AUTOMATIC PET FEEDER

A Project Report
Submitted as part of the SmartBridge Externship
Internet of things (Iot)
by IBM

Submitted By :

Akshara M
Sri balaji J
Vimal R S

Introduction

A. Overview

Automated pet feeder is one of the newest technologies for feeding pet. It will help pet proprietor to take care of their pet while they are away. Even if they are not at home, they still can feed their pet. Automated pet feeder is built to help pet owner to take care of their pet. Automated pet feeder is one of the pet feeders that will be controlled by a wireless control.The automated pet feeder will automatically dispense predetermined amounts of food at the exact time user chooses through a wireless control. As pet lovers, users should understand those pets also need a proper diet management. Sometimes, the pet owners could be busy in their daily work but still can properly care for their pets. If user is away from home unexpectedly,user can feel secure that the beloved pet will be cared for and fed on time, every time.Pet care should be fun, not burdensome and so the goal of this project is to assist owner with pet care by providing an automatic pet feeder. The purpose of the project helps the owner of the pet feeding their pet on time even when they are not at home. Other than that, it also can help the owner know the diet of their pet. Knowing the diet of the pet is very important for the owner to make sure that the pet is in good health. This system assist pet owner to feed the pet. The system act in two ways, one is feeding the pet and sends the feeding information to owner. And the other is owner can change the dispenser time according to the need. This project uses speech to text and text to speech converter which facilitates the user to inform the pet to eat properly at anytime.

B. Purpose

Due to concurrent tasks demanding owner's attention, couple with busy life style, management of these pets may not be as simple as expected. Hence, the need to migrate from manual to technology-based management of pet's daily needs.An Internet of Things (IoT) based automatic feeder system comes handy to assist in the management of pets needs. The latter technology will enable pet owners to remotely manage critical needs that are automatable while engaged in other time and attention demanding tasks. This project works with a fixed time to open and close the dispenser, it also provides a facility to the user to open or close the dispenser at customized time. The user can also provide his/her message as text or voice that will be converted to speech at the iot device installed in the home.

Literature Survey

1. R. V. Sweeney in “Time controlled Feeding device for domestic pets” had proposed: automatic feeder, the Kenl-Master: a covered food plate for dogs which pops its lid at feeding time if you remember to set its alarm-clock timer.
2. Jerome Frankel in “Kum pet feeder: Feeding device for Animals” has shown: a wind-up alarm clock, mounted flush into a galvanized metal frame, is set with the time for the pet’s meal. When mealtime is reached, the tension unwinding spring that operates the clock’s alarm also spins a metal spool that winds up a string. The string is attached to the underside of the metal feeding dish under the cover, kept in place by a cut-out in the metal base.
3. Krishnamurthy S. in “Automatic pet feeder” has proposed: The pet feeder comprises a base, a feeding bowl with pie shaped divisions, a timer module, a bowl cover, handle to bowl cover and locking mechanism to hold the entire unit in place. This invention relates to an integrated automatic device for training and feeding pet which also functions as a playmate while the owner is absent or otherwise engaged.

A. Existing problem

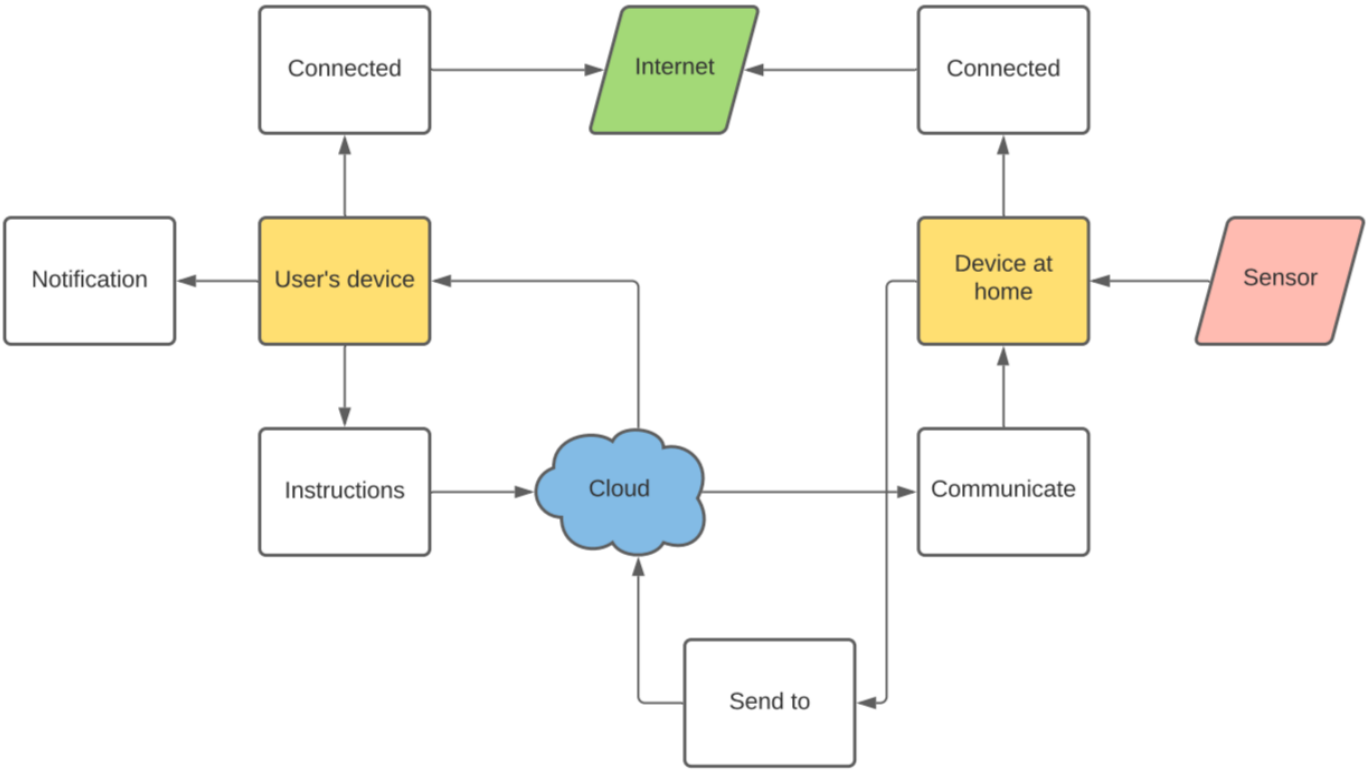
Analyzing the previous cases we can conclude that in the recent scenarios, the owners of the pet find it difficult to feed the pets at required time and at times it so happens that the owner gets busy or gets held up in some work hence he/she needs to address the need of feeding pets remotely. They also expect their messages to be said via device that is incorporated within the home where the dog resides.

B. Proposed solution

The proposed solution for the above problem can be addressed by linking the device incorporated within the house and the device that the user is using via internet based cloud communication. The user's device is given a facility to send his/her message to their pet using text or voice input via cloud this message is sent to the device that is installed at home. The user can also manually set the time for opening and closing the dispenser. The food and water levels are continuously monitored and are sent as notifications to the user's device. The user can also view the status for the water and food dispenser in the app.

Theoretical Analysis

A. Block diagram



Both the user's device and device at home are connected to internet. When the user's device gives instructions it passes through the cloud and the instructions from the cloud are communicated to the device at home as voice. The sensors at home take values periodically and are sent to the device at home as sensor data and are sent to the cloud, from the cloud it is sent to user's device as notification.

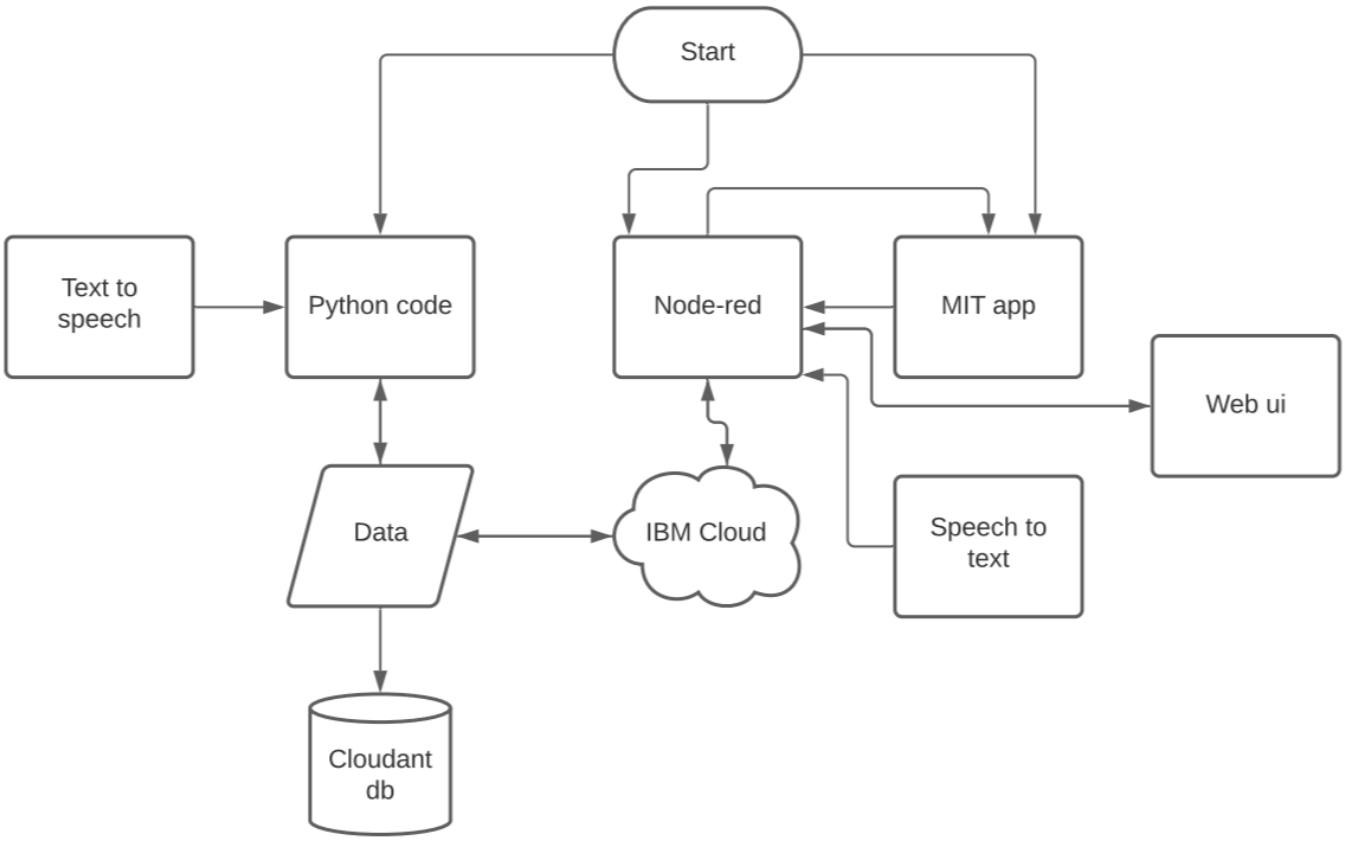
B. Hardware / Software designing

In this project we have used python code to generate random sensor values that is water and food level. From the python code these sensor values are sent to user's mobile that works on MIT app inventor. The user using MIT app has facility to open or close the dispenser using manual options when he gives instructions via MIT app it is processed in the cloud. From the cloud these instructions are processed as voice message at the device that is installed at home. All these cloud connections are done in the node-red. This node-red provides different dashboard nodes to view the interface in the web ui. The cloud we are using in this project is IBM cloud. The water and food level are recorded and stored in the cloudant database.

Experimental Investigations

We are using software to generate sensor data to show if user can view the data on his/her mobile screen. The conclusion of this investigation is that we are able to successfully generate output on the mobile interface and we are also able to send text message or voice message back to the device at home. As there is no sensor to open or close we are printing the operations on to the python code. This test set that we are using in this project works successfully on all operations from both user side and device connected at home.

Flowchart



Initially we need to run python code, deploy node red and open MIT app. The text to speech service is incorporated with in the python code. The speech to text service is given to the node-red application. The MIT app enables user to record their message as text or voice which is processed in the node-red and sent to python code where the received text is convert to speech. This speech is actually the audio that is received at the user's home. There is a web user interface that is connected to node-red through which the user can also give the input. Node-red and python code are connected to the IBM cloud. The data from python are stored in the cloudant database.

Result

After Performing the connections to python code, node-red, MIT app the randomly generated values of the water and food level can be viewed on the MIT app at the user's end. Similarly users can open or close the dispenser manually through the buttons that are given as the user interface on the user's device. These functions are converted as text to speech at the device which installed at home. This enables two modes of interactions that is automate the work of dispensers and the other one is manually the user can remotely operate the dispensers. As the data is stored as a json data it can be easily retrieved from the cloudant database that is used in order to record the data that is received from the sensors. The dashboard nodes enabled in the node-red allows the user to even use the web ui to operate the dispensers. The users can even set the timing for opening the dispensers.

Advantages & Disadvantages

Advantages

- Storing extra food in the additional reservoir, keeping it safe from pests as well as an uncontrolled appetite.
- Allowing for predictable mealtimes even if you cannot be home to feed your pet – ideal for long work days or traveling.
- Providing proportioned meals more frequently, essential for helping pets managing health conditions.
- As it is connected to the internet notifications are sent to the user's device and user can also control the dispenser remotely.
- If in case the owner gets busy the automatic settings would open and close the dispenser.
- If the user wants to leave any message to the pet then they can either send it as text or voice which will be converted back to voice that will be given out by the device that is installed at the home.

Disadvantages

- Generally lacking sensors and will continue dispensing pre programmed meals even if there is leftover, uneaten food.
- No way to monitor which pet is eating which meal if multiple pets are sharing the feeding area.
- Dispensing action can be noisy and may be disruptive to sensitive pets,

keeping them from feeding even when hungry.

- Generally works best only with dry food and kibbles of certain sizes, so they may not be versatile enough for every pet's diet.
- Cleaning can be difficult in more complicated designs and the unit may need disassembly to thoroughly clean moving parts.
- Feeder will not function if the batteries wear down or power is lost, which can mean missed meals and disrupted feeding.
- The voice that is heard by the pet frequently may irritate the pet and it may loose interest to have its meal at particular time.
- If there is an error with connection to internet then the automatic feeder system will not open or close the dispenser which may leave the pet to either starve or overeat.

Applications

An IoT-based feeder system should be able to automate feeding and other related provisions or needs of pets. Such IoT-based feeder system can be designed in a way that it dispenses precise amount of food or other provisions at specific time intervals, reduce the amount of time owners spend on feeding and monitoring of household pets. In addition to the relief the automated feeder system gives to pet owners, it can be programmed in such a way that it can be controlled with the push of a button or remotely through voice commands and via a web application with a good user friendly interface. Asides the benefits automatic pet feeders give its users, it can also regulates the amount of food given to pets since it can be programmed to dispense specific amount of food, thereby ensuring pets are not malnourished or overfed which may lead to obesity especially when the pets are still very young. The applications of pet feeder deals with the use of motor to open or close the dispenser whose operation is controlled by the python code. Different sensors are used for the detection of particular criteria which makes the iot device work properly. Currently, there are lots of pet feeding devices in the market, aimed at ensuring that pets get a healthy amount of feeds even when the pet owner is away. The major difference in the various pet feeding devices is the degree of control that these devices could give to pet owners and the methodology used to achieve it. Hence, this study proposes a pet feeder system that is IoT driven.

Conclusion

The proposed solution, with the aid of a user friendly interface and the integration of IoT into pet feeder device, would allow users / owners control the pet feeder remotely. Also, it will enable owners monitor the pet to ensure that proper feeding is always carried out. The proposed design would give pet owners the freedom to travel knowing that with the automatic pet feeder, the pet would be well taken care of. This design of an IoT based automatic pet feeder system was done in consideration of some factors such as: economic application, user convenience, availability of components and research materials, efficiency, compatibility, portability and durability. As earlier stated, this work aims to enhance the management of pets, giving their owners greater flexibility in the provision of essential care and nutritional and medical needs,despite their multiple time and attention demanding tasks and busy schedules. The prototype and subsequent evaluation,however, indicates that the research goal is feasible and achievable. Thus, current work extends previous efforts in the management of household pets.

Future Scope

Given the big demand there is a huge scope of innovation both in terms of improving the hardware and software aspects. Further innovations would comprise of a product in which all the features are included in a modernized machine with low maintenance. The old models still require remote control but can be easily automated thus completely removing human intervention. More upgrades can be made to the model by improving sanity levels due to constant food storage, the food may get spoiled, so sensors can be added to alert the owner about change of food also the material of the dispenser can be chosen accordingly. This technology can also be used with the Home Automation technologies. It can also be made more advanced such that it can be used with more sensors. This pet feeder is a must as it will make the lives of the owner easy. The owner, absent or present at home will be tension free as the pet feeder will keep the pet fed on time. This operation could be linked with the face detection module. Where first the python is trained with test set which comprise of the face of the pet based on the detection it will periodically send images of the pet to the user's device.

Bibliography

1. Wayne Intelligent Food Dispenser (Ifd) Hari N. Khatavkar, Rahul S. Kini, Suyash K. Pandey, Vaibhav V. Gijare, 2019

2. Proposed System for Animal Recognition Using Image Processing Ankur Mahanty, Ashutosh Engavle, Taha Bootwala, Prof. Ichhanchu Jaiswal,2019.

3. Digital Image Processing-A Quick Review R. Ravikumar, Dr V. Arulmozhi,2019.

4. Pet Feeding Dispenser Using Arduino And Gsm Technology Smruthi Kumar, 2018

5. Automatic Pet Feeder Aasavari Kank, Anjali Jakhariye, 2018

6. A Remote Pet Feeder Control System Via Mqtt Protocol Wen-Chuan Wu, Ke-Chung Cheng, Peiyu Lin, 2018

7. Iot Based Pet Feeder System Saurabh A. Yadav, Sneha S. Kulkarni, Ashwini S. Jadhav, Prof. Akshay R. Jain,2018

8. Simulation Of Automatic Food Feeding System For Pet Animals Dharanidharan.J, R.Puviarasi, 2018

9. Convolutional Neural Network (CNN) for Image Detection and Recognition Rahul Chauhan, Kamal Kumar Ghanshala, R.C Joshi, 2018

10. Convolutional Neural Networks for image classification Nadia Jmour, Sehla Zayen, Afef Abdelkrim, IEEE, 2018

Appendix

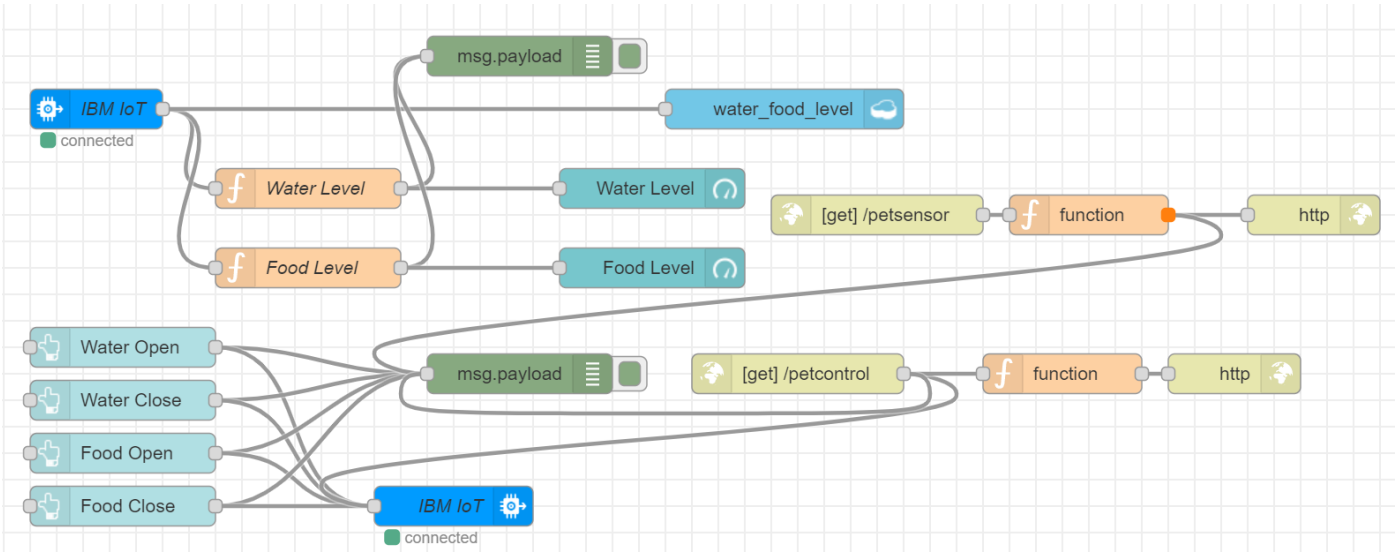
A. Source code

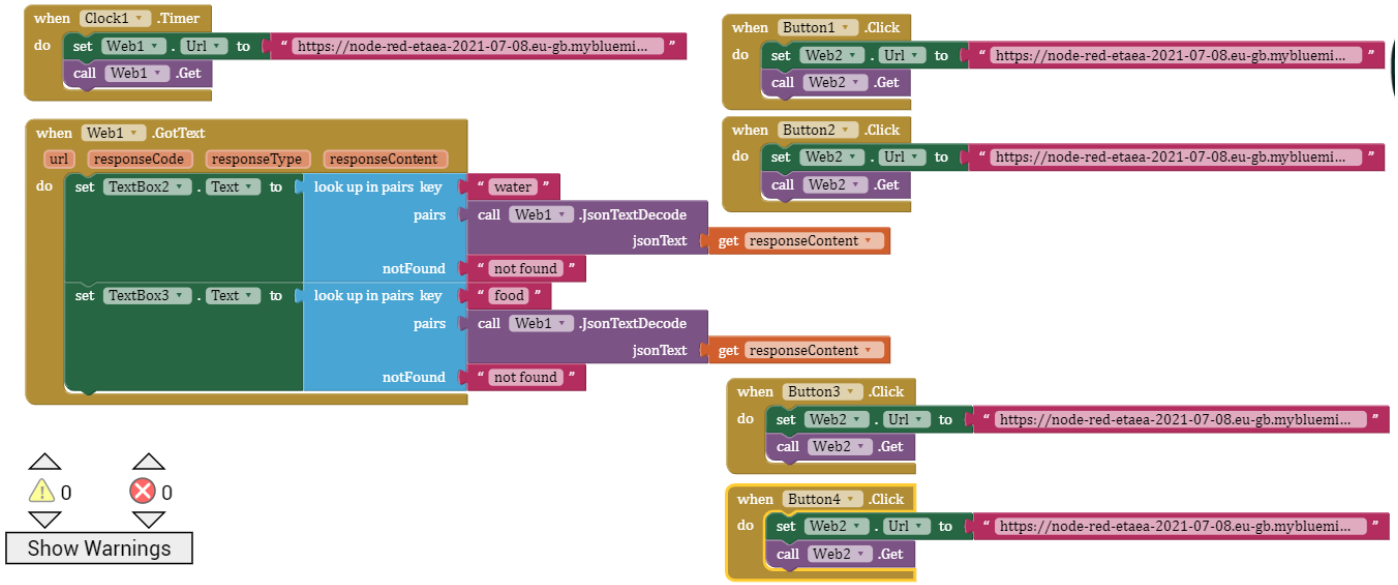
```
import wiotp.sdk.device

import time

import random

from ibm_watson import TextToSpeechV1

from ibm_cloud_sdk_core.authenticators import IAMAuthenticator

import simpleaudio as sa


authenticator = IAMAuthenticator('uQ-kJ7yvhpbFF1FhwsX_xaFeP-WSTuVqKYgOgLW3Zpfy')

text_to_speech = TextToSpeechV1(

    authenticator=authenticator

)


text_to_speech.set_service_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/73add8e6-13e5-43c5-a808-03d3eed007b7%27)


myConfig = {

    "identity": {

        "orgId": "iyqgoe",

        "typeId": "AkshuDevice",

        "deviceId":"240602"

    },

    "auth": {

        "token": "240602291205"

    }

}


def myCommandCallback(cmd):

    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
```

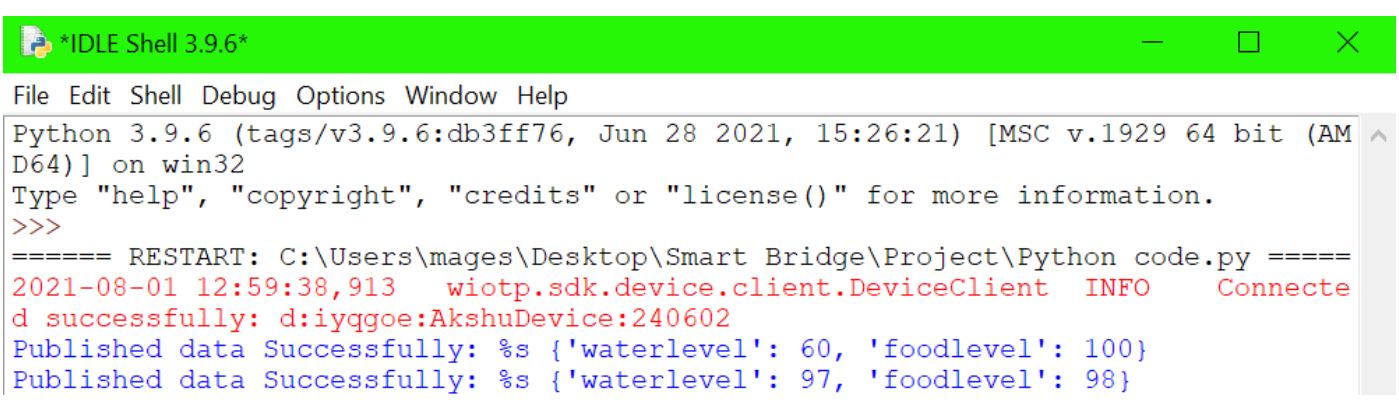
```
    m=cmd.data['command']
    if(m == "wateropen"):
        print(".......Water is OPEN.......")
    elif(m == "waterclose"):
        print(".......Water is CLOSE........")
    elif(m == "foodopen"):
        print(".......Food is OPEN........")
    elif(m == "foodclose"):
        print(".......Food is CLOSE........")
    print()
    with open('Hello.wav', 'wb') as audio_file:
            audio_file.write(
                text_to_speech.synthesize(
                    m,
                    voice='en-US_MichaelVoice',
                    accept='audio/wav'
                    ).get_result().content)
    filename = 'Hello.wav'
    wave_obj = sa.WaveObject.from_wave_file(filename)
    play_obj = wave_obj.play()
    play_obj.wait_done()

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

while True:
    waterlevel=random.randint(0,100)
    foodlevel=random.randint(0,100)
    myData={'waterlevel':waterlevel, 'foodlevel':foodlevel}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
```

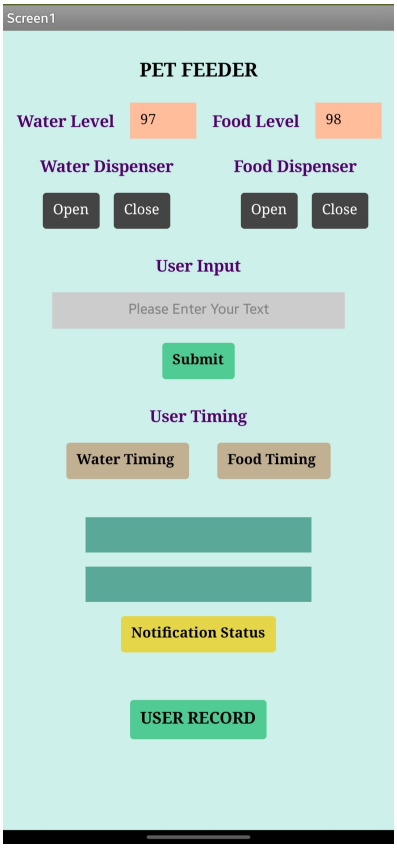
```
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

B. UI output Screenshot

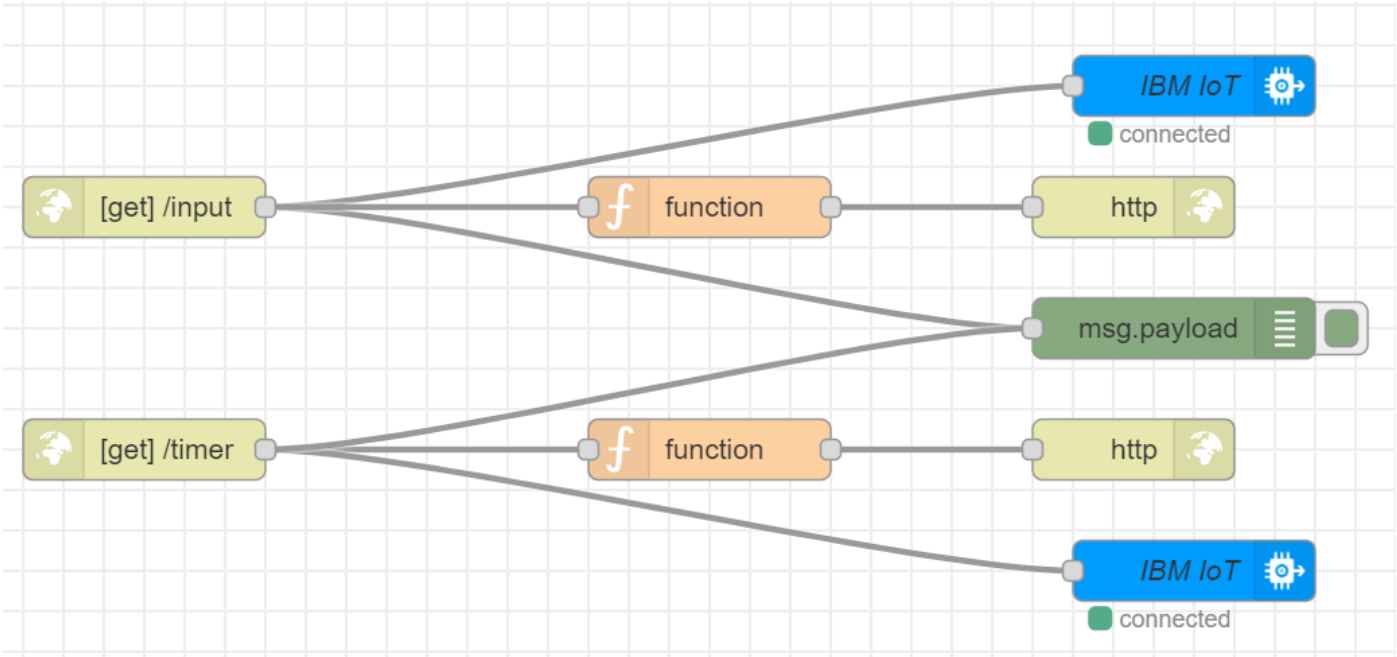


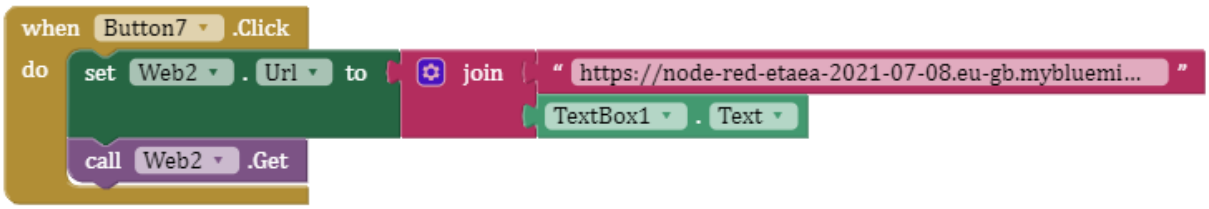


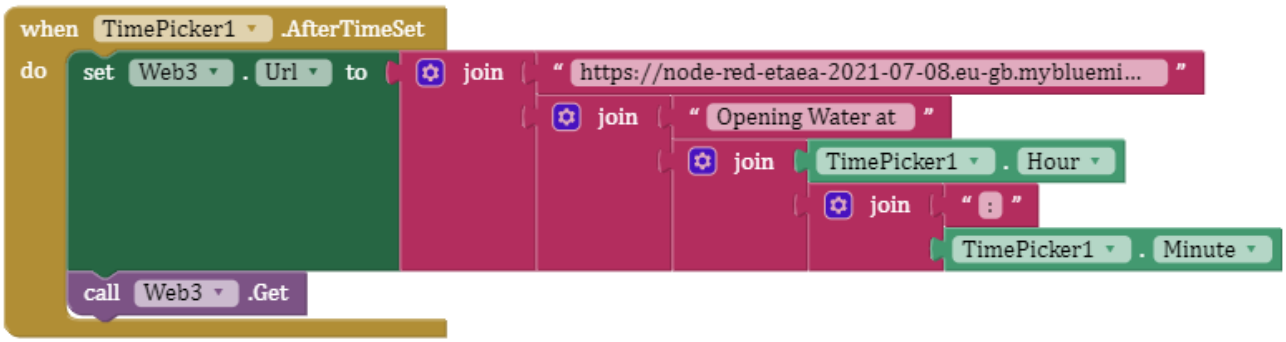


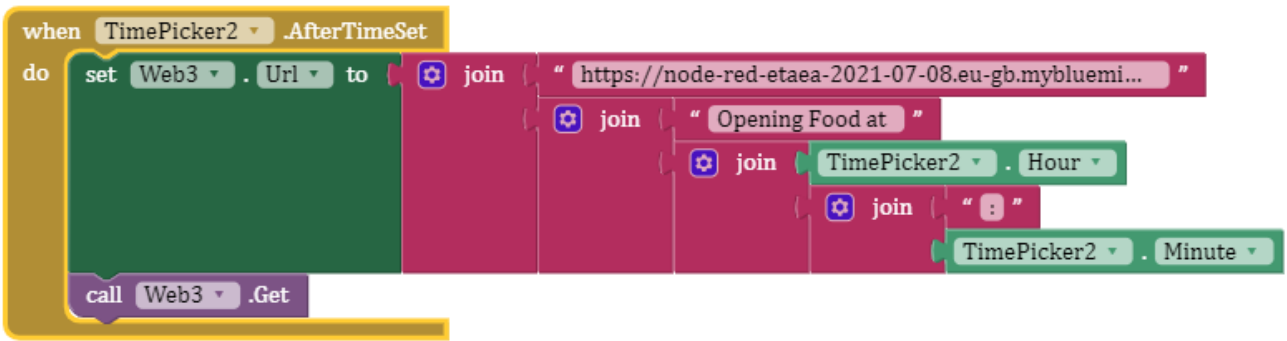


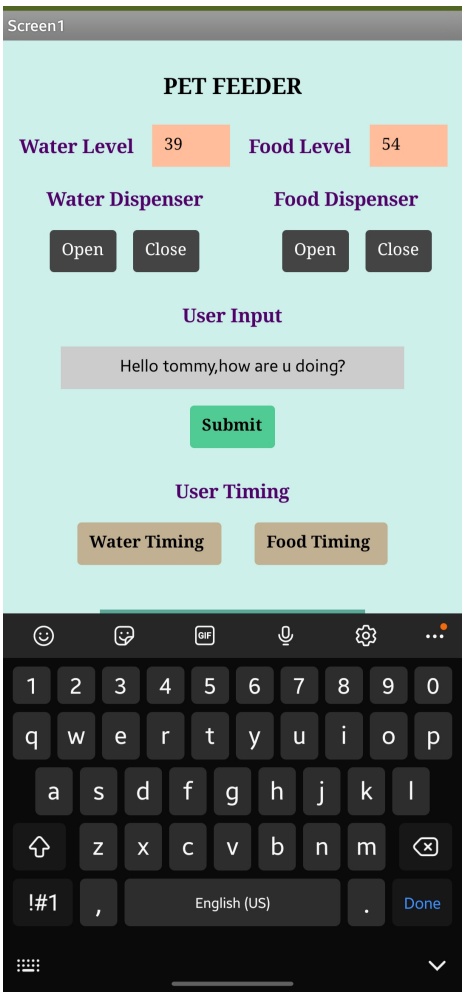

```
Published data Successfully: %s {'waterlevel': 38, 'foodlevel': 39}
Published data Successfully: %s {'waterlevel': 30, 'foodlevel': 14}
Message received from IBM IoT Platform: wateropen
.......Water is OPEN.......

Published data Successfully: %s {'waterlevel': 6, 'foodlevel': 40}
Published data Successfully: %s {'waterlevel': 3, 'foodlevel': 95}
Message received from IBM IoT Platform: waterclose
.......Water is CLOSE........

Published data Successfully: %s {'waterlevel': 49, 'foodlevel': 25}
Published data Successfully: %s {'waterlevel': 38, 'foodlevel': 7}
Message received from IBM IoT Platform: foodopen
.......Food is OPEN........

Published data Successfully: %s {'waterlevel': 2, 'foodlevel': 37}
Published data Successfully: %s {'waterlevel': 7, 'foodlevel': 20}
Message received from IBM IoT Platform: foodclose
.......Food is CLOSE........
```

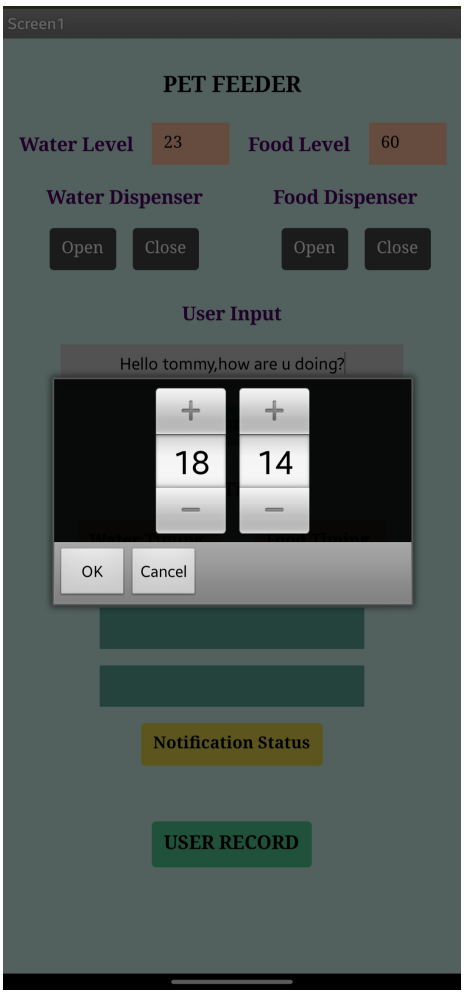

IBM IoT
connected
[get] /input
function
http
msg.payload
[get] /timer
function
http
IBM IoT
connected


when Button7 .Click
do set Web2 . Url to join " https://node-red-etaea-2021-07-08.eu-gb.mybluemi... "
TextBox1 . Text
call Web2 .Get

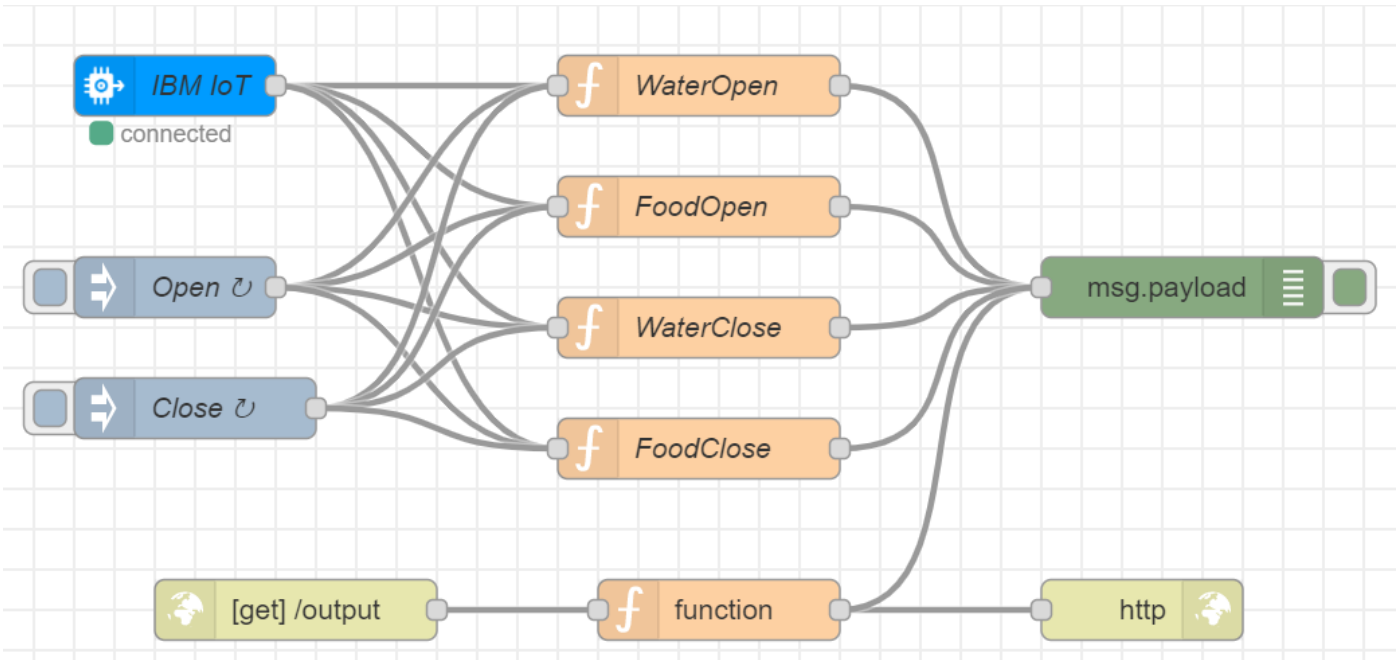

when TimePicker1 .AfterTimeSet
do set Web3 . Url to join " https://node-red-etaea-2021-07-08.eu-gb.mybluemi... "
join " Opening Water at "
join TimePicker1 . Hour
join " : "
TimePicker1 . Minute
call Web3 .Get

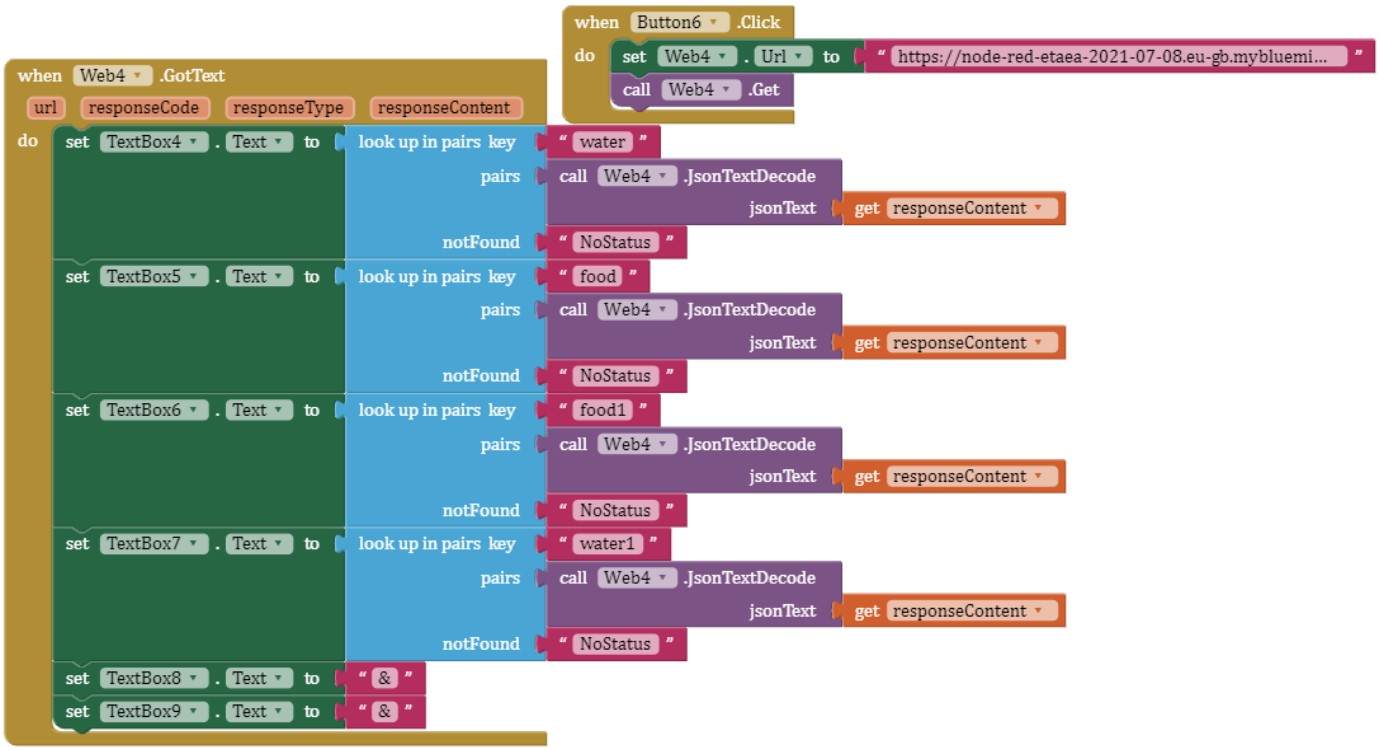

when TimePicker2 .AfterTimeSet
do set Web3 . Url to join " https://node-red-etaea-2021-07-08.eu-gb.mybluemi... "
join " Opening Food at "
join TimePicker2 . Hour
join " : "
TimePicker2 . Minute
call Web3 .Get

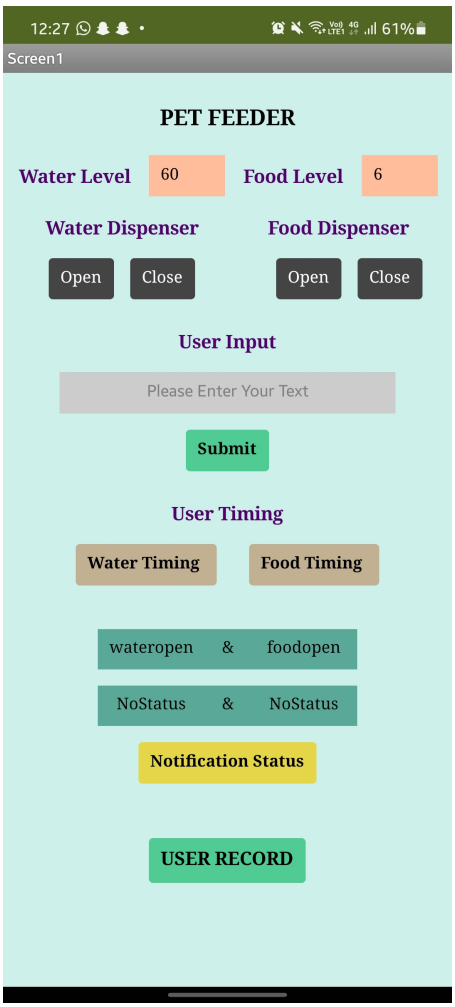


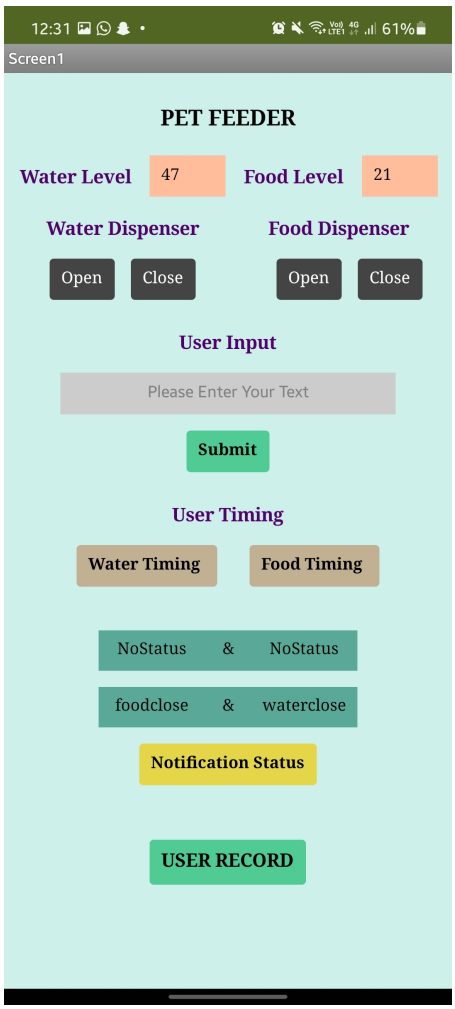


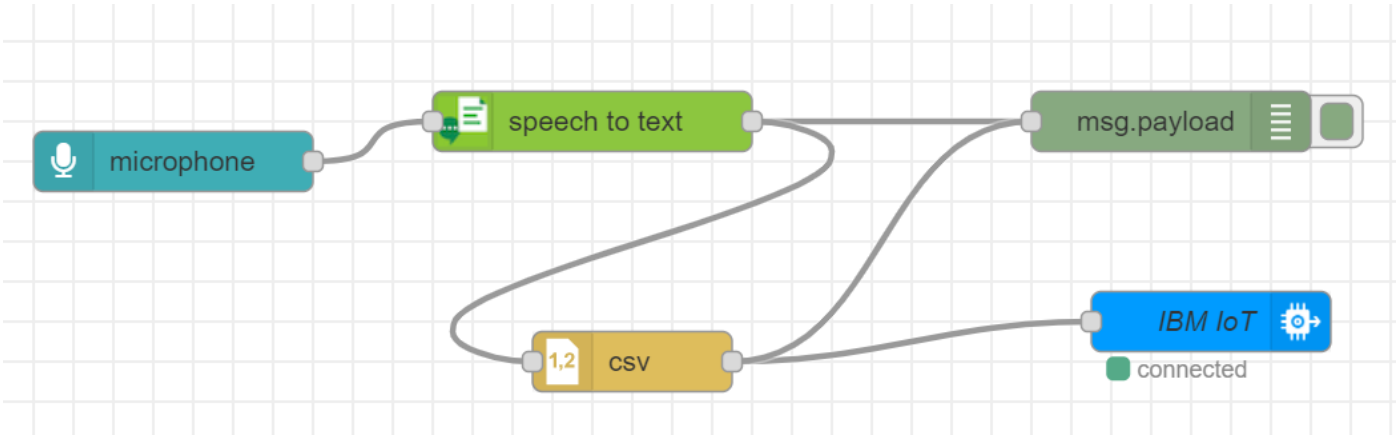

```
Published data Successfully: %s {'waterlevel': 21, 'foodlevel': 79}
Published data Successfully: %s {'waterlevel': 17, 'foodlevel': 26}
Message received from IBM IoT Platform: Hello tommy,how are u doing?
Published data Successfully: %s {'waterlevel': 32, 'foodlevel': 49}
Published data Successfully: %s {'waterlevel': 48, 'foodlevel': 50}
Message received from IBM IoT Platform: Opening Water at 13:40
```

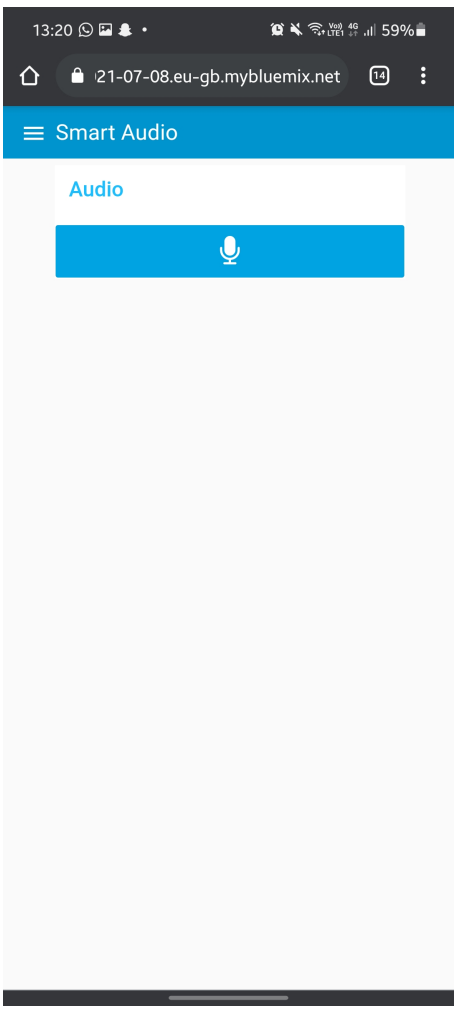



when Button6 .Click
do set Web4 . Url to " https://node-red-etaea-2021-07-08.eu-gb.mybluemi... "
call Web4 .Get
when Web4 .GotText
url responseCode responseType responseContent
do set TextBox4 . Text to look up in pairs key " water "
pairs call Web4 .JsonTextDecode
jsonText get responseContent
notFound " NoStatus "
set TextBox5 . Text to look up in pairs key " food "
pairs call Web4 .JsonTextDecode
jsonText get responseContent
notFound " NoStatus "
set TextBox6 . Text to look up in pairs key " food1 "
pairs call Web4 .JsonTextDecode
jsonText get responseContent
notFound " NoStatus "
set TextBox7 . Text to look up in pairs key " water1 "
pairs call Web4 .JsonTextDecode
jsonText get responseContent
notFound " NoStatus "
set TextBox8 . Text to " & "
set TextBox9 . Text to " & "


12:27
61%
Screen1
PET FEEDER
Water Level
60
Food Level
6
Water Dispenser
Food Dispenser
Open
Close
Open
Close
User Input
Please Enter Your Text
Submit
User Timing
Water Timing
Food Timing
wateropen & foodopen
NoStatus & NoStatus
Notification Status
USER RECORD

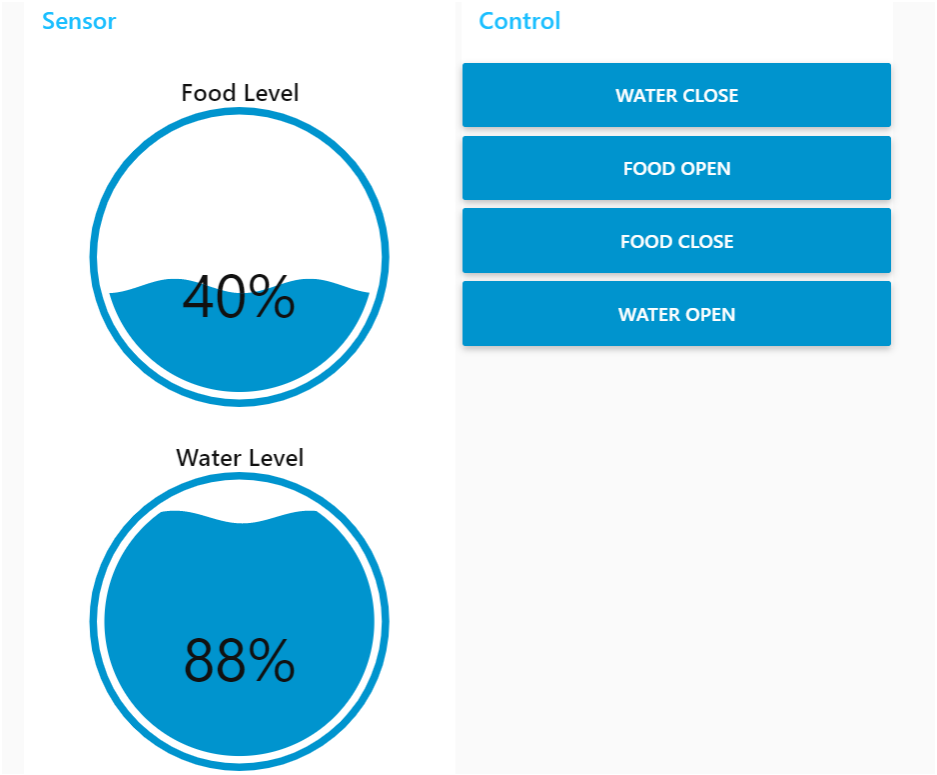

12:31
61%
Screen1
PET FEEDER
Water Level
47
Food Level
21
Water Dispenser
Food Dispenser
Open
Close
Open
Close
User Input
Please Enter Your Text
Submit
User Timing
Water Timing
Food Timing
NoStatus & NoStatus
foodclose & waterclose
Notification Status
USER RECORD

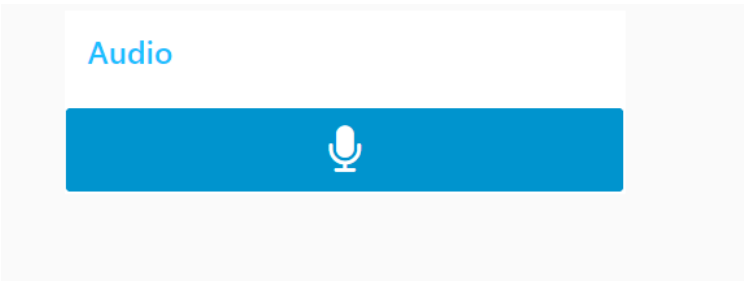





```
Published data Successfully: %s {'waterlevel': 82, 'foodlevel': 90}
Published data Successfully: %s {'waterlevel': 27, 'foodlevel': 68}
Message received from IBM IoT Platform: tell me what are you doing
```

Web UI


Sensor
Food Level
40%
Water Level
88%
Control
WATER CLOSE
FOOD OPEN
FOOD CLOSE
WATER OPEN


Audio