SmartFarmer - IoT Enabled Smart Farming Application

A PROJECT REPORT

Submitted by

CHANDRAKEERTHI R	**19ECL001**
SANTHOSH KUMAR P	**19ECR115**
SATHISHKUMAR E	**19ECR122**
VELMURUGAN D	**19ECR149**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

VELALAR COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Chennai)

ERODE-638012

NOVEMBER 2022

Table of Contents

CHAPTER 1

INTRODUCTION

1.1 Overview

This is a Smart Agriculture System project based on Internet Of Things (IoT), that can measure soil moisture, Humidity and temperature conditions for agriculture using Watson IoT services. IoT is network that connects physical objects or things embedded with electronics, software and sensors through network connectivity that collects and transfers data using cloud for communication. Data is transferred through internet without human to human or human to computer interaction.

In this project we have not used any hardware. Instead of real soil moisture, Humidity and Temperature data obtained from sensors we make use of IBM IoT Simulator which can transmit these parameters as required.

- **Project Requirements** : Node-RED, IBM Cloud, IBM Watson IoT, Node.js, IBM Device, Python 3.9.6, Open Weather API platform.
- **Project Deliverables** : Application for IoT based Smart Agriculture System (Software example)

1.2 Purpose

IoT based farming is grooming nowadays because it improves the entire agriculture system by monitoring the field in real-time. With the help of IoT in agriculture not only saves the time but also reduces the extravagant use of resources such as water and electricity.

Sometimes due to over or less supply of water in the agricultural field crops may not grow proper. Using IoT supply of water and growth of plants can be satisfied to a greater extent. The flow of water can be controlled from the application. Thus this approach towards Agriculture will help the farmers to get better yield at low cost and without much usage of resources.

1.2.1 Scope of Work

- Create an IBM account.
- Create a device in IBM Cloud Account.
- Install Node-RED and download all the required nodes and configure the nodes.
- Create the open weather map account and get the API key and the weather conditions using API key in the Node-RED.
- Create a web application for user interaction for observation and control actions.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing Problem

- Agriculture is a field which forms the basis of our economy. Yet it faces a lot of problems in terms of availability of resources, Irrigation, increasing rate of Pesticides, Climatic disasters, Insects which ruin the crops and makes a huge loss this sector.
- In agriculture water is needed for the crops for their growth. If the Soil gets dry it is necessary to supply water. But sometime if the farmer doesn't visit the field it is not possible to know the condition of soil.
- Sometimes over supply of water or less supply of water affects the growth of crops.
- Sometimes if the weather/temperature changes suddenly it is necessary to take certain actions.
- Specific crops grow better in specific conditions, they may get damaged due to bad weather.

2.1 Proposed Solution

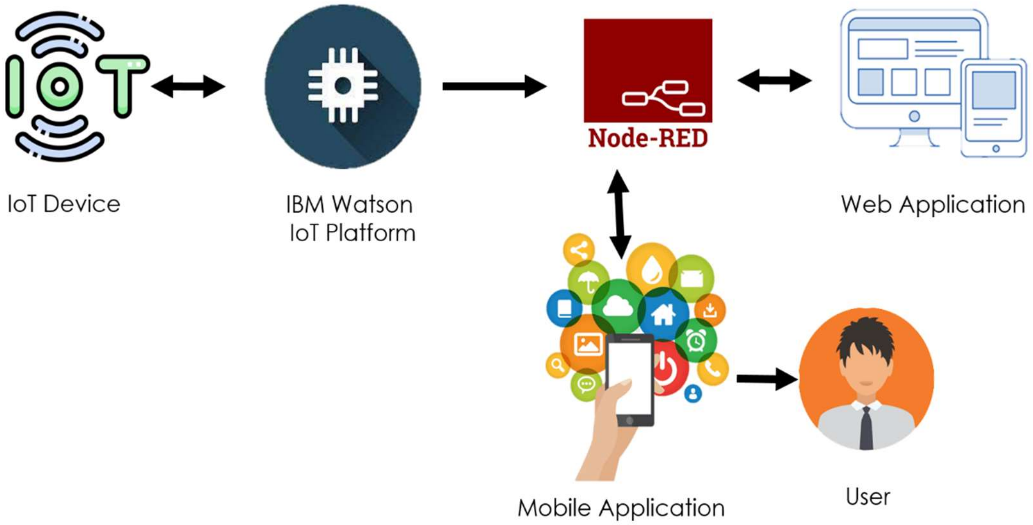
- To develop a Smart Agricultural System based on IOT which can give real time data and can help farmers in a very efficient manner.
- Soil Moisture can be checked by using the sensors that can sense the soil condition and send the data (moisture content in the soil) over the cloud services to the web application.

- The supply of water can be controlled from anywhere by controlling the motor state (ON/OFF), using web application.
- Surrounding temperature can also be sensed by the sensors and displayed on the application.
- Real time weather conditions can also be known by using different weather API's from different websites and displayed on our Application.

CHAPTER 3

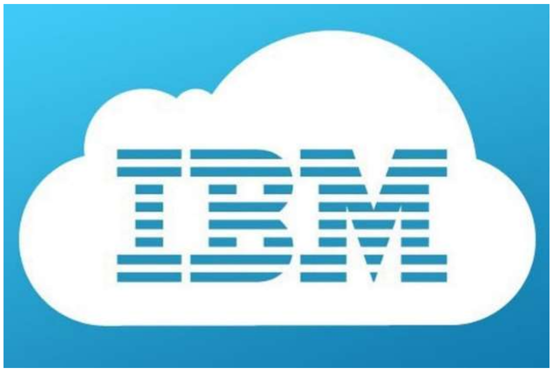
THEORETICAL ANALYSIS

3.1 Block Diagram



3.2Hardware / Software Designing

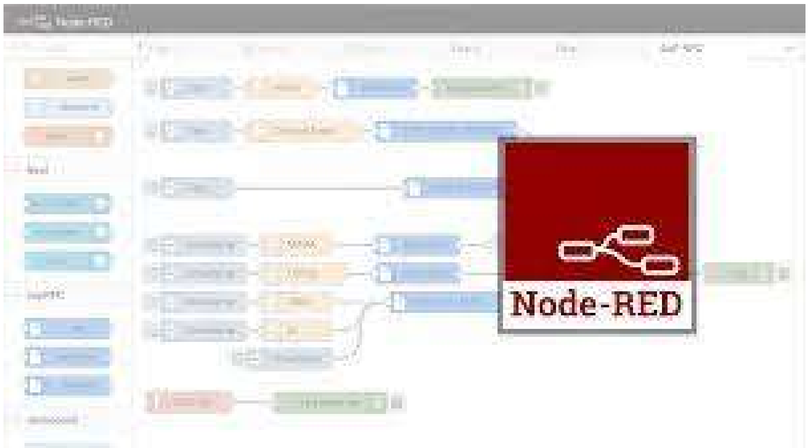
3.2.1 IBM Cloud



The IBM Cloud platform combines platform as a service (PaaS) with infrastructure as a service (IaaS) to provide an integrated experience. The platform scales and supports both small development teams and organizations, and large enterprise businesses. Globally deployed across data centers around the world, the solution you build on IBM Cloud® spins up fast and performs reliably in a tested and supported environment you can trust!

IBM Cloud provides solutions that enable higher levels of compliance, security, and management, with proven architecture patterns and methods for rapid delivery for running mission-critical workloads. Available in data centers worldwide, with multizone regions in North and South America, Europe, Asia, and Australia, you are enabled to deploy locally with global scalability.

3.2.2 Node-RED



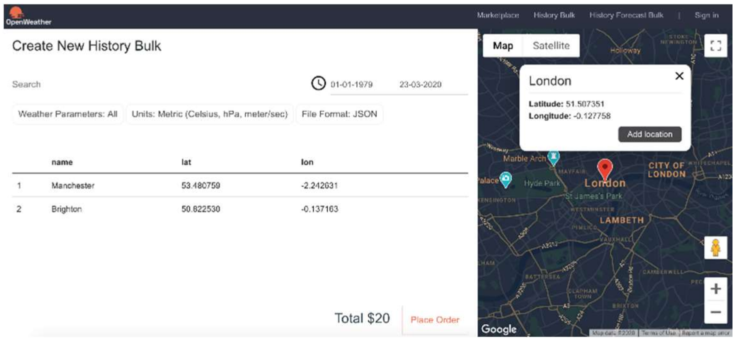
Node-RED is a flow-based programming tool, originally developed by IBM's Emerging Technology Services team and now a part of the OpenJS Foundation.

Invented by J. Paul Morrison in the 1970s, flow-based programming is a way of describing an application's behaviour as a network of black-boxes, or "nodes" as they are called in Node-RED. Each node has a well-defined purpose; it is given some data, it does something with that data and then it passes that data on. The network is responsible for the flow of data between the nodes.

It is a model that lends itself very well to a visual representation and makes it more accessible to a wider range of users. If someone can break down a problem into discrete steps they can look at a flow and get a sense of what it is doing; without having to understand the individual lines of code within each node.

Node-RED consists of a Node.js based runtime that you point a web browser at to access the flow editor. Within the browser you create your application by dragging nodes from your palette into a workspace and start to wire them together. With a single click, the application is deployed back to the runtime where it is run. The palette of nodes can be easily extended by installing new nodes created by the community and the flows you create can be easily shared as JSON files.

3.2.3 OpenWeatherMap



Create New History Bulk

Search 01-01-1979 23-03-2020

Weather Parameters: All | Units: Metric (Celsius, hPa, meter/sec) | File Format: JSON

	name	lat	lon
1	Manchester	53.480759	-2.242631
2	Brighton	50.822530	-0.137163

Total $20

London

Latitude: 51.507351
Longitude: -0.127758

OpenWeatherMap is an online service, owned by OpenWeather Ltd, that provides global weather data via API, including current weather data, forecasts, nowcasts and historical weather data for any geographical location. The company provides a minute-by-minute hyperlocal precipitation forecast for any location. The convolutional machine learning model is used to utilise meteorological broadcast services and data from airport weather stations, on-ground radar stations, weather satellites, remote sensing satellites, METAR and automated weather stations.

The variety of weather APIs provided by OpenWeatherMap have found a significant popularity among the software developers, which resulted in the growing multitude of repositories on GitHub. The APIs support multiple languages, units of measurement and industry standard data formats like JSON and XML.

3.2.4 Python

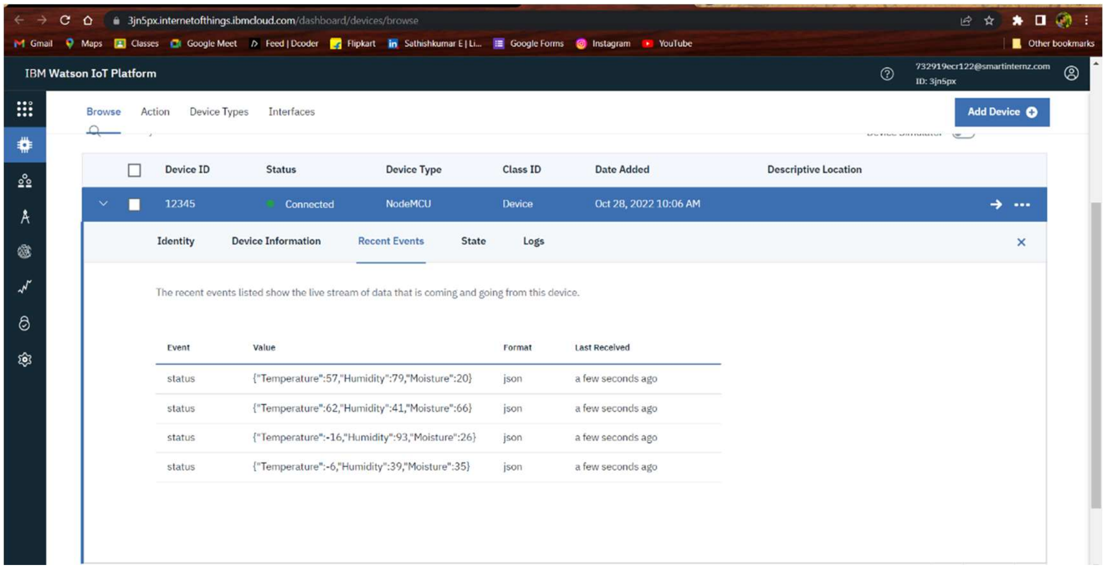
IoT development requires a database to store generated data. MySQL provides IoT app developers the go-to relational database. It is the most convenient tool that evades the requirements to execute shell commands within a Python script. In the IoT development other programming languages like C, C++, Assembly, Java, JavaScript, and PHP. Python is the developers’ favourite programming language when it comes to the development of IoT applications.

- Python has a clear syntax that gives developers the idea of code identification instead of {}
- Its syntax is similar to the English language

- Python has an interpreter system which it runs on. The developers can easily execute the code as soon as it is written. It also facilitates quick prototyping.
- Integration with other languages is possible while using Python. Python app developers can easily put their code in other programming languages including C++, Java, etc.
- An extensible language that enables developers to write programs with fewer lines.
- It has code portability. The developers do not have to write new codes every time for different machines.
- An Open-source framework that is easily available for public download.
- Huge community support enables developers to create user-friendly apps each time.

CHAPTER 4

EXPERIMENTAL INVESTIGATION



	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
□	12345	● Connected	NodeMCU	Device	Oct 28, 2022 10:06 AM	

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
status	{"Temperature":57,"Humidity":79,"Moisture":20}	json	a few seconds ago
status	{"Temperature":62,"Humidity":41,"Moisture":66}	json	a few seconds ago
status	{"Temperature":-16,"Humidity":93,"Moisture":26}	json	a few seconds ago
status	{"Temperature":-6,"Humidity":39,"Moisture":35}	json	a few seconds ago

Fig (i) Receiving data from IoT simulator to the IBM Watson IOT platform through IBM cloud

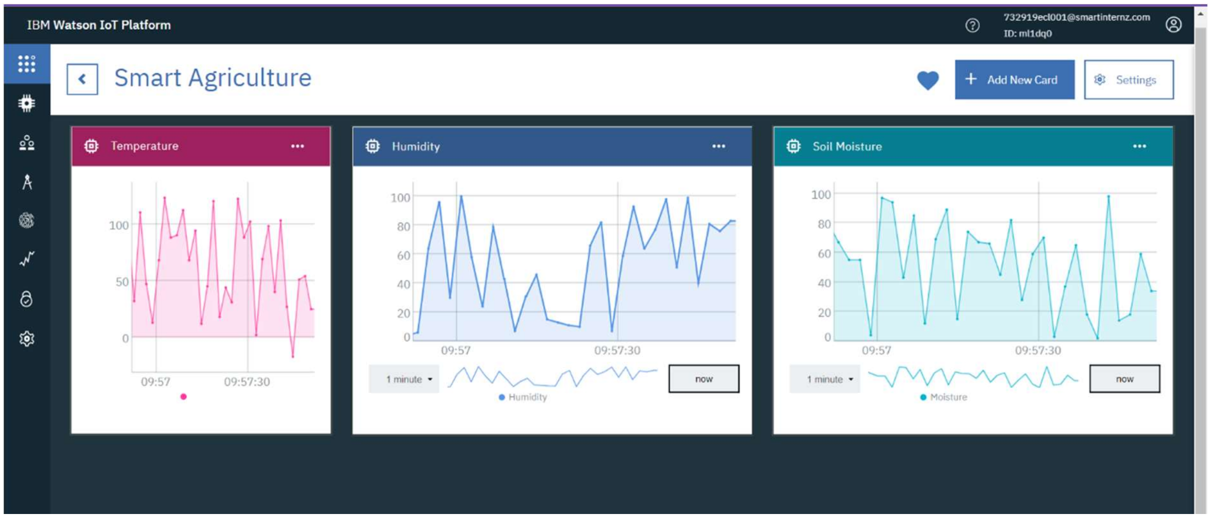
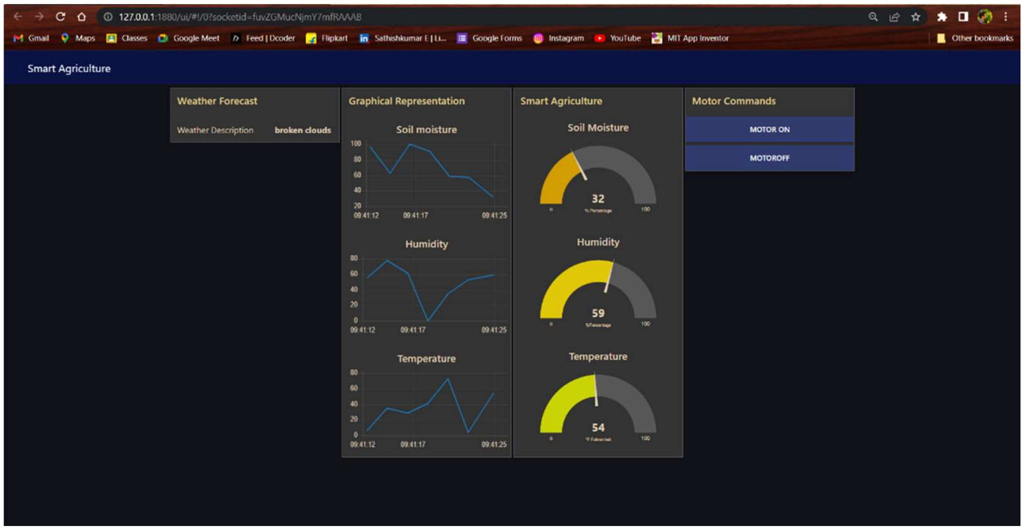
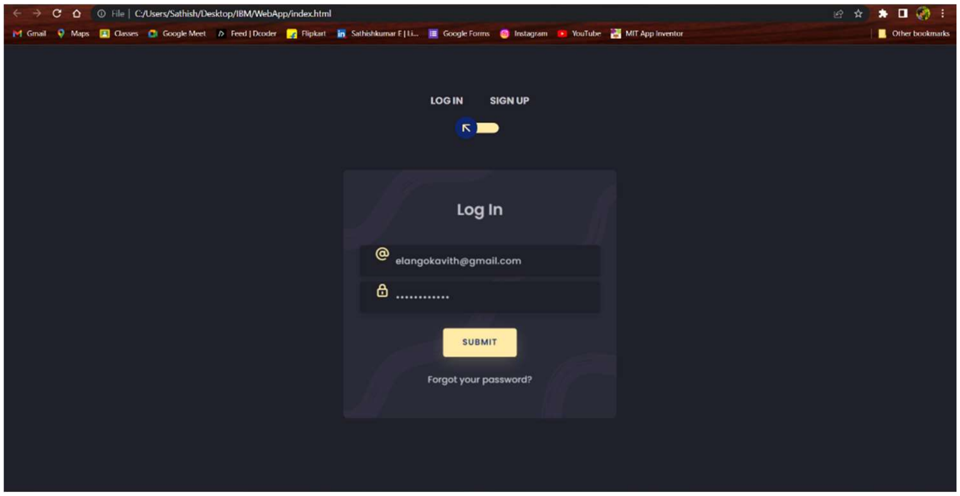


Fig (ii) Representation of Received data through graphs

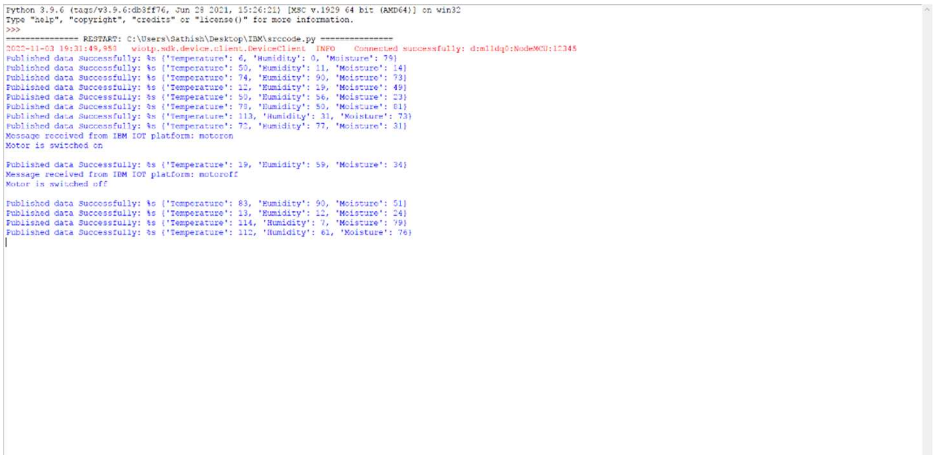




elangokavith@gmail.com

....

Fig (iii, iv) Web Application for Smart Agriculture System

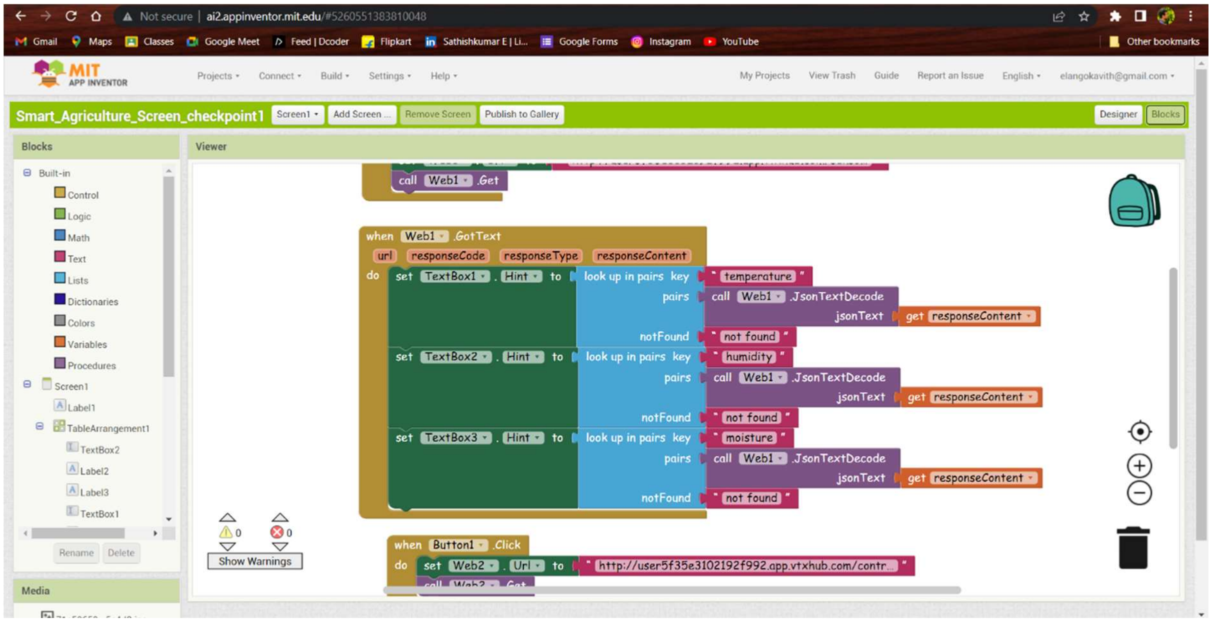


```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:\Users\Sathish\Desktop\IBM\srccode.py ===============
2022-11-03 19:31:49,958   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:m11dq0:NodeMCU:12345
Published data Successfully: %s {'Temperature': 6, 'Humidity': 0, 'Moisture': 79}
Published data Successfully: %s {'Temperature': 50, 'Humidity': 11, 'Moisture': 14}
Published data Successfully: %s {'Temperature': 74, 'Humidity': 90, 'Moisture': 73}
Published data Successfully: %s {'Temperature': 12, 'Humidity': 19, 'Moisture': 49}
Published data Successfully: %s {'Temperature': 50, 'Humidity': 56, 'Moisture': 23}
Published data Successfully: %s {'Temperature': 78, 'Humidity': 50, 'Moisture': 81}
Published data Successfully: %s {'Temperature': 113, 'Humidity': 31, 'Moisture': 73}
Published data Successfully: %s {'Temperature': 72, 'Humidity': 77, 'Moisture': 31}
Message received from IBM IOT platform: motoron
Motor is switched on

Published data Successfully: %s {'Temperature': 19, 'Humidity': 59, 'Moisture': 34}
Message received from IBM IOT platform: motoroff
Motor is switched off

Published data Successfully: %s {'Temperature': 83, 'Humidity': 90, 'Moisture': 51}
Published data Successfully: %s {'Temperature': 13, 'Humidity': 12, 'Moisture': 24}
Published data Successfully: %s {'Temperature': 114, 'Humidity': 7, 'Moisture': 79}
Published data Successfully: %s {'Temperature': 112, 'Humidity': 61, 'Moisture': 76}
```

Fig (v) Publishing Data and responding to the controls from cloud

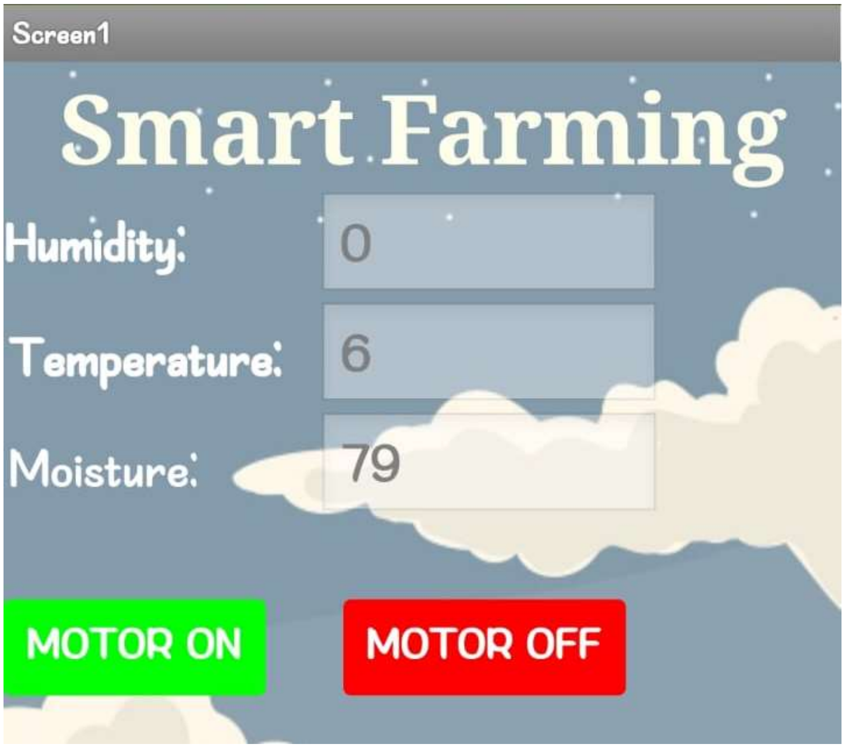


```
call Web1 .Get

when Web1 .GotText
 url  responseCode  responseType  responseContent
do  set TextBox1 . Hint to  look up in pairs  key  " temperature "
                                                pairs  call Web1 .JsonTextDecode
                                                                        jsonText  get responseContent
                                             notFound  " not found "
    set TextBox2 . Hint to  look up in pairs  key  " humidity "
                                                pairs  call Web1 .JsonTextDecode
                                                                        jsonText  get responseContent
                                             notFound  " not found "
    set TextBox3 . Hint to  look up in pairs  key  " moisture "
                                                pairs  call Web1 .JsonTextDecode
                                                                        jsonText  get responseContent
                                             notFound  " not found "

when Button1 .Click
do  set Web2 . Url to  " http://user5f35e3102192f992.app.vtxhub.com/contr... "
```

Fig (vi) Blocks in MIT App Inventor



Screen1

Smart Farming

Humidity: 0

Temperature: 6

Moisture: 79

Fig (vii) Responses in Mobile app

In this project we send the weather data through Python program shown in fig(a) instead of real soil and temperature conditions. Simulator passes the data through IBM Cloud to the web application. The data is displayed on the dashboard show in fig(iii & iv). Web Application is build using Node-RED. We have created 2 tabs:

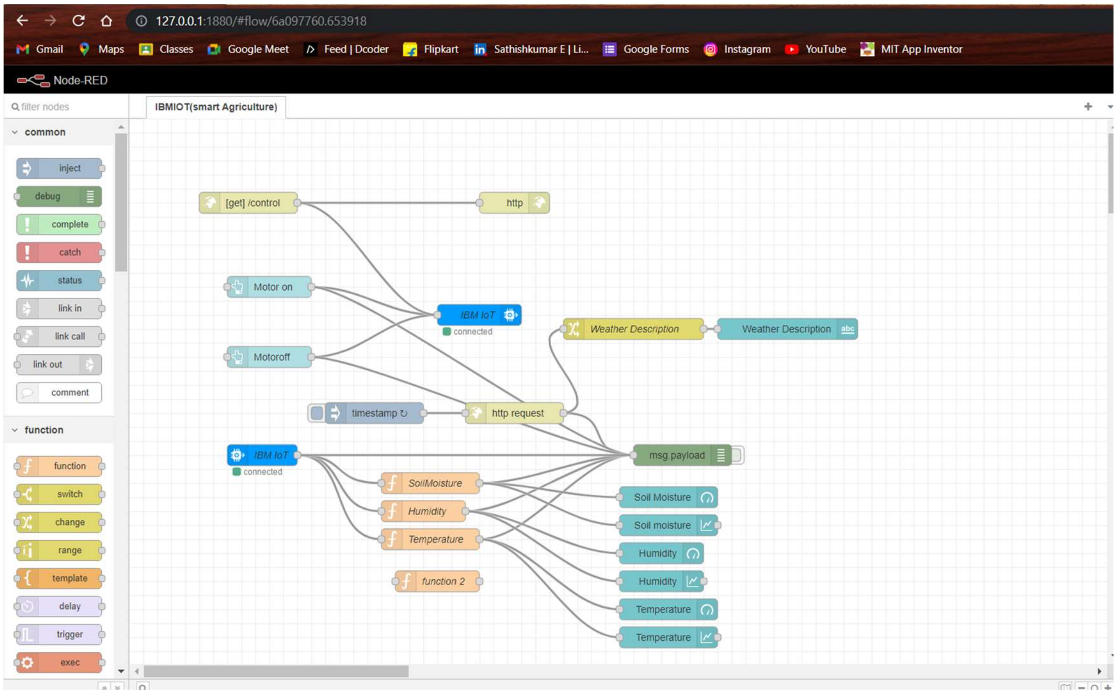
1. IoT Smart Agriculture.

2. Graphical Representation.

Web Application and Mobile application (fig(vii)) is also used to control the devices further like motor, pumps, lights, or any other devices in the agricultural field. In this project the output is passed using python code and the control action is displayed in python code console window in fig (v).

CHAPTER 5

FLOWCHART



Following are the nodes used in the project in the Web Application:

1. IBM IoT : IN and OUT Nodes
2. Function Nodes
3. Gauge Nodes
4. Chart Nodes
5. Debug Node
6. Button Nodes

Following are the nodes used for the weather condition from open weather map:

1. Timestamp Node
2. http request Node
3. Function Nodes
4. Text Nodes
5. Debug Node

CHAPTER 6

ADVANTAGES & DISADVANTAGES

6.1 Advantages

- All the data like climatic conditions and changes in them, soil or crop conditions everything can be easily monitored.
- Risk of crop damage can be lowered to a greater extent.
- Many difficult challenges can be avoided making the process automated and the quality of crops can be maintained.
- The process included in farming can be controlled using the web applications from anywhere, anytime.

6.2 Disadvantages

- Smart Agriculture requires internet connectivity continuously, but rural parts can not fulfill this requirement.
- Any faults in the sensors can cause great loss in the agriculture, due to wrong records and the actions of automated processes.
- IoT devices need much money to implement.

CHAPTER 7

APPLICATIONS

- Precision Farming that is farming processes can be made more controlled and accurate.
- Live monitoring can be done of all the processes and the conditions on the agricultural field.
- All the controls can be made just on the click.
- Quality can be maintained

CHAPTER 8

CONCLUSION

Smart farming can make agriculture more profitable for the farmer. Decreasing resource inputs will save the farmer money and labor, and increased reliability of spatially explicit data will reduce risks. Optimal, site-specific weather forecasts, yield projections, and probability maps for diseases and disasters based on a dense network of weather and climate data will allow cultivation of crops in an optimal way. Site-specific information also enables new insurance and business opportunities for the entire value chain, from technology and input suppliers to farmers, processors, and the retail sector in developing and developed societies alike. If all farming-related data are recorded by automated sensors, the time needed for prioritizing the application of resources and for administrative surveillance is decreased.

Smart farming also has the potential to boost consumer acceptance. In principle, optimizing management also permits increased product quality (e.g., higher amounts of antioxidants and other secondary metabolites based on optimal fruiting densities in orchards; or physiologically more amenable milk products based on individualized feeding rations of livestock). These products are not only healthier but can also sell at higher prices, a key strategy in using land more efficiently. In addition, the transparency of production and processing will increase along value chains because ICT allows registration as to which farm produced a certain product under which circumstances. This offers the potential for new, more direct forms of interaction among farmers and consumers.

CHAPTER 9

FUTURE SCOPE

Smart farming is certainly a leading enabler in producing more food with less for an increasing world population. In particular, smart farming enables increased yield through more efficient use of natural resources and inputs, and improved land and environmental management. While this is crucial to sustainably feeding the world's growing population, there are other benefits that smart farming provides farmers and communities all around the world.

Conventional supply chains have been characterized by a power imbalance with farmers often having less power because they've had less information about how their product performs relative to customer requirements. Smart farming provides a vital link between all players in the supply chain by enabling the efficient and equitable flow of information and in doing so, facilitating better decision making. This has the potential to rebalance power and redistribute profits more equitably throughout the supply chain.

For example, if a farmer receives timely feedback about their product from various parts of the supply chain (such as processors and consumers), they can identify opportunities to change their production system to meet the needs of their customers, thereby increasing the value of their product. Meeting the evolving needs of customers is crucial to farming businesses remaining sustainable into the future and smart farming can provide insights to allow this to happen.

Smart farming also supports verification activities, by connecting information through the supply chain so that production claims can be checked. These may relate to the safety of the food produced (such as making sure that no harmful chemical residues are present), where it was grown, treatment of animals on-farm, or sustainability practices that help protect the environment (such as reducing GHG emissions).

Smart farming helps farmers to better understand the important factors such as water, topography, aspect, vegetation and soil types. This allows farmers to determine the best uses of scarce resources within their production environment and manage these in an environmentally and economically sustainable manner. It also enables farmers to monitor the quantity and quality of their products in a timely manner and to adjust their production techniques when necessary.

For example, satellite imagery can be analysed to determine crop and pasture health using the "normalized difference vegetation index", or to detect pests and diseases earlier than with manual monitoring techniques. By having additional data, the farmer in both examples is able to implement timely and targeted strategies to prevent production losses and increased costs. That protects their livelihood, enables them to continue to supply food and other natural products to the general population, and improves environmental management.

CHAPTER 10

BIBILIOGRAPHY

- IBM Cloud :https://cloud.ibm.com/docs/overview?topic=overview-whatisplatform
- Watson IoT :https://www.iotone.com/software/ibm-watson-iot-platform/s62
- Open weather map: https://openweathermap.org/
- GitHUB: https://github.com/IBM-EPBL/IBM-Project-35205-1660282688
- Node-RED:
    - https://www.youtube.com/watch?v=cicTw4SEdxk
    - https://nodered.org/docs/getting-started/windows#3-run node-red

APPENDIX

A. PYTHON CODE:

```
import wiotp.sdk.device
import time
import random

myConfig = {
    "identity": {
        "orgId": "ml1dq0",
        "typeId": "NodeMCU",
        "deviceId":"12345"
    },
    "auth": {
        "token": "12345678987654321"
    }
}

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(" ")

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

while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    moist=random.randint(0,100)
    myData={'Temperature':temp, 'Humidity':hum,
'Moisture':moist }
    client.publishEvent(eventId="status", msgFormat="json",
data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

B. HTML & CSS Code

INDEX.HTML

```
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>LOG IN >> SIGN UP</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/4.5.0/css/bootstrap.min.css">
    <link rel="stylesheet"
href="https://unicons.iconscout.com/release/v2.1.9/css/unicons.css">
    <link rel="stylesheet" href="style.css" />
    <script>
        function verifyPassword() {
            var pw = document.getElementById("logpass").value;
  //check empty password field
            if(pw == "") {
            document.getElementById("message").innerHTML = "**Fill the
password please!";
            return false;
             }

 //minimum password length validation
            if(pw.length < 8) {
                 document.getElementById("message").innerHTML =
"**Password length must be atleast 8 characters";
                return false;
             }

//maximum length of password validation
             if(pw.length > 15) {
                 document.getElementById("message").innerHTML =
"**Password length must not exceed 15 characters";
                return false;
            } else {
                alert("Password is correct");
                return true;
            }
        }
    </script>
</head>

<body>
```

```
    <div class="section">
        <div class="container">
        <div class="row full-height justify-content-center">
        <div class="col-12 text-center align-self-center py-5">
        <div class="section pb-5 pt-5 pt-sm-2 text-center">
        <h6 class="mb-0 pb-3"><span>Log In </span><span>Sign 
Up</span></h6>
        <input class="checkbox" type="checkbox" id="reg-log" name="reg-
log" />
        <label for="reg-log"></label>
        <div class="card-3d-wrap mx-auto">
        <div class="card-3d-wrapper">
        <div class="card-front">
        <div class="center-wrap">
        <div class="section text-center">
        <h4 class="mb-4 pb-3">Log In</h4>
        <div onsubmit ="return verifyPassword()" class="form-group">
        <input type="email" name="logemail" class="form-style" 
placeholder="Your Email" id="logemail" autocomplete="on">
        <i class="input-icon uil uil-at"></i>
        </div>
        <div class="form-group mt-2">
        <input type="password" name="logpass" class="form-style" 
placeholder="Your Password" id="logpass" autocomplete="on">
        <i class="input-icon uil uil-lock-alt"></i>
        </div>
        <a 
href="http://127.0.0.1:1880/ui/#!/0?socketid=ZE1x1DgPUklFtif1AAAD" 
class="btn mt-4" type="submit">submit</a>
        <p class="mb-0 mt-4 text-center"><a href="#0" 
class="link">Forgot your 
                                                    password?</a></p>
        </div>
        </div>
        </div>
        <div class="card-back">
        <div class="center-wrap">
        <div class="section text-center">
        <h4 class="mb-4 pb-3">Sign Up</h4>
        <div class="form-group">
        <input type="text" name="logname" class="form-style" 
placeholder="Your Full Name" id="logname" autocomplete="on">
        <i class="input-icon uil uil-user"></i>
        </div>
        <div class="form-group mt-2">
        <input type="email" name="logemail" class="form-style" 
placeholder="Your Email" id="logemail" autocomplete="on">
        <i class="input-icon uil uil-at"></i>
        </div>
        <div class="form-group mt-2">
        <input type="password" name="logpass" class="form-style"
placeholder="Your Password" id="logpass" autocomplete="off">
        <i class="input-icon uil uil-lock-alt"></i>
         </div>
         <a href="#" class="btn mt-4">submit</a>
         </div>
         </div>
         </div>
         </div>
         </div>
         </div>
         </div>
         </div>
         </div>
         </div>
       </body>
</html>
```

STYLES.CSS

```
@import
url(https://fonts.googleapis.com/css?family=Poppins:400,500,600,700,800,900
);

body {
  font-family: "Poppins", sans-serif;
  font-weight: 300;
  font-size: 15px;
  line-height: 1.7;
  color: #c4c3ca;
  background-color: #1f2029;
  overflow-x: hidden;
}
a {
  cursor: pointer;
  transition: all 200ms linear;
}
a:hover {
  text-decoration: none;
}
.link {
  color: #c4c3ca;
}
.link:hover {
  color: #ffeba7;
}
p {
  font-weight: 500;
  font-size: 14px;
  line-height: 1.7;
}
h4 {
  font-weight: 600;
}
h6 span {
  padding: 0 20px;
  text-transform: uppercase;
  font-weight: 700;
}
.section {
  position: relative;
  width: 100%;
  display: block;
}
.full-height {
  min-height: 100vh;
}
[type="checkbox"]:checked,
[type="checkbox"]:not(:checked) {
  position: absolute;
  left: -9999px;
}
.checkbox:checked + label,
.checkbox:not(:checked) + label {
  position: relative;
  display: block;
  text-align: center;
  width: 60px;
  height: 16px;
  border-radius: 8px;
  padding: 0;
  margin: 10px auto;
  cursor: pointer;
  background-color: #ffeba7;
}
.checkbox:checked + label:before,
.checkbox:not(:checked) + label:before {
  position: absolute;
  display: block;
  width: 36px;
  height: 36px;
  border-radius: 50%;
  color: #ffeba7;
  background-color: #102770;
  font-family: "unicons";
  content: "\eb4f";
  z-index: 20;
  top: -10px;
  left: -10px;
  line-height: 36px;
  text-align: center;
  font-size: 24px;
  transition: all 0.5s ease;
}
.checkbox:checked + label:before {
  transform: translateX(44px) rotate(-270deg);
}

.card-3d-wrap {
  position: relative;
  width: 440px;
  max-width: 100%;
  height: 400px;
  -webkit-transform-style: preserve-3d;
  transform-style: preserve-3d;
  perspective: 800px;
  margin-top: 60px;
}
.card-3d-wrapper {
  width: 100%;
  height: 100%;
  position: absolute;
  top: 0;
  left: 0;
  -webkit-transform-style: preserve-3d;
  transform-style: preserve-3d;
  transition: all 600ms ease-out;
}
.card-front,
.card-back {
  width: 100%;
  height: 100%;
  background-color: #2a2b38;
  background-image: url("https://s3-us-west-
2.amazonaws.com/s.cdpn.io/1462889/pat.svg");
  background-position: bottom center;
  background-repeat: no-repeat;
  background-size: 300%;
  position: absolute;
  border-radius: 6px;
  left: 0;
  top: 0;
  -webkit-transform-style: preserve-3d;
  transform-style: preserve-3d;
  -webkit-backface-visibility: hidden;
  -moz-backface-visibility: hidden;
  -o-backface-visibility: hidden;
  backface-visibility: hidden;
}
.card-back {
  transform: rotateY(180deg);
}
.checkbox:checked ~ .card-3d-wrap .card-3d-wrapper {
  transform: rotateY(180deg);
}
.center-wrap {
  position: absolute;
  width: 100%;
  padding: 0 35px;
  top: 50%;
  left: 0;
  transform: translate3d(0, -50%, 35px) perspective(100px);
  z-index: 20;
  display: block;
}

.form-group {
  position: relative;
  display: block;
  margin: 0;
  padding: 0;
}
.form-style {
  padding: 13px 20px;
  padding-left: 55px;
  height: 48px;
  width: 100%;
  font-weight: 500;
  border-radius: 4px;
  font-size: 14px;
  line-height: 22px;
  letter-spacing: 0.5px;
  outline: none;
  color: #c4c3ca;
  background-color: #1f2029;
  border: none;
  -webkit-transition: all 200ms linear;
  transition: all 200ms linear;
  box-shadow: 0 4px 8px 0 rgba(21, 21, 21, 0.2);
}
.form-style:focus,
.form-style:active {
  border: none;
  outline: none;
  box-shadow: 0 4px 8px 0 rgba(21, 21, 21, 0.2);
}
.input-icon {
  position: absolute;
  top: 0;
  left: 18px;
  height: 48px;
  font-size: 24px;
  line-height: 48px;
  text-align: left;
  color: #ffeba7;
  -webkit-transition: all 200ms linear;
  transition: all 200ms linear;
}

.form-group input:-ms-input-placeholder {
  color: #c4c3ca;
  opacity: 0.7;
  -webkit-transition: all 200ms linear;
  transition: all 200ms linear;
}
.form-group input::-moz-placeholder {
  color: #c4c3ca;
  opacity: 0.7;
  -webkit-transition: all 200ms linear;
  transition: all 200ms linear;
}
.form-group input:-moz-placeholder {
  color: #c4c3ca;
  opacity: 0.7;
  -webkit-transition: all 200ms linear;
  transition: all 200ms linear;
}
.form-group input::-webkit-input-placeholder {
  color: #c4c3ca;
  opacity: 0.7;
  -webkit-transition: all 200ms linear;
  transition: all 200ms linear;
}
.form-group input:focus:-ms-input-placeholder {
  opacity: 0;
  -webkit-transition: all 200ms linear;
  transition: all 200ms linear;
}
.form-group input:focus::-moz-placeholder {
```

```css
  opacity: 0;
  -webkit-transition: all 200ms linear;
  transition: all 200ms linear;
}
.form-group input:focus:-moz-placeholder {
  opacity: 0;
  -webkit-transition: all 200ms linear;
  transition: all 200ms linear;
}
.form-group input:focus::-webkit-input-placeholder {
  opacity: 0;
  -webkit-transition: all 200ms linear;
  transition: all 200ms linear;
}

.btn {
  border-radius: 4px;
  height: 44px;
  font-size: 13px;
  font-weight: 600;
  text-transform: uppercase;
  -webkit-transition: all 200ms linear;
  transition: all 200ms linear;
  padding: 0 30px;
  letter-spacing: 1px;
  display: -webkit-inline-flex;
  display: -ms-inline-flexbox;
  display: inline-flex;
  -webkit-align-items: center;
  -moz-align-items: center;
  -ms-align-items: center;
  align-items: center;
  -webkit-justify-content: center;
  -moz-justify-content: center;
  -ms-justify-content: center;
  justify-content: center;
  -ms-flex-pack: center;
  text-align: center;
  border: none;
  background-color: #ffeba7;
  color: #102770;
  box-shadow: 0 8px 24px 0 rgba(255, 235, 167, 0.2);
}
.btn:active,
.btn:focus {
  background-color: #102770;
  color: #ffeba7;
  box-shadow: 0 8px 24px 0 rgba(16, 39, 112, 0.2);
```

```
}
.btn:hover {
  background-color: #102770;
  color: #ffeba7;
  box-shadow: 0 8px 24px 0 rgba(16, 39, 112, 0.2);
}

.logo {
  position: absolute;
  top: 30px;
  right: 30px;
  display: block;
  z-index: 100;
  transition: all 250ms linear;
}
.logo img {
  height: 26px;
  width: auto;
  display: block;
}
```