Comprehensive Report: Designing Webserver for ESP32 (AP & Sta)

1. Introduction

The objective of this project was to design a webserver using the ESP32 microcontroller that operates in both Access Point (AP) and Station (STA) modes. The webserver enables users to control an RGB LED by inputting RGB values through a web interface. Additionally, the system integrates a DHT11 sensor to display real-time temperature and humidity data on the web page. An OLED display was also utilized to show messages based on user interactions. Another web page was designed to perform custom tasks, extending the ESP32's functionalities. All web pages were styled using CSS for an attractive interface.

2. Hardware and Software Requirements

Hardware:

- ESP32-S3 microcontroller
- RGB LED
- DHT11 sensor
- OLED display
- Breadboard and jumper wires

Software:

- Thonny IDE
- GitHub for version control

3. Web Interface Design

Main Web Page:

- Input fields for Red (R), Green (G), and Blue (B) values.
- A 'Submit' button to send RGB values to the ESP32.
- Display of RGB values currently set on the RGB LED.
- Real-time display of temperature and humidity data from the DHT11 sensor.
- Stylish design using custom CSS.

Additional Web Page:

- A custom task page allowing users to toggle an LED and view system logs.
- Interactive buttons to trigger ESP32 actions.
- Styled with CSS for a professional look.

4. Sensor Integration

DHT11 Sensor:

- Connected to GPIO 4 of the ESP32.
- Reads temperature and humidity values every 2 seconds.
- Values sent to the web page using HTTP responses.
- Displayed with appropriate units (°C and %).

5. OLED Display

Integration:

- OLED connected to GPIO 9 (SDA) and GPIO 8 (SCL).
- Displays user-submitted RGB values and status messages.
- Shows sensor data in real-time.
- Utilized Adafruit_SSD1306 library for smooth text rendering.

6. ESP32 Modes

Access Point (AP) Mode:

- ESP32 hosts its own network.
- Users connect directly to the ESP32's Wi-Fi.
- Web pages served without the need for an external router.

Station (STA) Mode:

- ESP32 connects to a local Wi-Fi network.
- Web pages accessed via the local IP assigned by the router.

7. GitHub Version Control

Each task was pushed to a shared GitHub repository with clear commit messages:

- Initial setup and web server creation
- RGB LED control implementation
- DHT11 sensor integration
- OLED display functionality
- Additional task page design
- Final code cleanup and CSS enhancements

9. Testing and Observations

Without Interrupts:

- RGB LED response was slower.
- Sensor readings delayed due to sequential task execution.

With Interrupts:

- Immediate LED response.
- Smooth sensor updates.
- OLED refresh rate optimized.

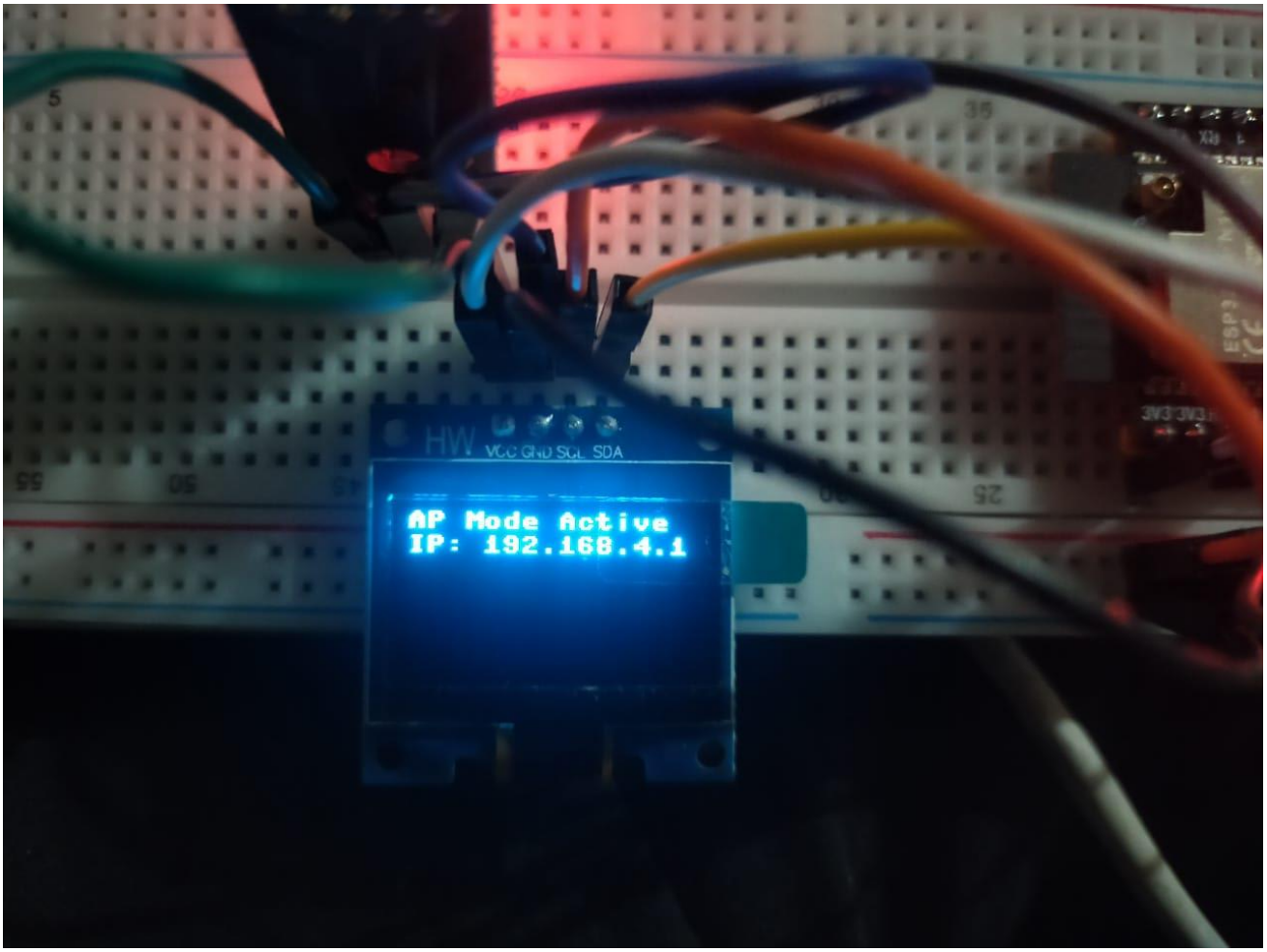
10. Conclusion

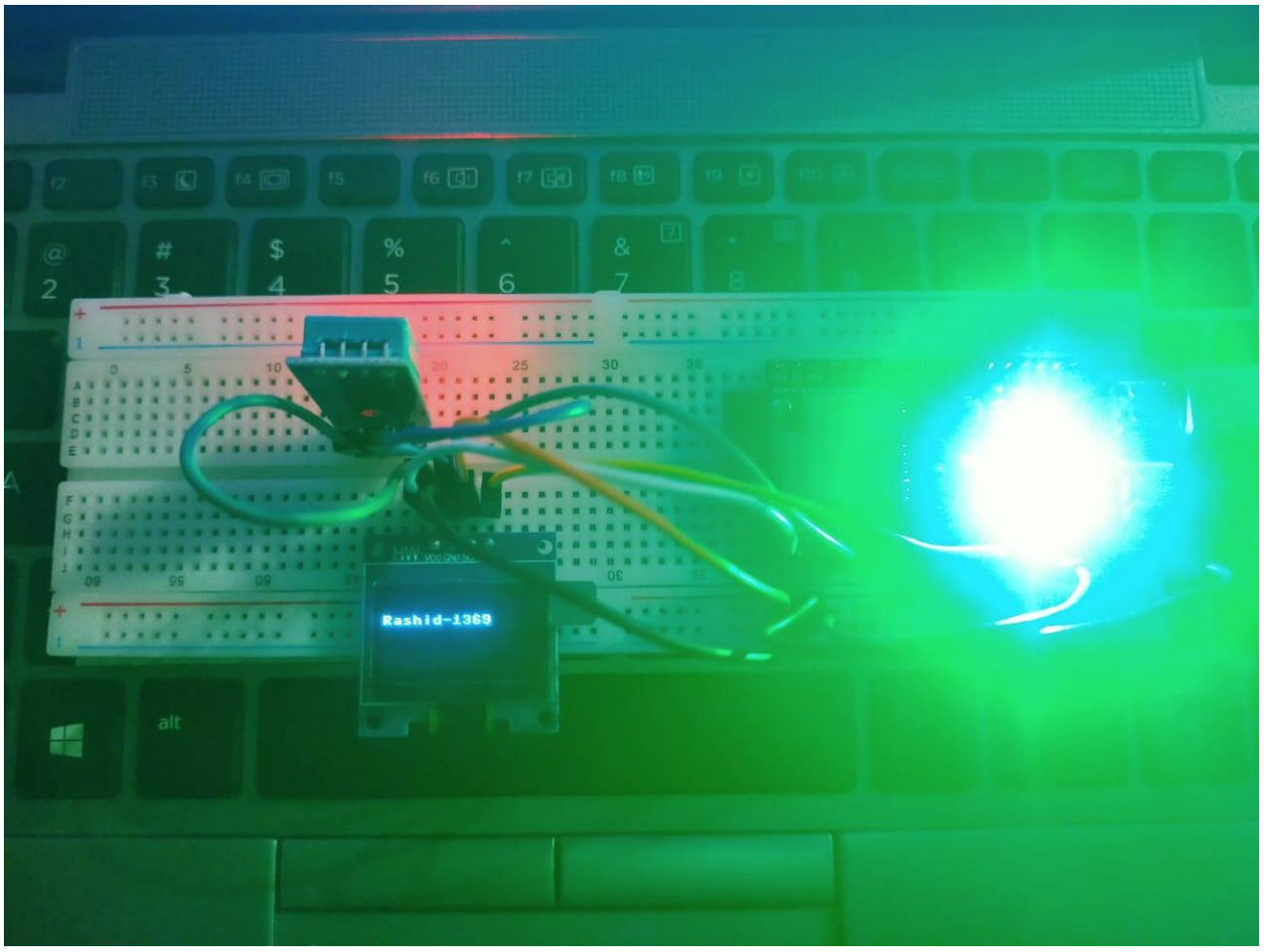
The project successfully achieved the development of an ESP32-based webserver operating in both AP and STA modes. The integration of RGB LED control, real-time sensor data display, and OLED interfacing provided a comprehensive understanding of IoT web servers. The use of CSS ensured a user-friendly interface. Each group member actively contributed to designing, coding, and testing the system.

12. Appendix

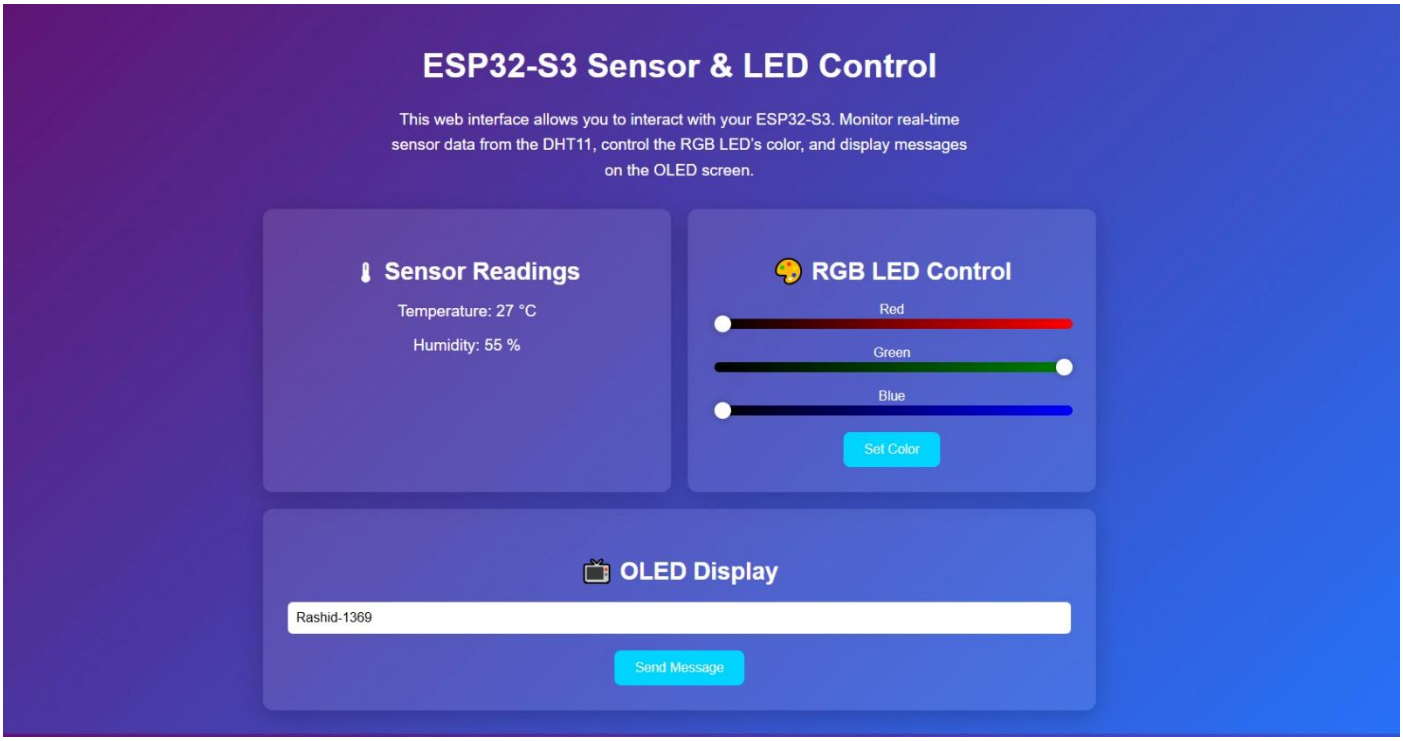
- **Source Code (provided in GitHub repository)**

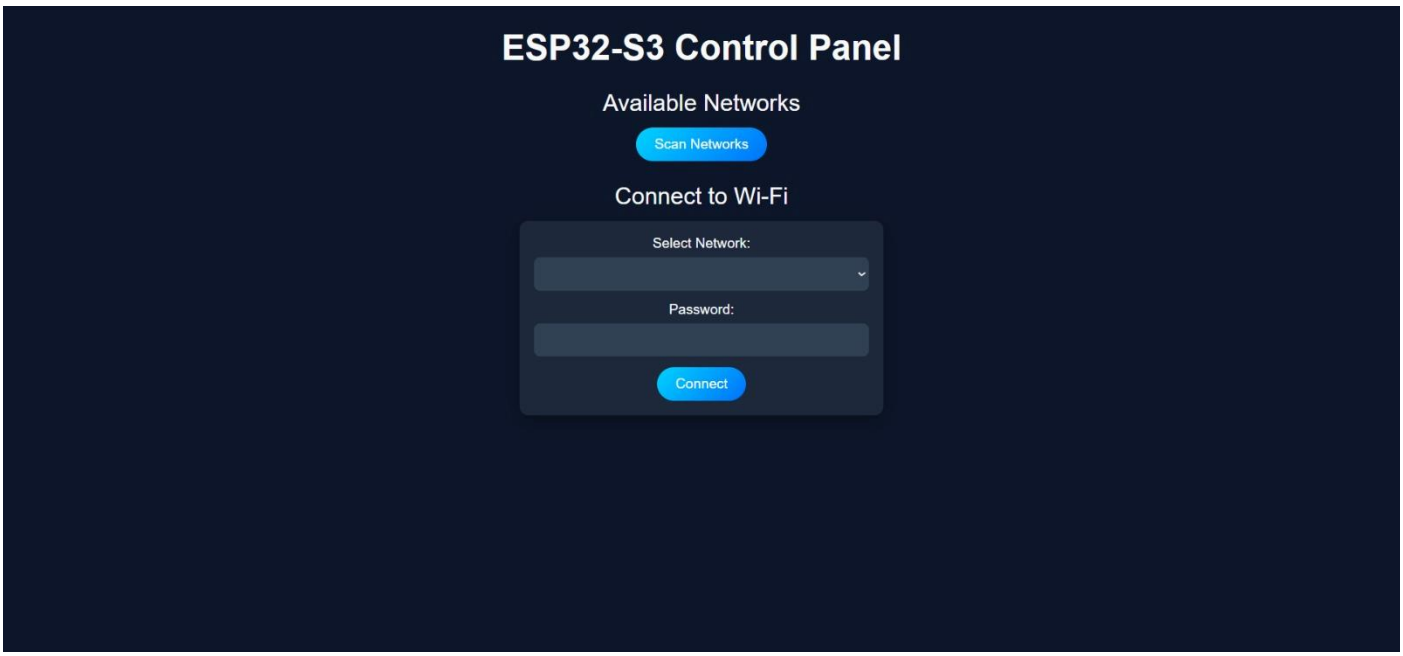
- **Circuit Diagram**

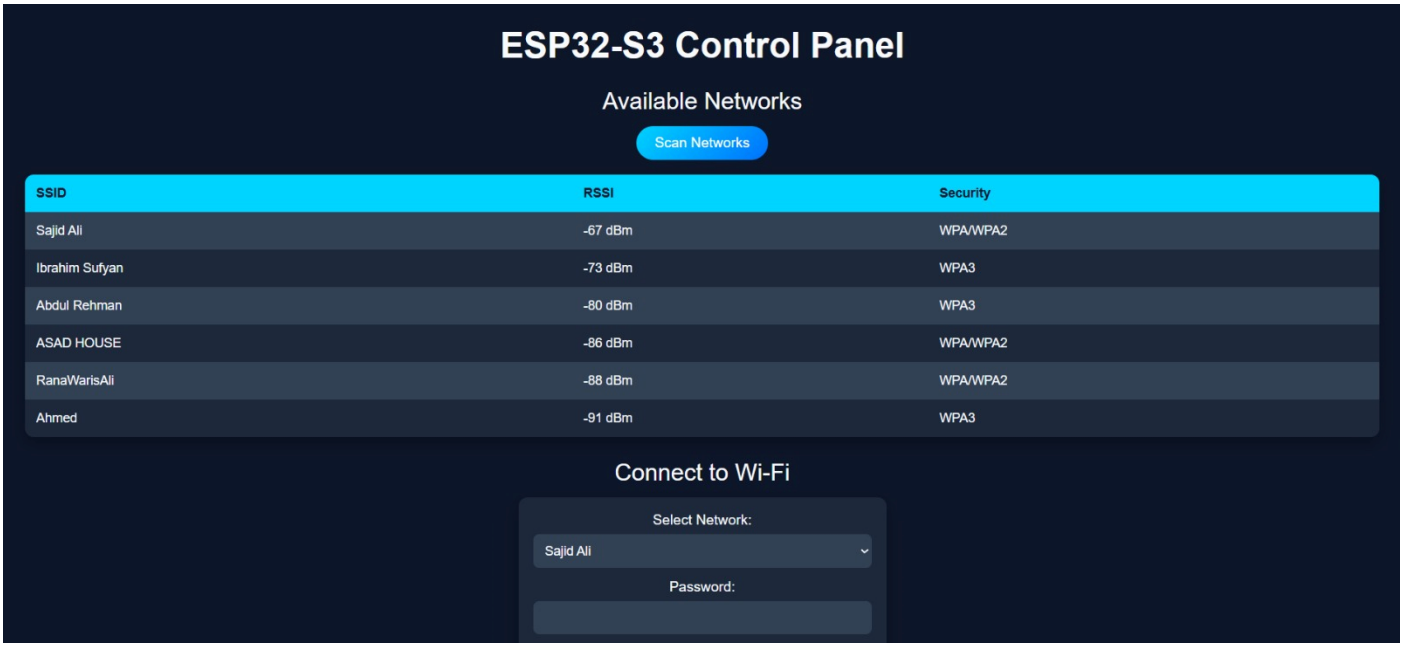

VCC GND SCL SDA
AP Mode Active
IP: 192.168.4.1

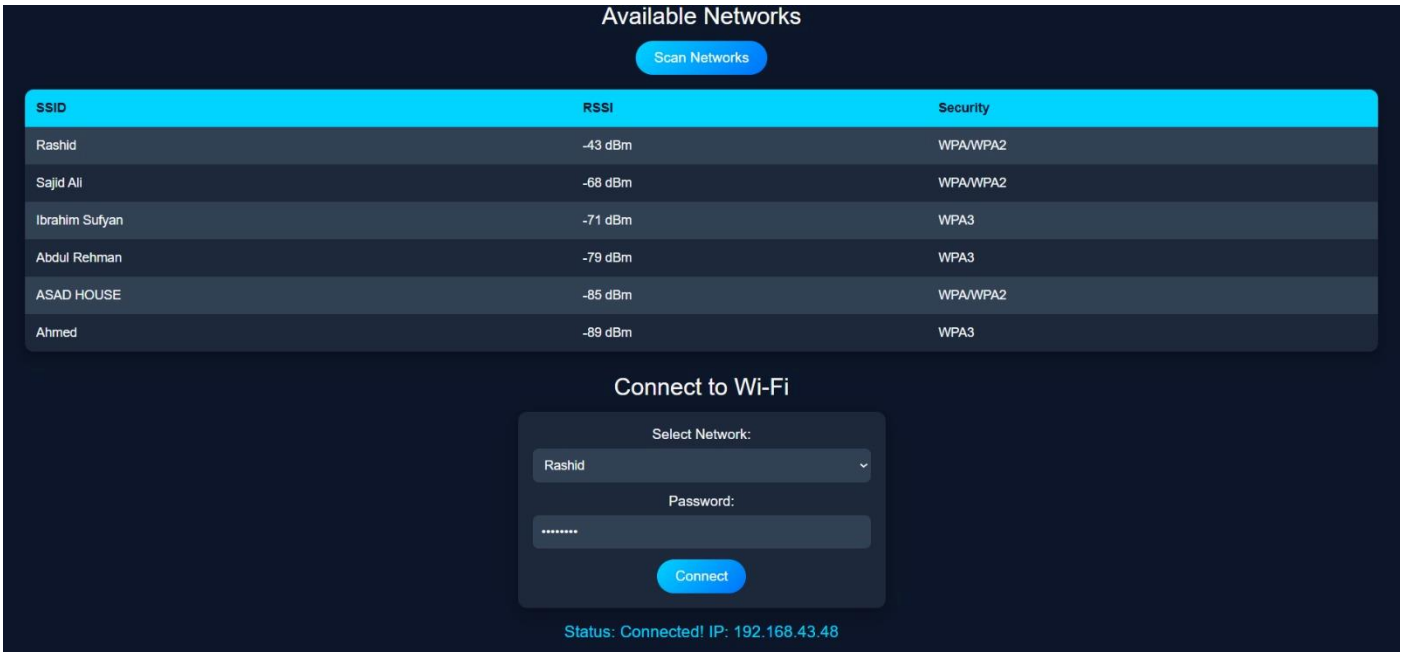

Rashid-1369

- **Screenshots of web interfaces**






ESP32-S3 Control Panel
Available Networks
Scan Networks
SSID
RSSI
Security
Sajid Ali
-67 dBm
WPA/WPA2
Ibrahim Sufyan
-73 dBm
WPA3
Abdul Rehman
-80 dBm
WPA3
ASAD HOUSE
-86 dBm
WPA/WPA2
RanaWarisAli
-88 dBm
WPA/WPA2
Ahmed
-91 dBm
WPA3
Connect to Wi-Fi
Select Network:
Sajid Ali
Password:


Available Networks
Scan Networks
SSID
RSSI
Security
Rashid
-43 dBm
WPA/WPA2
Sajid Ali
-68 dBm
WPA/WPA2
Ibrahim Sufyan
-71 dBm
WPA3
Abdul Rehman
-79 dBm
WPA3
ASAD HOUSE
-85 dBm
WPA/WPA2
Ahmed
-89 dBm
WPA3
Connect to Wi-Fi
Select Network:
Rashid
Password:
Connect
Status: Connected! IP: 192.168.43.48