College name: JP COLLEGE OF ENGINEERING
College code: 9512

Project ID :Proj_211934_Team_1

TEAM MEMBERS:

1. HEMAPRIYA.C (au951221106013)
2. MUTHURAMA.K (au951221106027)
3. MURUGESHWARI.N (au951221106026)
4. ANISHA PUSPHAM.A (au951221106004)
5. RAMYA.N (au951221106032)

Phase 4 : Development Part 2

Topic: Environmental Monitoring system

Front-End:

1. User Interface (UI): Create a user-friendly web interface using HTML, CSS, and JavaScript to display the real-time data. Consider using a framework like React or Angular for a more interactive UI.

2. Dashboard: Design a dashboard where users can view temperature and humidity data. You can use charts or graphs to present the information visually.

3. Real-time Updates: Implement WebSocket technology to enable real-time updates without the need for constant page refresh. Libraries like Socket.io can help with this.

Back-End:

4. IoT Devices: Set up IoT devices (e.g., sensors) to collect temperature and humidity data. These devices should send data to your platform periodically.

5. Data Collection: Create an API or server that can receive and process data from IoT devices. Use a suitable programming language like Node.js, Python, or Java.

6. Database: Store the received data in a database (e.g., MySQL, MongoDB) for historical records and analytics.

7. Authentication and Authorization: Implement user authentication to ensure that only authorized users can access the data.

8. APIs: Design RESTful or GraphQL APIs to handle data retrieval and ensure that your front-end can fetch data from the back-end.

9. Real-time Data Processing: Use a real-time data processing framework or message broker like Apache Kafka or MQTT to handle incoming data streams efficiently.

10. Data Presentation: Convert the received data into appropriate formats (e.g., JSON) and send it to the front-end for real-time display.

Security:

11. Security Measures: Implement security measures such as HTTPS for data transmission, secure authentication, and proper data encryption to protect user and device data.

Scalability:

12. Scalability: Ensure that your platform is scalable to accommodate a growing number of IoT devices and users. Consider containerization and cloud hosting for scalability.

Maintenance and Monitoring:

13. Monitoring: Set up monitoring tools to track the health and performance of your system and receive alerts for issues.

14. Maintenance: Regularly update and maintain your platform to ensure it continues to function smoothly.

Data Processing and Analytics:

15.The collected data is processed and analysed to derive valuable insights. Advanced analytics techniques may be used to detect trends, anomalies, and patterns within the data.

16.Creating a real-time Environment Management platform involves a combination of front end and back-end technologies. Here’s a simplified outline using C and C++ and python programming with Wi-Fi connection for the front end and Node.js for the back end

C ++ program:

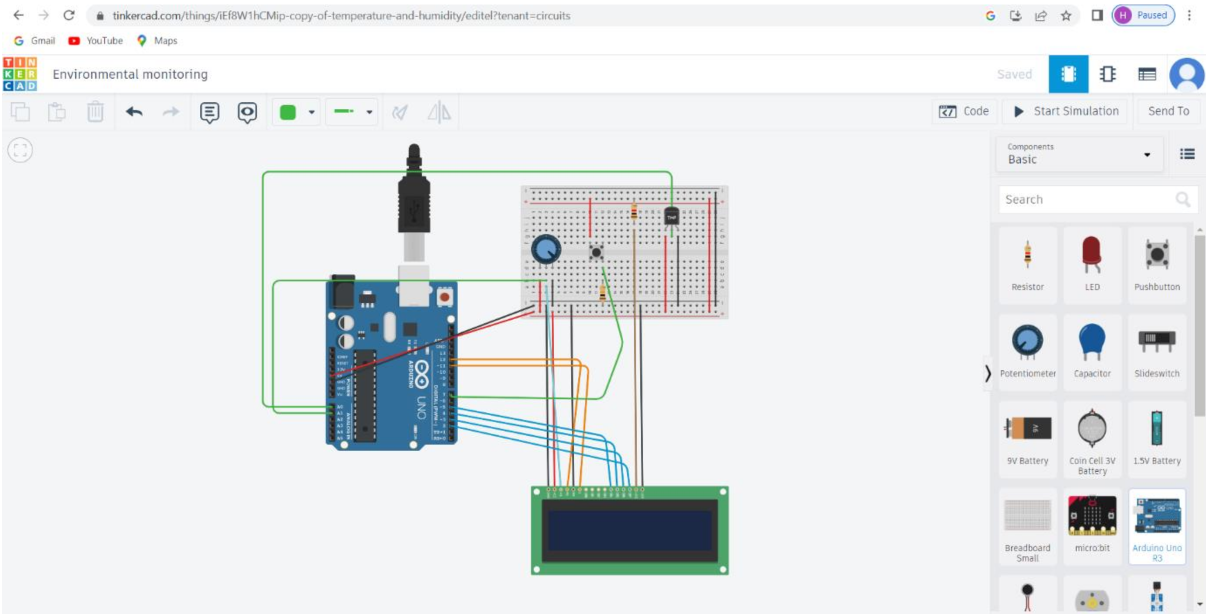
// include the library code:

#include <LiquidCrystal.h>

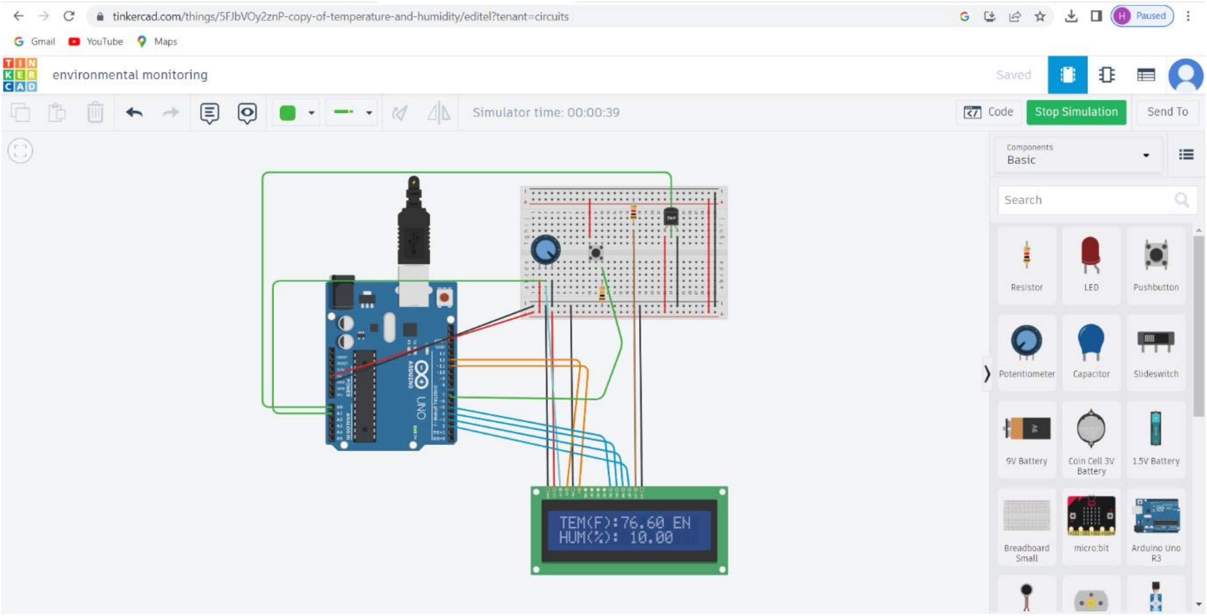
```
#include <string.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);


// Defining Variables
const int analogIn = A0;
int humiditysensorOutput = 0;
int RawValue= 0;
double Voltage = 0;
double tempC = 0;
double tempF = 0;
int buttonState = 0;
char LCD_LANGUAGE = 'E';

void setup()
{
		Serial.begin(9600);
		pinMode(A1, INPUT);
		pinMode(7, INPUT);
		lcd.begin(16, 2);
}

void loop()
{
  RawValue = analogRead(analogIn);
  Voltage = (RawValue / 1023.0) * 5000; // 5000 to get millivots.
  tempC = (Voltage-500) * 0.1; // 500 is the offset
```

```
tempF = (tempC * 1.8) + 32; // convert to F
double humidity = map(humiditysensorOutput, 0, 1023, 10, 70);
buttonState = digitalRead(7);

if (buttonState == HIGH) {
  LCD_LANGUAGE = 'T';
} else {
        LCD_LANGUAGE = 'E';
}

if (LCD_LANGUAGE == 'T') {
  lcd.setCursor(0,0);
  lcd.print("SIC(C):" + String(tempC,2) + " TR");
  humiditysensorOutput = analogRead(A1);
  lcd.setCursor(0,1);
  lcd.print("NEM(%): " + String(humidity,2));
}
else {
  lcd.setCursor(0,0);
  lcd.print("TEM(F):" + String(tempF,2) + " EN");
  humiditysensorOutput = analogRead(A1);
  lcd.setCursor(0,1);
  lcd.print("HUM(%): " + String(humidity,2));
}
}
```

DEVELOPED MODEL :



OUTPUT AFTER RECEIVING THE TEMPERATURE AND HUMIDITY LEVEL :



CONCLUSION :

In conclusion, an Environmental Monitoring System using the Internet of Things (IoT) represents a transformative and highly valuable technology for addressing a wide range of environmental challenges. This system harnesses the power of interconnected sensors, devices, and data analytics to collect, manage, and analyse environmental data in real-time.