Personal Health CareTaker Bot Using IOT

\*Note: Sub-titles are not captured in Xplore and should not be used

line 1: 1st Given Name Surname   
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 4th Given Name Surname  
line 2: *dept. name of organization*  
*(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCIDAnshul Nagar  
Department of electronics and telecommunications engineering   
Shri Govindram Seksaria Institute of Technology and Science Indore, M.P. ,India  
Nagaranshul9@gmail.com

line 1: 5th Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCIDline 1: 3rd Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 6th Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

***Abstract*—** **This paper presents an Automatic Medicine Dispenser and Health monitoring system prototype. There are many challenges that adults face, including taking their medication on time. Older people often forget to take their medication on time and find it difficult to remember if they have had their medication, which can sometimes lead to overdose and serious health problems. There are a few expensive pharmaceutical providers available in the market right now. However, the majority of older people around the world do not even know about such products and still turn to prescription drugs. A few types of pharmaceutical suppliers are commercially available worldwide. However, they have a few problems that need to be resolved. These issues can be resolved using a reliable, affordable and affordable Automatic Medicine Dispenser that lasts up to 2-3 weeks, in such a way that adults do not need to depend on anyone else. The product is designed to ensure that the amount and time of the pills to be released can be controlled and monitored using the app, making things easier for everyone, including children working abroad. Also, it offers a clear bet between customer and parent statistics as it will promptly notify the caregiver in case a patient misses a pill.**

# INTRODUCTION

Studies show that most people, regardless of age, are forgetful when it comes to taking medication. This, which is more common in the elderly, leads to many other health risks. This is where pharmacists start working. There are different types of dispensers all recently used using strong drugs. Most of them are simple with no reminders etc. This paper proposes a device that can dispense solid drugs and can also monitor health, have an integrated application through which we can set alarms and prescription messages. One of the biggest problems in the healthcare industry is adherence to medication. It was revealed that, in a study conducted at the University of Washington, 30.7 percent of participants did not like it and 18.3 percent preferred at least one drug. Pills manufacturers have created special collections to see their item in those of their competitors and create product uncertainty. Bundling configurations often include “memory aid” to help women follow their daily pill routine, such as stylish cases so that the pills can be carefully distributed through packets and handbags. A pharmacist is a tool that in a timely manner dispenses the required medication or prescription. It has been shown to be effective in improving adherence to medication. This paper aims to design a non-invasive and inexpensive bot to tackle the challenges . In Personal health caretaker bot, when it is time to take the medicine, the device can be designed dose into a small room that can be easily opened.. The patient is usually notified by a loud alarm signal. When patient presses the medicine dispenser button the required medicine gets dispensed and the alarm stops.

Also , IoT has been widely used to coordinate advanced medical services and to provide intelligent and effective health care services to individuals. Improved nerves can be worn or embedded in patients' bodies, to keep their health in check. Information collected in this way, can be analyzed, compiled and excavated to identify diseases early. Diagnostic algorithms help physicians make treatment more personal and help to make health care safer, at the same time, more effective.

# TECHNOLOGY IN GENERAL

## *Arduino Embedded C*

The term “Arduino-compatible coding” refers to all Arduino and Arduino-compatible microcontroller boards that can be programmed and uploaded using Arduino IDE. Arduino boards are programmed in “C.” C is a popular system programming language that has minimal execution time on hardware in comparison to other high-level programming languages. Much like other microcontrollers, the AVR microcontrollers housed in Arduino boards are programmed in a subset of C. A general term for such subsets is “Embedded C” because they apply to programming embedded controllers. The language in which Arduino is programmed is a subset of C and it includes only those features of standard C that are supported by the Arduino IDE.

## *Firebase*

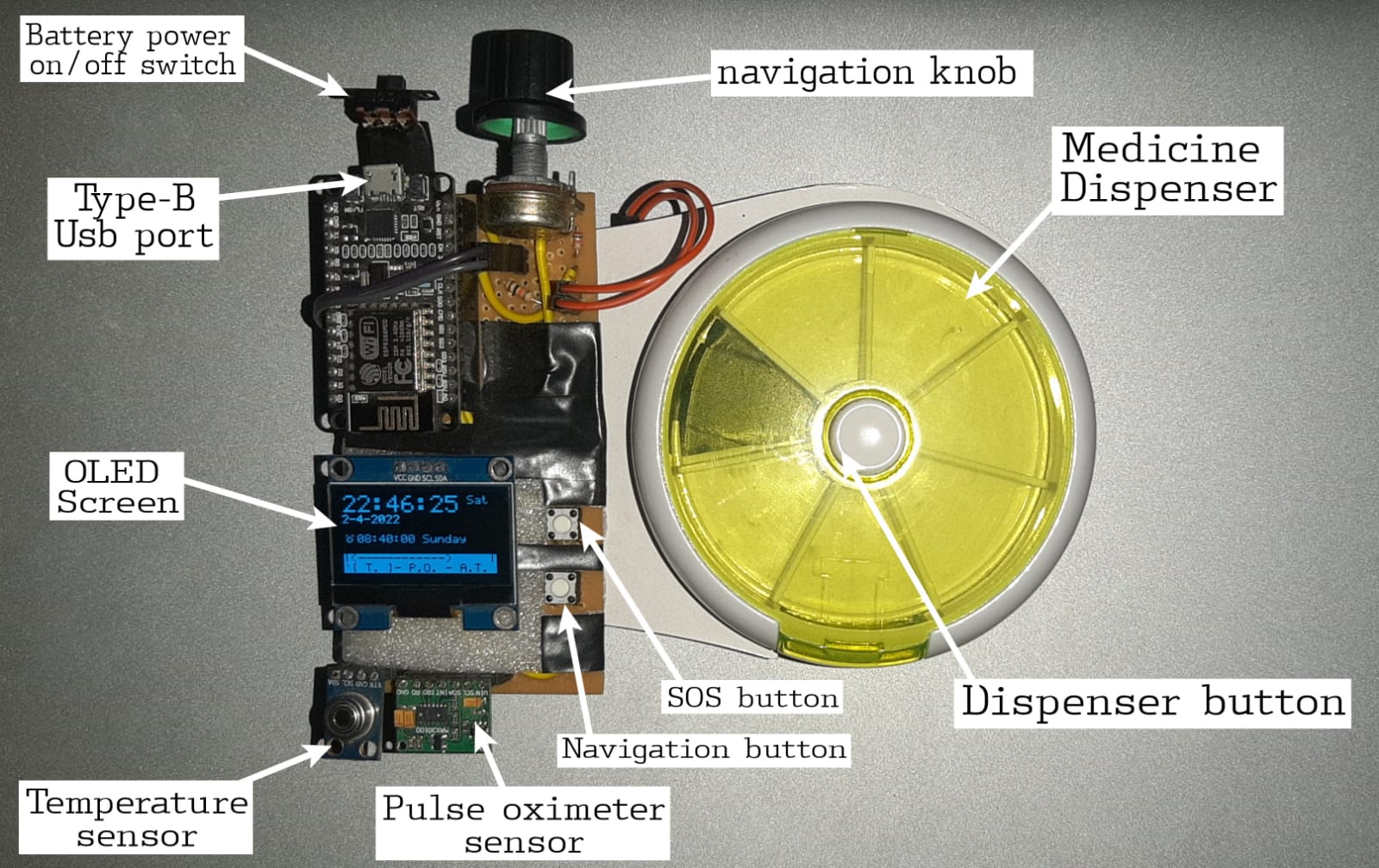
It is nothing but a set of tools used to create, improve and transform your app. It covers most of those processes and services that programmers often create themselves, but do not want to, because they often like to focus on the application. The cloud handles processes and services such as website, authentication, etc. The client SDKs provided by Firebase interact directly with these background services, without the need to establish any middleware between the service and your application.

## *Python Flask*

Flask is a small web framework written in Python. They are classified as microframework because they do not require special tools or libraries. It does not have a base for website summaries, form verification, or any other sections where existing third-party libraries provide similar services.

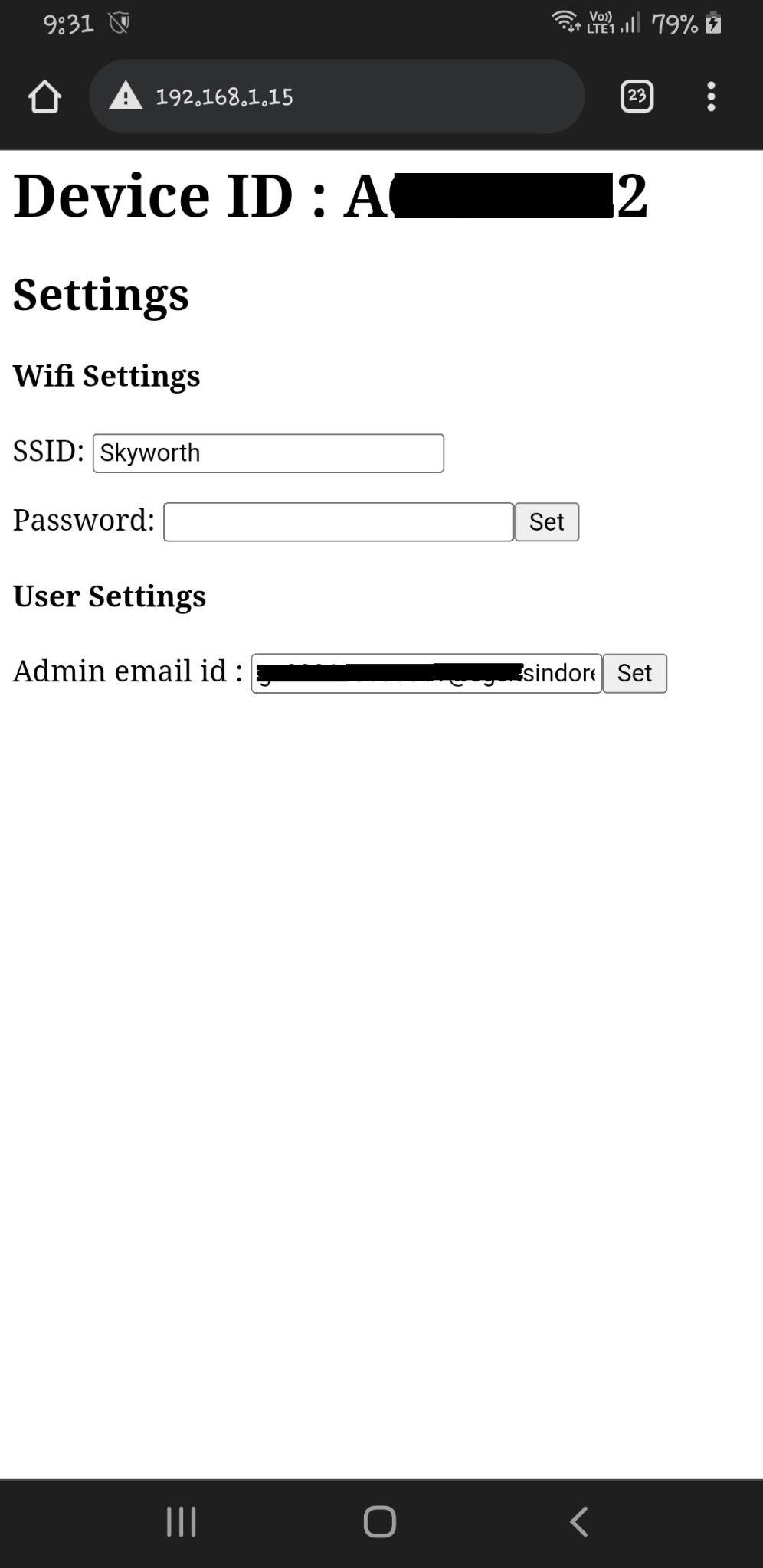
# PROJECT WORKFLOW

## *Hardware Level*



*Figure 1 Personal health caretaker bot hardware*

The personal-health caretaker bot consist of a switch to turn it on or off a usb-port for power supply and internal battery charging , an O-led display and two buttons and one knob for interaction with bot, from two given buttons one navigation button is used to select options and skipping medicine time and the other button is reserve only for SOS messages only. The knob is used to navigate between various options. It also contains a medicine dispensing assembly with an inbuilt button to dispense medicine, when this button is pressed the signal is send to bot and it updates it on cloud during the medicine time. Besides this it also contains temperature and pulse oximetry sensors for monitoring health.



*Figure 2 Accessing device settings from web browser by entering device local IP over LAN*

When the bot is turned on first it shows a greeting message just after that it tries to connect to a wifi network which is save in its eeprom memory or to a network whose SSID is “reset” and password is “reset@123” . Once it gets connected any one of the network it displays a local ip address of the device where you can access its settings and can set the wifi connection to which you want to connect and the admin account email for the device to get access of device from web app. It also displays next medicine alarm time after that and then the admin user for the device. After that we can see the home screen of the device in oled it shows the current time in 24 hours format day and date then in next to alarm clock icon it shows the next medicine alarm time with day and in below is the navigation menu.



*Figure 3 Home Screen at three different instances of time with different options on select*

As shown in figure the homescreen menu contains 5 different options they are :-

* T. – for taking temperature readings
* P.O. – for taking SPO2 and heart rate readings
* A.T. – for seeing all medicine alarm times
* M. – for seeing last note or message
* R. – for seeing last temperature and pulse oximetry readings.

The option inside the square brackets is the current option selected and on pressing the navigation button will enter into that option.

When T. option is opened it immediatly starts taking the reading and once it gets stable readings or after few seconds the final reading has been taken then it asks to upload the readings to cloud you can simply use knob to set it to yes or no the selected option font size is bigger after few seconds the action has been taken according to the option selected.

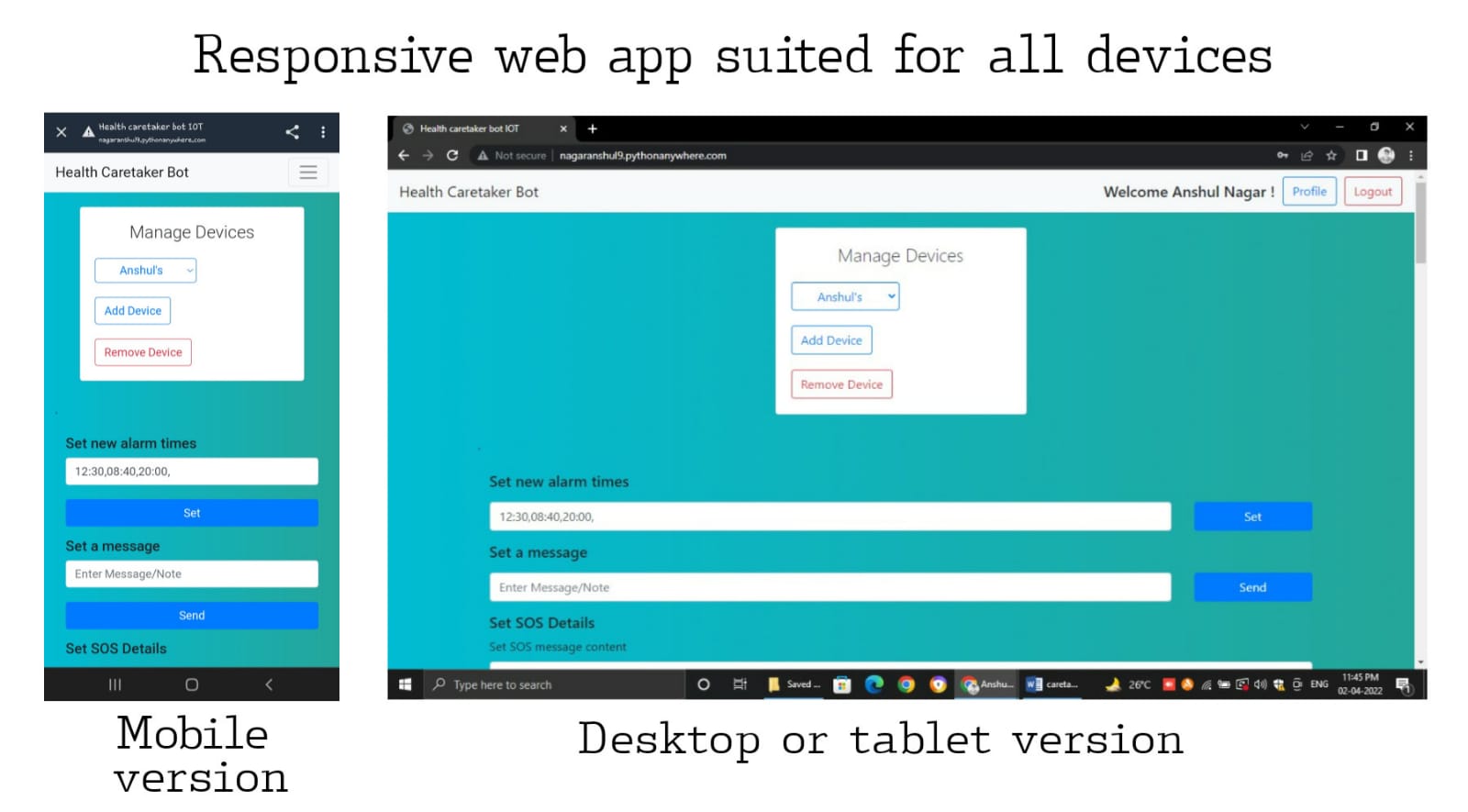
When P.O. option is opened the pulse oximetry sensor gets activated u have to place your finger on the sensor for 10 seconds to get a stable reading once the reading is recorded you can send it to cloud if you want just like with T. option.

On opening A.T.  M. or R. option the result will display on screen for few seconds then returned to the home screen.

## *Web Application :*

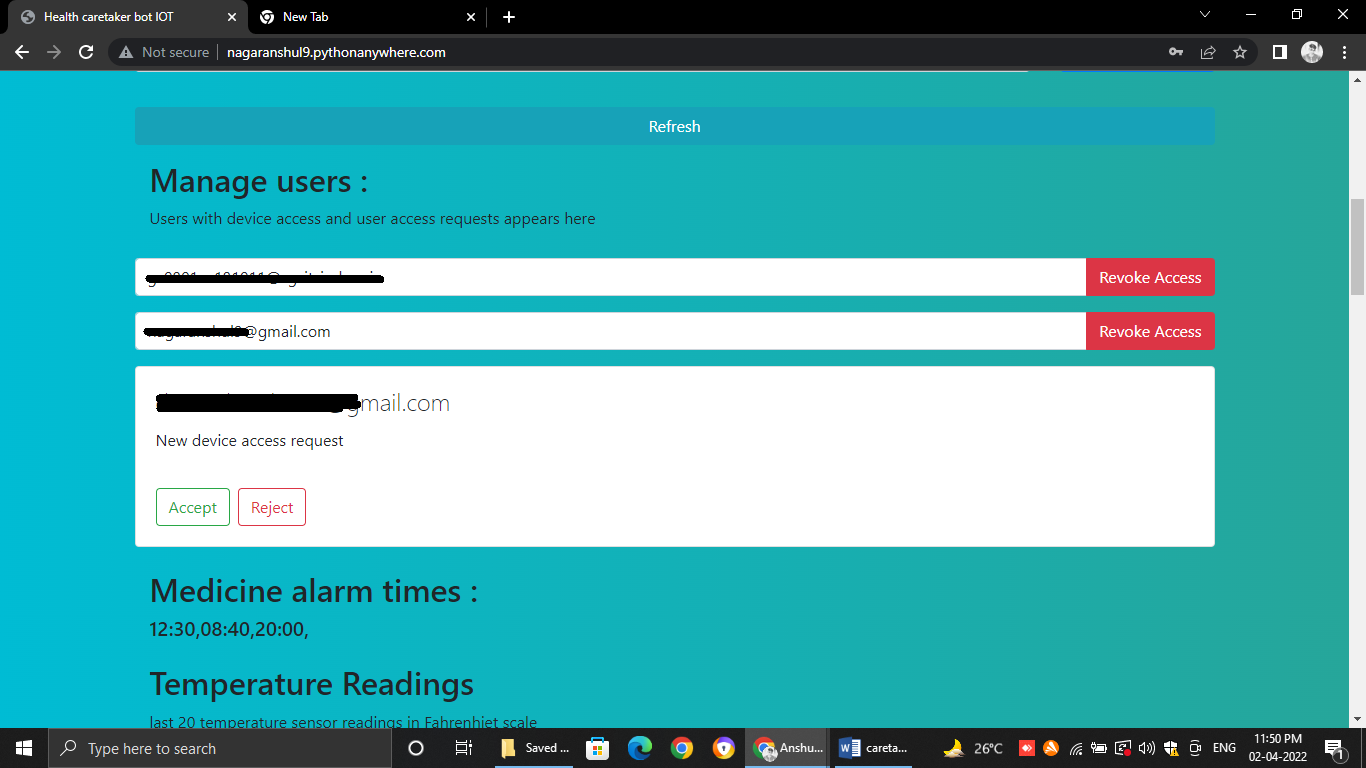
Link to Web app

<http://nagaranshul9.pythonanywhere.com/>



*Figure 4 Personal Health Caretaker bot web app*

To handle access manage and operate the personal health caretaker bots effectively the personal health-caretaker project has its own web based application which can be easily accessible on all devices. It is easy to manage bots in secure way. In order to get started with web app you need to first sign up to the web app and once your account has been created you have to verify your email id in order to get access to the application. Once you have verified your email you can manage your multiple devices linked with your id, in order to access a new device first you need to add your device by filling the details in add devices option, it asks for a device id which is a  unique Alphanumeric ID for identifying device ( you can find your device id by accessing settings as shown in figure 2 ), once you filled all the details click on add device to send a request to the user who has the admin access for the device. To get the admin access set your registered and verified email id in admin email id settings of your device. If the admin accepts the request you can access the device also only the admin of the device can see who have the access to the device and can revoke the access at any time. Once you can access a device you can set alarm times send messages to device, set SOS message and SOS message receivers , can see the medicine taken status also temperature,SPO2 and heart-rate readings . you can also link your telegram id to the web app in order to access device using telegram bot service.



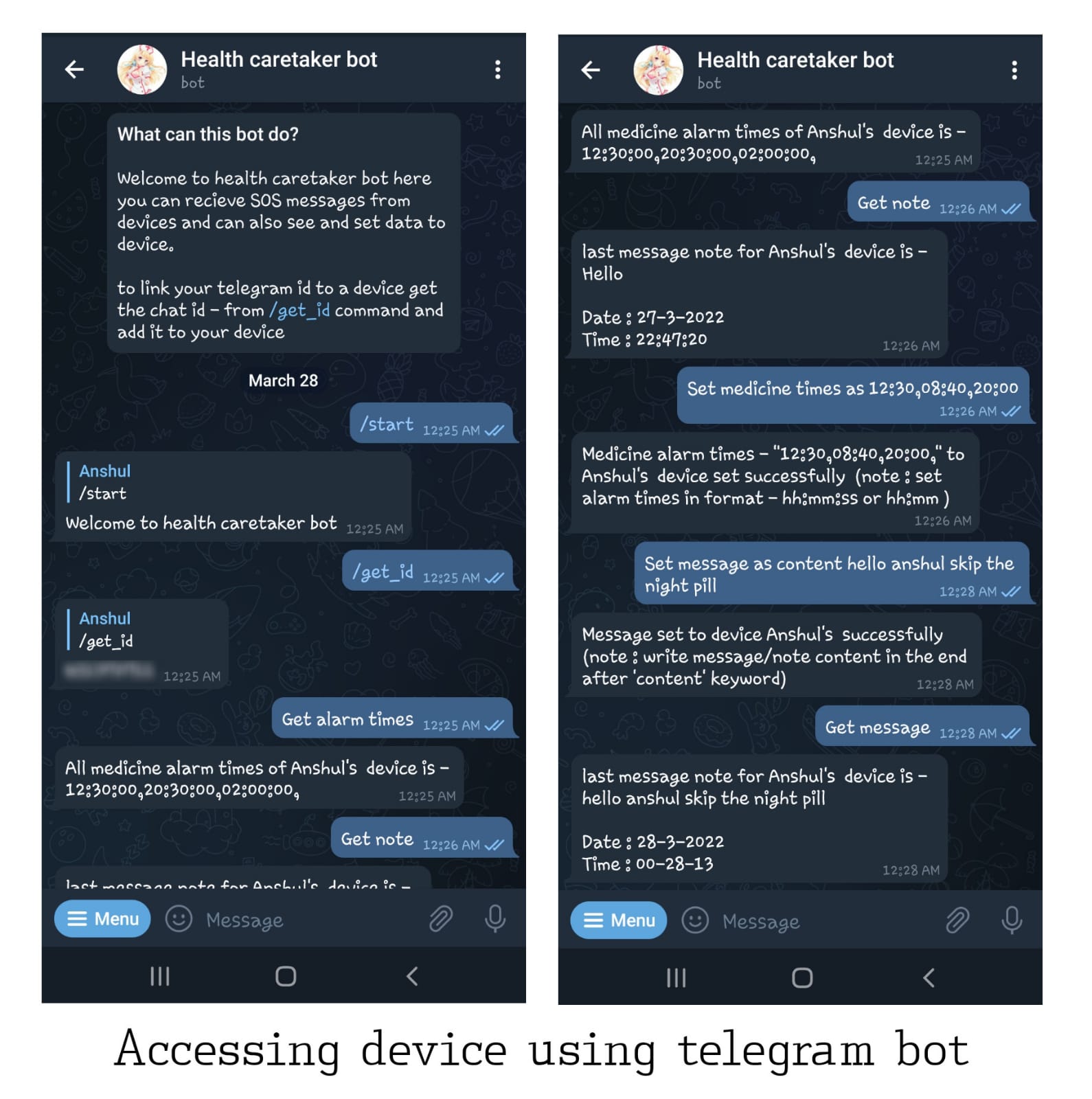
*Figure 5 Access to Manage Users tab only to device admin id to manage device access to its all users*

In order to access your devices using telegram you need to add your telegram chat id with telegram health caretaker bot in the profile settings Link telegram id

Scan the below QR Code to add Health caretaker bot to your telegram. once you added health care taker bot get the chat id by selecting Get telegram chat id in menu option then fill this chat id in web app link telegram id option and now you can access your devices using telegram only. You can also receive SOS messages from devices here.



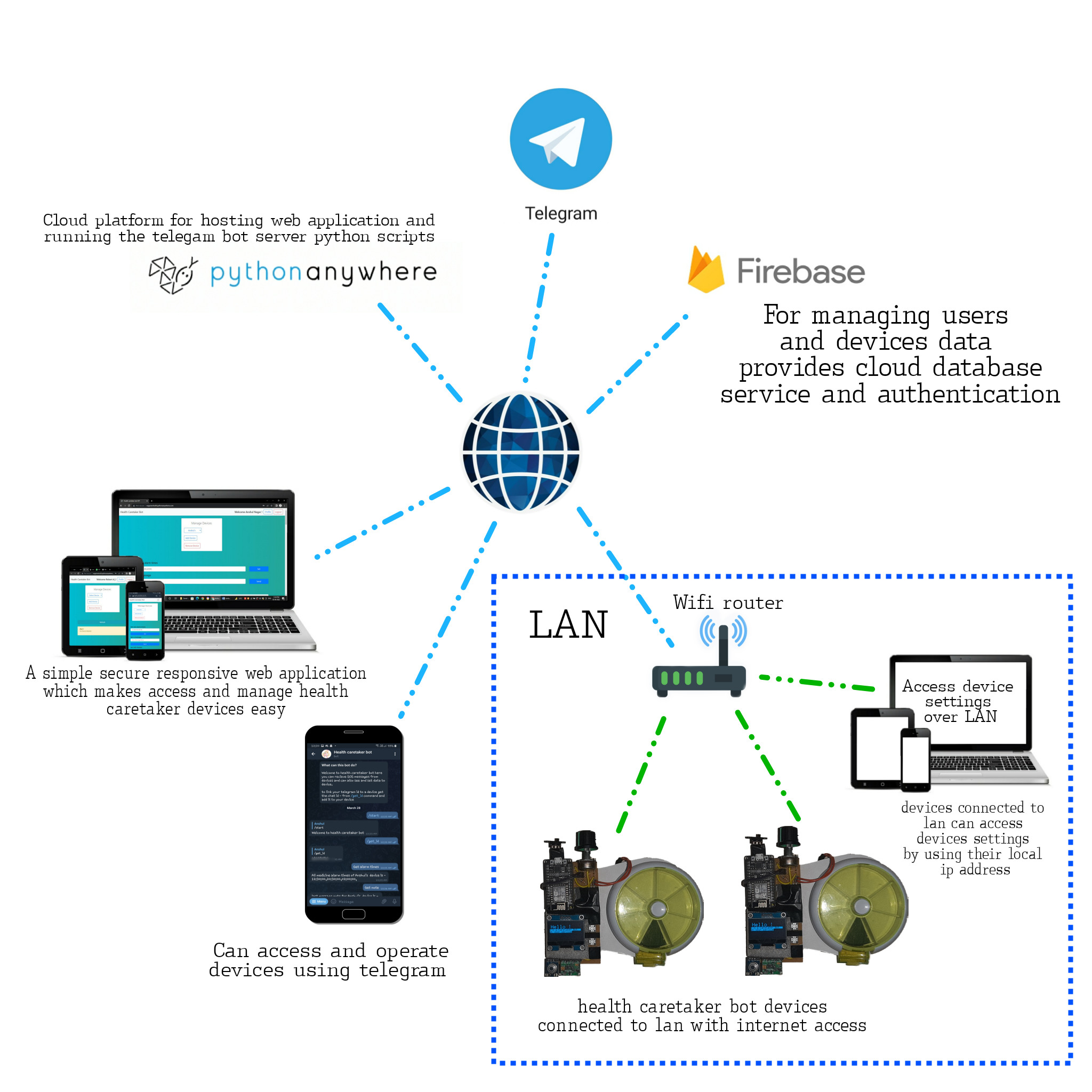
*Figure 6 Scan to add bot to your telegram*

*Figure 7 Using telegram bot to access devices*

# PROJECT STRUCTURE

## *Overview :*

The development of personal health caretaker bot project requires integration of various technologies and cloud based services. The hardware itself is integrated with firebase for storing data on cloud realtime-database and for retrieving data . All the data gathered by device is available on firebase database which can be accessed of modified using web application and telegram bot api hosted using python on python anywhere free cloud service.



*Figure 8 Overview architecture of personal health caretaker bot project*

## Hardware :

### Components Required :

#### I) NodeMCU :

The term Node MCU is a combination of the two words “node” and “MCU” which is a short term Microcontroller unit. Firmware and prototyping design suites are both open source platforms. The Node MCU is the most expensive IoT platform platform. The hardware is based on the ESP 12 module. It is a single board microcontroller with an active XTOS system. The CPU microcontroller is ESP8266. Lua Typing is used by firmware. Node MCU supports 32-bit ESP module. The Node MCU has an integrated WiFi board, ESP8266, so the Node MCU gets its widespread use in IoT applications. Fig.9 shows the NodeMCU image with the integrated ESP8266.

Features of ESP8266 :

* Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
* Operating Voltage: 3.3V
* Input Voltage: 7-12V
* Digital I/O Pins (DIO): 16
* Analog Input Pins (ADC): 1
* UARTs: 1
* SPIs: 1
* I2Cs: 1
* Flash Memory: 4 MB
* SRAM: 64 KB
* Clock Speed: 80 MHz
* USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
* PCB Antenna
* Small Sized module to fit smartly inside your IoT projects



*Figure 9.1 NodeMCU with integrated ESP8266*

**

*Figure 9.2 NodeMCU Pinout Diagram*

#### II) MAX30100 :

The MAX30100 is an integrated pulse oximetry and heart rate monitor sensor solution. It combines two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals. The MAX30100 operates from 1.8V and 3.3V power supplies and can be powered down through software with negligible standby current, permitting the power supply to remain connected at all times.

Features and Benefits of MAX300 :

* Complete Pulse Oximeter and Heart-Rate Sensor Solution Simplifies Design
* Integrated LEDs, Photo Sensor, and High-Performance Analog Front -End
* Tiny 5.6mm x 2.8mm x 1.2mm 14-Pin Optically Enhanced System-in-Package
* Ultra-Low-Power Operation Increases Battery Life for Wearable Devices
* Programmable Sample Rate and LED Current for Power Savings
* Ultra-Low Shutdown Current (0.7µA, typ)
* Advanced Functionality Improves Measurement Performance
* High SNR Provides Robust Motion Artifact Resilience
* Integrated Ambient Light Cancellation
* High Sample Rate Capability
* Fast Data Output Capability



*Figure 10 MAX30100 Pulse Oximeter sensor*

#### III) OLED-Display (SH1106 driver) :

SH1106 is a single-chip CMOS OLED/PLED driver with controller for organic/polymer light emitting diode dot matrix graphic display system.SH1106 consist of 132 segments,64 commons that can support a maximum display resolution of 132x64, it is designed for common cathode type oled panel. It embeds with contrast control display RAM oscillator and efficient DC-DC converter, which reduces the number of external components and power consumption. below is some of its features listed

* 132x64 dot matrix panel
* Embedded 132x64 bits SRAM
* Operating voltage

-Logic voltage supply Vdd1 = 1.65V - 3.5V

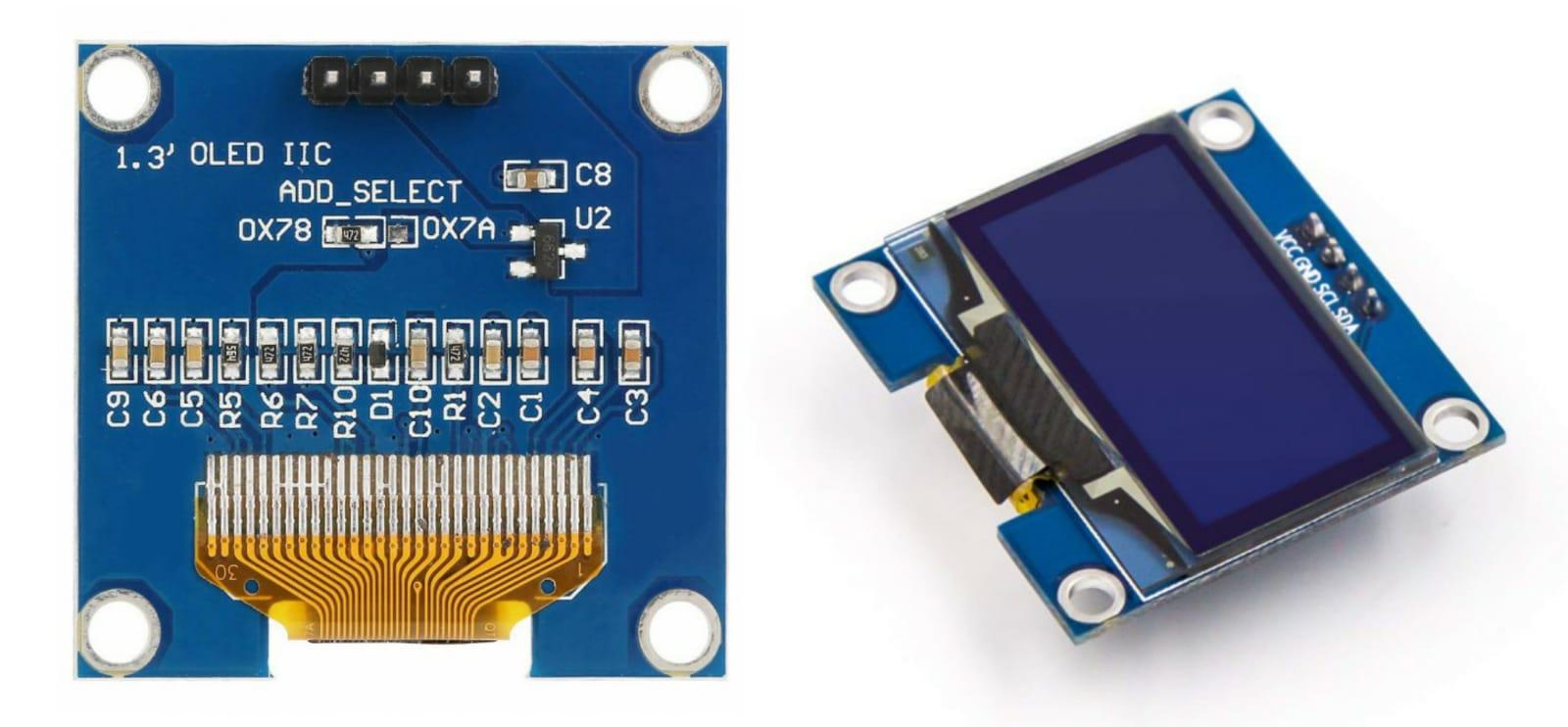
-DC-DC voltage supply Vdd2 = 3.0V - 4.2V

-OLED Operating voltage supply:

External Vpp supply = 6.4V - 14V

Internal Vpp generator = 6.4V - 9V

* Maximum segment output current: 200µA
* Maximum common sink current: 27mA
* 8-bit 8080-series parallel interface, 3-wire & 4-wire serial peripheral interface, 400KHz fast I2C bus interfaceI
* Wide range of operating temperatures: -40 to +85°C
* Programmable frame frequency and multiplexing ratio
* 256-step contrast control on monochrome passive OLED panel
* Programmable Internal charge pump circuit output
* Vertical Scrolling
* On-chip Oscillator
* Low power consumption , Sleep mode: <5µAI , VDD1=0V VDD2=3.0V – 4.2V: <5µAI ,VDD1,2=0V, VPP=6.4V–14.0V: <5µA1
* Available in COG form, thickness: 300umI



*Figure 11 SH1106 I2C Dotmatrix OLED Display*

#### 

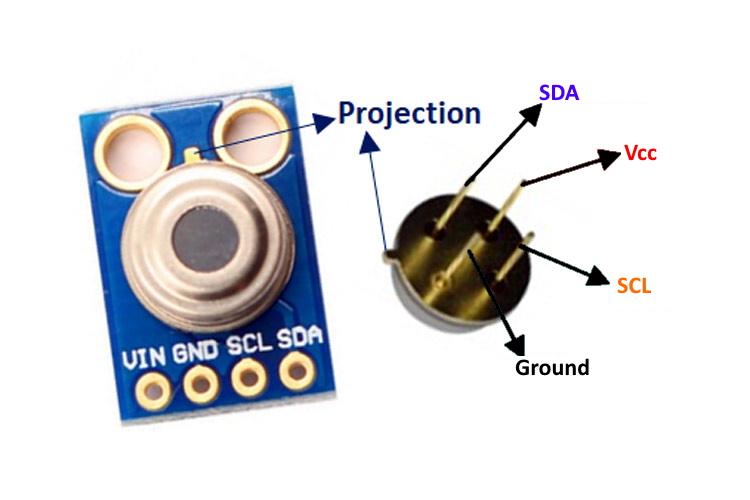
#### 

#### IV) MLX90614

The MLX90614 is an Infra Red thermometer for non contact temperature measurements. Both the IR sensitive thermopile detector chip and the signal conditioning ASSP are integrated in the same TO-39 can.

Features and Benefits of MLX90614

* Small size, low cost
* Easy to integrate
* Factory calibrated in wide temperature range:
* -40 to 125 ˚C for sensor temperature and
* -70 to 380 ˚C for object temperature.
* High accuracy of 0.5°C over wide temperature
* range (0..+50°C for both Ta and To)
* High (medical) accuracy calibration optional
* Measurement resolution of 0.02°C
* Single and dual zone versions
* SMBus compatible digital interface
* Customizable PWM output for continuous
* reading
* Available in 3V and 5V versions
* Simple adaptation for 8 to 16V applications
* Power saving mode
* Different package options for applications and
* measurements versatility
* Automotive grade



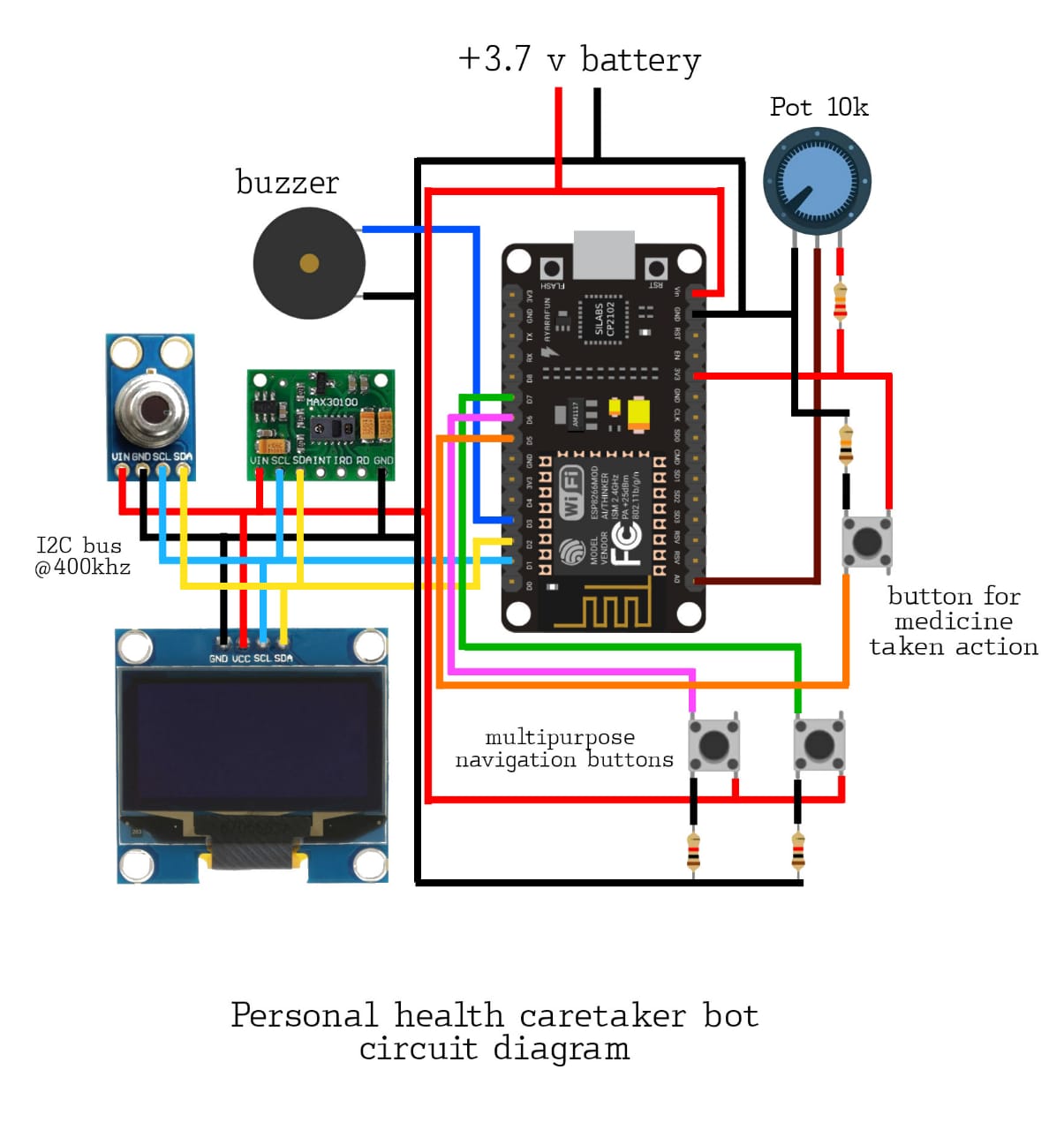
*Figure 12 MLX90614 Infrared Temperature sensor*

#### V) All Components list

TABLE 1 Components required

| **No** | **Component** | **Description** |
| --- | --- | --- |
| 1 | NodeMCU-ESP8266 | Microcontroller unit with inbuilt wifi ( IOT development board ) |
| 2 | MAX30100 | Pulse-Oximetry sensor for SPO2 and heart rate readings |
| 3 | OLED-Display (SH1106 driver) | For display |
| 4 | MLX90614 | Contactless IR temperature sensor |
| 5 | Knob Potentiometer | 10k value , used for navigation purpose |
| 6 | buzzer | For alarm purpose |
| 7 | Push buttons (x2) | For navigation and operation purpose |
| 8 | Safety press button | In medicine dispensing button |
| 9 | Medicine dispenser assembly | For dispensing medicine |
| 10 | 3.7v rechargeable battery | For powering the device |
| 11 | Resistors | 1k,2k2,100E,100E |
| 12 | PCB (Custom made or Zero) and Soldering cost and other miscellaneous small items | Variable |

### Circuit Diagram :



*Figure 13 : Circuit diagram of personal health caretaker bot hardware*

### Connections:

* GPIO 5 (D1 nodeMCU) -> SCL (I2C bus)
* GPIO 4 (D2 nodeMCU) -> SDA (I2C bus)
* GPIO 0 (D3 nodeMCU) -> Buzzer(+)
* GPIO 14 (D5 nodeMCU) -> press button signal medicine dispenser
* GPIO 12 (D6 nodeMCU) -> SOS button
* GPIO 13 (D7 nodeMCU) -> Menu select button
* ADC 0 (A0 nodeMCU) -> Potentiometer key
* VIN or VCC of I2C devices -> VIN of NodeMCU

### Embedded C coding using Arduino IDE

#### I) Main operations/functionalities to be performed by the code :

* + Connecting to wifi
  + Get current time using NTP client
  + Getting data from sensors using I2C comm.
  + Interfacing i2c oled display and buzzer for output
  + Interfacing buttons and potentiometer for navigation and operation of device
  + Designing menu for various options
  + Using EEPROM to store data like Wifi configurations and user email ID permanently
  + Setting up local esp8266 web server for setting configurations for wifi and firebase
  + Accessing firebase functionality like Authentication and RTDB(Real Time Database)
  + Sending Device and sensors data to RTDB
  + Getting realtime updates from database using streams
  + Processing the data and show output according to data received
  + Managing and Updating the Alarm times and output through buzzer and oled

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

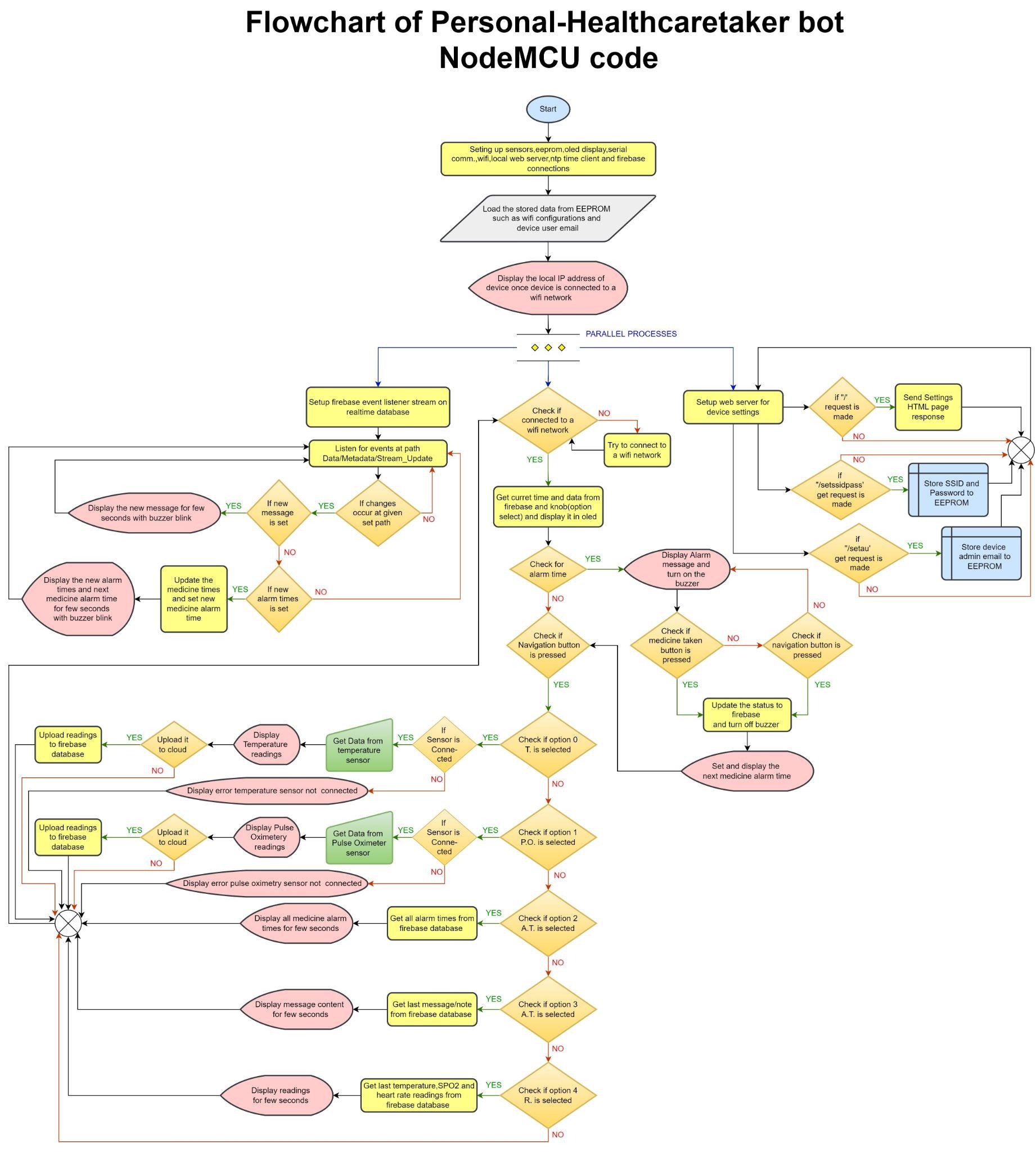
#### 

#### 

#### 

#### 

#### II) Flowchart :



#### 

#### 

*Figure 13 : Flowchart of ESP8266 code of personal health caretaker bot hardware*

#### 

#### 

#### III) Libraries Required and Modifications :

TABLE 2 Libraries required

| **Library** | **Version** |
| --- | --- |
| SPI | - |
| Wire | - |
| GyverOLED | 1.3.2 |
| ESP8266WiFiMulti | - |
| ESP8266mDNS | - |
| NTPClient (by fabrice Weinberg) | 3.2.0 |
| WiFiUdp | - |
| Firebase\_ESP\_Client (by Mobizt) | 2.5.5 (may cause some issues in newer versions) |
| ESP8266HTTPClient | - |
| DFRobot\_MLX90614 | 1.0.0 |
| MAX30100\_PulseOximeter (by OXullo Intersecans) | 1.2.1 |
| addons/TokenHelper | - |
| addons/RTDBHelper | - |
| ESP8266WiFi | - |
| ESP8266WebServer | - |
| EEPROM | - |

Modifications : Make sure that the I2C BUS SPEED for all the I2C devices is same for MAX30100 bus speed is set to 400000UL by default change I2C\_BUS\_SPEED from 400000UL to 100000UL in MAX30100.h file present in sensor’s library folder

#### IV) Code :

Link to documented esp8266 nodemcu code :

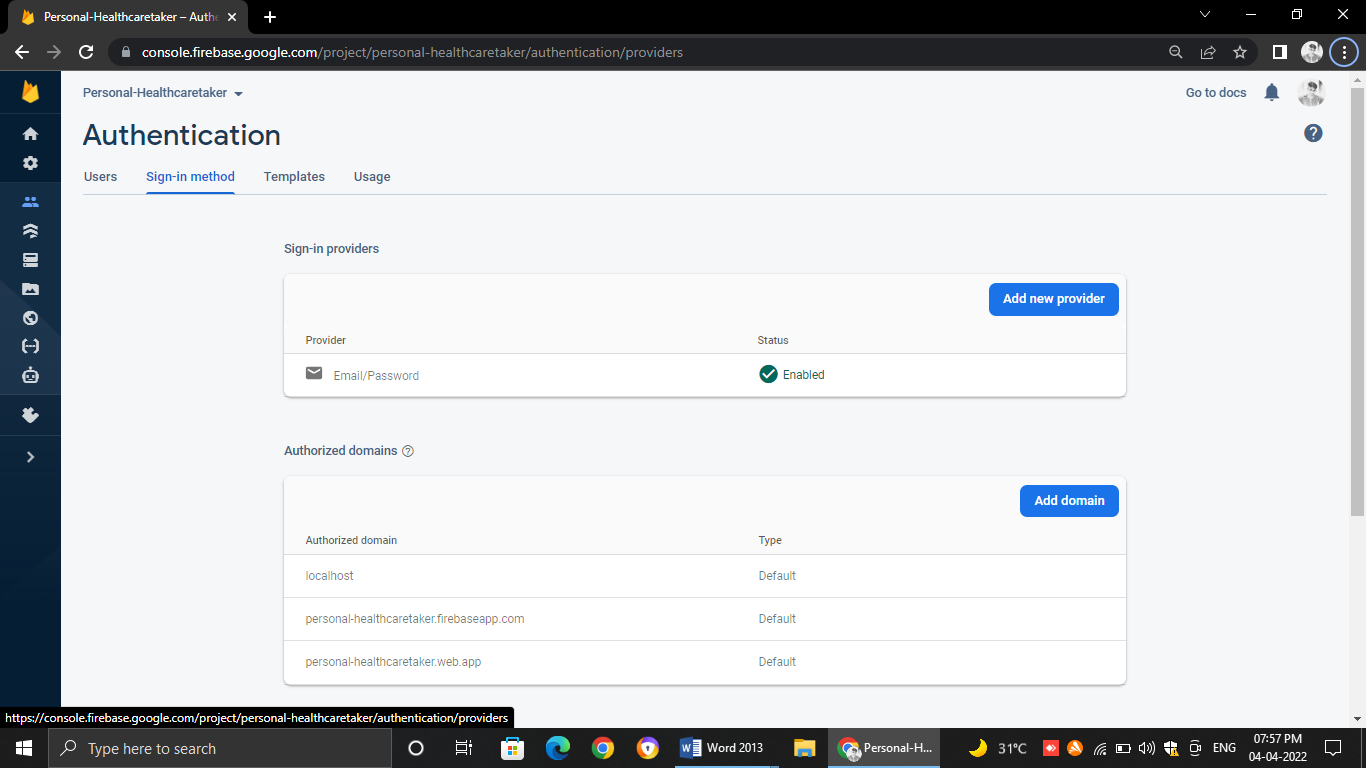
<https://drive.google.com/file/d/1AqbbM0pA_BHTgasPKTm9qreOUX5MfROS/view?usp=sharing>

## Firebase Setup

### Creating a new firebase project

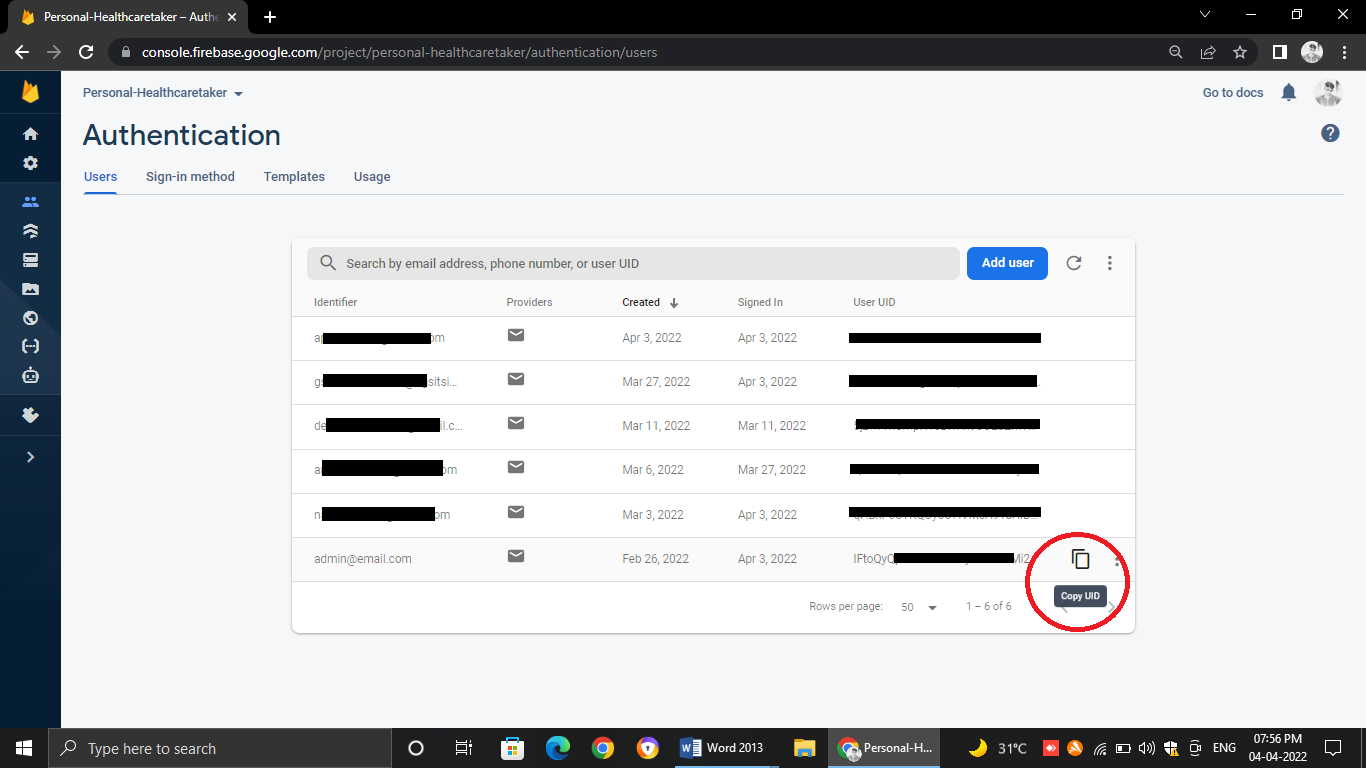
### Setting up the firebase Authentication

* Set up sign-in method add email/password as sign-in provider



*Figure 14 : Sign-in method tab after email/password provider is set*

* Create a admin access account by clicking add user then enter email and password you want
* Once the user is added you can see the User UID which will be needed to give access to database in security rules later

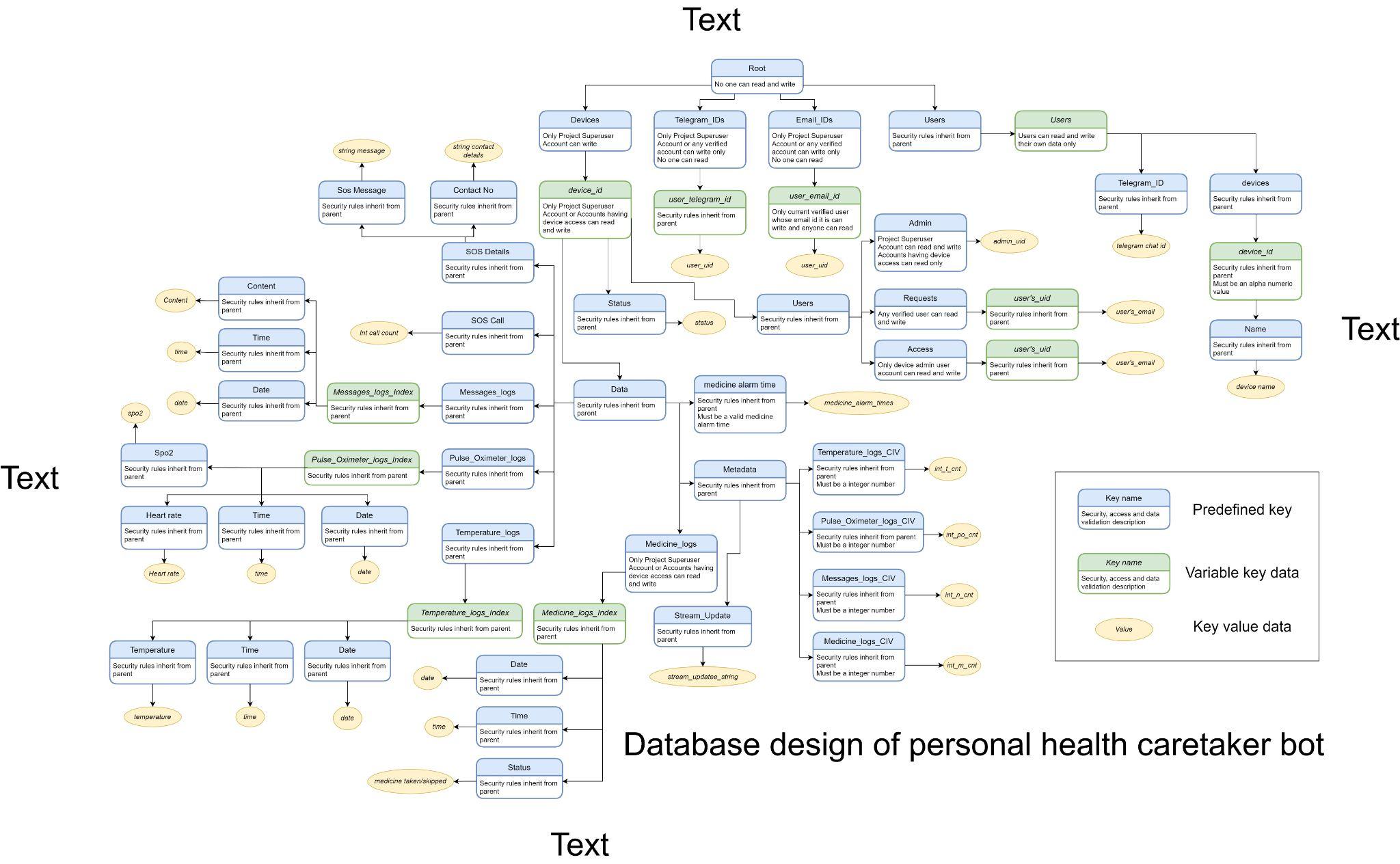
*Figure 15 Users tab where all users details can be seen copy the user uid from icon shown in red circle*

### Setting up the firebase real time database

Firebase realtime database is a no-sql non-relational database which means does not stores data in form of tables it uses a key value pair type structure or similar to JSON structure to store data.

It is free to use to some limit on amount of data downloaded and the number of queries processed in a particular time interval so we have to keep this in mind while using firebase realtime database in our project. It also provides security rules functionality to secure the data and give only limited and necessary access of our database to the users.

*Figure 16 Database Design of personal health caretaker bot*



## Web App development

For the development of Responsive web app the following technologies is been used :

1. HTML
2. CSS
3. Javascript
4. Bootstrap
5. Firebase

Below is the html web app code link

## Telegram chat bot development

Developement of telegram chat bot is done using python programming language in backend.but first we need to create a new bot in telegram using BotFather bot Search for BotFather in telegram and send ‘/newbot’ command to create a new bot once bot is created it provides a HTTP API token for the bot this API token is used for automating queries from bot using programming languages such as python. below is the main tasks done in order to prepare the functioning bot

* Automate reply messages and queries from telegram users through python script
* Accessing the firebase real time database from python and perform operations through it
* Implementing the SOS functionality by using event listeners functionality of firebase RTDB
* Running the Bot script on cloud console using pythonanywhere.com for free of charge

Code link for backend telegram bot :

## Hosting

### Setting up a python flask app on local machine:

For local testing put the html web app file and py file in same directory then run the below python code

Code : -

from flask import Flask,render\_template

app = Flask(\_\_name\_\_)

@app.route("/")

def home():

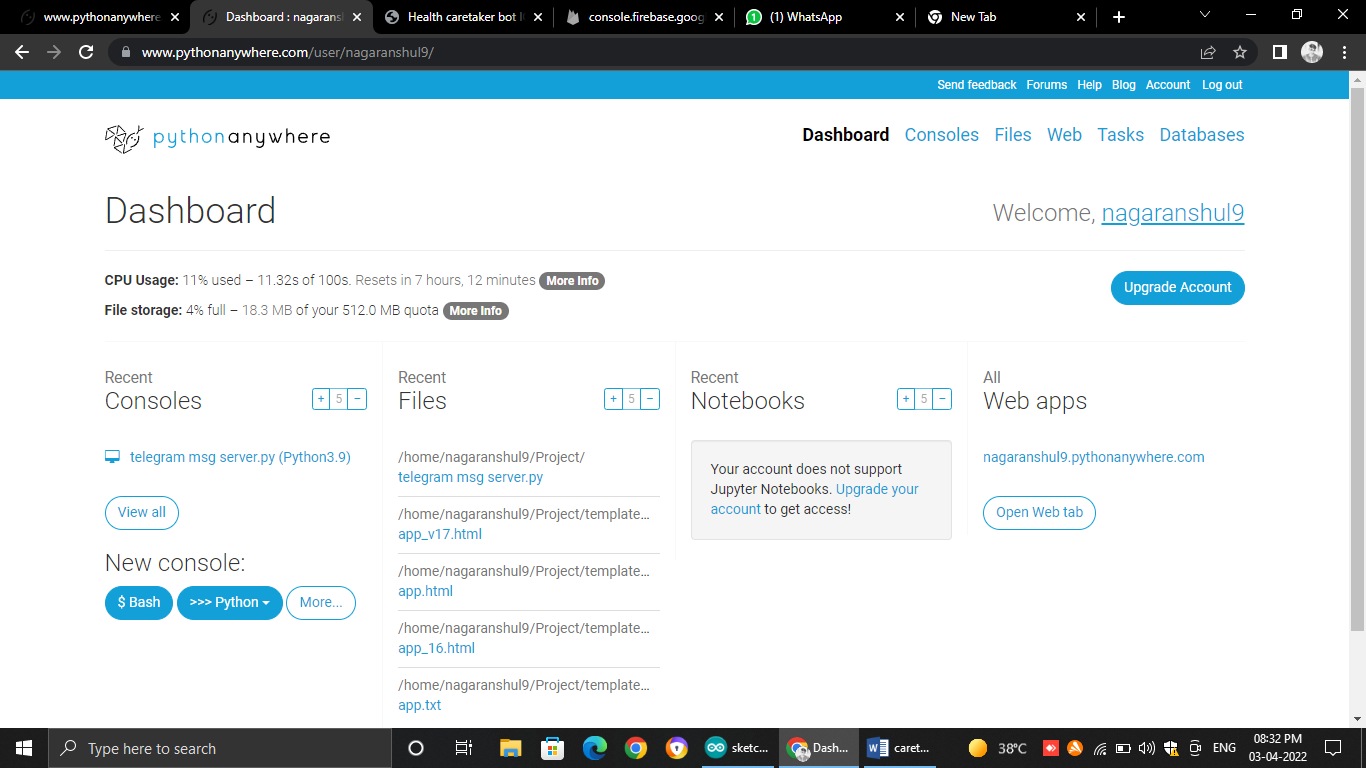
return render\_template("app.html")

# delete the below lines of code for deployment in pythonanywhere

if \_\_name\_\_ == "\_\_main\_\_" :

app.run(host="0.0.0.0",port=5500)

### Deploying the flask app on pythonanywhere.com



*Figure 17 Pythonanywhere.com dashboard*

First Create a Project folder inside which is python flask code file named ‘flask\_app.py’ is present and also a directory named templates in which the html file must be there then go to web tab and update the working,source code and static directory and then reload the webapp

# ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R.

B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

# REFERENCES

[1] Wissam Antoun, Ali Abdo, Suleiman Al-Yaman, Abdallah Kassem, Mustapha Hamad and Chady El-Moucary, “Smart Medicine Dispenser (SMD)”, 2018 IEEE 4th Middle East Conference on Biomedical Engineering (MECBME), pp. 20-23, 2018.

[2] A. V. Dhukaram and C. Baber, "Elderly Cardiac Patients' Medication Management: Patient Day-to-Day Needs and Review of Medication Management System," 2013 IEEE International Conference on Healthcare Informatics, pp. 107-114, 2013.

[3] World Health Organiztion, <http://www.who.int/medicines/en/> , accessed on Juy 21, 2017.

[4] S. L. Gray, J. E.Mahoney, and D. K.Blough, “Medication Adherence in Elderly Patients Receiving Home Health Services following Hospital Discharge.” Annals of Pharmacotherapy, 35(5), pp. 539-545, 2001.

[5] <https://iotdunia.com/iot-architecture/>

[6] [https://www.researchgate.net/figure/nternal-architecture-of-SQLite\_fig1\_2 71891972 [](https://www.researchgate.net/figure/nternal-architecture-of-SQLite_fig1_2)

[7] C. Fărcaú, I. Ciocan, N. PalaghiĠă and R. Fizeúan, "Weekly electronic pills dispenser with circular containers," 2015 IEEE 21st International Symposium for Design and Technology in Electronic Packaging (SIITME), pp. 125-129, 2015.

[8] N. B. Othman and O. P. Ek, "Pill dispenser with alarm via smart phone notification," 2016 IEEE 5th

[9] <https://datasheets.maximintegrated.com/en/ds/MAX30100.pdf>

[10] <https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet>

[11] <https://www.velleman.eu/downloads/29/infosheets/sh1106_datasheet.pdf>

[12] <https://datasheetspdf.com/pdf-file/774392/Melexis/MLX90614/1>

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published**