SMART AGRICULTURE

INTERNSHIP PROJECT

BY:

ABDUR RAHMAN

B.TECH (2nd YEAR)

DTU

OVERVIEW

Imagine a world where there is not enough food for you and your loved ones, an arid world where there is nothing to satisfy your hunger. Because of the current methods of agriculture, where excess fertilizers and pesticides have been used. These methods are not sustainable and we have to find new methods to optimize crops. We cannot create new land to grow crops, and the water reserves are being reduced each year. For that, Industry 3.0, with IoT, Big Data and predictive analysis solutions have to emerge to avoid this scenario from happening. If only there was a way to monitor the field through a network of sustainable sensors, which could be monitored in real time 365 days a year, performing data analytics in the cloud and taking weather predictions to generate models. Taking all this information to manage resources such as water.

It is essential to increase the productivity of agricultural and farming processes to improve yields and cost-effectiveness with this new technology (IoT). In particular, IoT can make agricultural and farming industry processes more efficient by reducing human intervention through automation. In this project, the aim to use IoT applications in the agriculture and farming industries and develop an efficient system.

The aim of the project is to monitor crucial aspects of the agricultural field such as soil moisture temperature coordinating these with actual weather data and predicting the steps to be taken wirelessly to ensure better crop yield and minimum wastage of resources most importantly WATER!

LITERATURE SURVEY:

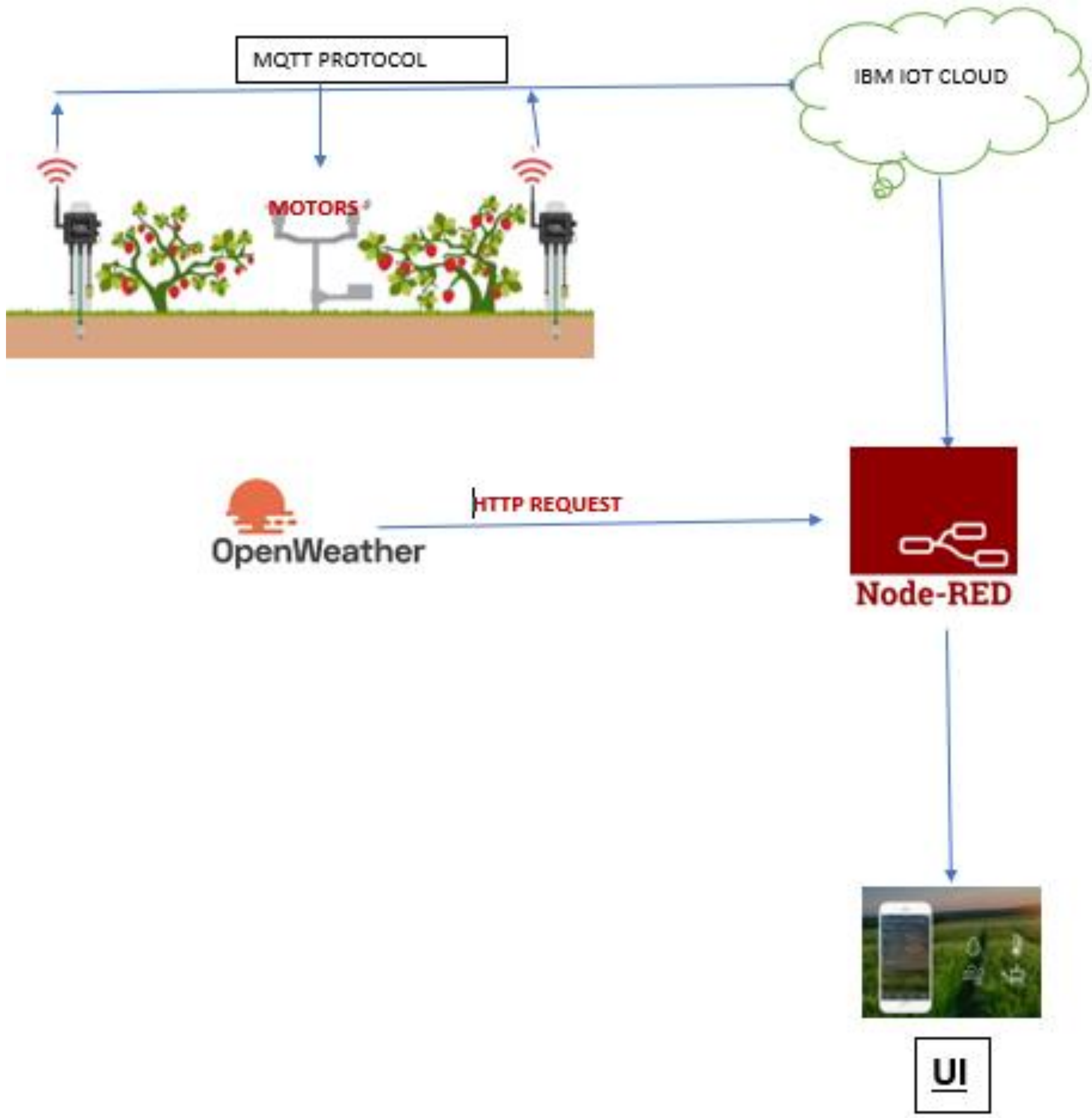
A recent Beecham's report entitled Towards Smart Farming: Agriculture Embracing the IoT Vision predicts that food production must increase by 70 percent in the year 2050 in order to meet our estimated world population of 9.6 billion people. It also describes growing concerns about farming in the future: climate change, limited arable land, and costs/availability of fossil fuels.

The existing method and one of the oldest ways in agriculture is the manual method of checking the parameters. In this method the farmers they themselves verify all the parameters and calculate the readings. The proposed solution focuses on the use of wireless sensor network which collects the data from different types of sensors and then send it to main server using wireless protocol. The collected data provides the information about different environmental factors which in turns helps to monitor the system. Also, on developing devices and tools to manage, display and alert the users using the advantages of a wireless sensor network system. It aims at making agriculture smart using automation and IoT technologies. The cloud computing devices that can create a whole computing system from sensors to tools that observe data from agricultural field images and from human actors on the ground and accurately feed the data into the repositories .This idea proposes a novel methodology for smart farming by linking a smart sensing system and smart irrigator system through wireless communication technology. It proposes a low cost and efficient wireless sensor network technique to acquire the soil moisture and temperature from various location of farm and as per the need of crop controller to take the decision whether the irrigation is enabled or not. It proposes an idea about how automated irrigation system was developed to optimize water use for agricultural practices.

The other major problem in field of agriculture is that climate changes and rainfall has been erratic over the past decade. Due to this in recent era, climate-smart methods have to be adopted by farmers. This problem is solved using an Open Weather API which is used to collect real time data of agriculture production environment that provides easy access for agricultural facilities like advices based on weather patterns.

THEORETICAL ANALYSIS:

The block diagram:



SOLUTION

We use a hardware device that communicates with an array of sensors gathering data from the soil as well as communicates with the irrigation system (the motors). The device then sends the data received from the sensors through node red gateway to the user interface. The user interface also gets the data from open weather api regarding current and forecasted weather conditions. All this information is given in the user interface along with the control buttons for the actuators which are motors for this project. The commands from the user interface are sent back through node red to the main controlling device and the motor is turned ON and OFF accordingly.

Node-red contains nodes that communicate with sensors for getting data through connecting to IBM IOT. Nodes for controlling motor at user’s command and also nodes to generate http request to get data from the open weather api

The functionality of machine learning can also be implemented in addition to the project where the device sends data to cloud. This data is stored in cloud and various algorithms are run to find out the right time to start irrigation in the field and the duration. The data from cloud can also be represented on a time scale for better visualization and understanding.

For the sake of this project the device along with the sensors and actuators have been simulated using IBM IOT services. The project makes use of two separate devices for controlling the sensors and the motors for simulation purposes but for actual hardware implementation a singe device can be used for both these purposes

Flowchart


Project Kick off
Getting started
Planning the scope objectives and deliverables of the project and setting up he development environment.
IBM Watson
Getting started with IBM cloud computing platform.
creating device in IBM IOT and and connecting it with simulator.
NODE-RED
Creating nodes to get simultaor data, control motors and generate HTTP request.
Node-RED
OPEN WEATHER API
Connecting the api with the user interface to get real time weather data and forcast.
Creating UI
Create a user interface using node-red and giving control bottons for the actuators(motors)
UI
Launch
Finally configuring th ebuttons to control motors from UI and launching the applocation.

STEP 1: Getting started

Github

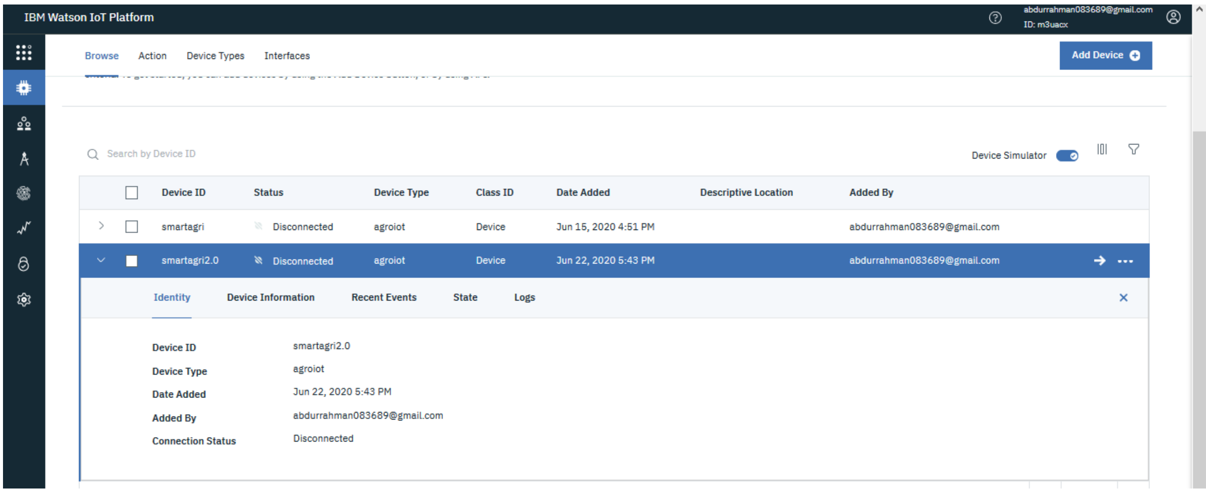
GitHub is a highly used software which is typically used for version control. It is helpful when more than just one person is working on a project. Say for example, a software developer team wants to build a website and everyone has to update their codes simultaneously while working on the project. In this case, GitHub helps them to build a centralized repository where everyone can upload, edit and manage the code files.

Made use of GitHub repositories in the project to track project development.

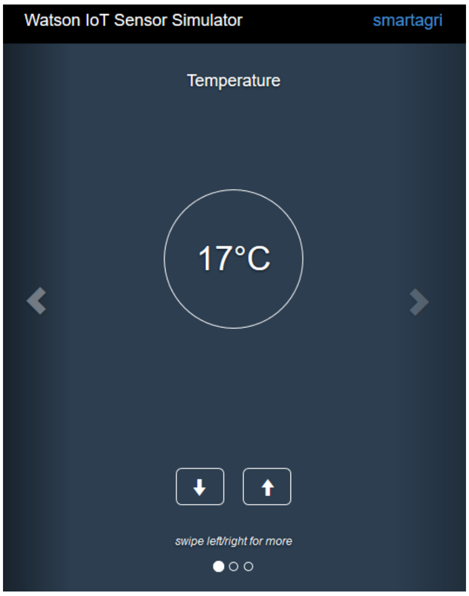
STEP 2: IBM Watson

The IBM Watson IoT Platform is a hub for connecting devices, gateways, and applications for IoT solutions. It supports REST and MQTT protocols for applications, devices, gateways, event processing, and administrative tasks. The IBM Watson IoT Platform is available on the IBM Cloud platform (formerly IBM Bluemix), a cloud platform based on Cloud Foundry and Kubernetes.

Made use of Watson IOT platform to create two devices. one to control sensors and get data from them and other to control the water pumping motors.



Connected the first device to IBM sensor simulator to get simulation data.



STEP 3: 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 JS Foundation.

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.

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.

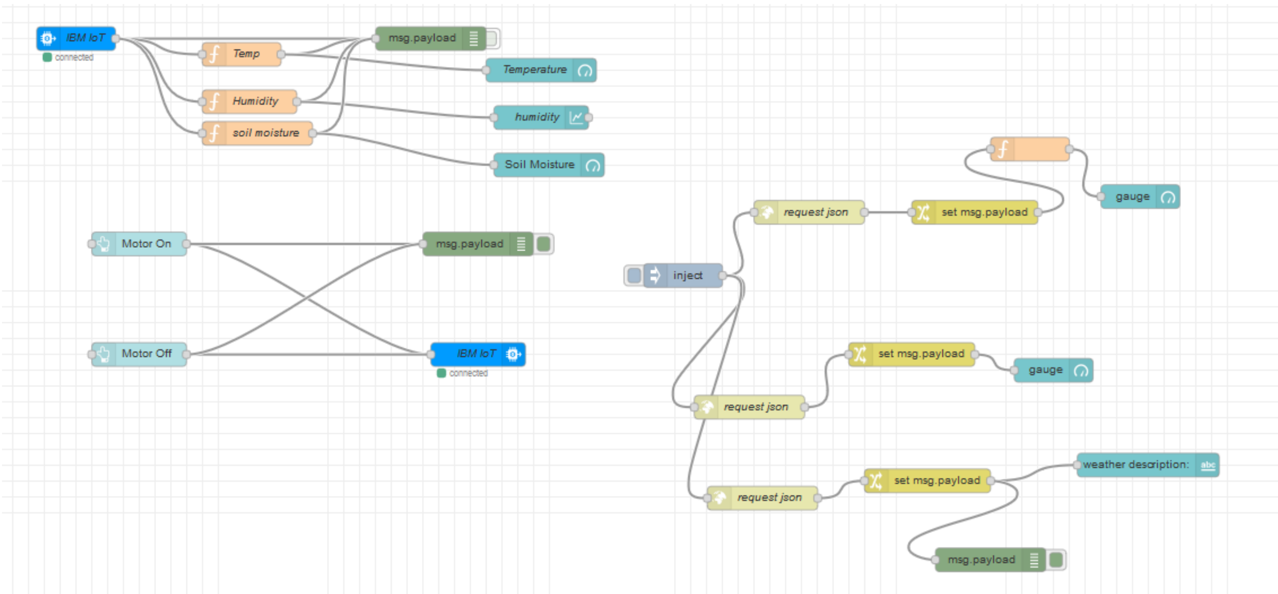
The palette of nodes can be easily extended by installing new nodes created by the community and the flows you create can be easily shared as JSON files.

The following flow of nodes was created to

1 get simulator data from IBM IOT sensor simulator and present it in UI.

2. connect control buttons to IBM IOT platform.

3. generate http request to get weather data using open weather api.



Communication protocols

A communication protocol is a system of rules that allow two or more entities of a communications system to transmit information via any kind of variation of a physical quantity. The protocol defines the rules, syntax, semantics and synchronization of communication and possible error recovery methods. Protocols may be implemented by hardware, software, or a combination of both.

Communicating systems use well-defined formats for exchanging various messages. Each message has an exact meaning intended to elicit a response from a range of possible responses pre-determined for that particular situation

MQTT (MQ Telemetry Transport or Message Queuing Telemetry Transport) is an open OASIS and ISO standard (ISO/IEC 20922)[4] lightweight, publish-subscribe network protocol that transports messages between devices. The protocol usually runs over TCP/IP; however, any network protocol that provides ordered, lossless, bi-directional connections can support MQTT. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. The MQTT protocol defines two types of network entities: a message broker and a number of clients. An MQTT broker is a server that receives all messages from the clients and then routes the messages to the appropriate destination clients. An MQTT client is any device (from a micro controller up to a full-fledged server) that runs an MQTT library and connects to an MQTT broker over a network.

Information is organized in a hierarchy of topics. When a publisher has a new item of data to distribute, it sends a control message with the data to the connected broker. The broker then distributes the information to any clients that have subscribed to that topic. The publisher does not need to have any data on the number or locations of subscribers, and subscribers, in turn, do not have to be configured with any data about the publishers.

Hypertext Transfer Protocol (HTTP) is an application-layer protocol for transmitting hypermedia documents, such as HTML. It was designed for communication between web browsers and web servers, but it can also be used for other purposes. HTTP follows a classical client-server model, with a client opening a connection to make a request, then waiting until it receives a response. HTTP is a stateless protocol, meaning that the server does not keep any data (state) between two requests. Though often based on a TCP/IP layer, it can be used on any reliable transport layer, that is, a protocol that doesn't lose messages silently like UDP does.

STEP 4: Open Weather API

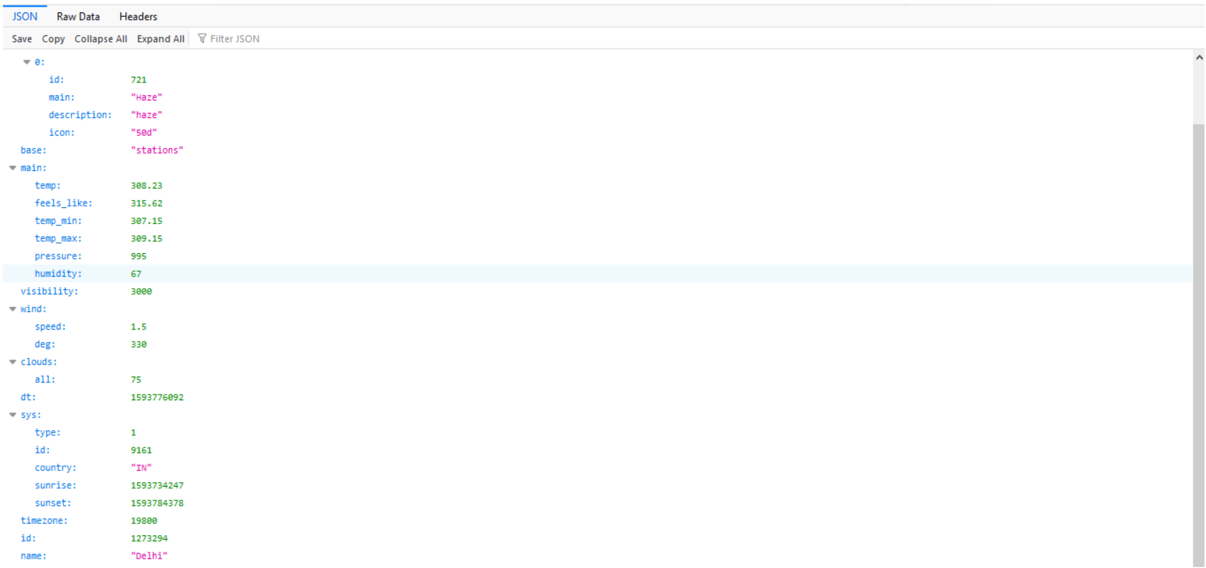
This api helps to Access current weather data for any location on Earth including over 200,000 cities. Current weather is frequently updated based on global models and data from more than 40,000 weather stations. Data is available in JSON, XML, or HTML format. You can call API by city name, by city ID, by geographic coordinates, by ZIP code.

5-days forecast is also available for any location or city. It includes weather data every 3 hours.

Weather maps including Precipitation, Clouds, Pressure, Temperature, Wind. You can connect any weather map to mobile and web apps.

Using simple syntax and a few API methods you can create triggers which will fire on an occurrence of the selected weather conditions (temperature, humidity, pressure, etc.) in a specified period of time.

JSON FILE FROM THE API:



The data from the api was requested by node red using http protocol.

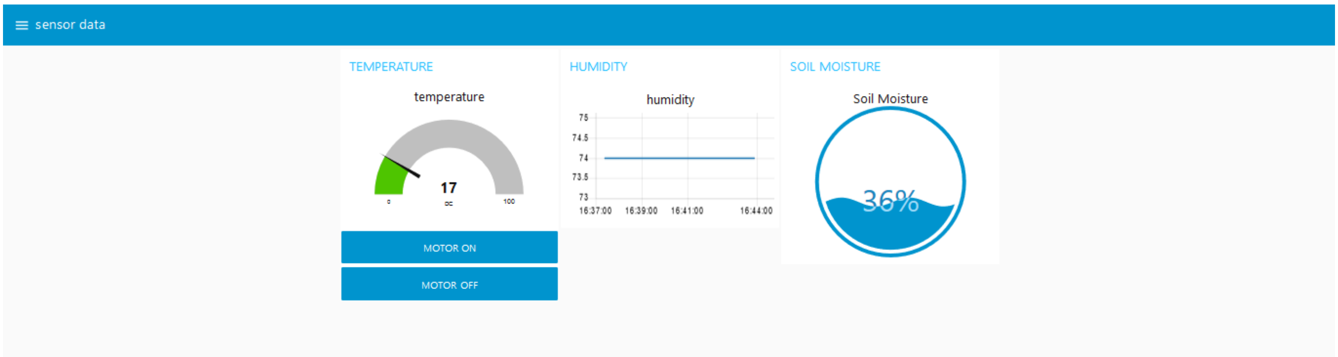
A python code was written to connect the control buttons to ibm device, meant to control the motor, and send requests for motor on or off .

RESULT

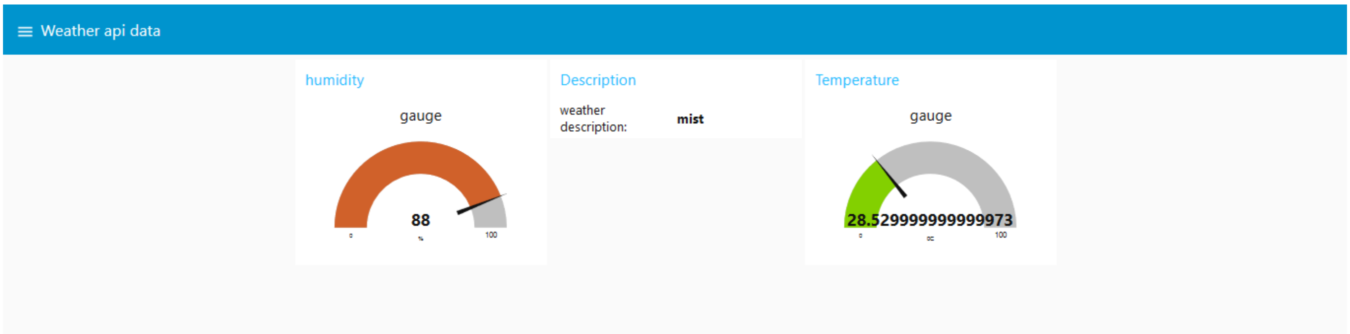
The Final UI

The UI created consisted of two tabs one for sensor data display and controls for motors and the other tab to depict data from the weather api.

Tab1(sensor data)



Tab2(weather data):



ADVANTAGES

Data collection

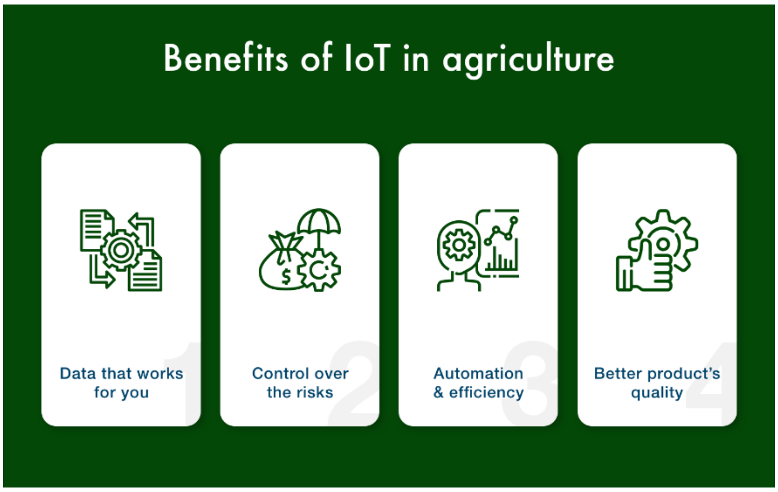
All data can be collected with the help of installed sensors. Such data like weather condition, health condition of crops. Data is stored in one place, and farmers can easily check it and analyse to make the right decision.

Reduction of risks

When farmers up-to-date information collected, they can understand what situation will be in the future, and they can predict some problems that may arise. Moreover, farmers may use data to improve their sales and change business processes.

Higher quality

Smart agriculture makes it possible to avoid challenges and remove all issues that may arise during farming processes. So the quality of the product is growing and consumers get a good product of high quality.



DISADVANTAGES

One huge disadvantage of smart farming is that it requires an unlimited or continuous internet connection to be successful. This means that in rural communities, especially in the developing countries where we have mass crop production, it is completely impossible to operate this farming method. In places where internet connections are frustratingly slow, smart farming will be an impossibility.

As pointed out earlier, smart farming makes use of high techs that require technical skill and precision to make it a success. It requires an understanding of robotics and ICT. However, many farmers do not have these skills. Even finding someone with this technical ability is difficult or even expensive to come by, at most. And, this can be a discouraging factor hindering a lot of promising farmers from adopting it.

While the use of smart technology in agriculture is impressive, it does incur a lot of costs. 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.

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.

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 change.

CONCLUSION AND FUTRE SCOPE

The project successfully implements the deliverables by making use of IBM WATSON IOT services to simulate implementations of devices, sensors and actuators.

IT also manages to successfully get data from Open weather api And depict everything in a clean and effective UI this helps farmer to analyze the crop conditions remotely and also gives him control over the irrigation system wirelessly.

The future scope of this project is to implement a database that store data and actions taken by farmers and the overall yield of crops and implement a machine earning algorithm that analyses this stored data and takes action regarding irrigation autonomously with high degree of precision and ensure a better yield by eliminating human error Other sensors can be added too that help prevent crops from pests

Another direction in which smart farming is headed involves intensively controlled indoor growing methods. The OpenAG Initiative at MIT Media Lab uses "personal food computers" (small indoor farming environments that monitor/administrate specific growing environments) and an open source platform to collect and share data. The collected data is termed a "climate recipe" which can be downloaded to other personal food computers and used to reproduce climate variables such as carbon dioxide, air temperature, humidity, dissolved oxygen, potential hydrogen, electrical conductivity, and root-zone temperature. This allows users very precise control to document, share, or recreate a specific environment for growing and removes the element of poor weather conditions and human error. It could also potentially allow farmers to induce drought or other abnormal conditions producing desirable traits in specific crops that wouldn't typically occur in nature.

BIBLIOGRAPHY

- *https://theiotmagazine.com*
- *https://nodered.org/*
- *https://ieeexplore.ieee.org/*
- *openweathermap.org*
- *https://app.lucidchart.coM*
- *https://github.com/*