SmartFarmer - IoT Enabled Smart Farming Application

Project Report

by

U.HARINI:953619106024 - (TEAM LEADER)

S.ABINAYA:953619106001 - (TEAM MEMBER 1)

V.ABIRAMI:953619106002 - (TEAM MEMBER 2)

R.DIVYASHREE :953619106014 - (TEAM MEMBER 3)

Abstract

Internet of Things (IoT) play crucial role in smart agriculture. Smart farming is an emerging concept, because IoT sensors capable of providing information about their agriculture fields. The pape feature of this paper includes monitoring temperature and humidity in agricultural field er aims making use of evolving technology i. e. IoT and smart agriculture using automation. Monitoring environmental factors is the major factor to improve the yield of the efficient crops. Ththrough sensors using CC3200 single chip. Camera is interfaced with CC3200 to capture images and send that pictures through MMS to farmers mobile using Wi-Fi. Agriculture is the primary occupation in our country for ages. But now due to migration of people from rural to urban there is hindrance in agriculture. So to overcome this problem we go for smart agriculture techniques using IoT. This project includes various features like GPS based remote controlled monitoring, moisture & temperature sensing, intruders scaring, security, leaf wetness and proper irrigation facilities. It makes use of wireless sensor networks for nothing the soil properties, and environmental factors continuously

Table OF Contents

1. INTRODUCTION

The advancement of science and technology, the global GDP has risen consistently. As a result, the presence of this reality has encouraged the development of smart farming, which use sensors and irrigation systems to manage crops as they grow. With sensor-based computer applications, more accurate information about the crop, soil, and environment may be gathered. It promotes high-quality process and raw materials throughout the entire product process. This is because utilising the Internet of Things in smarter agriculture makes it more competitive than traditional methods. Combined with IoT-based smart agriculture technologies, organic agricultural agriculture and family farming may see a benefit. It is the case that between agricultural producers and IoT technology, a digital breach has occurred, thus farmers are not vulnerable to IoT assaults. Sustainable use of water and input and treatment optimization will

allow farmers to produce more food while also preserving the environment. To include agriculture in smart use of natural resources, usage of technologies such as remote control, decision support tools, automated irrigation systems, frost avoidance, and fertilisation is required. These activities are supplied by IoT technologies, which provide devices like as hardware, intelligent apps, integration platforms, control procedures, operating systems, and cloud computing. The benefits of IoT and the Internet may be gained through the Cloud of Things, which combines IoT with cloud computing. Another requirement for the IoT is for it to provide society with information transparency. This work summarizes the current IoT-based agricultural tools and applications, which are broken down into distinct areas . The aim of this paper is to describe all these topics in detail, as well as discuss the advantages of using IoT-based technologies in agriculture. The yields obtained with less capital and labor have improved over time, with significant innovations having been made in human history. However, as long as the population rate is high, the demand and supply cannot balance, regardless of the periods. The numbers that have been estimated say that in 2050, the world population will be around 25 percent larger than it is now. In the vast majority of the growth predicted for the developed countries, it is expected that much of it will occur. Urbanization is expected to increase even more rapidly in the future, and about 70 percent of the world's population will be urban by 2050 (currently 49 percent). A further factor contributing to food demand is that income levels will be multiple times what they are now, which will lead to further increases in food demand particularly in developing countries [3,4]. Consequently, dietary preferences will change from wheat and grains to legumes and then to meat. To meet the increased demand for food due to an increasingly urbanized and wealthy population, food production must increase by a factor of two by the year 2050. This prediction is particularly important because of the current prediction of approximately 2.1 billion tons of annual cereal production and the prediction of a rise in meat production of over 200 million tons to meet the predicted demand of 470 million tons. Crop processing is playing an increasingly important role in industrial economies as well. On top of that, the bio energy demand began to increase in the food crops-based bio energy market. From the start of the twentieth century until now, only the manufacture of ethanol has used 110 million tons of coarse grains (approximately 10 percent of the world production). Food protection is in danger due to the rise in industrial and other uses of food crops for bio-fuel production. the scarcity of agricultural capital is making the demands even more onerous.

1.1. Project Overview

Agriculture is done in every country from ages. Agriculture is the science and art of cultivating plants. Agriculture was the key development in the rise of sedentary human civilization. Agriculture is done manually from ages. As the world is trending into new technologies and implementations it is a necessary goal to trend up with agriculture also. IOT plays a very important role in smart agriculture. IOT sensors are capable of providing information about agriculture fields. we have proposed an IOT and smart agriculture system using automation.

This IOT based Agriculture monitoring system makes use of wireless sensor networks that collects data from different sensors deployed at various nodes and sends it through the wireless protocol. This smart agriculture using IOT system is powered by Arduino, it consists of Temperature sensor, Moisture sensor, water level sensor, DC motor and GPRS module. When the IOT based agriculture monitoring system starts it checks the water level, humidity and moisture level. It sends SMS alert on the phone about the levels. Sensors sense the level of water if it goes down, it automatically starts the water pump. If the temperature goes above the level, fan starts. This all is displayed on the LCD display module. This all is also seen in IOT where it shows information of Humidity, Moisture and water level with date and time, based on per minute. Temperature can be set on a particular level, it is based on the type crops cultivated. If we want to close the water forcefully on IOT there is button given from where water pump can be forcefully stopped.

1.2. Purpose

Smart farming based on IoT technologies enables growers and farmers to reduce waste and enhance productivity ranging from the quantity of fertilizer utilized to the number of journeys the farm vehicles have made, and enabling efficient utilization of resources such as water, electricity, etc.IoT smart farming solutions is a system that is built for monitoring the crop field with the help of sensors (light, humidity, temperature, soil moisture, crop health, etc.) and automating the irrigation system. The farmers can monitor the field conditions from anywhere. They can also select between manual and automated options for taking necessary actions based on this data. For example, if the soil moisture level decreases, the farmer can deploy sensors to start the irrigation. Smart farming is highly efficient when compared with the conventional approach.
IoT have the potential to transform agriculture in many aspects and these are the main ones.
Data collected by smart agriculture sensors, in this approach of farm management, a key component are sensors, control systems, robotics, autonomous vehicles, automated hardware, variable rate technology, motion detectors, button camera, and wearable devices. This data can be used to track the state of the business in general as well as staff performance, equipment efficiency. The ability to foresee the output of production allows to plan for better product distribution.

2. LITERATURE SURVEY

Precision agriculture makes use of a range of technologies that include GPS services, sensors and big data to optimize crop yields. Rather than replace farmer expertise and gut feeling, ICT-based decision support systems, backed up by real time data, can additionally provide information concerning all aspects of farming at a level of granularity not previously possible.

This enables better decisions to be made, resulting in less waste and maximum efficiency in operations.The disciplines and skills now required for agriculture include computer-based imaging, GPS technology, science-based solutions, climate forecasting, technological solutions, environmental controls and more. Precision agriculture is sometimes known as ‘smart farming’, an umbrella term for easier comparison with other M2M based implementations such as smart metering, smart cities and so on. It is based on sensor technologies whose use is well established in other industries, e.g. Environmental monitoring for pollutants, eHealth monitoring in patients, buildings management for farm soil monitoring and so on.For all M2M implementations, IT systems gather, collate, analyze the data and present it in such a way as to initiate an appropriate response to the information received. For farmers and growers, a wide variety of information regarding soil and crop behavior, animal behavior, machine status, storage tank status emanating from remote sites is presented for action by the farmer

2.1.Existing problem

By adopting IoT in the agricultural sector we get numerous benefits, but still, there are challenges faced by IoT in agricultural sectors. The biggest challenges faced by IoT in the agricultural sector are lack of information, high adoption costs, and security concerns, etc. Most of the farmers are not aware of the implementation of IoT in agriculture. Major problem is that some of them are opposed to new ideas and they do not want to adopt even if it provides numerous benefits. The best thing that can be done to raise awareness of IoT’s impact is to demonstrate farmers the use of IoT devices like drones, sensors and other technologies and they could provide them ease at work and accompanied by real-world examples.

Even if the farmers adopt IoT technology they won’t be able to take benefit of this technology due to poor communication infrastructure. Farms are located in remote areas and are far from access to the internet. A farmer needs to have access to crop data reliably at any time from any location, so connection issues would cause an advanced monitoring system to be useless.

Equipment needed to implement IoT in agriculture is expensive. However sensors are the least expensive component, yet outfitting all of the farmers’ fields to be with them would cost more than a thousand dollars. Automated machinery cost more than manually operated machinery as they include cost for farm management software and cloud access to record data. To earn higher profits, it is significant for farmers to invest in these technologies however it would be difficult for them to make the initial investment to set up IoT technology at their farms.

2.2. References

We have refered the ibm videos to complete the project and we refered some websites for this report

https://www.geeksforgeeks.org/challenges-faced-by-iot-in-agricultural-sector/#:~:text=The%20biggest%20challenges%20faced%20by,implementation%20of%20IoT%20in%20agriculture.

https://link.springer.com/chapter/10.1007/978-3-030-23976-3_13#chapter-info

2.3. Problem Statement Definition

The traditional agriculture and allied sector cannot meet the requirements of modern agriculture which requires high-yield, high quality and efficient output. Thus, it is very important to turn towards modernization of existing methods and using the information technology and data over a certain period to predict the best possible productivity and crop suitable on the very particular land. The adoptions of access to high-speed internet, mobile devices, and reliable, low-cost satellites (for imagery and positioning) are few key technologies characterizing the precision agriculture trend. Precision agriculture is one of the most famous applications of IoT in the agricultural sector and numerous organizations are leveraging this technique around the world. Some products and services in use are VRI optimization, soil moisture probes, virtual optimizer PRO, and so on. VRI (Variable Rate Irrigation) optimization maximizes profitability on irrigated crop fields with topography or soil variability, improve yields, and increases water use efficiency. IoT has been making deep inroads into sectors such as manufacturing, health-care and automotive. When it comes to food production, transport and storage, it offers a breadth of options that can improve India's per capita food availability. Sensors that offer information on soil nutrient status, pest infestation, moisture conditions etc. which can be used to improve crop yields over time.

3 . IDEATION & PROPOSED SOLUTION

Smart Farming has enabled farmers to reduce waste and enhance productivity with the help of sensors (light, humidity, temperature, soil moisture, etc.) and automation of irrigation systems. Further with the help of these sensors, farmers can monitor the field conditions from anywhere. Internet of Things based Advanced Farming is highly efficient when compared with the conventional approach. The applications of intelligent Agriculture solutions not only targets conventional, large farming. With operations, but could also be new levers to uplift other growing or common trends in agricultural like organic farming, family farming (complex or small spaces, particular cattle and/or cultures, preservation of specific or high-quality varieties, etc.), and enhance highly transparent Farming.

3.1. Empathy Map Canvas

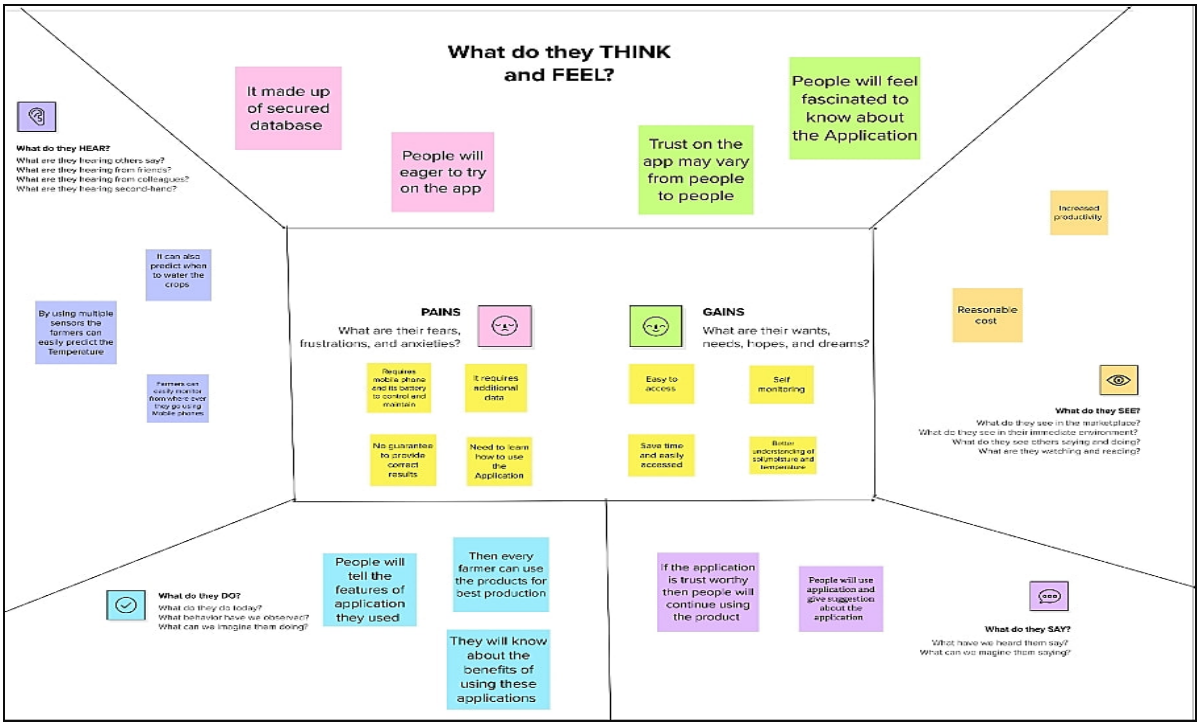
An **empathy map** is a collaborative visualization used to articulate what we know about a

particular type of user. It externalizes knowledge about users in order to

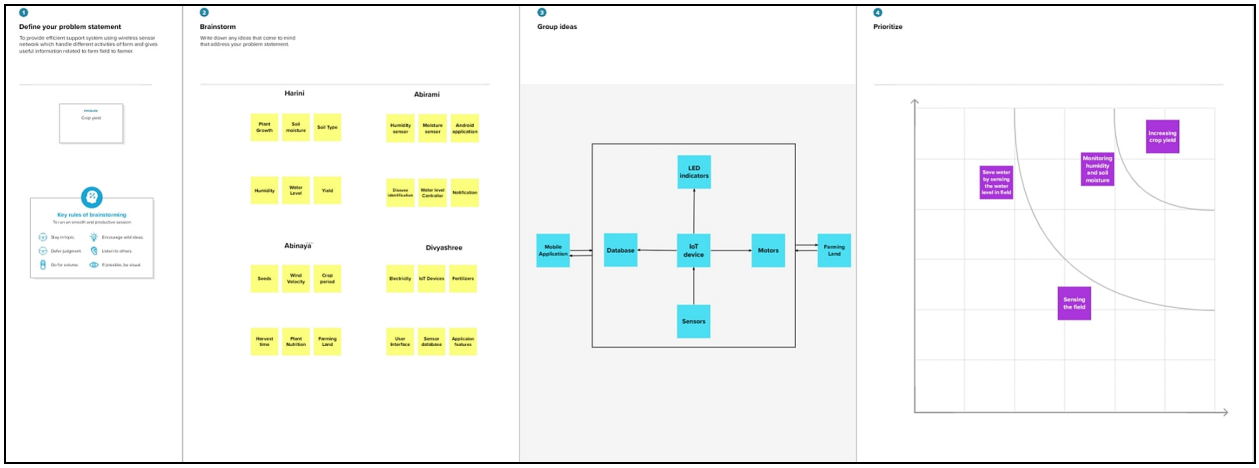
1) create a shared understanding of user needs, and

2) aid in decision making.

we have created the emapathy map for our project for this we have used mural.com



3.2. Ideation & Brainstorming



Ideation refers to the whole creative process of coming up with and communicating new ideas. It can take many different forms, from coming up with a totally new idea to combining multiple existing ideas to create a new process or organizational system. Ideation is similar to a practice known as brainstorming.

We have created Ideation & Brainstorming same using the Mural website

3.3. Proposed Solution

Your proposed solution should **relate the current situation to a desired result and describe the benefits that will accrue when the desired result is achieved**. So, begin your proposed solution by briefly describing this desired result.

We have created proposed solutions for our project which satisfies the needs of consumers

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To Develop affordable app-based solution for Soil health monitoring and suggest which crop to be sown based on it. (Technology Bucket: IoT, AI, ML etc.)
2.	Idea / Solution description	Create app-based solution to detect soil parameters like moisture content, temperature, relative humidity, nutrient, Ph, CEC, NPK etc. and provide crop suggestions to be produced based on soil parameters & environment values.
3.	Novelty / Uniqueness	Automatic crop maintenance and protection using embedded 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 with minimum cost with high performance .
6.	Scalability of the Solution	This can be developed to a scalable product by using sensors and transmitting the data through Wireless Sensor Network and Analysing the data in cloud and operation is performed using robots

3.4. Problem Solution fit

The Problem-Solution Fit simply means that **you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem**

We have created the problem solution fit for existing problems like smart farming , production increase and easy controlling from anywhere we go.

4. REQUIREMENT ANALYSIS

Requirements analysis is **critical to the success or failure of a systems or software project**. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

4.1. Functional requirement

Functional requirements are **product features or functions that developers must implement to enable users to accomplish their tasks**. So, it's important to make them clear both for the development team and the stakeholders. Generally, functional requirements describe system behavior under specific conditions.

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Log in to system	Check Credentials Check Roles of Access.
FR-4	Manage Modules	Manage System Admins Manage Roles of User Manage User permission
FR-5	Check whether details	Temperature details Humidity details
FR-6	Log out	Exit

4.2. Non-Functional requirements

To properly discuss examples of non-functional requirements (NFRs), we first need to understand the scope of this type of requirement.

Unlike their functional counterparts, non-functional requirements cover an incredibly broad scope. While a functional requirement specifies what a system should do exactly in a given scenario, the non-functional requirement instead specifies the overall qualities a project, system, or process should exhibit.

While these non-functional requirements might not “do anything specifically”, they do however outline concretely the attributes a system, process, or project must have on completion. Non-functional requirements will then be used to measure the overall success of a given project, process, or system, and provide measurable insights into how close to completion our project might be.

As time goes on, teams, companies, and stakeholders will notice that their list of non-functional

requirements will continue to grow. This is expected, as NFR's will need to be created, changed and adapted as the needs of an organization change.

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	✓ User friendly guidelines for users to avail the features. ✓ Most simplistic user interface for ease of use.
NFR-2	Security	✓ All the details about the user are protected from unauthorized access. ✓ Detection and identification of any misfunctions of sensors.
NFR-3	Reliability	✓ Implementing Mesh IoT Networks ✓ Building a Multi-layered defence for IoT Networks.
NFR-4	Performance	The use of modern technology solutions helps to achieve the maximum performances thus resulting in better quality and quantity yields.
NFR-5	Availability	This app is available for all platforms
NFR-6	Scalability	Scalability refers to the ability to increase available resources system capability without

5. PROJECT DESIGN

A project design is **the process of outlining all of a project's stages and creating a project plan**. It includes a strategy of ideas, resources and processes to achieve project goals and keep within a budget and deadline.

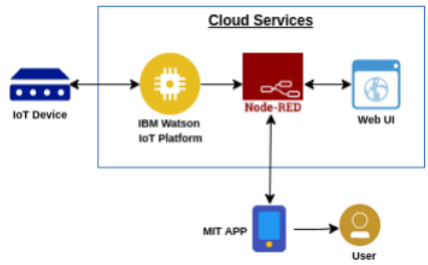
5.1. Data Flow Diagrams

A Data Flow Diagram (DFD) is **a graphical representation of the flow of data through an information system** (ie: shows business processes and the data that flows between them).

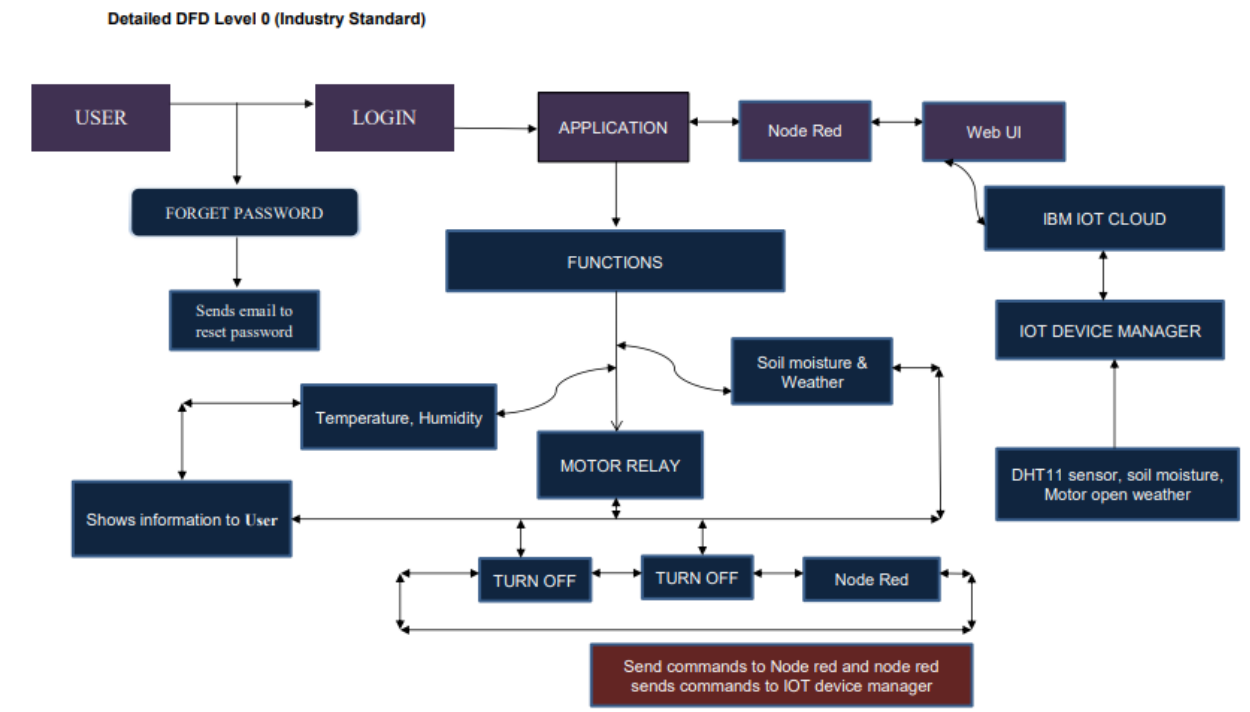
Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Example: (Simplified)



- Different parameters such as temperature, humidity, soil moisture are sensed using the sensors.
- Open weather API is used for collecting the weather information.
- Above data are processed with the help of microcontroller which is connected to internet.
- The processed data is updated to cloud for further process
- The IBM Watson IoT Platform is connected with node red services which is connected to the application.
- In application, user can see the parameters/data that obtained from sensors and APIs.
- With the help of application user can interact with IoT devices to perform some functions such turning ON & OFF motor.
- Web UI is also used for visualization of data.



b. Solution & Technical Architecture

A solution architecture (SA) is **an architectural description of a specific solution**. SAs combine guidance from different enterprise architecture viewpoints (business, information and technical), as well as from the enterprise solution architecture (ESA).

c. User Stories

A user story is **an informal, general explanation of a software feature written from the perspective of the end user or customer**. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer.

User Stories

Use the below template to list all the user stories for the product.

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, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	Medium	Sprint-1
		USN-3	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password		High	Sprint-1
		USN-5	As a user, I can reset the password, if I forget the password		High	Sprint-1
	Dashboard	USN-6	As a user I want to see everything in single widget		Medium	Sprint-2
		USN-7	As a user I want a organised widgets section		High	Sprint-2
		USN-8	As a user I want a graphical/pictorial representation		Low	Sprint-2
Customer (Web User)	Dashboard	USN-9	As a user I want a graphical representation of data for better understanding		High	Sprint-2
		USN-10	As a user I want to see a dashboard where I can customise myself	Dashboard with customization	Low	Sprint-2
Customer (Mobile and Web)	IoT Device Setup	USN-11	Have to use a least sensor and get better output		High	Sprint-2
		USN-12	As a user, I need a low cost IoT devices for farming		High	Sprint-2
		USN-13	As a user, I need multiple sensors for various data		High	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1. Sprint Planning & Estimation

In Scrum Projects, Estimation is done by the entire team during Sprint Planning Meeting. The objective of the Estimation would be **to consider the User Stories for the Sprint by Priority and by the Ability of the team to deliver during the Time Box of the Sprint**.

Sprint	Functional Requirement (Epic)	User Story Number	User Story /Task	Story Points	Priority	Team Member
Sprint-1	Registration (Farmer Mobile User)	UNS - 1	As a user, I can register to the application by simply entering my email, password, and confirming my password.	2	High	Harini U (Team Leader)
Sprint-1	Login	UNS - 2	As a user, I will receive confirmation email once I have done the registration process.	1	HIgh	Abinaya S (Team Member 1)
Sprint-2	User Interface	UNS - 3	As a user, I can register for the application through Facebook, if it is necessary	3	Low	Abirami V (Team member 2)
Sprint-1	Data Visualization	UNS - 4	As a user, I can register for the application through GMAIL	2	Medium	Divya shree R (Team Member 3)
Sprint-3	Registration (Farmer -Web User)	USN - 5	As a user, I can log into the application by entering email and password	3	High	Harini U (Team Leader)
Sprint – 2	Login	USN - 6	As a registered user, I need to easily login into my registered account easily which is an easy process	3	HIgh	Abinaya S (Team Member 1)
Sprint – 4	Web UI	USN - 7	As a user, I need to have a friendly user interface to easily view and access the resources	3	Medium	Abirami V (Team member 2)
Sprint – 2	Dashboard	USN - 8	As a user, I must watch my metrics displayed right in the front page of the application	2	High	Divya shree R (Team Member 3)
Sprint - 2		USN - 9	As a customer care executive, I must provide with the text box to get the user information using unique no. for every individual user	1	Medium	Harini U (Team Leader)
Sprint - 3	HIardware	USN - 10	Configuring Sensors Which will help to find the parameters like temperature, moisture etc.,	2	HIgh	Abinaya S (Team Member 1)
Sprint - 3		USN - 11	Connecting the Hardware to the cloud	2	High	Abirami V (Team member 2)
Sprint - 4	MIT APP INVENTOR & Web UI	USN - 12	Producing app and web UI that will help to analyze agricultural parameters.	2	Medium	Divya shree R (Team Member 3)
Sprint - 4	Testing and bugs fixing	USN - 13	Testing which is an important one through which we can resolve the bugs that are present.	1	Low	Harini U (Team Leader)

6.2. Project Tracker Velocity and Burndown chart

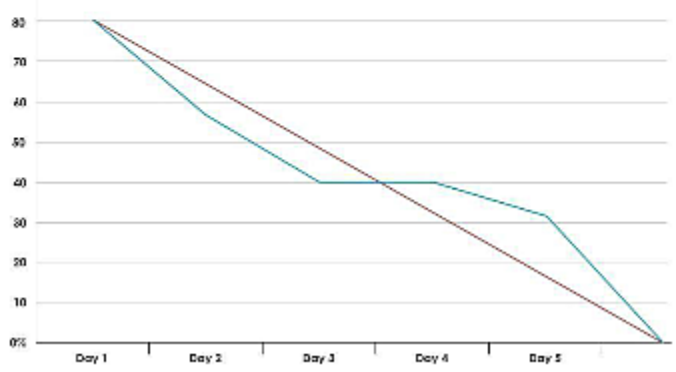
Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	12	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	6	6 Days	31 Oct 2022	05 Nov 2022	20	30 OCT 2022
Sprint-3	6	6 Days	07 Nov 2022	12 Nov 2022	20	6 NOV 2022
Sprint-4	6	6 Days	14 Nov 2022	19 Nov 2022	20	7 NOV 2022

Velocity:

AV for sprint 1= Sprint Duration /velocity =12/6=2
AV for sprint 2= Sprint Duration/Velocity=6/6=1 AV for Sprint 3=Sprint Duration/Velocity=6/6=1 AV for Sprint 4=Sprint Duration/Velocity=6/6=1

Burndown Chart:



6 .3 . Sprint Delivery Schedule

Since sprints take place over a fixed period of time, it's critical to avoid wasting time during planning and development. And this is precisely where sprint scheduling enters the equation.

In case you're unfamiliar, a sprint schedule is a document that outlines sprint planning from end to end. It's one of the first steps in the agile sprint planning process—and something that requires adequate research, planning, and communication.

S.NO	ACTIVITY TITLE	ACTIVITY DESCRIPTION	DURATION
1	**Understanding the project**	Assign the team members after that create repository in the GitHub and then assign task to each member and guide them how to access the GitHub while submitting the assignments	1 week
2	**Staring The Project**	Team Members to Assign All the Tasks Based on Sprints and Work on It Accordingly.	1 week
3	**Completing Every Task**	Team Leader should ensure that whether every team member have completed the assigned task or not	1 week
4	**Stand Up Meetings**	Team Lead Must Have a Stand-Up Meeting with The Team and Work on The Updates and Requirement Session	1 week

5	**Deadline**	Ensure that team members are completing every task within the deadline	1 week
6	**Budget and Scope of project**	Analyze the overall budget which must be within certain limit it should be favorable to every person	1 week

7. CODING & SOLUTIONING

7.1. Python IDLE

For running our source code we have used Python IDLE 3.7.4 latest version

a. Source code 1

```
#include <Servo.h> Servo s;
int Sensor = 0; int data = 0;
int motorPin = 9;
void setup()
{
Serial.begin(9600); pinMode(A0,INPUT);
//Temperature Sensor pinMode(A1,INPUT);
//Soil Moisture Sensor pinMode(10,OUTPUT);
//GREEN light for LED pinMode(11,OUTPUT);
//BLUE light for LED pinMode(12,OUTPUT);
//RED light for LED s.attach(3);
//Servo Motor
pinMode(motorPin, OUTPUT); //DC motor
}
```

```
void loop(){
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
double a = analogRead (A0); //Reads data from Temperature sensor
double t = (((a/1024)*5)-0.5)*100;
Serial.print("Temperature value:");
Serial.println(t);
if (t>40 & t<50)
{
digitalWrite(10,0);
digitalWrite(11,1);
 digitalWrite(12,0);
s.write(90);
digitalWrite(motorPin, HIGH);
Serial.println("Water Partially Flows");
}
else if (t>50)
{
digitalWrite(10,0);
digitalWrite(11,0);
digitalWrite(12,1);
s.write(180);
digitalWrite(motorPin, HIGH);
Serial.println("Water Fully Flows");
}
else if (t>30 & data<30)
{
digitalWrite(10,1);
digitalWrite(11,1);
digitalWrite(12,0);
 s.write(90);
digitalWrite(motorPin, HIGH);
```

```
Serial.println("Water Partially Flows");
}
else if (data<50)
{
digitalWrite(10,0);
digitalWrite(11,1);
digitalWrite(12,1);
 s.write(90);
digitalWrite(motorPin, HIGH);
Serial.println("Water Partially Flows");
}
else
{
digitalWrite(10,1);
digitalWrite(11,0);
digitalWrite(12,0);
s.write(0);
digitalWrite(motorPin, LOW);
 Serial.println(" ");
delay(1000);
}
}
```

Simulation Link :https://www.tinkercad.com/things/9T64ABJ1hL6-brilliant-jofo/editel?tenant=circuits

b.Source Code:

```
#include <Adafruit_LiquidCrystal.h> //Includes the library for LCD Display #include <Wire.h> //Includes the library for connections
#include <Servo.h> //Includes the library for Servo Motor
Servo s; int e = 4; int t = 5; int r = 12; int b = 11; int g = 10; int sec = 0;
int Sensor = 0; int soil = 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 soil = map(Sensor, 0, 1023, 0, 117);
// 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(soil);
//'data = 0' indicates total wetness and 'data = 100' indicates total dryness
// Temperature:
double a = analogRead(A0); // Reads data from Temperature sensor double t = (((a / 1024) * 5) - 0.5) * 100;
Serial.print("Temperature value:"); //Temperature value in Celsius Serial.println(t);
// Ultrasonic sensor:
float distance = readDistanceCM(); //Reads data from Ultrasonic sensor Serial.print("Measured distance: ");
Serial.println(readDistanceCM());
// LCD Display:
lcd.setBacklight(1); //ON the background light in LCD lcd.clear();
// Conditions:
/*If the temperature is Greater than 20 and less than 35 and also the moisture of soil is less than 60 then the
GREEN light will be turned ON indicating the Normal condition */
if (t >= 20 && t < 35 && soil >= 40 && soil < 50)
{
digitalWrite(b, 0);
digitalWrite(g, 1);
digitalWrite(r, 0); s.write(90);
digitalWrite(motorPin, HIGH); lcd.setCursor(3, 0); lcd.print("ON MOTOR"); delay(1000);
lcd.clear();
Serial.println("Water Partially Flows");
}
/*If the temperature is Greater than 35 and less than 45, then the BLUE light will be turned ON indicating the
Intermediate risk condition due to slightly warm weather */
else if (t >= 35 && t < 45)
{
digitalWrite(b, 1);
digitalWrite(g, 0);
digitalWrite(r, 0); s.write(90);
```

```
digitalWrite(motorPin, HIGH); lcd.setCursor(3, 0);
lcd.print("ON MOTOR"); delay(1000);
lcd.clear();
Serial.println("Water Partially Flows");
}
/*If the temperature is Greater than 45 or the moisture of soil is less than 30, then the RED light will be turned ON indicating the Critical condition due to highly warm weather or the low moisture content in soil */
else if (t >= 45 || soil < 30)
{
digitalWrite(b, 0);
digitalWrite(g, 0);
digitalWrite(r, 1); s.write(180); digitalWrite(motorPin, HIGH);
Serial.println("Water Fully Flows"); lcd.setCursor(2, 0);
lcd.print("ON MOTOR!!!"); lcd.setCursor(3, 1); lcd.print("Low Water"); 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 also the moisture of soil is greater than 80, then the
YELLOW light will be turned ON indicating the high water level */ else if (distance<10 && soil> 80)
{
digitalWrite(b, 0);
digitalWrite(g, 1);
digitalWrite(r, 1); s.write(0);
digitalWrite(motorPin, LOW); Serial.println("Water Does Not Flow"); lcd.clear();
lcd.setCursor(3, 0); lcd.print("OFF MOTOR"); delay(1000);
lcd.clear(); lcd.setCursor(1, 0);
lcd.print("DRAIN WATER!!!");
delay(1000); lcd.clear();
}
else
{
digitalWrite(b, 1);
digitalWrite(g, 1);
digitalWrite(r, 0); s.write(0);
digitalWrite(motorPin, LOW); lcd.setCursor(3, 0); lcd.print("OFF MOTOR");
delay(1000); lcd.clear();
Serial.println("Water Does Not Flow");
}
lcd.setCursor(0, 0); lcd.print("Temp:"); lcd.print(t); lcd.print("degree"); lcd.setCursor(0, 1);
lcd.print("SoilWetness:"); lcd.print(soil); lcd.print("%");
Serial.println(" "); delay(1000);
}
```

Simulation link : https://www.tinkercad.com/things/jPr2c9YZK1a-smart-farming/editel

c.Source code :

```
mport time import sys import ibmiotf.application import ibmiotf.device import random
#Provide your IBM Watson Device Credentials organization = "hzu4n4" deviceType = "abi" deviceId =
"2790" authMethod = "token" authToken = "ObNY5tR3)*hIq473Y0"
global y
# Initialize GPIO
def myCommandCallback(cmd):
print("Command received: %s" % cmd.data['command']) status=cmd.data['command']
if status=="motoron":
print ("motor is on") if status=="motoroff" :
print ("motor is off")
if status=="manual" :
print ("Motor Control is in Manual Mode") if status=="automatic" : print ("Motor control is in Automatic Mode")
if soilmoisture > 600:
print ("motor is on")
#print(cmd)
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()
# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10
times deviceCli.connect()
while True:
#Get Sensor Data from DHT11 temp=random.randint(0,100) Humid=random.randint(0,100)
soilmoisture=random.randint(0,1023) Phlevel=random.randint(0,14) y=soilmoisture
data = { 'temp' : temp, 'Humid': Humid,'soilmoisture' : soilmoisture ,'Phlevel' : Phlevel }
#print data
def myOnPublishCallback():
print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid,"Soil Moisture is %s
%%" % soilmoisture,"PH level is %s" %Phlevel ,"to IBM Watson")
success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback) if not
success:
print("Not connected to IoTF") time.sleep(10)
deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud deviceCli.disconnect()
```

8.Testing

a.Tinkercad

- We are sign up the tinkercad website using required credentials
- And then can access the what ever components we need
- We can give connections using essential components
- And using program we run and take the output

b.Python IDLE

- First we want to install the python IDLE 3.7.4 version
- And then we want type the program
- Before we run the program it must saved

c.IBM Watson Services

Goal:

Tocreate an IBM WatsonIOT service and create a device using it.

Steps to create an IBM WatsonIOT service:

- Click on cataloginIBM cloud account.
- Click on services.
- Enter as Internetof thing platform.
- Enter region and pricing plan.
- Enter service name and clickcreate.
- Click on launch.
- Then IBM Watson OT platformopens.
- Click on sign in.
- Enter IBM Id.
- Enter Password.
- Then you can access IBMWatson IOT platform.

Steps to createa device:

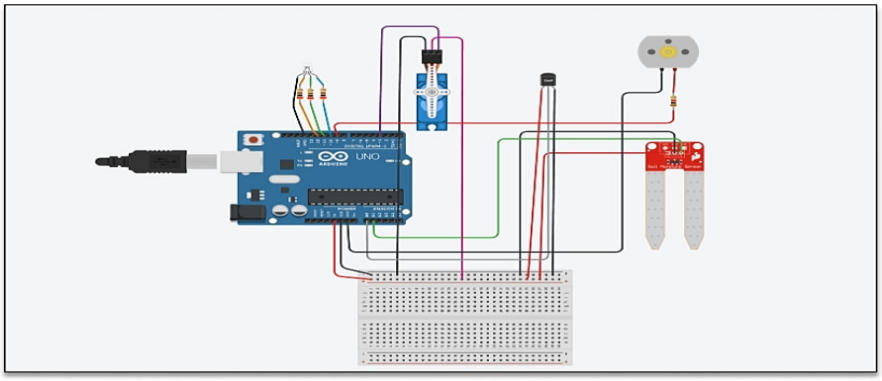
- Click on devicesin IBM Watson IOT platform.
- Choose to create a device.
- Enter the device type as Node MCU.
- Enter the deviceID as 12345.
- Click next.
- Enter device credentials (optional).
- Click next.
- Enter the authentication token (optional).
- Click on continue.
- Click on next.
- Click finish

9. Results

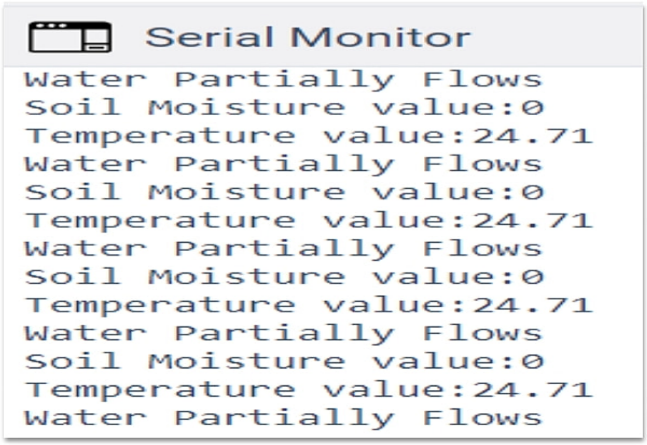
9.1.Performance Metrics

Tinker cad outputs

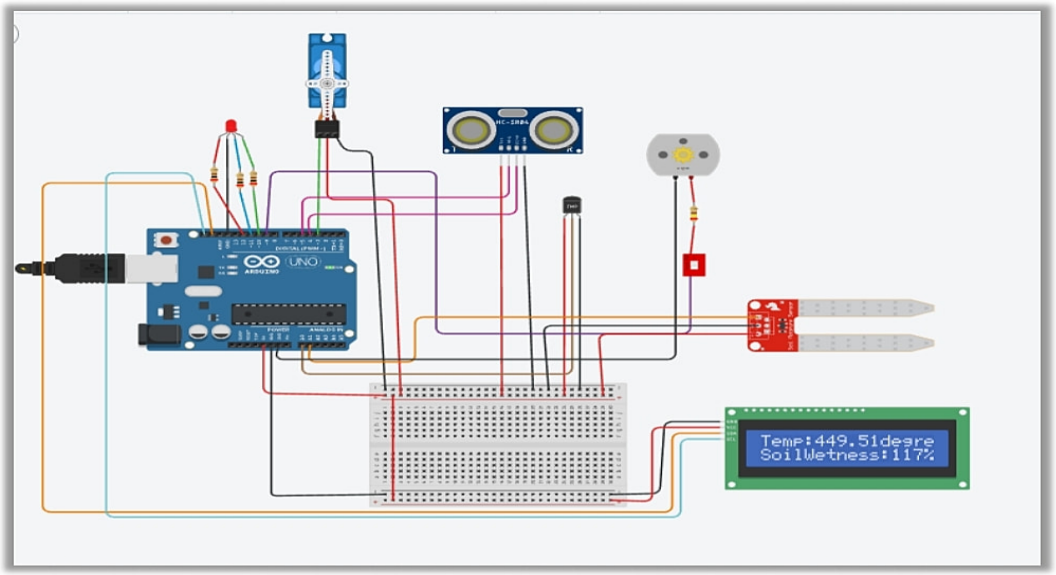
CIRCUIT DIAGRAM



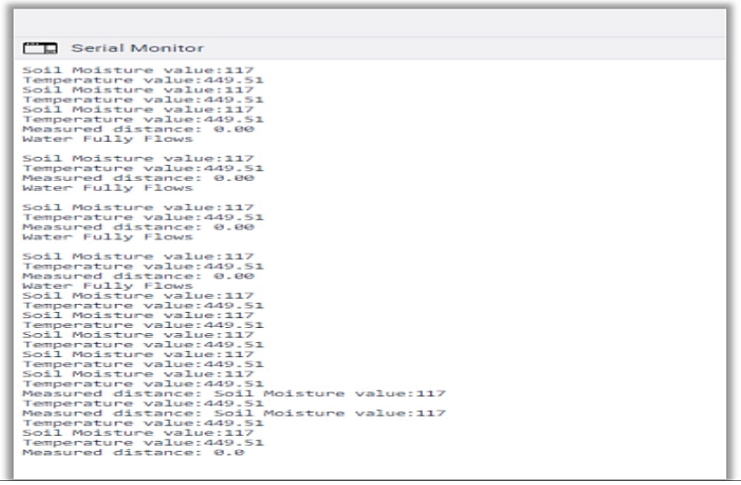
OUTPUT



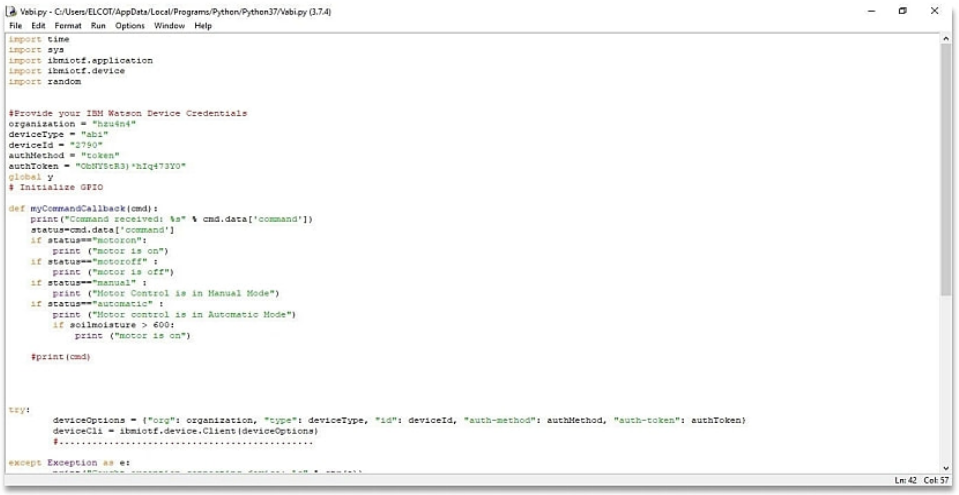
CIRCUIT CONNECTION



OUTPUT

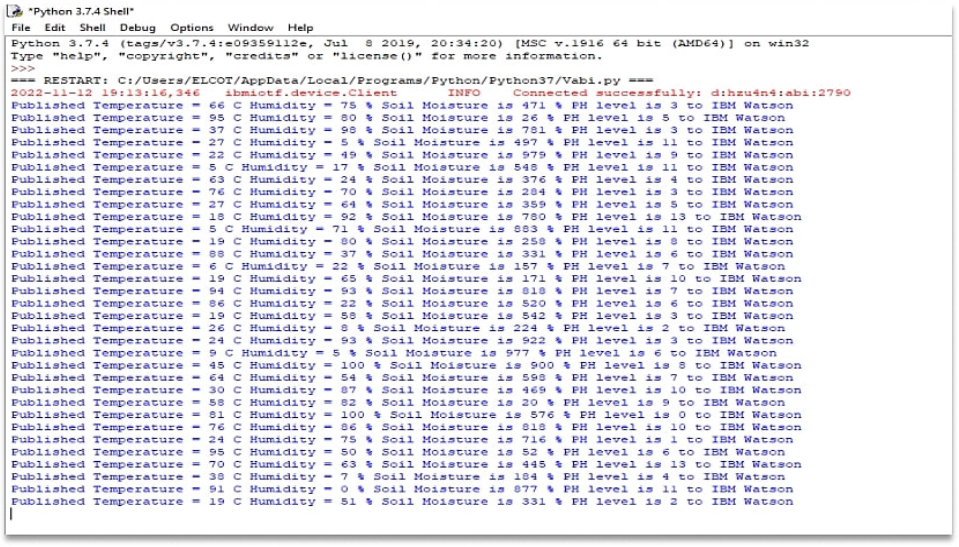


Python idle outputs



```
Vabi.py - C:/Users/ELCOT/AppData/Local/Programs/Python/Python37/Vabi.py (3.7.4)
File  Edit  Format  Run  Options  Window  Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random


#Provide your IBM Watson Device Credentials
organization = "hzu4n4"
deviceType = "abi"
deviceId = "2790"
authMethod = "token"
authToken = "ObNY5tR3)*hIq473YO"
global y
# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("motor is on")
    if status=="motoroff" :
        print ("motor is off")
    if status=="manual" :
        print ("Motor Control is in Manual Mode")
    if status=="automatic" :
        print ("Motor control is in Automatic Mode")
        if soilmoisture > 600:
            print ("motor is on")

    #print(cmd)



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

except Exception as e:
```

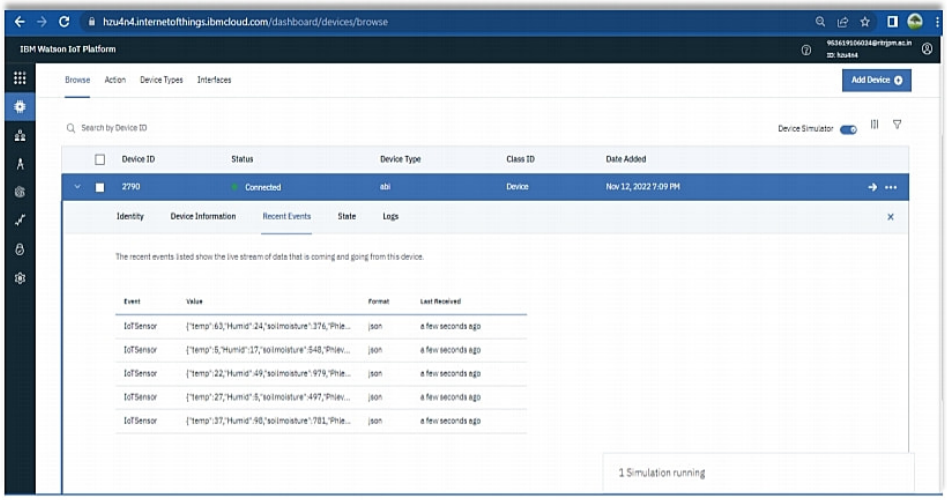
OUTPUT

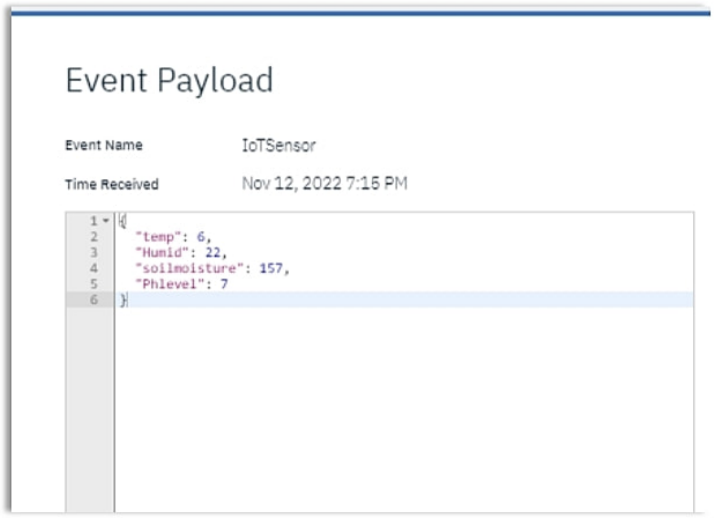


```
*Python 3.7.4 Shell*
File  Edit  Shell  Debug  Options  Window  Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:/Users/ELCOT/AppData/Local/Programs/Python/Python37/Vabi.py ===
2022-11-12 19:13:16,346    ibmiotf.device.Client       INFO    Connected successfully: d:hzu4n4:abi:2790
Published Temperature = 66 C Humidity = 75 % Soil Moisture is 471 % PH level is 3 to IBM Watson
Published Temperature = 95 C Humidity = 80 % Soil Moisture is 26 % PH level is 5 to IBM Watson
Published Temperature = 37 C Humidity = 98 % Soil Moisture is 781 % PH level is 3 to IBM Watson
Published Temperature = 27 C Humidity = 5 % Soil Moisture is 497 % PH level is 11 to IBM Watson
Published Temperature = 22 C Humidity = 49 % Soil Moisture is 979 % PH level is 9 to IBM Watson
Published Temperature = 5 C Humidity = 17 % Soil Moisture is 548 % PH level is 11 to IBM Watson
Published Temperature = 63 C Humidity = 24 % Soil Moisture is 376 % PH level is 4 to IBM Watson
Published Temperature = 76 C Humidity = 70 % Soil Moisture is 284 % PH level is 3 to IBM Watson
Published Temperature = 27 C Humidity = 64 % Soil Moisture is 359 % PH level is 5 to IBM Watson
Published Temperature = 18 C Humidity = 92 % Soil Moisture is 780 % PH level is 13 to IBM Watson
Published Temperature = 5 C Humidity = 71 % Soil Moisture is 883 % PH level is 11 to IBM Watson
Published Temperature = 19 C Humidity = 80 % Soil Moisture is 258 % PH level is 8 to IBM Watson
Published Temperature = 88 C Humidity = 37 % Soil Moisture is 331 % PH level is 6 to IBM Watson
Published Temperature = 6 C Humidity = 22 % Soil Moisture is 157 % PH level is 7 to IBM Watson
Published Temperature = 19 C Humidity = 65 % Soil Moisture is 171 % PH level is 10 to IBM Watson
Published Temperature = 94 C Humidity = 93 % Soil Moisture is 818 % PH level is 7 to IBM Watson
Published Temperature = 86 C Humidity = 22 % Soil Moisture is 520 % PH level is 6 to IBM Watson
Published Temperature = 19 C Humidity = 58 % Soil Moisture is 542 % PH level is 3 to IBM Watson
Published Temperature = 26 C Humidity = 8 % Soil Moisture is 224 % PH level is 2 to IBM Watson
Published Temperature = 24 C Humidity = 93 % Soil Moisture is 922 % PH level is 3 to IBM Watson
Published Temperature = 9 C Humidity = 5 % Soil Moisture is 977 % PH level is 6 to IBM Watson
Published Temperature = 45 C Humidity = 100 % Soil Moisture is 900 % PH level is 8 to IBM Watson
Published Temperature = 64 C Humidity = 54 % Soil Moisture is 598 % PH level is 7 to IBM Watson
Published Temperature = 30 C Humidity = 87 % Soil Moisture is 469 % PH level is 10 to IBM Watson
Published Temperature = 58 C Humidity = 82 % Soil Moisture is 20 % PH level is 9 to IBM Watson
Published Temperature = 81 C Humidity = 100 % Soil Moisture is 576 % PH level is 0 to IBM Watson
Published Temperature = 76 C Humidity = 86 % Soil Moisture is 818 % PH level is 10 to IBM Watson
Published Temperature = 24 C Humidity = 75 % Soil Moisture is 716 % PH level is 1 to IBM Watson
Published Temperature = 95 C Humidity = 50 % Soil Moisture is 52 % PH level is 6 to IBM Watson
Published Temperature = 70 C Humidity = 63 % Soil Moisture is 445 % PH level is 13 to IBM Watson
Published Temperature = 38 C Humidity = 7 % Soil Moisture is 184 % PH level is 4 to IBM Watson
Published Temperature = 91 C Humidity = 0 % Soil Moisture is 877 % PH level is 11 to IBM Watson
Published Temperature = 19 C Humidity = 51 % Soil Moisture is 331 % PH level is 2 to IBM Watson
```

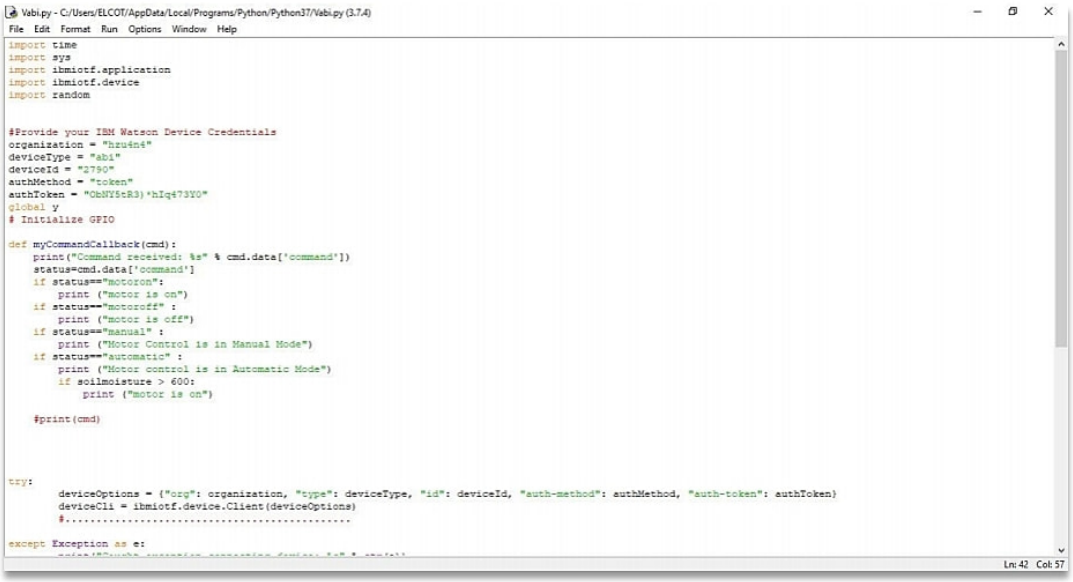
IBM Watson Cloud Output

IBM WATSON CLOUD OUTPUT

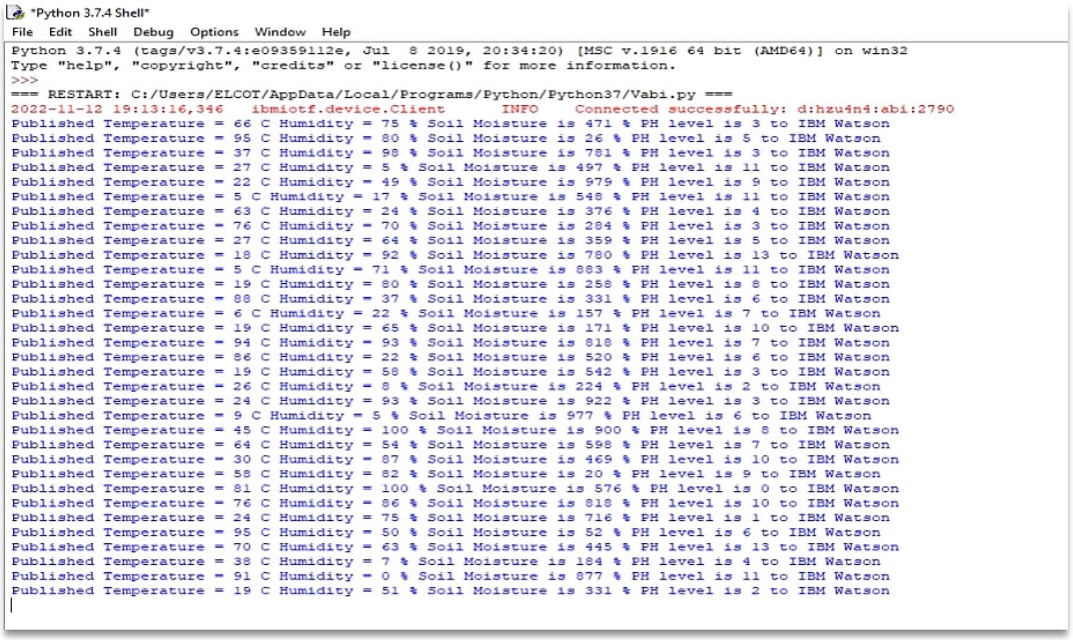




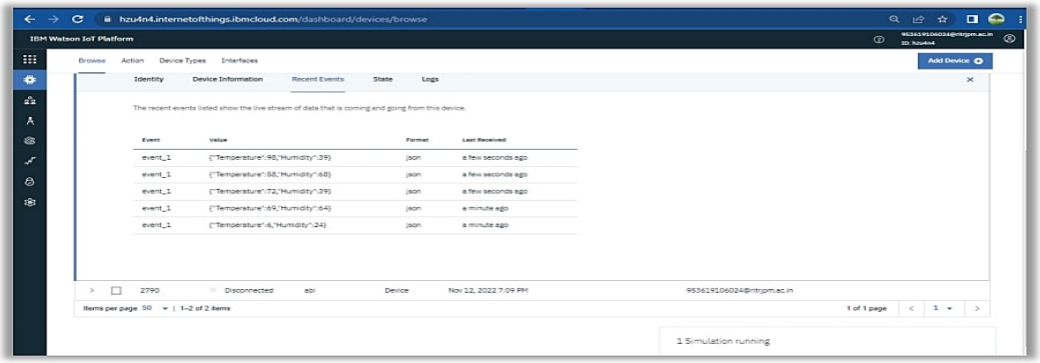
Node Red service output


Vabi.py - C:/Users/ELCOT/AppData/Local/Programs/Python/Python37/Vabi.py (3.7.4)
File Edit Format Run Options Window Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "hzu4n4"
deviceType = "abi"
deviceId = "2790"
authMethod = "token"
authToken = "ObNY5cR3)*hIq473YO"
global y
# Initialize GPIO
def myCommandCallback(cmd):
print("Command received: %s" % cmd.data['command'])
status=cmd.data['command']
if status=="motoron":
print ("motor is on")
if status=="motoroff" :
print ("motor is off")
if status=="manual" :
print ("Motor Control is in Manual Mode")
if status=="automatic" :
print ("Motor control is in Automatic Mode")
if soilmoisture > 600:
print ("motor is on")
#print(cmd)
try:
deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
deviceCli = ibmiotf.device.Client(deviceOptions)
#.....
except Exception as e:
Ln: 42 Col: 57

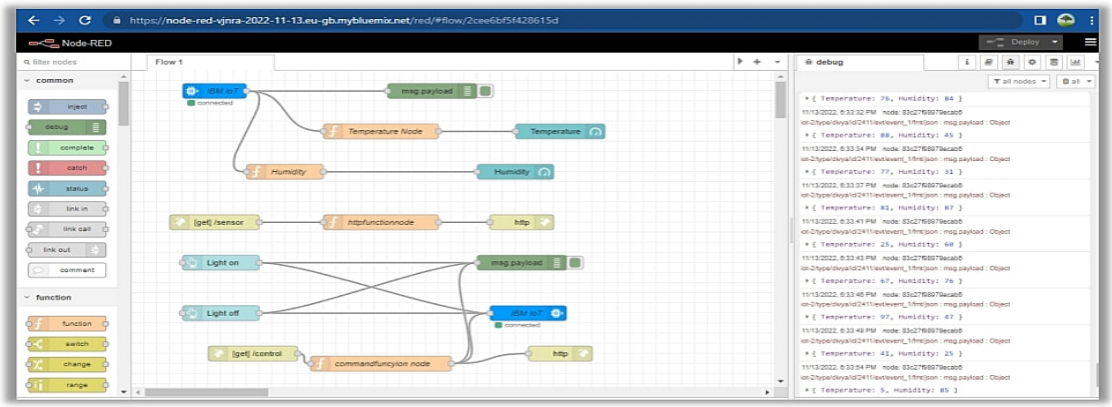
OUTPUT

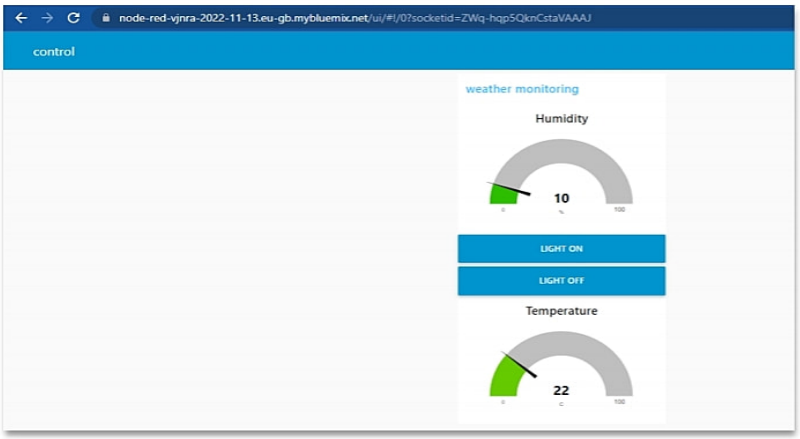

Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:/Users/ELCOT/AppData/Local/Programs/Python/Python37/Vabi.py ===
2022-11-12 19:13:16,346 ibmiotf.device.Client INFO Connected successfully: d:hzu4n4:abi:2790
Published Temperature = 66 C Humidity = 75 % Soil Moisture is 471 % PH level is 3 to IBM Watson
Published Temperature = 95 C Humidity = 80 % Soil Moisture is 26 % PH level is 5 to IBM Watson
Published Temperature = 37 C Humidity = 98 % Soil Moisture is 781 % PH level is 3 to IBM Watson
Published Temperature = 27 C Humidity = 5 % Soil Moisture is 497 % PH level is 11 to IBM Watson
Published Temperature = 22 C Humidity = 49 % Soil Moisture is 979 % PH level is 9 to IBM Watson
Published Temperature = 5 C Humidity = 17 % Soil Moisture is 548 % PH level is 11 to IBM Watson
Published Temperature = 63 C Humidity = 24 % Soil Moisture is 376 % PH level is 4 to IBM Watson
Published Temperature = 76 C Humidity = 70 % Soil Moisture is 284 % PH level is 3 to IBM Watson
Published Temperature = 27 C Humidity = 64 % Soil Moisture is 359 % PH level is 5 to IBM Watson
Published Temperature = 18 C Humidity = 92 % Soil Moisture is 780 % PH level is 13 to IBM Watson
Published Temperature = 5 C Humidity = 71 % Soil Moisture is 883 % PH level is 11 to IBM Watson
Published Temperature = 19 C Humidity = 80 % Soil Moisture is 258 % PH level is 8 to IBM Watson
Published Temperature = 88 C Humidity = 37 % Soil Moisture is 331 % PH level is 6 to IBM Watson
Published Temperature = 6 C Humidity = 22 % Soil Moisture is 157 % PH level is 7 to IBM Watson
Published Temperature = 19 C Humidity = 65 % Soil Moisture is 171 % PH level is 10 to IBM Watson
Published Temperature = 94 C Humidity = 93 % Soil Moisture is 818 % PH level is 7 to IBM Watson
Published Temperature = 86 C Humidity = 22 % Soil Moisture is 520 % PH level is 6 to IBM Watson
Published Temperature = 19 C Humidity = 58 % Soil Moisture is 542 % PH level is 3 to IBM Watson
Published Temperature = 26 C Humidity = 8 % Soil Moisture is 224 % PH level is 2 to IBM Watson
Published Temperature = 24 C Humidity = 93 % Soil Moisture is 922 % PH level is 3 to IBM Watson
Published Temperature = 9 C Humidity = 5 % Soil Moisture is 977 % PH level is 6 to IBM Watson
Published Temperature = 45 C Humidity = 100 % Soil Moisture is 900 % PH level is 8 to IBM Watson
Published Temperature = 64 C Humidity = 54 % Soil Moisture is 598 % PH level is 7 to IBM Watson
Published Temperature = 30 C Humidity = 87 % Soil Moisture is 469 % PH level is 10 to IBM Watson
Published Temperature = 58 C Humidity = 82 % Soil Moisture is 20 % PH level is 9 to IBM Watson
Published Temperature = 81 C Humidity = 100 % Soil Moisture is 576 % PH level is 0 to IBM Watson
Published Temperature = 76 C Humidity = 86 % Soil Moisture is 818 % PH level is 10 to IBM Watson
Published Temperature = 24 C Humidity = 75 % Soil Moisture is 716 % PH level is 1 to IBM Watson
Published Temperature = 95 C Humidity = 50 % Soil Moisture is 52 % PH level is 6 to IBM Watson
Published Temperature = 70 C Humidity = 63 % Soil Moisture is 445 % PH level is 13 to IBM Watson
Published Temperature = 38 C Humidity = 7 % Soil Moisture is 184 % PH level is 4 to IBM Watson
Published Temperature = 91 C Humidity = 0 % Soil Moisture is 877 % PH level is 11 to IBM Watson
Published Temperature = 19 C Humidity = 51 % Soil Moisture is 331 % PH level is 2 to IBM Watson

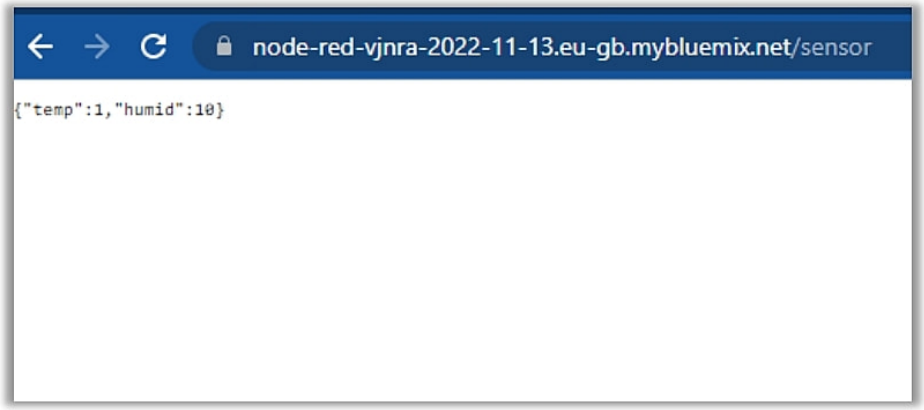
IBM WATSON CLOUD OUTPUT

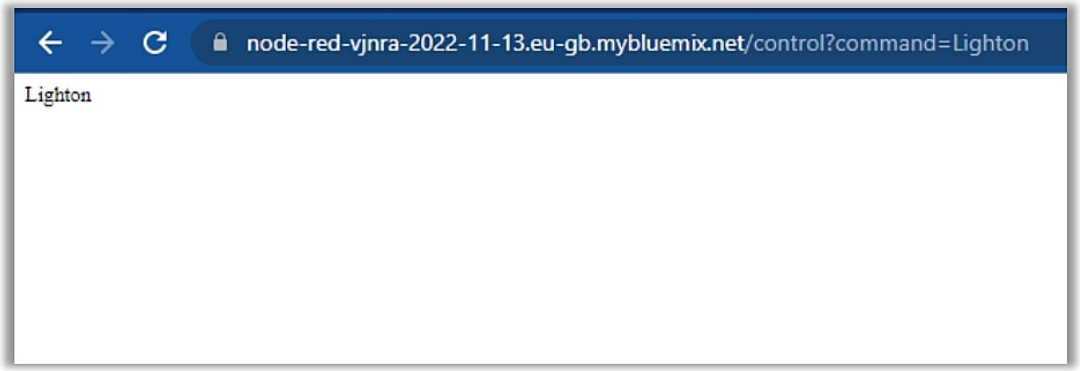


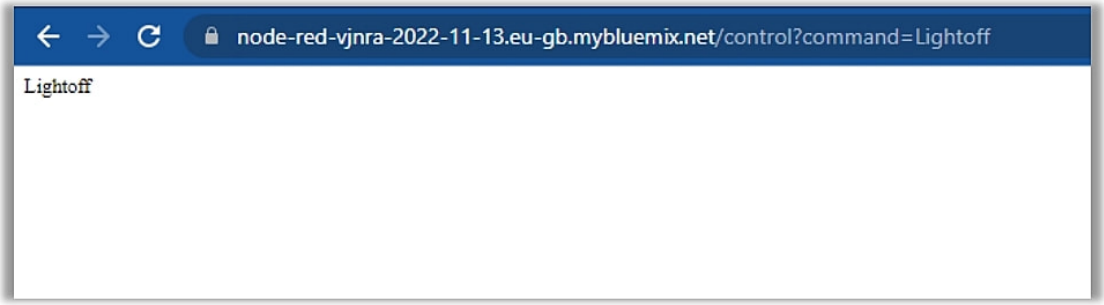
NODE-RED OUTPUT

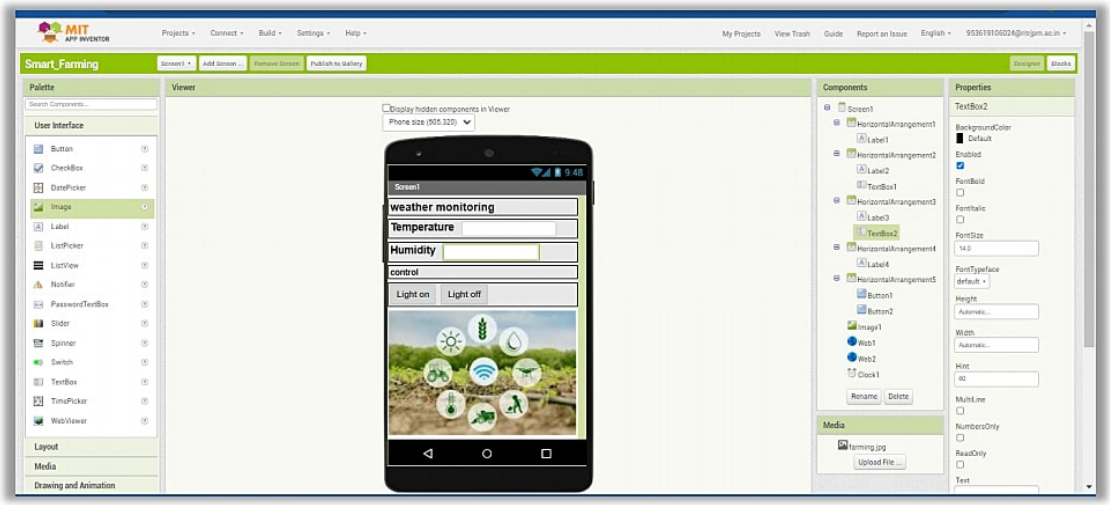




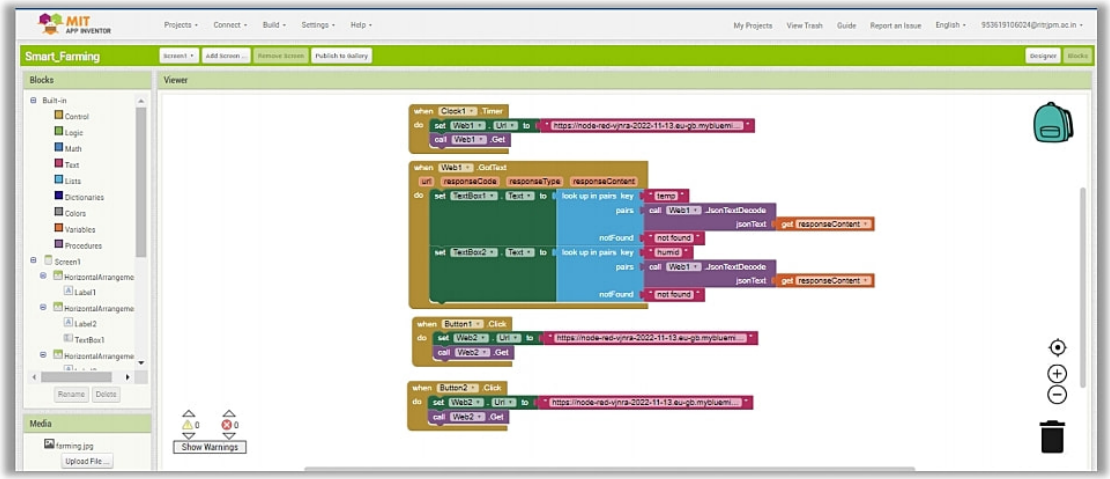

node-red-vjnra-2022-11-13.eu-gb.mybluemix.net/sensor
{"temp":1,"humid":10}

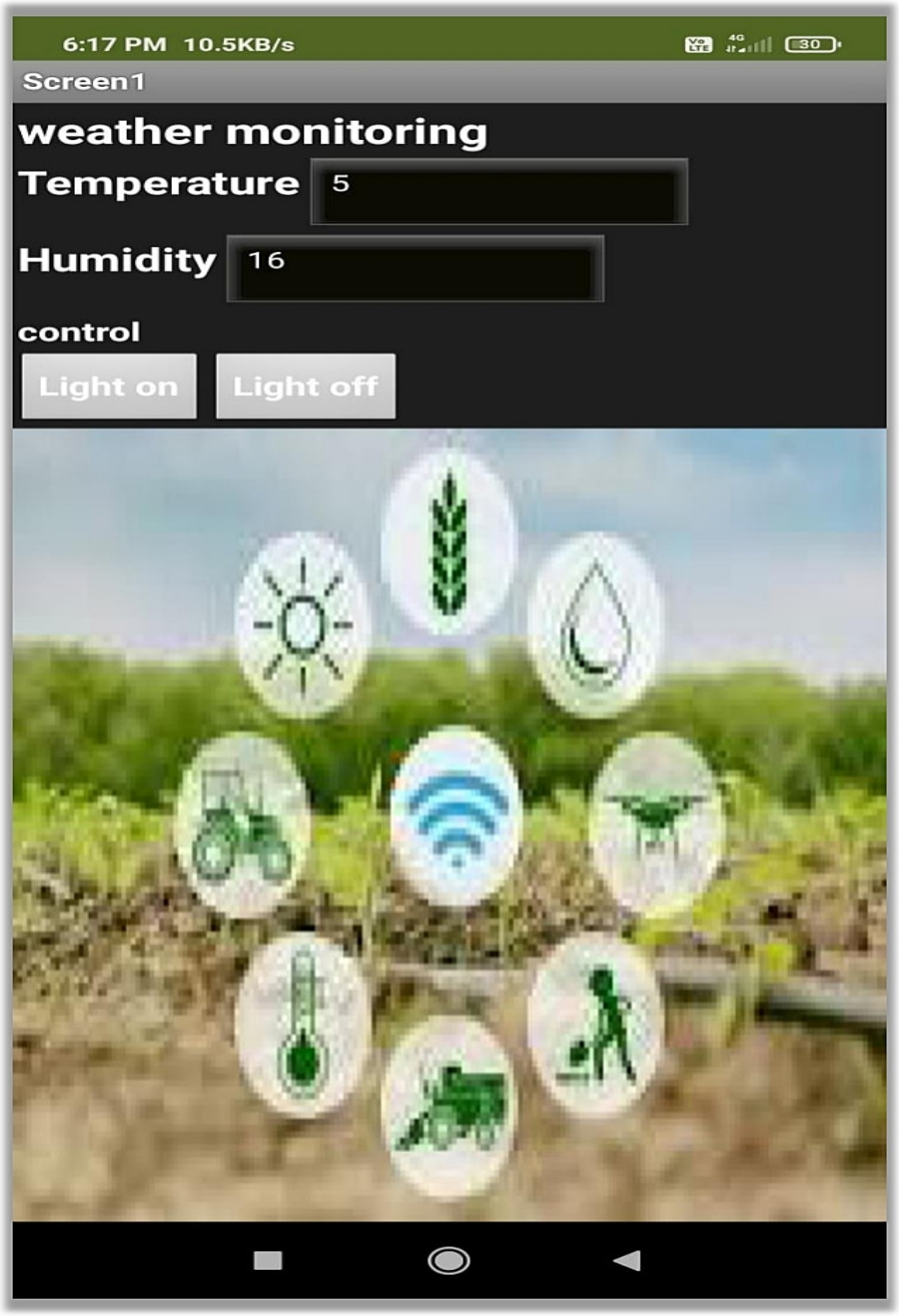

node-red-vjnra-2022-11-13.eu-gb.mybluemix.net/control?command=Lighton
Lighton


node-red-vjnra-2022-11-13.eu-gb.mybluemix.net/control?command=Lightoff
Lightoff



MIT APP INVENTOR – BLOCK





10. ADVANTAGES & DISADVANTAGES

Advantages

- Remote Management. With farms being located in far-off areas and distant lands, farmers are seeking a better solution to their management issues.
- Real-Time Crop Monitoring.
- Crop Protection.
- Soil Testing & its Quality.
- Real-time Analysis of Soil Demand.
- Smart Greenhouses

Disadvantages

- Deforestation. Intensive farming causes soil degradation and leads to the expansion of new lands
- Pest and weed resistance to chemicals
- Soil degradation
- Impact on natural habitats
- Water pollution
- Climate change.

11. Conclusion

Smart farming **reduces the ecological footprint of farming**. Minimized or site-specific application of inputs, such as fertilizers and pesticides, in precision agriculture systems will mitigate leaching problems as well as the emission of greenhouse gases

12.Future works

Through collecting data from sensors using IoT devices, you will learn about the real-time state of your crops. The future of IoT in agriculture allows predictive analytics to help you make better harvesting decisions. The agricultural folks can develop once the techniques are refined to guide poorness mitigation andrising the specification of the individuals.smart farming has a real potential to deliver a more productive and sustainable form of agricultural production, based on a more precise and resource-efficient approach.New farms will finally realize the eternal dream of mankind.It’ll feed our population, which may explodeto 9.6 billionby 2050.

13.Appendix

Source code

```
import time
import sys
import ibmiotf.application
import ibmiotf.device import random
#Provide your IBM Watson Device Credentials
organization = "hzu4n4"
deviceType = "abi"
deviceId = "2790"
 authMethod = "token"
authToken = "ObNY5tR3)*hIq473Y0"
global y
# Initialize GPIO
def myCommandCallback(cmd):
print("Command received: %s" % cmd.data['command']) status=cmd.data['command']
if status=="motoron":
print ("motor is on")
if status=="motoroff" :
print ("motor is off")
if status=="manual" :
print ("Motor Control is in Manual Mode")
if status=="automatic" :
print ("Motor control is in Automatic Mode")
if soilmoisture > 600:
print ("motor is on")
#print(cmd)
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()
# Connect and send a datapoint "hello" with value "world" into the cloud as an event of
type "greeting" 10 times deviceCli.connect()
while True:
```

```
#Get Sensor Data from DHT11 temp=random.randint(0,100)
Humid=random.randint(0,100) soilmoisture=random.randint(0,1023)
Phlevel=random.randint(0,14) y=soilmoisture
data = { 'temp' : temp, 'Humid': Humid,'soilmoisture' : soilmoisture ,'Phlevel' : Phlevel }
#print data
def myOnPublishCallback():
print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid,"Soil
Moisture is %s
%%" % soilmoisture,"PH level is %s" %Phlevel ,"to IBM Watson")
success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
if not success:
print("Not connected to IoTF")
 time.sleep(10)
deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

GITHUB LINK :

https://github.com/IBM-EPBL/IBM-Project-38756-1660385152

PROJECT DEMO LINK :

https://drive.google.com/file/d/1sI3u26AlE69bl5rebNtQb_i7L4V9GVO8/view?usp=drivesdk