMULATION
SIESFA-15 Application development usin... APP DEVELOPMENT
Create issue
SIESFA Sprint 4 13 Nov – 14 Nov (2 issues)
SIESFA-17 Connecting the developed applicati... SIMULATION
SIESFA-18 Testing the developed applica... APP DEVELOPMENT
Insights SIESFA Sprint 3
Sprint commitment
13 points
On target of 12 - 14 points
13
Average points completed over the last 2 sprints
Issue type breakdown
Your top issue type to focus on in this sprint.
Story
Type here to search
12:26 AM
11/13/2022


SIESFA board - Agile board - Jira
pnt202234110.atlassian.net/jira/software/projects/SIESFA/boards/1?isInsightsOpen=true&sprints=2
Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Smartfarmer-IoT enabl...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / Smartfarmer-IoT enabled smart farming application
SIESFA Sprint 2
0 days remaining
Complete sprint
Epic
Sprint 1
Clear filters
GROUP BY
None
Insights
TO DO
IN PROGRESS
DONE 3
Insights SIESFA SPRINT 2
Sprint progress
100% done
Sprint burndown
Add estimates to manage and maintain scope
This insight helps you compare planned worked against completed work, so you can track scope and pivot as needed. Learn more
Epic progress
This sprint is working towards 2 epics
SIESFA-10 Software
100% done
Type here to search
ENG
9:23 PM
11/12/2022


Smartfarmer-IoT enabled smart
pnt202234110.atlassian.net/jira/software/projects/SIESFA/boards/1/roadmap
Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Smartfarmer-IoT enabl...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / Smartfarmer-IoT enabled smart farming application
Roadmap
Give feedback
Share
Export
Status category
Epic
NOV
DEC
Sprints
SIESFA-5 Simulation
SIESFA-6 Login
SIESFA-10 Software
SIESFA-16 App development
Create Epic
Today
Weeks
Months
Quarters
Type here to search
ENG
9:32 PM
11/12/2022