Projektmunka Dokumentálása

Mozi helyfoglaló webalkalmazás

Premozi

Premontrei Szakgimnázium és Technikum

Keszthely

2025

Szoftverfejlesztő és -tesztelő szak

5-0613-12-03

Készítette:

Bató Bence, Magyarósi Bálint

Tartalom

[1. Bevezetés 3](#_Toc194481452)

[2. Fontos programok 3](#_Toc194481453)

[3. Technológiai felépítés 4](#_Toc194481454)

[4. NuGet csomagok 4](#_Toc194481455)

[5. Adatbázis szerkezete 4](#_Toc194481456)

[Adatbázis modell 5](#_Toc194481457)

[users 5](#_Toc194481458)

[film 6](#_Toc194481459)

[vetites 7](#_Toc194481460)

[terem 7](#_Toc194481461)

[rendeles 7](#_Toc194481462)

[Kapcsolatok 8](#_Toc194481463)

[6. REST API funkciók 8](#_Toc194481464)

[7. Weboldal működése 9](#_Toc194481465)

[8. NUnit teszt 9](#_Toc194481466)

[még nincs nunit teszt 9](#_Toc194481467)

[9. Fejlesztési javaslatok 9](#_Toc194481468)

//A projektünk a projekt előírásai szerint projekt formátumban, projekten belül egy projekttel készül, a projekttársammal.

# Bevezetés

A szakmai vizsgaprojektünk egy mozijegy-foglaló rendszer, amelynek megvalósítását azért választottuk, mert csapatunk egy korábbi iskolai projekt során már dolgozott mozi témájú alkalmazáson, így ezt az ismeretet tovább tudtuk fejleszteni. A projekt célja egy olyan online rendszer létrehozása, amely lehetővé teszi a felhasználók számára, hogy kényelmesen és gyorsan lefoglalják helyeiket a kívánt vetítésekre.

A rendszer fejlesztése során kiemelt figyelmet fordítottunk a hatékony csapatmunkára és a feladatok megfelelő elosztására. Minden csapattag a saját erősségei és tapasztalatai alapján kapott szerepet a projektben, így biztosítva a munka gördülékenységét és a lehető legjobb eredmény elérését. A kommunikáció és az együttműködés kulcsfontosságú volt, rendszeresen tartottunk megbeszéléseket, ahol megosztottuk az előrehaladásunkat, megbeszéltük a felmerülő problémákat, és közösen kerestünk megoldásokat.

A jegyfoglaló rendszer egy webalapú alkalmazásként működik, amely egy MySQL adatbázisra épül. A felhasználók regisztrálhatnak, bejelentkezhetnek, és a rendszer segítségével kiválaszthatják a kívánt vetítést, majd lefoglalhatják helyeiket. Az adatok kezelése és a foglalási folyamat biztosítása egy REST API segítségével történik, amely lehetővé teszi az adatok biztonságos és gyors elérését.

Dokumentációnk részletesen bemutatja a rendszer felépítését, működését és technológiai hátterét. Ismertetjük az alkalmazott fejlesztési eszközöket, az adatbázis szerkezetét, valamint a rendszer főbb funkcióit.

Összességében a projekt nemcsak technológiai kihívást jelentett számunkra, hanem lehetőséget adott arra is, hogy fejlesszük együttműködési készségeinket, megtanuljuk a hatékony munkamegosztás fontosságát, és valós fejlesztői környezetben szerezzünk tapasztalatot. Bízunk benne, hogy a mozijegy-foglaló rendszerünk egy jól működő, felhasználóbarát megoldást kínál a felhasználók számára.

# Fontos programok

* Visual Studio 2022

A programunk a Visual Studio 2022-vel készült

* XAMPP

A XAMPP segítségével oldottuk meg az adatbázishoz való kapcsolódást.

* Discord

A kommunikáció, pl.: feladatelosztás, teendők, megsbeszélések lehetőségének biztosítására szolgált.

* GitHUB

A GitHUB segítségével bárhonnan tudtunk dolgozni bármilyen környezetben. Ez elősegítette az otthoni munka, illetve a suliban termek között lévő vándorlás problémáját.

* ProtonVPN

A ProtonVPN segítségével eltudtunk érni bármilyen segítséget az interneten, ami valamilyen okból a suli neten le volt tiltva. pl.: Stackoverflow, Discord, és néha a w3schools

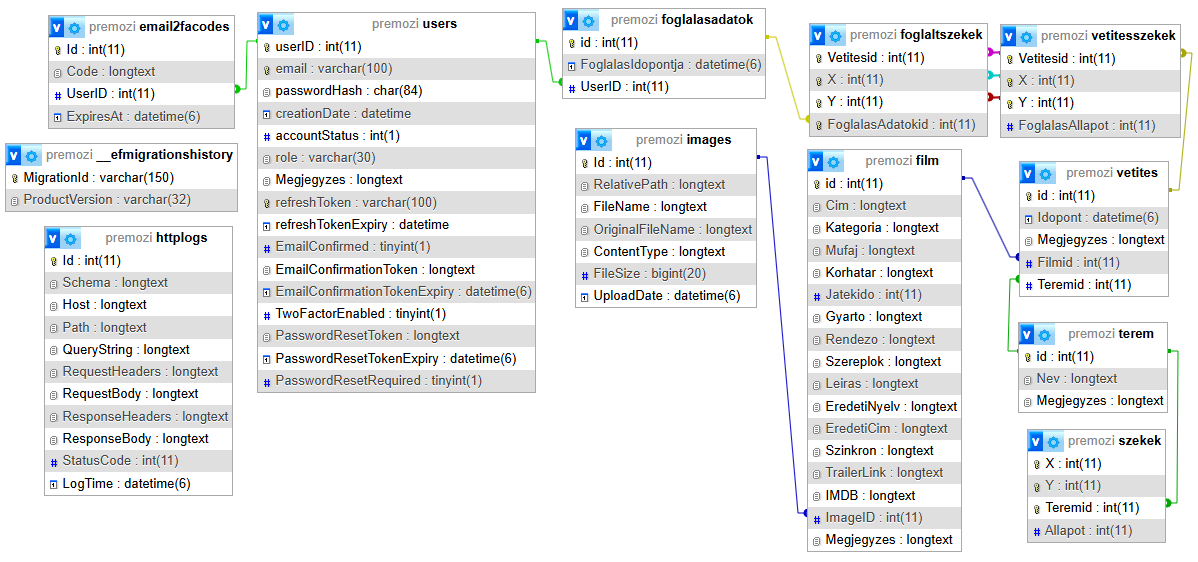
# Technológiai felépítés

* **Backend:** ASP.NET
* **Frontend:** React + Vite
* **Adatbázis:** MySQL
* **Adatbázis kommunikáció:** REST API

# NuGet csomagok

# Adatbázis szerkezete

Az adatbázis a következő táblákból áll:

* **users**: Felhasználók adatait tárolja.
* **filmek**: A mozi filmjeinek adatait tartalmazza.
* **vetites**: A vetítési időpontokat tárolja.
* **terem**: A mozitermek adatait tartalmazza.
* **rendeles**: A foglalások adatait kezeli. //Később átnevezésre kerül
* **login\_session\_log**: A bejelentkezési előzményeket tárolja.
* **action\_log**: A rendszer műveleteit rögzíti.

## Adatbázis modell

Az adatbázisunk egy mozi helyfoglaló rendszer működését szolgálja, melyben felhasználók tudnak helyet foglalni vetítésekre.

## Film tábla

film: A moziban vetített filmek adatait tárolja

id: int(11), A film egyedi azonosítója (elsődleges kulcs)

Cim: longtext, A film címe

Kategoria: longtext, A film kategóriája

Mufaj: longtext, A film műfaja

Korhatar: longtext, A korhatár besorolás

Jatekido: int(11), A film hossza percekben

Gyarto: longtext, A film gyártója

Rendezo: longtext, A rendező neve

Szereplok: longtext, Főszereplők listája

Leiras: longtext, A film rövid leírása

EredetiNyelv: longtext, Az eredeti nyelv

EredetiCim: longtext, Az eredeti cím

Szinkron: longtext, Szinkronizált-e

TrailerLink: longtext, A trailer linkje

IMDB: longtext, IMDB értékelés vagy link

ImageID: int(11), A filmhez tartozó kép azonosítója (külső kulcs az images táblára)

Megjegyzes: longtext, Egyéb megjegyzések

## Terem tábla

terem: A mozi termeinek adatait tárolja

id: int(11), A terem egyedi azonosítója (elsődleges kulcs)

Nev: longtext, A terem neve

Megjegyzes: longtext, Egyéb megjegyzések a teremről

## Vetítés tábla

vetites: A filmek vetítéseivel kapcsolatos adatokat tárolja

id: int(11), A vetítés egyedi azonosítója (elsődleges kulcs)

Idopont: datetime(6), A vetítés pontos időpontja

Megjegyzes: longtext, Egyéb megjegyzések

Filmid: int(11), A vetített film azonosítója (külső kulcs a film táblára)

Teremid: int(11), A terem azonosítója ahol a vetítés lesz (külső kulcs a terem táblára)

## VetítésSzékek tábla

vetitesszekek: A vetítéshez tartozó székek állapotát tárolja

Vetitesid: int(11), A vetítés azonosítója (összetett elsődleges kulcs része, külső kulcs a vetites táblára)

X: int(11), A szék X koordinátája a teremben (összetett elsődleges kulcs része)

Y: int(11), A szék Y koordinátája a teremben (összetett elsődleges kulcs része)

FoglalasAllapot: int(11), A szék állapota (pl. szabad, foglalt)

## Székek tábla

szekek: A termek székeinek alapkonfigurációját tárolja

X: int(11), A szék X koordinátája (összetett elsődleges kulcs része)

Y: int(11), A szék Y koordinátája (összetett elsődleges kulcs része)

Teremid: int(11), A terem azonosítója (összetett elsődleges kulcs része, külső kulcs a terem táblára)

Allapot: int(11), A szék alapállapota (pl. létező, nem létező)

## FoglalásAdatok tábla

foglalasadatok: A felhasználói foglalások fő adatait tárolja

id: int(11), A foglalás egyedi azonosítója (elsődleges kulcs)

FoglalasIdopontja: datetime(6), A foglalás időpontja

UserID: int(11), A felhasználó azonosítója aki a foglalást végzi (külső kulcs a users táblára)

FoglaltSzékek tábla

foglaltszekek: A foglalásokhoz tartozó konkrét székeket tárolja

Vetitesid: int(11), A vetítés azonosítója (összetett elsődleges kulcs része, külső kulcs a vetitesszekek táblára)

X: int(11), A szék X koordinátája (összetett elsődleges kulcs része)

Y: int(11), A szék Y koordinátája (összetett elsődleges kulcs része)

FoglalasAdatokid: int(11), A foglalás azonosítója (összetett elsődleges kulcs része, külső kulcs a foglalasadatok táblára)

## Users tábla (felhasználók)

users: A rendszer felhasználóinak adatait tárolja

userID: int(11), Felhasználó egyedi azonosítója (elsődleges kulcs)

email: varchar(100), Felhasználó email címe (egyedi)

passwordHash: char(84), Titkosított jelszó

creationDate: datetime, Regisztráció dátuma

accountStatus: int(1), Fiók állapota (pl. aktív, letiltott)

role: varchar(30), Felhasználói szerepkör (pl. admin, user)

Megjegyzes: longtext, Egyéb megjegyzések

refreshToken: varchar(100), Frissítési token (egyedi)

refreshTokenExpiry: datetime, Token lejárati ideje

EmailConfirmed: tinyint(1), Email megerősítve (0/1)

EmailConfirmationToken: longtext, Email megerősítő token

EmailConfirmationTokenExpiry: datetime(6), Token lejárati ideje

TwoFactorEnabled: tinyint(1), Kétlépcsős azonosítás bekapcsolva (0/1)

PasswordResetToken: longtext, Jelszó visszaállító token

PasswordResetTokenExpiry: datetime(6), Token lejárati ideje

PasswordResetRequired: tinyint(1), Kötelező jelszóváltoztatás (0/1)

## Images tábla (képek)

images: A rendszerben tárolt képeket kezeli

Id: int(11), Kép egyedi azonosítója (elsődleges kulcs)

RelativePath: longtext, Kép relatív elérési útja

FileName: longtext, Fájlnév a rendszerben

OriginalFileName: longtext, Eredeti fájlnév

ContentType: longtext, MIME típus (pl. image/jpeg)

FileSize: bigint(20), Fájlméret bájtban

UploadDate: datetime(6), Feltöltés időpontja

## Email2FACodes tábla (kétlépcsős azonosítás kódok)

email2facodes: Kétlépcsős azonosításhoz használt ideiglenes kódok

Id: int(11), Rekord egyedi azonosítója (elsődleges kulcs)

Code: longtext, A generált biztonsági kód

UserID: int(11), Felhasználó azonosító (külső kulcs a users táblára)

ExpiresAt: datetime(6), Kód lejárati ideje

HTTPLogs tábla (naplózás)

httplogs: HTTP kérések és válaszok naplózása

Id: int(11), Naplóbejegyzés azonosítója (elsődleges kulcs)

Schema: longtext, URL séma (http/https)

Host: longtext, Kiszolgáló hosztneve

Path: longtext, Kérés útvonala

QueryString: longtext, Query paraméterek

RequestHeaders: longtext, Kérés fejlécek

RequestBody: longtext, Kérés törzse

ResponseHeaders: longtext, Válasz fejlécek

ResponseBody: longtext, Válasz törzse

StatusCode: int(11), HTTP státuszkód

LogTime: datetime(6), Naplózás időpontja (alapértelmezett: aktuális UTC idő)

## \_\_EFMigrationsHistory tábla (migrációs előzmények)

\_\_efmigrationshistory: Entity Framework migrációs előzmények

MigrationId: varchar(150), Migráció azonosítója (elsődleges kulcs)

ProductVersion: varchar(32), Entity Framework verzió

## Kapcsolatok

users ↔ email2facodes

Kapcsolat típusa: 1:N (egy felhasználónak több kétlépcsős kódja lehet)

users ↔ foglalasadatok

Kapcsolat típusa: 1:N (egy felhasználó több foglalást hozhat létre)

images ↔ film

Kapcsolat típusa: 1:N (egy képhez több film is tartozhat, de egy filmnek csak egy képe van)

film ↔ vetites

Kapcsolat típusa: 1:N (egy filmet több vetítésen is leadhatnak)

terem ↔ vetites

Kapcsolat típusa: 1:N (egy teremben több vetítés is lehet)

terem ↔ szekek

Kapcsolat típusa: 1:N (egy teremhez több szék tartozik)

vetites ↔ vetitesszekek

Kapcsolat típusa: 1:N (egy vetítéshez több székállapot tartozik)

vetites ↔ foglaltszekek

Kapcsolat típusa: 1:N (egy vetítéshez több foglalt szék tartozhat)

foglalasadatok ↔ foglaltszekek

Kapcsolat típusa: 1:N (egy foglaláshoz több szék is tartozhat)

vetitesszekek ↔ foglaltszekek

Kapcsolat típusa: 1:1 (egy vetítés-székpárhoz pontosan egy foglalás-székpár tartozik)

Különleges kapcsolatok:

vetitesszekek ←(foglalasadatok)→ foglaltszekek: M:N kapcsolat van, de ez két 1:N kapcsolatként van megvalósítva a foglalasadatok táblán keresztül.

# REST API funkciók

A rendszer REST API-t használ az adatok kezelésére. Az alábbi fő végpontok érhetők el:

* **Felhasználók kezelése**
  + POST /api/users/login – Bejelentkezés
  + POST /api/users/register – Regisztráció
* **Foglalások kezelése**
  + GET /api/rendeles – Összes foglalás lekérése
  + POST /api/rendeles – Új foglalás létrehozása
  + DELETE /api/rendeles/{id} – Foglalás törlése
* **Vetítések és termek kezelése**
  + GET /api/vetites – Vetítési adatok lekérése
  + GET /api/terem – Mozitermek listázása

# Weboldal működése (majd insert képernyőképek)

A weboldal egy reszponzív felhasználói felületet biztosít, amely az alábbi fő oldalakat tartalmazza:

* Bejelentkezés és regisztráció

Elérés út: https://localhost:60769/account/login

A felhasználó be tud jelentkezni, egy letisztult, bejelnetkezési felülelten.

Ha nincs fiókja, az alsó „Nincs fiókja? Regisztráljon most!” -ra kattintva, át írányítódik a regisztrációs felületre, ahol létrehozhatja fiókját.

* Ha admin fiókkal jelenkezünk be, az admin fiókkal a következők tehetőek:
  + Felhasználók kezelése/modosítása/törlése
  + adatok kezelése/modosítása/törlése
* Helyfoglalás
  + A helyfoglaláshoz válasszunk egy filmet a listán, majd nyomjunk a helyfoglalás gombra. Ez után megjelenik a helyfoglalás felület. A helyfoglaláshoz, válasszunk egy vagy akár több széket, majd kattintsunk a foglalás gombra.
* Vetítési idők és filmek listázása
* Filmmel kapcsolatos adatok megjelenítése
* Helyfoglalás adott vetítésre
* Foglalt helyek megjelenítése

A weboldal az adatokat a REST API-n keresztül kéri le és küldi el az adatbázisba.

# NUnit teszt

# még nincs nunit teszt

# Fejlesztési javaslatok

* A foglalások kezeléséhez szükséges táblát még létre kell hozni az adatbázisban.
* Érdemes lehet adminisztrációs felületet fejleszteni a vetítések és foglalások menedzselésére.
* A felhasználói élmény javítása érdekében értesítési rendszert lehetne beépíteni.
* Az adatbázis normálforma szintjének növelése (igen rád nézünk filmek tábla)
* Foglalás visszavoása/törlése

vázlat

**Projekt Scope**

Létrehozható termék:

* REST API létrehozás

Out of scope:

* Adatbázishoz csatolás

Erőforrások:

* Programozók (3 ember), 1 óra
* Intel Pentium PC

Stakeholderek:

* Baranyai Előd Zsolt, inf. okl. mérnök-tanár

Haszna: Hotel vendégek bejelentkeztetése, szobába elhelyezése

Risk manadgement: Rosszabb lesz mint 4