Project Report

Chapter No

1. **INTRODUCTION**
   1.1 Project Overview
   1.2 Purpose
2. **LITERATURE SURVEY**
   2.1 Existing problem
   2.2 References
3. **IDEATION & PROPOSED SOLUTION**
   3.1 Empathy Map Canvas
   3.2 Ideation & Brainstorming
   3.3 Proposed Solution
   3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   4.1 Functional requirement
   4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   5.1 Data Flow Diagrams
   5.2 Solution & Technical Architecture
   5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   6.1 Sprint Planning & Estimation
   6.2 Sprint Delivery Schedule
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   7.1 Feature 1
   7.2 Feature 2
   7.3 Database Schema (if Applicable)
8. **TESTING**
   8.1 Test Cases
   8.2 User Acceptance Testing
9. **RESULTS**
   9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

INTERNET OF THINGS (IOT)

- The **Internet of things** (**IoT**) describes physical objects (or groups of such objects) with sensors, processing ability, software and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks. Internet of things has been considered a misnomer because devices do not need to be connected to the public internet, they only need to be connected to a network and be individually addressable.
- The field has evolved due to the convergence of multiple technologies, including ubiquitous computing, commodity sensors, increasingly powerful embedded systems, as well as machine learning. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), independently and collectively enable the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices and appliances (such as lighting fixtures, thermostats, home security systems, cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. IoT is also used in healthcare systems
- There are a number of concerns about the risks in the growth of IoT technologies and products, especially in the areas of privacy and security, and consequently, industry and governmental moves to address these concerns have begun, including the development of international and local standards, guidelines, and regulatory frameworks
- The main concept of a network of smart devices was discussed as early as 1982, with a modified Coca-Cola vending machine at Carnegie Mellon University becoming the first ARPANET-connected appliance, able to report its inventory and whether newly loaded drinks were cold or not. Mark Weiser's 1991 paper on ubiquitous computing, "The Computer of the 21st Century", as well as academic venues such as UbiComp and PerCom produced the contemporary vision of the IOT. In 1994, Reza Raji described the concept in *IEEE Spectrum* as "[moving] small packets of data to a large set of nodes, so as to integrate and automate everything from home appliances to entire factories". Between 1993 and

1997, several companies proposed solutions like Microsoft's at Work or Novell's NEST. The field gained momentum when Bill Joy envisioned device-to-device communication as a part of his "Six Webs" framework, presented at the World Economic Forum at Davos in 1999.

- The concept of the "Internet of things" and the term itself, first appeared in a speech by Peter T. Lewis, to the Congressional Black Caucus Foundation 15th Annual Legislative Weekend in Washington, D.C, published in September 1985. According to Lewis, "The Internet of Things, or IoT, is the integration of people, processes and technology with connectable devices and sensors to enable remote monitoring, status, manipulation and evaluation of trends of such devices."
- The term "Internet of things" was coined independently by Kevin Ashton of Procter & Gamble, later of MIT's Auto-ID Center, in 1999,[19] though he prefers the phrase "Internet *for* things". At that point, he viewed radio-frequency identification (RFID) as essential to the Internet of things, which would allow computers to manage all individual things. The main theme of the Internet of things is to embed short-range mobile transceivers in various gadgets and daily necessities to enable new forms of communication between people and things, and between things themselves.
- In 2004 Cornelius "Pete" Peterson, CEO of NetSilicon, predicted that, "The next era of information technology will be dominated by [IoT] devices, and networked devices will ultimately gain in popularity and significance to the extent that they will far exceed the number of networked computers and workstations." Peterson believed that medical devices and industrial controls would become dominant applications of the technology.
- Defining the Internet of things as "simply the point in time when more 'things or objects' were connected to the Internet than people", Cisco Systems estimated that the IoT was "born" between 2008 and 2009, with the things/people ratio growing from 0.08 in 2003 to 1.84 in 2010.

IoT devices

IoT devices are the nonstandard computing devices that connect wirelessly to a network and have the ability to transmit data, such as the many devices on the internet of things (IoT).

IoT involves extending internet connectivity beyond standard devices, such as desktops, laptops, smartphones and tablets, to any range of traditionally "dumb" or non-internet-enabled physical devices and everyday objects. Embedded with technology, these devices can communicate and interact over the internet. They can also be remotely monitored and controlled.

Example of an IoT device

Connected devices are part of an ecosystem in which every device talks to other related devices in an environment to automate home and industry tasks. They can communicate usable sensor data to users, businesses and other intended parties. The devices can be categorized into three main groups: consumer, enterprise and industrial.

Consumer connected devices include smart TVs, smart speakers, toys, wearables and smart appliances.

In a smart home, for example, devices are designed to sense and respond to a person's presence. When a person arrives home, their car communicates with the garage to open the door. Once inside, the thermostat is already adjusted to their preferred temperature, and the lighting is set to a lower intensity and colour, as their smart watch data indicates it has been a stressful day. Other smart home devices include sprinklers that adjust the amount of water given to the lawn based on the weather forecast and robotic vacuum cleaners that learn which areas of the home must be cleaned most often.

How do IoT devices work?

IoT devices vary in terms of functionality, but IoT devices have some similarities in how they work. First, IoT devices are physical objects designed to interact with the real world in some way. The device might be a sensor on an assembly line or an intelligent security camera. In either case, the device is sensing what's happening in the physical world.

The device itself includes an integrated CPU, network adapter and firmware, which is usually built on an open-source platform. In most cases, IoT devices connect to a Dynamic Host Configuration Protocol server and acquire an IP address that the device can use to function on the network. Some IoT devices are directly accessible over the public internet, but most are designed to operate exclusively on private networks.

Although not an absolute requirement, many IoT devices are configured and managed through a software application. Some devices, however, have integrated web servers, thus eliminating the need for an external application.

IoT device connectivity and networking

The networking, communication and connectivity protocols used with internet-enabled devices largely depend on the specific IoT application deployed. Just as there are many different IoT applications, there are many different connectivity and communication options.

Communication protocols include CoAP, DTLS, MQTT, DDS and AMQP. Wireless protocols include IPv6, LPWAN, Zigbee, Bluetooth Low Energy, Z-Wave, RFID and NFC. Cellular, satellite, Wi-Fi and Ethernet can also be used.

Each option has its trade-offs in terms of power consumption, range and bandwidth, all of which must be considered when choosing connected devices and protocols for a particular IoT application.

In most cases, IoT devices connect to an IoT gateway or another edge device where data can either be analysed locally or sent to the cloud for analysis. Some devices have integrated data processing capabilities that minimized the amount of data that must be sent to the cloud or to the data centre. This type of processing often uses machine learning capabilities that are integrated into the device, and is becoming increasingly popular as IoT devices create more and more data.

IoT device trends and anticipated growth

The estimations for future growth of IoT devices have been fast and furious. At the high end of the scale, Intel projected that internet-enabled device penetration would grow from 2 billion in 2006 to 200 billion by 2020, which equates to nearly 26 smart devices for each human on Earth. A little more conservative, IHS Markit said the number of connected devices will be 75.4 billion in 2025 and 125 billion by 2030.

Other companies have tempered their numbers, taking smartphones, tablets and computers out of the equation. Gartner estimated 20.8 billion connected things would be in use by 2020, with IDC coming in at 28.1 billion and BI Intelligence at 24 billion.

Gartner estimated total spending on IoT devices and services at nearly $2 trillion in 2017, with IDC projecting spending to reach $772.5 billion in 2018, 14.6% more than the $674 billion it estimated to be spent in 2017, with it hitting $1 trillion in 2020 and $1.1 trillion in 2021.

The key to making effective use of IoT devices is to make sure that you start your IoT strategy on the right foot and that you understand how the edge and IoT are intertwined with one another. Regardless of whether you have IoT devices already in use or are considering adopting IoT devices in your organization, make sure you're prepared to handle the unique security challenges presented by IoT devices.

Node red

Node-RED is a flow-based programming tool, originally developed by IBM’s Emerging Technology Services team and now a part of the OpenJS Foundation.

Flow-based Programming

Invented by J. Paul Morrison in the 1970s, flow-based programming is a way of describing an application’s behaviour as a network of black-boxes, or “nodes” as they are called in Node-RED. Each node has a well-defined purpose; it is given some data, it does something with that data and then it passes that data on. The network is responsible for the flow of data between the nodes.

It is a model that lends itself very well to a visual representation and makes it more accessible to a wider range of users. If someone can break down a problem into discrete steps, they can look at a flow and get a sense of what it is doing; without having to understand the individual lines of code within each node.

Runtime/Editor

Node-RED consists of a Node.js based runtime that you point a web browser at to access the flow editor. Within the browser you create your application by dragging nodes from your palette into a workspace and start to wire them together. With a single click, the application is deployed back to the runtime where it is run.

The palette of nodes can be easily extended by installing new nodes created by the community and the flows you create can be easily shared as JSON files.

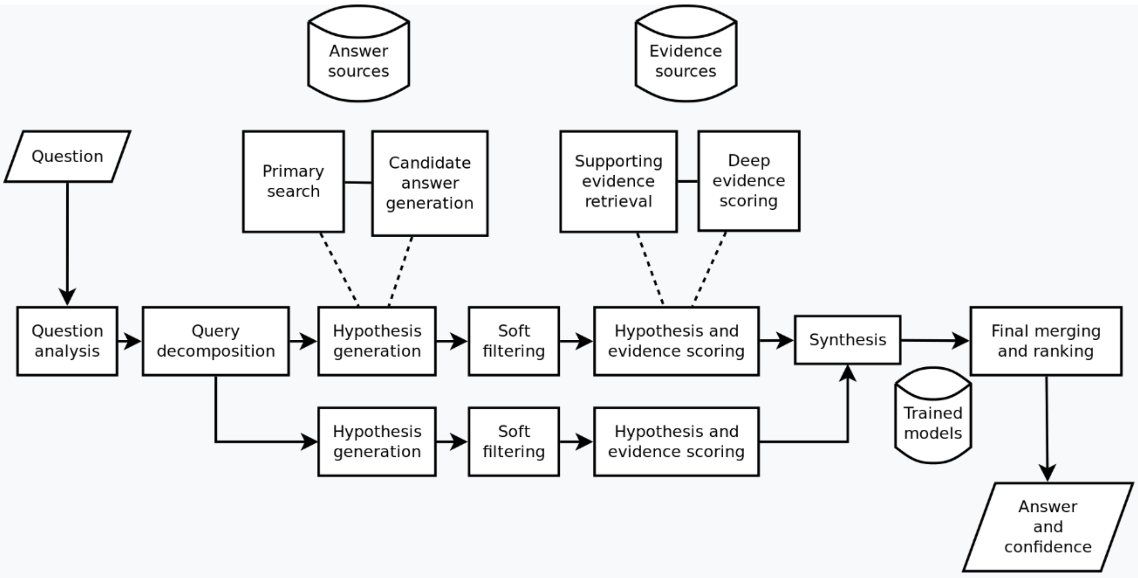
IBM Watson

IBM Watson is a question-answering computer system capable of answering questions posed in natural language, developed in IBM's DeepQA project by a research team led by principal investigator David Ferrucci. Watson was named after IBM's founder and first CEO, industrialist Thomas J. Watson.

The computer system was initially developed to answer questions on the quiz show *Jeopardy!* and, in 2011, the Watson computer system competed on *Jeopardy!* against champions Brad Rutter and Ken Jennings, winning the first place prize of $1 million.

In February 2013, IBM announced that Watson software system's first commercial application would be for utilization management decisions in lung cancer treatment at Memorial Sloan Kettering Cancer Center, New York City, in conjunction with WellPoint (now Anthem).

Description



The high-level architecture of IBM's DeepQA used in Watson

Watson was created as a question answering (QA) computing system that IBM built to apply advanced natural language processing, information retrieval, knowledge representation, automated reasoning, and machine learning technologies to the field of open domain question answering.

Software

Watson uses IBM's DeepQA software and the Apache UIMA (Unstructured Information Management Architecture) framework implementation. The system was written in various languages, including Java, C++, and Prolog, and runs on the SUSE Linux Enterprise Server 11 operating system using the Apache Hadoop framework to provide distributed computing.

Hardware

The system is workload-optimized, integrating massively parallel POWER7 processors and built on IBM's *DeepQA* technology, which it uses to generate hypotheses, gather massive evidence, and analyze data. Watson employs a cluster of ninety IBM Power 750 servers, each of which uses a 3.5 GHz POWER7 eight-core processor, with four threads per core. In total, the system has 2,880 POWER7 processor threads and 16 terabytes of RAM.

According to John Rennie, Watson can process 500 gigabytes (the equivalent of a million books) per second. IBM master inventor and senior consultant Tony Pearson estimated Watson's hardware cost at about three million dollars. Its Linpack performance stands at 80 TeraFLOPs, which is about half as fast as the cut-off line for the Top 500 Supercomputers list. According to Rennie, all content was stored in Watson's RAM for the Jeopardy game because data stored on hard drives would be too slow to compete with human Jeopardy champions.

Data

The sources of information for Watson include encyclopedias, dictionaries, thesauri, newswire articles and literary works. Watson also used databases, taxonomies and ontologies including DBPedia, WordNet and Yago. The IBM team provided Watson with millions of documents, including dictionaries, encyclopedias and other reference material, that it could use to build its knowledge.

Smart farming

There are numerous IoT applications in farming such as collecting data on temperature, rainfall, humidity, wind speed, pest infestation, and soil content. This data can be used to automate farming techniques, take informed decisions to improve quality and quantity, minimize risk and waste, and reduce the effort required to manage crops. For example, farmers can now monitor soil temperature and moisture from afar, and even apply IoT-acquired data to precision fertilization programs. The overall goal is that data from sensors, coupled with the farmer's knowledge and intuition about his or her farm, can help increase farm productivity, and also help reduce costs.

In August 2018, Toyota Tsusho began a partnership with Microsoft to create fish farming tools using the Microsoft Azure application suite for IoT technologies related to water management. Developed in part by researchers from Kindai University, the water pump mechanisms use artificial intelligence to count the number of fish on a conveyor belt, analyse the number of fish, and deduce the effectiveness of water flow from the data the fish provide. The FarmBeats project from Microsoft Research that uses TV white space to connect farms is also a part of the Azure Marketplace now.

- **Sensors**: soil, water, light, humidity, temperature management

OVERVIEW OF SMART FARMING

The advent of modern ICT empowers farmers to gain better productivity, quality, and profit. Hence, smart farming, also known as precision farming, emerges as today's trend. A smart farming system typically involves in sensing systems, communication systems, and data analytics. Figure 1 shows an overview of a smart farming system. The sensing systems including sensors and devices are built at the farm. The sensors are used for monitoring environment variables such as luminosity, temperature, and humidity while the actuators such as water pumps, fans, and lights are used for plant culture. These things communicate by ad-hoc wireless links. Outside the farm is the farmer supporting system. It consists of a web server, database, and knowledge base. Data are stored in the database while knowledge is managed in the knowledge base for further use. Moreover, end-users including farmers are able to access applications and services from the web server through a public network. The smart farming system aims at helping farmers solve daily problems such as properly watering and maintaining consistent plant health. In the next sub-sections, important modules are presented. Furthermore, their relevant issues are addressed and our proposed 310 A Cost-effective Smart Farming System with Knowledge Base SoICT ’17, December 7–8, 2017, Nha Trang City, Viet Nam solutions are given in an effort to design and develop our cost-effective but high level of knowledge smart farming system.

1. INTRODUCTION

1.1 PROJECT OVERVIEW

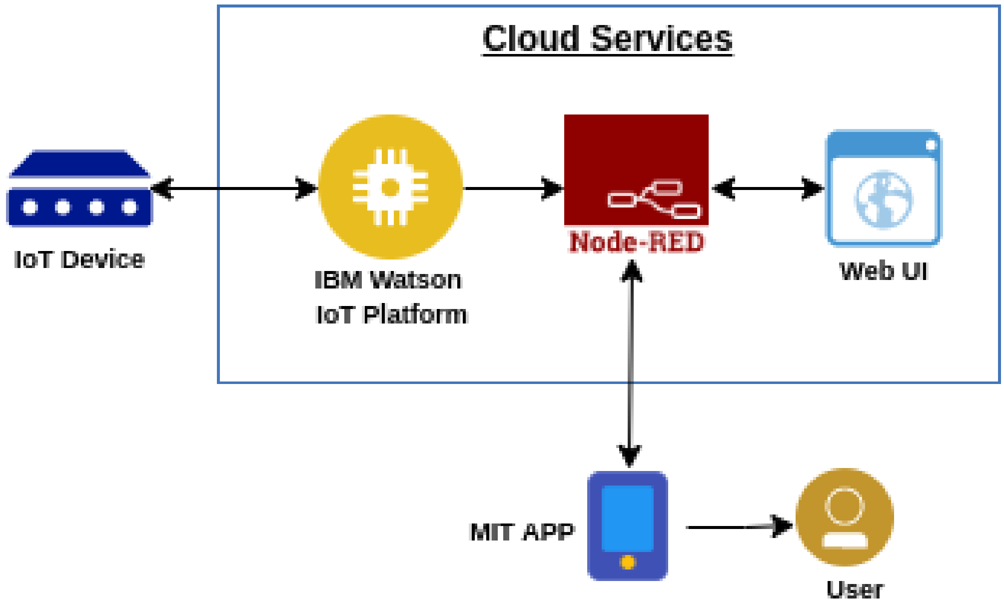
Agriculture is the basic source of livelihood of people in India. In the past decade, it is observed that there is not much crop development in agriculture sector. Food prices are continuously increasing because crop rate is declined. Some of the factors which are responsible for this may be wastage of water, low soil fertility, fertilizer abuse, climate change, diseases, etc.

Monitoring systems are used in the field to collect information on farming conditions (e.g., light intensity, humidity, and temperature) with the aim of enhancing crop productivity. Internet of things (IoT) technology is a recent trend in numerous fields, including monitoring systems for agriculture. In conventional farming, farmers need manual labour to handle crops and livestock, often leading to inefficient resource use. This downside can be addressed through the concept of smart farming, whereby farmers receive training in the use of IoT, access to the global positioning system (GPS), and data management capabilities to increase the quantity and quality of their products.

Latest technologies such as Internet of Things and Cloud in combination with Wireless Sensor Networks can lead to agricultural modernization. IoT is an ecosystem of connected physical devices that is accessible through the Internet. It consists of objects, sensor devices, communication infrastructure, computational and processing units. The sensors communicate the information over the Internet to the cloud server which is a computational and processing unit

In this project, we developed a new farming monitoring system that has a robust design, high accessibility, and wireless communication. The system was integrated by using the input from sensors, interfaced with Arduino Uno, and using GSM as the interface with the end-user (Farmer mobile). Since our aim is to help the farmers, we tried to design the system to be more.

Technical overview of IoT enabled smart farming



1.2 Purpose

To do smart farming with the help of IOT.

The IoT technology has realized the smart wearable's, connected devices, automated machines, and driverless cars. However, in agriculture, the IoT has brought the greatest impact.

Recent statistics reveal that the global population is about to reach 9.6 billion by 2050. And to feed this massive population, the agriculture industry is bounded to adopt the Internet of Things. Amongst the challenges like extreme weather conditions, climatic changes, environmental impact, IoT is eradicating these challenges and helping us to meet the demand for more food.

To modernize the farming by IOT

Internet of Things Smart technology enables new digital agriculture. Today technology has become a necessity to meet current challenges and several sectors are using the latest technologies to automate their tasks. Advanced agriculture, based on Internet of Things technologies, is envisioned to enable producers and farmers to reduce waste and improve productivity by optimizing the usage of fertilizers to boost the efficiency of plants. It gives better control to the farmers for their livestock, growing crops, cutting costs, and resources.

2. LITERATURE SURVEY

2.1 Existing problems

1. Lack of Infrastructure: 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.

2. High Cost: 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.

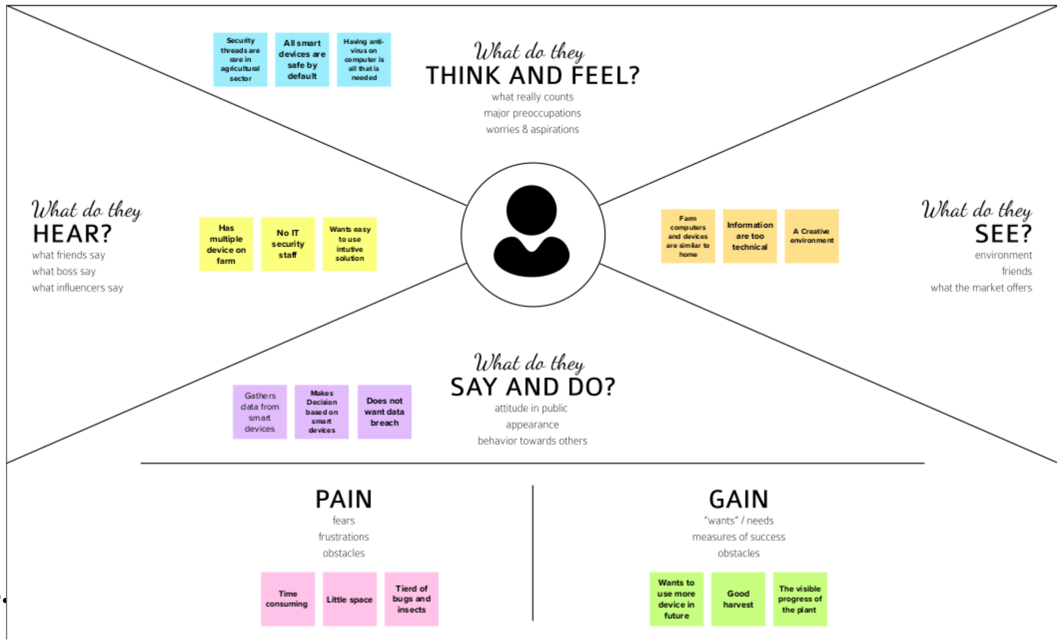
3. Lack of Security: Since IoT devices interact with older equipment they have access to the internet connection, there is no guarantee that they would be able to access drone mapping data or sensor readouts by taking benefit of public connection. An enormous amount of data is collected by IoT agricultural systems which is difficult to protect. Someone can have unauthorized access IoT providers database and could steal and manipulate the data.

REFERENCE

- ✓ **Divya J., Divya M., Janani V. "IoT based Smart Soil Monitoring System for Agricultural Production" 2017.**
- ✓ **R. Nageswara Rao, B. Sridhar. "IoT Based Smart Crop-Field Monitoring And Automation Irrigation System". 2018**
- ✓ **Anushree Math, Layak Ali, Pruthviraj U. "Development of Smart Drip Irrigation System Using IoT" 2018.**

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.

DEFINE YOUR PROBLEM STATMENT

To help medium scale and small-scale farmers to reduce the work load with help of smart farming application by monitoring the fields remotely

my team members have separated the plans into different sections and the allocations are described below

ABIRAMI M

*To monitor the plant growth.

*Testing the soil moisture.

*Based on the soil type determine the seed.

*To measure the humidity.

*Check for the water level that is to be in required quantity.

*Determine what crop should be yield

KARPAGAM N

*To measure the humidity using the humidity sensor.

* To measure the moisture in the field using moisture sensor.

*To develop an android application for the smart farming.

*To make disease identification.

*Check for water level using water level controller.

*Send the notification

SANDHIYA S

* Identify the seeds to be yield.

*To check wind velocity using the sensor.

* Determine the crop period.

*Give the plant nutrition for the better growth.

*Determine the farming land that the crop yield.

VISALATCHI T

*Determine the usage of electricity.

*Link the IOT device to the sensor that is connect in the land.

*Determine the usage of fertilizer.

*To make the user interface of IOT application.

*To collect and process the data given by sensor (sensor database).

* Listening the application features.

3.3 Proposed solution

1.Problem Statement (Problem to be solved)

Farmers are under pressure to produce more food AND use less energy and water in the process.

Ideally, each field should get just the right amount of water at just the right time. Under-watering causes crop stress and yield reduction. Overwatering can also cause yield reduction and consumes more water and fuel than necessary and leads to soil erosion and fertilizer, herbicide, and pesticide runoff.

Agricultural operations waste 60% of water consumed each year. Now more than ever, new technologies for water conservation must be adopted.

2. Idea / Solution description

Case studies have shown precision irrigation has a 5%–8% impact on yield and a similar impact on operating costs. Smart Farm's systems can be retrofitted on existing sites and provide immediate impact with a very short return on investment time period.

3. Novelty / Uniqueness

Smart farming application has been very useful in only large-scale farming, this helps us to implement in medium and small-scale farming.

4. Social Impact / Customer Satisfaction

Make use of the technology in both medium and small-scale farming.

5.Business Model (Revenue Model).

The revenue model can be described by income from purchase of the product, income from cloud service and income from maintenance and service.

6. Scalability of the Solution

Usage of smart farming method will increase among medium and small-scale farmers.

Instead of going to the field for each and every time, using IoT device connected with various sensors, farmer can know their field’s condition remotely.

3.4 Problem solution fit

1. **CUSTOMER SEGMENT(S)**

   Farmers can monitor their land like soil moisture, humidity, water level through application

2. **JOBS-TO-BE-DONE / PROBLEMS**

   Monitoring data fetch by sensors in the field to know about the current situation in the field

3. **TRIGGERS**

   Managing sensor, irrigation and IOT

4. **EMOTIONS: BEFORE / AFTER**

   Farmers didn't know what happened in their land but by using technology they can get knowledge about their field

5. **AVAILABLE SOLUTIONS**

   The available solution is to remotely monitor the field and the condition of the crops.

6. **CUSTOMER CONSTRAINTS**

   The major constraint is farmer cannot predict the crop yield through this given application

7. **BEHAVIOUR**

   They can make the decision whether to water the crop or postponed.

8. **CHANNELS of BEHAVIOUR**

   ONLINE

   Through online farmer can analyze the field using apt sensors

   OFFLINE

   In offline, each and every time farmer need to went to their field to analyze the field.

9. **PROBLEM ROOT CAUSE**

Lack of manpower to manage and monitor the field.

10. YOUR SOLUTION

Instead of going to the field for each and every time, using IoT device connected with various sensors, farmer can know their field’s condition remotely.

4. REQUIREMENT ANALYSIS

4.1 **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 Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Login	Login via Username and PasswordLogin via Google
FR-4	Password reset	Reset password via email Reset password via Phone Number
FR-5	Password Change	Change password via email Change password via Phone Number
FR-6	Settings	Change settings for the convenience

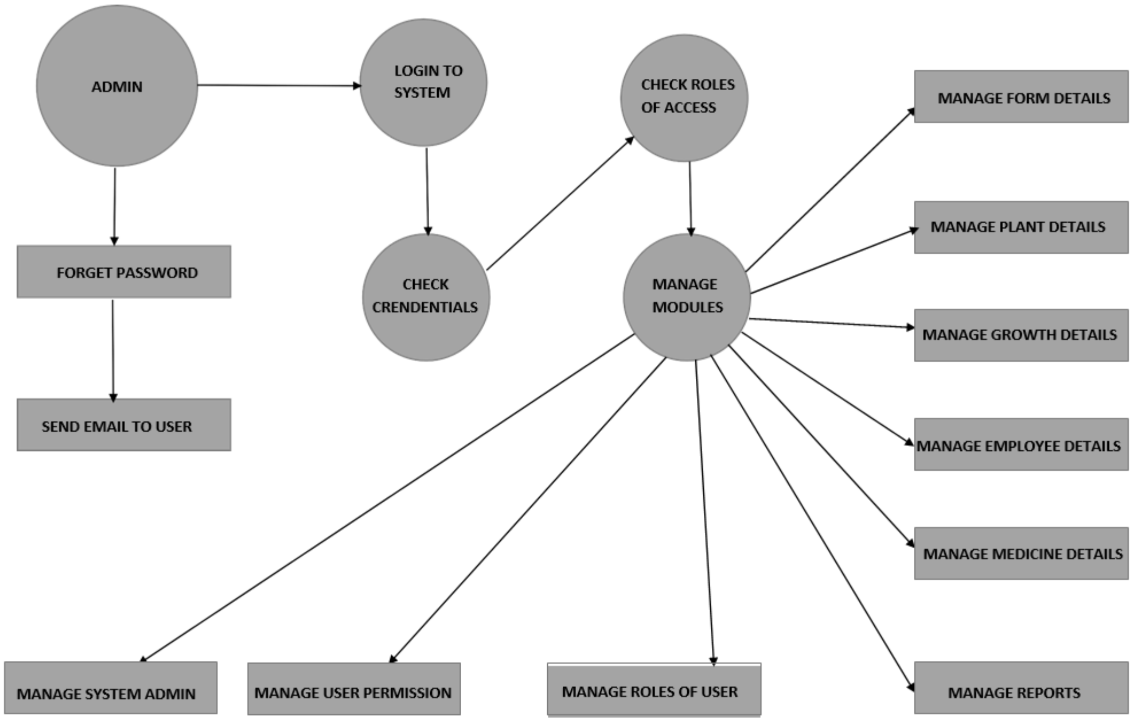
4.2 Non-functional Requirements:

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

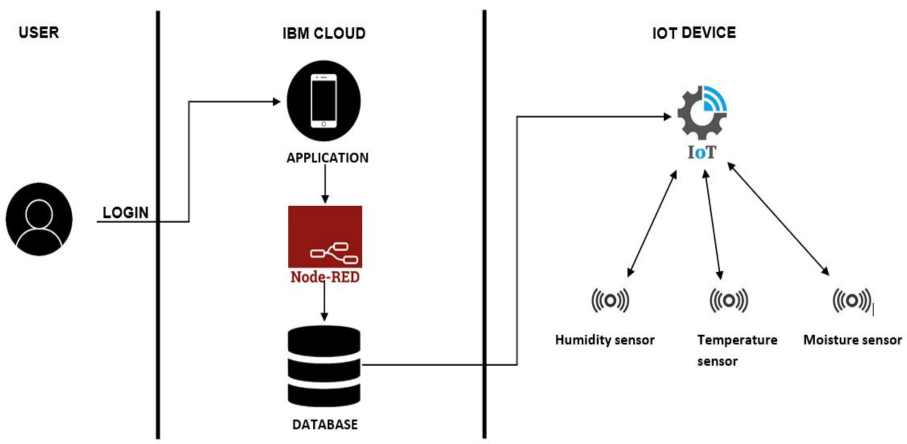
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Application is easy to use with better user experience and the controls given with that application.
NFR-2	**Security**	The user can register or login through their mail id and password. The security attacks could not be done until the user share his/her login credentials to someone.
NFR-3	**Reliability**	The data are stored in the trusted cloud storageand it can be kept confidential. The user and the developer are able to access the data stored in cloud storage.
NFR-4	**Performance**	The user can control and analyze the data about their field or farm through application given with many features.
NFR-5	**Availability**	The user can easily access the analyzed data from the sensors connected with IoT devices which placed in the farming land and the sensor analyzed data are stored in a cloud storage for future references.
NFR-6	**Scalability**	The application features are upgraded randomly for easy access and better user experience.

5.PROJECT DESIGN

5.1Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, 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	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I		High	Sprint-1

			can log into the application by entering email & password			
	Dashboard	USN-6	As a user, I can access the features of the application through dashboard		Medium	Sprint-2
Customer (Web user)						
Customer Care Executive						
Administrator						

Sensor used in this process

Humidity Sensor (DHT22)

The DHT-22 (also named as AM2302) is a digital-output, relative humidity, and temperature sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and sends a digital signal on the data pin.

In this example, you will learn how to use this sensor with Arduino UNO. The room temperature and humidity will be printed to the serial monitor.

The connections are simple. The first pin on the left to 3-5V power, the second pin to the data input pin and the right-most pin to the ground.

Technical Details

- Power − 3-5V
- Max Current − 2.5mA
- Humidity − 0-100%, 2-5% accuracy
- Temperature − 40 to 80°C, ±0.5°C accuracy

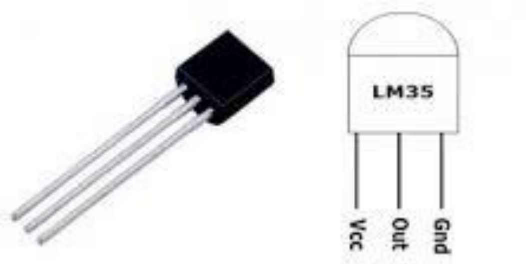
Code for humidity sensor

```
// Example testing sketch for various DHT humidity/temperature sensors
#include "DHT.h"
#define DHTPIN 2 // what digital pin we're connected to
// Uncomment whatever type you're using!
//#define DHTTYPE DHT11 // DHT 11
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
//#define DHTTYPE DHT21 // DHT 21 (AM2301)
// Connect pin 1 (on the left) of the sensor to +5V
// NOTE: If using a board with 3.3V logic like an Arduino Due connect pin 1
// to 3.3V instead of 5V!
// Connect pin 2 of the sensor to whatever your DHTPIN is
// Connect pin 4 (on the right) of the sensor to GROUND
// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor
// Initialize DHT sensor.
// Note that older versions of this library took an optional third parameter to
// tweak the timings for faster processors. This parameter is no longer needed
// as the current DHT reading algorithm adjusts itself to work on faster procs.
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
  Serial.println("DHTxx test!");
  dht.begin();
```

```
}
void loop() {
  delay(2000); // Wait a few seconds between measurements
  float h = dht.readHumidity();
  // Reading temperature or humidity takes about 250 milliseconds!
  float t = dht.readTemperature();
  // Read temperature as Celsius (the default)
  float f = dht.readTemperature(true);
  // Read temperature as Fahrenheit (isFahrenheit = true)
  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
   // Compute heat index in Fahrenheit (the default)
  float hif = dht.computeHeatIndex(f, h);
  // Compute heat index in Celsius (isFahreheit = false)
  float hic = dht.computeHeatIndex(t, h, false);
  Serial.print ("Humidity: ");
  Serial.print (h);
  Serial.print (" %\t");
  Serial.print ("Temperature: ");
  Serial.print (t);
  Serial.print (" *C ");
  Serial.print (f);
  Serial.print (" *F\t");
  Serial.print ("Heat index: ");
  Serial.print (hic);
  Serial.print (" *C ");
```

```
  Serial.print (hif);
  Serial.println (" *F");
}
```

Temperature sensor



The temperature sensor in Arduino converts the surrounding temperature to voltage. It further converts the voltage to Celcius, Celcius to Fahrenheit, and prints the Fahrenheit temperature on the LCD screen.

We will use a temperature sensor (TMP 36) of low voltage. Such sensors are also stable while dealing with large capacitive loads. It is also suitable for automotive applications.

The temperature sensors TMP 35, TMP 36, and TMP 37 are the sensors with the same features.

The operating voltage of the TMP 36 sensor ranges from 2.7V to 5.5V.

Principle

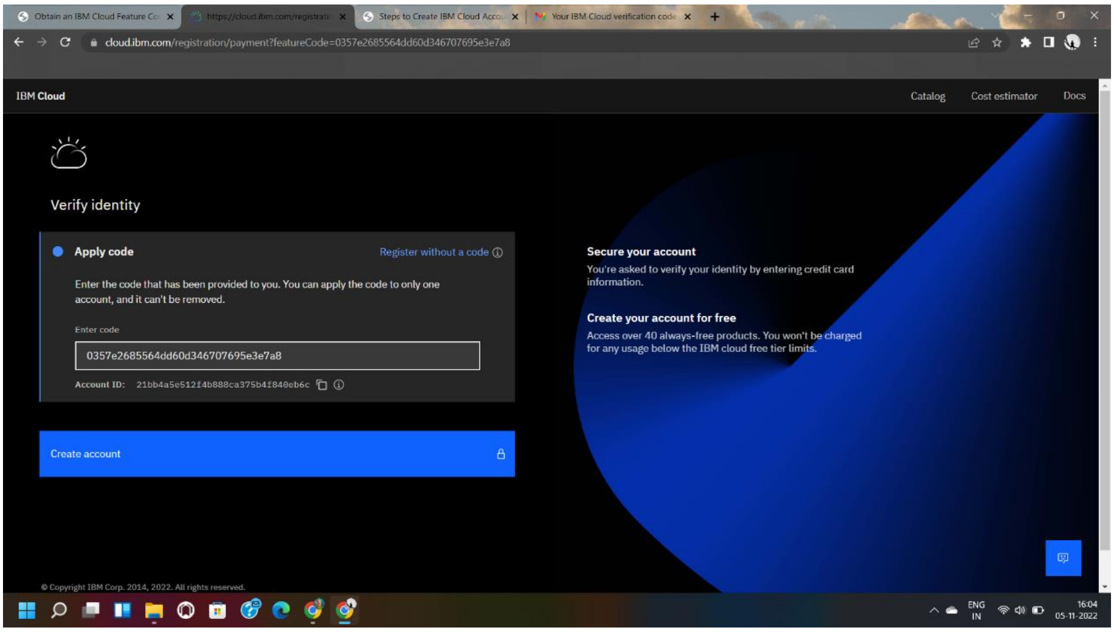
We will connect the LCD Display and TMP 36 temperature sensor with the Arduino UNO R3 board. The sensor detects the surrounding temperature and converts it into volts, to Celsius to Fahrenheit, and displays Fahrenheit temperature on the LCD screen.

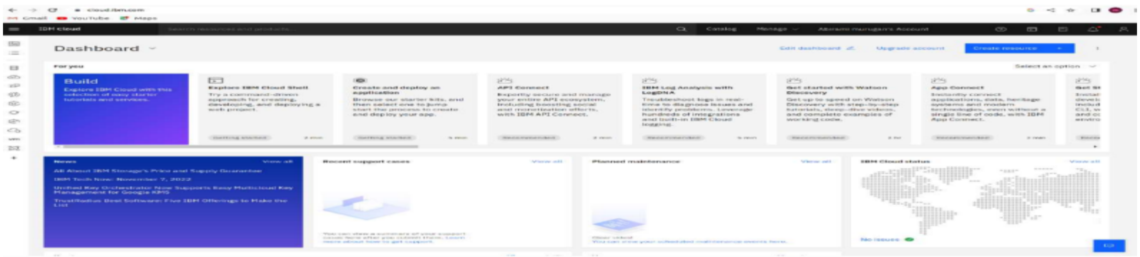
Code for temperature sensor

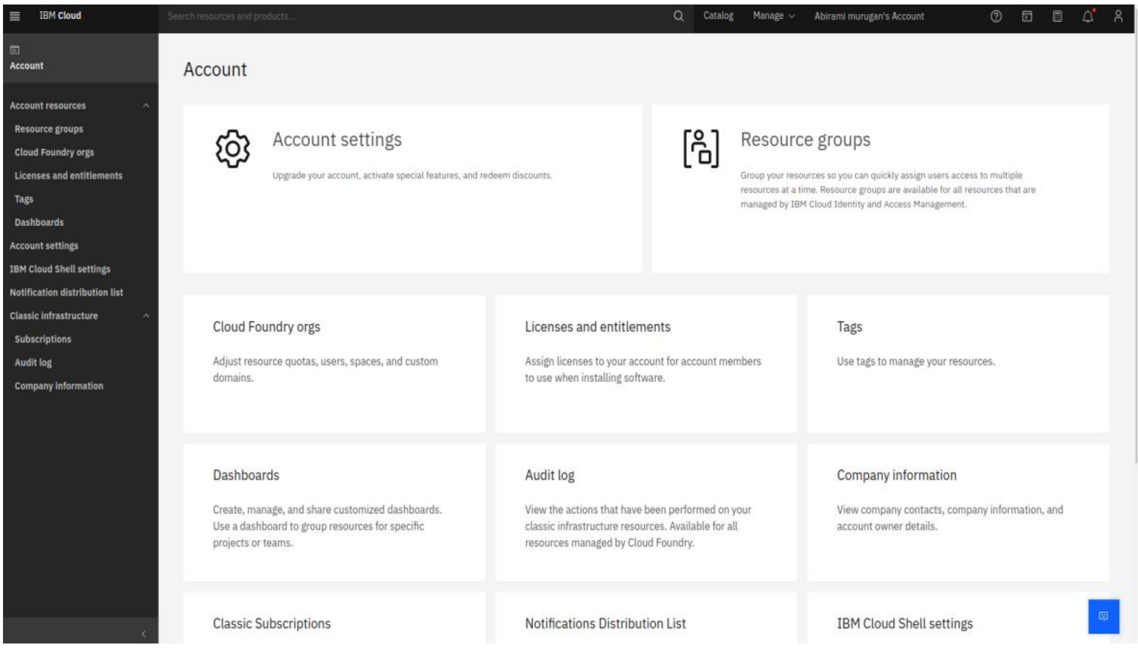
```
#include <LiquidCrystal.h>

// initialize the library with the pins on the Arduino board

LiquidCrystal lcd(13, 12, 6, 4, 3, 2);

const int temperature = A0; //A0 is the analog pin

const int D = 8; // Vo of LCD is connected to pin 8 of the Arduino

void setup()

{

  lcd.begin(16, 2);

  Serial.begin(9600);

  pinMode(D, OUTPUT);

}

void loop()

{

  digitalWrite(D,LOW);

  int Temp = analogRead(temperature);

  float volts = (Temp / 965.0) * 5;

  float celcius = (volts - 0.5) * 100;

  float fahrenheit = (celcius * 9 / 5 + 32);

  Serial.println(fahrenheit);

  lcd.setCursor(5, 0);
```

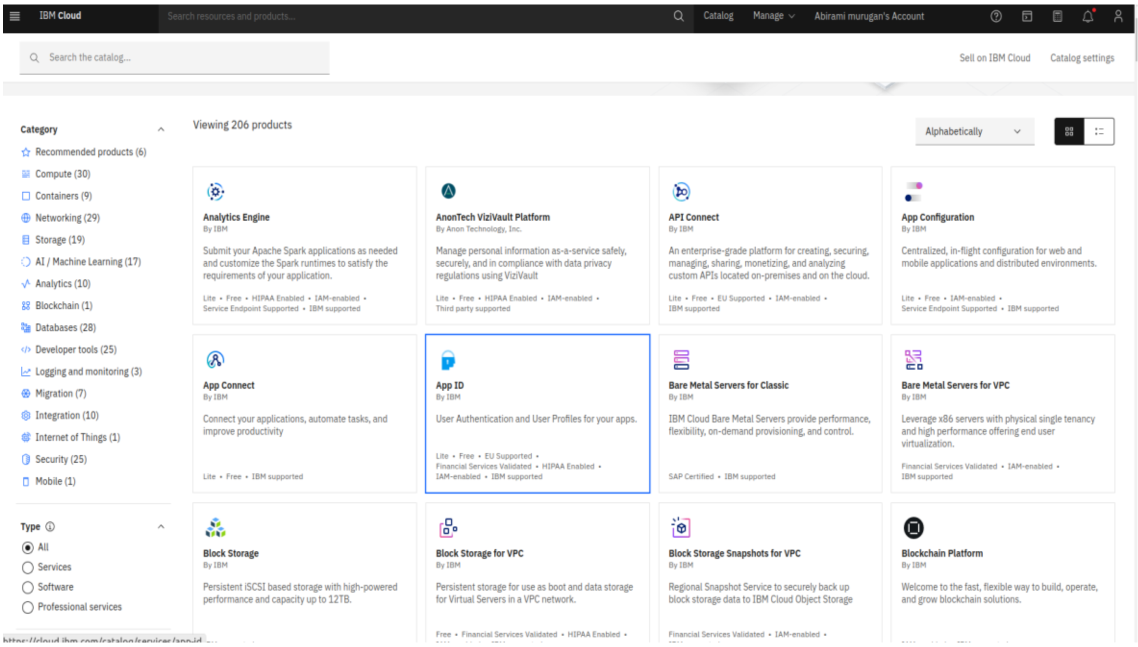
```
  lcd.print(fahrenheit);

  delay(2000);

  // time delay of 2000 microseconds or 2 seconds

}
```

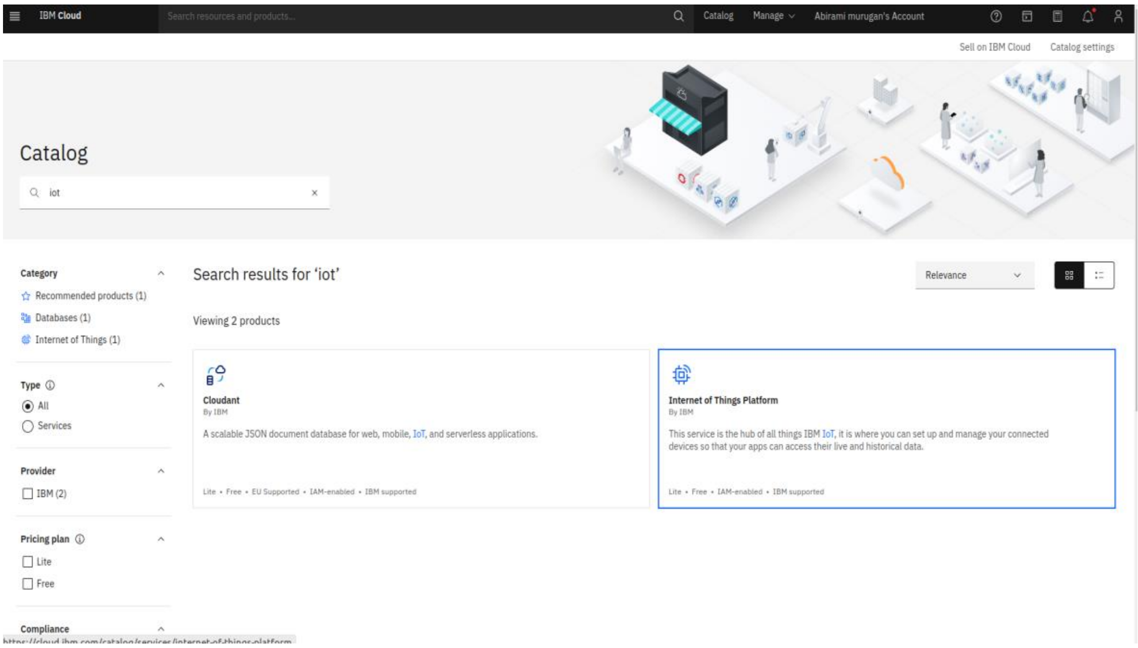
Steps to create a IoT device

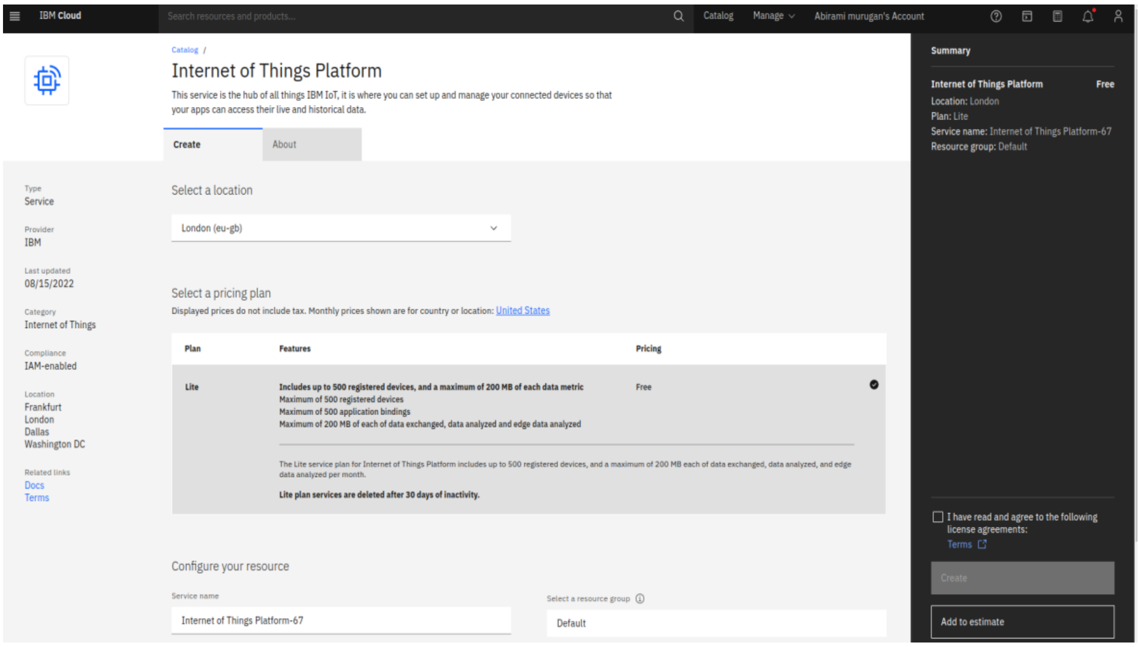


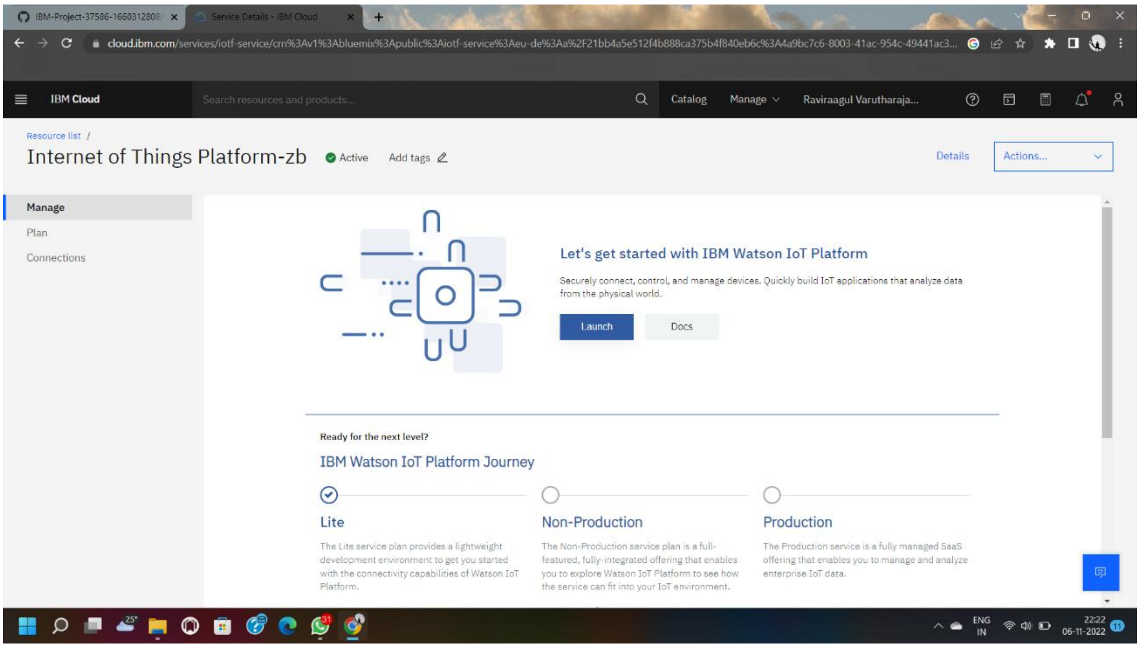


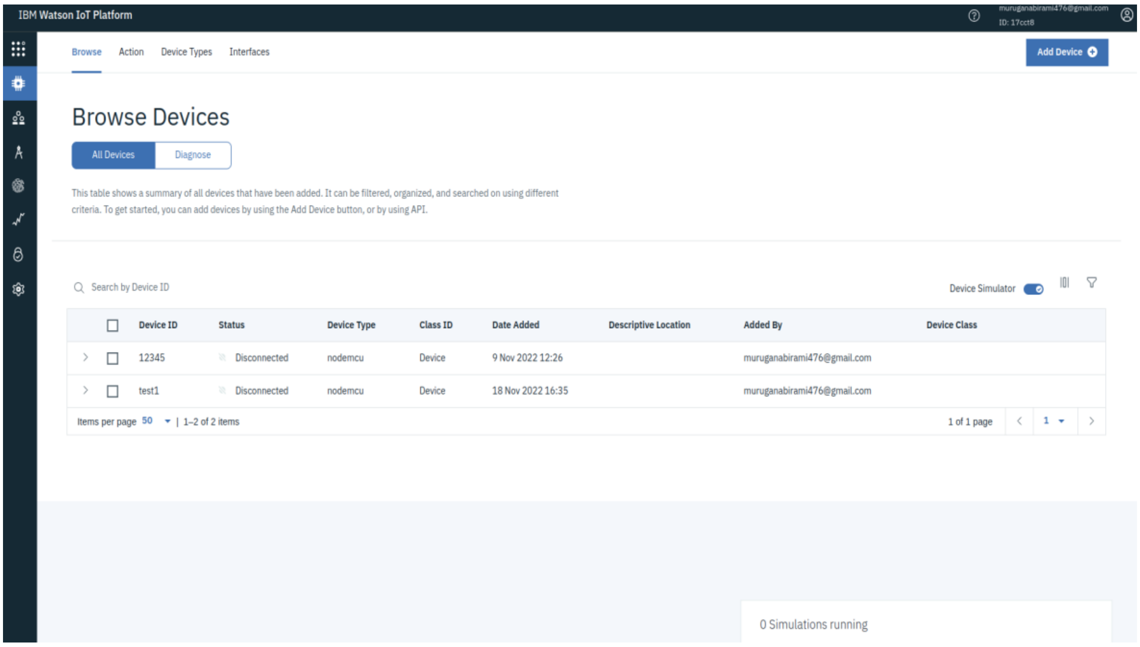

IBM Cloud
Search resources and products...
Catalog
Manage
Abirami murugan's Account
Account
Account resources
Resource groups
Cloud Foundry orgs
Licenses and entitlements
Tags
Dashboards
Account settings
IBM Cloud Shell settings
Notification distribution list
Classic infrastructure
Subscriptions
Audit log
Company information
Account
Account settings
Upgrade your account, activate special features, and redeem discounts.
Resource groups
Group your resources so you can quickly assign users access to multiple resources at a time. Resource groups are available for all resources that are managed by IBM Cloud Identity and Access Management.
Cloud Foundry orgs
Adjust resource quotas, users, spaces, and custom domains.
Licenses and entitlements
Assign licenses to your account for account members to use when installing software.
Tags
Use tags to manage your resources.
Dashboards
Create, manage, and share customized dashboards. Use a dashboard to group resources for specific projects or teams.
Audit log
View the actions that have been performed on your classic infrastructure resources. Available for all resources managed by Cloud Foundry.
Company information
View company contacts, company information, and account owner details.
Classic Subscriptions
Notifications Distribution List
IBM Cloud Shell settings

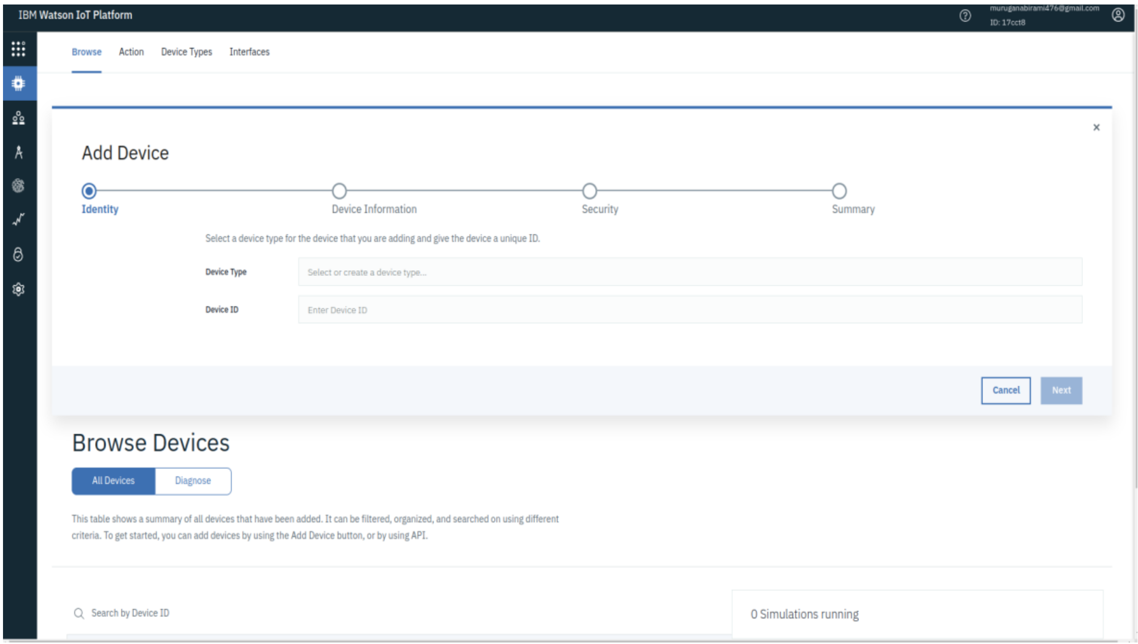

IBM Cloud
Search resources and products...
Catalog
Manage
Abirami murugan's Account
Search the catalog...
Sell on IBM Cloud
Catalog settings
Category
Recommended products (6)
Compute (30)
Containers (9)
Networking (29)
Storage (19)
AI / Machine Learning (17)
Analytics (10)
Blockchain (1)
Databases (28)
Developer tools (25)
Logging and monitoring (3)
Migration (7)
Integration (10)
Internet of Things (1)
Security (25)
Mobile (1)
Type
All
Services
Software
Professional services
Viewing 206 products
Alphabetically
Analytics Engine
By IBM
Submit your Apache Spark applications as needed and customize the Spark runtimes to satisfy the requirements of your application.
AnonTech ViziVault Platform
By Anon Technology, Inc.
Manage personal information as-a-service safely, securely, and in compliance with data privacy regulations using ViziVault
API Connect
By IBM
An enterprise-grade platform for creating, securing, managing, sharing, monetizing, and analyzing custom APIs located on-premises and on the cloud.
App Configuration
By IBM
Centralized, in-flight configuration for web and mobile applications and distributed environments.
App Connect
By IBM
Connect your applications, automate tasks, and improve productivity
App ID
By IBM
User Authentication and User Profiles for your apps.
Bare Metal Servers for Classic
By IBM
IBM Cloud Bare Metal Servers provide performance, flexibility, on-demand provisioning, and control.
Bare Metal Servers for VPC
By IBM
Leverage x86 servers with physical single tenancy and high performance offering end user virtualization.
Block Storage
By IBM
Persistent iSCSI based storage with high-powered performance and capacity up to 12TB.
Block Storage for VPC
By IBM
Persistent storage for use as boot and data storage for Virtual Servers in a VPC network.
Block Storage Snapshots for VPC
By IBM
Regional Snapshot Service to securely back up block storage data to IBM Cloud Object Storage
Blockchain Platform
By IBM
Welcome to the fast, flexible way to build, operate, and grow blockchain solutions.


IBM Cloud
Search resources and products...
Catalog
Manage
Abirami murugan's Account
Sell on IBM Cloud
Catalog settings
Catalog
iot
Category
Recommended products (1)
Databases (1)
Internet of Things (1)
Type
All
Services
Provider
IBM (2)
Pricing plan
Lite
Free
Compliance
Search results for 'iot'
Relevance
Viewing 2 products
Cloudant
By IBM
A scalable JSON document database for web, mobile, IoT, and serverless applications.
Lite • Free • EU Supported • IAM-enabled • IBM supported
Internet of Things Platform
By IBM
This service is the hub of all things IBM IoT, it is where you can set up and manage your connected devices so that your apps can access their live and historical data.
Lite • Free • IAM-enabled • IBM supported

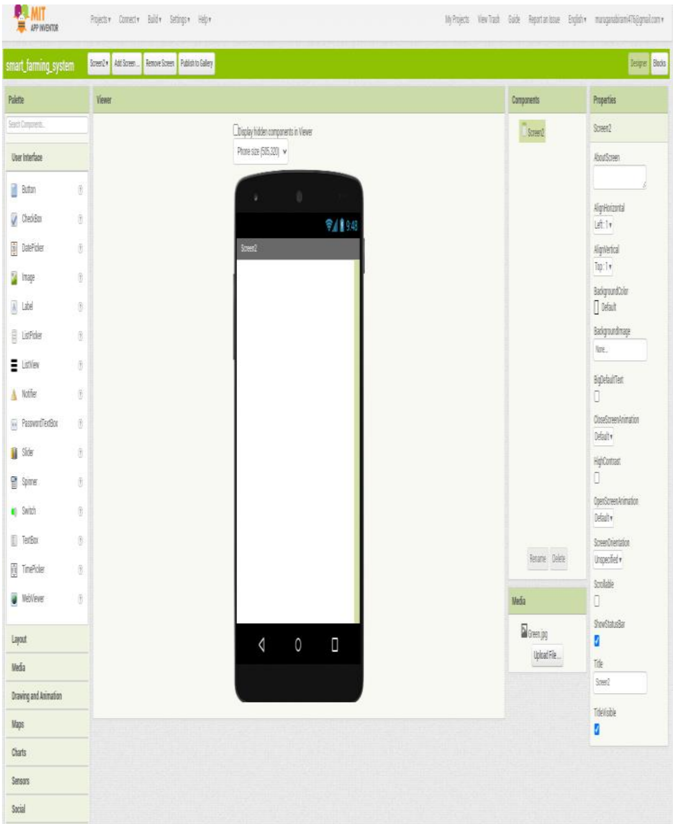

IBM Cloud
Search resources and products...
Catalog
Manage
Abirami murugan's Account
Catalog /
Internet of Things Platform
This service is the hub of all things IBM IoT, it is where you can set up and manage your connected devices so that your apps can access their live and historical data.
Create
About
Type
Service
Provider
IBM
Last updated
08/15/2022
Category
Internet of Things
Compliance
IAM-enabled
Location
Frankfurt
London
Dallas
Washington DC
Related links
Docs
Terms
Select a location
London (eu-gb)
Select a pricing plan
Displayed prices do not include tax. Monthly prices shown are for country or location: United States
Plan
Features
Pricing
Lite
Includes up to 500 registered devices, and a maximum of 200 MB of each data metric
Maximum of 500 registered devices
Maximum of 500 application bindings
Maximum of 200 MB of each of data exchanged, data analyzed and edge data analyzed
Free
The Lite service plan for Internet of Things Platform includes up to 500 registered devices, and a maximum of 200 MB each of data exchanged, data analyzed, and edge data analyzed per month.
Lite plan services are deleted after 30 days of inactivity.
Configure your resource
Service name
Internet of Things Platform-67
Select a resource group
Default
Summary
Internet of Things Platform
Free
Location: London
Plan: Lite
Service name: Internet of Things Platform-67
Resource group: Default
I have read and agree to the following license agreements:
Terms
Create
Add to estimate

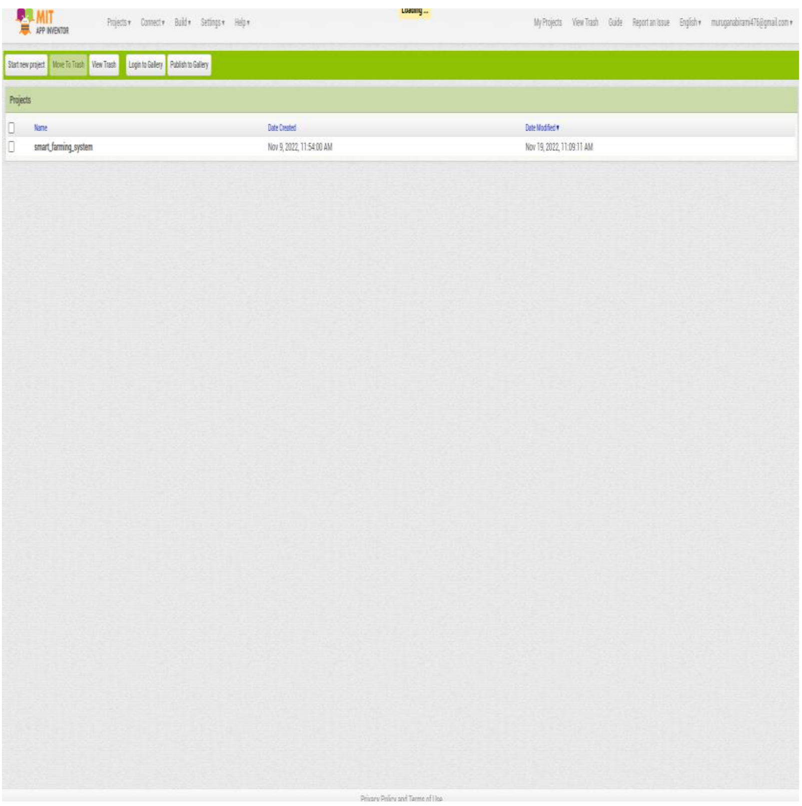

IBM Cloud
Search resources and products...
Catalog
Manage
Raviraagul Varutharaja...
Resource list /
Internet of Things Platform-zb
Active
Add tags
Details
Actions...
Manage
Plan
Connections
Let's get started with IBM Watson IoT Platform
Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.
Launch
Docs
Ready for the next level?
IBM Watson IoT Platform Journey
Lite
The Lite service plan provides a lightweight development environment to get you started with the connectivity capabilities of Watson IoT Platform.
Non-Production
The Non-Production service plan is a full-featured, fully-integrated offering that enables you to explore Watson IoT Platform to see how the service can fit into your IoT environment.
Production
The Production service is a fully managed SaaS offering that enables you to manage and analyze enterprise IoT data.

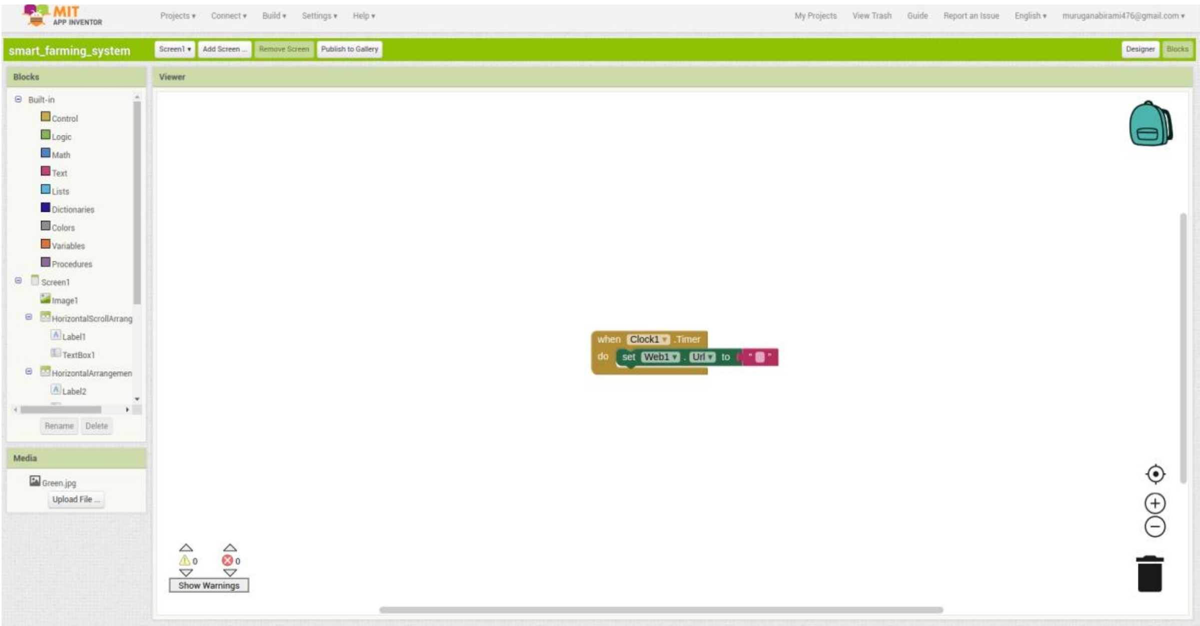

IBM Watson IoT Platform
Browse
Action
Device Types
Interfaces
Add Device
Browse Devices
All Devices
Diagnose
This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.
Search by Device ID
Device Simulator
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
Added By
Device Class
12345
Disconnected
nodemcu
Device
9 Nov 2022 12:26
muruganabirami476@gmail.com
test1
Disconnected
nodemcu
Device
18 Nov 2022 16:35
muruganabirami476@gmail.com
Items per page 50 | 1–2 of 2 items
1 of 1 page
0 Simulations running

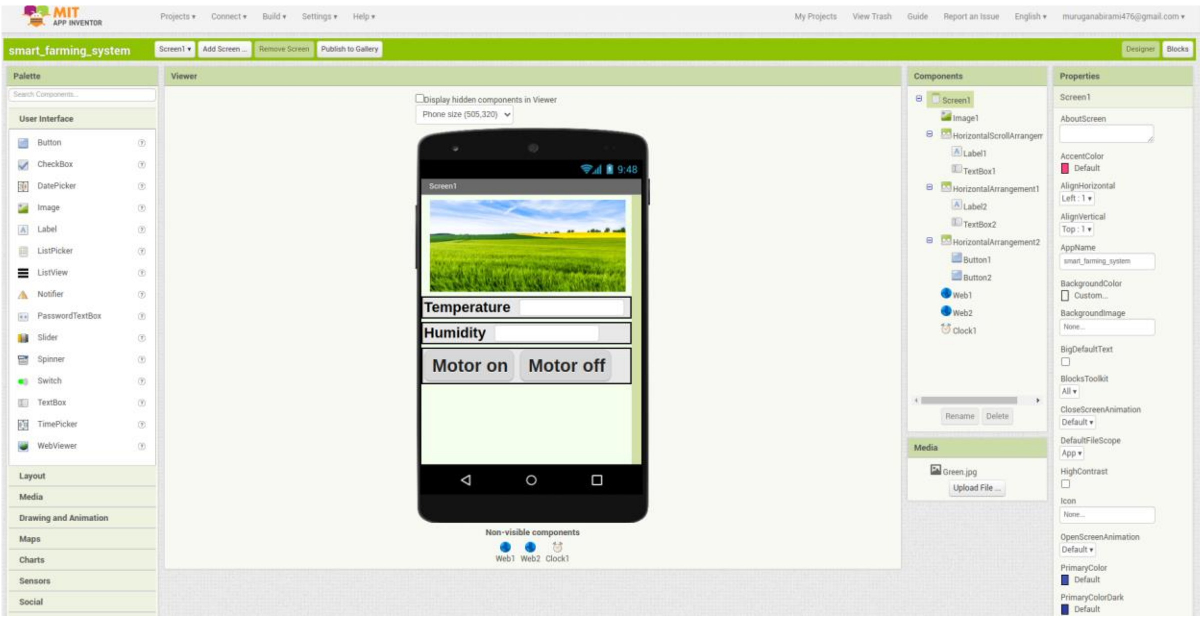

IBM Watson IoT Platform
ID: 17cct8
Browse
Action
Device Types
Interfaces
Add Device
Identity
Device Information
Security
Summary
Select a device type for the device that you are adding and give the device a unique ID.
Device Type
Select or create a device type...
Device ID
Enter Device ID
Cancel
Next
Browse Devices
All Devices
Diagnose
This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.
Search by Device ID
0 Simulations running

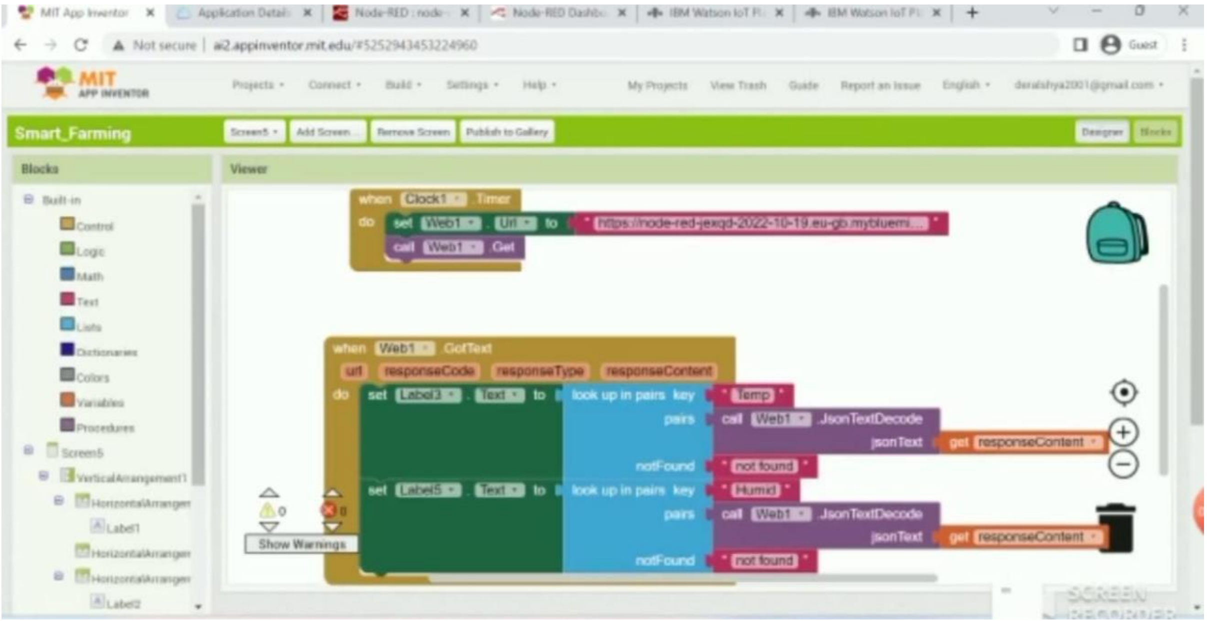
MIT app inventer





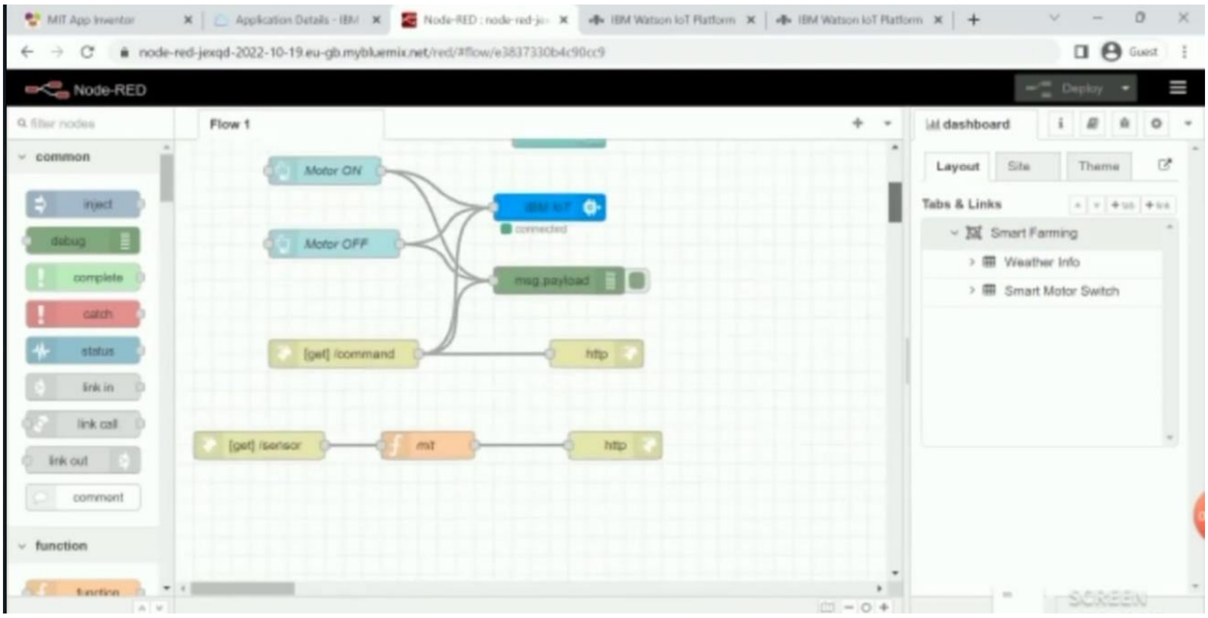

MIT APP INVENTOR
Projects ▾ Connect ▾ Build ▾ Settings ▾ Help ▾
My Projects View Trash Guide Report an Issue English ▾ muruganabirami476@gmail.com ▾
smart_farming_system
Screen1 ▾ Add Screen ... Remove Screen Publish to Gallery
Designer Blocks
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen1
Image1
HorizontalScrollArrang
Label1
TextBox1
HorizontalArrangemen
Label2
Rename Delete
Media
Green.jpg
Upload File ...
when Clock1 .Timer
do set Web1 . Url to
Show Warnings


MIT APP INVENTOR
Projects ▾ Connect ▾ Build ▾ Settings ▾ Help ▾
My Projects View Trash Guide Report an Issue English ▾ muruganabirami476@gmail.com ▾
smart_farming_system
Screen1 ▾ Add Screen ... Remove Screen Publish to Gallery
Designer Blocks
Palette
Viewer
Components
Properties
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Media
Drawing and Animation
Maps
Charts
Sensors
Social
Display hidden components in Viewer
Phone size (505,320)
9:48
Screen1
Temperature
Humidity
Motor on
Motor off
Non-visible components
Web1 Web2 Clock1
Screen1
Image1
HorizontalScrollArrangem
Label1
TextBox1
HorizontalArrangement1
Label2
TextBox2
HorizontalArrangement2
Button1
Button2
Web1
Web2
Clock1
Rename Delete
Media
Green.jpg
Upload File ...
Screen1
AboutScreen
AccentColor
Default
AlignHorizontal
Left : 1 ▾
AlignVertical
Top : 1 ▾
AppName
smart_farming_system
BackgroundColor
Custom...
BackgroundImage
None...
BigDefaultText
BlocksToolkit
All ▾
CloseScreenAnimation
Default ▾
DefaultFileScope
App ▾
HighContrast
Icon
None...
OpenScreenAnimation
Default ▾
PrimaryColor
Default
PrimaryColorDark
Default

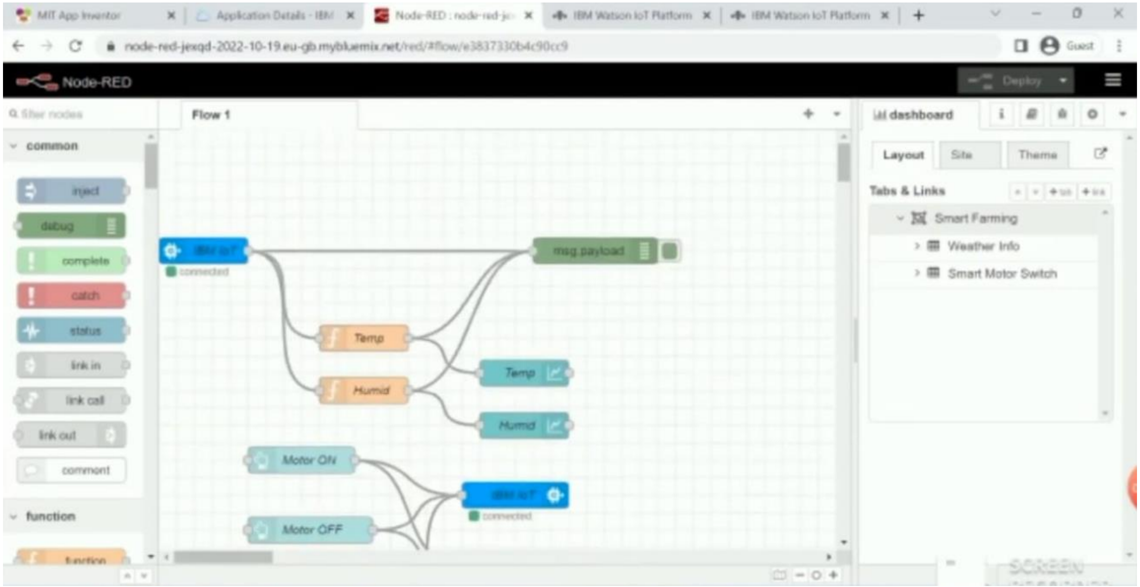


MIT App Inventor
ai2.appinventor.mit.edu/#5252943453224960
Smart_Farming
Blocks
Viewer
when Clock1 .Timer
do set Web1 . Url to "https://node-red-jexqd-2022-10-19.eu-gb.mybluemi..."
call Web1 .Get
when Web1 .GotText
url responseCode responseType responseContent
do set Label3 . Text to look up in pairs key "Temp"
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound "not found"
set Label5 . Text to look up in pairs key "Humid"
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound "not found"
Show Warnings

Node red



Node-RED
Deploy
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
Motor ON
Motor OFF
IBM IoT
connected
msg.payload
[get] /command
http
[get] /sensor
mit
http
dashboard
Layout
Site
Theme
Tabs & Links
Smart Farming
Weather Info
Smart Motor Switch



6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint planning

Product Backlog, Sprint Schedule, and Estimation:

Sprint	**Functional Requiremen t(Epic)**	**User Story Number**	**User Story /Task**	**Story Points**	**Priority**	**Team Members**
Sprint-1	Simulatio ncreation	USN-1	Connect Sensors Wi-fi Module with python code	2	High	Ravi Raagul V, Sanjay B, Vignesh P, Dinesh Kumar S
Sprint-2	Software	USN-2	Creating device in the IBM Watson IoT platform, workflow for IoT scenarios using NodeRed	2	High	Ravi Raagul V, Sanjay B, Vignesh P, Dinesh

						Kumar S
Sprint-3	MIT App Inventor	USN-3	To Develop an application for the Smart farmer project using MIT App Inventor	2	High	Ravi Raagul V, Sanjay B, Vignesh P, Dinesh Kumar S
Sprint-3	Dashboard	USN-3	Design the Modules and test the app	2	High	Ravi Raagul V, Sanjay B, Vignesh P, Dinesh Kumar S
Sprint-4	Web UI	USN-4	To make the user to interact with software.	2	High	Ravi Raagul V, Sanjay B, Vignesh P, Dinesh Kumar S

6.2 Sprint delivery schedule

Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Sprint Release Date (Actual)
Sprint-1	20	7 Days	30 Oct 2022	06 Nov 2022	01 Nov 2022
Sprint-2	20	9 Days	01 Nov 2022	09 Nov 2022	05Nov 2022
Sprint-3	20	7 Days	06 Nov 2022	13 Nov 2022	10 Nov 2022
Sprint-4	20	5 Days	11 Nov 2022	15 Nov 2022	12 Nov 2022

7. Coding and Solutioning

7.1 FEATURE 1

This code is used for connect the IBM Watson Iot platform.

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random


#Provide your IBM Watson Device Credentials

organization = "bnsfkk"

deviceType ="Weather_Monitor"

deviceId = "weather"

authMethod = "token"
```

```
authToken = "weatherravi"

# Initialize GPIO

temp=random.randint(0,100)
pulse=random.randint(0,100)
oxygen= random.randint(0,100)
lat = 17
lon = 18

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    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

    data = {"d":{ 'temp' : temp, 'pulse': pulse ,'oxygen': oxygen,"lat":lat,"lon":lon}} #print
data
def myOnPublishCallback():
    print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % pulse, "to
IBM Watson")



success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
if not success:
    print("Not connected to IoTF")
    time.sleep(1)

deviceCli.commandCallback = myCommandCallback



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

IBM Watson IoT Platform..

7.2 FEATURE 2

Connecting Sensors with wokwi using C++

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN 15     // what pin we're connected to
#define DHTTYPE DHT22   // define type of sensor DHT 11
#define LED 2

DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of dht connected

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

//-------credentials of IBM Accounts------

#define ORG "i3869j"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
```

```
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token
String data3;
float h, t;

//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id

//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by passing parameter like server id,portand wificredential

void setup()// configureing the ESP32
{
  Serial.begin(115200);
  dht.begin();
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}
```

```
void loop()// Recursive Function
{

  h = dht.readHumidity();
  t = dht.readTemperature();
  Serial.print("temp:");
  Serial.println(t);
  Serial.print("Humid:");
  Serial.println(h);

  PublishData(t, h);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}



/*....................................retrieving to Cloud..............................*/

void PublishData(float temp, float humid) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"temp\":";
  payload += temp;
  payload += "," "\"Humid\":";
  payload += humid;
```

```
  payload += "}";


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


  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it will print
publish ok in Serial monitor or else it will print publish failed
  } else {
    Serial.println("Publish failed");
  }

}

void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }

     initManagedDevice();
     Serial.println();
  }
}
void wificonnect() //function defination for wificonnect
```

```
{
  Serial.println();
  Serial.print("Connecting to ");

   WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{

  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
```

```
  for (int i = 0; i < payloadLength; i++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);
  if(data3=="lighton")
  {
Serial.println(data3);
digitalWrite(LED,HIGH);
  }
  else
  {
Serial.println(data3);
digitalWrite(LED,LOW);
  }
data3="";
}
```

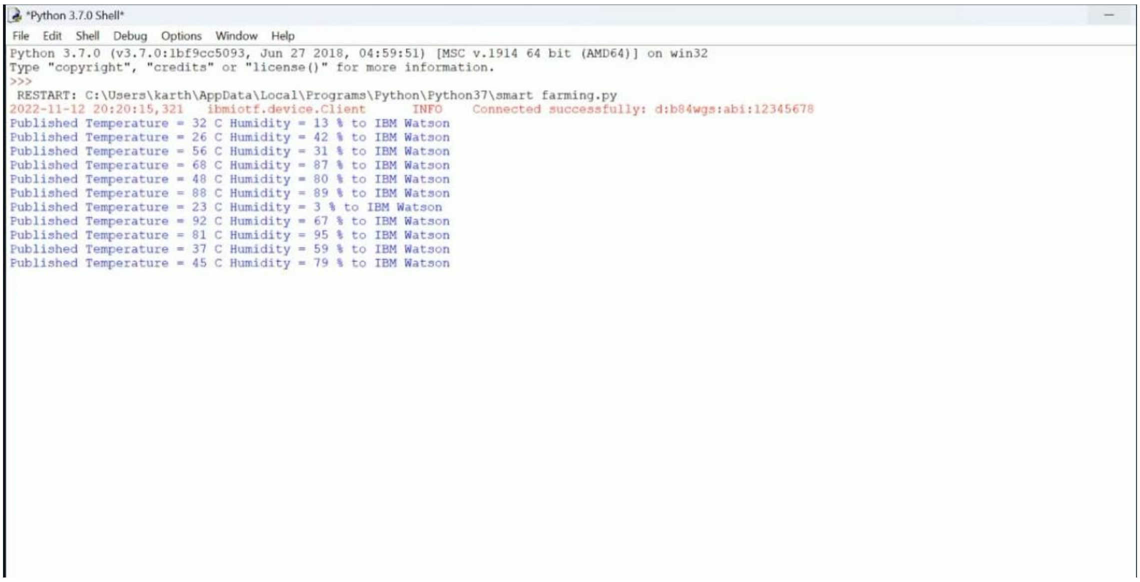
Program for resistor in Wokwi

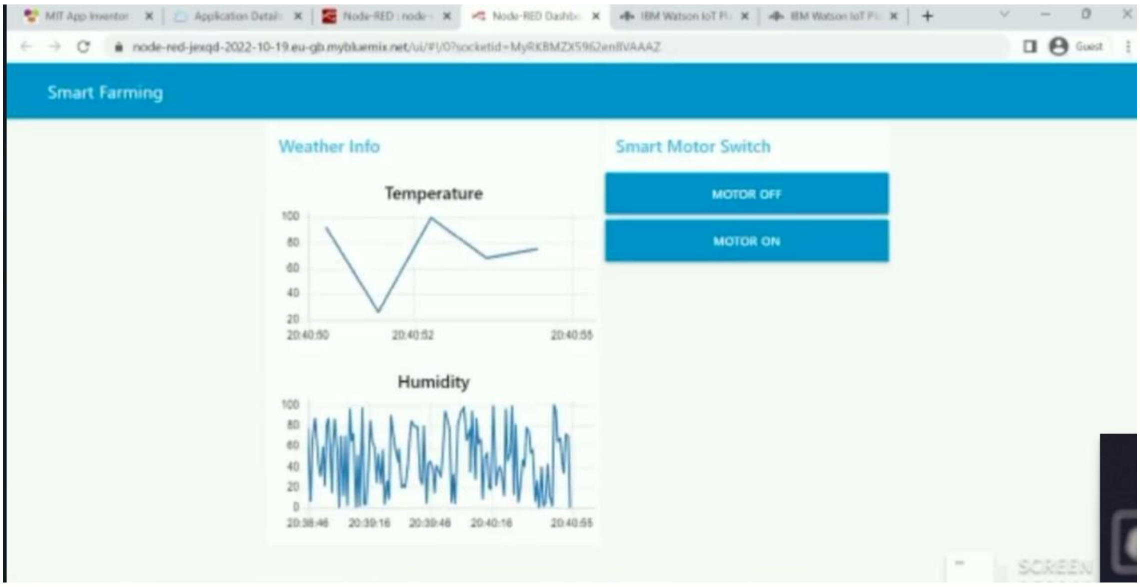

```
{
  "version": 1,
  "author": "Anonymous maker",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 4.8, "left": -127.69, "attrs": {}
},
    { "type": "wokwi-dht22", "id": "dht1", "top": -76.72, "left": 137.76, "attrs": {} },
    {
      "type": "wokwi-led",
```

```
      "id": "led1",
      "top": -16.04,
      "left": 21.83,
      "attrs": { "color": "red" }
    },
    {
      "type": "wokwi-resistor",
      "id": "r1",
      "top": 41.63,
      "left": 48.17,
      "attrs": { "value": "100" }
    }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "dht1:VCC", "esp:3V3", "red", [ "v0" ] ],
    [ "dht1:GND", "esp:GND.1", "black", [ "v0" ] ],
    [ "led1:A", "r1:1", "green", [ "v0" ] ],
    [ "led1:C", "esp:GND.1", "black", [ "v0" ] ],
    [ "dht1:SDA", "esp:D15", "green", [ "v101.76", "h-2.06" ] ],
    [ "r1:2", "esp:D2", "green", [ "v80.85", "h-3.49" ] ]
  ]
}
```

9. RESULTS

9.1 Performance metrics





10. ADAVANTAGES & DISADVANTAGES

Adavantages

1. Intelligent data collection

Sensors installed on IoT devices are able to collect a large volume of useful information for farmers. As we mentioned below, some examples are climatic conditions, soil quality and plantation progress. Such data can be used to monitor the status of the farm, as well as the performance of workers and the efficiency of the appliances.

2. Waste reduction

With greater production control, IoT in agriculture facilitates cost-efficient management. From smart devices, producers can more accurately identify any anomaly in the crop, for example. Consequently, it is easier to effectively prevent any infestation that will harm yields.

In addition, one can also save in the process of irrigation and fertilization. After all, there are sensors installed in the agricultural machinery, which can generate a lot of information about the soil.

Another advantage is the possibility of programming the sensors to notify about the ideal harvest time. In this way, waste is avoided in the crop.

3. Process automation

From smart devices, it is also possible to automate several stages of the production process, such as irrigation, fertilization or pest control.

By getting rid of these manual interventions, you get higher accuracy, better product quality and save resources. Thus, agriculture ensures higher standards of quality in the harvest.

4. Animal monitoring

With IoT, farmers can monitor the health of farm animals closely, even if they are physically distant. Thus, one can reduce the search time of cows and sheep in the pasture, for example, if they are part of the herd.

It is also possible to monitor the pregnancy of these animals and identify which of them are sick. If so, the sensor sends a notification to the producer to contact a veterinarian.

5. Competitive advantage

One more benefit is increased harvest —as we mentioned in the above topics—that yields a competitive advantage in business. To exemplify, we can mention preventive maintenance.

Once sensors are installed on a tractor, for example, the collected data can quickly notify whenever any technical failure arises.

DISADVANTAGES

1.Security: As the IoT systems are interconnected and communicate over networks. The system offers little control despite any security measures, and it can be lead the various kinds of network attacks.

2.Privacy: Even without the active participation on the user, the IoT system provides substantial personal data in maximum detail.

3.Complexity: The designing, developing, and maintaining and enabling the large technology to IoT system is quite complicated.

4. The Cost Involved in Smart Agriculture

While the use of smart technology in agriculture is impressive, it does incur a lot of costs. As said earlier, if the devices are to be altered according to the level of the farmers, it will involve a lot of money to transform these types of equipment. This, on the other hand, means that the process will cost huge money. Since the farming industry does not see higher profits, huge investments in this space are unlikely. Even after the altering of machines, there are chances where the farmers might tend to operate the machines wrongly causing it to damage or send it to repair. Since these pieces of equipment are already costly, repairing it or replacing it will again cost a lot of money. The cost of maintenance becomes high whether there is a repair or not.

5. There could be wrong Analysis of Weather Conditions

- In the case of agriculture, most of the process is dependent on weather conditions.
- It is a natural phenomenon which in spite of the updated technology can become unpredictable.
- There is no force which can change or control the weather conditions such as rain, sunlight, drought etc.
- Even when the smart systems are in place, the importance of natural occurrences cannot be changed.
- There is an issue where the machines used in smart agriculture can impact the environment in a negative manner.
- Since technology involves a lot of machines, there are chances where the data might get wrong at times.
- If there are faulty data processing equipment or sensors then it will lead to the situation where the wrong decisions are taken.
- This will lead to the overuse of resources like fertilizers or water.

11. CONCLUSION

From our results and literature survey of other papers, we saw that the hardware and materials we used to develop our protype allowed us to make an efficient and accurate, as well as cheap product for farmers. Which was economical and easily installable for farmers as well. Thus, we can conclude that this porotype will definitely help farmers in small farmland to effectively monitor their crops with the user-friendly app and other alert mean.

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. This may result in better access to capital in some cases and to specific support of investments in others. Moreover, the support of cooperatively used farm-monitoring technology (e.g., jointly owned unmanned aerial vehicles monitoring fields of entire villages) or investments in education and training may also support the sustainable use of these technologies. In all cases, however, the policy environment should provide a clear, legal setting that allows for effective ownership and user rights. IoT in agriculture only tends to evolve. As you can see, the benefits and facilities generated for producers allowed work in the field to be done more efficiently, quickly and with less resource expense.

12. FUTURE SCOPE

Future of IOT in smart farming

With the exponential growth of world population, according to the **UN Food and Agriculture Organization**, the world will need to produce 70% more food in 2050, shrinking agricultural lands, and depletion of finite natural resources, the need to enhance farm yield has become critical. Limited availability of natural resources such as fresh water and arable land along with slowing yield trends in several staple crops, have further aggravated the problem. Another impeding concern over the farming industry is the shifting structure of agricultural workforce. Moreover, agricultural labour in most of the countries has declined. **As a result of the declining agricultural workforce, adoption of internet connectivity solutions in farming practices has been triggered, to reduce the need for manual labour.**

IoT solutions are focused on helping farmers close the supply demand gap, by ensuring high yields, profitability, and protection of the environment. The approach of using IoT technology to ensure optimum application of resources to achieve high crop yields and reduce operational costs is called precision agriculture. IoT in agriculture technologies comprise specialized equipment, wireless connectivity, software and IT services.

BI Intelligence survey expects that the adoption of **IoT devices in the agriculture industry will reach 75 million in 2020**, growing 20% annually. At the same time, **the global smart agriculture market size is expected to triple by 2025**, reaching $15.3 billion (compared to being slightly over $5 billion back in 2016).

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.

Agricultural Drones Ground-based and aerial-based drones are being used in agriculture in order to enhance various agricultural practices: crop health assessment, irrigation, crop monitoring, crop spraying, planting, and soil and field analysis.

Livestock tracking and geofencing Farm owners can utilize wireless IoT applications to collect data regarding the location, well-being, and health of their cattle. This information helps to prevent the spread of disease and also lowers labour costs.

Smart Greenhouses A smart greenhouse designed with the help of IoT intelligently monitors as well as controls the climate, eliminating the need for manual intervention.

Predictive analytics for smart farming Crop predication play a key role, it helps the farmer to decide future plan regarding the production of the crop, its storage, marketing techniques and risk management. To predict production rate of the crop artificial network use information collected by sensors from the farm. This information includes parameters such as soil, temperature, pressure, rainfall, and humidity. The farmers can get an accurate soil data either by the dashboard or a customized mobile application.

Farmers have started to realize that the IoT is a driving force for increasing agricultural production in a cost-effective way.

13. APPENDIX

Source Code

https://wokwi.com/projects/345395196387656275

GitHub & Project Demo Link

Github link:

https://github.com/IBM-EPBL/IBM-Project-32455-1660209991

Stimulation using node red:

https://drive.google.com/file/d/10NNmSAlIlj-O5ePuLhJXwVTKE3RR7HP0/view?usp=share_link