IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

NAALAIYA TIRAN PROJECT BASED LEARNING ON PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY

AND

ENTREPRENEURSHIP

A PROJECT REPORT

KAVIARARSAN R	**611819104021**
BHOOBALAN K	**611819104005**
SANJAY KUMAR K	**611819104037**
THARUN KUMAR K V	**611819104051**

TEAM ID : PNT2022TMID40891
FACULTY MENTORS NAME : B SAKTHIVEL

INDUSRTY MENTORS NAME : DINESH

EVALUATOR NAME : Dr. S. CHANDRA SEKARAN

P.S.V. COLLEGE OF ENGINEERING AND TECHNOLOGY

(An ISO 9001:2015 Certified Institution)

(Accredited by NAAC with 'A' Grade)

KRISHNAGIRI-635108

NOVEMBER 2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "IOT BAESD SMART CROP PROTECTION SYSTEM FOR AGRICULTURE" is the bonafide work of "KAVIARASAN R (611819104021), BHOOBALAN K(611819104005), SANJAY KUMAR K (611819104037) and THARUN KUMAR K V (611819104051)" who carried out the project work under my supervision.

SIGNATURE

Prof. B. SAKTHIVEL.,M.E.,(Ph.D)

Head of the Department

Dept of CSE,
P.S.V.College of
Engineering & Technology,
Krishnagiri-635 108.

SIGNATURE

Prof. B. SAKTHIVEL.,M.E.,(Ph.D)

Faculty Mentors,

Dept of CSE,
P.S.V.College of
Engineering & Technology,
Krishnagiri-635 108.

Submitted for the **Naalaiya Tiran Project based learning on professional readiness for Innovation, Employability and Entrepreneurship** held on …….at P.S.V College of Engineering and Technology, Krishnagiri.

INTERNAL EXAMINAR

EXTERNAL EXAMINAR

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW:

Today, technology has penetrated every part of human life. But the contribution of technology to the field of agriculture is considerably low when compared to the other sectors, which saw an incremental growth over the last decade.

The domain of Agriculture contributes the most to the Indian economy and about 1/3rd of India's population is directly dependent on agriculture for their source of income. Considering this, even a small improvement in this sector will make a huge impact on the Indian economy and on the life of farmers. This helps farmers and consumers equally as it isthe consumers in the end, who get to enjoy low priced goods without deterioration in quality.

To achieve this, we have to overcome the hurdles faced by farmers, which mostly revolve around crop disease, improper maintenance of crops, lack of details about the qualityof soil and intervention of animals and birds. To overcome this, in this project we propose 'Anintelligent crop protection system', the main objective of which is to improve the yield and increase the profit for farmers. An intelligent crop protection system uses data from moisture,motion, temperature, humidity sensors and updates the data in real time in IBM cross platform IOT cloud interface. The motors and the sprinkling system are activated based on the data from the sensors. Also when the motion sensor detects motion, the farmer is notified with that through the mobile application. This helps the farmers in protecting the crop from the animalsand birds which destroy the crop.And also ease up the maintenance process. The historical data from sensors are storedin cloud, so this can also be used for soil evaluation and this also helps to plan, which type ofcrops are to be planted in the upcoming seasons so that the yield is high .

1.2 PURPOSE:

A vast majority of the people are invariably affected by the production of crops. Farmers, for example, rely on them for their survival. The consumers, on the other hand, depend on the crops as it provides them with a multitude of utilities. It therefore, becomes essential to protect and maintain these crops. The project aims at improving the farmers' situation by preventing them from incurring losses due to the damage of crops.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM:

In real time, it was learnt that the size of the animal is found out by using several PIR sensors. PIR sensors can be used to determine the height of the animals instead of using a camera for image processing. This reduces the processing time and power. The crop protectionis majorly dependent on the moisture content of the soil, the temperature and humidity of the surrounding environment. Additionally, tracking of the damaged crops location is done and the camera is activated only at that instant in order to capture the image.

From the literatures survey performed it is evident that image based animal intrusion identification is not necessary in all situations because it requires high computation power, andthe cost of the installation will be high when compared to that of a typical sensor based intrusion identification.

2.2 REFERNCES:

- Upl-ltd.com
- UPL-Agriculture solutions & services provider.
- https://smartinternz.com/assets/docs/Smart%20Home%20Automation%20using%20IBM%20cloud%20Service s%20(1).pdf
- https://smartinternz.com/assets/docs/Smart%20Home%20Automation%20using%20IBM%20cloud%20Service s%20(1).pdf
- https://openweathermap.org/
- https://smartinternz.com/assets/docs/Sending%20Http%20request%20to%20Open%20weather%20map%20website%20to%20get%20the%20weather%20forecast.pdf
- https://www.youtube.com/watch?v=cicTw4SEdxk
- https://smartinternz.com/assets/docs/Smart%20Home%20Automation%20using%20IBM%20cloud%20Service s%20(1).pdf
- https://github.com/rachuriharish23/ibmsubscribe

2.3 PROBLEM STATEMENT DEDINITION:

- Problem of getting damage of crops by animals remains a problem when a better smart crop Protection device is not built.
- Preventing damage from animals is quiet difficult.
- The movement in the crop production is usually determined by the millions of farmers who produce crops production.
- The role of Investing for building Electric Fence and Scarecrow can be avoided.

CHAPTER 3
IDEATION & PROPOSED SOLUTION

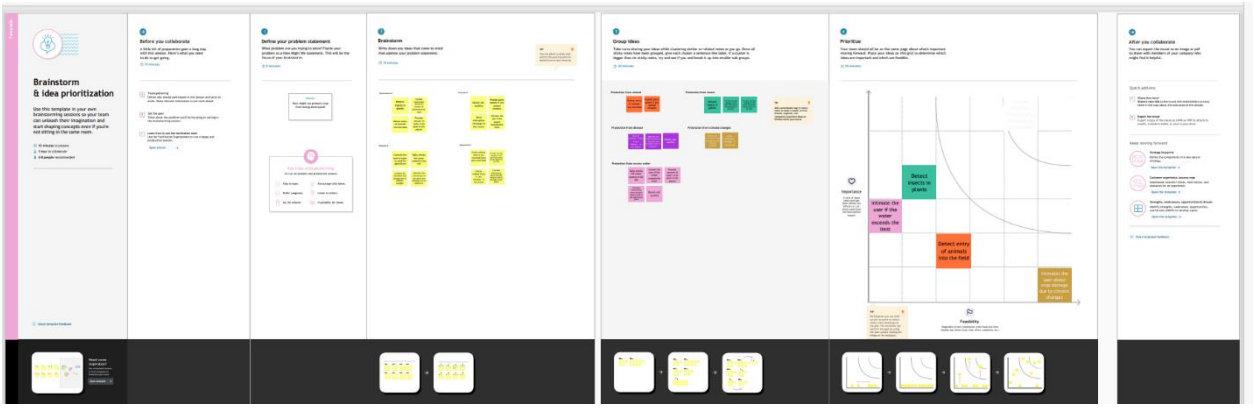
3.1 EMPATHY MAP CANVAS:

An empathy map is a collaborative visualization used to express clearly what one knows about a particular type of user. It externalizes knowledge about users in order to createa shared understanding of user needs, and aid in decision making.Empathy maps are split into 4 quadrants (Says, Thinks, Does, and Feels), with the userin the middle. Empathy maps provide a glance into who a user is as a whole.

3.2 IDEATION & BRAINSTORMING:

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. Brainstorming is usually conducted by getting a group of people together to come up with either general new ideas or ideas for solving a specific problem or dealing with a specific situation. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a

group activity. Both brainstorming and ideation are processes invented to create new valuable ideas, perspectives, concepts and insights, and both are methods for envisioning new frameworks and systemic problem solving.

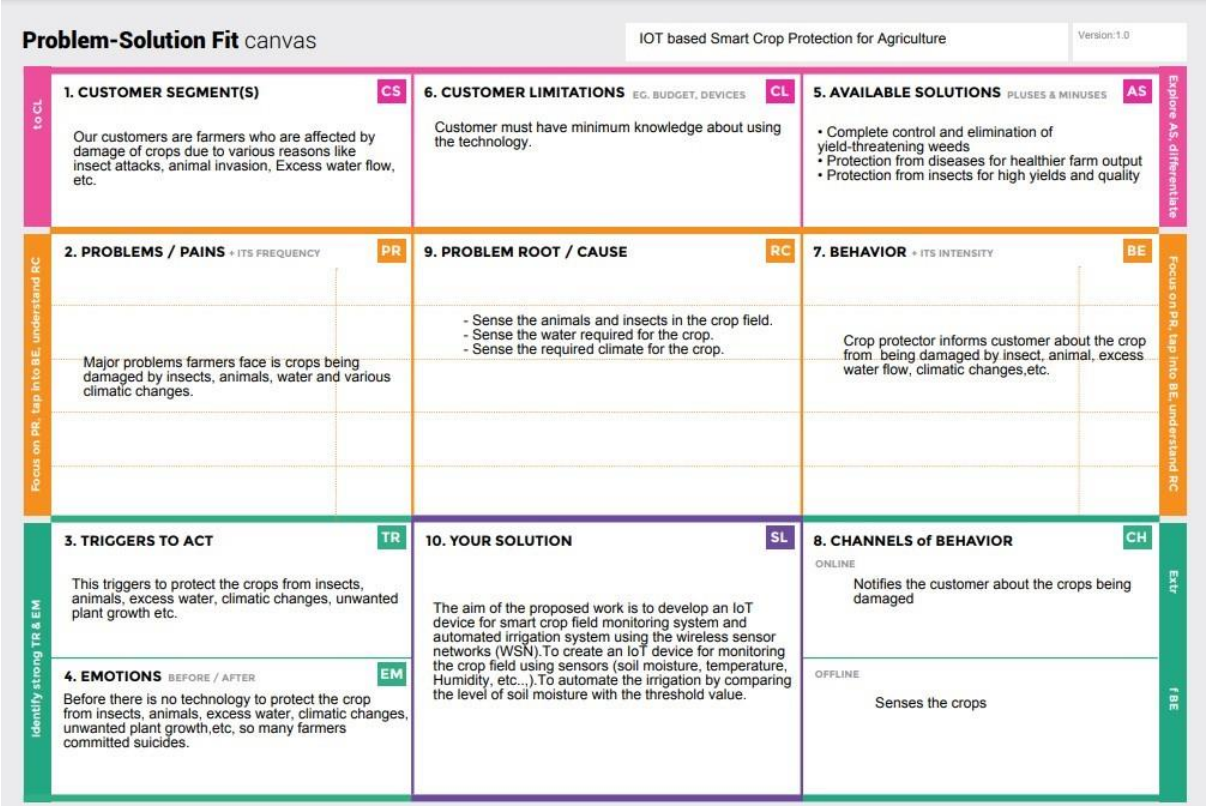


3.3 PROPOSED SOLUTION:

S.NO.	Parameter	Description
1.	Problem Statement. (Problem to be solved)	✓ Crops are not irrigated properly due toinsufficient labour forces. ✓ Improper maintenance of crops against variousenvironmental factors such as temperature climate, topography and soil quantity which results in crop destruction. ✓ Requires protecting crops from wild animals attacks birds and pests.

2.	Idea /Solution Description.	✓ Moisture sensor is interfaced with Arduino Microcontroller to measure the moisture level in soil and relay is used to turn ON & OFF the motorpump for managing the excess water level. It will be updated to authorities through IOT. ✓ Temperature sensor connected to microcontrolleris used to monitor the temperature in the field. ✓ Image processing techniques with IOT is followedfor crop protection against animal attack.
3.	Novelty / Uniqueness.	✓ Automatic crop maintenance and protection usingembedded and IOT Technology.
4.	Social Impact / Customer satisfaction.	✓ This proposed system provides many facilities which helps the farmers to maintain the crop field without much loss.
5.	Business Model (Revenue Model).	✓ This prototype can be developed as product withminimum cost with high performance.
6.	Scalability of the solution	✓ This can be developed to a scalable product by usingsolution sensors and transmitting the data through Wireless Sensor Network and Analysing the datain cloud and operation is performed using robots.

3.4 PROBLEM SOLUTION FIT:



CHAPTER 4

REQUIREMENT ANALYSIS

Requirements analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and Non-functional requirements.

4.1 FUNCTIONAL REQUIREMENTS:

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	Interfacing with hardware	Interface the sensors with the software application so as to alert the farmers in case of any harm for crops
FR-4	Database Connection	Databases are retrieved from IBM Cloud ant
FR-5	Mobile Application	Alarm and motors can be accessed from the mobile app

4.2 NON-FUNCTIONAL REQUIREMENTS:

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements. They basically deal with issues like Portability, Security, Maintainability, Reliability, Scalability, Performance, Reusability, Flexibility.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The smart crop protection alerts the farmers in case of any obstacles and helps in protecting the crops
NFR-2	**Security**	Smart Agriculture can improve the farming practices and maintain sustainable production of crops especially by preventing the animals into the agricultural lands through IoT enabled devices.
NFR-3	**Reliability**	With a proper power supply, SD card and programming the processor should be able to run 24/7 for years. The SD card and power supply will likely wear out faster than the Pi. The possible reasons behind Raspberry Pi failure can be power breakdowns, SD card failures, and ineligible environments.
NFR-4	**Performance**	Usage of an SD card module that helps to store a specified sound to scare the animals. Crop damage due to animal attack can be sensed. Network and Design Evaluation
NFR-5	**Availability**	Agriculture for different variety of crops is

		based on the monsoon changes, indoor and outdoor climatic temperatures, availability of rainfall and irrigationmethods.
NFR-6	**Scalability**	The product shall be made available to everyone especially in remote areas for better efficiency of crop yield with the better safety of crops as well asthe farmers.

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS:

A data flow diagram (DFD) maps out the flow of information for any process or system.It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show datainputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that digprogressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That's why DFDs remain so popular after allthese years. While they work well for data flow software and systems, they are less applicablenowadays to visualizing interactive, real-time or database-oriented software or systems.

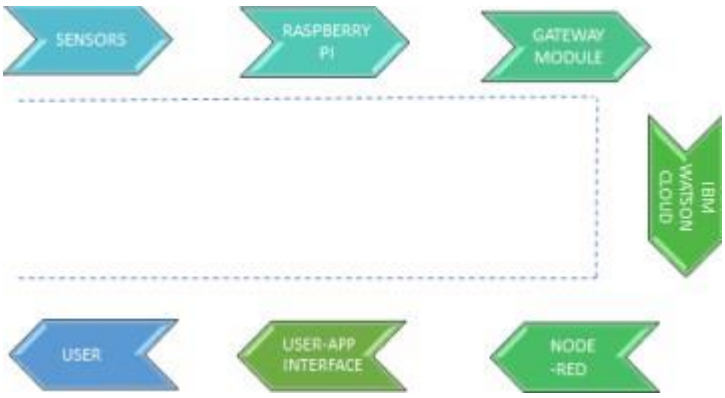


Fig 5.1 Data Flow

5.2 SOLUTION &TECHNICAL ARCHITECTURE:

Solution Architecture:

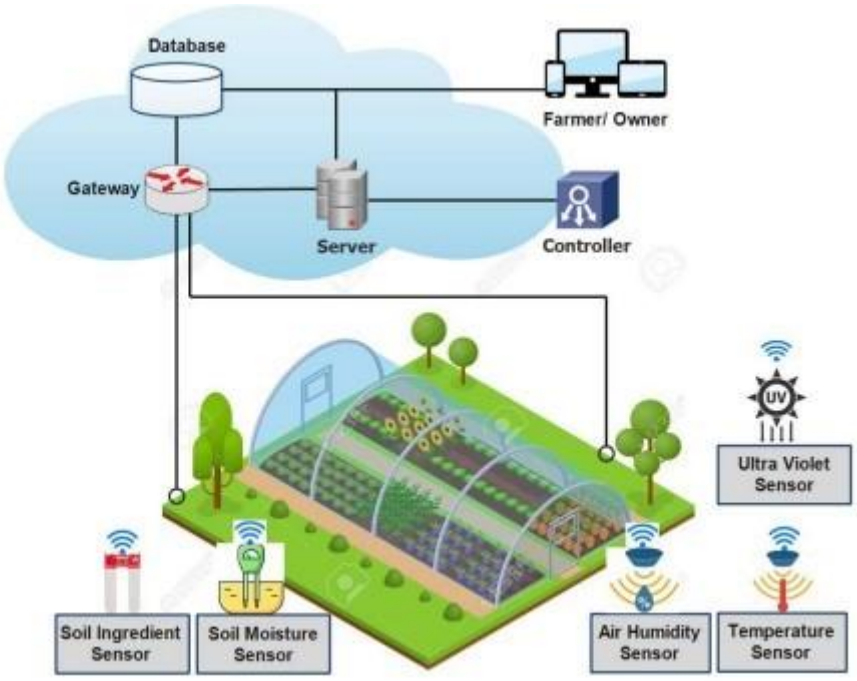
Solution architecture is a complex process – with many sub-processes – that bridgesthe gap between business problems and technology solutions. Its goals are to:

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

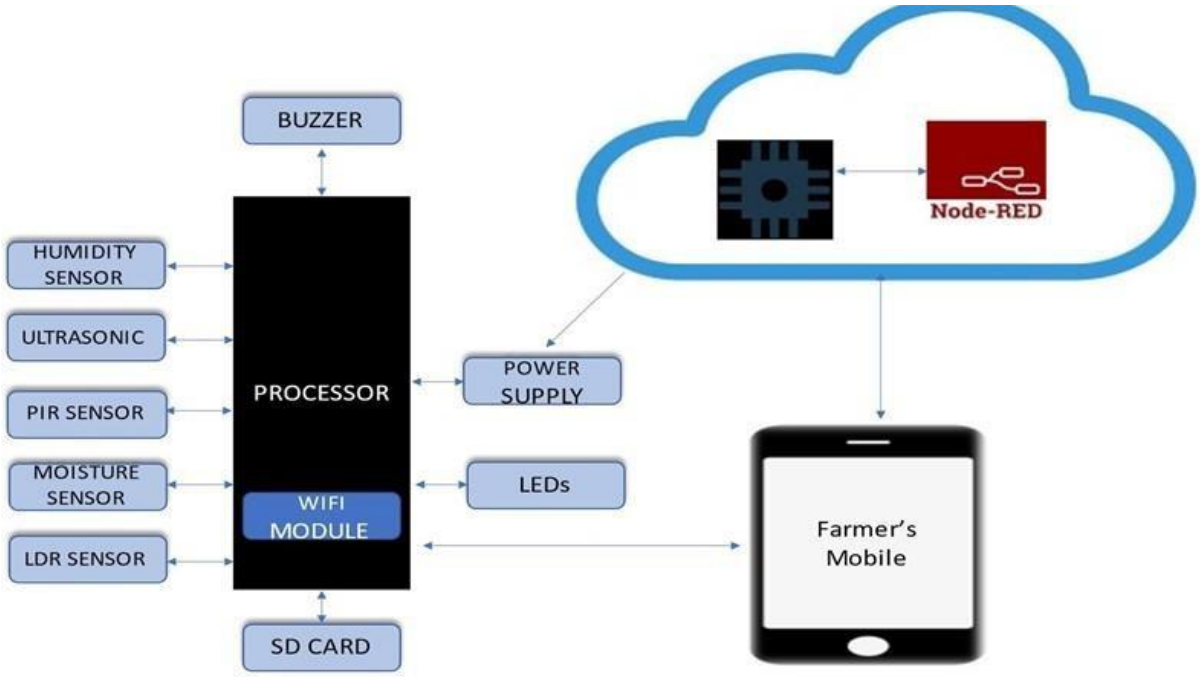
• Describe the structure, characteristics, behaviour, and other aspects of the software to projectstakeholders.

• Define features, development phases, and solution requirements.

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



Technical Architecture:



5.3 USER STORIES:

User Type	Functional Requiremet (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Farmer)	Maintaining Fields	USN-1	As a user, I can monitor the growth of crops and protect the crops against animals	I can maintain the fields with less labor	High	Sprint-1
	Analyzing Problems	USN-2	As a user, I collect the required information about the problems on agriculture fields	I can ask my field owner directly.	Low	Sprint-2
		USN-3	As a user, I can monitor the moisture level in soil and solve the problems by using Smart IOT System	I can take remedial action immediately	High	Sprint-1
Project Designers	Identifying the problem and	USN-4	As a user, I can sense the water level	I can perform this	Medium	Sprint-1

	Provide solutions		and flame in the field using sensor and monitor using IOT	actions via IoT.		
		USN-5	As a user, I can make services for Irrigation, pesticides, Fertilization, and Soil preparation	I can solve this problem using IOT	High	Sprint-1
			As a user, I can monitor the field against animal attacks using a camera interface module and appropriate actions can be taken	I can monitor the field continuously.	Medium	Sprint-2
Customer (Field Maintainer)	Problem solutions	USN-6	As a user, areas can be monitored from a remote place	Checking Process	Medium	Sprint-3

	Application	USN-7	As a user, I can respond to the problems in the fields immediately	Continuous monitoring and remedial actions.	Medium	Sprint-3
	Final Process	USN-8	This proposed smart IOT-based crop protection device is found to be cost effective and efficient	I can take necessary action if required.	Medium	Sprint-4

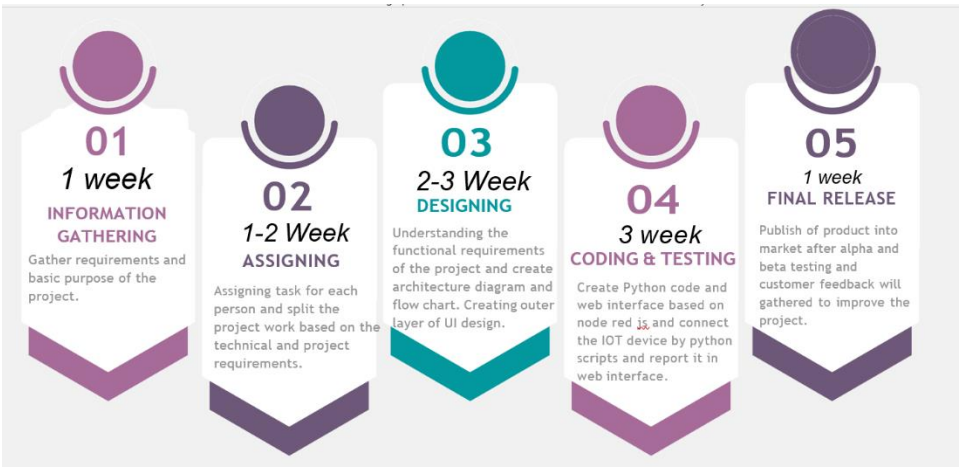
CHAPTER 6

PROJECT PLANNING &SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION:

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. Sprint planning is done in collaboration with the whole scrum team.

The sprint is a set period of time where all the work is done. However, before leap intoaction it is necessary to set up the sprint. It need to decide on how long the time box is going to be, the sprint goal, and where it is going to start. The sprint planning session kicks off the sprint by setting the agenda and focus. If done correctly, it also creates an environment wherethe team is motivated, challenged, and can be successful.



S.NO	Activity Title	Activity Description	Duration
1	Understanding the project requirement	Assign the team members & create the repository in GitHub. Assign the task to each members and teach how to use and open access the GitHub and IBM Career Education.	1 Week
2	Starting of Project	Advice student to attend classes of IBM portals create and develop an rough diagram based on the project description and gather information of IOT and IBM project.	1 Week
3	Attend Classes	Team members & team lead must watch and learn from classes provided by IBM and Nalaya Thiran and must gain access of MIT License for their project.	1 Week
4	Budget and scope of project	Budget & analyse the use of IOT in the project and discuss with the team for budget prediction to predict the favorability of the customer to buy the product for efficient use of the product among the environment.	1 Week

6.2 SPRINT DELIVERY SCHEDULE:

Sprint	Functional Requirement (Epic)	User Story Number	UserStory/Task	Story Points (40)	Priority (Low to High)	TEAM MEMBERS
Sprint-1	Registration	USN-1	As a user, I can register for the required dataset by entering my email, password, and confirming my password.	3	high	Tharun Kumar
Sprint-1		USN-2	As a user, I will receive confirmation email and the SMS once I have registered for the application	2	high	Tharun Kumar
Sprint-2	Cloud services	USN-3	As a user, I can register for the application through Facebook or any social media	1	low	Kaviarasan
Sprint-4		USN-4	As a user, I can register for the application through Gmail/web service	2	medium	Bhoobalan
Sprint-3	Login	USN-5	As a user, I can log into the application	4	high	Sanjay Kumar

			network by entering email & password			
Sprint-2	Preprocessing	USN-6	As a farmer, the user must be able to find the system easy to access so pre-processes and other task must be perfect.	3	high	Bhoobalan
Sprint-1	Preprocessing	USN-7	To collect various sources of animal threats and keep developing a dataset.	3	medium	Tharun Kumar
Sprint-4	Collecting Dataset	USN-8	To integrate the available dataset and keep improving the accuracy of finding animals	2	high	Kaviarasan
Sprint-3	Integrating	USN-9	To find and use appropriate compiler to run and test the data so that we can implement our program	1	low	Sanjay Kumar
Sprint-2		USN-10	Request P.S.V College Of Engineering	1	low	Bhoobalan

			And Technology to deploy the project in our campus and test			
Sprint-1		USN-11	As programmer, we need to train our data perfectly so that the program runs smoothly	3	high	Tharun Kumar
Sprint-3		USN-12	Train the data using out available services and IBM dataset from server and improve that	2	medium	Sanjay Kumar
Sprint-4	Coding	USN-13	To modify the code according to our program and improve the efficiency of that code	4	high	Kaviarasan
Sprint-2		USN-13	To improve performance	1	low	Sanjay Kumar
Sprint-2	Record	USN-5	To record the data and plot the graph to show the characteristics officially	4	high	Sanjay Kumar
Sprint-1	Planning	USN-4	Plan the programming	3	medium	Tharun Kumar

			language and feasibility			Kaviarasan
Sprint-4		USN-14	Demonstrate the working and improve accuracy overall	2	low	Kaviarasan

Project Tracker, Velocity & Burndown Chart:

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	5 Days	20 Oct 2022	24 Oct 2022	20	21 Oct 2022
Sprint-2	20	5 Days	25 Oct 2022	29 Oct 2022	20	27 Oct 2022
Sprint-3	20	5 Days	31 Oct 2022	4 Nov 2022	20	2 Nov 2022
Sprint-4	20	7 Days	5 Nov 2022	11 Nov 2022	20	8 Nov 2022

Velocity:

We have a 23-day sprint duration, and the velocity of the team is 20 (points per sprint).

To Find: Calculate the team's average velocity (AV) per iteration unit (story points per day)

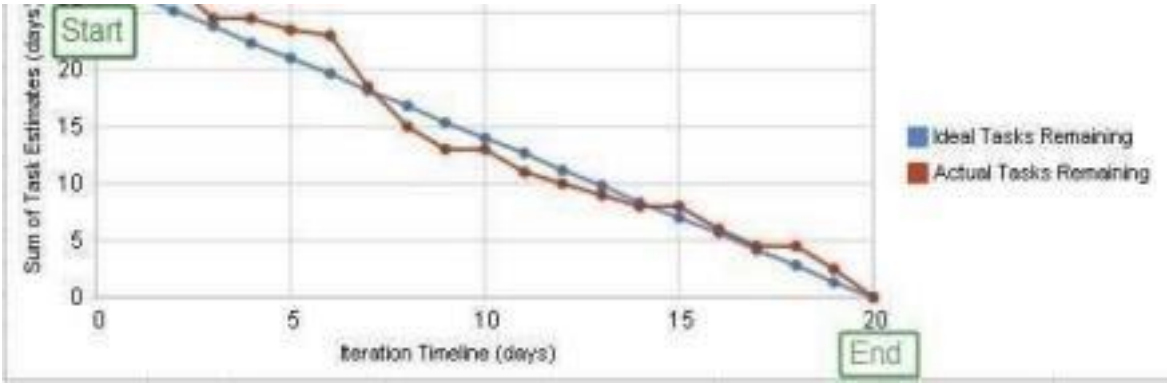
$$\frac{23}{20} AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

1.15

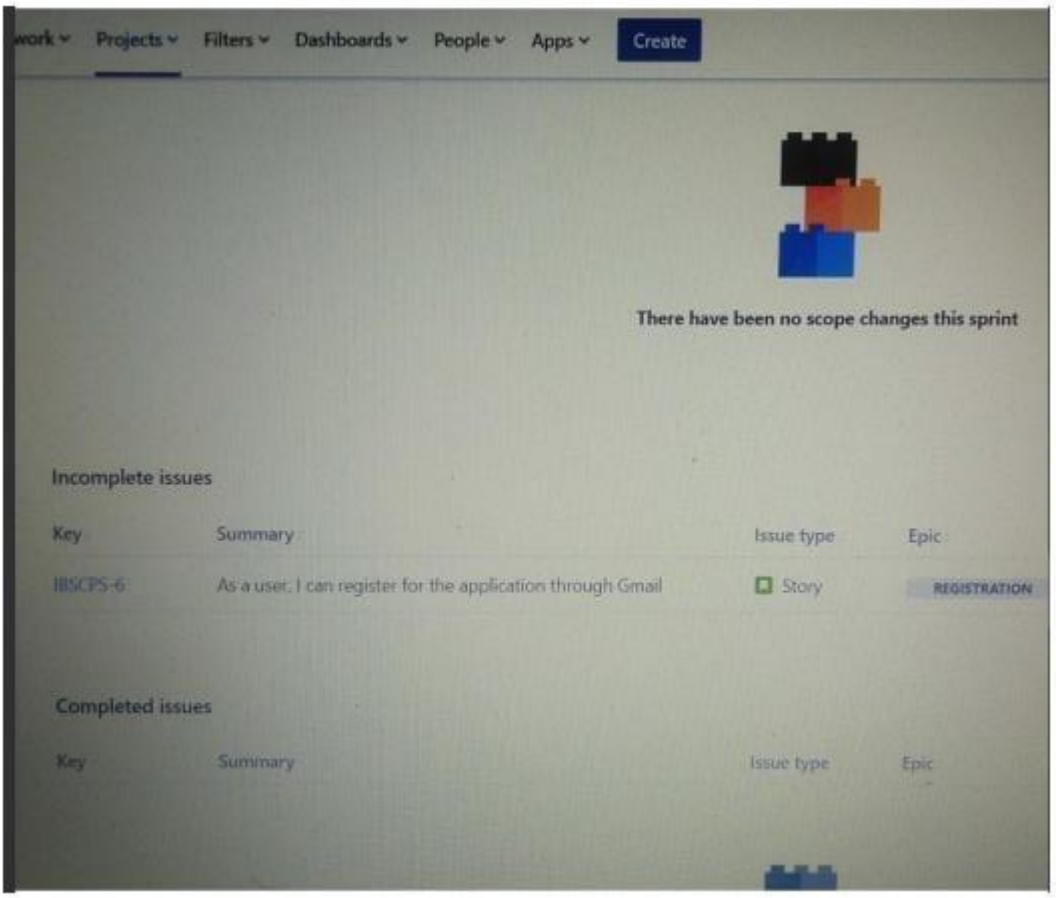
Burndown Chart:

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

Project: IoT Based Smart Crop Protection System for Agriculture



6.3 REPORTS FROM JIRA:



CHAPTER 7

CODING & SOLUTIONING

7.1 FEATURE 1:

PROJECT SIMULATION IN WOKMI:

```
#include <Wire.h>               //Includes the library for connections
#include <ESP32Servo.h>         //Includes the library for Servo motor
#include <LiquidCrystal_I2C.h> //Includes the library for LED
#include <DHTesp.h>             //Includes the library for DHT22 sensor

// WiFi libraries:
#include <WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>

#define ORG "oqy2ad" // Organization ID of IBM Cloud
#define DEVICE_TYPE "ESP32"
#define DEVICE_ID "NodeMCU"
#define TOKEN "123456789"

// Publishing Event in Watson IOT platform:
char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; //
oqy2ad.messaging.internetofthings.ibmcloud.com
char pubTopic[] = "iot-2/evt/status1/fmt/json";
char subTopic[] = "iot-2/cmd/command/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

const char *ssid = "Wokwi-GUEST";
const char *password = "";

const int led = 4;
const int servoPin = 2;
const int echo = 12;
const int trig = 14;
const int r = 27;
const int g = 26;
const int b = 25;
const int y = 33;
const int sec = 0;
const int dht = 15;
long lastMsg = 0;
Servo s;
String data3;

void callback(char *subTopic, byte *payload, unsigned int payloadLength);
```

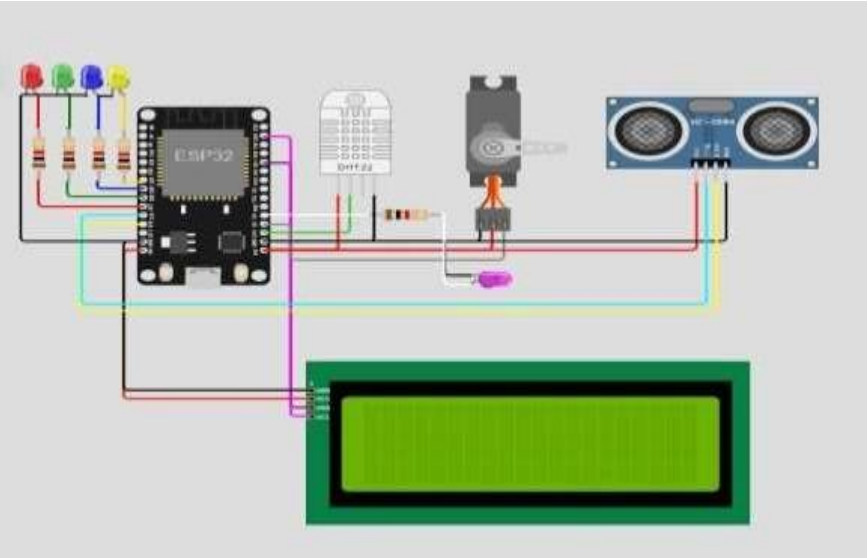
```
#define I2C_ADDR 0x27
#define LCD_COLUMNS 20
#define LCD_LINES 4

LiquidCrystal_I2C lcd(I2C_ADDR, LCD_COLUMNS, LCD_LINES);
DHTesp dhtSensor;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback, wifiClient);

void setup()
{
    Serial.begin(115200);
    Wire.begin();
    pinMode(A0, INPUT);            // Temperature Sensor
    pinMode(trig, OUTPUT);         // Ultra sonic Trigger
    pinMode(echo, INPUT);          // Ultra sonic Echo
    pinMode(b, OUTPUT);            // BLUE light for LED
    pinMode(g, OUTPUT);            // GREEN light for LED
    pinMode(r, OUTPUT);            // RED light for LED
    pinMode(y, OUTPUT);            // YELLOW light for LED
    pinMode(led, OUTPUT);          // LED for Motor Indication
    s.attach(servoPin, 500, 2400); // Servo Motor
    lcd.init();                    // LCD Display
    lcd.setBacklight(0);
    dhtSensor.setup(dht, DHTesp::DHT22);

    Serial.println();
    // Connecting the ESP32 with WiFi:
    Serial.print("Connecting to ");
    Serial.print(ssid);
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password, 6);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");

    Serial.print("WiFi connected, IP address: ");
    Serial.println(WiFi.localIP());

    // Connecting to IBM Cloud:
    if (!client.connected())
    {
        Serial.print("Reconnecting client to ");
        Serial.println(server);
        while (!client.connect(clientId, authMethod, token))
        {
            Serial.print(".");
            delay(500);
        }
```

```
        client.setCallback(callback);
        if (client.subscribe(subTopic))
        {
            Serial.println("Subscription to cmd OK");
        }
        else
        {
            Serial.println("Subscription to cmd FAILED");
        }

        Serial.println("Bluemix connected");
        Serial.println("");
    }
}

float readDistanceCM()
{
    digitalWrite(trig, LOW);
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);
    int duration = pulseIn(echo, HIGH);
    return duration * 0.034 / 2;
}

void loop()
{

    client.loop();
    long now = millis();

    // Temperature:
    TempAndHumidity data = dhtSensor.getTempAndHumidity();
    float t = data.temperature;
    float h = data.humidity;
    Serial.println("Temperature: " + String(t) + " degrees");
    Serial.println("Moisture: " + String(h) + " %");

    // Ultrasonic sensor:
    float distance = readDistanceCM();
    Serial.print("Measured distance: ");
    Serial.println(readDistanceCM());

    // Soil Moisture:
    int soil = random(0, 100); // As there is no soil moisture sensor, random
function is used for it.
    Serial.println("Soil Moisture: " + String(soil) + "%");

    // LCD Display:
    lcd.setBacklight(1);
```

```
    lcd.clear();

    digitalWrite(b, 0);
    digitalWrite(g, 0);
    digitalWrite(r, 0);
    digitalWrite(y, 0);

    // Conditions:
    /*If the temperature is Greater than 30 and less than 40 and also humidity
or soil moisture is greater than 30 and
    less than 70 then the GREEN light will be turned ON indicating the Normal
condition */
    if (t > 30 & t < 40 && h > 30 & h<70 | soil> 30 & soil < 70)
    {
        digitalWrite(g, 1);
        s.write(90);
        Serial.println("Normal Condition");
        Serial.println("Water Partially Flows");
        lcd.setCursor(3, 1);
        lcd.println("ON Motor");
        delay(1000);
        lcd.clear();
    }

    /*If the temperature is greater than 40 OR the humidity or soil moisture is
less than 30, then the RED light will
    be turned ON indicating the Hot or Low humid condition */
    else if (t > 40 | h < 30 | soil < 30)
    {
        digitalWrite(r, 1);
        s.write(180);
        Serial.println("High Temperature or Low humid condition");
        Serial.println("Water Fully Flows");
        lcd.setCursor(3, 1);
        lcd.println("ON Motor");
        delay(1000);
        lcd.clear();
    }

    /*If the level of water is MORE in the field it will be indicated by
distance sensor for less than
    10cm and soil moisture is greater than 70, then the YELLOW light will be
turned ON indicating the high water level */
    else if (distance<10 & soil> 70)
    {
        digitalWrite(y, 1);
        s.write(0);
        Serial.println("Water Does Not Flow");
        Serial.println("Water is Full in the field");
        lcd.setCursor(2, 1);
        lcd.println("Drain the water");
        delay(1000);
```

```
        lcd.clear();
    }

    /*If the temperature is less than 30 OR the humidity or soil moisture is
greater than 70, then the BLUE light will
    be turned ON indicating the Cool or High humid condition */
    else if (t<30 | h> 70 | soil > 70)
    {
        digitalWrite(b, 1);
        s.write(0);
        Serial.println("Cool Temperature or High Humid Condition");
        Serial.println("Water Does Not Flow");
        lcd.setCursor(3, 1);
        lcd.println("OFF Motor");
        delay(1000);
        lcd.clear();
    }

    else
    {
        digitalWrite(b, 1);
        s.write(0);
        Serial.println("Water Does Not Flow");
    }

    // Sending payload:
    Serial.println("");
    if (now - lastMsg > 1000)
    {
        lastMsg = now;

        // Payload for Parameters:
        String payload = "{\"Name\":\"" DEVICE_ID "\"";
        payload += ",\"Temperature\":";
        payload += t;
        payload += ",\"Humidity\":";
        payload += h;
        payload += ",\"Distance\":";
        payload += distance;
        payload += ",\"SoilMoisture\":";
        payload += soil;
        payload += "}";
        Serial.print("Sending payload: ");
        Serial.println(payload);
        Serial.println("");
        if (client.publish(pubTopic, (char *)payload.c_str()))
        {
            Serial.println("Publish ok for payload");
        }
        else
        {
            Serial.println("Publish failed");
```

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

    lcd.setCursor(1, 0);
    lcd.print("Temp: ");
    lcd.print(t);
    lcd.print(" degree");
    lcd.setCursor(1, 1);
    lcd.print("Humidity: ");
    lcd.print(h);
    lcd.print(" %");
    lcd.setCursor(1, 2);
    lcd.print("Distance: ");
    lcd.print(distance);
    lcd.print(" cm");
    lcd.setCursor(1, 3);
    lcd.print("Soil Moisture: ");
    lcd.print(soil);
    lcd.print(" %");
    delay(5000);
    lcd.clear();
}

void callback(char *subTopic, byte *payload, unsigned int payloadLength)
{
    Serial.println("----------------------Callback!!---------------------");
    Serial.print("Callback invoked for topic:");
    Serial.println(subTopic);
    for (int i = 0; i < payloadLength; i++)
    {
        data3 += (char)payload[i];
    }
    Serial.println("Data:" + data3);
    if (data3 == "motoron")
    {
        Serial.println("Motor is ON");
        digitalWrite(led, 1);
    }
    else
    {
        Serial.println("Motor is Off");
        digitalWrite(led, 0);
    }
    data3 = "";
    Serial.println("-----------------------------------------------------");
}
```



7.2 FEATURE 2:

PROJECT SIMULATION IN TINKERCAD:

```
#include <Wire.h>

#include <Servo.h>

#include <Adafruit_LiquidCrystal.h>


Servo s;

int e = 4;

int t = 5;

int r = 12;

int g = 11;

int b = 10;

int sec = 0;

int Sensor = 0;

int data = 0;

int motorPin = 9;

Adafruit_LiquidCrystal lcd(0);


void setup()
```

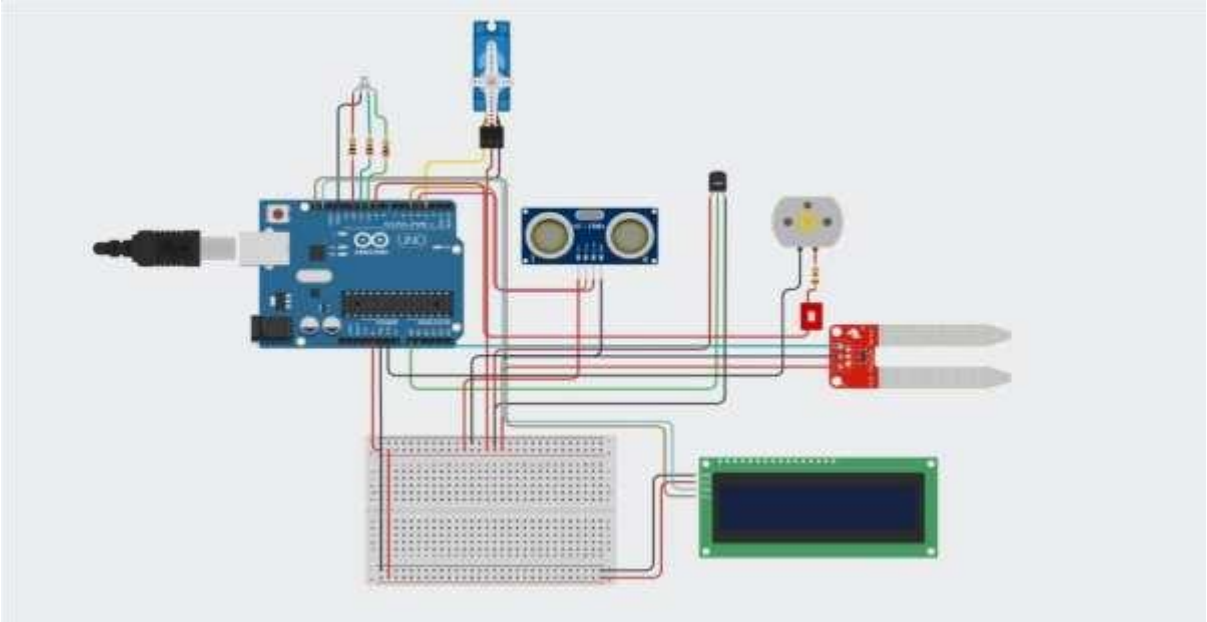
```
{
    Wire.begin();
    pinMode(A0,INPUT);  //Temperature Sensor
    pinMode(A1,INPUT);  //Soil Moisture Sensor
    pinMode(t,OUTPUT);    //Ultra sonic Trigger
    pinMode(e,INPUT);     //Ultra sonic Echo
    pinMode(b,OUTPUT);  //GREEN light for LED
    pinMode(g,OUTPUT);  //BLUE light for LED
    pinMode(r,OUTPUT);  //RED light for LED
    pinMode(motorPin, OUTPUT); //DC motor
    s.attach(3);               // Servo Motor
    lcd.begin(16, 2);                //LCD 16x2 Display
    lcd.setBacklight(0);
    Serial.begin(9600);
}

float readDistanceCM(){
    digitalWrite(t, LOW);
    delayMicroseconds(2);
    digitalWrite(t, HIGH);
    delayMicroseconds(10);
    digitalWrite(t, LOW);
    int duration = pulseIn(e, HIGH);
    return duration * 0.034 / 2;
}

void loop(){
    //Soil Moisture:
```

```
    Sensor = analogRead(A1);          //Reads data from Soil Moisture sensor
    data = map(Sensor,0, 1023, 0, 100);   //Low analog value indicates HIGH moisture level and High analog value indicates LOW moisture level
    //data = map(analogValue,fromLOW,fromHIGH,toLOW,toHIGH)
    Serial.print("Soil Moisture value:");
    Serial.println(data);
    //'data = 0' indicates wet and 'data = 100' indicates dry

    //Temperature:
    double a = analogRead (A0);     //Reads data from Temperature sensor
    double t = (((a/1024)*5)-0.5)*100;
    Serial.print("Temperature value:");
    Serial.println(t);

    //Ultrasonic sensor:
    float distance = readDistanceCM();
    Serial.print("Measured distance: ");
    Serial.println(readDistanceCM());

      //LCD Display:
      lcd.setBacklight(1);
      lcd.clear();

    //Conditions:
    if (t>40 & t<50){
        digitalWrite(b,0);
        digitalWrite(g,1);
        digitalWrite(r,0);
```

```
    s.write(90);
    digitalWrite(motorPin, HIGH);
    Serial.println("Water Partially Flows");
}

else if (t>50){
    digitalWrite(b,1);
    digitalWrite(g,1);
    digitalWrite(r,0);
    s.write(180);
    digitalWrite(motorPin, HIGH);
    Serial.println("Water Fully Flows");
}

else if (t>30 & data<30){
    digitalWrite(b,1);
    digitalWrite(g,1);
    digitalWrite(r,0);
      s.write(90);
    digitalWrite(motorPin, HIGH);
    Serial.println("Water Partially Flows");
}

else if (data<50){
    digitalWrite(b,0);
    digitalWrite(g,0);
    digitalWrite(r,1);
    s.write(90);
```

```
    digitalWrite(motorPin, HIGH);
    Serial.println("Water Partially Flows");
}

else if (distance < 10){
    digitalWrite(b, 0);
    digitalWrite(g, 0);
    digitalWrite(r, 1);
    s.write(0);
    digitalWrite(motorPin, LOW);
    Serial.println("Water Does Not Flow");
    lcd.clear();
        lcd.println("Drain the water");
}

else{
    digitalWrite(b,1);
    digitalWrite(g,0);
    digitalWrite(r,0);
    s.write(0);
    digitalWrite(motorPin, LOW);
    Serial.println("Water Does Not Flow");
}

  lcd.setCursor(0,0);
  lcd.print("Temp:");
lcd.println(t);
  lcd.println("degree");
```

```
      lcd.setCursor(0,1);
      lcd.print("Soil Moisture:");
    lcd.println(data);
      lcd.println("%");

    Serial.println("-----------------------------------------------------");
    delay(1000);
}
```



CHAPTER 8

TESTING

8.1 TEST CASES:

Section	Total Cases	Not Tested	Fail	Pass
Temperature and Humidity Sensor	20	0	1	20
Ultrasonic Sensor	10	0	1	10
Soil Moisture	30	0	0	30
Wi-Fi Module	6	0	1	6
Transmission Of Data To IBM Cloud	5	0	2	5

8.2 USER ACCEPTANCE TESTING:

1. **Purpose Of Document**

   The purpose of this document is to briefly explain the test coverage and open issues of the Smart Farmer project at the time of the release to User Acceptance Testing (UAT).

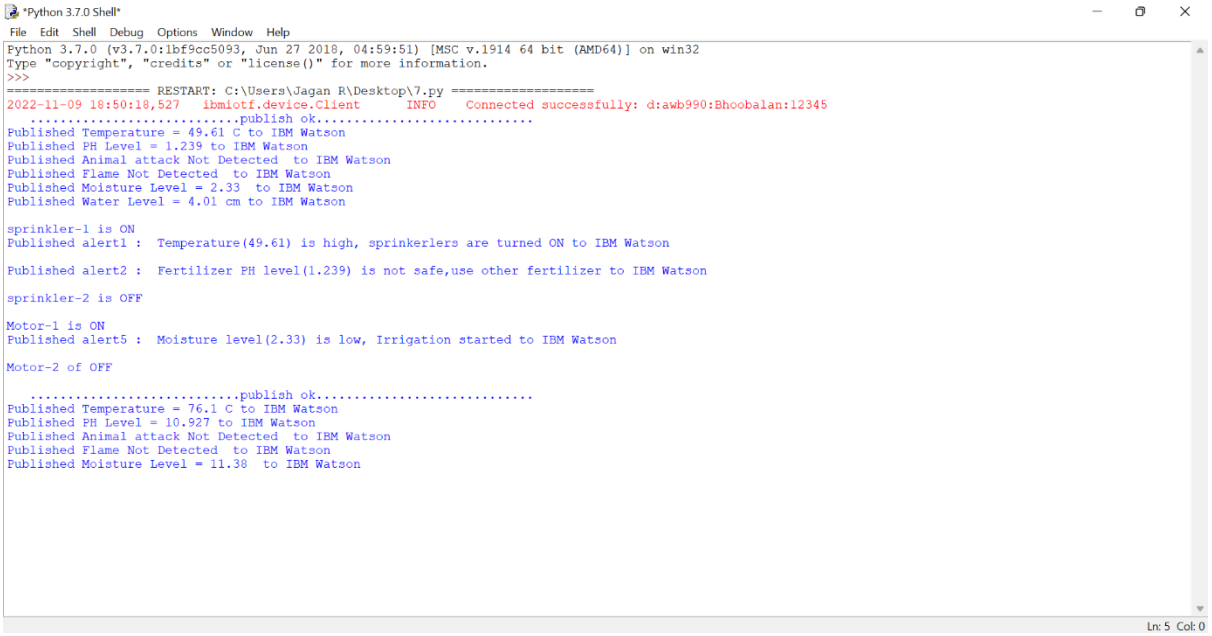
2. **Defect Analysis**

   This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

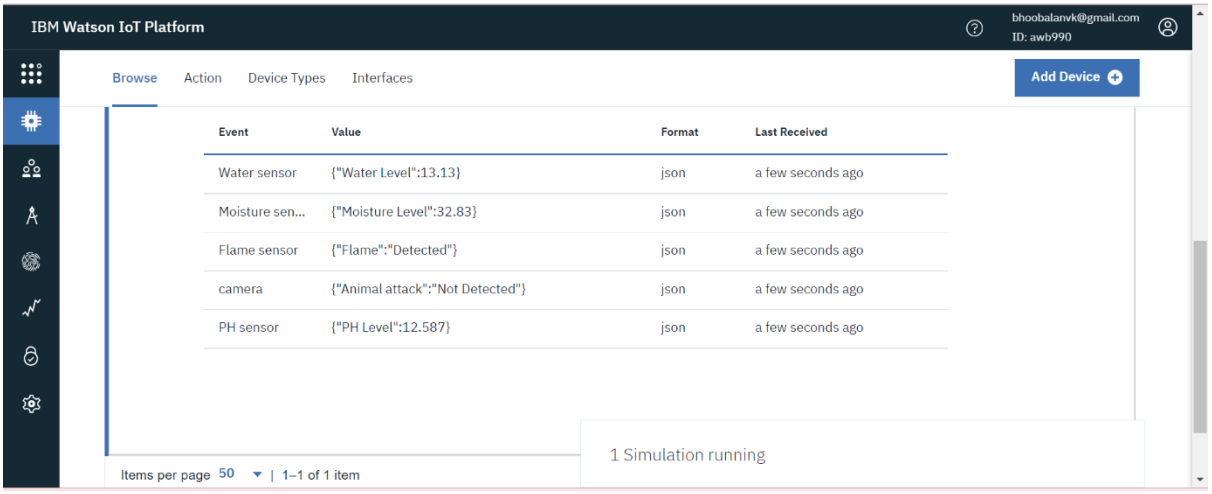
CHAPTER 9

RESULTS

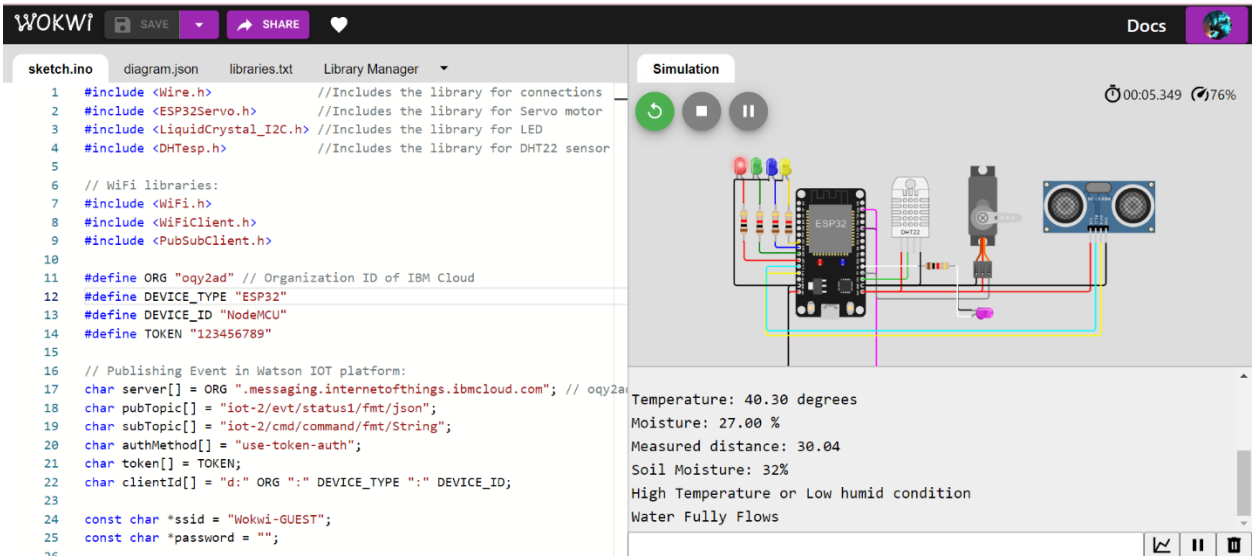
PYTHON CODE:



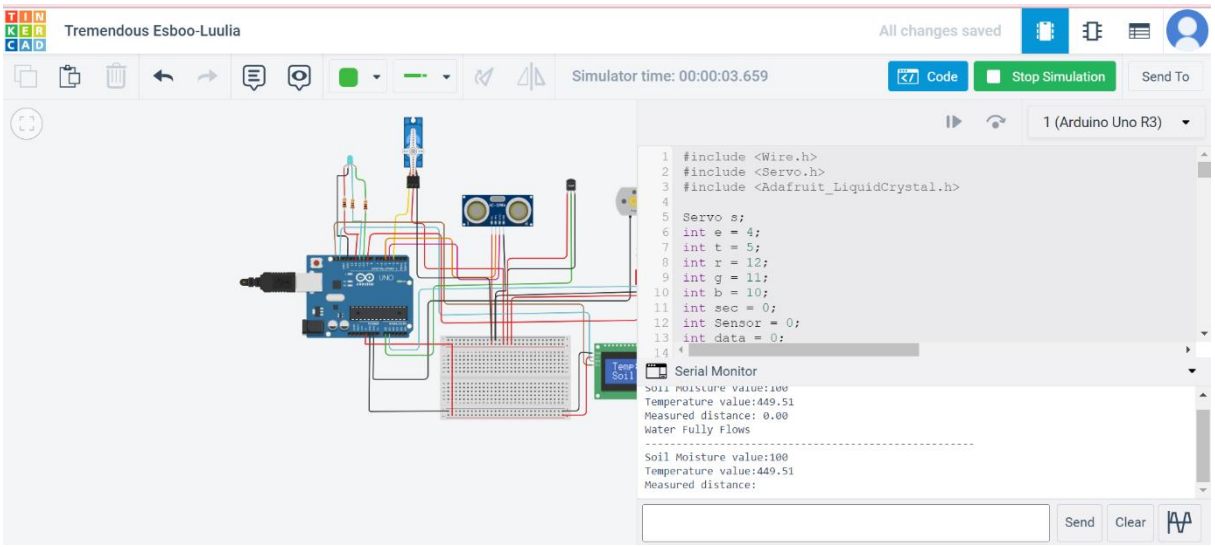
IBM WATSON IOT PLATFORM:



WOKMI:



TINKERCAD:



CHAPTER 10
ADVANTAGES & DISADVANTAGES

Advantage:

This proposed work successfully designs a working prototype of an integrated IoT hardware with an platforms like Twilio that fulfils the following functions of timer-based irrigation control, real-time monitoring of farm data and prediction of the weather condition ofthe crops.

It also integrated with the function of warning the activity of animals and birds in the fields to the users. The real-time data from the farm such as soil moisture are collected from the environment using the sensors that is interfaced respectively.

Another novel feature added in this model is weather API that notifies the weather condition when it goes beyond the expected weather condition. It also comes up with the usageof Twilio ,a platform which is provides programmable communication tools for making and receiving phone calls ,sending and receiving messages and performing other communication functions using its web service APIs.

Here a message is sent to the user when animal is found in the crop field. This proposedprototype system uses the Open Weather map API which provides the real time weather condition for any geographical location.

This system uses the Google Sheets which acts as a Database system that collects the sensor data and store it for future reference to the users.

Disadvantage:

The use of technology in farming and agriculture making it smart agriculture, is of course, a good initiative and a much-needed one with the present increasing demand in the foodsupply.

But there is the chance where this proposed smart farming protection system will require certain skill sets in particular in order to understand and operate the equipment. In the case of equipment computer-based intelligence for running the devices, it is highly unlikely that a normal farmer will be able to possess this knowledge or even develop them.

Farmers are not used to these high-end technologies. They do not understand computerlanguage or the artificial intelligence. For the smart agriculture, Internet of Things is essential which will require artificial intelligence and computer-based intelligence.

This cannot be balanced here. To overcome this challenge, the devices will have to bechanged in a dramatic fashion so as to make it understandable for farmers.

This also means that the devices should be somewhere in between where the technologyexperts and farmers can both communicate about it.

And also this system needs availability of internet continuously. Rural part of most of the developing countries do not fulfil this requirement. Moreover internet connection is slower.

CHAPTER 11

CONCLUSION

Smart farming is a modern farming management concept with IoT technology to increase the productivity in agriculture.

With the use of smart farming, users can effectively monitor the crop field the quality and quantity of their crops.

Users cannot be physically present on the field 24 hours a day. In addition, the farmersmay not have the knowledge to use different tools to measure the ideal environmentalconditions for their crops.

IoT provides them with the automated system, which can function without any humansupervision and can notify them to make proper decision to deal with different kind of problemsthey may face during farming.

It has the capability to reach and notify the farmer even if farmer is not on the field, which can allow farmer to manage more farmland, thus improving their production.

Thus, we can conclude that this IoT based smart crop protection system will definitelyhelp users in farmland to effectively monitor their crops with the user-friendly platforms and other alert means.

CHAPTER 12

FUTURE SCOPE

The proposed work system is a successful working prototype that fulfils to protect cropsfrom the intrusion of animals and birds.

This system will helps the users to monitor the temperature and to notify the weather conditions.

This system assuredly assists the users to know about the soil moisture level. And the IoT based smart crop protection system implemented here brings a naval approach crop protection system from animals.

This assures the early detection and prevention of incurring losses due to the damage of crops.

The following suggestions may be carried out in future implementation of the system; the smart crop prediction may be also carried out by considering the various factors like NPKcontent of the soil, UV radiation along with the tracking of the crop field location using GPS module system. The automated pest traps also be introduced using image recognition techniques and neural networks in smart protection system

CHAPTER 13

APPENDIX

SOURCE CODE:

Python Script:

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



#IBM Watson Device Credentials.
organization = "awb990"
deviceType = "Bhoobalan"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"



def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="sprinkler_on":
        print ("sprinkler is ON")
    else :
        print ("sprinkler is 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("Caught exception 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 = { 'Temperature' : temp_sensor }
     PH_data = { 'PH Level' : 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 data 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 Temperature = %s C" % temp_sensor,  "to IBM
Watson")



     success = deviceCli.publishEvent("PH sensor", "json", PH_data, qos=0)
     sleep(1)
     if  success:
          print ("Published PH Level = %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  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 of OFF")

         print("")







     #command recived by farmer

     deviceCli.commandCallback = myCommandCallback
```

```
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

GITHUB & PROJECT DEMO LINK:

GITHUB LINK: https://github.com/IBM-EPBL/IBM-Project-45006-1660727817

WOKMI LINK: https://wokwi.com/projects/348948915894092372

TINKERCAD LINK: **https://www.tinkercad.com/things/8UqGwhkCTsP-tremendous-esboo-luulia/editel**

PROJECT DEMO LINK: https://drive.google.com/drive/u/1/my-drive