Konzulens tanár: Készítette:

Lakatos Sándor Ármós Szabolcs

Berki Balázs Gáspár Marianna Dominika

Sankó Balázs

ScoreSchool projekt szakdolgozat

Tartalom

[Bevezető 3](#_Toc193948155)

[Weboldal felépítése 4](#_Toc193948156)

[Főoldal 4](#_Toc193948157)

[Keresési felület 5](#_Toc193948158)

[Adatkezelési panel 5](#_Toc193948159)

[Statisztikai modul 6](#_Toc193948160)

[Tervezés 7](#_Toc193948161)

[Photoshop 7](#_Toc193948162)

[Canva 7](#_Toc193948163)

[Draw.io 7](#_Toc193948164)

[Technológiai megvalósítás 8](#_Toc193948165)

[Adatbázis 8](#_Toc193948166)

[Backend – Node.js 11](#_Toc193948167)

[Frontend - Angular 12](#_Toc193948168)

[Programtervezési minta – MVC 15](#_Toc193948169)

[Bootstrap 16](#_Toc193948170)

[Verziókezelés 17](#_Toc193948171)

[Applikáció 17](#_Toc193948172)

[Tesztelés 18](#_Toc193948173)

[Tapasztalatok, csapatmunka és jövőkép 19](#_Toc193948174)

[Ábra jegyzék: 23](#_Toc193948175)

[Felhasznált irodalom: 24](#_Toc193948176)

# Bevezető

Az ötlet alapja csapatunk közös érdeklődési köre, mivel mindannyian lelkes futballrajongók vagyunk, és többen aktívan játsszuk is ezt a sportot. Szabadidőnkben rendszeresen követjük a hazai és nemzetközi bajnokságokat, és ehhez gyakran használjuk az eredmények.com weboldalt, amely átlátható módon biztosít információkat a mérkőzésekről, statisztikákról és csapatokról. Innen jött az ötlet, hogy egy hasonló elven működő, de kifejezetten iskolák számára készült webalkalmazást hozzunk létre. Célunk egy olyan rendszer kialakítása volt, amely nemcsak egyszerűbbé teszi az iskolai sportesemények nyilvántartását, hanem motiválja is a diákokat a sportolásra és a versengésre.

A projekt középpontjában egy olyan weboldal áll, amely az iskolai futballbajnokságok szervezését és lebonyolítását segíti. Iskolánkban, a Petőfi Sándor Technikumban, hosszú hagyománya van a Falcsik Ferenc Emléktornának, amelyet néhai testnevelő tanárunk emlékére rendeznek meg minden évben. A torna lehetőséget biztosít a diákok számára, hogy csapataikkal nevezzenek, és az osztályok közötti versenyen összemérjék tudásukat. Bár az esemény évek óta népszerű, a szervezés és az eredmények rögzítése eddig hagyományos, papíralapú módszerekkel történt, ami sok problémát okozott. Az adatok könnyen elveszhettek, a jegyzőkönyvek megsérülhettek, és az eredmények hosszú távú tárolása is nehézkes volt. Ezért született meg az ötlet, hogy egy digitális rendszerrel egyszerűsítsük és korszerűsítsük a torna szervezését.

A weboldal egyik legfontosabb funkciója, hogy nemcsak az adott iskola bajnokságainak adatait tárolja, hanem lehetőséget biztosít más iskolák tornáinak megtekintésére is. Így egy olyan rendszer jöhet létre, amely összekapcsolja a különböző intézmények sportéletét, és segíti a diákokat abban, hogy átfogóbb képet kapjanak az iskolai versenyekről. A rendszer statisztikai elemzéseket is kínál, amelyek segítenek nyomon követni a játékosok és csapatok teljesítményét. Például megmutathatja, hogy melyik játékos rúgta a legtöbb gólt, melyik csapat érte el a legjobb eredményt, vagy melyik osztály bizonyult a legeredményesebbnek egy adott szezonban. Ezek az elemzések nemcsak a verseny izgalmát növelik, hanem motiválják is a diákokat, hiszen így folyamatosan nyomon követhetik fejlődésüket.

A weboldal további előnye, hogy a korszerű technológiai megoldásokat kihasználva hatékonyabbá teszi az adatkezelést. A felhőalapú tárolás biztosítja, hogy az információk mindig elérhetőek maradjanak, és ne vesszenek el. Emellett a reszponzív kialakítás lehetővé teszi, hogy a diákok és tanárok mobileszközökről is könnyedén elérjék az oldalt, így bárhol és bármikor nyomon követhetik a bajnokság alakulását. Összességében projektünk nemcsak egy praktikus megoldást nyújt az iskolai sportesemények adminisztrációjára, hanem hozzájárul a diákok sport iránti lelkesedésének növeléséhez is. Emellett egy olyan platformot teremt, amely elősegíti az iskolák közötti együttműködést és közösségépítést. A Falcsik Ferenc Emléktorna digitalizálásával egy új korszakot nyitunk iskolánk sportéletében, és bízunk benne, hogy más intézmények számára is hasznos példát mutathatunk arra, hogyan lehet a technológiát a sport szolgálatába állítani.

# Weboldal felépítése

Weboldalunk négy fő pillérre épülő struktúrával rendelkezik, amelyek biztosítják a rendszer hatékony működését és könnyű használhatóságát. Az alábbiakban részletesen bemutatjuk ezeket, kiemelve a technikai megoldásokat és az alkalmazott funkciókat.

## Főoldal

Ez az oldal az elsődleges belépési pont a felhasználók számára. Itt egy átlátható és könnyen kezelhető felületen ismerkedhetnek meg a rendszer legfontosabb funkcióival. A főoldalon lehetőség van bejelentkezésre egy meglévő fiókkal, illetve regisztrációra egy gyorsan kitölthető űrlapon keresztül. A reszponzív dizájn biztosítja, hogy az oldal számítógépen, tableten és mobilon egyaránt kényelmesen használható legyen. A navigáció egyszerű és felhasználóbarát, így minden látogató könnyen eligazodhat rajta.

A képen szöveg, képernyőkép, labda, személy látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

1. ábra

## Keresési felület

Ez a modul lehetőséget biztosít a felhasználók számára, hogy könnyen böngésszenek az adatbázisban és gyorsan megtalálják a számukra releváns információkat. A keresés három fő kategória szerint történhet: tornák, csapatok és játékosok. A felhasználók egy legördülő menü segítségével választhatják ki, hogy melyik kategóriában szeretnének keresni, és a rendszer azonnal betölti a megfelelő adatokat. Emellett egy keresőmező is rendelkezésre áll, amely kulcsszavas keresést tesz lehetővé, így a találatok gyorsan és pontosan megjelennek.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

2. ábra

## Adatkezelési panel

Bejelentkezés után a felhasználók erre a felületre kerülnek, amely egy teljes körű adminisztrációs eszközt biztosít számukra. Itt lehetőség van új tornák, csapatok, játékosok és mérkőzések rögzítésére, meglévő adatok módosítására vagy törlésére. A rendszer valós idejű visszajelzést ad a műveletekről, és az adatkezelés aszinkron módon történik, így a felhasználók zökkenőmentesen végezhetik el a szükséges módosításokat. Az adatok kezelését egy korszerű, RESTful API-n keresztül működő háttérrendszer támogatja, amely biztosítja a gyors és biztonságos adatkommunikációt.

A képen képernyőkép, szöveg, Multimédiás szoftver, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

3. ábra

## Statisztikai modul

Ez a rész az adatok vizuális megjelenítésére szolgál, interaktív grafikonok segítségével. A felhasználók kiválaszthatják az elemzéshez szükséges adatokat, például egy adott torna eredményeit vagy a játékosok teljesítményét, és ezek alapján különböző statisztikákat generálhatnak. A rendszer az információkat könnyen érthető diagramok és grafikonok formájában jeleníti meg, amelyek letölthetőek vektorgrafikus formátumban, így bármilyen eszközön éles és jól olvasható marad a megjelenítés.

A képen szöveg, képernyőkép, diagram, Grafikai szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

4. ábra

Összességében weboldalunk olyan átgondolt struktúrára épül, amely nemcsak megkönnyíti a sportesemények szervezését és nyomon követését, hanem intuitív és modern felhasználói élményt is biztosít minden látogató számára.

# Tervezés

## Photoshop

A felhasznált képek szerkesztését Photoshoppal végeztük el. A képek kiválasztásakor fontos tényező volt számunkra, hogy azok szerzői jogdíjmentesek legyenek, így a legtöbb képünk olyan weboldalról származik, ahonnan jogtisztán használhatjuk fel őket (Undraw). Használatuk előtt átméreteztük, vagy éppen retusáltuk őket, hogy tökéletesen illeszkedjenek a weboldalunk kinézetéhez. A Photoshop használatával készült el logónk is, melyet digitálisan megrajzolva és szerkesztve később a jó skálázhatóság érdekében svg fájllá alakítottunk.

## Canva

A weboldal kinézetét megtervezéséhez a Canva grafika tervezési weboldalt használtuk. Így gyorsan és egyszerűen készíthettünk terveket, majd pedig a weboldal végső designját. A weboldal lehetőséget nyújtott arra, hogy kipróbáljunk különböző színeket, betűtípusokat és elrendezéseket.

## Draw.io

Az adatbázis felépítésének megtervezéséhez a draw.io (diagrams.net) online diagramkészítő platformot alkalmaztuk. Az ER-diagram elkészítésével az entitásait, attribútumait és kapcsolatait akartuk ábrázolni, hogy az adatbázis jól olvasható és könnyen értelmezhető legyen általa.

A képen szöveg, diagram, rajz, vázlat látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

5. ábra

# Technológiai megvalósítás

A projekt fejlesztése során olyan modern technológiákat és eszközöket alkalmaztunk, amelyekkel már korábban is dolgoztunk, így biztosítva a hatékony fejlesztési folyamatot és a rendszer stabil működését.

## Adatbázis

Az adatok tárolására egy MySQL alapú relációs adatbázist használunk, amelyet a phpMyAdmin segítségével terveztünk meg, készítettünk el és kezeltünk.

Az SQL (Structured Query Language – strukturált lekérdezőnyelv) relációsadatbázis-kezelők (például: Oracle, MariaDB, MySQL) lekérdezési nyelve. A relációs adatmodell annyit jelent, hogy az SQL a matematikában lévő halmaz műveleteken alapú. Az adatok adathalmazokban vannak, amikből szűrűnk halmaz műveltekkel (unió, metszet, különbség).

A MySQL az egyik legnépszerűbb adatbázis-kezelő rendszer webszerverek számára, amelyet 1990-es évek közepén fejlesztettek ki. Sikerének oka, hogy ingyenesen használható, erős és gyors.

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.

6. ábra

## Backend – Node.js

A backend fejlesztésére Node.js-t használtunk, amely egy nyílt forráskódú, Javascript alapú futtatókörnyezet. A Node.js aszinkron I/O műveleteket alkalmaz, így azok nem blokkolják a végrehajtást, és egyszerre több kérést is tud a szerver kezelni. A kód esemény vezérelt modellre épül, ami miatt képes kérésekre reagálni és visszahívásokat használni.

A JavaScript egy objektumorientált programozási nyelv, a legelterjedtebb szkriptnyelv, amelyet a weboldalak interaktív funkcióinak megvalósítására használnak. Lehetővé teszi a felhasználói interakciókat, az adatok manipulálását és a weboldalak dinamikus frissítését. A JavaScript kód vagy a HTML fájlban a <body> tagen belül található, vagy külön .js fájlban van. Brendan Eich fejlesztette ki 1996-ban, Mocha néven, majd LiveScript lett. A JavaScript nevet később kapta a Java programozási nyelvről.

A szkriptnyelvek olyan nyelvek, amelyeket gyorsan végrehajtható feladatokra, például automatizálásra vagy weboldalak működtetésére használnak. Nem kell előre lefordítani, hanem egy program vagy a böngésző (értelmező) azonnal végrehajtja.

A Node.js-hez tartozik az NPM (Node Package Manager), a legnagyobb nyílt forráskódú csomagkezelő is. Ezen keresztül könyvtárakat és modulokat telepíthetünk, amelyek nagyban segítik a projekt működését, fejlesztését és nagy előnyt jelent az, hogy a keretrendszer a Google Chrome V8 JavaScript-motorjára épül. 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. Támogatja az aszinkron lekéréseket és kapcsolatkezelést

A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

7. ábra

A backend egy RESTful API-n keresztül biztosít hozzáférést az adatbázishoz, amely HTTP-metódusok (GET, POST, PUT, DELETE) segítségével működik. Az API dinamikusan kezeli az adatokat, így minimalizálva a kód redundanciáját. 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

A felhasználói felület az Angular keretrendszerre épül. Az Angular egy TypeScript alapú webes alkalmazásfejlesztő keretrendszer, amelyet a Google fejlesztett. Az első verziója 2012-ben jelent meg, az AngularJs, egy JavaScript-alapú keretrendszer. 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. Egyszerre programozási nyelv, típusellenőrző, fordító és nyelvi szolgáltatás.

* Programozási nyelv: tartalmazza a meglévő JavaScript szintaxist és a TypeScript szintaxist is a típusok meghatározására és használatára.
* Típusellenőrző: képes JavaScript vagy TypeScript nyelven írt fájlokat értelmezni (a változóit és funkcióit) és figyelmeztet, ha esetleg a felépítés helytelen
* Fordító: lefuttatja a típusellenőrzést, jelzi a hibákat és előállítja a JavaScript megfelelőjét a kódnak.
* Nyelvi szolgáltatás: a típusellenőrzést használva a szerkesztők, mint a Visual Studio Code hasznos fejlesztői funkciókat tudnak ajánlani a felhasználóknak.

Az Angular keretrendszer egy olyan platform, amely mindent tartalmaz, ami szükséges egy webalkalmazás fejlesztéséhez és telepítéséhez.

Főbb jellemzői

* Az Angular komponensekre épül, amik független, újra felhasználható 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 projektünkben a service-ek segítségével kérdezzük le a backendtől az adatokat, és dinamikusan jelenítjük meg azokat a felhasználói felületen.

A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

8. ábra

* A routing lehetővé teszi az oldalak közötti navigációt anélkül, hogy bármikor újra töltődne a weboldalunk, így tökéletes SPA (Single Page Application) készítésére.

A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

9. 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

A projektünk az MVC (Model-View-Controller) tervezési mintát követi. Szétválasztja az adatot (adatbázis) és a felhasználói felülethez (frontend) tartozó dolgokat. Így a nézet változtatásai nem befolyásolják a modell és fordítva. Ezt úgy éri el, hogy elkülöníti az adatok elérését és az üzlet logikát az adatok megjelenítésétől és a felhasználói interakciótól a vezérlő bevezetésével. A vezérlő (a backend) az eseményeket, a felhasználói műveleteket dolgozza fel és válaszol rájuk, illetve a modellt is megváltoztatja. 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.

10. 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.

11. ábra

## Verziókezelés

A verziókezeléshez a GitHub-ot használtuk, ami egy Git-re épülő platform. A Git egy olyan technológia, amellyel a projektünk változásait követni tudjuk és ami lehetővé teszi, hogy egyszerre több ember dolgozzon ugyan azon a problémán. Lehetővé tette számunkra, hogy egy közös Github repository-ba tároljuk nem csak a weboldal kódját, hanem a mobil applikációt és a dokumentációt is. Ennek köszönhetően minden egy helyről volt elérhető számunkra és egymás munkáját is ellenőrizni tudtuk.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

12. ábra

Az alkalmazott technológiák és módszertanok együttesen biztosítják, hogy a weboldalunk gyors, megbízható és könnyen skálázható legyen, miközben a fejlesztési folyamat hatékonyságát és rugalmasságát is szem előtt tartottuk.

# 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 React Native egy komponensalapú fejlesztési keretrendszer, amellyel egy kód alapján natívan fejleszthetünk iOS és Android platformokra.

Az alkalmazásunk elsődlegesen informatív célt szolgál az alábbi funkciókat biztosítva:

* 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

Tesztelésre minden program esetén szükség van, hiszen a szoftvert emberek írják és az emberek hibázhatnak. A tesztekkel növelhetjük a termék minőségét és megbízhatóságát, viszont az elvégzésük után sem lehetünk abban biztosak, hogy már nincsenek hibák a kódunkban.

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 backend tesztelése során minden Api végpontra külön tesztet írtunk, hogy biztosra menjünk, mindegyik tökéletesen működik. Az egyes tesztek egy előre megadott adattáblát alkalmaznak és azt vizsgálják, hogy a státusz kód és a kapott üzenet megfelelő-e. Adat lekérés esetén pedig azt, hogy ténylegesen megkapjuk-e az adatot.

A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

13. 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 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.

14. ábra

A feladatok beadási dátumát és sorrendjét a Trello weboldalán tudtuk nyomon követni. A csapat a Trello segítségével szervezte a feladatokat, így mindenki átláthatta a projekt aktuális állapotát. A rendszeres megbeszélések során ellenőriztük az előrehaladást, és szükség esetén újraosztottuk a feladatokat vagy segítettünk egymásnak.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

15. ábra

A ScoreSchool mostani fejlesztési szakasza után egy interaktív tippelő platformot kívánunk létrehozni, amely lehetőséget biztosít a felhasználóknak arra, hogy tokenek segítségével fogadásokat tegyenek a különböző futballmérkőzések kimenetelére. Ez az új funkció jelentősen növelheti a weboldal vonzerejét és aktivitását, hiszen a játékosok nemcsak információkat és statisztikákat érhetnek el, hanem saját tudásukat is próbára tehetik egy biztonságos és szórakoztató környezetben.

A rendszer lényege, hogy a regisztrált felhasználók ingyenes tokeneket kapnak, amelyeket felhasználhatnak a mérkőzések kimenetelének megtippelésére. Ha a felhasználó helyesen tippel, akkor további tokeneket nyer, amelyeket később újabb fogadásokra, ranglistás versenyekre vagy egyéb prémium funkciókra használhat fel. Fontos hangsúlyozni, hogy a platform nem valódi pénzalapú szerencsejáték, hanem egy közösségi élményt nyújtó interaktív játék, amely a futball iránti rajongást hivatott erősíteni.

A jövőbeli fejlesztési terveink között szerepel egy olyan statisztikai rendszer integrálása is, amely segíteni fogja a felhasználókat a döntéshozatalban. Ez az algoritmus elemzi a múltbeli mérkőzéseket, a csapatok teljesítményét és egyéb releváns adatokat, hogy a játékosok számára valószínűségi alapú javaslatokat tegyen a fogadások során. Ezzel a funkcióval a ScoreSchool nemcsak egy egyszerű fogadási platform lesz, hanem egy olyan eszköz is, amely segíti a felhasználókat abban, hogy jobban megértsék a futballban rejlő statisztikai összefüggéseket.

A ScoreSchool célja, hogy egy dinamikusan fejlődő, folyamatosan bővülő platform legyen, amely a futballrajongók számára egyaránt nyújt információkat, szórakozást és közösségi élményt. A tokenalapú tippelő rendszer nemcsak az oldal interaktivitását fokozza, hanem a felhasználói élményt is magasabb szintre emeli. Ezzel a lépéssel a ScoreSchool egyedi helyet foglalhat el a futballal foglalkozó weboldalak között, és egy olyan platformmá válhat, amely hosszú távon is vonzó a felhasználók számára.

Összességében a jövőképünk egy olyan felület létrehozása, amely ötvözi a futball iránti szenvedélyt az interaktivitással és a közösségi élményekkel.

# Ábra jegyzék:

1. ábra: Főoldal
2. ábra: Keresés
3. ábra: Adatkezelés
4. ábra: Statisztika
5. ábra: Az adatbázis ER diagramja
6. ábra: Az adatbázis UML diagramja
7. ábra: Backend oldali szerver GET Api végpontja
8. ábra: adatok.service.ts fájl
9. ábra: app-routing.module.ts fájl
10. ábra: weblap megjelenése LG (monitor) méretben
11. ábra: weblap megjelenése MD és SM (tablet és telefon) méretben
12. ábra: a projekt github repository-jának felépítése
13. ábra: get.test.js fájl
14. ábra: Daily Scrum dokumentációja (2025.02.19)
15. 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. Anna Skoulikari *Learning Git* 2023. Elérhetőség: <https://learning.oreilly.com/library/view/learning-git/9781098133900/>
10. Josh Goldberg *Learning TypeScript* 2022. Elérhetőség: <https://learning.oreilly.com/library/view/learning-typescript/9781098110321/>