**SMART ANIMAL FARM USING IOT**

INTODUCTION:

OVERVIEW:

The applications of Information communication technology have brought a sea change in human life. The present day society is moving towards the adaptation of the digital environment. The earlier ‘internet of computers’ transformed into ‘internet of people’ by introduction of social websites. The next wave is mobile computing. The different generations of internet connection have made it possible for faster accessibility accompanied by better quality. The further advancement of this technology is the ‘Internet of Things’ through which, the interoperability and intelligence can be achieved. This is possible through communication between certain devices that are connected through the internet, wireless sensor networks and smart phones. These devices in the system are able to perceive, process and deliver the product as per the programming. The technologies such as sensors, Cloud Computing, Networking Technology and Nanotechnology have been used. The applications of IoT can be observed in number of areas in animal farms.

**PURPOSE:**

There has been strong relationship between humans and animals throughout the centuries. We depend on animals in many aspects of life such as sports, food, clothes and other product that supports and facilitate our living. Therefore, a good care of animals is very important. The livestock industry could greatly be benefitted from a sophisticated system capable of continuously monitoring the health of animals, aggregating the data and reporting the obtained results to owners and regional

**LITERATURE SURVEY**

**Existing problem:**

In the animal farm its tough to be there all the time and look over the animals and the if absence of the person if there are high temperatures and humidity conditions and ammonia gas levels in the farm rises if there any fire accidents are take places and these conditions should harm the animals in the farm and by these, we should save the animals and keep their environment clean and safe.

**PROPOSED SOLUTION**

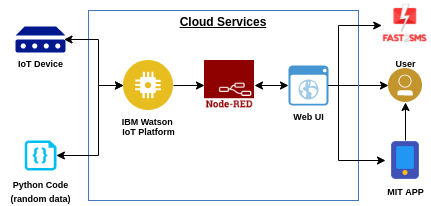
This project is used observe the environment around the animals of temperature, humidity, ammonia gas levels, and observe the water levels in the tank we should turn on blowers and draft fans when temperature and humidity values increase and we can control manually also lights in the farm for continuous supply of water to the farm water pump should be on and off and ammonia gas sensor and fire sensor are used to detect the ammonia and any fire in the farm if ammonia or any fire accident in the farm by using SMS we get alert message to your mobile.

**FEATURES OF OUR PROJECT:**

* The Temperature, humidity, ammonia gas level, water level and fire are measured using sensors.
* We can manually control the fans, lights, water pump etc.
* The Ammonia levels and fire sensor values increases its send alert msg to the mobile using fast2msg services.
* If the Temperature humidity values increases automatically the exhaust and blowers should on automatically
* We can also store the data in the cloudant db.
* All these parameters can be monitored by both Mobile App and Web App.
* And we can control all the operation using both Web application and mobile app.

**THEORITICAL ANALYSIS**

**Block Diagram:**

****

**Hardware/Software Designing:**

**Software Designing**

* Python code for taking initial values for the temperature, humidity, ammonia gas level, water level, fire sensor.
* We also used manual fan on, fan off, lights on, lights off, water pump on, water pump off.
* Designing of nodes for web application in node-red service
* Designing blocks in MIT app inventor for android application
* Storing the data in cloudant service
* Using fast2sms services to send alert messages to our phone by URL

**EXPERIMENTAL INVESTIGATION:**

The project is smart animal farm and we observe the environment in the farm by various sensor and by observing the data we do the necessary action are done in the farm we do the software version of the project so we use the python code for the taking the sensor values are default values are used in the code and the data is connected to the ibm could and by sing the node-red services we develop the application by using ibmiot input node we takes the data the from the python code and by using function nodes we can separates the parameters and gauge node is used to display parameters in the web application and all the function are connected to the msg.payload for the display in the debug and button are used for operation like we want light on light off etc and whatever the operations are should be done the buttons are connected to the ibmout node and the msg.payload and we can copy the link up to bot net and add ui to it’s the Dashboard(web application )we can observe and do the operations and the switch node is connected to the temperature function node if the temp rises automatically the fans should be on and ammonia and fire sensor values increases http request node used to send the msg to the mobile that the ammonia levels are high and the fire accident in the farm and the sensor data is connected to the ibminput node to store the data and the http nodes are used to transmit the data to the mobile app by using MIT app inventor we develop the mobile application we do the layout design and the blocks are used to do the receive the sensor data , and display in the corresponding label and the function of the buttons we use url to do the all the things when we click the button the url hits and output and operation should be done.

**Python (source code)**

import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

import json

organization = "a3m5xp"

deviceType = "iotdevice"

deviceId = "1001"

authMethod = "token"

authToken = "devesh1104"

t=0

h=0

a=0

w=0

f=0

def myCommandCallback(cmd):

print("Command received: %s" % cmd.data['command'])

if cmd.data['command']=='lighton':

print("LIGHT ON IS RECEIVED")

elif cmd.data['command']=='lightoff':

print("LIGHT OFF IS RECEIVED")

elif cmd.data['command']=='fanson':

print("FANS ON IS RECEIVED")

elif cmd.data['command']=='fansoff':

print("FANS OFF IS RECEIVED")

elif cmd.data['command']=='waterpumpon':

print("WATERPUMP ON IS RECEIVED")

elif cmd.data['command']=='waterpumpoff':

print("WATERPUMP OFF IS RECEIVED")

if cmd.command == "setInterval":

if 'interval' not in cmd.data:

print("Error - command is missing required information: 'interval'")

else:

interval = cmd.data['interval']

elif cmd.command == "print":

if 'message' not in cmd.data:

print("Error - command is missing required information: 'message'")

else:

print(cmd.data['message'])

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=32

h=56

a=30

w=70

f=20

data = {"d":{ 'humidity': h,'temperature': t,'ammonialevel': a,'waterlevel': w,'firesensor': f}}#'humidity': h,'temperature': t,

#print data

def myOnPublishCallback():

print ("Published humidity = %s %%" %h, "temperature = %s %%" %t, "ammonialevel = %s %%" %a, "waterlevel = %s %%" %w, "firesensor = %s %%" %f, "to IBM Watson")#humidity = %s %%" %h, "temperature = %s %%" %t, "

success = deviceCli.publishEvent("Data", "json", data, qos=0, on\_publish=myOnPublishCallback)

if not success:

print("Not connected to IoTF")

time.sleep(1)

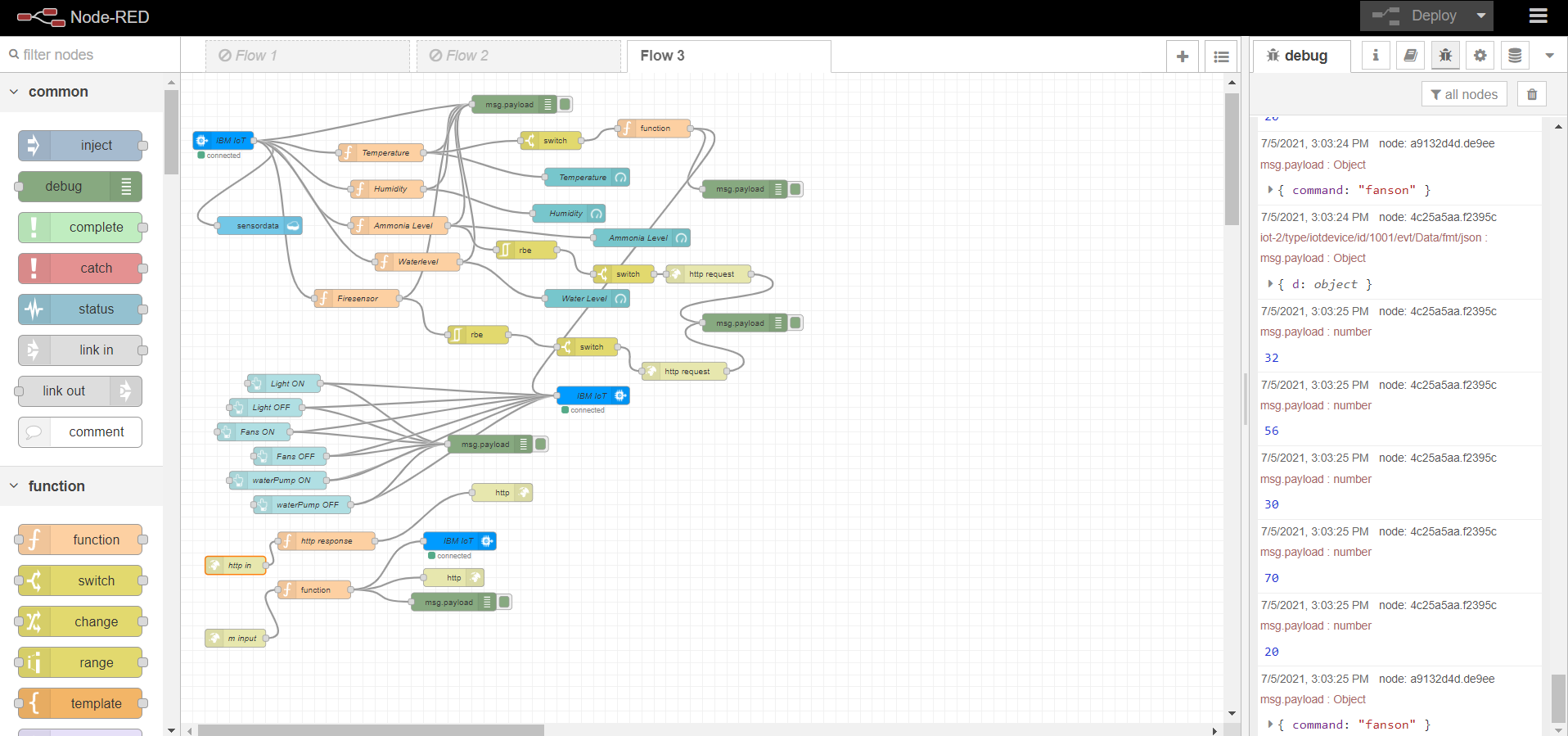
deviceCli.commandCallback = myCommandCallback

deviceCli.disconnect()

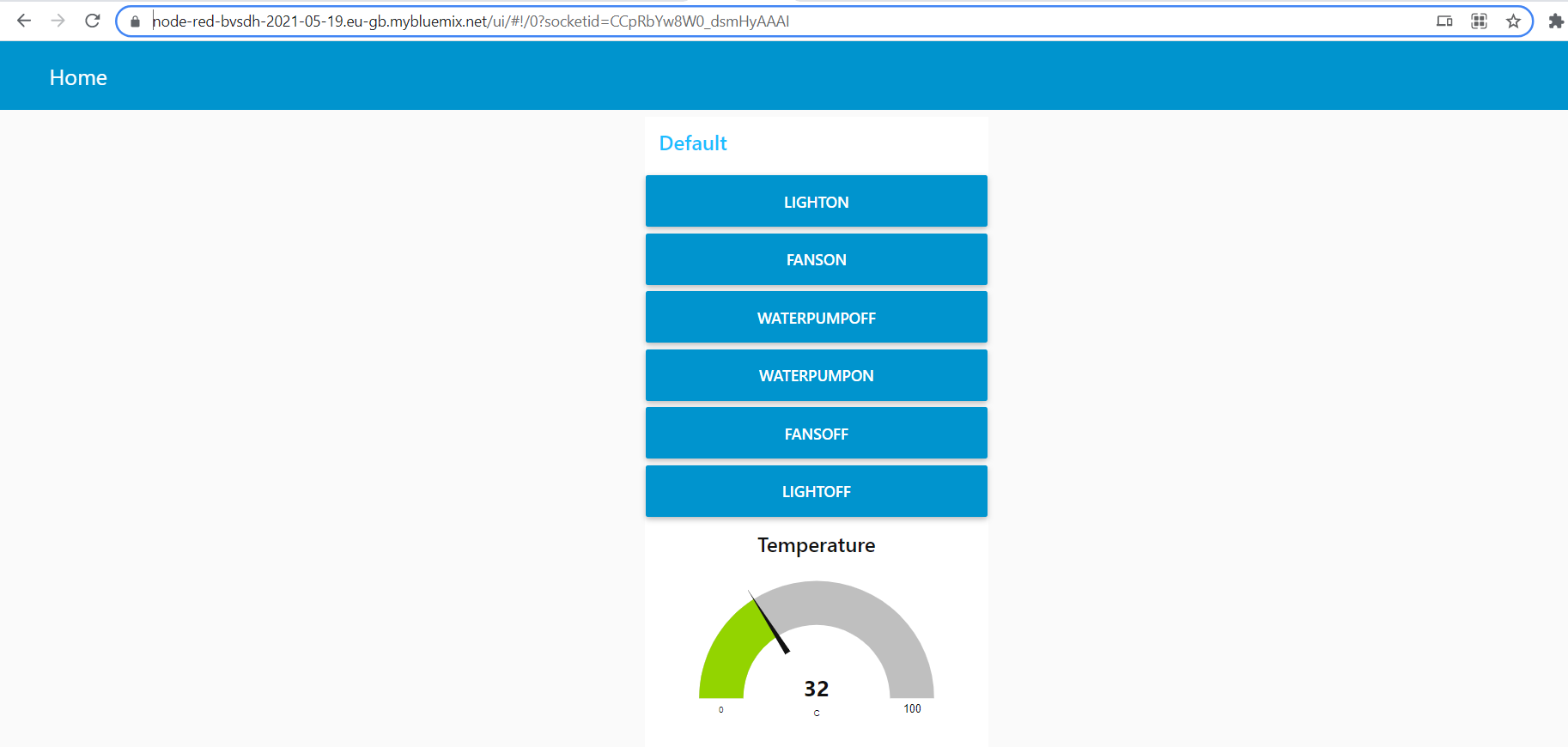
output of python code

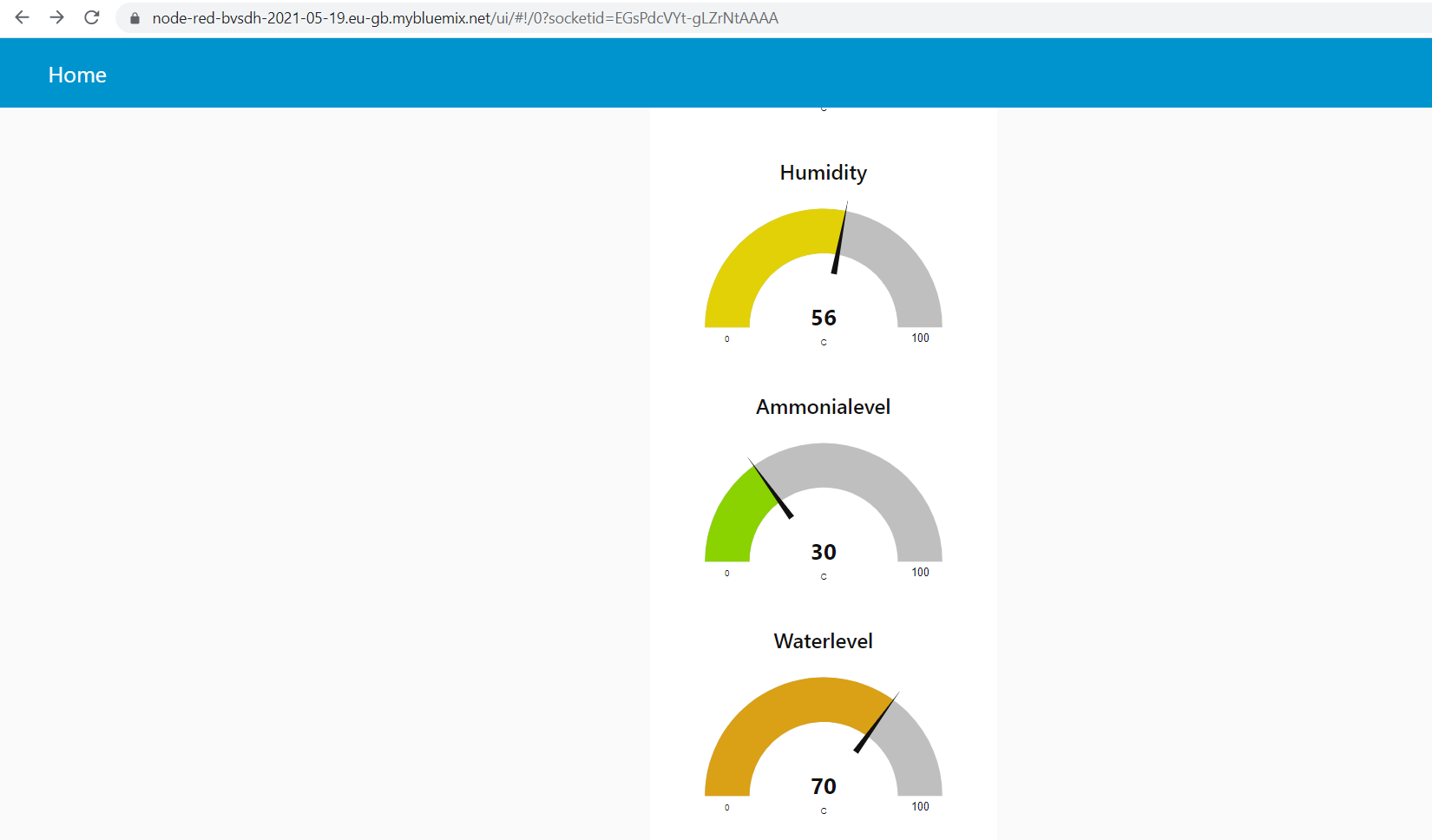


**Designing of node-red flow to visualize sensor data in web application**

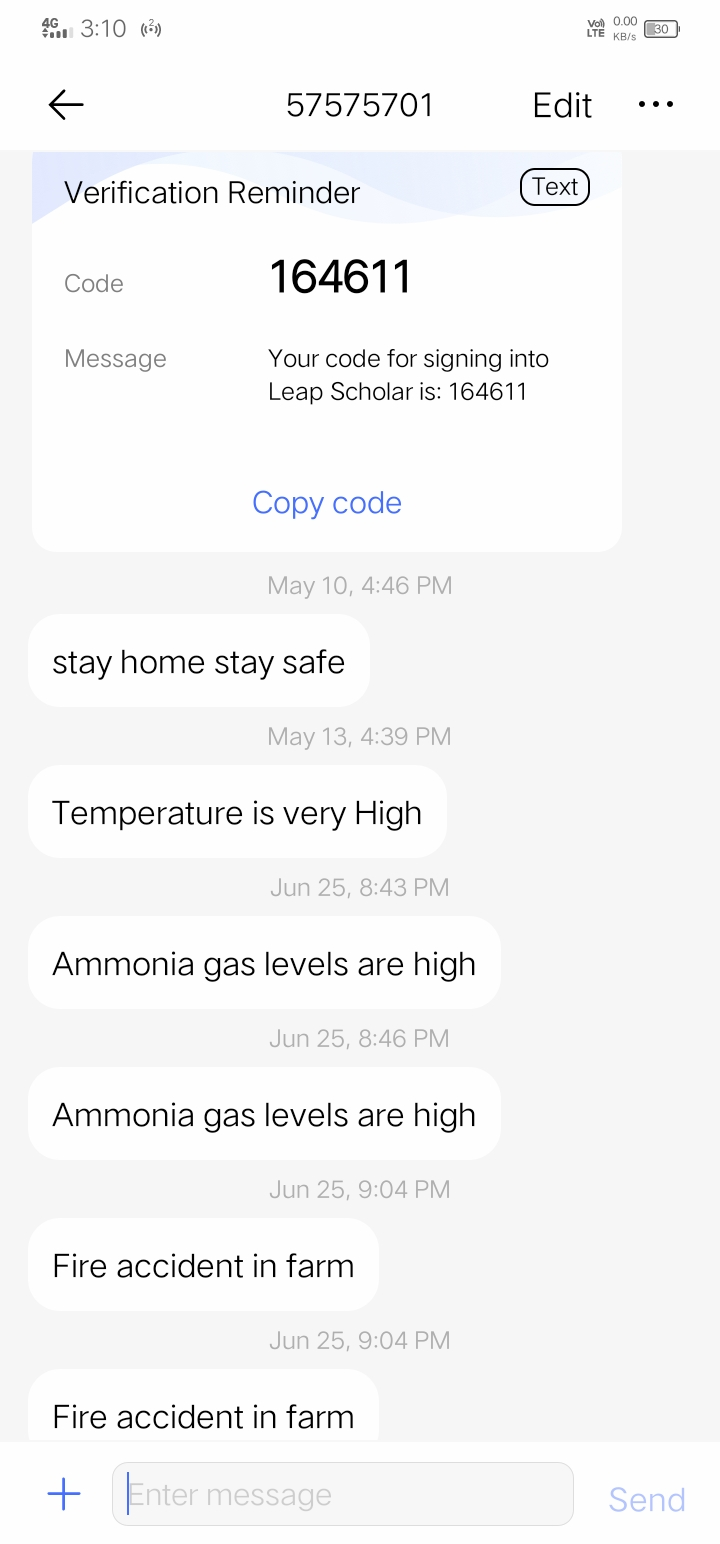


**Visualizing sensor data in web (ui) User Interface:**



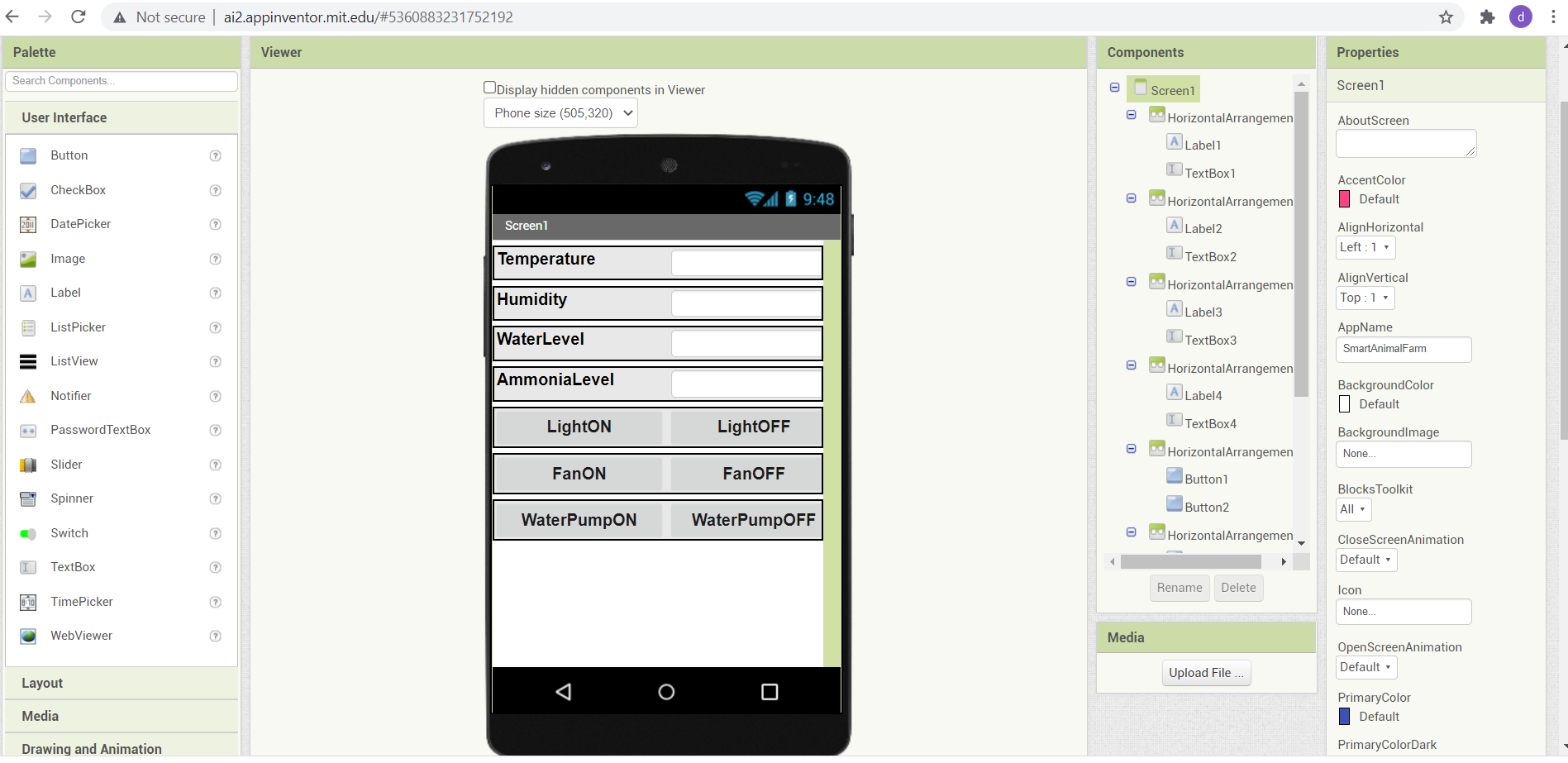


**RECEIVING ALERT MESSAGES BY UING FAST2SMS SERVICES**

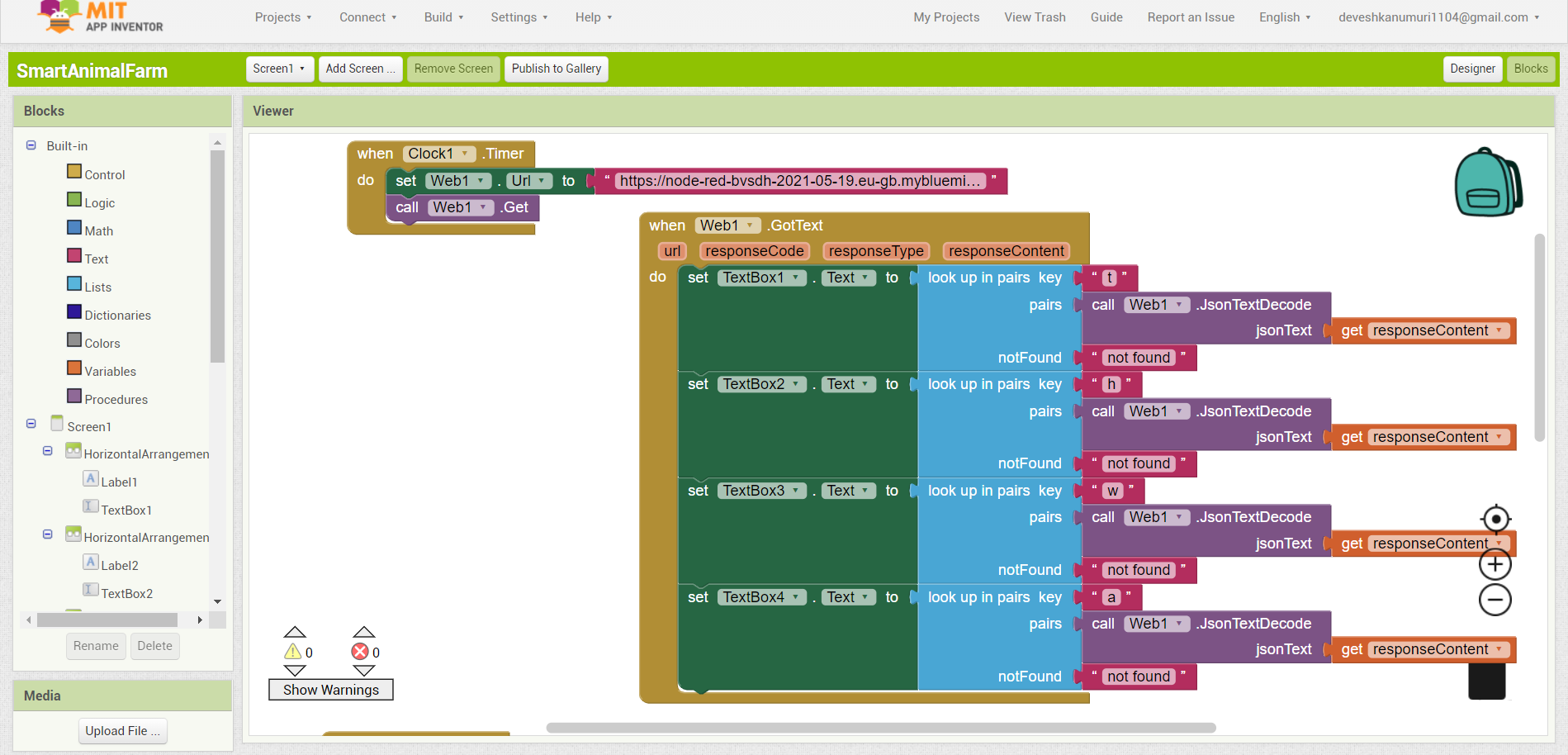


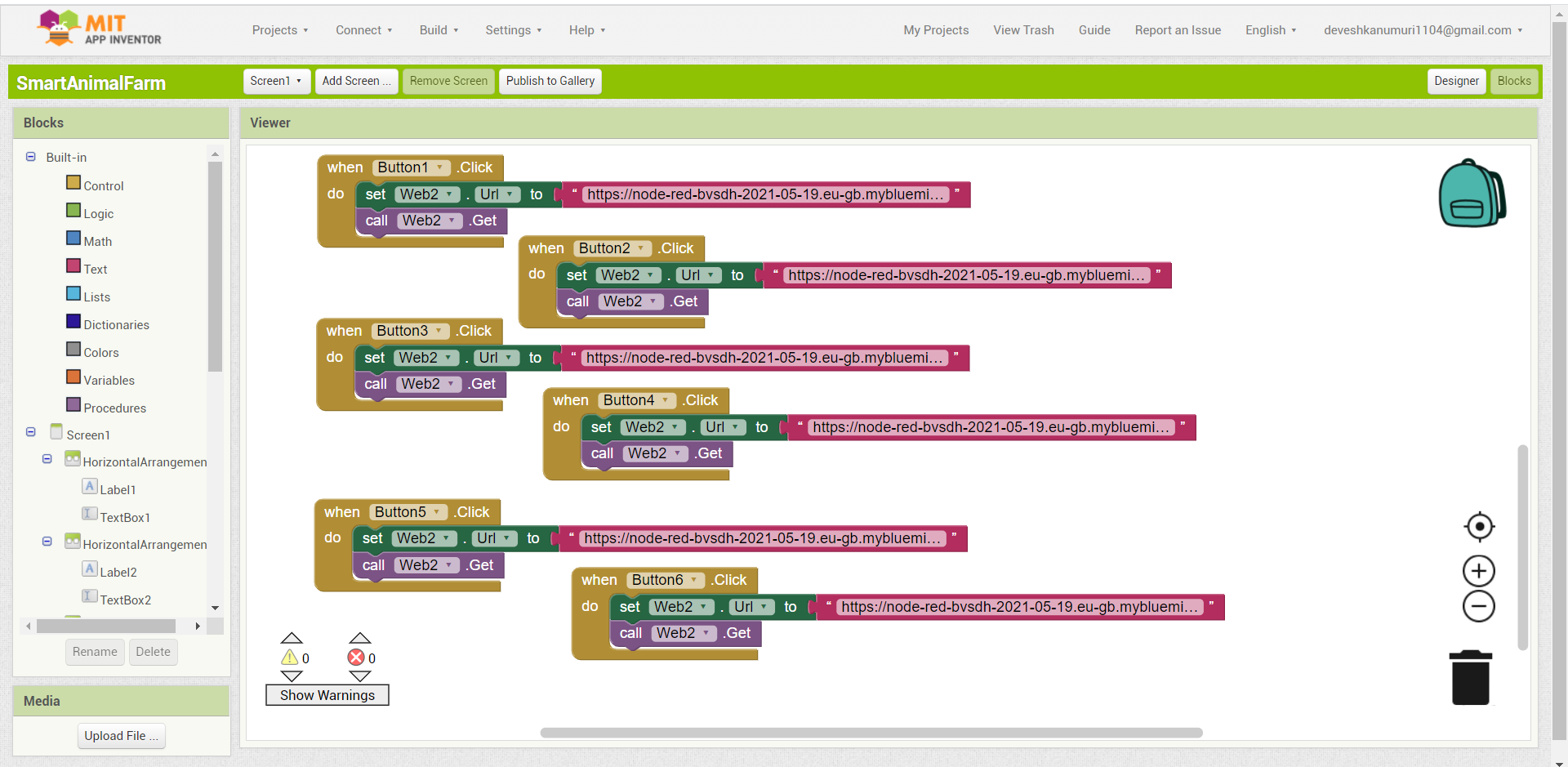
Designing app interface for the displaying the sensor data:

BY using MITapp inventer here we create an apk file by we receive data from node red payload

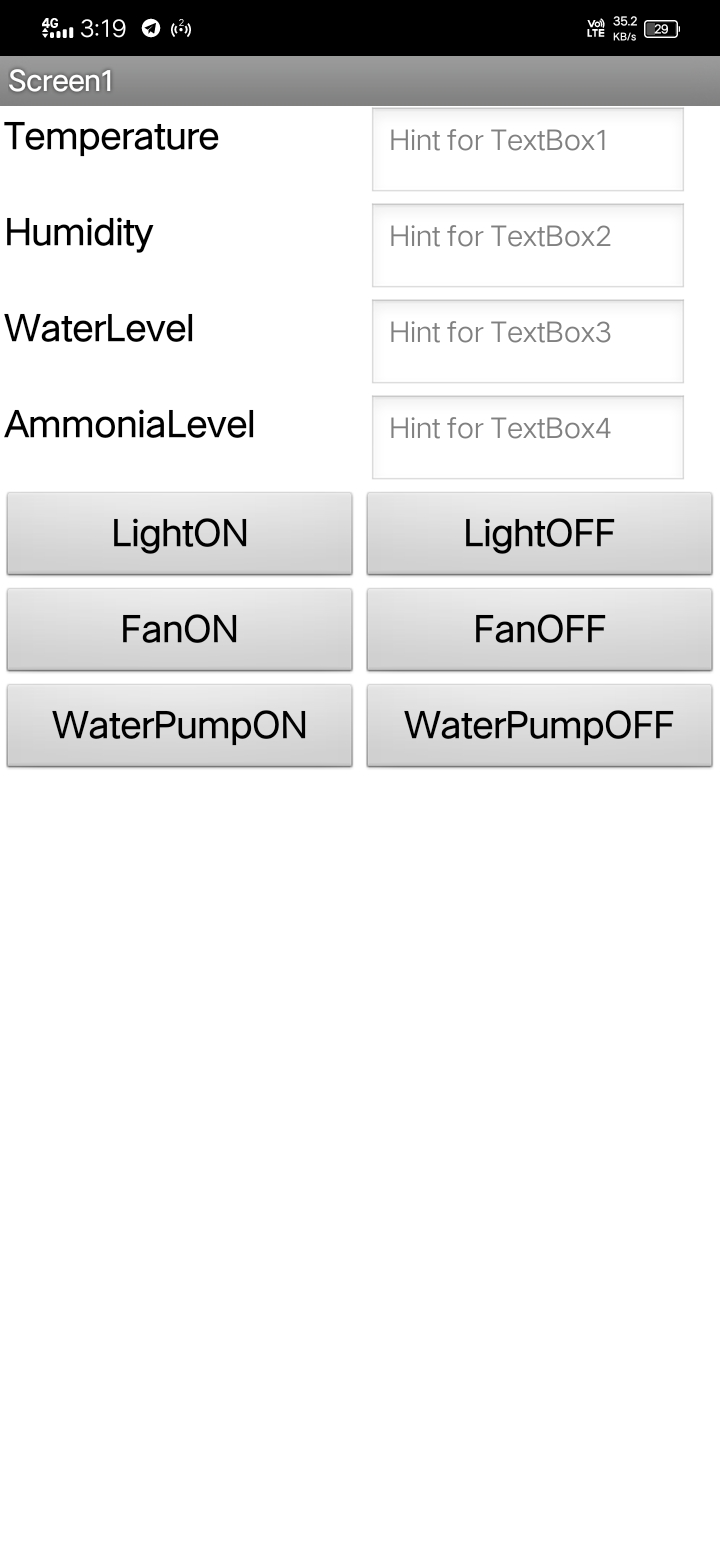


**Designing blocks in mit app inventor:**

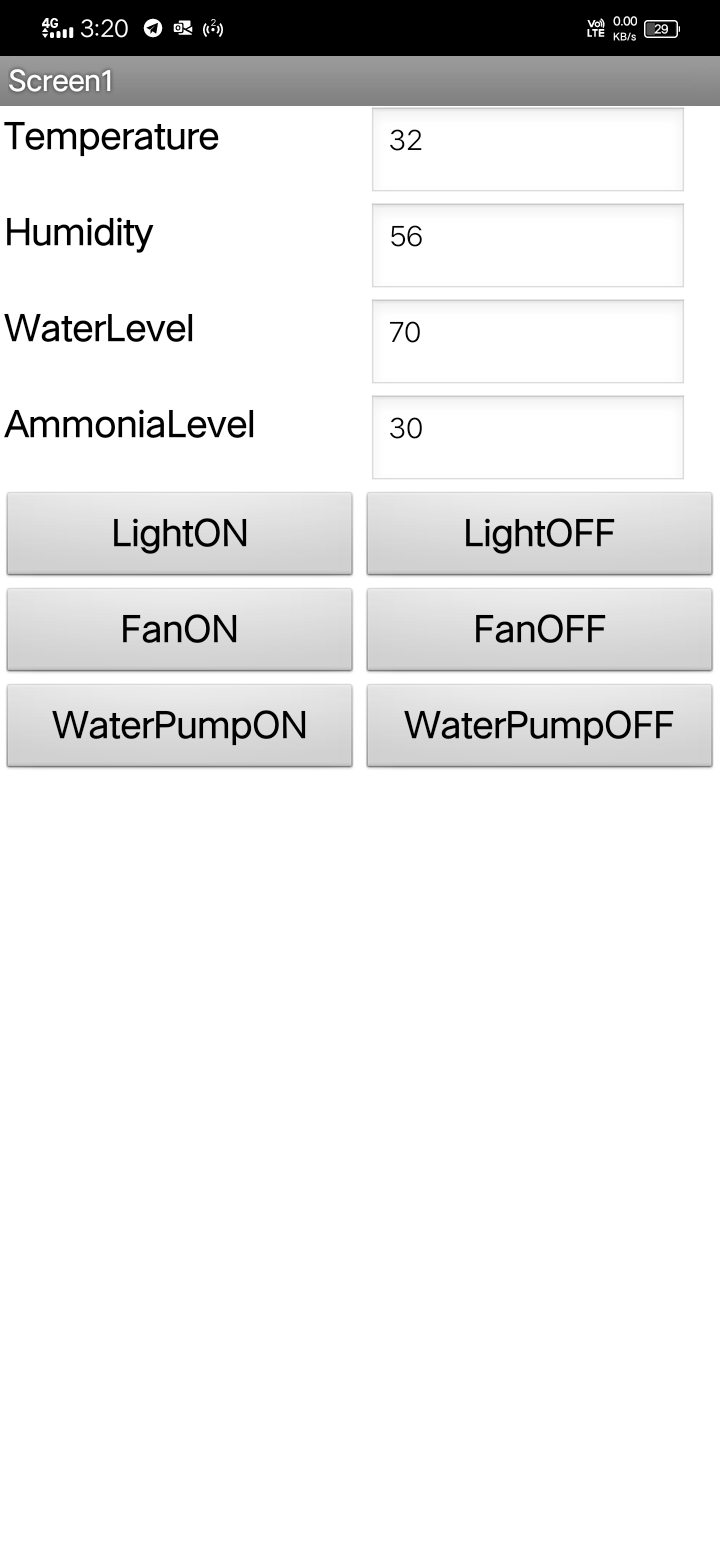




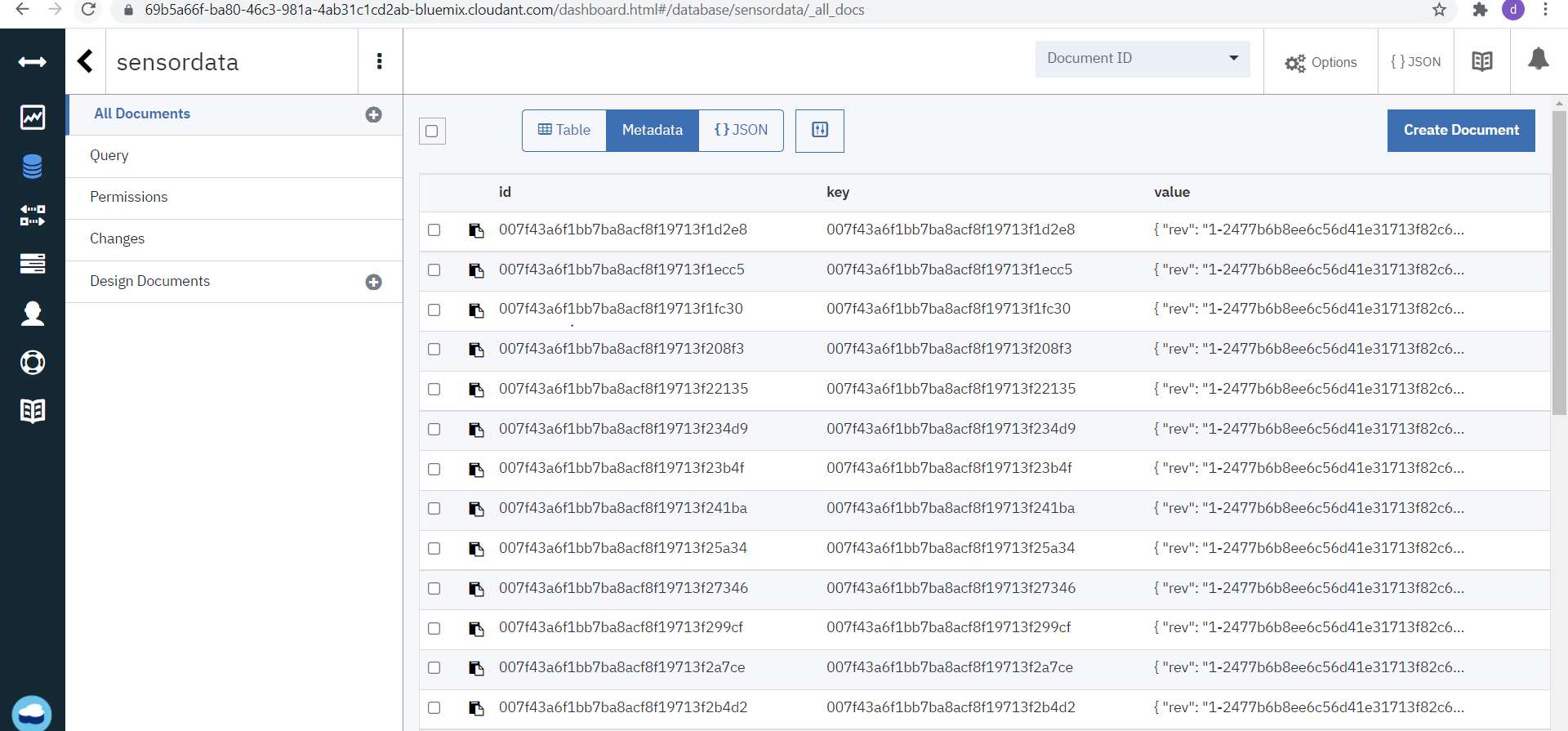
**Before hitting the url:**

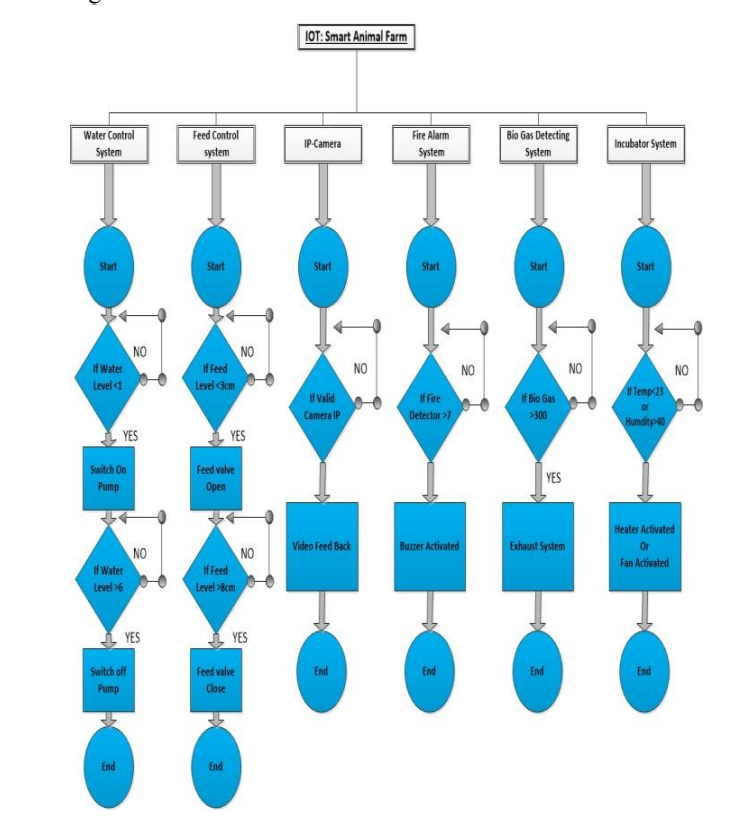


**After hitting the url:**



**Cloudant storage:(meta data)**



**FLOWCHART:**

**Result:**

The sensor data is received by the ibmplatform and the data is splited and visualized in ui interface (web application) and the app is designed to get the sensor values when we hit the url.

**Applications:**

* If the temperature and humidity of the incubator is less than a threshold value, the system turns on heater. Otherwise, fan is used to maintain the humidity.
* If the ammonia gas level is less than threshold value it will not turn on exhaust system. If it becomes more than a threshold value it will start an exhaust system.
* If the fire sensor value goes below the threshold value, alert msg using the fast2msg services to the mobile.
* Finally, if the water level becomes below the threshold value, the system turns on the water pump.

**CONCLUSION:**

IoT enabled smart animal farm. It continuously monitors the physical parameters of an animal farm. It can be controlled manually as well as automatically. This kind of system is suitable for any kind of animal farm with little modifications

**FUTURE WORK:**

One of the modifications is to provide the system with the cameras we should able to monitor what happening in the farm and we can also look into other parameters. Apply various techniques to make the system more secure. Also, we can increase some sensors.

**BIBLOGRAPY:**

<http://www.ni.com/white-paper/2983/en>

<http://www.ni.com/tutorial/4950/en/>

<http://www.libelium.com/>

<https://www.researchgate.net/publication/311223405_Internet_of_Things_IoT_Enabled_Smart_Animal_Farm>

<https://ieeexplore.ieee.org/document/7724630>

Appendix:

**UI output screenshot:**

