

Videotéka Kölcsönzési Rendszer

A **FilmFanatikusok Klubja** egy kis közösség, amely szenvedélyesen gyűjti és kölcsönzi egymásnak a kedvenc filmjeit DVD-n. Az évek során azonban egyre nehezebb lett fejben tartani, hogy kinél milyen film van, és mikor kell visszahozni. Ezért a közösség úgy döntött, hogy szükség van egy egyszerű nyilvántartó rendszerre, amely segíti a kölcsönzések követését.

A vizsgafeladat során egy olyan rendszert kell létrehozni, amely lehetővé teszi a felhasználók számára a filmek kölcsönzését. A rendszerben a filmek és a kölcsönzések nyilvántartása is szükséges, hogy a tagok mindig pontosan tudják, melyik film éppen kinél van, és mikor várható a visszahozása.

1. Backend (Laravel) – 15 pont

A backend biztosítja a REST API-kat az adatok kezelésére és a kölcsönzések nyilvántartására.

Adatbázis (2 pont)

Táblák:

movies:

- `id` (auto-increment, primary key)
- `title` (VARCHAR, max 255 karakter) – A film címe
- `director` (VARCHAR, max 255 karakter) – A film rendezője
- `available_copies` (INTEGER) – A film elérhető példányainak száma
- `created_at`, `updated_at`

rentals:

- `id` (auto-increment, primary key)
- `movie_id` (foreign key movies.id-ra)
- `borrower_name` (VARCHAR, max 255 karakter) – Kölcsönző neve
- `borrowed_at` (TIMESTAMP, alapértelmezett: aktuális időpont) – Kölcsönzés időpontja
- `returned_at` (TIMESTAMP, lehet NULL is, ha a filmet még nem adták vissza) – Visszahozás időpontja

Egy film többször is kikölcsönözhető, de csak akkor, ha van elérhető példány.

Modellek és kapcsolatok (3 pont)

- Hozz létre **Movie** és **Rental** modelleket.
- A **Movie** modell egy-a-többhöz kapcsolatban áll a **Rental** modellel (hasMany).
- A **Rental** modell egy filmhez tartozik (belongsTo).

Controller és API végpontok (8 pont)

- **GET /api/movies** – Listázza az összes filmet.
- **GET /api/movies/{query}** – Listázza a query-nek megfelelő rendezővel vagy címmel rendelkező filmeket.
- **POST /api/rentals** – Új kölcsönzés rögzítése (csak ha van elérhető példány).
- **PUT /api/rentals/{id}/return** – Kölcsönzés visszavétele (a film példányszámát növeli).
- **GET /api/rentals** – Listázza az aktív és lezárt kölcsönzéseket.

Seeder és validáció (2 pont)

Seeder fájlok:

- Hozz létre legalább **5 filmet**.
- Hozz létre **3-4 teszt kölcsönzést** (pár film legyen kikölcsönözve).

Validáció:

- Film **csak akkor kölcsönözhető, ha van szabad példány**.
- A **borrower_name** nem lehet üres.

2. Frontend (Angular) – 25 pont

A frontend biztosítja a kölcsönzések és filmek megjelenítését.

1. Alkalmazás felépítése (2 pont)

- Hozz létre egy új **Angular projektet** **angular-videoteka** néven!
- Konfiguráld az **HttpClientModule-t** az API hívásokhoz ha szükséges!!
-

2. MovieService és RentalService API-val (10 pont)

Készíts két szolgáltatást (**MovieService** és **RentalService**) Fetch API használatával:

- **getMovies()** – Összes film lekérése
- **searchMovies(query)** – Filmek keresése rendező vagy cím alapján
- **borrowMovie(movieId, borrowerName)** – Új kölcsönzés létrehozása
- **returnMovie(rentalId)** – Kölcsönzés visszavétele
- **getRentals()** – Kölcsönzések listázása

3. Komponensek (8 pont)

- **MovieComponent** – Listázza a filmeket, és lehetővé teszi keresést, valamint kölcsönzést.
- **RentalComponent** – Listázza a kölcsönzéseket, és lehetővé teszi a visszavételt.

4. Formok és validáció (5 pont)

Az űrlap validálja a mezőket:

- A **title** és **director** mezők **kötelezőek** és max **255 karakter** lehet.
- A **borrower_name** mező **kötelező** kölcsönzéskor.
- **Kölcsönzés csak akkor lehetséges, ha van elérhető példány.**