

Bookly

Vizsgaremek dokumentáció

Készítette: [Kálmán László , Mészáros Tibor]

2025

Tartalomjegyzék (a dokumentum végén automatikusan generálható)

1. Bevezetés

2. Fejlesztői dokumentáció

2.1 Fejlesztőkörnyezet

2.2 Adatbázis struktúra

2.3 Algoritmusok

2.4 Tesztelés

2.5 Fejlesztési lehetőségek

3. Felhasználói dokumentáció

3.1 Használati feltételek

3.2 Regisztráció, belépés

3.3 Profilkezelés

3.4 Könyvkeresés és értékelés

3.5 Kosár és vásárlás

3.6 Admin funkciók

4. Összefoglalás

5. Irodalomjegyzék

6. Mellékletek – képernyőképek

1. Bevezetés

A Bookli egy webalapú könyves közösségi és értékelő platform, melyet elsősorban könyvszerető felhasználóknak, valamint adminisztratív kezelőknek terveztünk. A weboldal lehetőséget biztosít regisztrált felhasználók számára könyvek böngészésére, értékelésére, vélemények beküldésére és kosárba helyezésre, miközben az adminisztrátorok a háttérben karbantarthatják a tartalmat és a felhasználói tevékenységeket.

A projekt célja egy olyan egyszerűen használható, mégis funkciógazdag webalkalmazás létrehozása volt, amely egyesíti a könyvesbolt, véleményező és közösségi portálok legfőbb tulajdonságait. A munkát PHP, MySQL, HTML, CSS és JavaScript technológiák felhasználásával valósítottuk meg, lokális szerveren, XAMPP környezetben.

A témaválasztás oka, hogy jelenleg több könyvértékelő weboldal létezik, de ezek túlnyomórészt idegennyelvűek, vagy hiányos a funkciókészletük. A Bookli egy magyar nyelvű, iskolai célra fejlesztett, de valódi célokat szolgáló rendszerként készült.

A dokumentáció célja, hogy bemutassa a fejlesztés menetét, az alkalmazott technológiákat, az adatbázis-felépítést, a főbb algoritmusokat, valamint részletes útmutatót nyújtson a program használatához. A végén mellékletek formájában képernyőképekkel is illusztráljuk a működést.

2. Fejlesztői dokumentáció

2.1 Fejlesztőkörnyezet

A Bookli webalkalmazás fejlesztése során az alábbi hardveres és szoftveres környezet került felhasználásra:

Hardver:

- Asztali számítógép
- Intel Core i3 3.20 GHz processzor
- 8 GB RAM
- 500 GB HDD
- 1920x1080 képernyőfelbontás

Szoftver:

- Operációs rendszer: Windows 10 (64 bit)
- Fejlesztőkörnyezet: Visual Studio Code
- Böngészők: Google Chrome
- Lokális webszerver: XAMPP (PHP 8.1, MySQL 5.7, Apache 2.4)
- Adatbázis-kezelő: phpMyAdmin

A választott környezet előnye, hogy könnyen beállítható, rugalmas, és széleskörűen támogatja a webes technológiákat. A fejlesztés során a Visual Studio Code beépített bővítményei segítették a gyors és hatékony munkát.

2.2 Adatbázis struktúra

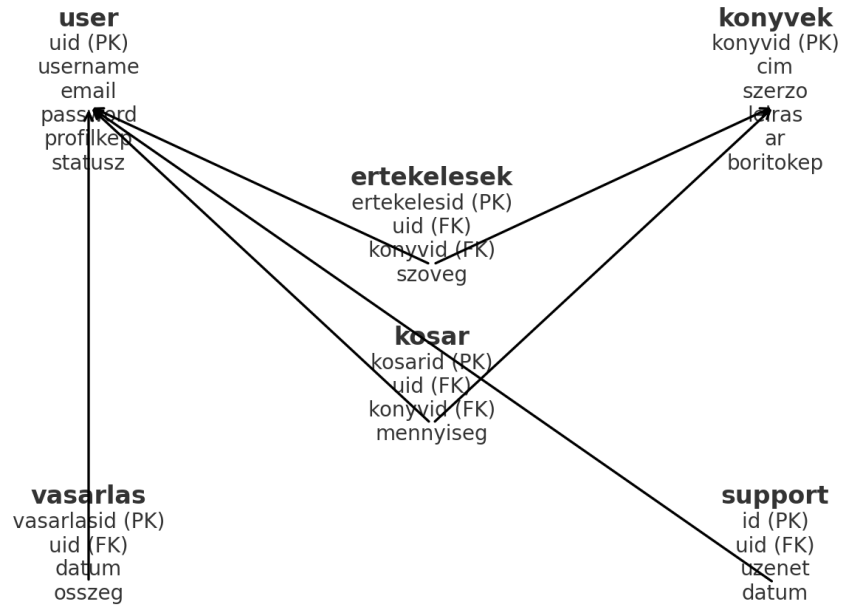
A Bookli alkalmazás egy `bookl` nevű relációs MySQL adatbázist használ, amely a könyvek, felhasználók, értékelések, kosár és egyéb adatok kezelésére szolgál. Az adatbázis főbb táblái:

- `user` – felhasználói fiókok adatai
- `konyvek` – könyvek részletes leírása
- `ertekelek` – felhasználói értékelések
- `kosar` – kosárba helyezett elemek
- `vasarlas` – fizetési és rendelési adatok
- `support` – kapcsolatfelvételi űrlapból érkező üzenetek

Az adatbázisban egyértelmű elsődleges kulcsok (`id`) találhatók minden táblában, és idegen kulcsos kapcsolatok biztosítják a táblák közötti integritást. A következő oldalon

ER-diagram mutatja a teljes szerkezetet.

Az alábbi ábra az adatbázis táblák közötti kapcsolatokat szemlélteti:



2.3 Algoritmusok

A program működésének kulcsa a szerveroldali logika, amelyet PHP nyelven valósítottunk meg. Az alábbiakban néhány fontosabb, egyedibb algoritmust ismertetünk, melyek jellemzően a felhasználói folyamatokat és az adminisztrációt vezérlik.

1. Regisztrációs adatok ellenőrzése (reg_ir.php)

```
3  include("kapcsolat.php");
4
5
6  if (!isset($_POST['username']) || trim($_POST['username']) == "") {
7      echo "<script>alert('Nem adtad meg a felhasználóneved!')</script>";
8      die("<script>parent.location.href = './?p=reg'</script>");
9  }
10
11 if (!isset($_POST['email']) || trim($_POST['email']) == "") {
12     echo "<script>alert('Nem adtad meg az email címed!')</script>";
13     die("<script>parent.location.href = './?p=reg'</script>");
14 }
15
16 if (!isset($_POST['password']) || trim($_POST['password']) == "") {
17     echo "<script>alert('Nem adtad meg a jelszavad!')</script>";
18     die("<script>parent.location.href = './?p=reg'</script>");
19 }
20
21 if (!isset($_POST['checkbox'])) {
22     echo "<script>alert('Nem fogadtad el a feltételeket!')</script>";
23     die("<script>parent.location.href = './?p=reg'</script>");
24 }
25
26
27 $username = mysqli_real_escape_string($adb, trim($_POST['username']));
28 $email = mysqli_real_escape_string($adb, trim($_POST['email']));
29 $password = mysqli_real_escape_string($adb, trim($_POST['password']));
30
31
32 $upw = md5($password);
33
34
35 $sql = "
36 INSERT INTO user (uid, username, uemail, upassword, udatum, uip, usession, ustatusz, ukomment)
37 VALUES (NULL, '$username', '$email', '$upw', NOW(), '', '', 'a', '')
38 ";
39
40 if (mysqli_query($adb, $sql)) {
41     echo "<script>alert('Sikeres regisztrálás')</script>";
42     echo "<script>parent.location.href = './?p=login'</script>";
43 } else {
44     echo "<script>alert('Hiba történt a regisztráció során!')</script>";
45     echo "<script>parent.location.href = './?p=reg'</script>";
46 }
```

A rendszer ellenőrzi, hogy minden kötelező mező ki van-e töltve, a jelszavak megegyeznek-e, illetve hogy az e-mail cím már nem szerepel-e az adatbázisban. Amennyiben minden feltétel teljesül, a felhasználó adatait elmenti az adatbázisba.

2. Bejelentkezés és jogosultságkezelés (login_ir.php)

```
<?php
    session_start() ;
    include( "kapcsolat.php" ) ;
    $email = $_POST["email"];
    $pw = md5($_POST["pw"]);
    $userek = mysqli_query( $adb , "
    SELECT * FROM user
    WHERE (uemail='$email' OR username='$email')
    AND      upassword = '$pw'
    " ) ;

    if( mysqli_num_rows($userek)==0 )
    {
        print "<script> alert('Hibás belépési adatok!') </script>" ;
    }
    else
    {
        $user = mysqli_fetch_array($userek) ;
        $_SESSION['uid'] = $user['uid'] ;
        $_SESSION['username'] = $user['username'] ;
        $logip = $_SERVER['REMOTE_ADDR'];
        $logsess = substr(session_id(), 0 , 8);
        if (isset($_SESSION['uid'])) {
            $uid = $_SESSION['uid'];
        } else{
            die("<script>alert('már be vagy jelentkezve')</script>") ;
        }

        mysqli_query($adb,
        "INSERT INTO login (    logid,    logdate,    logip,    logsession,    luid)
        VALUES                (    NULL,        NOW(), '$logip',    '$logsess', '$uid')");

        mysqli_close( $adb ) ;

        print "<script> parent.location.href = './?p=konyvek' </script>" ;
    }
?>
```

A bejelentkezés során az e-mail és jelszó páros validálása történik meg. A rendszer lekérdezi az adatbázisból az e-mail címet, és összehasonlítja a jelszót. Sikeres bejelentkezés esetén session változókat állít be.

3. Könyv értékelés rögzítése (konyv_velemeney_ir.php)

```
<div class="rating-section">
    <h3>Értékelje a könyvet</h3>
    <?php if (isset($error)) echo "<p style='color: red;'>$error</p>"; >
    <form action="konyv_velemeney_ir.php" method="POST" target="kisablak">
        <?php echo ' <input type="hidden" name="id" id="id" value=".' . $data['id'] . '>'; >
        <textarea name="comment" id="comment" rows="4" placeholder="Írja meg véleményét..."></textarea>
        <input type="submit" value="Küldés"></input>
    </form>
</div>

<div class="reviews-section">
    <h3>Felhasználói értékelések</h3>
    <?php
    if(!isset($velemeney))<|
        echo "Még nincsenek hozzászólások ehhez a könyvhöz." ;
    <|
    else{
        $ab = mysqli_query($adb , "SELECT * FROM erkekelesek INNER JOIN user ON erkekelesek.uid = user.uid WHERE kid = $_GET[id] AND erkekelesek.status = 'a'");
        while ($a = mysqli_fetch_array($ab, MYSQLI_ASSOC)) {
            echo '<div class="review">';
            if (isset($_SESSION['uid'])) {
                echo '<a href=".' . $a['uid'] . '" style="text-decoration: none; color: inherit;"><strong>'.htmlspecialchars($a["username"]).'</strong></a>';
            }else{
                echo '<a href=".' . $a['uid'] . '" style="text-decoration: none; color: inherit;"><strong>'.htmlspecialchars($a["username"]).'</strong></a>';
            }

            echo '<p>'. htmlspecialchars($a["eszoveg"]).'</p>';
            echo '<small>'. htmlspecialchars($a["edatum"]).'</small>';
            echo '</div>';
        }
    }
}
```

```
<?php
session_start() ;
include("kapcsolat.php");
if (isset($_SESSION['uid'])) {
    if (!empty($_POST['comment'])) {
        $time = date('Ymd');
        $comment = mysqli_real_escape_string($adb, $_POST['comment']);
        mysqli_query($adb, "INSERT INTO `erkekelesek`(`eid`, `uid`, `kid`, `eszoveg`, `edatum`, `status`)
            VALUES (NULL, $_SESSION[uid], $_POST[id], '$comment', '$time', 'a')");
        print "<script> parent.location.href = parent.location.href </script>" ;
    }else{
        print "<script>alert('')</script>";
    }
}
else{
    print "<script> parent.location.href = '.' . $a['uid'] . '" style="text-decoration: none; color: inherit;"><strong>'.htmlspecialchars($a["username"]).'</strong></a>';
}
}
mysqli_close($adb);
?>
```

A felhasználó kiválasztott könyvhöz értékelést írhat. A szöveg és a könyv azonosítója POST metódussal érkezik, és bekerül az `erkekelesek` táblába, hozzárendelve az aktuális felhasználóhoz.

4. Kosár művelet (add_to_cart.php, remove_from_cart.php)


```

<?php
session_start();
include('kapcsolat.php');

if (!isset($_SESSION['uid'])) {
    print "<script> alert('Jelentkez be!')</script>" ;
}else{
    $konyvek = mysqli_query($adb, "SELECT * FROM kosar WHERE uid = '$_SESSION[uid]' AND statusz = '1' AND kid = '$_POST[id]'");
    if (mysqli_num_rows($konyvek) > 0) {
        print "<script> alert('Már a kosárban van a könyv!')</script>" ;
    }else{
        mysqli_query($adb, "INSERT INTO `kosar` (`koid`, `uid`, `kid`, `statusz`)
        VALUES ( NULL, $_SESSION[uid], $_POST[id], 1)");
    }
}
print"<script>parent.location.href=parent.location.href</script>";
?>

```

```

<?php
session_start();
require_once 'kapcsolat.php';

if (!isset($_SESSION['uid'])) {
    echo "<script> alert('Jelentkezz be először!')</script>";
}else{
    if ($_POST['id']) {
        echo "<script> alert('nem létezik ilyen könyv!')</script>";
    }else {
        echo'lefut';
        $uid = $_SESSION['uid'];
        $query = "UPDATE kosar SET `statusz`= 0 WHERE uid = ? AND kid = ? AND statusz = 1 LIMIT 1";
        $stmt = mysqli_prepare($adb, $query);
        mysqli_stmt_bind_param($stmt, "is", $uid, $_POST['id']);
        mysqli_stmt_execute($stmt);
    }
}
print "<script> parent.location.href = parent.location.href </script>" ;
?>

```

A felhasználó könyveket tehet a kosárba. A rendszer ellenőrzi, hogy ugyanaz a könyv már szerepel-e. Törlés esetén a bejegyzés eltávolításra kerül a kosárból.

5. Admin felhasználó törlés művelete (admin_delete_user.php)

```

<?php
include("../kapcsolat.php");

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $uid = $_POST['uid'];

    $query = "UPDATE user SET ustatusz = 'c' WHERE uid = ?";
    $stmt = mysqli_prepare($adb, $query);
    mysqli_stmt_bind_param($stmt, "i", $uid);

    if (mysqli_stmt_execute($stmt)) {
        echo "Felhasználó sikeresen törölve.";
    } else {
        echo "Hiba történt a felhasználó törlésekor.";
    }

    mysqli_stmt_close($stmt);
}

mysqli_close($adb);
header("Location: ../?p=userek");
exit;
?>

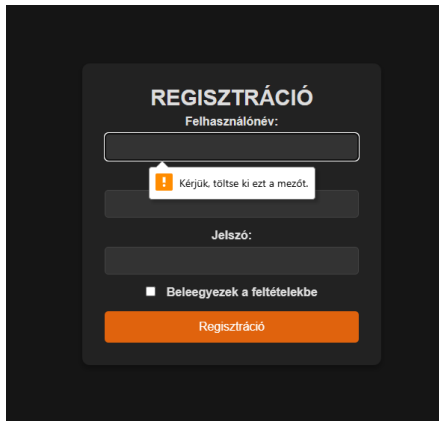
```

Adminisztrátor jogosultságú felhasználók lehetőséget kapnak más felhasználók törlésére. Az algoritmus SQL `DELETE` utasítással eltávolítja a kiválasztott felhasználót, és frissíti a felhasználói listát.

2.4 Tesztelés

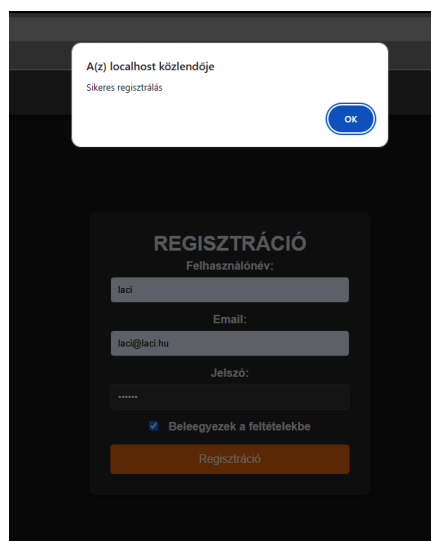
A tesztelés célja annak bizonyítása, hogy az alkalmazás minden funkciója megfelelően működik a különféle használati körülmények között is. A teszteket manuálisan végeztük a Google Chrome böngészőben. Az alábbi teszteseteket hajtottuk végre:

- 1. Regisztráció üres mezőkkel – Hibaüzenet jelenik meg.



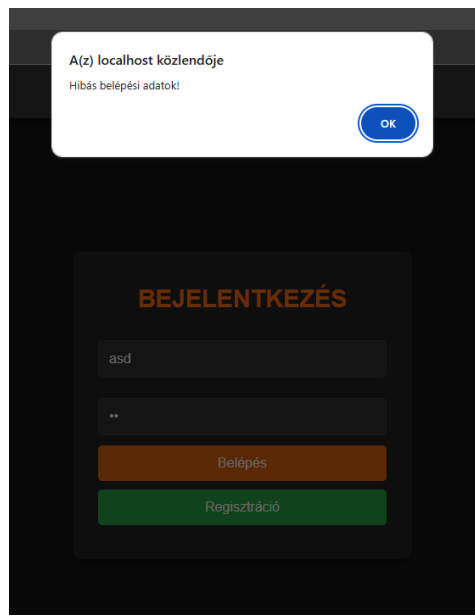
The screenshot shows a registration form titled "REGISZTRÁCIÓ" on a dark background. The form has two input fields: "Felhasználónév:" (Username) and "Jelszó:" (Password). The "Felhasználónév:" field is empty and has a red error message icon (an exclamation mark in a square) next to it with the text "Kérjük, töltsd ki ezt a mezőt." (Please fill in this field). Below the fields is a checkbox labeled "Beleegyezek a feltételekbe" (I agree to the terms) and a blue "Regisztráció" (Registration) button.

- 2. Helyes regisztráció – Sikeres mentés, átirányítás login oldalra.

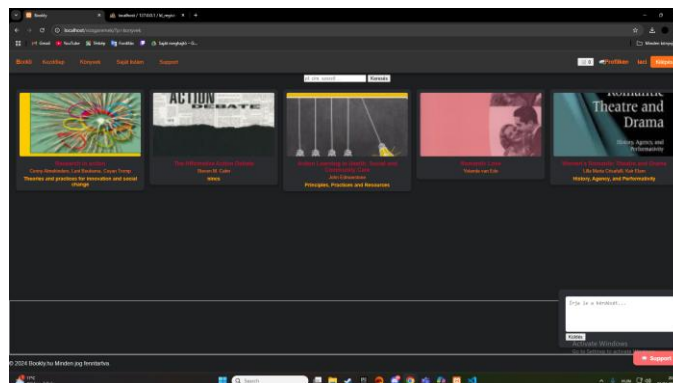


The screenshot shows the same registration form as above, but with a success message. A white toast notification is displayed at the top with the text "A(z) localhost közlendője" (Message from localhost) and "Sikeres regisztrálás" (Successful registration). Below the notification, the "Felhasználónév:" field now contains the text "laci" and the "Email:" field contains "laci@laci.hu". The "Jelszó:" field is still empty. The checkbox "Beleegyezek a feltételekbe" is now checked, and the blue "Regisztráció" button is still visible.

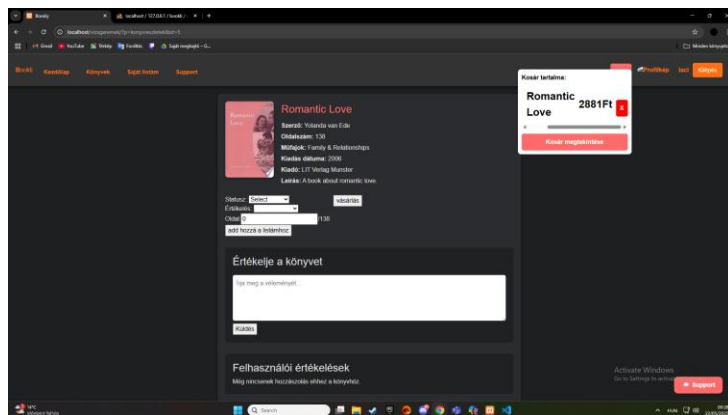
- 3. Bejelentkezés helytelen jelszóval – 'Hibás belépési adatok!' Üzenet.



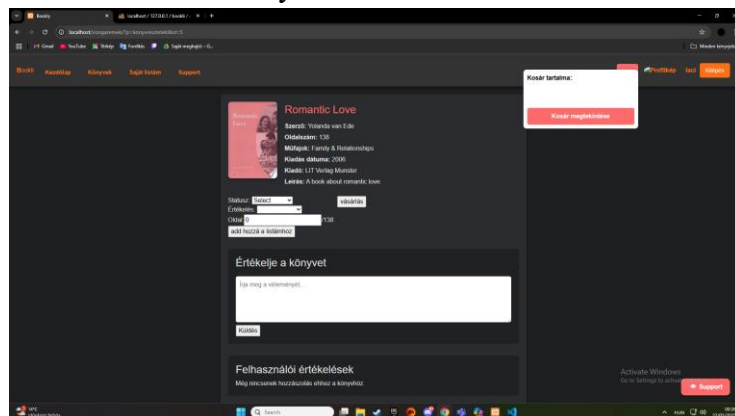
- 4. Helyes belépés – Főoldal betöltődik, session létrejön.



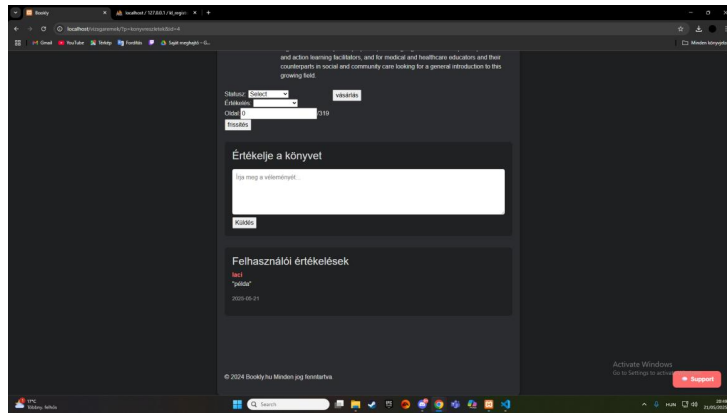
- 5. Könyv kosárba helyezése – Megjelenik a kosárban.



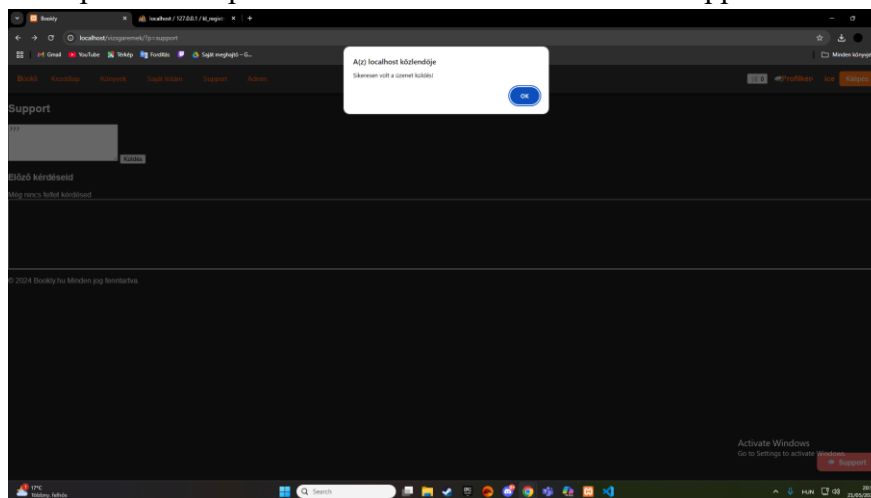
- 6. Kosárból könyv törlése – Eltűnik a kosárból.



- 7. Értékelés beküldése – Adatbázisba kerül, megjelenik a könyvnél.



- 8. Admin felhasználó töröl egy könyvet – Eltűnik a listából
- 9. Kapcsolat űrlap üzenetküldés – Üzenet bekerül a support táblába.



- 10. SQL injection próbálkozás – Értelmezhetetlen szöveggént tárolódik (védelem aktív).

A fenti tesztek mindegyike sikeresen zárult. A rendszer stabilan működik több böngészőben, és nem érzékeny a képernyőfelbontás változására sem.

2.5 Fejlesztési lehetőségek

A program továbbfejlesztésére számos lehetőség kínálkozik. A jelenlegi működési alapokon az alábbi irányokban tervezzük a bővítést:

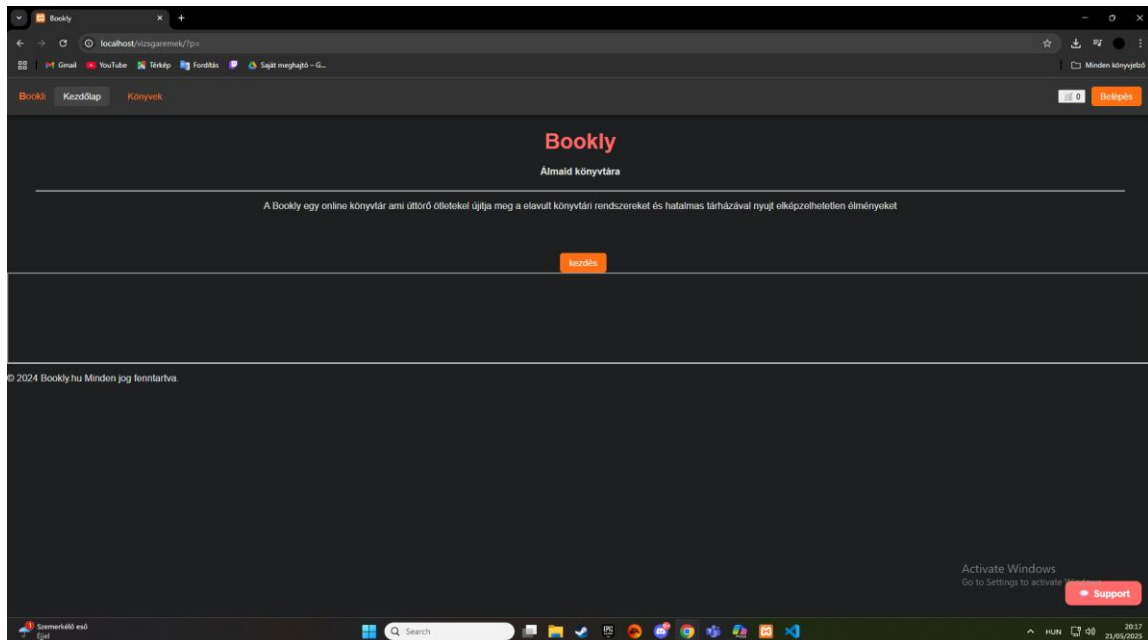
- Fiókaktiválás e-mailes megerősítéssel
- Profiloldalon elérhető könyvtörténet és kedvencek lista
- Reszponzív, mobilbarát dizájn bevezetése
- Dinamikus keresőmező AJAX alapokon
- Admin felület bővítése export és statisztika funkciókkal
- Vásárlások nyomon követése adminisztrációs nézetből ezek a lehetőségek növelnék a felhasználói élményt és a rendszer skálázhatóságát a jövőbeli alkalmazás során.

3. Felhasználói dokumentáció

3.1 Használati feltételek

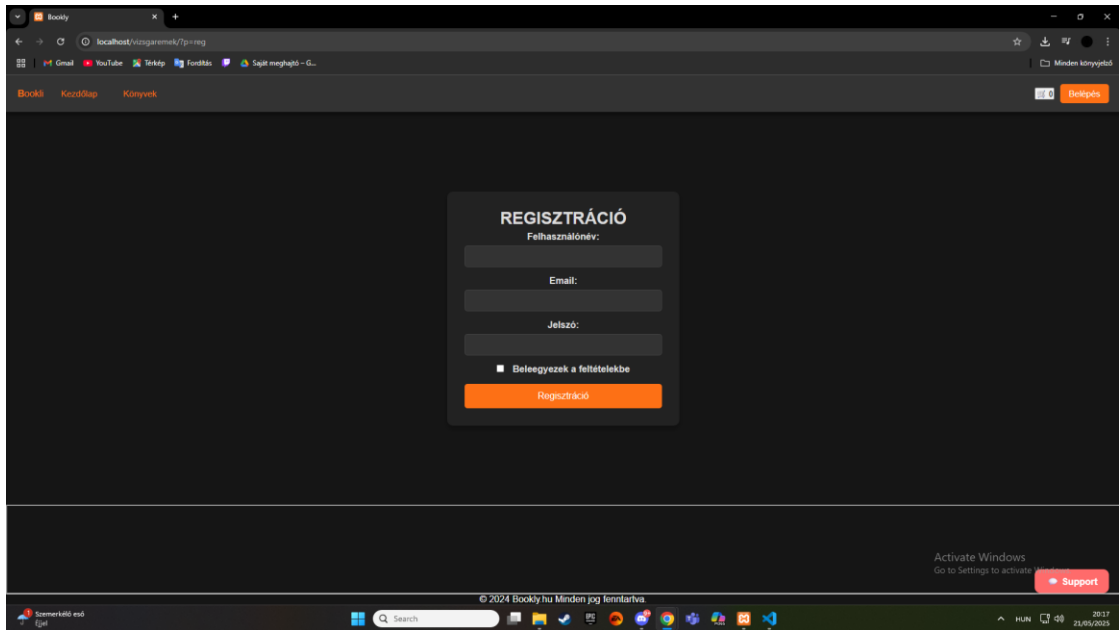
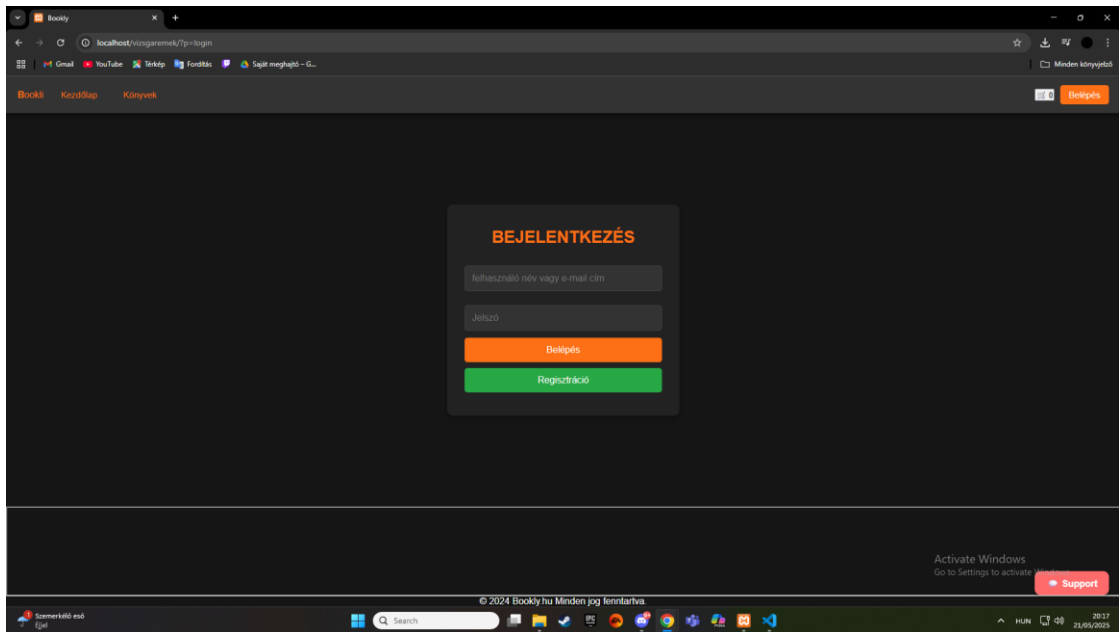
A Bookli egy online könyves alkalmazás, amely számítógépről és mobil böngészőből egyaránt elérhető. A rendszer használatához internetkapcsolat, egy modern webböngésző (pl. Google Chrome, Microsoft Edge), valamint egy lokális vagy távoli PHP + MySQL szerverkörnyezet szükséges.

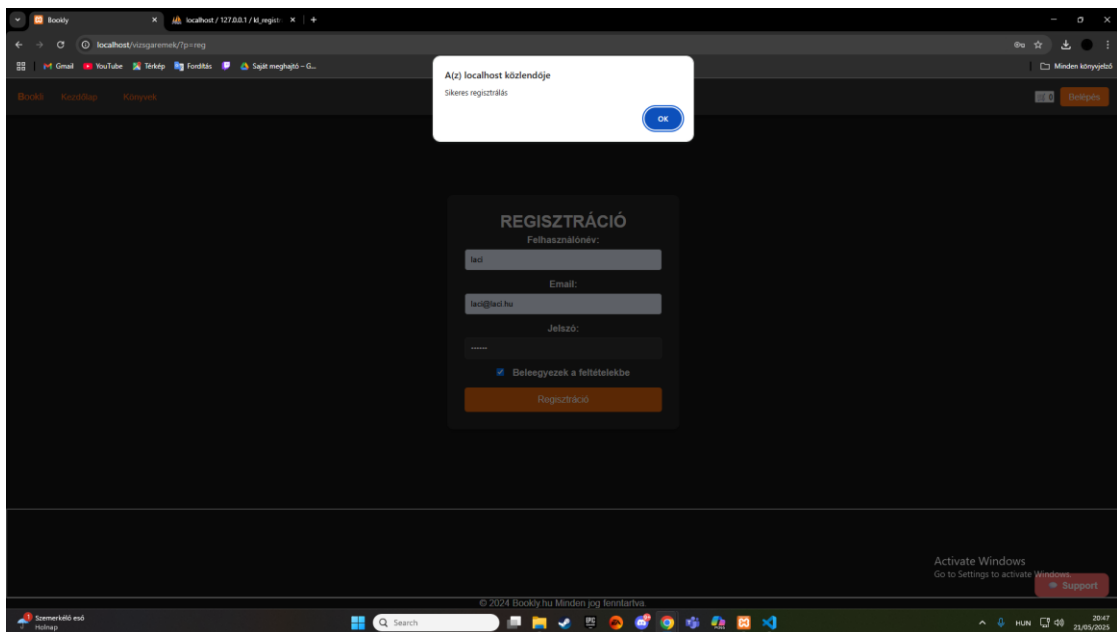
Kezdőlap:



3.2 Regisztráció, belépés

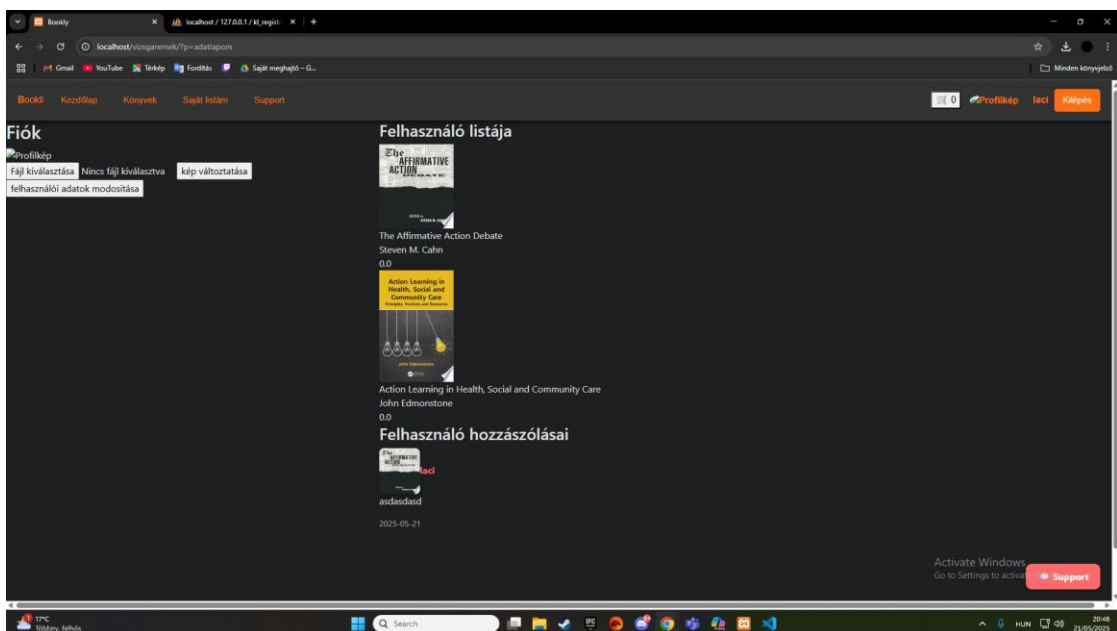
A kezdőoldalról elérhető a regisztrációs űrlap, ahol meg kell adni felhasználónevet, e-mail címet, jelszót, és profilképet lehet feltölteni. A regisztráció sikeres befejezése után a felhasználó bejelentkezhet a rendszerbe.





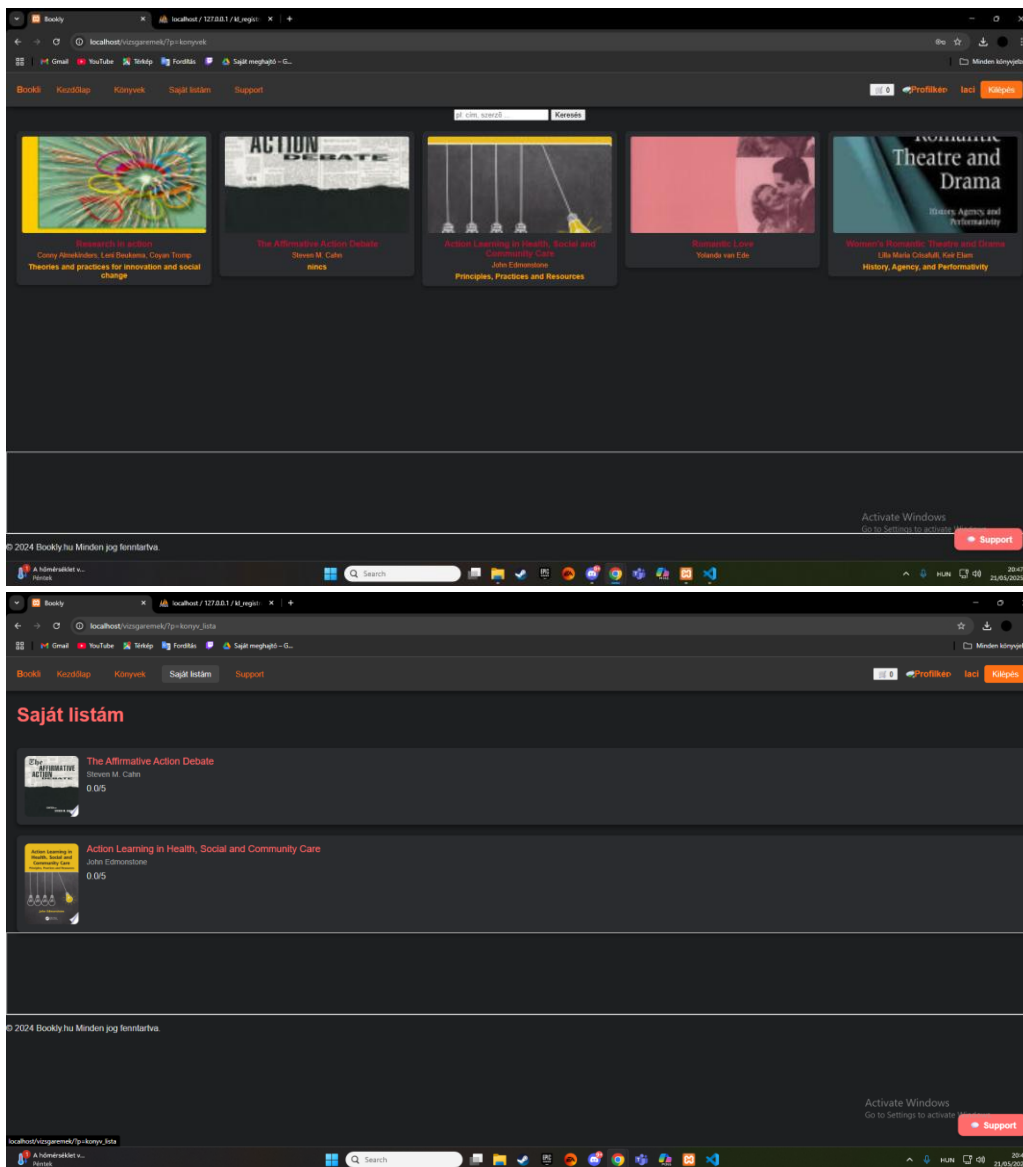
Hibás adatmegadás esetén a rendszer hibaüzenetet ad vissza, pl.: "Nem adtad meg a jelszót!" vagy "Ez az e-mail cím már foglalt!".

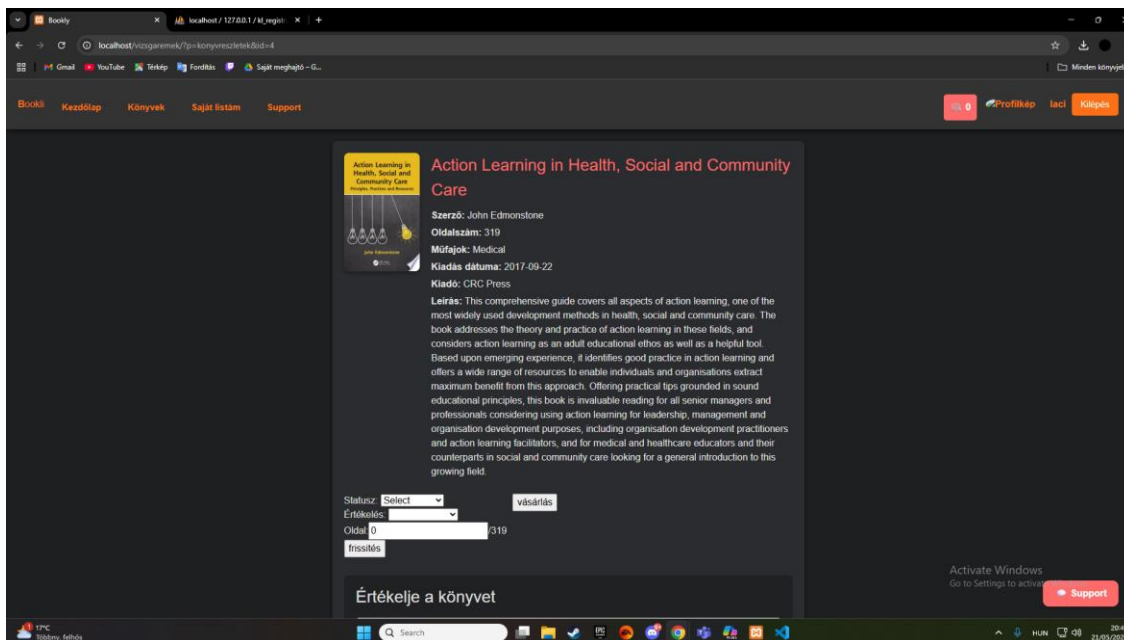
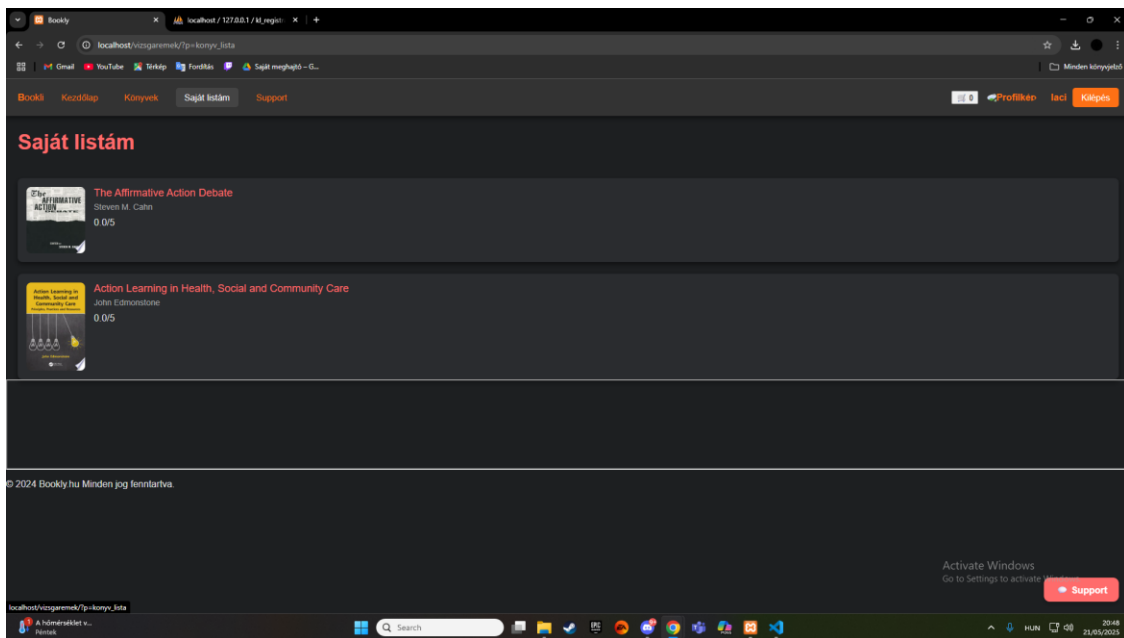
3.3 Profilkezelés

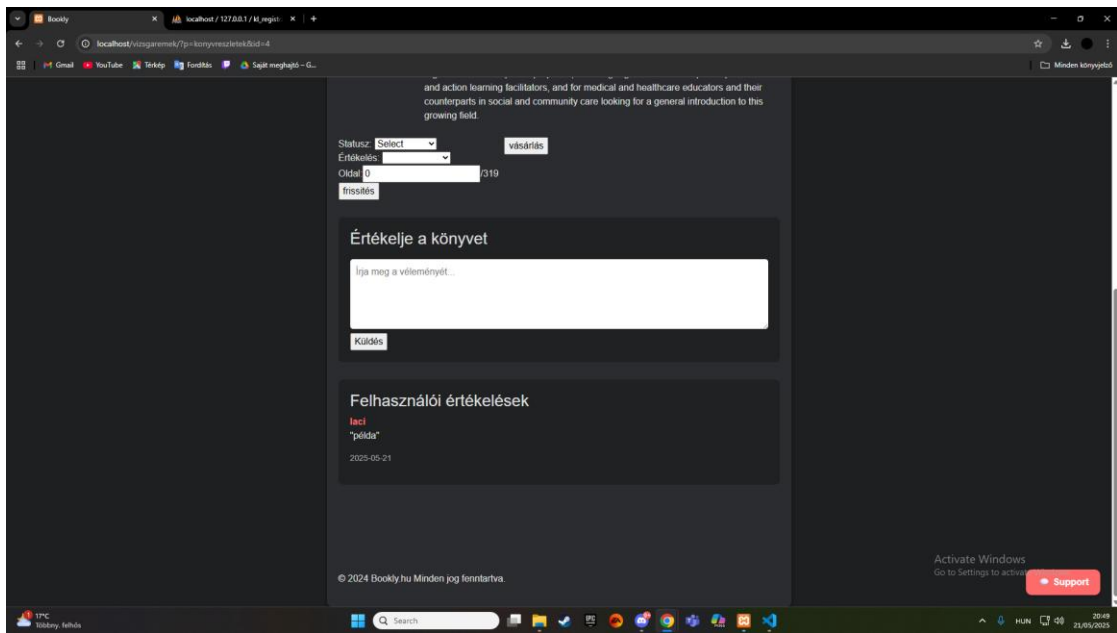
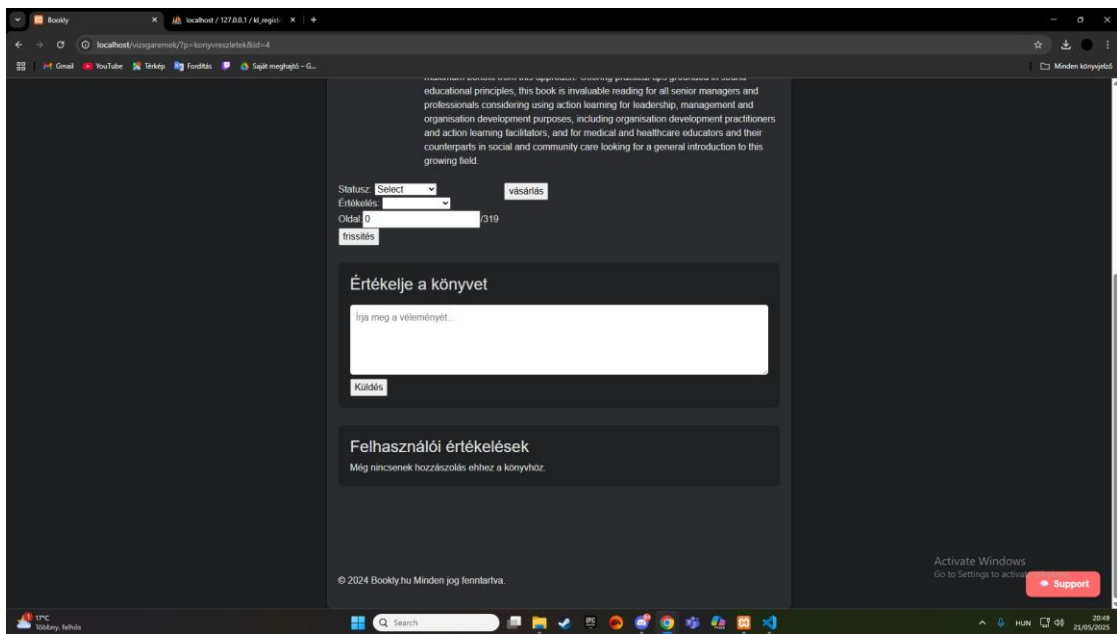


Belépést követően a felhasználó a saját profiljához fér hozzá, ahol módosíthatja adatait, jelszavát, illetve profilképet is cserélhet. Az oldalon megjelennek a korábbi értékelések is, valamint az esetlegesen beérkezett támogatási válaszok.

3.4 Könyvkeresés és értékelés



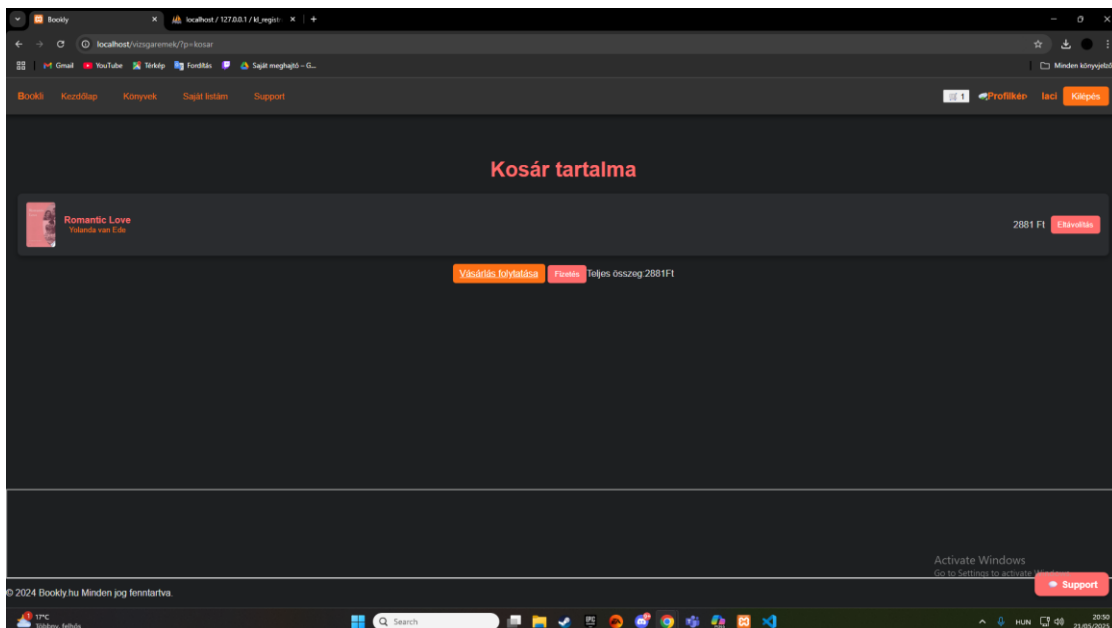
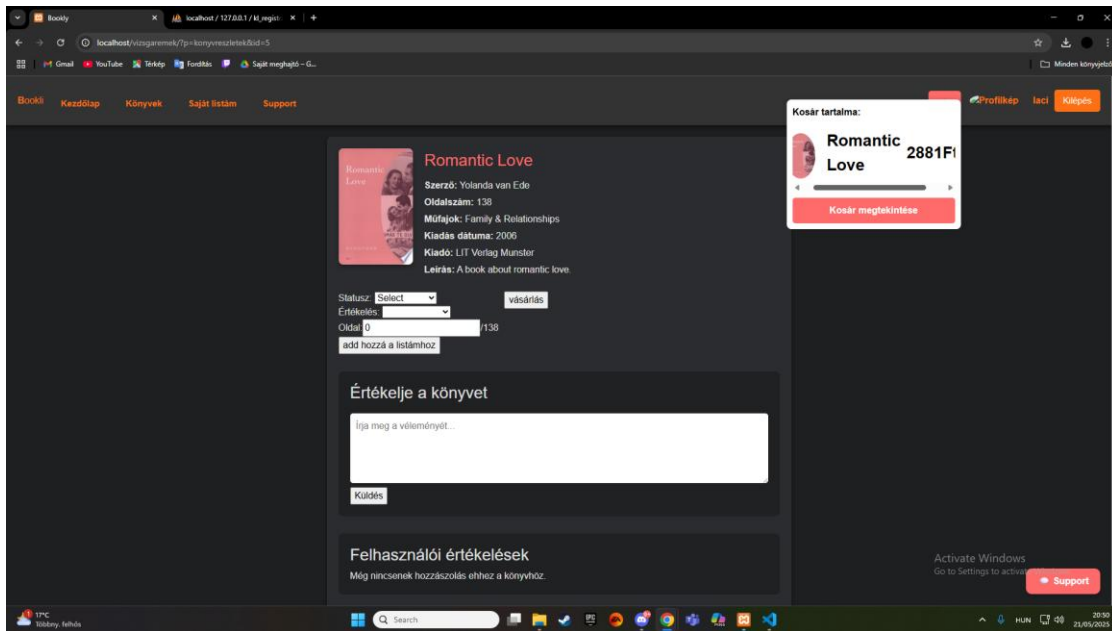




A főoldalon és a 'Könyvek' menüpontban elérhető az összes könyv listája. A felhasználó szöveges keresést alkalmazhat, valamint egy-egy könyv részletes adatlapjára léphet, ahol értékelést is írhat.

Az értékelés mentés után azonnal látható a könyv adatlapján és a felhasználó profilján is.

3.5 Kosár és vásárlás



A könyveket egy kattintással kosárba lehet tenni. A kosár oldal összesíti az elemeket, mennyiségeket és árakat. A vásárlási folyamat során a felhasználó megadja nevét és kártyaadatokat (szimulált űrlapon), majd a vásárlás rögzítésre kerül.

3.6 Admin funkciók

KérdőlapFelhasználókKönyvekKérdésekVélemények

Felhasználói Értékelések

Értékelés ID	Könyv ID	Felhasználó	Email	Szöveg	Dátum	Állapot	Műveletek
11	4	ice	ice@gmail.hu		2025-05-22	szólít	<input type="text"/> <button>Módosítás</button> <button>Törölés</button>
12	4	ice	ice@gmail.hu	asdfsd	2025-05-22	aktív	<input type="text"/> <button>Módosítás</button> <button>Törölés</button>

Activate Windows
Go to Settings to activate Windows.

Próbáld ki ingyen!
Nem kell fizetni!

Search

HUN

00:34

2025/05/22

KérdőlapFelhasználókKönyvekKérdésekVélemények

Felhasználók listája

ID	Felhasználó név	Email	Regisztráció dátuma	Állapot	Komment hozzáadása	Törölés
1	a	a@a.hu	2025-01-16 11:21:36	Admin	<input type="text"/> <button>Hozzáadás</button>	<button>Törölés</button>
2	ice	ice@gmail.hu	2024-12-11 13:51:56	Admin	<input type="text"/> <button>Hozzáadás</button>	<button>Törölés</button>
28	laci	laci@laci.hu	2025-05-22 00:21:12	Aktív	<input type="text"/> <button>Hozzáadás</button>	<button>Törölés</button>

Activate Windows
Go to Settings to activate Windows.

Próbáld ki ingyen!
Nem kell fizetni!

Search

HUN

00:34

2025/05/22

Az adminisztrátori jogosultsággal rendelkező felhasználók elérhetik a következő admin oldalszekciókat:

- Könyvek kezelése (új könyv feltöltése, törlés, módosítás)
- Felhasználók listázása és törlése
- Beérkezett támogatási üzenetek áttekintése és törlése
- Kommentek moderálása

Az admin felületek eléréséhez az adatbázisban 'admin' státuszú felhasználóra van szükség.

4. Összefoglalás

A Bookli alkalmazás fejlesztése során célunk egy könnyen kezelhető, funkciógazdag könyves webalkalmazás létrehozása volt, amely egyaránt lehetőséget ad a felhasználók számára könyvek böngészésére, véleményezésére és vásárlására, valamint az adminisztrátorok részére az adatbázis karbantartására.

A munka során hasznos tapasztalatokra tettünk szert az adatbázis-tervezés, a PHP-szerveroldali feldolgozás, és az interaktív HTML-formok kezelésének terén. Különösen hasznosnak bizonyult a projekt felépítésének strukturálása mappák szerint, valamint az újrafelhasználható komponensek alkalmazása (pl. header, footer, adatkezelők). Úgy véljük, hogy a célokat sikerült elérni, és az elkészült projekt nemcsak a vizsga, hanem a további fejlesztések alapjaként is megállja a helyét.

5. Irodalomjegyzék

- PHP alapok – szerveroldali programozás
<http://infojegyzet.hu/webszerkesztes/php/alapok/>
- HTML űrlapok
<http://infojegyzet.hu/webszerkesztes/html/urlapok/>
- W3Schools dokumentáció
<https://www.w3schools.com/>
- Bootstrap dokumentáció
<https://getbootstrap.com/>
- XAMPP hivatalos oldal
<https://www.apachefriends.org/>
- draw.io – ER-diagram készítő
<https://app.diagrams.net/>