

JEGYZŐKÖNYV

Web Technológia Alapjai

Féléves feladat

Cipőbolt weboldal

Készítette: Kun Dávid

Neptunkód: GEB15I

Dátum: 2025. December

Miskolc, 2025

Tartalomjegyzék

Bevezetés	2
1. Feladat: HTML	3
1.1. index.html fájl	3
1.2. catalog.html.....	5
1.3. cart.html.....	8
1.4. checkout.html.....	10
1.5 about.html	11
2. JavaScript, JQuery, AJAX, JSON	13
2.1. JSON	13
2.2. JQuery és AJAX.....	14
2.3 JavaScript és validation.js.....	16
2.4 CSS	17

Bevezetés

Az én féléves feladatom témája egy cipőbolt és az ehhez tartozó weboldal. Azért erre esett a választásom, mivel nagy cipő rajongó vagyok és úgy gondoltam egy cipőboltnak megvan a megfelelő struktúrája, ahhoz, hogy a féléves feladatomat elkészítsem.

A beadandóm elkészítéséhez 5 darab HTML fájlt készítettem el. Ezek a következők: index.html, checkout.html, catalog.html, cart.html és about.html. Ebben az öt fájlban van megvalósítva minden követelmény, amely ehhez a beadandóhoz lett megszabva.

A CSS-t egy külön fájlban készítettem el, amely minden oldalhoz tartalmazza a css adatokat.

A JavaScript szintén külső fájlban van megvalósítva és ebből kettő is van. Az egyik az maga a JavaScript fájl, ahol az oldal működéséhez szükséges kódolás van megvalósítva. Ez tartalmazza a JQuery-t, ami elég sok fealdatot ellát, illetve az AJAX-et is. A másik fájl pedig, ellenőrzést végez. Ez az ellenőrzés a checkout.html-ben van használva, hogy ott a megfelelő formátumban legyenek megadva az adatok.

Található még egy json fájl is a beadandóm fájlai között. Ebben a fájlban vannak eltárolva a különböző cipők adatai, amelyek a catalog.html-ben találhatóak meg.

1. Feladat: HTML

1.1. index.html fájl

Az index.html fájl maga a weboldalam főoldala. Innen lehet a különböző oldalakra elnavigálni, valamint tartalmaz egy rövid bemutatkozást is.

```
<header>

<nav>

  <a href="index.html" class="logo">KunKicks</a>

  <ul>

    <li><a href="index.html">Főoldal</a></li>
```

```
<li><a href="catalog.html">Katalógus</a></li>

<li><a href="about.html">Rólunk</a></li>

<li><a href="cart.html">Kosár (<span id="cart-count">0</span>)</a></li>

<li><a href="checkout.html" class="btn">Pénztár</a></li>

</ul>

</nav>

</header>
```

A navigáció a header részeként lett implementálva és egy táblázat formájában valósult meg. Itt megtalálhatóak a hivatkozások, amelyek el vannak látva a “btn” osztállyal, amely CSS-ért felel.

```
<main>

<section class="hero">

<div class="hero-text">

<h1>Lépj szintet a stílusoddal.</h1>

<p>A legújabb sneakerek, a legjobb árakon. Fedezd fel a KunKicks univerzumot még ma.</p>

<br>

<a href="catalog.html" class="btn">Vásárlás Most</a>

</div>

<div class="hero-image">



</div>
```

```
</section>
```

```
</main>
```

Az alábbi részlet egy rövid bemutatkozást tartalmaz. Itt is van egy referencia a catalog.html-re ami szintén a “btn” osztályal van ellátva. Ezen kívül itt még megtalálható egy kép is, amely szimplán dizájn szempontjából került ide. Ez a kép a “Nike Air Jordan” alt-ot kapta.

Az index.html itt érhető el: [index.html](#)

1.2. catalog.html

A catalog.html egy katalógus, ahol a felhasználó tud cipők közül választani. Itt is megtalálható a navigációs menü, amellyel az oldalak között lehet lépkedni. Ezen kívül még található itt egy táblázat is.

```
<main>
```

```
<section class="hero">
```

```
<div class="hero-text">
```

```
<h1>Lépj szintet a stílusoddal.</h1>
```

```
<p>A legújabb sneakerek, a legjobb árakon. Fedezd fel a KunKicks univerzumot még ma.</p>
```

```
<br>
```

```
<a href="catalog.html" class="btn">Vásárlás Most</a>
```

```
</div>
```

```
<div class="hero-image">
```

```

```

```
</div> <section>
```

<h2>Mérettáblázat</h2>

```
<table>
```

```
  <thead>
```

```
    <tr>
```

```
      <th>EU Méret</th>
```

```
      <th>US Méret</th>
```

```
      <th>UK Méret</th>
```

```
      <th>CM</th>
```

```
    </tr>
```

```
  </thead>
```

```
  <tbody>
```

```
    <tr>
```

```
      <td>40</td>
```

```
      <td>7</td>
```

```
      <td>6</td>
```

```
      <td>25</td>
```

```
    </tr>
```

```
    <tr>
```

```
      <td>41</td>
```

```
      <td>8</td>
```

```
      <td>7</td>
```

```
      <td>26</td>
```

```
</tr>

<tr>

<td>42</td>

<td>8.5</td>

<td>7.5</td>

<td>26.5</td>

</tr>

<tr>

<td>43</td>

<td>9.5</td>

<td>8.5</td>

<td>27.5</td>

</tr>

</tbody>

</table>

</section>

</section>

</main>
```

Ez a táblázat a különböző országok közötti cipő méret átváltásában segít. Egy egyszerű table struktúraként van megvalósítva, amely el van látva CSS-el.

Ezen az oldalon található még az a rész is, ahol a json fájlból betöltésre kerülnek különböző cipők.

```
<section id="product-container" class="product-grid">  
    <!-- AJAX content will be loaded here -->  
    <p>Termékek betöltése...</p>  
</section>
```

Ez a rész jelenleg csak ennyi a catalog.html-ben, ugyanis JavaScript és AJAX segítségével van ez megvalósítva, ami pedig külső fájlban lett elkészítve.

A catalog.html itt érhető el: [catalog.html](#)

1.3. cart.html

A cart.html fájl felelős azért, hogy a felhasználónak legyen egy kosara, ahol a cipőket tudja tenni. Ebben a fájlban is meg van valósítva a navigációs szegmens, valamint egy táblázat segítségével, illetve JavaScript segítségével vannak tárolva a cipők adatai amelyeket a felhasználó kiválasztott.

```
<main>  
    <h1>Kosarad tartamla</h1>  
  
    <div class="cart-container"  
        style="background: white; padding: 2rem; border-radius: 10px; box-shadow: var(--  
shadow);">  
        <!-- Static example content for functionality demo -->  
        <table>  
            <thead>  
                <tr>  
                    <th>Termék</th>  
                    <th>Méret</th>  
                    <th>Ár</th>  
                    <th>Törlés</th>
```

```
</tr>
</thead>
<tbody>
<tr>
    <td>Nike Air Max 90</td>
    <td>42</td>
    <td>45 000 Ft</td>
    <td><button style="background:none; border:none; color:red; cursor:pointer;">X</button></td>
</tr>
<tr>
    <td>Adidas Ultraboost</td>
    <td>43</td>
    <td>55 000 Ft</td>
    <td><button style="background:none; border:none; color:red; cursor:pointer;">X</button></td>
</tr>
</tbody>
</table>

<div id="cart-total">
    Összesen: 100 000 Ft
</div>

<div style="text-align: right; margin-top: 2rem;">
    <a href="catalog.html" class="btn" style="background-color: var(--secondary-color); margin-right: 1rem;">Vásárlás folytatása</a>
```

```
<a href="checkout.html" class="btn">Tovább a pénztárhoz</a>
</div>
</div>
</main>
```

Ez most egy fiktív kosár, amit az oldal automatikusan betölt majd minden demonstráció céljából, hogy látható legyen, hogy a kosár működik

A cart.html itt érhető el: [cart.html](#)

1.4. checkout.html

A checkout.html az után jön be a képhez, miután a felhasználó már választott magának cipőket és fizetni szeretne. Itt is megvalósul a navigációs fül, valamint, itt lehet beütni a szállítási adatokat is. Ezek a szállítási adatok az alábbi módon kerültek megvalósításra:

```
<div class="form-group">
    <label for="name">Teljes név:</label>
    <input type="text" id="name" name="name" placeholder="Pl. Kiss János">
</div>

<div class="form-group">
    <label for="delivery-date">Kért szállítási nap:</label>
    <input type="date" id="delivery-date" name="delivery-date">
</div>

<div class="form-group">
    <label for="city">Város (Datalist példa):</label>
    <input list="cities" id="city" name="city">
    <datalist id="cities">
        <option value="Budapest">
        <option value="Debrecen">
        <option value="Szeged">
    </datalist>
</div>
```

```

<option value="Miskolc">
    <option value="Pécs">
</datalist>
</div>

<div class="form-group">
    <label for="packaging-color">Díszcsomagolás színe (opcionális):</label>
    <input type="color" id="packaging-color" name="packaging-color" value="#ff4757">
</div>

<h2>Fizetési Mód</h2>

<div class="form-group radio-group">
    <div style="margin-bottom: 0.5rem;">
        <input type="radio" id="card" name="payment" value="card">
        <label for="card" style="display:inline;">Bankkártya</label>
    </div>
    <div>
        <input type="radio" id="cod" name="payment" value="cod">
        <label for="cod" style="display:inline;">Utánvét</label>
    </div>
</div>

```

Noha ez nem az összes mező, amivel a felhasználó szembe fog jönni, mikor fizetni óhajt, de ezek kiemelendőek, mivel van közte radio button, checkbox, dátum kiválasztás és még egy színválasztó is.

Ezek mind input-ként vannak definiálva és mindegyiket a validation.js ellenőrzi, hogy csak is a megfelelő formátumú adatok kerüljenek bele.

A checkout.html fájl itt található: [checkout.html](#)

1.5 about.html

Az about.html egy fiktív leírást tartalmaz a weboldalról és annak történetéről, valamint tartalmaz még egy rövid videót is, ami bemutatkozásként került bele.

```
<div style="display: flex; gap: 2rem; align-items: start;">

    <div style="flex: 1;">

        <!-- Requirement: Numbered List -->

        <h3>Történetünk lépései</h3>

        <ol>

            <li>2010: Az ötlet megszületése egy kis garázsban.</li>

            <li>2012: Első webshop megnyitása.</li>

            <li>2015: Fizikai üzlet nyitása Budapesten.</li>

            <li>2020: Nemzetközi terjeszkedés.</li>

            <li>2024: Magyarország vezető sneaker forgalmazója.</li>

        </ol>

    </div>
```

Az alábbi táblázat tartalmazza a weboldal fiktív történetét. Ez el van látva még megfelelő CSS-el is, hogy szépen nézzen ki.

```
<div style="flex: 1;">

    <!-- Requirement: Video with custom JS controls -->

    <h3>Nézd meg bemutatkozó videónkat!</h3>

    <div class="video-container">

        <div id="player"></div>

    </div>
```

```

<div class="video-controls">

    <button onclick="playVideo()" class="btn">Lejátszás</button>

    <button onclick="pauseVideo()" class="btn"
        style="background-color: var(--secondary-color);">Szünet</button>

    <button onclick="stopVideo()" class="btn" style="background-color:
#e74c3c;">Megállítás</button>

</div>

</div>

</div>

</section>

```

Az alábbi kódrészlet pedig a bemutatkozó videó lejátszásáért és vezérlésért felelős. Tartalmazza a kezeléshez szükséges gombokat, illetve a videó beágyazást is. Ez is el van látva megfelelő CSS-el.

Az about.html fájl itt található meg: [about.html](#)

2. JavaScript, JQuery, AJAX, JSON

2.1. JSON

A weboldalhoz készített JSON fájl, a products.json. Ez tartalmazza azoknak a cipőknek az adatait, amelyeket a felhasználó lát catalog.html oldalon. Ez a fájl szolgál "adatbázisként". Egy JSON formátumú tömböt tartalmaz, amelyben a termékobjektumok vannak tárolva. Ez teszi lehetővé, hogy a main.js dinamikusan töltse be az oldalt, ne kelljen kézzel beírni a HTML-be minden cipőt.

Egy cipőhöz tartozó adatblokk a fájlban a következőképpen épül fel:

{

 "id": 1,

```

    "name": "Nike Air Max 90",
    "price": 45000,
    "category": "Utcai",
    "image": "https://images.unsplash.com/photo-1542291026-7eec264c27ff?w=500&auto=format&fit=crop&q=60&ixlib=rb-4.0.3",
    "description": "Klasszikus kényelem és stílus."
}


```

Tartalmaz egy ID-t, ami azonosításra szolgál. Tartalmaz egy nevet, egy árat, valamint egy kategóriát is. Ezeken kívül található még benne egy kép is az adott cipőről, illetve egy rövid leírás is.

A teljes products.json itt található meg: [products.json](#)

2.2. JQuery és AJAX

A JQuery és AJAX egy pontba lettek válogatva, hiszen ugyanabban a main.js fájlból vannak megvalósítva. Ez a kettő szolgál az oldal motorjaként. Ez a fájl felelős az interaktivitásért. Főként jQuery könyvtárat használ.

- Funkciói:
 - Azonosítja az aktív menüpontot.
 - AJAX hívással betölti a termékeket a products.json-ból.
 - Kezeli a kosárba rakás folyamatát és animációit (értesítés megjelenítése).
 - Vezérli a YouTube videólejátszót (YouTube Iframe API).

AJAX kérés (Adatbetöltés) Ez a legtechnikásabb rész, ez köti össze az adatot a megjelenítéssel:

```

$.ajax({
  url: 'data/products.json', // Innen olvassa az adatokat
  type: 'GET',

```

```

dataType: 'json',

success: function (data) {

    var html = "";

    $.each(data, function (index, product) {

        // Itt generálja le dinamikusan a HTML kártyákat

        html += '<div class="product-card" id="product-' + product.id + '">';

        html += '';

        html += '<h3>' + product.name + '</h3>';

        // ... (többi elem)

        html += '</div>';

    });

    $('#product-container').html(html); // Beillesztés az oldalba

}

});

```

Értesítés és DOM manipuláció Hogyan ad visszajelzést a felhasználónak:

```

$(document).on('click', '.add-to-cart', function () {

    var name = $(this).data('name');

    // Új HTML elem (értesítés) létrehozása jQuery-vel

    var $notification = $('

...</div>');


```

```
$('#notification-area').append($notification);

// Animáció: becsúszik, vár, majd eltűnik

$notification.slideDown(300).delay(2000).slideUp(300);

});
```

A teljes kódot tartalmazó main.js itt található meg: [main.js](#)

2.3 JavaScript és validation.js

Ez már tisztán JavaScript. Még az előző pont tartalmazott JQuery-t is, illetve AJAX-et, addig ez már vanilla JavaScript. A validation.js űrlap ellenőrzést végez, amely a checkout.html-hez készült és itt is van alkalmazva.

Az eseményfigyelő és a tiltás Hogyan akadályozza meg a hibás űrlap beküldését:

```
form.addEventListener('submit', function (event) {

    let isValid = true;

    // 1. Name Validation (NotEmpty)

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

    if (nameInput.value.trim() === '') {

        showError(nameInput, 'A név megadása kötelező!');

        isValid = false;

    } else {

        showSuccess(nameInput);

    }

    // 2. Email Validation (Regex)
```

```

const emailInput = document.getElementById('email');

const emailRegex = /^[^@\s]+@[^\s]+\.\[^@\s]+\$/;

if (!emailRegex.test(emailInput.value)) {

    showError(emailInput, 'Adjon meg egy érvényes email címet!');

    isValid = false;

} else {

    showSuccess(emailInput);

}

if (!isValid) {

    event.preventDefault(); // Megállítja az űrlap elküldését, ha hiba van

} else {

    alert('Sikeres rendelés!');

}

});


```

Ez a kódrészlet jelenleg a megadott nevet és emailt ellenőrzi. Az emailcím ellenőrzéséhez pedig regex-et, vagyis reguláris kifejezésket használ.

A validation.js itt érhető el: [validation.js](#)

2.4 CSS

Az oldal stíluslapja. A weboldalhoz tartozó CSS úgy lett kialakítva, hogy minden az öt HTML fájl egy CSS-t használjon. Ezen kívül egy egyszerű és modern kinézetet ad. A legfontosabb technikák, amelyeket a CSS alkalmaz ennél az oldalnál: Flexbox (elrendezéshez), reszponzivitás (@media), hover effektek (gombok, linkek). Az oldal szívélegysége pedig a következő: Fehér, szürke és egy kiemelő piros szín (#ff4757).

```
.product-grid {
```

```
display: flex; /* Rugalmas dobozmodell /  
flex-wrap: wrap; / Ha nem fér el, új sorba töri /  
gap: 20px; / Távolság az elemek között */  
justify-content: space-between;  
}  
  
.product-card {  
width: 30%; /* Egy sorban kb. 3 termék fér el */  
}
```

Ezen felül még az oldal reszponzívan is lett kódolva, amit az alábbi kódrészlet valósít meg:

```
@media (max-width: 768px) {  
  
.product-card {  
width: 45%; /* Mobilon már csak 2 termék / sor /  
}  
  
.hero {  
flex-direction: column; / A hero szekció egymás alá kerül */
```

A teljes CSS kódot tartalmazó fájl ezen a linken érhető el: [style.css](#)