MENMO

A Smart Reminder For Alzheimer’s

BRIEF

To design a smart reminder device for people with Alzheimer’s, aiming to support their daily routines and ensure safety. Our vision was to go beyond a reminder tool and provide a sense of companionship, while also reducing the need for constant caregiver monitoring.”

USER GROUP

Primary Users: People with Alzheimer’s (mainly 60+, facing memory loss, confusion, and loneliness).
Secondary Users: Caregivers (family/nursing staff who need reduced supervision stress).
Tertiary Users: Doctors, elderly homes, and support communities (for patient tracking and care).

WHAT IS ALZHEIMER'S?

Alzheimer's is a progressive brain disorder that affects memory, thinking, and behavior. It is the most common cause of dementia. Over time, it impacts a person's ability to carry out daily activities and increases dependence on caregivers.

- Memory Loss & Confusion – forgetting people, places, tasks.
- Daily Routine Challenges – forgetting meals, medication, meetings
- Communication Issues – trouble finding words or following conversations.
- Disorientation – getting lost, wandering, difficulty recognizing surroundings.
- Behavioral & Emotional Changes – anxiety, agitation, loneliness, depression.

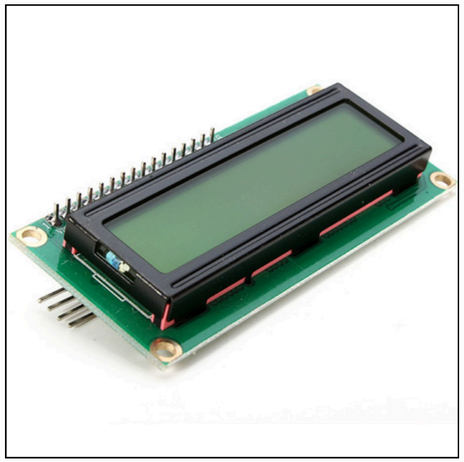
CONSTRAINTS:

The major challenges we faced were **keeping the device compact** while integrating multiple features, debugging custom PCBs where even small errors disrupted the system, and **developing a app** for the first time that could seamlessly track and update patient details. **Learning to code** from scratch and ensuring smooth sensor–microcontroller communication added to the complexity, making the project both demanding and rewarding.

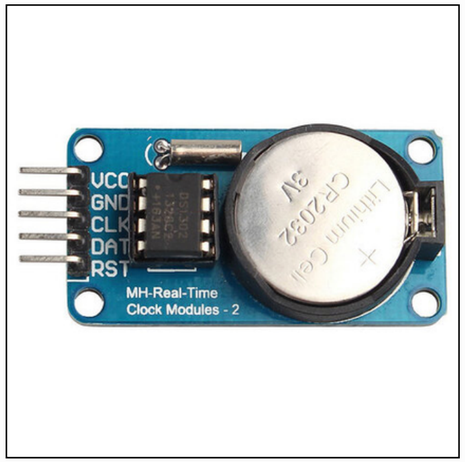
On basis of this we made a priority list of features :

1. Real-time GPS tracking (location visible to caregivers).
2. Geofencing alerts (notify caregiver if patient leaves a safe zone)
3. Routine reminders
4. Day & time orientation (clear display or voice prompt).
5. Friendly/motivational messages throughout the day.
6. Simple interface (large buttons, high contrast, loud audio).
7. Wearable / portable design (watch, pendant, or small device).
8. Mobile app dashboard (to track location, set reminders, send messages).
9. Remote reminder updates (caregiver can edit without patient input).
10. Instant notifications for wandering or SOS.

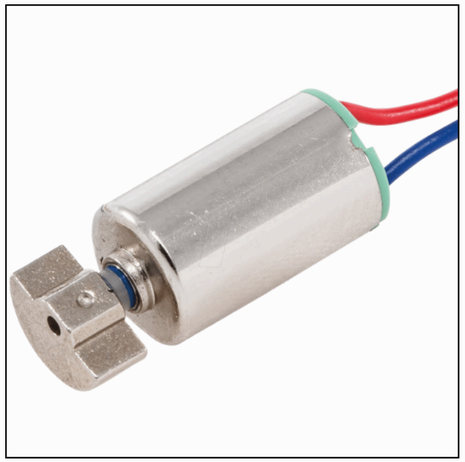
COMPONENTS REQUIRED:



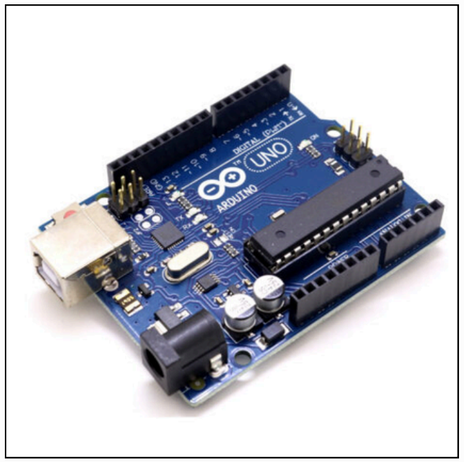
OLED display module



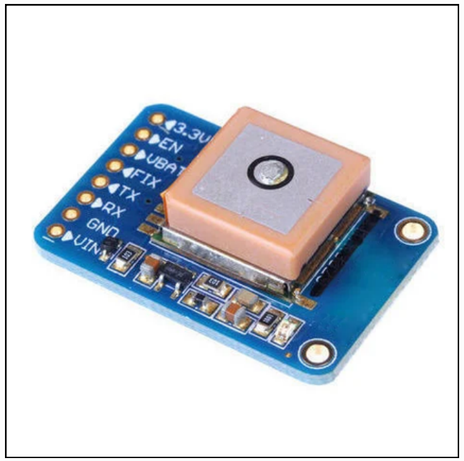
RTC Module



Vibrating Motar

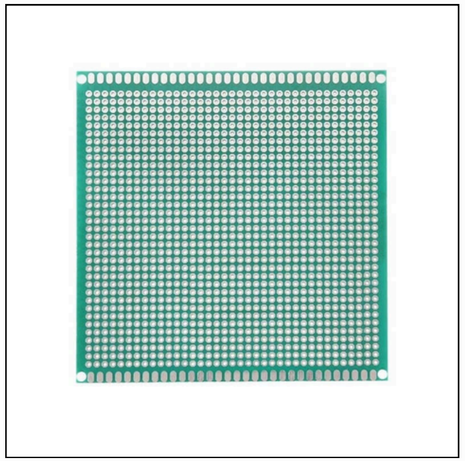


ESP 32 Micro Cont.



GPS Module

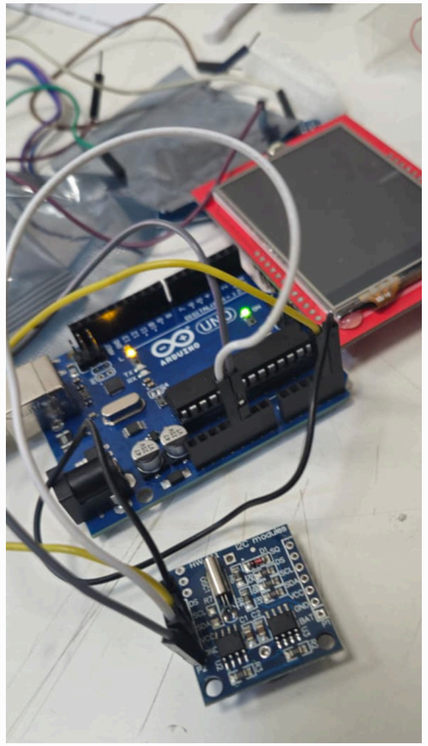
Push Button

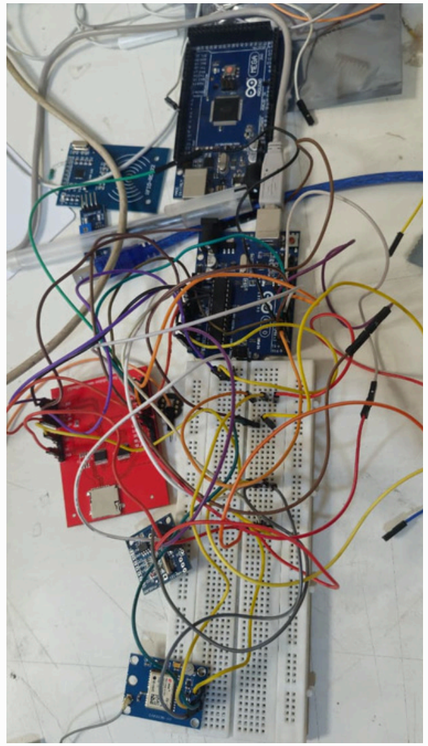


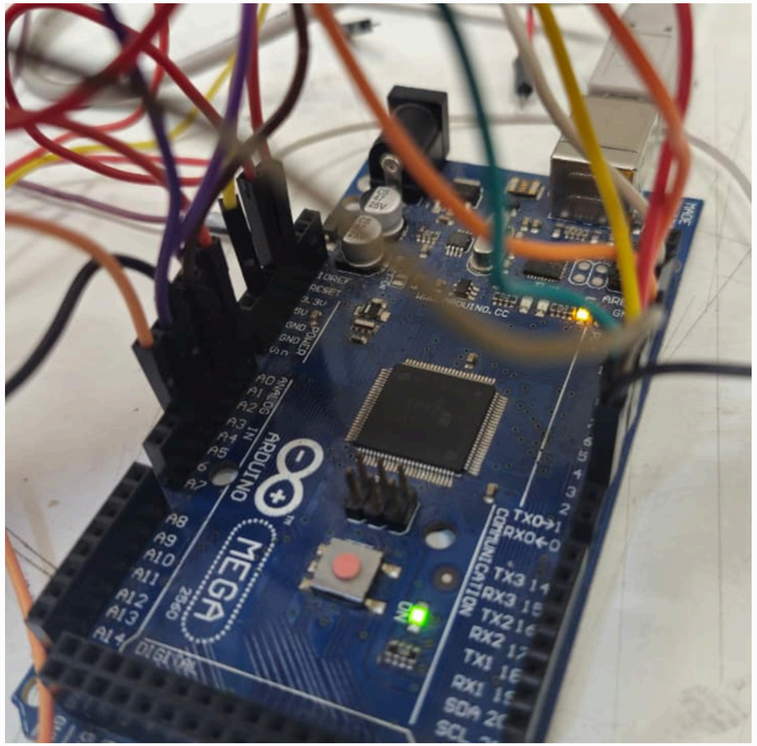
Vero Board

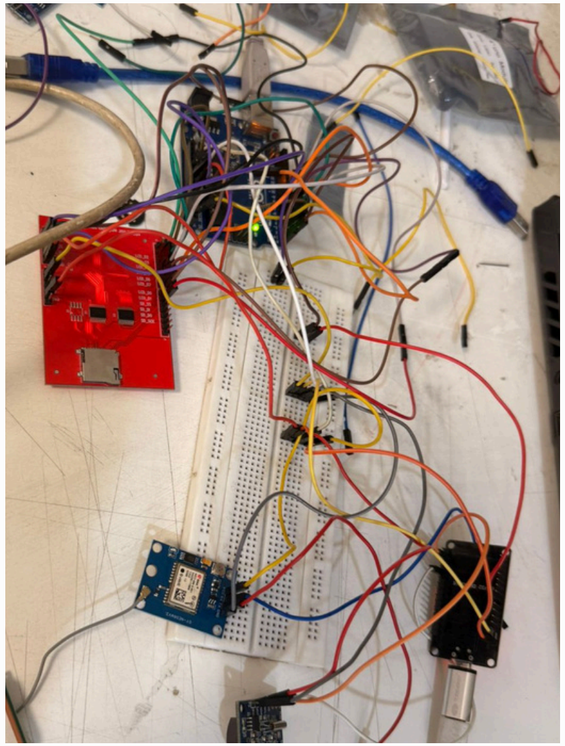
Piezo Buzzer

TRAIL N ERROR







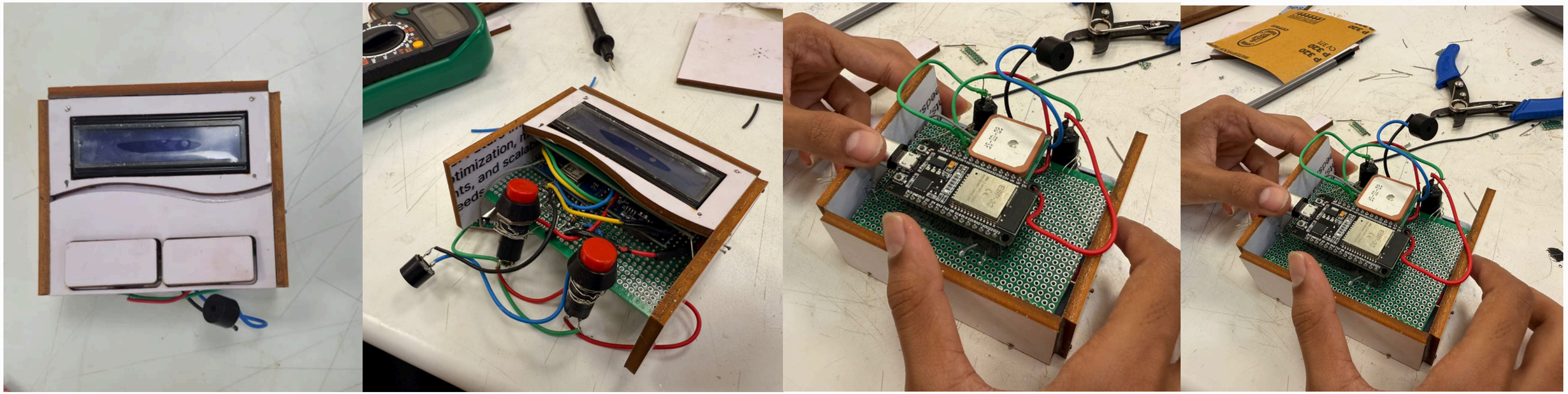


We first planned to use an OLED with ESP32 for Wi-Fi and Bluetooth, but the display only worked with Arduino. On Arduino Uno, pin limitations forced us to shift to Arduino Mega, which solved the display issue but lacked connectivity. Connecting ESP32 to Mega worked but reduced compactness, so we finally switched to a 16x2 I2C LCD, which required fewer pins and kept the design simpler.

CODE:

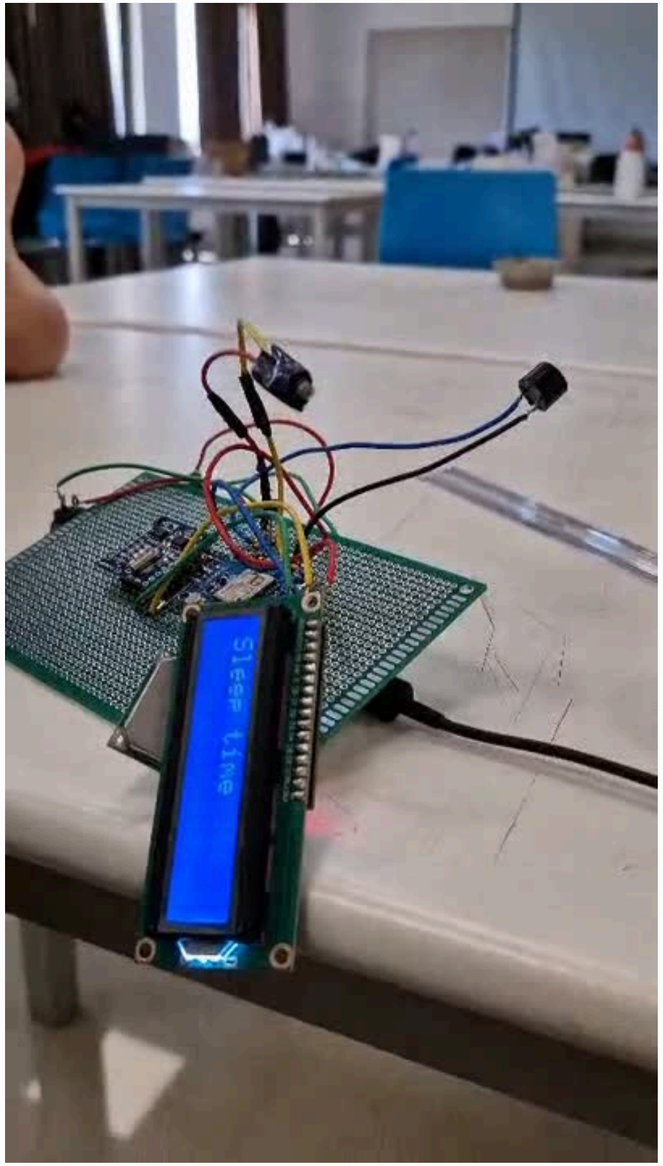
```
#include <Wire.h>
#include "RTClib.h"
#include <LiquidCrystal_I2C.h>

RTC_DS3231 rtc;
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Pins
const int BUZZER_PIN = 13;
const int MOTOR_PIN  = 14;
const int YES_PIN    = 26;
const int NO_PIN     = 27;

// LEDC channels
const int BUZZER_CH = 0;
const int MOTOR_CH  = 1;

// Reminder time
const int REM_H1 = 22;
const int REM_M1 = 30;

// Durations
const unsigned long REM_DURATION = 60000UL; // 1 min
const unsigned long SNOOZE_DELAY = 180000UL; // 3 min

// Melody
int melody[] = {523, 659, 784, 1046}; // C E G C
int dur[]    = {200, 200, 200, 300};
int notes = sizeof(melody) / sizeof(melody[0]);

// Snooze tracking
unsigned long snoozeUntil = 0;

void setup() {
  Serial.begin(115200);
  Wire.begin();
  lcd.init();
  lcd.backlight();

  // RTC
  if (!rtc.begin()) {
    Serial.println("RTC not found");
    lcd.clear(); lcd.print("RTC not found");
    while (1);
  }

  // Buzzer
  ledcSetup(BUZZER_CH, 2000, 10);
  ledcAttachPin(BUZZER_PIN, BUZZER_CH);

  // Motor
  ledcSetup(MOTOR_CH, 500, 8);
  ledcAttachPin(MOTOR_PIN, MOTOR_CH);
  ledcWrite(MOTOR_CH, 0);

  // Buttons
  pinMode(YES_PIN, INPUT_PULLUP);
  pinMode(NO_PIN, INPUT_PULLUP);
}

void loop() {
  DateTime now = rtc.now();
  int h = now.hour(), m = now.minute(), s = now.second();

  // Show time
  lcd.setCursor(0,0);
  char timestr[17];
  snprintf(timestr, sizeof(timestr), "%02d:%02d:%02d", h, m, s);
  lcd.print(timestr);
  Serial.println(timestr);

  if (s == 0) { // check only at start of each minute
    // Snooze case
    if (snoozeUntil > 0 && millis() >= snoozeUntil) {
      snoozeUntil = 0;
      int result = showReminder("Sleep time");
      if (result == 2) snoozeUntil = millis() + SNOOZE_DELAY; // NO
    }
    // Scheduled reminder case
    else if (h == REM_H1 && m == REM_M1) {
      int result = showReminder("Sleep time");
      if (result == 2) snoozeUntil = millis() + SNOOZE_DELAY; // NO
    }
  }

  delay(200);
}

// showReminder returns:
// 1 = YES pressed (stop)
// 2 = NO pressed (snooze)
int showReminder(const char *msg) {
  unsigned long start = millis();
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(msg);

  while (millis() - start < REM_DURATION) {
    // Check buttons
    if (digitalRead(YES_PIN) == LOW) {
      stopAlerts();
      return 1; // stop
    }
    if (digitalRead(NO_PIN) == LOW) {
      stopAlerts();
      return 2; // snooze
    }

    // Play melody
    for (int i = 0; i < notes && (millis() - start < REM_DURATION); ++i) {
      if (digitalRead(YES_PIN) == LOW) { stopAlerts(); return 1; }
      if (digitalRead(NO_PIN) == LOW) { stopAlerts(); return 2; }

      ledcWriteTone(BUZZER_CH, melody[i]);
      ledcWrite(MOTOR_CH, 180);
      delay(dur[i]);

      ledcWriteTone(BUZZER_CH, 0);
      ledcWrite(MOTOR_CH, 0);
      delay(60);
    }

    delay(500);
  }

  stopAlerts();
  return 1; // auto stop if no button pressed
}
```

RAPID PROTOTYPE:

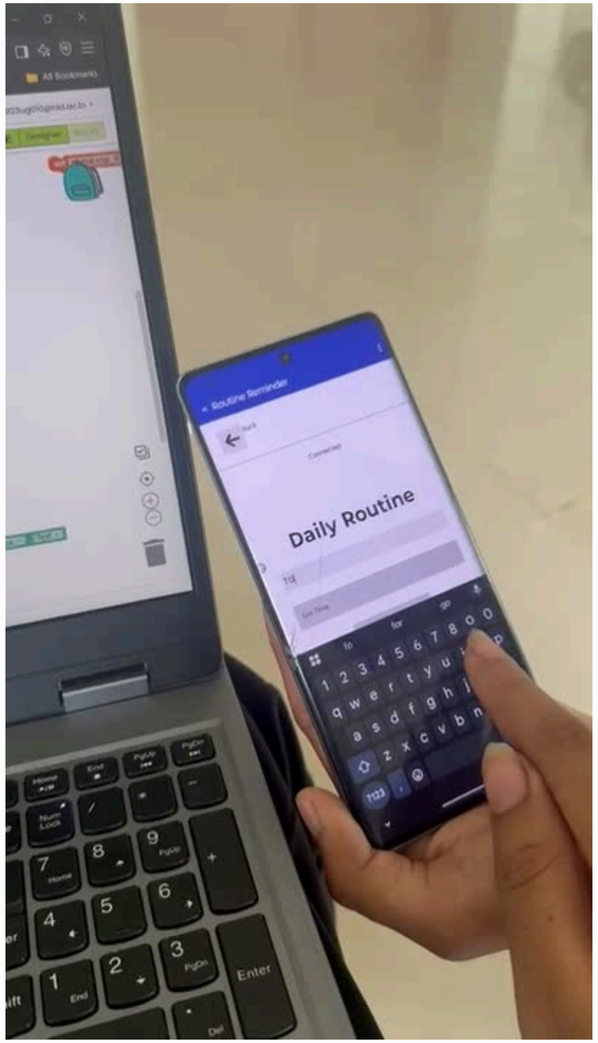


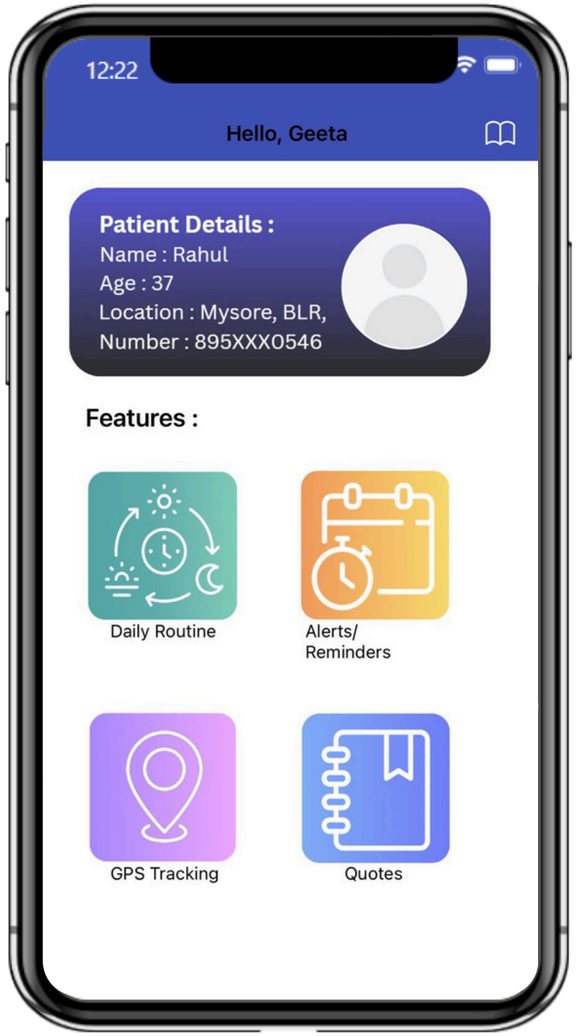
From our prototype testing, we learned that the device needs to be **more compact and lightweight**, with better ergonomic **grip zones** using textured or rubberized surfaces for elderly hands. The buttons should be larger, tactile, and well-placed, with **visual feedback** for easy use. A high-contrast, **big-font display** are essential for accessibility. Comfort and wearability of a product. Caregivers highlighted the need for **intuitive app controls** and clear alerts, while patients appreciate the **motivational** and personal messages, making emotional support a key feature to retain.

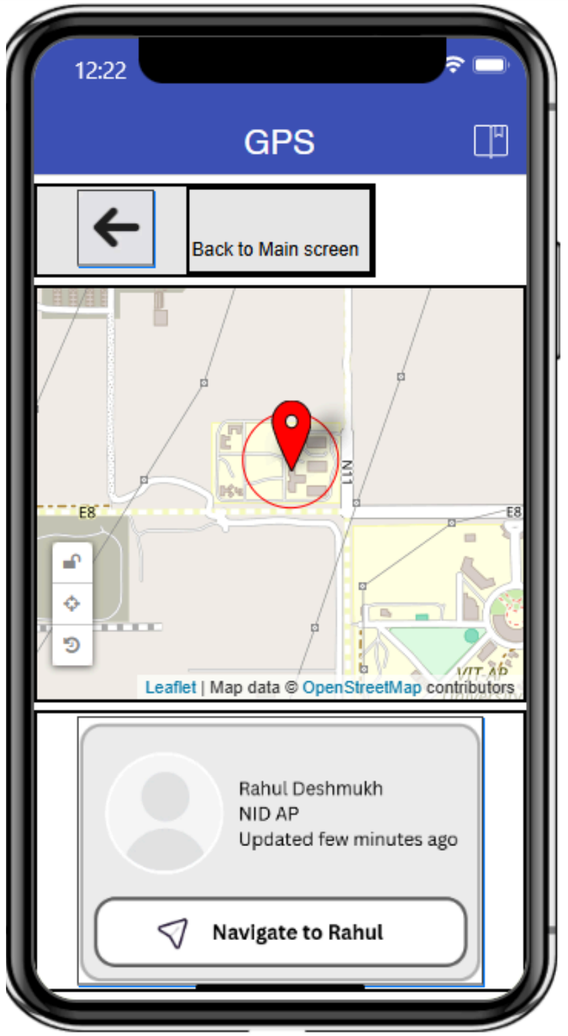
RAPID PROTOTYPE:

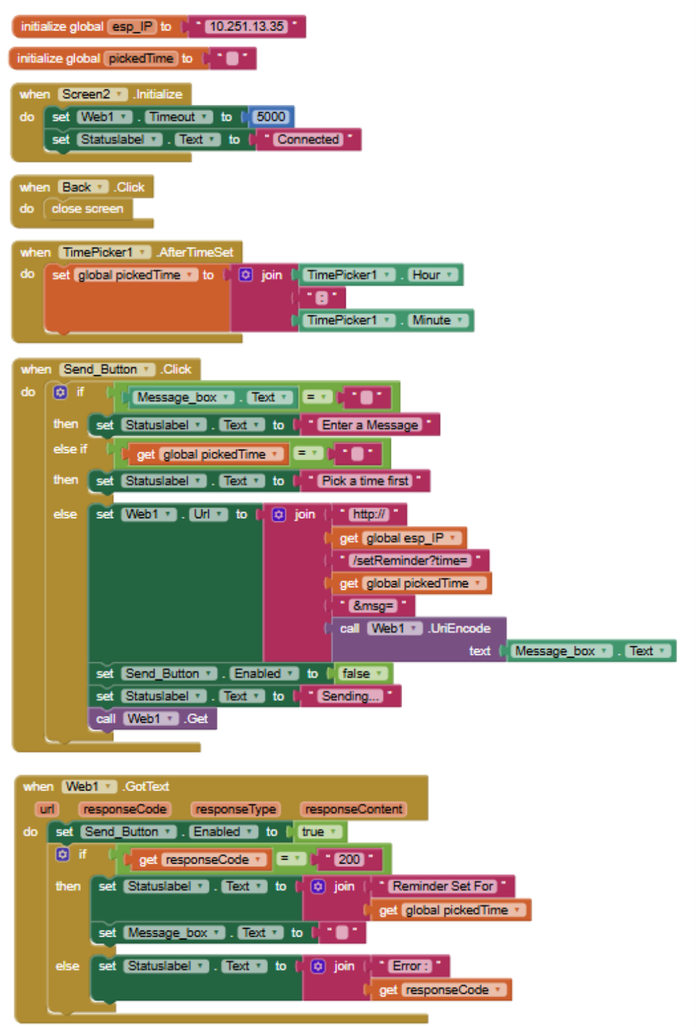


APP


Daily Routine


12:22
Hello, Geeta
Patient Details :
Name : Rahul
Age : 37
Location : Mysore, BLR,
Number : 895XXX0546
Features :
Daily Routine
Alerts/
Reminders
GPS Tracking
Quotes


12:22
GPS
Back to Main screen
Leaflet | Map data © OpenStreetMap contributors
Rahul Deshmukh
NID AP
Updated few minutes ago
Navigate to Rahul


initialize global esp_IP to 10.251.13.35
initialize global pickedTime to
when Screen2 .Initialize
do set Web1 . Timeout to 5000
set Statuslabel . Text to Connected
when Back .Click
do close screen
when TimePicker1 .AfterTimeSet
do set global pickedTime to join TimePicker1 . Hour : TimePicker1 . Minute
when Send_Button .Click
do if Message_box . Text =
then set Statuslabel . Text to Enter a Message
else if get global pickedTime =
then set Statuslabel . Text to Pick a time first
else set Web1 . Url to join http:// get global esp_IP /setReminder?time= get global pickedTime &msg= call Web1 .UriEncode text Message_box . Text
set Send_Button . Enabled to false
set Statuslabel . Text to Sending...
call Web1 .Get
when Web1 .GotText
url responseCode responseType responseContent
do set Send_Button . Enabled to true
if get responseCode = 200
then set Statuslabel . Text to join Reminder Set For get global pickedTime
set Message_box . Text to
else set Statuslabel . Text to join Error : get responseCode

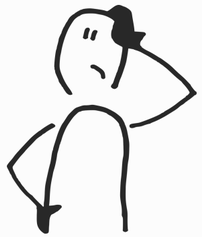
CMF :

Form – compact, rounded, and friendly in appearance to reduce intimidation.
Interface – only two large buttons (Yes/No) for a simple, confusion-free experience.
Colour Palette – pastel purple & yellow, chosen for a warm, cheerful, and calming vibe while ensuring visibility.
Material Finish – textured grip zones for secure handling by elderly users.
Accessibility – high-contrast fonts, and ergonomic button placement.

User Alignment:

Alzheimer's patients – need simplicity, reassurance, and emotional comfort → achieved with soft colors, friendly form, and easy Yes/No interface.
Caregivers – need peace of mind & control → achieved through GPS, alerts, and app-based remote management.

HOW IT WORKS:



Caregiver Setup – Caregiver uses a mobile app to set reminders (medicine, meals, hydration, activities) and define safe zones.

Daily Reminders – Device gives visual alerts at the right time, with friendly or personalized messages.

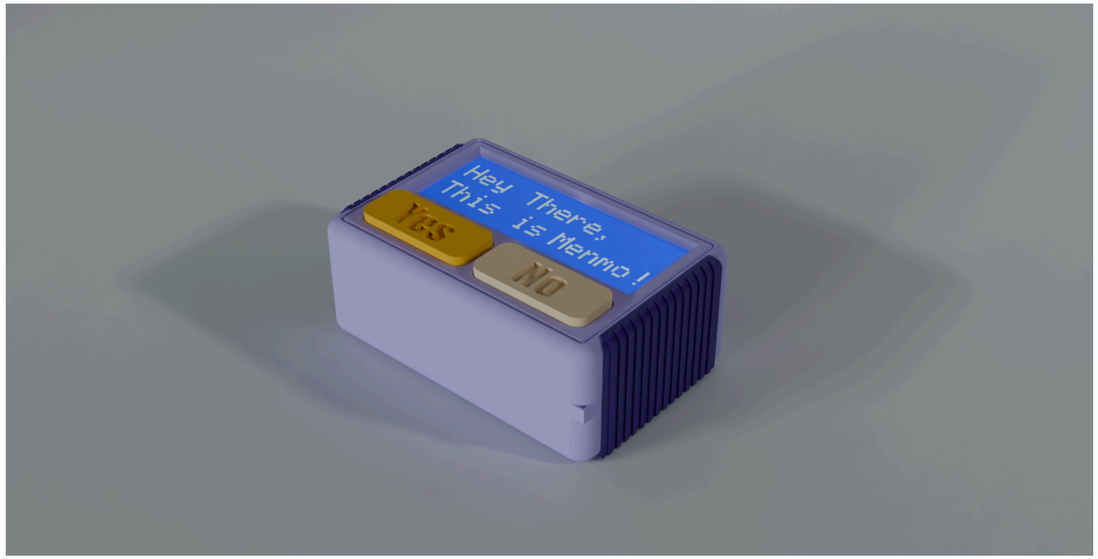
Location Tracking – Built-in GPS continuously shares patient's location with caregiver in real time.

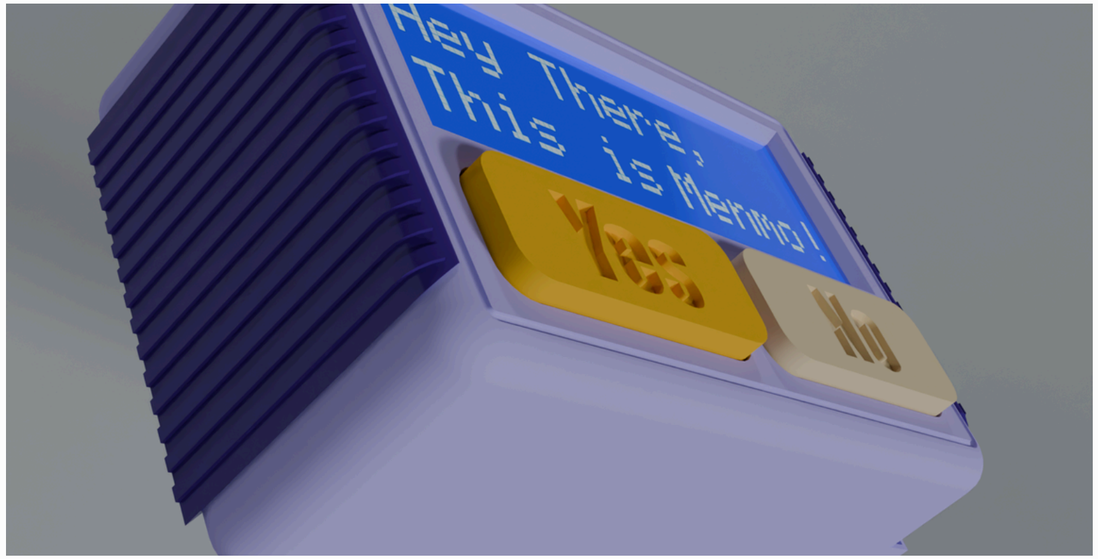
Safety Alerts – If the patient wanders outside a safe zone, caregiver instantly gets notified.

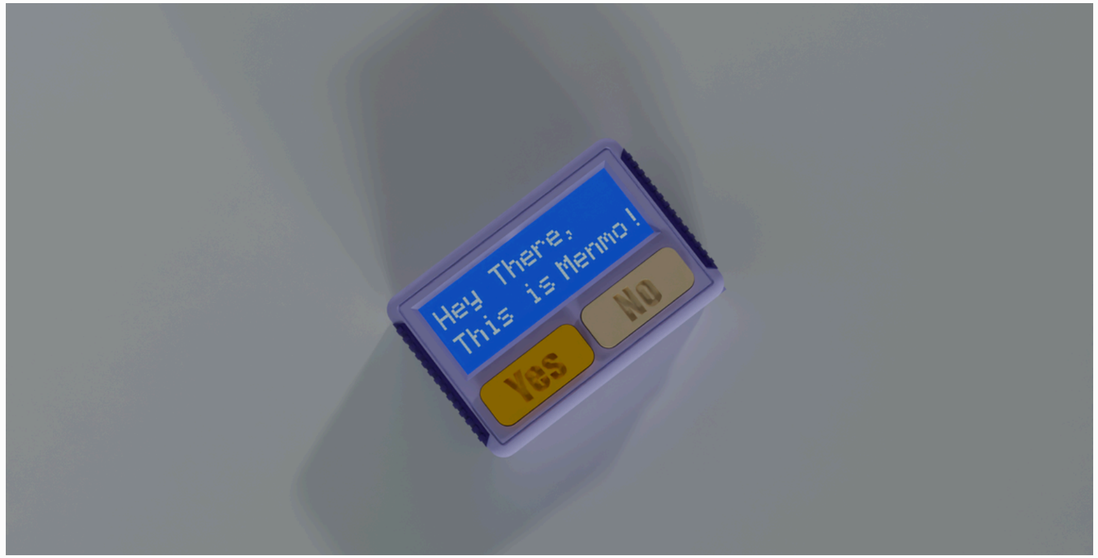
Emotional Support – In between reminders, the device sends motivational messages to encourage the patient.

Caregiver Dashboard – Caregiver can remotely monitor routines, receive alerts, and update reminders anytime.

Final Look:


Hey There,
This is Memmo!
Yes
No


Hey There,
This is Memmo!
Yes


Hey There,
This is Memmo!
No
Yes


Hey There,
This is Memmo!
Yes

KEY LEARNINGS:

- Basics of sensors, actuators, microcontrollers (like Arduino), wiring, and troubleshooting circuits.
- Writing simple code to control hardware, process inputs, and trigger outputs.
- Learning how hardware and software must communicate seamlessly for the device to function.
- Identifying user needs (like Alzheimer's reminders), translating them into technical requirements, and building a working prototype.
- Testing the device, identifying issues, and improving design like making it compact, ergonomic, or adding useful features.
- Working with peers to divide tasks, combine skills, and integrate different parts into one functional system.
- And a lot of patience...

THANK YOU