Békéscsabai SZC Nemes Tihamér Technikum és Kollégium

Szakképesítés megnevezése: Szoftverfejlesztő és –tesztelő

Azonosító száma: 5 0613 12 03

**VIZSGAREMEK**

CsocsoWeb

Készítették:

Medve Gergő Osztály: 13B Oktatási azonosító:72560126204

Varga Dávid Osztály: 13B Oktatási azonosító:72544562731

Viczián Dániel Osztály: 13B Oktatási azonosító:72559004343

Békéscsaba, 2024/2025

Tartalomjegyzék

[Bevezető 4](#_Toc198063264)

[A probléma ismertetése 4](#_Toc198063265)

[Választott téma indoklása 4](#_Toc198063266)

[Célunk 4](#_Toc198063267)

[Kinek szánjuk oldalunkat 5](#_Toc198063268)

[Téma kifejtése 5](#_Toc198063269)

[Fejlesztői dokumentáció 6](#_Toc198063270)

[Adatbázis 6](#_Toc198063271)

[Competition tábla: 6](#_Toc198063272)

[Registration tábla 6](#_Toc198063273)

[User tábla 7](#_Toc198063274)

[Categories tábla 7](#_Toc198063275)

[news tábla 7](#_Toc198063276)

[Backend 8](#_Toc198063277)

[Adatbázis tábláinak létrehozása 8](#_Toc198063278)

[Modellek 9](#_Toc198063279)

[Controllerek 9](#_Toc198063280)

[Excel tábla generálása 18](#_Toc198063281)

[API útvonalak 24](#_Toc198063282)

[Frontend 25](#_Toc198063283)

[Felépítés 25](#_Toc198063284)

[Reszponzivítás 29](#_Toc198063285)

[Minták 29](#_Toc198063286)

[Tesztek 32](#_Toc198063287)

[Miért controller tesztekre fókuszáltunk? 33](#_Toc198063288)

[Használt technológiák a teszteléshez 33](#_Toc198063289)

[Feature tesztek a gyakorlatban 34](#_Toc198063290)

[CategorieControllerTest.php 34](#_Toc198063291)

[ChangePasswordTest.php 34](#_Toc198063292)

[ContestControllerTest.php 34](#_Toc198063293)

[LoginTest.php 34](#_Toc198063294)

[NewsControllerTest.php 34](#_Toc198063295)

[RegisterTest.php 35](#_Toc198063296)

[UpdateProfileTest.php 35](#_Toc198063297)

[Felhasználói dokumentáció 36](#_Toc198063298)

[Rendszerkövetelmények 37](#_Toc198063299)

[Weboldal futtatása 37](#_Toc198063300)

[Weboldal használatának részletes ismertetése 39](#_Toc198063301)

[Regisztrációs folyamatok ismertetése 40](#_Toc198063302)

[Belépés folyamata 42](#_Toc198063303)

[Nevezés leadása egy versenyre 45](#_Toc198063304)

[Admin extra lehetőségei: 46](#_Toc198063305)

[Végszó 50](#_Toc198063306)

[Összegzés 50](#_Toc198063307)

[A munkánk értékelése 50](#_Toc198063308)

[Köszönetnyilvánítás 51](#_Toc198063309)

[Irodalomjegyzék 51](#_Toc198063310)

[Ábrajegyzék 52](#_Toc198063311)

# Bevezető

## A probléma ismertetése

A projektünk a Viharsarki csocsó egyesület számára egy verseny létrehozó és nevezésre szolgáló Webhely, amellyel rendkívül megkönnyíthetjük az egyesületet vezető és versenyeket szervező Földi Zsolt dolgát. Ezután Földi Zsolttal felvéve a kapcsolatot 3 fő problémával szembesültünk, amelyek megoldásra vártak. Az első probléma az volt, hogy egyes versenyek létrehozásánál éjjeleken át kellet dolgozni, mivel a nevezés egy facebook poston keresztül történt, amely mondanunk sem kell nem a legideálisabb. Ezeket kellett átvinnie egy Excel táblába, amellyel a későbbiekben további munkálatokat tudott elvégezni. Ide sorolandó a nevezett játékosok ranglistán való helyezése és kigyűjtése, valamint a besorolásuk. Ezek az adatok elengedhetetlenek a játékosok nevezési díja meghatározásához illetve a sorsolás elkészítéséhez. Az utolsó, ámbár nem elhanyagolható része a problémájának az volt, hogy a nevezett játékosok után ezeket a játékosokat excel táblába kellet felvezetni a pontszámaik és az egyes kategóriákban résztvevő játékosok létszáma meghatározása miatt.

# Választott téma indoklása

## Célunk

Célunk egy könnyen kezelhető weboldal elkészítése, amellyel a játékosok és a versenyt szervezők is könnyedén megbarátkozhatnak és megérthetik működését. A felhasználók regisztrálhatnak az oldalunkra, amelyen keresztül így lehetőségük nyílik, nevezi a versenyre, valamint nyomon is követhetik leadott nevezéseiket és a versenyek megrendezésnek fontos információit (ilyen például mennyibe került a nevezésük a versenyre, illetve hol és mikor rendezik azt). Ezen felül a szervezők dolgát is igencsak megkönnyíti, oldalunk mivel versenyeket hozhatnak létre és átfogó excel táblázat várja ezek után őket. Az excel táblázatban külön kigyűjtve szerepelnek a versenyzők és az álltaluk leadott kategóriákhoz tartozó díjjak valamint az is, hogy az adott versenyzőnek milyen végösszeget kell fizetni a nevezések befizetésekor.

## Kinek szánjuk oldalunkat

Oldalunkat minden Viharsarki csocsó versenyen induló versenyzőnek bátran ajánljuk, illetve a Viharsarki csocsó vezetősége is szeretné minél hamarabb elérni, hogy a versenyekre való nevezés teljes mértékben az oldalon keresztül valósuljon meg.

# Téma kifejtése

A projektünk 3 fő részből áll: Adatbázis, Backend és Frontend részekre osztható fel.

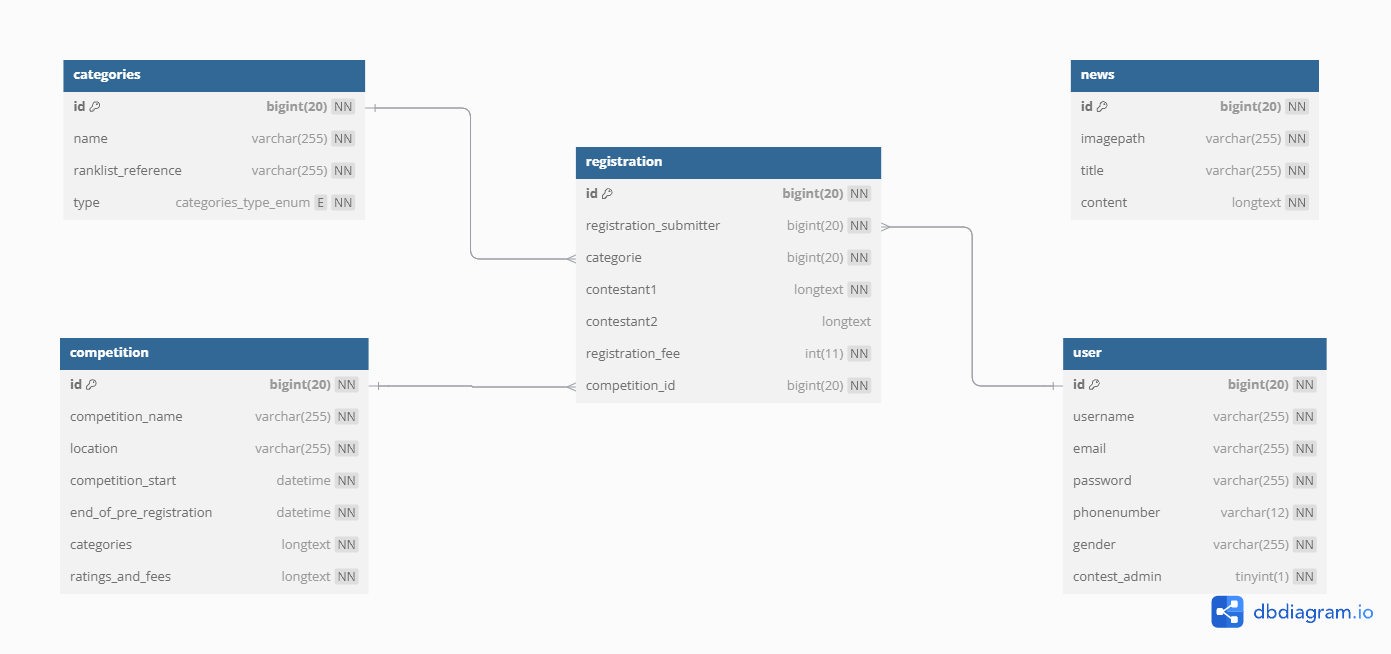
Az adatbázisunkban kerül eltárolása minden fontos információ, amelyre hosszú távon szükségünk lehet Pl.: versenyek, kategóriák. Mi MySQL adatbázis típust használtuk projektünk elkészítése során.

A backend feladat a frontend és az adatbázis között kommunikáció zökkenőmentes lebonyolítása, amelyhez mi PHP-t használtunk, és ahogyan korábban is említettük az adatbázist MySQL-ben készítettük el.

A frontend felelős a projektünk megjelenésért és minden oldal kinézete itt épül fel, legyen szó akár az oldal szöveg részéről akár a stílus formázásról ezzel kapcsolatban minden itt ment végbe. Ehhez mi React vite-ot használtunk, illetve a stílus elemekhez pedig tailwind css-t.

# Fejlesztői dokumentáció

## Adatbázis



**1. ábra: Adatbázis diagram**

### Competition tábla:

* Id: Elsődleges kulccsal rendelkezik, 20 számjegyű bigint. Ezzel azonosítjuk a versenyeket
* Competition\_name: A verseny neve, 255 karakter hosszú string formátumú beviteli mező, kötelező a kitöltése
* Location: A verseny helyszíne, 255 karakter hosszú string, kötelező mező.
* competition\_start: A verseny kezdési időpontja, datetime típusú, kötelező mező.
* end\_of\_pre\_registration: Az előregisztráció vége, datetime típusú, kötelező mező.
* categories: A versenyhez tartozó kategóriák szöveges formában, longtext típusú, kötelező mező.
* ratings\_and\_fees: Értékelési szempontok és díjak, longtext típusú, kötelező mező.

### Registration tábla

* id: Elsődleges kulcs, 20 számjegyű bigint. Egyedileg azonosítja a regisztrációkat.
* registration\_submitter: Idegen kulcs, az user tábla id mezőjére hivatkozik. A regisztrációt beküldő felhasználót azonosítója.
* categorie: Idegen kulcs, a categories tábla id mezőjére hivatkozik. A kiválasztott kategóriának megfelelő id-val tér vissza.
* contestant1: Az első versenyző adatai, longtext típusú, kötelező mező. Ez tartalmazza a versenyző nevét, pontszámát az adott kategóriában, illetve a besorolását is
* contestant2: A második versenyző adatai tartalmazza hasonló képpen mint az első versenyzőnél. Ez párosban történő indulás esetén fontos számunkra, longtext típusú, nem kötelező.
* registration\_fee: A regisztrációs díj, 11 számjegyű egész szám (int), kötelező mező.
* competition\_id: Idegen kulcs, a competition tábla id mezőjére hivatkozik.

### User tábla

* id: Elsődleges kulcs, 20 számjegyű bigint. Egyedileg azonosítja a felhasználót.
* username: A felhasználó neve, 255 karakter hosszú string, kötelező mező.
* email: A felhasználó email címe, 255 karakter hosszú string, kötelező mező.
* password: A jelszó (hash formában), 255 karakter hosszú string, kötelező mező.
* phonenumber: Telefonszám, 12 karakter hosszú string, kötelező mező.
* gender: A felhasználó neme, 255 karakter hosszú string, kötelező mező.
* contest\_admin: Logikai érték (tinyint), 1, ha a felhasználó adminisztrátor, 0, ha nem, kötelező mező. Ez segít abban, hogy milyen jogosultsága lehet egyes felhasználóknak.

### Categories tábla

* id: Elsődleges kulcs, 20 számjegyű bigint. Egyedileg azonosítja a kategóriákat.
* name: A kategória neve, 255 karakter hosszú string, kötelező mező.
* ranklist\_reference: A ranglista excel fileban szereplő munkalap neve, 255 karakter hosszú string, kötelező mező.
* type: A kategória típusa csak egyéni vagy páros lehet, ezért egy enum típusú mezőt hoztunk létre

### news tábla

* Ez a tábla nem kapcsolódik semelyik másik táblához, hiszen ezen keresztül csak a hírek létrehozása és tárolása történik az esetleges versenyek utáni hírek és események jelennek meg és vannak eltárolva itt.
* id: Elsődleges kulcs, 20 számjegyű bigint. Egyedileg azonosítja a híreknek.
* imagepath: A hír képének az útvonala, 255 karakter hosszú string, kötelező mező.
* title: A hír címe, 255 karakter hosszú string, kötelező mező.
* content: A hír részletesebb leírása, egy longtext típusú, kötelező mező.

## Backend

### Adatbázis tábláinak létrehozása

Ebben a részben a backend bemutatására kerül sor, kezdjük a táblák létrehozásával. A táblákat a laravel migrációs fájlba építettük fel

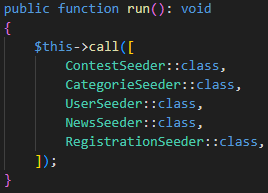
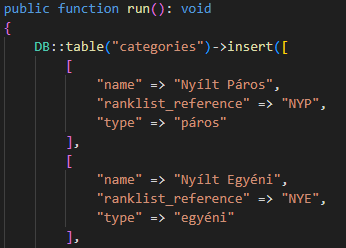
A screen shot of a computer program

AI-generated content may be incorrect.

**2. ábra: A táblák generálása**

A táblákat kezdetleges adatokkal is fel lehet tölteni néhány seeder segítségével

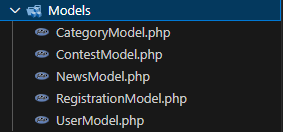
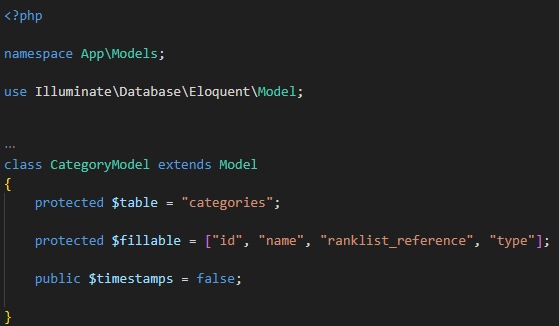
**3. ábra: Tábla seederek**



### Modellek

Az öt táblához öt modellt hoztunk létre az egyszerűbb adatkezelés érdekében. Ezek segítségével érhetőek el a táblák adatai és egyszerűsítik az adatfelvitelt és módosítást.

4**. ábra: Modellek**



### Controllerek

A controllerek a webalkalmazások szerves részét teszik ki. Feladatuk, hogy összekapcsolják a felhasználói kéréseket az alkalmazás logikájával, vagyis közvetítőként szolgálnak a beérkező HTTP-kérések, az adatbázis és a frontend között. A mi esetünkben sok controllerre volt szükség, mivel rengeteg adatot kezelünk.

NewsController A NewsController a hírek kezeléséért felelős a rendszerben. Ez a vezérlő 3 fő funkciót lát el a hírekkel kapcsolatban a mentésüket, lekérdezésüket és nem utolsó sorban a törlésüket.

A getnews metódus az összes hír lekérdezéséért felelős

A screen shot of a computer code

AI-generated content may be incorrect.

**5. ábra: A hírek beszerzése**

A storenews metódus az új hírek mentését végzi. Validálás alatt meggyőződünk arról, hogy ne lehessen üres hírt feltölteni és azt is ellenőrzi, hogy történt-e képfeltöltés.

A computer screen shot of code

AI-generated content may be incorrect.

**6. ábra: Hírek eltárolása**

A deletenews metódus egy adott hír törlését végzi. amennyiben nem található a keresett hír, akkor hibával tér vissza.

A screenshot of a computer program

AI-generated content may be incorrect.

**7. ábra: A hírek törlése**

AuthController segíti a felhasználó kezelésben és az autentikációs feladatok elvégzésére. Ez a controller felelős a regisztráció, bejelentkezés, profilmódosítás és jelszócsere végrehajtásáért.

A register metódus egy új felhasználó regisztrációját végzi el.

A logIn metódus a felhasználó hitelesítése és token generálásáért felelős.

Az updateProfile metódus a bejelentkezett felhasználó adatainak frissítését intézi

A changePassword metódus a jelszócserét hajt végre biztonsági ellenőrzéssel

Új felhasználó regisztrálásánál a jelszó hashelt állapotban kerül eltárolásra. Validációs hibakezelést is biztosít.

Egy új regisztráció során az adatok mentésre kerülnek az adatbázisban, majd ezután amikor a felhasználó belép az oldalra a frissen regisztrált fiókjával, ekkor generáltatunk Laravel sanctum segítségével egy tokent, amelyet később validációs célokra használunk fel.

A képen Betűtípus, képernyőkép, szöveg látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**8. ábra: Token generálása**

Ez a token API-kommunikáció során biztosítja a felhasználó hitelesítését, így API kérések során a felhasználó ezáltal hitelesíthető. A token minden egyes kérés során az Authorization A képen szöveg, Betűtípus, képernyőkép, sor látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**9. ábra: Api hitelesítése**

headerben kerül továbbításra, amelyet a Laravel ellenőriz. Ha a token érvényes, a rendszer biztosítja a hozzáférést a védett API-hoz.

Ezáltal a backend könnyen beazonosíthatja a felhasználót. Tekintve, hogy egyes API-k védve vannak mert érzékeny adatokat tartalmaznak, így megtudjuk akadályozni, hogy nem beazonosított felhasználók is hozzáférhessenek az érzékeny adatokhoz az API-n belül. A képen szöveg, képernyőkép, sor, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**10. ábra: Védett útvonalak**

Mivel ez a token localStorage-ban van eltárolva, így a felhasználónak nem kell minden adatmódosításnál újra bejelentkeznie, így a felhasználók zavartalanul használhatja az oldalt.

A RegistrationController segít a versenyre leadott regisztrációknak a kezelésében. Ez a controller a következő metódusokat tárolja:

A store metódus eltárolja a leadott regisztrációt (Képes egyszerre többet is eltárolni).

Lekéri a teljes json tömböt, és ha nincs benne semmi akkor hibaüzenettel tér vissza. A screen shot of a computer program

AI-generated content may be incorrect.

**11. ábra: Registration Controller teli változó**

Amennyiben van a tömbben adat végig megy azokon és beállítja a validációs szabályokat. Abban az esetben kell változtatni rajta, ha a regisztráció egy olyan kategóriában történt, ami páros típusú, ilyenkor kötelező lesz legalább a partner nevét megadni. A screen shot of a computer program

AI-generated content may be incorrect.

**12. ábra: Validálás**

Azt is ellenőrzi, hogy az adott személy ne tudjon többször ugyan azon a versenyen és ugyan abban a kategóriában regisztrálni.

A computer screen with text

AI-generated content may be incorrect.

**13. ábra: Leadott nevezés duplikálásának tiltása**

A helyes nevezések bekerülnek a $correctRegistrations tömbbe, és feltöltésre kerülnek a táblába.

A computer screen with text on it

AI-generated content may be incorrect.

**14. ábra: Feltöltés**

A destroy metódus az adott regisztrációt törli a táblából

A computer screen with text

AI-generated content may be incorrect.

**15. ábra: Regisztráció törlése**

A showUserRegistrations metódus lekérdezi az adott felhasználó által leadott összes regisztrációt egy lekérdezéssel, amiben csatlakoztatjuk a categories táblát.

A computer screen shot of text

AI-generated content may be incorrect.

**16. ábra: Felhasználó regisztrációinak megjelenítése**

Az exportExcel metódus pedig az nevezéseket tölti le egy excel táblázatba, ennek működéséről később részletesen beszélünk.

A black background with white text

AI-generated content may be incorrect.

**17. ábra: Regisztrációk exportálása**

A CategorieControllerben egyetlen metódus található, aminek a feladata, hogy visszaadja a aktuális versenyhez tartozó kategóriákat.

A screen shot of a computer program

AI-generated content may be incorrect.

**18. ábra: Kategória küldése**

A ContestControllerben az index metódus felelős azért, hogy az összes versenyt megjelenítse

A computer screen shot

AI-generated content may be incorrect.

**19. ábra: Összes verseny lekérése**

A destroy metódus törli az adott versenyt a táblából, ha létezik a lekért verseny.

A computer screen shot of a program code

AI-generated content may be incorrect.

**20. ábra: Verseny törlése**

A store metódus pedig a verseny tárolását hajtja végre. Először validáljuk a létrehozott verseny adatait,

A screen shot of a computer code

AI-generated content may be incorrect.

**21. ábra: Új verseny validálása**

majd a kategóriákat és a besorolások díját átalakítjuk, hogy a későbbiekben megfelelően felhasználhassuk (egy assoc tömbként kell encodolni) és eltároljuk az új versenyt. A computer screen shot of a program code

AI-generated content may be incorrect.

**22. ábra: Adat átalakítása és új verseny eltárolása**

### Excel tábla generálása

A projektünk egyik legfontosabb része, hogy a regisztrációk egy excel táblába kerüljenek. Ehhez a maatwebsite/excel csomag segítségével sikerült a következő megoldást felépíteni.

A RegistrationsExport osztály egy versenyre vonatkozó összes nevezését exportálja nevekre csoportosítva. A versenyző neve után az oszlopok az adott verseny kategóriáit sorolja fel és alattuk a versenyző besorolásának megfelelő fizetendő összeg szerepel, amennyiben nevezett az adott kategóriában. Majd az utolsó oszlopban a versenyző nevezésének teljes összege található.

A screenshot of a table

AI-generated content may be incorrect.

**23. ábra: Nevezések excel formája**

RegistrationsExportban először lekérdezzük az adott verseny nevezéseit, hozzácsatolja a kategória táblát és a jsonként tárolt versenyző adatai is lekérdezi. Külön lekérdezésre kerülnek még a verseny kategóriák.

A computer screen shot of a program code

AI-generated content may be incorrect.

**24. ábra Szükséges adatok lekérdezése az excelhez**

Utána csoportosítjuk a nevezéseket név szerint. Minden névhez készítünk egy tömböt, ahol a kulcsok a kategóriák és az értékek pedig a díjak. Ha nincs az adott kategóriában a nevezett személynek nevezése, akkor üres marad a mező.

A computer code with many colorful text

AI-generated content may be incorrect.

**25. ábra: Csoportosítás név szerint az excelben**

Elkészítjük a fejlécet utána pedig a elkészítjük minden versenyzőnek a sorát és kiszámoljuk a végösszeget.

A computer screen shot of a program code

AI-generated content may be incorrect.

**26. ábra Excel sorok elkészítése és fejléc beállítása**

A FullcompetitinExport feladata a munkalapok összefogása. Itt először meghívásra kerül a nevezések munkalap, amit az imént mutattunk be és utána generál annyi munkalapot, ahány kategória van a versenyen (a verseny azonosítója a contruktorban kerül átadásra).

A computer screen shot of a program code

AI-generated content may be incorrect.

**27. ábra: Munkalapok összefoglalása**

Az EmptyCategoriySheet-ben történik a kategóriára nevezett versenyzők felsorolása pontok szerint csökkenő sorrenben. Először lekérdezzük, hogy a kategória típusa egyéni vagy páros.

A screen shot of a computer program

AI-generated content may be incorrect.

**28. ábra: Kategória típusának megállapítása**

Utána, ha kategória típusa egyéni akkor csak készít egy lekérdezést és kiírja a versenyző nevét és pontját csökkenő sorrendbe

A screen shot of a computer screen

AI-generated content may be incorrect.

**29. ábra: Az egyéni kategóriák munkalapjának feltöltése**

Ha páros a kategória, akkor lekérdezésre kerül a versenyző párja is és ellenőrzésre kerül, hogy ha az adott kategóriában a pár mind a két tagja leadja a nevezését akkor ne jelenjen meg kétszer a felsorolt versenyzők között a páros.

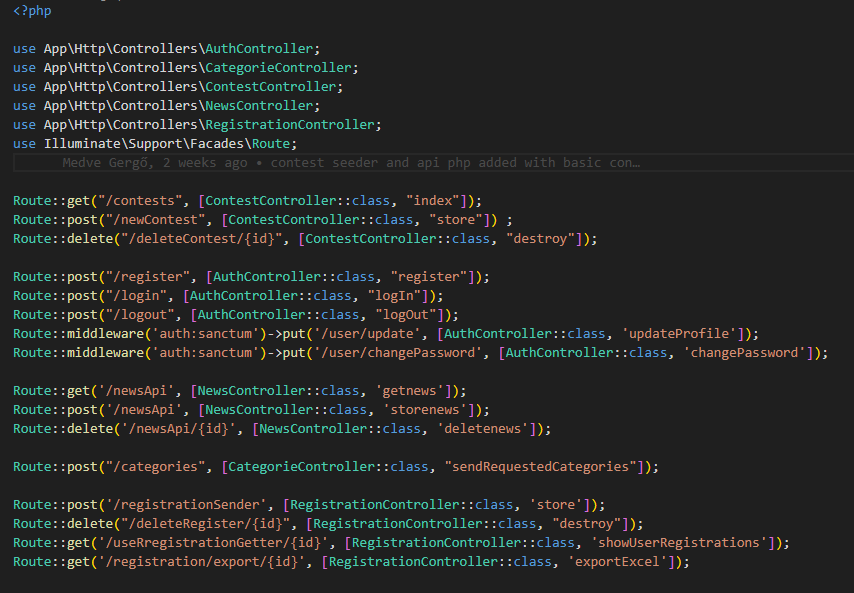
A screen shot of a computer program

AI-generated content may be incorrect.

**30. ábra Páros kategória munkalap feltöltése**

### API útvonalak

A projectünkben a következő API útvonalakat használjuk. Laravel Sanctum segítségével oldottuk meg az authentikációt és levédtünk bizonyos útvonalakat, amik csak bejelentkezett felhasználó férhet hozzá.



**31. ábra: Api útvonalak**

## Frontend

### Felépítés

A CsocsoWeb projektünk frontend része Vite build tool segítségével készült, és a felhasználói felületet React könyvtárral valósítottuk meg. Az útvonalak kezelése a React Router DOM könyvtár segítségével működik, és ez teszi lehetővé, hogy az oldalak között úgy navigáljunk, hogy az oldal ne töltődjön újra. Az App.jsx az alkalmazásunk fő komponense, amelyen keresztűl történik az útvonal választása a (react-router-dom) segítségével és ezen felül az állapotkezelést (useState) is kezeli.

A computer screen shot of a program code

AI-generated content may be incorrect.

**32. ábra: App.jsx felépítése**

A Header egy navigációs sáv (navbar) komponens, amely minden oldalon megjelenik, és segíti a felhasználót az oldalak közötti váltásban. A react-router-dom segítségével történik a navigáció. A komponens figyeli azt is, hogy a felhasználó be van e jelentkezve, ezt az userContext használatával teszi. Amennyiben a versenyző nincs bejelentkezve a Bejelentkezés menüpont jelenik meg, viszont, ha bejelentkezett, akkor e pont helyett már a profil gomb jelenik meg. A reszponzivítás pedig az oldalon, oly módon valósul, meg hogy mobil nézetbe egy lenyitható mobil ikon segítségével történik, amelyre kattintva lenyílóan jelennek meg a menüpontok

A Footer egy egyszerű lábléc komponens, amely az oldal alján jelenik meg, és információt nyújt a látogatóknak a Viharsarki csocsó egyesülettel kapcsolatban. Tartalmaz elérhetőségi adatokat (cím, telefonszám, email), valamint közösségi média hivatkozásokat is, ikonokkal kiegészítve.

A FontAwesome ikonok segítenek a vizuális megjelenítésben, amelyeket egy import segítségével tudjuk teljesen ingyenesen használni. Reszponzívan lett kialakítva, így kisebb képernyőkön is jól olvasható. Statikus elemeket tartalmaz, és nem használ állapotkezelést vagy eseménykezelőt.

A komponensek olyan önálló egységek, amelyeket meghívva az oldalra egy összefüggő dinamikus egységet képeznek így áll össze egy oldal. A mi projektünkben kimondott komponensek közül 5 darab van, ezek közül 2 állandó használatban és a components nevű mappában találhatók meg. A componentsek mellett található még egy pages mappa, ahol az oldalakat tároljuk.

A screenshot of a computer

AI-generated content may be incorrect.

**33. ábra: Frontend mappa struktúra**

Ha beletekintünk például a Contests.jsx-be, akkor a látjuk, hogyan épülnek a react fileok. File tetején láthatjuk azokat az importokat, amiket az oldalkészítés alatt felhasználtunk.

A screen shot of a computer

AI-generated content may be incorrect.

**34. ábra: React importok**

Utána láthatjuk, hogyan oldottuk meg azt, hogy bizonyos részeket csak az admin jogosultságú felhasználó érhessen el.

A screen shot of a computer

AI-generated content may be incorrect.

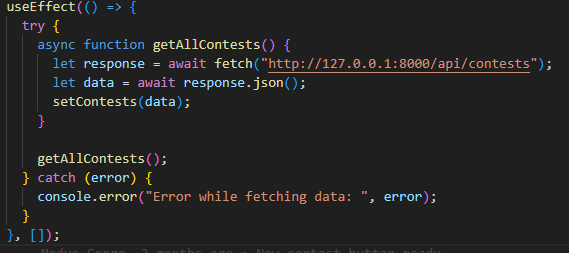
**35. ábra: Admin opciók elrejtése**

Utána következnek az állapotok, amiket létrehoztunk az adott oldal működéséhez



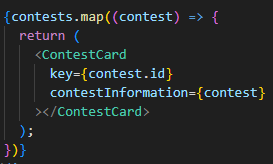
**36. ábra: Felhasznált useState**

Ezt követi egy useEffect, ami csak a componens mountolásakor futtatja le a benne lévő api hívást, ami lekérdezi az összes versenyt az adatbázisban. A lekérdezéshez async metódust használtunk, amivel bevárhatjuk az api kérés eredményét, majd azt egy változóban eltároljuk.



**37. ábra: UseEffectben történő api hívás**

Utolsó lépésnek bejárjuk az eredményül kapott listát a versenyekről és visszaadunk egy componenst, aminek átadjuk az összes információt a versenyről. Azoknak a feldolgozása pedig a ContestCard componensen belül történik.



**38. ábra: Lekérdezett versenyek megjelenítése**

### Reszponzivítás

Teljes munkánk során nagyon igyekeztünk a reszponzivítás megtartására, amelyhez a tailwind css keretrendszert használtunk. Ennek segítségével a HTML kódunkban rövid előre definiált osztályok segítségével tudtunk reszponzív elemeket, illetve stílus elemeket is állítani oldalainknak. Ez nagyon hasznos volt számunkra, hiszen a tailwind segítségével ebben az igen nagy terjedelmű projektben megspóroltunk rengeteg css stílusú fájl létrehozását.

### Minták

Kis kijelzőnél a navbart egy lekicsinyítettük és a következő formában jelenik meg.

A green square with white border

AI-generated content may be incorrect.

**39. ábra: Reszponzív navbar**

Ha a felhasználó rányom akkor legördül egy lista a lehetséges navigációs útvonalakkal.

A green rectangular object with white text

AI-generated content may be incorrect.

**40. ábra: Reszponzív lenyíló navbar**

A versenyek is egy teljesen új kinézetet szereznek, amiben már egymás alatt jelennek meg az információk.

A screenshot of a phone

AI-generated content may be incorrect.

**41. ábra: Reszponzív verseny kiírás**

Az űrlapok is változnak sokkal kompaktabbak és kezelhetőbbek a felhasználó számára.

A screenshot of a computer

AI-generated content may be incorrect.

**42. ábra: Reszponzív verseny regisztráció**

## Tesztek

A szoftvertesztelés célja, hogy biztosítsuk: az alkalmazás minden része megfelelően működik, és nem okoz problémát a felhasználás során. Mivel a projekt Laravel keretrendszerre épül, többféle módon is lehetőség van az alkalmazás logikájának ellenőrzésére. A tesztelés három fő területre osztható:

1. **Egységtesztek (Unit tests):** Ezek az egyes osztályokat és metódusokat vizsgálják külön, más rendszerektől függetlenül.
2. **Feature tesztek:** Itt már nagyobb egységek működését nézzük, például egy API végpont teljes folyamatát.
3. **End-to-End (E2E) tesztek:** Ezek a valós felhasználói viselkedést szimulálják – általában böngészőautomatizálással (pl. Laravel Dusk).

### Miért controller tesztekre fókuszáltunk?

Ebben a projektben elsősorban a controllerek tesztelésére koncentráltunk. Ennek több oka is volt:

* **Az API működése kulcsfontosságú:** Az alkalmazás fő funkciói HTTP kéréseken keresztül érhetők el, így logikus volt az API végpontok mögött álló controllerek alapos tesztelése.
* **Az adatok helyes kezelése itt történik:** Mivel a controllerek felelősek a beérkező adatok feldolgozásáért, fontos volt meggyőződnünk arról, hogy ezek a folyamatok hibátlanul zajlanak le.
* **Kiegyensúlyozott tesztelési szint:** Az egységtesztek gyakran túlságosan részletekbe mennek, az E2E tesztek pedig nehezen karbantarthatók. A controller tesztek viszont jól lefedik a lényegi működést, mégis viszonylag egyszerűen kezelhetők.
* **Közelebb áll a felhasználói élményhez**: Ezek a tesztek valós API interakciókat modelleznek, így jobban tükrözik, amit a felhasználó is tapasztalna.

A Laravel **php artisan test** parancsával könnyen futtathatók ezek a tesztek. A háttérben MySQL adatbázist használunk, így életszerű környezetben tudjuk ellenőrizni az adatkezelést is.

### Használt technológiák a teszteléshez

A tesztelési környezetben az alábbi eszközökre és verziókra támaszkodtunk:

* **Laravel 12.0**
* **PHP 8.2**
* **MySQL** (valós adatokkal)
* **Composer** (csomagkezelés)
* **Node.js + npm** (a Vite támogatásához)
* **PHPUnit** (a controller tesztekhez)
* **Laravel Sanctum** (API tokenekhez)
* **DatabaseTransactions trait** – ez gondoskodik arról, hogy a tesztek után minden adatbázis-módosítás visszaálljon

### Feature tesztek a gyakorlatban

A cél az volt, hogy az alkalmazás **fő funkcióit ellenőrizzük**. Minden esetben az **API végpontokat** hívtuk meg, és figyeltük, hogy a válaszok megfelelnek-e az elvárásoknak (pl. validálás, státuszkód, adat visszatérés stb.).

### CategorieControllerTest.php

* Lekérdezés valós kategória ID-val – a megfelelő adatokat kaptuk vissza.
* Nem létező ID esetén az API nem küldött hibás vagy felesleges adatot.
* Üres lekérdezésekre is megfelelő választ adott az API.

### ChangePasswordTest.php

* Sikeres bejelentkezésnél a felhasználó tokent kapott.
* Hibás jelszónál nem engedte be a felhasználót.
* Minden esetben ellenőriztük a válaszkódokat is.

### ContestControllerTest.php

* **Versenyek lekérdezése:** az API megfelelő formátumban adta vissza az adatokat.
* **Új verseny létrehozása:** csak akkor járt sikerrel, ha minden kötelező mezőt megadtunk.
* Validáció hiányzó mezők esetén is működött – ilyenkor hibaüzenetet kaptunk.

### LoginTest.php

* A jelszómódosítás akkor működött, ha a régi és új jelszót is helyesen adtuk meg.

### NewsControllerTest.php

* **Hírek listázása:** minden elérhető hír visszajött.
* **Létrehozás:** az admin új hírt tudott rögzíteni.
* **Törlés:** meglévő hír esetén az elem ténylegesen eltűnt az adatbázisból.
* Validáció itt is rendben működött, ha például kimaradt a cím vagy a tartalom.

### RegisterTest.php

* **Regisztráció:** ha minden mező megfelelő volt, a regisztráció sikeres lett.
* Az új felhasználó adatai valóban bekerültek az adatbázisba.
* Rosszul megadott vagy hiányzó adatoknál az API hibát jelzett.

### UpdateProfileTest.php

* A felhasználók módosíthatták profiljukat, például az e-mail címet, nevet vagy nemet.
* Ezeket a változtatásokat csak bejelentkezett felhasználók hajthatták végre.
* A frissített adatok valóban elmentődtek, és validálási hibák esetén megfelelő visszajelzést kaptunk.

A screenshot of a computer

AI-generated content may be incorrect.

**43. ábra: Sikeres tesztek**

# Felhasználói dokumentáció

Üdvözöllek!

Köszöntünk a Viharsarki csocsó oldalán, amely a csocsó versenyeken való könnyebb nevezésben és az egyszerűbb nyomon követésben segít neked.

A viharsarki csocsó weboldala egy egyszerű regisztráció után meg is nyílik a számodra. A regisztrációt, ha még nem tetted meg, ne aggódj néhány egyszerű lépés és már élvezheted is az oldal előnyeit.

Az oldalunk segítségével nevezést adhatsz le versenyekre és nyomon követheted azok helyét, idejét és nevezésének költségeit. Valamint ezután a profilodon, nyomon követheted nevezéseidet.

Ha azonban versenyszervező vagy, akkor egy admin bejelentkezéssel, (melyben a csapat bármikor segítségedre van) azt is megteheted és egy egyszerű form kitöltésével már is megnyílik a versenyzők számára a jelentkezés, és a lezárta után már csak egy kész excel táblával kell, dolgozz.

A viharsarki csocsó segítségével könnyítheted a verseny létrehozással és nevezéssel járó stresszt, így neked már csak élvezned kell a csocsó rendkívül izgalmas versenyeit.

## Rendszerkövetelmények

* A weboldalunk minden modern webböngészőn futtatható, például: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari
* A weboldalunk teljes mértékig reszponzív így minden kijelző felbontáshoz tökéletesen alkalmazkodik.

## Weboldal futtatása

A weboldalunk futtatásához a következő lépéseken sorban haladj végig.

1. Telepítsd le a szükséges követelményeket. A mi esetünkben szükséged lesz
   1. XAMPP
   2. PHP
   3. Node.js
   4. Composer
   5. Git
2. Miután ezeket letelepítetted néhány figyelemre méltó változtatást is el kell, végezz
   1. A XAMPP php.ini fájljában a következő sorokat kell változtatnod:
      * Bizonyosodj, meg hogy e bővítmények elöl, mindenféleképpen ki legyen, véve a komment:

extension=mysqli

extension=pdo\_mysql

extension=curl

extension=mbstring

extension=openssl

extension=zip

extension=gd

Amennyiben további hibakód jelenik meg a bővítményekkel kapcsolatban a hibakód szerint kell eljárni.

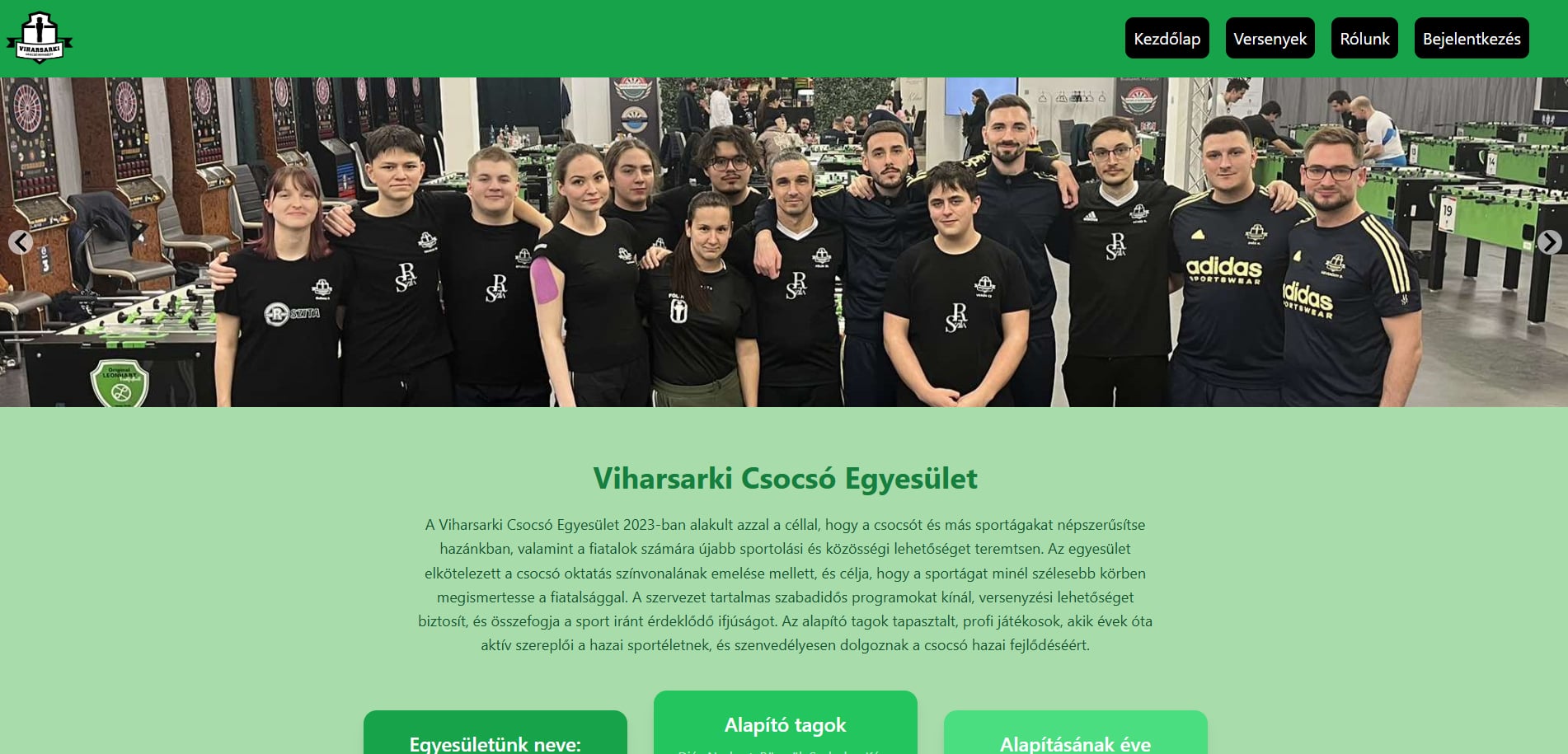
* 1. A php környezeti változói legyenek beállítva.

1. Követő lépés a projekt lemásolása kell, hogy legyen. Ezt legegyszerűbben a Git clone parancs kiadása után (egy terminálba) a Git repository linkjének megadásával tehetünk meg. Ez a link a mi esetünkben: <https://github.com/medve-g/CsocsoWeb.git>
2. XAMPP Apache és MySQL indítása a XAMPP control panel segítségével. 

**44****. ábra: XAMPP control panel**

1. Lépj be a projekt mappába a terminálon keresztül.
2. Majd először a CsocsoFrontend mappába, és add ki a következő utasításokat
   1. npm install – Ez segít a függőségek telepítésében
   2. npm run dev – Ezzel tudod a Frontend szerver részt elindítani
   3. Kattints a megjelenő localhost linkre
3. Majd lépj át a CsocsoBackend mappába és ide pedig ezt a parancsot add ki
   1. composer install – Segít a Backend függőségek letöltésében
   2. php artisan migrate – Ezzel tudod a migrációkat feltölteni, illetve ezzel automatikusan létre tudod hozni az adatbázist is
   3. php artisan db:Seed – Ennek segítségével tudod a táblákat a megfelelő kezdő adatokkal feltölteni
   4. php artisan serve – A backend szerver elindítása
4. Ha minden lépést elvégeztél, akkor az 5. lépésben megjelenő linken, amely a böngésződben már fut, a teljes projektet láthatod.

## Weboldal használatának részletes ismertetése



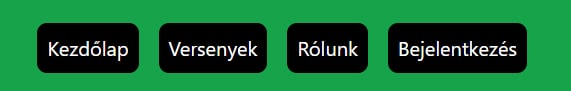
**45****. ábra: Kezdőlap**

A CsocsoWeb kezdő oldalát látod, amelyen a Kezdőlap, Versenyek, Rólunk és Bejelentkezés fülek szerepelnek.

* A kezdőlap gomb minden helyzetben a vissza irányít az alap oldalra. Ezen az oldalon találkozhatsz az egyesületről néhány alap információval, valamint lejjebb görgetve az egyesületről és versenyekről való hírekkel is.
* Az éppen aktuálisan megrendezés előtt álló versenyek megtekintéséhez a Versenyek gombra kattints. Majd a felsorolt versenyek egyikére kattintva megjelenik a regisztrációs űrlap, amennyiben a bejelentkezés már megtörtént. Addig viszont csak a verseny információk jelennek meg.
* A Rólunk gomb által az egyesületről tudhatsz rendkívül érdekes dolgokat. Ilyen a megalakulásának a menete illetve a céljaik és rengeteg más hasznos információt is.
* A „Bejelentkezés” gombra kattintva beléphetsz a már meglévő fiókodba vagy éppen egy új fiókot hozhatsz létre magadnak. A bejelentkezés után ez a fül „profil” fülre változik. Itt különböző a fiókkal, kapcsolatos információkkal találkozhatsz

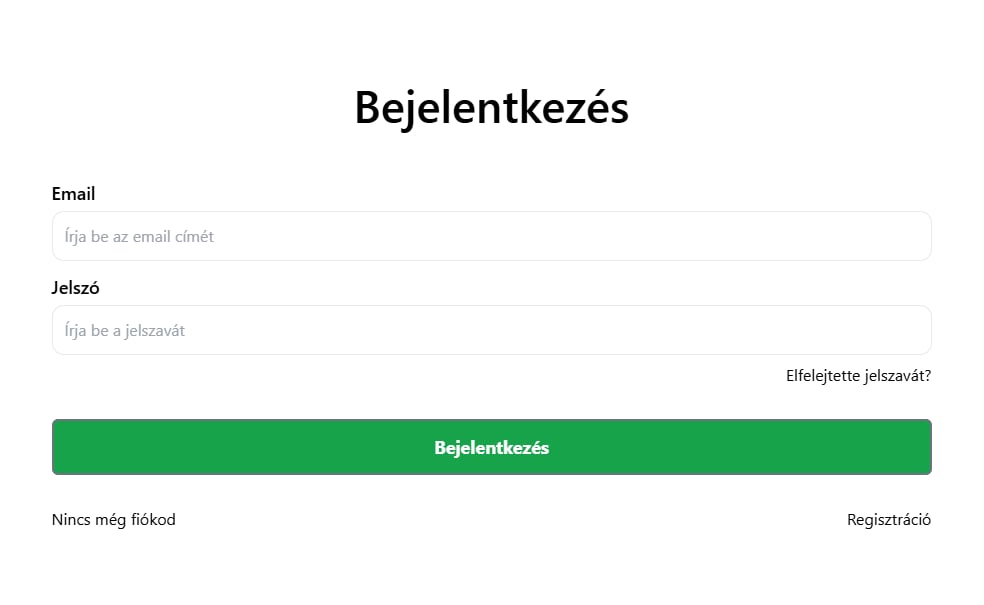
## Regisztrációs folyamatok ismertetése

**46****. ábra: Nav-bar**



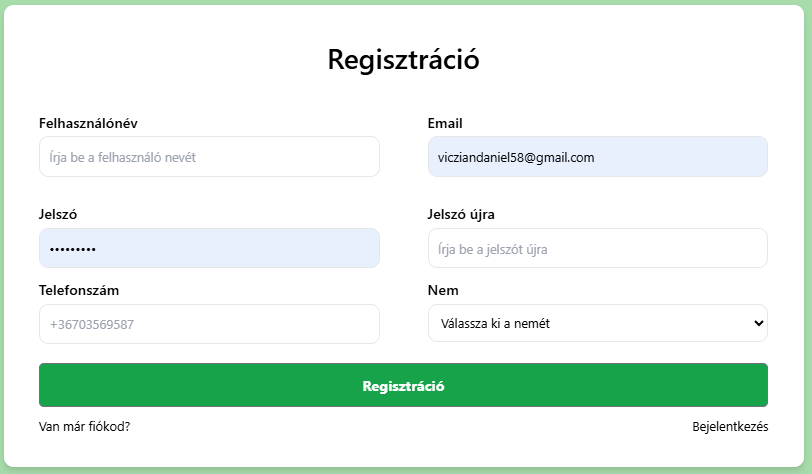
Az 5. ábrán láthatóak a gombok. Itt a belépés gombra kattintva juthatsz el a regisztrációhoz is.

**47****. ábra: Regisztráció gomb**



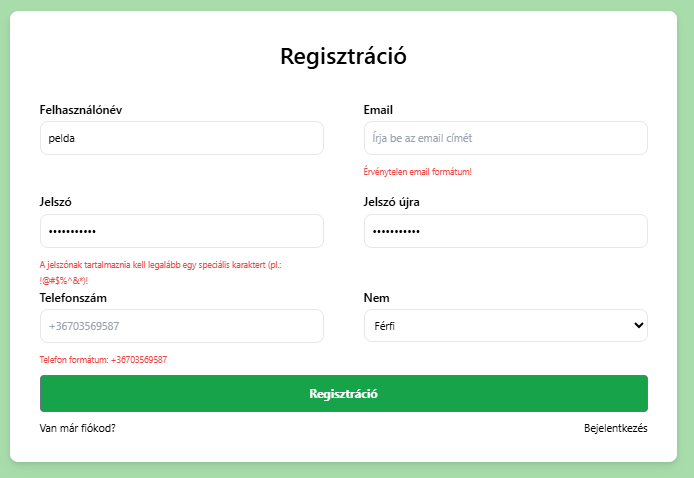
A 6. ábrán látható a Bejelentkezés gombra kattintás utáni felugró ablak. Itt a Regisztráció gombra kattintva léphetsz tovább a regisztrációhoz

**48****. ábra: Regisztráció oldal**

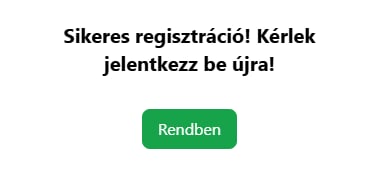


A 6. ábrán levő „regisztráció” gombra kattintva ez az oldal fogad minket, majd minden adat megadása után a „Regisztráció” gombra kattintva létre is hoztuk a felhasználói fiókunkat.  
Minden mező kitöltése szükséges, illetve a formátumok helyes megadása valamint a jelszó védelme érdekében kell, hogy speciális karakter is legyen benne. Ellenkező esetben a 6. ábrán látható hibák jelennek meg.

**49****. ábra: Hiba lehetőségek**



**50****. ábra: Sikeres regisztráció**

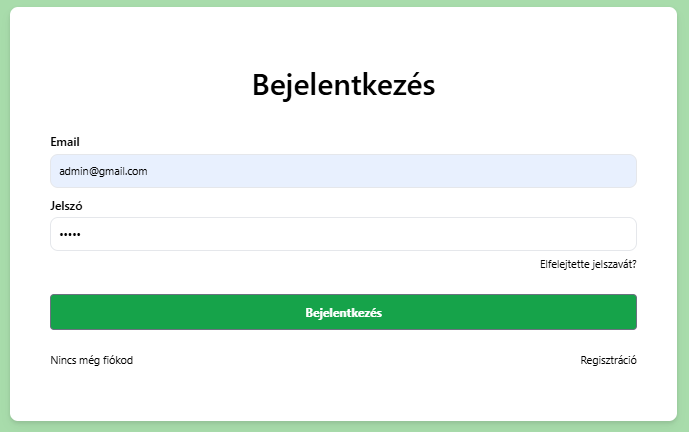


A Regisztráció megtörténése után a következő ablak jelenik meg, amely azt jelzi nekünk a regisztráció sikeresen megtörtént.

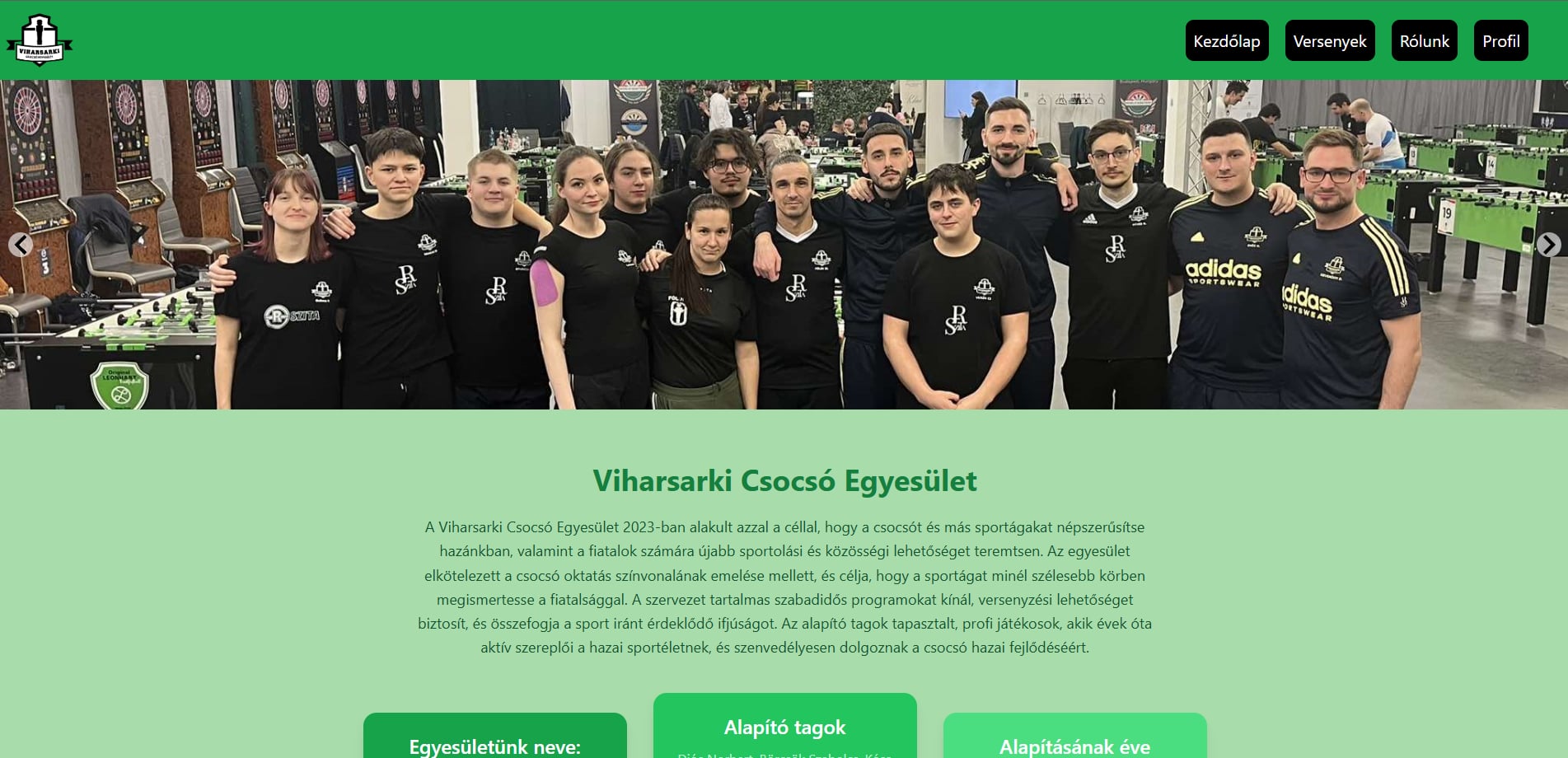
## Belépés folyamata

Az előbbi folyamat során már megmutattam hogyan lehet eljutni a bejelntkezés után, a folyamat lényegében ugyan az azonban a felugró bejelentkezési ablaknál már nem kell a „regisztráció” gombra kattintani. Megadhatod, a bejelentkezési adataidat majd kattints a bejelentkezés gombra.

**51****. ábra: Bejelentkezés**

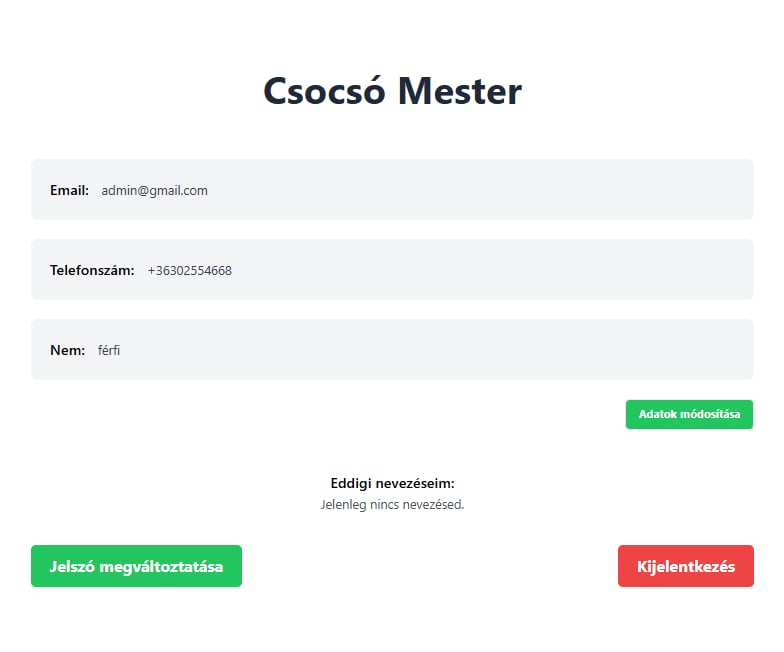


**52****. ábra: Bejelentkezett helyzet**



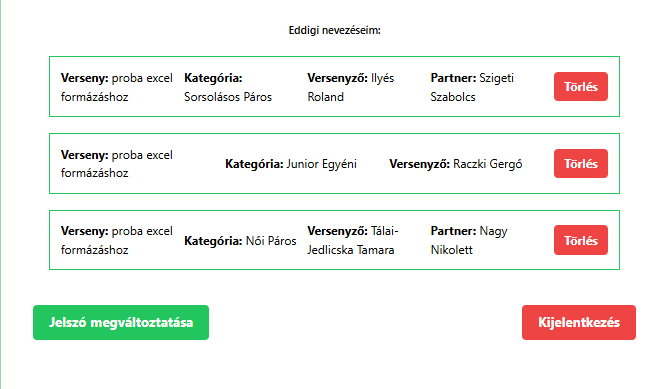
A Bejelentkezés után ez a Kezdő oldal fogad minket, ahol immáron a Bejelentkezés menüpont helyett a profil menüpont fogad minket.

**53. ábra: Profil menüpont**



A 11. ábrán látható a „profil” gombra való kattintás után elénk ugró ablak. Ezen ablak segítségével választhatunk az addig megadott adatokon, a jelszavunkat is megváltoztathatjuk, illetve ezen az oldalon tudunk kijelentkezni is a profilunkból.

**54. ábra: Eddigi nevezések**



A 12. ábrán látható a profil oldalhoz tartozó még egy nagyon fontos megjelenés, amely pedig az addig leadott nevezéseinket tekinthetjük meg, illetve törölhetjük is azokat, ha mégsem szeretnénk indulni a versenyen.

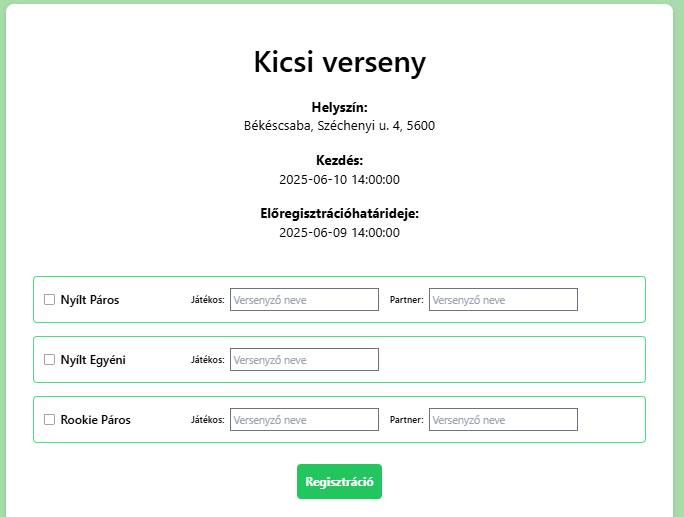
## Nevezés leadása egy versenyre

**55. ábra: Versenyek**



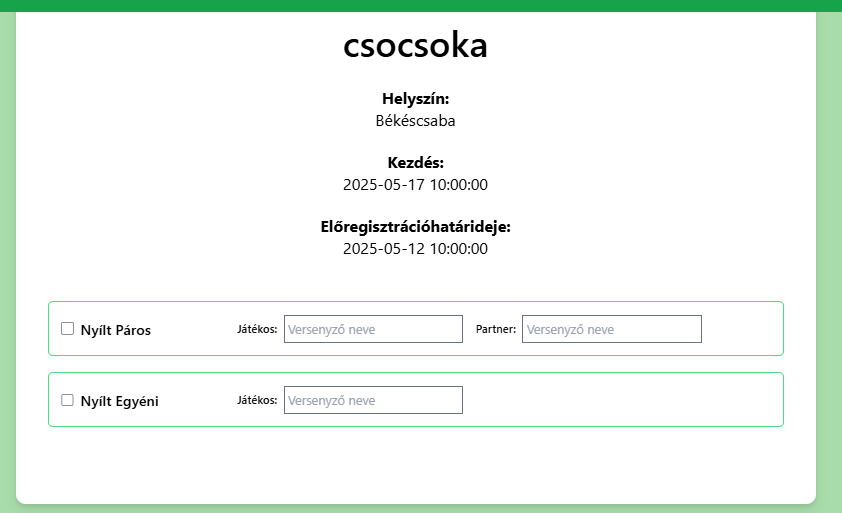
Mint azt már korábban említettük a nevezés leadás a „Versenyek” fülön történhet meg, akkor, ha már be vagyunk jelentkezve. A nevezést úgy adhatjuk le ha a zöld sávba kattintunk az adott versenyen, ekkor átdob minket a verseny panelhez ahol a nevezés megtörténhet.

**56. ábra: Regisztráció**



A versenyre való regisztrációs lehetőség a 14. ábrán látható. A verseny egy adott kategóriába úgy lehet nevezni, hogyha a mellette levő radio gombot bepipáljuk ezzel jelezve szeretnénk abban a versenyszámban részt venni. Azt is megfigyelhetjük, hogy az elő regisztráció határideje is meg van jelenítve. Ez azért nagyon fontos, mert a versenynél csak addig jelenik meg alul a regisztráció gomb ameddig az elő regisztráció aktív, ha ez azonban lejár, akkor a versenyre a további nevezés nem lehetséges. Lásd 15. ábra

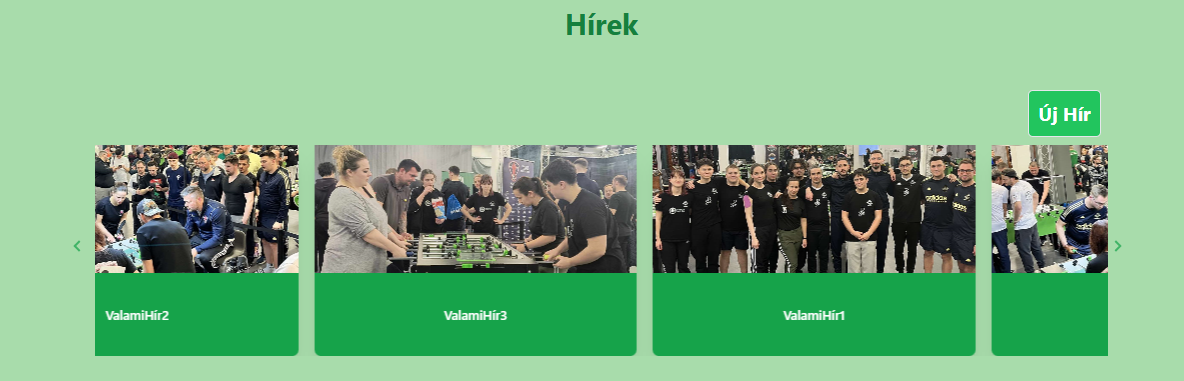
**57. ábra: Lezárult nevezés**



## Admin extra lehetőségei:

Admin jogosultsága egyelőre Földi Zsolt profiljának van azonban a későbbiekben kérvényezhető a profil admin-i jogosultságkörrel való bővítése. Az admin extra lehetőségekkel rendelkezik, ilyen például a Kezdőoldalon található hír menüpont, amelyekbe új híreket vehet, fel lásd 16-17.ábra.

**58. ábra: Új hír hozzáadása**

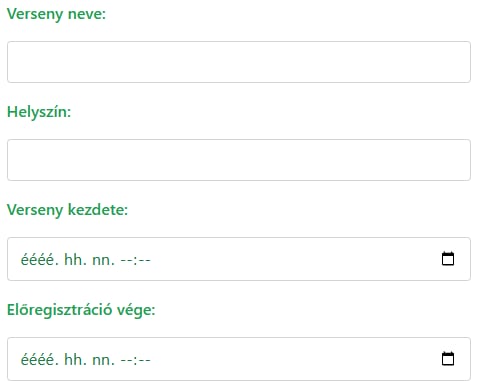


**59. ábra: Új hír**



Ezen felül új versenyeket is hozhat létre a „Versenyek” menüpontban az „új verseny” gombra kattintva. Amelyben meghatározhat minden fontos információt a versennyel kapcsolatban. Megadhatja verseny időpontját, a nevezési határidőt, illetve az egyes besorolásokhoz tartozó befizetendő összeget is. Illetve kijelölheti azokat a kategóriákat is, amelyeket szeretné az adott versenyen lásd 18-19-20. ábra

**60. ábra: Verseny létrehozás pt1**



**61. ábra: Verseny létrehozás pt2**

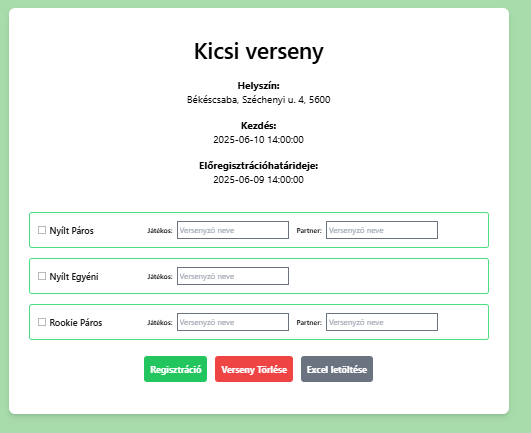


**62. ábra: Versenyek létrehozása pt3**



Az admin egy olyan joggal is rendelkezik, hogy amikor rákattint egy adott versenyre két újabb gomb jelenik meg neki alul a versenyre való nevezési lehetőség alatt. Az egyik a „verseny törlése” gomb, amellyel az adott versenyt törölheti a listáról, így az onnantól kezdve nem lesz elérethető továbbá a felhasználóknak jelentkezésre. A másik pedig az „excel letöltése”, amely egy excel fájlt tölt le, amelyben a versenyre nevezett versenyzők vannak kategóriákra szétbontva ki indul az egyes kategóriákban, illetve hogy mennyi nevezési díjat kell befizetniük.

**63. ábra: Verseny törlés és Excel letöltés**



### Végszó

Mostanra minden szükséges információ rendelkezésedre áll ahhoz, hogy sikeresen elindítsd és használd a CsocsoWeb alkalmazást. A dokumentáció végigvezetett a rendszer elindításához szükséges lépéseken, bemutatta a funkciókat és eligazítást nyújtott a használathoz. Akár adminisztrátorként, akár versenyzőként használod a rendszert, a CsocsoWeb célja, hogy egyszerűvé és átláthatóvá tegye a csocsóbajnokságok szervezését és lebonyolítását. Ha minden lépést követtél, már csak egy dolgod maradt: kezd el használni a rendszert, és élvezd a játékot!

# Összegzés

## A munkánk értékelése

A munkánk projekt része immáron elkészült. Úgy vélekedünk rendkívüli teljesítményt nyújtottunk mind a hárman és megtanultuk a csapatban dolgozás menetét, annak előnyeit, na és persze természetesen néhány hátránnyal is találkoztunk. A projektünkkel azonban mindennek a végén nagyon elégedettek vagyunk, hiszen egy kiindulási alapnak a további fejlődésnek tökéletesen megfelel.

Természetesen sok éjszakába nyúló agyalás és hiba javítás áll a hátunk mögött, azonban ez mind megérte, hiszen rendkívül sokat tanultunk ismét a programozás nyelvén, és ezek segítettek abban, hogy a jövőben még jobb programozókká válhassunk. Kiindulási pontnak azt tűztük ki célul, hogy egy olyan weboldalt hozzunk létre, amely egy minimalista és egyszerű bejelentkezés után nevezést tudjanak leadni az előre általunk létrehozott versenyeken. A végkifejlet most már állíthatom ezen elképzeléseinket teljesen túlszárnyalta, és bár persze vannak továbbra is hibáink, amelyeket látunk az oldalunkon, azonban ez ösztönöz bennünket arra, hogy a későbbiekben is dolgozzunk az oldalon és a lehető legtöbbet kihozzuk belőle.

Rengeteg további ötlet kavarog bennünk, amelyekre mi mind, mint egy kiaknázatlan lehetőségre tekintünk és szeretnénk a jövőben is tovább dolgozni ezeken. Ide sorolandó többek között egy olyan bővítés, amellyel már a sorsolás is megvalósítható lenne oldalunkon így mindent magában foglalna és a versenyzőknek megkönnyítve dolgát az oldalukon, nyomon tudnák követni az adott versenyre vonatkozó sorsolást is illetve annak haladását. Valamint egy felhasználók által testre szabható profil bevezetésén is dolgozni szeretnénk, amellyel azt szeretnénk elérni, hogy minden versenyzőt a saját tetszésének megfelelő profil fogadjon, ezzel remélhetőleg elérve, hogy még könnyebben zökkenő mentesebben történjen meg az átállás a facebook nevezésről. Illetve szeretnék azt is elérni, hogy az általunk generált excel tábla is további funkciókkal bővüljön, mint az egyes kategóriákban vissza osztásra kerülő pénzösszegek automatikus szétosztása illetve részletesebb adatok ismertetését is el szeretnénk érni benne.

## Köszönetnyilvánítás

A következő néhány sorban szeretnénk köszönetet mondani mindenkinek, aki a teljes munkánk során mögöttünk állt és segítő kezet nyújtott nekünk minden formában. Hatalmas köszönettel tartozunk a családunknak, akik minden helyzetben támogattak és elviseltek minket a hajnalba nyúló munkáinkkal és hangoskodásunkkal együtt. A munkánkat végig követő és segítő Viharsarki csocsó egyesületnek, akik bármely kérdésünket egyből meghallgatva azonnali segítségünkre siettek és rendkívül empatikus módon támogattak mindenben. Nem utolsó sorban pedig szeretnénk megköszönni tanárainknak a tudást, amelyet az évek alatt belénk tápláltak és segítettek elindulni azon az úton, amelyet a jövőnknek is nevezhetünk. Ezúton is köszönjük mindenkinek, hiszen nélkülük soha nem juthattunk volna, el ide ahol most tartunk.

# Irodalomjegyzék

<https://stackoverflow.com>

<https://www.w3schools.com>

<https://www.php.net>

<https://www.geeksforgeeks.org>

<https://fontawesome.com/>

<https://laravel-excel.com/>

<https://laravel.com/docs/12.x>

[<https://vite.dev/>](https://fontawesome.com/)

# Ábrajegyzék

[**1. ábra: Adatbázis diagram** 6](#_Toc198063372)

[**2. ábra: A táblák generálása** 8](#_Toc198063373)

[**3. ábra: Tábla seederek** 9](#_Toc198063374)

[4**. ábra: Modellek** 9](#_Toc198063375)

[**5. ábra: A hírek beszerzése** 10](#_Toc198063376)

[**6. ábra: Hírek eltárolása** 10](#_Toc198063377)

[**7. ábra: A hírek törlése** 11](#_Toc198063378)

[**8. ábra: Token generálása** 11](#_Toc198063379)

[**9. ábra: Api hitelesítése** 12](#_Toc198063380)

[**10. ábra: Védett útvonalak** 12](#_Toc198063381)

[**11. ábra: Registration Controller teli változó** 13](#_Toc198063382)

[**12. ábra: Validálás** 13](#_Toc198063383)

[**13. ábra: Leadott nevezés duplikálásának tiltása** 14](#_Toc198063384)

[**14. ábra: Feltöltés** 14](#_Toc198063385)

[**15. ábra: Regisztráció törlése** 15](#_Toc198063386)

[**16. ábra: Felhasználó regisztrációinak megjelenítése** 15](#_Toc198063387)

[**17. ábra: Regisztrációk exportálása** 15](#_Toc198063388)

[**18. ábra: Kategória küldése** 16](#_Toc198063389)

[**19. ábra: Összes verseny lekérése** 16](#_Toc198063390)

[**20. ábra: Verseny törlése** 17](#_Toc198063391)

[**21. ábra: Új verseny validálása** 17](#_Toc198063392)

[**22. ábra: Adat átalakítása és új verseny eltárolása** 18](#_Toc198063393)

[**23. ábra: Nevezések excel formája** 19](#_Toc198063394)

[**24. ábra Szükséges adatok lekérdezése az excelhez** 19](#_Toc198063395)

[**25. ábra: Csoportosítás név szerint az excelben** 20](#_Toc198063396)

[**26. ábra Excel sorok elkészítése és fejléc beállítása** 20](#_Toc198063397)

[**27. ábra: Munkalapok összefoglalása** 21](#_Toc198063398)

[**28. ábra: Kategória típusának megállapítása** 22](#_Toc198063399)

[**29. ábra: Az egyéni kategóriák munkalapjának feltöltése** 23](#_Toc198063400)

[**30. ábra Páros kategória munkalap feltöltése** 24](#_Toc198063401)

[**31. ábra: Api útvonalak** 25](#_Toc198063402)

[**32. ábra: App.jsx felépítése** 26](#_Toc198063403)

[**33. ábra: Frontend mappa struktúra** 27](#_Toc198063404)

[**34. ábra: React importok** 27](#_Toc198063405)

[**35. ábra: Admin opciók elrejtése** 28](#_Toc198063406)

[**36. ábra: Felhasznált useState** 28](#_Toc198063407)

[**37. ábra: UseEffectben történő api hívás** 28](#_Toc198063408)

[**38. ábra: Lekérdezett versenyek megjelenítése** 29](#_Toc198063409)

[**39. ábra: Reszponzív navbar** 29](#_Toc198063410)

[**40. ábra: Reszponzív lenyíló navbar** 30](#_Toc198063411)

[**41. ábra: Reszponzív verseny kiírás** 31](#_Toc198063412)

[**42. ábra: Reszponzív verseny regisztráció** 32](#_Toc198063413)

[**43. ábra: Sikeres tesztek** 36](#_Toc198063414)

[**44. ábra: XAMPP control panel** 38](#_Toc198063415)

[**45. ábra: Kezdőlap** 39](#_Toc198063416)

[**46. ábra: Nav-bar** 40](#_Toc198063417)

[**47. ábra: Regisztráció gomb** 40](#_Toc198063418)

[**48. ábra: Regisztráció oldal** 41](#_Toc198063419)

[**49. ábra: Hiba lehetőségek** 41](#_Toc198063420)

[**50. ábra: Sikeres regisztráció** 42](#_Toc198063421)

[**51. ábra: Bejelentkezés** 42](#_Toc198063422)

[**52. ábra: Bejelentkezett helyzet** 43](#_Toc198063423)

[**53. ábra: Profil menüpont** 43](#_Toc198063424)

[**54. ábra: Eddigi nevezések** 44](#_Toc198063425)

[**55. ábra: Versenyek** 45](#_Toc198063426)

[**56. ábra: Regisztráció** 45](#_Toc198063427)

[**57. ábra: Lezárult nevezés** 46](#_Toc198063428)

[**58. ábra: Új hír hozzáadása** 47](#_Toc198063429)

[**59. ábra: Új hír** 47](#_Toc198063430)

[**60. ábra: Verseny létrehozás pt1** 48](#_Toc198063431)

[**61. ábra: Verseny létrehozás pt2** 48](#_Toc198063432)

[**62. ábra: Versenyek létrehozása pt3** 49](#_Toc198063433)

[**63. ábra: Verseny törlés és Excel letöltés** 49](#_Toc198063434)