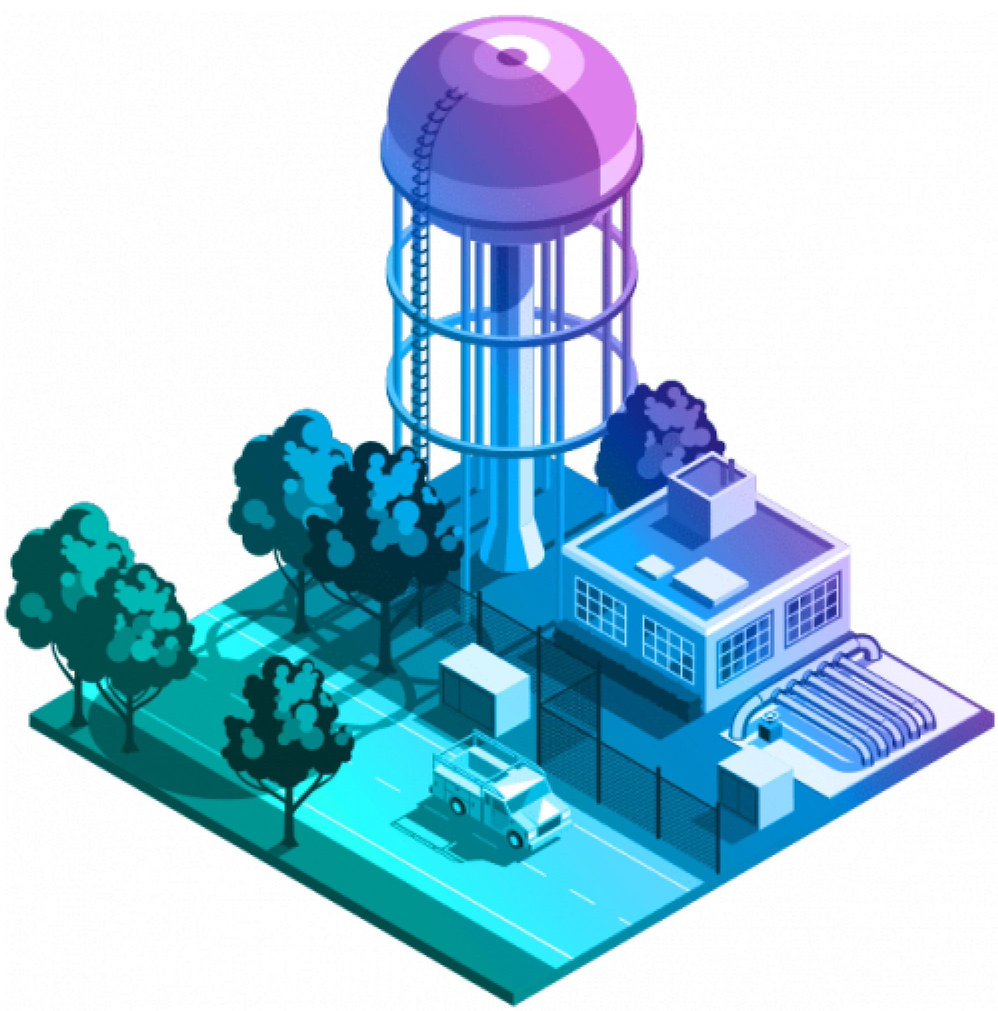
Intelligent Water Distribution System

by

Maram Tejaswanth Reddy

Rituraj Anand

Guturu Siva Naga Nihith



1. Introduction

a. Overview

The water is a significant resource for individual human and its existence. Nowadays, the population in cities are increasing rapidly, due to the migration of people from rural areas to urban areas. Water scarcity is among the main problems to be faced by many societies and the world in the 21st century. There is a lack of water management ICT standards in water supply system. Some areas get water while other some areas can't so, there is a need of continuous monitoring, water supply scheduling and proper distribution. With the advent of the Internet of Things (IoT), the water supply system is a natural choice for instrumentation with a network of sensors that can be communicated with each other and gather data for monitoring and automation.

b. Purpose:

In this system, a smart water monitoring and automation system model is proposed with integrating Internet of Things (IoT) technologies and controller with sensors. Intelligent monitoring is defined as a method which is used to monitor, control, manage and optimize the network by using different computational methods that will provide customers with relevant tools and information. The internet of things (IoT) forms an important part of intelligent monitoring which connects people and devices using wireless sensor technology. It is a fast growing research area in the military, energy management, healthcare and many more. IoT makes it possible to transfer information between different electronic devices embedded with new technology. Water distribution system(WDS) is a very important research area that affects the economic growth of our country.

2. Literature Survey:

a. Existing Problem:

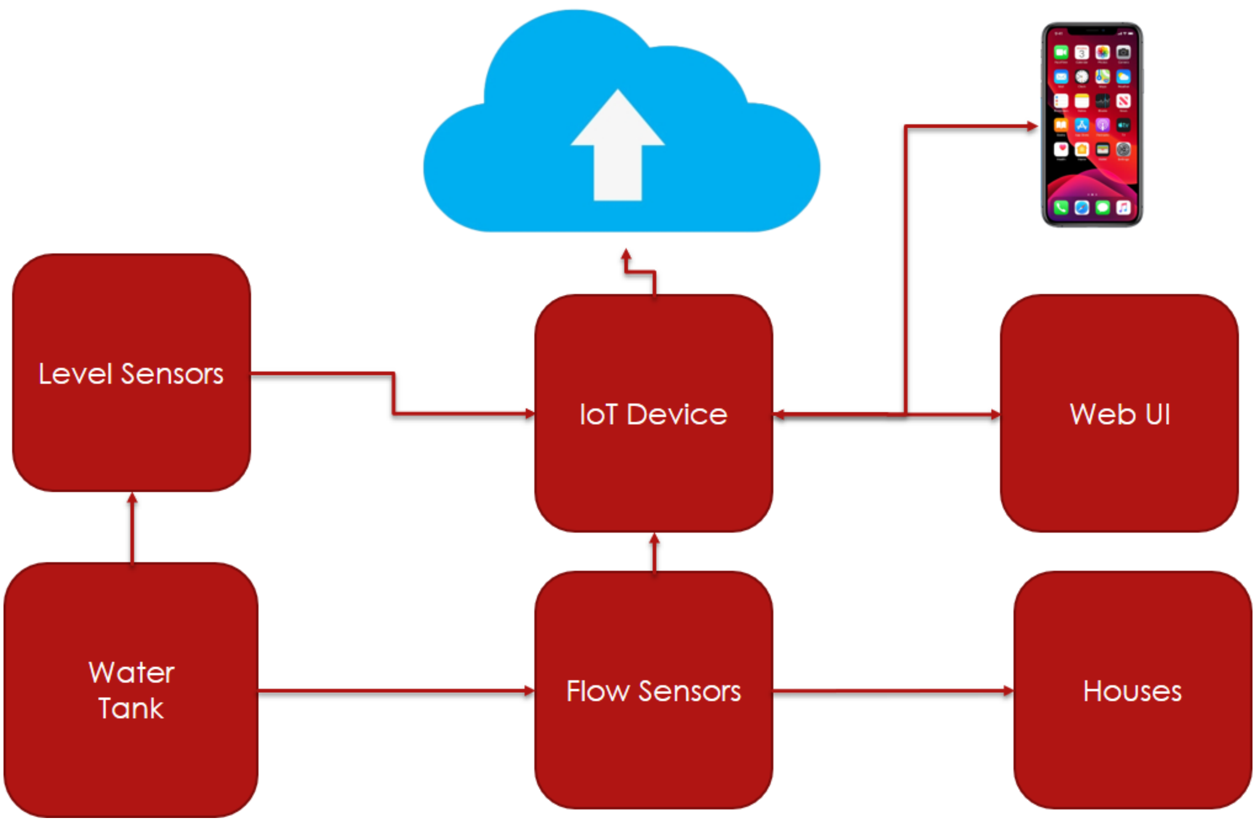
Water saving is the one of the main objectives behind all the development and control. Water scarcity happens to be one of the current issues in the world. Earth is full of water. However, only 1% of water is available as groundwater and surface water for human survival.Hence, According to a professional survey conducted recently, it is observed that almost 1.3 billion litres of water per day is needed in India. However, only 900 million litres of water are supplied to the citizen. unauthorized water connections and poor maintenance are some of the factors contributing to wastage. If it is possible to save this wastage of water, then required quantity of water can be supplied. In the existing system, distribution of water is based on the bill payment but not on the need of the consumers and history of pattern of consumption of the consumers. Advanced technology with implementation of IoT techniques can be a very efficient solution to this problem

b. Proposed System:

A system which can continuously monitor the water level of the main water tank and be capable of sending alerts whenever the tanks gets full or empty and the system should also be capable of monitoring the water consumed by each house by using flow sensors to track the individual usage of water. Bills are automatically generated and sent to the people directly to their phones as an SMS after every tank refill. A web app should also be generated to display the details of water consumption of all houses along with the bill and payment details.

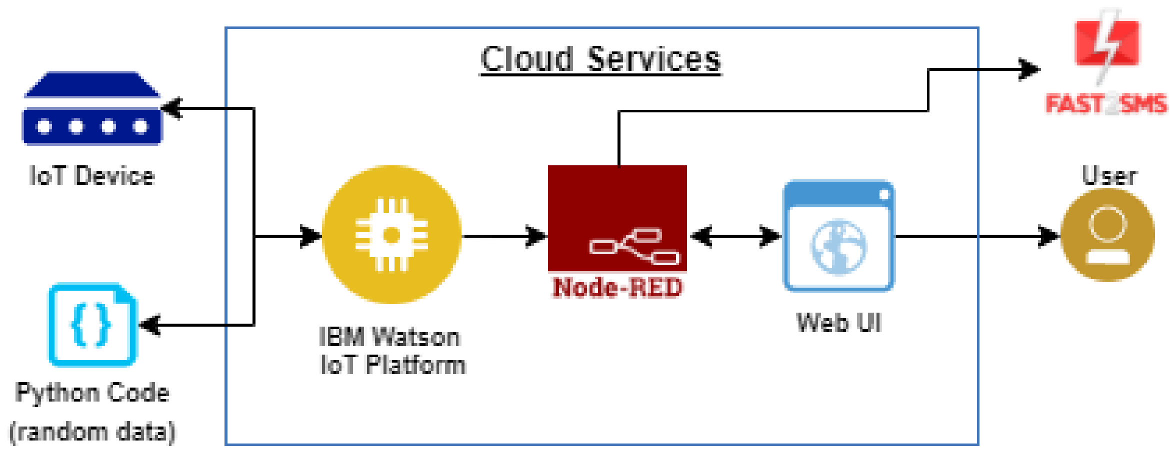
3. Theoritical Analysis:

a. Block Diagram:



The above block diagram depicts the flow of the system explaining the flow of information from the sensors to IoT device where all the information is passed to Node RED to display on the web UI and to the cloud storage. The device also sends the corresponding alerts to the mobile devices of the users and the maintainance authorities.

b. Software Designing:



While there is no exculusive hardware is used, all the working is simulated with the following softwares

1) Python IDLE
2) IBM cloud services
3) IBM watson IoT platform
4) Node-RED
5) Fast2SMS
6) Cloudant db

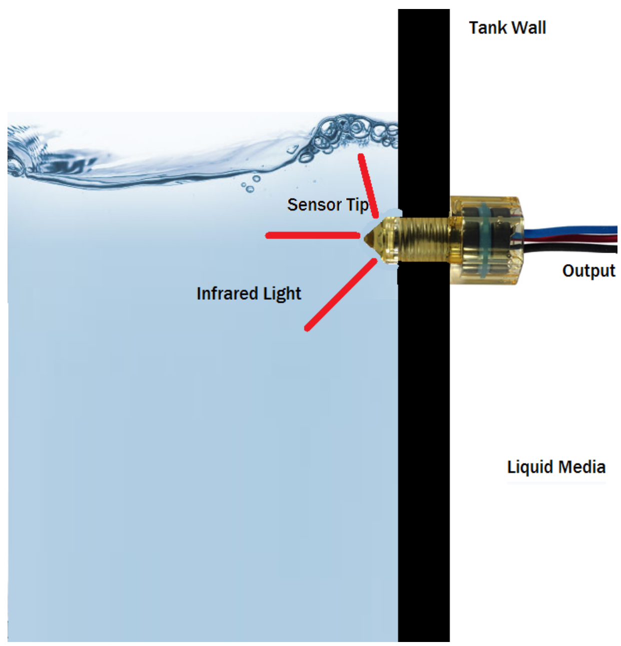
Python is used to simulate the sensor values and publish the data to the cloud and IoT device. IBM cloud services such as IBM watson IoT platform, Cloudant db is used to store the data and Node RED is used for the web UI to show the information in graphical representations to the user. Fast2SMS is used to send the corresponding alerts to the corresponding people.

4. Experimental Investigations:

How does the Level and Flow sensors work?

Level Sensor:

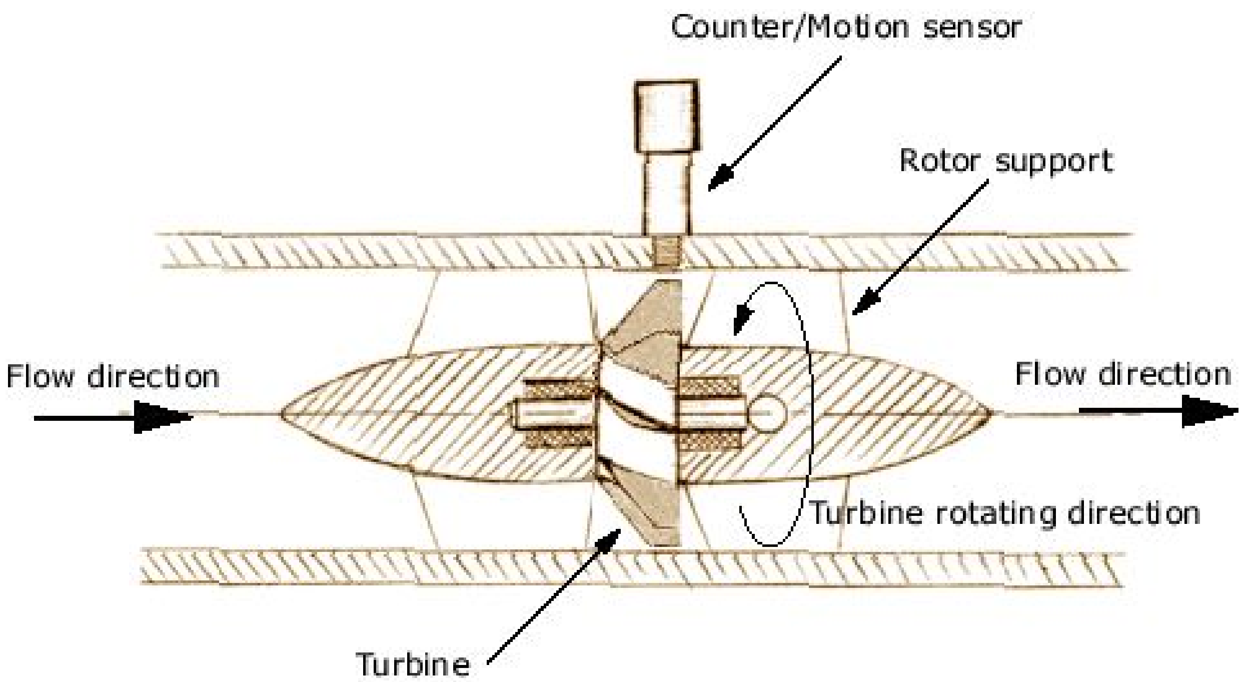
A level sensor is a device that is designed to monitor, maintain, and measure liquid (and sometimes solid) levels. Once the liquid level is detected, the sensor converts the perceived data into an electric signal. Level sensors are used primarily in the manufacturing and automotive industries, but they can be found in many household appliances as well, such as ice makers in refrigerators.



There are two main classifications for level sensors: point level sensors and continuous level sensors. Point level sensors are designed to indicate whether a liquid has reached a specific point in a container. Continuous level sensors, on the other hand, are used to render precise liquid level measurements. Level sensors can be divided further into invasive and non-contact sensors. Invasive sensors make direct contact with the substance they are measuring, while non-contact sensors use sound or microwaves.

Flow Sensor:

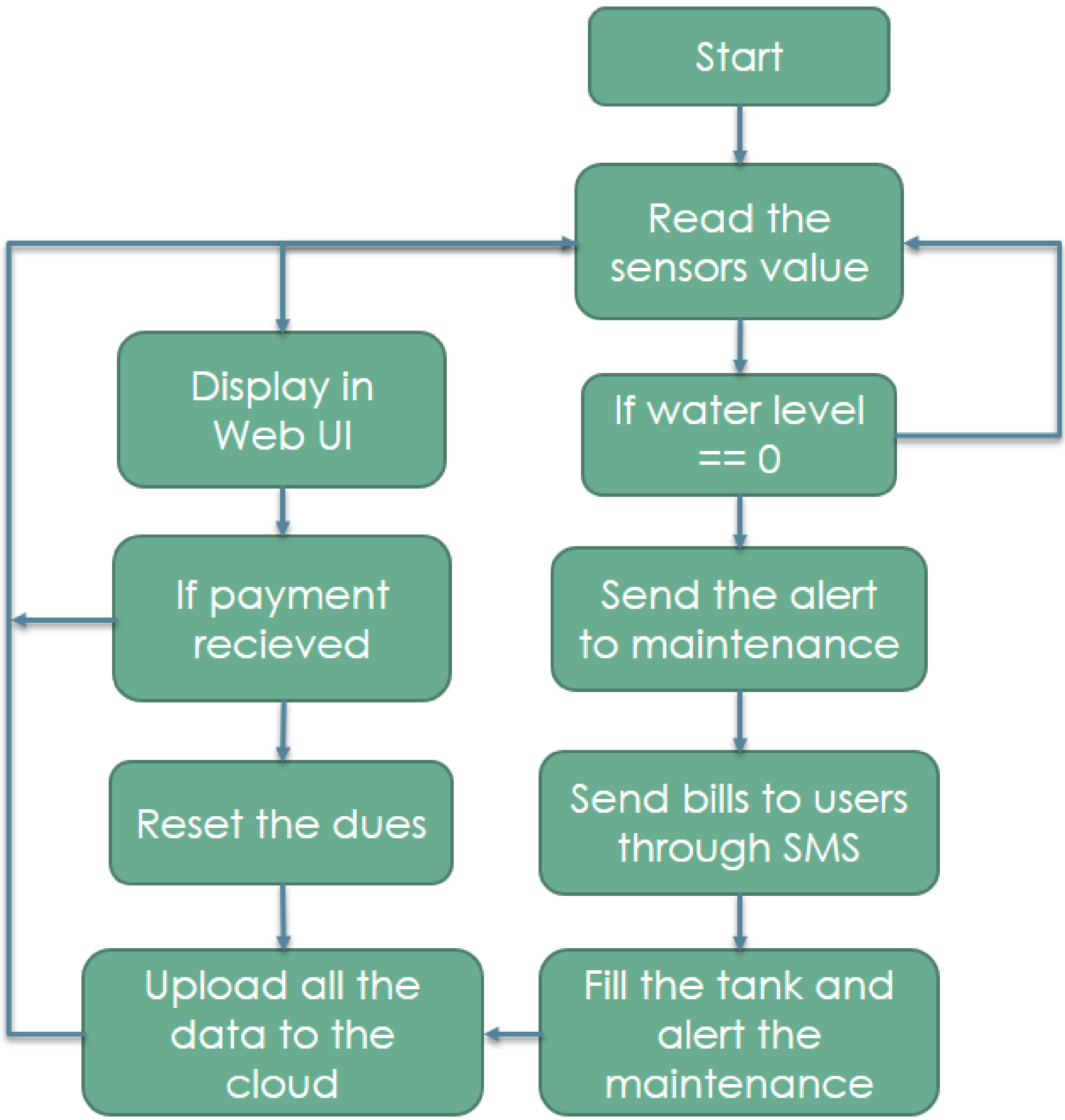
The most common and economical type of water flow meter are the mechanical types of water flow meters which measure flow through turbine rotation with a propeller, shunt or paddle wheel design. These mechanical water flow meters work by measuring the speed of flowing water running through the pipe that causes a turbine or piston to rotate. The volumetric flow rate of the water is proportional to the rotational speed of the blades.



The downside with mechanical water flow meters for water measurement is that they may clog up when the water is dirty or has larger particles which increases maintenance costs. Mechanical water meters also don't work well with a low flow of water.

5. Flow Chart:

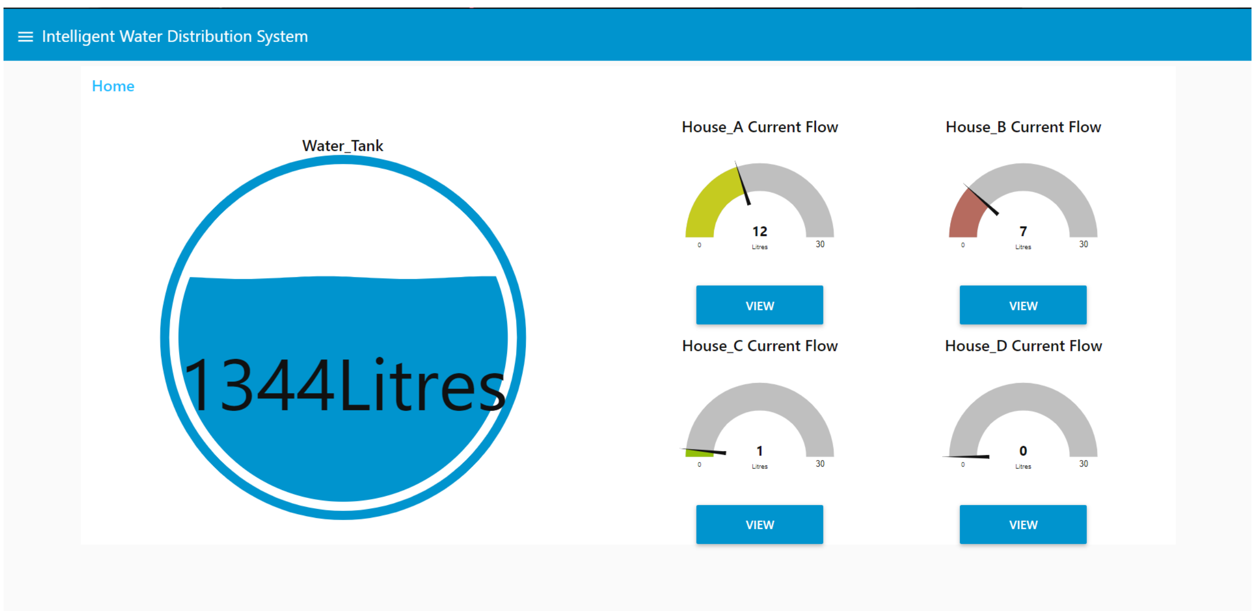
The following chart shows the flow of the python code covering all the desired features of the system.



6. Result:

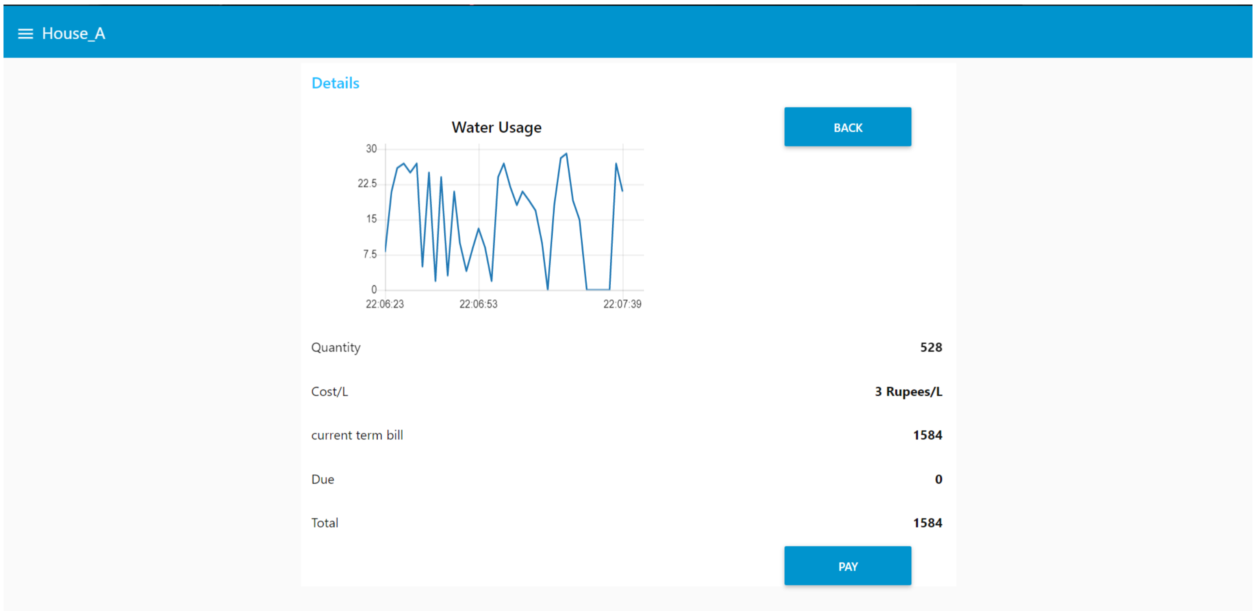
Web UI:

The below picture shows the Home page of the web application which displays the Live stats of the main water tank and flow of water into each house which are essentially the data from the level and flow sensors respectively.



Each house has their own deatails page which can be opened by clicking the view button under the corresponding house name. The details page consists of individual details of the selected house such as the quantity of the water used, cost of water per litre during the term, past dues (if there are any) and the total amount to be paid. The page also have a line chart depicting the water usage trend.

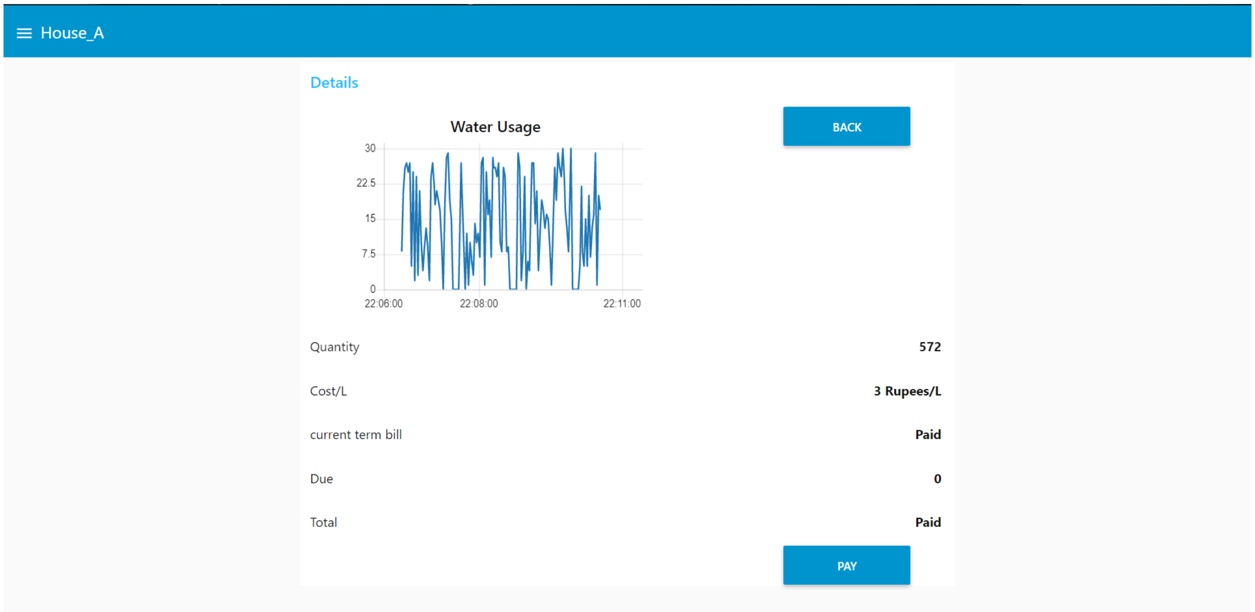
Below is the picture of House_A details after a term of usage of the service.



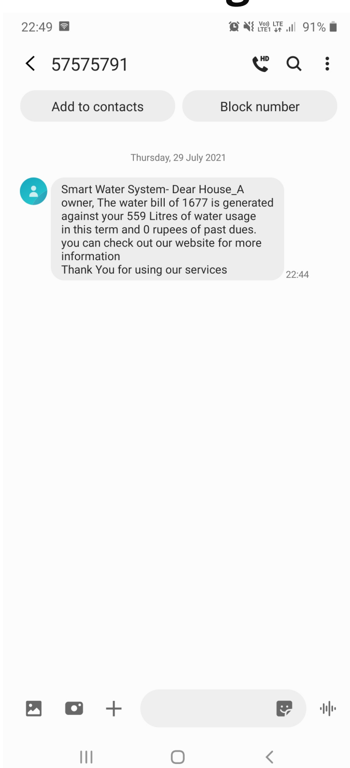
As it can be seen the House_A consumed 528 Litres of water at the cost of 3 rupees per Litre, which makes a bill of 1584 rupees and is displayed in the current term bill section. And as the due shows 0, there is no past dues for the user. The total displays the total amount to be paid which is the sum of due and current term bill. We can also see a chart representing the water usage trends of the user.

The page also contains two clickable buttons, one is the back button which can be found right next to the chart which takes us back to the home page. And the pay button clears the bill along with the dues and displays "paid" when done.

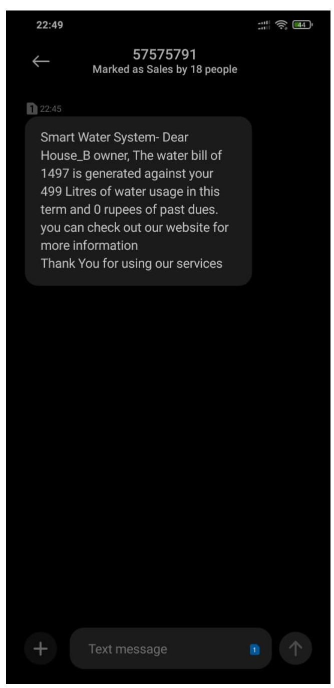
Below is the photo of the page after clicking pay.



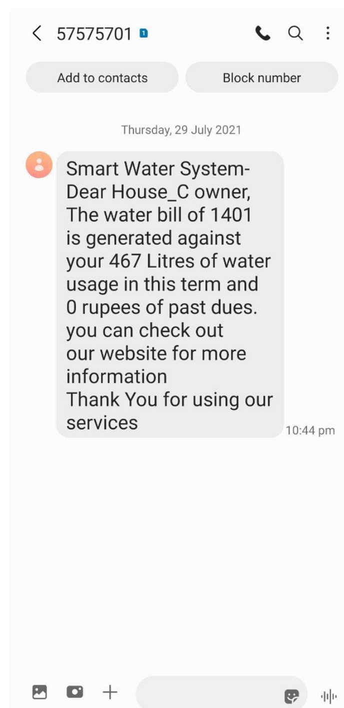
Alerts through Fast2SMS:



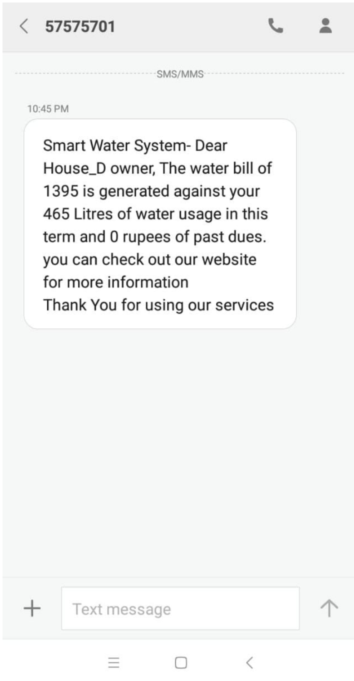
Bill for House_A

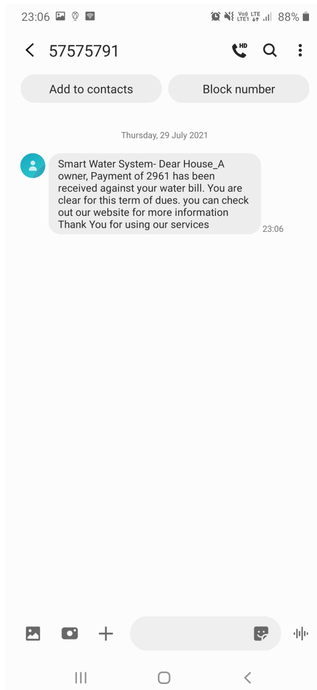


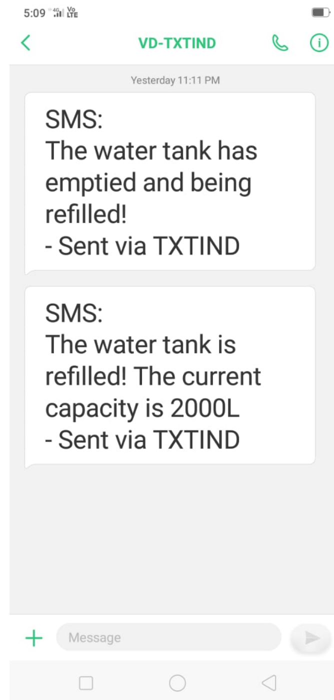
Bill for House_B



Bill for House_C

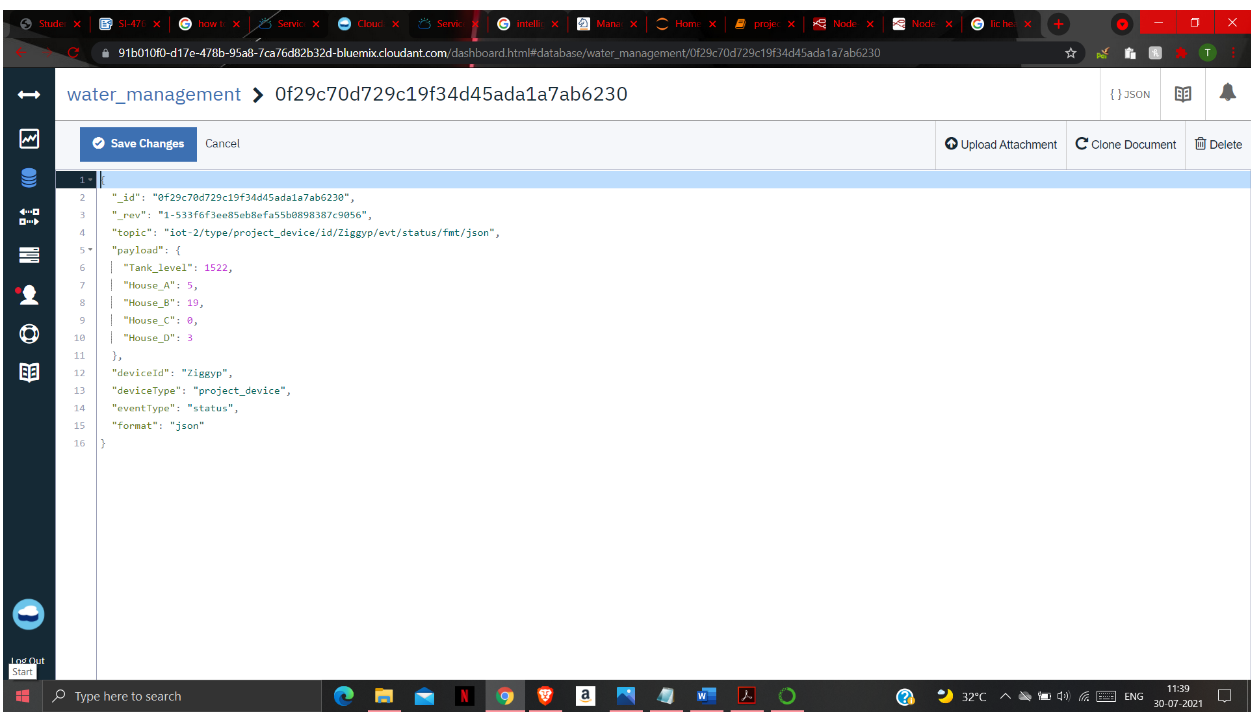






Bill for House_D payment notification Empty/fill alert

Cloudabt db:



A document uploaded in clodant db through Node RED with all the information

7. Advantages and Disadvantages:

a. Advantages:

Transparency

One of the biggest benefits of smart water management using IoT is improving the transparency of all the processes in the water supply chain. Thanks to the data collected throughout the supply chain, different stakeholders get important insights on their resources and system performance. As a result, they can make informed decisions on how to improve their operations.

Automation and optimized use of human resources

IoT water management solutions allow managers to partially or fully automate some processes and optimize the use of human power. The depth and scope of the automation vary on the sector and specific business needs. For instance, smart water supply companies and utility networks can automate the entire lifecycle of providing water to consumers by using connected meters, real-time monitoring systems and dynamic pricing models.

Sustainability

Sustainability goals are at the center of many retrofit and innovation projects not only in the smart water industry but across any other sector such as energy, construction, logistics, etc. Technologies for smart water use are no longer seen as a source of savings and higher efficiency only, but as the means to reach different environmental goals including reduced carbon footprint, pollution and, essentially, water preservation.

Optimized cost

Automation, optimized use of human resources, data-driven strategy and proactive approach to equipment maintenance and resource usage eventually translate in significant savings. One of the reasons why water companies look into the use of IoT in water management is to reduce operational costs in the long run.

Sustainability

Sustainability goals are at the center of many retrofit and innovation projects not only in the smart water industry but across any other sector such as energy, construction, logistics, etc. Technologies for smart water use are no longer seen as a source of savings and higher efficiency only, but as the means to reach different environmental goals including reduced carbon footprint, pollution and, essentially, water preservation.

Immediate response

Another benefit of integrating intelligent water management systems is the ability to identify or even predict issues and respond immediately to minimize damage. For instance, real-time monitoring of water quality and chemical composition allows to detect even a slight contamination and initiate rapid response before it becomes dangerous.

b. Disadvantages:

While there are no specific disdvantages, one can point out the general complications or drawbacks of an IoT based automated system such as privacy and security of the data, complexity for general troubleshooting, unemployment of unskilled workers due to automation, Internet and power connectivity dependence etc. For this particular project we can say that the system lacks some functionalities like integrating pH sensors, water quality measurement etc. but these features are considered as the future scope for the development of the project.

8. Applications/Features:

- Continuous monitoring of water level of the main water tank
- controlling the water pumps based on the level of the water
- An alert is triggered whenever the tank gets full/empty
- monitoring the amount of water consumed by each house by using flow sensors.
- bills are generated and sent to the people
- web app is developed to display the water consumption details of all the houses.
- By replacing the corresponding sensors and changing the relevant software programs, this system can be used to monitor other water quality parameters.

9. Conclusion:

In this project, IoT based water supply monitoring and distribution system is presented. Continous water monitoring in the main tank and flow into each house, generating bills based on usage, sending alerts to the concerned people, displaying all details in the web UI, storing the information in cloud, the main objectives of the project are fulfilled and implemented succesfully. An IBM IoT device as the main controller unit and Python code for simulating the sensors, publishing the data to the device, Fast2SMS to send multiple type of alerts, Node RED for the web UI and cloudant db for storing the information are all the stuff that are used in the project. In this system the data can be viewed from anywhere in the world with internet infrastructure. The system has good flexibility. Only by replacing the corresponding sensors and changing the relevant software programs, this system can be used to monitor other water quality parameters. The operation is simple. It has widespread application and extension value.

10. Future Scope:

In the future, the system can be upgraded to monitor and control by the cloud server with other sensors such as PH sensor for the PH level of water, soil moisture sensor and etc .In future Detecting the more parameters for most secure purpose .The system can be expanded to monitor hydrologic, air pollution, industrial and agricultural production and so on. By keeping the embedded devices in the environment for monitoring enables self protection to the environment. To implement this need to deploy the sensor devices in the environment for collecting the data and analysis. By deploying sensor devices in the environment, we can bring the environment into real life i.e. it can interact with other objects through the network. Then the collected data and analysis results will be available to the end user through the Wi-Fi.

11. Bibliography:

[1] Prachet Verma, Akshay Kumar, Nihesh Rathod, Pratik Jain, Mallikarjun S, Renu Subramanian, Bharadwaj Amrutur, M.S. Mohan Kumar, Rajesh Sundaresan, “Towards an IoT Based Water Management System for a Campus” IEEE, 2015.

[2] Joy Shah, "An Internet of Things Based Model for Water Distribution with Quality Monitoring", International Journal of Innovative Research in Science Engineering and Technology, vol. 6, no. 3, March 2017.

[3] Perumal, "Internet of Things (IoT) Enabled Water", Monitoring System 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE).

Google Scholar

[4] Pranita Vijaykumar Kulkarni, Mrs. M.S Joshi, “An IoT based water supply monitoring and controlling system with theft identification” IJIRSET Vol. 5, Issue 9, September 2016.

12. Appendix:

a. Source Code:

This is the python code for the project

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

wt = 2000
f1 = [0]
f2 = [0]
f3 = [0]
f4 = [0]
due = [0,0,0,0]
billed = [0,0,0,0]
total = [0,0,0,0]

myConfig = {
    "identity": {
        "orgId": "udbkdj",
        "typeId": "project_device",
        "deviceId":"Ziggyp"
    },
    "auth": {
        "token": "919347495679"
    }
}
```

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

def publish():
    global f1, f2, f3, f4, wt

    myData={'Tank_level': wt, 'House_A':f1[-1], 'House_B':f2[-1],
'House_C':f3[-1], 'House_D':f4[-1]}
    client.publishEvent(eventId="status", msgFormat="json",
data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)

def empty_alert():
    url =
"https://www.fast2sms.com/dev/bulkV2?authorization=RMnNiWhw
prVQPFd4BTzGuCbeH2EAvfJOcUqKDxY6LjsSla9y3txN2psik6Am04w
L75gIzt9fQTSYavhB&route=v3&sender_id=TXTIND&message=The%
20water%20tank%20has%20emptied%20and%20being%20refilled!
&language=english&flash=0&numbers=9030435887"
    sen = requests.get(url)
    print(sen.text)

def full_alert():
    url =
"https://www.fast2sms.com/dev/bulkV2?authorization=RMnNiWhw
prVQPFd4BTzGuCbeH2EAvfJOcUqKDxY6LjsSla9y3txN2psik6Am04w
L75gIzt9fQTSYavhB&route=v3&sender_id=TXTIND&message=The%
20water%20tank%20is%20refilled!%20The%20current%20capacity
```

```
%20is%202000L&language=english&flash=0&numbers=903043588
7"
    sen = requests.get(url)
    print(sen.text)

def billnotif(house, summ, quantity, loan, pnum):
    base_url =
"https://www.fast2sms.com/dev/bulkV2?authorization=RMnNiWhw
prVQPFd4BTzGuCbeH2EAvfJOcUqKDxY6LjsSla9y3txN2psik6Am04w
L75gIzt9fQTSYavhB&route=q&message=Smart%20Water%20Syste
m-%20Dear%20House_A%20owner,%20The%20water%20bill%20o
f%20summ_%20is%20generated%20against%20your%20quantity_
%20Litres%20of%20water%20usage%20in%20this%20term%20an
d%20loan_%20rupees%20of%20past%20dues.%20you%20can%20
check%20out%20our%20website%20for%20more%20information
%0AThank%20You%20for%20using%20our%20services&language=
english&flash=0&numbers=9347495679"
    url = base_url.replace("House_A", house)
    url = url.replace("summ_", str(summ))
    url = url.replace("quantity_", str(quantity))
    url = url.replace("loan_", str(loan))
    url = url.replace("9347495679", str(pnum))

    sen = requests.get(url)
    print(sen.text)

def publishbills():
    global f1, f2, f3, f4, due, billed, total
```

```
    billed[0] = billed[0]+sum(f1)
    billed[1] = billed[1]+sum(f2)
    billed[2] = billed[2]+sum(f3)
    billed[3] = billed[3]+sum(f4)

    due[0] = (billed[0]-sum(f1))*3
    due[1] = (billed[1]-sum(f2))*3
    due[2] = (billed[2]-sum(f3))*3
    due[3] = (billed[3]-sum(f4))*3

    total[0] = sum(f1)*3+due[0]
    total[1] = sum(f2)*3+due[1]
    total[2] = sum(f3)*3+due[2]
    total[3] = sum(f4)*3+due[3]

    myData={'Quan_A':sum(f1), 'Quan_B':sum(f2), 'Quan_C':sum(f3), 
'Quan_D':sum(f4),
            'Due_A': due[0], 'Due_B': due[1], 'Due_C': due[2], 'Due_D': 
due[3],
            'Cur_A': sum(f1)*3, 'Cur_B': sum(f2)*3, 'Cur_C': sum(f3)*3, 
'Cur_D': sum(f4)*3,
            'Total_A': total[0], 'Total_B': total[1], 'Total_C': total[2], 
'Total_D': total[3]}

    client.publishEvent(eventId="status", msgFormat="json", 
data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    billnotif("House_A", total[0], sum(f1), due[0], 9347495679)
    billnotif("House_B", total[1], sum(f2), due[1], 7903051892)
```

```
    billnotif("House_C", total[2], sum(f3), due[2], 9338971111)
    billnotif("House_D", total[3], sum(f4), due[3], 9290444031)

def consume(fa):
    global f1, f2, f3, f4, wt
    a = random.randint(0,30)
    if wt<a:
        publishbills()
        f1 = [0]
        f2 = [0]
        f3 = [0]
        f4 = [0]
        fa = [0]
        if(fa == "f1"):
            f1.append(wt)
        if(fa == "f2"):
            f2.append(wt)
        if(fa == "f3"):
            f3.append(wt)
        if(fa == "f4"):
            f4.append(wt)
        wt = 0
        publish()
        empty_alert()
        for i in range(0, 2001, 200):
            wt = i
            publish()
            time.sleep(0.5)
        full_alert()
```

```
    else:
        if(fa == "f1"):
            f1.append(a)
        if(fa == "f2"):
            f2.append(a)
        if(fa == "f3"):
            f3.append(a)
        if(fa == "f4"):
            f4.append(a)
        wt = wt-a

def paynotif(hs, tt,pn):
    base_url =
"https://www.fast2sms.com/dev/bulkV2?authorization=RMnNiWhw
prVQPFd4BTzGuCbeH2EAvfJOcUqKDxY6LjsSla9y3txN2psik6Am04w
L75gIzt9fQTSYavhB&route=q&message=Smart%20Water%20Syste
m-%20Dear%20House_A%20owner,%20Payment%20of%20Total_%
20has%20been%20received%20against%20your%20water%20bill.
%20You%20are%20clear%20for%20this%20term%20of%20dues.%
20you%20can%20check%20out%20our%20website%20for%20mor
e%20information%0AThank%20You%20for%20using%20our%20se
rvices&language=english&flash=0&numbers=9347495679"
    url = base_url.replace("House_A", hs)
    url = url.replace("Total_", str(tt))
    url = url.replace("9347495679", str(pn))
    sen = requests.get(url)
    print(sen.text)
```

```
def paybills(m):
    global due, billed, f1, f2, f3, f4, total
    if m == "Pay_A":
        paynotif("House_A", total[0], 9347495679)
        billed[0] = 0
        due[0] = 0
        total[0] = 0
        myData={'Total_A': "Paid", 'Cur_A': "Paid", 'Due_A': due[0]}
        client.publishEvent(eventId="status", msgFormat="json",
data=myData, qos=0, onPublish=None)
        print("Published data Successfully: %s", myData)
    elif m == "Pay_B":
        paynotif("House_B", total[1], 7903051892)
        billed[1] = 0
        due [1] = 0
        total[1] = 0
        myData={'Total_B': "Paid", 'Cur_B': "Paid", 'Due_B': due[1]}
        client.publishEvent(eventId="status", msgFormat="json",
data=myData, qos=0, onPublish=None)
        print("Published data Successfully: %s", myData)
    elif m == "Pay_C":
        paynotif("House_C", total[2], 9338971111)
        billed[2] = 0
        due[2] = 0
        total[2] = 0
        myData={'Total_C': "Paid", 'Cur_C': "Paid", 'Due_C': due[2]}
        client.publishEvent(eventId="status", msgFormat="json",
data=myData, qos=0, onPublish=None)
```

```
        print("Published data Successfully: %s", myData)
    elif m == "Pay_D":
        paynotif("House_D", total[3], 8978715993)
        billed[3] = 0
        due[3] = 0
        total[3] = 0
        myData={'Total_D': "Paid", 'Cur_D': "Paid", 'Due_D': due[3]}
        client.publishEvent(eventId="status", msgFormat="json",
data=myData, qos=0, onPublish=None)
        print("Published data Successfully: %s", myData)


def myCommandCallback(cmd):
    print("\nMessage received from IBM IoT Platform: %s" %
cmd.data['command'])
    m = cmd.data['command']
    print(m)
    if "Pay" in str(m):
        paybills(m)

while True:
    consume("f1")
    consume("f2")
    consume("f3")
    consume("f4")
    publish()
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

b. UI output Screenshot:

(more about this can be found in the section number 6, Result)