PROJECT BASED EXPERIENTIAL LEARNING PROGRAM (NALAIYA THIRAN)

PERSONAL ASSISTANCE FOR SENIORS WHO ARE SELF-RELIANT

A PROJECT REPORT

Submitted by

ARAVIND KM	**(111719106008)**
ASWATH NATHAN B	**(111719106014)**
CHANDRA SEKAR M.P	**(111719106024)**
GIDEON A	**(111719106041)**

TEAM ID: PNT2022TMID15932

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

R.M.K. ENGINEERING COLLEGE
(An Autonomous Institution)
R.S.M. Nagar, Kavaraipettai-601 206

NOVEMBER 2022

INTRODUCTION

PROJECT OVERVIEW:

The idea of digital world where different types of sensors and local processing connected to share information is used in many industries nowadays. There are various products which are developed based on these ideas. Healthcare industry is one where lot of improvements is taking place. Medicines play important role for prevention and cure for most of the diseases. Many Harmful and risky diseases can be cured through proper medication. This system consists of an IoT enabled medication reminder system and it gives timely alerts for the patients about their medication time. It alerts the patient to take medicines at proper time by providing audio-visual alert. The system helps to monitor whether patient has taken the medicine and it's healthcare data.

PROJECT PURPOSE:

The proposed model enables users to improve health related risks and reduce healthcare costs by reminding to take medicines at time, collecting, recording and analysing data in real time efficiently. With the help of this proposal the time of both patients and doctors are saved and doctors can also help in emergency scenario as much as possible. The proposed outcome of the project is to give proper and efficient medical services to patients by reminding them when to take medicines and collecting data of medicine. To monitor and remind we are using an IoT device which has the information about the medicine and at what time it has to be taken. It also gives an audio alert during the specified time for taking medicines. In this way, they can overcome the difficulties

LITERATURE SURVEY

Paper 1:

Prof. R.M Gawande, Ms. Shinde Gayatri, Ms. Supekar Pragati, Ms. Shevkar Vaishali, Mr Shevkar Akash.

Smart Medicine Box for Old Age People

Proper Medication is extremely problematic for the elderly patient who had problem in keeping track of their medicine. So to overcome this we made this Smart Medicine Box which keep tracks of the dosage and duration between each consumption. The common reasons are listed here Poor Eyesight, less remembrance due to which errors such as mis dosage since the elderly finds it troublesome to read the instruction on the medicine case and identifying the right dosage of the medicine. Hence, this Smart Medicine Box will track their medication and inform patient to take right dosage of right medicine at the right time.

Paper 2:

Priyanka J. Solanke1, Scientist. K. Lakshman2

Smart Medication & Monitoring System for Secure Health using IoT

The elderly people and the people victims of chronicle diseases who need to take the medicines timely without missing are suffering from dementia, which is forgetting things in their daily routine. Considering this situation study has been done in this. Reminding the scheduled medicine, remote monitoring and update new medicine Consumption data of patients, which can be done by prescriber through IOT. We need to be in advice of Doctor who tells us to take desired pills in desired way so that patients face problems like forgetting pills to take at right time and also when Doctor changes the prescription of medicine patients have to remember the new schedule of medicine

Paper 3:

Mohammad Alhaj, Abdullah Nizar, Shahd Al-Hatem, Athraa Leekha

A case study of medication reminder system

An important topic for treating the elderly who often take multiple medications simultaneously to treat different conditions and symptoms. Medications usually need to be taken in specific doses at set intervals. Missing doses or timing doses incorrectly can cause medical complications. Medication management can include everything from using devices that issue reminders to patients to take their medications to filling pill cases for patients and marking the lid of each compartment to indicate when the contents need to be taken. In this paper, we present a case study of medication reminder system that helps to alert patients who forget to take their medicines at prescribed time. We used HW/SW Codesign approach to allow the hardware and the software of the system designed and implemented in parallel and make sure that the non-functional properties are met.

Paper 4:

Vikas Anna Mali.

IoT Based Intelligent Medicine Box with Assistance

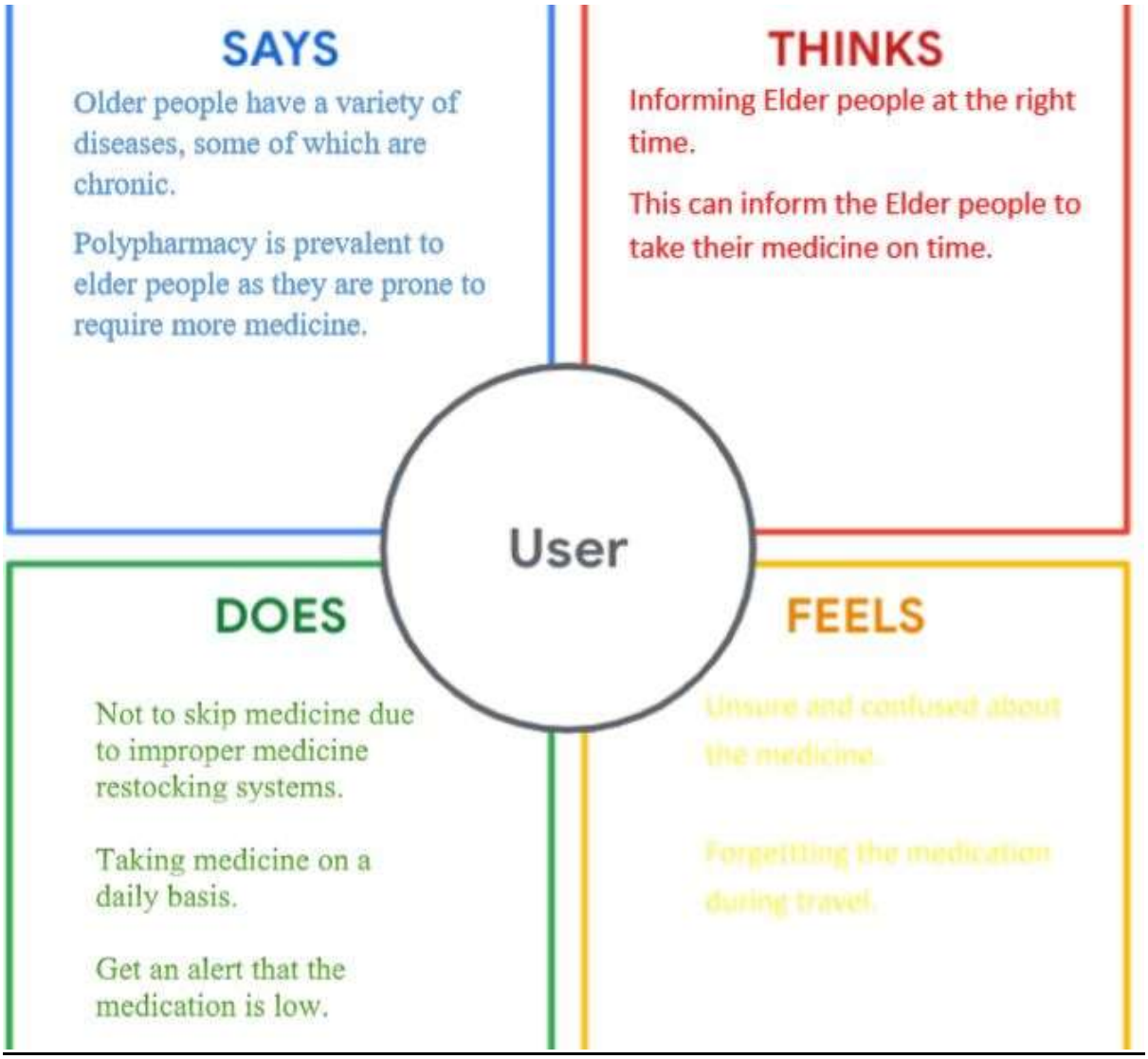
In this paper, the intelligent medicine box has been designed. This medicine box stores the medicines and which alerts and remind to patients to take the medicines time to time. So that it avoids the health hazards due to negligence of not taking medicines time to time. Also in this paper different health parameters such as pulse rate, blood pressure and temperature and ECG of heart are measured. These parameters are sent to firebase IoT. Doctors can monitor health parameters on firebase website and accordingly decisions can be taken about patient treatment.

PROBLEM STATEMENT:

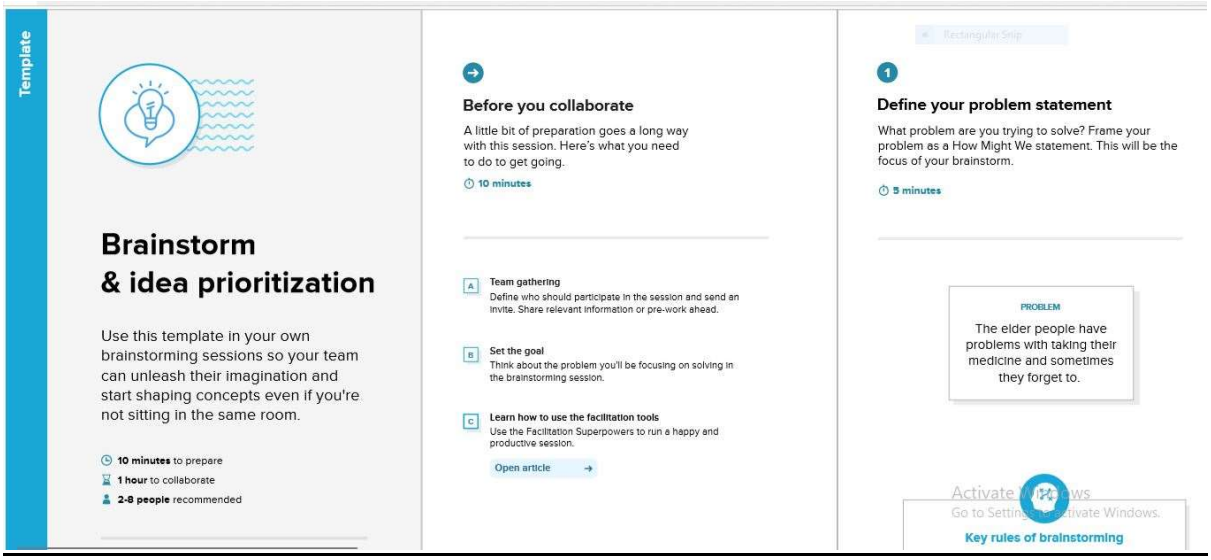
- Sometimes elderly people forget to take their medicine at the correct time.
- They also forget which medicine He / She should take at that particular time.
- And it is difficult for doctors/caretakers to monitor the patients around the clock. To avoid this problem, this medicine reminder system is developed.
- 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.

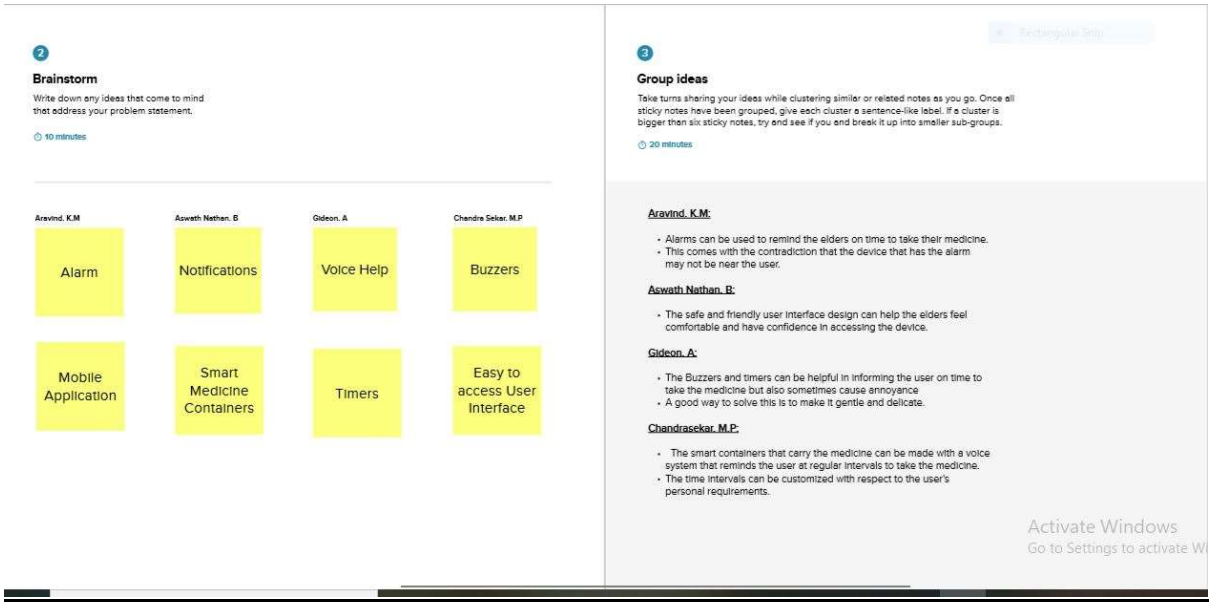
IDEATION & PROPOSED SOLUTION

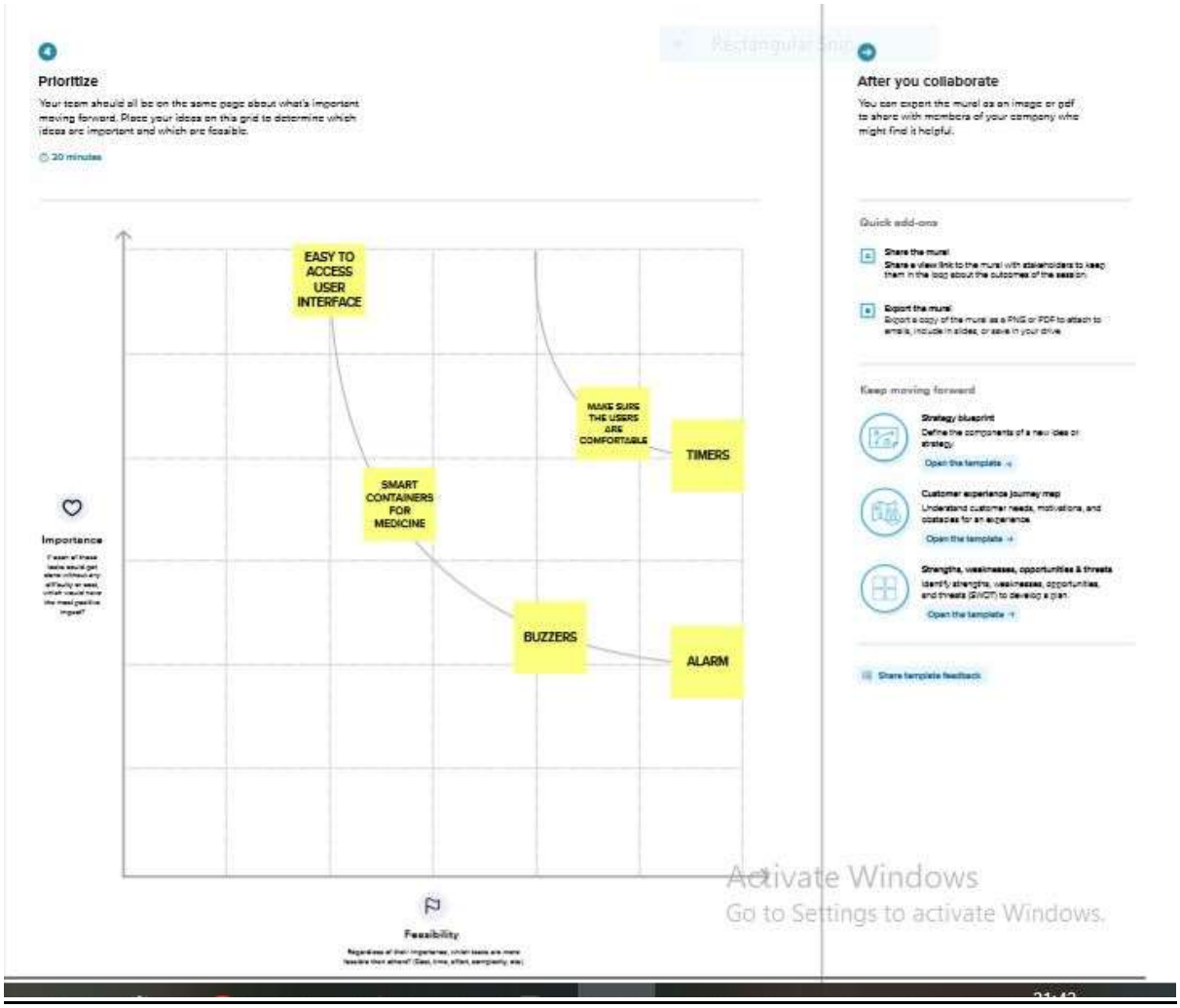
EMPATHY MAP:



BRAINSTORMING:







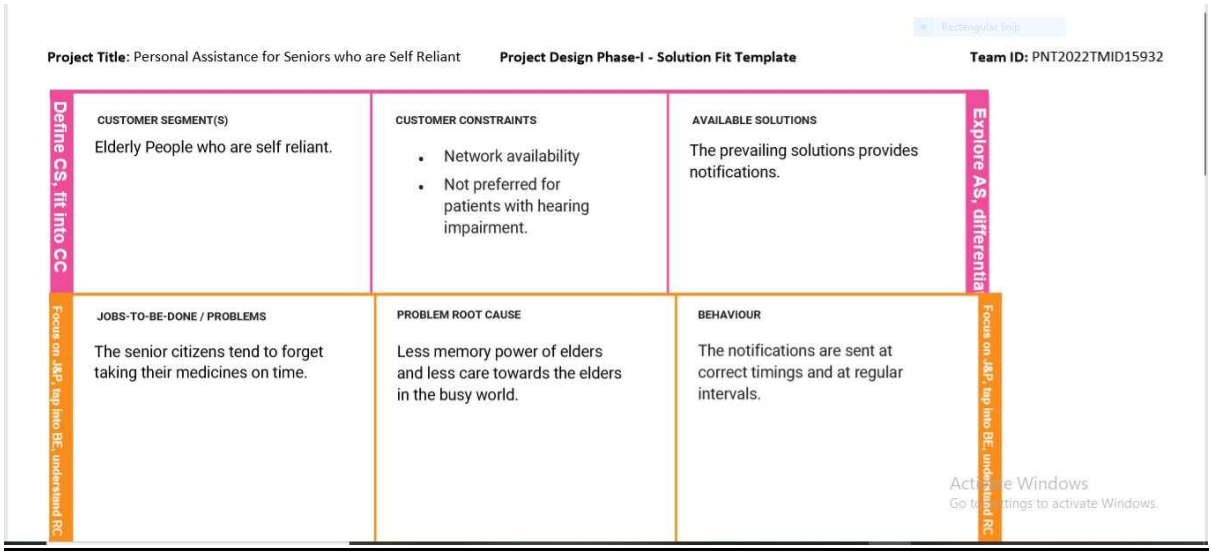
BRAINSTORM FILE LINK:

https://drive.google.com/file/d/1o5g_O7DcxwgpRvrS4eXrFVFQPpElOwnF/view?usp=sharing

PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Elderly people fail to take medicine regularly due to their health problems.
2.	Idea / Solution description	To remind the users to take medicines on time using voice command and notification using IoT device.
3.	Novelty / Uniqueness	Notifications to remind the intake of medicines at customizable time intervals.
4.	Social Impact / Customer Satisfaction	The users don't need to depend on anyone. They don't need to remember the medicines all the time.
5.	Business Model (Revenue Model)	The solution is the need of the hour in this fast-paced world.
6.	Scalability of the Solution	User – friendly interface and less man power involved in reminding the patients about medicines.

PROPOSED SOLUTION FIT:



Project Title: Personal Assistance for Seniors who are Self Reliant **Project Design Phase-I - Solution Fit Template** **Team ID:** PNT2022TMID15932

Define CS, fit into CC

CUSTOMER SEGMENT(S)

Elderly People who are self reliant.

CUSTOMER CONSTRAINTS

- Network availability
- Not preferred for patients with hearing impairment.

AVAILABLE SOLUTIONS

The prevailing solutions provides notifications.

Explore AS, differentia

Focus on J&P, tap into BE, understand RC

JOBS-TO-BE-DONE / PROBLEMS

The senior citizens tend to forget taking their medicines on time.

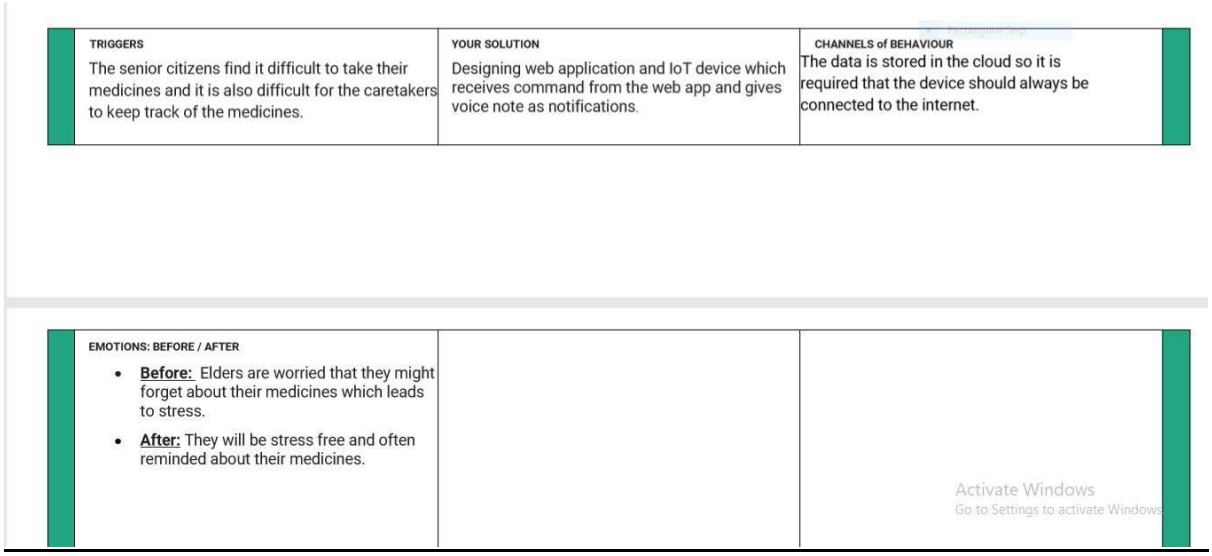
PROBLEM ROOT CAUSE

Less memory power of elders and less care towards the elders in the busy world.

BEHAVIOUR

The notifications are sent at correct timings and at regular intervals.

Focus on J&P, tap into BE, understand RC



TRIGGERS

The senior citizens find it difficult to take their medicines and it is also difficult for the caretakers to keep track of the medicines.

YOUR SOLUTION

Designing web application and IoT device which receives command from the web app and gives voice note as notifications.

CHANNELS of BEHAVIOUR

The data is stored in the cloud so it is required that the device should always be connected to the internet.

EMOTIONS: BEFORE / AFTER

- **Before:** Elders are worried that they might forget about their medicines which leads to stress.
- **After:** They will be stress free and often reminded about their medicines.

REQUIREMENT ANALYSIS

FUNCTION REQUIREMENTS:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form.
FR-2	User Confirmation	Confirmation will be done.
FR-3	Data Management	All the data are stored in the IBM cloud and retrieved.
FR-4	Internet Connectivity	User should have a stable internet connection.
FR-5	User Input Management	Required data are collected and stored.
FR-6	Acknowledgement	Required acknowledgements will be given.

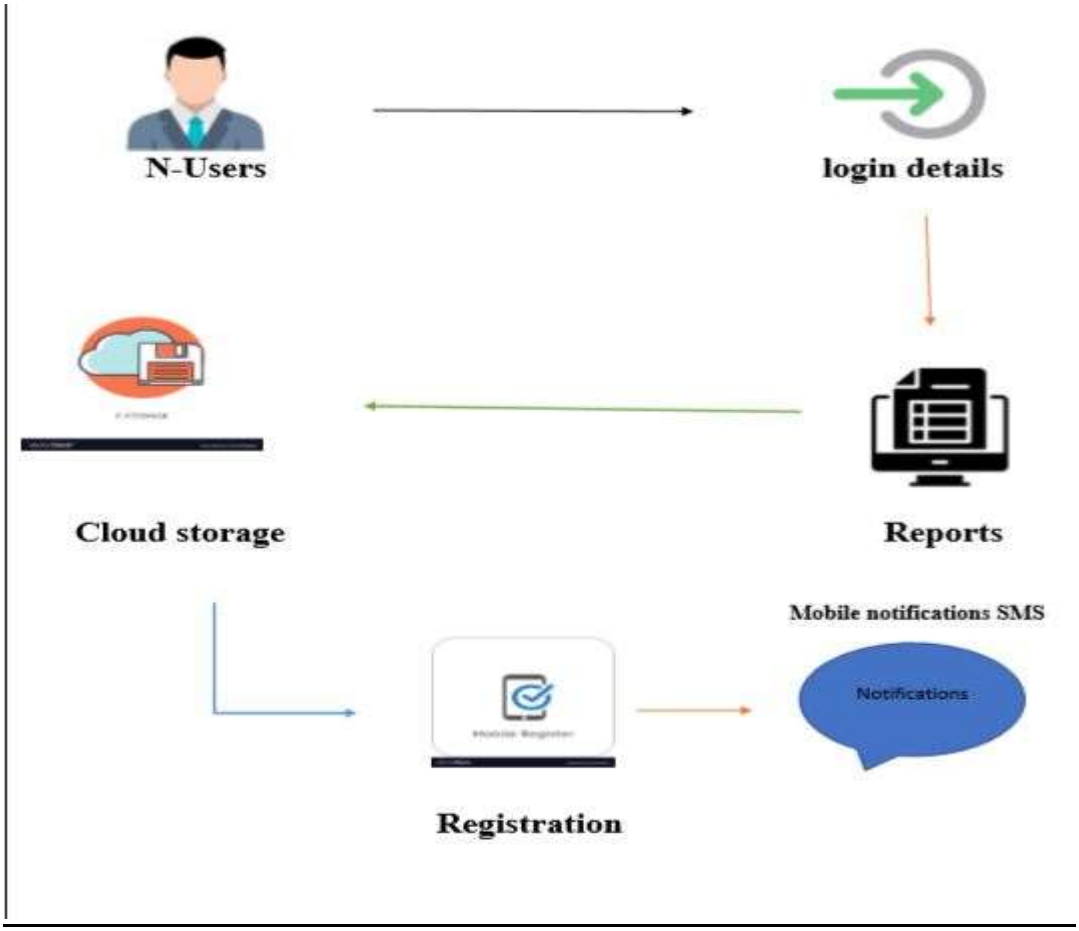
NON-FUNCTIONAL REQUIREMENTS:

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

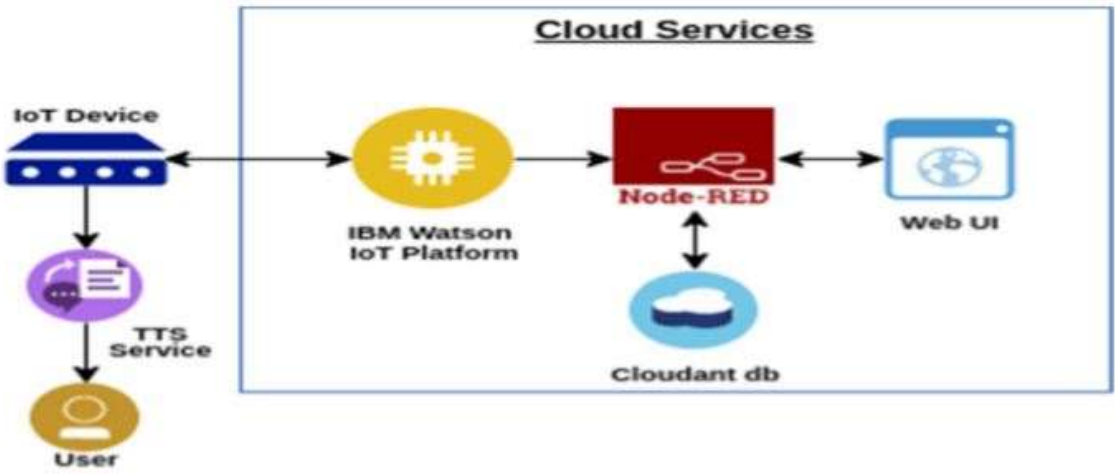
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	User friendly and simple to use.
NFR-2	**Security**	The user data is saved securely in the Cloud.
NFR-3	**Reliability**	The medicine details should be available.
NFR-4	**Performance**	Depends on internet connectivity.
NFR-5	**Availability**	Features are available according to users' requirement.
NFR-6	**Scalability**	The user can adjust the timings and it is convenient for the caregivers also.

PROJECT DESIGN

DATA FLOW DIAGRAM:



TECHNOLOGY ARCHITECTURE:



USER STORIES:

USER STORIES :

User Type	Functional Requirement	User Story Number	User Story/task	Acceptance Criteria	Priority	Release
User	Register User	USN-1	User can register themselves as a user	New user account is created	medium	sprint1
User	Login	USN-2	User login to the app	Going to the dashboard	medium	sprint2
User	Add Medicine Details	USN-3	User add his/her medicine details	Added medicine details successfully	high	sprint3
User	Activate/ Deactivate Reminder	USN-4	User activate/deactivate medicine reminder	Activated/ Deactivated successfully	high	sprint4
User	Logout	USN-5	User logout from the app	Logout the user	low	sprint4

PROJECT PLANNING & SCHEDULING

SPRINT PLANNING AND ESTIMATION:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Web UI Interface	USN-1	The user should be able to provide input to the web interface.	5	High	3
Sprint-1	Cloudant DB	USN-2	The Data will be saved in the Cloudant DB.	4	Low	1
Sprint-2	MIT App Interface	USN-3	An application that allows a user to gain access to the facility.	3	High	1
Sprint-2	Data save Acknowledgement	USN-4	Data will be saved and the user will be notified.	7	Medium	3
Sprint-3	IoT Watson	USN-5	Through IoT Watson, data from IoT devices should reach the cloud.	7	Medium	2
Sprint-3	Node-Red service	USN-6	IBM Cloud data should be correctly connected with Cloudant DB.	7	Medium	2
Sprint-3	Integration of Cloud & Node-Red Service	USN-7	It must be determined whether or not there is established communication between all services	7	High	3
Sprint-4	Text-To-Speech service	USN-8	Data processed by IBM Watson is converted into speech and returned to users.	10	High	4
Sprint-4	Voice Alert	USN-9	The voice alert is sent based on the time specified.	10	High	4

SPRINT DELIVERY SCHEDULE:

Sprint	Total Story Points	Duration	Sprint Date Start	Sprint Date Ended (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 days	24 Oct 2022	29 Oct 2022	20	28 Oct 2022
Sprint-2	20	6 days	31 Oct 2022	05 Nov 2022	20	04 Nov 2022
Sprint-3	20	6 days	07 Nov 2022	12 Nov 2022	20	10 Nov 2022
Sprint-4	20	6 days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

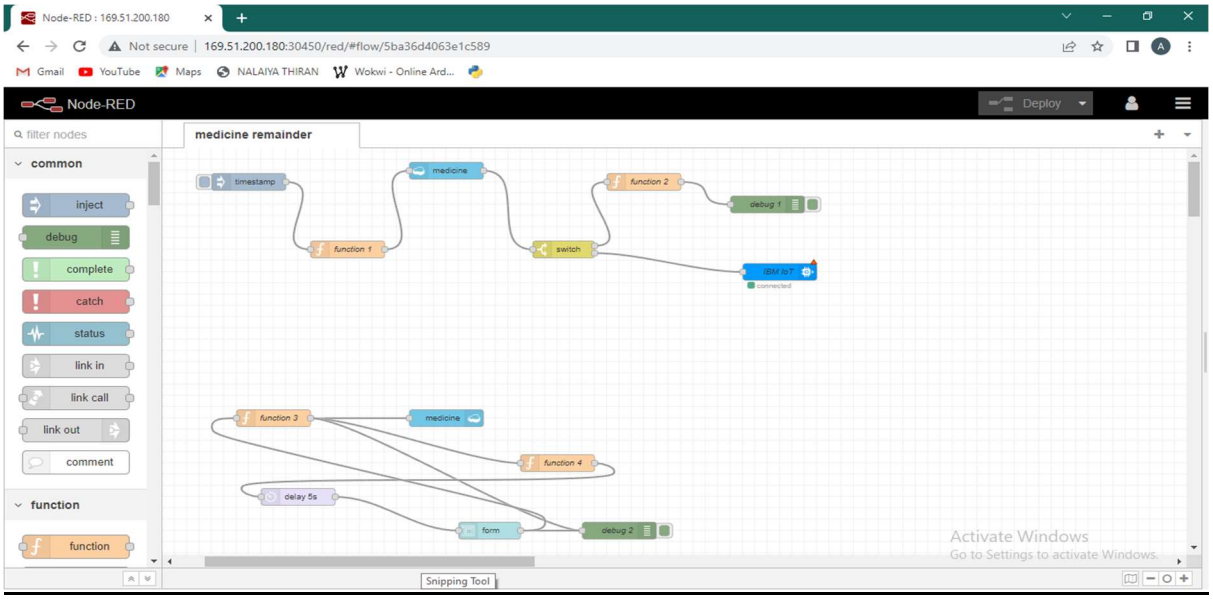
Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

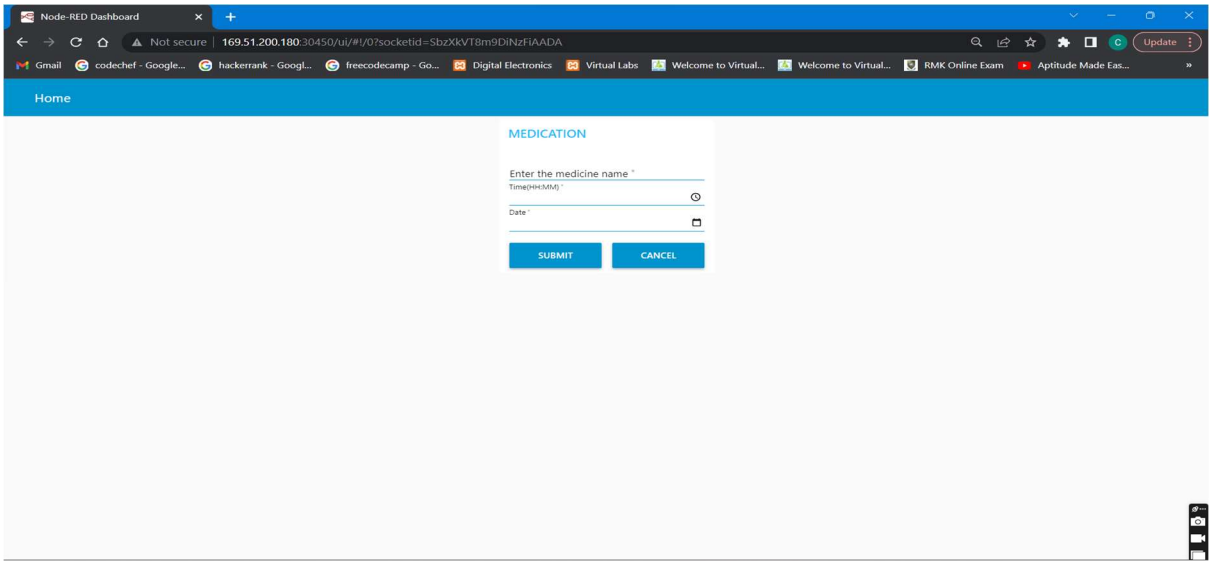
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

SPRINT 1:

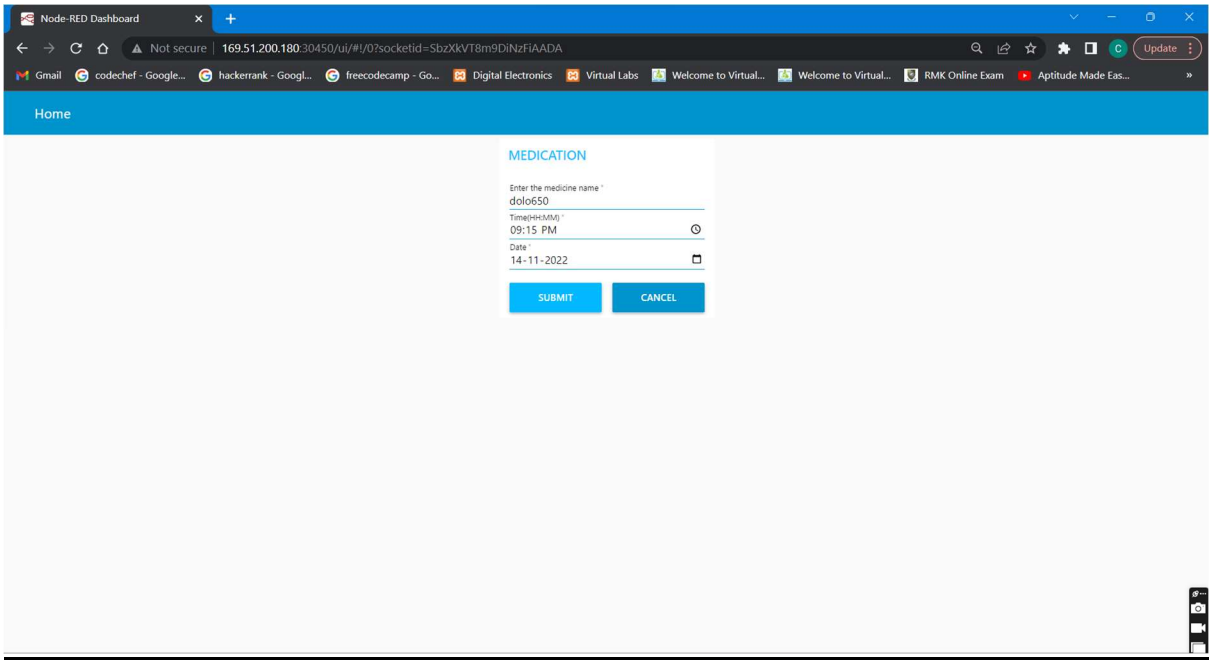
NODE RED FLOW:



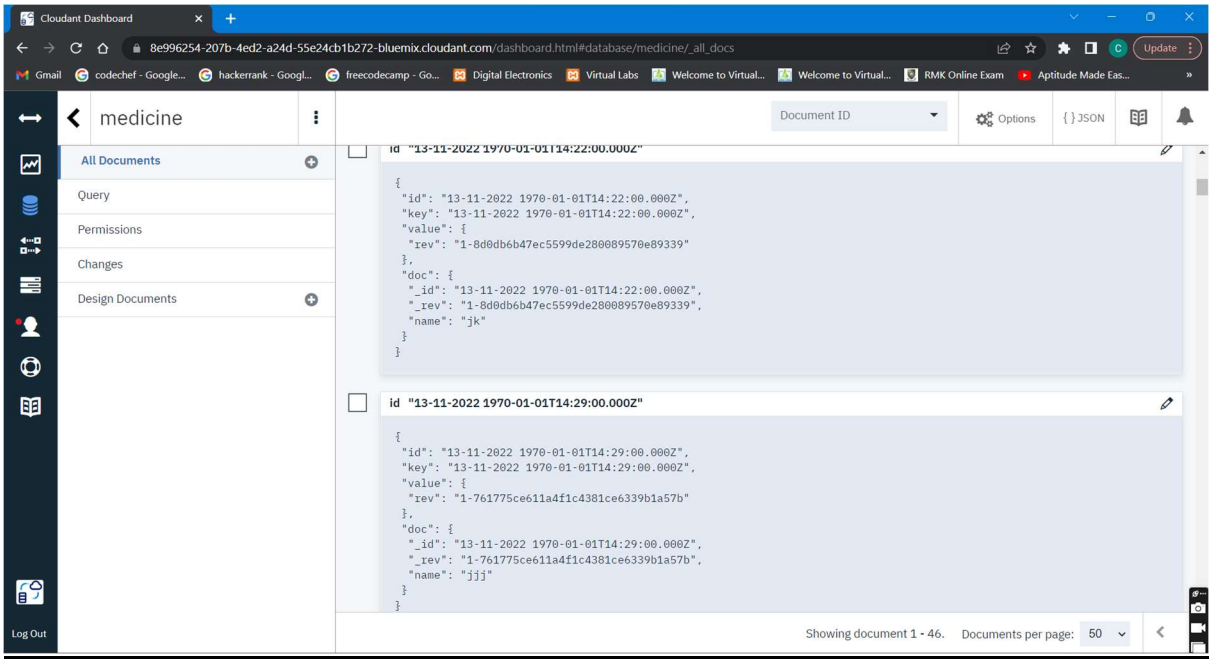
WEB UI:



DATA ENTERED BY THE USER:

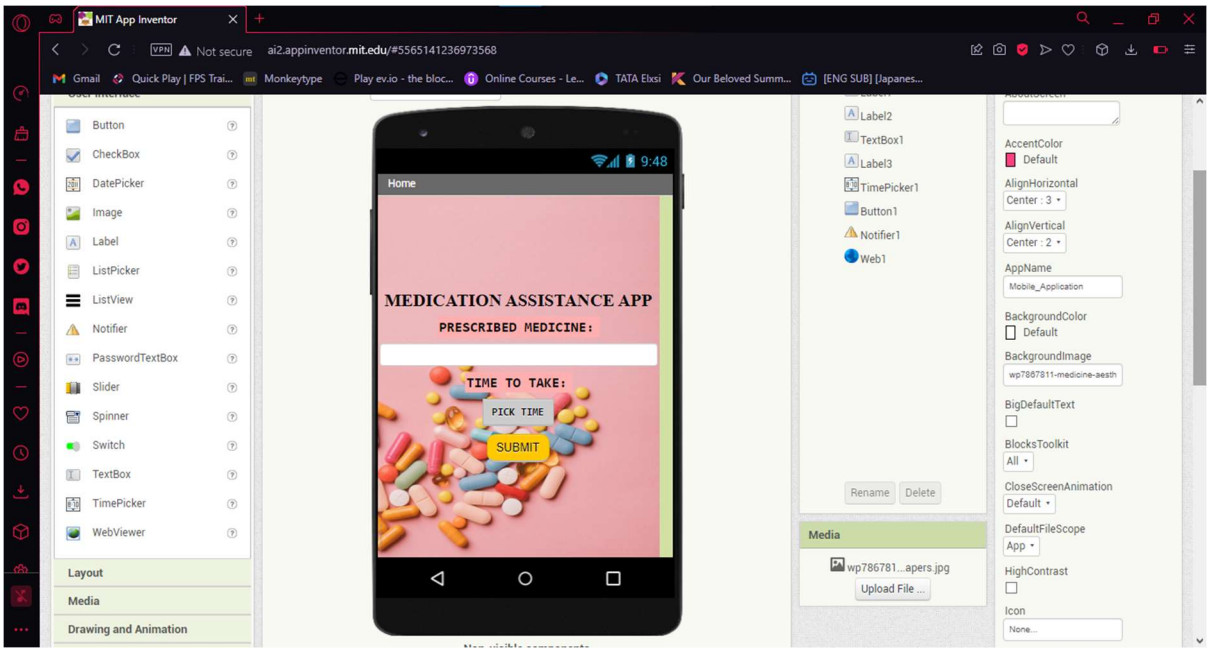


DATA SAVED IN CLOUDANT DB:

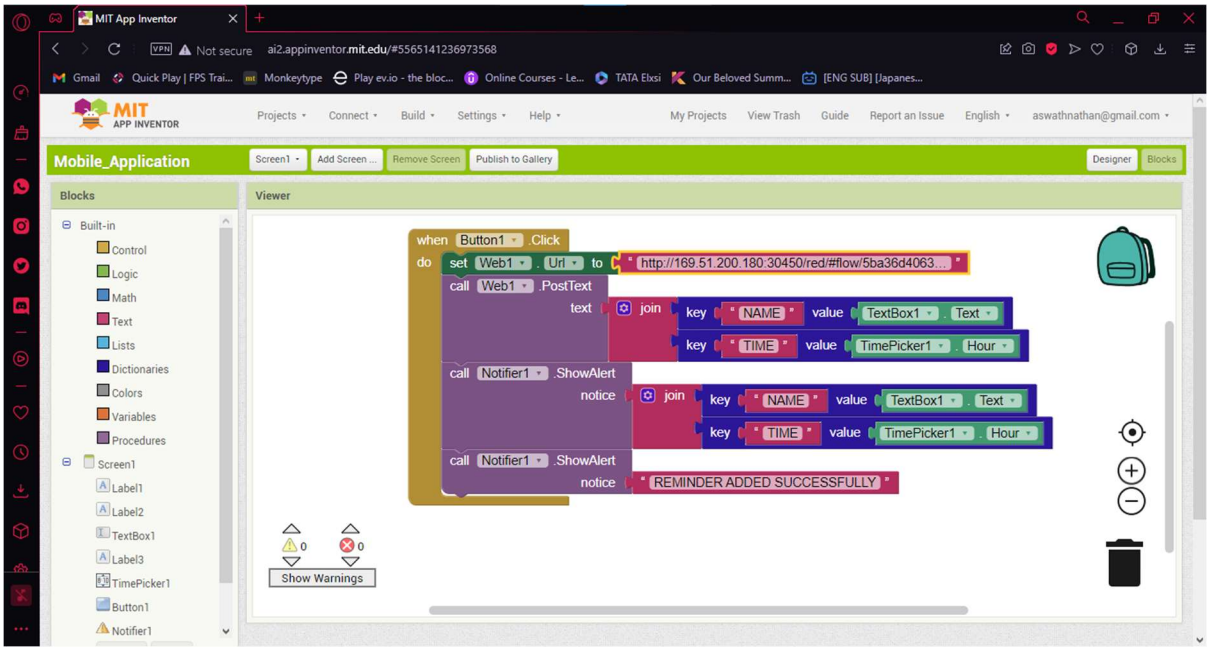


SPRINT 2:

MOBILE APPLICATION:



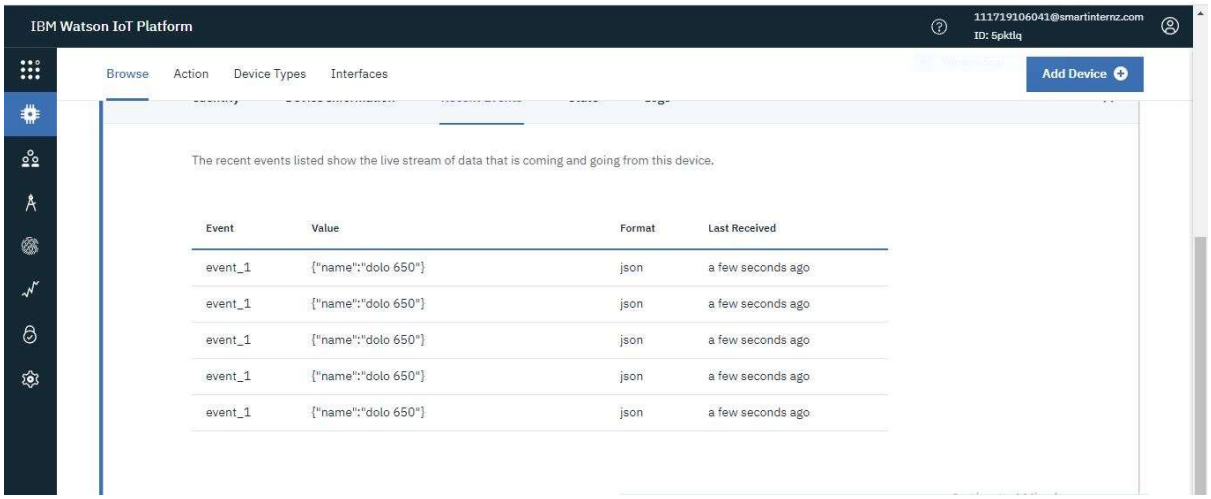
CODE BLOCKS:




Home
MEDICATION ASSISTANCE APP
PRESCRIBED MEDICINE:
dolo650
REMINDER ADDED SUCCESSFULLY
SUBMIT

SPRINT 3:

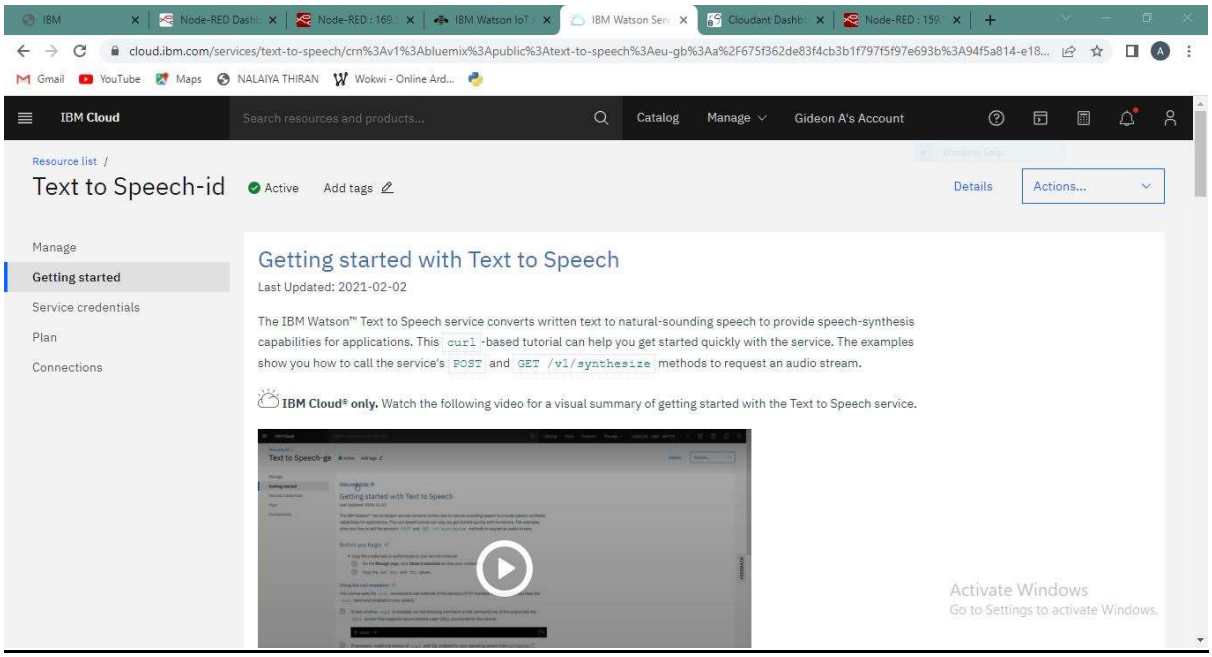
WATSON PLATFORM :



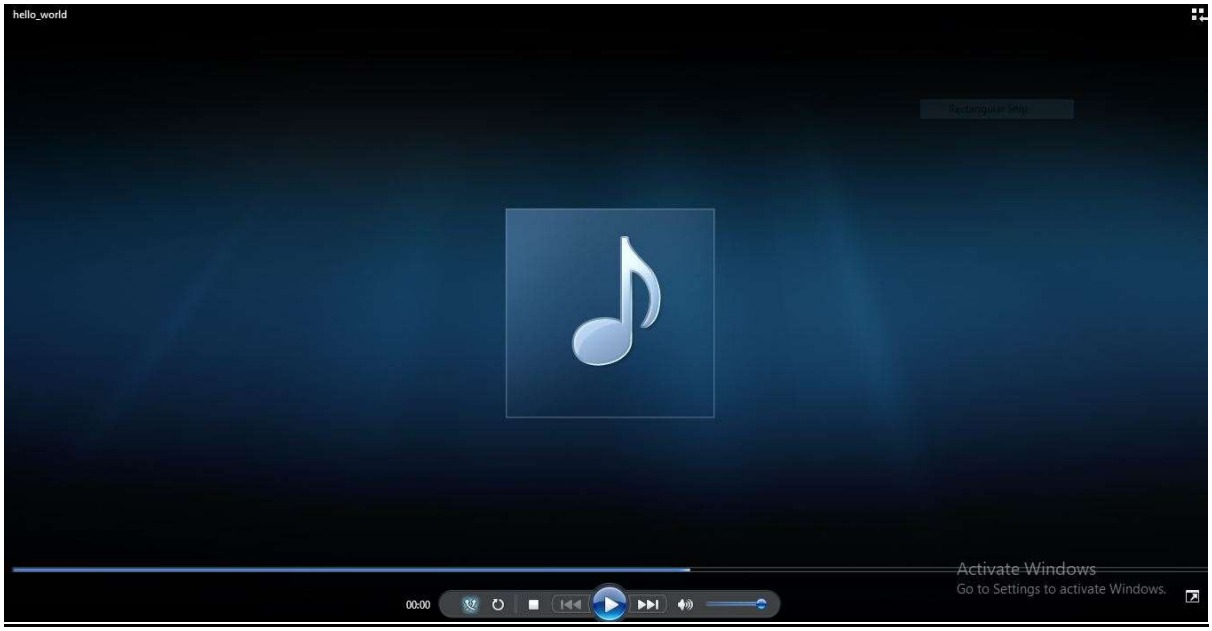
- Node red and cloudant db is connected.
- IBM IoT Watson platform device is created.

SPRINT 4

TEXT TO SPEECH:

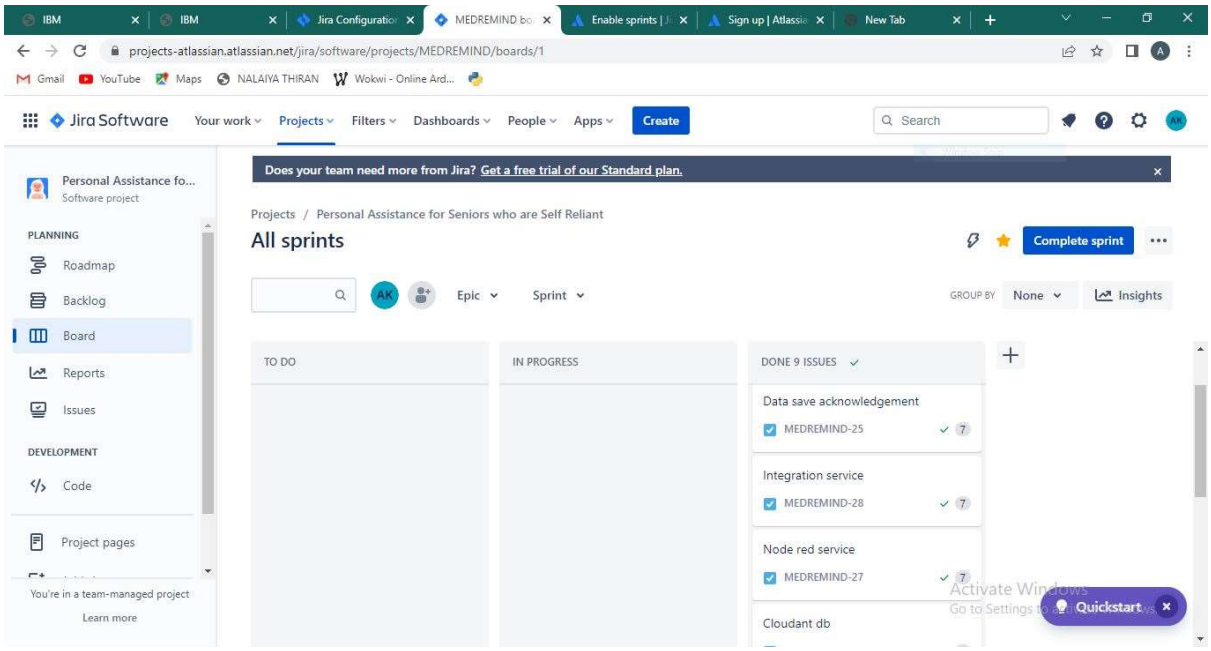


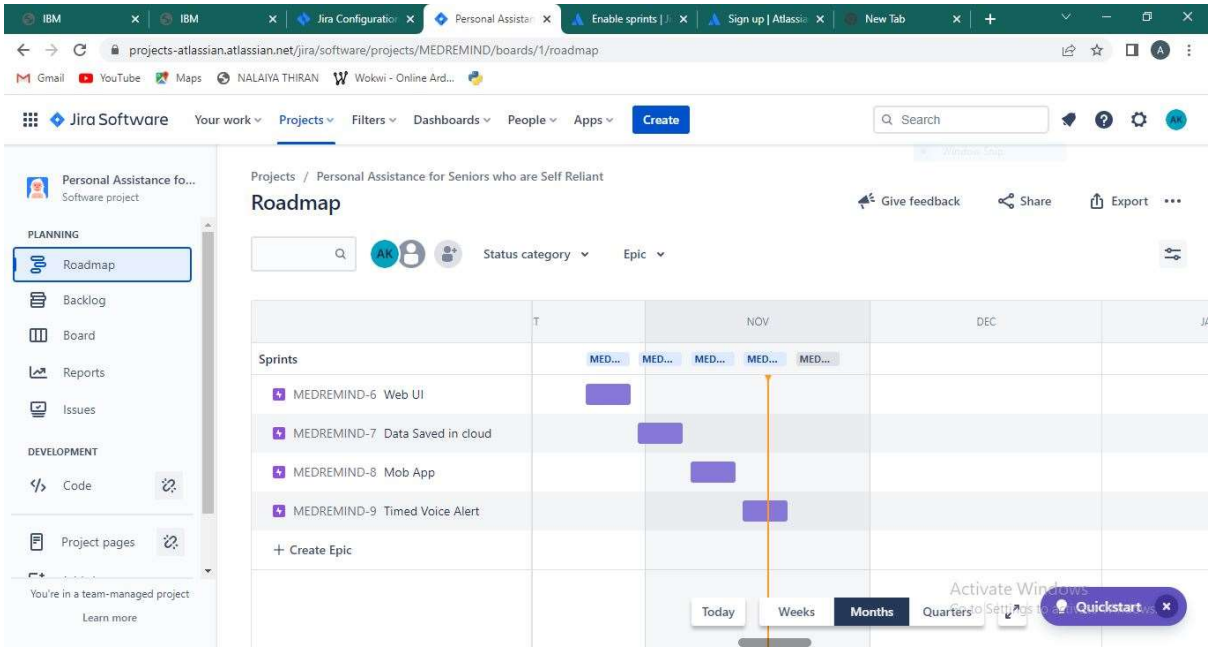
VOICE ALERT IS GIVEN:



A voice alert is given to the user using the TTS service at the time specified in the application.

JIRA FILES


projects-atlassian.atlassian.net/jira/software/projects/MEDREMIND/boards/1
Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Personal Assistance fo...
Software project
PLANNING
Roadmap
Backlog
Board
Reports
Issues
DEVELOPMENT
Code
Project pages
You're in a team-managed project
Learn more
Does your team need more from Jira? Get a free trial of our Standard plan.
Projects / Personal Assistance for Seniors who are Self Reliant
All sprints
Complete sprint
Epic
Sprint
GROUP BY
None
Insights
TO DO
IN PROGRESS
DONE 9 ISSUES
Data save acknowledgement
MEDREMIND-25
Integration service
MEDREMIND-28
Node red service
MEDREMIND-27
Cloudant db
Quickstart


projects-atlassian.atlassian.net/jira/software/projects/MEDREMIND/boards/1/roadmap
Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Personal Assistance fo...
Software project
PLANNING
Roadmap
Backlog
Board
Reports
Issues
DEVELOPMENT
Code
Project pages
You're in a team-managed project
Learn more
Projects / Personal Assistance for Seniors who are Self Reliant
Roadmap
Give feedback
Share
Export
Status category
Epic
NOV
DEC
Sprints
MEDREMIND-6 Web UI
MEDREMIND-7 Data Saved in cloud
MEDREMIND-8 Mob App
MEDREMIND-9 Timed Voice Alert
Create Epic
Today
Weeks
Months
Quarters
Quickstart

CODING AND SOLUTIONING

FEATURES:

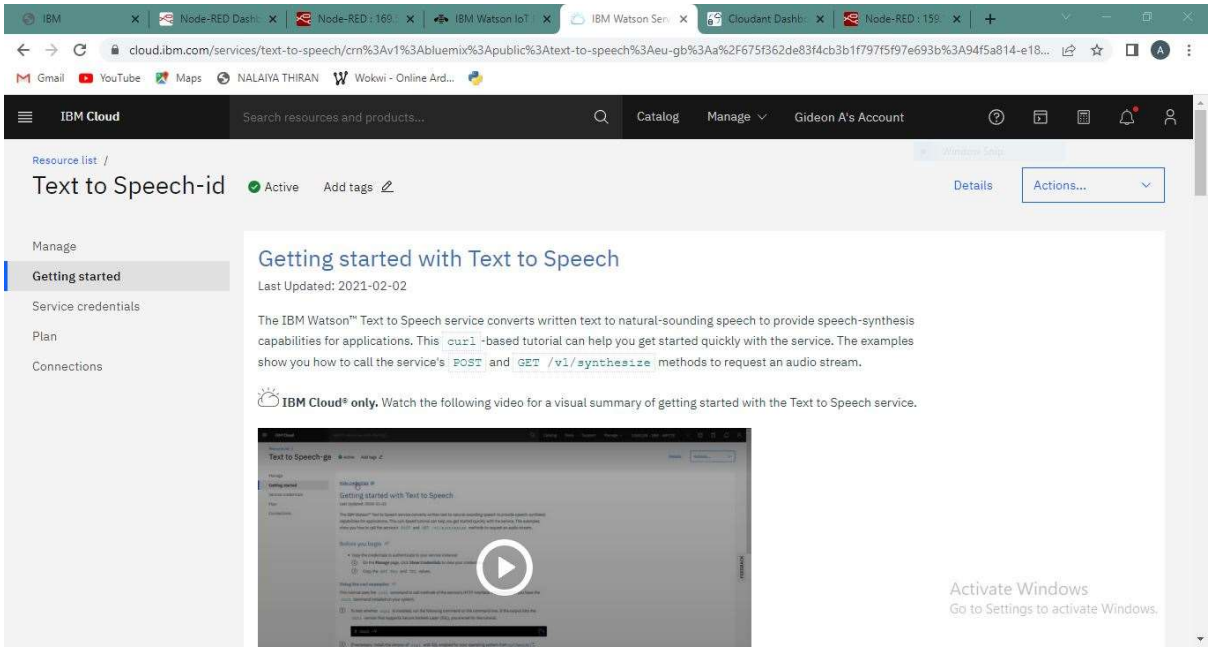
CODE:

Python:

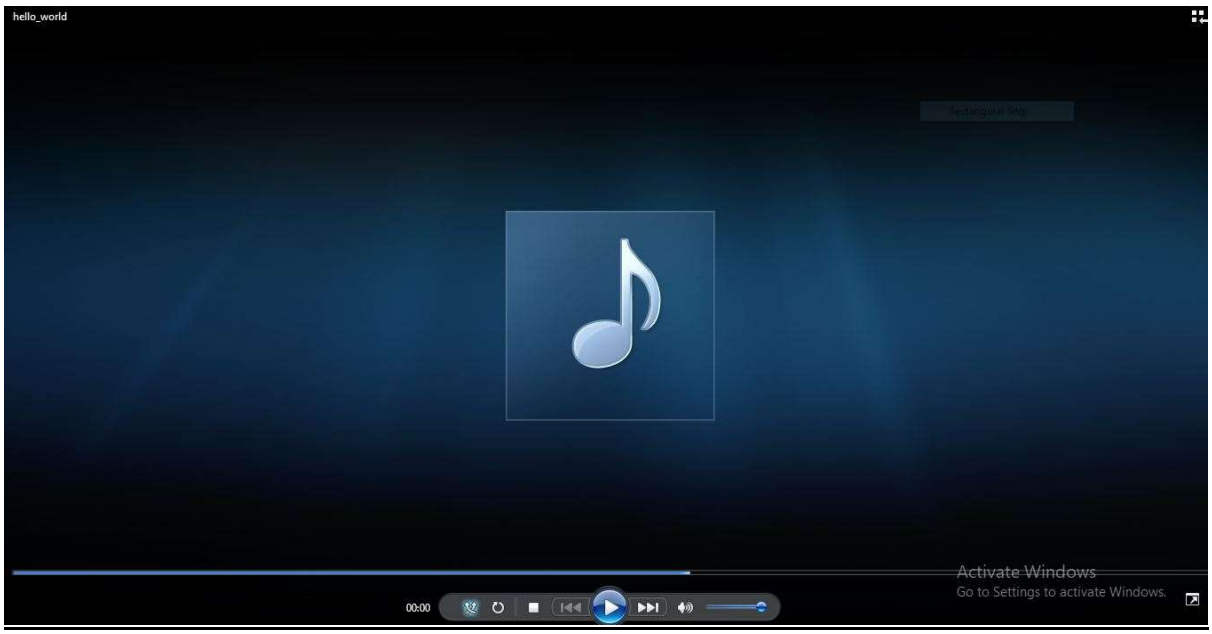
Text to Speech:

```
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from playsound import playsound

authenticator = IAMAuthenticator('msyiQReTrhmJwa67XQA2aAbs3ulcJz4w-aq1xDbFeNQu')
text_to_speech = TextToSpeechV1(
    authenticator=authenticator
)

text_to_speech.set_service_url('https://api.eu-gb.text-to-
speech.watson.cloud.ibm.com/instances/aa2dd035-028e-4fef-83fb-3f0d1e8a2423')

with open('helloworld.mp3', 'wb') as audio_file:
    audio_file.write(
        text_to_speech.synthesize(
            'Take dolo650', accept='audio/mp3', voice='en-US_AllisonV3Voice').get_result().content)
playsound('helloworld.mp3')
```

TEXT TO SPEECH:

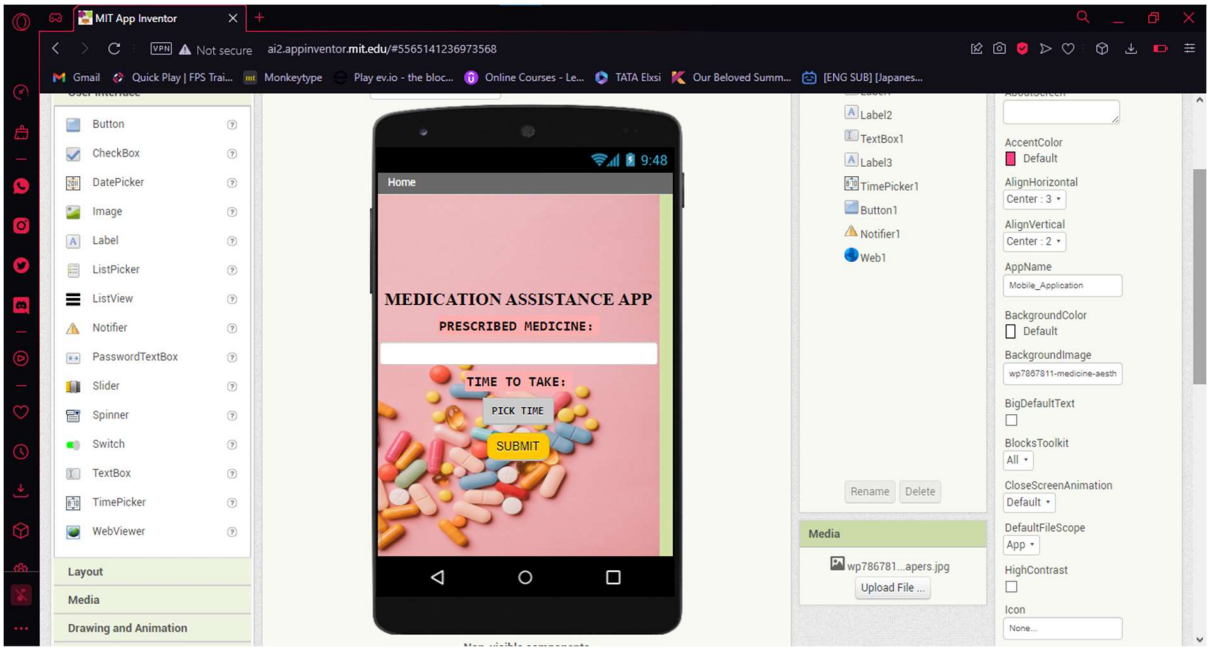


AUDIO FILE:

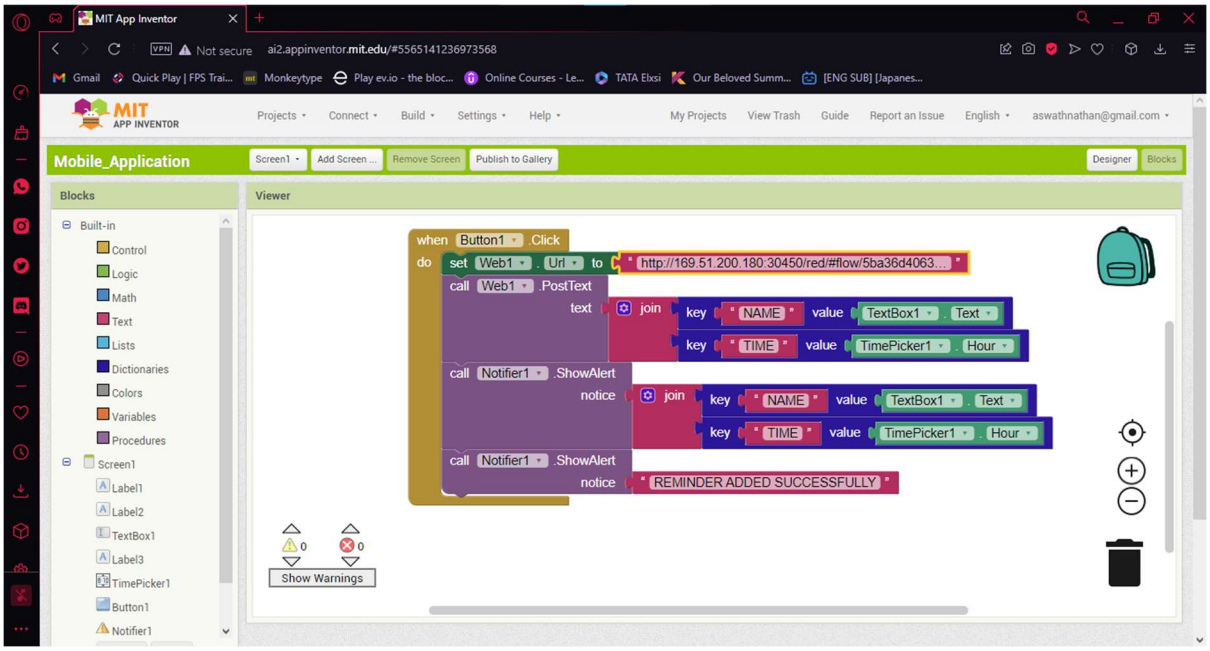


MIT APPLICATION INVENTOR:

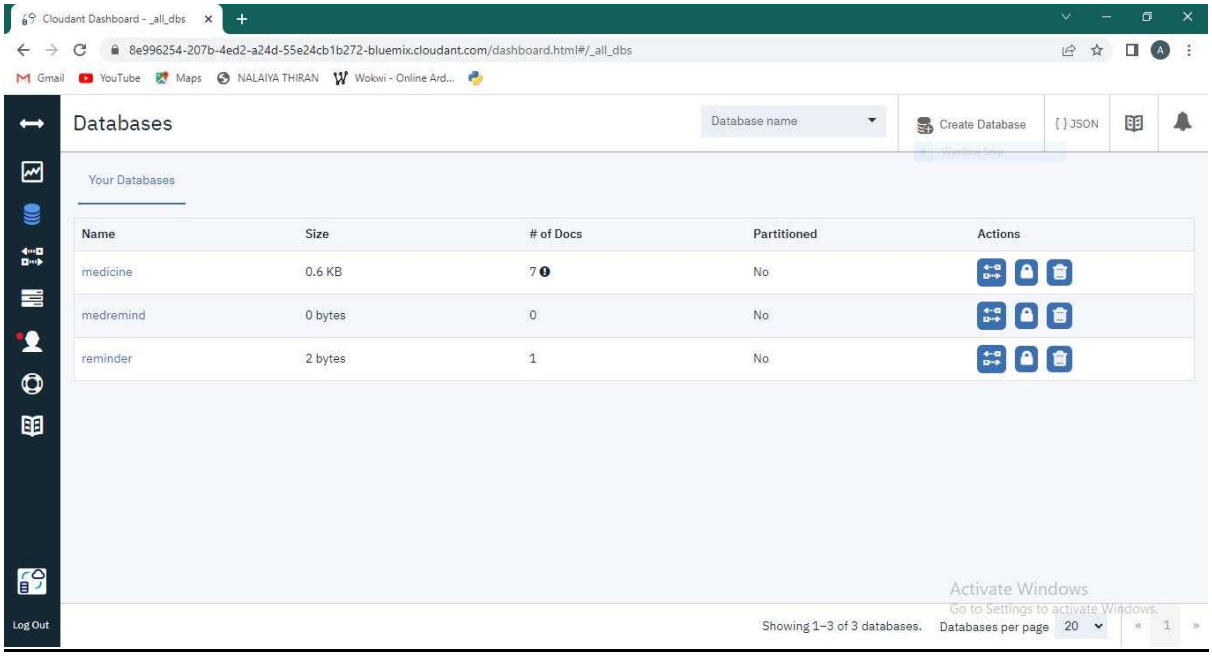
MOBILE APPLICATION:



CODE BLOCKS:



DATABASE:


Cloudant Dashboard - _all_dbs
8e996254-207b-4ed2-a24d-55e24cb1b272-bluemix.cloudant.com/dashboard.html#/_all_dbs
Gmail
YouTube
Maps
NALAIYA THIRAN
Wokwi - Online Ard...
Databases
Database name
Create Database
{ } JSON
Your Databases
Name
Size
# of Docs
Partitioned
Actions
medicine
0.6 KB
7
No
medremind
0 bytes
0
No
reminder
2 bytes
1
No
Log Out
Activate Windows
Go to Settings to activate Windows.
Showing 1–3 of 3 databases.
Databases per page
20
1

TESTING

TEST CASES:

				Date	03-Nov-22		
				Team ID	PNT2022TMID15932		
				Project Name	Personal Assistance for Seniors		
				Maximum Marks	4 marks		
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**	**Test Data**	
Web User Interface	UI	Medi page	User is able to enter medi details or not		1.Deploy the Node red flow 2. Open the Node red dashboard.	169.51.200.180:30450	
MIT App Inventer	UI	Medi page	User is able to enter medi details or not		1.Deploy the Blocks 2.Enter the User Details	http://ai2.appinventor.mit.edu/#5565141236973568	
Cloudand DB	Functional	Cloud	User entered data should be saved in database		1.Create cloudant db in IBM cloud 2. Create database in cloudant db 3. Use the database in the node red flow	cloud.ibm.com	
IBM Watson	IOT Platform	Watson	Verifying user is able to get the medicine details		1.Enter URL(https://internetofthings.ibmcloud.com/) 2.Open devices in the dash board	https://internetofthings.ibmcloud.com/	
Text to Speech	IBM	Cloud	Verify user is able to get the voice alert		1. Create TTS service 2. Configure using python code	Python script	

03-Nov-22									
PNT2022TMID15932									
Personal Assistance for Seniors									
4 marks									
Steps To Execute	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**Commnets**	**TC for Automation(Y/N)**	**BUG ID**	**Executed By**	
1.Deploy the Node red flow 2. Open the Node red dashboard.	169.51.200.180:30450	Web UI should be available	Working as expected	Pass	NIL	N	NIL	Chandrasekar MP,Aswath Nathan B	
1.Deploy the Blocks 2.Enter the User Details	http://ai2.appinventor.mit.edu/#5565141236973568	Application should be available	Working as expected	pass	NIL	N	NIL	Aswath Nathan B,Gideon A	
1.Create cloudant db in IBM cloud 2. Create database in cloudant db 3. Use the database in the node red flow	cloud.ibm.com	User entered data should be stored in the database	Working as expected	Pass	NIL	N	NIL	Aravind KM,Aswath Nathan B	
1.Enter URL(https://internetofthings.ibmcloud.com/) 2.Open devices in the dash board	https://internetofthings.ibmcloud.com/	Expected result is verified with web application input details	Working as expected	pass	NIL	N	NIL	Chandrasekar MP,Aravind KM	
1. Create TTS service 2. Configure using python code	Python script	Expected result is verified with web application input details	Working as expected	Pass	NIL	N	NIL	Gideon A,Aravind KM	

TEST SCENARIOS:

	A	B
1		
2		
3		**Test Scenarios**
4	1	To verify User is able to enter medi details in web user interface
5	2	To verify User is able to enter medi details in MIT app inventer
6	3	User entered data should be saved in database
7	4	Verifying user is able to get the medicine details
8	5	Verify user is able to get the voice alert
9		
10		
11		
12		
13		

TEST CASE FILE LINK:

https://docs.google.com/spreadsheets/d/1f-kYGWt24rRL6PFyVW8ZcU7YM_qvUGQW/edit?usp=sharing&ouid=110721793848027851816&rtpof=true&sd=true

USER ACCEPTANCE TESTING:

1. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	19
Web user interface creation	1	0	3	0	4
MIT App inventor creation	2	3	0	1	6
Fixed	10	2	4	0	16
Cloudant DB creation	5	0	2	0	7
IBM Watson	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	28	14	14	6	62

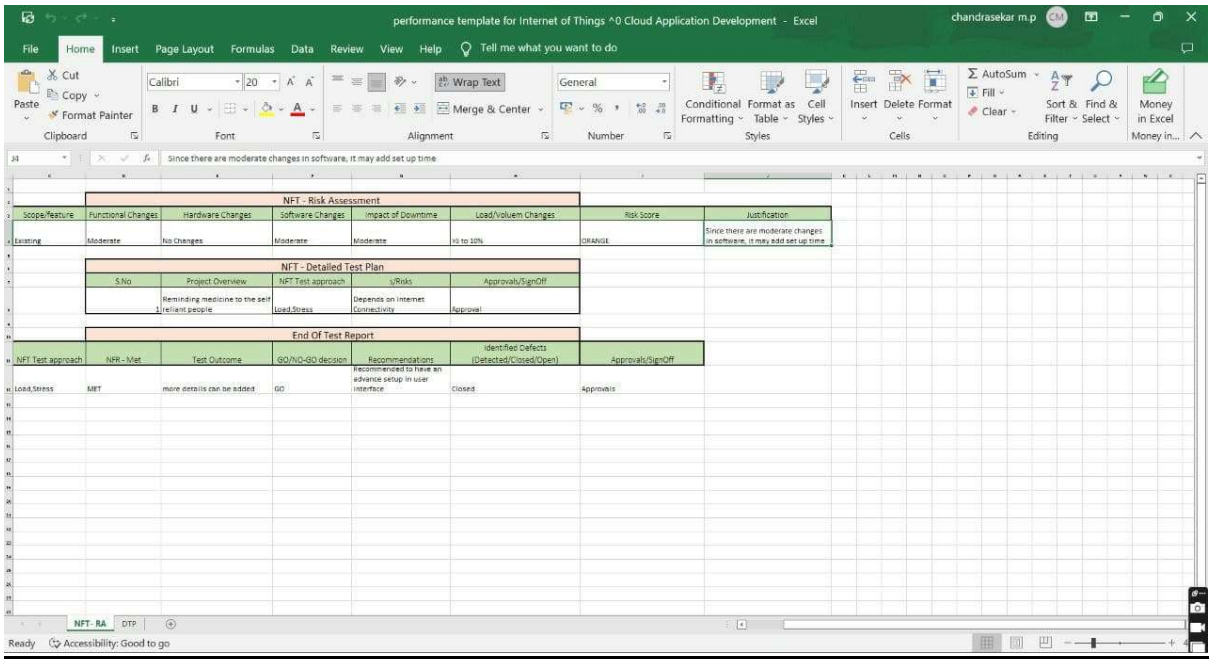
2. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

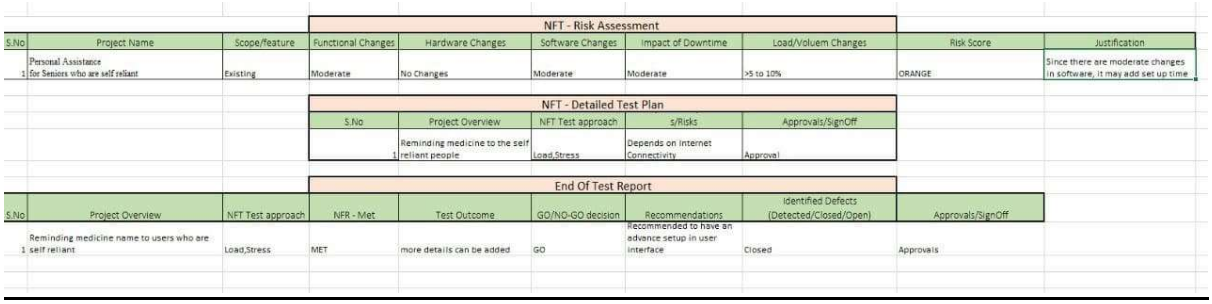
Section	Total Cases	Not Tested	Fail	Pass
Web User Interface	7	0	0	7
MIT App Inverter	5	0	0	5
Cloudant DB	2	0	0	2
IBM Watson	3	0	0	3
Text to speech	9	0	0	9
Final Report Output	4	0	0	4

RESULTS

PERFORMANCE METRICS:



ZOOMED SCREENSHOT:



PERFORMANCE FILE LINK:

https://docs.google.com/spreadsheets/d/11IMGa59UdO6lS94SiBAgKruToqukVE2p/edit?usp=sharing&ouid=110721793848027851816&rtpof=true&sd=true

ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- Personal Assistant does not administer or manage the medications, but they can remind the users to take the medicines when the time comes.
- The Personal Assistant is designed in a way that it can provide assistance to the self-reliant seniors and other users by timely reminding them to take their medicines.
- The Application used is user friendly and can be accessed by the users very easily.
- The data is stored in the cloud and it can be stored securely.
- The device is engineered with maximum probability of fabrication.
- The application can provide the service 24/7
- This project can be really helpful to amnestic patients.

DISADVANTAGES:

- Human errors can occur during setting up the time of the medication.
- Device failure during adverse conditions can end up giving a tough situation for the users.
- The application can be further improved to provide more efficient and helpful results to the seniors.

CONCLUSION:

With the technology growing at a very fast rate, a personal assistant application for the self-reliant seniors or any users could help them take care of themselves. With further improvement of this project, there could potentially be an application that could save the lives of the users. Thus, it can be concluded that a reminder that sends the timed voice alerts at specified time could be really helpful for senior citizens and also for people who often forget to take their medicines.

FUTURE SCOPE:

The Application can be improved slightly to provide much better results and additional help for the users. The Future Scope of the project can be the addition of medicine data for the users to select from the available choices instead of them entering the details. The project can also be modified to help the users to set reminders for the entire duration a particular medicine lasts.

APPENDIX

SOURCE CODE:

JSON CODE:


```
[
    {
        "id": "d10b32391db075e6",
        "type": "tab",
        "label": "Flow 7",
        "disabled": false,
        "info": "",
        "env": []
    },
    {
        "id": "aacef1e2a76faaa2",
        "type": "inject",
        "z": "d10b32391db075e6",
        "name": "",
        "props": [
            {
                "p": "payload"
            },
            {
                "p": "topic",
                "vt": "str"
            }
        ],
        "repeat": "",
        "crontab": "",
        "once": false,
        "onceDelay": 0.1,
```

```
        "topic": "",
        "payload": "",
        "payloadType": "date",
        "x": 120,
        "y": 100,
        "wires": [
            [
                "a143a99a060a26c7"
            ]
        ]
    },
```

// function for current time comparison

```
    {
        "id": "a143a99a060a26c7",
        "type": "function",
        "z": "d10b32391db075e6",
        "name": "function 11",
        "func": "\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 250,
        "y": 160,
        "wires": [
            [
                "1d3be3464db5b2b0"
            ]
        ]
    },
```

// cloudant in node

```
{
    "id": "1d3be3464db5b2b0",
    "type": "cloudant in",
    "z": "d10b32391db075e6",
    "name": "",
    "database": "",
    "service": "",
    "search": "_id_",
    "design": "",
    "index": "",
    "x": 400,
    "y": 100,
    "wires": [
        [
            "f16fafde54089487"
        ]
    ]
},
{
    "id": "f16fafde54089487",
    "type": "switch",
    "z": "d10b32391db075e6",
    "name": "",
    "property": "payload",
    "propertyType": "msg",
    "rules": [
        {
            "t": "eq",
            "v": "",
            "vt": "str"
        }
    ],
```

```
    "checkall": "true",
    "repair": false,
    "outputs": 1,
    "x": 490,
    "y": 180,
    "wires": [
      [
        "cddb77936ed3e3bb"
      ]
    ]
  },
```

//function for returning name of the medicine

```
  {
    "id": "cddb77936ed3e3bb",
    "type": "function",
    "z": "d10b32391db075e6",
    "name": "function 12",
    "func": "\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 590,
    "y": 100,
    "wires": [
      [
        "1c1c43a9b889b0a8",
        "63255dee21b43b0d"
      ]
    ]
  },
```

```
    {
        "id": "1c1c43a9b889b0a8",
        "type": "debug",
        "z": "d10b32391db075e6",
        "name": "debug 7",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "false",
        "statusVal": "",
        "statusType": "auto",
        "x": 760,
        "y": 100,
        "wires": []
    },
```

//ibm iot out node for device in the Watson platform

```
    {
        "id": "63255dee21b43b0d",
        "type": "ibmiot out",
        "z": "d10b32391db075e6",
        "authentication": "quickstart",
        "outputType": "",
        "deviceId": "",
        "deviceType": "",
        "eventCommandType": "",
        "format": "",
        "data": "",
        "qos": 0,
        "name": "IBM IoT",
        "service": "",
        "x": 720,
```

```
        "y": 200,
        "wires": []
    },
//function for sending the details to the cloud
    {
        "id": "5a2313c16068c53c",
        "type": "function",
        "z": "d10b32391db075e6",
        "name": "function 13",
        "func": "\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 190,
        "y": 300,
        "wires": [
            [
                "8fa573bc7b5ae6c9",
                "cf59d35d10997707",
                "f79be974e2b3b578"
            ]
        ]
    },
    {
        "id": "8fa573bc7b5ae6c9",
        "type": "function",
        "z": "d10b32391db075e6",
        "name": "function 14",
        "func": "\nreturn msg;",
        "outputs": 1,
```

```
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 478,
    "y": 308,
    "wires": [
      [
        "cae9e78eb60cb0c6"
      ]
    ]
  },
  {
    "id": "cae9e78eb60cb0c6",
    "type": "delay",
    "z": "d10b32391db075e6",
    "name": "",
    "pauseType": "delay",
    "timeout": "5",
    "timeoutUnits": "seconds",
    "rate": "1",
    "nbRateUnits": "1",
    "rateUnits": "second",
    "randomFirst": "1",
    "randomLast": "5",
    "randomUnits": "seconds",
    "drop": false,
    "allowrate": false,
    "outputs": 1,
    "x": 240,
    "y": 400,
    "wires": [
```

```
        [
            "4ab096667d0ec292"
        ]
    ]
},
```

//**form node to get the details from the user**

```
{
    "id": "4ab096667d0ec292",
    "type": "ui_form",
    "z": "d10b32391db075e6",
    "name": "",
    "label": "",
    "order": 0,
    "width": 0,
    "height": 0,
    "options": [
        {
            "value": "",
            "label": "",
            "type": "",
            "required": true
        }
    ],
    "formValue": {},
    "payload": "",
    "submit": "submit",
    "cancel": "cancel",
    "topic": "topic",
    "topicType": "msg",
    "splitLayout": "",
    "className": "",
    "x": 430,
```

```
        "y": 440,
        "wires": [
            [
                "f79be974e2b3b578",
                "5a2313c16068c53c"
            ]
        ]
    },
    {
        "id": "f79be974e2b3b578",
        "type": "debug",
        "z": "d10b32391db075e6",
        "name": "debug 8",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "false",
        "statusVal": "",
        "statusType": "auto",
        "x": 600,
        "y": 400,
        "wires": []
    },
//cloudant out node
    {
        "id": "cf59d35d10997707",
        "type": "cloudant out",
        "z": "d10b32391db075e6",
        "name": "",
        "database": "",
        "service": "",
```

```
        "payonly": false,
        "operation": "insert",
        "x": 460,
        "y": 260,
        "wires": []
    }
]
```


VIDEO LINK:

https://drive.google.com/file/d/19clWHaL400MXd6s6cACceXzyT4hLp-ko/view?usp=sharing

GIT HUB LINK:

https://github.com/IBM-EPBL/IBM-Project-30528-1660148061