ENVIRONMENTAL MONITORING USING IOT

PHASE 5 : DOCUMENTATION

PROJECT'S OBJECTIVES :

- Monitoring of the environment may be undertaken for a number of reasons. In general monitoring is done in order to gain information about the present levels of harmful or potentially harmful pollutants in discharges to the environment, within the environment itself or in living creatures that may be affected by these pollutants.
  This definition can be expanded as follows:-

- Monitoring may be carried out to assess pollution effects on man and his environment in order to identify any possible cause and effect relationship between pollutant concentration and health effects, climatic changes etc.

- To evaluate pollution interactions and patterns

- To assess the need for legislative controls and emissions of pollutants and to ensure compliance with emission standards.

- To evaluate the tempersture and humidity of the surroundings inorder to alert the people incase of any issues.

IOT DEVICE DEPLOYMENT:

1.Select the ESP32:

In the "Select a board" section, type "ESP32" in the search bar and choose an ESP32 board model (e.g., ESP32 Dev Module).

2. Add the DHT22 Sensor:

In the components panel on the left, search for "DHT22"

Drag and drop the DHT22 component onto the breadboard area.

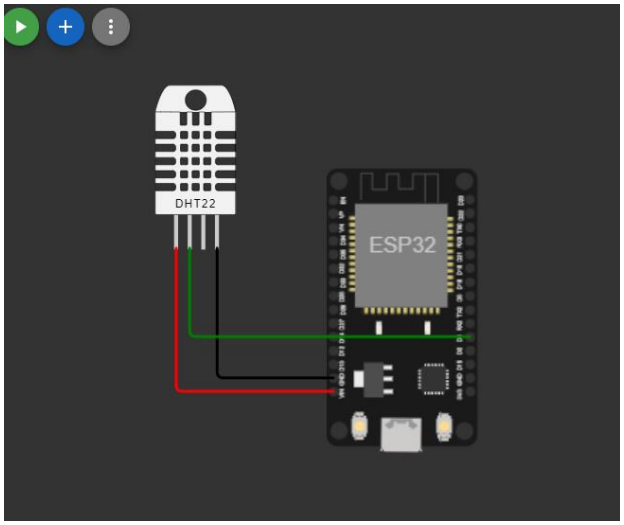
3. Connect the Components:

Connect the DHT22 sensor to the ESP32 as per your wiring instructions. You can click on the components to connect wires.

Connect the sensor's VCC and GND pins to the ESP32's 3.3V and GND, respectively.

Connect the sensor's data pin to a GPIO pin on the ESP32 (e.g., GPIO

4) Ensure the connections match your real-world wiring.



PLATFORM DEVELOPMENT:

Platform development refers to the development of the fundamental software that makes hardware work and that provides a platform for application development.

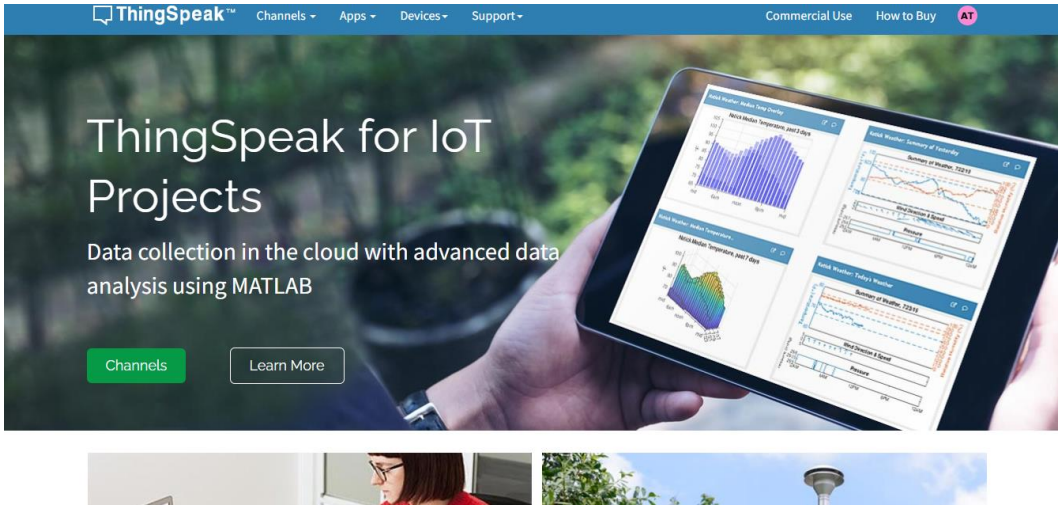
Virtual platforms can be used to efficiently deliver the platform to application developers, containing both hardware and software, booted, configured, and ready to go.

WOKWI:

Wokwi is an online simulator for Arduino, Raspberry Pi Pico, and ESP32 boards, or even your own custom microcontroller board designed to learn programming without the actual hardware.

Real-time temperature and humidity can be monitored using many methods here we use ThingSpeak and a Http Server.

THINGSPEAK:



ThingSpeak allows you to aggregate, visualize, and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices or equipment.

In real-time connect the components accordingly as given in the above wokwi components circuit .

The code will be same for both virtual and real-time sensors.

CODE IMPLEMENTATION :

```
#include <WiFi.h>
#include <Wire.h>
#include "DHT.h"
#define DHTPIN 4
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
// Replace with your network credentials
const char* ssid     = "Wokwi-GUEST"; //Your SSID
const char* password = ""; //Your Password
// Set web server port number to 80
WiFiServer server(80);
// Variable to store the HTTP request
String header;
void setup() {
  Serial.begin(9600);
  bool status;
// default settings
  // (you can also pass in a Wire library object like &Wire2)
  //status = bme.begin();
```

```
// Connect to Wi-Fi network with SSID and password
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  // Print local IP address and start web server
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  server.begin();
}
void loop(){

  WiFiClient client = server.available();   // Listen for incoming clients
if (client) {                     // If a new client connects,
    // Reading temperature or humidity takes about 250 milliseconds!
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
    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);
while(isnan(h) || isnan(t) || isnan(f)){
      delay(5000);
```

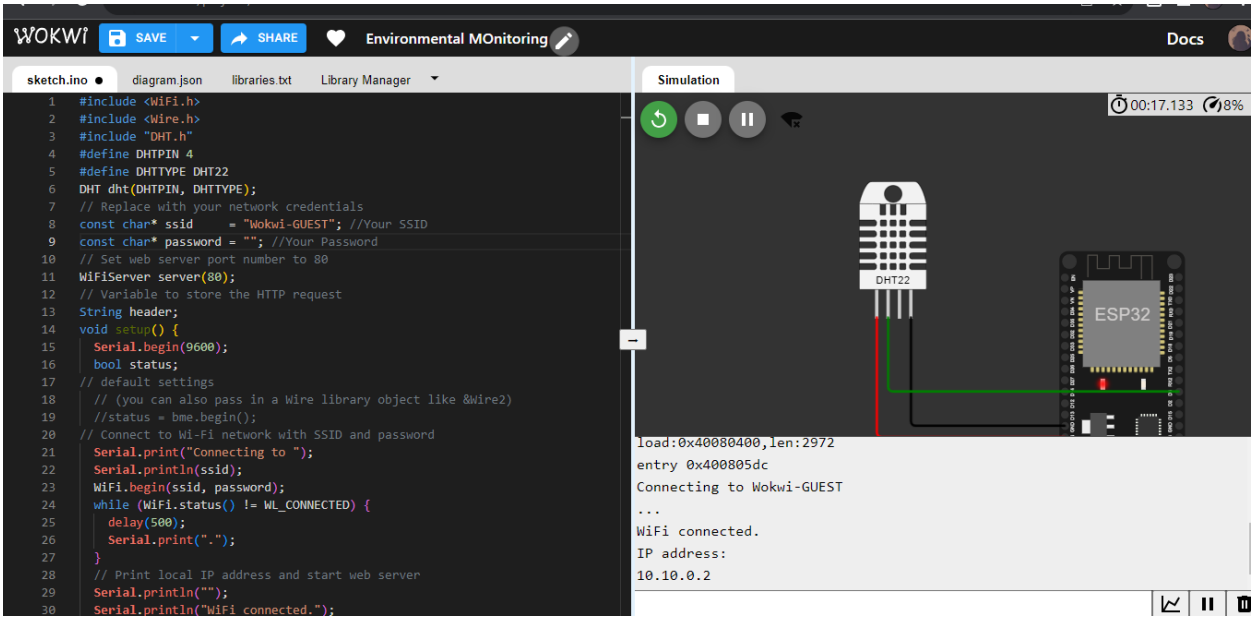
```
      h = dht.readHumidity();
      t = dht.readTemperature();
      f = dht.readTemperature(true);
    }
// Compute heat index in Fahrenheit (the default)
    float hif = dht.computeHeatIndex(f, h);
    // Compute heat index in Celsius (isFahreheit = false)
    float hic = dht.computeHeatIndex(t, h, false);
Serial.print(F("Humidity: "));
    Serial.print(h);
    Serial.print(F("%  Temperature: "));
    Serial.print(t);
    Serial.print(F("°C "));
    Serial.print(f);
    Serial.print(F("°F  Heat index: "));
    Serial.print(hic);
    Serial.print(F("°C "));
    Serial.print(hif);
    Serial.println(F("°F"));

    Serial.println("New Client.");          // print a message out in the serial port
    String currentLine = "";                // make a String to hold incoming data from the client
    while (client.connected()) {            // loop while the client's connected
      if (client.available()) {             // if there's bytes to read from the client,
        char c = client.read();             // read a byte, then
        Serial.write(c);                    // print it out the serial monitor
        header += c;
        if (c == '\n') {                    // if the byte is a newline character
```

```
// if the current line is blank, you got two newline characters in a row.
// that's the end of the client HTTP request, so send a response:
if (currentLine.length() == 0) {
  // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)
  // and a content-type so the client knows what's coming, then a blank line:
  client.println("HTTP/1.1 200 OK");
  client.println("Content-type:text/html");
  client.println("Connection: close");
  client.println();

  // Display the HTML web page
  client.println("<!DOCTYPE html><html>");
  client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");
  client.println("<link rel=\"icon\" href=\"data:,\">");
  // CSS to style the table
  client.println("<style>body { text-align: center; font-family: \"Trebuchet MS\", Arial;}");
  client.println("table { border-collapse: collapse; width:35%; margin-left:auto; margin-right:auto; }");
  client.println("th { padding: 12px; background-color: #0043af; color: white; }");
  client.println("tr { border: 1px solid #ddd; padding: 12px; }");
  client.println("tr:hover { background-color: #bcbcbc; }");
  client.println("td { border: none; padding: 12px; }");
  client.println(".sensor { color:white; font-weight: bold; background-color: #bcbcbc; padding: 1px; }");

  // Web Page Heading
  client.println("</style></head><body><h1>ESP32 with DHT11</h1>");
  client.println("<table><tr><th>MEASUREMENT</th><th>VALUE</th></tr>");
  client.println("<tr><td>Temp. Celsius</td><td><span class=\"sensor\">");
  client.println(t);
```

```
        client.println(" *C</span></td></tr>");
        client.println("<tr><td>Temp. Fahrenheit</td><td><span class=\"sensor\">");
        client.println(f);
        client.println(" *F</span></td></tr>");
        client.println("<tr><td>Humidity</td><td><span class=\"sensor\">");
        client.println(h);
        client.println(" %</span></td></tr>");
         client.println("<tr><td>Heat Index (Celcius)</td><td><span class=\"sensor\">");
        client.println(hic);
        client.println(" *C</span></td></tr>");
         client.println("<tr><td>Heat Index (Fahrenheit)</td><td><span class=\"sensor\">");
        client.println(hif);
        client.println(" *F</span></td></tr>");
        clien
        client.println("</body></html>");

        // The HTTP response ends with another blank line
        client.println();
        // Break out of the while loop
        break;
      } else { // if you got a newline, then clear currentLine
        currentLine = "";
      }
    } else if (c != '\r') {  // if you got anything else but a carriage return character,
      currentLine += c;      // add it to the end of the currentLine
    }
  }
}
```

```
  // Clear the header variable
  header = "";
  // Close the connection
  client.stop();
  Serial.println("Client disconnected.");
  Serial.println("");
 }
}
```

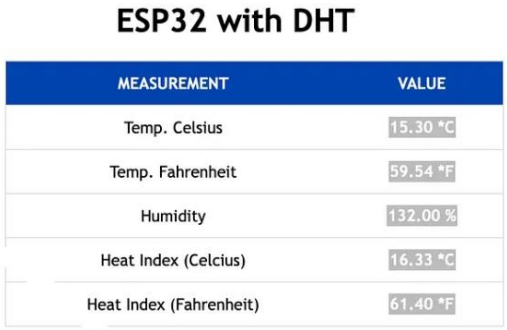
OUTPUT :



- **C**opy the Ip address given on the wokwi output and paste it on any browser
- It will display the temperature and humidity values from the wokwi.
- By this technology anyone can able to monitor the environment from any part of the world.
- The following is the example of the readings which are displayed on the http webserver . This can be accessed on the pc or even on your mobiles phones .

This is the sample output which is displayed in the wwb server .

We can access it from any where through the ip address given the wokwi output .



RESULT ANALYSIS :

Temperature and Humidity :

The DHT sensor provides environmental data , such as Temperature =40 C, Humidity = 76%.

HOW ENVIRONMENTAL MONITORING HELPS PARK VISITORS:

Risks faced by them:

The risks faced by very young children, those aged 4 and under, are the most acute of all.These children are often far more active than adults and they heat up more quickly. They also don’t sweat as much, which makes it much harder for their bodies to cool down. Dehydration is another potentially fatal risk during extreme heat, especially for babies and young children.

Young children’s developing organs are vulnerable to heat stroke once their body temperature goes beyond what the child’s system can regulate – about 107 ºF (41.6 ºC).

Children’s vulnerability can increase by fluid loss through the use of certain medications, or conditions such as diarrhoea. Other risks include caregivers dressing children too warmly for the ambient temperature, and poor ventilation, since many buildings are ill equipped to deal with high temperatures, especially in poorer countries or places traditionally unused to extreme heat.

Benefits of this project:

The parents of the children should monitor the temperature and humidity before taking the children to play in the parks.

This iot web server should accessable by everyone so that they can monitor the environment .

They should not allow the childrens to play if the temperature is more than 45degree celcius and the humidity is more than 60% .

Know how to protect yourself, your family and your community from heat stress:

1. Know how hot and humid it is going to get today, this week and this month to help plan
outside activities.
2. Avoid going out at the hottest times of the day. Remember that the temperature on
weather stations is always in the shade.65
3. Know how to get help. Know important contact information for the nearest health care
provider or ambulance/transport services and use it to receive help as soon as necessary.
4. Protect the vulnerable and yourself. Stay in the shade, drink water before feeling thirsty,
and rest! If your community has a cooling centre, use it.
5. Have everyone wear light, loose, breathable fabrics and use sunscreen. Cotton is ideal

during hot weather to help reduce heat rashes and absorb sweat. Use umbrellas and hats
outside for protection.
6. Choose water for rehydration over coffee, tea or soft drinks. Caffeine, alcohol and sugar
can affect sleep and well-being.
7. Keep an emergency kit at home containing towels, oral rehydration salt (ORS) packets and
a thermometer. Give ORS mixed into water to anyone with extreme sweating, especially
children who might have played outside.
8. Close the curtains during the hottest parts of the day, open windows when it is hotter
inside than outside and use fans and coolers when available. In dry heat, fans can be used
along with wet towels or misting the body.66
9. Ensure that infants, pregnant women and other vulnerable populations sleep in cooler
areas, such as lower floors of the house or building. Use cotton sheets for bedding.
10. Check on your neighbours and share information, particularly if they have young children,
pregnant women, or elderly individuals. Support efforts that encourage protection of
vulnerable workers from heat exposure.

CONCLUSION:

The Internet of Things (IoT) has begun to emerge as it has helped human in controlling and monitoring essential conditions using devices which are able to capture, evaluate, and transmit information from the environment to the cloud, where the data will be stored. Environment monitoring system is one of the most important IoT systems which mainly includes data collections through sensors and data reviewing for short-term measure as well as remote management and observations We can use this concept to publish our sensor

data and see it from our local network. With a real server and proper internet connection, we can use access it from anywhere. We could also use multiple sensors to make the data more accurate

With this solution, you can accurately monitor the temperature and humidity data in a particular space. The data is informatively displayed on the _IoT_dashboard_for analysis and generating reports. Environmental monitoring solution is powered with wireless sensors that allow you to monitor the temperature and relative humidity levels across your surroundings like offices, rooms, museums, and laboratories. The solution also allows you to set temperature and humidity thresholds for a customized experience. The sensors are well-equipped with automation concepts and personalize your surroundings according to your comfort level.