

JEGYZŐKÖNYV

Webkönyvtárak

Webprogramozás

Kedvenc sportcsapataim

Készítette: **Laczkó Lajos**

Neptunkód: **W6Y0XZ**

Dátum: 2024. május 10.

Sárospatak, 2024

Tartalomjegyzék

Bevezetés.....	3
1. Feladat: HTML.....	4
1.1. Legalább öt HTML fájl legyen.....	4
1.2. HTML5 szerkezet megtervezése (kép - oldal tetején, menü stb).....	4
1.3. HTML elemek: div, span, p, címsorok, képek, táblázat, linkek, html5 elemei stb.....	6
1.4. Űrlap elemek: szöveges beviteli mező: egy, több soros, adatlista, jelölőnégyzet, rádiógomb, színválasztó, dátumválasztás, gombok, számozott lista.....	7
1.5. Video: az adott témakörhöz tartozó rövid video elkészítése, egyes gombok vezérlése JavaScript kóddal.....	10
2. CSS.....	11
2.1. Style attribútumban néhány elem formázása.....	11
2.2. Azonosító alapján formázást.....	11
2.3. Osztály alapján formázást.....	12
2.4. Táblázat formázása.....	12
2.5. Menü kialakítása.....	12
2.6. Háttérszín.....	14
2.7. Linkek formázása.....	14
2.8. Űrlap elemek, gombok formázása.....	14
2.9. Egy vagy több .CSS fájlban is legyen formázás, és a html-ben a <style> tag-ben is.....	16
3. JavaScript, jQuery, JSON, AJAX.....	16
3.1. form ellenőrzés, hiba esetén az egyes űrlap elemeknél kiírja, hogy mi a hiba, css-t állít pl. piros keret, ha hibás az adott űrlap elem.....	16
3.2. jQuery animáció.....	17
3.3. elemek kiválasztása: html tag név alapján, osztály alapján, azonosító (id) alapján stb..	17
3.4. új html elem készítése, meglévő html elem módosítása.....	18
3.5. Egy fájl elkészítése JSON formátumba (beágyazott).....	18
3.6. JavaScript, jQuery – AJAX metódus használatával JSON fájl megjelenítése a weboldalon.....	18
Szerver.....	19
Adatbázis.....	20
Futtatás eredményei:.....	21
Források:.....	21

Bevezetés

A Webkönyvtárak tárgyhoz én Laczkó Lajos a “Kedvenc sportcsapataim” témát választottam. Ez a téma kütkösképpen közel áll hozzám ugyanis nagy sportrajongó hírében állok és ennek fényében feltételeztem, hogy nem lesz hiányom tartalmi és szerkezeti ötletekből, ezért is választottam ezt a témát. Ennek a témának megvalósítására HTML fájlokat használtam JavaScript kóddal és CSS dizájnnal ellátva. Ezek mellett még használtam a NodeJs keretrendszert és a XAMPP webservert szoftvercsomagot a localhost-on való futtatáshoz, valamint a MySQL adatbázis kapcsolathoz. A weblapok animációit JQuery segítségével oldottam meg illetve néhol előfordult, hogy CSS effekteket használtam.

A terv az volt, hogy létrehozzak egy olyan projektet, amely kellően bemutatja az általam preferált csapatokat (Real Madrid, Golden State Warriors, Mercedes AMG PETRONAS) és az adott sportokról (labdarúgás, kosárlabda, formula-1) nyújtson egy kisebb ismeretterjesztést. Ezen terv megvalósítása közben minden tudásom latba vetve igyekeztem eleget tenni a vizsga kiírásban lefektetett követelményeknek. Szerény véleményem szerint ennek eleget tettem de ezt majd az alábbiakban bővebben kifejtem.

A tartalmakat (szövegek, képek, videók, játékos adatok) több forrásból gyűjtöttem ezek egy része meg van jelölve az adott tartalomnál egy “<a>” taggel ellátva de a részletesebb forrásmegjelölés megtalálható ebben a dokumentumban.

Maga a projekt meglehetősen grandiózusra sikerült, legalább is számomra. Tartalmaz hat darab HTML oldalt, hét darab JavaScript fájlt (amennyiben nem számoljuk a “server.js” fájlt) , egy JSON fájlt és hét darab CSS fájlt. Ezeken felül még elláttam három videóval és megannyi képpel.

1. Feladat: HTML

1.1. Legalább öt HTML fájl legyen

```
<> f1-mercedes-page.html
<> golden-state-page.html
<> home-page.html
<> kerdoiv.html
<> player-details.html
<> real-madrid-page.html
```

Minden HTML-fájlom megtalálható a következő elérési úton:

WebProg/public/.

A HTML oldalakra külön hivatkozom localhoston keresztül, mivel a server.js ezen linkek segítségével valósítja meg a futtatást.

Ezen linkek a következők:

- [“http://localhost:3000/”](http://localhost:3000/) - **home-page.html**
- [“http://localhost:3000/real-madrid”](http://localhost:3000/real-madrid) - **real-madrid-page.html**
- [“http://localhost:3000/golden-state”](http://localhost:3000/golden-state) - **golden-state-page.html**
- [“http://localhost:3000/mercedes”](http://localhost:3000/mercedes) - **f1-mercedes-page.html**
- [“http://localhost:3000/kerdoiv”](http://localhost:3000/kerdoiv) - **kerdoiv.html**
- [“http://localhost:3000/player-details.html?player=player_name”](http://localhost:3000/player-details.html?player=player_name) - **player_details.html**

1.2. HTML5 szerkezet megtervezése (kép - oldal tetején, menü stb).

A html szerkezetét legjobb tudásomnak, remélhetőleg a követelményeknek is megfelelően elkészítettem. Az alábbiakban pár példán keresztül ezt be is mutatom.

Fej:

```
<!DOCTYPE html>
<html lang="hu">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
    <script src="/jquery/jquery-3.7.1.js"></script>
    <script src="/js/sidebar.js"></script>
    <script src="/js/home-page.js"></script>
    <script src="/js/background-color.js"></script>
    <link rel="stylesheet" href="/css-fajlok/sidebar.css">
    <link rel="stylesheet" href="/css-fajlok/home-page.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Kanit:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&displ
```

```
ay=swap" rel="stylesheet">
</head>
```

Ez a kód a home-page.html-em teteje. Ahogy a kódon látszódik a “<!DOCTYPE html>” tag jelzi a böngészőnek, hogy HTML a fájl formátuma. Alatta a “<html>”-tagen belül jelöljük a dokumentum nyelvét lang=”hu”. A “<head>” részben van eltárolva a különböző adatok deklarálása például a karakterkészlet “<meta charset=’UTF-8’>” illetve ide van elhelyezve a “<title>” amely a böngészőben megjelenő nevet szolgáltatja. Végül pedig még ide helyezzük el a különvéle hivatkozásokat, mind scriptekre, mind stílus fájlokra.

Fejléc:

```
<div id="header">
  
</div>
```

Ez a kódrészlet az “f1-mercedes-page.html” oldalam fejléce ahol, mint az oldalaim többségénél egy logó kép jelenik meg. Ezt a logót az oldal közepére igazítottam az oldal tetejére. Ezen stílusok kódja megtalálható a “public/css-fajlok/mercedes.css” fájlban.

Menü:

```
<div class="sidebar">
  <ul>
    <li><a href="http://localhost:3000/">Főoldal</a></li>
    <li><a href="http://localhost:3000/real-madrid">Real Madrid</a></li>
    <li><a href="http://localhost:3000/golden-state">Golden State Warriors</a></li>
    <li><a href="http://localhost:3000/mercedes">Mercedes-AMG PETRONAS</a></li>
    <li><a href="http://localhost:3000/kerdoiv">Kérdőív</a></li>
  </ul>
</div>
```

Ez a kód egy navigációs menü, amely a HTML oldalak közötti “közlekedésért” felel. Mindezt “<a>” tagekkel teszi. A “sidebar” osztályon keresztül végeztem ennek a menünek a dizájnolását a sidebar.css fájlban.

Kép:

```
<div id="sports">
  
  
  
</div>
```

A “<div>”-tag egy konténer, amelyet a könnyebb formázhatóság miatt hoztam létre és ennek köszönhetően gond nélkül egymás mellé helyezhettem

a képeket. Minden kép “” el van látva egy “sport-img” osztállyal, amin keresztül meghatároztam azok stílusát (width, border, stb..)

Lábléc:

```
<footer>
  <p class="kanit-regular">Készítette: Laczkó Lajos Neptunkód: W6Y0XZ</p>
</footer>
```

A lábléc (footer) a html oldal alján helyezkedik el. Itt megjelenítettem a nevem mint készítő illetve a neptun kódomat egy “<p>” szöveg tag segítségével. Ennek szövegnek formázását hivatott a “kanit-regulat” osztály elvégezni.

1.3. HTML elemek: div, span, p, címsorok, képek, táblázat, linkek, html5 elemei stb

A felsorolt elemek közül mindet felhasználtam, még többet is. Az alábbiakban felsorolok ezek közül néhányat, amelyeket még nem mutattam be a fentiekben.

- **span:** `×`
- **címsorok:** `<h1 class="right-content">A MERCEDES </h1>`
`<h2>Székhelyek</h2>`
`<h3>Lewis Hamilton</h3>`
- **táblázat:**

```
<table id="jatekosok-table">
  <thead>
    <tr>
      <th>Név</th>
      <th>Pozíció</th>
      <th>Születési Dátum</th>
      <th>Nemzetiség</th>
    </tr>
  </thead>
  <tbody></tbody>
</table>
```

Ez a táblázat azért üres, mivel egy JavaScript fájl tölti fel a játékosok adataival, amely egy json fájlból kapja ezen adatokat.

- Ezen elemeken felül használtam még szekciókat (section) elválasztó vonalat (hr) számozott és számozatlan listákat (ul, ol) illetve űrlap elemeket de azt a következő pontban be is mutatom.

1.4. Űrlap elemek: szöveges beviteli mező: egy, több soros, adatlista, jelölőnégyzet, rádiógomb, színválasztó, dátumválasztás, gombok, számozott lista.

Az űrlapot a *“WebProg/public/kerdoiv.html”* oldalon valósítottam meg, amely különböző adatokat kér a felhasználótól majd a *“Beküld”* gomb megnyomására a kérdőív alá kiírja a beírt adatokat.

Űrlap felépítése:

- **Név:** Egy egysoros szöveges beviteli mező, amely csak a megengedett karaktereket engedi beírni, kitöltése kötelező, automatikus kitöltése nem engedélyezett és az oldal betöltésekor automatikus fókuszba kerül.
- **Életkor:** Ez egy szám típusú beviteli mező, amelynek megvan határozva a legkisebb értéke valamint lépésköze 1.
- **Van kedvenc sportcsapatod?:** Ennek a kérdésnek a megválaszolására két rádiógomb felel (Igen-Nem). Az *“Igen”* értékű gomb *“checked”* azaz automatikusan az van bejelölve a lap betöltésekor. Ez persze változtatható és amint a másik gombra kattint a felhasználó az *“Igen”* értékű gomb nem lesz bejelölve.
- **Kedvenc sportod:** Ez szintén egy egysoros szöveges beviteli mező, de viszont ez egy adatlistából kéri le a kijelölt értéket.
- **Az alábbi csapatok közül követed valamelyiket?:** A kérdés megválaszolására fel van sorolva három darab sportcsapat, valamint mind mellett van egy jelölőnégyzet típusú beviteli mező.
- **Nagyjából mióta követed a csapatot?:** Egy dátum típusú beviteli mezővel adhat választ a felhasználó a kérdésre.
- **Mi a véleményed a csapatról/csapatokról?:** Ez az utolsó kérdés az űrlapról, amelynél egy többsoros beviteli mező (textarea) segíti a felhasználót a válaszadásban. A beviteli mező öt sor és 50 oszlop nagyságú.
- **Beküld gomb:** Az űrlap alján elhelyezett gomb felel azért, hogy az űrlap tartalma ellenőrzésre kerüljön a *“WebProg/public/js/kerdoiv.js”* fájlban, valamint ha minden megfelelő ezen adatok térjenek vissza és íródnak ki az oldalra.

Űrlap kód:

```
<form id="urlap" name="urlap">
  <b>Név:</b> <input id="name" type="text" pattern="[A-Za-z]{4,30}" required autocomplete="off" name="name"
autofocus placeholder="Írd be a neved"> <br><br>
  <b>Életkor:</b> <input id="age" name="age" type="number" min="0" step="1" placeholder="Írd be az
életkorod"></input> <br><br>
  <b>Van kedvenc sportcsapatod?</b> <br>
  <input type="radio" id="yes" name="yesno" class="yesno" value="Igen" checked>
  <label for="yesno">Igen</label><br>
  <input type="radio" id="no" name="yesno" class="yesno" value="Nem">
  <label for="yesno">Nem</label><br><br>
  <b>Kedvenc sportod:</b> <input id="fav-sport" name="fav-sport" type="text" list="kedvenc-sport" > <br><br>
  <datalist id="kedvenc-sport">
    <option value="Labdarúgás"></option>
    <option value="Kosárlabda"></option>
    <option value="Formula-1"></option>
    <option value="Egyéb"></option>
  </datalist>
  <b>Az alábbi csapatok közül követed valamelyiket?</b><br>
  Real Madrid <input type="checkbox" id="rm" class="teams" value="Real Madrid" ><br>
  Golden State Warriors <input type="checkbox" id="gs" class="teams" value="Golden State Warriors" ><br>
  Mercedes-AMG PETRONAS <input type="checkbox" id="f1m" class="teams" value="Mercedes-AMG PETRONAS" ><br><br>
  <b>Nagyjából mióta követed a csapatot?</b> <input type="date" id="date" > <br><br>
  <b>Mi a véleményed a csapatról/csapatokról?</b> <br>
  <textarea name="opinion" id="opinion" rows="5" cols="50" ></textarea>
  <br>
  <button id="bekuld">Beküld</button>
</form>
```

A feladat részét képezi, még egy számozott lista. Ezt egy másik oldalon valósítom meg, név szerint *“WebProg/public/golden-state-page.html”*. Itt felsorolom rangsor szerint az NBA kosárlabda liga történelmének top 10 csapat összeállítását. Minden egyes adat link-ek között van elhelyezve, amely az adott csapat összeállítás Wikipédia oldalára kalauzolja el a felhasználót.

Számozott lista kód:

```
<ol>
  <li><a href="https://en.wikipedia.org/wiki/1995-96_Chicago_Bulls_season" target="_blank">1995-96 Chicago Bulls</a></li>
  <li><a href="https://en.wikipedia.org/wiki/2016-17_Golden_State_Warriors_season" target="_blank">2016-17 Golden State Warriors</a></li>
  <li><a href="https://en.wikipedia.org/wiki/1971-72_Los_Angeles_Lakers_season" target="_blank">1971-72 Los Angeles Lakers</a></li>
  <li><a href="https://en.wikipedia.org/wiki/1970-71_Milwaukee_Bucks_season" target="_blank">1970-71 Milwaukee Bucks</a></li>
  <li><a href="https://en.wikipedia.org/wiki/1985-86_Boston_Celtics_season" target="_blank">1985-86 Boston Celtics</a></li>
  <li><a href="https://en.wikipedia.org/wiki/1990-91_Chicago_Bulls_season" target="_blank">1990-91 Chicago Bulls</a></li>
  <li><a href="https://en.wikipedia.org/wiki/1986-87_Los_Angeles_Lakers_season" target="_blank">1986-87 Los Angeles Lakers</a></li>
  <li><a href="https://en.wikipedia.org/wiki/2000-01_Los_Angeles_Lakers_season" target="_blank">2000-01 Los Angeles Lakers</a></li>
  <li><a href="https://en.wikipedia.org/wiki/1982-83_Philadelphia_76ers_season" target="_blank">1982-83 Philadelphia 76ers</a></li>
  <li><a href="https://en.wikipedia.org/wiki/2012-13_Miami_Heat_season" target="_blank">2012-13 Miami Heat</a></li>
</ol>
```

Szintén a feladat része egy színválasztó. Én ezt minden oldalnál felhasználtam, ugyanis ez egy olyan funkciót tölt be a projektben, ami egy kis szabad kezdet ad a felhasználónak abban, hogy saját maga változtathassa az adott oldal háttérszínét. Ezt egy JavaScript kóddal végzi, amely a kiválasztott értékre változtatja a háttérszín (background-color) értékét az oldal test részének (body).

Színválasztó kód:

```
<form id="setBackgroundColor">
  <input type="color" id="colorPicker" name="colorPicker" value="#42963c">
</form>
```

1.5. Video: az adott témakörhöz tartozó rövid video elkészítése, egyes gombok vezérlése JavaScript kóddal.

A projekt elkészítése alatt három videót használtam fel, melyek elérhetősége megtalálható a dokumentum alján. Ezen videók megtalálhatók a *“WebProg/public/videos”* elérési úton a projektben. A videókat a főoldalon használtam fel ahol fel van sorolva három sport (labdarúgás, kosárlabda, formula-1) és az adott sportra kattintva betöltődik annak ismertető szövege, az általam preferált csapat adatai illetve a csapatról egy montázs videó. A videók alá elhelyeztem három vezérlő gombot. Ezen gombok a *“Némítás/Hangadás”*, *“Indítás/Megállítás”* valamint a *“Gyorsítás/Visszaállítás”* gombok. A gombok vezérlését a *“WebProg/public/js/home-page.js”* fájlban valósítottam meg. Továbbá ezen fájlban hozom létre mind a gombokat illetve a videókat.

Video kód:

```
$(".current_sport").append("<video width='100%' controls> <source src='" + data.video +  
"'" type='video/mp4'> </video>");
```

Vezérlő gombok kód:

```
// Némítás/Hangadás gomb  
$("#main").on("click", ".mute-unmute", function () {  
    let video = $(".current_sport video")[0];  
    if (video.muted) {  
        video.muted = false;  
    } else {  
        video.muted = true;  
    }  
});  
  
// Indítás/Megállítás gomb  
$("#main").on("click", ".start-stop", function () {  
    let video = $(".current_sport video")[0];  
    if (video.paused) {  
        video.play();  
    } else {  
        video.pause();  
    }  
});  
  
// Gyorsítás/Visszaállítás gomb  
$("#main").on("click", ".speedup-slowdown", function () {  
    let video = $(".current_sport video")[0];  
    if (video.playbackRate == 1) {  
        video.playbackRate = 2;  
    } else {  
        video.playbackRate = 1;  
    }  
});
```

Az adott gombra kattintva a kód létrehoz egy *“video”* változót illetve rávizsgál a videó állapotára és annak megfelelően végzi el a módosítást.

2. CSS

2.1. Style attribútumban néhány elem formázása.

Az úgynevezett “*inline style*”-olást több alkalommal is használtam. Néhány példát bemutatnék a projektomból:

- `<h2 class="kanit-regular" style="text-align: center;">Válassz sportot:</h2>`

Ez az egyszerű stílus a h2-es címsort középre igazítja.

- `<div id="player-details-container" style="width:70%; margin-left: 15%;">`

Itt a style meghatározza a div-nek a szélességét illetve a div-től balra lévő külső teret növeli.

- `<p style='color:white'>`

A “`<p>`” tag esetén a betű színt fehérre állítja.

2.2. Azonosító alapján formázást.

Ez a stílusolás az egyik személyes kedvencem, amikor egy HTML elemnek adunk egy egyéni azonosítót (id) és erre hivatkozva formázzuk. Nagyon hasznos amikor sok hasonló elemünk van de csak egy bizonyos elemen szeretnénk végrehajtani a formázást. Ezen elemekre a “`<style></style>`” elemeken belül vagy egy külön CSS kiterjesztésű fájlban a “#” (hashtag) karakterrel hivatkozunk.

Példák:

- ```
#jatekosok-table {
 width: 80%;
 margin-left: 10%;
 font-size: 2vw;
 border: 5px solid black;
 border-radius: 5px;
 text-align: center;
}
```

A “*jatekos-table*” azonosítóval ellátott elemnek meghatározzuk a szélességét, baloldali külső terének méretét, betűméretét, keretét, és a szöveg igazítását.

- ```
#welcome {  
  display: flex;  
  flex-direction: row;  
}
```

A “*welcome*” azonosítóval ellátott elemnek flexbox elrendezést állítunk valamint azt is meghatározzuk, hogy a gyerek elemei sorokban rendezkedjenek el.

2.3. Osztály alapján formázást.

Az osztály (class) alapján formázás valamivel megengedőbb de ugyanolyan hatékony, mint az azonosító alapján történő formázás, de attól függetlenül ugyan olyan hatékony. Itt több elem is birtokolhatja ugyanazt az osztályt és ennek köszönhetően meg lesznek különböztetve a hasonló elemektől.

Példák:

- ```
.description {
 text-align: justify;
 font-size: 2ch;
}
```

A “description” osztállyal ellátott elemeknek állítunk szövegigazítást valamint karakter méretet.

- ```
.media-btn {  
    margin: 15px;  
    font-size: 22px;  
    border-radius: 10px;  
}
```

A “media-btn” osztálynak állít külső tér távolságot, karakter méretet illetve keret lekerekítést.

2.4. Táblázat formázása.

Mivel egy táblázatom volt és annak formázását felhasználtam [2.2. részben](#) az első példában ezért ezt nem mutatnám be még egyszer, a redundanciát elkerülve.

2.5. Menü kialakítása.

A menüt absztrakt módon egy külön CSS fájlban formáztam, hogy minden HTML oldal számára elérhető legyen. A menüsáv a képernyő bal oldalán kapott helyett felsorolva az oldalakat. Minden egyes elem kapott stílust. Volt amelyik osztállyal volt ellátva és volt amelyekre az szülő elem osztályán keresztül hivatkoztam. Az alábbi példában bemutatom a kódot.

```

.sidebar {
  padding: 10px;
  height: 100%;
  width: 250px;
  position: fixed;
  top: 0;
  left: -250px;
  background-color: #111;
  overflow-x: hidden;
  transition: 0.5s;
  font-family: "Kanit", sans-serif;
  font-weight: 400;
  font-style: normal;
}

.sidebar a {
  padding: 2%;
}

.sidebar ul {
  margin: 0;
  padding: 0;
  list-style: none;
}

.sidebar ul li {
  padding: 10px 15px;
}

.sidebar ul li a {
  text-decoration: none;
  color: #fff;
  display: block;
}

.sidebar ul li a:hover {
  background-color: #00a29c;
}

```

A .sidebar osztállyal ellátott menü elvégzett formázások:

- a konténer és a tartalom közötti térközt bállít
- „magasságát és szélességét meghatározza
- pozícióját rögzítetté teszi
- képernyő tetejétől és bal oldalától való eltérését beállítja
- háttérszínt állít
- a konténer görgető sávját elrejt
- áttűnés időtartamát meghatározza

- betűtípust, betűvastagságot, és betűstílust állít.

A többi stílus véleményem szerint majdnem mind említve volt már ezért azokat nem részletezném.

2.6. Háttérszín.

Minden HTML oldalam rendelkezik háttérszínnel valamint némely elem szintűgy mint az előző témánál bemutatott menü is. Amit itt bemutatnék az a háttérszín változtató kód lenne. Ezt a kódot a *“WebProg/public/js/background-color.js”* fájl tartalmazza.

background-color.js:

```
$(document).ready(function () {
    $("#colorPicker").change(function () {
        let color = $(this).val();
        $("body").css("background-color", color);
    });
});
```

Az kód lekéri a *“colorPicker”* azonosítóval ellátott elem értékét, ami egy rgb színkód és ezt a színkódot a program beállítja a HTML oldal háttérszínének.

2.7. Linkek formázása.

Linkeket több helyen is formáztam, köztük a menüben, de a precizitás kedvéért bemutatok pár példát a projektből.

- `a {color: rgb(0, 0, 0);}`
- `.sidebar ul li a {
 text-decoration: none;
 color: #fff;
 display: block;
}`

2.8. Űrlap elemek, gombok formázása.

Főképp az űrlapon végeztem formázás az elemein kevésbé, mivel személy szerint nekem tetszett az összkép de minden formázást bemutatok ami az űrlapon történt.

Úrlap formázás:

```
#urlap {  
  width: 35%;  
  text-align: justify;  
  font-size: 2ch;  
  padding: 2%;  
  margin-left: 30%;  
  border: 1px solid black;  
  border-radius: 10px;  
  background-color: white;  
}
```

input formázás:

```
input {text-align: center;}
```

gomb formázás:

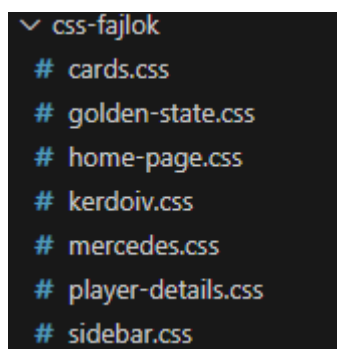
```
.media-btn {  
  margin: 15px;  
  font-size: 22px;  
  border-radius: 10px;  
}  
.back {  
  margin-top: 15px;  
  background-color: black;  
  color: white;  
  font-size: 1.5ch;  
  width: 10%;  
  border-radius: 10px;  
}
```

A gomb formázáshoz vett példák a
"WebProg/public/css-fajlok/home-page.css" fájlban találhatók.

2.9. Egy vagy több .CSS fájlban is legyen formázás, és a html-ben a <style> tag-ben is.

A formázások nagy része külön CSS fájlokban van kiszervezve, illetve egy oldalnál használtam a “<style>” tag-et formázásra a feladat kedvéért.

A CSS fájlok megtalálhatók a “WebProg/public/css-fajlok” mappában.



A “<style>” tag-ben történő formázás pedig a “WebProg/public/golden-state-page.html” fájlban található a “<head>” részben. Mivel az túl hosszú lenne ide berakni és nem szeretném nagyon elcsúfítani a dokumentumot ezért a következő github linken megtekinthető. [golden-state.html](https://github.com/...golden-state.html)

3. JavaScript, jQuery, JSON, AJAX

3.1. form ellenőrzés, hiba esetén az egyes űrlap elemeknél kiírja, hogy mi a hiba, css-t állít pl. piros keret, ha hibás az adott űrlap elem.

Ez a feladat megoldása a “WebProg/public/js/kerdoiv.js” JavaScript fájlban található. Mivel ez is egy elég terjedelmes kód lett ezért nem szívesen raknám be eme dokumentumba csak az github linkjét. [kerdoiv.js](https://github.com/...kerdoiv.js)

Egy rövid kód magyarázattal viszont szolgálnék, amely nagyvonalakban összefoglalja a működését.

Kód magyarázat: Elsősorban a kód megvizsgálja név alapján a “yesno” nevű rádiógombok értékét. Amennyiben az “Igen” értékű rádió gomb van bejelölve akkor az űrlapon alatta lévő elemek tiltás (disabled) attribútumát hamissá teszi, aminek hatására kitölthetővé válnak. Ellenkező esetben letiltódnak. Ezután amikor az “#urlap” azonosítójú űrlap elküldése megtörténik a kód először megvizsgálja, hogy minden kötelező mező kitöltött-e. Ha a név vagy az életkor mező üres, akkor piros színnel beállítja a mező keretét, és “formValid” változót hamisra állítja. Ha minden kötelező

mező kitöltve van, akkor létrehoz egy formázott adatsorokat tartalmazó HTML-kódot a beküldött adatokkal. Ezután hozzáadja az új adatokat az űrlap eredményeinek tartalmához (#form-eredmeny elem). Végül meggátolja az alapértelmezett űrlap beküldési működést az `“event.preventDefault()”` függvényvel, hogy ne történjen meg az oldal frissítése a gomb megnyomásával.

3.2. jQuery animáció

jQuery animációt főként a `“WebProg/public/js/home-page.js”`-ben használtam, ahol `“.animate”`-et használtam az adatok betöltésére amikor kiválasztanak egy adott sportot. Valamint még az oldal vissza állításánál használtam a `“.back”` gomb megnyomására. Ezen funkciók megtalálhatók az alábbi linken a 70. és 99. sorokban : [home-page.js](#)

Kód magyarázat: A 70.sorban kezdődő kód egy eseménykezelőt definiál az összes `“.sport-img”` osztályú elemre. Amikor ezekre az elemekre kattintanak, az eseménykezelő lefut. Először kinyeri az aktuális sport nevét az `“alt”` attribútumból. Ezután egy ciklusban végigiterál a `“sportsData”` tömbön, és megtalálja azt a sportot, amelynek a neve megegyezik az aktuálisan kiválasztott sporttal. Ha megtalálta a megfelelő sportot, akkor az adatokat egy új elembe rendereli, amelyet hozzáad a `“#main”` elemhez animációval (`$("#sports").animate()`). Az új elem tartalmazza a sport címét, képét, leírását, egy linket a leírás forrásához, a kedvenc csapatot, a csapat leírását és egy linket hozzá, valamint a kedvenc játékos adatait, képét, egy linket a játékos leírásához, valamint egy videót és néhány vezérlő gombot a videóhoz.

A 99.sorban kezdődő kód pedig szintúgy egy eseménykezelőt definiál a `“#main”` elemen belül az összes `.back` osztályú gombra. Amikor ezekre a gombokra kattintanak, az eseménykezelő lefut. Először animációval eltávolítja az aktuális sportot bemutató elemet, majd megjeleníti az `“#sports”` elemet ismét.

3.3. elemek kiválasztása: html tag név alapján, osztály alapján, azonosító (id) alapján stb

Ezen feladat véleményem szerint evidens ha JQuery-t használ az ember, hiszen a különféle események, animációk, stb mind igényelnek egy bizonyos “alanyt” vagy elemet ha úgy tetszik, amire alkalmazhatja a megírt kódot. Ezt viszonylag röviden mutatom be, hiszen felesleges lenne felsorolni minden egyes elem kiválasztását a projektben.

- **html tag név alapján:** `$("#body").css("background-color", color);`
- **osztály alapján:** `$(".position-nav").click()`

- azonosító alapján:

```
$("#player-details-container").html(playerDetails);
```

3.4. új html elem készítése, meglévő html elem módosítása

Ezt a feladatot kimeríti a [jQuery animáció](#) részben bemutatott kód, mivel ott “append”-el hozzáadunk a “#main” konténerhez adatokat ezzel változtatjuk annak tartalmát valamint ezek az adatok html tag-ekben vannak. Végül a “.back” osztállyal ellátott gombra kattintva törlődik a “#main” tartalma és visszatér régi állapotába.

3.5. Egy fájl elkészítése JSON formátumba (beágyazott)

Készítettem egy JSON fájlt a gyakorlati órán tanultak alapján majd ennek a fájlnak értékeit egy “getJSON” metódussal lekértam majd megjelenítettem egy táblázatban a HTML oldalon. A JSON fájl elérési útvonala “WebProg/public/js/golden-state.json” linkje [golden-state.json](#). A JavaScript kód elérési útvonala “WebProg/public/js/golden-state.js” linkje [golden-state.js](#).

3.6. JavaScript, jQuery – AJAX metódus használatával JSON fájl megjelenítése a weboldalon.

A [Egy fájl elkészítése JSON formátumba \(beágyazott\)](#) feladatban említett [golden-state.js](#) fájl tartalmazza a “getJSON” metódust, ami a lokális JSON fájlból olvassa ki az adatokat és jeleníti meg. Valamint van még egy JavaScript fájl, ami tartalmaz AJAX hívást és JSON adatokat jelenít meg a weboldalon, de viszont az szerver oldalról kapja az adatokat “http://localhost:3000/real-madrid-data” url-en keresztül a mySQL adatbázisból. Ennek linkje: [real-madrid-page.js](#)

Kódmagyarázat: A kód egy AJAX kérést küld a lokális szerverre, hogy lekérje a Real Madrid játékosainak adatait. Amikor a kérés sikeresen befejeződik, a válasz (response) adatait csoportosítja pozíció szerint (kapus, védő, középpályás, csatár, edző), majd ezeket az adatokat rendereli egy HTML konténerbe a “renderPlayers” függvény segítségével.

A “renderPlayers” függvény egy adott pozícióhoz tartozó játékosokat jeleníti meg egy adott konténerben. Minden játékoshoz hozzárendeli a “.card” osztályt az egyen dizájn érdekében. A végeredményben kártya jellegű megjelenést kölcsönöz a játékosoknak, amely tartalmazza a játékos nevét, mezszámát és pozícióját. Ha több mint 5 játékos van a konténerben, akkor a kártyák közül csak az első 5 jelenik meg, a többi pedig rejtett marad.

A *“.position-nav”* osztályú elemekre kattintva lehet lapozni a játékosok között. Ezek az elemek az adott pozícióhoz tartoznak, és a kattintásokra a konténerben lévő kártyák láthatóságát változtatják. Minden kattintásnál a következő 5 rejtett kártya válik láthatóvá, és az előző 5 látható kártya válik rejtetté. Emellett a kattintások számát is számolja, és a kattintások számától függően megváltoztatja a lapozó gombok megjelenítését.

Szerver

Feladat: NodeJS konfigurálása (server.js, package.json, package-lock.json, node_modules). A projekt megjelenítése a helyi szerveren (localhost:3000)

A projektem tartalmazza a feladatban meghatározott fájlokat a *“WebProg/”* elérési útvonalon. A *“server.js”* fájl felel a szerver futtatásért. A localhost:3000 porton keresztül elérhetőek a HTML oldalak. Valamint a *“server.js”* felel az adatbázis kapcsolatért is.

Összesen három függőséget implementáltam a projektbe, amiket használ a *“server.js”*. Ezek a következők: cors, express, mysql. Ezen függőségek megtalálhatók a *“package.json”* fájlban a *“dependencies”* alatt.

A github linkeket alább listázom:

- [server.js](#)
- [package.json](#)
- [package-lock.json](#)
- [node_modules](#)

Mivel régen foglalkoztam backendel ezért elővettem egy régi projektemet, amit még a középiskolában csináltam, hogy az alapján alkossam meg a server.js-t. Ennek a linkje: [régi projekt](#)

Adatbázis

Az adatbázisom nem túl nagy ezért úgy vélem egy kép megfelelő a szemléltetésére.

← T →		id	mez_szam	nev	teljes_nev	pozicio	erosebb_lab	szuletett	nemzetiseg
<input type="checkbox"/>	Módosítás Másolás Törölés	1	1	Courtois	Thibaut Courtois	Kapus	Jobb	1992-05-11	Belga
<input type="checkbox"/>	Módosítás Másolás Törölés	2	13	Lunin	Andriy Lunin	Kapus	Jobb	1999-02-11	Ukrán
<input type="checkbox"/>	Módosítás Másolás Törölés	3	25	Kepa	Kepa Arrizabalaga	Kapus	Bal	1994-10-03	Spanyol
<input type="checkbox"/>	Módosítás Másolás Törölés	4	2	Carvajal	Daniel Carvajal Ramos	Védő	Jobb	1992-01-11	Spanyol
<input type="checkbox"/>	Módosítás Másolás Törölés	5	3	Militao	Éder Gabriel Militão	Védő	Bal	1998-01-18	Brazil
<input type="checkbox"/>	Módosítás Másolás Törölés	6	4	Alaba	David Alaba	Védő	Bal	1992-06-24	Osztrák
<input type="checkbox"/>	Módosítás Másolás Törölés	7	6	Nacho	José I. Fernández Iglesias	Védő	Jobb	1990-01-18	Spanyol
<input type="checkbox"/>	Módosítás Másolás Törölés	8	17	Lucas	Lucas Vázquez Iglesias	Védő	Jobb	1991-07-01	Spanyol
<input type="checkbox"/>	Módosítás Másolás Törölés	9	20	Garcia	Francisco José García Torres	Védő	Jobb	1999-08-14	Spanyol
<input type="checkbox"/>	Módosítás Másolás Törölés	10	22	Rudiger	Antonio Rüdiger	Védő	Bal	1993-03-03	Német
<input type="checkbox"/>	Módosítás Másolás Törölés	11	23	Mendy	Ferland Mendy	Védő	Bal	1995-06-08	Francia
<input type="checkbox"/>	Módosítás Másolás Törölés	12	5	Bellingham	Jude Bellingham	Középpályás	Jobb	2003-06-29	Angol
<input type="checkbox"/>	Módosítás Másolás Törölés	13	8	Kroos	Toni Kroos	Középpályás	Bal	1990-01-04	Német
<input type="checkbox"/>	Módosítás Másolás Törölés	14	10	Modric	Luka Modric	Középpályás	Bal	1985-09-09	Horváth
<input type="checkbox"/>	Módosítás Másolás Törölés	15	12	Camavinga	Eduardo Camavinga	Középpályás	Bal	2002-11-10	Francia
<input type="checkbox"/>	Módosítás Másolás Törölés	16	18	Tchouameni	Aurélien Tchouaméni	Középpályás	Jobb	2000-01-27	Francia
<input type="checkbox"/>	Módosítás Másolás Törölés	17	19	Ceballos	Daniel Ceballos Fernández	Középpályás	Jobb	1996-08-07	Spanyol
<input type="checkbox"/>	Módosítás Másolás Törölés	18	15	Valverde	Federico Santiago Valverde Dipetta	Középpályás	Jobb	1998-07-22	Uruguay
<input type="checkbox"/>	Módosítás Másolás Törölés	19	24	Guler	Arda Güler	Középpályás	Jobb	2005-02-25	Török
<input type="checkbox"/>	Módosítás Másolás Törölés	20	7	Vini	Vinicius Paixao de Oliveira Junior	Csatár	Bal	2000-07-12	Brazil
<input type="checkbox"/>	Módosítás Másolás Törölés	21	11	Rodrygo	Rodrygo Goes	Csatár	Jobb	2001-01-09	Brazil
<input type="checkbox"/>	Módosítás Másolás Törölés	22	14	Joselu	Joselu Mato	Csatár	Jobb	1990-03-27	Spanyol
<input type="checkbox"/>	Módosítás Másolás Törölés	23	21	Brahim	Brahim Díaz	Csatár	Jobb	1999-08-03	Spanyol
<input type="checkbox"/>	Módosítás Másolás Törölés	24	0	Ancelotti	Carlo Ancelotti	Edző		1959-06-10	Olasz

Mint látható az adatbázis minden egyes játékoshoz rendel egy **“id”** értéket. Ez autoincrement, szám típusú adat és ez az elsődleges kulcs. Habár itt igazából elhagyható az elsődleges kulcs, mivel nincs több táblám, amivel össze tudnám kötni. Ez után a játékosokról tartalmaz különféle adatokat mint a **“mez_szam” “nev” “teljes_nev” “pozicio” “erosebb_lab” “szuletett” “nemzetiseg”** . Ezen adatokat egy táblázatban prezentálom.

Név:	mez_szam	nev	teljes_nev	pozicio	erosebb_lab	szuletett	nemzetiseg
Típus:	int(11)	varchar(250)	varchar(250)	varchar(250)	varchar(250)	date	varchar(250)

Futtatás eredményei:

A futtatás eredményeit felraktam egy nyilvános drive meghajtóra videó formájában.
Ez a linke kattintva elérhető: [futás videó](#)

Források:

Real Madrid játékosok adatai és dizájn ötlet: [real madrid adatok](#)

Real Madrid ismertető szöveg: [real madrid wiki](#)

Real Madrid videó: [rm video](#)

Labdarúgás ismertető szöveg: [labdarúgás wiki](#)

Kaká ismertető szöveg: [kaká wiki](#)

Golden State Warriors játékosok adatai és ismertető szöveg a csapatról: [golden state warriors wiki](#)

Golden State Warriors video: [gsw video](#)

Kosárlabda ismertető szöveg: [kosárlabda wiki](#)

Stephen Curry ismertető szöveg: [steph curry wiki](#)

Mercedes AMG PETRONAS ismertető szöveg: [mercedes wiki](#)

Mercedes oldal dizájn ötletek: [mercedes fooldal](#)

Lewis Hamilton ismertető szöveg: [lewis hamilton wiki](#)

Lewis Hamilton video: [hamilton video](#)

Formula-1 ismertető szöveg: [formula-1 wiki](#)

PNG képek: [png képek](#)

Color Picker: [imagecolorpicker](#)

Google Fonts: [poppins](#) [kanit](#) [open sans](#)

Video compressor: [veed.io](#)