**Váci Szakképzési Centrum  
Boronkay György  
Műszaki Technikum és Gimnázium**

**VIZSGAREMEK**

**Bátyi Bence**

**Horti-Nagy Bálint Máté**

**Novák Márton**

**2025**

**Váci Szakképzési Centrum  
Boronkay György  
Műszaki Technikum és Gimnázium**

****

**VIZSGAREMEK**

**BookHive**

**Készítette: Bátyi Bence**

**Horti-Nagy Bálint Máté**

**Novák Márton**

# Hallgatói nyilatkozat

Alulírottak, ezúton kijelentjük, hogy a vizsgaremek saját, önálló munkánk, és

korábban még sehol nem került publikálásra.

|  |  |  |
| --- | --- | --- |
| **Bátyi Bence** | **Horti-Nagy Bálint Máté** | **Novák Márton** |

Tartalomjegyzék

[Hallgatói nyilatkozat 3](#_Toc195040700)

[Fejlesztői dokumentáció 7](#_Toc195040701)

[Fejlesztői környezet 7](#_Toc195040702)

[Visual StudioCode 7](#_Toc195040703)

[Visual Studio 7](#_Toc195040704)

[phpMyAdmin 7](#_Toc195040705)

[Használt programozási nyelvek és technológiák 7](#_Toc195040706)

[PHP 7](#_Toc195040707)

[PHPMailer 7](#_Toc195040708)

[JavaScript 7](#_Toc195040709)

[JQuery 7](#_Toc195040710)

[CSS 7](#_Toc195040711)

[C# 7](#_Toc195040712)

[MySQL 7](#_Toc195040713)

[ChatGPT 7](#_Toc195040714)

[Asztali alkalmazás fejlesztői dokumentáció 8](#_Toc195040715)

[Adatbázis/API kommunikáció 8](#_Toc195040716)

[Kinézeti elemek 9](#_Toc195040717)

[Bejelentkezés 9](#_Toc195040718)

[Főoldal 9](#_Toc195040719)

[Részletes formok 10](#_Toc195040720)

[Hozzáadós formok 11](#_Toc195040721)

[Visszavevő form 11](#_Toc195040722)

[Alkalmazottak 11](#_Toc195040723)

[Kilistázó formok 11](#_Toc195040724)

[Popupselect 13](#_Toc195040725)

[Weboldal fejlesztői dokumentáció 13](#_Toc195040726)

[Alap tudnivalók a weboldal működéséről: 13](#_Toc195040727)

[Bejelentkezés és hozzákapcsolódó részek: 13](#_Toc195040728)

[Index.php-n történő folyamatok egy oldal megnyitásakor: 13](#_Toc195040729)

[Jelszó módosítás működése: 14](#_Toc195040730)

[Szerepkörök 15](#_Toc195040731)

[Weboldalon azonosítatlan felhasználó (vendég): 15](#_Toc195040732)

[Weboldalon regisztrált felhasználó: 15](#_Toc195040733)

[Weboldalon regisztrált felhasználó lejárt taggsággal: 15](#_Toc195040734)

[Asztali alkalmazásban normál dolgozói profil: 15](#_Toc195040735)

[Asztali alkalmazásban admin dolgozói profil: 16](#_Toc195040736)

[API dokumentáció 17](#_Toc195040737)

[API endpoints: 17](#_Toc195040738)

[Könyv katalógus lekérése (opcionálisan) szűrve: 17](#_Toc195040739)

[Felhasználó bejelentkezés: 17](#_Toc195040740)

[Jelszó módosítása a profil oldalról: 17](#_Toc195040741)

[Kivett könyv lejárati dátumának meghosszabbítása: 18](#_Toc195040742)

[Felhasználó bejelentkezve van-e?: 18](#_Toc195040743)

[Kijelentkezés: 18](#_Toc195040744)

[“Elfelejtettem a jelszavam” link kérés: 18](#_Toc195040745)

[Könyv levétele kedvencek közül: 18](#_Toc195040746)

[Könyv felvétele kedvencekbe: 19](#_Toc195040747)

[Jelszó módosítása az emaliben kapott linken keresztül (“elfelejtettem a jelszavamat” funkció): 19](#_Toc195040748)

[Alkalmazott bejelentkeztetése: 19](#_Toc195040749)

[Adatbázis 20](#_Toc195040750)

[Kapcsolatok 21](#_Toc195040751)

[Táblák 22](#_Toc195040752)

[A „book” tábla 22](#_Toc195040753)

[A „languages” tábla 23](#_Toc195040754)

[A „publishers” tábla 23](#_Toc195040755)

[A „genres” tábla 24](#_Toc195040756)

[A „books\_genres\_conn” tábla 24](#_Toc195040757)

[Az „authors” tábla 25](#_Toc195040758)

[A „books\_authors\_conn” tábla 25](#_Toc195040759)

[Az „inventory” tábla 26](#_Toc195040760)

[A „users” tábla 27](#_Toc195040761)

[A „reservation” tábla 28](#_Toc195040762)

[A „shelf” tábla 28](#_Toc195040763)

[Az „employees” tábla 30](#_Toc195040764)

[A „borrowings” tábla 31](#_Toc195040765)

[A „booking” tábla 32](#_Toc195040766)

[A „system\_settings” tábla 33](#_Toc195040767)

[A „tokens” tábla 34](#_Toc195040768)

[Eljárások 35](#_Toc195040769)

[Eljárások felsorolása 35](#_Toc195040770)

[Tesztelés 46](#_Toc195040771)

[SQL eljárások tesztelése 46](#_Toc195040772)

[Adatbázis tesztadatok 46](#_Toc195040773)

[Api asztali alkalmazással való összeköttetésének integrációs tesztje 46](#_Toc195040774)

[Asztali alkalmazás rendszertesztek 46](#_Toc195040775)

[Felhasználói dokumentáció 47](#_Toc195040776)

[Asztali alkalmazás felhasználói dokumentáció 47](#_Toc195040777)

[Belépés 47](#_Toc195040778)

[Főoldal 48](#_Toc195040779)

[Dolgozó adatai oldal 49](#_Toc195040780)

[Könyvek oldal 51](#_Toc195040781)

[Felhasználók oldal 55](#_Toc195040782)

[Kivett könyvek oldal 57](#_Toc195040783)

[Foglalások oldal 57](#_Toc195040784)

[Előjegyzés oldal 57](#_Toc195040785)

[Kiadók oldal 58](#_Toc195040786)

[WEB felhasználói dokumentáció 60](#_Toc195040787)

[Belépés 60](#_Toc195040788)

[Navigációs menü 61](#_Toc195040789)

[Menü – asztali nézet: 61](#_Toc195040790)

[Menü – mobil nézet: 61](#_Toc195040791)

[Navigálás az oldalak között a menü segítségével 62](#_Toc195040792)

[Oldalak 62](#_Toc195040793)

[Főoldal 62](#_Toc195040794)

[Könyveink oldal 62](#_Toc195040795)

[Könyv részletes oldal 63](#_Toc195040796)

[Profil részletei oldal 64](#_Toc195040797)

[Rólunk oldal 65](#_Toc195040798)

[Irodalomjegyzék 66](#_Toc195040799)

[Forráskód 66](#_Toc195040800)

# Fejlesztői dokumentáció

## Fejlesztői környezet

### Visual StudioCode

A teljes weboldal, az api és a backend is Visual StudioCode-ban volt írva elsősorban egyszerű elérhetősége és kezelhetősége, GitHub integrációja és beépített terminálja miatt. Az egész projekt egy PHP szerveren fut, amit a terminálból indítottunk el. Ennek köszönhetően egy ablakban tudtunk dolgozni a backend-en és látni a beérkező GET és POST kéréseket és azok hibáit.

### Visual Studio

Az asztali alkalmazás Visual Studio-ban volt fejlesztve, mivel itt a legegyszerűbb a Windows Forms applikációk fejlesztése és a C# nyelv használata. A debug funkciói nagyon hasznosak voltak és sokat segítettek a fejlesztés során

### phpMyAdmin

Az SQL eljárások írása és az adatbázis menedzselése phpMyAdmin-ban történt. Ezek mellet a könnyen kezelhető és átlátható elrendezése miatt egyszerűen lehetett keresni és módosítani az adatbázisban tesztelések során.

Programozási nyelvek, tehnológiák, könyvtárak

## Használt programozási nyelvek és technológiák

### PHP

Az összes weboldal, az api és a weboldal is PHP nyelven íródott, mivel ugyanolyan hasznos a szerver oldalon, mint a kliens oldalon.

### PHPMailer

Ezt a PHP könyvtárat email küldés véget alkalmaztuk. Ezzel küldjük a jelszóváltoztató és a regisztráció visszaigazoló email-eket. Azért választottuk a PHPMailer-t, mert a PHP-ba beépített email küldő funkció nem megbízható.

### JavaScript

A kliens oldali számításokat és a weboldal animációit ezzel oldottuk meg

### JQuery

Kizárólag az API kommunikációra volt használva

### CSS

Egyszerű CSS-el lett megoldva a weboldal arculata, semmilyen könyvtár segítsége nélkül.

### C#

Az asztali alkalmazás egy Windows Forms applikáció, ami a C# nyelvet használja

### MySQL

A rendszer mögötti adatbázis MySQL-t használ. Az összes hozzá tartozó eljárás is ebben a nyelvben lett írva.

### ChatGPT

A teszteléshez az adatbázis tesztadatait ChatGPT segítségével voltak generálva.

## Asztali alkalmazás fejlesztői dokumentáció

Az alkalmazás egy .NET framework alapú Windows Forms applikáció.

### Adatbázis/API kommunikáció

Az asztali alkalmazást az alkalmazottak használják az adatbázisban szereplő adatok kezelésére, így egy egyszerű gyors és biztonságos módot kell használni az adatbázissal való kommunikációra. Erre a feladatra az API-t használunk, ami az összes lekérdezést, hozzáadást, módosítást és törlést elvégzi, ami az asztali alkalmazásnak kell. Az API-val való kommunikációért az „ApiComm.cs” osztály végzi, aminek csak egyetlen statikus „SendPost” metódusa van.

using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

usingSystem.Net.Http;

usingSystem.Windows.Forms;

usingSystem.ComponentModel;

usingSystem.Web.Script.Serialization;

namespaceDesktop

{

internalclassApiComm

{

publicstaticasyncTask<object>SendPost(Dictionary<string, string>args)

{

varclient = newHttpClient();

//var content = newFormUrlEncodedContent(args);

stringparameters = "";

foreach(KeyValuePair<string, string> line inargs)

{

parameters += $"{line.Key}={line.Value}&";

}

parameters.Remove(parameters.Length - 1);

varcontent = newStringContent(parameters, Encoding.UTF8, "application/x-www-form-urlencoded");

varresponse = awaitclient.PostAsync("http://localhost:8000/BACKEND/api.php", content);

varresponseString = awaitresponse.Content.ReadAsStringAsync();

JavaScriptSerializerserializer = newJavaScriptSerializer();

varresponseDir = serializer.Deserialize(responseString, typeof(List<Dictionary<string, string>>));

client.Dispose();

returnresponseDir;

}

}

}

Ez a metódus egy „Dictionary<string, string>” típusú szótárat kér argumentumként, ami a POST kérés változóit és azok értékeit tartalmazza. A kérés megformálásáért NEM a FormUrlEncodedContent metódus felelős, mert kép feltöltése esetén túl nagyok voltak a paraméterek. Ennek orvoslása végett a POST request-et a metódus rakja össze magának, így csak a XAMPP beállításai korlátozzák egy POST kérés méreteit. A kérés formátuma és visszatérési értékei a fejlesztői dokumentáció adatbázis részénél található. Választ JSON formátumban kap, amit dekódol és visszaküldi „List<Dictionary<string,string>> ” típusú formátumban.

### Kinézeti elemek

Az alkalmazásban található összes bemeneti mező saját fejlesztés nem az alapértelmezett verzió. Elsősorban a kinézet testreszabhatósága végett készültek ezek az elemek.

### Bejelentkezés

A program indításakor a „loginForm.cs” lesz meghívva a felhasználó bejelentkeztetése mellet ez tárolja a bejelentkezett alkalmazott adatait, illetve az alkalmazás MainFrom-jának a referenciáját is itt lehet találni.

### Főoldal

Bejelentkezést követően a „MainForm.cs” mindig látható lesz, itt található a menü és a többi form is ezen belül fog megnyílni az „OpenChildForm” metódus segítségével.

publicvoidOpenChildForm(FormchildForm)

{

if(activeForm != null)

{

if(activeForm == usersPage || activeForm == allBooksPage)

{

activeForm.Hide();

}

else

{

activeForm.Close();

}

}

prev = activeForm;

roundedCornerPanel\_PageContainer.Controls.Remove(activeForm);

activeForm = childForm;

childForm.TopLevel = false;

childForm.FormBorderStyle = FormBorderStyle.None;

childForm.Dock = DockStyle.Fill;

roundedCornerPanel\_PageContainer.Controls.Add(childForm);

roundedCornerPanel\_PageContainer.Tag = childForm;

ThemeManager.ApplyThemeToAllForms();

childForm.BringToFront();

childForm.Show();

childForm.Visible = true;

}

A felhasználói élmény növelésének érdekében a leggyakrabban használt kilistázó form-okat nem bezárjuk, hanem csak eltüntetjük, hogy a keresési szűrő megmaradjon és így kényelmesebb legyen a használat. Erről az oldalról indítjuk automatikus funkcióikat is, például a lejárt foglalások küldését vagy az új foglalások keresését. Ezek a törlések minden indításkor futnak le és csak egy egyszerű api hívás. Ilyenkor indul el az automatikus „CheckForBookings” metódus is, ami a „bookings” táblában levő változásokat keresi 15 percenként.

publicasyncTaskCheckForBookings(boolnow)

{

if(!now)

{

while(true)

{

List<Dictionary<string, string>>resp = (List<Dictionary<string, string>>)awaitApiComm.SendPost(newDictionary<string, string>{{"type", "checkForBookings"}});

if(resp.First()["result"] == "true")

{

pbNewBookings.Visible = true;

}

awaitTask.Delay(TimeSpan.FromMinutes(15));

}

}

else

{

List<Dictionary<string, string>>resp = (List<Dictionary<string, string>>)awaitApiComm.SendPost(newDictionary<string, string>{{"type", "checkForBookings"}});

if(resp.First()["result"] == "true")

{

pbNewBookings.Visible = true;

}

}

}

Két módja van, ha a „now” paraméter hamis, akkor fogja a fix intervallumonkénti keresést indítani. Ha igaz a paraméter akkor a jelen pillanatban fogja megnézni, hogy történt-e változás.

### Részletes formok

privateasyncvoid Form1\_Load(objectsender, EventArgs e)

{

OpenChildForm(newHomePage());

lblEmplUname.Text = LoginForm.empl\_uname;

List<Dictionary<string, string>>temp = (List<Dictionary<string, string>>)awaitApiComm.SendPost(newDictionary<string, string>{{"type", "getSystemSettings"}});

system\_settings = temp.First();

CheckForBookings(false);

List<Dictionary<string, string>>reservationResult = (List<Dictionary<string, string>>)awaitApiComm.SendPost(newDictionary<string, string>(){{"type", "deleteExpiredReservations"}});

List<Dictionary<string, string>>bookingResult = (List<Dictionary<string, string>>)awaitApiComm.SendPost(newDictionary<string, string>(){{"type", "deleteExpiredBookings"}});

}

Két részletes oldal van „bookDetailedPage.cs” és „userDetailedPage” itt. Ezeken az oldalakon az adott tábla összes adatát módosítani lehet, illetve extra műveleteket is lehet az oldalakon csinálni (pl.: egy felhasználó részletes oldalán a tagságát lehet meghosszabítani). A form érzékeli, ha változás történt az adatokban úgy, hogy eltárolja az eredeti betöltött adatokat, és ha különbözik a jelenlegi adatoktól akkor történt változás. Figyelmeztet is ha be szeretnénk zárni az adott részletes oldalt. A „bookDetailedPage.cs” is egyedi, mivel pontosan úgy, mint a „PopupSelect.cs” két különböző módban indítható el. Az elsőben, ahogy már említve volt módosítani lehet egy könyvet, a második módjában könyvtípust lehet vele hozzáadni.

### Hozzáadós formok

Két hozzáadós form fajta van, mindegyiknek egyetlen lényege, hogy egy új példányt adjanak hozzá a megfelelő táblához. A kétfajta között a legfőbb különbség a kinézet. Az első fajtába tartozik az „addEmployee.cs”, „addPublisher.cs” és az „addUser.cs”. Egyszerű hozzáadási formok bemeneti mezőkkel. A másik fajta pedig a „bookLendingPage.cs”, amivel előjegyezni és kivenni lehet könyveket. Két gomb két popupselect-et nyit meg amikkel ki kell választani a könyvet és az olvasót. Itt az dönti el, hogy előjegyzést vagy kivételt add hozzá, hogy melyik form hívja meg és milyen paraméterrel, szóval itt is van a konstruktorban egy paraméter, hogy mit fog csinálni. Ehhez a formhoz tartozik még egy extra funkcióként azt is megjeleníti, hogy mikor olvasta legutóbb az adott könyvet az olvasó.

### Visszavevő form

Amikor visszavesznek egy könyvet, a „bookTakebackPage.cs” lesz meghívva, ami összesíti a késedelmet és egy végső jóváhagyást kér visszavétel előtt. Amikor a felhasználó oldaláról vesznek vissza könyvet ez nem fog megjelenni, mert ott egyszerre több könyvet is vissza lehet venni.

### Alkalmazottak

Az „EmployeeDetailed.cs” az jelenleg bejelentkezett alkalmazott adatait tölti be. A jelszóváltoztatás itt nem olyan biztonságos, mint a felhasználók jelszóváltoztatása, itt csak összehasonlíja a régi jelszót az újjal, illetve az új ismétlésével és ki is cseréli. Itt is van a konstruktorban egy paraméter, ami meghatározza milyen módban indul el. Itt, ha a felhasználó admin jogosultsággal rendelkezik a rendszer adatait és az alkalmazottakat is tudja felügyelni módosítani.

### Kilistázó formok

Ebbe a kategóriába tartozik a „AllBooksPage.cs”, „BookingsPage.cs”, „BorrowingsPage.cs”, „PublishersPage.cs”, „ReservationsPage.cs”, „UsersPage.cs” és gyakorlatilag a „PopupSelect.cs” is de arról egy másik pontban részletesebben is lesz írva. Az összes ilyen form gyakorlatilag az adatbázis valamelyik tábláját jeleníti meg egy datagridview-ben és onnan lehet további műveleteket végezni specifikus sorokkal, hozzáadni újat, meglévőt módosítani, illetve meglévőt törölni. Mindegyik hasonló frissítő metódust használ, de sosem ugyanazt.

privateasyncvoidupdateBooksDgv(stringsearch)

{

cdgvBooks.DataSource = null;

cdgvBooks.Columns.Clear();

ids.Clear();

List<Dictionary<string, string>>response = (List<Dictionary<string, string>>)awaitApiComm.SendPost(newDictionary<string, string> { { "type", "getBooks" }, { "search", search } });

if (response.Count>0)

{

DataTabledt = newDataTable();

DataColumn col;

DataRowrow;

//makethedatasetcolumns

foreach (KeyValuePair<string, string>iteminresponse[0])

{

if (item.Key != "book\_id"&&item.Key != "available")

{

col = newDataColumn();

col.DataType = typeof(string);

col.ReadOnly = true;

col.ColumnName = item.Key;

col.Caption = item.Key;

dt.Columns.Add(col);

}

}

//add buttonrowfordetailedpage

DataGridViewButtonColumnbtns = newDataGridViewButtonColumn();

btns.Name = "Részletesen";

btns.Text = "Részletes";

btns.UseColumnTextForButtonValue = true;

//add buttoncolumnforbooking/available/borrowed

DataGridViewButtonColumnbtncol = newDataGridViewButtonColumn();

btncol.Name = "Műveletek";

//makerows

for (int i = 0; i <response.Count(); i++)

{

row = dt.NewRow();

foreach (KeyValuePair<string, string>iteminresponse[i])

{

if (item.Key == "book\_id")

{

ids.Add(item.Value);

}

elseif(item.Key != "available")

{

row[item.Key] = item.Value;

}

}

dt.Rows.Add(row);

}

cdgvBooks.DataSource = dt;

cdgvBooks.Columns.Add(btns);

cdgvBooks.Columns.Add(btncol);

foreach (DataGridViewColumncolumnincdgvBooks.Columns)

{

column.SortMode = DataGridViewColumnSortMode.NotSortable;

}

addBtnTextsManual();

}

Feltölti a datagridview-t a kapott adatokkal, a sorok azonosítóját nem jeleníti meg, de eltárolja egy listában. Majd formtól függően különböző gomb sorokat ad hozzá.

### Popupselect

Ez a form azért speciális, mert nem csak egy funkciót lát el, hanem többet egyszerre. Mindig ez jelenik meg ha a felhasználónak választania kell valamit (pl.: kivételkor melyik könyvet szeretné kivenni az olvasó). Ezért először indítási módtól függően (ami az osztály konstruktorának paramétere) meghatározza, hogy mi fog megjelenni (pl.: nem minden esetben jelenik meg kereső mező), majd miután a felhasználó megtalálta a megfelelő elemet és kiválasztotta, a form visszaküldi az elem fontos paramétereit, amit a meghívó form fel fog dolgozni. A felhasználói élmény javítása végett bizonyos esetekben egyszerre több opciót is lehet választani (pl.: amikor egy könyvnek több kategóriája van ne egyesével kelljen kiválasztani mindegyiket).

## Weboldal fejlesztői dokumentáció

### Alap tudnivalók a weboldal működéséről:

Minden böngészőből megnyitott link, ami nem egy konkrét fájlra mutat, az átirányítódik az index.php-ra. Ott dől el a beérkező URL alapján, hogy melyik fájl jelenítődik meg. Ennek az az előnye, hogy változtatni tudjuk, hogy mi legyen a linkben az adott oldalakon tartózkodás során, és nem kell hozzá például változtatni a fájlok nevét és helyét a fájlstruktúrában. Ezen felül még nagy előny, hogy minden bejelentkezéssel kapcsolatos ellenőrzésnek elég csak ebben az egy fájlban szerepelnie.

### Bejelentkezés és hozzákapcsolódó részek:

Minden session indítás előtt beállítjuk a session cookietulajdonságait(mivel mindenhol így van ezért, későbbiekben nem kerül említésre), fontosabbak:

* “lifetime”: megadja, hogy mikor jár le, mivel azt szeretnénk, hogy a böngésző bezárásakor jelentkeztesse ki a felhasználót, ezért nulla értékre állítjuk
* “httponly”: igaz értékkor megakadályozza, hogy javascript-ből el lehessen érni a cookie-t
* “samesite”: “Strict” értékre való állításakor védelmet biztosít CSRF (Cross-site requestforgery) támadások ellen

### Index.php-n történő folyamatok egy oldal megnyitásakor:

* Elindul egy session
* Ellenőrzi a program, hogy mikor volt a legutóbbi aktivitás (API-al történő kommunikációnál, és egy másik oldalra való navigálásnál frissül az ideje), ha régebben volt, mint a megadott idő (jelen esetben 1800 másodperc -> 30 perc), akkor törli a session-t és adatait, majd új session-t indít (ezzel kilépteti a felhasználót, ha be volt jelentkezve)
  + Erre azért van szükség, mert vannak olyan böngészők, amik rendelkeznek olyan (vagy hasonló) funkcióval, hogy “Onnan folytatom, ahol abbahagytam”, ami visszaállítja a cookie-kat is, ezzel azt eredményezve, hogy bejelentkezve marad a felhasználó annak ellenére is, hogy be volt zárva a böngésző (hiába van beállítva a session cookie, élettartama nullára), ami problémákat okozhatna.
* Beállítja a legutóbbi aktivitás idejét a jelen időre
* Ellenőrzi (ha jelenleg nincs bejelentkezve senki), hogy be van-e kapcsolva az “emlékezz rám” funkció, ha be van, akkor ellenőrzi, hogy az adatbázisban megtalálható-e a cookie-ban található token és érvényes is, akkor új session id-t, és “emlékezz rám” tokent generál, majd bejelentkezteti a felhasználót. Ha nincs egyezés az adatbázissal, akkor törli a cookie-t.
* Megvizsgálja a kért linket, és az alapján (helyenként az oldalhoz szükséges feltételek vizsgálata után) eldönti, hogy melyik fájlt töltse be, vagy hova irányítsa át a kérést

Az “elementCreators.php”-ben található metódusok html kódot készítenek a kapott paraméterek függvényében.

### Jelszó módosítás működése:

* Profil oldalról:
  + Javascript ellenőrzi, hogy minden feltételnek megfelel-e a megadott új jelszó, és megegyezik-e amikor megvan ismételve, csak ezután küldi el az API-nak, az új és a régi jelszót. A kérés beérkezése után ellenőrésre kerül, hogy be van-e jelentkezve is a felhasználó, ha igen, akkor utána ellenőrzi a megadott jelszavat, ha az érvényes, akkor kicseréli az újra, majd a biztonság kedvéért újra ellenőrizzük, hogy az új jelszóval be lehet-e jelentkezni.
* Emailben kapott “elfelejtettem a jelszavamat” linkkel:
  + Az oldalon megadott jelszavak egyezése esetén, illetve ha teljesül minden jelszóra vonatkozó feltétel, csak akkor megy ki a kérés az API-nak. Ott ellenőrzésre kerül a linkben található token. Ha lejárt, vagy fel lett már használva, akkor érvénytelenként nyílvánítja ki. Ha érvényes, akkor kicseréli a jelszót, majd leellenőrzi, hogy az új jelszóval be lehet-e jelentkezni, ha be lehet, akkor törli a tokent az adatbázisból

## Szerepkörök

### Weboldalon azonosítatlan felhasználó (vendég):

* Bejelentkezés
* Jelszó módosító link kérése e-mailben (“Elfelejtettem a jelszavam”)
* Könyvtár katalógusának megtekintése
* Könyvek részleteinek megtekintése

### Weboldalon regisztrált felhasználó:

* Könyvtár katalógusának megtekintése
* Könyvek részleteinek megtekintése
* Könyvek kedvencekhez adása, vagy levétele
* Könyvek lefoglalása/előjegyzése (a könyv státuszától függően)
* Felhasználói profil megtekintése
  + Jelszó módosítás
  + Kijelentkezés
  + Lefoglalt, előjegyzett könyveinek megtekintése, és azok törlése
  + Jelenleg a felhasználó által kivet könyvekmegtekintése és meghosszabbítása (állapottól függően maximum a system\_settings adatbázis táblában meghatározott alkalommal)
  + Korábban a felhasználó által kivett könyvek megtekintése
  + Kedvencek megtekintése, és levétel a kedvencekről

### Weboldalon regisztrált felhasználó lejárt taggsággal:

* Az azonosítatlan felhasználó jogosultságain felül tud:
  + Kedvencekhez adni könyveket
  + Profilján megtekinteni mindent, amit érvényes taggsággal, viszont nem tudja meghosszabbítani a kivett könyveit

### Asztali alkalmazásban normál dolgozói profil:

* Téma váltás
* Felhasználók regisztrálása
* Könyvek
  + Adatainak módosítása
  + Kiadása
  + Visszavétele
  + Lefoglalása (adott felhasználó számára)
  + Előjegyzése (adott felhasználó számára)
  + Visszaadása
  + Hozzáadása
  + Raktárba vétele
  + Leselejtezése
* Felhasználó
  + Adatainak módosítása
  + Tagság meghosszabbítása
  + Felhasználó deaktiválása (törlése)
* Összes korábban kiadott, jelenleg kiadott, lefoglalt, előjegyzett könyv megtekintése
* Kiadó
  + Hozzáadása
  + Adatainak módosítása
* Dolgozói profil
  + Jelszavának módosítása
  + Deaktiválása
  + Kijelentkezés

### Asztali alkalmazásban admin dolgozói profil:

* A normál dolgozói profilon felül tud:
  + Új dolgozót felvinni
  + Másik dolgozói profilt deaktiválni
  + Rendszerbeállításokat módosítani

## API dokumentáció

### API endpoints:

#### Könyv katalógus lekérése (opcionálisan) szűrve:

* Metódus: GET
* Paraméterek:
  + Title: a könyv címére szűr
  + Genre: a könyv kategóriákra szűr
  + Author: a könyv szerzőire szűr
  + Release\_date: kiadás dátumát szűri
  + Lang: meghatározza milyen nyelvű könyvek jelenjenek meg
  + Isbn: érték amire szűr az isbn-nél
  + Page\_number: egy szám, ami representálja, hogy hanyadik oldalát kéri a találati listának
* Visszatérési érték:
  + A találati lista html kódja

#### Felhasználó bejelentkezés:

* Metódus: POST
* Paraméterek:
  + Uname: felhasználónév
  + Pw: jelszó
  + remember\_me: logikai változó, ami megadja, hogy az “emlékezz rám” opciót bepipálta-e bejelentkezéskor a felhasználó
* Visszatérési érték:
  + String:
    - “notfound” ha a felhasználónév nem található
    - “inactiveuser” ha a felhasználó fiókja deaktiválva (törölve) lett
    - “incorrect” ha a jelszó felhasználónév kombináció helytelen
    - “notregistered” ha a felhasználó a könyvtárban történő regisztrálás után még nem adott meg jelszót a webes bejelentkezéshez
    - “success” sikeres bejelentkezéskor

#### Jelszó módosítása a profil oldalról:

* Metódus: POST
* Paraméterek:
  + CurrentPassword: a jelenlegi jelszó
  + NewPassword: a jelszó amire cserélni szeretné
* Visszatérési érték:
  + JSON ami “status” értéke megadja, hogy sikeres volt-e a csere, vagy ha nem akkor mi a probléma

#### Kivett könyv lejárati dátumának meghosszabbítása:

* Metódus: POST
* Paraméterek:
  + Book\_id: a könyv egyedi azonosítója
  + Action=”extend”: megadja, hogy mit kell tenni, jelen esetben meghosszabbítani
* Visszatérési érték:
  + JSON ami “message” értéke megadja, hogy sikeres volt-e a hosszabbítás

#### Felhasználó bejelentkezve van-e?:

* Metódus: POST
* Paraméterek:
  + Action=”isLoggedIn”: megadja, hogy mit kell tenni, jelen esetben megmondani, hogy a felhasználó be van-e jelentkezve
* Visszatérési érték:
  + Felhasználó egyedi azonosítója, vagy “”

#### Kijelentkezés:

* Metódus: POST
* Paraméterek:
  + Action=”logout”: megadja, hogy mit kell tenni, jelen esetben kijelentkeztetni a felhasználót
* Visszatérési érték:
  + Nincs

#### “Elfelejtettem a jelszavam” link kérés:

* Metódus: POST
* Pareméterek:
  + Username: felhasználónév
  + Action=”resetPassword”: megadja, hogy mit kell tenni, jelen esetben “elfelejtettem a jelszavamat” funkcióhoz linket küldeni email-re
* Visszatérési érték:
  + JSON ami “status” értéke “invalid” ha nem található a felhasználó a megadott felhasználónévvel, vagy “failed\_to\_send\_email”, ha nem sikerült valami miatt kiküldni az emailt, vagy “success” ilyenkor az “email” értéke a felhaszálóhoz kapcsolt email cím lesz, az első négy karakter kivételével az @-ig kicsillagozva (pl.: hort\*\*\*\*\*\*\*@gmail.com)

#### Könyv levétele kedvencek közül:

* Metódus: POST
* Paraméterek:
  + ISBN\_id: a könyv isbn-jének az egyedi azonosítója
  + Action=”removeFromFavorites”” megadja, hogy mit kell tenni, jelen esetben törölni a megadott könyvet a kedvencek közül
* Visszatérési érték:
  + Nincs

#### Könyv felvétele kedvencekbe:

* Metódus: POST
* Paraméterek:
  + ISBN\_id: a könyv isbn-jének az egyedi azonosítója
  + Action=”addToFavorites” megadja, hogy mit kell tenni, jelen esetben törölni a megadott könyvet a kedvencek közül
* Visszatérési érték:
  + Nincs

#### Jelszó módosítása az emaliben kapott linken keresztül (“elfelejtettem a jelszavamat” funkció):

* Metódus: POST
* Paraméterek:
  + Reset\_token: a kapott linkben található token
  + Password: jelszó
* Visszatérési érték:
  + JSON ami “status” értéke “success” ha sikeresen ki lett cserélve a jelszó, “failed” ha valamilyen hiba folytán nem sikerült, “expired” ha a link már nem érvényes

#### Alkalmazott bejelentkeztetése:

* Metódus: POST
* Paraméterek:
  + Username: felhasználónév
  + Passw: jelszó
* Visszatérési érték:
  + JSON-ban visszaadja a “loginEmployee” sql eljárás értékét

**A maradék endpoint:**

A további 48 endpointot az asztali alkalmazás használja, a kommunikáció egyszerűsítésének és egységesítésének érdekében ugyanazt a sémát követi az összes. Mindegyik POST metódus, és van egy “type” paramétere, ami meghatározza melyik SQL eljárást fogja meghívni az API. Az endpoint többi paramétere az eljáráshoz szükséges paramétereket tartalmazza ugyanazzal a névvel, ahogy az eljárásban szerepel. A válaszként pedig az eljárás visszatérési értékét küldi vissza JSON formátumban. Ahol nincs visszatérési értéke az eljárásnak ott nem küld vissza semmit. Az egyetlen kivétel az “addUser” “type” paraméterű endpointnál van, ott a fent említettek mellet még egy jelszó változtatást is meghív az API.

## Adatbázis

Adatok

Adatbázis szerver: **MYSQL**

Adatbázis neve**: konyvtar**

Adatbázis illesztés: **utf8\_hungarian\_ci**

Adatbázis motor**: InnoDb**

Adatbázis létrehozó parancs:

CREATE DATABASE IF NOT EXISTS konyvtar CHARACTER SET utf8 COLLATE utf8\_hungarian\_ci;

### Kapcsolatok

### Táblák

#### A „book” tábla

A „book” tábla tárolja az egyes könyv típusok fontos adatait, amik elsősorban a weboldalon fognak megjelenni, az egyik legfontosabb és leggyakrabban használt tábla

**Szerkezet**

|  |  |  |  |
| --- | --- | --- | --- |
| Mező neve | Mező típusa | Leírás | Kulcs/index |
| ISBN\_id | INT | A könyvtípus egyedi azonosítója | primarykey |
| ISBN | VARCHAR | 2007.01.01-ig megjelent könyveknél 11 karakter utána 13 |  |
| title | VARCHAR | A könyv címe |  |
| publisher\_id | INT | A könyv kiadójának azonosítója | foreignkey |
| release\_date | DATE | A könyv megjelenési dátuma |  |
| lang\_id | INT | A könyv nyelvének azonosítója | foreignkey |
| description | TEXT | A könyv rövid leírása |  |
| picture\_base64 | TEXT | HTML által felismert base64 formátumban vannak a képek tárolva |  |

SQL parancs

CREATE TABLE IF NOT EXISTS book(

ISBN\_id INT AUTO\_INCREMENT PRIMARY KEY,

ISBN VARCHAR(13),

titleVARCHAR(255),

publisher\_id INT,

release\_date DATE,

lang\_id INT,

description TEXT,

picture\_base64 TEXT,

FOREIGN KEY(lang\_id) REFERENCES languages(lang\_id),

FOREIGN KEY(publisher\_id) REFERENCES publishers(publisher\_id)

);

#### A „languages” tábla

A „languages” tábla tárolja a könyvek nyelveit.

Szerkezet

|  |  |  |  |
| --- | --- | --- | --- |
| Mező neve | Mező típusa | Leírás | Kulcs/index |
| lang\_id | INT | A könyv nyelvének egyedi azonosítója | primarykey |
| lang | VARCHAR | A könyv nyelve |  |

SQL parancs

CREATE TABLE IF NOT EXISTS languages(

lang\_id INT AUTO\_INCREMENT PRIMARY KEY,

langVARCHAR(255)

);

#### A „publishers” tábla

A „publishers” tábla a könyvekhez tartozó kiadók adatait tárolja, elsősorban a különféle elérhetőségeket.

Szerkezet

|  |  |  |  |
| --- | --- | --- | --- |
| Mező neve | Mező típusa | Leírás | Kulcs/index |
| publisher\_id | INT | A kiadó egyedi azonosítója | primarykey |
| name | VARCHAR | A kiadó neve |  |
| phone | VARCHAR | A kiadó telefonszáma |  |
| email | VARCHAR | A kiadó e-mail elérhetősége |  |
| webpage | VARCHAR | A kiadó weboldalálnak címe |  |

SQL parancs

CREATE TABLE IF NOT EXISTS publishers(

publisher\_id INT AUTO\_INCREMENT PRIMARY KEY,

nameVARCHAR(255),

phoneVARCHAR(255),

email VARCHAR(255),

webpageVARCHAR(255)

);

#### A „genres” tábla

A „genres” tábla a könyvekhez tartozó kategóriákat tartalmazza.

Szerkezet

|  |  |  |  |
| --- | --- | --- | --- |
| Mező neve | Mező típusa | Leírás | Kulcs/index |
| genre\_id | INT | A kategória egyedi azonosítója | primarykey |
| genre | VARCHAR | A kategória megnevezése |  |

SQL parancs

CREATE TABLE IF NOT EXISTS genres(

genre\_id INT AUTO\_INCREMENT PRIMARY KEY,

genreVARCHAR(255)

);

#### A „books\_genres\_conn” tábla

A „books\_genres\_conn” egy több-több kapcsolatú kapcsolótábla, ami a könyveket kapcsolja össze a hozzá tartozó kategóriákkal.

Szerkezet

|  |  |  |  |
| --- | --- | --- | --- |
| Mező neve | Mező típusa | Leírás | Kulcs/index |
| ISBN\_id | INT | Könyv egyedi azonosítója | foreignkey |
| genre\_id | INT | Kategória egyedi azonosítója | foreignkey |

SQL parancs

CREATE TABLE IF NOT EXISTS books\_genres\_conn(

ISBN\_id INT NOT NULL,

genre\_id INT NOT NULL,

FOREIGN KEY(ISBN\_id) REFERENCES book(ISBN\_id),

FOREIGN KEY(genre\_id) REFERENCES genres(genre\_id)

);

#### Az „authors” tábla

Az „authors” tábla a könyvekhez tartozó szerzőket tartalmazza.

Szerkezet

|  |  |  |  |
| --- | --- | --- | --- |
| Mező neve | Mező típusa | Leírás | Kulcs/index |
| author\_id | INT | A szerző egyedi azonosítója | primarykey |
| author | VARCHAR | A szerző neve |  |

SQL parancs

CREATE TABLE IF NOT EXISTS authors(

author\_id INT AUTO\_INCREMENT PRIMARY KEY,

authorVARCHAR(255)

);

#### A „books\_authors\_conn” tábla

A „books\_authors\_conn” egy több-több kapcsolatú kapcsolótábla, ami a könyvtípusokat a hozzá tartozó szerzőkkel kapcsolja össze

Szerkezet

|  |  |  |  |
| --- | --- | --- | --- |
| Mező neve | Mező típusa | Leírás | Kulcs/index |
| ISBN\_id | INT | A könyv egyedi azonosítója | foreignkey |
| author\_id | INT | A szerző egyedi azonosítója | foreignkey |

SQL parancs

CREATE TABLE IF NOT EXISTS books\_authors\_conn(

ISBN\_id INT NOT NULL,

author\_id INT NOT NULL,

FOREIGN KEY(ISBN\_id) REFERENCES book(ISBN\_id),

FOREIGN KEY(author\_id) REFERENCES authors(author\_id)

);

#### Az „inventory” tábla

Az „inventory” tábla az adott könyvtípushoz tartozó tényleges könyveket tárolja egy-több kapcsolatú kapcsolótábla.

Szerkezet

|  |  |  |  |
| --- | --- | --- | --- |
| Mező neve | Mező típusa | Leírás | Kulcs/index |
| ISBN\_id | INT | A könyvtípus egyedi azonosítója | foreignkey |
| book\_id | INT | A könyv egyedi azonosítója | primarykey |
| available | BOOLEAN | Elérhetőségi flag. Ha hamis, a könyv nem elérhető, le lett selejtezve |  |

SQL parancs

CREATE TABLE IF NOT EXISTS inventory(

ISBN\_id INT NOT NULL,

book\_id INT AUTO\_INCREMENT PRIMARY KEY,

available BOOLEAN,

FOREIGN KEY(ISBN\_id) REFERENCES book(ISBN\_id)

);

#### A „users” tábla

A „users” tábla tárolja az olvasók/felhasználók összes személyes adatát. Szintén egy gyakran használt tábla

Szerkezet

|  |  |  |  |
| --- | --- | --- | --- |
| Mező neve | Mező típusa | Leírás | Kulcs/index |
| user\_id | INT | Felhasználó egyedi azonosítója | primarykey |
| surname | VARCHAR | Vezetékné |  |
| firstname | VARCHAR | Keresztnév |  |
| username | VARCHAR | Felhasználónév, automatikusan generált (vezeték- és keresztév első két betűje + 3 random generált szám) |  |
| birth\_date | DATE | Születési dátum |  |
| membership\_end\_date | DATE | A felhasználó tagságának lejárati dátuma |  |
| email | VARCHAR | Felhasználó e-mail címe |  |
| phone\_number | INT | Felhasználó telefonszáma |  |
| password | VARCHAR | Felhasználó jelszava, a MYSQL PASSWORD nevű függvényével lett titkosítva |  |
| birth\_place | VARCHAR | Születési hely |  |
| address | VARCHAR | Jelenlegi lakcíme |  |
| mother\_maiden\_name | VARCHAR | Anyja lánykori neve |  |
| active | BOOLEAN | Deaktivált felhasználók úgy viselkednek mintha törölve lett volna a fiók |  |

SQL parancs

CREATE TABLE IF NOT EXISTS users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

surnameVARCHAR(255),

first\_nameVARCHAR(255),

usernameVARCHAR(7),

birth\_date DATE,

membership\_end\_date DATE,

email VARCHAR(255),

phone\_numberVARCHAR(255),

passwordVARCHAR(255),

birth\_placeVARCHAR(255),

addressVARCHAR(255),

mother\_maiden\_nameVARCHAR(255),

active BOOLEAN

);

#### A „reservation” tábla

A „reservation” tábla tárolja az aktuális előjegyzéseket, több-több kapcsolatú kapcsolótáblában, néhány művelet könnyítése érdekeben kapott egy egyedi azonosítót is.

Szerkezet

|  |  |  |  |
| --- | --- | --- | --- |
| Mező neve | Mező típusa | Leírás | Kulcs/index |
| reservation\_id | INT | Az előjegyzés egyedi azonosítója | primarykey |
| user\_id | INT | A felhasználó egyedi azonosítója | foreignkey |
| ISBN\_id | INT | A könyvtípus egyedi azonosítója | foreignkey |

SQL parancs

CREATE TABLE IF NOT EXISTS reservation(

reservation\_id INT AUTO\_INCREMENT PRIMARY KEY,

ISBN\_id INT NOT NULL,

user\_id INT NOT NULL,

FOREIGN KEY(ISBN\_id) REFERENCES book(ISBN\_id),

FOREIGN KEY(user\_id) REFERENCES users(user\_id)

);

#### A „shelf” tábla

A „shelf” tábla, a kedvencek funkció. Tárolja a felhasználók kedvenc könyveit. Több-több kapcsolatú kapcsolótábla.

Szerkezet

|  |  |  |  |
| --- | --- | --- | --- |
| Mező neve | Mező típusa | Leírás | Kulcs/index |
| user\_id | INT | A felhasználó egyedi azonosítója | foreignkey |
| ISBN\_id | INT | A könyvtípus egyedi azonosítója | foreignkey |

SQL parancs

CREATE TABLE IF NOT EXISTS shelf(

user\_id INT,

ISBN\_id INT,

PRIMARY KEY(user\_id, ISBN\_id),

FOREIGN KEY(user\_id) REFERENCES users(user\_id),

FOREIGN KEY(ISBN\_id) REFERENCES book(ISBN\_id)

);

#### Az „employees” tábla

Az „employees” tábla tárolja az összes alkalmazott adatait

Szerkezet

|  |  |  |  |
| --- | --- | --- | --- |
| Mező neve | Mező típusa | Leírás | Kulcs/index |
| empl\_id | INT | Az alkalmazott egyedi azonosítója | primarykey |
| empl\_name | VARCHAR | Az alkalmazott neve |  |
| empl\_uname | VARCHAR | Az alkalmazott felhasználóneve automatikusan generált, mint a felhasználók nevei |  |
| password | VARCHAR | Az alkalmazott jelszava az asztali alkalmazáshoz, a MYSQL PASSWORD függvényével titkosítva |  |
| active | BOOLEAN | Deaktivált fiókot nem lehet újra aktiválni, a deaktivált alkalmazottak nem tudnak bejelentkezni |  |
| admin | BOOLEAN | Adminflag, admin alkalmazottból csak egy létezhet és nem lehet deaktiválni |  |

SQL parancs

CREATE TABLE IF NOT EXISTS employees(

empl\_id INT AUTO\_INCREMENT PRIMARY KEY,

empl\_nameVARCHAR(255),

empl\_unameVARCHAR(5),

passwordVARCHAR(255),

active BOOLEAN,

admin BOOLEAN

);

#### A „borrowings” tábla

A „borrowings” tábla tárolja az összes aktív és már lejárt/befejezett könyv kivételt. Több-több kapcsolatú kapcsolótábla, műveletek egyszerűsítése érdekében saját egyedi azonosítót is kapott

Szerkezet

|  |  |  |  |
| --- | --- | --- | --- |
| Mező neve | Mező típusa | Leírás | Kulcs/index |
| borrowing\_id | INT | Kivétel egyedi azonosítója | primarykey |
| book\_id | INT | Könyv egyedi azonosítója | foreignkey |
| user\_id | INT | Felhasználó egyedi azonosítója | foreignkey |
| empl\_e | INT | Alkalmazott egyedi azonosítója. Amíg nincs visszahozva, azt az alkalmazottat tárolja, aki kiadta, ha már vissza lett véve, akkor azt tárolja, aki visszavette | foreignkey |
| start\_date | DATE | Kivétel kezdetének dátuma |  |
| end\_date | DATE | Vége dátum, amíg nincs visszahozva a várt dátumot tárolja, visszahozást követően a visszavétel dátumát |  |
| returned | BOOL | Vissza lett-e már hozva flag |  |

SQL parancs

CREATE TABLE IF NOT EXISTS borrowings(

borrowing\_id INT AUTO\_INCREMENT PRIMARY KEY,

book\_id INT NOT NULL,

user\_id INT NOT NULL,

empl\_e INT NOT NULL,

start\_date DATE,

end\_date DATE,

returnedTINYINT(1),

FOREIGN KEY(book\_id) REFERENCES inventory(book\_id),

FOREIGN KEY(user\_id) REFERENCES users(user\_id),

FOREIGN KEY(empl\_e) REFERENCES employees(empl\_id)

);

#### A „booking” tábla

A „booking” táblában található az összes aktív könyv foglalás, ha egy foglalás lejár automatikusan kitörlődik a táblából. Több-több kapcsolatú kapcsolótábla, néhány művelet végett kapott saját egyedi azonosítót.

Szerkezet

|  |  |  |  |
| --- | --- | --- | --- |
| Mező neve | Mező típusa | Leírás | Kulcs/index |
| booking\_id | INT | Foglalás saját azonosítója | primarykey |
| user\_id | INT | Felhasználó saját azonosítója | foreignkey |
| book\_id | INT | Könyv saját azonosítója | foreignkey |
| end\_date | DATE | Foglalás lejárati dátuma |  |
| handled | INT | Állapot flag, három állapot 1 <- új foglalás, a könyvet félre kell tenni, 0 <- a könyvvel nincs semmi tennivaló, -1 <- foglalás törölve lett a könyvet vissza kell tenni a helyére |  |

SQL parancs

CREATE TABLE IF NOT EXISTS booking(

booking\_id INT AUTO\_INCREMENT PRIMARY KEY,

book\_id INT NOT NULL,

user\_id INT NOT NULL,

end\_date DATE,

handled TINYINT DEFAULT 0,

FOREIGN KEY(book\_id) REFERENCES inventory(book\_id),

FOREIGN KEY(user\_id) REFERENCES users(user\_id)

);

#### A „system\_settings” tábla

A „system\_settigns” tábla a könyvtár rendszeréhez tartozó alapbeállításokat tárolja nem kapcsolódik más táblához szimplán egységes tárolás végett létezik

Szerkezet

|  |  |  |  |
| --- | --- | --- | --- |
| Mező neve | Mező típusa | Leírás | Kulcs/index |
| membership\_fee | INT | Tagság díja, mértékegység: forint |  |
| borrowing\_time | INT | Kivétel ideje, mértékegység: nap |  |
| lengthening\_time | INT | Hosszabbítás ideje, mértékegység: nap |  |
| booking\_time | INT | Foglalás ideje, mértékegység: nap |  |
| max\_reservations | INT | Maximum előjegyzések száma per felhasználó, mértékegység: darab |  |
| max\_lengthenings | INT | Maximum kivétel hosszábbítás per felhasználó, mértékegység: darab |  |
| latency\_fee | INT | Késedelemi bírság per késett nap, mértékegység: forint |  |
| reset\_token\_expiration | INT | Resettoken élettartama, mértékegység: óra |  |
| login\_token\_expiration | INT | Login token élettartama, mértékegység: nap |  |

SQL parancs

CREATE TABLE IF NOT EXISTS system\_settings (

membership\_fee INT,

borrowing\_time INT,

lengthening\_time INT,

booking\_time INT,

max\_reservations INT,

max\_lengthenings INT,

latency\_fee INT,

reset\_token\_expiration INT,

login\_token\_expiration INT

);

#### A „tokens” tábla

A „tokens” tábla emlékezz rám és elfelejtett jelszó funkciókhoz tárol tokeneket.

Szerkezet

|  |  |  |  |
| --- | --- | --- | --- |
| Mező neve | Mező típusa | Leírás | Kulcs/index |
| token | VARCHAR | A generált egyedi token | primarykey |
| user\_id | INT | A felhasználó egyedi azonosítója | foreignkey |
| type | VARCHAR | A token típusa remember\_me/reset |  |
| exipry\_date | DATE | A token lejárati dátuma |  |

SQL parancs

CREATE TABLE IF NOT EXISTS tokens(

tokenVARCHAR(11) PRIMARY KEY,

user\_id INT NOT NULL,

expiry\_date DATETIME,

typeVARCHAR(255),

FOREIGN KEY(user\_id) REFERENCES users(user\_id)

);

### Eljárások

A projekt összes adatbázissal kapcsolatos lekérdezése és parancsa eljárásokkal (storedprocedure) van megvalósítva. Mivel minden lekérdezés, hozzáadás, törlés és módosítás procedure-ként van tárolva így, minden átlátható, könnyen módosítható és tesztelhető, illetve újra használható akár egy másik eljáráson belül.

Egy másik előnye az eljárásoknak, hogy egy php file segítségével egy Sqlinjection biztos php metódust lehet belőle csinálni. Az adatbázissal való kommunikációt kizárólag ilyen metódusokkal valósítottuk meg. Az alábbi kódrészlet a projektben megvalósított metódusokra példa

functionGetBorrowedBooks($user\_id){

        require"databaseConnect.php";

        $query = "CALL getBorrowedBooks(?);";

        $stmt = $conn->prepare($query);

        $stmt->bind\_param("i", $user\_id);

        $stmt->execute();

        $results = $stmt->get\_result();

        $conn->close();

        return $results->fetch\_all(MYSQLI\_ASSOC);

}

#### Eljárások felsorolása

|  |  |  |  |
| --- | --- | --- | --- |
| Név | Bemenet | Kimenet | Leírás |
| listBooksFiltered | title, genres, authors,  releasedate,  language, ISBN, offseter, limiter | ISBN\_id, title, picture base64, genres, authors, publisher, releasedate, language, ISBN, description | Kilistázza a filtereknek megfelelő könyvek adatait, oldalakra bontva |
| getBookByISBN | ISBN | ISBN\_id, title, picture base64, genres, authors, publisher, releasedate, language, ISBN, description | Kilistázza egy adott ISBN kódú könyv összes adatát |
| getGenres |  | genres | Kilistázza az összes adatbázisban létező kategóriát |
| getLangs |  | languages | Kilistázza az összes adatbázisban létező nyelvet |
| getIdFromISBN | ISBN | ISBN\_id | A megadott ISBN kódot ISBN\_id-ra fordítja |
| getBookById | ISBN\_id | ISBN\_id, title, picture base64, genres, authors, publisher, releasedate, language, ISBN, description | Kilistázza egy ISBN\_id alapján megadott könyv összes adatát |
| getBookedAndReservedBooks | user\_id | title, authors, state, reservationid/bookingid | Kilistázza egy adott felhasználó összes foglalását és előjegyzését |
| cancelReservation | reservation\_id, user\_id |  | Törli egy adott felhasználó előjegyzésést |
| cancelBooking | booking\_id, user\_id |  | Módosítja az adott foglalás handled értékét és ha van előjegyzés az adott könyvre akkor a legrégebbit teszi új foglalásnak |
| removeBooking | booking\_id |  | Törli az adott foglalást |
| getBorrowedBooks | user\_id | book\_id, auhtors, ISBN, title, end date, start date, meghosszabítható, fizetendő | Kilistázza egy felhasználó a jelenleg kivett könyveinek adatait pl.: a késésből jövő fizetendőt és hosszabbítható-e |
| getPreviouslyBorrowedBooks | user\_id | book\_id, authors, ISBN, title, picture base64, end date, start date, | Kilistázza egy felhasználó már korábban kivett, de már visszahozott könyveinek néhány adatát |
| deleteLateBookings |  | output | Törli a lejárt foglalásokat illetve a lejárt tagsággal rendelkező felhasználók foglalásait és új foglalást add hozzá ha van előjegyezve output mezőben a töröltbook\_id és user\_id , ha lett hozzáadva új foglalás a hozzá tartozó user\_id, végül a törlés okát |
| deleteLateReservations |  | output | Törli azon felhasználók előjegyzéseit, akiknek a tagsága lejárt, output mezőben a törölt előjegyzések adatai user\_id, ISBN\_id |
| loginUser | username, password | result, user\_id, username | Ellenőrzi a felhasználó bejelentkezési adatait és a megfelelő választ adja rá |
| getUserId | username | user\_id | A username-hez tartozó user\_id-t adja vissza |
| getEmailData | user\_id | email, surname, firstname | Egy user mail küldéshez szükséges adatainak kiírása |
| updateUserDetails | user\_id, surname, firstname, birthplace, birthdate, address, email, phonenumber, mothersmaidenname(mmn) |  | Módosítja egy felhasználó adatait |
| updateBookDetails | ISBN\_id, title, releasedate, language, publisher, authors, genres, description, picture base64 |  | Módosítja egy könyv adatait |
| getUser | user\_id | surname, firstname, username, birthplace, birthdate, email, phonenumber, address, mothersmaidenname(mmn), membership end date, active | Adott user\_id alapján megadja a felhasználó összes adatát |
| getEmployee | empl\_id | name, username, admin | Adott empl\_id alapján megadja a felhasználó összes adatát |
| checkAvailability | ISBN\_id, user\_id | status, available | Egy adott könyv-felhasználó kombinációra adja meg, hogy előjegyzés vagy foglalás lenne ->status, illetve egyáltalán lehetséges-e ->available |
| addReservationOrBooking | ISBN\_id, user\_id | selected | Vagy azelőjegyzés, vagy a foglalás táblába rakja a könyv-felhasználó párost és visszaadja melyikbe tette |
| extendReturnDate | book\_id, user\_id |  | Meghosszabítja az adott kivétel lejárati dátumát |
| extendable | book\_id, user\_id | result | Visszaadja, hogy a megadott kivétel hosszabítható-e |
| deleteExpiredTokens |  |  | Törli a lejárt tokeneket a táblából |
| generateToken | user\_id, token, type | result | Generál egy új tokent a táblába a megadott adatok alapján és visszaküldi, hogy sikerült-e |
| changePassword | password, user\_id |  | Megváltoztatja az adott felhasználó jelszavát |
| changeEmplPass | empl\_id, old pasword, newpassword | result | Megváltoztatja az adott alkalmazott jelszavát és visszaküldi, hogy sikerült-e |
| deleteToken | token |  | Törli az adott tokent-t az adatbázisból |
| addBookType | ISBN, title, genres, authors, publisher, releasedate, lang, description, picture base64 | state | Hozzáad egy új könyvet, ha már létezik a könyv azt jelzi |
| loginEmployee | username, password | info | Az alkalmazott bejelentkezési adatait nézi, ha jó az empl\_id-t ad vissza |
| getUsers | searchTerm | user\_id, surname, firstname | Kilistázza a keresésnek megfelelő felhasználók néhány adatát |
| getEmployees |  | empl\_id, name, username | Kilistázza az aktív alkalmazottak adatait |
| getBooks | searchTerm | booki\_id, ISBN, cím, available, status | Kilistázza a keresésnek megfelelő könyvek néhány adatát, illetve, hogy jelenleg mi a státuszuk (elérhető/kivéve/foglalva) |
| getPublishers | searchTerm | publisher\_id, name, phonenumber, email, website | Kilistázza a keresésnek megfelelő kiadók összes adatát |
| updatePublishers | publisher\_id, name, phonenumber, email, website |  | Módosítja egy adott kiadó adatait |
| deletePublisher | publisher\_id | state | Törli az adott kiadót, ha már nincs egy könyv kiadójaként sem megadva és visszaküldi, hogy sikeres volt-e |
| borrowInfo | book\_id, state | title, ISBN, name, username, user\_id | A state-től függően ad vissza adatokat egy új kivétellelhez („booking” -> előre le volt foglalva, minden adatot visszaad, „book\_id” -> csak a könyv adatait, „user\_id” -> csak a felhasználó adatait adja vissza) |
| borrowBook | user\_id, book\_id, empl\_id | state | Véglegesít egy kivételt és visszajelez, hogy sikeres-e |
| returnBook | user\_id, book\_id, empl\_id | new | Visszavesz egy kiadott könyvet és ha volt rá előjegyzés, akkor a legrégebbit foglalásra váltja |
| returnInfo | book\_id | title, ISBN, name, user\_id, end date, late | Könyv visszavételéhez szükséges adatokat adja vissza |
| addPublisher | name, phonenumber, email, website | state | Hozzáad egy új kiadót az adatbázishoz és visszaküldi, hogy sikeres volt-e a művelet |
| addUser | surname, firstname, username, birthdate, birthplace, email, phonenumber, address, mothersmaidenname(mmn) | state | Hozzáad egy új felhasználót, és visszaküldi, hogy sikeres volt-e a művelet |
| addEmployee | name, username, jelszó |  | Hozzáad egy új alkalmazottat |
| renewMembership | user\_id | end\_date | Meghosszabítja az adott felhasználó tagságát egy évvel és visszaadja az új lejárati dátumot |
| getSystemSettings |  | membeshipfee, borrowingtime, lengtheningtime, bookingtime, maxreservations, maxlengthenings, latencyfee, resettokenexpiraiton, login tokenexpiration | Kilistázza az összes rendszer beállítást |
| deactiveUser | user\_id |  | Deaktivál egy felhasználót |
| deactiveEmpl | empl\_id |  | Deaktivál egy alkalmazottat |
| checkPermissions | user\_id | reservation, booking | Megadja, hogy az adott felhasználó tud-e előjegyezni és foglalni |
| getBookStates | search | ISBN\_id, ISBN, title, state | Kilistázza a keresésnek megfelelő könyvek néhány adatát és hogy foglalható-e, vagy csak előjegyezhető |
| getUsernames | name | username | A felhasználónév generálásához kilistázza a már létező, hasonló felhasználóneveket |
| getEmplUsernames | name | username | Az alkalmazott felhasználónév generálásához kilistázza a már létező, hasonló alkalmazotti felhasználóneveket |
| getHistory | book\_id, user\_id | result | Kivételnél megmondja, hogy a felhasználó olvasta-e már az adott könyvet, ha igen azt is megmondja hány napja olvasta |
| getAuthors | ISBN\_id, inverse | author\_id, author | Kilistázza egy könyv szerzőit, ha az inverse igaz akkor azokat a szerzőket listázza ki, amik nem tartoznak a könyvhöz |
| getCategories | ISBN\_id, inverse | genre\_id, genre | Kilistázza egy könyv kategóriáit, ha az inverse igaz akkor azokat a kategóriákat listázza ki, amik nem tartoznak a könyvhöz |
| deactivateBook | book\_id |  | Deaktiválja az adott könyvet |
| addToInventory | ISBN\_id | book\_id | Hozzáad egy adott típusú könyvet a raktárhoz és megadja az azonosítóját |
| getReservations | searchTerm | reservation\_id, username, title | Kilistázza a keresésnek megfelelő előjegyzéseket |
| getBookings | searchTerm | booking\_id, username, title, end datehandled | Kilistázza a keresési feltételnek megfelelő foglalások összes adatát |
| getBorrowings | searchTerm, returnedToo | borrowing\_id, username, title | Kilistázzat a keresési feltételnek megfelelő kivételeknek adatait, ha returnedToo igaz akkor a visszavitt kivételek között is keres |
| updateSystemSettings | membeshipfee, borrowingtime, lengtheningtime, bookingtime, maxreservations, maxlengthenings, latencyfee, resettokenexpiraiton, login tokenexpiration |  | Megváltoztatja a rendszer beállításokat |
| getBooksFromFavorites | user\_id | title, ISBN\_id, authors, ISBN, picture base64 | Kilistázza az adott felhsználó kedvenceit |
| addBooksToFavorites | user\_id, ISBN\_id |  | Hozzáad egy könyvet a kedvencekhez |
| removeBooksFromFavorites | user\_id, ISBN\_id |  | Töröl egy könyvet a kedvencek közül |
| isItOnFavorites | user\_id, ISBN\_id | result | Megnézi, hogy az adott könyv a kedvencek között van-e |
| getSimilarBooks | ISBN\_id | ISBN, title, picture base64, authors, genres | A megadott könyvhöz hasonló könyveket listáz ki |
| getReservables | searchTerm | ISBN\_id, ISBN, title, state | Kilistázza a keresésnek megfelelő csak előjegyezhető könyveket |
| checkForBookings |  | result | Nézi, hogy jött-e új foglalás |
| changeHandled | booking\_id |  | Megváltoztatja az adott foglaláson a handled értéket |

## Tesztelés

A fejlesztés során kizárólag manuális teszteket végeztünk. Az alkalmazások minden komponensét végiteszteltük, és rendeltetés szerű működésükről megbizonyosodtunk. A továbbiakban azt részletezzük, hogy a rendszer egyes részeit milyen környezetben és hogyan teszteltük. Komponens tesztekre, integrációs tesztekre és rendszer tesztekre egyaránt kiterünk.

### SQL eljárások tesztelése

Az eljárásokat PhpMyAdmin-ban teszteltük Az SQL terminálban és megadtuk a procedure-nek a megfelelő paramétereket és meghívtuk. Majd ellenőriztük, hogy helyesen dolgozta-e fel a kérést és helyes adatokat adott-e vissza.

### Adatbázis tesztadatok

A fejlesztés során több tesztadathalmaz is készült. A legelső LLM által volt generálva és a legtöbb ideig ezek az adatok voltak használva. A második adathalmazra azért volt szükségünk, hogy lássuk hogyan működik rendszerünk nagyobb számú adatnál (weboldalon lapozás funkció működik-e stb.). Az elsőben mindössze 2 könyv szerepelt és még nem volt hozzá megfelelő állapotban a rendszer, hogy alkalmazáson belül bővítsük az első tesztadatokat. Ez az adathalmaz már manuálisan lett legépelve, viszont nem voltak olyan pontosak és helyesek. A könyveknek nem volt címük például csak helyettesítő szöveg pl.: Book1. A harmadik tesztadathalmaz a végleges és legteljesebb adatbázis, a vizsgaremek védés is ezzel kerül bemutatásra. Ez is egy kézzel írott adatbázis, amit már az asztali alkalmazáson belül lett felvéve az adatbázisba és aztán ki is lett exportálva. A tesztadatok között található könyvek a Libri könyvesbolt webáruházából lettek kigyűjtve.

### Api asztali alkalmazással való összeköttetésének integrációs tesztje

Egyik legrészletesebben tesztelt rész az adatbázis és az alkalmazás közötti kapcsolat volt. Mivel ez a leggyakrabban használt eleme az asztali alkalmazásnak. Minden módosítás és lekérdezés ezen keresztül megy végig, ezért nem keletkezhet hiba ezen a helyen. Miután elkészült az alkalmazás egyik olyan komponense, amiben egy POST kérés is szükséges volt ezeket a teszteket hajtottuk végre. Meghívtuk az elkészült részt és megnéztük milyen hibát ad a API. Némely hibákat a Visual StudioCode terminálján lehetett észrevenni, ha ilyen történt akkor vagy nem létezett az adott végpont vagy az eljárásban volt hiba. A másik hely, ahol hibát lehetett észlelni a Visual Studio volt mivel, ha nem megfelelő JSON formátumban érkezett a válasz a rendszer hibát dobott.

### Asztali alkalmazás rendszertesztek

Az asztali alkalmazás rendszertesztjeit a csapat mindhárom tagja végzett. Megbizonyosodtunk, hogy minden funkció hiba nélkül és rendeltetésszerűen működik a tesztadatokon. Ha hibát találtunk a Visual Studiodebug funkciói segítségével kerestük a hiba helyét és hárítottuk el.

# Felhasználói dokumentáció

## Asztali alkalmazás felhasználói dokumentáció

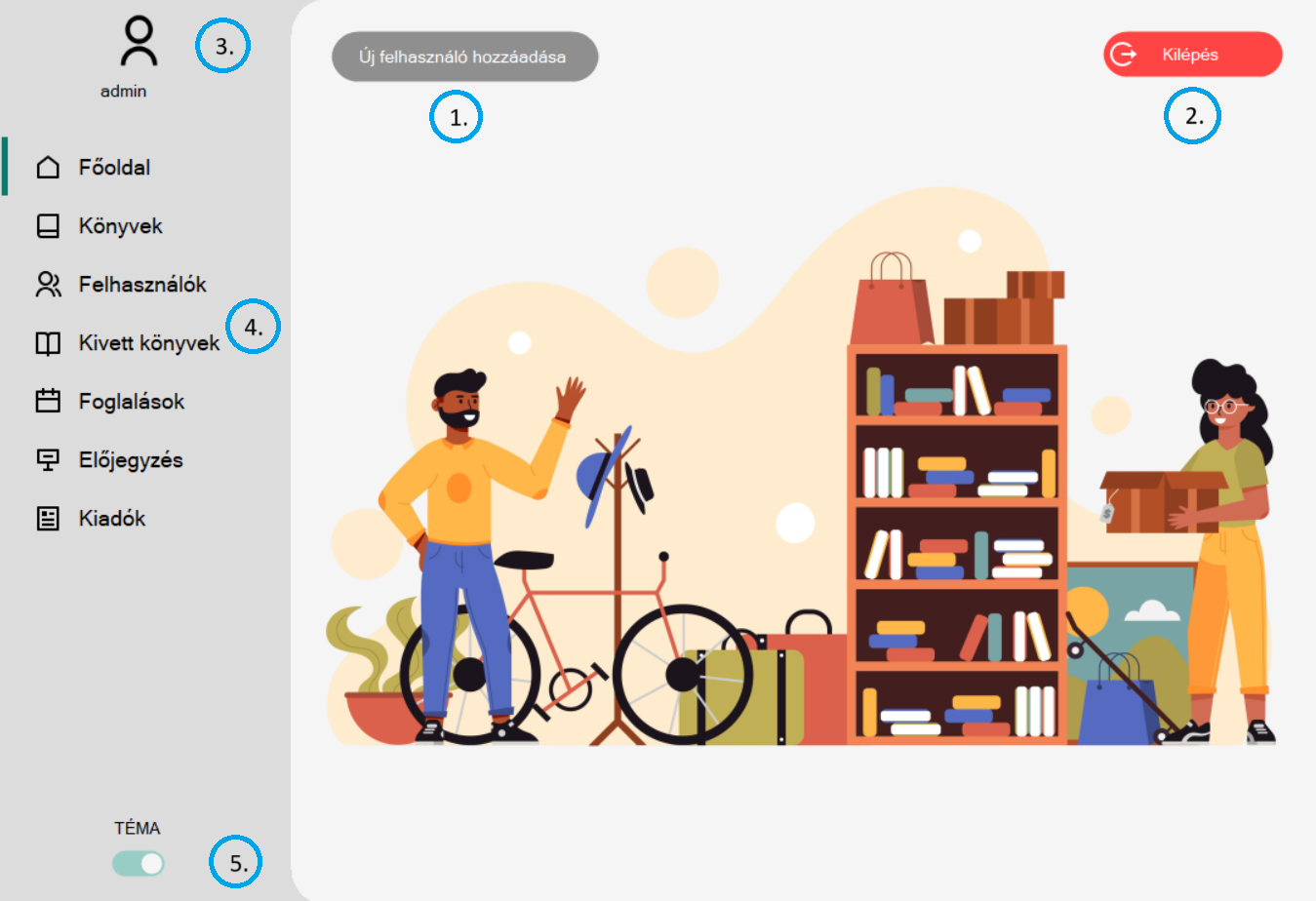
Ez a dokumentáció segít a könyvtár alkalmazottainak megérteni, hogyan használhatják a weboldal különböző funkcióit.

### Belépés

Az alkalmazás indulásánál egyből a bejelentkező felület jelenik meg, ahova be kell írnunk a felhasználónevünket és a jelszavunkat az ezekre kijelölt szöveges beviteli mezőbe, ezután az „Enter” billentyű vagy a „Bejelentkezés” gomb lenyomásával tudunk bejelentkezni. Amennyiben helytelen a felhasználónév vagy a jelszó, az alkalmazás figyelmeztetni fog minket. Ha mégsem szeretnénk bejelentkezni, akkor a bal felső sarokban található X-szel be tudjuk zárni az ablakot.

Sikeres bejelentkezéskor a főoldalra kerülünk.

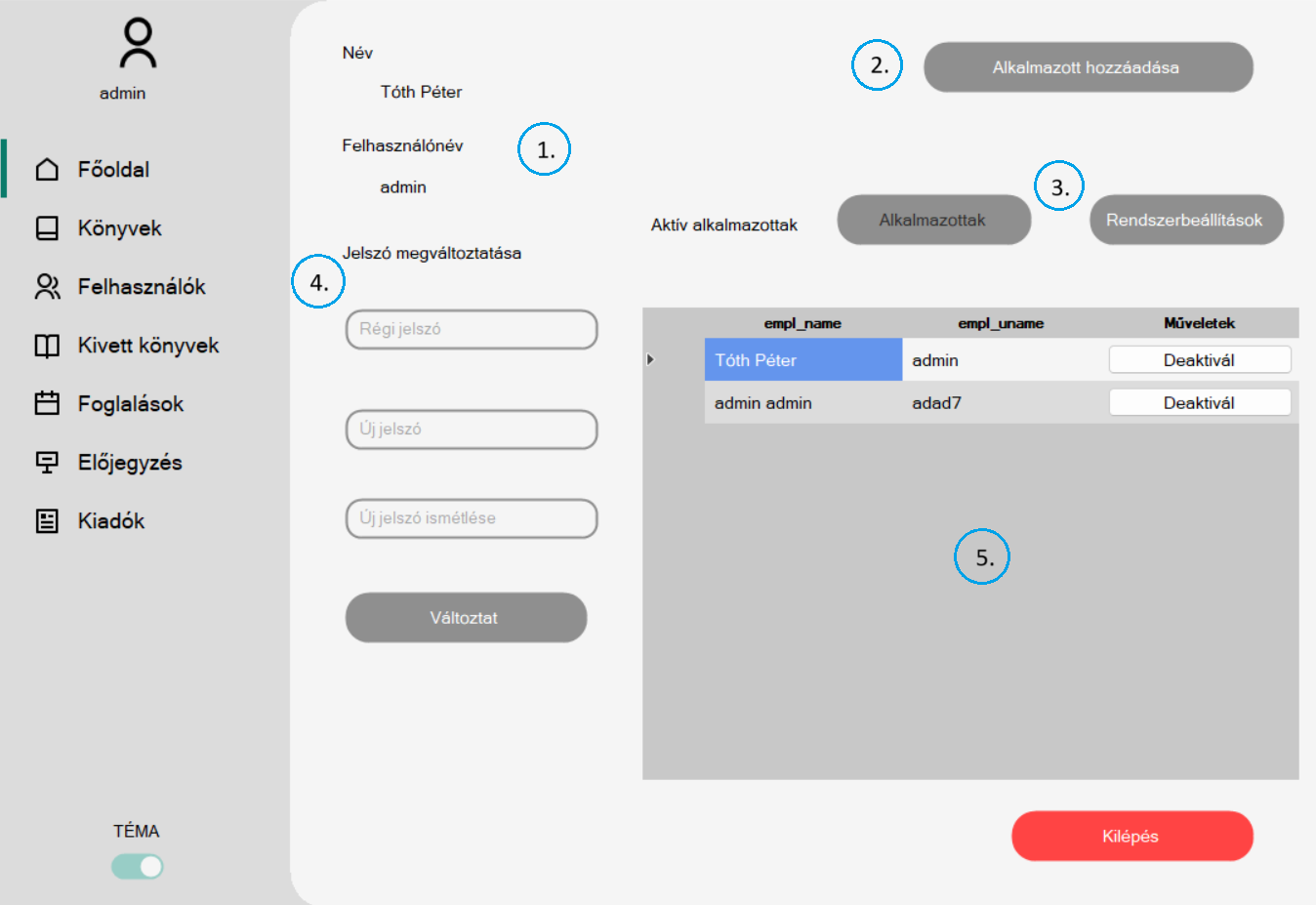
### Főoldal



1. Ezzel az „Új felhasználó hozzáadása” gombbal megnyithatjuk azt az ablakot (vagyis űrlapot), amely segítségével hozzáadhatunk egy új felhasználót az adatbázishoz. Ezzel nem lépünk el a főoldalról.
2. Ezzel a kilépés gombbal kiléphetünk a programból. Ha legközelebb megnyitjuk a programot, újból be kell majd lépnünk.
3. Erre a felhasználó ikonra kattintva elnavigálhatunk arra az oldalra, amelyen megtalálunk részletes információkat a jelenleg bejelentkezett felhasználóról. Erről az oldalról részletesebben a dokumentáció „Dolgozó adatai oldal” szekciójában olvashat.
4. Az ablak bal szélén találjuk a navigációs menüt, mely segítségével navigálni tudunk az oldalak között. A zöld sáv mutatja, hogy jelenleg melyik oldalon vagyunk.
5. Ezzel a váltógombbal tudunk átváltani sötétről világosra, világosról pedig sötét témára egyetlen kattintással. Az alkalmazás bezárásakor a téma elmentésre kerül.

### Dolgozó adatai oldal

A jelenleg bejelentkezett dolgozó adatait az ablak bal felső sarkában található felhasználó ikonra való kattintással tudunk átlépni. (ez a felhasználó ikon =>)

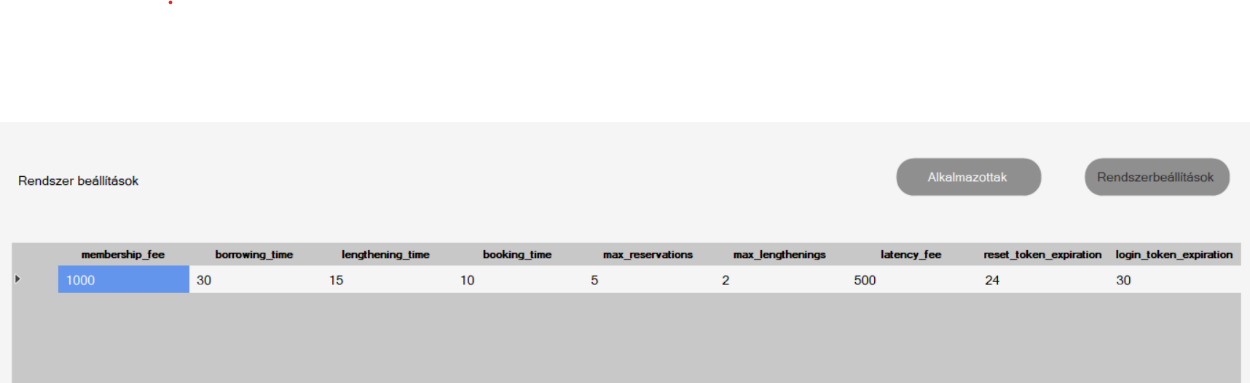


1. Itt találjuk a bejelentkezett alkalmazott adatait.
2. Az „Alkalmazott hozzáadása” gombbal tudjuk megnyitni azt az űrlapot, mellyel hozzáadhatunk egy új alkalmazottat, miután megadtuk az új alkalmazott nevét és a használni kívánt jelszót. A hozzáadás gombbal véglegesen hozzáadhatjuk az alkalmazottat az adatbázishoz.



1. Ezzel a két gombbal („Alkalmazottak” és „Rendszerbeállítások”) tudunk váltani, hogy mi jelenjen meg a táblázatban.

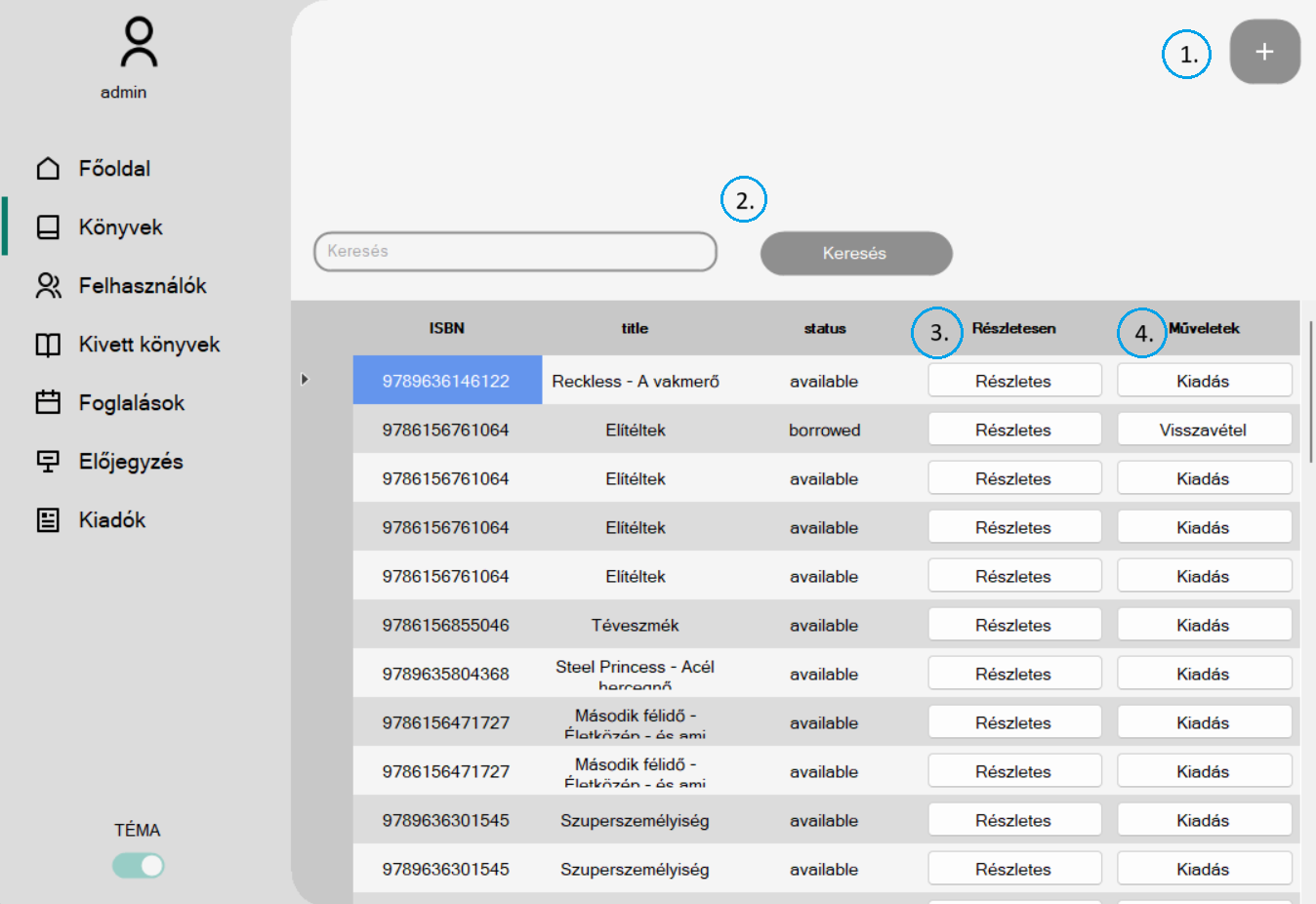
* Az alkalmazottak kilistázásával megtekinthetjük az összes aktív alkalmazottat. Itt megtekinthetjük az alkalmazottak nevét és felhasználónevét, továbbá innen tudunk alkalmazottat deaktiválni az adott alkalmazott sorában található „Deaktivál” gombra kattintva.
* A „Rendszerbeállítások” gombra kattintva átválthatunk arra, hogy a táblázat a rendszerbeállításokat mutassa.



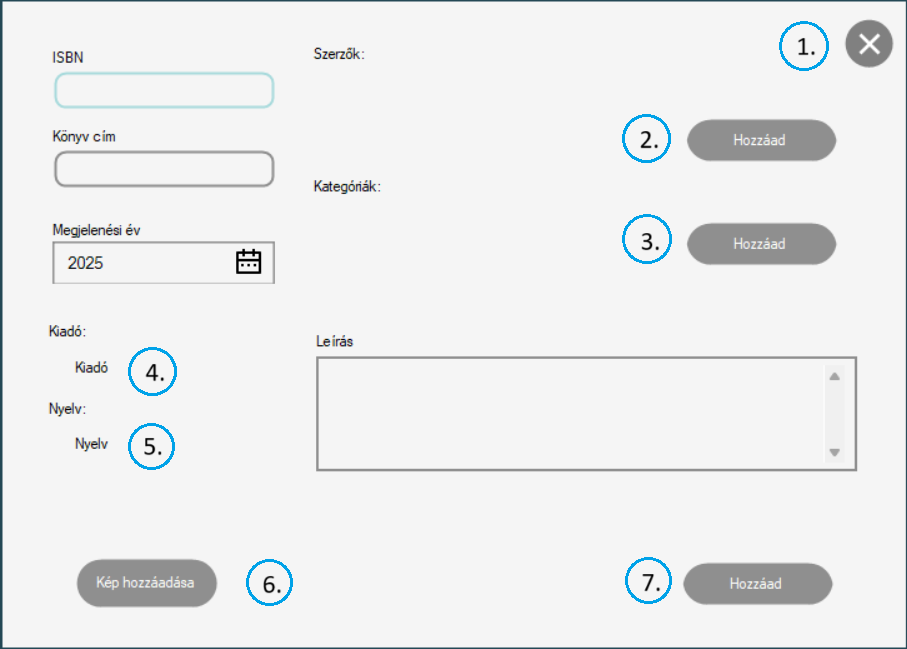
1. Itt módosíthatjuk a jelenleg bejelentkezett alkalmazott jelszavát a beviteli mezők kitöltésével és a „Változtat” gombra kattintva módosíthatjuk véglegesen a jelszót.
2. Ez a táblázat, melynek tartalmát váltani tudjuk a 3-as pontban ismertetett módon.

Az oldal alján található piros „Kilépés” gomb megnyomásával bezárhatjuk a programot.

### Könyvek oldal

A „Könyvek” menüpontra kattintva megnyithatjuk az oldalt, ahol listába van szedve az adatbázisban szereplő összes könyv. A táblázatban szerepel a könyv ISBN száma, címe és státusza, mely azt jelzi, hogy elérhető-e az adott könyv, vagy valamelyik felhasználónál van (available = elérhető; borrowed = jelenleg nem elérhető, ki van véve).  


1. Ezzel a + jellel jelölt gombbal megnyithatjuk azt az űrlapot, melynek segítségével hozzá tudunk adni új könyvet az adatbázishoz. Az alábbi kép mutatja, hogy hogy néz ki az űrlap. Az adatokat a hozzájuk kijelölt beviteli mezőbe.



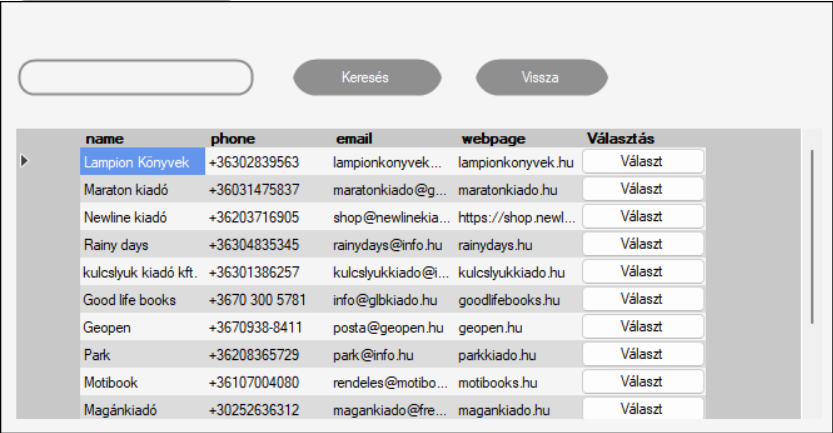
* 1. Ezzel az X-szel tudjuk bezárni az ablakot.
  2. Ha erre a gombra kattintunk, választhatunk szerzőt vagy szerzőket a könyvhöz. Miután kiválasztottuk, a „Választás” gombra kattintva tudjuk véglegesíteni.



* 1. Ha erre a gombra nyomunk, választhatunk kategóriát vagy kategóriákat a könyvhöz. Miután kiválasztottuk, a „Választás” gombra kattintva tudjuk véglegesíteni.



* 1. A „Kiadó” szövegre kattintva kiválaszthatunk egy kiadót a könyvünkhöz a felugró listából a „Választ” gombra kattintva.



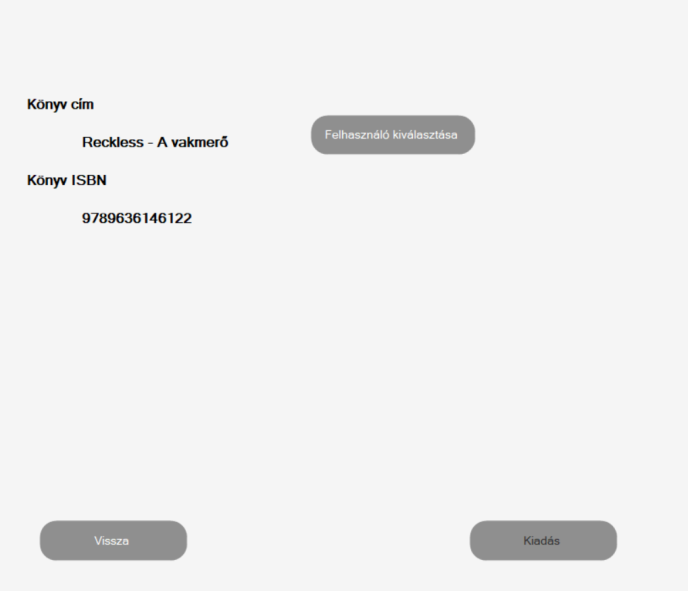
* 1. A „Nyelv” szövegre kattintva kiválaszthatjuk a könyv nyelvét a felugró listából a „Választ” gombra kattintva.



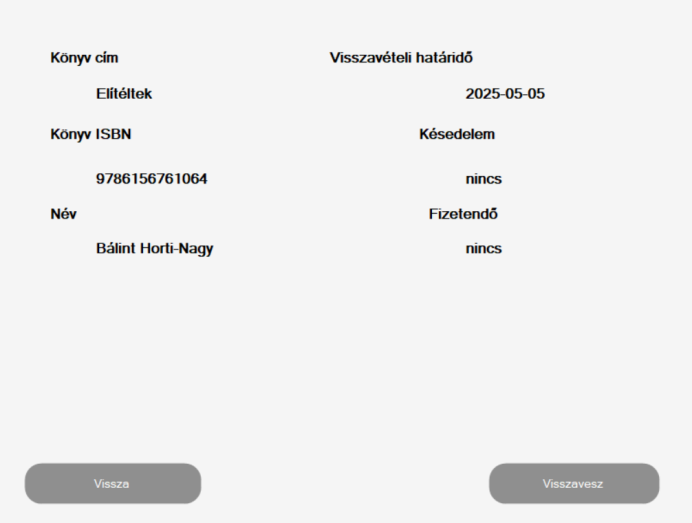
* 1. A „Kép hozzáadása gombra kattintva a fájlkezelőt nyithatjuk meg, ahonnan kiválaszthatunk egy képet a könyvünkhöz.
  2. Ha megadtunk minden adatot, a „Hozzáad” gomb megnyomásával véglegesíthetjük a könyv hozzáadását a megadott adatokkal.

1. Ezzel a szöveges beviteli mező kitöltésével és a „Keresés” gomb megnyomásával keresni tudunk a könyvek között.
2. Ezzel a gombbal megnyithatjuk az adott könyv részleteit leíró oldalt. Itt módosíthatjuk legtöbb adatát. Ugyanazok a mechanikák, mint a „Könyv hozzáadása” oldalon, a különbség csak 3 új gomb.  
   

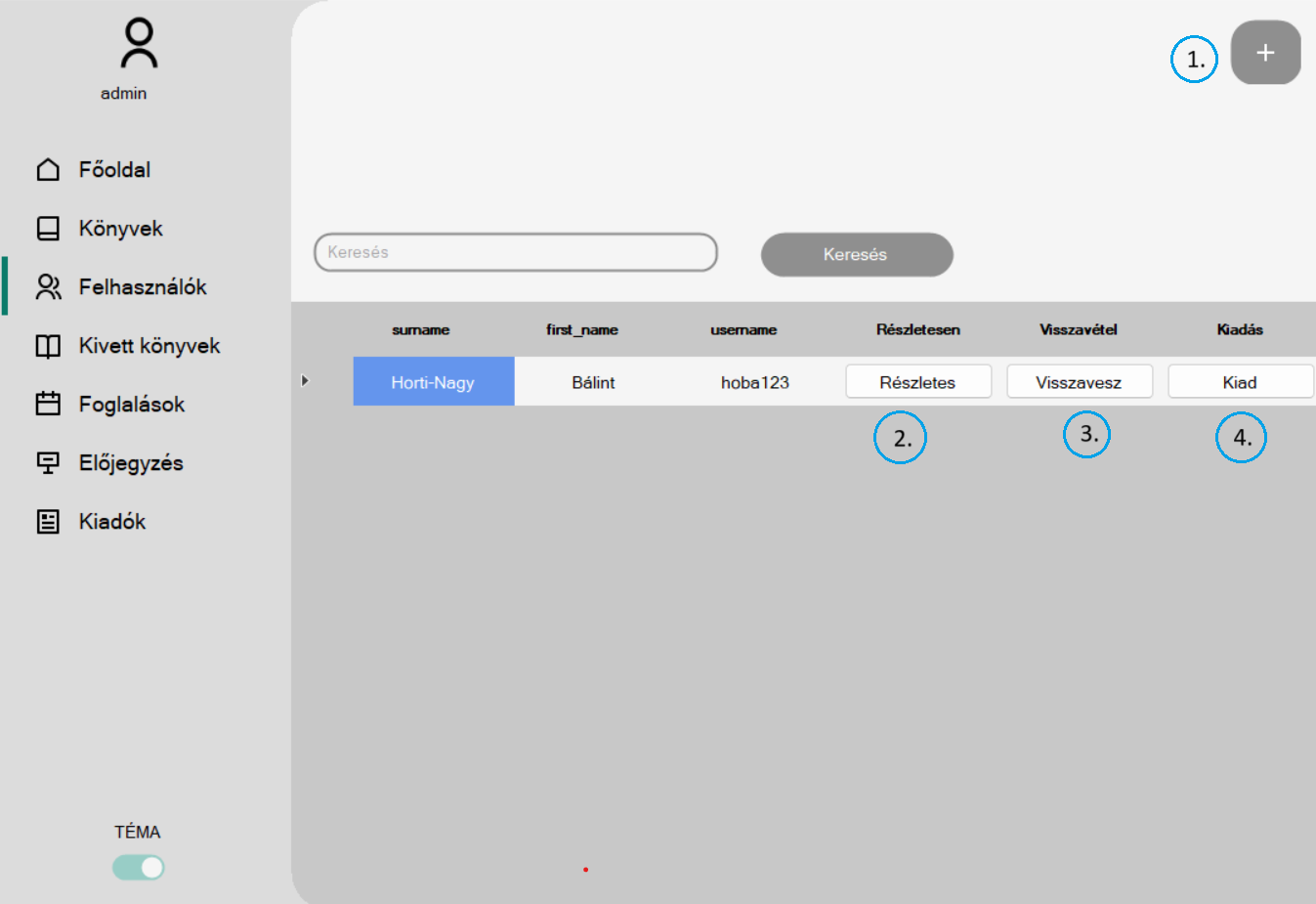
A „Leselejtez” gombbal le tudjuk selejtezni az adott könyvet. A felugró ablakban kijelölhetjük, hogy pontosan melyik könyvet akarjuk leselejtezni, ha az adott könyvből több példányban is van a könyvtárnak. A „Felvesz” gombbal hozzáadhatjuk a könyvet, mint egy új példány a könyvből, a „Vissza” gombbal pedig visszaléphetünk az előző, a kilistázott könyvek oldalára.

1. Ezen a gombon vagy „Kiadás” vagy „Visszavétel” felirat van. A „Kiadás” gombra kattintással kiadhatjuk az adott könyvet. Az oldalon, ahova vitt a kattintás, nyomjunk a „Felhasználó kiválasztása” gombra. Itt válasszuk ki a felhasználót a „Választ” gombra kattintással, majd a „Kiadás” gombra kattintva hajtsuk végre a kiadást.  
   

A „Visszavétel” gombra kattintással átléphetünk a könyv visszavételére használt oldalra, ahol a „Visszavesz” gombbal tudjuk véglegesíteni a könyv visszavevését.



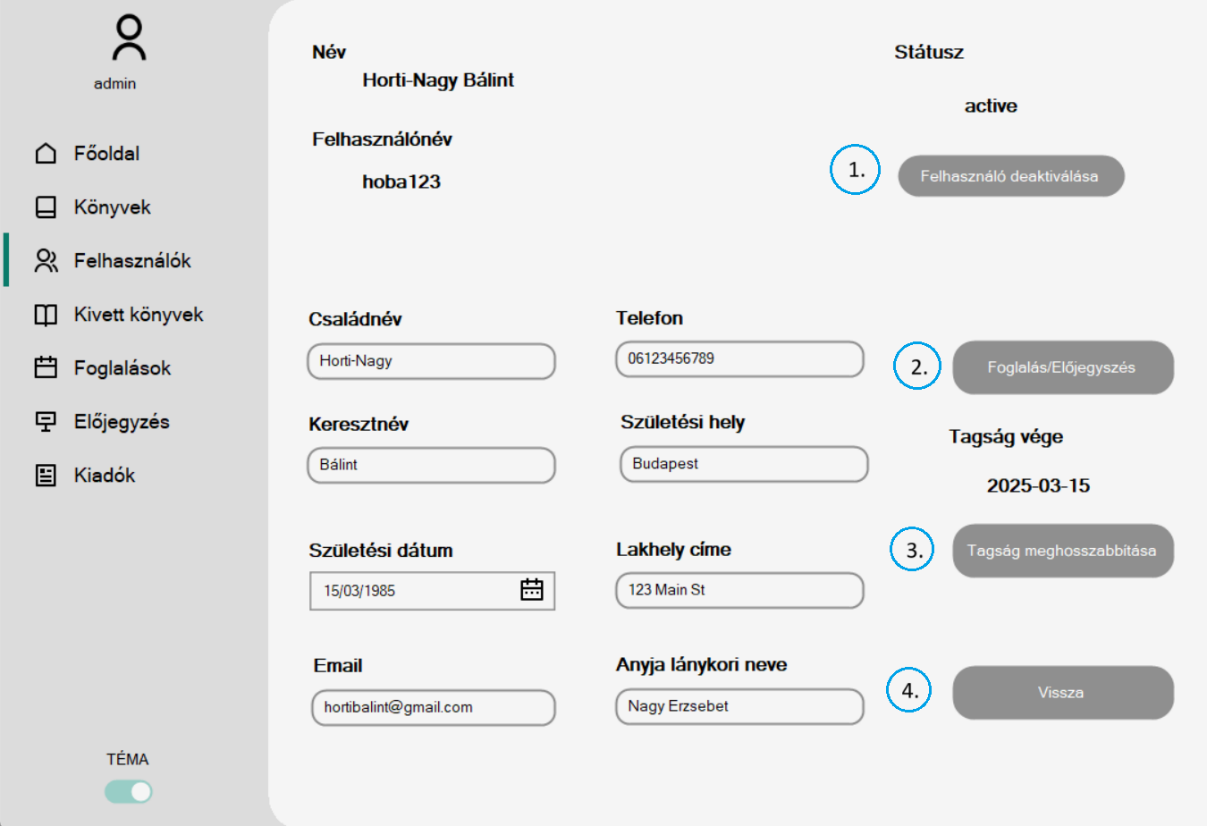
### Felhasználók oldal



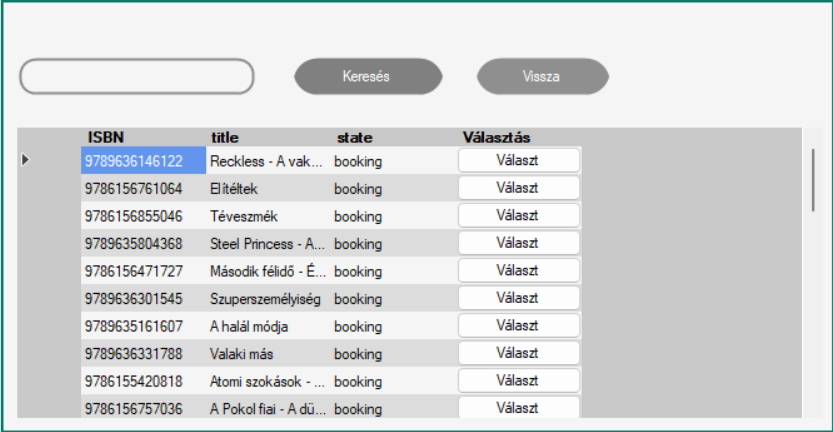
1. Ezzel a gombbal megnyithatjuk azt az ablakot, melynek kitöltésével hozzáadhatunk egy új felhasználót az adatbázishoz.



1. A „Részletes” gombra megnyithatjuk a felhasználó részletes adatait



* 1. Ezzel a gombbal deaktiválni tudjuk az adott felhasználót.
  2. Ezzel a gombbal foglalni vagy előjegyezni tudunk könyveket, attól függően, hogy jelenleg elérhető-e az adott könyv.



* 1. Ezzel a gombbal meghosszabbíthatjuk a felhasználó tagságát egy évvel.
  2. Ezzel a gombbal visszaléphetünk arra az oldalra, ahol ki vannak listázva a felhasználók.

1. Ha erre a gombra rányomunk, megnyitunk egy ablakot, melyben ki tudjuk választani, hogy melyik, jelenleg a felhasználónál lévő könyvet szeretnénk visszavenni.
2. Ha erre a gombra rányomunk, megnyitunk egy ablakot, melyben ki tudjuk választani, hogy melyik könyvet szeretnénk kiadni az adott felhasználónak. Mindezt a „Kiadás” gombbal tudjuk véglegesíteni.

### Kivett könyvek oldal

Ezen az oldalon megtaláljuk az összes olyan könyvet, amely jelenleg az egyik felhasználónál van. A kijelölőnégyzet megjelölésével azt is beállíthatjuk, hogy a már visszahozott, de legalább egyszer már kivett könyveket is mutassa. Természetesen tudunk keresni a kilistázott könyvek között.



### Foglalások oldal

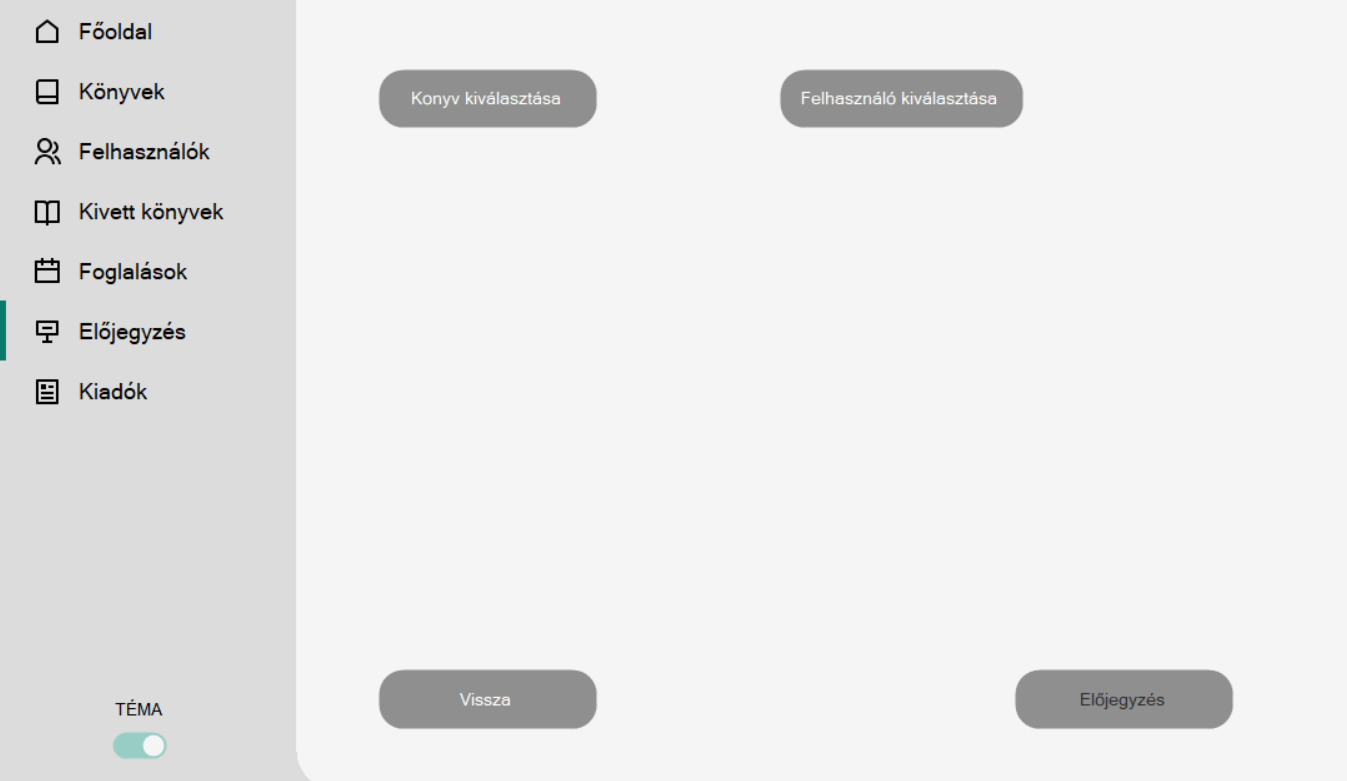
Ezen az oldalon találjuk a lefoglalt könyveket, amelyeket alkalmazottként félre kell tennünk. Ha van újonnan beérkezett foglalás, azt a „Foglalások” menüpontnál egy piros felkiáltójel jelzi, amely eltűnik, amint megtekintettük a foglalást.

### Előjegyzés oldal

Ezen az oldalon találjuk az előjegyzett könyveket. Amennyiben nincsen, ez az oldal üres.



Itt tudunk a bal felső sarokban található + gombbal hozzáadni előjegyzést.



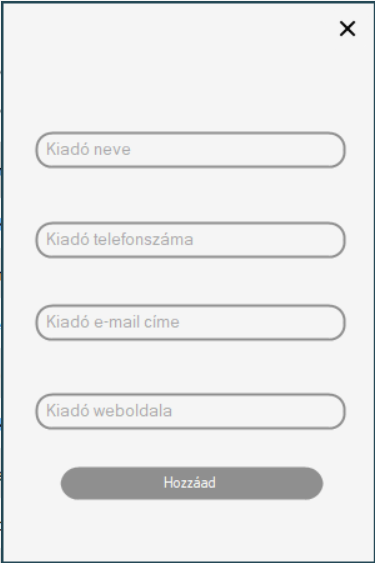
Ezen az oldalon ki kell választanunk egy jelenleg ki nem adható könyvet, majd egy aktív felhasználót. Ezek után az „Előjegyzés” gombra kattintva véglegesíthetjük az előjegyzést.

### Kiadók oldal

Ezen az oldalon megtaláljuk az adatbázisban szereplő összes kiadót és azok elérhetőségeit. A táblázatban szereplő adatokat tudjuk módosítani.



A bal felső sarokban található + gomb megnyomásával megnyithatunk egy űrlapot, melynek kitöltésével és a „Hozzáad” gomb lenyomásával véglegesítve hozzáadhatunk egy új kiadót az adatbázishoz. A mentés nélküli kilépéshez nyomjuk meg a jobb felső sarokban található X gombot.



## WEB felhasználói dokumentáció

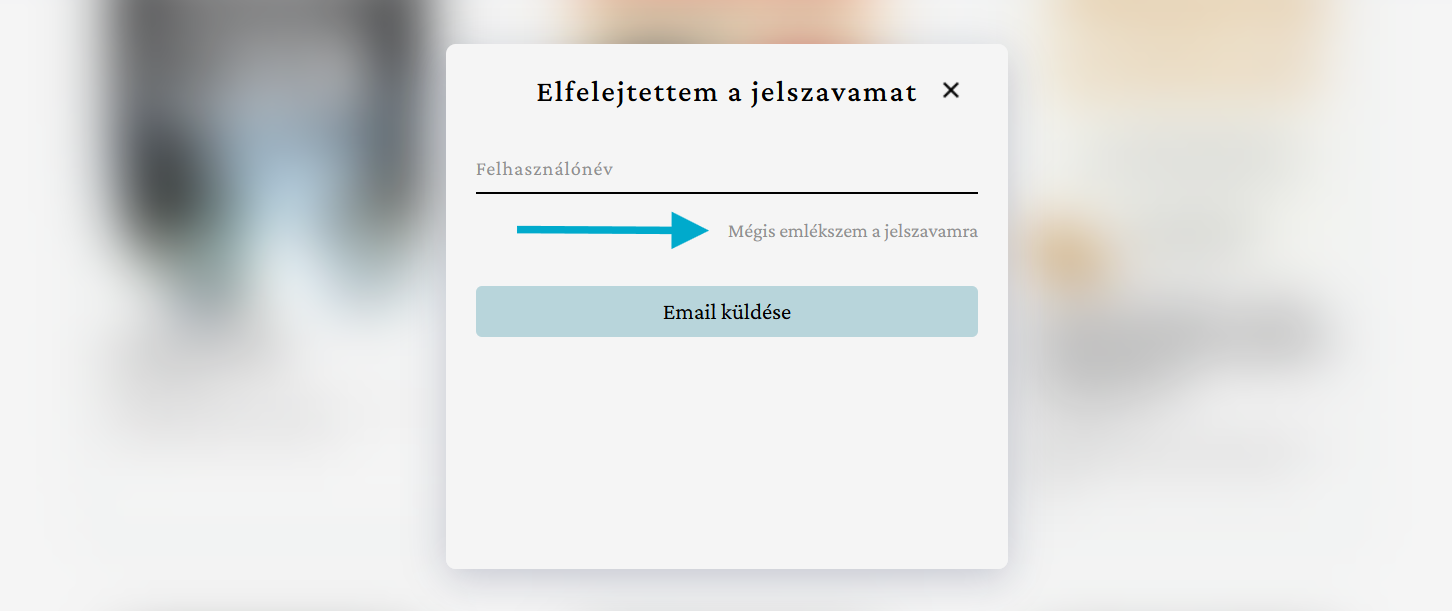
Ez a dokumentáció segít a felhasználóknak megérteni, hogyan használhatják a weboldal különböző funkcióit.

### Belépés

A weboldalt használhatjuk bejelentkezés nélkül is, így viszont csak böngészni tudjuk a könyveket (sem elmenteni sem előfoglalni nem tudjuk azokat). Bejelentkezni úgy tudunk, hogy a jobb felső sarokban található ember körvonalra kattintunk:

A „Bejelentkezés” ablakot a jobb felső sarkában található X-re kattintva (az alábbi kép is mutatja), vagy az ablakon kívülre (az elhomályosított részre) kattintással tudjuk bezárni:



* A weboldalon regisztrációra nincs lehetőség, ezt a könyvtárban, személyesen lehet megtenni a személyazonosság igazolása miatt. Regisztrációkor a felhasználó kap egy a nevéből generált felhasználónevet, mely az olvasókártyáján a teljes nevével együtt szerepelni fog, valamint a felhasználó kap egy e-mail-t a megadott e-mail címre, melyben kap egy linket, amelyen be tud magának állítani jelszót.
* Amennyiben a link már nem érvényes, amikor a felhasználó beállítaná a jelszót, a weboldalunkon a bejelentkezés ablakban az „Elfelejtettem a jelszavamat” hivatkozásra kattintva van lehetőség új e-mailt kérni. Ha mégis emlékszünk a jelszóra, akkor rákattinthatunk az alábbi képen is bejelölt „Mégis emlékszem ajelszavamra” feliratra, amely visszavisz minket a bejelentkezés ablakra.

Az ekkor felugró ablakba begépeljük a kapott felhasználónevünket és a korábban beállított jelszót. Ahhoz, hogy lássuk, hogy mit írtunk be jelszónak, rákattinthatunk a jelszó beviteli mezőjének jobb oldalán található áthúzott szem ikonra. A jelszó elrejtéséhez kattintsunk a már nem áthúzott szem ikonra. (Az alábbi képeken: 1. kép: áthúzott szem ikon = nem látszik a begépelt jelszó, 2. kép: nem áthúzott szem ikon = látszik a begépelt jelszó)

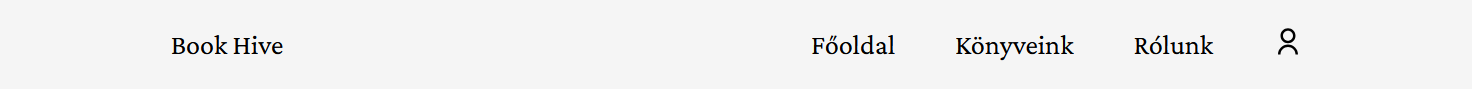


* Tetszés szerint pipáljuk ki az „Emlékezz rám” feliratú jelölőnégyzetet. Ha ezt kipipáljuk, akkor nem kell később a weboldal újbóli megnyitásakor ismét bejelentkezni. Ezt felülírja az, ha a fiókunk részleténél rányomunk a „Kijelentkezés” gombra, ugyanis ezzel kijelentkezünk a fiókunkból és legközelebb megint be kell jelentkeznünk.
* Ezek után a „Bejelentkezés” gombra kattintva tudunk bejelentkezni.

### Navigációs menü

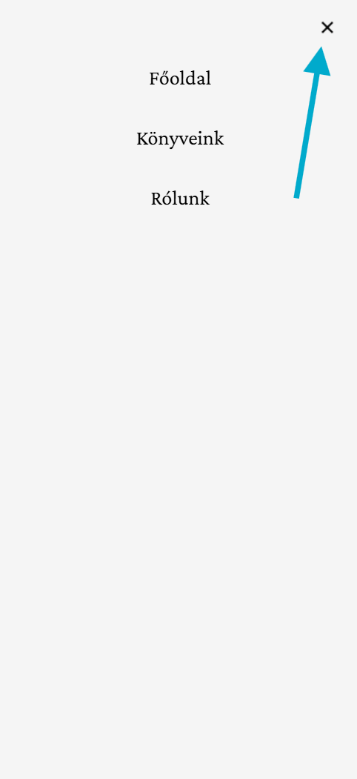
#### Menü – asztali nézet:

A menüsáv a weboldal legtetején található, tehát bárhova görgetünk, mindig elérhető lesz:



#### Menü – mobil nézet:

Mobil nézetben úgy tudjuk előhozni a menüt és navigálni az oldalak között, hogy a jobb felső sarokban található három vízszintes vonalra kattintunk. A menü eltűnik, amikor egy másik oldalra navigálunk vele, de ha nem kívánunk elnavigálni a jelenlegi oldalról, be is zárhatjuk a jobb felső sarokban található X-szel.



#### Navigálás az oldalak között a menü segítségével

A menüsáv bal oldalán található a könyvtár neve (BookHive). Erre kattintva a főoldalra navigálhatunk. Ugyanígy a főoldalra navigálhatunk a menüsáv jobb oldalán lévő „Főoldal” linkkel. Az „Könyveink” -re kattintva arra az oldalra navigálhatunk, ahol megtalálható a könyvtár rendszerébe felvitt összes könyv, a „Rólunk” menüpontra kattintva pedig azt az oldalt kereshetjük fel, mely bemutatja a könyvár rendszerét fejlesztő csapatot.

Ezen felül megtalálhatóak a menüpontok minden oldal legalján lábléc formájában.

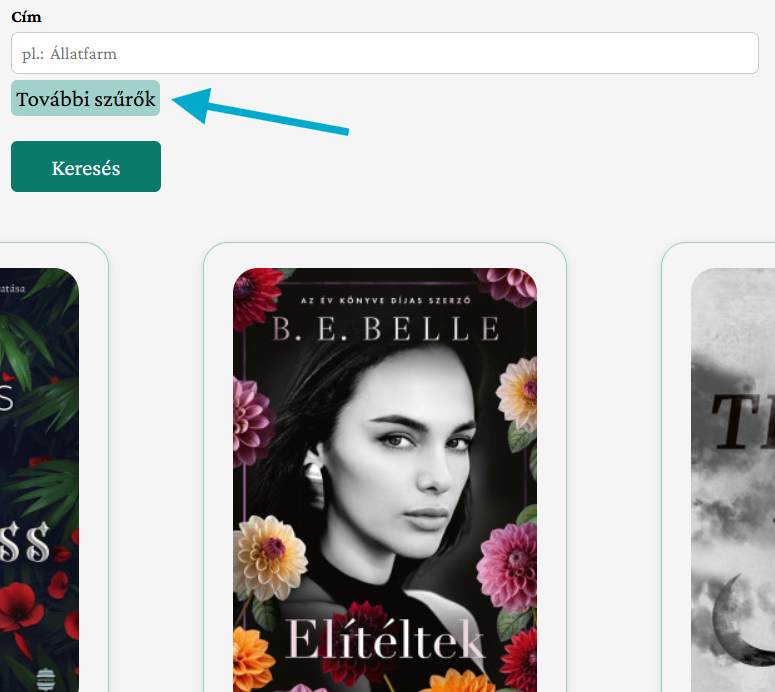
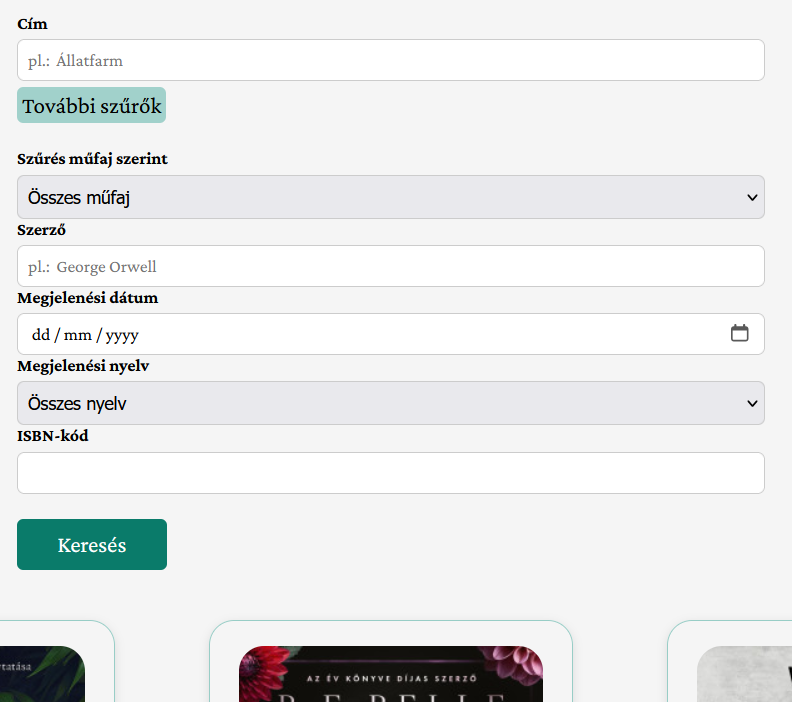
### Oldalak

#### Főoldal

A főoldal bemutatja a könyvtári rendszer főbb funkcióit és lehetőség van innen navigálni a többi oldalra, a profilunk részleteire, vagy ha nem vagyunk bejelentkezve, akkor a bejelentkező felületre.

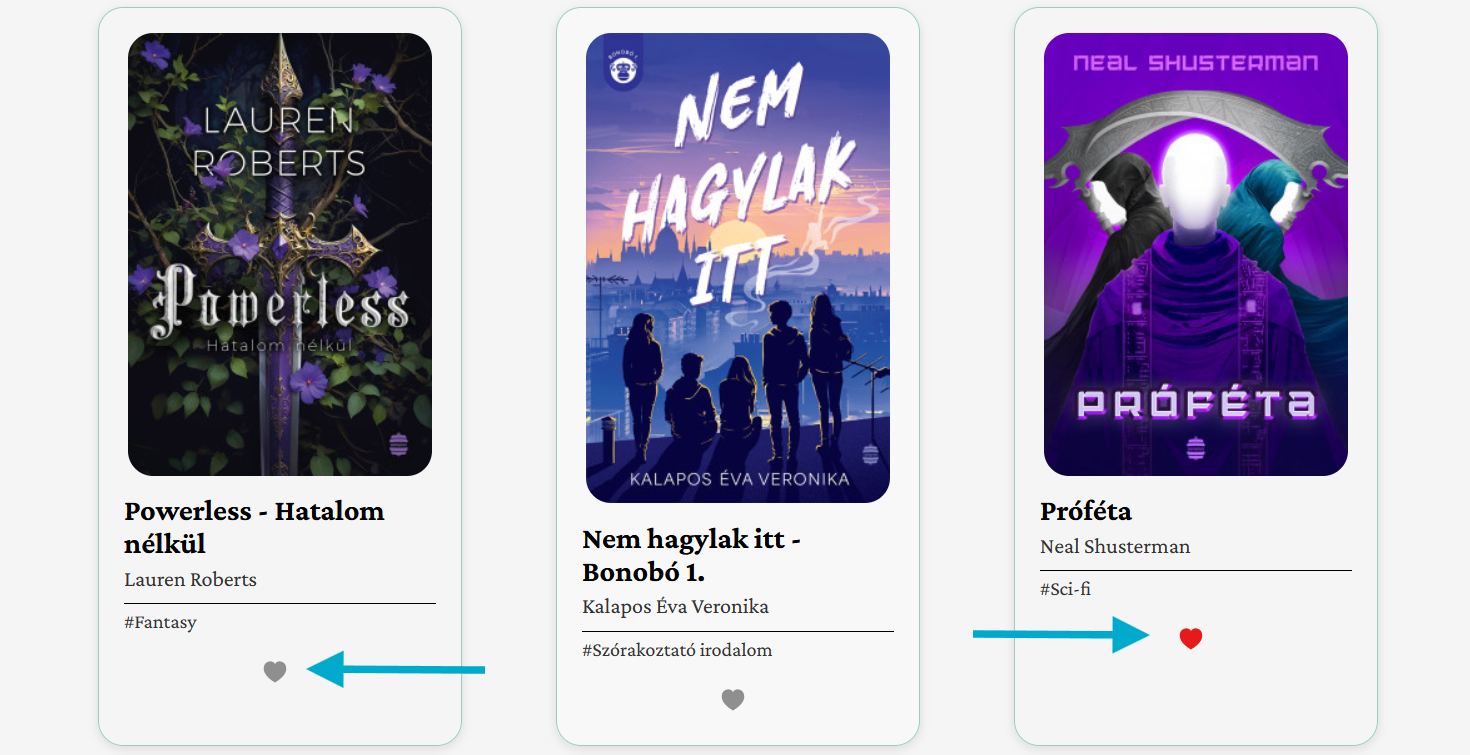
#### Könyveink oldal

A „Könyveink” oldalon megtalálhatóak könyveink. Az oldal tetején található szöveges beviteli mezővel rá tudunk keresni a keresett könyv címére. A keresést az Enter billentyűzet vagy a „Keresés” feliratú gomb lenyomásával tudjuk elindítani. Ezzel a keresővel nem csak teljes címekre lehet rákeresni, hanem címfoszlányokra vagy akár betűkre is. A részletesebb kereséshez kattintsunk a „További szűrők” gombra. Ekkor megjelenik számos további beviteli mező, melyek mindegyikével más-más paraméter alapján tudunk könyveket szűrni. A további szűrőket úgy tudjuk eltüntetni, hogy ismét rákattintunk a „További szűrők” gombra.



Van egy legördülő menü, melyből kiválaszthatunk egy műfajt, ami alapján szűrni szeretnénk.  
Van egy szöveges beviteli mező, mellyel szerzőre tudunk szűrni, de van ezeken kívül van lehetőségünk megjelenési dátum, megjelenési nyelv és ISBN-kód alapján is keresni.

Ha be vagyunk jelentkezve, akkor el tudjuk menteni a könyveket a kedvencek közé azzal, hogy az adott könyv kártyája alján rákattintunk a szürke szív ikonra és ismételt kattintással kivehetjük a kedvencek közül (ha szürke a szív, akkor az adott könyv még nincs a kedvencek között, ha piros a szí, akkor a könyv már a kedvencek között van). Azt, hogy mi kedveltünk egy könyvet, rajtunk kívül senki más nem látja. Ezzel csak elmentjük magunknak az adott művet.



Az oldal alján található a lapozósáv, mellyel navigálhatunk a következő és előző oldalra, továbbá, mutatja, hogy hányadik oldalon vagyunk.

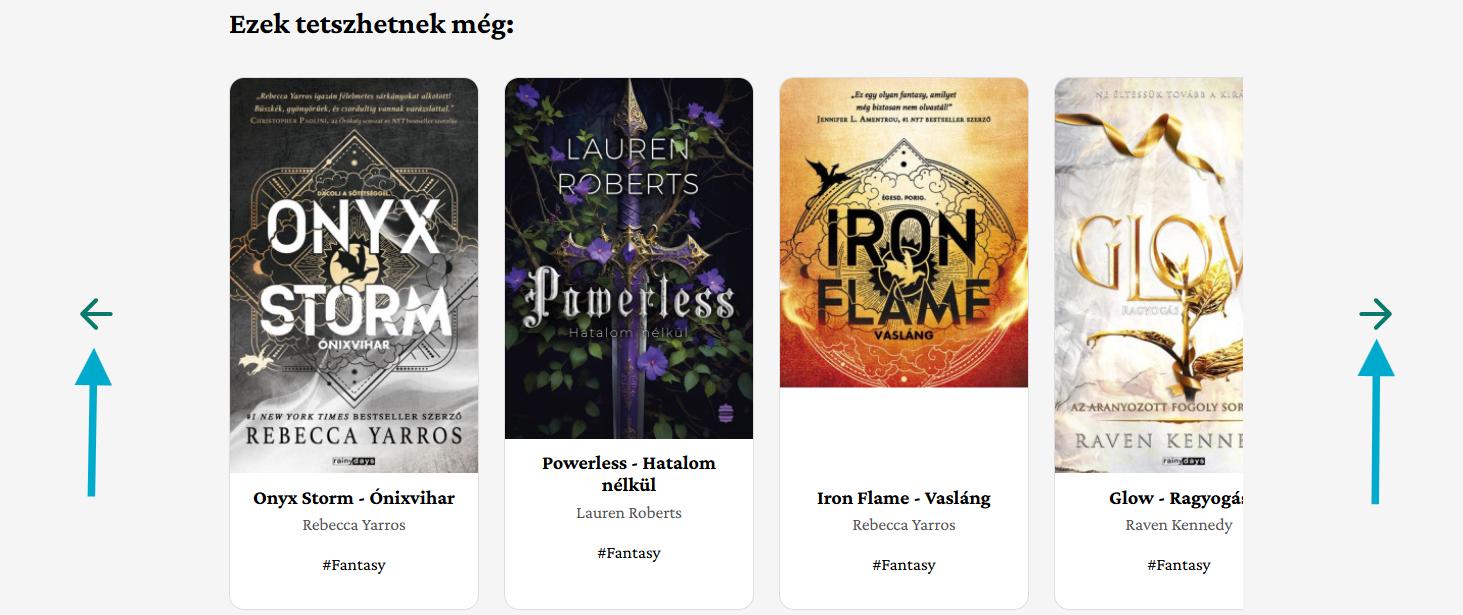
#### Könyv részletes oldal

Egy könyv további részleteit (leírás, ISBN, megjelenés éve) úgy tudjuk megtekinteni, hogy az adott könyv kártyájára kattintunk (ha be vagyunk jelentkezve, akkor a szívre kattintva csak a kedvencek közé rakjuk a könyvet, azzal nem nyitjuk meg a könyv részleteit).

A könyv részletes oldalán is lehetőségünk van a kedvencek közé tenni a könyvet a további adatok alatt található szürke szív ikonra kattintással, amennyiben be vagyunk jelentkezve.



Az oldal alján található egy ajánló szekció, ahova kilistázzuk az adott könyvhöz hasonló köteteket. Amennyiben a könyvek nem fértek ki a képernyőre, az ajánló szekció bal és jobb oldalán található nyilakkal, vagy a szekció alján található görgetősávval lehet tovább léptetni.



#### Profil részletei oldal

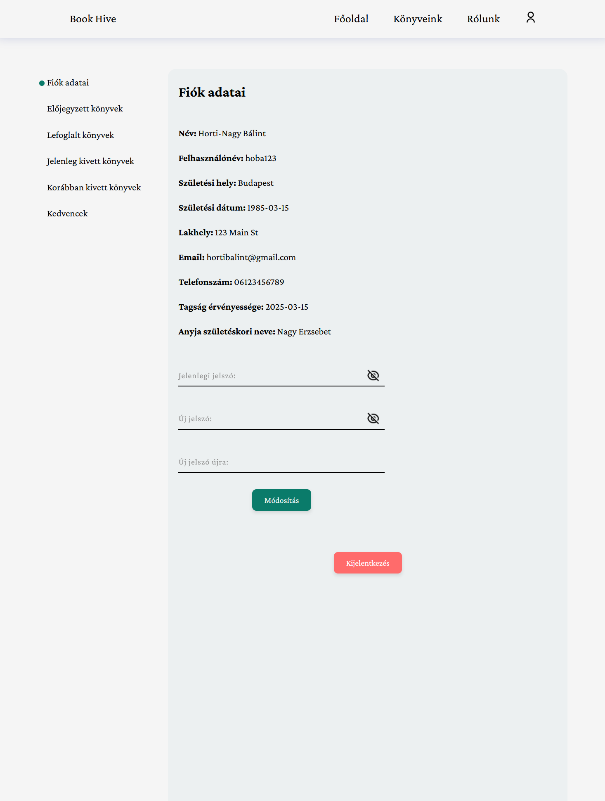
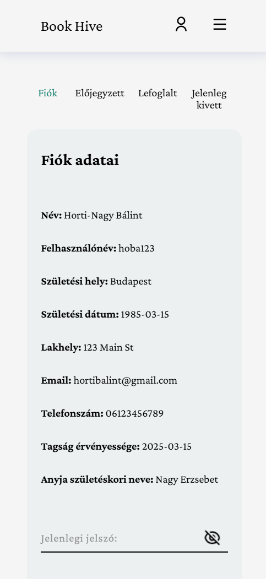
A profilunk részleteit a jobb felső sarokban található ember körvonalra kattintva tudunk lépni, amennyiben be vagyunk jelentkezve a fiókunkba. Ezen az oldalon megtaláljuk saját adatainkat és fiókunk részleteit. Asztali nézetben az oldal bal felső sarkában, mobilos nézetben az oldal tetején, a főmenü alatt találunk egy navigációs menüt, mellyel az oldalon megjelenített adatok között tudunk váltani. Asztali nézetben egy zöld pont, mobilos nézetben pedig a menü szövegének zöldre váltása mutatja, hogy milyen adatokat nézzük éppen.

Az első menüpont a Fiók/Fiók adatai, melyen megtekinthetjük a fiókunk adatait (a nevünk, felhasználónevünk, születési helyünk, születési dátumunk, lakhelyünk, e-mail címünk, telefonszámunk, a tagságunk lejárati dátuma és az édesanyánk születési neve). Ezen az oldalon tudjuk megváltoztatni a fiókunk jelszavát, amelyet a következőképpen tehetünk meg:

Van három szöveges beviteli mező:

1. beírjuk a jelenlegi jelszót
2. ezután beírjuk az új jelszót
3. majd az új jelszót még egyszer.

Ha mindezzel megvagyunk, rányomhatunk a zöld „Módosítás” gombra, mellyel véglegesíthetjük az új jelszót. Ennek az oldalnak az alján található a „Kijelentkezés” gomb, mellyel kijelentkezhetünk a fiókunkból.



* Az „Előjegyzett/ Előjegyzett könyvek oldalon találjuk a könyveket, amelyekre előjegyeztünk. Ezeket az előjegyzéseket egyenként le tudjuk mondani az adott könyv kártyájában található piros „Lemondás” gombra kattintva. Ha jelenleg nincs egy sem, akkor ezt egy „Nincs előjegyzett könyve!” felirat fogja nekünk jelezni.
* A „Lefoglalt/Lefoglalt könyvek oldalon találjuk a könyveket, amelyeket lefoglaltunk. Ezeket a foglalásokat egyenként le tudjuk mondani az adott könyv kártyájában található piros „Lemondás” gombra kattintva. Ha jelenleg nincs egy sem, akkor ezt egy „Nincs lefoglalt könyve!” felirat fogja nekünk jelezni.
* A „Jelenleg kivett/ Jelenleg kivett könyvek” oldalon találjuk azokat a könyveket, melyeket még nem vittünk vissza. Ha van még rá lehetőségünk, a „Meghosszabbítás” gombra kattintva meghosszabbíthatjuk azt az időt, ameddig az adott könyv nálunk lehet. Ha jelenleg nincs egy sem, akkor ezt egy „Nincs kivett könyve!” felirat fogja nekünk jelezni.
* A „Korábban kivett/ Korábban kivett könyvek” oldalon találjuk azokat a könyveket, melyeket korábban már kivettünk és vissza is vittünk. Ha jelenleg nincs egy sem, akkor ezt egy „Még nem hozott vissza könyvet!” felirat fogja nekünk jelezni.
* A „Kedvencek” oldalon találjuk azokat a könyveket, melyeket a kis szürke szív ikonra kattintással kedveltünk. Innen eltávolíthatjuk bármelyik könyvet a „Levétel a kedvencekről” gombra kattintva. Ha jelenleg nincs egy sem, akkor ezt egy „Még nem tett könyvet a kedvencek közé” felirat fogja nekünk jelezni.

#### Rólunk oldal

Ezen az oldalon megtaláljuk a könyvtár rendszerét fejlesztő és tervező csapat bemutatóját képekkel, nevekkel és szerepkörökkel. Ezután megismerhetjük a célt, melyet kitűzött magának a csapat, továbbá találunk néhány érdekes adatot a csapat eddigi közös munkáiról.

## Irodalomjegyzék

* <https://www.w3schools.com/>
* <https://stackoverflow.com/questions>
* <https://chatgpt.com/>
* <https://jquery.com/>
* <https://github.com/PHPMailer/PHPMailer>

## Forráskód

* <https://github.com/BaBence04/Konyvtarazo-rendszer-projekt>