

BKSZC Pogány Frigyes Technikum

Random Play Store

szoftverfejlesztő és -tesztelő vizsgaremek

2025. május

Készítette:

György Péter Attila

Zauber Boglárka

Budapest, 2025. május

# **Tartalomjegyzék**

# Bevezetés:

* A feladat leírása
* GitHub repository

Fejlesztői dokumentáció:

* Az adatbázis elkészítése
* Fejlesztői környezet
* Backend és frontend

# Felhasználói dokumentáció:

* Weboldal elérése
* Regisztráció és bejelentkezés
* Játék bérlése és visszaküldése
* Vélemény írása

# **Bevezetés**

# **A feladat leírása**

A Random Play bolt játékokat ajánl fel kölcsönzésre. Regisztrált felhasználóként kölcsönözhetsz játékokat, és megoszthatod a tapasztalataidat azzal, hogy véleményt írsz mások számára.

**GitHub repository**

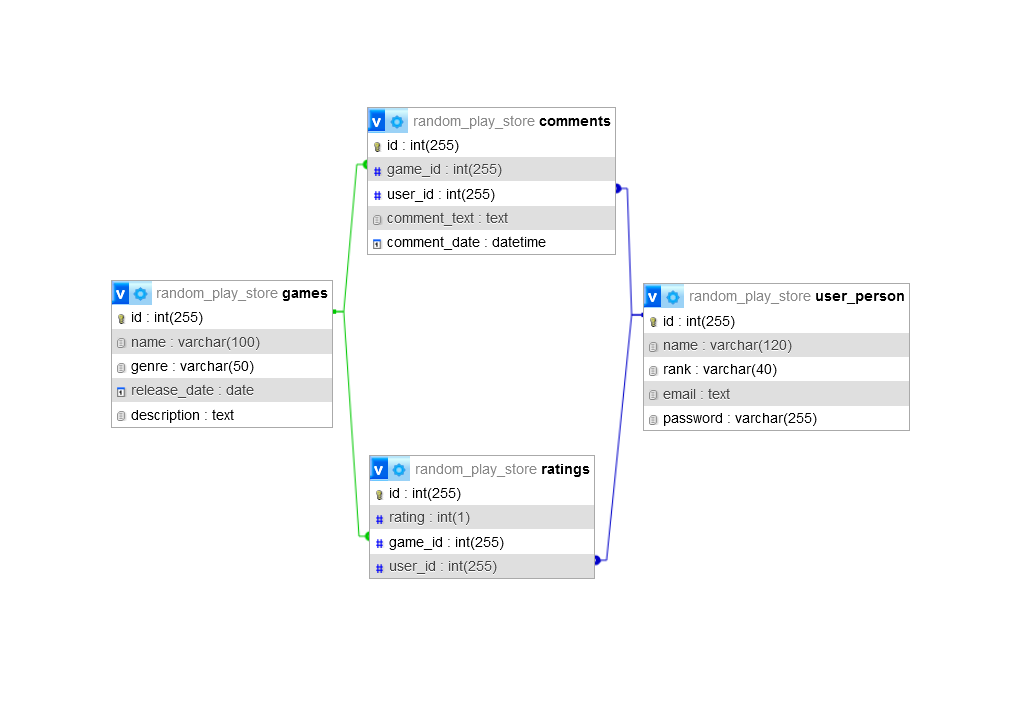
Az alkalmazások kódja ebben a repository-ban érhetők el:  
https://github.com/Zauberbogi/jatek\_katalogus

**Fejlesztői dokumentáció**

**Az adatbázis elkészítése**

Az adatbázist a MySQL adatbázis-kezelővel és a PHPMyAdmin programmal készítettük el.

Az adatokat három táblára bontottuk:

Egy commentet lehet azonosítani a felhasználó azonosítója (id -je) alapján. 

**A games adatai**

Minden játéknak kell egy azonosító (id) ami a elsődleges kulcs lesz. Ezenkívül szükséges van a játék nevére (name), fajtájára (genre),  megjelenése dátum (release\_date) és a leírására (description).

Ennek megfélően a games tábla szerkezete:



Az alkalmazás teszteléséhez az alábbi tesztadatokat vittünk be:  


**A user person tábla**

id: a felhasználó azonosító száma (elsődleges kulcs),

name: a felhasználó neve,

rank: a felhasználó rangja, admin, moderátor és sima felhasználó az opció,

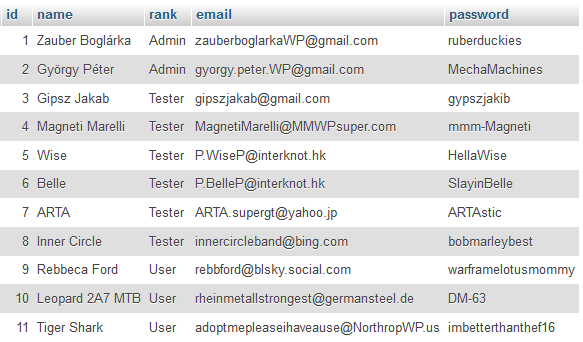
email: a felhasználó emailje,

password: a felhasználó jelszava (hash).

Ennek megfélően a games tábla szerkezete:



Az alkalmazás teszteléséhez az alábbi tesztadatokat vittünk be:



**A comments tábla**

Ez a tábla a commentek adatait tartalmazza:

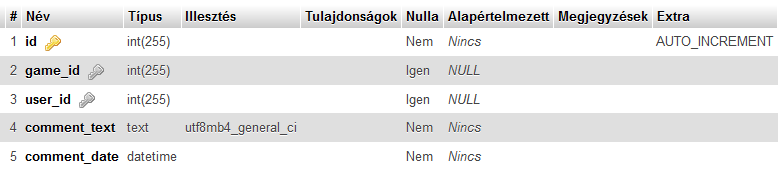
id: a kommentnek az azonosító száma (elsődleges kulcs),

game\_id: a azonosító száma (idegen kulcs),

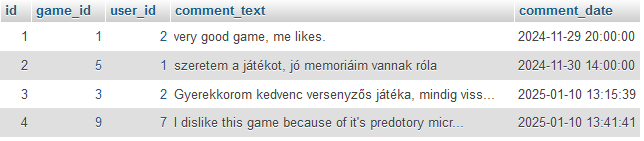
user\_id: a felhaszáló azonosító száma (idegen kulcs),

comment\_text: a kommentnek a szöveg tartalma (pl. jó játék),

comment\_date: a kommentnek a dátuma.



Az alkalmazás teszteléséhez az alábbi tesztadatokat vittünk be:



**A ratings tábla**

rating (int) az 1-töl 5-ig terjedő skálán való értékelés hogy mennyire jó egy játék, minél nagyobb annál jobb.

id: az értéklés azonosítója(elsődleges kulcs),

rating: értékelés, 1 jegyű érték,

game\_id: a azonosító száma (idegen kulcs),

user\_id: a felhaszáló azonosító száma (idegen kulcs).  


Az alkalmazás teszteléséhez az alábbi tesztadatokat vittünk be:



**Fejlesztői környezet**

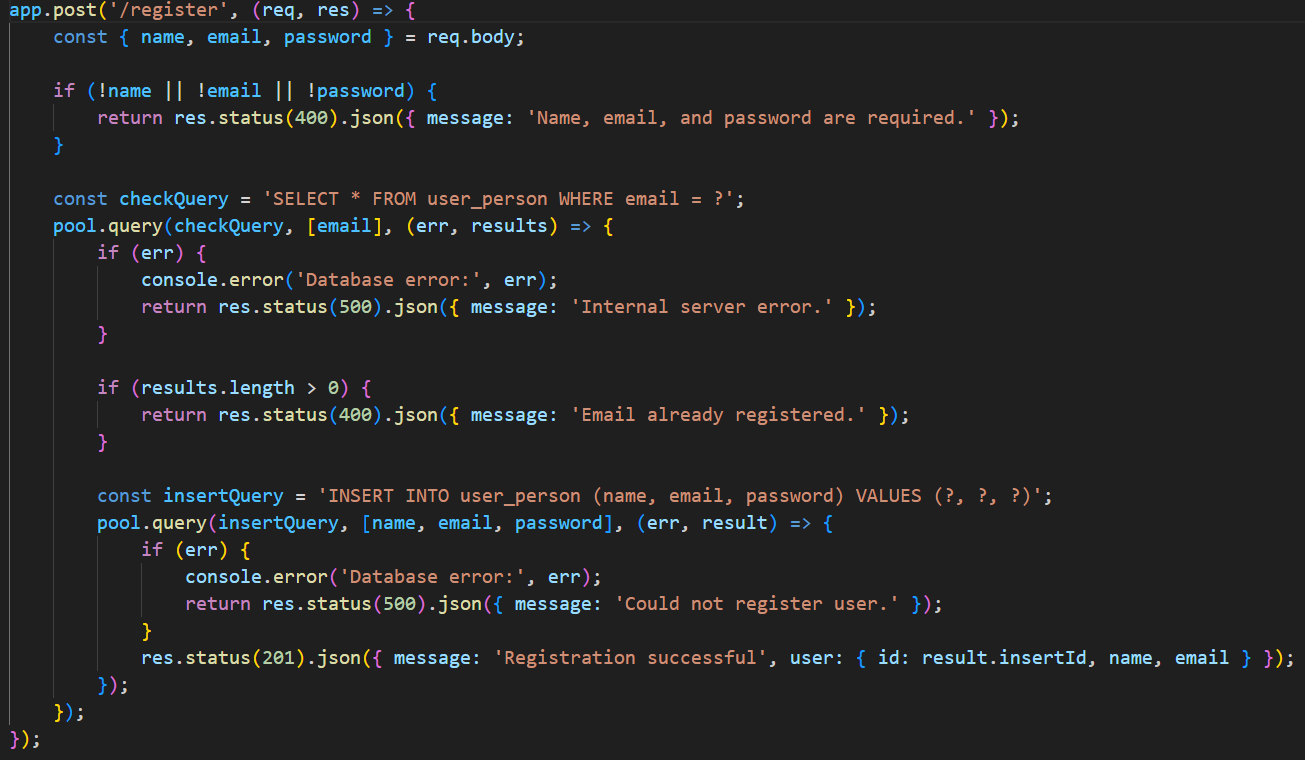
A Random Play Store weboldal fejlesztéséhez az alábbi eszközöket használtuk:

* **Visual Studio Code**: A HTML, CSS, JavaScript és PHP fájlok szerkesztéséhez.
* **XAMPP**: MySQL adatbázist és PHP-t tartalmazó program, amely lehetővé tette a weboldal helyi tesztelését.
* **PHPMyAdmin**: Az adatbázis kezelésére és karbantartására használtuk.
* **MySQL**: Az adatbázis-kezelő rendszer, amely a játékok, felhasználók, vélemények és értékelések adatait tárolja.
* **Git és GitHub**: A verziókezeléshez és a fájlok tárolásához.
* **Böngészők fejlesztői eszközei**: A Google Chrome fejlesztői eszközeit használtuk a weboldal tesztelésére és hibakeresésére.

**Backend és Frontend**

A regisztráció során a server.js egy POST kérést kezel a /register végponton. Ellenőrzi, hogy az e-mail már létezik-e, és ha nem, hozzáadja az új felhasználót a user\_person táblához.

**(server.js):**



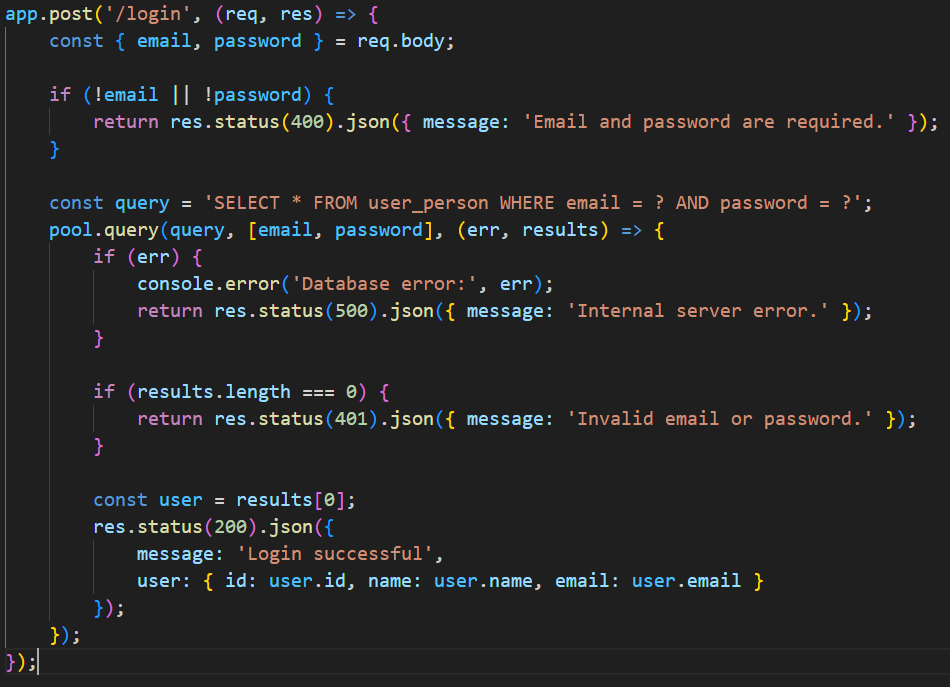
A login.js fájlban a regisztrációs űrlap adatait ellenőrizzük, majd fetch API-val küldjük a backendnek.

**(login.js):**



A server.js a /login végponton ellenőrzi az e-mailt és jelszót, majd visszaküldi a felhasználó adatait, ha a bejelentkezés sikeres.

**(server.js):**



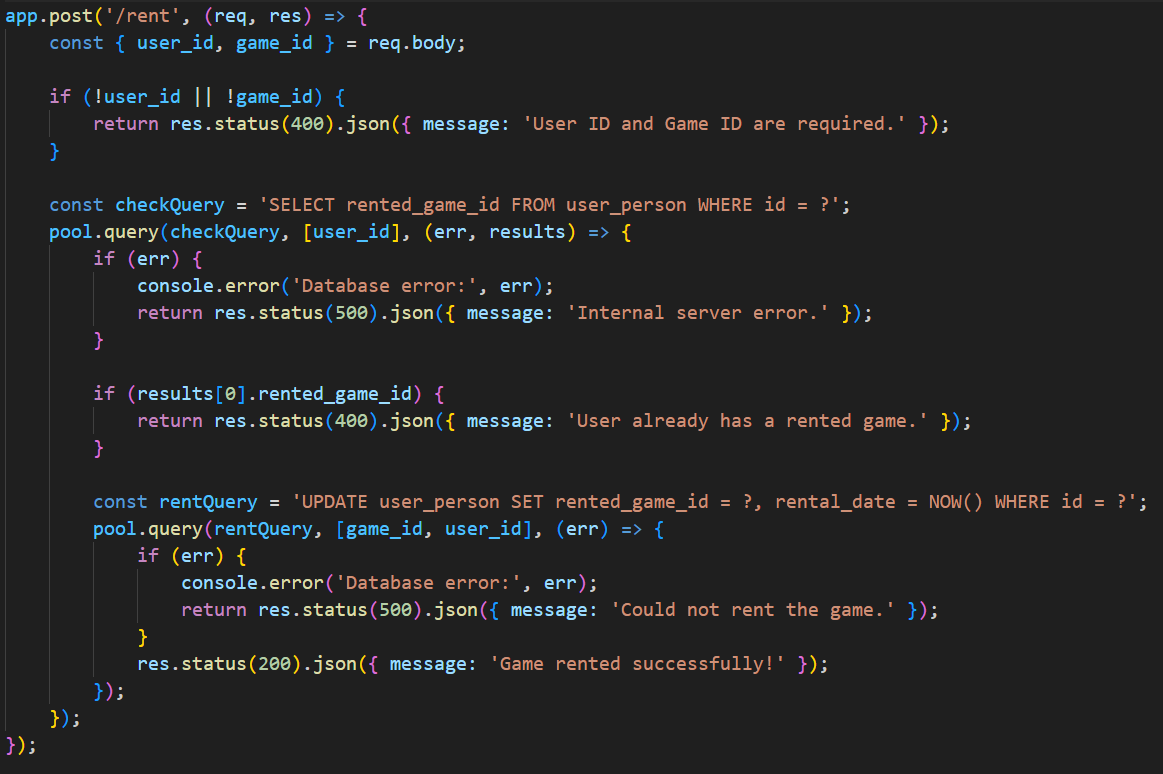
A login.js fájlban a bejelentkezési űrlap adatait fetch API-val küldjük a backendnek, és a userId-t a localStorage-ban tároljuk.

**(login.js):**



A server.js a /rent végponton ellenőrzi, hogy a felhasználó bérelt-e már játékot, és ha nem, frissíti a user\_person táblát a bérléssel.

**(server.js):**



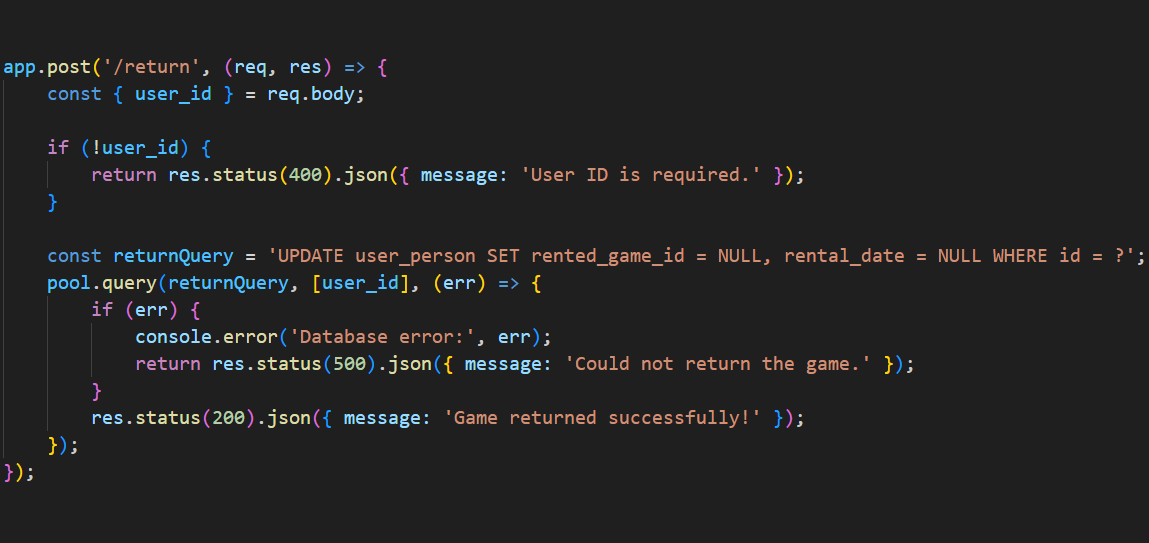
A games.js fájlban a rentGame függvény fetch API-val küldi a bérlési kérelmet, és figyelmezteti a felhasználót a bérlés részleteiről.

**(games.js):**



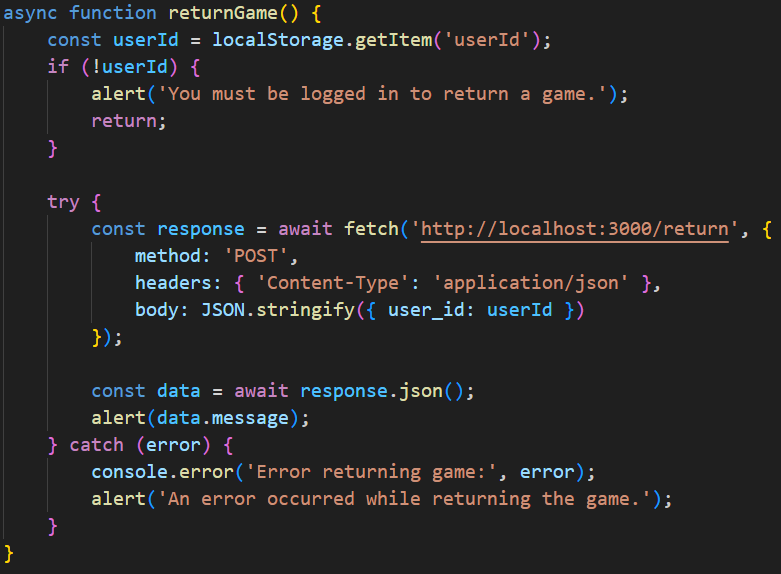
A server.js a /return végponton törli a bérlést a user\_person táblából.

**(server.js):**



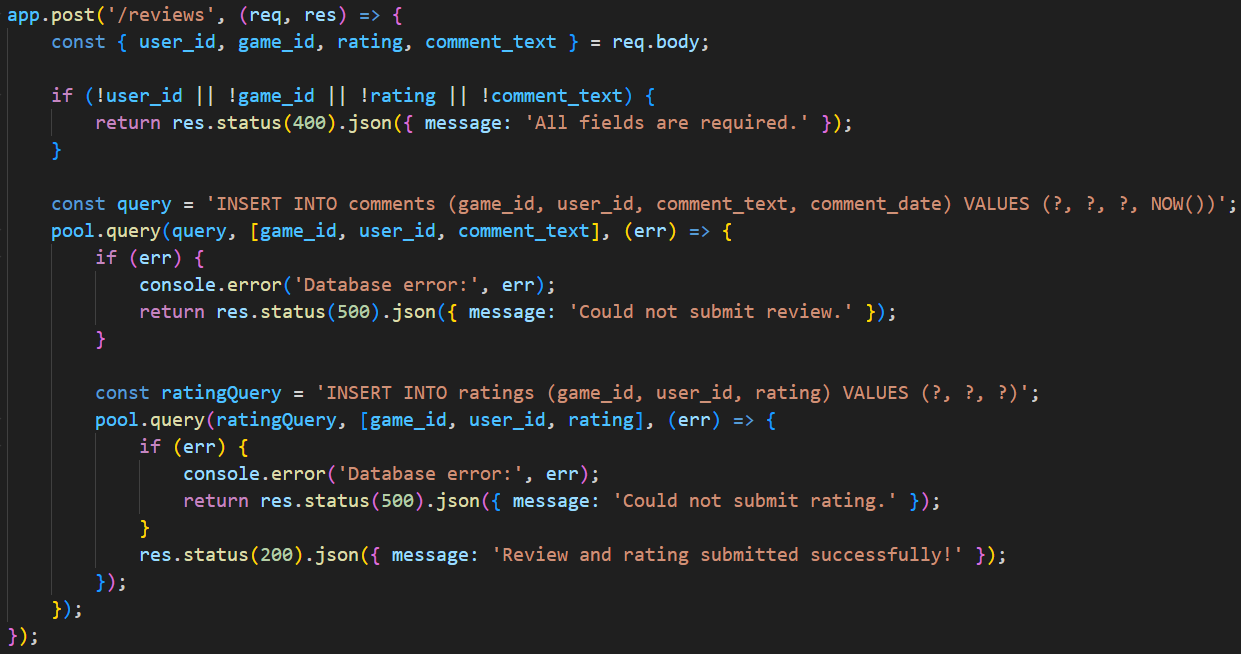
A games.js fájlban a returnGame függvény fetch API-val küldi a visszaküldési kérelmet.

**(games.js):**



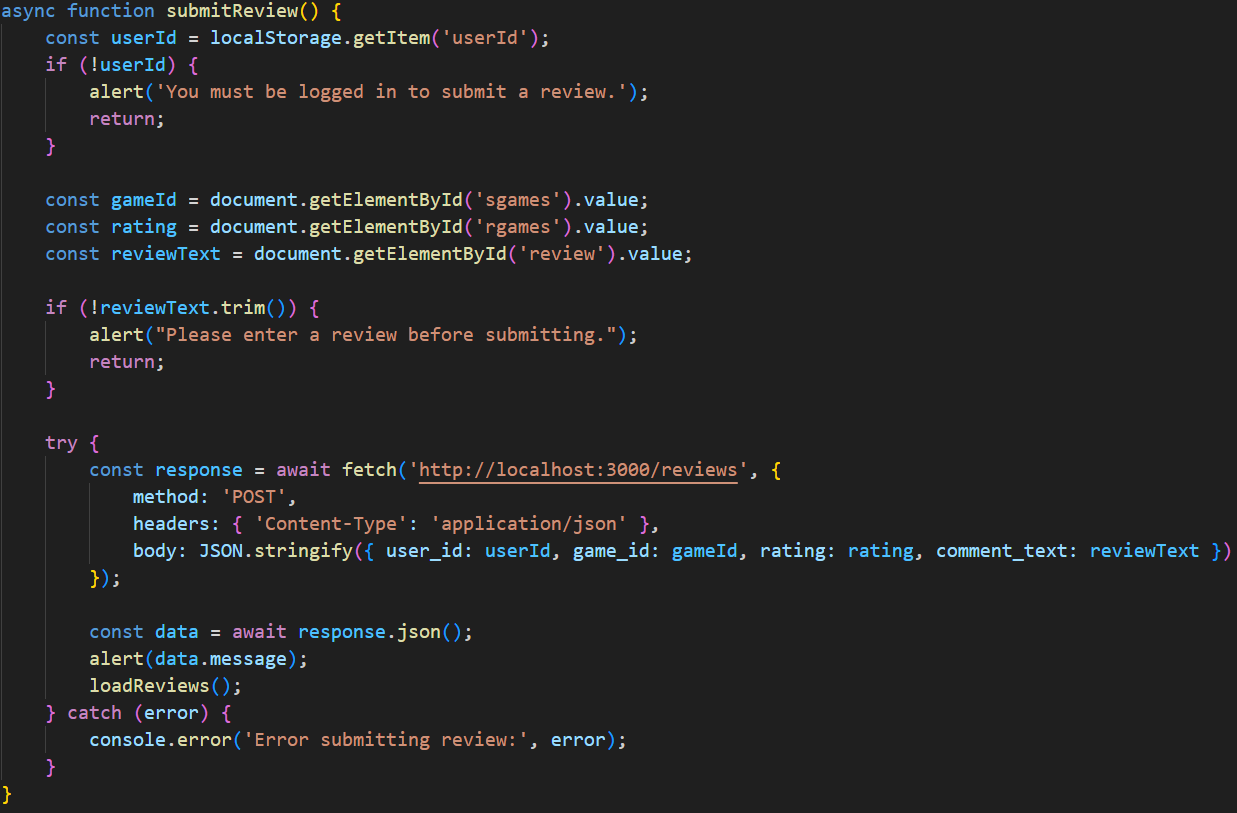
A server.js a /reviews végponton beszúrja a kommentet és az értékelést a megfelelő táblákba.

**(server.js):**



**A games.js fájlban a submitReview függvény fetch API-val küldi a véleményt, és frissíti a vélemények listáját.**

**(games.js):**



# **Felhasználói dokumentáció:**

**Weboldal elérése**

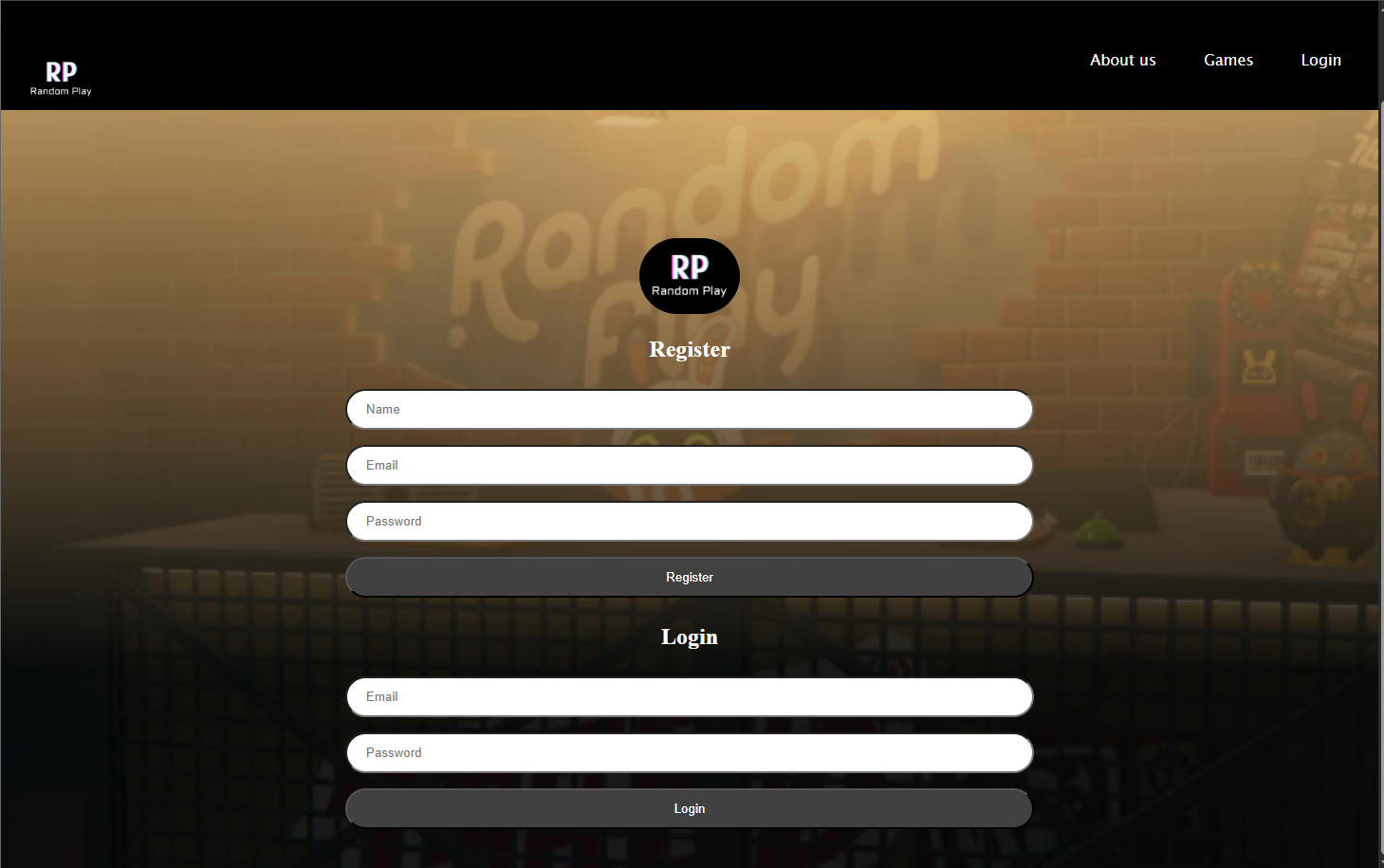
Helyi környezetben: [http://localhost:3000](http://localhost:3000" \t "_blank)

**Regisztráció és bejelentkezés**

Lépj login oldalra.

Regisztráció: Add meg a neved, e-mail címed és jelszavad, majd kattints a "Register" gombra.

Bejelentkezés: Add meg az e-mail címed és jelszavad, majd kattints a "Login" gombra.

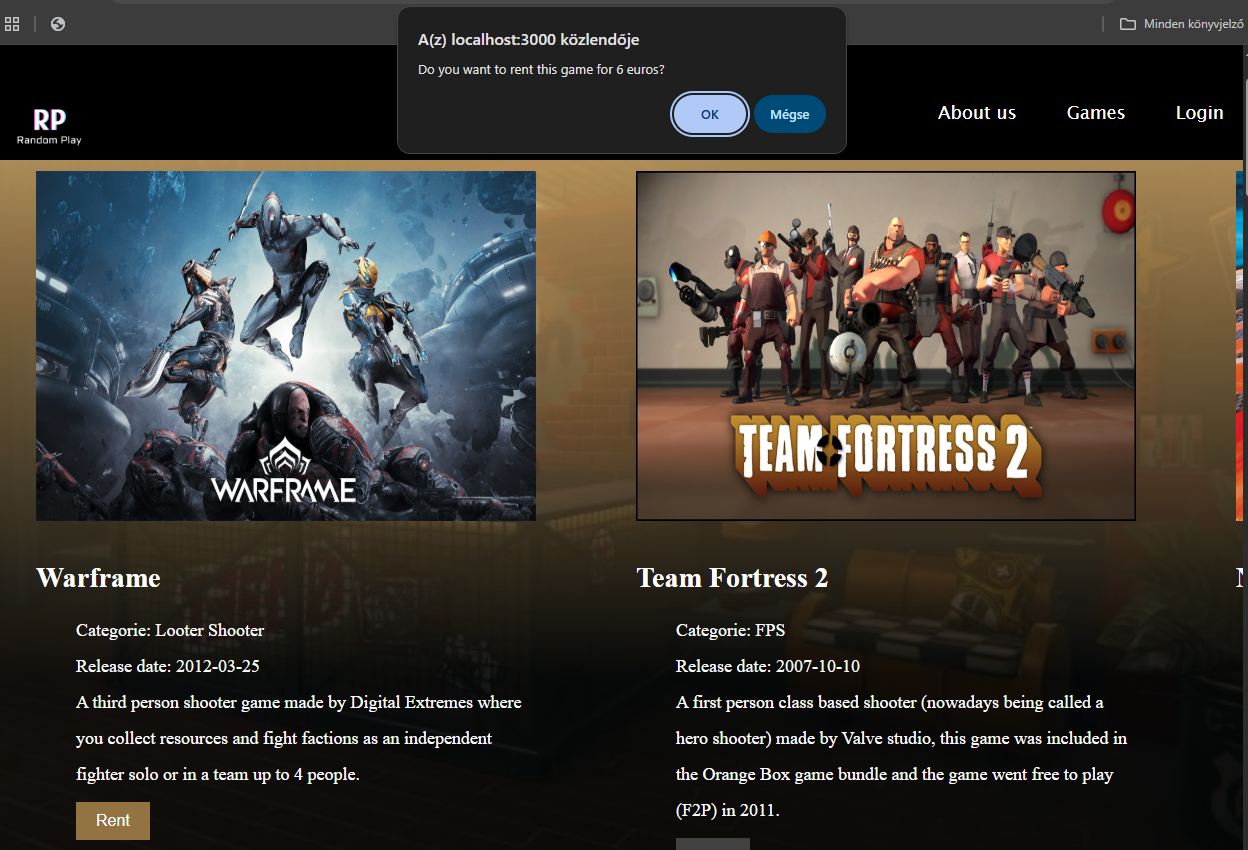


**Játék bérlése és visszaküldése**

Lépj a games oldalra.

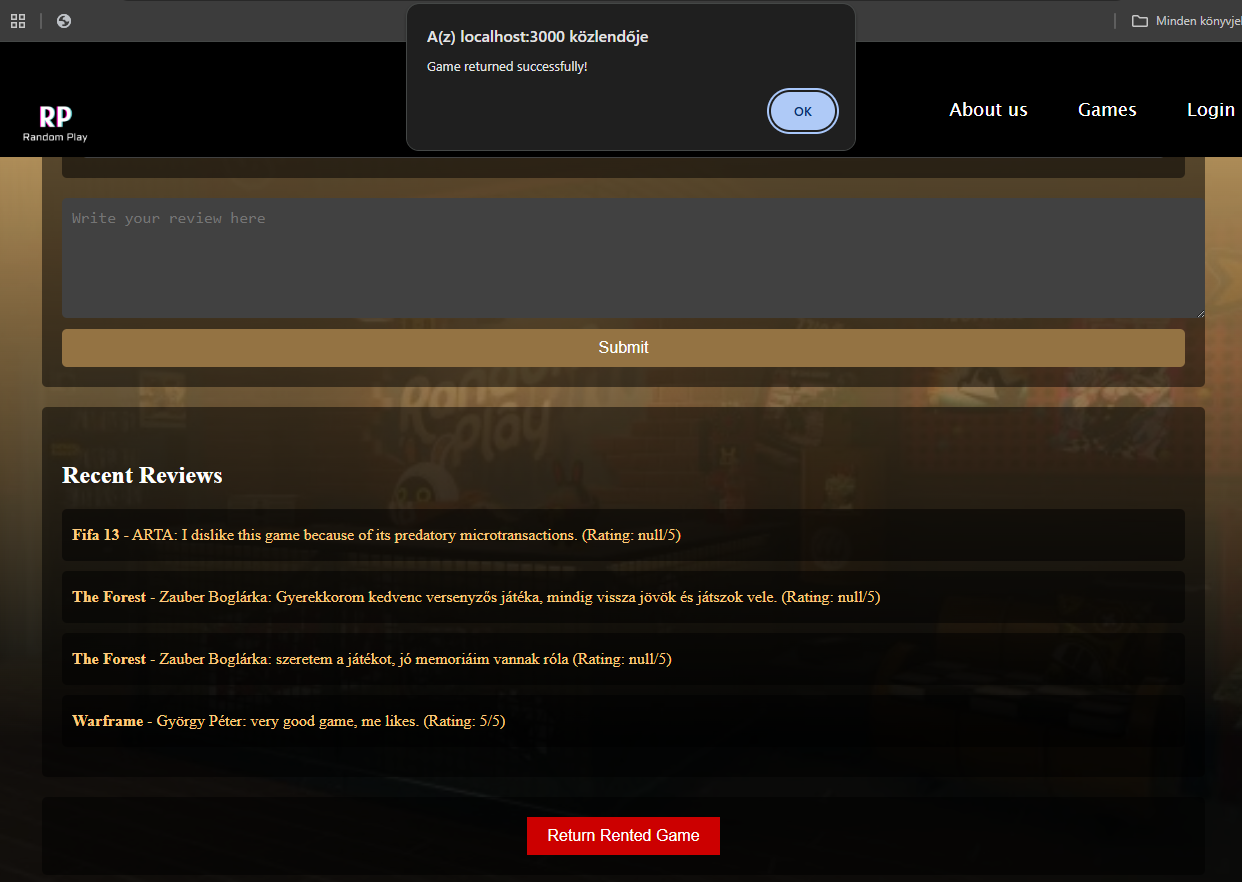
Bérlés: Válassz egy játékot, és kattints a "Rent" gombra.

Megerősítés: "Do you want to rent this game for 6 euros?"



Visszaküldés: Kattints a "Return Rented Game" gombra.

Sikeres visszaküldés esetén: "Game returned successfully!"



**Vélemény írása**

A games oldalon válassz egy játékot, adj meg egy értékelést (1-5) és a véleményedet.

Kattints a "Submit" gombra.

Sikeres beküldés esetén: "Review and rating submitted successfully!"

A véleményed megjelenik a "Recent Reviews" szekcióban.

