Mini-Project Report On

Temperature Regulated Fan

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering

By

Abhinav R Nair (U2003004)
Abin Abraham Menacherry (U2003007)
Antony Francis (U2003041)
Ashok Gopal K.A. (U2003046)

Under the guidance of
Ms.Liya Joseph

Department of Computer Science & Engineering
Rajagiri School of Engineering and Technology (Autonomous)
Rajagiri Valley, Kakkanad, Kochi, 682039

July 2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJAGIRI SCHOOL OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)
RAJAGIRI VALLEY, KAKKANAD, KOCHI, 682039

CERTIFICATE

This is to certify that the mini-project report entitled ***"Temperature Regulated Fan"*** *is a bonafide work done by* ***Mr. Abhinav R Nair (U2003004), Mr. Abin Abraham Menacherry (U2003007), Mr. Antony Francis (U2003041), Mr. Ashok Gopal K.A. (U2003046))****, submitted to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2022-2023.*

Dr. Preetha K. G.	**Dr.Sminu Izudheen**	**Dr. Ms.Liya Josephl**
Head of Department	Mini-Project Coordinator	Mini-Project Guide
Dept. of CSE	Professor	Asst. Professor
RSET	Dept. of CSE	Dept. of CSE
	RSET	RSET

ACKNOWLEDGEMENTS

We wish to express our sincere gratitude towards **Dr. P. S. Sreejith**, Principal of RSET, and **Dr. Preetha K. G.**, Head of Department of Computer Science and Engineering for providing us with the opportunity to undertake our mini-project, "Temperature Regulated Fan".

We are highly indebted to our mini-project coordinators, **Dr.Sminu Izudheen**, Professor, Department of Computer Science and Engineering, **Dr. Renu Mary Daniel**, Assistant Professor, Department of Computer Science and Engineering for their valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our mini-project guide **Ms.Liya Joseph**, for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, we would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Abhinav R Nair

Abin Abraham Menacherry

Antony Francis

Ashok Gopal K A

Contents

List of Figures

Chapter 1

Purpose and Need

- **Energy Efficiency**: One of the primary purposes of an automatic temperature-controlled fan is to improve energy efficiency. Traditional fans operate at a constant speed regardless of the surrounding temperature, which can lead to unnecessary energy consumption. By automatically adjusting the fan speed based on the ambient temperature, the fan can operate at lower speeds or even turn off when cooling demands are lower, saving energy and reducing electricity costs.
- **Comfort and Temperature Regulation:** An automatic temperature-controlled fan ensures a more comfortable environment by maintaining a consistent temperature range. When the temperature rises, the fan speed increases to provide effective cooling, and when the temperature drops, the fan speed decreases, preventing the space from becoming too cold. This consistent temperature regulation enhances comfort and creates a more pleasant living or working environment.
- **Noise Reduction:** Traditional fans running at constant high speeds can generate considerable noise, which can be distracting and annoying. An automatic temperature-controlled fan can reduce noise levels by operating at lower speeds when cooling requirements are lower, contributing to a quieter and more peaceful atmosphere.
- **Climate Adaptability:** Different climates and seasons require varying cooling levels. An automatic temperature-controlled fan can adapt to changing environmental conditions, ensuring optimal cooling regardless of the external temperature. This adaptability makes it suitable for use in various locations and during different times of the year.
- **Automation and Convenience:** The automatic nature of the temperature-controlled

fan eliminates the need for constant manual adjustments. Users do not have to worry about turning the fan on or off or adjusting the speed based on temperature changes. This automation provides convenience and allows users to focus on other tasks without the need for constant fan monitoring.

- **Environmental Impact:** Energy-efficient and well-regulated fans have a positive impact on the environment. By reducing energy consumption, the carbon footprint associated with power generation decreases, contributing to a greener and more sustainable future.
- **Equipment Protection:** In environments with sensitive electronic equipment, such as server rooms or electronics laboratories, temperature control is crucial to prevent overheating and potential damage. An automatic temperature-controlled fan helps maintain a stable temperature, protecting valuable equipment from heat- related failures.

Chapter 2

Project Objective

- **Temperature Regulation:** Develop a system that can accurately sense and monitor the ambient temperature in real-time using an appropriate temperature sensor.
- **Dynamic Fan Speed Control:** Design an intelligent control algorithm that automatically adjusts the fan speed according to the temperature readings. The system should increase the fan speed when the temperature rises and decrease it when the temperature falls, ensuring efficient cooling without manual intervention.
- **Energy Efficiency:** Optimize the fan's speed control algorithm to minimize power consumption during periods of low cooling demand. The system should only operate the fan at the necessary speed required to maintain a comfortable temperature range, thereby conserving energy and reducing electricity costs.
- **User-Friendly Interface:** Implement a user-friendly interface that allows users to set their desired temperature range or operating modes, such as automatic control or manual fan speed adjustment. The interface should provide real-time temperature information and fan speed feedback.
- **Safety and Reliability:** Incorporate safety mechanisms to protect the system from overheating or other malfunctions. Implement temperature thresholds to shut down the fan or system in extreme conditions, ensuring safe and reliable operation.
- **Noise Reduction:** Consider noise reduction techniques or use fans with low noise levels to maintain a quiet and comfortable environment, especially during low-speed operations.
- **Cost-Effectiveness:** Design a cost-effective system by carefully selecting suitable

components, sensors, and fan types without compromising performance and reliability.

- **Versatility:** Ensure that the system can be applied in various settings, such as homes, offices, server rooms, greenhouses, and industrial environments, to address diverse cooling requirements.

- **Adaptability:** Develop the system to be adaptable to different temperature ranges and environmental conditions, allowing it to function optimally in various climates.

- **Monitoring and Feedback:** Incorporate a feedback mechanism to continuously monitor the system's performance and temperature regulation. The system should be capable of self-adjusting and fine-tuning the fan speed control algorithm for better efficiency.

Chapter 3

Existing Models

3.1 Dyson Pure Hot + Cool Link

This model is a versatile fan that not only regulates the room temperature but also functions as an air purifier and heater. It comes with a built-in temperature sensorand can be controlled remotely through a smartphone app or via voice commands with compatible smart home platforms.

3.2 Honeywell HYF290B QuietSet Tower Fan

The Honeywell QuietSet Tower Fan features multiple speed settings and an auto-dimming LED display. It has a built-in thermostat that allows the fan to adjust its speed auto- matically based on the room temperature. It also comes with a remote control for added convenience.

3.3 Vornado 660 Large Whole Room Air Circulator Fan

The Vornado 660 is a powerful air circulator fan with adjustable speeds. While it does not have a built-in thermostat, it can be paired with a separate smart plug or thermostat device that allows for automatic temperature regulation based on external temperature readings.

3.4 Lasko 2551 Wind Curve Tower Fan

The Lasko Wind Curve Tower Fan features a built-in thermostat and can oscillate to distribute airflow evenly throughout the room. It offers various fan speeds and also includes a remote control for easy operation.

3.5 Rowenta VU5670 Turbo Silence Extreme Stand Fan

The Rowenta Turbo Silence Extreme Stand Fan is equipped with a temperature sensor that allows the fan to adjust its speed automatically based on the ambient temperature. It also features an electronic control panel and a remote control.

3.6 Ozeri 3x Tower Fan with Passive Noise Reduction Technology

The Ozeri 3x Tower Fan comes with a built-in thermostat and three independently controllable fans for customizable airflow. It features a touch-sensitive control panel and a remote for ease of use.

3.7 Honeywell HT-900 TurboForce Air Circulator Fan

The Honeywell HT-900 is a compact yet powerful fan with a variable speed control. While it doesn’t have an integrated thermostat, it can be paired with external smart devices for automatic temperature regulation.

Chapter 4

Proposed Models

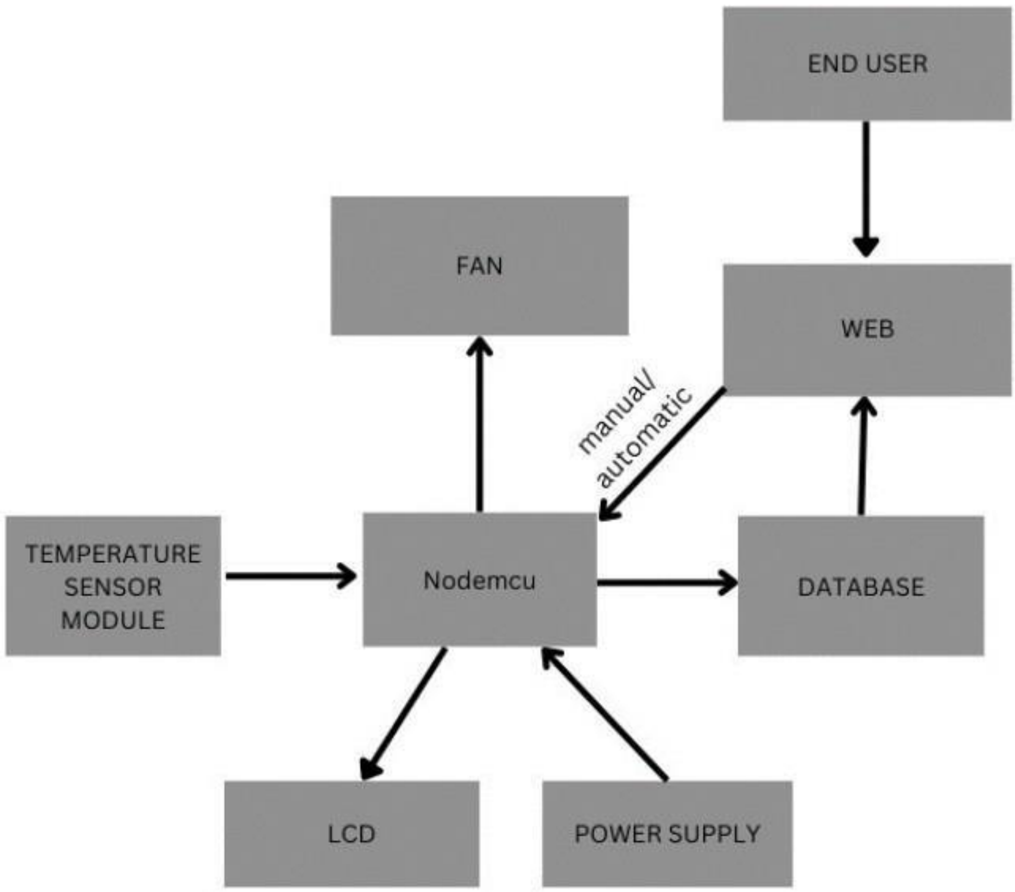
- **PC Case Fans with Temperature Sensors:** These fans are designed to be installed inside computer cases to provide cooling for the internal components. They come with built-in temperature sensors or connectors to attach external temperature sensors. Users can set the desired temperature threshold and the fan will adjust its speed accordingly.
- **Room Fans with Thermostat Control:** These are standalone room fans equipped with a temperature sensor and a control panel. User can set the desired room temperature, and in the fan will automatically adjust its speed to maintain the set temperature.
- **Smart Fans with App Control:** Some modern fans come with built-in Wi-Fi or Bluetooth connectivity, allowing users to control them using smartphone app. These smart fans often have temperature sensors and can be set to operate based on their requirements.
- **DIY-Temperature Controlled Fans:** Hobbyists often create their temperature-controlled fan systems using microcontrollers like Arduino or Raspberry Pi. They can integrate various sensors and customize the fan’s behavior based on their requirements.
- **Industrial Temperature-Controlled Fans:** These fans are designed for indus- trial settings where temperature regulation is crucial. They may have more ad- vanced features, such as communication interfaces for integration into larger control systems.
- **Greenhouse Fans with Thermostat:** Greenhouse fans help regulate temperature and humidity levels inside greenhouses. They often come with built-in thermostats

that allow users to adjust the fan speed based on greenhouse temperature.

- **Celling Fan with Temperature Sensing:** Some high-end celling fans come with temperature sensors that adjust the fan speed based on the room temperature, providing a more comfortable environment.

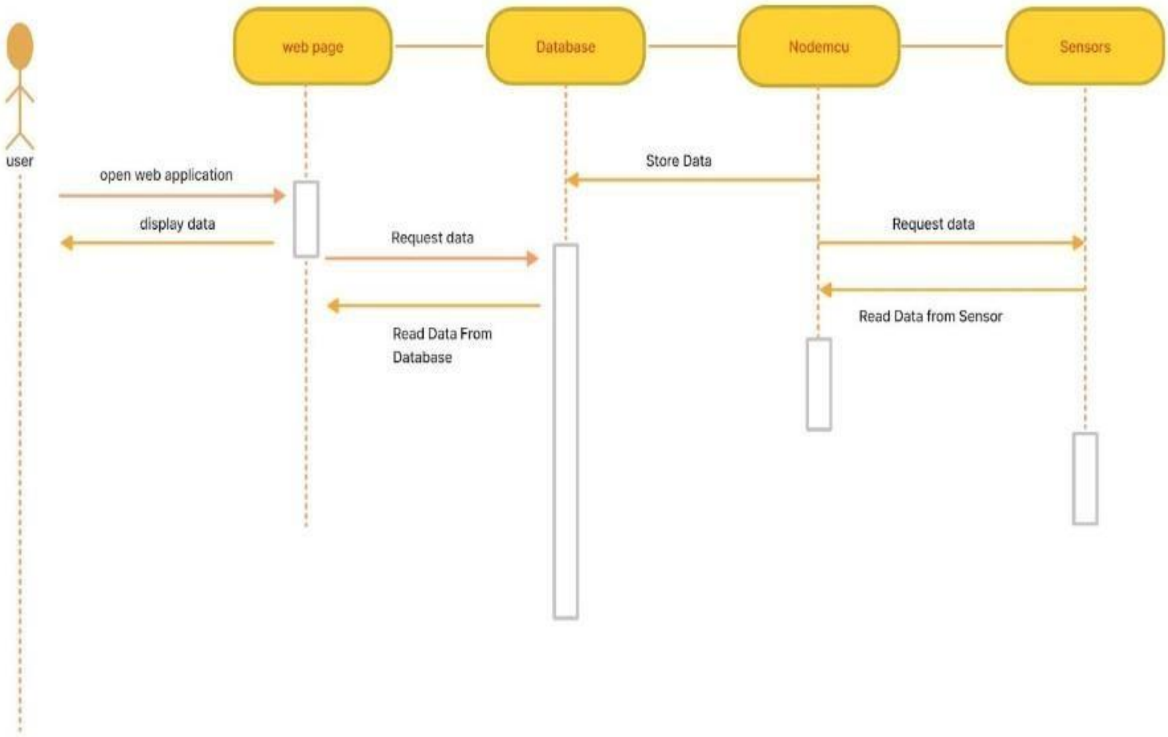
Chapter 5

Architecture Diagram



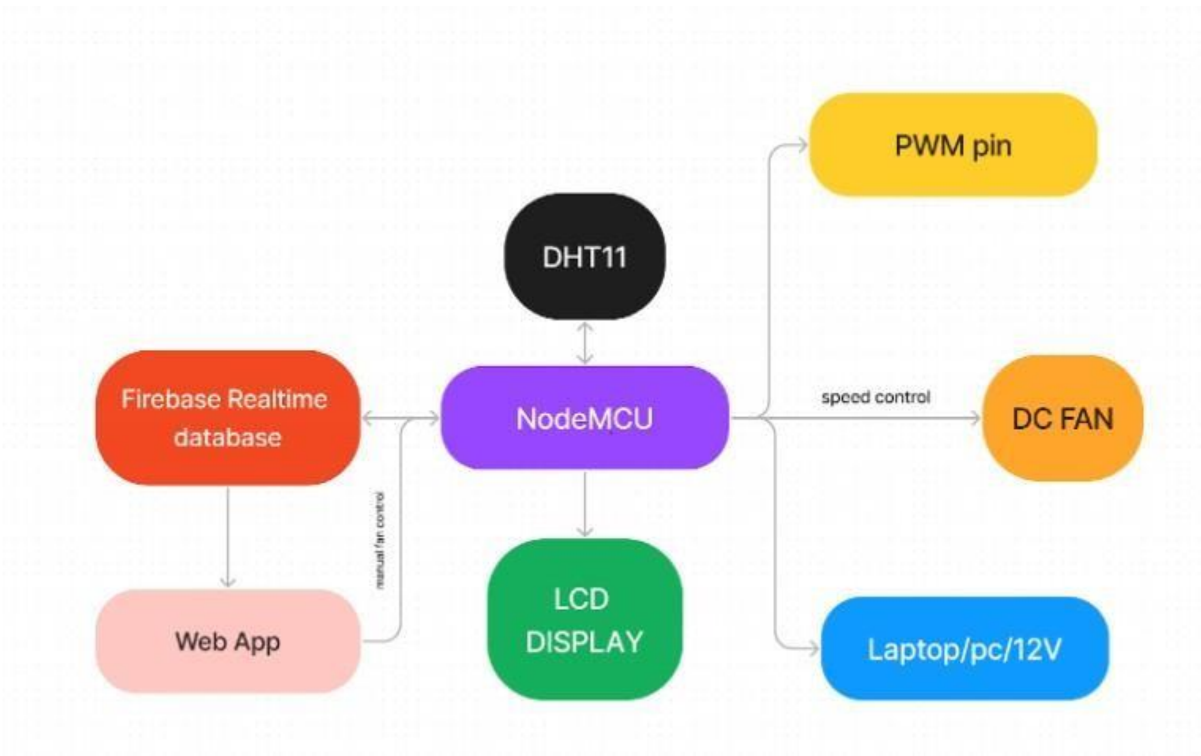
Chapter 6

Sequence Diagram



Chapter 7

Module-Wise Diagram



Chapter 8

Module Division

8.1 Temperature Sensing Module:

This module includes the temperature sensor responsible for measuring the ambient temperature. It can be an analog or digital sensor like a thermistor, thermocouple, or digital temperature sensor (e.g., DS18B20). The module reads temperature data and sends it to the control module for further processing.

8.2 Control Module:

The control module is the brain of the system and houses the microcontroller or microprocessor. It receives temperature data from the temperature sensing module and processes it using a control algorithm. The algorithm determines the appropriate fan speed based on the temperature readings.

8.3 Fan Speed Regulation Module:

This module is responsible for controlling the fan's speed. It can use Pulse Width Modulation (PWM) or variable voltage control to adjust the fan's RPM (Revolutions Per Minute) according to the instructions from the control module.

8.4 User Interface Module:

The user interface module allows users to interact with the system. It can include an LCD display, touch screen, or buttons for setting preferred temperature ranges, operating modes (automatic/manual), and viewing real-time temperature information and fan speed.

8.5 Hardware Implementation Module:

This module involves the physical implementation of the system, including the selection and integration of components such as the fan(s), temperature sensor, micro controller, power supply, and other electronic components.

Chapter 9

Assumptions

- **Temperature Sensor Accuracy:** It is assumed that the temperature sensor used in the system provides accurate and reliable temperature readings. The sensor's precision and calibration are assumed to be sufficient for maintaining a comfortable environment and making effective fan speed adjustments.

- **Stable Internet Connection:** The web app's functionality relies on a stable in- ternet connection. It is assumed that users will have access to a reliable internet connection to interact with the system through the web application.

- **Firebase Real-time Database:** The assumption is made that the Firebase real-time database will be used to store and retrieve temperature values, PWM settings, and timestamps. It is assumed that Firebase will operate without major downtime and will effectively store and retrieve data as required by the system.

- **Web Application Accessibility:** It is assumed that the web application will be accessible from various devices, including smartphones, tablets, and computers. The web app's responsiveness is assumed to provide a consistent user experience across different screen sizes.

Chapter 10

10.0.1 Work Breakdown and Responsibilities

ABHINAV R NAIR	INTERFACING WEB WITH NODEMCU PROGRAM,HARDWARE CONNECTION
ABIN ABRAHAM	CSS,HARDWARE CONNECTIONS,INTERFACING WITH LCD
ANTONY FRANCIS	INTERFACING WITH DC FAN,DHT11
ASHOK GOPAL K A	HTML,JAVASCRIPT,FIREBASE

10.0.2 Work Flow Diagram

Week 1	Week 2	Week 3	Week 4	Week 5	
Sensor Data Processing	Interfacing Nodemcu with LCD	Interfacing with DC Fan with Nodemcu	Web Page Development	Database Integration	Graph Plotting

Chapter 11

Hardware Requirements

11.1 NodeMcu:

NodeMCU is an open-source development board that is based on the ESP8266 micro-controller. The ESP8266 is a Wi-Fi-enabled system-on-a-chip (SoC) that has gained significant popularity in the world of Internet of Things (IoT) due to its low cost, ease of use, and wireless connectivity capabilities.

11.2 DHT11(Temperature Humidity Sensor):

The DHT11 is a low-cost digital temperature and humidity sensor widely used in vari- ous electronics and IoT projects. It is popular among hobbyists and beginners due to its simplicity and ease of use. The sensor is capable of measuring temperature and hu- midity and providing these values in a digital format, making it easy to interface with microcontrollers like Arduino, Raspberry Pi, NodeMCU, and other platforms.

11.3 DC Fan:

A ”DC fan” refers to a fan that operates using direct current (DC) power. These fans are commonly used in various electronic devices, appliances, and cooling systems.

11.4 16x2 LCD Display:

A 16x2 LCD display, also known as a 16x2 character LCD or simply a 16x2 LCD, is a type of liquid crystal display (LCD) that can display 16 characters in each of its two rows, totaling 32 characters in a 2-line format.

11.5 I2C Module:

An I2C module, also known as an I2C interface module or I2C breakout board, is a hardware component that facilitates communication between devices using the I2C (Inter- Integrated Circuit) protocol. I2C is a popular serial communication protocol that allows multiple devices to communicate with each other over a shared bus using only two wires:one for data (SDA) and one for the clock (SCL).

11.6 Transistor:

A transistor is a semiconductor device that can act as a switch or an amplifier. It has three terminals: the base, the emitter, and the collector

11.7 Resistor:

A resistor is a passive electronic component that limits the flow of electric current in a circuit. It is used to control the amount of current or voltage in a specific part of a circuit.

11.8 Connecting Wires:

Connecting wires, also known as jumper wires or hookup wires, are used to establish electrical connections between different components in an electronic circuit.

11.9 Breadboard:

A breadboard is a construction base used to build semi-permanent prototypes of elec- tronic circuits. It is a device used to create temporary electronic circuits for testing and experimentation without the need for soldering. The breadboard allows you to quickly connect electronic components, such as resistors, capacitors, integrated circuits, and other discrete elements, to build and test various electronic circuits. The board typically has two sets of horizontal rows running along the top and bottom. These are called terminal strips. Each terminal strip is usually powered and internally connected, with all holes within a strip electrically connected. In the centre of the breadboard, there are usually two sets of vertical rows, known as bus strips. These rows are used for power and ground connections. All holes within a bus strip are also electrically connected.

Chapter 12

Software Requirements

12.1 VS-Code:

Visual Studio is a popular integrated development environment (IDE) developed by Microsoft. It is primarily used for creating, debugging, and deploying software applications, including desktop applications, web applications, mobile apps, games, and cloud-based solutions. Visual Studio supports a variety of programming languages, such as C, C++, Visual Basic, JavaScript, Python, and more.

12.2 Arduino IDE:

The Arduino IDE (Integrated Development Environment) is a software application used for programming and developing applications for Arduino boards. Arduino is an open- source electronics platform that provides a range of microcontroller boards and a software library to facilitate the creation of various electronic projects.

12.3 Firebase Realtime Database:

Firebase Realtime Database is a cloud-based NoSQL database provided by Google as part of the Firebase platform. It is designed to store and sync data in real-time between clients and servers, making it ideal for building real-time applications and collaborative systems.

Chapter 13

Risks and Challenges

Temperature-regulated fans, also known as smart fans or variable-speed fans, are designed to adjust their speed based on the ambient temperature or the temperature inside a specific system. While they offer various benefits, they also come with some risks and challenges:

1. Inaccurate temperature readings: If the temperature sensor used to regulate the fan speed is not calibrated properly or becomes faulty, it may lead to incorrect temperature readings. This can cause the fan to operate at inappropriate speeds, potentially leading to inadequate cooling or unnecessary noise.

2. Limited compatibility: Temperature-regulated fans may not be universally compatible with all systems or environments. They often rely on specific connectors or interfaces, which might not match the existing hardware, necessitating additional adapters or modifications.

3. System overheating: In some cases, temperature-regulated fans may not ramp up their speed quickly enough to handle sudden increases in temperature. This delay could lead to temporary overheating of the system components before the fan reaches an adequate cooling level.

4. Fan noise fluctuations: While temperature-regulated fans can reduce noise levels when the system is not under heavy load, they might produce more noticeable fluctuations in noise as the fan speed adjusts in response to temperature changes.This could be distracting for users sensitive to changes in fan noise.

5. Mechanical failure: Like any mechanical component, temperature-regulated fans can fail. If a fan stops working or malfunctions, it could lead to inadequate cooling, potentially causing damage to the system’s components.

Chapter 14

Economic feasibility

- **Initial Investment:** The initial investment includes the cost of hardware components such as the temperature sensor, microcontroller or microprocessor, PWM controller, fan, and any other required electronics.
- **Ongoing Operational Costs:** The ongoing operational costs include electricity consumption by the fan and the web server hosting the web app. The fan's energy consumption will vary depending on the fan's size, motor efficiency, and usage patterns.
- **Potential Savings:** The economic feasibility of the system can be justified by potential energy savings achieved through the intelligent fan speed control based on temperature sensing. By optimizing the fan speed and avoiding unnecessary cooling, the system can lead to reduced energy consumption, resulting in energy cost savings over time.
- **Market Demand:** Economic feasibility is also influenced by the potential market demand for such a system. If there is a significant demand for energy-efficient temperature-regulated fans, the system can have a positive economic impact.
- **Lifespan and Maintenance Costs:** The lifespan of the components and the system's reliability play a crucial role in economic feasibility. Longer lifespans and lower maintenance costs lead to more cost-effective long-term operation.
- **Scalability and Upgrades:** If the system is designed with scalability in mind, it can accommodate future upgrades and improvements, enhancing its long-term economic viability.

Chapter 15

Conclusion

In conclusion, the temperature-regulated fan system with manual control using a web app and storing temperature values, PWM value, date, and time in Firebase offers a practical and efficient solution for maintaining a comfortable environment while providing users with convenient control and real-time monitoring capabilities. By integrating various modules, the system ensures accurate temperature sensing, PWM-based fan control, and seamless communication with the web application for both automatic and manual fan speed adjustments.

The automatic mode of the system ensures a hands-free approach to fan speed regulation, as it intelligently adapts the fan speed based on the measured temperature. This feature optimizes energy consumption and creates a more comfortable living or working space by eliminating the need for constant manual adjustments. On the other hand, the manual control through the web app empowers users to customize the fan speed according to their preferences, giving them full flexibility and control over the environment.

The web application provides an intuitive and user-friendly interface, allowing users to monitor the current temperature, fan speed, and operating mode in real-time. This level of transparency empowers users with valuable insights into the system's performance and temperature fluctuations, aiding in better decision-making for temperature regulation.

The integration with Firebase, a cloud-based real-time database, ensures reliable data storage and logging capabilities. All temperature readings, PWM values, date, and time are securely stored in Firebase, allowing users to access historical data for analysis, comparison, and long-term monitoring. The data logging feature provides an essential tool for understanding temperature trends and optimizing the fan system's performance over time.

Chapter 16

Results

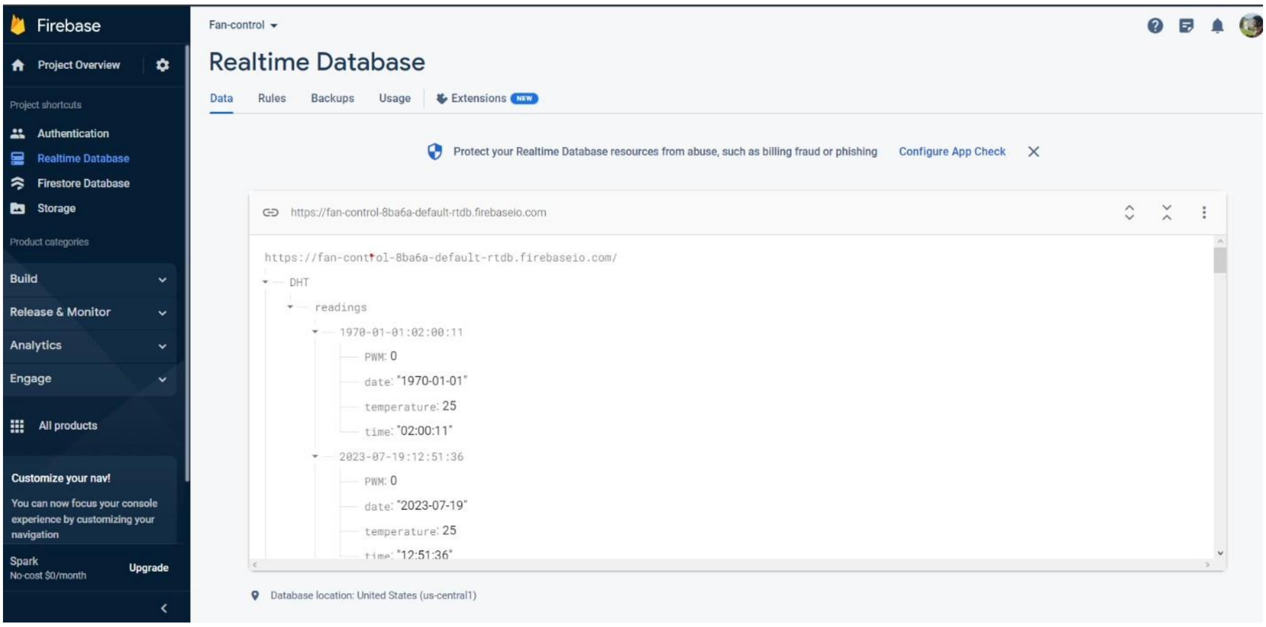


Figure 16.1 Firebase database readings

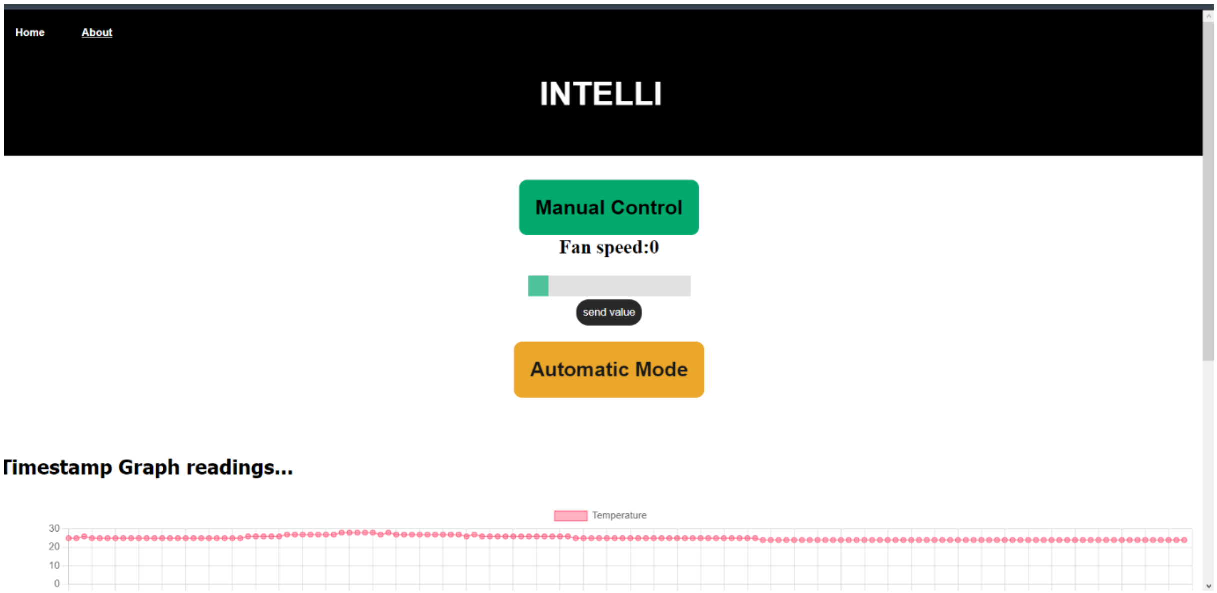


Figure 16.2 Webpage home

Results

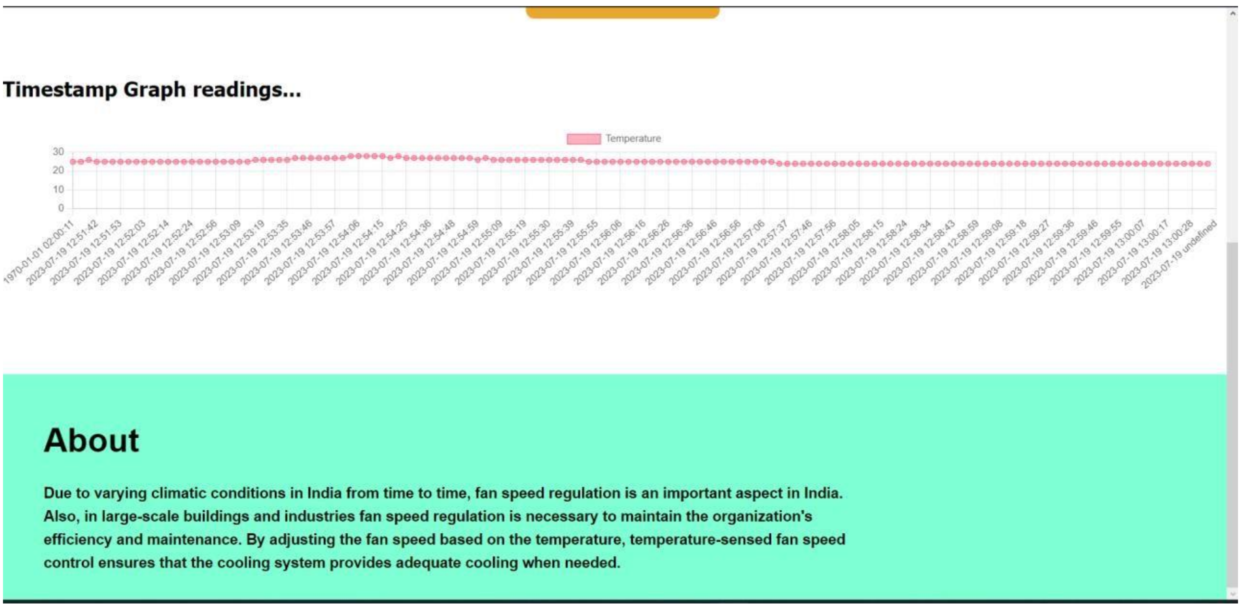


Figure 16.3 Web About section

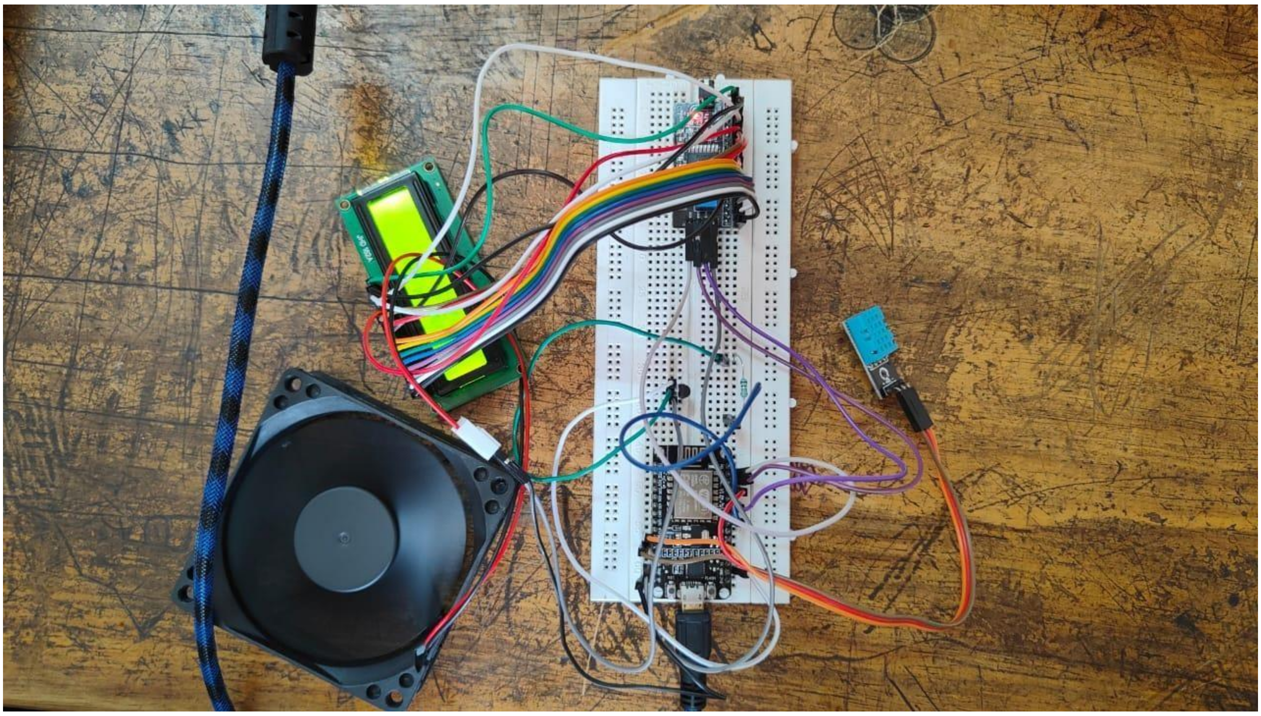


Figure 16.4 Hardware components and connection

References

[1] https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf

[2] ”Arduino Cookbook” by Michael Margolis: This book covers various Arduino projects, including temperature sensing with DHT sensors and PWM controlling for fan speed or other applications.2.

[3] https://handsontec.com/dataspecs/module/esp8266-V13.pdf

[4] https://www.vishay.com/docs/37299/37299.pdf

[5] https://www.w3schools.co

[6] https://circuitdigest.com

[7] https://embetronicx.com

Appendix A: Sample Code

NodeMcu- code (sensor reading and temperature control)

```
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <DHT11.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <TimeLib.h>
#include <Arduino.h>
#include <Firebase_ESP_Client.h>
#include <LiquidCrystal_I2C.h>
#include <ESP8266WebServer.h> //
#include <WiFiClient.h>  //
String timestamp;
int pwm_value=0;
bool flag = false; // Flag variable

LiquidCrystal_I2C lcd(0x27, 16, 2);

// Provide the token generation process info.
#include "addons/TokenHelper.h"
// Provide the RTDB payload printing info and other helper functions.
#include "addons/RTDBHelper.h"

ESP8266WebServer server(80);

#define WIFI_SSID "realme GT 2"
#define WIFI_PASSWORD "12345678"

// Insert Firebase project API Key
#define API_KEY "AIzaSyAIHizGPzxfkuXvK8xPmMAakwc4n4rN8yE"

// Insert RTDB URLefine the RTDB URL */
#define DATABASE_URL "https://fan-control-8ba6a-default-rtdb.firebaseio.com/"

// Define Firebase Data object
FirebaseData fbdo;

FirebaseAuth auth;
FirebaseConfig config;

bool signupOK = false;

WiFiClient client;

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");


DHT11 dht11(D3);
```

```
#define pwm D5


void setup() {

  lcd.init();  // initializing the LCD
  lcd.backlight(); // Enable or Turn On the backlight
  Serial.begin(115200);
  delay(10);
  pinMode(pwm, OUTPUT);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi...");

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

  }
  Serial.println();
  Serial.print("Connected to Wi-Fi. IP address: ");
  Serial.println(WiFi.localIP());

  /* Assign the API key (required) */
  config.api_key = API_KEY;

  /* Assign the RTDB URL (required) */
  config.database_url = DATABASE_URL;

  /* Sign up */
  if (Firebase.signUp(&config, &auth, "", ""))
  {
    Serial.println("ok");
    signupOK = true;
  }
  else
  {
    Serial.printf("%s\n", config.signer.signupError.message.c_str());
  }

  /* Assign the callback function for the long running token generation task */
  config.token_status_callback = tokenStatusCallback; // see addons/TokenHelper.h

  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);

  configTime(0, 0, "pool.ntp.org");
  while (!time(nullptr)) {
  Serial.print(".");
```

```
    delay(1000);
  }
  Serial.println();
  Serial.println("Time and date initialized.");

  server.on("/slider", handleSliderValue);

  server.begin();
  Serial.println("HTTP server started");
}



void handleSliderValue() {
  if (server.hasArg("value") && server.hasArg("value2")) {
    String value = server.arg("value");
    pwm_value = value.toInt();  //slider value
    int value2 = server.arg("value2").toInt(); // manual mode =11 / automatic mode=22
    // Do something with the slider value
    // For example, print it to the serial monitor
    Serial.print("Slider value: ");
    Serial.println(pwm_value);

    // Check the value of value2
    if (value2 == 22) {
      // Set the flag to true
      Serial.print("value=22");
      lcd.clear();
      Serial.print("Automatic mode\n");
      lcd.print("Automatic Mode");
      flag = true;
    }
    else
    flag= false;
    delay(200);
  }
}

void loop() {

  int temp = dht11.readTemperature();  // Read temperature value in Celsius
  Serial.print("Temperature: ");
  Serial.print(temp);
  Serial.println("°C");


  if (flag) {
  // Execute code if the flag is true (value2 is 22)
    lcd.clear();
    lcd.setCursor(0, 0);
```

```
lcd.print("Automatic Mode");
lcd.setCursor(0, 0);
lcd.print("Temperature:");
lcd.print(temp);
Serial.print("Automatic Mode");
lcd.setCursor(0, 1);
if (temp < 25)
{
  pwm_value = 0;
  analogWrite(9, 0);
  lcd.print("Fan OFF           ");
  delay(100);
}

else if (temp == 25)
{
  pwm_value = 55;
  analogWrite(pwm, 55);
  lcd.print("Fan Speed: 20%  ");
  delay(100);

}

else if (temp == 26)
{
  pwm_value = 110;
  analogWrite(pwm, 110);
  lcd.print("Fan Speed: 40%  ");
  delay(100);
}

else if (temp == 27)
{
  pwm_value = 165;
  analogWrite(pwm, 255);
  lcd.print("Fan Speed: 100%  ");
  delay(100);
}
else if (temp == 31)
{
  pwm_value = 165;
  analogWrite(pwm, 200);
  lcd.print("Fan Speed: 80%  ");
  delay(100);
}

else if (temp > 28)

{
```

```
      pwm_value = 255;
      lcd.print("Fan Speed: 100%  ");
      analogWrite(pwm, 255);
      delay(100);
    }
  }
   else {
   // Execute code if the flag is false (value2 is not 22)
      lcd.setCursor(0, 0);
      lcd.clear();
      lcd.print("Manual Mode");
      lcd.setCursor(0, 1);
      lcd.print("PWM:");
      lcd.print(pwm_value);
      Serial.print("Manual Mode\n");
      analogWrite(pwm, pwm_value);
  }


   time_t now = time(nullptr);                    //read time and date from web
   struct tm *timeinfo = localtime(&now);
   char timestampBuffer[20],date[20],time[20];
   sprintf(timestampBuffer, "%04d-%02d-%02d:%02d:%02d:%02d", timeinfo->tm_year + 1900, timeinfo->tm_mon +
1,
          timeinfo->tm_mday, timeinfo->tm_hour+2, timeinfo->tm_min, timeinfo->tm_sec);
   sprintf(date,"%04d-%02d-%02d",timeinfo->tm_year + 1900, timeinfo->tm_mon + 1,timeinfo->tm_mday);
   sprintf(time,"%02d:%02d:%02d",timeinfo->tm_hour+2, timeinfo->tm_min, timeinfo->tm_sec);
   timestamp = String(timestampBuffer);

   if (Firebase.ready() && signupOK)
   {

     // Create a new child node with the timestamp
     String path = "DHT/readings/" + timestamp;
     Firebase.RTDB.setString(&fbdo, path + "/date", date);
     Firebase.RTDB.setString(&fbdo, path + "/time", time);

     // Write pwm_value value to the database
     if (Firebase.RTDB.setFloat(&fbdo, path + "/PWM", pwm_value))
     {
       Serial.print("\nPWM: ");
       Serial.println(pwm_value);
     }
     else
     {
       Serial.println("FAILED");
       Serial.println("REASON: " + fbdo.errorReason());
     }
```

```
    // Write temperature value to the database
    if (Firebase.RTDB.setFloat(&fbdo, path + "/temperature", temp))
    {
      Serial.print("Temperature: ");
      Serial.println(temp);
    }
    else
    {
      Serial.println("FAILED");
      Serial.println("REASON: " + fbdo.errorReason());
    }
  }
  server.handleClient();
}
```

Web-application(html,css,javascript) with firebase integration

```
<!DOCTYPE html>
<html>
	<head>
		<title>Intelli</title>
		<script src="https://www.gstatic.com/firebasejs/8.7.1/firebase-app.js"></script>
		<script src="h ttps://www.gstatic.com/firebasejs/8.7.1/firebase-database.js"></script>
		<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
		<style>
			html {
				scroll-behavior: smooth;
			}
			.body{
				margin: 0px;
				padding: 0px;
			}

			.chart-container {
				display: flex;
				flex-direction: column;
				align-items: center;
				margin-top: 30px;
			}
			.background{
				background-color: rgb(0, 0, 0);
```

```
        margin: 0px;
        height: 25vh;
        font-size: 20px;
        font-family: Arial, Helvetica, sans-serif, Times, serif, sans-serif;
        margin-bottom: 50px;
}
.head{
        text-align:  center;
        margin-top:  25px;
        color: white;
}

.head2{

        margin-top: 70px;
        font-size: 25px;
        font-family:  Bree Verdana , Tahoma, sans-serif;
}
.top{
        color: white;
        padding: 20px;
        margin: 0px;
        font-family: Arial, Helvetica, sans-serif;
        font-size: 13px;
        font-weight: bold;
        display:inline-flex;
}

.slider_bar {
        height: 25px;
        background: #d3d3d3;
        outline: none;
        -webkit-appearance: none;
        -moz-appearance: none;
        appearance: none;
        -webkit-transition: .2s;
        transition: .2s;
        width: 200px;
        opacity: 0.7;
}

.slider_bar:hover {
        opacity: 1;
}

.slider_bar::-webkit-slider-thumb {
        -webkit-appearance: none;
```

```
	-moz-appearance: none;
	appearance: none;
	width: 25px;
	height: 25px;
	background: #04AA6D;
	cursor: pointer;
}

#sliderValue {
	margin-top: 10px;
}
.manual{
	padding: 20px;
	font-size: 25px;
	font-weight: bold;
	background-color:#04AA6D;
	border: none;
	font-family: Helvetica, sans-serif;
	border-radius: 10px;

}
.content{
	margin-left: 20px;
	text-align: center;
}
.send-value-button{
	background-color: #282828;
	color: white;
	border: none;
	border-radius: 15px;
	font-family: Helvetica, sans-serif;
	font-weight: medium;
	padding:8px;

}
.send-value-button:hover{
	cursor: pointer;
	background-color: black;
	opacity: 1;
}
.button-automatic{
	background-color: rgb(232, 158, 20);
	border: none;
	font-size: 25px;
	font-weight: bold;
```

```
				padding: 20px;
				border-radius: 10px;
				margin-top: 20px;

				opacity: .9;
			}

			.button-automatic:hover{
				cursor: pointer;
				opacity: 1;
			}
			.about{
				margin-top: 100px;
				background-color: aquamarine;
				padding: 30px;
			}

			.about-head{

				font-family: Helvetica, sans-serif;
				font-weight: bolder;
				margin-left: 20px;
				font-size: 40px;

			}
			.about-desc{
				margin-left: 20px;
				font-family: Arial, Helvetica, sans-serif;
				font-weight: bold;
				width: 1000px;
				margin-bottom: 0px;
				line-height: 28px;
				font-size: 18px;
			}
		</style>

	</head>

	<body class="body">
		<div class="background">
			<p class="top">Home</p>
			<a href="#About" class="top">About</a>
			<h1 class="head">INTELLI</h1>
		</div>

		<div class="content">
```

```
			<label class="manual">Manual Control</label>
			<h2>Fan speed:<span id="sliderValue">50</span></h2>
			<input type="range" id="slider" min="0" max="255" step="1" value="0"
class="slider_bar" oninput="sendSliderValue(this.value)"> <br>

			<button class="send-value-button" onclick="sendSliderValue()">
				send value
			</button> <br>

			<div>
				<button class="button-automatic" onclick="sendSliderValue2()">
					Automatic Mode
				</button>
			</div>
		</div>

		<h3 class="head2">Timestamp Graph readings...</h3>
	</body>

	<body>
		<div class="chart-container">
			<canvas id="lineChart" style="height: 200px;"></canvas>
		</div>
		<div id="About" class="about">
			<h1 class="about-head">About</h1>
			<p class="about-desc"> Due to varying climatic conditions in India from time to time,
fan speed regulation is an
				important aspect in India. Also, in large-scale buildings and industries fan
speed regulation is
				necessary to maintain the organization's efficiency and maintenance. By
adjusting the fan
				speed based on the temperature, temperature-sensed fan speed control
ensures that the
				cooling system provides adequate cooling when needed.
			</p>
		</div>

		<script>

			var slider = document.getElementById("slider");
			var output = document.getElementById("sliderValue");
			output.innerHTML = slider.value;

			slider.oninput = function() {
			  output.innerHTML = this.value;
			}
```

```
function sendSliderValue() {
	// Get the slider value
	var value = document.getElementById("slider").value;

	var value2=11;
	// Create a new XMLHttpRequest object
	var xhr = new XMLHttpRequest();

	// Prepare the request URL
	var url = "http://192.168.136.92/slider?value=" + value + "&value2=" +
value2.toString();

	// Set up the request
	xhr.open("GET", url, true);

	// Send the request
	xhr.send();
}


function sendSliderValue2() {
	var value=0;
	var value2=22;
	// Create a new XMLHttpRequest object
	var xhr = new XMLHttpRequest();
	window.alert("Fan set on Automatic Mode");
	// Prepare the request URL
	var url = "http://192.168.136.92/slider?value=" + value + "&value2=" +
value2;

	// Set up the request
	xhr.open("GET", url, true);

	// Send the request
	xhr.send();
}

// Initialize Firebase
var firebaseConfig = {
  apiKey: "AIzaSyAIHizGPzxfkuXvK8xPmMAakwc4n4rN8yE",
  authDomain: "",
  databaseURL: "https://fan-control-8ba6a-default-rtdb.firebaseio.com/",
  projectId: "Fan-control",
  storageBucket: "",
  messagingSenderId: "YOUR_MESSAGING_SENDER_ID",
  appId: "YOUR_APP_ID",
```

```
};
firebase.initializeApp(firebaseConfig);

// Get a reference to the database
var database = firebase.database();

// Get a reference to the canvas element
var chartCanvas = document.getElementById('lineChart');

// Create the chart instance
var lineChart = new Chart(chartCanvas, {
  type: 'line',
  data: {
        labels: [],
        datasets: [{
              label: 'Temperature',
              data: [],
              borderColor: 'rgba(255, 99, 132, 1)',
              backgroundColor: 'rgba(255, 99, 132, 0.5)',
              borderWidth: 1,
              fill: false
        }]
  },
  options: {
        responsive: true,
        maintainAspectRatio: false,
        scales: {
              y: {
                beginAtZero: true
              }
        }
  }
});

// Read data from Firebase database and update the chart
var dataRef = database.ref('DHT/readings');
dataRef.on('value', function(snapshot) {
  var data = snapshot.val();
  var labels = [];
  var temperatureData = [];

  for (var timestamp in data) {
        if (data.hasOwnProperty(timestamp)) {
              var temperature = data[timestamp].temperature;
              var date = data[timestamp].date;
```

```
                        var time = data[timestamp].time;

                        labels.push(date + ' ' + time);
                        temperatureData.push(temperature);
                    }
                }

                lineChart.data.labels = labels;
                lineChart.data.datasets[0].data = temperatureData;
                lineChart.update();
               });
        </script>

    </body>

</html>
```

COURSE OUTCOMES:

After completion of the course the student will be able to

SL. NO	DESCRIPTION	Blooms' Taxonomy Level
CO1	Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO2	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO4	Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO5	Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)	Level 3: Apply

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO3
CO1	3	3	3	3		2	2	3	2	2	2	3	2	2	2
CO2	3	3	3	3	3	2		3	2	3	2	3	2	2	2
CO3	3	3	3	3	3	2	2	3	2	2	2	3			2
CO4	2	3	2	2	2			3	3	3	2	3	2	2	2
CO5	3	3	3	2	2	2	2	3	2		2	3	2	2	2

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/ MEDIUM/ HIGH	JUSTIFICATION
100003/CS622T.1-PO1	**HIGH**	Identify technically and economically feasible problems by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.1-PO2	**HIGH**	Identify technically and economically feasible problems by analysing complex engineering problems reaching substantiated conclusions using first principles of mathematics.
100003/CS622T.1-PO3	**HIGH**	Design solutions for complex engineering problems by identifying technically and economically feasible problems.
100003/CS622T.1-PO4	**HIGH**	Identify technically and economically feasible problems by analysis and interpretation of data.
100003/CS622T.1-PO6	**MEDIUM**	Responsibilities relevant to the professional engineering practice by identifying the problem.
100003/CS622T.1-PO7	**MEDIUM**	Identify technically and economically feasible problems by understanding the impact of the professional engineering solutions.
100003/CS622T.1-PO8	**HIGH**	Apply ethical principles and commit to professional ethics to identify technically and economically feasible problems.
100003/CS622T.1-PO9	**MEDIUM**	Identify technically and economically feasible problems by working as a team.
100003/CS622T.1-PO10	**MEDIUM**	Communicate effectively with the engineering community by identifying technically and economically feasible problems.
100003/CS622T.1-P011	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles by selecting the technically and economically feasible problems.
100003/CS622T.1-PO12	**HIGH**	Identify technically and economically feasible problems for long term learning.
100003/CS622T.1-PSO1	**MEDIUM**	Ability to identify, analyze and design solutions to identify technically and economically feasible problems.
100003/CS622T.1-PSO2	**MEDIUM**	By designing algorithms and applying standard practices in software project development and Identifying technically and economically feasible problems.
100003/CS622T.1-PSO3	**MEDIUM**	Fundamentals of computer science in competitive research can be applied to Identify technically and economically feasible problems.
100003/CS622T.2-PO1	**HIGH**	Identify and survey the relevant by applying the knowledge of mathematics, science, engineering fundamentals.

100003/CS622T.2-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems get familiarized with software development processes.
100003/CS622T.2-PO3	**HIGH**	Design solutions for complex engineering problems and design based on the relevant literature.
100003/CS622T.2-PO4	**HIGH**	Use research-based knowledge including design of experiments based on relevant literature.
100003/CS622T.2-PO5	**HIGH**	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes by using modern tools.
100003/CS622T.2-PO6	**MEDIUM**	Create, select, and apply appropriate techniques, resources, by identifying and surveying the relevant literature.
100003/CS622T.2-PO8	**HIGH**	Apply ethical principles and commit to professional ethics based on the relevant literature.
100003/CS622T.2-PO9	**MEDIUM**	Identify and survey the relevant literature as a team.
100003/CS622T.2-PO10	**HIGH**	Identify and survey the relevant literature for a good communication to the engineering fraternity.
100003/CS622T.2-PO11	**MEDIUM**	Identify and survey the relevant literature to demonstrate knowledge and understanding of engineering and management principles.
100003/CS622T.2-PO12	**HIGH**	Identify and survey the relevant literature for independent and lifelong learning.
100003/CS622T.2-PSO1	**MEDIUM**	Design solutions for complex engineering problems by Identifying and survey the relevant literature.
100003/CS622T.2-PSO2	**MEDIUM**	Identify and survey the relevant literature for acquiring programming efficiency by designing algorithms and applying standard practices.
100003/CS622T.2-PSO3	**MEDIUM**	Identify and survey the relevant literature to apply the fundamentals of computer science in competitive research.
100003/CS622T.3-PO1	**HIGH**	Perform requirement analysis, identify design methodologies by using modern tools & advanced programming techniques and by applying the knowledge of mathematics, science, engineering fundamentals.
100003/CS622T.3-PO2	**HIGH**	Identify, formulate, review research literature for requirement analysis, identify design methodologies and develop adaptable & reusable solutions.

100003/CS622T.3-PO3	**HIGH**	Design solutions for complex engineering problems and perform requirement analysis, identify design methodologies.
100003/CS622T.3-PO4	**HIGH**	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/CS622T.3-PO5	**HIGH**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools.
100003/CS622T.3-PO6	**MEDIUM**	Perform requirement analysis, identify design methodologies and assess societal, health, safety, legal, and cultural issues.
100003/CS622T.3-PO7	**MEDIUM**	Understand the impact of the professional engineering solutions in societal and environmental contexts and Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions.
100003/CS622T.3-PO8	**HIGH**	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions by applying ethical principles and commit to professional ethics.
100003/CS622T.3-PO9	**MEDIUM**	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
100003/CS622T.3-PO10	**MEDIUM**	Communicate effectively with the engineering community and with society at large to perform requirement analysis, identify design methodologies.
100003/CS622T.3-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering requirement analysis by identifying design methodologies.
100003/CS622T.3-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by analysis, identify design methodologies and develop adaptable & reusable solutions.
100003/CS622T.3-PSO3	**MEDIUM**	The ability to apply the fundamentals of computer science in competitive research and prior to that perform requirement analysis, identify design methodologies.
100003/CS622T.4-PO1	**MEDIUM**	Prepare technical report and deliver presentation by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.4-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems by preparing technical report and deliver presentation.

100003/CS622T.4-PO3	**MEDIUM**	Prepare Design solutions for complex engineering problems and create technical report and deliver presentation.
100003/CS622T.4-PO4	**MEDIUM**	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions and prepare technical report and deliver presentation.
100003/CS622T.4-PO5	**MEDIUM**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and Prepare technical report and deliver presentation.
100003/CS622T.4-PO8	**HIGH**	Prepare technical report and deliver presentation by applying ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
100003/CS622T.4-PO9	**HIGH**	Prepare technical report and deliver presentation effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
100003/CS622T.4-PO10	**HIGH**	Communicate effectively with the engineering community and with society at large by prepare technical report and deliver presentation.
100003/CS622T.4-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work by prepare technical report and deliver presentation.
100003/CS622T.4-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by prepare technical report and deliver presentation.
100003/CS622T.4-PSO1	**MEDIUM**	Prepare a technical report and deliver presentation to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas.
100003/CS622T.4-PSO2	**MEDIUM**	To acquire programming efficiency by designing algorithms and applying standard practices in software project development and to prepare technical report and deliver presentation.
100003/CS622T.4-PSO3	**MEDIUM**	To apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs by preparing technical report and deliver presentation.
100003/CS622T.5-PO1	**HIGH**	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.5-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems by applying engineering and management principles to achieve the goal of the project.

100003/CS622T.5-PO3	**HIGH**	Apply engineering and management principles to achieve the goal of the project and to design solutions for complex engineering problems and design system components or processes that meet the specified needs.
100003/CS622T.5-PO4	**MEDIUM**	Apply engineering and management principles to achieve the goal of the project and use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/CS622T.5-PO5	**MEDIUM**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO6	**MEDIUM**	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities by applying engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO7	**MEDIUM**	Understand the impact of the professional engineering solutions in societal and environmental contexts, and apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO8	**HIGH**	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice and to use the engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO9	**MEDIUM**	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PSO1	**MEDIUM**	The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas. Apply engineering and management principles to achieve the goal of the project.

100003/CS622T.5-PSO2	**MEDIUM**	The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PSO3	**MEDIUM**	The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur and apply engineering and management principles to achieve the goal of the project.