National Textile University, Faisalabad

Department of Computer Science

ASSIGNMENT # 2

Name:	SAJAL NAEEM
Class:	BSCS 5TH B
Registration No:	23-NTU-CS-1089
Course Name:	IOT AND EMBEDDED SYSTEMS
Submitted To:	SIR NASIR MEHMOOD
Submission Date:	17TH DECEMBER 2025

QUESTION # 1:

PART A: SHORT QUESTIONS:

1. **What is the purpose of Webserver server (80); and what does port 80 represent?**

This line in the code helps to create a server on ESP 32. And it connects the web browser which responds to all the incoming requests on http. Port 80 represents the default port which is for http websites.

2. **Explain the role of server. On ("/", handle Root); in this program.**

This line in the code helps to guide the browser about what to do when URL / is opened. When IP of ESP 32 is opened on the browser then automatically the handle Root function is called. This function creates and sends the HTML Webpage

3. **Why is server.handleClient(); placed inside the loop () function? What will happen if it is removed?**

Server. handle client () checks for incoming web requests repeatedly. It is placed inside the loop so the ESP 32 can respond to requests at any time.

4. **In handle Root (), explain the statement: server. Send (200, "text/html", html);**

This line in the code sends data from ESP 32 to the web browser. The number 200 says that the request is successful. Text/html tells the browser that its webpage. Html contains the content to display.

5. **What is the difference between displaying last measured sensor values and taking a fresh DHT reading inside handle Root ()?**

LAST MEASURED SENSOR VALUES	**FRESH DHT READINGS**
THESE VALUES USES STORED SENSOR DATA AND AVOIDS READING THE DHT SENSOR EVERY TIME THE PAGE RELOADS.	THESE READINGS ARE TAKEN USING THE DHT SENSOR EVERY TIME THE PAGE IS REFRESHED.

PART B: LONG QUESTION:

Describe the complete working of the ESP32 webserver-based temperature and humidity monitoring system.

Answer:

ESP 32 WIFI CONNECTION:

When esp. 32 is powered on then it initializes a serial monitor and OLED display. Esp 32 connects to the Wi-Fi by using the one SSID and password which the user provides in the code. The line WIFI. Begin (SSID, password); helps to connect the esp. with Wi-Fi.

IP ADDRESS:

Once esp. 32 is connected to the Wi-Fi then the router allots an Ip address to the esp. 32.the user can use this Ip address to access the esp. 32 webpages.

WEB SERVER CREATION AND REQUEST HANDLING:

Web server is created using port number 80 and the command: Webserver server (80).

This command starts an HTTP server on port number 80.the line given below helps to respond to the browser requests whenever someone uses the root URL with /

server. On ("/", handle Root).

BUTTON BASED SENSOR READING AND OLED UPDATE:

The button is normally attached to esp. 32 using pullup mode. The button is released in high state and pressed in low state. Inside the loop the esp. continuously checks for the button inputs and reads the readings from DHT sensor and stores them Inside global variables and then displays the readings on the OLED

DYNAMIC HTML WEBAGE GENERATION:

The handle root function creates a dynamic html webpage by using a string variable. The webpage may include the system named DHT readings or the option to press the button. The page is sent to the browser using the following command: server. Send (200, "text/html", html).

PURPOSE OF META REFRESH:

The purpose of meta refresh is to display the updated values of the temperature and humidity on the web browser. This allows the webpage to display the newest values, creating a real time effect. The values are read after every 5 seconds

The following command helps to meta refresh the page: <meta http-equiv='refresh' content='5'>

Common issues in ESP32 webserver projects and their solutions

The common problems may include:

- The Wi-Fi is not connected or secured
- The webpage is not loading
- The OLED is not displaying readings
- The DHT is giving invalid readings.

QUESTION # 2:

PART A: SHORT QUESTIONS:

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

The BLYNK ID helps ESP 32 connect with specific projects on the BLYNK cloud app. It tells the BLYNK app about which dashboard and settings should be used. So, it needs to match with cloud template to connect device successfully to cloud.

2. **Differentiate between Blynk Template ID and Blynk Auth Token.**

TEMPLATE ID	**AUTH TOKEN**
Helps to Identifies a specific dashboard or project in the BLYNK app	The auth token helps to connect your device securely to the cloud

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

DHT22 and DHT 11 uses different data and accuracy so that is why it causes incorrect reading results if the code is not according to the correct DHT model. The key difference between both is that DHT 11 has low accuracy and smaller temperature and humidity reading range.

4. **What are Virtual Pins in Blynk? Why are they preferred over physical GPIO pins for cloud communication?**

Virtual pins are software pins that are not connected to ESP 32 GPIO's physically but allows us to send and receive data virtually providing very flexible communication.

5. **What is the purpose of using Blynk Timer instead of delay () in ESP32 IoT applications?**

Delay () function stops the entire ESP 32 working and does not allow any other tasks to run. Whereas the Blynk Timer is flexible to use as it does not stop the working of other tasks and does not provide blocking.

PART B: LONG QUESTION:

Explain the complete workflow of interfacing ESP32 with Blynk Cloud to display temperature and humidity values.

Answer:

CREATION OF TEMPLATE:

Template is the main part of a project in IOT. It defines how the device is connected to the BLYNK cloud and how data is shown on dashboard.

CREATION OF DATASTREAMS:

Inside the template the data streams are created. The data streams are used to receive data from esp. 32 and they are connected through a virtual pin and not physically by a GPIO.

ROLE OF TEMPLATE ID AND TEMPLATE NAME AND AUTH TOKEN:

TEMPLATE ID	Uniquely identifies a project on BLYNK cloud. It ensures that esp. 32 connects to the correct template
TEMPLATE NAME	A readable name given to project helps the project to be identifiable inside the BLYNK cloud
AUTH TOKEN	A unique security key assigned to each device which makes sure the security connection of the device with the BLYNK cloud.

SENSOR CONFIGURATION ISSUES DHT22 AND DHT11:

These two sensors may look same but follow different data and accuracy and ranges. The issues may arise due to difference between accuracy level and temp ranges.

SENDING DATA USING FUNCTION:

The data is sent to BLYNK cloud app using the BLYNK.virtual write () function. This function sends specific data to specific virtual pin. Such as temp data is sent on V0 pin.

COMMON PROBLEMS FACED WITH SOLUTIONS:

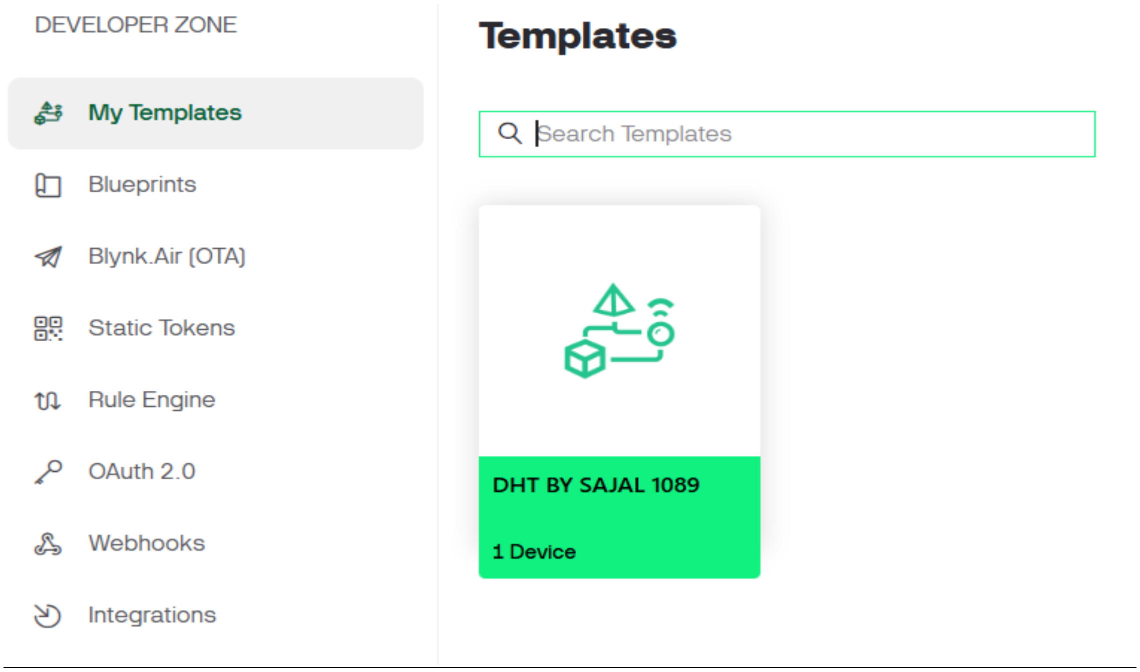
- Incorrect template id and auth token
- No data appearing on blynk app
- Incorrect sensor readings.

Solutions:

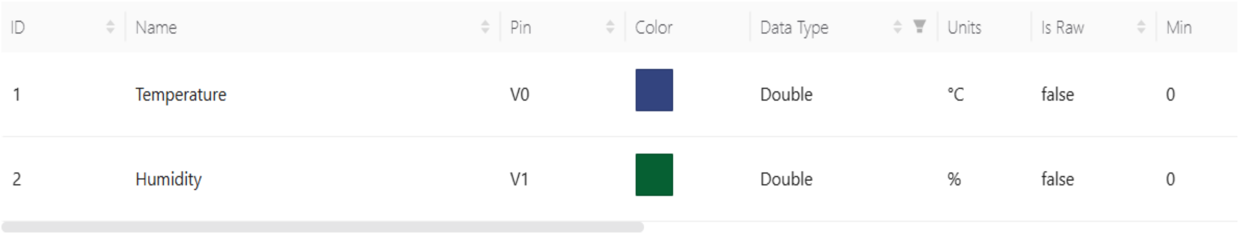
- Copy these two directly from BLYNK dashboard
- Ensure the same virtual pins are used in code and template.
- Define correct sensor type in code.

BLYNK SCREENSHOTS:

Templates:

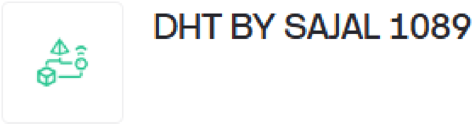


DATASTREAMS:

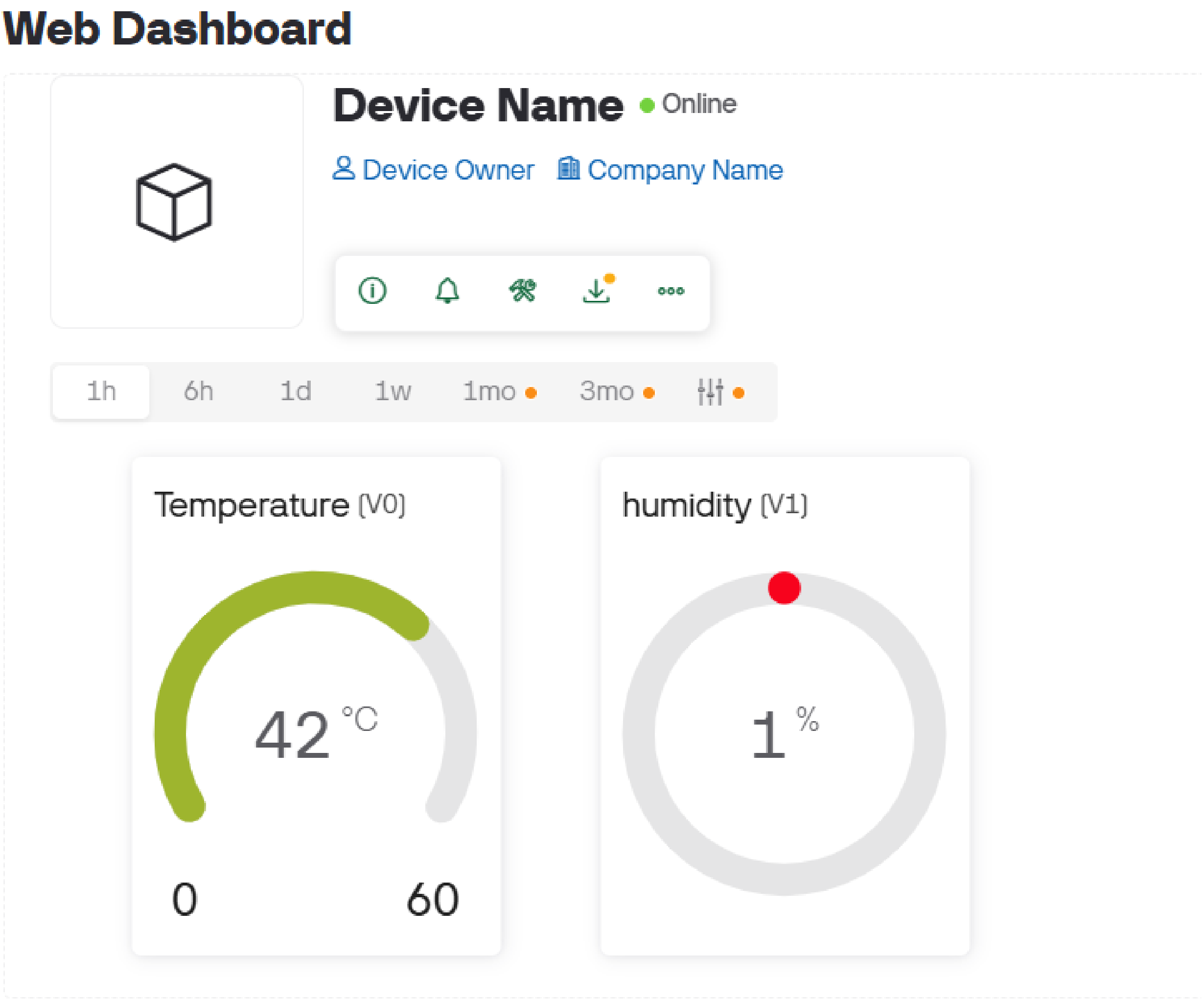


ID	Name	Pin	Color	Data Type	Units	Is Raw	Min
1	Temperature	V0	🟦	Double	°C	false	0
2	Humidity	V1	🟩	Double	%	false	0

DASHBOARD:



DHT BY SAJAL 1089



Web Dashboard

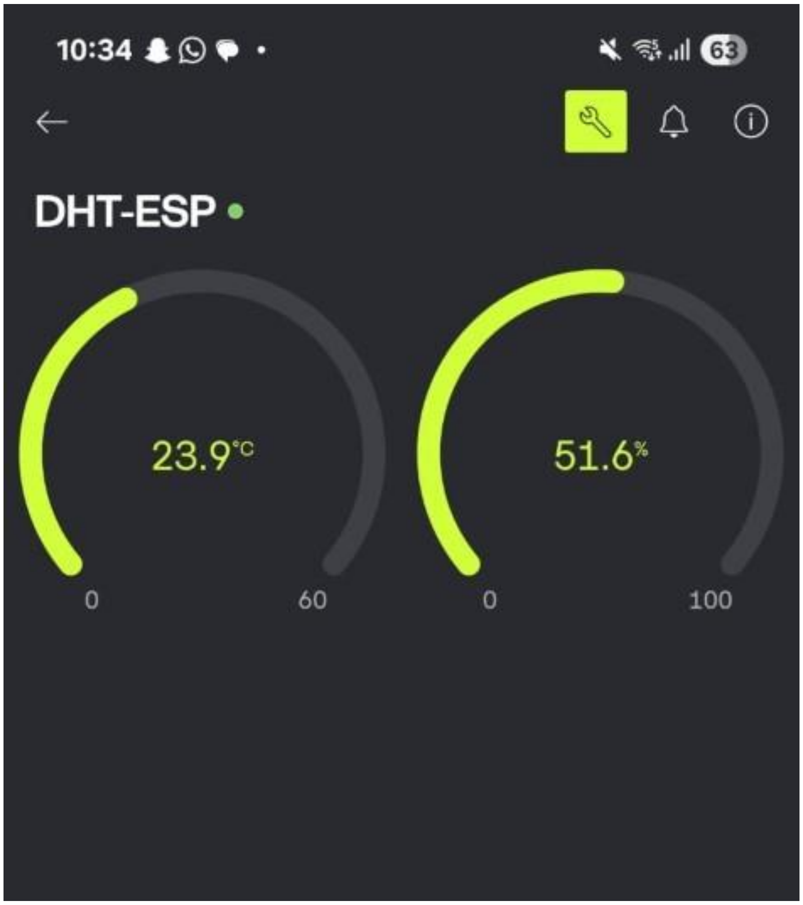
Device Name ● Online

Device Owner Company Name

Temperature (V0)

humidity (V1)

MOBILE DASHBOARD:



10:34
63
DHT-ESP
23.9°C
0
60
51.6%
0
100