

## Időpontok funkció működés

### PHP forrásfájlok:

#### Időpontok létrehozása

```
<?php
// Csatlakozom az adatbázishoz.
require_once "kapcsolat.php";

// Beállítom, hogy a válasz JSON formátumú legyen.
header('Content-Type: application/json');

// Csak POST metódus engedélyezett, más esetben 405 hibát küldök vissza.
if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    http_response_code(405);
    echo json_encode(['error' => 'Csak POST metódussal érhető el ez a végpont.']);
    exit;
}

// Beolvasom a JSON formátumú bemenetet és tömbbé alakítom.
$input = json_decode(file_get_contents('php://input'), true);

// Ellenőrzöm, hogy minden szükséges adat megérkezett-e.
if (
    !is_array($input) ||
    !isset($input['szolgaltatok_id'], $input['datum'], $input['ido'])
) {
    http_response_code(400);
    echo json_encode(['error' => 'Hiányzó mezők (szolgaltatok_id, datum, ido).']);
    exit;
}

// Az értékeket előkészítem a beszúráshoz.
$szId = (int)$input['szolgaltatok_id'];
$datum = trim($input['datum']);
$ido = trim($input['ido']);
$fogl = isset($input['foglalhato']) ? (int)$input['foglalhato'] : 1; // ha nincs megadva, 1-et állítok be

try {
    // Elkészítem a SQL utasítást az új időpont beszúráshoz.
    $stmt = $pdo->prepare("
        INSERT INTO idopontok (szolgaltatok_id, datum, ido, foglalhato)
        VALUES (:szid, :datum, :ido, :foglalhato)
    ");

    // Lefuttatom a lekérdezést a megadott adatokkal.
```

```
$stmt->execute([
    ':szid' => $szId,
    ':datum' => $datum,
    ':ido' => $ido,
    ':foglalhato' => $fogl
]);

// Sikeres beszúrás esetén visszaküldöm az új rekord azonosítóját.
http_response_code(201);
echo json_encode([
    'status' => 'Időpont sikeresen létrehozva',
    'idopont_id' => (int)$pdo->lastInsertId()
]);
} catch (PDOException $e) {
    // Ha hiba történik, visszajelzést adok róla JSON formátumban.
    http_response_code(500);
    echo json_encode(['error' => 'Adatbázis hiba: ' . $e->getMessage()]);
}
```

**Időpontok módosítása**

```
<?php
// Betöltöm az adatbázis kapcsolatot.
require_once "kapcsolat.php";
header('Content-Type: application/json');

// Csak PUT metódust engedek, egyéb esetben hibát dobok.
if ($_SERVER['REQUEST_METHOD'] !== 'PUT') {
    http_response_code(405);
    echo json_encode(['error' => 'Csak PUT metódussal érhető el ez a végpont.']);
    exit;
}

// Ellenőrzöm, hogy kapok-e érvényes id-t GET paraméterként.
if (empty($_GET['id']) || !ctype_digit($_GET['id'])) {
    http_response_code(400);
    echo json_encode(['error' => 'Hiányzó vagy érvénytelen id.']);
    exit;
}
$id = (int)$_GET['id']; // Az ID számot egész számként tárolom.

// Beolvasom a JSON-bemenetet, és tömbbé alakítom.
$input = json_decode(file_get_contents('php://input'), true);
if (!is_array($input)) {
    http_response_code(400);
    echo json_encode(['error' => 'Nem sikerült értelmezni a JSON bemenetet.']);
    exit;
}
```

```
// Ellenőrzöm, hogy az adott ID-val létezik-e időpont az adatbázisban.
try {
    $stmtChk = $pdo->prepare('SELECT COUNT(*) FROM idopontok WHERE id = :id');
    $stmtChk->execute([':id' => $id]);
    if ($stmtChk->fetchColumn() == 0) {
        http_response_code(404);
        echo json_encode(['error' => 'Nincs ilyen id-jú időpont.']);
        exit;
    }
} catch (PDOException $e) {
    http_response_code(500);
    echo json_encode(['error' => 'Adatbázis hiba (ellenőrzés): ' . $e-
    >getMessage()]);
    exit;
}

// Felépítem a frissítendő mezők listáját és az értékeket.
$fields = [];
$params = [':id' => $id];

if (isset($input['szolgaltatok_id'])) {
    $fields[] = 'szolgaltatok_id = :szid';
    $params[':szid'] = (int)$input['szolgaltatok_id'];
}
if (isset($input['datum'])) {
    $fields[] = 'datum = :datum';
    $params[':datum'] = trim($input['datum']);
}
if (isset($input['ido'])) {
    $fields[] = 'ido = :ido';
    $params[':ido'] = trim($input['ido']);
}
if (isset($input['foglalhato'])) {
    $fields[] = 'foglalhato = :fogl';
    $params[':fogl'] = (int)$input['foglalhato'];
}

// Ha egyetlen mező sincs megadva, nem végzek módosítást.
if (empty($fields)) {
    http_response_code(400);
    echo json_encode(['error' => 'Nincs módosítandó mező.']);
    exit;
}

// Összeállítom a dinamikus SQL lekérdezést.
$sql = 'UPDATE idopontok SET ' . implode(', ', $fields) . ' WHERE id = :id';
```

```
try {
    // Lefuttatom a frissítést az előkészített adatokkal.
    $stmt = $pdo->prepare($sql);
    $stmt->execute($params);
    echo json_encode(['status' => 'Időpont sikeresen módosítva']);
} catch (PDOException $e) {
    // Hiba esetén részletes hibaüzenetet küldök vissza.
    http_response_code(500);
    echo json_encode(['error' => 'Adatbázis hiba (update idopont): ' . $e-
>getMessage()]);
    exit;
}
```

### **Időpontok törlése**

```
<?php
// Csatlakozom az adatbázishoz a külön fájlban tárolt kapcsolat segítségével.
require_once "kapcsolat.php";

// Beállítom a válasz formátumát JSON-ra.
header('Content-Type: application/json');

// Ellenőrzöm, hogy a kérés metódusa valóban DELETE-e.
if ($_SERVER['REQUEST_METHOD'] !== 'DELETE') {
    http_response_code(405); // Nem engedélyezett metódus
    echo json_encode(['error' => 'Csak DELETE metódussal érhető el ez a
végpont.']);
    exit;
}

// Ellenőrzöm, hogy kaptam-e érvényes numerikus id paramétert.
if (empty($_GET['id']) || !ctype_digit($_GET['id'])) {
    http_response_code(400); // Rossz kérés
    echo json_encode(['error' => 'Hiányzó vagy érvénytelen id.']);
    exit;
}

$id = (int)$_GET['id']; // Az ID-t egész számmá alakítom.

try {
    // Lekérdezem, hogy létezik-e az adott ID-jú időpont.
    $stmtChk = $pdo->prepare('SELECT COUNT(*) FROM idopontok WHERE id = :id');
    $stmtChk->execute([':id' => $id]);
    if ($stmtChk->fetchColumn() == 0) {
        http_response_code(404); // Nem található.
        echo json_encode(['error' => 'Nincs ilyen id-jú időpont.']);
        exit;
    }
}
```

```
} catch (PDOException $e) {
    // Hiba történt az ellenőrzés közben.
    http_response_code(500);
    echo json_encode(['error' => 'Adatbázis hiba (ellenőrzés): ' . $e-
>getMessage()]);
    exit;
}

try {
    // Törölöm az időpontot az adatbázisból.
    $stmt = $pdo->prepare('DELETE FROM idopontok WHERE id = :id');
    $stmt->execute([':id' => $id]);

    // Visszajelzést adok, ha sikeres volt a törlés.
    echo json_encode(['status' => 'Időpont sikeresen törölve']);
} catch (PDOException $e) {
    // Hiba történt a törlés során.
    http_response_code(500);
    echo json_encode(['error' => 'Adatbázis hiba (delete idopont): ' . $e-
>getMessage()]);
    exit;
}
```

## HTML forrásfájlok:

### Időpontok létrehozása

```
<!DOCTYPE html>
<html lang="hu">
<head>
    <!-- Dokumentum típusa és nyelv beállítása -->
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <!-- Menü stíluslap betöltése -->
    <link rel="stylesheet" href="menu.css" />
    <!-- Oldal címe -->
    <title>Új időpont hozzáadása</title>
    <style>
        /* Alapértelmezett test és háttér */
        body {
            font-family: Arial, sans-serif;
            background-color: #f5f5f5;
        }
        /* Űrlap konténer */
        .container {
            max-width: 500px;
            margin: 0 auto;
            background-color: white;
```

```
padding: 30px;
border-radius: 8px;
box-shadow: 0 2px 8px rgba(0,0,0,0.1);
}
/* Oldalcím */
h2 {
text-align: center;
color: black;
}
/* Űrlap címkék */
label {
display: block;
margin-top: 15px;
font-weight: bold;
}
/* Beviteli mezők és legördülő listák */
input,
select {
width: 100%;
padding: 10px;
margin-top: 5px;
border: 1px solid #ccc;
border-radius: 4px;
}
/* Küldés gomb */
button {
margin-top: 20px;
width: 100%;
padding: 12px;
background-color: #28a745;
color: white;
border: none;
border-radius: 5px;
cursor: pointer;
}
button:hover {
background-color: #218838;
}
/* Középre igazítás */
.center {
text-align: center;
color: black;
}
/* Figyelmeztető szöveg */
.color {
color: red;
}
/* Rejtett elem */
```

```
.hidden {
  display: none !important;
}
/* Kisebb képernyő esetén */
@media (max-width: 768px) {
  .container {
    width: 90%;
    padding: 20px;
    margin-top: 20px;
  }
  h2 {
    font-size: 24px;
  }
  label,
  input,
  select,
  button {
    font-size: 15px;
  }
  button {
    padding: 10px;
  }
}
/* Mobil nézet */
@media (max-width: 480px) {
  h2 {
    font-size: 20px;
  }
  label,
  input,
  select,
  button {
    font-size: 14px;
  }
  button {
    padding: 10px;
  }
}
</style>
</head>
<body>
  <!-- Fejléc és navigáció -->
  <header>
    <a href="index.html" class="logo">Időpontfoglaló <span id="felhasznalo-
nev"></span></a>
    <nav>
      <a href="szolgaltatok.html">Szolgáltatók</a>
      <a href="idopontok.html">Időpontok</a>
```

```
<a href="ertekelesek.html">Értékelések</a>
<a href="bejelentkezes.html" id="nav-login"><b>Bejelentkezés</b></a>
<a href="fiokom.html" id="nav-account" class="hidden">Fiókom</a>
<a href="#" id="nav-logout" onclick="kijelentkezes()" class="hidden"><span
class="color">Kijelentkezés</span></a>
</nav>
</header>

<!-- Oldalcím a form felett -->
<h2 class="center">Időpont létrehozása</h2>

<!-- Űrlap konténer -->
<div class="container">
  <form id="letrehozas-form">
    <!-- Szolgáltató kiválasztása -->
    <label for="szid">Szolgáltató:</label>
    <select id="szid" required>
      <option value="">Betöltés...</option>
    </select>

    <!-- Dátum kiválasztása -->
    <label for="datum">Dátum:</label>
    <input type="date" id="datum" required />

    <!-- Idő kiválasztása -->
    <label for="ido">Idő:</label>
    <input type="time" id="ido" required />

    <!-- Foglalhatóság jelölése -->
    <label for="foglalhato">Foglalható:</label>
    <select id="foglalhato">
      <option value="1">Igen</option>
      <option value="0">Nem</option>
    </select>

    <!-- Létrehozás gomb -->
    <button type="submit">Létrehozás</button>
  </form>
</div>

<script>
  // Ellenőrzöm, hogy admin-e a felhasználó
  const felhasznalo = JSON.parse(sessionStorage.getItem('felhasznalo'));
  if (!felhasznalo || felhasznalo.szerepkor !== 'admin') {
    alert('Csak admin jogosultsággal hozható létre időpont.');
```



```
// Szolgáltatók betöltése a legördülő listába
fetch('Szolgáltatok_listazasa.php')
  .then(res => res.json())
  .then(data => {
    if (data.status === 'siker') {
      const select = document.getElementById('szid');
      select.innerHTML = '';
      data.data.forEach(s => {
        const option = document.createElement('option');
        option.value = s.id;
        option.textContent = s.nev;
        select.appendChild(option);
      });
    }
  });

// Űrlap elküldése AJAX-szal
document.getElementById('letrehozasa-form').addEventListener('submit',
function(e) {
  e.preventDefault();
  const payload = {
    szolgaltatok_id: parseInt(document.getElementById('szid').value),
    datum: document.getElementById('datum').value,
    ido: document.getElementById('ido').value,
    foglalhato: parseInt(document.getElementById('foglalhato').value)
  };
  fetch('Idopontok_letrehozasa.php', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify(payload)
  })
  .then(res => res.json())
  .then(data => {
    if (data.status) {
      alert('Időpont sikeresen létrehozva.');
```

```
      window.location.href = 'idopontok.html';
    } else {
      alert('Hiba: ' + (data.error || 'Ismeretlen hiba'));
    }
  })
  .catch(err => {
    console.error(err);
    alert('Hiba történt a létrehozás során.');
```

```
  });
});

// Kijelentkezés funkció
function kijelentkezés() {
```

```
        sessionStorage.removeItem('felhasznalo');
        window.location.href = 'index.html';
    }

    // Navigáció frissítése bejelentkezési állapot szerint
    const user = JSON.parse(sessionStorage.getItem('felhasznalo'));
    if (user && user.nev) {
        document.getElementById('felhasznalo-nev').textContent = ` - ${user.nev}`;
        document.getElementById('nav-login').classList.add('hidden');
        document.getElementById('nav-account').classList.remove('hidden');
        document.getElementById('nav-logout').classList.remove('hidden');
    }
</script>
</body>
</html>
```

### Időpontok módosítása

```
<!DOCTYPE html>
<html lang="hu">
<head>
    <!-- Dokumentum típusa és nyelv beállítása -->
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <!-- Menü stíluslap betöltése -->
    <link rel="stylesheet" href="menu.css" />
    <!-- Oldal címe -->
    <title>Időpont módosítása</title>
    <style>
        /* Alapértelmezett test és háttér */
        body {
            font-family: Arial, sans-serif;
            background-color: #f5f5f5;
        }
        /* Űrlap konténer */
        .container {
            max-width: 500px;
            margin: 0 auto;
            background-color: white;
            padding: 30px;
            border-radius: 8px;
            box-shadow: 0 2px 8px rgba(0,0,0,0.1);
        }
        /* Oldalcím */
        h2 {
            text-align: center;
            color: black;
        }
    </style>
</head>
<body>
```

```
/* Űrlap címkék */
label {
    display: block;
    margin-top: 15px;
    font-weight: bold;
}
/* Beviteli mezők és legördülő listák */
input,
select {
    width: 100%;
    padding: 10px;
    margin-top: 5px;
    border: 1px solid #ccc;
    border-radius: 4px;
}
/* Küldés gomb */
button {
    margin-top: 20px;
    width: 100%;
    padding: 12px;
    background-color: #007BFF;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}
button:hover {
    background-color: #0056b3;
}
/* Középre igazítás */
.center {
    text-align: center;
    color: black;
}
/* Figyelmeztető szöveg */
.color {
    color: red;
}
/* Rejtett elem */
.hidden {
    display: none !important;
}
/* Kisebb képernyő esetén */
@media (max-width: 768px) {
    .container {
        width: 90%;
        padding: 20px;
        margin-top: 20px;
    }
}
```

```
    }
    h2 {
        font-size: 24px;
    }
    label,
    input,
    select,
    button {
        font-size: 15px;
    }
    button {
        padding: 10px;
    }
}
/* Mobil nézet */
@media (max-width: 480px) {
    h2 {
        font-size: 20px;
    }
    label,
    input,
    select,
    button {
        font-size: 14px;
    }
    button {
        padding: 10px;
    }
}
</style>
</head>
<body>
    <!-- Fejléc és navigáció -->
    <header>
        <a href="index.html" class="logo">Időpontfoglaló <span id="felhasznalo-
nev"></span></a>
        <nav>
            <a href="szolgaltatok.html">Szolgáltatók</a>
            <a href="idopontok.html">Időpontok</a>
            <a href="ertekelesek.html">Értékelések</a>
            <a href="bejelentkezes.html" id="nav-login"><b>Bejelentkezés</b></a>
            <a href="fiokom.html" id="nav-account" class="hidden">Fiókom</a>
            <a href="#" id="nav-logout" onclick="kijelentkezes()" class="hidden"><span
class="color">Kijelentkezés</span></a>
        </nav>
    </header>

    <!-- Oldalcím a form felett -->
```

```
<h2 class="center">Időpont módosítása</h2>

<!-- Űrlap konténer -->
<div class="container">
  <form id="modositas-form">
    <!-- Szolgáltató kiválasztása -->
    <label for="szid">Szolgáltató:</label>
    <select id="szid" required>
      <option value="">Betöltés...</option>
    </select>

    <!-- Dátum kiválasztása -->
    <label for="datum">Dátum:</label>
    <input type="date" id="datum" required />

    <!-- Idő kiválasztása -->
    <label for="ido">Idő:</label>
    <input type="time" id="ido" required />

    <!-- Foglalhatóság jelölése -->
    <label for="foglalhato">Foglalható:</label>
    <select id="foglalhato">
      <option value="1">Igen</option>
      <option value="0">Nem</option>
    </select>

    <!-- Mentés gomb -->
    <button type="submit">Mentés</button>
  </form>
</div>

<script>
  // Ellenőrzöm, hogy admin-e a felhasználó
  const felhasznalo = JSON.parse(sessionStorage.getItem('felhasznalo'));
  if (!felhasznalo || felhasznalo.szerepkor !== 'admin') {
    alert('Csak admin jogosultsággal módosítható időpont.');
    window.location.href = 'index.html';
  }

  // URL paraméterből ID kinyerése
  const urlParams = new URLSearchParams(window.location.search);
  const id = urlParams.get('id');
  if (!id) {
    alert('Hiányzó időpont azonosító.');
    window.location.href = 'idopontok.html';
  }

  // Szolgáltatók betöltése a legördülő listába
```

```
fetch('Szolgaltatok_listazasa.php')
  .then(res => res.json())
  .then(data => {
    if (data.status === 'siker') {
      const select = document.getElementById('szid');
      select.innerHTML = '';
      data.data.forEach(s => {
        const option = document.createElement('option');
        option.value = s.id;
        option.textContent = s.nev;
        select.appendChild(option);
      });
    }
  });

// Időpont adatok betöltése
fetch('idopontok_listazasa.php')
  .then(res => res.json())
  .then(data => {
    if (data.status === 'siker') {
      const idopont = data.data.find(i => i.id == id);
      if (!idopont) {
        alert('Nincs ilyen ID-jú időpont.');
        window.location.href = 'idopontok.html';
        return;
      }
      document.getElementById('datum').value = idopont.datum;
      document.getElementById('ido').value = idopont.ido;
      document.getElementById('foglalhato').value = idopont.foglalhato;

      // Szolgáltató előválasztása
      const setSzid = () => {
        const select = document.getElementById('szid');
        if (select.options.length > 0) {
          select.value = idopont.szolgaltatok_id;
        } else {
          setTimeout(setSzid, 100);
        }
      };
      setSzid();
    }
  });

// Űrlap elküldése AJAX-szal PUT metódussal
document.getElementById('modositas-form').addEventListener('submit',
function(e) {
  e.preventDefault();
  const payload = {
```

```
        szolgaltatok_id: parseInt(document.getElementById('szid').value),
        datum: document.getElementById('datum').value,
        ido: document.getElementById('ido').value,
        foglalhato: parseInt(document.getElementById('foglalhato').value)
    });
    fetch(`Idopontok_modositasa.php?id=${id}`, {
        method: 'PUT',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify(payload)
    })
    .then(res => res.json())
    .then(data => {
        if (data.status) {
            alert('Időpont sikeresen módosítva.');
```

window.location.href = 'idopontok.html';

```
        } else {
            alert('Hiba: ' + (data.error || 'Ismeretlen hiba'));
        }
    })
    .catch(err => {
        console.error(err);
        alert('Hiba történt a módosítás során.');
```

});

```
});

// Kijelentkezés funkció
function kijelentkezés() {
    sessionStorage.removeItem('felhasznalo');
    window.location.href = 'index.html';
}

// Navigáció frissítése bejelentkezési állapot szerint
const user = JSON.parse(sessionStorage.getItem('felhasznalo'));
if (user && user.nev) {
    document.getElementById('felhasznalo-nev').textContent = ` - ${user.nev}`;
    document.getElementById('nav-login').classList.add('hidden');
    document.getElementById('nav-account').classList.remove('hidden');
    document.getElementById('nav-logout').classList.remove('hidden');
}
</script>
</body>
</html>
```

## Időpontok törlése

```
<!DOCTYPE html>
<html lang="hu">
<head>
  <!-- Dokumentum típusa és nyelv beállítása -->
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <!-- Menü stíluslap betöltése -->
  <link rel="stylesheet" href="menu.css" />
  <!-- Oldal címe -->
  <title>Időpontok törlése</title>
  <style>
    /* Oldal alapstílusai */
    body {
      font-family: Arial, sans-serif;
      background-color: #f5f5f5;
      margin: 0;
      padding: 0;
    }
    /* Űrlap konténer középre igazítása */
    .container {
      max-width: 400px;
      margin: 30px auto;
      background-color: white;
      padding: 30px;
      border-radius: 10px;
      box-shadow: 0 2px 8px rgba(0,0,0,0.1);
      text-align: center;
    }
    /* Beviteli mező stílusa */
    .container input {
      width: 100%;
      padding: 12px;
      margin-bottom: 20px;
      font-size: 16px;
      border: 1px solid #ccc;
      border-radius: 5px;
    }
    /* Gomb stílusa */
    .container button {
      width: 100%;
      padding: 12px;
      font-size: 16px;
      color: white;
      background-color: #007bff;
      border: none;
      border-radius: 5px;
      cursor: pointer;
    }
  </style>
</head>
<body>
  <div class="container">
    <input type="text" value="Időpontot törölj" />
    <button type="button" value="Töröl" />
  </div>
</body>
</html>
```



```
}
.container button:hover {
  background-color: #0056b3;
}
/* Oldalcím */
h2 {
  text-align: center;
  color: black;
}

.color {
  color: red;
}
/* Rejtett elem */
.hidden {
  display: none !important;
}
/* Mobil nézet 768px alatt */
@media (max-width: 768px) {
  .container {
    width: 90%;
    padding: 20px;
  }
  h2 {
    font-size: 24px;
  }
  input,
  button {
    font-size: 15px;
  }
  button {
    padding: 10px;
  }
}
/* Mobil nézet 480px alatt */
@media (max-width: 480px) {
  h2 {
    font-size: 20px;
  }
  input,
  button {
    font-size: 14px;
  }
  button {
    padding: 9px;
  }
}
</style>
```

```

</head>
<body>
  <!-- Fejléc és navigáció -->
  <header>
    <a href="index.html" class="logo">Időpontfoglaló <span id="felhasznalo-
nev"></span></a>
    <nav>
      <a href="szolgaltatok.html">Szolgáltatók</a>
      <a href="idopontok.html">Időpontok</a>
      <a href="ertekelesek.html">Értékelések</a>
      <!-- Bejelentkezés/Fiók/Kijelentkezés linkek kezelése -->
      <a href="bejelentkezes.html" id="nav-login"><b>Bejelentkezés</b></a>
      <a href="fiokom.html" id="nav-account" class="hidden">Fiókom</a>
      <a href="#" id="nav-logout" onclick="kijelentkezes()" class="hidden"><span
class="color">Kijelentkezés</span></a>
    </nav>
  </header>

  <!-- Oldalcím -->
  <h2>Időpont törlése</h2>

  <!-- Törlő űrlap -->
  <div class="container">
    <input type="number" id="torles-id" placeholder="Időpont ID" required />
    <button id="btn-delete">Törlés</button>
  </div>

  <script>
    // Törlés gomb eseménykezelője
    document.getElementById('btn-delete').addEventListener('click', () => {
      const id = document.getElementById('torles-id').value.trim();
      if (!id) {
        alert('Kérem, adj a törlendő ID-t.');
```

```
        alert('Hiba történt a törlés során.');
```

```
    });
```

```
});
```

```
  
// Kijelentkezés funkció
```

```
function kijelentkezés() {
```

```
    sessionStorage.removeItem('felhasznalo');
```

```
    window.location.href = 'index.html';
```

```
}
```

```
  
// Navigáció frissítése bejelentkezési állapot szerint
```

```
const user = JSON.parse(sessionStorage.getItem('felhasznalo'));
```

```
if (user && user.nev) {
```

```
    document.getElementById('felhasznalo-nev').textContent = ` - ${user.nev}`;
```

```
    document.getElementById('nav-login').classList.add('hidden');
```

```
    document.getElementById('nav-account').classList.remove('hidden');
```

```
    document.getElementById('nav-logout').classList.remove('hidden');
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

## HTML kimenetek:

← → ↻ ⓘ localhost/idopontfoglalo/Idopontok\_letrehozasa.html ☆ 📄 ⬇ 🌐 ⋮

**Időpontfoglaló** - Kiss Dalma

Szolgáltatók Időpontok Értékelések Fiókom **Kijelentkezés**

### Időpont létrehozása

**Szolgáltató:**

SmileDent ▼

**Dátum:**

éééé. hh. nn. 📅

**Idő:**

--:-- ⌚

**Foglalható:**

Igen ▼

Létrehozás

## Szakmai Gyakorlat I. – Frontend – Backend integrációs szakasz

Orosz Kristóf – EYZWG9

← → ↺ ⓘ localhost/idopontfoglalo/ldopontok\_modositasa.html?id=133 ☆ 📄 ⬇️ 🌐 ⋮

**Időpontfoglaló** - Kiss Dalma Szolgáltatók Időpontok Értékelések Fiókom **Kijelentkezés**

### Időpont módosítása

**Szolgáltató:**

Cardio4U Fitness ▾

**Dátum:**

2025. 06. 19. 📅

**Idő:**

20 : 14 ⌚

**Foglalható:**

Igen ▾

Mentés

← → ↺ ⓘ localhost/idopontfoglalo/ldopontok\_torlese.html?id=133 ☆ 📄 ⬇️ 🌐 ⋮

**Időpontfoglaló** - Kiss Dalma Szolgáltatók Időpontok Értékelések Fiókom **Kijelentkezés**

### Időpont törlése

Időpontok ID

Törles