Pollák Könyvklub

A Pollák Könyvklub Teljes Dokumentációja

Készítette:
László Csongor
Zsigó Dávid

Tartalomjegyzék

Bevezetés

Tézis

Napjainkban a könyvolvasás és – ebből kiindulva – a könyvklubok sajnos a háttérbe szorultak. Ami egykor a diákság és fiatalság közkedvelt elfoglaltsága volt, az manapság már ritkaság; helyettesítette a közösségi média. Ezek a tények inspiráltak minket arra, hogy létrehozzunk egy online Pollák Antal Könyvklubot, amely egy olyan „közösségi platform”, amely arra ösztönzi a felhasználóit, hogy új könyveket / sorozatokat / filmeket éljenek át, majd beszéljék át a többiekkel. Abban reménykedünk, hogy ez az oldal képes lesz több embert a könyvolvasásra motiválni, és tudunk a felhasználóinknak egy friss perspektívát ajánlani a kedvenc műveikről, miközben új barátságokat alakítanak ki diáktársaikkal.

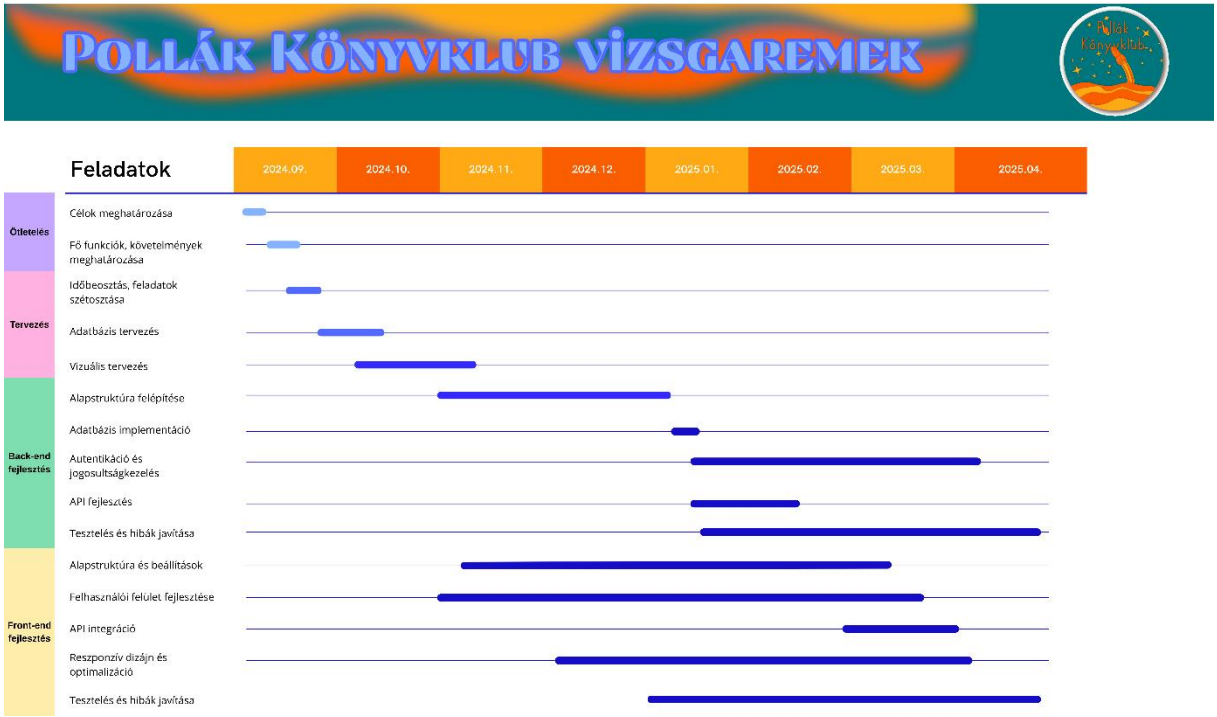
Célunk

A célunk a Pollák Könyvklubbal, ahogyan ezt már a tézisben említettük, hogy ösztönözzük a Pollák Antal Technikum tanulóit arra, hogy a mélyebb kapcsolatot építsenek ki azokkal a művészi alkotásokkal, amelyeket elolvasnak vagy megnéznek. Iskolánk kiváló irodalomoktatással rendelkezik, amely sokunkkal újra megszerettette a fikciónális történetek mélyebb, részletesebb elemzését. Mi ennek a szeretetnek szeretnénk lehetőséget adni kibontakozni. A Pollák Könyvklub ösztönzi a felhasználóit (diáktársainkat és pedagógusainkat), hogy megosszák egymással véleményeiket különböző alkotásokról, és ezek alapján új barátságokat építsenek ki. A tervünk az, hogy a Pollák Könyvklub egy szerves részévé tud válni a Pollák sokszínű technológiai ökoszisztémájának a PollákMix és a Pollák Büfé mellett.

Fejlesztői Dokumentáció

Csapat

A csapat 2 tagból áll, **László Csongor**ból és **Zsigó Dávid**ból. A feladat egyenletesen volt megosztva, László Csongor feladata volt a projekt általános vezetése, tervek kiszabása, program tervezése és vizuális elemeinek elkészítésé, miközben Zsigó Dávid foglalkozott a projekt finomításával, különböző, kényesebb funkciók létrehozásával, a projekt korai, alapszintű megtervezésével, és a projektmunkát megkönnyítő kommunikációs csatornák létrehozásával. Ezek mellett közösen lettek elkészítve az eredeti látványtervek, es közösen lettek eltervezve a projekt alapvető funkciói. Az alábbiakban lehet látni a csoportunk által elvégzett munkát hónapok szerint egy Gantt diagramban.



Fejlesztőkörnyezet

Hardverek

A projekt elkészítésére használt hardverek **2 darab Windows 11 PC** és **2 darab Windows 10 PC** (iskolai számítógépek), illetve 1 darab **Steam Deck**, amely **Arch Linux**-ot futtat. A projektfeladat elkészült változata stabilan fut az összes előbb említett készüléken. A vizuális elemek elkészítésére használtunk egy **XP-PEN Artist 12 Pro** rajztablettet.

Kódfejlesztés

A projekt fejlesztése közben számos különböző szoftvert használtunk fel. A kód létrehozása és további frissítése exkluzívan **Visual Studio Code**-on belül történt. VS Code egy egyértelmű választás volt, tagjaink számára ez a legkényelmesebb és legismertebb IDE (Integrated Development Evironment Software) a piacon, úgyhogy ez a választás magától értetődő volt.

Adatbáziskezelés

Az adatbázisokat a **XAMPP CONTROL PANEL** (Cross-Platform, Apache, MySQL, PHP and Perl) segítségével futtattuk, azon belül is a MySQL-nek köszönhetően. Ezt legfőbbként a MySQL által nyújtott phpMyAdmin oldal egyszerű kezelhetősége miatt választottuk.

API teszelés

A backendhez elkészített Service és Controller fájlok kódjait utána a **Postman** segítségével teszteltük le. Ez segített megmutatni, hogy hol kellett még finomítani az API lekérdezéseken. Ez a program végső fázisára már elavulttá vált, hiszen a hitelesítési folyamat nem engedi át a Postman lekérdezéseket, viszont a projekt ezen fázisában már nem volt komolyabb szükségünk erre, hiszen minden endpointot részletesen leteszteltünk, mielőtt a hitelesítést integráltunk volna.

Vizuális dizájn

A projekt vizuális oldala kifejezetten sok figyelmet kapott. Extenzív finomításokon, újradolgozásokon és egyéb változtatásokon ment keresztül, mielőtt elérte a jelenlegi verzióját. Ennek a kinézetnek az eléréséhez rengeteg különböző szoftvert / szolgáltatást használtunk fel. A kettő legfőbb az **Adobe Photoshop** és az **Adobe Illustrator**. Az Adobe programokkal hoztuk létre a UI / UX különböző részeit, illetve a jelenlegi stílusirányzat mindegyik elemét. Illustrator-t használtuk a vektorgrafikus elemek elkészítésére, majd ezeket Photoshop-on belül raszterizálás után kiegészítettük hiányzó részletekkel, hogy megadjuk neki a DIY kinézetet, amely az egész projektre jellemző. Ezeken kívül a korai dizájnok és helytartók elkészítésére alkalmaztuk a **GIMP**-et, amely egy iskolai környezetben könnyen elérhető alternatívája a lényegesen erősebb Adobe szoftvereknek. Ennek ellenére számos előrelépést tettünk a GIMP segítségének köszönhetően. Egyéb minimális vázlattervek kirajzolására emellett még felhasználtuk a **Clip Studio Paint 3.0** változatát egy XP-PEN Artist 12 Pro rajztableten. A korai látványtervek **Canva** használatával születtek. Fontos megemlíteni, hogy a projekthez 0 AI képgenerálás lett felhasználva, hiszen az teljes mértékben szembe menne a csoport értékrendjeivel.

Dokumentáció

A dokumentáció és prezentáció a **Microsoft Office Word** és **PowerPoint** programjaival készült el, mivel ezek a piacvezető szoftverek ezen a téren, amelyeket csapatunk mindkét tagja mélyen ismer. Az ER (Entity Relations) és Gantt diagramok **Draw.io** használatával készült el.

Munkamegosztás

A munkamegosztás több platformon történt. Az effektív kódírás érdekében a kódot egy **GitHub** repository-ba lettek feltöltve, ahol a csoport mindkét tagja hozzá tudott járulni a kód fejlesztéséhez. A projekt pontosabb megtervezése **Discord**on és **Messenger**en keresztül történt, majd ezek a terveket **Trello**-val lettek nyomon tartva.

Fejlesztési technológiák

A Pollák Könyvklub létrehozásához modern webes technológiák kombinációját alkalmaztuk, amelyek lehetővé teszik a hatékony fejlesztést és megbízható működést.

A felhasználói felület a Vue.js JavaScript keretrendszerre épül, amely komponens-alapú architektúrájával rugalmas és reaktív felhasználói élményt biztosít. A navigációt a Vue Router modul valósítja meg, amely lehetővé teszi a zökkenőmentes átjárást az oldal különböző részei között anélkül, hogy a teljes oldalt újra kellene tölteni. Az API kéréseket az Axios könyvtár segítségével kezeljük, biztosítva a megbízható kommunikációt a frontend és a backend között.

A háttérrendszer Node.js alapú, amely az Express.js webalkalmazás-keretrendszert használja az API végpontok létrehozásához és kezeléséhez. Az adatok tárolását SQL adatbázis biztosítja, amelyet a Prisma ORM segítségével érünk el - ez jelentősen leegyszerűsíti az adatbázis-műveletek kezelését és biztosítja a típusbiztonságot.

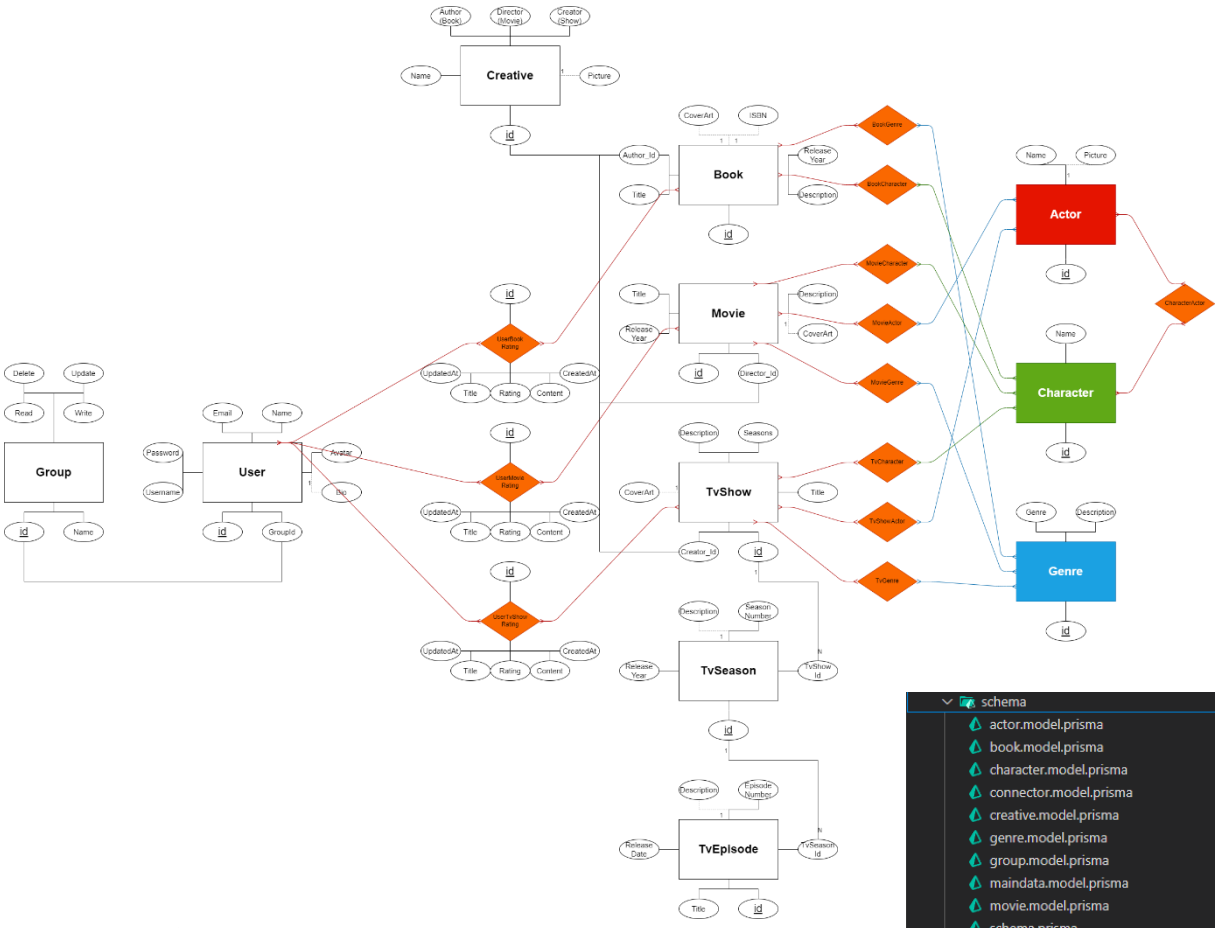
A felhasználói hitelesítést JWT (JSON Web Tokens) technológiával valósítottuk meg, amely biztonságos módszert kínál a felhasználói munkamenetek kezelésére. A jelszavak tárolását a bcrypt könyvtár segítségével biztonságosan, hashelve végezzük. Az express-session middleware gondoskodik a felhasználói munkamenetek kezeléséről.

A fájlkezelést a Multer könyvtár biztosítja, amely lehetővé teszi a profilképek és médiaanyagok feltöltését. Az automatikus email-küldéseket a Nodemailer modul valósítja meg, amely mind a regisztrációs megerősítéseket, mind az elfelejtett jelszavakkal kapcsolatos értesítéseket kezeli.

A rendszer környezeti változók kezelésére a dotenv csomagot használja, amely lehetővé teszi a bizalmas adatok (például API kulcsok és adatbázis-hitelesítési adatok) biztonságos tárolását. A kereszthivatkozásokat a CORS middleware kezeli, míg a szerveroldali rendereléshez az EJS sablonmotort alkalmazzuk.

Adatszerkezet

A Pollák Könyvklub egy összetett, részletes adatbázisrendszert használ fel, amely az alábbi relációs diagramban tekinthető meg:



Ezeket a táblákat **Prisma** segítségével generáltuk le.

Az adatbázisok kapcsolata eleinte nehezen értelmezhető, ezért alábbiakban az adatbázistáblák struktúrájának részletesebb bemutatása következik.

Book | Movie | TvShow

A Book, Movie és TvShow táblák a legfontosabb részei a projektnek. Ezek tartalmaznak minden információt a könyvekről, filmekről és sorozatokról, ebbe beleértve a borítókat, a leírásokat, címeket és minden egyéb információt.

Actor | Character | Genre

Az Actor, Character és Genre táblák mind hasonló célt töltenek be. Olyan adatokat tárolnak, amelyeket a **Book**, **Movie** és **TvShow** táblák felhasználnak. A Genre tábla tárolja el a műfajok listáját, a Character tábla a karakterek listáját, és az Actor tábla a színészekét. Ezek utána következő művekhez vannak rendelve az alábbi Many-to-many relációs táblák által:

Színészek:

- **MovieActor** (Összeköti a filmet egy színésszel)

- **TvShowActor** (Összeköti a sorozatot egy színésszel)
- **CharacterActor** (Összeköti a karaktert egy színésszel)

Karakterek:

- **BookCharacter** (Összeköti a könyvet egy karakterrel)
- **MovieCharacter** (Összeköti a filmet egy karakterrel)
- **TvCharacter** (Összeköti a sorozatot egy karakterrel)

Műfajok:

- **BookGenre** (Összeköti a könyvet egy műfajjal)
- **MovieGenre** (Összeköti a filmet egy műfajjal)
- **TvGenre** (Összeköti a sorozatot egy műfajjal)

Ezek a táblák Many-to-many relációt használnak, mivel egy műnek lehet több műfaja, szereplője vagy színésze.

Creative

A Creative tábla tárolja az adatait a különböző íróknak, rendezőknek és sorozatkészítőknek. A tábla Many-to-one kapcsolattal kötődik a **Book**, **Movie** és **TvShow** táblákhoz. Emellett tartalmaz egy-egy boolean-t, amely eldönti, hogy az ember egy író, rendező illetve sorozatkészítő e.

User

A User tábla tárolja az adatbázisunkban a felhasználó információit. Ez a tábla az egyik legközpontibb eleme a projektnek. Tárolja a felhasználó nevét, E-mail címét, titkosított jelszavát, profilképét, profilleírását és hitelesítő funkcióit. Emellett csatlakozik a **Group** táblához, amely denominálja, hogy a USER vagy ADMIN csoporthoz tartozik a felhasználó. Ezen túl össze van még kötve a **UserBookRating**, **UserMovieRating** és **UserTvShowRating** adatbázisoknak, amelyek tartalmazzák egy megadott értékelés adatait, illetve a felhasználó és az értékelt mű azonosítóit.

Az alábbi szakaszban demonstrációk következnek a kód különböző részeiből, majd egy magyarázat a működésükről.

Kódrészletek

Login háttér

A könyvklub hitelesítési felületén (bejelentkezés és regisztráció) dinamikus hátteret alkalmaztunk, amely folyamatosan mozog és periodikusan más képre vált. Ezt a hatást CSS keyframe animációkkal értük el.

```vue
<div class="background-slideshow">
  <div
    v-for="(image, index) in backgroundImages"
    :key="index"
    class="background-image"
    :class="{
      active: currentImageIndex === index,
      'pan-top-left':
        panDirections[index % panDirections.length] === 'top-left',
      'pan-top-right':
        panDirections[index % panDirections.length] === 'top-right',
      'pan-bottom-left':
        panDirections[index % panDirections.length] === 'bottom-left',
      'pan-bottom-right':
        panDirections[index % panDirections.length] === 'bottom-right',
    }"
    :style="{
      backgroundImage: `url(${image})`,
      animationDelay: `-${(index * 6) % 60}s`,
    }"
  ></div>
</div>

<div class="login-form-container">
  <div class="logo-container">
    <img
      src="/src/assets/images/text_logo.png"
      alt="Pollák Könyvklub"
      class="text-logo"
    />
  </div>

  <div class="login-form">

    <slot></slot>
  </div>
</div>
</div>
</template>

<script setup>
import { ref, onMounted, onUnmounted } from "vue";

const backgroundImages = [
  "/src/assets/images/home/BB.png",
  "/src/assets/images/home/BCS.png",
  "/src/assets/images/home/CSM.png",
  "/src/assets/images/home/GOT.png",
  "/src/assets/images/home/HP.png",
  "/src/assets/images/home/INDY.png",
  "/src/assets/images/home/JJBA.png",
  "/src/assets/images/home/M&M.png",
  "/src/assets/images/home/SW.png",
];

const panDirections = ["top-left", "top-right", "bottom-left", "bottom-right"];
const currentImageIndex = ref(0);
let slideInterval;

const changeBackgroundImage = () => {
  currentImageIndex.value =
    (currentImageIndex.value + 1) % backgroundImages.length;
};
```

Az AuthLayout komponens biztosítja ezt a vizuális élményt kilenc különböző háttérkép segítségével, amelyek népszerű filmekből és sorozatokból származnak. A rendszer 15 másodpercenként automatikusan vált a képek között finom átmenettel. Minden háttérkép lassú, folyamatos pásztázó mozgást végez négy lehetséges irányban, ami növeli a felület vizuális érdekességét.

A komponens réteges felépítésű: az animált háttér fölött helyezkedik el a könyvklub logója, majd erre épül a bejelentkezési vagy regisztrációs űrlap. Az űrlap konténere félig átlátszó, mintázott háttérrel rendelkezik, amely biztosítja a szövegek olvashatóságát, miközben látni

engedi a háttéranimációt. A komponens rugalmasan befogadja a hitelesítési oldalak (bejelentkezés, regisztráció) tartalmát, és egységes stílust biztosít számukra. A beágyazott elemek - címsorok, űrlap mezők, gombok és üzenetek - következetes megjelenést kapnak, ami hozzájárul a könyvklub felhasználói felületének egységes arculatához.

```css
@keyframes pan-top-left {
  0% {
    transform: scale(1.5) translate(8%, 5%);
  }
  100% {
    transform: scale(1.5) translate(-8%, -5%);
  }
}

@keyframes pan-top-right {
  0% {
    transform: scale(1.5) translate(-8%, 5%);
  }
  100% {
    transform: scale(1.5) translate(8%, -5%);
  }
}

@keyframes pan-bottom-left {
  0% {
    transform: scale(1.5) translate(8%, -5%);
  }
  100% {
    transform: scale(1.5) translate(-8%, 5%);
  }
}

@keyframes pan-bottom-right {
  0% {
    transform: scale(1.5) translate(-8%, -5%);
  }
  100% {
    transform: scale(1.5) translate(8%, 5%);
  }
}
```

Emailküldés

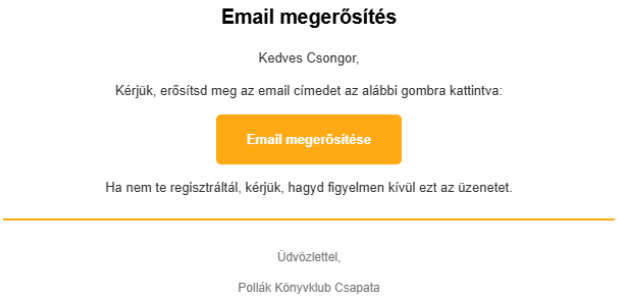
A regisztrációs és jelszókezelési folyamatokhoz a rendszer automatikus email-küldési funkcióval rendelkezik. Az EmailSender szolgáltatás a Node.js Nodemailer csomagját használja és Gmail SMTP szerveren keresztül kommunikál. Erre egy külön e-mail címet hoztunk létre, amelynek címe pollak.bookclub@gmail.com.

```
import nodemailer from "nodemailer";
import dotenv from "dotenv";

dotenv.config();

export const transporter = nodemailer.createTransport({
    host: "smtp.gmail.com",
    port: 587,
    secure: false,
    auth: {
        user: process.env.EMAIL_USER,
        pass: process.env.EMAIL_PASS,
    },
});
```

```
EMAIL_USER=pollak.bookclub@gmail.com
EMAIL_PASS=acoz chud fxei ymec
```

A szolgáltatás két fő funkciót valósít meg:

Email cím megerősítés:



Amikor egy felhasználó regisztrál, a rendszer automatikusan küld egy email-megerősítő üzenetet a megadott címre. Ez az email tartalmazza a felhasználó nevét és egy egyedi megerősítési linket. A felhasználói élmény növelése érdekében az email formázott HTML tartalommal rendelkezik, beleértve a Pollák Könyvklub logóját és a karakterisztikus narancssárga színű megerősítési gombot. Az email tájékoztatja a felhasználót, hogy ha nem ő regisztrált, hagyja figyelmen kívül az üzenetet.

Elfelejtett jelszó kezelése:

A szolgáltatás lehetővé teszi jelszó-visszaállító emailek küldését. Ezek az üzenetek zöld színű jelszó-visszaállító gombot tartalmaznak, és tájékoztatják a felhasználót, hogy a link egy órán keresztül érvényes.

Értékelések

A Pollák Könyvklub projekt egyik különlegessége a típusfüggetlen értékelőrendszer, amely lehetővé teszi, hogy a felhasználók egységes felhasználói élményt kapjanak, miközben teljesen különböző médiaelemeket (könyvek, filmek, sorozatok) értékelnek.

Az értékelőrendszer ötletes megvalósítása egy közös frontend komponensen alapszik (**MediaRating.vue**), amely típusparaméter alapján dönti el, melyik háttérszolgáltatást kell használnia. Ez a módszer jelentősen csökkenti a kódismétlést, miközben lehetővé teszi, hogy minden médiatípus megtartsa a saját egyedi tulajdonságait és adatmodelljét.

A megoldás központi eleme az útvonal (**route**) paraméterek kihasználása:

```
const getEndpoint = () => {
  const typeMap = {
    'books': 'book',
    'movies': 'movie',
    'series': 'tvshow'
  };
  return typeMap[props.type];
};
```

Amikor egy felhasználó egy médiaelem részleteit megtekinti, a rendszer automatikusan azonosítja a médiatípust az URL-ből, és a megfelelő értékelési API végpontot hívja meg. Bár a háttérben három külön adattáblát használ (**UserBookRating**, **UserMovieRating**, **UserTVShowRating**), a felhasználó szemszögéből ez teljesen transzparens.

Az értékelések lekérésekor az API azonnal visszaadja az aktuális felhasználó értékelését (ha létezik), így a rendszer képes előre kitölteni az értékelési űrlapot. A háttérszolgáltatások a három különböző értékelési típust egységes formátumra hozzák, ami lehetővé teszi a közös megjelenítést és kezelést, miközben a különböző médiatípusok sajátosságai megmaradnak az adatmodellben.

Ez a megközelítés különösen hasznos az alkalmazás bővíthetősége szempontjából - új médiatípusok hozzáadásakor csak az adatmodellt és a hozzá tartozó API végpontot kell létrehozni, miközben a felhasználói felület minimális módosítással újrahasználható.

Fájlfeltöltés

A Pollák Könyvklub projektben a különböző médiatartalmak megjelenítéséhez elengedhetetlen a képfájlok kezelése. Legyen szó felhasználói profilképekről, könyvborítókról, filmplakátokról vagy alkotók fotóiról, minden esetben egy jól strukturált és biztonságos fájlfeltöltési rendszerre van szükség. A projekt ezt a funkcionalitást a Node.js Multer middleware segítségével valósítja meg, amely kifejezetten a multipart/form-data típusú kérések hatékony kezelésére lett tervezve.

Multer Alkalmazása a Projektben

A Pollák Könyvklub backend rendszerében a Multer konfigurálása az egyes fájltípusoknak megfelelően történik. A felhasználói profilképek esetében például a rendszer az auth.controller.js fájlban definiálja a tárolási stratégiát, amely meghatározza, hogy a feltöltött képek az uploads/profile mappában kerülnek elhelyezésre, egyedi fájlnévvel, amely időbélyeget és egy véletlenszerű számot tartalmaz a nevek ütközésének elkerülése érdekében.

A profilképek feltöltésénél a rendszer szigorú validációs szabályokat alkalmaz: csak a megfelelő formátumú képfájlok (JPEG, PNG, GIF) fogadhatók el, és a maximális méretük 2 MB lehet. Ez a korlátozás biztosítja, hogy a szerver nem terhelhető túl nagy méretű fájlokkal, valamint a felhasználói élmény is megfelelő sebességű marad.

Fájlfeltöltési Folyamat a Háttérrendszerben

A fájlfeltöltési folyamat a backend oldalon több lépésből áll. Amikor egy felhasználó profilképet szeretne frissíteni, az Express router a következő mintát követi:

```
router.put("/profile", uploadAvatar.single('avatar'), async (req, res) => {
    try {
        const authHeader = req.headers.authorization;
        if (!authHeader || !authHeader.startsWith('Bearer ')) {
            return res.status(401).json({ message: "Authorization header required" });
        }

        const token = authHeader.split(' ')[1];

        const data = await prisma.maindata.findFirst();
        if (!data) {
            return res.status(500).json({ message: "Server configuration error" });
        }

        const decoded = jwt.verify(token, data.JWTSecret, {
            algorithm: data.JWTAlgorithm || 'HS256'
        });

        const userId = decoded.sub || decoded.id;

        const { name, email, password } = req.body;

        const updateData = {};

        if (name) updateData.name = name;
        if (email) updateData.email = email;

        if (req.file) {
            updateData.avatar = `/uploads/profile/${req.file.filename}`;
        }

        if (password) {
            updateData.password = await bcrypt.hash(password, 10);
        }
            You, 2 weeks ago • KB jo …
        const updatedUser = await prisma.user.update({
            where: { id: userId },
            data: updateData
        });

        res.status(200).json({
            message: "Profile updated successfully",
            user: {
                id: updatedUser.id,
                name: updatedUser.name,
                email: updatedUser.email,
                avatar: updatedUser.avatar
            }
        });
    } catch (error) {
        console.error("Error updating profile:", error);
        if (error.name === 'JsonWebTokenError') {
            return res.status(401).json({ message: "Invalid token" });
        }
        res.status(500).json({ message: "Failed to update profile" });
    }
});
```

Tesztdokumentáció

A backend és frontend teszteléséhez a **JEST** és **PlayWright**-ot használtuk fel.

Backend

A Pollák Könyvklub projekt backend részének tesztelése gondosan megtervezett folyamat, amely biztosítja az alkalmazás megbízható működését és a szolgáltatások megfelelő funkcionalitását. A backend tesztek a **services** könyvtárstruktúrában helyezkednek el, ahol minden szolgáltatás saját tesztfájljával rendelkezik. A tesztfájlok elnevezése következetes mintát követ: a szolgáltatás neve után a .test.js kiterjesztés szerepel.

Minden főbb funkcionalitás le van takarva tesztekkel.

A rendszer gerincét adó könyvekkel kapcsolatos szolgáltatásokat a book.service.test.js fájl teszteli, amely átfogó módon ellenőrzi az összes alapvető művelet helyes működését. Ezek közé tartozik az új könyvek létrehozása, az összes könyv lekérdezése, egy adott könyv azonosító alapján történő keresése, valamint az egy adott szerzőhöz tartozó könyvek listázása. A teszt továbbá kiterjed a könyvek adatainak frissítésére és a könyvek törlésének funkciójára is.

```
describe("createBook", () => {
  it("should create a new book", async () => {
    const testAuthor = await createCreative({
      name: "Test Author",
      author_book: true,
      director_movie: false,
      creator_show: false,
    });
    createdAuthorIds.push(testAuthor.id);

    const data = {
      title: "Test Book",
      author_Id: testAuthor.id,
      releaseYear: 2021,
      description: "Testing createBook",
      coverArt: null,
    };
    const result = await createBook(data);
    createdBookIds.push(result.id);

    expect(result).toHaveProperty("id");
    expect(result.title).toBe(data.title);
  });
});

describe("getAllBooks", () => {
  it("should return an array of books", async () => {
    const books = await getAllBooks();
    expect(Array.isArray(books)).toBe(true);
  });
});
```

A tesztek futtatása során a rendszer ideiglenes adatokat hoz létre az adatbázisban, amelyek kizárólag a tesztekhez használatosak. Az adatok integritásának megőrzése érdekében minden teszt után ezek az ideiglenes adatok automatikusan törlődnek, amit az **afterEach hook** biztosít. Ez a megközelítés garantálja, hogy minden teszt tiszta környezetben futhasson, és az eredmények ne befolyásolják egymást.

```
Test Suites: 7 passed, 7 total
Tests:       35 passed, 35 total
Snapshots:   0 total
Time:        11.7 s
Ran all test suites.
```

A jelenlegi tesztlefedettség elsősorban a könyvekkel kapcsolatos funkcionalitásra összpontosít, de a jövőbeni fejlesztések során tervezzük a felhasználói hitelesítés, a film- és TV-sorozat szolgáltatások, valamint az értékelési rendszer tesztelésének bővítését is. A backend API végpontjainak integrációs tesztjei szintén a

fejlesztési roadmap részét képezik, amelyek biztosítják majd, hogy a teljes rendszer minden komponense megfelelően együttműködjön.

A Pollák Könyvklub backend tesztrendszere így biztosítja, hogy az alkalmazás robusztus maradjon a jövőbeni fejlesztések és változtatások során, valamint megbízhatóan szolgálja ki a felhasználói igényeket, amelyek a virtuális könyvklub zökkenőmentes működéséhez szükségesek.

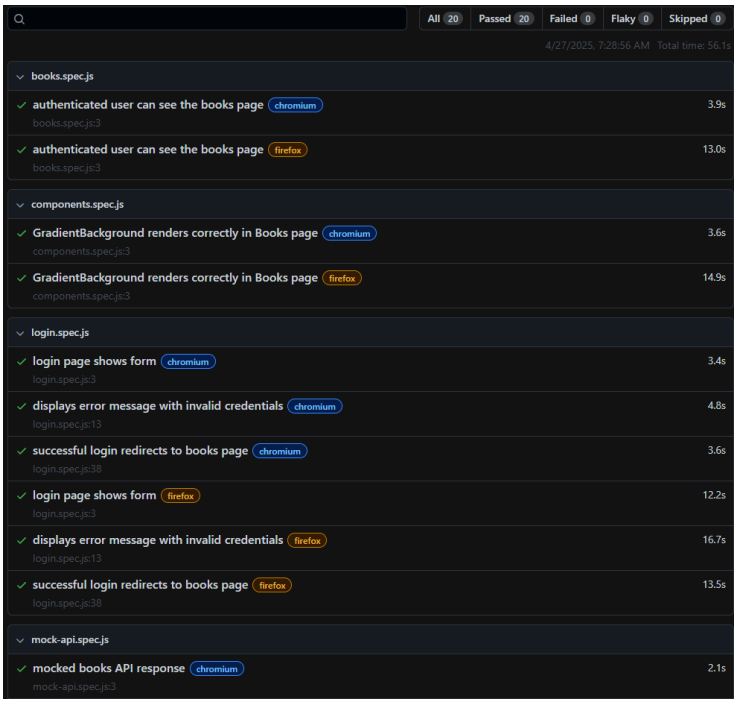
Frontend

```
import { test, expect } from './utils/auth.js';

test('authenticated user can see the books page', async ({ authenticatedPage: page }) => {
  await expect(page).toHaveURL(/.*\/books$/);

  await expect(page.getByRole('heading', { level: 1, name: 'Könyvek' })).toBeVisible({ timeout: 10000 });

  await expect(page.locator('.home-card')).toBeVisible({ timeout: 10000 });

  await expect(page.locator('.gradient-background')).toBeVisible();

  await expect(page.locator('.content-holder')).toBeVisible({ timeout: 10000 });
});
```

A Pollák Könyvklub weboldal frontend részének tesztelése a **PlayWright** eszközrendszerre támaszkodik. A frontend tesztek elsősorban a felhasználói felület működését és a különböző funkciók megbízhatóságát vizsgálják, biztosítva a rendszer stabilitását különböző böngészőkben.

Az end-to-end tesztek a valós felhasználói élményt utánozzák, lefedve a bejelentkezést, a tartalmak böngészését és a részletes adatok megtekintését. A tesztek fontos eleme az autentikáció kezelése, amely lehetővé teszi a védett oldalak és funkciók megbízható ellenőrzését. A segédmodulok gondoskodnak arról, hogy a tesztek következetesen szimuláljanak bejelentkezett állapotot.

A komponenstesztek az alkalmazás vizuális elemeinek működését ellenőrzik, mint például a háttér komponensek vagy az adatmegjelenítő kártyák helyes működését. Ezek a tesztek biztosítják, hogy a könyvek, filmek és sorozatok információi megfelelően jelenjenek meg, beleértve a címeket, szerzőket és egyéb adatokat.



A navigációs tesztek az oldalak közötti átjárhatóságot ellenőrzik, míg a felhasználói profil tesztjei a személyes adatok kezelését és frissítését vizsgálják. A tesztelési folyamat **Chromium** és **Firefox** alapú böngészőket használ, így biztosítva a keresztböngésző-kompatibilitást.

A frontend tesztrendszer átfogó célja, hogy a Pollák Könyvklub felhasználói felülete megbízható és

következetes maradjon, miközben támogatja a fejlesztőket az új funkciók bevezetésében és a meglévő funkciók fejlesztésében. A jól tervezett tesztek segítenek a hibák korai felismerésében és a magas minőség fenntartásában a projekt teljes életciklusa során.

Vizuális tervezés

A projekt készítése során lényeges figyelem lett az oldal kinézetébe fektetve. Rengeted dizájn iteráción esett át a Pollák Könyvklub, mielőtt eljutottunk a végső stílushoz.

1. Fázis

Az első fázis volt a dizájn legkorábbi verziója, ez az első hónapja a projektnek, amikor csak enyhe tervek voltak az elrendezésről. Ilyenkor egy zöldes stílussal volt megálmodva az oldal, bár azt már akkor is tudtuk, hogy ez csak egy ideiglenes kinézet lesz.

2. Fázis

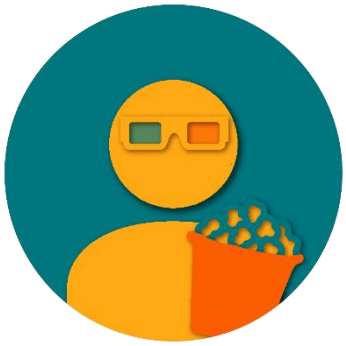
A második fázisban kezdett el kialakulni a projekt kék és narancs színkombinációja, bár ilyenkor még nem voltuniformis kinézet, csak a színek. Ennek ellenére már egyértelmű volt, hogy a stílus kezd egy szép formát felvenni

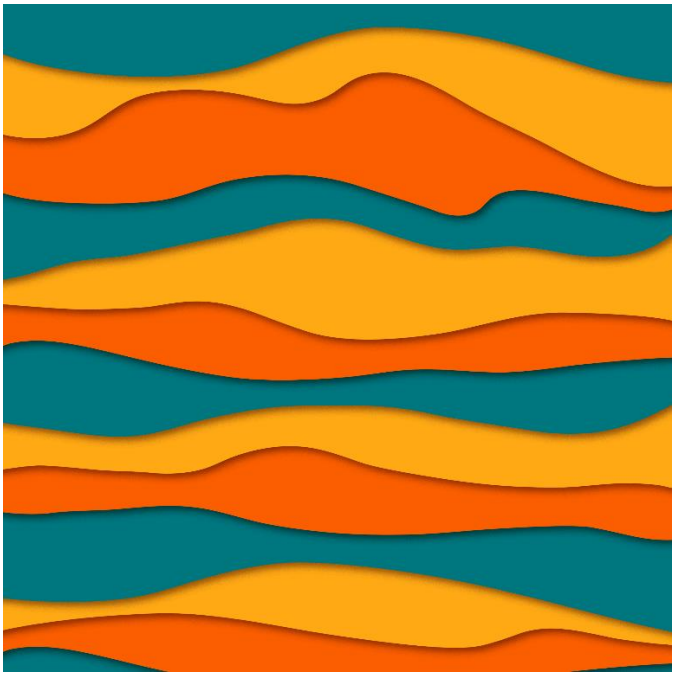
3. Fázis

A harmadik fázisban kezdtek megjelenni a jobban elkülönült, szolid színek, amelyek részben poszterizálással lettek elérve. Ennél a verziónál vált biztossá, hogy jó úton haladunk

4. Fázis

A negyedik dizájnfázis a végső, amely egy DIY / amatőr esztétikát használ fel. A projektünk a művészetek értékeléséről szól, úgyhogy egyértelmű volt egy kézműves téma bevezetése. A megjelenítési stílusra jellemző három kontrasztos szín, erős árnyékok, és egy többrétegű, ugyanakkor minimalista kinézet.





Pollák Könyvklub

Felhasználói dokumentáció

Felhasználói élmény

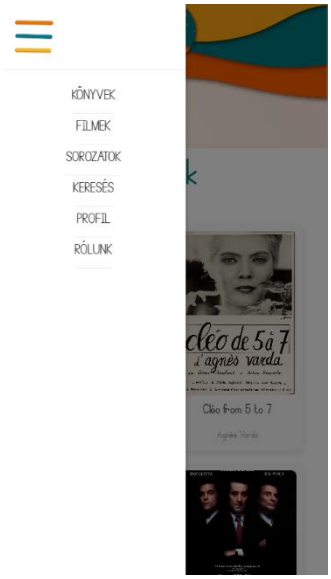
Sok munka lett belefektetve abba, hogy a Pollák Könyvklub egy kényelmes, könnyen értelmezhető felhasználói élményt nyújtson. Ez meglátszódik az oldal intelligens elrendezésében, egyedi vizuális stílusában.

Immerzív Vizuális Világ

A Pollák Könyvklub legszembetűnőbb felhasználói élménye a vizuális bevezetés, amely már a belépési felületen érezhető. A bejelentkezési és regisztrációs oldalon látható háttér egy kinematikus élményt nyújt, ahol a népszerű filmekből és sorozatokból származó képek lassú, panorámázó animációval váltakoznak 15 másodpercenként. A háttéranimációk négy különböző irányba mozognak, amely vizuális érdeklődést kelt, miközben a felhasználó figyelmét a középen elhelyezett, félig átlátszó bejelentkezési panelre irányítja. Ez a rétegzett megjelenés egyensúlyt teremt a látványos háttér és a funkcionalitás között, lehetővé téve a tartalom olvashatóságát az animáció megtekintése közben.

Adaptív Felhasználói Felület

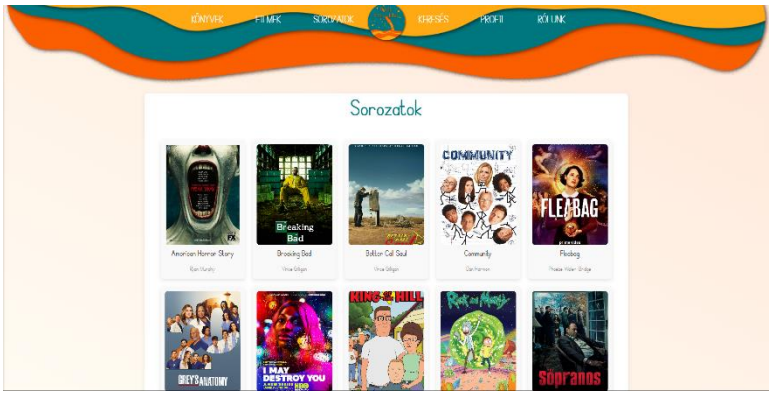
A könyvklub navigációs rendszere intelligensen alkalmazkodik a különböző eszközökhöz. Asztali nézetben a felhasználók egy elegáns, horizontális navigációs sávot látnak a fejléc két oldalán, amely szimmetrikusan helyezkedik el a központi logó körül. Mobileszközökön ez átalakul egy hamburger menüre, amely megnyitja az oldalsávot, ahol a navigációs elemek függőleges elrendezésben jelennek meg. A fejléc vizuális eleme, amely minden oldalon megtalálható, egy hullámzó, narancssárga háttérrel rendelkező banner, amely azonnal felismerhető és következetesen jelzi a felhasználónak, hogy a Pollák Könyvklub ökoszisztémán belül tartózkodik.



Intelligens Médiatartalom Megjelenítés

A tartalommegjelenítés egy rugalmas rácsos elrendezésben történik, amely automatikusan alkalmazkodik a képernyő méretéhez: kisebb képernyőkön 1-2 elem, míg nagyobb

felbontáson akár 5 elem is megjelenik egy sorban. A könyvek, filmek és sorozatok egységes kártyákon jelennek meg, amelyek tartalmazzák a borítóképet, címet és alkotó nevét. Ezek a kártyák interaktívak, enyhe nagyítási effektussal reagálnak az egér ráhúzására, vizuális visszajelzést adva a felhasználónak a kattintható elemekről. A borítóképek egységes méretűek, biztosítva az esztétikus elrendezést, miközben a hiányzó képek helyén egy elegáns helykitöltő kép jelenik meg.



Felhasználói Profilmegjelenítés

A profiloldal egy gondosan strukturált felhasználói felületet biztosít, ahol a felhasználó adatai és tevékenységei jól szervezett blokkokban jelennek meg. A profil fejlécében a felhasználó profilképe vagy monogramja látható egy kör alakú keretben, mellette a névvel és felhasználónévvel. A felhasználói statisztikák vizuálisan hangsúlyos, kattintható kártyákon jelennek meg, amelyek az értékelések és kedvencek számát mutatják. Ezek nemcsak információt közölnek, hanem interaktív navigációs pontként is szolgálnak a részletes értékelések és kedvencek oldalakra. A profil szerkesztése gomb csak a saját profilnál jelenik meg, míg mások profiljánál ez a funkció értelemszerűen elrejtésre kerül.

Szükséges hardver

A Pollák Könyvklub úgy lett tervezve, hogy akármilyen rendszeren működjön, de mivel a szervert lokálisan kell futtatni ebben a formájában a projektnek, egy számítógép kötelező, bár az operációs rendszer nem számít.

Telepítés és indítás

A Pollák Könyvklub projekt futtatásához több szoftverre és beállításra van szükség. Az alábbiakban részletesen bemutatjuk a telepítés és futtatás folyamatát.

A projekt használatához először szükséges telepíteni a Node.js rendszert (16-os vagy újabb verzió), amely mind a frontend, mind a backend futtatásához elengedhetetlen. Továbbá szükséges a XAMPP telepítése is, amely az adatbáziskezelést (MySQL) és a phpMyAdmin webes felületet biztosítja.

Az adatbázis beállításához először indítsuk el a **XAMPP Control Panelt**, majd aktiváljuk az **Apache** és **MySQL** szolgáltatásokat. Ezután nyissuk meg a **phpMyAdmin** felületét a böngészőben a **http://localhost/phpmyadmin/** címen, és hozzunk létre egy új adatbázist "vizsgaremek" néven. A projekt **Prisma** ORM-et használ az adatbáziskezeléshez, így az adattáblák létrehozása automatizált folyamat lesz.

Backend

- cd backend
- npx prisma generate
- npx prisma migrate dev
- node .

Frontend

- cd frontend
- npm run dev

Használat

Regisztráció és bejelentkezés

A Pollák Könyvklub használatához először regisztrálnia kell. A regisztráció során adja meg felhasználónevét, email címét, teljes nevét és jelszavát. Sikeres regisztráció után a megadott email címre érkezik egy megerősítő link, amelyre kattintva aktiválhatja fiókját. Bejelentkezéskor a felhasználónevet és jelszót kell megadnia. Az alkalmazás elmenti a bejelentkezési adatokat, így később automatikusan belépteti.

Tartalmak böngészése

A műveknél a részletes oldalon találja az értékelési funkciót. Itt:

- 1-5 csillaggal értékelheti a művet
- Megjelölheti kedvencként a szív ikonra kattintva
- Adhat címet az értékelésének
- Írhat szöveges véleményt a műről

Az értékeléseket és kedvenceket később megtalálja a profiloldalán is. Más felhasználók értékeléseit a mű részletes oldalán olvashatja, ami segíthet új, érdekes művek felfedezésében.

Profil kezelése

A **PROFIL** gombra kattintva érheti el személyes profiloldalát, ahol:

- Megtekintheti és szerkesztheti adatait (név, e-mail, jelszó)
- Feltölthet profilképet vagy generált monogramot használhat

- Láthatja értékeléseinek és kedvenceinek számát
- A számokra kattintva megtekintheti értékeléseinek és kedvenceinek listáját
- Kijelentkezhet a rendszerből

Más felhasználók profilját is megtekintheti, ahol láthatja az ő értékeléseiket és kedvenceiket, így felfedezhet hasonló érdeklődési körrel rendelkező diáktársakat.

Felhasználók keresése

A **KERESÉS** funkcióval megtalálhatja diáktársait a platformon:

- Kereshet név vagy felhasználónév alapján
- A találati listában láthatja a felhasználók profilképét vagy monogramját és szerepkörét
- A felhasználó kártyájára kattintva megtekintheti annak profilját

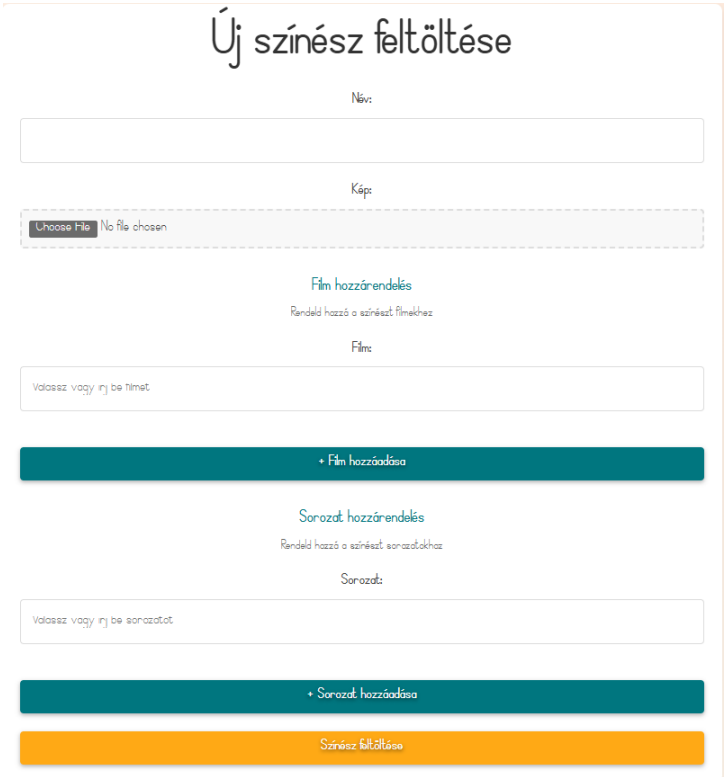
A keresési funkció segítségével felfedezheti, hogy diáktársai milyen könyveket, filmeket vagy sorozatokat kedvelnek, és így hasonló érdeklődési körű barátokra találhat a Pollák Antal Technikum közösségén belül.

Admin jogosultságok

Az adminisztrátoroknak joguk van különböző műveket feltölteni különböző műveket, színészeket és kreatívokat.

Ezek elérhetőek a következő linkeken:

- upload-actor
- upload-creative
- upload-movie
- upload-tvshow
- upload-book



Szakirodalom

Ebben a fejezetben vannak megemlítve a felhasznált szakirodalmi művek, amelyeket a projekt során felhasználtunk.

Multer

https://www.geeksforgeeks.org/multer-npm/

https://blog.logrocket.com/multer-nodejs-express-upload-file/

https://dev.to/codexam/how-to-use-multer-to-upload-files-in-nodejs-and-express-2a1a

https://expressjs.com/en/resources/middleware/multer.html

Prisma

https://www.prisma.io/docs/orm/prisma-schema

https://www.prisma.io/docs/orm/prisma-schema/data-model/models

https://medium.com/@imvinojanv/mastering-data-relationships-a-comprehensive-guide-to-building-prisma-schemas-99e1fe50a91d

Vite

https://vite.dev/guide/

https://dev.to/bcostaaa01/getting-started-with-vite-4fah

Axios

https://axios-http.com/docs/intro

https://www.geeksforgeeks.org/what-is-axios/

Adobe Photoshop

https://designshack.net/articles/software/how-to-vectorize-an-image-in-photoshop/

https://helpx.adobe.com/photoshop/user-guide.html

https://www.adobe.com/products/photoshop/how-to-use.html

Adobe Illustrator

https://www.creativebloq.com/digital-art/adobe-illustrator-tutorials-1232697

https://www.adobe.com/hu/learn/illustrator/web/ai-basics-fundamentals

MySQL

https://www.w3schools.com/mysql/

Összefoglalás

Összességében a Pollák Könyvklubba mind László Csongor és Zsigó Dávid számtalan órát fektetett. Mind a ketten abban reménykedünk, hogy a végeredményben látszódik a beletett rengeteg időnk, és az a gondoskodás, amire kifejezetten büszkék vagyunk.

A Pollák Könyvtár abból a közös szeretetből született, amit mindketten érzünk a művészet világa felé. A Pollák Antal Technikum tele van kiváló tanárokkal, bár sajnos sokszor tanításuk süket fülekre esik. Álmunk az, hogy ezt a helyzetet meg tudjuk változtatni azzal, hogy diáktársainkat ösztönözzük a művészettel való mélyebb interakcióra.

Büszkén mutatjuk be az elkészült Pollák Könyvklubot!

Sok szeretettel a csapatunktól, László Csongortól és Zsigó Dávidtól!