SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING

Fall Semester 2023 – 2024

LAB ASSESSMENT – V

IoT Wireless Weather Monitoring Station

Programme Name and Branch: B.Tech (IT)

Course Name & Code: Embedded Systems and IoT Lab & BITE403P

Faculty Name: Prof. N. Bharani Dharan

Project Members:

Name	Registration Number
Subalakshmi V	21BIT0058
Vishwajith Prabhakar	21BIT0444
Kadam Shivanjali	21BIT0456
S Kavya Sree	21BIT0483
Nithyasri K	21BIT0596
Ishika Khandelwal	21BIT0657

INDEX

Contents	Page no.

Introduction:

The IoT Weather Station project aims to measure and monitor humidity, temperature, and pressure using NodeMCU (ESP8266) microcontroller, DHT11 sensor for humidity and temperature, and BMP180 sensor for pressure. The collected data is then displayed on a web server, allowing users to monitor the weather conditions remotely over the internet.

Components Used:

1. NodeMCU (ESP8266) - microcontroller board with built-in Wi-Fi capabilities
2. DHT11 sensor - for measuring humidity and temperature
3. BMP180 sensor - for measuring atmospheric pressure
4. Breadboard - for circuit connections
5. Jumper wires - for interconnecting components

Need of the Project:

- The IoT Weather Station project addresses the need for monitoring and understanding weather conditions in a convenient and accessible manner.
- Traditional weather monitoring systems often require specialized equipment and are limited to specific locations or regions.
- By utilizing IoT technology and the power of NodeMCU microcontroller, this project enables users to monitor humidity, temperature, and pressure remotely over the internet.
- This level of accessibility is particularly valuable for various applications, including agriculture, home automation, and research. The project can be used by the following people:
    - Farmers can make informed decisions regarding irrigation and crop management based on real-time weather data.
    - Homeowners can optimize energy usage by adjusting heating and cooling systems based on current temperature and humidity levels.
    - Researchers can gather valuable environmental data for analysis and forecasting.
- Overall, the IoT Weather Station project fulfills the need for a user-friendly, remote weather monitoring system that can enhance decision-making and improve understanding of weather patterns

Novelty:

1. Remote Weather Monitoring:

The project allows users to remotely monitor weather conditions from anywhere with an internet connection. By accessing the web server hosted on the NodeMCU, users can obtain real-time data on humidity, temperature, and pressure.

2. Integration of Multiple Sensors:

The project combines the DHT11 sensor for humidity and temperature measurement and the BMP180 sensor for pressure measurement. Integrating multiple sensors enables a more comprehensive understanding of environmental conditions.

3. Web-Based User Interface:

The project presents weather data through a user-friendly web interface. Users can access the data by simply entering the IP address of the NodeMCU in a web browser, eliminating the need for dedicated software or hardware.

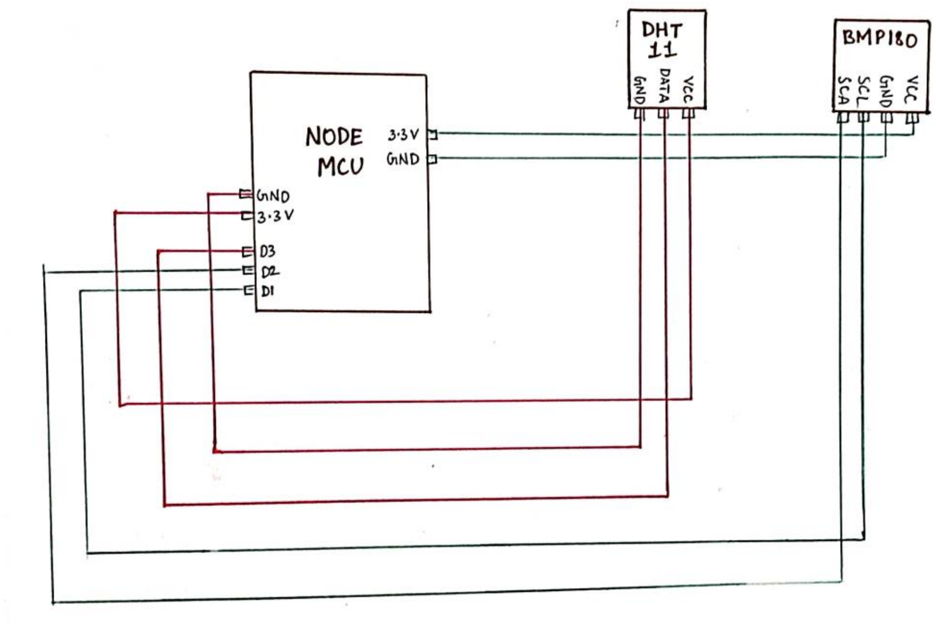
4. Dynamic HTML Response:

The project generates HTML responses dynamically, incorporating sensor readings into the HTML code. This dynamic approach allows for real-time updates of weather data on the web page without requiring manual refreshes.

5. Application Potential:

The IoT Weather Station project has broad application potential. It can be used in agriculture for optimizing irrigation and crop management, in home automation systems for controlling temperature and humidity settings, and in research for collecting environmental data for analysis and forecasting.

Circuit Diagram:



Implementation:

1. Hardware Setup:

The components are connected as follows:

1. DHT11 sensor is connected to digital pin 0 (DHTPIN) of NodeMCU.
2. BMP180 sensor is connected to the I2C pins (SCL and SDA) of NodeMCU.
3. NodeMCU is connected to the internet via Wi-Fi.

2. Software Setup:

The project requires the following libraries to be included in the Arduino IDE:

1. ESP8266WiFi: For connecting to Wi-Fi network
2. DHT: For interacting with DHT11 sensor
3. SFE_BMP180: For interacting with BMP180 sensor
4. Wire: For I2C communication with BMP180 sensor

3. Connections:

These are the connections to be made:

a) NodeMCU GND to BMP180 GND
b) NodeMCU 3.3V to BMP180 VCC
c) NodeMCU D1 Pin to BMP180 SCL
d) NodeMCU D2 Pin to BMP180 SCA
e) NodeMCU GND to DHT11 GND
f) NodeMCU 3.3V to DHT11 VCC
g) NodeMCU D3 Pin to DHT11 Data

4. Configuration:

1. Set your Wi-Fi credentials by modifying the "ssid" and "password" variables in the code.
2. Adjust the "ALTITUDE" constant to match your location's altitude if necessary.

5. Upload and Run:

1. Select the appropriate NodeMCU board and COM port in the Arduino IDE.
2. Upload the code to the NodeMCU board.
3. Open the Serial Monitor to monitor the initialization process and check for any errors.
4. Once the code is successfully uploaded, disconnect the NodeMCU from the computer and power it with an external power source.

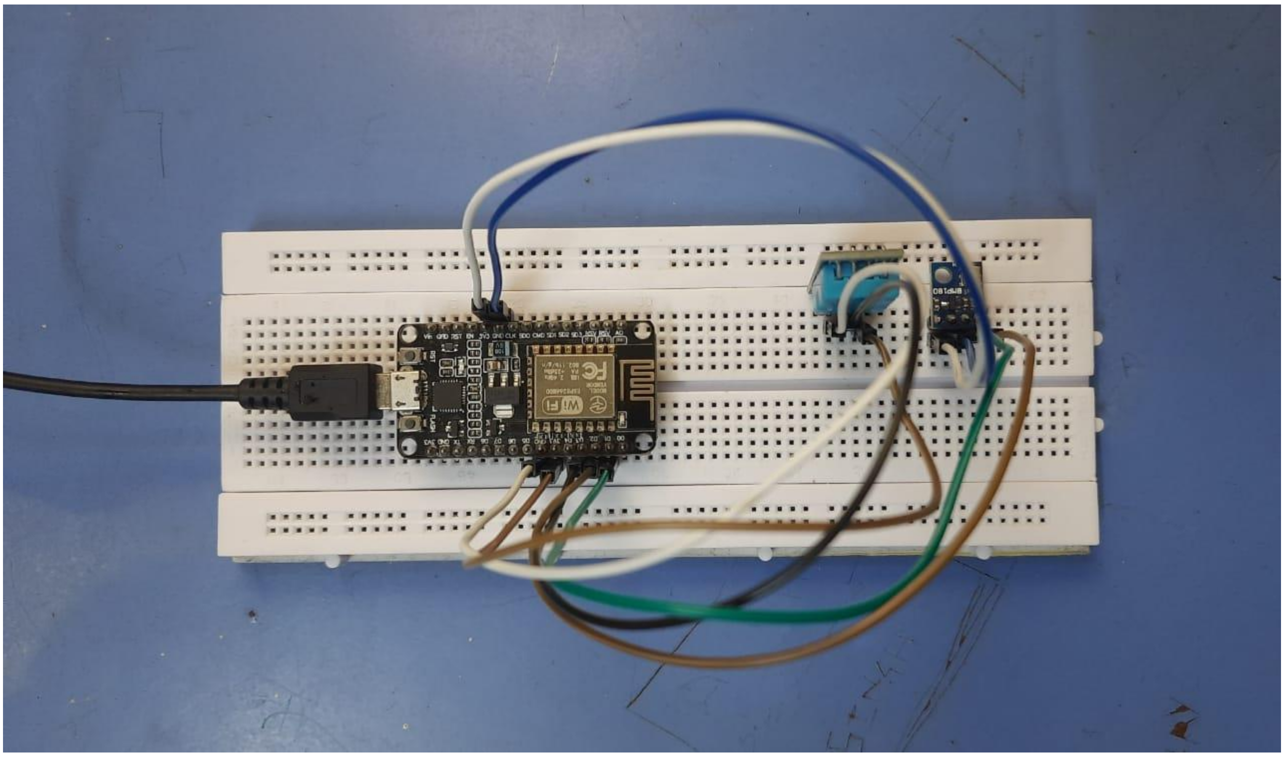
6. Monitor the Weather Data:

1. After the NodeMCU is powered up, it will connect to the specified Wi-Fi network.
2. Once connected, it will start a web server on port 80 and display its IP address on the Serial Monitor.
3. Connect any device (e.g., smartphone, computer) to the same Wi-Fi network and enter the NodeMCU's IP address in a web browser.
4. The web page will display the current humidity, temperature (in Celsius and Fahrenheit), and pressure readings.

7. Remote Monitoring:

1. As long as the NodeMCU is powered and connected to the internet, the weather data can be accessed from anywhere by entering its IP address in a web browser.
2. The web page will automatically refresh every 10 seconds to provide updated weather information.

Photograph of the Circuit:



Key Learnings from the Project:

1. IoT Integration:

The project provides hands-on experience in integrating IoT capabilities into a real-world application. It demonstrates how to leverage a microcontroller (NodeMCU) with built-in Wi-Fi to connect to the internet and transmit sensor data.

2. Sensor Interfacing:

The project involves interfacing with sensors such as the DHT11 (humidity and temperature) and BMP180 (pressure). It teaches the process of initializing, reading data from sensors, and utilizing the data for further processing and display.

3. Web Server Development:

By setting up a web server on the NodeMCU, the project showcases the development of a basic web server that serves HTML content. It covers concepts such as HTTP response handling, generating dynamic HTML, and client-server communication.

4. Data Visualization:

The project demonstrates how to present sensor data in a user-friendly format using HTML and CSS. It highlights the importance of data visualization for easy interpretation and provides a foundation for creating interactive and visually appealing web interfaces.

5. Remote Monitoring:

The ability to access weather data remotely over the internet is a significant learning from the project. It illustrates the power of IoT in enabling real-time monitoring and control from anywhere, allowing users to make informed decisions based on the collected data.

6. Troubleshooting and Debugging:

During the implementation process, troubleshooting and debugging skills are honed. It is crucial to identify and resolve issues related to hardware connections, software code, and network connectivity to ensure the smooth operation of the IoT Weather Station.

7. Practical Applications:

The project offers insights into the practical applications of IoT in various domains. It highlights the potential benefits of remotely monitoring weather conditions for agriculture, home automation, research, and other areas where environmental data plays a crucial role.

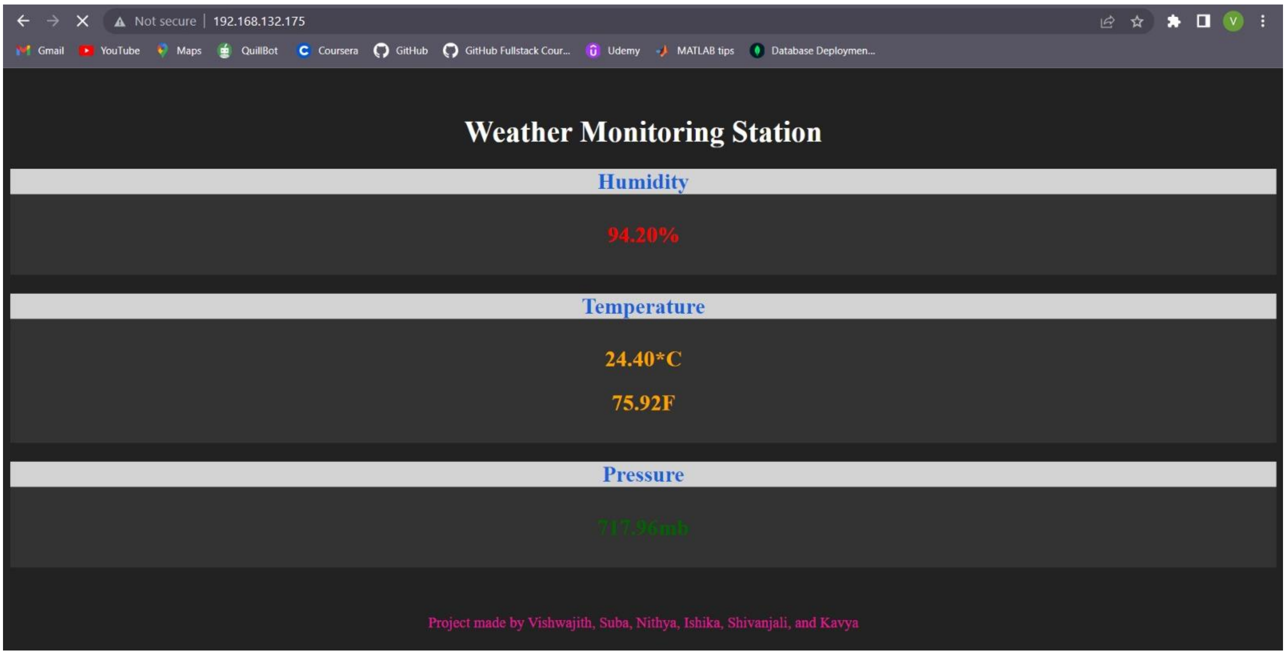
Code:

```
#include <ESP8266WiFi.h>
#include "DHT.h"
#include <SFE_BMP180.h>
#include <Wire.h>

#define ALTITUDE 239.0
#define DHTPIN 0
#define DHTTYPE DHT11

SFE_BMP180 pressure;
DHT dht(DHTPIN, DHTTYPE);

const char* ssid     = "Chinu"; // Your ssid
const char* password = "chinu3101"; // Your Password

char status;
double T, P, p0, a;

WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(100);

  dht.begin();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

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

  Serial.println("");
  Serial.println("WiFi is connected");
  server.begin();
  Serial.println("Server started");
  Serial.println(WiFi.localIP());

  if (pressure.begin())
    Serial.println("BMP180 init success");
  else {
    Serial.println("BMP180 init fail\n\n");
    while (1);
  }

  delay(1000);
```

```
}

void loop() {
  status = pressure.getPressure(P, T);

  if (status != 0) {
    p0 = pressure.sealevel(P, ALTITUDE);
    Serial.print("relative (sea-level) pressure: ");
    Serial.print(p0, 2);
    Serial.print(" mb, ");
  }

  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);

  WiFiClient client = server.available();
  client.println("HTTP/1.1 200 OK");
  client.println("Content-Type: text/html");
  client.println("Connection: close");
  client.println("Refresh: 10");
  client.println();
  client.println("<!DOCTYPE HTML>");
  client.println("<html>");
  client.println("<head>");
  client.println("<meta name=\"viewport\" content=\"width=device-width, initial-
scale=1\">");
  client.println("<link rel=\"stylesheet\" href=\"bootstrap.min.css\">");
  client.println("<script src=\"jquery.min.js\"></script>");
  client.println("<script src=\"bootstrap.min.js\"></script>");
  client.println("<style>html { font-family: Cairo; display: block; margin: 0px
auto; text-align: center;color: #333333; background-color: #222222;}");
  client.println("body{margin-top: 50px;}");
  client.println(".card {border: none; background-color: #333333; color: #ffffff;
margin-bottom: 20px;}");
  client.println(".card-header {background-color: lightgrey; color: #1a5fd7; font-
size: 24px; font-weight: bold;}");
  client.println(".card-body {padding: 10px;}");
  client.println(".humidity {color: #ff0000;}");
  client.println(".temperature {color: #ffa500;}");
  client.println(".pressure {color: #006400;}");
  client.println(".title {color: #ffffff;}");
  client.println(".footer {color: #f612a2; margin-top: 50px;}");
  client.println("</style>");
  client.println("</head>");
  client.println("<body>");
  client.println("<div class=\"container\">");
```

```
  client.println("<h1 class=\"title\">Weather Monitoring Station</h1>");
  client.println("<div class=\"card-deck\">");
  client.println("<div class=\"card col-xs-12 col-sm-4\">");
  client.println("<div class=\"card-header\">Humidity</div>");
  client.println("<div class=\"card-body\">");
  client.print("<h2 class=\"card-title humidity\">");
  client.print(h);
  client.println("%</h2>");
  client.println("</div>");
  client.println("</div>");
  client.println("<div class=\"card col-xs-12 col-sm-4\">");
  client.println("<div class=\"card-header\">Temperature</div>");
  client.println("<div class=\"card-body\">");
  client.print("<h2 class=\"card-title temperature\">");
  client.print(t);
  client.println("*C</h2>");
  client.print("<h2 class=\"card-title temperature\">");
  client.print(f);
  client.println("F</h2>");
  client.println("</div>");
  client.println("</div>");
  client.println("<div class=\"card col-xs-12 col-sm-4\">");
  client.println("<div class=\"card-header\">Pressure</div>");
  client.println("<div class=\"card-body\">");
  client.print("<h2 class=\"card-title pressure\">");
  client.print(p0, 2);
  client.println("mb</h2>");
  client.println("</div>");
  client.println("</div>");
  client.println("</div>");
  client.println("<div class=\"footer\">Project made by Vishwajith, Suba, Nithya,
Ishika, Shivanjali, and Kavya</div>");
  client.println("</div>");
  client.println("</body>");
  client.println("</html>");

  delay(4000);
}
```

Output:



Video Link:

Here is the video link that shows the demonstration of the project:

https://youtu.be/h-9IhD0xdVY

Conclusion:

The IoT Weather Station project successfully measures and monitors humidity, temperature, and pressure using the DHT11 and BMP180 sensors. The data is displayed on a web server, allowing users to access and monitor the weather conditions remotely via the internet. This project demonstrates the capabilities of IoT and provides a foundation for further enhancements, such as data logging, alerts, and integration with other IoT platforms.