

IBM - NALAIYATHIRAN



THIAGARAJAR COLLEGE OF ENGINEERING

Department of Electronics and Communication Engineering

PERSONAL ASSISTANCE FOR SENIORS WHO ARE SELF RELIANT

Team ID: PNT2022TMID21391



TEAM DETAILS

TEAM-ID: PNT2022TMID21391

TEAM MEMBERS

Reg. No.	IBM Reg. No.	Name
19D058	917719D058	Nisha C
19D022	917719D022	Fathima D
19D028	917719D028	Harni V
19D134	917719D134	Sangeetha M

ABSTRACT

The history of agriculture began thousands of years ago. After gathering wild grains beginning at least 105,000 years ago, nascent farmers began to plant them around 11,500 years ago. Agriculture is the practice of cultivating plants and livestock. Agriculture was the key development in the rise of sedentary human civilization, whereby farming of domesticated species created food surpluses that enabled people to live in cities. Modern agronomy, plant breeding, agrochemicals such as pesticides and fertilizers, and technological developments have sharply increased crop yields, but cause ecological and environmental damage. Agriculture is the backbone of Indian Economy. The improvement of agriculture sector provides enormous opportunities for common folks, business people. So, in order to improve the farming methods, we use technology (Iot applications) rather than modern technology. This helps in providing precise condition of the farms which include temperature, humidity, range, soil conditions. This information when known to famers beforehand or to owners in their remote location helps them to improve the field and the farming conditions. The improved farming method thus result in increase in output

Content

CHAPTER	TITLE
1	Introduction
2	Literature Survey
3	Ideation and proposed solution
4	Requirement analysis
5	Project design
6	Project planning and scheduling
7	Coding and solutioning
8	Testing
9	Results
10	Advantages and Disadvantages
11	Conclusion
12	Future scope
13	Appendix

1.INTRODUCTION

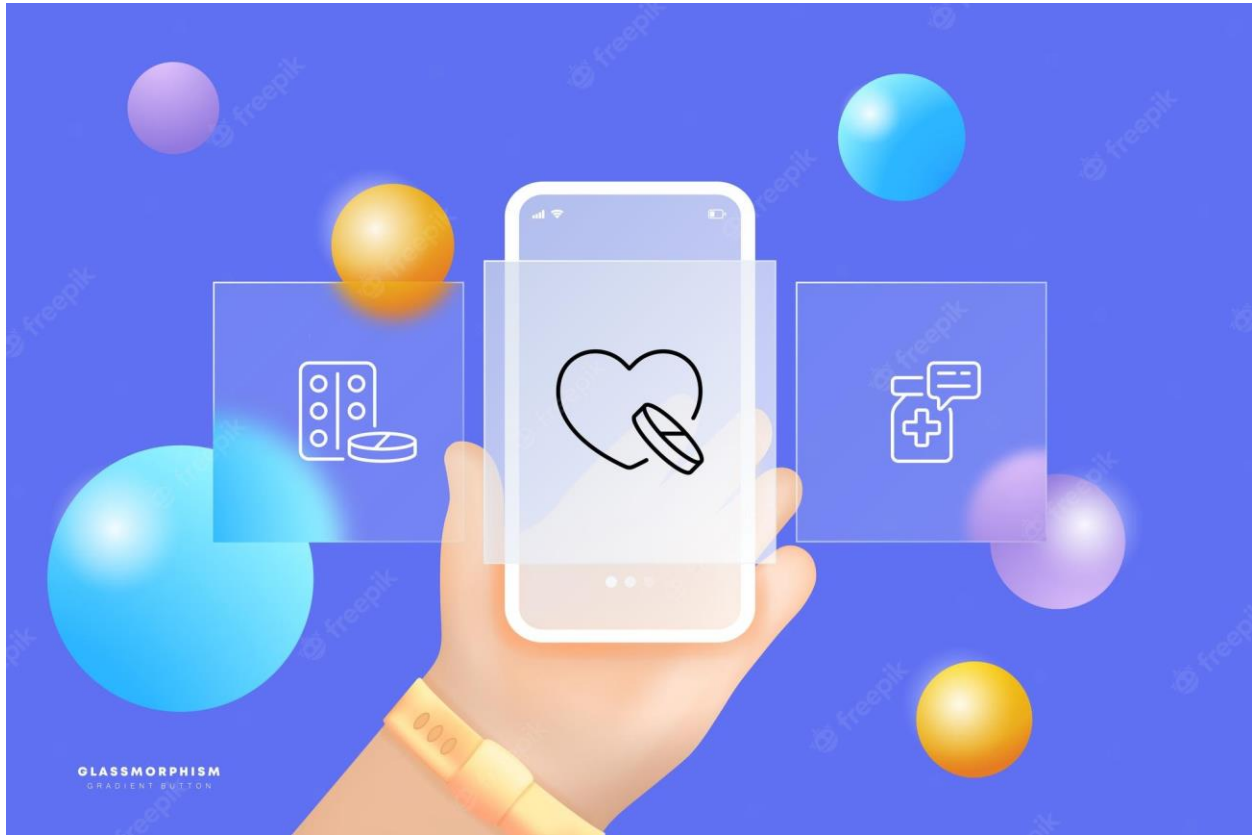


Fig.1.1. IoT based Medicine Remainder System

1.1 Project Overview

- Personal assistance application is especially meant for elders who may forget to take their daily medications on time. It is designed for users who are in need of their medication schedule.
- It is difficult for doctors/caretakers to monitor the patients regularly. To avoid this problem, this medicine reminder system is developed, it helps to take medications at correct time.
- Personal assistance application will ask users to add the medicine names and time based upon their prescription, reminder will be updated which will be visible to the user. Also, users can get voice commands to take their medications. They have an option to set an alarm which will notify

them to take medications, they can either stop or snooze the alarm as per their need.

1.2.Purpose

- Sometimes elderly people forget to take their medicine at the correct 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.
- An app is built for the user (caretaker) which enables him to set the desired time and medicine. These details will be stored in the IBM Cloudant DB. If the medicine time arrives the web application will send the medicine name to the IoT Device through the IBM IoT platform. The device will receive the medicine name and notify the user with voice commands.

2.LITERATURE SURVEY

2.1.Existing problem

S.NO .	PAPER TITLE	AUTHOR NAME	PUBLICATION YEAR	RESULTS
1.	A Research on Home Automation for Elderly and Physically Challenged People by using IOT.	Chaitra N, Ashwini CS, Shyma Zaidi.	2019	In the paper , the main objective is to build smart home and to make the elderly and physically challenged population more self-dependent. Which is cheaper when compared to all The interfaced system so simple as possible so that it makes physically impaired people more efficiently use it. Here the system can be operated using google assistant and android application.
2.	HABITAT: An IoT Solution for Independent Elderly	Elena Borelli , Giacomo Paolini , Francesco Antoniazzi	2019	Here main aim is to provide flexible and extensive digital platform for Smart Homes is presented. The HABITAT (Home Assistance Based on the IOT for the autonomy of Everybody)is for assisting elderly and non-self-sufficient people in smart homes and embedding them in everyday life objects, thus reducing

				the expenses for healthcare due to the lower need for personal assistance.
3.	Development of an IoT-Based Health Promotion	Sathish Kumar, Nivedha, Anitha, Jayaprakash.	2020	In this paper they use the software which contains three core elements We-Watch, Service Board for We-Care and the cloud application. The We-Watch is composed of a discreet small bracelet which the elderly individual uses. This is responsible for tracking and gathering data from the available sensors. The solution alerts family members or appointed caretakers about sudden daily shifts
4.	Development of an IoT-Based Health Promotion System for Seniors	Chia-Hui Liu , Jih-Fu Tu .	2020	Using wireless technology combined with physiological measurement techniques and home care equipment can assist elderly individuals to promote health and the health care service at home. In this research the system is divided into three subsystems the IoT-based ,the context awareness-based service subsystem, and the elderly nutrition diet and health promotion subsystem

Table 2.1. Existing solutions

2.2.References

- <https://ieeexplore.ieee.org/Xplore/home.jsp>
- <https://link.springer.com/>

2.3.Problem Statement Definition

Our problem statement is to remind the elderly persons to take their medicines on correct time and monitor them.

Customer Problem Statement:

I am	I'm trying to	But	Because	Which makes me feel
A Caretaker	To monitor the elder people whether they take their medicine at the correct time	The elder people forget to take their medicine at the correct time	It requires medical remainder system for the elder people	It is difficult for the user to monitor the patient around the clock

3.IDEATION & PROPOSED SOLUTION

3.1.Empathy Map Canvas

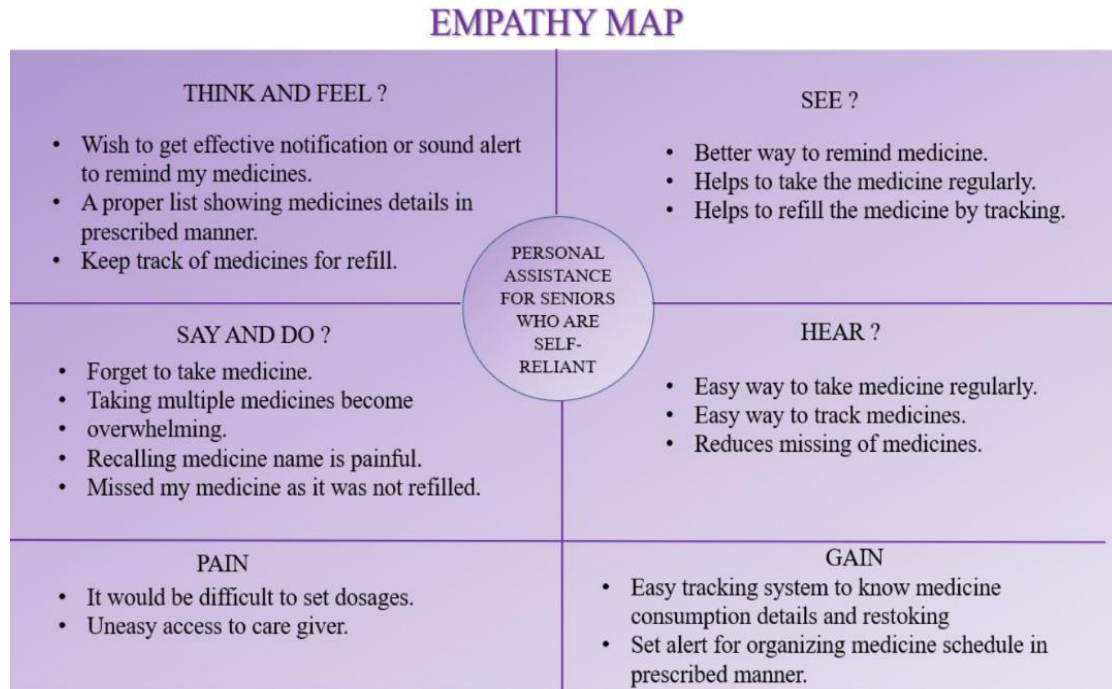


Fig 3.1. Empathy map

3.2.Ideation & Brainstorming

Step-1: Team Gathering, Collaboration and Select the Problem Statement

We have followed the first step of brainstorming, we have discussed as a team to decide a problem statement .As per the guideline the following is done

- TEAM GATHERING
- COLLABORATION
- DECIDING THE PROBLEM STATEMENT



Problem Statement:

A caretaker\doctor trying to monitor the elder people whether they take their medicine at the correct time. But the elder people forget to take their medicine at the correct time, so that it is difficult for the user to monitor the patient around the clock. To avoid this it requires medical remainder system for elders and notify the user with voice commands.

Step-2: Brainstorm, Idea Listing and Grouping

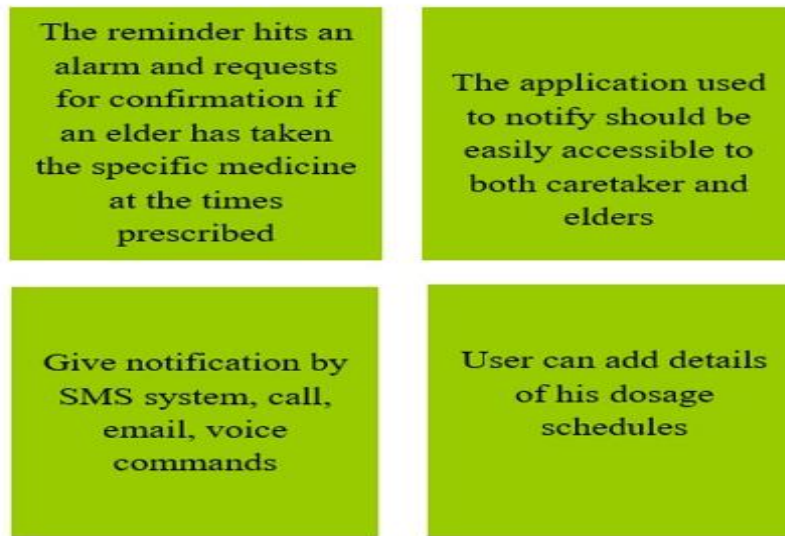
NISHA C		FATHIMA D	
The reminder hits an alarm and requests for confirmation if an elder has taken the specific medicine at the times prescribed	If not responded to, or responded negatively, the app sends a notification to the user	Notifying when the medicine box is empty	Alert when the medicine is expiring
Help to avoid unnecessary risk and serious illness	User can enter medicine and desired time manually in the application	The monitoring details can be shared to doctors	The application used to notify should be easily accessible to both caretaker and elders
HARNI V		SANGEETHA M	
Uphold an appropriate schedule to maintain body health	Give notification by SMS system, call, email, voice commands	User can add details of his dosage schedules	Feature which helps the person to track progress in health vitals such as sleep, weight, blood pressure, sugar level etc.
Medication histories of an elders can be monitored to give further medical tips	Reminder for the elderly to stay on schedule for their daily medicines	Alert sound or voice commands should be understandable and safe for an elders	The application should be customizable to user (care taker)

Table 3.1. Ideas of all the teammates

Step-3: Idea Prioritization:



Fig 3.3. Prioritization matrix



3.3.Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	➤ Sometimes elderly people forget to take medicines at correct time ,to avoid this medicine remainder system is developed.
2.	Idea / Solution description	➤ In order to monitor elders we proposed medicine reminder system. An app is built for the user (caretaker) which enables them to set the desired time and medicine. ➤ These details will be stored in the Cloud DB. If the medicine time arrives the web application will send the medicine name to the IoT Device through the IoT platform. The device will receive the medicine name and notify the user with voice commands.

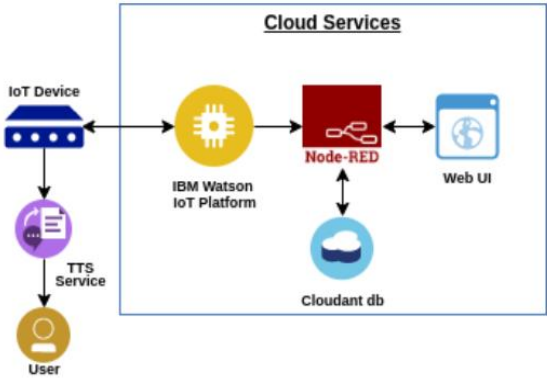
3.	Novelty / Uniqueness	<ul style="list-style-type: none"> ➤ Digital Voice assistant technology ➤ Notification System.
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"> ➤ This will be helpful for the caretakers/doctors for monitoring the health conditions of elders and also this creates positive impact on seniors health.
5.	Business Model (Revenue Model)	 <pre> graph LR IoT[IoT Device] <--> Watson[IBM Watson IoT Platform] Watson <--> NodeRED[Node-RED] NodeRED <--> WebUI[Web UI] NodeRED <--> Cloudant[Cloudant db] IoT --> TTS[TTS Service] TTS --> User((User)) </pre>
6.	Scalability of the Solution	<ul style="list-style-type: none"> ➤ In future increase the number of users, Maintains best possible user experience. ➤ Helps the elders to take medicines regularly ➤ And also helps to refill the medicines.

Table 3.2. Proposed solution

3.4.Problem Solution fit

1. CUSTOMER SEGMENT(S) <small>Who is your customer?</small> <p>The elder peoples those who are forget to take medicines on time</p>	CS	6. CUSTOMER CONSTRAINTS <small>What constraints prevent your customers from taking action or limit their choices of solutions?</small> <p>Being aged and having less knowledge on technology are the major constraints.</p>	CC	5. AVAILABLE SOLUTIONS <small>Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do</small> <p>The existing solutions are some medicine reminder system which only gives alarm, but our app remind the patients with medicine name and amount of intakings & also giving alert to caretakers are the pros of our solution</p>	AS
2. JOBS-TO-BE-DONE / PROBLEMS <small>Which jobs-to-be-done (or problems) do you address for your customers?</small> <p>An app should remind medicine on time through voice commands. When patients forget to take medicine, It alerts the caretakers of those patients</p>	J&P	9. PROBLEM ROOT CAUSE <small>What is the real reason that this problem exists? What is the back story behind the need to do this job?</small> <p>The major reason behind the problem is aging and the forgetness of the patients to take medicines on time. It makes aged patients' body condition weak and it should be addressed.</p>	RC	7. BEHAVIOUR <small>What does your customer do to address the problem and get the job done?</small> <p>The patients need to update their medicines details & the time of intaking to their caretakers</p>	BE
3. TRIGGERS <small>What triggers customers to act? I.e seeing their neighbour installing</small> <p>Seeing their friends using this app and getting recovery easily. Doctors and caretakers advise using the app</p>	TR	10. YOUR SOLUTION <small>If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations.</small> <p>It is the usual thing for older patients to forget to take their medicines on time. so our solution is to make an app that reminds elders to take medication on time.</p>	SL	8. CHANNELS of BEHAVIOUR <small>8.1 ONLINE</small> <small>What kind of actions do customers take online? Extract online channels from #7</small> <p>Through online, patients & caretakers can see the medicines and the time of intaking.</p>	CH
4. EMOTIONS: BEFORE / AFTER <small>How do customers feel when they face a problem or a job and afterwards?</small> <p>Before, patients feel worried and stressed about their health. But after using this app, they feel productive & happy about their health</p>	EM			<small>8.2 OFFLINE</small> <small>What kind of actions do customers take offline? Extract offline channels from #7</small> <p>Through offline, patients can get alerting sound with medicine name, that helps patients to take medicines on time.</p>	

Fig 3.5. Problem fit

4.REQUIREMENT ANALYSIS

4.1.Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration can be done through Gmail or either using phone number.
FR-2	User Confirmation	Confirmation via Email or through Message.
FR-3	User Login (mobile app)	Login with registered mobile number and password
FR-4	User Medical Information	In the app user can enter the medicine details with date. Then set the time in the app for alarm remainder.

Table 4.1. Functional requirements

4.2.Non-Functional requirements

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	It is used to remind the medicine on time and it alerts the users through voice commands.
NFR-2	Security	The login information should not be

		accessed by any other users than respective users and caretakers.
NFR-3	Reliability	It reminds on correct time. The user data will get updated and examined after certain period of time.
NFR-4	Performance	The voice message will be delivered accurately to the given time.
NFR-5	Availability	It can be used by registered user from any place.
NFR-6	Scalability	It is easily adaptable. The device is compatible and portable.

2.

Table 4.2. Non – functional requirements

5.PROJECT DESIGN

5.1.Data Flow Diagram

DATA FLOW DIAGRAM :

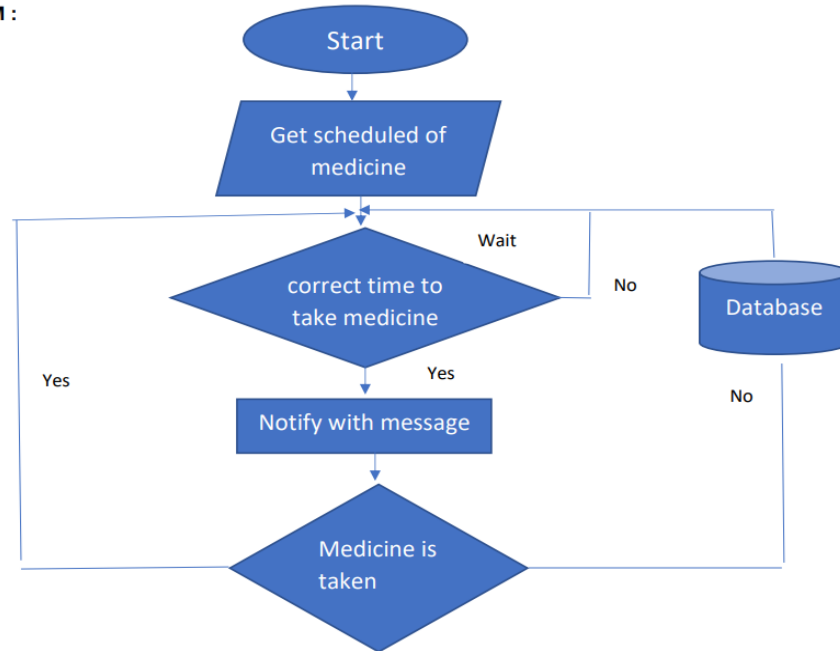


Fig 5.1. Dataflow diagram

5.2.Solution & Technical Architecture

Solution Architecture:

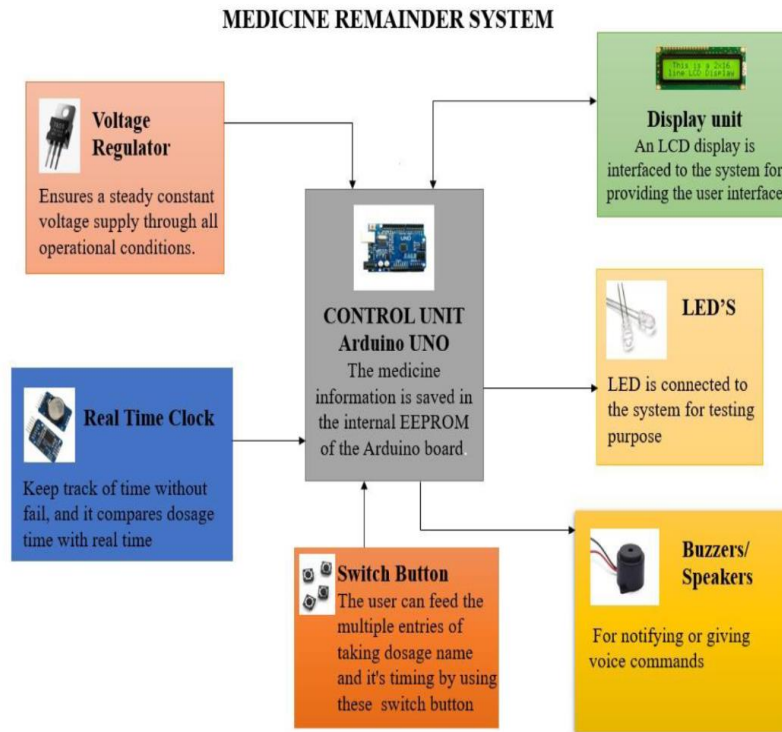


Fig 5.2. Solution architecture

Technical stack:

S. No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the	IBM Watson TTS

		application	service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	Infrastructure (Server /Cloud)	Application Deployment on Local System / Cloud	Local, Cloud Foundry, Kubernetes, etc.

Table 5.1. Components and technologies

S. No	Characteristics	Description	Technology
1.	Scalable Architecture	Can be used at <ul style="list-style-type: none"> ➤ Hospitals ➤ Old age Homes ➤ In working people homes where it will be used for elders 	IOT(Internet Of Things)
2.	Availability	To remind to take medicine with uninterrupted services, we have to implement in distributed service	IBM Cloud

3.	Performance	If it was implemented, it is given with medicine name and desired time	IBM Cloud DB,RTC DS1307 Module
----	-------------	---	-----------------------------------

Table 5.2. Application Characteristics

5.3.User Stories

User Type	Functional Requirement (Epic)	User Story	User Story / Task	Acceptance criteria	Priority	Release
Customer (Senior user)	Caretaker	USN-1	As a user, I want to take medicines on time and monitor my health.	I want to take medicines on time.	High	Sprint-1
Customer (Diabetes Patient)	Smart medicine box	USN-2	As a user, I want to take my tablets on time by voice command.	I want to take my tablet on time by voice command.	High	Sprint-1
Customer (disabled person)	Smart medicine box	USN-3	As a user, I need to take my medicine in nearby places with	I need to take my medicine in nearby places with	Medium	Sprint-3

			notification.	notification.		
--	--	--	---------------	---------------	--	--

Table 5.3. User stories

6.PROJECT PLANNING & SCHEDULING

6.1.Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation:

Sprint	Functional Requirement	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Medication Details	USN-1	As a user, I take the medicines that shouldintake daily and put them in the medicine box by the day order separetely	10	Low	Nisha C Fathima D Harni V Sangeetha M
Sprint-1		USN-2	As a user, by setting the particular time, medicine names the elders can be notified appropriately.	10	High	Nisha C Fathima D Harni V Sangeetha M
Sprint-2	Set Time	USN-3	As a user,I set the desired time,medicine	10	High	Nisha C Fathima D Harni V

			names and the dosages in the medicine remainder system.			Sangeetha M
Sprint-2		USN-4	The above setted details were stored in IBM Cloudant DB,which helps to notify the people on time	10	Medium	Nisha C Fathima D Harni V Sangeetha M
Sprint-3	Notification	USN-5	The time,medicine names and the dosagesare setted and these details were stored in IBM Cloudant DB. When the medicine time arrives the web application send the medicine name to the IOT device Arduino.	10	High	Nisha C Fathima D Harni V Sangeetha M
Sprint-3		USN-6	By that device elders can be notified and also caretaker can receive the conformation	10	High	Nisha C Fathima D Harni V Sangeetha M
Sprint 4	Monitoring	USN-7	In IBM	20	High	Nisha C

	andTracking healthStatus		Cloudant DB the time,medicine names and the dosages are stored.By which the elders medication and health records were keep on tracking			Fathima D Harni V Sangeetha M
--	-----------------------------	--	---	--	--	--

Table 6.1. Estimation chart

Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Point s	Duration	Sprint Start Date	Sprint End Date (Plann ed)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Table 6.2. Burndown chart

6.2.Sprint Delivery Schedule

S.N O	ACTIVITY TITLE	ACTIVITY DESCRIPTION	DURATION
1	Literature Survey	Literature Survey on the selected project and gathering information byreferring the technical papers,research papers etc.	10 October 2022
2	Empathy Map	Prepare Empathy Map to know the user's thought,pain and gain.Also forpreparing list of problem statements	11 October 2022
3	Ideation and Brainstorming	List the ideas by brainstorming sessions.Prioritize the ideas that isfeasible and optimal.	13 October 2022
4	Problem Solution Fit	Prepare the Problem Solution Fit which describes the solution for the problem statement.	14 October 2022
5	Proposed Solution	Prepare the proposed solution document which includes the feasibility of idea,novelty,business model and scalability of solution.	17 October 2022

6	Solution Architecture	Defines the process of developing solution based on predefined process.	18 October 2022
7	Customer Journey	Prepare Customer Problem Statement by customer interactions. From which optimal solutions can be obtained.	20 October 2022
8	Functional Requirements	Prepare the Functional Requirements which specifies the requirements.	21 October 2022
9	Data Flow Diagram	Draw the data flow diagram based on the problem statements.	22 October 2022
10	Technical Architecture	Prepare the Technical Architecture which describes the working.	24 October 2022
11	Prepare Milestone and Activity List	Prepare the milestones & Activity list of the project.	4 November 2022
12	Project Development and Delivery of Sprint 1,2,3,4	Develop and submit the developed code by testing it.	IN PROGRESS

Table 6.3. Sprint delivery schedule

6.3.Reports from JIRA

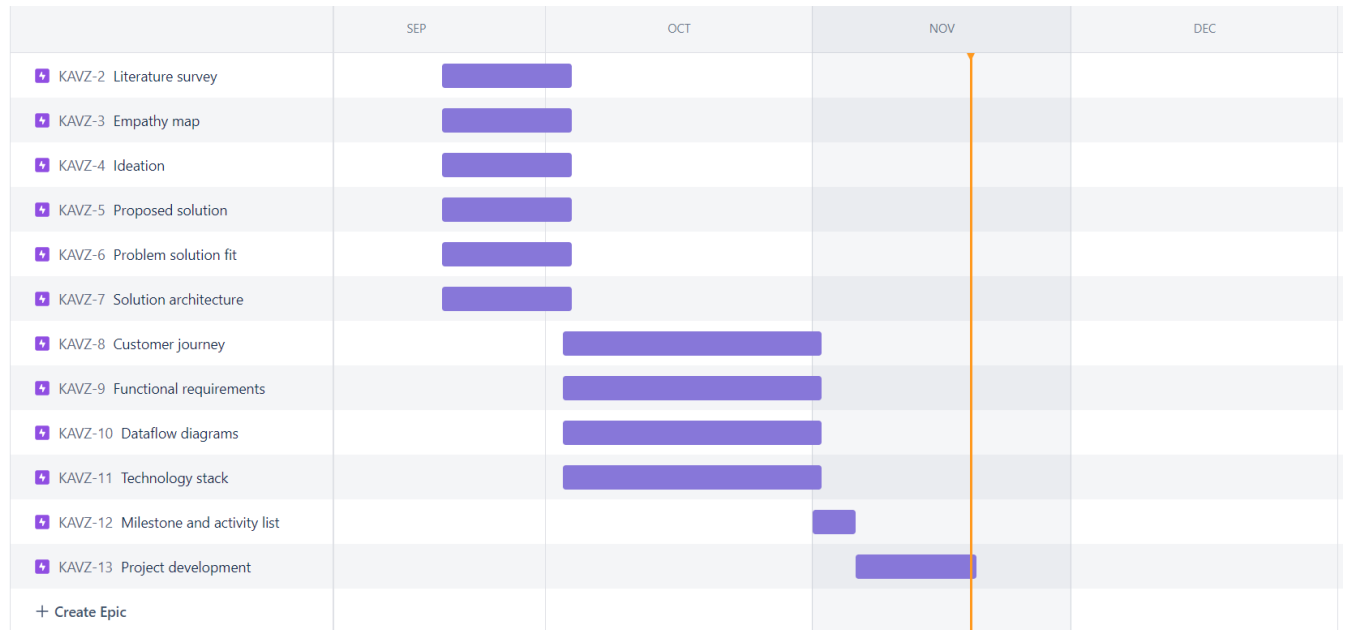


Fig 6.1. JIRA report

7.CODING AND SOLUTION:

7.1 FEATURE 1

MEDICINE.PY (PYTHON SCRIPT TO RECEIVE DATA FROM NODE-RED)

```
import time
#import ibmiotf.application
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
import ibmiotf.device
import pygame
pygame.init()

config={
    "org":"k54tgp",
    "type" : "abcd",
    "id":"123",
    "auth-method":"token",
    "auth-token":"123456789"
}

url="https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/3337f7ab-
eb47-4558-8344-cf4e0e94bee7"
api="ri9CfHaOitFC4s_8nN2fYhKmdVBku6yqzeedtTcCOQf1"
client= ibmiotf.device.Client (config)
client.connect()
auth=IAMAuthenticator(api)
tts=TextToSpeechV1(authenticator=auth)
tts.set_service_url(url)
```

```

def myCommandCallback (cmd):
    a=cmd.data
    instruction="Please Take following Medicine. "
    c=1
    if len(a["command"])==0:
        pass
    else:
        client.disconnect()
        client.connect()
        for i in a["command"]:
            instruction+=str(c)+". "
            instruction+=i
            instruction+=" ".
            c+=1
        with open("./speech.wav","wb") as audio_file:
            res=tts.synthesize(instruction,accept="audio/mp3",voice='en-US_AllisonExpressive').get_result()
            audio_file.write(res.content)
            play("speech.wav")

def play(a):
    p=pygame.mixer.Sound(a)
    pygame.mixer.Sound.play(p)
    time.sleep(20)
    pygame.mixer.Sound.play(p)
    time.sleep(20)

```

```
pygame.mixer.Sound.play(p)  
time.sleep(20)
```

```
while True:
```

```
    client.commandCallback = myCommandCallback
```

```
client.disconnect()
```

FEATURES:

The first feature is the use of web application

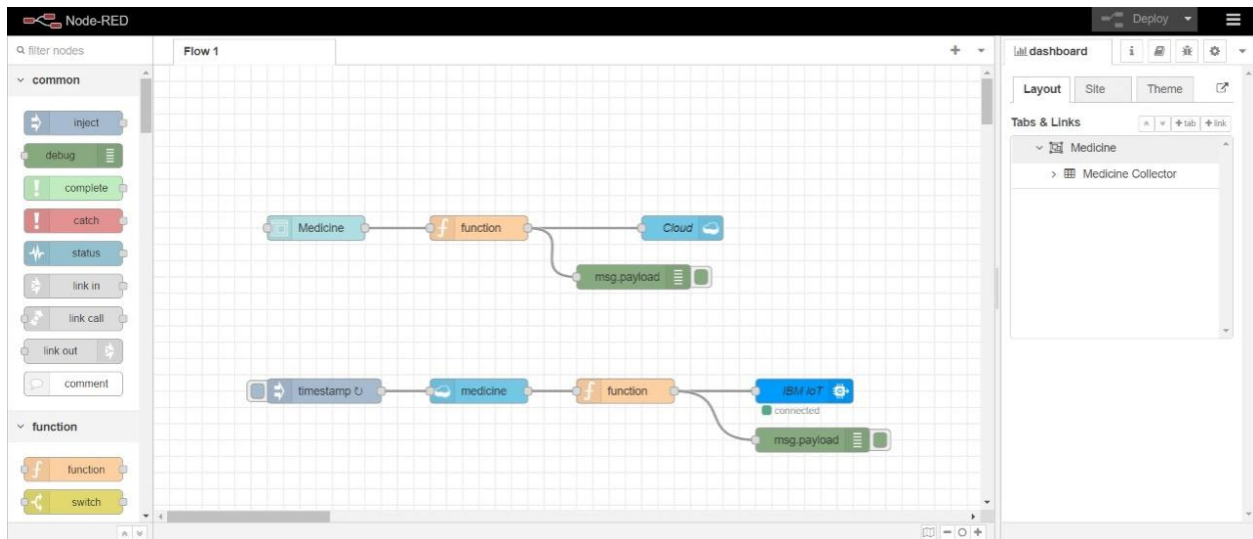
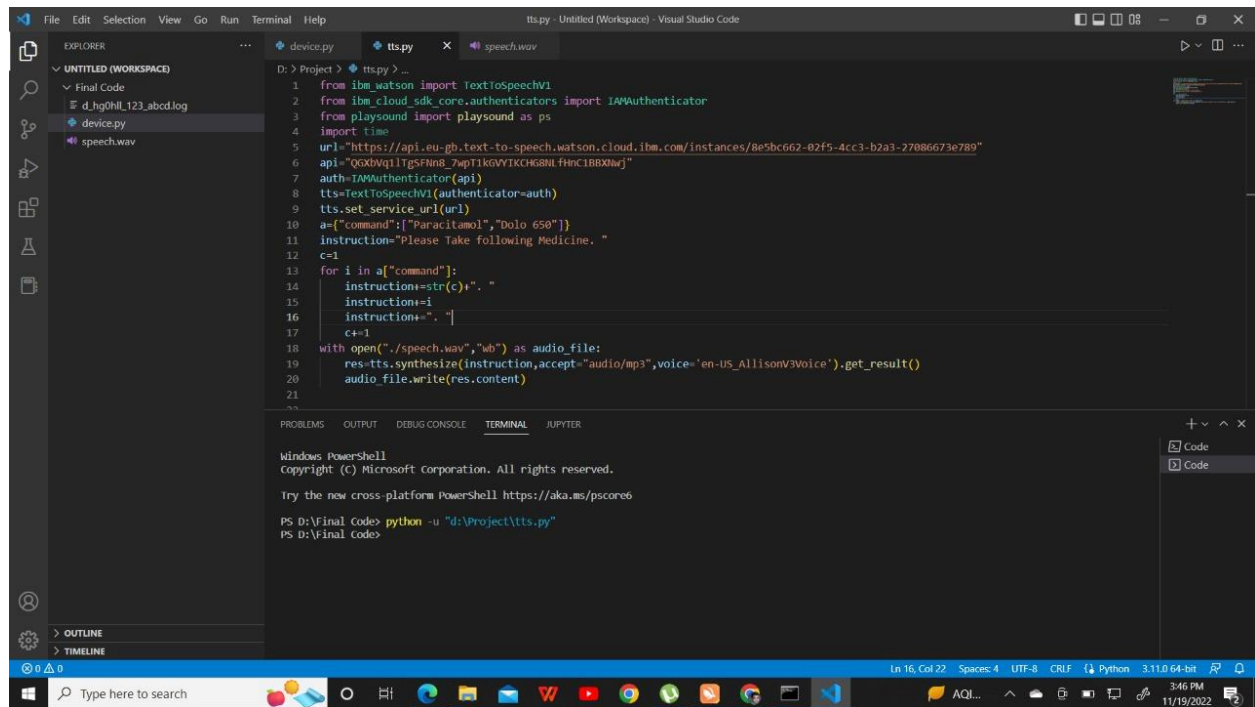


Fig. 7.1. Visual results in node red dashboard

7.2 TTS.PY (PROGRAM FOR ACCESSING TEXT TO SPEECH SERVICE) FEATURE 2:



The screenshot shows the Visual Studio Code editor with a file named `tts.py` open. The code imports `TextToSpeechV1` from `ibm_watson`, `IAMAuthenticator` from `ibm_cloud_sdk_core.authenticators`, and `playsound` from `playsound`. It sets up an API key, authenticator, and service URL. The main logic is a loop that processes a list of commands, concatenating them into a single instruction string. Finally, it uses `res-tts.synthesize` to generate audio and writes it to a file named `speech.wav`.

```
1 from ibm_watson import TextToSpeechV1
2 from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
3 from playsound import playsound as ps
4 import time
5 url="https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/8e5bc662-02f5-4cc3-b2a3-27086673e789"
6 api="QcXbWq1lrgSPHn8_7wpT3kgVYIKCH8NLfhnc18BXWj"
7 auth=IAMAuthenticator(api)
8 tts=TextToSpeechV1(authenticator=auth)
9 tts.set_service_url(url)
10 a=["command":["Paracetamol","Dolo 650"]]
11 instruction="Please Take following Medicine. "
12 c=1
13 for i in a["command"]:
14     instruction+=str(c)+". "
15     instruction+=i
16     instruction+=" "
17     c+=1
18 with open("./speech.wav","wb") as audio_file:
19     res=tts.synthesize(instruction,accept="audio/mp3",voice="en-US_AllisonV3Voice").get_result()
20     audio_file.write(res.content)
21
```

The terminal window shows the command `python -u "d:\Project\tts.py"` being executed, and the output indicates that the program is running successfully.

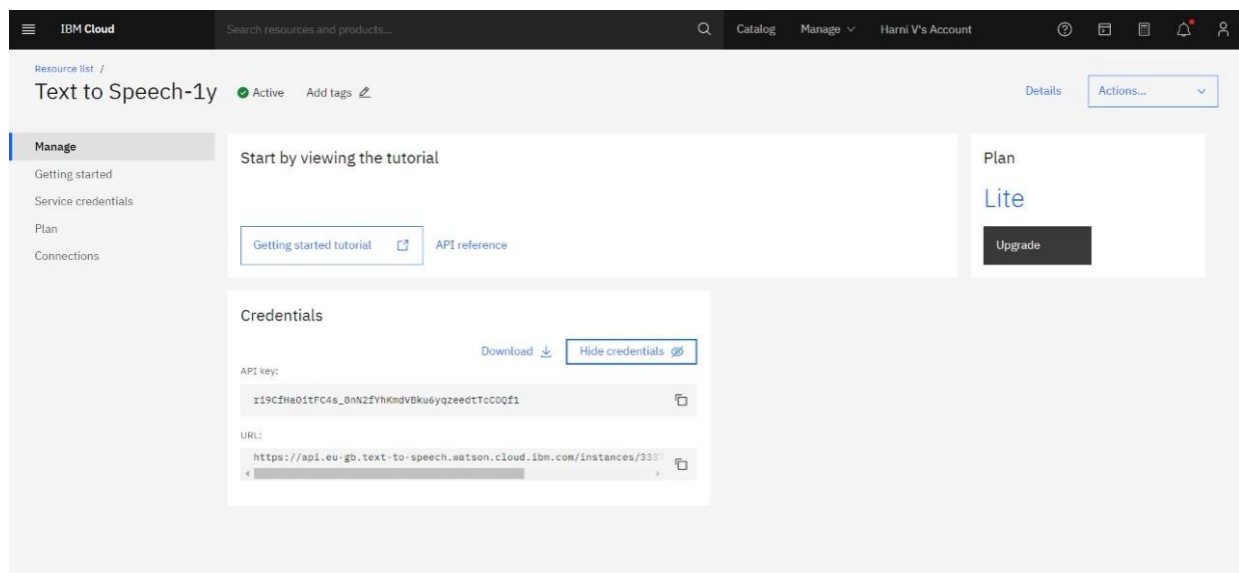


Fig. 7.2. Text to speech

7.3.DATABASE SCHEMA

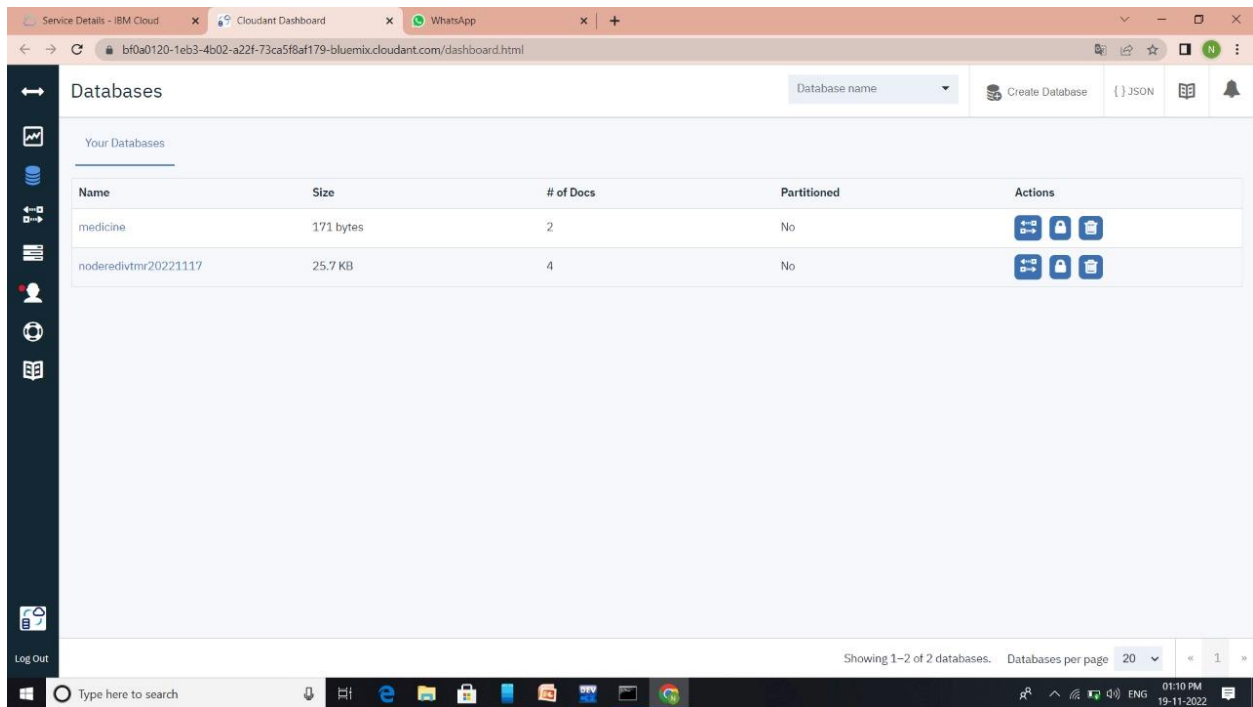
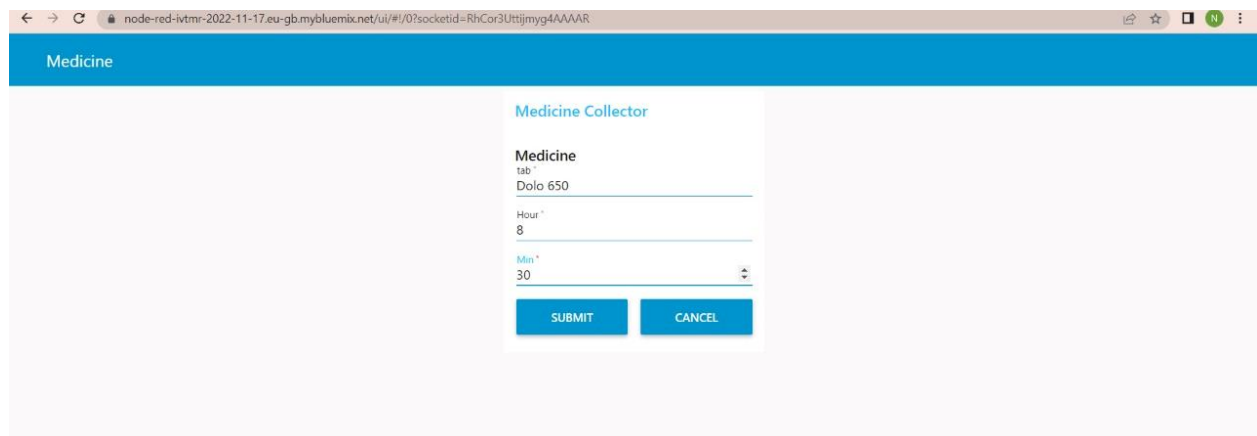
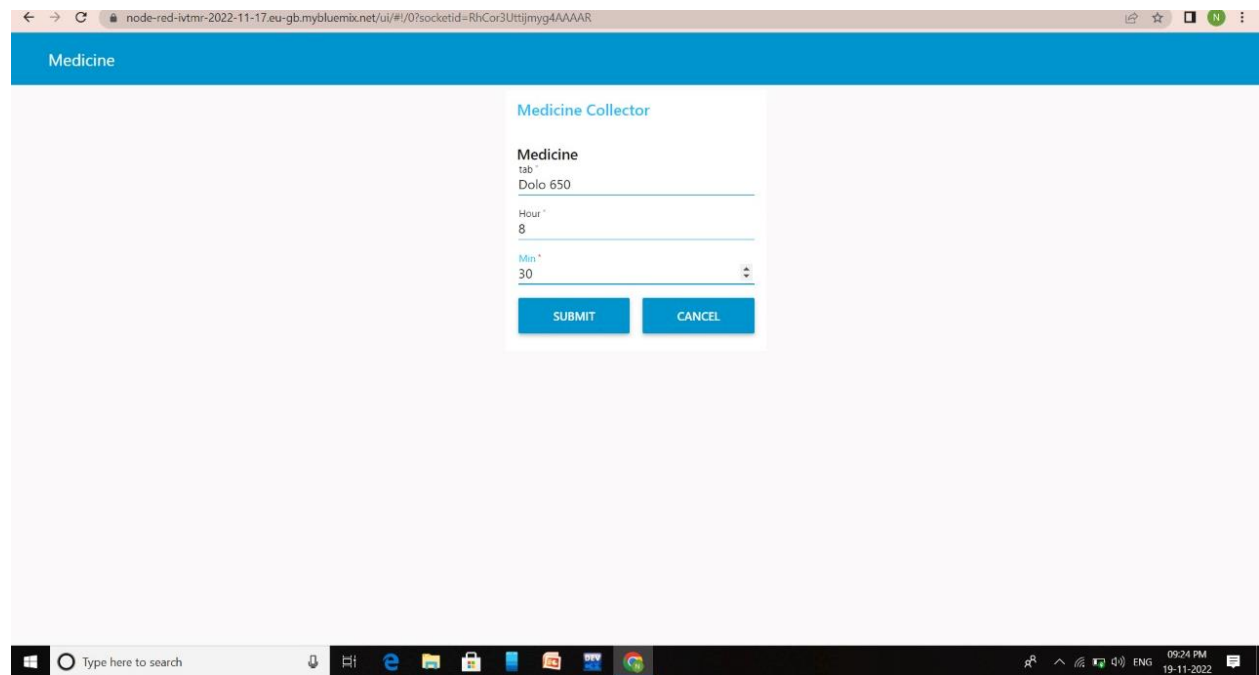


Fig. 7.3. Database

WEB USER INTERFACE:



APPLICATION :



The screenshot displays a web browser window with the address bar showing a URL from a Node-RED instance. The page has a blue header with the word 'Medicine'. The main content area is light gray and contains a white 'Medicine Collector' form. The form has a title 'Medicine' and three input fields: 'tab' with the value 'Dolo 650', 'Hour' with the value '8', and 'Min' with the value '30'. At the bottom of the form are two blue buttons labeled 'SUBMIT' and 'CANCEL'. The Windows taskbar is visible at the bottom of the screen, showing the search bar, taskbar icons, and system tray with the time '09:24 PM' and date '19-11-2022'.

Medicine

Medicine Collector

Medicine

tab
Dolo 650

Hour
8

Min
30

SUBMIT CANCEL

Voice commands are also been generated

8.TESTING AND USE CASES:

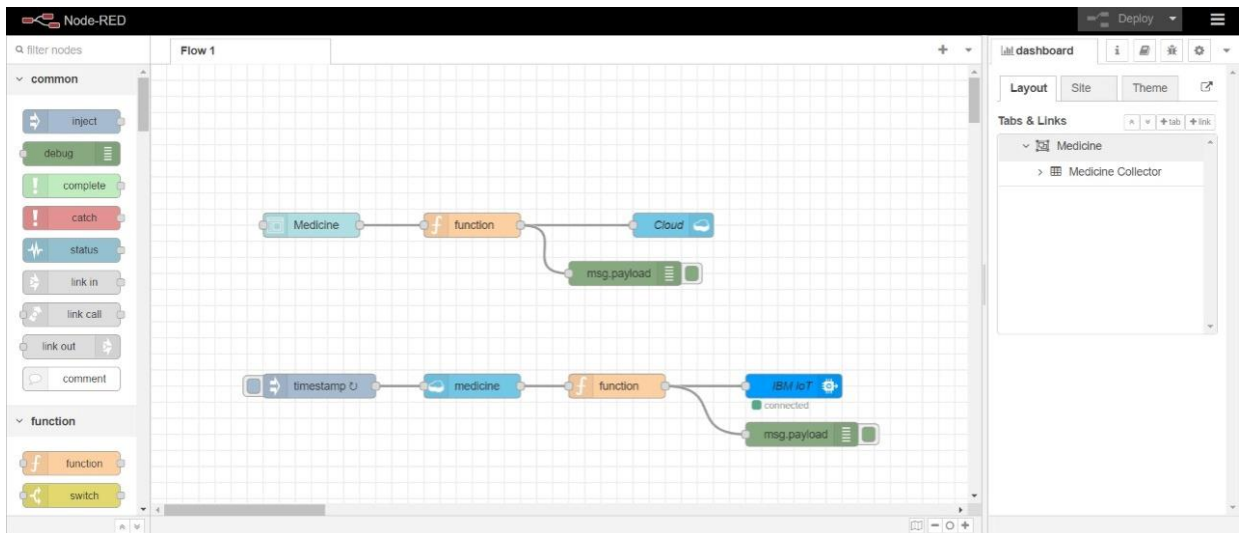
8.1.TEST CASES

S.No	MEDICINE NAME	TIME (H:M)	Expected output	Actual Output
1	Dolo	11:20	Alarm is set	Alarm is set
2	Flucold	12:00	Alarm is set	Alarm is set
3	Wikoryl	23:54	Alarm is set	Alarm is set
4	vicks	1:20	Alarm is set	Alarm is set

8.2.USER ACCEPTANCE TESTING

- The web application will be served to the user for setting reminder.
- The user can set alarm or reminder by running the application.
- Based on the medicine name and time entered in the app , an alarm is set and a voicecommand is played at the entered time.

9.RESULT:



Medicine name and time can be entered to get alert

The screenshot shows the 'Edit form node' dialog for the 'Medicine Collector' form. The form has three input fields: 'tab', 'Hour', and 'Min'. The 'tab' field is a text input, while 'Hour' and 'Min' are number inputs. The 'submit' button is visible at the bottom. The right sidebar shows the 'dashboard' with the 'Medicine' tab and 'Medicine Collector' sub-tab.

Label	Name	Type	Required	UIRows	Remove
tab	tab	Text	<input checked="" type="checkbox"/>		
Hour	Hour	Number	<input checked="" type="checkbox"/>		
Min	Min	Number	<input checked="" type="checkbox"/>		

IBM Watson IoT Platform

Generate API Key

or by using the API. For more information about adding API keys, see [API key connection](#).

Key	Description	Role	Expires
a-k54tgp-bix5tgsfgd		Standard Application	

1 result

API Key Information

Key	a-k54tgp-bix5tgsfgd	Last Edited By	harni@student.tce.edu
Description	-	Expires	Never
Date Added	17 Nov 2022 16:05		
Last Update	17 Nov 2022 16:05		

IBM Watson IoT Platform

Add Device

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
123	Disconnected	abcd	Device	17 Nov 2022 16:04	

Identity

Device ID	123
Device Type	abcd
Date Added	17 Nov 2022 16:04
Added By	harni@student.tce.edu
Connection Status	Disconnected Last Connected: 17 Nov 2022 16:36 Client Address: 42.106.177.188 SecureToken Duration: 3 minutes Data Transferred: 11.2 KB

Items per page 50 | 1-1 of 1 item

1 of 1 page

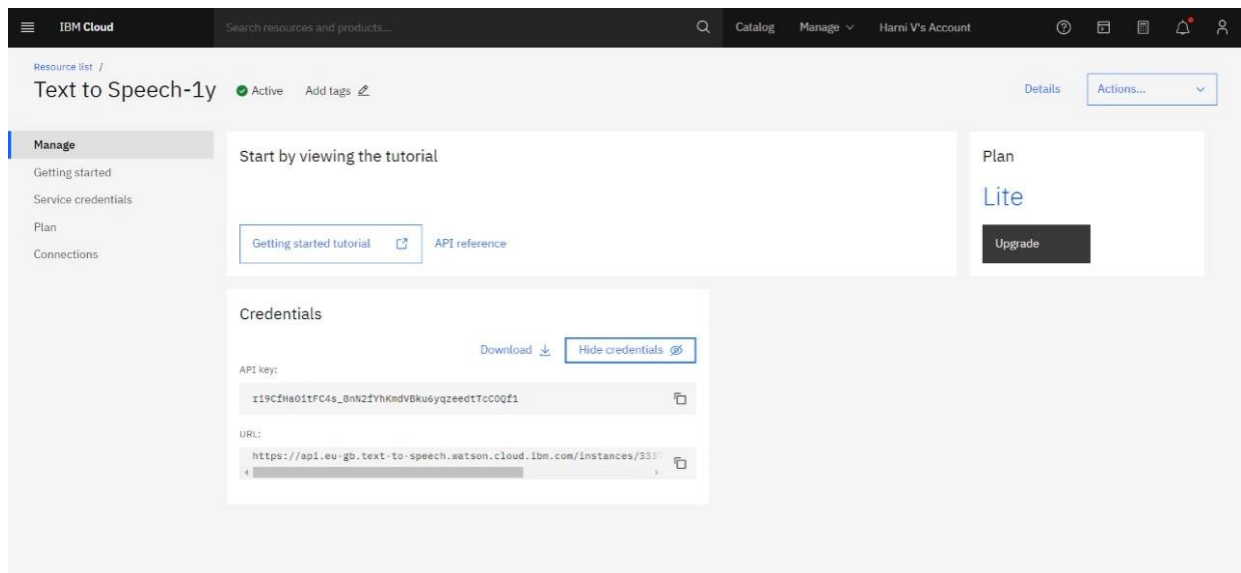


Fig 9.1. Results

10.ADVANTAGES AND DISADVANTAGES:

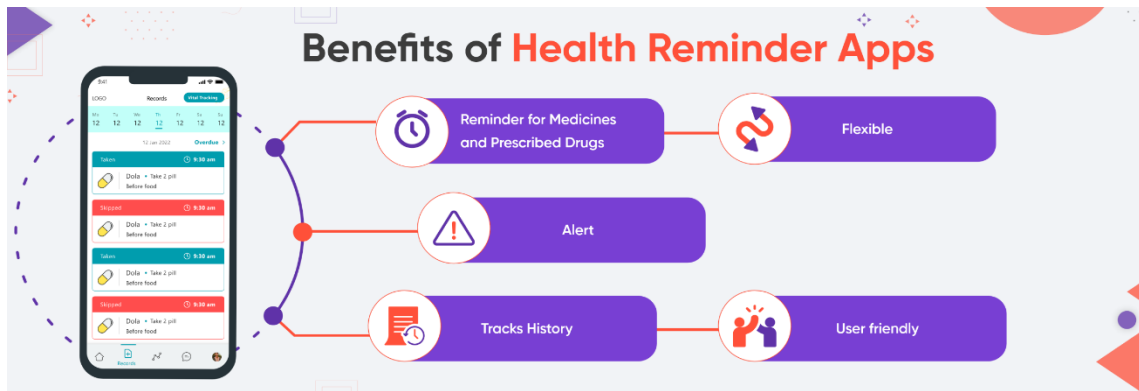


Fig. 10.1. Medicine Remainder Advantages

ADVANTAGES:

- Receiving remainders to take medications at correct time.
- Alarm is set at particular time as per the user requirement.
- Voice commands are received at particular time to take medications.
- No need of internet connection.

DISADVANTAGES:

- Possibility of run time problem when user enters two medicines at the same time..

11.CONCLUSION:

- The objective of this project was to design and develop a simple, reliable, efficient medicine remainder system that has a precise and quick notification mechanism. Appropriate services were used to make reminder alarm and voice commands .A step-by- step approach was followed in the design of the system. The design was carried out based on the study and analysis of existing similar systems and user perceptions. Several tests were conducted, and the results were analysed to ensure that the system produced the intended results. The system has been implemented and tested, showing satisfactory performance.

12.FUTURE SCOPE:



Fig. 12.1. Future scope

Health information:

The application helps family members to keep track of medicines.

Alert message:

If medicines are out of stock it gives alert message.

Order nearby:

We can order medicines in nearby pharmacy.

Users:

Single point to multi point user.

13.APPENDIX:

SOURCE CODE:

```
import time
#import ibmiotf.application
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
import ibmiotf.device
import pygame
pygame.init()

config={
    "org":"k54tgp",
    "type" : "abcd",
    "id":"123",
    "auth-method":"token",
    "auth-token":"123456789"
}

url="https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/3337f7ab-
eb47-4558-8344-cf4e0e94bee7"
api="ri9CfHaOitFC4s_8nN2fYhKmdVBku6yqzeedtTcCOQf1"
client= ibmiotf.device.Client (config)
client.connect()
auth=IAMAuthenticator(api)
tts=TextToSpeechV1(authenticator=auth)
tts.set_service_url(url)
```

```
def myCommandCallback (cmd):
    a=cmd.data
    instruction="Please Take following Medicine. "
    c=1
    if len(a["command"])==0:
        pass
    else:
        client.disconnect()
        client.connect()
        for i in a["command"]:
            instruction+=str(c)+". "
            instruction+=i
            instruction+=" ". "
            c+=1
        with open("./speech.wav","wb") as audio_file:
            res=tts.synthesize(instruction,accept="audio/mp3",voice='en-US_AllisonExpressive').get_result()
            audio_file.write(res.content)
            play("speech.wav")
```

```
def play(a):
    p=pygame.mixer.Sound(a)
    pygame.mixer.Sound.play(p)
    time.sleep(20)
    pygame.mixer.Sound.play(p)
    time.sleep(20)
    pygame.mixer.Sound.play(p)
    time.sleep(20)
```

```
while True:
```

```
    client.commandCallback = myCommandCallback
```

```
client.disconnect()
```

PROJECT DEMO LINK: