AgriTech:
Farm Water Management

Prepared For

Petrichor AgriTech
(Great Learning Capstone Project)

Prepared By

Chinmoy Maity
Karthik Bhat
Vijayalakshmi
Joel Tom
Kanika Saraswat

Index

Introduction:

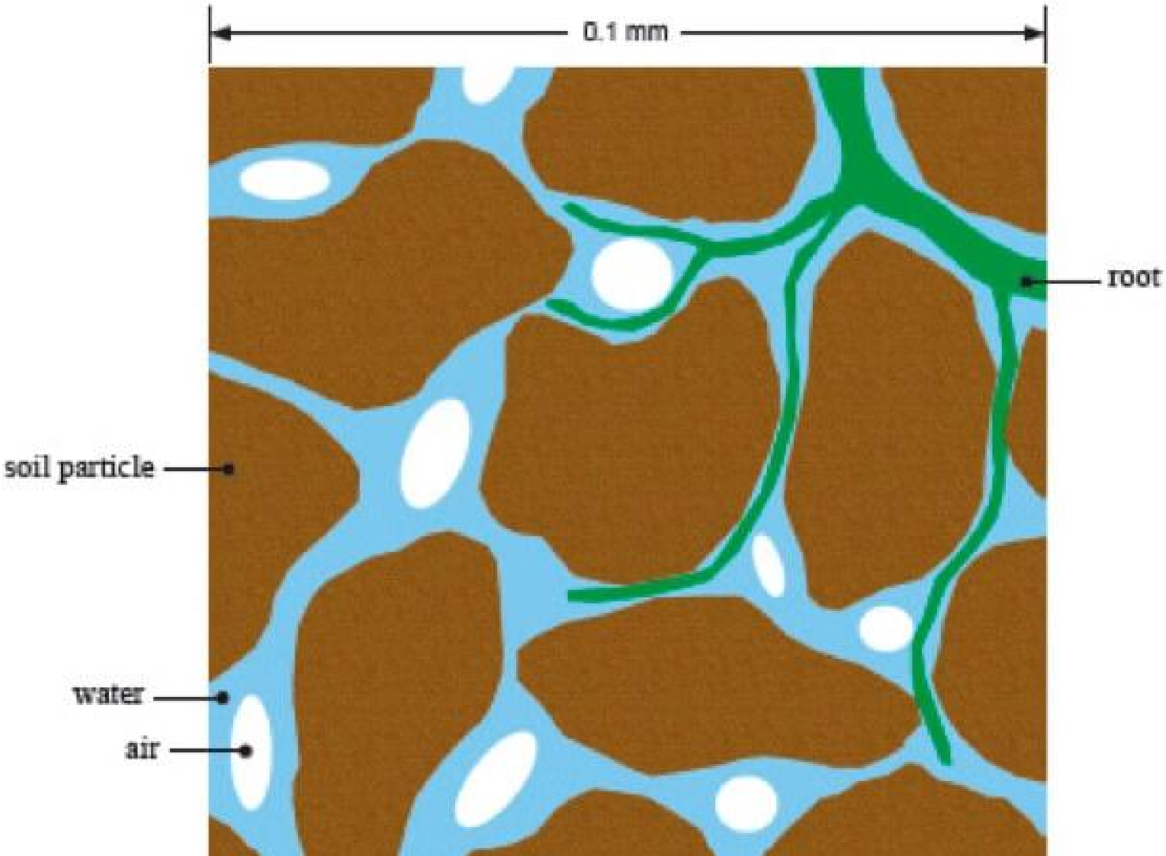
Maintaining the right balance between water consumption and soil moisture management is crucial to get a good crop yield at efficient costs. We'll develop an automatic sprinkler system based on soil and air parameters, with information coming from embedded sensors.

We are helping Petrichor AgriTech to develop an innovative solution for farm water Management.

Soil Moisture and Irrigation:

Soil moisture refers to the amount of water contained within a soil sample. The moisture content of a soil is an indicator of the degree of saturation of the specimen and is represented by the ratio of the mass of water to the mass of solids in the soil sample.

Soil moisture content plays an essential role in groundwater recharge and soil chemistry. Soil moisture values are particularly important for irrigation optimization and the health of a crop.



Unsaturated soil is composed of solid particles, organic material and pores. The pore space will contain air and water.

There are two different approaches for determining an irrigation schedule from soil moisture data:

1. Fill Point Irrigation Scheduling
2. Mass Balance Irrigation Scheduling

Fill Point Irrigation Scheduling

The fill point method is qualitative in that the irrigator looks at changes in soil moisture. With experience and knowledge of the crop, an irrigation schedule can be developed to fill the soil back up to a fill point. The fill point is an optimal soil moisture value that is related to the soil's field capacity. The fill point for a particular sensor is determined by looking at soil moisture data containing a number of irrigation events. This can be an effective and simple way to optimize irrigation. Because it is qualitative, accuracy of the soil moisture sensor is less important because the fill point is determined by looking at changes in soil moisture and not the actual soil moisture itself. This in some ways can be more efficient because lower cost soil moisture sensors can be used without calibration. While the fill point method can be easy to implement and is widely used for many crops, the mass balance method however can better optimize the irrigation, provide better control of salinity build up, and minimize the negative impacts of over-irrigation.

Mass Balance Irrigation Scheduling

The mass balance method (sometimes called scientific irrigation scheduling) is an irrigation schedule determined by calculating how much water is needed based on accurate soil moisture readings and the soil properties

As part of this project , we'll be implementing **Mass Balance Irrigation Scheduling** process to optimize the irrigation process.

Soil Moisture Measurement Considerations for Irrigation

Most soil moisture sensors provide measurements in the unit "water fraction by volume" (wfv or m3m-3) and are symbolized with the Greek letter theta (θ). Multiplying the water fraction by volume measurement by 100 will equal the volumetric percent of water in soil.

For example, a water content of 0.20 wfv means that a 1 cubic meter soil sample contains 200 cubic centimeters of water, or 20% by volume. Full saturation (all the soil pore spaces filled with water) occurs typically between 0.35-0.55 wfv for mineral soil and is quite soil-dependent.

SOIL WATER TERMINOLOGY

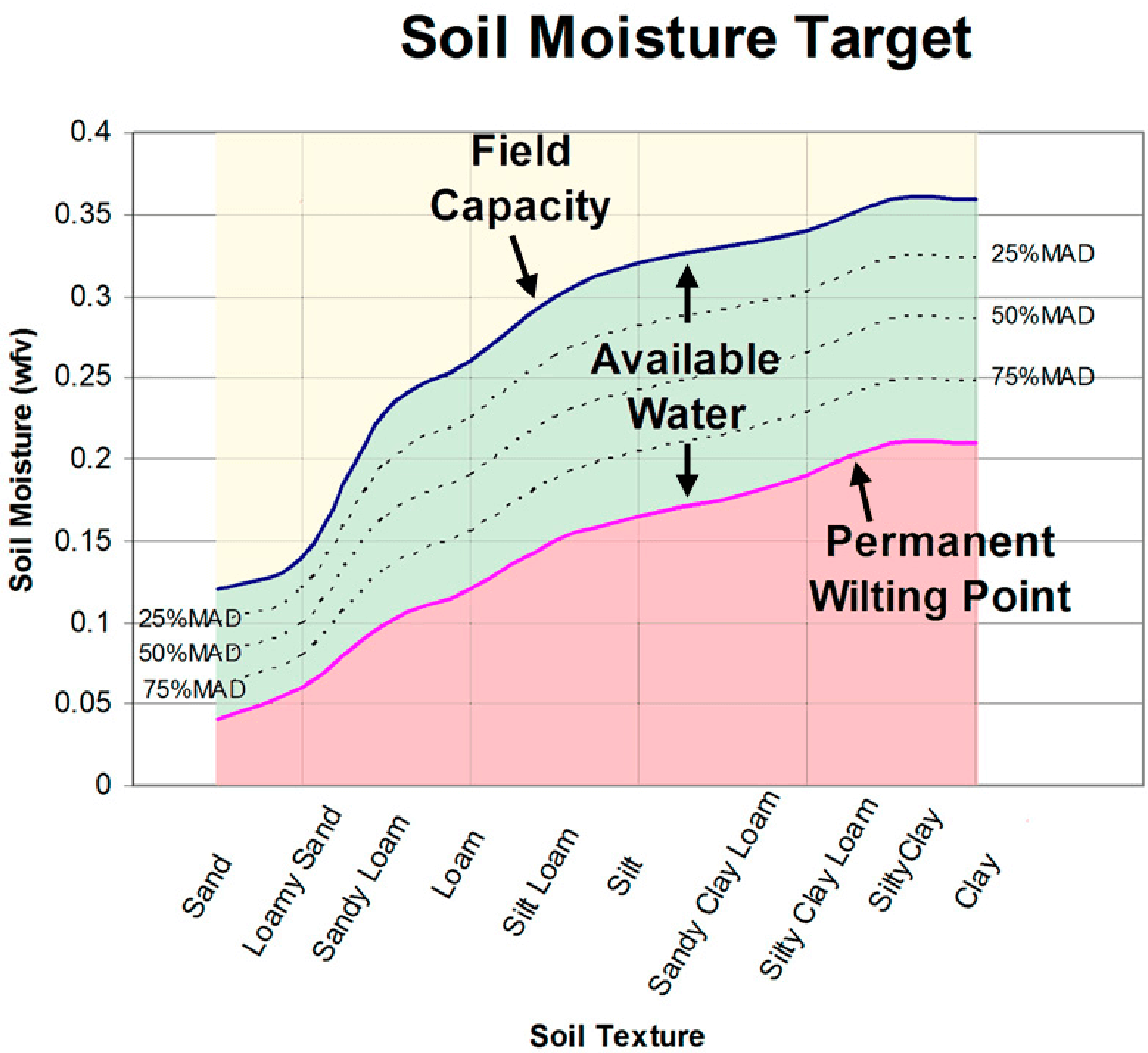
The following are terms commonly used in soil hydrology:

Available Water Capacity (θAWC) is the amount of water in the soil that is available to the plant.	**Soil saturation** (θSAT) refers to the situation where all the soil pores are filled with water. This occurs below the water table and in the unsaturated zone above the water table after a

The lower soil moisture limit is a very important value because dropping to or below this value will affect the health of the crops. The equations below show how to calculate the lower soil moisture limit and the soil moisture target for irrigation optimization. $$\theta AD = (\theta FC - \theta PWP) \times MAD$$ $$\theta AWC = \theta FC - \theta PWP$$ $$\theta LL = \theta FC - \theta AD$$	heavy rain or irrigation event. After the rain event, the soil moisture (above the water table) will decrease from saturation to field capacity. **Field Capacity** (θFC) refers to the amount of water left behind in soil after gravity drains saturated soil. Field capacity is an important hydrological parameter for soil because it can help determine the flow direction. Soil moisture values above field capacity will drain downward recharging the aquifer/water table. Also, if the soil moisture content is over field capacity, surface runoff and erosion can occur. If the soil moisture is below field capacity, the water will stay suspended in between the soil particles from capillary forces. The water will basically only move upward at this point from evaporation or evapotranspiration. **Permanent Wilting Point** (θPWP) refers to the amount of water in soil that is unavailable to the plant. **Allowable Depletion** (θAD) depletion represents the amount of soil moisture that can be removed by the crop from the soil before the crop begins to stress. **Lower soil moisture Limit** (θLL) is the soil moisture value below which the crop will become stressed because it will have insufficient water. When the lower limit is reached, it is time to irrigate. **Maximum Allowable Depletion** (MAD) is the fraction of the available water that is 100% available to the crop. MAD can depend on soil or crop type. the maximum level of depletion to which the soil can dry without causing water deficit stress in a crop that has a fully expanded root zone

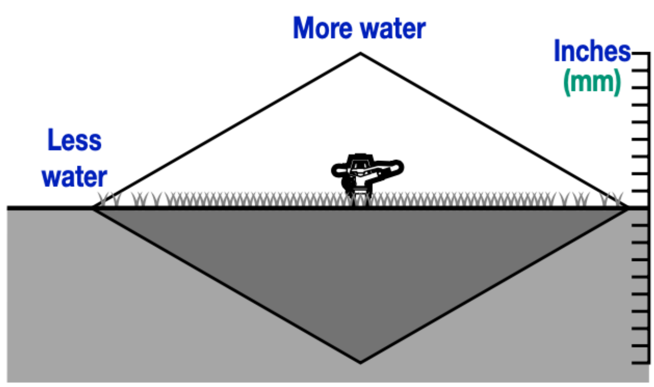
Crop	Maximum Allowable Depletion (MAD)	Effective Root Depth (Inches)
Apples	75%	36
Blueberries	50%	18
Carrot	50%	18
Cauliflower	40%	18
Cucumber	50%	24
Grass	50%	7
Green beans	50%	18
Leafy greens	40%	18
Peppermint	35%	24
Potatoes	35%	35
Strawberry	50%	12
Sweet corn	50%	24
Table beet	50%	18
Winter squash	60%	36

Typical maximum allowable depletion based on crop and effective root zone depth.



Soil textures and the available water

Sprinkler Selection :



Sprinkler water distribution graph

The area under the first 75% of the sprinkler's radius is generally sufficiently irrigated to grow vegetation without the need for an overlapping sprinkler. Beyond this 75% line, the amounts of water, diminishing with distance, become less and less effective and eventually will not support plants.

Thus, assume our sprinkler is 75% efficient

<u>Basic Calculation for the Irrigation time:</u>

As there is a lack of actual data , we have assumed the following parameter.

Crop Type : Cucumber
Soil Type : Silt
Maximum Allowable Depletion for Cucumber : 50% (0.5)
Effective Root Length : 24 Inches
Soil Moisture is 20% throughout the root zone
Area of land (sq. m) : 10,000
A **pump** with an average discharge of 15 liters/second
Sprinkler with 75% efficiency
The Permanent Wilting Point (**θPWP**): 16%
Field Capacity (**θFC**): 32%
Allowable Depletion (θAD) : 32%
Lower soil moisture Limit (θLL) : 24%

First we need to find out Depth of irrigation (cm), and post that we'll calculate how long we need to keep the pump on to reach the expected depth of irrigation

Example :

The soil is a silt, the MAD is 50%, and the soil moisture is 20% throughout the root zone which is down to 24 cm. The sprinkler is 75% efficient. ***How much water should be applied?***

Available Water Capacity (θAWC) = θFC – θPWP
θAWC = 32-16 = 16%

Therefore, the soil needs to be irrigated to increase the soil moisture by 16% down to 24 cm
16% X 24 cm = 3.8 cm of water applied

If the sprinkler is 75% efficient then 3.8 cm/0.75 = **5.12 cm of water should be applied once** the moisture level goes below 24%

How long do we need the pump to be switched ON ?

$$\text{Depth of irrigation (cm)} = \frac{\text{Volume of water (Cu. m)}}{\text{Area of land (sq. m)}} \times 100$$

5.12 = Volume of water / 10000 x 100

5.12*10000/100 = Volume of water = 512 Cu. m (M^3) = 512x1000 = 512000 Lt

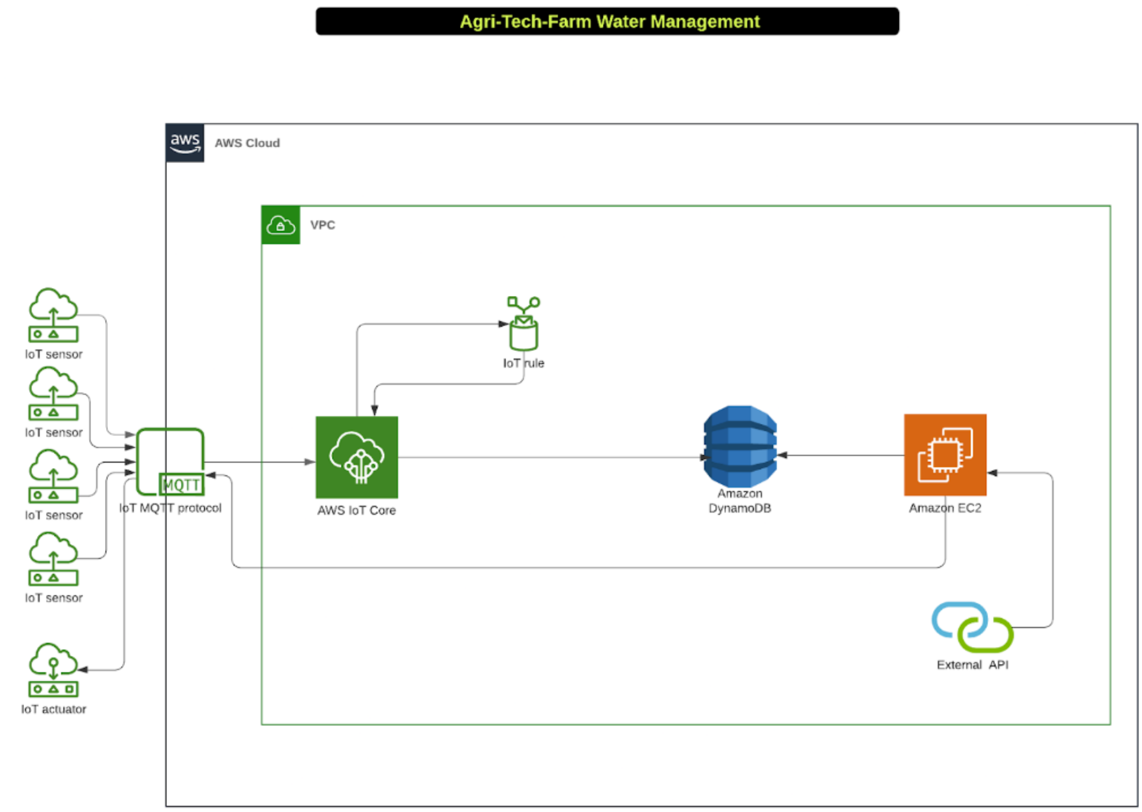
If 15 liters/second is discharge then = 34133 second required ~ **9 Hrs**

Project Delivery Planning :

	GL Action Plan	Our Action Plan
18-10-2021	**Project Start**	
19-10-2021		1. Architecture Diagram 2. Internally tryout the architecture implementation and see if this is feasible to complete the Basic Requirement for the project 3. Have a call with the team to discuss and finalize architecture to present it in mentor session
20-10-2021		
21-10-2021		
22-10-2021		
23-10-2021		1. Start working to finalize the Basic Requirements based on the final architecture
24-10-2021	**Mentor Session 1**	
25-10-2021		
26-10-2021		
27-10-2021		
28-10-2021		
29-10-2021		1. We can connect if all are available for discussion
30-10-2021	**Mentor session 2**	1. Finish the all required coding for the Basic Requirement on or before this date 2. We can have an internal demo for the Basic requirement and may take a few screerecord of working demo etc.
31-10-2021		1. Finalized Advance features implementation process
1-11-2021		
2-11-2021		
3-11-2021		
4-11-2021		
5-11-2021		
6-11-202		1. Review the Advance Feature implementation progress 2. Finish the Advance Feature implementation
7-11-2021		
8-11-2021	**Intermediate Milestone**	
9-11-2021		
10-11-2021		
11-11-2021		

12-11-2021		1.Internal Demo for the final version 2. Documentation & Ready for submission
13-11-2021		
14-11-2021		
15-11-2021		
16-11-2021		
17-11-2021		
18-11-2021		
19-11-2021	**Last Submission Date**	
20-11-2021	Demo Day	
21-11-2021	Demo Day	

Architecture Diagram:



IoT Device Mapping with Farm:

We have 20 Soil sensors , 5 Sprinklers.
In real implementation we don't need more than 1 Soil sensor in each farm (for about 2-3 acres of land). For the better understanding We'll considering the following

We have 4 farms , each farm has 4 soil sensors , and each farm has sprinklers that are connected to the irrigation pump.

So, We'll get an average value of the soil sensor data and based on that we'll plan the irrigation.

To be noted , there are many parameters that we need to consider in the process like evapotranspiration activity but we can't consider those as we don't have much real case in hand.

Code Overview & Implementation:

Our code repository : https://github.com/chinmoym2004/agri-project-iot/tree/advance

Getting Started with the application

This will show you how to run this application on your local system successfully.

Prerequisites

- AWS account

You should have an actual AWS account or AWS Student Account to perform the following action.

1. From the given .env.example create .env file in the same location
2. Open ~/.aws/credential file and update the details like this

   [default]
   aws_access_key_id=<UR ACCOUNT KEY>
   aws_secret_access_key=<ACCOUNT SECRET>
   aws_session_token=<ACCESS TOEK WITH ALL PERMISSION>
   region=<UR REGION i.e. us-east-1 or us-east-2>
3. In your AWS account you should have a rule under IoT/Act that ready data from iot/agritech MQTT topic and push to "soildata" table
4. Get the eng point from IoT Settings and set in .env for ENDPOINT
5. Install all python dependency from the requirements.txt file
6. Get API key from OpenWeatherAPI and set in .env for OPEN_WEATHER_API_KEY
7. You can modify the other values in .env as you wish.

Run the application

Below are the steps to run the application. Move to the application directory

1. First we'll create tables & the Things in IoT from the code . To do that , run

   python3 console/src/createTables.py

```
chinmoy@MacBook-Pro agri-project-iot % python3 console/src/createTables.py
Deleted "farms" table
Created "farms" table
Deleted "devices" table
Created "devices" table
Deleted "soildata" table
Created "soildata" table
Created "sprinklerlogs" table
```

2. Once the above action success then we'll start pushing the data from the Sprinkler and publish to MQTT. Run

   python3 console/src/ssPublishData.py

```
chinmoy@MacBook-Pro agri-project-iot % python3 console/src/ssPublishData.py
Connecting to SS007 ...
Connection Established

Connecting to SS0013 ...
Connection Established

Connecting to SS001 ...
Connection Established

Connecting to SS006 ...
Connection Established

Connecting to SS0016 ...
Connection Established

Connecting to SS0014 ...
Connection Established

Connecting to SS008 ...
Connection Established

Connecting to SS002 ...
Connection Established

Connecting to SS003 ...
Connection Established

Connecting to SS005 ...
Connection Established

Connecting to SS009 ...
Connection Established

Connecting to SS0012 ...
Connection Established

Connecting to SS0010 ...
Connection Established

Connecting to SS0015 ...
Connection Established

Connecting to SS0011 ...
Connection Established

Connecting to SS004 ...
Connection Established

Published: '{"device_id": "SS007", "timestamp": "2021-11-19 22:56:31", "datatype": "Soil Moisture", "value": 23.9}' to the topic: iot/agritech
Published: '{"device_id": "SS0013", "timestamp": "2021-11-19 22:56:31", "datatype": "Soil Moisture", "value": 33.8}' to the topic: iot/agritech
Published: '{"device_id": "SS001", "timestamp": "2021-11-19 22:56:31", "datatype": "Soil Moisture", "value": 24.7}' to the topic: iot/agritech
Published: '{"device_id": "SS006", "timestamp": "2021-11-19 22:56:32", "datatype": "Soil Moisture", "value": 20.6}' to the topic: iot/agritech
Published: '{"device_id": "SS0016", "timestamp": "2021-11-19 22:56:32", "datatype": "Soil Moisture", "value": 24.1}' to the topic: iot/agritech
Published: '{"device_id": "SS0014", "timestamp": "2021-11-19 22:56:33", "datatype": "Soil Moisture", "value": 31.6}' to the topic: iot/agritech
Published: '{"device_id": "SS008", "timestamp": "2021-11-19 22:56:33", "datatype": "Soil Moisture", "value": 34.8}' to the topic: iot/agritech
Published: '{"device_id": "SS002", "timestamp": "2021-11-19 22:56:33", "datatype": "Soil Moisture", "value": 30.9}' to the topic: iot/agritech
Published: '{"device_id": "SS003", "timestamp": "2021-11-19 22:56:34", "datatype": "Soil Moisture", "value": 30.7}' to the topic: iot/agritech
Published: '{"device_id": "SS005", "timestamp": "2021-11-19 22:56:34", "datatype": "Soil Moisture", "value": 30.5}' to the topic: iot/agritech
Published: '{"device_id": "SS009", "timestamp": "2021-11-19 22:56:35", "datatype": "Soil Moisture", "value": 27.5}' to the topic: iot/agritech
Published: '{"device_id": "SS0012", "timestamp": "2021-11-19 22:56:35", "datatype": "Soil Moisture", "value": 34.3}' to the topic: iot/agritech
Published: '{"device_id": "SS0010", "timestamp": "2021-11-19 22:56:35", "datatype": "Soil Moisture", "value": 24.5}' to the topic: iot/agritech
Published: '{"device_id": "SS0015", "timestamp": "2021-11-19 22:56:36", "datatype": "Soil Moisture", "value": 20.5}' to the topic: iot/agritech
Published: '{"device_id": "SS0011", "timestamp": "2021-11-19 22:56:36", "datatype": "Soil Moisture", "value": 21.7}' to the topic: iot/agritech
Published: '{"device_id": "SS004", "timestamp": "2021-11-19 22:56:36", "datatype": "Soil Moisture", "value": 27.4}' to the topic: iot/agritech
Published: '{"device_id": "SS007", "timestamp": "2021-11-19 22:56:37", "datatype": "Soil Moisture", "value": 27.5}' to the topic: iot/agritech
Published: '{"device_id": "SS0013", "timestamp": "2021-11-19 22:56:37", "datatype": "Soil Moisture", "value": 21.0}' to the topic: iot/agritech
```

3. Keep the above thing running for as long as you with (timeout set to 2 hrs so the program will exit after 2 hrs if you don't take any action). You should see data being pushed like this
4. Open a new tab and tun

   python3 console/src/SprinklerCallV2.py

```
[chinmoy@MacBook-Pro agri-project-iot % python3 console/src/SprinklerCallV2.py
Preparing Data for: SP002
Preparing Data for: SP001
Preparing Data for: SP003
Preparing Data for: SP004
Starting thread .. SP002
In sprinkler action, Device ID : SP002
Starting thread .. SP001
Data from :2021-11-19 23:11:58.946020
In sprinkler action, Device ID : SP001
Starting thread .. SP003
Data from :2021-11-19 23:11:58.946208
Data to :2021-11-19 23:12:28.946020
Data to :2021-11-19 23:12:28.946208
Starting thread .. SP004
In sprinkler action, Device ID : SP003
Data from :2021-11-19 23:11:58.952109
Data to :2021-11-19 23:12:28.952109
In sprinkler action, Device ID : SP004
Data from :2021-11-19 23:11:58.954908
Data to :2021-11-19 23:12:28.954908
Total SS count : 4
Avg soil moister : 26.333333333333332
Total SS count : 4
Avg soil moister : 25.7
Total SS count : 4
Avg soil moister : 31.25
Current Moister is :31.25 .. No Action Required
Excution Finished for SP003
Total SS count : 4
Avg soil moister : 28.0
Sprinkler:SP002 No action: Humidity is above expected range. Exp:70.0 Location Humidity :89
Excution Finished for SP002
Closing thread .. SP002
Sprinkler:SP004 No action: Humidity is above expected range. Exp:70.0 Location Humidity :87
Excution Finished for SP004
Sprinkler:SP001 No action: Humidity is above expected range. Exp:70.0 Location Humidity :92
Excution Finished for SP001
Closing thread .. SP001
Closing thread .. SP003
Closing thread .. SP004
Closing sprinkler actions
```

5. To Keep the step 4 automated , we can add this in a cronjon and execute in each 5 min. edit cron file using
   crontab -e
   And then enter this
   * 5 * * * cd /Users/chinmoy/6_IITM_GREAT_LEARNING/Final\ Project/Flask/agri-project-iot && python3 python3 console/src/SprinklerCallV2.py >> /dev/null 2>&1

   * 1 * * * cd /Users/chinmoy/6_IITM_GREAT_LEARNING/Final\ Project/Flask/agri-project-iot && python3 python3 console/src/SprinklerOff.py >> /dev/null 2>&1

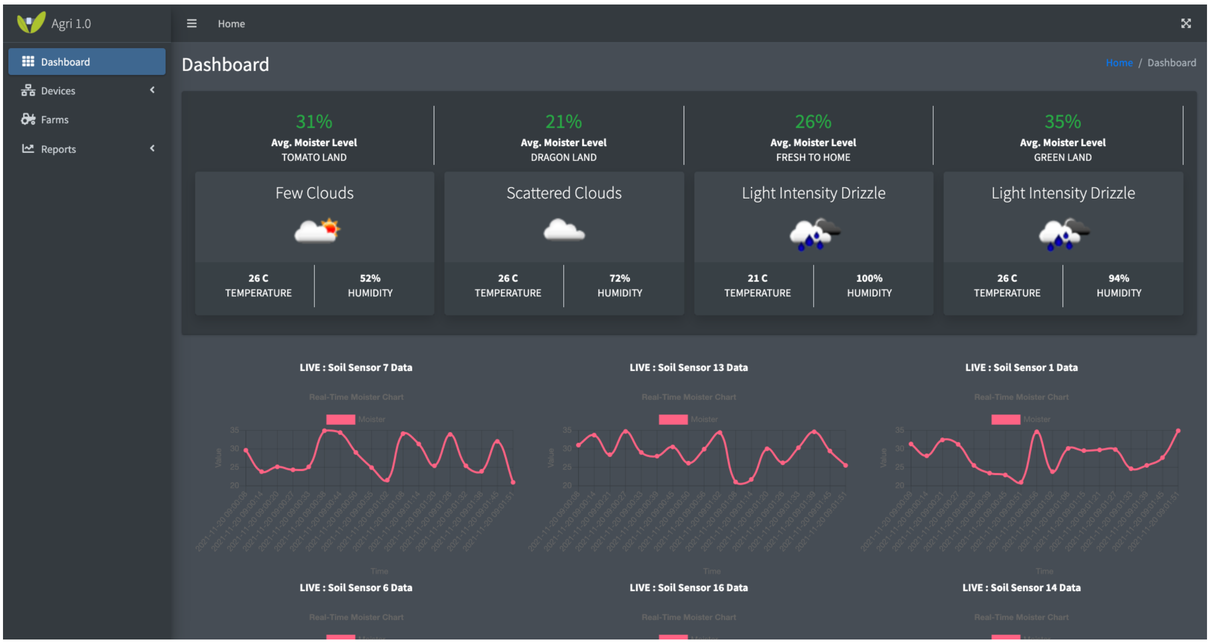
   You need to update the CD path based on your system.

6. To View the data in a dashboard , you can open a new tab and run the following

   python3 app.py

This will start the Python Flask application and you can then open http://localhost:81 in your browser to see the dashboard.

Sample Screenshots:



Conclusion :

This project is for academic purposes and there is a huge room for improvement to bring this to production level and market ready.
In this we have used multithreading , which will become a bottleneck in future once the number of farms increases , and then we need to bring the load balancer in place to manage the situation .
Currently we are storing per second data in dynamo db , which will not be a good solution once the number of farms increases. We can replace those with kinesis firehose and maybe have the average of per hr data and then store it in dynamoDB.

There is room for this kind of improvement to improve scalability as well as performance.
With this, we'll conclude the project work and thank you all for your time.