

# National Textile University, Faisalabad



## Department of Computer Science

|                         |                 |
|-------------------------|-----------------|
| <b>Name:</b>            | Mariha Iftikhar |
| <b>Class:</b>           | BSCS 5th _A     |
| <b>Registration No:</b> | 23-NTU-CS-1045  |
| <b>Course Name:</b>     | Embeded IOT     |
| <b>Submitted To:</b>    | Sir Nasir       |
| <b>Submission Date:</b> | 17/12/2025      |

## Assignment NO 2

### Question no 1:(Part A)

1.

**What is the purpose of web server(80) and what does port 80 represent?**

Web server (80) is used to create a web server object on ESP32 that listen for incoming HTTP requests.

Port 80 is the default port for HTTP communication .When a user enter the IP address of ESP32 in web browser without specifying port number the browser automatically use port 80. This allows ESP32 to act as a web server and communicate with web browser using standard HTTP protocol.

2.

**Explain the role of server.on (“/”,handleRoot)?**

Server .on(“/”,handleroot);defines the behaviour of web server when the root URL is accessed.

When a client open the ESP32 IP address in a web browser ,this statement calls the handleroot()function.The function is responsible for generating and sending the web page to the client.

3.

**Why is server.handleclient():placed inside the loop() function ?what will happen if it is removed?**

Server.handleclient is placed inside the loop () function because it continuously check for incoming client request and process them.

If this statement is removed the server will response to any browser requests,and the web page will not load even through the ESP32 is connected to wifi.

4.

**In handleroot() , explain the statement:**

**Server.send(200,”text/html”,html)**

This statement send a response from ESP32 web server to the client.

200 indicate a successful HTTP request.

“text/html” specifies that response content is an HTML web page.

HTML contain the actual code that will be displayed in the browser.

5.

### **Difference between displaying last measured sensor values and taking a fresh DHT reading inside roothandle()?**

Displaying the last measured sensor values means DHT sensor reading are taken earlier and stored in variables, and handleroot() only displays those stored values. this approach is faster and more reliable.

Taking a fresh DHT reading inside handleroot() means the sensor is read every time the web page is requested . This can slow down the webpage and may cause inaccurate reading because DHT sensors require proper timing between readings.

## **Question no 1(Part B):**

- Working of ESP32 Webserver-Based Temperature and humidity Monitoring System:**

An ESP32 webserver-based temperature and humidity monitoring system is designed to measure environment conditions and make them available both locally and remotely. ESP32 play a central role by connecting to wi-fi , reading sensor data ,displaying values on an OLED screen, and hosting a web page that shows real-time readings.

- ESP32 Wi-fi connection and IP Address Assignment:**

When powered on the ESP32 activites its wifi module and uses the predefined SSID and password to connect to the network .It continuously checks the connection status until successfully connected. After connection the router assigns an IP address to ESP32 which is shown on the serial moniter This IP address can be entered into a web browser to access the ESP32 web server from any device on the same network.

- Web server initialization and Request Handling:**

After connecting to Wifi ESP32 starts its internal web server on port 80. When a user enters ESP32 IP address in a browser it receive an HTTP request and respond with HTML page showing temperature and humidity data.

- Button-Based Sensor Reading and OLED Update Mechanism:**

A push button is connected to GPIO pin is used to control sensor readings. When pressed it changes the pin's logic level , prompting the ESP32 to read temperature and humidity data. The OLED display is cleared and update with the new values, ensuring efficient, user-controlled data updates.

- Dynamic HTML Webpage Generation**

The ESP32 hosts a dynamic web page that updates with real-time sensor readings. The HTML is generated within the program , inserting the latest temperature and humidity values.

Each page refresh shows updated data allowing real time monitoring through a web browser without extra application.

- **Purpose of Meta Refresh in the Webpage**

The meta refresh tag is used in the HTML page to automatically reload the webpage after a fixed time interval. This feature ensures that the displayed temperature and humidity values are updated regularly without requiring the user to manually refresh the page.

By using meta refresh, the system provides near real-time monitoring while keeping the webpage design simple and lightweight.

- **Common Issues in ESP32 Webserver Projects and Their Solutions**

. Incorrect Wi-Fi credentials can prevent the ESP32 from connecting to the network, which can be resolved by verifying the SSID and password. Sometimes the webpage does not load due to an incorrect IP address.

Sensor readings may show invalid values if the sensor is not properly connected or if readings are taken too quickly; this can be fixed by checking wiring and adding appropriate delays.

## **Question no 2(Part A):**

**1.**

**What is the role of Blynk Template ID in an ESP32 IoT project? Why must it match the cloud template?**

The Blynk Template ID uniquely identifies the device template created in the Blynk cloud .It links the ESP32 firmware to the correct cloud dashboard configuration.The template ID must match the cloud template so that the device can correctly communicate with the widgets ,virtual pins, and settings defined in the blynk cloud.

**2.**

**Difference between Blynk Template ID and Blynk Auth Token?**

The Blynk template ID identifies the type of device and its dashboard layout in the Blynk cloud.

The Blynk Auth Token is a uniquely security key used to authenticate a specific device to connect to blynk cloud.

**3.**

**Why does using DHT22 code with a DHT11 sensor produce incorrect readings? Mentions one key difference between the two sensor?**

Using DHT22 code with DHT11 sensor produces incorrect readings because both sensor have different data formats , resolution and timing requirements.

Difference is that DHT22 provides higher resolution and wider temperature and humidity range , while DHT11 has lower accuracy and limited range.

4.

**What are virtual pins in Blynk?Why are they preferred over physical GPIO pins for cloud communication?**

Virtual pins are software -defined pins in Blynk that are used to send and receive data between the ESP32 and Blynk Cloud.They are preferred over physical GPIO pins because they allow cloud-based data exchange without depending on the hardware pin configuration , making the system more flexible and scalable.

5.

**What is the purpose of using BlynkTimer instead of delay() in ESP32 IoT applications?**

BlynkTimer is used to execute tasks at regular intervals without blocking program execution .Unlike delay() it allows ESP32 to handle WiFi communication and cloud updates continuously. This ensures stable connectivity and responsive performance in IoT applications.

## **Question no 2(Part B):**

**Workflow of Interfacing ESP32 with Blynk Cloud for Temperature and Humidity Monitoring**

1.

**Creation of Blynk Template and Datastreams:**

Blynk template is created in Blynk cloud. This template defines the type of devices and the structure of data to be sent. Within template datastreams are created to represent specific values like temperature and humidity.These datastreams store , visualize and allow interaction with the data on Blynk app.

2.

**Role of template ID ,template Name, Auth Token:**

**Template ID:**a unique identifier for the device template in Blynk cloud .ESP32 uses it to connect to the correct template.

**Template Name:**The human-readable name of the template for easy identification.

**Auth Token:**A secret key assigned to each device . the esp32 uses it to authenticate and securely send data to the Blynk cloud

**3.**

### **Sensor configuration issues(DHT11 &DHT12)**

**DHT11:**Lower accuracy , slower update rate and limited temperature range.

**DHT22:**Higher accuracy , wider range and faster response.

**4.**

### **Sending data usingBlynk.virtualWrite():**

- Vx is the virtual pin assigned to a datastream in the template.
- Value is the sensor reading.

This updates the blynk app in real time, allowing user to see temperature and humidity instantly

**5.**

### **Common problem and solutions in ESP32-blynk projects:**

#### **Problems:**

- Incorrect sensor reading:
- Data not showing in app
- Wifi -issues

#### **Solutions:**

- Respect sensor timing and check connections.
- Ensure code pins match Blynk datastreams pins
- Use correct SSID/password and good network coverage.

#### **Blynk Cloud (web) and Blynk mobile app dashboards:**

Welcome to BlynkConsole - n... Devices - Blynk.Console

sgp1.blynk.cloud/dashboard/949961/global/devices/1/organization/949961/devices/3695896/dashboard

Blynk.Console My organization - 2216OS

dht 11 • Inactive

bxHL & Mariha My organization - 2216OS

Get Started Dashboards Custom Data Developer Zone

Devices Automations Users Organizations Locations

Snapshots Fleet Management

In-App Messaging

Collapse

temperature humidity

46.4 °C 50%

Live 1h 6h 1d 1w 1mo 3mo 6mo 1y

Messages used: 55 of 200.0k

Add Tag

Region: SGP1 Privacy Policy Terms of Service

17°C Mostly sunny

The screenshot shows the Blynk Console interface. On the left is a sidebar with navigation links like Get Started, Dashboards, and Devices. The main area displays a device named "dht 11" which is inactive. It shows two circular gauge charts: one for temperature at 46.4 °C and one for humidity at 50%. Below the charts is a timeline selector from Live to 1 year. At the bottom, there's a weather widget showing 17°C and Mostly sunny. The status bar at the bottom right shows the region as SGP1, language as ENG US, and the date/time as 11:24 am 19/12/2025.



