

Budapesti Műszaki Szakképzési Centrum
Verebély László Technikum
5 0613 12 03 Szoftverfejlesztő és -tesztelő szakképesítés

Elmeecsata – Online kvízzjáték, készítő-játszó

Készítette: Rostás András Péter, Digruber Emese
Konzulens tanárok: Somogyi Erika, Horváth Attila

2025. Április

NYILATKOZAT A VIZSGAREMEK EREDETISÉGÉRŐL

Alulírott Rostás András Péter és Digruber Emese a BMSZC Verebély László Technikum 5 0613 12 03 Szoftverfejlesztő és -tesztelő képzésében részt vevő tanulói, büntetőjogi felelősségünk tudatában nyilatkozunk és aláírásunkkal igazoljuk, hogy az Elmezsata című vizsgaremek saját, önálló munkánk, és abban betartottuk az iskola által előírt, a vizsgaremek készítésére vonatkozó szabályokat. Tudomásul vesszük, hogy a vizsgaremekben plágiumnak számít:

- szó szerinti idézet közlése idézőjel és hivatkozás nélkül,
- tartalmi idézet hivatkozás megjelölése nélkül,
- más publikált gondolatainak saját gondolatként való feltüntetése.

E nyilatkozat aláírásával tudomásul vesszük továbbá, hogy plágium esetén vizsgaremekünk visszautasításra kerül.

Budapest, 2025. április 16.

Tanulók aláírása:

Rostás András Péter

Digruber Emese

Tartalomjegyzék

1.	Bevezetés, megismerkedés a projekttel.....	4
2.	Fejlesztői dokumentáció.....	5
2.1.	Alkalmazott fejlesztői eszközök	5
2.2.	Alkalmazott technológiák	5
2.3.	Futtatási környezet	5
2.4.	Fejlesztői ütemterv	5
2.5.	Telepítési útmutató.....	6
2.5.1.	A web telepítése.....	6
2.5.2.	Az asztali alkalmazás telepítése	7
2.6.	Fejlesztés menete	7
2.7.	Webes projekt szerkezete (<i>monorepo</i>).....	9
2.8.	Adatbáziskezelés és ORM (Objektum-relációs leképezés).....	11
2.9.	Főbb algoritmusok	14
2.9.1.	Világos / sötét téma	14
2.9.2.	Kvíz szerkesztő.....	15
2.9.3.	Kvíz lejátszás – valós idejű kapcsolat.....	17
2.10.	Az asztali alkalmazás felépítése, működése.....	18
3.	Felhasználói dokumentáció, felhasználói kézikönyv	19
3.1.	Vendég funkciók.....	19
3.2.	Regisztrált profillal rendelkező felhasználói funkciók.....	22
4.	További fejlesztési lehetőségek.....	28
4.1.	Napi/Heti kihívások.....	28
4.2.	Közöségi funkciók.....	28
4.3.	Mobilapp	28

1. Bevezetés, megismerkedés a projekttel

A vizsgaremek témája egy már meglévő szoftver ihlette, a [Kahoot](#). Az alap elgondolás az volt, hogy a felhasználók kvizeket tudjanak létrehozni, azokat szerkeszteni, megosztani, illetve a legfőbb feladata, hogy lehessen játszani bárkivel bárhol a világon valós időben. Fontosnak tartottuk, hogy ne egy Kahoot másolatot készítsünk, hanem az alap elgondolást bővítsük ki a saját elképzeléseinkkel és megoldásainkkal. Szerettünk volna egy korszerűbb kialakítású oldalt, amitől az embernek megjön a kedve ahhoz, hogy bizonyos témaköröket dolgozzon fel és azt gondoljuk, hogy ezt is sikerült megvalósítani. Az oldalon lehetőség van regisztrálni felhasználónév-jelszó párossal, majd létrehozni egy kvízt. A kvíznek 2 képe is feltölthető: egy borítókép és egy ikon. A képek mellett egy nevet, leírást lehet megadni. Választhatunk különböző kvíz címkék közül is, amelyek kategorizálják a kvízünket. Ilyenek például a „Történelem”, „Földrajz”, „Vallás”, „Politika” stb. Végsősoron kiválasztható a kvízünk nyelve, illetve, hogy publikus vagy privát legyen. Privát esetében csak a készítő látja és indíthatja el. A létrehozást követően egyből szerkeszthetjük a kvíz minden elemét, beleértve az előbb felsorolt lehetőségeket is. Úgy nevezett „slide” -okat hozhatunk létre, amelyek fognak lepörögni a lejátszás alatt. 4 féle slide típus közül lehet választani:

- Feleletválasztós
- Igaz – Hamis
- Infó slide
- Szavazás

Minden slide-ra külön-külön be lehet állítani, hogy hány másodpercig jelenjen meg, milyen sorrendben legyenek a kérdések, sőt még a slideokat is egyszerűen tudjuk mozgatni, ha netán nem tetszene a sorrend.

Asztali alkalmazásunk egy amolyan adminisztrációs felületként működik. Lehetőség van meglévő kvíz tagek, országok szerkesztésére vagy akár újak hozzáadására.

Amellett, hogy milyen összetett egy ilyesféle webapplikáció elkészítése, rendkívül izgalmas és motiváló is, hiszen rengeteg lehetőséget rejt magában a tanulás és a szórakozás ötvözése egy interaktív platformon keresztül.

A weboldal az elmezsata.hu-n elérhető és megtekinthető.

2. Fejlesztői dokumentáció

2.1. Alkalmazott fejlesztői eszközök

- IDE (Integrált fejlesztői környezet): [Visual Studio 2022](#), [Visual Studio Code](#)
- Csomag és monorepo-kezelő: [pnpm workspace](#)
- Adatbázis-adminisztrációs eszköz: [HeidiSQL](#)
- Fejlesztői környezet: [Node.js](#), [.NET Framework](#)
- Kódformázás és ellenőrzés: [Prettier](#), [ESLint](#)

2.2. Alkalmazott technológiák

- Programozási nyelvek: [TypeScript](#), [C#](#)
- Webes technológiák: [Next.js](#), [React](#), [TailWindCSS](#), [Socket.io](#)
- Adatbáziskezelés és ORM: [MariaDB/MySQL](#), [Drizzle ORM](#)
- UI komponens könyvtárak: [shadcn](#), [Origin UI](#), [Magic UI](#), [RealTailor](#)

2.3. Futtatási környezet

A weboldal bármilyen okos eszközön használható, amely rendelkezik internethozzáféréssel, legyen szó egy tabletről, asztali számítógépről vagy telefonról. Az asztali alkalmazás kizárólag asztali, Windows alapú számítógépeken működik.

2.4. Fejlesztői ütemterv

Határidő	Feladat
2024. december 19.	Témaválasztás, csoportalakítás
2025. január 27.	Weboldal sablon és drótváz elkészítése
2025. február 21.	Az adatbázis elkészítése
2025. február 24.	A weboldal autentikációjának elkészítése
2025 április 7.	A dokumentáció elkészítése
2025. április 7-11.	A vizsgaremek előzetes értékelése
2025. április 16.	A vizsgaremek leadása

2.5. Telepítési útmutató

2.5.1. A web telepítése

Szükségünk lesz pár programra, ami elengedhetetlen, hogy telepítve legyen a projekt elindításához:

- Nyissunk egy terminált (parancssor) és navigáljunk a projekt gyökerébe (./web).
- Győződjünk meg, hogy van-e feltelepítve Node.js a `node` paranccsal. Ha nem található a parancs (tehát ezt kapjuk vissza: `'qwe' is not recognized as an internal or external command, operable program or batch file.`) telepítsük [innen](#).
- Telepítsük a pnpm csomag-kezelőt a `winget install -e --id pnpm.pnpm` paranccsal.
- Telepítsük a Bun Node runtime-ot a `winget install --id=Oven-sh.Bun -e` paranccsal.
- Telepítsük a függőségeket a `pnpm install` paranccsal. Ez ilyenkor minden applikációba és csomagba letölti a neki szükséges függőségeket, ha van közöttük közös, akkor a gyöker mappába.
- Töltsük ki a környezeti változókat:
 - Navigáljunk a `./web/apps/next` mappába. Hozzunk létre egy `.env` fájlt és illesszük bele a `.env.example` fájl tartalmát. A `NEXTAUTH_SECRET` környezeti változó értéke bármi lehet, csak lényeg, hogy legyen benne valami. Szerezzünk ingyenesen egy kulcsot az [Uploadthing](#) oldaláról és illesszük be az `UPLOADTHING_TOKEN` értékének. A `NEXT_PUBLIC_UPLOADTHING_PROJECTID` legyen az Uploadthing-es projekt id.
 - Navigáljunk a `./web/apps/ws` mappába. Hozzunk létre egy `.env` fájlt és illesszük bele a `.env.example` fájl tartalmát.
 - Navigáljunk a `./web/packages/database` mappába. Hozzunk létre egy `.env` fájlt és illesszük bele a `.env.example` fájl tartalmát.
- Indítsunk el egy helyi MariaDB vagy MySQL szerver a 3306-os alapértelmezett porton. (Ha valamilyen okból kifolyólag más a port vagy a hosztnév, akkor értelemszerűen a `.env` fájlokban azt kell megadni az adatbázis-kapcsolat résznél.)
- Hozzunk létre egy „elmecsata” nevű adatbázist.
- Futtassuk a migrációkat, hogy legenerálódjanak a táblák a séma alapján a `pnpm run migrations:migrate` paranccsal. Fontos, hogy ezt a gyöker (./web) mappában tegyük.
- Indítsuk el a projektet fejlesztői módban a `pnpm run dev` paranccsal. Ilyenkor mind a WebSocket szerver és a Next applikáció is elindul. A konzolba érkező üzeneteket színük alapján tudjuk megítélni, hogy honnan érkezik:
 - Next: Magenta
 - WebSocket: Kék

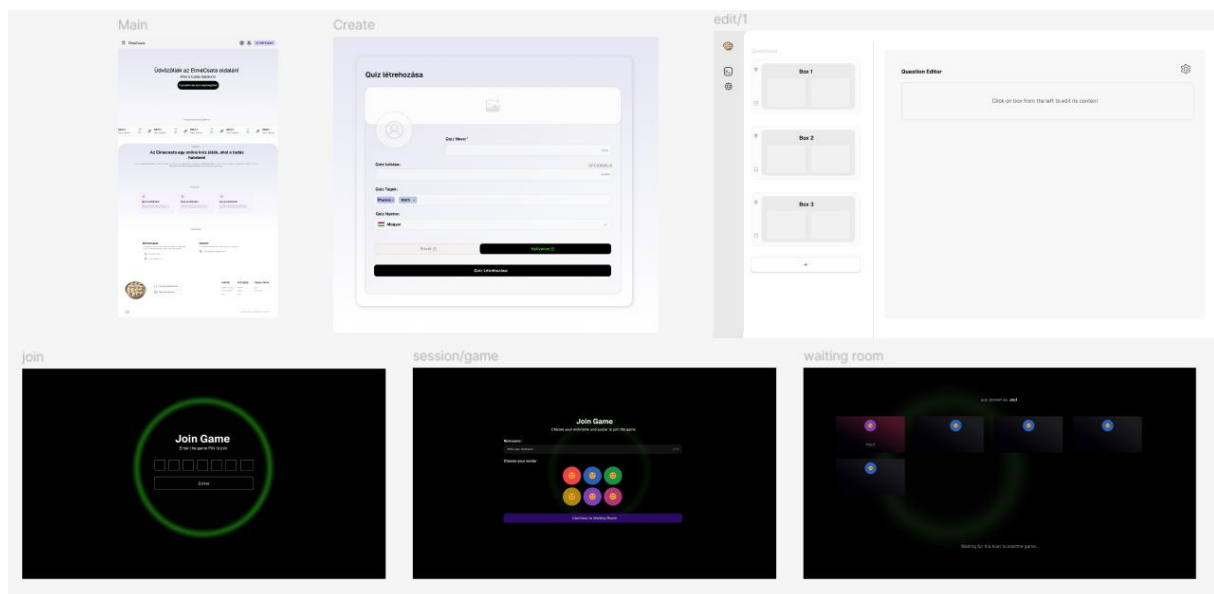
2.5.2. Az asztali alkalmazás telepítése

- Nyissa meg a desktop mappát, majd kattintson azon belül a vizsgaremek.sln fájlra (duplán) a projekt megnyitásához.
- A Visual Studio felületén, a bal felső sarokban található menüsorban kattintson a Project menüpontra, majd a lenyíló menüből válassza a Manage NuGet Packages... lehetőséget.
- Miután megnyílt a NuGet csomagkezelő, kattintson az **Installed** fülre. Itt megjelennek a projektben jelenleg telepített NuGet-csomagok, így egyszerűen megtalálhatja a [RealTaiizor](#) nevű komponenst. Válassza ki, majd kattintson az **Uninstall** gombra a komponens eltávolításához.
- Ezután váltson át a **Browse** fülre, és keresse meg ismét a korábban eltávolított **RealTaiizor** komponenst. Ha megtalálta, kattintson az **Install** gombra a csomag újratelepítéséhez. A telepítést követően már elindítható az alkalmazás, és a komponens megfelelően fog működni.
- **Fontos:** Ahhoz, hogy az alkalmazás megfelelően működhessen, szükség van az adatbázis elindítására is. A legegyszerűbb és leggyorsabb megoldás, ha végig követi a webes telepítéshez tartozó útmutatót, mivel az adatbázist így is migrálni kell majd a későbbiekben.

2.6. Fejlesztés menete

Mielőtt bármilyen fejlesztésbe kezdtünk volna, fontosnak tartottuk felállítani a sorrendet, hogy az alkalmazásnak mely funkcióit valósítsuk meg elsőként, és azokat milyen sorrendben, hiszen ezek egymásra épülhetnek. Tehát nem kezdhettünk egyből a Kvíz lejátszás részével úgy, hogy még kvizeket sem lehetett létrehozni. Ez segített abban, hogy kizárjuk a rengeteg hiba lehetőségét és az alkalmazás bizonyos, kész részeinek újra gondolását.

Tervezéskor a UI/UX részével kezdtünk: ezalatt egy teljes képet kaptunk arról, hogy milyen interakciókat végezhet majd a felhasználó. Gyakorlatilag maga az applikáció tervezése a UI és a UX kialakítása alatt történt. Erre a célra használtuk a [Figma](#) nevű szoftvert, hogy a lehető legjobb képet kapjuk az elképzeléseinkről.



1. ábra - A Figma-ban készült alap design ötlet

Ahhoz, hogy a kódolást ellehessen kezdeni megvalósítani, kutatómunkát kellett végeznünk, hogy megtaláljuk a legmegfelelőbb technológiákat, keretrendszereket és könyvtárakat annak érdekében, hogy megkönnyítsük a fejlesztést. Így esett a választás a **Next.js**-re, amely a **React**-alapú webes alkalmazásokat kínál a szerveroldali és statikus megjelenítéssel.

2.7. Webes projekt szerkezete (*monorepo*)

A projekt egy monorepo¹ struktúrát követ, amely két fő alkalmazásból áll:

WebSocket alkalmazás:

- Ez az applikáció indít egy HTTP- és egy WebSocket-szerveret. Fő feladata a valós idejű kommunikáció biztosítása, a folyamatban lévő játékok kezelése, események továbbítása a frontend részére, és a játékmenet logikájának kiszolgálása.

Next.js alkalmazás:

- Ez a frontend applikáció felelős minden felhasználói felülettel kapcsolatos funkcióért, például a regisztrációért, illetve a kvizek szerkesztéséért és a kitöltéséért.

Packages (csomagok):

Saját csomagokat hoztunk létre annak érdekében, hogy a közösen használt logika és típusdefiníciók könnyen újrahasznosíthatók legyenek a különböző alkalmazások között.

Mindkét alkalmazás TypeScript alapú. A TypeScript nagy előnye – ahogy azt a neve is mutatja – a típusbiztonság:

JavaScript kódot írhatunk úgy, hogy közben a típusok biztosítják a megbízhatóságot és a hibák korai kiszűrését. Ehhez azonban elengedhetetlen, hogy mindenhol következetesen ugyanazokat a típusokat használjuk, különben könnyen típusütközések és hibák léphetnek fel.

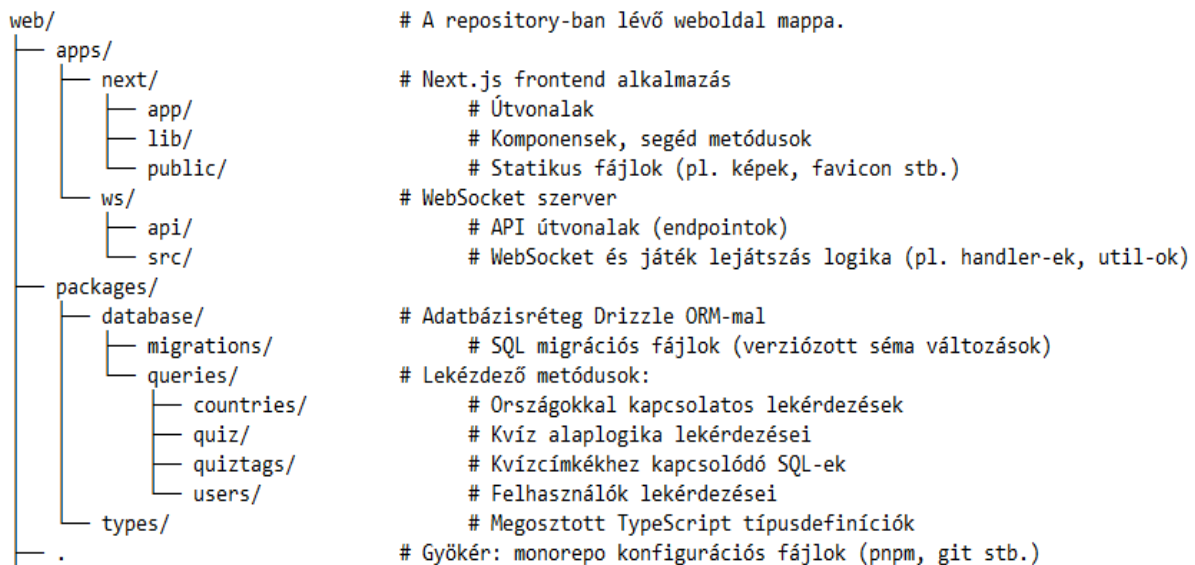
packages/types:

Ebben a csomagban található a projektben használt összetett megosztott TypeScript típusdefiníció. Például a játéklogikához, országokhoz, kvizekhez és felhasználókhoz kapcsolódó típusok innen származnak. Ezeket mind a frontend, mind a backend alkalmazás importálja, így garantálható az egységes típushasználat.

¹ monorepo: A monorepo egy közös kódtár, amelyben több projekt vagy csomag együtt kezelhető és fejleszthető.

packages/database:

Ez a csomag felel az adatbáziskapcsolatért és a Drizzle ORM beállításaiért. Tartalmazza az adatbázis sémáját, a migrációs fájlokat, valamint az SQL-lekérdezéseket. Mivel TypeScript-ben íródott, a types csomag típusait is közvetlenül be lehet benne írni, így az adatbázisos műveletek pontosan olyan típusokkal dolgoznak, amilyenekkel kell. A lekérdezések külön metódusokba vannak rendezve, ezeket a Next.js és a WebSocket alkalmazás is ugyanúgy eléri, így nem kell kétszer megírni ugyanazt, és minden ott van, ahol lennie kell.



2. ábra - Web struktúrája, felépítése

```
import { getUserById } from '@elmeccsata/database';
import { Quiz } from '@elmeccsata/types';
```

3. ábra - Az adatbázis meghívásához használt kód

2.8. Adatbáziskezelés és ORM (Objektum-relációs leképezés)

Az adatbázismotornak a MariaDB-t választottuk, ami a MySQL egyik változata (úgynevezett „fork” -ja). Azért döntöttünk mellette, mert stabil, gyors, és sok projektben bevált megoldás.

Az adatbázis kezelését megkönnyíti a Drizzle ORM, amit a projektben használtunk. Ez egy TypeScript-alapú eszköz, amit egyszerre lehet lekérdezésekhez (mint egy „query builder”) és ORM-ként is használni. Emellett ez szolgál az SQL séma forrásaként, vagyis ebben írjuk le az adatbázis felépítését. Ráadásul saját parancssoros eszközt (CLI-t) is ad, amivel automatikusan tudunk migrációs fájlokat generálni, így nem kell kézzel írni őket. Elsőre ezek a fogalmak bonyolultnak tűnhetnek, de a gyakorlatban a működés meglepően egyszerű. Képzeljünk el egy fájlt – a mi esetünkben ez a `schema.ts`. Ebben írjuk le az összes adatbázis-táblát, azok mezőit (pl. szöveg, szám, dátum stb.), valamint a táblák közötti kapcsolatokat. Ha valamit módosítunk ebben a fájlban, például új oszlopot adunk hozzá egy táblához, a Drizzle CLI segítségével egyetlen parancs kiadásával generálhatunk egy új migrációs fájlt, ami végrehajtja ezt a változást az adatbázisban. Fontos, hogy ilyenkor csak is a változtatáshoz szükséges SQL parancsokat kapjuk meg: például, hogy ha egy oszlopot adtunk hozzá egy táblához, akkor csak is kizárólag ennek az utasítása lesz a migrációs fájlban. Ahogy a fejlesztés halad úgy a migrációs fájlok alapján visszavezethető a teljes adatbázis struktúra. Ha egy új számítógépre tesszük fel a projektet, akkor a migrálást a legelső migrációs fájlal kezdjük és így halad egyesével végig az összesen.

```
1  const users = mysqlTable('users', {
2    id: int('user_id').autoincrement().primaryKey(),
3    username: varchar('username', { length: 15 }).notNull(),
4    password: varchar('password', { length: 60 }).notNull(),
5    bio: varchar('bio', { length: 100 }).notNull(),
6    avatar: varchar('avatar', { length: 48 }).notNull(),
7    created_at: timestamp('created_at').defaultNow(),
8    updated_at: timestamp('updated_at').defaultNow().onUpdateNow(),
9  });
```

4. ábra - A users tábla sémája

Users tábla:

- user_id → felhasználó egyedi azonosítója, int, auto_increment, PK
- username → felhasználó felhasználóneve, varchar(15), not null
- password → felhasználó jelszava, varchar(60), not null
- bio → felhasználó bemutatkozó szövege, varchar(100), not null
- avatar → profilkép fájlazonosítója, varchar(48), not null
- created_at → regisztráció időpontja, timestamp, defaultNow()
- updated_at → utolsó módosítás időpontja, timestamp, defaultNow(), onUpdateNow()

Quizzes tábla:

- quiz_id → kvíz egyedi azonosítója, int, auto_increment, PK
- name → kvíz címe, varchar(30), not null
- icon → kvízhez tartozó ikon fájlazonosítója, varchar(48), not null
- banner → kvíz borítóképe, varchar(48), not null
- description → kvíz leírása, varchar(255), not null
- creator → kvízt létrehozó felhasználó azonosítója, int, FK, not null
- tags → kvízhez tartozó címkék, not null
- country → ország azonosítója, int, FK, not null
- isPrivate → kvíz privát állapotát jelző mező, int, not null
- questions → kvíz kérdései, not null
- played → kvíz lejátszásainak száma, int, not null
- created_at → kvíz létrehozásának időpontja, timestamp, defaultNow()
- updated_at → utolsó módosítás időpontja, timestamp, defaultNow(), onUpdateNow()

Quiz_tags tábla:

- quiz_tag_id → címke egyedi azonosítója, int, auto_increment, PK
- tag → címke értéke, varchar(50), not null
- label → címke megjelenített neve, varchar(50), not null
- color → címkéhez tartozó színkód, varchar(50), not null

Countries tábla:

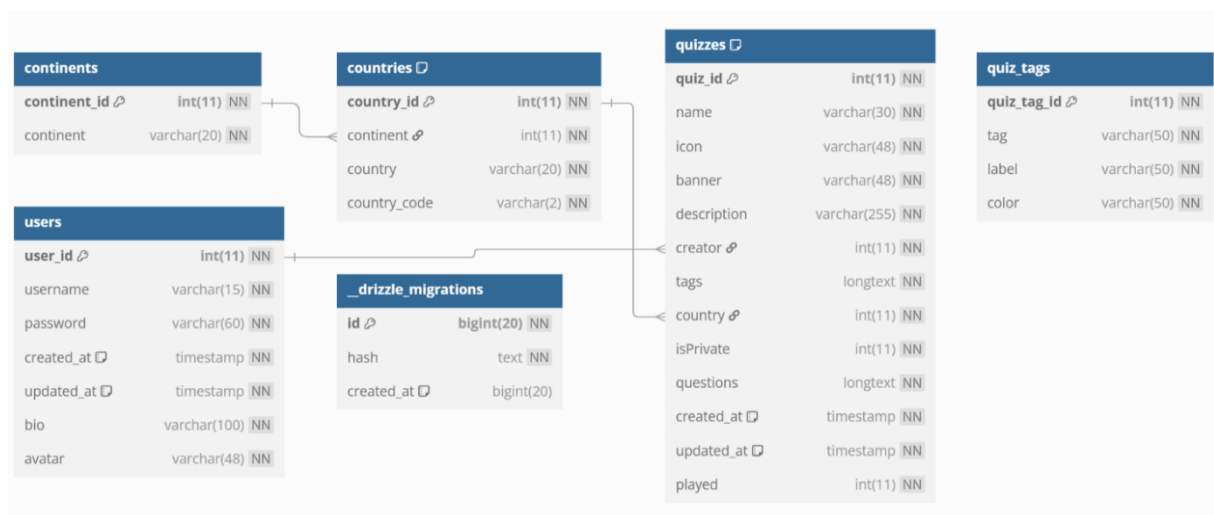
- country_id → ország egyedi azonosítója, int, auto_increment, PK
- continent → kontinens azonosítója, int, FK, not null
- country → ország neve, varchar(20), not null
- country_code → ország kétbetűs kódja, varchar(2), not null

Continents tábla:

- continent_id → kontinens egyedi azonosítója, int, auto_increment, PK
- continent → kontinens neve, varchar(20), not null

```
1 const usersRelations = relations(users, ({ many }) => {
2   return {
3     quizzes: many(quizzes),
4   };
5 });
```

5. ábra - Kapcsolat a users és a quiz tábla között



6. ábra - Az adatbázis EER diagramja

2.9. Főbb algoritmusok

2.9.1. Világos / sötét téma

A mai modern weboldalakon szerintünk elengedhetetlen az, hogy a felhasználó különböző megjelenési téma között választhasson a felhasználói felületre. Erre a leggyakoribb példa a világos/sötét téma. Ennek tudatában implementáltuk világos és sötét módot minden oldalra. Azt, hogy a felhasználó milyen témát választott ki sütikben tároljuk. Ennek az az oka, hogy szerver oldalon a kérés megérkezésekor már megkaphatjuk a választott témát, így „szerver oldali renderelés”²-nél már a választott témát tudjuk használni ezzel elősegítve a gyors betöltést. Erre egy úgynevezett provider³-t készítettünk, amely lehetővé teszi azt, hogy az applikációkban bárhol könnyedén el tudjuk érni a választott témát.

```
1 export const ThemeProvider = ({ children }: { children: React.ReactNode }) => {
2   const [theme, setTheme] = useState('light');
3   const systemPrefersDark = useMediaQuery({ query: '(prefers-color-scheme: dark)' }, undefined, (isSystemDark) =>
4     setTheme(isSystemDark ? 'dark' : 'light')
5   );
6
7   useEffect(() => {
8     const defaultTheme = Cookies.get('theme');
9     const theme = defaultTheme || (systemPrefersDark ? 'dark' : 'light');
10
11     switchDefaultTheme(theme);
12   }, [theme, systemPrefersDark]);
13
14   useEffect(() => {
15     Cookies.set('preferred-theme', systemPrefersDark ? 'dark' : 'light', { expires: 400 });
16   }, [systemPrefersDark]);
17
18   const switchDefaultTheme = (theme: string) => {
19     setTheme(theme);
20     Cookies.set('theme', theme, { expires: 400 });
21     const root = document.documentElement;
22     root.classList.remove('light', 'dark');
23     root.classList.add(theme);
24     root.style.colorScheme = theme;
25   };
26
27   return (
28     <ThemeProviderCtx.Provider
29       value={{
30         switchDefaultTheme,
31         theme,
32       }}
33     >
34       {children}
35     </ThemeProviderCtx.Provider>
36   );
37 };
```

7. ábra - A sötét / világos téma algoritmus

² szerver oldali renderelés: A weboldalt a szerver állítja össze, nem a böngésző.

³ provider: Adatok megosztására szolgáló eszköz React-ben.

2.9.2. Kvíz szerkesztő

Kétségtelen, hogy a webapplikáció egyik leglényegesebb pontja az a kvíz szerkesztő, ugyanis a felhasználók itt tudják a létrehozott kvízek kérdéseit törölni, szerkeszteni. Több fajta kérdés típust lehet hozzáadni egy kvízhez, mint például a „Feleletválasztós” vagy „Igaz-Hamis”. Ez viszont azt eredményezi, hogy nem tudjuk az összes kérdést ugyanabban az adatszerkezetben eltárolni, hiszen kérdés típustól függően eltérő mezőkre van szükségünk. E probléma kiküszöbölésére egy egyszerű JSON⁴ formátumban tároljuk a kérdéseket. Erre készült a lent látható algoritmus, amely közvetlen egy kvíz lekérdezésekor dolgozza fel a kvíz kérdéseit és alakítja át a megfelelő típusúvá.

```
1  const formatQuizTypes = async (quizData: RawQuiz[]): Promise<Quiz[]> => {
2    const formattedQuizzes: Quiz[] = await Promise.all(
3      quizData.map(async (quiz) => {
4        const tagIds = JSON.parse(quiz.tags as string);
5        const tags = await Promise.all(
6          tagIds.map(async (tagId: number) => {
7            return await getQuizTagById(tagId);
8          })
9        );
10       });
11     let questions: Question[] = [];
12
13     try {
14       const unescapedString = unescape(quiz.questions as string).trim();
15
16       let parsedData = JSON.parse(unescapedString);
17
18       if (typeof parsedData === 'string') {
19         parsedData = JSON.parse(parsedData);
20       }
21
22       if (!Array.isArray(parsedData)) {
23         throw new Error('Parsed questions is not an array: ' + JSON.stringify(parsedData));
24       }
25
26       questions = parsedData.map((q: any) => ({
27         id: q.id,
28         title: q.title,
29         content: q.content ?? undefined,
30         type: q.type,
31         duration: q.duration ?? 20,
32         answers: q.answers.map((a: any) => ({
33           id: a.id,
34           text: a.text,
35           isCorrect: a.isCorrect ?? undefined,
36         })) as QuestionOption[],
37       })) as Question[];
38     } catch (error) {
39       throw new Error('Error while parsing questions: ' + error);
40     }
41
42     return {
43       id: quiz.id,
44       name: quiz.name,
45       icon: quiz.icon,
46       banner: quiz.banner,
47       description: quiz.description,
48       creator: quiz.creator,
49       tags,
50       country: quiz.country,
51       isPrivate: !!quiz.isPrivate,
52       questions,
53       played: quiz.played,
54       created_at: quiz.created_at as Date,
55       updated_at: quiz.updated_at as Date,
56     };
57   });
58 };
59 return formattedQuizzes;
60 };
```

8. ábra - Kvíz szerkesztésének algoritmus

⁴ JSON: Egyszerű adatformátum, amit a gépek is értenek.



```

1  [
2    {
3      "id": "1",
4      "title": "Mi magyarország fővárosa?",
5      "type": "quiz",
6      "answers": [
7        {
8          "id": "bfe88a2a-04d5-423f-9b5b-54a602a01080",
9          "text": "Vecsés",
10         "isCorrect": false
11       },
12       {
13         "id": "caee9c2-3832-4d60-92a7-af15e67d9f3a",
14         "text": "Budapest",
15         "isCorrect": true
16       },
17       {
18         "id": "8264d29c-d733-4b2c-ae3b-44e20fa036d7",
19         "text": "Debrecen",
20         "isCorrect": false
21       },
22       {
23         "id": "8c91e3c8-4a3f-4ba6-880d-a6c7fc1e494f",
24         "text": "Pozsony",
25         "isCorrect": false
26       }
27     ],
28     "duration": 10
29   },
30   {
31     "id": "3",
32     "title": "Új szöveges slide",
33     "type": "slide",
34     "content": "Add meg a slide tartalmát...",
35     "answers": [],
36     "duration": 10
37   }
38 ]
39

```

9. ábra - átalakítás a megfelelő formátumra

2.9.3. Kvíz lejátszás – valós idejű kapcsolat

Ahhoz, hogy valós időben, az oldal betöltése után tudjon a kliens és a szerver adatot cserélni, más protokollt kell használnunk, hiszen a http protokoll a kérés teljesítése után lezárja a kapcsolatot. Ellentétben a http-vel, a WebSocket protokoll kapcsolatot nyit a szerverrel, ami egészen addig marad nyitva, amíg el nem hagyjuk az oldalt. Ez lehetővé teszi, hogy interakciókra, például egy gomb lenyomásra szóljunk a szervernek, hogy: *„Helló-belló, megnyomtam a gombot, értesítsd a többi felhasználót.”*. Ez az alapja a játék résznek is. Lejátszás indításakor egy kódot hoz létre, amellyel csatlakozni tud bárki. Ezt a kódot használva egy „szobát” hoz létre a WebSocket szerver, amihez csatlakozunk. Így nyomon tudjuk követni, hogy melyik felhasználó melyik játékhoz csatlakozott, és csak az adott játékuk adatait és eseményeit küldjük le számukra. A szerver ilyenkor elindít egy játékot, ami gyakorlatilag egy „while ciklus”: egészen addig megy, amíg el nem fogynak a kérdések. Folyamatosan gyűjti a beérkező válaszokat, tovább lép, ha mindenki válaszolt és feldolgozza azokat. A játék végén összesíti a pontokat és értesít mindenkit, hogy vége lett és jelenjen meg a pódium a már pontok alapján szortírozott adatokkal.

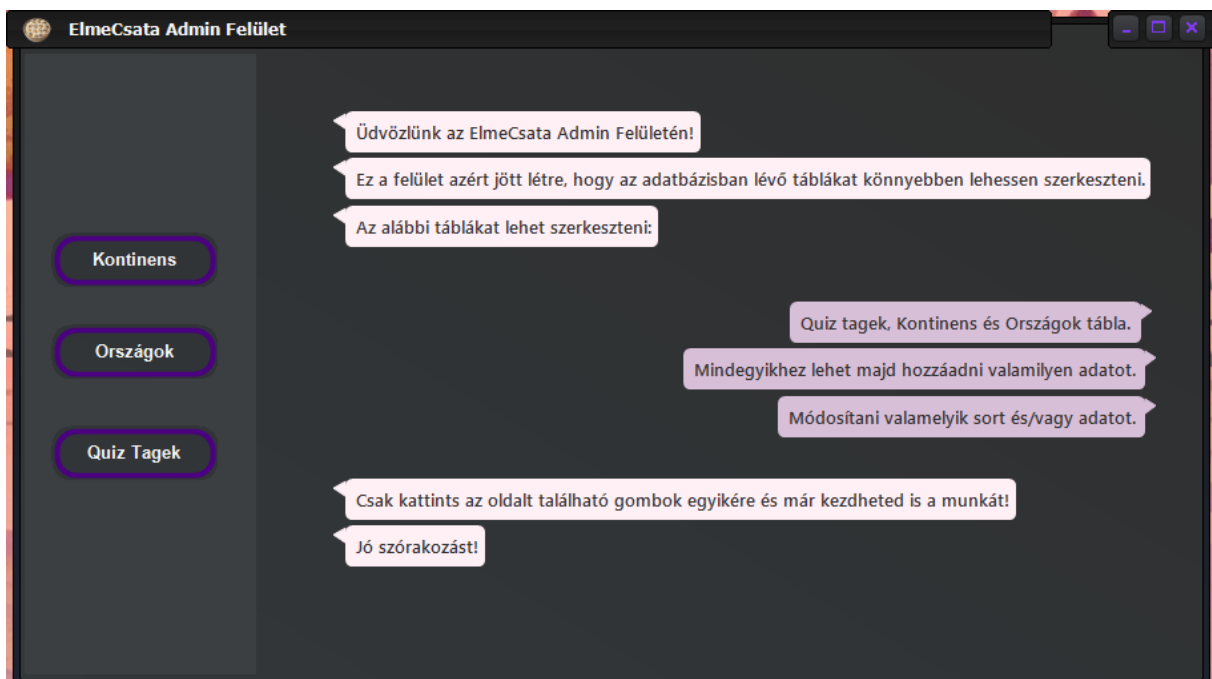
```
1 export const Games: Game[] = [];
2 let timesUpTimer: NodeJS.Timer;
3
4 export const endQuestion = (game: Game) => {
5   game.answers = 0;
6   clearTimeout(timesUpTimer);
7   sortPlayers(game);
8
9   io.to(game.code).emit('everyone-answered', game);
10
11   setTimeout(() => {
12     showCurrentLeaderboard(game);
13     setTimeout(() => {
14       nextQuestion(game);
15     }, 3000);
16   }, 3000);
17 };
18
19 export const nextQuestion = (game: Game) => {
20   console.log('Current question: ' + game.currentQuestion);
21   if (game.currentQuestion < game.quiz.questions.length - 1) {
22     game.currentQuestion++;
23     io.to(game.code).emit('next-question', game);
24
25     const nextQuestion: Question = game.quiz.questions[game.currentQuestion];
26     timesUpTimer = setTimeout(() => {
27       timesUp(game);
28     }, nextQuestion.duration * 1000);
29
30     return;
31   }
32
33   endGameLeaderboard(game);
34 };
```

10. ábra - Kvíz lejátszásának valós idejű algoritmus

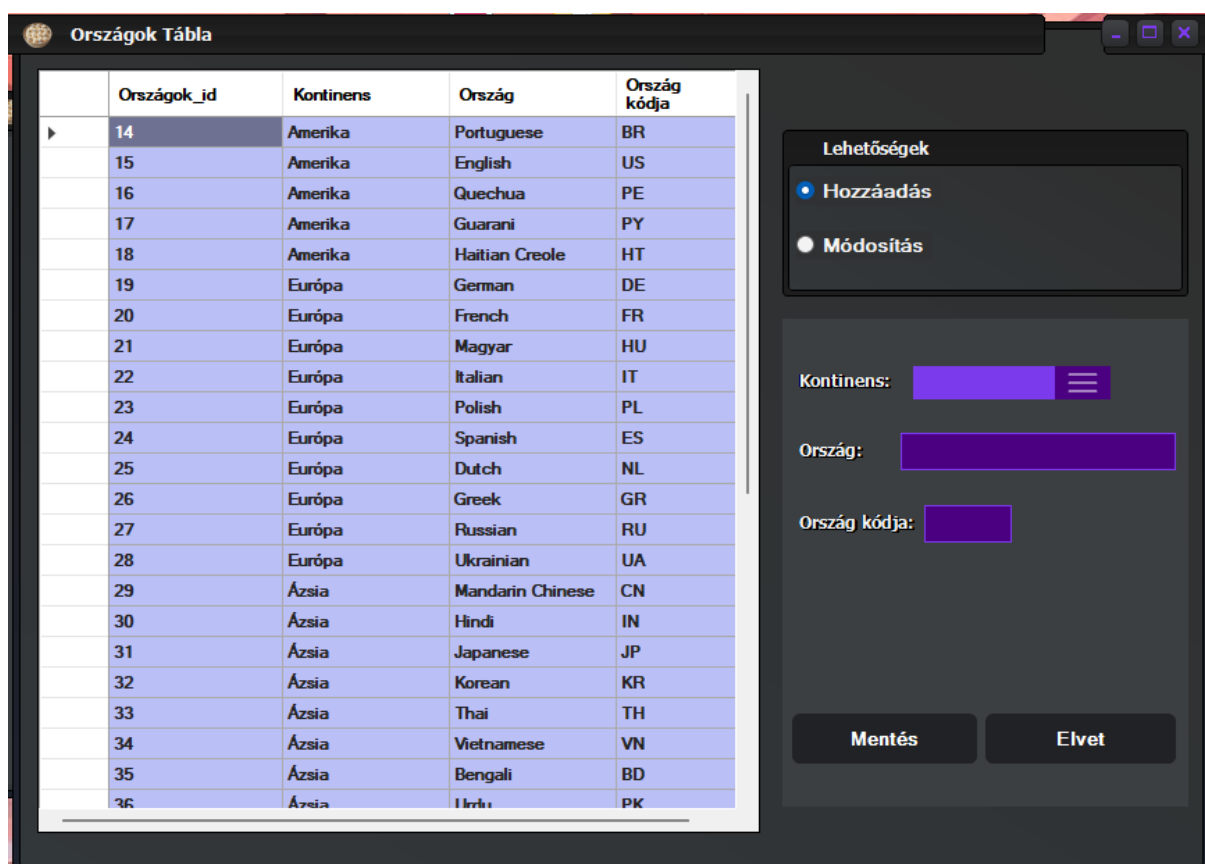
2.10. Az asztali alkalmazás felépítése, működése

Ennek a projektnek része volt még az, hogy egy asztali alkalmazás is készüljön mellé. A C# Windows Forms alkalmazás a .NET Framework környezetben készült, és belső, adminisztrációs célokra szolgál. A felületet kizárólag a fejlesztők használják, a publikus webes rendszer részeként nem elérhető. Az alkalmazás lehetőséget biztosít három adatbázistábla – Kontinens, Országok és Quiz Tag-ek – szerkesztésére. A felhasználók új adatokat adhatnak hozzá ezekhez a táblákhoz, valamint meglévő adatokat módosíthatnak. Minden művelet közvetlenül az adatbázison keresztül történik, lekérdezések és módosító utasítások segítségével, így a változások azonnal mentésre kerülnek az adatbázisban.

A felhasználói felület kialakításához kerestünk egy számunkra megfelelő komponenscsomagot, mivel fontosnak tartottuk, hogy a felület – még ha csak egy belső, adminisztrációs célokat szolgáló alkalmazásról is van szó – átlátható és esztétikusan rendezett legyen. Így találtunk rá a [RealTaiizor](#) készítőnek a komponens könyvtárára, amely számos modern megjelenési lehetőséget kínál. Bár az alkalmazás nem a nagyközönség számára készült, igyekeztünk a dizájnt koherensen illeszteni a webes rendszer megjelenéséhez. Ennek köszönhető például a domináns lila színvilág, itt is visszaköszön úgy, mint a webes felületen.



11. ábra - Az indítás után kapott felület



12. ábra - Példa, hogy hogyan néz ki egy tábla szerkesztése

3. Felhasználói dokumentáció, felhasználói kézikönyv

A weboldalt vendégként is lehet használni a kvíz lejátszására, tehát nem kötelessége a felhasználónak regisztrálni, ha csak játszani szeretne. Ellenben a kvizek létrehozásához, csak regisztrált fiókkal van erre lehetőség.

3.1. Vendég funkciók

A főoldal alapvetően mindenkinek megegyező, függetlenül attól, hogy a felhasználó regisztrált fiókkal tekinti ezt meg, vagy csak „vendégként” lép fel az oldalra. Tájékoztató jellegű információkat talál rajta a felhasználó, például, hogy milyen népszerű kvizek léteznek az oldalon, a csatlakozási lehetőség, általános tájékoztatás

az oldal előnyeiről, illetve az elérhetőséget is itt tűntettük fel. Az oldal alján fellelhető a készítő nevei és a hozzátartozó tevékenységeik. Nem utolsó sorban az általános szabályzatok, közösségi oldalakat és ehhez hasonló fontos információk.

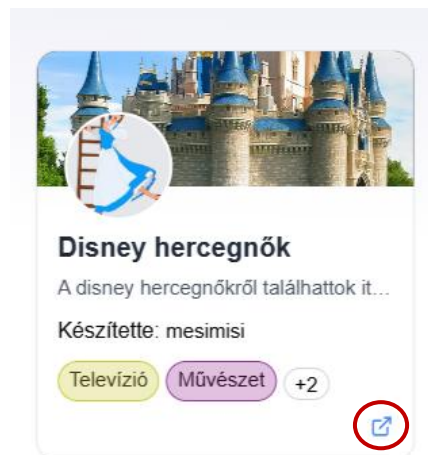


13. ábra - Bejelentkezés és a világos/sötét téma állítása

Tartalmaz még sötét és világos módot is, így a felhasználó az igénye szerint tudja az oldal kinézetét változtatni.

A kvíz kérdéseinek megtekintéséhez, elég csak erre a kis ikonra kattintani, és már látja is a felhasználó a kvíz előnézetét. Itt található még egy általános tájékoztató a játék elindításáról és egy statisztika is, az általános kérdésekről, ami esetleg egy felhasználóban felmerülhet, például: Átlagos időkorlát.

A „játék indítása” gombbal, a felhasználó, aki elindította a kvízt (Ő lesz a Házigazda) átugrik a csatlakozás oldalra, ahol egy random generált kóddal, illetve egy QR-kóddal fogja szembe találni magát, így könnyítve meg a csatlakozást a többi játékosnak.



14. ábra - A kvíz előnézetének megtekintése



15. ábra - A várószoba csatlakozásnál

👑 Házigazda mód

Házigazdaként csak megfigyelheted a játékot, nem válaszolhatsz a kérdésekre.

Marvel tudástár

Teszteljétek a tudásotokat a Marvel világban és győzzön a legjobb!

Kérdés 1/7

🕒 4s

Ki alapította az eredeti Bosszúállókat a filmekben?

▲ Amerika Kapitány

◆ Thor

● Nick Fury

■ Hulk

16. ábra - Házigazda mód a kvíz lejátszásánál

Marvel tudástár

Teszteljétek a tudásotokat a Marvel világban és győzzön a legjobb!

Kérdés 2/7

🕒 13s

Igaz vagy hamis: A Végtelen Kövek száma összesen hét.

▲ Igaz

◆ Hamis

17. ábra - Kérdés a játékosoknál

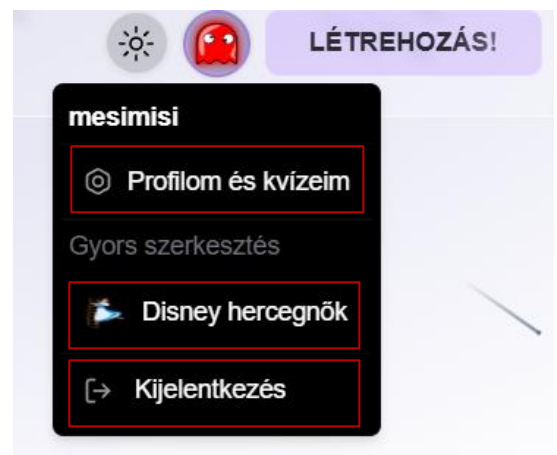
A Házigazdának lehetősége van moderálni a játékosok között, eltávolíthatja őket a játékból, ha úgy gondolja, erre szükség van. A játék elindításakor visszaszámol, majd elkezdődik a versengés. A Házigazda látja a kérdéseket és válaszokat, de nem játszat, ellenben tovább pörgetheti őket.

A játékosok hasonlóképpen látják a kérdéseket és a válaszokat, nekik a választási lehetőség is opció, tovább pörgetni kérdéseket viszont nem tudnak. A végén az összesített pontszámmal láthatják az első három helyezést pódiumon, a többi játékos alatta van felsorolva a helyezési számával. A játék befejeztével csak simán a „vissza a főoldalra” gombra kattintva (értelemszerűen) visszadob a főoldalra.

3.2. Regisztrált profillal rendelkező felhasználói funkciók

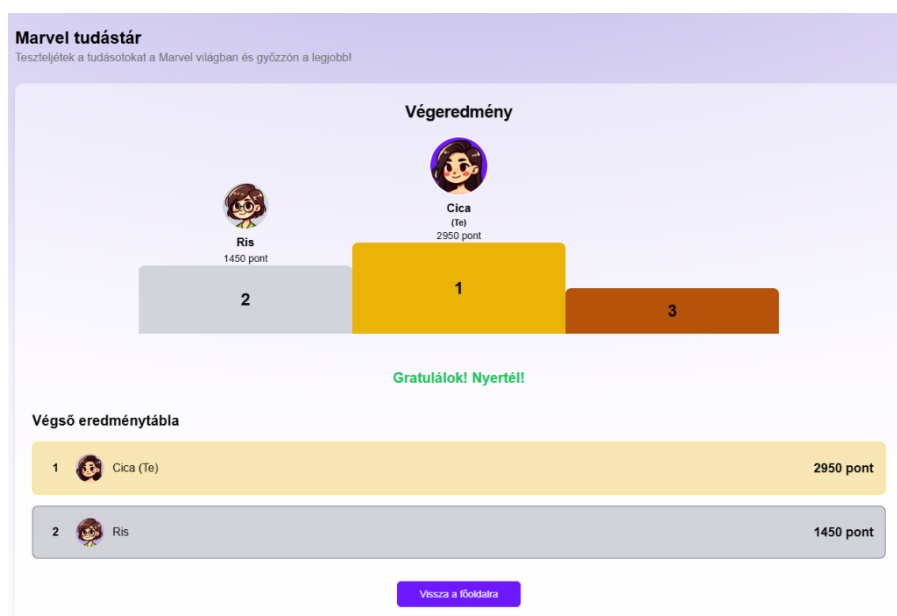
A regisztrálás/bejelentkezés után a profil ikonra kattintva látható három menüikon.

- A „Profilom és kvizeim” -re kattintva a felhasználó megtekintheti és szerkesztheti a profilját, valamint az általa létrehozott kvizeket.
- A „Gyors szerkesztés” alatt látható a felhasználó kvizei, amiket létrehozott. Bármelyikre rákattintva átvissz az adott kvíz szerkesztésére. Itt lehet módosítani a kérdések/válaszokat és a sorrendet is. (Ez persze bővülni fog, ahogy a felhasználó gyártja a kvizeket.)



18. ábra - Bejelentkezés után található három opció

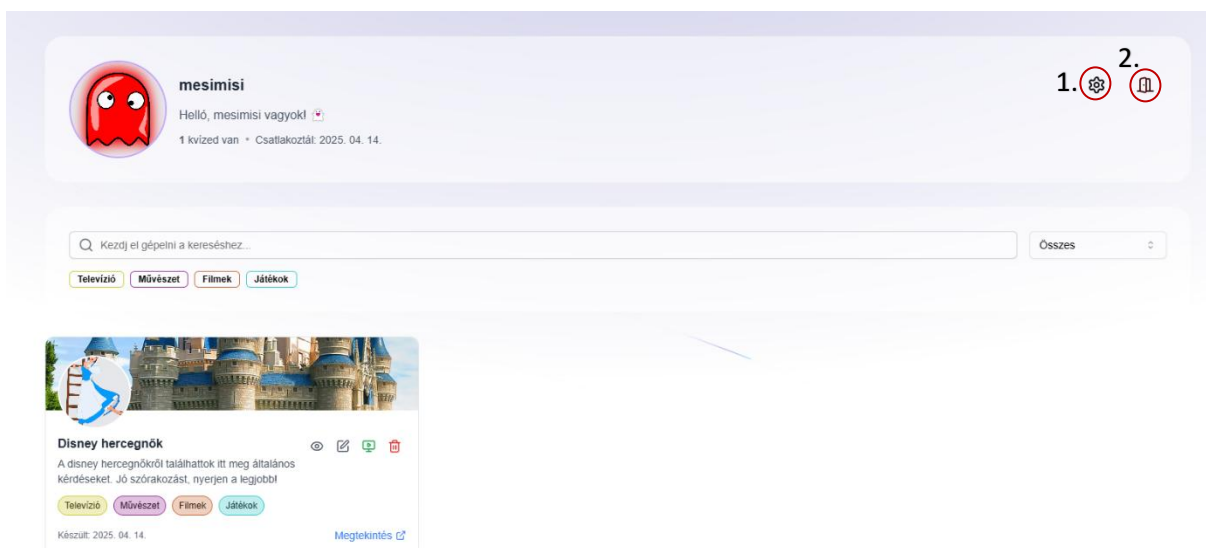
- A „Kijelentkezés” -re kattintva a fiókjából való kilépést teszi lehetővé a felhasználónak.



19. ábra - A játék végén lévő pontszámok és pódium

A „Profilom és kvizeim” előnézete. Ha a felhasználó erre rákattint, ez fogja őt fogadni. Általános kép a profiljáról, adatairól, kvizeiről.

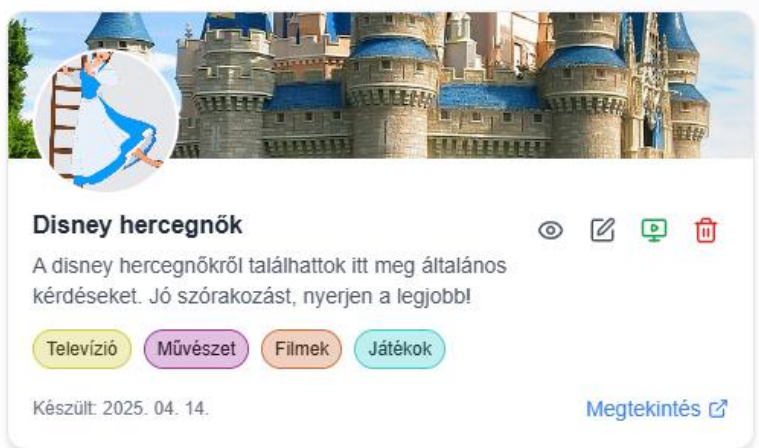
1. Az első pontnál van lehetőség szerkeszteni a profilt. A felhasználónév, bemutatkozás (bio), esetleg egy új jelszó beállítására is van opció. Nem utolsó sorban a regisztrált fiókot törölni is lehet, azonban ilyenkor az összes profilhoz tartozó kvíz is törlődni fog.
2. A második pontnál szintén ki lehet jelentkezni az adott profilból.



20. ábra - A profil oldal előnézete

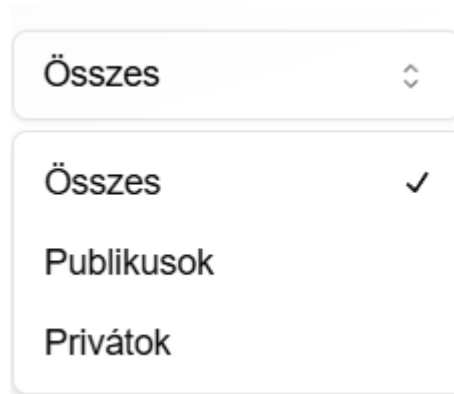
Lentebb található a felhasználó kvizei. A kártyán látható a kvíznek egy kis ismertetője, a tagek, amik a kvíz besorolásában segít.

- Szemikon: Opció, a kvíz privát vagy nyilvános legyen.
- Szerkesztés ikon: A kvíz szerkesztése, módosítása.
- Játék indítása ikon: A kvíz elindítására alkalmas.
- Kuka ikon: A kvíz törlése, eltávolítása.
- Megtekintés ikon: A kvíz kérdéseinek előnézetét tudja megtekinteni a felhasználó.



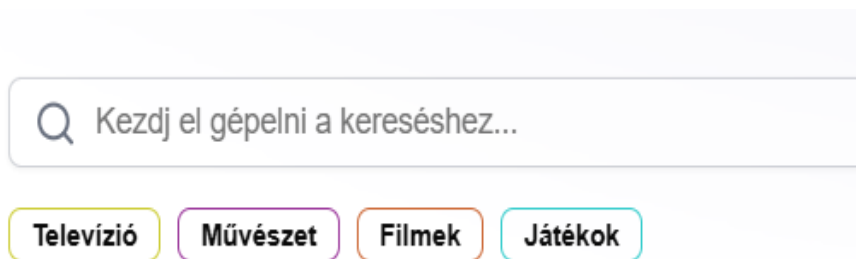
21. ábra - A kvíz kártyának leírása

Ebben a legördülő részben a kvizek állapotához képest lehet egy-egy szűrést létrehozni. A képen is látható, hogy lehet szűrni, hogy az összes, publikus, vagy csak a privát kvizek legyenek láthatóak. Ez akkor nagyon hasznos, amikor már egy profil több tíz kvízzel is rendelkezik, így könnyebb keresni a felhasználónak a saját kvizei között.



22. ábra - legördülő menü a kvizek keresési szűkítéséhez

A keresés ikonnál tud a felhasználó keresni és/vagy szűrni a kvizei között. Akár a Kvíz nevében szerepel egy szó, vagy a leírásában.

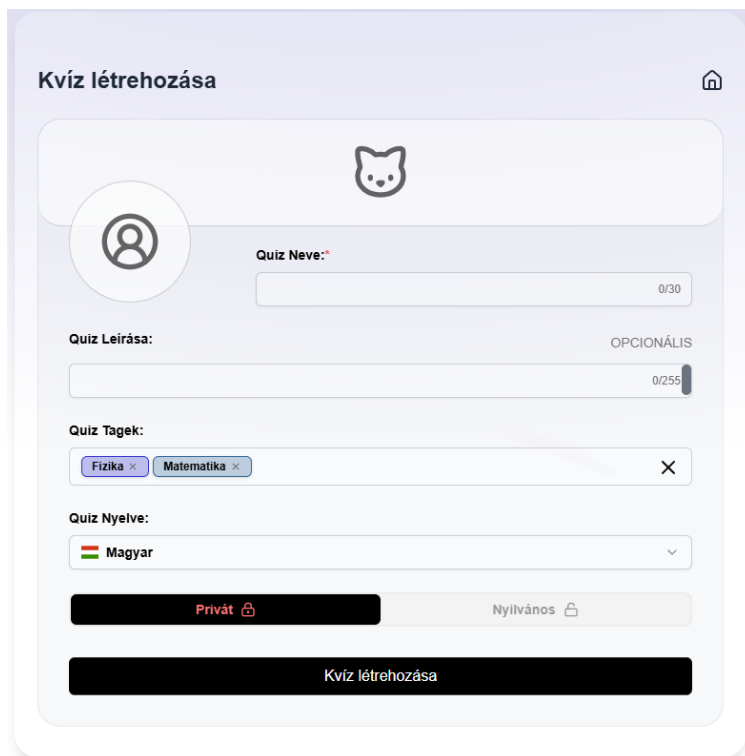


23. ábra - keresés a kvizeim között

Alatta jelennek meg azok a tagek, amivel már a felhasználó már hozott létre kvizeket. Ezekre kattintva szintén lehet szűkíteni a keresést, és könnyebben tájékozódhat a felhasználó (akié a profil), vagy aki a másnak a profilján szeretne böngészni.

A kvíz létrehozásának folyamata a következő lépésekből áll. Minden lépésnél megadható egy-egy fontos elem, ami elengedhetetlen a kvíz létrehozásához:

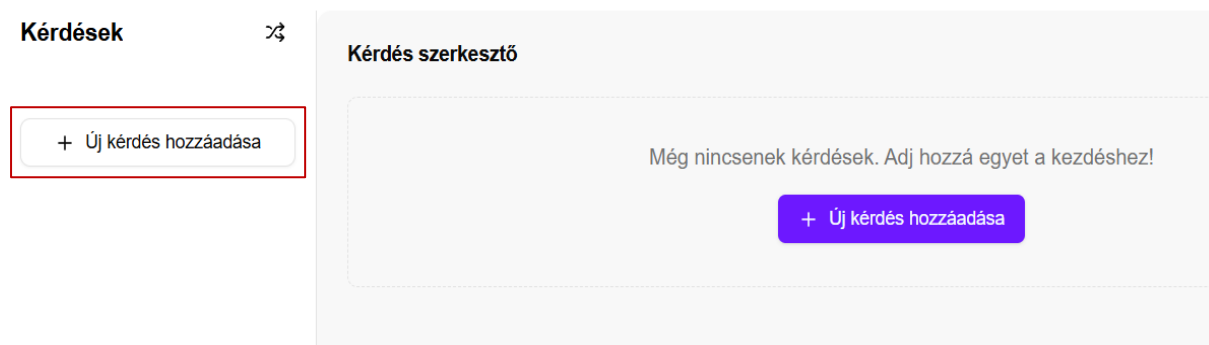
- Elsőként az úgynevezett „banner” -nek választható egy kép. Érdekes egy témához illőt keresni és azt feltölteni.
- Egy profilképet is kell beállítani a kvíznek. Segít egyedivé varázsolni a kvízedet.
- Természetesen az egyik legfontosabb lépés a kvíz neve. Érdekes már most kiválasztani, bár részletezni még nem szükséges. Fontos azonban, hogy ne térjen el a témától, mert az megtévesztő lehet a többiek számára.

The image shows a web form titled "Kvíz létrehozása" (Create Quiz). At the top, there is a cat icon and a home icon. Below this is a circular profile picture placeholder. The form contains several input fields: "Quiz Neve:" (Quiz Name) with a character count of 0/30, "Quiz Leírása:" (Quiz Description) with a character count of 0/255 and a label "OPCIONÁLIS" (Optional), and "Quiz Tagek:" (Quiz Tags) with two tags "Fizika" and "Matematika" and a close button. There is also a "Quiz Nyelve:" (Quiz Language) dropdown menu showing "Magyar" (Hungarian). At the bottom, there are two buttons: "Privát" (Private) and "Nyilvános" (Public), and a large black button labeled "Kvíz létrehozása" (Create Quiz).

24. ábra - kvíz létrehozása

- A leírása a kvíznek opcionális, ennek ellenére ajánlott írni, már csak amiatt is, mert kedvcsinálónak nagyon jó és esetlegesen a kifejtést itt meglehet tenni.
- A tagek kiválasztása szintén az egyik legfontosabb rész. Egy taget csak egyszer tudsz felhasználni, de később a keresésben ugyanúgy segíthet, mint például a kvíz neve.
- A privát vagy nyilvános opció elég egyértelmű. Ha privát csak te láthatod és használhatod, viszont, ha publikus, bárki a világban játszhat vele.
- Végül a legalsó gombra kattintva elmenthető az, amit eddig csinált a felhasználó. Ezután dob át a létrehozott kvíznek a szerkesztésére.

Ezen az oldalon van lehetőség a kvíz kérdéseidet létrehozni, hozzáadni és szerkeszteni őket. Ha a kérdéseknél bármilyen változtatás történt azonnal felugrik egy „mentés” gomb, amivel a szerkesztéseket van lehetőség elmenteni.



25. ábra - kvíz szerkesztő, kérdések, válaszok

A(z) „+ Új kérdés hozzáadása” -ra kattintva, négy úgynevezett kérdéslapból tud a felhasználó válogatni. Mindegyik kérdéslapnál van lehetőség még az időtartam beállítására, illetve a helyes válasz bejelölésére.

1. A feleletválasztós talán a legismertebb az emberek körében. A legkevesebb a válaszlehetőségeknél a kettő, 2. míg a legtöbb 6 darab, amit hozzá lehet adni.

2. Az Igaz – Hamis lapnál csak a helyes választ lehet kiválasztani, a válaszadás az adott ennél a kérdéslapnál.

3. A felmérést azért hoztuk létre, hogy bármilyen kutatómunkához felhasználható legyen – például egy munkatárs, osztályfőnök is alkalmazhatja a

figyelemfelmérésre. Nem kizárólag játékok lejátszására szolgál, és a játékosok nem kapnak érte pontot.

4. Az infó lap, általános tájékoztatásra, úgynevezett „fun fact” -re használható, vagy csak a játékosok biztatására.



26. ábra - a négy slide lehetőség

Kérdés szerkesztő

Kérdés Quiz

❓ Mi a jelentése annak, hogy "Nem minden arany, ami fénylik"?

Időtartam

⊖ 15 ⊕

Válasz lehetőségek: 🔗 Válaszok összekeverés + Válaszlehetőség hozzáadása

 Csak az arany a valódi érték.

☐ Helyes válasz-e?

 A fényes dolgok mindig a legjobbak.A fényes dolgok mindig a legjobbak.

☐ Helyes válasz-e?

 Mindig csak a fényes dolgokat kell keresni, mert azok a legértékesebbek.

☐ Helyes válasz-e?

 A dolgok értékét nem mindig könnyű felismerni.

☒ Helyes válasz-e?

28. ábra - Feleletválasztós kérdés szerkesztő

Kérdés szerkesztő

Kérdés Igaz/Hamis

❓ "Aki másnak vermet ás, maga esik bele" azt jelenti, hogy aki másokat próbál becsapni, végül ő maga lesz az, aki bajba kerül.

Időtartam

⊖ 10 ⊕

Válasz lehetőségek:

 Igaz

☒ Helyes válasz-e?

 Hamis

☐ Helyes válasz-e?

27. ábra - igaz - Hamis kérdés szerkesztő

Kérdés szerkesztő

Kérdés Infó Slide

❓ Hogyan változott a mondás jelentése? --> "Ne ítélj könyvet a borítójáról."

Időtartam

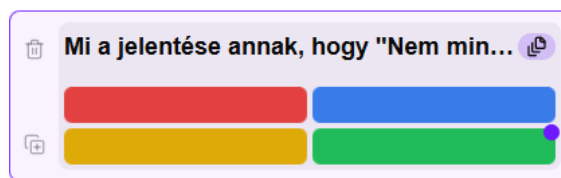
⊖ 15 ⊕

Tartalom

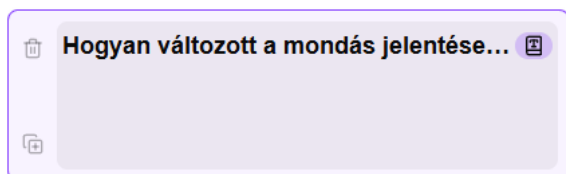
A mondás eredetileg arra figyelmeztet, hogy ne ítéljünk meg valamit csupán a külsőségek alapján. Azóta inkább a személyek megítélésére is használják, figyelmeztetve arra, hogy ne hozzunk előítéleteket másokról.

29. ábra - Infó kérdés szerkesztő

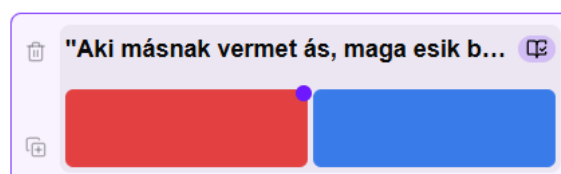
A kérdéslapoknál lévő kis kártya, ami jelzi, hogy melyik lapot választotta a felhasználó, hány darab válaszlehetőség van (a feleletválasztósnál), illetve a lila kis pötty jelzi a készítőnek, hogy melyik választ jelölte be helyesnek.



31. ábra - feleletválasztós kártya



32. ábra - infó kártya



30. ábra - Igaz - Hamis kártya

4. További fejlesztési lehetőségek

4.1. Napi/Heti kihívások

Tervben van napi és heti kihívások bevezetése, amelyek rövid ideig elérhető, különleges kvizeket kínálnának a felhasználóknak. Ezekkel szeretnénk ösztönözni a rendszeres visszatérést, miközben extra pontszerzési és versengési lehetőségeket biztosítunk azoknak, akik szeretik a kihívásokat.

4.2. Közöségi funkciók

Szeretnénk lehetőséget adni a felhasználóknak arra, hogy értékelhessék és megjegyzéseket adhassanak a kvizekhez, ezzel is ösztönözve a visszajelzések és párbeszéd kialakulását. Emellett egy követési rendszert is beleépítenénk, ami segítené a felhasználókat abban, hogy könnyen nyomon követhessék kedvenc kvíz készítőik vagy ismerőseik tevékenységét.

4.3. Mobilapp

A jövőben szeretnénk elkészíteni egy mobilalkalmazást is, amely még kényelmesebbé tenné a kvizek elérését, akár útközben is. Az alkalmazás lehetőséget adna akár az offline használatra is, valamint push értesítéseken keresztül segítené abban, hogy a felhasználók ne maradjanak le az új kihívásokról vagy eseményekről. Ez lenne a legnagyobb volumenű fejlesztés a tervek között, így egyelőre távolabbi célként szerepel, amely külön erőforrásokat és alaposabb előkészítést igényel.

Ábrajegyzék

1. ábra - A Figma-ban készült alap design ötlet	8
2. ábra - Web struktúrája, felépítése	10
3. ábra - Az adatbázis meghívásához használt kód	10
4. ábra - A users tábla sémája	11
5. ábra - Kapcsolat a users és a quiz tábla között.....	13
6. ábra - Az adatbázis EER diagramja.....	13
7. ábra - A sötét / világos téma algoritmus	14
8. ábra - Kvíz szerkesztésének algoritmus	15
9. ábra - átalakítás a megfelelő formátumra.....	16
10. ábra - Kvíz lejátszásának valós idejű algoritmus.....	17
11. ábra - Az indítás után kapott felület.....	18
12. ábra - Példa, hogy hogyan néz ki egy tábla szerkesztése	19
13. ábra - Bejelentkezés és a világos/sötét téma állítása.....	19
14. ábra - A kvíz előnézetének megtekintése.....	20
15. ábra - A várószoba csatlakozásnál	20
16. ábra - Házigazda mód a kvíz lejátszásánál.....	21
17. ábra - Kérdés a játékosoknál	21
18. ábra - Bejelentkezés után található három opció	22
19. ábra - A játék végén lévő pontszámok és pódium.....	22
20. ábra - A profil oldal előnézete	23
21. ábra - A kvíz kártyának leírása	23
22. ábra - legördülő menü a kvizek keresési szűkítéséhez.....	24
23. ábra - keresés a kvizeim között	24
24. ábra - kvíz létrehozása.....	25
25. ábra - kvíz szerkesztő, kérdések, válaszok.....	26
26. ábra - a négy slide lehetőség	26
27. ábra - igaz - Hamis kérdés szerkesztő.....	27
28. ábra - Feleletválasztós kérdés szerkesztő	27
29. ábra - Infó kérdés szerkesztő.....	27
30. ábra - Igaz - Hamis kártya	28
31. ábra - feleletválasztós kártya.....	28
32. ábra - info kártya	28