```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include <LEAmDNS.h>

#ifndef STASSID
#define STASSID "your-ssid"
#define STAPSK "your-password"
#endif

const char* ssid = STASSID;
const char* password = STAPSK;

WebServer server(80);

const int ledPin = 15; // GP15
bool ledState = false; // LEDの状態を保持する変数

/* エラーの原因となったHTML表記
void handleRoot() {
  String html = R"(
    <html>
    <head>
      <script>
        function toggleLED() {
          fetch('/toggle');
        }
      </script>
    </head>
    <body>
      <button onclick="toggleLED()">Toggle LED</button>
    </body>
    </html>
  )";
  server.send(200, "text/html", html);
}
*/

void handleRoot() {
  String html = "<html>\n"
                "<head>\n"
                "  <script>\n"
                "    function toggleLED() {\n"
                "      fetch('/toggle');\n"
                "    }\n"
                "  </script>\n"
                "</head>\n"
                "<body>\n"
                "  <button onclick=\"toggleLED()\">Toggle LED</button>\n"
                "</body>\n"
                "</html>";
  server.send(200, "text/html", html);
```

```
}


void handleToggle() {
  ledState = !ledState; // LEDの状態を切り替える
  digitalWrite(ledPin, ledState); // 新しい状態をセット
  server.send(200, "text/plain", "OK"); // レスポンスを送信
}

void handleNotFound() {
  digitalWrite(ledPin, 1);
  String message = "File Not Found\n\n";
  message += "URI: ";
  message += server.uri();
  message += "\nMethod: ";
  message += (server.method() == HTTP_GET) ? "GET" : "POST";
  message += "\nArguments: ";
  message += server.args();
  message += "\n";
  for (uint8_t i = 0; i < server.args(); i++) {
    message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
  }
  server.send(404, "text/plain", message);
  digitalWrite(ledPin, 0);
}

void setup(void) {
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, ledState);
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.println("");

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  if (MDNS.begin("picow")) {
    Serial.println("MDNS responder started");
  }

  server.on("/", handleRoot);
  server.on("/toggle", handleToggle);
  server.onNotFound(handleNotFound);
```

```
  server.begin();
  Serial.println("HTTP server started");
}

void loop(void) {
  server.handleClient();
  MDNS.update();
}
```