TASK 3

NAME: SREENIVASAN S

REG NO: 21BEC0256

SLOT: L3+L4

FACULTY: SUJATHA R

DHT11 Web Server with ESP32 (Arduino IDE)

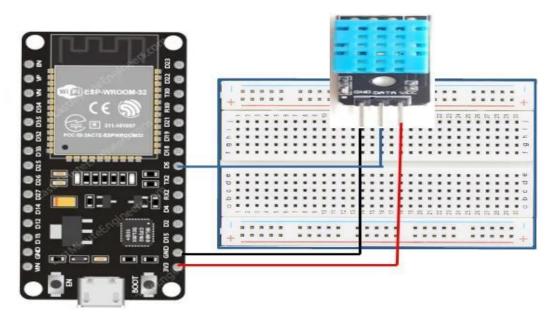
<u>Aim</u>

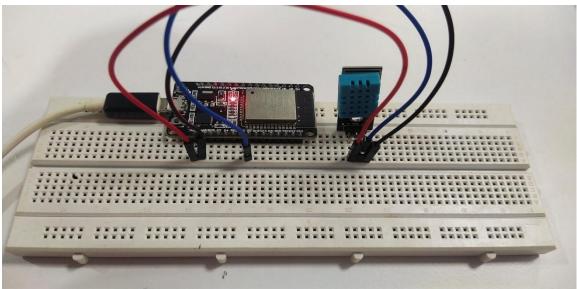
To develop a system that displays values from DHT11 sensor using a webserver with ESP32 as a control device in station mode, which connects to an existing Wi-Fi network.

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">
k rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css"
integrity="sha384fnmOCqbTlWIIj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr"
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>
 >
  <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>
 >
  <i class="fas fa-tint" style="color:#00add6;"></i>
```

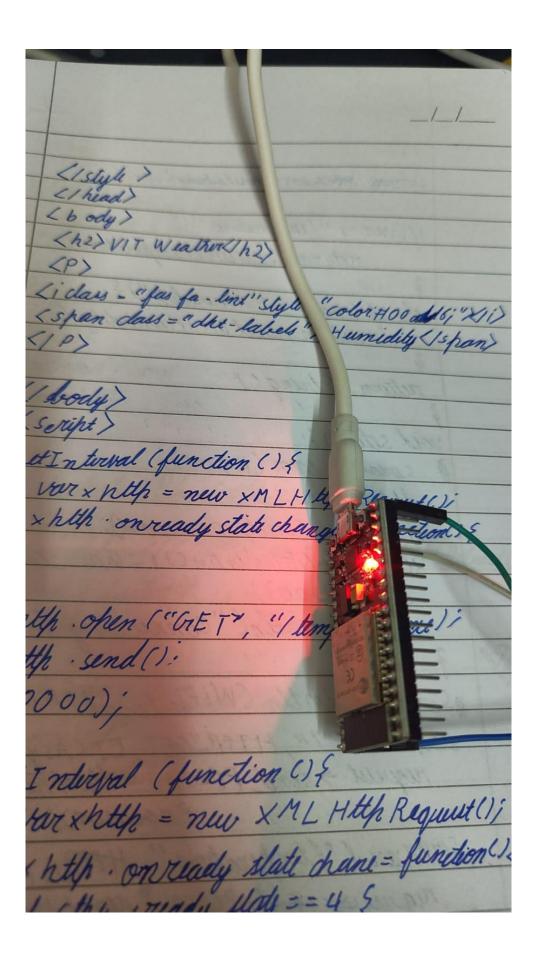
```
<span class="dht-labels">Humidity</span>
  <span id="humidity">%HUMIDITY%</span>
  <sup class="units">&percnt;</sup>
 </body>
<script> setInterval(function ( ) { var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() { if (this.readyState == 4 &&
this.status == 200) {
                       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 &&
                       document.getElementById("humidity").innerHTML
this.status == 200) {
= 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());
 });
```

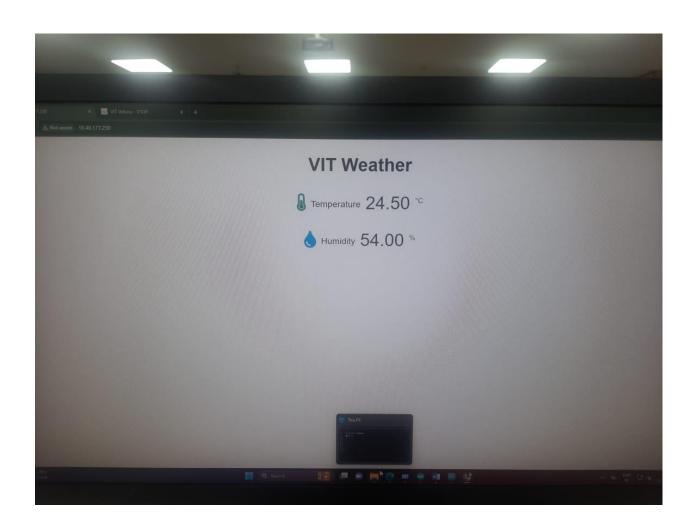
PROCESS AND STEPS TO DO IN THE PC:

- 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:



```
"text/html", index
                server.on("/temperature", HTTP_GET, [](Asyn
                  request->send_P(200, "text/plain", readDF
                server.on("/humidity", HTTP_GET, [](AsyncWeb
                  request->send_P(200, "text/plain", readDH
      132
                // Start server
      133
                server.begin();
      134
     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
(2)
      29°C
      Haze
```



	2 BECO256 Socenivasours/
14/2/24	DHT11 ovebeer with ESP32 (Arallino 1DE)
	Ye.
Coole	Hinclud "Wifi, h"
	Hinclude "ESPA sync WebServer. h"
	Hincholo L Ada Fruit Sensor. h>
	Hinchad & OHT. h>
	const chart password - "vit@ 10345";
	Holehne OHTPW 05
	Holefine OHITYDE OHIII
	DHT albe (OHTPLN. DHIT ME)
	Asyn Web Server Server (80);
	Stone or cod OHT Town per a trade ()
	float t = and read Temperature (s)
	if (isnan (t)) { not animor
	Serial Printly (Failed to read from OHT Sensor! ")
	setum & "";
	3
	else
	Serial Println (t);
	4
	1
	String & and DHT Humidily () {
1	OHE OHTOW, OH HYPE);
	Async Web Server Server (80);
	Slim red DHT Impertuse ()
	flood t = oth read Temperature();
	float h = dht. rod Hamidily ();
	if (Gnan(h)) (
	Serial printly ("Failed to sead from 1917 sensor.")
)	return " ";
•)
	Colle college (See 1981)

	else {
	. Serial printly (a)
	return String (h);
	1
	J
	const choos index-hlmf[] PROGRAM = R'seculitetel (
	C: DOCTIFE HTML> < hlome>
	(had)
	Lmota name = "viewport" content = "width = device - width, intal-scale
	L line rel > "Styleched" hard = "http://
	integrity: "shazere-
	Famolog b
	Cross congin - "anonymous">
	(style)
	name {
	Font-family: Arral:
	display: inline-block;
	may Opx outo;
	textalign center;
	j O
	h2 font-Size: 3.00em;)
	0 } tent-5i), 3.08em;
	cenity [Pont-size: 1.2 som]
	. cimity { font-size: 1.2 som}
	P + Che a L Formai
	vertical-align: middle:
	vertical-align: middle: Padding. bottom: 15px;
	3 0
10.7	
	Lhody>
	(h2) VIT Weather Kh2>

(4)
Li class= "fasto fa-thermometer-harp" Style="colour, 4059 esa"; ">41)
Span claces - "alut-vabols"> Temperature
< Spen id "temperature"> % TEMPERATURE % < 15pan>
(sup class - " unity" > & oleg; c < /sup>
218>
LP>
Li clas: "Pas fa tint' style=" color: # 00add6;")
xhttp. open (" GET" "/ temperature", true)
alto xhite send ();
], 10000);
set Interval Churchon () {
var with : new XMI Http Regast ()
xhttp. onxody state change - function () {
if (this Grand State == Le DU this stales == 200) {
document. get Element By 10 ("humidily"). inner HTML =
this response Text;
<i>3</i> ;
xhttp.open ("GET", "I (humidily", true);
xhttp. send csi
3, (0000);
< 1script>
Shire Discours ((out String to year) {
String processes (const String & vas) { iP (var == "[EMPERATORE") {
return read DHITemperature ();
else if (var == " (HUMIDITY") {
setum sad OHT Humidy ();
3
return String ()

	<u> </u>	197
	<u> </u>	7 31
1111 -	void selepes {	- 1 × 1
	Senal bigin (115200);	e e e gri à
	alm bagines;	i v
	Witi begin (SSIN parkoord);	1.0
	cohile (W.Fi Starty () 6 2 WL-	CONNECLED {
	attle deleny (1000);	
	Serial printer (Connating to Wifi	, ,
	3	
	Sexal Printin (Wir: 106d 1P	());
	Server.on ("/", HITP DET ET	France Web Server Regard Frances
	organst -> Sand P (200, test/hfm	0):
	1)	
	Call 11 HTP CCT 5	CARLLE Sound Down of trees
	Server on C"/humidity", HTTP_CAET, I x19ust -> Send -P (200, "Lext/Op)	of They was sent a sent a sent
	719 mst -> Send -7 (200, cext 100)	up, read DIT Homoure C. Ste
)/	
	Server begin ()	4.2
	3	
	void loop() }	3
Lind	3 was at Suranie I'm	65
2000		
W. 32		
211.		

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.