```
#include <WiFi.h>

// Replace with your network credentials
const char* ssid = "xelse 2";
const char* password = "";  // No password for Wokwi-GUEST

// Set web server port number to 80
WiFiServer server(80);

// Variable to store the HTTP request
String header;

// Auxiliary variables to store the current output state
String output12State = "off";
String output14State = "off";

// Assign output variables to GPIO pins
const int output12 = 12;
const int output14 = 14;

// Current time
unsigned long currentTime = millis();
// Previous time
unsigned long previousTime = 0;
// Define timeout time in milliseconds
const long timeoutTime = 2000;

void setup() {
  Serial.begin(115200);

  // Initialize the output variables as outputs
  pinMode(output12, OUTPUT);
  pinMode(output14, OUTPUT);
```

```
  // Set outputs to LOW
  digitalWrite(output12, LOW);
  digitalWrite(output14, LOW);

  // Connect to Wi-Fi network with SSID and password
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  // Print local IP address and start web server
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  server.begin();
}

void loop(){
  WiFiClient client = server.available(); // Listen for incoming clients
  if (client) { // If a new client connects
    currentTime = millis();
    previousTime = currentTime;
    Serial.println("New Client."); // Print a message out in the serial port
    String currentLine = ""; // Make a String to hold incoming data from the client

    while (client.connected() && currentTime - previousTime <= timeoutTime) {
      currentTime = millis();
      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();

    // Turns the GPIOs on and off
    if (header.indexOf("GET /12/on") >= 0) {
      Serial.println("GPIO 12 on");
      output12State = "on";
      digitalWrite(output12, HIGH);
    } else if (header.indexOf("GET /12/off") >= 0) {
      Serial.println("GPIO 12 off");
      output12State = "off";
      digitalWrite(output12, LOW);
    } else if (header.indexOf("GET /14/on") >= 0) {
      Serial.println("GPIO 14 on");
      output14State = "on";
      digitalWrite(output14, HIGH);
    } else if (header.indexOf("GET /14/off") >= 0) {
      Serial.println("GPIO 14 off");
      output14State = "off";
      digitalWrite(output14, LOW);
    }

    // Display the HTML web page
    client.println("<!DOCTYPE html><html>");
    client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");
    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>");

            // Display current state, and ON/OFF buttons for GPIO 12
            client.println("<p>GPIO 12 - State " + output12State + "</p>");
            client.println("<p><a href=\"/12/" + String(output12State == "off" ? "on" : "off") + "\">");
            client.println("<button class=\"button\">" + String(output12State == "off" ? "ON" : "OFF") + "</button></a></p>");

            // Display current state, and ON/OFF buttons for GPIO 14
            client.println("<p>GPIO 14 - State " + output14State + "</p>");
            client.println("<p><a href=\"/14/" + String(output14State == "off" ? "on" : "off") + "\">");
            client.println("<button class=\"button button2\">" + String(output14State == "off" ? "ON" : "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.");
  }
}
```