Konzulens tanár: Készítette:

Lakatos Sándor Ármós Szabolcs

Berki Balázs Gáspár Marianna Dominika

Sankó Balázs

ScoreSchool projekt szakdolgozat

Tartalom

[Bevezető 3](#_Toc194230033)

[Weboldal felépítése 4](#_Toc194230034)

[Főoldal 4](#_Toc194230035)

[Keresés 5](#_Toc194230036)

[Adatkezelés 5](#_Toc194230037)

[Statisztika 6](#_Toc194230038)

[Tervezés 7](#_Toc194230039)

[Photoshop 7](#_Toc194230040)

[Canva 8](#_Toc194230041)

[Draw.io 8](#_Toc194230042)

[Technológiai megvalósítás 9](#_Toc194230043)

[Adatbázis 9](#_Toc194230044)

[Backend – Node.js 11](#_Toc194230045)

[Frontend - Angular 12](#_Toc194230046)

[Programtervezési minta – MVC 13](#_Toc194230047)

[Bootstrap 13](#_Toc194230048)

[Verziókezelés 15](#_Toc194230049)

[Applikáció 15](#_Toc194230050)

[Tesztelés 15](#_Toc194230051)

[Tapasztalatok, csapatmunka és jövőkép 16](#_Toc194230052)

[Ábra jegyzék 19](#_Toc194230053)

[Felhasznált irodalom 20](#_Toc194230054)

# Bevezető

Projektünk ötletének alapja az, hogy mindannyian lelkes futballrajongók vagyunk és Szabolcs és Balázs aktívan űzik a sportot. A hazai és nemzetközi bajnokságokat is rendszeresen követjük, amelyek eredményeit több online weboldalon is megtalálhatjuk. Ezek mind információkat szolgálnak a meccsekről, csapatokról és játékosokról. Innen jött az ötlet egy olyan weboldalhoz, amely kifejezetten az iskolai tornák adatait tárolná.

Iskolánkban, a Petőfi Sándor Technikumban, régóta hagyomány a Falcsik Ferenc Emléktorna, amelyet néhai testnevelés tanárunkról, Falcsik Ferencről neveztek el. A tornára bárki jelentkezhet saját csapattal az iskola tanulói közül, így akár osztályok vagy vegyes csapatok is nevezhetnek. Ez remek lehetőség arra, hogy a diákok összemérjék tudásukat és versenyezzenek különböző díjakért (legjobb kapus, gól király, legjobb játékos). A torna január elején kezdődik és március elejéig tart. Minden szerdán 2-2 csapat játszik egymás ellen és a torna során mindegyik csapat játszik mindegyik csapattal. A meccsek alapján pontokat gyűjthetnek (győzelem: 3, döntetlen: 1, vereség: 0), amelyek alapján a torna végén rangsorba kerülnek. Ha két csapat ugyan annyi pontot szerzett, akkor vagy az dönt, hogy az egymás ellen játszott meccsen ki nyert vagy (döntetlen esetén) az, hogy kinek jobb a gólaránya (kapott gólokból kivonjuk a rúgott gólokat). Viszont az eredmények vezetése eddig papír alapon történt, amelynek több hátránya is van:

1. Az adatok bármikor elveszhetnek, hiszen csak egy lapon szerepelnek.
2. Az eredmények kézi vezetése sok időt igényel.
3. Nehézségeket okozhat a meccsek vezetése és ezekből kimutatásokat készíteni (például: ki lett a gólkirály, körbeverés esetén melyik csapat rendelkezik a legtöbb ponttal, vagy melyiknek jobb a gólaránya).
4. A tornák eredményei csak a lebonyolítást végző tanárnak vagy bírónak vannak meg, így a diákok nem nézhették vissza teljesítményüket.

Weboldalunk nem csak egyszerűsíti, de korszerűsíti a torna szervezését. A weboldal nem csak arra nyújt lehetőséget, hogy saját tornáink adatait mentésük, hanem más iskolák eseményeit megtekinthetjük, így egy olyan rendszer jönne létre, ahol összekapcsolja a különböző iskolák sportéletét. A weblapunk lehetőséget nyújt statisztikai kimutatások készítésére is. Például, hogy melyik játékos rúgta a legtöbb gólt, melyik csapat rendelkezik a legtöbb ponttal, gólaránnyal vagy hány kapott és rúgott góljuk van. A kimutatások növelhetik a versenyszellemet, hiszen így a diákok folyamatosan nyomon követhetik a fejlődésüket.

Összegezve, weboldalunk nem csak a tornák vezetésére nyújt korszerű megoldást, de a diákokat is jobb teljesítményre ösztönzi. Ezenkívül olyan közöl platformot biztosít, ahol nemcsak saját, de más iskolák sporteseményeit is követhetjük.

# Weboldal felépítése

Weboldalunk 1 főoldalból és 3 aloldalból áll, amelyek mind más funkciókkal rendelkeznek. Az aloldalak csak bejelentkezés után érhetők el, és a navigációs sáv segítségével tudunk közöttük egyszerűen váltani.

## Főoldal

A weboldal megnyitásakor először a főoldal köszönti a felhasználót. A weblap alján található sávon a felhasználó megismerkedhet a legfontosabb funkciókkal. A „Gyere kezdjünk” gombra kattintva lehetősége van bejelentkezni, vagy ha még nem rendelkezik fiókkal, akkor regisztrálhat. A reszponzív dizájnnak köszönhetően oldalunk nem csak számítógépen, hanem tableten és mobilon is elérhető és használható.

A képen szöveg, képernyőkép, labda, személy látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

1. ábra

## Keresés

A keresés lap lehetőséget nyújt a felhasználónak arra, hogy az adatbázisban szereplő adatok között böngésszen. A keresés 3 kategória szerint lehetséges: torna, csapat, játékos. A felhasználók egy legördülő listából választhatják ki melyik kategória adataira kíváncsiak és a weblap azonnal frissül e szerint. A keresőmező arra szolgál, hogy egy konkrét adatra keressünk a neve alapján.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

2. ábra

## Adatkezelés

Bejelentkezés után erre a felületre navigálja át az oldal a felhasználót, ahol az általa feltöltött adatokat találhatja. Itt hozhat létre új tornákat, rögzítheti a meccsek, csapatok, csoportok vagy játékosok eredményeit, vagy a meglévő adatokat módosíthatja és törölheti. A weblap a műveletek után azonnal frissül. Az adatok kezelését egy REST API-n keresztül működő backend oldali szerver biztosítja, amely aszinkron módon egyszerre több kérésre is tud reagálni.

A képen képernyőkép, szöveg, Multimédiás szoftver, Grafikai szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

3. ábra

## Statisztika

Ez az oldal szolgál a statisztikai kimutatások készítésére. A felhasználó az általa feltöltött tornák közül választva jelenítheti meg a megfelelő adatokat, majd kiválaszthatja, hogy pontosan milyen elemzést szeretne készíteni, például, hogy a csapatok pontjait, gólarányait, kapott vagy rúgott góljait szeretné megjeleníteni vagy a játékosok között szeretne egy ranglistát gólok alapján. A weboldal az adatokat egy oszlop diagrammon jeleníti meg, amelyet a felhasználó .svg (skálázható vektorgrafika) kiterjesztésben letölthet, így az minden méretben jól olvasható és éles.

A képen képernyőkép, szöveg, diagram, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

4. ábra

# Tervezés

## Photoshop

A weboldalunkon található képek szerkesztését Photoshoppal végeztük el. A képek többsége olyan weboldalakról származnak, amelyek szerzői jogdíjmentes képeket kínálnak (például Undraw). Ahhoz, hogy minden eszközön tökéletesek legyenek a grafikák, törekedtünk arra, hogy .svg formátumban használjuk őket. Amelyeknél erre nem volt lehetőség azt Photoshop használatával átméreteztük és eltávolítottuk a hátterüket. Logónk is ebben az alkalmazásban készült, amelyet először digitálisan megrajzoltunk, majd átalakítottuk .svg-vé.

A képen szöveg, képernyőkép, Multimédiás szoftver, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

5. ábra

## Canva

A weboldal megtervezéséhez a Canva grafika tervezési weboldalt használtuk, amelyen nem csak a különböző elemek elhelyezkedését próbálhattuk ki, hanem különböző színeket és betűtípusokat is. Ennek köszönhetően gyorsan és egyszerűen tervezhettük meg a weboldalunk elemeinek elrendezését.

## Draw.io

Az adatbázisunkról külön dokumentációt készítettünk, amelyhez nemcsak UML (Unified Modeling Language – Általános célú modellező nyelv), de ER (Entity-Relationship model – Egyed-kapcsolat modell) diagrammot is készítettünk. Ennek köszönhetően adatbázisunk és a táblák közötti kapcsolat könnyen értelmezhető és jól olvasható.

A képen szöveg, diagram, rajz, vázlat látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

6. ábra

# Technológiai megvalósítás

## Adatbázis

Adatbázisunk elkészítésekor a figyelembe vettük a normalizáció szabályait, így tudatosan elkerültük a felesleges adat ismétlődést, a null értékeket és minden adattábla saját primary key-el rendelkezik.

A képen szöveg, képernyőkép, diagram, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

7. ábra

## Backend – Node.js

Brendan Eich fejlesztette ki 1996-ban, Mocha néven, majd LiveScript lett. A JavaScript nevet később kapta a Java programozási nyelvről.

A szerveroldal elkészítéséhez olyan modulokra volt szükségünk, mint az Express, CORS, Body-parser és MySQL2.

* Express: webkeretrendszer Node.js számára, amely segít az API-k fejlesztésében. Megkönnyíti a REST API-k készítését és támogatja a middleware-eket.
* CORS: Node.js middleware, ami lehetővé teszi a Cross-Origin Rescource Sharing-t, így a weboldal hozzáférhet majd a szerver által lekért adatokhoz.
* Body-parser: olyan middleware, amely a HTTP-kérések body-ját képes JSON formátumban feldolgozni.
* MySQL2: MySQL kliensmodul Node.js számára.

A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

8. ábra

A megírt szerver tökéletes volt a mobil applikációnak számára is, így mind a kettő ugyan azt a kódot használja. Az API végpontok működésének tesztelésére először a ThunderClient bővítményt használtuk. A teszteket kollekcióként mentettük . kiterjesztésben.

## Frontend - Angular

2016-ban vezették be a TypeScript nyelvet.

A TypeScriptet a Microsoft fejlesztette ki 2010-ben és 2012-ben tette nyílt forráskódúvá. Gyakran „JavaScript kiterjesztés” vagy „JavaScript típusrendszerrel” néven emlegetik. Viszont a TypeScript ennél sokkal több.

* Az Angular komponensekre épül, amik független, újrahasználható kódmodulok. Külön-külön tartalmaznak HTML-t, CSS-t és TypeScriptet is.
* A servicek a weboldalunk logikai részének kezelésére szolgálnak. Olyan feladatokat végeznek, amelyeket nem egy-egy komponens hajt végre közvetlenül, hanem egy központi helyen, például az adatlekérés. Így ez minden komponens számára egyaránt elérhető.

A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

9. ábra

A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

10. ábra

* Modulok: az Angular modulok köré épül, amik logikailak csoportosítják a különböző komponenseket, szolgáltatásokat és egyéb funkciókat. Minden alkalmazás legalább egy modult, a gyökérmodult tartalmazza.

## Programtervezési minta – MVC

Felhasználó → View → Controller → Model → View → Felhasználó

## Bootstrap

Az oldal reszponzivitását Bootstrap-el oldottuk meg, ami egy nyílt forráskódú frontend keretrendszer tervezéshez és fejlesztéshez. A rács szerkezete kiváló bármilyen projekt számára, hiszen a szükséges CSS osztályok előre definiálva vannak 6 különböző méretben (xs, sm, md, lg, xl, xxl). A weboldalunk monitorra, tabletre és telefonra lett optimalizálva.

A képen szöveg, képernyőkép, labda, személy látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

11. ábra

A képen szöveg, képernyőkép, Betűtípus, tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen. A képen szöveg, labda, sportfelszerelés, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

12. ábra

## Verziókezelés

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

13. ábra

# Applikáció

Az alkalmazást Expo segítségével készítettük, amely egy nyíltforráskódú platform. JavaScript és React alapú natív mobilalkalmazásokat készíthetünk vele.

Az appunk adatbázisa és backendoldali szervere megegyezik a weboldaléval. Ennek a legfőbb oka az, hogy a projekt során ne kelljen két különböző adatbázist kezelni és hogy az adatok valós időbe frissüljenek bármelyik platformon is változtatunk rajta.

Az alkalmazásunk frontendje React Native és az Expo keretrendszer segítségével készült.

A főoldal ismerteti a weboldal fő funkcióit.

* Bejelentkezhetünk meglévő fiókunkba vagy újjal regisztrálhatunk.
* A keresőoldalon a legördülő listából kiválasztva a kategóriát szűrhetjük az adatokat, vagy a kereső mezővel pontosan kereshetünk bárkire vagy bármire.

# Tesztelés

A projektünk white box teszteléséhez a Jest-et alkalmaztunk. A Jest egy univerzális tesztkeretrendszer, ami képes bármilyen JavaScript könyvtárhoz vagy keretrendszerhez igazodni. Kiváló backend és frontend tesztelésre is, hiszen képes a weboldal főbb funkcióin kívül az API végpontokat is egyszerűen tudtuk tesztelni a használatával. Támogatja az egység és az aszinkron teszteket is.

A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

14. ábra

A frontend tesztelésekor leginkább az egyes funkciók tényleges működését akartuk ellenőrizni. A projekt létrehozásakor az automatikus generált az egyes componensekhez és servicekhez külön .spec.ts fájlt, ezeket használtuk fel a tesztek megírásakor. A servicek esetén minden eljárást külön teszteltünk le, hogy megbizonyosodjunk róla, megkapja az adatokat a backend szervertől. A componensek esetén pedig a célunk az volt, hogy megvizsgáljuk, minden funkció sikeresen végrehajtódik ha az elvárt adatokat kapja és hiányos vagy rossz adat esetén, ne hajtódjon végre.

# Tapasztalatok, csapatmunka és jövőkép

A ScoreSchool fejlesztése során az egyik legnagyobb kihívás számunkra az volt, hogy hogyan osszuk fel és hangoljuk össze a munkát hármunk között. A munkamegosztás alapja a képességeink és érdeklődési köreink voltak. A csoportunk tagjai mind másban jók és más részeket tartanak számukra közelinek így nem volt kérdéses ki milyen feladatot vállal. Dominika feladata volt a backend létrehozása, Szabolcs a frontendet készítette el, míg Balázs a mobil applikációt valósította meg. A munka során voltak olyan részek is, amelyeket közösen dolgoztunk ki, ilyen volt például az adatbázis. Mindhárman külön tervet készítettünk a szerintünk legoptimálisabb adatbázisról, majd ezeket egy Daily Scrum során átnéztük és megvitattuk melyik és miért lenne a legjobb a projektünk számára.

A tanórák keretein belül a tanáraink lehetőséget adtak 15 perces Daily Scrum meetingekre. Ilyenkor megbeszélhettük ki hol tart, milyen problémába ütközött vagy éppen milyen új ötlete támadt az előző nap. Ezek a megbeszélések elengedhetetlenek voltak a fejlesztés során, hiszen ilyenkor a tanáraink is ráláttak a projektünk és segítettek, ha mi nem is tudtunk egymásnak.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

15. ábra

A feladatok beadási dátumát és sorrendjét a Trello weboldalán tudtuk nyomon követni.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

16. ábra

# Ábra jegyzék

1. ábra: Főoldal
2. ábra: Keresés
3. ábra: Adatkezelés
4. ábra: Statisztika
5. ábra: Logónk a Photoshop alkalmazásba
6. ábra: Az adatbázis ER diagramja
7. ábra: Az adatbázis UML diagramja
8. ábra: Backend oldali szerver GET Api végpontja
9. ábra: adatok.service.ts fájl
10. ábra: app-routing.module.ts fájl
11. ábra: weblap megjelenése LG (monitor) méretben
12. ábra: weblap megjelenése MD és SM (tablet és telefon) méretben
13. ábra: a projekt github repository-jának felépítése
14. ábra: get.test.js fájl
15. ábra: Daily Scrum dokumentációja (2025.02.19)
16. ábra: a projekt feladatainak követése a Trello weboldalán

# Felhasznált irodalom

1. GOOGLE, LLC. Express dokumentáció, Elérhetőség: <https://expressjs.com/>
2. GOOGLE, LLC. CORS dokumentáció, Elérhetőség: <https://www.npmjs.com/package/cors>
3. GOOGLE, LLC. Body-parser dokumentáció, Elérhetőség: <https://www.npmjs.com/package/body-parser>
4. GOOGLE, LLC. MySQL2 dokumentáció, Elérhetőség: <https://www.npmjs.com/package/mysql2>
5. GOOGLE, LLC. React Native, Elérhetőség: <https://reactnative.dev/>
6. GOOGLE, LLC. Expo dokumentáció, Elérhetőség: <https://docs.expo.dev/>
7. <https://learning.oreilly.com/library/view/node-js-web-development/9781838987572/>
8. David Herron *Node.js Web Development* 2020. Elérhetőség: <https://learning.oreilly.com/library/view/node-js-web-development/9781838987572/>
9. Aristeidis Bampakos *Learning Angular – Fifth Edition* 2025. Elérhetőség: <https://learning.oreilly.com/library/view/learning-angular/9781835087480/>
10. Brett Jephson, Lewis Coulson, Ana Carolina Silveira *Practical HTML and CSS – Second Edition* 2024. Elérhetőség: <https://learning.oreilly.com/library/view/practical-html-and/9781835080917/>
11. Anna Skoulikari *Learning Git* 2023. Elérhetőség: <https://learning.oreilly.com/library/view/learning-git/9781098133900/>
12. Josh Goldberg *Learning TypeScript* 2022. Elérhetőség: <https://learning.oreilly.com/library/view/learning-typescript/9781098110321/>