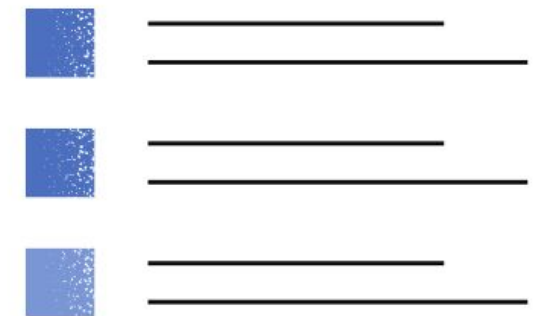
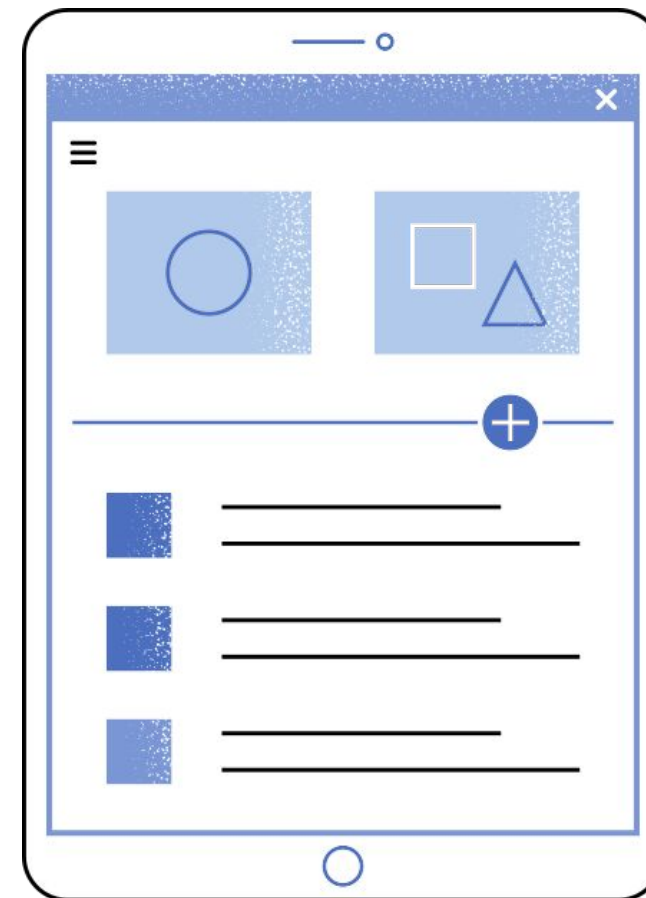
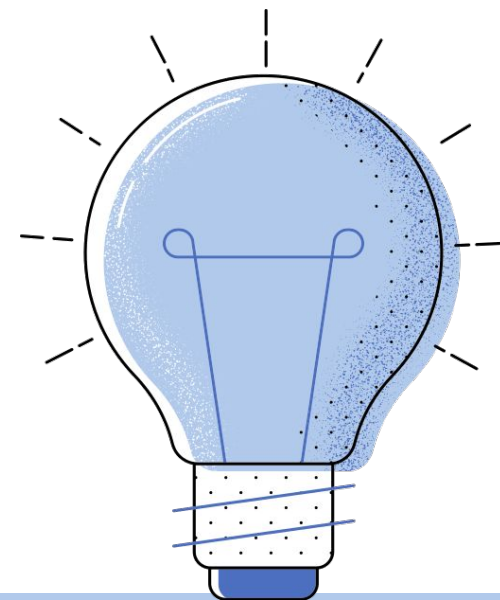
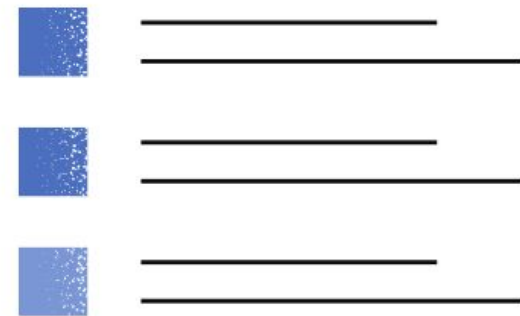
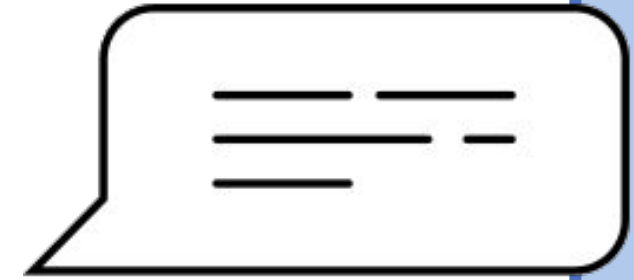




IOT PROJECT ON SAFETY FOR ELDERLY SAFETY SHIELD



SAMSUNG IOT INTERNSHIP 2023-24

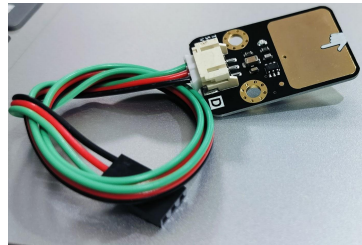
TEAM WARRIORS



01 PROJECT OVERVIEW

- In this busy society, there are many who are not able to take care of our parents or grandparents.
- When the elderly are in need of us, we are busy with our works. There have been many cases, where the old people are left at old age homes in order to avoid the need of taking care of them.
- There are thousands of cases, where the elders' call of distress goes unheard. In order to curb these problems, we bring forth our project to bridge the gap between us and our elders.
- This project aims to provide protection to elderly. Along with this, we are trying to make the elderly more comfortable and cope up with being away from their families.
- With our IoT project, the people can continue doing their work and reach out to their parents whenever they want.

MAIN PROJECT COMPONENTS



TOUCH SENSOR

To keep in check the medicine intake



RASPBERRY PI

Acts as a broker



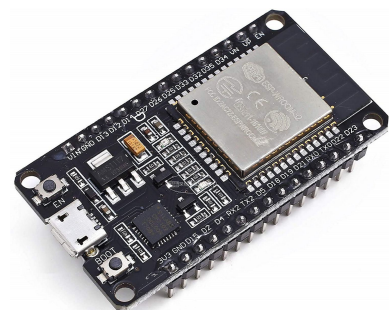
PANIC BUTTON

To send distress signals to guardians



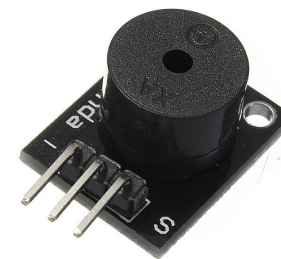
MONGODB

To store the medicine data and send alerts or reminders based on historical data



ESP

Microcontroller



BUZZER

Alarm for danger

02

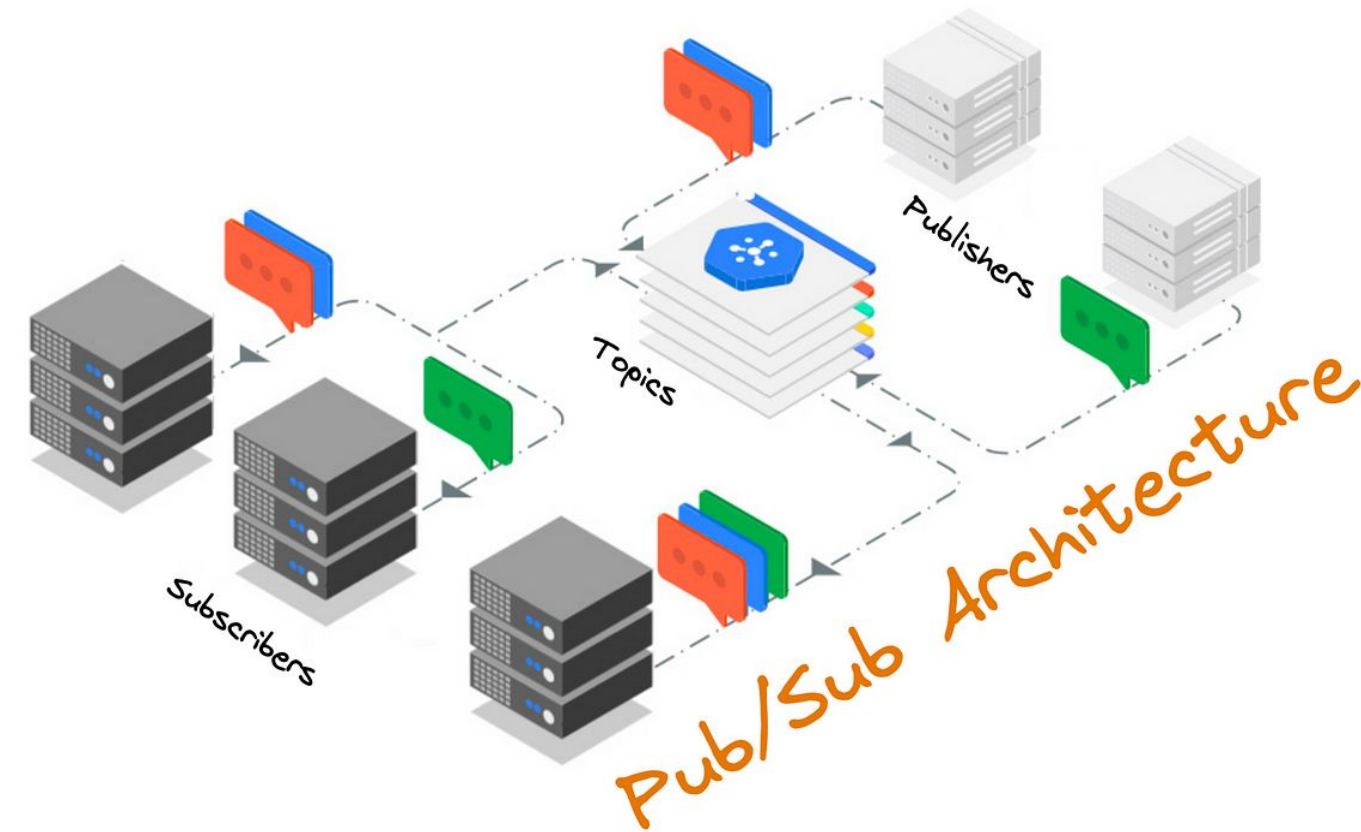
PROJECT

METHODOLOGY

Let's see how our Safety Shield actually works!



PUBLISH-SUBSCRIBE MODEL

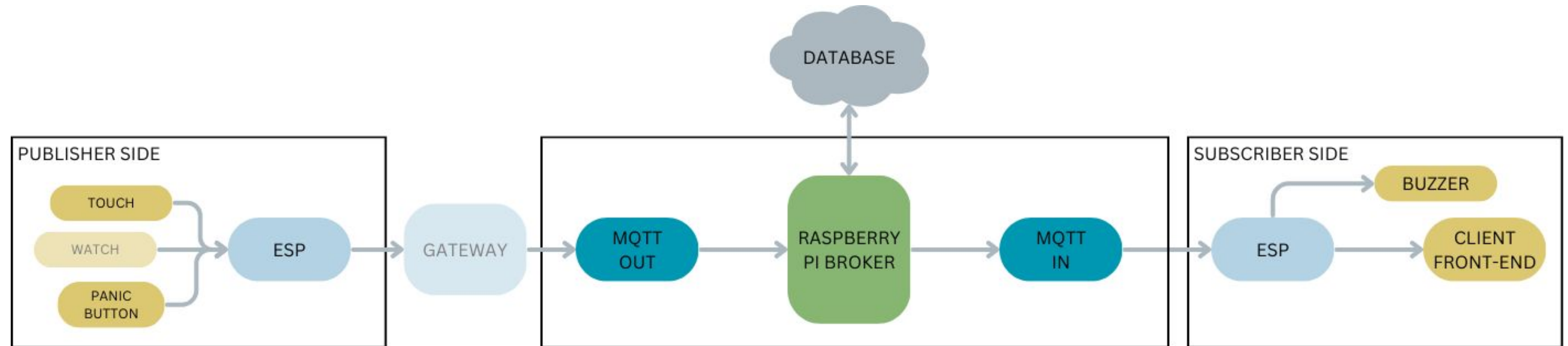


- Pub-Sub is a pattern used in distributed systems for asynchronous communication between different components or services.
- It provides a framework for exchanging messages between *publishers* and *subscribers*.
- The publishers don't send messages to specific subscribers in a point-to-point manner. Instead, an intermediary is used - a Pub-Sub message **broker**, which groups messages into entities called channels (or **topics**).

MQTT PROTOCOL

- Mainly we will be using MQTT protocol(Message Queuing Telemetry Transport) so as to connect many devices remotely.
- To get a rough idea of what MQTT is, we can take the well-known example of smart homes, that connects many devices such as lights, cctv, intrusion detection, motion and smoke sensors.
- MQTT works on the pub-sub principle and is operated via a central broker.
- The data sources report their data via a 'publish' and all recipients with interest in certain messages get the data delivered because they have registered as 'subscribers' ("marked by the **topic**").

WORKFLOW OF SAFETY SHIELD



1. **Publisher:** Old people or elderly, who will be having sensors with them.
2. **Subscriber:** Their guardians or doctors.
3. **Broker:** Raspberry pi
4. **Database provider:** MongoDB
5. **Sensors:** touch, panic button, buzzer
6. **Microcontroller:** ESP32
7. **Protocols:** MQTT, HTTP

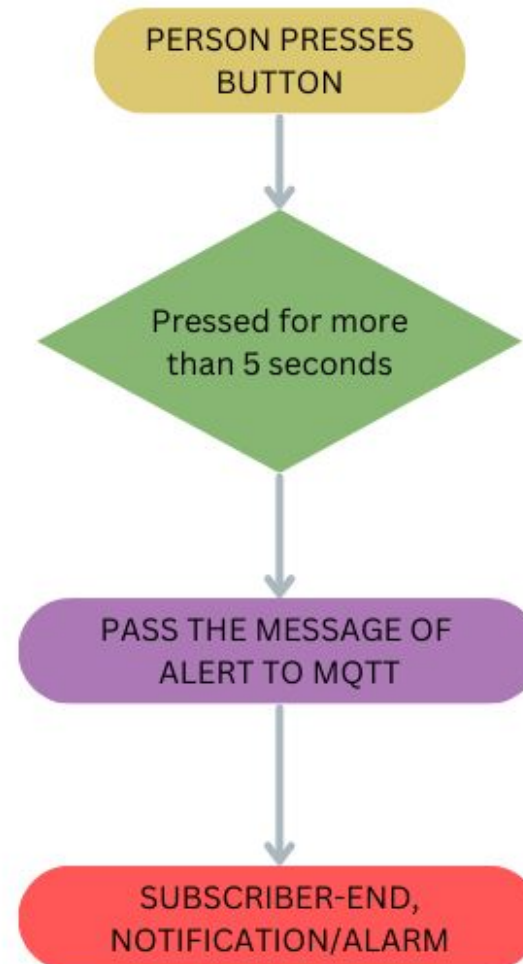
PUBLISHER SIDE: ELDERLY

Two types of data is collected:

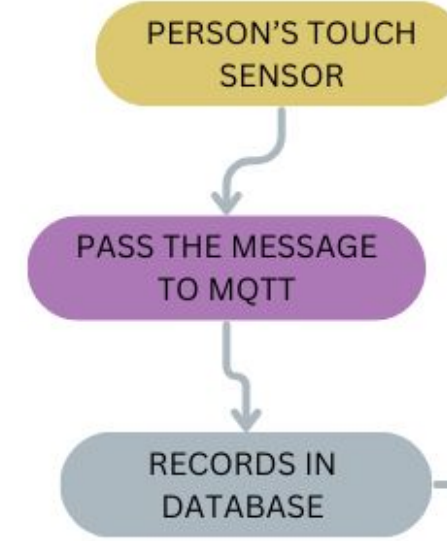
1. Medicine taken or not
2. Panic button pressed or not

The panic button may be sometimes mistakenly pressed, so a trigger alert of 4-5 seconds is kept.

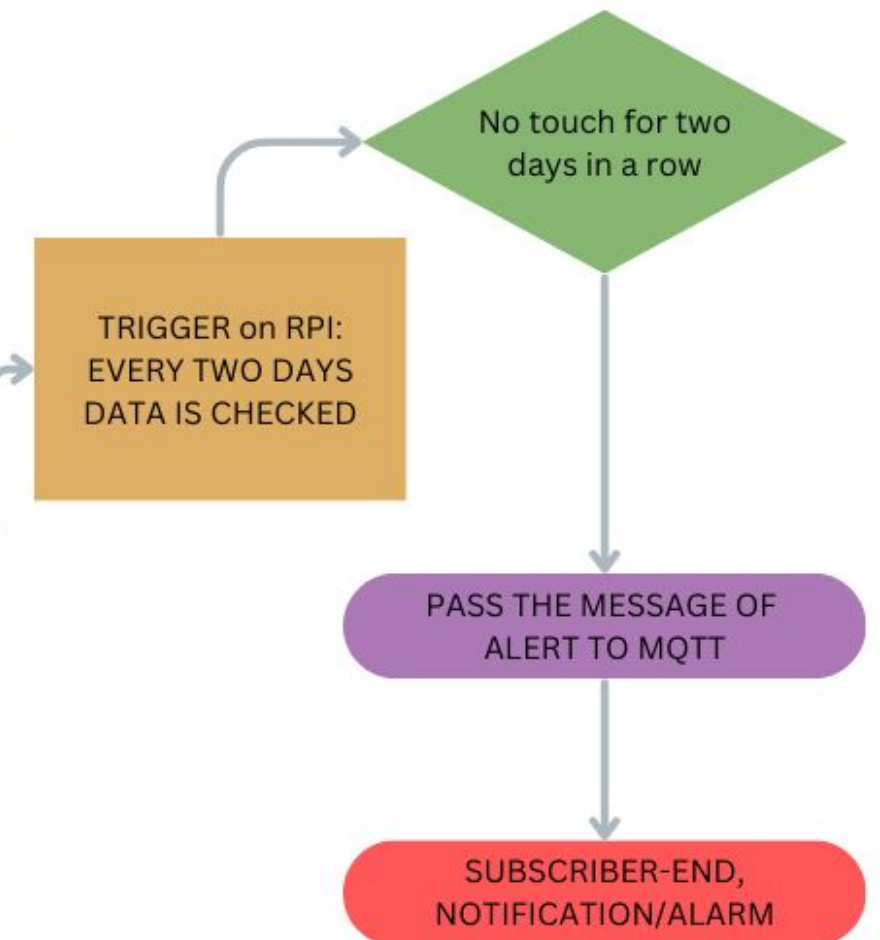
PANIC ALERT THROUGH BUTTON



MEDICINE INTAKE STORAGE



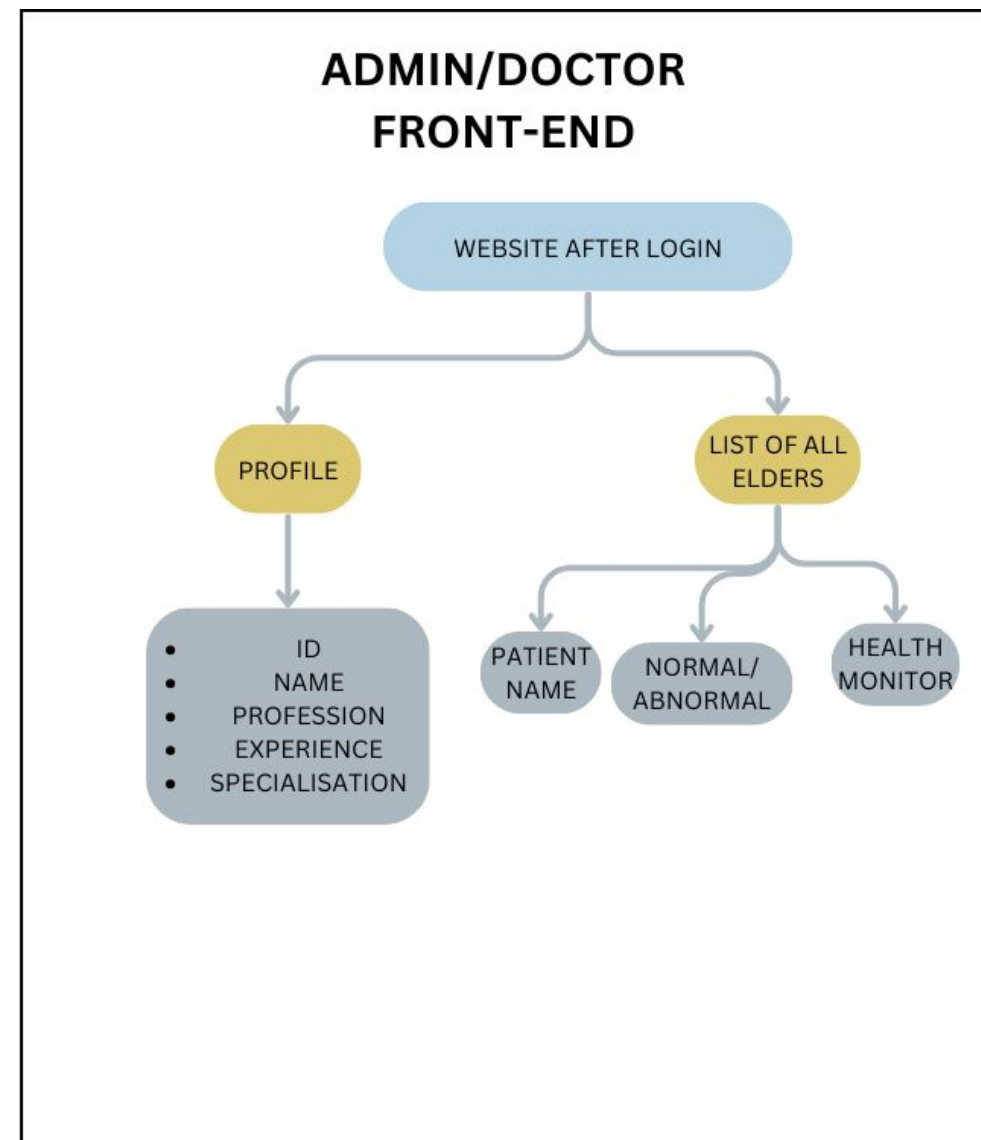
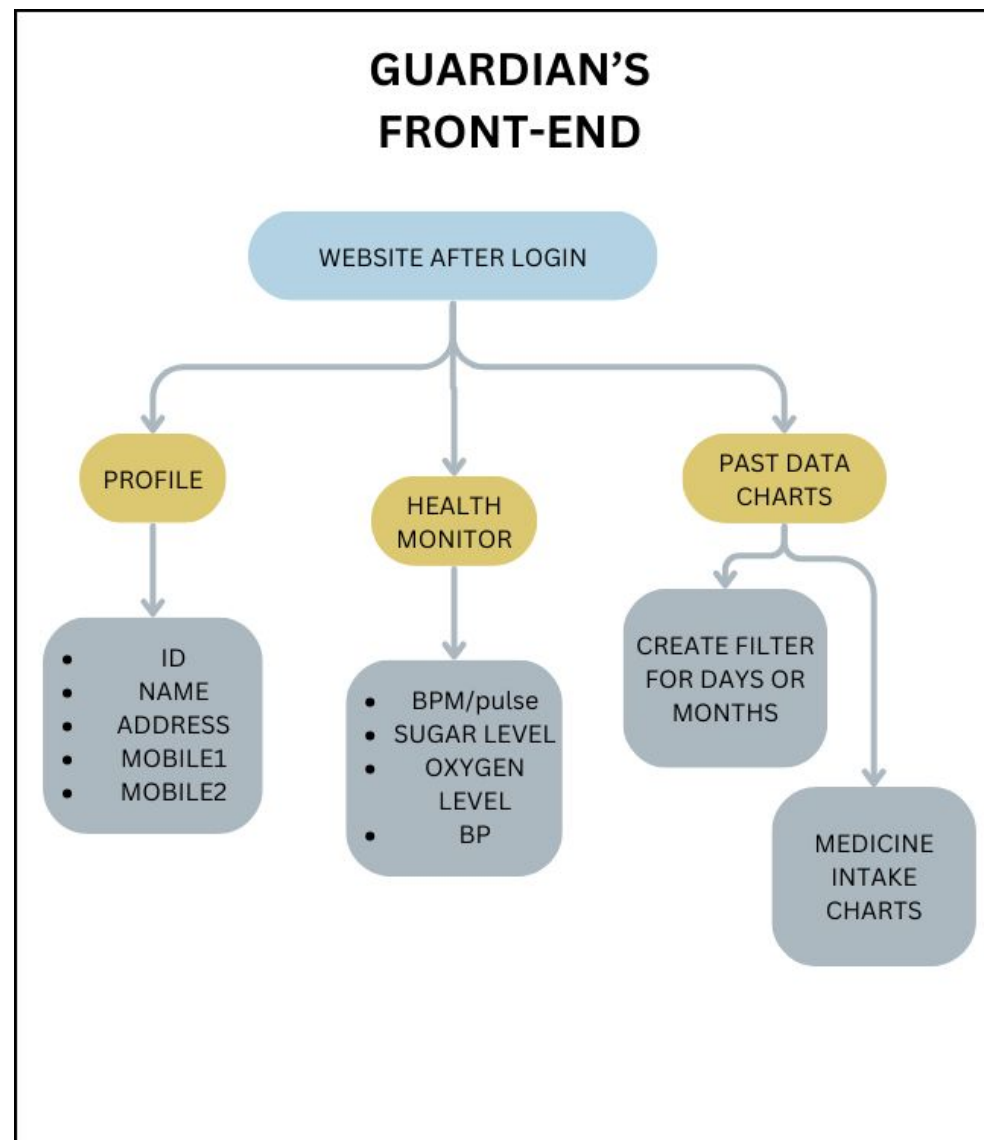
MEDICINE TRACKER ALERTS



BROKER SIDE: RASPBERRY PI

1. If person misses medicines for two or more days, then alerts will be sent to the respective doctor allotted and to their guardians. This is where we will be integrating the historical data and real time data analysis.
2. Similarly, we can check whether the panic button has been pressed for more than 5 seconds to judge whether it is really a distress signal or a mistake.
3. These both aspects - medicine check and panic alert are then published to a topic, which will be subscribed by the guardians' side.
4. The real time data can be visualized through the interactive dashboards using Node RED or any web server.

SUBSCRIBER SIDE: GUARDIAN



- Reaching towards the end, we have our final front-end application for doctors and guardians separately.
- We can see that the individual user's details can be viewed by their respective guardians.
- Also, a doctor may view all the elders under him/her to judge them on an overall basis.

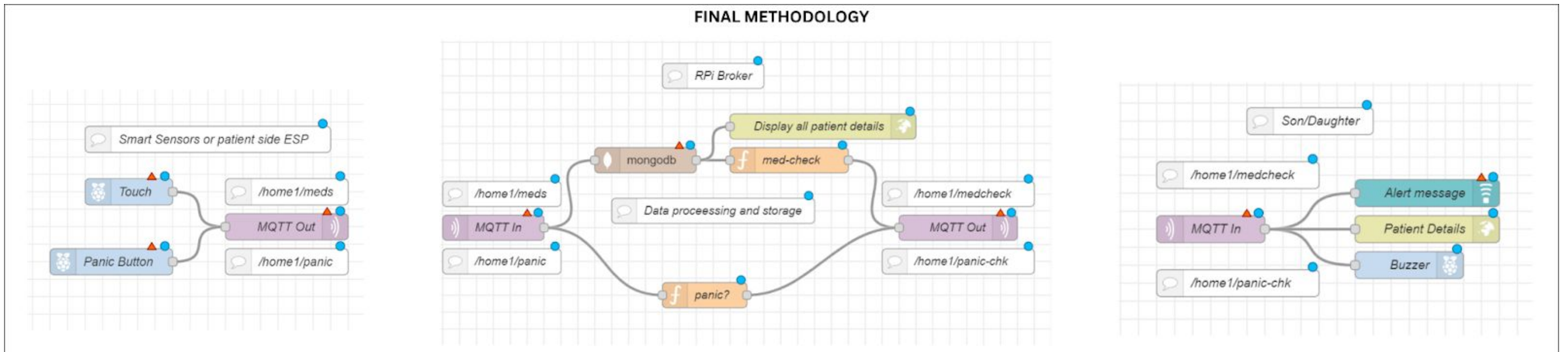


03

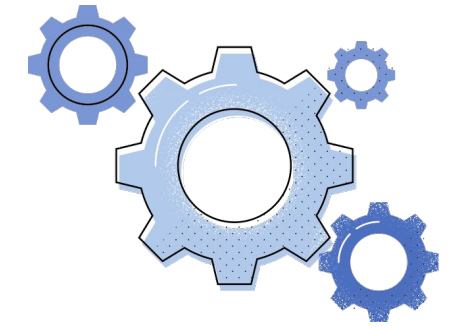
PROJECT

IMPLEMENTATION

We have created a simple overview of how the communication will be established.



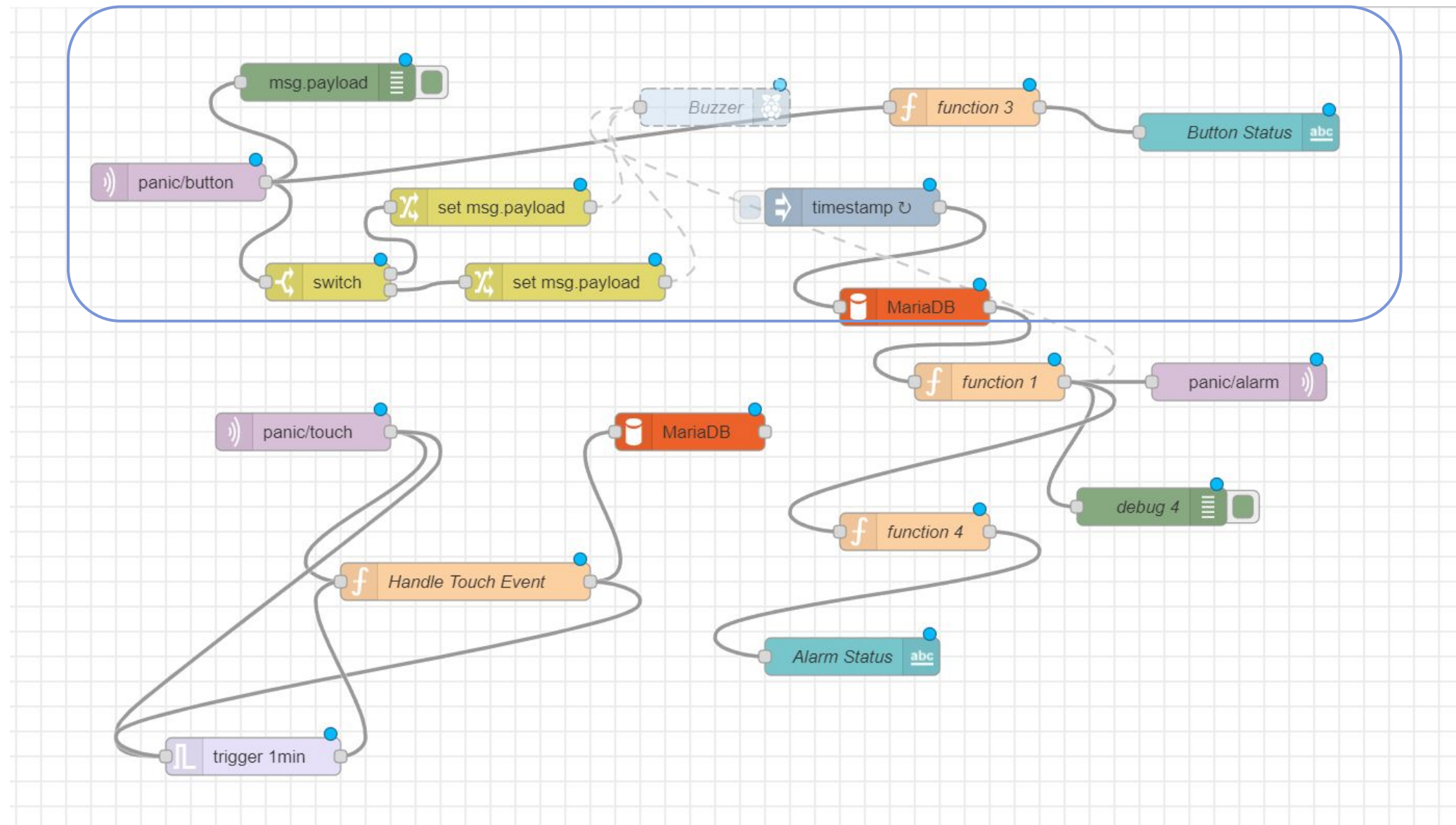
PANIC BUTTON FLOW



Panic button is pressed. A message is sent, 'pressed' or 'not pressed'.

This data is exported to dashboard as well.

Before putting up the subscriber ESP, we tried the same flow using buzzer which worked successfully.



BUTTON DATA

- The button data is directly sent to subscriber ESP and to Raspberry Pi through topic(panic/button) via an open-source MQTT broker, **emqx**.
- Subscriber ESP has connected a red LED symbolizing an alarm, which will glow on sending 'pressed' payload.
- Dashboard will also receive this information and will show the same.

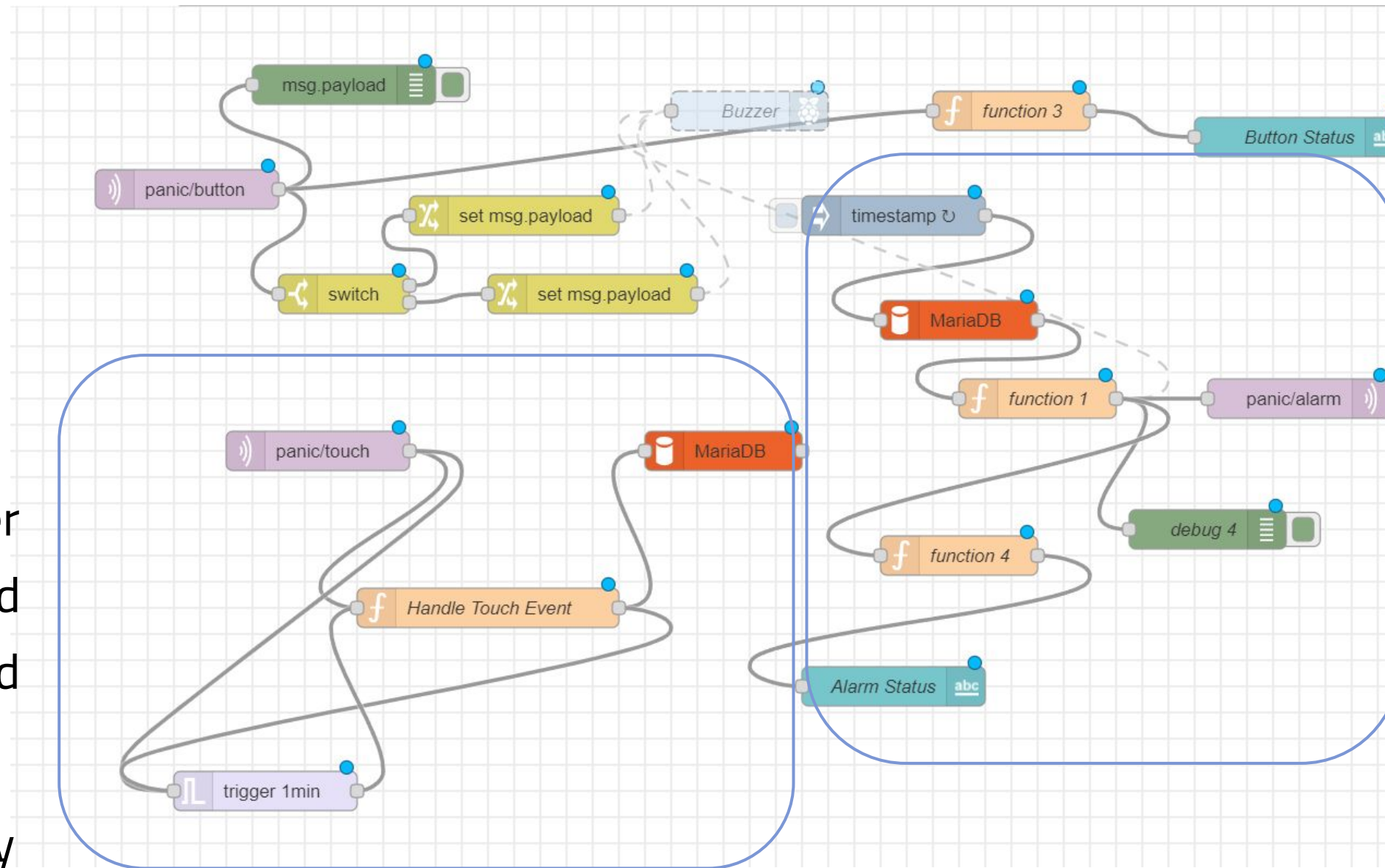


TOUCH SENSOR FLOW

Enters touch data.

Function will enter either touched/ not touched depending on payload received.

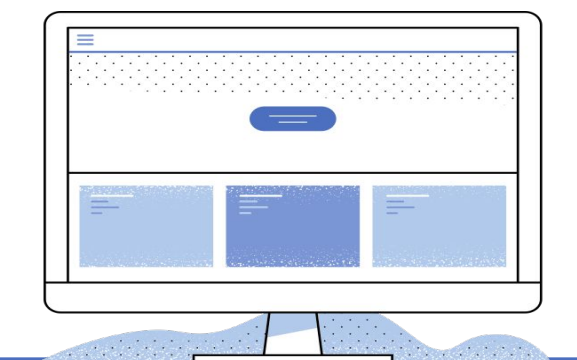
Trigger resets after every touch. If no touch for a minute, it writes not touched to DB.



Every second the database is checked if last two entries are 'not touched'. That means medicine not taken for two minutes.

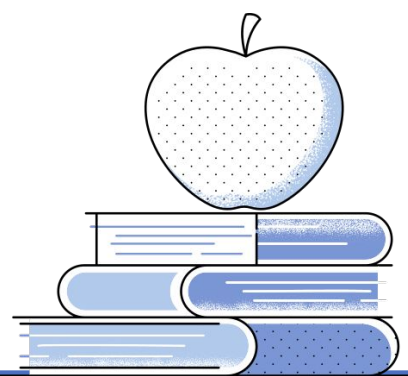
A message for alarm is sent through MQTT.

Another function is for displaying alarm status on dashboard.



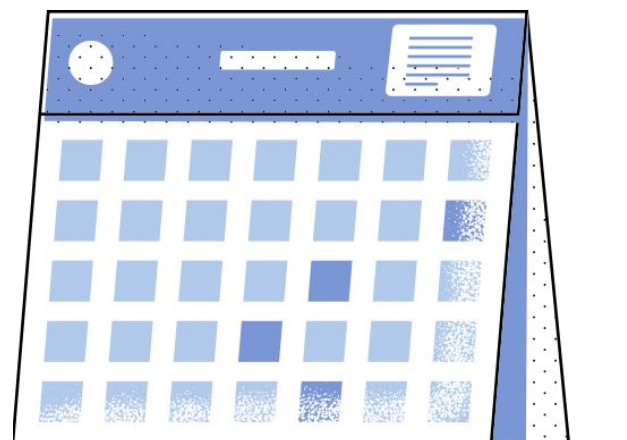
TOUCH DATA

- The touch data will be collected to record whether medicine is taken or not.
- We have used mariadb as database to store the touch data. We couldn't use the mongoDB due to raspberry pi OS constraints.
- The database stores "touched" with timestamp whenever the box is touched. If the box is not touched for two days in row, then the alarm will beep and alert the guardian.
- To showcase this, we have scaled the a medicine 'touch per day' to a 'touch per minute' and 'medicine not taken for two days' to 'no touch received for past two minutes'.
- Historical and real time data is being merged to give the alarm to guardian.



DATABASE

- This is how we get the data stored in MariaDB.
- We can see the timestamps changing according to given conditions of 'touched' and 'not touched'.

A screenshot of a MariaDB terminal window. The window has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. The main area displays a table with two columns: a timestamp and a status. The status is either 'touched' or 'not touched'. The table is sorted by timestamp. At the bottom, it shows '95 rows in set (0.007 sec)' and the prompt 'MariaDB [touchData]>'.

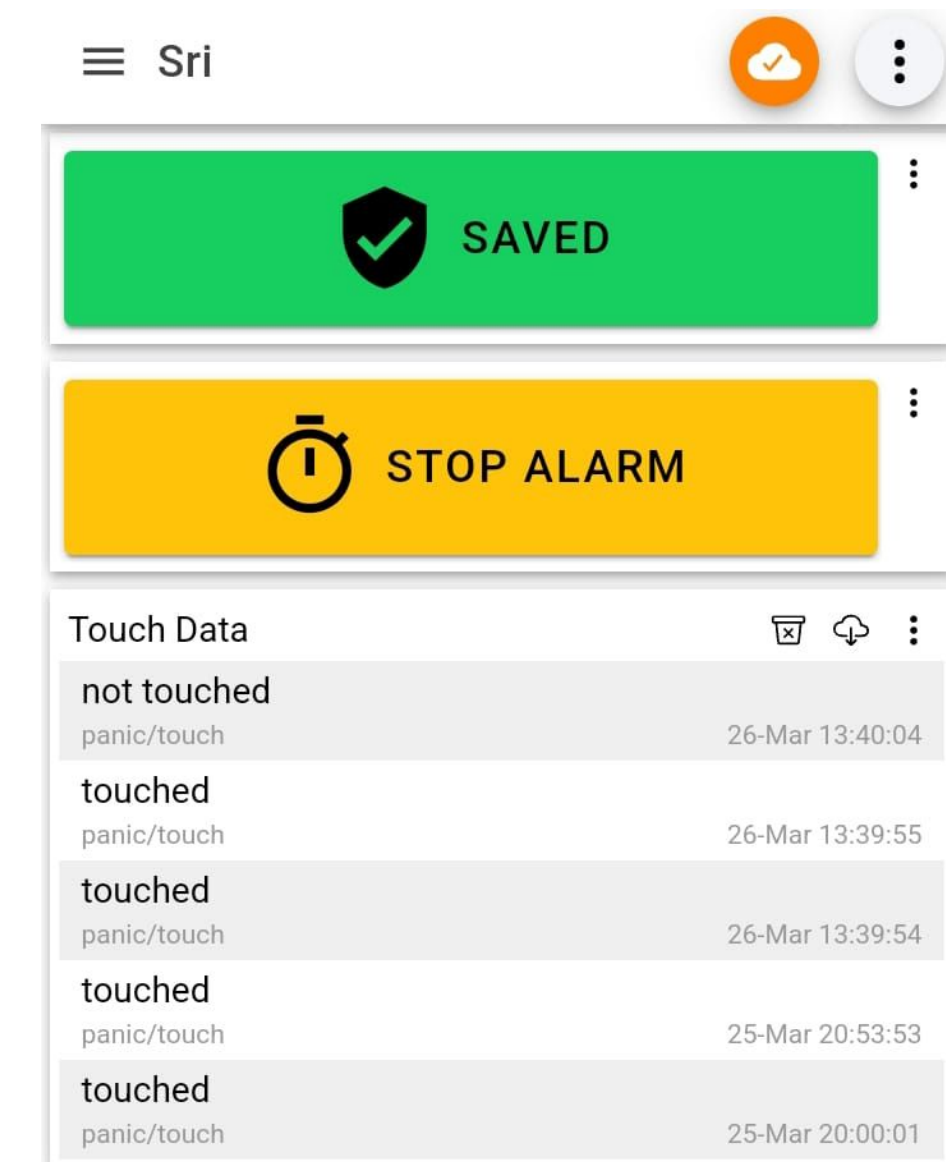
Timestamp	Status
2024-03-25 19:19:57	touched
2024-03-25 19:19:59	touched
2024-03-25 19:20:01	touched
2024-03-25 19:21:19	touched
2024-03-25 19:22:07	touched
2024-03-25 19:22:09	touched
2024-03-25 19:23:09	not touched
2024-03-25 19:24:09	not touched
2024-03-25 19:24:17	touched
2024-03-25 19:25:17	not touched
2024-03-25 19:26:17	not touched
2024-03-25 19:26:22	touched
2024-03-25 19:26:26	touched
2024-03-25 19:27:26	not touched
2024-03-25 19:28:26	not touched
2024-03-25 19:28:40	touched
2024-03-25 19:42:12	touched
2024-03-25 19:43:12	not touched
2024-03-25 19:44:12	not touched
2024-03-25 19:44:25	touched
2024-03-25 19:44:26	touched
2024-03-25 19:45:26	not touched
2024-03-25 19:46:26	not touched
2024-03-25 19:46:52	touched
2024-03-25 19:47:52	not touched
2024-03-25 19:48:52	not touched
2024-03-25 19:49:52	not touched
2024-03-25 19:50:52	not touched
2024-03-25 19:51:52	not touched
2024-03-25 19:52:52	not touched
2024-03-25 20:00:02	touched
2024-03-25 20:01:02	not touched
2024-03-25 20:02:02	not touched
2024-03-25 20:03:02	not touched
2024-03-25 20:04:02	not touched
2024-03-25 20:05:02	not touched
2024-03-25 20:06:02	not touched
2024-03-25 20:07:02	not touched
2024-03-25 20:08:02	not touched
2024-03-25 20:53:53	touched
2024-03-25 20:54:53	not touched
2024-03-26 17:16:02	touched
2024-03-26 17:16:04	touched

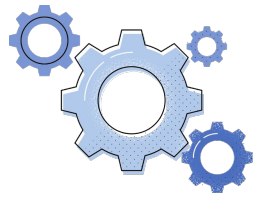
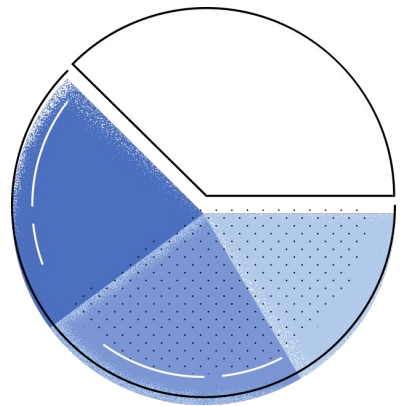
MQTT CLIENT APP

We have attached all the data to a client app, which shows:

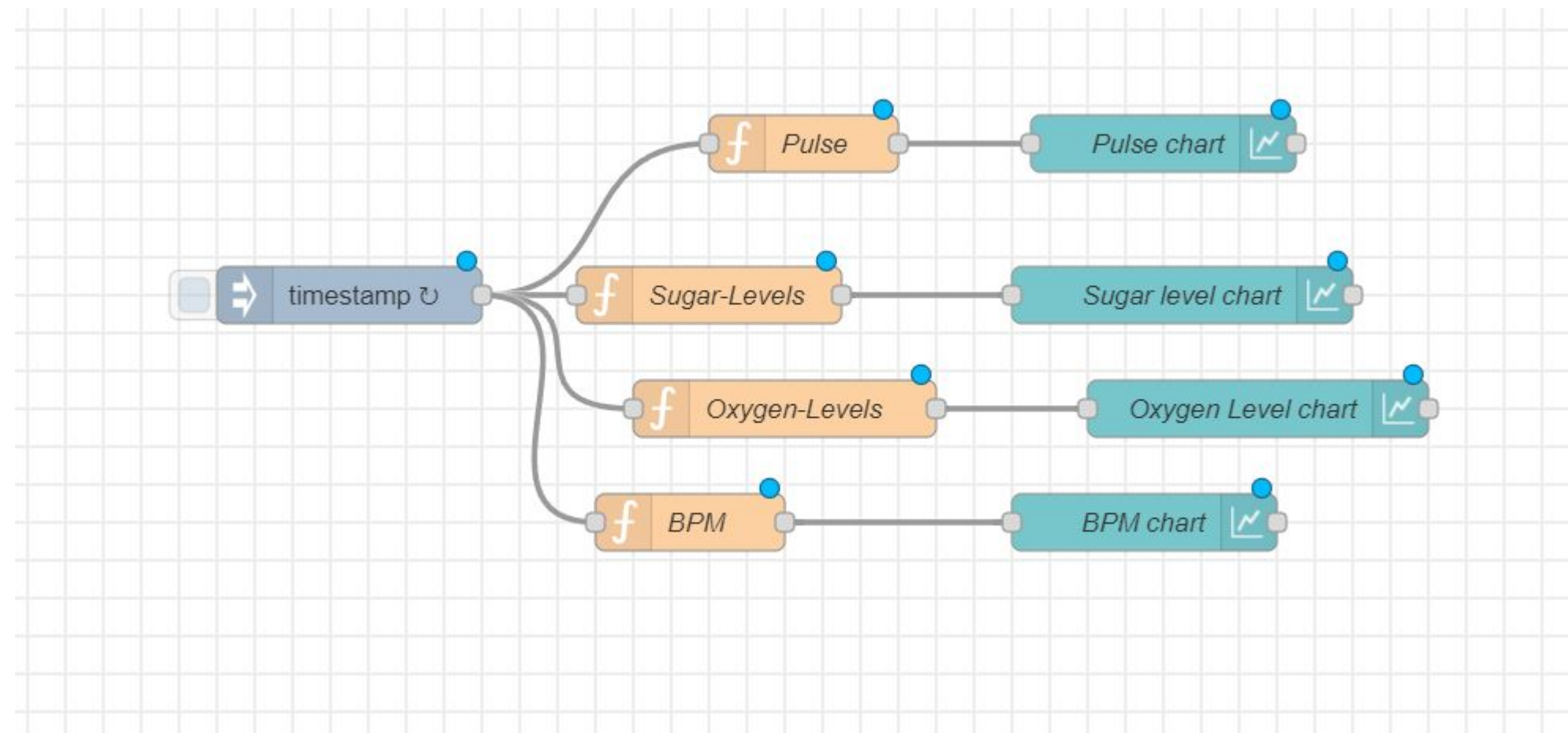
- Information on touch
- Button to stop the Panic Alarm
- Button to stop the Medicine Alert (when medicine is not taken for two days)

This gives the guardians an overview of when the elderly are taking medicine.





NODE-RED DASHBOARD



- The dashboard will show the data on whether the medicine is taken or not.
- Also it will show the alarm(panic button) status.
- Along with that, we have also tried to integrate some dummy data such as Pulse, Sugar and oxygen levels, and BPM, to be shown as charts to the guardian on the dashboard.



04

FUTURE

POTENTIALS

Looking ahead, we see great potential for improvement in specific areas, such as:

1. Integrating our dashboard with a fully deployed web application.
2. Integrating real time health data with smart watches.



OUR TEAM



K. Vyshnavi

PULLA REDDY COLLEGE
CSE Dept.

S. Afsha

PULLAIAH COLLEGE
CSE Dept.

E. Rishitha

IIITDMK
CSE Dept.

M. Deepika

IIITDMK
CSE Dept.

M Bhuvana

IIITDMK
CSE Dept.

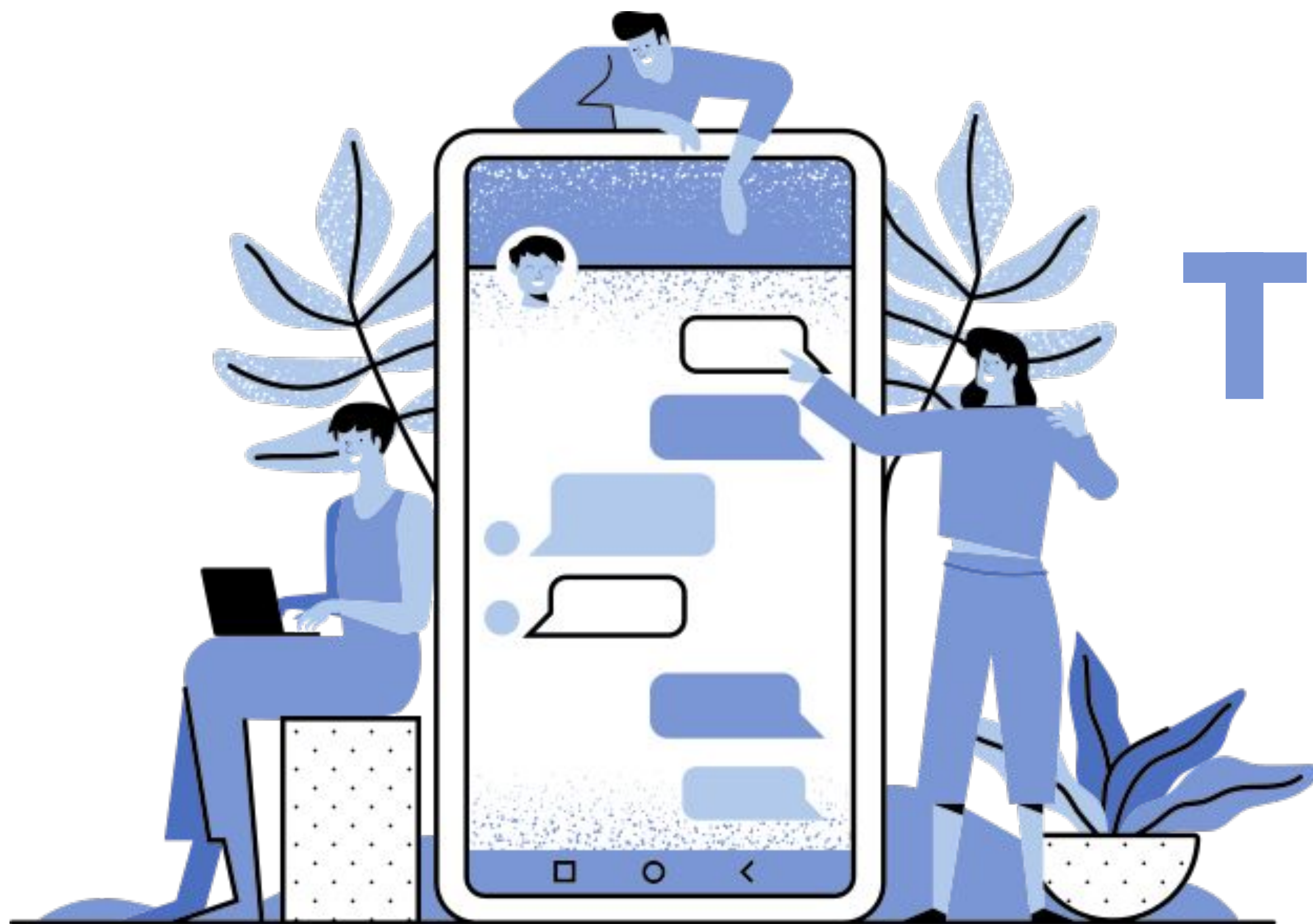
N Farda

RAVINDRA COLLEGE
CSE Dept.

G Sri Sougandhika

IIITDMK
CSE Dept.





**THANK
YOU**