PROJECT REPORT

IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

TEAM ID: PNT2022TMID03551

ROLL NUMBER	NAME
613019104031	KARPAGAM K
613019104004	ANJALI B
613019104021	DIVYA J
612019104033	KAVINA G

CONTENT

ABSTRACT

IOT Based Smart Crop-Protection for Agriculture monitoring is a system describes how to monitor crop field. It is developed by using sensors and according to the decision from a server based on sensed data, the irrigation and monitoring system is enhanced. Through wireless transmission the sensed data is forwarded to web server database. If the irrigation is automated, then the moisture and temperature fields are decreased below the potential range. The user can monitor and control the system remotely with the help of application which provides a web interface to user. By smart Agriculture monitoring system and one of the oldest ways in agriculture is the manual method of checking the parameters. In this method farmers by themselves verify all the parameter and calculate the reading. It aims at making agriculture smart using automation and IoT. The cloud computing devices are used at the end of the system that can create a whole computing system from sensors to tools that observe data from agriculture field. It proposes a novel methodology for smart farming by including a smart sensing system and smart irrigator system through wireless communication technology. This system is cheap at cost for installation. Here one can access and control the agriculture system in laptop, cell phone or a computer.

Chapter – 1

INTRODUCTION

A system using sensors that monitor different conditions of environment like humidity, temperature etc., the processor and GUI module is used. The field condition is sent to the farmer via mobile text messages. With thissystem Soil moisture, humidity and energy efficiency are managed. A system is proposed for intelligent agriculture monitoring system based on IOT technology.The main aim of this project is to help farmers to automate their farms by providing them with a Web App through which they can monitor the parameters of the field like Temperature, humidity etc. and control the equipment like watermotor and other devices remotely via internet without their actual presence in thefield.

Crops in the farms are many times devastated by the wild as well as domestic animals and low productivity of crops is one of the reasons for this. It is not possible to stay 24 hours in the farm to sentinel the crops. So to surmount this issue an automated perspicacious crop aegis system is proposed utilizing Internet of Things (IOT). The system consists of esp8266 (node MCU), soil moisture sensor, dihydrogen monoxide sensor, GPRS and GSM module, servo motor, dihydrogen monoxide pump, etc. to obtain the required output. As soon as any kineticism is detected the system will engender an alarm to be taken and the lights will glow up implemented at every corner of the farm. This will not harm any animal and the crops will stay forfended.

Chapter-2

LITERATURE SURVEY

S.No	Author and journal		Name of the topic	Features
	P. Rekha IEEE		Preventing agricultural land from animal and automated irrigation system.	❖ This system makes use of IR sensor for the detection of animals and a soil moisture sensors or to find the moisture of soil and automatically control the waterpump an for auto irrigation system.. ❖ A system by using wireless sensor networks to detect the intrusion of birds and animals in agricultural lands is discussed
2	Tejas Kharee IEEE	2019	automated crop field surveillance using computer vision	❖ In this system the long range camera are placed at the corner of field or land withconsidering maximum field of view of camera. ❖ When animal is detected by the camera the distance between camera and speaker is calculated. ❖ The speaker nearest to the animal is identified.

3	Damini kalra IEEE	2020	proposed a system for crop protection	❖ automatic irrigation system by sensing moisture, humidity and temperature of soil. ❖ The main advantage of this system is this system works in different circumstances like in night and dark (shadow).
4	M Jaya Prabha	2019	a smart crop protection system from animalsusing Arduino UNO.	❖ The system is consisting of IR sensor for animal detection, ultrasonic sensor which rotates 360 degrees for detection of birds and a GSM module to send alert message to the farmer. ❖ It is a very simple system and cannot differentiate between human and animals. Stefano Giordano et al, [5] this paper's motive is to design IOT based system to prevent animal intrusion inthe crop field and providing weather conditions

5	Alaa Adel Araby, Mai Mohamed Abd Elhameed, Hassan Mostafa	2020	Smart IoT Monitoring System for Agriculture with Predictive Analysis	❖ Precision agriculture is a new concept in agriculture, it is defined as the farm management system using information technology to identify, analyze and manage the variability of fields to ensure profitability, sustainability, and protection of the environment. ❖ It is obvious that precision agriculture increases the efficiency that can be realized by understanding and dealing

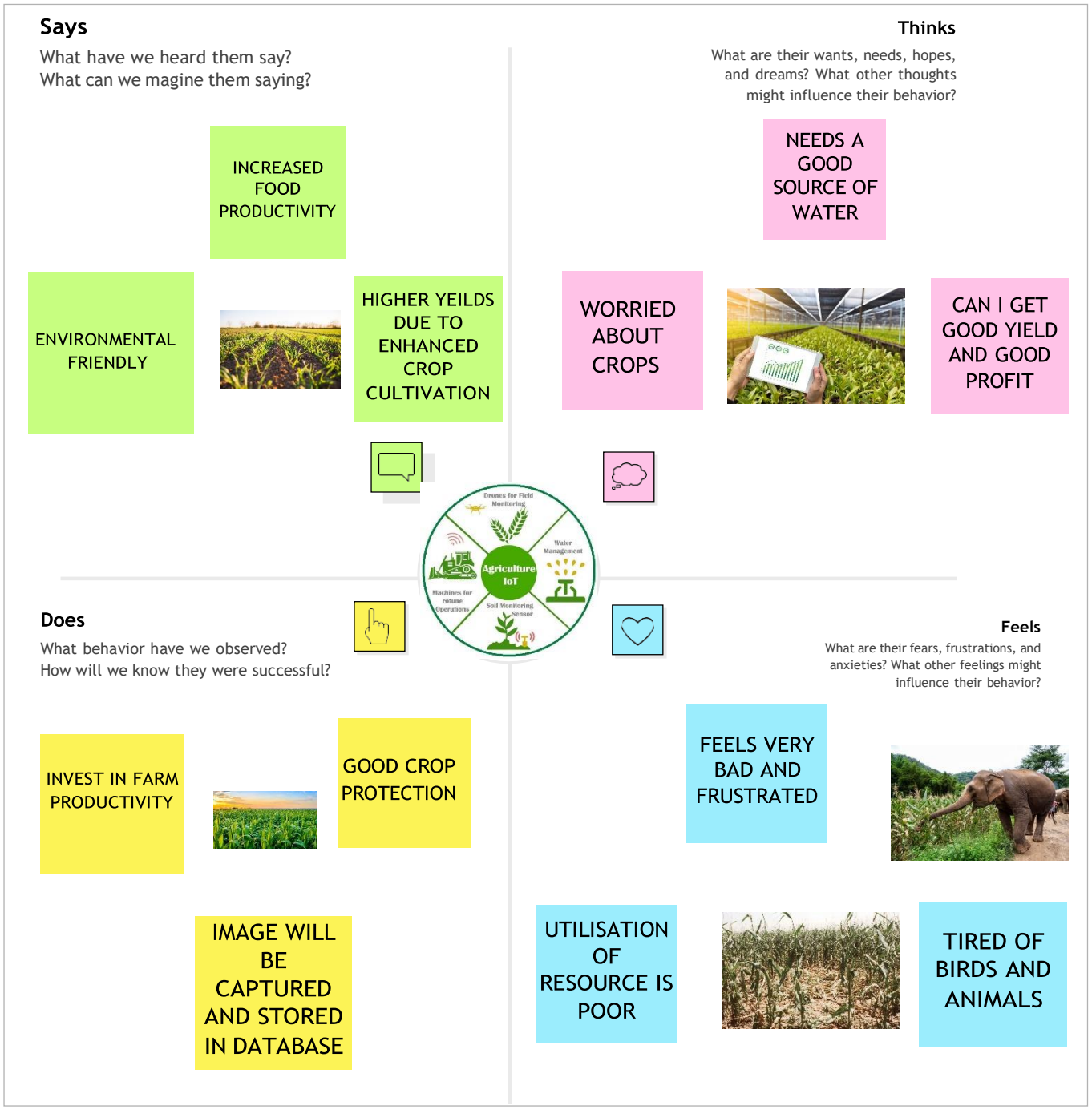
- Farmers are to be present at farm for maintenance irrespective of the weather conditions.
- They must ensure that the crops are well irrigated, and the routine activities ofthe field must be monitored by them physically.
- To get good, cultivate or yield farmers need to stay in the field for longer timefor good yield
- Demand and supply are more if the field is to be monitored monotonously if itcovers vast area
- Anytime crops may prone to various calamities which leads to poor yield of thecrops

And these are some cons that is been faced by the farmers and which leads them to cultivate crops for less yield despite heavy work

Chapter-3

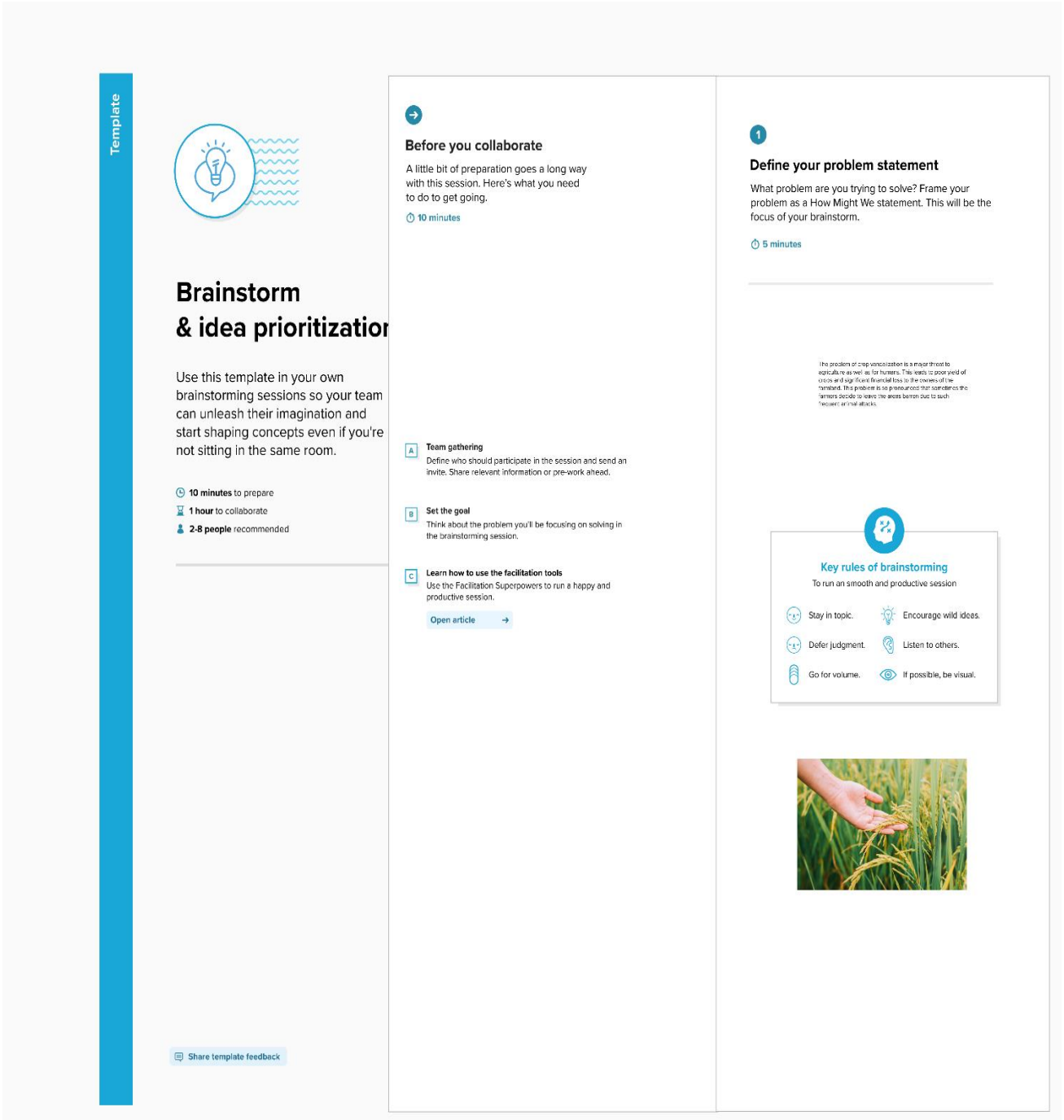
IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:



3.2 Ideation & Brainstorming:

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping

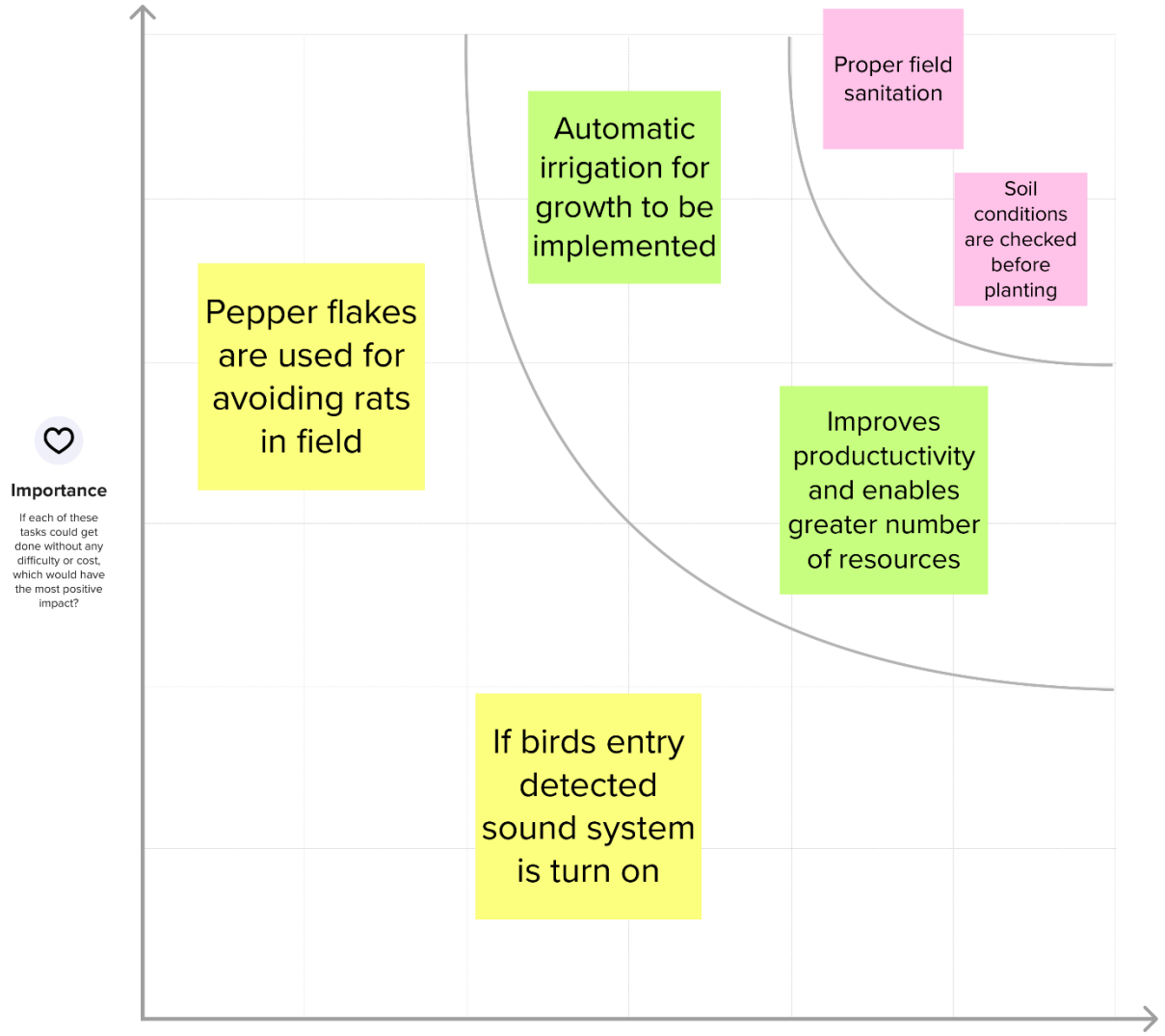
Step-3: Idea Prioritization

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



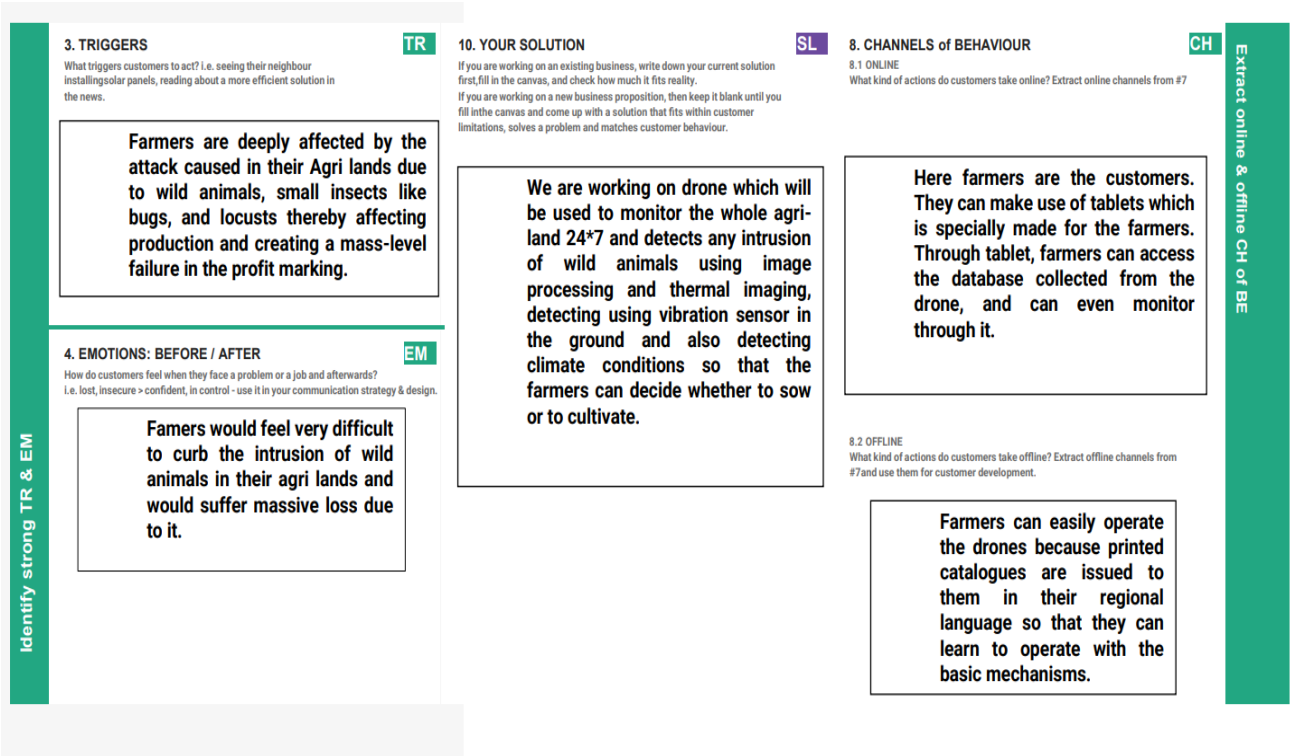
3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Wild animal attacks are a special challenge for farmers throughout the world. Animals cause serious damage to crops. They can damage the plants by feeding on plant parts or simply by running over the field and trampling over the crop fields. Therefore, wild animals may easily cause significant yield losses and provoke additional financial problems. Another aspect to consider is that wild animal crop protection requires a particularly cautious approach.
2.	Idea / Solution description	We are developing a drone, which monitors the entry of wild animals inside the field in the absence of farmers. In addition, it sprays fertilizers periodically and senses the presence of weeds growing along the crops.
3.	Novelty / Uniqueness	In the existing solutions, they have developed drones only for monitoring purposes. But we have proposed additional solutions for sensing weeds alongside crops and doing add-on work like spraying fertilizers and detecting crop growth.
4.	Social Impact / Customer Satisfaction	This technology will protect the crops from damage and hence farmers will not suffer from crop losses and it will result in an increase in crop yield.
5.	Business Model (Revenue Model)	Deploying drones for crop protection purposes with understandable technology and multipurpose tasking will enhance the protection of crops. The investment in drone technology for crop protection is higher than in the conventional types of equipment as well as functionalities.

6.	Scalability of the Solution	Crop protection using drones in case of a wild animal breach, weed formation, and climatic hazards gives the farmers great relief from this and can fully concentrate on crop production.

3.4 Problem Solution fit:

Project Title: IoT-based smart crop protection system for agriculture Project Design Phase-I – Solution Fit Template Team ID: PNT2022TMID30551



Chapter-4

REQUIREMENT ANALYSIS

4.1 Functional Requirement:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Farm monitoring (drone)	➢ Sensors (data transfer) ➢ Monitoring intrusion of wild animals, rodents, macro insects ➢ Releasing RF waves to destroy them
FR-2	Weather forecast (drone)	➢ Current weather in the field as well as in the region ➢ Temperature and humidity ➢ Wind speed and direction to detect storm ➢ Rainfall detection before impact
FR-3	Field Livestream to tablets	➢ All images and live casts from the field ➢ Live Weather forecast readings updated ➢ Live forecast from sensors in the land portion ➢ Customized tablets for easy operation, andmultilingual facilities. ➢ Generates alarm messages when any of the parameters goes abnormal
FR-4	Farm monitoring (land part)	➢ Sensors (data transfer) ➢ Footprints of the animals detected ➢ The sounds of the animals detected
FR-5	Power consumption (drone & tablet)	➢ Solar panels used for conventional supply

4.2 Non-Functional Requirement:

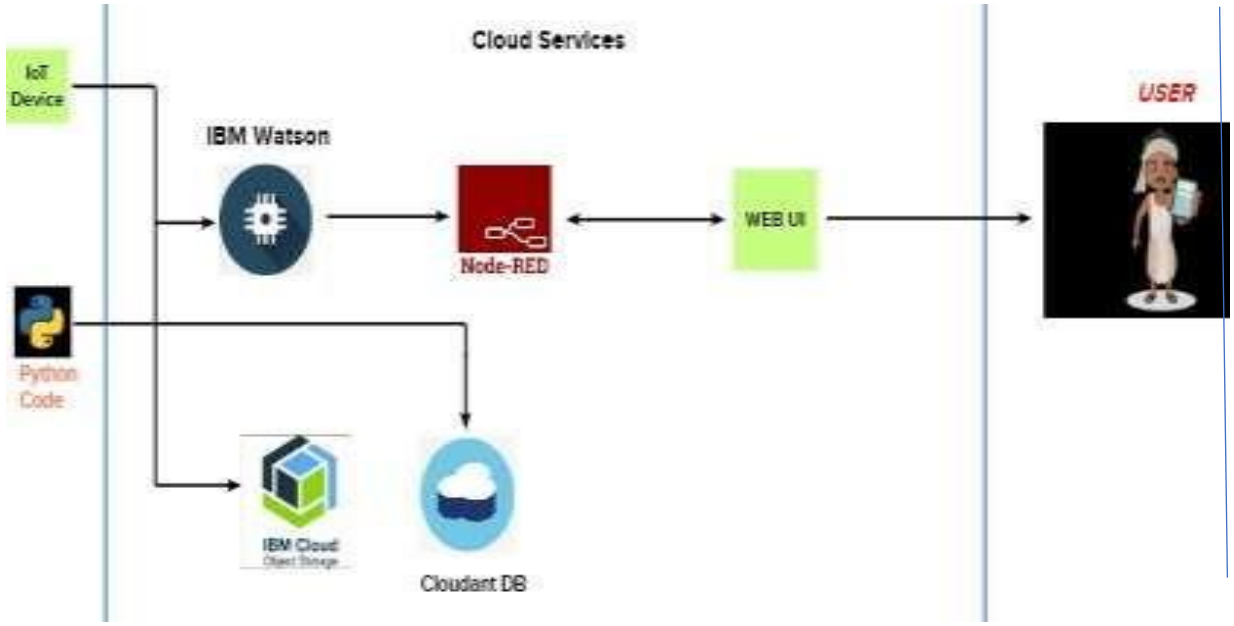
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Understandable, multilingual, user-friendly applications in the tablet
NFR-2	**Security**	➢ Database collected are accessed using TCP-IP protocol system (esp. UDP), stored in the cloud. ➢ The communication interface is done using SP-D2GCS protocol ➢ Security was established using Transport layer security protocol and IoT Security.

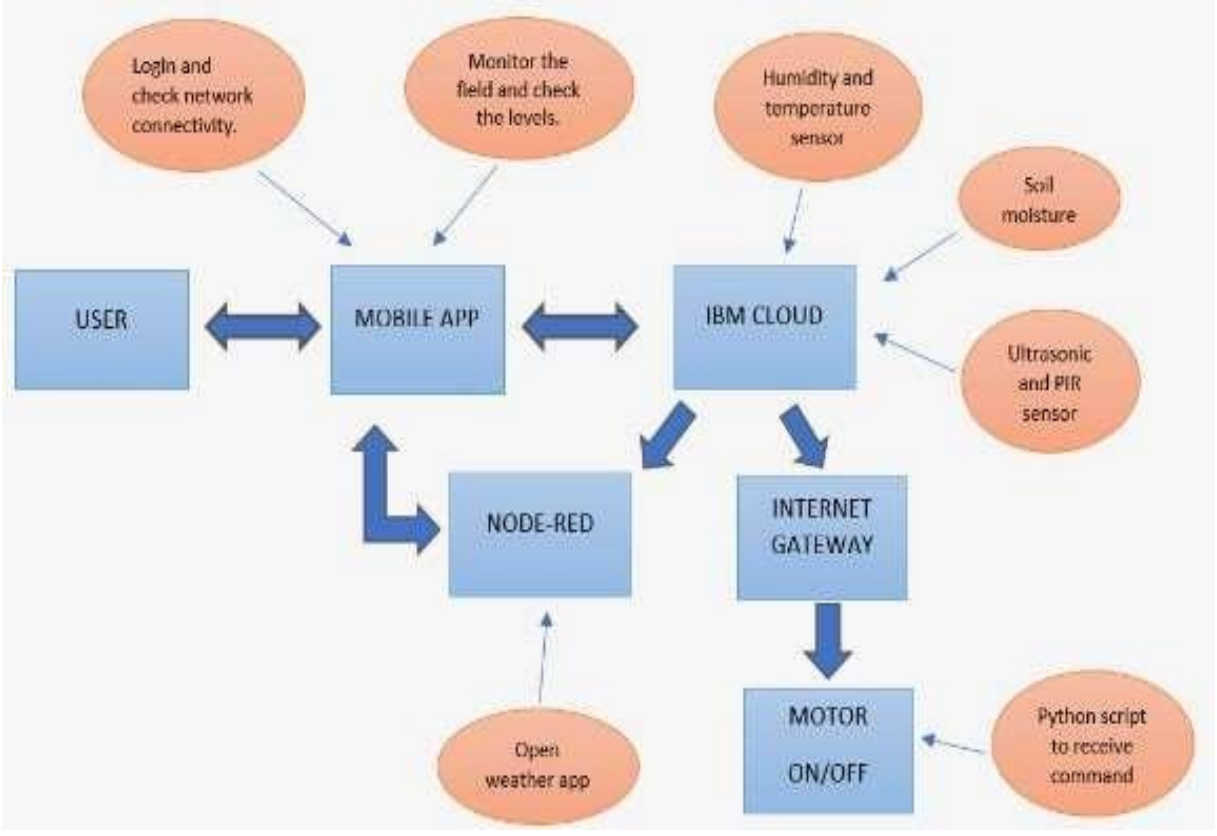
NFR-3	**Reliability**	Consistency in tolerance, accuracy maiained, application uptime enhanced
NFR-4	**Performance**	Provides accurate data, efficient functioning despite unexpected variations in climatic conditions and geographical terrains
NFR-5	**Availability**	Drone's downtime: available 90% of the time in every month Tablet's downtime: available 99% of the time

Chapter-5

PROJECT DESIGN

5.1 Data Flow Diagrams:



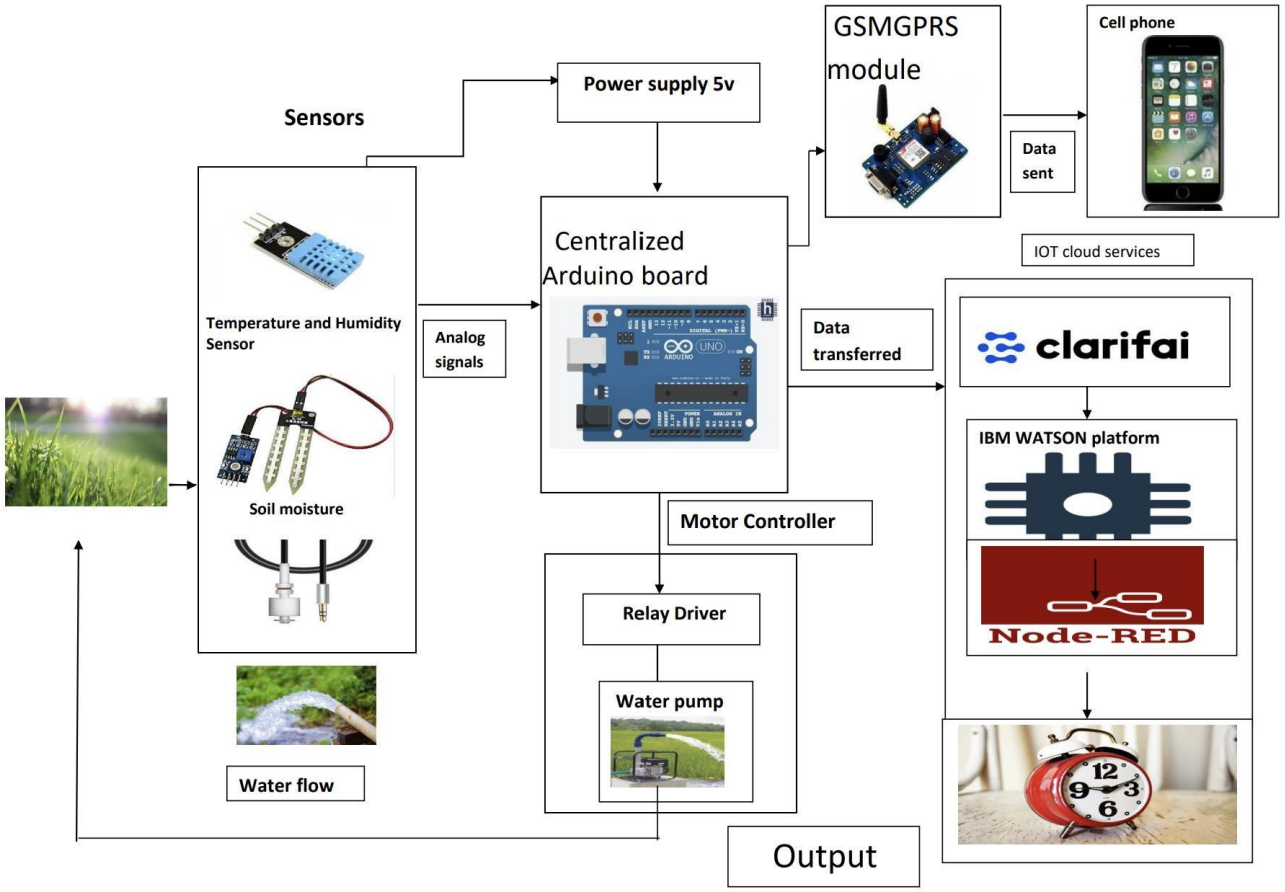


5.2 Solution & Technical Architecture:

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, behavior, 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.

GR



Explanation for the Architecture Diagram:

- 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.
- 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 applications

5.3 User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, and password, and confirmingmy password.	I can access my account/dashboard	High	Sprint-1
	Mail Confirmation	USN-2	As a user, I will receive a confirmationemailonce I have registered for the application	I can receive a confirmationemail & clickconfirm	High	Sprint-1
	Facebook Access	USN-3	As a user, I can register for the applicationthrough Facebook	I can register & access the dashboard with FacebookLogin	Low	Sprint-2
	Register	USN-4	As a user, I can register for the applicationthrough Gmail	I can register for the application	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering my email & password	I can log into the required application	High	Sprint-1
Customer (Webuser)	Same as Mobileuser	Same as Mobile user	Same as Mobile user	Same as Mobile user	High	Sprint-1
Customer Care Executive	Farmer Welfare Department	USN-1	As a user, I manage a team of representatives offering customer support	I can communicate with them in a propermanner	High	Sprint-1
	Agriculture Extension Department	USN-2	As a user, I provide technical aid to farmerson any agriculture issues	I can implementation of agriculture extension activities	High	Sprint-1
Administrator	Farm Administrator	USN-1	As a user, I provide administrative support forfarmers	I informed them about the financial and physical performance	High	Sprint-1
		USN-2	As a user, I live and work on a farm or anestate	I take responsibility	Medium	Sprint-1

Chapter – 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

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.	3	High	K.Karpagam
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	2	High	J.Divya
Sprint-2	Cloud Service	USN-3	As a user, I can register for the application through Facebook or any social media	1	Low	B.Anjali
Sprint-4		USN-4	As a user, I can register for the application through Gmail / web service	2	Medium	G.kavina.
Sprint-3	Login	USN-5	As a user, I can log into the application by entering email & password	4	High	J.Divya
Sprint-2	Pre processing	USN-6	As a farmer, the user must be able to find the system easy to access so the Prep-processes and other task must be perfect	3	High	K.Karpagam
Sprint-1	Collecting Dataset	USN-7	To collect various sources of animal threats and keep developing a dataset using Clarifai.	3	Medium	B.Anjali
Sprint-4	Integrating	USN-8	To integrate the available dataset and keep improving the accuracy of finding animals	2	Medium	G.kavina.
Sprint-3		USN-9	To find and use appropriate compiler to run and test the data so that we can implement our program	1	Low	B.Anjali
Sprint-2		USN-10	Request AVS Engineering College to deploy the project in our campus and test	1	Low	J.Divya
Sprint	**Functional**	**User**	**User Story / Task**	**Story**	**Priority**	**Team**

	Requirement (Epic)	Story Number		Points		Members
Sprint-1	Training	USN-11	As programmer, we need to train our data perfectly so that the program runs smoothly	3	High	G.kavina
Sprint-3		USN-12	Train the data using out available service and IBM dataset from server and improve that	2	Medium	K.Karpagam
Sprint-4	Coding	USN-13	To modify the code according to our program and improve the efficiency of that code	4	High	B.Anjali
Sprint-2		USN-13	To improve performance	1	Low	J.Divya
Sprint-2	Record	USN-5	To record the data and plot the graph to show the characteristics officially	4	Medium	G.Kavina
Sprint-1	Planning	USN-4	Plan the programming language and feasibility	3	High	K.Karpagam
Sprint-4		USN-14	Demonstrate the working and improve accuracy overall	2	Low	B.Anjali

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	20	6 Days	20Oct 2022	24 Oct 2022	20	31 Oct 2022
Sprint-2	20	6 Days	25 Oct 2022	29 Oct 2022	20	7 Oct 2022
Sprint-3	20	6 Days	31 Oct 2022	4 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	5 Nov 2022	11 Nov 2022	20	19 Nov 2022

6.2 Sprint Delivery Schedule:

TITLE	DESCRIPTION	DATE
Literature Survey on The Selected Project and Information Gathering	A Literature Survey is a compilation summary of research done previously in the given topic. Literature survey can be taken from books, research paper online or from any source.	19 September 2022
Prepare Empathy Map	Empathy Map is a visualization tool which can be used to get a better insight of the customer	19 September 2022
Ideation-Brainstorming	Brainstorming is a group problem solving session where ideas are shared, discussed and organized among the team members.	19 September 2022
Define Problem Statement	A Problem Statement is a concise description of the problem or issues a project seeks to address. The problem statement identifies the current state, the desired future state and any gaps between the two.	19 September 2022
Problem Solution Fit	This helps us to understand the thoughts of the customer their likes, behaviour, emotions etc.	12 October 2022
Proposed Solution	Proposed solution shows the current solution and it helps is going towards the desired result until it is achieved.	12 October 2022
Solution Architecture	Solution Architecture is a very complex process I.e it has a lot of sub-processes and branches. It helps in understanding the components and features to complete our project.	12 October 2022
Customer Journey	It helps us to analyse from the perspective of a customer, who uses our project.	15 October 2022
Functional Requirement	Here functional and nonfunctional requirements are briefed. It has specific features like usability, security, reliability, performance, availability and scalability.	15 October 2022
Data Flow Diagrams	Data Flow Diagram is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement.	15 October 2022
Technology Architecture	Technology Architecture is a more well defined version of solution architecture. It helps us analyze and understand various technologies that needs to be implemented in the project.	15 October 2022
Prepare Milestone & Activity List	It helps us to understand and evaluate our own progress and accuracy so far.	29 October 2022

Spring Delivery Plan	Sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved.	In Progress

Chapter-7

CODING & SOLUTIONING

7.1 Feature 1: Coding for Animals or pests enter into the field

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>
#include "DHT.h"

const char* ssid = "SMART-G";
const char* password = "10112019";

#define DHTPIN D6
#define G D0
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

#define ID "ryup3j"
#define DEVICE_TYPE "ESP8266"
#define DEVICE_ID "TEST"
#define TOKEN "TEST-12345"

char server[] = ID ".messaging.internetofthings.ibmcloud.com";
char publish_Topic1[] = "iot-2/evt/Data1/fmt/json";
char publish_Topic2[] = "iot-2/evt/Data2/fmt/json";
char publish_Topic3[] = "iot-2/evt/Data2/fmt/json";
char publish_Topic4[] = "iot-2/evt/Data2/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ID ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server, 1883, NULL, wifiClient);

void setup() {
  pinMode(D0,OUTPUT);
  digitalWrite(D0,HIGH);
    Serial.begin(115200);
    dht.begin();
    Serial.println();
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
```

```
47      Serial.print(".");
48    }
49    Serial.println("");
50    Serial.println(WiFi.localIP());
51
52    if (!client.connected()) {
53       Serial.print("Reconnecting client to ");
54       Serial.println(server);
55       while (!client.connect(clientId, authMethod, token)) {
56          Serial.print(".");

57          delay(500);
58       }
59       Serial.println("Connected TO IBM IoT cloud!");
60    }
61 }
62
63 long previous_message = 0;
64 void loop() {
65    client.loop();
66    long current = millis();
67    if (current - previous_message > 3000) {
68       previous_message = current;
69        float hum = dht.readHumidity();
70        float temp = dht.readTemperature();
71        float MOI =  map(analogRead(A0), 0, 1023, 100, 0);
72        float bi = map(digitalRead(D1), 0, 1, 100, 0 );
73        if (isnan(hum) || isnan(temp)  ){
74    Serial.println(F("Failed to read from DHT sensor!"));
75    return;
76   }
77
78   Serial.print("Temperature: ");
79   Serial.print(temp);
80   Serial.print("°C");
81   Serial.print(" Humidity: ");
82   Serial.print(hum);
83   Serial.print("%");
84   Serial.print("SOIL MOITURE: ");
85   Serial.print(MOI);
86   Serial.print("ANIMAL AND BIRD: ");
87   Serial.print(bi);
88   if(MOI<=10)
89   {
90      digitalWrite(D0,LOW);
91      delay(100);
92      digitalWrite(D0,HIGH);
93    }
```

```
94    else
95    {
96      digitalWrite(D0,HIGH);
97    }
98
99
100     String payload = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
101           payload += ",\"Temperature\":";
102           payload += temp;
103           payload += "}}";
104
105     Serial.print("Sending payload: ");
106     Serial.println(payload);
107
108     if (client.publish(publish_Topic1, (char*) payload.c_str())) {
109        Serial.println("Published successfully");
110     } else {
111        Serial.println("Failed");
112     }
113     String payload1 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
114           payload1 += ",\"Humidity\":";
115           payload1 += hum;
116           payload1 += "}}";


117           Serial.print("Sending payload: ");

118           Serial.println(payload1);
119           Serial.println('\n');
120
121      if (client.publish(publish_Topic2, (char*) payload1.c_str())) {
122        Serial.println("Published successfully");
123     } else {
124        Serial.println("Failed");
125     }
126
127
128     String payload3 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
129           payload3 += ",\"Moiture\":";
130           payload3 += MOI;
131           payload3 += "}}";
132
133     Serial.print("Sending payload: ");
134     Serial.println(payload3);
135
136     if (client.publish(publish_Topic3, (char*) payload3.c_str())) {
137        Serial.println("Published successfully");
138     } else {
```

```
            Serial.println("Failed");
        }



String payload4 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
            payload4 += ",\"Animal&Bird\":";
            payload4 += bi;
            payload4 += "}}";

        Serial.print("Sending payload: ");
        Serial.println(payload4);

        if (client.publish(publish_Topic4, (char*) payload4.c_str())) {
            Serial.println("Published successfully");
        } else {
            Serial.println("Failed");
        }
    }
}
```

158.1 Feature 2: coding for moisture level checking

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>
#include "DHT.h"

const char* ssid = "SMART-G";
const char* password = "10112019";

#define DHTPIN D6
#define G D0
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

#define ID "ryup3j"
#define DEVICE_TYPE "ESP8266"
#define DEVICE_ID "TEST"
#define TOKEN "TEST-12345"

char server[] = ID ".messaging.internetofthings.ibmcloud.com";
char publish_Topic1[] = "iot-2/evt/Data1/fmt/json";
char publish_Topic2[] = "iot-2/evt/Data2/fmt/json";
char publish_Topic3[] = "iot-2/evt/Data2/fmt/json";
char publish_Topic4[] = "iot-2/evt/Data2/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ID ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server, 1883, NULL, wifiClient);

void setup() {
  pinMode(D0,OUTPUT);
  digitalWrite(D0,HIGH);
    Serial.begin(115200);
    dht.begin();
    Serial.println();
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    }
```

```
48      Serial.println("");
49      Serial.println(WiFi.localIP());
50
51      if (!client.connected()) {
52          Serial.print("Reconnecting client to ");
53          Serial.println(server);
54          while (!client.connect(clientId, authMethod, token)) {
55              Serial.print(".");
56              delay(500);
57          }

58          Serial.println("Connected TO IBM IoT cloud!");
59      }
60  }
61
62  long previous_message = 0;
63  void loop() {
64      client.loop();
65      long current = millis();
66      if (current - previous_message > 3000) {
67          previous_message = current;
68           float hum = dht.readHumidity();
69           float temp = dht.readTemperature();
70           float MOI =  map(analogRead(A0), 0, 1023, 100, 0);
71           float bi = map(digitalRead(D1), 0, 1, 100, 0 );
72           if (isnan(hum) || isnan(temp)  ){
73      Serial.println(F("Failed to read from DHT sensor!"));
74      return;
75    }
76
77    Serial.print("Temperature: ");
78    Serial.print(temp);
79    Serial.print("°C");
80    Serial.print(" Humidity: ");
81    Serial.print(hum);
82    Serial.print("%");
83    Serial.print("SOIL MOITURE: ");
84    Serial.print(MOI);
85    Serial.print("ANIMAL AND BIRD: ");
86    Serial.print(bi);
87    if(MOI<=10)
88    {
89        digitalWrite(D0,LOW);
90        delay(100);
```

```
91        digitalWrite(D0,HIGH);
92      }
93      else
94      {
95        digitalWrite(D0,HIGH);
96      }
97
98
99        String payload = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
100             payload += ",\"Temperature\":";
101             payload += temp;
102             payload += "}}";
103
104       Serial.print("Sending payload: ");
105       Serial.println(payload);
106
107       if (client.publish(publish_Topic1, (char*) payload.c_str())) {
108          Serial.println("Published successfully");
109       } else {
110          Serial.println("Failed");
111       }
112       String payload1 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";

113             payload1 += ",\"Humidity\":";
114             payload1 += hum;
115             payload1 += "}}";
116             Serial.print("Sending payload: ");

117             Serial.println(payload1);
118             Serial.println('\n');
119
120        if (client.publish(publish_Topic2, (char*) payload1.c_str())) {
121          Serial.println("Published successfully");
122       } else {
123          Serial.println("Failed");
124       }
125
126
127      String payload3 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
128             payload3 += ",\"Moiture\":";
129             payload3 += MOI;
130             payload3 += "}}";
131
132       Serial.print("Sending payload: ");
```

```
133       Serial.println(payload3);
134
135       if (client.publish(publish_Topic3, (char*) payload3.c_str())) {
136          Serial.println("Published successfully");
137       } else {
138          Serial.println("Failed");
139       }
140
141
142
143String payload4 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
144            payload4 += ",\"Animal&Bird\":";
145            payload4 += bi;
146            payload4 += "}}";
147
148       Serial.print("Sending payload: ");
149       Serial.println(payload4);
150
151       if (client.publish(publish_Topic4, (char*) payload4.c_str())) {
152          Serial.println("Published successfully");
153       } else {
154          Serial.println("Failed");
155       }
156    }
157}
```

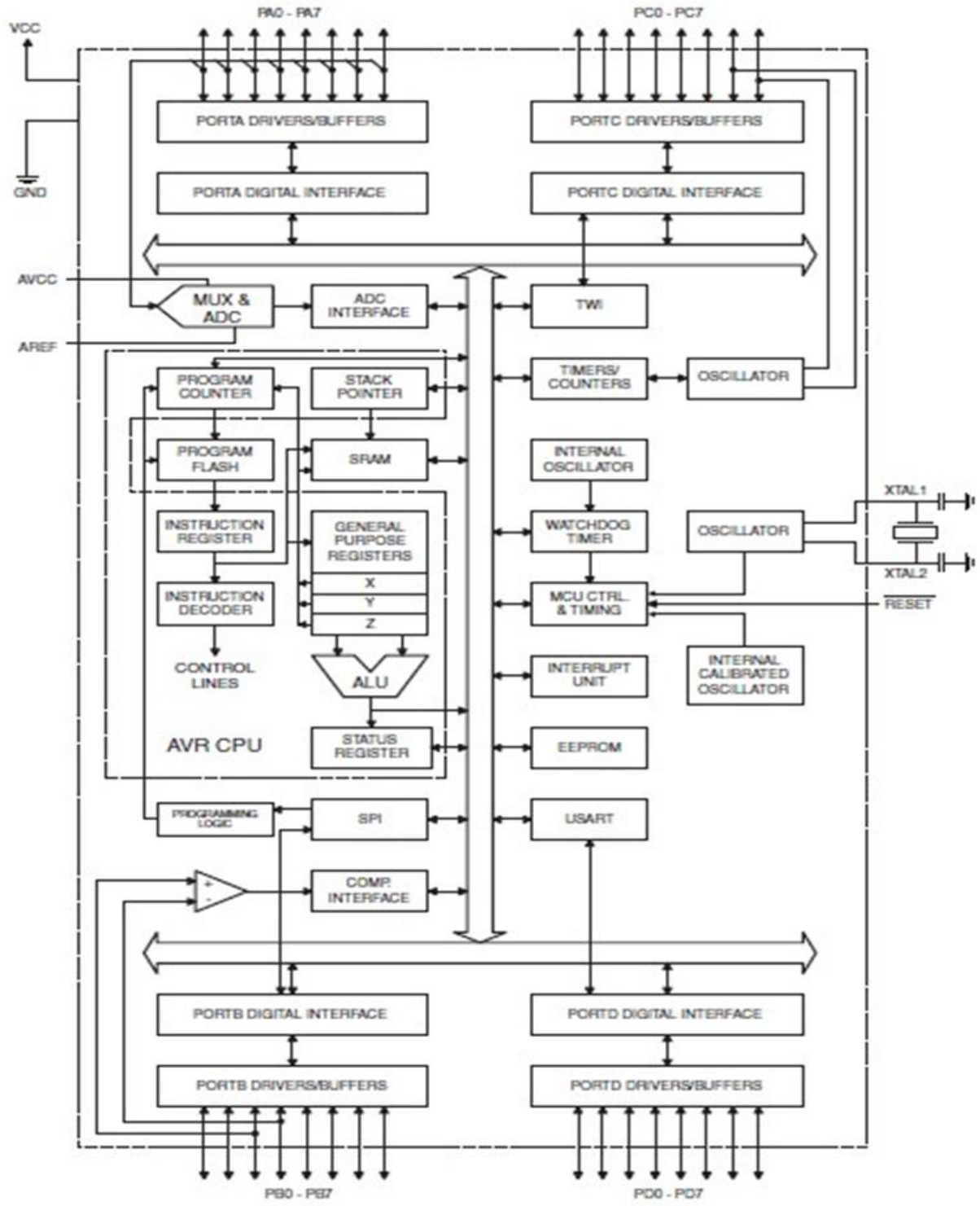
157.1 Feature 3:

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>
#include "DHT.h"

const char* ssid = "SMART-G";
const char* password = "10112019";

#define DHTPIN D6
#define G D0
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

#define ID "ryup3j"
#define DEVICE_TYPE "ESP8266"
#define DEVICE_ID "TEST"
#define TOKEN "TEST-12345"

char server[] = ID ".messaging.internetofthings.ibmcloud.com";
char publish_Topic1[] = "iot-2/evt/Data1/fmt/json";
char publish_Topic2[] = "iot-2/evt/Data2/fmt/json";
char publish_Topic3[] = "iot-2/evt/Data2/fmt/json";
char publish_Topic4[] = "iot-2/evt/Data2/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ID ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server, 1883, NULL, wifiClient);

void setup() {
  pinMode(D0,OUTPUT);
  digitalWrite(D0,HIGH);
    Serial.begin(115200);
    dht.begin();
    Serial.println();
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
```

```
46        Serial.print(".");
47      }
48      Serial.println("");
49      Serial.println(WiFi.localIP());
50
51      if (!client.connected()) {
52          Serial.print("Reconnecting client to ");
53          Serial.println(server);
54          while (!client.connect(clientId, authMethod, token)) {
55              Serial.print(".");
56              delay(500);
57          }

58          Serial.println("Connected TO IBM IoT cloud!");
59      }
60  }
61
62  long previous_message = 0;
63  void loop() {
64      client.loop();
65      long current = millis();
66      if (current - previous_message > 3000) {
67          previous_message = current;
68           float hum = dht.readHumidity();
69           float temp = dht.readTemperature();
70           float MOI =  map(analogRead(A0), 0, 1023, 100, 0);
71           float bi = map(digitalRead(D1), 0, 1, 100, 0 );
72           if (isnan(hum) || isnan(temp)  ){
73      Serial.println(F("Failed to read from DHT sensor!"));
74      return;
75    }
76
77    Serial.print("Temperature: ");
78    Serial.print(temp);
79    Serial.print("°C");
80    Serial.print(" Humidity: ");
81    Serial.print(hum);
82    Serial.print("%");
83    Serial.print("SOIL MOITURE: ");
84    Serial.print(MOI);
85    Serial.print("ANIMAL AND BIRD: ");
86    Serial.print(bi);
87    if(MOI<=10)
88    {
```

```
89        digitalWrite(D0,LOW);
90        delay(100);
91        digitalWrite(D0,HIGH);
92      }
93      else
94      {
95        digitalWrite(D0,HIGH);
96      }
97
98
99        String payload = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
100             payload += ",\"Temperature\":";
101             payload += temp;
102             payload += "}}";
103
104       Serial.print("Sending payload: ");
105       Serial.println(payload);
106
107       if (client.publish(publish_Topic1, (char*) payload.c_str())) {
108          Serial.println("Published successfully");
109       } else {
110          Serial.println("Failed");
111       }
112       String payload1 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";

113             payload1 += ",\"Humidity\":";
114             payload1 += hum;
115             payload1 += "}}";
116             Serial.print("Sending payload: ");

117             Serial.println(payload1);
118             Serial.println('\n');
119
120        if (client.publish(publish_Topic2, (char*) payload1.c_str())) {
121          Serial.println("Published successfully");
122       } else {
123          Serial.println("Failed");
124       }
125
126
127       String payload3 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
128             payload3 += ",\"Moiture\":";
129             payload3 += MOI;
```

```
130             payload3 += "}}";
131
132         Serial.print("Sending payload: ");
133         Serial.println(payload3);
134
135         if (client.publish(publish_Topic3, (char*) payload3.c_str())) {
136             Serial.println("Published successfully");
137         } else {
138             Serial.println("Failed");
139         }
140
141
142
143String payload4 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
144             payload4 += ",\"Animal&Bird\":";
145             payload4 += bi;
146             payload4 += "}}";
147
148         Serial.print("Sending payload: ");
149         Serial.println(payload4);
150
151         if (client.publish(publish_Topic4, (char*) payload4.c_str())) {
152             Serial.println("Published successfully");
153         } else {
154             Serial.println("Failed");
155         }
156     }
157}
```

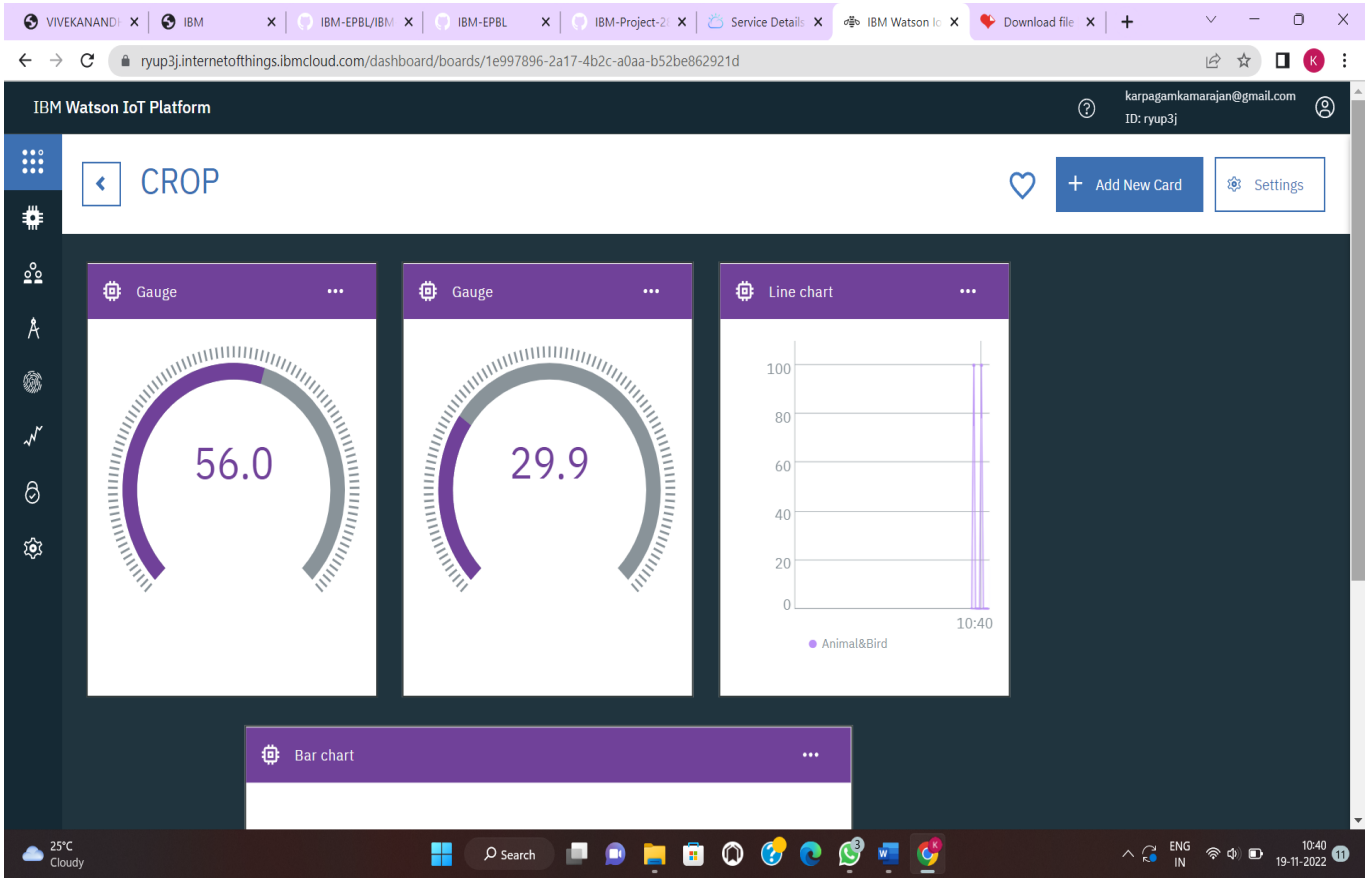

PA0 - PA7
PC0 - PC7
VCC
GND
PORTA DRIVERS/BUFFERS
PORTC DRIVERS/BUFFERS
PORTA DIGITAL INTERFACE
PORTC DIGITAL INTERFACE
AVCC
AREF
MUX & ADC
ADC INTERFACE
TWI
PROGRAM COUNTER
STACK POINTER
TIMERS/ COUNTERS
OSCILLATOR
PROGRAM FLASH
SRAM
INTERNAL OSCILLATOR
XTAL1
XTAL2
INSTRUCTION REGISTER
GENERAL PURPOSE REGISTERS
WATCHDOG TIMER
OSCILLATOR
X
Y
Z
INSTRUCTION DECODER
MCU CTRL. & TIMING
RESET
CONTROL LINES
ALU
INTERRUPT UNIT
INTERNAL CALIBRATED OSCILLATOR
AVR CPU
STATUS REGISTER
EEPROM
PROGRAMMING LOGIC
SPI
USART
COMP. INTERFACE
PORTB DIGITAL INTERFACE
PORTD DIGITAL INTERFACE
PORTB DRIVERS/BUFFERS
PORTD DRIVERS/BUFFERS
PB0 - PB7
PD0 - PD7

Chapter-8

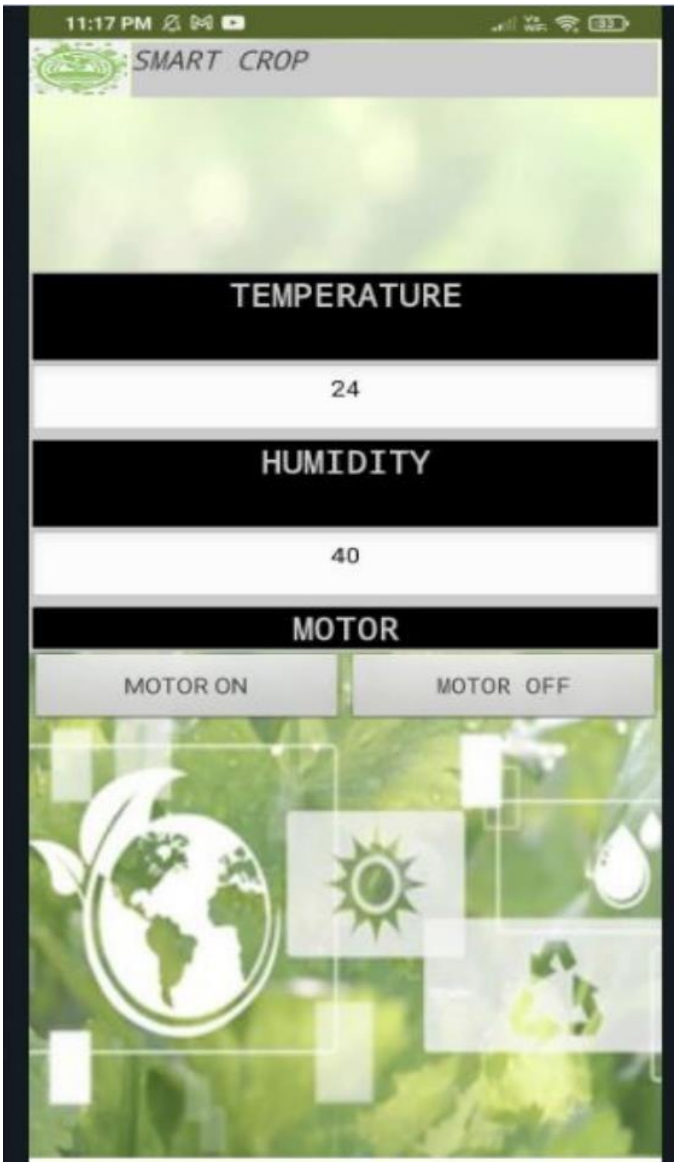
RESULTS

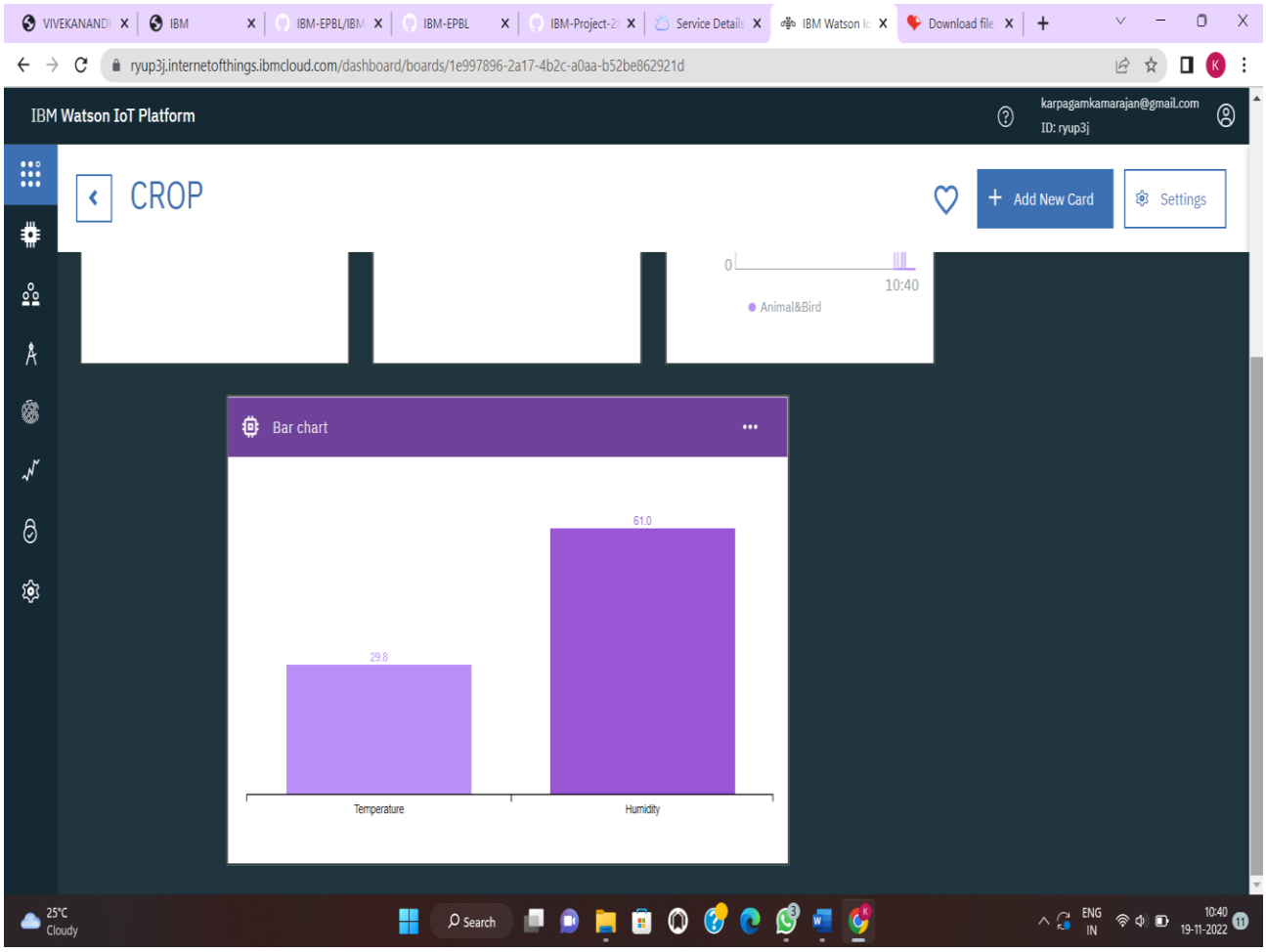
SOIL MOISTURE

ANIMAL AND BIRD DETECT

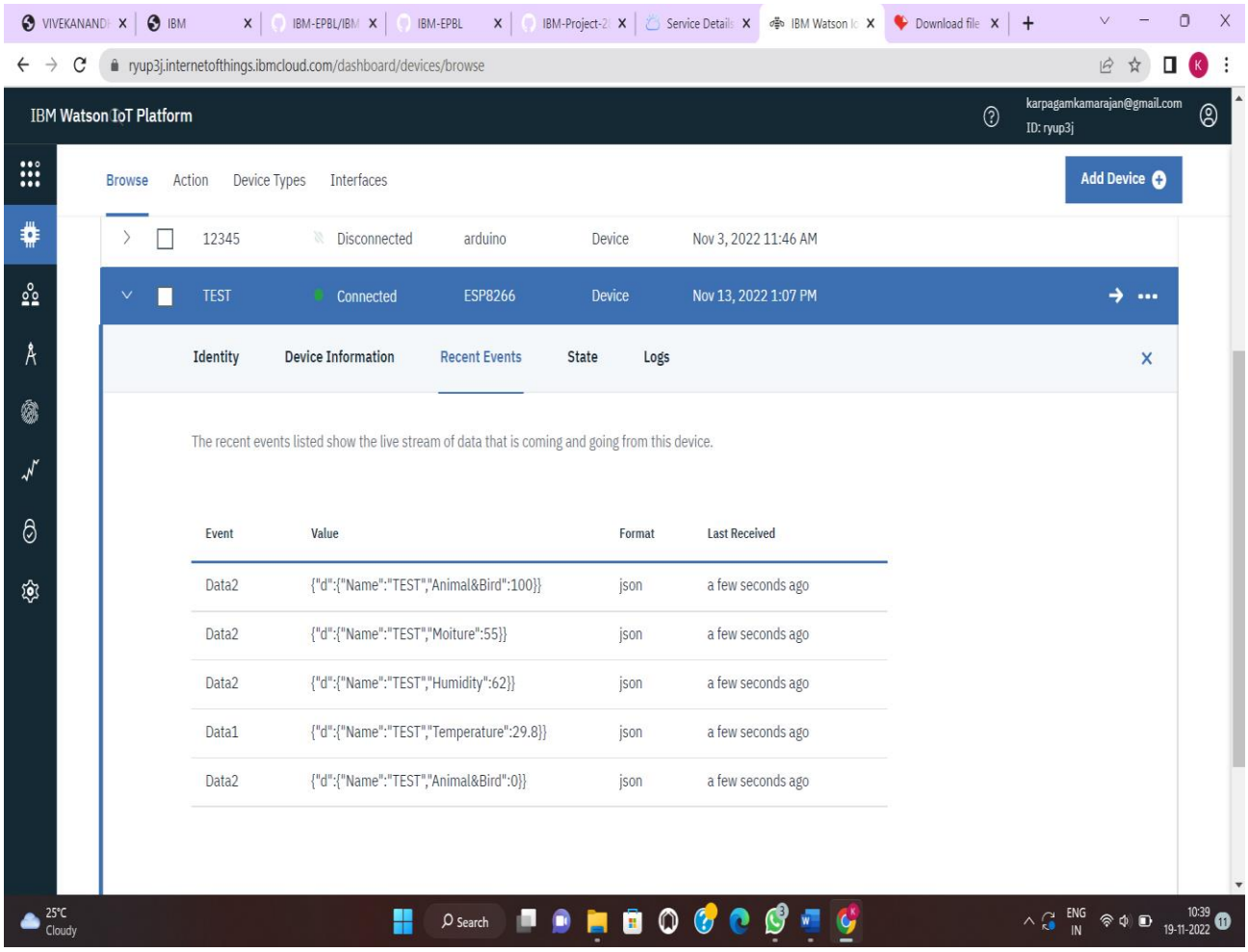


MOBILE APP


11:17 PM
SMART CROP
TEMPERATURE
24
HUMIDITY
40
MOTOR
MOTOR ON
MOTOR OFF



TEMPERATURE AND HUMIDITY


IBM Watson IoT Platform
karpagamkamarajan@gmail.com
ID: ryup3j
Browse
Action
Device Types
Interfaces
Add Device
12345
Disconnected
arduino
Device
Nov 3, 2022 11:46 AM
TEST
Connected
ESP8266
Device
Nov 13, 2022 1:07 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
Data2
{"d":{"Name":"TEST","Animal&Bird":100}}
json
a few seconds ago
Data2
{"d":{"Name":"TEST","Moiture":55}}
json
a few seconds ago
Data2
{"d":{"Name":"TEST","Humidity":62}}
json
a few seconds ago
Data1
{"d":{"Name":"TEST","Temperature":29.8}}
json
a few seconds ago
Data2
{"d":{"Name":"TEST","Animal&Bird":0}}
json
a few seconds ago

Chapter – 9

ADVANTAGES AND DISADVANTAGES

Advantages:

- Farms can be monitored and controlled remotely.
- Increase in convenience to farmers.
- Less Manpower.
- Better standards of living.

Disadvantages:

- Lack of internet/connectivity issues.
- Added cost of internet and internet gateway infrastructure.
- Farmers wanted to adapt the use of WebApp.

Chapter – 10

CONCLUSION

This system focuses on developing devices and tool to manage, display andalert the users using the advantages of a wireless sensor network system. It aims at making agriculture smart using automation and IoT. The cloud computing devices are used at the end of the system that can create a whole computing system from sensors to tools that observe data from agriculture field. It proposes a novel methodology for smart farming by including a smart sensing system and smart irrigator system through wireless communication technology. Thus, the objective of the project to implement an IoT system in order to help farmers to control and monitor their farms has been implemented successfully.

Chapter – 11

FUTURE SCOPE

Agriculture domains encounters with many challenges starting from soil parameters, seed sowing, crop growth and its quality, weed handling, disease management till harvesting and storing crop. Artificial intelligence driven techniques along with other available tools and automation can address these challenges and proven the revolution in agriculture. Most popular AI application in agriculture is use of Robot and Drones, they perform almost all task like humans even at a faster rate with accuracy. From literature review it is clear that precision farming is probable by integrating sensors, cameras, data analytics, GPS and remote sensing. Image recognitions software's, IoT sensors can be used for disease recognition at primary stages and hence crop health can be supervised which increases superior quality production with minimum loss. Table 1 demonstrate the various applications in view of Smart Agriculture for improved evolution as well as superiority. Still there are several challenges associated with AI and IoT application in smart agriculture which is the promising future to be explored area for researchers. Some of major challenges are: • Awareness issues • Hardware implementation challenges • Cost of software and hardware • Network management • Energy management • Privacy issues • Security challenges • Interoperability of systems with the induction of Computer vision, Deep learning, Big data also agriculture sector has influenced a lot. Researchers can integrate IoT sensors along with smart systems and computational optimization algorithms to overcome the limitations/shortcomings. Smart Agriculture has a budding potential towards productivity, precision, optimization, adaptive resource management and intelligent food traceability. It will contribute to environment also in terms of efficient use of water, prevent disease contamination and precise use of pesticides.

Chapter – 12

APPENDIX

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>
#include "DHT.h"

const char* ssid = "SMART-G";
const char* password = "10112019";

#define DHTPIN D6
#define G D0
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

#define ID "ryup3j"
#define DEVICE_TYPE "ESP8266"
#define DEVICE_ID "TEST"
#define TOKEN "TEST-12345"

char server[] = ID ".messaging.internetofthings.ibmcloud.com";
char publish_Topic1[] = "iot-2/evt/Data1/fmt/json";
char publish_Topic2[] = "iot-2/evt/Data2/fmt/json";
char publish_Topic3[] = "iot-2/evt/Data2/fmt/json";
char publish_Topic4[] = "iot-2/evt/Data2/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ID ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server, 1883, NULL, wifiClient);

void setup() {
  pinMode(D0,OUTPUT);
  digitalWrite(D0,HIGH);
    Serial.begin(115200);
    dht.begin();
    Serial.println();
    WiFi.begin(ssid, password);
```

```
    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    }
    Serial.println("");
    Serial.println(WiFi.localIP());

    if (!client.connected()) {
        Serial.print("Reconnecting client to ");
        Serial.println(server);
        while (!client.connect(clientId, authMethod, token)) {
            Serial.print(".");
            delay(500);
        }
        Serial.println("Connected TO IBM IoT cloud!");
    }
}

long previous_message = 0;
void loop() {
    client.loop();
    long current = millis();
    if (current - previous_message > 3000) {
        previous_message = current;
         float hum = dht.readHumidity();
         float temp = dht.readTemperature();
         float MOI =  map(analogRead(A0), 0, 1023, 100, 0);
         float bi = map(digitalRead(D1), 0, 1, 100, 0 );
         if (isnan(hum) || isnan(temp)  ){
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  Serial.print("Temperature: ");
  Serial.print(temp);
  Serial.print("°C");
  Serial.print(" Humidity: ");
  Serial.print(hum);
  Serial.print("%");
  Serial.print("SOIL MOITURE: ");
  Serial.print(MOI);
  Serial.print("ANIMAL AND BIRD: ");
  Serial.print(bi);
  if(MOI<=10)
```

```
{
   digitalWrite(D0,LOW);
   delay(100);
   digitalWrite(D0,HIGH);
  }
  else
  {
   digitalWrite(D0,HIGH);
  }


    String payload = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
          payload += ",\"Temperature\":";
          payload += temp;
          payload += "}}";

     Serial.print("Sending payload: ");
     Serial.println(payload);

     if (client.publish(publish_Topic1, (char*) payload.c_str())) {
        Serial.println("Published successfully");
     } else {
        Serial.println("Failed");
     }
     String payload1 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
          payload1 += ",\"Humidity\":";
          payload1 += hum;
          payload1 += "}}";
          Serial.print("Sending payload: ");
          Serial.println(payload1);
          Serial.println('\n');

      if (client.publish(publish_Topic2, (char*) payload1.c_str())) {
        Serial.println("Published successfully");
     } else {
        Serial.println("Failed");
     }


    String payload3 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
          payload3 += ",\"Moiture\":";
          payload3 += MOI;
          payload3 += "}}";
```

```
        Serial.print("Sending payload: ");
        Serial.println(payload3);

        if (client.publish(publish_Topic3, (char*) payload3.c_str())) {
           Serial.println("Published successfully");
        } else {
           Serial.println("Failed");
        }



String payload4 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
             payload4 += ",\"Animal&Bird\":";
             payload4 += bi;
             payload4 += "}}";

        Serial.print("Sending payload: ");
        Serial.println(payload4);

        if (client.publish(publish_Topic4, (char*) payload4.c_str())) {
           Serial.println("Published successfully");
        } else {
           Serial.println("Failed");
      }
    }
}
```