**Projektmunka dokumentálása**

**Sporthaver webalkalmazás**

Premontrei Szakgimnázium és Technikum

Keszthely

2025

Szoftverfejlesztő és tesztelő szak

5-0613-12-03

Készítette:

Kovács Kevin, Lancz Csaba

Tartalom

[Bevezetés 3](#_Toc195293020)

[A projekt megvalósításhoz használt programok 5](#_Toc195293021)

[Az adatbázis 7](#_Toc195293022)

[Kötések és kapcsolatok: 14](#_Toc195293023)

[Backend 16](#_Toc195293024)

[MVC technológia 16](#_Toc195293025)

[Models mappa 16](#_Toc195293026)

[Controller mappa 17](#_Toc195293027)

[Route mappa 18](#_Toc195293028)

[Test mappa 19](#_Toc195293029)

[Auth mappa 21](#_Toc195293030)

[Config mappa 21](#_Toc195293031)

[Node\_modules mappa 22](#_Toc195293032)

[Upload mappa 22](#_Toc195293033)

[Utils mappa 22](#_Toc195293034)

[App.js fájl 23](#_Toc195293035)

[A server.js fájl: 23](#_Toc195293036)

[Frontend 25](#_Toc195293037)

[Auth mappa 25](#_Toc195293038)

[26](#_Toc195293039)

[Főoldal 27](#_Toc195293040)

[Reflexió 32](#_Toc195293041)

# Bevezetés

A projekt munkánkat ketten valósítottuk meg: Kovács Kevin és Lancz Csaba.

A projektünk neve: Sporthaver

Azért választottuk ezt a témát mivel a csoportunk tagjai szeretik szabadidejüket különböző sportok kipróbálásával tölteni. Továbbá szeretnénk segítséget nyújtani a hasonló érdeklődési körrel rendelkező emberek számára társaságot találni a különböző sportok végzésére. Hisszük, hogy a jó közösségben eltöltött idő még értékesebb és hasznosabb, mint az egyedül végzett tevékenységek. A program lehetővé teszi, hogy az időjárás függvényében lehetséges legyen választani a kültéri és beltéri elfoglaltságok közül. Kiemelten fontosnak tartjuk, hogy az emberek több időt töltsenek a szabad levegőn és más hasonló érdeklődési körű személyek társaságában, mivel ez pozitív hatással van az emberekre fizikailag és mentálisan is.

A projekt témájának megválasztásakor fontosnak találtuk, hogy valami egyedit hozzunk létre. Nem találtunk hasonló alkalmazást, amelyik rendelkezik azon funkciókkal, amiket mi szerettünk volna megvalósítani.

Összességében a Sporthaver projekt nemcsak a sport iránti szenvedélyünket tükrözi, hanem a közösségi értékek fontosságát is hangsúlyozza. Reméljük, hogy a jövőben sok ember számára nyújt lehetőséget a sportolásra és a barátságok kialakítására, és hozzájárul a közösségi élet fellendítéséhez.

A projekt során kiemelkedően jól tudtunk együttműködni, ami nagyban hozzájárult a sikerünkhöz. Mindketten aktívan támogattuk egymás munkáját, és a feladatokat egyenlő arányban osztottuk meg. Ez a megközelítés nemcsak a munka hatékonyságát növelte, hanem erősítette a csapatszellemet is, amely elengedhetetlen volt a közös céljaink eléréséhez. A különböző szakmai háttérrel rendelkezve lehetőségünk nyílt arra, hogy kamatoztassuk egymás tudását, így sok új ismerettel gazdagodtunk.

A projekt során folyamatosan fejlődött a kommunikációnk és az empátiánk. Ráébredtünk, hogy mennyire fontos a nyílt és őszinte párbeszéd, valamint a másik fél véleményének figyelembevétele. Ez a tapasztalat nemcsak a projekt szempontjából volt értékes, hanem hosszú távon is hasznos lesz számunkra a jövőbeni együttműködések során.

A határidők precíz betartása szintén kulcsszerepet játszott a projekt sikerében. Az általunk előre meghatározott időpontok segítettek abban, hogy mindenki tudja, mikorra mit kell teljesítenie, így elkerülhettük a kapkodást és a stresszes helyzeteket. A tervezésünk és a szervezettségünk lehetővé tette, hogy minden lépést időben végrehajtsunk.

A modern technológia, különösen a felhőszolgáltatások, nagyban megkönnyítette a munkánkat. Ezáltal nem voltunk fizikailag a számítógéphez kötve, hiszen bármikor, bármilyen helyszínen hozzáférhettünk a projektünkhöz. Az állandó mentések lehetővé tették, hogy mindig a legfrissebb verzióval dolgozhassunk, és biztonságban tudhattuk az adatainkat. Ezen technikai előnyök révén még rugalmasabbá vált a munkafolyamatunk, és ez hozzájárult a hatékonyságunk növeléséhez.

Összességében a közös munka során nemcsak a szakmai tudásunkat fejlesztettük, hanem barátságos és támogató légkört is teremtettünk, ami elengedhetetlen volt a projekt sikeres megvalósításához.

# A projekt megvalósításhoz használt programok

* Discord

Az általunk leggyakrabban használt kommunikációs alkalmazás, amely lehetővé teszi a szöveges üzenetküldést, hang- és videóhívásokat, valamint a fájlok megosztását. Ezen keresztül beszéltünk egymással az iskola utáni munkafolyamatok során. Kedvenc funkciónk a képernyőmegosztás, amely lehetővé tette, hogy közösen dolgozzunk a programon és segítséget nyújtsunk egymásnak.

* Visual Studio

Az általunk használt integrált fejlesztőkörnyezet (IDE), amelyet a Microsoft fejlesztett. Node.js JavaScript keretrendszerben írtuk a programkódot. Az évek során ezt használtuk a legtöbbet az iskolai tanulmányaink során. Kiváló szinten ismerjük és alkalmazzuk a rejtett lehetőségeket, mely lehetővé tette számunkra a bővítmények hozzáadásával. A tesztelést is megkönnyítette mivel jól kiemeli a hibákat, így ezeket gyorsan orvosolni tudjuk.

* Microsoft Office

Wordot használtunk a dokumentáció elkészítésére. A PowerPointot a prezentáció megalkotásához vettük igénybe, a diákhoz áttűnéseket és animációkat alkalmazva.

* Xampp

A Dockerre használata való áttérés előtt a Xamppot használtuk webkiszolgáló környezetként. Az adatbázis megtervezés is ezen oldalon történt. Alkalmas volt továbbá arra is, hogy ellenőrizzük az adatok valóban feltöltésre kerültek-e az adatbázisba.

* Github

A közös munka során használt online verziókezelő platformunk a GitHub volt, amely lehetővé tette számunkra a kódok tárolását, kezelését és megosztását. A beszédes commitok segítségével könnyebbé tudtuk tenni egymás számára a kódok megértését. Továbbá munkák bárhonnan elérhetővé vált így nem kellett tartani, attól, hogy otthon felejtjük vagy az iskolai gépen törlésre kerül.

* Docker

Az Xampp után Dockerre váltottunk, amelynek előnye, hogy bármelyik gépről elérhetővé teszi az adatbázisokat, illetve kivédi azt a hibát, hogy az egyik gépen működik a program a másikon nem. A projektünk könnyebben kezelhető, skálázható és hordozható lett általa.

* Beekeeper Studio

Az SQL lekérdezéseinket, tábláink átláthatóságát tette egyszerűbbé. Adatokat voltunk képesek a használatával ellenőrizni és módosítani. A felhasználói felülete sokkal dinamikusabb és letisztultabb a Xampphoz képest. Kattintással tudtunk új rekordokat létrehozni és régieket szerkeszteni vele, ami rengeteg időt spórolt meg számunkra a programrészletek tesztelésekor.

* DataGrip

Adatbázis-kezelő eszköz, melynek a táblák közötti kapcsolatok grafikus megjelenítése során vettük hasznát. A táblákat számos alakzat formájában képes megjeleníti, amiből akkor profitáltunk jelentősen amikor terveztük az adatbázist és a táblák közötti kapcsolatokat.

* Figma

Webalapú dizájn és prototípus-készítő, amit az oldalak megjelenésének megtervezésére használtunk. Köszönhetően annak, hogy a projektjeink eltárolhatóak benne, képesek voltunk meg mutatni egymásnak munkáinkat és megvitatni melyik kerüljön végleges felhasználásra.

Az adatbázisA képen szöveg, képernyőkép, szoftver, Számítógépes ikon látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Az adatbázisunk 5 táblát tartalmaz.

* Felhasználók (users) tábla
  + Id: Az azonosító mező, amely elsődleges kulcsként funkcionál a rendszerben. Az értéke automatikusan generálódik, mivel az autoIncrement beállítás igazra van állítva. Ez biztosítja, hogy minden felhasználónak egyedi és egyértelmű azonosítója legyen, amely segít a rekordok nyomon követésében.
  + email: Ez a mező egyedi értéket vár, amely nem vehet fel NULL értéket. A felhasználónak kötelező megadnia egy érvényes email címet, amelyet a rendszer ellenőriz, hogy biztosan megfeleljen a szabványos email formátumnak. Ez a mező alapvető fontosságú a felhasználó azonosítása és a kommunikáció szempontjából.
  + password: Az adatvédelmi előírásoknak megfelelően a jelszó titkosítva kerül eltárolásra. Ezt a titkosítást általában a Bcrypt könyvtár segítségével végezzük, amely biztosítja, hogy a jelszavak biztonságosan tárolódjanak, és ne legyenek hozzáférhetők a rendszer adminisztrátorai számára sem.
  + username: A felhasználónév minimum 4 és maximum 16 karakter hosszúságú lehet. Fontos, hogy a felhasználónév csak betűket és számokat tartalmazzon, mivel ez biztosítja, hogy a felhasználó könnyen beazonosítható legyen a rendszerben. Ez a mező kulcsszerepet játszik a felhasználói élményben, hiszen a felhasználók gyakran ezt a nevet használják a profiljuk megjelenítésére.
  + phone: Ez a mező kizárólag a magyarországi szabványoknak megfelelő telefonszámot fogad el. Az érvényes telefonszám formátuma külön szabályozás alá esik, így a felhasználóknak figyelmesen kell megadniuk ezt az információt, hogy elkerüljék a hibás bejegyzéseket.
  + firstName: A keresztnév mező 2-15 karakter hosszú lehet, és csak betűket tartalmazhat. Ez segít abban, hogy a felhasználók ne csak azonosíthatóak legyenek, hanem hogy az adatok egységesek és rendezettek maradjanak a rendszerben.
  + lastName: A családnév mező szintén 2-15 karakter hosszú lehet, és csak betűket fogad el. A vezetéknév megadása segít a felhasználók közötti különbségek kiemelésében, valamint a személyes adatok kezelése során fontos az azonosíthatóság szempontjából.
  + birthDate: A születési dátum mező lehetővé teszi a felhasználó életkorának kiszámítását a regisztráció során. A kód ellenőrzi, hogy a felhasználó életkora 6 és 100 év között legyen, különben nem engedi a regisztrációt. Ez a funkció segít abban, hogy a felhasználói bázis életkort tekintve megfelelő legyen.
  + profileBackground: A felhasználó profiljának háttérét határozza meg, amelyet a felhasználó szabadon változtathat. Az alapértelmezett háttérszín "gradient1", de a felhasználók számára lehetőség nyílik az egyedi háttér kiválasztására is.
  + customBackground: A felhasználók egyedi háttérképet is feltölthetnek, amely lehetővé teszi számukra, hogy személyre szabják profiljuk megjelenését. E mező nagyobb méretű TEXT típusú adatokat tárolhat, így a felhasználók nagyobb képfájlokat is feltölthetnek.
  + profilePicture: A felhasználók profilképeit tárolja, amelyeket más felhasználók láthatnak az esemény résztvevői között. A profilkép hozzájárul a felhasználói élményhez és a közösség érzésének fokozásához.
  + bio: A rövid bemutatkozás mező lehetőséget ad a felhasználóknak arra, hogy egy pár mondatban bemutassák magukat. Ez a mező opcionális, de segíthet más felhasználóknak megismerni az egyént és a közösségi interakciók során fontos információkat nyújthat.
* Sportok (sportoks) tábla:
* Id: Azonosító szerepet tölt be a rendszerben, amely elsődleges kulcsként funkcionál. Ez a mező automatikusan kap értéket, mivel az autoIncrement beállítás igazra van állítva. Ennek köszönhetően minden sportág egyedi és megkülönböztethető azonosítóval rendelkezik, amely segíti a rekordok nyomon követését.
* Nev: A sportág nevét tárolja, amely elengedhetetlen az adatbázisban található sportágak azonosításához. Ez a mező nem lehet üres, és fontos, hogy egyértelmű és jól érthető neveket tartalmazzon, amelyek segítségével a felhasználók könnyen kereshetnek és azonosíthatják az egyes sportágakat.
* Leiras: Lehetőséget ad arra, hogy a felhasználók megértsék a sportág sajátosságait, valamint az ahhoz szükséges eszközöket. A leírások segítenek a sportágak népszerűsítésében és a felhasználók tájékoztatásában, ami elengedhetetlen a közösségi interakciók során.
* Kepurl: Az adott sportra jellemző kép URL-jét tárolja. Ez a mező opcionális, tehát nem kötelező a kép megadása, de a megfelelő kép segíti a felhasználókat abban, hogy vizuálisan is azonosíthassák a sportágakat, és ezáltal jobban megértsék azok jellegét.
* Helyszínek (helyszins) tábla
* Id: Az adatbázisban található helyszínek azonosítójaként szolgál és elsődleges kulcsként funkcionál. Ez a mező automatikusan kap értéket, mivel az autoIncrement beállítás aktiválva van. Ennek köszönhetően minden helyszín egyedi és megkülönböztethető azonosítóval bír, amely segíti az adatok nyomon követését.
* Nev: A helyszín nevét tárolja, amely elengedhetetlen a különböző helyszínek azonosításához. Ez a mező nem lehet üres, tehát a helyszínnek minden esetben rendelkeznie kell egy jól meghatározott névvel, amely segíti a felhasználókat abban, hogy könnyen megkereshessék a kívánt helyszínt.
* Cim: A közterület nevét és a házszámot tartalmazza, amely a helyszín pontos címét adja meg. Ez a mező szintén kötelező, így nem vehet fel NULL értéket. A cím megadása elengedhetetlen a helyszín könnyű megtalálásához, hiszen segít a felhasználóknak az adott helyszín azonosításában és elérésében.
* Telepules: A helyszín településének nevét tárolja, amely nélkülözhetetlen információ a helyszín azonosításához. Ez a mező szintén kötelező, nem lehet üres, hiszen a település neve hozzájárul a helyszín pontos elhelyezkedésének meghatározásához.
* Iranyitoszam: A helyszínhez tartozó irányítószámot tartalmazza, amely a magyarországi szabályoknak megfelelően minimum 1000 és maximum 9999 közötti szám lehet. Ez a mező nem vehet fel NULL értéket, így biztosítva, hogy minden helyszín rendelkezzen a megfelelő irányítószámmal, amely megkönnyíti a helyszínek keresését és azonosítását.
* Fedett: Logikai (boolean) értéket tárol, amely megmutatja, hogy a helyszín fedett-e vagy sem. Ez a mező hasznos információt ad a felhasználóknak, akik érdeklődnek a helyszín körülményei iránt, például időjárási viszonyok figyelembevételével.
* Oltozo: Logikai értéket tárol, amely jelzi, hogy a helyszínen van-e öltöző. Ez fontos lehet a sportesemények lebonyolításához, hiszen az öltözők jelenléte hozzájárul a sportolók kényelméhez.
* Parkolás: A parkolási lehetőségeket tartalmazza, ahol a lehetséges értékek az „ingyenes”, „fizetős” vagy „nincs”. Ez az információ elengedhetetlen a látogatók számára, mivel segít nekik tájékozódni, hogy milyen parkolási lehetőségek állnak rendelkezésre a helyszín közelében.
* Leiras: Lehetőséget ad további információk megadására a helyszínről. Itt a felhasználók részletesebb adatokat találhatnak, például a helyszín sajátosságait, szolgáltatásait vagy bármilyen egyéb releváns információt, amely segíthet a döntéshozatalban.
* Berles: Logikai értéket tárol, amely azt jelzi, hogy a helyszín bérelhető-e. Ez fontos információ a felhasználók számára, akik érdeklődnek a helyszín használatának lehetőségei iránt, például sportesemények vagy egyéb rendezvények lebonyolítása céljából.
* Események (eseménies) tábla
* Id: Az események azonosítójaként szolgál, amely elsődleges kulcsként funkcionál. Ez a mező automatikusan kap értéket, mivel az autoIncrement beállítás aktiválva van. Ennek köszönhetően minden esemény egyedi azonosítót kap, amely megkönnyíti az események nyilvántartását és kezelését.
* helyszinId: A helyszinId mező egy idegen kulcsot képvisel, amely a helyszínek azonosítójára mutat. Ez a mező elengedhetetlen a rendezvény helyszínének megjelöléséhez, és nem vehet fel NULL értéket, így biztosítva, hogy minden esemény pontosan egy adott helyszínhez legyen kapcsolva.
* userId: A userId mező szintén egy idegen kulcs, amely a felhasználó azonosítójára utal, aki az eseményt szervezi vagy létrehozza. Ez a mező kötelező, hiszen fontos, hogy tudjuk, ki a felelős az eseményért, ezzel biztosítva a megfelelő kapcsolatok nyilvántartását az adatbázisban.
* sportId: Az sportId mező egy újabb idegen kulcs, amely a sportok azonosítójára mutat. Ez a mező szintén kötelező, mivel meghatározza, hogy melyik sportág kapcsolódik az eseményhez, és így segít az események kategorizálásában és kezelésében.
* kezdo\_ido: A rendezvény kezdő időpontját tárolja, amely nem lehet korábbi a jelenlegi időpontnál. Ez a mező kötelező, és érvényességi ellenőrzéseket végez annak biztosítása érdekében, hogy a rendezvény időpontja mindig a jövőbeli időpontok közé essen.
* zaro\_ido: A rendezvény záró időpontját jelöli, amely nem lehet korábbi vagy azonos a kezdő időponttal. Ez a mező is kötelező, biztosítva, hogy a rendezvény végső időpontja logikusan kövesse a kezdő időpontot, így elkerülve a programozási hibákat és a zavarokat.
* Szint: A sportolók elvárt képességi szintjét jelöli az adott sportágban. Ez a mező kötelező, és segít a rendezvény szintjének megfelelő meghatározásában, amely fontos lehet a résztvevők számára.
* minimumEletkor: Azt a legkisebb életkort tárolja, amelynek meg kell felelnie a résztvevőknek az eseményhez való csatlakozáshoz. Ez a mező kötelező, és biztosítja, hogy a rendezvényre jelentkezők megfeleljenek a korhatároknak.
* maximumEletkor: A legnagyobb életkort határozza meg, amelynek szintén meg kell felelnie a résztvevőknek. Ez a mező kötelező, és fontos információt nyújt a rendezvény szervezője számára a résztvevők korosztályának meghatározásához.
* maximumLetszam: Azt a maximális létszámot tárolja, amelyet a szervező meg kíván határozni az eseményre. Ez a mező nem lehet kevesebb mint 2, így biztosítva, hogy elegendő résztvevő legyen a rendezvény lebonyolításához.
* imageUrl: Lehetőséget biztosít egy kép feltöltésére az eseményhez, amely vizuálisan gazdagítja a rendezvényt és segíti a promóciót. Ez a mező nem kötelező, ha a szervező nem kíván képet feltölteni, azt szabadon megteheti.
* Résztvevők (resztvevős) tábla:
* Id: Az események résztvevőinek azonosítójaként szolgál. Ez a mező elsődleges kulcsként funkcionál az adatbázisban, és automatikusan kap értéket, mivel az autoIncrement beállítás aktív. Ennek a tulajdonságnak köszönhetően minden résztvevő egyedi azonosítót kap, amely megkönnyíti a résztvevők kezelését és nyilvántartását az események keretein belül.
* esemenyId: Az eseményId mező idegen kulcsot képvisel, amely az esemény azonosítójára mutat. Ez a mező kötelező, mivel elengedhetetlen ahhoz, hogy tudjuk, melyik eseményhez csatlakozik a résztvevő. A hivatkozás biztosítja, hogy minden résztvevő szoros kapcsolatban álljon a megfelelő eseménnyel.
* userId: A userId mező szintén egy idegen kulcs, amely a felhasználó azonosítóját tárolja. Ez a mező is kötelező, és lehetővé teszi annak nyilvántartását, hogy ki a résztvevője az eseménynek. A kapcsolatok pontos rögzítése érdekében ez a mező elengedhetetlen a résztvevők azonosításához.
* szerep: ENUM típust képvisel, amely meghatározza a résztvevő szerepét az eseményben. A lehetséges értékek közé tartozik a 'szervező' és a 'játékos', amelyek jelzik, hogy a felhasználó szervezőként vesz részt a rendezvény lebonyolításában, vagy pedig csak résztvevőként van jelen. Ez a mező kötelező, és alapértelmezett értéke a 'játékos', amely azt jelzi, hogy ha a szerep nem kerül kifejezetten meghatározásra, akkor a résztvevő játékosnak számít.
* csatlakozásDátuma: Azt az időpontot tárolja, amikor a résztvevő csatlakozott az eseményhez. Ez a mező kötelező, és alapértelmezett értéke a jelenlegi időpont, amely automatikusan beállításra kerül a rekord létrehozásakor. Ez a mező segít nyomon követni, hogy a résztvevők mikor csatlakoztak az eseményhez.
* státusz: ENUM típust képvisel, amely a résztvevő állapotát jelzi az eseménnyel kapcsolatban. A lehetséges értékek közé tartozik az 'elfogadva', 'elutasítva', 'függőben' és 'meghívott'. Ez a mező kötelező, és alapértelmezett értéke a 'függőben', ami azt jelenti, hogy a résztvevő státuszát még nem véglegesítették, és további lépésekre van szükség ahhoz, hogy a részvétel megerősítést nyerjen.
* megjegyzés: Lehetőséget biztosít a résztvevő számára, hogy kiegészítő információkat adjon meg az eseménnyel kapcsolatban. Ez a mező nem kötelező, ha a résztvevő nem kíván megjegyzést fűzni, akkor azt szabadon kihagyhatja.

A képen szöveg, képernyőkép, Betűtípus látható

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

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

## Kötések és kapcsolatok:

Az események tábla a projektünkben a legtöbb kapcsolattal rendelkező tábla, mely kapcsolatban áll az összes többivel.

Az esemény tábla elsődleges kulcsa az Id amely a résztvevők tábla eseményId indexére csatlakozik annak érdekében, hogy követhető legyen melyik eseménynek kik a résztvevői. Többhöz a többhöz kapcsolat, mert több eseményben lehet több résztvevő is.

Az esemény tábla helyszinId indexe a helyszín tábla Id nevű elsődleges kulcsához van kötve, hogy megkapjuk az esemény helyszínét. Egy az egyhez kapcsolat mert egy esemény csak egy helyszínen játszódhat.

Az esemény tábla userId indexe a user tábla Id nevű elsődleges kulcsához van kötve, hogy megkapjuk az esemény létrehozójának azonosítóját. Többhöz a többhöz kapcsolat, mert több user lehet több eseményen is részt vehet.

Az esemény tábla sportId indexe a sport tábla Id nevű elsődleges kulcsához van kötve, hogy megkapjuk az eseményen milyen sportot fognak játszani. Egy az egyhez kapcsolat mert egy eseményen csak egy sport játszódhat.

A user tábla Id elsődleges kulcsa a helyszin tábla userId nevű indexéhez van kötve, hogy megkapjuk a helyszín létrehozójának Id-ját. Egy a többhöz kapcsolat, mert egy user több helyszínt is létrehozhat.

A user tábla Id elsődleges kulcsa a résztvevők tábla userId nevű indexéhez van kötve, hogy megkapjuk a résztvevő Id-ját. Többhöz a többhöz kapcsolat, mert több user lehet több eseményen is részt vehet.

# Backend

## MVC technológia

A backendünket MVC (Model-View-Controller) technológiával valósítottuk meg.

 **Model (Modell)**:

Ez a réteg tartalmazza az adatokat és a logikát. A modell kezeli az adatokat, az adatbázis lekérdezéseket, a formátumokat és a szabályokat. Ha valami történik az alkalmazásban, ami adatváltozást igényel (pl. új felhasználó regisztrálása), a modell az, amelyik kezeli és tárolja ezeket a változásokat.

 **View (Nézet)**:

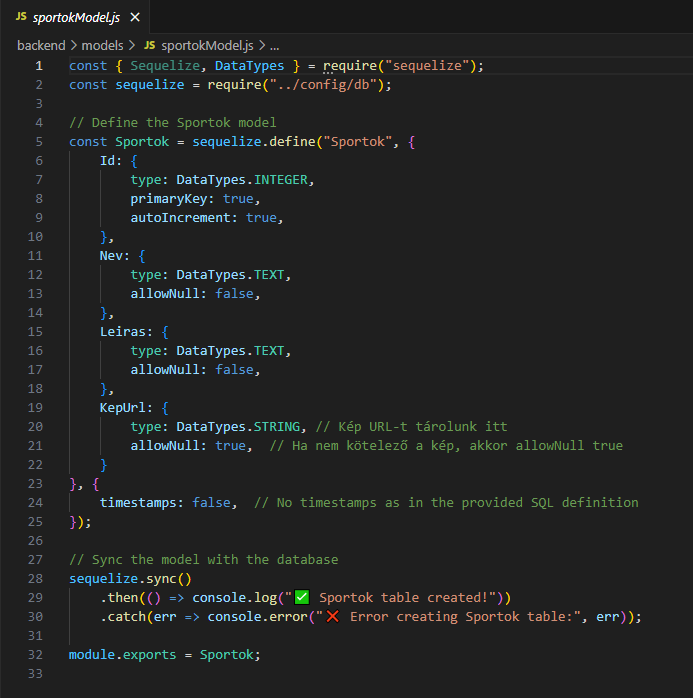
A nézet a felhasználói interfészt (UI) és a megjelenítést kezeli. Ez a réteg az, ami a felhasználóval kommunikál. A nézet megjeleníti az adatokat, amelyeket a modell biztosít, és bemutatja azokat a felhasználónak. A nézetnek nem kell tudnia, hogyan kezeljük az adatokat, csak azt kell tudnia, hogy hogyan kell azokat megjeleníteni.

 **Controller (Vezérlő)**:

A vezérlő közvetítőként működik a modell és a nézet között. A vezérlő fogadja a felhasználói interakciókat (pl. gombnyomásokat, űrlap beküldéseket), elvégzi az ezekhez szükséges műveleteket (például adatbázis lekérdezést), majd frissíti a nézetet az új adatokat vagy állapotokat figyelembe véve. A vezérlő biztosítja, hogy a megfelelő adatokat a megfelelő nézethez küldje.

## Models mappa

A models mappában vannak az adatbázisunk tábláinak model fájljai, melyek szerepe, hogy definiálják a Sequalizet és az ORM-et. Az ORM egy **programozási technika**, amit azért használtuk, mert **az adatbázis táblákat és azok mezőit objektumokként** jeleníti meg a kódban ezért **nem kellett SQL nyelven** lekérdezéseket írnunk, ami rengeteg időt spórolt meg számunkra és könnyebbé tette a dolgunkat. **Objektumorientált módon** tudtuk kezelni az adatokat.

* Kapcsolódik más modelekhez, vagyis a többi táblához.
* Elvégzi a szükséges validációkat az adott mezőkhöz például születési dátum alapján minimum 6 évesnek kell legyen, hogy regisztrálhasson.
* Létrehozza vagy szinkronizálja a táblát az adatbázissal.
* Exportálja, vagyis bárhonnan meghívhatóvá teszi magát.

## Controller mappa

A controllers mappában lévő fájljaink tartalmazzák az adott modelhez kapcsolódó logikákat. Majd az utolsó sorokban ezeket a függvényeket (funkciókat) exportálja a fájlból, vagyis **más fájlokban is használhatóvá teszi** őket.

A képen szöveg, képernyőkép, szoftver, Operációs rendszer látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

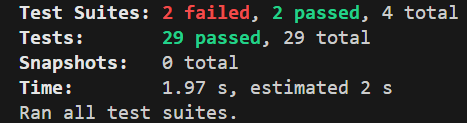
## Route mappa

* A route mappában lévő fájljaink feladata, hogy:
* Importálják a szükséges controller fájlt, az expresst, a routert továbbá jwt-t.
* Kezelik az adott táblához tartozó útvonalakat.
* Exportálják a modulet biztosítva, hogy ezt a route-ot fel lehessen használni máshol.

A képen szöveg, képernyőkép, szoftver, képernyő látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

## Test mappa

A test mappában tároltuk el a controller fájlok jest tesztetjeit. A **Jest** egy modern és nagyon népszerű **JavaScript tesztelő keretrendszer**, amit leggyakrabban **Node.js**-es vagy **frontend (React)** projektekben használnak szoftvertesztelők. Képes tesztelni a funkciók viselkedését, az API végpontok elérhetőségét, a hiba kezelés megfelelően működését, az adatbázis logikai működését, továbbá a user interface komponensek helyes működését. **getSportokById**: Teszteli a sport azonosító szerinti lekérdezését, beleértve a sikeres lekérdezést, a 404-es hibát és a szerverhibákat. 

## Auth mappa

Auth mappában tároltuk a Middleware.js fájlt, ami kezeli, hogy a felhasználó rendelkezik-e érvényes tokennel.A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A Middleware.js generateToken függvénye:

* Létrehoz egy **JWT tokent**, amely a userId-t és az email-t tartalmazza.
* A token titkosítva lesz a "secretkey" kulccsal.
* A token érvényességi ideje: **1 óra**.

## Config mappa

A db.js fájl hozza létre a kapcsolatot az adatbázissal.

* A 4. sor meghatározza a MySQL típusú az adatbázis.
* Az 5. sor az adatbázis felhasználóneve.
* A 6. sor az adatbázishoz tartozó jelszó.
* A 7. sor az adatbázis neve

Ez a konfiguráció teszi lehetővé, hogy a Node.js alkalmazás kapcsolatot létesítsen a sportpartner\_kereso nevű MySQL-adatbázissal.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

## Node\_modules mappa

A **node\_modules** mappa a **Node.js** alkalmazásokban a telepített csomagokat és könyvtárakat tárolja, amelyek az alkalmazás működéséhez szükségesek. Az **npm (Node Package Manager)** segítségével telepített csomagok mind a **node\_modules** mappában találhatók. Továbbá tárolja azokat a csomagokat, amelyekre az alkalmazásnak szüksége van a működéshez. Ezeket a csomagokat a projektben a package.json fájlban definiálhatjuk. A **package-lock.json** fájl is biztosítja, hogy minden fejlesztő és környezet ugyanazokat a csomag verziókat használja.

## Upload mappa

Az upload mappában kerülnek eltárolásra a feltöltött képek.

## Utils mappa

Az utils mappában van az eventScheduler.js fájl, aminek feladata, hogy **automatikusan törli azokat az eseményeket, amelyek 31 nappal a lezárásuk után jártak le**. Az események archívumba rakásával kapcsolatban volt szükségünk az alkalmazására.

## App.js fájl

Az app.js file teszi lehetővé, hogy **nagy méretű kérések (pl. űrlapok, JSON adatok, képek)** feldolgozása is megvalósítható legyen. A projektünkben a képek feltöltése miatt volt szükség a használatára.

## A server.js **fájl:**

* **Importálja az expresst modult: amely a http szerver és az útvonalak kezelését biztosítja.**
* **Importálja a corst modult: amely lehetővé teszi, hogy az alkalmazás más domainekről érkező kéréseket is kezelni tudja.**
* importálja a Routes fájlokat.
* Elvégzi a Middleware beállításokat.
* Beállítja az útvonalakat és a route-okat.
* Szinkronizálja az adatbázist Sequelizevel.
* Elindítja a szervert.

A képen szöveg, képernyőkép, képernyő, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

# Frontend

A vite.svg fájl a Vite logóját tartalmazza.

Az src mappában helyeztük el az összes .jsx kiterjesztésű frontend fájlunkat.

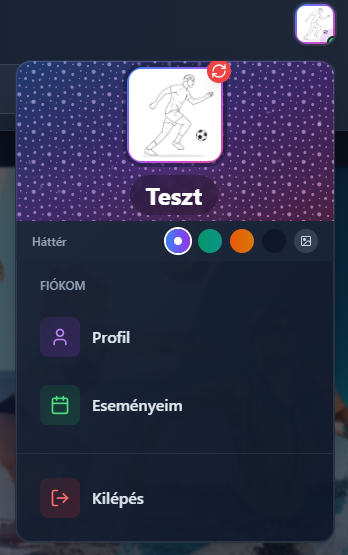
## Auth mappa

A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Az auth mappában szereplő login.jsx és register.jsx fájlok a felhasználók regisztrációjának és belépésének grafikus megjelenítéséért felelnek. A register.jsx továbbá tartalmazza a felhasználási feltételek és Adatvédelmi irányelvek footer hivatkozását is. A beírt jelszó a szem ikonra katintva láthatóvá válik. Az összes mező kitöltése kötelező.

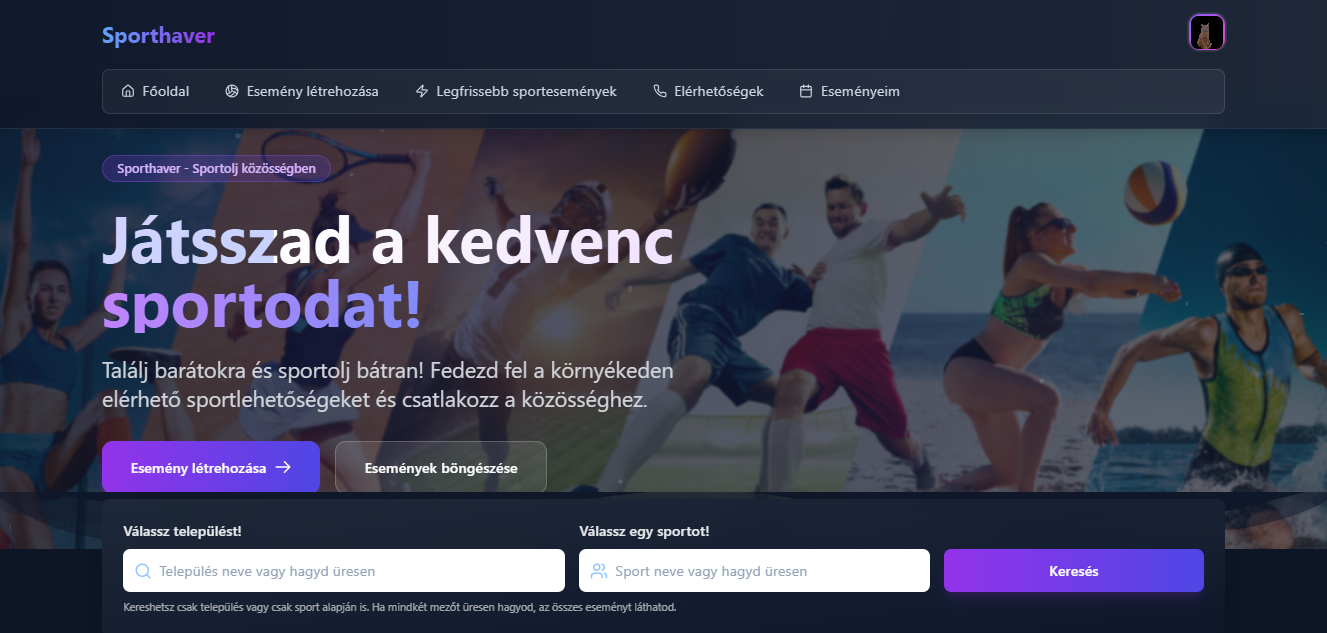
## A képen szöveg, képernyőkép, Betűtípus, szoftver látható Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

## Főoldal

A Header.jsx fájlunkban található az oldalunk neve, amely egyszerre szolgál navigációs gombként és design elemként is. Itt van a regisztráció és belépés gomb is. Továbbá az 5 pontból álló menüsor is, ami hamburger menüvé képes átalakulni a képernyő szűkítésekor, biztosítva ezzel a reszponzívitást. Ezenkívül helyet kapott egy lenyíló profil modal a jobboldalon, melyre feltölthető profilkép és háttérkép is felhasználó galériából. A 3 pontot tartalmazó menüben a található meg a Profil pont, amely a Profile.jsx filera visz át. Az eseményeim pont azonos funkciót lát el, mint a header megtalálható párja, melyről később részletesen írtunk. Legalul el egy Kilépés pont intézi a felhasználó kiléptetését a fiókjából.

A HeroSection.jsx fileban helyeztük el a képet, amely menüsorunk és a főoldali kereső sávunk közötti területet tölti ki. Emellett két gomb is elhelyezésre került benne.

A SearchForm.jsx-ben került elhelyezésre a település- és sportkereső. A keresés logikája megengedi, hogy a település ésvagy sport mező üresen maradhasson. A keresés gombra kattintva az oldal átnavigál a találatokat megjelenítő sport-mate-finder.jsx oldalra.



Azt event-modal.jsx felel az események létrehozásáért. Megtalálható benne a helyszín mellett egy + gomb, amely a helyszin-modal.jsx-et hozza be, ahol a felhasználók sport helyszíneket hozhatnak létre. Az esemény létrehozásánal kötelezően kitöltendő mező a helyszín, sport, minimum-, maximuméletkor, létszám, Szint, minimum-, maximuméletkor és a képfeltöltése. Az adatok mentésére az ,,Esemény létrehozása” feliratú gomb szolgál.

A képen szöveg, képernyőkép, szoftver, multimédia látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A PopularDestination.jsx jeleníti a maximum 8 darab legaktuálisabban létrehozott sporteseményt card-ként egy Részletek gombbal, ami szintén a sport-mate-finder.jsx oldalra navigálja a bejelentkezett felhasználókat.

A Footer mappába 5 fájl került. A Rólunk, Általános Felhasználási Feltételek, Adatvédelmi tájékoztató, Impresszum felugró modal-ként funkciónálnak a szükséges feltételeknek megfelelően, de fiktív adatokkal kitöltve. Elhelyezés került a benne az oldal készítőinek elérhetőségek és közösségi médiás oldalai – fiktív adatokkal – megfelelve modern világ elvárásainak. A Sporthaver szövegre kattintva felscrollozik az oldal tetejére.

A myEvent.jsx fájlban kerültek eltárolásra az Eseményeim pont tartalma. A 6 menüpont:

Összes: Az összes esemény, melynek a felhasználó részese

Szervezőként: Az összes esemény, melynek a felhasználó a szervezője

Résztvevőként: Az összes esemény, melynek a felhasználó játékosa

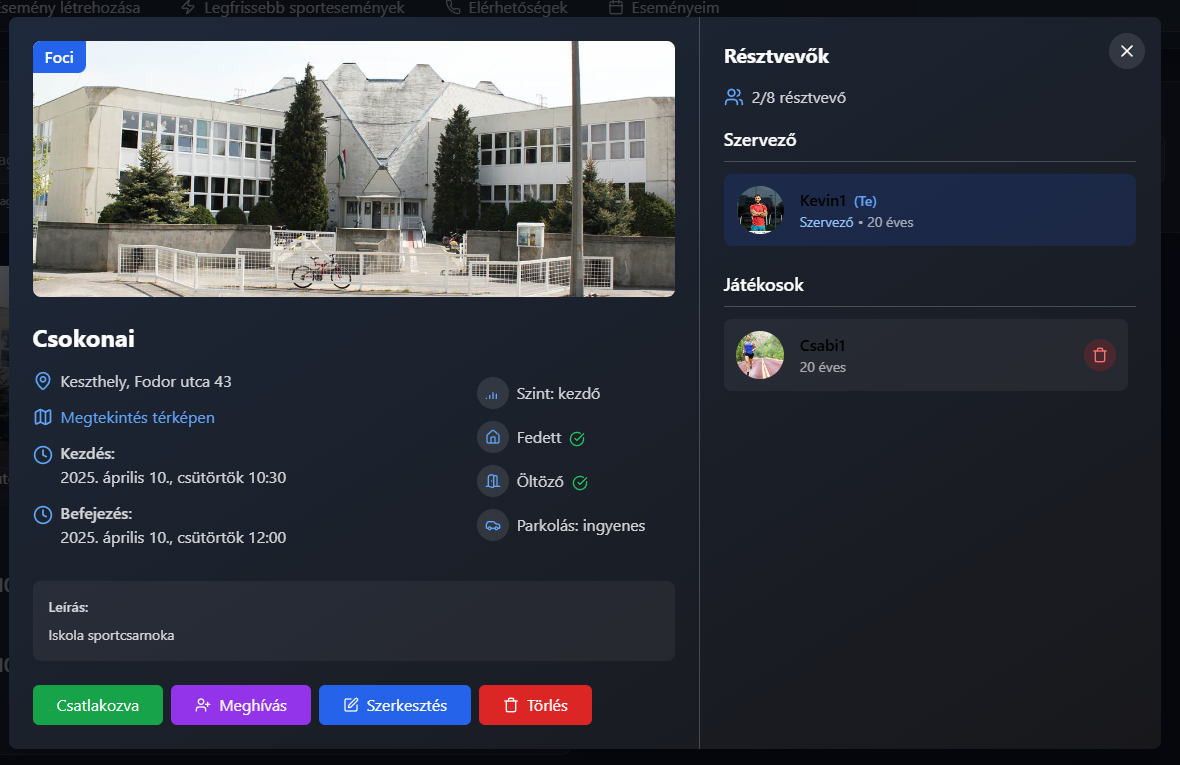
Függőben: Az összes esemény, ahova a felhasználó jelenetkezett, de még nem vették fel és el sem utasították

Meghívásaim: Az összes esemény, ahova a felhasználó a szervező vagy játékos meghívta, de még nem fogadta el és nem utasította el a felkérést.

A képen képernyőkép, szöveg, Multimédiás szoftver, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Archívum: Az összes esemény, amely már megtörtént de még nem telt el az időpont vége óta 30 nap. A csatlakozási kérelmek automatikusan törlődnek ahogy archívumba kerül az esemény.

Mindegyik menüpont alatt az esemény ,,Részletek” feliratú gombra kattinva megjelenik a sport-event-details-modal.jsx a menüpont lényegének megfelelő funkciójú gombokkal. Jogosultságok alapján két kategóriára bontottuk a felhasználókat:



Szervező: Játékosokat hívhat be az eseményre, játékosok jelentkezéseit fogadhatja és utasíthatja el, szerkesztheti az esemény és a helyszín adatait, törölheti az eseményt.

Játékos: Jelentkezhet az eseményre vagy fogadhatja el a meghívást, kiléphet az eseményből.

sport-event-details-modal.jsx jobboldali részén megjelenik a szervező és a játékosok felhasználó neve és életkor, amelyre kattintva egy új modal megjeleníti a legfontosabb adatait:

Bemutatkozás: Profile.jsx-ben megadott szöveg

Elérhetőség: gmail cím és telefonszám, továbbá, hogy hány eseményben volt szervező és játékos.

A sport-mate-finder.jsx megmutatja az összes olyan eseményt, amely megfelel az felhasználó által a kereső mezőbe beírt feltételeknek.

# Reflexió

A Sporthaver projekt számunkra sokkal több volt, mint egy iskolai feladat: egy közös kaland, amelyben nemcsak szakmailag fejlődtünk, hanem emberileg is. Az együtt eltöltött órák, a sok beszélgetés, nevetés, néha vita és újratervezés közelebb hozott minket egymáshoz. Megtanultunk igazán együttműködni – nem csak megosztani a feladatokat, hanem figyelni a másikra, segíteni, ha elakadt, bátorítani, ha bizonytalan volt.

A projekt során fontos tapasztalat volt számunkra, hogy milyen sokat számít a bizalom a csapaton belül. Bíztunk abban, hogy a másik megcsinálja a részét, hogy lehet rá számítani, és ez hatalmas motivációt adott. Jó érzés volt látni, ahogy az ötleteinkből fokozatosan valóság lett – és nemcsak egy működő alkalmazás, hanem egy olyan platform, amit egyszer mások is használhatnak majd.

Emellett rengeteget tanultunk önmagunkról is: hogyan reagálunk stresszhelyzetben, mennyire vagyunk képesek alkalmazkodni, és milyen jó érzés, amikor a közös munka végén büszkék lehetünk arra, amit együtt létrehoztunk. Ez a projekt nemcsak egy technológiai megoldás, hanem egy közös élmény, ami örökre megmarad bennünk.