IoT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

NALAIYATHIRAN PROJECT BASED LEARNING

On

PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITYAND ENTERPRENEURSHIP

A PROJECT REPORT

Submitted by

ABHISHEK NIGAM

AAKASH V

REVANOORI BHARGAVI

SALIVENDRA MEGHANA ANJALI

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

SAVEETHA ENGINEERING COLLEGE, CHENNAI
ANNA UNIVERSITY: CHENNAI 600 025

NOV 2022

BONAFIDE CERTIFICATE

Certified that this project report titled “IoT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE by NALAIYA THIRAN PROJECT BASED LEARNING Program”, is the bonafide work of ABHISHEK NIGAM (212219060004), AAKASH V (212219060001), REVANOORI BHARGAVI (212219060217), S MEGHANA ANJALI (212219060160) who carried out the work under faculty mentor and industry mentor supervision, for the partial fulfilment of the requirements for the award of the degree of BACHELOR OF ENGINEERING IN ELECTRONICS AND COMMUNICATION ENGINEERING.

Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion.

ABSTRACT

The system will provide a complete technical solution to the destruction of crops by animals using internet of things to prevent crops from animals. Agriculture is the backbone of the economy but because of animal attacks, climate changes in agricultural lands there will be huge loss of crops. The feature of this paper includes the development of the system that can monitor Temperature, Humidity, Soil moisture and even the movement of animals which may destroy the crops in agricultural fields. The IoT based smart farming system being proposed via this report is integrated with Microcontroller mixed with different sensors and a WiFi module producing live data feed that can be obtained online. The moisture contents in the soil sensed by using the moisture sensor and it will identify the amount of water supplied required to the crop and sends data to ARM cortex and enables sensor to supply water which automatically turn on the water source and turn off it when need is satisfied. PIR sensor used to detect whether a human has moved in or out of the sensors range. After processing the available information, if the human is not found the system raised the buzzer sound, to alert people about intrusion. The system to monitor agricultural land is developed by using WSN. IoT monitored data is sent to cloud so that farmers can get the data easily. IoT enabled agriculture system is greatly beneficial to the farmers as it reduces the manpower and harmful chemical for increasing the amount of the crops. Using IoT technology it helps the farmers to control their fields anywhere is simple and now it is cost effective. If any problem arises, the announcement sends to mobile of farmers. The farmers can rectify the problem by through mobile. IoT based smart farming is used to monitor the field in proper time by any time and being anywhere.

1. Introduction

1.1 Project Overview

Although there have been several initiatives and studies on the amount of irrigation in agriculture as a result of technological developments, our strategy is to focus on this development. The "Internet of Things" describes the connections between items that have electronic parts, software, and sensors that allow for data transfer. Because it allows for direct interaction between the real world and the digital one, along with its efficiency and accuracy, it reduces human creativity. This project uses the principles of the Internet of Things to describe an advanced irrigation system for the cultivation of crops. IOT application for environmental monitoring uses wireless sensors to monitor various environmental factors in order to protect the environment. There are many crops that need water in significantly larger volumes, thus it must be carefully maintained because failure to properly monitor the crops could lead to potential issues. Providing secure agricultural cultivation from wild animals and other elements is one of them. Manual irrigation is the standard irrigation technique for agricultural land, which on larger farms may require specialist labor. Manual labor or conventional methods have some disadvantages, such as poor farm maintenance and insufficient crop watering. Resources must be handled effectively and promptly to avoid major loss and a reduction in the quantity and quality of agricultural output. Continuous farm monitoring will necessitate additional personnel and professionals on large farms. Farmers in outlying locations might have to travel a great distance to get professional help and acquire a solution. Fig. 1 shows the block diagram of proposed system here.

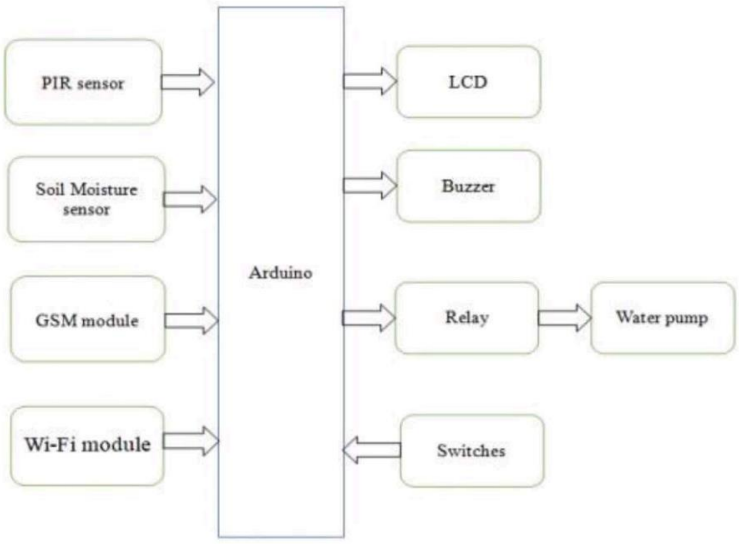


Fig. 1 Block diagram of proposed system

This sophisticated system has a key feature that allows us to create the strategy for our crop specimens: values are compared with the cut off range depending on the dataset. A confirmation message is sent to the farmer through SMS and email alert if the range exceeds the cut-off zone utilising the collected parameters, based on the precise course of action that needs to be done. The server oversees and manages the entire system. This server can be accessed with the aid of the Android software AgriApp, which was created to assist farmers in having information about their farms at their fingertips. Before deploying the system in the farm, UserId and password must be registered with farmer beforehand, and the link must be established. This system explains how the Internet of Things works in real time over the internet. Overall, implementation is simple, and since setup is one-time only, maintenance costs are also low. Based on our small-scale studies, this method can be used in the field and a significant amount of resources can be saved at an early stage, preventing resource waste.

1.2 Purpose

Many farmers are having a difficult time securing their farms at night. Farmers find it challenging to maintain field security around-the-clock. Due to increased agricultural output, the fields must be regularly watered. The work of farmers needs to be reduced because they are crucial to the production of food. Precision agriculture reduces environmental impact while increasing productivity, yields, and profitability. The Internet of Things is now playing a significant part in the transformation of traditional technology from homes to offices. One of the main drivers of this transformation is the ongoing research into IoT and the daily launch of new products aimed at improving productivity in "Agriculture". IoT-based agriculture convergence technology adds significant value in terms of improved output and quality, as well as significant load reduction for farmers. The future of agriculture will also include "Precision Agriculture," which is predicted to expand at a rate of $3.7 billion by 2020.

Farmers would be able to increase crop yields and make effective use of water thanks to data generated by GPS and Smart Sensors on agricultural fields, integration of smart farming equipment, and Big Data analytics. As a result, waste of any kind would be dramatically decreased. The need for IoT-based Smart Farming is urgent given the current state of agriculture, which is beset by numerous issues. In order to implement smart farming in real world.

2. Literature Survey

2.1 Existing Problem

The demands of modern agriculture, which demands high-yield, high-quality, and efficient output, are incommensurable with those of traditional agriculture and its linked sector. In order to estimate the best possible productivity and crop suitable for the accurate land, it is crucial to modernize existing methods and use information technology and data collected over a period. Access to high-speed internet, mobile devices, and dependable, low-cost satellites (for images and positioning) are only a few of the major technologies defining the precision agriculture trend. One of the most well-known IoT applications in the agricultural industry is precision agriculture, and many companies are using this method globally.

Manufacturing, healthcare, and the automotive industries have all seen significant IoT adoption. When it comes to food production, transportation, and storage, it provides a wide range of solutions that can increase India's per capita food availability. sensors that provide data on the condition of the soil's nutrients, pest infestation, moisture levels, and other factors that can be utilized to gradually increase crop yields.

Here are few problems for the agricultural and associated industries where the Internet of Things will be useful:

- *Tea Industry*

1. Tea production is rejected if pesticides and fertilizers are used in excess of what is necessary.
2. Plucking coarse leaves will lower the quality of the tea that is produced.
1. Tea plants are frequently pruned to maintain their perfect shape. However, excessive pruning causes the plant to die.

- *Livestock*

  IoT applications can help large farm owners gather information about the whereabouts, comfort, and health of their cattle. They can use this information to identify unwell animals and separate them from the herd to stop the spread of disease. As ranchers can locate their cattle with the use of IoT-based sensors, it also reduces labor costs.

- *Eco-Harvester for fruits*

  In order to minimise harvesting loss, the Eco-harvester injects an artificial PME enzyme activator that enables only the mature fruits to be detached from the branches of the tree, leaving the immature ones behind and giving them time to mature. To monitor when and which area of the field is prepared for the procedure, IoT applications may be employed.

- *Smart Greenhouses*

A technique that aids in increasing the yield of fruits, vegetables, and other crops is greenhouse farming. In greenhouses, the environmental variables are managed manually or using a proportional control system. IoT applications can be very helpful to farmers that employ greenhouse technology and can simplify and ease their labour.

- *Cold Chain*

Agriculture produce waste can be considerably decreased by using improved storage monitoring systems and cold chain monitoring tools. IoT applications can be used to enhance storage and delivery procedures while extending the produce's shelf life. There is not much need for human involvement in doing this.

- *Fish-farming and breeding*

The consumption of fish in Assam is lofty. But the fish-farming and breeding techniques haven't improved much to meet the growing demand of fish in the region. To increase the productivity and reduce input cost, IoT application to monitor dissolved oxygen, pH indicator, Ammonium nitrate indicator as well as automatic fish feed system can help reduce manpower and improve quality & quantity.

We need to find a proper solution for our problem and work in team to solve the tough situations faced by our farmer brothers.

2.2 Reference Papers

In [1], the author clearly explains about how one of the primary issues facing farmers in our nation is low crop output. There are two basic causes for this. crops damaged because of severe weather and wild animals. This essay offers a remedy for agricultural destruction caused by animals. This system will give farmers a full technological answer using the Internet of Things (IOT) to protect their crops from wild animals and give them information to increase their output. PIR sensors and cameras are used to detect animals, and TensorFlow image processing techniques are used to identify the detected animals. The system's processing component is a Raspberry PI, and sound buzzers are utilized to transmit the ultrasound frequencies. In [2], we understand that in, day to day lives are heavily reliant on technology. The need for Internet of Things (IoT) has increased significantly across many industries, which has attracted significant research interest from both the academic community and the business community. IoT adoption has facilitated smart farming and precision agriculture, to name a few, solely in the agriculture industry. To stop animals from entering the crop field, this paper describes the development of an Internet of Things application for crop protection. To guard against potential harm to agriculture from wild animal attacks and meteorological conditions, a repelling and monitoring system is offered.

References:

1. N. S. Gogul Dev, K. S. Sreenesh and P. K. Binu, "IoT Based Automated Crop Protection System," 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 2019, pp. 1333-1337, doi: 10.1109/ICICICT46008.2019.8993406.

2. S. Giordano, I. Seitanidis, M. Ojo, D. Adami and F. Vignoli, "IoT solutions for crop protection against wild animal attacks," 2018 IEEE International Conference on Environmental Engineering (EE), 2018, pp. 1-5, doi: 10.1109/EE1.2018.8385275.

2.3 Problem statement definition

Food security has become a key worry for all governments globally as the global population grows, natural resources, cropland, and unpredictable environmental conditions deteriorate. Due to these issues, the agricultural sector is moving toward "smart agriculture," which aims to increase operational effectiveness and production using big data and Internet of Things (IoT) technologies. Cloud computing, big data, wireless sensor networks, cognitive radio ad hoc networks, and end-user applications are just a few of the state-of-the?art technologies and solutions that are integrated into the Internet of Things (IoT). In this project, we try to show IoT solutions, and the integration of IoT into the smart agriculture industry is illustrated that can be implemented also.

3. Ideation and Proposed Solution

3.1 Empathy map

Fig. 2 shows empathy map of the proposed system.

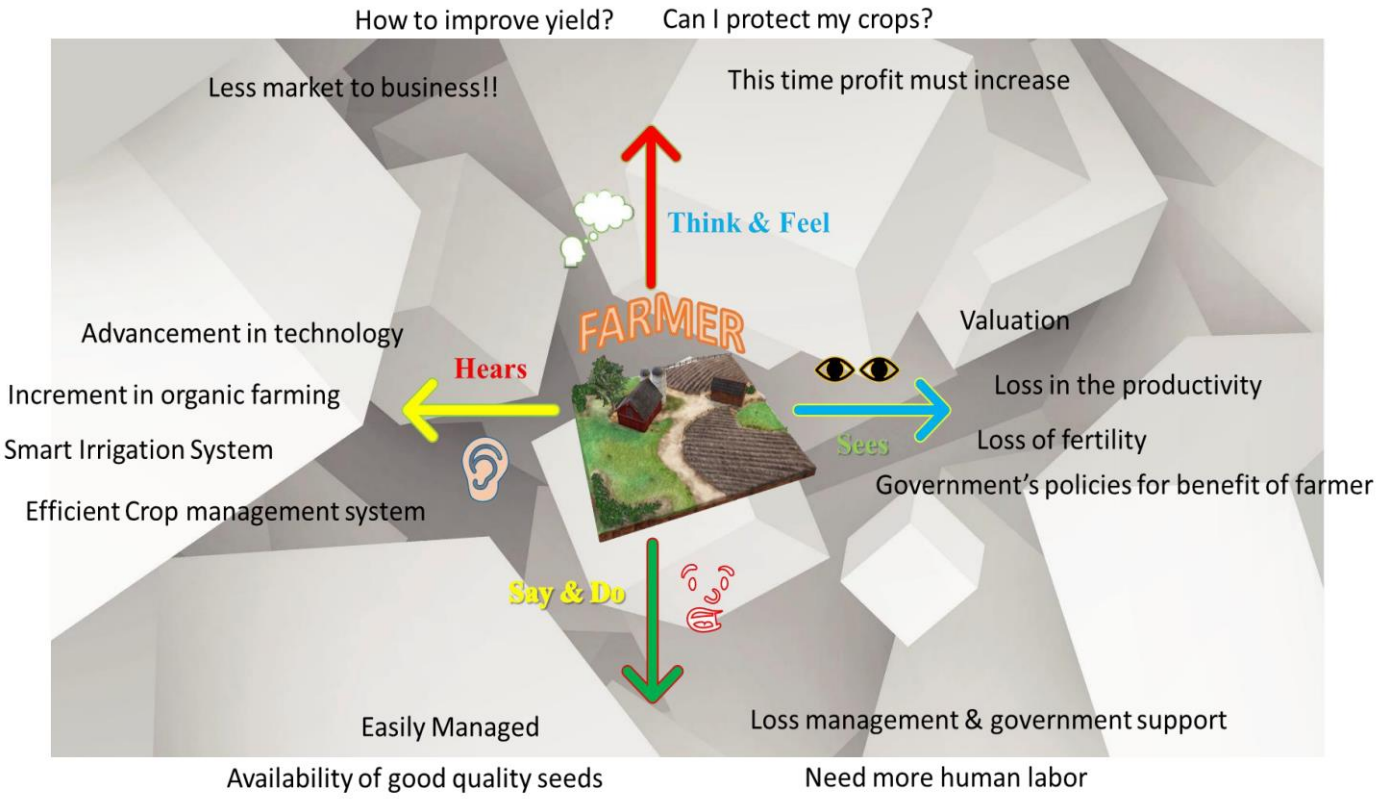


Fig.2 Empathy map of Proposed System

3.2 Ideation and Brainstorming

Fig.3 shows the brainstorming of proposed system.

Fig.3 Brainstorming of Proposed System

3.3 Proposed Solution

Tab. 1 below shows the Parameters and their Description given by us to propose the solution for existing problems.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Our project's primary goals are to safeguard crops from animal damage and safely move animals away from crops.
2.	Idea / Solution description	Using latest technology ultrasonic sensors, we can find a solution by detecting the animal's presence and sending a signal to the controller to tell it to do something about it.
3.	Novelty / Uniqueness	Although the system is already employed by many, but this project promises to give an accurate results and also act unique in many of the regions of operation.
4.	Social Impact / Customer Satisfaction	To secure a field, farmers cannot barricade off vast tracts of land or stay there all day, they can take immediate action, which results in improved crop yields and increased profitability. Also limited from often visiting the farm and may feel more at peace with this approach.

5.	Business Model (Revenue Model)	Highly reliable, reduction of cost, and fully/partial automatic that makes the system pretty smarter.
6.	Scalability of the Solution	The potential to expand because smart farming is a modern emerging technology, minimal soil disturbance, plant diversity, ongoing live plant/root, and livestock integration are examples of soil armor.

Tab.1 Proposed Solution

3.4 Problem Solution fit

Fig. 4 depicts problem solution fit for this system.

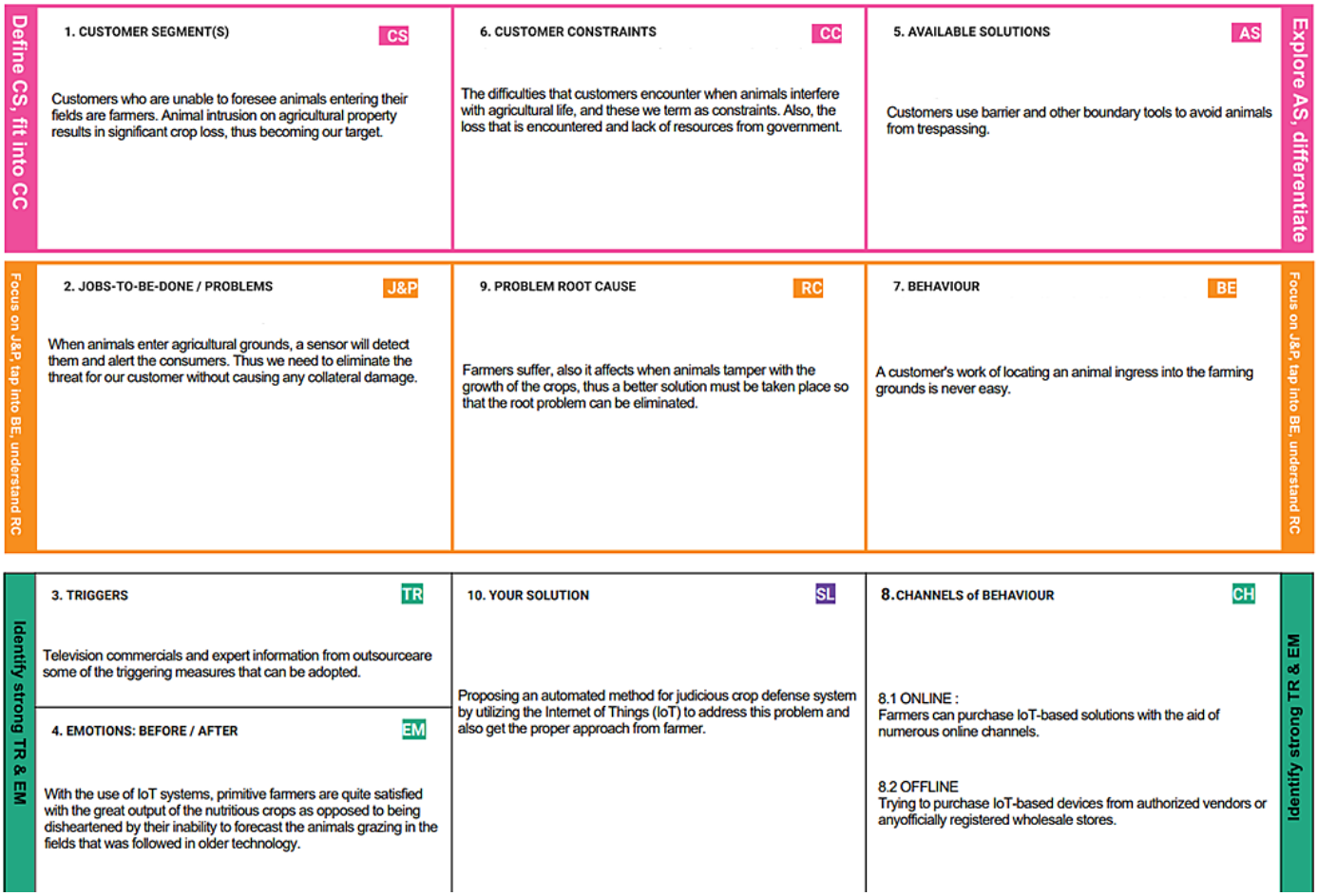


Fig. 4 Problem solution fit

4. Requirement Analysis

4.1 Functional Requirement

In Tab. 2 we can find the functional requirements of proposed solution.

S.NO.	Functional Requirement.	Sub Requirement.
1.	User Visibility	Sends an SMS to the farmer via cloud service when it detects animals approaching the crop field and sounds an alert to entice them away.
2.	User Reception	The values of the Data, the SMS messages are delivered from temperature, humidity, and soil moisture sensors.
3.	User Understanding	To obtain information about the current state of farming land, based on sensor data value.
4.	User Action	Actions that user must take include agricultural residue destruction, deep ploughing, crop rotation, fertilizers, strip cropping, and scheduled planting activities.

Tab. 2 Functional Requirement

4.2 Non- Functional Requirement

In tab. 3 we can find the non-functional requirements of the proposed solution.

S.NO.	Non-Functional Requirement	Description
1.	Usability	Given the capabilities of mobile devices, Mobile Support Users must be able to interact in the same roles & duties on desktops & mobile devices, if possible.
2.	Security	Authorized users of the system who share information must be able to register and communicate securely on devices with data that requires secure access.
3.	Reliability	Data could detect disturbances close to the field and doesn't issue an erroneous warning signal.
4.	Performance	Regardless of the amount of data that is saved and the background analytics, it must offer users acceptable response speeds. Communications that are bidirectional and nearly real-time must be supported. The necessity to support industrial and device protocols at the edge is connected to this requirement.

5.	Availability	Systems with high availability are necessary for IOT solutions and domains to operate around the clock. is not a vital production application, thus if the IOT solution goes down, neither operations nor production are affected.
6.	Scalability	The system must be able to handle growing load and data retention requirements based on an upscaling of the solution scope, such as additional buildings and manufacturing facilities.

Tab. 3 Non-functional Requirements

5. Project Design

5.1 Data flow Diagram

Fig. 5 below shows the milestone that is decided by us in Data flow diagram.

Fig. 5 Milestone of the Proposed System

Fig. 6 depicts the activity list planned by us for this system.

S. No	Activity Title	Activity Description	Duration
1.	Understanding the Project Requirement	Assign the team members & create the repository in GitHub. Assign the task to each members and teach how to use and open access the GitHub and IBM Career Education.	1 Week
2.	Starting of Project	Advice student to attend classes of IBM portals create and develop an rough diagram based on the project description and gather information of IOT and IBM project.	1 week
3.	Attend classes	Team members & team lead must watch and learn from classes provided by IBM and Nalaiya Thiran and must gain access of MIT License for their project.	4 Week
4.	Budget and scope of the project	Budget & analyse the use of IOT in the project and discuss with the team for budget prediction to predict the favourability of the customer to buy the product for efficient use of the product among the environment.	1 week

Fig. 6 Activity list of system design

Fig. 7 expands the blocks where in we are saying about the data flow diagram of this system.

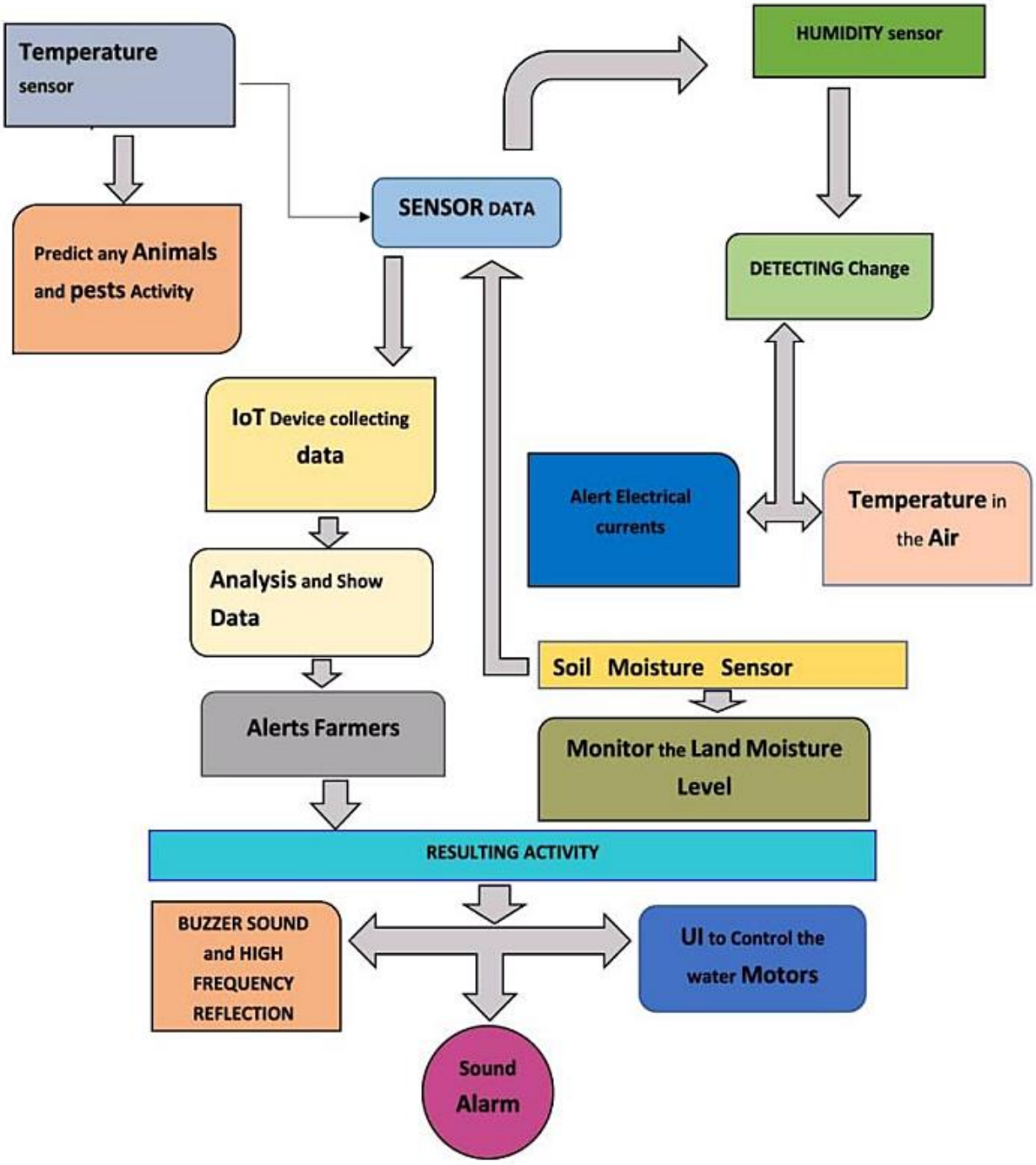


Fig. 7 Data flow diagram

In fig. 8 we can understand the implementation for this system as user data.

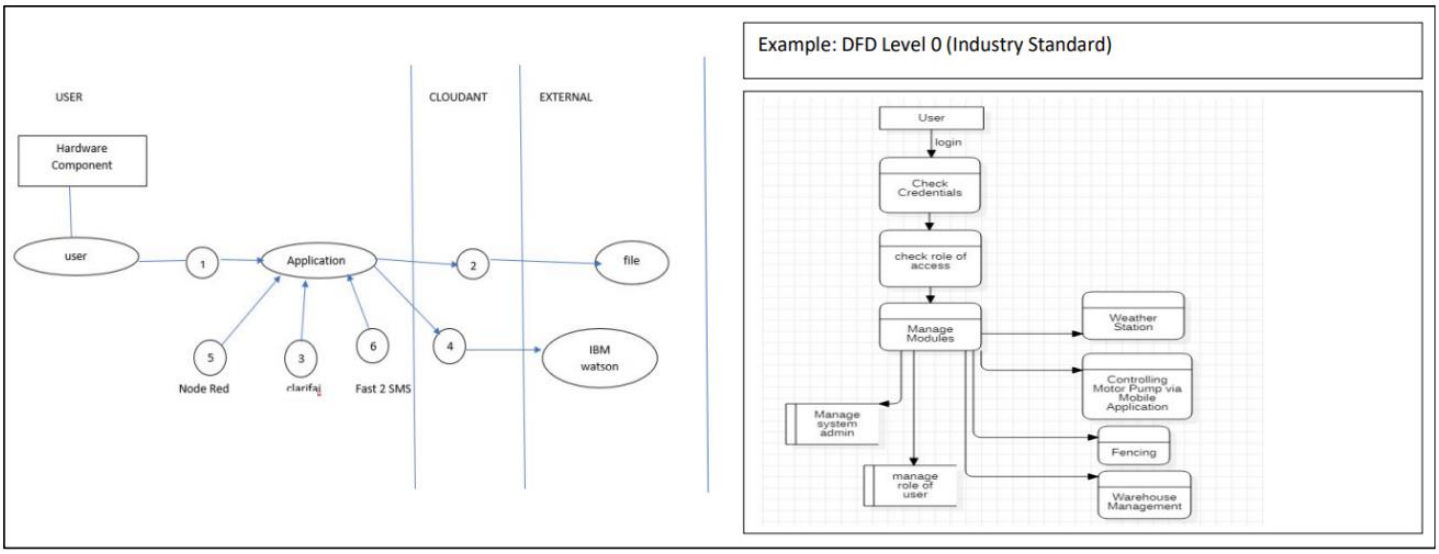


Fig. 8 Implementation

5.2 User stories

Tab. 4 which is tabulated below to make the User stories numbers.

User Type	Functional requirement (Epic)	User Story number	User Story/Task	Acceptance criteria	Priority	Release
Customer (Mobileuser)	Registration	USN-1	User can enter into the web application	I can access my account /dashboard	High	Sprint 1
		USN-2	User can register their credentials like email id and password	I can receive confirmation email & click confirm	High	Sprint 1
	Login	USN-3	User can log into the application by entering email & password	I can login to my account	High	Sprint 1
	Dashboard	USN-4	User can view the temperature	I can view the data given by the device	High	Sprint 2
		USN-5	User can view the level of sensor monitoring value	I can view the data given by the device	High	Sprint 2
Customer (Web user)	Usage	USN-1	User can view the web page and get the information	I can view the data given by the device	High	Sprint 3
Customer	Working	USN-1	User act according to the alert given by the device	I can get the data work according to it	High	Sprint 3
		USN-2	User turns ON the water motors/Buzzer/Sound Alarm when occur the disturbance on field.	I can get the data work according to it		Sprint 4

Customer care Executive	Action	USN-1	User solve the problem when some faces any usage issues	I can solve the issues when some one fails to understanding the procedure	High	Sprint 4
Administration	Administration	USN-1	User store every information	I can store the gained information	High	Sprint 4

Tab. 4 User Stories

5.3 Solution and Technical architecture

Solution Architecture:

Modern agriculture companies based on irrigation have become indispensable to Indian agriculture in recent years. With the help of this proposed technology, it is possible to accurately regulate the soil moisture, water level, humidity, and temperature of plants, thus improving the fertility and avoiding any interference of any threats like animals. It is possible for large farms to experience these variables varying from location to location because of varying atmospheric conditions, which makes it extremely difficult to maintain uniformity from one location to another. Using an Android phone as an advanced system is now possible for the first time, offering the possibility of maintaining constant environmental conditions and threat conditions. By going through fig. 9 below which shows about the solution architecture of this proposed system. In fig. 10 we can find the process flow diagram for the technical architecture that is designed by us.

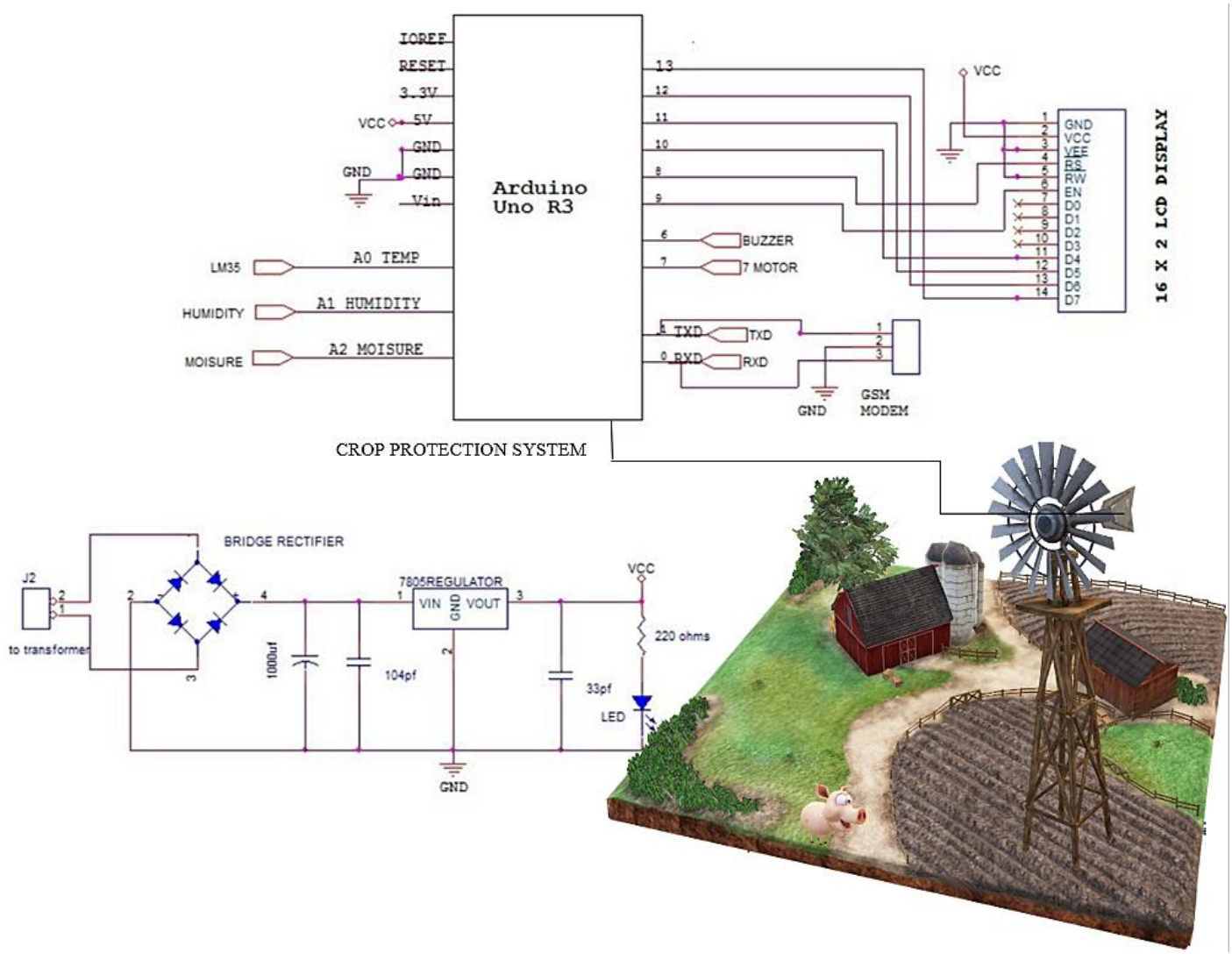


Fig. 9 Solution Architecture Diagram

Technology Stack (Architecture & Stack):

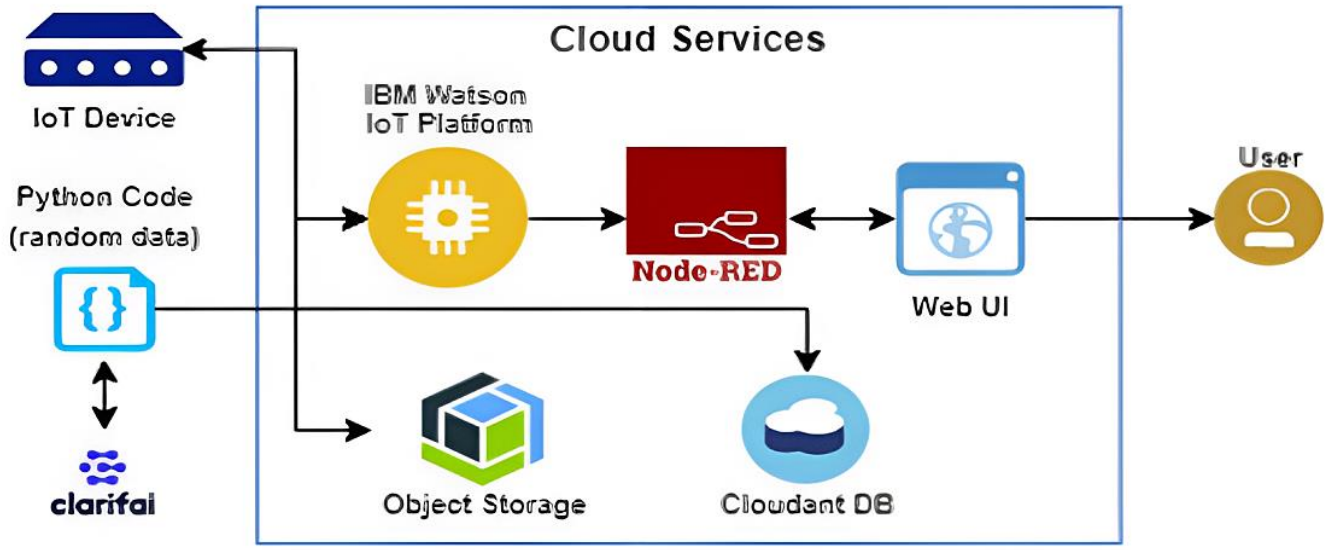


Fig. 10 Process Diagram of Technology Stack

Now coming towards the technology architecture, in tab. 5 we can get the details about the components and technologies that is govering this system. From tab. 6 we come to know about application characteristics.

S.No	Component	Description	Technology
1.	User Interface	How user interacts with the Web UI	App development
2.	Application Logic-1	Logic for a process in the application	Python Objectives
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	Node-RED service
5.	Database	Data Type	Database Cloudant DB
6.	Cloud Database	Database Service on Cloud	Cloud Object store service
7.	File Storage	File storage requirements	IBM Block Storage
8.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration:	Cloud Foundry

Tab. 5 Components & Technologies

S.No	Characteristics	Description	Technology
1.	Open-source Frameworks	The open-source frameworks used	SAN-SAF
2.	Security Implementations	List all the security / access controls implemented	IBM cloud encryptions
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	IBM cloud Architecture
4.	Availability	Justify the availability of applications (e.g. use of load balancers, distributed servers etc.)	Web Application can even be used by the framers in the horticulture
5.	Performance	Design consideration for the performance of the application	Since the web application is high efficient, it can be used by the farmers irrespective of time.

Tab. 6 Application Characteristics

6. Project planning and Scheduling

6.1 Sprint Planning

We have made a tabulation in the following format that can be seen in tab. 7. Here we can see the spring planning and we can then schedule the tasks its priority and task assigned team member name.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points (40)	Priority (Low to High)	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the required dataset by entering my email, password, and confirming my password.	3	High	R Bhargavi
Sprint-1		USN-2	As a user, I will receive confirmation email and the SMS once I have registered for the application	2	High	R Bhargavi
Sprint-2	Cloud services	USN-3	As a user, I can register for the application through Facebook or any social media	1	Low	S Meghana
Sprint-4		USN-4	As a user, I can register for the application through Gmail/web service	2	Medium	Abhishek
Sprint-3	Login	USN-5	As a user, I can log into the application network by entering email & password	4	High	Aakash V
Sprint-2	Pre processing	USN-6	As a farmer, the user must be able to find the system easy to access so pre-processes and other task must be perfect.	3	High	S Meghana
Sprint-1	Collecting Dataset	USN-7	To collect various sources of animal threats and keep developing a dataset.	3	Medium	R Bhargavi
Sprint-4	Integrating	USN-8	To integrate the available dataset and keep improving the accuracy of finding animals	2	High	Abhishek
Sprint-3		USN-9	To find and use appropriate compiler to run and test the data so that we can implement our program	1	Low	Aakash V

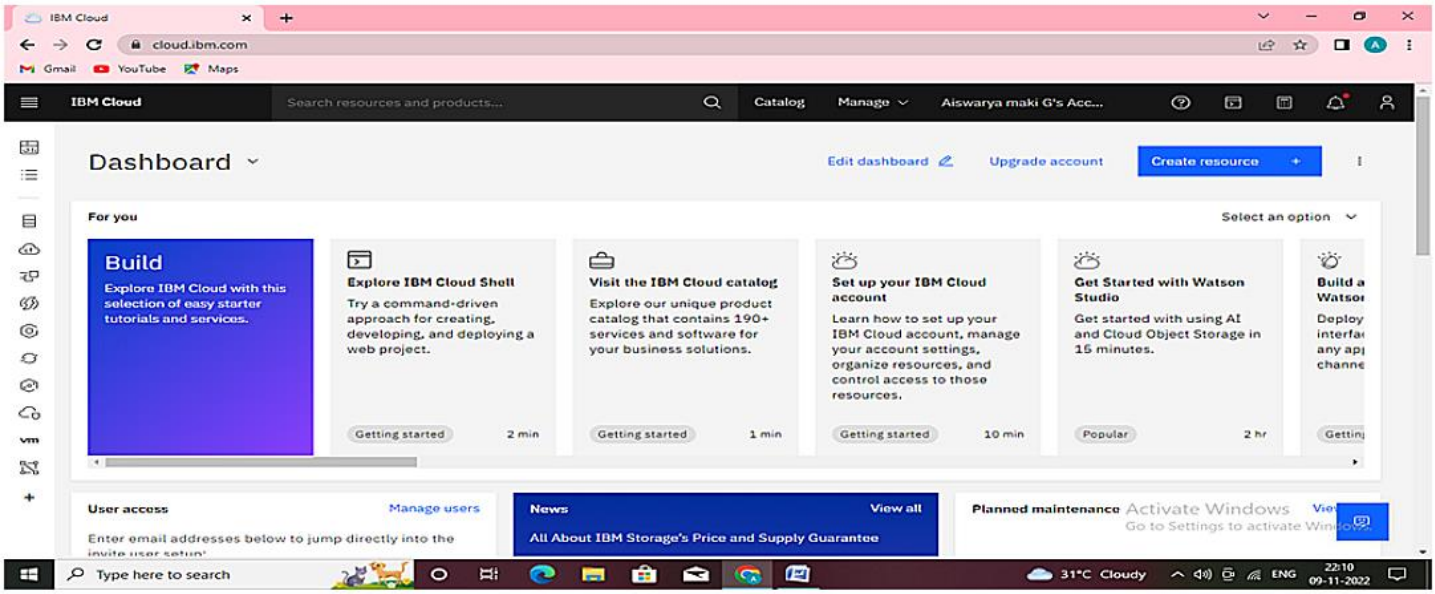
Sprint-2		USN-10	Request Saveetha Engineering College to deploy the project in our campus and test	1	Low	S Meghana
Sprint-1	Training	USN-11	As programmer, we need to train our data perfectly so that the program runs smoothly	3	High	R Bhargavi
Sprint-3		USN-12	Train the data using out available services and IBM dataset from server and improve that	2	Medium	Aakash
Sprint-4	Coding	USN-13	To modify the code according to our program and improve the efficiency of that code	4	High	Abhishek
Sprint-2		USN-13	To improve performance	1	Low	S Meghana
Sprint-2	Record	USN-5	To record the data and plot the graph to show the characteristics officially	4	High	S Meghana
Sprint-1	Planning	USN-4	Plan the programming language and feasibility	3	Medium	Bhargavi, Abhishek
Sprint-4		USN-14	Demonstrate the working and improve accuracy overall	2	Low	Abhishek

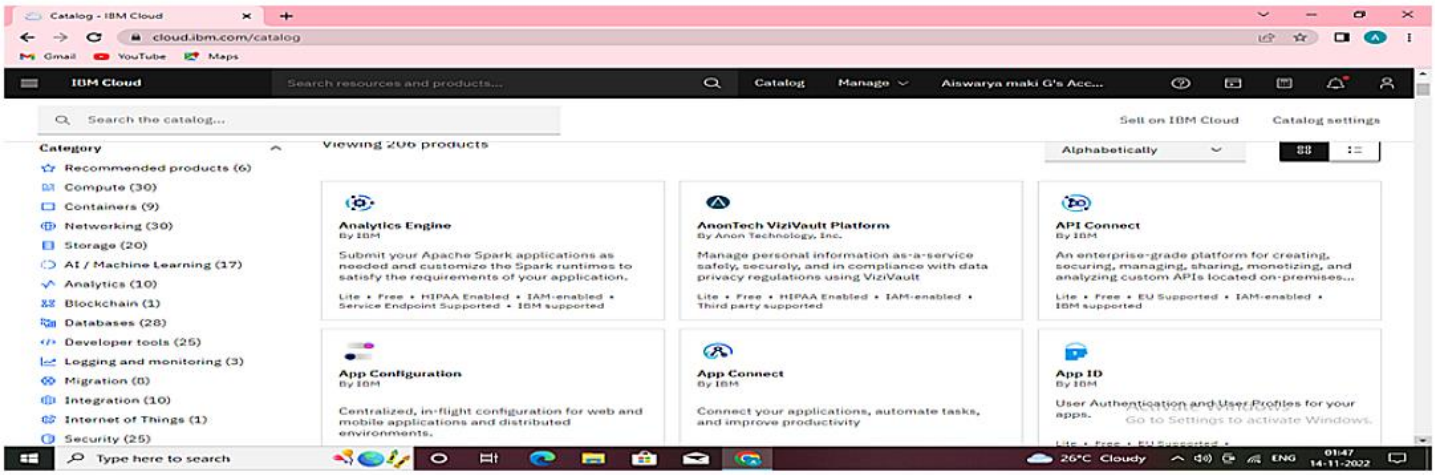
Tab. 7 Sprint planning

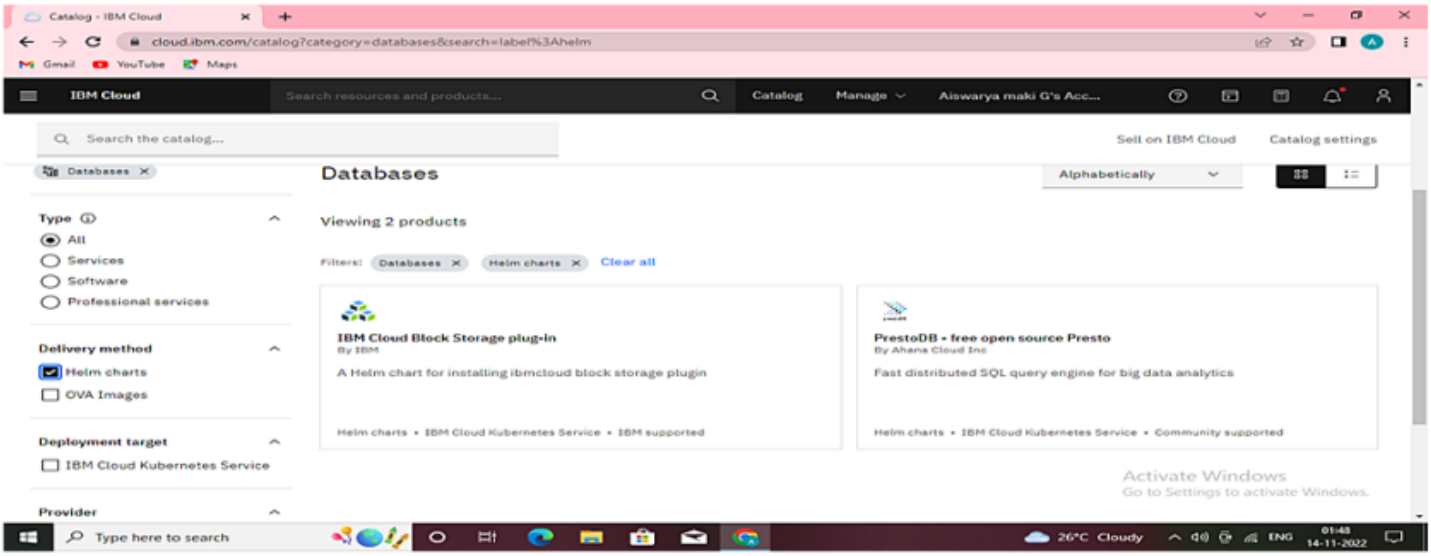
Sprint Estimation

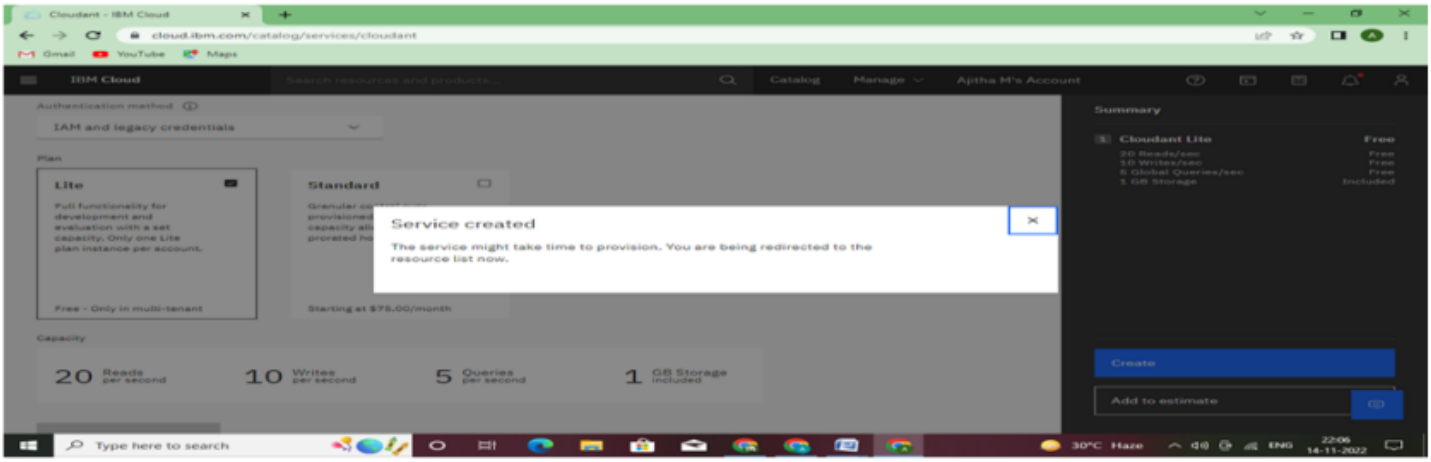
A sprint estimation shows how much effort a series of tasks require. It's based on assumptions, requirements, and dependencies of a project.

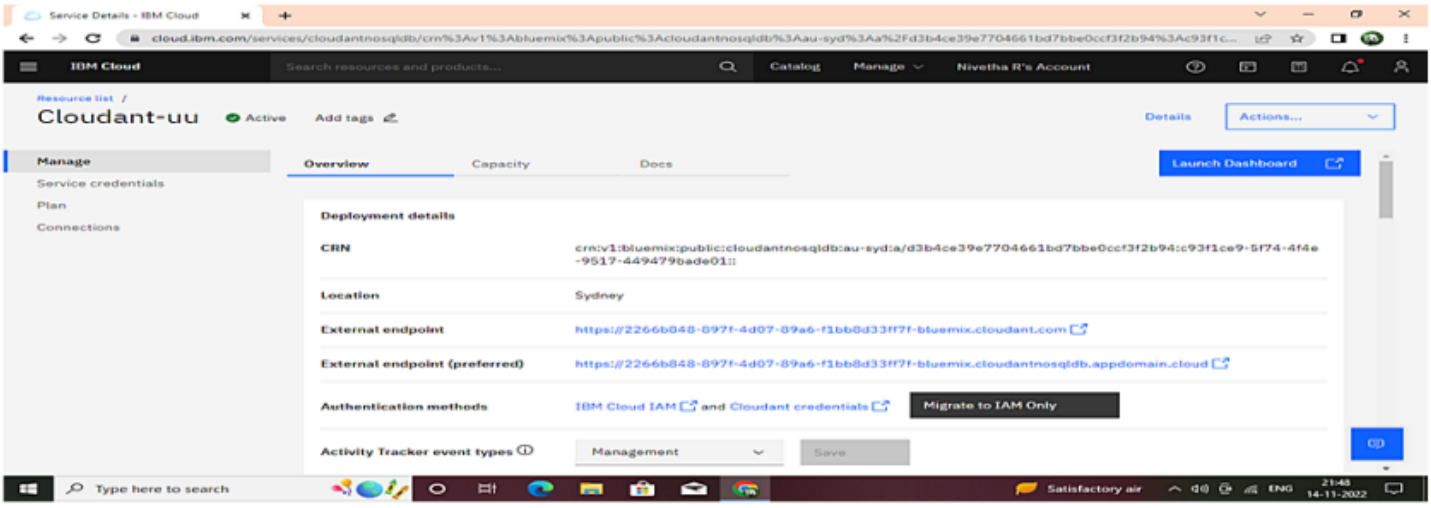
SPRINT 1:

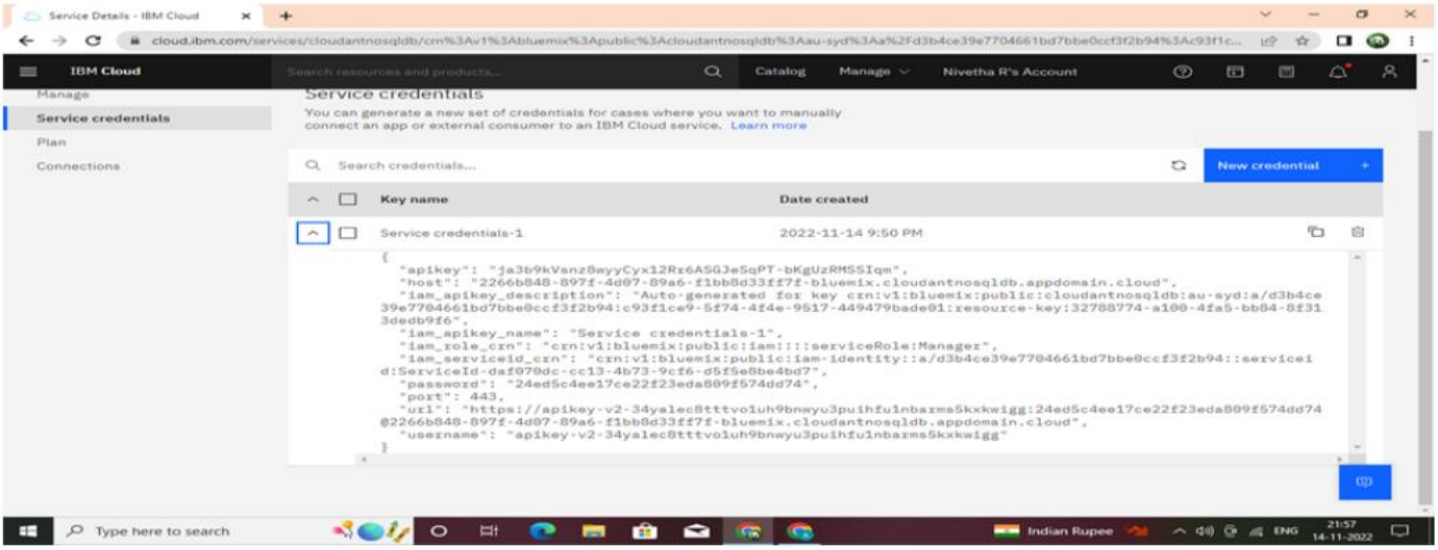




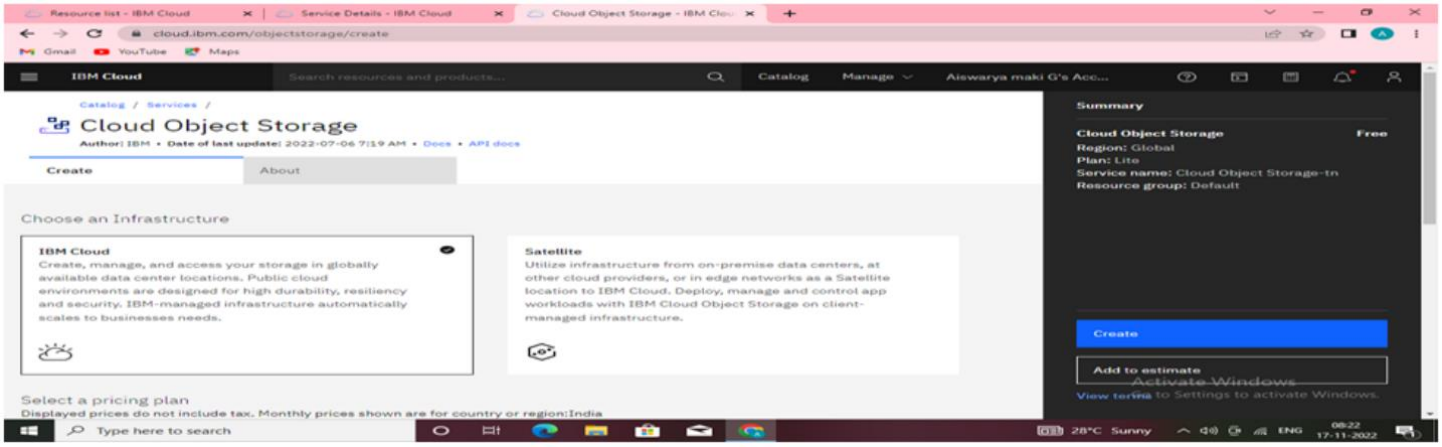

Catalog - IBM Cloud
cloud.ibm.com/catalog?category=databases&search=label%3Ahelm
IBM Cloud
Search resources and products...
Catalog
Manage
Aiswarya maki G's Acc...
Search the catalog...
Sell on IBM Cloud
Catalog settings
Databases
Databases
Alphabetically
Type
All
Services
Software
Professional services
Viewing 2 products
Filters: Databases Helm charts Clear all
Delivery method
Helm charts
OVA Images
IBM Cloud Block Storage plug-in
By IBM
A Helm chart for installing ibmcloud block storage plugin
Helm charts • IBM Cloud Kubernetes Service • IBM supported
PrestoDB - free open source Presto
By Ahana Cloud Inc
Fast distributed SQL query engine for big data analytics
Helm charts • IBM Cloud Kubernetes Service • Community supported
Deployment target
IBM Cloud Kubernetes Service
Provider
Activate Windows
Go to Settings to activate Windows.
Type here to search
26°C Cloudy
ENG
01:48
14-11-2022

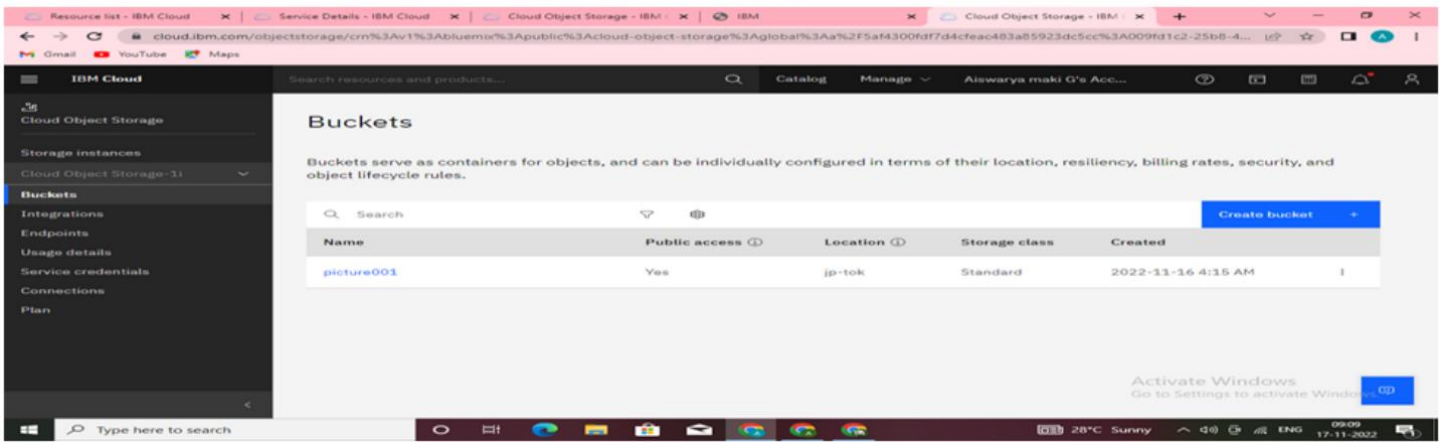

Cloudant - IBM Cloud
cloud.ibm.com/catalog/services/cloudant
IBM Cloud
Catalog
Manage
Ajitha M's Account
Authentication method
IAM and legacy credentials
Plan
Lite
Full functionality for development and evaluation with a set capacity. Only one Lite plan instance per account.
Free - Only in multi-tenant
Standard
Starting at $79.00/month
Capacity
20 Reads per second
10 Writes per second
5 Queries per second
1 GB Storage included
Service created
The service might take time to provision. You are being redirected to the resource list now.
Summary
Cloudant Lite
Free
Create
Add to estimate
Type here to search
30°C Haze
ENG
22:06
14-11-2022

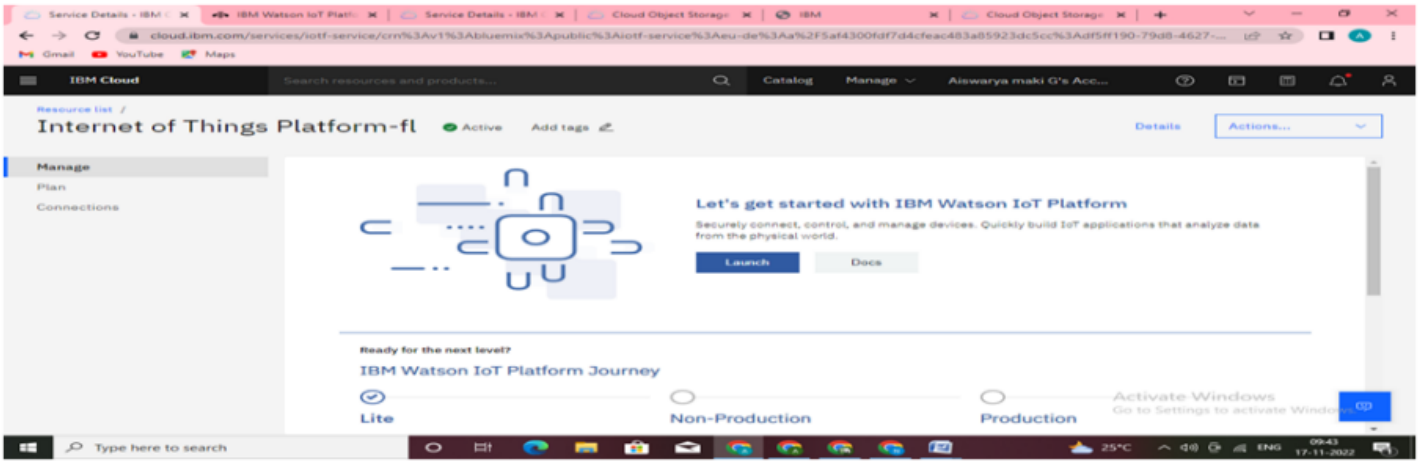

Service Details - IBM Cloud
cloud.ibm.com/services/cloudantnosqldb/crn%3Av1%3Abluemix%3Apublic%3Acloudantnosqldb%3Aau-syd%3Aa%2Fd3b4ce39e7704661bd7bbe0ccf3f2b94%3Ac93f1c...
IBM Cloud
Catalog
Manage
Nivetha R's Account
Resource list /
Cloudant-uu
Active
Add tags
Details
Actions...
Manage
Service credentials
Plan
Connections
Overview
Capacity
Docs
Launch Dashboard
Deployment details
CRN
crn:v1:bluemix:public:cloudantnosqldb:au-syd:a/d3b4ce39e7704661bd7bbe0ccf3f2b94:c93f1ce9-5f74-4f4e-9517-449479bade01::
Location
Sydney
External endpoint
https://2266b848-897f-4d07-89a6-f1bb8d33ff7f-bluemix.cloudant.com
External endpoint (preferred)
https://2266b848-897f-4d07-89a6-f1bb8d33ff7f-bluemix.cloudantnosqldb.appdomain.cloud
Authentication methods
IBM Cloud IAM and Cloudant credentials
Migrate to IAM Only
Activity Tracker event types
Management
Save
Type here to search
Satisfactory air
ENG
21:48
14-11-2022

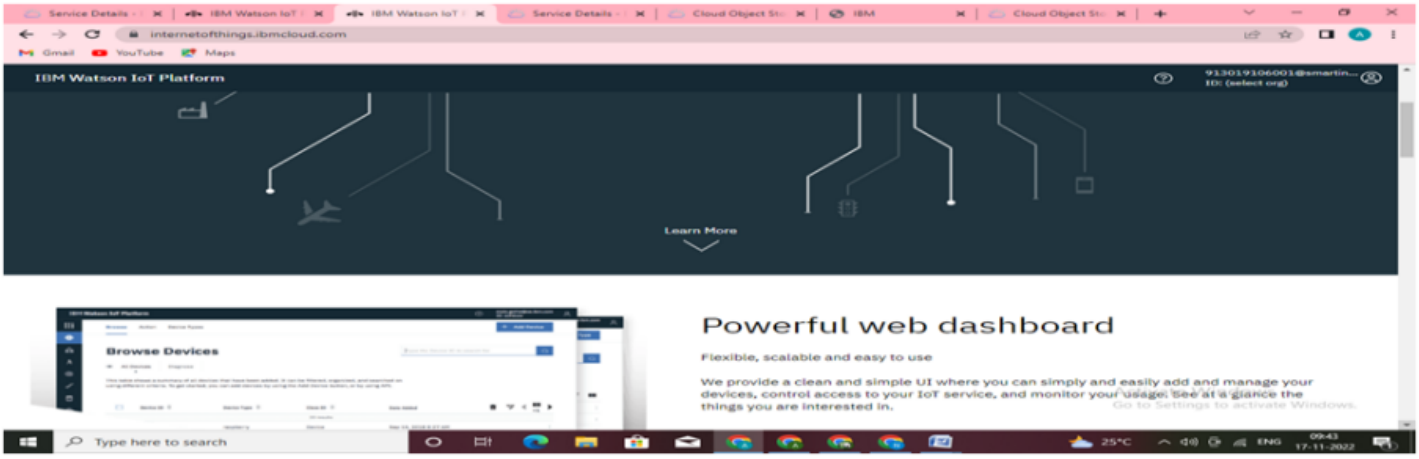

Service Details - IBM Cloud
IBM Cloud
Catalog
Manage
Nivetha R's Account
Service credentials
You can generate a new set of credentials for cases where you want to manually connect an app or external consumer to an IBM Cloud service. Learn more
Plan
Connections
Search credentials...
New credential
Key name
Date created
Service credentials-1
2022-11-14 9:50 PM
"apikey": "ja3b9kVanz8wyyCyx12Rr6ASGJeSqPT-bKgUzRMSSIqm",
"host": "2266b848-897f-4d07-89a6-f1bb8d33ff7f-bluemix.cloudantnosqldb.appdomain.cloud",
"iam_apikey_description": "Auto-generated for key crn:v1:bluemix:public:cloudantnosqldb:au-syd:a/d3b4ce39e7704661bd7bbe0ccf3f2b94:c93f1ce9-5f74-4f4e-9517-449479bade01:resource-key:32788774-a100-4fa5-bb84-8f313dedb9f6",
"iam_apikey_name": "Service credentials-1",
"iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Manager",
"iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity::a/d3b4ce39e7704661bd7bbe0ccf3f2b94::serviceid:ServiceId-daf070dc-cc13-4b73-9cf6-d5f5e8be4bd7",
"password": "24ed5c4ee17ce22f23eda809f574dd74",
"port": 443,
"url": "https://apikey-v2-34yalec8tttvo1uh9bnwyu3puihfu1nbarms5kxkwigg:24ed5c4ee17ce22f23eda809f574dd74@2266b848-897f-4d07-89a6-f1bb8d33ff7f-bluemix.cloudantnosqldb.appdomain.cloud",
"username": "apikey-v2-34yalec8tttvo1uh9bnwyu3puihfu1nbarms5kxkwigg"
Type here to search
Indian Rupee
ENG
21:57
14-11-2022

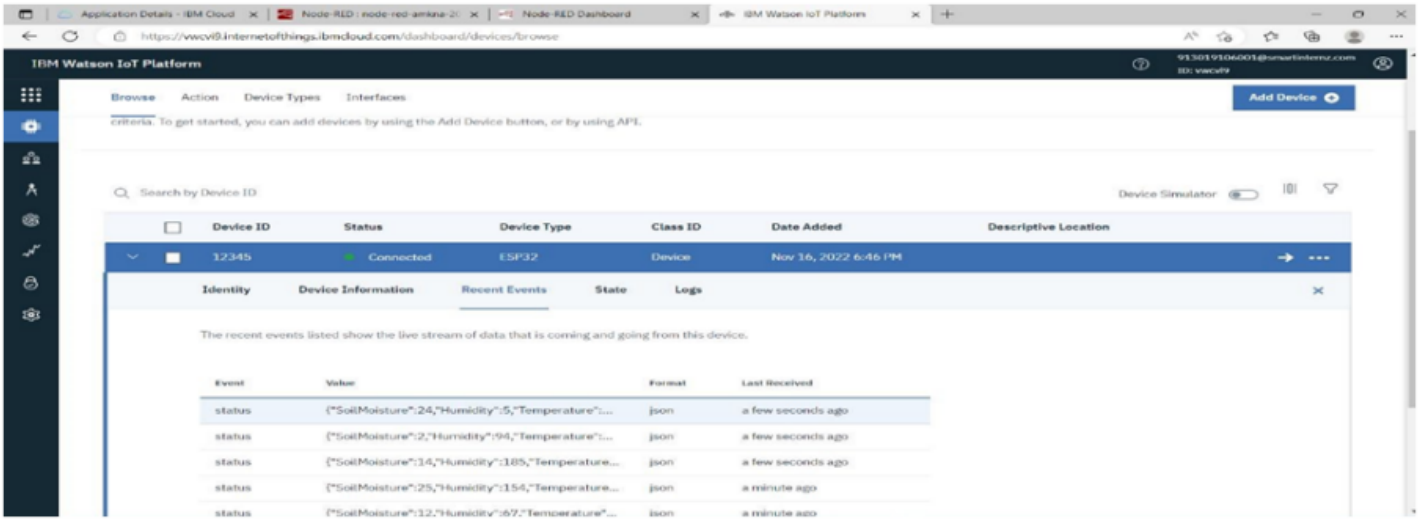
SPRINT 2:


Resource list - IBM Cloud
Service Details - IBM Cloud
Cloud Object Storage - IBM Cloud
cloud.ibm.com/objectstorage/create
IBM Cloud
Catalog
Manage
Aiswarya maki G's Acc...
Catalog / Services /
Cloud Object Storage
Author: IBM • Date of last update: 2022-07-06 7:19 AM • Docs • API docs
Create
About
Choose an Infrastructure
IBM Cloud
Create, manage, and access your storage in globally available data center locations. Public cloud environments are designed for high durability, resiliency and security. IBM-managed infrastructure automatically scales to businesses needs.
Satellite
Utilize infrastructure from on-premise data centers, at other cloud providers, or in edge networks as a Satellite location to IBM Cloud. Deploy, manage and control app workloads with IBM Cloud Object Storage on client-managed infrastructure.
Select a pricing plan
Displayed prices do not include tax. Monthly prices shown are for country or region:India
Summary
Cloud Object Storage
Free
Region: Global
Plan: Lite
Service name: Cloud Object Storage-tn
Resource group: Default
Create
Add to estimate
View terms
Activate Windows
Go to Settings to activate Windows.
Type here to search
28°C Sunny
ENG
08:22
17-11-2022

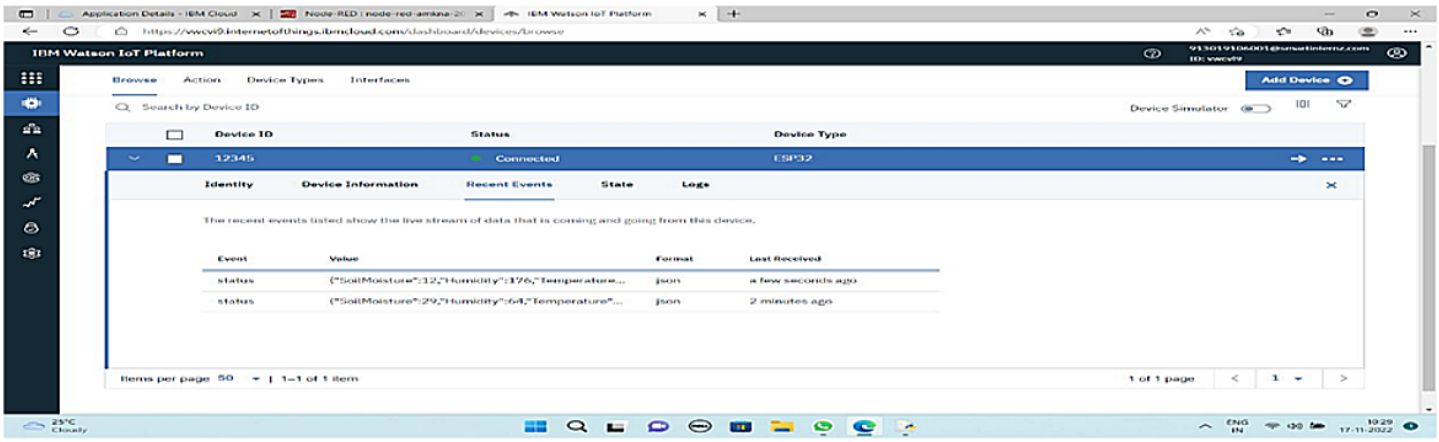

Resource list - IBM Cloud
Service Details - IBM Cloud
Cloud Object Storage - IBM
IBM
cloud.ibm.com/objectstorage/crn%3Av1%3Abluemix%3Apublic%3Acloud-object-storage%3Aglobal%3Aa%2F5af4300fdf7d4cfeac483a85923dc5cc%3A009fd1c2-25b8-4...
IBM Cloud
Catalog
Manage
Aiswarya maki G's Acc...
Cloud Object Storage
Storage instances
Cloud Object Storage-1
Buckets
Integrations
Endpoints
Usage details
Service credentials
Connections
Plan
Buckets
Buckets serve as containers for objects, and can be individually configured in terms of their location, resiliency, billing rates, security, and object lifecycle rules.
Search
Create bucket
Name
Public access
Location
Storage class
Created
picture001
Yes
jp-tok
Standard
2022-11-16 4:15 AM
Activate Windows
Go to Settings to activate Windows.
Type here to search
28°C Sunny
ENG
09:09
17-11-2022

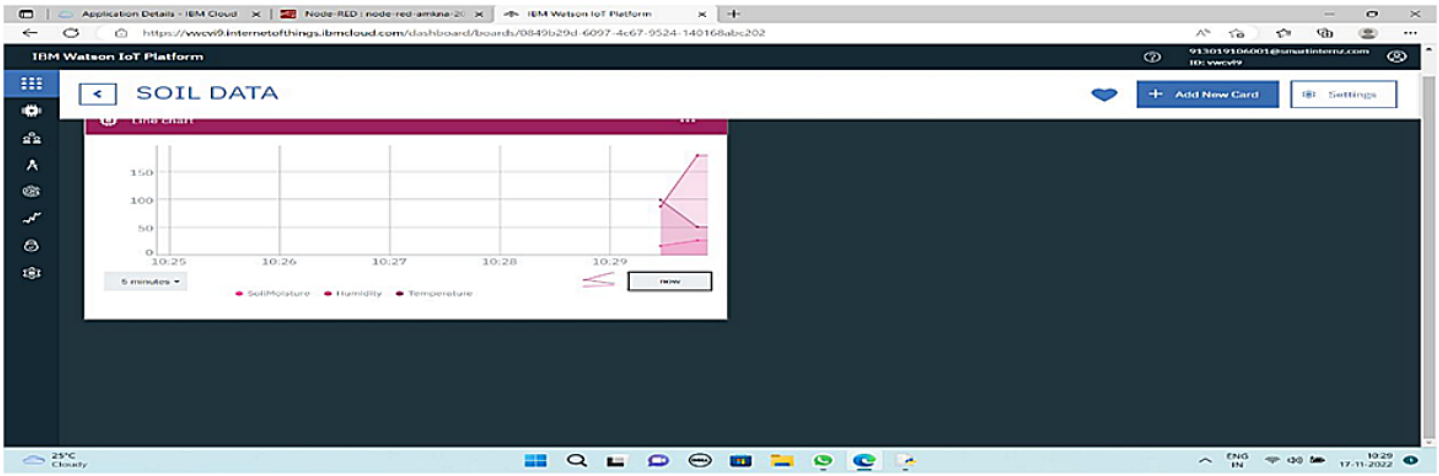

IBM Cloud
Internet of Things Platform-fl
Active
Add tags
Details
Actions...
Manage
Plan
Connections
Let's get started with IBM Watson IoT Platform
Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.
Launch
Docs
Ready for the next level?
IBM Watson IoT Platform Journey
Lite
Non-Production
Production
Activate Windows
Type here to search

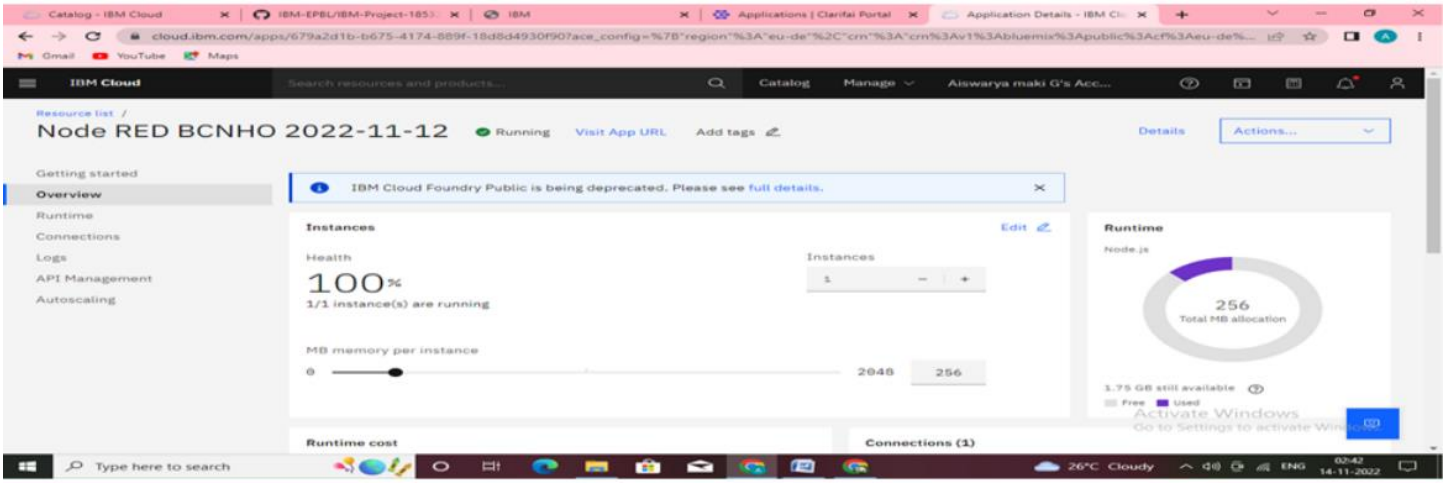

IBM Watson IoT Platform
Learn More
Browse Devices
Powerful web dashboard
Flexible, scalable and easy to use
We provide a clean and simple UI where you can simply and easily add and manage your devices, control access to your IoT service, and monitor your usage. See at a glance the things you are interested in.
Type here to search

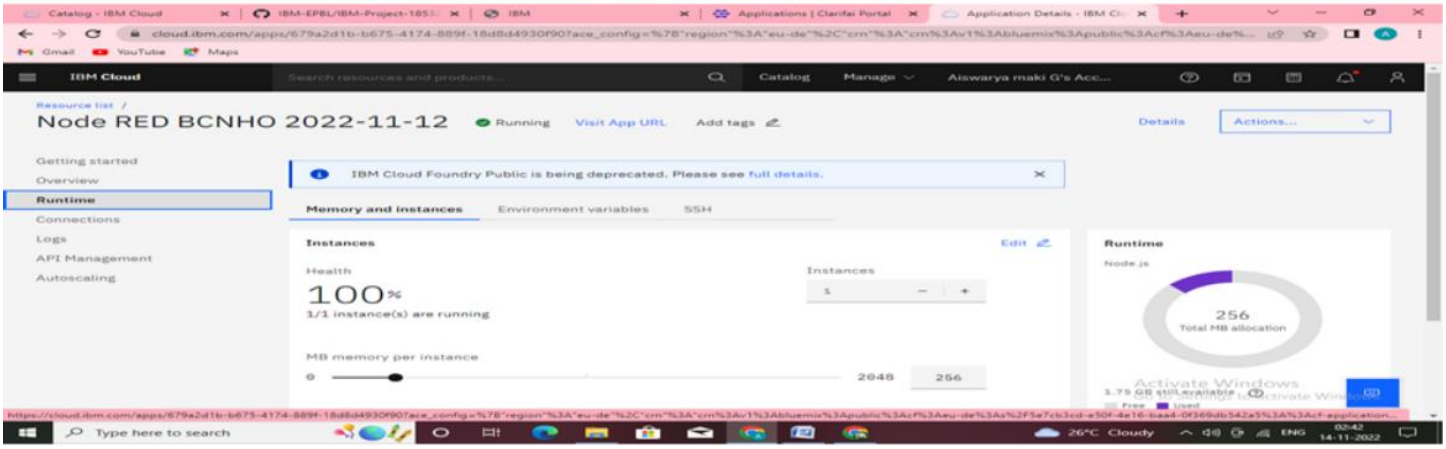

IBM Watson IoT Platform
Browse
Action
Device Types
Interfaces
Add Device
Search by Device ID
Device Simulator
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
12345
Connected
ESP32
Device
Nov 16, 2022 6:46 PM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
status
{"SoilMoisture":24,"Humidity":5,"Temperature":...
json
a few seconds ago
status
{"SoilMoisture":2,"Humidity":94,"Temperature":...
json
a few seconds ago
status
{"SoilMoisture":14,"Humidity":185,"Temperature...
json
a few seconds ago
status
{"SoilMoisture":25,"Humidity":154,"Temperature...
json
a minute ago
status
{"SoilMoisture":12,"Humidity":67,"Temperature"...
json
a minute ago

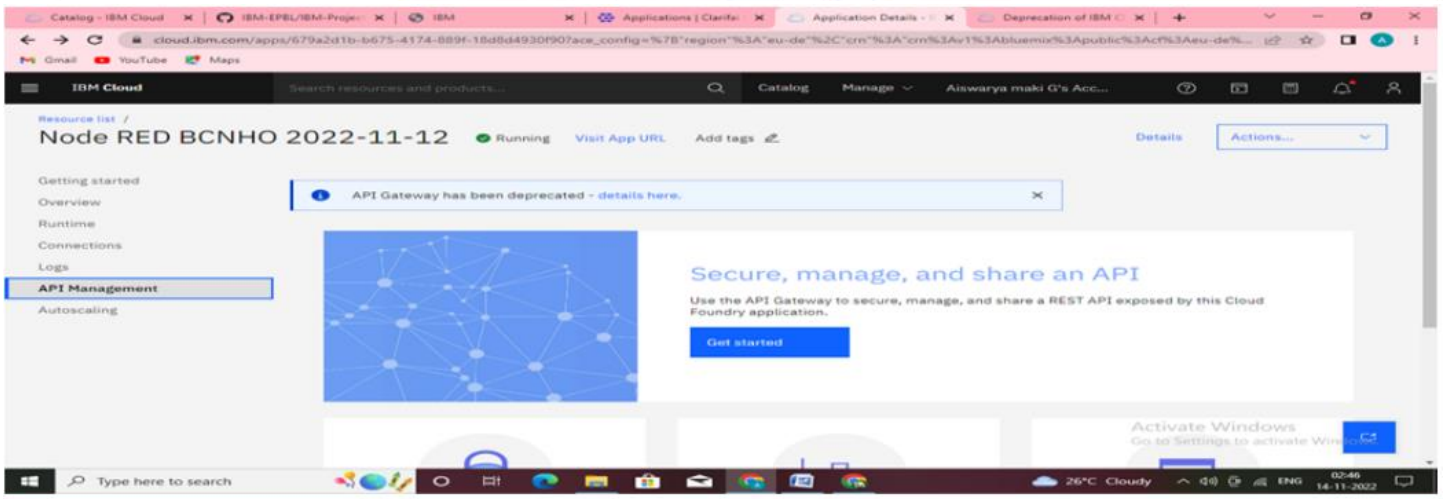
SPRINT 3:

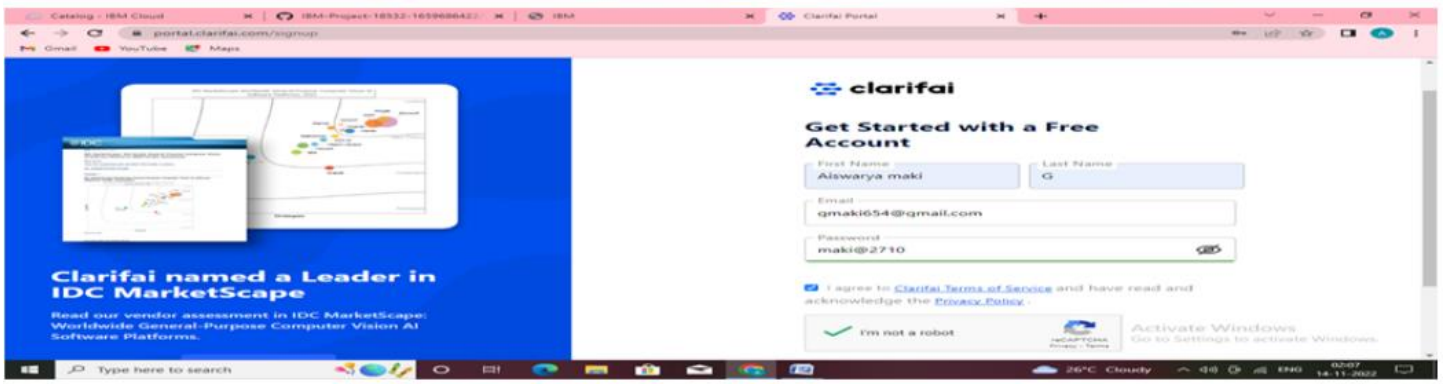

IBM Watson IoT Platform
Browse
Action
Device Types
Interfaces
Add Device
Search by Device ID
Device Simulator
Device ID
Status
Device Type
12345
Connected
ESP32
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
status
{"SoilMoisture":32,"Humidity":176,"Temperature...
json
a few seconds ago
status
{"SoilMoisture":29,"Humidity":64,"Temperature"...
json
2 minutes ago
Items per page 50
1–1 of 1 item
1 of 1 page

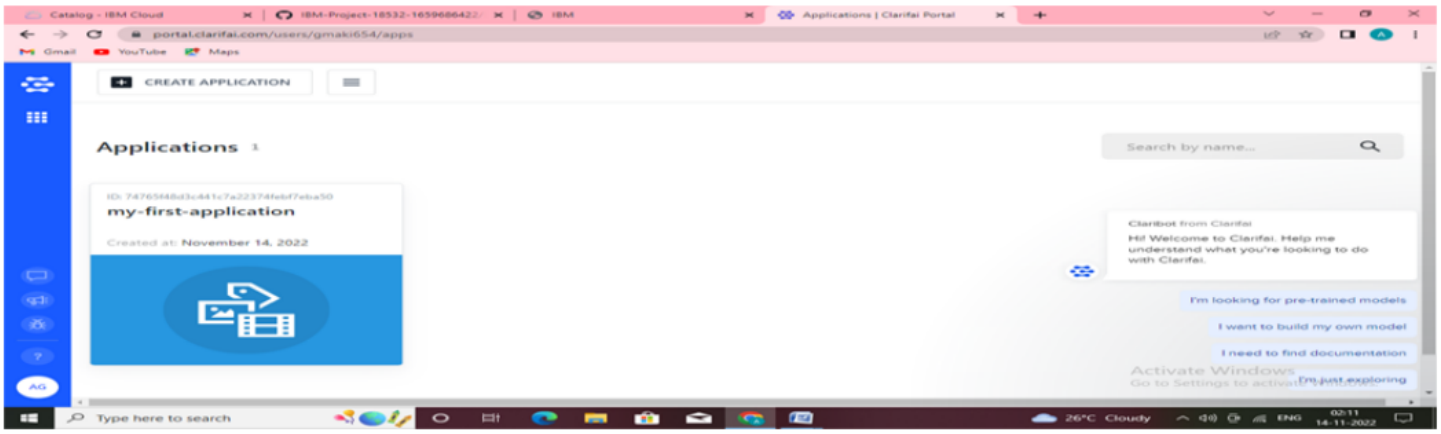

IBM Watson IoT Platform
SOIL DATA
Add New Card
Settings
Line Chart
150
100
50
0
10:25
10:26
10:27
10:28
10:29
5 minutes
now
SoilMoisture
Humidity
Temperature

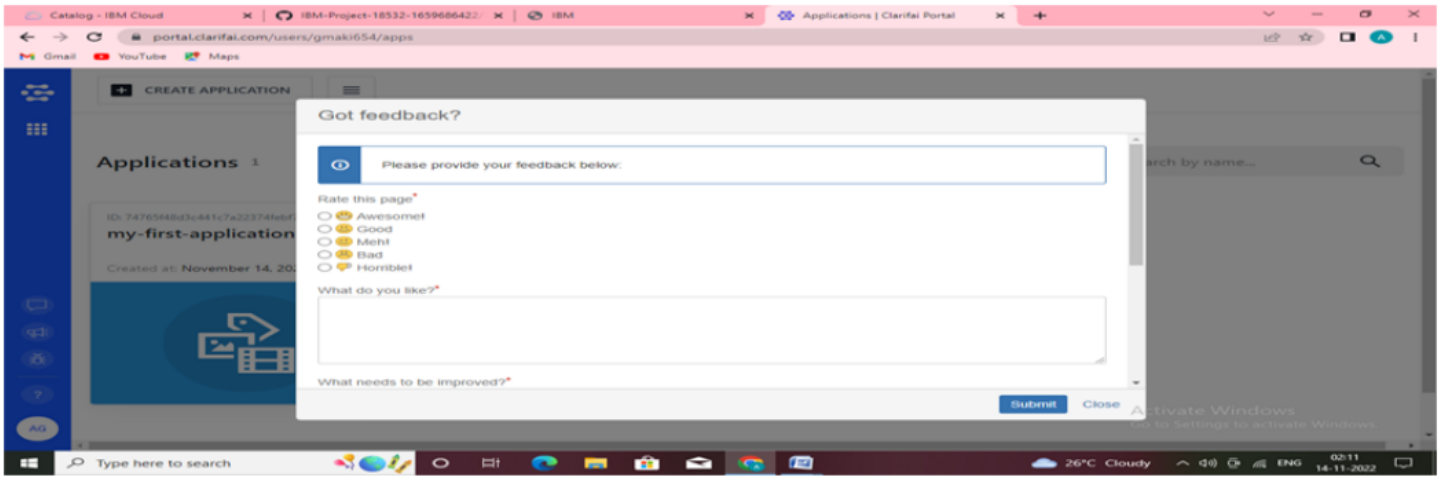

IBM Cloud
Node RED BCNHO 2022-11-12
Running
Visit App URL
Add tags
Details
Actions...
Getting started
Overview
Runtime
Connections
Logs
API Management
Autoscaling
IBM Cloud Foundry Public is being deprecated. Please see full details.
Instances
Edit
Health
100%
1/1 instance(s) are running
Instances
MB memory per instance
2048
256
Runtime
Node.js
256
Total MB allocation
1.75 GB still available
Runtime cost
Connections (1)


IBM Cloud
Node RED BCNHO 2022-11-12
Running
Visit App URL
Add tags
Details
Actions...
Getting started
Overview
Runtime
Connections
Logs
API Management
Autoscaling
IBM Cloud Foundry Public is being deprecated. Please see full details.
Memory and instances
Environment variables
SSH
Instances
Edit
Health
100%
1/1 instance(s) are running
Instances
MB memory per instance
2048
256
Runtime
Node.js
256
Total MB allocation


IBM Cloud
Node RED BCNHO 2022-11-12
Running
Visit App URL
Add tags
Details
Actions...
Getting started
Overview
Runtime
Connections
Logs
API Management
Autoscaling
API Gateway has been deprecated - details here.
Secure, manage, and share an API
Use the API Gateway to secure, manage, and share a REST API exposed by this Cloud Foundry application.
Get started


Clarifai named a Leader in IDC MarketScape
Read our vendor assessment in IDC MarketScape: Worldwide General-Purpose Computer Vision AI Software Platforms.
clarifai
Get Started with a Free Account
First Name
Aiswarya maki
Last Name
G
Email
gmaki654@gmail.com
Password
maki@2710
I'm not a robot



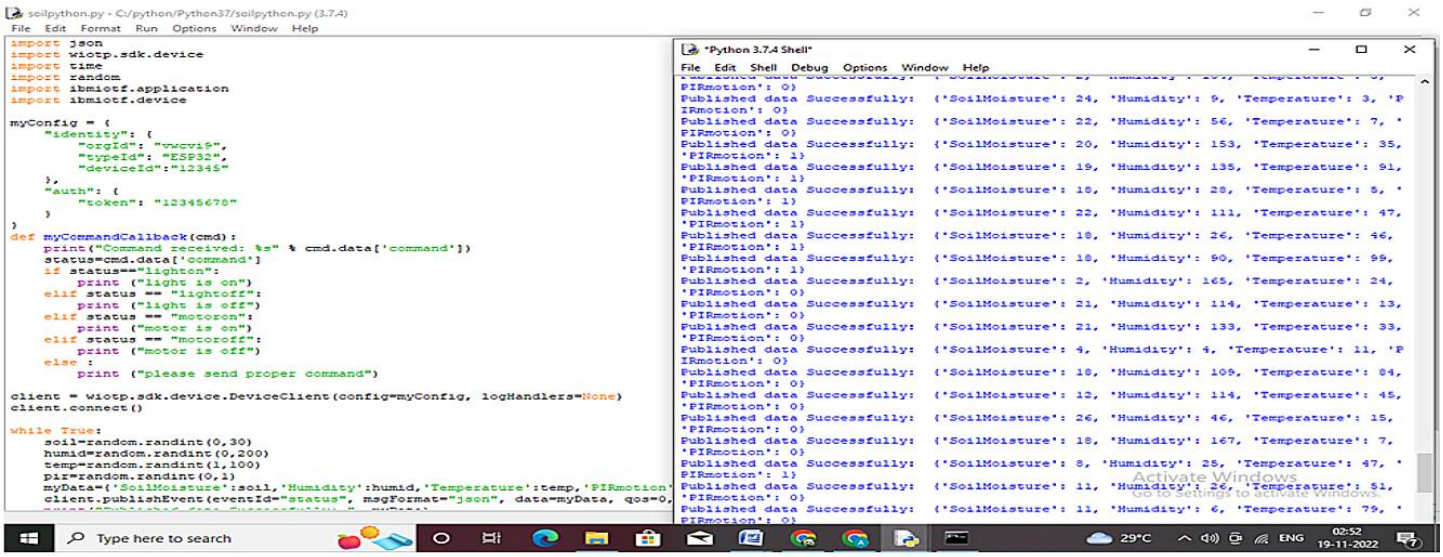


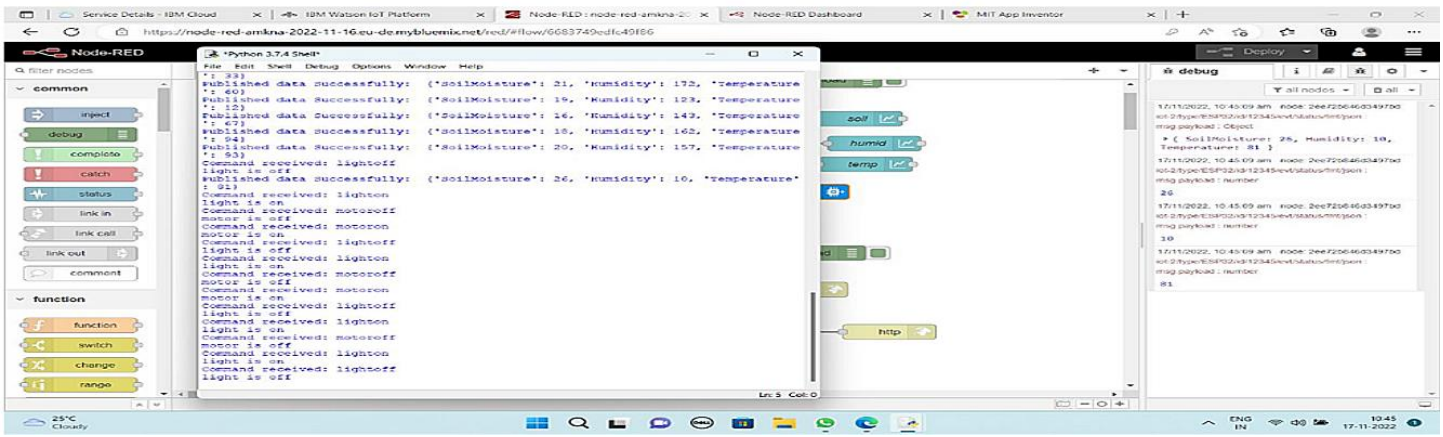
SPRINT 4:

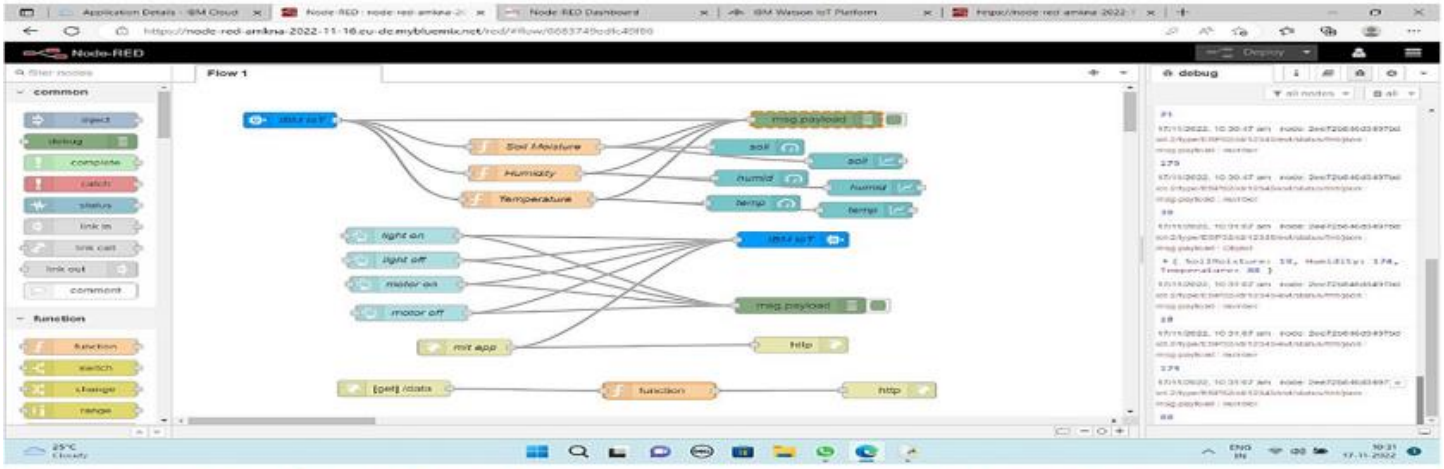
Programming:

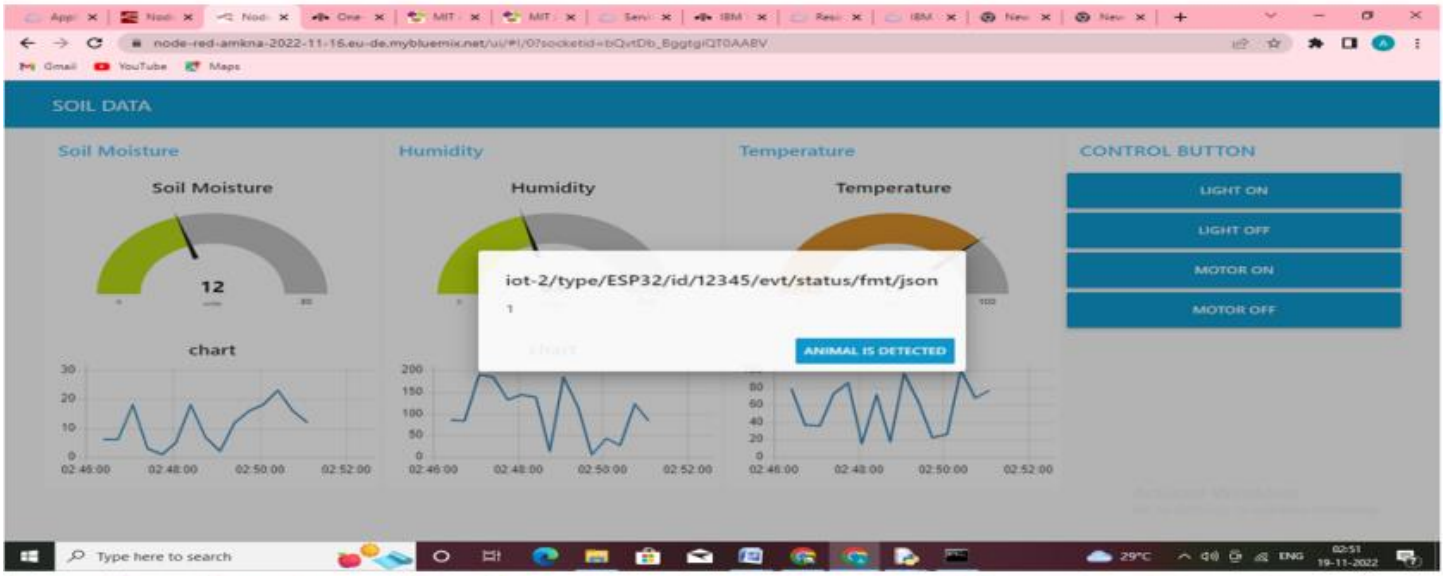
```
import json
import wiotp.sdk.device
import time
import random
import ibmiotf.application
import ibmiotf.device
myConfig = { "identity": { "orgId": "vwcvi9", "typeId": "ESP32", "deviceId":"12345" }, "auth": { "token": "12345678" } } def
myCommandCallback(cmd):
        print("Command received: %s" % cmd.data['command'])
        status=cmd.data['command']
        if status=="lighton":
        print ("light is on")
elif status == "lightoff":
        print ("light is off")
elif status == "motoron":
        print ("motor is on")
elif status == "motoroff":
        print ("motor is off")
else :
        print ("please send proper command")

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
```

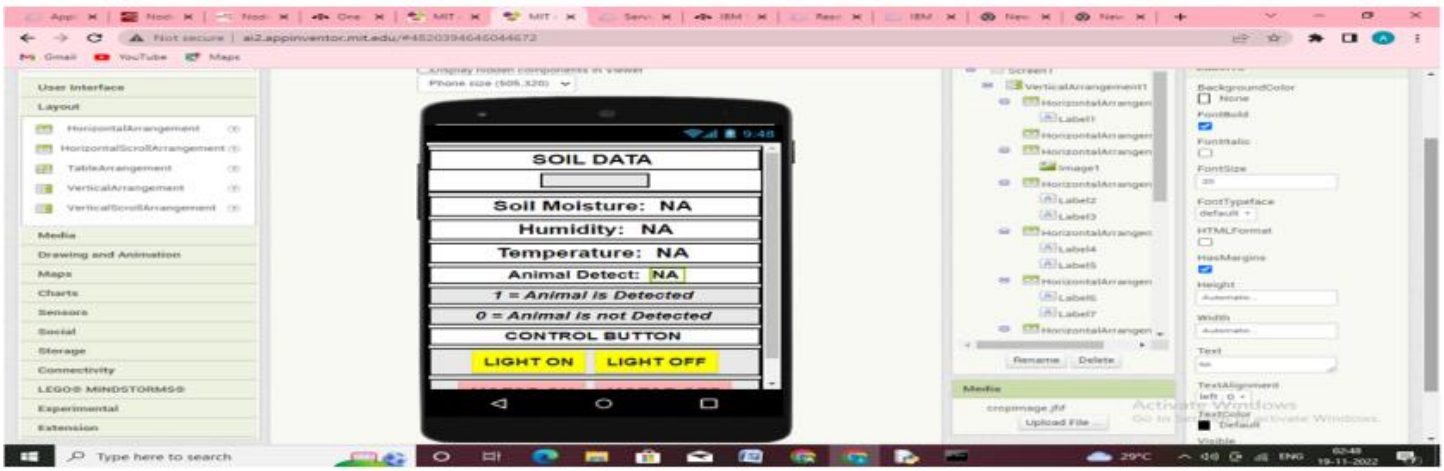
```
client.connect()
while True:
        soil=random.randint(0,30)
humid=random.randint(0,200)
temp=random.randint(1,100)
pir=random.randint(0,1)
myData={'SoilMoisture':soil,'Humidity':humid,'Temperature':temp,'PIRmotion ':pir}
      client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
print("Published data Successfully: ", myData)
        time.sleep(20)
        client.commandCallback = myCommandCallback
client.disconnect()
```

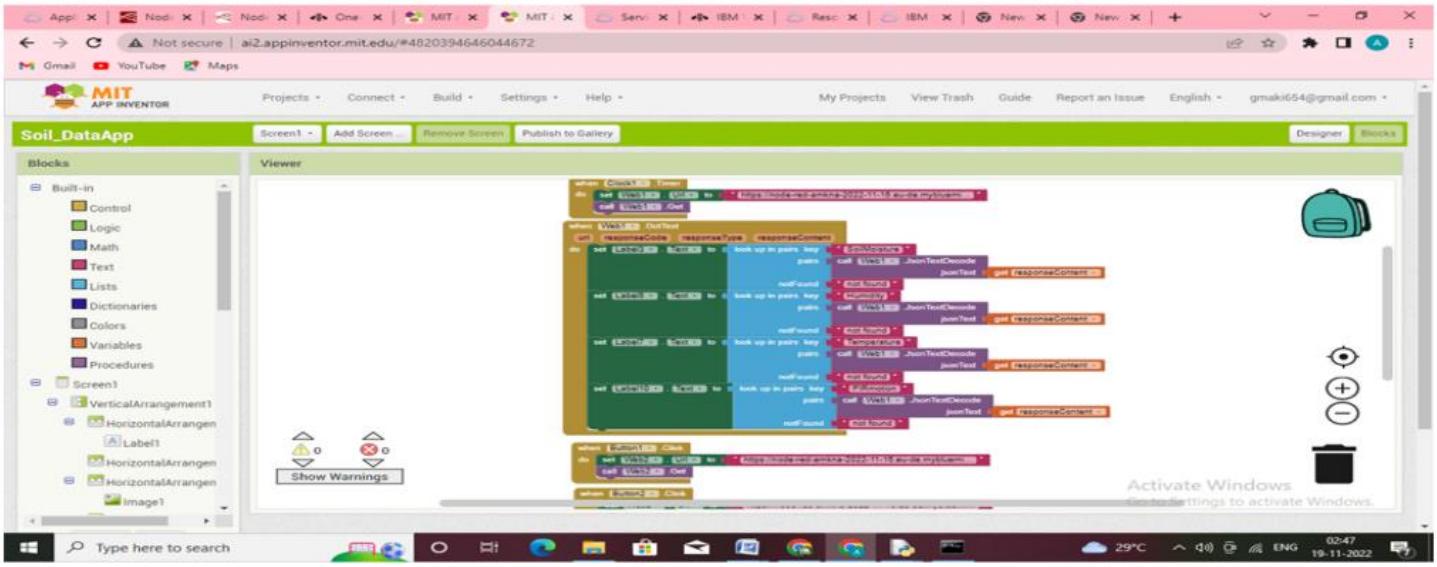





Node-RED
Flow 1
Soil Moisture
Humidity
Temperature
light on
light off
motor on
motor off
mit app
msg.payload
http
function


SOIL DATA
Soil Moisture
Humidity
Temperature
CONTROL BUTTON
LIGHT ON
LIGHT OFF
MOTOR ON
MOTOR OFF
12
chart
iot-2/type/ESP32/id/12345/evt/status/fmt/json
ANIMAL IS DETECTED


SOIL DATA
Soil Moisture: NA
Humidity: NA
Temperature: NA
Animal Detect: NA
1 = Animal is Detected
0 = Animal is not Detected
CONTROL BUTTON
LIGHT ON
LIGHT OFF



6.2 Sprint delivery schedule

In the tabulation given below, tab. 8 speaks about various sprints that are planned by us for making possible to complete this project.

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	5 Days	20 Oct 2022	24 Oct 2022	20	21 Oct 2022
Sprint-2	20	5 Days	25 Oct 2022	29 Oct 2022	20	27 Oct 2022
Sprint-3	20	5 Days	31 Oct 2022	4 Nov 2022	20	2 Nov 2022
Sprint-4	20	7 Days	5 Nov 2022	11 Nov 2022	20	8 Nov 2022

Tab. 8 Project Tracker

Velocity:

We have a 23-day sprint duration, and the velocity of the team is 20 (points per sprint).

To Find: Calculate the team's average velocity (AV) per iteration unit (story points per day).

Solution:

$$AV = \frac{sprint\ duration}{velocity} = \frac{23}{20} = 1.15$$

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time. Fig. 11 depicts the burndown chart for our system.

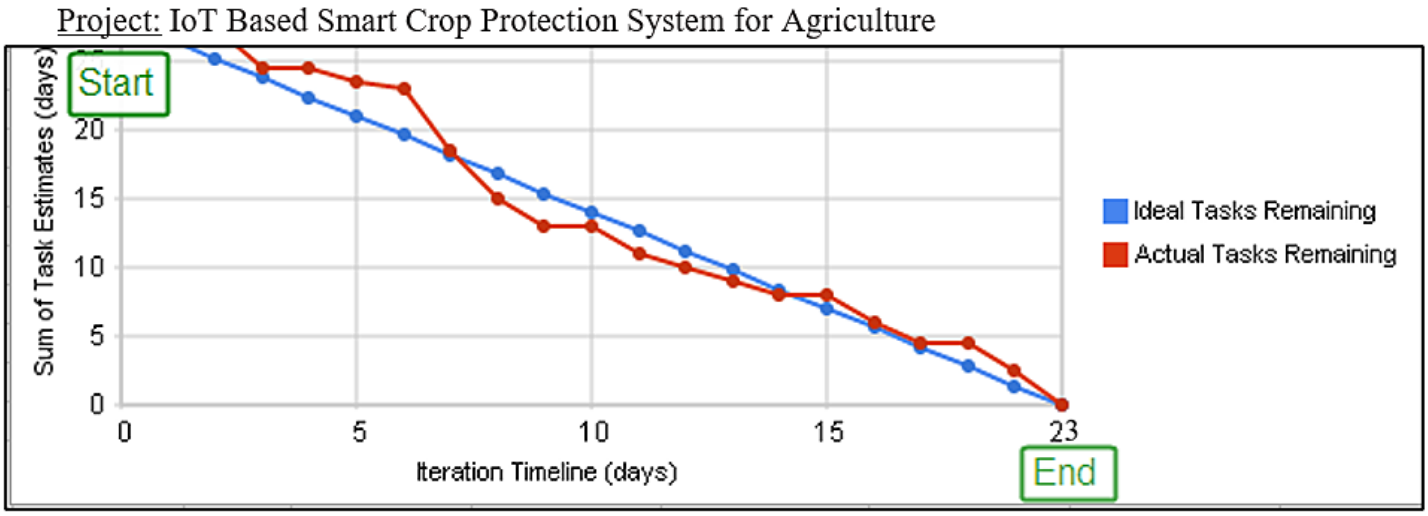
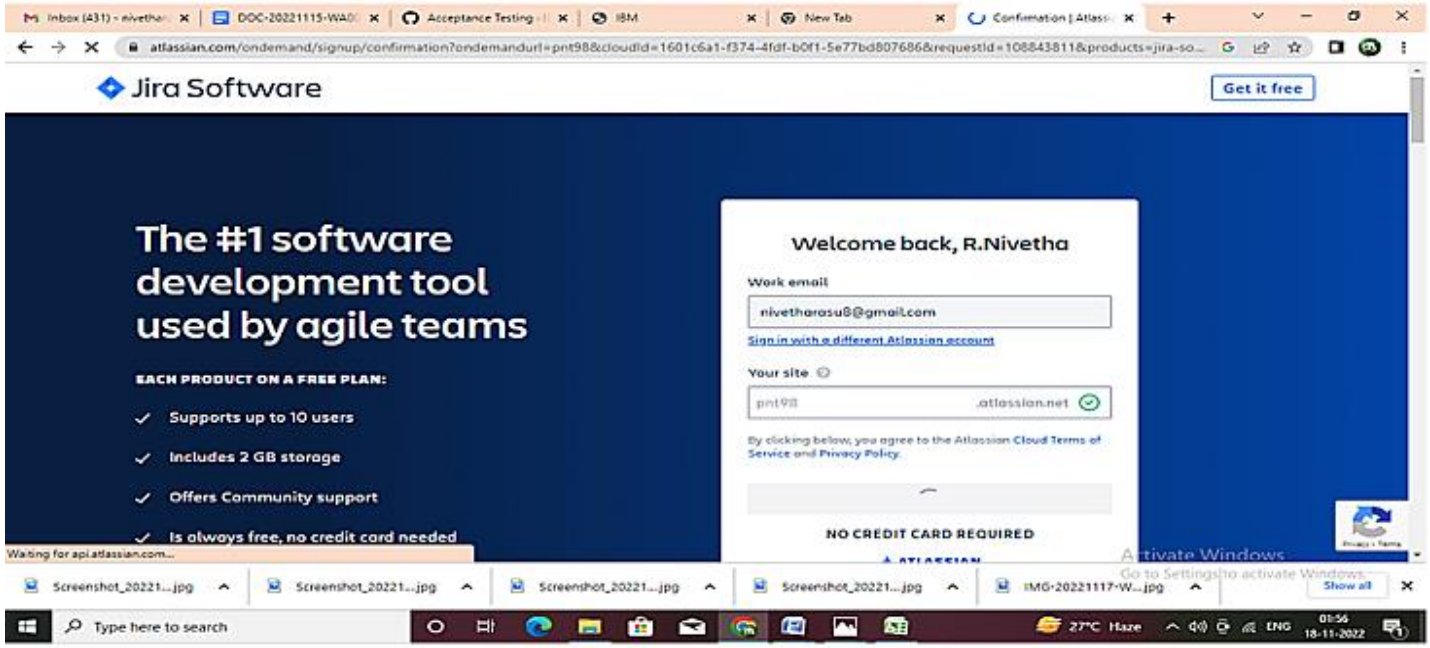
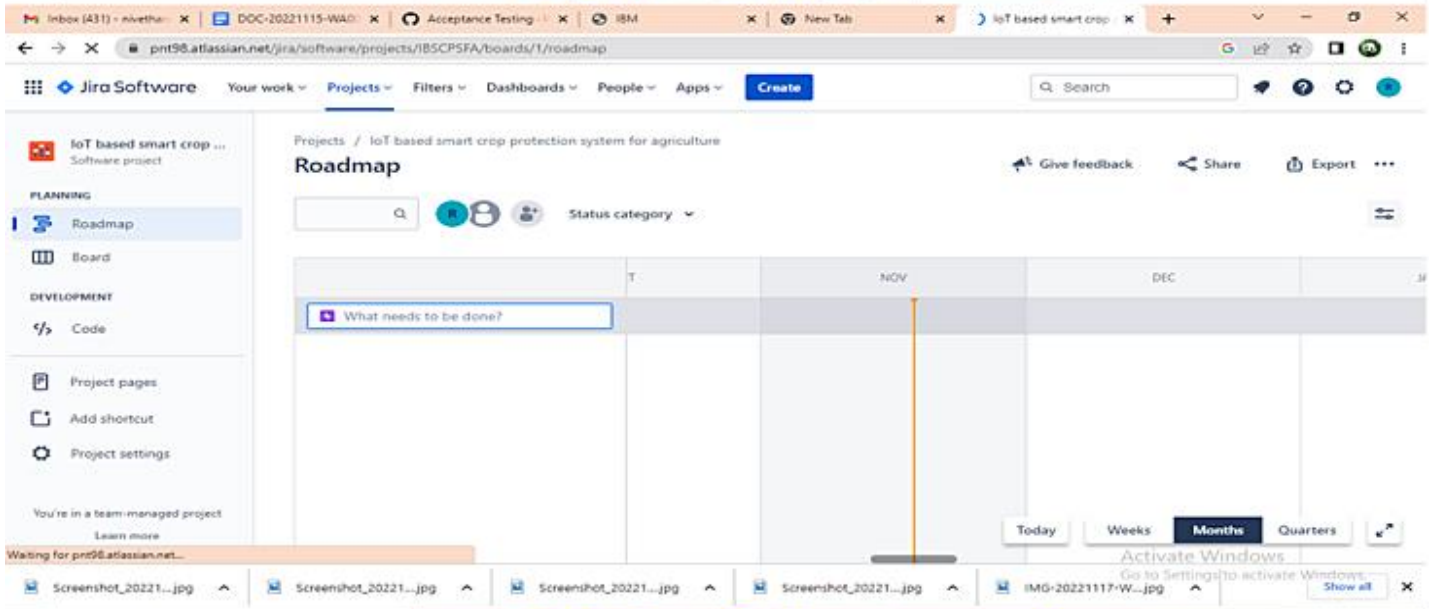


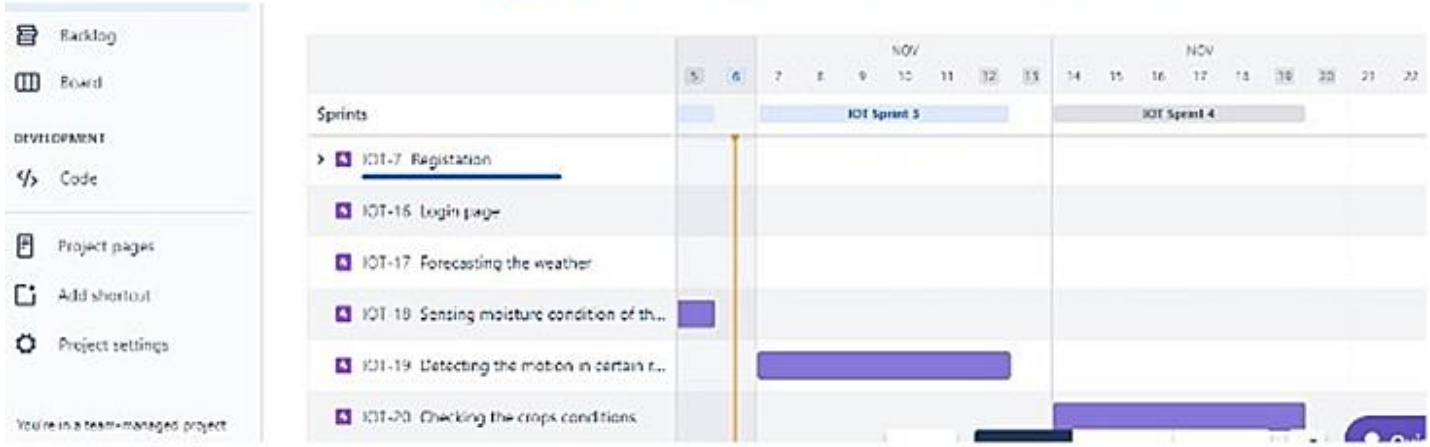
Fig. 11 Burndown Chart

6.3 Report from JIRA

JIRA is a very effective and easy to use tool for project management using agile methodologies. Each work item can be linked to a change set of the code delivered.


Jira Software
Get it free
The #1 software development tool used by agile teams
EACH PRODUCT ON A FREE PLAN:
Supports up to 10 users
Includes 2 GB storage
Offers Community support
Is always free, no credit card needed
Welcome back, R.Nivetha
Work email
nivetharosu8@gmail.com
Sign in with a different Atlassian account
Your site
pnt98
.atlassian.net
By clicking below, you agree to the Atlassian Cloud Terms of Service and Privacy Policy.
NO CREDIT CARD REQUIRED
Activate Windows
Go to Settings to activate Windows.
Type here to search
27°C Haze
ENG
18-11-2022


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
IoT based smart crop ...
Software project
PLANNING
Roadmap
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / IoT based smart crop protection system for agriculture
Roadmap
Give feedback
Share
Export
Status category
What needs to be done?
NOV
DEC
Today
Weeks
Months
Quarters
Activate Windows
Go to Settings to activate Windows.
Type here to search
27°C Haze
ENG
18-11-2022


Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
Sprints
IOT Sprint 5
IOT Sprint 4
IOT-7 Registation
IOT-16 Login page
IOT-17 Forecasting the weather
IOT-18 Sensing moisture condition of th...
IOT-19 Detecting the motion in certain r...
IOT-20 Checking the crops conditions

Customer Journey Map

In the image below which is numbered fig. 12, shares much information on customer journey map. As whatever product we are designing its the public and the users who are going to employ this system and make use and give feedback for us to make a better advancement in technology.

Fig. 12 Customer Journey map for this system.

7. Coding and Solution

7.1 Feature-1

The moisture contents in the soil sensed by using the moisture sensor and it will identify the amount of water supplied required to the crop and sends data to ARM cortex and enables sensor to supply water which automatically turn on the water source and turn off it when need is satisfied.

soilpython.py - C:/python/Python37/soilpython.py (3.7.4)

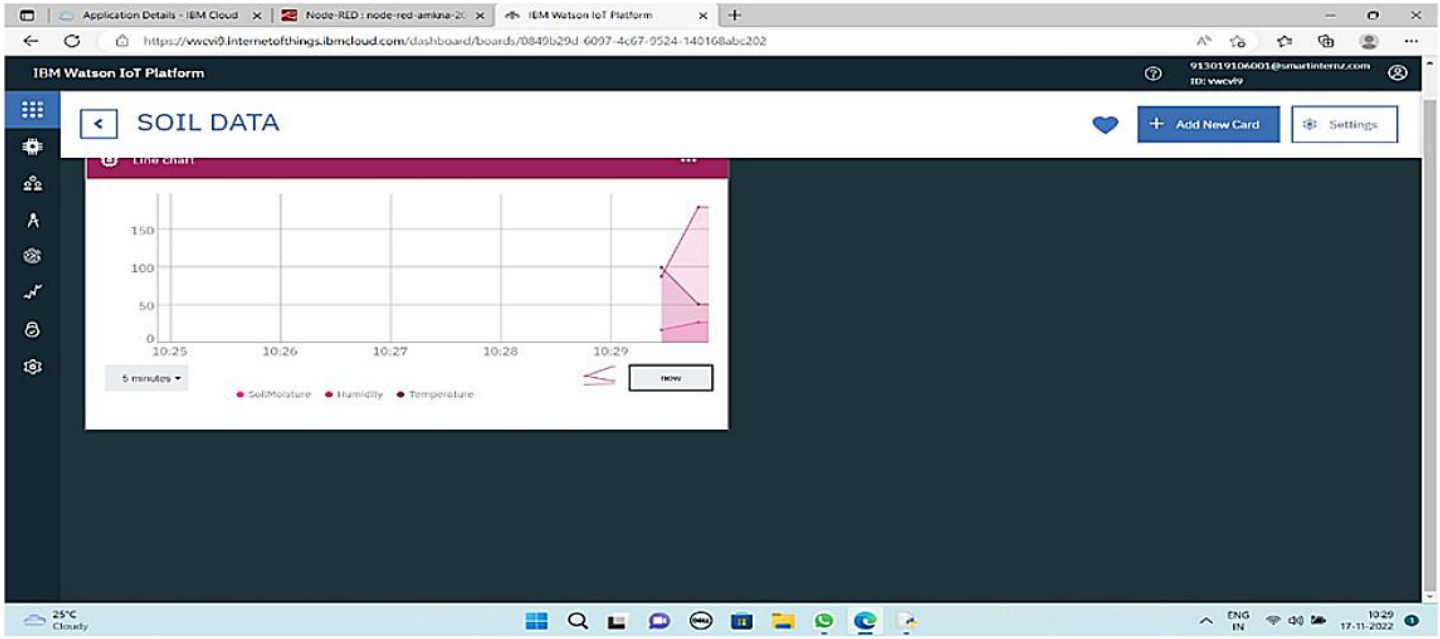
File Edit Format Run Options Window Help

```
import json
import wiotp.sdk.device
import time
import random
import ibmiotf.application
import ibmiotf.device

myConfig = {
    "identity": {
        "orgId": "vwcvi9",
        "typeId": "ESP32",
        "deviceId":"12345"
    },
    "auth": {
        "token": "12345678"
    }
}
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("light is on")
    elif status == "lightoff":
        print ("light is off")
    elif status == "motoron":
        print ("motor is on")
    elif status == "motoroff":
        print ("motor is off")
    else :
        print ("please send proper command")

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    soil=random.randint(0,30)
    humid=random.randint(0,200)
    temp=random.randint(1,100)
    pir=random.randint(0,1)
    myData={'SoilMoisture':soil,'Humidity':humid,'Temperature':temp,'PIRmotion':pir}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
```

soilpython.py - C:/python/Python37/soilpython.py (3.7.4)

File Edit Format Run Options Window Help

```
myConfig = {
    "identity": {
        "orgId": "vwcvi9",
        "typeId": "ESP32",
        "deviceId":"12345"
    },
    "auth": {
        "token": "12345678"
    }
}
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("light is on")
    elif status == "lightoff":
        print ("light is off")
    elif status == "motoron":
        print ("motor is on")
    elif status == "motoroff":
        print ("motor is off")
    else :
        print ("please send proper command")

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    soil=random.randint(0,30)
    humid=random.randint(0,200)
    temp=random.randint(1,100)
    pir=random.randint(0,1)
    myData={'SoilMoisture':soil,'Humidity':humid,'Temperature':temp,'PIRmotion':pir}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: ", myData)
    time.sleep(20)

    client.commandCallback = myCommandCallback

client.disconnect()
```



To publish and access sensor parameters like Temperature, Humidity, and Soil Moisture to the IBM IoT platform.

Solution C++ Coding:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(8,9,10,11,12,13);//rs,en,data pins d4 -d7

float TEMP;

int MOISURE,HUM;

const int buzzer=6;

const int motor=7;

int led=4,temp=0,i=0;

char str[30];

int aa=0, dt=0////temp;

int dh=0;///hum

int s1=0,s3=0;

void setup()  {

lcd.begin(16,2);

Serial.begin(9600);

pinMode(buzzer, OUTPUT);

pinMode(motor, OUTPUT);

digitalWrite(buzzer, LOW);

digitalWrite(motor, LOW);

lcd.clear();

lcd.setCursor(0,0);lcd.print("IOT Based ");

lcd.setCursor(0,1);lcd.print("Agriculture Crop");
```

```
delay(5000);lcd.clear();
lcd.setCursor(0,0);lcd.print("Field Monitoring");
lcd.setCursor(0,1);lcd.print("Irrigation");
delay(5000);lcd.clear();
lcd.setCursor(0,0);lcd.print(" Automation");
lcd.setCursor(0,1);lcd.print("using GPRS ");
delay(5000);lcd.clear();
gsm_init();lcd.clear();
digitalWrite(buzzer, LOW);
digitalWrite(motor, LOW);
}
void loop() {
aa=aa+1;
digitalWrite(buzzer, LOW);
lcd.clear();
TEMP = analogRead(0);
TEMP=(TEMP*500)/1023;
lcd.setCursor(0,0);lcd.print("T:");lcd.setCursor(3,0);lcd.print(TEMP);delay(200);if(TEMP<50){dt=0;delay(100);}
if(TEMP>50){
dt=dt+1;
if(dt==2){
delay(200);digitalWrite(buzzer, HIGH);delay(200);digitalWrite(buzzer, LOW);
delay(200);digitalWrite(buzzer, HIGH);delay(200);digitalWrite(buzzer, LOW);lcd.clear();send_gprs();delay(500);
delay(200);digitalWrite(buzzer, HIGH);delay(200);digitalWrite(buzzer, LOW);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS");lcd.setCursor(0,1);lcd.print("TEMP ALERT");
Serial.println("AT+CMGF=1");delay(400);
Serial.println("AT+CMGS=\"9148300815\"");delay(400);
Serial.println("Over Temperature\n");delay(100);
Serial.print("Temp=");delay(100);Serial.print(TEMP);delay(500);Serial.write(26);delay(500);
Serial.print("AT\r\n");delay(1000);Serial.print("AT\r\n");delay(1000);Serial.println("AT+CMGF=1");delay(1000);
}}
HUM= analogRead(1);HUM =HUM/2;
HUM=HUM+12;
lcd.setCursor(0,1);lcd.print("H:");lcd.setCursor(3,1);lcd.print(HUM);delay(2500);if(HUM<30){dh=0;delay(100);}
if(HUM>45){
dh=dh+1;
if(dh==2){
delay(200);digitalWrite(buzzer, HIGH);delay(200);digitalWrite(buzzer, LOW);
```

```
delay(200);digitalWrite(buzzer, HIGH);delay(200);digitalWrite(buzzer, LOW);lcd.clear();send_gprs();delay(500);
delay(200);digitalWrite(buzzer, HIGH);delay(200);digitalWrite(buzzer, LOW);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS");lcd.setCursor(0,1);lcd.print("HUMIDITY ALERT");
Serial.println("AT+CMGF=1");delay(400);
Serial.println("AT+CMGS=\"9148300815\"");delay(400);
Serial.println("HUMIDITY ALERT\n");delay(100);
Serial.print("HUM=");delay(100);Serial.print(HUM);delay(500);Serial.write(26);delay(500);
Serial.print("AT\r\n");delay(1000);Serial.print("AT\r\n");delay(1000);Serial.println("AT+CMGF=1");delay(1000);
}}
MOISURE = analogRead(2); MOISURE= MOISURE/4;MOISURE=256-MOISURE;
lcd.setCursor(9,0);
lcd.print("M: ");
lcd.setCursor(11,0);
lcd.print(MOISURE);
delay(500);
if(MOISURE<100){
s1=s1+1;
if(s1==5){
s3=0;
delay(200);digitalWrite(buzzer, HIGH);delay(200);digitalWrite(buzzer, LOW);
delay(200);digitalWrite(buzzer, HIGH);delay(200);digitalWrite(buzzer, LOW);lcd.clear();send_gprs();delay(500);
delay(200);digitalWrite(buzzer, HIGH);delay(200);digitalWrite(buzzer, LOW);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS");delay(2000);
lcd.clear();
lcd.setCursor(0,0);lcd.print("FEILD AT DRY");
lcd.setCursor(0,1);lcd.print("MOTOR ON");delay(2000);
digitalWrite(motor, HIGH);
Serial.print("AT\r\n");delay(2000);Serial.print("AT\r\n");delay(2000);
Serial.println("AT+CMGF=1");delay(400);Serial.println("AT+CMGS=\"9148300815\"");delay(400);   // use your 10 digit cell no. here
Serial.println("FEILD AT DRY\n");delay(100);
Serial.println("MOTOR ON\n");delay(100);
Serial.write(26);delay(100);
Serial.print("AT\r\n");delay(1000);Serial.print("AT\r\n");delay(1000);Serial.println("AT+CMGF=1");delay(1000);
motor_gprs();
delay(2000);
pump_gprs();
delay(2000);
}
```

```
delay(100);
}
if(MOISURE>200){
s3=s3+1;
if(s3==5){
s1=0;
delay(200);digitalWrite(buzzer, HIGH);delay(200);digitalWrite(buzzer, LOW);
delay(200);digitalWrite(buzzer, HIGH);delay(200);digitalWrite(buzzer, LOW);lcd.clear();send_gprs();delay(500);
delay(200);digitalWrite(buzzer, HIGH);delay(200);digitalWrite(buzzer, LOW);
lcd.clear();lcd.setCursor(0,0);lcd.print("SENDING SMS");delay(2000);
lcd.clear();
lcd.setCursor(0,0);lcd.print("FEILD AT WET");
lcd.setCursor(0,1);lcd.print("MOTOR OFF");delay(2000);digitalWrite(motor, LOW);
Serial.print("AT\r\n");delay(2000);Serial.print("AT\r\n");delay(2000);
Serial.println("AT+CMGF=1");delay(400);Serial.println("AT+CMGS=\"9148300815\"");delay(400);   // use your 10 digit cell no. here
Serial.println("FEILD AT WET\n");delay(100);
Serial.println("MOTOR OFF\n");delay(100);
Serial.write(26);delay(100);
Serial.print("AT\r\n");delay(1000);Serial.print("AT\r\n");delay(1000);Serial.println("AT+CMGF=1");delay(1000);
motor_gprs();
delay(2000);
pump_gprs();
delay(2000);
}
delay(100);
}
if(aa==10){
lcd.clear();
send_gprs();
delay(2000);
pump_gprs();
delay(2000);
aa=0;
}}
void motor_gprs(){
lcd.clear();lcd.print("GPRS SENDING");
boolean test47_flag=1;
while(test47_flag){Serial.print("AT+HTTPPARA=\"URL\",\"http://iotbabycare.com/iot_green_agrimnr/put_motor.php?mot=motor");
```

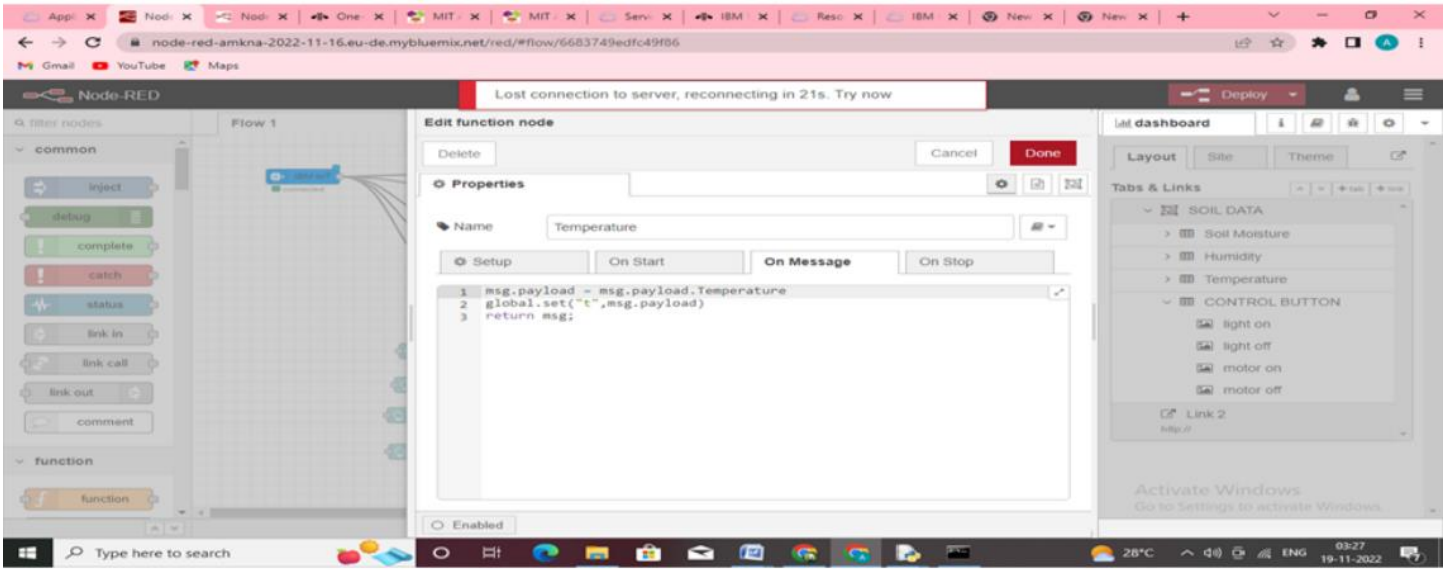
```
Serial.print("\"");Serial.print("\r\n");

while(Serial.available()>0){if(Serial.find("OK"))test47_flag=0;}delay(1000);}

lcd.clear();lcd.print("SENT COMPLETED");delay(10000);lcd.clear()

lcd.clear();lcd.print("ACTION");

boolean test48_flag=1;while(test48_flag){Serial.print("AT+HTTPACTION=0\r\n");

while(Serial.available()>0){if(Serial.find("OK"))test48_flag=0;}delay(1000);}

lcd.clear();lcd.print("SEND  OK");delay(2000); delay(2000);delay(2000);

boolean at_flagd=1;while(at_flagd){Serial.println("AT");while(Serial.available()>0){if(Serial.find("OK"))at_flagd=0;}delay(1000);}

}

void send_gprs(){

lcd.clear();lcd.print("GPRS SENDING");

boolean test7_flag=1;

while(test7_flag){Serial.print("AT+HTTPPARA=\"URL\",\"http://iotbabycare.com/iot_green_agrimnr/put_data.php");

Serial.print("?temp=");Serial.print(TEMP);

Serial.print("&hum=");Serial.print(HUM);

Serial.print("&mos=");Serial.print(MOISURE);

Serial.print("\"");Serial.print("\r\n");

while(Serial.available()>0){if(Serial.find("OK"))test7_flag=0;}delay(1000);}

lcd.clear();lcd.print("SENT COMPLETED");delay(10000);lcd.clear();

lcd.clear();lcd.print("ACTION");

boolean test8_flag=1;while(test8_flag){Serial.print("AT+HTTPACTION=0\r\n");

while(Serial.available()>0){if(Serial.find("OK"))test8_flag=0;}delay(1000);}

lcd.clear();lcd.print("SEND  OK");delay(2000); delay(2000);delay(2000);

}

void pump_gprs(){

boolean at_flagd=1;while(at_flagd){Serial.println("AT");while(Serial.available()>0){if(Serial.find("OK"))at_flagd=0;}delay(1000);}

lcd.clear();lcd.print("PING TO WEBSITE");

boolean
test17_flag=1;while(test17_flag){Serial.print("AT+HTTPPARA=\"URL\",\"http://iotbabycare.com/iot_green_agrimnr/get_data.php");Serial.print("\"
");Serial.print("\r\n");

while(Serial.available()>0){if(Serial.find("OK"))test17_flag=0;}delay(1000);}

lcd.clear();lcd.print("WEBLINK SUCESS");delay(1000);

lcd.clear();lcd.print("HTTP ACTION");

boolean test18_flag=1;while(test18_flag){Serial.print("AT+HTTPACTION=0\r\n");

while(Serial.available()>0){if(Serial.find("OK"))test18_flag=0;}delay(1000);}

lcd.clear();lcd.print("ACTION COMPLETED");delay(5000);

lcd.clear();lcd.print("GET THE DATA");

boolean test19_flag=1;while(test19_flag){Serial.print("AT+HTTPREAD\r\n");
```

```
SeriallEvent();
while(Serial.available()>0){if(Serial.find("OK"))test19_flag=0;}delay(1000);}
lcd.clear();lcd.print("DATA OK");delay(5000);
if(temp==1){
check();
temp=0;
i=0;
delay(100);
}}
void SeriallEvent() {
while(Serial.available()) {
if(Serial.find("?ID=")){
digitalWrite(led, HIGH);
delay(100);
digitalWrite(led, LOW);
while (Serial.available()) {
char inChar=Serial.read();
str[i++]=inChar;
if(inChar=='$'){
temp=1;
return;
} } }}
}
void check(){
if(!(strncmp(str,"ON",2))){lcd.setCursor(1,1);lcd.print("MOTOR  ON");delay(500);digitalWrite(motor,HIGH);}
else if(!(strncmp(str,"OFF",3))){lcd.setCursor(1,1);lcd.print("MOTOR OFF");digitalWrite(motor,LOW);}
}
void gsm_init(){
lcd.clear();lcd.print("GSM TESTING..");
boolean at_flag=1;while(at_flag){Serial.println("AT");while(Serial.available()>0){if(Serial.find("OK"))at_flag=0;}delay(1000);}
lcd.clear();lcd.print("GSM CONNECTED");delay(1000);lcd.clear();
lcd.print("ECHO");
boolean echo_flag=1;
while(echo_flag)
{Serial.println("ATE0"); while(Serial.available()>0){if(Serial.find("OK"))echo_flag=0;}delay(1000);}
lcd.clear();  lcd.print("Echo OFF");delay(1000);lcd.clear();
lcd.print("Finding Network..");
boolean net_flag=1;while(net_flag){Serial.println("AT+CPIN?");
```

```
while(Serial.available()>0){if(Serial.find("+CPIN: READY"))net_flag=0;}delay(1000);}
lcd.clear();lcd.print("Network Found..");
lcd.setCursor(0,1);lcd.print("GSM NETWORK OK");delay(2000);lcd.clear();
lcd.clear();lcd.print("TEST MESS");
boolean test_flag=1;while(test_flag){Serial.println("AT+CMGF=1");
while(Serial.available()>0){if(Serial.find("OK"))test_flag=0;}delay(1000);}
lcd.clear();lcd.print("TEST MESSAGE");delay(1000);
lcd.clear();lcd.print("AT+CGATT");
boolean test1_flag=1;while(test1_flag){Serial.println("AT+CGATT=1");
while(Serial.available()>0){if(Serial.find("OK"))test1_flag=0;}delay(1000);}
lcd.clear();lcd.print("AT+CGATT=1");delay(1000);
lcd.clear();lcd.print("GPRS START");
boolean test2_flag=1;while(test2_flag){Serial.print("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\"\r\n");
while(Serial.available()>0){if(Serial.find("OK"))test2_flag=0;}delay(1000);}
lcd.clear();lcd.print("GPRS START1");delay(1000);
lcd.clear();lcd.print("GPRS START");
boolean test3_flag=1;while(test3_flag){Serial.print("AT+SAPBR=3,1,\"APN\",\"internet\"\r\n");
while(Serial.available()>0){if(Serial.find("OK"))test3_flag=0;}delay(1000);}
lcd.clear();lcd.print("GPRS START2");delay(1000);
lcd.clear();lcd.print("GPRS MAIN");
boolean test4_flag=1;while(test4_flag){Serial.print("AT+SAPBR=1,1\r\n");
while(Serial.available()>0){if(Serial.find("OK"))test4_flag=0;}delay(1000);}
lcd.clear();lcd.print("GPRS FIND");delay(1000);
lcd.clear();lcd.print("HTTP STARTS");
boolean test5_flag=1;while(test5_flag){Serial.print("AT+HTTPINIT\r\n");
while(Serial.available()>0){if(Serial.find("OK"))test5_flag=0;}delay(1000);}
lcd.clear();lcd.print("HTTP STARTS1");delay(1000);
lcd.clear();lcd.print("HTTP STARTS");
boolean test6_flag=1;while(test6_flag){Serial.print("AT+HTTPPARA=\"CID\",1\r\n");
while(Serial.available()>0){if(Serial.find("OK"))test6_flag=0;}delay(1000);}
lcd.clear();lcd.print("HTTP STARTS2");delay(1000);}
```

7.2 Feature-2

PIR sensor used to detect whether a human has moved in or out of the sensors range. After processing the available information, if the human is not found the system raised the buzzer sound, to alert people about intrusion.



Display the image and pre-process the level of the Node-RED web UI and display the temperature, humidity, and soil moisture levels. Integrate the buttons in the UI to control the Motors.

Solution Coding C++(Approach):

```
#include <ESP8266WiFi.h>

const char *wifissid = "SSID";

const char *wifipass = "Password";

void setup() {

// put your setup code here, to run once:

Serial.begin(115200);

}

void loop() {

// put your main code here, to run repeatedly:

delay(1000);

reconnectWiFi();

}

void reconnectWiFi(){

 WiFi.mode(WIFI_STA);

delay(200);

WiFi.begin(wifissid,wifipass);

                 while(WiFi.status()!= WL_CONNECTED){

                Serial.print(".");

                delay(500);

}

Serial.println("Connected to: \t");

Serial.println(WiFi.localIP());

#include <Wire.h>

#define Addr 0x40

Wire.beginTransmission(Addr);
```

```
// Send humidity measurement command, NO HOLD master
Wire.write(0xF5);
// Stop I2C transmission
Wire.endTransmission();
delay(500);
// Request 2 bytes of data
Wire.requestFrom(Addr, 2);
// Read 2 bytes of data
// humidity msb, humidity lsb
if(Wire.available() == 2)
{
  data[0] = Wire.read();
  data[1] = Wire.read();
}
float humidity = (((data[0] * 256.0 + data[1]) * 125.0) / 65536.0) - 6;
float cTemp = (((data[0] * 256.0 + data[1]) * 175.72) / 65536.0) - 46.85;
float fTemp = (cTemp * 1.8) + 32;
void reconnect()
{
// Loop until we're reconnected
while (!client.connected()) {
Serial.print("Attempting MQTT connection...");
if (client.connect("ESP8266Client")) {
Serial.println("connected");
}
else {
Serial.print("failed, rc=");
Serial.print(client.state());
Serial.println(" try again in 5 seconds");
// Wait 5 seconds before retrying
delay(5000);
    }
  }
}
void loop()
{


if (!client.connected()) {
reconnect();
}
```

client.publish(Topic to publish, Payload(message to publish), Return value (true or false));

client.loop();

}

Console Output (Based on Approach):

Fig. 12 Output in console

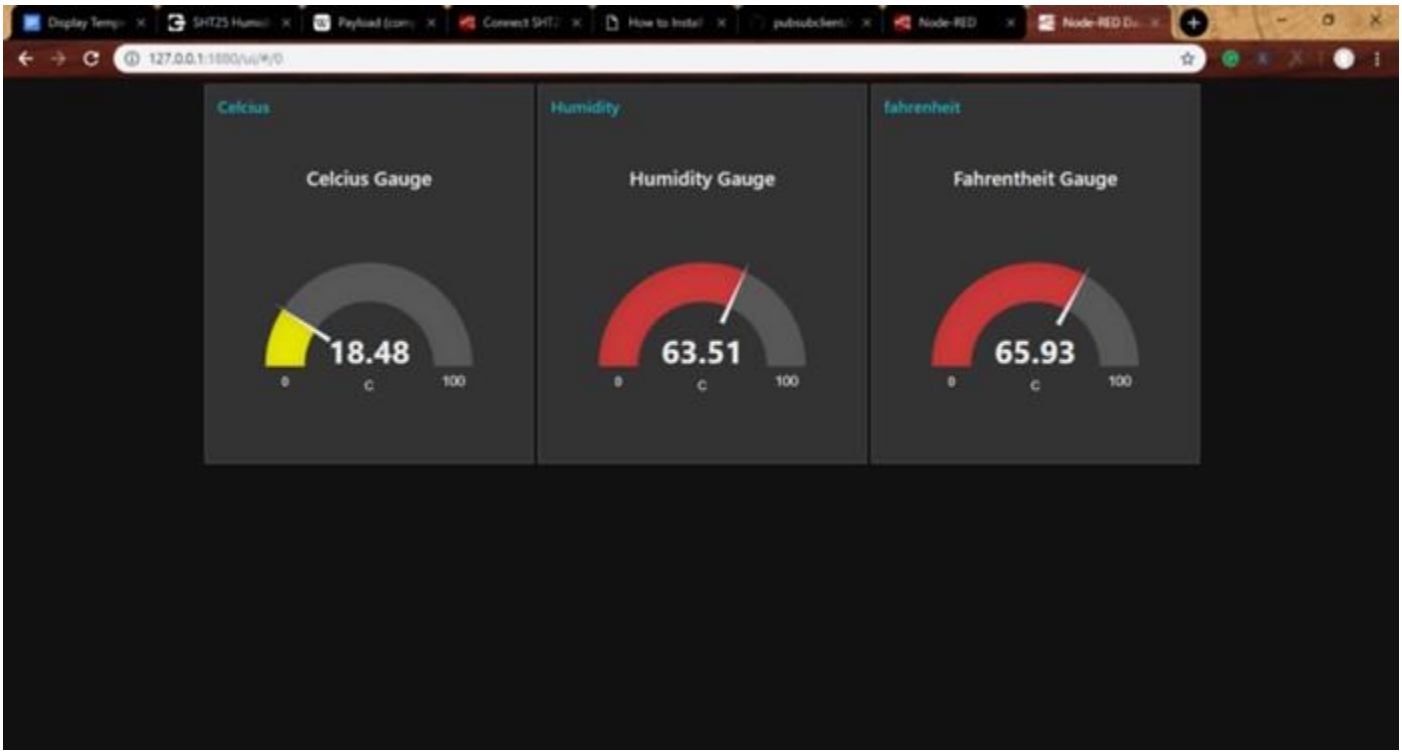


Fig. 13 Parameters in console

8. Testing

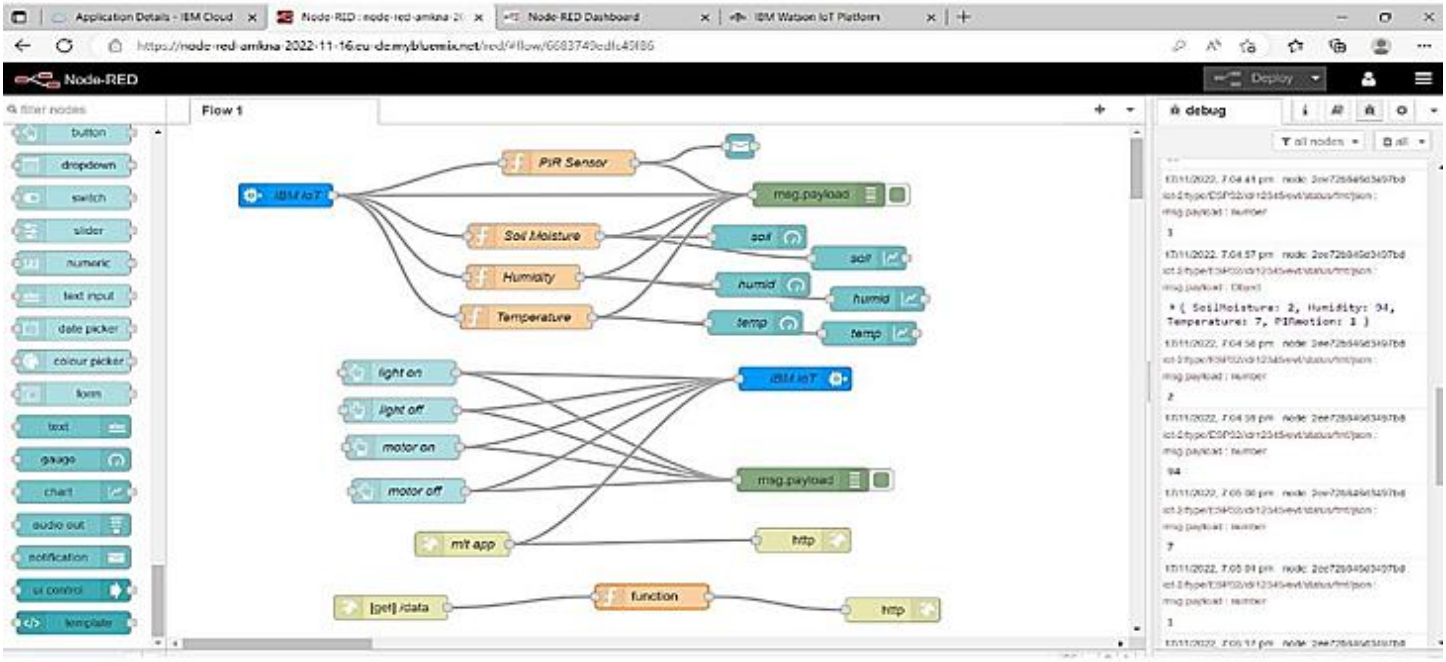
8.1 Test Cases

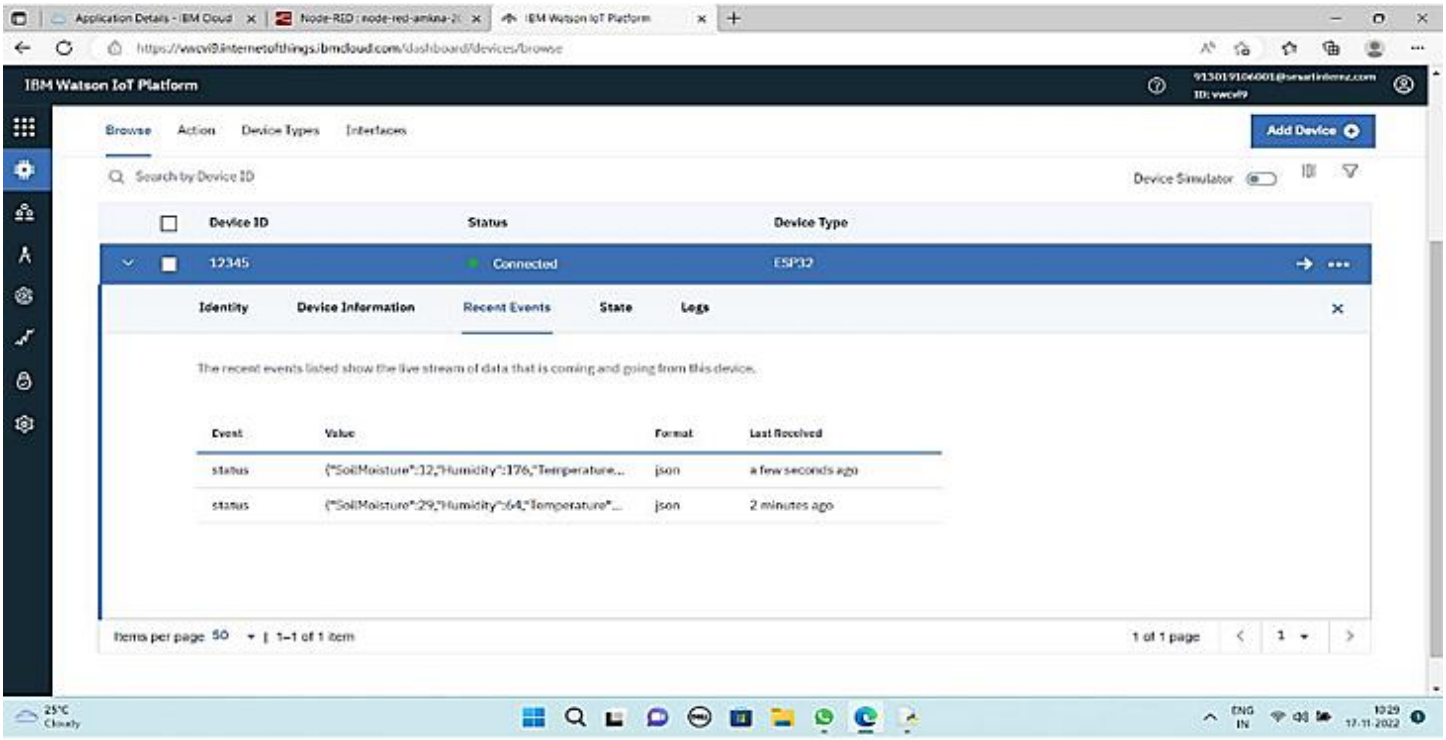
A test cases is a specification of the inputs, execution conditions, test and procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	6	0	0	6
Client Application	48	0	0	48
Security	3	0	0	3
Outsource Shipping	2	0	0	2
Exception Reporting	8	0	0	8
Final Report Output	5	0	0	5

8.2 User Acceptance Testing

User Acceptance Testing(UAT) also called application testing or end-user testing, is a phase of software development in which the software is tested in the real world by it’s intended audience





9. Results

9.1 Performance Metrics

Performance testing comes under quality assurance checks of software / application in which speed, capacity and stability are the major checks.

NFT - Risk Assessment

S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Voluem Changes	Risk Score	Justification
1	IoT based smart cro	Existing	Moderate	No Changes	Low	Low	No Changes	ORANGE	It is cost effective due to no hardware changes

NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach	mptions/Dependencies/R	Approvals/SignOff
1	monitoring crops by using senson	load	I need development team su	Approved

End Of Test Report

S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
1	monitoring crops by	The farmers feel happ	we need good network	This test usually simulates the curr	GO decision because t	Increase db size and low mem	Bulk load	Approved

Real time monitoring of the field parameters

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `iotbabyc_agriiot`
--

-- --------------------------------------------------------

--
-- Table structure for table `sens`
--

CREATE TABLE `sens` (
  `id` bigint(30) NOT NULL,
  `temp` varchar(120) NOT NULL,
  `hum` varchar(120) NOT NULL,
  `mos` varchar(120) NOT NULL,
  `createdat` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `sens`
--

INSERT INTO `sens` (`id`, `temp`, `hum`, `mos`, `createdat`) VALUES
(1, '30', '40', '120', '2022-09-28 11:38:06'),
(2, '22.48', '35', '1', '2022-09-29 05:32:55'),
(3, '24.44', '35', '1', '2022-09-29 05:33:28'),
(4, '32.26', '35', '1', '2022-09-29 05:34:56'),
(5, '55.72', '35', '1', '2022-09-29 05:35:41'),
(6, '34.70', '34', '1', '2022-09-29 05:36:32'),
```

Obtained after getting 10 counts from the controller and the database of SQL

```
(30, '21.99', '33', '1', '2022-09-29 07:41:57'),
(31, '63.05', '34', '1', '2022-09-29 07:43:24'),
(32, '63.05', '35', '1', '2022-09-29 07:43:51'),
(33, '30.79', '35', '220', '2022-09-29 07:45:05'),
(34, '30.79', '35', '220', '2022-09-29 07:45:37'),
(35, '30.79', '35', '220', '2022-09-29 07:46:18'),
(36, '22.97', '35', '109', '2022-09-29 07:47:32'),
(37, '25.90', '52', '123', '2022-09-29 07:48:32'),
(38, '24.44', '35', '122', '2022-09-29 07:49:10'),
(39, '31.28', '33', '115', '2022-09-29 07:50:23'),
(40, '31.28', '32', '137', '2022-09-29 07:51:36'),
(41, '35.68', '29', '1', '2022-09-29 09:08:39');

--
-- Indexes for dumped tables
--

--
-- Indexes for table `sens`
--
ALTER TABLE `sens`
  ADD PRIMARY KEY (`id`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `sens`
--
ALTER TABLE `sens`
  MODIFY `id` bigint(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=42;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
```

Running output obtained from phpMyAdmin SQL:

```
-- phpMyAdmin SQL Dump
-- version 4.9.5
-- https://www.phpmyadmin.net/
--
-- Host: localhost:3306
-- Generation Time: Sep 29, 2020 at 04:46 PM
-- Server version: 10.3.24-MariaDB-cll-lve
-- PHP Version: 7.3.6

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `iotbabyc_agriiot`
--

-- --------------------------------------------------------

--
-- Table structure for table `bulbs_data`
--

CREATE TABLE `bulbs_data` (
  `id` bigint(30) NOT NULL,
  `motor` varchar(120) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `bulbs_data`
--

INSERT INTO `bulbs_data` (`id`, `motor`) VALUES
(1, 'ON(Manual)');
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

10. Advantages and Disadvantages

Advantages

1. They are easy to operate and use and easy to maintain.
2. Sensors are cheaper in price and best in quality.
3. Automating processes in planting, treatment and harvesting can reduce consumption, human error and overall cost.
4. It is cost effective method.
5. It delivers high quality crop production.

Disadvantages

1. 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 over.
2. 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

It is now possible to employ sophisticated irrigation systems to properly regulate the water used on water fields to water crops. By incorporating PIR sensors and their values, we can also safeguard the farm from predators and animals. Additionally, it has made it possible to remotely control and monitor the irrigation systems. The system communicates with and processes the newest technologies via the internet via the Internet of Things. The level of moisture and humidity on the farm is alerted to by an Android application connected to the sensor data interface, allowing the farmer to further manage the water supply farm based on the level of moisture, humidity, and temperature provided by the application. Crops need to be constantly watched since they are extremely susceptible to over- or under-water tricks. Many resources are saved and our farmer brothers can receive a lot of assistance by implementing this cutting-edge system on our farm. Without a doubt, the method of indicating based on the illness that has been identified will help to prevent new farmers from being unable to plant seeds.

12. Future Scope

The number of challenges and limitations considers the most IoT-based devices for smart agriculture. The main focus is cost effectiveness in the IoT devices in the reduction of hardware and software cost with compromising precision system output. The standardization of the data format for the process will also provide improved device consistency and execution time.

13. APPENDIX

Source code

```
import json
import wiotp.sdk.device
import time
import random
import ibmiotf.application
import ibmiotf.device
myConfig = { "identity": { "orgId": "vwcvi9", "typeId": "ESP32", "deviceId":"12345" }, "auth": { "token": "12345678" } }
def myCommandCallback(cmd):
        print("Command received: %s" % cmd.data['command'])
        status=cmd.data['command']
        if status=="lighton":
                print ("light is on")
        elif status == "lightoff":
                print ("light is off")
```

```
        elif status == "motoron":
                print ("motor is on")
        elif status == "motoroff":
                print ("motor is off")
        else :
                print ("please send proper command")

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
while True:
        soil=random.randint(0,30)
        humid=random.randint(0,200)
        temp=random.randint(1,100)
        pir=random.randint(0,1)
myData={'SoilMoisture':soil,'Humidity':humid,'Temperature':temp,'PIRmotion ':pir}
          client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
            print("Published data Successfully: ", myData)
        time.sleep(20)
        client.commandCallback = myCommandCallback
client.disconnect()
```
