**Project Tittle:**

Smart Medicine Reminder and Caretaker Alert System.

**Project Description:**

This project is designed to help patients take medicines on time and alert caretakers if a dose is missed. It uses an Arduino UNO with an RTC DS3231 module to schedule reminders. A buzzer and LEDs alert the user at the prescribed times, while a push button allows confirmation of medicine intake. If the user doesn’t confirm, the system signals a NodeMCU ESP8266 which sends a notification via the Blynk IoT app to the caretaker.

Additionally, an LCD displays real-time information and refill alerts to ensure the medicine supply is always monitored.

**Components Used:**

1.Arduino UNO

2.RTC DS3231 Real Time Clock (for accurate timing)

3.16×2 LCD with I2C module (to display reminders)

4.Push Button (to confirm medicine taken)

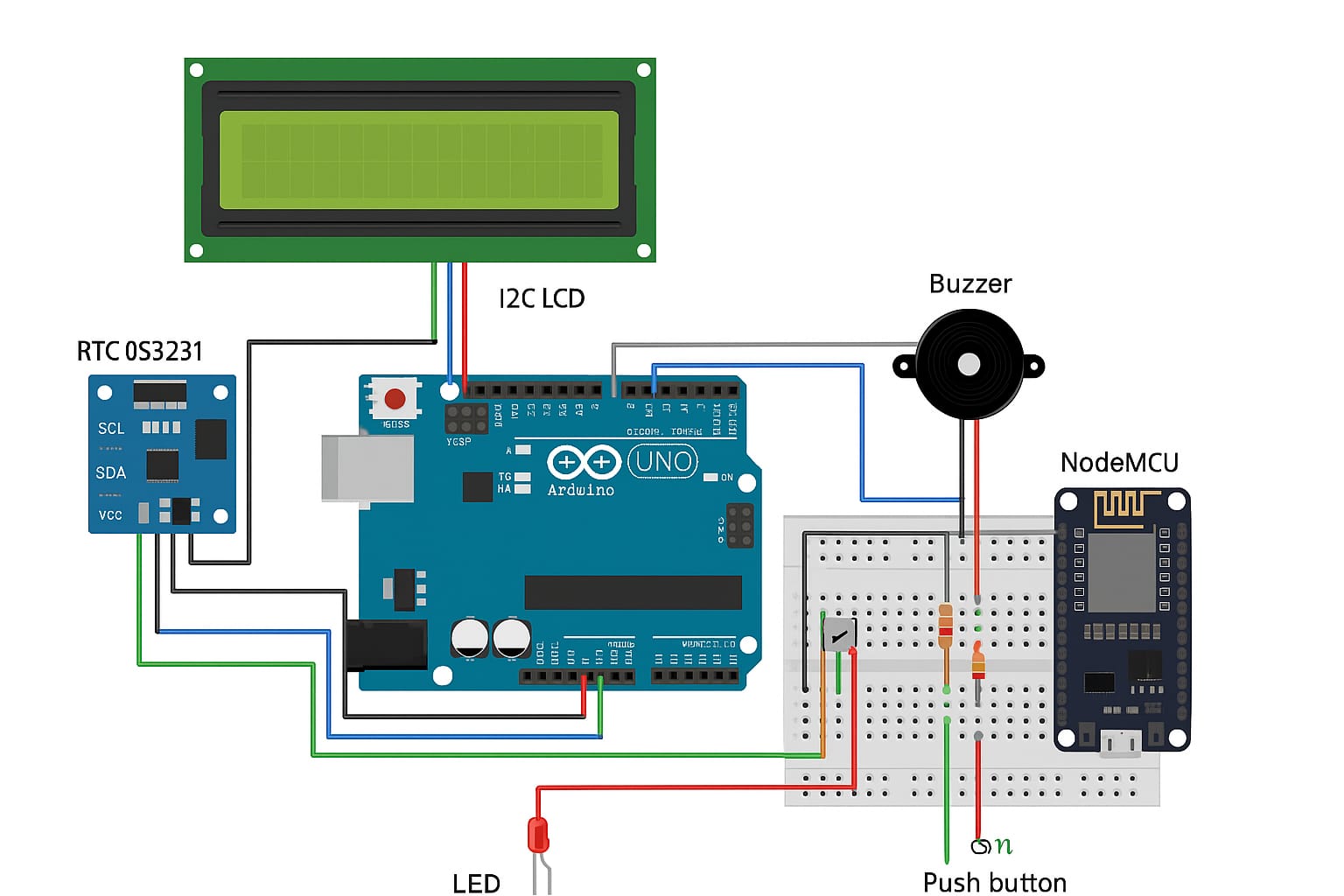
5.Buzzer (audio alert)

6.Two LEDs (visual indicators for reminders)

7.NodeMCU ESP8266 Wi-Fi Module (to send Blynk notifications)

8.Breadboard & Jumper Wires.

**Circuit Diagram:**



**Connection table:**

Arduino UNO pins → Component

A4 (SDA) → SDA on DS3231 module and SDA on I2C LCD

A5 (SCL) → SCL on DS3231 module and SCL on I2C LCD

5V → VCC on DS3231, VCC on I2C LCD, VCC of buzzer if it’s 5V type (or use separate supply)

GND → GND on DS3231, GND on I2C LCD, GND on buzzer, GND on NodeMCU (common ground required)

D8 → Buzzer (negative to GND). If passive buzzer, use a transistor (NPN) with base resistor; for active buzzer you can connect directly (through small resistor) but check buzzer current.

D9 → LED1 → through 220Ω resistor → GND

D10 → LED2 → through 220Ω resistor → GND

D2 → Push button (see button wiring options below)

D7 → Signal to NodeMCU input pin (used to tell NodeMCU “missed dose”) — use a level shifter or voltage divider (Arduino 5V → NodeMCU 3.3V)

Push-button wiring (two safe options)

Option A — use Arduino internal pull-up (recommended, no external resistor):

One side of button → D2

Other side of button → GND

In code: pinMode(2, INPUT\_PULLUP);

Button pressed = LOW (active low). Adjust your code accordingly.

Option B — external pull-down (if you prefer active-high):

One side of button → D2

Other side of button → 5V

10kΩ resistor between D2 and GND (pull-down)

In code: pinMode(2, INPUT); and read HIGH when pressed.

NodeMCU connection:

Arduino UNO outputs are 5V. NodeMCU pins require 3.3V max.

Do not connect Arduino D7 directly to a NodeMCU input. Use one of these:

Simple voltage divider (works for digital signals):

From Arduino D7 → R1 (20kΩ) → NodeMCU input pin

NodeMCU input pin → R2 (10kΩ) → GND

This will reduce 5V → ~3.3V.

OR use a proper logic level shifter or a MOSFET-based bi-directional level shifter.

Also connect GND (Arduino) ↔ GND (NodeMCU) (common ground).

Connect the NodeMCU input pin to an available digital pin like D2 or D5 on the NodeMCU (just pick one and update your NodeMCU code to listen on that pin).

**Short description:**

The Arduino UNO connects to the RTC DS3231, LCD via I2C, buzzer, LEDs, and a push button.

Pin 7 of the Arduino sends a HIGH signal to the NodeMCU ESP8266 if the medicine is missed.

The NodeMCU, connected to Wi-Fi, sends a notification to the caretaker’s Blynk app.

**Working Principle:**

At the scheduled time, the Arduino reads the time from the RTC DS3231.

It activates the buzzer and LEDs, displaying the message on the LCD.

The patient presses the push button to confirm they took the medicine.

If the button is not pressed within 30 seconds, the Arduino sets pin 7 HIGH.

The NodeMCU ESP8266 detects the signal on its input pin and sends a notification to the caretaker via Blynk IoT.

A refill alert is also displayed on the LCD when medicine stock is low.

**Arduino UNO Code:**

#include <Wire.h>

#include <RTClib.h>

#include <LiquidCrystal\_I2C.h>

// RTC & LCD

RTC\_DS3231 rtc;

LiquidCrystal\_I2C lcd(0x27, 16, 2);

// Pin Definitions

const int buzzer = 8;

const int led1 = 9;

const int led2 = 10;

const int buttonPin = 6;

const int missedSignalPin = 7;   // To ESP D7

const int refillSignalPin = 5;   // To ESP D5

 // Alarm Times

const int alarm1Hour = 10;

const int alarm1Minute =39;

const int alarm2Hour = 10;

const int alarm2Minute =00;

// Medicine Tracking

int medicineCount = 3;

const int refillThreshold = 2;

void setup() {

  Wire.begin();

  lcd.begin(16, 2);

  lcd.backlight();

if (!rtc.begin()) {

    lcd.print("RTC not found!");

    while (1);

  }

// Uncomment once to set the date/time

 // rtc.adjust(DateTime(2025, 8, 23, 10,30 , 0));

pinMode(buzzer, OUTPUT);

  pinMode(led1, OUTPUT);

  pinMode(led2, OUTPUT);

  pinMode(buttonPin, INPUT\_PULLUP);

  pinMode(missedSignalPin, OUTPUT);

  pinMode(refillSignalPin, OUTPUT);

digitalWrite(missedSignalPin, LOW);

  digitalWrite(refillSignalPin, LOW);

lcd.setCursor(0, 0);

  lcd.print("Date & Time:");

  delay(2000);

  lcd.clear();

}

void loop() {

  DateTime now = rtc.now();

// Show Date

  lcd.setCursor(0, 0);

  lcd.print("Date: ");

  if (now.day() < 10) lcd.print("0");

  lcd.print(now.day()); lcd.print("/");

  if (now.month() < 10) lcd.print("0");

  lcd.print(now.month()); lcd.print("/");

  lcd.print(now.year());

// Show Time

  lcd.setCursor(0, 1);

  lcd.print("Time: ");

  if (now.hour() < 10) lcd.print("0");

  lcd.print(now.hour()); lcd.print(":");

  if (now.minute() < 10) lcd.print("0");

  lcd.print(now.minute()); lcd.print(":");

  if (now.second() < 10) lcd.print("0");

  lcd.print(now.second());

  // --- Alarm 1 ---

  if (now.hour() == alarm1Hour && now.minute() == alarm1Minute && now.second() == 0) {

    triggerAlarm(led1);

  }

  // --- Alarm 2 ---

  if (now.hour() == alarm2Hour && now.minute() == alarm2Minute && now.second() == 0) {

    triggerAlarm(led2);

  }

  delay(1000); // Refresh every second

}

void triggerAlarm(int ledPin) {

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("Take Medicine!");

  digitalWrite(ledPin, HIGH);

  tone(buzzer, 1000);

  bool taken = false;

  unsigned long startTime = millis();

  // Wait 30 seconds for button press

  while (millis() - startTime < 30000) {

    if (digitalRead(buttonPin) == LOW) {

      taken = true;

      break;

    }

  }

digitalWrite(ledPin, LOW);

  noTone(buzzer);

  if (taken) {

    medicineCount--;

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Taken. Left: ");

    lcd.print(medicineCount);

    delay(2000);

    // Refill check

    if (medicineCount <= refillThreshold) {

      lcd.clear();

      lcd.setCursor(0, 0);

      lcd.print("Refill Needed!");

      digitalWrite(refillSignalPin, HIGH);

      delay(1000);

      digitalWrite(refillSignalPin, LOW);

    }

  } else {

    // Missed medicine alert

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Missed Dose!");

    digitalWrite(missedSignalPin, HIGH);

    delay(1000);

    digitalWrite(missedSignalPin, LOW);

  }

  delay(1000); // small buffer

}

**NodeMCU ESP8266 Code:**

#define BLYNK\_TEMPLATE\_ID "TMPL3UvNBglWv"

#define BLYNK\_TEMPLATE\_NAME "Medicine Reminder"

#define BLYNK\_AUTH\_TOKEN "MIJaFl-o-OroeckOmQH1QSV8Qa89JCFE"

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

// WiFi credentials

char ssid[] = "FTTH-C1EC";

char pass[] = "12345678";

int espInputPin = D1;     // From Arduino when missed dose

void setup() {

  Serial.begin(9600);

  Blynk.begin(BLYNK\_AUTH\_TOKEN, ssid, pass);

  pinMode(espInputPin, INPUT);

}

void loop() {

  Blynk.run();

// Missed medicine

  if (digitalRead(espInputPin) == HIGH) {

    Blynk.logEvent("missed\_\_medicine");

    delay(1000);  // debounce

  }

}

**How to Upload & Test:**

1. Arduino UNO:

Connect Arduino to your PC via USB.

Open Arduino IDE.

Paste the Arduino sketch and upload.

2. NodeMCU ESP8266:

Install ESP8266 board package in Arduino IDE.

Select NodeMCU ESP8266 from “Tools > Board”.

Paste the NodeMCU sketch and upload.

3. Blynk Setup:

Download the Blynk app.

Create a project and note your Auth Token.

Add a notification widget.

Enter your Wi-Fi credentials and Auth Token in the NodeMCU code.

4. Test:

Power on Arduino and NodeMCU.

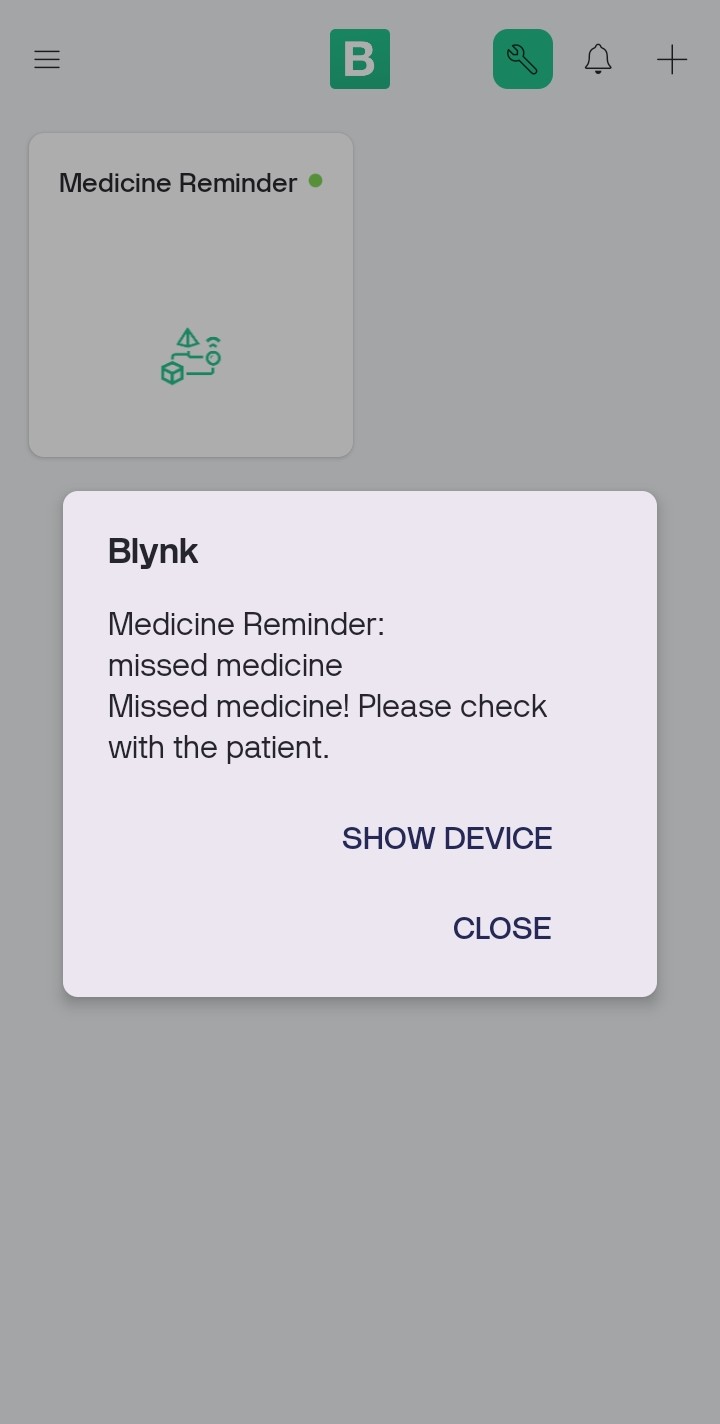
Wait for scheduled time.

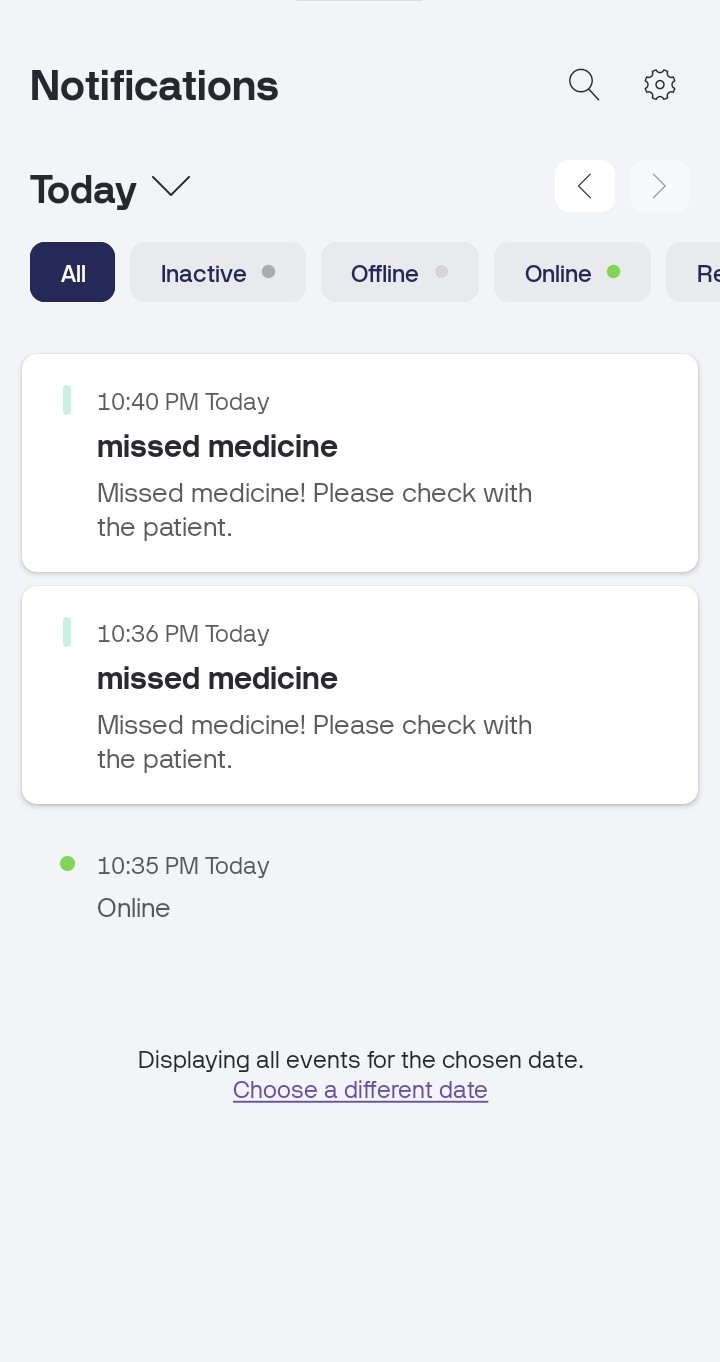
Check buzzer/LED alert and LCD message.

If button not pressed, check caretaker phone for Blynk notification.

**Project Images:**



****

****

**Conclusion:**

The Smart Medicine Reminder and Dispenser successfully reminds patients to take their medicines on time and alerts caretakers if a dose is missed.

It also displays a refill alert on the LCD and sends notifications through Blynk.

This project demonstrates how Arduino + IoT integration can improve healthcare and patient safety.