Project Report

Date	15 November 2022
Team ID	PNT2022TMID29181
Project Name	Project–Smart Farmer-IoT Enabled smart Farming Application

INTRODUCTION:

Project Overview:

The main aim of this project is to help farmers automate their farms by providing them with a Web App through which they can monitor the parameters of the field like Temperature, soil moisture, humidity and etc. and control the equipment like water motor and other devices remotely via internet without their actual presence in the field

Purpose:

To improve the farmer's working conditions and make them easier, we introduce IoT services to him in which we use cloud services and internet to enable farmer to continue his work remotely via internet. He can monitor the field parameters and control the devices in the farm.

LITERATURE SURVEY:

Existing problem and example:

Farmers are to be present at the farm for its maintenance irrespective of the weather conditions. They must ensure that the crops are well watered, and the farm status is monitored by them physically. Farmer must stay most of the time in field in order to get a good yield. In difficult times like in the presence of pandemic, also they must work hard in their fields risking their lives to provide food for the country.

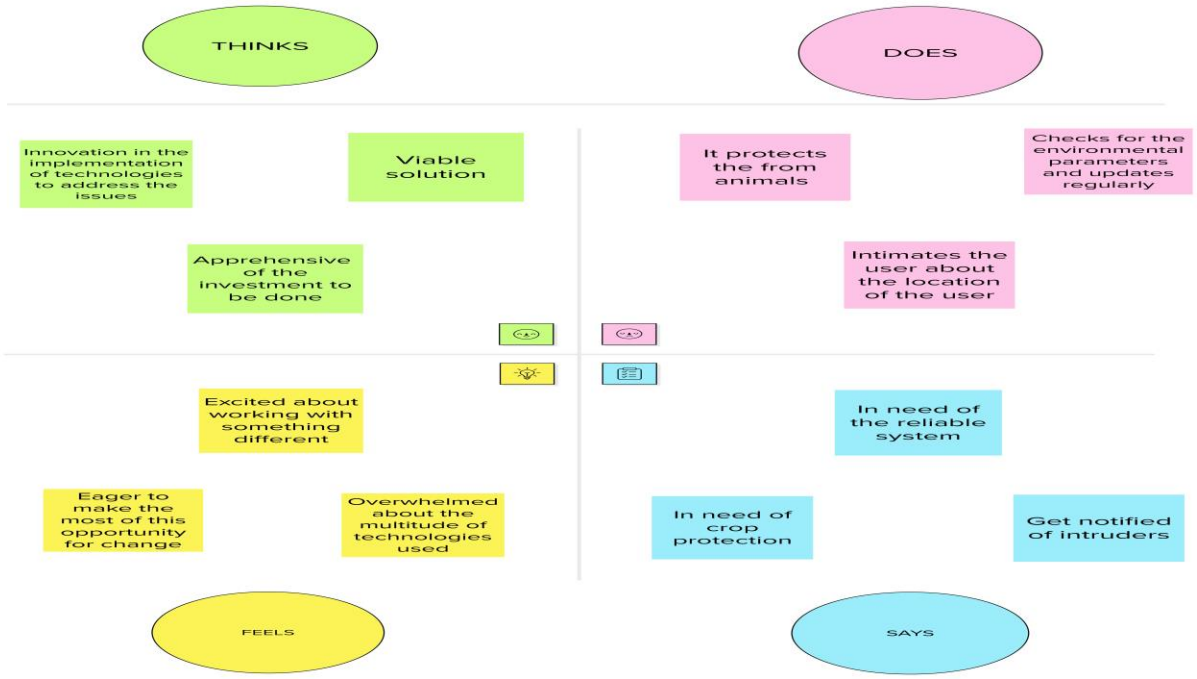
Problem Statement Definition:

Mr. Xavier is a farmer who has an interest in implementing technology in agriculture to solve the problems in agriculture. He's moved into agriculture with his father. Since he is a beginner in farming, he needs someone to guide him in the initial years and he plan to incorporate technology into farming to reduce the work and labor, improve productivity, more yield, suggestions to improve soil, and next crop planting ideas.

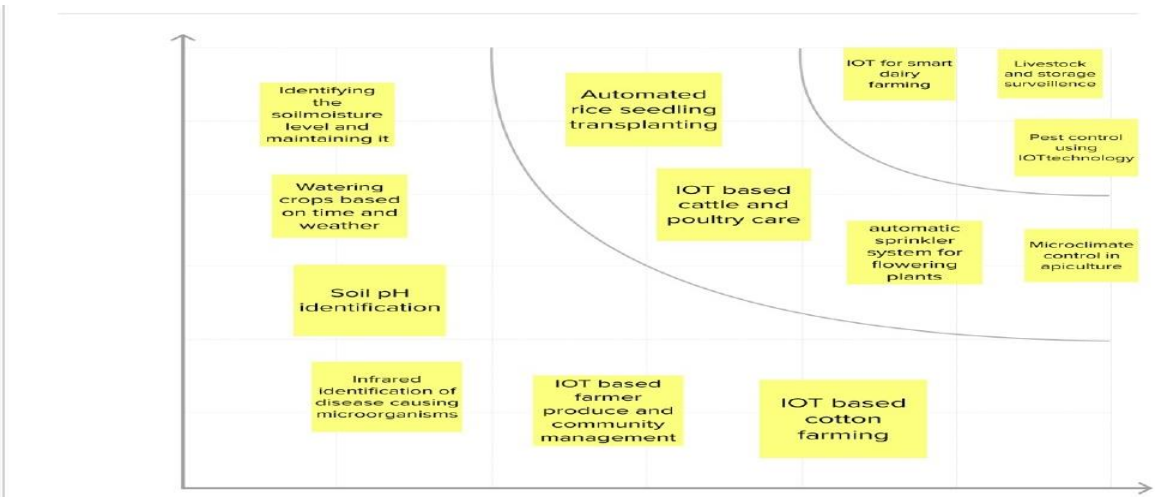
He is actively researching a few agriproducts that solve his problem. These problems are common to many beginning and experienced farmers.

IDEATION & PROPOSED SOLUTION :

Empathy Map Canvas:



Ideation & Brainstorming:



Proposed Solution:

S.No.	Parameter	Description
1	Problem Statement	Watering the field is a difficult process, Farmers must wait in the field until the water covers the whole farm field.
2	Solution description	As is the case of precision Agriculture, Smart Farming Technique Enables Farmers better to monitor the fields and maintain the humidity level accordingly.
3	Uniqueness	Alert notification and remote access reduce the stress of farmers
4.	Customer Satisfaction	Easily identify maintenance needs, build better products, send personalized communications, and it saves a lot of time
5	Scalability of the Solution	Scalability in smart farming refers to the adaptability of a system to increase the capacity

Problem Solution fit:

- Our goal is to help the farmers to monitor field parameters remotely
- Using many sensors is difficult. An unlimited or continuous internet connection is required for success
- The irrigation process is automated using IOT.

REQUIREMENT ANALYSIS:

1. Functional Requirements:

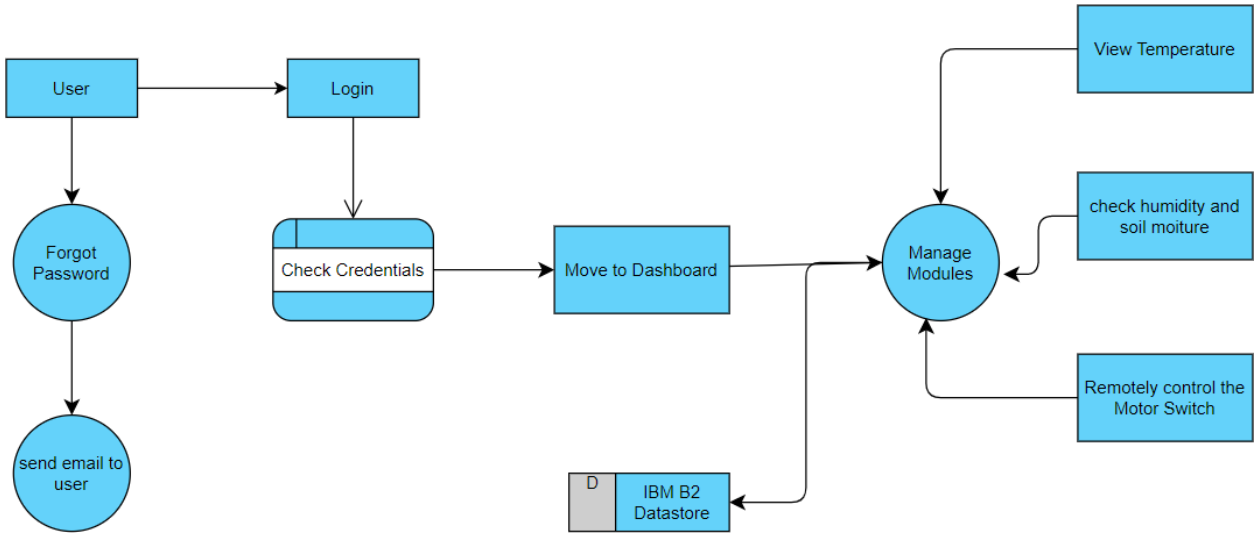
FR No.	Functional Requirement	Sub Requirement
FR-1	User Registration	Registration through Gmail or form
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Sensor Function for framing System	Measure the Temperature and Humidity Measure the Soil Monitoring Check the crop diseases
FR-4	Manage Modules	Manage Roles of User Manage User permission
FR-5	Check whether details	Temperature and Humidity details
FR-6	Data Management	Manage the data of weather conditions Manage the data of crop conditions

2. Non-functional Requirements:

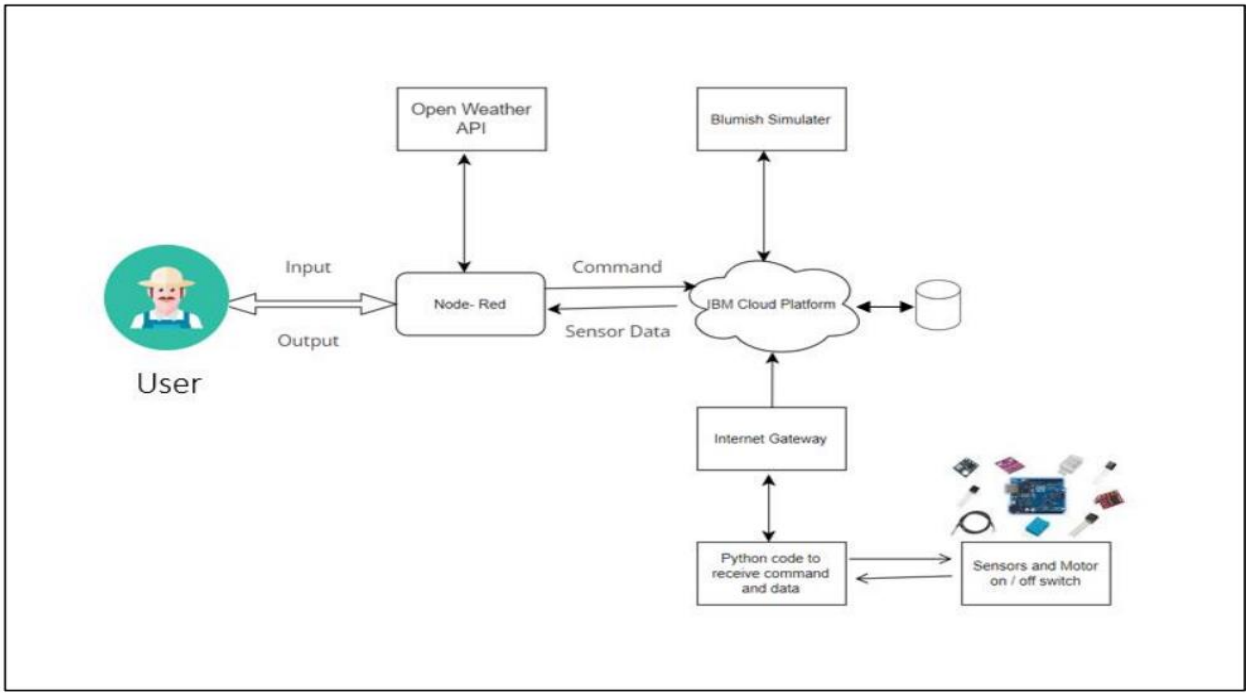
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	✓ User friendly guidelines for users to avail themselves of the features. ✓ Most simplistic user interface for ease of use.
NFR-2	**Security**	Detection and identification of any misfunctions of sensors.
NFR-3	**Reliability**	Building a Multi-layered defense for IoT Networks.
NFR-4	**Performance**	The use of modern technology solutions helps to achieve the maximum performance thus resulting in better quality and quantity yields.
NFR-5	**Availability**	This app is available for all platforms
NFR-6	**Scalability**	Scalability refers to the ability to increase available resources and system capability without the need to go through a major system redesign or implementation.

PROJECT DESIGN:

Data Flow Diagram



Technical Architecture:



User Stories

User Type	Functional Requirement	User Story Number	User Story / Task	Acceptance criteria	Priority
Mobile user	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High
	Login	USN-2	As a user, I can log into the application by entering email & password		High
Customer (Web user)	Dashboard	USN-3	As a User can view the dashboard, and this dashboard includes the check roles of access and then move to the manage modules.	I can view the dashboard in this smart farming application system.	High
		USN-4	User can remotely access the motor switch	In the smart farming app	High
Administrator			As a user once view the manage modules this Describes the Manage system Admins and Manage Roles of User etc.		

PROJECT PLANNING:

m

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

Sprint			User Story / Task	Story	Priority	
Sprint-3	Dashboard	USN-3	Design the Modules and test the app	2	High	Sathiyanarayanan,Velmurugan
Sprint-4	Web UI	USN-4	To make the user interact with software.	2	High	Sathiyanarayanan,Velmurugan,Boopathi,Shyam Chander

CODING & SOLUTIONING

Features:

```
import wiotp.sdk.device
 import time
 import os
 import datetime
import random
 myConfig ={
    "identity": {
        "orgId": "0hzydu",
        "typeId": "NodeMCU",
        "deviceId": "12345"
       },
```

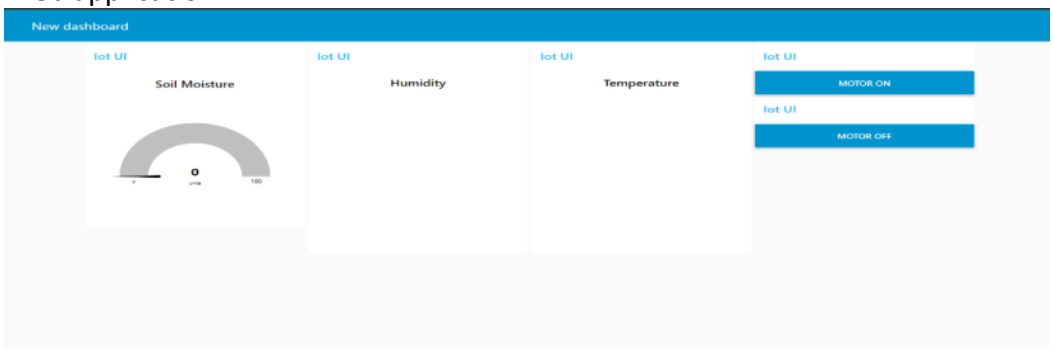
```
"auth": {
        "token": "12345678"
         }
           }
        client =
        wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=Non
        e)
         client.connect ()
        def myCommandCallback (cmd) :
                 print("Message received from IBM IoT Platform: %s"
                %cmd.data['command'])
                 m=cmd.data['command']
                 if (m=="motoron"):
                    print("Motor is switchedon")
                elif (m=="motoroff"):
                     print ("Motor is switchedOFF")
                 print (" ")
                while True:
                     moist =random.randint (0,100)
                temp=random.randint (-20, 125)
                 hum=random.randint (0, 100)

                myData={'moisture':moist,'temperature':temp,'humidity':hu
                m} client.publishEvent (eventId="status",
                msgFormat="json", data=myData, qos=0 , onPublish=None)
                print ("Published data Successfully: %s",myData)
```

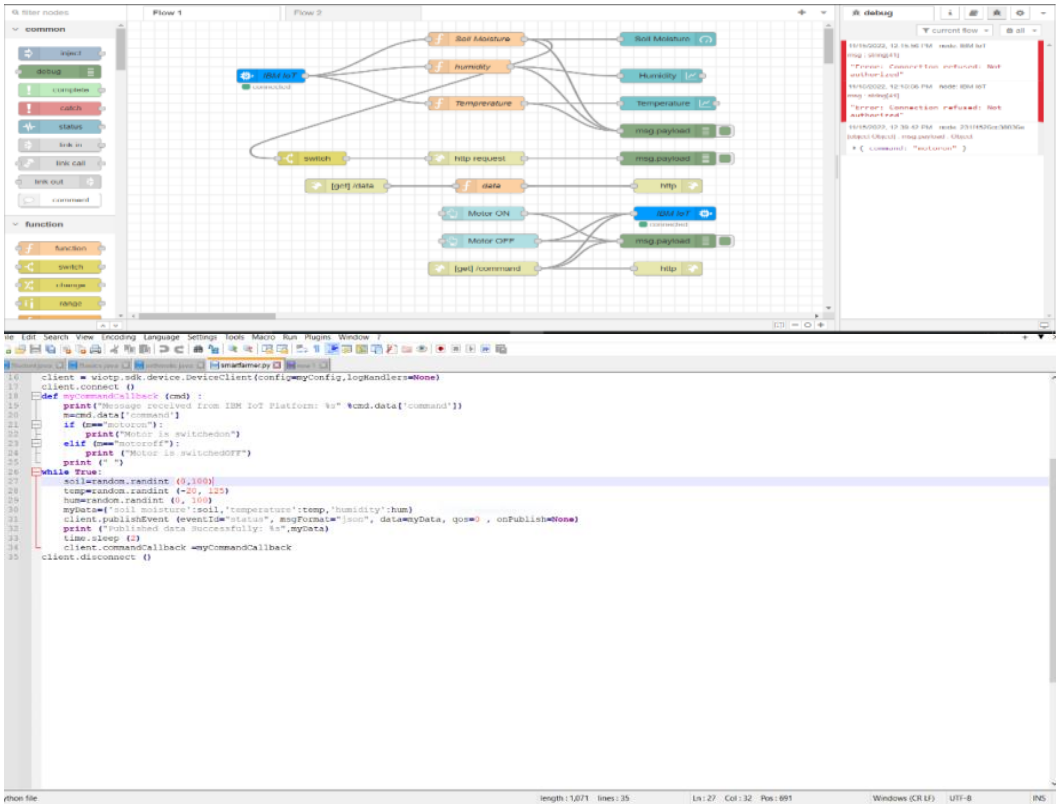
```
 time.sleep (2)
 client.commandCallback =myCommandCallback
client.disconnect ()
```

TESTING:

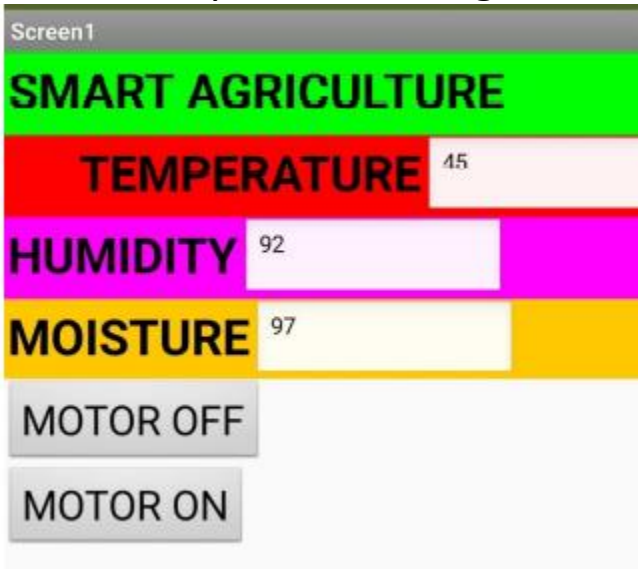
Test case

Web application





User Acceptance Testing:



RESULTS:

Performance Metrics

ADVANTAGES:

➢ A remote control system can help in working irrigation system valves dependent on schedule. Irrigating remote farm properties can be exceptionally troublesome and labor-intensive. It gets hard to comprehend when the valves were started and whether the ideal measure of water was distributed.

➢ For situations where a quick reaction is required, manual valve actuation may not be conceivable constantly. Thus, remote observing and control of irrigation systems, generators or wind machines or some other motor-driven hardware become the next logical step.

DISADVANTAGE:

➢ The smart agriculture needs availability of internet continuously. The rural part of most of the developing countries do not fulfil this requirement. Moreover, internet connection is slower.

➢ The smart farming-based equipment require farmers to understand and learn the use of technology. This is a major challenge in adopting smart agriculture farming at a large scale across the countries.

CONCLUSION:

Farmers can benefit greatly from an IoT-based smart agriculture system. As a result of the lack of irrigation, agriculture suffers. Climate factors such as humidity, temperature, and moisture can be adjusted depending on the local environmental variables. This technology also detects animal invasions, which are a major cause of crop loss. This technology aids in the scheduling of irrigation based on present data from the field and records from a climate source.

FUTURE SCOPE:

In the current project we have implemented the project that can protect and maintain the crop. In this project the farmer monitors and controls the field remotely. In future we can add or update a few more things to this project.

• We can create a few more models of the same project, so that the farmer can have information.

• We can update this project by using solar power mechanism. So that the power supply from electric poles can be replaced with solar panels. It reduces the power line cost. It will be a one-time investment. We can add solar fencing technology to this project.

Appendix:

```
import wiotp.sdk.device
 import time
 import os
 import datetime
import random
 myConfig ={
```

```
    "identity": {
        "orgId": "0hzydu",
        "typeId": "NodeMCU",
        "deviceId": "12345"
       },
```

```
"auth": {
        "token": "12345678"
         }
           }
        client =
        wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=Non
        e)
         client.connect ()
        def myCommandCallback (cmd) :
                 print("Message received from IBM IoT Platform: %s"
                %cmd.data['command'])
                 m=cmd.data['command']
                 if (m=="motoron"):
                    print("Motor is switchedon")
                elif (m=="motoroff"):
                     print ("Motor is switchedOFF")
                 print (" ")
                while True:
                     moist =random.randint (0,100)
                temp=random.randint (-20, 125)
                 hum=random.randint (0, 100)
```

```
myData={'moisture':moist,'temperature':temp,'humidity':hu
m} client.publishEvent (eventId="status",
msgFormat="json", data=myData, qos=0 , onPublish=None)
print ("Published data Successfully: %s",myData)
 time.sleep (2)
 client.commandCallback =myCommandCallback
client.disconnect ()
```

GitHub link: https://github.com/IBM-EPBL/IBM-Project-30683-1660153918
Project demo link: https://drive.google.com/file/d/13-w_2pGSfg7C53rbaj0HtWGaSSb-dgII/view?usp=share_link