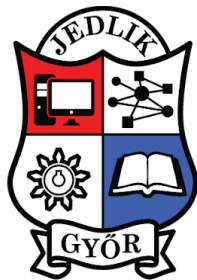




győri szakképzési centrum

Jedlik Ányos
Gépipari és Informatikai
Technikum és Kollégium



9021 Győr, Szent István út 7.

+36 (96) 529-480

+36 (96) 529-448

OM: 203037/003

jedlik@jedlik.eu

www.jedlik.eu

Záródolgozat feladatkiírás

Tanuló(k) neve: Hatos Zoltán, Erdős Dániel, Pápay Bálint
Képzés: Felnőttoktatás - esti munkarend
Szak: 5 0613 12 03 Szoftverfejlesztő és -tesztelő technikus

A záródolgozat címe:

Dör Községi Könyvtár

Konzulens: Kottra Richárd

Beadási határidő: 2025. 04. 15.

Győr, 2025. 04. 15.

Módos Gábor
igazgató

Konzultációs lap

	A konzultáció		Konzulens aláírása
	ideje	témája	
1.	2024.11.07.	Témaválasztás és specifikációk	
2.	2025.03.20.	Záródolgozat készültsége fokának értékelése	
3.	2025.04.15.	Dokumentáció véglegesítése	

Tulajdonosi nyilatkozat

Ez a dolgozat a saját munkánk eredménye. Dolgozatunk azon részeit, melyeket más szerzők munkájából vettünk át, egyértelműen megjelöltük.

Ha kiderülne, hogy ez a nyilatkozat valótlan, tudomásul vesszük, hogy a szakmai vizsgabizottság a szakmai vizsgáról kizár minket és szakmai vizsgát csak új záródolgozat készítése után tehetünk.

Győr, 2025. április 15.

tanuló aláírása

tanuló aláírása

tanuló aláírása

Tartalom

1. Bevezetés	4
2. Fejlesztői dokumentáció	5

2.1 Munkamegosztás	5
2.1.1 Feladatok elosztása	5
2.2 A rendszer követelményei.....	5
2.2.1 A rendszer által megvalósítandó szolgáltatások	6
2.3 Rendszer logika tervezése	7
2.3.1 Szerepkörök	7
2.3.2 Use Case forgatókönyvek	8
2.4 Felhasználó felület tervezése.....	10
2.4.1 Az oldalra történő belépés	11
2.4.2 Belépést követően	11
2.4.3 Kölcsönzések menüpont	12
2.4.4 Profil menüpont.....	13
2.4.5 Adatbázis tervezet	14
2.5 Alkalmazott technológiák, és a fejlesztés elméleti háttere	17
2.5.1 Felhasznált programozási, és jelölési nyelvek:	17
2.5.2 Fejlesztői környezet, keretrendszerek:	17
2.5.3 Adatbázis:	19
2.5.4 Verziókezelés	21
2.6 Frontend és Backend	23
2.6.1 Frontend	23
2.6.2 Reszponzivitás	24
2.6.3 Backend.....	25
2.6.4 HTTP-protocoll	25
2.6.5 Laravel Sanctum.....	26
2.7 Projekt tesztelése	27
2.7.1 ThunderClient – API tesztek	27
3.Felhasználói dokumentáció	29
3.1 Felhasználói fiók.....	29
3.1.1 Regisztráció	29
3.1.2 Bejelentkezés	31
3.1.3 Regisztrált, valamint már bejelentkezett felhasználók/admin számára elérhető funkciók	33

3.1.4 Főoldal (Katalógus)	34
3.1.5 Kölcsönzések	35
3.1.6 Profil.....	36
3.2 Admin jogosultság lehetőségei	38
3.2.1 Tartalom módosítás, kölcsönzés a katalógusban admin jogkörrel.....	39
3.2.2 Tartalom felvétel.....	40
3.2.3 Felhasználók	41
4. Összefoglalás.....	43
5. Idézett források	44

1. Bevezetés

Szakdolgozatunk témaként egy könyvtári rendszer fejlesztését választottuk, mely egy megbízható, könnyen használható platformot kínál a felhasználók, és az ott dolgozók számára. Itt a felhasználók könnyen, átlátható módon követhetik a kölcsönzéseiket, böngészhetnek a katalógusból, a könyvtárban dolgozók pedig ezen keresztül lesznek képesek rögzíteni a felhasználók által kölcsönzött termékeket. Célunk egy rendkívül felhasználóbarát weboldal készítése volt, mely nagyban megkönnyíti mind a felhasználók, mind pedig a könyvtári dolgozók számára a könyvek, és egyéb a katalógusban szereplő termékek kezelését, valamint nyomon követését. A program letöltés nélkül elérhető, eléréséhez csupán internet kapcsolatra van a felhasználóknak szüksége. Fontosnak tartottuk a projekt elkészítése során, hogy naprakész technológiákat alkalmazzunk, ezzel fejlesztve a saját piacképes tudásunkat. A választásunk kliensoldali részre a Vue-ra esett. A Vue egy JavaScript alapú keretrendszer mely rendkívül népszerű, és keresett világszerte, valamint egyik nagy előnye, hogy egyszerűen elsajátítható a konkurenciát tekintve. Szerver oldalról a világ egyik legnépszerűbb keretrendszerét, a Laravelt választottuk.

Projektünk fejlesztése során számos tudásra tettünk szert. Megtanultuk elsajátítani egy weboldal megtervezését nulláról, felépíteni egy adatbázist a weboldalhoz, valamint az API-k használatát.

A projektmunkánk az alábbi linken keresztül elérhető:

<https://github.com/papaybalint/Szakdogac>

Bejelentkezéshez szükséges adatok:

Felhasználói bejelentkezés:

Email: teszt@jedlik.eu

Jelszó: teszt

Admin bejelentkezés adatok:

Email: admin@jedlik.eu

Jelszó: admin

2. Fejlesztői dokumentáció

2.1 Munkamegosztás

2.1.1 Feladatok elosztása

A vizsgaprojektet 3 fős csapattal terveztük. A projekt legelején döntöttünk arról, hogy ki milyen szerepet fog vállalni a fejlesztés során. Hatos Zoltán volt felelős az adatbázis tervezetért, valamint a backendért, Erdős Dániel mind a frontend mind pedig a backend fejlesztésében részt vett. Pápay Bálint a Frontendért, a dokumentáció, és különböző az oldalhoz tartozó dizájntervekért volt felelős.

Az adatbázis választás során az SQLite-ra esett a választás. Részfeladatok:

- GUI tervezés
- Fejlesztési fázisok meghatározása
- Adatbázis modell készítés
- Frontend elkészítése
- Backend elkészítése
- Tesztelés

2.2 A rendszer követelményei

Minden rendszerrel kapcsolatban merülnek fel elvárások a felhasználók, üzemeltetők és fejlesztők részéről. Ezeknek a követelményeknek való megfelelés létfontosságú ahhoz, hogy a rendszer zökkenőmentesen működjön. Így hát mi is szeretnénk

meghatározni néhány alapelvet, amelyeket az általunk tervezett rendszerrel szemben elvárhatnak annak jövőbeli felhasználói.

Ezen alapelvek közé tartozhat a felhasználói felület intuitív, könnyen kezelhető kialakítása, a megbízható működés, és az azonnali reakcióidő biztosítása. A rendszer részletes, és pontos leírása kulcsfontosságú lehet az üzemeltető számára az általa nyújtott hatékony támogatás érdekében. Ugyancsak fontos szempont az adatkezelés, az adatvédelmi szabályoknak való megfelelés, valamint az esetleges hibák gyors, és hatékony kezelése.

Mindezen alapelvek betartása lehetővé teszi, hogy a rendszer hosszútávon kielégítse a felhasználók igényeit, és biztosítsa a stabil működést.

Általánosságba véve egy informatikai rendszernek 7 szempontnak kell megfelelnie, amik a következők:

- Megbízhatóság
- Biztonság
- Teljesítmény
- Skálázhatóság
- Karbantarthatóság
- Kompatibilitás
- Felhasználóbarátság

2.2.1 A rendszer által megvalósítandó szolgáltatások

A felhasználóinknak lehetőségük lesz a könyvtár katalógusát böngészni, ezen belül lehetőség van a katalóguson belül a szűrésre, a profiljuk adatainak megtekintésére, szerkesztésére, valamint az aktuális kölcsönzéseik megtekintésére.

- **Felhasználói oldalról:**
 - Katalógus böngészése
 - Személyes adatok szerkesztése
 - Aktuális kölcsönzések megtekintése
- **Üzemeltetői oldalról:**
 - Felhasználók kezelése, jogosultságok beállítása
 - Katalógus frissítése, bővítése
 - Kapcsolattartás fejlesztőkkel

- Kölcsönzések menedzselése

2.3 Rendszer logika tervezése

A rendszer logikai felépítésének alapos megtervezése egy projekt előkészítésének alapvető része, hiszen ez által minimalizálhatjuk a projekt során felmerülő problémákat. Ezáltal csökkenthető a projekt kivitelezéséhez szükséges pénz/idő ráfordítás. A részletes tervezés lehetővé teszi, hogy már a tervezési fázisok korai szakaszban felismerjük a logikai ellentmondásokat és azokat megfelelően kezeljük. Ezért szükséges tisztázni, hogy kik és hogyan fogják használni a programot és milyen funkcionalitásokra lesz szükségük, valamint milyen jogosultsági szintekhez lesz rendelve az egyes funkcionalitások elérése.

Fontos előre feltérképezni, hogy a felhasználók milyen adatokat kívánnak majd tárolni és ezekkel milyen műveleteket szeretnének végezni. Ezt követően pedig meg kell vizsgálni, hogy az adatok milyen összefüggésben állnak egymással. A pontos funkcionalitások és jogosultságok meghatározása kulcsfontosságú a rendszer hatékony és biztonságos működése szempontjából. Ezek alapján lehet meghatározni a weboldal mögött felépített adatbázis struktúráját.

Egy rendszer tervezése során továbbá fontos figyelembe venni az üzemeltetési szempontokat. Alaposan meg kell határozni az ezek során előforduló feladatokat, valamint az esetlegesen felmerülő problémákat és az azokra szolgáló válaszlépéseket. A „Use Case diagram” kiváló eszköz a felhasználók és a rendszer közötti lehetséges interakciók modellezésére. Ez a diagram segítségével szolgál abban, hogy részletesen feltérképezzük, hogyan használhatják a felhasználók a rendszert és használat közben milyen funkcionalitásokat érhetnek el.

Ezek mellett a „Use Case” diagram nem csupán a felhasználói interakciók során előforduló scénáriókat képes szemléletesen ábrázolni, hanem akár az üzemeltetési folyamatokat is. Ennek segítségével tehát a fejlesztők és üzemeltetők könnyebben fel tudnak készülni az üzemeltetés során felmerülő kihívásokra.

2.3.1 Szerepkörök

A kezdeti fázisában a rendszer két szerepkörrel fog működni: normál felhasználó és az adminisztrátor. (A rendszer továbbfejlesztése során elképzelhető, hogy új szerepkörök jönnek létre.) Az adminisztrátori szerepkör bővített jogosultságokkal rendelkezik egy normál felhasználóval szemben, ezzel lehetővé téve számára az

érzékenyebb adatokhoz való hozzáférést. Az adminisztrátor feladatai közé tartozik a rendszer karbantartása, a felhasználók általi igényeinek kielégítése, a regisztrált felhasználók kezelése, a felhasználókkal való kapcsolattartás, a szoftver verziók nyomon követése, valamint a rendszer folyamatos karbantartása és frissítése. Az adminisztrátor felel a felhasználók kezeléséért, ezzel biztosítva a weboldal zavartalan működését. Ezen felül az adminisztrátor felelős a rendszert használó személyek jogosultságainak beállításáért is.

A felhasználók böngészhetik a katalógust, megtekinthetik a kölcsönzéseiket, valamint módosíthatják a profiljuk adatait.

2.3.2 Use Case forgatókönyvek

Ebben a részben "Use Case" forgatókönyvekkel szemléltetnénk a rendszeren belüli eseményeket. Célunk bemutatni az eltérő jogosultsággal rendelkező szereplők által végezhető műveleteket és azt, hogy az egyes funkciókhoz való hozzáféréshez milyen jogosultság szükséges. Részletesen ismertetjük az átlagos felhasználók számára elérhető funkciókat, valamint azokat, amelyek kizárólag a rendszer üzemeltetéséért felelős személyek számára állnak rendelkezésre.

Felhasználói Use Case: Bejelentkezés/Regisztráció

Amikor valaki a weboldalra látogat csupán a kezdőlap érhető el számára. Ez magába foglalja a Rólunk, és a Footer (Nyitvatartás)-t. Ha valaki a teljes weboldal tartalmához hozzá szeretne férni, ahhoz be kell jelentkeznie. Ha a látogató nem rendelkezik felhasználói fiókkal, először regisztrálnia kell. Ekkor a rendszer általános adatokat fog kérni a felhasználótól, mint például név, felhasználónév, email, telefonszám, jelszó. A bejelentkezés során ellenőrzi a rendszer a bejelentkezni kívánó személy jogosultságait. Ennek megfelelő kezdőoldalra irányítja a személyt a rendszer és ezt követően elérhetővé válik számukra a felhasználói vagy adminisztrátori felületen található menüpontok.

Felhasználói Use Case: Katalógus böngészése

A felhasználó a sikeres belépést követően átirányítva lesz a katalógusra. Ezen fülön a felhasználó böngészhet a könyvtár rendszerében elérhető különböző tartalmak közül. A katalógusban természetesen különböző szűrési opciók is elérhetőek a felhasználók számára.

Felhasználói Use Case: Kölcsönzések

A felhasználóknak a rendszerben lehetőségük adódik a saját kölcsönzésük monitorozására is. A "Kölcsönzések" fülre kattintva a rendszer átirányítja a felhasználót az aktuális kölcsönzéseikhez. A felhasználó itt tekintheti meg a kölcsönzéseit. Minden kölcsönzés egy egyedi kölcsönzési azonosítót kap, ezen kívül látható a kölcsönző neve, a kölcsönzés dátuma, valamint a kölcsönzés lejárat dátuma, és a kölcsönzött item neve.

Adminisztrátor Use Case: Katalógus szerkesztése

Az admin-nak lehetősége lesz majd a katalógusban szereplő különböző elemek részletes megtekintésére, és ezeknek a szerkesztésére, esetleg adatbázisból való törlésükre is.

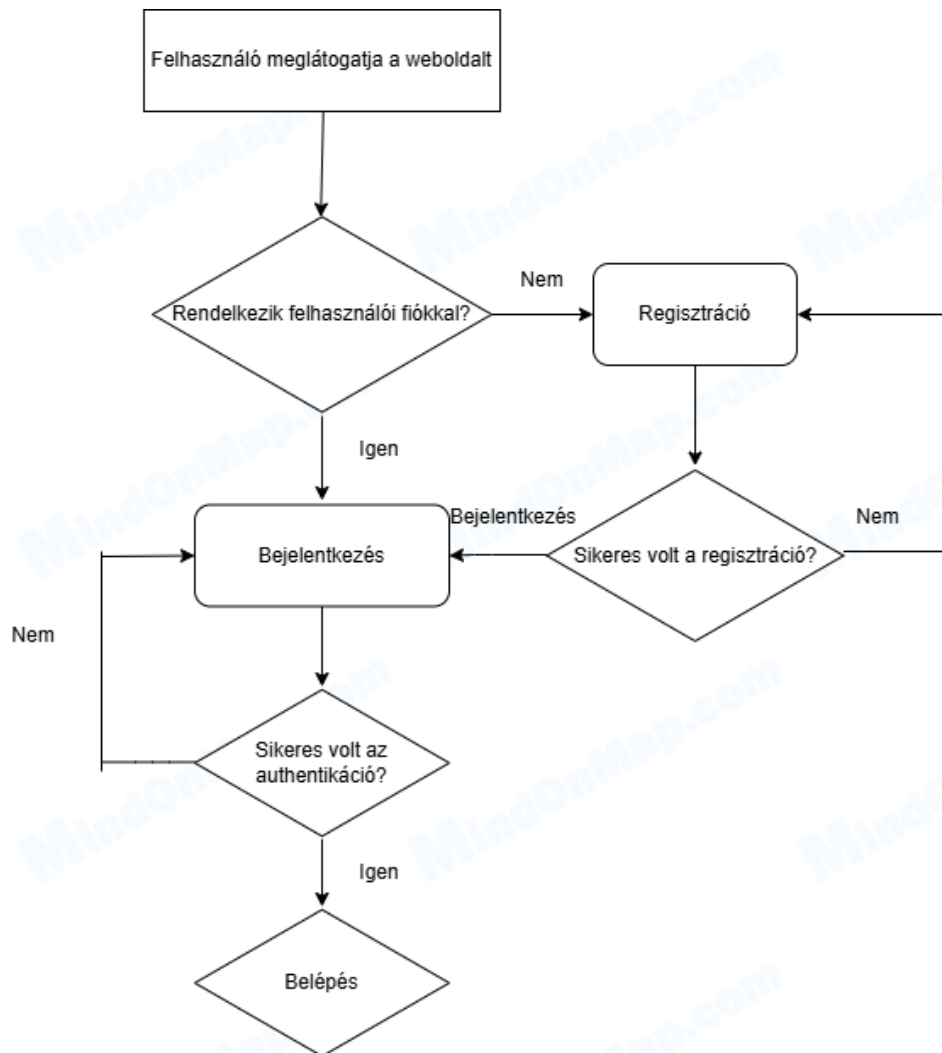
Adminisztrátor Use Case: Tartalom hozzáadása

Az adminisztrátor a bejelentkezést követően további menüpontokhoz férhet majd hozzá. Ezen egyike a Tartalom hozzáadása, melyen új elemeket lesz lehetősége a katalógushoz hozzáadni.

Adminisztrátor Use Case: Felhasználók kezelése

Az adminisztrátorok rendelkeznek hozzáféréssel a felhasználók adataihoz, a jelszavaik kivételével melyeket a rendszer az adatbázisba történő mentés előtt titkosít. A "Felhasználók" menüpontot választva megjelennek a regisztrált felhasználók, különböző szűrési opciók is elérhetőek az adminisztrátorok munkája megkönnyítése céljából. Az adminisztrátor továbbá képes lesz eltávolítani a felhasználókat a rendszerből.

Egy folyamatra a Bejelentkezés/Regisztrációhoz



1. ábra: Bejelentkezés/Regisztráció folyamatára.

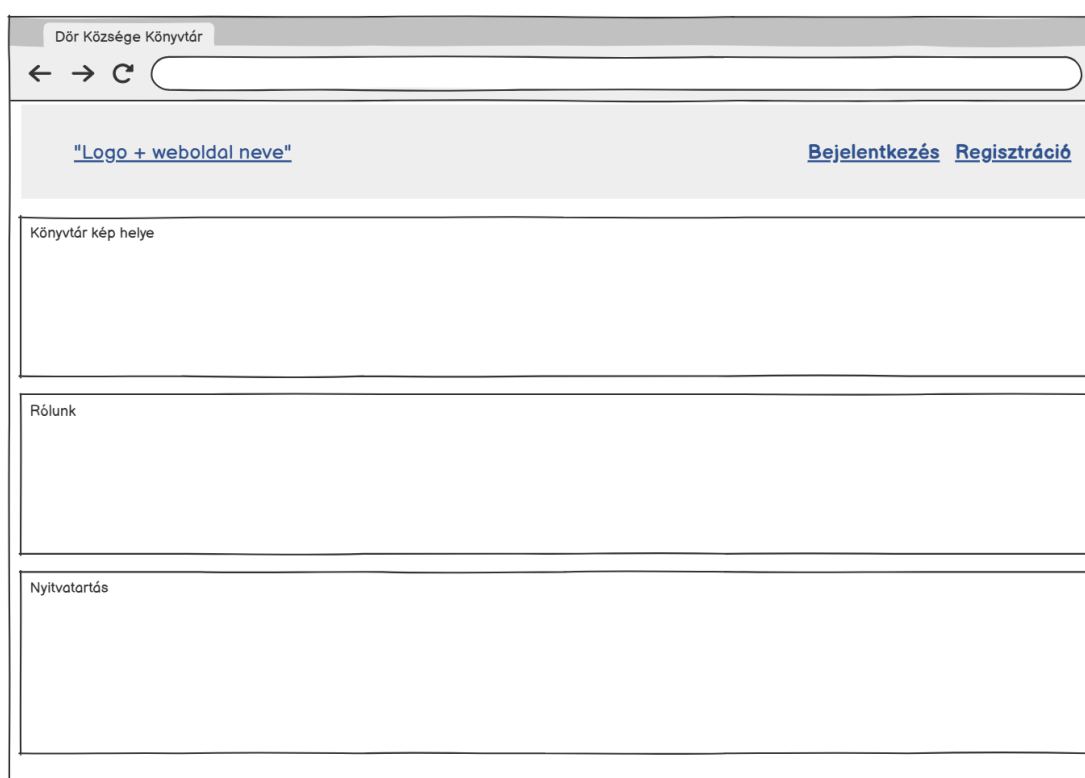
2.4 Felhasználó felület tervezése

A weboldal tervezése során készítettünk néhány grafikai tervet arról, hogy hogyan kívánjuk megvalósítani a weboldalt. Ezek segíthetnek az esetleges hibák felismerésében és működéshez szükséges funkciók megtervezésében, valamint átláthatóvá teszik, hogy az egyes funkciók hogyan kapcsolódnak egymáshoz.

A fejlesztést megelőző tervezés során készítettünk GUI (Graphical User Interface) terveket a weboldalról, ezek közül néhányat be szeretnénk mutatni. Igyekeztünk az egyes oldalakat és az azokhoz tartozó funkciókat minél alaposabban megtervezni úgy, hogy azok a mai standardoknak megfelelő legyen. Egy GUI terv segíthet a funkciók tervezésének folyamatában, segíthet feltárni, az esetleges logikai ellentmondásokat, továbbá a weboldalon keletkező adatok feltérképezésében is segítséget nyújthat, így ez segítségül szolgálhat az adatbázis tervezése során.

2.4.1 Az oldalra történő belépés

A weboldalra meglátogatását követően egy "Rólunk" szekció, egy kép a könyvtárról, valamint a nyitvatartás lesz megtekinthető a látogatók számára. Ezen felül a nav-bar-on a bal felső sarokban lesz egy kép a címről, a könyvtár nevével ellátva melyre kattintva lesz a továbbiakban lehetőségük a felhasználóknak a főoldalra visszanavigálnia. A jobb felső sarokban lesz lehetőség bejelentkezni, vagy regisztrálni a látogatóknak a megfelelő gombra kattintva.



2. ábra: Weboldal bejelentkezés nélkül.
(Forrás: Saját szerkesztés.)

2.4.2 Belépést követően

A belépést követően az alábbi felületen találja magát a felhasználó. A főoldalon látni fogja a katalógus teljes tartalmát, melyben böngészhet a teljes katalógusban és innen további szűrési feltételek is elérhetők. A navbar-on további menüpontok válnak elérhetővé, innen tud majd továbbmenni az alábbi menüpontok segítségével.

Dör Községe Könyvtár

← → ↻

"Logo + weboldal neve"
Kölcsönzések
Profil
Kijelentkezés

Könyvtár kép helye

Rólunk

A könyvtárunkban található tartalom:

Minden kategória
Cím
Szerző
Kiadás éve

Könyv A
Könyv B
Könyv C
Könyv D
Könyv E

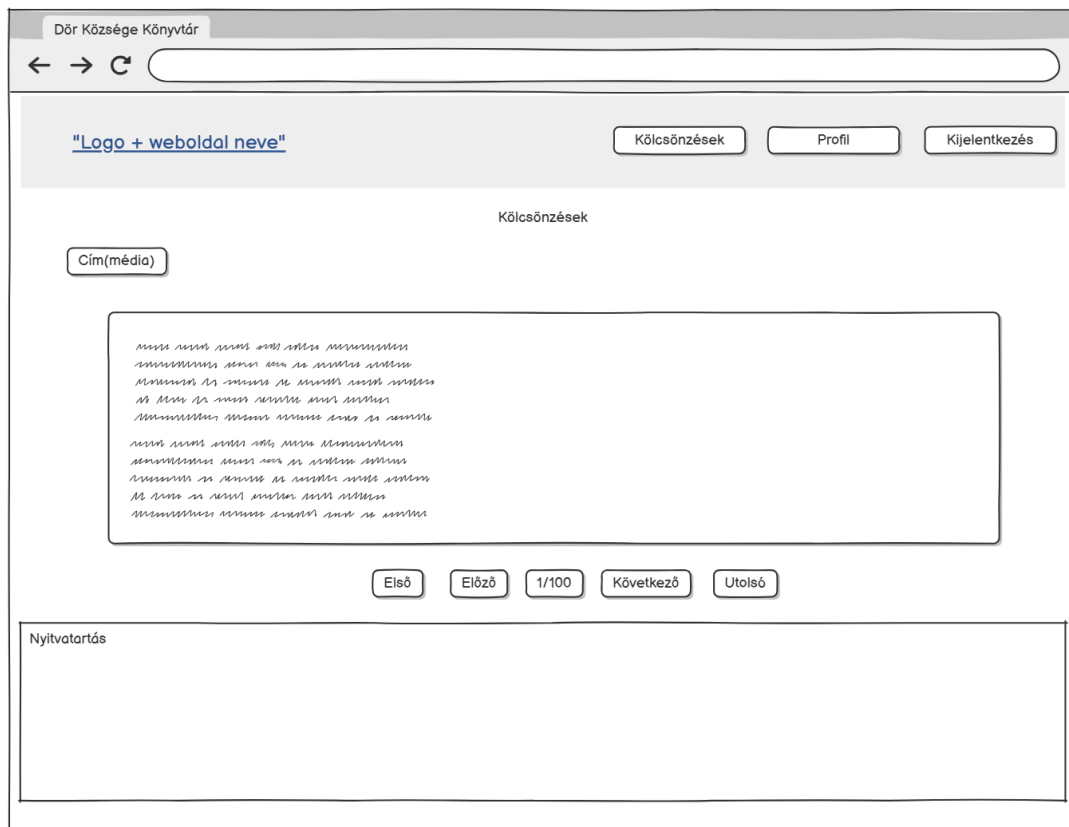
Első
Előző
1/100
Következő
Utolsó

Nyitvatartás

3. ábra: Főoldal bejelentkezve.
(Forrás: Saját szerkesztés.)

2.4.3 Kölcsönzések menüpont

A "Kölcsönzések" menüpontban lehet megtekinteni az adott felhasználó aktuális kölcsönzéseit.



4. ábra: Kölcsönzések
(Forrás: Saját szerkesztés.)

2.4.4 Profil menüpont

A "Profil" menüpont alatt lesz a felhasználónak lehetősége a már korábban regisztráció során megadott adatok módosítására, valamint amennyiben a felhasználó úgy dönt, hogy nem szeretne a továbbiakban szerepelni az adatbázisban, ugyanezen fülön belül lesz lehetősége a profilját törölni is.

The screenshot shows a web browser window with the address bar displaying "Dör Községi Könyvtár". The page has a header with a logo placeholder "Logo + weboldal neve" and navigation buttons: "Kölcsönzések", "Profil", and "Kijelentkezés".

The main content area is titled "Profil információk" and contains the following form elements:

- Felhasználónév:** Input field with "teszt" entered.
- Vezetéknév:** Input field with "Teszt" entered.
- Keresztnév:** Input field with "Elek" entered.
- Születéshely:** Input field with "Győr" entered.
- Születési idő:** Input field with "1999.06.29" entered.
- Telefonszám:** Input field with "+36 20 7463234" entered.
- Email:** Input field with "teszt@jedlik.eu" entered.
- Mentés:** Button below the email field.
- jelszó frissítése:** Section header for password change.
- Jelenlegi jelszó:** Input field.
- Új jelszó:** Input field.
- Button:** Button below the new password field.
- Jelszó megerősítése:** Input field.
- Button:** Button below the confirmation field.
- Mentés:** Button below the password change section.
- Fiók törlése:** Section header for account deletion.
- Fiók törlése:** Button below the account deletion header.

At the bottom left, there is a "Nyitvatartás" section.

5. ábra: Profil
(Forrás: Saját szerkesztés.)

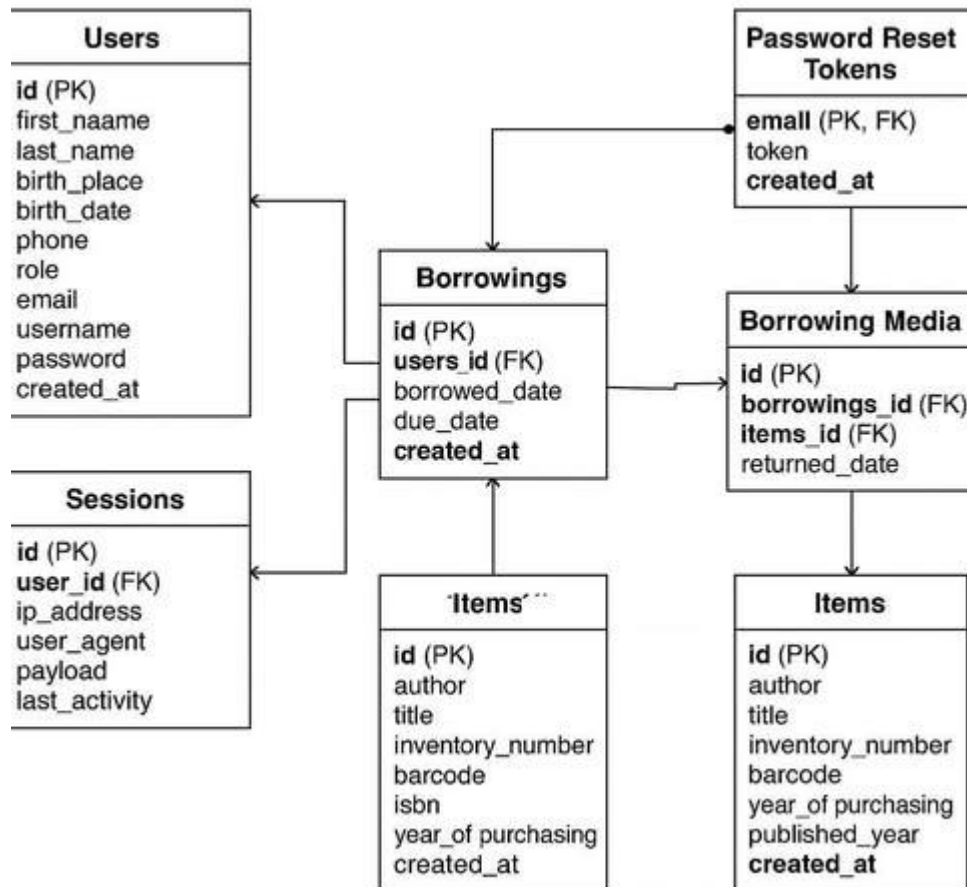
2.4.5 Adatbázis tervezet

Az adatbázis tervezethez a könyvtári online alkalmazásunkhoz szükségünk lesz, egy jól megtervezett és felépített adatbázisra, ahonnan kezelni tudjuk a könyvtár tartalmát, regisztrált tagjainak adatait és nyomon követhető legyen a könyvek kölcsönzése. Ehhez az alábbi rövid leírásban mutatnánk be az adatbázis felépítésének tervezetét. Öt táblát hoztunk létre a következő formában:

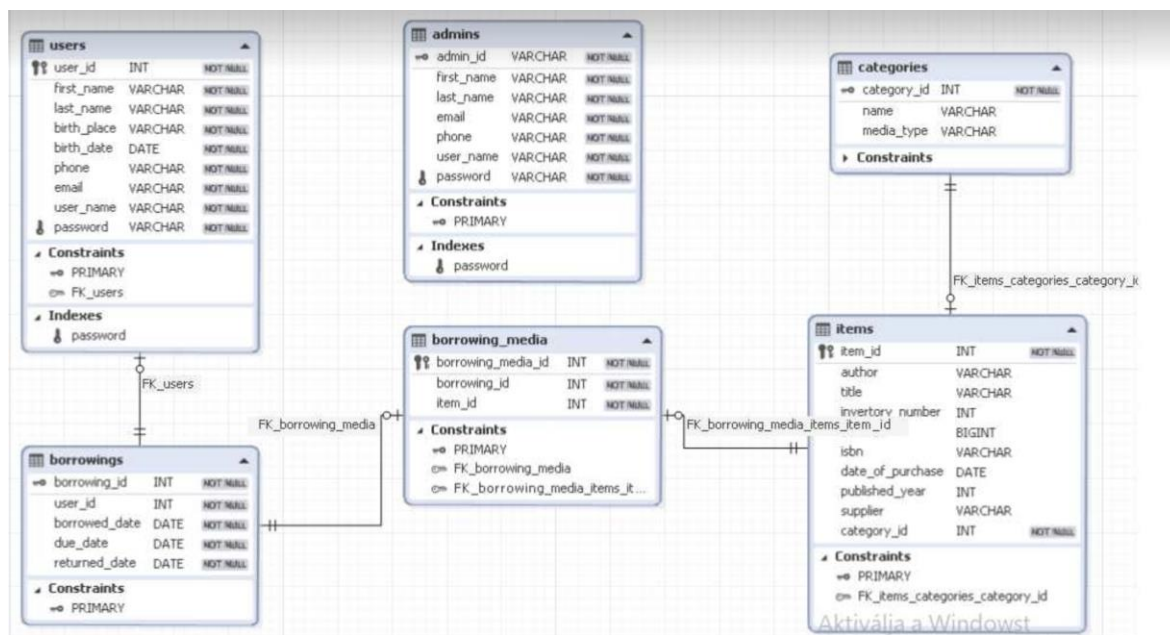
- Tábla: user (felhasználók)
- id, vezetéknév, keresztnév, születési hely, születési dátum, telefonszám, role(user/admin), email cím, felhasználónév, jelszó

- Tábla: kölcsönzések id, kölcsönzés időpontja, kölcsönzés lejárat időpontja, idegen kulcs felhasználó_id
- Tábla: kölcsönzött média id, visszavétel időpontja, idegen kulcs kölcsönzés_id, idegen kulcs tétel_id
- Tábla: tételek id, szerző, cím, leltári szám, vonalkód, isbn, beszerzés dátuma, kiadás dátuma, szállító, idegen kulcs kategória_id
- Tábla: kategóriák id, kategória neve, média típusa,
- Tábla: Adminok Ugyanazok az adatok mint a felhasználók táblánál, csak a role itt fix admin lenne. (Ezt a táblát külön nem használjuk, de lehetőség van a későbbiekben a bevezetésére)

A táblák közötti kapcsolat: Felhasználók tábla egy a többhöz kapcsolatban áll a kölcsönzések táblával, több kölcsönzés tartozhat egy felhasználóhoz. A kölcsönzés táblánk is egy a többes kapcsolatban áll a kölcsönzött média táblával, és kapcsolótáblaként funkcionál, tételek és a kölcsönzések között. A tétel tábla is egy a többes kapcsolatban áll a kategória táblával, ami így jól normalizált, minden könyv vagy médium egy kategóriához tartozik. A táblák szerkezete és a táblák közötti kapcsolatok szerkezeti ábrája ER diagram:



6. ábra: ER Diagram
(Forrás: Saját szerkesztés.)



7. ábra: Adatbázis tervezet
(Forrás: Saját szerkesztés.)

Az adatbázis kezelésére az online felületen keresztül ADMIN rendszerjogosultsággal lesz lehetőség, új könyvek hozzáadására, illetve eltávolítására, a tagságok felügyeletére és a kölcsönzés nyomon követésére. A regisztrációs oldal fogja biztosítani a mindenki számára elérhető tagsági regisztrációt, és az oldal használatához szükséges belépési jelszavak megadását a leendő tagoknak.

2.5 Alkalmazott technológiák, és a fejlesztés elméleti háttere

2.5.1 Felhasznált programozási, és jelölési nyelvek:

A projekt készítése során programozási és jelölő nyelveket kombináltunk, hogy ezzel biztosítsuk a webalkalmazáshoz szükséges funkcionalitást, és esztétikai megjelenést. Erre a célra a következő nyelveket alkalmaztuk:

- **HTML (Hypertext Markup Language):** Ez egy olyan leíró nyelv, amelyet arra terveztek, hogy segítségével strukturálisan felépíthetővé váljon egy weboldal szerkezete. Ez a kód tartalmazza a weboldalon megjelenített tartalmi elemeket, mint például a különböző szövegek, képek, linkek.
- **CSS (Cascading Style Sheet):** Ez a nyelv felelős az egyes HTML elemek formázásáért és dizájnjáért, mely magába foglalja a színek, betűtípusok és animációk beállítását.
- **Javascript:** Ez egy programozási nyelv, amely felelős a weboldalak interaktivitásáért. Segítségével lehetővé válik a weboldal tartalmának dinamikus módosítása, események kezelése és általában véve a dinamikus működés biztosítása.

2.5.2 Fejlesztői környezet, keretrendszerek:

- **Fejlesztői környezet:** A fejlesztőkörnyezet (IDE) egy olyan szoftveralkalmazás, mely segíti a programozókat a kód hatékony fejlesztésében. Növeli a fejlesztők produktivitását azzal, hogy biztosít olyan funkciókat, mint a kódszerkesztés, tesztelés mindezt egy könnyen átlátható, használható alkalmazásban.

A projekt megvalósításához a **visual studio code** nevű alkalmazásra esett a választásunk.

- **Keretrendszer:** A keretrendszer, más néven munkakeret olyan eszközök, könyvtárak és bevált gyakorlatok összessége, amelyek egyszerűsítik az alkalmazások vagy rendszerek fejlesztésének folyamatát. A keretrendszer sablonként működik, és lehetővé teszi a fejlesztők számára, hogy anélkül építsenek alkalmazásokat, hogy a nulláról kellene kezdeniük. A keretrendszert úgy tervezték, hogy újra felhasználható legyen, és biztosítsa a konzisztenciát a projekt során.
 - **Laravel:** A Laravel a PHP nevű szerveroldali szkriptnyelvhez tartozó egyik keretrendszer, egy olyan nyílt forráskódú webes alkalmazás, amellyel gyorsan és egyszerűen lehetséges testre szabott webes alkalmazásokat tervezni.

A fejlesztők azért részesítik előnyben a Laravelt a többi keretrendszerhez képest, mert kiemelkedő teljesítményt nyújt, sok funkciót kínál, illetve skálázható is. Az MVC-t, vagyis a Model View Controllert követi, ezért hatékonyabb és biztonságosabb, mint a natív PHP. Megkönnyíti az olyan általános feladatok elvégzését, mint a hitelesítés (authentication), a munkafázis (sessions) a routing, vagy a gyorsítótárba másolás (caching). A projektmunka elkészítésekor a Laravel (x.11) verzióját használtuk.
 - **Vue:** A Vue.js egy modern, nyílt forráskódú JavaScript keretrendszer, amelyet Evan You hozott létre. Főként a felhasználói felületek és egyoldalas alkalmazások (SPA-k) fejlesztésére használják. Előnyei többek közt, hogy könnyen tanulható, komponens alapú a felépítése. Továbbá kiválóan működik együtt Laravellel.

Wireframing tool , GUI (Graphical User Interface) elkészítéséhez:

- A dizájn tervezéséhez a Balsamiq nevű wireframing toolra esett a választásunk.

2.5.3 Adatbázis:

Az adatbázis a weboldal működéséhez szükséges, valamint annak használata során keletkező adatok tárolására szolgál. Adatok az adatbázis gyakran különböző adatbázis kezelő rendszerek segítségével kerülnek tárolásra, hiszen ezek felelnek az adatok hatékony kezeléséért. Népszerűbb adatbázis kezelő rendszerek például a(z) SQLite, MySQL, a PostgreSQL, a MongoDB. Mi ezek közül a SQLite-t választottuk adatbázisnak. Az SQLite egy relációs adatbázis-kezelő rendszer, amely a sebességre helyezi a hangsúlyt. Az SQLite nem egy önálló, hanem egy beágyazott adatbázis, ezáltal egyszerűbb használatot és kevesebb adminisztrációs beállítást kínál, mint az önálló vagy kliens-szerver alapú adatbázisok. A Laravelnek köszönhetően rendkívül egyszerű az adatbázisokhoz való kapcsolódás, illetve a lekérdezésük egyaránt. Az adatbázisunk konfigurációs fájlja a `librarydor.sqlite`. Ebben a fájlban található meg az összes eltárolt adatbázis kapcsolat. A Laravel jelenlegi a dokumentáció írásakor a legfrissebb verziója(11.x) 5 adatbázisrendszert támogat, melyek a következők: SQLite, PostgreSQL, SQL Server 2017+, MySQL, MariaDB. Mi, ahogy azt már említettük, az SQLite-ot használjuk.

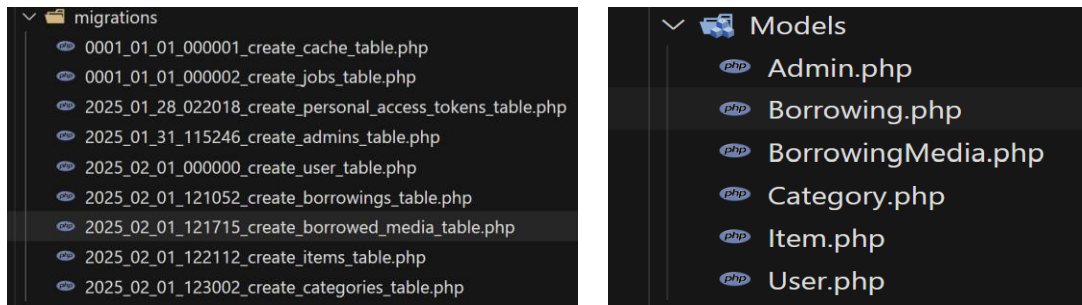
Az adatbázis létrehozása után, a különböző konfigurációk a `.env` nevű fájlban történnek. Itt határozhatjuk meg a kapcsolódási paramétereket, melyek lehetővé teszik az alkalmazásunk számára, hogy sikeresen kommunikáljon az adatbázissal.

```
DB_CONNECTION=sqlite
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=./database/librarydor.sqlite
DB_USERNAME=root
DB_PASSWORD=
```

8. ábra: `.env` kódrészlet

Az alábbi kódrészletben azt láthatjuk, hogy egy SQLite adatbázisra csatlakozunk, az adatbázisunk IP címe pedig 127.0.0.1, vagyis localhost. Az adatbázisunk neve `librarydor.sqlite`, felhasználónév alapértelmezett `root`, jelszó pedig a mi esetünkben nincs beállítva.

Amennyiben a kapcsolat sikeresen létrejött, az adatbázis, valamint az alkalmazás között, megkezdődhetnek a táblák létrehozása “php artisan make:model “táblanév” -m” paranccsal. A parancs létrehoz mind a “Models” mind pedig a “migrations” mappába egy-egy fájlt a tábláknak.



9. ábra: Migrations, Models kódrészlet.

A “migrations” mappában létrehozott fájlban adható meg, hogy az egyes adattáblák milyen attribútumokkal rendelkezzenek.

```
public function up(): void
{
    Schema::create('admins', function (Blueprint $table) {
        $table->id();
        $table->string('first_name');
        $table->string('last_name');
        $table->string('role')->default('admin');
        $table->string('email')->unique();
        $table->integer('phone');
        $table->string('username');
        $table->string('password')->unique();
        $table->timestamps();
    });
}
```

10. ábra: Migrations kódrészlet

Miután az összes adattáblát létrehoztuk, a “php artisan migrate” parancs segítségével hozhatjuk létre az adatbázisban is.

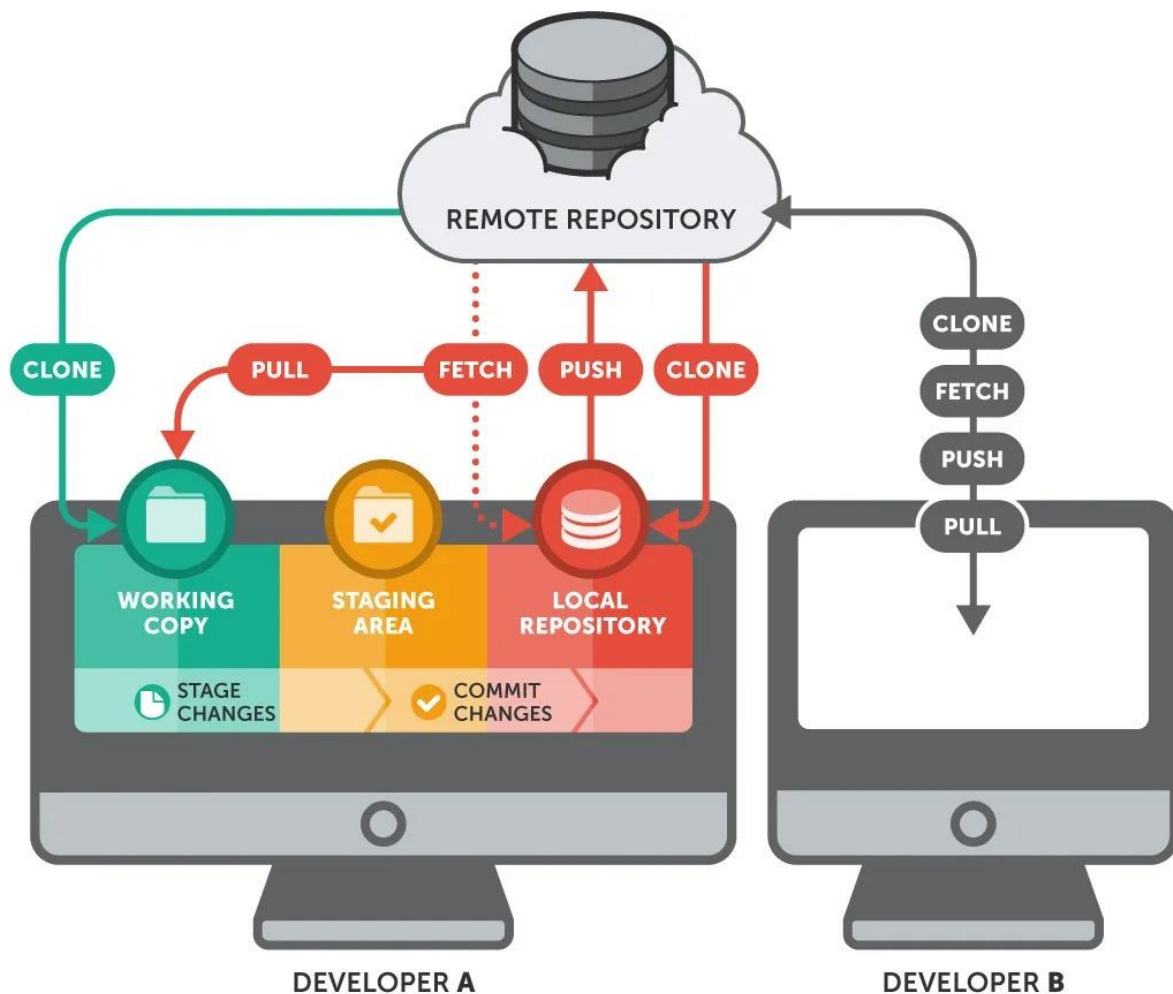
	id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT	first_name varchar NOT NULL	last_name varchar NOT NULL	birth_place varchar NOT NULL	birth_date date NOT NULL	phone varchar NOT NULL	rd
1	1	Admin	Úr	Kunzefurt	1997-01-06	+36 20 7830733	ad
2	2	Teszt	Elek	Győr	2004-01-15	+36 20 9824419	us
3	3	Lexi	Murazik	North Karlee	2014-06-19	+36 20 6301063	us
4	4	Cade	Kassulke	Jairoborough	1987-11-29	+36 20 5042654	us
5	5	Kamille	Corkery	South Macey	1995-04-10	+36 20 1928896	us
6	6	Felipa	Smith	South Sincere	1997-03-01	+36 20 3721226	us
7	7	Amya	McGlynn	North Noe	2002-01-20	+36 20 8248973	us
8	8	Elda	Adams	North Teresafort	2010-05-20	+36 20 9899465	us

10.ábra: Adatbázis Users tábla.

2.5.4 Verziókezelés

Mivel a projektet többen készítettük, és dolgoztunk rajta párhuzamosan, fontos volt számunkra a projekt követhetősége. Választásunk a GitHubra esett, mely manapság a világon az egyik legismertebb verziókövető rendszere, amit számos kis- és nagyvállalatnál is használnak különböző projektek során. Olyan ismert cégeknél is mint például a Pinterest, a Mercedes-Benz. A GitHub segítségével képesek voltunk lekövetni a különböző módosításokat.

A GitHub egy felhőalapú platform, amely a Gitet, egy elosztott verziókövetési rendszert használ a magjában. A GitHub platform leegyszerűsíti a projekteken való együttműködés folyamatát, és egy webhelyet, parancssori eszközöket és általános folyamatot biztosít, amely lehetővé teszi a fejlesztők és a felhasználók együttműködését.



11.ábra: Github műveletek.

Forrás: (<https://medium.com/mindorks/what-is-git-commit-push-pull-log-aliases-fetch-config-clone-56bc52a3601c>)

Az ábrán látható módon a git segítségével a következő műveletek végezhetőek el:

- **Push:** Ez a művelet arra szolgál, hogy a projekten végzett lokális változásainkat megoszthassuk a projekten dolgozó többi csapattal úgy, hogy feltöltsük azt a verziókövető rendszerbe, vagyis a GitHub-ra.
- **Pull:** Ennek a műveletnek a segítségével tudjuk frissíteni lokálisan a kódunkat a többi fejlesztő által eszközölt módosításokkal.
- **Fetch:** Ez a művelet szintén a GitHub-ról húzza le a változtatásokat azzal a különbséggel, hogy nem hajtja végre automatikusan az összefésülési műveletet, hanem lehetőséget biztosít ellenőrizni a változtatásokat mielőtt azokkal felülírnánk a lokális mentésünket.

- **Clone:** Ez egy tipikus inicializáló művelet, amikor elkezdünk dolgozni egy projektben. Lehetővé teszi, hogy egy projekt teljes tartalmáról lokális másolatot hozzunk létre.

2.6 Frontend és Backend

A frontend és backend összehangolt fejlesztése szükséges egy webalkalmazás létrehozásához. Szeretnénk röviden ismertetni, hogy mely feladatok kapcsolódnak a frontend és melyek pedig a backend fejlesztéshez, illetve melyik miért felelős az oldal működésének szempontjából.

A frontend fejlesztő felelős a vizuális elemek tervezéséért és megvalósításáért, amelyekkel a felhasználók kapcsolatba lépnek a felületen. Biztosítja, hogy a termék ne csak esztétikus legyen, hanem jól és könnyen használható is, továbbá zökkenőmentes és felhasználóbarát élményt nyújtson. Ebbe beletartozik az elrendezés megalkotása, interaktív funkciók fejlesztése és a különböző böngészők és eszközök közötti kompatibilitás biztosítása.

Ezzel szemben a backend fejlesztő a színpalak mögött dolgozik, kezeli azokat a technikai részeket, amelyek lehetővé teszik az alkalmazások zavartalan működését. A szerveroldali logikára összpontosít, karbantartja az adatbázisokat, és megoldja, hogy az adatok zökkenőmentesen átjussanak a szerver és a frontend között. A frontend az, amit a felhasználók látnak, a backend amely az alkalmazást működteti, biztosítva a működést. Bár a működési területeik eltérnek, a frontend fejlesztő a felhasználói felületre és élményre koncentrál, a backend fejlesztő az alkalmazás belső mechanikájával foglalkozik, szerepeik összefonódnak.

Együttműködve építenek és tartanak karban funkcionális, hatékony és vizuálisan attraktív alkalmazásokat és weboldalakat. Mindkét szerepkör egyedi készségeket igényel: a frontend fejlesztőnek jártasnak kell lennie a HTML, CSS és JavaScript használatában, a backend fejlesztőnek pedig szüksége van szerveroldali programozási ismeretekre, adatbázis-kezelésben és szerverkonfigurációban való jártasságra.

2.6.1 Frontend

A frontendünk létrehozására használt jelölő és programozási nyelveket már fentebb bemutatottuk.

Az egyik eszköz, ami nagyban segítette a Dör Községi Könyvtár dizájnjának kialakítását az a Tailwind CSS volt. A Tailwind CSS úgy működik, hogy beolvassa az összes HTML fájlt, JavaScript komponenseket, és minden más sablont, amely osztályneveket tartalmaz. Ezek alapján létrehozza a szükséges stílusokat, majd ezeket egy statikus CSS fájlba írja. Előnyei:

- Gyors
- Megbízható
- Futásidejű feldolgozás nélkül fut

2.6.2 Reszponzivitás

A kész projekt kiszolgálja a mai modern elvárásokat. A részponzivitás egy alap elvárás napjainkban, ennek megfelelően minden megjelenítő eszközt támogat. Ezt szintén a Tailwind segítségével tudtuk elérni.

A továbbiakban a ListPage.vue-ból szeretnénk egy kódrészletet prezentálni.

Ez a kódrészlet biztosítja a kártyás megjelenítést Vue.js segítségével. A `paginatedItems` listában található elemeket egy rácsszerkezetben jelenítjük meg, amely részponzív módon alkalmazkodik a különböző képernyőméretekhez: mobilon egy oszlopban, nagyobb kijelzőkön pedig akár több oszlopban is láthatóak a kártyák. Minden egyes elemhez tartozik egy kártya, amely tartalmazza a címét, a szerzőjét, a kiadás évét, valamint a kategóriáját és a médiatípusát, amennyiben az is meg van adva. A kártyákon megjelenik egy figyelmeztetés is abban az esetben, ha az adott tétel éppen ki van kölcsönözve. Ezt az `isBorrowed(item)` függvény dönti el, és ha igaz, akkor egy piros Kikölcsönözve felirat jelenik meg a kártya alján. Amennyiben az aktuális felhasználó adminisztrátori jogosultsággal rendelkezik „Részletek” gomb is megjelenik, amelyre kattintva egy modális ablakban további információk jeleníthetők meg az adott itemről. A stílusokat Tailwind CSS segítségével alakítottuk ki, amely modern, részponzív megjelenést és animált átmeneteket biztosít, mint például hover-effektek.


```

<!-- Kártyák -->
<div class="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 xl:grid-cols-5 gap-6">
  <div v-for="item in paginatedItems" :key="item.id"
    class="bg-white p-4 rounded-lg shadow-lg hover:shadow-xl transition-shadow duration-300 flex flex-col relative"
    :class="{ 'pb-10': isBorrowed(item) && auth.user.role !== 'admin' }">
    <h3 class="text-lg font-semibold mb-2 text-gray-800">{{ item.title }}</h3>
    <p class="text-sm text-gray-600 mb-1">Szerző: {{ item.author }}</p>
    <p class="text-sm text-gray-600 mb-1">Kiadás éve: {{ item.published_year }}</p>
    <p class="text-sm text-gray-600 mb-1">Kategória: {{ categories[item.categories_id - 1].name }} <span
      v-if="categories[item.categories_id - 1].media_type">{{ categories[item.categories_id -
        1].media_type
      }}</span></p>

    <!-- Kikölcsönözve felirat -->
    <div v-if="isBorrowed(item)"
      class="absolute bottom-4 left-4 bg-red-500 text-white py-1 px-3 rounded-md text-xs">
      Kikölcsönözve
    </div>

    <!-- Részletek Gomb -->
    <div v-if="auth.user.role === 'admin'"
      class="mt-auto flex justify-end space-x-2 space-y-2 sm:space-y-0">
      <button @click="openModal(item)"
        class="px-4 py-2 bg-blue-500 text-white rounded-lg hover:bg-blue-600 text-xs sm:text-sm">
        Részletek
      </button>
    </div>
  </div>
</div>

```

12.ábra: ListPage.vue kódrészlet.

2.6.3 Backend

Csakúgy, mint a frontendnél, fentebb a backend létrehozásánál használt eszközöket is ismertettük így ezeket ezúttal sem részletezzük.

2.6.4 HTTP-protocoll

A HTTP (HyperText Transfer Protocol) a World Wide Web alapvető protokollja, amely lehetővé teszi a böngészők és a web szerverek közötti kommunikációt.

A szabvány több metódust fogalmaz meg:

- **GET:** Ezt a metódust használják a kliensek egy adott erőforrás lekérésére a szerverről. A GET kérések általában csak olvasásra szolgálnak, és nem módosítják az erőforrást. Példa: egy weboldal letöltése.
- **POST:** Ezt a metódust használják az adatok elküldésére a szerverre, amely feldolgozza azokat. A POST kérések általában az erőforrások létrehozására vagy módosítására szolgálnak. Példa: egy űrlap kitöltése és elküldése.
- **PUT:** Ezt a metódust használják egy erőforrás létrehozására vagy teljes helyettesítésére a szerveren.
- **DELETE:** Ezt a metódust használják egy adott erőforrás törlésére a szerverről. Példa: egy fájl vagy rekord törlése.

- **HEAD:** Ezt a metódust használják egy erőforrás metaadatainak lekérésére a szerverről, anélkül, hogy magát az erőforrást letöltenék. A HEAD kérés válasza ugyanazokat a fejléceket tartalmazza, mint a GET válasz, de az üzenettörzs nélkül.
- **OPTIONS:** Ezt a metódust használják annak megállapítására, hogy a szerver milyen HTTP metódusokat támogat egy adott erőforráson. Példa: egy szerver támogatott metódusainak lekérdezése.

Az alábbi képen néhány HTTP metódus látható az alkalmazásunk forráskódjából.

```
Route::get('/categories', [CategoryController::class, 'index']);
Route::get('/categories/{id}', [CategoryController::class, 'show']);

Route::get('/import', [ImportController::class, 'run']);

Route::get('api/items/{id}', [ItemController::class, 'show']);
Route::get('api/categories/{id}', [CategoryController::class, 'show']);
Route::delete('api/items/{id}', [ItemController::class, 'destroy']);
Route::put('api/items/{id}', [ItemController::class, 'update']);
```

13. ábra:Route kódrészlet.

2.6.5 Laravel Sanctum

A fejlesztés során a hitelesítésre a Laravel Sanctumot használtuk. A Sanctum egy egyszerű hitelesítési rendszert nyújt SPA-knak(egyoldalas alkalmazások),mobil alkalmazásoknak, és szimpla token alapú API-k számára egyaránt. A Sanctum lehetővé teszi minden felhasználó számára az alkalmazásból, hogy több API token jöjjön létre a fiókjukhoz. Ezek a tokenek olyan hatóköröket kapnak, melyek meghatározzák, hogy a tokenek milyen műveletet hajthatnak végre.

A Sanctum egyetlen adatbázisban tárolja a felhasználói API-tokeneket és hitelesíti a bejövő HTTP kéréseket az Authorization fejlécen keresztül amelynek tartalmaznia kell egy érvényes API-tokenet. Ezen kívül azért is egyszerű, mivel az SPA-k(egyoldalas alkalmazások) hitelesítésekor, amelyek a Laravel által üzemeltetett API-val kommunikálnak nem használ semmilyen token-t. A Sanctum beépített cookie alapú hitelesítési szolgáltatásait használja. A Sanctum a Laravel web hitelesítési védelmét használja, amely biztosítja CSRF védelmet, a munkamenet hitelesítést, valamint védelmet nyújt XSS-en keresztüli hitelesítéskor fellépő kiszivárgás ellen.

A Sanctum csak akkor próbál hitelesíteni sűtik használatával, ha a bejövő kérés a saját SPA frontendjéről származik. Amikor a Sanctum megvizsgálja a beérkező HTTP kérést, először ellenőrzi a hitelesítési cookie-kat, és ha nincs ilyen, akkor a Sanctum megvizsgálja, hogy van-e Authorization érvényes API-token a fejlécben.

Az API token-ek kiadásával hitelesítettük az alkalmazás API kérését. Mikor API kérés érkezik, a token-nek szerepelnie kell az Authorization fejlécben. Ezt láthatjuk a User.php fájlunkban.

```
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    /** @use HasFactory<\Database\Factories\UserFactory> */
    use HasFactory, Notifiable, HasApiTokens;
```

14.ábra: Sanctum kódrészlet.

2.7 Projekt tesztelése

2.7.1 ThunderClient – API tesztek

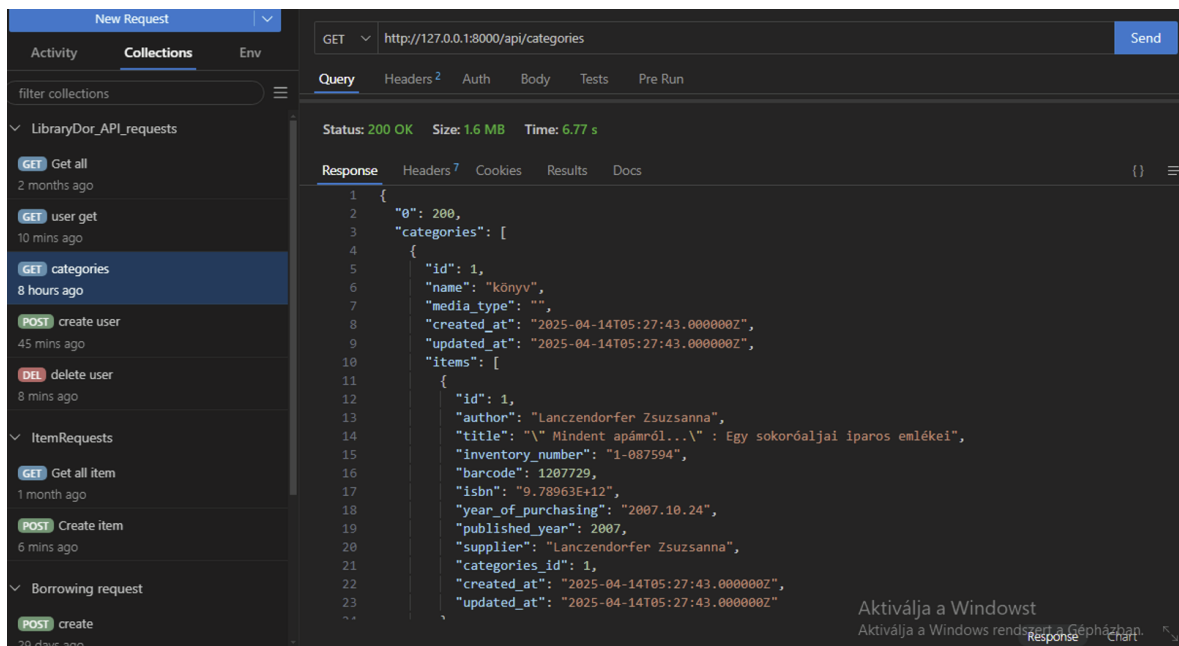
A ThunderClient egy könnyű Rest API kliens kiegészítő a Visual Studio Code számára.

Előnyei:

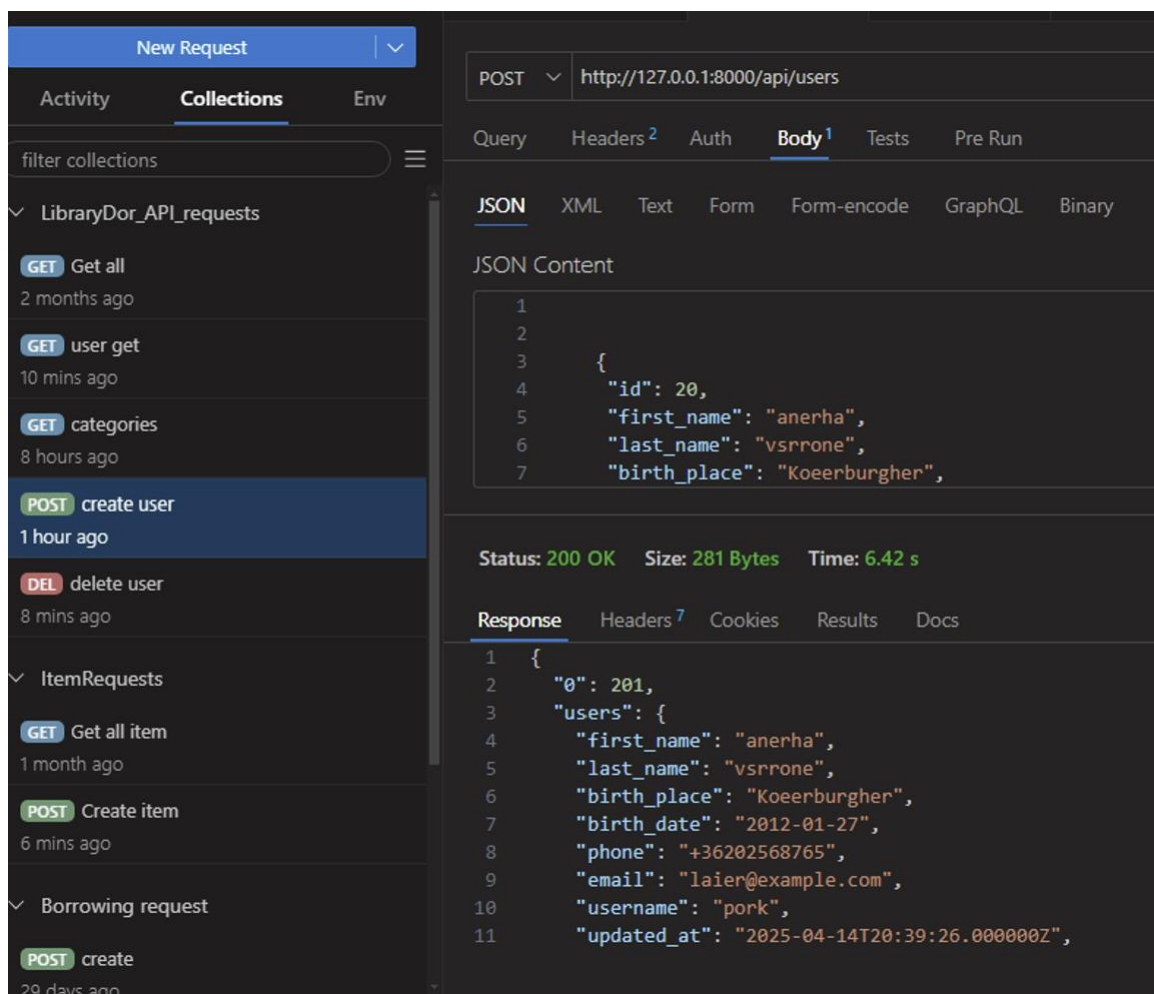
- Könnyen használható
- Átlátható
- Lokális tárhely
- Szkript nélküli tesztelést tesz lehetővé

A tesztelés során lehetőség van különböző HTTP metódusok például: GET, POST, PUT, DELETE, stb. Rendkívül egyszerű, átlátható felhasználói felülettel rendelkezik, melynek köszönhetően a kérések konfigurálása, válaszok vizsgálása egyszerű.

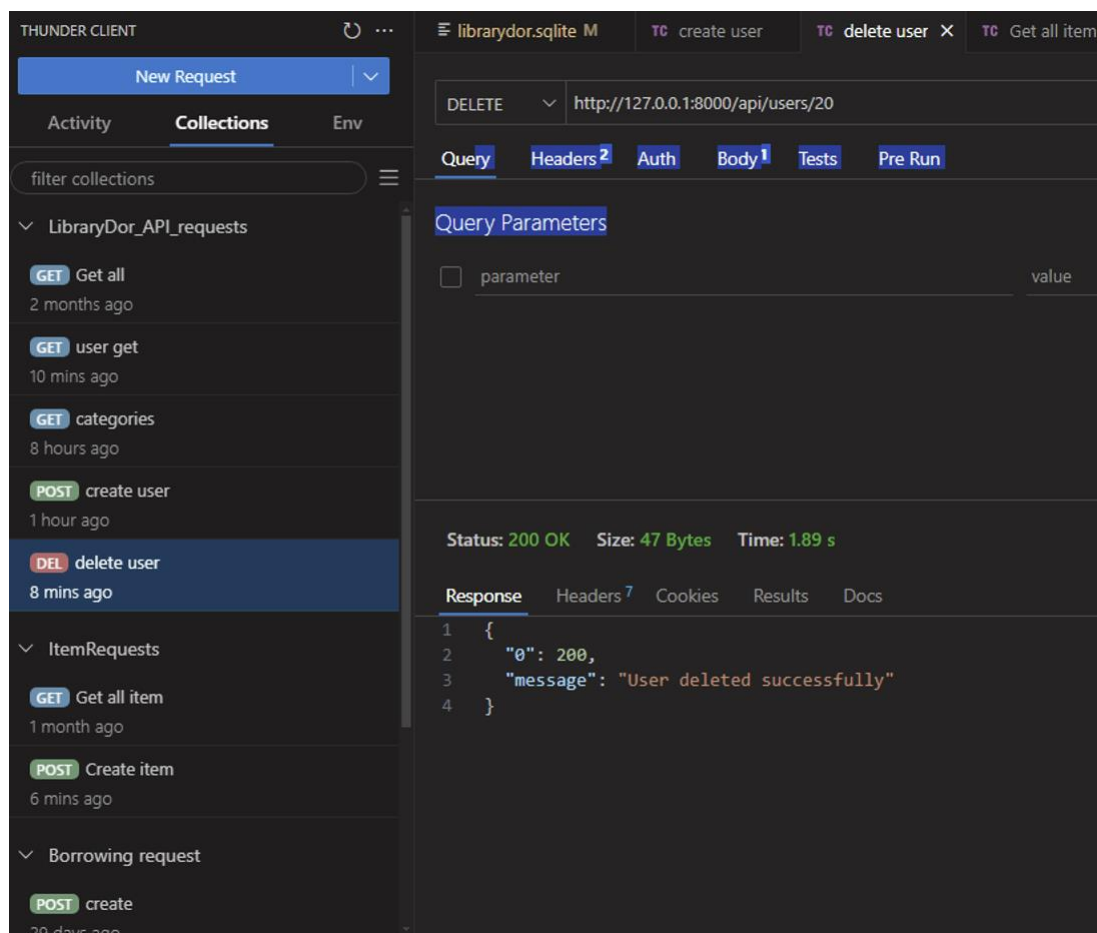
Néhány API teszt a projektből



15. ábra: Thunderclient teszt Kategóriák.



16. ábra: Thunderclient teszt Felhasználó regisztráció



17.ábra: Thunderclient teszt Felhasználó törlése.

3.Felhasználói dokumentáció

3.1 Felhasználói fiók

3.1.1 Regisztráció

A Dör Községi Könyvtár oldalon annak érdekében,hogy a felhasználó böngészni tudjon a katalógusból először regisztrálnia kell. Az oldal látogatását követően a navigáción a jobb felső sarokban a “Regisztráció” gombra kattintva lesz erre lehetőség.



18.ábra: Navigációs sáv.

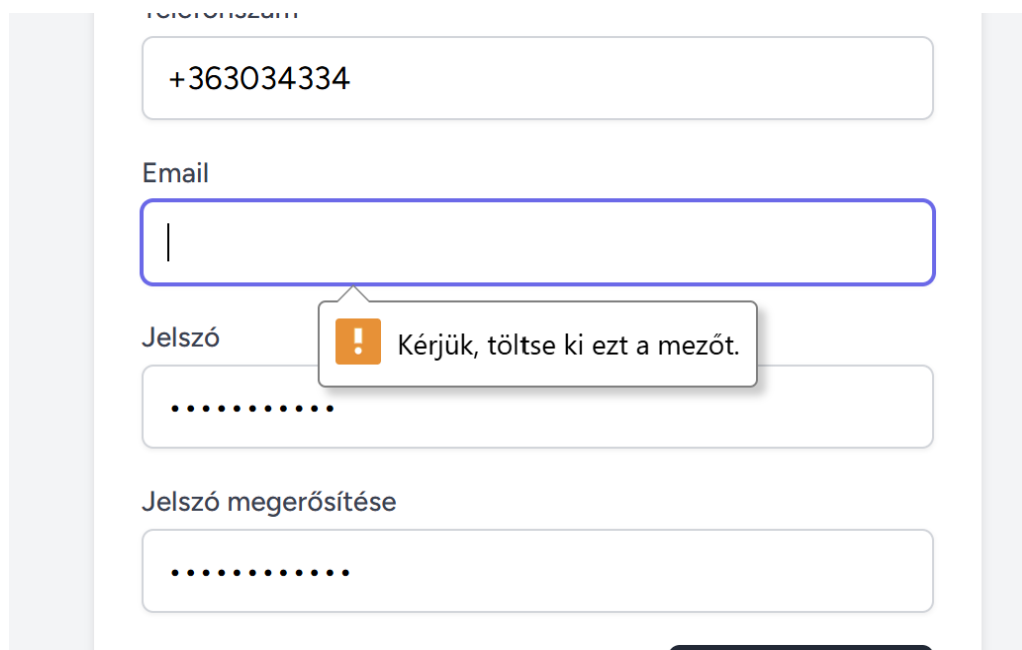
Esetleges hibák elkerülése érdekében az úrlapon hibakezelés működik, és amennyiben a felhasználó valamit hibát ejt, a rendszer az azt megfelelő hibaüzenettel fogja jelezni.

Regisztrációhoz szükséges adatok a következők:

- Felhasználónév
- Vezetéknév
- Keresztnév
- Születési hely
- Születési idő
- Telefonszám
- Email cím
- Jelszó
- Jelszó megerősítése

Az alábbi két képen bemutatjuk két rosszul kitöltött űrlapot, és az annak megfelelő hibaüzeneteket.

Az első képen az látható, amikor valamilyen mezőt a felhasználó nem töltött ki, ebben az esetben a rendszer jelzi, hogy ez nem lett kitöltve, és ameddig ez nincs pótolva a regisztráció nem lesz sikeres.

The image shows a portion of a registration form. It includes a 'Telefonszám' (Phone number) field with the value '+363034334'. Below it is an 'Email' field, which is empty and has a red border around it. Under the email field is a 'Jelszó' (Password) field with masked characters (dots). Below the password field is a 'Jelszó megerősítése' (Confirm password) field, also with masked characters. A tooltip with an orange exclamation mark icon and the text 'Kérjük, töltse ki ezt a mezőt.' (Please fill out this field.) points to the empty email field.

19.ábra: Regisztráció email cím nincs megadva.

A második képen az látható, amin a felhasználó az adatok megadása során két hibát is vét, egyik esetben a telefonszám hossza, másik esetben pedig a jelszavak nem egyeztek meg, ezeket a rendszer mindkét esetben a megfelelő hibaüzenettel jelezte a felhasználónak.

+363034334

A telefonszám legalább 12 karakter hosszú kell legyen (pl. +36 20 1234567).

Email

tesztelod@jedlik.eu

Jelszó

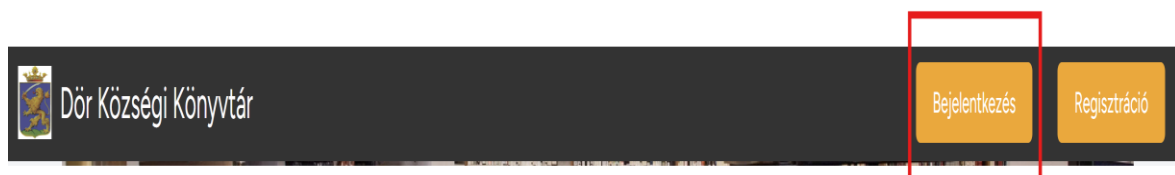
A jelszó megerősítése nem egyezik.

Jelszó megerősítése

20. ábra: Regisztráció telefonszám/jelszó hibásan megadva.

3.1.2 Bejelentkezés

A weboldalra való belépéshez a főoldalon a navbar-on a szintén jobb felső sarokban a "Bejelentkezés" gombra kattintva lehetséges.



21. ábra: Navigációs sáv.

A felhasználó bejelentkezéséhez szükséges adatok a következők:

- Email cím
- Jelszó

Email

Jelszó

[Nincs még fiókja?](#)

BELÉPÉS

22. ábra: Bejelentkezés.

Amennyiben a felhasználó még nem rendelkezik saját profillal, a “Nincs még fiókja?” linkre kattintva a rendszer átirányítja a regisztrációs űrlapra ahol elvégezheti profilja regisztrálását.

A könyvtár katalógusa megtekintése regisztrációhoz kötött, így amennyiben a felhasználó egyből a bejelentkezés gombra kattint az alábbi képen látott hibaüzenet fogja majd fogadni:

Email

Jelszó



Kérjük, töltsse ki ezt a mezőt.

[Nincs még fiókja?](#)

BELÉ

23. ábra: Bejelentkezés email mező nincs kitöltve.

Amennyiben a megadott adatok nem találhatók az adatbázisban (vagy akár csak elírás történt) a következő hibaüzenetet kapjuk:

Email

nemletezik@jedlik.eu

Rossz email-jelszó páros.

Jelszó

[Nincs még fiókja?](#) **BELÉPÉS**

24. ábra: Bejelentkezés rossz email/jelszó páros.

3.1.3 Regisztrált, valamint már bejelentkezett felhasználók/admin számára elérhető funkciók

A Dör Községi Könyvtár funkciói a bejelentkezést követően válik a felhasználók számára elérhetővé. Amint belépnek a főoldalon már a katalógus várja a felhasználókat, ezen felül a navbar-ban új menüpontok, a "Kölcsönzések", "Profil", "Kijelentkezés" menüpontok oldódnak fel.

Dör Községi Könyvtár

Kölcsönzések Profil Kijelentkezés

Rólunk

A Dör Könyvtár egy titokzatos hely, ahol a könyvek suttogása és az idő lassú ketyegése hatja át a falakat. Az oldalak poros, elfeledett titkokat rejtenek, és minden sarkon egy új világ vár, hogy elnyeljen. Itt az olvasó nem csupán könyveket talál, hanem elveszett emlékeket és eltévedt gondolatokat is. A könyvtár nem csupán egy tér, hanem egy szürreális utazás a tudás sötét zugaiba, ahol minden könyv egy újabb kérdést hagy maga után. Várunk, hogy elmerülj a valóság és a képzelet határán.

Könyvtárunkban megtalálható tartalom:

Minden kategória Cím Szerző Kiadás éve

"Mindent apámról..." : Egy sokoráljai iparos emlékei
Szerző: Lanczendorfer Zsuzsanna
Kiadás éve: 2007

Arthur király : A sötét kor harcosa és legendás hőse
Szerző: Matthews, John
Kiadás éve: 2004

A kereszties hadjáratok : A kétszáz éves háború
Szerző: Harpur, James
Kiadás éve: 2005

Tükörkép
Szerző: Steel, Danielle
Kiadás éve: 2006
Kategória: könyv

25. ábra: Főoldal.

3.1.4 Főoldal (Katalógus)

Belépést követően egyből a főoldalra, azaz a katalógusra irányít minket a rendszer. Itt a könyvtár teljes elérhető tartalmát megtekinthetjük, akár egyesével, vagy akár a beépített szűrők segítségével a keresett tárgyra is kereshetünk.

Szűrhetünk:

- Kategóriára
- Címre
- Szerzőre
- Kiadási évre

Könyvtárunkban megtalálható tartalom:

Minden kategória

Cím

Szerző

Kiadás éve

"Mindent apámról..." : Egy sokoróaljai iparos emlékei Szerző: Lanczendorfer Zsuzsanna Kiadás éve: 2007 Kategória: könyv	Arthur király : A sötét kor harcosa és legendás hőse Szerző: Matthews, John Kiadás éve: 2004 Kategória: könyv	A keresztes hadjáratok : A kétszáz éves háború Szerző: Harpur, James Kiadás éve: 2005 Kategória: könyv	Tükörkép Szerző: Steel, Danielle Kiadás éve: 2006 Kategória: könyv
Az otthoni gyógymódok nagykönyve Szerző: Frank Zsófia Kiadás éve: 2007 Kategória: könyv	Az ötödik gyerek Szerző: Lessing, Doris Kiadás éve: 2007 Kategória: könyv	Biztonságos kikötő Szerző: Roberts, Nora Kiadás éve: 2007 Kategória: könyv	Utazás Szerző: Steel, Danielle Kiadás éve: 2001 Kategória: könyv
A paradicsomban is ott a pokol Szerző: Remarque, Erich Maria Kiadás éve: 2007 Kategória: könyv	Táltosmackó Szerző: Donászy Magda Kiadás éve: 2007 Kategória: könyv	A zarándoklat : Egy mágus naplója Szerző: Coelho, Paulo Kiadás éve: 2006 Kategória: könyv	Hekja : Egy lány a vikingek fogságában Szerző: French, Jackie Kiadás éve: 2007 Kategória: könyv
Fétis Szerző: Moss, Tara Kiadás éve: 2008 Kategória: könyv	Bogar bárd meséi Szerző: Rowling, Joanne Kathleen Kiadás éve: 2008 Kategória: könyv	Búcsú Liverpooltól Szerző: Lee, Maureen Kiadás éve: 2008 Kategória: könyv	

Első

Előző

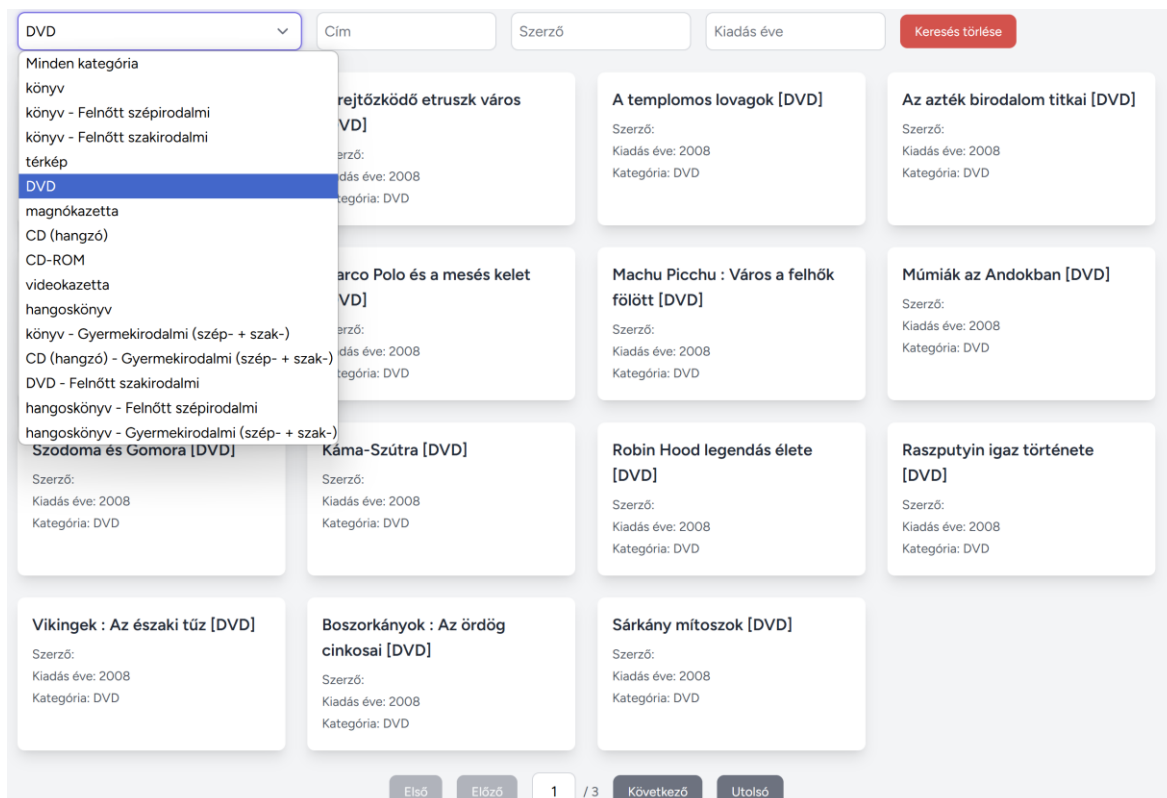
1 / 323

Következő

Utolsó

26. ábra: Katalógus.

Keresni a kategóriák között, egy lenyíló menü segítségével tudunk, a könyvtár rendszerében elérhető összes kategóriát itt láthatjuk, és kiválasztásával megjelennek az adott kategórián belül elérhető itemek.

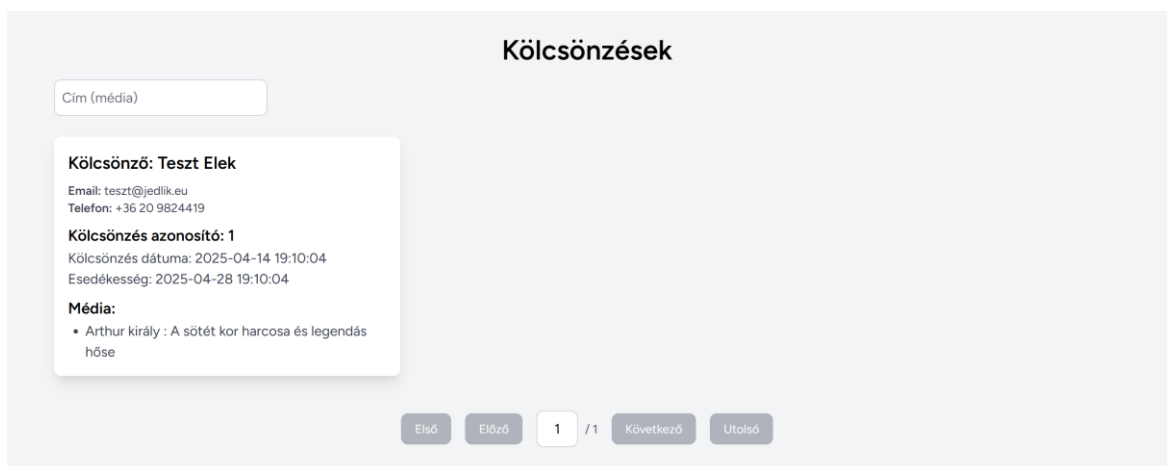


27. Ábra: Katalógus szűrési lehetőségek.

További szűrés is lehetséges, a Cím, Szerző, Kiadási év kitöltése által.

3.1.5 Kölcsönzések

Ezen a menüponton belül a felhasználó megtekintheti az eddigi kölcsönzéseit, és azok státuszát. A kölcsönzések között a cím alapján szűrés is lehetséges.



28. ábra: Aktuális kölcsönzések.

Az aktuális kölcsönzés a Katalóguson (Főoldalon) is látszik.



29. ábra: Katalógusban a kikölcsönzött item.

3.1.6 Profil

A "Profil" menüpontra kattintva megjelennek a felhasználó által megadott adatok. A felhasználóknak ezen az oldalon lesz lehetőségük a korábban megadott adataik módosítására, mint például, vezetéknév, keresztnév, telefonszám, születési idő, illetőleg itt lesz lehetőség is majd a profil végleges törlésére is.

Profil információk

Frissítse a fiókjának profilinformációit.

Felhasználónév

teszt

Vezetéknév

Teszt

Keresztnév

Elek

Születési hely

Győr

Születési idő

2004. 01. 15.



Telefonszám

+36 20 9824419

Email

teszt@jedlik.eu

MENTÉS

30. ábra: Profil.

Jelszó frissítése

Győződj meg róla, hogy a fiókod egy hosszú, véletlenszerű jelszót használ a biztonság érdekében.

Jelenlegi jelszó

Új jelszó

Jelszó megerősítése

MENTÉS

Fiók törlése

Miután a fiókotat törölted, annak összes erőforrása és adata véglegesen törlődik.

FIÓK TÖRLÉSE

31. ábra: Profilon jelszó frissítés/fiók törlése.

3.2 Admin jogosultság lehetőségei

Az admin jogosultsággal számos plusz opció, és funkció nyílik meg, ugyanakkor alapjaiban a design nem tér el, a felhasználóéhoz képest. A navbar-on plusz menüpontok válnak elérhetővé:

- Tartalom hozzáadás
- Felhasználók

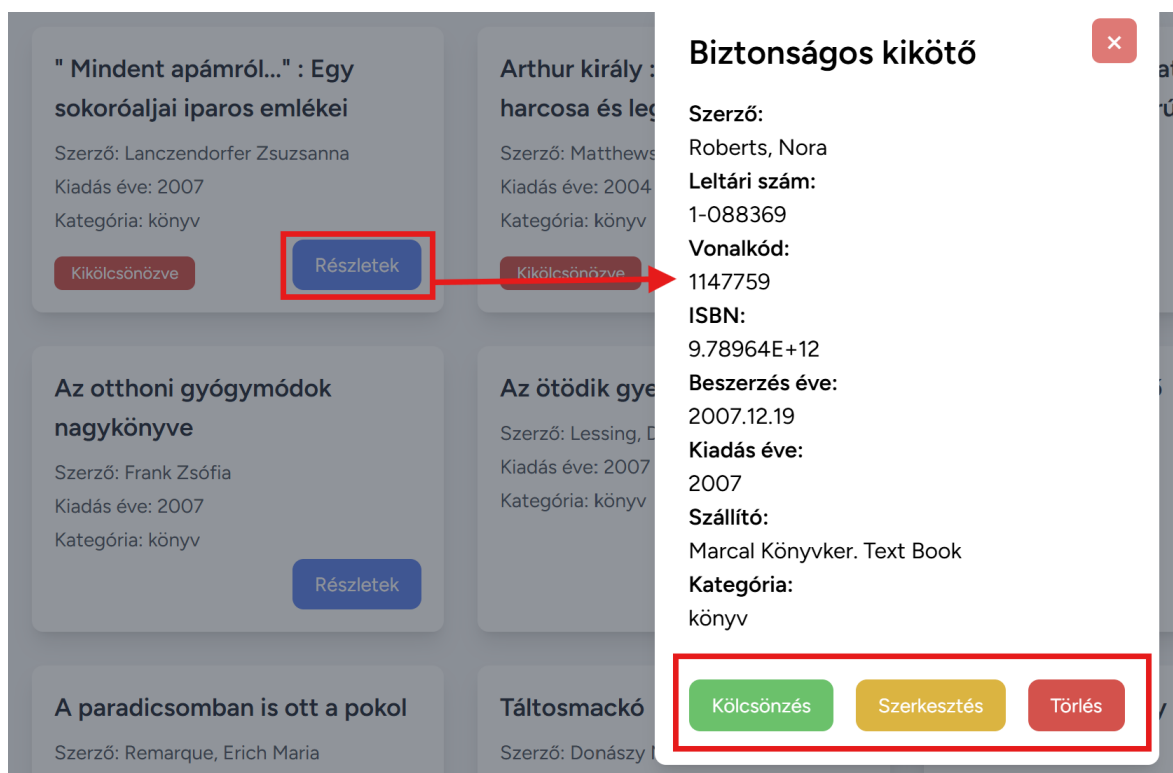
Az admin jogkörei:

- Tartalmakat vehet fel, szerkeszthet, és törölhet a katalógusból

- Nyomon követheti a felhasználók kölcsönzéseit, és törölheti azokat
- Manageheti a már regisztrált felhasználókat.

3.2.1 Tartalom módosítás, kölcsönzés a katalógusban admin jogkörrel

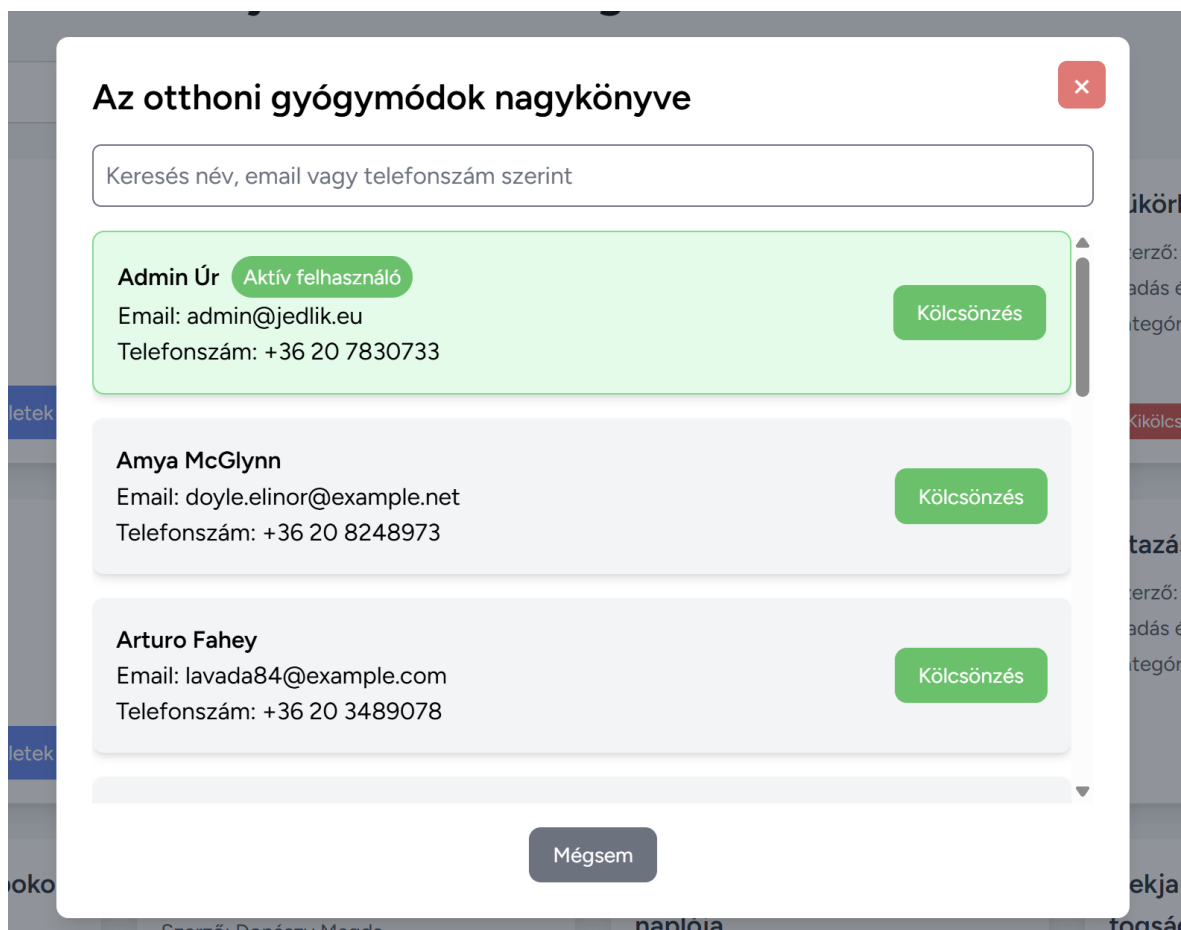
Az admin módosíthatja a katalógus tartalmát, törölhet, szerkeszthet és akár új tételekkel is bővítheti azt. A katalógus(Főoldal) bármelyik card-on a részletek gombra kattintva felugrik egy ablak lentebb három gombbal, a szerkesztés, kölcsönzés, és a törlés.



32. ábra: Admin jogosultság, katalógusban részletek fül.

A Szerkesztésre kattintva a fentebb prezentált kép kategóriái alapján szerkeszthetővé válik az adatbázisban.

A Kölcsönzésre kattintva előugrik egy ablak, melyen az összes regisztrált felhasználó megtalálható, és a kívánt felhasználóhoz lehet rendelni a kölcsönzést. Szűrésre itt is van lehetőség, név, email, vagy telefonszám alapján



33. ábra: Admin aktuális kölcsönzések.

3.2.2 Tartalom felvétel

Az admin jogkörrel lesz lehetőségünk a könyvtár katalógusának a bővítésére. A navbar-on „a „Tartalom hozzáadása” gombra kattintva lesz lehetőség új tartalom hozzáadására.

Új tartalom hozzáadásakor szükséges minden mezőt kitölteni. A kitöltendő mezők a következők:

- Cím
- Szerző
- Leltári Szám
- ISBN
- Vonalkód
- Beszerzés éve
- Kiadás éve

- Szállító
- Kategória

The image shows a web form for adding book content. It consists of several input fields stacked vertically, each with a label to its left. The labels are: 'Szerző' (Author), 'Leltári szám' (Inventory number), 'Vonalkód' (Barcode), 'ISBN', 'Beszerzési éve' (Purchase year), 'Kiadás éve' (Release year), 'Szállító' (Supplier), and 'Kategória' (Category). The 'Beszerzési éve' field contains the placeholder text 'éééé. hh. nn.' and a small calendar icon. The 'Kategória' field is a dropdown menu showing '-- Kategória választása --'. Below these fields is a prominent blue button labeled 'Hozzáadás' (Add).

34. ábra: Admin tartalom felvétel.

3.2.3 Felhasználók

Admin jogkörrel a navbaron az egyik plusz menüpont a “Felhasználók”. Erre a gombra kattintva a rendszer átirányít minket, majd ezen az oldalon láthatóak a regisztrált felhasználók. Az admin itt lesz majd képes monitorozni a felhasználókat.

Felhasználók

<p>Admin Úr</p> <p>Saját felhasználó Admin</p> <p>Felhasználónév: admin Email: admin@jedlik.eu Telefonszám: +36 20 7830733 Születési hely: Kunzefurt Születési dátum: 1997-01-06</p>	<p>Amya McGlynn</p> <p>Felhasználónév: jess.gutmann Email: doyle.elinor@example.net Telefonszám: +36 20 8248973 Születési hely: North Noe Születési dátum: 2002-01-20</p> <p style="text-align: right;">Törlés</p>	<p>Arturo Fahey</p> <p>Felhasználónév: schmeler.linda Email: lavada84@example.com Telefonszám: +36 20 3489078 Születési hely: Raushire Születési dátum: 1996-04-29</p> <p style="text-align: right;">Törlés</p>	<p>Cade Kassulke</p> <p>Felhasználónév: michaela.hammes Email: cummerata.harvey@example.net Telefonszám: +36 20 5042654 Születési hely: Jairoborough Születési dátum: 1987-11-29</p> <p style="text-align: right;">Törlés</p>	<p>Elda Adams</p> <p>Felhasználónév: mwillms Email: leanna.steuber@example.com Telefonszám: +36 20 9899465 Születési hely: North Teresafort Születési dátum: 2010-05-20</p> <p style="text-align: right;">Törlés</p>
<p>Ellis Pfeffer</p> <p>Felhasználónév: francisco.keebler Email: mabelle.streich@example.org Telefonszám: +36 20 2722092 Születési hely: Ameliaville Születési dátum: 1972-12-20</p> <p style="text-align: right;">Törlés</p>	<p>Felipa Smith</p> <p>Felhasználónév: hermiston.salvador Email: critchle@example.com Telefonszám: +36 20 3721226 Születési hely: South Sincere Születési dátum: 1997-03-01</p> <p style="text-align: right;">Törlés</p>	<p>Kamille Corkery</p> <p>Felhasználónév: brakus.adah Email: schiller.ignatius@example.org Telefonszám: +36 20 1928896 Születési hely: South Macey Születési dátum: 1995-04-10</p> <p style="text-align: right;">Törlés</p>	<p>Laurianne Kihn</p> <p>Felhasználónév: ora03 Email: brakus.horace@example.com Telefonszám: +36 20 2929352 Születési hely: Reingertown Születési dátum: 1999-07-26</p> <p style="text-align: right;">Törlés</p>	<p>Lexi Murazik</p> <p>Felhasználónév: dena26 Email: jonn@example.net Telefonszám: +36 20 6301063 Születési hely: North Karlee Születési dátum: 2014-06-19</p> <p style="text-align: right;">Törlés</p>

35. ábra: Admin Regisztrált felhasználók.

Az admin munkájának megkönnyítése érdekében, hogy az esetleges több száz felhasználó között gyorsan tudjon keresni különböző szűrési opciókat implementáltunk a rendszerbe, melyek a következők:

- Teljes név
- Email
- Felhasználónév
- Telefonszám
- Születési hely
- Születési idő
- Szerepkör

Az alábbi képen Teljes névre szűrtünk.

Teszt

Email

Felhasználónév

Telefonszám

Születési Hely

éééé. hh. nn.

Szerepkör (mind)

Keresés törlése

Teszt Elek

Felhasználónév: teszt
Email: teszt@jedlik.eu
Telefonszám: +36 20 9824419
Születési hely: Győr
Születési dátum: 2004-01-15

Törölés

36. ábra: Admin felhasználók közötti szűrés.

4. Összefoglalás

A szakdolgozatunk célja tehát az volt, hogy a felhasználók, és a könyvtárban dolgozó személyzet részére egy könnyen átlátható, skálázható alkalmazást hozzunk létre mely megoldást nyújt a könyvtár jelenlegi rendszer problémáira.

A tervezési fázisban megfogalmaztunk néhány lehetséges Use Case-t, szerepköröket, elosztottuk egymás közt a ránk váró feladatokat. Az év elején megismerkedtünk a Vue, és a Laravel keretrendszerekkel, amit annyira megkedveltünk, hogy nem volt kérdéses a továbbiakban, hogy a fejlesztés során ezeket a technológiákat fogjuk majd alkalmazni. A fejlesztés során továbbá egy standup jellegű prezentációs formát használtunk, ahol mindenki röviden ismertette az általa fejlesztett részt, és a végén meghatároztuk a jövőbeli célokat. Ezzel a módszerrel a szoftverfejlesztésben megszokott napi standup-megbeszélések hangulatát és struktúráját idéztük meg.

A fejlesztés során számos kihívással találkoztunk, amelyek megoldásával és különböző témákban végzett kutatómunkákkal voltunk képesek elmélyíteni szakmai tudásunk.

Összességében elmondható, hogy elégedettek vagyunk az elért eredményekkel, úgy gondoljuk sikeresen képesek voltunk megvalósítani elképzeléseinket. Mivel a program számos továbbfejlesztési lehetőséget rejt magában, elképzelhetőnek tartjuk, hogy a jövőben számos plusz funkcióval bővítjük azt.

5. Idézett források

<https://www.monterail.com/blog/software-qa-standards-iso-25010> - 2025.01.05.

<https://www.figma.com/resource-library/what-is-a-use-case/> - 2025.01.10.

<https://tadabase.io/blog/html-css-javascript-explained> - 2025.01.25.

<https://aws.amazon.com/what-is/ide/> - 2025.01.25.

<https://www.prooktatas.hu/hirek/laravel,-a-legjobb-php-keretrendszer> - 2025.02.02.

<https://vuejs.org/guide/introduction> - 2025.02.10.

<https://builtin.com/data-science/sqlite> - 2025.02.25.

<https://laravel.com/docs/11.x/database> - 2025.03.10.

<https://medium.com/mindorks/what-is-git-commit-push-pull-log-aliases-fetch-config-clone-56bc52a3601c> - 2025.03.20.

<https://bluebird.hu/frontend-backend-developer-kulonbseg/> - 2025.03.26.

<https://tailwindcss.com/docs/installation/framework-guides> - 2025.04.01.

<https://peggi.hu/http-protokoll/> - 2025.04.01.

<https://laravel.com/docs/11.x/sanctum#main-content> - 2025.04.05.

docs.thunderclient.com - 2025.04.10.