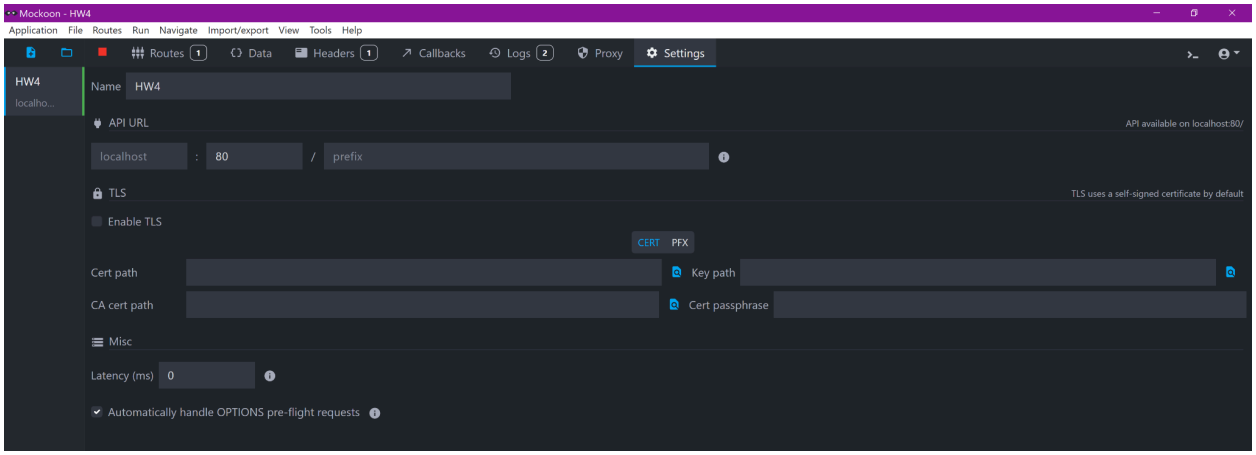
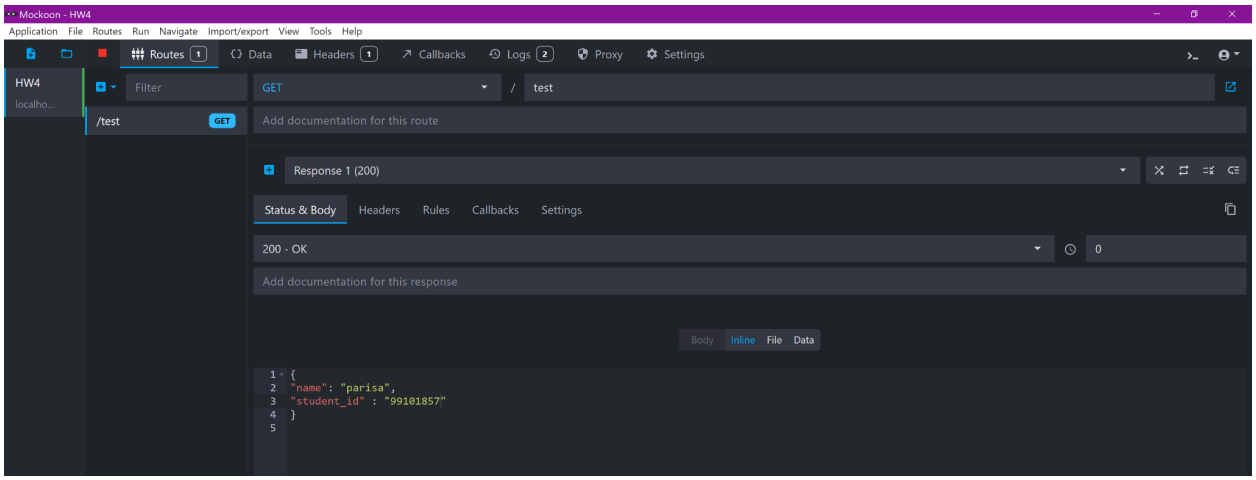
SIoT ➔HW 4 ➔HTTP
Parisa Toumari
99101857

1. Setting up a Mock Server:

To begin the process, the system was connected to a Wi-Fi modem, allowing it to access the internet. Following the connection, an IP address was assigned to the system. The assigned IP address is crucial for communication between devices on the network. Utilizing the information provided in class and leveraging tools like Postman and Mockoon, a mock server was set up on port 80. This mock server allows for the simulation of HTTP requests and responses, enabling the testing of various scenarios without interacting with a live server.

Mockoon configuration was set as below:





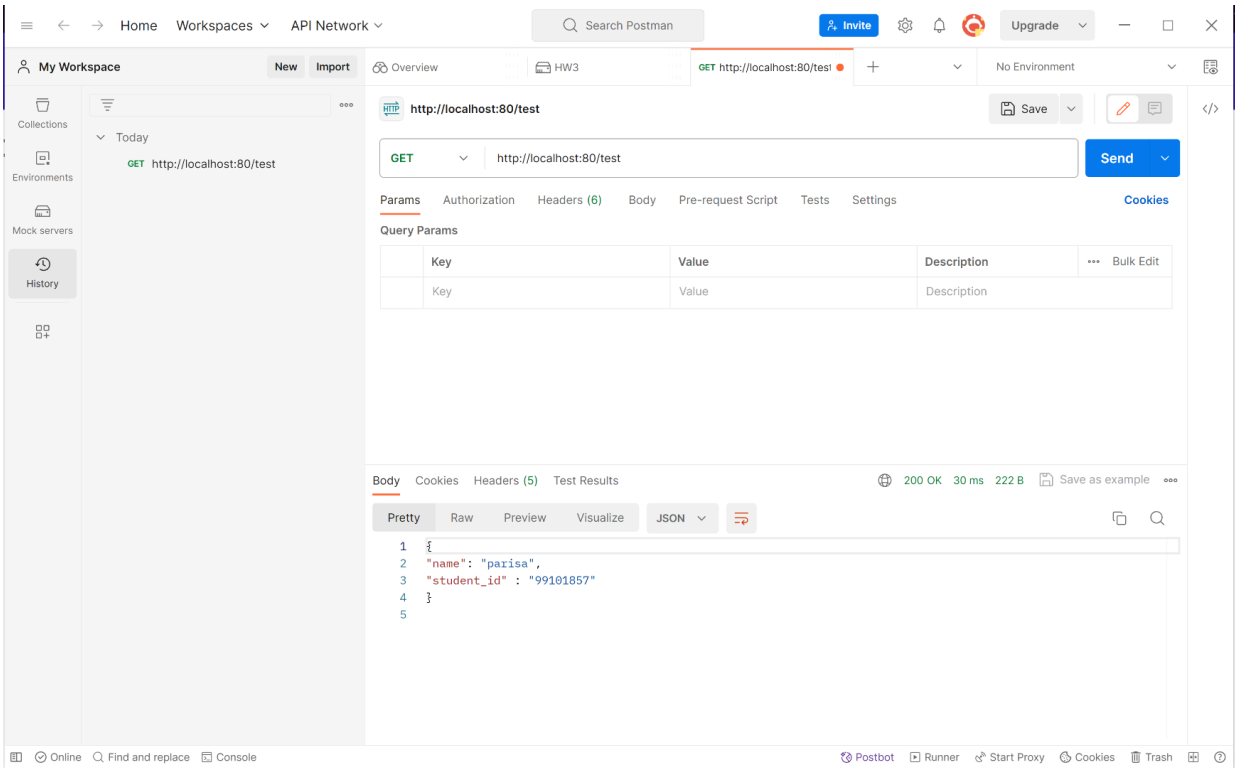
A service was created with the GET method and the specified address. The response to this service includes the following JSON parameters:

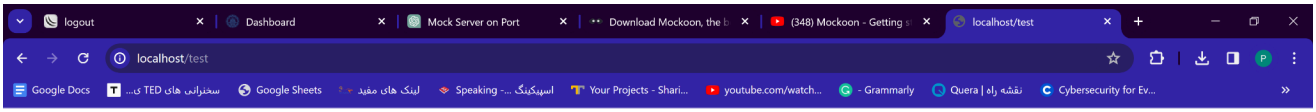
```
{
  "name": "your_name",
  "student_id": "your_student_id"
}
```

Additionally, the HTTP status code in the response was set to 200. This status code signifies a successful request. In this context, a status code of 200 indicates that the request was processed successfully, and the response contains the expected JSON parameters.

In the context of HTTP status codes, a code of 200 is known as the "OK" status code. It indicates that the request has been successfully processed, and the server is returning the expected data. In the implemented scenario, a status code of 200 in response to the specified service means that the server mock has successfully processed the GET request and provided the requested JSON parameters.

Sending a request using Postman:





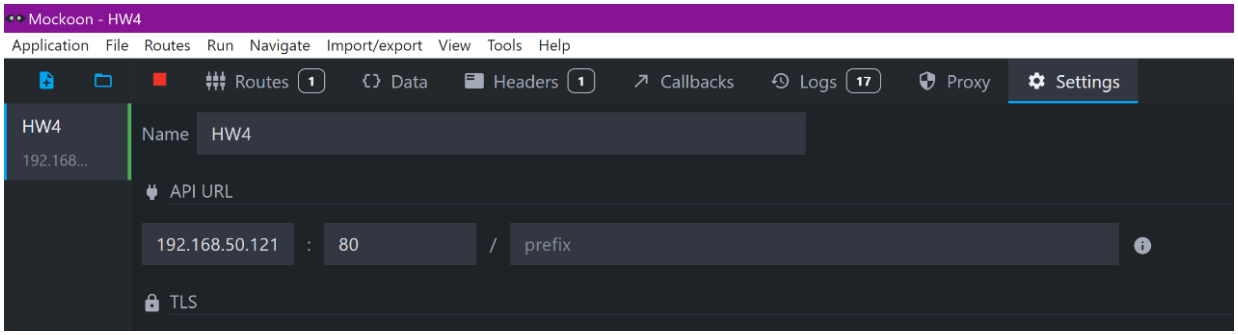

```
{
"name": "parisa",
"student_id" : "99101857"
}
```

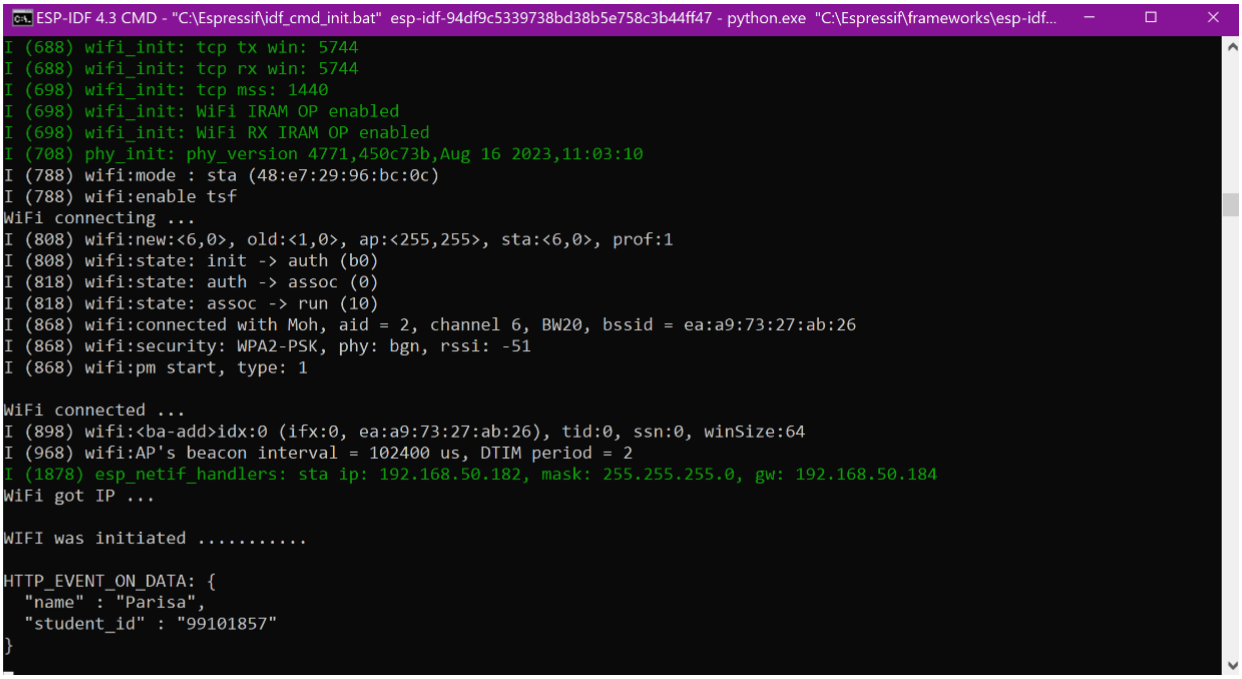
2. ESP32 as Client:

Using the ESP module, the system successfully connected to the local Wi-Fi network. The ESP module provides wireless connectivity, enabling the system to communicate with devices on the network, including the mock server.

Following the successful network connection, a test request was crafted according to the service created in the previous section for the mock server. This request was sent to the mock server over the established Wi-Fi connection.

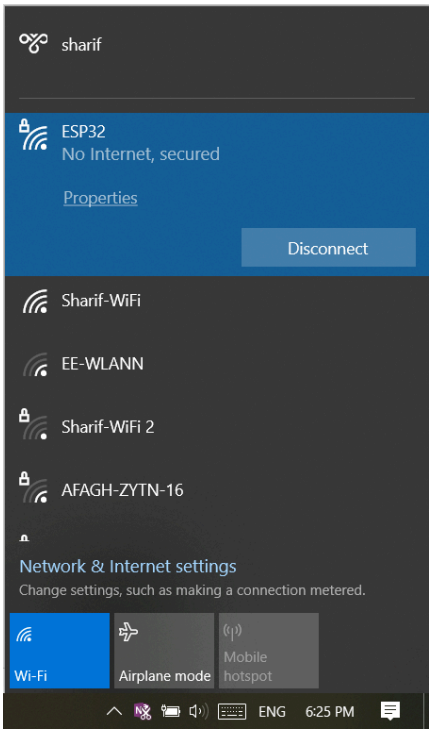
After sending the test request, the serial output was captured and printed for analysis. The serial output contains valuable information about the communication between the system and the mock server. Any errors, response data, or status information will be displayed in the serial output.





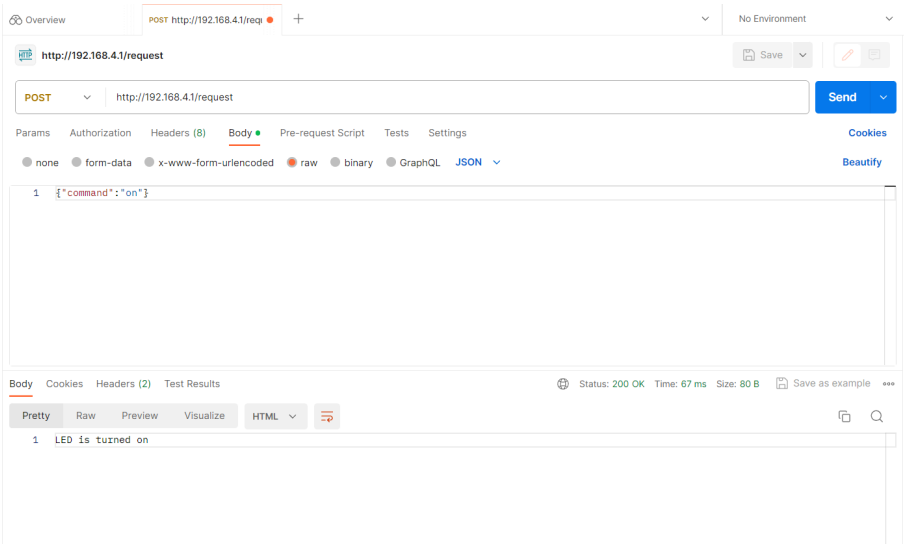
3.ESP32 as Server:

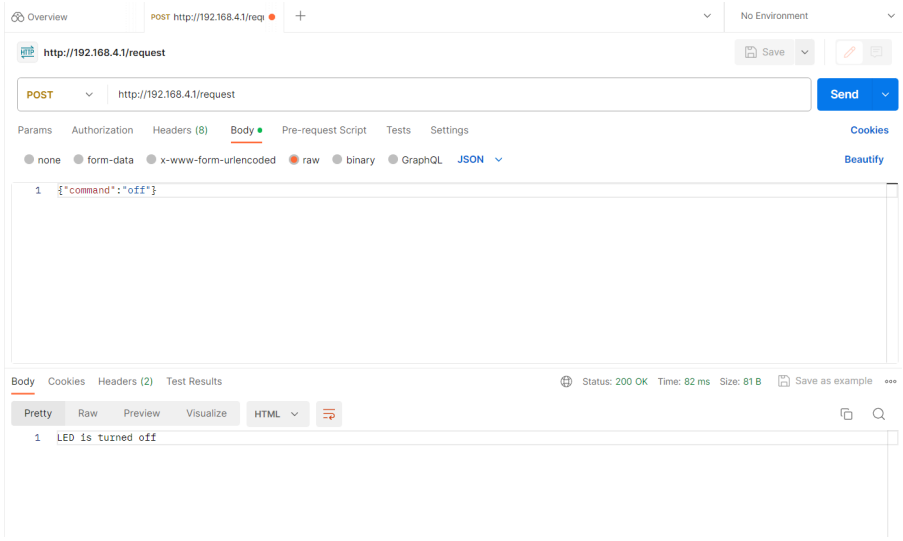
Initially, the ESP module was configured to create a Fi-Wi hotspot, allowing other devices to connect to its network. This is achieved through programming the ESP to function as an access point (AP) and set up a Wi-Fi hotspot.

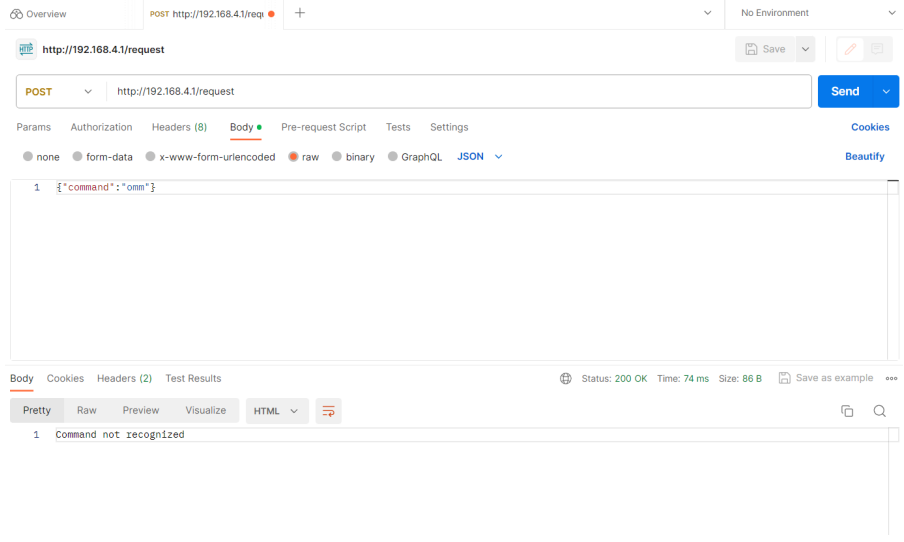


The ESP module was programmed to act as a server, listening for HTTP requests on port 80. The server's primary task involves controlling the state of an LED on the board in response to specific requests.

A client device connected to the ESP Wi-Fi network was used to send HTTP requests to the ESP server, commanding the LED to turn on or off. The serial output was utilized to report the results of the LED control commands. The serial output provides insights into the server's response to incoming requests, indicating the success or failure of the LED control operations.


Overview
POST http://192.168.4.1/requ
No Environment
http://192.168.4.1/request
Save
POST
http://192.168.4.1/request
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
1 {"command":"on"}
Body
Cookies
Headers (2)
Test Results
Status: 200 OK
Time: 67 ms
Size: 80 B
Save as example
Pretty
Raw
Preview
Visualize
HTML
1 LED is turned on


Overview
POST http://192.168.4.1/requ
No Environment
http://192.168.4.1/request
Save
POST
http://192.168.4.1/request
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
1 {"command":"off"}
Body
Cookies
Headers (2)
Test Results
Status: 200 OK
Time: 82 ms
Size: 81 B
Save as example
Pretty
Raw
Preview
Visualize
HTML
1 LED is turned off


Overview
POST http://192.168.4.1/requ
No Environment
http://192.168.4.1/request
Save
POST
http://192.168.4.1/request
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
1 {"command":"omm"}
Body
Cookies
Headers (2)
Test Results
Status: 200 OK
Time: 74 ms
Size: 86 B
Save as example
Pretty
Raw
Preview
Visualize
HTML
1 Command not recognized

```
ESP-IDF 4.3 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-94df9c5339738bd38b5e758c3b44ff47 - python.exe "C:\Espressif\frameworks\esp-idf...
I (33830) wifi_softAP: ====================================
I (41750) wifi_softAP: ========== RECEIVED DATA ==========
I (41750) wifi_softAP: "{"command": "off"}"
I (41750) wifi_softAP: ====================================
I (67610) wifi_softAP: ========== RECEIVED DATA ==========
I (67610) wifi_softAP: {"command":"off"}
I (67610) wifi_softAP: ====================================
I (75670) wifi_softAP: ========== RECEIVED DATA ==========
I (75670) wifi_softAP: {"command":"off"}
I (75670) wifi_softAP: ====================================
I (86790) wifi_softAP: ========== RECEIVED DATA ==========
I (86790) wifi_softAP: {"command":"on"}
I (86790) wifi_softAP: ====================================
I (93420) wifi_softAP: ========== RECEIVED DATA ==========
I (93420) wifi_softAP: {"command":"off"}
I (93420) wifi_softAP: ====================================
I (166480) wifi_softAP: ========== RECEIVED DATA ==========
I (166480) wifi_softAP: {"command":"on"}
I (166480) wifi_softAP: ====================================
GetOverlappedResult failed (PermissionError(13, 'Access is denied.', None, 5))
Waiting for the device to reconnect
I (189510) wifi_softAP: ========== RECEIVED DATA ==========
I (189510) wifi_softAP: {"command":"on"}
I (189510) wifi_softAP: ====================================
I (226610) wifi_softAP: ========== RECEIVED DATA ==========
I (226610) wifi_softAP: {"command":"off"}
I (226610) wifi_softAP: ====================================
I (265540) wifi:station: 70:1c:e7:5b:18:af leave, AID = 1, bss_flags is 658531, bss:0x3ffb9184
I (265540) wifi:new:<1,0>, old:<1,1>, ap:<1,1>, sta:<255,255>, prof:1
```