CSE1901 - Technical Answers to Real World Problems (TARP)

Project Report

Patient Healthcare System

By

19BAI1069 Araadhye Jain

19BAI1007 Piyush Parashrampuria

19BCE1640 Yashwant Reddy

19BCE1352 Anshuman Singh

19BCE1657 Tatvavid Tripathi

B. Tech Computer Science and Engineering

Submitted to

Dr Geetha S

School of Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

29 April 2022

DECLARATION

We hereby declare that the report titled **"Patient Healthcare System"** submitted by us to VIT Chennai is a record of bona-fide work undertaken by us under the supervision of **Dr. Geetha S**, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai.

Signature of the Candidates

Araadhya

Araadhye Jain

19BAI1069

Piyush Parashrampuria

19BAI1007

Yashwant Reddy

19BCE1640

Anshuman Singh

19BCE1352

Tatvavid Tripathi

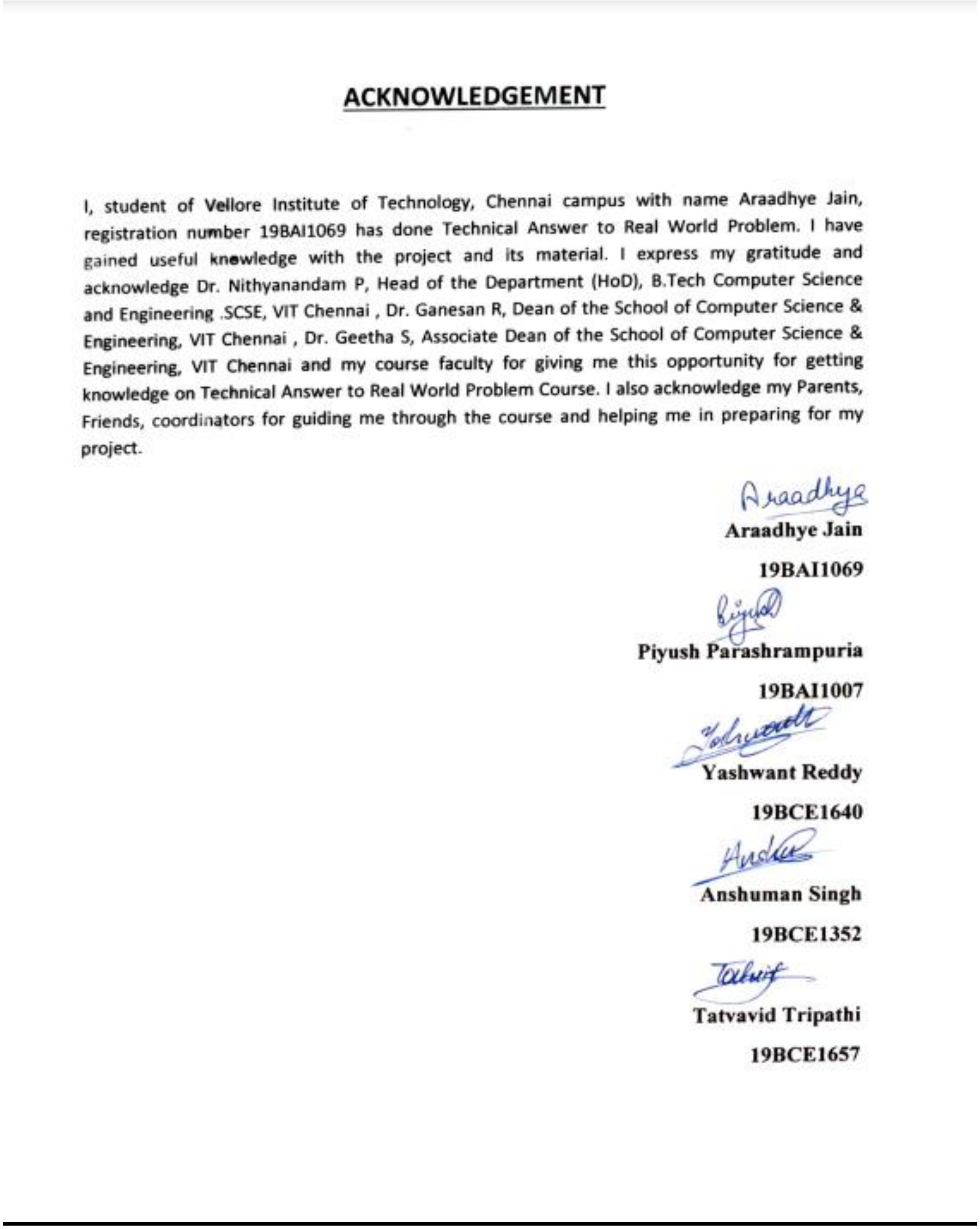
19BCE1657

CERTIFICATE

Certified that this project report entitled “**Patient HealthCare System”** is a bonafide work of **Piyush Parashrampuria(19BAI1007), Araadhye Jain(19BAI1069),Anshuman Singh(19BCE1352), Yashwanth Reddy (19BCE1640), Tatvavid Tripathi (19BCE1657)**and they carried out the Project work under my supervision and guidance for CSE1901 - Technical Answers to Real World Problems (TARP).

Dr. Geetha S

SCOPE, VIT Chennai



ACKNOWLEDGEMENT

I, student of Vellore Institute of Technology, Chennai campus with name Araadhye Jain, registration number 19BAI1069 has done Technical Answer to Real World Problem. I have gained useful knowledge with the project and its material. I express my gratitude and acknowledge Dr. Nithyanandam P, Head of the Department (HoD), B.Tech Computer Science and Engineering .SCSE, VIT Chennai , Dr. Ganesan R, Dean of the School of Computer Science & Engineering, VIT Chennai , Dr. Geetha S, Associate Dean of the School of Computer Science & Engineering, VIT Chennai and my course faculty for giving me this opportunity for getting knowledge on Technical Answer to Real World Problem Course. I also acknowledge my Parents, Friends, coordinators for guiding me through the course and helping me in preparing for my project.

Araadhye

Araadhye Jain

19BAI1069

Piyush Parashrampuria

19BAI1007

Yashwant Reddy

19BCE1640

Anshuman Singh

19BCE1352

Tatvavid Tripathi

19BCE1657

ABSTRACT

IoT has seen significant growth in the field of health-sector, but those IoT solutions are limited to hospitals or places where health facilities are provided. With the advent of new technology both in the field of IoT simulation and Application deployment, we can now create a system capable of assisting a regular patients or patient left in self-care in their daily day-to-day medication routines and providing them ways to interact with the room environment around them.

Such a system will help those patients who either forget their medication doses or are handicapped in controlling common room appliances such as TV, AC, room light, etc, also providing the means to track their body state such as oxygen level, and pulse rate, body temperature, etc. Apart from that, the system will be capable of identifying the critical conditions and providing an emergency solution for the same.

The proposed system will be capable of providing the user/patient an Application based control. The IoT system will be developed as hardware, with data analysis done on cloud and cross-platform applications.

CONTENTS

1. INTRODUCTION

1.1. Objective and goal of the project

The Patient Monitoring System will be characterized as a system used to monitor physiological signs including parameters such as an electrocardiogram (ECG), respiratory signs, invasive and non-invasive blood pressure, bodytemperature, and room temperature and humidity. Understanding and validating the monitoring system is an innovation at Mhealth. can be called mhealth or mobile health. This system is used for medical and general care using cell phones. These watch frames can be used locally or remotely. Monitoring of patient relates to various situations in which patient has a comorbid condition.

Patient monitoring is not another new framework in medicinal services as it was first begun in the year 1625 for checking the body temperature and pulse of patients. Subsequently, this framework has started to discover its utilization and acknowledgment for checking diverse sorts of physiological parameters and health related angles that are being performed as of not long ago.

1.2 Problem Statement

These days' patient monitoring frameworks are accessible in two structures: z Single parameter monitoring system: This system is utilized for measuring the blood pressure of a human body, observing ECG, checking SPO2 (oxygen level in the blood), etc. z Multi parameter monitoring system: This system is utilized for checking different crucial physiological indications of patients by transmitting the fundamental data like ECG, breath rate and blood pressure, and so on.

Because of these reasons, multiparameter observing system holds a huge part in the field of medicinal devices. These days, the health care sensors are playing a fundamental part in hospitals. The patient checking monitoring is one of the significant improvements as a result of its creative innovation. A programmed remote health observing system is utilized to quantify patient's body temperature, pulse by utilizing implanted innovation.

1.3 Related Works

The proposed system utilizes sensors like pulse sensor, oximeter, temperature sensor, accelerometer and gyroscope. These sensors mostly include in observing the health condition, fall detection and sleep pattern of the patient.

In today's social protection system for patients who remains in home amid post operational days checking is done either by means of administrator/medicinal guardian. Endless watching may not be expected by this system, in light of the fact that anything can change in prosperity parameter within some fraction of seconds and in the midst of that time if the specialist is not in the premises causes more important damage. So with this advancement made period where the web directs the world gives an idea to add to doctors from a group of specialized doctors present all over the globe where time to time consistent checking of the patient is refined.

Also, if the health detecting gadget is made to speak with a compact system like a tab or a cell phone which has the default capacity of speaking with Cloud (hospital or clinic database), then the entire system would be considerably more financially effective. This is on the grounds that these days a great many people have entry to versatile specialized devices and these devices have turned out to be very shabby.

The system can likewise be made IoT (Internet of Things) empowered and M2M (Machine To Machine) is good. This system, usage of such a healthcare checking system is displayed.

Thus, this will possibly profit an extensive population. For the healthcare checking system to be solid, every sensor should timely measure the information taking the recommended examining rate of the parameter, and the information should be sent to the data processor with no overlap. Every sensor has fluctuating necessities regarding information length or size and examining rate the sensor information gathered without overlap by information processor can replace notepad at patient's bed with smart gadget and patient's information can be accessed to from specialist's Smartphone

1.4 Existing Works

In a hospital, either the nurse or the doctor has to move physically from one person to another for health check, which may not be possible to monitor their conditions continuously. Thus, any critical situations cannot be found easily unless the nurse or doctor checks the person's health at that moment. This may be a strain for the doctors who have to take care of a lot number of people in the hospital. Also, when medical emergencies happen to the patient, they are often unconscious and unable to press an Emergency Alert Button.

One of the application protocols that are being used to transfer data is Hyper Text Transfer Protocol (HTTP) for general communication over Internet. However, when HTTP is used for IOT communication, the protocol overhead and the resulting performance degradation are a serious problem. Also, IP addressing depends on the physical location causing network management complexity issues.

2. Literature Survey

Some related work has already been done in terms of building an environmental management system. Here's what it describes:

IbrarYakuk et al. developed a state-of-the-art intelligent environment using IoT to assist and empower residents in tasks such as navigating unfamiliar spaces and moving heavy objects for the elderly.

R. Kumar et al. A framework for IoT-based patient monitoring system using Raspberry Pi was proposed. The proposed system measures the patient's body temperature and heart rate by using sensors such as a temperature sensor and a heart rate sensor.

Pradhan B. et al. Proposed medical devices to help integrate the benefits of IoT technology and cloud computing into the medical field. It also describes protocols for communicating patient data from multiple sensors and medical devices to a designated medical network.

During the ongoing COVID19 pandemic, Smart health monitoring systems are potentially immensely beneficial for COVID19 patients. This study presents that this system is a realtime health monitoring system utilizing the measured values of body temperature, pulse rate, and oxygen saturation of the patients, which are the most important measurements required for critical care. This system has a liquid crystal display (LCD) that shows the measured temperature, pulse rate, and oxygen saturation level and can be easily synchronized with a mobile application for instant access. The proposed IoTbased method uses an Arduino Unobased system, and it was tested and verified for five human test subjects. The results obtained from the system were promising: the data acquired from the system are stored very quickly. The results obtained with the system were found to be accurate when compared to other commercially available devices. IoT-based tools could be useful in saving lives during the COVID-19 pandemic.

3. Requirements Specification

3.1 Hardware Requirements

- NodeMcu

- Arduino Uno R3
- Jumper Wires
- Servo Motor
- Dht11 Sensor
- Body Temperature Sensor
- Heart rate Sensor
- OLED Display
- 16X2 LCD Display
- BreadBoard

3.2 Software Requirements

- Platform.io
- Postman
- Flutter(For app)
- Arduino IDE
- C++ Programming(For Arduino Coding)
- HTML/CSS(Designing the Page)

4. System Design

We have a flutter-based front end and java spring boot-based main server and a node MCU server for hardware.

Java Spring boot main server is made on microservices architecture. The first layer is the authentication service. It authorizes any request from the server to check if it is a valid user. When the user will log in to the system it will send to the java spring boot to authenticate. If yes then it will show a successful response and an access token will be given. The access token is stored by the front-end server and is used for a y future login.

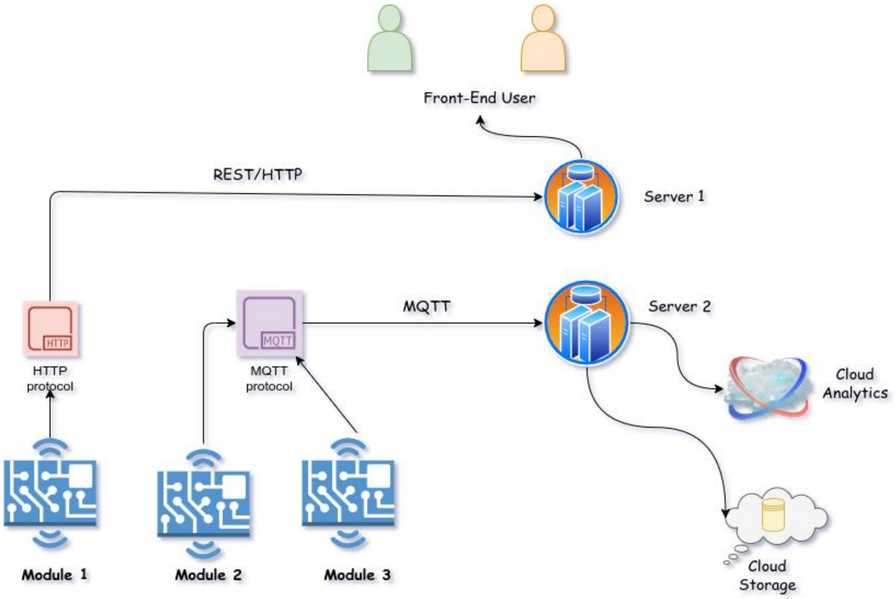
If it fails to pass the access token user can't use any kind of service in the application.Behind the Authentication service, we have two more services namely medicine dispatcher service and analytics. We have three-layer of neural networks 3 dense layers 3 hidden layers each of size 10 units.

In data analytics, we take a health dataset and predict the user's cholesterol, blood pressure, oximeter level, and body temp we predict the health condition and whether he is in good condition.

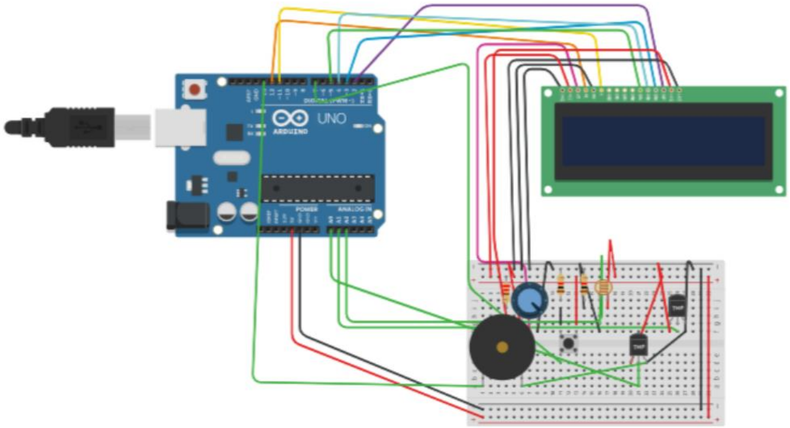
Now the Medicine reminder, If a user wants to set the medicine it will send the post request to the server. It will note the time in the scheduler. When the time of medicine approaches the main server sends a signal to the node MCU server to ring the reminder. Node MCU will accept the request and through 12C serial communication will send it to Arduino. The original message will display on LCD and the medicine dispatcher will move alternately.

5. System Design

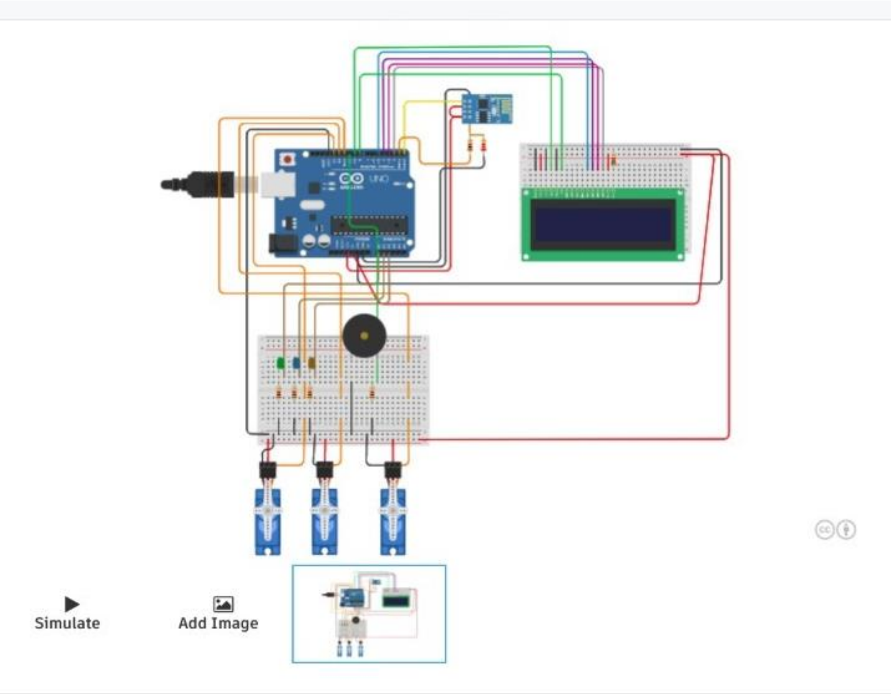
Flow Diagram/ system design of proposed work



PATIENT HEALTHCARE MODULE

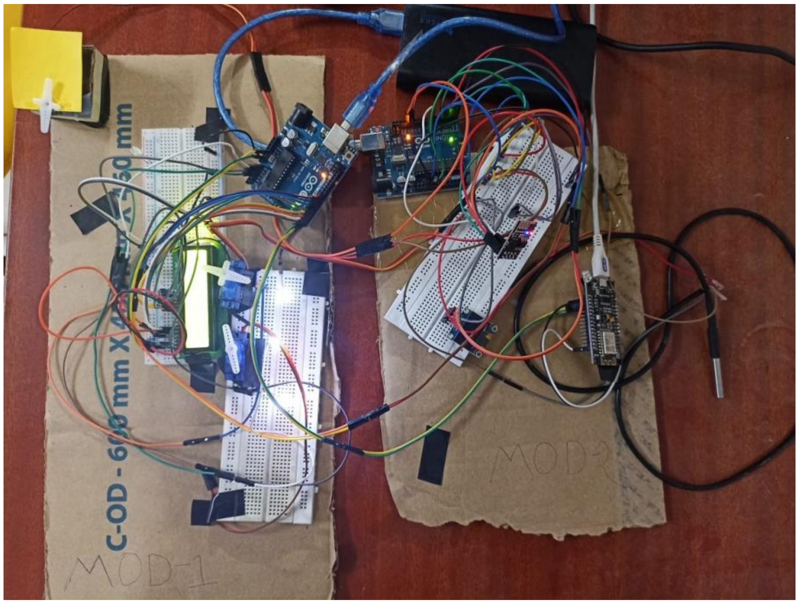


MEDICINE DISPATCHER CIRCUIT DIAGRAM

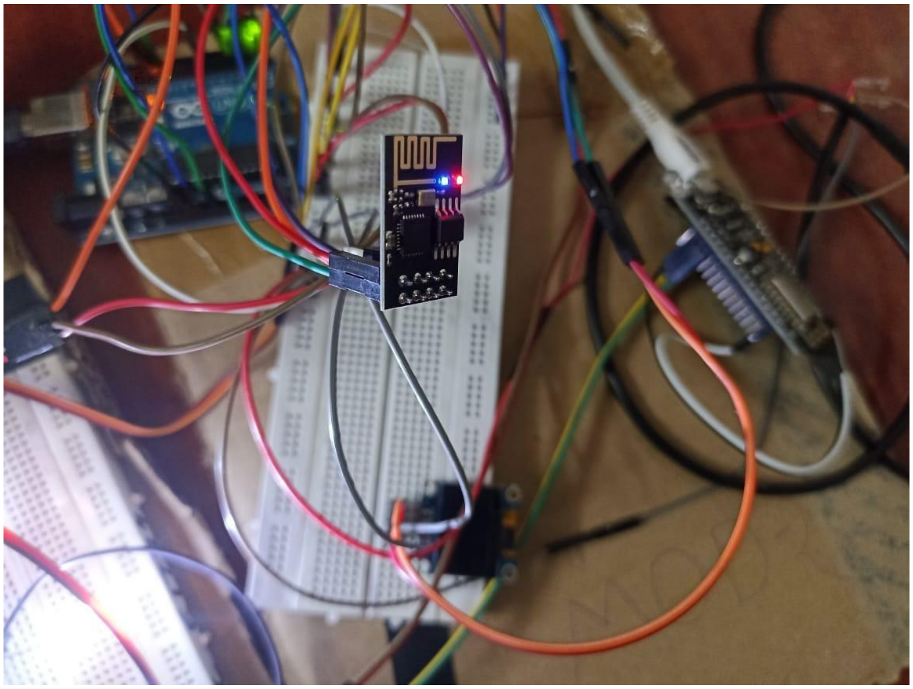


6. RESULTS AND DISCUSION

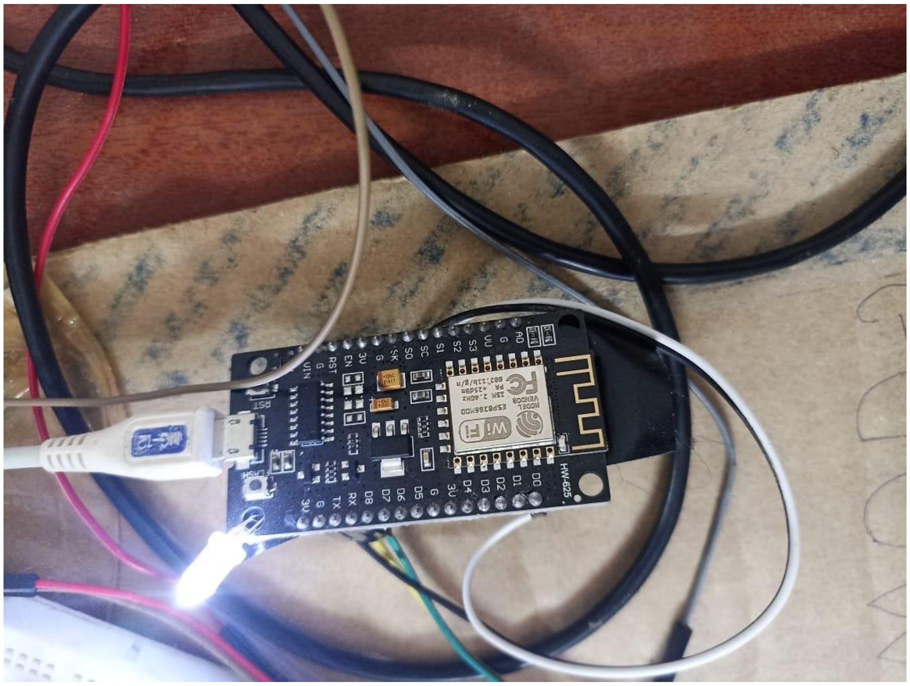
6.1 RESULTS



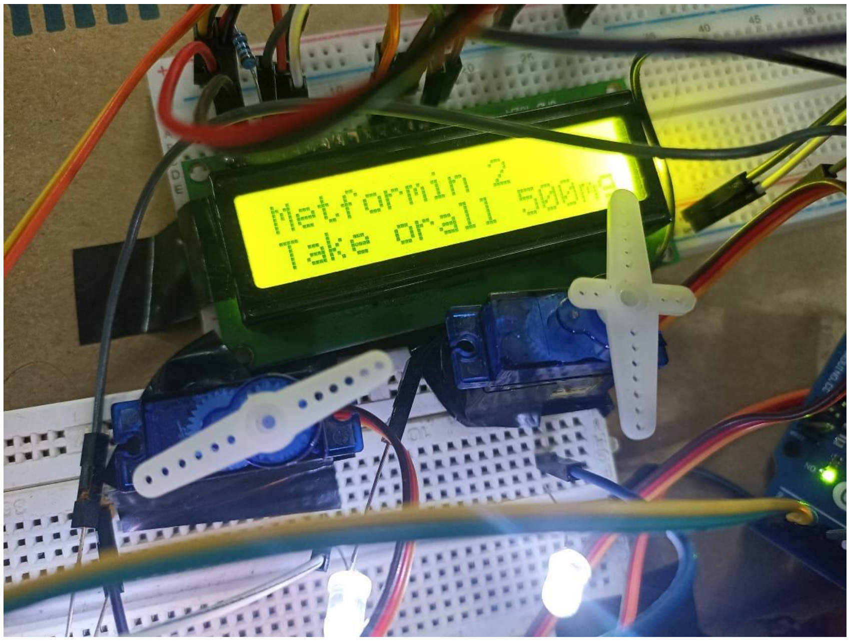
NODEMCU (ESP8266 – 01)



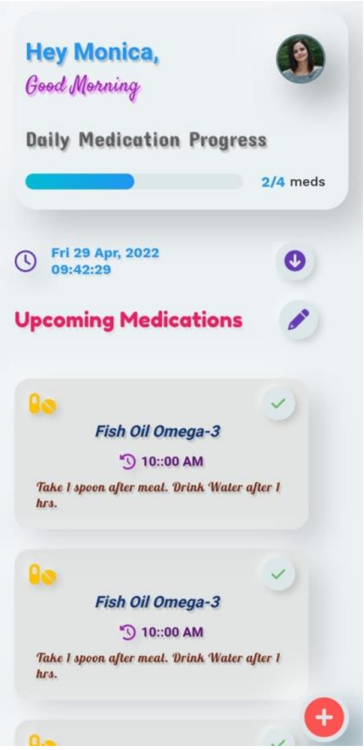
NODEMCU(ESP8266)



LCD screen displaying the output:



Screenshot of the working app:



Hey Monica,

Good Morning

Daily Medication Progress

2/4 meds

Fri 29 Apr, 2022
09:42:29

Upcoming Medications

Fish Oil Omega-3

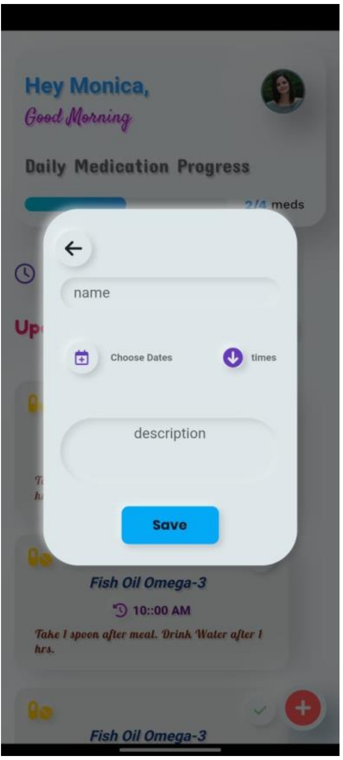
10::00 AM

Take 1 spoon after meal. Drink Water after 1 hrs.

Fish Oil Omega-3

10::00 AM

Take 1 spoon after meal. Drink Water after 1 hrs.



Hey Monica,

Good Morning

Daily Medication Progress

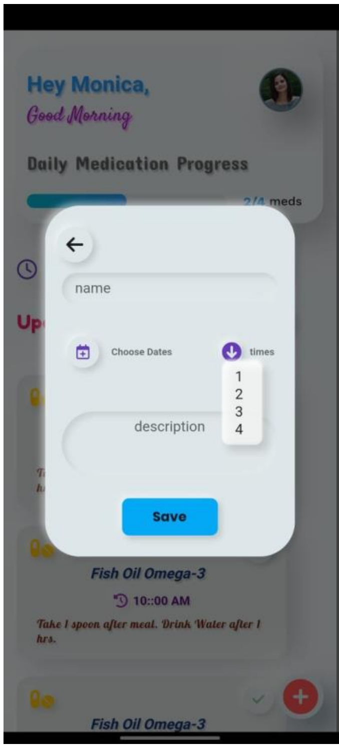
2/4 meds

Fish Oil Omega-3

10::00 AM

Take 1 spoon after meal. Drink Water after 1 hrs.

Fish Oil Omega-3



Hey Monica,

Good Morning

Daily Medication Progress

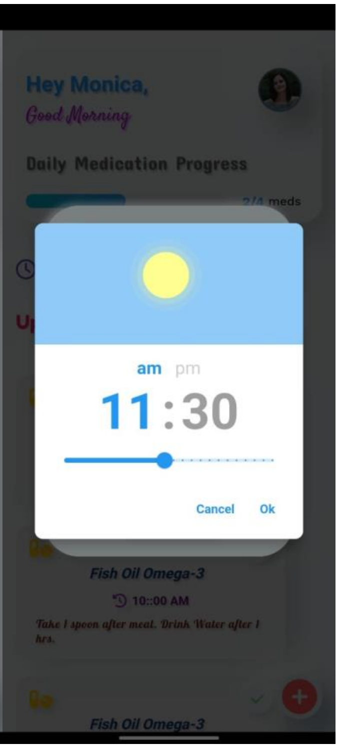
2/4 meds

Fish Oil Omega-3

10::00 AM

Take 1 spoon after meal. Drink Water after 1 hrs.

Fish Oil Omega-3



Hey Monica,

Good Morning

Daily Medication Progress

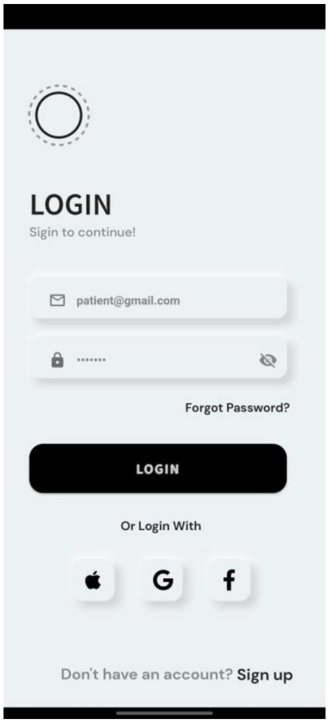
2/4 meds

Fish Oil Omega-3

10::00 AM

Take 1 spoon after meal. Drink Water after 1 hrs.

Fish Oil Omega-3



patient@gmail.com

•••••••

6.2 Discussion

6.2.1 Hardware Module

This module consists of two components. First is medicine reminder and other is patient monitoring system.

In the medicine reminder we have used servo motor, Arduino, LEDs, LCD display and node MCU. When the data is feed in the node MCU then the data is displayed in the LCD screen and the servo motor is turned on and it rotates the tray connected to it in the direction of the patient. This reminds the patient to take the medicine on time.

In the patient monitoring system we have used various sensors to sense the data from the patient so that it can be studied by the doctor and can be treated accordingly. We have used DHT11 sensor, Body temperature sensor, pulse sensor, OLED display, 16*2 LCD display, Arduino and node MCU. This component takes the data from the patient so that he or she can be treated in a better way. This modules sends the data gathered to the LCD and OLED screen display.

6.2.2 Software Module

This module consists of Arduino ide and the function of the application built using Flutter.

In this app we have created a login page and a home page which is connected to a backend using java springboot. In the login page you have to login via mail id and password. After this login you will be directed to the home page. In this home page you will have the medicine reminder software in which you can feed the data accordingly and after feeding the data when you will click save button then a small widget will appear on the screen of the home page which will consist of the name os the medicine, times need to be taken and the description of it. After this is done then on the basis of the data stored in the app the servo motor and the leds in the hardware will start working and the reminder will start reminding the patient about the time to take the medicine.

Then comes the coding for Arduino and node MCU which is done on Platformio. This coding has been done in C++ language. This code contains code for all the hardware components and devices.

7. CONCLUSION AND FUTURE WORKS

7.1 Conclusion

By this project we were successful in implementing all our modules. IoT is a great help and innovation in healthcare sector. With this belief our project is also concluded by making the IoT health care monitoring system and medicine dispatcher. We have assembled both the modules properly and integrated them with the help of two arduinos. The patient monitoring system was used to monitor the temp, pulse rate, heart rate, room temperature and humidity. In the medicine reminder we have used servo motors for the stay system.

We have also concluded with our software modules. And we have used platformio, learnt about it and implemented our hardware components code on it with no faults. We have also deployed the application made for medicine reminder which takes the data from the user and displays it in the app and also on the led screen.

So this project helped us to gain a lot of knowledge and we have used new software and hardware tools in it which was a great learning.

7.2 Future Works

Our Future work includes incorporating Deep learning model so that it allows the user to capture the medicine image and process it using the deep learning model and capture requireddetailsand embed it directly into the application. It completely automate the process of capturing the properties, description dosage from the image itself.

Apart from that we can also incorporate sensor which can include medicine reminder which can automatically identify the situation when the user has run out of medicine and can order it.

Hence the system can order the required medicine itself from the nearest medical store

8. REFERENCES

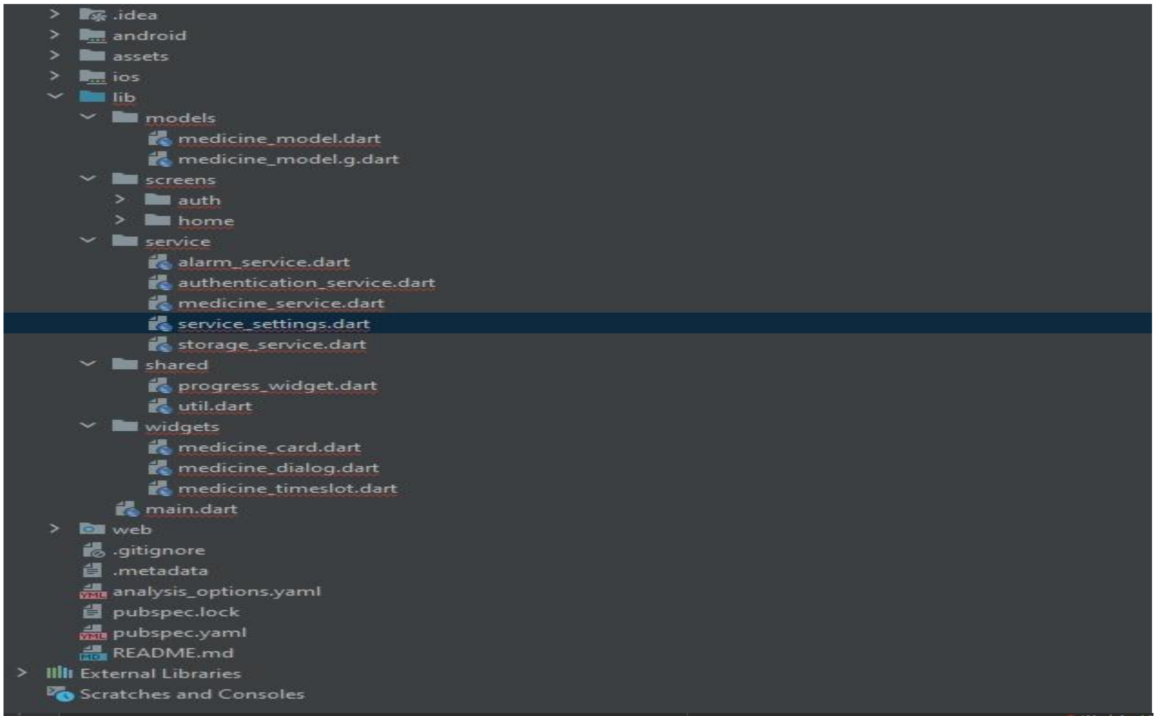
[1] Shola Usha Rani, “IoT Patient Health Monitoring System”, Indian Journal of Public Health Research and Development · October 2017, Available: DOI: 10.5958/0976-5506.2017.00519.8

[2] B.Naveen Naik, Rekha, Ajay Singh, “Real-Time Smart Patient Monitoring and Assessment Amid COVID-19 Pandemic”, Springer, Published: 13 June 2020,

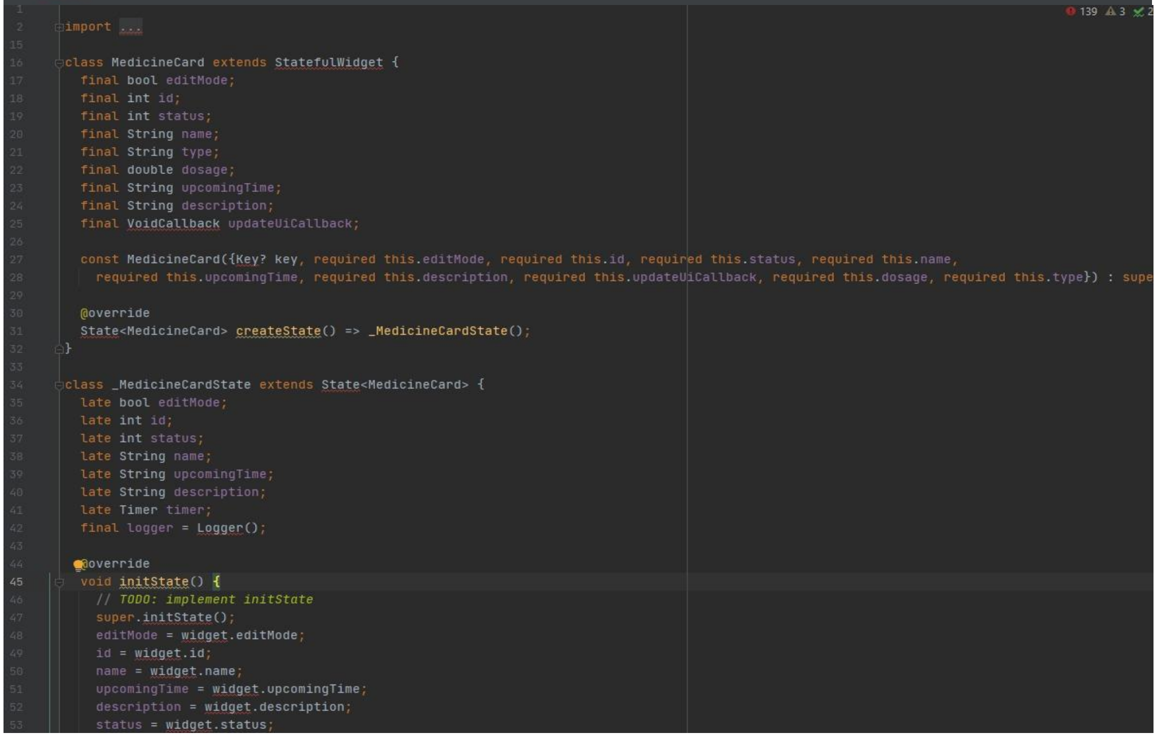
[3] JorgeGómeza,yronOviedo,” Patient Monitoring System Based on Internet of Things”,

9. APPENDIX

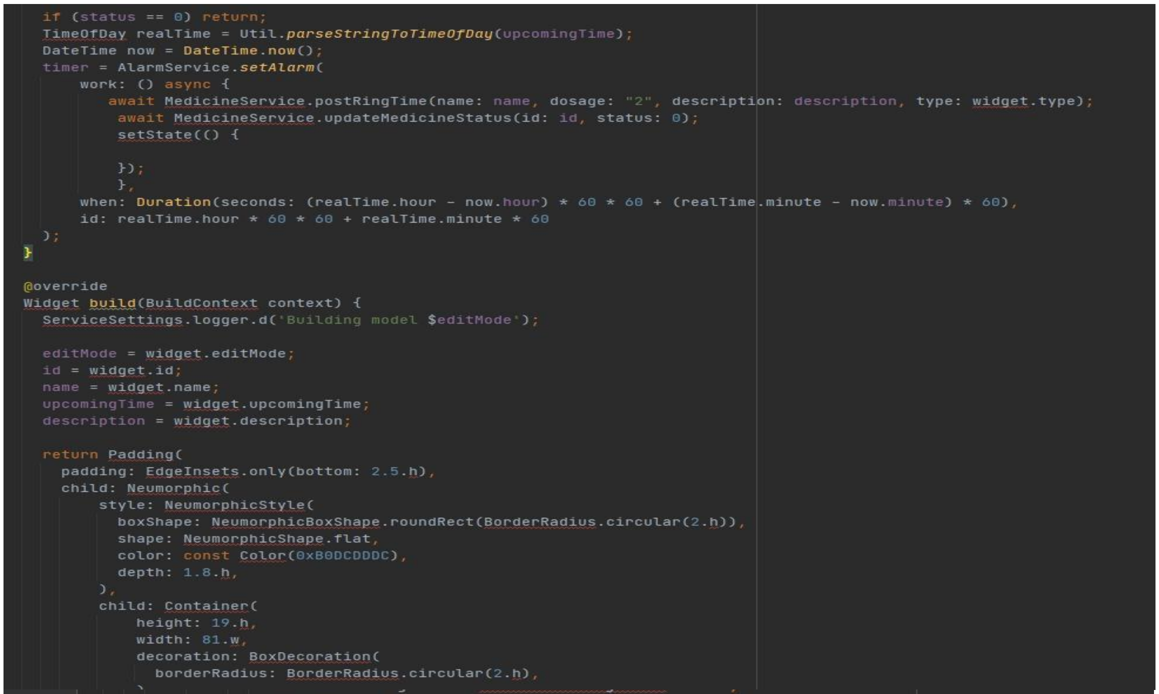
9.1 App Card code



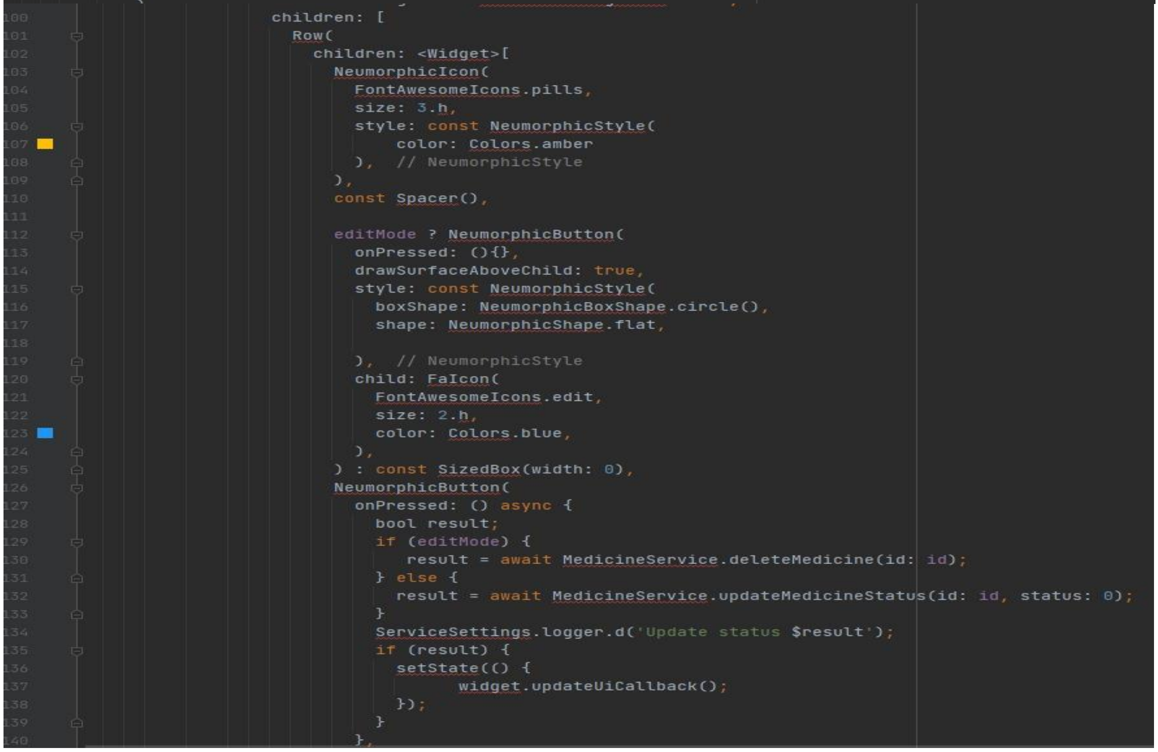
```
> .idea
> android
> assets
> ios
v lib
  v models
      medicine_model.dart
      medicine_model.g.dart
  v screens
    > auth
    > home
  v service
      alarm_service.dart
      authentication_service.dart
      medicine_service.dart
      service_settings.dart
      storage_service.dart
  v shared
      progress_widget.dart
      util.dart
  v widgets
      medicine_card.dart
      medicine_dialog.dart
      medicine_timeslot.dart
    main.dart
> web
  .gitignore
  .metadata
  analysis_options.yaml
  pubspec.lock
  pubspec.yaml
  README.md
> External Libraries
  Scratches and Consoles
```



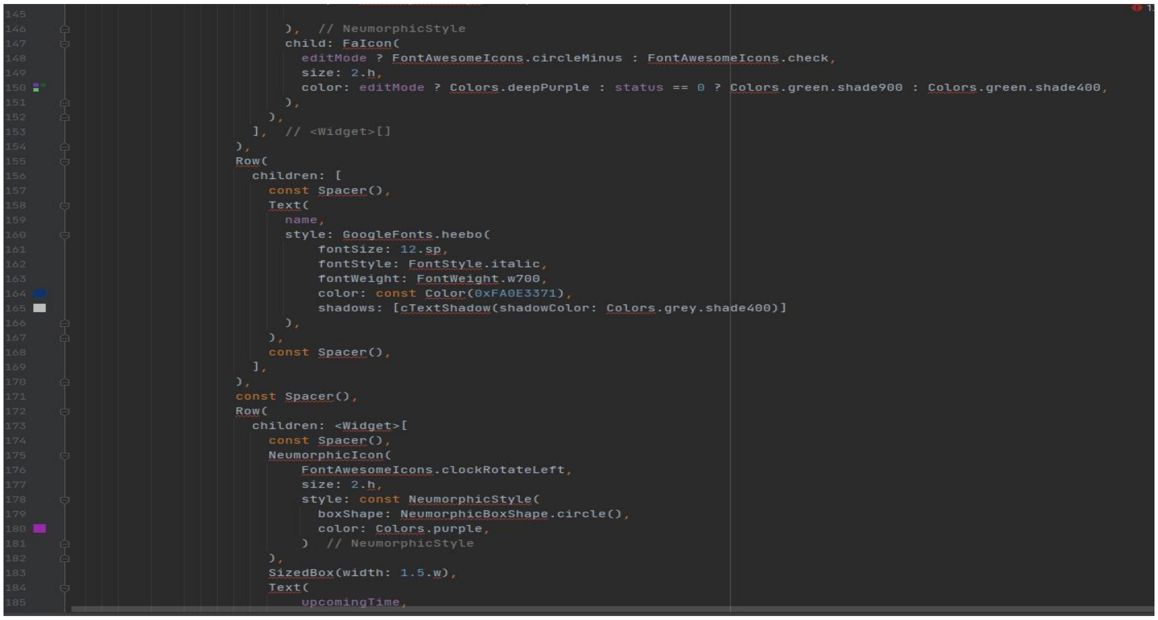
```
import ...

class MedicineCard extends StatefulWidget {
  final bool editMode;
  final int id;
  final int status;
  final String name;
  final String type;
  final double dosage;
  final String upcomingTime;
  final String description;
  final VoidCallback updateUiCallback;

  const MedicineCard({Key? key, required this.editMode, required this.id, required this.status, required this.name,
    required this.upcomingTime, required this.description, required this.updateUiCallback, required this.dosage, required this.type}) : supe

  @override
  State<MedicineCard> createState() => _MedicineCardState();
}

class _MedicineCardState extends State<MedicineCard> {
  late bool editMode;
  late int id;
  late int status;
  late String name;
  late String upcomingTime;
  late String description;
  late Timer timer;
  final logger = Logger();

  @override
  void initState() {
    // TODO: implement initState
    super.initState();
    editMode = widget.editMode;
    id = widget.id;
    name = widget.name;
    upcomingTime = widget.upcomingTime;
    description = widget.description;
    status = widget.status;
```



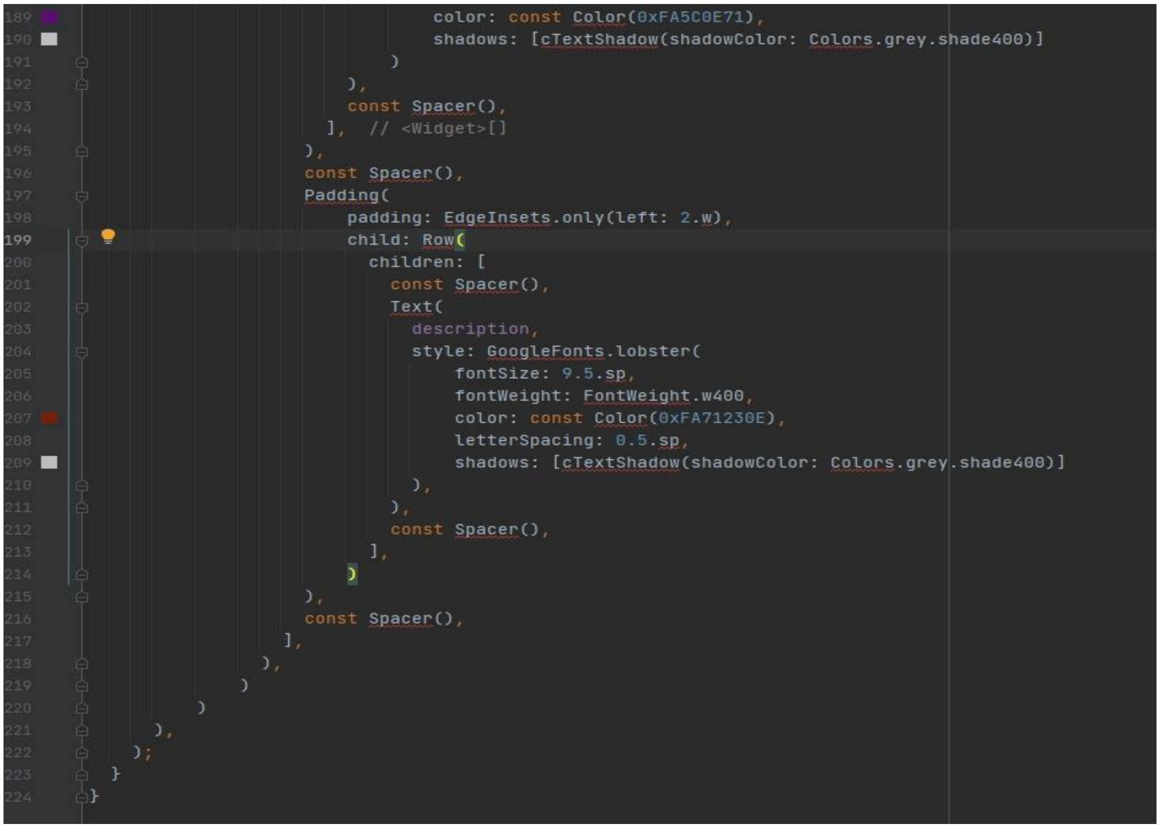
```
  if (status == 0) return;
  TimeOfDay realTime = Util.parseStringToTimeOfDay(upcomingTime);
  DateTime now = DateTime.now();
  timer = AlarmService.setAlarm(
      work: () async {
         await MedicineService.postRingTime(name: name, dosage: "2", description: description, type: widget.type);
          await MedicineService.updateMedicineStatus(id: id, status: 0);
          setState(() {

          });
          },
      when: Duration(seconds: (realTime.hour - now.hour) * 60 * 60 + (realTime.minute - now.minute) * 60),
      id: realTime.hour * 60 * 60 + realTime.minute * 60
  );
}

@override
Widget build(BuildContext context) {
  ServiceSettings.logger.d('Building model $editMode');

  editMode = widget.editMode;
  id = widget.id;
  name = widget.name;
  upcomingTime = widget.upcomingTime;
  description = widget.description;

  return Padding(
    padding: EdgeInsets.only(bottom: 2.5.h),
    child: Neumorphic(
        style: NeumorphicStyle(
          boxShape: NeumorphicBoxShape.roundRect(BorderRadius.circular(2.h)),
          shape: NeumorphicShape.flat,
          color: const Color(0xB0DCDDDC),
          depth: 1.8.h,
        ),
        child: Container(
            height: 19.h,
            width: 81.w,
            decoration: BoxDecoration(
              borderRadius: BorderRadius.circular(2.h),
```



```
              children: [
                Row(
                  children: <Widget>[
                    NeumorphicIcon(
                      FontAwesomeIcons.pills,
                      size: 3.h,
                      style: const NeumorphicStyle(
                          color: Colors.amber
                      ),  // NeumorphicStyle
                    ),
                    const Spacer(),

                    editMode ? NeumorphicButton(
                      onPressed: (){},
                      drawSurfaceAboveChild: true,
                      style: const NeumorphicStyle(
                        boxShape: NeumorphicBoxShape.circle(),
                        shape: NeumorphicShape.flat,

                      ),  // NeumorphicStyle
                      child: FaIcon(
                        FontAwesomeIcons.edit,
                        size: 2.h,
                        color: Colors.blue,
                      ),
                    ) : const SizedBox(width: 0),
                    NeumorphicButton(
                      onPressed: () async {
                        bool result;
                        if (editMode) {
                           result = await MedicineService.deleteMedicine(id: id);
                        } else {
                          result = await MedicineService.updateMedicineStatus(id: id, status: 0);
                        }
                        ServiceSettings.logger.d('Update status $result');
                        if (result) {
                          setState(() {
                                widget.updateUiCallback();
                          });
                        }
                      },
```



```
      ),  // NeumorphicStyle
      child: FaIcon(
        editMode ? FontAwesomeIcons.circleMinus : FontAwesomeIcons.check,
        size: 2.h,
        color: editMode ? Colors.deepPurple : status == 0 ? Colors.green.shade900 : Colors.green.shade400,
      ),
    ),
  ],  // <Widget>[]
),
Row(
  children: [
    const Spacer(),
    Text(
      name,
      style: GoogleFonts.heebo(
          fontSize: 12.sp,
          fontStyle: FontStyle.italic,
          fontWeight: FontWeight.w700,
          color: const Color(0xFA0E3371),
          shadows: [cTextShadow(shadowColor: Colors.grey.shade400)]
      ),
    ),
    const Spacer(),
  ],
),
const Spacer(),
Row(
  children: <Widget>[
    const Spacer(),
    NeumorphicIcon(
        FontAwesomeIcons.clockRotateLeft,
        size: 2.h,
        style: const NeumorphicStyle(
          boxShape: NeumorphicBoxShape.circle(),
          color: Colors.purple,
        )  // NeumorphicStyle
    ),
    SizedBox(width: 1.5.w),
    Text(
        upcomingTime,
```



```
                                color: const Color(0xFA5C0E71),
                                shadows: [cTextShadow(shadowColor: Colors.grey.shade400)]
                            )
                        ),
                        const Spacer(),
                      ],  // <Widget>[]
                    ),
                    const Spacer(),
                    Padding(
                        padding: EdgeInsets.only(left: 2.w),
                        child: Row(
                          children: [
                            const Spacer(),
                            Text(
                              description,
                              style: GoogleFonts.lobster(
                                  fontSize: 9.5.sp,
                                  fontWeight: FontWeight.w400,
                                  color: const Color(0xFA71230E),
                                  letterSpacing: 0.5.sp,
                                  shadows: [cTextShadow(shadowColor: Colors.grey.shade400)]
                              ),
                            ),
                            const Spacer(),
                          ],
                        )
                    ),
                    const Spacer(),
                  ],
                ),
              )
          )
      ),
    );
  }
}
```