**Dokumentáció a Web-programozás 1 – Beadandó feladathoz**

## Hogyan készült az alkalmazás?

A projektet kétfős csoportban készítettük el teljesen a nulláról. A csapat tagjai **Balogh Róbert (SSDPW9)** és **Petrik Tibor (GDAZA8)**. Online konzultációval szervezetten készítettük el a projektet. A célunk az volt, hogy egy gyakorlati példán keresztül bemutassuk az AJAX alapú adatkezelést, REST API kommunikációt és a JavaScript dinamikus lehetőségeit.

### **A munka menete:**

1. **Tervezés:**
   * Meghatároztuk, milyen adatokkal szeretnénk dolgozni (pl. név, magasság, súly).
   * Megterveztük az oldal HTML szerkezetét (form, gombok, táblázat).
   * Felosztottuk egymás között a feladatokat (HTML, JS, CSS, backend).
2. **Fejlesztés:**
   * Először elkészítettük a statikus HTML oldalt (index.html), majd fokozatosan beillesztettük az űrlapokat és a vezérlőelemeket.
   * A JavaScript fájlban megírtuk a CRUD műveletekhez szükséges AJAX (fetch) hívásokat.
   * Minden funkció külön eseménykezelőt kapott (létrehozás, olvasás, módosítás, törlés).
   * Beállítottuk a hibakezelést, validációt és a felhasználói visszajelzéseket.
   * A megjelenítéshez alap CSS-t használtunk, szem előtt tartva az olvashatóságot.
   * A projektet a GitHub-on készítettük el, ami a következő: <https://github.com/PetrikTibi/Webprog_beadando>
3. **Tesztelés:**
   * A kész funkciókat lokálisan majd a weboldalon próbáltuk ki
   * Ellenőriztük az adatkezelés sikerességét, valamint a szerver válaszait (JSON formátumban).
4. **Feltöltés:**
   * Az elkészült projektet feltöltöttük a <http://ssdgda.nhely.hu/> tárhelyre melynek az FTP belépési adatai a következők:

Felhasználónév: tibi2025

Jelszó: Jelszo2025!

FTP kiszolgáló: [ftp.nethely.hu](ftp://ftp.nethely.hu)

* + A fájlokat rendszereztük, és biztosítottuk, hogy minden link és script működjön.

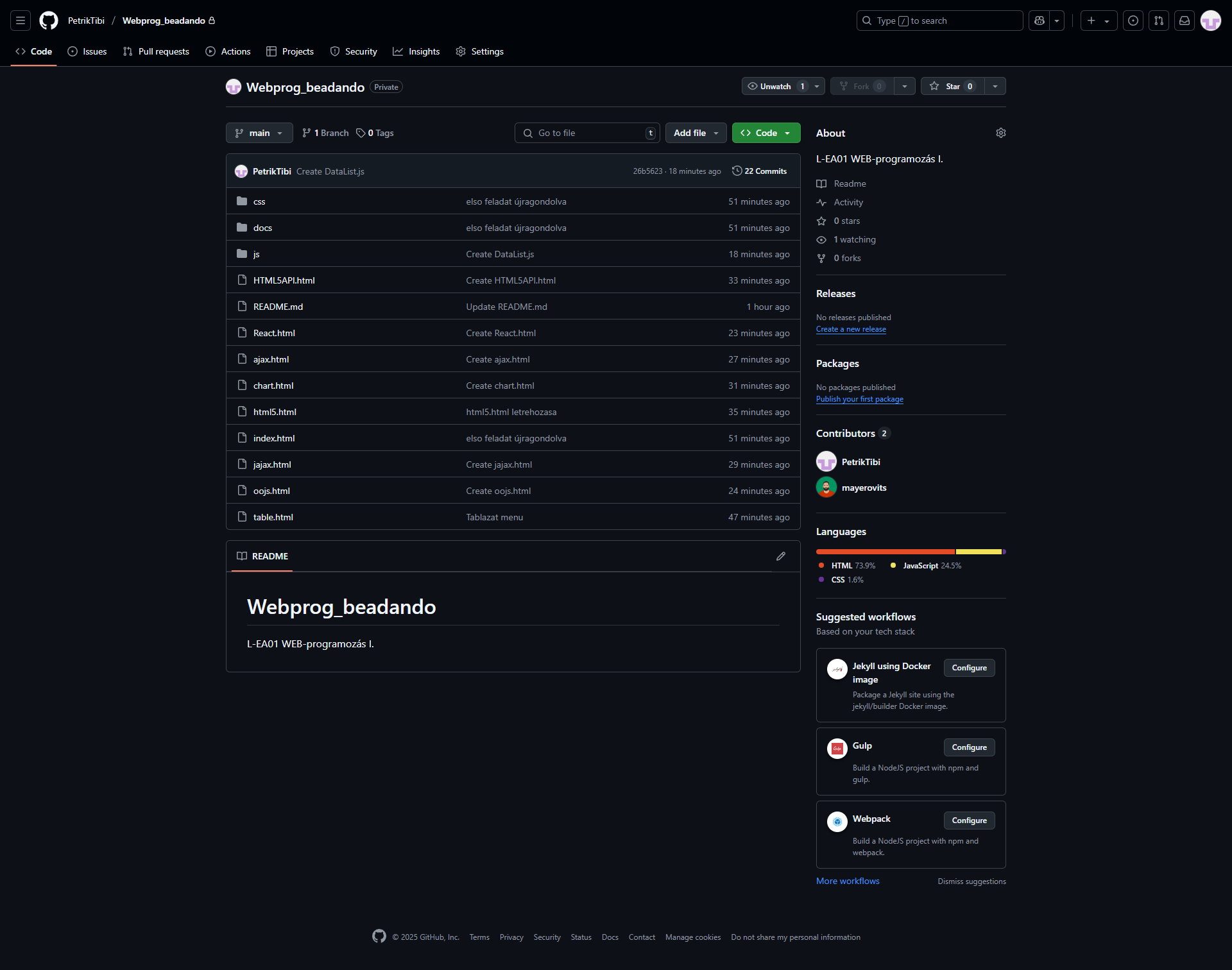
1. **Dokumentáció:**
   * Oldalanként bemutattuk a rendszer működését, kódokat és képernyőfotókat mellékelve.
   * Kiemeltük a legfontosabb technikai megvalósításokat és fejlesztési lépéseket.

Ez a projekt nemcsak a tanultak elmélyítését segítette, hanem kiváló tapasztalat volt a csapatmunka és a frontend-fejlesztés terén is.

## A fejlesztés eredménye

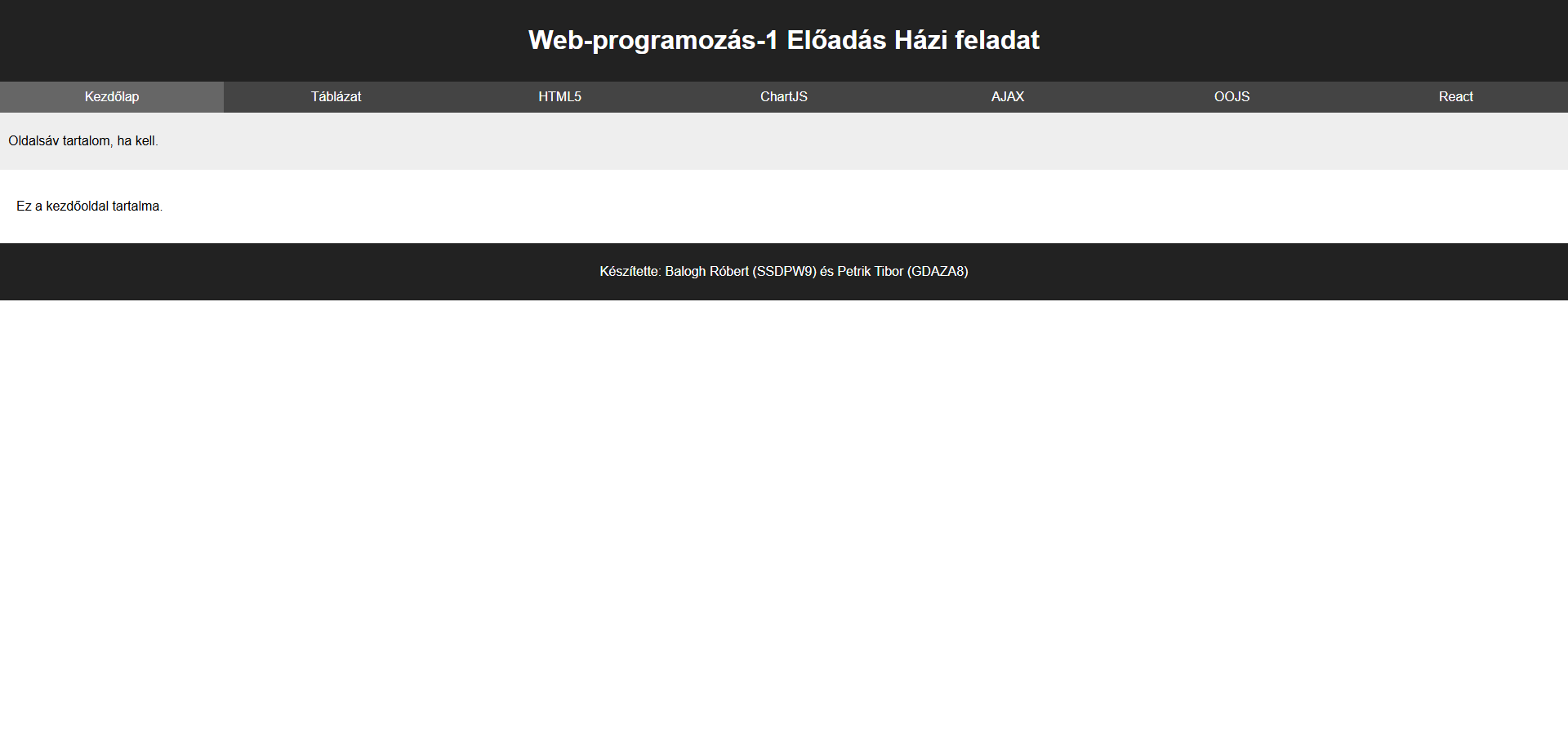
Az általunk készített munkát a következő GitHub repositoryba helyeztük el, a felosztott feladatoknak megfdelelően, lekövethetően, hogy ki melyik alegységét készítette a projectnek. A repo elérhetősége a következő: <https://github.com/PetrikTibi/Webprog_beadando>

A csapat résztvevői Balogh Róbert (<https://github.com/mayerovits>) és Petrik Tibor (<https://github.com/PetrikTibi/> ):



Az elkészült projekt tesztelés után letöltöttük majd feltöltöttük az ingyenesen létrehozott Nethely szolgáltató által biztosított domainre a hozzá kapcsolt webtárhellyel együtt. A projekt elérhetősége a következő:

<http://ssdgda.nhely.hu/index.html>



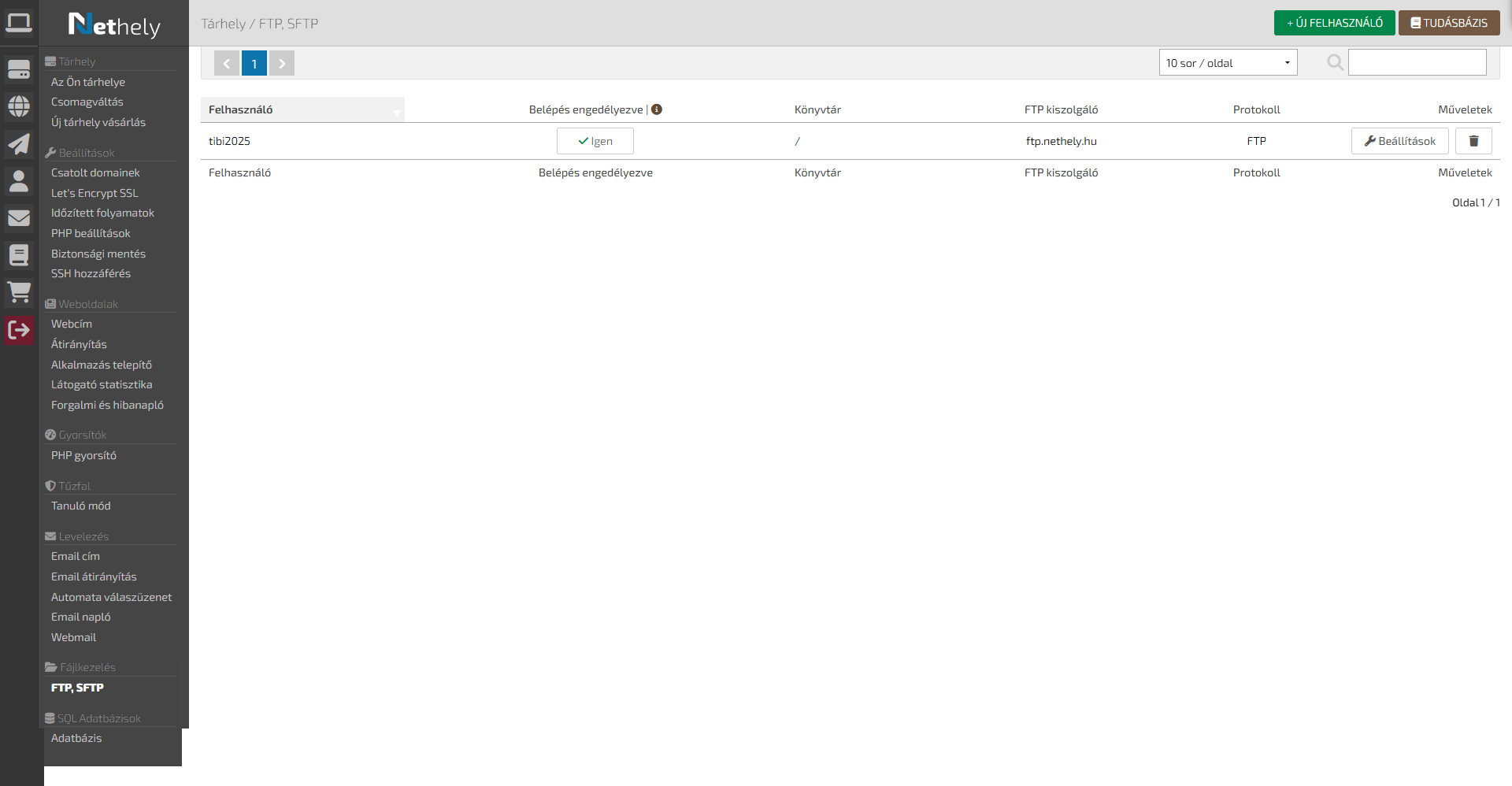
**Az ellenőrzéshez szükséges belépési adatok a következők:**

**FTP**

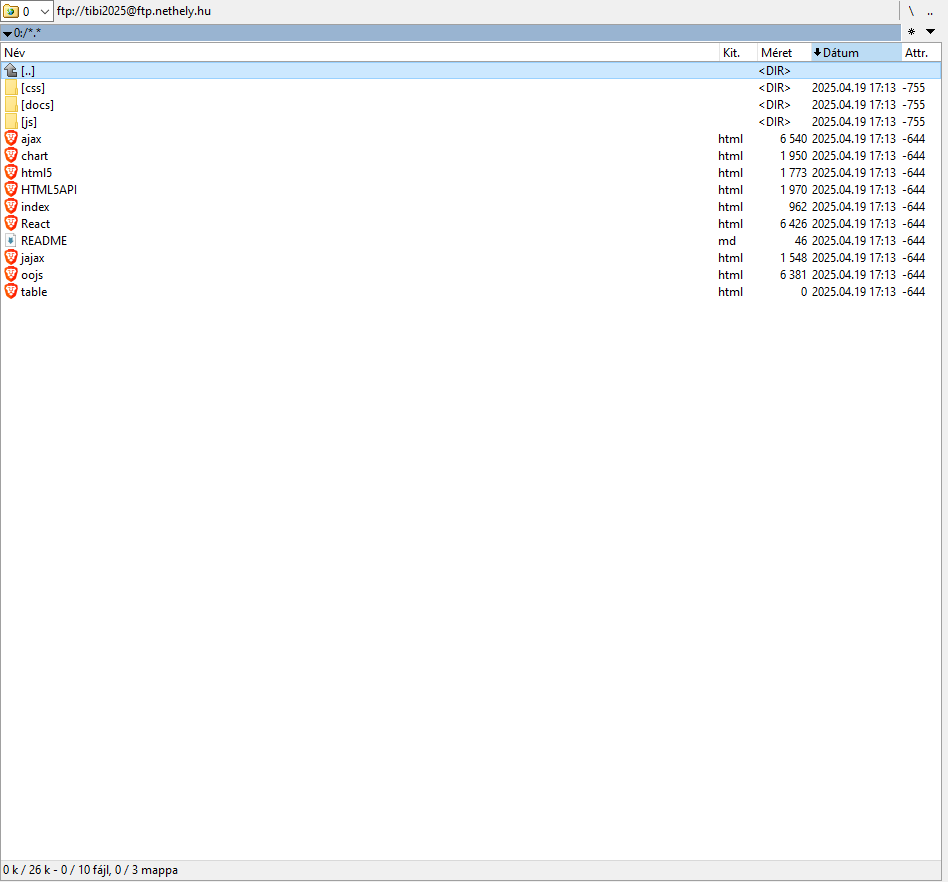
Felhasználónév: tibi2025

Jelszó: Jelszo2025!

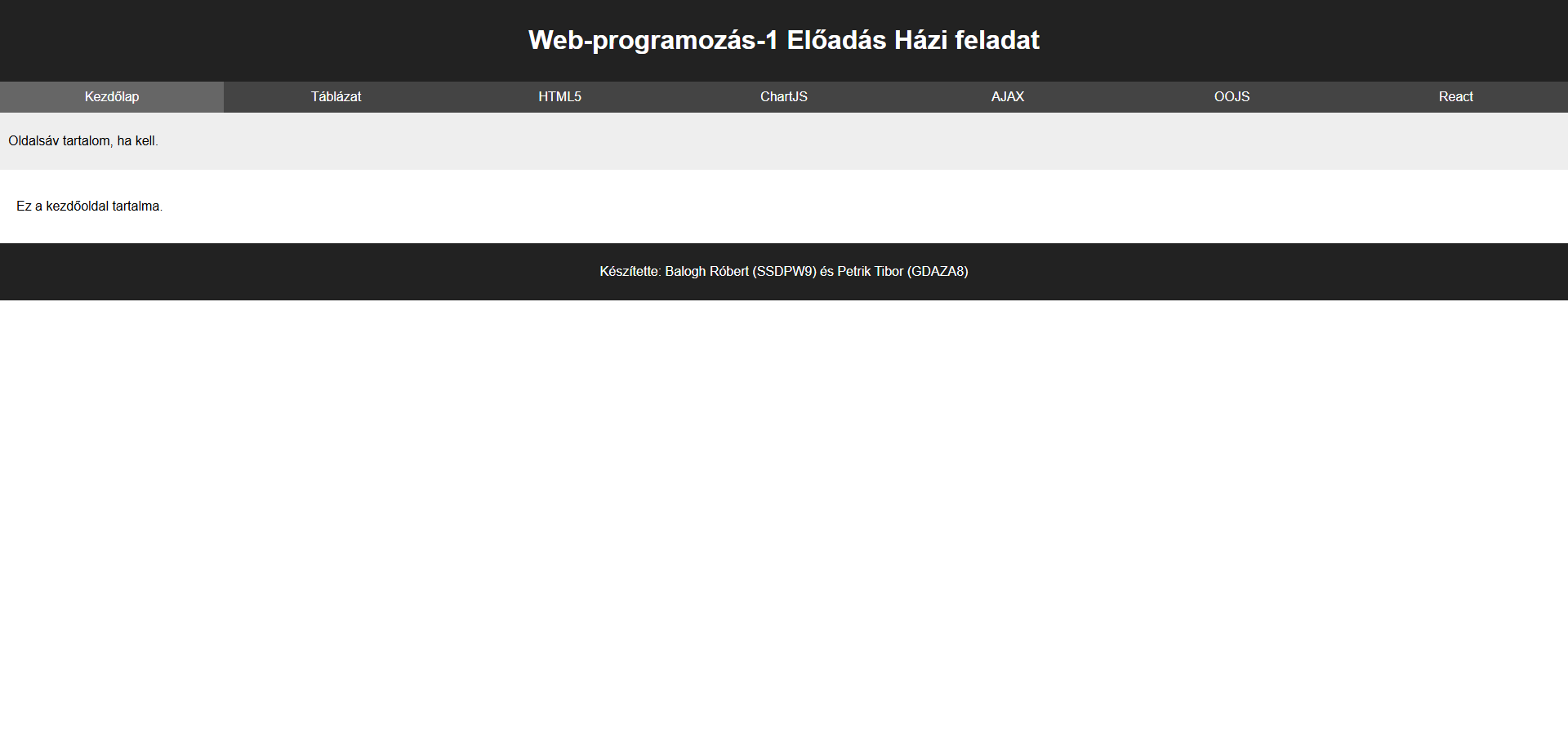
FTP kiszolgáló: [ftp.nethely.hu](ftp://ftp.nethely.hu)



**A jelenlegi tartalma:**



## Kezdőlap



**Forráskódja:**

<!DOCTYPE html>

<html lang="hu">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <title>Web-programozás-1 Házi feladat</title>

  <link rel="stylesheet" href="css/style.css" />

</head>

<body>

  <header>

    <h1>Web-programozás-1 Előadás Házi feladat</h1>

  </header>

  <nav>

    <ul>

      <li><a href="index.html" class="active">Kezdőlap</a></li>

      <li><a href="table.html">Táblázat</a></li>

      <li><a href="html5.html">HTML5</a></li>

      <li><a href="chart.html">ChartJS</a></li>

      <li><a href="ajax.html">AJAX</a></li>

      <li><a href="oojs.html">OOJS</a></li>

      <li><a href="react.html">React</a></li>

    </ul>

  </nav>

  <aside>

    <p>Oldalsáv tartalom, ha kell.</p>

  </aside>

  <div id="content">

    <p>Ez a kezdőoldal tartalma.</p>

  </div>

  <footer>

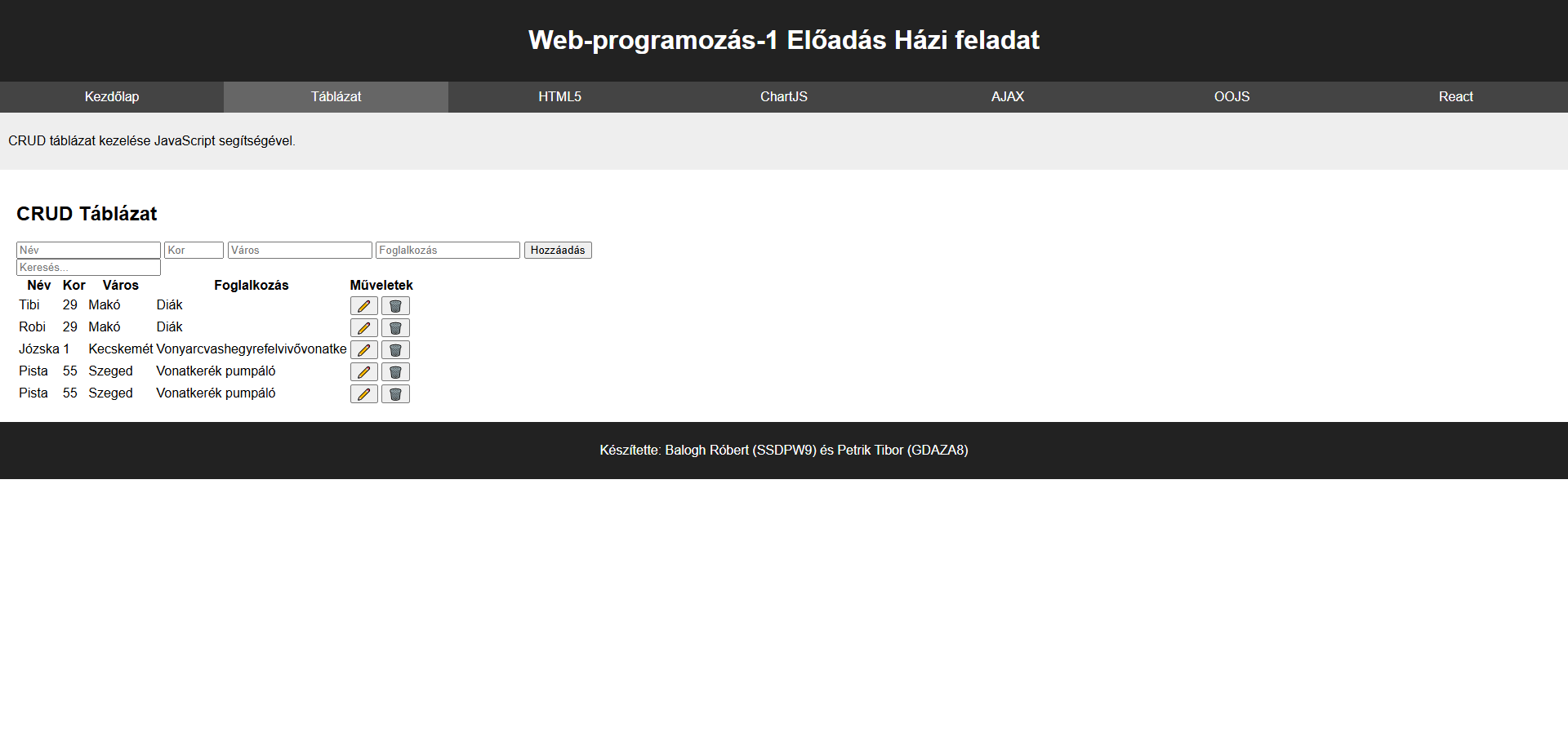
    <p>Készítette: Balogh Róbert (SSDPW9) és Petrik Tibor (GDAZA8)</p>

  </footer>

</body>

</html>

## HTML5



**A forráskódja**

<!DOCTYPE html>

<html lang="hu">

<head>

  <meta charset="UTF-8">

  <title>Táblázat - Webprog1</title>

  <link rel="stylesheet" href="css/style.css">

  <script src="js/table.js" defer></script>

</head>

<body>

  <header>

    <h1>Web-programozás-1 Előadás Házi feladat</h1>

  </header>

  <nav>

    <ul>

      <li><a href="index.html">Kezdőlap</a></li>

      <li><a href="table.html" class="active">Táblázat</a></li>

      <li><a href="html5.html">HTML5</a></li>

      <li><a href="chart.html">ChartJS</a></li>

      <li><a href="ajax.html">AJAX</a></li>

      <li><a href="oojs.html">OOJS</a></li>

      <li><a href="react.html">React</a></li>

    </ul>

  </nav>

  <aside>

    <p>CRUD táblázat kezelése JavaScript segítségével.</p>

  </aside>

  <div id="content">

    <h2>CRUD Táblázat</h2>

    <form id="dataForm">

      <input type="text" id="name" placeholder="Név" required minlength="3" maxlength="20">

      <input type="number" id="age" placeholder="Kor" required min="1" max="120">

      <input type="text" id="city" placeholder="Város" required minlength="2" maxlength="30">

      <input type="text" id="job" placeholder="Foglalkozás" required minlength="2" maxlength="30">

      <button type="submit">Hozzáadás</button>

    </form>

    <input type="text" id="searchInput" placeholder="Keresés...">

    <table id="dataTable">

      <thead>

        <tr>

          <th data-col="name">Név</th>

          <th data-col="age">Kor</th>

          <th data-col="city">Város</th>

          <th data-col="job">Foglalkozás</th>

          <th>Műveletek</th>

        </tr>

      </thead>

      <tbody></tbody>

    </table>

  </div>

  <footer>

    <p>Készítette: Balogh Róbert (SSDPW9) és Petrik Tibor (GDAZA8)</p>

  </footer>

</body>

</html>

**table.js forráskódja**

let records = [];

document.addEventListener("DOMContentLoaded", () => {

  const form = document.getElementById("dataForm");

  const searchInput = document.getElementById("searchInput");

  form.addEventListener("submit", addRecord);

  searchInput.addEventListener("input", renderTable);

  document.querySelectorAll("th[data-col]").forEach(th => {

    th.addEventListener("click", () => sortTable(th.dataset.col));

  });

  renderTable();

});

function addRecord(e) {

  e.preventDefault();

  const name = document.getElementById("name").value.trim();

  const age = parseInt(document.getElementById("age").value.trim());

  const city = document.getElementById("city").value.trim();

  const job = document.getElementById("job").value.trim();

  if (!name || !age || !city || !job) return alert("Minden mező kötelező!");

  records.push({ name, age, city, job });

  e.target.reset();

  renderTable();

}

function deleteRecord(index) {

  if (confirm("Biztosan törlöd ezt a sort?")) {

    records.splice(index, 1);

    renderTable();

  }

}

function editRecord(index) {

  const rec = records[index];

  document.getElementById("name").value = rec.name;

  document.getElementById("age").value = rec.age;

  document.getElementById("city").value = rec.city;

  document.getElementById("job").value = rec.job;

  records.splice(index, 1); // töröljük, újként újra hozzáadjuk

  renderTable();

}

let currentSort = { col: "", asc: true };

function sortTable(col) {

  if (currentSort.col === col) {

    currentSort.asc = !currentSort.asc;

  } else {

    currentSort = { col, asc: true };

  }

  records.sort((a, b) => {

    if (a[col] < b[col]) return currentSort.asc ? -1 : 1;

    if (a[col] > b[col]) return currentSort.asc ? 1 : -1;

    return 0;

  });

  renderTable();

}

function renderTable() {

  const tbody = document.querySelector("#dataTable tbody");

  const filter = document.getElementById("searchInput").value.toLowerCase();

  tbody.innerHTML = "";

  records

    .filter(r => Object.values(r).some(val => val.toString().toLowerCase().includes(filter)))

    .forEach((rec, index) => {

      const row = document.createElement("tr");

      row.innerHTML = `

        <td>${rec.name}</td>

        <td>${rec.age}</td>

        <td>${rec.city}</td>

        <td>${rec.job}</td>

        <td>

          <button onclick="editRecord(${index})">✏️</button>

          <button onclick="deleteRecord(${index})">🗑️</button>

        </td>

      `;

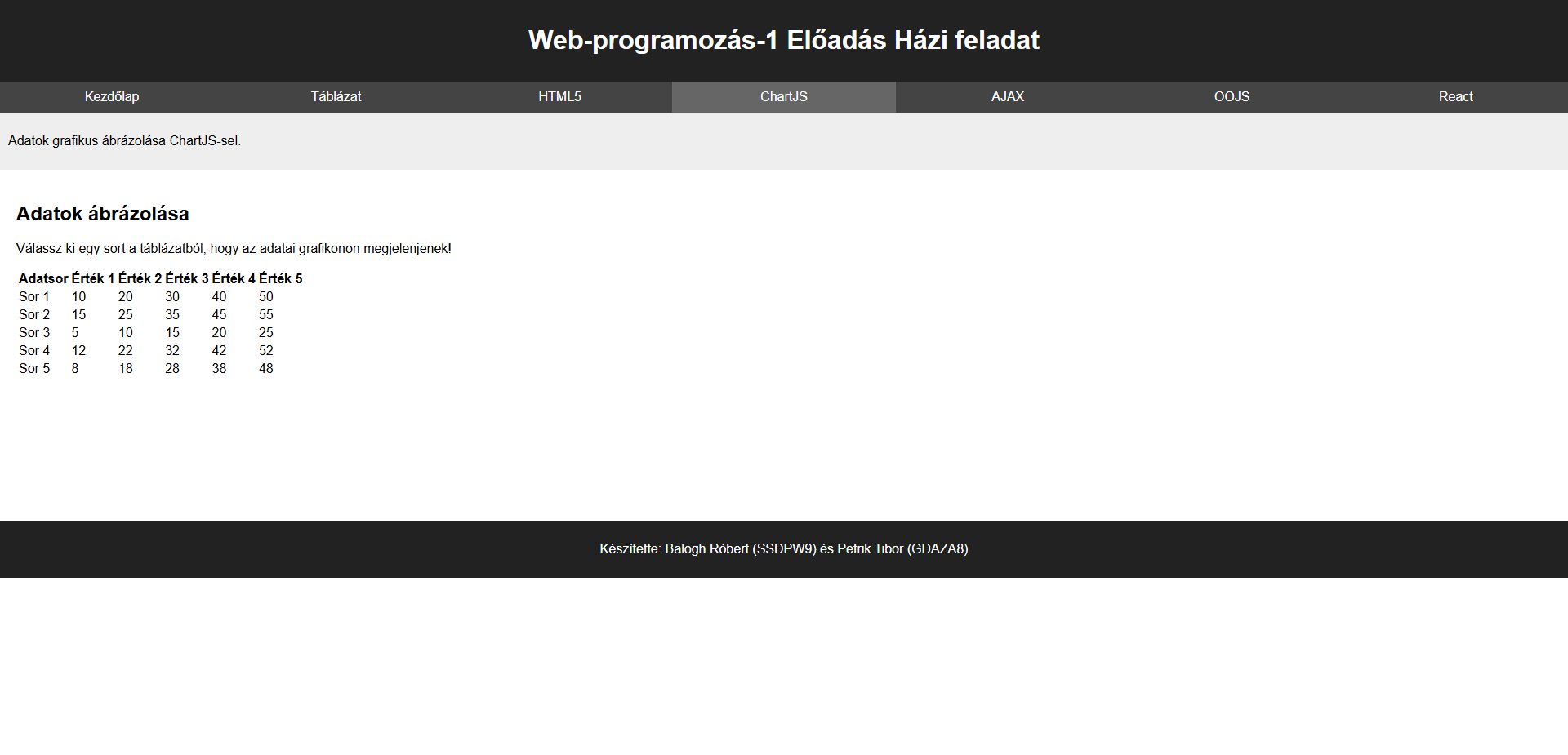
      tbody.appendChild(row);

    });

}

Az adatokat fel tudjuk vinni, lehet szűrni, törölni és módosítani. Az életkor 1 és 120 között lehet. Városszűrést nem állítottunk be, de tovább lehetne tervezni ország és városkorlátozással, ha előzetesen betáplálnánk az országokat a városainak listájával.

## Chart.html



A forráskódja:

<!DOCTYPE html>

<html lang="hu">

<head>

  <meta charset="UTF-8">

  <title>ChartJS - Webprog1</title>

  <link rel="stylesheet" href="css/style.css">

  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

  <script src="js/chart.js" defer></script>

</head>

<body>

  <header>

    <h1>Web-programozás-1 Előadás Házi feladat</h1>

  </header>

  <nav>

    <ul>

      <li><a href="index.html">Kezdőlap</a></li>

      <li><a href="table.html">Táblázat</a></li>

      <li><a href="html5.html">HTML5</a></li>

      <li><a href="chart.html" class="active">ChartJS</a></li>

      <li><a href="ajax.html">AJAX</a></li>

      <li><a href="oojs.html">OOJS</a></li>

      <li><a href="react.html">React</a></li>

    </ul>

  </nav>

  <aside>

    <p>Adatok grafikus ábrázolása ChartJS-sel.</p>

  </aside>

  <div id="content">

    <h2>Adatok ábrázolása</h2>

    <p>Válassz ki egy sort a táblázatból, hogy az adatai grafikonon megjelenjenek!</p>

    <table id="chartTable">

      <thead>

        <tr>

          <th>Adatsor</th>

          <th>Érték 1</th>

          <th>Érték 2</th>

          <th>Érték 3</th>

          <th>Érték 4</th>

          <th>Érték 5</th>

        </tr>

      </thead>

      <tbody>

        <tr>

          <td>Sor 1</td><td>10</td><td>20</td><td>30</td><td>40</td><td>50</td>

        </tr>

        <tr>

          <td>Sor 2</td><td>15</td><td>25</td><td>35</td><td>45</td><td>55</td>

        </tr>

        <tr>

          <td>Sor 3</td><td>5</td><td>10</td><td>15</td><td>20</td><td>25</td>

        </tr>

        <tr>

          <td>Sor 4</td><td>12</td><td>22</td><td>32</td><td>42</td><td>52</td>

        </tr>

        <tr>

          <td>Sor 5</td><td>8</td><td>18</td><td>28</td><td>38</td><td>48</td>

        </tr>

      </tbody>

    </table>

    <canvas id="chartCanvas"></canvas>

  </div>

  <footer>

    <p>Készítette: Balogh Róbert (SSDPW9) és Petrik Tibor (GDAZA8)</p>

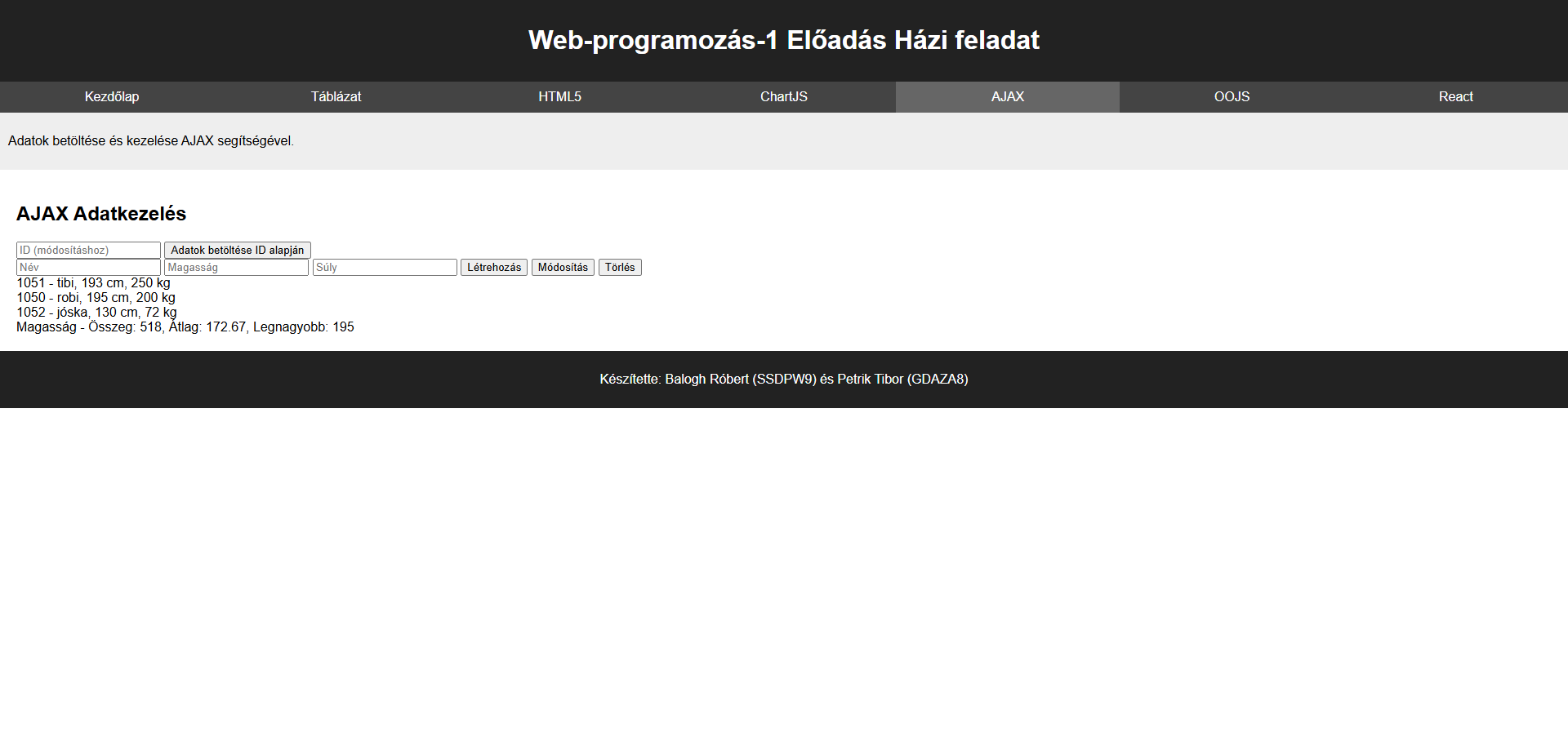
  </footer>

</body>

</html>

A feladat elvárásainak megfelelő kinézet.

## Ajax.html



A forráskódja:

<!DOCTYPE html>

<html lang="hu">

<head>

  <meta charset="UTF-8">

  <title>AJAX - Webprog1</title>

  <link rel="stylesheet" href="css/style.css">

  <script src="js/ajax.js" defer></script>

</head>

<body>

  <header>

    <h1>Web-programozás-1 Előadás Házi feladat</h1>

  </header>

  <nav>

    <ul>

      <li><a href="index.html">Kezdőlap</a></li>

      <li><a href="table.html">Táblázat</a></li>

      <li><a href="html5.html">HTML5</a></li>

      <li><a href="chart.html">ChartJS</a></li>

      <li><a href="ajax.html" class="active">AJAX</a></li>

      <li><a href="oojs.html">OOJS</a></li>

      <li><a href="react.html">React</a></li>

    </ul>

  </nav>

  <aside>

    <p>Adatok betöltése és kezelése AJAX segítségével.</p>

  </aside>

  <div id="content">

    <h2>AJAX Adatkezelés</h2>

    <div id="form-section">

      <input type="text" id="idField" placeholder="ID (módosításhoz)">

      <button id="getDataForId">Adatok betöltése ID alapján</button><br>

      <input type="text" id="name" placeholder="Név">

      <input type="text" id="height" placeholder="Magasság">

      <input type="text" id="weight" placeholder="Súly">

      <button id="createBtn">Létrehozás</button>

      <button id="updateBtn">Módosítás</button>

      <button id="deleteBtn">Törlés</button>

    </div>

    <div id="dataDisplay"></div>

    <div id="statsDisplay"></div>

    <div id="messageDisplay"></div>

  </div>

  <footer>

    <p>Készítette: Balogh Róbert (SSDPW9) és Petrik Tibor (GDAZA8)</p>

  </footer>

  <script>

    const code = "SSDPW9xyz123"; // saját kód

    const apiUrl = "http://gamf.nhely.hu/ajax2/";

    const nameInput = document.getElementById("name");

    const heightInput = document.getElementById("height");

    const weightInput = document.getElementById("weight");

    const idInput = document.getElementById("idField");

    const dataDisplay = document.getElementById("dataDisplay");

    const statsDisplay = document.getElementById("statsDisplay");

    const messageDisplay = document.getElementById("messageDisplay");

    function validateInputs() {

      const name = nameInput.value.trim();

      const height = heightInput.value.trim();

      const weight = weightInput.value.trim();

      if (!name || !height || !weight) {

        showMessage("Minden mező kitöltése kötelező.", true);

        return false;

      }

      if (name.length > 30 || height.length > 30 || weight.length > 30) {

        showMessage("A mezők max. 30 karakter hosszúak lehetnek.", true);

        return false;

      }

      return true;

    }

    function showMessage(msg, isError = false) {

      messageDisplay.textContent = msg;

      messageDisplay.style.color = isError ? "red" : "green";

    }

    function fetchData() {

      fetch(apiUrl, {

        method: "POST",

        headers: { "Content-Type": "application/x-www-form-urlencoded" },

        body: new URLSearchParams({ op: "read", code })

      })

        .then(res => res.json())

        .then(data => {

          dataDisplay.innerHTML = "";

          statsDisplay.innerHTML = "";

          if (data.list && data.list.length > 0) {

            data.list.forEach(row => {

              const div = document.createElement("div");

              div.textContent = `${row.id} - ${row.name}, ${row.height} cm, ${row.weight} kg`;

              dataDisplay.appendChild(div);

            });

            const heights = data.list.map(row => parseFloat(row.height)).filter(h => !isNaN(h));

            const sum = heights.reduce((a, b) => a + b, 0);

            const avg = (sum / heights.length).toFixed(2);

            const max = Math.max(...heights);

            statsDisplay.innerHTML = `Magasság - Összeg: ${sum}, Átlag: ${avg}, Legnagyobb: ${max}`;

          }

        });

    }

    document.getElementById("createBtn").addEventListener("click", () => {

      if (!validateInputs()) return;

      const data = {

        op: "create",

        code,

        name: nameInput.value,

        height: heightInput.value,

        weight: weightInput.value

      };

      fetch(apiUrl, {

        method: "POST",

        headers: { "Content-Type": "application/x-www-form-urlencoded" },

        body: new URLSearchParams(data)

      })

        .then(res => res.text())

        .then(res => {

          showMessage("Sikeres létrehozás.");

          fetchData();

        });

    });

    document.getElementById("updateBtn").addEventListener("click", () => {

      if (!validateInputs()) return;

      const id = idInput.value.trim();

      if (!id) return showMessage("Adja meg a módosítandó ID-t.", true);

      const data = {

        op: "update",

        code,

        id,

        name: nameInput.value,

        height: heightInput.value,

        weight: weightInput.value

      };

      fetch(apiUrl, {

        method: "POST",

        headers: { "Content-Type": "application/x-www-form-urlencoded" },

        body: new URLSearchParams(data)

      })

        .then(res => res.text())

        .then(res => {

          showMessage("Sikeres módosítás.");

          fetchData();

        });

    });

    document.getElementById("deleteBtn").addEventListener("click", () => {

      const id = idInput.value.trim();

      if (!id) return showMessage("Adja meg a törlendő ID-t.", true);

      const data = {

        op: "delete",

        code,

        id

      };

      fetch(apiUrl, {

        method: "POST",

        headers: { "Content-Type": "application/x-www-form-urlencoded" },

        body: new URLSearchParams(data)

      })

        .then(res => res.text())

        .then(res => {

          showMessage("Sikeres törlés.");

          fetchData();

        });

    });

    document.getElementById("getDataForId").addEventListener("click", () => {

      const id = idInput.value.trim();

      if (!id) return showMessage("Adja meg az ID-t.", true);

      fetch(apiUrl, {

        method: "POST",

        headers: { "Content-Type": "application/x-www-form-urlencoded" },

        body: new URLSearchParams({ op: "read", code })

      })

        .then(res => res.json())

        .then(data => {

          const found = data.list.find(row => row.id === id);

          if (!found) {

            showMessage("Nem található adat az ID alapján.", true);

          } else {

            nameInput.value = found.name;

            heightInput.value = found.height;

            weightInput.value = found.weight;

            showMessage("Adatok betöltve.");

          }

        });

    });

    fetchData();

  </script>

</body>

</html>

Feladat leírásainak megfelelően adatok felvitele, módosítása és törlésére szolgál.

**Tartalom**

[**Dokumentáció a Web-programozás 1 – Beadandó feladathoz** 1](#_Toc195983771)

[Hogyan készült az alkalmazás? 1](#_Toc195983772)

[**A munka menete:** 1](#_Toc195983773)

[A fejlesztés eredménye 3](#_Toc195983774)

[Kezdőlap 6](#_Toc195983775)

[HTML5 8](#_Toc195983776)

[Chart.html 12](#_Toc195983777)

[Ajax.html 14](#_Toc195983778)