

JEGYZŐKÖNYV

Web Technológiák 1.

Féléves feladat

Wild West Saloon weboldal

Készítette: Tóth Péter

Neptunkód: A24E5Z

Dátum: 2025. november

Miskolc, 2025

Tartalomjegyzék

1. Bevezetés	2
2. Feladat leírása	2
3. Weblap felépítése	2
3.1. Navigációs gombok	2
3.2. Oldalsáv	2
3.3. Kép megjelenítő	3
3.4. Videó kezelő gombok	4
4. Scriptek	5
4.1. Kép megjelenítő	5
4.2. Animáció	6
4.3. Adat-lekérdezés	7
4.4. Json-ből beolvasás	7

1. Bevezetés

A web technológiák napjainkra a digitális kommunikáció és információcsere alapvető pilléreivé váltak. A HTML, CSS, JavaScript és a hozzájuk kapcsolódó könyvtárak, mint a jQuery, olyan szabványos eszközökészletet biztosítanak, amelyek segítségével egységesen, minden eszközön elérhető módon jeleníthető meg tartalom.

2. Feladat leírása

A jelen dokumentum a Wild West Saloon vadnyugattal foglalkozó többoldalas weboldal felépítését, működését és technikai megoldásait mutatja be. A weboldal HTML, CSS, JavaScript, jQuery és JSON technológiákra épül.

3. Weblap felépítése

3.1. Navigációs gombok

Az oldal tetején találhatóak navigációs gombok. A jobb oldali részben találhatóak a html dokumentumok közötti navigáláshoz szükséges gombok ezek a "Kezdőlap", "Legendák", "Fegyverek", "Média" és "Úrlap" gombok, ha rájuk viszed az egeret besárgulnak ezzel jelezve a felhasználónak, hogy kattinthatóak.



1. ábra. A navigáció sáv

3.2. Oldalsáv

Az oldalsávon lévő linkek a könnyebb navigálásért felelősek. minden fejezet egy külön linkkel rendelkezik, ezért azonnal ahhoz tudunk ugrani, amire szükségünk van.



2. ábra. Az oldalsáv

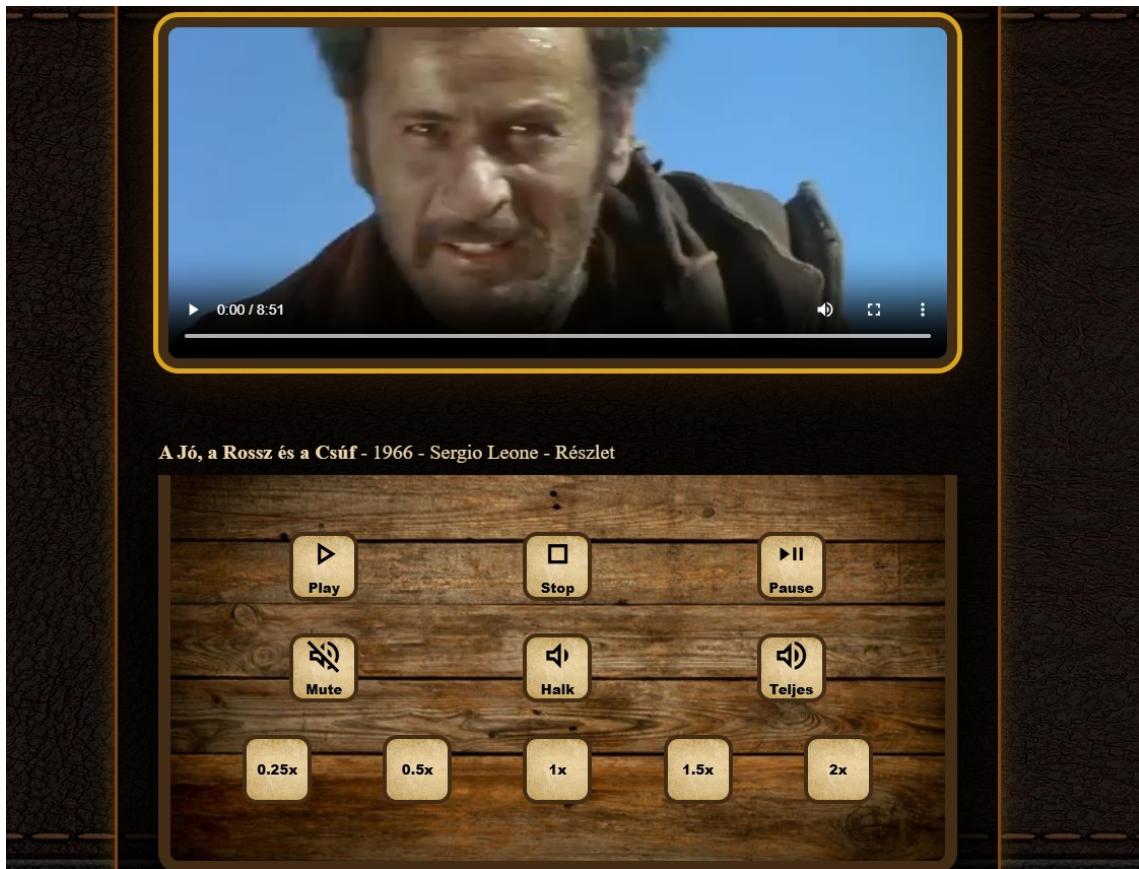
3.3. Kép megjelenítő

Egy kép jelenik meg ha a kurzurt a történetnél lévő névre visszük rá ami oldalt fog megjelenni. A név színe átvált fehérre, jelezve a felhasználónak, hogy ott valami történik.

3. ábra. Kép megjelenítő

3.4. Videó kezelő gombok

Minden videó alatt találhatóak gombok, amelyek a videó alapvető kezelésére szolgál. Az első sorban a videó elindítására, leállítására, illetve elindít/leállít gombok találhatóak. A második sorban a hangsabályzó gombok, az első némi tja a videót lejjebb vesz a hangerőt, a második lejjebb vesz a hangerőt, a harmadik felveszi a maximum hangerőre. A harmadik sorban a videó lejátszási sebessége állítható be. Az öt gomb sorrendben a következő sebességeket kínálja: 0.25x, 0.5x, 1x (normál sebesség), 1.5x és 2x. A kiválasztott sebességre kattintva a videó azonnal az adott tempóban folytatja a lejátszást.



4. ábra. Videó kezelő gombok

4. Scriptek

4.1. Kép megjelenítő

Ez a JavaScript-kód kizárálag 1024 px-nél szélesebb képernyőkön (tehát elsősorban asztali környezetben) fut. Amikor a felhasználó a kurzorral egy szereplő neve fölre viszi az egert (azok az elemek, amelyek a .legend-name osztályt viselik), a szkript megkeresi a hozzá tartozó kártyaelem (.legend-card) legközelebbi szülőjét, majd onnan kinyeri a szereplő képének elérési útját. Ezt a képet azonnal betölti egy előre definiált előnézet konténerbe (az azonosítója legend-hover-preview). Eközben az előnézet elem megkapja az active osztályt, amelynek köszönhetően láthatóvá válik és szép fade-in animációval megjelenik (általában jobb oldalon, fix pozícióban). Amint az egér elhagyja a nevet, az active osztály eltávolításra kerül, így az előnézet elhalványul és eltűnik. Egy további eseményfigyelő a teljes dokumentum mouseleave eseményét is kezeli: ha az egér egyenesen kilép a böngészőablakból, akkor is garantáltan eltűnik az előnézet, elkerülve a „beragadt”

állapotot.

Listing 1. Kép megjelenítés

```
1 if (window.innerWidth > 1024) {
2     const preview = document.getElementById('legend-hover-
3         preview');
4     const previewImg = document.getElementById('hover-preview-
5         img');
6
7     document.querySelectorAll('.legend-name').forEach(name => {
8         name.addEventListener('mouseenter', function() {
9             const card = this.closest('.legend-card');
10            const imgSrc = card.querySelector('.legend-img').src
11            ;
12
13            previewImg.src = imgSrc;
14            preview.classList.add('active');
15        });
16
17        name.addEventListener('mouseleave', function() {
18            preview.classList.remove('active');
19        });
20    });
21
22    document.addEventListener('mouseleave', function(e) {
23        if (!e.relatedTarget && !e.toElement) {
24            preview.classList.remove('active');
25        }
26    });
27}
28
```

4.2. Animáció

képek és bizonyos tartalmi elemek megjelenését látványos, görgetéshez kötött animáció teszi dinamikussá. Ezt a CSS `@keyframes appear` szabály és a modern `scroll-driven animations` (görgetésvezérelt animációk) kombinációja biztosítja:

Listing 2. Animáció

```
1 @keyframes appear {
2     from{
```

```

3     opacity: 0;
4     scale: 0.5
5 }
6 to {
7     opacity: 1;
8     scale: 1;
9 }
10 }
11
12 animation: appear linear;
13 animation-timeline: view();
14 animation-range: entry 0% cover 40%;
```

4.3. Adat-lekérdezés

Ez a program a DomRead-el szemben csak szimplán kiírja az adatokat, hanem emberileg olvasható és feldolgozható formában és mennyiségben írja ki azt.

Listing 3. Javaban rendezett adatkiírás

```

1 System.out.println("3. minden tankolás dátuma:");
2 NodeList tankList = doc.getElementsByTagName(
3     "Tankolas");
4 for (int i = 0; i < tankList.getLength(); i++) {
5     Element t = (Element) tankList.item(i);
6     String datum = t.getElementsByTagName("Datum") .
7         item(0).getTextContent();
8     System.out.println(" - " + datum);
9 }
System.out.println();
```

Ebben a kódrészletben először kinyer a program minden hasznos elementet és attribútumot, majd azt egy kompakt módon olvashatóan kiírja.

4.4. Json-ból beolvasás

A TWWguns.html oldalon a táblázatok adatai dinamikusan, egy külső JSON fájlból kerülnek betöltésre. A program minden egyes fegyver objektumot beolvas, majd annak tulajdonságait egy előre elkészített HTML-es táblázat struktúrába illeszti, ezzel automatikusan felépítve a teljes táblázatot. Így az adatok karbantartása egyszerűen a JSON fájl szerkesztésével megoldható, nincs szükség a HTML kód manuális módosítására.

Listing 4. Json beolvasás

```
1  fetch("TWWguns.json")
2      .then(res => res.json())
3      .then(data => loadWeapons(data));
4
5  function loadWeapons(weapons) {
6
7      // Oldalsó menü generálása
8      const menu = document.getElementById("weapon-menu");
9      menu.innerHTML = weapons.map(w =>
10          '<li><a href="#${w.id}">${w.title}</a></li>‘
11      ).join("");
12
13      // Fegyver blokkok ide kerülnek
14      const container = document.getElementById("weapon-
15          container");
16
17      weapons.forEach(w => {
18
19          // Táblázat sorok
20          const rows = Object.entries(w.data)
21              .map(([key, val]) => ‘<tr><th>${key}</th><td>
22                  ${val}</td></tr>‘)
23              .join("");
24
25          container.innerHTML += ‘
26              <section id="${w.id}" class="weapon-block">
27                  <div class="weapon-img">
28                      
29                  </div>
30                  <div class="weapon-info">
31                      <h2 class="weapon-title">${w.title}</h2>
32                      <table>${rows}</table>
33                  </div>
34              </section>
35          ‘;
36      });
37  }
```

Listing 5. Egy fegyver adatait tároló JSON objektum (részlet)

```
1 {  
2     "id": "colt-saa",  
3     "title": "Colt Single Action Army \"Peacemaker\"",  
4     "img": "kepek/Colt.png",  
5     "data": {  
6         "Típus": "Forgópisztoly (single-action revolver)",  
7         "Úrméret": ".45 Colt",  
8         "Tárkapacitás": "6 lőszer",  
9         "Gyártás": "1873-napjainkig",  
10        "Becenév": "\"The Peacemaker\" - \"Az egyenlítő\""  
11    }  
12 }
```