**Sixth Semester Mini Project (Based on Internship)**

Project Title**: “Website to Monitor Temperature and Humidity using Node MCU Hardware”**

Name & Address of Company**: Cognifyz Technologies**

Duration of Internship**: 1 Month**

**Submitted By**

**Name of Project Members:**

**1.** AKASH SHATPALKAR

**2.** AKSHAT DESHBHRATAR

3. VALAY SUKHDEVE

4. ANIKET MARSHETTIWAR

5. POOJA MANDAL

**Under the guidance of**

**1.** Cognifyz Technologies leads the tech sector, delivering refined software solutions with advanced AI, ML, and data analytics tools

**2. Prof.Rohit Iyer, Intership Incharge**

****

**Department of Electronics and Telecommunication Engineering**

**Priyadarshini College of Engineering,**

**Nagpur – 440019**

**Session: 2023-24**

**Priyadarshini College of Engineering,**

**Department of Electronics and Telecommunication Engineering, Nagpur**

**CERTIFICATE**

This is to certify that the project entitled “**Website to Monitor Temperature and Humidity using Node MCU Hardware**” has been carried out by

**1.** AKASH SHATPALKAR

**2.** AKSHAT DESHBHRATAR

3. VALAY SUKHDEVE

4. ANIKET MARSHETTIWAR

5. POOJA MANDAL

under my guidance during the academic year 2023-24 is a bonafide work prepared by them.

This work fulfills the requirements relating the standard of work for the Sixth Semester of Bachelor of Engineering in Electronics and Telecommunication to be awarded by Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur.

Place: Nagpur

Date: 22/03/2024

|  |  |  |
| --- | --- | --- |
| **Prof.Rohit Iyer**  **Guide**  **(Company Consern)** |  | **Dr. V.K. Takasande**  **Head of Department** |

# Acknowledgement

First of all, I would like to thank **Cognifyz Technologies** for giving me the opportunity to do an internship with the organisation.

I would like to express my gratitude to Priyadarshini college of Engineering and RTMNU university for including internship program. as a credit course which has provided an opportunity to gain practical working experience in the organisation. My sincere gratitude to Dr. V. K. Taksande, HOD, dept. Of ETC and Honorable Dr. S.A. Dhale, Principal, PCE, Nagpur.

I would also like to thank our Internship coordinator Department of ETC for their Support and advices to complete Internship. I am extremely grateful for my department staff members and friends who helped me in successful completion of this internship.

**Index**

* **Abstract**
* **Introduction**
* **Software and Languages Required**
* **Aim & objectives**
* **Resources (About the company and software used).**
* **Code**
* **Result & Output**
* **Conclusion**
* **Future scope**
* **References**

# Abstract

This project focuses on the development of a website designed to monitor temperature and humidity in real-time using Node MCU hardware. Leveraging the Node MCU's built-in Wi-Fi capabilities, the system collects environmental data from connected sensors, such as DHT11 or DHT22, and transmits this information to a cloud-based server. The collected data is then stored in a database, enabling historical data analysis and long-term monitoring.

A user-friendly web interface presents the data, featuring real-time updates and graphical visualizations, making it accessible on various devices including desktops, tablets, and smartphones. The interface also supports user authentication to secure data access, and provides customizable alerts and notifications via email or SMS when the readings exceed predefined thresholds.

The project aims to deliver a scalable, maintainable, and energy-efficient monitoring solution that can be used in a variety of applications such as home automation, agricultural management, and environmental monitoring. Comprehensive documentation and tutorials are provided to facilitate the replication and educational use of the system, ensuring that users can easily set up and maintain their own monitoring systems. Through this project, users gain the ability to effectively track and manage environmental conditions, enhancing their control over various environments.

# Introduction

**Website to Monitor Temperature and Humidity** is a common task in various fields such as engineering, science, and everyday life. A temperature converter project aims to create a tool that allows users to easily convert temperature values between different units of measurement, such as Celsius, Fahrenheit, and Kelvin. This project provides a practical application for understanding programming concepts, particularly data input, processing, and output.

The temperature converter project typically involves creating a user interface where users can input a temperature value in one unit and select the desired unit for conversion. The program then performs the necessary calculations to convert the temperature and displays the result to the user.

Key components of a temperature converter project include:

1**. User Interface:** A graphical interface or command-line interface where users can input the temperature value and select the units for conversion. This interface should be user-friendly and intuitive, providing clear instructions and feedback to the user.

2. **Temperature Conversion Logic:** The core logic of the program involves converting the input temperature value from one unit to another. This may require mathematical formulas or algorithms to accurately convert between Celsius, Fahrenheit, and Kelvin, considering their different scales and relationships.

3. **Error Handling:** Proper error handling is essential to ensure the program can handle invalid inputs or unexpected situations gracefully. This includes validating user input, handling edge cases, and providing informative error messages to the user.

4. **Output Display:** After performing the temperature conversion, the program should display the result to the user in a clear and understandable format. This may include formatting the output with appropriate units and precision

# Software and Languages Requirement

1. VS CODE
2. JAVA SCRIPT
3. HTML and CSS
4. Arduino IDE
5. Pragramming language C

# AIM and Objective

**Aim:**

The primary aim of this project is to develop an efficient, real-time monitoring system that tracks environmental conditions such as temperature and humidity using Node MCU hardware. This system should provide accessible, accurate, and timely data through a user-friendly web interface.

**Objectives:**

1. Data Acquisition:

- Utilize Node MCU hardware to collect temperature and humidity data from connected sensors (e.g., DHT11 or DHT22 sensors). Ensure accurate and consistent readings by calibrating the sensors appropriately.

2. Data Transmission:

- Implement Wi-Fi connectivity through the Node MCU to transmit the collected data to a server or cloud service. Ensure reliable and secure data transmission using appropriate protocols (e.g., HTTP, MQTT).

3. Data Storage:

- Set up a database (e.g., MySQL, Firebase) to store the temperature and humidity data for historical analysis and reference. Implement efficient data management strategies to handle large volumes of data over time.

4. Web Interface Development:

- Design a responsive and intuitive web interface for users to view real-time and historical data. Include features such as graphical representations (charts/graphs) of data, alerts/notifications for threshold breaches, and data export options.

5. Real-Time Monitoring:

- Ensure the web interface updates in real-time or near-real-time as new data is received from the Node MCU. Implement features like auto-refresh and data caching to enhance user experience.

6. User Accessibility:

- Ensure the website is accessible on various devices (desktops, tablets, smartphones) and supports multiple web browsers. Implement user authentication and authorization mechanisms to secure sensitive data.

7. Alert and Notification System:

- Implement an alert system that notifies users via email, SMS, or push notifications when temperature or humidity readings exceed predefined thresholds. Allow users to customize alert settings according to their preferences.

8. Educational and Demonstrative Purposes:

- Provide clear documentation and tutorials on setting up the hardware and software components, aimed at educating users and enabling replication of the project.

9. Power Management:

- Optimize the power consumption of the Node MCU and connected sensors to ensure long-term, uninterrupted monitoring.

- Implement features like deep sleep mode for the Node MCU to conserve power when data is not being transmitted.

# Resources

* **HTML And CSS :**



HTML (Hypertext Markup Language) is the standard markup language used to create the structure and content of web pages. It consists of a series of elements, represented by tags, which define the various components of a webpage such as headings, paragraphs, images, links, forms, and more. HTML provides the foundation for organizing and presenting information on the web.

CSS (Cascading Style Sheets) is a style sheet language used to define the visual presentation of HTML documents. It allows developers to control the layout, appearance, and styling of HTML elements, including aspects such as colors, fonts, spacing, borders, and backgrounds. CSS separates the content of a webpage from its presentation, enabling consistent styling across multiple pages and facilitating responsive design for different screen sizes and devices.

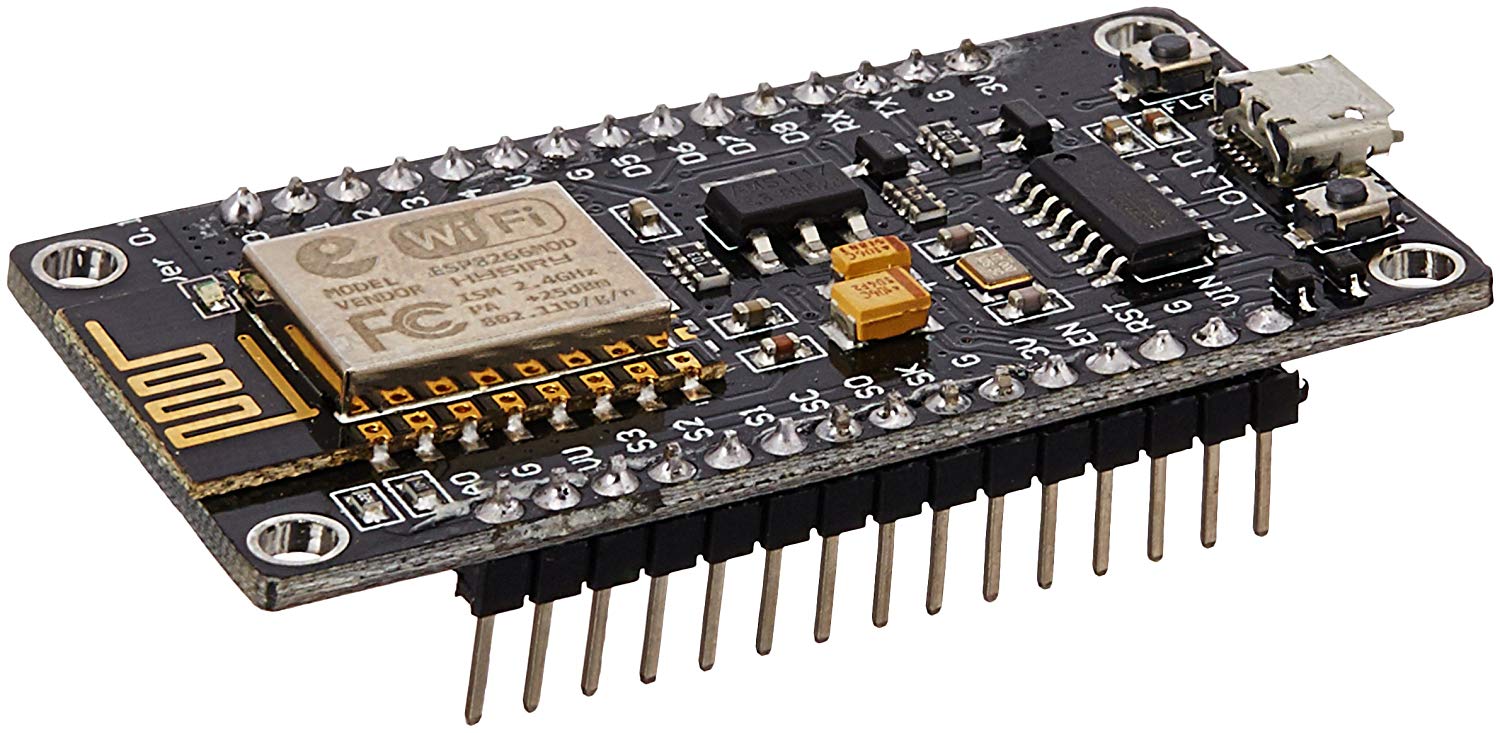
Together, HTML and CSS form the core technologies of web development, with HTML defining the structure and content of web pages and CSS determining their visual appearance and layout. By combining HTML for structure and CSS for styling, developers can create visually appealing, well-organized, and responsive web experiences.

* **JAVA SCRIPT :**

****

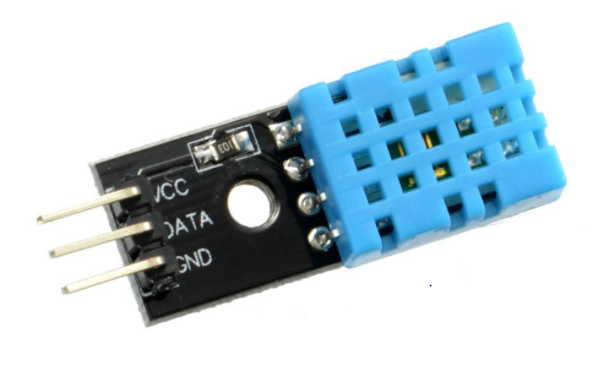
JavaScript is a versatile scripting language primarily used for web development, enabling interactive features and dynamic content on websites. It facilitates event-driven programming, allowing developers to define actions in response to user interactions or other events, and provides powerful APIs for manipulating the Document Object Model (DOM) to dynamically update webpage content

* **Node MCU**



NodeMCU is an open-source LUA based firmware developed for the ESP8266 wifi chip. By exploring functionality with the ESP8266 chip, NodeMCU firmware comes with the ESP8266 Development board/kit i.e. NodeMCU Development board

* DHT 11



DHT11 Temperature and Humidity Sensor can measure temperature and humidity with calibrated digital signal output. Its is highly reliable and excellent long-term stability. This sensor includes an element and a sensor for wet NTC temperature measuring. It is excellent quality, fast response,anti-interference with high performance

**CODE :**

<! // Import required libraries #include <Arduino.h> #include <ESP8266WiFi.h> #include <Hash.h>

#include <ESPAsyncTCP.h> #include <ESPAsyncWebServer.h> #include <Adafruit\_Sensor.h> #include <DHT.h>

// Replace with your network credentials const char\* ssid = "LG G8x";

const char\* password = "12345678990";

#define DHTPIN D5 // Digital pin D1 connected to the DHT sensor

// Uncomment the type of sensor in use:

#define DHTTYPE DHT11 // DHT 11

//#define DHTTYPE DHT22 // DHT 22 (AM2302)

//#define DHTTYPE DHT21 // DHT 21 (AM2301) DHT dht(DHTPIN, DHTTYPE);

// current temperature & humidity, updated in loop() float t = 0.0;

float h = 0.0;

// Create AsyncWebServer object on port 80

AsyncWebServer server(80);

// Generally, you should use "unsigned long" for variables that hold time

// The value will quickly become too large for an int to store

unsigned long previousMillis = 0; // will store last time DHT was updated

// Updates DHT readings every 10 seconds const long interval = 10000;

const char index\_html[] PROGMEM = R"rawliteral(

<!DOCTYPE HTML>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" integrity="sha384-

fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr" crossorigin="anonymous">

<style>

html {

font-family: Arial; display: inline-block; margin: 0px auto; text-align: center;

}

h1 {

font-size: 3.0rem; background-color: blue; color: white;

padding: 10px;

}

h2 {

font-size: 2.5rem;

}

p {

font-size: 2.0rem;

}

.units {

font-size: 1.2rem;

}

.dht-labels {

font-size: 1.5rem; vertical-align: middle; padding-bottom: 15px;

}

.temperature-section { background-color: orange; padding: 10px;

}

.humidity-section { background-color: lightblue; padding: 10px;

}

</style>

</head>

<body>

<h1>PRIYADARSHINI COLLEGE OF ENGINEERING</h1>

<h2>Department Of Electronics & Telecommunication</h2>

<h3>Website To Monitor Temperature & Humidity Using Nodemcu Hardware</h3>

<div class="temperature-section">

<p>

<i class="fas fa-thermometer-half" style="color:#059e8a;"></i>

<span class="dht-labels">Temperature</span>

<span id="temperature">%TEMPERATURE%</span>

<sup class="units">°C</sup>

</p>

</div>

<div class="humidity-section">

<p>

<i class="fas fa-tint" style="color:#00add6;"></i>

<span class="dht-labels">Humidity</span>

<span id="humidity">%HUMIDITY%</span>

<sup class="units">%</sup>

</p>

</div>

<p>

<script src="https://apps.elfsight.com/p/platform.js" defer></script>

<div class="elfsight-app-65e091b0-d33c-4191-81f3-be77c921660a"></div>

</p>

</body>

<script> setInterval(function () {

var xhttp = new XMLHttpRequest(); xhttp.onreadystatechange = function () {

if (this.readyState == 4 && this.status == 200) { document.getElementById("temperature").innerHTML = this.responseText;

}

};

xhttp.open("GET", "/temperature", true); xhttp.send();

}, 10000);

setInterval(function () {

var xhttp = new XMLHttpRequest(); xhttp.onreadystatechange = function () {

if (this.readyState == 4 && this.status == 200) { document.getElementById("humidity").innerHTML = this.responseText;

}

};

xhttp.open("GET", "/humidity", true); xhttp.send();

}, 10000);

</script>

</html>)rawliteral";

// Replaces placeholder with DHT values String processor(const String& var){

//Serial.println(var);

if(var == "TEMPERATURE"){

return String(t);

}

else if(var == "HUMIDITY"){ return String(h);

}

return String();

}

void setup(){

// Serial port for debugging purposes Serial.begin(9600);

dht.begin();

// Connect to Wi-Fi WiFi.begin(ssid, password);

Serial.println("Connecting to WiFi");

while (WiFi.status() != WL\_CONNECTED) { delay(500);

Serial.println(".");

}

// Print ESP8266 Local IP Address Serial.println(WiFi.localIP());

// Start server server.begin();

// Route for root / web page

server.on("/", HTTP\_GET, [](AsyncWebServerRequest \*request){ request->send\_P(200, "text/html", index\_html, processor);

});

server.on("/temperature", HTTP\_GET, [](AsyncWebServerRequest \*request){ request->send\_P(200, "text/plain", String(t).c\_str());

});

server.on("/humidity", HTTP\_GET, [](AsyncWebServerRequest \*request){ request->send\_P(200, "text/plain", String(h).c\_str());

});

}

void loop(){

unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= interval) {

// save the last time you updated the DHT values previousMillis = currentMillis;

// Read temperature as Celsius (the default) float newT = dht.readTemperature();

// Read temperature as Fahrenheit (isFahrenheit = true)

//float newT = dht.readTemperature(true);

// if temperature read failed, don't change t value

if (isnan(newT)) Serial.println("Failed to read from DHT sensor!");

}

else {

t = newT; Serial.println(t);

}

// Read Humidity

float newH = dht.readHumidity();

// if humidity read failed, don't change h value if (isnan(newH)) {

Serial.println("Failed to read from DHT sensor!");

}

else {

h = newH;

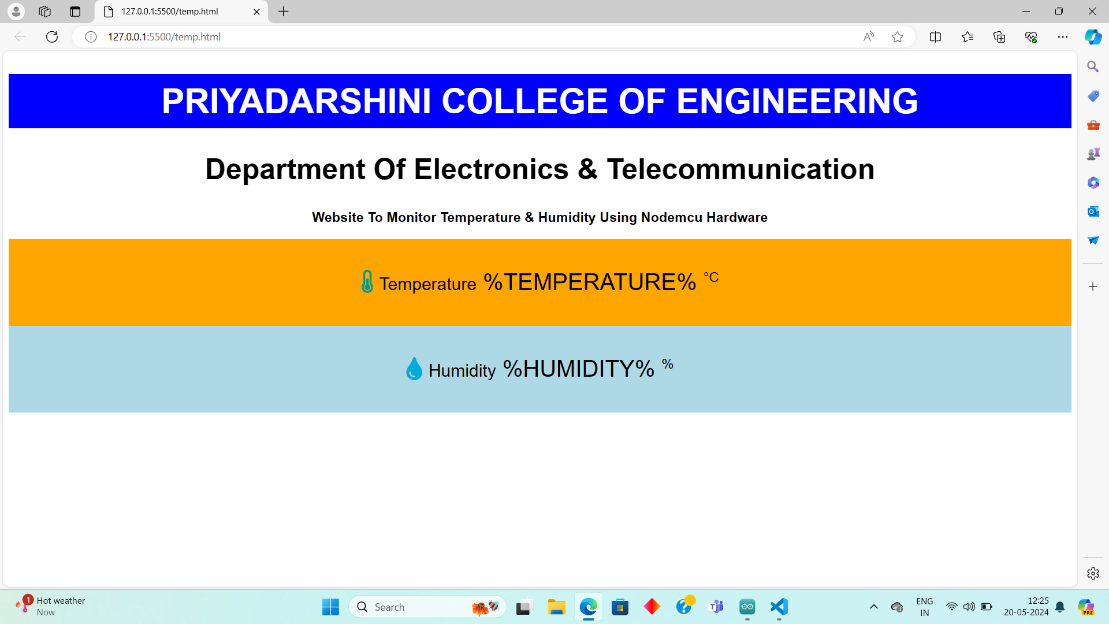
Serial.println(h);

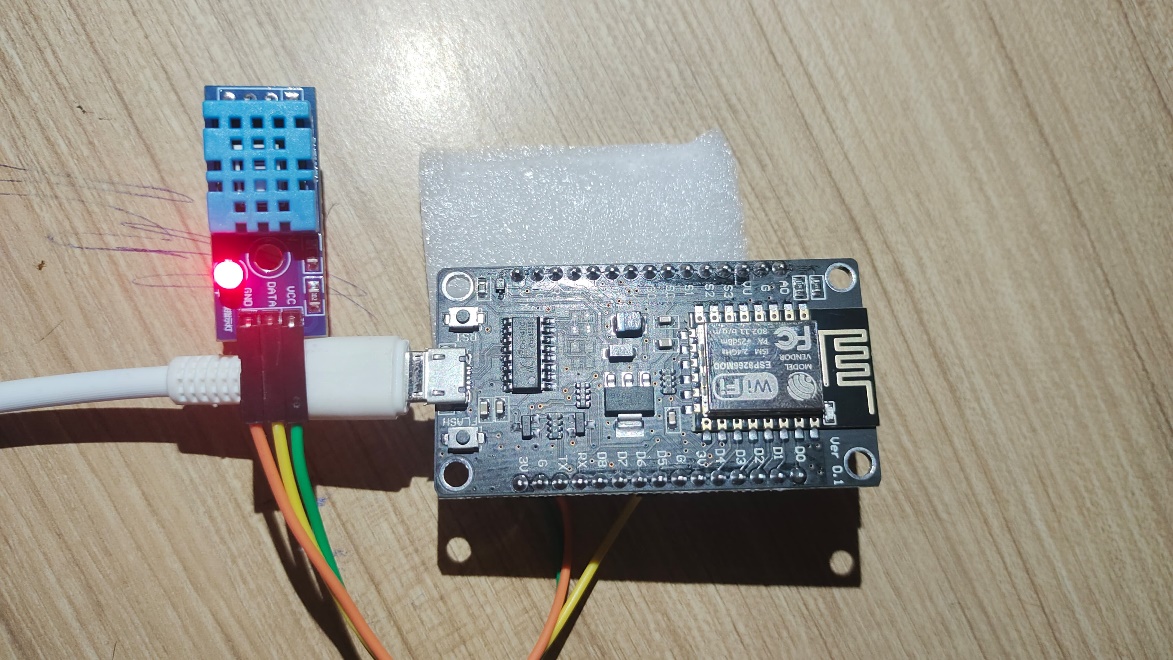
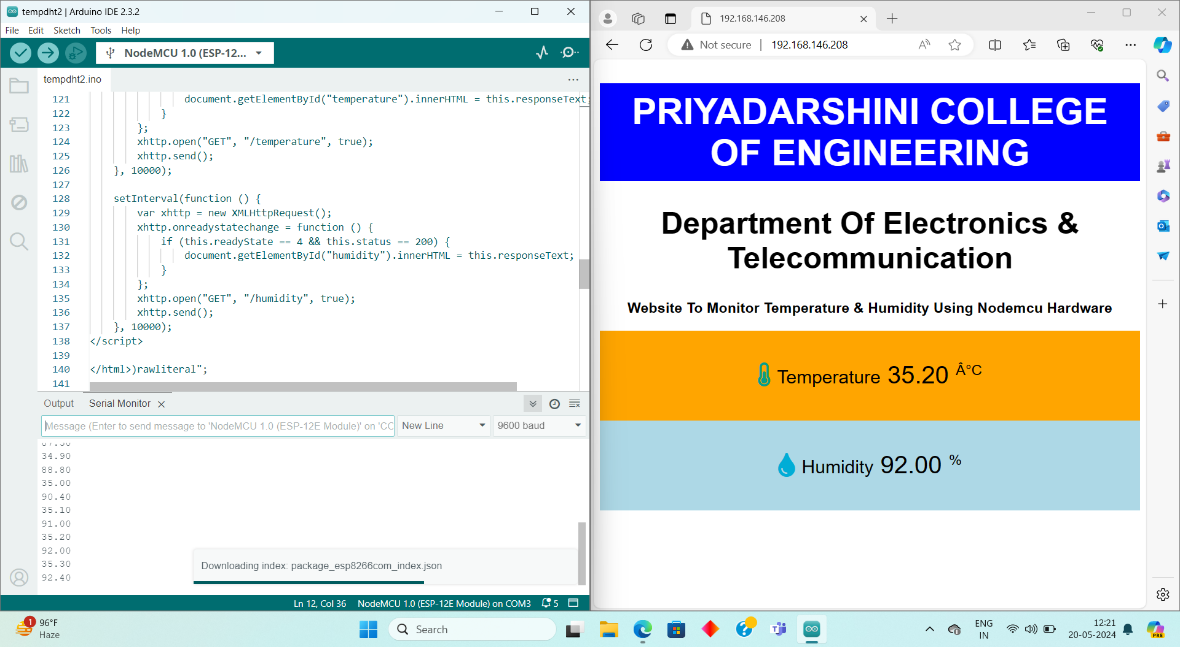
}

}

}

* **Output :**





**Result :**

In short, the result of the Temperature Converter Project is a user-friendly web application featuring accurate temperature conversion between Celsius, Fahrenheit, and Kelvin units. It offers a clean interface, real-time output display, robust error handling, responsive design, and cross-browser compatibility, providing users with a convenient tool for temperature conversions across various devices and browsers.

**Conclusion :**

the Temperature Converter Project stands as a testament to the effectiveness of modern web development techniques in providing practical solutions. Through meticulous design and implementation, the project has yielded a user-friendly web application capable of seamlessly converting temperature values across different units.

By prioritizing intuitive interface design, the project ensures that users of all levels of technical proficiency can easily navigate and utilize the application. The real-time output display enhances user experience by providing immediate feedback, while robust error handling mechanisms instill confidence in the reliability of the conversion process.

**Future Scope :**

Looking ahead, the Temperature Converter Project holds significant potential for future expansion and enhancement. By incorporating additional temperature units, advanced features, and customization options, the application can offer a broader range of functionality to cater to diverse user needs. Localization and internationalization efforts can extend its reach to a global audience, while integration with external APIs and databases can ensure real-time accuracy and relevance. Furthermore, exploring mobile application development and fostering user engagement through feedback mechanisms and social media integration will further solidify the project's position as a valuable tool in temperature management. With a commitment to innovation and responsiveness to user feedback, the Temperature Converter Project is poised to evolve into a comprehensive temperature management platform that addresses the evolving needs of users across various industries and domains.

**References:**

When providing references for a project like a temperature converter, it's essential to cite any sources you used for information, algorithms, or code snippets. Here's a generic list of potential references for a temperature converter project:

Official Documentation: Refer to the official documentation of web development technologies used in the project, such as HTML, CSS, JavaScript, and any frameworks or libraries utilized (e.g., jQuery, Bootstrap).

Programming Tutorials and Guides: Online tutorials, guides, and articles from reputable sources like MDN Web Docs, W3Schools, and CSS-Tricks can provide valuable insights and code examples for implementing specific features or functionalities. Textbooks and Academic Papers: If you referenced specific algorithms, theories, or concepts related to temperature conversion, cite relevant textbooks or academic papers in physics, thermodynamics, or related fields.