SMARTFARMER - IOT ENABLED SMART FARMING APPLICATION

IBM PROJECT REPORT

Submitted by

SUBITHRA R - 613019104083

SWARNASHRI S - 613019104086

SNEGA M - 613019104075

TAMILSELVI N - 610919104062

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

VIVEKANANDHA COLLEGE OF TECHNOLOGY FOR WOMEN,TIRUCHENGODE,NAMAKKAL

ANNA UNIVERSITY : CHENNAI – 600 025

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Project Overview

People who use the internet of things can live and work more intelligently and have total control over their life. IoT is crucial to business in addition to providing smart home automation devices. With the help of IoT, organizations can see in real time how their systems actually function, gaining insights into anything from equipment performance to supply chain and logistics activities.This paper presents an Internet of Things (IoT) based Smart Farmer - IoT Enabled Smart Farming Application.The system is implemented using anultrasonic sensor which is connected to Arduino UNO as to monitor the water level. In this system, the depth level will be sent via Arduino Ethernet Shield with an Internet connection to the IBM Cloud.

1.1 Purpose

Since the dawn of human civilization, agriculture has been considered to be the most significant activity. Traditional irrigation techniques, such as flood irrigation and overhead sprinkler irrigation, are not very effective. They waste a significant amount of water, and the excessive moisture in the soil can also encourage the growth of diseases like fungus. Since water is a valuable resource and indirectly supports the survival of the farm, an automated irrigation system is crucial. This need is anticipated to rise in the coming years due to population growth. In an automation system, sensors are usedto monitor the crop's access to water, and controlled irrigation is used to water the crop as needed.

IoT is possible because of adequate power supply and internet connectivity. The term "Internet of Things" is commonly used to describea framework in which sensors are connected to objects and allow these objects to share their "digital voice" with the outside world via an internet connection.

Based on a wireless sensor network, this system created an automated irrigation system for farmers. The soil's moisture content, humidity, and temperature are all continuously monitored Depending on the soil's moisture content, the irrigation system either starts or stops. This method suggests a low cost data collecting systembased on moisture sensors

CHAPTER -2

LITERATURE SURVEY

2.1 Existing Problem

S.NO	TITLE	AUTHOR AND YEAROF PUBLICATIONS	METHODOLOGY USED
1.	Mobile Integrated Smart Irrigation Management and Monitoring SystemUsing IOT	S. Vaishali et.al, 08 February 2018	In order to control and monitor the irrigation process, smart and automated irrigation systemis developed, Implemented and tested. There is a need for automated irrigation system because it is simple and easy to install. This system uses values ON and OFF to control water motor.Python programming language is been used for automation purpose.
2.	IoT Based Smart Irrigation Monitoring And Controlling System	Shweta B. Saraf et.al ,15 January 2018	In this paper proposedsystem is based on IoT that uses real time input data. Smart farm irrigation system uses android phone for remote monitoring and controlling of drips throughwireless sensor network. Zigbee is used for communication between sensor nodes and base station. Real time sensed data handling and demonstration on the serveris accomplished using web based java graphical user interface.

3	Smart waste collection monitoring and alert system via IoT	Zainal Hisham Che Soh et.al, 24 June 2019	The system is implemented using an ultrasonic sensor which is connected to Arduino UNO as to monitor waste bin garbage level. In this system, waste bin depth level will be sent via Arduino Ethernet Shield with an Internet connection to the Ubidots IoT Cloud.The Ubidots store the collected waste bin level data into IoT database and display the waste bin depth level on online dashboard for real- time visualization.

2.2 References

1. Mobile Integrated Smart Irrigation Management and Monitoring System Using IOT
   Date of Conference: 06-08
   April 2017Publisher: IEEE
   Date Added to IEEE Xplore: 08
   February 2018DOI:
   10.1109/ICCSP.2017.8286792
2. IoT Based Smart Irrigation Monitoring And Controlling System
   Date Added to IEEE Xplore: 15
   January 2018 ISBN Information:
   Electronic ISBN: 978-1-5090...Date
   of Conference: 19-20 May 2017
   INSPEC Accession Number: 17504411
3. Smart Waste Collection Monitoring and Alert System via IoT
   Date Added to IEEE Xplore:
   24 June 2019DOI:
   10.1109/ISCAIE.2019.874376
   Print on Demand(PoD) ISBN: 978-1-5386-854

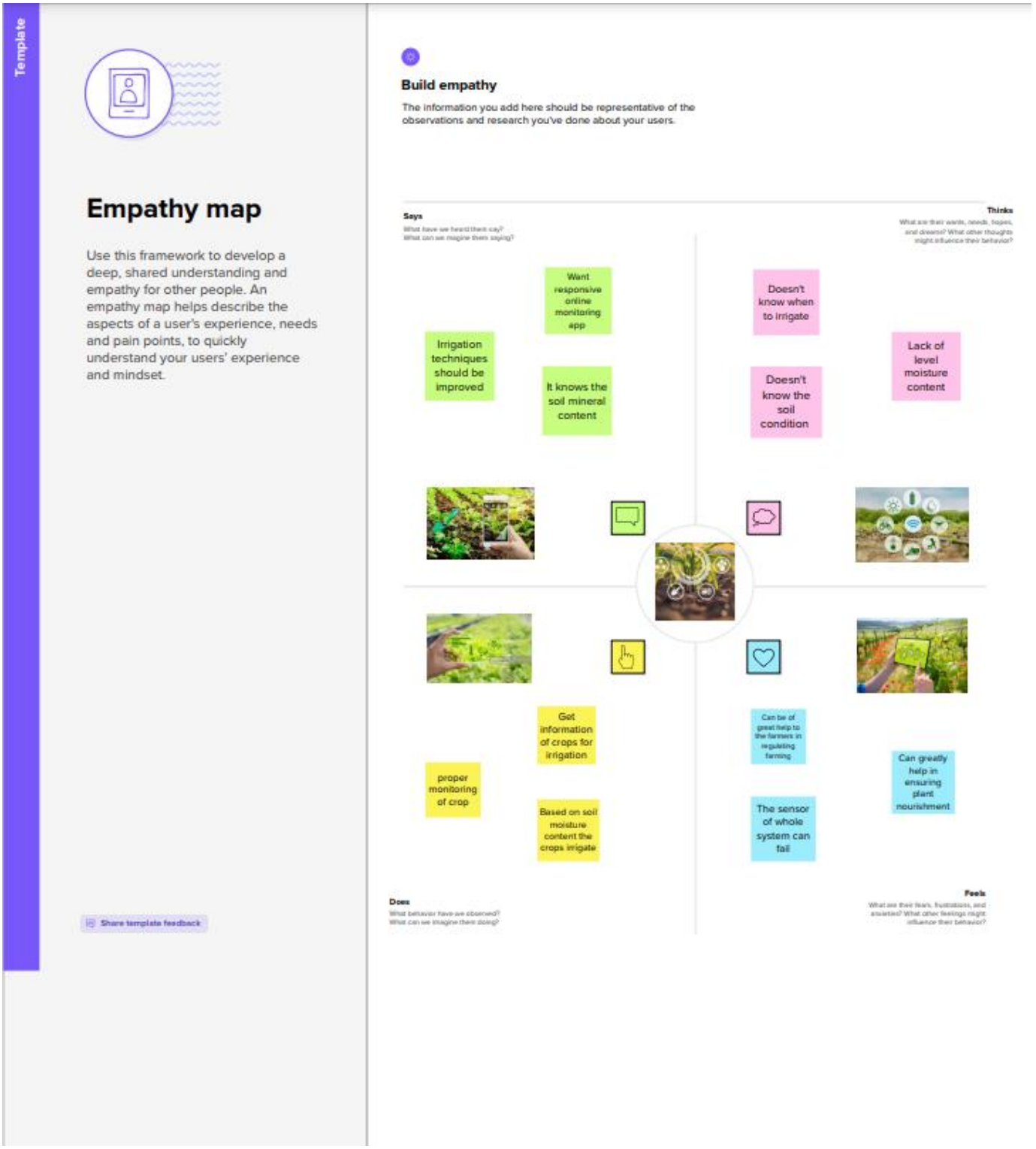
2.3 Problem Statement Definition

Farmers are under pressure to produce more food and use less energy and water in the process. A remote monitoring and control system will help farmers deal effectively with these pressures Ideally, Each feld should get just the right amount of water at just the right time.Most of the rural area people cant implement the IOT devices because of they dont know about the device to use. This idea is also to focus on parameters such as temperature and soil moisture. The main objective of this project is to control reduce the water supply,save the crops and monitor the plants. The system is implemented using an ultrasonic sensor which is connected to Arduino UNO as to monitor Farm Field level.In this system, Farm Field depth level will be sent via Arduino Ethernet Shield with an Internet connection to the IBM IoT Cloud. The IBM Cloud store the collected Farm field level data into IoT database and display the Farm Field depth level on online dashboard for real-time visualization. The IBM Event manager invoke a notification alert to the Owner of the farmer mobile phone via a SMS when the farm field is nearly filled and It automatically Switch Off the Water Motor

CHAPTER 3

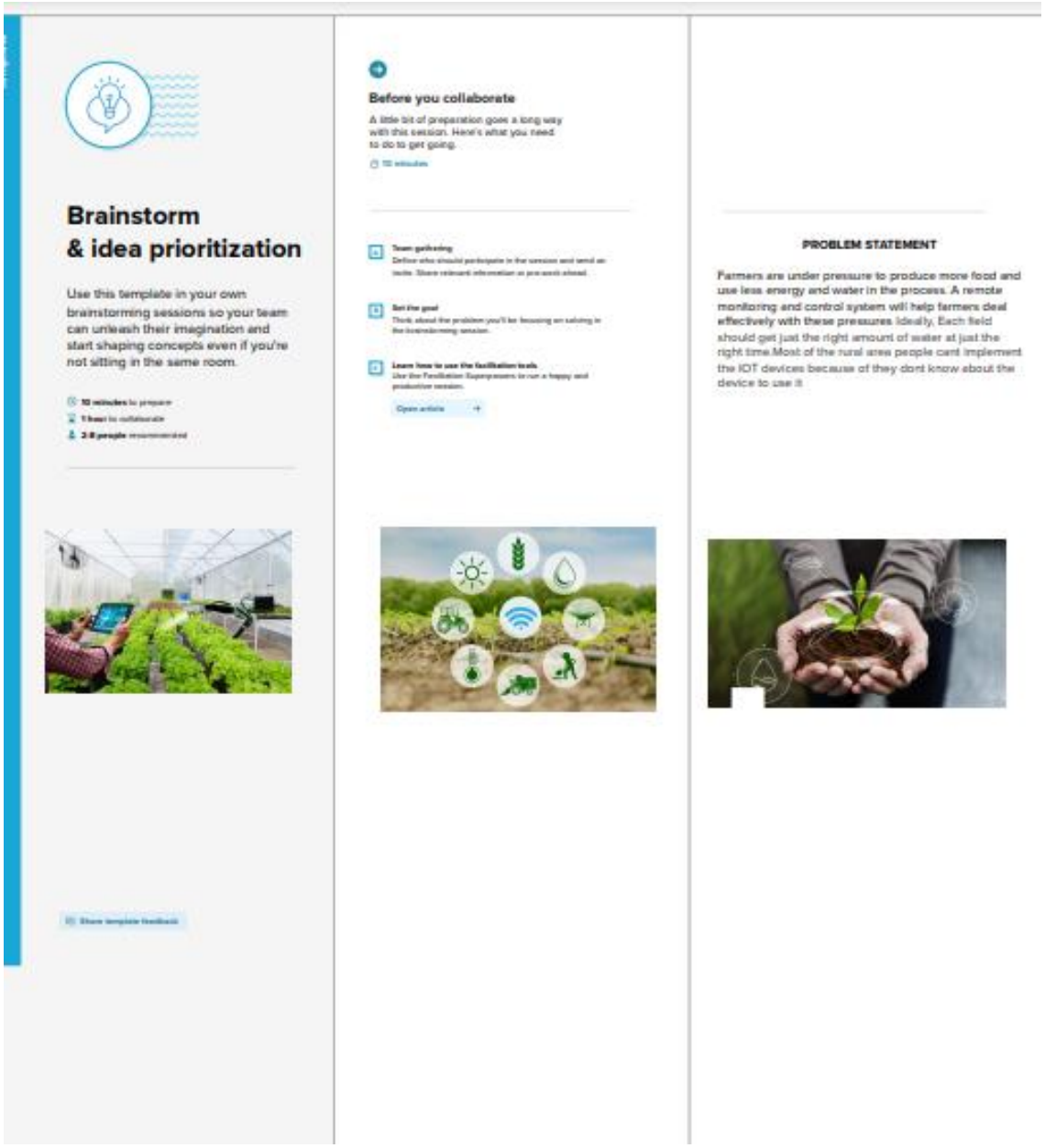
IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping

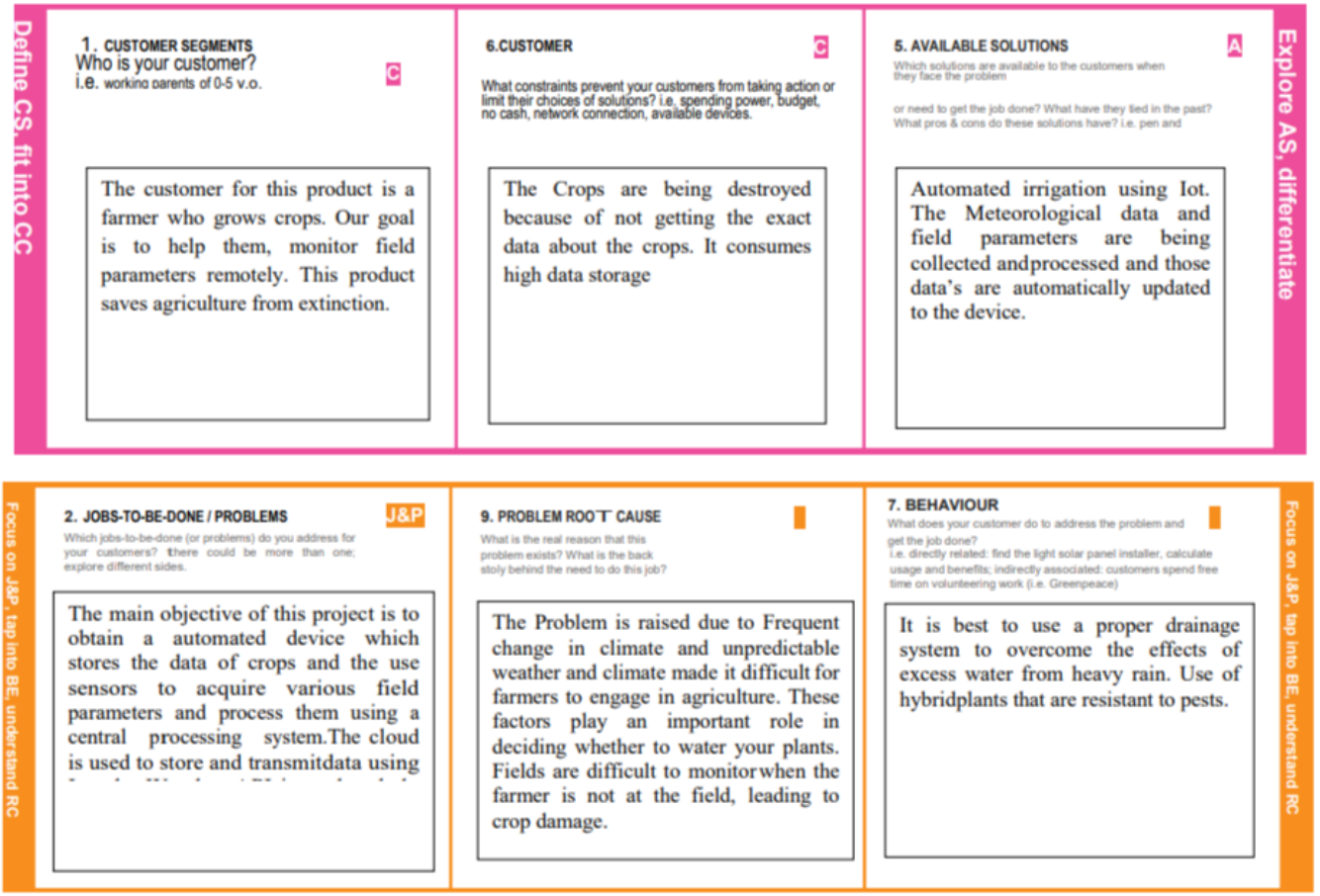
Step-3: Idea Prioritization

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	Farmers are under pressure to produce more food and use less energy and water in the process. A remote monitoring and control system will help farmers deal effectively with these pressures Soil erosion and soil nutrient loss,product satisfaction customers adaptation to climate, usage of harmful fertilizer and manure andpesticides are major problem faced by farmers For taking agriculture to the next generation, only fresh water can be used for farming. Farmers are more affected and annoyed by theabove factors.
2.	Idea / Solution description	The IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, temperature, and humidity using some sensors. Farmers can monitor all the sensor parameters by using a web or mobile application even if the farmer is not near his field. Watering the crop is one of the important tasks for the farmers. They can make the decision whether to water the crop or postpone it by monitoring the sensor parameters and controlling the motor pumps from the mobile application itself.
3.	Novelty / Uniqueness	Building a community of farmers around the globe will definitely do great, where they can find lots and lots of information about crops and plantations directly from the other farmersand the agricultural experts around the globe.

4.	Social Impact / Customer Satisfaction	The good thing is that IoT instructs or alerts them to do the amount of work at the right time. So that the yielding will be more and good, it also reduces the attention and time required to the field, which makes the Customer Satisfaction. On the other hand the main disadvantage is that IoT reduces the number of laborers and their wages, as a resultmany people may lose their work.
5.	Business Model (Revenue Model)	Many agricultural products like fertilizers, pesticides, manure, and field equipment canalso be promoted in the form of ads. A small amount of subscription fee can also be collected from the farmers.
6.	Scalability of the Solution	The scalability of the above proposed solution is not limited. Here a lot of sensors and analyzing tools and algorithms can be integrated to provide the best experience.

3.4 Problem Solution Fit



CHAPTER 4

REQUIREMENT ANALYSIS

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
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 cropdiseases
FR-4	Manage Modules	Manage Roles of User Manage User permission
FR-5	Check whether details	Temperature details Humidity details
FR-6	Data Management	Manage the data of weather conditions Manage the data of crop conditions Manage the data of live stock conditions

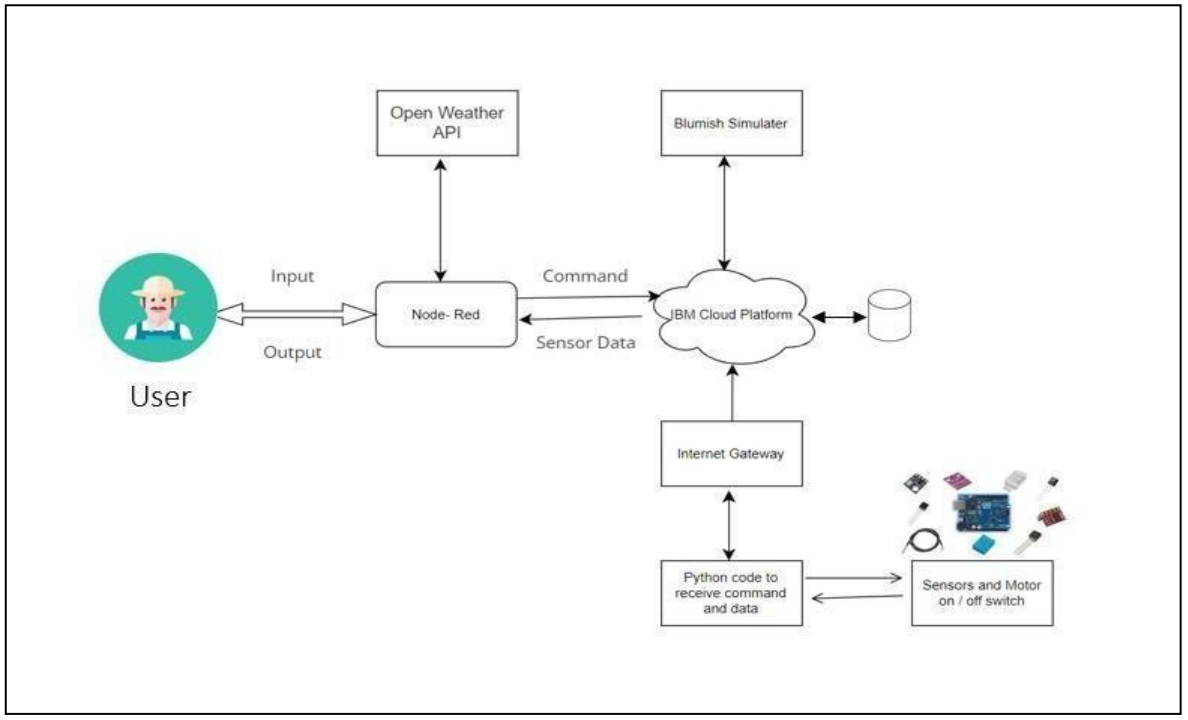
Non-functional Requirements:

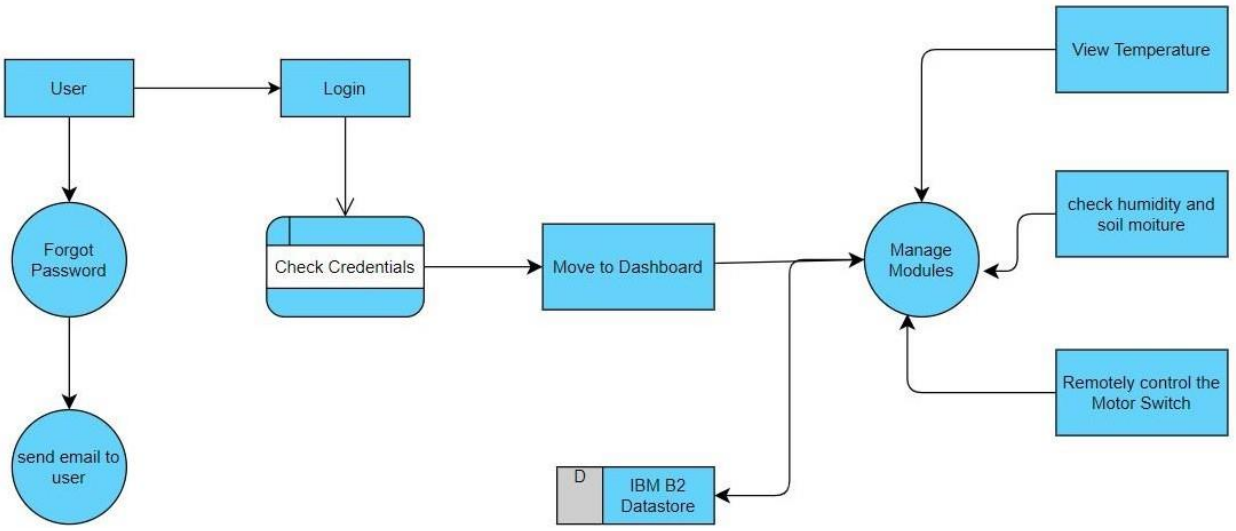
Following are the non-functional requirements of the proposed solution.

S No.	Non-Functional Requirement	Description
NFR-1	**Usability**	User friendly guidelines for users to avail the features. Most simplistic user interface for ease of use and it includes easy learn ability, efficiency in use, remember ability, lack of errors in operation and subjective pleasure
NFR-2	**Security**	All the details about the user are protected from unauthorized access. Detection and identification of any misfunctions of sensors. Sensitive and private data must be protected from their production until the decision-making and storage stages
NFR-3	**Reliability**	Implementing Mesh IoT Networks Building a Multi-layered defence for IoT Networks and This model uses dedicated and shared protection schemes to avoid farm service outages
NFR-4	**Performance**	The use of modern technology solutions helps to achieve the maximum performances thus resulting in better quality and quantity yields. the idea of implementing integrated sensors with sensing soil and environmental in farming will be more efficient for overall monitoring.
NFR-5	**Availability**	This app is available for all platforms Then Automatic adjustment of farming equipment madepossible by linking information like crops/weather and equipment to auto-adjust temperature,humidity
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.

CHAPTER 5
PROJECT DESIGN

5.1 Data Flow Diagram





5.2 Solution and Technical Architecture

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

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

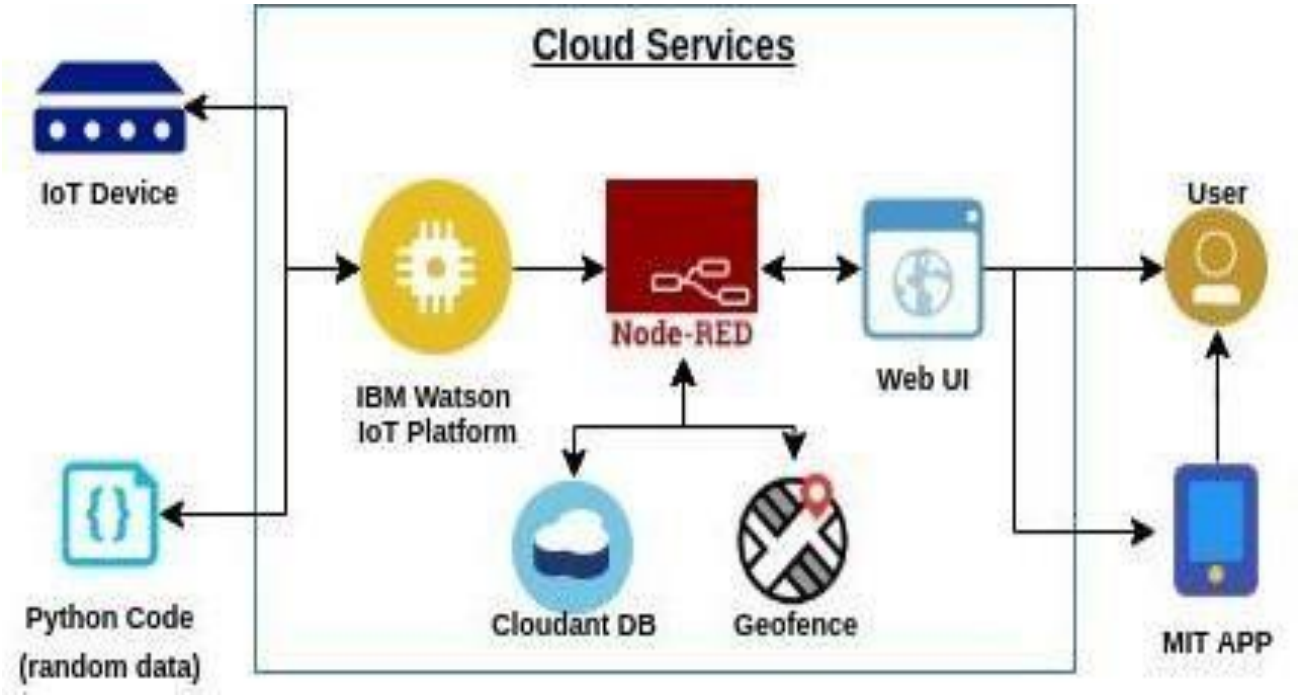


Figure 1: Architecture and data flow of the smart farming application

Table-1 : Components & Technologies:

Description	Technology
How user interacts with applicationе.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript
Logic for a process in the application	Python
Logic for a process in the application	IBM Watson service
Logic for a process in the application	IBM Watson Assistant
Data Type, Configurations etc.	MySQL
Database Service on Cloud	IBM DB2, IBM Cloudant etc.
File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
Purpose of External API used in the application	IBM Weather API, etc.
Application Deployment on LocalSystem / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry,Kubernetes, etc. **Table-2: Application Characteristics:**

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Python IDE
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	GSM module
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Node red service
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	IBM Watson IoT Platform
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	NPK Sensors

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Custoer (Mobile user)	Registration	USN-1	As a user, I can register for the applicationby entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmationemail once I have registered for the application	I can receive confirmation email &click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password		High	Sprint-1
Customer (Web user)	Dashboard	USN-5	As a User can view the dashboard, and this dashboard include the check roles of access and then move to the manage modules.	I can view the dashboardin this smart farming application system.	High	Sprint 2
		USN-6	User can remotely access the motor switch	In the smart farming app	High	Sprint 3
Administrator			As a user once view the manage modules this describes the Manage system Admins and Manage Roles of User			Sprint 2

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature review on the chosen project and information gathering using references from IEEE Papers.	25 September 2022
Prepare Empathy Map	Get an Empathy Map Canvas ready to record the user's gains and pains and also prepare list of problem statements	26 September 2022
Ideation	Create a list of them by arranging the brainstorming session, then rank the top three concepts according to their viability and significance.	27 September 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, workability of idea, business pattern, social clash and so on.	5 October 2022
Problem Solution Fit	Prepare problem - solution fit document.	8 September 2022

Solution Architecture	Prepare solution architecture document.	11 September 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application	14 October 2022
Functional Requirement	Create the project's functional requirements.	17 October 2022
Data Flow Diagrams	Create the data flow diagrams, then submit them for evaluation.	17 October 2022
Technology Architecture	Prepare the technology By using the architecture diagram.	18 October 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	22 October 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	Towards Progress

6.2 Sprint Delivery Schedule

Product Backlog, Sprint Schedule, and Estimation:

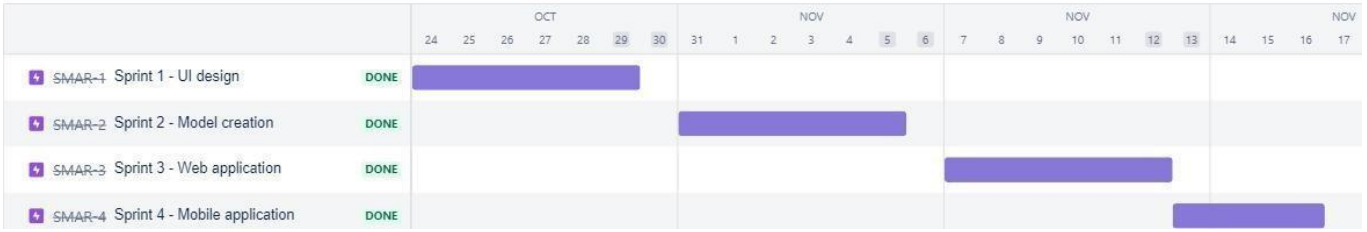
Sprint	Functional Requirement (Epic)	User Story Number	User Story /Task	Story Points	Priority	Team Member
Sprint-1	Registration (Farmer Mobile User)	UNS-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Subithra.R (Leader)
Sprint-1	Login	UNS-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Swarna shri (Member 1)

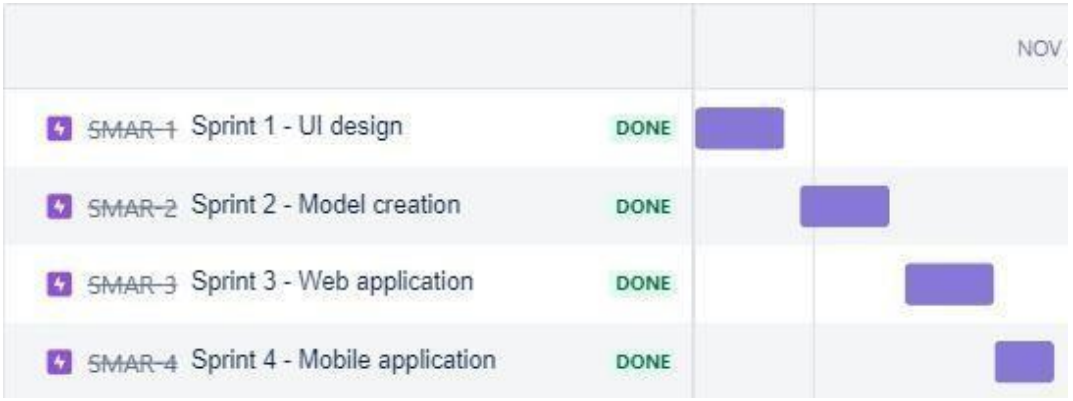
Sprint-2	User Interface	UNS-3	As a user, I can register for the application through Facebook	3	Low	Tamil selvi . N (Member 2)
Sprint-1	Data Visualization	UNS-4	As a user, I can register for the application through GMAIL	2	Medium	Snega . m (Member 3)
Sprint-3	Registration(Farmer Web User)	USN - 1	As a user, I can log into the application by entering email and password	3	High	Subithra.R (leader)
Sprint 2	Login	USN - 2	As a registered user, I need to easily login log into my registered account via the web page in minimum time	3	High	Swarna shri
Sprint 4	Web UI	USN - 3	As a user, I need to have a friendly user interface to easily view and access the resources	3	Medium	Tamil selvi .N

Sprint - 1	Registration (Chemical Manufacturer - Web user)	USN - 1	As a new user, I want to first register using my organization email and create a password for the account.	2	High	Snega.M
Sprint - 4	Login	USN - 2	As a registered user, I need to easily log in using the registered account via the web page.	3	High	Subithra. R
Sprint - 3	Web UI	USN - 3	As a user, I need to have a user-friendly interface to easily view and access the resources.	3	Medium	Swarna shri
Sprint - 1	Registration (Chemical Manufacturer - Mobile User)	USN - 1	As a user, I want to first register using my email and create a password for the account.	1	High	Tamil Selvi . N

Sprint - 1	Login	USN - 2	As a registered user, I need to easily log in to the application.	2	Low	Snega . M

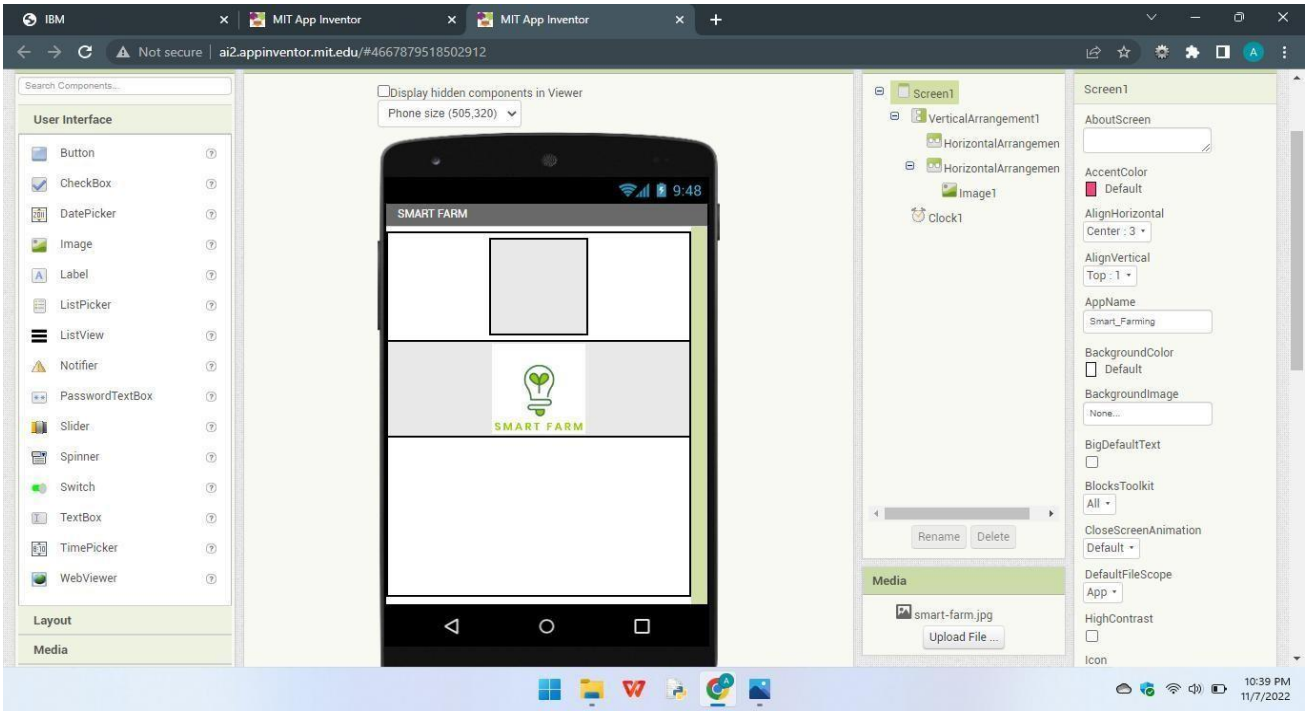
6.3 Reports from JIRA

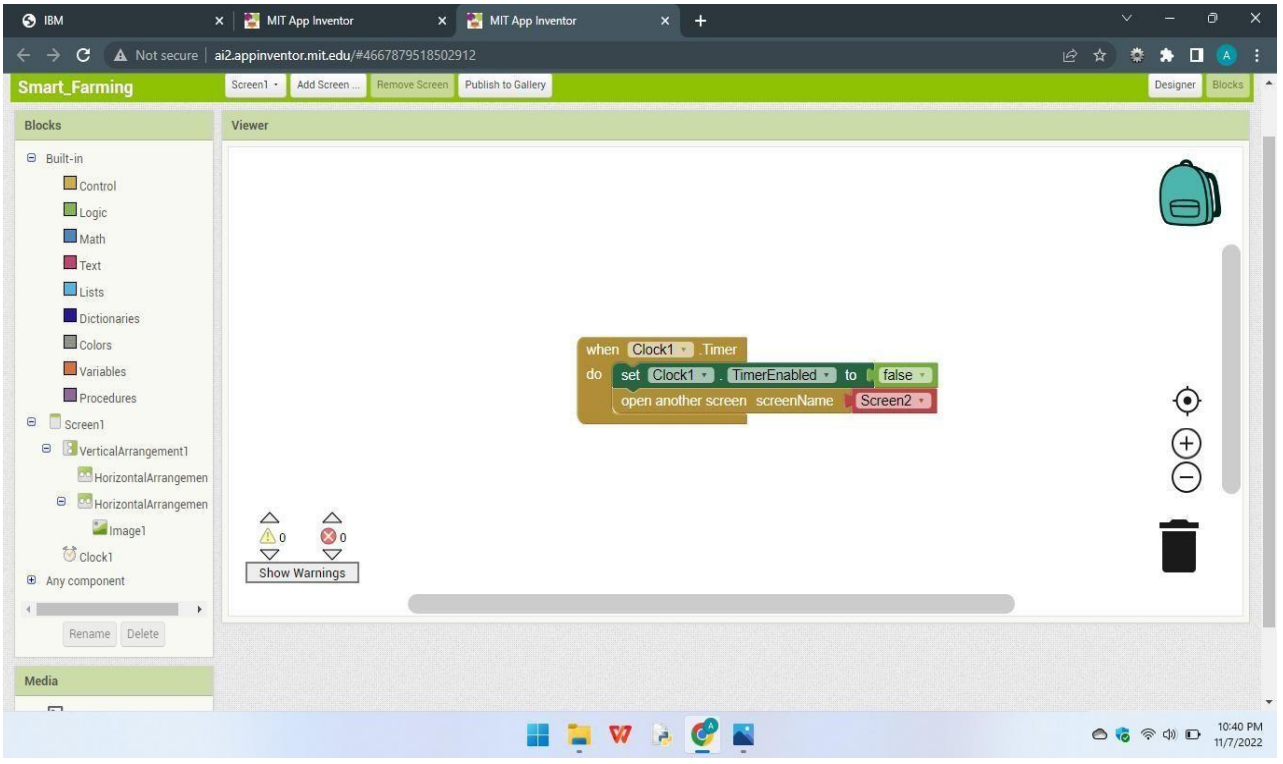


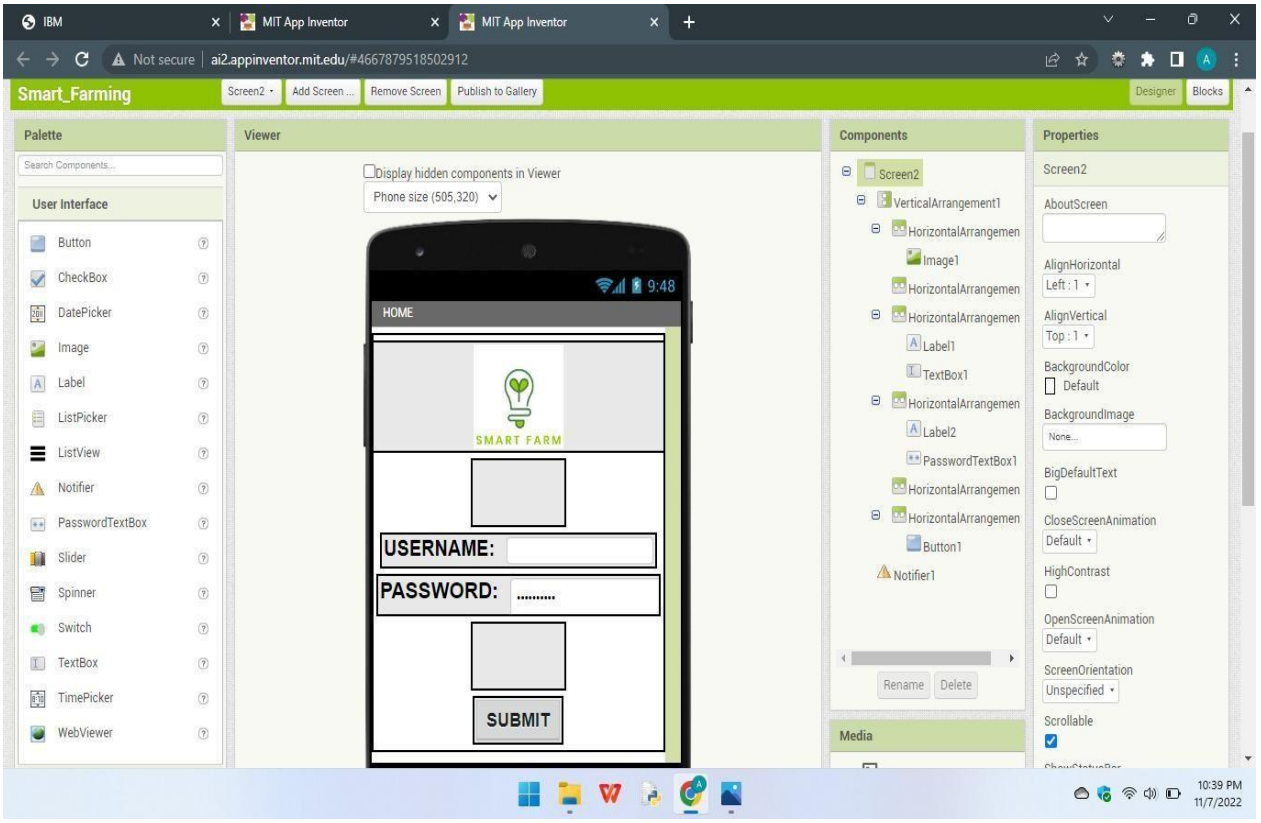


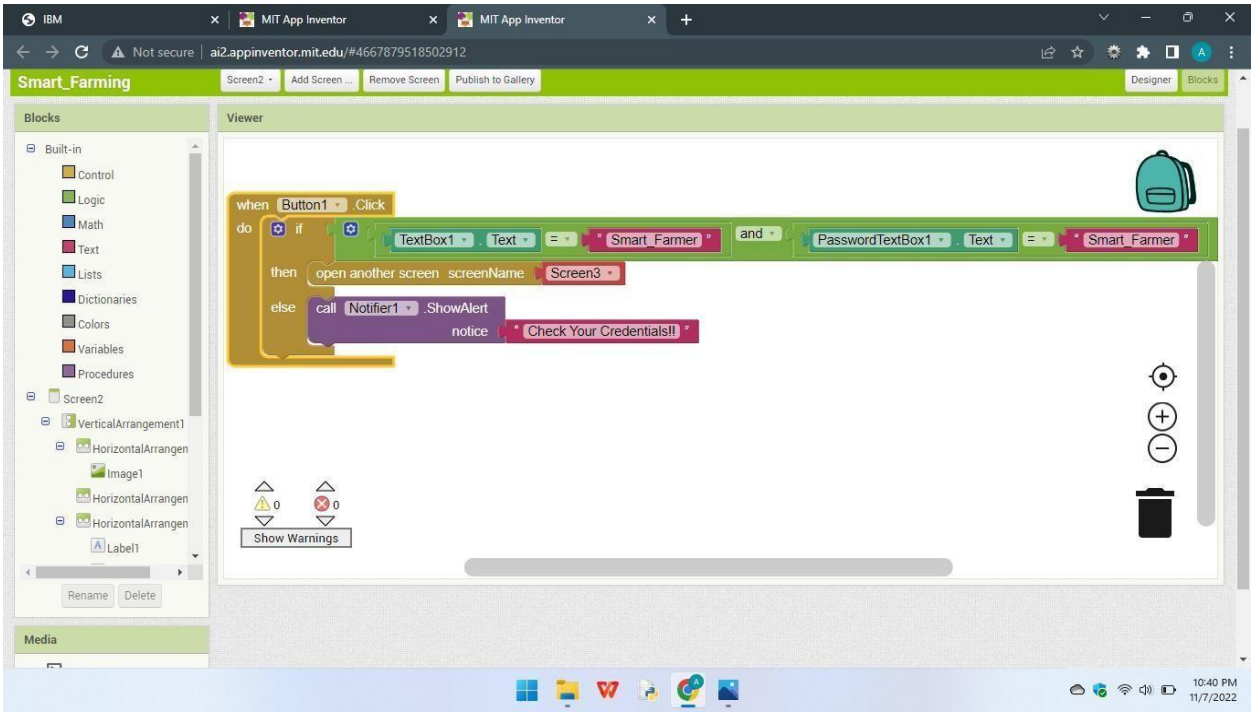
CHAPTER 7 CODING & SOLUTIONING

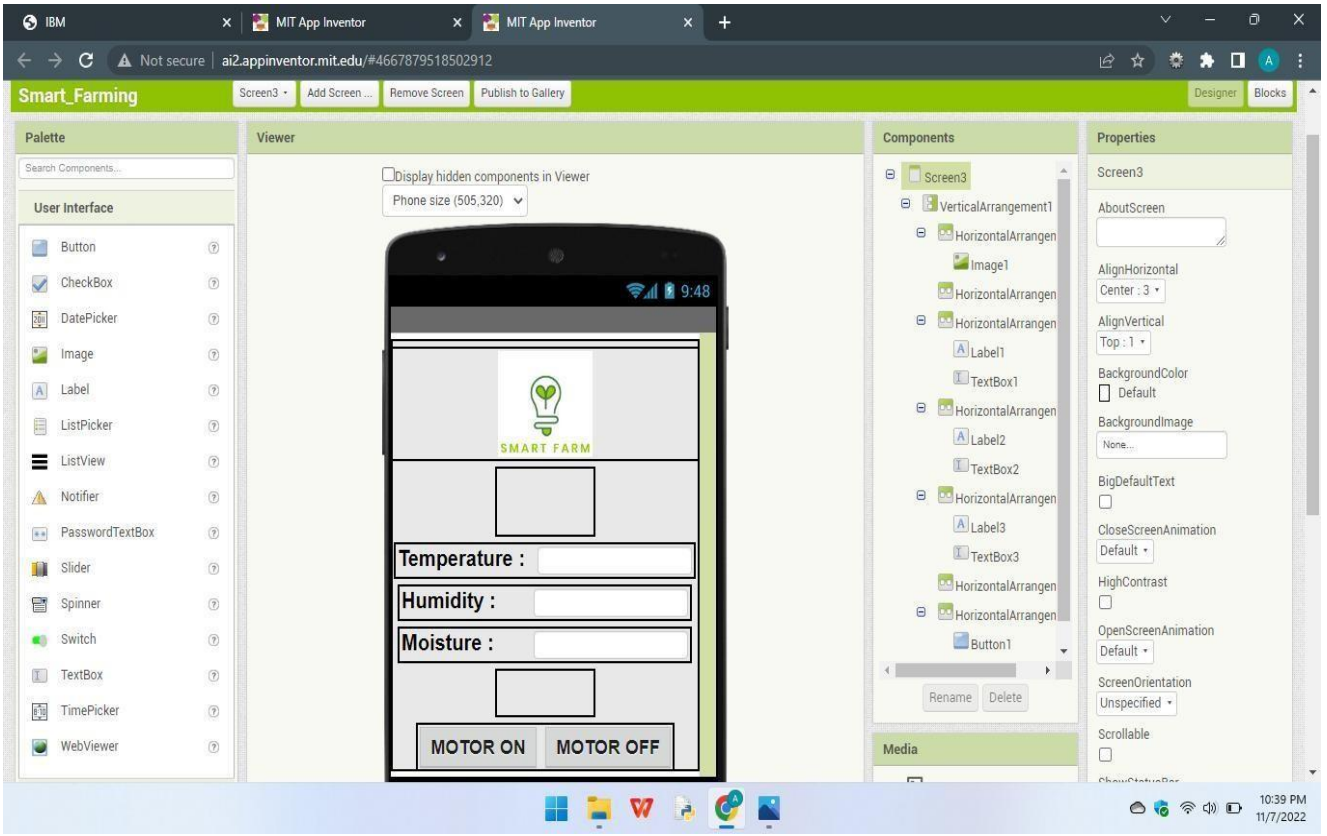
7.1 Features - Development of Sprint-1 (User Interface Design)






Smart_Farming
Screen2
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Palette
Viewer
Components
Properties
Display hidden components in Viewer
Phone size (505,320)
HOME
SMART FARM
USERNAME:
PASSWORD:
SUBMIT
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Screen2
VerticalArrangement1
Image1
Label1
TextBox1
Label2
PasswordTextBox1
Button1
Notifier1
Rename
Delete
Media
AboutScreen
AlignHorizontal
AlignVertical
BackgroundColor
BackgroundImage
BigDefaultText
CloseScreenAnimation
HighContrast
OpenScreenAnimation
ScreenOrientation
Scrollable


Smart_Farming
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen2
when Button1 .Click
do if TextBox1 . Text = " Smart_Farmer " and PasswordTextBox1 . Text = " Smart_Farmer "
then open another screen screenName Screen3
else call Notifier1 .ShowAlert
notice " Check Your Credentials!! "
Show Warnings
Rename
Delete
Media


IBM
MIT App Inventor
MIT App Inventor
Not secure
ai2.appinventor.mit.edu/#4667879518502912
Smart_Farming
Screen3
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Palette
Search Components...
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Viewer
Display hidden components in Viewer
Phone size (505,320)
9:48
SMART FARM
Temperature :
Humidity :
Moisture :
MOTOR ON
MOTOR OFF
Components
Screen3
VerticalArrangement1
HorizontalArrangen
Image1
HorizontalArrangen
HorizontalArrangen
Label1
TextBox1
HorizontalArrangen
Label2
TextBox2
HorizontalArrangen
Label3
TextBox3
HorizontalArrangen
HorizontalArrangen
Button1
Rename
Delete
Media
Properties
Screen3
AboutScreen
AlignHorizontal
Center : 3
AlignVertical
Top : 1
BackgroundColor
Default
BackgroundImage
None...
BigDefaultText
CloseScreenAnimation
Default
HighContrast
OpenScreenAnimation
Default
ScreenOrientation
Unspecified
Scrollable
10:39 PM
11/7/2022


10:06
SMART FARM


10:07
HOME


SMART FARM


SMART FARM


USERNAME:
Smart_Farmer
PASSWORD:


SUBMIT


10:07


SMART FARM


Temperature :
Humidity :
Moisture :


MOTOR ON
MOTOR OFF

Testing - Compatibility Testing :

- Compatibility Testing is a type of Software testing to check whether your software is capable of running on different hardware, operating systems, applications, network environments or Mobile devices.
- Compatibility testing is a type of non-functional testing.
- Herewith, our test report is attached with log and output.

Test Report for Sprint 1 :

PROJECT NAME : SMART FARMER – IOT ENABLED SMART FARMING APPLICATION

TEAM ID	PNT2022TMID30597
MODULE DESCRIPTION	Here we tested the compatibility of our Registration, Login and Dashboard Module with high Authentication
TYPE	Testing/verification

Development of Sprint-2 (Python Code for Publish & Subscribe)

Smart_Farming.py - C:\Users\A S ABISHEK\AppData\Local\Programs\Python\Python37\Smart_Farming.py (3.7.0)

File Edit Format Run Options Window Help

```
import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig={
    "identity":{
        "orgId":"uq23sr",
        "typeId":"Smart_Farming",
        "deviceId":"32826"
    },
    "auth": {
        "token":"3wNLT0O1g8VpEJEpsq"
    }
}
client=wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
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 switched ON")
    elif(m=="motoroff"):
        print("Motor is switched OFF")
    print(" ")
while True:
    soil=random.randint(0,100)
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    myData={'soil_moisture':soil,'temperature':temp,'humiduty':hum}
    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()
```

Ln: 1 Col: 0

1:12 AM 11/6/2022

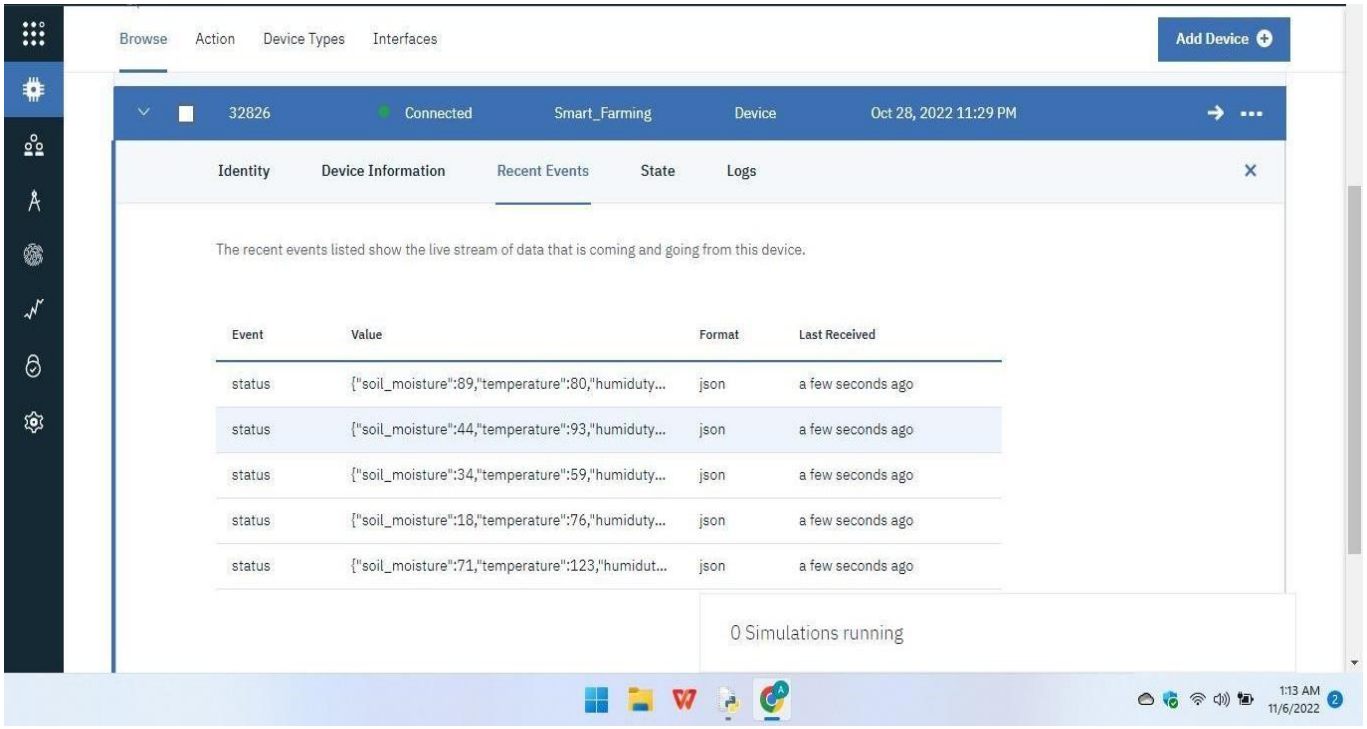
Python 3.7.0 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
 RESTART: C:\Users\A S ABISHEK\AppData\Local\Programs\Python\Python37\Smart_Farming.py
2022-11-06 01:11:23,500   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:uq23sr:Smart_Farming:32826Published
data successfully: %s
 {'soil_moisture': 83, 'temperature': 29, 'humiduty': 36}
Published data successfully: %s {'soil_moisture': 47, 'temperature': 96, 'humiduty': 25}
Published data successfully: %s {'soil_moisture': 45, 'temperature': 5, 'humiduty': 49}
Published data successfully: %s {'soil_moisture': 82, 'temperature': 29, 'humiduty': 80}
Published data successfully: %s {'soil_moisture': 92, 'temperature': 49, 'humiduty': 86}
Published data successfully: %s {'soil_moisture': 60, 'temperature': 11, 'humiduty': 98}
Published data successfully: %s {'soil_moisture': 40, 'temperature': -18, 'humiduty': 100}
Published data successfully: %s {'soil_moisture': 17, 'temperature': 80, 'humiduty': 33}
Published data successfully: %s {'soil_moisture': 60, 'temperature': 93, 'humiduty': 95}
Published data successfully: %s {'soil_moisture': 22, 'temperature': 15, 'humiduty': 70}
Published data successfully: %s {'soil_moisture': 78, 'temperature': 72, 'humiduty': 15}
Published data successfully: %s {'soil_moisture': 42, 'temperature': 41, 'humiduty': 63}
Published data successfully: %s {'soil_moisture': 21, 'temperature': 61, 'humiduty': 50}
Published data successfully: %s {'soil_moisture': 36, 'temperature': 95, 'humiduty': 56}
Published data successfully: %s {'soil_moisture': 14, 'temperature': 102, 'humiduty': 74}
Published data successfully: %s {'soil_moisture': 34, 'temperature': 31, 'humiduty': 44}
Published data successfully: %s {'soil_moisture': 95, 'temperature': 1, 'humiduty': 37}
Published data successfully: %s {'soil_moisture': 58, 'temperature': 61, 'humiduty': 97}
Published data successfully: %s {'soil_moisture': 94, 'temperature': 86, 'humiduty': 14}
Published data successfully: %s {'soil_moisture': 56, 'temperature': 89, 'humiduty': 82}
Published data successfully: %s {'soil_moisture': 100, 'temperature': -12, 'humiduty': 61}
Published data successfully: %s {'soil_moisture': 82, 'temperature': 63, 'humiduty': 41}
Published data successfully: %s {'soil_moisture': 30, 'temperature': 53, 'humiduty': 84}
Published data successfully: %s {'soil_moisture': 42, 'temperature': 83, 'humiduty': 71}
Published data successfully: %s {'soil_moisture': 80, 'temperature': 107, 'humiduty': 38}
Published data successfully: %s {'soil_moisture': 64, 'temperature': 14, 'humiduty': 73}
Published data successfully: %s {'soil_moisture': 95, 'temperature': 0, 'humiduty': 100}
Published data successfully: %s {'soil_moisture': 67, 'temperature': 124, 'humiduty': 21}
Published data successfully: %s {'soil_moisture': 51, 'temperature': 109, 'humiduty': 5}
Published data successfully: %s {'soil_moisture': 25, 'temperature': 35, 'humiduty': 58}
```

Ln: 57 Col: 4

1:13 AM 11/6/2022


Browse
Action
Device Types
Interfaces
Add Device
32826
Connected
Smart_Farming
Device
Oct 28, 2022 11:29 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
{"soil_moisture":89,"temperature":80,"humiduty...
json
a few seconds ago
status
{"soil_moisture":44,"temperature":93,"humiduty...
json
a few seconds ago
status
{"soil_moisture":34,"temperature":59,"humiduty...
json
a few seconds ago
status
{"soil_moisture":18,"temperature":76,"humiduty...
json
a few seconds ago
status
{"soil_moisture":71,"temperature":123,"humidut...
json
a few seconds ago
0 Simulations running
1:13 AM
11/6/2022

Development of Sprint-3 (Web Application Using Node-Red)

Smart_Farming.py - C:\Users\A S ABISHEK\AppData\Local\Programs\Python\Python37\Smart_Farming.py (3.7.0)

File Edit Format Run Options Window Help

```
import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig={
    "identity":{
        "orgId":"uq23sr",
        "typeId":"Smart_Farming",
        "deviceId":"32826"
    },
    "auth": {
        "token":"3wNLT0O1g8VpEJEpsq"
    }
}
client=wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
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 switched ON")
    elif(m=="motoroff"):
        print("Motor is switched OFF")
    print(" ")
while True:
    soil=random.randint(0,100)
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    myData={'soil_moisture':soil,'temperature':temp,'humidity':hum}
    client.publishEvent(eventId="status",msgFormat="json",data=myData,qos=0,onPublish=None)
    print("Published data successfully",myData)
    time.sleep(2)
    client.commandCallback=myCommandCallback
client.disconnect()
```

Ln: 32 Col: 38

4:04 PM 11/10/2022

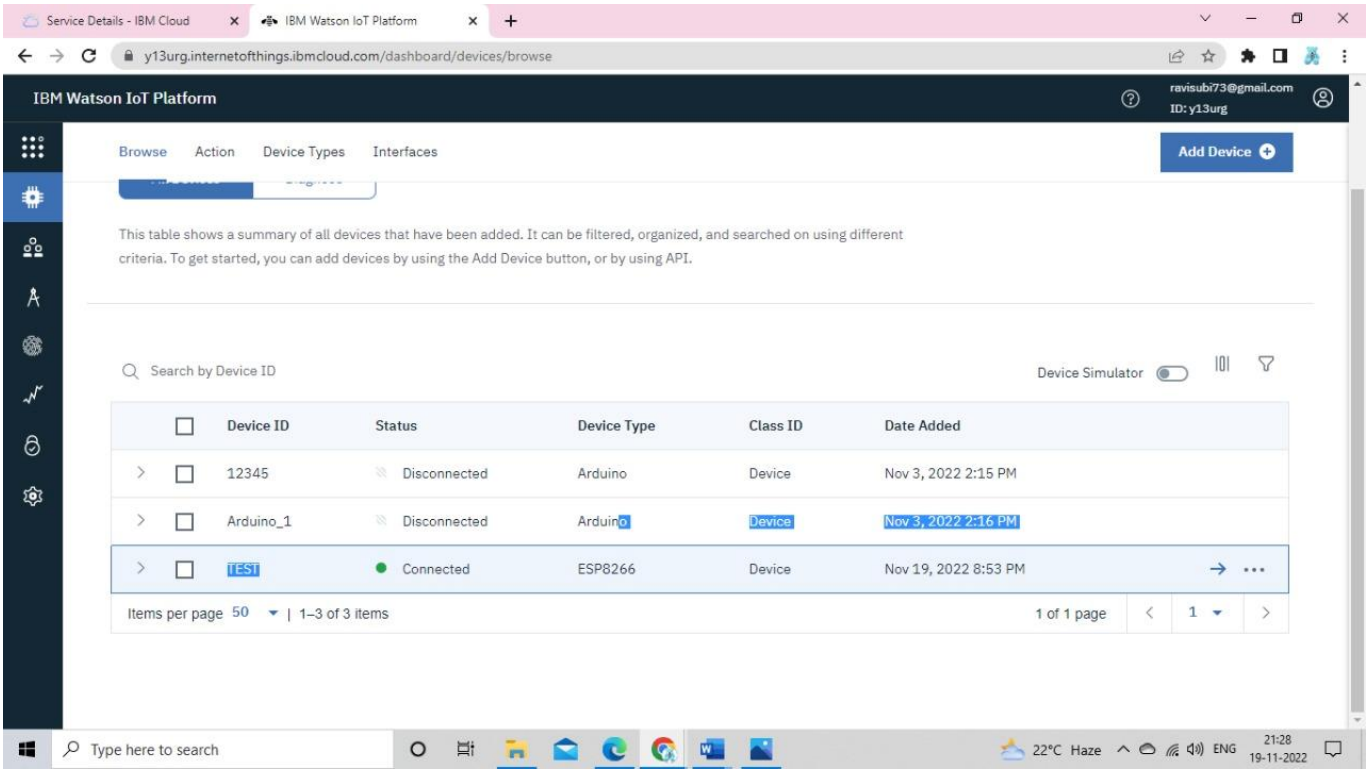
Python 3.7.0 Shell

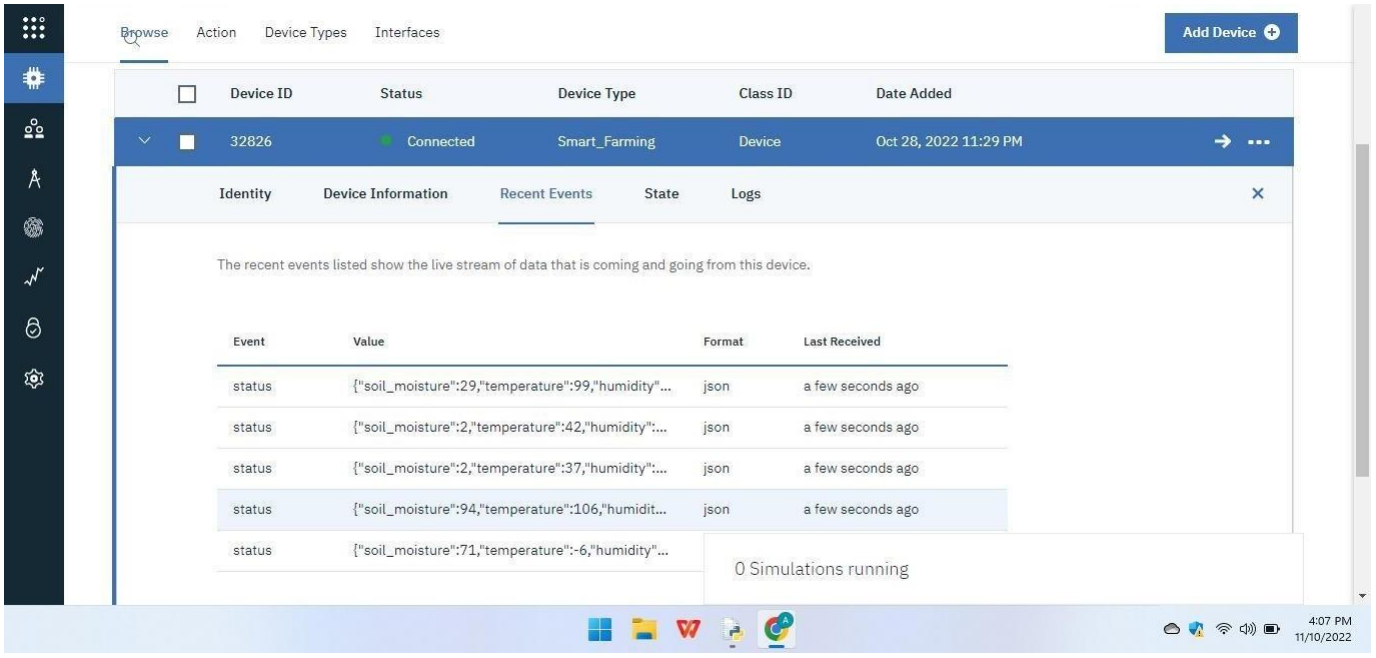
File Edit Shell Debug Options Window Help

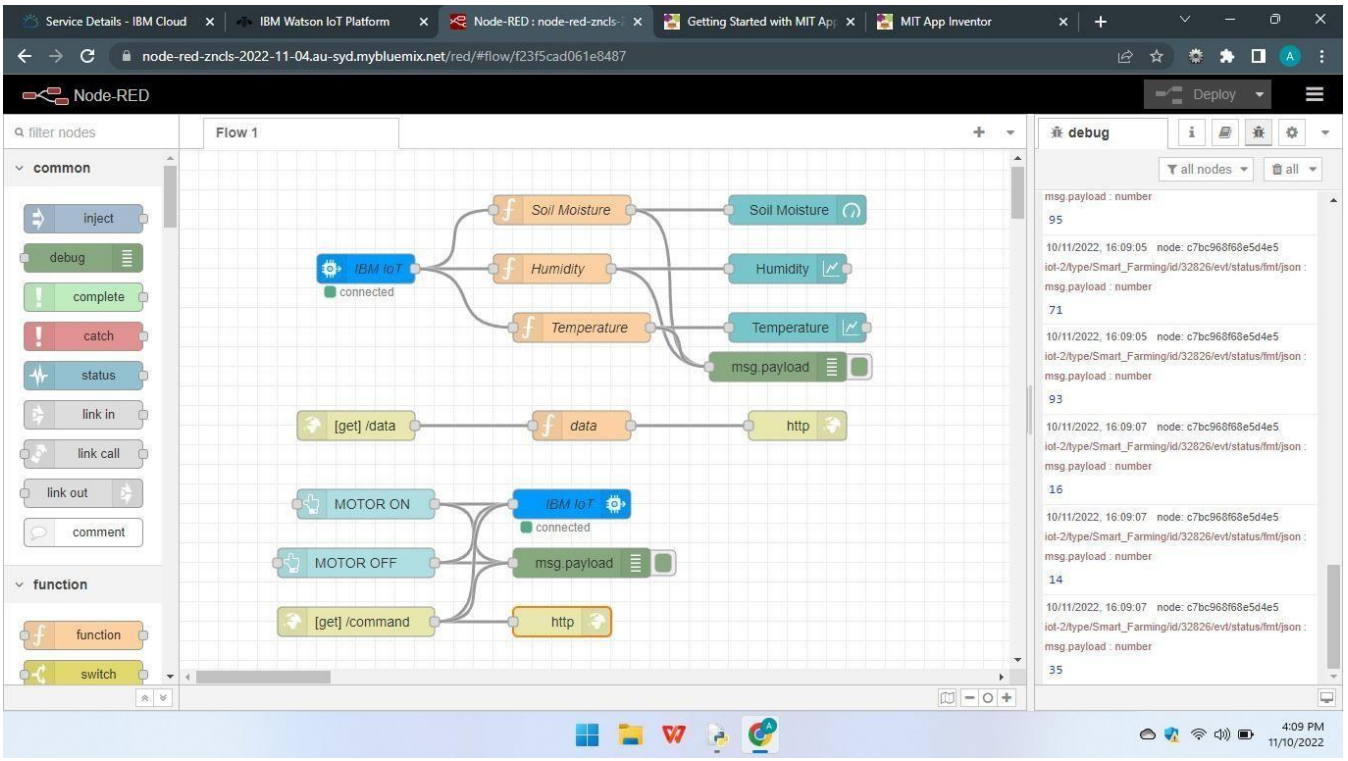
```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
 RESTART: C:\Users\A S ABISHEK\AppData\Local\Programs\Python\Python37\Smart_Farming.py
2022-11-10 16:04:42,463   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:uq23sr:Smart_Farming:32826Published
data successfully
 {'soil_moisture': 37, 'temperature': 1, 'humidity': 35}
Published data successfully {'soil_moisture': 89, 'temperature': 94, 'humidity': 24}
Published data successfully {'soil_moisture': 57, 'temperature': 28, 'humidity': 90}
Published data successfully {'soil_moisture': 65, 'temperature': -18, 'humidity': 4}
Published data successfully {'soil_moisture': 87, 'temperature': 81, 'humidity': 92}
Published data successfully {'soil_moisture': 62, 'temperature': -16, 'humidity': 33}
Published data successfully {'soil_moisture': 99, 'temperature': 105, 'humidity': 62}
Published data successfully {'soil_moisture': 41, 'temperature': 114, 'humidity': 78}
Published data successfully {'soil_moisture': 26, 'temperature': -15, 'humidity': 49}
Published data successfully {'soil_moisture': 55, 'temperature': 84, 'humidity': 87}
```

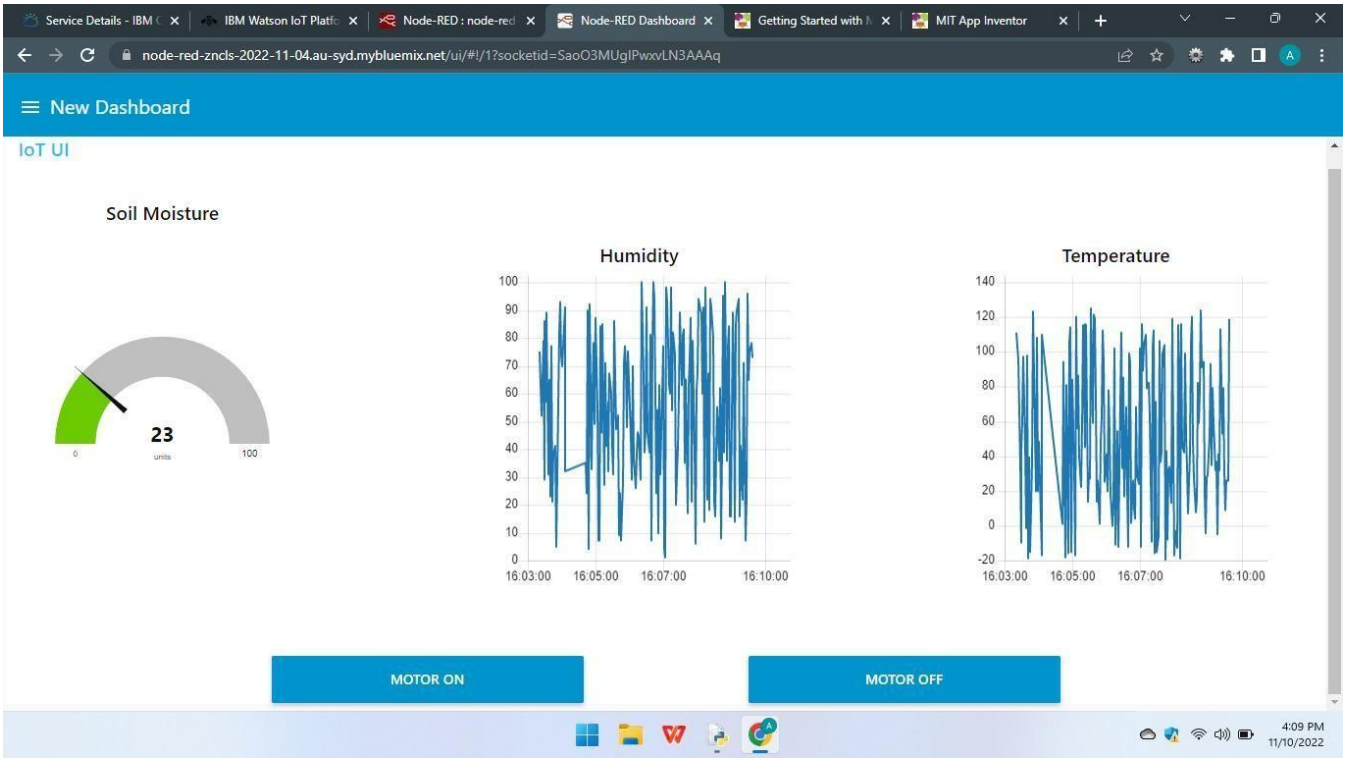
Ln: 5 Col: 0

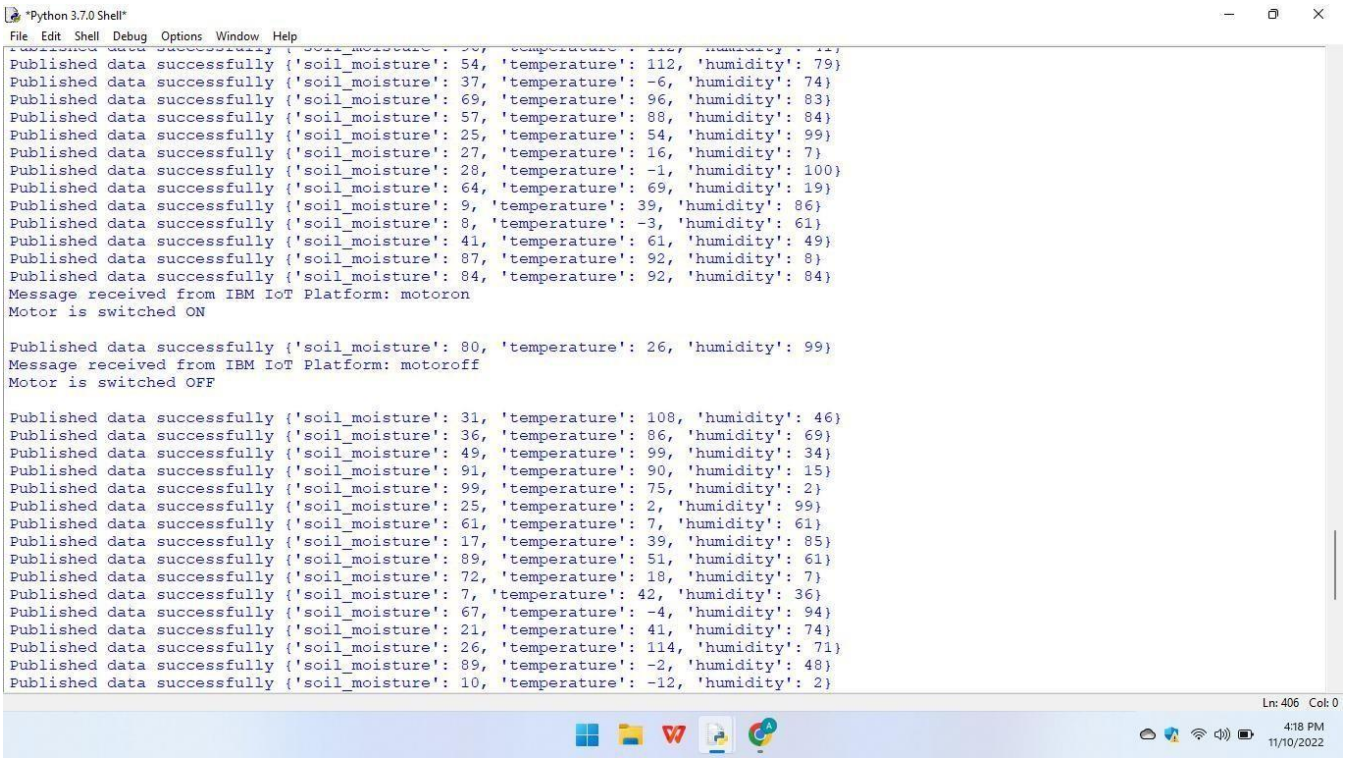
4:05 PM 11/10/2022

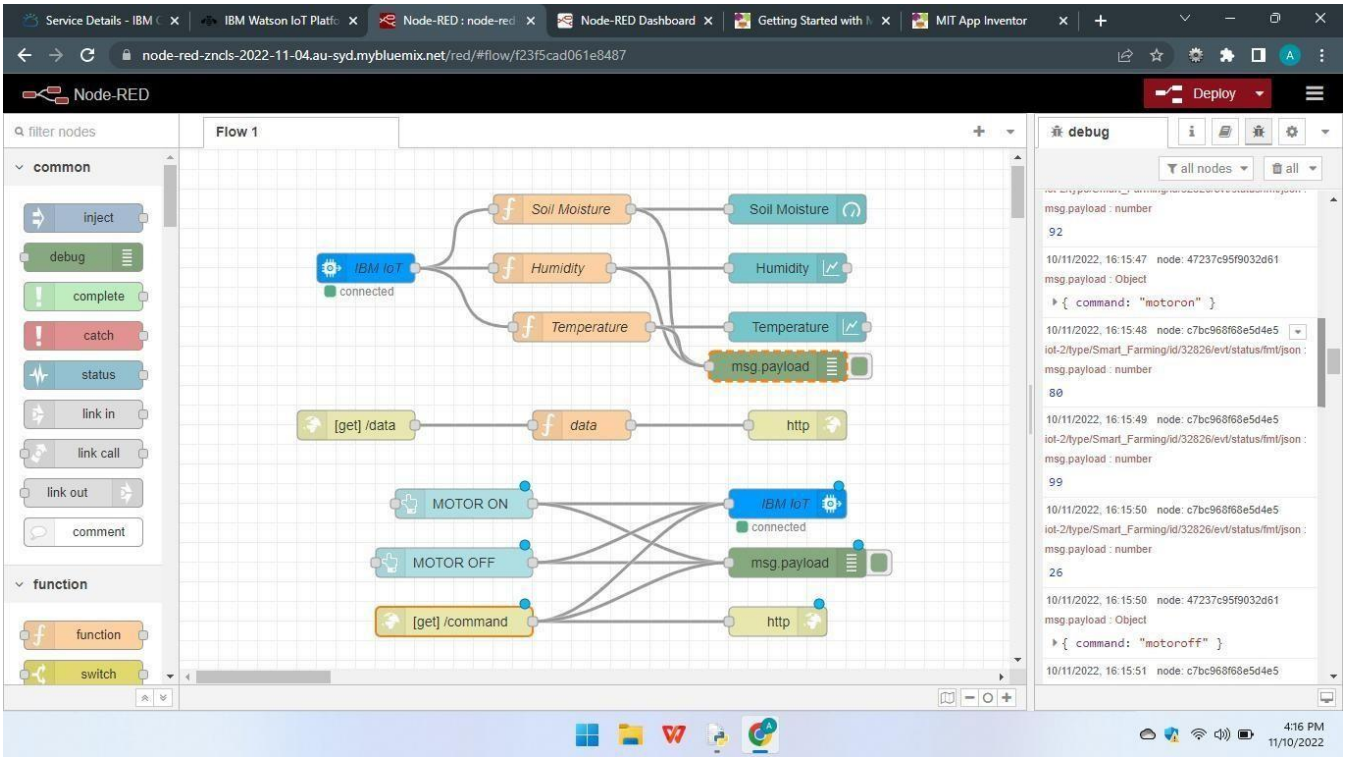

Service Details - IBM Cloud
IBM Watson IoT Platform
y13urg.internetofthings.ibmcloud.com/dashboard/devices/browse
IBM Watson IoT Platform
ravisubi73@gmail.com
ID: y13urg
Browse
Action
Device Types
Interfaces
Add Device
This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.
Search by Device ID
Device Simulator
Device ID
Status
Device Type
Class ID
Date Added
12345
Disconnected
Arduino
Device
Nov 3, 2022 2:15 PM
Arduino_1
Disconnected
Arduino
Device
Nov 3, 2022 2:16 PM
TEST
Connected
ESP8266
Device
Nov 19, 2022 8:53 PM
Items per page 50
1–3 of 3 items
1 of 1 page
1
Type here to search
22°C Haze
ENG
21:28
19-11-2022


Browse
Action
Device Types
Interfaces
Add Device
Device ID
Status
Device Type
Class ID
Date Added
32826
Connected
Smart_Farming
Device
Oct 28, 2022 11:29 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
{"soil_moisture":29,"temperature":99,"humidity"...
json
a few seconds ago
status
{"soil_moisture":2,"temperature":42,"humidity":...
json
a few seconds ago
status
{"soil_moisture":2,"temperature":37,"humidity":...
json
a few seconds ago
status
{"soil_moisture":94,"temperature":106,"humidit...
json
a few seconds ago
status
{"soil_moisture":71,"temperature":-6,"humidity"...
0 Simulations running
4:07 PM
11/10/2022


Service Details - IBM Cloud
IBM Watson IoT Platform
Node-RED : node-red-zncls-
Getting Started with MIT Ap
MIT App Inventor
node-red-zncls-2022-11-04.au-syd.mybluemix.net/red/#flow/f23f5cad061e8487
Node-RED
Deploy
filter nodes
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
Soil Moisture
Humidity
Temperature
IBM IoT
connected
Soil Moisture
Humidity
Temperature
msg.payload
[get] /data
data
http
MOTOR ON
MOTOR OFF
[get] /command
IBM IoT
connected
msg.payload
http
debug
all nodes
all
msg.payload : number
95
10/11/2022, 16:09:05 node: c7bc968f68e5d4e5
iot-2/type/Smart_Farming/id/32826/evt/status/fmt/json : msg.payload : number
71
10/11/2022, 16:09:05 node: c7bc968f68e5d4e5
iot-2/type/Smart_Farming/id/32826/evt/status/fmt/json : msg.payload : number
93
10/11/2022, 16:09:07 node: c7bc968f68e5d4e5
iot-2/type/Smart_Farming/id/32826/evt/status/fmt/json : msg.payload : number
16
10/11/2022, 16:09:07 node: c7bc968f68e5d4e5
iot-2/type/Smart_Farming/id/32826/evt/status/fmt/json : msg.payload : number
14
10/11/2022, 16:09:07 node: c7bc968f68e5d4e5
iot-2/type/Smart_Farming/id/32826/evt/status/fmt/json : msg.payload : number
35
4:09 PM
11/10/2022


Service Details - IBM
IBM Watson IoT Platfc
Node-RED : node-red
Node-RED Dashboard
Getting Started with
MIT App Inventor
node-red-zncls-2022-11-04.au-syd.mybluemix.net/ui/#!/1?socketid=SaoO3MUglPwxvLN3AAAq
New Dashboard
IoT UI
Soil Moisture
23
0
units
100
Humidity
100
90
80
70
60
50
40
30
20
10
0
16:03:00
16:05:00
16:07:00
16:10:00
Temperature
140
120
100
80
60
40
20
0
-20
16:03:00
16:05:00
16:07:00
16:10:00
MOTOR ON
MOTOR OFF
4:09 PM
11/10/2022


Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Published data successfully {'soil_moisture': 54, 'temperature': 112, 'humidity': 79}
Published data successfully {'soil_moisture': 37, 'temperature': -6, 'humidity': 74}
Published data successfully {'soil_moisture': 69, 'temperature': 96, 'humidity': 83}
Published data successfully {'soil_moisture': 57, 'temperature': 88, 'humidity': 84}
Published data successfully {'soil_moisture': 25, 'temperature': 54, 'humidity': 99}
Published data successfully {'soil_moisture': 27, 'temperature': 16, 'humidity': 7}
Published data successfully {'soil_moisture': 28, 'temperature': -1, 'humidity': 100}
Published data successfully {'soil_moisture': 64, 'temperature': 69, 'humidity': 19}
Published data successfully {'soil_moisture': 9, 'temperature': 39, 'humidity': 86}
Published data successfully {'soil_moisture': 8, 'temperature': -3, 'humidity': 61}
Published data successfully {'soil_moisture': 41, 'temperature': 61, 'humidity': 49}
Published data successfully {'soil_moisture': 87, 'temperature': 92, 'humidity': 8}
Published data successfully {'soil_moisture': 84, 'temperature': 92, 'humidity': 84}
Message received from IBM IoT Platform: motoron
Motor is switched ON
Published data successfully {'soil_moisture': 80, 'temperature': 26, 'humidity': 99}
Message received from IBM IoT Platform: motoroff
Motor is switched OFF
Published data successfully {'soil_moisture': 31, 'temperature': 108, 'humidity': 46}
Published data successfully {'soil_moisture': 36, 'temperature': 86, 'humidity': 69}
Published data successfully {'soil_moisture': 49, 'temperature': 99, 'humidity': 34}
Published data successfully {'soil_moisture': 91, 'temperature': 90, 'humidity': 15}
Published data successfully {'soil_moisture': 99, 'temperature': 75, 'humidity': 2}
Published data successfully {'soil_moisture': 25, 'temperature': 2, 'humidity': 99}
Published data successfully {'soil_moisture': 61, 'temperature': 7, 'humidity': 61}
Published data successfully {'soil_moisture': 17, 'temperature': 39, 'humidity': 85}
Published data successfully {'soil_moisture': 89, 'temperature': 51, 'humidity': 61}
Published data successfully {'soil_moisture': 72, 'temperature': 18, 'humidity': 7}
Published data successfully {'soil_moisture': 7, 'temperature': 42, 'humidity': 36}
Published data successfully {'soil_moisture': 67, 'temperature': -4, 'humidity': 94}
Published data successfully {'soil_moisture': 21, 'temperature': 41, 'humidity': 74}
Published data successfully {'soil_moisture': 26, 'temperature': 114, 'humidity': 71}
Published data successfully {'soil_moisture': 89, 'temperature': -2, 'humidity': 48}
Published data successfully {'soil_moisture': 10, 'temperature': -12, 'humidity': 2}
Ln: 406 Col: 0
4:18 PM
11/10/2022


Service Details - IBM
IBM Watson IoT Platfo
Node-RED : node-red
Node-RED Dashboard
Getting Started with
MIT App Inventor
node-red-zncls-2022-11-04.au-syd.mybluemix.net/red/#flow/f23f5cad061e8487
Node-RED
Deploy
filter nodes
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
Soil Moisture
Soil Moisture
IBM IoT
connected
Humidity
Humidity
Temperature
Temperature
msg.payload
[get] /data
data
http
MOTOR ON
IBM IoT
connected
MOTOR OFF
msg.payload
[get] /command
http
debug
all nodes
all
msg.payload : number
92
10/11/2022, 16:15:47 node: 47237c95f9032d61
msg.payload : Object
{ command: "motoron" }
10/11/2022, 16:15:48 node: c7bc968f68e5d4e5
iot-2/type/Smart_Farming/id/32826/evt/status/fmt/json :
msg.payload : number
80
10/11/2022, 16:15:49 node: c7bc968f68e5d4e5
iot-2/type/Smart_Farming/id/32826/evt/status/fmt/json :
msg.payload : number
99
10/11/2022, 16:15:50 node: c7bc968f68e5d4e5
iot-2/type/Smart_Farming/id/32826/evt/status/fmt/json :
msg.payload : number
26
10/11/2022, 16:15:50 node: 47237c95f9032d61
msg.payload : Object
{ command: "motoroff" }
10/11/2022, 16:15:51 node: c7bc968f68e5d4e5
4:16 PM
11/10/2022

Development of Sprint-4 (MIT Application and its Execution)

Smart_Farming.py - C:\Users\A S ABISHEK\AppData\Local\Programs\Python\Python37\Smart_Farming.py (3.7.0)

File Edit Format Run Options Window Help

```
import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig={
    "identity":{
        "orgId":"uq23sr",
        "typeId":"Smart_Farming",
        "deviceId":"32826"
    },
    "auth": {
        "token":"3wNLT0O1g8VpEJEpsq"
    }
}
client=wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
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 switched ON")
    elif(m=="motoroff"):
        print("Motor is switched OFF")
    print(" ")
while True:
    soil=random.randint(0,100)
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    myData={'soil_moisture':soil,'temperature':temp,'humidity':hum}
    client.publishEvent(eventId="status",msgFormat="json",data=myData,qos=0,onPublish=None)
    print("Published data successfully",myData)
    time.sleep(2)
    client.commandCallback=myCommandCallback
client.disconnect()
```

Ln: 32 Col: 38

4:04 PM 11/10/2022

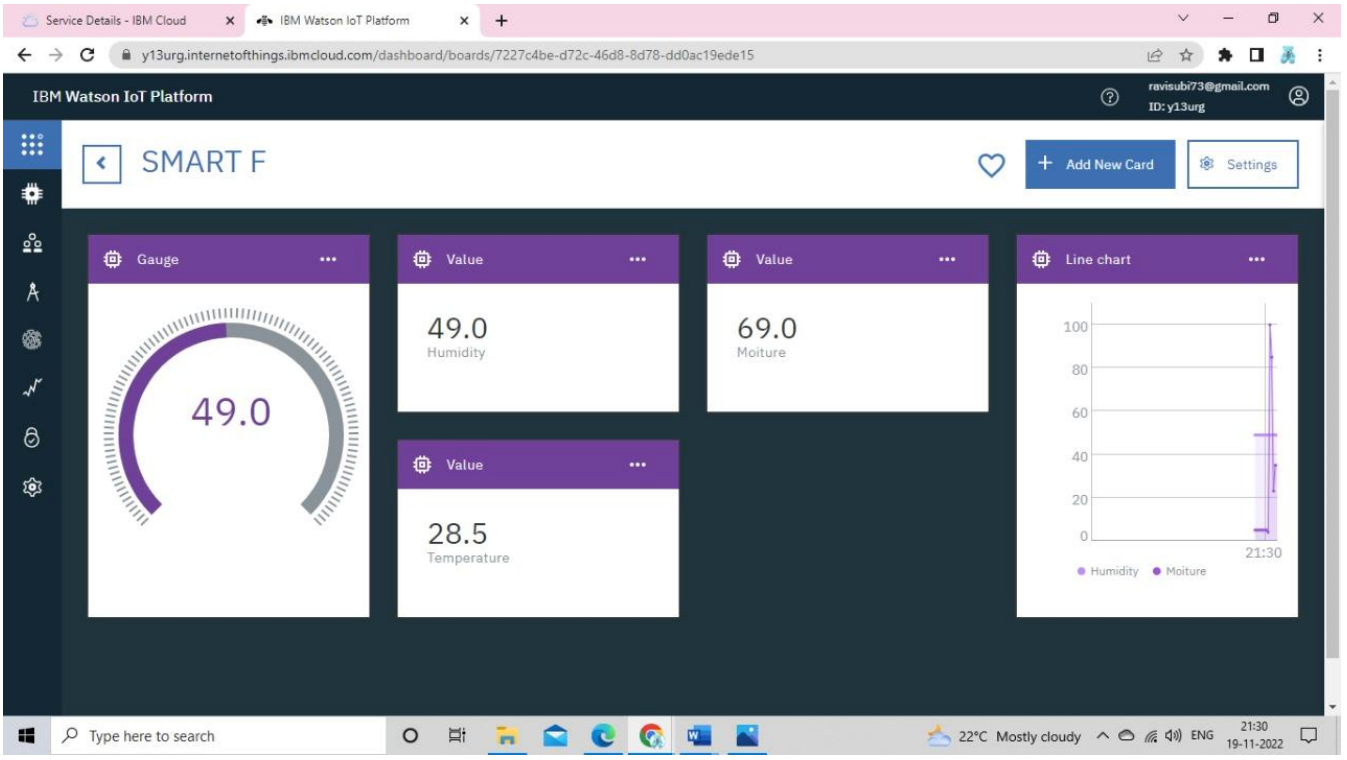
Python 3.7.0 Shell

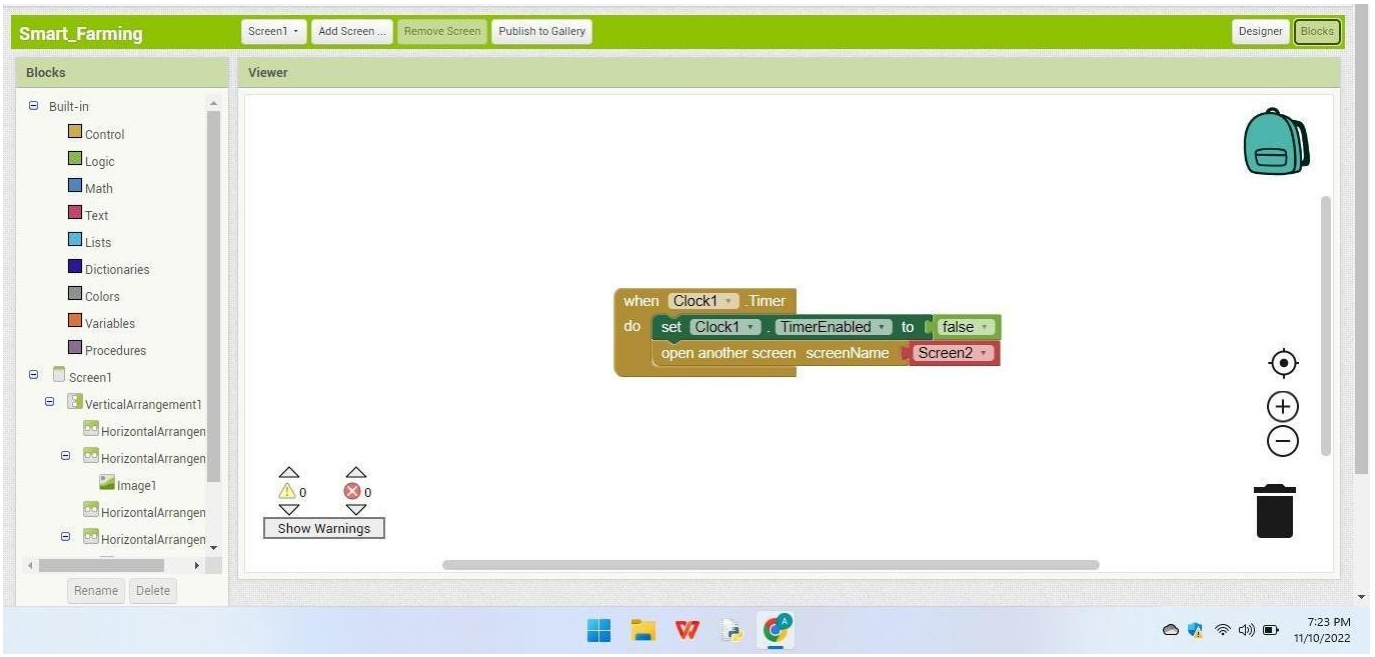
File Edit Shell Debug Options Window Help

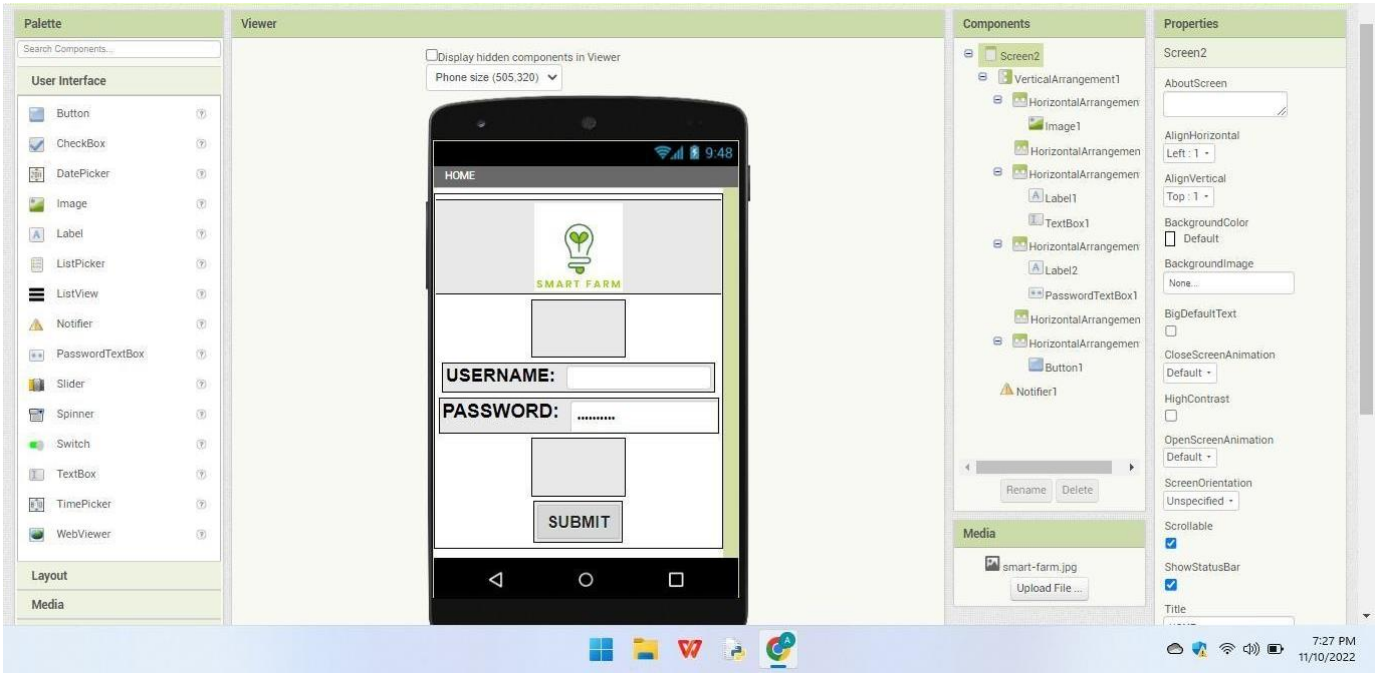
```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
 RESTART: C:\Users\A S ABISHEK\AppData\Local\Programs\Python\Python37\Smart_Farming.py
2022-11-10 16:04:42,463   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:uq23sr:Smart_Farming:32826Published
data successfully
 {'soil_moisture': 37, 'temperature': 1, 'humidity': 35}
Published data successfully {'soil_moisture': 89, 'temperature': 94, 'humidity': 24}
Published data successfully {'soil_moisture': 57, 'temperature': 28, 'humidity': 90}
Published data successfully {'soil_moisture': 65, 'temperature': -18, 'humidity': 4}
Published data successfully {'soil_moisture': 87, 'temperature': 81, 'humidity': 92}
Published data successfully {'soil_moisture': 62, 'temperature': -16, 'humidity': 33}
Published data successfully {'soil_moisture': 99, 'temperature': 105, 'humidity': 62}
Published data successfully {'soil_moisture': 41, 'temperature': 114, 'humidity': 78}
Published data successfully {'soil_moisture': 26, 'temperature': -15, 'humidity': 49}
Published data successfully {'soil_moisture': 55, 'temperature': 84, 'humidity': 87}
```

Ln: 5 Col: 0

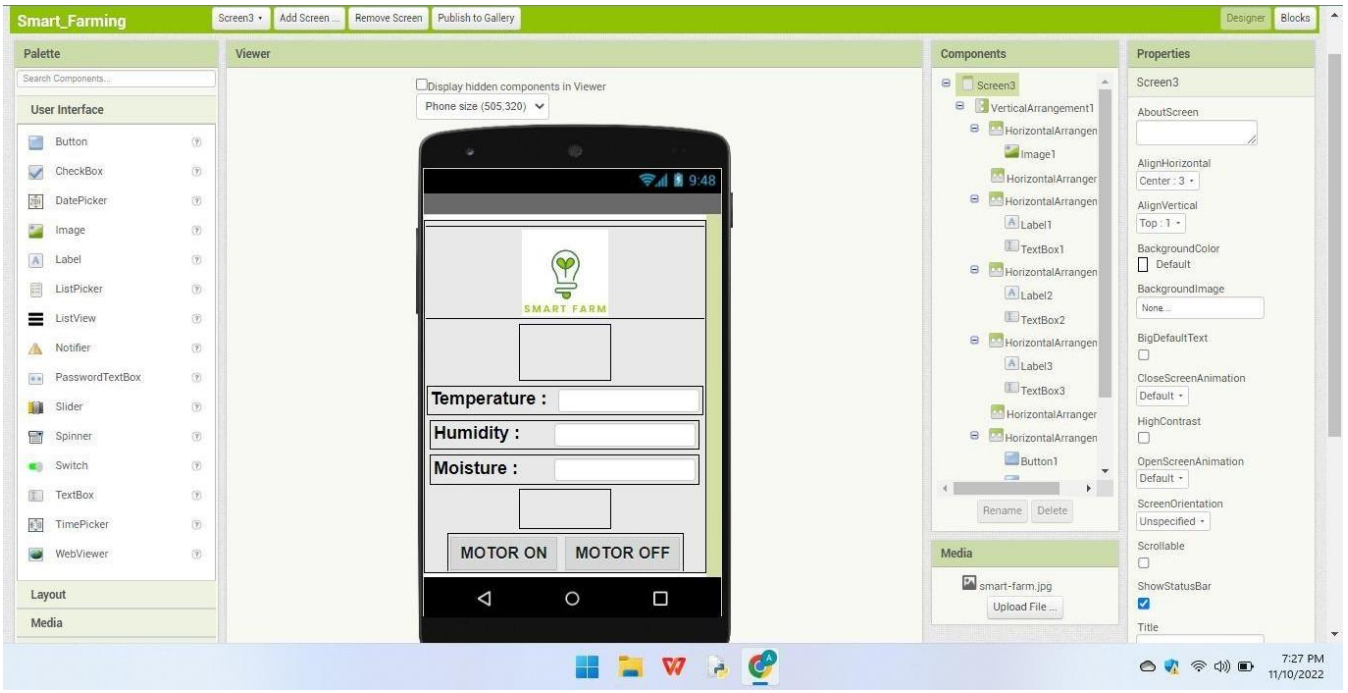
4:05 PM 11/10/2022

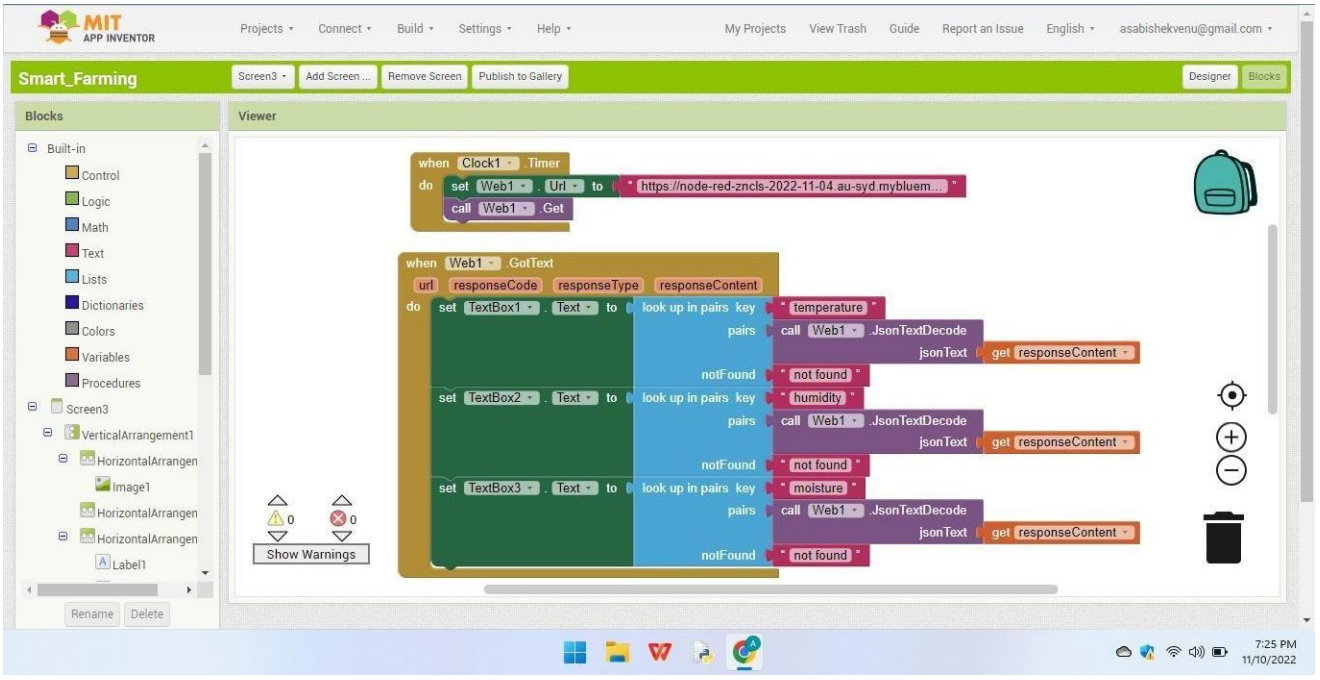

Service Details - IBM Cloud
IBM Watson IoT Platform
y13urg.internetofthings.ibmcloud.com/dashboard/boards/7227c4be-d72c-46d8-8d78-dd0ac19ede15
IBM Watson IoT Platform
ravisubi73@gmail.com
ID: y13urg
SMART F
Add New Card
Settings
Gauge
49.0
Value
49.0
Humidity
Value
69.0
Moiture
Value
28.5
Temperature
Line chart
100
80
60
40
20
0
21:30
Humidity
Moiture
Type here to search
22°C Mostly cloudy
ENG
21:30
19-11-2022

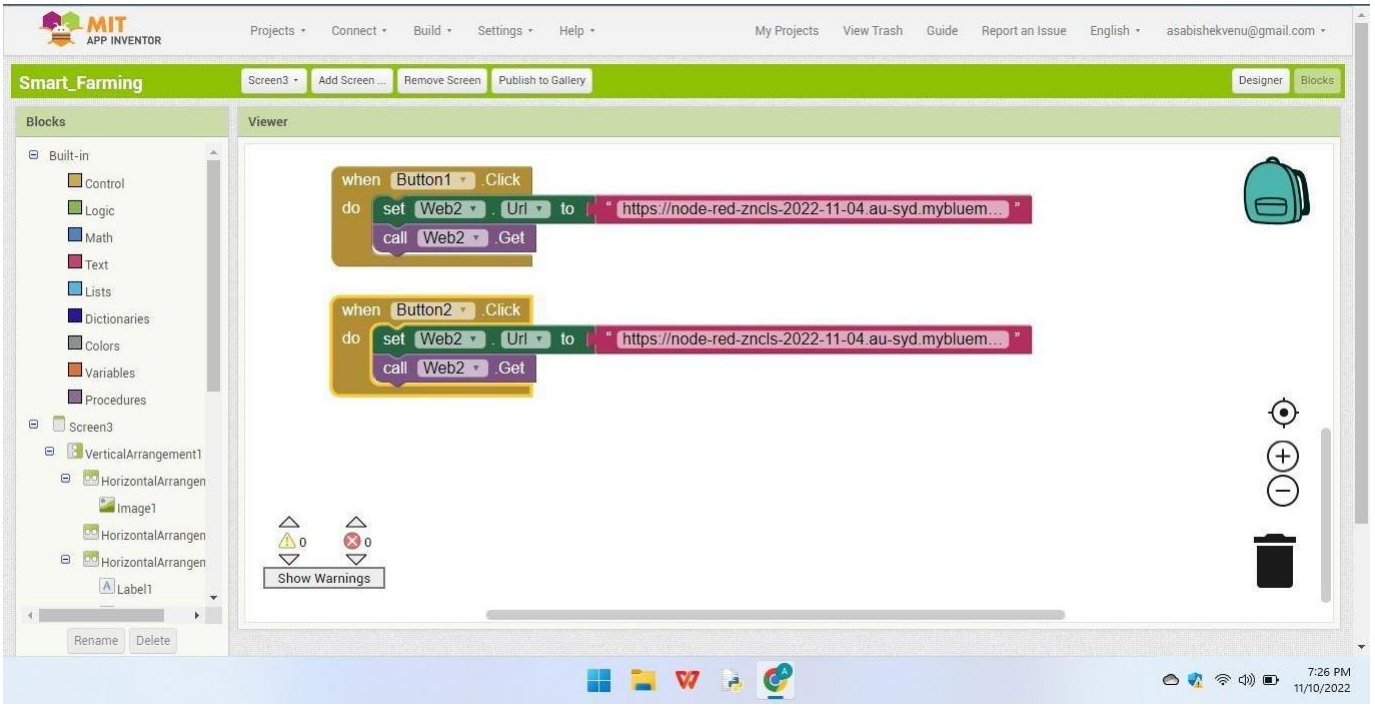

Smart_Farming
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen1
VerticalArrangement1
HorizontalArrangen
HorizontalArrangen
Image1
HorizontalArrangen
HorizontalArrangen
Rename
Delete
when Clock1 .Timer
do set Clock1 . TimerEnabled to false
open another screen screenName Screen2
0
0
Show Warnings
7:23 PM
11/10/2022


Palette
Search Components...
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Media
Viewer
Display hidden components in Viewer
Phone size (505,320)
9:48
HOME
SMART FARM
USERNAME:
PASSWORD:
SUBMIT
Components
Screen2
VerticalArrangement1
HorizontalArrangemen
Image1
Label1
TextBox1
Label2
PasswordTextBox1
Button1
Notifier1
Rename
Delete
Media
smart-farm.jpg
Upload File ...
Properties
Screen2
AboutScreen
AlignHorizontal
Left : 1
AlignVertical
Top : 1
BackgroundColor
Default
BackgroundImage
None...
BigDefaultText
CloseScreenAnimation
Default
HighContrast
OpenScreenAnimation
Default
ScreenOrientation
Unspecified
Scrollable
ShowStatusBar
Title
7:27 PM
11/10/2022


Blocks
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen2
VerticalArrangement1
HorizontalArrangen
Image1
HorizontalArrangen
HorizontalArrangen
Label1
Rename
Delete
Viewer
when Button1 .Click
do if TextBox1 . Text = " Smart_Farmer " and PasswordTextBox1 . Text = " Smart_Farmer "
then open another screen screenName Screen3
else call Notifier1 .ShowAlert
notice " Check Your Credentials!! "
0
0
Show Warnings
7:23 PM
11/10/2022


Smart_Farming
Screen3
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Palette
Search Components...
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Media
Viewer
Display hidden components in Viewer
Phone size (505,320)
9:48
SMART FARM
Temperature :
Humidity :
Moisture :
MOTOR ON
MOTOR OFF
Components
Screen3
VerticalArrangement1
HorizontalArrangen
Image1
Label1
TextBox1
Label2
TextBox2
Label3
TextBox3
Button1
Rename
Delete
Media
smart-farm.jpg
Upload File ...
Properties
Screen3
AboutScreen
AlignHorizontal
Center : 3
AlignVertical
Top : 1
BackgroundColor
Default
BackgroundImage
None...
BigDefaultText
CloseScreenAnimation
Default
HighContrast
OpenScreenAnimation
Default
ScreenOrientation
Unspecified
Scrollable
ShowStatusBar
Title
7:27 PM
11/10/2022


MIT
APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
asabishekvenu@gmail.com
Smart_Farming
Screen3
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Blocks
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen3
VerticalArrangement1
HorizontalArrangen
Image1
HorizontalArrangen
HorizontalArrangen
Label1
Rename
Delete
Viewer
when Clock1 .Timer
do set Web1 . Url to https://node-red-zncls-2022-11-04.au-syd.mybluem...
call Web1 .Get
when Web1 .GotText
url responseCode responseType responseContent
do set TextBox1 . Text to look up in pairs key temperature
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound not found
set TextBox2 . Text to look up in pairs key humidity
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound not found
set TextBox3 . Text to look up in pairs key moisture
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound not found
0
0
Show Warnings
7:25 PM
11/10/2022


MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
asabishekvenu@gmail.com
Smart_Farming
Screen3
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen3
VerticalArrangement1
HorizontalArrangen
Image1
HorizontalArrangen
HorizontalArrangen
Label1
Rename
Delete
when Button1 .Click
do set Web2 . Url to " https://node-red-zncls-2022-11-04.au-syd.mybluem... "
call Web2 .Get
when Button2 .Click
do set Web2 . Url to " https://node-red-zncls-2022-11-04.au-syd.mybluem... "
call Web2 .Get
0
0
Show Warnings
7:26 PM
11/10/2022


4:29
HOME
SMART FARM
USERNAME:
Smart_Farmer
PASSWORD:
SUBMIT

SMART FARM

Temperature :	60
Humidity :	48
Moisture :	17

MOTOR ON MOTOR OFF

CHAPTER 8

TESTING

8.1 Unit Testing

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. This testing methodology is done during the development process by the software developers and sometimes QA staff.

Unit testing is a type of testing in which individual units or functions of software testing. Its primary purpose is to test each unit or function. A unit is the smallest testable part of an application. It mainly has one or a few inputs and produces a single output.

8.2 Integration Testing

Integration testing is also known as integration and testing (I&T) , is a type of software testing in which the different units, modules or components of a software application are tested as a combined entity. However, these modules may be coded by different programmers.

Integration Testing is a type of software testing, which is performed on software to determine the flow between two or more modules by combining them. Integration testing makes sure that the interactions between different components of the software is completed smoothly without any complication.

The purpose of the integration testing is to expose faults in the interaction between integrated units. Once all the modules have been unit tested, integration testing is performed.

8.3 Test Cases

Table 8.1

S.NO	TEST CASE	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Temperature Detection	Username and Password	60	60	PASS

Table 8.2

S.NO	TEST CASE	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Humidity Detection	Username and Password	48	48	PASS

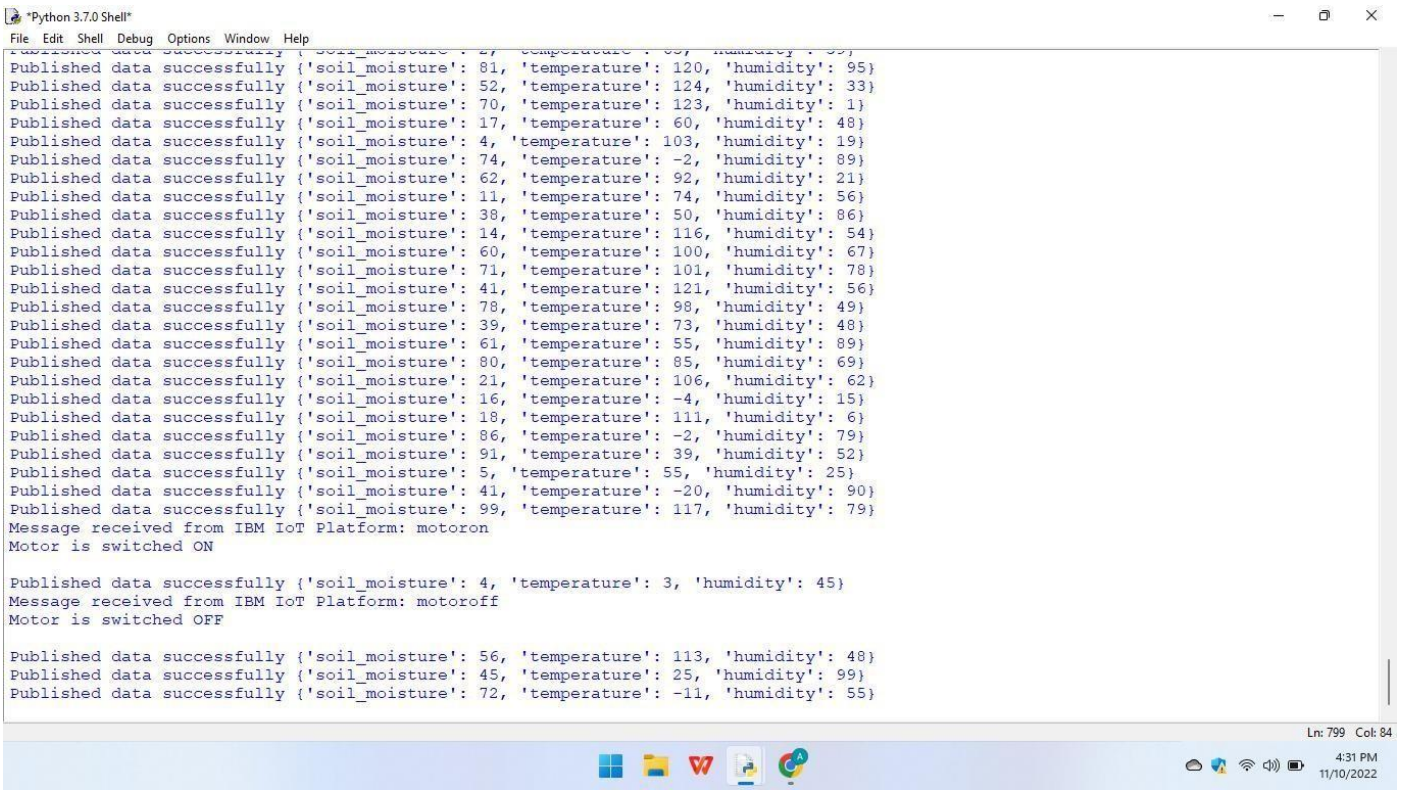
Table 8.3

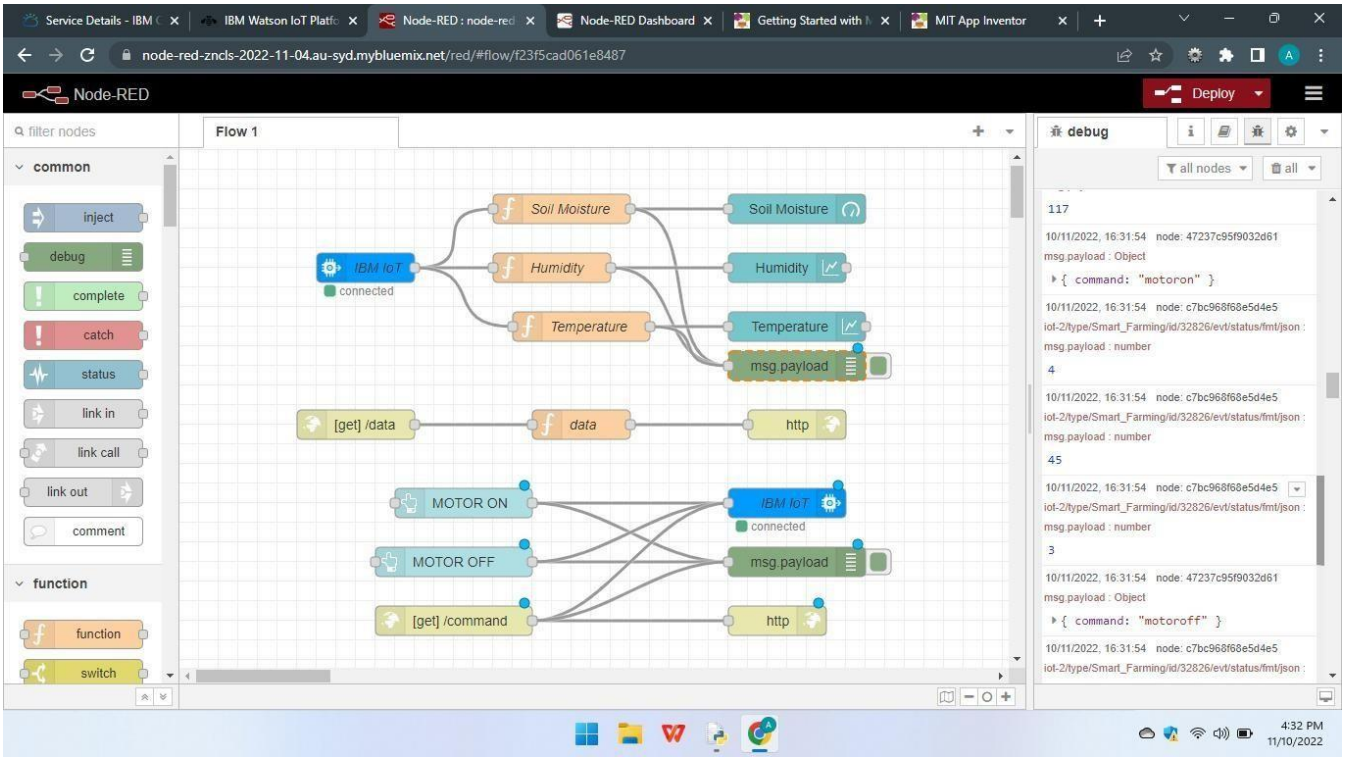
S.NO	TEST CASE	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Moisture Detection	Username and Password	17	17	PASS

*Note : The Output Values may vary accordingly.

CHAPTER 9

RESULTS





SMART FARM

Temperature :	60
Humidity :	48
Moisture :	17

MOTOR ON MOTOR OFF

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

Advantages:

- As it is a mobile friendly application one can access all the metrics in one touch.
- It has clean User interface so that user have smooth control over the application.
- The consumption of electric power is less as compared to other application.
- The moisture level and the temperature levels are monitored at regular intervals.
- It can run on all android versions.
- The application requires less memory and storage space.

Disadvantages:

- When the network connectivity is poor the performance of the application will be affected
- As it is platform dependent it cannot run on all devices.
- The application will produce inaccurate values when there is a fault or any change in API.
- The user should be more aware on the results produced.

CHAPTER 11

CONCLUSION

In this work, we successfully develop a system that can help in an automated irrigation system by analyzing the moisture level of the ground.

The smart irrigation system proves to be auseful system as it automates and regulates the watering without any manual intervention. The primary applications for this project are for farmers and gardeners who do not have enough timeto water crops/plants.

The farmers are facing major problems in watering their agriculture fields.So that the Farmers can Watering their plant Smart.

CHAPTER 12

FUTURE SCOPE

- It helps in automatic irrigation for crops and also helps to maintain the water level in field.
- The system will notify on the critical conditions.
- As this is an automated device it can works even the absence of farmer.

CHAPTER 13

APPENDIX

13.1 GitHub Link for Source Code :

https://github.com/IBM-EPBL/IBM-Project-15427-1659598581.git