

Jegyzőkönyv

Web technológiák 1

Féléves feladat

Kedvenc kutyaajtáim

Készítette: **Takács Bálint Zétény**

Neptunkód: **GJWXEU**

Dátum: **2025.11.18.**

Miskolc, 2025

Tartalomjegyzék:

- [Tartalomjegyzék](#)
- [Bevezetés](#)
- [A feladat leírása](#)
- [HTML](#)
 - [HTML5 szerkezet és oldalterv](#)
 - [Menüszerkezet](#)
 - [Űrlap megvalósítása](#)
 - [Videó és vezérlő gombok](#)
- [CSS](#)
 - [A stíluslap elején az alapértelmezett beállítások törlése található](#)
 - [HTML elemek azonosítása CSS-ben](#)
 - [Főbb osztályok](#)
 - [Háttérszínek](#)
 - [Linkek formázása, az alapértelmezett formázás elhagyása](#)
 - [CSS Grid és Gap](#)
- [JavaScript, JQuery, JSON, AJAX](#)
 - [Videó vezérlése JavaScript segítségével](#)
 - [jQuery-animációk](#)
 - [Űrlap-ellenőrzés JavaScript-tel](#)
 - [Új HTML elem és meglévő módosítása](#)
 - [Egy fájl elkészítése JSON formátumba, Node.js](#)
 - [Egy fájl elkészítése JSON formátumba, Node.js nélkül](#)
 - [AJAX metódus használatával JSON fájl megjelenítése a weboldalon](#)

Bevezetés

A beadandó feladatom célja egy többoldalas, reszponzív weboldal létrehozása volt, amely bemutatja a kedvenc kutyafajtáimat. A projekt témáját személyes érdeklődés alapján választottam, hiszen a kutyák az ember legjobb barátai, és szerettem volna ezt a gondolatot egy esztétikus, informatív, mégis könnyen kezelhető webes felületen megjeleníteni. A munka során törekedtem a HTML5 szemantikus elemek, a CSS3 formázási lehetőségek és az alapvető JavaScript-funkciók (interaktív gombok, videóvezérlés) megfelelő alkalmazására. A projekt kiváló gyakorlási lehetőséget nyújtott a modern webes technológiák integrált használatára.

A feladat leírása

A feladat egy többoldalas weboldal elkészítése volt, amely négy, számomra kedves kutyafajtát – a Tacsikót, Dobermannt, Rottweilert és Pitbullt – mutatja be. A projekt célja nem csupán az információátadás, hanem az is, hogy a felhasználó egy vizuálisan egységes, könnyen kezelhető és interaktív felületen keresztül ismerkedhessen meg ezekkel a fajtákkal.

A weboldal fő elemei:

- Egy fejléc (header), amely tartalmazza a főcímet és egy kutya ikont.
- Egy navigációs menü, amelyből elérhetők az egyes fajták saját aloldalai, valamint az „Összehasonlítás” oldal.
- Egy fő tartalom (article), amely tartalmazza az adott lap tartalmát. Ezt a rész (section) blokkok osztják fel.
- Egy oldalsáv (aside), amely érdekességeket és idézetet tartalmaz a kutyákról.
- Egy lábléc (footer), ahol a szerzői információk és elérhetőségek találhatók.

A weboldal megjelenése és funkcionalitása tudatos tervezés eredménye volt. A HTML5 szerkezeti elemei (header, nav, article, section, aside, footer) segítségével átlátható, logikusan tagolt dokumentumot hoztam létre.

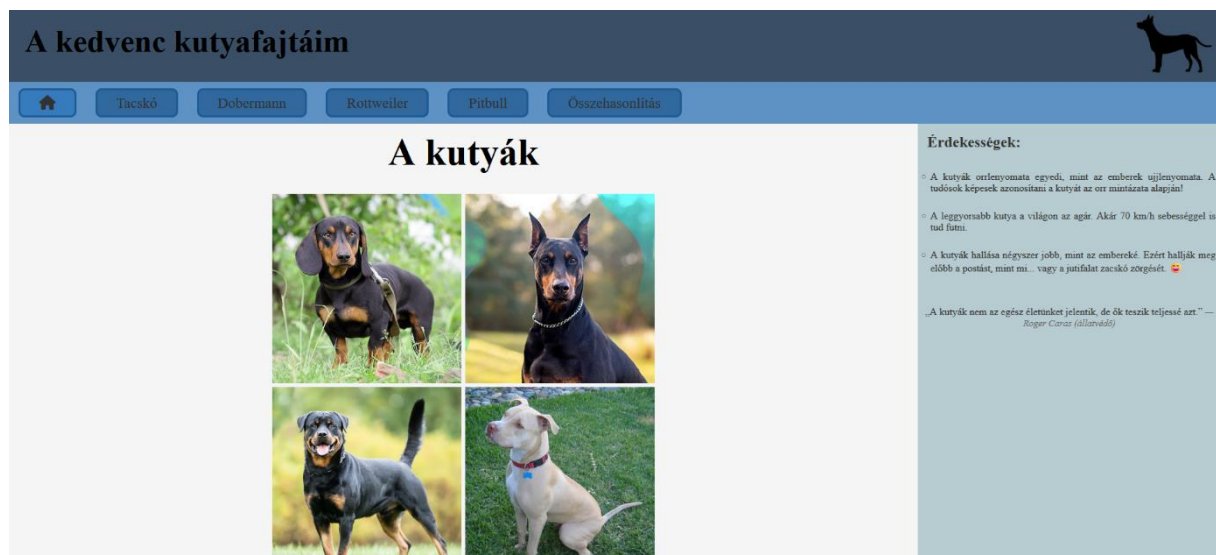
A CSS a vizuális egységért felelt:

- rugalmas elrendezés,
- jól olvasható tipográfia,
- esztétikus színvilág,
- lekerekített gombok, árnyékok és ikonok használata (Font Awesome segítségével).

A JavaScript interaktív funkciókat biztosít, mint például:

- a videó lejátszásának/szüneteltetésének vezérlése,
- az űrlap adatainak ellenőrzése és a „Beküldés” gomb megjelenítése,
- bizonyos elemek animációját is kezeli.

A projekt során figyelmet fordítottam arra, hogy a felhasználói élmény és az esztétikai megjelenés harmóniában legyen. A weboldal könnyen navigálható, vizuálisan vonzó, és technikailag megfelel a modern webes követelményeknek. A többoldalas struktúra, az egységes stíluslap, valamint a médiaelemek (képek, videók) integrációja mind hozzájárultak ahhoz, hogy a végeredmény egy teljes értékű, informatív és szórakoztató webes alkalmazás legyen.



1. ábra: A kezdőoldal

1. HTML

Oldal	Tartalom	Cél
index.html	Bevezetés a kutyák világába, videó, űrlap	Téma bemutatása, interaktivitás
1tacko.html	Tackó jellemzése, története, típusai	Konkrét fajta bemutatása
2dobermann.html	Dobermann tulajdonságai	Konkrét fajta bemutatása
3rottweiler.html	Rottweiler jellemzése	Konkrét fajta bemutatása
4pitbull.html	Pitbull jellemzése, tévhitek tisztázása	Konkrét fajta bemutatása
5tablazat.html	Összehasonlító táblázat a fajtákról	Áttekintő, információs oldal

HTML5 szerkezet és oldalterv:

- Header: oldal címe, ikon
- Nav: főmenü
- Article: fő tartalom
- Aside: érdekességek, kiegészítő információ
- Footer: elérhetőség, szerzői jogok

Menüszerkezet:

- Kezdőlap
- Tackó
- Dobermann
- Rottweiler
- Pitbull
- Összehasonlítás

Űrlap megvalósítása:

Egy nevezést terveztem, de csak a 4 kutyafajtát, és az űrlapba meg lehet adni a kutya adatait.

- Egy soros szöveges mező (text): kutyanéve
- Többsoros szöveg (textarea): leírás
- Legördülő lista (select): Faj kiválasztás
- Rádiógomb (radio): Nem kiválasztása
- Jelölő négyzetek (checkbox): Tulajdonságok
- Színválasztó (color): Szőrzet színe
- Dátumválasztó (date): Születési dátum
- Gombok (button): Ellenőrzés, beküldés

Videó és vezérlő gombok:

Egy videó kutyákról és kettő vezérlő gomb, ami JavaScript-tel van vezérelve, megállító és elindító, elejére tekerés.

2. CSS

A stílustervezés célja egy modern, harmonikus, kutyás témához illő megjelenés megalkotása volt.

A színek a természetet és nyugalmat idézik:

- Kék és szürke árnyalatok
- Világos háttér
- Sötét fejlécek/láblécek

A tervezés fő szempontjai:

- Egységes színpaletta minden oldalon
- Áttekinthető, reszponzív elrendezés
- Kiemelések (hover-effektek, árnyékok) a jobb felhasználói élményért
- Könnyen olvasható, kontrasztos betűszínek

A stíluslap nagyrésze a **style.css** fájlban található, ezen kívül néhány apró formázás közvetlenül a HTML head részében, a <style> tag-en belül és a style="" attribútumban is megjelenik.

A stíluslap elején az alapértelmezett beállítások törlése található:

```
* {  
    padding: 0;  
    margin: 0;  
}
```

HTML elemek azonosítása CSS-ben:

Azonosító (id)	Osztály (class)
<pre>#footerMargin1 { margin-top: 2500px; }</pre>	<pre>.formGomb { width: 100px; height: 25px; }</pre>

Főbb osztályok:

- container1 – container6: az egyes oldalak tartalmi keretei
- focim, mainh2, mainp: címsorok és szövegek formázása
- mainul, asideLista: listák elrendezése és margózás
- gomb, formGomb: gombok mérete, lekerekítése és hover-effektje
- grid: CSS-grid alapú kétszlopos elrendezés

Háttérszínek:

- header, footer: sötétkék
- nav: világoskék
- article: törtfehér
- aside: eltérő, kiemelő, de hasonló stílusú szín

Linkek formázása, az alapértelmezett formázás elhagyása:

```
nav ul li a {  
    color: #333;  
    text-decoration: none;  
}
```

CSS Grid és Gap:

Grid a főoldalon található, kétoszlopos szakasz elrendezésére való.

```
.grid {  
    display: grid;  
    grid-template-columns: auto auto;  
    gap: 200px;  
}
```

3. JavaScript, jQuery, JSON, AJAX

A JavaScript-kód célja a felhasználói interakciók kezelése:

- A videó vezérlése gombokkal (play/pause, újraindítás)
- Az űrlapadatok ellenőrzése
- A jQuery animációk segítségével az oldal dinamikusabbá tétele
- CSS-stílusok módosítása JavaScriptből, hogy vizuálisan is jelezze a hibákat

A kód szerkezete három nagy részre bontható:

1. Videóvezérlés (JavaScript)
2. jQuery animációk
3. Űrlap-ellenőrzés és hibajelzés

Videó vezérlése JavaScript segítségével:

```
function toggleVideo() {  
    var video = document.getElementById("videoId");  
    if (video.paused)  
        video.play();  
    else  
        video.pause();  
}
```

A HTML gombjaiba írtam onClick funkciót, ami meghívja az adott függvényeket.

jQuery-animációk:

- Tartalom elrejtése/megjelenítése
- Képek és táblázatok elrejtése/megjelenítése
- Oldalsáv-tartalom animált betöltése
- Rejtett kép animálása

Űrlap-ellenőrzés JavaScript-tel:

- 1) Az adatok lekérése
- 2) Alapértelmezett stílus visszaállítása
- 3) Ellenőrzések és hibajelzések
- 4) Sikeres ellenőrzés ellenőrzése

Új HTML elem és meglévő módosítása:

A fenti JavaScript kód egy összehasonlító táblázat kezelésére szolgál, amely kutyaajták adatait jeleníti meg. A `beszur()` függvény segítségével a felhasználó által megadott adatok új oszlopként kerülnek a táblázatba, miközben a táblázat sorainak száma és a korábbi adatok figyelembevételével szükség esetén a régi oszlopok törlődnek. Az adatok között szerepel a fajta, eredet, méret, marmagasság, testalkat, szőrzet, élettartam és mozgásigény. A függvény gondoskodik arról is, hogy az első sorban a tulajdonságok megjelenése külön stílusos legyen. Ezt követően az `ellenorizMarmagassag()` függvény elemzi a marmagasság értékeket, kinyerve az átlagos értéket a szövegből, majd meghatározza a legnagyobb marmagasságot. Az eredményt a kód a felhasználó számára egy külön div-ben jeleníti meg.

Egy fájl elkészítése JSON formátumba, Node.js:

A megvalósítás során a HTML form mezőin keresztül lehetett megadni az értékeket, amelyekből a JavaScript lekérte az adatokat, majd egy objektumba rendezte őket. Node.js környezetben a `fs` modul segítségével a függvény képes volt a JSON objektumot fájlba írni, így az automatikusan létrejött a számítógépen vagy a szerveren. A JSON szöveg létrehozásához a `JSON.stringify` metódust használtuk, amely a JavaScript objektumot JSON formátumba alakítja.

Egy fájl elkészítése JSON formátumba, Node.js nélkül:

Ebben az esetben a JSON objektumot egy Blob objektumból hoztuk létre, amelyet egy ideiglenes URL segítségével lehetett letölteni a felhasználó számítógépére. A függvény így a felhasználó közvetlen gombnyomásra indítja a fájl letöltését, biztosítva ezzel a JSON adat tárolását anélkül, hogy szerveroldali kódra lenne szükség.

AJAX metódus használatával JSON fájl megjelenítése a weboldalon:

A megoldás során jQuery AJAX metódust alkalmaztunk, amely lehetővé teszi a JSON fájl lekérését és a kapott adatok feldolgozását JavaScript objektumként. Az oldal betöltődése után a `$(document).ready()` biztosítja a DOM elemek elérhetőségét, majd a `$.ajax()` metódus GET kéréssel lekéri az adatokat, és a success callback függvény segítségével a JSON tartalmat HTML listaként jeleníti meg az oldalon.