IV DRIP and HEART RATE MONITORING SYSTEM

A PROJECT REPORT

for

LEAN START – UP MANAGEMENT (MGT1022)

by

JAYA SRI S (20MID0092)

MUDUMBA SIDDHARTHA (20MID0144)

NITHISH KUMAR S (20MIS0024)

ADITYAN (20MIS0109)

YASH HEDA (20MIS0300)

6th Semester, 3rd Year

Under the Guidance of

Prof. RAJAY VEDARAJ

Professor Grade 1

VIT®

Vellore Institute of Technology

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

School of Information Technology and Engineering

MAR, 2023

IV Drip and Heart Rate Monitoring System

Jaya Sri S[1], Mudumba Siddhartha[2], Nithish Kumar S[3], Adityan[4], Yash Heda[5]

[1, 2, 3, 4, 5] **VIT University, Vellore, Tamil Nadu, India**

Abstract

This paper presents an introduction of numerous gadgets in the medical profession that had a significant impact on the interdisciplinary monitoring of bodily functions like blood pressure, heart rate, heart attack symptoms, and much more. The value of the healthcare system is rising right now. The suggested device automatically notifies the nurse using GSM technology when the IV fluid level drops below a certain level. On the liquid crystal display, a continuous monitoring system continuously shows the patient's blood pressure and heart rate (LCD). Instead of constantly monitoring an IV fluid system, this technology decreases the workload of the nurse. As intravenous (IV) fluid is typically supplied in crowded operating rooms (ORs) using the gravity-fed IV administration technique rather than an infusion pump, IV bags often empty unnoticed. Each person's heart rate helps to determine their fitness level and by monitoring this, you are able to avoid overtraining, which in turn can reduce the risk of injury and mental fatigue. The rates of death linked to, or caused by, cardiovascular disease have declined over the years, but according to the World Health Organization, it is still the number one cause of death globally, taking an estimated 17.9 million lives each year. So, there has never been a better time to build your understanding of your own heart health and what you can do to improve it. Regular, accurate, monitoring will not only reduce your risk of heart and circulatory disease but it is also a great start to understanding and improving your overall health. IV therapy is susceptible to human error because, traditionally, medical professionals estimate how long it will take an IV bottle to empty based on their experience.

Keywords – Drop rate Monitoring, IV infusion, Heart Rate Monitoring, Health care

I. INTRODUCTION

Intravenous Drip System is used mainly in hospitals for patients who were dehydrated, nutrient deficient or unable to take medications orally. In our current medical care system, monitoring of patients in a hospital throughout the day is a tiresome process. Sometimes doctors or nurses are too busy, so they cannot monitor each patient. This causes many problems. The health related work should be done properly and with accurate manner. In busy operating rooms (ORs), intravenous (IV) fluid is generally administered via the gravity – fed IV delivery system and not through an infusion pump and so IV bags often empty unnoticed. A procedure performed in a dimly lit OR that might increase the risk of an IV bag running dry. If the drip system is not monitored on time, it will cause problems like backflow of fluid, blood loss etc. Heart rate is important because the heart's function is so important. The heart circulates oxygen and nutrient – rich blood throughout the body. When it's not working properly, just about everything is affected. Heart rate is central to this process because the function of the heart is directly related to heart rate and stroke volume. Keeping track of heart rate can give us insight into fitness level, heart health and emotional health of a person. Many people are walking around with a resting heart rate that is too high, due to factors such as too much caffeine, dehydration, inactivity and persistent stress. Those extra heart beats over time can be taking years off of people's life. Therefore, in order to reduce the workload of nursing staff and to overcome such critical situation in the area of an intravenous drip and heart rate monitoring, we proposed a system called Intravenous Drip and Heart Rate Monitoring System.

II. BACKGROUND

Internet of Things (IoT) has changed people's lives in many areas, especially in healthcare sector. It has enabled devices which made remote monitoring in the healthcare sector possible. It has the potential to keep patients safe and healthy, and empowering doctors and nursing staff to deliver high quality care. One of them is Intravenous Drip and Heart Rate Monitoring System. Monitoring patients during IV therapy is still a challenging problem. In our current medical care system, we manually do all this monitoring task. We need to alert the medical staff about the drip level in a saline bottle that is being injected through the patient's vein and the patient condition on a real time. IV therapy is an effective and fast way to administers fluid and medications treatment in emergency situations, and for patients who are unable to take medications orally. Approximately 80% of all patients in the hospital will receive intravenous therapy. The main perk of IV is that the fluids can be delivered in the fastest mode throughout the body and immediate effect of the medication can be achieved. The rates of death linked to, or caused by, cardiovascular disease have declined over the years, but according to the World Health Organization, it is still the number one cause of death globally, taking an estimated 17.9 million lives each year. So, there has never been a better time to build your understanding of your own heart health and what you can do to improve it. Regular, accurate, monitoring will not only reduce your risk of heart and circulatory disease but it is also a great start to understanding and improving your overall health. The flow measurement and control system can be developed by using flow sensor the output of the sensor which is in the form of pulses is given to the Arduino Uno R3 controller. Hence to assure the safety of the patient during IV period there is need to develop an efficient health monitoring system. This is going to reduce the stress in continual monitoring by the doctor or nurse at an affordable cost.

III. PROPOSED IDEA

Our project is aimed in automating the intravenous fluid monitoring system using Arduino Uno R3. IV volume and fluid level can be precisely controlled. Also human can contact the system through GSM (Global System for Mobile communication). In IV fluid monitoring system is failed to disconnect the drip system to patient, Air-in line sensor will be activated. All most in all hospitals, assist / nurse is responsible for monitoring the IV fluid level system. But unfortunately, the observer may forget to change or stop the drip bottle at correct time due to their schedule. This may lead to several problems to the patients. Our project is overcome for this critical situation. This technology reduces the work of the observer. Intravenous therapy is the infusion of fluid substances directly into a vein. Intravenous simply means "within vein". IV system may be used to correct fluid imbalances, to deliver machines, for blood transfusion or as fluid replacement to correct. This way is the fastest way to deliver medicines or fluids. Therefore, it is necessary to monitor treatment through IV therapy.

IV. LITERATURE SURVEY

- **Real-time cost-effective e-saline monitoring and control system**

  The experimental arrangement is verified for various test cases for normally running condition and at different failure condition is mentioned in below on GUI screen the GUI obtained on the PC. When sensor cannot detect a drop in 30sec then the GUI displays the notification along with green light glow and when the level of fluid is below at set of limits then it alerts with notification along with yellow light glow and buzzer played.

- **Smart Saline Level Monitoring System using ESP32 and MQTT-S**

  By the weight of the saline bottle, the level of liquid can be estimated so that when the liquid reaches to its minimum level, ESP32 WIFI module is used to send alerts to end subscribers via MQTT-S communication protocol.

- **Intravenous Drip Monitoring System**

  An Automatic intravenous drip monitoring system is developed which directly sends an alert message to the assigned nurse when the fluid level of the bottle reaches a certain limit. This system measures the weight of the saline bottle with the help of a load cell and then using an automatic alerting and indicating device namely GSM sends the alert signal. This system would be a significant serve to build a different approach toward the intravenous therapy.

- **A new drip infusion solution with a free flow detection function**

  This high-accuracy system needs only three copper foil electrodes as its drop and free-flow sensor. Since these electrodes are on the external surface of the tubes and the chamber, they are not in electrical contact with the infusion solution.

- **Intravenous Infusion Control and Monitoring System**

  The proponents were able to develop a wireless infusion control and monitoring system for intravenous fluids. The developed system allows users to input IV prescriptions for various patients in a graphical user interface that is connected to a database. The prototype can then wirelessly retrieve the prescription from the database and use the data to calculate the drop rate. This calculated drop will then facilitate the gripping mechanism of the flow control.

- **Real-time drip infusion monitoring through a computer vision system**

  A method based on deep learning computer vision for IV drip monitoring was proposed. This technique was found to be less invasive than other available solutions, which require a direct contact with the infusion kit, offering good accuracy performance anyway.

- **Intravenous (IV) Drip Rate Controlling and Monitoring for Risk-Free IV Delivery**

  The traditional method of infusion was replaced by using embedded system technology, a system for detecting the variations in light transmission between a LED and a photodiode placed around IV drip chamber to monitor IV drops. This system was designed to run with a battery but it does not have a regulator to save battery power

- **Live tracking of saline for betterment of patient**

  The planned system includes of devices which can act as tier sensor for observance the crucial level of the saline within the saline bottle. Whenever the amount of the saline reaches to the predefined crucial level, then the nurses, caretaker, doctors are alerted through the alarm associate in Nursing an alert message are sent through the utilization of web to the involved nurses and doctors that there's a requirement for replacement of the saline bottle.

- **IV Bottle Level Monitoring System**

  We were able to detect the variation in reflected Wi-Fi signal and convert this reflected signal to DC voltage. Keeping a threshold value of indication, controller compares received DC voltage with threshold voltage and proper alert is given. Alert messages are sent once the fluid reaches 1/3rd of the bottle. Novelty of the proposed system is that no overhead expense is imposed on the hospital management as Wi-Fi signals are used for signal source.

- **Smart Hospital Saline Monitoring System**

  Blood leakage detection and saline level system is used to monitor the blood leakage occurrence is detected by non-invasively and it can be accessed by easy installation of the detector on the human arm and with the help of load cell saline level will monitor. Based on patient's temperature, heart beat value, saline flows solenoid valves will adjust the saline flow speed automatically slow or fast.

- **An IoT Based Intravenous Drip Rate Controlling and Monitoring Device**

  Generic design to match all sizes of drip sets, control and monitoring through Smartphone / web application, robust design, cost effective and user-friendly solution. Implementation of the proposed device helps in curbing multiple issues and ensures accurate monitoring and control of IV system leading to an efficient, affordable and reliable indigenous solution.

- **Automatic Saline Level Monitoring System Using IOT**

  The programming is based on Arduino platform which is done using C compiler. The results are obtained on Web page or Smartphone App with the help of Bluetooth terminal software and are obtained on computer or laptop using serial port test software. The results contain number of droplets coming from saline bottle, the solution given to patient in ml, the droplet rate and remaining solution in bottle. With IoT based saline level monitoring system, the manual effort on the part of the nurses is saved. As the entire proposed system is automated, it requires very less human intervention. It will be advantageous at night as there will be no such requirement for the nurses to visit patient's bed every time to check the level of saline in the bottle since an alert notification will be sent to the nurses, doctors, caretakers when saline reaches the critical level. It will save the life of the patients. This will reduce the stress in continual monitoring by the doctor or nurse at an affordable cost

- **IoT Based Drip Infusion Monitoring System**

  The use of an ultrasonic sensor simplifies and expedites system implementation because it eliminates the need to calibrate the system for different fluids. LDR is used to detect bubble formation of the fluid which eliminates the risk of arterial air embolism which can cause heart attacks, stroke or respiratory failure. This system may be quickly installed on the stand where the drip bottle is hung

- **IV Drip Monitoring and Control System**

The IV Drip Monitoring and Controlling System is designed and tested successfully. the system includes NodeMCU ESP8266 Controller, Load Cell, Servo Motor with clam, Saline Bottle, Buzzer, and External Switch. The system is designed to capture the changes in the level of saline bottle and determine the level of saline bottle. When the determined level is less than predefined threshold weight, then the buzzer sounds to notify the nursing staff.

- **Smart Intravenous Monitoring System with Bubble Detection Indicator**

Implementing a smart intravenous fluid level indicator which relieves the medical practitioner's stress from having a constant check on IV fluid containers. The fluid level is measured and transformed into an electric signal, which is then sent to the microcontroller. If the predetermined point is exceeded, alert system gets activated which notifies the nurse to halt the flow of fluid in the patient's vein.

- **Saline Level Monitoring using Liquid Level Switch Contactless Sensor**

The MQTT-S protocol was chosen because it is efficient and facilitates communication between the publisher and subscriber. In offline mode, the message is read using the Buffering mechanism. It is obvious that the developed system can remotely monitor the patient's saline level and assist doctors, nurses, and caretakers as an added benefit to the smart Healthcare service.

- **DripOMeter: An open-source opto-electronic system for intravenous (IV) infusion monitoring**

The system presented here accurately tracks the fluid flow and assists the users in monitoring the infusion sessions. The system generates alarm upon detecting significant deviation from set drip rate. The system keeps track of total volume infused and alerts when a desired volume is about to be administered. The device offers a solution to reduce the risks associated with the IV infusion therapy especially in low-resource setting and provide peace of mind to caregivers.

- **IoT Based Automatic Saline Level Monitoring System**

  It requires very less human intervention. So, the manual effort on the part of the nurses is saved. It will be more advantageous at night as there will be no such requirement for the nurses to visit the patient's bed every time to check the level of saline in the bottle since an alert notification will be sent to the nurses, doctors, and caretakers when saline reaches the critical level. Our system provides more flexibility to doctors, thereby the patient's care is enhanced

- **IoT based Saline Monitoring System**

  This paper proposes the automated approach to monitoring the Saline Fluid in the bottle and furthermore to stop the flow of saline using solenoid valve. The proposed system is suitable for use in hospitals via a computer or smartphone, doctors or nurses can screen the Saline level, temperature, oxygen level in the blood, and any patient's heart rate can be accessed at any time and from any place

- **Drip Monitoring and Reverse Blood Flow Prevention System**

  It can be fixed with the electrolyte bottles. Thus, it will eliminate all the errors caused due to manual monitoring of electrolyte bottles. It can also avoid other problems such as flow of air bubbles and reverse flow of blood caused due to the improper monitoring of electrolyte levels.

Title	Year	Advantages	Issues
Real-time cost-effective e-saline monitoring and control system	2016	If the level of fluid is above the critical set level, then the system will notify “bottle almost finished status” on control server (GUI) as a yellow light with a buzzer sound	These systems can malfunction, leading to incorrect readings or alerts. This can result in harm to the patient or delays in treatment.
Smart Saline Level Monitoring System using ESP32 and MQTT-S	2018	By the weight of the saline bottle, the level of liquid can be estimated so that when the liquid reaches to its minimum level, ESP32 Wi-Fi module is used to send alerts to end subscribers via MQTT-S communication protocol.	Any slightest change in the drip rate of IV fluid might cause severe side effects.
Intravenous Drip Monitoring System	2018	The IV can be monitored in mobile phone by the doctors and the staffs. The IV gets turned off automatically.	Mobile app is message based, not user friendly, drip level not shown to nurse continuously, uses weight sensor which can malfunction, doesn’t detect bubbles
A new drip infusion solution with a free flow detection function	2019	Free-flow is monitored by the impedance between two electrodes wrapped around the infusion supply polyvinyl chloride tubes from the solution bag and the drip chamber.	Certain medications or fluids may not be compatible with the materials used in the IV monitoring system, which can lead to problems with infusion or inaccurate readings.

Intravenous Infusion Control and Monitoring System	2019	Based on the current drop rate as observed by the sensing module, the grip will tighten or loosen.	Over hydration must be prevented particularly for patients who require small volumes of fluid.
Real-time drip infusion monitoring through a computer vision system	2020	The usage of deep learning algorithms makes the estimation method more robust to variable environmental conditions and more versatile.	The IV monitoring system can become obstructed, which can prevent the proper delivery of fluids or medications to the patient.
Intravenous (IV) Drip Rate Controlling and Monitoring for Risk-Free IV Delivery	2020	They have used the latest technology.	Their project doesn't concentrate on detecting any bubble formation in the solution.
Live tracking of saline for betterment of patient	2020	The IV can be monitored in mobile phone by the doctors and the staffs. The IV gets turned off automatically.	Their project doesn't concentrate on detecting any bubble formation in the solution.
IV Bottle Level Monitoring System	2021	This IV bag monitoring system can not only be used for the detection and signaling of glucose in the bag but also for blood and some of its components. In addition to that, it can be applied to urine bags in immobile patients or drainage bags in postoperative care	IV bottle level monitoring systems require maintenance, such as regular cleaning and calibration, which can add to the workload of healthcare providers.

Smart Hospital Saline Monitoring System	2021	1. Saline level using load cell. 2. Monitor Saline speed according to heat beat. 3. Temperature and Blood leakage detector during hemo-dialysis process.	Some patients may find the constant monitoring of their saline level to be intrusive or uncomfortable.
An IoT Based Intravenous Drip Rate Controlling and Monitoring Device	2021	● Dual notification in terms of Light and sound alarms to indicate termination of flow ● Automatic start / stop of the device through Smartphone or web applications.	Decrease in the drip rate leads to decrease in the amount of fluid received by the patient, this leads to dehydration and metabolic imbalance.
Automatic Saline Level Monitoring System Using IOT	2021	Stops of the supply of the drip on the value that has been set by the doctors or staffs.	This project uses desktop application, which is not portable.
IV Drip Monitoring and Control System	2022	Stops of the supply of the drip on the value that has been set by the doctors or staffs.	The load sensors used in this project can malfunction. Their project doesn’t concentrate on detecting any bubble formation in the solution.
Smart Intravenous Monitoring System with Bubble Detection Indicator	2022	Used to automate and regulate the flow of fluid as per the requirement as prescribed by the medical practitioner	The drawback of this system is that it does not have any means of notifying the medical staff if they are not in the patient’s room.

Saline Level Monitoring using Liquid Level Switch Contactless Sensor	2022	MQTT-S is a low-cost, low-power, and lightweight protocol designed primarily for Internet of Things (IoT) applications. It employs publish/subscribe approaches	The contactless sensor may malfunction, leading to incorrect readings or alerts. This can result in harm to the patient or delays in treatment.
DripOMeter: An open-source opto-electronic system for intravenous (IV) infusion monitoring	2022	This project is efficiently built and contains lesser number of components compared to other projects.	Their project doesn't concentrate on detecting any bubble formation in the solution. The implementation of a mobile application isn't focused in this project.
IoT Based Automatic Saline Level Monitoring System	2022	This project makes sure to monitor oxygen levels and heart rate of the patient.	The implementation of a mobile application isn't focused in this project.
IoT based Saline Monitoring System	2022	This project makes sure to monitor temperature and heart rate of the patient.	Sends message to nurse continuously rather than using an application for monitoring.
Drip Monitoring and Reverse Blood Flow Prevention System	2023	Avoid problems such as flow of air bubbles and reverse flow of blood caused due to the improper monitoring of electrolyte levels.	Allows users only to monitor the electrolyte level in the Intravenous Fluid (IV) saline bottle. And hence it increases the chance of reverse blood which may cause air embolism.

V. BILL OF MATERIALS

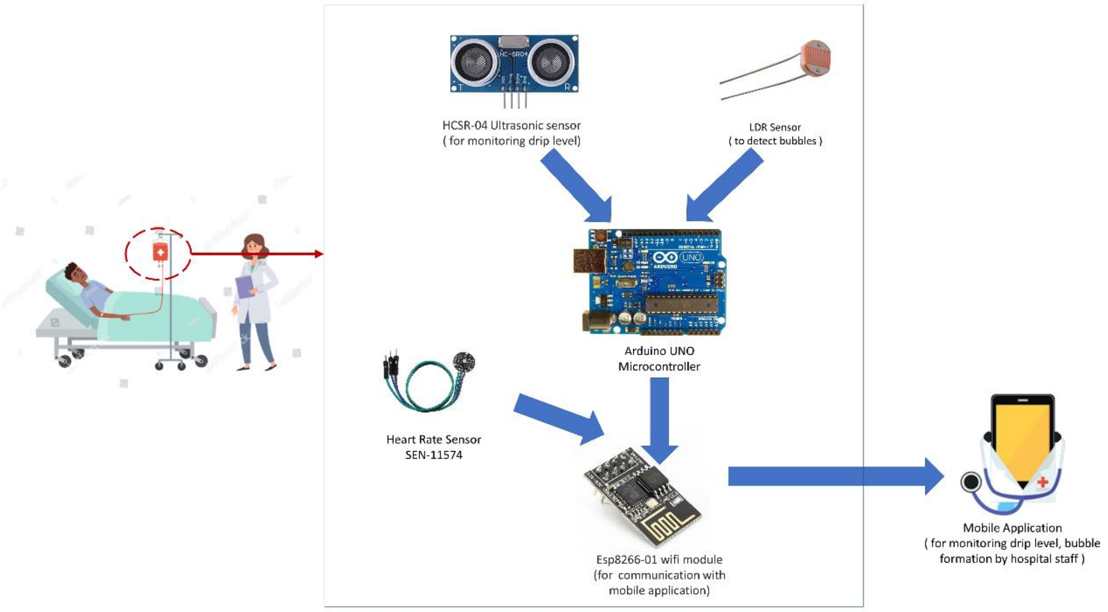
Our project is aimed in reducing the stress of continual monitoring by the doctor or nurse at an affordable cost. The components involved in the IV Drip and Bubble Monitoring System are the following:

- HC - SR04 Ultrasonic Sensor
- LDR Sensor
- Arduino UNO R3 Controller
- ESP8266 – 01 Wi – Fi Module
- Heart Rate Sensor
- Jumper Cables

Component	Cost (₹)
HC - SR04 Ultrasonic Sensor	119
LDR Sensor	7
Arduino UNO R3 Controller	295
ESP8266 – 01 Wi – Fi Module	107
Heart Rate Sensor	210
Jumper Cables	30
Total	**768**

VI. ARCHITECTURE DESIGN

Architecture Design For Drip & Bubble Monitoring System



VII. EXISTING SYSTEM

The existing hospital drip systems work by constantly monitoring the administration of intravenous (IV) medications, fluids, and nutrients to patients. The system typically consists of several interconnected components. A nursing staff required continuous monitoring of the saline bottle level to avoid backflow of blood from the patient to the saline bottle. Existing system uses a roller clamp mechanism for controlling the flow rate of the intravenous fluid. If the nursing staff delays to notice the flow rate, then it will lead to backflow of blood from patient which is risky.

VIII. INNOVATION COMPONENT

- Wireless technology:

  IV drip monitoring systems incorporate wireless technology, which allows healthcare providers to monitor the IV administration remotely. This can be especially useful for patients who are in critical condition or for those who are unable to leave their bed.

- Smart infusion sets:

Smart infusion sets are infusion sets that use sensors to monitor the administration of the IV solution. For example, they may use sensors to detect if there is a blockage in the tubing or a change in the flow rate.

- Heart Rate Monitoring:

Heart rate monitoring is important because the heart's function is so important. The heart circulates oxygen and nutrient-rich blood throughout the body. Heart rate is central to this process because the function of the heart is directly related to heart rate and stroke volume

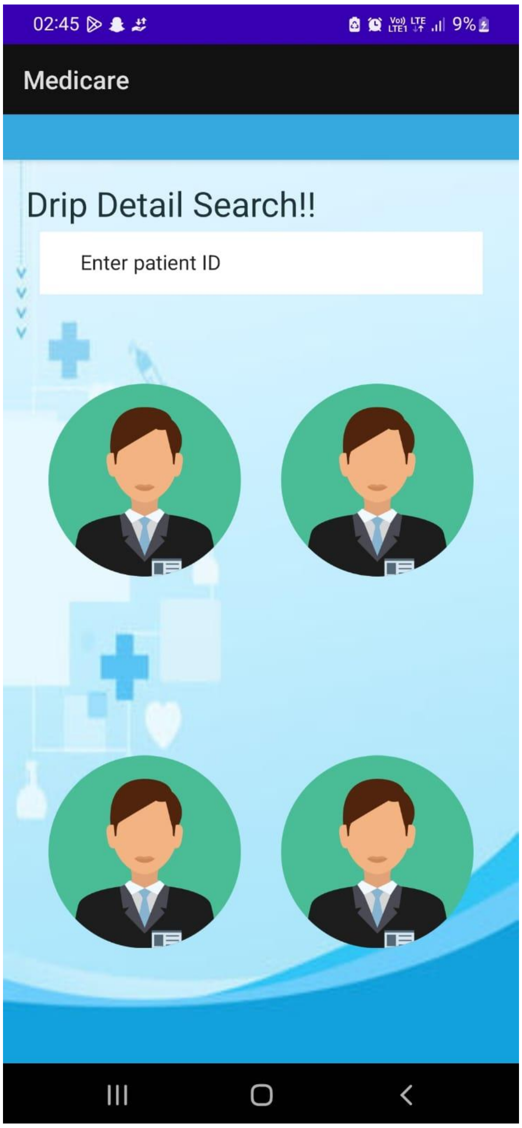
- Mobile apps

IV drip monitoring systems come with mobile apps that allow healthcare providers to monitor the IV administration from their mobile devices. This allows them to access important information about the patient and the IV administration even when they are not in the hospital.

- Interoperability:

Interoperability is becoming increasingly important in hospital IV drip monitoring systems, as it allows different devices and systems to work together seamlessly. This can help improve the overall efficiency of the administration of IV treatments and ensure that patients receive the right treatments at the right time.

IX. SCREENSHOT



X. APPENDIX

```
#include <ESP8266WiFi.h>
#include <FirebaseESP8266.h>
#define FIREBASE_HOST "heda-drip-default-rtdb.firebaseio.com"          //Your Firebase Project URL goes here without "http:" , "\" and "/"
#define FIREBASE_AUTH "rNJ0EuDxgEJznMGVHP4dYsVIGzMws7wwkxIHy9ys" //Your Firebase Database Secret goes here
#define WIFI_SSID "username"          //WiFi SSID to which you want NodeMCU to connect
#define WIFI_PASSWORD "12345678"
```

```
 //Password of your wifi network
#define echoPin 14 // attach pin D5 Arduino to pin Echo of HC-SR04
#define trigPin 12 //attach pin D6 Arduino to pin Trig of HC-SR04

// defines variables
long duration; // variable for the duration of sound wave travel
int distance; // variable for the distance measurement
int const buzzPin = 2; //D4

// Declare the Firebase Data object in the global scope
FirebaseData firebaseData;

void setup() {

  Serial.begin(115200);                    // Select the same baud rate if you want to see
the datas on Serial Monitor

  Serial.println("Serial communication started\n\n");
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT
  pinMode(buzzPin, OUTPUT); // buzz pin is output to control buzzering
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);                    //try to connect with
wifi
  Serial.print("Connecting to ");
  Serial.print(WIFI_SSID);



  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
```

```
  Serial.println();
  Serial.print("Connected to ");
  Serial.println(WIFI_SSID);
  Serial.print("IP Address is : ");
  Serial.println(WiFi.localIP());                              //print local IP address
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);   // connect to firebase

  Firebase.reconnectWiFi(true);
  delay(1000);
}

void loop() {
  int sensorValue = analogRead(A0);
   // Clears the trigPin condition
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)
  // Displays the distance on the Serial Monitor
 if (distance >= 12) {
// Buzz
digitalWrite(buzzPin, HIGH);
} else {
// Don't buzz
digitalWrite(buzzPin, LOW);
}
// Firebase Error Handling And Writing Data At Specifed Path********
```

```
if (Firebase.setInt(firebaseData, "/drip", distance)) {     // On successful Write operation, function returns 1
            Serial.println(" Ultrasonic Value Uploaded Successfully");
            Serial.print("Distance = ");
            Serial.println(distance);
            Serial.println("\n");


            delay(1000);

    }

else {
   Serial.println(firebaseData.errorReason());
  }
  if (Firebase.setInt(firebaseData, "/heartrate", sensorValue)) {     // On successful Write operation, function returns 1
            Serial.println(" Ldr Value Uploaded Successfully");
            Serial.print("HEART RATE = ");
            Serial.println(sensorValue);
            Serial.println("\n");


            delay(1000);

    }

else {
   Serial.println(firebaseData.errorReason());
  }

}
```

MAIN ACTIVITY:

```
package com.undamped.medicare;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;
import android.util.Log;
import android.widget.TextView;
import android.widget.Toast;

import java.util.Calendar;
import java.util.Collections;
import java.util.List;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class MainActivity extends AppCompatActivity {

    private RecyclerView recyclerView;
    private TextView drip_text;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        recyclerView = findViewById(R.id.recyclerView);
        recyclerView.setHasFixedSize(true);
```

```java
recyclerView.setLayoutManager(new LinearLayoutManager(this));

drip_text = findViewById(R.id.drip_text);

Retrofit retrofit = new Retrofit.Builder()
        .baseUrl("https://api.thingspeak.com/channels/1352492/")
        .addConverterFactory(GsonConverterFactory.create())
        .build();

JSONPlaceHolder jsonPlaceholder = retrofit.create(JSONPlaceHolder.class);
Call<Feeds> call = jsonPlaceholder.getFeeds();
call.enqueue(new Callback<Feeds>() {
    @Override
    public void onResponse(Call<Feeds> call, Response<Feeds> response) {
        if (!response.isSuccessful()) {
            Log.e("Info", String.valueOf(response.code()));
            return;
        }

        List<Feeds> feeds = response.body().getFeeds();
        Log.e("Info", feeds.get(0).getCreated_at());
        Collections.reverse(feeds);
        checkDripActivity(feeds);
        FieldAdapter fieldAdapter = new FieldAdapter(feeds);
        recyclerView.setAdapter(fieldAdapter);
    }

    @Override
    public void onFailure(Call<Feeds> call, Throwable t) {
        Toast.makeText(MainActivity.this,                                    t.getMessage(),
Toast.LENGTH_SHORT).show();
        Log.e("Error", t.getMessage());
    }
});
```

```java
    }

    private void checkDripActivity(List<Feeds> feeds) {
        boolean stopped = false;

        int firstMin = Integer.parseInt(feeds.get(0).getCreated_at().substring(14,16));
        int secondMin = Integer.parseInt(feeds.get(1).getCreated_at().substring(14,16));
        int firstHour = Integer.parseInt(feeds.get(0).getCreated_at().substring(11,13));
        int secondHour = Integer.parseInt(feeds.get(1).getCreated_at().substring(11,13));

        Calendar c1 = Calendar.getInstance();
        Calendar c2 = Calendar.getInstance();
        c1.set(Calendar.HOUR_OF_DAY, firstHour);
        c1.set(Calendar.MINUTE, firstMin);
        c2.set(Calendar.HOUR_OF_DAY, secondHour);
        c2.set(Calendar.MINUTE, secondMin);

        if((feeds.get(0).getField2() == feeds.get(1).getField2()) && (c1.getTimeInMillis() -
c2.getTimeInMillis()) > 300000)
            stopped = true;

        if (stopped)
            drip_text.setText("Drip has Stopped");
        else
            drip_text.setText("Drip is Flowing");
    }
}
```

LOGIN MODULE:

```
package com.undamped.medicare;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.TextView;

import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

import butterknife.BindView;
import butterknife.ButterKnife;

public class LoginActivity extends AppCompatActivity {

    @BindView(R.id.emailEditText)
    EditText emailEditText;
    @BindView(R.id.passwordEditText) EditText passwordEditText;
    @BindView(R.id.loginButton)
    Button loginButton;
    @BindView(R.id.clickRegister)
    TextView clickRegister;
    @BindView(R.id.loginProgressBar)
    ProgressBar loginProgressBar;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        ButterKnife.bind(this);

        FirebaseAuth mAuth = FirebaseAuth.getInstance();

        clickRegister.setOnClickListener(view -> {
            startActivity(new Intent(LoginActivity.this, RegisterActivity.class));
            finish();
        });

        loginButton.setOnClickListener(view -> {
            startActivity(new Intent(LoginActivity.this, ValuesActivity.class));
            finish();
        });
    }
}
```

VALUES ACTIVITY:

```
package com.undamped.medicare;

import java.util.List;

public class Feeds {
    private List<Feeds> feeds;

    private int  field1,field2;
    private String created_at;
    private int entry_id;
```

```
public Feeds(String  created_at, int  field1, int field2, int entry_id) {
    this.created_at = created_at;
    this.field1 = field1;
    this.field2 = field2;
    this.entry_id = entry_id;
}

public List<Feeds> getFeeds()
{
    return feeds;
}

public void setFeeds(List<Feeds> feeds)
{
    this.feeds = feeds;
}

public String getCreated_at()
{
    return created_at;
}

public void setCreated_at(String created_at)
{
    this.created_at = created_at;
}

public int  getField1()
{
    return field1;
}

public void setField1(int field1)
{
```

```
        this.field1 = field1;
    }
    public int getField2()
    {
        return field2;
    }

    public void setField2(int  field2)
    {
        this.field2 = field2;
    }

    public int getEntry_id()
    {
        return entry_id;
    }

    public void setEntry_id(int entry_id)
    {
        this.entry_id = entry_id;
    }
}
```

REGISTER ACTIVITY:

```
package com.undamped.medicare;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
```

```
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.auth.FirebaseAuth;

import butterknife.BindView;
import butterknife.ButterKnife;

public class RegisterActivity extends AppCompatActivity {

    @BindView(R.id.emailREditText)
    EditText emailREditText;
    @BindView(R.id.passwordREditText) EditText passwordREditText;
    @BindView(R.id.confirmPasswordEditText) EditText confirmPasswordEditText;
    @BindView(R.id.registerBtn)
    Button registerBtn;
    @BindView(R.id.regProgressBar)
    ProgressBar regProgressBar;
    @BindView(R.id.clickLogin)
    TextView clickLogin;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        ButterKnife.bind(this);

        clickLogin.setOnClickListener(view -> {
            startActivity(new Intent(RegisterActivity.this, LoginActivity.class));
            finish();
        });
```

```
        registerBtn.setOnClickListener(v -> {
            String email = emailREditText.getText().toString();
            String password = passwordREditText.getText().toString();
            String confPassword = confirmPasswordEditText.getText().toString();

            if (email.isEmpty() || password.isEmpty() || confPassword.isEmpty())
                Snackbar.make(v, "Please fill all the fields", Snackbar.LENGTH_LONG).show();
            else if (password.length() < 8)
                Snackbar.make(v, "Password should contain at least 8 characters", Snackbar.LENGTH_LONG).show();
            else if (!confPassword.equals(password))
                Snackbar.make(v, "Passwords don't match", Snackbar.LENGTH_LONG).show();
            else {
                FirebaseAuth firebaseAuth = FirebaseAuth.getInstance();
                regProgressBar.setVisibility(View.VISIBLE);
                firebaseAuth.createUserWithEmailAndPassword(email, password)
                        .addOnCompleteListener(RegisterActivity.this, task -> {
                            if (task.isSuccessful()) {
                                Toast.makeText(RegisterActivity.this, "Registration successful. Logging in", Toast.LENGTH_LONG).show();
                                startActivity(new Intent(RegisterActivity.this, ValuesActivity.class));
                                finish();
                            } else {
                                regProgressBar.setVisibility(View.INVISIBLE);
                                Toast.makeText(RegisterActivity.this, "Registration failed. Try again", Toast.LENGTH_LONG).show();
                                Log.e("Info", task.getException().getMessage());
                            }
                        });
            }
        });
```

```
    }
}
```

FEEDS:

```
package com.undamped.medicare;

import java.util.List;

public class Feeds {
    private List<Feeds> feeds;

    private int  field1,field2;
    private String created_at;
    private int entry_id;



    public Feeds(String  created_at, int  field1, int field2, int entry_id) {
        this.created_at = created_at;
        this.field1 = field1;
        this.field2 = field2;
        this.entry_id = entry_id;
    }

    public List<Feeds> getFeeds()
    {
        return feeds;
    }

    public void setFeeds(List<Feeds> feeds)
    {
        this.feeds = feeds;
    }
```

```
public String getCreated_at()
{
    return created_at;
}

public void setCreated_at(String created_at)
{
    this.created_at = created_at;
}

public int  getField1()
{
    return field1;
}

public void setField1(int field1)
{
    this.field1 = field1;
}
public int getField2()
{
    return field2;
}

public void setField2(int  field2)
{
    this.field2 = field2;
}

public int getEntry_id()
{
    return entry_id;
}
```

```
    public void setEntry_id(int entry_id)
    {
        this.entry_id = entry_id;
    }
}
```

FIELD ADAPTER:

```
package com.undamped.medicare;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.util.List;

public class FieldAdapter extends RecyclerView.Adapter<FieldAdapter.ViewHolder> {
    List<Feeds> feedsList;
    Context context;

    public FieldAdapter(List<Feeds> feeds) {
        feedsList = feeds;

    }

    @NonNull
    @Override
```

```
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType)
{
        context = parent.getContext();
        View view = LayoutInflater.from(context).inflate(R.layout.list_element, parent,
false);
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
        Feeds feed = feedsList.get(position);

        holder.textDateTime.setText(feed.getCreated_at());

        if ((feed.getField1()) >= 800) {
            holder.bubble_value.setText("No Bubble");
        } else {
            holder.bubble_value.setText("Bubbles Formed");
        }

        switch (feed.getField2()) {
            case 12:
                holder.depth_value.setText("100%");
                break;
            case 11:
                holder.depth_value.setText("91.3%");
                break;
            case 10:
                holder.depth_value.setText("83%");
                break;
            case 9:
                holder.depth_value.setText("74.7%");
                break;
            case 8:
```

```
            holder.depth_value.setText("66.4%");
            break;
        case 7:
            holder.depth_value.setText("58.1%");
            break;
        case 6:
            holder.depth_value.setText("49.8%");
            break;
        case 5:
            holder.depth_value.setText("41.5%");
            break;
        case 4:
            holder.depth_value.setText("33.2%");
            break;
        case 3:
            holder.depth_value.setText("24.9%");
            break;
        case 2:
            holder.depth_value.setText("16.6%");
            break;
        case 1:
            holder.depth_value.setText("8.3%");
            break;
        case 0:
            holder.depth_value.setText("0%");
            break;
    }
}

@Override
public int getItemCount() {
    return feedsList.size();
}
```

```
    public class ViewHolder extends RecyclerView.ViewHolder {

        TextView depth_value, textDateTime, bubble_value;
        ImageView icon_image_view, icon_image_view_bubble;

        public ViewHolder(@NonNull View itemView) {
            super(itemView);

            bubble_value = itemView.findViewById(R.id.bubble_value);
            depth_value = itemView.findViewById(R.id.depth_value);
            textDateTime = itemView.findViewById(R.id.textDateTime);
        }
    }
}
```