



Vas Megyei Szakképzési Centrum  
Nádasy Tamás Technikum és Kollégium

# **SZAKDOLGOZAT**

## **Könyvtári Szoftver**

**Szakács Gergő**

**Konzulens:**  
**Balics Gábor**

**2022**

# Nyilatkozat

Alulírott, Szakács Gergő szakos hallgató kijelentem, hogy a Könyvtári Szoftver című szakdolgozat feladat kidolgozása a saját munkám, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtam fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeim saját munkán, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudásom szerint hitelesek.

Szakony, 2022.04.01.

---

hallgató

# Kivonat

## Könyvtári Szoftver

A szakdolgozatom célja a könyvtárak modernebbé és gyorsabbá tételére egy lépés, amely modernizálni tudná a már elavult könyvtári rendszereket.

A könyvtárba járók számára egy kényelmes és rugalmas megoldás lenne. Otthonról, buszról, vonatról bárholnan utánajárhat a könyveknek, majd lefoglalhatja, hogy a könyv mihamarabbi maximális élményben részesítse.

A könyvtárosoknak is egy kényelmesebb megoldást biztosítana. Nem kellene felesleges telefonokra válaszolni. Értesíteni tudnák a kölcsönzőket a könyvtár híreiről, eseményeiről.

# **Abstract**

## **Library Software**

The aim of my dissertation to make libraries more modern and faster is a step that could modernize already outdated library systems.

It would be a convenient and flexible solution for those who go to the library. You can search for books from home, bus, train from anywhere and book them so that the book can have the maximum experience as soon as possible.

It would also provide librarians with a more convenient solution. They shouldn't have to answer unnecessary phones. They could inform the customers about the news and events of the library.

# Tartalomjegyzék

<b>Bevezetés</b>	<b>7</b>
<b>Felhasznált technológiák</b>	<b>8</b>
Választott nyelv, segédeszközök	8
PHP	8
HTML	8
CSS	8
SASS	8
Bootstrap	8
Javascript	8
jQuery	9
Felhasznált programok	9
Visual Studio Code	9
MySQL Workbench	9
MAMP	9
Adobe XD	9
DrawSQL	10
<b>Felhasználói dokumentáció</b>	<b>10</b>
Hardware és Szoftver igény	10
Lehetőségek regisztráció előtt és után	10
Regisztráció, Belépés	10
Navigációs sáv	11
Főoldal és tartalma	11
Könyvek listája	12
Bármely könyv oldala	12
Profil felület	12
Könyvtáros felülete	13
Navigációs sáv	13
Hírek	13
Hozzászólások	13
Foglalások	14
Jelszómódosítás	14
<b>Fejlesztői dokumentáció</b>	<b>15</b>
Tervezés folyamata	15
Adatbázis tervek (Database)	15

Felhasználó felület tervek (User Interface)	15
A szoftver	16
Összegző	16
Struktúra (Fájlok rendezése)	16
Továbbfejleszthetőség	16
Biztonsági kérdések	17
Regisztráció / Belépés	17
Navigációs menü	18
Kezdőlap	19
Könyvek	19
Specifikált Könyv	20
Profil	20
Könyvtáros felület	21
Az adatbázis	22
Kolcsonzok(Felhasználók) tábla	22
Hirek(Hírek) tábla	22
Tickets(Hibajegyek) tábla	23
Konyvek(Könyvek) tábla	23
Lefoglalva tábla	24
Velemenyek(Vélemények) tábla	24
<b>Összegzés</b>	<b>25</b>
<b>Irodalomjegyzék</b>	<b>26</b>
<b>Mellékletek</b>	<b>26</b>

## Bevezetés

Véleményem szerint napjainkban a magyar könyvtárak már nem annyira zsúfoltak, mint régen. Ha valaki el akar olvasni egy könyvet, akkor megveszi (ha megteheti) vagy online formában hozzájut és digitális formában olvassa el a kívánt könyvet. Viszont, ahogy haladunk a korral, úgy a könyvtáraknak is haladni kell.

Nem is kell sokat visszarepülni az időben, ha belegondolunk, 5 éve teljesen más volt. Mikor voltam utoljára könyvtárban? Akkor hogyan kölcsönöztem ki egy könyvet? Teheti fel magának az ember a kérdést...

Manapság vagy nem járunk már könyvtárba vagy ha egyszer-egyszer beugrunk, akkor a könyvtárost kérjük meg, hogy adja ki számunkra a könyvet, amit oda vittünk neki.

Az elmúlt években egyre jobban elterjedtebbek lettek az online könyvtárak különböző verziói. Vannak olyan oldalak, ahol bérelhetünk digitális könyveket, vannak online webáruházak, ahonnan rendelhetünk magunknak könyvet.

Az emberek rohannak. Semmire nincs már időnk, ha valaki könyvet olvasva akar kikapcsolódni, akkor előbb szereznie kell valahonnan egy könyvet, hogy el tudja érni a kívánt élményt, melyet egy könyv nyújtani tud.

Mikor a szakdolgozati témát kerestem, egyből eszembe jutott, hogy a nagymamám úgy kölcsönöz ki magának könyvet, hogy előtte szól a könyvtárba telefon segítségével, hogy gyűjtsenek össze neki pár könyvet és akkor, amikor tud, érte megy.

Ez egy opcionális megoldás arra, hogy lefoglaljon könyveket, de mi lenne akkor, ha lenne erre egy kezelőfelület, ahol a könyvtár összes könyvét meg tudná nézni és akár le is tudná foglalni azokat, amelyek felkeltették az érdeklődését? Esetleg még véleményezni is tudná a már olvasott könyveket, amiket más felhasználók is el tudnak olvasni? És ez volt az a pont, amikor leültem és összeírtam magamnak, hogy milyen lépéseket kellene követnie a programnak.

# 1. Felhasznált technológiák

## 1.1. Választott nyelv, segédeszközök

### 1.1.1.PHP

A PHP egy általános szerveroldali szkriptnyelv dinamikus weblapok készítésére. Az első szkriptnyelvek egyike, amely külső fájl használata helyett HTML oldalba ágyazható.

### 1.1.2.HTML

A HTML (angolul: HyperText Markup Language=hiperszöveges jelölőnyelv) egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált a W3C (World Wide Web Consortium) támogatásával.

### 1.1.3.CSS

A CSS (Cascading Style Sheets, magyarul: lépcsőzetes stíluslapok) a számítástechnikában egy stílusleíró nyelv, amely a HTML vagy XHTML típusú strukturált dokumentumok megjelenését írja le.

### 1.1.4.SASS

A Sass egy előfeldolgozó szkriptnyelv , amelyet Cascading Style Sheets -be (CSS) értelmeznek vagy fordítanak le . A SassScript maga a szkriptnyelv.

### 1.1.5.Bootstrap

Bootstrap egy HTML, CSS, JS reszponzív keretrendszer. Számos webhely használja. Előre elkészített osztályokat tartalmaz.

### 1.1.6.Javascript

A JavaScript programozási nyelv egy objektumorientált, prototípus-alapú szkriptnyelv, amelyet weboldalakon elterjedten használnak.



### 1.1.7.JQuery

A jQuery népszerű JavaScript könyvtár, mely a HTML kód és a kliens oldali JavaScript közötti kapcsolatot hangsúlyozza. Időt és felesleges sorok írását könnyíti meg.

## 1.2. Felhasznált programok

### 1.2.1. Visual Studio Code

A Visual Studio Code (rövidítve: VSCode vagy VS Code) egy ingyenes, nyílt forráskódú kódszerkesztő, melyet a Microsoft fejleszt Windows, Linux és OS X operációs rendszerekhez.

### 1.2.2. MySQL Workbench

A MySQL Workbench egy vizuális adatbázis-tervező eszköz, amely integrálja az SQL fejlesztést, adminisztrációt, adatbázis-tervezést, -létrehozást és -karbantartást egyetlen integrált fejlesztői környezetbe a MySQL adatbázis rendszer számára.

### 1.2.3. MAMP

A MAMP egy ingyenes, nyílt forráskódú és szabadalmaztatott kereskedelmi szoftverekből álló megoldáscsomag, melyet együtt használnak dinamikus webhelyek fejlesztésére és futtatására Apple Macintosh számítógépeken.

### 1.2.4. Adobe XD

Az Adobe XD (más néven Adobe Experience Design) egy vektor-alapú felhasználói élmény-tervező eszköz web alkalmazásokhoz és mobil alkalmazásokhoz, amelyet az Adobe Inc fejlesztett ki és adott ki.

### 1.2.5. DrawSQL

A DrawSQL egy egyszerű adatbázis-diagram szerkesztő fejlesztők számára. Kapcsolatok létrehozásához és azok megjelenítéséhez alkalmas weboldal.

## 2. Felhasználói dokumentáció

A weboldal célja, hogy egy konkrét könyvtár minden könyvét a felhasználók bármikor el tudják érni a róla ismert adatokat és ezen könyveket le tudják foglalni. Így amikor legközelebb mennek könyvtárba csak meg kelljen fogni a könyvet és rohanni a további teendőit elintézni.

### 2.1. Hardware és Szoftver igény

Az oldal eléréséhez szükség lesz internet elérésre, szükség van még egy böngészőre (Google Chrome, Firefox, Microsoft Edge stb.), egy fordító környezetre és egy adatbázis szerverre.

### 2.2. Lehetőségek regisztráció előtt és után

#### 2.2.1. Regisztráció, Belépés

Regisztráció előtt a felhasználó előtt 3 felület látható. A konkrét regisztrációs felület, ahol a felhasználó az adataival regisztrálhat az oldalra. Ha már rendelkezik fiókkal akkor a “Már regisztráltam” felírra kattintva átirányít a weblap a belépő felületre. Ezen a felületen az összes felhasználó be tud lépni a megfelelő adataival, melyet az adatbázis tartalmaz. Amennyiben a felhasználó elfelejtette a jelszavát, korlátoltságok és biztonsági okokból a könyvtár területén tudja csak megváltoztatni a jelszavát. A(z) “Elfelejtettem a jelszavam” felírra kattintva egy olyan felület jelenik meg a felhasználó előtt, ahol a regisztrált e-mail címét megadva küld egy hibajegyet a könyvtáros felületére, ahol majd közösen meg tudják változtatni a felhasználó jelszavát.

### 2.2.2. Navigációs sáv

A navigációs felület minden oldalon megtalálható, kivéve a könyvtáros felületét. Ez a sáv az oldalak felső részén helyezkedik el. Minden, ami ezen a sávon belül megtalálható egy hivatkozásként felel meg. Balról jobbra haladva az első elem, amit láthatunk egy visszanyíl. Ezt megnyomva az előző oldalra tér vissza. A következő elem a “Kezdőlap” felirat. Ezt megnyomva a főoldalra térünk vissza. Mellette a “Hírek” gomb található. A gomb megnyomása után a főoldalra navigálja át a felhasználót, viszont a Hírek szekcióhoz görget. A “Könyvek” hivatkozásra kattintva a könyvek oldalára navigál át a weboldal. A “Profil” felirat a felhasználó adatainak eléréséhez mutat. A felsorolás végén, ha a megfelelő(könyvtáros) fiókba lépünk be (melyhez a jelszó “*konyv1234*”), akkor látható számunkra egy “Kezelőfelület” gomb. Ez a gomb a könyvtáros felületére irányítja át. A sor végén egy kijelentkeztető gomb található, mely a Belépő felületre irányítja át a felhasználót.

### 2.2.3. Főoldal és tartalma

Sikeres Belépés és Regisztráció után a felhasználó előtt a Kezdőlap látható. Ezen a felületen egy üdvözlő szöveg látható, mely tartalmazza a felhasználó keresztnévét (amit megadott a regisztráció során). Az üdvözlő szöveg alatt a könyvtár hírei láthatóak. Ezen híreket a könyvtáros tudja létrehozni. Automatikusan a program a legújabb híreket rakja előre és limitálva jelennek meg. Hat darab hír jelenik meg minden felhasználó kezdőlapján. Mind a Kezdőlap, mind a többi oldal is reszponzív (alkalmazkodik az eszköz méreteihez). Tehát a hírek felület is ugyanúgy olvasható, látható mobil, tablet eszközökön egyaránt, mint asztali számítógépen vagy laptopokon.

### 2.2.4. Könyvek listája

Ez a felület felel azért, hogy a felhasználó a könyvtár összes könyvét megtudja nézni. Alapvetően egy táblázat szerű lista jelenik meg a könyvekről. A bal oldali mező a keresésre szolgál. A jobb oldali két gomb a nézetet változtatja. A nem aktív

gombra kattintva nézetet változtathatunk, amely egy webshop szerű elrendezést és kinézetet nyújt a felhasználónak. Ez bármikor visszaállítható bármelyik megoldásra. A sorok a táblázatban és a kártya szerű mezők mind a könyv dedikált oldalára mutatnak.

#### 2.2.5. Bármely könyv oldala

Bal oldalon a könyv borítóképe látható. Mellette a könyvről az adatbázisban szereplő adatok. Minden adat az összes könyvről valószínűleg nincsen meg egy könyvtárban sem, így a weblap csak a kapott adatokat jeleníti meg. A felsorolt információk alatt megtalálható egy gomb “Lefoglalom” felirattal. Ebben a formában bármely felhasználó lefoglalhatja a kiválasztott könyvet. A Gomb megnyomása után egy beépített funkció megkérdezi, hogy biztos le szeretné-e foglalni a könyvet. A foglalás után egy szöveg jelenik meg. Amennyiben ez a könyv le van már foglalva- akár más akár maga a felhasználó által- ugyanez a szöveg jelenik meg minden felhasználó előtt. Az “itt” gomb megnyomása után a felhasználó meg tudja nézni a már lefoglalt könyveit. A weblap alján egy szöveges mező található, ahol a felhasználó a könyvről tud véleményt mondani és bírálni comment formájában.

#### 2.2.6. Profil felület

A felhasználó előtt rögtön a profil szerkesztés felület jelenik meg. Ezen a felületen az e-mail címén és jelszaván kívül mindent tud szerkeszteni. A módosításokhoz a felhasználónak be kell írnia a jelszavát a módosítások megerősítése érdekében. A lap alján található a felhasználó összes foglalása. A foglalások automatikusan lejárnak hét nap után. Viszont, ha a felhasználó le szeretné mondani a foglalását, akkor erre is van lehetősége.

#### 2.2.7. Könyvtáros felülete

A Kezelőfelületre kattintva a könyvtáros felülete jelenik meg a felhasználó előtt. Ez a felület felel azért, hogy a könyvtáros a neki szánt feladatokat el tudja intézni.

#### 2.2.7.1. Navigációs sáv

Ellentétben a többi oldalon található navigációs sávval ez egy teljesen más menü, amely csak ezen az egy felületen található meg. A hivatkozások megnyomása után a megfelelő ablakok kapcsoló szerű módon jelennek meg. Ha mindegyiket megnyomja a felhasználó akkor mindegyik lenyílik és az összes látható lesz.

#### 2.2.7.2. Hírek

A könyvtár hírei ugyanolyan módon és alakban jelennek meg, mint a kezdőlapban lévő -a felhasználók számára. Viszont ezen a felületen az összes hír felett megjelenik egy ceruza és egy kuka motívum. Ez azt a lehetőséget biztosítja a könyvtáros felhasználónak, hogy egyszerű módon tudja szerkeszteni avagy törölni. A szerkesztés felülete a ceruza megnyomása után jelenik meg. Itt látható a hír egyedi azonosítója, mely nem szerkeszthető. Ez alatt található a címsor, majd a szöveges rész. Az alatta lévő két mező az utolsó módosítást mutatja és a hír készítőjét.

A kuka ikon a hírt rögtön törli.

A hírek alatt található egy kék alapon lévő plusz jel. Ezt a gombot megnyomva új hírt hozhatunk létre. Ez a felület biztosít a felhasználóknak egy címsor mezőt és a szöveges mezőt.

#### 2.2.7.3. Hozzászólások

A hozzászólások fület lenyitva egy hasonló üveges mező jelenik meg, mint a híreknél. Itt megtalálható a könyvtáros számára az összes el nem bírált megjegyzés. A felsősor a felhasználó nevét mutatja. Alatta egy e-mail cím látható. Majd a könyv címét és a hozzá tartozó megjegyzést kötőjellel elválasztva. A hozzászólások fölött megtalálható egy pipa és egy kuka ikon. A kuka ugyanúgy, mint a híreknél, rögtön törli. Ezzel ellentétben a pipa az elfogadást jelenti, és a megnyomása után látható lesz minden felhasználó számára a megjegyzés.

#### 2.2.7.4. Foglalások

Ezt a mezőt megjelenítve az összes jelenlegi foglalást tudja megnézni a könyvtáros. Egy táblázat jelenik meg.

A táblázat mutatja a foglaló e-mail címét. A foglalás kezdetét és a lejárat dátumát. Majd a könyv adatait. Egyedi azonosító, cím, szerző, kiadó.

#### 2.2.7.5. Jelszómódosítás

A jelszó módosításhoz szükséges a felhasználó által küldött hibajegy. Ha ezt elfelejtette megcsinálni, akkor a könyvtárban is van rá lehetősége ezt pótolni. Ez után az új jelszót kell beírni, majd újra. Ha minden megegyezik akkor a “Mentés” gomb lenyomása után automatikusan módosul a jelszava.

## 3. Fejlesztői dokumentáció

### 3.1. Tervezés folyamata

Mint minden nagy programot, úgy az én szakdolgozatomat is meg kellett tervezni. A tervezés folyamata kezdetben jegyzetömbbe felírt adatbázis terv volt. Majd elkezdtem megtervezni a felületet, melyet minden felhasználó lát. A tervezés alatt éreztem a munkámat kifizetődőnek. Ezek voltak azok a pillanatok, amikor úgy éreztem, hogy egy jó terv már fél siker.

#### 3.1.1. Adatbázis tervek (Database)

Az adatbázis terv elkészítéséhez kezdetben tollat és ceruzát használtam. Majd mikor kezdett összeállni a kép, akkor találtam egy weboldalt, amit a tervezés során használtam. Az oldal ezen a domain címen érhető el: “drawsql.app”.

Az adatbázis elkezdése óta már a negyedik verziója készült el a tervnek. A sok variáció meghozta az egyik olyan verziót, ami nem tökéletes, de a program működése szempontjából megfelelő. Ahogy a programot, úgy az adatbázist is lehetne fejleszteni.

#### 3.1.2. Felhasználó felület tervek (User Interface)

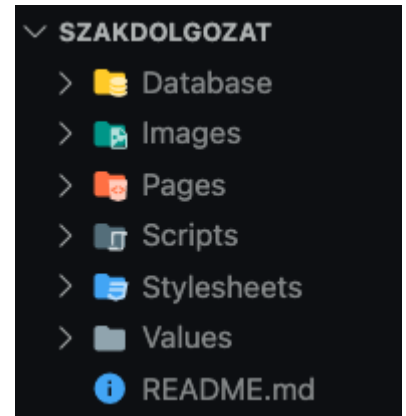
A felhasználói felület elkészítéséhez a(z) Adobe XD nevű programot használtam, melyről az 1.2.4.-es pontban olvashat bővebben. A weblap kinézete így utólag visszatekintve nem lett a legszebb, de ezen későbbiekben lehet alakítani. A tervezés ezen folyamata alatt az áttetsző elemekre építettem fel mindent. Véleményem szerint egy átlátható, tiszta felület sok mindent elárul az alkotóról és ez esetben az oldalról is. Egy egyszerű de átlátható weboldal volt a cél, melyet szerintem sikerült abszolválni.

## 3.2. A szoftver

### 3.2.1. Összegző

#### 3.2.1.1. Struktúra (Fájlok rendezése)

Ahogy fentebb is említettem, számomra igenis fontos az, hogy valami- amit nyújtok vagy látok- átlátható legyen. Ezért a weboldalt is próbáltam a legátláthatóbb módon előállítani. Minden fájlt a neki szánt mappákba csoportosítva helyeztem el. Található egy *Database* nevű mappa, amely az összes tábla adatát és felépítését tartalmazza, melyet csak importálni kell. A következő mappa az *Images* névre



hallgat. Ebben a mappában található meg a háttér a szoftvernek. Alatta a *Pages* mappa szerepel. Ebben a mappában van az összes olyan oldal, amit elérhetnek a felhasználók. Majd a *Scripts* mappa található. Ez a mappa felel azért, hogy az összes .js kiterjesztésű fájl egy helyen megtalálható legyen. Majd a *Stylesheets* mappa található amely tartalmaz egyetlen egy .css fájlt és tizenkettő .scss fájlt. A Sass használata lehetővé tette, hogy átláthatóbb és kényelmesebb fejlesztési környezetben tudjam elkészíteni a tervezett felületet. Az utolsó mappa a listán a *Values* mappa, amely tartalmazza az összes olyan .php fájlt, amelyek szükségesek voltak akár többszöri használatra vagy csak a korlátoltságok miatt külön fájlba kellett megoldani az adott problémákat. A mappák mellett szerepel egy *README.md* fájl, amely egyéb adatokat tartalmaz a munkálatok folyamatáról.

#### 3.2.1.2. Továbbfejleszthetőség

Mint minden alkalmazást, szoftvert tovább lehet fejleszteni, úgy ezt is lehetne akár sokkal felhasználóbaráttá fejleszteni, ahogy bővíteni is lehetne nagyon sok funkcióval. A tervezés folyamata alatt az értesítések jutottak eszembe. Arra gondoltam, hogy amikor lefoglalható egy olyan könyv, ami a kedvencek között van a weboldal küldene egy értesítést a felhasználónak. Ebből következően a



saját lista vagy a kedvelt könyvek listája is egy plusz funkció, ami dobna az egész program felépítésén. Ezek mellett még a kategóriák alapján történő szűrést is meg lehetne valósítani. A könyvtáros felhasználók különböző jogosultságokkal is rendelkezhetnének. Ezt a listát még sorolhatnánk sokáig. Egy nagyon összetett és jó rendszert lehetne még ebből a szoftverből kihozni.

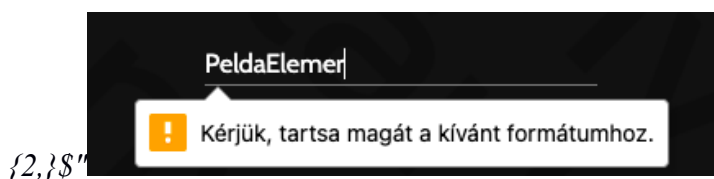
### 3.2.1.3. Biztonsági kérdések

Bármely alkalmazás vagy webes felületnél elvárható az, hogy a felhasználók adatait biztonságosan tárolják. Ahogy az is, hogy a szoftverben ne legyen rés, ahol a rosszakaró felhasználók tönkre tudják tenni a programot vagy akár más felhasználókat vagy fiókjukat. A regisztráció után az összes felhasználónak a jelszava titkosítva kerül az adatbázisba. Ezek a jelszavak MD5-ös titkosítást használnak és salt-ot tartalmaznak. A másik fontos biztonsági rés a belépés nélküli használat volt. Minden oldal látogatása előtt szerepel egy vizsgálat, ami azt felügyeli, hogy be van-e jelentkezve valaki. Amennyiben nem történt meg a bejelentkezés hiába írja be a felhasználó az URL címet, nem fogja tudni elérni, ugyanis rögtön a belépő felületre irányítja át a felhasználót.

Amikor egy átlagos felhasználó el akarja érni a könyvtáros felületét URL cím alapján, erre ugyanúgy nincs lehetősége, mert ezen felületen is szerepel egy vizsgálat, mely a jogosultságokat figyeli.

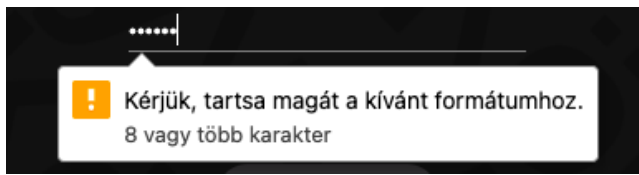
### 3.2.2. Regisztráció / Belépés

A regisztráció során a felhasználóknak a megfelelő mező kitöltésére a beépített *pattern* lehetőséget használtam fel. Az email cím mezőben található egy nem szép, de annál remekebb szöveg, amely azért felel, hogy ez a mező a megfelelő módon legyen kitöltve. *pattern="[a-zA-Z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]*



A jelszó ellenőrzés ennél kicsit egyszerűbb és szebb megoldást használ *pattern* terén.

`pattern="{8,}"`



Ez egy egyszerű megoldás arra, hogy a karakterek száma minimum nyolc karaktert elérje. Emellett még vizsgálni kellett a két jelszó egyezését is. Erre viszont már a php-t kellett segítségül hívni. Amennyiben nem egyezik a két jelszó a felhasználó egy hibaüzenettel találkozik. Egy hasonló hibaüzenet jelenik meg akkor is, ha a felhasználó nem töltött ki egy mezőt.

A Bejelentkezésnél hasonló hibaüzenet várhatja a felhasználót, ha nem töltött ki egy mezőt. Ha egy olyan email címet ad meg a felhasználó, ami nem szerepel az adatbázisban, vagy a jelszó nem megfelelő; akkor egy hasonló hibaüzenet jelenik meg a felhasználónak. A belépés gomb fölött még található egy *“Elfelejtettem a jelszavam”* hivatkozás. Ez a hivatkozás megnyit egy másik oldalt, ahol mint egy hibajegyet ki kell tölteni. Amennyiben nem szerepel az adatbázisban az email cím, akkor egy hibát dob a felhasználónak. Ezt egy egyszerű Select paranccsal lekérdezi az adatbázisból, majd megvizsgálja a program, hogy az input mezőben megadott email cím tartalmazza-e a választ, amit az adatbázistól kapott.

### 3.2.3. Navigációs menü

Alapvetően a navigációs mező egy külső .php fájl, amit minden egyes oldalon (ahol ezt a fajta navigációs menüt használja a program) *Include\_once* paranccsal meghív a weblap. A navigációs menü elkészítésén gondolkoztam a legtöbbet. Először azt terveztem, hogy lesz egy külön belépő felület ahol a könyvtárosoknak be kell jelentkezni, aztán eszembe jutott, hogy lenne egy lustább és okosabb megoldás. Így nem kellett feleslegesen dolgoznom azon, ami egy egyszerű megoldással megoldható. Ez a megoldás azt takarja, hogy a Navigációs menü egy átlagos felhasználónak így néz ki:



Addig egy könyvtáros felhasználónak pedig a következőképpen:

Ezt a vizsgálatot egy egyszerű lekérdezéssel kezdi a szoftver- ahol a felhasználó jogosultságára keres rá-, majd miután kiderült, hogy rendelkezik-e a jogosultsággal, akkor egy javascript fájl hív meg a program, amiben a menüben található szövegre egy “*display: block*” CSS értéket rak.

### 3.2.4.Kezdőlap

Mivel a felhasználó email címét- mely az egyedi kulcs a belépés után- rögtön tárolja a program futásának végéig. Addig bármilyen műveletet (megjegyzést ír, lefoglal egy könyvet) végez a felhasználó, egyértelműen tud hivatkozni a program ezen személyre. Így van ez az üdvözlő szövegnél is. Egy egyszerű lekérdezés lekéri a felhasználó email címe alapján a keresztnévét és kiírja azt. A másik lekérdezés a hírek láthatóságának kellett. Itt egy Limitált lekérdezést hajt végre a szoftver. Egyszerre csak 6 hír jelenik meg. Ezen hírek rendezve vannak dátum szerint. Így csakis az utolsó hat jelenik meg mindenki számára.

### 3.2.5.Könyvek

Két felület közül választhat a felhasználó. Egy lista nézet és egy képes nézet közül. Alaphelyzetben a lista nézetet látja mindenki amelyben minden sor egy linknek felel meg. A linkek végén szerepel a könyv egyedi azonosítója, melyet egy *GET* paranccsal érhet majd el a következő oldal. A keresés mező minden adatra keres, amely a táblázatban található. Ha a könyv azonosítójára keres a felhasználó, akkor az alapján is megtudja találni a keresett könyvet. A keresés mezővel egy vonalban találhatók a nézetet változtató gombok. Ezeken a gombok megnyomásakor egy-egy javascript fájl hív meg a program, amellyel a CSS attribútumokkal ki és bekapcsolni tudjuk az adott elemeket. A *switchToImages.js* fájl a listára rak egy “*display: none*” elemet és a képes nézetre pedig egy “*display: block*” elemet ilyenkor eltüntetni róluk az ellenkező elemet. Ellenkező esetben a *switchToList.js* fájl hívja meg a program, amely pont az ellenkezőjét hajtja végre. A Keresés mind a kettő nézetben működik.

Csak más elemekre keres ezen esetben a “*search.js*” és a “*search2.js*” fájlt.

### 3.2.6. Specifikált Könyv

Amennyiben kiválasztotta a felhasználó a kívánt könyvet, akkor egy előre elkészített statikus oldalt lát meg. Az URL címben tárolt id-t egy *GET* paranccsal kap meg az oldal majd ezzel az id-val készít egy SQL lekérdezést, amely által minden adatot a könyvről betölt a neki szánt helyére.

A felsorolt adatok alatt található a *Lefoglalom* gomb, amely megnyomása után egy beépített *alert()* metódusban rákérdez az alkalmazás, hogy biztosan le szeretné-e foglalni vagy sem. A foglalás után ugyanaz a szöveges üzenet fogadja a felhasználót, mint más felhasználókat- ha ugyan erre a specifikus oldalra kattintanak. Ekkor egy SQL paranccsal bekerül egy új sor az adatbázisba, így más nem tudja lefoglalni a könyvet. Hasonló módon kerül a megjegyzése a felhasználónak egy adatbázis táblába, ha hozzászól az adott könyvhöz.

### 3.2.7. Profil

A Profil felületen az input mezőkben megjelenített adatokért ismételten egy SQL lekérdezés felel. Az elmentett email cím alapján minden mezőt kitölt a program, majd ezeket szabadon tudja a felhasználó szerkeszteni. Viszont csak az után, hogy a felhasználó megadta a jelszavát. Ez egy gyakori megoldás arra, hogy a felhasználók szerkeszteni tudják a profiljukat. Ez egyfajta biztonsági lépcső, ha illetéktelenül lépne be valaki a felhasználó fiókjába.

A másik felület amit a profil felületen lát a felhasználó, az a “*Foglalásaim*”-nak a lekérdezett táblázata. Itt ismételten egy *Inner join* által használt lekérdezés látható. Úgy gondoltam a felhasználóknak egy könyv azonosító nem lett volna egyértelmű, melyik könyv is az, amelyikről látja a foglalás adatait; így 1-2 adat a könyvről bekerült a táblázatba. A táblázat végén látható művelet oszlopban található egy “*Törlés*” gomb. A gomb átirányít egy scripthez hasonló oldalra, ahol egy Update paranccsal a foglalás értékét töröltre állítja így más felhasználók is képesek lennének lecsapni a könyvre. Miután az Update parancs sikeres volt, az oldal

visszairányít a profil felületre - ami egyfajta frissítést is csinál az oldalnak - így biztosan eltűnik az adat az oldalról.

### 3.2.8. Könyvtáros felület

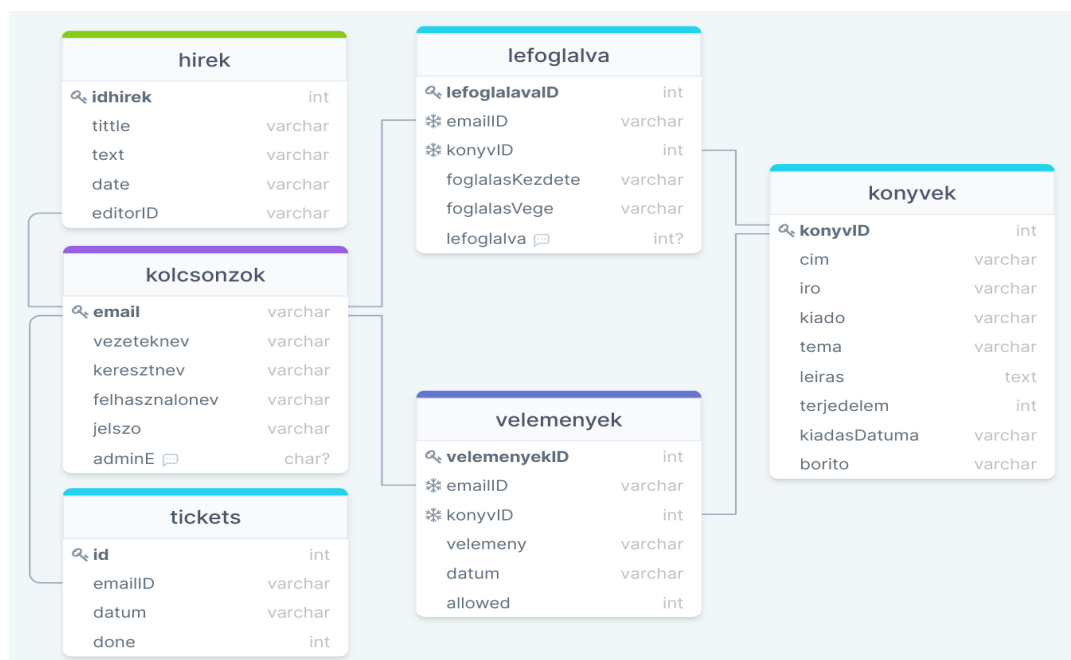
A könyvtáros felületén egy hasonló megoldással jelenítődnek meg az egyes ablakok, mint a könyveknél láthattuk. Egy olyan megoldást gondoltam ki, hogy ne kelljen különböző oldalakat nyitnia a programnak, de kényelmesebb legyen. Arra gondoltam, hogy szebb és kényelmesebb élményt nyújthat egy olyan megoldás, amely mögött a *showA.js* fájl rejlik. Ebben a fájlban a navigációs menü adott gombjaira hivatkozás kép egy *"a"* tag-ben található *function*-t hív meg a program.

```
<li class="p-3"><a href="javascript:showHirek();">Hírek</a></li>
```

A *showHirek* és a hasonló function-ökben egy egyszerű *"classList.toggle('d-none');"* oldja meg a program az eltűnést és megjelenést.

Az ablakokon megjelenített táblázatok és gombok ismételten Select, Insert Into vagy Update SQL parancsokkal vannak ellátva.

## 3.3. Az adatbázis



### 3.3.1.Kolcsonzok(Felhasználók) tábla

**email** - A felhasználók egyedi azonosítója (Primary-Key). varchar típus.

**vezeteknev** - A felhasználók vezetéknévét tárolja. varchar típus.

**keresztnev** - A felhasználók keresztnévét tárolja. varchar típus.

**felhasznalonev** - A felhasználók felhasználónevét tárolja. varchar típus.

**jelszo** - A felhasználók jelszavát tárolja titkosított formában. varchar típus.

**adminE** - A felhasználók jogosultságát tartalmazó mező. Ez a mező felel azért, hogy a program futása közben csak a megfelelő személyek léphessenek be adott felületekre. Int típus.

### 3.3.2.Hirek(Hírek) tábla

**idhirek** - Bármely hír egyedi azonosítója. (Primary-Key) Int típusú.

**tittle** - A hír címe. Int típus. Auto-Increment.

**text** - A hírek szöveges mezője. Amely képes a sortöréseket eltárolni és később ezek alapján megjeleníteni a híreket. Ez is egy varchar típus.

**date** - A MySQL dátumkezelése és a későbbi használat miatt varchar típus a dátum is. A hír elkészítésének pontos dátumát tartalmazza.

**editorID** - A hír készítőjének az egyedi kulcsát, azaz az email címét tárolja ezen mezőben. Ez az a mező, amely össze van kapcsolva a *kolcsonzok* tábla *email* nevű mezőjével.

### 3.3.3.Tickets(Hibajegyek) tábla

**id** - Minden hibajegynek van egy egyedi azonosítója. Ez a mező erre a célra van kitalálva. Int típus. Auto-Increment.

**emailID** - Varchar típus, amely által hivatkozni tud a program a későbbiekben, hogy melyik email cím az, amelyhez tartozó jelszót meg szeretnék változtatni.

**datum** - Varchar típus. A hibajegy létrehozásának pontos dátumát mutatja meg ez a mező.

**done** - egy int típusú mező, mely arra szolgál, hogy a már befejezett műveleteket ne jelenítse meg a programon belül. Ha kész egy folyamat, akkor ezt a mezőt is megváltoztatja.

#### 3.3.4.Konyvek(Könyvek) tábla

**konyvID** - Minden könyvnek van egy egyedi azonosítója. Ez a mező ezt a célt szolgálja. Int típus. Auto-Increment.

**cim** - Varchar típusú mező. A könyvek címét tartalmazzák.

**iro** - Avagy költő. Ez is egy varchar típusú mező. A könyv szerzőjének a nevét tartalmazza. Varchar típus.

**kiado** - A könyv kiadójának a nevét tartalmazza. Varchar típus.

**tema** - A könyv témáját határozza meg ez a mező. Varchar típus.

**leiras** - A könyvnek a témája található meg ebben a mezőben. Mivel ez egy hosszabb szöveg, ezért egy text típusú mező.

**terjedelem** - Int típusú mező. Az oldalak számát tárolja.

**kiadasDatuma** - A könyv kiadásának dátumát tárolja ez a mező. Varchar típus.

**borito** - Ez is egy varchar típus. Viszont ez a mező a könyvről található képet tárolja.

#### 3.3.5.Lefoglalva tábla

**lefoglalvaID** - Minden foglalásnak van egy egyedi azonosítója. Ez a mező erre a célra van kitalálva. Int típus. Auto-Increment.

**emailID** - Varchar típus. Az egyszerű azonosítást szolgálja ez a mező. Idegen kulcs. A *kolcsonzok* tábla *email* nevű mezőjére mutat.

**konyvID** - Idegen kulcs, amely a *konyvek* táblára mutat. Int típus. Auto-Increment.

**foglalasKezdet** - A foglalás megkezdését tárolja varchar típusú mezőben.

**foglalasVege** - A foglalás végét tárolja varchar típusú mezőben.

**lefoglalva** - Egy int típusú mező, amely a könyv foglaltsági állapotát jelzi. Ez a változó felel azért, hogy egy könyvet le tudjon foglalni a felhasználó vagy sem.

### 3.3.6. Velemenyek(Vélemények) tábla

***velemenyekID*** - Int típusú mező. Auto-Increment. Minden véleménynek van egy egyedi azonosítója. Ez a mező ezt a célt szolgálja.

***emailID*** - Idegen kulcs, amely a *kolcsonzok* táblára mutat. Azon belül pedig az *email* mezőre. Később ez a mező alapján látható, hogy ki írta a megjegyzést az adott könyvről. Varchar típus.

***konyvID*** - Idegen kulcs, amely a kiválasztott könyvre mutat. Int típus.

***velemeny*** - Egy Varchar mező mely szöveges hozzászólást képes eltárolni.

***datum*** - A hozzászólás készítésének dátuma. Varchar típus.

***allowed*** - A hozzászólás láthatóságát mutatja. Ez a mező felel azért, hogy a megjegyzések láthatóak legyenek-e vagy sem. Int típusú mező.



## 4. Összegzés

Az elkészített programnak vannak hiányosságai viszont, ahogy említettem korábban is a szakdolgozatban, ezen hiányosságokat bármikor tovább lehetne fejleszteni.

A programozásnak az a szépsége, hogy egy kis ötlet vagy csak egy hiba és már rögtön neki is ugrik az ember, hogy megoldja (persze ha van affinitása hozzá).

Míg írtam a dokumentációt a programról, tesztelnem kellett, hogy tudjam pontosan mit és hogyan csináltam. Ezen idő alatt amilyen hibákat találtam, ezeket mind sikerült orvosolni. Próbáltam a legtöbbet kihozni belőle.

Ez a szakdolgozat nem az utolsó pillanatban lett elkészítve, hanem hónapok sorozata alatt készült el. Ha valaki megnézi a programkódot; látja, hogy - a kezdeti munkák, mint a regisztráció és a belépés felülete - nagyon alapszintű php megoldásokat használ, amiket már csak azért is úgy hagytam. Reprezentálja a fejlődést, amin átmentem a készítés alatt. Véleményem szerint egy nagyon jó témát választottam, ugyanis a készítés alatt olyan megoldásokat tanultam meg és néztem utána, amiket valószínűleg fogok is használni a jövőben.

Ez a szakdolgozat nagyon sokat jelent számomra. Egy olyan élmény és érzés volt bennem a készítés alatt, amely még jobban meghozta a kedvemet, hogy ebben az irányban szeretnék elhelyezkedni a későbbiekben.

## 5. Irodalomjegyzék

<https://hu.wikipedia.org/wiki/PHP> - 2022.03.12.

<https://hu.wikipedia.org/wiki/HTML> - 2022.03.12.

[https://hu.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](https://hu.wikipedia.org/wiki/Cascading_Style_Sheets) - 2022.03.12.

[https://en.wikipedia.org/wiki/Sass\\_\(stylesheet\\_language\)](https://en.wikipedia.org/wiki/Sass_(stylesheet_language)) - 2022.03.12.

[https://en.wikipedia.org/wiki/Bootstrap\\_\(front-end\\_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)) - 2022.03.12.

<https://hu.wikipedia.org/wiki/JavaScript> - 2022.03.12.

<https://hu.wikipedia.org/wiki/JQuery> - 2022.03.12.

[https://hu.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://hu.wikipedia.org/wiki/Visual_Studio_Code) - 2022.03.12.

[https://en.wikipedia.org/wiki/MySQL\\_Workbench](https://en.wikipedia.org/wiki/MySQL_Workbench) - 2022.03.12.

<https://en.wikipedia.org/wiki/MAMP> - 2022.03.12.

[https://en.wikipedia.org/wiki/Adobe\\_XD](https://en.wikipedia.org/wiki/Adobe_XD) - 2022.03.12.

## 6. Mellékletek

A weboldalon felhasznált képek, melyeket a könyveknél felhasználtam ezen az oldalon találhatóak: <https://www.regikonyvek.hu/>