

Projektmunka

Szücs Gergely

11. D

10.27.

Tartalomjegyzék

Bevezető	3
▶ Feladat ismertetése.....	3
Felhasználói Dokumentáció	3
▶ Követelmények.....	3
○ Támogatott böngészők	3
○ Hardware minimum követelmény	3
▶ Lapok tartalmának és használatának ismertetése	4
Fejlesztői Dokumentáció	6
▶ Főmappa tartalmának részletezése	6
▶ HTML dokumentumok felépítése.....	7
▶ JavaScriptek elmagyarázása	8
▶ Hogyan lett megoldva az órarend táblázat?	12
Hibák, hiányosságok	13

Bevezető

Feladat a következő volt:

Egy weboldal, amely a következő elemeket tartalmazza:

- ↗ Sakktábla
- ↗ Önéletrajz
- ↗ Órarend
- ↗ Képi galéria
- ↗ Videó galéria

Illetve legyen elkészítve egy ezt dokumentáló irat és PowerPoint előadás

A felhasználót elősegítő tudnivalók

A weboldal elérhető a [szgery.site]-domainen.

Követelmények:

⇒ A honlapot támogató böngészők:

- ↗ Google Chrome - (v119.0)
- ↗ Microsoft Edge - (v119.0)
- ↗ Firefox - (v105.0)
- ↗ Opera - (v105.0)
- ↗ Safari - (v17.1)



⇒ Hardware minimum követelmények:

☞ Processzor:

- ↗ Intel Pentium III/M 1.4 GHz
- ↗ Pentium IV 1.4 GHz

☞ RAM:

- ↗ 512MB

☞ 300MB felhasználatlan lemezterület

⇒ Operációs rendszer minimum:

☞ Windows:

- ↗ Microsoft Windows 8 avagy;
- ↗ Microsoft Windows 7

Oldalak ismertetése:

A honlap először megnyitásakor egy töltőképernyő fogad mely 4 másodpercig tart.

⇒ Navbar:

- ☞ Minden oldal tartalmaz egy menüt jobb oldalt, ami által lehet lépkedni az oldalak között, illetve az oldal háttérszínét is be lehet állítani az alján a sötét mód ki- vagy bekapcsolásával.

⇒ Kezdőlap:

- ☞ Az oldal tartalmazza a feladat ismertetésében leírt sakktáblát bábukkal feltöltve (nem működik)
- ☞ Ezen kívül tartalmaz egy visszaszámlálót mely 2024ig számol vissza.

⇒ Órarend:

- ☞ A megjelenítés gombra kattintva elő hív egy táblázatot mely tartalmazza az órarendet.
 - ☞ A tanórákra ráhúzva az egeret feltűnik a tanterem hol az óra tartva van, kattintásra pedig szaktanárt jeleníti meg.
- ☞ A megjelenítésre ismét rányomva eltünteti a táblát.

⇒ Önéletrajz:

- ☞ A levélikonra kattintva megjeleníti a teljes önéletrajzot.
- ☞ Az önéletrajz aljában van egy x gomb, ami bezárja a dokumentumot, és ismét a levél jelenik meg.
- ☞ Az előbb említett x gombtól jobbra található egy Download feliratú gomb, melynek megnyomásával egy PDF változatát tölti le az életrajznak.



⇒ Kép galéria

- ☞ A kép galéria 6 képből áll, kik a 2 világháborúban vezetők voltak, Sztálint leszámítva
- ☞ A „kártyákra” ráhúzva az egeret megtekinthető a személy teljes neve, születésének dátuma, jogköre, hivatali ideje.
 - ☞ Szövegre rákattintva pedig a Wikipédia oldalát hozza elő új lapon.
- ☞ Ha a honlap háttérzenéje esetleg zavar, akkor azt lenémíthatod a jobb felső sarokban.





⇒ Videó galéria

- ☞ 4 videó közül lehet válogatni, váltani az indexképre nyomva lehet.
- ☞ A videót megállítani úgy lehet, ha a videóra rányom a felhasználó vagy alul a [▶] gombra kattintva.



- ☞ A videót teljes képernyőbe a jobb oldali „  ” gombra nyomva lehet kitenni.
- ☞ A videó hangerejét változtatni a „” gombra nyomra lehet, e közül van 3 mód;
 - ☞ egy maximum hangerejű, ez az alap,
 - ☞ egy teljesen némított,
 - ☞ egy 50% hangerejű.

⇒ Tetris:

- ☞ Egy Tetris játék, klasszikus szabályokkal,
 - ☞ Jobb [] és bal [] nyíllal mozgatandó az objektum horizontálisan,
 - ☞ Lefele [] nyíllal lenyomni tudja.
 - ☞ Felfele [] nyíllal az alakzatot lehet forgatni.

⇒ Dokumentáció:

- ☞ Két gomb található;
 - ☞ A PDF feliratú a dokumentációt nyitja meg PDF formában,
 - ☞ A WORD feliratú pedig Google Docs -ból nyitja meg.

Fejlesztői dokumentáció:

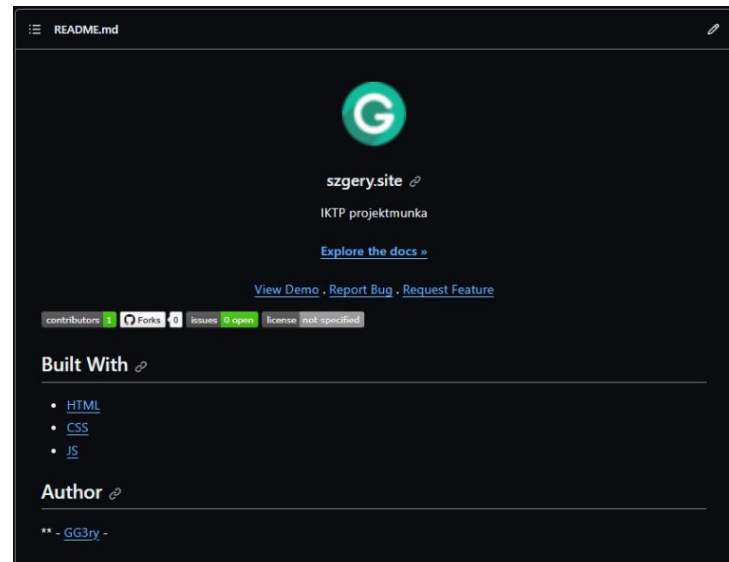
A honlap minden fájlja megtalálható Github Repositoryként, [itt](#). Amik a weblapon megtalálható kép / videó / hang tartalmak megtalálhatóak a ['src'](#) mappában. Ezen belül is rendszerezve vannak a fájlok 4 mappában ezek a következők;

- ['Cursors'](#) ➔ A honlapon használt kurzor variációk találhatóak meg, ennek 3 variációja a 'default', 'active', 'click', és ezeknek 2 fajta megjelenése van, függően attól, hogy éppen aktív-e az éjszakai funkció.
- ['Gallery'](#) ➔ A képi galériában felhasznált képek találhatóak meg itt.
- ['Thumbnails'](#) ➔ A videó galériában felhasznált indexképeket tartalmazza.
- ['Video'](#) ➔ A videó galériában szereplő mozgóképek fellelhetők eme mappában.

A többi mappáról / fájlról röviden:

- ['html'](#) ➔ Az oldalak „csontszerkezetéért” felelős HTML fájlokat tartalmazza.
- ['scripts'](#) ➔ A HTML csontszerkezetet kisegítő JavaScript fájlok találhatóak meg itt.
- ['styles'](#) ➔ A HTML megjelenését, formázását segítő CSS fájlok szerepelnek itten.
- ['index.html'](#) ➔ Eme fájlt tölti be legelőször a böngésző mikor a webcímre rákeres. Ehhez tartozó ['loading.js'](#) az ami segíti elő azt, hogy 4 másodperccel a betöltése után át legyen irányítva.
- ['CNAME'](#) ➔ A weboldal - Domain-cím kapcsolatfenntartását elősegítő definiálatlan fájl.

→ `'README.md'` →→
 Eme leírás megjelenéséért felelős fájl, a Github megjelenéshez kapcsolódik, nem részletezném.



A HTML-ek (az index.html-t leszámítva) és `menu.css` szerkezete általánosságban:

A HTML-k két részre szedhetőek, egyik a `<nav>` tag a másik pedig a `<main>` tag. A `<nav>` mondhatni konstans, azaz szinte állandó, azt leszámítva, hogy mi a tartalma a legfelső szövegnek, ami az adott oldal címe, megnevezése. Ezzel szemben a `<main>` tag minden esetben változik, ez tartalmazza az adott oldal tartalmát.

A `menu.css` az általános, mindegyik HTML oldalra vonatkozó stíluslap. Ez kitér a nav bar szerkezetére, illetve a main tartalmának általános jellemzőire, a sötét-világos módot is tartalmazza, illetve a kurzorok változását, s a nem létező rezponzivitásért is felelős.

A fájl által definiált első dolog az a „:root”, mely a html betöltésekor tölt be, ebben vannak definiálva a css változóinak alap értéke. A változók többsége színek, ezek közül 2 a nav háttérhez kapcsolódó színek, 1 a main háttér színe, 2 szövegszín. Ezen

```
:root {
  font-size: 16px;
  font-family: 'Open Sans';
  --text-primary: #b6b6b6;
  --text-secondary: #ecec;
  --bg-primary: #23232e;
  --bg-secondary: #141418;
  --bg-third: #3d3d3d;
  --transition-speed: 1000ms;
  --cursor-def: url('/src/cursors/cursor_white.png');
  --cursor-active: url('/src/cursors/cursor_active_white.png');
  --cursor-click: url('/src/cursors/cursor_click_white.png');
}
```

kívül van 3 kurzorváltozó melyek képeket tartalmaznak. Illetve a található még egy, ez a menü animációját befolyásoló változó, milyen gyorsan változzanak át a színek a <nav> -ban és a <main> -ben. Ezen felül még mindig ebben a fájlban megtalálható a jobb oldalt megjelenő „görgőszáv”. Ez csak annyiban van módosítva, hogy a szélessége változott, mint a böngésző által használt alap, illetve ennek a színe lett megváltoztatva narancssárgára. Ezen kívül még részletezve van a nav bar különböző elemei, a nav bar kinyitása, a világos, sötét mód változóinak értéke.

Ehhez szorosan kapcsolódik a [bg.js](#). Ez az, ami azért felel, hogy változzon a változók értéke s azért, hogy melyik kép jelenjen meg a nav bar alján (hold vagy nap), értse úgy, hogyha rányom a Sötét mód gombra, akkor az eddigi sötét háttérszín változóját kicseréli oly módon, hogy a „light” osztály változóinak értéke legyen aktív ameddig még egyszer le nem lesz nyomva.

További JavaScript fájlok a weboldalon:

➔ [cd.js](#) → A kezdőlapen megjelent visszaszámlálóért felelős. Ez visszaszámol újév előtt lévő egy másodpercig, ha a dátum már a múltban található írja ki azt, hogy „LEJÁRT DÁTUM”

➔ [letter.js](#) → Annyiért felelős, hogyha a „letter” azonosítóval rendelkező tag látható és rányomunk. tüntesse el és a helyette az önéletrajzot jelenítse meg, majd ha az x re nyomunk, akkor tüntesse el a megjelent dokumentumot és rakja vissza helyére az animált levelet.

```
scripts > bg.js > ...
1  const themeMap = {
2    dark: "light",
3    light: "dark",
4  };
5
6  const theme = localStorage.getItem('theme')
7    || (tmp = Object.keys(themeMap)[0],
8      localStorage.setItem('theme', tmp),
9      tmp);
10 const bodyClass = document.body.classList;
11 bodyClass.add(theme);
12
13 function toggleTheme() {
14   const current = localStorage.getItem('theme');
15   const next = themeMap[current];
16
17   bodyClass.replace(current, next);
18   localStorage.setItem('theme', next);
19 }
20
21 document.getElementById('themeButton').onclick = toggleTheme;
22
```

```
letter.js > onoff
function onoff() {
  var x = document.getElementById("letter");
  var y = document.getElementById("doc");
  if (x.style.display === "none") {
    x.style.display = "block";
  }
}

1  function megjelenes() {
2    var x = document.getElementById("hidden");
3    if (x.style.display === "none") {
4      x.style.display = "block";
5    } else {
6      x.style.display = "none";
7    }
8  }
```


→ `'orarend.js'` → Az előzőhöz hasonló módon megjeleníti és eltünteti az órarend táblázatát.

→ `'video.js'` → A videó `<source>` tagen belül található forrást cseréli ki, annak megfelelően melyik indexképre nyom a felhasználó.

Ennek értéke úgy van megadva, hogy „inline” megvan adva az

```
function videoselector (videoLink) {
    document.getElementById('video').src=videoLink;
```

„út”/ „path” a videó indexképéhez tartozó videó fájlhoz. Példa kedvéért így néz ki egy videó indexképe:

```
<li onclick="videoselector('/src/video/video1.mp4')">
  
</li>
```

Látszik, hogy paraméterként van megadva az adott videó útja
A videó megállítása és újbóli lejátszása a következőképpen van megoldva:

Itt az `y` változó azért felelős, hogy érzékelje, a videó megvan-e állítva vagy nem, ha a felhasználó megnyomja a gombot vagy a videóra nyom, akkor 2 lehetőség van:

➔ Az `y` értéke 0 tehát a videó mozog, ez esetben a videó megáll a felhasználó utasítására, és kicseréli a gomb képét.

➔ Az `y` értéke 1 tehát a videó meg van állítva, a videó megáll a felhasználó kérése szerint s kicseréli a gomb képét, ahogy eredetileg is volt.

Ehhez hasonlóan néz ki a videó hangerejének szabályozása, a különbség annyi, hogy 3 variáció van; (a „v” változó alap értéke 2)

```
let y = 0;
function pl() {
    vid = document.getElementById('video')
    playbutton = document.getElementById('play');
    pausebutton = document.getElementById('pause');
    if (y==0) {
        vid.pause();
        playbutton.style.display = 'none';
        pausebutton.style.display = 'block';
        y = 1;
    }
    else if (y==1) {
        vid.play();
        playbutton.style.display = 'block';
        pausebutton.style.display = 'none';
        y = 0;
    }
}
```

- A v értéke 2 tehát a videó maximális hangerőn van, ez esetben a magas hangerő képe kicserélődik a némított képére, illetve a videót lenémítja, a változó értéke 0 lesz.
- A v értéke 0 tehát a videó némított, a gomb ekkori megnyomása feltételében teljesül az, hogy a némított hangszóró képén kicseréli egy közepes hangerejűre, a hangerőt szabályozza 0.5re, azaz a maximális felére, ezen felül a változó értéke 1 lesz.
- Az egyetlen hátramaradott az, ha a változó az 1-es értékkel rendelkezik, ilyenkor a közepes hangerő képét cseréli le a maximum hangerő képére, a változó értéke 2 lesz, a videó hangereje 1, azaz a maximum.

Pontosan ugyanezen az alapelven működik a képi galéria háttérzenéjének kontrolálása is.

Visszatérve a videó galériához, van egy funkció, ami a felhasználói dokumentációban említve volt, a fejlesztőiben pedig nem, ez pedig a mozgókép teljes képernyőre tétele. Ez a következőképpen lehetséges:

Az if tartalma semmi para, az egész annyi, hogy kompatibilitás legyen safari és Firefox böngészőkkel is. Ugyanez a függvény van meghívva duplakattintás esetén a dblclick használatával.

```
function vol() {
    vid = document.getElementById('video');
    high = document.getElementById('high');
    muted = document.getElementById('muted');
    medium = document.getElementById('medium');

    if (v==2) {
        high.style.display = 'none';
        muted.style.display = 'block';
        v = 0;
        vid.volume = 0;
    }
    else if (v==0) {
        muted.style.display = 'none';
        medium.style.display = 'block';
        v = 1;
        vid.volume = 0.5;
    }
    else {
        medium.style.display = 'none';
        high.style.display = 'block';
        v = 2;
        vid.volume = 1;
    }
}
```

```
function fs() {
    vid = document.getElementById('video');
    if (vid.requestFullscreen) {
        vid.requestFullscreen();
    } else if (vid.webkitRequestFullscreen) {
        vi.webkitRequestFullscreen();
    } else if (vid.msRequestFullscreen) {
        vid.msRequestFullscreen();
    }
}
```

A billentyűs vezérléshez a következő függvény segít elmenteni rövid memóriába a lenyomott billentyűket a billentyu nevű függvényt hívja meg lenyomás esetén. Ez a függvény egy hosszú egybefüggő if és elseifekből áll.

Például az [M] lenyomása esetén a függvény a következő részt lépteti életbe: Röviden ez annyit csinál, hogy megnézi milyen erősségű jelenleg a hangerő, s ha nincsen lenémítva akkor elmenti a jelenlegi hangerőt egy másik változóba, majd lenémítja a videót, ha leván némítva a videó akkor az "a" változó értéke alapján cselekszik.

A space lenyomása esetén van a legegyszerűbb feladat ugyanis csak meghív egy már korábban bemutatott függvényt mely a pl() függvény.

A fel [↑] és lenyíl [↓] csak annyit csinál hogy a hangerőhöz hozzáad vagy éppenséggel elvesz.

Az előre és hártamenet pedig így lett megoldva: Tulajdonképpen csak hozzáad a videó idejéhez 5 másodpercet vagy elvesz annyit.

```
67 let a = 2;
68 document.addEventListener('DOMContentLoaded', function() {
69     document.addEventListener('keydown', billentyu);
70 });
```

```
if (k.code === 'KeyM') {
    if (v !== 0) {
        a = v;
        v=0;
        vid.volume = 0;
        muted.style.display = 'block';
        high.style.display = 'none';
        medium.style.display = 'none';
    }
    else if (a == 1) {
        v = a;
        vid.volume = 0.3;
        muted.style.display = 'none';
        high.style.display = 'none';
        medium.style.display = 'block';
    }
    else if (a == 2) {
        v = a;
        vid.volume = 1;
        high.style.display = 'block';
        muted.style.display = 'none';
        medium.style.display = 'none';
    }
}
```

```
else if (k.code === 'Space' || k.key === ' ') {
    pl();
}
```

```
else if (k.code === 'ArrowRight') {
    vid.currentTime += 5;
}
else if (k.code === 'ArrowLeft') {
    vid.currentTime -= 5;
}
```

Hogy hogyan lett az megoldva hogy az órarendben mást jelenítsen meg, ha ráhúzza az egeret vagy, ha kattint?

Nos ehhez nem kell egy csepp JavaScript sem, a felépítése a következő egy

Alapvetően e 3 közül csakis az ora osztályjelzésű felirat jelenik meg. Ha azonban ráhúzzuk az egeret a következő lép érvénybe:

Erősen hasonló eset áll fent a rányomás esetében:

```
<td> <span class="ora"> Magyar nyelvtan</span>  
| <span class="terem"> 14 -es terem </span>  
<span class="tanar">Balogh Péter</span></td>
```

cellának:

```
td:hover .terem {  
    display: inline;  
}  
td:hover .ora {  
    display: none;  
}
```

```
td:active .terem, td:active .ora {  
    display: none;  
}  
td:active .tanar {  
    display: inline;  
}
```

Ismert hibák / hiányosságok:

Lassú válaszidő:

- ↪ A honlapra jellemző az, hogy lassú, amennyiben a gép csak a minimum követelményeknek felel meg.

A videó galéria lassú betöltése:

- ↪ Ennek az oka az, hogy a betöltendő videók nem egy nagy cég (pl. Youtube) szerverein tárolt és onnan beágyazott, hanem az oldal fájljai közé van elmentve mp4 formátumban. E probléma minden eszközön való megjelenésre jellemző.

~~Dokumentáció oldalon nem reagál első lenyomásra a gomb~~

Reszponzivitás részleges hiánya:

- ↪ Az oldalak telefonos nézete hanyagolt, nem tanácsos mobilkészülékről megtekinteni. Valamilyen szinten működik a mobilnézet, ám vannak olyan aloldalak melyekkel nem megfelelően működik.

Tetris hibái:

- ↪ Amennyiben a felhasználó megállítja, „lepauseolja” és újra elindítja, akkor új alakzat generálódik a következőként eső alak helyett.
- ↪ Amennyiben a játék véget ér az irányítás nem áll meg, értsé úgy, hogy az alakzatok esése, illetve dőlése még mindig irányítható.

A Tetris hibáit kereső munkatársam: Szücs Sándor.

