SMART FARMER - IOT ENABLED SMART FARMING APPLICATION

A PROJECT REPORT

Submitted by

ANUSUYA.M

ABIGAIL DERALSHYA.S

BAVITHRA.S

KAVIPRIYA DEVI.V

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

PONESLY ENGINEERING COLLEGE, NAGERCOIL

ANNA UNIVERSITY:: CHENNAI 600025

NOV 2022

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"SMART FARMER –IOT ENABLED SMART FARMING APPLICATION"** is the bonafide work of "ANUSUYA.M (961819106301),ABIGAIL DERALSHYA.S **(961819106002), BAVITHRA.S(961819106010), KAVIPRIYADEVI.V (961819106032)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.M.R.Geetha	MRS.A.K.S.SARANYA M.E
HEAD OF THE DEPARTMENT	**MENTOR**
	Assistant Professor
Electronics and Communication Engineering	Electronics and Communication Engineering
Ponjesly College of Engineering, Nagercoil-3	Ponjesly College of Engineering, Nagercoil-3

CONTENTS

1. Introduction:

Digital technologies like the Internet of Things (IoT) are reshaping agriculture.When it comes to farming, what is IoT? The IoT connects “dumb” devices. IoT is all about data . Data is becoming a valuable resource for our world. Farmers may become more intelligent and safe by using data from gadgets to adapt to changing conditions more readily and farm more efficiently . To free up resources,farmers can use the ability to monitor agricultural conditions and infrastructure from afar . Many sectors and industries have adopted IoT to reduce errors and improve performance in manufacturing, energy,health care, and communication . Farm devices can collect and deliver data remotely to their owners using IoT.

Farmers can save time and money using IoT to keep tabs on-farm operations and efficiency, make more informed decisions about boosting productivity, and respond more quickly to changing conditions. In this case, it is putting data ahead of the farmer’s intuition . Atrough’s water supply, the amount of fertilizer to use on a crop, and which ewe to check when lambing are all things a farmer could know about.

Smart agriculture is necessary since 70% of the farming time is spent monitoring and analyzing crop status rather than performing actual field labor Given the industry’s size, it needs various technology and precise solutions to ensure sustainability while reducing environmental damage. Sensors and communication technologies have provided farmers with a remote sight of their fields, allowing them to watch what is happening without leaving home. Wireless sensors make monitoring crops in real-time with greater precision and, more importantly, detecting the early stages of undesirable conditions easier .This is why “smart agriculture uses innovative equipment and kits from seeding to crop harvesting, storage, and transportation. The operation is smart and cost-effective due to its accurate monitoring capabilities and prompts reporting using a variety of sensors. Various autonomous tractors, harvesters, robotic weeders, drones, and satellites supplement agriculture equipment . Sensors can be instantly deployed, started collecting data, and made available for further online study. By enabling precise data collection at each area, sensor technology allows crop and site-specific agriculture”. IoT and its apps are only scratching the surface of what they can do and have yet to impact people’s lives significantly, and everyone can see this. However, given the recent rise in IoT technology in agricultural applications, we can expect it to play a significant role.

1.1 Project Overview:

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

1.2 Purpose:

Smart Farming has a real potential to deliver a more productive and sustainable agricultural production, based on a more precise and resource-efficient approach.However,while in the USA possibly up to 80% of farmers use some kind of SFT, in Europe it is no more than 24%.
From the farmer's point of view, Smart Farming should provide the farmer with added value in the form of better decision making or more efficient exploitation operations and management. In this sense, smart farming is strongly related, to three interconnected technology fields addressed by Smart AKIS Network.

2. LITERATURE SURVEY:

Abstract:

An IoT-based farming system is referred to as smart agriculture.
A greater variety and higher quality of agricultural goods are produced under this new approach. IoT devices offer details about the characteristics of farming fields and then act in response to input from the farmer. An sophisticated IoT-based method for tracking the atmosphere and soil conditions for productive crop growth is given in this study. Using Node MCU and a number of sensors attached to it, the built system is capable of monitoring temperature, humidity, and soil moisture level. Additionally, an SMS message regarding the field's environmental state will be transmitted to the farmer's phone over Wi-Fi.

Introduction:

The objective of this outline is to present an IoT-based smart farming system that will give farmers access to real-time information about soil moisture and environmental temperature at a very low cost, allowing for real-time monitoring. Following the investigation, it was discovered that each crop field has unique qualities that can be assessed independently in terms of both quality and quantity.

Important elements that determine a soil's appropriateness and capacity for a particular crop include soil type, nutritional content, irrigation flow, pest resistance, etc. In light of conventional farming practices, farmers must frequently compute the agriculture plot during the crop life to have a better understanding of the crop circumstances.

As a result, smart agriculture is required because farmers spend 70% of their time monitoring and comprehending crop conditions rather than working in the fields. Wireless sensors make it easier to continuously monitor crops with greater accuracy and, most crucially, allow for the early detection of unfavourable states.

This is why modern agriculture uses sophisticated equipment throughout the process, from planting to crop harvesting to storage and transportation.

The process is made smarter and more cost-effective by timely reporting using sensors, which have accurate monitoring capabilities. Agriculture equipment is currently supplemented by a variety of autonomous tractors, harvesters, robotic weeders, drones, and satellites.

Sensors may be deployed and begin gathering data quickly. This data is then almost immediately available online for additional investigations. Sensor allows accurate data gathering for each site, which is essential for the application of scientific crop and site- specific agriculture.

EXISTING WORKS:

Monitoring the state of the climate.

The weather stations that incorporate numerous smart farming sensors are arguably the most well-liked smart agricultural technology. They are spread out around the area and gather various environmental data before sending it to the cloud. The measurements offered can be used to map the climate conditions, select the suitable crops, and implement the necessary improvements (i.e. precision farming)

AllMETEO, Smart Elements, and Pycno are a few instances of these agricultural IoT devices.

Automation in greenhouses.

In order to manage the greenhouse environment, farmers frequently require manual intervention. They can obtain precise real-time information on greenhouse parameters including illumination, temperature, soil quality, and humidity thanks to the usage of IoT sensors.

Weather stations can autonomously change the conditions to reflect the specified parameters in addition to sourcing environmental data. In particular, automation systems for greenhouses operate on a similar concept. Examples of IoT agriculture products that offer such features are Farmapp and Growlink.

Another intriguing device that makes use of smart agriculture sensors is GreenIQ. You can remotely control your irrigation and lighting systems with this intelligent sprinkler controller.

Crop administration

Crop management tools are an additional IoT product category in agriculture and a component of precision farming. They should be set up in the field to gather information pertaining to crop farming, such as temperature and precipitation as well as leaf water potential and general crop health, just as weather stations.

So you can successfully stop any diseases or pests that could reduce your crop's output, you can keep an eye on your crop's growth and any irregularities. Arable and Semios are excellent examples of how this use case might be put to use in practice.

Precision agriculture

Precision farming, also referred to as precision agriculture, is all about effectiveness and making precise data-driven decisions. It's also one of the most popular and successful IoT uses in agriculture.

Farmers may gather a wide range of metrics on every aspect of the field ecosystem and microclimate with IoT sensors, including illumination, temperature, soil quality, humidity, CO_2 levels, and pest infestations. With the aid of this information, farmers can more accurately predict the water, fertiliser, and pesticide requirements of their crops, cut costs, and produce better, healthier crops.

For instance, CropX creates Internet of Things (IoT) soil sensors that assess soil moisture, temperature, and electric conductivity, allowing farmers to tailor their practises to the particular requirements of each crop. When combined with GIS information, this technology aids in producing accurate soil maps for each fields.

Similar services are provided by Mothive, which assists farmers in reducing waste, increasing yields, and improving farm sustainability.

.

Drones used in agriculture

The use of agricultural drones in smart farming is arguably one of the most exciting developments in agritech. Drones, also referred to as unmanned aerial vehicles or UAVs, are more effective in gathering agricultural data than satellites and aircraft. Aside from surveillance, drones are also capable of carrying out a wide range of jobs that formerly needed human labour, such as planting crops, eradicating pests and diseases, spraying for agriculture, monitoring crops, etc.

For instance, DroneSeed creates drones to plant trees in sparsely wooded areas. Such drone use is six times more productive than using human labour. A Sense Fly eBee SQ agriculture drone, which is
reasonably priced, employs multispectral image analysis to gauge the health of crops.

Predictive analytics for intelligent agriculture

Predictive data analytics and precision agriculture go hand in hand. Despite the fact that IoT and smart sensor technology are a gold mine for extremely relevant real-time data, using data analytics enables farmers to make sense of it and make key forecasts, such as when to harvest crops, the likelihood of illnesses and pests, yield volume, etc. Farming, which is fundamentally very dependent on weather, is made more controlled and predictable with the aid of data analytics tools. For instance, the Crop Performance platform enables farmers to access crop volume and quality as well as their susceptibility to adverse weather circumstances like floods and drought in advance. Additionally, it enables farmers to choose the ideal amount of nutrients and water for each crop.

Solutions like SoilScout, when utilised in agriculture, help farmers save up to 50% on irrigation water, lessen the loss of nutrients due to overwatering, and provide actionable information regardless of the time of year or weather.

Total farm management programmes

The so-called agricultural productivity management systems might be seen as an example of a more advanced approach to IoT products in agriculture. They often comprise numerous on-site sensors and IoT devices for agriculture, as well as a robust dashboard with analytical tools and built-in accounting and reporting functions.

This enables remote farm monitoring and streamlines the majority of commercial processes. FarmLogs and Cropio both offer related solutions.
Other notable prospects include vehicle tracking (or even automation), storage management, logistics, etc., in addition to the IoT agriculture use cases that have been described.

Comprehensive farm management programmes

The so-called agricultural production management systems can be viewed as a more intricate method of utilising IoT devices in agriculture. A comprehensive dashboard with analytical capabilities and built-in accounting/reporting functions is typically included together with a variety of farm IoT devices and sensors that are deployed on the premises.

By streamlining the majority of corporate procedures, this provides remote farm monitoring capabilities. FarmLogs and Cropio are examples of comparable solutions. In addition to the IoT agriculture use cases mentioned, other significant prospects include logistics, storage management, vehicle tracking (or even automation), and so on.

Reference	architecture	Protocol	Technology	Benefits	Limits
Debauche (2022)	Short Supply Circuit Internet of Things (SSCIoT)	5G-MEC LoRa WSN Wi-Fi 3GPP	Edge, Fog &Cloud computing	Efficient data processing. ultra-low latency. limit exchange with cloud.	Complex architecture
Dahane (2021)	Edge-Fog-IoTCloud based architecture	nRF24L01 Http	Edge, fog,cloud computing and AI	Real time processing Interoperability and Accessibility The Edge Computing paradigm reduces congestion due to the need for computing	Complex architecture
Nguyen (2020)	5layer sensor edge fog cloud terminal system architecture	LoRa nRF Wifi AVR	Edge and fog computing LPWAN	Distributed local storage. Interoperability.security. The Edge Computing paradigm reduces congestion due to the need for computing	Complex architecture
				Reduce the size of data up 67%	
Perez-Pons (2020)	SmartDairyTracer platform based on Global Edge Computing Architecture (GECA)	ZigBee Wifi Http MQTT CoAP	Edge, Fog Computing AI Blockchain	The Edge Computing paradigm reduces congestion due to the need for computing. Network or cloud storage resources reduce data traffic by more than 46%. Blockchain technologies guarantee the security data integrity and traceability. Low-cost	Complex architecture

COMPONENTS:

Based on data from many sensors, including temperature, humidity, soil moisture and soil nutrients, this gadget monitors the farm or greenhouse and informs the farmer of the current circumstances in order for him to act quickly. The farmers' prompt actions will enable them to boost their farming production and make proper use of natural resources, which will also make our product environmentally friendly. By carefully monitoring the various current conditions, our product will improve the crops' quantity and quality. It is an Internet of Things device that uses the "Plug and Sense" idea. On a laptop or a smartphone, you can view real-time data for many metrics.

Different Components:

Breadboard

DHT11 Temperature and Humidity Sensor Soil

Moisture Sensor

LEDs

Passive buzzer

Power Supply-Power Bank
IBM Cloud

Watson IoT Platform

Node-Red

IBM CloudantDB

CONCLUSION:

The technology and materials we employed to produce our porotype allowed us to create an effective and accurate solution for farmers that was also affordable, as evidenced by our results and a literature review of other studies. which was affordable and simple for farmers to install. With the user-friendly app and various alarm mechanisms, we can therefore draw the conclusion that this prototype will undoubtedly assist farmers on tiny acreage to successfully monitor their crops.

2.1 Existing problem :

Farmers need to deal with many problems, including how to:

Cope with climate change, soil erosion and biodiversity loss.Satisfy consumers' changing tastes and expectations.Meet rising demand for more food of higher quality.Invest in farm productivity.Adopt and learn new technologies.Stay resilient against global economic factors. Inspire young people to stay in rural areas and become future farmers.

2.2 References :

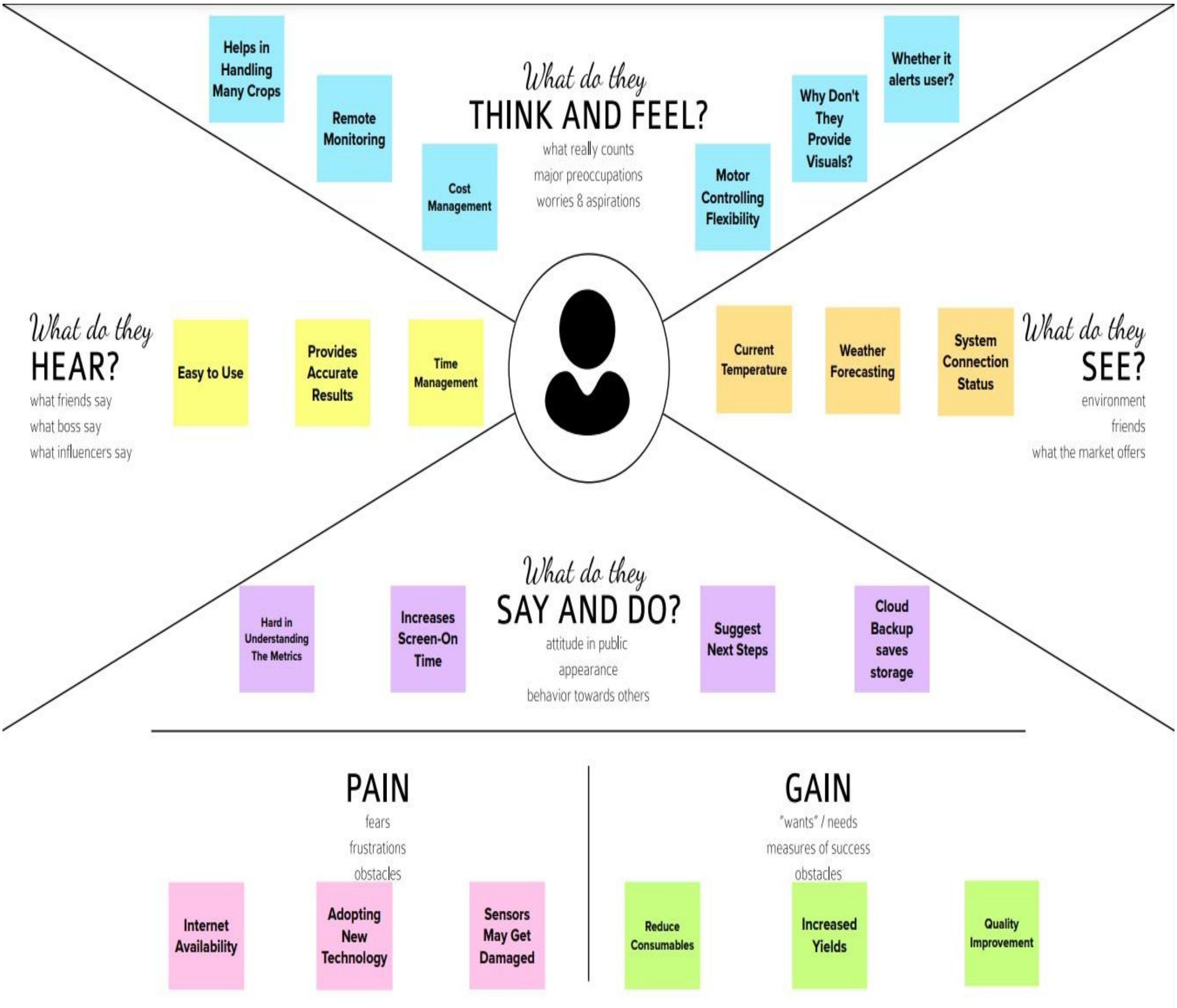
1. https//:www.researchgate.net
2. https//:www.wikipedia.org
3. https//:www.rapidonline.com
4. https//:www.schematics.com
5. https//:www.batteryuniversity.com

2.3 Problem Statement Definition :

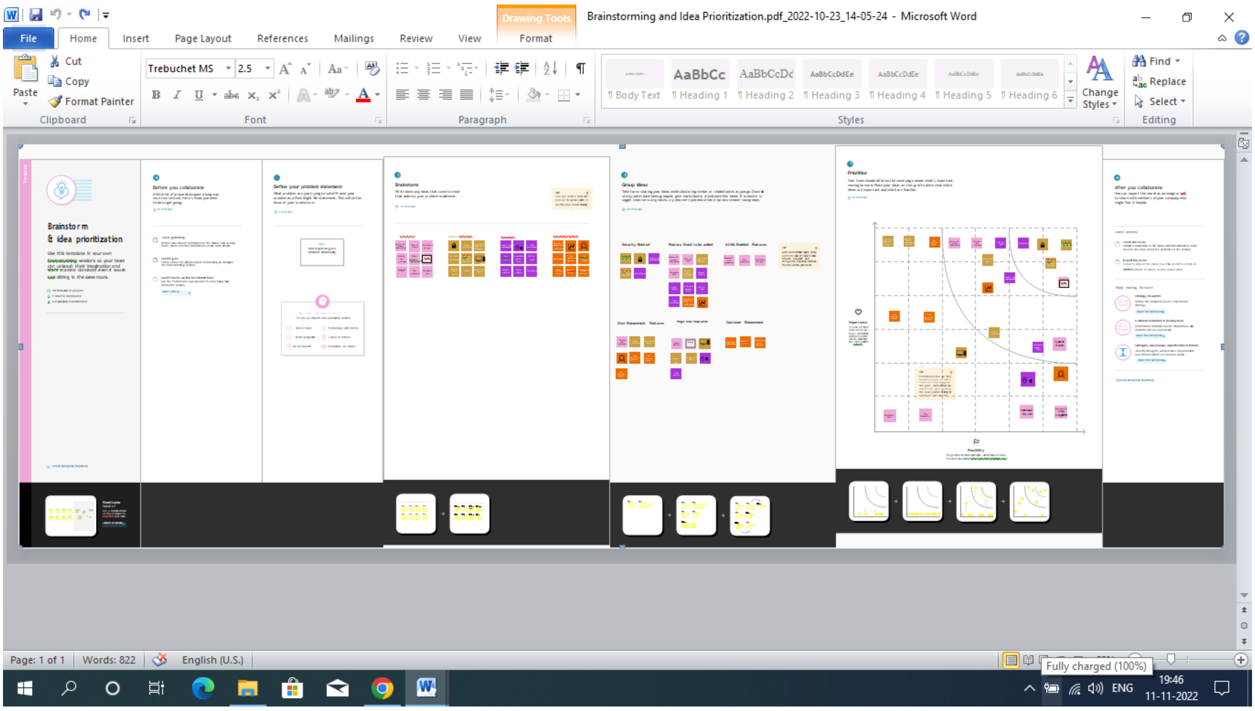
Farmers are to be present at farm for its maintenance irrespective of the weather conditions. They have to ensure that the crops are well watered and the farm status is monitored by them physically. Farmer have to stay most of the time in field in order to get a good yield. In difficult times like in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

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:



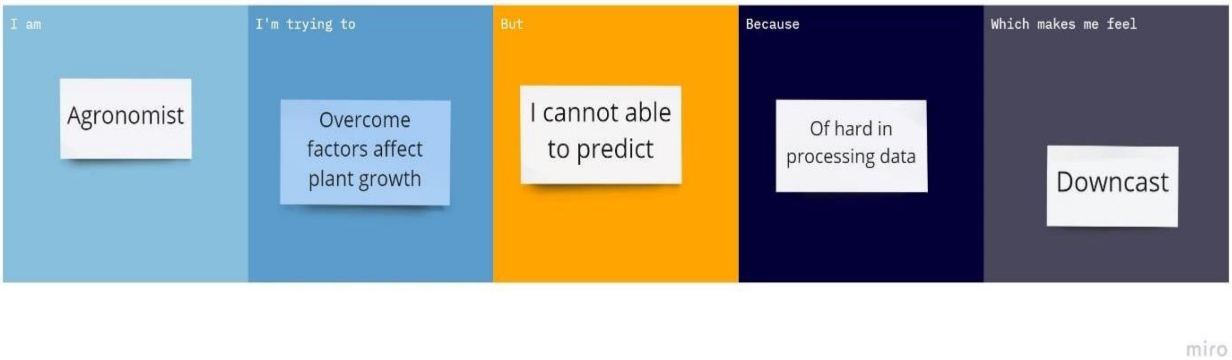
3.2 Ideation & Brainstorming:



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	It solves the monitoring problem by enabling remote monitoring feature
2.	Idea / Solution description	We are using cellular modems for connectivity along with other sensors to monitor soil,humidity and crops growth.
3.	Novelty / Uniqueness	We are building at lost cost.
4.	Social Impact / Customer Satisfaction	It unlocks thousands of farmers to monitor crops from anywhere/Customers are extremely satisfied with the product after using it for a while.
5.	Business Model (Revenue Model)	In the business model we are adding some ofthe subscription plans for revenue.
6.	Scalability of the Solution	It’s a product as a service model.scalable by connecting other devices to it.

3.4 Problem Solution fit



Problem stateme nt (PS)	I am (Custome	I'm trying	Bu	Becaus	Which makes me
PS-	Farm Expert	Improve agricult ure	I have difficulty in getting real time	Of low network coverag	Disappointe
PS-	Agronomist	Overcome factors affect	I cannot able to	Of Hard in processin g data	Downcas

4. REQUIREMENT ANALYSIS

4.1 Functional requirement:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Verifying Email	Verification Email is sent to respective Email IDs for verification.
FR-4	Authentication	Using of Biometrics or PIN Authentication to perform some sensitive actions on the app.
FR-5	Exporting Reports	Allowing users to export their yearly, monthly, weekly stats.
FR-6	Sharing of data to third-party application	Allowing users to share some data via whatsapp and other medium.

4.2 Non-Functional requirements:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Dashboard must be simple, clean and customizable.
NFR-2	**Security**	Using two factor authentication and maintaining session period for some actions.
NFR-3	**Reliability**	Ensure the code is well and good before making it to production.
NFR-4	**Performance**	Writing a efficient code to give better performance to the low end devices too.
NFR-5	**Availability**	Deploying the application in two or more Availability Zones to ensure the availability SLA of 99.999%
NFR-6	**Scalability**	Using auto scalable services in cloud for database, compute, etc….

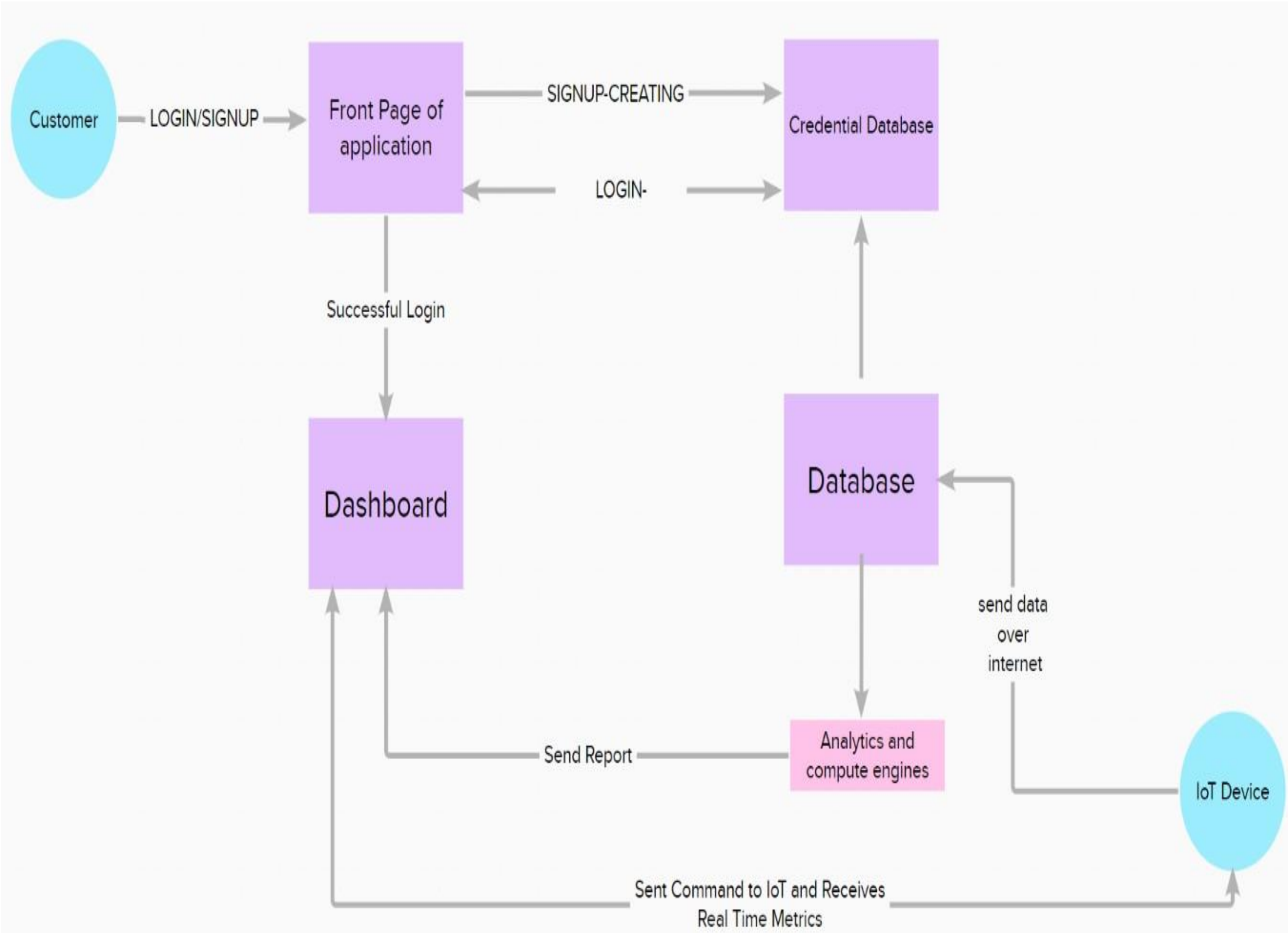
5. PROJECT DESIGN

Processes are something that are often overlooked in our industry, but are absolutely essential for a number of reasons.

They help you create a repeatable template for a winning formula.

They help your team understand how to move through a project in the correct way.

5.1 Data Flow Diagrams:



5.2Solution & Technical Architecture:

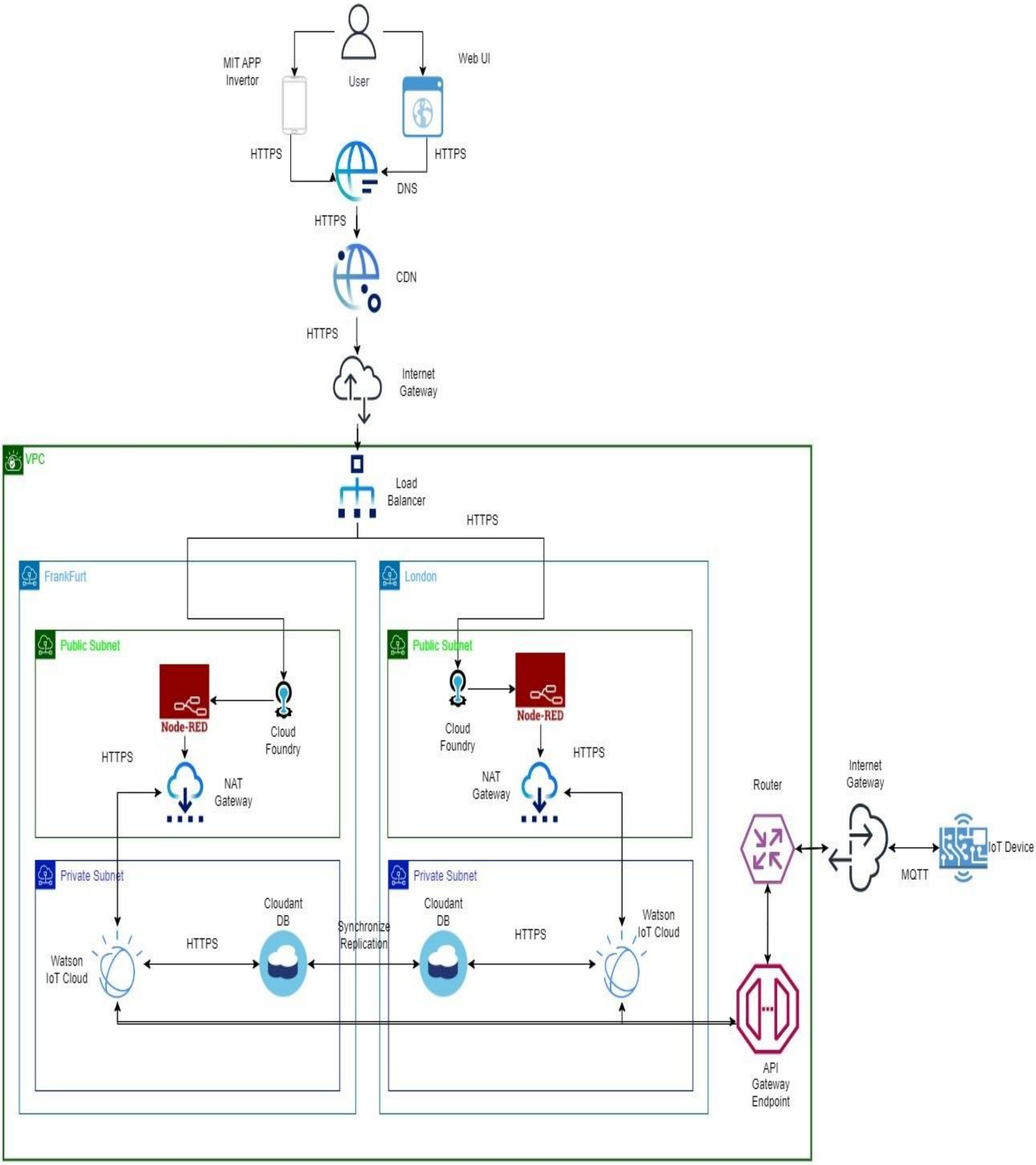


Table-1 : Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	How user interacts with application Web UI and Mobile App	NodeRed
2.	Application Logic-1	Logic for a process in the IoT Device to sense	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson Assistant
4.	Cloud Database	Database Service on Cloud	IBM Cloudant DB
5.	External API-1	Purpose of External API used in the application	IBM Weather API
6.	Infrastructure (Server / Cloud)	Application Deployment on Cloud Server Configuration	Cloud Foundry.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Python
2.	Security Implementations	List all the security / access controls implemented,use of firewalls etc.	Use API Gateway and Internet Gateway as firewall Protection
3.	Scalable Architecture	Every Cloud Services are hosted separately andmake is scalable separately	Public and Private Gateway
4.	Availability	Application is hosted on two regions for availability	London and Frankfurt Region Data Centres are used
5.	Performance	Used Content Delivery Network and API gatewayto scale millions of users and IoT Devices as well	IBM CDN and IBM API Gateway areused

5.3User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login
		USN-4	As a user, I can register for the application through Gmail	I can get registered in one tap
	Login	USN-5	As a user, I can log into the application by entering email & password	I can login to the application
	Dashboard	USN-6	As a user, I must watch my metrics displayed right in the front page of the application	I can may analysis from the dashboard itself
Customer (Web user)	Registration	USN-7	As a web user, I can fill the form right in the front page of the application	I can get registered without clicking and navigating to other page
Customer Care Executive	Dashboard	USN-8	As a customer care executive, I must provide with the text box to get the user information using unique no. for every individual user	I can get the user stats easily
	Chat Interface	USN-9	As a customer care executive, I must be indicated with the typing… indication	I can find whether the user on the other end is activeor not
Administrator	Dashboard	USN-10	As an admin, I must be provided with stats of the instance running behind	I can figure out the healthof the application
	Adding new employee	USN-11	As an admin, I must be provided with the option to add new employee	I can give the respective credential for the employee

6. PROJECT PLANNING & SCHEDULING

The definition of a sprint is a dedicated period of time in which a set amount of work will be completed on a project. It's part of the agile methodology, and an Agile project will be broken down into a number of sprints, each sprint taking the project closer to completion.

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	As a user, I can register for the application byentering my email, password, and confirming my password.	2	High	ANUSUYA
Sprint-1		USN-2	As a user, I will receive confirmation email onceI have registered for the application	1	High	ABIGAIL DERALSHYA
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	BAVITHRA
Sprint-1		USN-4	As a user, I can register for the applicationthrough Gmail	2	Medium	KAVIPRIYA DEVI
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	ANUSUYA
Sprint-2	Dashboard	USN-6	As a user, must watch my metrics displayedright in the front page of the application	2	High	KAVIPRIYA DEVI
Sprint-2		USN-8	As a customer care executive, I must providewith the text box to get the user information using unique no for every individual user.	1	Medium	ABIGAIL DERALSHYA
Sprint-2		USN-10	As an admin,I must be provide with status of theinstance running behind.	2	High	BAVITHRA

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Hardware	USN-11	Configuring Sensors	2	High	ANUSUYA, ABIGAIL DERALSHYA
Sprint-3		USN-12	Connecting Hardware to cloud	2	High	BAVITHRA, KAVIPRIYA DEVI
Sprint-4	MIT APP and Web UI	USN-13	Making a final APP and Web	1	Medium	ANUSUYA,KAVIPRIYADEVI
Sprint-4	Testing and Bug Fixing	USN-14	Testing Performance and Fixing Bugs	1	Low	**ABIGAIL DERALSHYA, BAVITHRA**

6.2 Sprint Delivery Schedule:

Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	1	27Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	2	03 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	1	10 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	2	17 Nov 2022

Velocity:

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

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

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

https://www.visual-paradigm.com/scrum/scrum-burndown-chart/
https://www.atlassian.com/agile/tutorials/burndown-charts

Reference:

https://www.atlassian.com/agile/project-management
https://www.atlassian.com/agile/tutorials/how-to-do-scrum-with-jira-software
https://www.atlassian.com/agile/tutorials/epics
https://www.atlassian.com/agile/tutorials/sprints
https://www.atlassian.com/agile/project-management/estimation
https://www.atlassian.com/agile/tutorials/burndown-charts

6.3 Reports from JIRA:

7. CODING & SOLUTIONING

(Explain the features added in the project along with code):

7.1 Feature 1 (coding and result):

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

#Provide your IBM Watson Device Credentials
organization = "b84wgs"
deviceType = "abi"
deviceId = "12345678"
authMethod = "token"
authToken = "87654321"

# Initialize GPIO
def myCommandCallback(cmd):
   print("Command received: %s" % cmd.data['command'])
   status=cmd.data['command']
   if status=="motoron":
      print ("Motor is ON")
   else :
      print ("Motor is OFF")

 #print(cmd)

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

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

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()

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

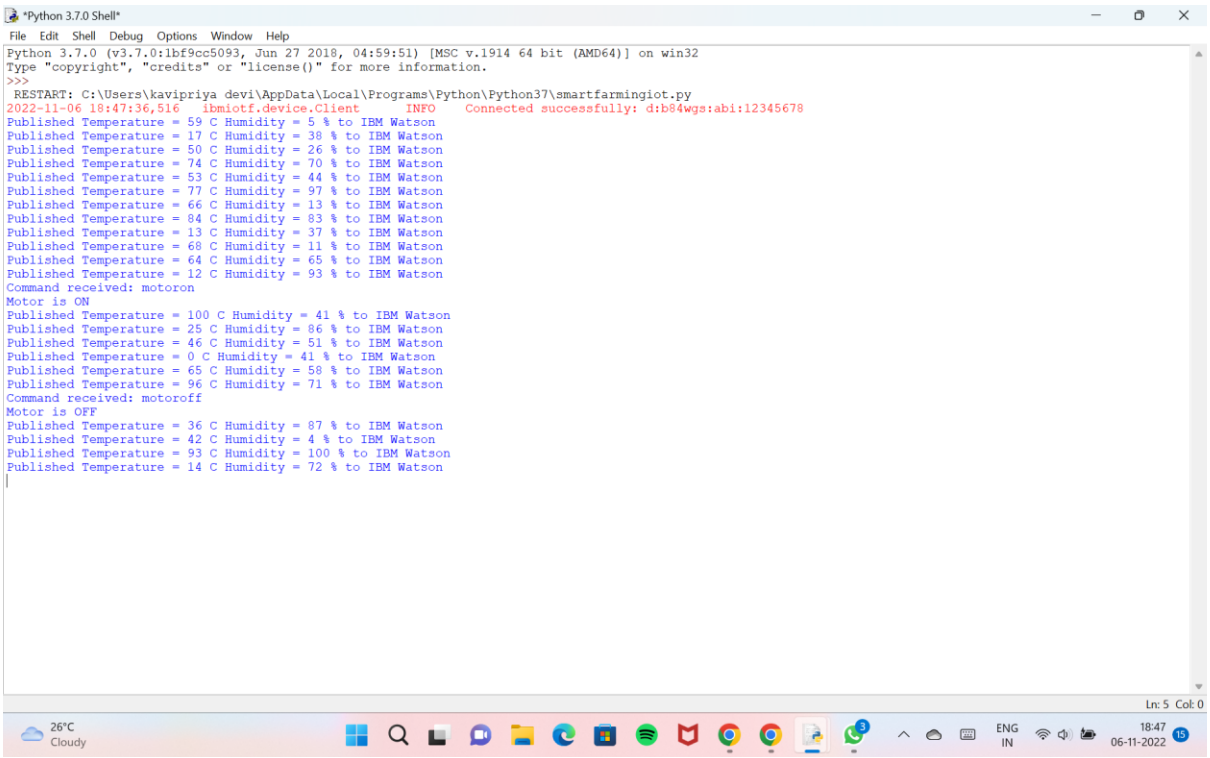
```
    Temp=random.randint(0,100)
    Humid=random.randint(0,100)

    data = { 'Temp' : Temp, 'Humid': Humid }
    #print data
    def myOnPublishCallback():
      print ("Published Temperature = %s C" % Temp, "Humidity = %s %%" % Humid, "to IBM Watson")

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
      print("Not connected to IoTF")
    time.sleep(1)

            deviceCli.commandCallback = myCommandCallback
```

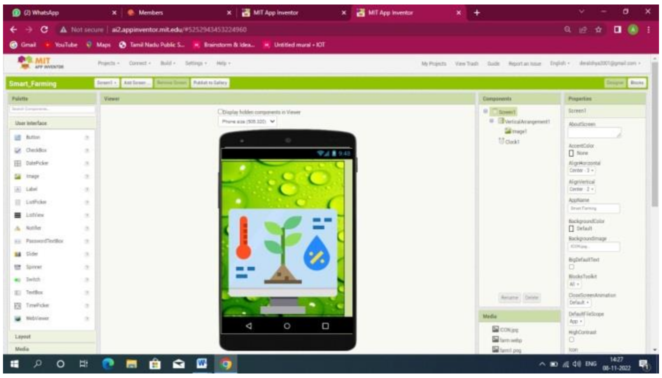
output:

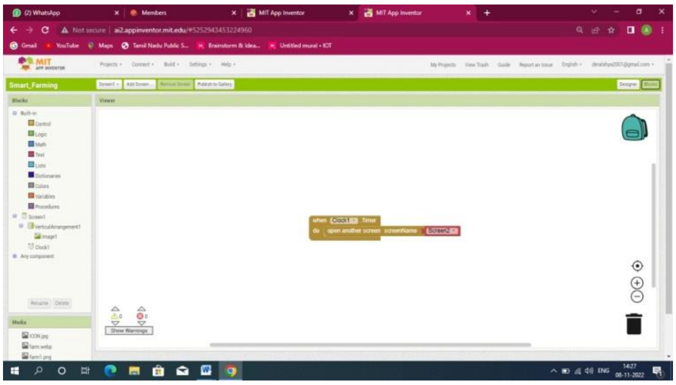


7.2 Feature 2: (MITAPP INVENTER)

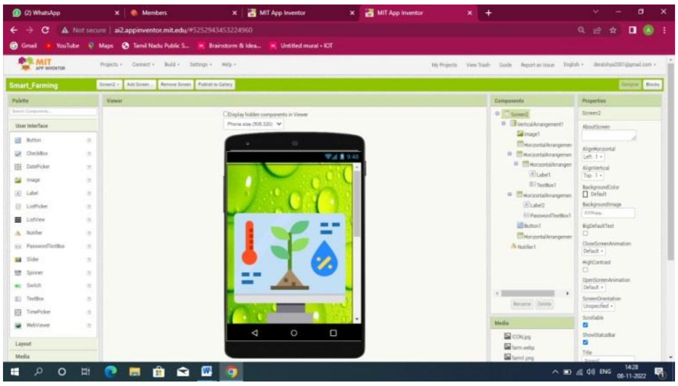
MIT APP INVENTOR: ICON PAGE:

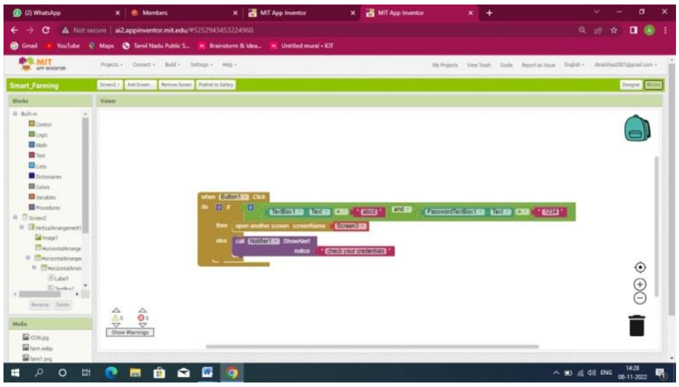
For Screen 1:



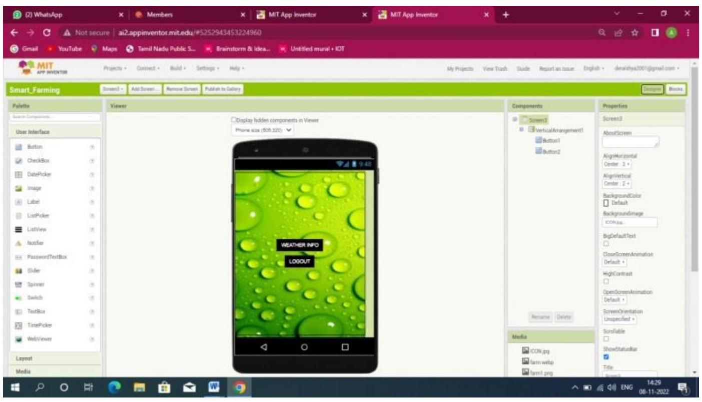


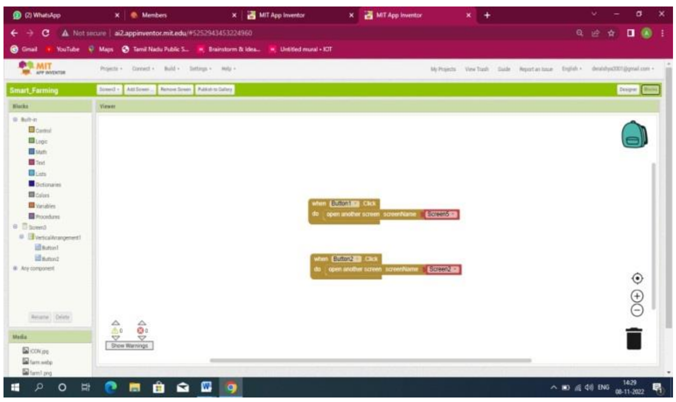
For Screen 2:



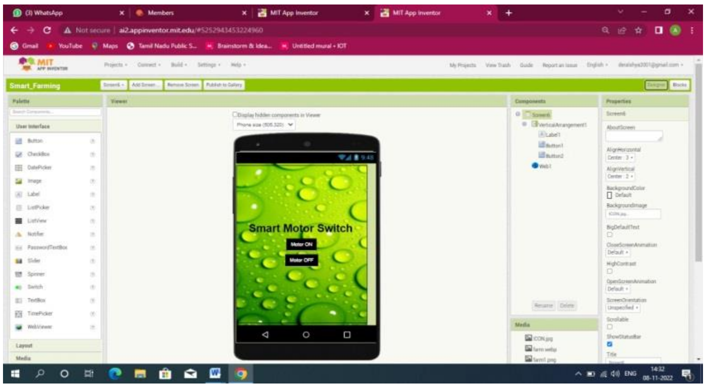


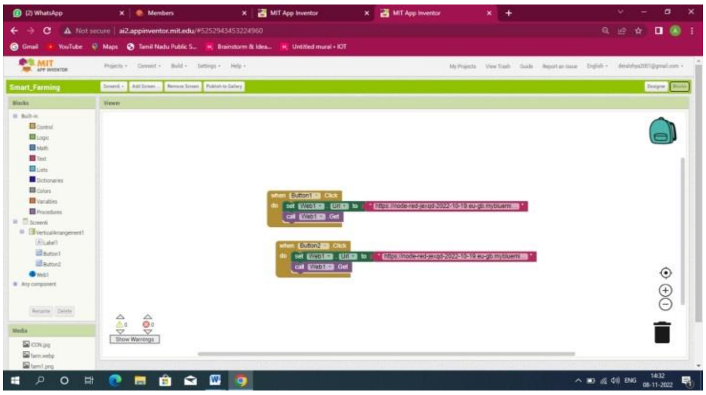
For Screen 3:



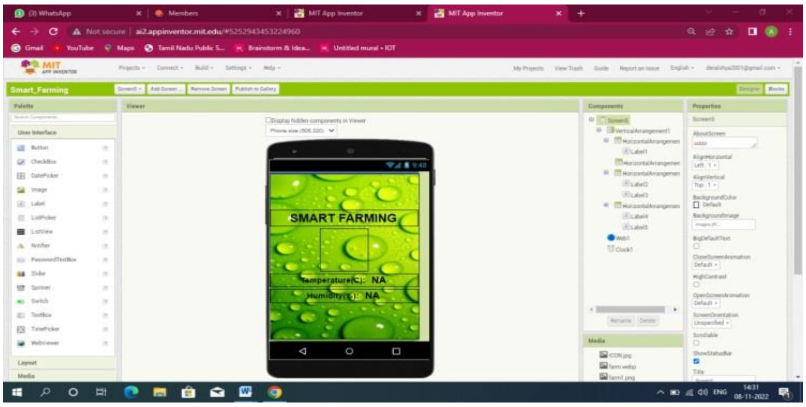


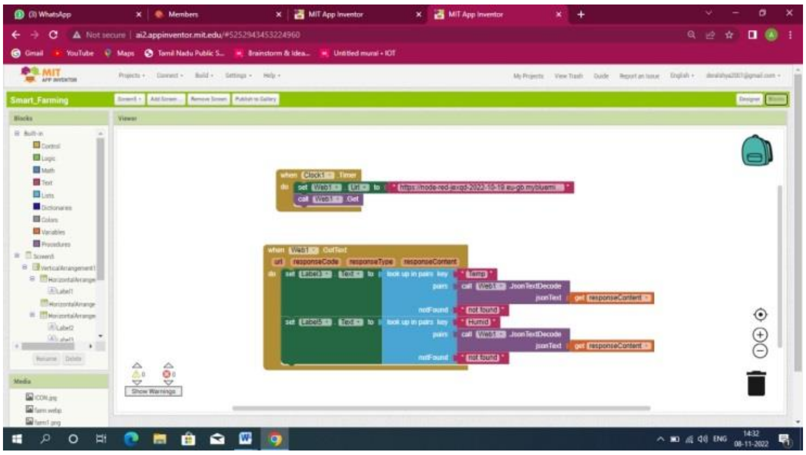
For Screen 4:





For Screen 5:





8. TESTING:

Test cases help guide the tester through a sequence of steps to validate whether a software application is free of bugs, and working as required by the end-user. Learning how to write test cases for software requires basic writing skills, an attention to detail, and a good understanding of the application under test (AUT).

8.1Test Cases:

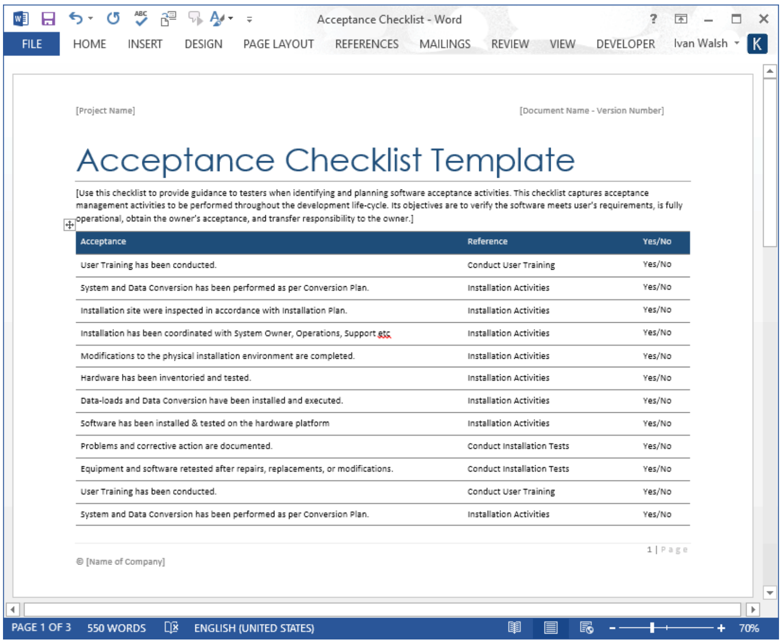
Step	Procedures	Expected Result	Result
1	Insert admin, username, and password	Save the insert data into database	Success
2	Insert correct username, password for login	Verify the admin	Success
3	Click 'Register,' 'Login' button	Application redirect admin to Login page after register and Main page after login	Success
4	Repeat step 2 and 3 for login using false username, password	Application display error message	Success
5	Update Admin Account	New update data saved into database	Success
6	Log Out Account	Log out redirected to Login page	Success
Precondition		No credentials are currently login	
Post-condition		New and updated Admin name, username, and password saved in database	

Based on Table 1, only authenticated users are allowed access to the application.

S.No	Action	Inputs	Expected Output	Actual Output	Test Browser	Test Result	Test Comments
1	Launch application	https://www.facebook.com/	Facebook home	Facebook home	IE-11	Pass	[Priya 10/17/2017 11:44 AM]: Launch successful
2	Enter invalid Email & any Password and hit login button	Email id : invalid@xyz.com Password: ******	The email address or phone number that you've entered doesn't match any account. Sign up for an account.	The email address or phone number that you've entered doesn't match any account. Sign up for an account.	IE-11	Pass	[Priya 10/17/2017 11:45 AM]: Invalid login attempt stopped
3	Enter valid Email & incorrect Password and hit login button	Email id : valid@xyz.com Password: ******	The password that you've entered is incorrect. Forgotten password?	The password that you've entered is incorrect. Forgotten password?	IE-11	Pass	[Priya 10/17/2017 11:46 AM]: Invalid login attempt stopped

8.2User Acceptance Testing:

UAT consists, in practice, of people from the target audience using the application. The defects they find are then reported and fixed. This scenario is what most closely resembles “the real world.” The process allows users to “get their hands dirty” with the application. They can see if things work as intended.



Acceptance Checklist - Word

FILE HOME INSERT DESIGN PAGE LAYOUT REFERENCES MAILINGS REVIEW VIEW DEVELOPER Ivan Walsh K

[Project Name] [Document Name - Version Number]

Acceptance Checklist Template

[Use this checklist to provide guidance to testers when identifying and planning software acceptance activities. This checklist captures acceptance management activities to be performed throughout the development life-cycle. Its objectives are to verify the software meets user's requirements, is fully operational, obtain the owner's acceptance, and transfer responsibility to the owner.]

Acceptance	Reference	Yes/No
User Training has been conducted.	Conduct User Training	Yes/No
System and Data Conversion has been performed as per Conversion Plan.	Installation Activities	Yes/No
Installation site were inspected in accordance with Installation Plan.	Installation Activities	Yes/No
Installation has been coordinated with System Owner, Operations, Support etc	Installation Activities	Yes/No
Modifications to the physical installation environment are completed.	Installation Activities	Yes/No
Hardware has been inventoried and tested.	Installation Activities	Yes/No
Data-loads and Data Conversion have been installed and executed.	Installation Activities	Yes/No
Software has been installed & tested on the hardware platform	Installation Activities	Yes/No
Problems and corrective action are documented.	Conduct Installation Tests	Yes/No
Equipment and software retested after repairs, replacements, or modifications.	Conduct Installation Tests	Yes/No
User Training has been conducted.	Conduct User Training	Yes/No
System and Data Conversion has been performed as per Conversion Plan.	Installation Activities	Yes/No

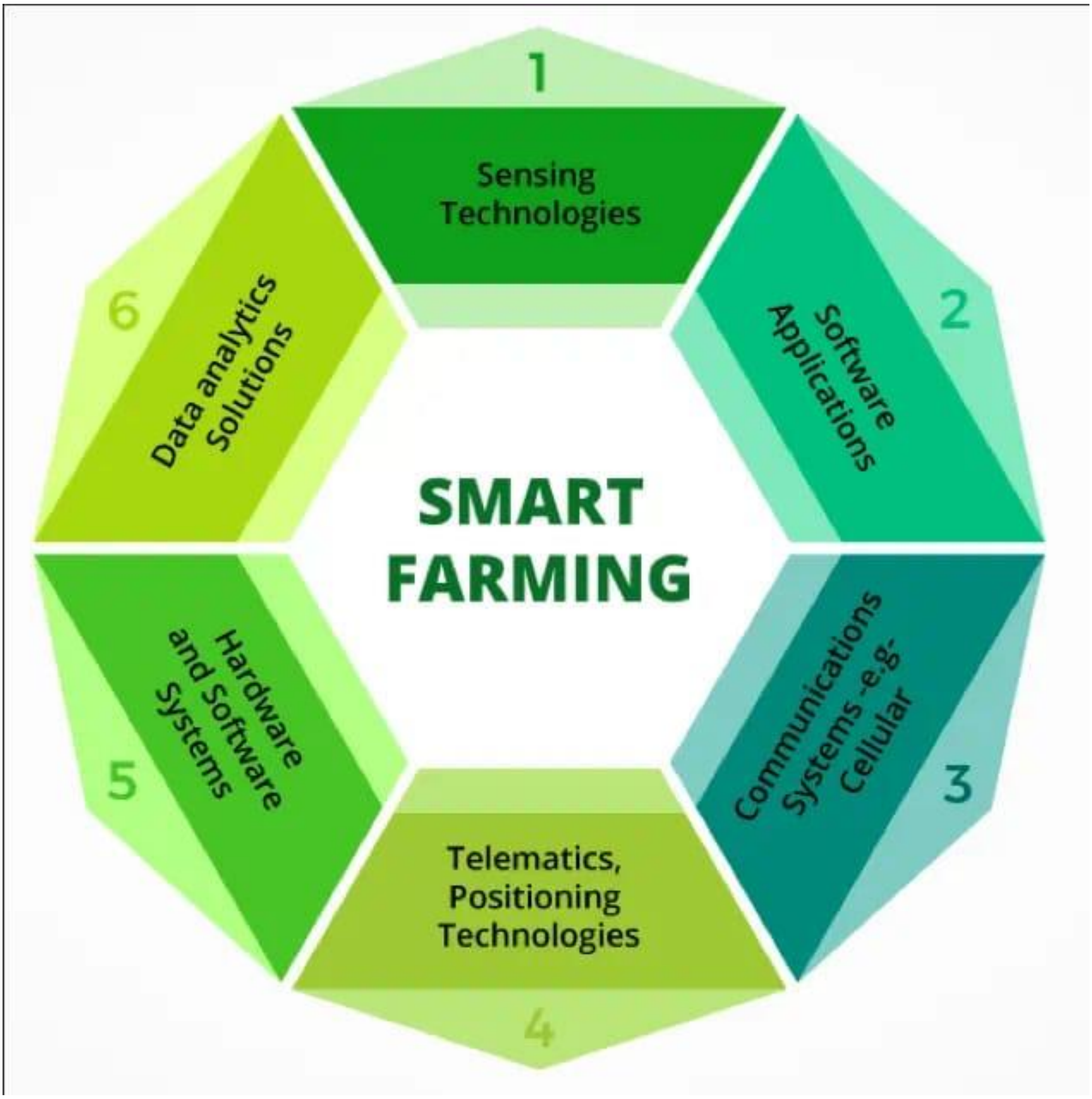
1 | Page

© [Name of Company]

PAGE 1 OF 3 550 WORDS ENGLISH (UNITED STATES) 70%

9. RESULTS:

9.1 Performance Metrics:



10. Advantages & Disadvantages:

Advantages:

- Farms can be monitored and controlled remotely.
- Increase in convenience to farmers.
- Less labour cost.
- Better standards of living.

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 has been implemented successfully.

12. FUTURE SCOPE:

From a business perspective, farmers are seeking ways to improve profitability and efficiency by on the one hand looking for ways to reduce their costs and on the other hand obtaining better prices for their product. Therefore they need to take better, more optimal decisions and improve management control. While in the past advisory services were based on general knowledge that once was derived from research experiments, there is an increasing need for information and knowledge that is generated on-farm in its local-specific context. It is expected that Big Data technologies help to achieve these goals in a better way

13.APPENDIX:

Source Code:

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


#Provide your IBM Watson Device Credentials
organization = "b84wgs"
deviceType = "abi"
deviceId = "12345678"
authMethod = "token"
authToken = "87654321"


# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("Motor is ON")
  else :
        print ("Motor is OFF")
    #print(cmd)
 try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": 
authMethod, "auth-token": authToken}
deviceCli = ibmiotf.device.Client(deviceOptions)
 #..............................................

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

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type 
"greeting" 10 times
deviceCli.connect()
```

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

        Temp=random.randint(0,100)
        Humid=random.randint(0,100)

        data = { 'Temp' : Temp, 'Humid': Humid }
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % Temp, "Humidity = %s %%" % Humid,
"to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)

        deviceCli.commandCallback = myCommandCallback
```

Github Link :

https://github.com/IBM-EPBL/IBM-Project-47881-1660803035/tree/main

Demo Link:

https://youtu.be/Hx0Fx_fmBrE