IOT HOLIDAY ASSIGNMENT

A.VINEESHCHOWDARY

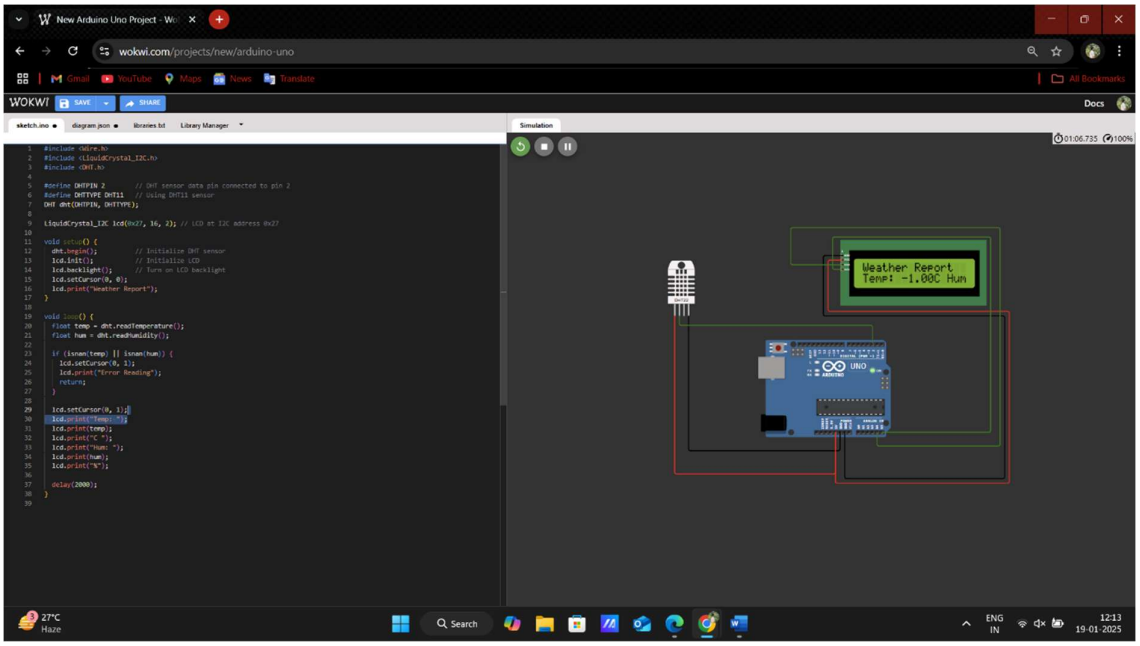
2211CS020004 AIML-ALPHA

1) Write a Embedded C Program to Create a Weather Reporting System that provides real- time environmental data to users.

A)

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>

#define DHTPIN 2        // DHT sensor data pin connected to pin 2
#define DHTTYPE DHT11   // Using DHT11 sensor
DHT dht(DHTPIN, DHTTYPE);

LiquidCrystal_I2C lcd(0x27, 16, 2); // LCD at I2C address 0x27

void setup() {
  dht.begin();          // Initialize DHT sensor
  lcd.init();           // Initialize LCD
  lcd.backlight();      // Turn on LCD backlight
  lcd.setCursor(0, 0);
  lcd.print("Weather Report");
}

void loop() {
  float temp = dht.readTemperature();
```

```
  float hum = dht.readHumidity();

  if (isnan(temp) || isnan(hum)) {
    lcd.setCursor(0, 1);
    lcd.print("Error Reading");
    return;
  }

  lcd.setCursor(0, 1);
  lcd.print("Temp: ");
  lcd.print(temp);
  lcd.print("C ");
  lcd.print("Hum: ");
  lcd.print(hum);
  lcd.print("%");

  delay(2000);
}
```



2) Write a Embedded C Program to Create a Home Automation System that simplifies daily routines (Any 2 Devices) by controlling devices remotely

A) #define LED1 2

#define LED2 3

void setup() {

// Initialize the LEDs as outputs

pinMode(LED1, OUTPUT);

pinMode(LED2, OUTPUT);

// Start serial communication

Serial.begin(9600);

Serial.println("Home Automation System");

Serial.println("Commands: ");

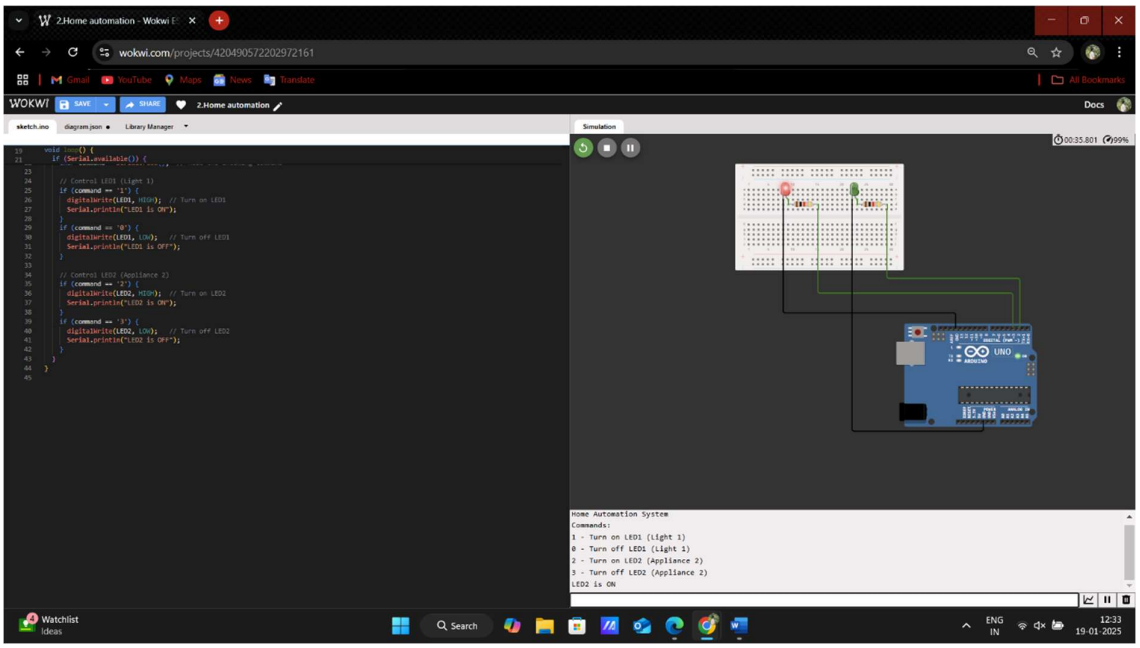
Serial.println("1 - Turn on LED1 (Light 1)");

Serial.println("0 - Turn off LED1 (Light 1)");

Serial.println("2 - Turn on LED2 (Appliance 2)");

```
  Serial.println("3 - Turn off LED2 (Appliance 2)");
}

void loop() {
  // Check if data is available on Serial
  if (Serial.available()) {
    char command = Serial.read();  // Read the incoming command

    // Control LED1 (Light 1)
    if (command == '1') {
      digitalWrite(LED1, HIGH);  // Turn on LED1
      Serial.println("LED1 is ON");
    }
    if (command == '0') {
      digitalWrite(LED1, LOW);   // Turn off LED1
      Serial.println("LED1 is OFF");
    }

    // Control LED2 (Appliance 2)
    if (command == '2') {
      digitalWrite(LED2, HIGH);  // Turn on LED2
      Serial.println("LED2 is ON");
    }
    if (command == '3') {
      digitalWrite(LED2, LOW);   // Turn off LED2
      Serial.println("LED2 is OFF");
```

```
    }
  }
}
```



3) Write a Embedded C Program to Create an Air Pollution Monitoring System that tracks air quality levels in real-time to ensure a healthier environment.

A) #include <Wire.h>

#include <Adafruit_SSD1306.h>

#include <Adafruit_GFX.h>

#define SSD1306_I2C_ADDRESS 0x3C // I2C address for OLED display

#define POT_PIN A0 // Analog pin for potentiometer

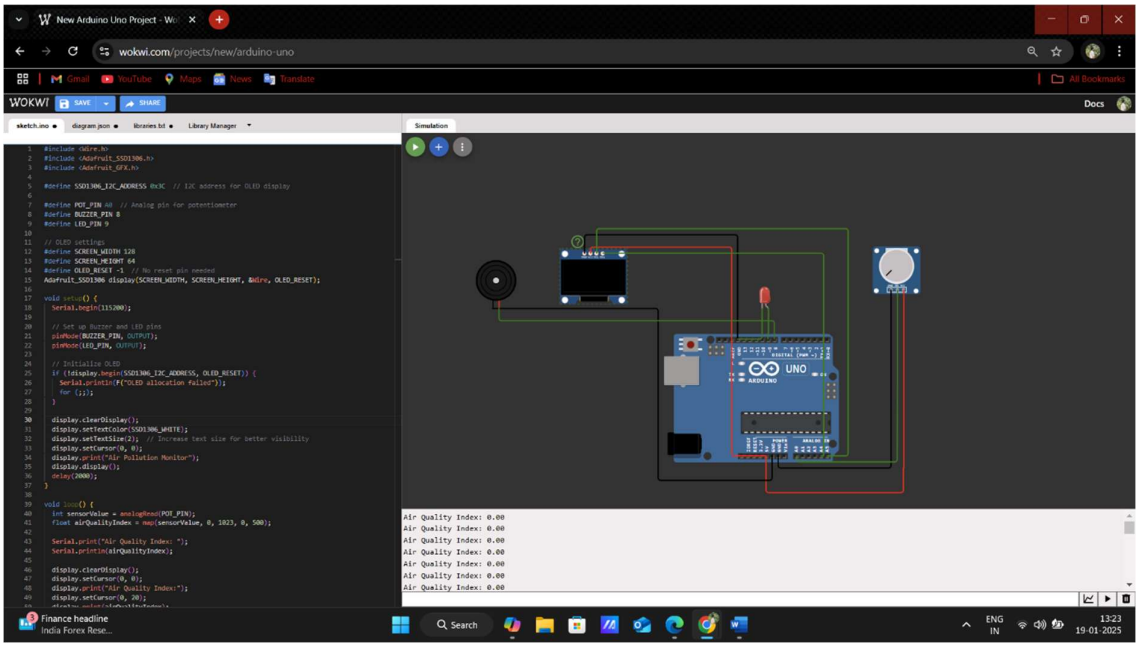
#define BUZZER_PIN 8

#define LED_PIN 9

```
// OLED settings
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1  // No reset pin needed
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

void setup() {
  Serial.begin(115200);

  // Set up Buzzer and LED pins
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(LED_PIN, OUTPUT);

  // Initialize OLED
  if (!display.begin(SSD1306_I2C_ADDRESS, OLED_RESET)) {
    Serial.println(F("OLED allocation failed"));
    for (;;);
  }

  display.clearDisplay();
  display.setTextColor(SSD1306_WHITE);
  display.setTextSize(2);  // Increase text size for better visibility
  display.setCursor(0, 0);
  display.print("Air Pollution Monitor");
  display.display();
```

```
  delay(2000);
}

void loop() {
  int sensorValue = analogRead(POT_PIN);
  float airQualityIndex = map(sensorValue, 0, 1023, 0, 500);

  Serial.print("Air Quality Index: ");
  Serial.println(airQualityIndex);

  display.clearDisplay();
  display.setCursor(0, 0);
  display.print("Air Quality Index:");
  display.setCursor(0, 20);
  display.print(airQualityIndex);
  display.print(" ppm");

  if (airQualityIndex > 300) {
    display.setCursor(0, 40);
    display.print("Warning: Poor Air Quality!");
    digitalWrite(BUZZER_PIN, HIGH);
    digitalWrite(LED_PIN, HIGH);
  } else {
    display.setCursor(0, 40);
    display.print("Air Quality is Good");
    digitalWrite(BUZZER_PIN, LOW);
```

```
  digitalWrite(LED_PIN, LOW);
 }

 display.display();
 delay(1000);
}
```



4) Write a Embedded C Program to Create an IoT-based Smart Irrigation System for Agriculture that automates watering based on weather and soil conditions

A)) #include <DHT.h>

// Pin Definitions

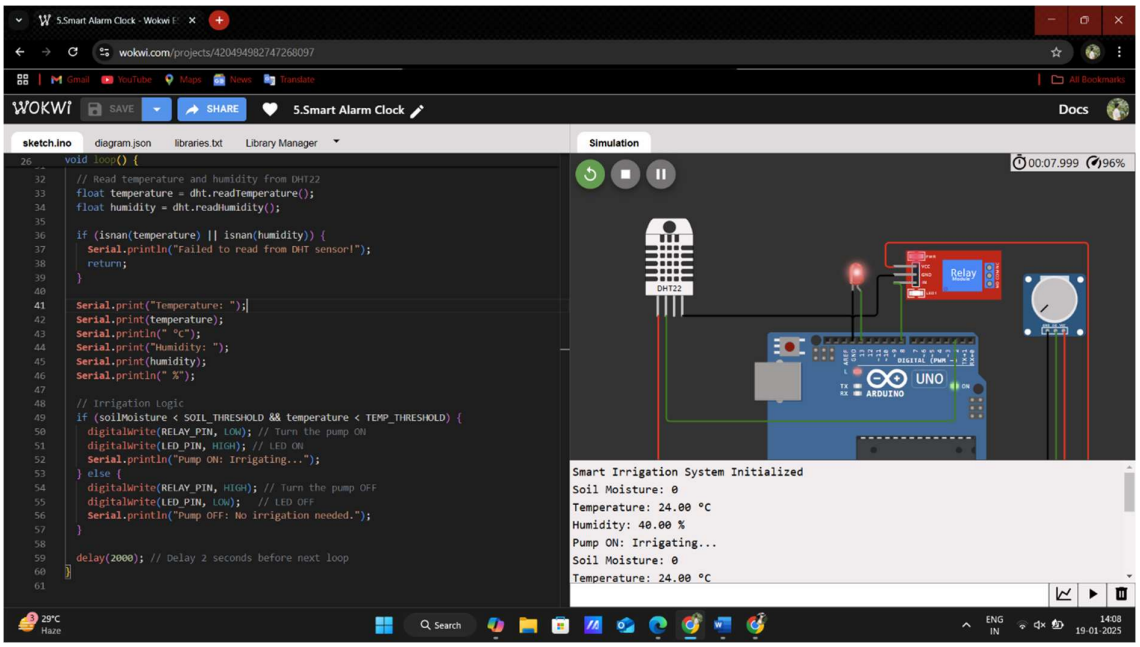
#define DHTPIN 2 // DHT22 data pin connected to Arduino pin 2

#define DHTTYPE DHT22 // Sensor type: DHT22

#define SOIL_PIN A0 // Potentiometer connected to A0

```
#define RELAY_PIN 8    // Relay module connected to pin 8
#define LED_PIN 13     // LED connected to pin 13 (optional)

// Threshold Values
#define SOIL_THRESHOLD 400  // Adjust to simulate soil dryness
#define TEMP_THRESHOLD 35   // Maximum temperature threshold

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  pinMode(RELAY_PIN, OUTPUT);
  pinMode(LED_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, HIGH); // Initially turn the relay OFF
  Serial.begin(9600);
  dht.begin();

  Serial.println("Smart Irrigation System Initialized");
}

void loop() {
  // Read soil moisture value from potentiometer
  int soilMoisture = analogRead(SOIL_PIN);
  Serial.print("Soil Moisture: ");
  Serial.println(soilMoisture);

  // Read temperature and humidity from DHT22
  float temperature = dht.readTemperature();
```

```
  float humidity = dht.readHumidity();

  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" °C");
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.println(" %");

  // Irrigation Logic
  if (soilMoisture < SOIL_THRESHOLD && temperature <
TEMP_THRESHOLD) {
    digitalWrite(RELAY_PIN, LOW); // Turn the pump ON
    digitalWrite(LED_PIN, HIGH); // LED ON
    Serial.println("Pump ON: Irrigating...");
  } else {
    digitalWrite(RELAY_PIN, HIGH); // Turn the pump OFF
    digitalWrite(LED_PIN, LOW);   // LED OFF
    Serial.println("Pump OFF: No irrigation needed.");
  }

  delay(2000); // Delay 2 seconds before next loop
}
```



5) Write a Embedded C Program to Create a Smart Alarm Clock that adjusts to your schedule and environment, waking you up intelligently.

A) {

"version": 1,

"author": "Vineesh Chowdary",

"editor": "wokwi",

"parts": [

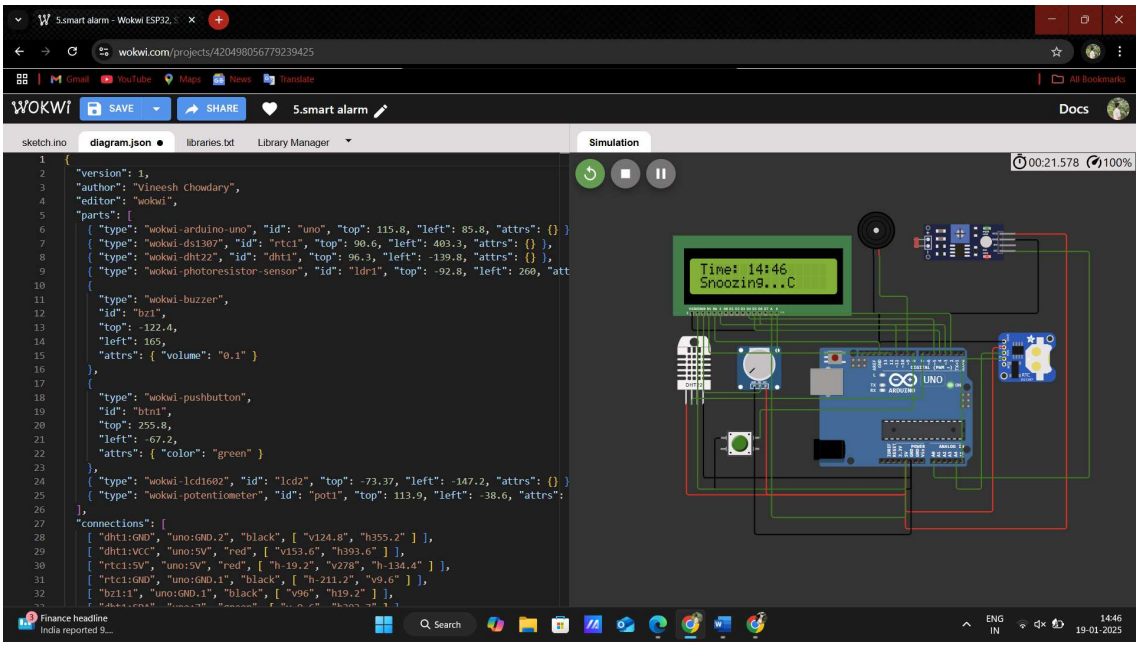
{ "type": "wokwi-arduino-uno", "id": "uno", "top": 115.8, "left": 85.8, "attrs": {} },

{ "type": "wokwi-ds1307", "id": "rtc1", "top": 90.6, "left": 403.3, "attrs": {} },

{ "type": "wokwi-dht22", "id": "dht1", "top": 96.3, "left": -139.8, "attrs": {} },

```
    { "type": "wokwi-photoresistor-sensor", "id": "ldr1", "top": -92.8, "left": 260,
"attrs": {} },
    {
      "type": "wokwi-buzzer",
      "id": "bz1",
      "top": -122.4,
      "left": 165,
      "attrs": { "volume": "0.1" }
    },
    {
      "type": "wokwi-pushbutton",
      "id": "btn1",
      "top": 255.8,
      "left": -67.2,
      "attrs": { "color": "green" }
    },
    { "type": "wokwi-lcd1602", "id": "lcd2", "top": -73.37, "left": -147.2, "attrs":
{} },
    { "type": "wokwi-potentiometer", "id": "pot1", "top": 113.9, "left": -38.6,
"attrs": {} }
  ],
  "connections": [
    [ "dht1:GND", "uno:GND.2", "black", [ "v124.8", "h355.2" ] ],
    [ "dht1:VCC", "uno:5V", "red", [ "v153.6", "h393.6" ] ],
    [ "rtc1:5V", "uno:5V", "red", [ "h-19.2", "v278", "h-134.4" ] ],
    [ "rtc1:GND", "uno:GND.1", "black", [ "h-211.2", "v9.6" ] ],
    [ "bz1:1", "uno:GND.1", "black", [ "v96", "h19.2" ] ],
    [ "dht1:SDA", "uno:7", "green", [ "v-9.6", "h393.7" ] ],
```

```
    [ "ldr1:VCC", "uno:5V", "red", [ "h86.4", "v499.2", "h-278.4" ] ],
    [ "ldr1:AO", "uno:A0", "green", [ "h124.8", "v383.3", "h-268.8" ] ],
    [ "ldr1:GND", "uno:GND.1", "black", [ "h38.4", "v124.4", "h-259.2" ] ],
    [ "bz1:2", "uno:9", "green", [ "v67.2", "h47.6" ] ],
    [ "btn1:1.r", "uno:8", "green", [ "v-48", "h269" ] ],
    [ "btn1:1.l", "uno:GND.2", "black", [ "h-9.6", "v86.4", "h336" ] ],
    [ "lcd2:VSS", "uno:GND.1", "black", [ "v28.8", "h288" ] ],
    [ "lcd2:VDD", "uno:5V", "green", [ "v297.6", "h28.9" ] ],
    [ "pot1:GND", "uno:GND.2", "black", [ "v249.6", "h230.4" ] ],
    [ "pot1:VCC", "uno:5V", "red", [ "v182.4", "h258.4" ] ],
    [ "lcd2:V0", "pot1:SIG", "green", [ "v134.4", "h95.7" ] ],
    [ "lcd2:RS", "uno:12", "green", [ "v0" ] ],
    [ "lcd2:RW", "uno:GND.1", "green", [ "v48", "h268.7" ] ],
    [ "lcd2:E", "uno:11", "green", [ "v19.2", "h211.2" ] ],
    [ "lcd2:D4", "uno:5", "green", [ "v9.6", "h326.4" ] ],
    [ "lcd2:D6", "uno:3", "green", [ "v19.2", "h317" ] ],
    [ "lcd2:D5", "uno:4", "green", [ "v28.8", "h307.3" ] ],
    [ "lcd2:D7", "uno:2", "green", [ "v0", "h288.3" ] ],
    [ "lcd2:A", "uno:5V", "green", [ "v345.6", "h134.3" ] ],
    [ "lcd2:K", "uno:GND.1", "green", [ "v0" ] ],
    [ "rtc1:SDA", "uno:A4", "green", [ "h-38.4", "v230.1", "h-48" ] ],
    [ "rtc1:SCL", "uno:A5", "green", [ "h0" ] ]
  ],
  "dependencies": {}
}
```


5.smart alarm - Wokwi ESP32,
wokwi.com/projects/420498056779239425
WOKWI
SAVE
SHARE
5.smart alarm
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:21.578
100%
Time: 14:46
Snoozing...C
UNO
Finance headline
India reported 9....
Search
ENG
IN
14:46
19-01-2025