PROJECT-BASED EXPERIENTIAL LEARNING

PROGRAM (NALAIYA THIRAN)

Project Report

Project Name:	IoT Based Smart Crop Protection System for Agriculture
Team ID:	PNT2022TMID38376
A.Arul Prakasam(Team Leader)	412719106003
J.R.Balaji(Team Member)	412719106004
R.Ragul(Team Member)	412719106013
N.Sriram(Team Member)	412719106301
N.R.Vaitheshwaran(Team Member)	412719106015

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

TAGORE ENGINEERING COLLEGE

(Approved by AICTE, Affiliated to ANNA University)

Project overview

• The device will detect the animals and birds using the Clarifai service.

• If any animal or bird is detected the image will be captured and stored in the IBM Cloud object storage.

• It also generates an alarm and avoid animals from destroying the crop .

• The image URL will be stored in the IBM Cloudant DB service.

• The device will also monitor the soil moisture levels, temperature, and humidity values and send them to the IBM IoT Platform.

• The image will be retrieved from Object storage and displayed in the web application.

• A web application is developed to visualize the soil moisture, temperature, and humidity values .

• Users can also control the motors through web application.

LITERATURE SURVEY

Digital agriculture, sometimes known as smart farming or e-agriculture is tools that digitally collect, store, analyze, and share electronic data and information in agriculture. Smart agriculture is a broad term that collects agriculture and food production practices powered by Internet of Things, big data and advanced analytics technology. When we talk about IoT, we generally refer to adding sensing, automation and analytics technology to modern agricultural processes. By using this methods we get more efficient, more quality, less resources consumption than compared to regular agriculture

Book/journal	Author's name	explanation
ICT for Agriculture and Environment,CITAMA 2019	Tanya Recalde, Karina Real-Aviles, Cesar Moran, Paola Grijalva, Raquel Gomez chabla	The objective of this paper is to offer an overview of the IoT applications in agriculture through topics such IoT-based software applications for agriculture available in the market, IoT-based devices used in the agriculture, as well as the benefits provided by this kind of technologies.
Internet of Things (IoT)-Based Wireless Health: Enabling Technologies and Applications	Yousaf Bin Zikria, Tariq Umer, Adnan Abid, Shamyla Riaz, Muhammad Shoaib Farooq	The objective of this paper is the collection of all relevant research on IoT agricultural applications, sensors/devices, communication protocols, and network types. Furthermore, it also discusses the main issues and challenges that are being investigated in the field of agriculture

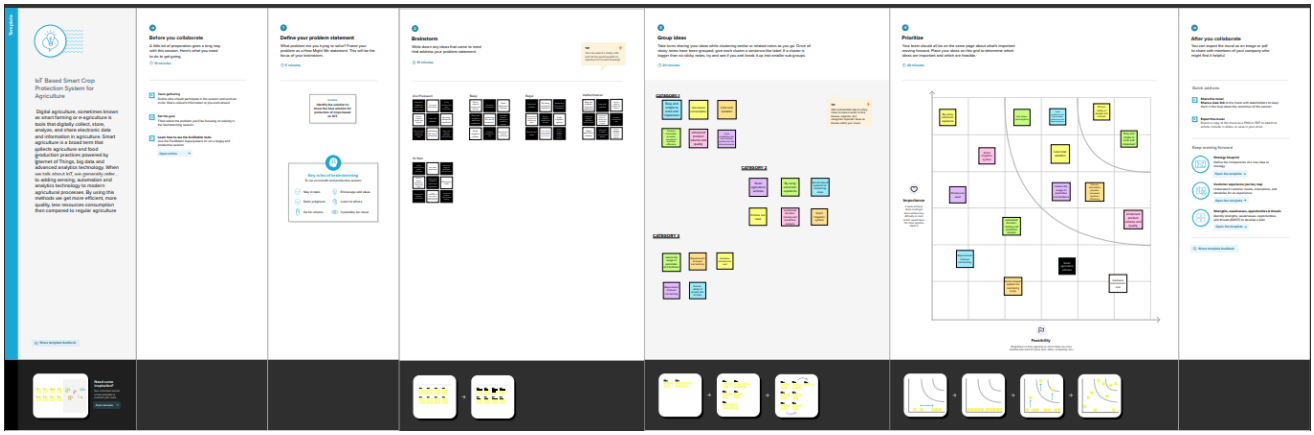
Problem Statement Definition

Most of the farmers are facing many problems nowadays due to many reasons. Our problem to solve is the invasion of various species such as birds and animals that harm the crops that are being cultivated. Various types of species such as birds and animals come to the cultivation field according to the crop that is being cultivated and also according to the season of cultivation. Some wild animals enter the field during night times when the field is near a forest region or when the farm cultivates some fruits and other crops that attract animals

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A farmer	Protect crops	Is difficult	The field is destroyed by animals	Angry
PS-2	A farmer	Protect crops	Is not convenience	Need to be at field at any time	Frustrated
PS- 3	A farmer	Protect safety of animals and others	Is difficult	Of placing of fences to protect crops	Sympathy
PS- 4	A farmer	Protect crops	Is difficult	Less amount of crops is harvested	Angry

Empathy Map Canvas

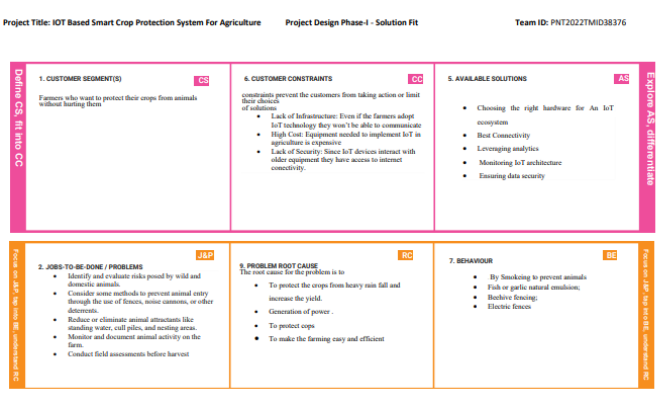
Brainstorming:



Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Agriculture was the key development in the rise of sedentary human civilization, whereby farming of domesticated species created food surpluses that enabled people to live in cities. But there are many problems occuring during agriculture one of them is animals most of animals near forest and villages depent on farm crops during summer and winter to solve this proble smart agiculture is adopted nowdays. Smart agriculture is a broad term that collects ag and food production practices powered by Internet of Things, big data and advanced analytics technology. When we talk about IoT, we generally refer to adding sensing, automation and analytics technology to modern agricultural processes.
2.	Idea / Solution description	Smart agriculture is one of the best approaches to increase production for competing with the increasing population of our country. Smart Agriculture helps to increase the production of different crops by transforming or reorienting agricultural systems. Since it is also known as precision agriculture which helps to maintain the production of food using minimal resources such as fertilizer, water and seeds irrespective of climate change to protect crops from harm.
3.	Novelty / Uniqueness	With the presence of high-quality sensors in the system which supports Real-time monitoring of agricultural systems even in isolated locations which in turn in controlling or monitoring the use of resources in large proportions, also helps in reducing impacts on the environmental system.
4.	Social Impact / Customer Satisfaction	The result from this smart farming process is – high precision and 24/7 control, eventually leading to considerable savings in all key resources used – water, energy, fertilizers, time spent by strategic people, time spent by lower-qualification human resources. Customers using Smart agriculture solution on their farms can save up to 50% in energy consumed. They also report up to 40% increase in crop yield, while reducing the cost of fertilization and chemical treatment, and up to 60% less losses resulting from animals.
5.	Business Model (Revenue Model)	Many farmers have already begun implementing this technique of smart agriculture for improving their efficiency. Sensors installed in the fields can be used to obtain detailed information for the presence of acidity and rapid increase in the temperature of the soil. Farmers with the use of sensors can also get access to know weather patterns or climatic forecasts remotely. Nowadays many farmers are using this technique of smart agriculture in their lands to avoid over-irrigation and to maintain soil erosion to a large extent. Overall the system is cheaper, easy to use, and efficient than other automation system.

Solution Fit:



Data flow and Solution Architecture:

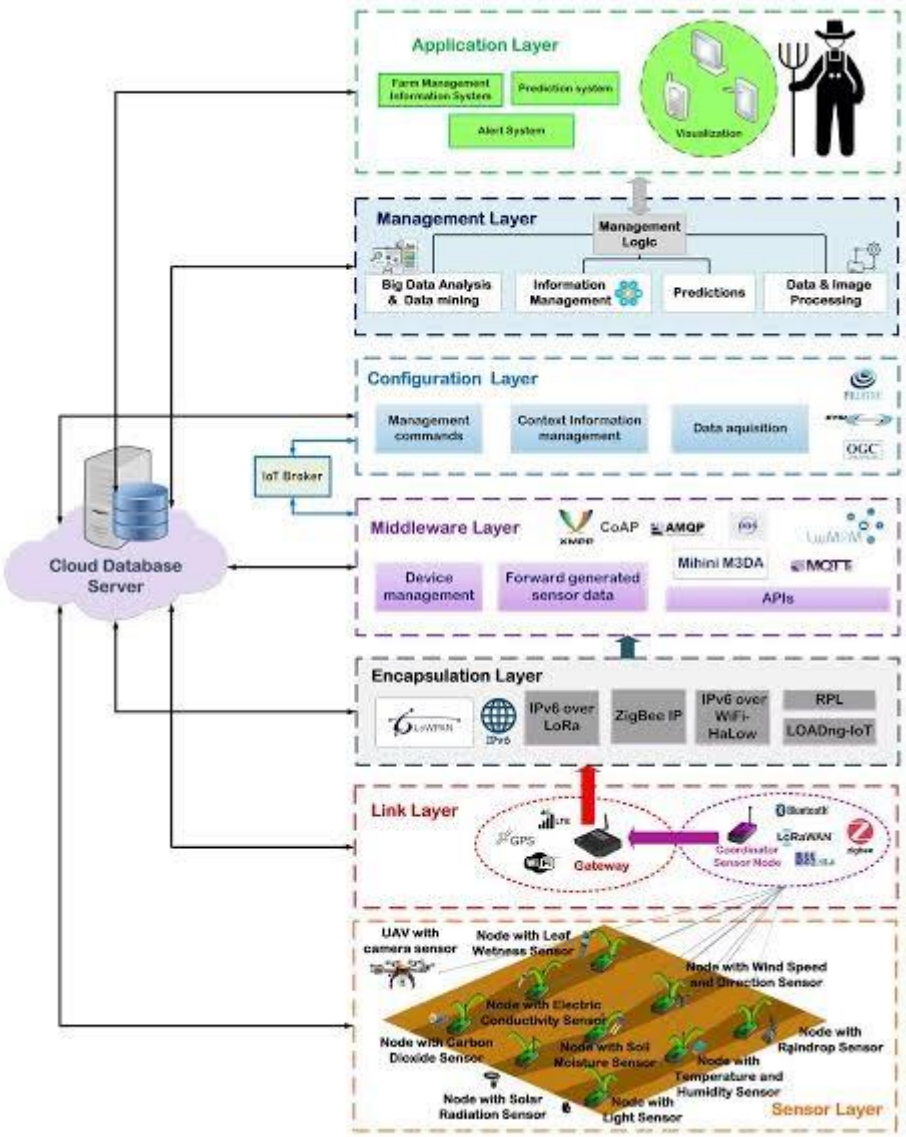
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to :

• Find the best tech solution to solve existing business problems.

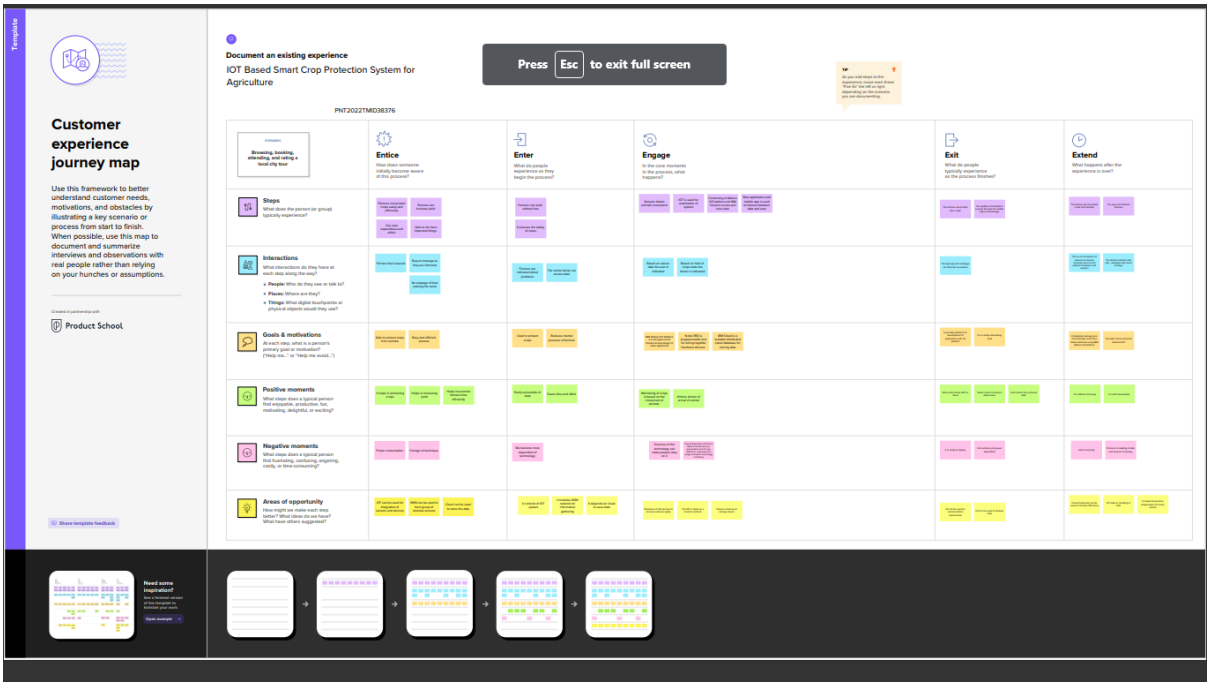
• Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.

• Define features, development phases, and solution requirements.

• Provide specifications according to which the solution is defined, managed, and delivered.



Customer Journey:



Sprint Planning:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Balaji
Sprint-1	Registration	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Ragul
Sprint-2	Registration	USN-3	As a user, I can register for the application through Facebook	2	Low	Sriram
Sprint-1	Registration	USN-4	As a user, I can register for the application through Gmail	2	Medium	Vaitheshwaran
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Arul Prakasam
Sprint-1	Dashboard	USN-6	As a user, I can log into the application by entering my email & password and access all the resources and services available	2	High	Arul Prakasam
Sprint-1	Login	USN-6	As a data controller, I log into my profile and start monitoring the data updates	3	High	Ragul
Sprint-1	Dashboard	USN-6	I receive all the information about farm from sensor. Whenever there is change in data, corresponding updates are given to customers.	2	Medium	Balaji
Sprint-1	Login	USN-6	As a data controller, I keep note of all the	3	High	

Technology Stack

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Mobile App	MIT App Inventor
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
6.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
7.	External API-1	Purpose of External API used in the application	Aadhar API, user ID.
8.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration:IBM Cloud Cloud Server Configuration : IBM Cloud	Local, IBM cloud..

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	IBM cloud framework is used	IBM cloud , Red node, MIT app invertor
2.	Security Implementations	Security to access controls implemented and use of firewalls is used.	Encryptions, Dyscryptions, OTP password.
3.	Scalable Architecture	In future it can be linked to form a smart agriculture system without the need of farmers supervise.	IOT, Cloud, RED Node
4.	Availability	Application is available 24/7 as it is hosted on IBM Cloud and can be used anywhere.	Technology used
5.	Performance	Customers will have a smooth experience while using the application, as it is simple and well optimized	Technology used

Python Code:

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

#IBM Watson Device Credentials...
organization = "pcig8v"
deviceType = "arul"
deviceId = "arul_1"
authMethod = "token"
authToken = "123456789"

def myCommandCallback(cmd):
 print("Command received: %s" % cmd.data['command'])
 status=cmd.data['command']
 if status=="sprinkler_on":
 print ("sprinkler is turning ON")
 else :
 print ("sprinkler is turning OFF")

try:
 deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-
method":
authMethod, "auth-token": authToken}
 deviceCli = ibmiotf.device.Client(deviceOptions)

except Exception as e:
 print("Exception detected in connecting device: %s" % str(e))
```

```
        sys.exit()


        #Connecting to IBM watson...
        deviceCli.connect()


        while True:
         #Getting values from sensors...
         temp_sensor = round( random.uniform(0,80),2)
         PH_sensor = round(random.uniform(1,14),3)
         camera = ["Detected","Not Detected","Not Detected","Not Detected","Not
Detected","Not
        Detected",]
         camera_reading = random.choice(camera)
         flame = ["Detected","Not Detected","Not Detected","Not Detected","Not
Detected","Not
        Detected",]
         flame_reading = random.choice(flame)
         moist_level = round(random.uniform(0,100),2)
         water_level = round(random.uniform(0,30),2)


         #storing the sensor data to send in json format to cloud.
         temp_data = { 'Temp' : temp_sensor }
         PH_data = { 'PH value' : PH_sensor }
         camera_data = { 'Animal attack' : camera_reading}
         flame_data = { 'Flame' : flame_reading }
         moist_data = { 'Moisture level' : moist_level}
         water_data = { 'Water level' : water_level}


         # publishing Sensor datas to IBM Watson for every 5-10 seconds.
         success = deviceCli.publishEvent("Temperature sensor", "json", temp_data, qos=0)
```

```
sleep(1)

if success:
print ("... ...publish ok... ... ...")
print ("Published Temp = %s C" % temp_sensor, "to IBM Watson")
success = deviceCli.publishEvent("PH sensor", "json", PH_data, qos=0)
sleep(1)

if success:
print ("Published PH value = %s" % PH_sensor, "to IBM Watson")
success = deviceCli.publishEvent("camera", "json", camera_data, qos=0)
sleep(1)

if success:
print ("Published Animal attack %s " % camera_reading, "to IBM Watson")
success = deviceCli.publishEvent("Flame sensor", "json", flame_data, qos=0)
sleep(1)

if success:
print ("Published Flame %s " % flame_reading, "to IBM Watson")
success = deviceCli.publishEvent("Moisture sensor", "json", moist_data, qos=0)
sleep(1)

if success:
print ("Published Moisture level = %s " % moist_level, "to IBM Watson")
success = deviceCli.publishEvent("Water sensor", "json", water_data, qos=0)
sleep(1)

if success:
```

```
        print ("Published Water level = %s cm" % water_level, "to IBM Watson")
        print ("")

        #Automation to control sprinklers by present temperature an to send alert
message to IBM
        Watson.

        if (temp_sensor > 35):
        print("sprinkler-1 is ON")
        success = deviceCli.publishEvent("Alert1", "json",{ 'alert1' : "Temperature(%s) is
high,
        sprinkerlers are turned ON" %temp_sensor }, qos=0)
        sleep(1)
        if success:
        print( 'Published Alert1 : ', "Temperature(%s) is high, sprinkerlers are turned ON"
        %temp_sensor,"to IBM Watson")
        print("")
        else:
        print("sprinkler-1 is OFF")
        print("")

        #To send alert message if farmer uses the unsafe fertilizer to crops.

        if (PH_sensor > 7.5 or PH_sensor < 5.5):
        success = deviceCli.publishEvent("Alert2", "json",{ 'alert2' : "Fertilizer PH level(%s)
is not
        safe,use other fertilizer" %PH_sensor } , qos=0)
        sleep(1)

        if success:
```

```
print('Published Alert2 : ' , "Fertilizer PH level(%s) is not safe,use other fertilizer"
%PH_sensor,"to IBM Watson")
print("")

#To send alert message to farmer that animal attack on crops.

if (camera_reading == "Detected"):
success = deviceCli.publishEvent("Alert3", "json", { 'alert3' : "Animal attack on
crops detected" },
qos=0)
sleep(1)
if success:
print('Published Alert3 : ' , "Animal attack on crops detected","to IBM
Watson","to IBM
Watson")
print("")

#To send alert message if flame detected on crop land and turn ON the splinkers
to take
immediate action.

if (flame_reading == "Detected"):
print("sprinkler-2 is ON")
success = deviceCli.publishEvent("Alert4", "json", { 'alert4' : "Flame is detected
crops are in
danger,sprinklers turned ON" }, qos=0)
sleep(1)

if success:
print( 'Published Alert4 : ' , "Flame is detected crops are in danger,sprinklers
turned ON","to IBM
```

```
Watson")
print("")

else:
print("sprinkler-2 is OFF")
print("")

#To send alert message if Moisture level is LOW and to Turn ON Motor-1 for irrigation.

if (moist_level < 20):
print("Motor-1 is ON")
success = deviceCli.publishEvent("Alert5", "json", { 'alert5' : "Moisture level(%s) is low, Irrigation
started" %moist_level }, qos=0)
sleep(1)

if success:
print('Published Alert5 : ' , "Moisture level(%s) is low, Irrigation started" %moist_level,"to IBM
Watson" )
print("")
else:
print("Motor-1 is OFF")
print("")

#To send alert message if Water level is HIGH and to Turn ON Motor-2 to take water out.

if (water_level > 20):
print("Motor-2 is turning ON")
```

```
success = deviceCli.publishEvent("Alert6", "json", { 'alert6' : "Water level(%s) is high, so motor is
ON to take water out " %water_level }, qos=0)
sleep(1)

if success:
print('Published Alert6 : ' , "water level(%s) is high, so motor is ON to take water out "
%water_level,"to IBM Watson" )
print("")
else:
print("Motor-2 is turning OFF")
print("")

#command recived by farmer
deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

Node-Red code:

```
msg.payload=msg.payload.”temp”
return msg;
```

HUMIDITY:

```
msg.payload=msg.payload.”Humid”
return msg;
```

MOISTURE:

```
msg.payload=msg.payload.”Moist”
return msg;
```

ANIMAL DETECTION:

msg.payload=msg.payload.”Animal_dect”

return msg;

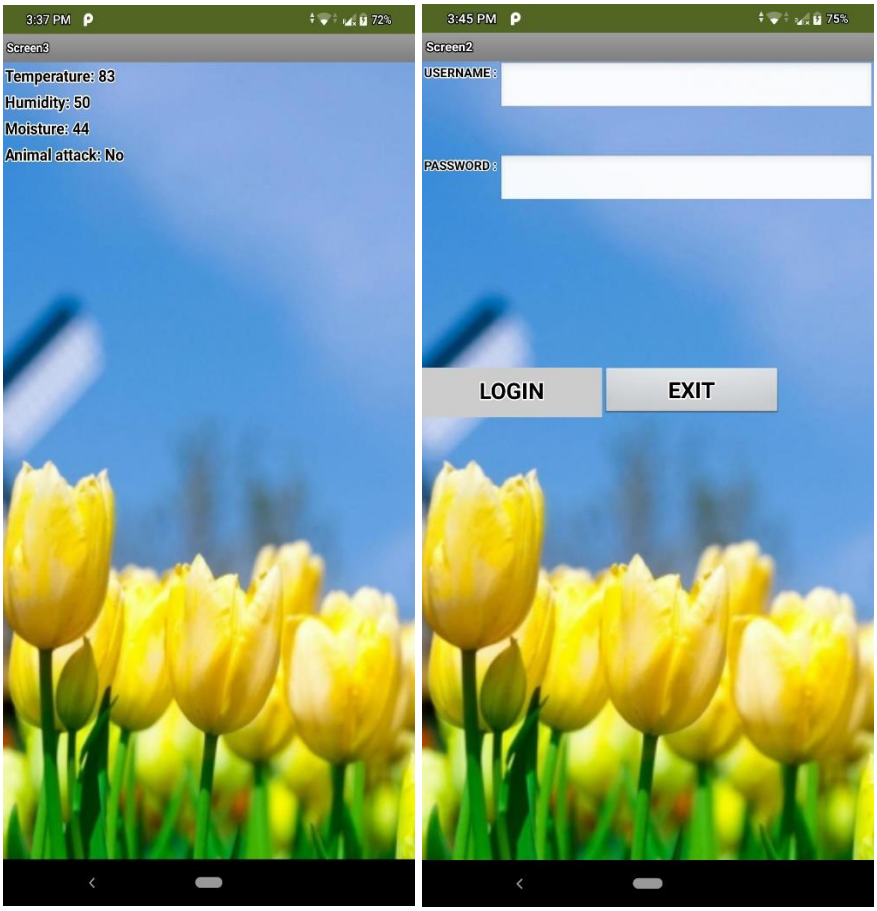
Wokwi :

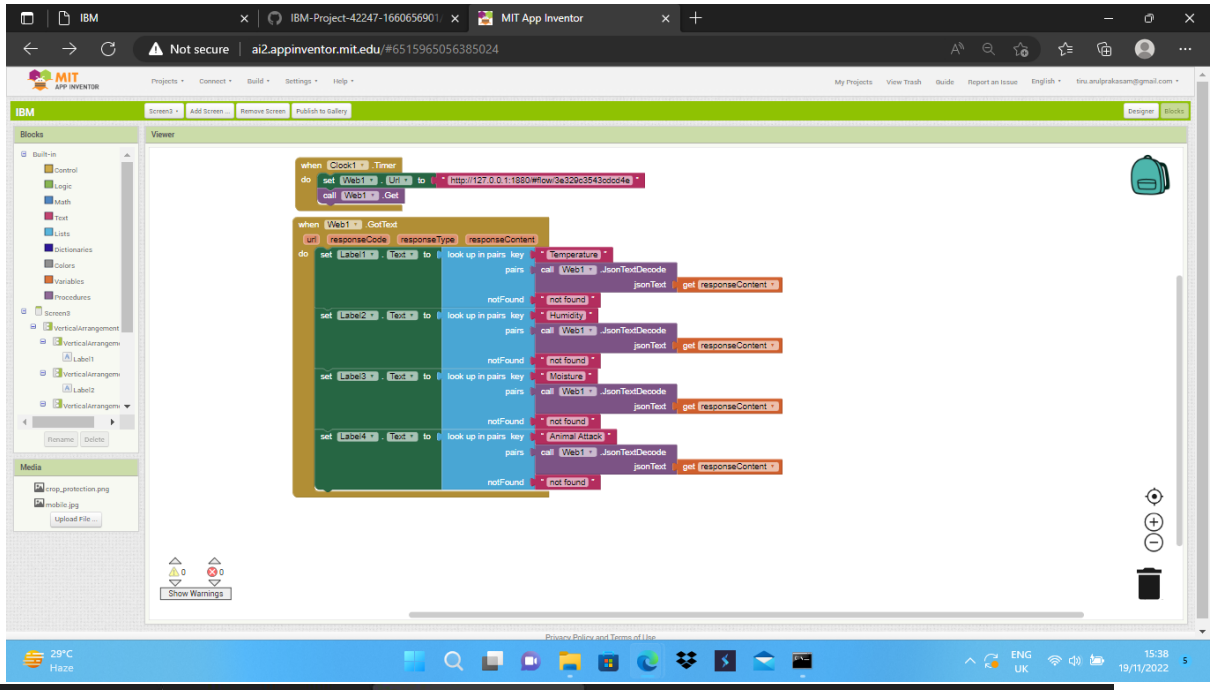
```
include <WiFi.h>
#include <PubSubClient.h>

WiFiClient wifiClient;

#define ORG "pcig8v"
#define DEVICE_TYPE "Arul"
#define DEVICE_ID "2002"
#define TOKEN "123456789"
#define speed 0.034

char server[] = ORG".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/raspberrypi_1/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
const int trigpin=5;
const int echopin=18;
String command;
String data="";
long duration;
float dist;
void setup()
{
Serial.begin(115200);
pinMode(trigpin, OUTPUT);
pinMode(echopin, INPUT);
wifiConnect();
mqttConnect();
}
void loop() {
publishData();
delay(500);
if (!client.loop()) {
mqttConnect();
}
}
void wifiConnect() {
Serial.print("Connecting to "); Serial.print("Wifi");
WiFi.begin("Wokwi-GUEST", "", 6);
```

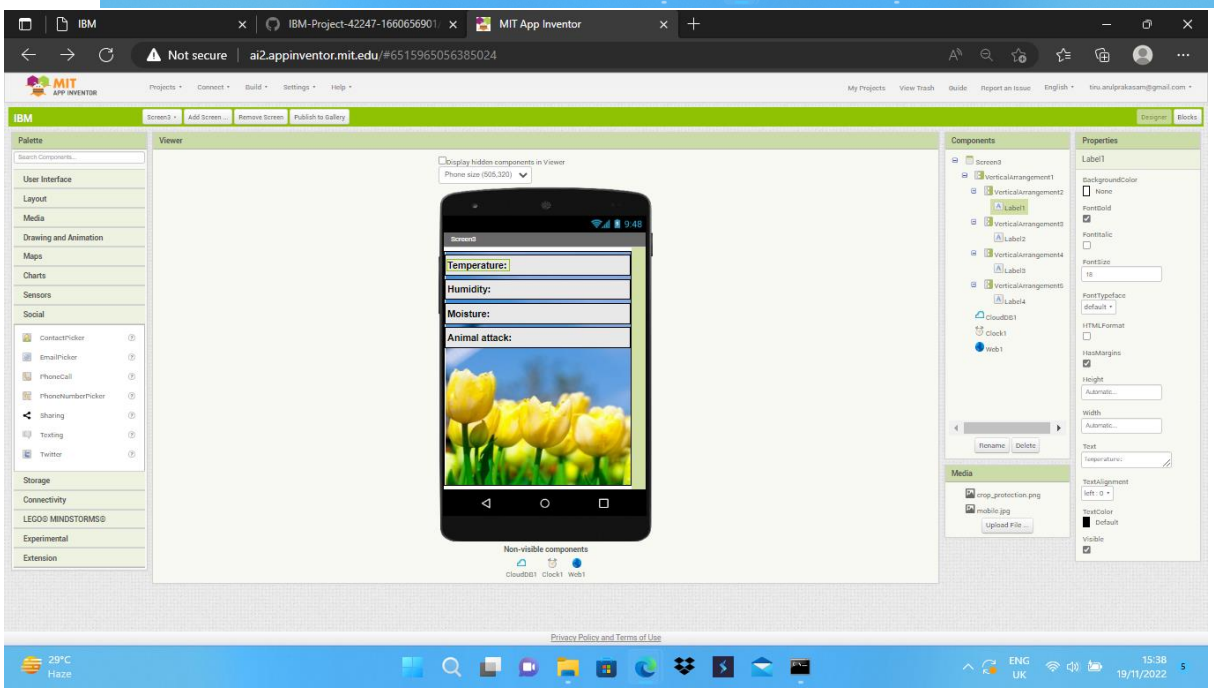
```
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP()); }
void mqttConnect() {
if (!client.connected()) {
Serial.print("Reconnecting MQTT client to ");
Serial.println(server);
while (!client.connect(clientId, authMethod, token)) {
Serial.print(".");
delay(500);
}
initManagedDevice();
Serial.println();
}
}
void initManagedDevice() {
if (client.subscribe(topic)) {
// Serial.println(client.subscribe(topic));
Serial.println("subscribe to cmd OK");
}
else {
Serial.println("subscribe to cmd FAILED");
}
}
void publishData()
{
digitalWrite(trigpin,LOW);
digitalWrite(trigpin,HIGH);
delayMicroseconds(10);
digitalWrite(trigpin,LOW);
duration=pulseIn(echopin,HIGH);
dist=duration*speed/2;
if(dist<100){
String payload = "{\"Alert distance\":";
payload += dist;
payload += "}";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str()))
{ Serial.println("Publish OK");
} else {
Serial.println("Publish FAILED");
}
}
```

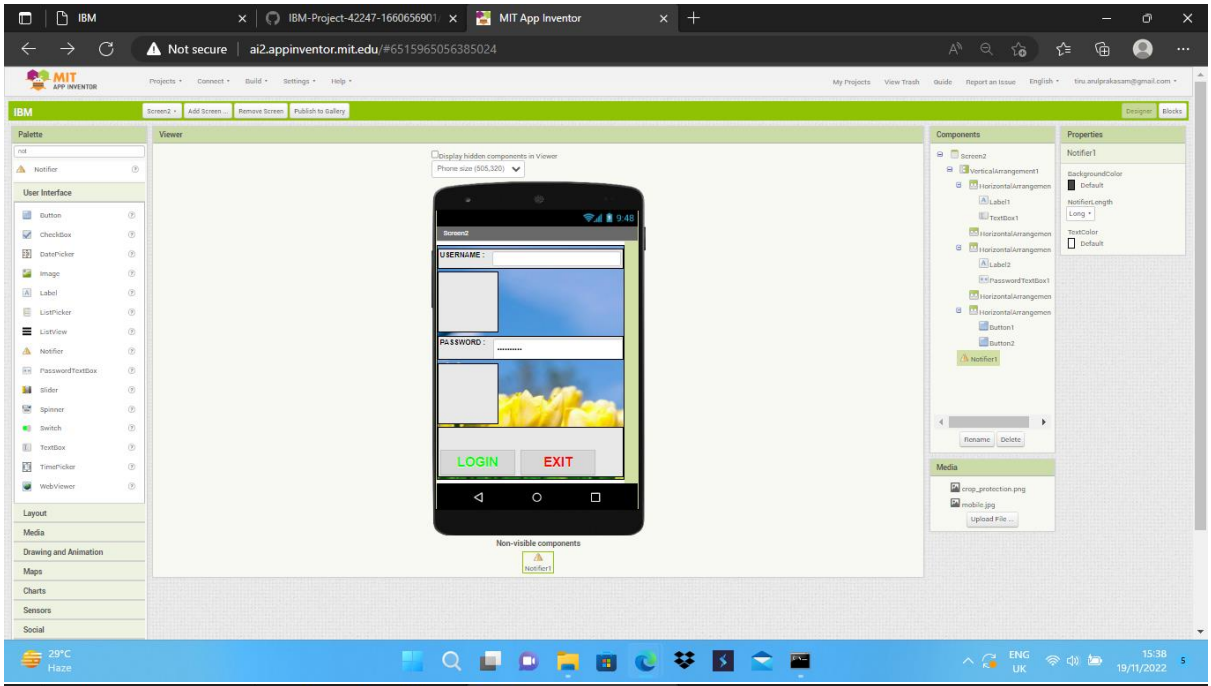
```
}
```

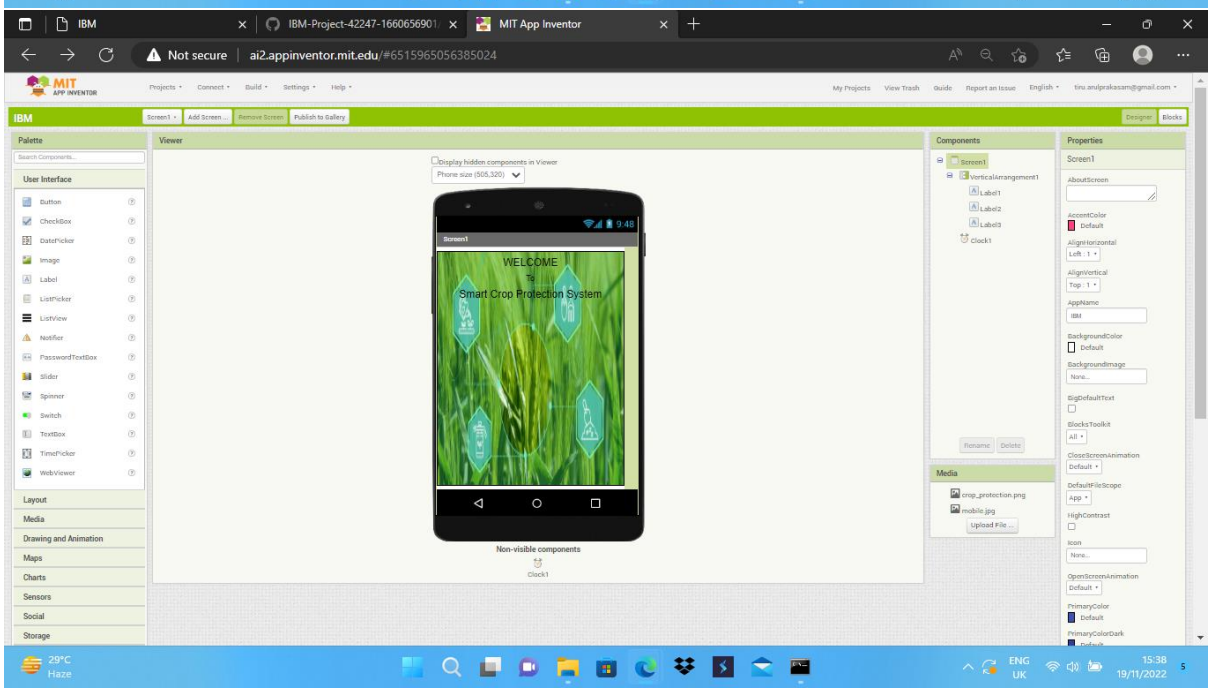
Result:

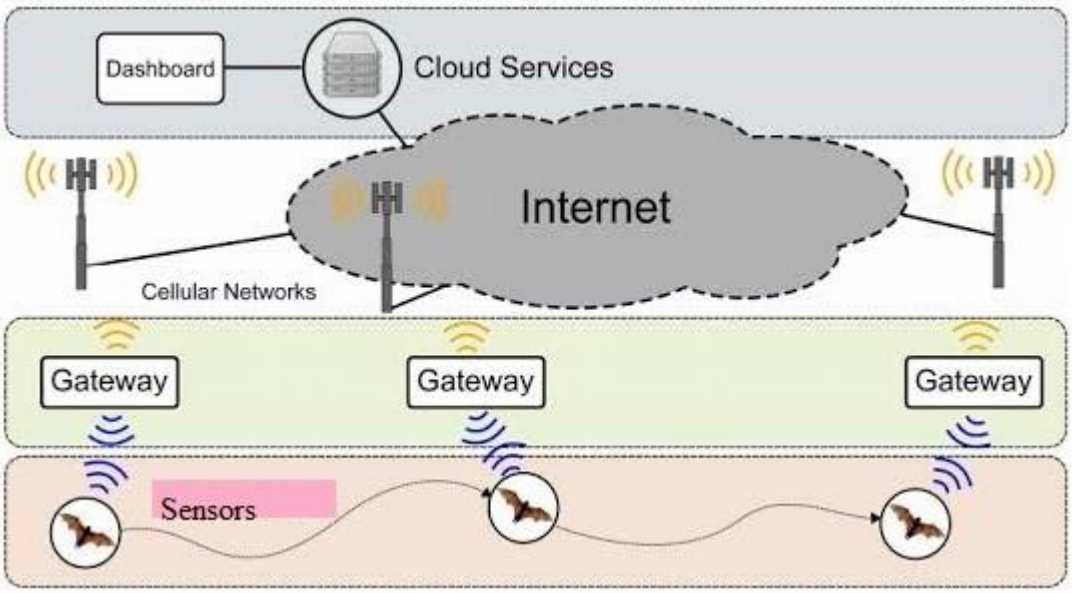

3:37 PM
72%
Screen3
Temperature: 83
Humidity: 50
Moisture: 44
Animal attack: No
3:45 PM
75%
Screen2
USERNAME :
PASSWORD :
LOGIN
EXIT

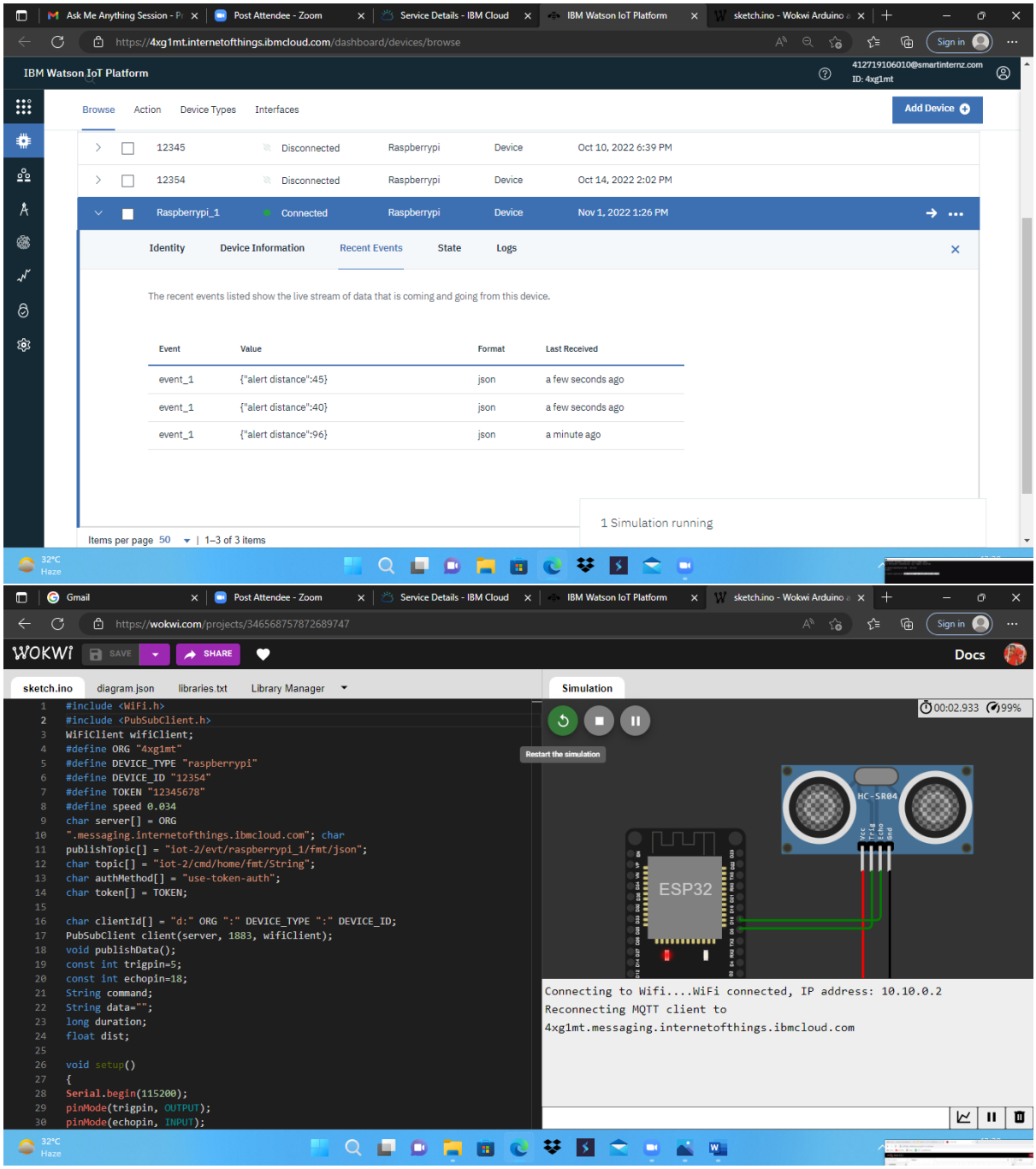

IBM
IBM-Project-42247-1660656901
MIT App Inventor
Not secure
ai2.appinventor.mit.edu/#6515965056385024
Blocks
Viewer
Media
Show Warnings
29°C
Haze
ENG
UK
15:38
19/11/2022


IBM
IBM-Project-42247-1660656901
MIT App Inventor
Not secure
ai2.appinventor.mit.edu/#6515965056385024
Palette
Viewer
Components
Properties
Temperature:
Humidity:
Moisture:
Animal attack:
Non-visible components
Privacy Policy and Terms of Use
29°C
Haze
ENG
UK
15:38
19/11/2022

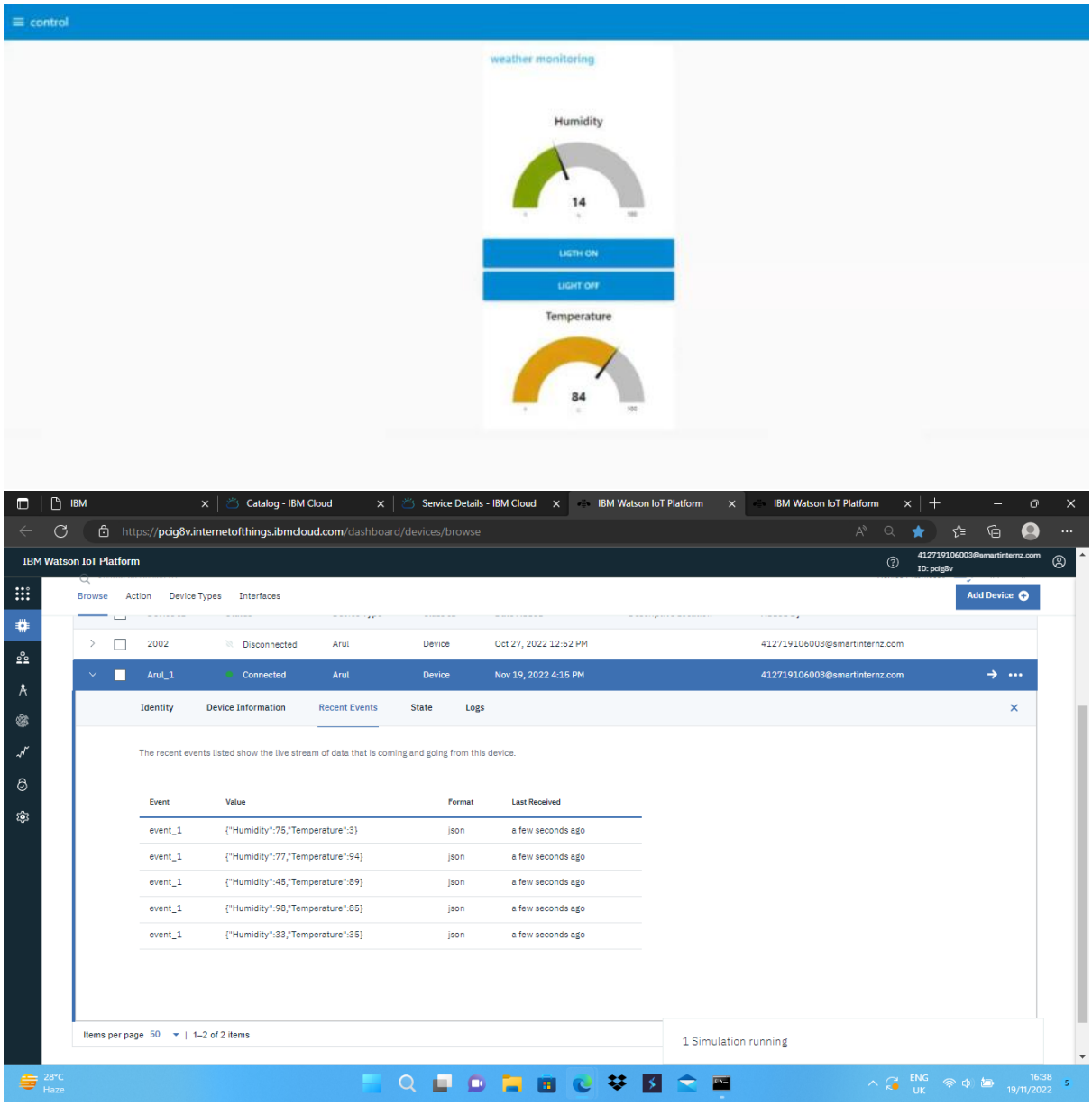

IBM
IBM-Project-42247-1660656901
MIT App Inventor
Not secure
ai2.appinventor.mit.edu/#6515965056385024
Screen2
USERNAME :
PASSWORD :
LOGIN
EXIT
Non-visible components
Notifier1


IBM
IBM-Project-42247-1660656901
MIT App Inventor
Not secure
ai2.appinventor.mit.edu/#6515965056385024
Screen1
WELCOME
To
Smart Crop Protection System
Non-visible components
Clock1


Dashboard
Cloud Services
Internet
Cellular Networks
Gateway
Gateway
Gateway
Sensors




```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
#define ORG "4xg1mt"
#define DEVICE_TYPE "raspberrypi"
#define DEVICE_ID "12354"
#define TOKEN "12345678"
#define speed 0.034
char server[] = ORG
".messaging.internetofthings.ibmcloud.com"; char
publishTopic[] = "iot-2/evt/raspberrypi_1/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
const int trigpin=5;
const int echopin=18;
String command;
String data="";
long duration;
float dist;

void setup()
{
Serial.begin(115200);
pinMode(trigpin, OUTPUT);
pinMode(echopin, INPUT);
```



APPLICATIONS:

• Monitoring the crop field with the help of sensors (light , humidity, temperature, soil moisture, etc.)

• Automating the irrigation system

• Soil Moisture Monitoring (including conductivity)

CONCLUSION:

The problem of crop vandalization by wild animals and fire has become a major social problem in current time. It requires urgent attention as no effective solution exists till date for this problem.Thus, this project carries a great social relevance as it aims to address this problem. This project will help farmers in protecting their orchards and fields and save them from significant financial losses and will save them from the unproductive efforts that they endure for the protection their fields. This will also help them in achieving better crop yields thus leading to their economic well being.

GITHUB LINK :

GitHub - IBM-EPBL/IBM-Project-42247-1660656901: IoT Based Smart Crop Protection System for Agriculture

DEMO VIDEO:

On GITHUB