

ABSTRACT:

In India, about *11.8 crore* farmer is living in various states across the country, Since the industrial revolution has upgraded a lot of the fields it also brought a huge impact in the agricultural field . But in developing countries like India, the agricultural field has not achieved its considerable momentum. There are about 296,438 Indian farmers had committed suicide since 1995. Inorder to bring in the required momentum (INTERNET OF THINGS) IOT a very handy technology will help to boost up the agricultural industry with an intelligent application designed to advise the farmers about the climatic changes and more information about their respectively fields .Also help them to controls their motors for anywhere

Table of Content:

S.NO	CONTENT	PAGE NO
1	INTRODUCTION <ul style="list-style-type: none">• OVERVIEW• OUTCOME	3
2.	Problem and Solution <ul style="list-style-type: none">• Existing problem• solution	5
3.	Diagramatic Sketch:	6
4.	Skills and Tools	6
5.	Source code and Output	24
6.	Advantage and link	30

INTRODUCTION:

1. OVERVIEW:

Digital agriculture refers to tools that digitally collect, store, analyze, and share electronic data and/or information along the agricultural value chain. Digital technology changes economic activity by lowering the costs of replicating, transporting, tracking, verifying, and searching for data. Due to these falling costs, digital technology will improve efficiency throughout the agricultural value chain. The proposed system enables the farmer with the climatic data and helps them to control the motor with the internet.

2. OUTCOME:

- The digital platform provides the climatic data by which farmer can understand the climate in an efficient way
- Field data to understand the crop in an efficient way
- Motors can be controlled with this digital platform so that farmer can control motors from anywhere.

3.Scope of work:

- A device being created from IBM Account.
- Install Node-RED
- get the API key from openwheather.org
- Application (.apk) created for user interaction

PROBLEM AND SOLUTION:

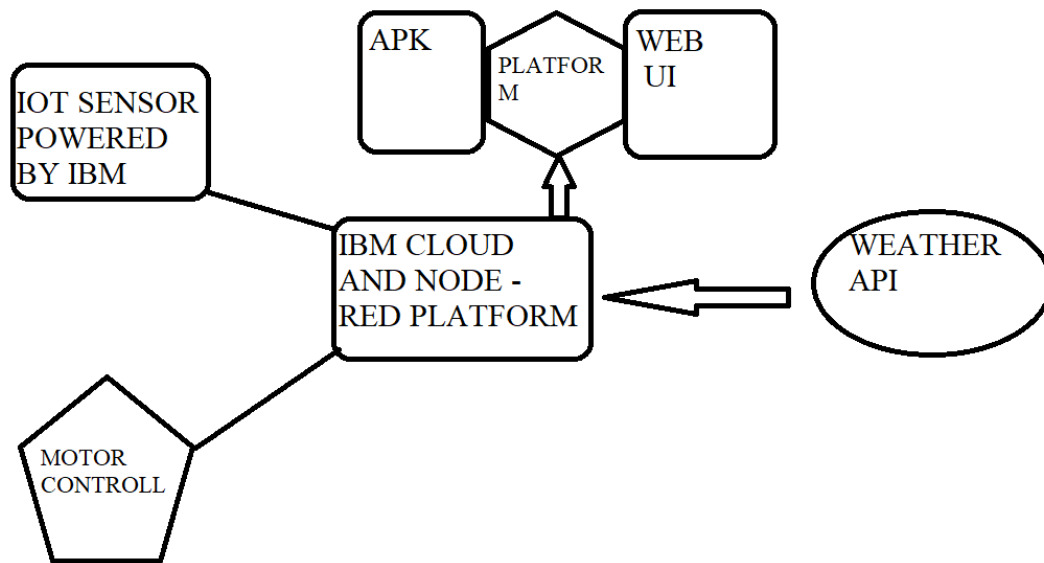
1.EXISTING PROBLEM:

- Lack of weather forecasting knowledge
- Lack of agricultural field understanding
- Inefficient usage of water
- Lack of advice about the specific field

2.Solution:

- Digital platform with integrated sensor data
- User interaction to know about the field data
- Climatic forecasting
- Enables the motor controlling through the power of the internet

Diagrammatic Sketch:



SKILL AND TOOLS:

1. SKILLS REQUIRED:

- IOT Application Development
- IBM cloud
- Node-red platform

2.TOOLS:

- IBM Cloud to integrate the data
- Iot simulator
- Digital platform android and web UI to view the data and outcome
- Weather forecasting API integrated with apk

CODE :

1.NODE-RED FLOW:

```
[
  {
    "id":"3b9a4f06.57db7",
    "type":"tab",
    "label":"Flow 1",
    "disabled":false,
    "info":""
  },
  {
    "id":"8b6c6a33.315118",
    "type":"ibmiot in",
    "z":"3b9a4f06.57db7",
    "authentication":"apiKey",
    "apiKey":"6a1ac3e2.d74e5c",
    "inputType":"evt",
    "logicalInterface":"","
    "ruleId":"","
    "deviceId":"0001",
    "applicationId":"","
    "deviceType":"controllerdata",
    "eventType":"+",
    "commandType":"","
    "format":"json",
    "name":"iot_data",
    "service":"registered",
    "allDevices":"","
```

```
"allApplications":"","  
"allDeviceTypes":"","  
"allLogicalInterfaces":"","  
"allEvents":true,  
"allCommands":"","  
"allFormats":"","  
"qos":0,  
"x":90,  
"y":80,  
"wires":[  
  [  
    "cd97e148.fa108",  
    "f64bbb0c.b10fa8",  
    "1cde58e4.ec6557",  
    "ee5ef002.87a51"  
  ]  
],  
{  
  "id":"cd97e148.fa108",  
  "type":"debug",  
  "z":"3b9a4f06.57db7",  
  "name":"display",  
  "active":true,  
  "tosidebar":true,  
  "console":false,  
  "tostatus":false,  
  "complete":"payload",
```



```
    "targetType":"msg",
    "x":760,
    "y":80,
    "wires":[

    ]
  },
  {
    "id":"f64bbb0c.b10fa8",
    "type":"function",
    "z":"3b9a4f06.57db7",
    "name":"temperature",

    "func":"global.set('temperature',msg.payload.d.tem
perature)\nreturn
{payload:msg.payload.d.temperature}",
    "outputs":1,
    "noerr":0,
    "x":300,
    "y":140,
    "wires":[
      [
        "1888aae1.34a475",
        "cd97e148.fa108"
      ]
    ]
  },
  {
```

```
"id":"1cde58e4.ec6557",
"type":"function",
"z":"3b9a4f06.57db7",
"name":"humadity",

"func":"global.set('humidity',msg.payload.d.humidi
ty)\nreturn {payload:msg.payload.d.humidity}",
"outputs":1,
"noerr":0,
"x":290,
"y":220,
"wires":[
  [
    "47a6bb52.d6bde4",
    "cd97e148.fa108"
  ]
],
},
{
  "id":"ee5ef002.87a51",
  "type":"function",
  "z":"3b9a4f06.57db7",
  "name":"object temperature",

  "func":"global.set('objectTemp',msg.payload.d.obje
ctTemp)\nreturn
{payload:msg.payload.d.objectTemp}",
  "outputs":1,
```

```
"noerr":0,
"x":300,
"y":320,
"wires":[
  [
    "36ece532.17386a",
    "cd97e148.fa108"
  ]
],
{
  "id":"1888aae1.34a475",
  "type":"ui_gauge",
  "z":"3b9a4f06.57db7",
  "name":"temperature",
  "group":"12d0d1a6.80c83e",
  "order":0,
  "width":0,
  "height":0,
  "gtype":"gage",
  "title":"temperature",
  "label":"C",
  "format":"{{ value }}",
  "min":0,
  "max":"100",
  "colors":[
    "#00b500",
    "#e6e600",
```

```
        "#ca3838"  
    ],  
    "seg1": "",  
    "seg2": "",  
    "x": 760,  
    "y": 140,  
    "wires": [  
  
    ]  
},  
{  
    "id": "36ece532.17386a",  
    "type": "ui_gauge",  
    "z": "3b9a4f06.57db7",  
    "name": "object_temperature",  
    "group": "12d0d1a6.80c83e",  
    "order": 2,  
    "width": 0,  
    "height": 0,  
    "gtype": "gage",  
    "title": "objtem",  
    "label": "%",  
    "format": "{{value}}",  
    "min": 0,  
    "max": "100",  
    "colors": [  
        "#ff8080",  
        "#fa7405",
```

```
        "#ca3838"  
    ],  
    "seg1": "",  
    "seg2": "",  
    "x": 720,  
    "y": 320,  
    "wires": [  
  
    ]  
},  
{  
    "id": "47a6bb52.d6bde4",  
    "type": "ui_gauge",  
    "z": "3b9a4f06.57db7",  
    "name": "humidity",  
    "group": "12d0d1a6.80c83e",  
    "order": 1,  
    "width": 0,  
    "height": 0,  
    "gtype": "gage",  
    "title": "humidity",  
    "label": "%",  
    "format": "{{ value }}",  
    "min": 0,  
    "max": "100",  
    "colors": [  
        "#80ff80",  
        "#008000",
```

```
        "#00530b"
    ],
    "seg1": "",
    "seg2": "",
    "x": 750,
    "y": 220,
    "wires": [

    ]
},
{
    "id": "43858b37.3ffb14",
    "type": "ui_button",
    "z": "3b9a4f06.57db7",
    "name": "",
    "group": "12d0d1a6.80c83e",
    "order": 3,
    "width": 0,
    "height": 0,
    "passthru": false,
    "label": "MOTOR_ON",
    "tooltip": "",
    "color": "",
    "bgcolor": "",
    "icon": "",
    "payload": "{ \"command\": \"MOTOR ON\" }",
    "payloadType": "json",
    "topic": "",
```

```
"x":250,
"y":660,
"wires":[
  [
    "2ec7e769.4f7f78",
    "4966e5fd.5ca4cc",
    "5d403a9e.774974"
  ]
],
{
  "id":"2b8a58d2.eaa268",
  "type":"ui_button",
  "z":"3b9a4f06.57db7",
  "name": "",
  "group":"12d0d1a6.80c83e",
  "order":4,
  "width":0,
  "height":0,
  "passthru":false,
  "label":"MOTOR_OFF",
  "tooltip": "",
  "color": "",
  "bgcolor": "",
  "icon": "",
  "payload": "{\\\"command\\\":\\\"MOTOR OFF\\\"}",
  "payloadType":"json",
  "topic": ""
```

```
"x":250,  
"y":760,  
"wires":[  
  [  
    "2ec7e769.4f7f78",  
    "4966e5fd.5ca4cc",  
    "5d403a9e.774974"  
  ]  
],  
{  
  "id":"2ec7e769.4f7f78",  
  "type":"debug",  
  "z":"3b9a4f06.57db7",  
  "name":"DISPLAY",  
  "active":true,  
  "tosidebar":true,  
  "console":false,  
  "tostatus":false,  
  "complete":"payload",  
  "targetType":"msg",  
  "x":650,  
  "y":720,  
  "wires":[  
  
  ]  
},  
{
```



```
"id":"5d403a9e.774974",
"type":"function",
"z":"3b9a4f06.57db7",
"name":"command",

"func":"msg.payload=msg.payload.command\nreturn msg;",
  "outputs":1,
  "noerr":0,
  "x":380,
  "y":1140,
  "wires":[
    [
      "296ff002.13e2c",
      "c54d955b.386c98"
    ]
  ]
},
{
  "id":"296ff002.13e2c",
  "type":"debug",
  "z":"3b9a4f06.57db7",
  "name":"",
  "active":true,
  "tosidebar":true,
  "console":false,
  "tostatus":false,
  "complete":"true",
```

```
    "targetType":"full",
    "x":750,
    "y":1200,
    "wires":[

    ]
  },
  {
    "id":"f3adb356.75c96",
    "type":"http in",
    "z":"3b9a4f06.57db7",
    "name":"",
    "url":"/controllmsg",
    "method":"get",
    "upload":false,
    "swaggerDoc":"",
    "x":160,
    "y":1120,
    "wires":[
      [
        "5d403a9e.774974"
      ]
    ]
  },
  {
    "id":"f420b4ea.f15b48",
    "type":"function",
    "z":"3b9a4f06.57db7",
```

```
"name":"http_responce",

"func":"msg.payload={\"temperature\":global.get('t
emperature'),\"humidity\":global.get('humidity'),\"o
bjectTemp\":global.get('objectTemp')}\nreturn
msg;",
  "outputs":1,
  "noerr":0,
  "x":380,
  "y":460,
  "wires":[
    [
      "d74efd92.8e18d"
    ]
  ],
},
{
  "id":"d74efd92.8e18d",
  "type":"http response",
  "z":"3b9a4f06.57db7",
  "name": "",
  "statusCode": "",
  "headers":{

  },
  "x":610,
  "y":460,
  "wires":[
```

```
]
},
{
  "id":"3dc4343e.17154c",
  "type":"http in",
  "z":"3b9a4f06.57db7",
  "name": "",
  "url":"/msg",
  "method":"get",
  "upload":false,
  "swaggerDoc": "",
  "x":120,
  "y":440,
  "wires":[
    [
      "f420b4ea.f15b48"
    ]
  ]
},
{
  "id":"4966e5fd.5ca4cc",
  "type":"ibmiot out",
  "z":"3b9a4f06.57db7",
  "authentication":"apiKey",
  "apiKey":"6a1ac3e2.d74e5c",
  "outputType":"cmd",
  "deviceId":"0001",
```

```
"deviceType":"controllerdata",
"eventCommandType":"home",
"format":"json",
"data":"data",
"qos":0,
"name":"IBM IoT",
"service":"registered",
"x":740,
"y":940,
"wires":[

]
},
{
  "id":"c54d955b.386c98",
  "type":"http response",
  "z":"3b9a4f06.57db7",
  "name": "",
  "statusCode": "",
  "headers":{

  },
  "x":600,
  "y":1100,
  "wires":[

]
},
```

```
{
  "id":"6a1ac3e2.d74e5c",
  "type":"ibmiot",
  "z": "",
  "name":"iot_agri_api",
  "keepalive":"60",

  "serverName":"o9hvpe.messaging.internetofthings.
ibmcloud.com",
  "cleansession":true,
  "appId": "",
  "shared":false
},
{
  "id":"12d0d1a6.80c83e",
  "type":"ui_group",
  "z": "",
  "name":"smart_agriculture",
  "tab":"ebd39399.4941b",
  "order":1,
  "disp":true,
  "width":"6",
  "collapse":false
},
{
  "id":"ebd39399.4941b",
  "type":"ui_tab",
  "z": "",
```

```

        "name":"Home_climate",
        "icon":"dashboard",
        "disabled":false,
        "hidden":false
    }
]

```

2. PYTHON CODE :

Python code is used to send the command about the motor on/off

```

import time
import sys
import ibmiotf.application
import ibmiotf.device

```

```

# Provide your IBM Watson Device Credentials
organization = "o9hvpe" # organization ID
deviceType = "controllerdata" # device type
deviceId = "0001" # device id
authMethod = "token"
authToken = "8072958226" # token

```

```

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)
    if cmd.data['command'] == 'motor_on':
        print("MOTOR ON")
    elif cmd.data['command'] == 'motor_off':
        print("MOTOR OFF")

```

```

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()
```

```
deviceCli.connect()
```

```
while True:
```

```
    T = 59;
```

```
    H = 51;
```

```
    O = 2;
```

```
    data = {'d':{'temperature': T, 'humidity':  
H, 'objectTemp':O,}}
```

```
    # print data
```

```
    def myOnPublishCallback():
```

```
        print("Published Temperature = %s C" % T,  
"Humidity = %s %" % H, "objectTemp = %s C" % O  
,"to IBM Watson")
```

```
    success = deviceCli.publishEvent("event", "json", data,  
qos=0, on_publish=myOnPublishCallback)  
    if not success:
```



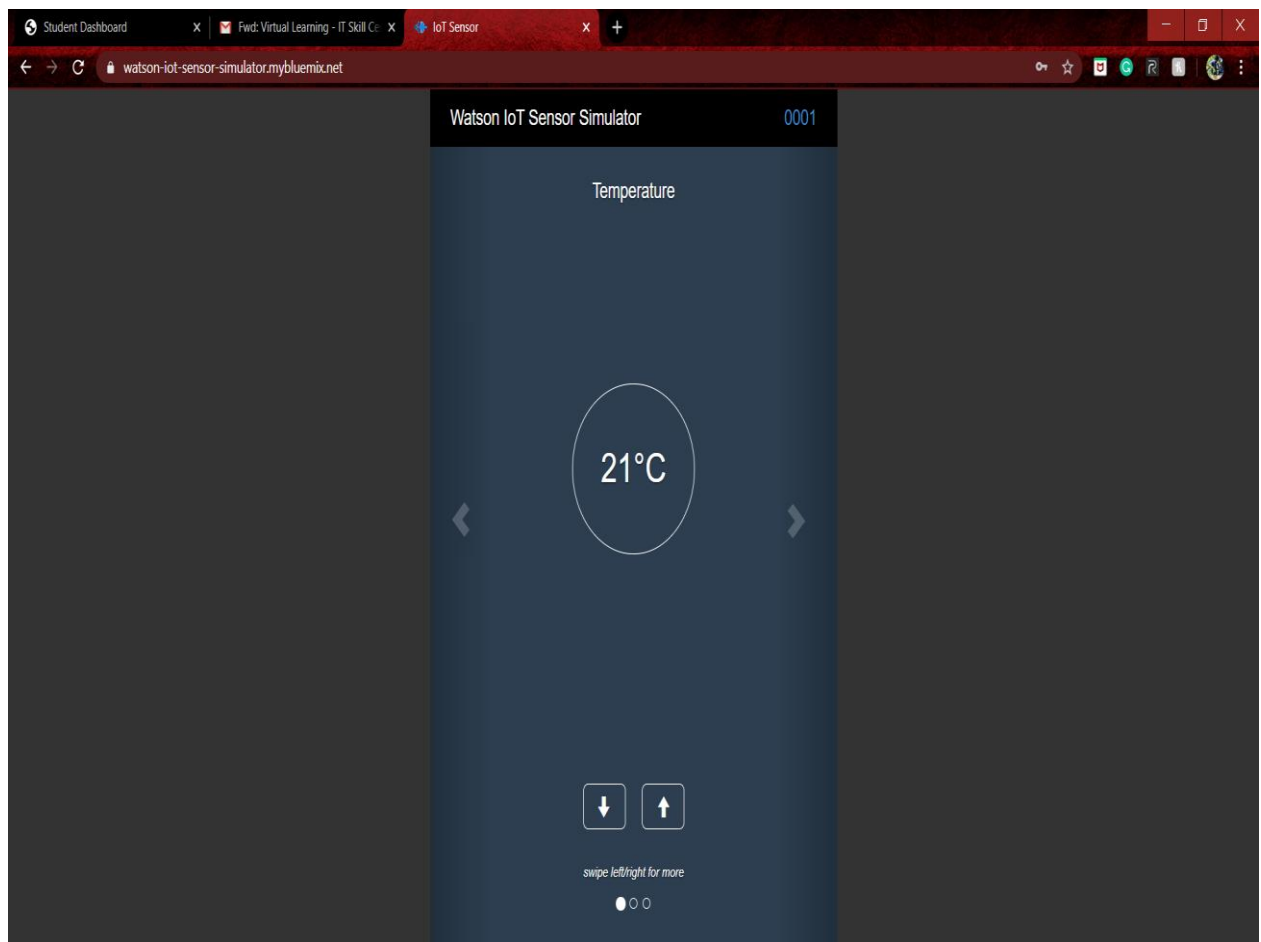
```
print("Not connected to IoT")  
time.sleep(1)
```

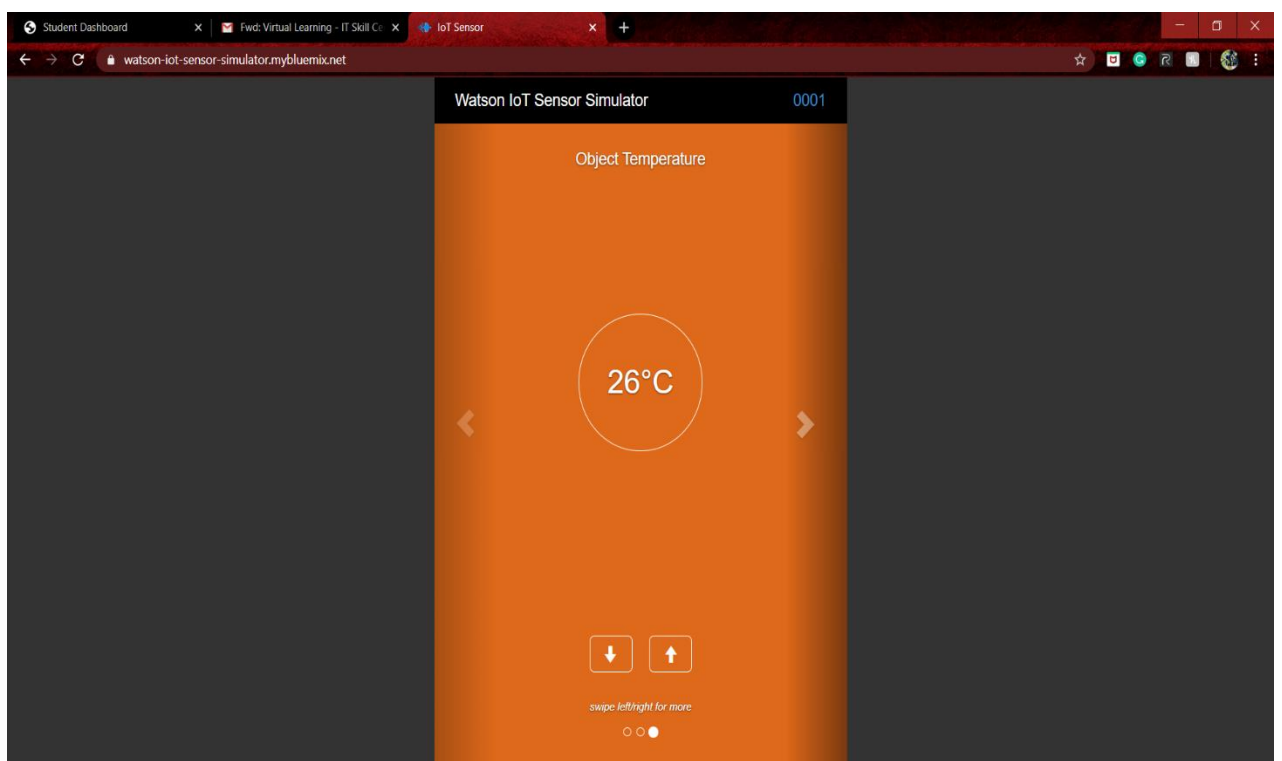
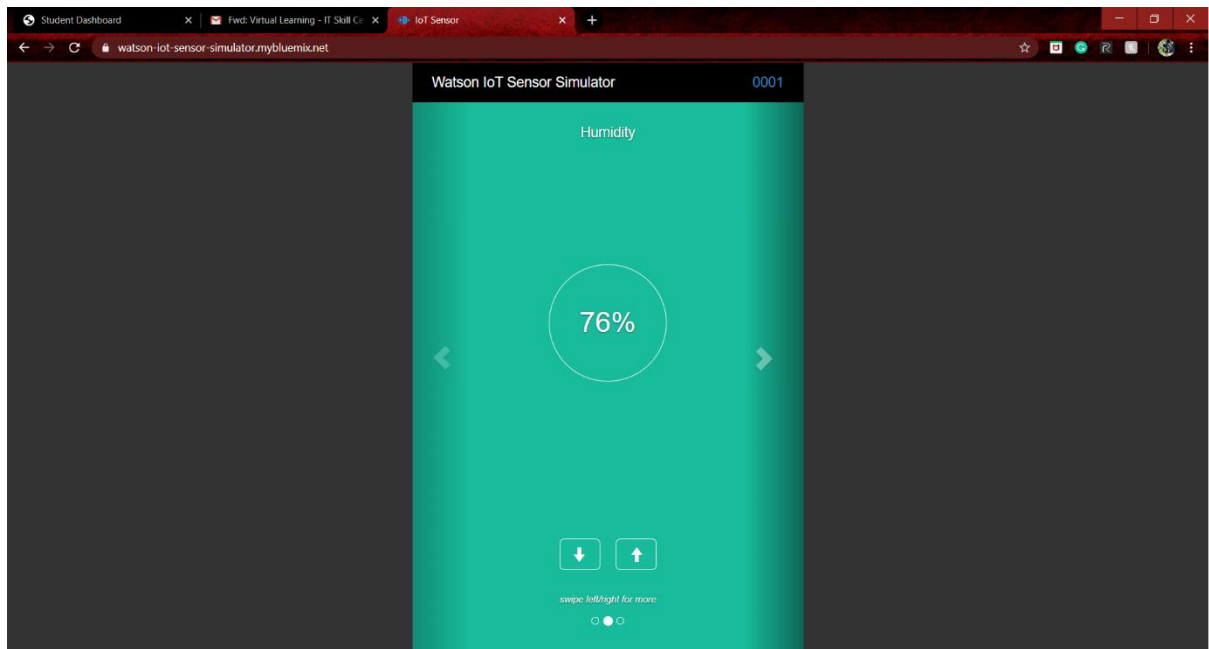
```
deviceCli.commandCallback = myCommandCallback
```

```
deviceCli.disconnect()
```

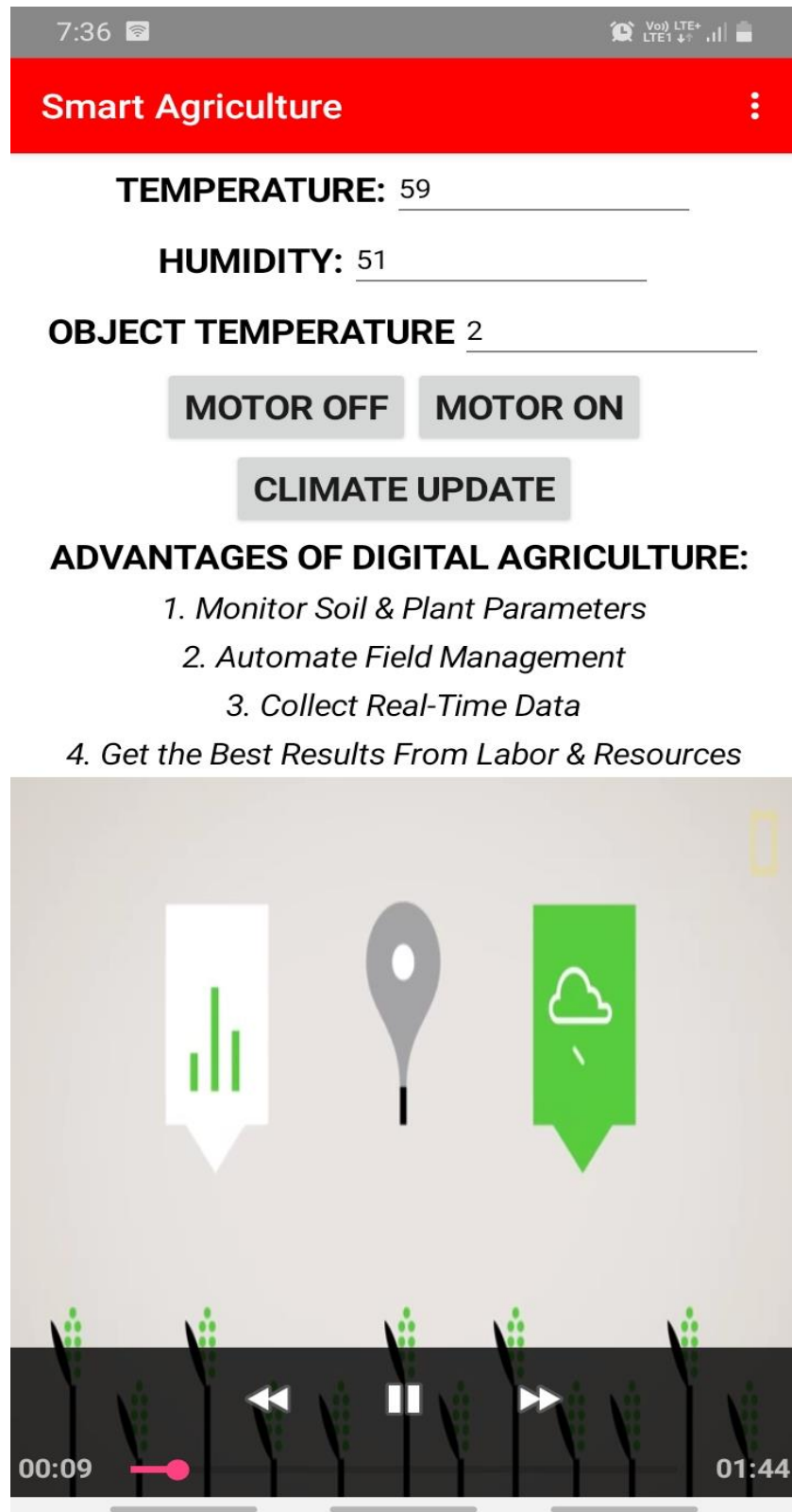
OUTPUT:

1.IOT SENSORS:





ANDROID APP:



7:36

VoLTE+
LTE1

CLIMATE UPDATE



ENTER LOCATION: Mumbai

GET CLIMATE UPDATE

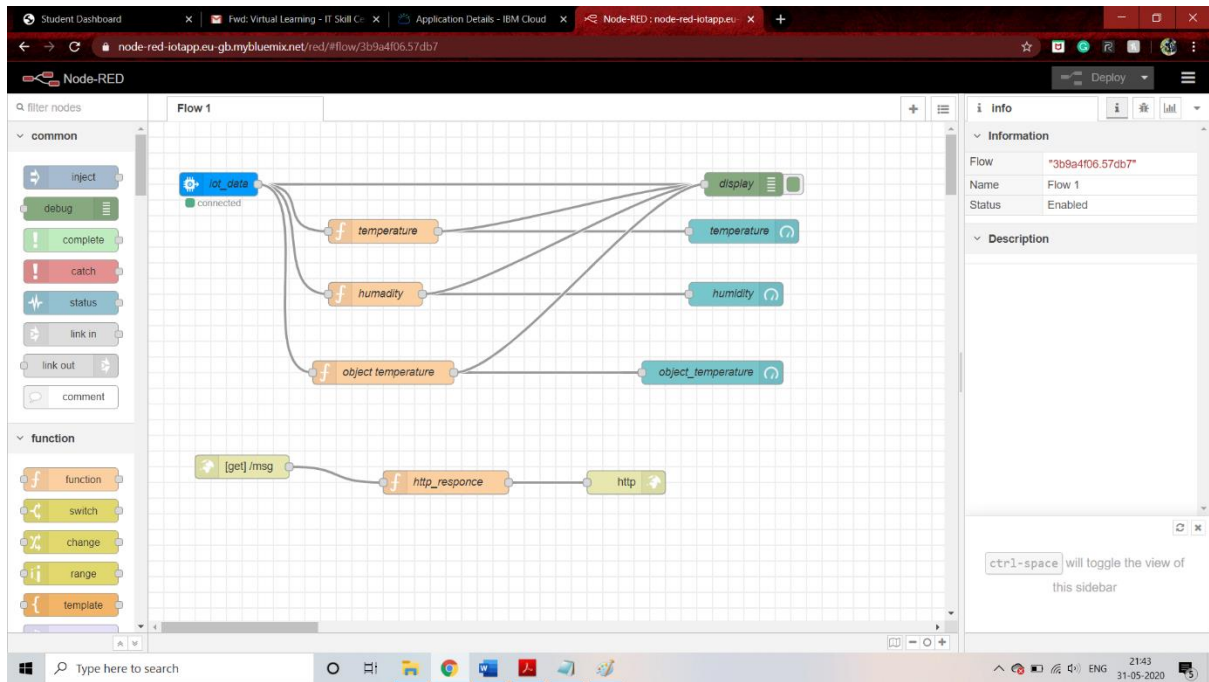
DESCRIPTION : haze

TEMPERATURE : 305.15

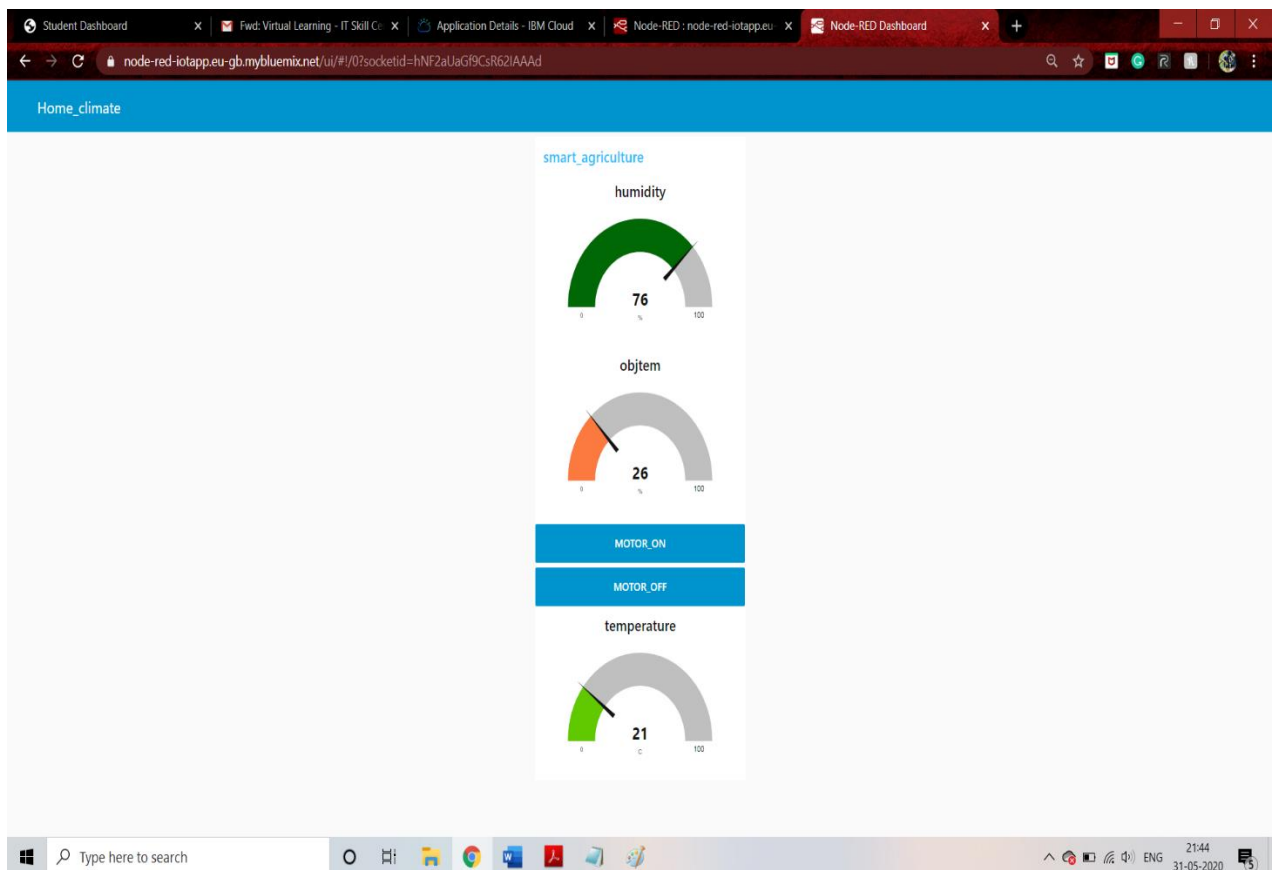
WIND SPEED : 5.1



NODE – RED FLOW:



WEB UI:



ADVANTAGES:

- Digital platform advice the farmer to know about the weather
- Data from agriculture field are displayed in the platform
- The motor on/off can be done in the platform

URL LINKS :

1.For web UI: <https://node-red-iotapp.eu-gb.mybluemix.net/ui/#!/0?socketid=hNF2aUaGf9CsR62IAAAd>

2.Node editor: <https://node-red-iotapp.eu-gb.mybluemix.net/red/#flow/3b9a4f06.57db7>

3.Weather api: <https://openweathermap.org/api>

3.Android apk:

https://drive.google.com/open?id=1WEb0jl6FrOkvAMiRkJqK9MdP5gk6p_mD