

## SPRINT 3 - MIT APP INVENTOR

Team ID	PNT2022TMID43471
Project Name	Smart Former IOT Based smart Farming Application
Max Mark	2 Mark

To make the user interact with software :

Link: <http://ai2.appinventor.mit.edu/#5873177709117440>

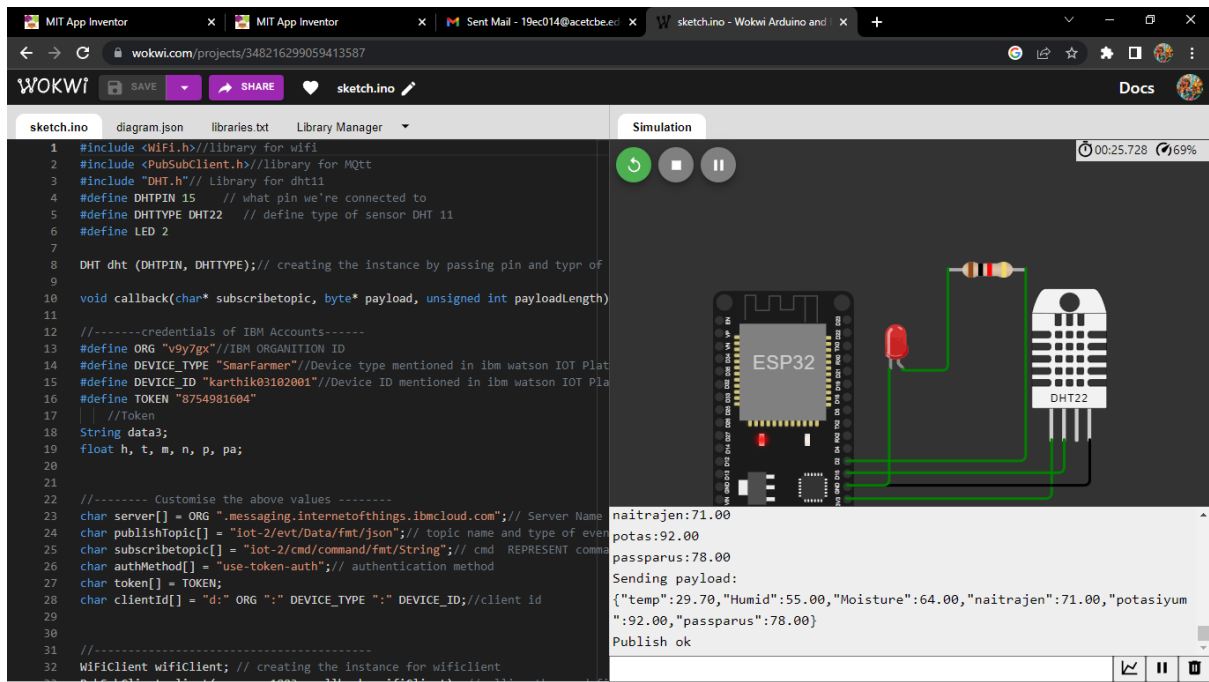
The screenshot displays the MIT App Inventor web interface with a Wokwi simulation running. The left pane shows the 'sketch.ino' file with the following code:

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #include "DHT.h" // Library for dht11
4 #define DHTPIN 15 // what pin we're connected to
5 #define DHTTYPE DHT22 // define type of sensor DHT 11
6 #define LED 2
7
8 DHT dht (DHTPIN, DHTTYPE); // creating the instance by passing pin and type of
9
10 void callback(char* topic, byte* payload, unsigned int payloadLength)
11
12 //-----credentials of IBM Accounts-----
13 #define ORG "v9y7gx" // IBM ORGANIZATION ID
14 #define DEVICE_TYPE "SmarFarmer" // Device type mentioned in IBM Watson IoT Platform
15 #define DEVICE_ID "karthik03102001" // Device ID mentioned in IBM Watson IoT Platform
16 #define TOKEN "8754981604"
17 //Token
18 String data3;
19 float h, t, m, n, p, pa;
20
21 //----- Customise the above values -----
22 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
23 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event
24 char subscribTopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command
25 char authMethod[] = "use-token-auth"; // authentication method
26 char token[] = TOKEN;
27 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; // client id
28
29 //-----
30
31 //-----
32 WiFiClient wificlient; // creating the instance for wificlient
```

The right pane shows the 'Simulation' tab with a visual representation of the hardware. An ESP32 microcontroller is connected to a DHT22 temperature and humidity sensor via I2C. A red LED is connected to the ESP32's GPIO pins. The simulation is running, as indicated by the 'Restart the simulation' button and the '00:19.929 87%' status.

The output console at the bottom right shows the following data:

```
naitrajen:41.00
potas:84.00
passparus:47.00
Sending payload:
{"temp":29.70,"Humid":55.00,"Moisture":43.00,"naitrajen":41.00,"potasium":84.00,"passparus":47.00}
Publish ok
```



( Note : Here we can use LED light instead of motor because there is no motor component in the simulation app )

10:29 AM

11.6KB/s



83

## Smart farming

tempreture

29.7

humidity

55

moisture

69

nitrogen

70

potassium

91

phosphorus

63

**Motor ON**

**Motor OFF**

