

**Békéscsabai Szakképzési Centrum
Trefort Ágoston Technikum
Szakképző Iskola és Kollégium**

Záródolgozat

Online Könyvkölcsönző oldal

**Készítette: Dézsi János
OKJ szám: 5421305**

2024.

Tartalomjegyzék

Bevezető	1
Rendszerkövetelmények.....	2
Választott téma indoklása.....	2
A program telepítése	3
Adatmodell Leírás	9
Algoritmusok.....	14
Használata	23
Fejlesztés terv	29
Összegzés	31
Források.....	32

Bevezető

Üdv kedves *Felhasználó / Olvasó* !

A bemutatott weboldal amit majd lát/láthat a **Libar-E** nevet kapta azzal a céllal hogy az emberek akik nem mennének el könyvárba vagy nincs idejük akár telefonról vagy otthoni eszközükről is tudjanak a könyvek világába élni.

Mivel az oldal még csak nemrég lett készen így a választék nem a legszélesebb még de a közel jövőben tervezünk a könyvárakkal kapcsolatot létesíteni amivel el tudjuk azt érni hogy bárki bárholnan bármikor kölcsönözhet .

Rajtunk keresztül még akár online is olvashatja vagy egy cím megadása után eljuttatjuk önhöz az említett könyvet

De nagyon fontos hogy a rendszerünket ne használják ki így nekünk is vannak szabályaink amit be kell tartani az *Olvasónak / Felhasználónak* mivel ha ezeket a szabályokat nem tartják be akkor a weboldalról ki lesznek tiltva és pénz büntetésben lesznek részesítve hogy a szabályaink fontosságát és a könyvtárak büntetését ne lehessen elkerülni



Rendszerkövetelmények

a Gép amin sikeresen tesztelve lett a program egy elég gyenge gépen történt meg

Minimális gépigény:

Processzor: Intelcore i-3 8100

Ram: 4 GGB ram

Operációs rendszer: Windows 10 (64-bit)

Tárhely: Minimum 5gb

Programok: Node.js

XAMPP

Visual Studio Code

Ajánlott specifikációk:

Processzor: Intel Core i5-8400

RAM: 8 GB

Operációs rendszer: Windows 10 (64-bites)

Tárhely: 10 GB szabad lemezterület

Programok: Node.js

XAMPP

Visual Studio Code

Tesztelés alapján a programnak lehet van minimálisabb specifikációja de a legtöbb alkalommal ezen a specifikáción elfutott hiba/probléma nélkül sajnos más gépeken nem nagyon volt esélyem tesztelni még így a leg gyakrabban használt specifikációt tudom megadni csak

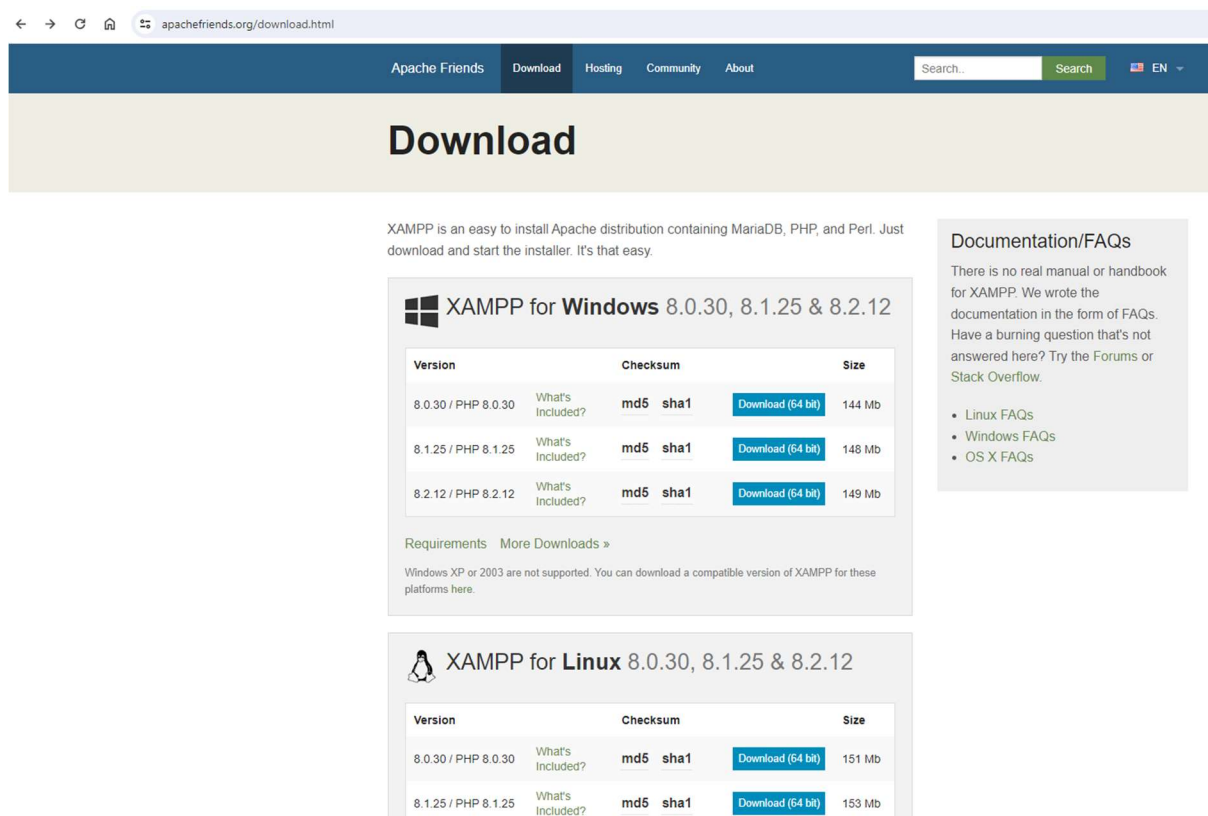
Választott téma indoklása

A könyvek régen is érdekelték és hogyha például nem volt otthon egy könyv így mindig el kellett menni a könyvtárba de mint minden emberben bennem is van lustaság így nem mindig mentem el kölcsönözni a könyvet amit szerettem volna mert lusta voltam elmenniDe így a mostani tapasztalatommal megtudom csinálni azt hogy csináljak egy oldalt amivel lehet online kölcsönözni ezzel meg spórolva az időt és a fáradtságot a könyvtárba elmenetellel

A program telepítése

A működés első részéhez először is kell a program ami miatt működni fog az egész az adatbázis kezelő programunk ez esetben a XAMPP.

1. ábra



← → ↺ 🏠 📄 apachefriends.org/download.html

Apache Friends Download Hosting Community About Search.. Search EN ▾

Download

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy.

XAMPP for Windows 8.0.30, 8.1.25 & 8.2.12

Version	Checksum	Size
8.0.30 / PHP 8.0.30 What's Included?	md5 sha1 Download (64 bit)	144 Mb
8.1.25 / PHP 8.1.25 What's Included?	md5 sha1 Download (64 bit)	148 Mb
8.2.12 / PHP 8.2.12 What's Included?	md5 sha1 Download (64 bit)	149 Mb

[Requirements](#) [More Downloads »](#)

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

XAMPP for Linux 8.0.30, 8.1.25 & 8.2.12

Version	Checksum	Size
8.0.30 / PHP 8.0.30 What's Included?	md5 sha1 Download (64 bit)	151 Mb
8.1.25 / PHP 8.1.25 What's Included?	md5 sha1 Download (64 bit)	153 Mb

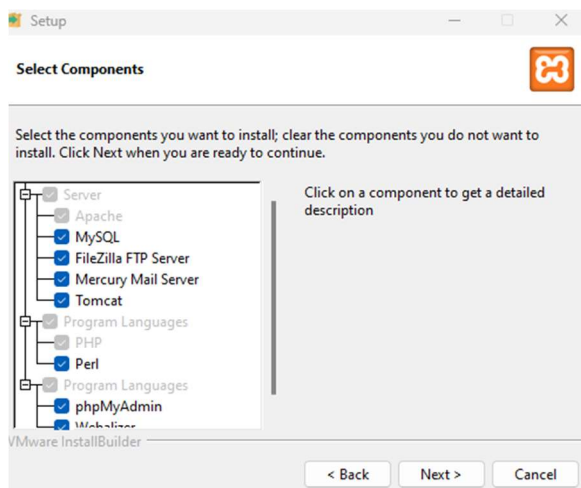
Documentation/FAQs

There is no real manual or handbook for XAMPP. We wrote the documentation in the form of FAQs. Have a burning question that's not answered here? Try the Forums or [Stack Overflow](#).

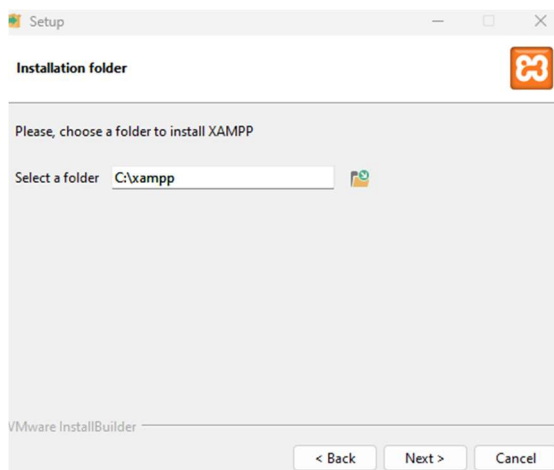
- [Linux FAQs](#)
- [Windows FAQs](#)
- [OS X FAQs](#)

Miután leszedtük a telepítőt fel is kell telepítenünk a programot válasszuk ki a beállításokat amiket szeretnénk a telepítéssel és hogy hova töltsse le ahol foglalja majd a számítógépen a helyet

2. ábra

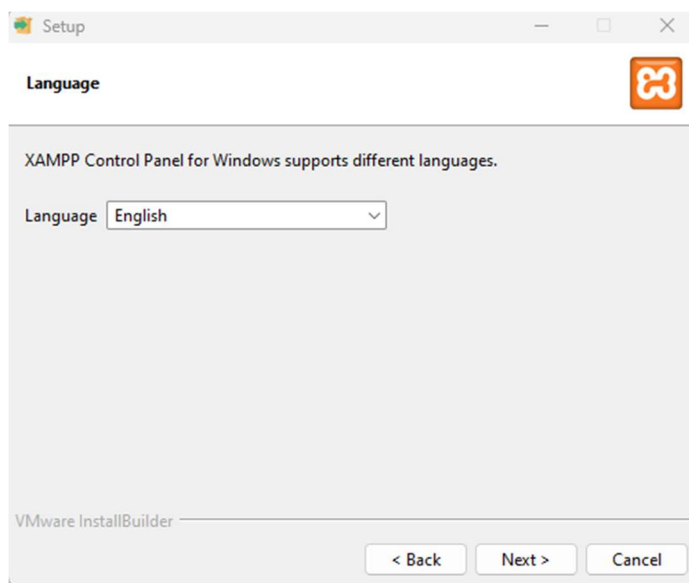


3. ábra



A következő lépés a nyelv kiválasztás lesz az személyes, hogy milyen nyelven akarjuk használni mivel van olyan ember aki angolul jobban preferálja mivel jobban eligazodik de van ember aki inkább maradna az anyanyelvénél

4. ábra



és ezek után a beállítások után a program el is kezdi magát telepíteni a gépre ami után már csak a programot kell elindítani

A másik program ami kelleni fog az a Visual Studio Code amit a <https://code.visualstudio.com/docs/?dv=win64user/> linken tölthetünk le, válasszuk ki hogy melyik platformra akarjuk letölteni amint ez megvan

5. ábra



amint letöltött a telepítő mennyünk végig az általános beállításokon amiket be akarunk állítani mint például telepítési hely azokat állítsuk be és startra kész az alkalmazás

A Visual Studio Code amivel tudjuk futtatni magát a programot és azon belül utána a megfelelő mappába kell letölteni még külön csomagokat amik működtetni fogják az egészet a kódokat itt a telepítőbe

Kódok:

```
npm i
```

```
npm i express
```

```
npm i cors
```

```
npm i nodemon --save-dev
```

```
npm init -y
```

```
npm i prisma --save-dev
```

```
npx prisma
```

```
npx prisma migrate dev --name init
```

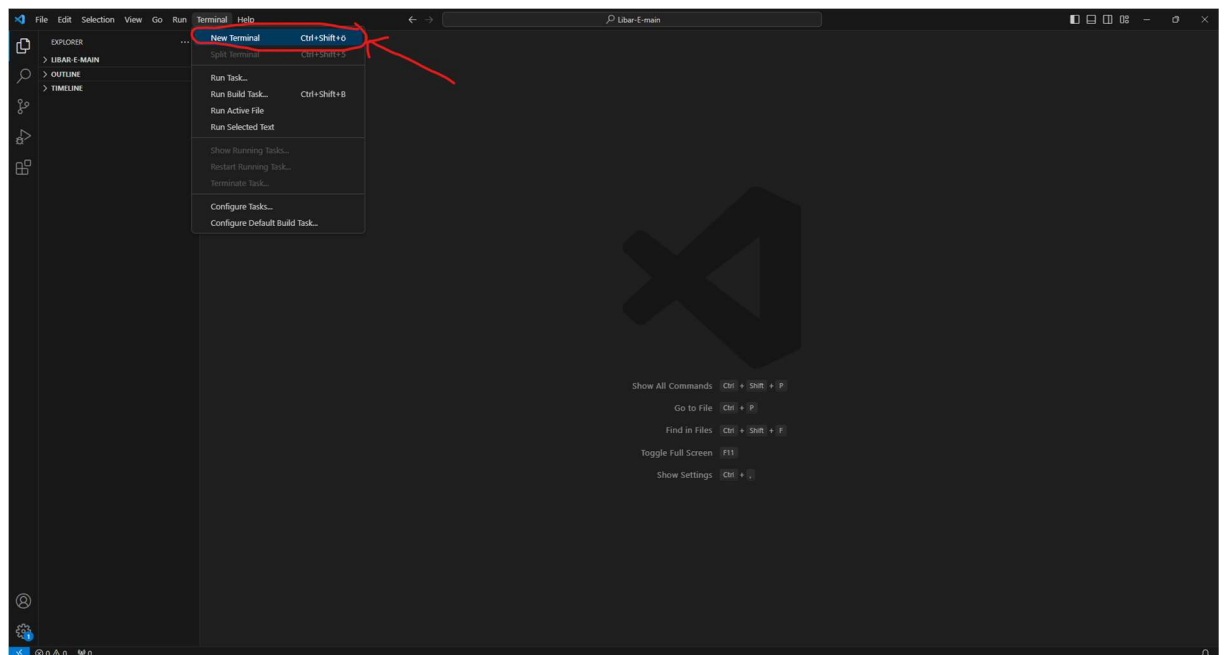
 (ha megvan már az adatbázis tervünk/schema ami már benne van alaptól)

```
npm i @prisma/client
```

az i betű az install rövidítése (install = telepítés/letöltés)

A felül látott parancsokat a Visual Studio Codeon belül kell be írni egy külön helyre ami végre hajtja ezeknek a letöltését és miután azok megvannak van egy működőképes programunk

6. ábra

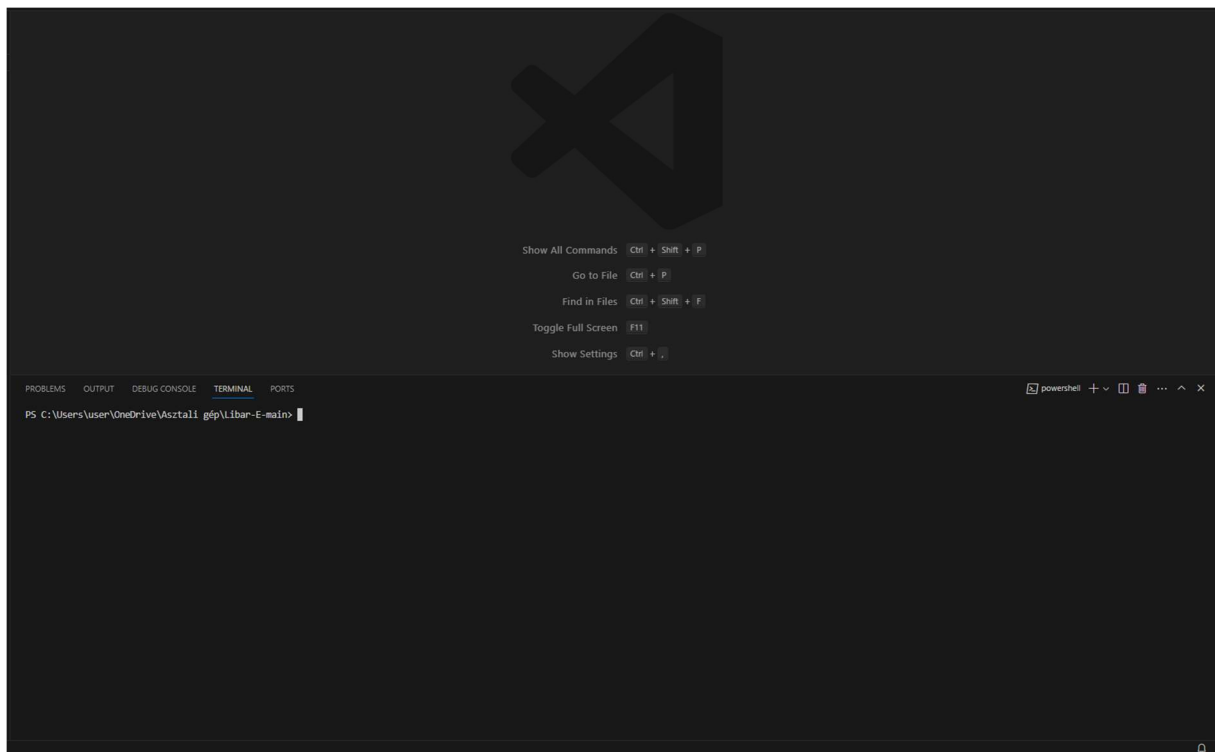


A kiemelt *New Terminal* gomb megnyomásával megnyitjuk ezt a külön helyet

Ez a külön hely a terminál úgy működik mint egy parancssor vagy máshogyan megnevezve olyan mint egy kis számítógép amit használhatunk arra hogy a programunkat futtatni tudjuk és kezelni

A Terminál jó a hibakezelésre is mivel abba segít ha hibába ütközünk hogy ki írja pontosan hogy mi a hiba és hogy hol a hiba és hogy mikor történt a hiba

7. ábra



Miután az végrehajtottuk az előző lépéseket készen állunk arra hogy az előbb megadott parancsokat megadjuk hogy a programhoz való dolgok letöltődjenek és a programot működésre bírjuk és ezeket mind a megadott mappába ahol a program többi része fut

Ha mindent jól telepítettünk és az npm start nevű parancs használatával elindítjuk akkor a megadott mappából az alábbi képen látható szöveget/ üzenetet kéne kapnunk

8. ábra

```
PS C:\Users\diak\Desktop\konyvtar> cd backend
PS C:\Users\diak\Desktop\konyvtar\backend> npm start

> backend@1.0.0 start
> nodemon index.js

[nodemon] 3.0.3
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
listening on 8000
```

.env fájl létrehozása

```
DATABASE_URL="mysql://root:@localhost:3306/konyvtar"
```

```
JWT_SECRET = "szupertitkostitok"
```

Használt eszközök

A program elkészítéséhez használt programozási nyelv az javascript volt és Visual studio Code ba dolgoztam a képeket én készítettem a gépen alpból megtalálható Képmetsző programmal adatbázis-kezelésben xampp meg prisma volt használva és a dokumentáció megíráshoz Microsoft Wordot használtam

A program megírásához használt eszközök amik segítettek működtetni ezt:

Használt programozási nyelvek:

Javascript: A weboldal ezen a programozási nyelvet a Node.js segítségével

SQL: az oldal adatbázis kezelő programozás nyelve amivel az adatokat tudom lekérni és feltölteni

Keretrendszer:

Express.js: a weboldal működését biztosítja ahol a kölcsönzések és a lekérések végre hajtódnak

Prisma: az eszköz ami megegyszerűsítette hogy az adatbázist tudjam kezelni mivel a nem kellett fájlként az adatbázist folyamatosan hozzá kapcsolni

Egyéb Eszközök:

Node.js: Ez az eszköz ami segíti hogy a letöltött eszközök amik működtetik a programot működjenek és hogy egyáltalán le lehessen tölteni azokat az eszközöket

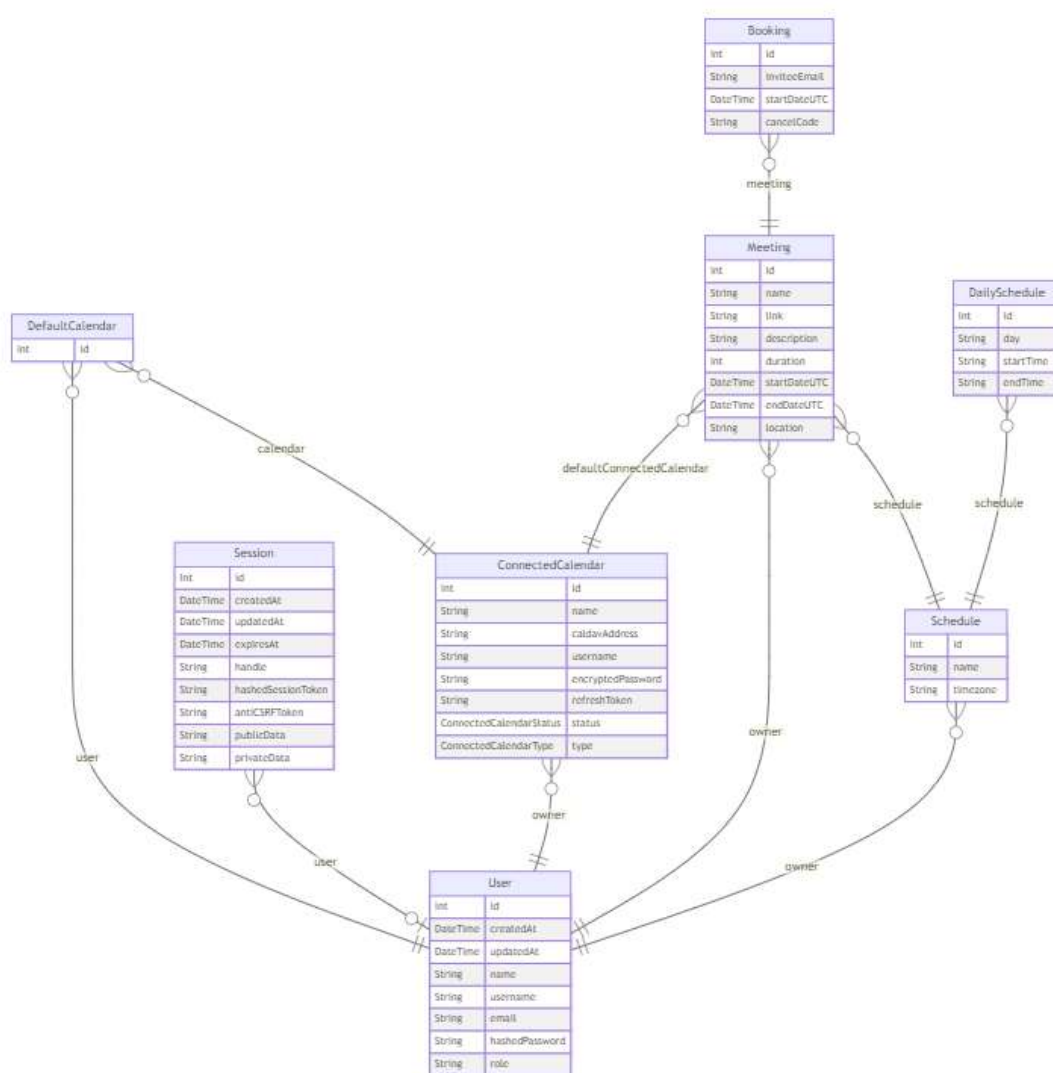
XAMPP: a weboldal adatbázisát kezelő program ami tartalmazza magát az adatbázist amit SQL kódolási nyelvel lehet kezelni is

Visual Studio Code: A weboldal megírásához használt kódolási program

Adatmodell Leírás

Az adatbázis kapcsolatai

9. ábra



A program **kölcsönzőkből** indul ki mivel a *felhasználónak/olvasónak* kell egy profil amivel tudnak kölcsönözni a profil adatai segítenek abba hogy rendszerünk megfelelően működjön mivel a lakcím a pénz büntetéssel és egyéb dolgokkal segít az email cím az ember azonosítója a telefonszámuk is azért kell hogy el tudjuk önöket érni ne csak emailen keresztül hanem azonnal is

a Kölcsönző tábla szerkezete:

Név: A felhasználó teljes neve amit a weboldalon használ

Email cím: A felhasználó által megadott email cím amit elérhetünk és használja a weboldalon

Telefonszám: A felhasználó által megadott telefonszám amin keresztül kapcsolatot létesíthetünk a felhasználóval ha az email címen keresztül nem tudnánk elérni

Lakcím: A felhasználó által megadott lakcím ahova küldhetjük a számlázást ha olyan dolog történne a könyvvel vagy a felhasználó nem hozná vissza vagy más dolog miatt ki kéne küldenünk

Jelszó: a felhasználó által megadott jelszó amit a felhasználó a weboldalon a bejelentkezésre használhat

Ezek mezők céljai

Felhasználói azonosítás

Kapcsolatfelvétel a felhasználóval

Szállítási cím

A kölcsönzések nyomon követése

Személyre szabott élmény biztosítása

A következő a **könyvek** tábla amibe egyértelműen a könyvek vannak eltárolva amiket lehet kölcsönözni

Könyvek tábla:

Cím: A ki listázott könyv címe

Kategória: A ki listázott könyv kategóriája

Író: A ki listázott könyv írója

Kiadási Dátum: A ki listázott könyv kiadási dátuma

A **kölcsönzött** tábla számon tartja hogy a kölcsönzéseket hogy melyik könyvek vannak ki kölcsönözve és meddig mekkora határidővel és mikortól érvényes a kölcsönzése és hogy a könyv vissza lett e már hozva vagy még nem vagy esetleg határidőn átment e. a tábla még tartalmaz két mezőt ami 2 azonosító mező az egyik a kölcsönző táblából a kapcsolat miatt hozzáfér az ottani azonosítóhoz amivel az bejelentkezett illető látja a saját kölcsönzéseit. a másik azonosító az a könyv táblából az azonosító amit szintén egy kapcsolattal érünk el

Kölcsönzés tábla:

Kölcsönzés azonosítója : a kölcsönzést külön tétele a többi könyvtől azzal, hogy saját azonosítót kap ami alapján a felhasználók kölcsönzéseit egyszerűbben lehessen kezelni

Kölcsönző azonosítója (a kölcsönző táblából): a felhasználó azonosítója amivel a kölcsönzéseit lehet nyomon követni

Könyv azonosítója (a könyvek táblából): A kikölcsönzött könyvek nyomon követhetőségét segíti hogy a kikölcsönzött könyveket meg lehessen találni

Kölcsönzés dátuma: A dátum amikor a felhasználó kikölcsönözte a könyvet amit szeretett volna vagy amit a kölcsönző ki fog kölcsönözni

Határidő: A dátum amire a felhasználónak vissza kell szolgáltatnia a könyvet vagy különben a pénzbüntetést kell ki szabnunk és a weboldalt szabályzatát követően kell eljárunk vele

Visszahozatali dátum: A dátum amire a felhasználó a kikölcsönzött könyvet visszaszolgáltatta a megadott időre

Az utolsó tábla a **késés** tábla ami számon tartja hogy ki hányszor és mennyit késett a mivel a késő emberekkel szabály szerint kell eljárni (a weboldal fő oldalán található a szabályzat) hogy az illető ne kövessen el ilyet többször, és hogy a károkat fizesse ki

Késés tábla:

A késés azonosítója: A külön késések nyomon követését segíti, hogy könnyebben meglehessen találni a keresett késéseket

A kölcsönző azonosítója (a kölcsönző táblából): a felhasználó azonosítója ami alapján a felhasználó késéseit lehet könnyeben követni

A könyv azonosítója (a könyvek táblából): A könyv azonosítója hogy melyik könyv nincsen visszaszolgáltatva és melyik felhasználónál van

A kölcsönzés azonosítója (a kölcsönzés táblából): A kölcsönzés azonosítója hogy le lehessen követni a késő kölcsönzéseket és megfelelően szabály szerint lehessen velük eljárni

Késett napok: Számon tartja, hogy mennyi napot késett a felhasználó és hogy mekkora lesz a késési díja a késett napok mennyisége által

Késési Díj: A késett napok alapján kiszabott pénz büntetés amit a lakcímre fogunk kiküldeni egy számlázás formájában

A késés táblába 3 azonosító is van és mivel több az egyhez kapcsolat így elég egy kapcsolatot létrehozni és így a többi azonosító is kapcsolódik így a kölcsönzés azonosítójával ki lehet keresni melyik kölcsönzés volt az amivel késett az illető és hogy melyik könyv volt az

Röviden megmagyarázva a kölcsönző táblából kezdünk, hogy a felhasználónak legyen egy profilja amivel hozzá férhet a weboldalhoz ezek után miután a weboldalhoz kapott hozzáférést amivel a könyvek oldalhoz hozzáfér és keresheti a kívánt könyvet/könyveket, amint a könyvet kiválasztotta a kölcsönzés oldalon tudja a kölcsönzését befejezni ahol amint a kölcsönzése felkerült az adatbázisba a kölcsönzéseim oldalnál a kölcsönzött táblából le lesznek kérve a kölcsönzései azonosító által és ki lesz írva az oldalra

A kapcsolatok ezt segítik röviden leírva

10. ábra

```
generator client {
  provider = "prisma-client-js"
}

datasource db {
  provider = "mysql"
  url      = env("DATABASE_URL")
}

model Kolcsonzo{
  id Int @id @default(autoincrement())
  nev String
  email String @unique
  jelszo String
  elsolatogatas DateTime @default(now())
  lakcim String
  telszam String
  kesesek Int?
  kolcsonzesek Kolcsonzesek[]
  @@map("Kolcsonzo")
}

model Konyvek{
  id Int @id @default(autoincrement())
  cim String
  iro String
  kategoria String
  kiadasDatuma String
  kolcsonzesek Kolcsonzesek[]
  @@map("Konyvek")
}

model Kolcsonzesek{
  id Int @id @default(autoincrement())
  kivitel DateTime @default(now())
  hatarido DateTime
  Visszahozatal DateTime?
  kcs_id Int
  kny_id Int
  kolcsonzo Kolcsonzo @relation(fields: [kcs_id], references: [id])
  konyv Konyvek @relation(fields: [kny_id], references: [id])
  kesett Kesek[]
  @@map("Kolcsonzott")
}

model Kesek{
  id Int @id @default(autoincrement())
  kolcsonzott Kolcsonzesek @relation(fields: [kcsn_id], references: [id])
  kcs_id Int
  kny_id Int
  kcsn_id Int
  kesettNapok Int
  kesesDij Int
  @@map("Kesek")
}
```

a prisma ORM nagyon meg egyszerűsítette dolgomat annyival hiszen nem kellett külön módszereket alkalmaznom amivel sokkal több időbe került volna az alkalmazás megírása mert a fájlok közötti kapcsolat így egyszerűbb és könnyeb kezelni

Algoritmusok

Az algoritmusok avagy külön kódok feladatai:

11. ábra

```
const {PrismaClient} = require("@prisma/client")
const jwt = require('jsonwebtoken');
const argon2 = require('argon2');

const prisma = new PrismaClient();
```

Először is ez a 4 sor segít abban hogy a felhasználónak legyen védett hozzáférése a weboldalhoz és a prisma kliens fölvegye az adatbázisba a regisztrálni kívánó illetőt.

A Prisma Client segít abba hogy az adatbázissal lehessen kapcsolatot létesíteni és fenn tartani azt és adatot vigyünk fel az adatbázisba

a jwt másnevén a jsonwebtoken a token rendszert nyújtja amivel a felhasználó tud biztonságosan igazol hozzáférést kapni az oldalhoz

az argon2 nyújtja a felhasználó kódjának a biztonságát mivel a felhasználó jelszavát az adatbázisban látszólag megváltoztatja, hogy még a készítő se tudja megnézni mi a felhasználó jelszava

a jsonwebtoken a gépén van tárolva így ha valaki az ön profiljával szeretne hozzáférni a weboldalhoz először a felhasználó tokenjét kell elérnie

a tokenje minden belépésnél frissülni fog így ha kijelentkezne nem kell megvárnia amíg a token lejár hanem azonnal egy újat fok kapni a rendszer által


```

const generateToken = (id) => {
  return jwt.sign({id}, process.env.JWT_SECRET, {expiresIn: '12h'});
}

const register = async (req, res) => {

  const {email, jelszo, telszam, lakcim, nev} = req.body;

  if(!email || !jelszo || !telszam || !nev){
    res.json({message: "Hiányos adatok!"});
    return;
  }

  const kolcsonzo = await prisma.kolcsonzo.findUnique({
    where: {
      email: email
    }
  });

  if(kolcsonzo){
    res.json({message: 'Az email cím már foglalt'});
    return;
  }

  const hasheltJelszo = await argon2.hash(jelszo);

  const ujKolcsonzo = await prisma.kolcsonzo.create({
    data: {
      email: email,
      nev: nev,
      telszam: telszam,
      lakcim : lakcim,
      jelszo: hasheltJelszo
    }
  });

  const token = generateToken(ujKolcsonzo.id);
  res.json({token: token});
}

```

A generateToken biztosítja a felhasználó biztonságát mivel ad neki egy külön hozzáférési azonosítót ad neki ami 12 óránként lejár

A register a regisztrációt kezeli ahol egy email csak egyszer szerepelhet és a kódja titkosítva van az argon 2 nevezetű csomagnak köszönhetően

Ha az email már egyszer szerepel az adatbázisban akkor nem fog tudni azzal regisztrálni még egyszer mivel az emailt használjuk ember azonosításra így nem tudják más felhasználóként kiadni magukat és hamis információkat megadni

13. ábra

```
const login = async(req, res) => {  
  
  const {email, jelszo} = req.body;  
  if(!email || !jelszo){  
    res.json({message: 'Hiányzó adatok'});  
  }  
  
  const kolcsonzo = await prisma.kolcsonzo.findUnique({  
    where: {  
      email: email  
    }  
  });  
  
  if(!kolcsonzo) {  
    res.json({message: 'Felhasználó nem található'});  
    return;  
  }  
  
  if(!(await argon2.verify(kolcsonzo.jelszo, jelszo))){  
    res.json({message: 'Nem megfelelő a jelszó!'});  
    return;  
  }  
  
  const token = generateToken(kolcsonzo.id);  
  res.json({  
    message: 'Sikeres bejelentkezés',  
    token: token  
  });  
}
```

A képen a login azaz a bejelentkezés látható ami csak akkor ad ki token-t az illetőnek ha megfelelő jelszót ad meg és a belépéshez csak email és jelszó kell

Ha a login visszadobja hogy nem felel meg az email vagy a jelszava kérem csináljon egy profilt vagy próbáljon meg elérni minket a jelszó problémája iránt

```
const addKönyv = async (req, res) => {
  const {cim, iro, kategoria, kiadasDatuma } = req.body;

  const ujKönyv = await prisma.konyvek.create({
    data: {
      cim: cim,
      iro: iro,
      kategoria: kategoria,
      kiadasDatuma: kiadasDatuma
    }
  });

  res.json({message: "Sikeres termékfelvitel", ujKönyv});
}

const getKönyv = async (req, res) => {
  const konyvek = await prisma.konyvek.findMany({});
  res.json({konyvek});
}
```

Az `ujKönyv` nevű művelet amint elindul és befejezi a műveletet a megadott értékkel ez a könyv fel fog kerülni az adatbázisba és hivatalosan is a weboldalon elérhető lesz megtekintésre és kölcsönzésre is

A `getKönyv` nevezetű művelet felel azért hogy ki lehessen íratni a könyveket mert az adatbázisból lekéri a könyveket és azoknak az adatait amit a ki íratásnál használ az alkalmazás

```

const placeKolcsonzes = (req, res) => {
  const {konyvek} = req.body;
  console.log(req.body);
  const kolcsonzo = req.kolcsonzo;
  console.log(kolcsonzo)
  try{
    konyvek.forEach(async (konyv) => {
      const now = new Date();
      now.setDate(now.getDate()+30)
      for(let i = 0; i < konyv.id; i++) {
        const kolcsonzes = await prisma.kolcsonzesek.create({
          data: {
            kcs_id: Number(kolcsonzo.id),
            kny_id: Number(konyv.id),
            hatarido: new Date(
              now.getFullYear(),
              now.getMonth(),
              now.getDate()
            )
          }
        });
      }
    });
  } catch(err){
    res.json(err.message)
  }
}

```

A kód ami megoldja hogy lehessen kölcsönözni

kölcsönzés jelenleg csak úgy működik hogy az az napi dátumtól kezdődik a kölcsönzés a kiválasztott határidőig amit erősen ajánlok hogy pontosan állítson be és hozza vissza időbe arra amit megadott mert ha nem tudja ezt megtenni a szabályzat alapján kell eljárni

A kölcsönzés még emellé ellátja a magát a művelet a felhasználó és a könyv azonosítójával, hogy a készítő lássa ki kölcsönözte melyik könyvet ki és azért, hogy az adott kölcsönzéseket csak a kölcsönző azaz felhasználó lássa

```

const getkolcsonzes = async (req, res) => {

  const kolcsonzo = req.kolcsonzo;

  const getKonyv = await prisma.kolcsonzesek.findMany({
    where: {
      kcs_id: kolcsonzo.id,
    },
    select: {
      kivitel: true,
      hatarido : true,
      konyv: {
        select: {
          cim: true,
          iro: true,
          kategoria: true,
        }
      }
    }
  });

  res.json({
    kolcsonzesek: getKonyv
  })
}

```

A getkolcsonzes nevezetű művelet azért felel hogy a felhasználó lássa kölcsönzéseit és csakis a SAJÁT kölcsönzéseit és hogy tudja a saját kölcsönzéseinek adatait mint például mit és meddig kölcsönzött

a kód lekéri a felhasználó azonosítójával ellátott rendeltést aki éppen be van jelentkezve és ki írja a kölcsönzéseket

17. ábra

```

const jwt = require('jsonwebtoken');

const protect = async(req, res, next) => {
  let token;
  if(req.headers.authorization && req.headers.authorization.startsWith('Bearer')){
    try{
      token = req.headers.authorization.split(' ')[1];
      const idFromToken = jwt.verify(token, process.env.JWT_SECRET);
      req.kolcsonzo = await prisma.kolcsonzo.findUnique({
        where: {
          id: idFromToken.id
        },
        select: {
          id: true,
          email: true,
          nev: true,
          telszam: true,
        }
      });
      next();
    } catch(err){
      res.status(401).json({message: 'Gondok vannak'})
    }
  }

  if(!token) {
    res.json({message: 'Be kell jelentkezni!'});
  }
}

```

a protect művelet biztosítja hogy a többi művelet ami az oldalon megy mint például maga a kölcsönzés és a kölcsönzések megtekintése csak a protectnek megfelelő felhasználók férhetnek hozzá ez esetben akiknek van tokenje és felhasználónak kellett megadni a emailt, nevet és telefonszámot hogy a rendszer adjon neki egy tokent amit használhat a weboldalon

18. ábra

```

1  const express = require('express');
2  const router = express.Router();
3  const {protect} = require('../middlewares/authMiddleware');
4
5
6  const {register, login ,getKolcsonzo} = require('../controllers/kolcsonzoController');
7
8  router.post('/register', register);
9  router.post('/login', login);
10 router.get('/me', protect,getKolcsonzo )
11
12 module.exports = router;

```

A képen látható kód biztosítja azt, hogy a linkek működjenek és a weboldal hozzá tudjon férni a kölcsönzőkkel/felhasználókkal kapcsolatos műveleteket ezek a belépést és a regisztrációt biztosítják

19. ábra

```
router.get('/all', protect, getKönyv);
router.post('/add', protect, addKönyv);
router.get('/:id', getSpecifiedKönyvId);
router.get('/:kategoria', getSpecifiedKönyvKategoria);
router.get('/:iro', getSpecifiedKönyvIro);
router.get('/:cim', getSpecifiedKönyvCim);
```

A kód amit látni lehet a képen a könyvek kezelését biztosítja azaz ki íratását hozzáadását és a könyvek tulajdonságainak alapján való keresést

A /:iro azt biztosítja, hogy a felhasználó tudjon az alapján keresni és válogatni a könyvek közül

A /:kategoria biztosítja ,hogy a felhasználó tudjon kategória alapján keresni és válogatni a könyvek közül

A /:cim biztosítja, hogy a felhasználó tudjon konkrét könyv címeket keresni és úgy válogatni a könyvek közül

20. ábra

```
const express = require('express');
const router = express.Router();
const {protect} = require('../middlewares/authMiddleware');

const {placeKolcsonzes, getkolcsonzes, getKolcsonzesId, getElerhetoKonyvek, putSingleKolcsonzes} = require('../controllers/kolcsonzesekController')

router.post('/order', protect, placeKolcsonzes)
router.post('/singleOrder', protect, putSingleKolcsonzes)
router.get('/kolcsonzeseim', protect, getkolcsonzes)
router.get('/elerheto', getElerhetoKonyvek)

module.exports = router;
```

Az itteni kód miatt lehet az adatbázisba feltölteni a kölcsönzést és lekérni a kölcsönzéseket

A /order ami a kölcsönzés gomb megnyomásánál lefut és végrehajtja a kölcsönzést és emiatt kerülnek fel a kölcsönzések az adatbázisba

A /kolcsonzeseim akkor fut le amikor a saját kölcsönzések oldalra átmegyünk és ki kell írni a weboldalra a felhasználó kölcsönzéseit

21. ábra

```
const express = require('express');
const cors = require('cors');

const app = express();
const PORT = 8000;

app.use(express.json());
app.use(cors());
app.use('/kolcsonzesek', require('../backend/routes/kolcsonzesekRoutes'))
app.use('/kolcsonzo', require('../backend/routes/kolcsonzoRoutes'))
app.use('/konyv', require('../backend/routes/konyvekRoutes'))

app.listen(PORT, () => {
  console.log(`listening on ${PORT}`);
});

app.get('/', (req, res) => {
  res.json({message: 'Hello'});
});
```

Ez pedig a központja az oldalnak ahonnan eligazodik a weboldal

22. ábra

```
window.onload = async () => {
  const token = JSON.parse(localStorage.getItem("userToken"));

  if (!token || token === undefined) {
    window.location.href = "login.html";
  }

  const response = await fetch('http://localhost:8000/kolcsonzo/me', {
    headers: {
      'Content-Type': 'application/json',
      'Authorization': 'Bearer ${token}'
    }
  })
  const data = await response.json();
  username.textContent = data.nev
}

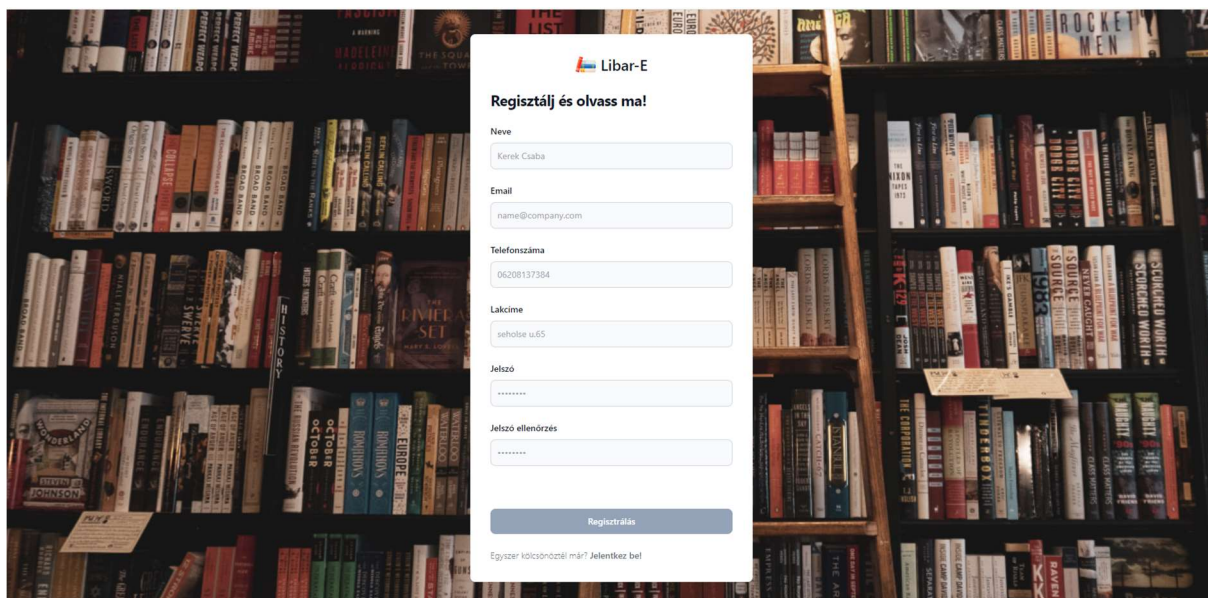
logout.addEventListener('click', () => {
  localStorage.removeItem("userToken");
  window.location.href = "login.html";
});
```

a kód működteti a kijelentkezés gombot ami megnyomásával a felhasználó kijelentkezik a profiljából és a tokenje addig ki lesz törölve

Használata

Miután sikeresen telepítetted és beállítottad a programot, könnyedén hozzáférhetsz a könyvkölcsönző weboldalhoz, hogy élvezd az online könyvkölcsönzés kényelmét és előnyeit.

23. ábra



The image shows a registration form for 'Libar-E' overlaid on a background of bookshelves filled with books. The form is titled 'Regisztrálj és olvasd ma!' and contains the following fields: 'Neve' (Name) with the example 'Kerek Csaba', 'Email' with 'name@company.com', 'Telefonszáma' (Phone number) with '06208137384', 'Lakcíme' (Address) with 'seholse u.65', 'Jelszó' (Password) with masked characters, and 'Jelszó ellenőrzés' (Password confirmation) with masked characters. A 'Regisztrálás' (Registration) button is at the bottom. Below the button, there is a link: 'Egyszer kölcsönözni már? Jelentkez bel'.

Az oldal teljes eléréséhez először is regisztrálni kell egy fiókot. A regisztráció egy egyszerű folyamat, amelyhez csak pár adat megadása szükséges. Ezután regisztrálással végzett és el is kezdheti használni a weboldalt

Név:

Elsőnek is adja meg a teljes nevét. A név megadása azért fontos, hogy a weboldal azonosítani tudja önt, és szükség esetén meg próbálják elérni önnel a kapcsolatot. A név megadásakor ügyeljen arra, hogy a pontos és hivatalos nevét adja meg, hogy a könyvek kölcsönzésére és visszaküldésére azonosítani tudják. ha a későbbiekben változik a neve, frissítse a profilját a weboldalon, hogy a legfrissebb adatokkal rendelkezzenek.

Email cím:

Adja meg az email címét. Az email címre fogja a weboldal küldeni a jelszó emlékeztetőt, és ezen a címen fogja a weboldal értesíteni önt a kölcsönzésekkel kapcsolatos fontos információkról. menjen biztosra azzal, hogy egy érvényes és aktívan használt email

címet ad meg, hogy biztosan megkapja a szükséges üzeneteket. Ha a későbbiekben változik az email címe, frissítse a profilját a weboldalon, hogy a legfrissebb adatokkal rendelkezzenek.

Telefonszám:

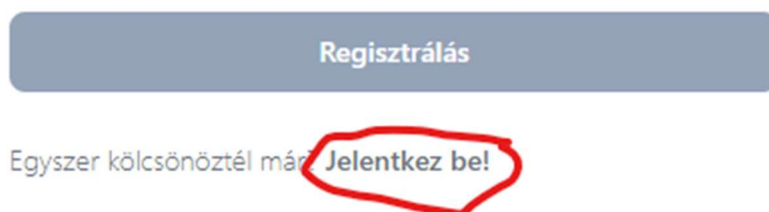
Adja meg a telefonszámát. A telefonszám megadása nem kötelező, de hasznos lehet, ha sürgősen fel kell vennünk Önnel a kapcsolatot. A telefonszám megadása lehetővé teszi számunkra, hogy gyorsabban és hatékonyabban kommunikáljunk Önnel, ha bármilyen probléma merülne fel a kölcsönzéssel kapcsolatban. Ha nem szeretné megadni a telefonszámát, üresen hagyhatja ezt a mezőt.

Lakcím:

Adja meg a lakcímét. A lakcím megadása azért fontos, hogy szükség esetén a számla kiszabás esetén azonosítani tudjuk a címet. A lakcím megadásakor ügyeljen arra, hogy a pontos és hivatalos címét adja meg, hogy a könyvek biztosan megérkezzenek Önhöz. Ha a későbbiekben változik a lakcíme, frissítse a profilját a weboldalon, hogy a legfrissebb adatokkal rendelkezzenek.

Ha már van profilja

24. ábra



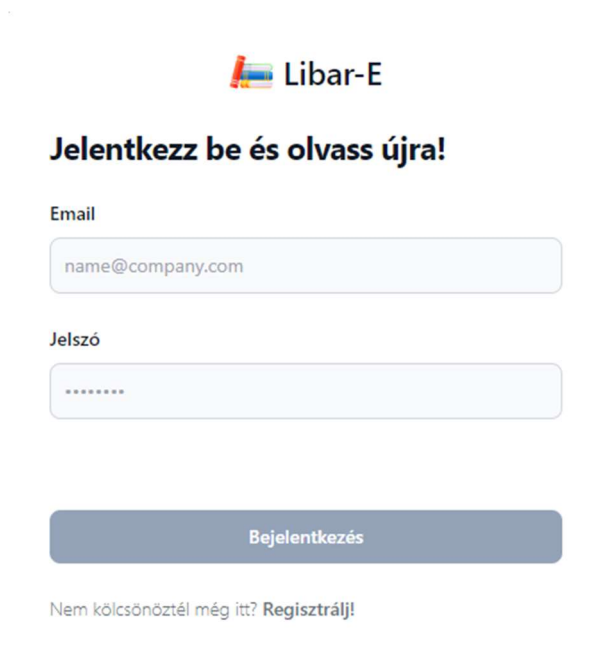
az alábbi linkre nyomjon rá és átvizsi a belépés oldalára

A bejelentkezésnél csak az emailjét és a jelszavát kérjük hogy biztosítani tudjuk hogy ugyan az a felhasználó jelentkezett be aki csinálta a profilt és nem feltörték

így ezzel a módszerrel csak az emailt nem lehet többször felhasználni egy profil csinálásához így nevet és jelszót is lehet többször felhasználni hogy az egyszerű jelszavak a

minden napi élet kedvéért ne legyenek foglalva hisz nem minden ember fogja felírni magának a jelszót vagy megjegyezni a nagyon hosszúakat

25. ábra



A bejelentkezésnél pedig csak az email meg a jelszó kell de ha nem lenne profilja akkor ugyanott ahol előbb található volt a gomb van megint egy gomb ami átvisz a regisztrációs oldalra

Amint regisztrál az adat el lesz tárolva az adatbázisba de ha csak belépett akkor kap egy szimpla tokent és élvezheti a weboldalt

Amint sikerült belépnie/regisztrálnia azonnal a fő oldalra fogja dobni



A fő oldalon lehet látni a szabályzatot ami amit erősen ajánlok elolvasni mivel egy fontos része a weboldalnak és az alapján működik a weboldal is

a szabályok:

Késés: Késés esetén az olvasó kap 1 jelzést amit a profilja alatt láthat és pénzbüntetést szabunk ki 5 jelzés után a felhasználót kitiltjuk

Könyvek: A könyvek nem megfelelő állapotba visszajuttatása pénz büntetést és 1 jelzést von maga után ha a könyv egyáltalán nem jelenik meg nálunk egy bizonyos idő után súlyosabb pénz büntetés fogja érni és teljes kitiltás

Információ: Téves Információ esetén csak jelezzen az nekünk és segítettünk önnek ennek a javításával de ha téves információt használ rendelésre és nem tervezi változtatni akkor kitiltásban részesítjük

Amint a könyveknél sikerült kiválasztania melyik könyvet szeretné ki kölcsönözni a kölcsönzés gomb megnyomásával átmegy arra az oldalra ahol teljesen befejezheti a kölcsönzést

27. ábra

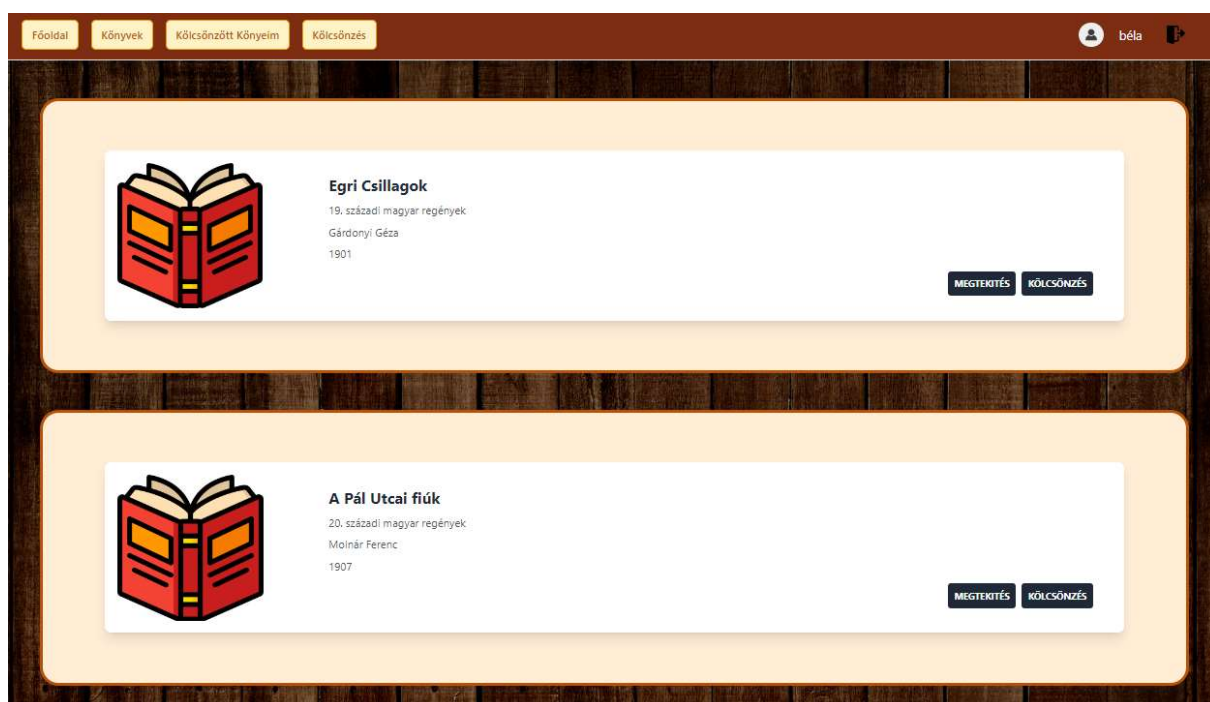


A navigációs sávon lehet eligazodni, hogy mit szeretne. Megnézni a könyveket vagy a éppen mit kölcsönözne vagy a saját kölcsönzéseit?

A weboldal fő oldalán pedig lehet látni emlékeztetés képpen mindig a szabályzatunkat akár angolul akár magyarul

A könyvek oldal ahol lehet kutatni a könyvek között

28. ábra



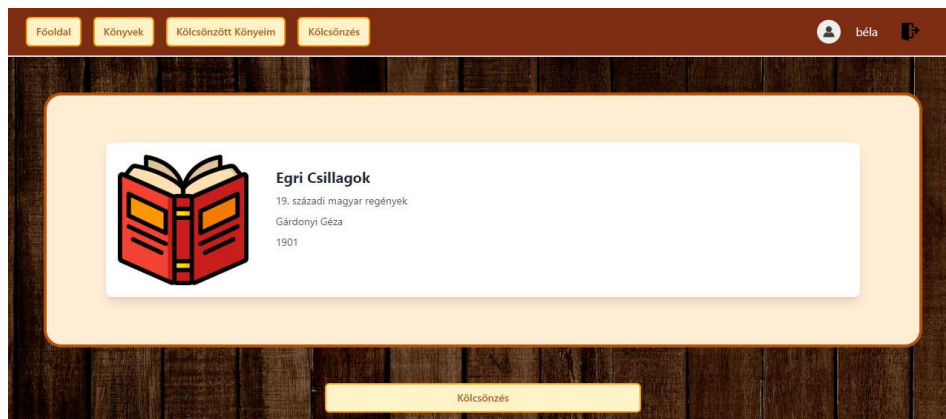
a könyveknek a címe , kategóriája, írója és kiadási dátuma van megjelenítve 2 gomb van a megtekintés és a kölcsönzés

A megtekintés a könyvről egy részletesebb oldalt láthat majd a könyvről leírással és egyéb más információval

A kölcsönzés gomb eltárolja hogy melyik könyvet szeretné kikölcsönözni amit a kölcsönzés oldalnál látni

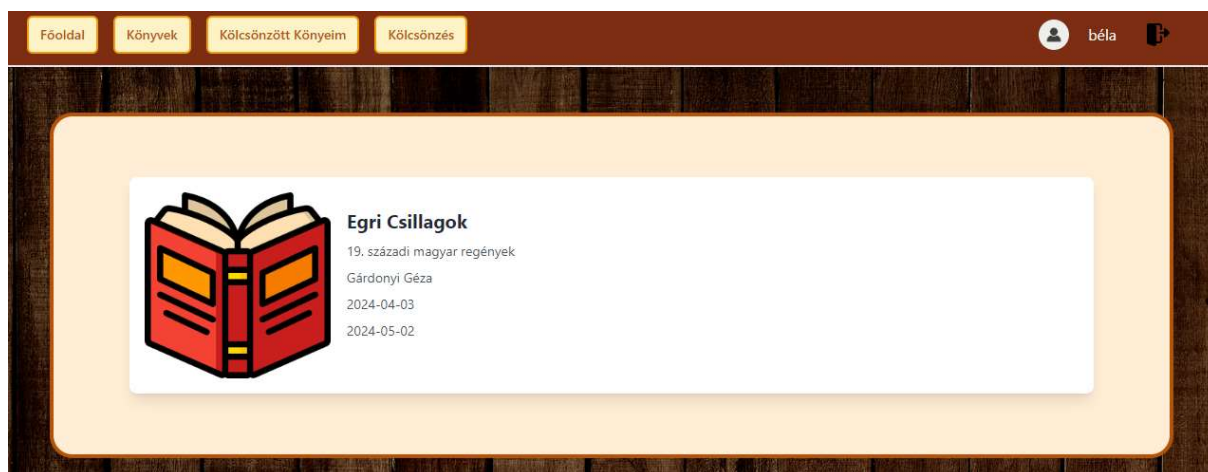
A kölcsönzés oldal ahol egyértelműen kölcsönözhet

29. ábra



A kölcsönzés oldalon a kiválasztott könyvet/könyveket kikölcsönözhet

30. ábra



És végül a kikölcsönzött könyvek ahol a felhasználó láthatja a saját kikölcsönzött könyveit és hogy mettől meddig vannak kölcsönözve

a határidő fontos pontja még mindig a szabályok miatt

Fejlesztés terv

A Késes megoldása elsőnek mivel nem volt időm sajnálatos módon rá a profil megnézés a könyveknek rendesen képek a könyveknek leírás vagy megnézni a könyveket

A profil megtekintést és magát a szabály rendszert nem sikerült mivel szintúgy idő miatt megoldani és egyéb kódokat nem tudtam rendesen hozzáadni sajnos de megvannak csinálva

Felhasználói élmény javítása: A weboldal használatának egyszerűsítése és élvezhetőbbé tétele.

Kezelhetőség bővítése: Új funkciók hozzáadása a weboldalhoz, amelyek javítják a felhasználók számára elérhető szolgáltatásokat.

Teljesítmény optimalizálása: A weboldal betöltési sebességének és teljesítményének javítása.

Biztonság és megbízhatóság: A weboldal biztonságának és megbízhatóságának biztosítása és fejlesztése hogy, a felhasználók az adataikat biztonságba érezzék.

A weboldal designjának frissítése modern és vonzó megjelenésűre.

A navigáció egyszerűsítése.

A keresőfunkció hozzátétele hogy a felhasználók megtalálhassák a keresett könyveket.

Profilkezelés: Lehetőséget biztosítunk a felhasználóknak a profiljuk szerkesztésére, a jelszó megváltoztatására, a kölcsönzési előzmények megtekintésére, a kívánságlisták kezelésére, és az értesítési beállítások módosítására.

Könyvek ki íratása: Bővített szűrési és rendezési lehetőségek a könyvek listájában.

Könyvinformációk: Részletesebb információk a könyvekről, beleértve a tartalomjegyzéket, a bejegyzéseket, a kapcsolódó könyveket, a szerzői információkat, a könyv értékeléseit, a könyv ismertetőjét, a könyv díjait és elismeréseit,

Kölcsönzés: Egyszerűsített kölcsönzési folyamat,

Kölcsönzési emlékeztetők: Emlékeztetők küldése a felhasználóknak a kölcsönzés lejártá előtt.

Kölcsönzés hosszabbítása: Lehetőség a kölcsönzés manuális vagy automatikus hosszabbítására.

Könyvek előjegyzése: Lehetőség a felhasználóknak, hogy előjegyezzenek egy könyvet, ha az kölcsönzésre van kiadva.

Könyvek véleményezése: Lehetőség a felhasználóknak, hogy véleményt nyilvánítsanak a kölcsönzött könyvekről.

Könyvek értékelése: Lehetőség a felhasználóknak, hogy értékeljék a kölcsönzött könyveket.

Virtuális könyvtár: Lehetőség a felhasználóknak, hogy e-könyveket és hangoskönyveket kölcsönözzenek online.

Közösségi funkciók: Lehetőség a felhasználóknak, hogy könyvekről szóló bejegyzéseket írjanak,

A weboldal betöltési sebességének optimalizálása.

A weboldal HTTPS titkosítással történő védelme.

Adatvédelmi szabályzat és GDPR megfelelés.

Jelszókezelő rendszer bevezetése.

Fiókvédelmi beállítások hozzáadása.

Összegzés

A Libar-E könyvkölcsönző online platformot azzal a céllal hozta, létre, hogy lehetővé tegyem az emberek számára, hogy könnyedén hozzáférjenek a könyvek világához, bármikor és bárhol, a nap 24 órájában. A projekt során hangsúlyt fektettem az egyszerűsége, a kényelemre és a biztonságra, hogy a felhasználók teljes mértékben élvezhessék az online könyvkölcsönzés előnyeit.

Az oldal létrehozásának kezdő ötlete az volt, hogy megkönnyítsük az emberek számára az olvasásra való lehetőséget, anélkül, hogy azoknak el kellene menniük a könyvtárba. A könyvkölcsönző weboldal lehetővé teszi a felhasználók számára, hogy böngésszenek a különböző könyvek között, kölcsönözzenek, és élvezzék az olvasás örömét otthonról vagy akár útközben is.

A projekt során kiemelt figyelmet fordítottam az alkalmazás biztonságára és a felhasználói élményre. A regisztráció és bejelentkezés folyamata biztonságos és egyszerű, hogy a felhasználók mindig védettek legyenek az adatvédelem terén. Emellett a felhasználói profil lehetőséget biztosít személyre szabott könyvkölcsönzési preferenciák beállítására, és a korábbi kölcsönzések nyomon követésére.

Az oldal kialakítása és navigációja intuitív és felhasználóbarát, hogy a felhasználók könnyen megtalálják a keresett könyveket és funkciókat. A főoldalon található ajánlások és legnépszerűbb könyvek listája segít a felhasználóknak az új olvasnivalók felfedezésében, míg a kategóriák és címek szerinti rendezés lehetővé teszi a pontosabb keresést.

Az online könyvkölcsönző weboldal jelenleg alapvető funkcionalitással rendelkezik, azonban a jövőben terveink között szerepel további funkciók és fejlesztések bevezetése. Ide tartozik a kölcsönzések pontosabb nyomon követése, a könyvek részletes leírásainak hozzáadása, valamint a profilok bővítése és a szabályrendszer továbbfejlesztése.

Az eddigi tapasztalatok alapján úgy véljük, hogy a Libar-E könyvkölcsönző weboldal jelentős előrelépést jelent az olvasás és könyvkölcsönzés terén, és reményeink szerint hozzájárul az emberek kényelmesebb és élvezetesebb olvasási élményéhez.

Források

1. Express.js hivatalos: <https://expressjs.com/>
2. CORS (Cross-Origin Resource Sharing): <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>
3. Prisma ORM (Object-Relational Mapping): <https://www.prisma.io/>
4. Nodemon: <https://nodemon.io/>
5. Argon2: <https://www.npmjs.com/package/argon2>
6. JSON Web Token (JWT): <https://jwt.io/>
7. Tailwind CSS: <https://tailwindcss.com/>

Ezek a források hozzájárultak a Libar-E könyvkölcsönző weboldal fejlesztéséhez és a dokumentáció elkészítéséhez, biztosítva a szükséges információkat és eszközöket a projekt megvalósításához.

Plágium-nyilatkozat

Alulírott (név)
..... (szül.hely)
..... (szül.idő)
..... (anyja neve)

jelen nyilatkozat aláírásával kijelentem, hogy a

..... című záródolgozat

(a továbbiakban: dolgozat) önálló munkám, a dolgozat készítése során betartottam a szerzői jogról szóló 1999. évi LXXVI. tv. szabályait, valamint a Békéscsabai Szakképzési Centrum által előírt, a dolgozat készítésére vonatkozó szabályokat.

Tudomásul veszem, hogy a dolgozat esetén plágiumnak/szerzői jogsértésnek számít:

- szó szerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más szerző publikált gondolatainak saját gondolatként való feltüntetése

Kijelentem továbbá, hogy a dolgozat készítése során az önálló munka kitétel tekintetében a konzulenszt, illetve a feladatot kiadó oktatót nem tévesztettem meg.

Jelen nyilatkozat aláírásával tudomásul veszem, hogy amennyiben bizonyítható, hogy a dolgozatot nem magam készítettem vagy a dolgozattal kapcsolatban szerzői jogsértés ténye merül fel, a Békéscsabai Szakképzési Centrum a dolgozatot elégtelennek minősíti.

Békéscsaba, 20hónap

Tanuló aláírása