Final Code

#include <SimpleDHT.h>  // temp sensor

#include <Wire.h>

#include <RTClib.h>  // Library for RTC

#include <Servo.h>   // Library for Servo

int trigpin = 13;  // ultrasonice output

int echopin = 11;  // ultrasonic input

float distance;    // ultrasonice

float duration;    //ultrasonic

DateTime now;

String dayOfWeek[]= {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};

RTC\_DS3231 rtc;  // Create RTC object

Servo myServo;   // Create a Servo object

// nahid

//      for DHT11,

//      VCC: 5V or 3V

//      GND: GND

//      DATA: 2

// Pin connected to the DHT11 sensor

int pinDHT11 = 2;

SimpleDHT11 dht11(pinDHT11);

void setup() {

  pinMode(trigpin, OUTPUT);  // this will emit the signal  // ultra sonic

  pinMode(echopin, INPUT);   // this will receive the returned signal  // ultrasonic

  Serial.begin(9600);  // Initialize Serial Monitor

  rtc.begin();

  myServo.attach(10);  // Attach servo to pin 10

  // Initialize RTC

  rtc.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_)));  // Set RTC to compile time

  if (!rtc.begin()) {

    Serial.println("Couldn't find RTC");

    while (1)

      ;  // Stop execution if RTC not found

  }

  if (rtc.lostPower()) {

    Serial.println("RTC lost power, setting default time.");

    rtc.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_)));  // Set RTC to compile time

  }

  myServo.write(0);  // Initialize servo at 0 degrees

}

void UltraSonic() {

  digitalWrite(trigpin, LOW);

  delay(2);  // in mil sec

  digitalWrite(trigpin, HIGH);

  delayMicroseconds(10);

  digitalWrite(trigpin, LOW);         // generated a signal and stopped it later

  duration = pulseIn(echopin, HIGH);  // calculates for how long the echopin was high. Micro sec

  distance = duration \* 0.034 / 2;    // in cm

  if (distance > 0 && distance < 40000) {

    Serial.print("Distance: ");

    Serial.print(distance);

    Serial.println();

    if (distance <= 2) {

      Serial.println("Notification: Storage Full");

    } else if (distance > 2 && distance <= 5) {

      Serial.println("Notification : Storage OK");

    } else if (distance > 5 && distance <= 8) {

      Serial.println("Notification: Storage LOW");

    } else {

      Serial.println("Notification: storage Empty");

    }

  }

}

void temp\_humidity() {

  // start working...

  Serial.println("\*========NOTIFICATIONS========\*");

  // read without samples

  byte temperature = 0;

  byte humidity = 0;

  // Read DHT11 data

  int err = SimpleDHTErrSuccess;

  if ((err = dht11.read(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess) {

    Serial.print("Read DHT11 failed, err=");

    Serial.print(SimpleDHTErrCode(err));

    Serial.print(",");

    Serial.println(SimpleDHTErrDuration(err));

    delay(1000);

    return;

  }

  // Display temperature and humidity data

  Serial.print("Temperature: ");

  Serial.print(temperature);

  Serial.println(" °C");

  Serial.print("Humidity: ");

  Serial.print(humidity);

  Serial.println(" %");

  // Determine health condition

  if (temperature < 20 || temperature > 30) {

    Serial.println("⚠️ Temperature is out of safe range!");

  }

  if (humidity < 40 || humidity > 60) {

    Serial.println("⚠️ Humidity is out of safe range!");

  }

  if (temperature >= 20 && temperature <= 30 && humidity >= 40 && humidity <= 60) {

    Serial.println("✅ Environment is healthy for your pet!");

  }

  delay(1500);  // Update data every 5 seconds

}

void printDateTime(DateTime now) {

    Serial.print("Date: ");

    Serial.print(now.day());

    Serial.print("/");

    Serial.print(now.month());

    Serial.print("/");

    Serial.print(now.year());

    Serial.print("       ");

    Serial.print("Time: ");

    Serial.print(now.hour());

    Serial.print(":");

    Serial.print(now.minute());

    Serial.print(":");

    Serial.print(now.second());

    Serial.print("       ");

    Serial.println(dayOfWeek[now.dayOfTheWeek()]);

    delay(1000);

}

void rtcAndServo() {

   now = rtc.now();  // Get current time from RTC

  // Servo activation conditions

  printDateTime(now) ;

  if (now.hour() == 0 && now.minute() == 3) {

    //if(isOverweight() == false) // kuttay khaise

    //{

    myServo.write(90);

    //while(isOverweight() == false)

    //{

    delay(500);

    // }

    myServo.write(0);

  }

  else if (now.hour() == 0 && now.minute() == 4) {

    //if(isOverweight() == false) // kuttay khaise

    //{

    myServo.write(90);

    //while(isOverweight() == false){

    delay(500);

    //}

    myServo.write(0);

    //}

  } else if (now.hour() == 0 && now.minute() == 5) {

    //if(isOverweight() == false) // kuttay khaise

    //{

    myServo.write(90);

    //while(isOverweight() == false){

    delay(500);

    //}

    myServo.write(0);

  }

}

void loop() {

  // manually getting in to low state to avoid problems

  UltraSonic();

  delay(1500);  // not overwhelming the sensor with constant checking.

  temp\_humidity();

  delay(1500);

  myServo.write(0);

  rtcAndServo();

  delay(60000);

}