Budapesti Műszaki Szakképzési Centrum

Verebélÿ László Technikum

Szakképzés megnevezése: Szoftverfejlesztő (54 213 05)

Webáruház könyvek értékesítésére

Készítette: Kiss Bence

Konzulens tanár: Horváth Attila

Készítés dátuma: 2023. 04. 10.

Tartalomjegyzék

[1. Bevezetés 3](#_Toc131964968)

[1. Követelményspecifikáció 3](#_Toc131964969)

[2. Fejlesztői dokumentáció 5](#_Toc131964970)

[2. Alkalmazott fejlesztői eszközök 5](#_Toc131964971)

[3. Alkalmazott technológiák 5](#_Toc131964972)

[4. Futtatási környezet 5](#_Toc131964973)

[5. Fejlesztői ütemterv 5](#_Toc131964974)

[6. Fejlesztés menete 6](#_Toc131964975)

[1. Piackutatás 6](#_Toc131964976)

[2. Adatbázis megtervezése 6](#_Toc131964977)

[3. Főbb algoritmusok 9](#_Toc131964978)

[3. Felhasználói dokumentáció 18](#_Toc131964979)

[1. A program működésének leírása, felhasználói kézikönyv 18](#_Toc131964980)

[1. Alapvető funkciók 18](#_Toc131964981)

[4. Tesztelés 19](#_Toc131964982)

[5. Továbbfejlesztési lehetőségek 20](#_Toc131964983)

[6. Ábrajegyzék 21](#_Toc131964984)

# Bevezetés

Szakdolgozatom témájának kiválasztásánál fontosnak tartottam, hogy egy hozzám személyes kötődéssel bíró témát válasszak. A választásom egy kedves tevékenységemre, az olvasásra esett.

A digitális korszakban egyre többen keresik a gyors szórakozás lehetőségét, amit online rövid videók formájában találnak meg. Az amerikai felnőtt lakosság picivel több, mint fele nem olvasott könyvet 2022-ben, és a fiatalabb generációk olvasási kedve egy csökkenő tendenciát mutat[[1]](#footnote-1). Sajnos én is így vagyok ezzel, holott régebben nagyon szerettem olvasni. A könyveken gondolkodva egy könyvek értékesítésére szakosodott webáruház jutott eszembe, így ezt választottam szakdolgozatom témájának.

Két okból döntöttem a webes implementáció mellett. Egyrészt azért, mert így bármely eszközön elérhető, amely támogatja az internetelérést és rendelkezik böngészőprogrammal, ami egy webáruház szempontjából kulcsfontosságú. A másik személyesebb: az utolsó évben kezdtünk el PHP-t tanulni, és szerettem volna magamnak egy megfelelő kihívást jelentő feladatot.

## Követelményspecifikáció

Célom egy sablon megtervezése volt alapvető funkcionalitással, amit akár egy a piacon lévő könyvesbolt webes felületeként integrálni lehessen minimális módosítással. A felület megtervezésekor elsődleges szempontként az intuitivitást és a felhasználói élményt vettem figyelembe.

A weboldalt a terveim szerint négyféleképpen lehet elérni:

* Vendégként szabadon lehet böngészni a könyvek között, berakni őket a kosárba és rendelni a megadott szállítási címre.
* Regisztrált felhasználóként ugyanúgy lehet könyveket rendelni, továbbá lehet őket értékelni, lehet kommenteket írni a könyv oldalára, el lehet menteni a szállítási címet vagy címeket a rendelés felgyorsításához és a megadott adatok alapján ajánl könyveket a weboldal.
* Moderátorként lehetőség adódik a kommentszekció ellenőrzésére, valamint a nem megfelelő tartalmú kommentek törlésére.
* Adminként fel lehet venni új könyveket az adatbázisba, valamint módosítani és törölni lehet azokat. Az admin látja továbbá a valós idejű eladási statisztikákat.

# Fejlesztői dokumentáció

## Alkalmazott fejlesztői eszközök

* Kódszerkesztő: Visual Studio Code
* Fejlesztői környezet: LAMPP

## Alkalmazott technológiák

* Programozási nyelvek: PHP, JavaScript
* Webes technológiák: HTML, CSS és a népszerű BootStrap 5.3 könyvtár
* Adatbázis motor: MariaDB (MySQL nyelvvel)

## Futtatási környezet

A weboldal bármely modern böngészőben elfut. A weblap reszponzív, így mobil- és táblagépekről is elérhető és kényelmesen használható.

## Fejlesztői ütemterv

|  |  |
| --- | --- |
| **Határidő** | **Feladat** |
| 2022 október vége | Adatbázisterv elkészítése |
| 2022 november vége | Weblapterv elkészítése |
| 2022 december első felének vége | Weblapterv megvalósítása statikusan |
| 2022 december vége | Könyvajánló algoritmus kidolgozása és leprogramozása |
| 2023 január első felének vége | Az adatbázis feltöltése mintaadatokkal |
| 2023 január vége | A dinamikus weboldal prototípusának elkészítése |
| 2023 február vége | A felhasználókezelés és a felhasználók funkcióinak elkészítése |
| 2023 március vége | Végső simítások, előkészítés bemutatásra |

## Fejlesztés menete

### Piackutatás

Mielőtt nekiálltam a munkának, keresgéltem az interneten hasonló weboldalak után. Mind magyar, mind nemzetközi példákat találtam, ami megerősített benne, hogy van ilyen oldalakra igény. Ezen túl ötleteket adtak a design és a funkcionalitás területén is.

### Adatbázis megtervezése

A weboldal hátterében egy MySQL adatbázis fut MariaDB motorral.

Az adatbázis 20 táblából, számos nézetből, tárolt eljárásból, valamint függvényből áll. Igyekeztem a harmadik normálformának megfelelni, a táblákat logikusan és következetesen felvenni, valamint a redundanciát (vagyis a felesleges adatismétlést) elkerülni.

Mint ahogyan az akár egy hozzá nem értő számára is látható a következő oldali ábrán, a legnagyobb hangsúlyt a könyvek és a felhasználók kapták. A továbbiakban leírom a táblák jelentőségét, funkcióik szerint csoportosítva.

A könyvek táblában olyan atomi adatokat tárolok, mint a könyv címe, oldalainak száma, kiadásának dátuma, ára, készlete és leírása. Egyes adatok, mint a könyv súlya vagy akciós ára opcionálisak, nem kötelező őket megadni (ha például nem ismert a könyv súlya, vagy nem akciós). A tábla elsődleges kulcsa a minden könyvet egyértelműen azonosító 10 vagy 13 karakter hosszú ISBN.

A könyvek táblához az ábrán bal oldalt csatlakozó 4 tábla (fentről lefelé: kiadók, nyelvek, kötések, sorozatok) olyan, a nevüknek megfelelő adatot tárolnak, ami több könyvnél is előfordulhat. Ezeket a szöveges adatokat csak egyszer tárolom, és az azonosítójuk alapján hivatkozok rájuk. Mivel nem minden könyv egy könyvsorozat része, az a mező is kihagyható a könyvek táblában.

A könyvek táblától az ábra szerint jobbra lévő 4 tábla a műfajok és az írók tárolására szolgál. Az alapvető logika az, hogy egy könyv több műfajhoz is tartozhat, valamint többen is írhatnak egy-egy könyvet (erre nagyon jó példák a tankönyvek.) A műfajok és írók neveit az előző mezőkhöz hasonlóan tárolom, viszont nem közvetlenül kapcsolom össze a könyvek táblával, hanem egy segédtáblát veszek igénybe. Ezekben a táblákban minden adott könyvhöz egyesével több műfajt és írót is hozzárendelhetek az azonosítójuk alapján.

Lefelé haladva további 4 táblán keresztül eljutunk a felhasználók táblához. Ezek a táblák a kommentek, az értékelések, valamint a rendelések és azok részletezéséről szólnak és összekötik a felhasználókat a könyvekkel. A kommentek és értékelések lehetőséget adnak felhasználóknak tartalom gyártására és az egymással történő kommunikációra. A rendeléseket részletező táblában helyet kap a könyv azonosítója, a vásárolt mennyiség, és az ára. Ez közvetlen összeköttetésben áll a rendeléseket összesítő és lezáró táblával, amiben az egy adott rendelés összesített ára és a rendelés dátuma és időpontja is szerepel. Az időpontot nem kézzel adom meg, hanem az adatbázismotor automatikusan beilleszti a sor beszúrásának idejét. Ezek az adatok alapján egyszerűen elő lehet állítani egy bizonylatot vagy számlát.

A felhasználók táblában egy adott felhasználó személyes adatait tárolom. Ezeket természetesen nem kötelező megadnia. Ilyen adatok a vezeték-, illetve keresztneve, a neme, a születési ideje és a telefonszáma. A nem és a születési idő a korcsoport meghatározására szolgál, ami alapján célzott ajánlásokat tesz nekik az alkalmazás. Amennyiben nem adják meg, ebből kimaradnak.

Több fajta felhasználó van. Jelenleg a felhasználó, moderátor és adminisztrátor szintek állnak rendelkezésre. Ezek a szintek tárolására szolgál a felhasználói típusok tábla.

Az adatbiztonság megvalósítása érdekében a felhasználók személyes adatait és a bejelentkezési adatait külön táblákban tárolom. Az utóbbira szolgál a bejelentkezés tábla, ami tartalmazza a felhasználónevet, a jelszót, az e-mail címet, és egy úgynevezett „sót.” A „sózás” a jelszó védelmére szolgál. Lényege, hogy egy előre generált karaktersorozatot ad hozzá a fiókhoz, amelyet a jelszó hashelése előtt mindig hozzáad. Alapvetően egy adott karaktersorozatot titkosítva mindig ugyanazt az eredményt kapjuk. Ennek eredményeként a gyakrabban használt jelszavak mindig ugyanazzal a hash kóddal szerepelnek az adatbázisban. Viszont, ha egy véletlen karaktersorozatot is hozzáadunk, az ugyanolyan jelszóval rendelkező felhasználóknak is máshogy szerepel a jelszava, így nehezebb őket feltörni.

A felhasználóknak lehetőségük van tárolni a szállítási címjeiket, valamint a vendégeknek is muszáj megadniuk, legkésőbb a rendelés pillanatában. Ezeknek a címeknek van kötelezően megadandó és opcionális része is. Opcionálisan lehet cégnek rendelni, vagy megjegyzést írni a cím pontosítására. Kötelezően megadandó a megye, a város, a közterület és az irányítószám. A megyék a redundancia elkerülése érdekében egy külön táblában kerülnek tárolásra, ahonnan azonosító alapján hivatkozhatóak.

A képen diagram látható

Automatikusan generált leírás

. ábra: EER diagram a `bookstore` adatbázisról

### Főbb algoritmusok

A weboldal két fő programozási nyelve a PHP és a JavaScript, melyeket vegyítve használtam. Ez gyakran AJAX lekérésekkel történt a JavaScript fájlból a PHP fájlok felé. Ez a módszer nagyon hasznos olyan helyzeteknél, amikor nem szerettem volna, hogy a weboldal újratöltsön az adatok frissítéséhez (például, ha lapozunk a kommentszekcióban). A továbbiakban kiemelek eljárásokat és függvényeket a programkódomból, amelyeket jelentősebbnek tartok a teljesség igénye nélkül.

Általános függvény egy tábla egy megadott oszlopának lekérdezésére: Ezt leginkább az admin felület könyvfeltöltéssel foglalkozó formjában használtam, céljuk a <datalist> HTML tagek feltöltése, melyek felgyorsítják a kitöltési munkát. Amikor az admin fókuszba hoz egy <datalist> taggel rendelkező beviteli mezőt, a böngésző felkínálja a korábban már használt értékeket.

A képen szöveg látható

Automatikusan generált leírás

. ábra: A <datalist> tag segítségével a korábban előfordult kötéstípusokat felajánlja a böngésző

function GetResultSetOfSingleColumn($column, $table)

{

    $con = GetConnection();

    $sql = "SELECT DISTINCT " . $column . " FROM " . $table . ";";

    $rs = mysqli\_query($con, $sql);

    mysqli\_close($con);

    return $rs;

}

Könyv beillesztése az adatbázisba: Ezt az eljárást mindösszesen egyszer hívtam meg a program során, de terjedelmét illetően jelentősnek számít, emellett a könyvek adatait frissítő eljárás is ezen alapszik. A PHP beépülő mysqli\_real\_escape\_string() függvényét használtam a speciális karakterek elmentésére. A str\_replace() függvény segítségével cserélem le az admin által beszúrt sortöréseket <br> tagekre, amelyeket a böngésző is értelmezni tud. A műfajokat és az írók neveit @ karakterrel elválasztva tároltam. Az eljárás végén egy prepared statement segítségével meghívtam az InsertBook() MySQL eljárásomat, ami minden adatot beilleszt a megfelelő táblába.

function InsertBook($bookdata)

{

    $con = GetConnection();

    $isbn = mysqli\_real\_escape\_string($con, $bookdata["isbn"]);

    $pages = mysqli\_real\_escape\_string($con, $bookdata["pages"]);

    $publisher = (empty($bookdata["publisher"]) ? null :

    mysqli\_real\_escape\_string($con, $bookdata["publisher"]));

    $weight = (empty($bookdata["weight"]) ? null :

    mysqli\_real\_escape\_string($con, $bookdata["weight"]));

    $title = mysqli\_real\_escape\_string($con, $bookdata["title"]);

    $series = (empty($bookdata["series"]) ? null :

    mysqli\_real\_escape\_string($con, $bookdata["series"]));

    $cover = mysqli\_real\_escape\_string($con, $bookdata["covertype"]);

    $datePublished = mysqli\_real\_escape\_string($con,

    $bookdata["date\_published"]);

    $price = mysqli\_real\_escape\_string($con, $bookdata["price"]);

    $discountedPrice = (empty($bookdata["discounted\_price"]) ? null :

    mysqli\_real\_escape\_string($con, $bookdata["discounted\_price"]));

    $language = mysqli\_real\_escape\_string($con, $bookdata["language"]);

    $stock = mysqli\_real\_escape\_string($con, $bookdata["stock"]);

    $description = mysqli\_real\_escape\_string($con,

    $bookdata["description"]);

    $description = str\_replace("\\r\\n", "<br>", $description);

    $genres = "";

    for ($i = 0; $i < count($bookdata["genres"]); $i++) {

        $genres = $genres . $bookdata["genres"][$i];

        if ($i < count($bookdata["genres"]) - 1)

            $genres = $genres . '@';

    }

Könyvadatok beolvasása: Ez egy gyakran meghívott függvény az alkalmazásban. A feladata, hogy a könyv azonosítója, vagyis ISBN-e alapján kiolvassa az adatbázisból az adatait. Ez alapvetően nem volna egy nehéz feladat, viszont egy könyvhöz több író és műfaj is tartozhat, így azokat egy-egy tömbben tárolom a $bookdata változóban és külön függvénnyel olvasom ki az adatbázisból.

    $writers = "";

    for ($i = 0; $i < count($bookdata["writers"]); $i++) {

        $writers = $writers . $bookdata["writers"][$i];

        if ($i < count($bookdata["writers"]) - 1)

            $writers = $writers . '@';

    }

    $genres = mysqli\_real\_escape\_string($con, $genres);

    $writers = mysqli\_real\_escape\_string($con, $writers);

    $sql = "CALL InsertBook(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?);";

    $stmt = mysqli\_prepare($con, $sql);

    mysqli\_stmt\_bind\_param($stmt, "sisissssiisisss", $isbn, $pages,

    $publisher, $weight, $title, $series, $cover, $datePublished, $price,

    $discountedPrice, $language, $stock, $description, $genres, $writers);

    mysqli\_execute($stmt);

    mysqli\_stmt\_close($stmt);

    mysqli\_close($con);

}

function GetBookByISBN($isbn)

{

    $con = GetConnection();

    $sql = 'CALL GetBookByISBN("' . $isbn . '");';

    $rs = mysqli\_query($con, $sql);

    $assoc = mysqli\_fetch\_assoc($rs);

    $assoc["genres"] = GetGenresByISBN($isbn);

    $assoc["writers"] = GetWritersByISBN($isbn);

    mysqli\_close($con);

    return $assoc;

}

A regisztációhoz és belépéhez tartozó kód: Mint azt az adatbázis részletezésénél már tárgyaltam, egy egyedi sót generálok a felhasználók számára a belépés biztonságának biztosításához. Mind a regisztráció, mind a belépés AJAX lekérések formájában történik. A jelszavak titkosítása az SHA256 algoritmussal történik. Az adatok ellenőrzését JavaScript-tel valósítottam meg.

Rendelés mentése: Ez az egyik legösszetettebb eljárás az alkalmazásban. Első körben kiszámolom a fizetendő összeget, amihez a kosár árából kivonom a felhasználó által felhasználni kívánt pontokat. Ez után a rendelés összesített adatait beírom az adatbázisba, és lekérem az éppen beillesztett rendelés azonosítóját. A következő feladat populálni a rendelés részletesebb adatait tartalmazó táblát, amellyel párhuzamosan csökkentem a könyvekből rendelkezésre álló példányok számát. Utolsó lépésben pedig, ha belépett felhasználó rendel, levonom a fiókjáról a felhasznált pontokat, majd hozzáadom az aktuális vásárlással szerzetteket. Az átláthatóság növelése érdekében további két eljárást írtam, melyeket a fő eljárás törzséből hívok meg.

function Login($username, $password)

{

    $con = GetConnection();

    $username = mysqli\_real\_escape\_string($con, $username);

    $password = mysqli\_real\_escape\_string($con, $password);

    $sql = "SELECT user\_id FROM login WHERE username = ? AND password =

    ?;";

    $stmt = mysqli\_prepare($con, $sql);

    mysqli\_stmt\_bind\_param($stmt, "ss", $username, $password);

    mysqli\_execute($stmt);

    mysqli\_stmt\_bind\_result($stmt, $user\_id);

    mysqli\_stmt\_fetch($stmt);

    mysqli\_stmt\_close($stmt);

    mysqli\_close($con);

    return $user\_id;

}

function SignUp($username, $password, $email, $salt)

{

    $con = GetConnection();

    $username = mysqli\_real\_escape\_string($con, $username);

    $password = mysqli\_real\_escape\_string($con, $password);

    $email = mysqli\_real\_escape\_string($con, $email);

    $salt = mysqli\_real\_escape\_string($con, $salt);

    $sql = "CALL SignUp(?, ?, ?, ?);";

    $stmt = mysqli\_prepare($con, $sql);

    mysqli\_stmt\_bind\_param($stmt, "ssss", $username, $password, $email,

    $salt);

    mysqli\_execute($stmt);

    mysqli\_stmt\_close($stmt);

    mysqli\_close($con);

}

function CompleteOrder($books, $address\_id, $user\_id, $used\_points) {

    if (is\_null($user\_id)) $used\_points = 0;

    $price\_sum = GetPriceSum($books) - $used\_points;

    $con = GetConnection();

    $sql = "SELECT CompleteOrder(?, ?, ?);";

    $stmt = mysqli\_prepare($con, $sql);

    mysqli\_stmt\_bind\_param($stmt, "iii", $user\_id, $address\_id,

    $price\_sum);

    mysqli\_execute($stmt);

    $rs = mysqli\_stmt\_get\_result($stmt);

    mysqli\_stmt\_close($stmt);

    $order\_id = 0;

    while ($row = mysqli\_fetch\_row($rs)) {

        $order\_id = $row[0];

    }

    $isbns = array\_keys($books);

    $item\_count = count($isbns);

    for ($i = 0; $i < $item\_count; $i++) {

        $bookdata = GetBookByISBN($isbns[$i]);

        $price = 0;

        if (is\_null($bookdata["discounted\_price"])) {

            $price = $bookdata["price"];

        } else {

            $price = $bookdata["discounted\_price"];

        }

        AddOrderDetail($con, $order\_id, $isbns[$i], $books[$isbns[$i]],

        $price);

        $sql = "UPDATE books SET stock = stock - ? WHERE isbn = ?;";

        $stmt = mysqli\_prepare($con, $sql);

        mysqli\_stmt\_bind\_param($stmt, "is", $books[$isbns[$i]],

        $isbns[$i]);

        mysqli\_execute($stmt);

        mysqli\_stmt\_close($stmt);

    }

    if (!is\_null($user\_id)) {

        $sql = "UPDATE users SET points = points - ? WHERE user\_id = ?;";

        $stmt = mysqli\_prepare($con, $sql);

        mysqli\_stmt\_bind\_param($stmt, "ii", $used\_points, $user\_id);

        mysqli\_execute($stmt);

        mysqli\_stmt\_close($stmt);

Könyvek keresése: Ez az első függvény, amiben megvalósul az oldalakra tördelés. Átadom a függvénynek, hogy hányadik oldalra keresek, valamint, hogy oldalanként hány találatot szeretnék kapni. Ezek alapján limitálom a lekérdezés által visszatérített sorokat. Ezt a módszert később használtam a kommentek, a különböző könyveket kilistázó aloldalak, valamint az írók és kiadók oldalán megjelenített könyvek oldalakra osztásánál is. Úgy működik, hogy a MySQL LIMIT kulcsszavával megadom az adatbázisnak mettől és mennyi könyv azonosítojával térjen vissza.

function SearchBooks($query, $page, $books\_per\_page) {

    $offset = ($page - 1) \* $books\_per\_page;

    $con = GetConnection();

    $sql = "SELECT DISTINCT isbn FROM books WHERE title LIKE \"%$query%\"

    LIMIT $offset, $books\_per\_page;";

    $rs = mysqli\_query($con, $sql);

    mysqli\_close($con);

    return GetArrayFromResultSet($rs);

}

        $points\_to\_add = floor(($price\_sum - $used\_points) / 10);

        $sql = "UPDATE users SET points = points + ? WHERE user\_id = ?;";

        $stmt = mysqli\_prepare($con, $sql);

        mysqli\_stmt\_bind\_param($stmt, "ii", $points\_to\_add, $user\_id);

        mysqli\_execute($stmt);

        mysqli\_stmt\_close($stmt);

    }

    mysqli\_close($con);

}

Könyvajánlatok lekérése: Ez a függvény volt az egyik célja a szakdolgozatomnak. Úgy működik, hogy egy változóba felveszem az összes könyv ISBN-jét az adatbázisból kulcsokként, 0 értéket párosítva hozzájuk, majd egy másik változóba azoknak a könyveknek az azonosítóit, melyeket a felhasználó elolvasott ugyanazzal a módszerrel. Ezután kivonom egymásból a két halmazt, így egy asszociatív tömböt kapok az olvasatlan könyvek ISBN-jeivel, melyekhez mind a 0 értéket rendeltem. Ez után elmentem a következő adatokat:

* a felhasználó által kedvelt műfajokat,
* a műfajokat, melyekből könyvet vásárolt,
* az írókat, akik könyveiből vásárolt,
* a könyvsorozatokat, amelyekből legalább egy kötetet vásárolt,
* valamint az életkorban és nemben hasonló vagy ugyanolyan egyéb felhasználók által gyakran vásárolt könyveket.

Ezután végigmegyek az asszociatív tömbön. Amennyiben a könyv nem áll rendelkezésre, vagy még nem jött ki, átugrom, különben az értékeket a következőképpen növelem:

* ha a könyv bármely műfaja benne van a felhasználó által kedvelt műfajok listájában, műfajonként 2-vel megnövelem az értéket,
* ha a könyv bármely műfaja vagy írója benne van a felhasználó által megvásárolt műfajú vagy író által írt könyvek listájában, műfajonként és írónként 1-gyel megnövelem az értéket,
* ha a könyv egy könyvsorozat része, és a felhasználó abból olvasott már könyvet, akkor 3-mal növelem az értéket,
* ha pedig a könyv a felhasználó neme és korosztálya szerint a legnépszerűbbek között van, kategóriánként 2-vel növelem az értékét.

Utolsó lépésben az array\_filter() függvénnyel kiszűröm a 0 értékű tömbelemeket, és az arsort() metódussal az értékek alapján csökkenő sorrendbe rendezem az asszociatív tömböt. Ezt a rendezett tömböt térítem vissza.

function GetRecommendations($username, $user\_id) {

    $all\_books = array\_fill\_keys(GetISBNs(), 0);

    $read\_books = array\_fill\_keys(GetReadISBNs(

    $\_SESSION["user"]["username"]), 0);

    $books = array\_diff\_key($all\_books, $read\_books);

    $preferences = GetGenrePreferencesByUsername($username);

    $genres\_purchased = GetGenresPurchasedByUsername($username);

    $authors\_purchased\_from = GetAuthorsPurchasedFromByUsername($username);

    $serieses\_purchased\_from =

    GetSeriesesPurchasedFromByUsername($username);

    $most\_purchased\_books\_by\_age = GetMostPurchasedBooksByAge($user\_id);

    $most\_purchased\_books\_by\_gender =

    GetMostPurchasedBooksByGender($user\_id);

    foreach ($books as $isbn => $value) {

        $bookdata = GetBookByISBN($isbn);

        $publication\_date = strtotime($bookdata["date\_published"]);

        $current\_time = time();

        $is\_book\_available = $bookdata["stock"] > 0 && $publication\_date <=

        $current\_time;

        if (!$is\_book\_available) {

            continue;

        }

        for ($i = 0; $i < count($bookdata["genres"]); $i++) {

            if (in\_array($bookdata["genres"][$i][1], $preferences)) {

                $books[$isbn] += 2;

            }

        }

        for ($i = 0; $i < count($bookdata["genres"]); $i++) {

            if (in\_array($bookdata["genres"][$i][1], $genres\_purchased)) {

                $books[$isbn] += 1;

            }

        }

        for ($i = 0; $i < count($bookdata["writers"]); $i++) {

            if (in\_array($bookdata["writers"][$i][1],

            $authors\_purchased\_from)) {

                $books[$isbn] += 1;

            }

        }

        if (!is\_null($bookdata["series"]) && in\_array($bookdata["series"],

        $serieses\_purchased\_from)) {

            $books[$isbn] += 3;

        }

        if (in\_array($isbn, $most\_purchased\_books\_by\_age)) {

            $books[$isbn] += 2;

        }

        if (in\_array($isbn, $most\_purchased\_books\_by\_gender)) {

            $books[$isbn] += 2;

        }

    }

    $books = array\_filter($books);

    arsort($books);

    return array\_keys($books);

}

Író és műfaj mezők hozzáadása és törlése: A könyv hozzáadása, módosítása, és törlése formokon az író és műfaj mezők alatt található egy „+” és egy „-” gomb. Ezek a gombok ebben a sorrendben hozzá tudnak adni, illetve el tudnak venni egy mezőt a formhoz az adott kategóriában. A feladat megoldásához kihasználom a JavaScript DOM manipuláló képességeit.

function addField(fieldName) {

    let row = document.getElementById(fieldName + "-fields");

    let col = document.createElement("div");

    col.classList.add("col-12", "mb-3");

    let fieldCount = document.getElementsByClassName(fieldName +

    "-field").length;

    let nextFieldId = fieldCount + 1;

    let field = document.createElement("input");

    field.type = "text";

    field.classList.add("form-control", fieldName + "-field");

    field.id = fieldName + "-" + nextFieldId;

    field.name = fieldName + "-" + nextFieldId;

    field.setAttribute("list", fieldName + "s");

    col.appendChild(field);

    row.appendChild(col);

}

function removeField(fieldName) {

    let fieldIdToRemove = document.getElementsByClassName(fieldName +

    "-field").length;

    if (fieldIdToRemove > 1) {

        let elementToRemove = document.getElementById(fieldName + "-" +

    fieldIdToRemove).parentElement;

        let parentElement = elementToRemove.parentElement;

        parentElement.removeChild(elementToRemove);

    } else {

        alert("Legalább egy mezőnek maradnia kell!");

    }

}

# Felhasználói dokumentáció

## A program működésének leírása, felhasználói kézikönyv

A felhasználói kézikönyv a felhasználók képességeinek sorrendje szerint épül fel. A szakaszai megfelelnek a felhasználótípusoknak a weboldalon. Ez alapján az első szakasz a vendégek számára nyújt áttekintést arról, hogy mit tudnak csinálni az oldalon, míg a második a felhasználóknak, a harmadik a moderátoroknak, a negyedik pedig az adminisztrátornak. A szintek között felfelé haladva átmenet van, mint ahogyan azt korábban leírtam. Ennek megfelelően a vendégek számára írt szakasz vonatkozik a felhasználókra is, a felhasználóknak írt szekció vonatkozik a moderátorokra is, és a moderátoroknak írt szekció vonatkozik az adminisztrátorra.

### Vendég funkciók

Bármely vendég vagy regisztrált felhasználó az oldalra térve egy a számára megfelelő aloldalon találja magát. A vendégek számára ez a bestselleres aloldal, a felhasználók és moderátorok számára az ajánlásaink aloldal, az adminisztrátor számára pedig a statisztikák aloldal.

A képen Webhely látható

Automatikusan generált leírás

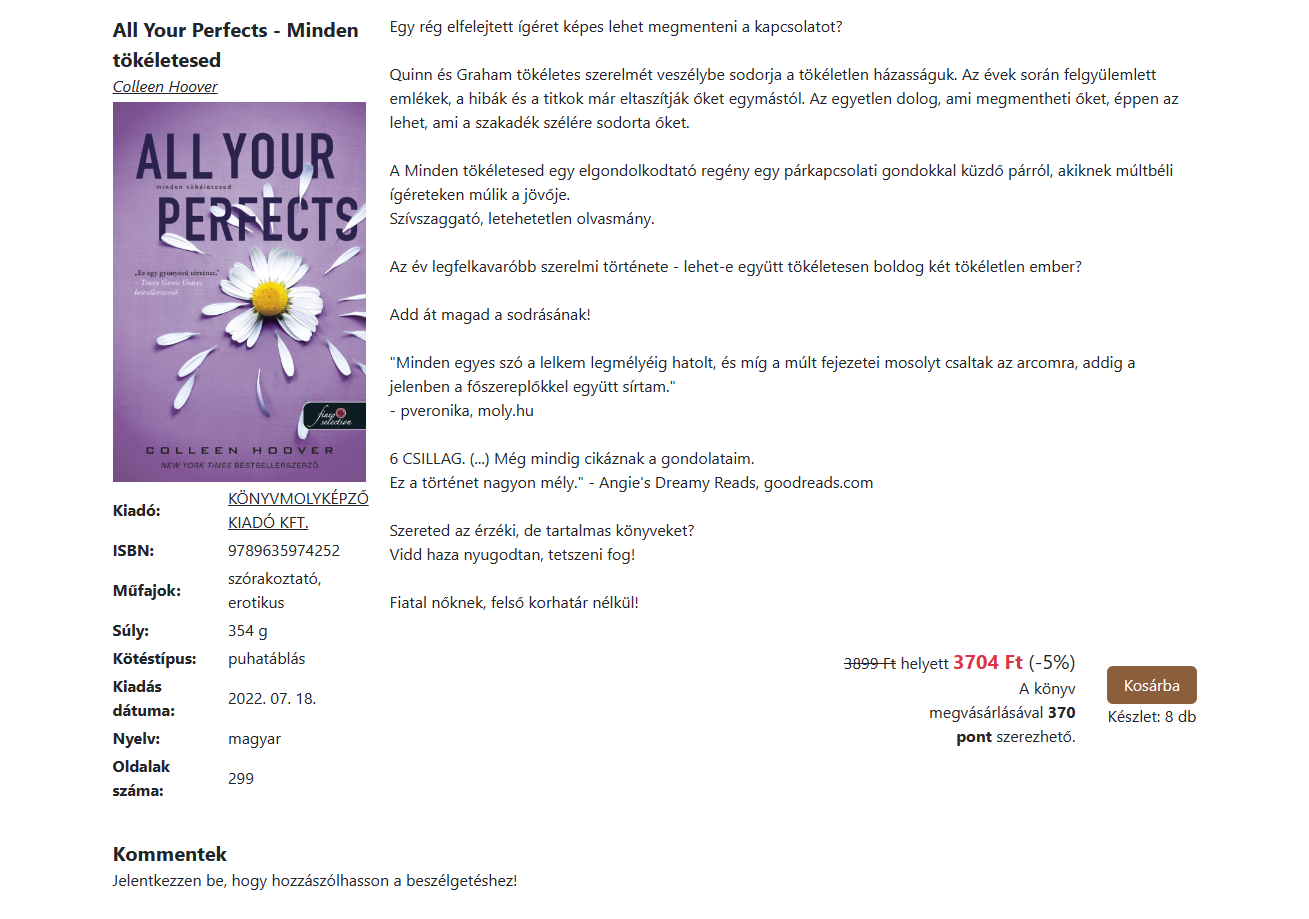
. ábra: A vendégeket a bestsellerek aloldalra írányítja az oldal

A menüben a „Könyváruház” gomb a főoldalra megy (ami ebben az esetben a bestsellerek oldal, ahogyan az a 3. ábrán is látható). A „Bestsellerek” menüpont alatt a legkeresettebb könyvek találhatóak, az „Újdonságok” menüpont alatt az elmúlt egy hónapban kiadott könyvek találhatóak, a „Hamarosan megjelenik” menüpont alatt pedig a még ki nem jött könyvek találhatóak, a kiadás dátuma szerint növekvő sorrendben. A „Véletlen könyv” menüpontra kattintva az oldal egy véletlenszerűen kiválasztott könyv adatlapjára viszi a látogatót.

A menüsorban a keresőmezőbe írva böngészni lehet az oldal könyvei között. Az enter billentyű lenyomásával vagy a „Keresés” gomb megnyomásával megtörténik a keresés, és megjelennek a találatok a keresett kifejezésre.

Ezeken az aloldalakon a könyvek desktop és mobil nézetben is táblázatos formában vannak felsorolva, az egyetlen különbség az adatok megjelenítésének formája és az adatok mennyisége. Minden könyv egy „blokkban” helyezkedik el. Ezekben benne van a könyv borítója, címe, írója vagy írói, leírása és ára. Mobil nézetben a leírás nem jelenik meg. Ezeken a blokkokon kiterjesztett link van, vagyis a blokk bármely pontjára kattintva az adott könyv adatlapjára kerül a felhasználó. Ezeken az aloldalakon maximum 10 darab könyv megjelenítésére van mód

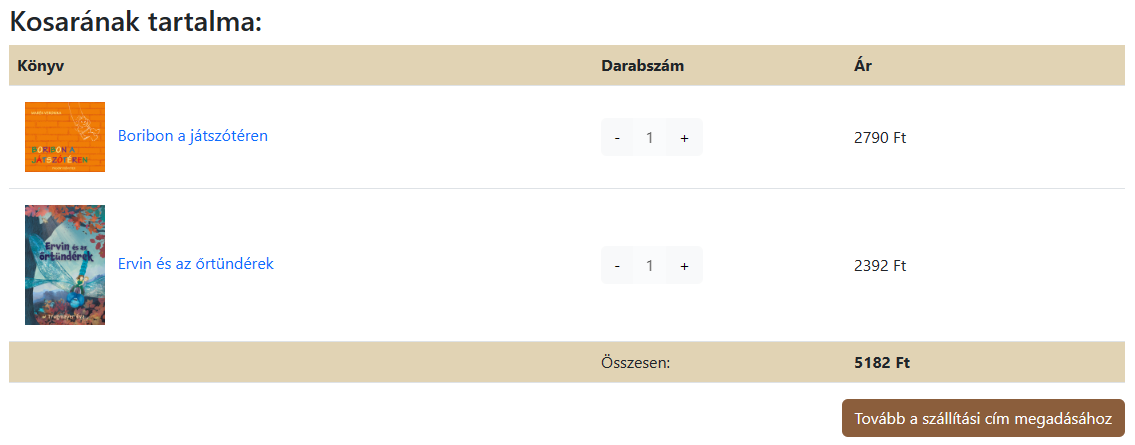
A könyv adatlapján olvasható a teljes leírás, valamint az egyéb adatok, amik az előző oldalon nem kerültek feltüntetésre. Látható továbbá, hogy a könyvből mennyi van készleten és hogy a megvásárlásával mennyi hűségpont szerezhető. Fontos megjegyezni, hogy a vendégek nem vesznek részt a hűségprogramban, így ők sem kapni, sem költeni nem tudnak pontokat.



. ábra: Minta egy könyv adatlapjára

A „Kosárba” gombra kattintva bekerül egy példány a könyvből a kosárba. Erről a felhasználó egy pop-up értesítést kap. A bal oldalon az író(k) neve(i) és a kiadó neve linkek, amik az író(k) és a kiadó oldalaira visznek, ahol megtekinthetőek a további könyveik.

A menüsáv jobb felső sarkában lévő kosár ikonra kattintva jut el a felhasználó a kosár aloldalra. Ezentúl a könyv kosárba helyezésénél megjelenő információs pup-upról is oda lehet jutni. A kosárban táblázatos formában megjelennek a kosár tartalmai. Amennyiben a kosárban nincsenek könyvek, az „A kosara üres.” felirat jelenik meg. Ha a kosár nem üres, táblázatos formában megjelennek a vásárolni kívánt könyvek, a vásárolni kívánt mennyiség, valamint az összesített ár.



. ábra: Minta a kosárra

A darabszám melletti „-” és „+” gombokkal csökkenthető, illetve növelhető a vásárolni kívánt mennyiség. Ha a darabszám egy, egy felugró ablak figyelmezteti a felhasználót, hogy ezzel törlésre kerül a könyv a kosárból, és a művelet nem visszavonható. Ha a felhasználó növeli a vásárolni kívánt mennyiséget, de az már elérte a készlet korlátját, egy információs pop-up értesíti a felhasználót, hogy elérte a készlet határát. A táblázat alján az összesen fizetendő összeg is feltüntetésre kerül. Amennyiben egy adott könyv éppen le van értékelve, azonnal a lejjebb vitt ár jelenik meg és kerül bele az összesítésbe.

Innen az online vásárlás folyamatában a szállítási cím megadásának lehetősége a "Tovább a szállítási cím megadásához" gomb használatával érhető el. E gombra kattintva a felhasználó azonnal továbblép a szállítási címek megadásának felületére.



. ábra: Az új szállítási cím megadása form

A megyét a legördülő listából kell kiválasztani, a várost, közterületet és az irányítószámot a megszokottak szerint kell megadni. A látogatóknak lehetőségük van cég nevében rendelni, és megjegyzést fűzni a szállítási címhez. A „Folytatás” gomb, mint ahogyan az a 6. ábrán látható, kezdetben le van tiltva. Ahogy a látogató kitölti a formot, a mezők ellenőrzésre kerülnek, és ha minden megfelelően ki van töltve, megnyomhatóvá válik a gomb. A gomb megnyomására a látogató továbblép a fizetési mód kiválasztásához. Az oldalon jelenleg csak az utánvételes fizetés elfogadására van mód.

A képen szöveg látható

Automatikusan generált leírás

. ábra: A fizetési mód kiválasztása form

A „Rendelés véglegesítése” gombra kattintva a látogató leadja rendelését, melynek kiszállítása a megadott címre egy hét elteltével várható, amiről az oldal tájékoztatja is.

A menüsor jobb felső sarkában jobbról a második gombra kattintva megjelenik a bejelentkezési pop-up. Ezen a formon amennyiben a látogató rendelkezik fiókkal, a megfelelő helyekre beírva a felhasználónevet és a jelszót képes azonosítani magát és belépni. Ehhez a „Belépés” gombra kell rányomnia a látogatónak. A „Mégse” vagy az „X” gombokra kattintva bezárhatja a pop-upot.

A képen szöveg látható

Automatikusan generált leírás

. ábra: A belépés pop-up

Amennyiben a látogató még nem rendelkezik fiókkal, a „Regisztráljon most!” feliratú linkre kattintva átválthat a regisztrációs pop-upra. Egy e-mail cím, egy felhasználónév és egy jelszó megadásával regisztrálhat egy új fiókot.

A képen szöveg látható

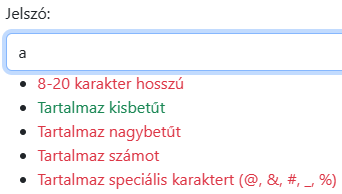
Automatikusan generált leírás

. ábra: a regisztrációs pop-up

Fontos, hogy ugyanazzal az e-mail címmel vagy felhasználónévvel nem rendelkezhet két fiók. A jelszó mezőre kattintva megjelennek a jelszó elfogadásának a feltételei. A jelszónak:

* minimum 8 és maximum 20 karakter hosszúnak kell lennie,
* tartalmaznia kell legalább egy kisbetűs karaktert,
* tartalmaznia kell legalább egy nagybetűs karaktert,
* tartalmaznia kell legalább egy számot,
* valamint tartalmaznia kell legalább egy speciális karaktert.

A pillanatban a következő speciális karaktereket fogadja el a weboldal: @, &, #, \_, %. A jelszó érvényességének vizsgálatát a látogató valós időben követheti, ahogyan az látható a 10. és 11. ábrákon. Ezekre a szabályokra a felhasználói fiókok biztonsága érdekében van szükség. Fontos, hogy minden felhasználó jól jegyezze meg a megadott felhasználónevét és jelszavát, mert azok módosítására a pillanatban nem áll mód!



. ábra: példa helytelen jelszóra

A képen szöveg látható

Automatikusan generált leírás

. ábra: példa helyes jelszóra

A „Regisztráció” gombra kattintva a látogató létrehozhatja új fiókját, melybe a rendszer automatikusan belépteti. Ha mégis rendelkezne fiókkal, a „Jelentkezzen be!” feliratú linkkel visszatérhet a bejelentkezés pop-upra. A belépés pop-uphoz hasonlóan a „Mégse” és az „X” gombok bezárják a felugró ablakot.

### Felhasználói funkciók

A képen szöveg látható

Automatikusan generált leírásSikeres belépés után a főoldalon módosításokat vehet észre a felhasználó. A menüsorban a profil ikon zöld színűre vált, ezzel jelezve, hogy be van lépve, mint ahogyan ez a 12. ábrán létható. Erre rákkattintva megjelenik a belépett felhasználók számára szolgáló pop-up.

. ábra:belépés után a profil ikon zöld színűre vált

A képen szöveg látható

Automatikusan generált leírás

. ábra: A belépett felhasználók pop-up "jazehin" nevű példafelhasználónál

A 13. ábrának megfelelően a felugró ablakban megjelenik a felhasználó neve. Ezen túl a „Profil megtekintése” gombra kattintva a felhasználó a profil oldalára kerülhet, míg a „Kijelentkezés” gomba kattintva a kiléphet a fiókjából, visszatérve a vendégi státuszhoz. A korábbiakhoz hasonlóan a „Bezárás” és az „X” gombokkal vissza lehet térni a weboldal tartalmához.



. ábra: "jazehin" nevű példafelhasználó profilja

Egy felhasználó profilja nagyjából hasonlóan néz ki, mint a 14. ábrán látható mintaprofil. A profilok öt darab szekciót tartalmaznak, amelyek a következők:

* Felhasználói adatok: itt tekinthetőek meg a felhasználó adatai, mint a felhasználóneve, e-mail címe, teljes neve, neme, születési dátuma és az ez alapján számolt kora, valamint telefonszáma.

A „Szerkesztés” linkre kattintva a táblázat egy formmá alakul, ahol a felhasználó meg tudja adni a nevét, a nemét, a születési idejét és a telefonszámát. Fontos megjegyezni, hogy ezeket nem kötelező megadni, viszont a nem és az életkor segít személyre szabott könyvajánlatokat adni a felhasználónak. A megadott adatokat a „Mentés” linkre kattintva tudja elmenteni.

* Szállítási címek: itt láthatóak felsorolásszerűen a felhasználó által megadott és elmentett szállítási címek.

Az „Új cím megadása” linkre kattintva a panel alján megjelenik egy form, ahova meg lehet adni egy elmentendő szállítási címhez tartozó céget, megyét, települést, közterületet, irányítószámot és megjegyzést. Cég és megjegyzés megadása opcionális, azonban a többi adat felvitele kötelező. A „Mentés” gombra kattintva a cím hozzáadódik a listához az oldal újratöltése után.

Bármely cím mellett a „Törlés” linkre kattintva az adott cím kitörlődik a listából.

* Érdeklődések: itt láthatóak a témák, vagyis műfajok amelyekben a könyvesbolt könyveket árul. Az üres karika azt jelenti, hogy a felhasználó nem érdekelt a témában, a teli karika pedig azt, hogy érdekli a felhasználót az adott műfaj.
* Hűségprogram: egy rövid magyarázó szöveget követően láthatja a felhasználó, hogy hány hűségponttal rendelkezik, vagyis hogy mennyit tud jövőbeli vásárlásain megtakarítani.
* Korábbi rendelések: itt láthatja a felhasználó táblázatos formában a korábbi rendeléseinek pontos idejét, valamint a kifizetett összeget.

# Tesztelés

Weboldalam fejlesztését dinamikus tesztelési módszerekkel támogattam, ami azt jelenti, hogy a weboldalt folyamatosan, futás közben teszteltem. Ennek kettő fajtája van, a fekete- és a fehérdobozos tesztelés. Előbbiben a kód nem ismert vagy nem átlátható, így csak a bemenetet és a kimenetet ellenőrizzük, az utóbbiban viszont a futási logikát is vizsgáljuk. Ezekhez érdemes és kell is teszteseteket írni. A jó teszteset nagy valószínűséggel egy még felfedetlen hibát mutat ki az alkalmazásban.

Az egyik legátfogóbb tesztelést kívánó része a weboldalamnak az adminisztrátori felület könyveket felvevő és módosító formja volt. Ennek oka a form mérete: 16-féle adatot kell felvinni a könyvről (amelyből csak 4 opcionális). Erről további információ olvasható az adatbázisról szóló szekcióban, és a felhasználói dokumentációban.

[tesztelés részletei]

Megjegyzendő, hogy nincs hibamentes program, így nem vállalok garanciát a hibamentességre.

# Továbbfejlesztési lehetőségek

A szoftver fejlesztése közben rengeteg további ötlet jutott eszembe, amelyekkel a funkcionalitása kibővíthető lenne. Azokat, amelyeket időszűke keretében nem sikerült megvalósítani, az alábbiakban részletezem:

* Az adminként elérhető könyvmódosítási és -törlési felületén a könyvekre jelenleg csak ISBN alapján lehet rákeresni. Ez észrevételem szerint időigényes, mert az emberek nem azonosítókhoz vannak szokva, hanem nevekhez. Pont ezért érdemes lehetne az oldal betöltésekor egy táblázatban megjeleníteni a könyveket (például oldalakra osztva a könyveket, hogy ne legyen túl hosszú az oldal), ahol ISBN és könyvcím alapján lehet szűrni a könyveket, majd a táblázatból kiválasztani a módosítani kívántat.
* Az oldal menüsorában lehetőség van könyvekre keresni cím alapján. Előnyös lehetne, ha a címen kívül például az író nevére vagy a kiadóra is lehetne szűrni.
* Az egyes aloldalak találati listájában a sorbarendezés az adatbázisra van bízva. Az oldalak többségénél lehetőség van ár és egyéb paraméterek alapján szűrni és rendezni a találatokat. Ennek a megvalósítása is előnyös lehetne az oldal számára.
* Mindenkivel előfordul, hogy elfelejti a jelszavát, vagy rosszul emlékszik arra. Gyakori funkciója a webáruházaknak (és más weboldalaknak is), hogy adnak valamilyen lehetőséget új jelszó megadására egy e-mailben kiküldött linken keresztül. Ez a funkció is a webáruház előnyére lenne.

# Ábrajegyzék

1. ábra: EER diagram a `bookstore` adatbázisról 8

2. ábra: A <datalist> tag segítségével a korábban előfordult kötéstípusokat felajánlja a böngésző 9

3. ábra: A vendégeket a bestsellerek aloldalra írányítja az oldal 18

4. ábra: Minta egy könyv adatlapjára 19

5. ábra: Minta a kosárra 20

6. ábra: Az új szállítási cím megadása form 21

7. ábra: A fizetési mód kiválasztása form 21

8. ábra: A belépés pop-up 22

9. ábra: A regisztrációs pop-up 23

10. ábra: Példa helytelen jelszóra 11. ábra: Példa helyes jelszóra 23

12. ábra: A profil ikon belépés után zöld színűre vált 23

13. ábra: A belépett felhasználók pop-up "jazehin" nevű példafelhasználónál 24

14. ábra: "jazehin" nevű példafelhasználó profilja 24

1. Forrás: Wordsrated (<https://wordsrated.com/american-reading-habits-study/>), letöltés ideje: 2023. 04. 03. [↑](#footnote-ref-1)