

MEDICINE REMINDER FOR ELDERLY PEOPLE

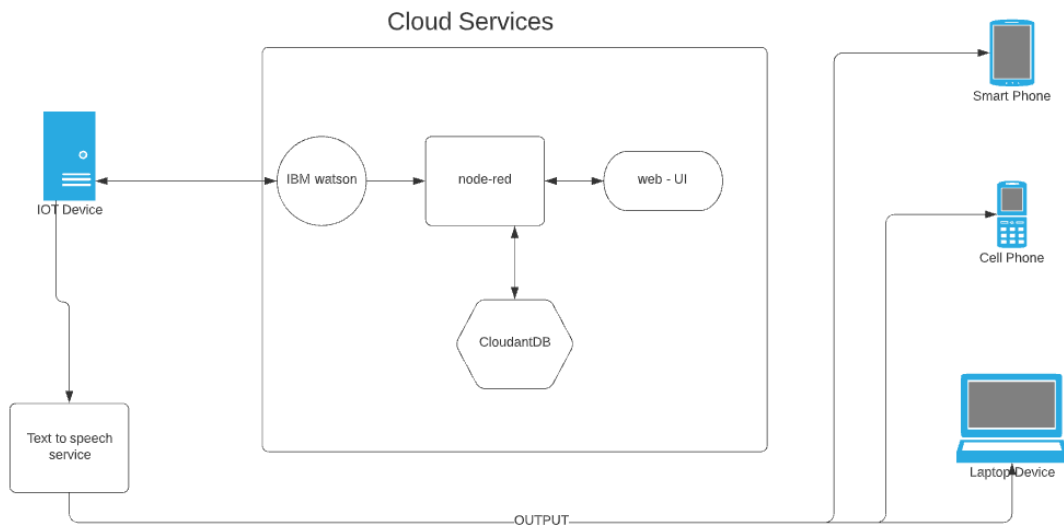
Introduction:

Our medicine reminder is designed for helping old people in taking care of themselves in taking their medications at the correct time and in the correct day. It has been observed that people in general neglect their health and give preference to other things than taking their medicines. Some patients are so occupied with their day-to-day activities that they just forget to take their medications. This is particularly true for old patients who have to take more than one medicine at more than one time in a day. Setting alarm clocks is a tedious task which patients are too lazy to set again and again. If asked about what time people have to take their medicines, many forget to answer the correct times or remember whether they have already taken the medicine in the day already. The patients can leave taking medicines to just our web application. Whenever the time for the medicine is up, they will be notified and our voice assistance will notify you about your medicines. Sometimes elderly people forget to take the medicine at the correct time. They also forget which medicine He / She should take at that particular time... And it is difficult for doctors/caretakers to monitor the patients around the clock. To avoid this problem, this medicine reminder system is developed.

System Overview:

Our system has only 1 part i.e, software part. The software portion will do the reminder part of the task, which is to remind patients to take their medicines at correct time as they are supposed to take. The reminder can be set using the web application, where users need to add the medicine name, time and date and your work is done. Now our web application will take care rest of the work for you. It will remind you according to your given time.

Flow Diagram:

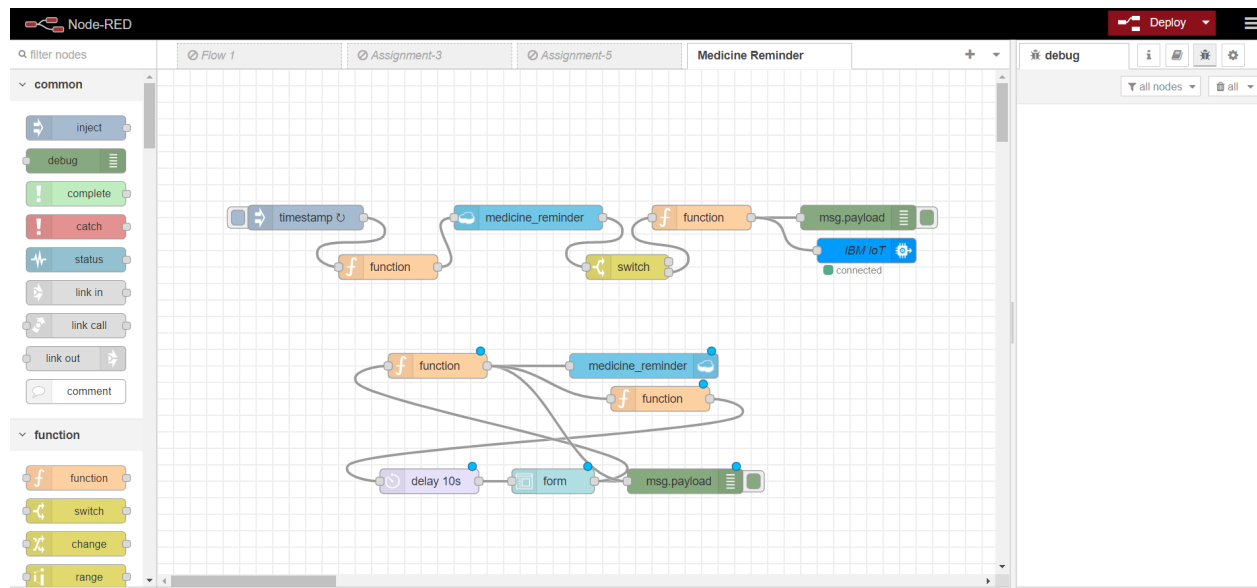


Software and Services Used:

We used the IoT Watson platform as a base for our project. After that, we create accounts on the IBM cloud for making various other services. In IBM Cloud we first created a Node-red service, in which we created different flows for our web applications. we used some nodes like debug console, Functions, Form, delay, etc to our system work properly. Then we created an account in CloudantDB to store data into the database. This is the main tool for our project. First, we take data from users and store that data into this database and according to this provided data, our system works. Lastly, we used the Text-To-Speech service so that our users get their reminders on time and it is very effective for elder people.

Problem: Medicine Reminder For Elderly People Using IBM Cloud

Node-RED Nodes:



CloudbantDB:

Databases

Database name

Create Database

{ } JSON

🔔

Name	Size	# of Docs	Partitioned	Actions
assignment-5	38 bytes	1	Yes	🔍 🔒 🗑️
medicine_reminder	2 bytes	1	No	🔍 🔒 🗑️
nodoredxrdm20211202	36.6 KB	4	No	🔍 🔒 🗑️
products	301 bytes	1	Yes	🔍 🔒 🗑️
vit-bhopal	37 bytes	1	No	🔍 🔒 🗑️

Database For Medicine/date/time

Web UI:

VIT-B Dashboard

Medical Reminder

Enter the Medicine

Time (HH:MM) 24H

Date (YYYY-MM-DD)

SUBMIT

CANCEL

Form Setting:

Edit form node

Delete

Cancel

Done

Properties

Group

[VIT-B Dashboard] Medical Reminder

Size

6 x 3

Label

optional label

Form elements

	Label	Name	Type	Required	UiRows	Remove
≡	Enter the Medicine	Name	Text	<input checked="" type="checkbox"/>		
≡	Time (HH:MM) 24H	Time	Time	<input checked="" type="checkbox"/>		
≡	Date (YYYY-MM-D	Date	Date	<input checked="" type="checkbox"/>		

element

Buttons

submit

cancel

Enabled

The function used to create id for DB:

Edit function node

Delete

Cancel

Done

Properties

Name

Setup

On Start

On Message

On Stop

```
1 var d=msg.payload.date
2 var t=msg.payload.time
3 msg.payload = {
4   "_id":d+" "+t,
5   "name":msg.payload.name
6 }
7 return msg;
```

☐ Enabled

Function used :

Edit function node

Delete

Cancel

Done

⚙ Properties



📌 Name

Name



⚙ Setup

On Start

On Message

On Stop

```
1 var d = new Date();
2 var utc = d.getTime() + (d.getTimezoneOffset() * 60000);
3 var offset = 5.5;
4 newDate = new Date(utc + (3600000*offset));
5 var n=newDate.toISOString()
6 var date = n.slice(0,10)
7 var time = n.slice(11,16)
8 global.set('time',time)
9 msg.payload=date+" "+time
10 return msg;
```

☐ Enabled

Edit function node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🔗

🔖 Name

Name

📄 ▼

⚙ Setup

On Start

On Message

On Stop

1 ▾ msg.payload = {

2 "date":"" ,

3 "name":"" ,

4 "time":""

5

6 ▲ }

7 return msg;

↗

☐ Enabled

Advantages of this system:

- You can manually set all medicine and reminder time according to your need.
- It is very easy to use.
- Very user friendly UI for users.
- It has a pre build voice alert feature.

Disadvantages of this system:

- Reminders cannot be set automatically. There is a need for manual work in setting the reminder.
- A lot of time is consumed in manually setting the reminders.
- Sometime it may stop working due to cloud database issue.

Source Code:

```
import wiotp.sdk.device
import time
import os
import datetime
from ibm_watson import TextToSpeechV1
from ibmcloudant.cloudant_v1 import CloudantV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator , BasicAuthenticator
import playsound

"authenticator =
BasicAuthenticator('apikey-v2-34uhhlix6hcy13v2gisbh6vsiegrpl5wkwasymqnf16',
'7f517d7221399711c579625ed6d3e874')
service = CloudantV1(authenticator=authenticator)
service.set_service_url('https://apikey-v2-34uhhlix6hcy13v2gisbh6vsiegrpl5wkwasymq
nf16:7f517d7221399711c579625ed6d3e874@ff20a5d8-f28e-4fa4-b046-473245d0e43f-
bluemix.cloudantnosqldb.appdomain.cloud')"

authenticator = IAMAuthenticator('kZsyZ69sNctJRNfM7QxalUZ-setaah0drfP3s2tcXTr_')
text_to_speech = TextToSpeechV1(
    authenticator=authenticator
)

text_to_speech.set_service_url('https://api.us-south.text-to-speech.watson.cloud.ibm.co
m/instances/a445e70c-f7f6-4625-8ced-2434293c320c')
```



```

myConfig = {
    "identity": {
        "orgId": "y9045l",
        "typeId": "mobile",
        "deviceId": "09876"
    },
    "auth": {
        "token": "12345678"
    }
}
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform %s" % cmd.data['command'])
    m=cmd.data['command']
    with open('medicine.mp3', 'wb') as audio_file:
        audio_file.write(
            text_to_speech.synthesize(
                'You have to take'+m+'medicine now',
                voice='en-US_AllisonV3Voice',
                accept='audio/mp3'
            ).get_result().content)
    os.remove('medicine.mp3')
    while True:
        client.commandCallback = myCommandCallback
        client.disconnect()

```

Conclusion

The IoT device market has undergone radical changes in only a few short years. Starting with disparate devices and no ecosystems to speak of, the market has now grown to encompass enterprise players working together to create ecosystems, tailored for mobile technology, which allows IoT devices to become interconnected.

Automation of the medical service may have once seemed like a peculiar and unlikely concept, but as our devices become smarter and more investment is

poured into the development of IoT consumer products, we are likely to see increased competition spur on further innovation in the field.

Group Members:-

Sourabh Patnayaik

Sai Ganaamruth Kasturi

Daksh Pathak

Harsh Agarwal