|  |  |
| --- | --- |
| Beiskolázás és Felvételi | Pogány Frigyes Technikum | EDIR  BKSZC Pogány Frigyes Technikum | |
| F1 Statisztikák és Hírek  Szoftverfejlesztő és -tesztelő vizsgaremek  dokumentáció  2025 május | |
| Készítette:  Ináncsi Krisztián  Bartók Krisztián  Fábián Tamás | Konzulens:  - |

Tartalomjegyzék

[Ismertető 4](#_Toc192791474)

[Projekt neve és célja 4](#_Toc192791475)

[Technológiai stack 4](#_Toc192791476)

[Adatbázisok 4](#_Toc192791477)

[Indoklás 4](#_Toc192791478)

[Szerepek 5](#_Toc192791479)

[Telepítési útmutató 5](#_Toc192791480)

[Felhasználói dokumentáció 6](#_Toc192791481)

[Híroldal 6](#_Toc192791482)

[Statisztikai oldal 10](#_Toc192791483)

[Fejlesztői dokumentáció 11](#_Toc192791484)

[Rendszerarchitektúra 11](#_Toc192791485)

[News adatbázis 12](#_Toc192791486)

[Ismertető 12](#_Toc192791487)

[Szerkezet 12](#_Toc192791488)

[Kapcsolatok 12](#_Toc192791489)

[Stats Adatbázis 13](#_Toc192791490)

[Ismertető 13](#_Toc192791491)

[Szerkezet 13](#_Toc192791492)

[Kapcsolatok 13](#_Toc192791493)

[HUB 14](#_Toc192791494)

[News (Hírek és szezoninformációk 15](#_Toc192791495)

[Frontend 16](#_Toc192791496)

[HTML Fájlok 16](#_Toc192791497)

[Stílusfájlok 33](#_Toc192791498)

[JavaScript fájlok 35](#_Toc192791499)

[Backend 36](#_Toc192791500)

[API végpontok 36](#_Toc192791501)

[Server.js 42](#_Toc192791502)

[auth-api-manager.js 44](#_Toc192791503)

[dropdown-api-manager.js 44](#_Toc192791504)

[forum-api-manager.js 44](#_Toc192791505)

[news-api-manager.js 44](#_Toc192791506)

[race-result-api-manager.js 44](#_Toc192791507)

[schedule-api-manager.js 44](#_Toc192791508)

[standlist-api-manager.js 44](#_Toc192791509)

[tracks-api-manager.js 44](#_Toc192791510)

[Stats (statisztikai oldal) 45](#_Toc192791511)

[Frontend 45](#_Toc192791512)

[HTML Fájlok 45](#_Toc192791513)

[Stílusfájlok 45](#_Toc192791514)

[JavaScript fájlok 45](#_Toc192791515)

[Backend 45](#_Toc192791516)

[API végpontok 45](#_Toc192791517)

[Server.js 45](#_Toc192791518)

[Források 46](#_Toc192791519)

# Ismertető

## Projekt neve és célja

Az **F1 Stats & News** egy weboldal, amely átfogó információkat biztosít a Formula 1 rajongók számára. Az oldal két fő szekcióból áll:

* **Stats**: A Formula 1 történelmi statisztikáit és eredményeit tartalmazza.
* **News**: A legfrissebb híreket és a szezon aktuális információit jeleníti meg.

A projekt célja, hogy a felhasználók gyorsan és könnyen hozzáférjenek a legfrissebb F1-es hírekhez és statisztikákhoz, egy jól strukturált, felhasználóbarát platformon keresztül.

## Technológiai stack

A weboldal fejlesztéséhez az alábbi technológiákat és eszközöket használtuk:

* **Frontend**:
  + HTML5
  + CSS3
  + JavaScript
* **Backend**:
  + Node.js
* **Adatbázis**:
  + MySQL

## Adatbázisok

A projekt két fő adatbázist használ:

* **F1-Stats**: Az összes statisztikai adat forrása.
* **F1-News**: Az aktuális hírek és szezoninformációk tárolása.

## Indoklás

* **HTML5 & CSS3**: Reszponzív és modern dizájn biztosítása.
* **JavaScript & Node.js**: Dinamikus, gyors és skálázható alkalmazás fejlesztése.
* **MySQL**: Nagy mennyiségű adat gyors és hatékony kezelése.

## Szerepek

* Ináncsi Krisztián
  + Statisztikai oldal kinézetének megtervezése, megvalósítása
  + Híroldal kinézetének megtervezése, megvalósítása
  + Híroldal működése: Főoldal funkciók, dropdown funkciók, pályainformációk
* Bartók Krisztán
  + Híroldal adatbázisa.
  + Híroldal működése: Fórum oldal, Hír részleg, Bejelentkezés és Regisztráció, Profil funkciók, futam végeredmények.
* Fábián Tamás
  + HUB részleg megvalósítása.
  + Domain és webszerver megvalósítása.
  + Statisztikai oldal kinézetének megtervezése, megvalósítása.
  + Statisztikai oldal teljes funkciójának megvalósítása.

## Telepítési útmutató

# Felhasználói dokumentáció

**Weboldal elérése:** [Link]

A weboldal két, egymástól különálló témát dolgoz fel, így első lépésként a **Start** oldalon kell választanod, hogy melyik weboldalra szeretnél navigálni. Az oldal két fő részből áll, és mindkét téma saját funkciókkal rendelkezik, így bármikor lehetőséged van átváltani a két szekció között.

Miután kiválasztottad a kívánt témát, a választott oldal főoldalára kerülsz. Itt különböző információk és funkciók lesznek elérhetők, amelyek a választott témára vonatkoznak.

## Híroldal

1. Funkcionalitás:
   1. **Főoldal:**
      1. A főoldalon egy dinamikus panel található, amelyen a következő verseny visszaszámlálása látható. A visszaszámláló pontosan mutatja, hogy hány óra és perc van hátra az adott futam kezdetéig. A panel alatt a hétvége legfontosabb eseményeinek dátumai jelennek meg, valamint egy gomb, amely lehetővé teszi a futamok megtekintésére szolgáló helyszínek listázását.
      2. Ezt követően a Formula-1 aktuális hírei találhatók, amelyek az újdonságokat és friss információkat tartalmazzák. Amennyiben régebbi hírek iránt érdeklődsz, a "Read More" gombra kattintva minden korábbi hír elérhetővé válik egy külön oldalon.
      3. A következő szekcióban a szezon állása kerül bemutatásra, amely tartalmazza a Pilóta- és Konstruktőri tabellákat. Ezen kívül egy harmadik lehetőségként az eddigi futamok eredményei is elérhetők, így könnyen áttekintheted a versenyek eddigi kimenetelét.
      4. Lentebb a versenynaptár található, ahol egy adott futamra kattintva részletes információkat találsz az adott versenyről, beleértve a pálya adatait, a verseny időpontjait, valamint egyéb statisztikai információkat is.
      5. Végül, a következő szekció a Formula-1 technológiai híreivel foglalkozik, bemutatva minden olyan cikket, amely a sportág technikai újításait és fejlesztéseit érinti. Itt naprakész információkat találhatsz a legújabb technológiai trendekről és fejlesztésekről.
   2. **Pályainformációkkal kapcsolatos oldal:**
      1. Ezen az oldalon a futamokat rendező versenypályákkal kapcsolatos információk láthatóak. A fejléc alatt egy visszaszámláló helyezkedik el, amely pontosan mutatja, hogy óra és perc van hátra az éppen megtekintett pálya versenyéig.
      2. A visszaszámláló alatt az oldal két részre oszlik. Egyik az adatokat megjelenítő panel, másik pedig az adott pálya vonalvezetését mutatja be.
         1. Az információs panel 3 fontos információt mutat be. Általános adatok a pályával kapcsolatban: Első nagydíj dátuma, Futamon megtett körök száma, Pálya hossza, Versenytáv és a körrekordot. Utolsó futamgyőztesek listáját, illetve a hétvége eseményeinek dátumát.  
            Az információk megjelenítésének váltását 3 gomb teszi lehetővé, ami az adatok felett jelenik meg.
   3. **Fórum**
      1. Főoldal:
         1. A fórum főoldalon a tartalom 3 fő témára oszlik:
            1. Verseny topikok

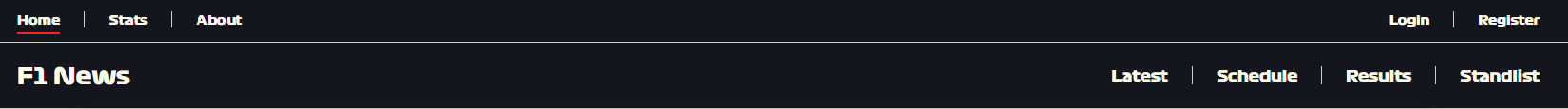
Minden versenyhétvégén a rendszer automatikusan létrehoz egy dedikált topikot, ahol a felhasználók valós időben megvitathatják az adott futamon történt eseményeket. Ezek a topikok biztosítják a versenyekkel kapcsolatos élmények és vélemények megosztásának lehetőségét a többi rajongóval.

* + - * 1. Felhasználói Topikok

A felhasználók saját témájú beszélgetéseket is indíthatnak egyéni vagy specifikus témákban. Új topik létrehozásához kattintson a fő tartalom felett található **„Create topic”** gombra, majd adja meg a kívánt címet és leírást. Az így létrehozott topikok minden felhasználó számára elérhetők és hozzászólhatóak.

* 1. Futamok végeredménye
  2. Hírek

1. Navigáció, Menürendszer



* 1. Az oldalak tetején található navigációs sáv minden oldalon azonos elrendezéssel és funkciókkal rendelkezik, hogy biztosítsa a felhasználók számára a könnyű és gyors navigációt a weboldalon. A navigációs sáv két fő részre osztható:
     1. **Felső sáv:**
        1. **Home gomb:** Ezen a gombon keresztül bármikor visszatérhetsz a főoldalra, függetlenül attól, hogy melyik aloldalon tartózkodsz.
        2. **Stats gomb:** A "Stats" gomb megnyomásával átirányítunk a statisztikai oldalra, ahol az aktuális adatok és eredmények találhatók.
        3. **About gomb:** Az "About" gombra kattintva hozzáférhetsz az oldal működésével kapcsolatos információkhoz, valamint a szerzői jogi adatokat és egyéb jogi nyilatkozatokat.
        4. **Login és Register gombok:** A **Login** és **Register** gombok segítségével bejelentkezhetsz vagy regisztrálhatsz egy új felhasználói fiókot. Ha már bejelentkeztél, ezek a gombok a felhasználónevedet jelenítik meg, és egy legördülő menüt kínálnak, amely lehetővé teszi a fiók kezelését, például:
           1. **Fiókadatok módosítása**
           2. **Kijelentkezés**
     2. **Alsó sáv:**
        1. **Latest Gomb**
        2. **Schedule Gomb**
        3. **Results Gomb**
        4. **Standlist Gomb**

1. **Felhasználói beállítások:**
   * 1. A felhasználói fiók módosításához a felső navigációs sávon a felhasználó neveddel megegyező gombra kell kattintanod, miután megjelenik egy menü sáv, ahol az első opcióra, a „View Profile”-ra kattintva az oldal átirányít egy másik oldalra, ahol megjelenik a felhasználóneved, az emailed. Emellett pedig jobb oldalon a módosítási lehetőségeid: Felhasználónév módosítása, Email módosítása, Jelszó módosítása.
2. **Regisztráció és bejelentkezés**
   1. **Regisztráció**
   2. **Bejelentkezés**

## Statisztikai oldal

# Fejlesztői dokumentáció

## Rendszerarchitektúra

Központi HUB (localhost:3000)

* Ez a központi komponens indítja el a teljes rendszert.
* Egy egyszerű HTML fájlt kezel, amely tartalmaz két gombot: Stats és News
* A felhasználó ezekkel a gombokkal érheti el az egyes adatbázisokhoz tartozó oldalakat.

News oldal (localhost:3001)

* Az F1-news adatbázis kiszolgálója.
* Hírek és aktualitások megjelenítésére koncentrál az F1 világából és a jelenlegi szezonról.
* Támogatja az F1 témájú hírtartalmak gyors elérését és kezelését.

Adatáramlás és működés

* A HUB komponens közvetít a felhasználó és az adatbázisokkal kommunikáló Stats illetve News oldalak között.
* Amikor a felhasználó rákattint a megfelelő gombra a HUB-on, az az adott oldalra irányítja őt, amely egyedi kiszolgálóval működik.

## News adatbázis

### Ismertető

### Szerkezet

### Kapcsolatok

## Stats Adatbázis

### Ismertető

### Szerkezet

### Kapcsolatok

## HUB

## News (Hírek és szezoninformációk

A news oldal a Formula-1 aktuális szezonjának legfontosabb híreit és kapcsolódó információkat jeleníti meg. A modern, felhasználóbarát kialakítás lehetővé teszi a látogatók számára, hogy könnyedén hozzáférjenek az alábbi tartalmakhoz és funkciókhoz:

1. Következő futam információi
   1. Megjeleníti a következő verseny dátumát, helyszínét és időpontjait.
   2. Streaming és televíziós közvetítési információk
   3. Dinamikusan frissülő visszaszámláló
2. Friss hírek és szezoninformációk
   1. Formula-1-gyel kapcsolatos általános hírek listázása
   2. Cikkek API-n keresztüli lekérése és megjelenítése
3. Tabellák és futamok végeredményei
   1. Versenyek eredményeinek táblázatos megjelenítése
   2. Pontverseny és egyéni/csapatbajnokság állása
   3. Adatok valós idejű frissítése adatbázisból
4. Technológiai hírek és elemzések
   1. Új fejlesztések, szabályváltozások és autókkal kapcsolatos technikai újdonságok
   2. Interaktív cikkek és szakértői elemzések
5. Fórum és közösségi interakció
   1. Versenyhétvégékhez kapcsolódó beszélgetési lehetőség
   2. Felhasználók által létrehozott egyéni topikok
   3. Népszerű témák kiemelése.

### Frontend

#### HTML Fájlok

Általános Struktúra minden HTML fájlban

A weboldalak szerkezete moduláris és megegyező, amely megkönnyíti a karbantartást és a bővítést. Az alap HTML szerkezet az alábbi részekből áll:

* Header (Közös)
* Main (Tartalmi blokkok)
* Footer (Közös elem)

HTML Doctype és Nyelv

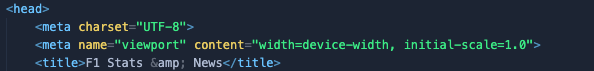
A dokumentum HTML5 doctype-ot használ:

A nyelvi attribútum (lang="en") az angol nyelvre van állítva, jelezve a böngészőknek és a keresőmotoroknak, hogy a dokumentum nyelve angol.



Metaadatok és Cím

A <head> szakasz tartalmazza a metaadatokat és a címkét:



* A <meta charset=”UTF-8”> beállítás a karakterkódolást UTF-8-ra állítja, biztosítva az univerzális karakterkészlet támogatását.
* A <meta name=”viewport” content=”width=device-width, initial-scale=1.0”> beállítás a reszponzív dizájnt támogatja, amely mobilbarát megjelenítést segíti elő.
* A <title> elem az oldal címét határozza meg, amely a böngésző fülén jelenik meg.

CSS Stíluslapok

A fájl különböző CSS fájlokat hivatkozik, amelyek a weboldal megjelenését és stílusát határozzák meg.

A screen shot of a computer code

Description automatically generated

A különböző CSS fájlok biztosítják az egységes stílusbeállítások betöltését segítik elő, illetve az adott elemek könnyebb módosítását minden oldalon:

* Alapértelmezett stílusok visszaállítása: reset.css
* Tipográfiai stílusok: typography.css
* változók variables.css
* gombok: buttons.css
* táblázatok: table.css
* navigációs elemek: nav.css
* fejléc header.css
* Lábléc: footer.css
* Reszponzív megjelenítés: responsive.css
* Oldalspecifikus stílusok: pl.: index.css, tracks.css

Betöltési Animáció és Tartalom:

A betöltési animáció minden oldal számára iframe-en keresztül jelenik meg, teljesképernyős méretben.



Minden oldal tartalma egy content osztályú div-ben helyezkedik el, amely a betöltési animáció megjelenítése és elrejtése érdekében kapott helyet:

A betöltési animáció és tartalom megjelenítéséért felelős JavaScript kód gondoskodik arról, hogy a betöltés után a megfelelő tartalom jelenjen meg minden oldalon.

Fejléc és Navigációs sáv

A fejléc és Navigációs sáv a header-container-ben találhatóak:

A screen shot of a computer code

Description automatically generated

A navigáció és fejkéc tartalmát JavaScript fájlok generálják. A navigációs sáv esetében a tartalom változik annak függvényében, hogy a felhasználó be van-e jelentkezve vagy sem.

A dropdown.js kód segítségével az alábbi gombok jelenítenek meg menüsávokat:

* Latest
* Schedule
* Result

A menüsáv tartalmáról a dropdown-api-manager.js gondoskodik.

Lábléc

A lábléc tartalmát a footer elem tartalmazza:

A blue background with white text

Description automatically generated

A lábléc tartalmát JavaScript fájl generálja, az módosítások egyszerűsítése, és egységesítés megkönnyítésé érdekében.

loader.html

Ez az HTML fájl az oldalak közti navigálás között jelenik meg, mint töltő animáció. Minden oldal egy <iframe> tagben indulásnál betölti ezt az oldalt, és addig jeleníti meg, amíg a saját tartalma be nem töltődik.

Index.html

Az index.html az alábbi szekciókat tartalmazza:

* event-container
* daily-news
* Standlist
* Schedule
* Tech News

Ezeknek a szekcióknak a tartalmai dinamikusan kerülnek betöltésre. Mindegyik szekciónak megvan a maga JavaScript fájlja a könnyebb átláthatóság érdekében.

Event-container szekció

Az event-container szekció felel a következő hétvége eseményeinek megjelenítéséért. Emellett egy visszaszámláló is helyet kapott rajta, amely mindig a következő eseményhez számolja vissza az időt.

A screen shot of a computer program

Description automatically generated

Működés:

#race-div-be töltődik az éppen következő versenyhétvégét rendező ország neve és zászlaja

#timer-paragraph, itt jelenik meg a következő eseményhez hátralévő idő.

#regular-event-table és #sprint-event-table: Ezen a táblák valamelyike fogja megjeleníteni a versenyhétvége eseményeinek időpontjait. A kód a kapott adatok alapján eldönti, hogy melyik táblát tölti be.

Ezeket az adatokat a schedule-api-manager.js tölti be.

News Szekció

A news szekció a legfrissebb híreket tartalmazza, és bemutatja a legfontosabb F1-es eseményeket.

A screen shot of a computer program

Description automatically generated

Működés:

#daily-news-container: Itt jelennek meg a legfrissebb hírek. A news-api-manager.js szkript felelős az API hívásból kapott adatok megjelenítéséért.

Regular-news-btn : Ezzel a gombbal a felhasználó több hírhez is hozzáférhet, amelyeket az API biztosít. A gomb kattintására a felhasználót átirányítjuk a news-layout.html/regular-news oldalra.

Standlist Szekció

A standlist szekció az aktuális F1-es bajnoki állásokat mutatja be. Itt jelennek meg a pilóták, csapatok, és a versenyeredmények.

A screen shot of a computer program

Description automatically generated

Működés

Navigáció: Az oldalon három különböző tábla között lehet váltani: pilóta állás, csapat állás és versenyeredmények. A driver-standlist-btn, constructor-standlist-btn, és race-result-btn gombokkal válthatunk a különböző táblák között. A multi-page-ui-manager.js felelős az interakciók kezeléséért.

Táblák: A driver-standlist-table, constructor-standlist-table, és race-result-table táblák tartalmazzák a különböző információkat, és dinamikusan töltődnek be a standlist-api-manager.js segítségével.

Schedule szekció

A schedule szekció a versenynaptárt listázza ki egy görgethető menü keretei között.

A screen shot of a computer code

Description automatically generated

Működés

#racenames-container: Itt jelennek meg a közelgő versenyek nevei, és hétvége dátuma egy görgethető listában. Mind ezt a schedule-api-manager.js szkript jeleníti meg.

Tech News szekció

A tech-news szekcuó a legújabb technikai híreket tartalmazza az F1 világából.

A screen shot of a computer code

Description automatically generated

Működés

A #tech-news-container: itt jelennek meg a legfrissebb technikai hírek. A news-api-manager.js szkript felelős a technikai hírek megjelenítéséért.

Tech-news-btn: Ezzel a gombbal a felhasználó több technikai hírhez is hozzáférhet a news-layout.html/tech-news oldalon.

auth.html

Ez az oldal a bejelentkezési és regisztrációs szekciót tartalmazza.

Bejelentkezés szekció

A bejelentkezési szekció lehetővé teszi a felhasználók számára, hogy bejelentkezhessenek saját felhasználói profiljukba.

A screen shot of a computer program

Description automatically generated

Működés:

A login-section a bejelentkezési űrlapot tartalmazza. A felhasználóknak itt kell megadniuk a felhasználónevüket vagy email címüket és jelszavukat.

A form tartalmaz két bemeneti mezőt: felhasználónév vagy email címet és a jelszót. Illetve egy check boxot, ami a remember me opcióval beállíthatja, hogy a rendszer megjegyezze a bejelentkezési adatokat.

A felhasználó a Login gombra kattintva próbálkozhat a bejelentkezéssel. Sikeres bejelentkezés esetén pedig az index.html-re irányítjuk.

Regisztráció szekció

A regisztráció szekció lehetővé teszi az új felhasználók számára a fiók létrehozását.

A screenshot of a computer

Description automatically generated

Működés:

A register-section a regisztrációs űrlapokat tartalmazza. Az űrlapok kitöltésével új felhasználók regisztrálhatják fiókjukat a weboldalon.

A register-form tartalmazza a következő mezőket: felhasználónevet, e-mail címet és jelszót.

**Checkbox**: A felhasználónak el kell fogadnia az oldalon található jogi feltételeket, mielőtt folytathatja a regisztrációt.

A Register gomb megnyomásával a felhasználó adatokat küldhet a szervernek az új fiók létrehozásához.

A regisztrációs űrlapot illetve bejelentkezési űrlapot az auth-api-manager.js szkript használa, hogy az adatokat eljuttassa a backend rendszerhez. A regisztrációs információkat az API tárolja el az adatbázisban.

tracks.html

Timer szekció

A timer szekcióban az adott futam kezdetéhez számított visszaszámláló található.

Emellett ebben a szekcióban megjelenik a versenyhétvége hivatalos neve és a versenyt rendező ország zászlaja.

A computer screen shot of a code

Description automatically generated

Működés:

<span id=”headline-flag”></span> elem az adott verseny országának zászlaját fogja megjeleníteni.

<div id =”race-name-headline”></div> tartalmazza az adott verseny nevét.

A tracks-api-manager.js tölti fel ezeket az elemeket, emellett ez a szktipt a race-name-headline div-be betölt egy timer-paragraph-ot amely visszaszámol az adott verseny kezdetéhez.

Main Szekció:

A main szekció az oldal fő tartalmát jeleníti meg. Ez a szekció a versenypályával kapcsolatos adatokat tartalmazza.

Működés:

[KÉP]

A pályával kapcsolatos információkat a trackData-content jeleníti meg. Ezen belül az információk váltakozását a navigációs gomb teszi lehetővé, aminek működését a multi-page-ui-manager.js szkript biztosítja.

A versenypálya képét a track-img jeleníti meg.

Forum-layout.html

Ez a HTML fájl az oldal fórum részét hivatott megjeleníteni.

A main szekció a fórum oldal teljes tartalmát jeleníti meg, ahol a felhasználók topicokat hozhatnak létre, és meglévő fórum topicok között böngészhetnek.

A screen shot of a computer code

Description automatically generated

Működés

* #container: A fórum fő tartalmi blokkját foglalja magába.
* #topic-controls: Egy navigációs rész, amely lehetőséget biztosít új topicok létrehozására, emellett admin jogosúltságú profilok esetén a Report lista megtekintésére.
* #race-section: A versenyekkel kapcsolatos fórumtopikok listáját tartalmazza. Minden versenyhétvégén a rendszer megnyit egy topikot az adott futammal kapcsolatban.
* #user-section: A felhasználók által létrehozott topikok listáját jeleníti meg.

A forum-api-manager.js szkript kezeli a rendszer- és felhasználók által létrehozott topikok betöltését, és egyéb funkcióit.

news-layout.html

Ez a HTML fájl az összes cikket hivatott megjeleníti az adott témával kapcsolatban.

Két fő cikk téma:

* Legfrissebb hírek a Formula-1 világából
* Technikai hírek

[A screen shot of a computer code

Description automatically generated

Működés:

* #news-section: Ez a szekció dinamikusan töltődik be a news-api-manager.js segítségével.
* Az oldalon két típusú hír érhető el:
  + Legfrissebb hírek
  + Technikai hírek

Az oldal a szerint dönti el, hogy melyik típust jeleníti meg, hogy a felhasználó az index oldalon melyik hír szekciónál kattintott a Read More gombra.

profile.html

Ez az oldal a felhasználói profil adatainak megjelenítéséért és szerkesztéséért felelős.

A screen shot of a computer program

Description automatically generated

Működés:

* #profile-datas: A felhasználói profil adatait jeleníti meg.
* #settings-form: A felhasználói adatok módosítására szolgáló űrlap, amely három fő szekcióból áll:
  + Felhasználónév módosítása
  + Email cím módosítása
  + Jelszó módosítása
* Navigációs elemek: Tartalmaz két fontos gombot:
  + Kijelentkezés (#logout)
  + Fiók törlése (#delete-account)

A profile-api-manager.js kezeli a profiladatok betöltését és módosítását.

raceresult.html

Ez a HTML fájl két fő szekciót tartalmaz:

* Versenylista (#race-list-section)
* Veresnyeredmények (#race-result-section)

A screen shot of a computer code

Description automatically generated

Működés

* #race-list-section: Itt jelennek meg azon F1-es futamok listája, melyeknek már van hivatalos végeredménye.
* #race-result-section: Itt jelennek meg a kiválasztott futam eredményei.

A race-result-api-manager.js kezeli az adatok betöltését az adatbázisból, és gondoskodik a dinamikus tartalom megjelenítéséről.

race-uploader.html

Ez az oldal a futamok végeredményének feltöltését hivatott segíteni. A feltöltését a result-uploader-script.js biztosítja.

Az oldal fő tartalmi szekciója egy konténer (#container), amelyben a szkript segítségével select elemek kerülnek be.

A computer screen shot of a code

Description automatically generated

Működés:

* A #container: Ez a fő tartalmi terület, amelybe betöltésre kerülnek az alábbi select elemek: Futam neve, illetve 20db pilóta nevekkel ellátott elem, amellyel a pilóták helyezési sorrendjét lehet kiválasztani.
* A dinamikus tartalom kezelését a result-uploader-script.js végzi, amely biztosítja a select elemek megjelenítését, a tartalom betöltését és az eredmények feltöltését az adatbázisba.

about.html

A screenshot of a computer

Description automatically generatedEz az oldal az a weboldal bemutatására, licenc-információkra és elérhetőségekre koncentrál.

Működés:

about-section: Az oldal teljes tartalmát tartalmazza, amely az alábbi egységekre tagolódik:

* Bevezető szöveg
* Kapcsolati szekció
* Licenc szekció
* Github szekció
* Hivatkozások szekció

#### Stílusfájlok

A weboldal megjelenésének megtervezésénél kiemelt célunk volt egy modern, felhasználóbarát és esztétikus dizájn kialakítása. Emellett fontosnak tartottuk, hogy a későbbi módosítások során a stílusfájlok átláthatóak és könnyen karbantarthatóak legyenek.

Stílusfájlok felosztása

A stílusfájlokat két csoportra osztottuk annak érdekében, hogy a közös elemeket és az egyedi megoldásokat könnyebben tudjuk kezelni:

* **Általános beállítások**: Ezek a stílusfájlok minden HTML oldal megjelenésében szerepet kapnak, biztosítva az egységes dizájnt.
* **Oldalspecifikus beállítások**: Ezek a stílusfájlok kifejezetten az egyedi igények szerinti megjelenést szabályozzák.

Mind erre azért van szükség, hogy az ugyan olyan kinézetű elemeket könnyebben betudjuk állítani, illetve későbbiekben akár módosítani.

##### Általános beállításokat tartalmazó Stílusfájlok

* **reset.css**: Minimalizálja a böngészők alapértelmezett stílusait, biztosítva az egységes megjelenést különböző platformokon.
* **typography.css**: A weboldal tipográfiai megjelenését szabályozza, beleértve egyedi betűtípusok és szövegformázások beállítását.
* **variables.css**: Tartalmazza a weboldal színét, betűtípusait és méreteit definiáló változókat, melyek megkönnyítik a stílus egységes változtatását.
* **buttons.css**: A gombok és hivatkozások megjelenését szabályozza, biztosítva a konzisztens dizájnt.
* **table.css**: A weboldalon megjelenő táblázatok egységes megjelenését kezeli.
* **footer.css**: Az egységes lábléc kialakításáért felel.
* **header.css**: Az egységes fejléc szekció megjelenését határozza meg.
* **nav.css**: A navigációs sáv stílusát szabályozza.
* **animista.css**: A weboldal animációinak meghatározására szolgál.
* **main.css**: Az oldal közös elemeinek megjelenését tartalmazza, beleértve a dropdown menük és egyéb egységes stílusok beállításait.
* **responsive.css**: A különböző képernyőméreteken történő problémamentes megjelenésért felel.

##### Oldalspecifikus beállításokat tartalmazó Stílusfájlok

Minden HTML fájl kapott egy saját stílusfájlt, amelyben az egyedi beállítások kerülnek szabályozásra:

* **about.css**
* **auth.css**
* **forum.css**
* **index.css**
* **news.css**
* **profile.css**
* **raceresult.css**
* **tracks.css**

Ezen felosztás segítségével a közös elemeket könnyebben kezelhetjük és módosíthatjuk a későbbiekben, miközben fenntartjuk a weboldal egységes és professzionális megjelenését.

#### JavaScript fájlok

header-slider.js

horizontal-scroll.js

result-uploader-renderer.js

animation.js

dropdown.js

footer-renderer.js

loader-script.js

multi-page-ui-manager.js

nav-auth-controller.js

result-uploader-script.js

standlist-redirect.js

### Backend

#### API végpontok

1. /news/race-schedule

Metódus: **GET**

Lekéri a versenynaptárat, beleértve az események dátumait és a pályainformációkat.

Lekérdezés:

A close-up of a computer screen

Description automatically generated

* + - rd és rn aliasokkal a racedates és a racenames táblákat összecsatlakoztatjuk.
    - Az ORDER BY használatával a raceNumber alapján növekvő sorrendbe rendezzük az eredményeket.
    - A lekérdezés célj a versenyek dátumainak és információinak lekérdezése a megfelelő táblákból.

1. /news/trackinfo

Metódus: **GET**

Lekéri a pálya információit a megadott pálya azonosító alapján.

Paraméterek: id (kötelező)

Lekérdezés



* + - A trackName és fullName mezőket kérdezi le a racenames táblából az adott id alapján.
    - A ? helyettesítő karakter az előkészített lekérdezés része, amely segít megelőzni az SQL injekciós támadásokat.

1. /news/circuitdatas

Metódus: **GET**

Lekéri a részletes pálya adatokat egy adott pálya azonosító alapján.

Paraméterek: id (kötelező)

Lekérdezés

* + - Az adott pálya részletes adatait kérdezi le a circuitdatas táblából az id alapján.
    - Az adatok tartalmazzák a pálya első versenyét, körök számát, hosszát, versenytávot, rekordot, a rekordot tartó pilótát és rekord évét.

1. /news/driverStandlist

Metódus: **GET**

Lekéri a pilóták listáját és azokhoz tartozó konstruktórokat.

Lekérdezés



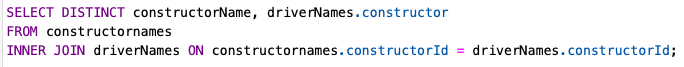
* + - Az összes pilóta ID-jét, nevét és konstruktorát kérdezi le a driverNames táblából.

1. /news/constructorStandlist

Metódus: **GET**

Lekéri az egyedi konstruktorokat és azok pilótáit.

Lekérdezés



* + - A constructornames és driverNames táblákat csatlakoztatja, hogy lekérdezze az egyedi konstruktor neveket és hozzájuk tartozó pilótákat.
    - Az INNER JOIN használatával a constructorId alapján kapcsolódnak össze a táblák.

1. /news/seasonRaceResults

Metódus: **GET**

Lekéri a teljes szezon versenyeredményeit.

Lekérdezés



* + - Az összes szezonbeli verseny eredményeit kérdezi le a seasonRaceResult táblából

1. /news/raceResults

Metódus: **GET**

Lekéri a részletes versenyeredményeket, beleértve a versenyzők pozícióit minden egyes versenyen.

Lekérdezés

A screenshot of a computer

Description automatically generated

* + - A seasonRaceResult, raceNames és driverNames táblákat csatlakoztatja, hogy részletes verseny eredményeket kapjon, beleértve a pilóta pozíciókat minden versenyen.
    - Az eredméyeket a raceNumber alapján növekvő sorrendve rendezi.
    - A LEFT JOIN használatával a pilótanevek csalatkoztatása biztosítja, hogy az eredményeknél üres érték esetén is szerepeljenek rekordok.

1. /news/seasonRaceResults

Metódus: **POST**

Elmenti a versenyeredményeket egy adott versenyhez

Minden mező egy számot tölt fel az adatbázisba.

Lekérdezés

* + - A versenyeredményeket mentéskor az összes pozíciót (P1-től P20-ig) egyetlen sorban tároljuk a seasonRaceResults táblában.
    - Az előkészített lekérdezés használatával a beviteli értékeket paraméterekkel (?) helyettesítjük a biztonságos adatkezelés érdekében.
    - Az értékeket egy tömbben (values) adjuk át, ahol minden pozíció vagy az eredmények értékét, vagy null-t kap, ha nincs érték megadva.

1. /news/forumTopics

Metódus: **GET**

Lekéri az összes fórum topikot.

Lekérdezés



* + - A forumTopics táblát a user táblával kapcsoljuk össze a userId alapján, hogy minden fórum topicchoz a felhasználónevet hozzárendeljük.
    - A lekérdezés visszaadja a topic azonosítóját, felhasználónevet, a felhasználó azonosítóját, a topic címét, tartalmát és létrehozás dátumát.

1. /news/forumTopics/:topicId

Metódus: GET

Lekéri az adott fórum topic részleteit a topicId alapján

topicId (kötelező)

Lekérdezés



* + - Az adott topicId alapján lekérdezi a fórum topic részleteit a forumTopics táblából, és kapcsolja össze a user táblával a felhasználónevekért.
    - Az előkészített lekérdezés használatával a topicId paraméterként van megadva.

1. /news/forumComments/:topicId

Metódus: GET

Lekéri az adott fórum topic hozzászólásait a topicId alapján

Paraméterek: topicId (kötelező)

Lekérdezés

KÉP

* + - Az adott topicId alalpján lekérdezi a fórum topic hozzászólásait a topicComments táblából, és kapcsolja a user táblával a felhasználónevekért.
    - Az előkészített lekérdezés használatával a topicId paraméterként van megadva.
    - Az INNER JOIN biztosítja, hogy csak olyan hozzászólások jelenjenek meg, amelyekhez kapcsolódik felhasználó.

1. /news/last-topicComment

Metódus GET

Lekéri egy adott fórum topic legutolsó hozzászólását.

Paraméterek: topicId (kötelező)

Lekérdezés

KÉP

* + - Az adott topicID alalpján lekérdezi a legutolsó hozzászólást a topicComments táblából.
    - A TIMESAMPDIFF függvényt használjuk a dátumok közötti különbség kiszámításához másodpercben, és az ABS függvénnyel abszolút értéket számítunk.
    - Az eredményt növekvő sorrendbe rendezzük, a különbségek alapján, és az első (legutóbbi) eredményt adjuk vissza a LIMIT 1 segítségével.

1. /news/loadReports

Metódus: GET

Lekéri a fórum témákra vonatkozó reportokat.

Lekérdezés:

KÉP

* + - Lekérdezi a reportokat a topicReports táblából és kapcsolja a user táblával a felhasználónevekért, valamit a forumtopics tábláva a topic címéért.
    - Az eredményeket dátum szerint csökkenő sorrendbe rendezzük (ORDER BY tr.date DESC), hogy legújabb reportok legyenek az elején.

1. /news/forumTopics/:topicId

Metódus: DELETE

Törli az adott fórum témát a topicId alapján

Paraméterek: topicId (kötelező)

Lekérdezés

KÉP

* + - Az adott topicId alalpján törli a fórum topicot a forumTopics táblából.
    - Az előkészített lekérdezés használatával a topicId paraméterként van megadva.
    - Ha nincs találat (az érintett sorok száma 0), akkor 404-es hibáz küld vissza „Topic not found” üzenettel.

1. /news/forumComments/:commentId

Metódus: DELETE

Letörli az adott fórum topic hozzászólását a commentId alapján.

Paraméterek: commentId (kötelező)

Lekérdezés:

KÉP

* + - Az adott commentId alapján törli a hozzászólást a topicComments táblából
    - Az előkészített lekérdezés használatával a commentId paraméterként van megadva.
    - Ha nincs találat (az érintett sorok száma 0), akkor 404-es hibát küld vissza „Comment not Found” üzenettel.
    - Sikeres törlés esetén 204 No Content státuszkódot küld vissza, jelezve a sikeres törlést.

1. /news/forumTopics

Metódus: POST

Új fórum topic létrehozása

Lekérdezés:

KÉP

* + - Egy új fórum topic létrehozása a forumTopics táblában.
    - Az előkészített lekérdezés használatával beviteli értékek maraméterekkel (?) helyettesítjók a biztonságos adatkezelés érdekében.
    - Az értékeket egy tömbben (values) adjuk át, ahol a userId, topicTitle, topicOntent és date értékek kerülnek beszúrásra.

1. /news/upload-comment

Metódus: POST

Új hozzászólás létrehozása egy adott fórum témához.

Lekérdezés:

KÉP

* + - Egy új hozzászólás létrehozása.
    - Az előkészített lekérdezés használatával beviteli értékek maraméterekkel (?) helyettesítjók a biztonságos adatkezelés érdekében.
    - Az értékeket egy tömbben (values) adjuk át, ahol a topicId, userId, CommentContent és date értékek kerülnek beszúrásra.
    - Hibakezelés biztosítása érdekében az err értékelése, és a megfelelő HTTP státuszkód visszaadása.

1. /news/report-topic

Metódus POST

Fórum téma jelentése

Lekérdezés:

KÉP

* + - Egy fórum téma jelentése a topicReports táblában.
    - Az előkészített lekérdezés használatával beviteli értékek maraméterekkel (?) helyettesítjók a biztonságos adatkezelés érdekében.
    - Az értékeket egy tömbben (params) adjuk át, ahol a userId, topicId és date értékek kerülnek beszúrásra.
    - Az authorize middleware biztosítja, hogy csak hitelesített felhasználók végezhessenek jelentést.

1. /news/register

Metódus: POST

Új felhasználó létrehozása

Lekérdezés:

* + - Felhasználó ellenőrzése:

KÉP

* + - Felhasználó regisztrálása:

KÉP

Magyarázat:

* + - Először ellenőrizzük, hogy a megadott felhasználónév vagy email cím már létezik-e a user táblában.
    - Ha nem létezik, akkor a jelszó hashelése után létrehozunk egy új felhasználót a user táblában.
    - Az új felhasználó autómatikusan bejelentkezik, és JWT tokent kap a válaszban.

1. /news/login

Metódus: POST

Felhasználó bejelentkezése.

Lekérdezés

KÉP

* + - A megadott email vagy felhasználónév alapján ellenőrizzük a felhasználót a user táblában.
    - Ha a felhasználó létezik, akkor a hashelt jelszót összehasonlítjuk a megadott jelszóval.
    - Ha a jelszó egyedik, akkor egy JWT tokent generálunk és visszaküldünk a válaszban.

1. /news/get-profile

Metódus: GET

Felhasználói profil lekérése.

Lekérdezés

KÉP

* + - Az adott felhasználó profiljának lekérése a user táblából az id alapján.
    - Az authorize middleware biztosítja, hogy csak hitelesített felahsználók férjenek hozzá a végponthoz.

1. /news/get-all-profiles

Metódus: GET

Az összes felhasználói proful lekérése.

Lekérdezés

KÉP

* + - Az összes felhasználói profil lekérése a user táblából.
    - Az authorize middleware biztosítja, hogy csak hitelesített felhasználók férjenek hozzá a végponthoz.

1. /news/update-permission

Metódus: PUT

Frissíti egy felhasználó jogosultsági szintjét.

Lekérdezés:

KÉP

* + - A felhasználó jogosultsági szintjét (permission) frissíti a user táblában az adott id alapján
    - Az authorize middleware biztosítja, hogy csak hitelesített felhasználók végezhessenek jogosultság frissítést.
    - Az előkészített lekérdezés használatával a permission és id értékeket paraméterekkel (?) helyettesítjük a biztonságos adatkezelés érdekében.

1. /news/update-username

Metódus: PUT

Frissíti a felhasználó felhasználónevét.

Lekérdezés

KÉP

* + - A felhasználó felhasználónevét (username) frissíti a user táblában az adott id alapján.
    - Az authorize middleware biztosítja, hogy csak hitelesített felhasználók végezhessenek felhasználónév frissítést.
    - Az előkészített lekérdezés használatával a username értéket paraméterekkel (?) helyettesítjük a biztonságos adatkezelés érdekében.

1. /news/update-email

Metódus: PUT

Frissíti a felhasználó e-mailcímét.

Lekérdezés

KÉP

* + - A felhasználó e-mail címét (email) frissíti a user táblában az adott id alapján.
    - Az authorize middleware biztosítja, hogy csak hitelesített felhasználók végezhessenek felhasználónév frissítést.
    - Az előkészített lekérdezés használatával a email értéket paraméterekkel (?) helyettesítjük a biztonságos adatkezelés érdekében.

1. /news/update-password

Metódus: PUT

Frissíti a felhasználó jelszavát.

Lekérdezés

KÉP

* + - A felhasználó jelszavát (password) frissíti a user táblában az adott id alapján.
    - Az authorize middleware biztosítja, hogy csak hitelesített felhasználók végezhessenek felhasználónév frissítést.
    - Az előkészített lekérdezés használatával a password értéket paraméterekkel (?) helyettesítjük a biztonságos adatkezelés érdekében.

1. /news/delete-account

Metódus: DELETE

Törli a felhasználói fiókot.

Lekérdezés

KÉP

* + - A felhasználói fiókot zörli a user táblából az adott id alapján.
    - Az authorize middleware biztosítja, hogy csak hiteleített felhasználók végezhessenek fiók törlést.
    - Az előkészített lekérdezés használatával az id értéket paraméterekkel (?) helyettesítjük a biztonságos adatkezelés érdekében.

#### Server.js

A kód egy Express.js alkalmazást definiál, amely egy Formula 1-es hírekkel, versenyekkel, fórumokkal és felhasználói interakciókkal kapcsolatos weboldalt szolgál ki. Az alkalmazás interakcióba lép egy MySQL adatbázissal és különböző API végpontokat biztosít a frontend számára. A kód célja a dinamikus tartalom kiszolgálása, felhasználói adatkezelés, valamint különböző funkciók, például fórumok versenyek és hírek kezelésére szolgáló API-k.

**Alapvető függőségek betöltésre és konfigurációk:**

* Az alkalamzás az express keretrendszert használja, amely lehetővé teszi a RESTful API-k gyors létrehozását.
* A mysql2 csomag biztosítja az adatbázis kapcsolódást, a body-parser a HTTP kérés törzsének olvasásához, a cors pedig a kereszt-domain lérések engedélyezésére szolgál.
* A bycryptjs a jelszavak titkosítására, a jsonwebtoken pedig az autentikációs tokenek kezelésére szolgál.
* A NodeCache egy memória alapú gyorsítótárat biztosít, amelyekben a hírek és technikai hírek tárolásra kerülnek, hogy elkerüljük az ismételt API hívásokat.

**Gyorsítótár frissítése:**

Az updateCache függvény két külső API-tol (newsapi.org) lekérdezi a híreket és technikai Formula-1 híreket, majd azokat elmenti a gyorsítótárba. Az adatokat óránként frissíti (setInterval(ipdateCache, 3600\*1000)), hogy mindig naprakész információkat szolgáltasson.

**A statikus fájlok kiszolgálása:**

Az express.static() segédlet segítségével az alkalmazás statikus fájlokat, például HTML, CSS és JS fájlokat szolgáltat a frontend számára. Ezek a fájlok a public könyvtárban találhatók.

**Felhasználói autentikáció:**

Az authorize middleware minden olyan API végpont előtt fut, amely autectikációt igényel. A felhasználó JWT tokenjét ellenőrzi, hogy biztosítsa a jogosultságokat, és hozzáférést biztosítson a privát adatokhoz.

API végpontok:

* Hírek és versenyek: Az alkalmazás API végpontokat biztosít a legújabb hírek, technikai hírek, versenyek időpontjai és eredményei lekérésére, beleértve a versenypályák információit, valamint a versenyeredményeket.
* *Fórumok kezelése:* Fórum témákat és kommenteket lehet lekérdezni, új fórumokat és kommenteket hozzáadni valamint törölni. A fórumok törléséhez és kommentek kezeléséhez admin jogosultság szükséges. Az API biztosítja a témák listázását, a kommentek lekérdezését és kommentek törlését is.
* *Felhasználói regisztráció és bejelentkezés*: A felhasználók regisztrálhatnak az alkalmazásba, a jelszavukat titkosítva tárolja. Bejelentkezéskor JWT tokent generál, hogy biztosítsa az autentikált kéréseket. A felhasználói adatok, például a név, e-mail és jelszó is frissíthetőek, ha a felhasználó jogosultsággal rendelkezik.
* *Adminisztrációs funkciók:* Az adminisztrátorok képesek a felhasználói jogosultságok módosítására (pl. Adminisztrátori jogok adása), valamint a fórumok és kommentek törlésére. Továbbá, az admin felhasználók hozzáférhetnek a jelentett fórumtémák listájához és azokkal kapcsolatos műveletekhez.

Adatbázis műveletek:

* Az adatbázisba történő lekérdezések (pl. Versenyek, eredmények, fórumok) a queryDB segédlettel történnek, amely egyszerűsíti az SQL lekérdezések végrehajtását.
* A fórum témák és kommentek hozzáadásához a POST végpontok kezelik, míg a törlés a DELETE végpontoknál történik.
* Az eredmények mentése és frissítése is SQL lekérdezésekkel történik.

Felhasználói adatkezelés és profilok:

* A felhasználói profilok, mint például a felhasználónév, e-mail cím és jelszó, frissíthetőek.
* A felhasználói adatokat és jogosultságokat biztonságosan tárolja és kezeli a rendszer, az adminisztrátorok pedig képesek más felhasználók jogosultságait módosítani.

**Törlés és biztonság:**

* Az alkalmazás biztonsági intézkedéseket tartalmaz, például jelszavak titkosítását (bcryptjs), valamint JWT alapú autentikációt.
* A fórumtémák és kommentek törléséhez a felhasználónak megfelelő jogosultságokkal kell rendelkeznie.

#### auth-api-manager.js

#### dropdown-api-manager.js

#### forum-api-manager.js

#### news-api-manager.js

#### race-result-api-manager.js

#### schedule-api-manager.js

#### standlist-api-manager.js

#### tracks-api-manager.js

## Stats (statisztikai oldal)

### Frontend

#### HTML Fájlok

#### Stílusfájlok

#### JavaScript fájlok

### Backend

#### API végpontok

#### Server.js

## Források