IOT DOMAIN ANALYST LAB

TASK 3

NAME: SREENIVASAN S

REG NO: 21BEC0256

SLOT: L3+L4

FACULTY: SUJATHA R

DHT11 Web Server with ESP32 (Arduino IDE)

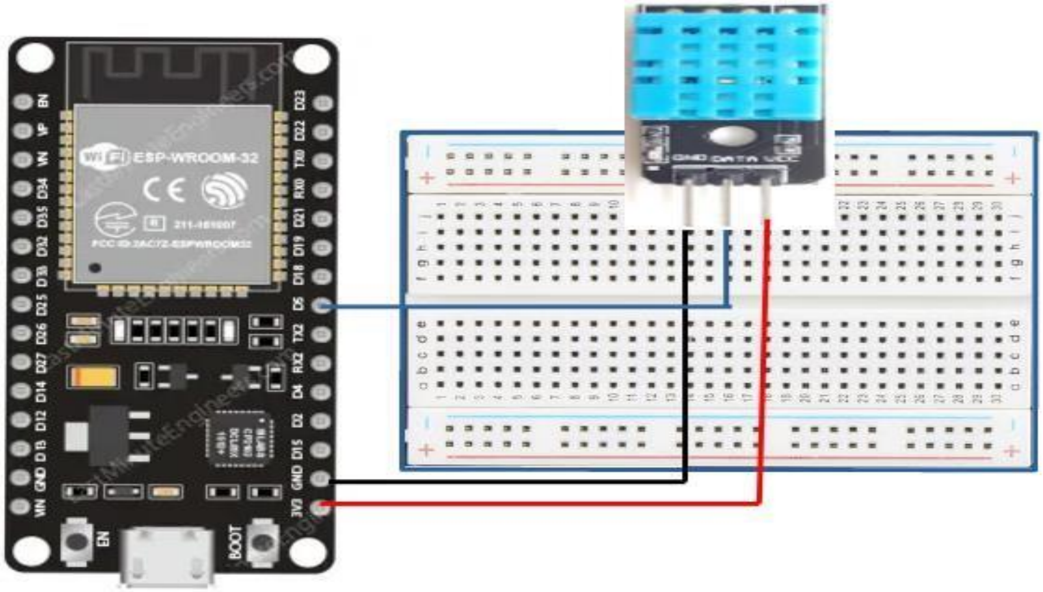
Aim

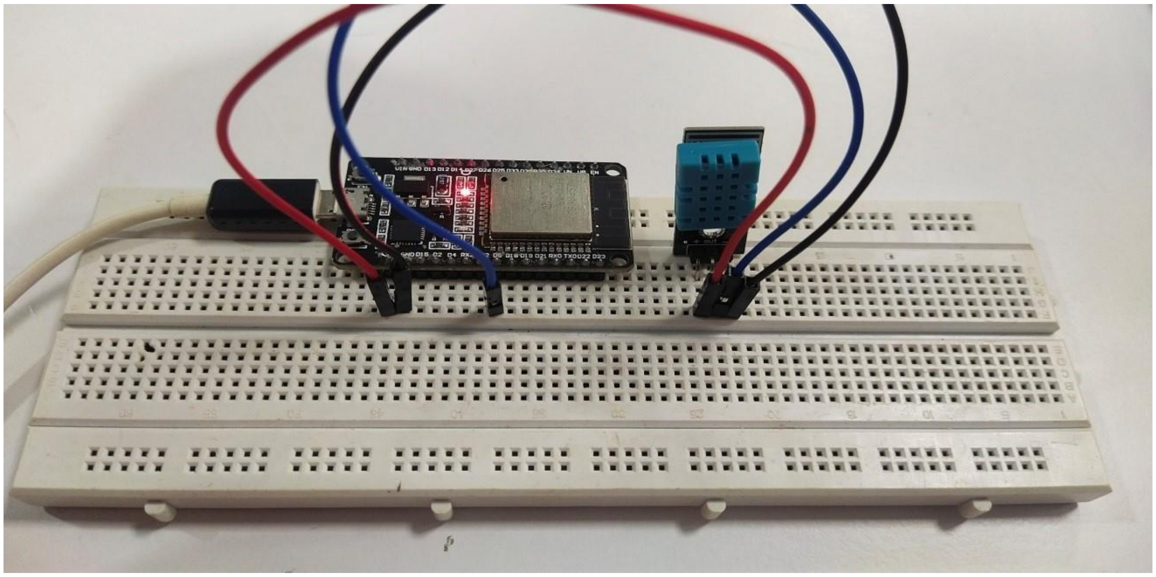
To develop a system that displays values from DHT11 sensor using a webserver with ESP32 as a control device in station mode, which connects to an existing Wi-Fi network.

Apparatus Required

ESP32
DHT 11 sensor
Micro USB cable
Jumper Wires
Breadboard
Arduino ide software

Circuit Diagram





Procedure

1. Acquire the necessary materials for the project, which encompass an ESP32 board, a DHT11/DHT22 sensor, a USB cable, and a computer equipped with the Arduino IDE.
2. Follow the guidelines outlined in the sensor's datasheet to properly link the DHT11/DHT22 sensor to the ESP32 board.
3. Establish a connection between the ESP32 board and the computer by utilizing the USB cable.
4. Launch the Arduino IDE and initiate a new sketch, incorporating the essential libraries for ESP32, DHT sensor, Wi-Fi, and Webserver.

5. Specify the Wi-Fi network credentials within the sketch, which include the SSID and password.
6. Transmit the sketch to the ESP32 board by employing the upload function within the Arduino IDE.
7. Detach the ESP32 board from the computer and supply power to it via an external power source.
8. Join a device to the Wi-Fi network and access the ESP32's IP address through a web browser to monitor the temperature and relative humidity readings.

Code :

```
#include "WiFi.h"

#include "ESPAsyncWebServer.h"

#include <Adafruit_Sensor.h>

#include <DHT.h>

// Replace with your network credentials

const char* ssid = "AMC"; const char*

password = "vit@12345";

#define DHTPIN D5     // Digital pin 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);

// Create AsyncWebServer object on port 80

AsyncWebServer server(80);

String readDHTTemperature() {

  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

  // Read temperature as Celsius (the default)

float t = dht.readTemperature();

  // Read temperature as Fahrenheit (isFahrenheit = true)

  //float t = dht.readTemperature(true);
  // Check if any reads failed and exit early (to try again).
if (isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
return "--";
  }
else {
    Serial.println(t);
return String(t);   }

}
String readDHTHumidity() {
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
float h = dht.readHumidity();   if (isnan(h)) {
    Serial.println("Failed to read from DHT sensor!");
return "--";
  }
else {
    Serial.println(h);
return String(h);
  }
} 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="sha384fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr"
crossorigin="anonymous">
```

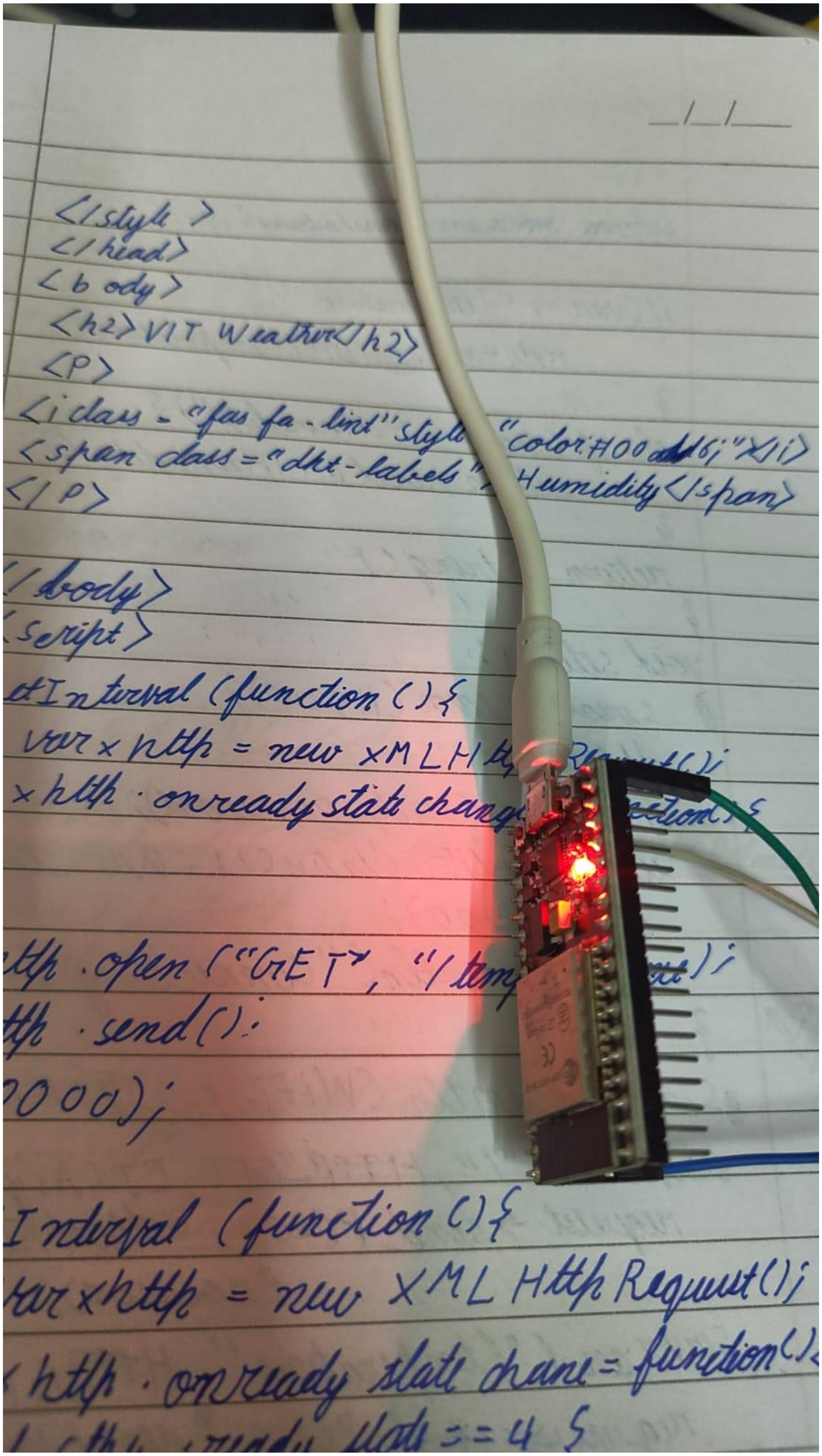
```
  <style>     html {
font-family: Arial;
display: inline-block;
margin: 0px auto;
text-align: center;

    }
    h2 { font-size: 3.0rem; }
p { font-size: 3.0rem; }
    .units { font-size: 1.2rem; }
    .dht-labels{       font-
size: 1.5rem;       vertical-
align:middle;       padding-
bottom: 15px;
    }
  </style>
</head>
<body>
  <h2>VIT Weather</h2>
  <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">&deg;C</sup>
  </p>
  <p>
    <i class="fas fa-tint" style="color:#00add6;"></i>
    <span class="dht-labels">Humidity</span>
    <span id="humidity">%HUMIDITY%</span>
    <sup class="units">&percnt;</sup>
  </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 readDHTTemperature();

  }

  else if(var == "HUMIDITY"){

return readDHTHumidity();

  }   return

String();

} void

setup(){

  // Serial port for debugging purposes

Serial.begin(115200);   dht.begin();

  // Connect to Wi-Fi

  WiFi.begin(ssid, password);   while

(WiFi.status() != WL_CONNECTED) {

    delay(1000);

    Serial.println("Connecting to WiFi..");

  }

  // Print ESP32 Local IP Address

  Serial.println(WiFi.localIP());   // 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", readDHTTemperature().c_str());

  });
  server.on("/humidity", HTTP_GET, [](AsyncWebServerRequest *request){     request-
>send_P(200, "text/plain", readDHTHumidity().c_str());
  });
  // Start server
server.begin();
} void
loop(){
}
```

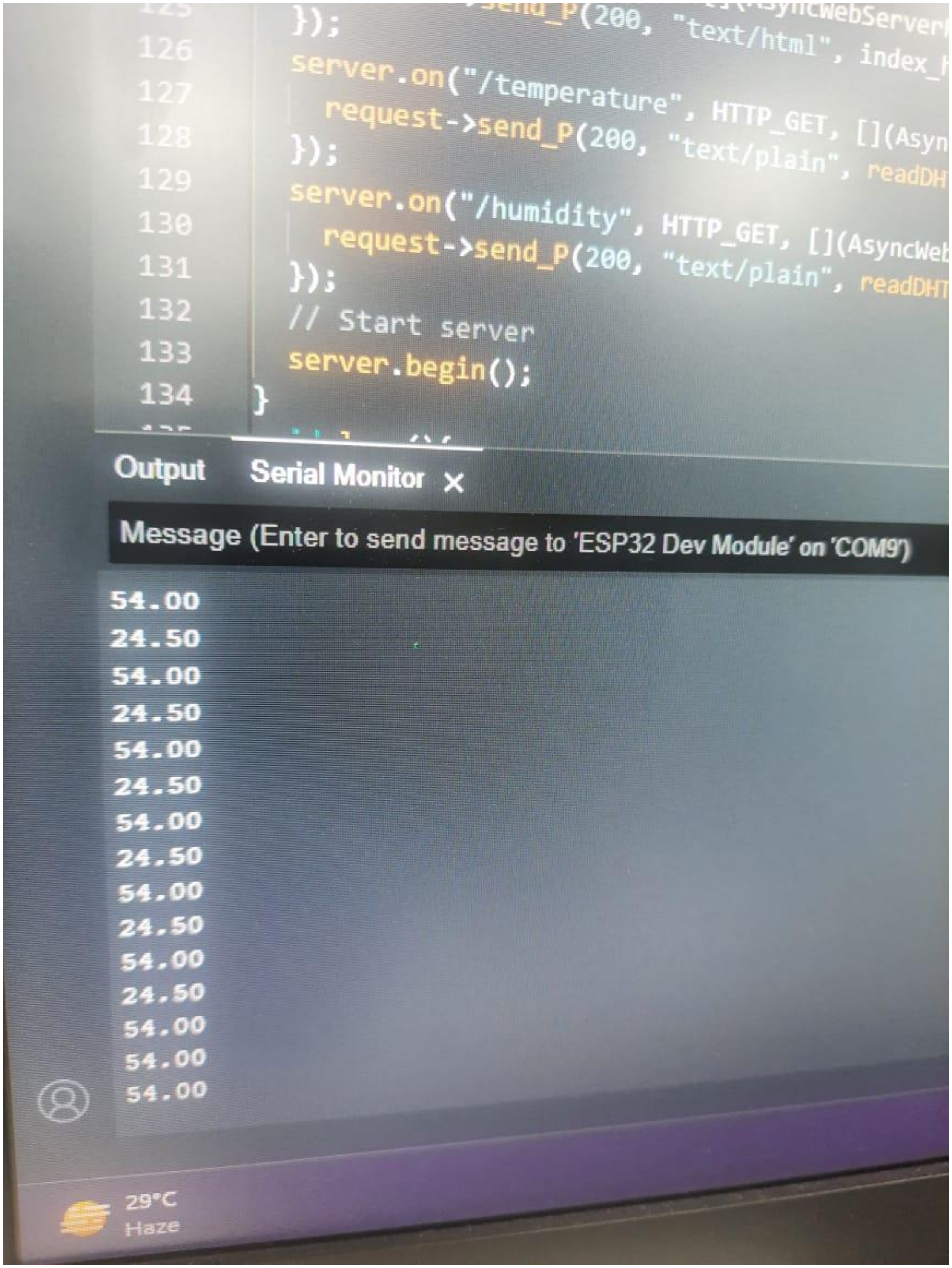
PROCESS AND STEPS TO DO IN THE PC:

1. Open Arduino ide software Type the code Board Manager Type ESP32 Dev Module
2. Select ESP32 Dev Module -> Select Port
3. Change the SSID and Password ->Verify and Upload
4. Serial Monitor ->Change the baud rate ->Copy the IP Address
5. Open Google Chrome Type or Paste the IP Address and Enter
6. View the temperature and humidity values in web browser.

LAB TAKEN SCREENSHOTS:



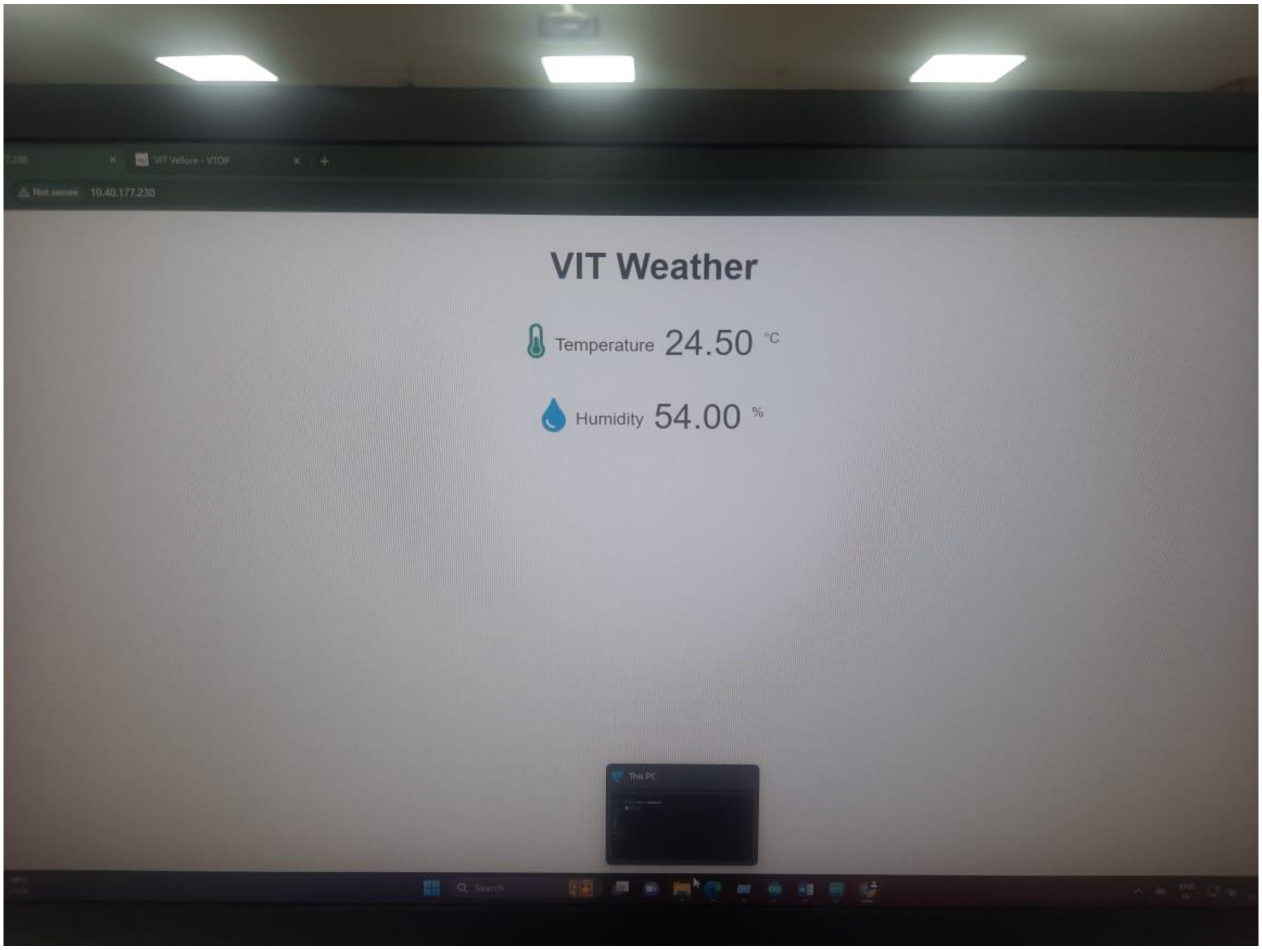
```
</style >
</head>
<b ody>
<h2>VIT Weather</h2>
<P>
<i class = "fas fa-lint" style "color:#006;"></i>
<span class="dht-labels" Humidity</span>
</P>

</body>
<script>
etInterval (function (){
var xhttp = new xMLHttp
xhtth . onready state chang ction(){


htth .open ("GET", "/temp e");
tth .send():
000);

Interval (function (){
var xhtth = new XML Http Request();
x http . onready state chane = function()
ready state == 4 {
```



```
    request->send_P(200, "text/html", index_H
  });
  server.on("/temperature", HTTP_GET, [](Asyn
    request->send_P(200, "text/plain", readDH
  });
  server.on("/humidity", HTTP_GET, [](AsyncWeb
    request->send_P(200, "text/plain", readDHT
  });
  // Start server
  server.begin();
}
```

Output
Serial Monitor ×
Message (Enter to send message to 'ESP32 Dev Module' on 'COM9')

```
54.00
24.50
54.00
24.50
54.00
24.50
54.00
24.50
54.00
24.50
54.00
24.50
54.00
54.00
54.00
```



VIT Weather

Temperature 24.50 °C

Humidity 54.00 %

21BEC0256
Sreenivasayns

19/2/24 DHT11 webser with ESP32 (Arduino IDE)

Code
```
#include "WiFi.h"
#include "ESPAsyncWebServer.h'
#include <Adafruit_Sensor.h>
#include < DHT.h>
const char* password = "vit@12345";
#define DHTPIN 05
#define DHTTYPE DHT11
DHT dht(DHTPIN. DHTTYPE))
AsynWebServer server(80);
String readDHTTemperature() {
float t = dht.readTemperature();
                      not a number
if (isnan(t)) {
    Serial.println ('Failed to read from DHT sensor!");
    return "--";
  }
else {
       Serial.println (t);
       return String (t);
     }
  }
String readDHTHumidity(){
~~DHT dht (DHTPIN, DHTTYPE);~~
~~AsyncWebServer server(80);~~
~~String readDHTTemperature() {~~
~~float t = dht.readTemperature();~~
float h = dht.readHumidity();
if ( isnan(h)) {
    Serial.println ("Failed to read from DHT sensor!");
     return "--";
   }
```

```
else {
     Serial.println(h);
     return String(h);
     }
 }
const char index_html[] PROGRAM = R"rawliteral(
 <!DOCTYPE HTML><html>
<head>
   <meta name="viewport" content="width=device-width, intal-scale=
   <link rel = "stylesheet" href = "http://                          "
   integrity = "sha384-
   FnmOCqb
   crossorigin = "anonymous">
  <style>
  html {
       font-family: Arial;
       display: inline-block;
       margin: 0px auto;
       text-align: center;
   }
   h2 { font-size: 3.0rem; }
   p  { font-size: 3.0rem; }
   .units { font-size : 1.2rem }
   .dht-labels {
           font-size : 1.5rem;
           vertical-align : middle;
           padding-bottom: 15px;
       }
  </style>
  </head>
<body>
 <h2> VIT Weather </h2>
```

__/__/__

```
<p>
 <i class="fas fa-thermometer-half" style="colour.#059e8a;"></i>
 <span class="dht-labels"> Temperature </span>
 <span id="temperature"> %TEMPERATURE% </span>
 <sup class='units'> &deg;C </sup>
 </p>
<p>
 <i class="fas fa-tint" style="color:#00add6;"> </i>
 xhttp.open ("GET", "/temperature", true);
  xhttp.send ( );
 }, 10000);
 setInterval (function() {
 var xhttp = new XMLHttpRequest ();
 xhttp.onreadystatechange = function () {
  if ( this.readState == 4 && this.status == 200) {
   document.getElementByID ("humidity"). innerHTML =
     this.responseText;
   }
 };
 xhttp.open ("GET", "/humidity", true);
  xhttp.send ();
 }, 10000);
 </script>
 </html>)rawliteral";
 String processor (const String& var) {
  if (var == "TEMPERATURE") {
    return readDHTTemperature ();
   }
  else if (var == "HUMIDITY") {
     return readDHTHumidity ();
    }
  return String();
```

```
}
void setup(){
Serial. begin ( 115200);
dht.begin();
WiFi.begin (ssid, password);
while (WiFi . Status () != WL-CONNECTED) {
~~atta~~ delay (1000);
 Serial.println (" Connecting   to WiFi.. ");
}
   Serial. println (WiFi .localIP());
   server.on ("/", HTTP_GET, [] (AsyncWebServer Request *request){
    request -> Send_P (200, text/html);
  });
Server.on ("/humidity", HTTP_GET, [] (AsynWebServerRequest *request){
 request -> send_P (200, "text/plain", readDHTHumidity().c_str());
});
  Server.begin();
  }
  void loop() {
  }
```

verified

11/3/24

Materials Gathering: Gather the necessary components for the project: ESP32 board, DHT11/DHT22 sensor, USB cable, and a computer with Arduino IDE installed.

Hardware Connection: Refer to the sensor's datasheet for instructions on connecting the DHT11/DHT22 sensor to the ESP32 board.

Computer Connection: Establish a connection between the ESP32 board and the computer using the USB cable.

Software Setup: Open the Arduino IDE on the computer and create a new sketch.

Include the required libraries for ESP32, DHT sensor, Wi-Fi, and Webserver in the sketch.

Network Configuration: Define the Wi-Fi network credentials, including SSID and password, within the sketch.

Sketch Upload: Upload the sketch to the ESP32 board using the upload button in the Arduino IDE.

Power Supply: Disconnect the ESP32 board from the computer and power it using an external power supply.

Monitoring Access: Connect a device to the Wi-Fi network. Access the ESP32's IP address through a web browser to view the temperature and relative humidity values.

RESULTS:

The humidity and temperature are displayed and verified.