KARNATAKA LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG BELAGAVI -590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

A Course Activity Report on

Implementation of WebSockets using ESP32

Submitted for the requirements of 7th semester B.E. in CSE

for **"Networking Programming (18CS72)"**

Submitted by

NAME	USN
Yash Herekar	2GI20CS184

Under the guidance of

Prof. Arundhati Nelli

Asst. Prof., Dept. of CSE

Academic Year 2023 - 2024 (Odd semester)

KARNATAKA LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG BELAGAVI -590008

KARNATAKA, INDIA.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Course Project work titled **"Implementation of Websockets using ESP32"** carried out by **Student Yash Herekar** bearing **USNs: 2GI20CS184** for **NETWORKING PROGRAMMING (18CS72) COURSE** is submitted in partial fulfilment of the requirements for 7th semester B.E. in **COMPUTER SCIENCE AND ENGINEERING,** Visvesvaraya Technological University, Belagavi. It is certified that all corrections/ suggestions indicated have been incorporated in the report.

The course project report has been approved as it satisfies the academic requirements prescribed for the said degree.

Date: 2nd November 2023

Place: Belagavi

Signature of guide

Prof. Arundhati Nelli

Asst., Prof., Dept of CSE

KLS Gogte Institute Of Technology, Belagavi

Marks allocation:

	Batch No.: 1		
1.	Seminar Title: **Implementation of WebSockets using ESP32**	Marks Range	**USN** **2GI20CS184**
2.	Abstract (PO2)	0-2	
3.	Application of the topic to the course (PO2)	0-3	
4.	Literature survey and its findings (PO2)	0-4	
5.	Methodology, Results and Conclusion (PO1, PO3, PO4)	0-6	
6.	Report and Oral presentation skill (PO9, PO10)	0-5	
	Total	20	

*** 20 marks is converted to 10 marks for CGPA calculation**

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialisation to the solution of complex engineering problems.
2. **Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and Engineering sciences.
3. **Design/Development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognise the need for and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological channel

Signature of Staff

ABSTRACT

The integration of web technologies and microcontrollers has opened up a world of possibilities for IoT applications, where real-time communication and data exchange are essential. This project explores the utilisation of WebSocket technology with the ESP32 microcontroller, seamlessly blending HTML, CSS, JavaScript, the Arduino framework, and various WiFi libraries to create a dynamic and interactive web-based interface.

This research project delves into several key areas, including the HTTP protocol, the Ajax model, and the power-efficient capabilities of the ESP32 microcontroller. The use of Web Sockets allows for bidirectional communication between the ESP32 and a web browser in real-time, providing a responsive and dynamic user experience.

The project leverages HTML and CSS to create an intuitive web interface, while JavaScript and the Ajax model are employed to establish asynchronous communication with the ESP32. The Arduino framework facilitates the programming of the ESP32 microcontroller, making it accessible to a wider audience of developers.

In addition, the project explores various WiFi libraries, highlighting their role in establishing and maintaining connections between the ESP32 and the web application, ensuring data exchange with minimal latency.

This research delves into the intricate interplay of these technologies to showcase how they can collectively enable a seamless and responsive IoT system, suitable for a range of applications, from smart home automation to industrial monitoring. The findings of this project contribute to the growing field of IoT and provide valuable insights for developers and researchers aiming to leverage the power of Web Sockets for real-time communication with ESP32 microcontrollers.

CONTENTS

Implementing Web Sockets using ESP32

1. Introduction

In the rapidly evolving landscape of the Internet of Things (IoT), the synergy between web technologies and microcontrollers has become pivotal, ushering in an era of innovative and interconnected devices.

One of the key technologies driving this revolution is Web Sockets, enabling real-time, bidirectional communication between web browsers and microcontrollers. This integration is particularly evident in the context of the ESP32 microcontroller, renowned for its versatility and efficiency in IoT applications.

In this project, HTML, CSS, and JavaScript provide the backbone for creating interactive web interfaces, while the Arduino framework simplifies microcontroller programming. Additionally, various WiFi libraries facilitate seamless connectivity, ensuring efficient data exchange.

This project explores the fusion of these elements, delving into the intricacies of Web Sockets, the HTTP protocol, and the Ajax model. By examining these components, this research aims to unlock the potential of ESP32 microcontrollers, offering insights into creating responsive and dynamic IoT applications. Through this exploration, a deeper understanding of the interplay between these technologies emerges, paving the way for the development of innovative solutions in the realm of IoT.

2. Hardware Used

1. ESP32 Microcontroller
2. Leds
3. Resistors
4. Breadboard

2.2. ESP32 Microcontroller

Using an ESP32 as a server for a web-based IoT application is a powerful and versatile approach. The ESP32 microcontroller, equipped with built-in Wi-Fi capabilities, allows you to create a compact and efficient server that can interface with various IoT devices and web clients.

ESP32 Specifications:

- Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz.
- 520 KB of SRAM, 448 KB of ROM and 16 KB of RTC SRAM.
- Supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 Mbps.
- Support for both Classic Bluetooth v4.2 and BLE specifications.
- 34 Programmable GPIOs.
- Up to 18 channels of 12-bit ADC and 2 channels of 8-bit DAC
- Serial Connectivity include 4 x SPI, 2 x I2C, 2 x I2S, 3 x UART.
- Ethernet MAC for physical LAN Communication (requires external PHY).

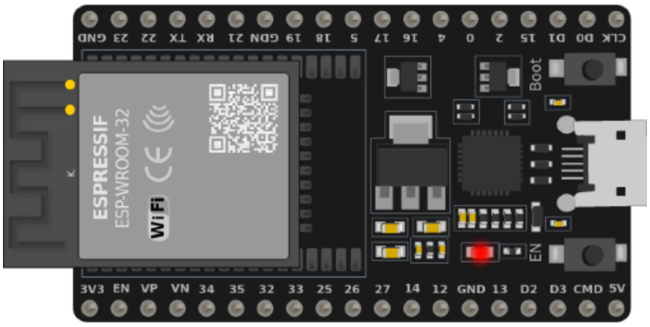

Fig 2.1 ESP32 Microcontroller

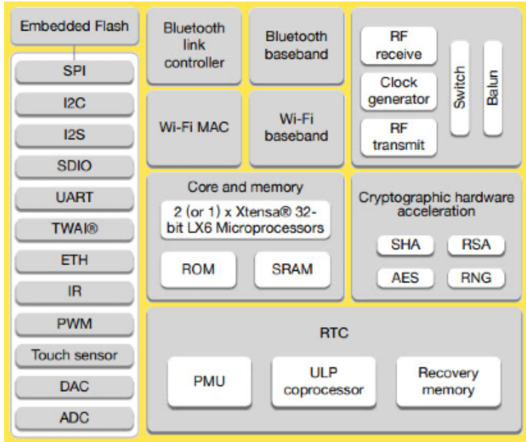

Fig 2.2 ESP32 Architecture

2.3. LED's

When demonstrating WebSocket functionality in web development, the colours **red**, **blue**, and **RGB** are used as the primary means of demonstration.

An led is connected in series with the resistor with a suitable value.

The polarity must be considered while wiring the LED, as its a diodes, connecting it in the reverse will destroy the LED.

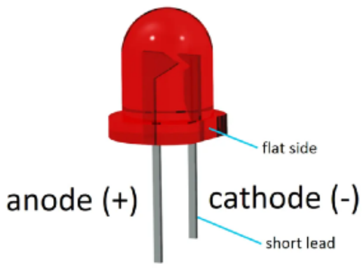

Fig 2.2 LED

2.4. Resistors

Resistors of suite value are used, referring the table given below we can calculate the exact value of the resistor for blue, red and RGB led

The formula given below can be used to calculate the exact value of the resistor

$$R = \frac{V_s - V_f}{I_f}$$

The current rating of each led is around **20 mA**

Resistor value can be calculated using the Vf given in the table below.

- BLUE Led = 15 Ω 
- RED Led = 75 Ω 
- RGB Led = 15 Ω 

LED Color	Typical Vf Range
Red	1.8 to 2.1
Amber	2 to 2.2
Orange	1.9 to 2.2
Yellow	1.9 to 2.2
Green	2 to 3.1
Blue	3 to 3.7
White	3 to 3.4

Fig 2.3 Resistor forward voltage table

2.5. Breadboard

A bread board is used for prototyping the whole project, all the components are placed on it without soldering.

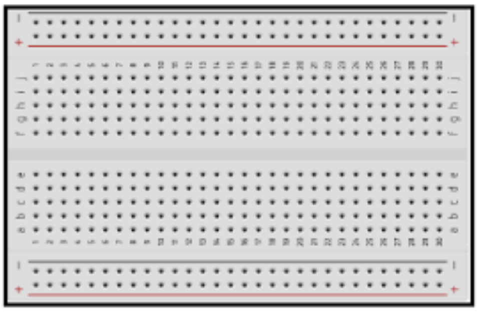

Fig 2.5 Breadboard

3. Connection Details

RGB led is connected to pin 21 with a resistor of 15 Ω

Blue Led is connected to pin 22 with a resistor is 75 Ω

Red Led is connected to pin 23 with a resistor of 15 Ω

Vcc is connected to 3.3v

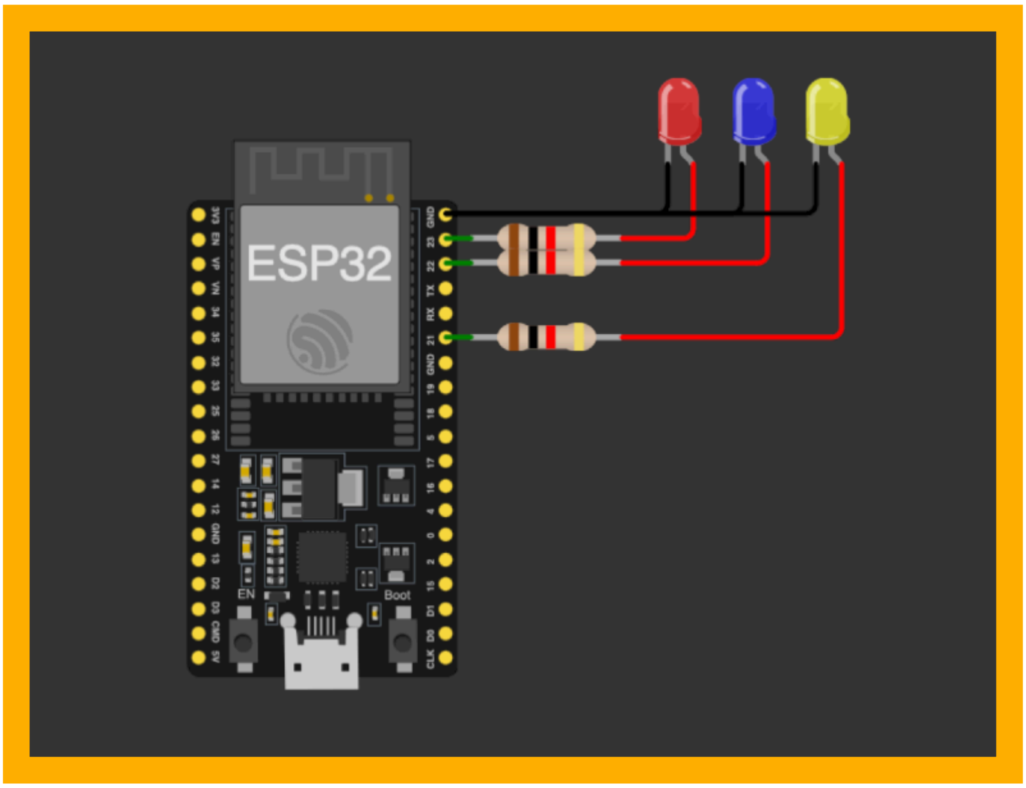

Fig 3.1 Connection details

4. Connection Protocols

1. HTTP Model
2. AJAX Model
3. Web sockets Model

4.2. HTTP Model

Request-Response Architecture: HTTP is a request-response protocol that operates on a client-server architecture. A client (typically a web browser) sends an HTTP request to a web server, which then processes the request and sends back an HTTP response.

The HTTP protocol is well-suited for serving static web content due to its efficiency, simplicity, scalability, caching capabilities, compatibility, and stateless nature

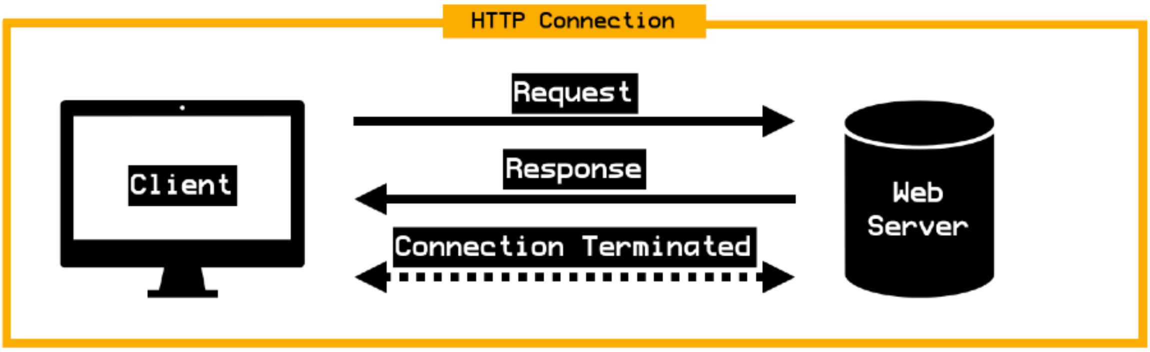

Fig 4.1 HTTP model

4.3. Ajax Model

Asynchronous Communication: The AJAX (Asynchronous JavaScript and XML) model is a powerful and flexible approach to web communication, often seen as an enhancement to the standard HTTP model. It enables web applications to exchange data with a web server in the background without requiring a full page refresh

While AJAX offers many benefits, it's not necessarily "better" than HTTP; rather, it complements HTTP by providing a means to enhance the user experience in specific use cases.

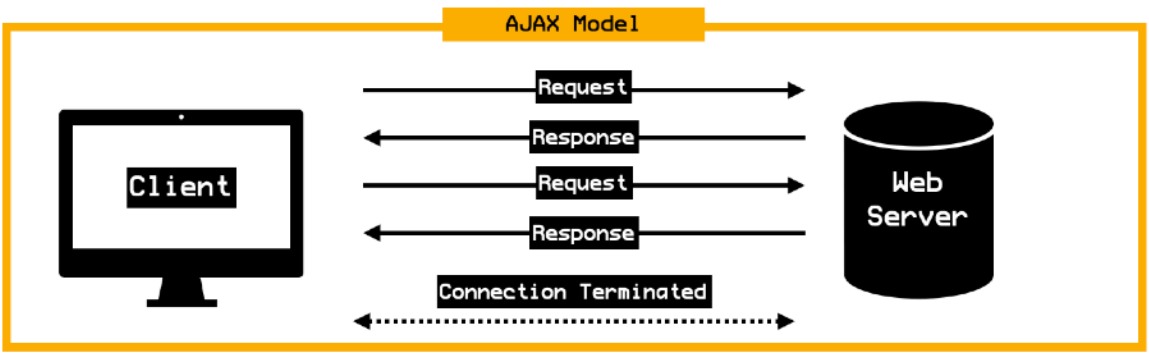

Fig 4.2 Ajax Model

4.4. WEBSOCKETS MODEL

Persistent connection: Web Sockets are a critical technology in the realm of real-time communication, and they excel in time-critical IoT (Internet of Things) applications. Unlike traditional HTTP and AJAX, Web Sockets offer distinct advantages when it comes to low-latency and real-time data transfer.

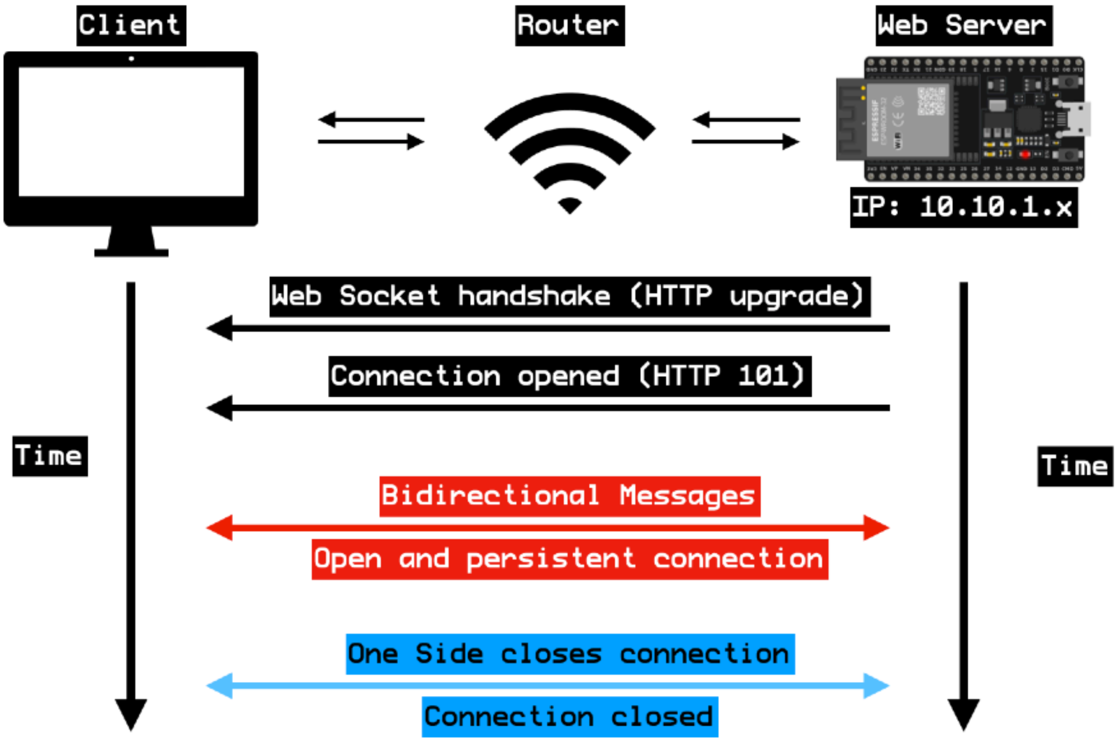

Fig 4.3 Web sockets

- **Full-Duplex Communication:** Web Sockets allow simultaneous data transfer in both directions, ideal for real-time IoT.
- **Asynchronous and Event-Driven:** They enable immediate responses and real-time updates in IoT applications.
- **Low Latency:** Designed for minimal delay, essential in time-critical IoT scenarios
- **Efficient Data Transfer:** Minimal overhead and efficient bandwidth use.
- **Reduced Server Load:** Scalable for managing numerous IoT devices efficiently.
- **Stateful Connection:** Maintains ongoing communication without reestablishing connections.

Web Sockets excel in real-time and interactive applications, such as chat applications, live data feeds, online gaming, and IoT systems where low latency and immediate feedback are crucial. However, it's important to note that each technology has its place, and the choice between Web Sockets, HTTP, or AJAX depends on the specific requirements of the project.

5. Working and Steps involved

Step1: Scan through the list of networks

```
Number of networks found: 2
Network name: EternalBlue
Signal strength: -43
MAC address: 8C:A3:99:C3:4B:3C
Encryption type: WPA2_PSK
-----------------------
Network name: AGNELO
Signal strength: -91
MAC address: B0:BE:76:A9:29:BE
Encryption type: WPA2_PSK
-----------------------
Establishing connection to WiFi.
...................
```

Fig 5.1 Scanning networks

Step2: Scan through the list of WiFi and connect

```
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1184
load:0x40078000,len:13260
load:0x40080400,len:3028
entry 0x400805e4
Establishing connection to WiFi with SSID: EternalBlue
.Connected to network with IP address: 192.168.29.51
```

Fig 5.2 Establishing connection with the network

Step3: Create a Web Server host it using the IP address

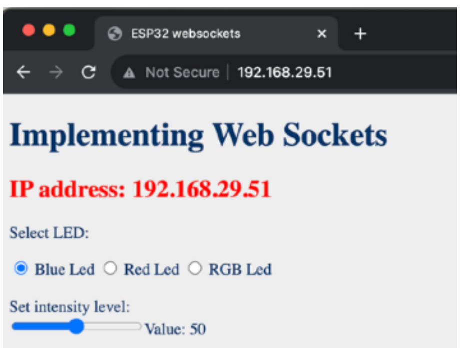

Fig 5.3 Web Server

STEP4: CREATE WEB SOCKETS TO CREATE PERSISTENT CONNECTIONS ADDRESS

```
#include <WiFi.h>                                     // needed to connect to WiFi
#include <WebServer.h>
#include <WebSocketsServer.h>

// Initialisation of web server and web socket
WebServer server(80);                                 // the server uses port 80
WebSocketsServer webSocket = WebSocketsServer(81);    // the web socket uses port 81

void setup() {
  Serial.begin(115200);                               // init serial port

  server.on("/", []() {
    server.send(200, "text/html", website);
  });

  server.begin();                                     // start server
  webSocket.begin();                                  // start web socket
}
```

STEP5: USE JSON TO TRANSFER DATA

```
// Simple function to send information to the web clients
void sendJson(String l_type, String l_value) {
    String jsonString = "";                           // create a JSON string
    StaticJsonDocument<200> doc;                      // create JSON container
    JsonObject object = doc.to<JsonObject>();         // create a JSON Object
    object["type"] = l_type;
    object["value"] = l_value;
    serializeJson(doc, jsonString);                // convert JSON object to string
    webSocket.broadcastTXT(jsonString);
}
```

6. OUTPUT

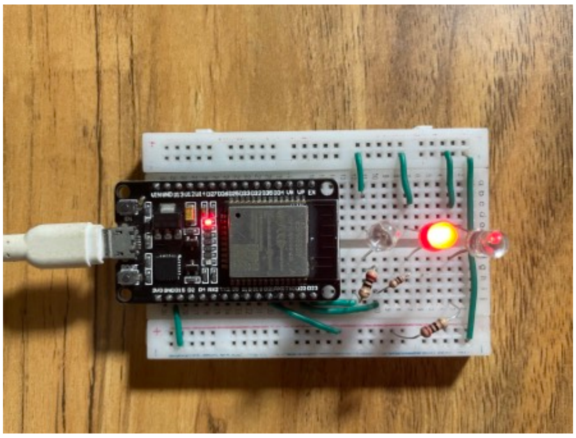

Fig 6.1 Working demonstration of web server

Fig 6.2 Working of the website

7. Conclusion

In conclusion, this project has explained Web Sockets and unveiled their significant role in revolutionising real-time communication within IoT applications. By exploring the asynchronous and low-latency capabilities of Web Sockets, we've gained a profound understanding of their superiority over traditional HTTP and AJAX models, particularly in time-critical IoT scenarios.

Web Sockets have proven themselves as a game-changer in the world of IoT, offering instant feedback, real-time data updates, and efficient bandwidth utilisation. Their ability to establish persistent, full-duplex connections and efficiently manage concurrent clients has made them a cornerstone technology for applications that demand immediate responses and dynamic, low-latency interactions.

As IoT continues to grow and permeate various aspects of our lives, the potential of Web Sockets in enhancing the responsiveness and efficiency of IoT systems cannot be overstated. From emergency alerts to remote monitoring and control, Web Sockets serve as the bridge between devices and the digital world, optimising the user experience and ensuring that time-critical data is handled with the swiftness and efficiency it demands.

In a rapidly evolving landscape where IoT applications are becoming increasingly integral to our daily lives, the adoption of Web Sockets holds great promise. This project's exploration has shed light on the capabilities and advantages of Web Sockets in IoT, paving the way for the development of even more responsive and dynamic IoT applications in the future.

8. References

https://github.com/mo-thunderz/Esp32WifiPart1

https://github.com/mo-thunderz/Esp32WifiPart2

https://github.com/mo-thunderz/Esp32WifiPart3

https://www.digikey.in/en/resources/conversion-calculators/conversion-calculator-led-series-resistor