

```

1  //--//Webszerver, fetch(), async/await
2  #include <ESP8266WiFi.h>
3  #include <ESP8266WebServer.h>
4
5  const char* ssid = "BIR_WIFI";
6  const char* password = "OeBir2019";
7
8  ESP8266WebServer server(80);
9  const int ledPin = LED_BUILTIN;
10 bool ledState = false;
11
12 const char MAIN_page[] PROGMEM = R"rawliteral(
13 <!DOCTYPE html>
14 <html>
15   <meta charset="UTF-8">
16   <head>
17     <meta name="viewport" content="width=device-width, initial-scale=1">
18     <title>LED vezérlés</title>
19     <style>
20       body {text-align: center; font-family: Arial; margin: 40px;}
21       h1 {color: #333;}
22       button {background: #202a66; color: white; border: none;
23         padding: 10px 20px; font-size: 16px; border-radius: 5px;
24         cursor: pointer;}
25       .button:hover {background: #000940;}
26     </style>
27   </head>
28   <body>
29     <h1>ESP8266 LED Vezérlés</h1>
30     <button class="button" onclick="toggleLED('on')">LED BE</button>
31     <button class="button" onclick="toggleLED('off')">LED KI</button>
32     <p>ADC állapot: <span id="adcStatus">N/A</span></p>
33     <hr>
34     <input type="number" id="num1" placeholder="Első szám">
35     <input type="number" id="num2" placeholder="Második szám">
36     <p>Összeg: <span id="sumRes">N/A</span></p>
37     <p>Szorzat: <span id="mulRes">N/A</span></p>
38     <button class="button" onclick="calculate()">
39     PARAMÉTEREK LEKÉRDEZÉSE
40   </button>
41   <script>
42     //általános státuszt lekezelő függvény
43     function handleResponseStatus(response){
44       console.log("Válasz státusz:", response.status);
45       return response.text();
46     }
47
48     //általános hibát lekezelő függvény
49     function handleError(error){
50       console.error("Hiba történt:", error);
51     }
52
53     //LED vezérlő
54     function toggleLED(state){
55       let url = '/led?state=' + state;
56       fetch(url)
57         .then(handleResponseStatus) //szerver státusz kiírása
58         .then(getToggleLEDResponse) //adat feldolgozás
59         .catch(handleError);       //hibakezelés
60     }
61     //LED response data feldolgozó
62     function getToggleLEDResponse(data){
63       console.log("Szerver válasz:", data);
64     }
65
66     //aszinkron lekérdezés
67     async function calculate(){
68       try{
69         //érték beolvas
70         let num1value = document.getElementById('num1').value;
71         let num2value = document.getElementById('num2').value;
72
73         if (num1value === "" || num2value === ""){

```

```

74         alert("Add meg mindkét számot!");
75         return;
76     }
77
78     //query építés
79     let params = new URLSearchParams({ num1: num1value, num2: num2value});
80     let url = `/calculate?${params.toString()}`;
81
82     //kérés indítás
83     let response = await fetch(url);
84
85     console.log("HTTP státusz: ", response.status);
86
87     //visszakapott adat jsonné konvertálása
88     let data = await response.json();
89
90     console.log("Kapott adatt:", data);
91
92     //kapott adat megjelenítése
93     document.getElementById("sumRes").innerHTML = data.sumRes;
94     document.getElementById("mulRes").innerHTML = data.mulRes;
95
96     } catch (error){
97         console.error("Hiba történt:", error.message);
98     }
99     }
100
101     function getADCState(){
102         let url = '/adc';
103         fetch(url)
104             .then(handleResponseStatus)
105             .then(getADCResponse)
106             .catch(handleError);
107     }
108
109     function getADCResponse(data){
110         document.getElementById("adcStatus").innerHTML = data;
111         console.log("Szerver válasz -> ADC érték:", data);
112     }
113
114     function startAutoFetch(){
115         setInterval(() => {
116             getADCState();
117         }, 2000);
118     }
119
120     document.addEventListener("DOMContentLoaded", startAutoFetch);
121
122     </script>
123 </body>
124 </html>
125 )rawliteral";
126
127 void handleRoot() {
128     server.send_P(200, "text/html", MAIN_page);
129 }
130
131 void handleAdc() {
132     int adc = analogRead(A0);
133     server.send(200, "text/plain", String(adc));
134 }
135
136 void handleLED() {
137     if (server.hasArg("state")) {
138         String state = server.arg("state");
139         if (state == "on") {
140             digitalWrite(ledPin, LOW);
141             ledState = true;
142         } else if (state == "off") {
143             digitalWrite(ledPin, HIGH);
144             ledState = false;
145         }
146         server.send(200, "text/plain", "LED " + state);

```

```

147     } else {
148         server.send(400, "text/plain", "Hiányzó paraméter");
149     }
150 }
151
152 void handleCalculate() {
153     if (server.hasArg("num1") && server.hasArg("num2")) {
154         int num1 = server.arg("num1").toInt();
155         int num2 = server.arg("num2").toInt();
156         int sumResValue = num1 + num2;
157         int mulResValue = num1 * num2;
158         char jsonResponse[100];
159         sprintf(jsonResponse, "{\"sumRes\": %d, \"mulRes\": %d}", sumResValue, mulResValue
160 );
161         //String jsonResponse = "{\"param1\": \"\" + ()param1 + "\", \"param2\": \"\" + param2 +
162         //\"\"}";
163         server.send(200, "application/json", jsonResponse);
164     } else {
165         server.send(400, "application/json", "{\"error\": \"Missing parameters\"}");
166     }
167 }
168
169 void setup() {
170     pinMode(ledPin, OUTPUT);
171     digitalWrite(ledPin, LOW);
172
173     Serial.begin(115200);
174     WiFi.begin(ssid, password);
175     Serial.println("Csatlakozás Wi-Fi hálózathoz...");
176
177     while (WiFi.status() != WL_CONNECTED) {
178         delay(500);
179         Serial.print(".");
180     }
181
182     Serial.println("Csatlakozva!");
183     Serial.print("IP cím: ");
184     Serial.println(WiFi.localIP());
185
186     server.on("/", handleRoot);
187     server.on("/led", handleLED);
188     server.on("/adc", handleAdc);
189     server.on("/calculate", handleCalculate);
190
191     server.begin();
192     Serial.println("Webszerver elindult.");
193 }
194
195 void loop() {
196     server.handleClient();
197 }

```