

**„Schola Europa Akadémia” Technikum, Gimnázium és Alapfokú Művészeti Iskola a Magyarországi Metodista Egyház fenntartásában**

**SZOFTVERFEJLESZTŐ ÉS -TESZTELŐ**  
**5 0613 12 03**

**Vizsgaremek dokumentáció**  
***2025***

**Készítette:**

**Ambrus Erik Milán**

**Galbács Norbert**

**Gyuricza Ferenc Máté**

**Bevezető**

Az online játékok világában az adatelemzés és statisztikai nyilvántartás egyre fontosabb szerepet tölt be. A játékosok szeretnek fejlődésükhöz visszajelzést kapni, látni a teljesítményüket, összehasonlítani másokkal, és követni a játékon belüli előrehaladásukat. Ennek megfelelően ma már számtalan weboldal és platform kínál különféle statisztikai elemzéseket, ranglistákat, vagy épp egyéni játékosprofilokat különböző játékokhoz. Ezek az oldalak azonban általában kizárólag az adatokra fókuszálnak, ezért hiányzik belőlük az emberi kapcsolat lehetősége.

Ez a projekt egy olyan weboldal létrehozását tűzte ki célul, amely ötvözi a játékos statisztikák precíz és informatív megjelenítését egy közösségi, ismerkedésre is alkalmas felülettel. Lehetőséget nyújt a játékosok számára, hogy kapcsolatba lépjenek egymással, barátságokat kössenek, és akár hosszú távú közösségeket építsenek ki. A legtöbb hasonló oldal figyelmen kívül hagyja azt a tényt, hogy a játékok legfontosabb része nem feltétlenül az eredmény, hanem a közösségi élmény. E projekt célja egy olyan egyedi platform létrehozása, ahol a játékosok nemcsak elemezhetik saját teljesítményüket, hanem egy támogató, aktív közösség részévé is válhatnak.

**Program** **leírás**

Weboldalunk neve StatLink, egy játékos statisztikai adatgyűjtő és raktározó oldal közösségi és kommunikációs funkciókkal. Elsődlegesen böngészői felületre készült, reszponzív, telefonon is használható. HTML nyelvvel készült, CSS és Bootstrap 5 keretrendszer segítségével. A háttérfolyamatokhoz PHP-t használtunk, mint elsődleges programozónyelv, JavaScript segítségével. Az adatbázis kezeléséhez MySQL ls PhpMyAdmin-t használtunk.

**Felhasznált program nyelvek**

**HTML**

A HTML egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált a W3C (World Wide Web Consortium) támogatásával. HTML általában szöveges állományokban található meg szervereken. Ezek az állományok tartalmazzák azt az információt, ami a megfelelő programnak leírja, hogyan is kell megjeleníteni, illetve feldolgozni az adott állomány tartalmát. Megjelenítő program lehet egy webböngésző „,aural” böngésző, Braille olvasó, amely átalakítja a szöveget Braille "formátumba", levelező program, valamint egyéb eszközök, például mobiltelefon.

**CSS**

A Cascading Style Sheets (CSS) adja a leíró kódokon belüli elemeknek a megjelenési tulajdonságokat. A CSS létrejöttének az volt az oka, hogy a weboldalakon sokkal komplexebb és kifinomultabb elemekre és formázásokra volt szükség, amelyeket már az eredetileg erre kitalált HTML nem volt képes megfelelően vagy egyáltalán nem is volt képes kezelni. Emellett felgyorsítja, átláthatóvá és könnyen szerkeszthetővé teszi az oldalakon lévő dizájn elemeket. A CSS segítségével formázási beállításokat hozhatunk létre, amely lehet szöveg formázása vagy egész weboldalak kinézetének módosítása. Így a felhasználók számára is minimális kényelmet tudunk vele biztosítani ugyanis az úgynevezett stíluslapot elmenti a cache (gyorsító tár) mappába a böngészőnk, ezáltal mikor újra ellátogat, a weblapunkra azt nem kell letöltenie. A legelterjedtebb böngészők nagy része ma már támogatja a CSS nyelv elemeit, de még mindig vannak olyan böngészők, amelyek egyes nyelvi elemeket rosszul jelenítenek, meg vagy egyáltalán meg sem jelenítenek. Minden stílus és dizájn css fájlokban készült, amik jelentősen meggyorsították a munkánkat.

**JavaScript**

A JavaScript egy programozási nyelv, amelyet kifejezetten az internetre fejlesztettek ki. A legtöbb webböngésző szoftver, és a modern okostelefonok is mind támogatják ezt a nyelvet. A JavaScriptet első sorban arra használják, hogy gazdagabb, felhasználóbarát élményeket teremtsenek vele az internetet böngészők számára, például dinamikusan frissülő weboldalakat, intuitív felhasználói felületeket, menüket, párbeszédpaneleket, 2D-s és 3D-s grafikákat, interaktív térképeket, videólejátszókat, és számos egyéb elemet, illetve funkciókat. A JavaScript ilyesfajta, webböngészőkben történő alkalmazását kliensoldalú JavaScriptnek szokás nevezni. Az oldalunk összes animációja JavaScript segítségével történt, illetve a bejelentkezéshez ellenőrzés, felugró ablakok is.

**PHP**

A PHP egy nyílt forráskódú, szerveroldali szkriptnyelv, amelyet elsősorban webfejlesztésre terveztek. A PHP-t a webszerver dolgozza fel, és az eredményként kapott HTML-t küldi el a böngészőnek. Segítségével dinamikus weboldalak és webalkalmazások hozhatók létre, például űrlapkezelés, adatbázisokkal való kapcsolattartás, felhasználói fiókok kezelése, vagy éppen tartalomkezelő rendszerek működtetése. A PHP-t gyakran használják együtt adatbázisokkal, hogy komplex, interaktív webes rendszereket hozzanak létre. Könnyen tanulható, széleskörű közösségi támogatással rendelkezik, és a legtöbb tárhelyszolgáltató alapból támogatja.

**Felhasznált szoftverek**

**Notepad++**

A Notepad++ egy ingyenes, könnyű és gyors szövegszerkesztő Windows operációs rendszerre, amelyet főként programozáshoz és kódszerkesztéshez használnak. Támogat számos programozási nyelvet – például PHP, HTML, JavaScript, C++, Python stb. – és olyan kényelmi funkciókat kínál, mint a szintaxiskiemelés, automatikus kiegészítés, többszörös dokumentummegjelenítés, makrók, illetve bővítmények telepítése. Kiváló eszköz kezdőknek és haladó felhasználóknak egyaránt, akik gyorsan és hatékonyan szeretnének kóddal dolgozni egyszerű környezetben.

**Visual Studio Code**

A Visual Studio Code, röviden VSC vagy VS Code, egy ingyenes, platformfüggetlen forráskód-szerkesztő a Microsofttól. Kifejezetten fejlesztőknek készült, és széles körű támogatást nyújt szinte minden népszerű programozási nyelvhez és fejlesztési eszközhöz. Funkciói közé tartozik a szintaxiskiemelés, hibakeresés, Git-integráció, intelligens kódkiegészítés, valamint egy kiterjedt bővítményrendszer, amellyel szinte bármilyen fejlesztési környezet testreszabható. A Visual Studio Code az egyik legnépszerűbb kódszerkesztő a modern web- és szoftverfejlesztésben.

**Discord**

A Discord egy ingyenes, többplatformos kommunikációs alkalmazás, amelyet főként közösségek, baráti társaságok és csapatok közötti csevegésre használnak. Lehetővé teszi a szöveges, hang- és videóalapú kommunikációt, valamint támogatja a szerverek és csatornák létrehozását, ahol témák szerint lehet beszélgetéseket folytatni. Különösen népszerű a játékosok, fejlesztők és online közösségek körében, mivel egyszerűen testreszabható, botokkal és integrációkkal bővíthető. Kiváló választás azok számára, akik egy gyors, stabil és sokoldalú közösségi platformot keresnek.

**GitHub**

A GitHub egy webalapú verziókezelő platform, amelyet főként szoftverfejlesztők használnak kód tárolására, kezelésére és megosztására. A Git verziókezelő rendszerre épül, és lehetővé teszi a projektek együttműködésen alapuló fejlesztését, hibakövetést, dokumentációkezelést és kiadások kezelését. Támogatja a nyílt forráskódú és privát projektek kezelését, emellett számos fejlesztési eszközzel – például GitHub Actions, Codespaces vagy Pull Request rendszer – segíti a hatékony munkát. Ideális környezet egyéni fejlesztőknek, csapatoknak és nyílt forráskódú közösségeknek egyaránt.

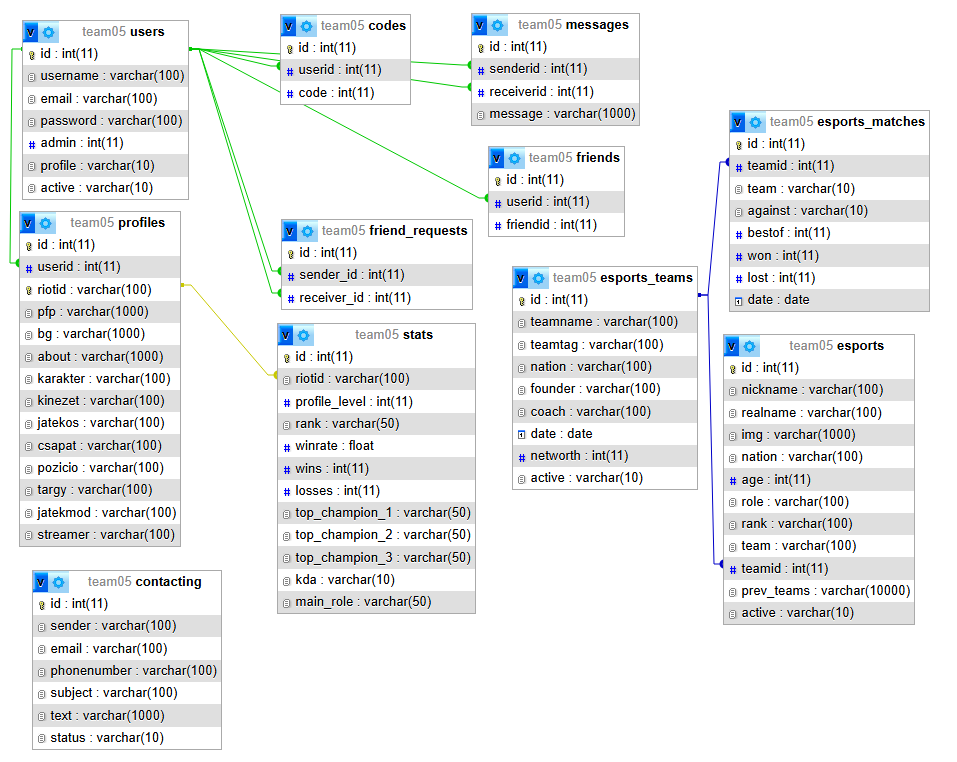
**Xampp**

A XAMPP egy ingyenes, könnyen telepíthető fejlesztőkörnyezet, amely egy komplett szerverkörnyezetet biztosít a számítógépen, anélkül hogy külön szervert kellene beállítani. A név a következő komponensekre utal: **X** (bármilyen operációs rendszer), **A**pache (webszerver), **M**ySQL (adatbázis), **P**HP és **P**erl (programozási nyelvek). A XAMPP segítségével helyben lehet futtatni PHP-alapú weboldalakat és adatbázisokat, így ideális tanulásra, fejlesztésre vagy tesztelésre anélkül, hogy valódi online tárhelyre lenne szükség. Tartalmazza a phpMyAdmin eszközt is, amely egy grafikus felület az adatbázisok kezeléséhez.

Először a Notepad++ volt az elsődleges kódíró szoftver, amit használtunk, majd a későbbiekben átváltottunk a Visual Studio code-ra, ugyanis sokkal megkönnyítette a munkát, köszönhetően a beépített funkcióinak, amik közé a gyors kitöltés, adott elemek szinezése és a saját magunk által létrehozott parancsok segítettek. A Xampp PhpMyAdmin segítségével hoztuk létre és teszteltük az adatbázist, amit később az iskola által biztosított szerverre feltöltötünk online. A csoport beszélgetésekre Discordot használtunk, valamint Github-on és Google Drive-on tároltuk a projektet.

**Adatbázis**

Az adatbázis 2 fő részből áll, a felhasználók és esport. A felhasználók legfontosabb táblája a users, amiben a regisztráció során megadott adatokat tároljuk, mint például a felhasználónév, email és a titkosított jelszó. Itt található a profil és admin státusz is, valamit a profil aktív állapota, amit a codes nevű táblával ellenőrzünk, mivel regisztráció után emailben kap a felhasználó egy kódot, amivel beaktiválhatja a fiókját. Ezután lehetőség van a profil létrehozásához, amivel a játékban szereplő nevet tároljuk el és kötjük össze a játék saját API-val. Ennek köszönhetően le tudjuk kérni az adott létező játékosnak a profil szintjét, rangját, győzelmi rátáját, legtöbbet játszott karaktereit, és még sok mást. Ezen kívül saját adatokkal lehet testre szabni a profilt, mint a leírás, profilkép vagy kedvenc karakter.



A játékból kinyert, kiemelten fontos statisztikai adatokat, mint például a játékos rangja vagy győzelmeinek száma a stats nevű adatbázistáblában tároljuk. Ez a tábla kapcsolatban áll a profiles táblával, mégpedig a riotid oszlopon keresztül, amely másodlagos kulcsként funkcionál. A riotid oszlop unique típusú értéket kapott, mivel ez az azonosító egyedisége megegyezik a primary key tulajdonságával, így minden felhasználóhoz egyértelműen hozzárendelhető.

A felhasználók közötti barátkezelési funkciók megvalósításához több adatbázistábla is kialakításra került. Az első ilyen a friends nevű tábla, amelyben a users táblából származó userid értékek kerülnek eltárolásra. Fontos megjegyezni, hogy ez a tábla kizárólag abban az esetben frissül, ha a barátfelkérés a friend\_requests nevű táblában elfogadásra kerül. A friend\_requests tábla tehát a barátküldési folyamat átmeneti tárolását szolgálja, míg a friends tábla az érvényes, kölcsönösen elfogadott baráti kapcsolatokat rögzíti.

Az üzenetküldési funkciót egy külön messages nevű tábla kezeli, amelyben a felhasználók között váltott szöveges üzenetek kerülnek tárolásra. Ebben a táblában minden üzenethez hozzárendeljük a küldő és a fogadó felhasználó azonosítóját, az üzenet tartalmát, valamint az elküldés időbélyegét, így biztosítva a kommunikáció visszakövethetőségét és strukturált tárolását.

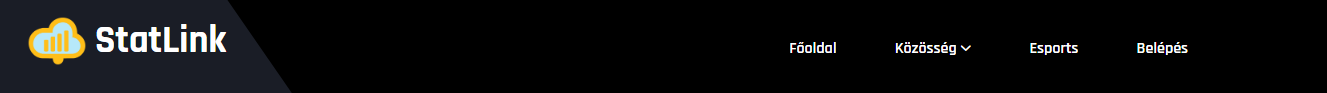
A másik csoport az esports, ahol a profi játékosok, csapataik, illetve meccseiket tároljuk. Ezek a táblák külön léteznek a profil mellett, és a profi játékosoknak saját adataik vannak, mint teljes név, életkor, aktív csapat, csapattársak.

**Felhasználói dokumentáció**

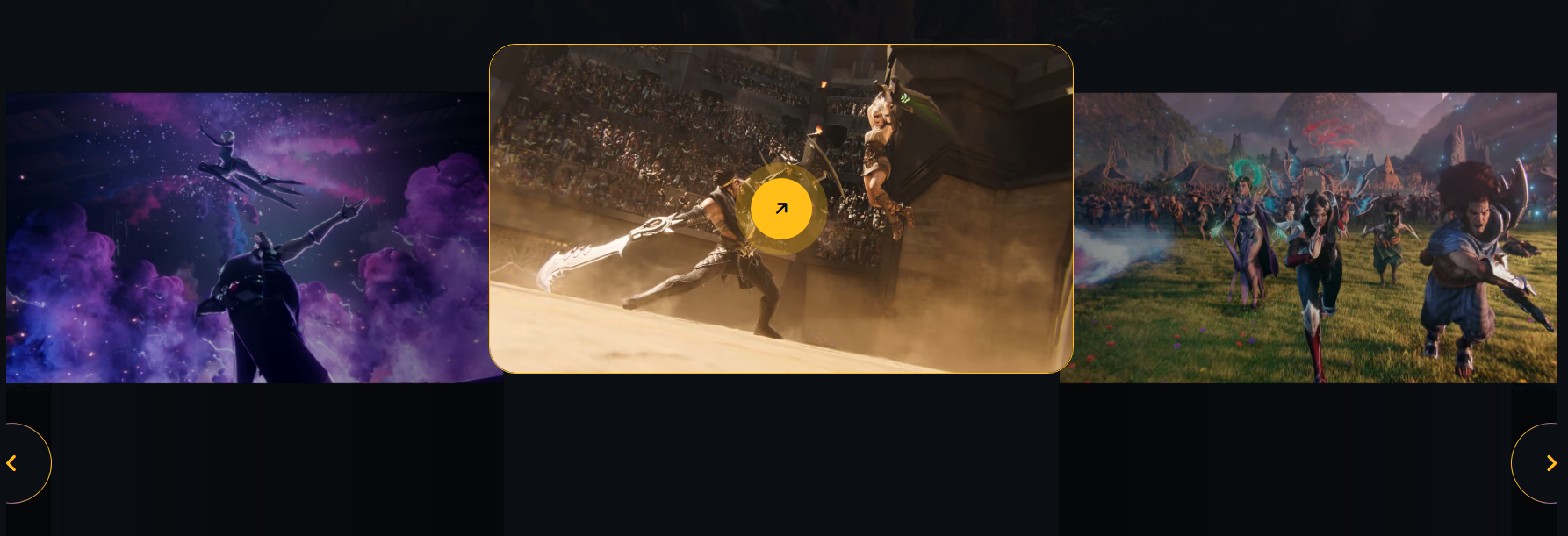
A weboldalunk betöltésekor a látogató automatikusan a főoldalra irányítódik, ahol a kiválasztott játék, jelen esetben a League of Legends, kerül bemutatásra. Elsődleges funkcióként a játékostárs keresése, valamint a profi játékosok megtekintése válik elérhetővé.



Az oldal bejelentkezés nélkül is látogatható, azonban ilyenkor a funkciók korlátozottan érhetők el.



A főmenüben az alábbi pontok szerepelnek: főoldal, hírek, esport játékosok, valamint a belépés lehetősége. A főoldalon lefelé haladva egy képgaléria található, amely a játék hangulatát mutatja be. Ezalatt az aktív profi játékosokról egy “slideshow” található.



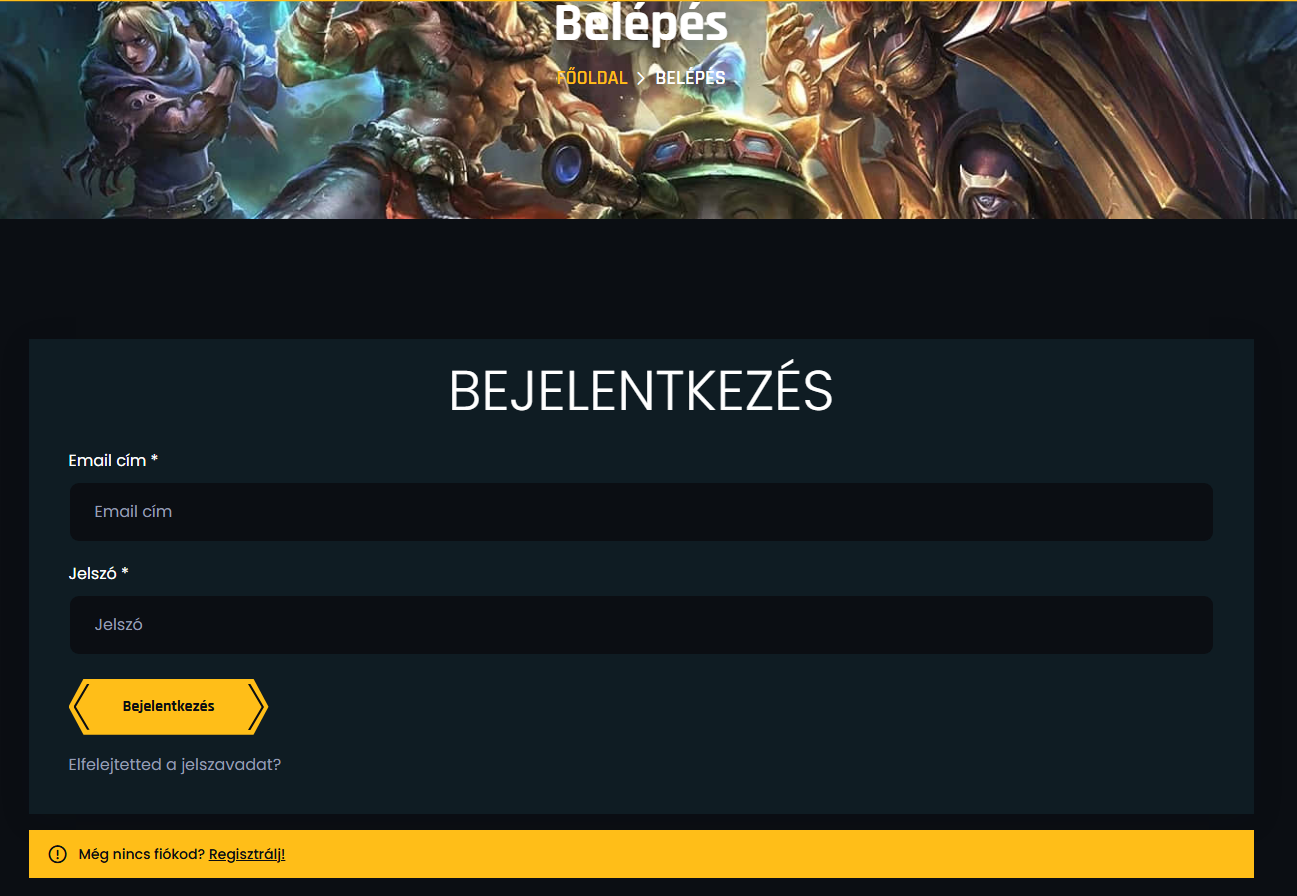
Az oldal alsó szakaszán elhelyeztünk egy CTA-szekciót (call to action), amely lehetőséget biztosít a felhasználók számára a csatlakozásra.



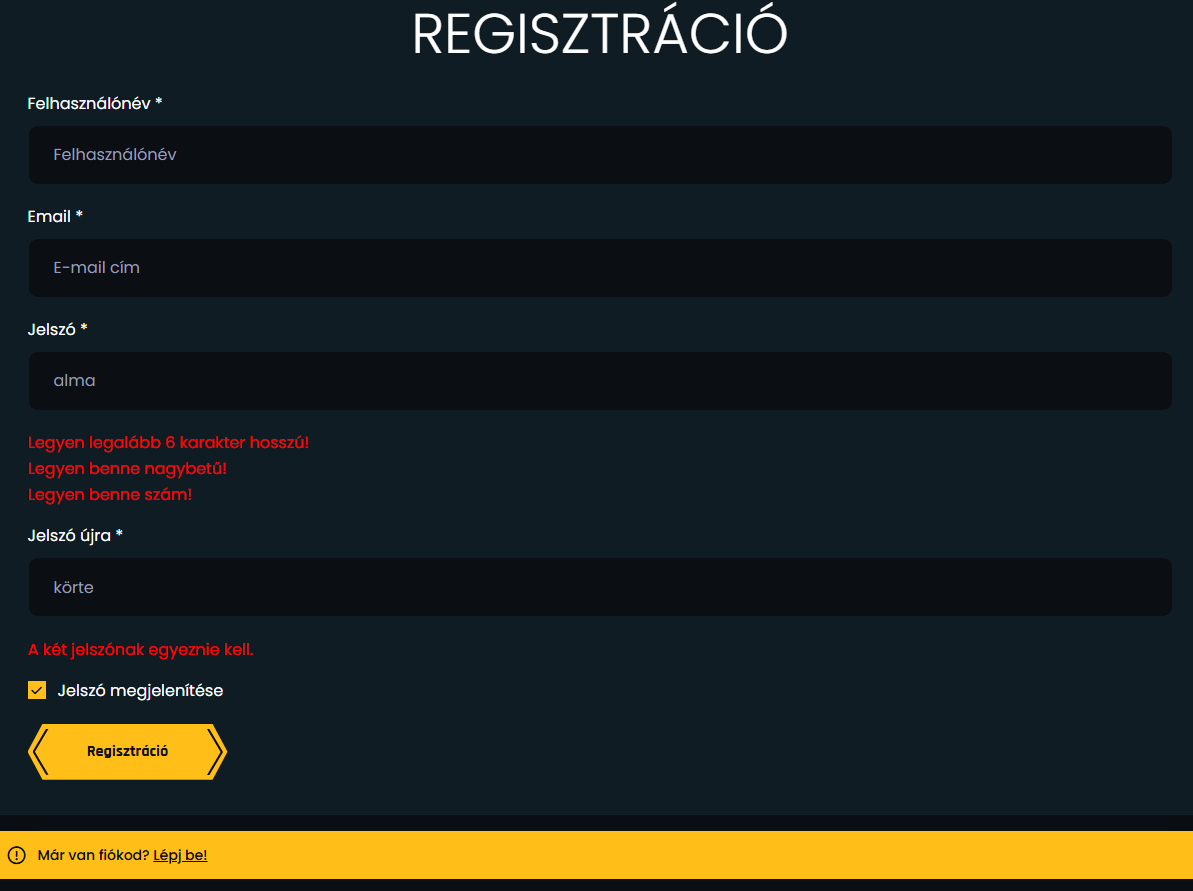
A láblécben kapcsolatfelvételi lehetőséget biztosítunk, valamint hibabejelentő funkció is rendelkezésre áll az esetleges technikai problémák jelzésére. Emellett egy kisebb almenü is található kevesebb opcióval, valamint lehetőség nyílik hírlevélre való feliratkozásra, amelyen keresztül a látogatók értesülhetnek a legfrissebb információkról és frissítésekről.



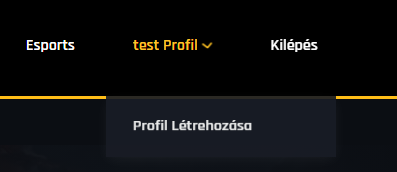
A Belépés menüpontra történő kattintás után a felhasználót automatikusan átirányítjuk a regisztrációs oldalunkra, ahol alapértelmezetten a bejelentkezési felület jelenik meg. Amennyiben a látogatónak még nincs felhasználói fiókja, úgy a felületen elhelyezett „Még nincs fiókod?” szöveges sávban található „Regisztrálj!” hivatkozásra kattintva megjelenik a regisztrációs űrlap.



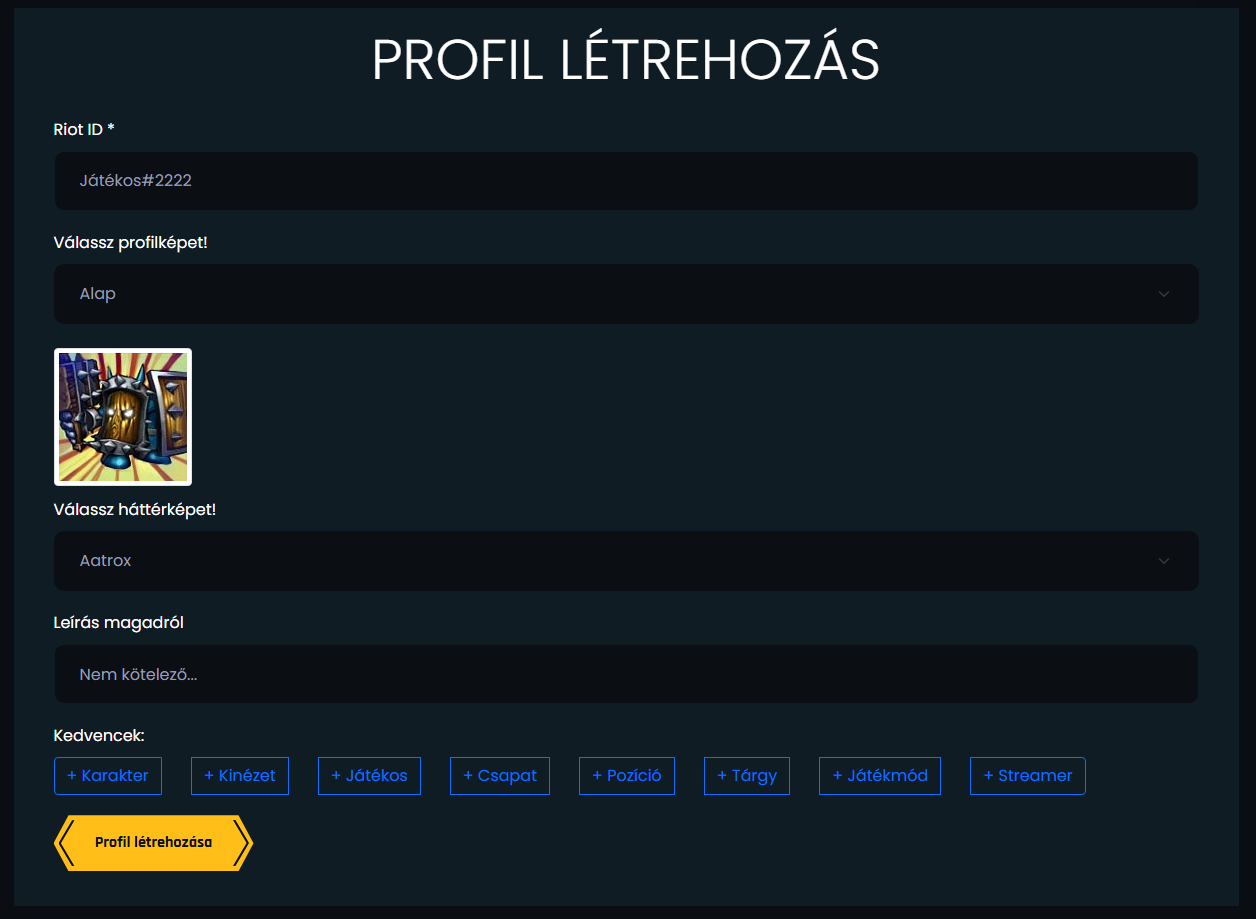
A regisztrációs felületen a felhasználónak lehetősége nyílik saját felhasználónevének, e-mail címének és jelszavának megadására. A jelszó megadásakor biztonsági szempontból kötelezően érvényesíteni kívánunk néhány alapvető feltételt: a megadott jelszónak legalább nyolc karakter hosszúnak kell lennie, továbbá tartalmaznia kell minimum egy számot és egy nagybetűt is. A rendszer ezen feltételek teljesülését valós időben ellenőrzi, és hiányosság esetén figyelmeztető üzenetet jelenít meg. Továbbá biztosítunk egy lehetőséget a jelszó megjelenítésére, amely segítséget nyújt annak pontos beírásában. A második jelszómezőnél a rendszer automatikusan összeveti az értékeket, és visszajelzést ad arról, hogy a két megadás egyezik-e.



A „Regisztráció” gombra történő kattintás után a megadott e-mail címre automatikusan kiküldésre kerül egy megerősítő levél, amely egy aktivációs linket tartalmaz. A fiók sikeres aktiválása után a felhasználót visszairányítjuk a bejelentkezési felületre, ahol érvényes hitelesítési adatok megadását követően a rendszer belépteti, majd automatikusan visszavezeti a főoldalra.



Ezt követően megjelenik a navigációs menüsávban a Profil menüpont, amely újonnan regisztrált felhasználó esetén a Profil létrehozása lehetőséget kínálja. Ennek a gombnak a megnyomásával megnyílik a profilkészítő oldal, ahol elsődlegesen a Riot ID megadására van szükség. Felhívjuk a figyelmet arra, hogy ennek az adatnak a pontos és helyes megadása kiemelten fontos, mivel hibás formátum vagy elírás esetén a rendszer nem tudja beazonosítani a játékos profilját a játékban.



A Riot ID megadásán túl a felhasználónak lehetősége nyílik arra is, hogy egy előre definiált listából kiválassza a számára szimpatikus profilképet és háttérképet, továbbá megadhat egy rövid, egyéni leírást, amely a profil nyilvános részén jelenik meg. A felület további, jövőben bővülő lehetőségeket is kínál, többek között a kedvenc karakter, kedvenc profi játékos, valamint kedvenc csapat megadására, amelyek később személyre szabottabb élményt biztosítanak a felhasználói fiókhoz kapcsolódóan.

A “Profil létrehozása” gomb megnyomásával visszairányítja a felhasználót a profil oldalra, ahol már a frissült adatokkal látható egyedi profilját láthatja a játékból lekérdezett saját adatokkal, mint például a játékos szint, elért rang, győzelmi arány.

A profil oldalon lehetőség van játékos adatainak szerkesztésére, ha nem a saját profilját nézi és rendelkezik profillal, akkor a barát felvételére. Ha már rendelkezik barátokkal, akkor a a Barátok kezelésével átirányít a Barátok oldalra, ami a Közösség menüpont alatt is megtalálható, de eddig nem volt elérhető, ugyanis nem volt bejelentkezve a felhasználó.

A közösségi felület alapértelmezés szerint a hírek szekcióval nyílik meg, ahol folyamatosan tájékoztatjuk a felhasználókat a játékkal kapcsolatos legfontosabb újdonságokról. Ezt a tartalmat két fő kategóriára bontottuk: az első az általános bejelentések gyűjteménye, amelyben például egy újonnan megjelent videóra, eseményre vagy egy frissen elérhető fizikai termékre hívjuk fel a figyelmet, amely akár egy online áruházból is megrendelhető. A második kategóriában a játék frissítéseit közöljük, lehetőséget biztosítva arra, hogy nyomon követhessék a játékmenet egyensúlyával, technikai fejlesztéseivel és hibajavításaival kapcsolatos változásokat.

A „Barátok” menüpont alatt a felhasználók részletes áttekintést kapnak a kapcsolataikról. Itt minden ismerős megjelenik egy jól strukturált listában, amelyben kezelhetik a beérkezett barátfelkéréseket, visszaigazolhatják azokat, valamint lehetőségük nyílik kapcsolatok törlésére is. Emellett közvetlenül innen indítható privát beszélgetés az ismerősökkel, elősegítve ezzel az azonnali kommunikációt.

**Fejlesztői dokumentáció**

A fejlesztési folyamat során az agilis módszertant választottuk, mivel ez lehetővé tette számunkra a folyamatos iterációt, a gyors alkalmazkodást és az azonnali visszajelzésekre épülő módosításokat. A projekt hatékony verziókövetéséhez a Git verziókezelőt és a GitHub platformját használtuk, míg a belső kommunikáció és a csapatmunka összehangolása a Discord segítségével valósult meg.

A felhasználói felület fejlesztéséhez a Bootstrap 5 keretrendszert alkalmaztuk, amely jelentősen felgyorsította a reszponzív és esztétikus design kialakítását. A frontend logikához natív JavaScriptet használtunk, így elkerültük a fölösleges külső függőségeket, ami egyszerűbb és könnyebben karbantartható kódot eredményezett. A backend kialakításához PHP-t és JavaScriptet alkalmaztunk, amelyek együttesen rugalmas és megbízható környezetet biztosítottak, különösen a valós idejű adatkezelési igények esetén.

Az adatkezeléshez elsősorban MySQL-t és a hozzá kapcsolódó PhpMyAdmin felületet vettük igénybe, amely jól skálázható megoldásként szolgált a strukturált adatok kezelésére.

A választott technológiák korszerűek, széles körben támogatottak, és hosszú távon fenntarthatóak. Az agilis megközelítés segített abban, hogy gyorsan tudjunk reagálni a felmerülő igényekre és hatékonyan javítsuk az esetleges hibákat. A Bootstrap segítségével gyors és látványos felhasználói élményt tudtunk nyújtani, míg a backend technológiák megfelelő alapot biztosítottak a rendszer skálázhatóságához. A verziókezelő és kommunikációs eszközök együttesen biztosították a fejlesztési folyamat átláthatóságát és a csapatmunka zökkenőmentes lebonyolítását.

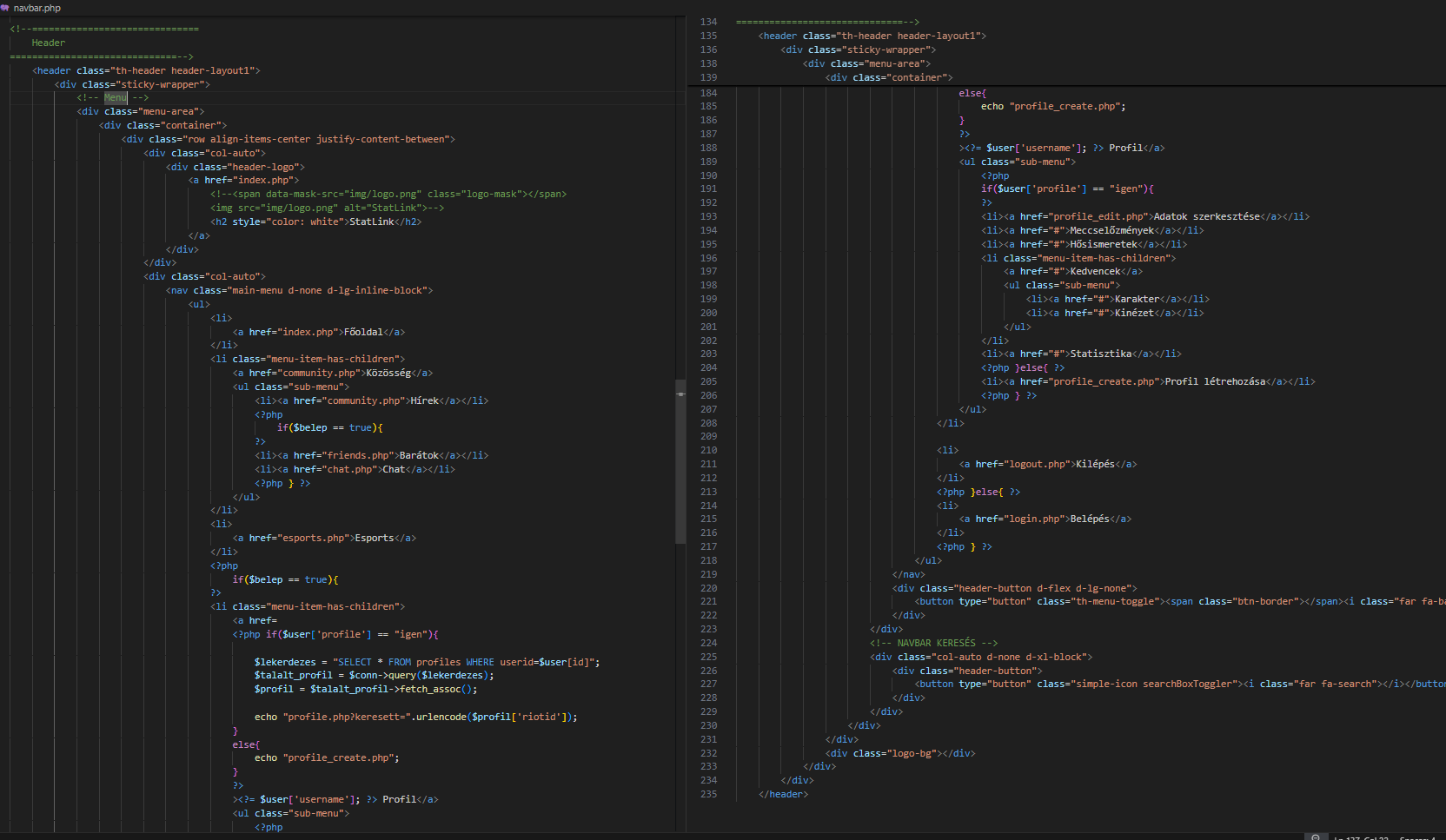
**Az oldal fontosabb funkciói:**

Weboldalunk minden oldalán egységes módon jelenik meg a navigációs sáv (navbar) és az oldal alján elhelyezett lábjegyzet (footer), amelyek külön PHP fájlokban kaptak helyet. Ezzel a megközelítéssel jelentősen csökkentettük az ismétlődő kódrészek jelenlétét az egyes oldalak forráskódjában, így a fejlesztés és karbantartás is jóval átláthatóbbá vált.

A **navbar.php** fájl nem csupán a navigációs elemek megjelenítéséért felelős, hanem már betöltéskor ellenőrzi a felhasználó bejelentkezési állapotát is. Amennyiben a látogató nem rendelkezik aktív munkamenettel, azaz nem jelentkezett be, a navigációs menü korlátozott funkcionalitást biztosít számára. Ebben az esetben kizárólag az alapértelmezett, mindenki számára elérhető oldalak jelennek meg. Ellenkező esetben a rendszer automatikusan bővebb menüpontokat tesz elérhetővé, amelyek csak bejelentkezett felhasználók számára hozzáférhetők.

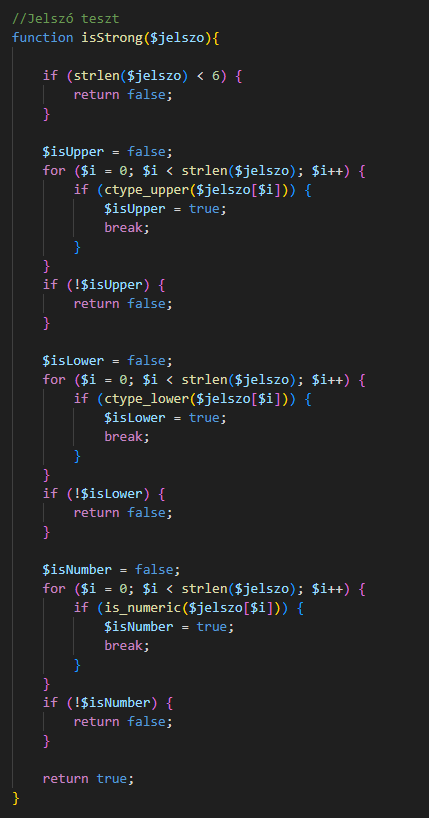


A fájl ezen kívül felelős a különböző eszközökön való megjelenés optimalizálásáért is. Mobilnézet esetén a menüsáv átalakul egy oldalról előhívható, rejtett menürendszerré, amelyet egy ikonra vagy gombra kattintva tud előhozni a felhasználó. Ez a funkció nemcsak a felhasználói élményt növeli kisebb kijelzőkön, hanem biztosítja az oldal reszponzív, modern megjelenését is.

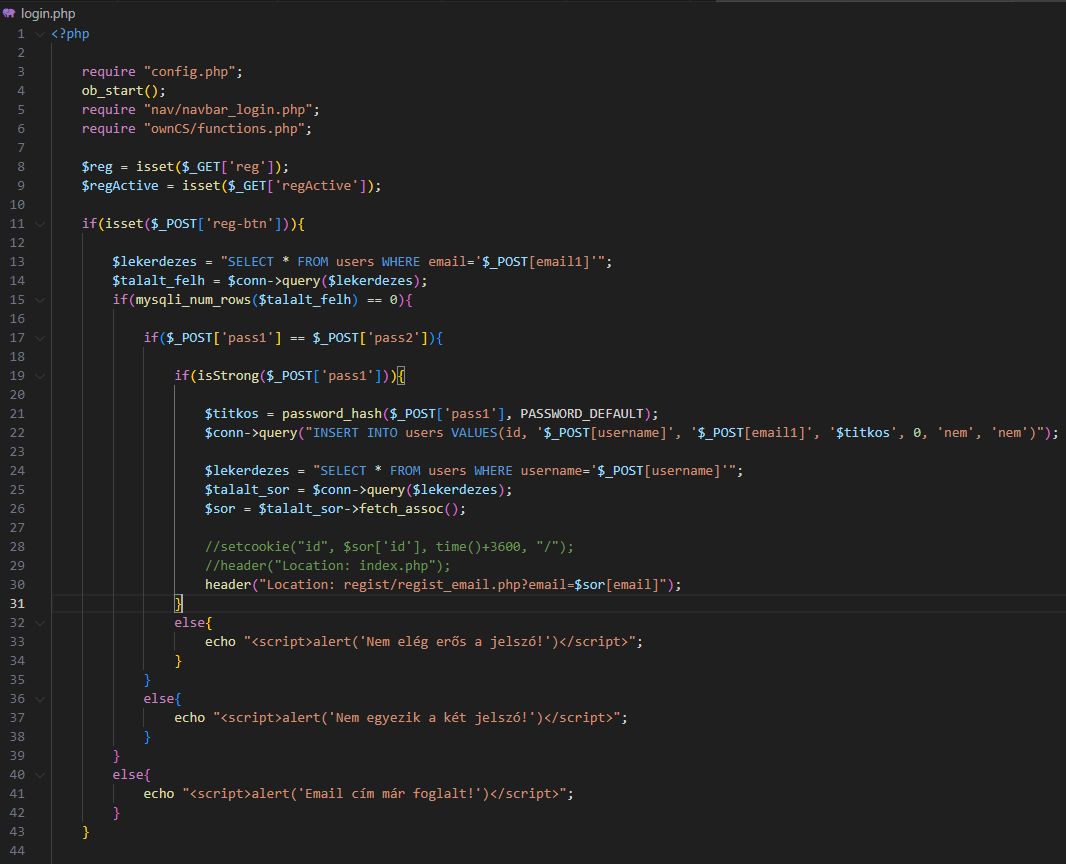


A második kiemelt PHP-fájlunk, a login.php, felhasználói regisztráció és bejelentkezés kezelésére szolgál. Ezen a felületen kétszintű ellenőrzést alkalmazunk annak érdekében, hogy a felhasználói élmény és az adatbiztonság egyaránt magas színvonalú legyen. Az első ellenőrzést szerveroldalon, PHP segítségével végezzük el, míg a második szintű validálást kliensoldalon, JavaScript-tel hajtjuk végre. Ez utóbbi lehetővé teszi, hogy a felhasználó azonnali visszajelzést kapjon például arról, hogy a megadott jelszó és annak megerősítése egyezik-e, illetve megfelel-e az előírt biztonsági követelményeknek.

A regisztráció során kötelező megadni az e-mail címet, amely alapján előszűrjük az adatbázist, kizárva a már használt, duplikált e-mail címeket. A jelszó amennyiben megfelel a minimálisan elvárt biztonsági feltételeknek miszerint legalább 8 karakter hosszú legyen és tartalmazzon legalább egy számot és nagybetűt, biztonságosan és titkosítva tároljuk a users nevű adatbázistáblában. Az erősségi feltételek ellenőrzését egy külön függvény, az isStrong() látja el, amely valós időben értékeli a jelszó komplexitását.



Amennyiben minden feltétel teljesül, azaz a megadott e-mail cím még nem szerepel az adatbázisban, a két jelszó megegyezik, és az ellenőrzés szerint megfelelően erős, a rendszer rögzíti a felhasználó adatait az adatbázisban. Az újonnan létrejött fiók automatikusan 0 értéket kap az admin oszlopban, jelezve, hogy nem rendelkezik rendszergazdai jogosultsággal. Emellett az alapértelmezett beállítás szerint még nem rendelkezik aktív fiókkal és profiloldallal sem. A regisztrációs folyamat azonban itt még nem ér véget, a rendszer automatikusan visszaigazoló e-mailt küld a megadott címre, amely egy aktiváló hivatkozást tartalmaz. Ezen link segítségével a felhasználó visszakerül a login.php oldalra, ahol egy üzenet tájékoztatja arról, hogy az e-mail cím megerősítése sikeresen megtörtént. Ezt követően a felhasználó már be tud jelentkezni az előzőleg megadott hitelesítő adatokkal.



A login.php oldal működéséhez két további fájl kapcsolódik, ezek közé tartozik a pass\_forgot.php, amely az elfelejtett jelszó kezelését végzi, valamint a pass\_change.php, amely lehetőséget biztosít a jelszó módosítására. Ez a két oldal a felhasználói fiókkezelés kiegészítő, de biztonság szempontjából elengedhetetlen részeit látja el.

Amikor a felhasználó elfelejtette a jelszavát, a rendszer első lépésként bekéri az e-mail címét, majd ellenőrzi, hogy az adott címhez tartozik-e regisztrált fiók az adatbázisban. Ha igen, akkor a rendszer egy visszaigazoló levelet küld a megadott címre. Ebben a levélben tájékoztatjuk a felhasználót arról, hogy jelszó-visszaállítási kérelmet nyújtott be, és ha valóban ő kérte ezt, akkor egy biztonságos űrlapon keresztül új jelszót adhat meg. Az űrlapon a jelszót kétszer kell megadnia, és mindkét mezőnek egyeznie kell. Ezen kívül a jelszónak meg kell felelnie az előírt biztonsági szabályoknak, vagyis legalább nyolc karakter hosszúnak kell lennie, valamint tartalmaznia kell számot és nagybetűt is. Ha minden feltétel teljesül, akkor az új, titkosított jelszót az adatbázisban a korábbi helyére mentjük, majd a felhasználót visszairányítjuk a login.php oldalra, ahol újra megpróbálhat bejelentkezni.



A jelszó módosítása funkció azok számára érhető el, akik már bejelentkeztek. Itt a rendszer egy űrlapot biztosít, amelyben először a jelenlegi jelszót kell megadni, majd kétszer az új jelszót. Fontos szempont, hogy az új jelszó nem lehet azonos a korábbi jelszóval, azonban az ismételten megadott új jelszónak természetesen egyeznie kell az első mezőben megadottal. A rendszer ezek után ellenőrzi, hogy az új jelszó megfelel-e a biztonsági követelményeknek, majd ha minden megfelelő, akkor a rendszer frissíti az adatbázisban tárolt jelszót, és a felhasználót tájékoztatja a változtatás sikerességéről.

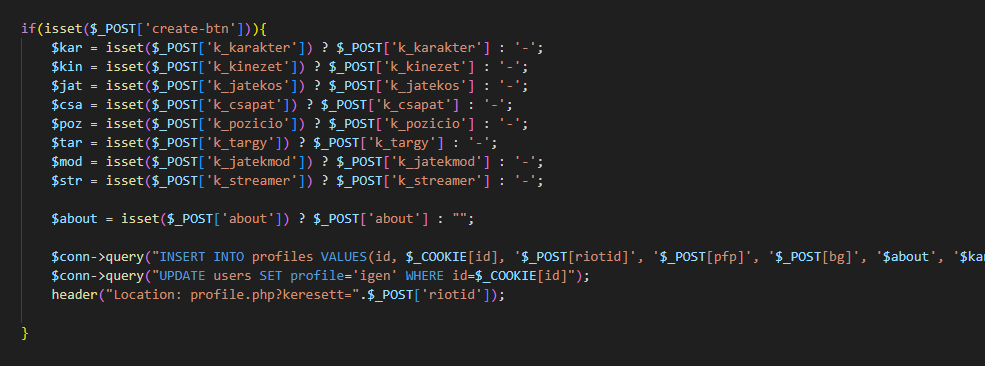


A profil.php oldal működése több feltételhez kötött, amelyek biztosítják az oldal helyes használatát és a felhasználói adatok megfelelő kezelését. Az oldal nem érhető el önmagában közvetlen linkeléssel, tehát ha valaki csupán a profil.php címet írja be a böngészőbe, akkor az oldal nem jelenít meg semmilyen tartalmat. Az oldal működéséhez szükséges egy, az URL-ben megadott paraméter, amelyet "keresett" név alatt kezel a rendszer. Ez a változó határozza meg, hogy melyik profil adatait kell megjeleníteni.

Amennyiben a felhasználó be van jelentkezve, a rendszer automatikusan újratölti az oldalt, és a keresett változót a bejelentkezett felhasználó adataival tölti ki. Ez alapján az oldal az adatbázisból betölti a felhasználóhoz tartozó profilt, amennyiben az már létezik. Ha viszont a felhasználó még nem rendelkezik saját profillal, a rendszer átirányítja őt a profile\_create.php oldalra, ahol lehetősége van új profilt létrehozni.

A profil létrehozásakor kötelező megadni a játékhoz kapcsolódó Riot ID értéket, amely egyéni azonosítóként szolgál. Ez az azonosító egy tetszőleges névből áll, amelyet egy # jel követ, majd egy négyszámjegyű vagy betűkből és számokból álló egyedi kód zár. Korábban ez a kód régiókra utalt, de ma már személyre szabható karakterkombinációkat tartalmaz, például így nézhet ki: TestJatekos#1234. A Riot ID helyes megadása kiemelten fontos, mert e nélkül nem tudjuk az adott játékos profilját elérni és a hozzá tartozó adatokat lekérni az API-n keresztül.

A profil létrehozása során a felhasználónak lehetősége van profilképet és háttérképet is beállítani. Ezeket a rendszer egy előre feltöltött képkönyvtárból kínálja fel, és a kiválasztott fájlokat ebből a mappából rendeli hozzá a felhasználó profiljához. Továbbá elérhető egy úgynevezett kedvencek opció is, ahol a felhasználó egyéni információkat adhat meg, mint például a kedvenc karaktere, kinézete vagy más, játékhoz kapcsolódó tartalom, amelyet személyre szabottan jelenítünk meg a profiloldalon.



Miután a felhasználó a profil létrehozására szolgáló űrlapot kitöltötte és a mentés gombra kattintott, a rendszer első lépésként feltölti az adatokat az adatbázis profil nevű táblájába. Az egyik fontos oszlop, a userid, amelybe automatikusan bekerül a jelenleg bejelentkezett felhasználó azonosítója. Ezt az értéket a rendszer a felhasználóhoz tartozó, sütiben tárolt azonosító alapján határozza meg. Így biztosítjuk, hogy az újonnan létrehozott profil egyértelműen kapcsolódjon a megfelelő fiókhoz.

Ezt követően a megadott Riot ID alapján az oldal kapcsolatba lép a Riot Games nyilvános API-jával. Az API-kulcs segítségével lekérjük az adott játékoshoz tartozó adatokat, majd ezeket a friss információkat elmentjük a saját adatbázisunk megfelelő mezőibe. Ez biztosítja, hogy a profiloldalon nemcsak a felhasználó által megadott adatok, hanem a játékból származó információk is naprakészen megjelenhessenek.



A sikeres mentést követően a rendszer visszairányítja a felhasználót a profil.php oldalra, ahol ezúttal már automatikusan felismeri, hogy a bejelentkezett felhasználónak létezik profilja. Ennek köszönhetően a rendszer képes értelmezni a keresett nevű változót, és ennek alapján megjeleníti a teljes profilt. Az oldalon ezután láthatóvá válik minden lényeges információ, beleértve a saját, manuálisan megadott adatokat, valamint azokat az értékeket is, amelyeket a Riot API segítségével automatikusan töltöttünk be.



További funkcióként az oldal lekérdezi a friends nevű táblát is, amelyben a rendszer ellenőrzi, hogy vannak-e barátai az adott felhasználónak. Ha a keresett profil a bejelentkezett felhasználóé, és az adatbázis talál barátokat, akkor ezeket kilistázzuk az oldalon. Abban az esetben, ha még nincsenek barátok rögzítve, egy üzenet jelenik meg, amely erről tájékoztatja a felhasználót.

Az oldalon megjelenik egy adatok szerkesztése gomb is, amely lehetőséget biztosít arra, hogy a felhasználó módosítsa korábban megadott profiladatait. A jelszó ebben a szakaszban nem módosítható, de minden más mező frissíthető. A profil mentése gomb lenyomásával a rendszer összeveti a módosított adatokat a korábbi értékekkel, és csak a változtatott mezőket frissíti az adatbázisban. Amennyiben a felhasználó üresen hagy egy mezőt, az adott helyen megtartjuk az előzőleg rögzített adatot, így nem vesznek el a korábbi információk.

**Tesztelési dokumentáció**

A fejlesztési folyamat során kiemelt figyelmet fordítottunk a kód minőségének folyamatos ellenőrzésére és a hibák kiszűrésére. A frontend oldalon HTML, CSS és JavaScript validátorokat alkalmaztunk annak érdekében, hogy biztosítsuk a forráskód szabványosságát, hibamentességét és kompatibilitását a különböző böngészőkkel. Ezek az eszközök segítettek a szintaktikai hibák, elavult elemek és nem ajánlott gyakorlatok kiszűrésében.

A backend oldali fejlesztéseknél PHP és JavaScript nyelven írt unit teszteket futtattunk, amelyek célja az volt, hogy automatikusan ellenőrizzük az egyes modulok és függvények működését. Ezen tesztek biztosították, hogy az egyes funkciók a vártnak megfelelően működnek, és a későbbi módosítások ne vezessenek regressziós hibákhoz.

Az adatbázis-integráció tesztelését manuálisan végeztük, ahol különféle lekérdezések és adatmódosítások végrehajtásával ellenőriztük az adatkapcsolatok helyes működését. Emellett REST API-hívások segítségével validáltuk az adatok helyes továbbítását és feldolgozását a frontend és backend között.

A weboldalt többféle böngészőben teszteltük annak érdekében, hogy garantáljuk a platformfüggetlen megjelenést és működést. A különböző operációs rendszerek használata szintén hozzájárult a kompatibilitási problémák kiszűréséhez.

A reszponzivitás vizsgálatához különféle képernyőméretű és felbontású eszközökön, például okostelefonokon, táblagépeken és asztali számítógépeken teszteltük az oldalt, hogy biztosítsuk a tartalom megfelelő megjelenését minden környezetben.

A felhasználói élmény optimalizálása érdekében különböző életkorú és technológiai háttérrel rendelkező felhasználókat vontunk be a tesztelésbe. Az általuk adott visszajelzések alapján célzott módosításokat hajtottunk végre, amelyek elősegítették a könnyebb navigációt, az átláthatóbb felületet és a jobb hozzáférhetőséget.

A fejlesztési folyamat során elsősorban manuális tesztelést alkalmaztunk, mivel ez bizonyult számunkra a legkézenfekvőbb megoldásnak. A projekt jellegéből adódóan „élőben” dolgoztunk. A fejlesztés közben folyamatosan láttuk az oldal vizuális megjelenését és a kapcsolódó kódrészleteket is, így szinte azonnal tudtuk tesztelni az elvégzett módosításokat. Ez a közvetlen visszacsatolás nagyban segítette a gyors hibakeresést és finomhangolást, különösen szoros határidők mellett, amikor nem volt idő komplex automatizált tesztrendszerek kiépítésére.

A manuális tesztelés során főként a funkciók működését, az oldalelemek elrendezését, a reszponzivitást és a böngésző-kompatibilitást vizsgáltuk. Kézzel próbáltuk ki a különböző felhasználói interakciókat, és ellenőriztük, hogy minden úgy működik-e, ahogy terveztük.

Emellett korlátozott mértékben alkalmaztunk unit tesztelést is, főként a backend oldali logikához kapcsolódó részeknél. Az egyes függvények, adatkezelési logikák vagy API-hívások ellenőrzésére automatikus teszteseteket írtunk, hogy biztosak lehessünk azok helyes működésében. Bár a hangsúly a manuális tesztelésen volt, ezek a kisebb egységtesztek hozzájárultak a stabilitáshoz és a hibák korai felismeréséhez.

Összességében a választott tesztelési stratégia jól illeszkedett a munkamódszerünkhöz, gyors, rugalmas, és lehetővé tette a fejlesztési és tesztelési folyamatok párhuzamos futtatását.

**Továbbfejlesztés**

A fejlesztési folyamat során számos olyan ötlet és funkció merült fel, amelyek jelentősen hozzájárulhatnak oldalunk hatékonyabb és élvezetesebb működéséhez. Ezek közül azonban több megvalósítására a rendelkezésre álló idő korlátozottsága miatt nem nyílt lehetőség. Egyes elképzelések jelentős rendszerbeli átalakítást igényeltek volna, ezért úgy döntöttünk, hogy ezeket hosszabb távra ütemezzük át, jövőbeli fejlesztési célként kezelve őket.

Az egyik legnagyobb volumenű tervezett fejlesztés a többnyelvűség támogatása lenne. Ennek elsődleges célja, hogy az oldalunk angol nyelven is teljes mértékben elérhetővé váljon, és a felhasználók egy jól látható navigációs felületen elhelyezett zászló ikon vagy gomb segítségével könnyedén válthassanak nyelvet. Ezzel a nemzetközi elérhetőség és a felhasználói élmény is jelentősen növelhető lenne.

Szintén kiemelt figyelmet igényel az általunk jelenleg használt API-k korlátozottsága. Az ingyenesen elérhető API nem teszi lehetővé bármely felhasználó adatainak lekérdezését, továbbá a hozzáférést biztosító kulcs is mindössze 24 órán keresztül érvényes. A jelenlegi API nem támogatja a különböző régiókban való keresést, valamint a lekérdezések száma is erősen korlátozott. Ezek a tényezők jelentős akadályt képeznek a rendszer megbízható és folyamatos működése szempontjából. A probléma megoldására egy fizetős, megbízhatóbb API beszerzése jelenti a prioritást, amely nemcsak a lekérdezési lehetőségeket bővítené, hanem pontosabb és részletesebb adatokat is biztosítana a játékosokról és mérkőzéseikről.

Eredeti célkitűzésünk szerint az oldal több különböző játékot is támogatott volna, azonban gyorsan felismerésre került, hogy ez a megközelítés drasztikusan megnövelné a fejlesztési időt és a komplexitást. Ezért stratégiai döntésként jelenleg egyetlen játékra fókuszálunk, így biztosítva a magasabb szintű minőséget és a közösség koncentráltabb kiszolgálását.

Oldalunk jelenleg teljes egészében ingyenesen használható, legyen szó pusztán látogatásról vagy regisztrációról. A jövőben azonban szeretnénk bevezetni egy úgynevezett prémium rendszer bevezetését, amely kétféle formában lenne elérhető. Az egyik lehetőség egy egyszeri vásárlás lenne, amely révén a felhasználó prémium státuszt szerezne, ezáltal kiemelt támogatóvá válik az oldalon, profilján pedig egyedi cím és megkülönböztető jelzés jelenne meg. A másik opció egy havi előfizetés formájában lenne elérhető, amelyet „Boost” név alatt tervezünk bevezetni. Ez a szolgáltatás további lehetőségeket biztosítana a profil testre szabásában, valamint lehetővé tenné, hogy a felhasználó részletesebb és gazdagabb tartalmat jelenítsen meg az oldalán.

Az API jelenlegi korlátozásai miatt az egyik fő funkció, nevezetesen az élő játékoskeresés, nem működik megfelelően. Pedig eredeti koncepciónk szerint ez kulcsszerepet játszott volna az oldal működésében. A célunk az, hogy lehetővé tegyük akár több száz felhasználó aktuális játéktevékenységének valós idejű lekérdezését, így élő meccskeresési funkciót tudnánk biztosítani a felhasználók számára. Ennek megvalósítása szintén szoros összefüggésben áll a megbízható, fizetett API-rendszer beszerzésével.

További kisebb, ám fontos fejlesztési céljaink közé tartozik a profiloldalak testreszabhatóságának bővítése. Jelenleg csak minimális szintű egyedi beállítás érhető el, azonban hosszú távon szeretnénk lehetőséget biztosítani sokkal részletesebb és személyre szabottabb profilkialakításra.

A hírek oldalon szintén a közösségi élmény erősítését szolgálná egy új, oldalsávos megjelenítés, amely a felhasználó legfontosabb adatait, valamint barátlistáját tartalmazná. Ennek kiegészítéseként azonnali hírek és élő frissítések megjelenítését is tervezzük. Fontosnak tartjuk továbbá egy csoportkezelő funkció bevezetését, amely lehetőséget biztosítana új csoportok létrehozására, más csoportokhoz való csatlakozásra, valamint a csoportokon belüli kommunikációra és tartalommegosztásra. A csoportok szerkeszthetőek, törölhetőek lennének, a tagok számára pedig különböző jogosultsági szinteket tudnánk biztosítani, mint például szerkesztési vagy moderálási lehetőségek.

Végül, de nem utolsósorban az esports oldal jelentős fejlesztését is célul tűztük ki. Jelen állapotában nem lehetséges csapatok adatainak megtekintése vagy mérkőzéseik kezelése, ezért a jövőben ezen funkciók bővítése kiemelt figyelmet kap.

Összefoglalva elmondható, hogy számos fejlesztési tervet tartunk számon, amelyek megvalósítása nemcsak a felhasználói élményt növelné, hanem az oldal funkcionalitását is jelentősen bővítené. Törekvésünk, hogy lépésről lépésre minden tervezett újítást megvalósítsunk, és ezzel hosszú távon egy stabil, sokoldalú és vonzó platformot alakítsunk ki a felhasználóink számára.

**Munka elosztás**

A projekt munkafolyamatai átlátható módon lettek szétosztva a csapattagok között. A feladatokat oldalak szerint bontottuk le, így mindenki egy-egy funkcionális egység kialakításáért felelt. A felosztás során figyelembe vettük a csapattagok technikai felkészültségét és érdeklődési körét, ennek köszönhetően mindenki a saját tudásához és erősségeihez leginkább illő részfeladatokon dolgozhatott.

Fontos szempont volt, hogy mindenki kipróbálhassa magát a webfejlesztés különböző területein, ezért igyekeztünk úgy alakítani a munkamegosztást, hogy mindenkinek jusson frontend és backend programozás is, valamint legyen tapasztalata az adatbázis-tervezés és -kezelés terén is. Ez nemcsak a szakmai fejlődést szolgálta, hanem azt is elősegítette, hogy a csapattagok jobban átlássák a projekt egészének működését.

A fejlesztés során rendszeresen egyeztettünk a csapaton belül, így szükség esetén módosítottuk a feladatokat, vagy segítettük egymás munkáját. Ennek köszönhetően a csapatmunka gördülékenyen zajlott, és mindenki aktívan hozzájárult a projekt sikeréhez.

**Ambrus Erik Milán**

* Főoldal
* Bejelentkezés / Regisztráció
* Esports oldal

**Galbács Norbert**

* Profil oldal
* Tesztelés

**Gyuricza Ferenc Máté**

* Közösség oldal
* Chat és Barátok oldal
* API