

# **Online könyvkölcsönző**

## **Záródolgozat**

Készítette:

**Hencz Kornél Tibor**

SZOFTVERFEJLESZTŐ  
OKJ 5421305

## **Feladatlap**

záródolgozat készítéséhez

A tanuló neve: Hencz Kornél Tibor

A dolgozat címe:

## Online könyvkölcsönző

Belső konzulens: Borzok Károly

### A feladat leírása:

- Egy szoftver, amin keresztül a felhasználók a fennálló "kredit" pontjaikból különböző típusú könyveket(pdf) kölcsönözhetnek ki, online, webes felületen.
- A regisztrálást követően a felhasználó minden nap kap egy darab kreditet.
- A webes felületen tudja a felhasználó a már kibérelt könyveit olvasni, vagy új könyvet bérelni. Illetve itt található egy teljes katalógus a már elérhető könyvekből.
- A webfelülethez tartozik egy admin felület, ahol az ezzel a joggal rendelkező felhasználók tudnak új könyveket feltölteni, és törölni.

## Konzulensi lap

**A záródolgozat címe: Online könyvkölcsönző**

**Készítőjének neve: Hencz Kornél Tibor**

**Konzulens neve: Borzok Károly**

**Konzultációs időpontok:**

**Év, hónap, napkonzulens aláírása**

- 
- 
- 
- 
- 
- 
- 

**A záródolgozat beadható.**

**A beadás dátuma: 2020.04.30**

**Konzulens aláírása**

## Nyilatkozat

Alulírott Hencz Kornél Tibor a Székesfehérvári SZC Széchenyi István Műszaki Szakgimnáziuma és Szakközépiskolája Szoftverfejlesztő szakos tanulója kijelentem, hogy az általam készített *Online könyvkölcsönző* című záró dolgozat és annak részei egyenként is saját szellemi termékek. Ellenkező esetben az előírt hivatkozásokat az előírásoknak megfelelően alkalmaztam.

**Tudomásul veszem továbbá, hogy plágium esetén a szakdolgozat az elbírálás bármelyik fázisa során elégtelennek minősíthető.**

Hencz Kornél Tibor

## Tartalom

### Fejlesztői dokumentáció

*A témaválasztás - - 6*

*Fejlesztői eszközök - - 6*

*Adatbázis felépítése - - 6*

*“users” tábla - - 8*

*“books tábla” - - 9*

*“rents tábla” - - 10*

*“admins tábla” - - 11*

*A szoftver megvalósítása - - 12*

*Index - - 12*

*A szerver oldal - - 13*

*Modulokra bontás - - 14*

*Tesztelés - - 17*

*Fehér doboz - - 17*

*Fekete doboz - - 18*

*Egyéb tesztek - - 19*

*A rendszer beüzemelése - - 19*

*Továbbfejlesztési lehetőségek - - 21*

*Irodalomjegyzék, Források - - 22*

### Felhasználói dokumentáció

*A program általános specifikációja - - 23*

*Rendszerkövetelmény - - 23*

*A program telepítése/elérése - - 23*

*Használati útmutató - - 24*

*Adminisztrátori felület útmutató - - 27*

## Fejlesztői dokumentáció

### A témaválasztás

Az online könyvkölcsönzés webes közegben azért tűnt megfelelőnek, mert egy ilyen rendszerben elengedhetetlen az adatbázis használata, illetve a szerver és kliens oldali scriptek, különböző programozási, és leíró nyelveken. Így komplexitásából adódóan a téma megfelel a feladatnak. Másfelől pedig egy olyan megoldás létrehozása volt a cél, ami közel áll hozzám. A könyvek digitális formában történő terjesztése pedig pont ilyen. Ezért kitaláltam egy rendszert, ahol a felhasználók egyszerűen, és dinamikusan tudnak digitálisan hasznos információkhoz, vagy akár különböző szórakoztató könyvekhez jutni. A dinamika nem csupán a felhasználói felületre vonatkozik! Szempont volt az ötlet kialakításánál, hogy a könyvek egyszerűen kezelhetőek legyenek. Vagyis egy "adminisztrátori" felület elengedhetetlen volt. Valamint a legfontosabb szempont az volt, hogy maga a webes szoftver portolható legyen. Tehát többnyire akármilyen php szerveren beüzemelhető, egy megfelelő mysql adatbázissal.

### Fejlesztői eszközök

A fejlesztői eszközök kiválasztása az eddigi ismereteim, tapasztalataim alapján történt. Tehát csak olyan szoftvereket használtam, amiket már használtam korábban is fejlesztésre, tervezésre. Ennek oka azok kézenfekvősége, és megbízhatóságuk.

Az adatbázis megalkotására a MySQL Workbench nevű szoftver 8.0 verzióját használtam, majd a menedzselésére pedig az XAMPP programban található phpMyAdmin legfrissebb változatát. A forráskód megírása pedig a Notepad++ szöveg/forráskód szerkesztőben történt. Böngészős felületre írt, Windowsban nem futtatható algoritmusaim, programjaim kódolására kivétel nélkül ezt használom, mert az összes olyan nyelvet, amikre szükségem van ilyen esetben, vagyis a php-t, a HTML-t, a JavaScriptet, CSS-t kiválóan kezeli. Elkülöníthetően színezi a függvényeket, iterációkat, és javasolja a lehetőségeket. A felületen történő különböző formázások pedig a Bootstrap CSS rendszerével történtek.

Továbbá a dokumentálás a Microsoft szövegszerkesztő szoftverében a Word-ben készült.

### Adatbázis felépítése

A rendszer számára elengedhetetlen egy megfelelően működő, összetett adatbázis. Így a fejlesztés maga, ennek megtervezésével kezdődött.

A megoldás kitalálásának menete a szoftver későbbi működésének alapján történt. Vagyis végig kellett gondolni, hogy milyen adattáblákra, oszlopokra lesz szükség. Maga a feladat röviden, lényegében: *“Digitális dokumentumokat lehessen bérelni, kredit pontokért cserébe”*.

- Tehát szükségünk lesz egy olyan adattáblára, amely tárolja az adott digitális dokumentum, könyv összes adatát.
- A felhasználóknak szükségük van egy fiókra, hogy össze lehessen kötni, hogy melyik felhasználónak milyen könyvek vannak kibérelve, illetve az aktuális kredit pontjaikat is menteni kell valamilyen módon. Magyarán szükség van egy felhasználói adattáblára, ami a fiók adatait tárolja.
- Miután megvan a megoldás a felhasználói adatok, és a könyv adatok tárolására, már csak a bérletek mentését kell kitalálni. Vagyis biztosan kelleni fog egy tábla az adatbázisban, amely az összes bérletet tárolja.
- Illetve a feladat része lesz egy adminisztrátori felület is, ami azt jelenti, hogy egyes felhasználóknak joguk lesz a szerverre könyveket feltölteni, és törölni azokat. Ezt egy külön táblával oldanám meg, hogy melyik felhasználó tartozik ide, és nem a felhasználói adatok között lesz mentve, az egyszerűbb adatkezelés végett.

Szóval összesen négy darab adattábla lesz, amik angol nyelven kerülnek elnevezésre:

1. users: A felhasználói adatok mentésére.
2. books: A feltöltött könyvek adatainak a tárolására.
3. rents: Ez a tábla felel az összes létrejött bérlet tárolásáért.
4. admins: Itt pedig akkor az adminisztrátori joggal rendelkező felhasználók lesznek mentve.



A következő lépésben azt kell kitalálni, hogy az egyes adattáblákban pontosan milyen oszlopok szerepeljenek, és azok az oszlopok milyen tulajdonságokkal rendelkezzenek.

Hasonlóan a táblák megalkotásához, itt is a feladat végleges működésére kell gondolni.

## “users” tábla

A felhasználónak regisztrálnia kell, vagyis szüksége van olyan adatokra, amivel létrehozza fiókját, majd később hozzá is fér ahhoz, a már megadott adatokkal. Ez a hagyományos felhasználónév, jelszó kombináció lesz. Ehhez két külön oszlop kell, egy “name”, és egy “password”. Továbbá bekerül egy “email” oszlop is annak érdekében, hogy legyen a felhasználónak egy elérhetősége, illetve, ha elfelejtené valamelyik bejelentkezési adatát, az e-mail címe segítségével lehetősége lenne visszaszerezni a hozzá férését.

A felhasználói adatok után már csak a kreditrendszer kigondolása van hátra a “users” táblában. Biztosan kelleni fog egy “credits” oszlop, ami az aktuális kredit egyenleget foglalja magába. A feladat úgy szól, hogy minden nap egy új kredit pont kerül az egyenlegre. Ezt úgy oldottam meg, hogy bekerült egy “reg\_date” oszlop, ami a regisztráció pontos dátumát menti, így minden bejelentkezésnél kiszámolható, hogy mennyi nap telt el azóta, hogy regisztrált a felhasználó, és annyi kredit pont jár neki.

- Ez a megoldás problémát vet fel:
  - Minden egyes bejelentkezésnél a felhasználó megkapná mindig azt a kredit mennyiséget, amennyi az eltelt napok száma a regisztrálástól.
- Megoldás a hibára:
  - Egy új oszlop a táblában a “given\_credits”, ez az oszlop menti a már kiadott napi kredit pontjait a felhasználónak. Tehát bejelentkezéskor ez egy lekérdezendő adat lesz, és az eltelt napok számából ezt az adatot kikell majd vonni, így a napi kreditrendszer teljesen működőképes lesz.

Végül az egész adatsor elejére bekerül egy automatikusan növekvő, elsődleges kulcs, ami a felhasználónak a saját egyedi azonosítója lesz, “id” néven.

Az oszlopai a users táblának: “ id, name, password, email, credits, given\_credits, reg\_date ”

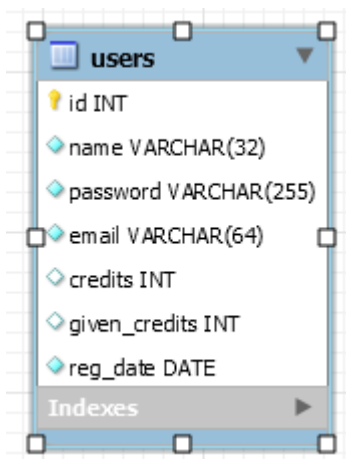
Már csak a tulajdonságaikat kell beállítani és elkészül teljesen a tábla.

- Id: Automatikusan növekvő egész szám, ebből adódóan egyedi is, és nem lehet üres. Elsődleges kulcs.
- name: Szöveges, maximum 32 karakter hosszú, egyedi, és nem lehet üres.
- password: Szöveges, maximum 255 karakter hosszú a titkosítás miatt. Nem lehet üres!



## Online könyvkölcsönző

- email: Szöveges, maximum 64 karakter hosszú, nem lehet üres.
- credits: Egész szám.  
given\_credits: Egész szám.
- reg\_date: Date típusú adat oszlop, és nem lehet üres.



### “books” tábla

Ennél a táblánál végig kell gondolni, hogy milyen adatokat mentenénk egy könyvről. Alapvetően van egy címe, amire a neveként lesz hivatkozva, tehát lesz egy “name” oszlop. A címén túl fontos adat szokott lenni még a szerző, így szükség lesz egy “author” oszlopra. A könyveknek szokott lenni egy összefoglaló leírása, hogy miről szól, milyen információkat tartalmaz, hogy az olvasó mit várjon, mit kap. Ezt egy “describe”, az az leírás nevű oszlopban tárolnám.

Lényegében ezzel megvan minden, amit egy könyvről tárolni kell, most kellenek az igazán technikai adatokra vonatkozó oszlopok.

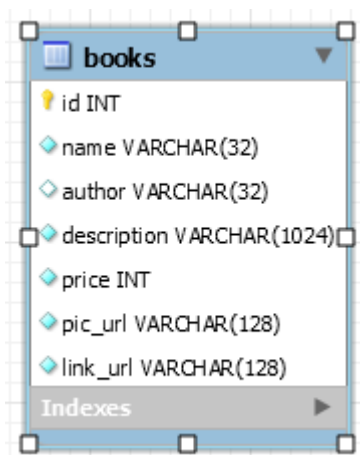
A könyvek a felhasználó kreditpontjaiból bérelhetőek. Így szükség lesz egy oszlopra, ami az árát jelöli, ez lesz a “price”. A könyvek a szerverre lesznek feltöltve, és a borítójuk is. Ez még két oszlopot kíván. Az egyik a “pic\_url” lesz, ami a szerveren lévő elérési útvonal lesz a borítóhoz, illetve a “link\_url”, ez pedig magának a pdf állománynak az elérési útvonala. Illetve, ahogy a “users” táblánál, úgy itt is lesz egy “id” oszlop, ami a saját számszerű azonosítója lesz az adott könyvnek / adatsornak.

Összesítve a tábla oszlopai: “id, name, author, description, price, pic\_url, link\_url”

- Id: Automatikusan növekvő egész szám, ebből adódóan egyedi is, és nem lehet üres.  
Elsődleges kulcs.

## Online könyvkölcsönző

- name: Szöveges, maximum 32 karakter hosszú, egyedi, és nem lehet üres.
- author: Szöveges, maximum 32 karakter hosszú.
- description: Szöveges, maximum 1024 karakter hosszú, és nem lehet üres.
- price: Egész szám, és nem lehet üres.
- pic\_url: Szöveges, maximum 128 karakter hosszú, nem lehet üres.
- link\_url: Szöveges, maximum 128 karakter hosszú, nem lehet üres.



### “rents” tábla

A rents tábla felel az összes bérlet mentéséért. Egy adatsor, egy bérlet. Arra kell csak gondolni, hogy mi alapján köthető össze, hogy kinek milyen bérlete van. Szóval biztosan kell egy oszlop, ami a bérletet azonosítja, és biztosan kell egy, ami azt azonosítja, hogy a bérlethez melyik könyv tartozik. Ezek lesznek a “users\_id” és a “books\_id” sorok, biztosan idegen kulcsok lesznek, hiszen a felhasználó és a könyv “id” sorai a tartalmuk.

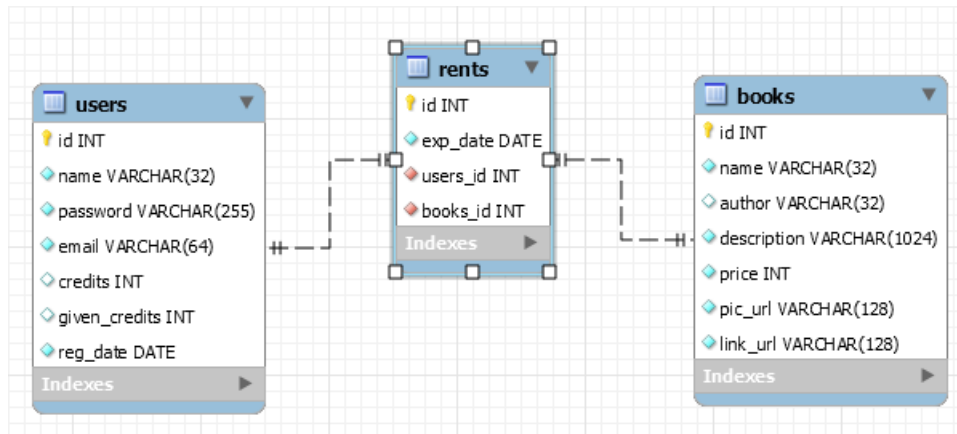
Ezen felül mivel bérletéről beszélünk, így annak van egy lejárat ideje is, amit szintén célszerű itt külön oszlopban tárolni, ez lesz az “exp\_date” oszlop. Végül, ahogy eddig is, lesz minden adatsornak egy saját “id” nevű azonosítója.

Összesítve a tábla oszlopai: “id, exp\_date, users\_id, books\_id”

- Id: Automatikusan növekvő egész szám, ebből adódóan egyedi is, és nem lehet üres. Elsődleges kulcs.
- exp\_date: Date típusú, nem lehet üres.
- users\_id: Idegen kulcs, a users táblából a felhasználó azonosítója (egész szám).

## Online könyvkölcsönző

- books\_id: Idegen kulcs, a books táblából a könyv azonosítója (egész szám).



### “admins” tábla

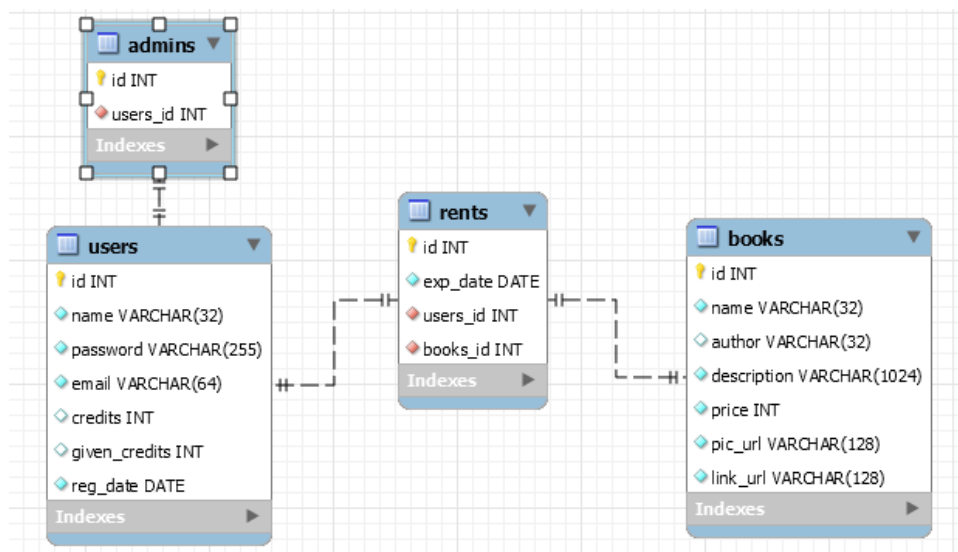
Végül a tábla, ami azt tárolja, hogy melyik felhasználónak van adminisztrátori joga a felületen. Ehhez nem kell mást menteni ide, mint magát a felhasználót.

Egy oszlop jön ide, ami a sor azonosítója, ez lesz a szokásos “id”, illetve egy idegen kulcs, ami pedig a users tábla id-je.

Összesítve a tábla oszlopai: “id, users\_id”

- Id: Automatikusan növekvő egész szám, ebből adódóan egyedi is, és nem lehet üres. Elsődleges kulcs.
- users\_id: Idegen kulcs, a users táblából a felhasználó azonosítója (egész szám).

Ezzel maga az adatbázis elkészült!

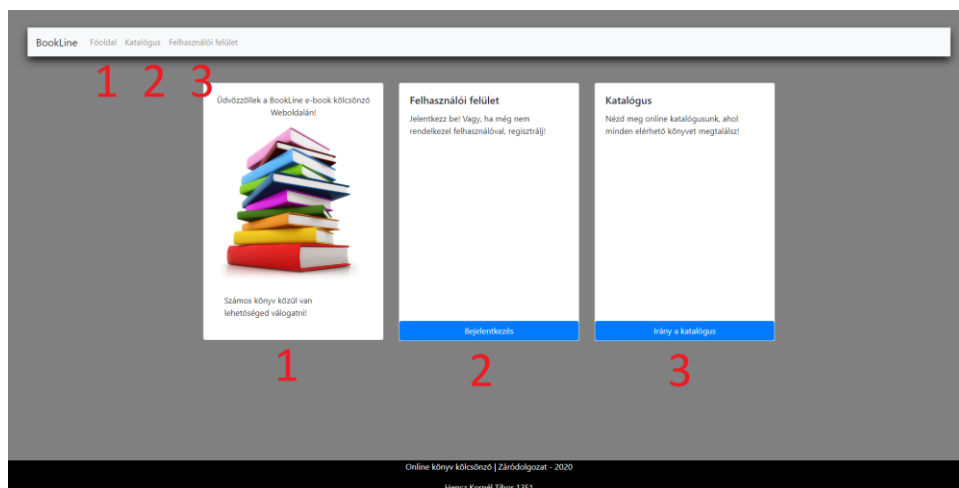


(Adatbázis: “bookline” - utf8\_hungarian\_ci)

## A szoftver megvalósítása

### index

Első lépésként egy letisztult index oldal létrehozása volt a cél, amin a felhasználó egyszerűen, és gyorsan képes navigálni, átláthatóságából, és felhasználóbarátságából kifolyólag.



Ez az oldal egy egyszerűbb HTML kóddal van leírva a böngésző számára. A menüpontok pedig a Bootstrap Navbar, illetve Card CSS rendszerének segítségével lettek kialakítva, megformázva.

Az érdekessége az index oldal HTML kódjának, hogy amint a felhasználó megnyitja azt a böngészőjéből, a kód azonnal leírja neki mind a három számozott menüpont tartalmát, de egy JavaScript függvény segítségével a főmenüt leszámítva, a másik két menüpont láthatóságát blokkolja.

A függvény működése egyszerű. A navigációs sávban, és a főoldalon látható "kártya" menüpontok hívják meg azt, "onclick" eseményükkel. Tehát rákattintást követően lefut a függvény, és az attribútumában átadja azt, hogy melyik számozott menüpont legyen látható, a többinek a láthatóságát pedig blokkolja.

Ezzel a megoldással lényegesen gyorsított a fő oldal felületének működése, hiszen szinte nincsen semmi töltési idő.

A főoldalt leszámítva a menüpontok egy-egy konténert tartalmaznak, a konténerek pedig egy-egy "iframet". Viszont még mielőtt a katalógus, vagy a felhasználói felület megtervezésébe belekezdtem volna, fontosnak tartottam, hogy készítsek egy php scriptet.

## A szerver oldal

Az adatbázis a szoftver egyik legfőbb magja, és az adatokkal, amikkel dolgozik, azok nagy részét abból kérdezi le. Ebből adódóan a megvalósítás során rengeteg lekérdezésre lesz szükség. Viszont, hogy a php tudjon adatbázisból lekérdezni előtte definiálni kell neki egy kapcsolatot. Ennek a kapcsolatnak a leírását minden egyes külön állományban elkerülendő, írtam egy "sql\_connect.php" fájlt, amit a későbbiek során, ahol szükség van a kapcsolatra, ott "include"-ként felhasználhatom, és csupán egy darab sort fog igénybe venni ez a kapcsolódás.

Másik oldalról ez a megoldás azért hasznos, mert egyszerűbben portolható a szoftver másik szerverre. Vagy csak másik SQL szerverre, hisz így csupán egyetlen helyen kell megadni az adatbázis címét, és egyéb adatait. Ezáltal egy magasabb szintű felhasználó is gond nélkül eltudja végezni ezt a folyamatot, bármiféle programozási ismeret nélkül.

A következő lépés a felhasználói felületen kívül lévő katalógus elkészítése volt. Ennek a katalógusnak a kiírása teljes mértékben szerver oldalról érkezik. Ez azt jelenti, hogy a katalógus menüvel behozott iframenek a forrása egy olyan php fájl, ami indít egy lekérdezést az adatbázisban, ami lekéri az összes könyv, összes adatát. Az eredményként kapott sorokat, ciklus segítségével asszociatív tömbbe rendezhetjük. Majd a tömb elem index számával tudunk rá hivatkozni az adott adatra.

Ebben az esetben mivel lekértünk minden elemet, az adattábla oszlopain megyünk végig balról jobbra. (pl tömb[nulladik eleme] : könyv azonosítója, tömb[első eleme] : könyv neve)

```
include 'sql_connect.php';
$sql = "SELECT * FROM books";
$result = $conn->query($sql);
if ($result->num_rows > 0)
{
    echo "<table><tr>";
    $index = 0;
    while($row = $result->fetch_assoc())
    {
        echo IDE JÖN A HTML beágyazott kiírás;
        $index++;
        if($index == 3) { echo "</tr><tr>"; $index = 0; }
    }
    echo "</tr></table>";
} else { echo "<center>A katalógusunk egyenlőre üres!</center>"; }
$conn->close();
```

Ezután a cikluson belül kitudjuk írni egyesével a könyveket az "echo" függvény segítségével, de, hogy formázva legyen a kiírás az "echo"-ba beleágyaztam HTML kódot is, ami táblázatba állítja, és a Bootstrap Card rendszerébe foglalja a lekért

adatokat, például a könyv címét, leírását, és borítóképét.

## Modulokra bontás

Lényegében az eddigiekkel elkészült a felület, ami nem igényel külön regisztrációt. Most következik maga a felhasználói "felület". Ahhoz, hogy ez létrejöjjön modulokra bontottam a rendszert. Ahogy az adatbázis megtervezésénél, úgy itt is végig kell gondolni, hogy milyen modulokra lesz szükség.

Elsősorban biztosan szükség lesz egy regisztrációs, illetve bejelentkezési rendszerre. Mind a kettőnek a működése roppant egyszerű. Mind a kettő egy-egy külön php állomány, különböző HTML leírás, CSS formázás által kiegészítve. A HTML kódjukban található úgynevezett "form", aminek a tartalma olyan szöveges mezők, amik regisztráció vagy bejelentkezés során szükségesek. Például: Felhasználónév mező, Jelszó mező. Ezen mezők tartalmát POST metódus segítségével megkapja a PHP, és dolgozni kezdetünk velük. Megvizsgálhatjuk, hogy az adott adatok megfelelőek-e, ha igen elkezdhet a PHP lekérdezni. Regisztráció esetén azt kell megnézni az adatbázisán, hogy a "users" táblában, van-e olyan sor, aminek a "name" vagy az "email" mezője megegyezik-e a felhasználó által beküldött adattal. Abban az esetben, ha van, hiba üzenetet kell visszaadni a felhasználónak, hogy az általa választott adat már foglalt. Ha pedig nincs, akkor tulajdonképpen a regisztráció maga sikeres. A bejelentkezésnél is hasonlóan kell gondolkodni. A beküldött adatokat (név, jelszó) ellenőrizni kell, hogy van-e ilyen sor táblában, ami ezeket az adatokat tartalmazza, ha van ilyen, akkor sikeres a belépés. Ha nincs, akkor hibát kell visszaadni.

```
felhasználói adatok = POST metódus által beküldött értékek

ha<megfelelnek a felhasználói adatok>
    akkor
        bejelentkezés(felhasználói adatok);
    különben
        Hiba üzenet;

függvény bejelentkezés (felhasználói adatok)

    <eredmény> = <lekérdezés a felhasználói adatok alapján>

    ha <eredmény>
        akkor
            Sikeres bejelentkezés;
        különben
            Sikertelen bejelentkezés;
```

A következő modul a külön felhasználói felület. A kivitelezés célja az volt, hogy ha a felhasználó kiválasztja felhasználói menüpontot, akkor az eredeti felületen belül megjelenjen a saját felhasználói felülete is. Amennyiben nincsen neki fiókja, vagy nincsen bejelentkezve, akkor kérje fel arra, hogy lépjen be egy létezőbe, vagy hozzon létre egy újat.

- Megoldás: A PHP saját "Session" rendszere segítségével tudjuk azonosítani az eszközöket. Ha a felhasználónak a session-ben van beállítva felhasználónév, tehát már bejelentkezett, akkor a felület jelenik meg. Ha nincs, akkor pedig a bejelentkezési felület, ahol egy gomb segítségével váltani tud a regisztrációs felületre.

A felhasználói felület felépítése egyszerű. A PHP echo függvényével HTML formokkal beágyazott gombokat/szöveget ír ki, sorban. Balról jobbra: Könyvek, Bérelt könyveim, Kredit szám, Felhasználónév. A kredit szám kiíráshoz csak egy adatbázis lekérdezés tartozik, míg a felhasználónévéhez a Sessionben található név.

A "Könyvek", illetve a "Bérelt könyveim" pontok foglalnak magában egy-egy külön modult. A "Könyvek" egy olyan állományra hivatkozik, ami egy Bootstrap táblázat segítségével kilistázza az összes könyvet, azoknak az árát, illetve könyvenként egy gombot. A gomb pedig újabb modul, ami egy olyan oldal, ahol lekérdezésre kerül, hogy melyik könyv mellett szerepelt, és annak a könyvnek az összes adatát táblázatosan kiírja, és itt ezen a modulon keresztül van lehetőség kibérelni azt. A bérlet gombra kattintva lefut a bérleti modul, vagy is lekérdezésre kerül, hogy van-e elegendő kreditje a felhasználónak, illetve, hogy még nem-e bérli a könyvet. Ha megfelel a feltételeknek, akkor a "rents" táblába felveszi a bérletet egy új sorként.

Vissza a térve a "Bérelt könyveim" menü moduljára, ez szintén egy Bootstrap táblázat segítségével kiíratott menü, ahol indul egy SQL lekérdezés, az összes olyan bérletre a "rents" táblából, ami a felhasználóhoz tartozik, és még nem járt le. Ezeket a bérleteket kiírja, és ellátja egy "olvasás" gombbal. Az olvasás gomb pedig elvezet az utolsó felhasználói modulhoz. Ez a modul csupán annyit tesz, hogy megnyitja a szerveren található pdf fájlt, a könyvhöz tartozó elérési út alapján (link\_url adatoszlop a "books táblában").

Amennyiben az adott felhasználó felvan véve az adatbázisban, az adminisztrátori joggal rendelkező személyek közé, akkor a felhasználói felületen az echo függvény két újabb

menüpontot irat ki. A “Könyv felvétele”, és “Könyv törlése”. A könyv felvételi modul működése nagyon hasonló a regisztrációéhoz. HTML formon belüli szöveges, és fájl feltöltő mezők, amiket a PHP-nak továbbít POST módszerrel, a PHP pedig feltölti őket az adatbázisba, a fájlokat (borító, pdf) pedig közvetlenül a szerverre. A könyv törlése modul pedig a katalógushoz hasonlít, kilistázza a könyveket egy lekérdezést követően, hozzáfűz HTML kód segítségével egy törlő gombot, amire kattintva a könyv eltűnik az adatbázisból, és a hozzátartozó a fájlok a szerverről.

```
//USER PANEL FELŐ MENÜBAR
echo "<div class='container'>";
echo "<div class='row'>";
//--
echo "<div class='col-sm'>";
echo "<a href='user_catalog.php' target='useriframe'>Könyv bérlet</a>";
echo "</div>";
//--
echo "<div class='col-sm'>";
echo "<a href='user_rentedbooks.php' target='useriframe'>Bérelt könyveim</a>";
echo "</div>";
//--
echo "<div class='col-sm'>";
echo "<p>".$_SESSION['user_name']."' - ".fcredit_layer." kredit | <a href='user_logout.php'>Kijelentkezés</a> |</p>";
echo "</div>";
//--
if(isset($_SESSION['user_admin']))
{
    echo "<div class='col-sm'>";
    echo "<a href='book_upload.php' target='useriframe'>Könyv feltöltése</a>";
    echo "</div>";
    echo "<div class='col-sm'>";
    echo "<a href='book_remove.php' target='useriframe'>Könyv törlése</a>";
    echo "</div>";
}
echo "</div>";
echo "</div>";
echo "<div class='embed-responsive embed-responsive-4by3'><iframe class='embed-responsive-item' src='user_catalog.php' name='useriframe'></iframe></div>";
```

Ezekből a modulokból áll össze maga a szoftver a maga!

Állományok:

- Bookremove : A könyv, törlés modulja.
- Bookrent: A könyv, bérlet modulja.
- Bookupload: A könyv, feltöltési modulja.
  - Catalog: Az index oldali katalógus.
- Footer: A weboldalon található footer.
  - Index: A főoldal.
  - Login: A bejelentkezési modul.
  - Msgerr: Sajátos kiíratási függvény.
  - Read: A könyv megnyitási modulja.
  - Register: A regisztrációs modul
- Remove: A könyvtörlés közvetlen algoritmus
- Sql\_connect: Az adatbázis kapcsolat leírása.
  - User: A felhasználói felület.
- User\_bookinfo: A felhasználóifelületen található bérleti felület.
- User\_catalog: A Felhasználóifelület katalógusa.



- User\_logout: Kijelentkezési modul (sessionöket szétrombolja).
- User\_rentedbooks: A felhasználói felületen "Bérelt könyveim" modul.

## Tesztelés

A program a készülése közben folyamatosan tesztelés alatt volt, így a végleges tesztelésre az esetleges hibák száma jelentősen csökkenhetett.

### Fehér doboz

A komplexebb tesztelés előtt, a program sűrűbb elágazásain mentem végig, sorban. Rögtön az első ilyen a regisztrációs modulban az ellenőrzés, hogy a felhasználó által megadott adatok megfelelnek-e az elvárásoknak. Amiket itt ellenőriz a program:

- Minden mező kivan töltve.
- A megadott e-mail cím valós formátumú.
- A felhasználónév, és a jelszó is legalább 4 maximum 32 karakter hosszú.
- A Jelszó megegyezik azzal, amit másodjára adott meg a felhasználó.
- A felhasználónév szabad.
- Az e-mail cím szabad.

Mindegyik pontra szándékosan vétettem egy-egy hibát, és minden esetben a megfelelő hibaüzenet fogadott!

A bejelentkezésnél kizárólag annyi kritériumot hoztam, hogy nem lehetnek üresek az adatmezők. Ennek megfelelően, ha nem adtam meg jelszót, vagy felhasználónevet, a megfelelő kimenetet kaptam. "Név és jelszó mező kitöltése kötelező!" Abban az esetben, ha szándékosan elírtam a felhasználónevet vagy a jelszót, szintén a megfelelő kimenet érkezett! "Hibás felhasználónév vagy jelszó!"

Hasonló rendszer a könyvfeltöltés, ahol szintén sok ellenőrző elágazás van.

- A borítókép, és a könyvfájl nem létezik eddig a szerveren.
- A borítókép csak jpg, jpeg, png vagy gif lehet.
- A könyv maga kizárólag pdf formátumú lehet.
- Egyik adat mező se üres.
- A könyv neve legalább 8 maximum 32 karakter hosszú.
- A leírás minimum 10 maximum 1000 karakter hosszú lehet.

- Az ára csak egész szám lehet.

Az összes ponton végig haladva hibákat produkáltam szándékosan, és minden alkalommal a megfelelő hiba üzenetet kaptam, "A borítókép, és a könyvfájl nem létezik eddig a szerveren." kivételével. Ugyanis feltudtam tölteni ugyan azt a dokumentumot, és az előző fájlt pedig felül írta.

- Hiba oka: A fájl leendő elérési útját egy php változó tárolja. Az ellenőrzés nem erre a változóra szólt, hanem egy már nem használt, ami hibás volt.
- Megoldás: a "file\_exists()" függvény attribútumában a változó átírása a megfelelőre, a régiek pedig teljes eltávolítása a kódból. A tesztnek köszönhető további előny, hogy kikerült olyan felesleges kód részlet, ami véletlenül bent maradt.

### Fekete doboz

Miután átnézésre került szándékosan a kód alapján a program, az elejétől a végig átnéztem minden funkcióját egy átlagos felhasználó szemszögéből.

Ellátogattam a főoldalra, és minden menüpontot többször is, véletlenszerűen végig nyitogattam. Ezt követően a felhasználói felületre lépve, fiókot regisztráltam, általánosnak mondható adatokkal. Megkaptam az öt kredit pontomat, kijelentkeztem. Az adatbázisban átírtam a regisztráció dátumát kettő nappal korábbra, ezzel szimulálva a napok elteltét. Ismételten bejelentkeztem és már látható is volt, hogy megkaptam a plusz két darab kredit pontot. Megnyitottam a Könyv bérlet menüpontot, és találómra választottam két könyvet a tesztből feltöltött dokumentumok közül. Átnéztem, hogy az "info" menüpontban látható a könyv borítóképe, címe, leírása, és itt ki is tudtam őket bérelni. A bérlet menüpontra kattintva, láttam mind a kettő könyvet, amit béreltem, és mind a kettőt megtudtam nyitni, és olvasni.

Felhasználói szemszögből már csak az adminisztrátori modulok tesztelése volt hátra. Felvettem a teszt fiókot az "admins" adattáblába, és bejelentkezés után láttam is az adminisztrátori menüket. Egyszerűen tudtam közöttük lépkedni. Feltöltöttem egy könyvet, egy másikat pedig töröltem. A feltöltött könyvet kitudtam bérelni, a kredit pontjaim levonásra kerültek, és tudtam olvasni szabadon a dokumentumot.

### Egyéb tesztek

- A teszt fiókba ugyan azon a napon többször is beléptem, annak érdekében, hogy teszteljem, hogy nem-e kapja meg a felhasználó többször is a kredit pontokat, amiket már egyszer megkapott.
- Különböző speciális karakterekkel is megpróbáltam regisztrálni, mint például “الْحُرُو” vagy “¥”. Sikeresek voltak ezek a próbálkozások. Az adatbázis megfelelően fogadta az adatokat, és a fiókok is tökéletesen működnek, helyesen írja ki a fiók nevet is az echo függvény.
- Feltöltöttem 9 darab teszt pdf fájlt, hogy a sima katalógus megfelelően táblába rendezi-e őket.
- Teszteltem, hogy a megfelelő kredit pont mennyiség nélkül lehet-e bérelni.
- Megnéztem, ha Adminisztrátorként törölök egy könyvet, eltűnik-e a szerverről, az adatbázisból. Illetve, hogy megszűnik minden bérlet, ami az adott könyvhöz tartozott.

Összeségében a végső tesztelés során nem ütköztem kritikus hibába. Az eredmények jók voltak, a szoftver pedig működőképes és használható.

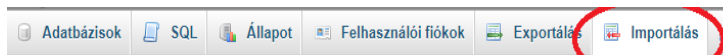
### A rendszer beüzemelése php, és sql szerverrel

Rövid leírás arról, hogyan kell a szoftvert beüzemelni, haladó felhasználók számára! Két állománya van a kész programnak. Egy tömörített, rar fájl, és egy sql fájl. Az sql fájlt az sql szerver menedzserében be kell importálni, ezzel az adatbázis maga már kész is van. A tömörített állományban található php, és egyéb fájlokat, mappákat a webtárhelyre ki kell csomagolni.

Ezt követően a kicsomagolt fájlok közül az “sql\_connect.php” -ban az adatbázis, és sql szerver adatait át kell írni a megfelelőre, és gyakorlatilag teljesen működni fog az egész program!

Illusztráció:

## Online könyvkölcsönző



### Importálás a jelenlegi kiszolgálóra

#### Az importálandó fájl:

A fájl lehet tömörített (gzip, bzip2, zip) vagy tömörítetlen.

A compressed file's name must end in: [típus].[compression]. Example: .sql.zip

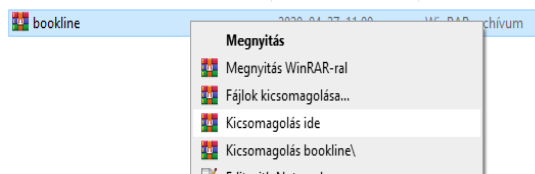
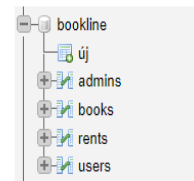
Számítógép tallózása:  database.sql (Max: 10MB)

Továbbá behúzhat és bejelölhet egy fájlt bármely oldalra.

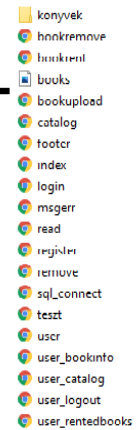
A fájl karakterkészlete:

#### Részleges importálás:

+  =



=



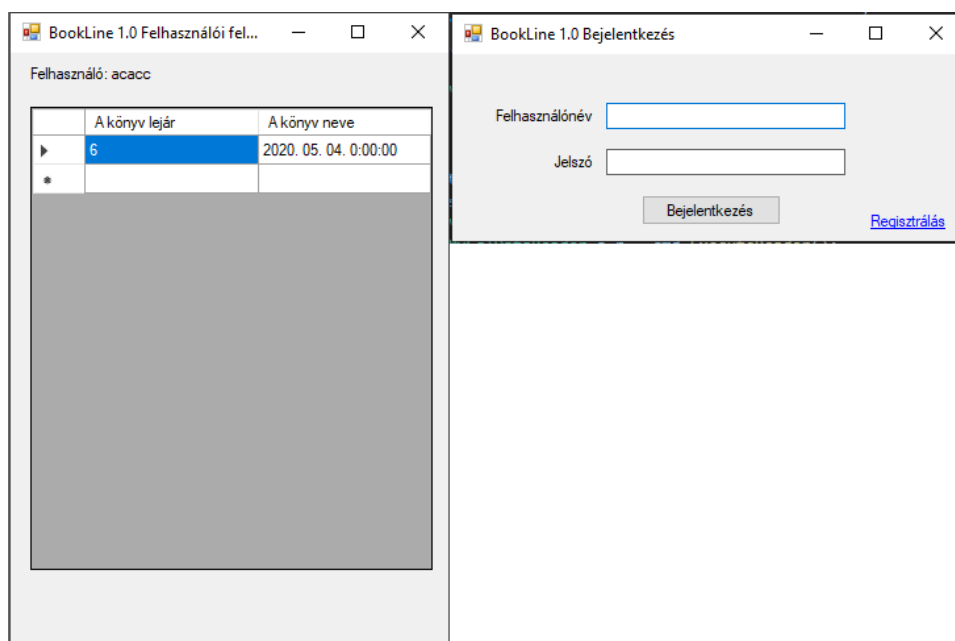
```
$sql_server = "localhost"; // Az sql szerver címe \  
$sql_usn = "root"; // Felhasználónév |-- EZEK AZ  
$sql_pw = ""; // Jelszó |-- ÁTÍRANDÓK  
$dbname = "bookline"; // Adatbázis neves /
```

sql\_connect.php

## Továbbfejlesztési lehetőségek

Habár összességében átlátható, és egyszerű felülete van a könyvkölcsönzőnek, a jövőben mindenképpen szeretném a külsőjét, modernebbé, részletgazdagabbá, szebbé tenni. Maga a felhasználói felület, túlságosan letisztultnak hat.

Az ötlet kitalálásakor tervben volt egy asztali felület is, ami annyiban kiegészítette volna a webest, hogy ott a már kibérelt könyveket lehetett volna olvasni. A szoftver elkészült, de idő hiányában túlságosan letisztult. A felhasználói fiókba lehet vele bejelentkezni, és a szerverről megnyitja az adott pdf fájlt böngészőben.



Végül ebben a formában feleslegesnek tartottam a záródolgozat részévé tenni, lényegében nem tud hozzá adni a végeredményhez. Viszont a későbbiekben szeretném úgy kialakítani, áttervezni, hogy egyenértékű legyen a webes felülettel. Lehesen bérelni, elérhető legyen az adminisztrátori felület. Illetve itt is az egyik legfontosabb szempont, hogy korszerűbben, modernebben nézzen ki.

Szeretnék egy felhasználó támogatási, helpdesk felületet, ahol az adminisztrátorokkal fel lehet venni a kapcsolatot, segítséget lehet kérni.

Ebből adódóan fontosnak érzem még az adminisztrátori funkciók bővítését. Tervezem, hogy az ezzel a joggal rendelkező felhasználók tudják állítani a többi felhasználó kreditpont számát, tudjanak manuálisan bérletet létrehozni, megszüntetni.

A legfontosabb funkció, ami még hiányzik a kész programból, hogy az adminisztrátorok a már meglévő könyvek adatain, fájljain, árán tudjanak módosítani. Jelenleg, ha egy könyv rosszul kerül feltöltésre nincs lehetőség a módosításra az interfészből, maximum adatbázis kezelőből. Ez kellemetlenséget okozhat, hiszen törölni kell, majd újra létrehozni, minden adat újbóli megadásával.

## Irodalomjegyzék, Források

A webfelület formázására a Bootstrap CSS rendszere által nyújtott lehetőségek lettek használva! A főoldalon található "kártyás menü", a navigációs sáv, és a táblázatok!

Weboldal: <https://getbootstrap.com/>

Egyéb információs forrás: <https://www.w3schools.com/>

Szoftverek: <https://www.apachefriends.org/>

<https://notepad-plus-plus.org/>

<https://www.mysql.com/products/workbench/>

## Felhasználói dokumentáció

### A program általános specifikációja

A "Bookline" fantázianévvel ellátott online könyvkölcsönző arra kínál lehetőséget, hogy a felhasználók ingyen olvashassanak különböző fajta digitális könyveket, akárhol, webes környezetben. A weboldalra ellátogatva található egy katalógus, ami az összes elérhető könyvet tartalmazza. Ez alapján a látogató eldöntheti, hogy szeretne-e fiókot regisztrálni. A regisztrálást követően öt kreditpont kerül jóváírásra a felhasználói egyenlegre. Az egyenlegre minden nap után a regisztrálást követően egy új pont íródik jóvá, más út nincs a kredit növelésére!

A kreditpontokból pedig a könyvek kibérlése lehetséges, minden könyvnek más "ára" van, és minden bérlet egy hétre szól. Vagyis a bérlet megkezdése után a hetedik napon még elérhető, de a nyolcadikon már nem!

### Rendszerkövetelmény

A szoftver használatához szükséges konfiguráció:

- Operációs rendszer: Windows 7, Windows 8, Windows 8.1, Windows 10 vagy későbbi.
- Processzor: Intel Pentium 4 processzor legalább.

Támogatott böngészők:

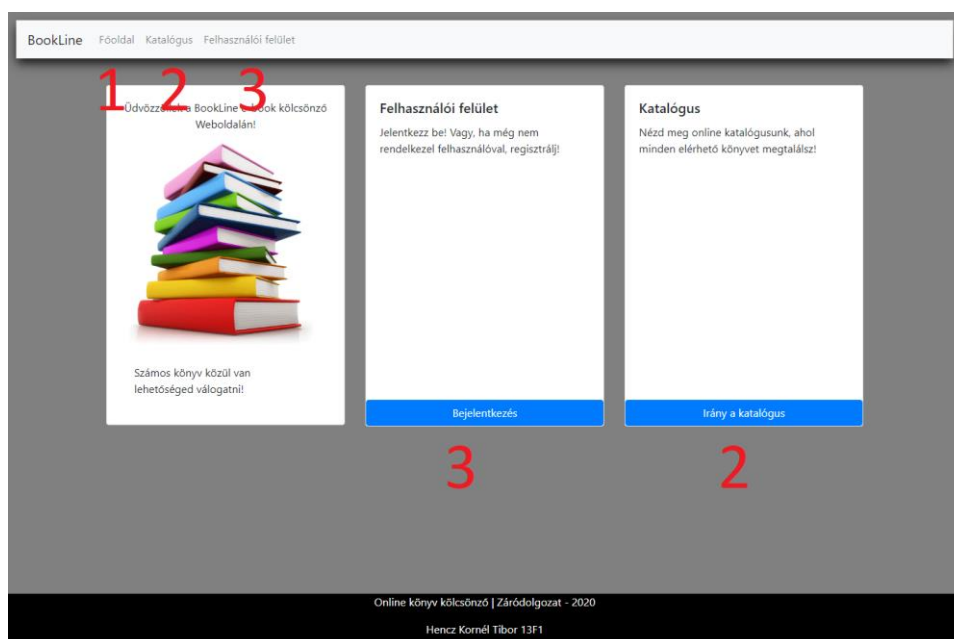
- Mozilla Firefox
- Google Chrome
- Microsoft Edge
- Safari
- Opera

### A program telepítése/elérése

Amennyiben rendelkezik a felhasználó a minimum hardverrel, és a felsorolt böngészők valamelyikével, akkor nincs más dolga, mint a weboldal címén keresztül elérni azt. (pl: [bookline.hu](http://bookline.hu))

Tipp: Ha gyorsítani szeretnénk az elérés folyamatán, van lehetőség az adott portálhoz parancsikont készíteni, például az asztalon. Annyit kell tenni, hogy a kívánt oldalt a könyvjelzők közé tesszük, majd az egérrel, bal gomb folyamatos nyomva tartása mellett kihúzzuk azt az asztalra

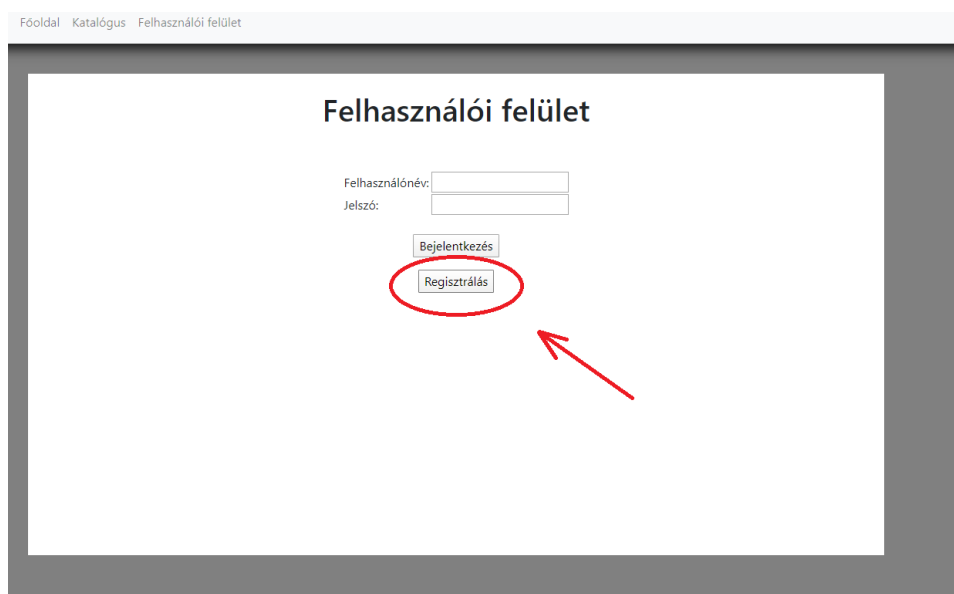
## Használati útmutató



Az ábrán a főoldal látható, a pirossal számozottak pedig a különböző menüpontok. Az egyes a főoldal, ami a képen is látszik, a kettes a katalógus, amit bárki megtekinthet bármikor, a hármas pedig a Felhasználói felület menüpontja. A menüpontok között a bal egérgombbal válthatunk.

### Regisztráció és Bejelentkezés

Amennyiben a katalógus átböngészése után úgy dönt a felhasználó, hogy szeretne a szoftver használója lenni, a Felhasználói felület menüpontot kell, hogy válassza. Itt az alábbival fog találkozni.





## Online könyvkölcsönző

Ezen a felületen fog tudni mindig belépni, de előtte fontos, hogy létrehozza a saját fiókját, a regisztrálás gombra kattintva.

Név:	<input type="text"/>	1
Jelszó:	<input type="text"/>	2
Jelszó ism:	<input type="text"/>	3
E-mail cím:	<input type="text"/>	4

A regisztráláshoz három adatot kell megadni.:

1. Egy nevet, ami lehet fantázia név is, csak fontos, hogy meg kell jegyezni!
2. Egy jelszót, amit kétszer kell megadni, hogy biztosan a kívánt karakterlánc legyen az!
3. Illetve egy e-mail címet, arra az estere, ha szándékon kívül megszűnne a felhasználó hozzáférése a fiókjához.

Amire fontos odafigyelni! A felhasználónév és a jelszó minimum 4-4 karakterből álljon! De a jelszónak ajánlatos legalább 8 darab karakterből állni, kis betűt, nagy betűt, számot, írásjelet vegyesen tartalmazva. A legbiztonságosabb, ha a saját belépési adatait senki nem osztja meg senkivel.

### Könyv bérlete

A sikeres regisztrációt követően, vissza kell térni a bejelentkezés oldalra (felső ábra), majd az újonnan regisztrált felhasználói adatokkal már be is lehet jelentkezni a kezelőfelületre.

## Online könyvkölcsönző

Könyv bérlet

Bérelt könyveim

TesztFioK - 5 kredit | [kijelentkezés](#)

Könyv Neve	Ára	Részletek
Harry Potter 1	1	<a href="#">Info/bérlés</a>
Harry Potter 2	1	<a href="#">Info/bérlés</a>
Harry Potter 3	1	<a href="#">Info/bérlés</a>
Harry Potter 4	1	<a href="#">Info/bérlés</a>
Harry Potter 5	1	<a href="#">Info/bérlés</a>
pdf teszt	1	<a href="#">Info/bérlés</a>

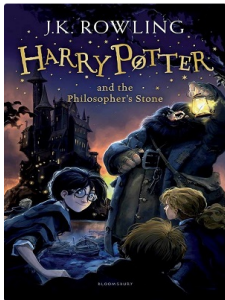
A bejelentkezés után ez fogadja a felhasználót. A Felső sávban található két menüpont, a felhasználónév, a kredit egyenleg, illetve egy kijelentkezés gomb. A Könyv bérlet menüpont van alpból megnyitva. Itt megtalálható az összes elérhető könyv, azok ára, valamint az Info/Bérlés gombra kattintva, elolvashatunk minden információt a könyvről.

BookLine Főoldal Katalógus Felhasználói felület

[Könyv bérlet](#)

[Bérelt könyveim](#)

TesztFioK - 5 kredit | [kijelentkezés](#)



Kibérlés

viSSza

Harry Potter 1 - Cím

Harry Potter első része, nagyon mókás könyv

- Leírás

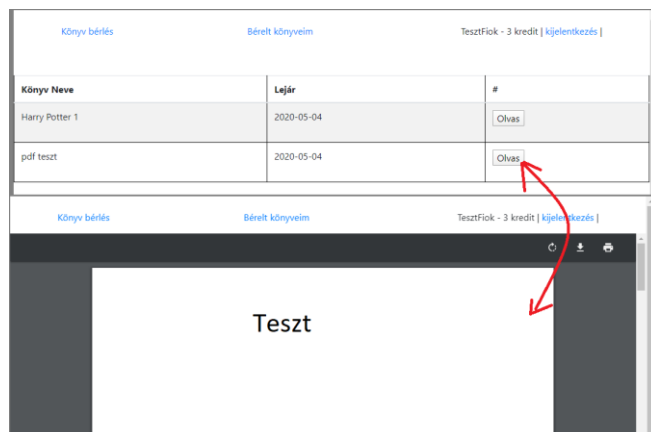
Itt két lehetőség van. Visszamenni, ha mégsem kelti fel az érdeklődést a könyv, vagy a “Kibérlés” gombra kattintva kibérelni azt egy hétre.

### Bérelt könyv olvasása

Ha a már bérelt könyvünket szeretnénk olvasni, akkor a fenti ábrán is látható “Bérelt könyveim” menüpontot kell választani. Ebben a menüben láthatja felhasználó, hogy a bérlete

## Online könyvkölcsönző

mikor jár le pontosan, a lejáratí idő mellett pedig található az “olvasás” gomb, ami megnyitja a weboldalon a könyvet.

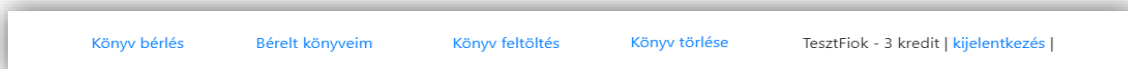


### Kijelentkezés a felületről

Kijelentkezni a felhasználói fiokból a fehér sáv, jobb felső sarkában található “kijelentkezés” gombbal kell. Ilyenkor, ha leközelebb jövünk vissza az oldalra újra meg kell adni a belépési adatokat. Célszerű bejelentkezve maradni olyan eszközön, amit csak egyedül használ a felhasználó, és nincs szüksége több fiokra a weben.

### Adminisztrátori felület útmutató.

Ha adminisztrátori joggal ellátott fiokkal vagyunk bejelentkezve, a felhasználói interfészen eleve kettővel több menüpont található.



### Könyv feltöltés

A fehér navigációs sávból válasszuk ki a “Könyv feltöltés” pontot.

## Könyv Feltöltés

Ár:	<input type="text"/>
Cím:	<input type="text"/>
Leírás:	<input type="text"/>
Szerző:	<input type="text"/>
Borítókép:	<input type="button" value="Fájl kiválasztása"/> Nincs fájl kiválasztva
Könyv:	<input type="button" value="Fájl kiválasztása"/> Nincs fájl kiválasztva
<input type="button" value="Feltöltés"/>	

Ezeket az adatokat kéri a rendszer a könyvről! Fontos, hogy odafigyeljünk a kitöltés helyességére, mert nem módosítható a már feltöltött könyv. Csak törölhető, és újra feltölthető az adatok újra megadásával.

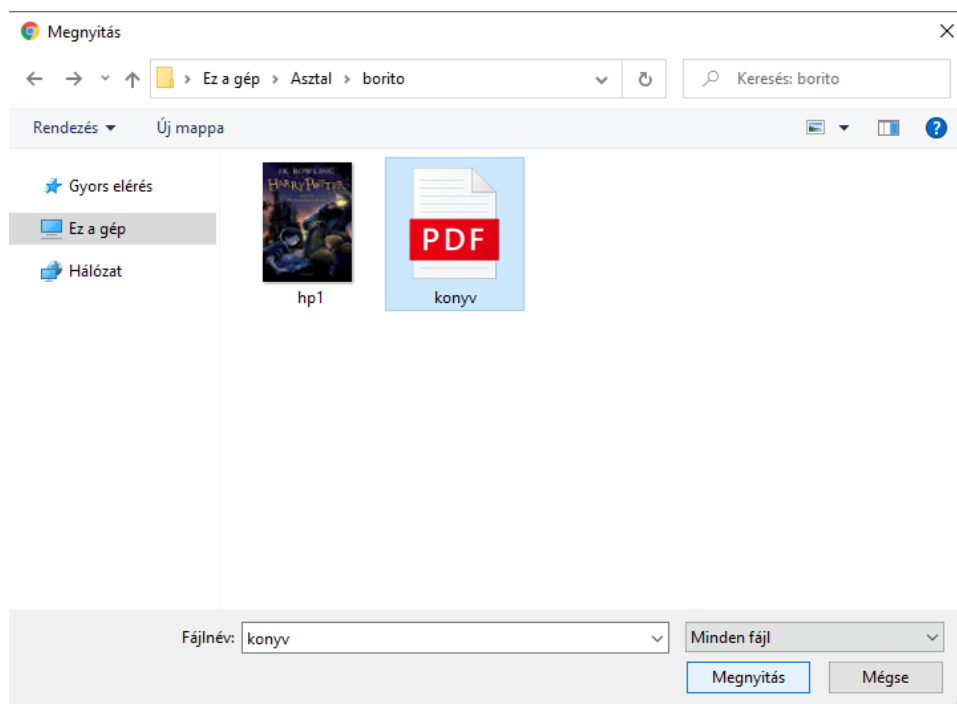
### Specifikációk, kritériumok:

- Ár: Csak és kizárólag egész szám lehet! Azt a mennyiséget jelöli amennyi kreditpontért cserébe a felhasználók kibérelhetik az adott könyvet egy hét időtartamra.
- Cím: A könyv vagy dokumentum pontos címe, ez lesz a könyv "neve", későbbi hivatkozások során. Ennek legalább 8 karakter hosszúnak kell lennie, maximum 32 karakter.
- Leírás: Egy hosszabb leírás a könyv tartalmáról, összefoglaló, hogy miről szól, milyen információkat tartalmaz, hogy az olvasó mit várjon, mit kap. Maximum 1000 karakter hosszú lehet!
- Szerző: A szerző, író neve.
- Borítókép: A könyv borítója, jpg, png, jpeg, vagy gif formátumban. Az ajánlott méret 300x200 pixel, de nincs megszabva pontosan!
- Könyv: Ide magát a pdf állományt kell, kitölteni. Azon kívül, hogy pdf-nek kell lennie, nincs más kritériuma.

## Feltöltendő fájl tallózása

A fájl kiválasztása gomb megnyomása után felugrik egy új ablak. Ez a fájlkezelő. Itt a felhasználónak a saját számítógépén vagy egyéb eszközén meg kell keresnie, hogy hol találhatóak a könyv állományai, amiket feltervez tölteni a szerverre.

Ha megvannak, akkor csak kikell választani őket, és "leokézni", majd ezután a feltöltés gombra kattintva, ha minden adat megfelel, akkor már be is kerül a rendszerbe az új könyv



## Könyv törlése

A "Könyv törlése" menüpontot választva egy táblázat jelenik meg. A táblázat elemei: könyv azonosítója, könyv címe, és egy "Törlés" gomb. A törlés gombot kiválasztva, a könyv törlődik a rendszerből. Fontos, hogy ezután már nem lehet visszaállítani!

Könyv ID	Könyv Neve	
6	Harry Potter 1	Törlés
7	Harry Potter 2	Törlés
8	Harry Potter 3	Törlés
9	Harry Potter 4	Törlés
10	Harry Potter 5	Törlés
11	pdf teszt	Törlés