#### **Professional Readiness for**

## **Innovation, Employability and Entrepreneurship**

# (IBM)

# Project Report

# DOMAIN: INTERNET OF THINGS

TOPIC : **Personal Assistance for Seniors Who Are Self-Reliant**

TEAM ID: **PNT2022TMID45231**

**TEAM MEMBERS:**

RAJESHKUMAR M

RAMASUBRAMANIYAN S

SENTHIL KUMAR S

RAMSUNDAR A

**Industry Mentor Faculty Mentor**

Kumar Juluri B.RAMA

**Project Report Format**

1. **INTRODUCTION**
   1. Project Overview
   2. Purpose
2. **LITERATURE SURVEY**
   1. Existing problem
   2. References
   3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   1. Empathy Map Canvas
   2. Ideation & Brainstorming
   3. Proposed Solution
   4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   1. Functional requirement
   2. Non-Functional requirements
5. **PROJECT DESIGN**
   1. Data Flow Diagrams
   2. Solution & Technical Architecture
   3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
   1. Sprint Planning & Estimation
   2. Sprint Delivery Schedule
   3. Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   1. Feature 1
   2. Feature 2
   3. Database Schema (if Applicable)
8. **TESTING**
   1. Test Cases
   2. User Acceptance Testing
9. **RESULTS**
   1. Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

Source Code

GitHub & Project Demo Link

1. **Introduction**

# Project Overview

* + - 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.

# Purpose

* + - 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.

# Literature survey

* 1. **Existing problem**

Elderly people let slip the medications at the correct time and the existing solutions for this problem is setting reminders or using pill boxes, calendars, Personal Assistance. Though the solutions give reminders, the voice commands or assistance given by this system is more efficient.

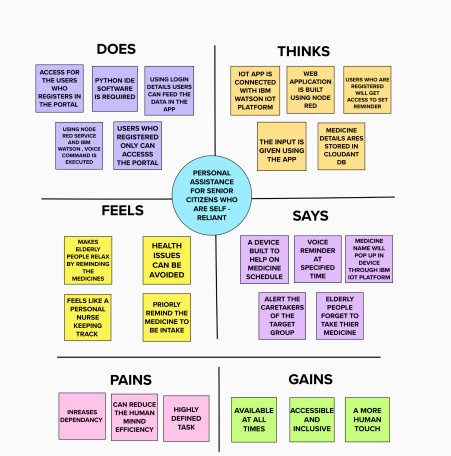
# References

* + 1. Visual Health Reminder: A Reminder for Medication Intake and Measuring Blood Pressure to Support Elderly People ; René Baranyi; Sascha Rainer; Stefan Schlossarek; Nadja Lederer; Thomas Grechenig
    2. Cloud Computing based Medical Assistance & Pill Reminder ; A. Chinnasamy; Ram Prasad J; Syed Rafeeq Ahmed; Akash S

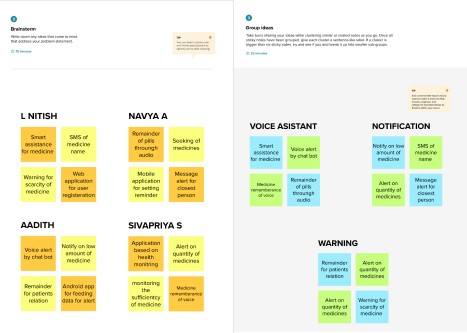
# Problem statement definition

Skipping medicines can be serious for some medical health conditions; Sometimes elderly people forget to take their medicine at the correct time. They also forget which medicine one should take at that particular time. And it is difficult for doctors/caretakers to monitor the patients around the clock.

# Ideation and proposed solution

* 1. **Empathy Map Canvas**

# Ideation and Brainstorming

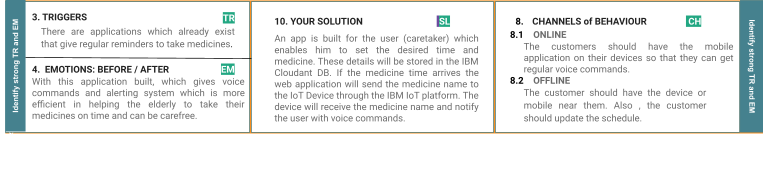
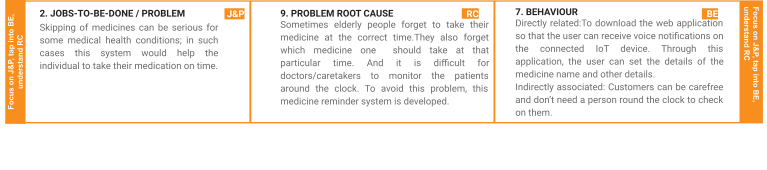
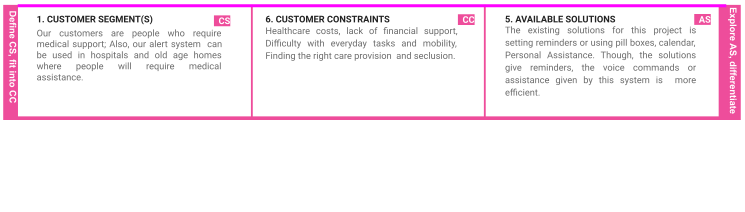


* 1. **Proposed solution**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
| 1. | Problem Statement (Problem to be solved) | Sometimes elderly people forget to take their medicine at the correct time. They also forget which medicine should be taken at that particular time. And it is difficult for doctors/caretakers to monitor the patients around the clock. |
| 2. | Idea / Solution description | * A 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. |
| 3. | Novelty / Uniqueness | * Keeping track of the medicines taken by the user at each time interval. * Information is stored in the secured IBM cloud. |
| 4. | Social Impact / Customer  Satisfaction | The reminder system enables the user to  take tablets at regular intervals prescribed by the physicians. |
| 5. | Business Model (Revenue  Model) | **Direct Mode:** We gain revenue from  selling the medical reminder system to hospitals, medical health centres and even in old age homes.  **Indirect Mode:** We gain profit by having partnership with pharmaceutical companies. |
| 6. | Scalability of the Solution | The medical alert system can be used in  hospitals, medical health centres and even in old age homes for dispensing medicines. |

# Problem Solution fit



1. **Requirement analysis**

# Functional Requirements:

|  |  |  |
| --- | --- | --- |
| **FR**  **No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Form Registration through Gmail |
| FR-2 | User Confirmation | Confirmation via Email Confirmation via OTP |
| FR-3 | Access Cloud services | Accessing the cloud service with correct credentials. Storing the details in the cloud database. |
| FR-4 | IOT configuration | Fine Tuning the IOT device based Cloud DB access via device.  Manage the data request and response effectively |

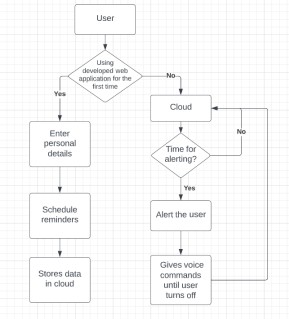
* 1. **Non-functional Requirements:**

|  |  |  |
| --- | --- | --- |
| **FR**  **No.** | **Non-Functional Requirement** | **Description** |
| NFR-  1 | **Usability** | App can be used by anyone who has knowledge about applications and computers. |
| NFR-  2 | **Security** | For security, TFA is enabled and biometrics arealso added for user safety. |
| NFR-  3 | **Reliability** | Highly reliable since, It usestrusted and authentic cloud services like IBM |

|  |  |  |
| --- | --- | --- |
| NFR-  4 | **Performance** | Performance is better compared to other marketproducts. |
| NFR-  5 | **Availability** | Available on mobile app. |
| NFR-  6 | **Scalability** | Using Cloud services, makes the scalability higher the using traditional locally stored database. |

# Project Design

* 1. **Data Flow Diagrams**



# Technical architecture

* 1. **User Stories**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional Requireme nt (Epic)** | **User Story Num ber** | **User Story / Task** | **Acceptanc e criteria** | **Priorit y** | **Release** |
| Customer (Mobile user) | Registratio n | USN- 1 | As a user, I can register for the application by entering my email or mobile number, password, and confirming my  password. | I can access my account / dashboard | High | Sprint-1 |
|  |  | USN- 2 | As a user, I will receive confirmation email once I have registered for the application | I can receive confirmati on email & click confirm | High | Sprint-1 |
|  |  | USN- 3 | As a user, I can register for the application  through Gmail |  | Mediu m | Sprint-1 |
|  | Login | USN- 4 | As a user, I can log into the application by entering email or mobile number & password | I can access my account / dashboard | High | Sprint-1 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional Requireme nt (Epic)** | **User Story Num**  **ber** | **User Story / Task** | **Acceptanc e criteria** | **Priorit y** | **Release** |
|  | Dashboard | USN- 5 | As a user, I can update my reminders and medicines wherever required |  | High | Sprint-2 |
|  |  | USN-  6 | As a user, I can check the  application whether the medicine dosage is completed. |  | Mediu  m | Sprint-2 |
| Customer  Care Executiv e |  | USN-  7 | For any troubleshooting,  the user can send a mail to the technical team. |  | Low |  |
| Administ  rator |  | USN-  8 | Ensures smooth  functioning and data warehousing strategies |  | Mediu  m | Sprint-3 |

# Project Planning and Scheduling

* 1. **Sprint Planning and Estimation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requiremen t (Epic)** | **User Story Number** | **User Story / Task** | **Stor y Poin ts** | **Priori ty** | **Team Members** |
| Sprint- 1 | Registratio n | USN-1 | As a user, I can register for the application by entering my email or mobile number, password, and confirming my password. | 2 | High | Ramasubramaniyan S |
| Sprint- 1 |  | USN-2 | As a user, I will receive confirmation email once I have registered for the application | 1 | High | Ramasubramaniyan S |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint-1 |  | USN-3 | As a user, I can register for the application through Gmail | 2 | Medium | Rajeshkumar |
| Sprint-1 |  | USN-4 | As a user, I can log into the application by entering email or mobile number & password | 2 | High | Senthilkumar |
| Sprint-2 | Login | USN-5 | As a user, I can update my reminders and medicines wherever required | 1 | High | Ramasubramaniyan S |
| Sprint-2 | Dashboard | USN-6 | As a user, I can check the application whether the medicine dosage is completed | 1 | Mediu m | Ramasubramaniyan S |
|  |  | USN-7 | For any troubleshooting, the user can send a mail to the technical team | 1 | Low | Senthilkumar |
| Sprint-3 |  | USN-8 | Ensures smooth functioning and data  warehousing strategies | 1 | Mediu m | Rajeshkumar S |

# Sprint Delivery Schedule

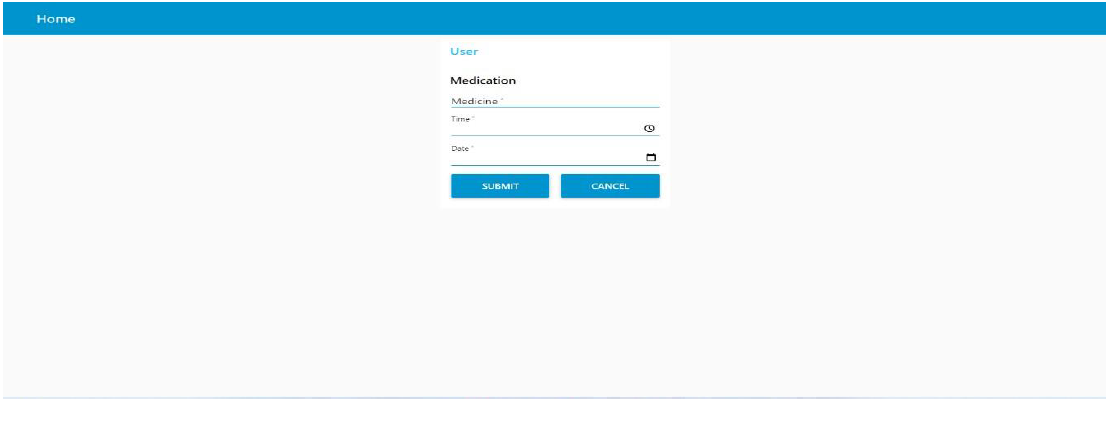
**Table

Description automatically generated**

**7.0 Coding and Solutioning**

# Feature 1

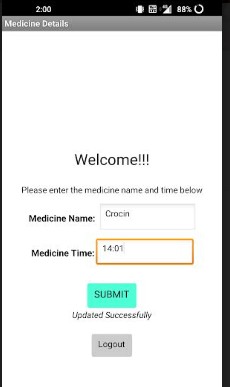
# The mobile application and web application developed has a feature of individual login by different users.

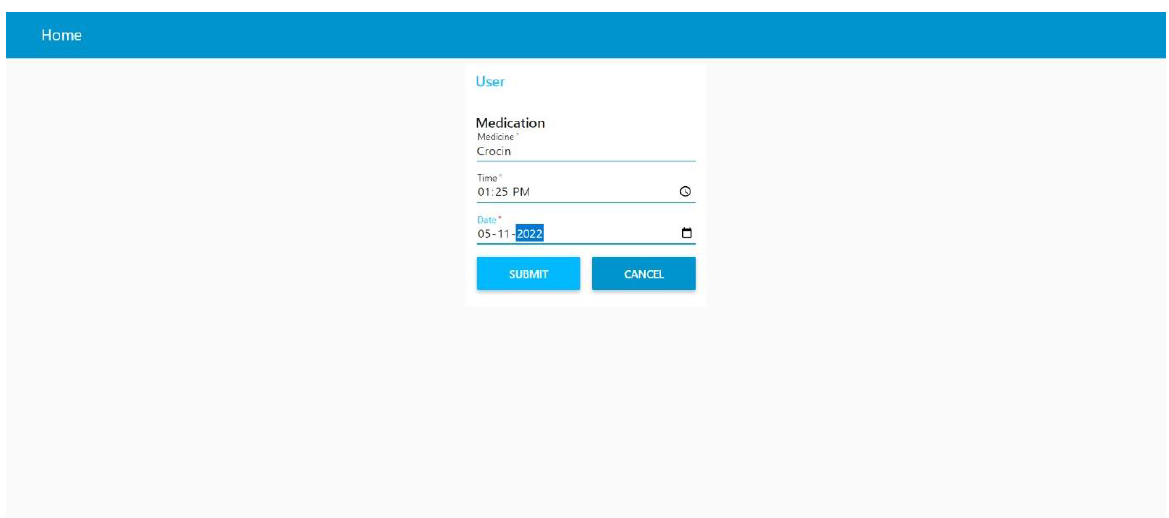


A screenshot of a computer

Description automatically generated with medium confidence

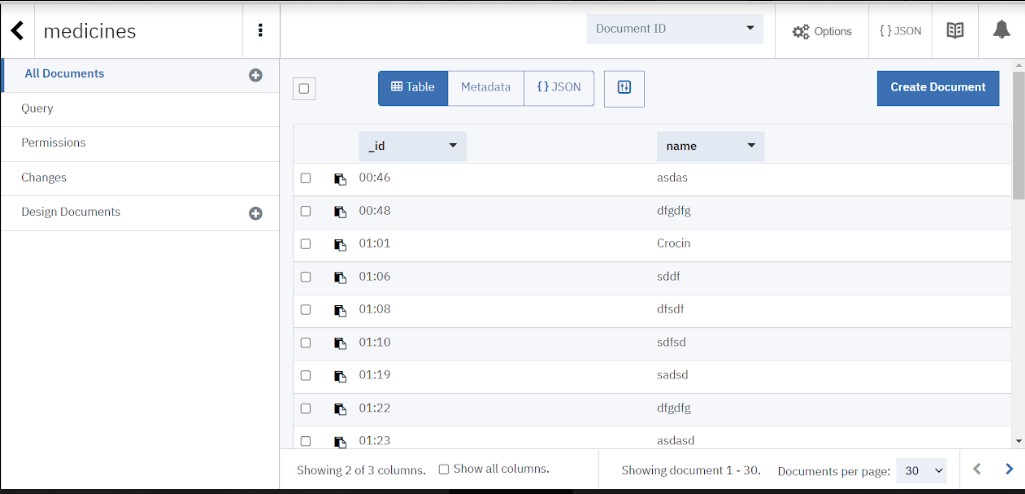
# Feature 2

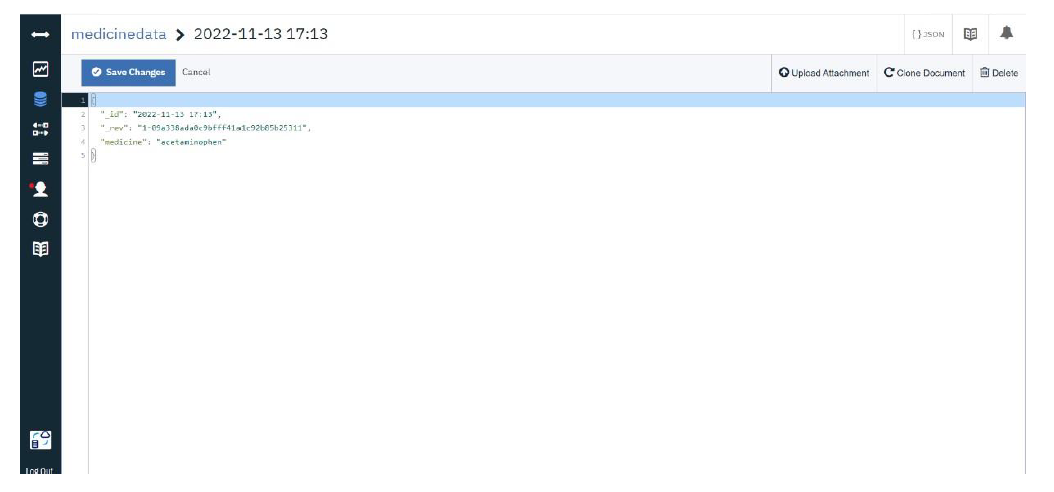
The mobile application and web application also has the feature of uploading medicine names in the cloud.





# 7.3. Feature 3

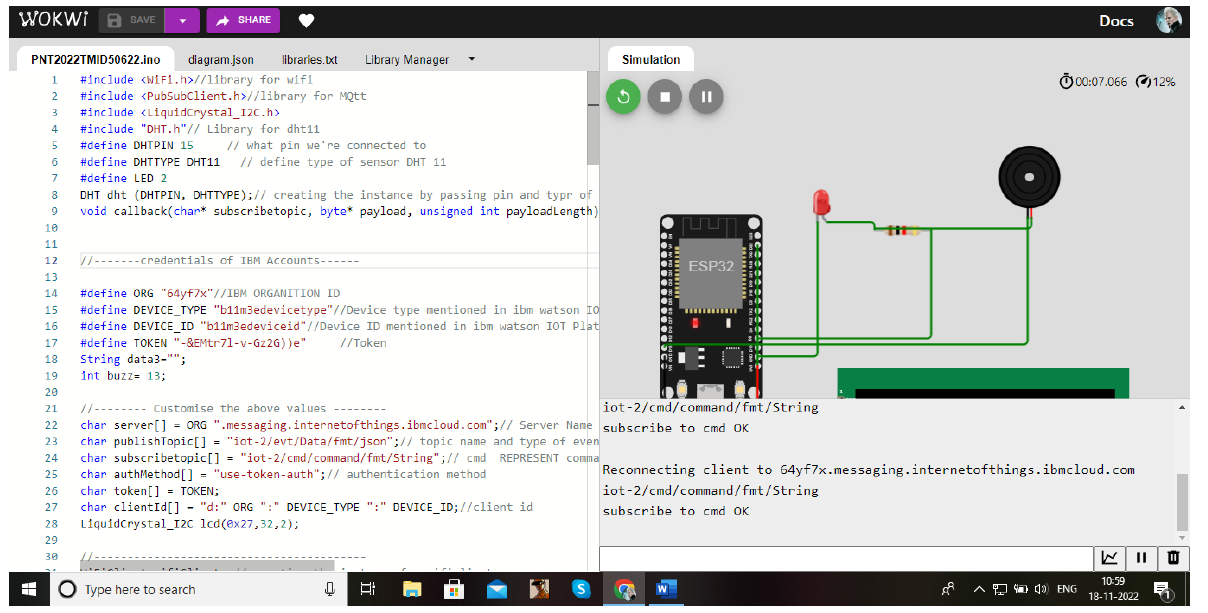
The project includes a cloud database system.



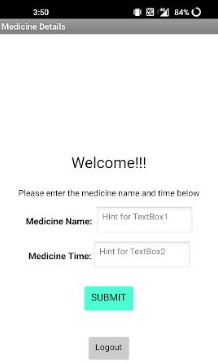
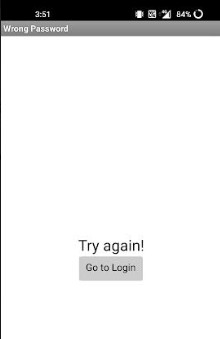
# 8.Testing

* 1. **Test cases**

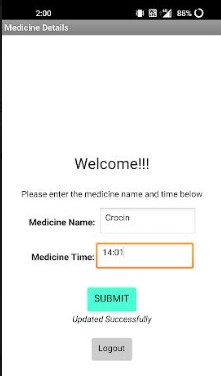
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Precondition** | **Test steps** | **Test data** | **Expected result** |
| Verify login with | User should | 1. Launch URL | Username: | Users should |
| valid credentials | have a network | 2. Enter valid username. | Navya | be able to login |
|  | connection | 3. Enter valid password. | Password: | successfully. |
|  |  | 4. Click on the “Login” | 12345 |  |
|  |  | button. |  |  |
| Verify login with | User should | 1. Launch URL | Username: | Users should |
| invalid credentials | have a network | 2. Enter valid username. | Navya | not be able to |
|  | connection | 3. Enter invalid password. | Password: | login. |
|  |  | 4. Click on the “Login” | Navya123 |  |
|  |  | button. |  |  |
| Update the | User should | 1. Enter valid medicine | Medicine | Users should |
| medicine name | have a network | name. | Name: | be able to |
| with the time. | connection | 2. Enter the time when the | Cetirizine | update it |
|  |  | medicine has to be | Medicine | successfully. |
|  |  | consumed. | Time: |  |
|  |  | 3. Click on the “Submit” | 20.00 |  |
|  |  | button. |  |  |



# User acceptance testing

 **Login page testing Incorrect login attempt**

# Medicine page testing



1. **Results**

# Performance Metrics

|  |  |  |
| --- | --- | --- |
| **S. NO** | **Parameter** | **Performance** |
| 1. | Response Time | 0.2s (Average of 10 trials) |
| 2. | Workload | 500 users ( Calculated based on Cloud Space) |
| 3. | Revenue | Individual users and pharmaceutical industries. |
| 4. | Efficiency | Simple and straightforward workflow, which makes the process efficient. |
| 5. | Down Time | Almost no down time due to IBM Cloud enabled solution. |

1. **Advantages and Disadvantages Advantages**
   * + Help the elderly people to take their medicine at the correct time.
     + Avoid personal assistants or caretakers needed for medically sick people.
     + Cost efficient.
     + Can store multiple data and many notifications can be generated.
     + Since it includes voice assistance, even blind people can use our device.

# Disadvantages

* + - Makes people lethargic and makes them dependent always on others.
    - Requires a stable internet connection.

# Conclusion

The project offers the elderly or medically sick people a personal assistant which reminds them of the medicines to be consumed at the particular time. Skipping tablets may lead to serious problems if the person has a severe illness and this can be avoided. Since the cloud is integrated with the mobile application, numerous data can be fed into the database and notifications can be generated. The mobile application and web application developed is highly customisable by the user and easy to use.

# Future Scope

The project can be further developed by bringing into the feature of informing the medicine name during the notification. The voice assistance which is given can be customized by adding the user's voice or the caretaker’s voice. Further the mobile application and web application can update medicines by taking voice commands as an input from the user.

# Appendix Source Code:

#include <WiFi.h>//library for wifi

#include <PubSubClient.h>//library for MQtt #include "SoundData.h"

#include "XT\_DAC\_Audio.h"

XT\_Wav\_Class Sound("voice\_command.wav"); XT\_DAC\_Audio\_Class DacAudio(2,0); uint32\_t DemoCounter=0;

void callback(char\* subscribetopic, byte\* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "ut4tn5"//IBM ORGANITION ID

#define DEVICE\_TYPE "Arduino"//Device type mentioned in ibm watson IOT Platform #define DEVICE\_ID "nitish123"//Device ID mentioned in ibm watson IOT Platform #define TOKEN "123456789" //Token

String data3; float h, t;

//-------- Customise the above values --------

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name

char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send

char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING

char authMethod[] = "use-token-auth";// authentication method char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;//client id

//

WiFiClient wifiClient; // creating the instance for wificlient

PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by passing parameter like server id,portand wificredential

void setup()// configureing the ESP32

{

Serial.begin(115200);

delay(10); Serial.println(); wificonnect(); mqttconnect();

}

void loop()// Recursive Function

{

delay(1000);

if (!client.loop()) { mqttconnect();

}

}

/\*.....................................retrieving to Cloud \*/

void mqttconnect() {

if (!client.connected()) { Serial.print("Reconnecting client to "); Serial.println(server);

while (!!!client.connect(clientId, authMethod, token)) {

Serial.print("."); delay(500);

}

initManagedDevice(); Serial.println();

}

}

void wificonnect() //function defination for wificonnect

{

Serial.println(); Serial.print("Connecting to ");

WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection

while (WiFi.status() != WL\_CONNECTED) { delay(500);

Serial.print(".");

}

Serial.println(""); Serial.println("WiFi connected"); Serial.println("IP address: "); Serial.println(WiFi.localIP());

}

void initManagedDevice() {

if (client.subscribe(subscribetopic)) { Serial.println((subscribetopic)); Serial.println("subscribe to cmd OK");

} else {

Serial.println("subscribe to cmd FAILED");

}

}

void callback(char\* subscribetopic, byte\* payload, unsigned int payloadLength)

{

Serial.print("callback invoked for topic: "); Serial.println(subscribetopic);

for (int i = 0; i < payloadLength; i++) {

//Serial.print((char)payload[i]); data3 += (char)payload[i];

}

Serial.println("data: "+ data3); if(data3=="announce")

{

Serial.println(data3); for(int i=0;i<5;i++){ DacAudio.FillBuffer(); if(Sound.Playing==false)

DacAudio.Play(&Sound); Serial.println(DemoCounter++);

}

}

else

{

pass;

}

data3="";

}

**Github link**: <https://github.com/IBM-EPBL/IBM-Project-44928-1660727470>

**Demo link:** <https://drive.google.com/drive/folders/1SbsBP55HZlJCdi607KfEjp0V-hbQ0db5>