PROJECT REPORT

Project Name:
SmartFarmer - IoT Enabled Smart Farming Application

Team Id:
PNT2022TMID47947

Team Members:
B.Chitrabanu(Team Lead)
G.Priyanka
D.Shanmugapriya
N.D.Sharmila

INDEX

SMART FARMING

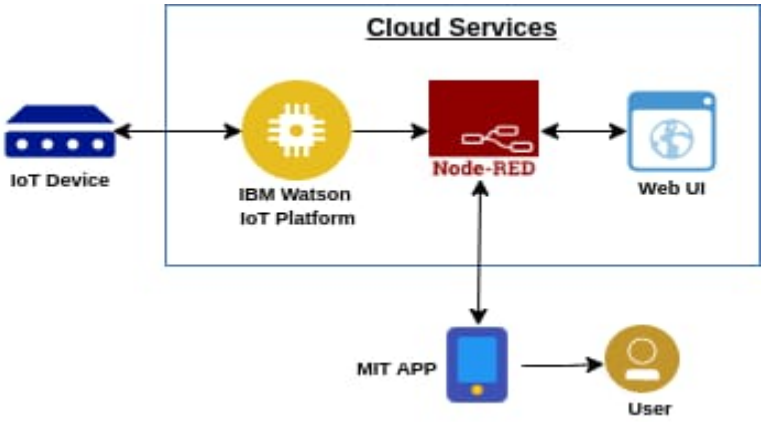
1.INTRODUCTION:

1.1 PROJECT OVERVIEW:

•IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, Temperature, humidity using some sensors.

•This is system that enables framers to monitor and control their forms with a web-based application build with Node-RED.

•It uses the IBM IOT Watson cloud platform as its Backend.



1.2 PURPOSE:

Smart Farming reduce the ecological foot print of farming. Minimized or Site Specific application of inputs, such as fertilizers and pesticides, in precision agriculture systems will mitigate leaching problems as well as the emission of greenhouse gases.

2.LITERATURE SURVEY:

2.1 EXISTING PROBLEM:

The biggest challenges faced by IoT in the agricultural sector are lack of information, high adoption costs, and security concers etc. Most of the farmers are not aware of the implementation of IoT in agriculture.

To successfully deploy a smart agriculture system, consider setting up a communications network that can integrate a limited number of sensors across a large area of farmland. This will require third-party network provisioning or setting up a private network consisting of access points and uplinks to a private backhaul network,

which channels all the data traffic to centralized monitoring software or an analytics head-end system • It is not a secure system.

•There is no motion detection for protection of agriculture field.

•Automation is not available.

2.2.REFERENCES:

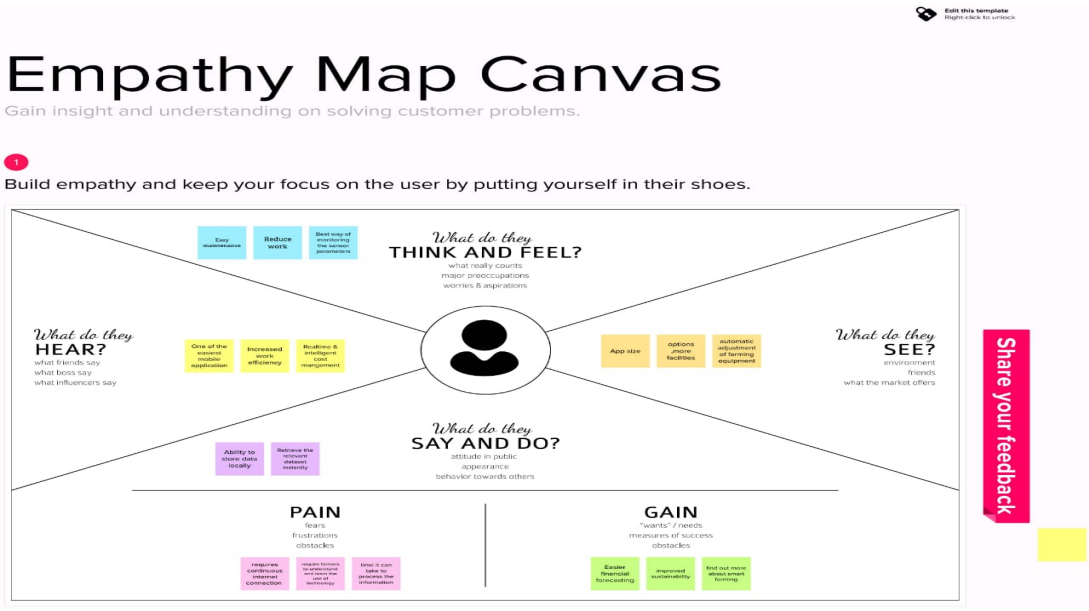
It is the application of modern ICT (Information and Communication Technologies) into agriculture. In IOT- based smart farming, a system is built for monitoring the crop field with the help of sensors (light, humidity, temperature, soil moisture, etc.). The farmers can monitor the field conditions from anywhere.

2.3.PROBLEM STATEMENT DEFINITION:

Overuse of pesticides and fertilizer in agricultural fields leads to destruction of the crop as well as reduces the efficiency of the field increasing the soil vulnerability toward pest. IoT applications may be used to update the farmer/user about type & quantity of pesticide required by the crop.

3.IDEATION & PROPOSED SOLUTION:

3.1.EMPATHY MAP CANVAS:



3.2.IDEATION AND BRAINSTORMING:

Introduction on Internet of Things (IoT), application of IoT in agricultural field to improve the yield and quality by reducing the cost is provided. The sensors which are used in the architecture are discussed briefly and the process of transmission of data from the agriculture field to the central system is explained. The proposed system advantages are included. In addition, open research issues, challenges, and future of IoT in agricultural field are highlighted. The concept is

basically developed on an idea, where there are numerous things or objects - such as Arduino, sensors, GSM models, LCD display, etc., that are connected with the Internet. Each of the objects has a different address and is able to interact with other items. The things or objects co-operate with each other to reach a common goal.

We are going to construct a smart agricultural monitoring system which can collect crucial agricultural data and send it to an IoT platform called Thing speak in real time where the data can be logged and analyzed. The logged data on Thing speak is in graphical format, a botanist or a reasonably knowledged farmer can analyze the data (from anywhere in the world) to make sensible changes in the supplied resources (to crops) to obtain high quality yield.

Smart agriculture monitoring system or simply smart farming is an emerging technology concept where data from several agricultural fields ranging from small to large scale and its surrounding are collected using smart electronic sensors. The collected data are analyzed by experts and local farmers to draw short term and long-term conclusion on weather pattern, soil fertility, current quality of crops, amount of water that will be required for next week to a month etc.

We can take smart farming a step further by automating several parts of farming, forexample smart irrigation and water management. We can apply predictive algorithms on microcontrollers or SoC to calculate the amount of water that will be required today for a particular agriculture field. Say, if there was rain yesterday and the quantity of water required today is going to be less. Similarly, if humidity was high the evaporation of water at upper ground level is going to be less, so water required will be less than normal, thus reducing water usage.

3.3.PROPOSED SOLUTION:

•To develop a Smart Agricultural System based on IOT which can give real time data and can help farmers in a very efficient manner.

•Soil Moisture can be checked by using the sensors that can sense the soil condition and send the data (moisture content in the soil) over the cloud services to the web

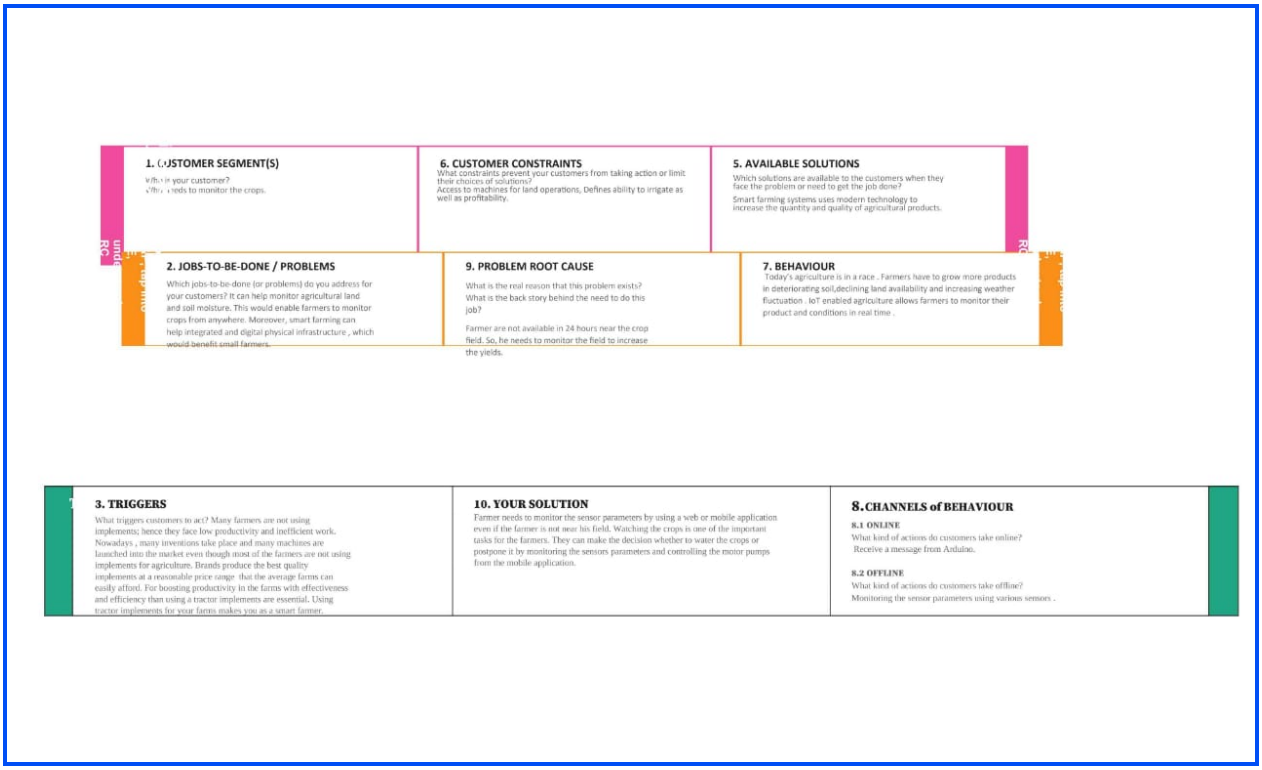
application.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Farmer needs to monitoring sensors parameters by using a web or mobile application even if the farmer is not near his field . Watering the crops is one of the important task for the farmers . They can make the decision whether to water the crops or postpone it by monitoring the sensors parameters and controlling the motor pumps from the mobile application.
2.	Idea / Solution description	Best way of monitoring the sensor parameters. One of the easiest mobile application. Automatic adjustment of farming equipment.
3.	Novelty / Uniqueness	Ability to store data locally. Retrieve the relevant dataset instantly. Find out more about smart farming
4.	Social Impact / Customer Satisfaction	Real time & intelligent cost mangement.Increased work efficiency.Easy maintenance
5.	Business Model (Revenue Model)	access to capital for service providers and removing financial risks, collective action across organisations and villages, alignment with governmental institutions and farmer cooperatives, and recognition that water is a valuable resource and should be treated as such. To develop investment funds for the implementation of the custom hire model that include both public and private funds
6.	Scalability of the Solution	Easier financial forecasting. Improved sustainability.

3.4.PROBLEM SOLUTION FIT:

•With the help of the IoT devices, you can know the real-time status of the crops by capturing the data from sensors.

•Using predictive analytics, you can get an insight to make better decisions related to harvesting.



4.REQUIREMENT ANALYSIS:

4.1.FUNCTIONAL REQUIREMENTS:

Following are the functional requirements of the proposed solution.

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User control	monitor and track weather conditions in their area and receive alerts about changes in weather conditions
FR-4	User tracking	view information about the current status of their crops receive alerts about changes in the status of their crops.
FR-5	Crop status	track and view information about their irrigation system receive alerts about changes in their irrigation system
FR-6	User login	The application shall provide authentication for users to login and access the application.

4.2.NON -FUNCTIONAL REQUIREMENTS:

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

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application should be easy to use and should have a user-friendly interface.
NFR-2	**Security**	The application shall provide a mechanism for farmers to securely receive data from the application.
NFR-3	**Reliability**	The application should be able to analyse the data and provide useful insights to the farmers.
NFR-4	**Performance**	The application should be able to predict the yield of crops and help farmers in taking necessary actions.
NFR-5	**Availability**	The application should be able to provide information about the weather, market prices, etc. to farmers.
NFR-6	**Scalability**	The application should be able to collect data from various sensors and devices installed on the farm.

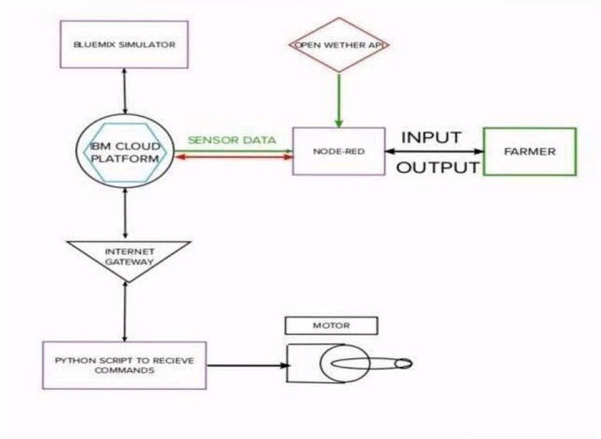
5.PROJECT DESIGN:

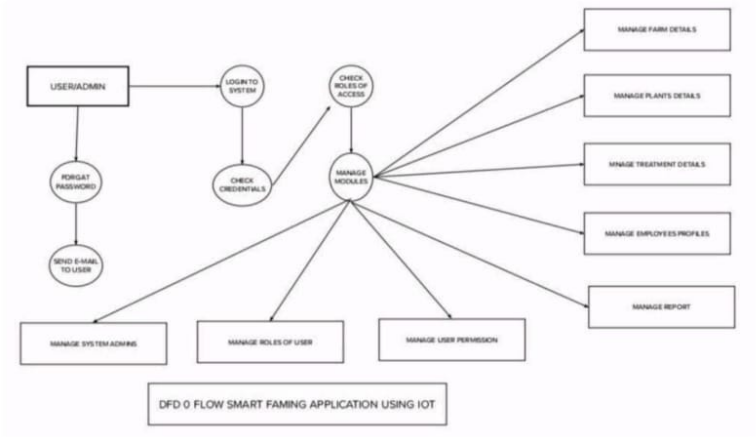
5.1.DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.





- The different soil parameters temperature, soil moistures and then humidity are sensed using different sensors and obtained value is stored in the IBM cloud.
- Arduino UNO is used as a processing Unit that process the data obtained from the sensors and whether data from the weather API.
- NODE-RED is used as a programming tool to write the hardware, software, and APIs. The MQTT protocol is followed for the communication.
- All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could plan through an app, weather to water the crop or not depending upon the sensor values. By using the app they can remotely operate to the motor switch.

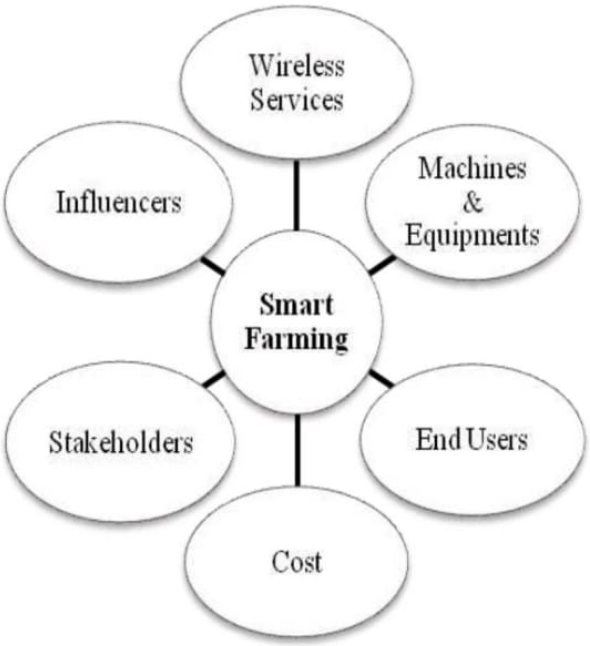
5.2.SOLUTION AND TECHNICAL ARCHITECTURE:

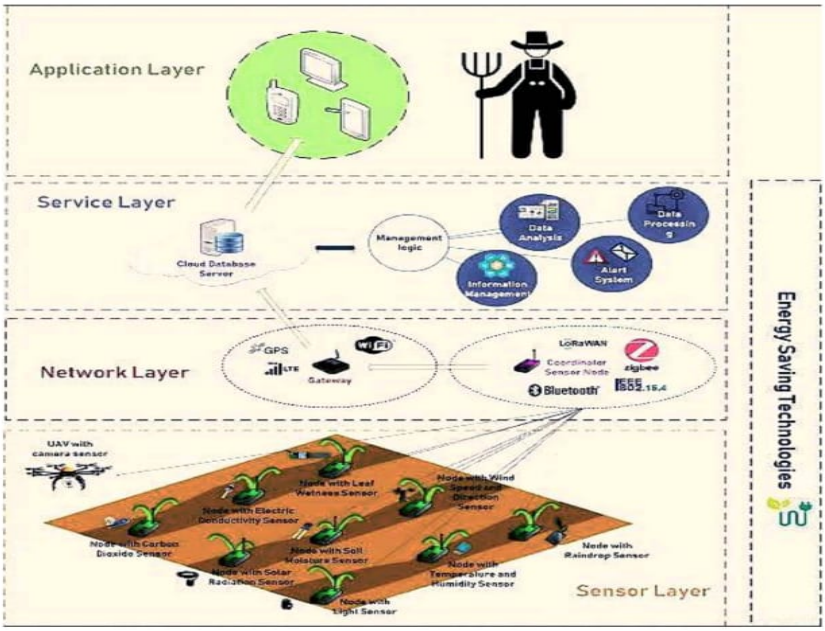
Solution Architecture:

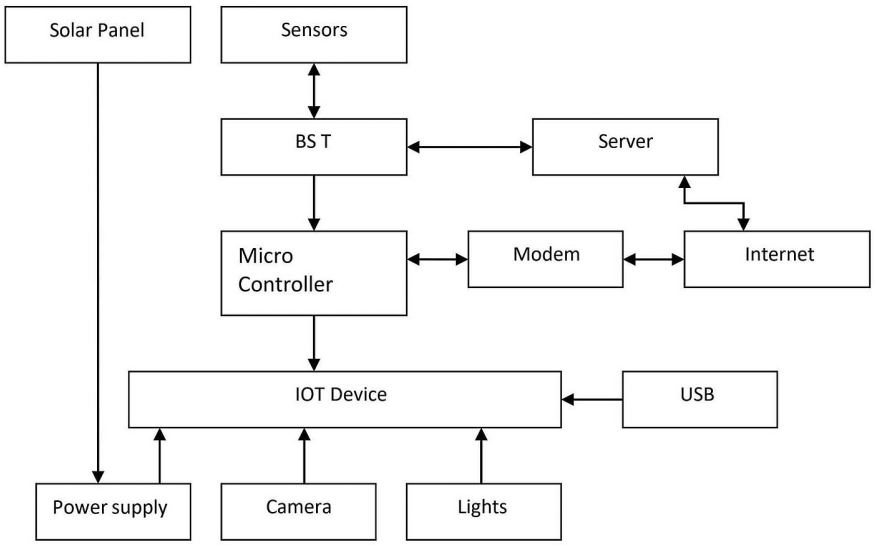
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Solution Architecture Diagram:




Application Layer
Service Layer
Cloud Database Server
Management logic
Data Analysis
Data Processing
Alert System
Information Management
Network Layer
GPS
WiFi
LTE
Gateway
LoRaWAN
Coordinator Sensor Node
zigbee
IEEE 802.15.4
Bluetooth
UAV with camera sensor
Node with Leaf Wetness Sensor
Node with Wind Speed and Direction Sensor
Node with Electric Conductivity Sensor
Node with Carbon Dioxide Sensor
Node with Soil Moisture Sensor
Node with Raindrop Sensor
Node with Solar Radiation Sensor
Node with Temperature and Humidity Sensor
Node with Light Sensor
Sensor Layer
Energy Saving Technologies



TECHNICAL ARCHITECTURE:

Technical Architecture:

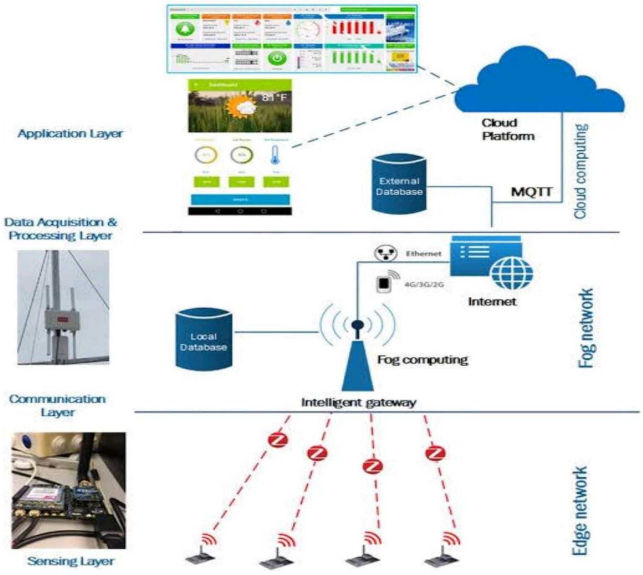


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Mobile App Front End Client side code, which is downloaded and executed by the browser. Back End Server side code, which is executed on the server.	HTML, CSS, JavaScript / Angular Js / React Js Node Js / PHP / Java / Python
2.	Application Logic-1	Collect sensor data from various field sensors and save to a remote database	Java / Python
3.	Application Logic-2	Provide information to farmers about the current conditions of their crops	IBM Watson STT service
4.	Application Logic-3	Help farmers in making decisions about irrigation, fertilization, and pest control	IBM Watson Assistant
5.	Database	Database Persistent storage to store the data The database should be able to store data relating to soil moisture, temperature, humidity, light intensity, and water level.	MySQL, NoSQL, etc.
6.	Cloud Database	MongoDB - Used for storing data	IBM DB2, IBM Cloudant etc.
7.	File Storage	The app also includes a database of crop information, so farmers can quickly look up the best practices for growing their crops	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	The purpose of this API is to provide a means for developers to interact with the Smartfarmer application in order to automate various tasks related to smart farming	IBM Weather API, etc.
9.	External API-2	This includes retrieving data from sensors, controlling actuators, and managing user account	Aadhar API, etc.
10.	Machine Learning Model	The purpose of machine learning in the Smartfarmer application is to provide farmers with predictions about crop yields, based on data collected by sensors in the field. This information can help farmers to make decisions about when to plant, how to irrigate, and what type of fertilizer to use.	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Local Server Configuration: Prerequisites :- You need to have java installed on your system. How to deploy :- Step 1 :- Download the source code zip file and extract it. Step 2 :- Open terminal and navigate to the extracted folder location. Step 3 :- Run the command 'mvn install' to install the dependencies. Step 4 :- Run the command 'mvn spring-boot:run'. The application will start running on port 8080. Cloud Server Configuration : There are a few steps that need to be taken in order to deploy a cloud for the Smartfarmer iot enabled smart farming application. 1. Create a new project in the Google Cloud Platform console. 2. Within the project, create a new App Engine application. 3. Download and install the Google Cloud SDK. 4. Within the App Engine application, create a new service. 5. Within the service, create a new version. 6. Deploy the application to the App Engine service	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Crop Sensor Data Analysis	Python, Tkinte
2.	Security Implementations	Use of firewalls: Firewalls are used to protect the network from unauthorized access	SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	To ensure scalability, the application will need to be designed for horizontal scaling. This means that the application can be run on multiple servers at the same time, with each server handling a portion of the traffic	Using MongoDB, NodeJS, ReactJS, ExpressJS, Redux
4.	Availability	The application would allow farmers to track the status of their crops and soil health in real-time, as well as receive information on the best practices for crop care. The application would also provide alerts to farmers in the event of changes in weather or pests.	sensors, software applications, and cloud-based data storage and analytics.
5.	Performance	The performance of an IoT enabled smart farming application depends on a number of factors, including the quality of the sensors and devices used, the connectivity of the system, the amount of data being collected, and the algorithms used to analyze the data.	soil moisture levels, temperature, and air quality sensors.

6.PROJECT PLANNING AND SCHEDULING :

6.1.SPRINT DELIVERY

Title	Description	Date
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	07 OCTOBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements.	26 SEPTEMBER 2022
Brainstorming ideas	List the ideas by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	09 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	09 OCTOBER 2022
Problem Solution Fit	Prepare problem - solution Fit document.	09 0CTOBER 2022
Solution Architecture	Prepare solution Architecture document.	09 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application	20 OCTOBER 2022

SCHEDULE :

Technology Architecture	Architecture diagram.	20 OCTOBER 2022
Data Flow Diagrams	Draw the data flow Diagrams and submit for review.	20 OCTOBER 2022

Milestone & Activity List	Prepare the milestones & Activity list of the project.	08 NOVEMBER 2022
Project Development Delivery of Sprint- 1,2,3&4	Develop & submit the developed code by testing it.	IN PROGRESS…

6.2.SPRINT DELIVERY SCHEDULE :

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

	Functional Requirement (Epic)	**User Story Number**		**Points**		**Team Members**
Sprint-1	Simulation creation	USN-1	Connect Sensors and Arduino with python code	2	High	Chitrabanu, Priyanka, Shanmugapriya, Sharmila.
Sprint-2	Software	USN-2	Creating device in the IBM Watson IoT platform, workflow for IoT scenarios using Node-Red	2	High	Chitrabanu, Priyanka, Shanmugapriya, Sharmila.
Sprint-3	MIT App Inventor	USN-3	Develop an application for the Smart farmer project using	2	High	Chitrabanu, Priyanka, Shanmugapriya, Sharmila.

			MIT App Inventor			

Sprint	User Story / Task			Story Priority		
Sprint-3	Dashboard	USN-3	Design the Modules and test the app	2	High	Chitrabanu, Priyanka, Shanmugapriya, Sharmila.
Sprint-4	Web UI	USN-4	To make the user to interact with software.	2	High	Chitrabanu, Priyanka, Shanmugapriya, Sharmila.

Project Tracker, Velocity & Burndown Chart: (4 Marks)

	Total Story Points	**n**	**Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	7 Days	30 Oct 2022	06 Nov 2022	20	29 Oct 2022
Sprint-2	20	9 Days	31 Oct 2022	09 Nov 2022		05 Oct 2022

Sprint-3	20	6 Days	06 Nov 2022	13 Nov 2022		12 Oct 2022
Sprint-4	20	6 Days	11 Nov 2022	17 Nov 2022		15 Oct 2022

Start Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

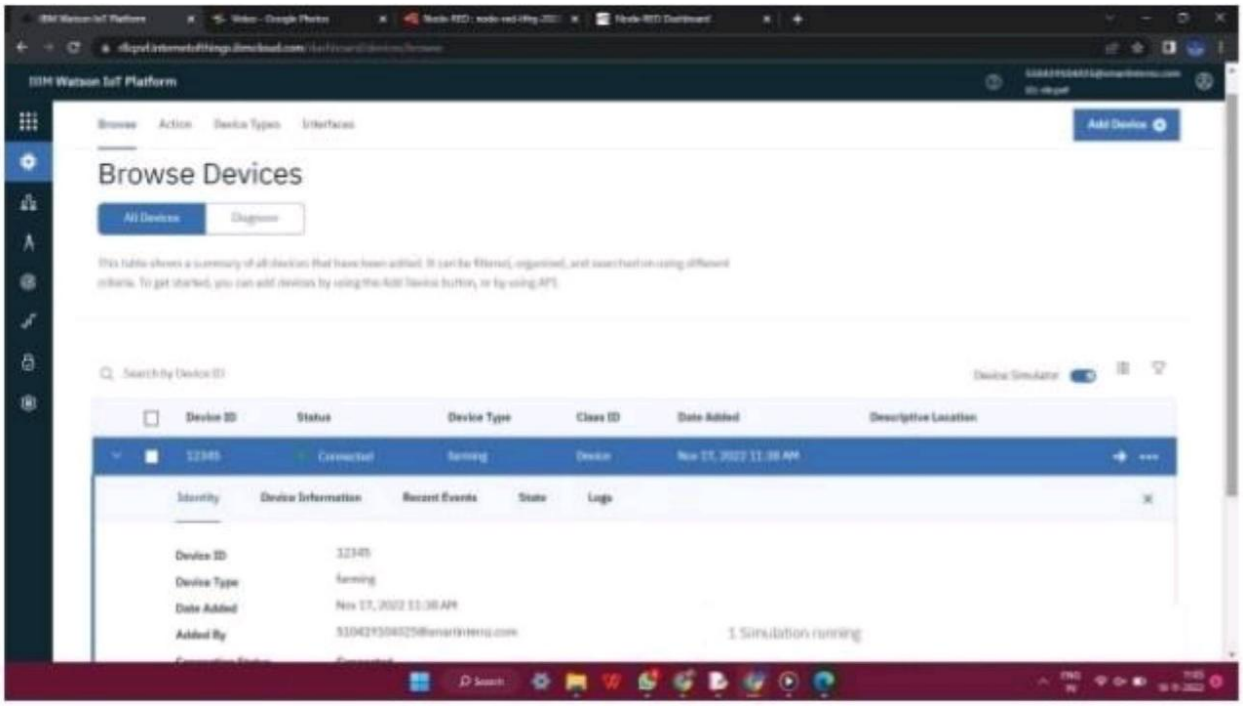
7.CODING AND SOLUTIONING:

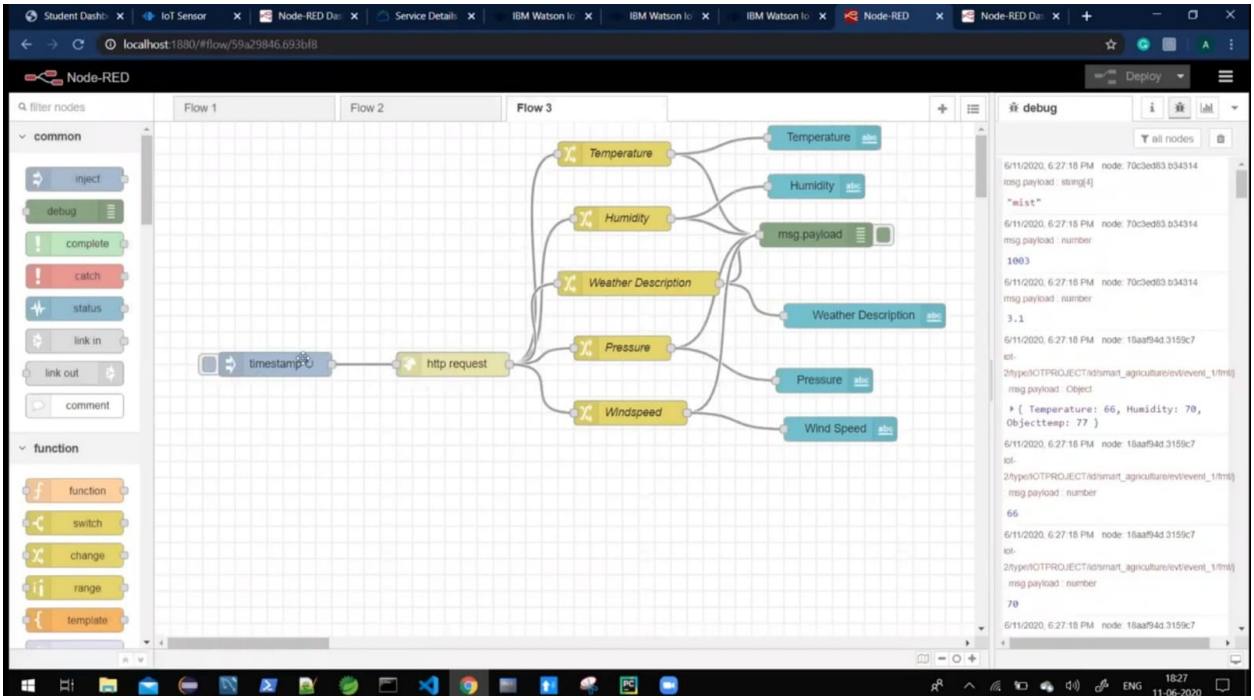
7.1.CODING:

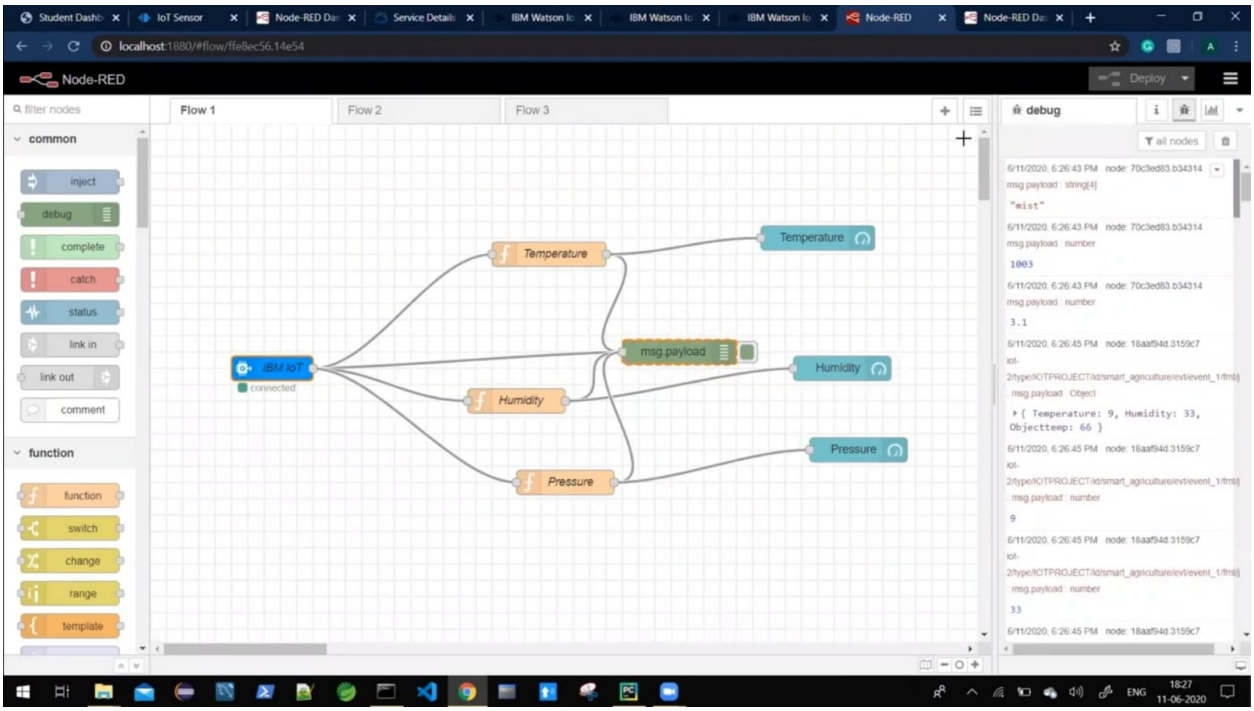
```
import time
import sys import ibmiotf.application import ibmiotf.device import random
#Provide your IBM Watson Device organization = "r8cpvf" deviceType = "farming"
deviceId = "12345" authMethod = "token" authToken ="87654321" # Initialize GPIO def
myCommandCallback(cmd): print("Commandreceived: %s" % cmd.data['command'])
status=cmd.data['command'] if status=="motoron":
print ("motor is on")
elif status == "motoroff":
print("motor is off")
else : print ("please send proper command")
try:
deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"auth-method": authMethod, "auth-token": authToken} deviceCli =
ibmiotf.device.Client(deviceOptions)
#..... except Exception as e:
print("Caught exception connecting device: %s" %str(e)) sys.exit()
# Connect and send a datapoint "hello" with value "world" into the cloud as aneventof
type "greeting" 10 times deviceCli.connect() while True:
#Get Sensor Data fromDHT11 temp=random.randint(90,110)
Humid=random.randint(60,100) Mois=random. randint(20,120) data = { 'temp' : temp,
'Humid': Humid ,'Mois': Mois}
#print data def myOnPublishCallback():
print ("Published Temperature = %s C" % temp, "Humidity
= %s %%" %Humid, "Moisture =%s deg c" % Mois, "to IBM Watson") success =
deviceCli.publishEvent("IoTSensor", "json",
data,qos=0,on_publish=myOnPublishCallback) if not success:
print("Not connected to IoTF")
time.sleep(10) deviceCli.commandCallback = myCommandCallback #Disconnect the
device and application from the cloud deviceCli.disconnect()
```

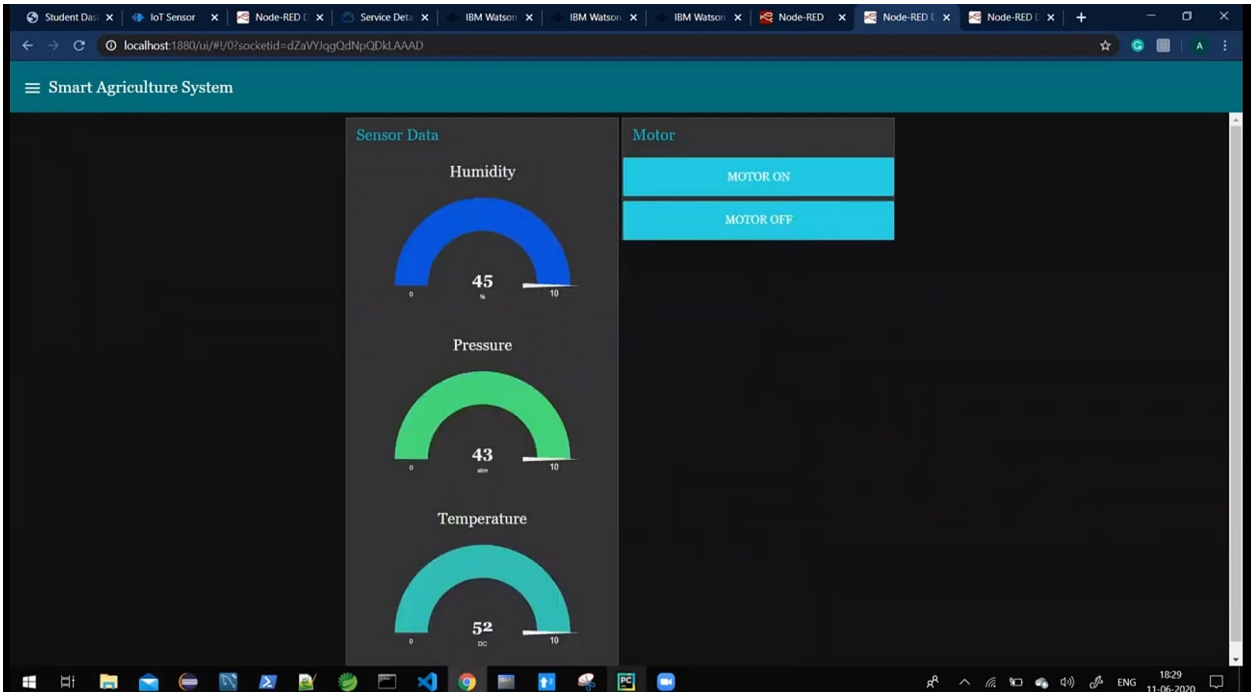
7.2 SOLUTION:

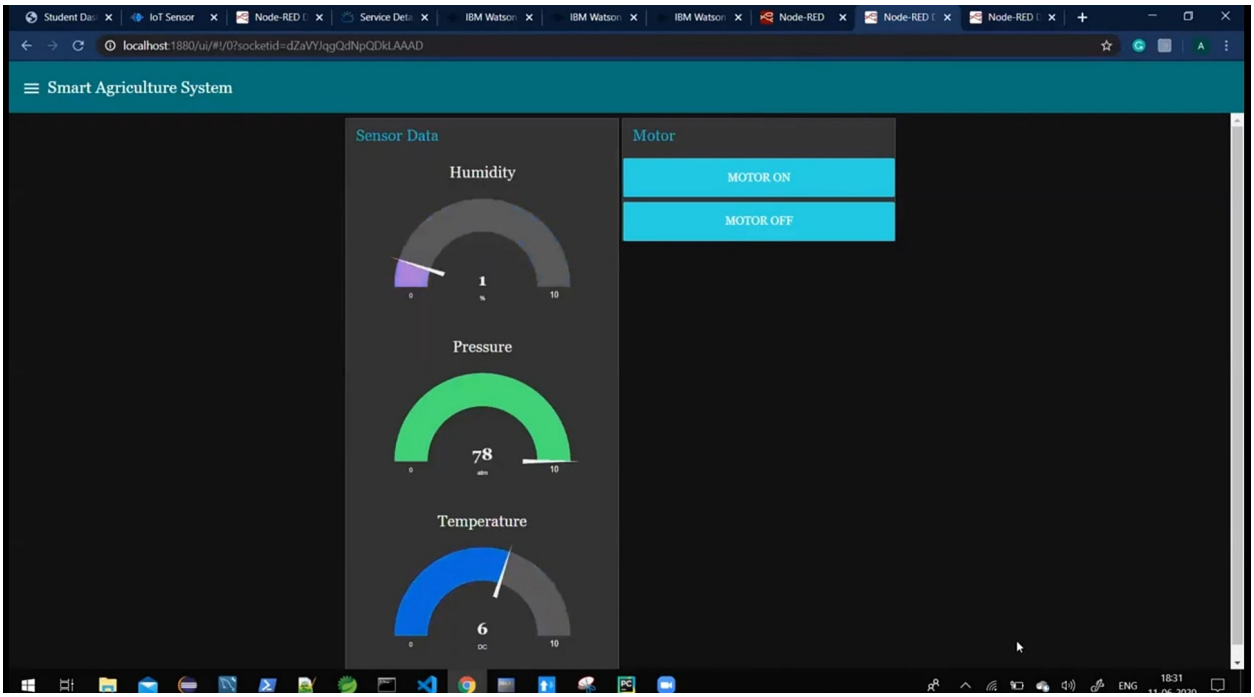
8.TESTING:


IBM Watson IoT Platform
Browse
Action
Device Types
Interfaces
Add Device
Browse Devices
All Devices
Diagnose
Device Simulator
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
12345
Connected
farming
Device
Identity
Device Information
Recent Events
State
Logs
Device ID
12345
Device Type
farming
Date Added
Added By
1 Simulation running


Node-RED
Deploy
Flow 1
Flow 2
Flow 3
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
range
template
timestamp
http request
Temperature
Humidity
Weather Description
Pressure
Windspeed
Temperature
Humidity
msg.payload
Weather Description
Pressure
Wind Speed
debug
all nodes
"mist"
1003
3.1
{ Temperature: 66, Humidity: 70, Objecttemp: 77 }
66
70

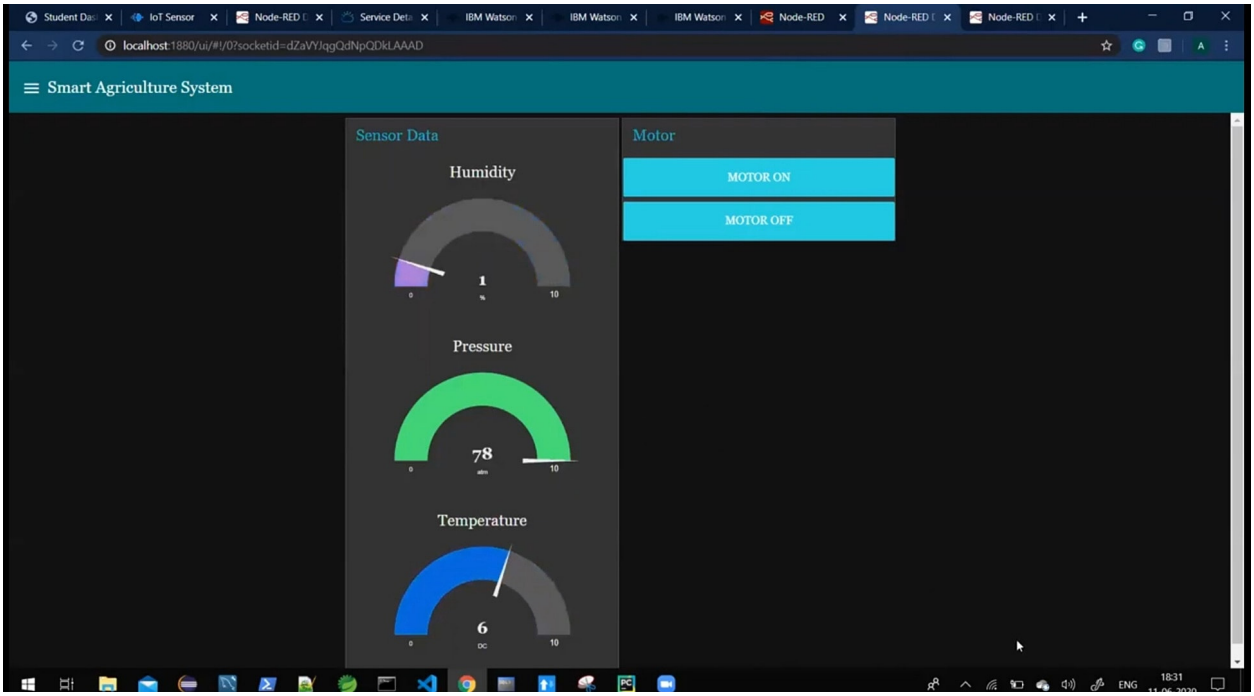

Node-RED
Deploy
Flow 1
Flow 2
Flow 3
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
range
template
IBM IoT
connected
Temperature
Humidity
Pressure
msg.payload
Temperature
Humidity
Pressure
debug
all nodes
"mist"
1003
3.1
{ Temperature: 9, Humidity: 33, Objecttemp: 66 }
9
33





9.RESULTS:

We have successfully bulit a web based UI and integrated all the services using Node-RED.



10.ADVANTAGES AND DISADVANTAGES:

10.1.ADVANTAGES:

•All the data like climatic conditions and changes in them, soil or crop conditions everything can be easily monitored.

•Risk of crop damage can be lowered to a greater extend

•Many difficult challenges can be avoided making the process automated and the quality of crops can be maintained.

•The process included in farming can be controlled using the web applications from anywhere, anytime**.**

10.2.DISADVANTAGES:

•Smart Agriculture requires internet connectivity continuously, but rural parts can not fulfill this requirement.

•IoT devices need much money to implement.

11.CONCLUSION:

An IOT based smart agriculture system using Watson IOT Platfrom , Watson simulator , IBM cloud and Node-RED.

12.FUTURE SCOPE :

In future due to more demand of good and more farming in less time, for betterment of the crops and reducing the usage of extravagant resources like electricity and water IoT can be implemented in most of the places.

13.APPENDIX :

13.1.SOURCE CODE:

```
import time
import sys
 import ibmiotf.application
 import ibmiotf.device
 import random
#Provide your IBM Watson Device organization = "r8cpvf" deviceType = "farming"
deviceId = "12345" authMethod = "token" authToken ="87654321" # Initialize GPIO def
myCommandCallback(cmd): print("Commandreceived: %s" % cmd.data['command'])
status=cmd.data['command'] if status=="motoron":
print ("motor is on")
elif status == "motoroff":
print("motor is off")
else :
print ("please send proper command")
try:
deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"auth-method": authMethod, "auth-token": authToken} deviceCli =
ibmiotf.device.Client(deviceOptions)
#..... except Exception as e:
print("Caught exception connecting device: %s" %str(e)) sys.exit()
# Connect and send a datapoint "hello" with value "world" into the cloud as aneventof
type "greeting" 10 times deviceCli.connect() while True:
#Get Sensor Data fromDHT11 temp=random.randint(90,110)
Humid=random.randint(60,100) Mois=random. randint(20,120) data = { 'temp' : temp,
'Humid': Humid ,'Mois': Mois}
#print data def myOnPublishCallback():
print ("Published Temperature = %s C" % temp, "Humidity
= %s %%" %Humid, "Moisture =%s deg c" % Mois, "to IBM Watson") success =
deviceCli.publishEvent("IoTSensor", "json",
data,qos=0,on_publish=myOnPublishCallback) if not success:
print("Not connected to IoTF")
time.sleep(10) deviceCli.commandCallback = myCommandCallback
#Disconnect the device and application from the cloud deviceCli.disconnect()
```

13.2. GITHUB AND PROJECT DEMO LINK:

PROJECT DEMO LINK:

https://drive.google.com/file/d/1k-DrGlUcFn3Mjg14t2lJz5tfMS6p-RDM/view?usp=drivesdk

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-40742-1660633765