TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1PROJECT OVERVIEW

The objective of this project is to offer assistance to farmers in getting Live Data (Temperature, Humidity, Soil Moisture,Ph-Level,CO_2 intake of plants) for efficient environment monitoring which will enable them to increase their overall yield and quality of products.

The parameters like temperature, humidity, and soil moisture are updated to the Watson IoT platform. The device will subscribe to the commands from the mobile application and control the motors accordingly. APIs are developed using Node-RED service for communicating with Mobile Application.A mobile application is developed using the MIT App inventor to monitor the sensor parameters and control the motors.Fast2sms app will send the SMS if there is in need.

1.2 PURPOSE

Nowadays Internet of Things (IoT) technology is one of the fastest growing fields in different domains including agriculture. Thanks to this technology, Smart farming systems know a cultural change toward modern agriculture which is more productive, consuming less water and even cheaper.

The main goal of my project is to use IoT in the agriculture field in order to collect data instantly (Soil moisture , Temperature , Humidity, Ph level ,CO2 intake), which will help one to monitor some environment conditions remotely, effectively and enhance tremendously the production and therefore the income of farmers. The present prototype is developed using python which generates random data, IBM Watson that helps to collect instant data online, The mobile application is created to view the collected data. The alert message will be sent to the respective user through FAST2SMS.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

CropX offers an integrated hardware and software system with a suite of decision and planning tools based on continuous monitoring of soil and crop conditions.

The sensors send the collected data to the Cloud and send back advice and insights accessible from any mobile device, tablet, or computer (iOS or Android). Know more to grow more with less - less water, less chemicals, less energy, less time, and less stress.

The CropX software uses GPS monitoring system to collect and show the data. The main problem is that it will not indicate the alert messages which warns the user on how to monitor the plants and immediate results will not be generated which is a major disadvantage.

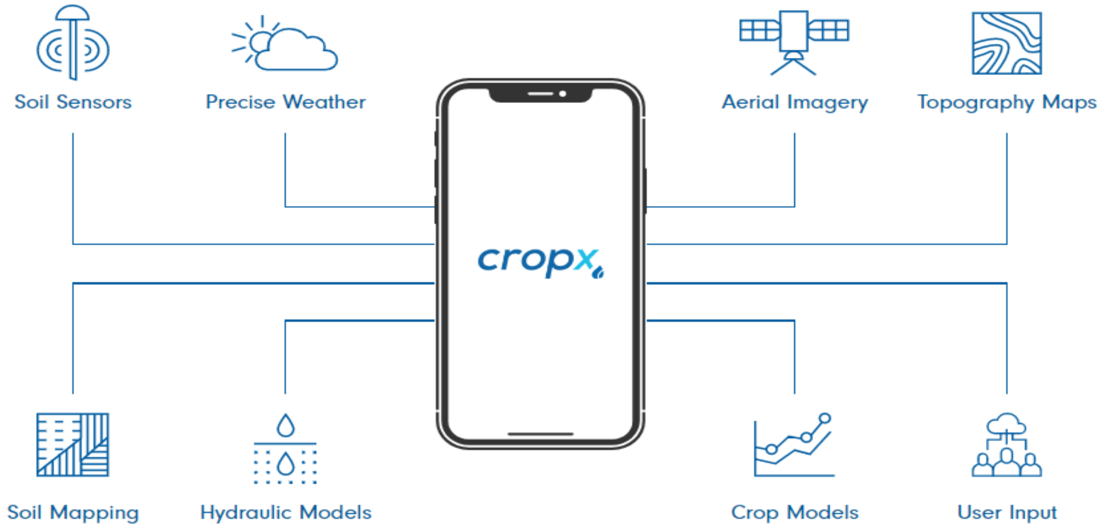


Fig.2.1.1 Existing Technology

2.2 REFERENCE

https://cropx.com/technology/

2.3 PROBLEM STATEMENT DEFINITION

To make farming easier by choosing several constraints in agriculture and to overcome those constraints and increase production quality & quantity using IoT by building a mobile application.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

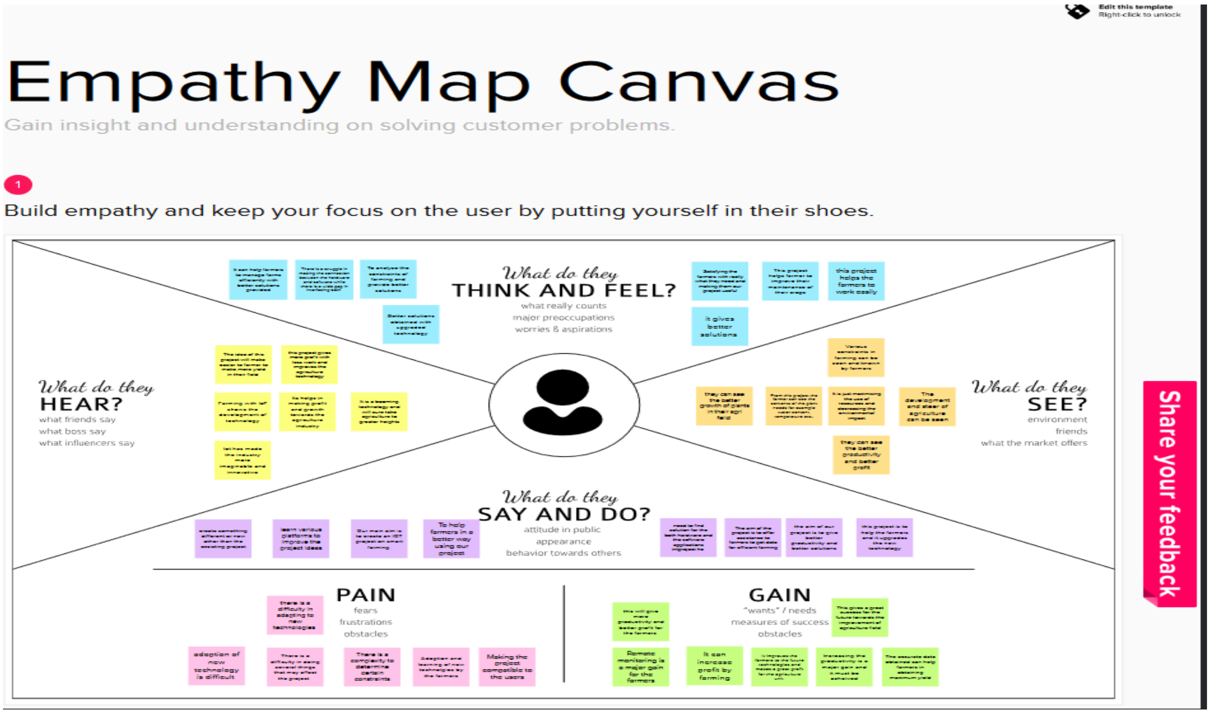


Fig3.1.1 Empathy Map Canvas

3.2 IDEATION AND BRAINSTORMING

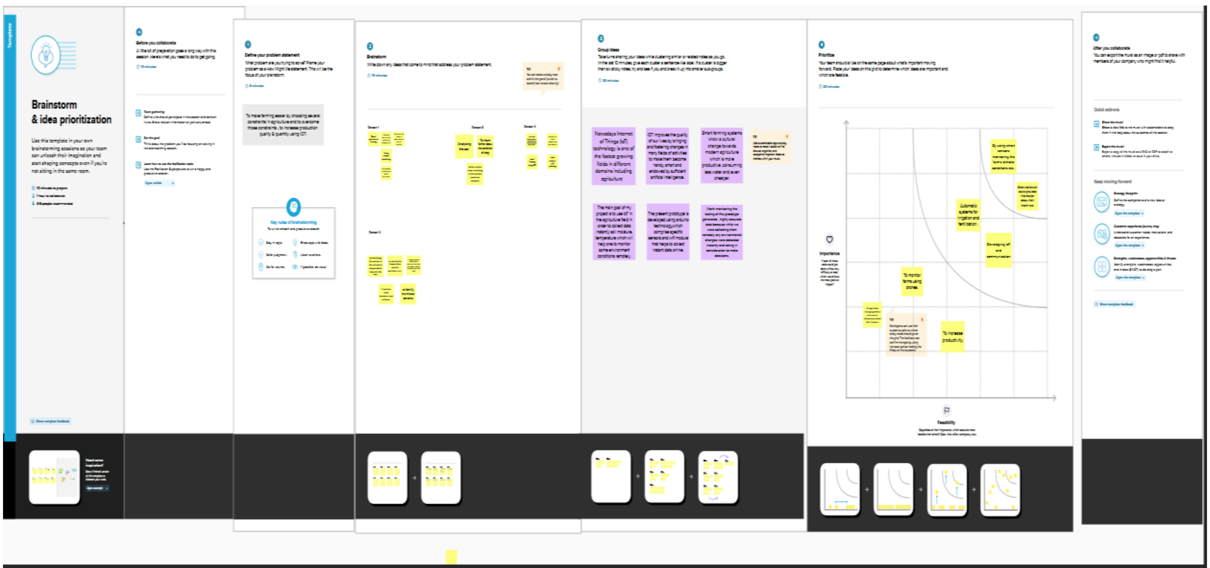


Fig3.2.1 Ideation And Brainstorming

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To make farming easier by choosing several constraints in agriculture and to overcome those constraints, to increase production quality and quantity using IOT.
2.	Idea / Solution description	Using smart techniques like monitoring farms climate, smart irrigation and soil analysis.
3.	Novelty / Uniqueness	Solar power smart irrigation system which helps you to monitor temperature, moisture ,humidity using smart sensors.
4.	Social Impact / Customer Satisfaction	It is better than the present modern irrigation system by using this method we can control soil erosion. There will be better production yield.
5.	Business Model (Revenue Model)	As the productivity increases customer satisfaction also increases and hence need for the application also increases, which increases the revenue of the business.
6.	Scalability of the Solution	It is definetly scalable we ca increase the constraints when the problem arises.

3.4 PROPOSED SOLUTION FIT

Project Title: **Project Design Phase-I - Solution Fit Template** **Team ID:** PNT2022TMIDxxxxxx

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Farmers are our Customers.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

The availabilty of device, proper Network facilities and budget are several constraints ,Knowledge about the application.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem
or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Most commonly used irrigation type is Drip irrigation the most common disadvantage is when the water is not filtered properly there will be clogs and the tubes will get affected easily. In smart farming we can use solar empowered smart irrigation system to overcome this.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

To make farming easier more quantitately.

1.Monitoring farms climatic conditions.
2.Automatic systems for irrigation and Fertilization.
3.Soil analysis.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

When there is no knowledge about the soil problem arises on what to be sowed, climatic conditions also play a major role. Knowledge on how to water the plants accordingly

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

The customers will reach us when they dont have idea on how to analyse the soil and to improve the current irrigation system

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

To get correct accuracy on what to be done on the farm and to produce more crops and livestock quantitavely.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

As when the productivity increases farmers will be satisfied. They will not worry about the loss. Irrigation will be more efficient than before.

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

There will be less weed growth, Maximum use of water efficiently,Control of soil erosion and maximum crop yield.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

we will reach the customer directly ask about their problems and provide effective solutions if their problems match our application and provide them knowledge about our application to make their farming even more easier.

In online mode will do digital marketing using advertisements.

Identify strong TR & EM

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Following are the functional requirements of the proposed solution

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	IoT devices	Sensors and Wifi module.
FR-2	Software	Web UI, Node-red, IBM Watson, MIT app

Table4.1 Functional Requirements

4.2 NON-FUNCTIONAL REQUIREMENTS

Following are the non-functional requirements o the proposed solution

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Time consumability is less, Productivity is high.
NFR-2	**Security**	It has low level of security features due to integration of sensor data.
NFR-3	**Reliability**	Accuracy of data and hence it is Reliable.
NFR-4	**Performance**	Performance is high and highly productive.
NFR-5	**Availability**	With permitted network connectivity the application is accessible
NFR-6	**Scalability**	It is perfectly scalable many new constraints can be added

Table 4.2.1 Non-Functional Requirements

CHAPTER 5

PROJECT DESIGN

5.1 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.

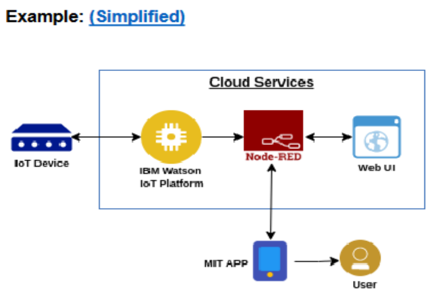


Fig5.1.1 Simplified DFD

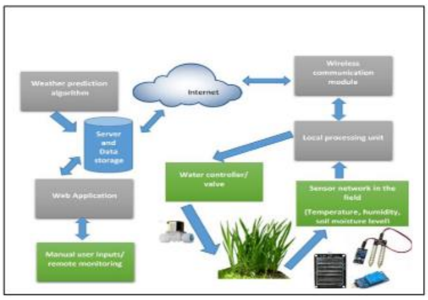


Fig5.1.2 Level 0 DFD

5.2 SOLUTION & TECHNICAL ARCHITECTURE

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	MIT app
2.	Application Logic-1	Logic for a process in the application	Node red/IBM Watson/MIT app
3.	Application Logic-2	Logic for a process in the application	Node red/IBM Watson/MIT app
4.	Application Logic-3	Logic for a process in the application	Node red/IBM Watson/MIT app
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM cloud.
7.	Temperature sensor	Monitors the temperature of the crop	
8.	Humidity sensor	Monitors the humidity	
9.	Soil moisture sensor (Tensiometers)	Monitors the soil temperature	
10.	Weather sensor	Monitors the weather	.
11.	Solar panel		.
12.	RTC module	Date and time configuration	
13.	Relay	To get the soil moisture data	

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	MIT app,Node-Red	Software
2.	Scalable Architecture	Drone technology, pesticide monitoring ,Mineral identification in soil	Hardware

Fig 5.2.1 Components and Technologies

5.3 USER STORIES

The user stories of the proposed solution are listed below

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	IoT devices	USN-1	Sensors and wi-fi module		High	Sprint-1
Customer	Software	USN-2	IBM Watson IoT platform, Workflows for IoT scenarios using Node-red		High	Sprint-2
Customer	MIT app	USN-3	To develop an application using MIT		High	Sprint-3
Customer	Web UI	USN-4	To make the user to interact with the software.	User can access the app for the services.	High	Sprint-4

Fig 5.3.1 User Stories

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

The below table shows the planning and estimation of sprints

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Hardware	USN-1	Sensors and wi-fi module with python code.	2	High	Vishnu Priya, Thenmolhi, Swarna Prabha, Sneha
Sprint-2	Software	USN-2	IBM Watson IoT platform, Workflows for IoT scenarios using Node-red	2	High	Vishnu Priya, Thenmolhi, Swarna Prabha, Sneha
Sprint-3	MIT app	USN-3	To develop an mobile application using MIT	2	High	Vishnu Priya, Thenmolhi, Swarna Prabha, Sneha
Sprint-4	Web UI	USN-4	To make the user to interact with software.	2	High	Vishnu Priya, Thenmolhi, Swarna Prabha, Sneha

Table 6.1.1 Sprint Planning and Estimation

6.2 SPRINT DELIVERY SCHEDULE

The below table shows the schedule of sprints to be delivered

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	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		5th NOV 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		12th NOV 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		14th NOV 2022

Table 6.2.1 Sprint Delivery Schedule

6.3 REPORTS FROM JIRA

Jira is a software application used for issue tracking and project management. The tool, developed by the Australian software company Atlassian, has become widely used by agile development teams to track bugs, stories, epics, and other tasks.

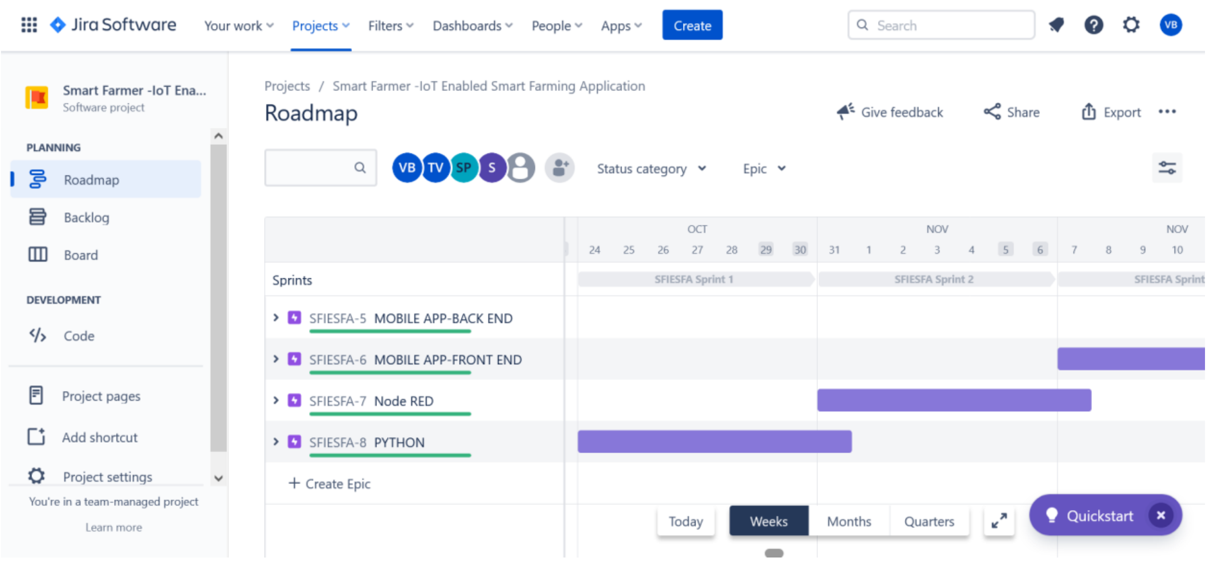


Fig 6.3.1 Report from JIRA

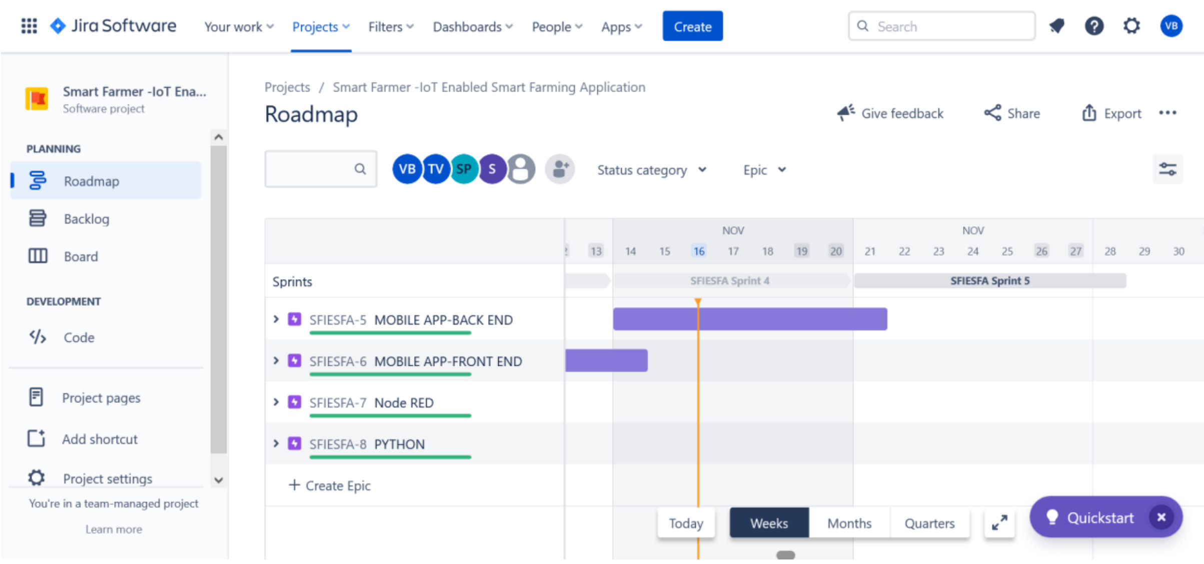


Fig 6.3.2 Report from JIRA

CHAPTER 7

CODING & SOLUTIONING

7.1 FEATURE 1

After installation of python version3.6.2 ,we have to install wiotp.sdk package to connect with IBM Watson IoT platform.

To install the package open the command prompt window and type the command as **pip install wiotp.sdk.**

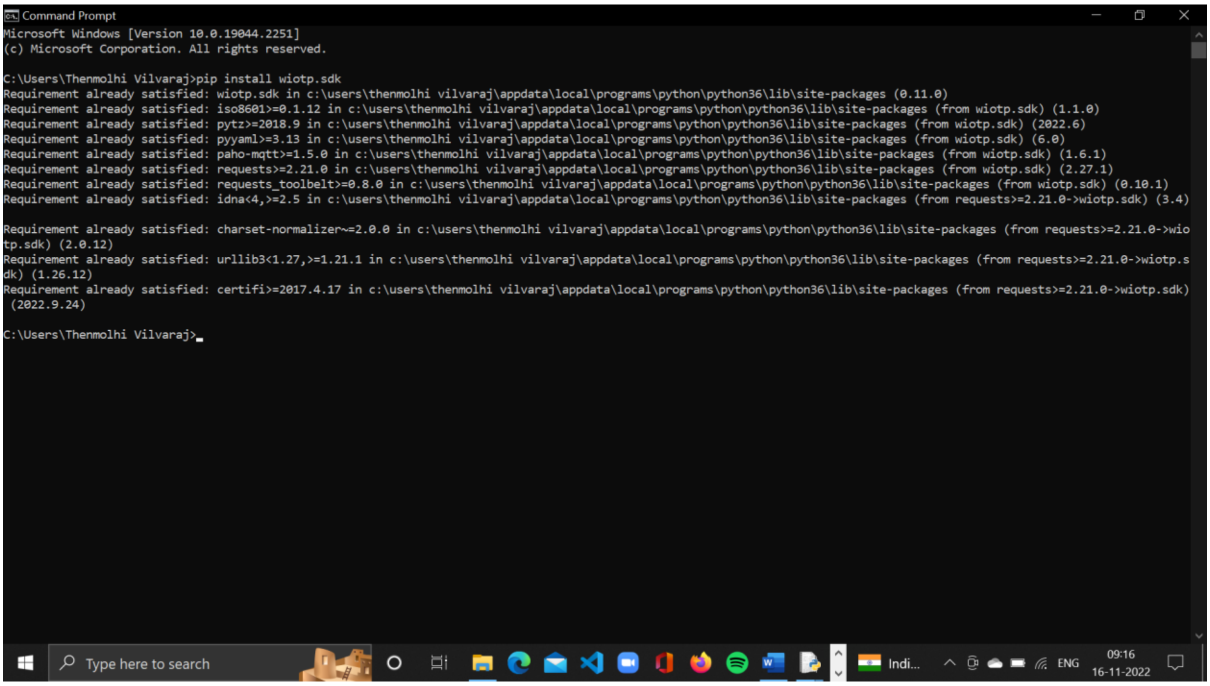


Fig 7.1.1 Installing wiotp.sdk

7.2 FEATURE 2

Python code:

```
import wiotp.sdk.device

import time

import os

import datetime

import random

myConfig={

    "identity":{

       "orgId":"92zbfc",

        "typeId":"ESP32",

       "deviceId":"1234"

        },

       "auth":{

          "token":"12345678"

           }

    }

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

client.connect()

def myCommandCallback(cmd):
```

```
    printf("Message received from IBM IoT
platform:%s"%cmd.daata['command'])
    m=cmd.data['command']
    if(m=="motoroff"):
        print("motor is switched on")
    elif(m=="motoroff"):
        print("motor is switched off")
    print("")
while True:
    soil=random.randint(14,20)
    temp=random.randint(21,37)
    hum=random.randint(60,80)
    soil_ph=random.randint(5,8)
    Co2=random.randint(1000,1300)

myData={'soil_moisture':soil,'temperature':temp,'humidity':hum,'soil_ph':soil_p
h,'Co2':Co2}
    client.publishEvent(eventId="status",msgFormat="json",data=myData,qos=0,
onPublish=None)
    print("published data successfully: %s",myData)
      time.sleep(5)
    client.commandCallback=myCommandCallback
client.disconnect()
```

Output:

Wiotp.sdk library is imported and we are using IBM Watson IoT platform's device details such as organisation id , device id , device type , authentication token.

Parameters like Soil moisture, temperature, humidity, Ph level, co2 are to be monitored by generating random values which decides the switching of the Motor, Delay is given in general.

In the Output screen you can visualize the connectivity to the IBM IoT platform and the values will be generated.

Fig 7.2.1 Output of the code

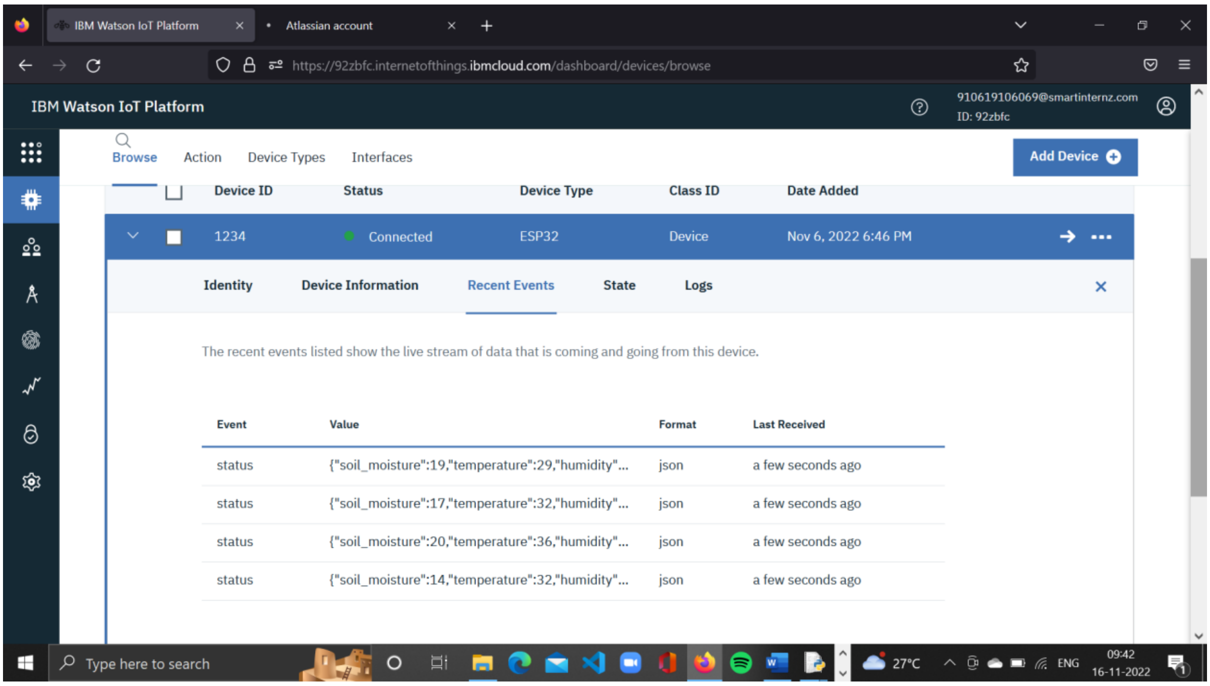
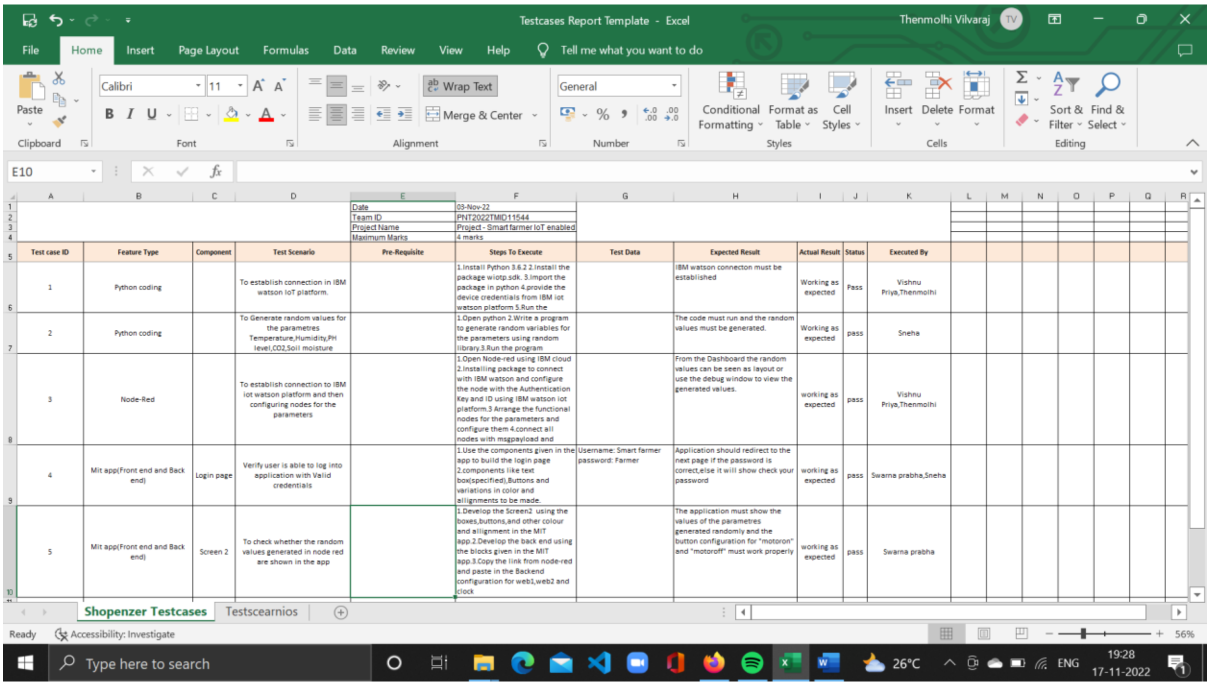


Fig 7.2.2 Output of code in IBM Watson IoT Platform

CHAPTER 8

TESTING

8.1 TEST CASES



Date	03-Nov-22
Team ID	PNT2022TMID11544
Project Name	Project - Smart farmer IoT enabled
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Executed By
1	Python coding		To establish connection in IBM watson IoT platform.		1.Install Python 3.6.2 2.Install the package wiotp.sdk. 3.Import the package in python 4.provide the device credentials from IBM iot watson platform 5.Run the		IBM watson connecton must be established	Working as expected	Pass	Vishnu Priya,Thenmolhi
2	Python coding		To Generate random values for the parametres Temperature,Humidity,PH level,CO2,Soil moisture		1.Open python 2.Write a program to generate random variables for the parameters using random library.3.Run the program		The code must run and the random values must be generated.	Working as expected	pass	Sneha
3	Node-Red		To establish connection to IBM iot watson platform and then configuring nodes for the parameters		1.Open Node-red using IBM cloud 2.Installing package to connect with IBM watson and configure the node with the Authentication Key and ID using IBM watson iot platform.3 Arrange the functional nodes for the parameters and configure them 4.connect all nodes with msgpayload and		From the Dashboard the random values can be seen as layout or use the debug window to view the generated values.	working as expected	pass	Vishnu Priya,Thenmolhi
4	Mit app(Front end and Back end)	Login page	Verify user is able to log into application with Valid credentials		1.Use the components given in the app to build the login page 2.components like text box(specified),Buttons and variations in color and allignments to be made.	Username: Smart farmer password: Farmer	Application should redirect to the next page if the password is correct,else it will show check your password	working as expected	pass	Swarna prabha,Sneha
5	Mit app(Front end and Back end)	Screen 2	To check whether the random values generated in node red are shown in the app		1.Develop the Screen2 using the boxes,buttons,and other colour and allignment in the MIT app.2.Develop the back end using the blocks given in the MIT app.3.Copy the link from node-red and paste in the Backend configuration for web1,web2 and clock		The application must show the values of the parametres generated randomly and the button configuration for "motoron" and "motoroff" must work properly	working as expected	pass	Swarna prabha

Fig 8.1.1 Test Case

8.2 USER ACCEPTANCE TESTING

The purpose of user acceptance testing is to briefly explain the test coverage and open issues of the product.

T

1. PurposeofDocument

The purpose of this document is to briefly explain thetestcoverageandopenissuesofthe [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2.TestCaseAnalysis

This report shows the number of test cases that have passed,failed,and untested

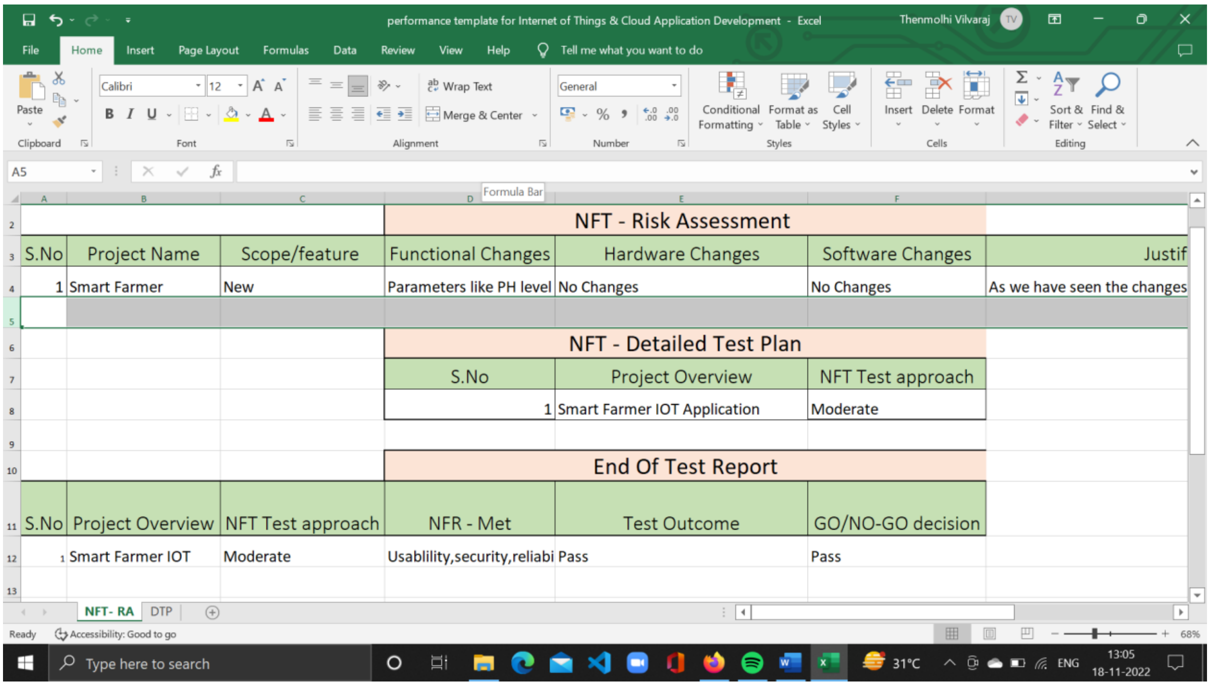
Section	TotalCases	Not Tested	Fail	Pass
Python coding	3	0	0	3
Node-Red	4	0	0	4
Mit app(Front end and Back end) screen1	3	0	0	3
Mit app(Front end and Back end) screen2	4	0	0	4
FinalReportOutput	4	0	0	4

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS

Smart farmer IoT application project has a better future scope as its functional and non functional characteristics are scalable .



NFT - Risk Assessment

S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Justif
1	Smart Farmer	New	Parameters like PH level	No Changes	No Changes	As we have seen the changes

NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach
1	Smart Farmer IOT Application	Moderate

End Of Test Report

S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision
1	Smart Farmer IOT	Moderate	Usablility,security,reliabi	Pass	Pass

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

Increased Production

Optimized crop treatment such as accurate planting, watering, pesticide application and harvesting directly affects production rates.

Water Conservation

Weather predictions and soil moisture sensors allow for water use only when and where needed.

Real-Time Data and Production Insight

Farmers can visualize production levels, soil moisture, sunlight intensity and more in real time and remotely to accelerate decision making process.

Lowered Operation Costs

Automating processes in planting, treatment and harvesting can reduce resource consumption, human error and overall cost.

Increased Quality of Production

Analysing production quality and results in correlation to treatment can teach farmers to adjust processes to increase quality of the product.

Accurate Farm and Field Evaluation

Accurately tracking production rates by field over time allows for detailed predicting of future crop yield and value of a farm.

Improved Livestock Farming

Sensors and machines can be used to detect reproduction and health events earlier in animals. Geofencing location tracking can also improve livestock monitoring and management.

Reduced Environmental Footprint

All conservation efforts such as water usage and increased production per land unit directly affect the environmental footprint positively.

Remote Monitoring

Local and commercial farmers can monitor multiple fields in multiple locations around the globe from an internet connection. Decisions can be made in real-time and from anywhere.

Equipment Monitoring

Farming equipment can be monitored and maintained according to production rates, labour effectiveness and failure prediction.

DISADVANTAGE

Lack of Infrastructure

Even if the farmers adopt IoT technology they won't be able to take benefit of this technology due to poor communication infrastructure. Farms are located in remote areas and are far from access to the internet. A farmer needs to have access to crop data reliably at any time from any location, so connection issues would cause an advanced monitoring system to be useless.

High Cost

Equipment needed to implement IoT in agriculture is expensive. However sensors are the least expensive component, yet outfitting all of the farmers' fields to be with them would cost more than a thousand dollars. Automated machinery cost more than manually operated machinery as they include cost for farm management software and cloud access to record data. To earn higher profits, it is significant for farmers to invest in these technologies however it would be difficult for them to make the initial investment to set up IoT technology at their farms.

Lack of Security

Since IoT devices interact with older equipment they have access to the internet connection, there is no guarantee that they would be able to access drone mapping data or sensor readouts by taking benefit of public connection. An enormous amount of data is collected by IoT agricultural systems which is difficult to protect. Someone can have unauthorized access IoT providers database and could steal and manipulate the data.

CHAPTER 11

CONCLUSION

This project gave us the chance to learn new technologies and work with new tools, this was a real proof that IBM has taught us to be long-life learners and to master self-learning before teaching us other class materials. Of course, this project is a combination of what we learned from all the training and orientation sessions conducted by IBM, altogether with what we learned from other disciplines and also by ourselves about IoT.

In general, the project was successful and worked properly and succeeded in delivering the prototype on due time. We are proud and happy for this achievement especially that this our first theoretically, practically online stimulated project . It enabled us to get concrete results and to realize that we can indeed build products that would be beneficial in real life and that we can customize it upon demand as future projects.

CHAPTER 12

FUTURE SCOPE

We can add other Arduino components or increasing sensors for more features, and to fetch more data that can be collected. An artificial intelligent system can be added to predict the production of goods. A GPS module can be integrated to enhance this Agriculture IoT Technology to full-fledged Agriculture Precision ready product.

CHAPTER 13

APPENDIX

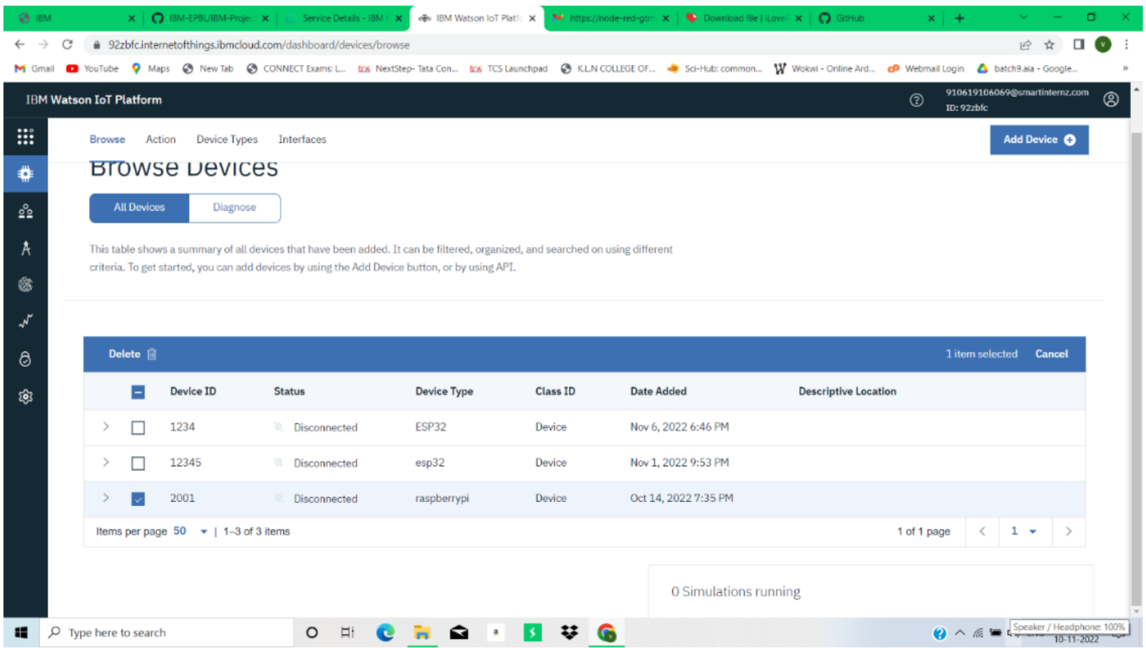
13.1 SOURCE CODE


ibm python code.py - C:\Users\Thenmolhi Vilvaraj\AppData\Local\Programs\Python\Python36\ibm python code.py (3.6.2)

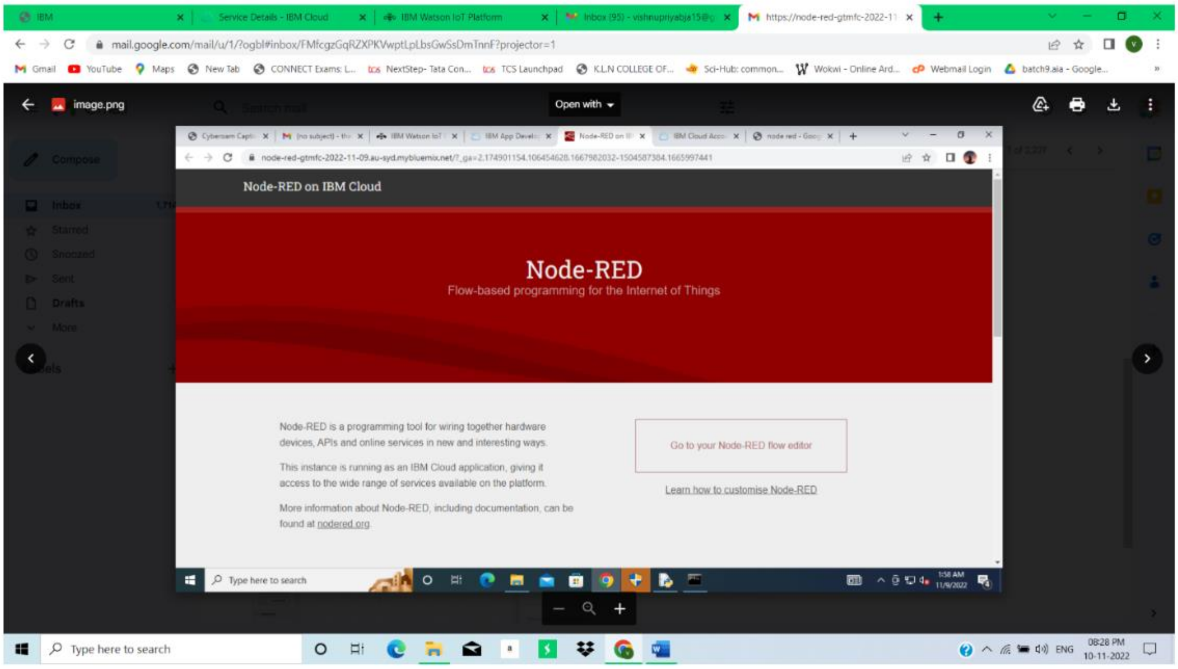
```python
import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig={
    "identity":{
        "orgId":"92zbfc",
        "typeId":"ESP32",
        "deviceId":"1234"
        },
        "auth":{
            "token":"12345678"
            }
    }
client=wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
client.connect()
def myCommandCallback(cmd):
    printf("Message received from IBM IoT platform:%s" %cmd.daata['command'])
    m=cmd.data['command']
    if(m=="motoroff"):
        print("motor is switched on")
    elif(m=="motoroff"):
        print("motor is switched off")
    print("")
while True:
    soil=random.randint(14,20)
    temp=random.randint(21,37)
    hum=random.randint(60,80)
    soil_ph=random.randint(5,8)
    Co2=random.randint(1000,1300)
    myData={'soil_moisture':soil,'temperature':temp,'humidity':hum,'soil_ph':soil_ph,'Co2':Co2}
    client.publishEvent(eventId="status",msgFormat="json",data=myData,qos=0,onPublish=None)
    print("published data successfully: %s",myData)
    time.sleep(5)
    client.commandCallback=myCommandCallback
client.disconnect()
```

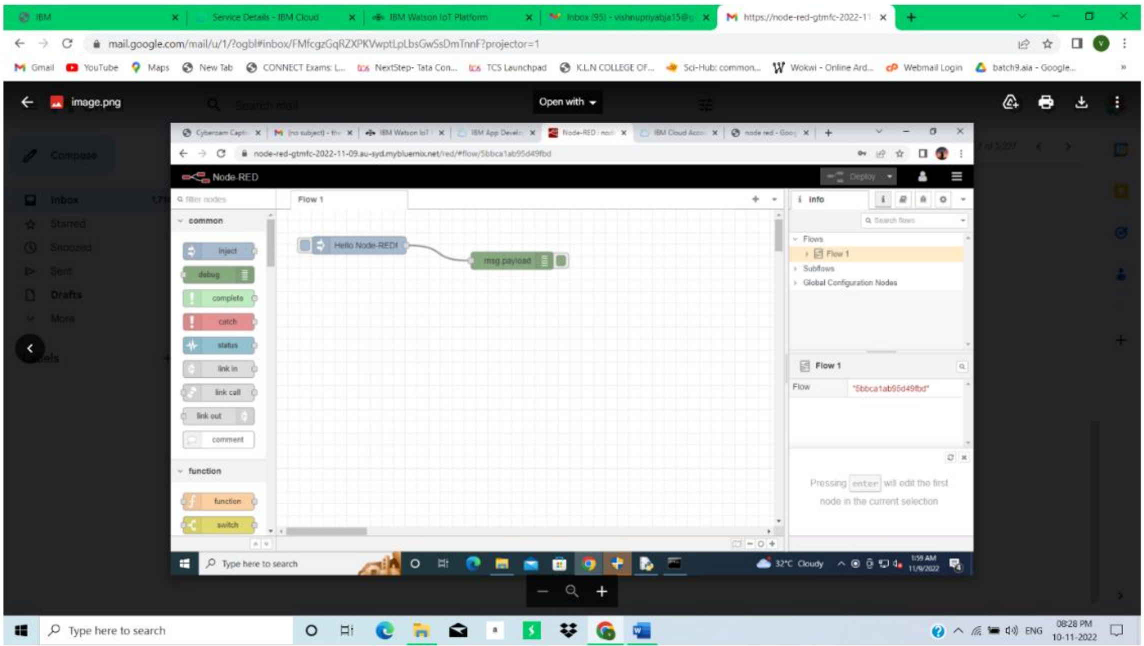
13.2 SOFTWARES USED

IBM Watson IoT Platform

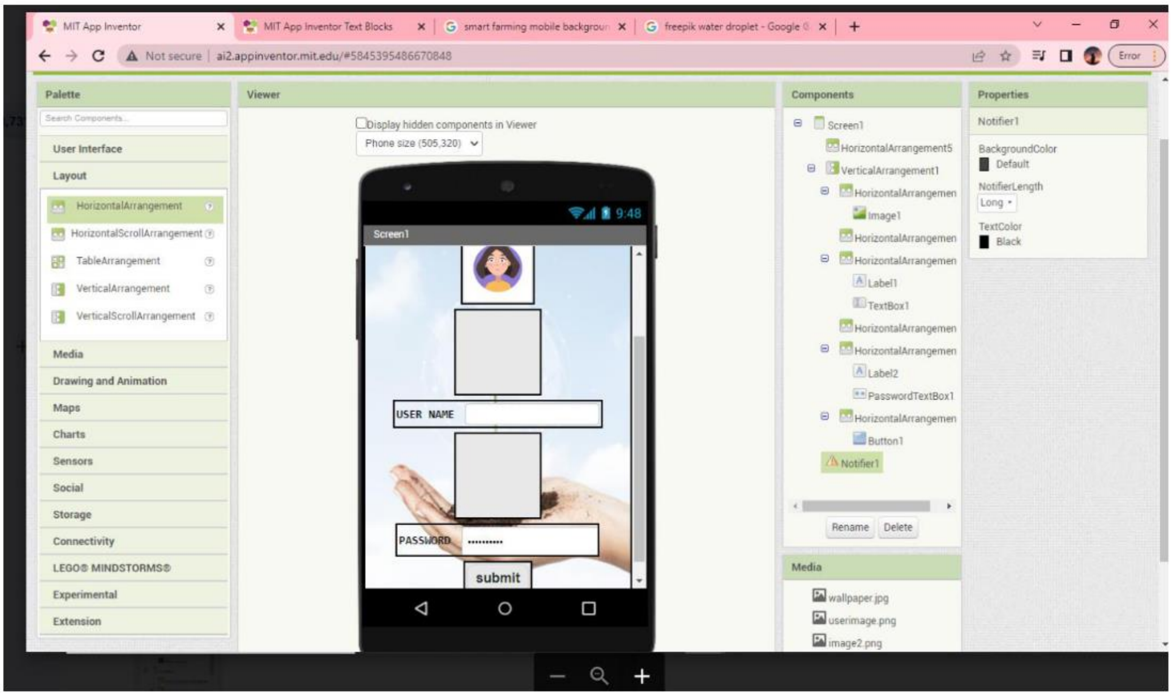


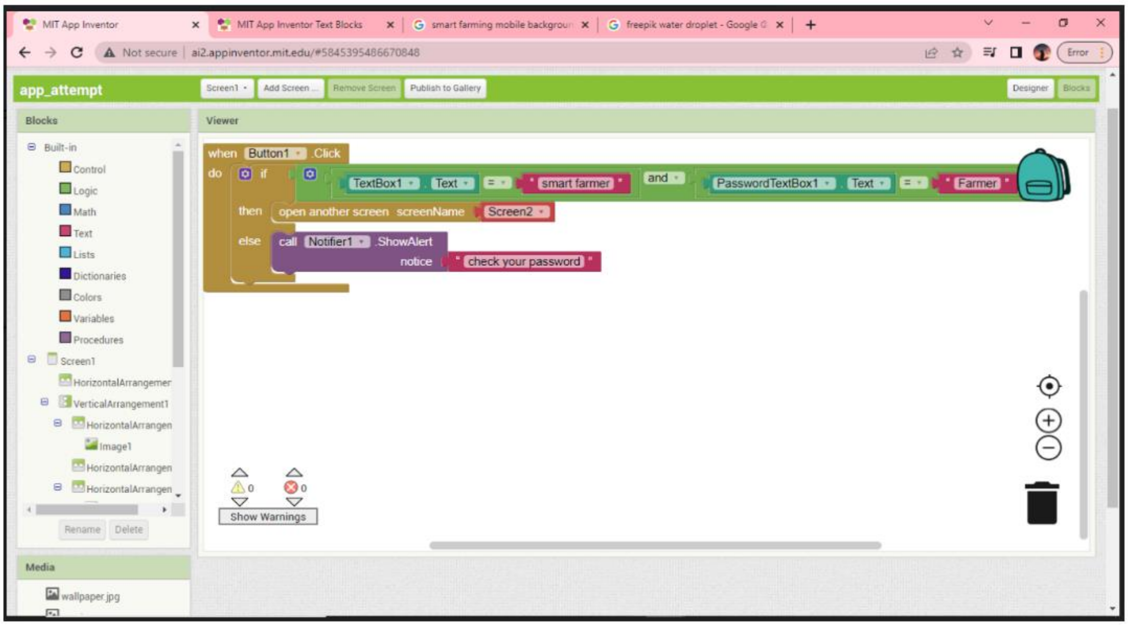
NodeRED


Node-RED on IBM Cloud
Node-RED
Flow-based programming for the Internet of Things
Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.
This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.
More information about Node-RED, including documentation, can be found at nodered.org.
Go to your Node-RED flow editor
Learn how to customise Node-RED


Node-RED
Flow 1
Hello Node-RED!
msg.payload
inject
debug
complete
catch
status
link in
link call
link out
comment
function
switch

MIT APP INVENTOR





13.2 GitHub and Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-24552-1659944427

https://youtu.be/kbHSVkiWDRo