SmartFarmer - IoT Enabled Smart Farming Application

PROJECT REPORT

Submitted by

JEEVA ANAND A (717819F218)

JEEVANANTHAM S (717819F219)

MADHANGOKUL R (717819F225)

SATHISH S (717819F245)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

ANNA UNIVERSITY : CHENNAI 600 025

NOVEMBER 2022

Table Of Contents

1. INTRODUCTION

1.1 Project Overview

Project Name:	**Project Mentor:**	**Project Lead:**	**Project Members:**
Smart Farmer – IoT Enabled Smart Farming Application	**Nagaraj S**	**Jeevanantham S**	**Jeeva anand A** **Madhangokul R** **Sathish S**

Problem/Opportunity:

Farmers are to be present at farm for its maintenance irrespective of the weather conditions. They have to ensure that the crops are well watered and the farm status is monitored by them physically. Farmer have to 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 have to work hard in their fields risking their lives to provide food for the country.

Goal:

Sustainably increasing agricultural productivity and incomes. Adapting and building resilience to climate change and saving energy resources where possible.

Objectives:

- Enables farmers to monitor the live data from sensors
- Low cost setup
- Control the devices/motors via application
- Create an application for interaction and viewing the live data
- Create web-UI to access the data across the devices.
- Integrate sensors to cloud

Proposed Budget and Costs :

1500 – 2000

Assumptions, Risks, Obstacles:

- Need proper internet connection
- Advanced Farming is the lack of awareness among consumers.
- Due to various service providers, it becomes really difficult to maintain interoperability between different IoT systems.
- A scalable solution that can be integrated with thousands of IoT devices for large farms.

2. LITERATURE SURVEY

Literature Survey on "Smart Farmer – IOT Enabled Smart Farming Application"

Reference	Technologies used	Advantages	Disadvantages
[1]	Microcontroller: CC3200 Chip, MCU Communication Technologies: MMS, Wi-Fi Module Sensors: Camera, Temperature Sensor, Humidity Sensor	• Sends the information about humidity and temperature in air of field to farmer. • Uses MMS • Technology to send captured images.	• MMS added extra cost • No automatic support system
[2]	Microcontroller: ATMEGA328P Cloud server: Adafruit Server Communication Technologies: Wi-Fi Sensors: Soil Moisture Sensor	Controlling the actions of motor Pump (ON/OFF) based on the threshold value.	• No sprinkles • No smart drains • No automatic support system
[3]	Microcontroller: Arduino Cloud server: ThingSpeak Sensors: Light Intensity, pH, Electrical Conductivity, Water Temperature, Relative Humidity	• Hydroponic System • Bayesian Network • Model • System has manual and automatic mode	• Extremely computationally expensive model
[4]	Microcontroller: Arduino UNO Cloud server: ThingSpeak Communication Technologies: Wi-Fi Sensors: Water Level Sensor, Moisture Sensor	Farmers can monitor their fields remotely Irrigation control system	• Lack of automated decision support system
[5]	Microcontroller: Arduino Sensors: Temperature Sensor, Humidity Sensor, Soil Moisture Sensor	Data regarding sensors stored on server and user can view via GUI application.	• Decision making is rely on user or farmer • No automatic support

2.1 Existing problem

In today's world Climate have been changed Because of the global warming these are mainly affecting farmers and agricultural lands .Some of the problems facing by the farmers are Cannot monitoring the weather situation near his or her land ,soil moisture, humidity and motor on off for 24/7.

2.2 References.

[1] Prathibha S., Hongal A., and Jyothi M. (2017). IOT Based Monitoring System in Smart Agriculture. 2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT). doi: 10.1109/icraect.2017.52.

[2] Lahande P., and Mathpathi D. (2018). IOT Based Smart Irrigation System. International Journal of Trend in Scientific Research and Development Volume-2(Issue-5), pp. 359-362. doi: 10.31142/ijtsrd15827.

[3] Alipio M., Dela Cruz A., Doria J., and Fruto R. (2019). On the design of Nutrient Film Technique hydroponics farm for smart agriculture. Engineering in Agriculture, Environment and Food, 12(3), pp.315- 324.
doi: 10.1016/j.eaef.2019.02.008.

[4] Benyezza H., Bouhedda M., Djellout K., and Saidi A. (2018). Smart Irrigation System Based Thingspeak and Arduino. International Conference on Applied Smart Systems (ICASS).doi: 10.1109/icass.2018.8651993.

[5] Kiani F., and Seyyedabbasi A. (2018). Wireless Sensor Network and Internet of Things in Precision Agriculture. International Journal of Advanced Computer Science and Applications, 9(6). doi:
10.14569/ijacsa.2018.090614

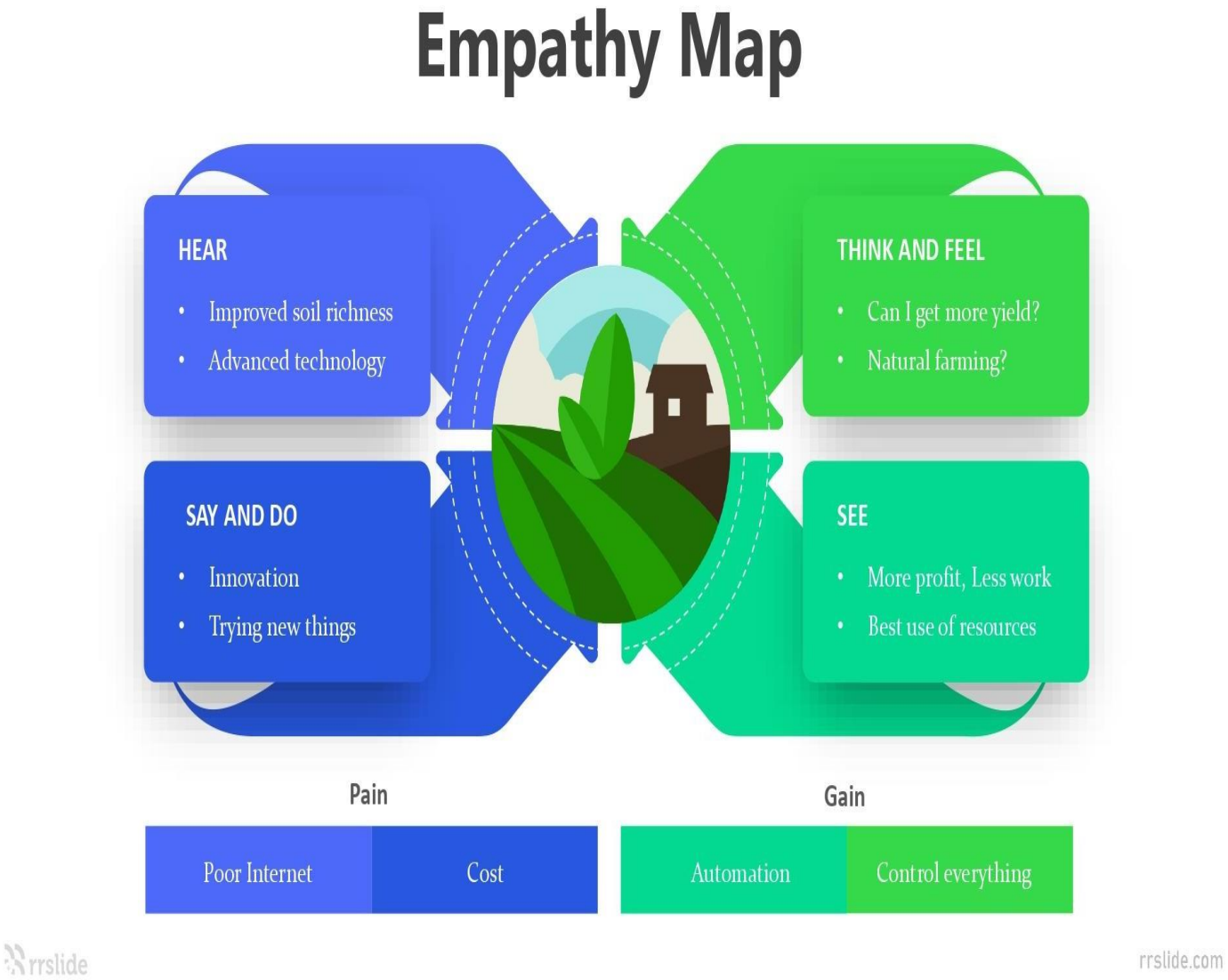
2.3 Problem statement Definition

Who does the problem affect?	Persons who do Agriculture
What are the boundaries of the problem?	Cope with climate change, soil erosion and biodiversity loss
What is the issue?	Loss of agricultural land and the decrease in the varieties of crops and livestock produced.
When does the issue occur?	Increasing pressures from climate change, soil erosion, its mostly starts from first day farming
Why is it important that we fix the problem?	It is required for the growth of better quality food products. It is important to maximize the crop yield. It is important to maintain soil richness
What solution to solve this issue?	An application is introduced to know about various data about their land remotely, where they can schedule some events for a month or a day. It also provides suggestions to users based on the crop they planted.
What methodology used to solve the issue?	Some search results info from internet based on crop planted. Arduino microcontroller to control the process and various sensors for data. An app built using MIT App Inventor

3. IDEATION & PROPOSED SOLUTION

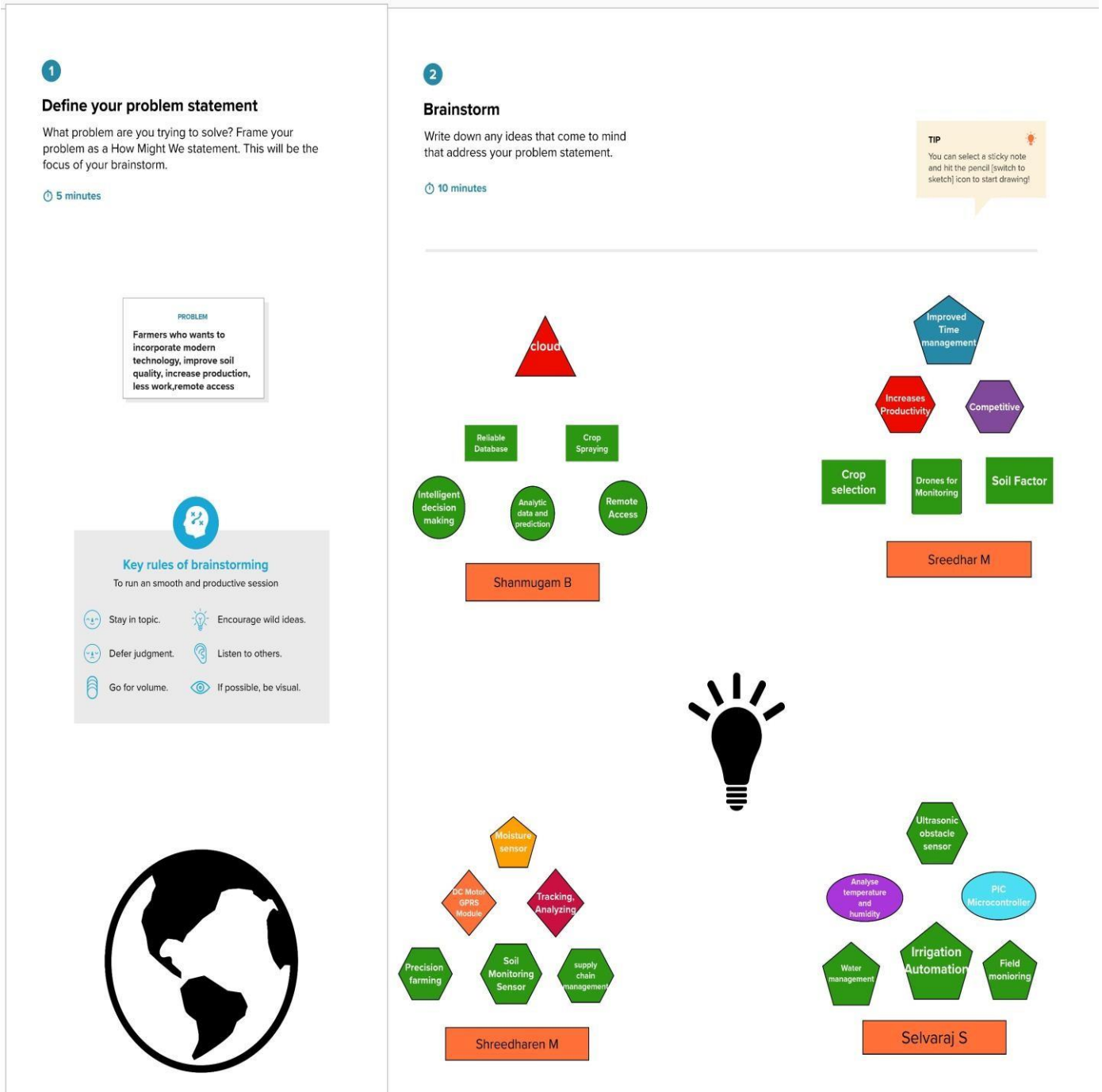
3.1 Empathy Map Canvas

An empathy map is **a collaborative tool teams can use to gain a deeper insight into their customers**. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.



3.2 Ideation & Brainstorming

Brainstorming is a method of generating ideas and sharing knowledge to solve a particular commercial or technical problem, in which participants are encouraged to think without interruption. Brainstorming is a group activity where each participant shares their ideas as soon as they come to mind.



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

3.3 Proposed Solution

Proposed solution should **relate the current situation to a desired result and describe the benefits that will accrue when the desired result is achieved**. So begin your proposed solution by briefly describing this desired result**.**

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To solve farmer issues like • Lack of Modernization and Mechanization • Invest in farm productivity and improving yield production. • Cope with climate change, soil erosion
2.	Idea / Solution description	An application and device is introduced to know about various data about their land remotely, where they can schedule some events for a month or a day. It also provides suggestions to users based on the crop they planted.
3.	Novelty / Uniqueness	Providing suggestions, Planning events
4.	Social Impact / Customer Satisfaction	Farmers can track and control their land, suggestions of next plant crops and improving yield gives satisfaction.
5.	Business Model (Revenue Model)	• It’s a subscription model, where user have to pay for their internet. • Customer services are supported • It supports third party devices also • Reach customers via Referral, Agents, Third party applications
6.	Scalability of the Solution	Our product is scalable with our devices (extra add-ons) as well as third party devices also. Ability to provides various features in a application like reports generation etc.

3.4 Problem Solution fit

Problem-Solution Fit - this occurs when you have evidence that customers care about certain jobs, pains, and gains. At this stage you've proved the existence of a problem and have designed a value proposition that addresses your customers' jobs, pains and gains. Unfortunately, you still do not have clear evidence that your customer really care enough about your value proposition enough to buy it.

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioural patterns and recognize what would work and why.

Purpose:

□ Solve complex problems in a way that fits the state of your customers.

□ Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behaviour

□ Sharpen your communication and marketing strategy with the right triggers and messaging.

□ Increase touch-points with your company by finding the right problem behaviour fit and building trust by solving frequent annoyances, or urgent or costly problems.

Understand the existing situation in order to improve it for your target group

Problem-Solution fit canvas 2.0

Team ID: PNT2022TMID28579

Project Title :- SmartFarmer - IoT Enabled Smart Farming Application

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 who want to use modern technology
Beginner farmers

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.

Initial Invest cost
Internet Access
Unable to access right resources
Don't know whether the product will work or not

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

Incorporate new technology in agriculture.
Need to gather information from various farmers
Need to use things that improve soil quality

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.

Maintain Crops and increase yield production
Provide remote access to their land
Improve soil quality

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.

No Modernization
Sticking to the old things
Cope with climate change
Decrease in soil quality

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)

Make sure that they know their requirements
Make sure that product meets their requirements
Cost of the product and performance
Scalability of the product
Customer service

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.

Farmers know to improve their soil quality and improve productivity.

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.

Before - Low production, Need to visit land daily.
After - High Production, No need to visit land daily.

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.

To design an application which helps to monitor and controls the land operations.
By using various sensors data are used to provide suggestions and current status of land.
To improve production, soil quality through our app.
Our solution allows the farmers to incorporate new technology.

8. CHANNELS of BEHAVIOUR — CH

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

Remote Access and Security

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

Make sure whether the product provides best solution and provides control to most of things.
Crop inspection and check their production.

Extract online & offline CH of BE

AMALTAMA

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
1	User Registration	Registration Through Gmail
2	User Confirmation	Confirmation Via Email Confirmation Via OTP
3	User Login	Login with Email Id and Password
4	Forgot Password	Login with Email Confirmation Of OTP
5	Query Form	Make a note of the problems and issues faced by user when using the application
6	Weather	Make a note of the problems and issues faced by user when using the application
7	Agro Note	To list of agriculture related information like how to plant, how much litres of water that plant need in a day etc.
8	Sensors	To show various data from different sensors like temperature, humidity, soil moisture
9	Database Management	To show various agriculture related data are stored
10	Exit	After user checked every information, user can exit the application

4.2 Non-Functional requirements

Following are the non-functional requirements of the proposed solution

FR No.	Non-Functional Requirement	Description
1	**Usability**	Effective and Easy to Use
2	**Security**	The process of protecting data from Unauthorized Access
3	**Reliability**	Consistency and Accuracy and the shared protection achieves a better trade-off between costs and reliability
4	**Performance**	Measured and estimate the performance of the Productivity
5	**Availability**	24/7 services
6	**Scalability**	Scalability is main concern for IoT platforms. It supports third party sensors. It can be easily scalable for large farming

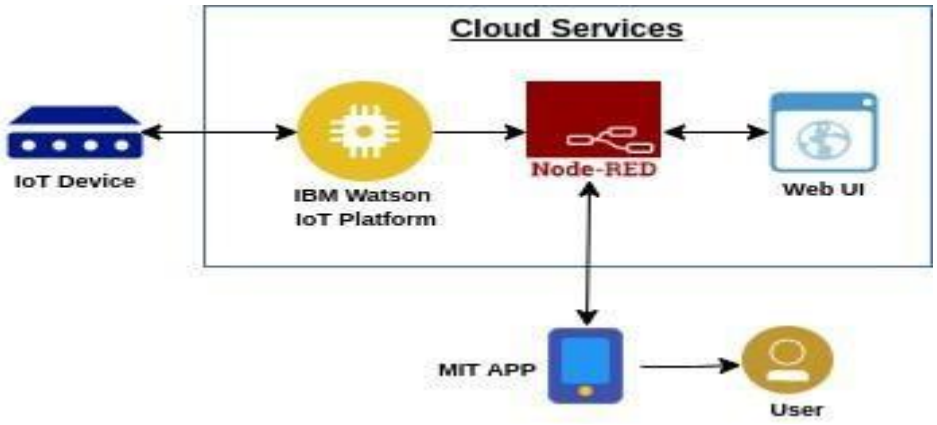
5. PROJECT DESIGN

5.1 Data Flow Diagrams

A data flow diagram (DFD) is **a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement**. They are often elements of a formal methodology such as Structured Systems Analysis and Design Method (SSADM).

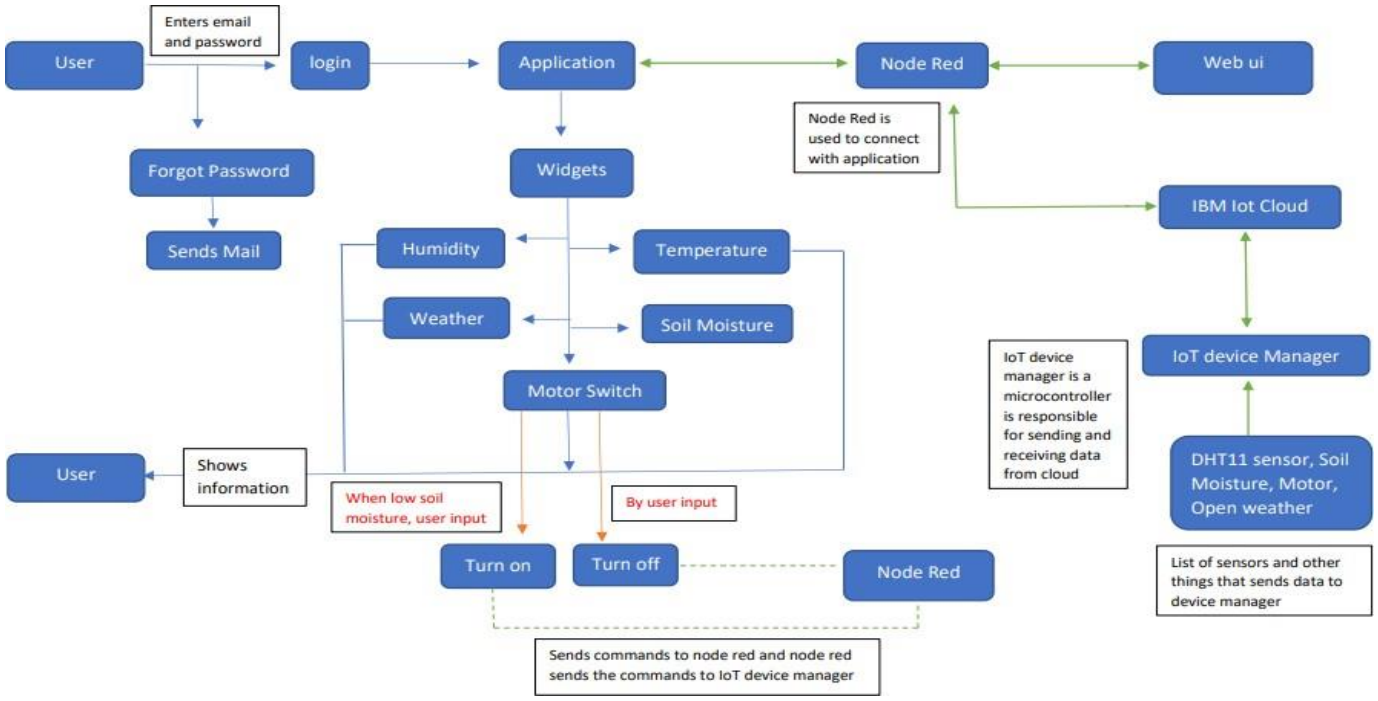
A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Simplified:

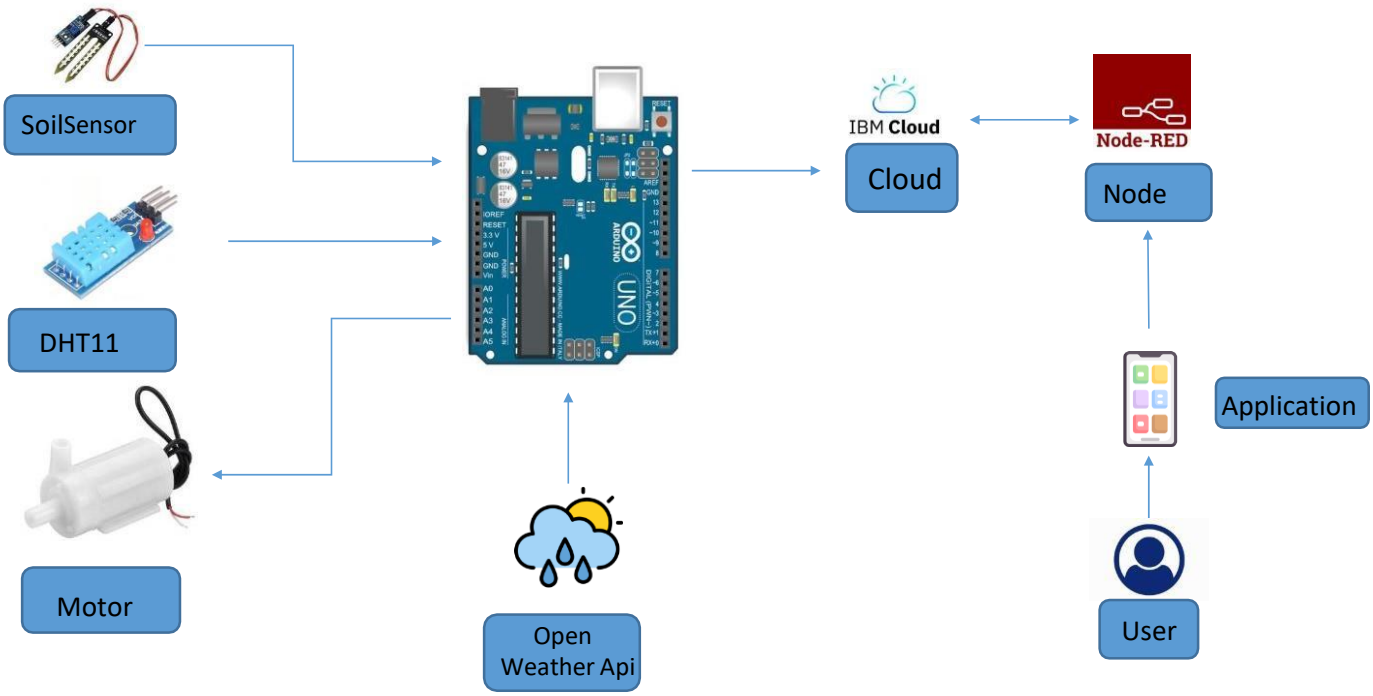


- Different parameters such as temperature, humidity, soil moisture are sensed using the sensors.
- Open weather API is used for collecting the weather information.
- Above data are processed with the help of microcontroller which is connected to internet.
- The processed data is updated to cloud for further process
- The IBM Watson IoT Platform is connected with node red services which is connected to the application.
- In application, user can see the parameters/data that obtained from sensors and APIs.
- With the help of application user can interact with IoT devices to perform some functions such turning ON & OFF motor.
- Web UI is also used for visualization of data.

Detailed DFD Level 0 (Industry Standard)



5.2 Solution Architecture



Technical Architecture

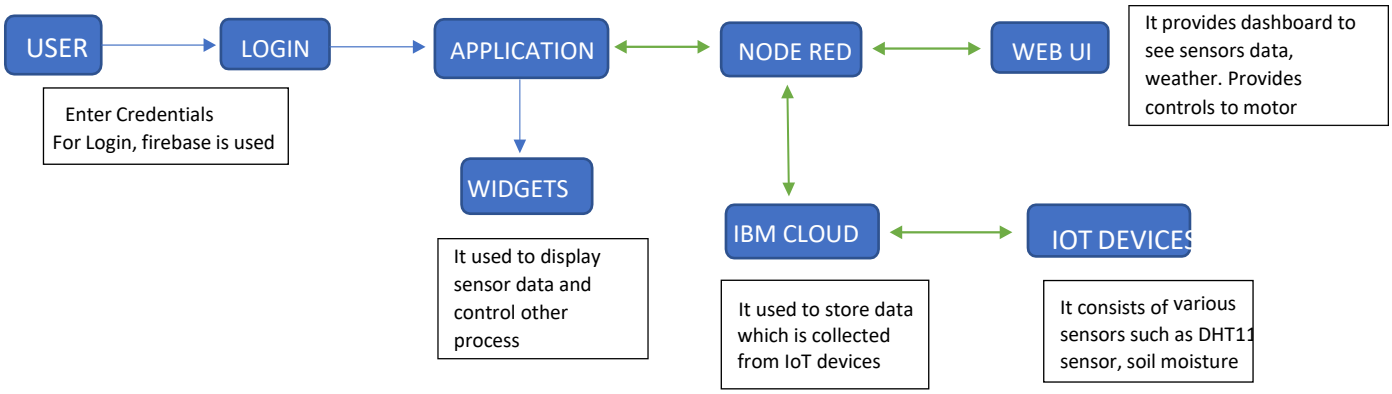


Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Mobile app. In our application, were data are displayed using widgets like structure. Users interacts with widgets to additional info	MIT App Inventor , React Native
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data base type	Firebase is Nosql database
6.	Cloud Database	Database Service on Cloud	Firebase, IBM Watson IoT Cloud Platform
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local File system
8.	External API-1	Purpose of the API is get to weather information	Open Weather API
9.	External API-2	Purpose of the API is to connect with firebase for login purpose	Firebase API
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, IBM Cloud, Firebase

11.	DHT11 sensor, Soil Moisture sensor	It used to monitor the soil, temperature, humidity.	

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Node Red, MIT App Inventor, Arduino IDE Node Red for connecting with application, MIT App Inventor for building app, Arduino is open source electronics platform to build hardware and software.	It is a software, which helps in connecting and building application. Node Red, MIT App Inventor, Arduino IDE.
2.	Security Implementations	HTTPS Connections, X-Force Red IoT Testing	Encryptions, Secured Connection
3.	Scalable Architecture	Architecture is scalable from 10 devices to 300 devices easily and account is also scalable upto thousand connections. For very high scalability we need to upgrade our cloud plan.	Firebase, IBM Cloud
4.	Availability	Availability of our application is 24/7 because which use a cloud technology. Firebase will use commercially reasonable efforts to make Firebase available with a Monthly Uptime Percentage of at least 99.95% and distributed servers.	Firebase, IBM Cloud
5.	Performance	No of requests is 2 requests per 20 seconds or 4 requests per 30 second and sometimes user request will be added with respective to the requests	MIT App Inventor, Node Red, Cloud

5.3 User Stories

What are user stories?

A user story is **an informal, general explanation of a software feature written from the perspective of the end user or customer**. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	As a user, I can register for the application by entering my email, password, and confirming my password.	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	Medium	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-1
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1

	Dashboard	USN-6	As a user I want to see everything in single widget		Medium	Sprint-2
		USN-7	As a user I want a organised widgets section		High	Sprint-2
		USN-8	As a user I want a graphical/pictorial representation		Low	Sprint-2
Customer (Web User)	Dashboard	USN-9	As a user I want a graphical representation of data for better understanding		High	Sprint-2
		USN-10	As a user I want to see a dashboard where I can customise myself	Dashboard with customisation	Low	Sprint-2
	IoTDeviceSetup	USN-10	Have to use a least sensor and get better output		High	Sprint-2
		USN-11	As a user, I need a low cost IoT devices for farming		High	Sprint-2
		USN-12	As a user, I need a multiple sensors for various data		High	Sprint-2

User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**
Customer Care Executive	User Problems	USN-13	As a user, I don’t how to use the application	Manual guide will be there	Medium	Sprint-3
		USN-14	As a user, I need my application to work on		High	Sprint-3

			most of the mobiles			
		USN-15	As a user, I am facing issue in the application	Query form will be there	High	Sprint-3
Administrator	Query Clarification	USN-16	As a admin, I give solutions to their queries		High	Sprint-3
	Particular Access	USN-17	As a admin, I give access only to authorised person		High	Sprint-3
	Connection with IoTdevices	USN-18	As a admin, I ensure the correct working of the devices. If any problem arises it will be shared to user		Medium	Sprint-4
Customer (Mobile user)	Application	USN-19	As a user, I need to control my devices	Commands for devices	High	Sprint-4
		USN-20	As a user, I need to control my devices		Low	Sprint-4
		USN-21	As a user, I need a more info about plants inside a application		Medium	Sprint-4

6. Project Planning & Scheduling

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	Creating of Login page in application	4	Highest	Jeevanantham Sathish
Sprint-1	Registration	USN-2	Developing logic for sign in and sign up and Database Integration	5	Highest	Jeevanantham Sathish
Sprint-1	Registration	USN-3	Testing the created sign in and sign up page in our app and Database Integration	3	High	Jeevanantham Sathish
Sprint-1	Login	USN-4	User can login into application by entering email and password	3	Medium	Jeevanantham Sathish
Sprint-2	IoT Device Setup	USN-5	Least Device and Better Output	2	Highest	Madhangokul Jeevanantham
Sprint-2	Dashboard	USN-6	Graphical / Pictorial Representation for app and web ui	3	Low	Madhangokul Jeevanantham
Sprint-2	IoT Device Setup	USN-7	Low cost setup	2	Highest	Madhangokul Jeevanantham
Sprint-2	Dashboard	USN-8	Single widget Representation	5	Medium	Madhangokul Jeevanantham
Sprint-2	Dashboard	USN-9	Organized widget section	3	Highest	Madhangokul Jeevanantham
Sprint-3	IoT Device Setup	USN-10	Multiple sensors in setup	2	Highest	Jeeva anand Sathish

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	User Problems	USN-11	Manual Guide creation for application	3	Medium	Jeeva anand Sathish
Sprint-3	Query Clarification	USN-12	Solution to the queries	4	High	Jeeva anand Sathish
Sprint-3	User Problems	USN-13	Query form in the application	2	High	Jeeva anand Sathish
Sprint-3	Application	USN-14	Provide Commands through application	4	Highest	Jeeva anand Sathish
Sprint-4	Particular Access	USN-15	Only authorized person access	4	High	Madhan gokul Jeeva anand
S	User Problems	USN-16	Testing the application in multiple platform and ensure the working	3	High	Madhan gokul Jeeva anand
Sprint-4	Connection with IoT devices	USN-17	Testing the hardware setup and ensure the working	4	Medium	Madhan gokul Jeeva anand
Sprint-4	Application	USN-18	Agricultural Notes	4	Medium	Madhan gokul Jeeva anand

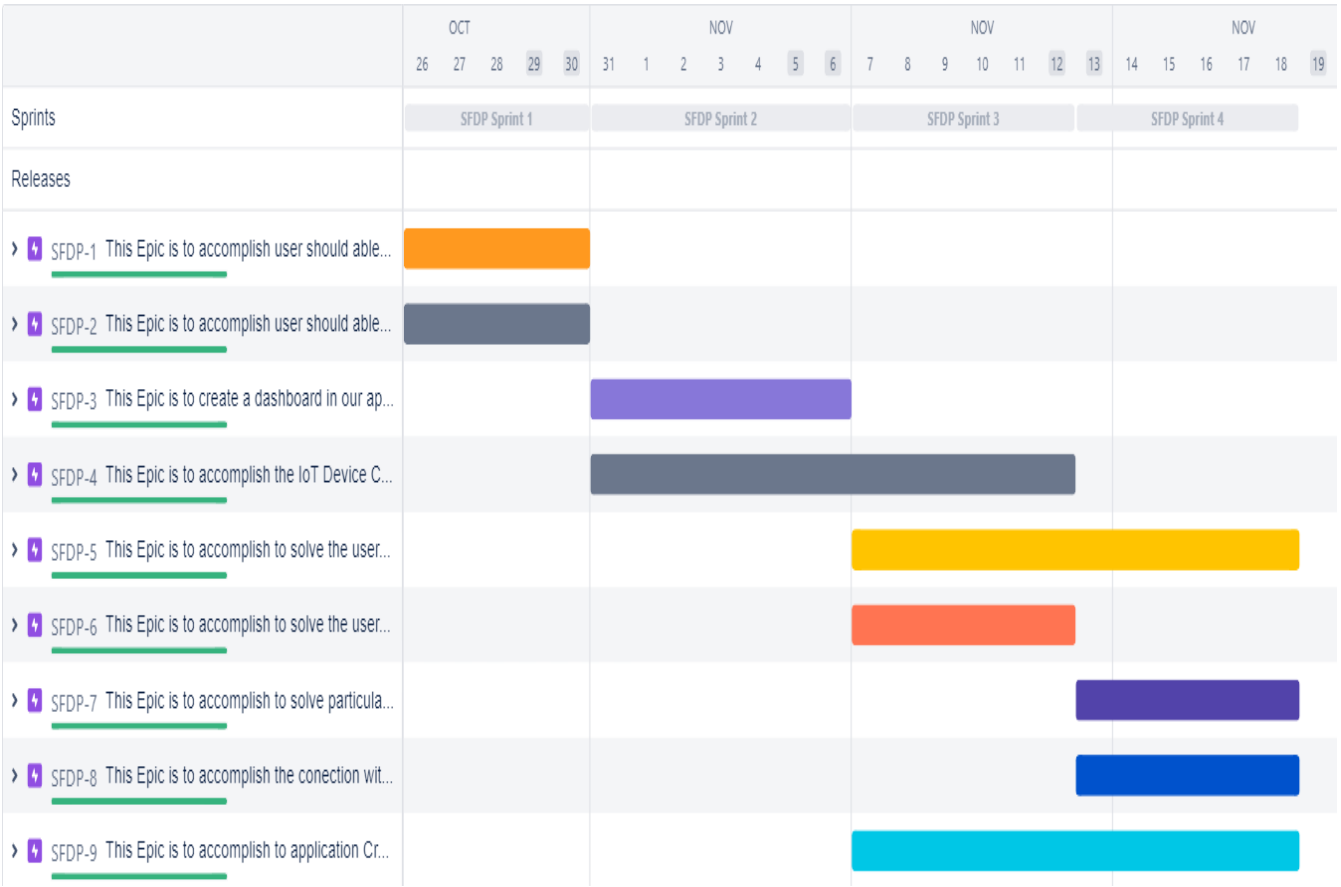
6.2 Sprint Delivery Schedule

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	15	5 Days	26 Oct 2022	30 Oct 2022	15	30 Oct 2022
Sprint-2	15	7 Days	31 Oct 2022	06 Nov 2022	15	07 Nov 2022
Sprint-3	15	6 Days	07 Nov 2022	12 Nov 2022	15	13 Nov 2022

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-4	15	6 Days	13 Nov 2022	18 Nov 2022	15	18 Nov 2022 – 19 Nov 2022

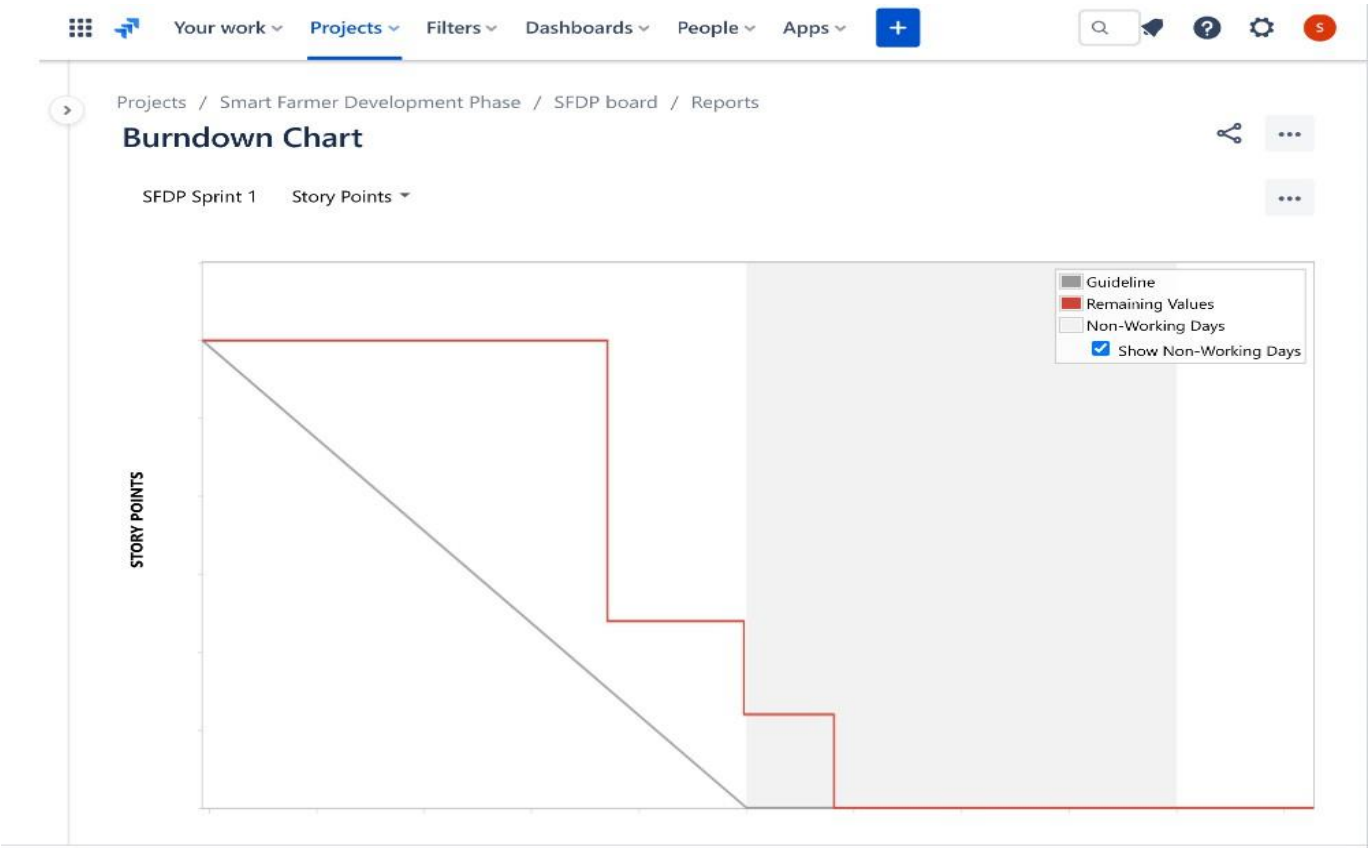
6.3 Reports from Jira

6.3a Roadmap

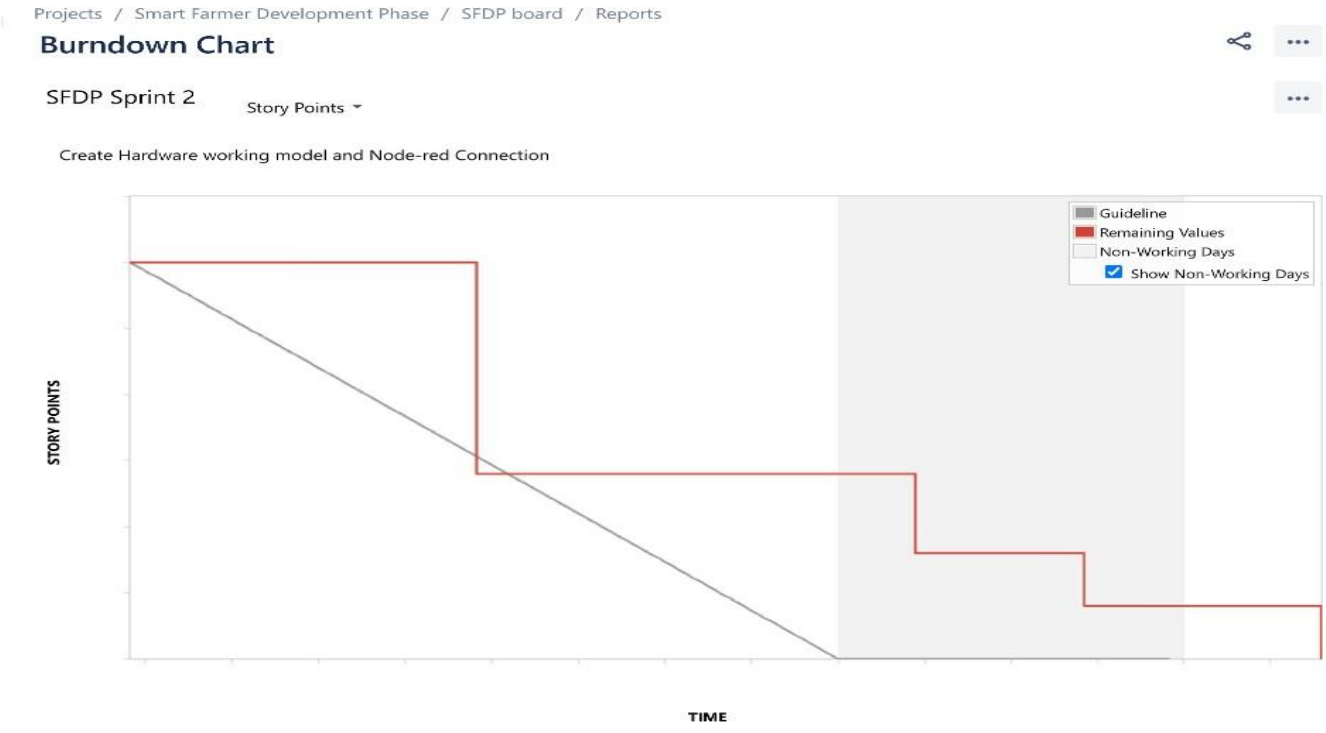


6.3b Burn down Chart

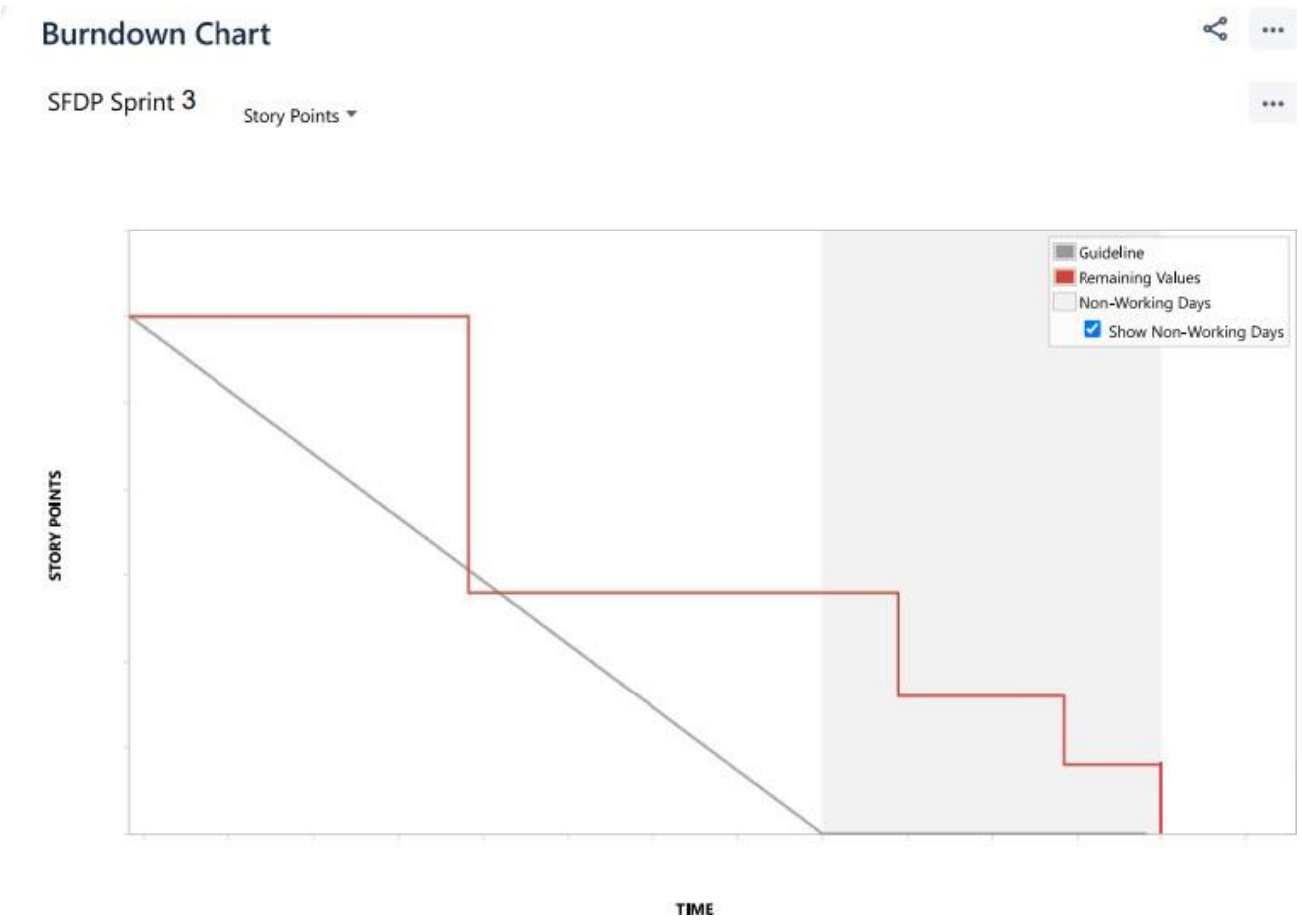
Sprint 1



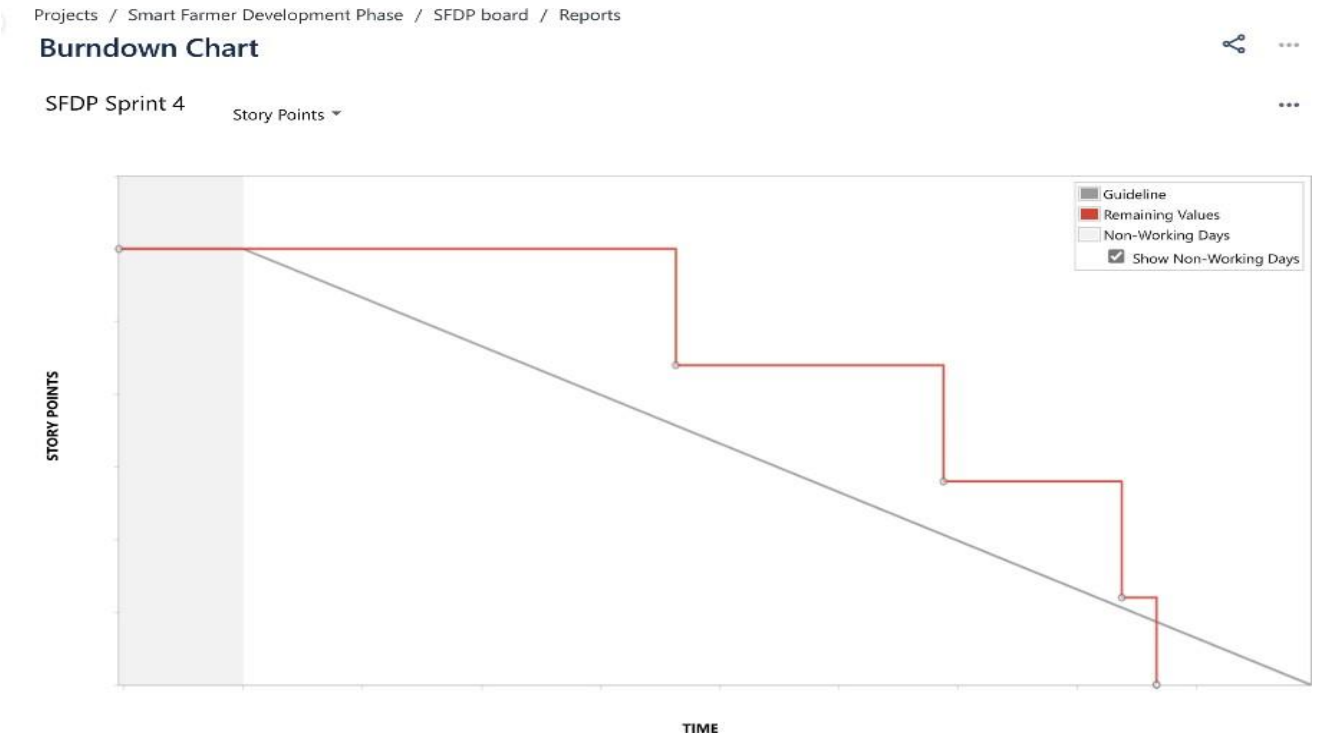
Sprint 2



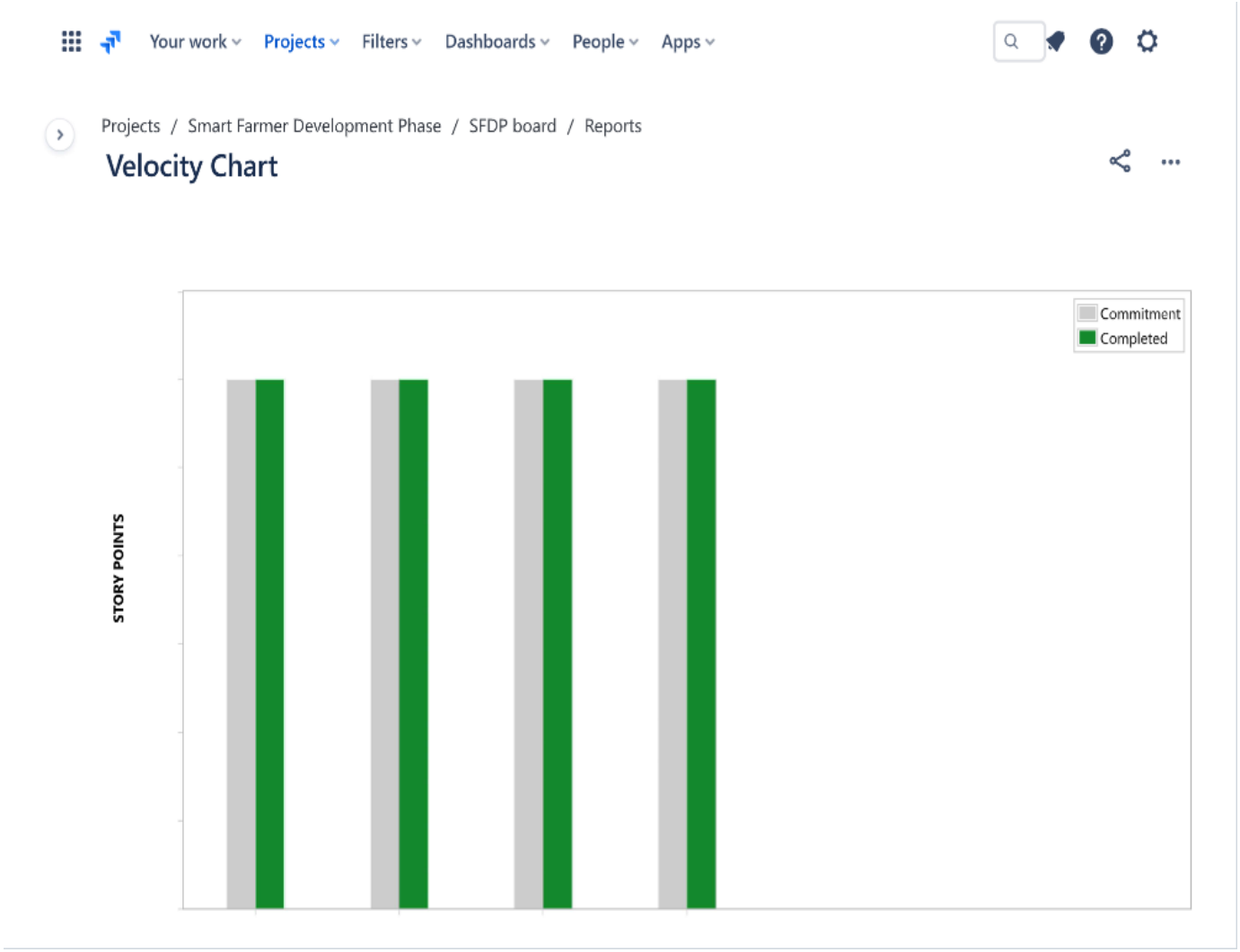
Sprint 3

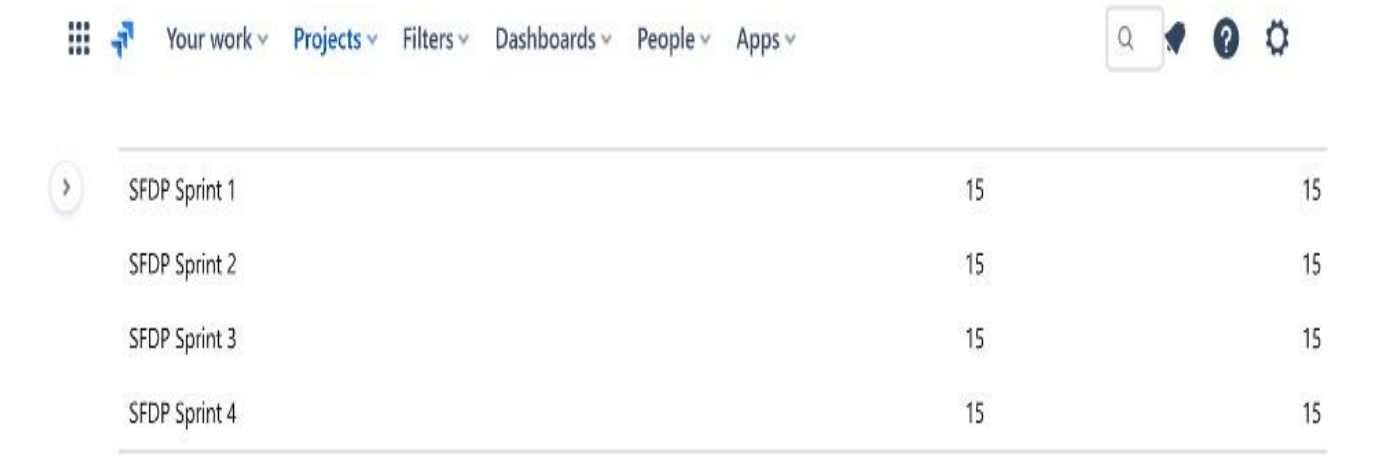


Sprint 4



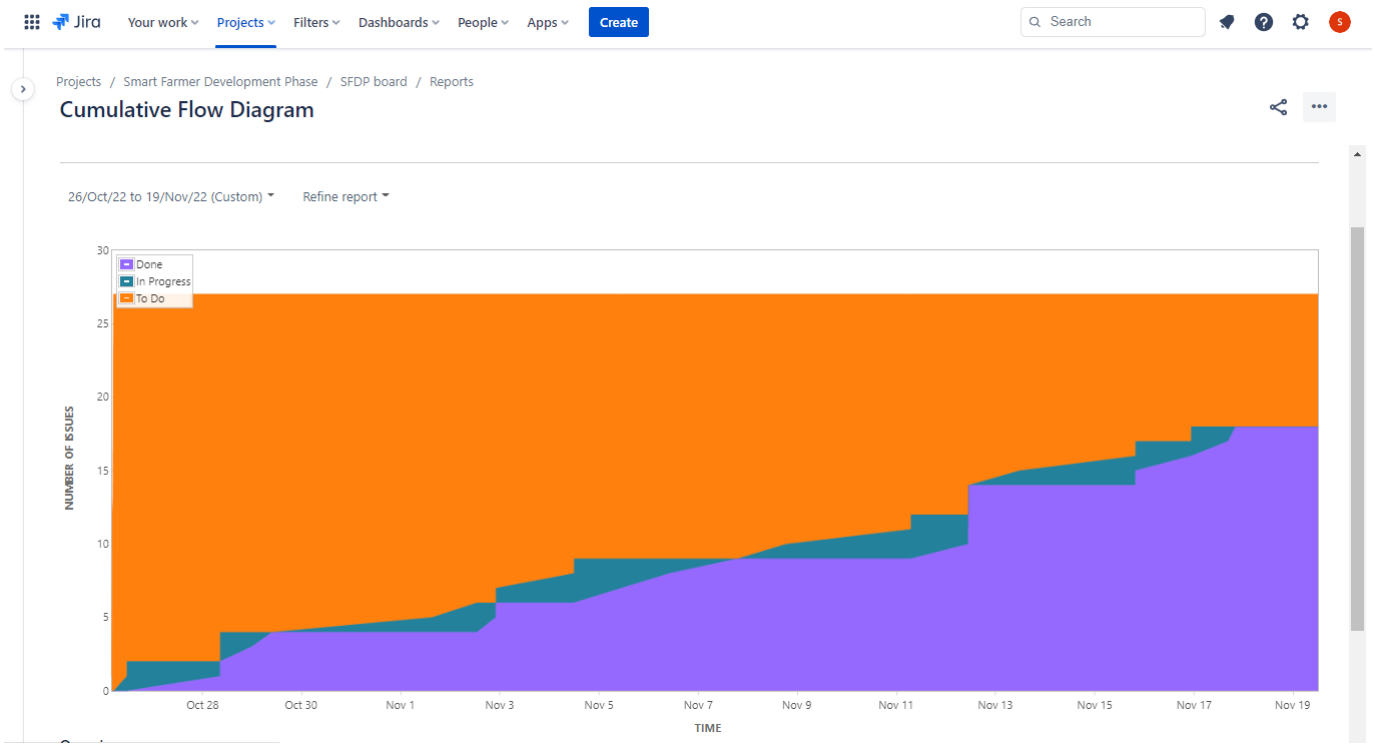
Velocity Chart





SFDP Sprint 1	15	15
SFDP Sprint 2	15	15
SFDP Sprint 3	15	15
SFDP Sprint 4	15	15

Cumulative Flow Diagram



7 CODING & SOLUTIONING

7.1 Feature 1 (Open weather API)

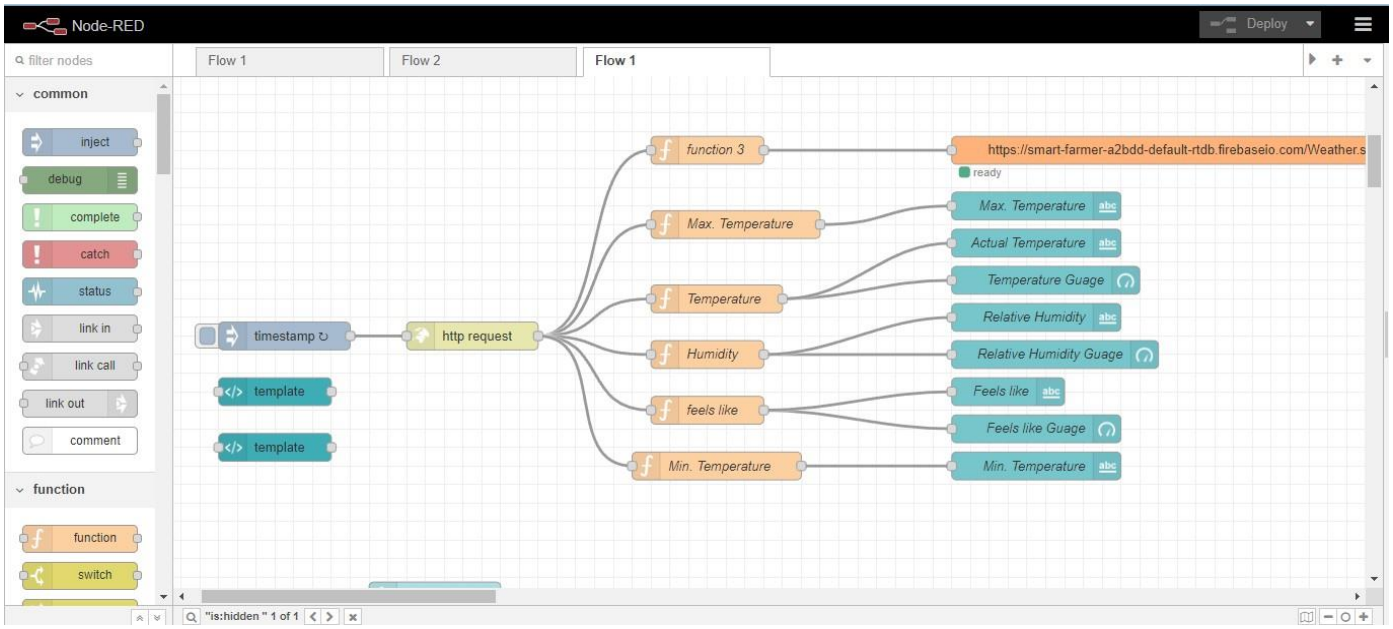


Figure 1

Open weather api provides various insights about the farm located area. It gives data like Min Temp, Max Temp, Weather, Humidity etc.

Example open weather api output:

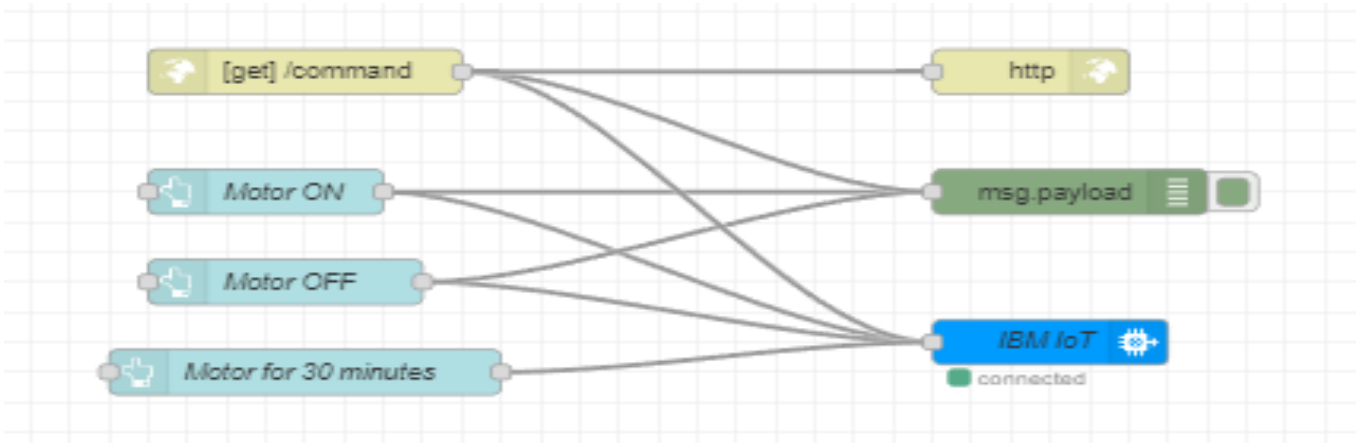
We request open weather api it returns object data

{"coord":{"lon":80.2785,"lat":13.0878},"weather":[{"id":721,"main":"Haze","description":"haze","icon":"50d"}],"base":"stations","main":{"temp":301.14,"feels_like":303.16,"temp_min":301.14,"temp_max":301.14,"pressure":1008,"humidity":65},"visibility":5000,"wind":{"speed":5.66,"deg":20},"clouds":{"all":75},"dt":1668857418,"sys":{"type":1,"id":9218,"country":"IN","sunrise":1668818370,"sunset":1668859751},"timezone":19800,"id":1264527,"name":"Chennai","cod":200}

We don't need all data, we extracted the needed ones by the help of the function, it also shown figure 1.

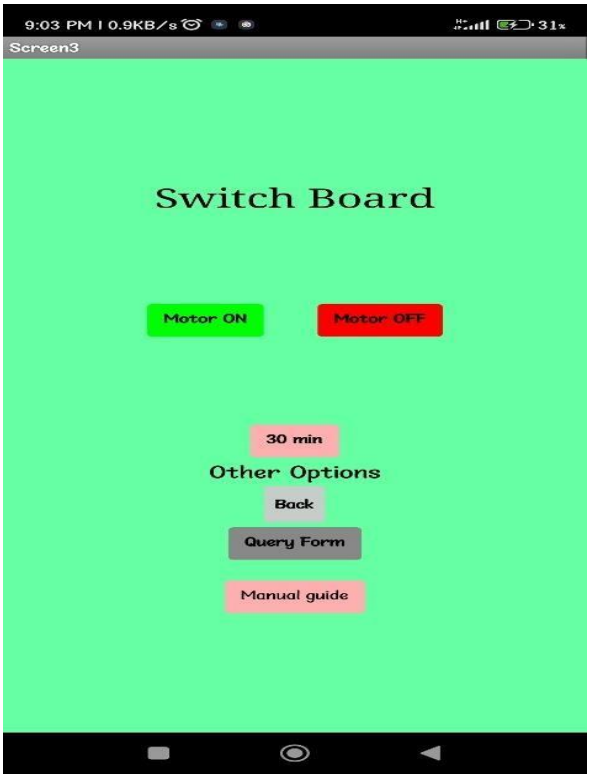
7.2 Feature 2 (Motor On/OFF/30 minutes)

```
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status = cmd.data['command']
    if status == "motoron":
        print("motor is on")
    elif status == "motoroff":
        print("motor is off")
    elif status == "motor30":
        print("motor is on for 30 minutes")
        print("motor Started")
        for i in range(1,31):
            print("%d minutes to stop"%(30-i))  # use time.sleep(60) for delay of one minute in each iteration
        print("motor stopped")
```



Farmers can control their motor in three ways one is motor on, motor off, motor for 30 minutes where they can run motor for 30 minutes and motor will automatically off.

7.3 Feature 3 (Query form and Manual Guide)



If any queries user can fill the form by clicking query form, the smartfarmer team resolve it as soon as possible.

Manual guide is also there, which guide the farmers/users to know how to use the app and functions.

8 Testing

8.1 Test Cases

Test Id	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	Test For Automation (Y/N)	Executed By
Login 1	UI	Authorization	Verify User Is Able To See Login Page	1. Open The Application		User Able To See The Login Page	Working As Expected	Pass		Y	Jeevanantham Jeeva anand Madhan gokul Sathish
Login 2	UI	Authorization	Verify User Is Able To See Login Page	1.Open The Application		Application Should Show Below Elements 1.Username 2.Password 3.Login 4.Register	Everything Is There Working As Expected	Pass		N	Jeevanantham Jeeva anand Madhan gokul Sathish
Login3	Functional	Authorization	User Able To Register The Account	1.Open The Application 2.Enter The Username And Password 3.Click Signup Button	Username: Dondon Password: 12345678	User Able To Sign Up And Now They Can Login By Clicking Login Button	Working as expected	Pass		Y	Jeevanantham Jeeva anand Madhan gokul Sathish
Login 4	Functional	Authorization	User Able To Sign Up And Now They Can Login By Clicking Login Button	1.Open The Application 2.Enter The Username And Password 3.Click Login Button	Username: Dondon123 Password:123456789 The Given Username Is Already Registered One	User Able To Go To Next Screen	Working as expected	Pass		Y	Jeevanantham Jeeva anand Madhan gokul Sathish

Login 5	Functional	Authorization	User Able To Go To Next Scree n	1. Open The Application. 2a. Enter The Username And Invalid Password And Click Login Button 2b. He Enter The Invalid Username And Correct Password.	2a.Username:Dondon Password: Asdfghsjjy 2a. Username: Dondon1 Password: 12345678	A. User Can't Login B. User Can't Login	Working as expected	Pass		Y	Jeevananthm Jeeva anand Madhan gokul Sathish
Home1	UI		Verif y theUI elements	1.Open The Application 2.Enter The Username And Password 3.Click Login Button	Username:dondon password: 12345678	Verify the below UI elements 1. weather 2. humidity 3. temperature 4. Soil moisture 5. Buttons	Everything isok	Pass		N	Jeevanantham Jeeva anand Madhan gokul Sathish
Home2	Functional		User able to navigate across the screen User able to click the buttons	1.Open The Application 2.Enter The Username And Password 3.Click Login Button 4.check the button and navigation		1. Buttonis clickable 2. Go tonext screen	Working as expected	Pass	Y		Jeevanantham Jeeva anand Madhan gokul Sathish

Home 3	Functional		User able to access query form	1. User need to log in 2.click on query form button		Able to access query form	Working as expected	Pass			Jeevanantham Jeeva anand Madhan gokul Sathish
Home 4	Functional		User able to access manual guide	1. User need to log in 2. Click on manual guide button		Able to access manual guide	Working As expected	Pass			Jeevanantham Jeeva anand Madhan gokul Sathish

8.2 User Acceptance Testing

UAT Execution & Report Submission

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the SmartFarmer project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	2	0	0	7
Duplicate	1	0	0	0	1
External	0	1	0	1	2
Fixed	11	0	0	0	11
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	17	3	0	1	21

3. Test Case Analysis

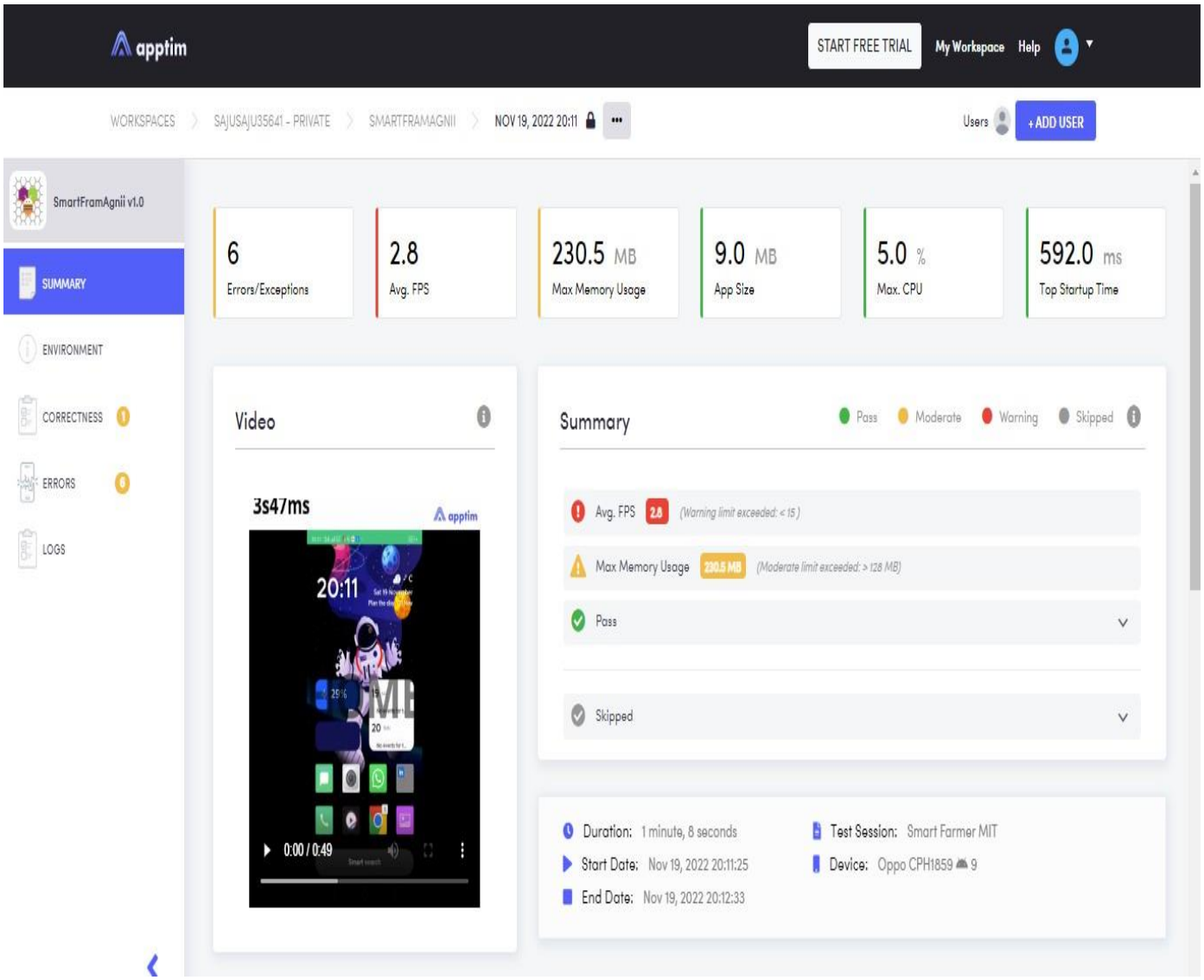
This report shows the number of test cases that have passed, failed, and untested

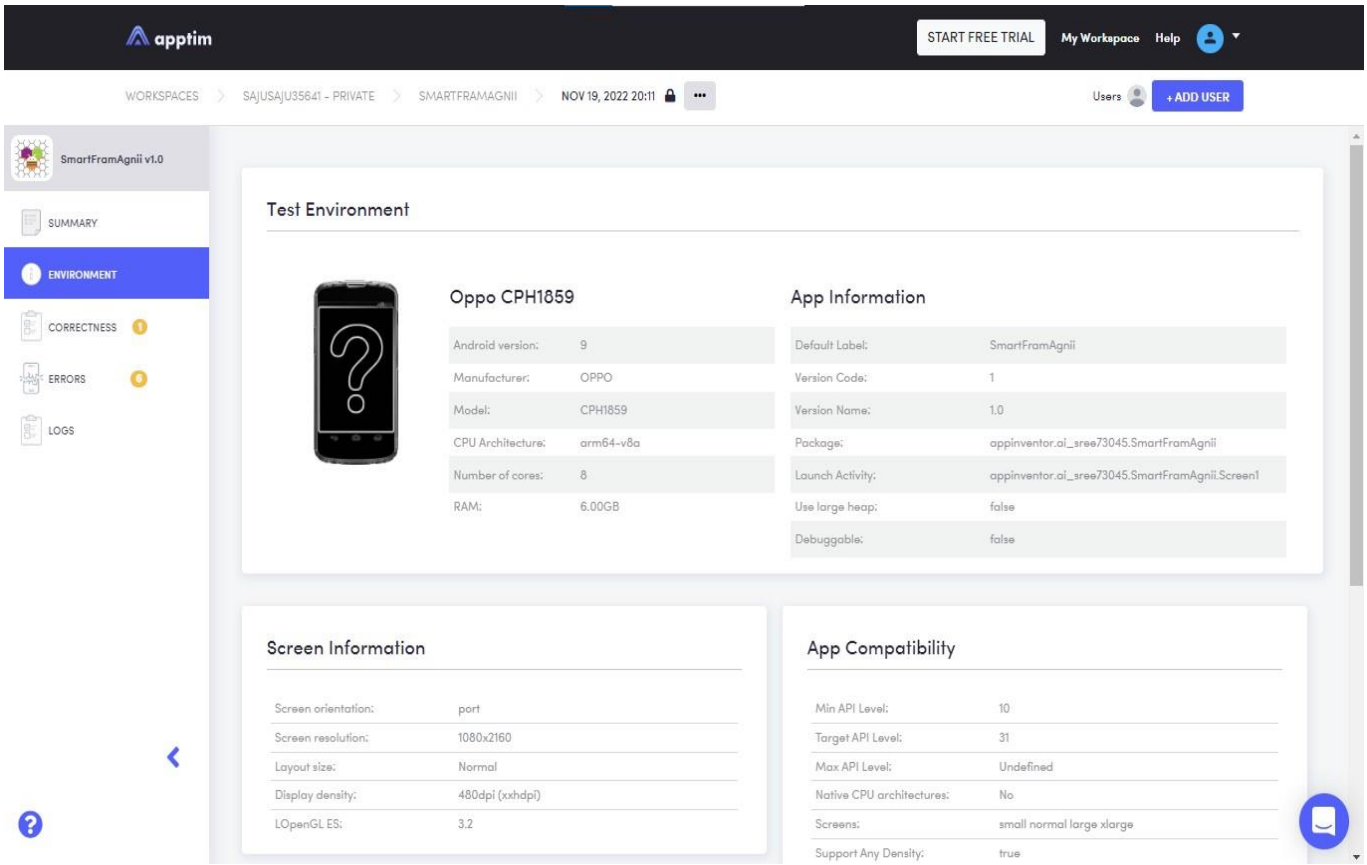
Section	Total Cases	Not Tested	Fail	Pass
Authorization (MIT Based)	5	0	0	5
Home Page (MIT Based)	4	0	0	9

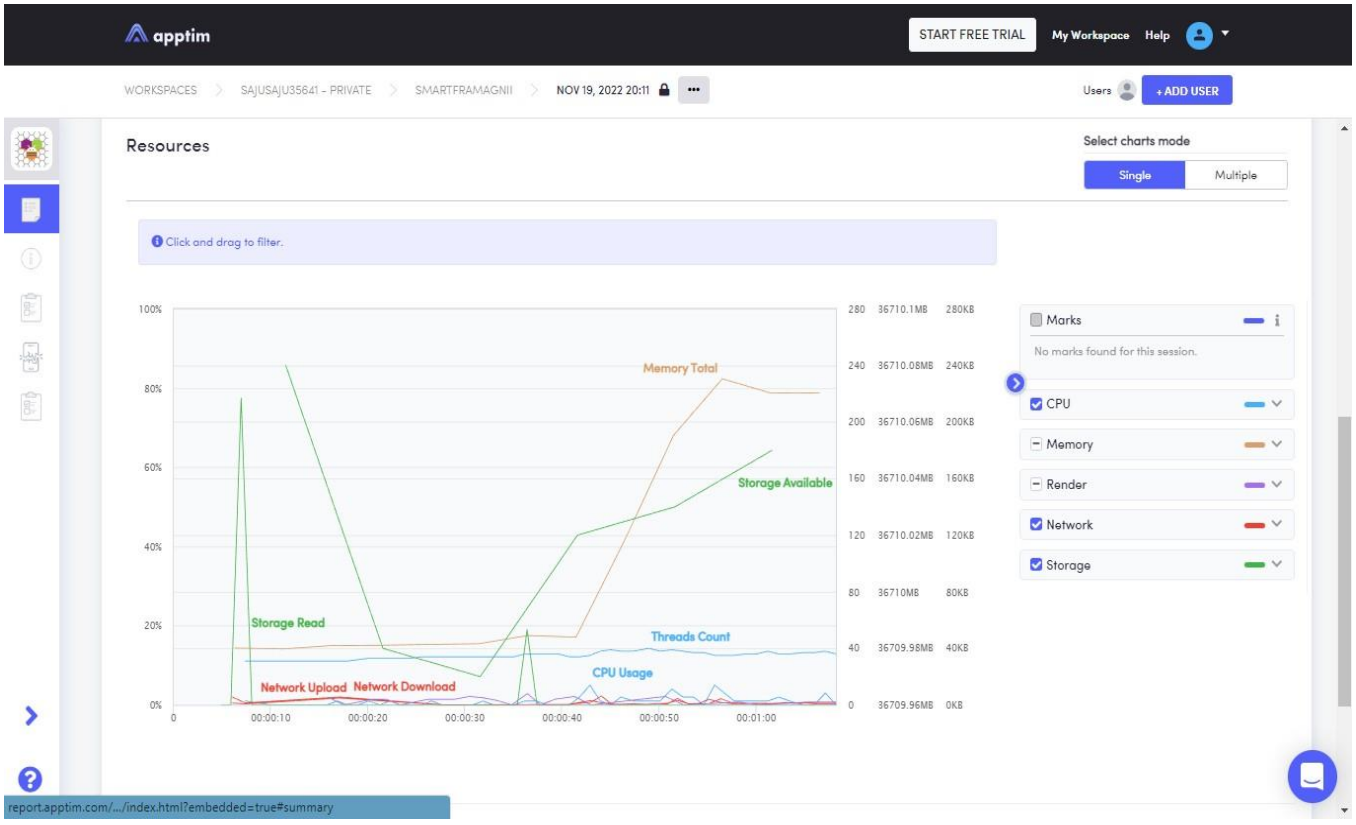
9. RESULTS

9.1 Performance Metrics

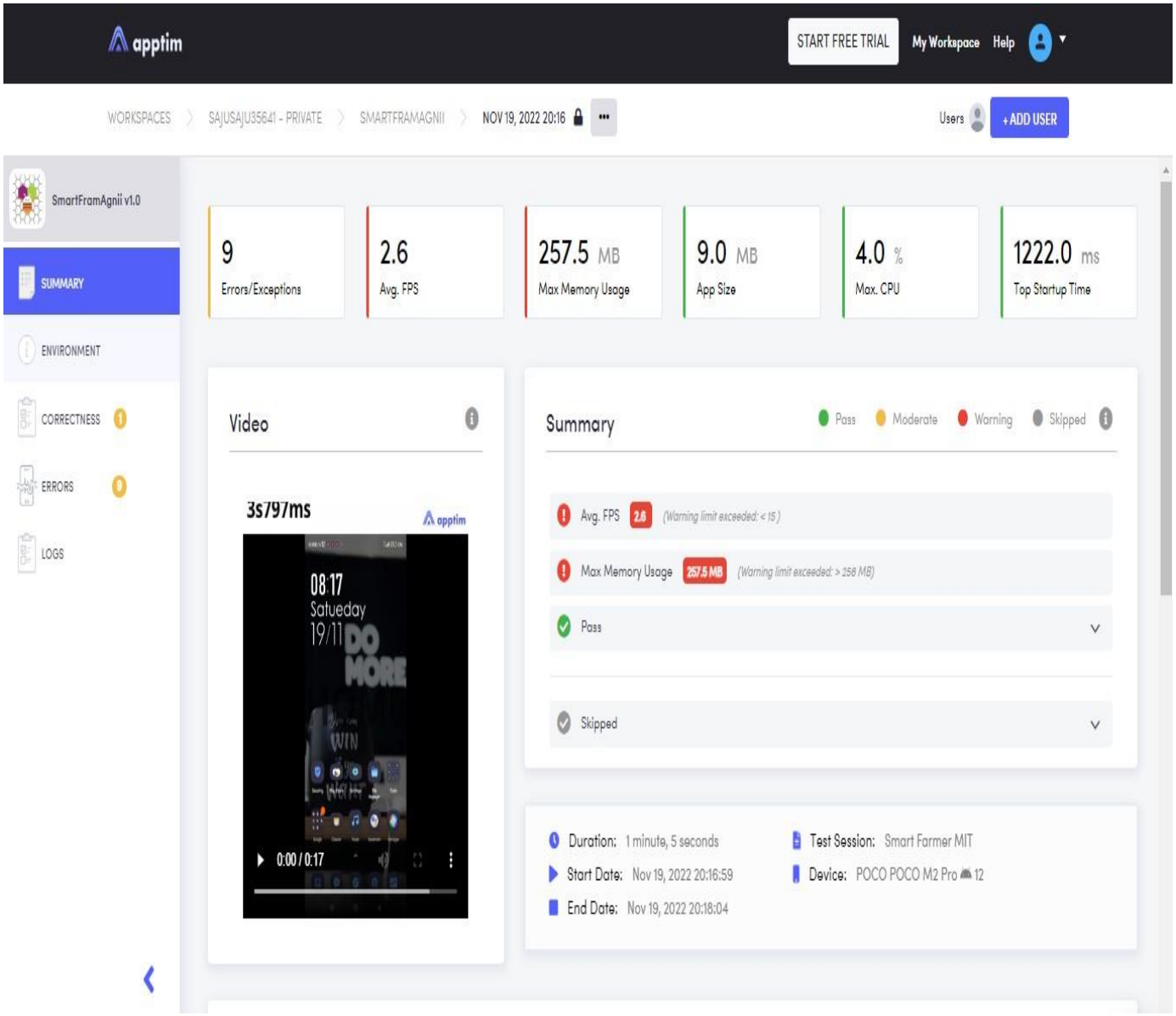
MOBILE 1

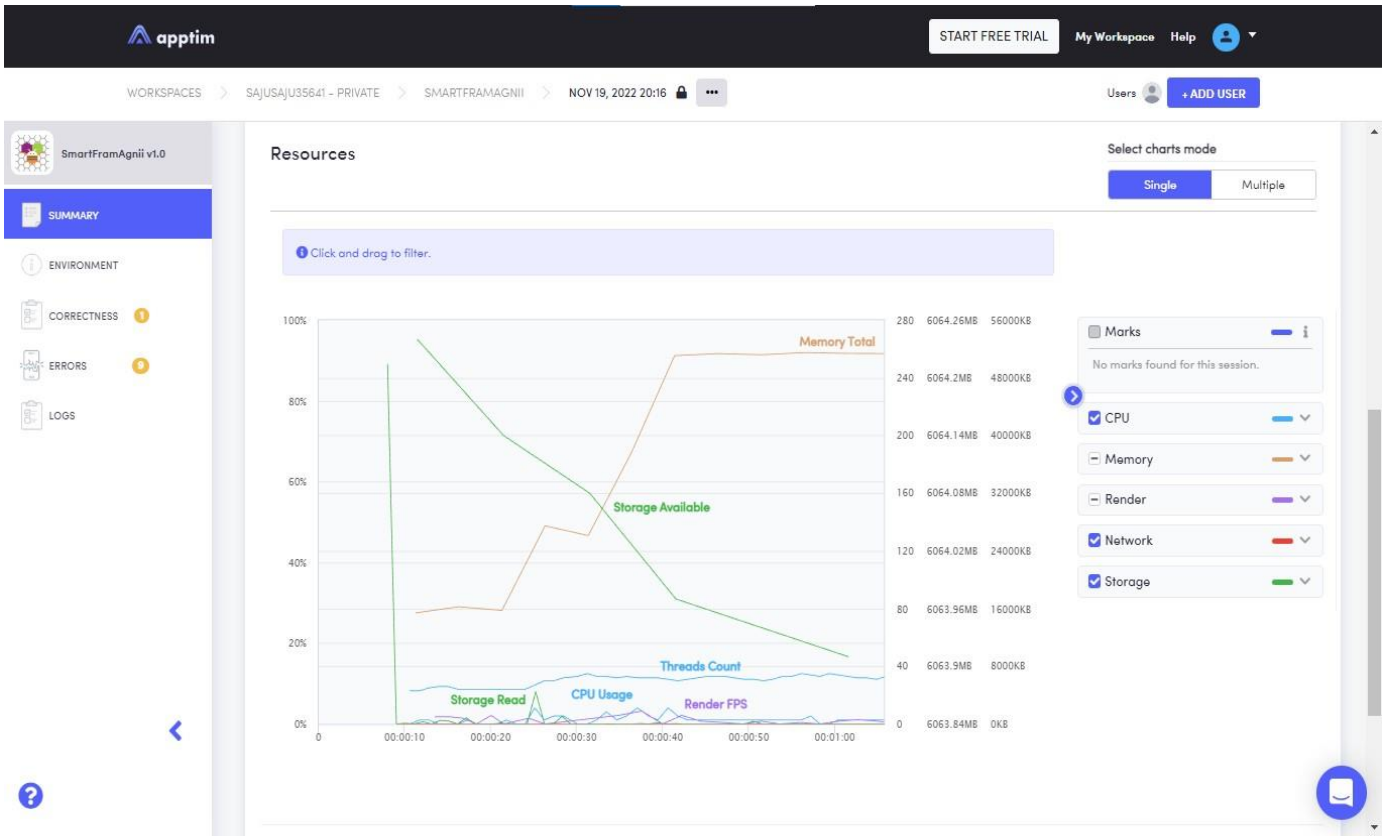


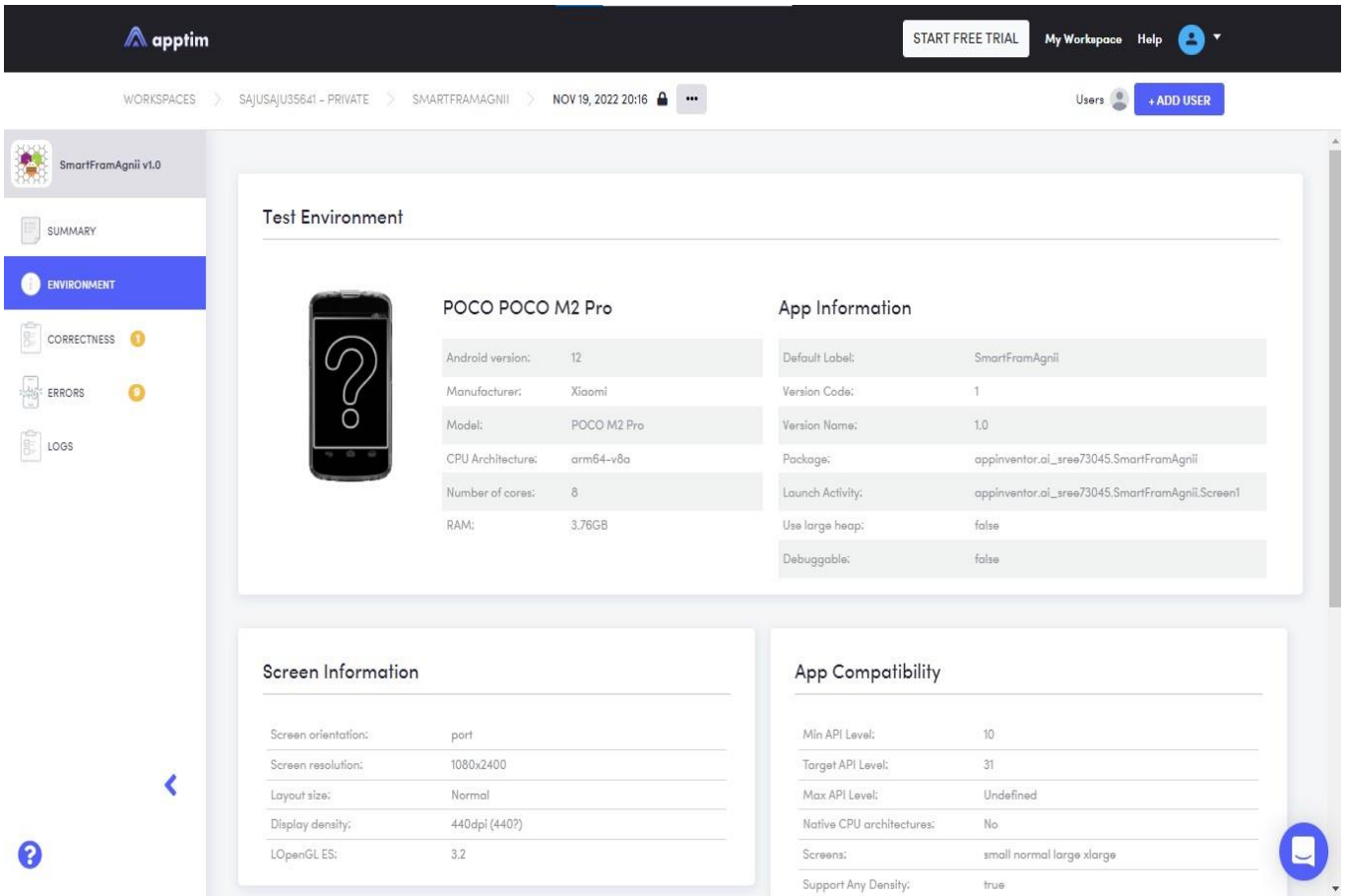

apptim
START FREE TRIAL
My Workspace
Help
WORKSPACES
SAJUSAJU35641 - PRIVATE
SMARTFRAMAGNII
NOV 19, 2022 20:11
Users
+ ADD USER
SmartFramAgnii v1.0
SUMMARY
ENVIRONMENT
CORRECTNESS
ERRORS
LOGS
Test Environment
Oppo CPH1859
Android version: 9
Manufacturer: OPPO
Model: CPH1859
CPU Architecture: arm64-v8a
Number of cores: 8
RAM: 6.00GB
App Information
Default Label: SmartFramAgnii
Version Code: 1
Version Name: 1.0
Package: appinventor.ai_sree73045.SmartFramAgnii
Launch Activity: appinventor.ai_sree73045.SmartFramAgnii.Screen1
Use large heap: false
Debuggable: false
Screen Information
Screen orientation: port
Screen resolution: 1080x2160
Layout size: Normal
Display density: 480dpi (xxhdpi)
LOpenGL ES: 3.2
App Compatibility
Min API Level: 10
Target API Level: 31
Max API Level: Undefined
Native CPU architectures: No
Screens: small normal large xlarge
Support Any Density: true


apptim
START FREE TRIAL
My Workspace
Help
WORKSPACES
SAJUSAJU35641 - PRIVATE
SMARTFRAMAGNII
NOV 19, 2022 20:11
Users
+ ADD USER
Resources
Select charts mode
Single
Multiple
Click and drag to filter.
Memory Total
Storage Available
Storage Read
Threads Count
CPU Usage
Network Upload
Network Download
Marks
No marks found for this session.
CPU
Memory
Render
Network
Storage
report.apptim.com/.../index.html?embedded=true#summary

MOBILE 2


apptim
START FREE TRIAL
My Workspace
Help
WORKSPACES
SAJUSAJU35641 - PRIVATE
SMARTFRAMAGNII
NOV 19, 2022 20:16
Users
+ ADD USER
SmartFramAgnii v1.0
SUMMARY
ENVIRONMENT
CORRECTNESS
ERRORS
LOGS
9
Errors/Exceptions
2.6
Avg. FPS
257.5 MB
Max Memory Usage
9.0 MB
App Size
4.0 %
Max. CPU
1222.0 ms
Top Startup Time
Video
3s797ms
apptim
08:17
Satueday
19/11
DO
MORE
WIN
0:00 / 0:17
Summary
Pass
Moderate
Warning
Skipped
Avg. FPS
2.6
(Warning limit exceeded: < 15)
Max Memory Usage
257.5 MB
(Warning limit exceeded: > 256 MB)
Pass
Skipped
Duration: 1 minute, 5 seconds
Start Date: Nov 19, 2022 20:16:59
End Date: Nov 19, 2022 20:18:04
Test Session: Smart Farmer MIT
Device: POCO POCO M2 Pro 12


apptim
START FREE TRIAL
My Workspace
Help
WORKSPACES
SAJUSAJU35641 - PRIVATE
SMARTFRAMAGNII
NOV 19, 2022 20:16
Users
+ ADD USER
SmartFramAgnii v1.0
SUMMARY
ENVIRONMENT
CORRECTNESS
ERRORS
LOGS
Resources
Select charts mode
Single
Multiple
Click and drag to filter.
Memory Total
Storage Available
Threads Count
CPU Usage
Storage Read
Render FPS
Marks
No marks found for this session.
CPU
Memory
Render
Network
Storage


apptim
START FREE TRIAL
My Workspace
Help
WORKSPACES
SAJUSAJU35641 - PRIVATE
SMARTFRAMAGNII
NOV 19, 2022 20:16
Users
+ ADD USER
SmartFramAgnii v1.0
SUMMARY
ENVIRONMENT
CORRECTNESS
ERRORS
LOGS
Test Environment
POCO POCO M2 Pro
Android version: 12
Manufacturer: Xiaomi
Model: POCO M2 Pro
CPU Architecture: arm64-v8a
Number of cores: 8
RAM: 3.76GB
App Information
Default Label: SmartFramAgnii
Version Code: 1
Version Name: 1.0
Package: appinventor.ai_sree73045.SmartFramAgnii
Launch Activity: appinventor.ai_sree73045.SmartFramAgnii.Screen1
Use large heap: false
Debuggable: false
Screen Information
Screen orientation: port
Screen resolution: 1080x2400
Layout size: Normal
Display density: 440dpi (440?)
LOpenGL ES: 3.2
App Compatibility
Min API Level: 10
Target API Level: 31
Max API Level: Undefined
Native CPU architectures: No
Screens: small normal large xlarge
Support Any Density: true

10. ADVANTAGES:

Farms can be monitored and controlled remotely.

• Increase in convenience to farmers.

• Less labor cost.

Better standards of living.

Increase in yield and production.

Work made easy.

DISADVANTAGES

Lack of internet/connectivity issues.

• Added cost of internet and internet gateway infrastructure.

• Farmers wanted to adapt the use of WebApp

11. CONCLUSION

Thus the objective of the project to implement an IoT system in order to help farmers to control and monitor their farms remotely has been implemented successfully.

12. FUTURE SCOPE

In future, more different sensors can be integrated in order to give more insights about the farm.

In application, we display the market trends and suitable plant for next planting based on real time data it can done by data analytics. To work standalone we can add solar panel to the hardware setup for own power generation. Camera can also be added to the project to monitor their farms very easily and also they can know what is currently happening.

13.APPENDIX

13 a Source Code :-

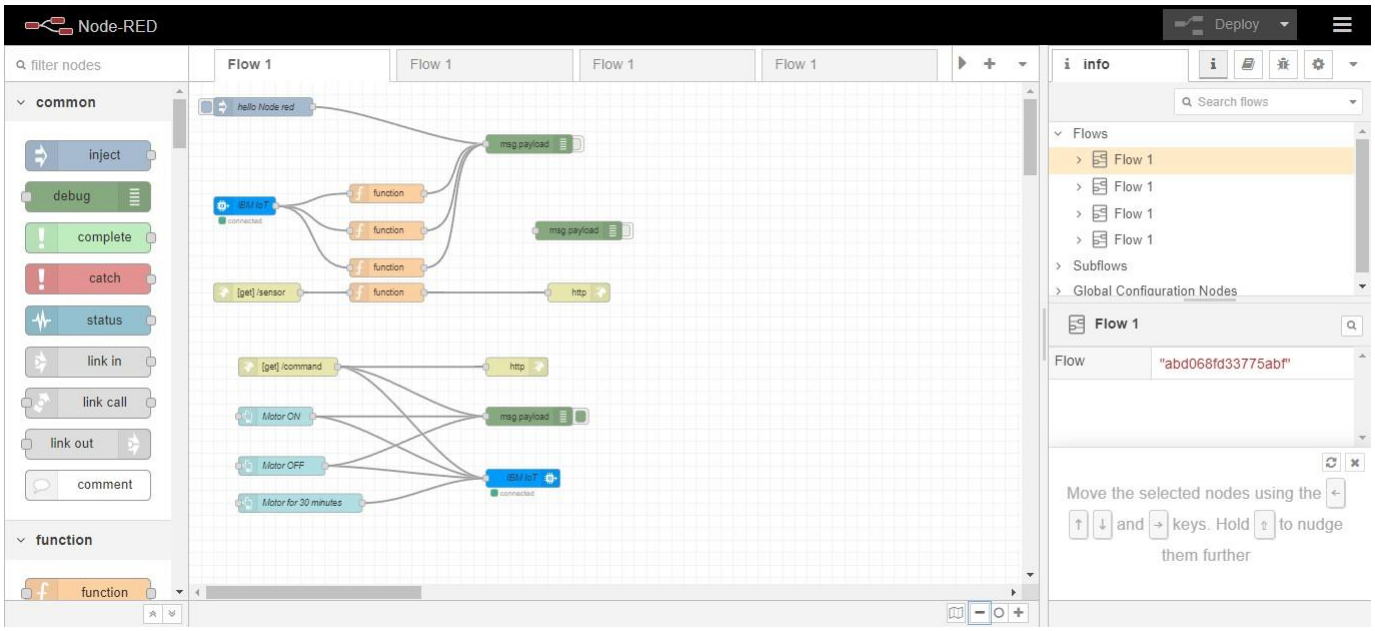
Python Code:

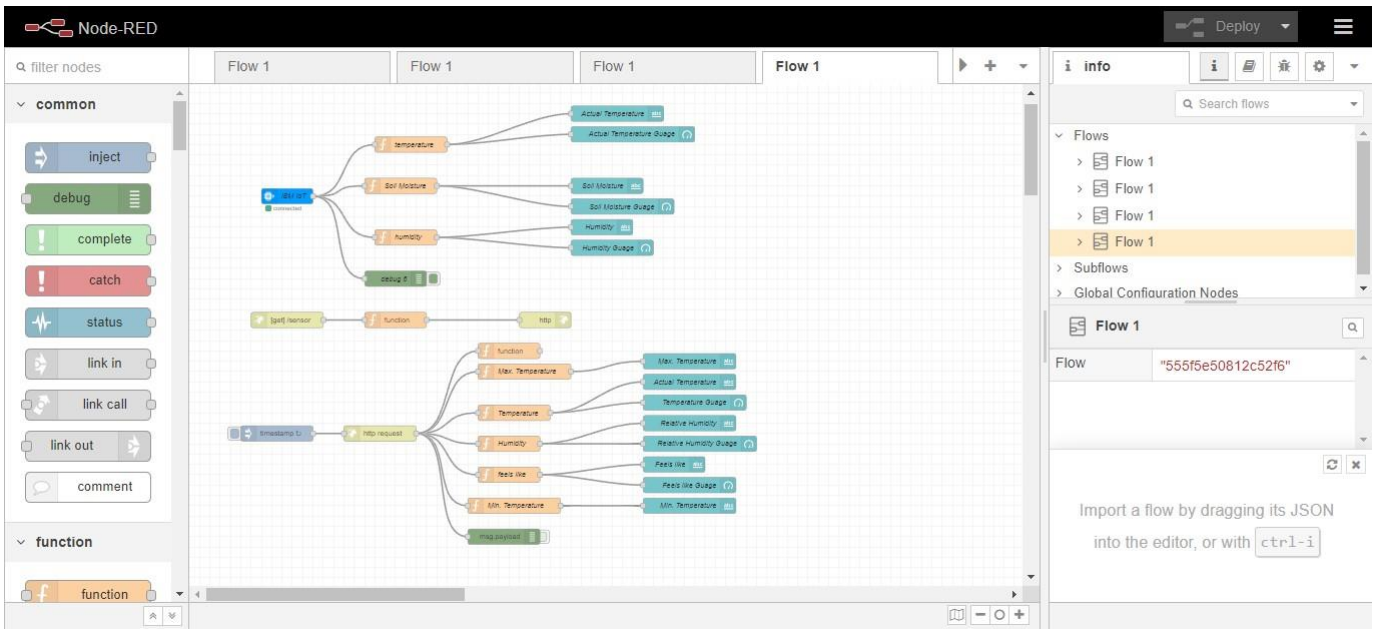
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

# Provide your IBM Watson Device Credentials
organization = "x0fxss"  # replace the ORG ID
deviceType = "smartfarmapplication"  # replace the Device type wi
deviceId = "98712345"  # replace Device ID
authMethod = "token"
authToken = "1234567890" # Replace the authtoken
# Initialize GPIO

# Receives Command from Node-red


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status = cmd.data['command']
    if status == "motoron":
        print("motor is on")
    elif status == "motoroff":
        print("motor is off")
```

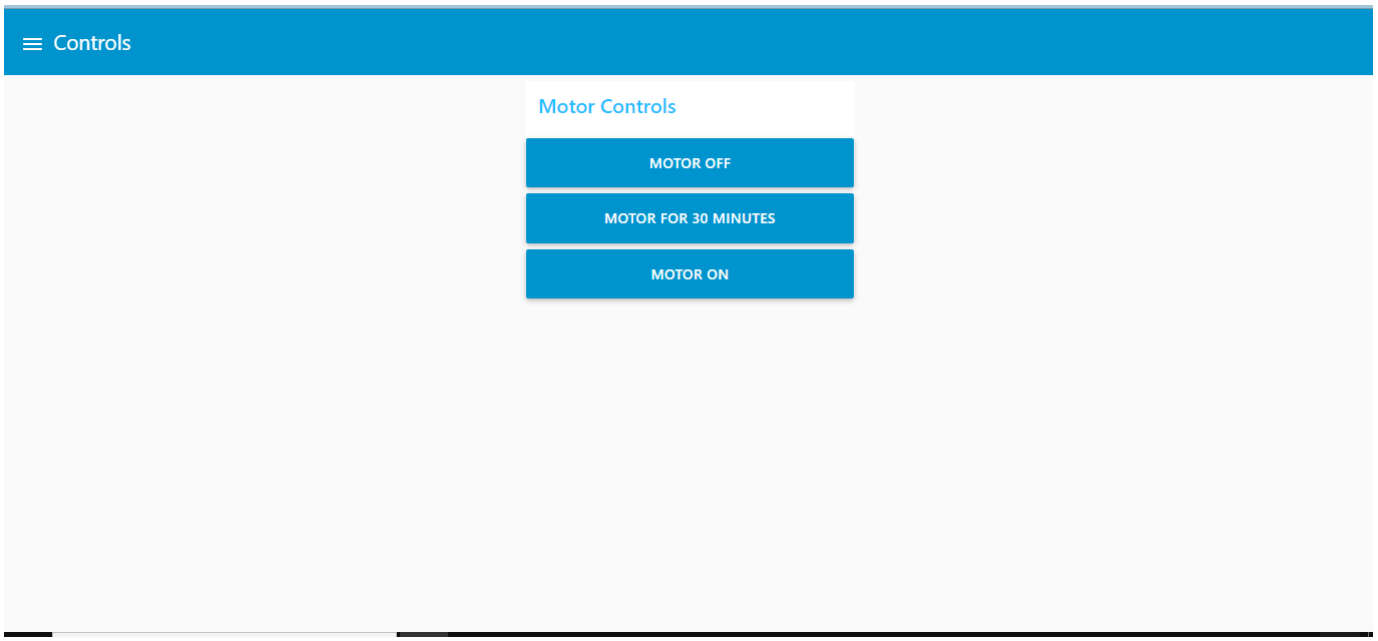
```
        elif status == "motorthirty":
            print("motor is on for 30 minutes")
            print("motor Started")
            for i in range(1,31):
                print("%d minutes to stop"%(30-i)) # use time.sleep(60) for delay of one minute in each iteration
            print("motor stopped")



    try:
        deviceOptions = {"org": organization, "type": deviceType,
                         "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
        deviceCli = ibmiotf.device.Client(deviceOptions)
        # .....................................................

    except Exception as e:
        print("Caught exception connecting device: %s" % str(e))
        sys.exit()

    '''Connect and send a datapoint like
    "{'temp:45, 'Humid':57, 'soilmoisture':76}"
    with value in the name of event "IoTSensor"'

    deviceCli.connect()

    while True:
        # Get Sensor Data from DHT11
        # Get Sensor Data from Soil Moisture Sensor
```

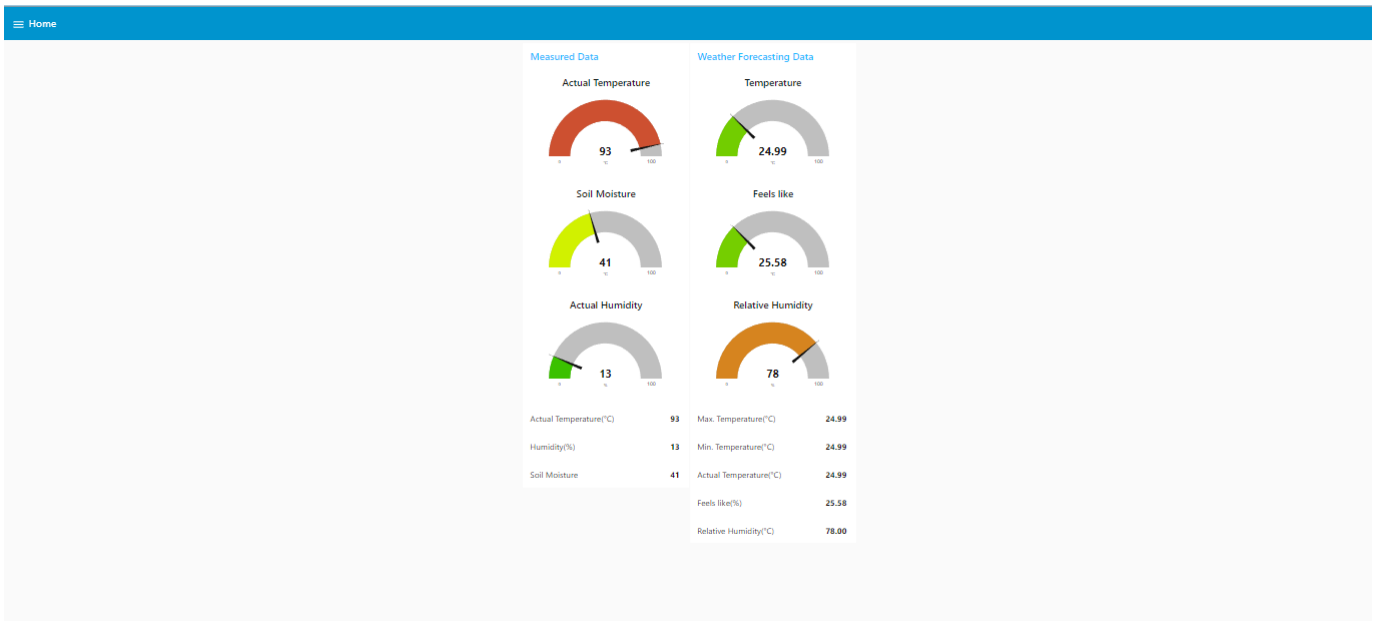
```
    temp = random.randint(0, 100)  # Generates random value
    Humid = random.randint(0, 100) # Generates random value
    soilmoisture = random.randint(0, 100)  # Generates random value

    data = {'temp': temp, 'Humid': Humid, 'soilmoisture': soilmoisture}
    # print data

    def myOnPublishCallback():
        print("Published Temperature = %s C" % temp, "Humidity = %s %%" %
              Humid, "soilmoisture = %s %%" % soilmoisture, "to IBM Watson")

    success = deviceCli.publishEvent(
        "IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoTF")
    time.sleep(5)  # sends a datapoint with delay of 5 seconds

    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

Node Red Connection:

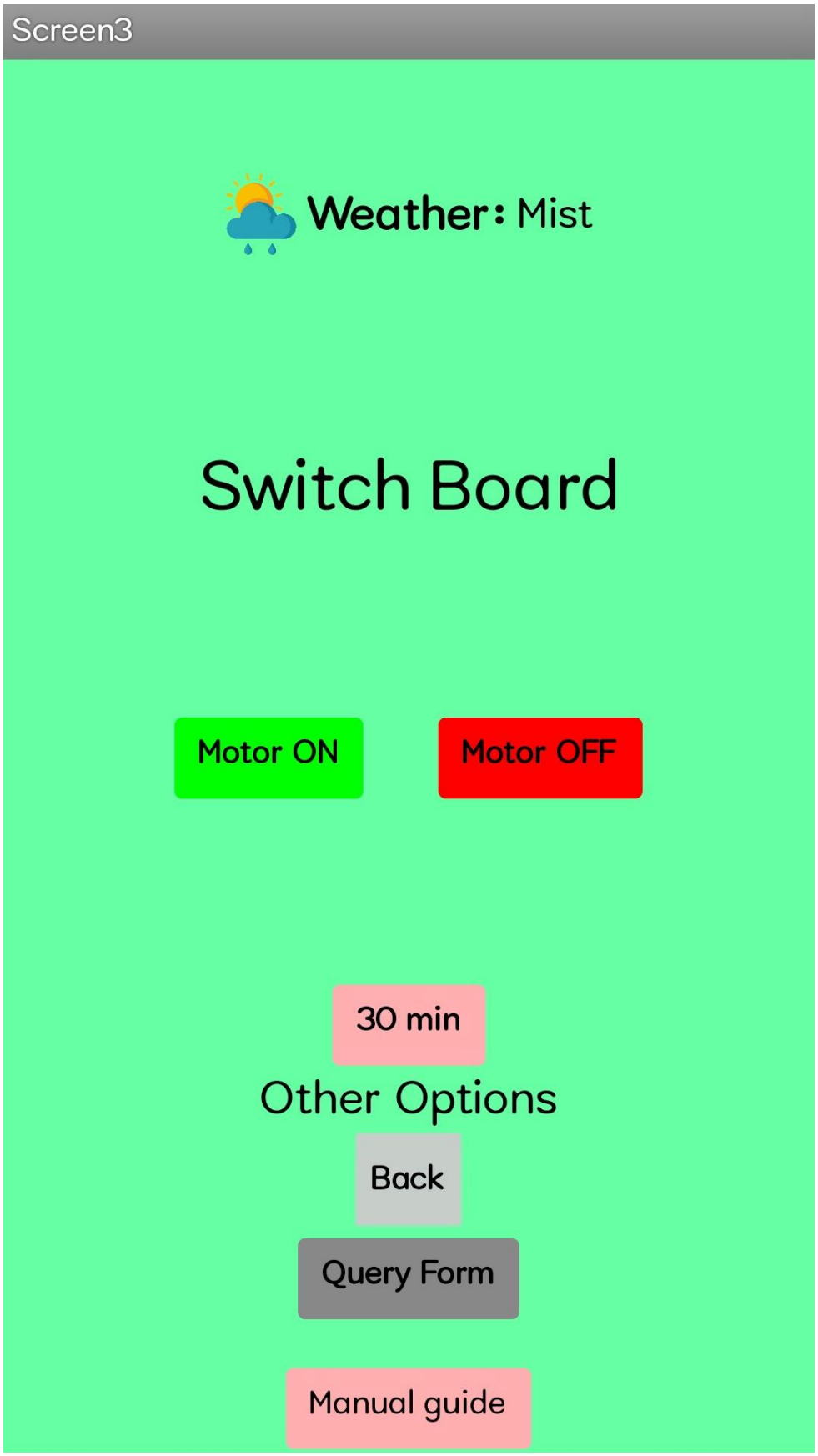





Controls
Motor Controls
MOTOR OFF
MOTOR FOR 30 MINUTES
MOTOR ON


Home
Measured Data
Actual Temperature
93
Soil Moisture
41
Actual Humidity
13
Actual Temperature(°C) 93
Humidity(%) 13
Soil Moisture 41
Weather Forecasting Data
Temperature
24.99
Feels like
25.58
Relative Humidity
78
Max. Temperature(°C) 24.99
Min. Temperature(°C) 24.99
Actual Temperature(°C) 24.99
Feels like(%) 25.58
Relative Humidity(°C) 78.00


Temperture
not found
Humidity
not found
soilmoisture
not found
Motor Control


Screen3
Weather: Mist
Switch Board
Motor ON
Motor OFF
30 min
Other Options
Back
Query Form
Manual guide

13 b Github link:- https://github.com/IBM-EPBL/IBM-Project-14686-1659588658

13 c Project Demo Link:-
https://drive.google.com/file/d/1G6gzYEwv9TADdLIBqQ1LolowYgAdS1FCgfyh/view?usp=drivesdk