IOT BASED SMART FARMING APPLICATION

TEAM ID : PNT2022TMID49209

TEAM LEADER : SUBHASHINI.N

TEAM MEMBER : DEEPIKA.S

TEAM MEMBER : ANUPRIYA.P

TEAM MEMBER : NAGALAKSHMI.S

ABSTRACT

Internet of things is the each and every field of common man’s life by making every thing bright and intelligent.internet of things refers to a network of things which make a self configuring network. the main intension of smart farming is to improve agriculture yield with few resources and less labour efforts.

TABLE OF CONTENT

INTRODUCTION

1.1PROJECT OVERVIEW

Smart Farm is all about intelligent irrigation and smarter farming. We have developed a comprehensive state-of-the-art monitoring and control system that delivers the information farmers need on a daily and even hourly basis. Our patented wireless communications system integrates smart sensors and pump modules with RF wireless technology for remotefield monitoring and well pump control.

1.2PURPOSE

We are increasingly reliant on technology in today's world for almost everything we do. And when it comes to farming, the reliance is only going to grow in the years ahead. Thanks to the Internet of Things (IoT) more and more farmers are using smart technology to increase productivity and efficiency.

2. LITERATURE SURVEY

2.1EXISTING PROBLEM

India is agriculture sector, on either side, is losing ground every day, affecting the Ecosystem\’s output capacity. In order to restore vitality and put agriculture back on a path of higher growth, there is a growing need to resolve the issue. a large-scale agricultural system necessitates a great deal of upkeep, knowledge, and oversight.

The IoT is a network of interconnected devices that can transmit and receive data over The internet and carry out tasks without human involvement. Agriculture provides a Wealth of data analysis parameters, resulting in increased crop yields. The use of IoT Devices in smart farming aids in the modernization of information and communication. For better crop growth moisture, mineral, light and other factors can be assumed. This Research looks into a few of these characteristics data analysis with the goal of assisting users in making better agricultural decisions using IoT. The technique is intended to help farmers increase their agricultural output.

2.2REFERENCES

- Yuridia Muhammad, Muhammad Azri Asyraf Mohd Hafez, Nor Adni MatLehZariah Moha Yusoff , Shabina Abd Hamid [1] The term “Internet of Things” refers To the connection of objects, equipment, vehicles, and other electronic devices to a Network for the purpose of data exchange (IoT).

- The Internet of Things (IoT) is in creasingly being utilized to connect objects and collect data. As a result, the Internet of Things’ use in agriculture is crucial. The idea behind the project is to create a smart Agriculture system that is connected to the internet of things.

- In Malaysia, employing IoT-based irrigation Systems saves roughly 24.44 percent per year when compared to traditional irrigation Systems. This would save money on labour expenditures while also preventing water Waste in daily needs.

- H.G.C.R. Laksiri, H.A.C. Dharmagunawardhana, J.V. Wijayakulasooriya

Development of an effective loT-based smart irrigation system is also Crucial Demand for farmers in the field of agriculture.

This research develops a low-cost, Weather-based smart watering system. To begin, an effective drip irrigation system Must be devised that can automatically regulate water flow to plants based on soil Moisture levels.

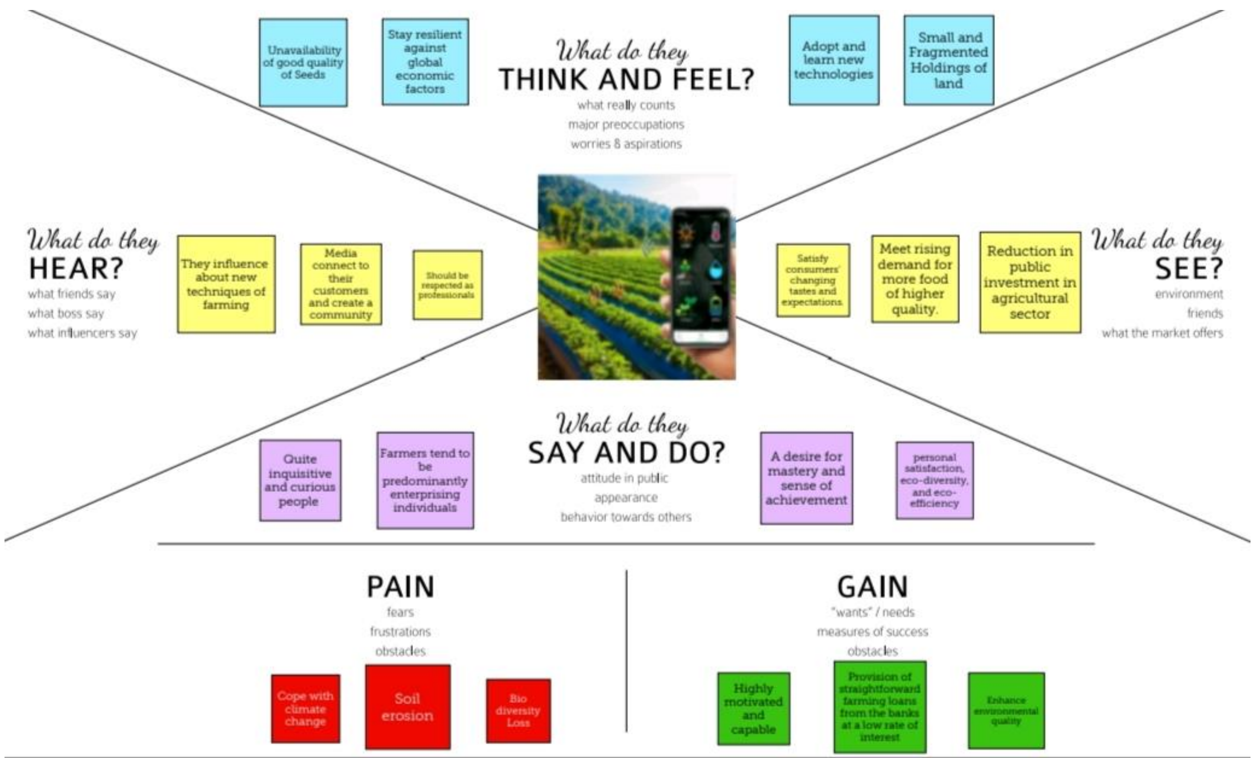
2.3PROBLEM STATEMENT DEFINITION

The India is an agricultural country. Nowadays, at regular intervals the lands are manually irrigated by the farmers. There is a chance that the water consumption will be higher or that the time it takes for the water to reach the destination will be longer, resulting in crop dryness. Real-time temperature and humidity monitoring is crucial in Many agricultural disciplines. However, the old method of wired detection control is Inflexible, resulting in several application limitations. This project achieves irrigation Automation as a crucial answer to this problem. This is accomplished with the aid of a Raspberry Pi, which controls the moisture and temperature sensors based on the input Provided. Moisture sensors are used in the construction of an automated plant Watering system for this purpose. The main aim of our project is to reduce the Complexity of supervision and to avoid the continuous monitoring. We can accomplish Smart agriculture using our system. This system includes IoT-based agricultural Monitoring. The Internet of Things (IOT) is transforming the agriculture business and Addressing the enormous difficulties and huge obstacles that farmers confront today in the field. The soil moisture sensor is put into the soil to determine whether the soil is Wet or dry, and If the moisture level in the soil is low, the relay unit attached to the Motor switch must be monitored on a regular basis. When the soil is dry, it will turn on the motor, and when the soil is moist, it will turn off the engine.

3. IDEATION & PROPOSED SOLUTION

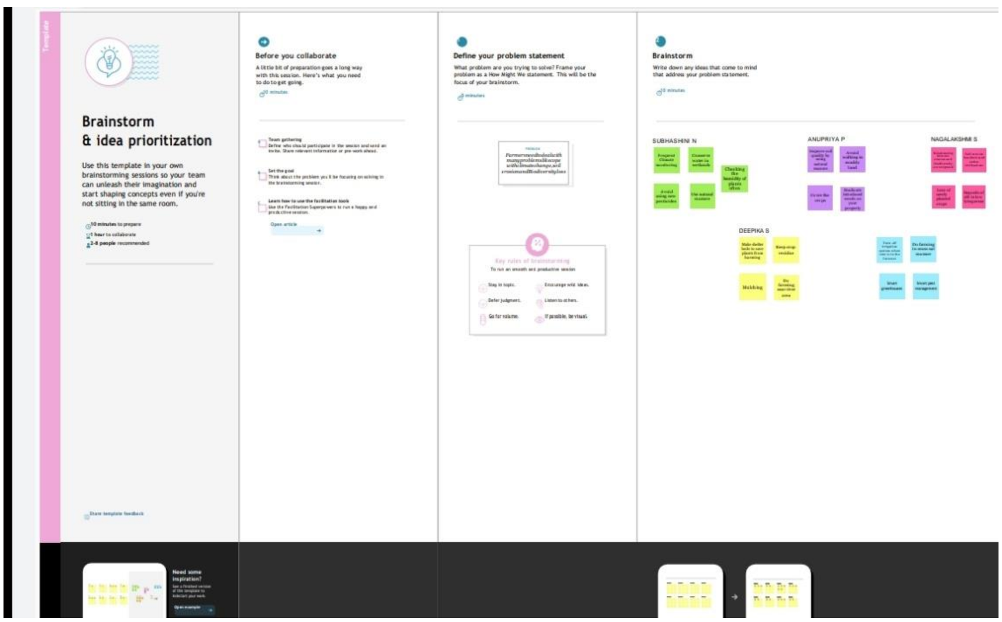
3.1EMPATHY MAP CANVAS

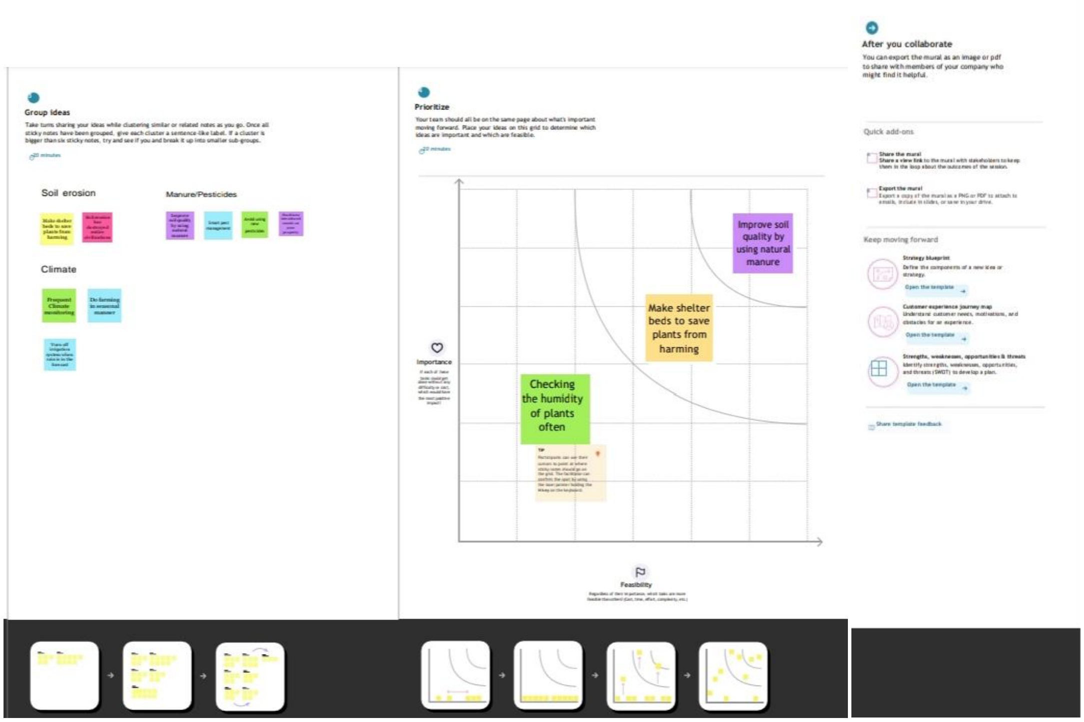
An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathymap was originally created by Dave Gray and has gained much popularitywithin the agile community.



3.2 IDEATION & BRAINSTORMING

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.



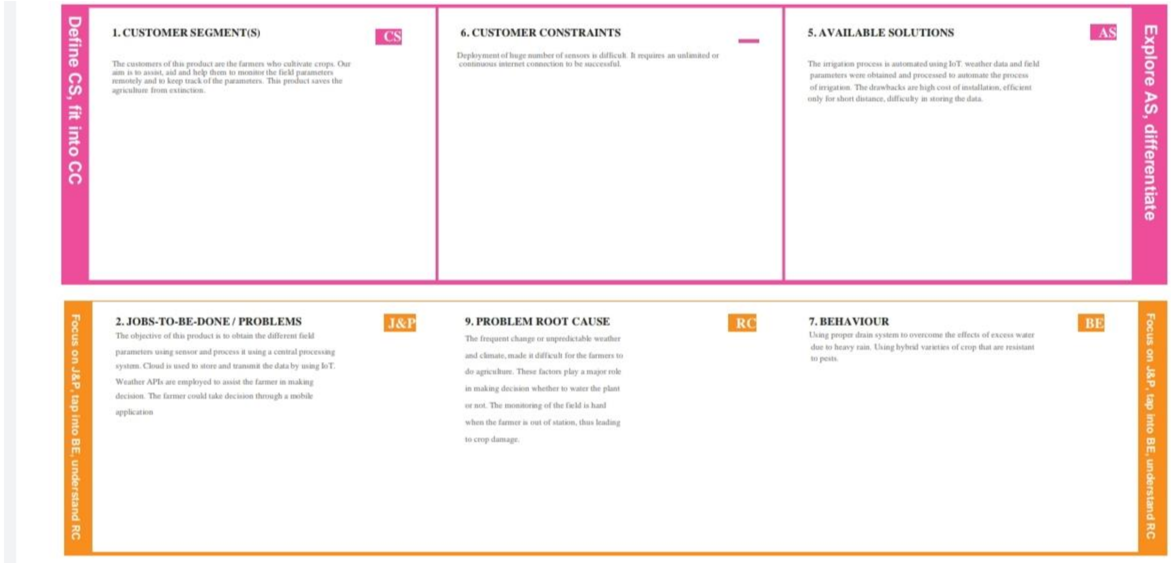


3.3PROPOSED SOLUTION

s.no	Parameter	Description
1	Problem Statement (Problem to be solved)	To provide efficient decision support system using wireless sensors network which handle different activities of farm and gives useful information related to soil moisture, Temparature and Humidity content.Due to the weather condition,water level increasing Farmers get lot of distractions which is not good for Agriculture.
2	Idea / Solution description	It is a network of different devices which make a self- configuring network. The new developments of Smart Farming with use of IoT, by day turning the face of conventional agriculture methods by not only making it optimal but also making it cost efficient for farmers and reducing crop wastage.
3	Novelty / Uniqueness	IoT based Smart Farming improves the entire Agriculture system by monitoring the field in real-time. With the help of sensors and interconnectivity, the Internet of Things in Agriculture has not only saved the time of the farmers but has also reduced the extravagant use of resources such as Water and Electricity

4	Social Impact / CustomerSatisfaction	Smart farming, the dependency on manual labor has reduced significantly. The processes like pest control, fertilizing, and irrigation are increasingly becoming automated, and farmers can control them remotely. The use of smart IoT sensors can maintain these processes, increasing crop production.
5	Scalability of the Solution	Scalability in smart farming refers to the adaptability of a system to increase the capacity, for example, the number of technology devices such as sensors and actuators, while enabling timely analysis.

3.4PROBLEM SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1FUNCTIONAL REQUIREMENT

These are the requirements that the end user specifically demands as basicfacilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected.

FR No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	EMAIL: Enter email address PASSWORD: Enter password
FR-2	User Confirmation	EMAIL: Enter email address PASSWORD: Enter password
FR-3	Log in to system	Serve authenticated content
FR-4	Manage Modules	Manage System Admins Manage Roles of User Manage User permission
FR-5	Check whether condition	Temperature monitoring status Humidity monitoring Status
FR-6	Log out	Exit

4.2NON FUNCTIONAL REQUIREMENT

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

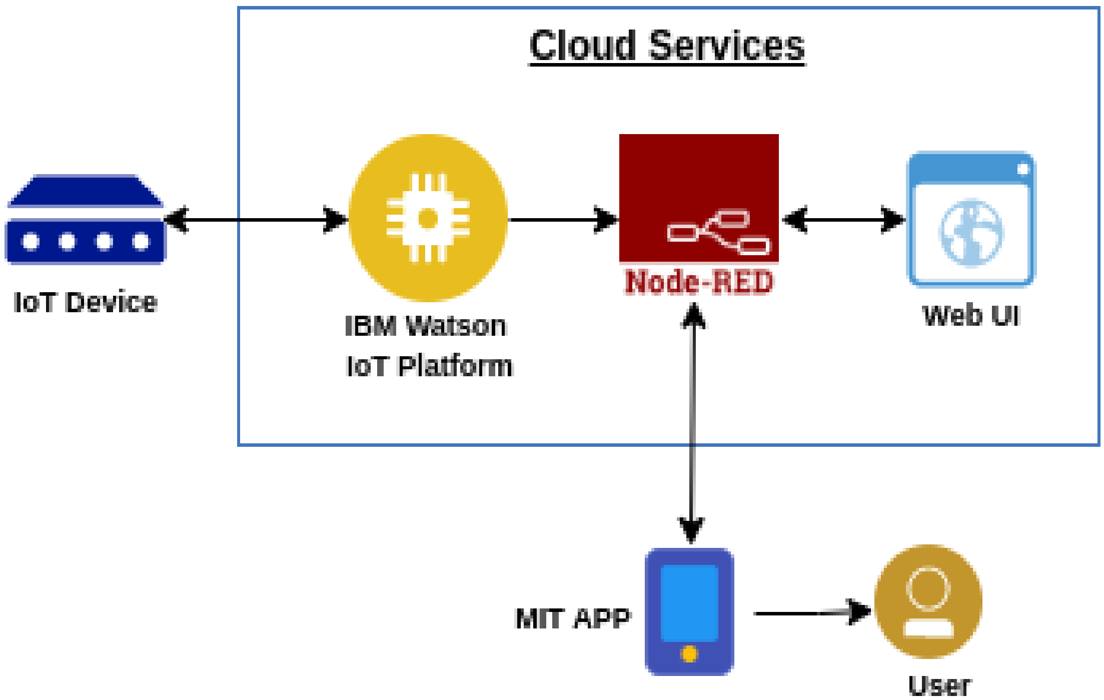
FR No.	Non-Functional Requirement	Description
NFR- 1	Usability	Usability includes easy understanding and learn ability, efficiency in use,remember ability, lack of errors in operation and subjective pleasure.
NFR- 2	Security	Sensitive and private data must be protected from their production until the decision-making and storage stages.

They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability
- Flexibility

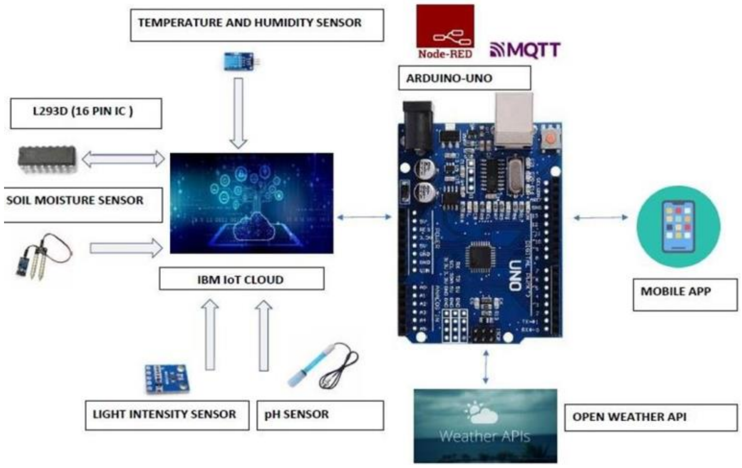
5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS



5.2SOLUTION &TECHNICAL ARCHITECTURE

- The different soil parameters temperature, soil moistures and then humidity are sensed using different sensors and obtained value is stored in the IBM B2 cloud.
- Arduino UNO is used as a processing Unit that process the data obtained from the sensors and whether data from the weather API.
- NODE-RED is used as a programming tool to write the hardware, software andAPIs. The MQTT protocol is followed for the communication.
- All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could make a sensor values. By using the app they can remotely operate the motor switch.



USER STORIES

User Type	Functional Requirement	User Story Number	User Story/Task	Acceptanc ecriteria	Prior ity	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password and confirmingmy password.	I can access myaccount/ dashboard	High	Sprint-1
	Permission	USN-2	As a user, I will receive confirmation email once I have registered for the application.	I can receive confirmatio n email & click confirm.	High	Sprint-1
Customer (Web user)	Login	USN-3	As a user, I can log into the application by entering email & password.	I can register &access the dashboard with Login	High	Sprint-2
	Check credentials	USN-4	As a user, I can register for the application through mobile application	Temperature and Humidity details	Mediu m	Sprint-1
	Dashboard	USN-5	As a user can view the dashboard and this dashboardinclude the check roles of access and then move to the manage modules.	I can view the dashboard in thissmart farming application system.	Mediu m	Sprint-1
Customer care Executive	MIT app	USN-6	To make the user to interactwith the software.	Database to store in cloud services.	High	Sprint-1

Administrator	IOT devices	USN-7	As a user once view the manage modules this describes the manage system admins and Manage Roles of user and etc..,		Mediu m	Sprint-1
	Log out	USN-8	Exit	Sign out	High	Sprint-1

Our system will store an entire growing season's worth of field-monitored, event-triggered, maintenance event and fault data locally on the base station. Reports can easily be generated and printed based on this database, including reports that provide the information a farmer needs to submit to government monitoring agencies.

Our customers own their data — and we will not disclose it to any outside entity without written permission. We will keep a copy of all data from our customers' systems for backup in case it is needed. We will compile data from multiple locations to be used in our regional data analytics package. No farmer-specific information will be disclosed in that software package.

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional requirement (epic)	User story number	User Story / Task	Story points	Priority
Sprint-1	Simulation creation	USN-1	Connect Sensors and Arduino with python code	2	High
Sprint-2	Software	USN-2	Creating device in the IBM Watson IoT platform, workflow for IoT scenarios using Node-Red	2	High
Sprint-3	MIT App Inventor	USN-3	Develop an application for the Smart farmer project using MIT App Inventor	2	High

7. CODING & SOLUTION

```
#include <ESP8266WiFi.h>
#include <DallasTemperature.h>
#include <OneWire.h>
#include “DHT.h”
#include “Adafruit_MQTT.h”
#include “Adafruit_MQTT_Client.h”
#include <ArduinoJson.h>
Const char *ssid =  “Galaxy-M20”;     // Enter your WiFi Name
Const char *pass = “ac312124”; // Enter your WiFi Password
WiFiClient client;
#define MQTT_SERV “io.adafruit.com”
#define MQTT_PORT 1883
#define MQTT_NAME “aschoudhary” // Your Adafruit IO Username
#define MQTT_PASS “1ac95cb8580b4271bbb6d9f75d0668f1” // Adafruit IO AIO
key
Const char server[] = “api.openweathermap.org”;
String nameOfCity = “Jaipur,IN”;
String apiKey = “e8b22b36da932dce8f31ec9be9cb68a3”;
String text;
Const char* icon=””;
Int jsonend = 0;
```

```
Boolean startJson = false;

Int status = WL_IDLE_STATUS;

#define JSON_BUFF_DIMENSION 2500

Unsigned long lastConnectionTime = 10 * 60 * 1000;     // last time you connected to the server, in milliseconds

Const unsigned long postInterval = 10 * 60 * 1000;  // posting interval of 10 minutes (10L * 1000L; 10 seconds delay for testing)

Const int ldrPin = D1;

Const int ledPin = D0;

Const int moisturePin = A0;  // moisteure sensor pin

Const int motorPin = D8;

Float moisturePercentage;              //moisture reading

Int temperature, humidity, soiltemp;

#define ONE_WIRE_BUS 4   //D2 pin of nodemcu

#define DHTTYPE DHT11   // DHT 11

#define dht_dpin D4

DHT dht(dht_dpin, DHTTYPE);

OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature sensors(&oneWire);

Const unsigned long Interval = 50000;

Unsigned long previousTime = 0;

//Set up the feed you're publishing to

Adafruit_MQTT_Client mqtt(&client, MQTT_SERV, MQTT_PORT, MQTT_NAME, MQTT_PASS);

Adafruit_MQTT_Publish Moisture = Adafruit_MQTT_Publish(&mqtt,MQTT_NAME "/f/Moisture");  // Moisture is the feed name where you will publish your data

Adafruit_MQTT_Publish Temperature = Adafruit_MQTT_Publish(&mqtt,MQTT_NAME "/f/Temperature");
```

```
Adafruit_MQTT_Publish Humidity =
Adafruit_MQTT_Publish(&mqtt,MQTT_NAME     "/f/Humidity");

Adafruit_MQTT_Publish SoilTemp =
Adafruit_MQTT_Publish(&mqtt,MQTT_NAME     "/f/SoilTemp");

Adafruit_MQTT_Publish WeatherData =
Adafruit_MQTT_Publish(&mqtt,MQTT_NAME "/f/WeatherData");

//Set up the feed you're subscribing to

 Adafruit_MQTT_Subscribe LED = Adafruit_MQTT_Subscribe(&mqtt,
MQTT_NAME "/f/LED");

 Adafruit_MQTT_Subscribe Pump = Adafruit_MQTT_Subscribe(&mqtt,
MQTT_NAME "/f/Pump");

Void setup()

{

  Serial.begin(9600);

  Delay(10);

  Dht.begin();

  Sensors.begin();

  Mqtt.subscribe(&LED);

  Mqtt.subscribe(&Pump);

  pinMode(motorPin, OUTPUT);

  pinMode(ledPin, OUTPUT);

  pinMode(ldrPin, INPUT);

  digitalWrite(motorPin, LOW); // keep motor off initally

  digitalWrite(ledPin, HIGH);

  text.reserve(JSON_BUFF_DIMENSION);

  Serial.println("Connecting to ");

  Serial.println(ssid);

  WiFi.begin(ssid, pass);
```

```
 While (WiFi.status() != WL_CONNECTED)
 {
   Delay(500);
   Serial.print(“.”);
// print … till not connected
 }
 Serial.println(“”);
 Serial.println(“WiFi connected”);
}
Void loop()
{
 Unsigned long currentTime = millis();
 MQTT_connect();
 If (millis() – lastConnectionTime > postInterval) {
   // note the time that the connection was made:
   lastConnectionTime = millis();
   makehttpRequest();
 }
//}
 Int ldrStatus = analogRead(ldrPin);
   If (ldrStatus <= 200) {
     digitalWrite(ledPin, HIGH);
     Serial.print(“Its DARK, Turn on the LED : “);
     Serial.println(ldrStatus);
   }
   Else {
    digitalWrite(ledPin, LOW);
```

```
    Serial.print("Its BRIGHT, Turn off the LED : ");
    Serial.println(ldrStatus);
   }
  moisturePercentage = ( 100.00 – ( (analogRead(moisturePin) / 1023.00) * 100.00 ));
  Serial.print("Soil Moisture is  = ");
  Serial.print(moisturePercentage);
  Serial.println("%");
If (moisturePercentage < 35) {
  digitalWrite(motorPin, HIGH);        // tun on motor
}
If (moisturePercentage > 38) {
  digitalWrite(motorPin, LOW);         // turn off mottor
}
 Temperature = dht.readTemperature();
 Humidity = dht.readHumidity();
 //Serial.print("Temperature: ");
 //Serial.print(temperature);
 //Serial.println();
 //Serial.print("Humidity: ");
 //Serial.print(humidity);
 //Serial.println();
 Sensors.requestTemperatures();
 Soiltemp = sensors.getTempCByIndex(0);
// Serial.println("Soil Temperature: ");
// Serial.println(soiltemp);
If (currentTime – previousTime >= Interval) {
```

If (! Moisture.publish(moisturePercentage)) //This condition is used to publish the Variable (moisturePercentage) on adafruit IO. Change thevariable according to yours.

```
{
 }
If (! Temperature.publish(temperature))
{
 }
If (! Humidity.publish(humidity))
{
//delay(30000);
 }
If (! SoilTemp.publish(soiltemp))
{
 }
If (! WeatherData.publish(icon))
{
}
 previousTime = currentTime;
}
Adafruit_MQTT_Subscribe * subscription;
```

While ((subscription = mqtt.readSubscription(5000))) //Don’t use this one until you are conrolling something or getting data from Adafruit IO.

```
{
If (subscription == &LED)
{
 //Print the new value to the serial monitor
 Serial.println((char*) LED.lastread);
```

```
        If (!strcmp((char*) LED.lastread, “OFF”))
       {
        digitalWrite(ledPin, LOW);
        }
        If (!strcmp((char*) LED.lastread, “ON”))
         {
        digitalWrite(ledPin, HIGH);
         }
      }
     If (subscription == &Pump)
      {
      //Print the new value to the serial monitor
      Serial.println((char*) Pump.lastread);
      If (!strcmp((char*) Pump.lastread, “OFF”))
       {
        digitalWrite(motorPin, HIGH);
       }
     If (!strcmp((char*) Pump.lastread, “ON”))
       {
        digitalWrite(motorPin, LOW);

      }
     }
    }
  Delay(9000);
 // client.publish(WeatherData, icon)
}
```

```
Void MQTT_connect() {
  Int8_t ret;
  // Stop if already connected.
  If (mqtt.connected())
{
    Return;
  }
  Uint8_t retries = 3;
  While ((ret = mqtt.connect()) != 0) // connect will return 0 for connected
  {
       Mqtt.disconnect();
       Delay(5000);  // wait 5 seconds
       Retries--;

      If (retries == 0)
       {
         // basically die and wait for WDT to reset me
         While (1);
       }
  }
}
Void makehttpRequest() {
  // close any connection before send a new request to allow client make connection
to server
  Client.stop();
  // if there's a successful connection:
  If (client.connect(server, 80)) {
```

```
  Client.println("GET /data/2.5/forecast?q=" + nameOfCity + "&APPID=" +
apiKey + "&mode=json&units=metric&cnt=2 HTTP/1.1");
  Client.println("Host: api.openweathermap.org");
 Client.println("User-Agent: ArduinoWiFi/1.1");
 Client.println("Connection: close");
 Client.println();
  Unsigned long timeout = millis();
  While (client.available() == 0) {
   If (millis() – timeout > 5000) {
    Serial.println(">>> Client Timeout !");
    Client.stop();
    Return;
   }
  }
  Char c = 0;
  While (client.available()) {
   C = client.read();
   // since json contains equal number of open and close curly brackets, this
means we can determine when a json is completely received  by counting
   // the open and close occurences,
   //Serial.print©;
   If (c == '{') {
    startJson = true;        // set startJson true to indicate json message has started
    jsonend++;
   }
   If (c == '}') {
    Jsonend--;
```

```
      }
      If (startJson == true) {
        Text += c;
      }
   // if jsonend = 0 then we have have received equal number of curly braces
      If (jsonend == 0 && startJson == true) {
        parseJson(text.c_str());  // parse c string text in parseJson function
        text = “”;                // clear text string for the next time
        startJson = false;        // set startJson to false to indicate that a new message
has not yet started
      }
    }
  }
  Else {
    // if no connction was made:
    Serial.println(“connection failed”);
    Return;
  }
}
//to parse json data received from OWM
Void parseJson(const char * jsonString) {
  //StaticJsonBuffer<4000> jsonBuffer;
  Const size_t bufferSize = 2*JSON_ARRAY_SIZE(1) + JSON_ARRAY_SIZE(2)
+ 4*JSON_OBJECT_SIZE(1) + 3*JSON_OBJECT_SIZE(2) +
3*JSON_OBJECT_SIZE(4) + JSON_OBJECT_SIZE(5) +
2*JSON_OBJECT_SIZE(7) + 2*JSON_OBJECT_SIZE(8) + 720;
  DynamicJsonBuffer jsonBuffer(bufferSize);
//  DynamicJsonDocument(bufferSize);
```

```
  // FIND FIELDS IN JSON TREE
  JsonObject& root = jsonBuffer.parseObject(jsonString);
  If (!root.success()) {
    Serial.println(“parseObject() failed”);
    Return;
  }
JsonArray& list = root[“list”];
  JsonObject& nowT = list[0];
  JsonObject& later = list[1];
  JsonObject& tommorow = list[2];
//  String conditions = list.weather.main;
  // including temperature and humidity for those who may wish to hack it in
  String city = root[“city”][“name”];
  String weatherNow = nowT[“weather”][0][“description”];
  String weatherLater = later[“weather”][0][“description”];
  String list12 = later[“weather”][0][“list”];
  Serial.println(list12);
  Serial.println(weatherLater);
  If(weatherLater == “few clouds”){
  Icon = “Few Clouds”;
  Serial.print(icon);
  }
  Else if(weatherLater == “rain”){
    Icon = “Rain”;
    Serial.print(icon);
  }
  Else if(weatherLater == “broken clouds”){
```

```
  Icon = “Broken Clouds”;
  Serial.print(icon);
 }
 Else {
  Icon = “Sunny”;
  }
}
```

7.1FEATURE 1

- IOT device
- IBM Watson platform
- Node red
- Cloudant DB
- Web UI
- MIT App
- Python code

8. TESTING

8.1WHITE BOX TESTING

White Box Testing (also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing) is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system. This method is named so because the software program, in the eyes of the tester, is like a white/ transparent box; inside which one clearly sees.

8.2USER ACCEPTANCE TESTING

User acceptance testing (UAT), also called application testing and end user testing, is a phase of software development in which the software is tested in the real world by its intended audience. UAT is often the last phase of the software testing process and is performed before the tested software is released to its intended market. The goal of UAT is to ensure software can handle real-world tasks and perform up to development specifications.

9.RESULTS

9.1PERFORMANCE METRICS

Users, memory utilization, workload efficiency, and command response times. Performance testing is a non-functional software testing technique that determines how the stability, speed, scalability, and responsiveness of an application holds up under a given workload. It’s a key step in ensuring software quality, but unfortunately, is often seen as an after thought, in isolation, and to begin once functional testing is completed, and in most cases, after the code is ready to release.

The goals of performance testing include evaluating application output, processing speed, data transfer velocity, network bandwidth usage, maximum concurrent.

10.1ADVANTAGES & DISADVANTAGES

- Allows farmers to maximize yields using minimum resources such as water, fertilizers, seeds etc.
- Solar powered and mobile operated pumps save cost of electricity.
- Smart agriculture use drones and robots which helps in many ways. These improves data collection process and helps in wireless monitoring and control.
- It is cost effective method.
- It delivers high quality crop production.

10.2DISADVANTAGES

- The smart agriculture needs availability of internet continuously. Rural part of most of the developing countries do not fulfil this requirement. Moreover internet connection is slower.
- The smart farming based equipments require farmers to understand and learn the use of technology. This is major challenge in adopting smart agriculture farming at large scale across the countries.

11.CONCLUSION

This intensive programme has been implemented through the full collaboration of the Host Committee composed of the IoT Forum, the International Telecommunication Union (ITU), the University of Applied Science Western Switzerland (HESSO) and Mandat International.

Week is a nomadic yearly conference and following the previous editions hosted in Barcelona, Berlin, Venice, Bald, Helsinki, London, Lisbon, Belgrade and Geneva, we were pleased to announce at the Closing Plenary Session that the next IoT Week Conference will take place next year for the first time in Bilbao (Spain) from 4th to 7th June 2018.

12.FUTURE SCOPE

Pump Control – remote and automated turn-off control of most electric and dieselirrigations well pumps used on farms today.

Pump Monitoring – the essential information a farmer needs to know about theoperation condition of his well pumps.

• Water flow rate, flow start and stop times/flow rate variance alert.

• Seasonal total volume of ground water extraction.

• Diesel pump monitoring parameters.

• Fuel tank level/fuel consumption/fuel tamper alert.

• Engine tachometer/speed variance alert.

• Battery voltage/low voltage alert.

• Murphy panel system faults.

Pump Automation Features – easy-to-use, easy-to-understand automationfeatures that have a tremendous positive impact on field operations.

13.APPENDIX:

13.1 OUTPUT LINK: : https://node-red-cwwhn-2022-11-10.eu-gb.mybluemix.net/ui/#!/0?socketid=GvuucIcZPmO6lef_AABc

13.2 GITHUB LINK : https://github.com/IBM-EPBL/IBM-Project-28723-1660115484

13.3PROJECT DEMO LINK:

https://youtu.be/AutomS83vug