**LUCRARE DE ATESTAT PROFESIONAL**

**LA INFORMATICĂ**

*Csaknekedkislány tánczenekar*

An școlar 2024-2025

*ÎNTOCMIT DE*  *PROFESOR ÎNDRUMĂTOR*

Tolnai Márk – XII. B. Boda Szilárd

**BEVEZETÉS**

Az általam választott téma: a Csaknekedkislány tánczenekar. Azért választottam ezt, mint téma, mert nagy rajongója vagyok a zenekarnak, és szerettem volna valami személyeset, hozzám közelállót alkotni.

A Csaknekedkislány 2011-ben alakult, és azóta is meghatározó szereplője a magyar könnyűzenei színtérnek. A zenekar stílusára nagy hatással volt a beatkorszak. Szövegeik lelki, problémákat és társadalomkritikkat is megfogalmaznak egyaránt. Az oldalon meghallgatható a zenekar 5 lemeze.

A weboldal elkészítéséhez HTML, CSS és JavaScript nyelveket használtam, amelyek lehetővé teszik az egyszerre informatív, esztétikus és dinamikus megjelenést. A HTML felelős a weboldal tartalmának strukturálásáért, míg a CSS segítségével a vizuális elemeket, például a színeket, betűstílust és az elrendezést alakítottam ki. A JavaScript felelős bizonyos elemek interaktív és logikus működéséért.

**Html**

A HTML (HyperText Markup Language) egy jelölőnyelv, amelyet weboldalak készítésére használnak. Ez a nyelv határozza meg, hogy egy weboldal tartalma hogyan épül fel, és hogyan jelenik meg a böngészőkben. A HTML nem programozási nyelv, mivel nem tartalmaz logikai műveleteket vagy vezérlési szerkezeteket, hanem inkább szerkezeti alapot nyújt a weboldalak számára. A dokumentumokat HTML-címkék segítségével építik fel, amelyek jelölik az oldalon megjelenő különböző elemeket, például címsorokat, bekezdéseket, listákat, képeket és hivatkozásokat.

A HTML a webes technológiák egyik alapköve, amelyet gyakran használnak CSS-sel (Cascading Style Sheets) és JavaScript-tel együtt, hogy vizuálisan vonzó, interaktív és dinamikus weboldalakat hozzanak létre.

**CSS**

A CSS (Cascading Style Sheets) egy stílusleíró nyelv, amelyet weboldalak megjelenésének meghatározására használnak. A CSS segítségével szabályozhatjuk az elemek kinézetét, például a színeket, betűtípusokat, elrendezést stb. Míg a HTML a weboldal tartalmi részének meghatározására van, a CSS felelős annak formázásáért. A CSS-t többféleképpen lehet alkalmazni egy weboldalon: beágyazottan, a dokumentum fejében, vagy egy külső fájlban, amelyet a **<link>** elem segítségével csatolunk a HTML fájlhoz. Az utóbbi módszer a legelterjedtebb, mivel a külső CSS fájlok megkönnyítik a stílusok karbantartását és az egységes megjelenést több oldal között.

**JavaScript**

A JavaScript egy magas szintű, dinamikus, objektumorientált programozási nyelv, amelyet elsősorban webfejlesztéshez használnak. A JavaScript teszi lehetővé, hogy a weboldalak interaktívak és dinamikusak legyenek, például animációk vagy felhasználói interakciók kezelése révén. Ez a nyelv a webes technológiák egyik alapköve a HTML és a CSS mellett, és szinte minden modern böngészőben natívan támogatott.

A JavaScript esemény vezérelt, vagyis az eseményekre reagálva hajt végre műveleteket. A nyelv szintaxisa a C-nyelv alapjaira épül, és támogatja az objektumorientált, funkcionális és imperatív programozási paradigmákat is, ami rugalmassá teszi különböző projektekhez.

**A WEBOLDAL BEMUTATÁSA**

A weboldalam 4 oldalból áll, ezek között kapcsolatot hoztam létre, így bármelyik oldalról mindegyikre át lehet ugrani.

A weboldal tartalmi részét (HTML), stilizálását (CSS) és helyenkénti működési logikáját (JavaScript) külön fájlokban határoztam meg, majd ezeket <link> elemekkel kötöttem össze.

(pl: **<link rel="stylesheet" href="../css/navbar\_styles.css">).**

A projektet tartalmazó mappák felépítése/hierarhiája, ahol a html, css, js és content, a weboldal ideális működéséhez szükséges fájlokat tartalmazó mappák:

|  |  |
| --- | --- |
| **Project** |  |
|  | **html** |
|  | **css** |
|  | **js** |
|  | **content** |

**Navigációs sáv**

Az oldal tetején horizontálisan, fixált megjelenésű sávban elterülő linkek listája, navigációs sávként funkcionálnak. Minden oldal tartalmi része ezen sáv és a hozzá tartozó CSS fájl meghatározásával kezdődik.



Az itt látható HTML kód a navigációs sávot definiálja listaként, amely a Kezdőlap, Albumok, Alapítótagok és Személyes kötődések menüpontokat tartalmazza, és közösségi média hivatkozásokat a banda oldalaihoz. Az **id** és **class** attribútumok segítik a stílus és funkciók kezelését **CSS** használatával.

A **navbar\_styles.css** nevű fájlban határoztam meg a navigációs sáv kinézetét:

A színeket **HEX** kódokkal határoztam meg. A sáv magasságát és szélességét a height és width értékei adják meg. Jelen esetben a “**width:100%**;” azt jelenti, hogy a sáv szelessége az oldal teljes szélességével megegyező. A **height** egy konkét értéket az a sáv magasságának (**50px**). A “**position: fixed;**” sor eredményeként marad a navigációs sáv folyon a weboldal tetején, le és felfele görgetéstől függetlenül. A **display**, **align**, **margin**, **padding**, stb. tulajdonságok a dinamikus és reszponzív elrendezések kialakítására szolgálnak.

A **hover effect** egy olyan interaktív elem, amely akkor aktiválódik, amikor a felhasználó az egérmutatót/kurzort egy elem fölé viszi. Lehetővé teszi bizonyos animációk bejátszását, ezáltal interaktívabbá téve weboldalunkat.

Mikor a kurzor az egyes menüpontok felett van, azokon megjelenik egy zöld aláhúzás, de amennyiben a kurzort eltávolítjuk, az aláhúzás is eltűnik. A **::before** elem lehetővé teszi, hogy tartalmat helyezzünk el egy elem előtt.

A közösségi média ikonjainak esetében is jelen van ezen **hover effect**, sőt, ebben az esetben egy másik is jelentkezik. Az ikonok fekete fehérek, egészen addig, amíg a felhasználó felé nem viszi a kurzort, mert abban az esetben színessé válnak. Ezt a viselkedést a “**filter:** **grayscale(0%)**;” és a **hover effect** kombinálásával értem el. (a “**transition**” értéke megmondja mennyi idő alatt és hogyan menjen végbe a változás; pl: “**transition: filter 0.3s ease**;”)

**Kezdőlap**

A kezdőlapom tartalmát a **home\_styles.css** fájlban stilizáltam. Ezen fájlra az oldal header részében hivatkozok, ugyanakkor, amikor a navigációs fájl stilizálására használt **CSS** fájlra is.

Az oldal tartalma egy **<div></div>** elemben van, melyre “tartalom” -ként lehet hivatkozni. A **<div></div>** (division) egy **HTML** elem, amely egy konténerként működik, és lehetővé teszi a weboldal tartalmának logikai csoportosítását.

Az oldal tetején a projektnek címadó banda neve szerepel (Csaknekedkislány tánczenekar), egy **<h1></h1>** elem belsejében. A **<h1></h1>** egy **HTML** tag, amely a legfontosabb (legnagyobb) címsort jelöli az oldalon. Ezen címsor méretét **CSS** használatával nagyobbra állítottam, hogy hangsúlyosabbnak tűnjön.

Az oldalon található többi szöveg **<p></p>** (paragraph) elemekben található. Ezek színe és elrendezése (sorköz és hasonló) a **home\_styles.css** fájlban van meghatározva.

A szövegben azon szavakhoz melyek fehérek, hivatkozások vannak csatolva:



Azon szavakat, amelyekhez linkeket csatoltam, a “**jeloles**” **class** segítségével jelöltem meg, hogy ezáltal egyszerre tudjak az egészre hivatkozni a stilizálás során.

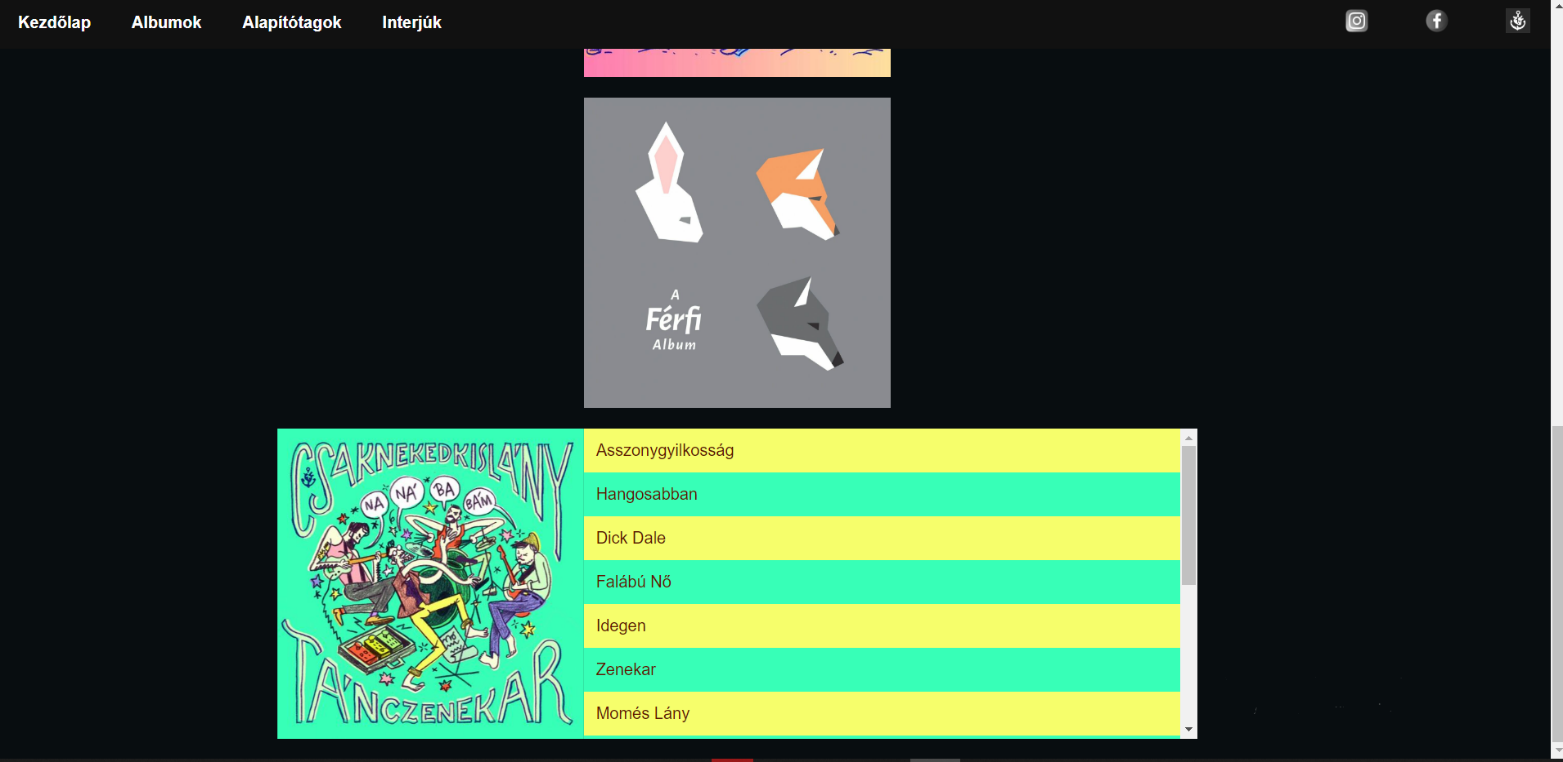
Az oldalon két képet helyeztem el, az. Egyet a konténer aljához igazítva, egyet pedig két bekezdés közé.

A “tartalom” konténer rózsaszín, és a képernyő közepének nagyobb részét fedi le(75%), viszont nem az egészet. Jobb és bal oldalon, továbbá az oldal tetején egy vékony csíkban nem terjed kis a konténer, ehelyett egy másik színt tartalmaz fix háttérként. Ez azt jelenti, hogy azon a területen nem lehet tartalom, és a háttér azon része nem görgethető vagy interaktálható.

**Albumok**

Első látásra, az oldal megnyitásakor egy szürke háttéren a Csaknekedkislány eddig megjelent albumainak borítóit láthatjuk, középen, oszlopba rendezve, egymás alá. Minimalista dizájn tárul elénk. Viszont, ha az egérmutatót az albumborítók felé visszük, az egyes albumborítók kitárulnak, és felfedik ezen oldal igazi érdekességét. Az oldal beépített zenelejátszóját.

Mikor a borító félrecsúszik, elénk tárul az albumon található dalok listája. Az egyes dalokra kattintva, ezek hangja elindul a háttérben. Újra ugyanazon dalra kattintva, a zene megáll. Egy másik dalra kattintva, az a dal indul el.



Az oldal stilizálásáért az **albumstyles.css** fájl felel. Ebben a fájlban meghatároztam a lemezek formáját, elhelyezkedését, oldalra csúszás logikáját, amennyiben a kurzor az albumborító felett található. Az albumok elhelyezésében a flexbox elrendezési modell rengeteget segített.

*Flexbox, vagyis Flexible Box Layout, egy hatékony elrendezési modul a CSS-ben, amelyet arra terveztek, hogy egyszerűsítse az elemek igazítását, elosztását és rendezését egy konténeren belül. Segítségével könnyedén hozhatsz létre rugalmas és dinamikus elrendezéseket, amelyek különböző képernyőméreteken is jól működnek. Flexbox alapjaiban változtatta meg a weboldalak felépítését azáltal, hogy egyszerűbbé tette a középre helyezést, a térközök beállítását, és az elemek méretének automatikus igazítását.*

Az albumok kurzorra való reagálása pedig ugyan azon logika alapján működik, mint a korábban kifejtett navigációssáv jelölései.

A zeneszámok címei linkekként vannak meghatározva (a zeneszámok YouTube linkjei). Ezen linkekre kattintva, egy JavaScript fájl segítségével, melynek neve **albumok.js**, a zeneszámok elindulnak, egy **<iframe></iframe>** elembe beágyazva, melynek szélessége és magassága egyaránt **0px**. Mivel a videólejátszónak gyakorlatilag nincs magassága és szélessége, tulajdonképpen meg is kaptuk lehető legminimalistább zenelejátszót.

Az említett JavaScript lényegében kód 3 függvényből áll, melyek biztosítják a zeneszámok helyes betöltését youtuberól. Ezen három függvény elnevezése és szerepe:

1. **onYouTubeIframeAPIReady**:

* Ez a függvény akkor fut le, amikor a YouTube IFrame API betöltődött és használatra kész.
* Létrehoz egy YouTube lejátszót a weboldalon.
* Beállít két eseménykezelőt:
* onReady: amikor a lejátszó készen áll.
* onStateChange: amikor a lejátszó állapota megváltozik (pl. lejátszás, szünet).

1. **onPlayerStateChange**:

* Figyeli a lejátszó állapotváltozásait.
* Ha a videó lejátszás alatt van, isPlaying értéke true.
* Ha a videó szünetel, isPlaying értéke false.

1. **onPlayerReady:**

* Amikor a lejátszó készen áll, megkeresi az összes .video-link osztályú elemet a weboldalon.
* Minden linkhez hozzáad egy kattintás eseménykezelőt.
* Az eseménykezelő megakadályozza az alapértelmezett link viselkedést és betölti a megfelelő videót a lejátszóba.

**Alapítótagok**

Ez az oldal hasonlóan minimalista kinézetet kapott, mint az előző. Ugyan olyan sötétszürke háttéren kapott helyet a tartalom. A tartalom elrendezése is hasonló az előző oldaléhoz, annyi különbséggel, hogy jelen esetben nem az albumok borítói helyezkednek el oszloposan középen, hanem két tematikus **slideshow**, képekkel a banda alapítótagjairól, valamint pár szó az alapítótagokról, és a banda megalakulásáról.

A slideshow-k kinézetéért, a **tagok\_styles.css** fájl felelős. Ebben a fájlban meghatároztam, hogy a képek maximális mérete mekkora lehet, továbbá a **height:auto** kulcsszavak segítségével megoldottam, hogy a képek ne elnyújtva jelenjenek meg. ennek eredményeképpen a slideshow magassága képenként változik. A slideshow-k tartalmaznak egy sávot a jelenleg kinagyított kép alatt, amelyben látható a többi kép a sorban, kicsiben, enyhén átlátszóan (a tárgyak ezen tulajdonságát az **opacity** értéke határozza meg CSS-ben), a jelenleg kijelölt kép **opacity**-je ismét maximumon lesz, míg a nem kijelölt elemekre a következő értéket állítottam be: **opacity: 0.6**.

A képek váltakozásáért a **tagok\_script.js** fájl felelős. A képek két módszerrel váltakozhatnak:

1. **Automatikusan:** a JavaScript kódban, a **setInterval(autoSlides, 6000);** kódsorral meghatározott időközönként
2. **Manuálisan:** a megjelenített kép alatti sorból rákattintva egy képre. Ezen viselkedés is az adott JavaScript fájlban van meghatározva, mindkét slideshow esetén

**Bibliográfia**

[1] Bela Pelda *Könyv címe/cikk címe* Kiadó, Kiadás éve

[2] Wikipedia Wiki cikk címe *wiki link* [megnézve 2017 február 10. én]

1. What is the Flexbox? – Codefinity: <https://codefinity.com/courses/v2/a5d6ef50-cbf5-4a95-89d4-475b704cd811/ed5e1da4-9290-4a1f-80c1-82b391c19040/c8e9799d-81d5-4688-9181-64f7ef8ef017> [megnézve ..........]