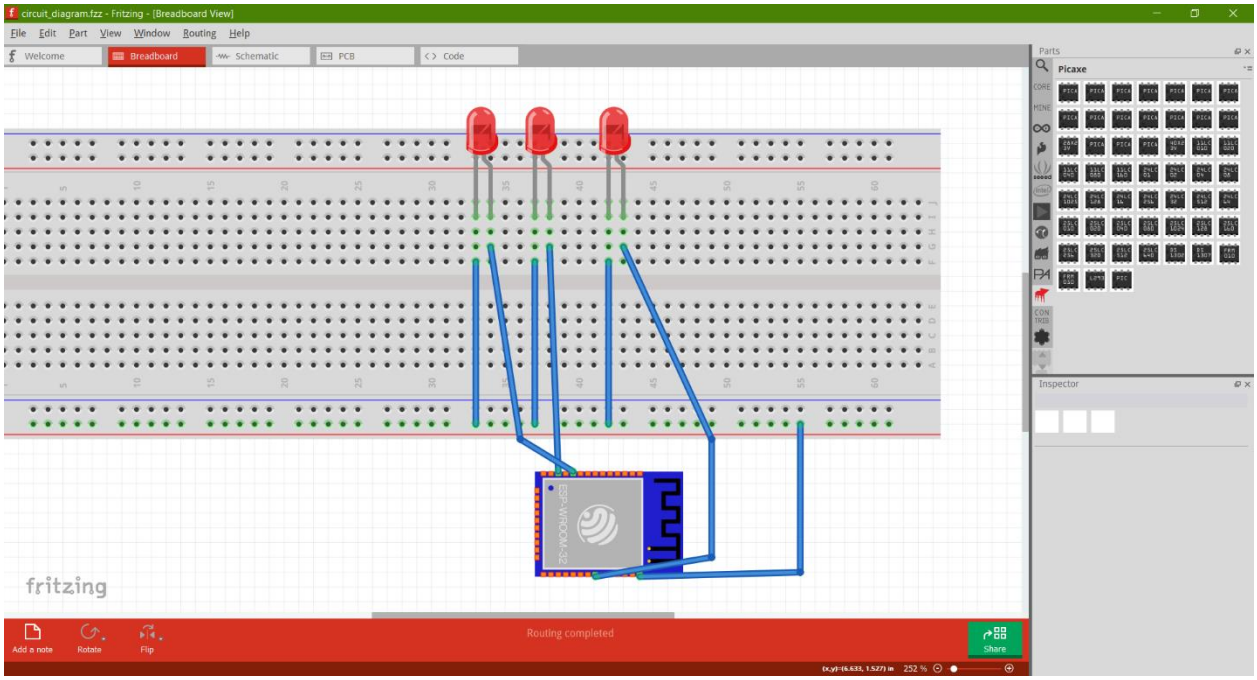
Project statement: Design a lightweight web server to control three LEDs on Local Network.

Aim: To design a lightweight web server to control three LEDs on Local Network.

Material used:

- ESP-32 micro-controller
- 3-LED's
- 10-jumper wires
- 1-Bread board
- Good Wi-Fi-connection for local network

Circuit diagram:



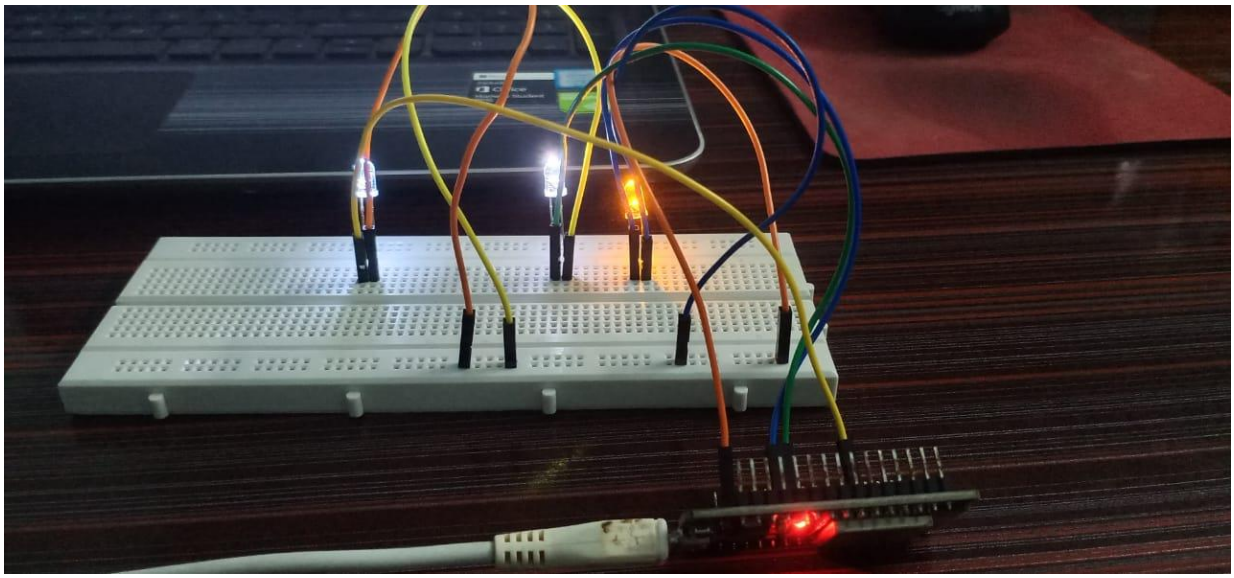
This is diagram is made in Fritzing software. In this diagram three led's and ESP-32 micro-controller is used. Connections are made as per the code. Cathode pins of the led's are connected to GRD and anode pins of the led's are connected the GPIO pins of ESp-32. And ESP32 is connected to a laptop via an USB cable.

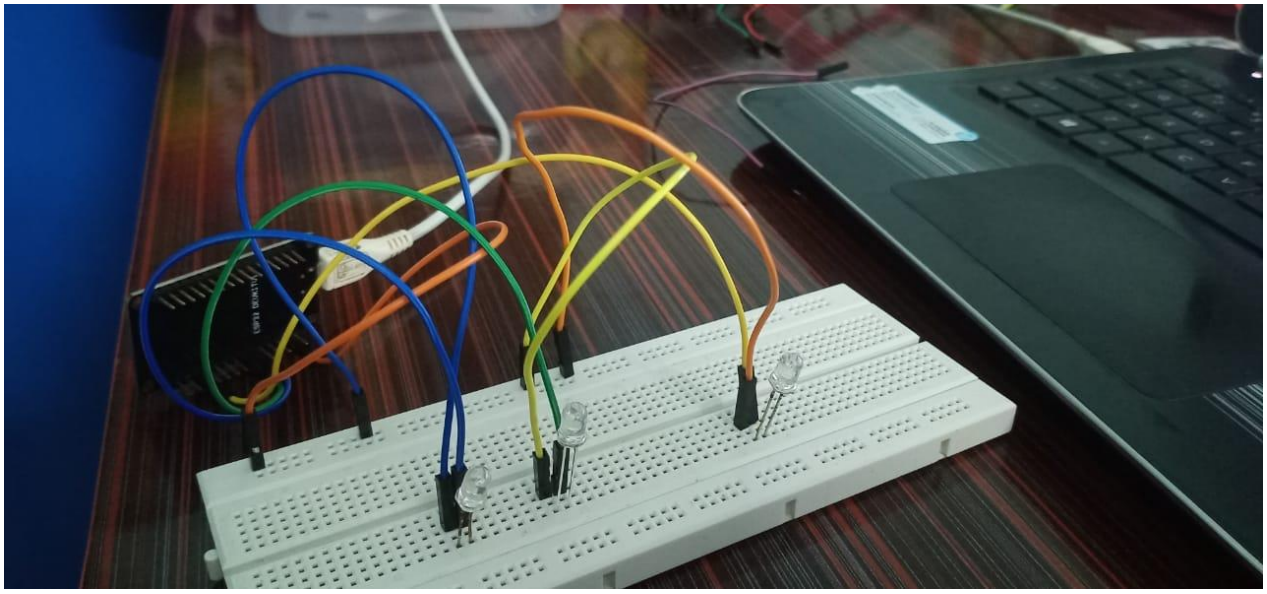
Procedure:

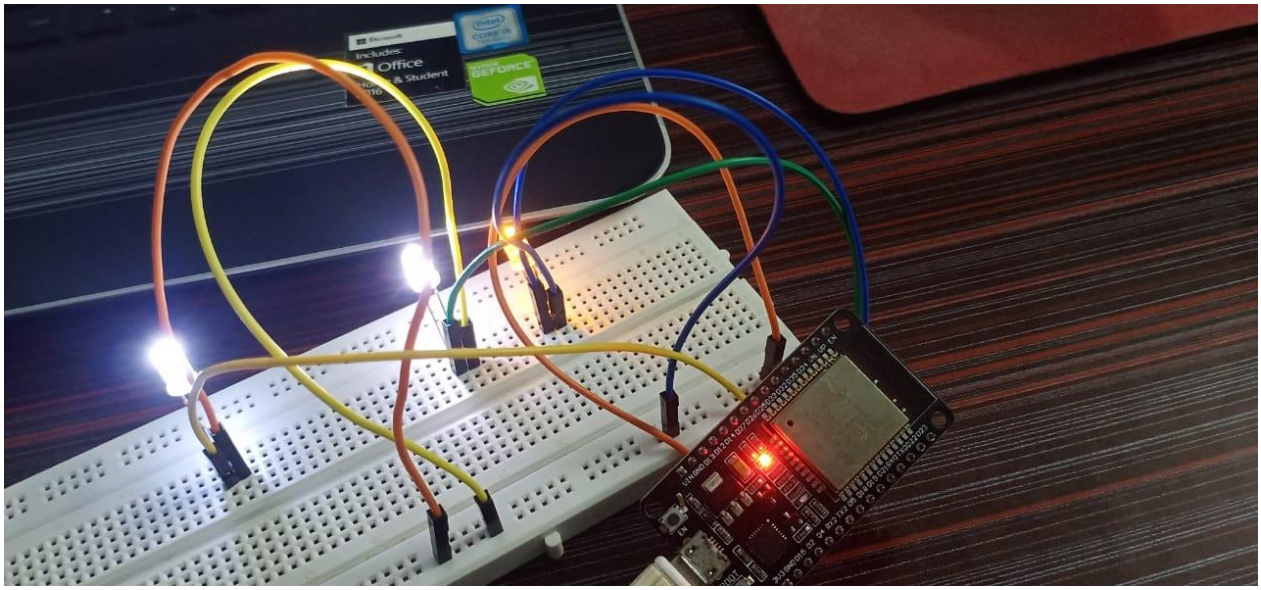
1. Write the code in Arduino IDE(software) for connecting to the microcontroller to Wi-Fi, to get the IP address, to turn on and turn off the led's through web server in local network and HTML code for the web page.
2. Connect the ESP32 to the anode of the led's with their respective pins assigned in the code
3. Ground the cathode pins of the led's to the GRD pin in the ESP32
4. Connect ESP32 to laptop via USB cable and select proper port and baud rate in order to avoid disturbances.
5. Verify and upload the code in ESP32 and your IP address will be sent to you through serial monitor
6. Check your network, all devices should be connected to single network

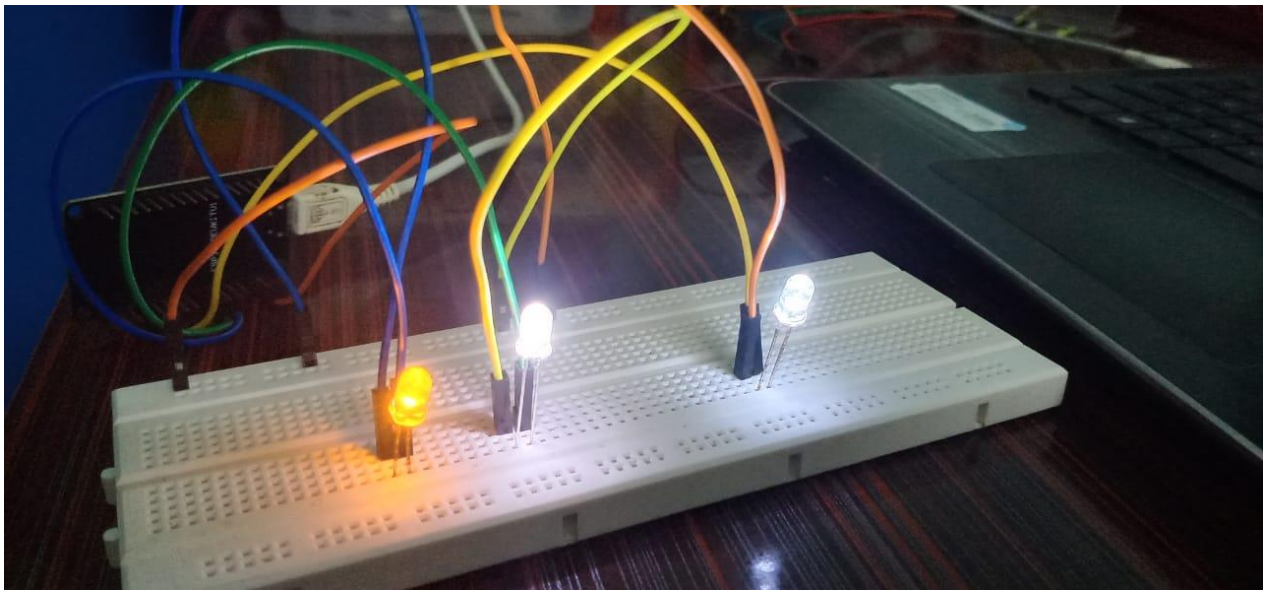
7. Type that IP address in your web browser and it will open a HTML page for turning on/off the led’s
8. Finally, now you can switch off/on the led’s

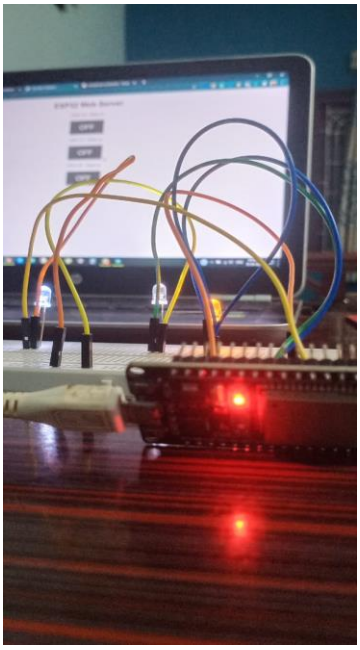
Pictures of this project:

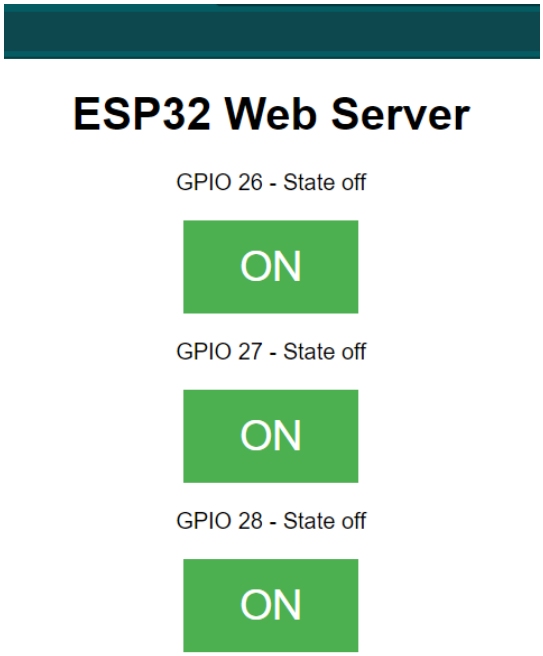


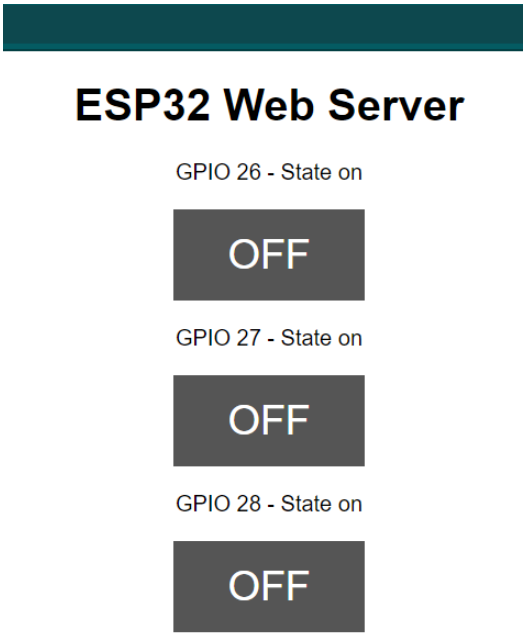



Includes:
Office
& Student
GEFORCE









Code for this project:

```
#include <WiFi.h>
const char* ssid     = " {your wifi name}";
const char* password = " {your wifi password}";

WiFiServer server(80);
String header;

String output26State = "off";
String output27State = "off";
String output28State = "off";

const int output26 = 3;
const int output27 = 17;
const int output28 = 16;

void setup()
{
  Serial.begin(115200);
  pinMode(output26, OUTPUT);
  pinMode(output27, OUTPUT);
  pinMode(output28, OUTPUT);

  digitalWrite(output26, LOW);
  digitalWrite(output27, LOW);
  digitalWrite(output28, LOW);

  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
```

```
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  server.begin();
}

void loop()
{
  WiFiClient client = server.available();
  if (client)
  {
    Serial.println("New Client.");
    String currentLine = "";
    while (client.connected())
    {
      if (client.available())
      {
        char c = client.read();
        Serial.write(c);
        header += c;
        if (c == '\n')
        {
          if (currentLine.length() == 0)
          {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println("Connection: close");
            client.println();

            if (header.indexOf("GET /26/on") >= 0)
            {
              Serial.println("GPIO 26 on");
              output26State = "on";
              digitalWrite(output26, HIGH);
            }
            else if (header.indexOf("GET /26/off") >= 0)
            {
              Serial.println("GPIO 26 off");
              output26State = "off";
              digitalWrite(output26, LOW);
            }
            else if (header.indexOf("GET /27/on") >= 0)
            {
              Serial.println("GPIO 27 on");
              output27State = "on";
```

```
            digitalWrite(output27, HIGH);
          }
          else if (header.indexOf("GET /27/off") >= 0)
          {
            Serial.println("GPIO 27 off");
            output27State = "off";
            digitalWrite(output27, LOW);
          }
          else if (header.indexOf("GET /28/on") >= 0)
          {
            Serial.println("GPIO 28 on");
            output28State = "on";
            digitalWrite(output28, HIGH);
          }
          else if (header.indexOf("GET /28/off") >= 0)
          {
            Serial.println("GPIO 28 off");
            output28State = "off";
            digitalWrite(output28, LOW);
          }

          client.println("<!DOCTYPE html><html>");
          client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-
scale=1\">");
          client.println("<link rel=\"icon\" href=\"data:,\">");

          client.println("<style>html { font-family: Helvetica; display: inline-block; margin: 0px
auto; text-align: center;}");

          client.println(".button { background-color: #4CAF50; border: none; color: white;
padding: 16px 40px;");

          client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}");
          client.println(".button2 {background-color: #555555;}</style></head>");

          client.println("<body><h1>ESP32 Web Server</h1>");

          client.println("<p>GPIO 26 - State " + output26State + "</p>");
          if (output26State=="off")
          {
            client.println("<p><a href=\"/26/on\"><button
class=\"button\">ON</button></a></p>");
          }
          else
          {
            client.println("<p><a href=\"/26/off\"><button class=\"button
button2\">OFF</button></a></p>");
          }

          client.println("<p>GPIO 27 - State " + output27State + "</p>");
          if (output27State=="off")
```

```
          {
            client.println("<p><a href=\"/27/on\"><button
class=\"button\">ON</button></a></p>");
          }
          else
          {
            client.println("<p><a href=\"/27/off\"><button class=\"button
button2\">OFF</button></a></p>");
          }

          client.println("<p>GPIO 28 - State " + output28State + "</p>");
          if (output28State=="off")
          {
            client.println("<p><a href=\"/28/on\"><button
class=\"button\">ON</button></a></p>");
          }
          else
          {
            client.println("<p><a href=\"/28/off\"><button class=\"button
button2\">OFF</button></a></p>");
          }
          client.println("</body></html>");
          client.println();
          break;
        }
        else
        {
          currentLine = "";
        }
      }
      else if (c != '\r')
      {
        currentLine += c;
      }
    }
  }
  header = "";
  client.stop();
  Serial.println("Client disconnected.");
  Serial.println("");
 }
}
```

Result:

Hence, lightweight web server to control three LEDs on Local Network is successfully designed and implemented.