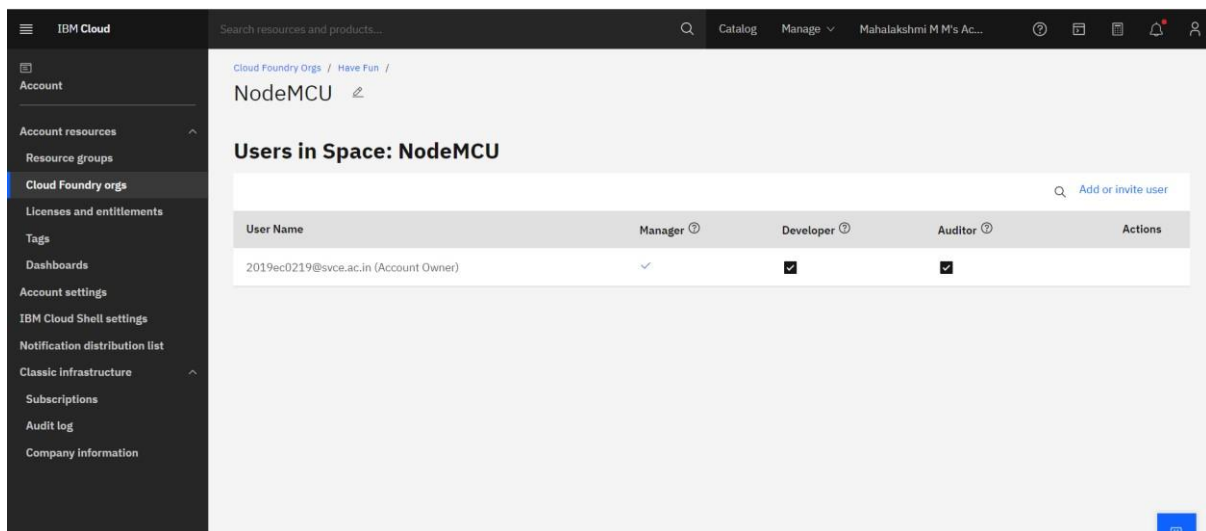


SPRINT 1

| | |
|--------------|--|
| Date | 29 October 2022 |
| Team ID | PNT2022TMID10157 |
| Project Name | Smart Farmer – IOT Enabled Smart Farming Application |

Configuration:

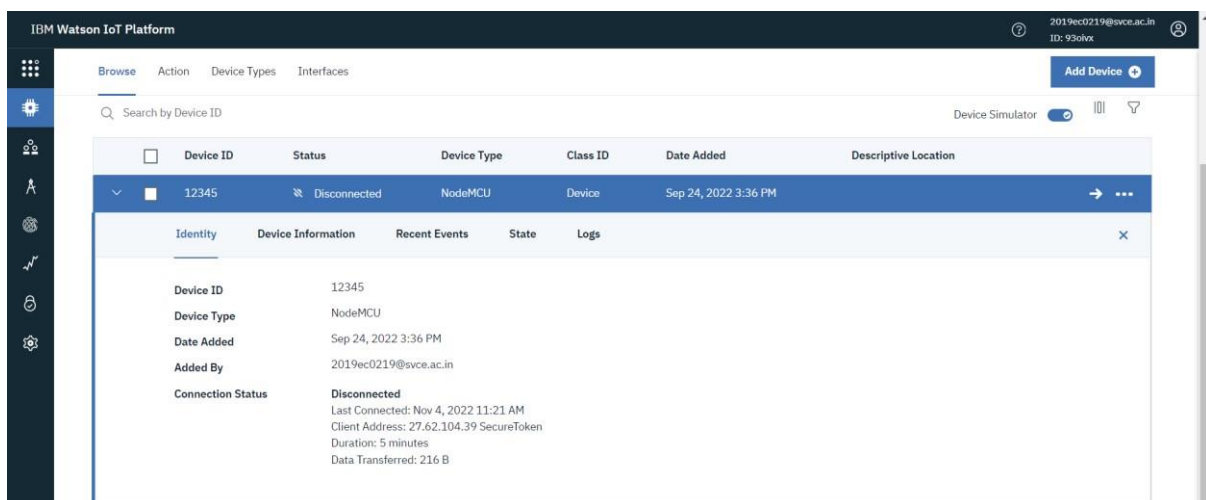
An account has been created on the required platforms (IBM Cloud, IBM Watson, NodeRed, MIT App Inventor) **IBM Cloud:**



The screenshot shows the IBM Cloud console interface. On the left is a navigation menu with options like Account, Account resources, Resource groups, Cloud Foundry orgs, Licenses and entitlements, Tags, Dashboards, Account settings, IBM Cloud Shell settings, Notification distribution list, Classic infrastructure, Subscriptions, Audit log, and Company information. The main area displays the 'Users in Space: NodeMCU' page. It includes a search bar with 'Add or invite user' and a table of users.

| User Name | Manager | Developer | Auditor | Actions |
|---------------------------------------|---------|-----------|---------|---------|
| 2019ec0219@svce.ac.in (Account Owner) | ✓ | ✓ | ✓ | |

IBM Watson:



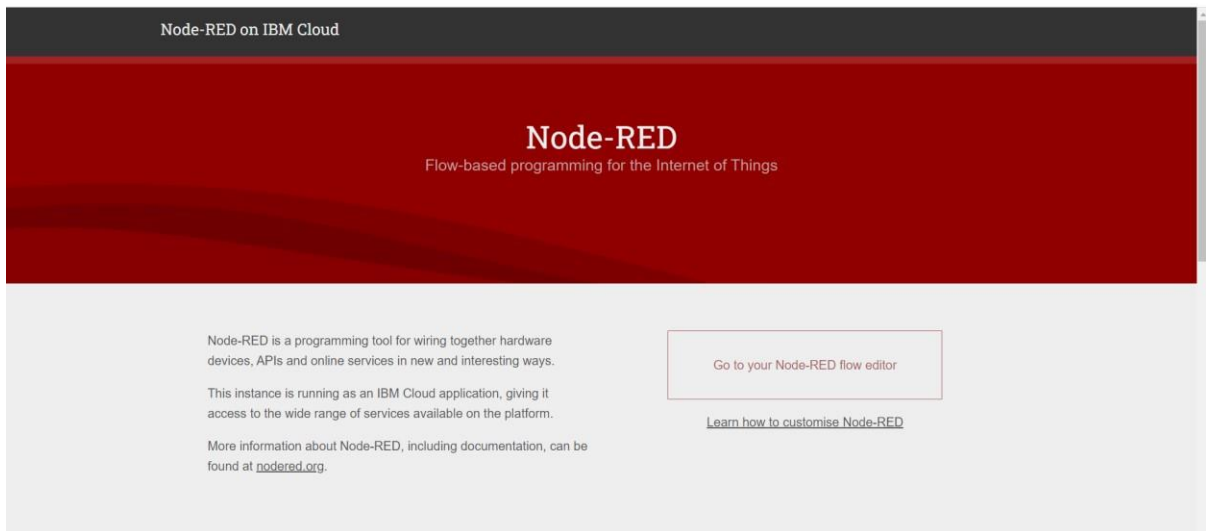
The screenshot shows the IBM Watson IoT Platform console. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. The main area displays a table of devices. The selected device is 12345, a NodeMCU, with a status of 'Disconnected'.

| Device ID | Status | Device Type | Class ID | Date Added | Descriptive Location |
|-----------|--------------|-------------|----------|----------------------|----------------------|
| 12345 | Disconnected | NodeMCU | Device | Sep 24, 2022 3:36 PM | |

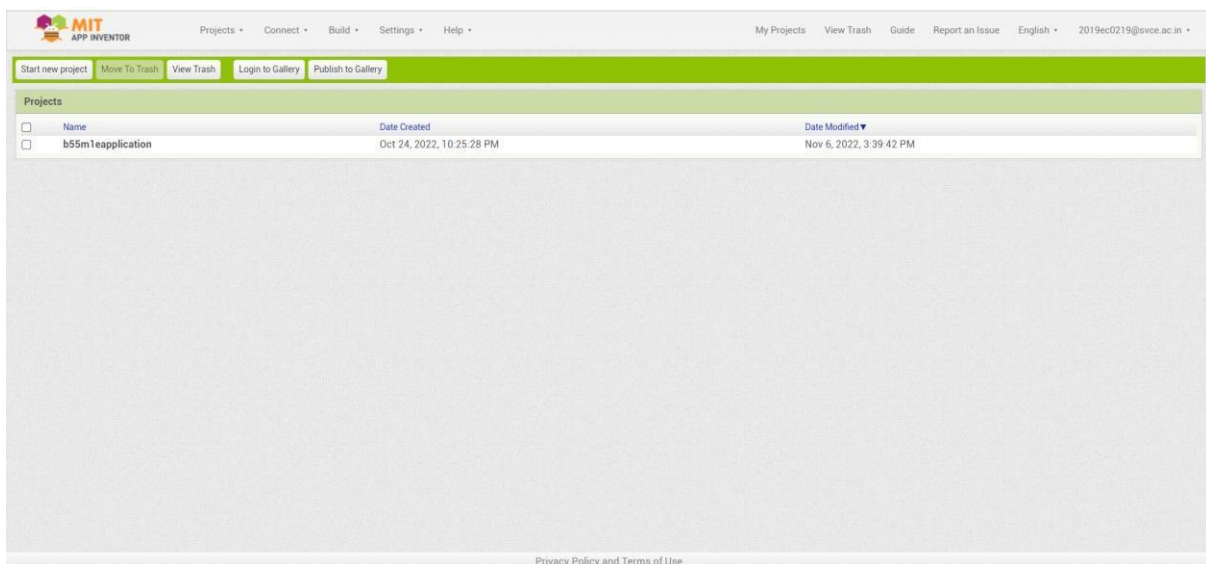
Below the table, there is a detailed view of the selected device, showing its identity, device information, recent events, state, and logs.

| Identity | Device Information | Recent Events | State | Logs |
|-------------------|--|---------------|-------|------|
| Device ID | 12345 | | | |
| Device Type | NodeMCU | | | |
| Date Added | Sep 24, 2022 3:36 PM | | | |
| Added By | 2019ec0219@svce.ac.in | | | |
| Connection Status | Disconnected | | | |
| | Last Connected: Nov 4, 2022 11:21 AM | | | |
| | Client Address: 27.62.104.39 SecureToken | | | |
| | Duration: 5 minutes | | | |
| | Data Transferred: 216 B | | | |

Node-Red:



MIT App Inventor:



Simulation:

Program:

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#define DHTPIN 4
#define DHTTYPE DHT22

DHT_Unified dht(DHTPIN, DHTTYPE);
uint32_t
delayMS; void
setup() {
  Serial.begin(9600
```

```

0); //
Initialize
device.
    dht.begin();
    Serial.println(F("DHTxx Unified Sensor Example"));
sensor_t sensor;
    dht.temperature().getSensor(&sensor);
    Serial.println(F("-----"));
    Serial.println(F("Temperature Sensor"));
    Serial.print(sensor.resolution);
    Serial.println(F("°C"));
    Serial.println(F("-----"));
dht.humidity().getSensor(&sensor);
Serial.println(F("Humidity Sensor"));
    Serial.print(sensor.resolution);
    Serial.println(F("%"));
    Serial.println(F("-----"));
delayMS = sensor.min_delay / 1000;
} void loop() {    delay(delayMS);
sensors_event_t event;
dht.temperature().getEvent(&event);
if (isnan(event.temperature)) {
    Serial.println(F("Error reading temperature!"));
}
else {
    Serial.print(F("Temperature: "));
    Serial.print(event.temperature);
    Serial.println(F("°C"));
}    dht.humidity().getEvent(&event);
if (isnan(event.relative_humidity)) {
    Serial.println(F("Error reading humidity!"));
}
else {
    Serial.print(F("Humidity: "));
    Serial.print(event.relative_humidity);
    Serial.println(F("%"));
}
}

```

OUTPUT:

WOKWI

SAVE

SHARE

Docs

M

sketch.ino

diagram.json

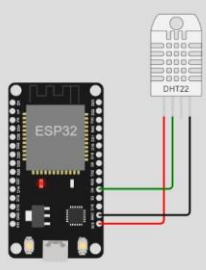
libraries.txt

Library Manager

```
1 #include <Adafruit_Sensor.h>
2 #include <DHT.h>
3 #include <DHT_U.h>
4
5 #define DHTPIN 4
6 #define DHTTYPE DHT22
7
8 DHT_Unified dht(DHTPIN, DHTTYPE);
9
10 uint32_t delayMS;
11
12 void setup() {
13   Serial.begin(9600);
14   // Initialize device.
15   dht.begin();
16   Serial.println(F("DHTxx Unified Sensor Example"));
17   sensor_t sensor;
18   dht.temperature().getSensor(&sensor);
19   Serial.println(F("-----"));
20   Serial.println(F("Temperature Sensor"));
21   Serial.print(sensor.resolution);
22   Serial.println(F("°C"));
23   Serial.println(F("-----"));
24   dht.humidity().getSensor(&sensor);
25   Serial.println(F("Humidity Sensor"));
26   Serial.print(sensor.resolution);
27   Serial.println(F("%"));
28   Serial.println(F("-----"));
29   delayMS = sensor.min_delay / 1000;
30 }
31
32 void loop() {
33   ...
```

Simulation

00:26.883 65%



The diagram shows an ESP32 microcontroller board connected to a DHT22 digital temperature and humidity sensor. The sensor is connected to the ESP32 via three wires: a green wire to the VCC pin, a red wire to the GND pin, and a white wire to the data pin.

Humidity: 40.00%
Temperature: 24.00°C
Humidity: 40.00%
Temperature: 24.00°C
Humidity: 40.00%
Temperature: 24.00°C
Humidity: 40.00%