

**Szakképesítés neve:** .....

**OKJ száma:** .....

# SZAKDOLGOZAT

ReviewRiot film értékelő weboldal

Témavezető:

**Beke Béla**

**Boros Sándor**

Készítette:

**Pataki Dávid Ferenc**

*Debrecen*

*2023*

## Tartalomjegyzék

Tartalomjegyzék.....	2
1 Bevezetés .....	5
1.1 Miért ezt választottam? .....	5
1.2 Köszönetnyilvánítás .....	5
2 Felhasználói dokumentáció.....	6
2.1 Rendszerkövetelmény .....	6
2.2 Weboldal elérhetősége .....	6
2.3 Weboldal funkciói.....	6
2.4 Weboldal használata .....	7
2.4.1 Főoldal .....	7
2.4.2 Filmek .....	7
2.4.3 Keresés .....	8
2.4.4 Felhasználói profil.....	8
2.4.5 Film profil .....	8
2.4.6 Bejelentkezés / Regisztráció .....	8
2.4.7 Bejegyzéslista.....	9
2.4.8 Értékelés készítése .....	9
2.4.9 Értékelés kártyák.....	9
2.4.10 Beállítások.....	9
2.4.11 Új film készítése és módosítása .....	10
2.4.12 Felhasználók szintjének módosítása .....	10
2.4.13 Kijelentkezés .....	10
3 Fejlesztői dokumentáció .....	11

---

3.1 Használt npm csomagok .....	11
3.2 Telepítés és indítás .....	12
3.3 Backend.....	13
3.3.1 API .....	13
3.3.2 Tervezési minta (Architektúra) .....	13
3.4 Adatbázis.....	14
3.4.1 user tábla .....	14
3.4.2 auth tábla .....	14
3.4.3 follow tábla .....	15
3.4.4 permission tábla .....	15
3.4.5 review tábla .....	15
3.4.6 movie tábla .....	15
3.4.7 movie_genre tábla .....	15
3.4.8 genre tábla .....	16
3.5 Algoritmusok .....	16
3.5.1 Regisztráció.....	16
3.5.2 Belépés .....	16
3.5.3 JsonWebToken / JWT .....	17
3.5.4 Képfeltöltés .....	17
3.5.5 Bejegyzéslista.....	17
3.6 Tesztdokumentáció .....	17
3.7 TEST //TODO .....	18
3.8 Fejlesztési lehetőségek .....	18
3.9 Fejlesztői környezet .....	18
3.9.1 Kötelező (Futtatáshoz) .....	18

---

3.9.2 Ajánlott (Fejlesztéshez).....	19
4 Weboldal designja.....	20
4.1.1 Szín paletta .....	20
5 Összefoglalás .....	21
5.1 Szakdolgozat célja.....	21
6 Irodalomjegyzék.....	22

# 1 Bevezetés

A záródolgozatom témája egy filmeket értékelő és ajánló platform, ahol a felhasználók bejegyzéseket írhatnak megtekintett filmjeikről, hogy megosszák élményüket másokkal is. A felhasználók követhetnek más fiókokat, hogy könnyedén lássák azok értékeléseit. A weblap funkciói közt említést érdemel még a filmek keresése név, értékelések és műfajai alapján.

## 1.1 Miért ezt választottam?

Ezt a témát választottam, mert szeretek filmeket nézni és érdekesnek találtam a filmekről való vélemények összegyűjtését és megosztását. Továbbá úgy véltem, egy ilyen weboldal hasznos lehet mások számára is, hogy segítsen nekik a film választásában.

## 1.2 Köszönetnyilvánítás

Szeretnék köszönetet mondani Beke Béla és Boros Sándor tanárainknak a weboldalam fejlesztése során nyújtott értékes segítségért és támogatásért. Nagyon hálás vagyok a szakmai irányításért, amelyet tanáraink biztosítottak nekem, valamint a munkám fejlesztéséért nyújtott értékes tanácsokért.

## 2 Felhasználói dokumentáció

### 2.1 Rendszerkövetelmény

Mivel egy weboldaltól beszélünk így, a webszerver megfelelő futása esetében, bármilyen internettel és böngészővel rendelkező eszközön meg lehet nyitni az oldalt, ez lehet asztali számítógép, laptop vagy esetleg telefon. Böngészők között ajánlom a Google Chrome, Firefox, Opera, illetve Microsoft Edge böngészőt.

Ha ön a webszervert is futtatni szeretné akkor több szoftveres követelmény kell, ezekről a [\*Fejlesztői dokumentáció\*](#)ban találhatunk több információt.

### 2.2 Weboldal elérhetősége

A weboldal az alábbi linken lehet elérni,

[\*http://reviewriot.devla.xyz/\*](http://reviewriot.devla.xyz/)

vagy a fejlesztő által beállított címen.

### 2.3 Weboldal funkciói

A weboldalt 3 különböző „módon” tekinthetjük meg, **bejelentkezés nélkül**, **bejelentkezve**, és valamiféle **privilegizált felhasználóval belépve** (admin/moderátor).

Bejelentkezés nélkül elérhető funkciók:

- Filmek keresése és megtekintése
- Film/Felhasználói értékelések megtekintése

Bejelentkezve elérhető funkciók:

- Saját értékelés létrehozása és törlése
- Más felhasználók bekövetése
- Bekövetett felhasználók értékeléseinek könnyű megtekintése

- Saját profil módosítása (név, jelszó, leírás, profilkép)

Privilegizált felhasználók képességei:

- Moderátor:
  - Más felhasználók értékelésének törlése
- Admin:
  - Moderátor jogai+
  - Filmek módosítása és létrehozása
  - Más felhasználók jogi szintjének változtatása

## 2.4 Weboldal használata

Az oldalra érve az első felület, amit látni fogunk az a [Főoldal](#). Az oldal főbb részeit elérhetjük a bal oldalon, telefon esetén alul, található oldalsávon. Az éppen megnyitott felület mindig ki lesz jelölve. A következő paragrafusokban ezeket az oldalakat fogom bemutatni.

### 2.4.1 Főoldal

Ezen az oldalon egy általános üdvözlés és film ajánlások találhatók.

Az üdvözlés alatt található filmek két részre vannak bontva. Felkapottak, azok a filmek, amelyeket az utolsó 7 napban legtöbbször értékeltek. A Toplistás filmek pedig a legmagasabb általános értékelésűek. Rákattintás esetén átmegyünk a [film profiljára](#).

### 2.4.2 Filmek

Ezen az oldalon, ahogy a név is utalja, filmeket egy táblázatszerű kinézetben lehet végig nézni.

A felső sávban be tudjuk állítani a rendezési sorrendet, név szerinti keresést és a műfajok szerinti szűrést. Amint beállítunk egy filtert, az alatta lévő sorban meg is jelenik, innét rákattintással ki tudjuk szedni az adott filtert.

Ezután a beállított szűrőknek megfelelően megjelennek a filmek. Lefele görgetve dinamikusan jelennek meg a filmek a gyors betöltés és alacsony adat használat érdekében. Egyes filmek rákattintásával átmegyünk a kiválasztott [film profiljára](#).

### 2.4.3 Keresés

Ezen az oldalon egy gyors kereső sávot találhatunk, ahol név szerint tudunk keresni, mind felhasználókra, mind filmekre.

A felső sorban található mezőbe való beírás után, a leginkább egyformább találatok megjelennek listaszerűen. Itt lehetőségünk van rákattintani az elemre, hogy az adott elem profilját megnézhessük.

### 2.4.4 Felhasználói profil

Ezen az oldalon érhetjük el a felhasználók adatait és értékeléseit

Legfelül látható a felhasználó adatai. Név, létrehozás dátuma, saját leírás és a profilkép. Ha bevagyunk jelentkezve akkor van lehetőségünk bekövetni a felhasználót, ami a [bejegyzéslistában](#) megjeleníti a felhasználó értékeléseit.

Adatok alatt található a felhasználó értékelései, időszerinti sorrendben. Ezek szintén dinamikusnak vannak betöltve a főleges adathasználat elkerülése érdekében.

### 2.4.5 Film profil

Ezen az oldalon találhatjuk az adott filmről az összes információt és értékeléseket.

Az első rubrikában a film alap információit találhatjuk, Neve, kiadásának éve, műfajai, toplistán hanyadik helyen áll, értékeléseinek száma, átlagos értékelése, az utolsó hétben mennyi értékelés jött rá és a borítóképét. Ha admin fiókba vagyunk bejelentkezve akkor egy Módosítás gomb is megjelenik, ami átvizsgálja a film módosításának oldalára.

Alatta találhatjuk a filmre leadott értékeléseket, időszerinti sorrendben. Ha valamiféle fiókba be vagyunk jelentkezve akkor itt van lehetőségünk értékelést adni az adott filmre. Értékelés leadásakor egyből megjelenik a többi értékelés között. Itt is jelen van a dinamikus betöltés.

### 2.4.6 Bejelentkezés / Regisztráció

Az oldalsáv alján, telefon esetén a jobbszélső, ikonra rákattintva érkezünk a belépési oldalra.

Ha van fiókja, akkor felhasználónév és jelszó megadása után, Belépés gombra rákattintva tovább lépünk az oldalba. Amennyiben hibás adatokat adott meg vagy szintaktikailag helytelen a beírt szöveg. A honlap értesíti önt.



Amennyiben még nincs fiókja, akkor a Regisztráljon link rakattintása után egy hasonló oldallal találkozunk. Itt ügyelni kell, hogy a felhasználónév hosszának 5-32 karakter között kell lennie, az email az betarja a „felhasználónév@szolgáltató.com” formátumot és a jelszó hossza is 8 és 55 karakter között legyen, de hiba esetén szól a honlap. Ha véletlenül már ezzel a felhasználónévvel vagy email címmel regisztráltak volna, szintien hibaüzenetet kapunk.

Sikeres regisztráció esetén egy felugró ablakkal jelez és visszaküldi önt a Bejelentkezés ablakra.

#### 2.4.7 Bejegyzéslista

Ez a fül csak bejelentkezett fiókkal jelenik meg és érhető el. Ez az oldal a bekövetett fiókok értékeléseit mutatja.

//todo

#### 2.4.8 Értékelés készítése

Értékeléseket a Bejegyzéslistában és a Film profilján készíthet. (Ha a bejegyzéslistában vagyunk akkor kiválasztjuk a filmet) Beállítjuk, hogy egy 1-5 skálán mennyire élveztük az adott filmet, és amennyiben kívánjuk megjegyzést is írhatunk. Ezután az Értékelés gombra rákattintva megjelenik a többi értékelés között.

#### 2.4.9 Értékelés kártyák

Értékelés kártyákkal 3 helyen találkozhatunk, bejegyzéslista, felhasználó/film profil. Ez tartalmazza a felhasználó által megadott értékelést és leadás dátumát, amennyiben a kártya környezetéből nem egyértelmű tartalmazhatja a készítő nevét és profilképét és/vagy a film nevét és borítóképét.

Amennyiben az értékelést ön készítette vagy a belépett fiók moderátor, vagy annál magasabb joggal rendelkezik, akkor van lehetőség a postot törölni. A törlés gombra kattintva megjelenik megerősítő ablak, igen lenyomása után a post törlődik.

#### 2.4.10 Beállítások

Ez a fül csak bejelentkezett fiókkal jelenik meg és érhető el. Itt van lehetőség új-felhasználónevet, jelszót, leírást és profilképet berakni. A jelszó változtatáshoz a jelenlegi

jelszót is meg kell adni biztonsági okok miatt. A profilkép méretének 1 megabájt alatt kell lennie.

Amennyiben ön admin szintű fiókkal van belépve egy felső navigációs sáv is megjelenik. Itt van lehetőség a filmeket változtatni és más felhasználók szintjét módosítani.

#### 2.4.11 Új film készítése és módosítása

Ide csak admin fiókkal tudunk eljutni az előző paragrafusban megjelölt helyen. Az új film készítése és jelenlegiek módosítása hasonló felületen történik.

Miután a listából kiválasztottuk melyik lehetőséggel szeretnénk élni. Az oldalon van lehetőség a film címének, kiadási évének, műfajának és borítóképének változtatására. Műfajoknál többet is megadhatunk és a buborékokra kattintva ki is szedhetünk. A borítókép 2 megabájt alatt kell lennie.

#### 2.4.12 Felhasználók szintjének módosítása

Ide csak admin fiókkal tudunk eljutni az előző paragrafusban megjelölt helyen. Itt van lehetőség más fiókok jogi szintjének változtatására.

Egy listaszerű módon fel vannak sorolva a felhasználók és a rubrikában található kiválasztó segítségével megváltozik az adott felhasználó jogköre.

#### 2.4.13 Kijelentkezés

Amennyiben úgy gondoljuk nincs más dolgunk az oldalon, az oldalsáv alján található felhasználónévre rákattintva és az utána történő megerősítő ablak elfogadása után, kilépünk a fiókból.

## 3 Fejlesztői dokumentáció

A weboldal fejlesztéséhez [Node.js](#)-t alkalmaztam. Azért ezt használtam mivel a JavaScript programozási nyelvet jobban elsajátítottam szabad időmben, mint más interaktív weboldal készítésére használt nyelveket, például a PHP-t. Abban is előnyösebb nekem ez a környezet, mert a Frontend - Backend kódot jobban tudom szeparálni.

A JavaScript programozási nyelvet kiegészítem a **TypeScript** nevezetű szintaxis és statikus típus ellenőrző készlettel, a fölösleges hibák elkerülése érdekében.

A Node.js-en belül a [Next.js](#) **keretrendszert** használom, ami a React frontend könyvtárat kiegészíti egy Express-hez hasonló backend RESTful API készítő csomaggal.

A [React](#) a *Meta* által létrehozott JavaScript könyvtár, mellyel könnyen lehet készíteni interaktív felhasználói felületeket.

Adatbázisnak a MySQL alapú **MariaDB**-t alkalmaztam.

### 3.1 Használt npm csomagok

A fentiekén kívül használtam még másmilyen különböző úgynevezett npm csomagokat kisebb problémák megoldására:

- [bcrypt](#): Egy Kriptografikus könyvtár, amely a jelszók biztonságosabb eltárolása érdekében használlok.
- [jsonwebtoken](#): Röviden JWT, egy token alapú autentikációs módszer, amely a session-el ellentétben a kliensen tárolja a bejelentkezett fiók adatait. A szerver csak egy kriptografikus kulcs segítségével validálja a kienstől kapott tokent.
- [cookies-next](#): Next.js-ben nincs alapból cookie-k módosítására lehetőség, de van erre kifejlesztett csomag a készítőktől.
- [mysql2](#): Az adatbázis kapcsolat létrehozására és felhasználására használt könyvtár
- [multiparty](#): Képek feltöltéséhez űrlapot/form-ot kell alkalmazunk, ennek a kódolása JSON helyett *multipart/form-data*, amelyet Next.js alapból nem képes fordítani, ezért használnom kell egy külön csomagot.

### 3.2 Telepítés és indítás

A project optimális futása érdekében erősen ajánlott a *Fejlesztői környezet*-ben megjelölt Node.js verzió használata. Régebbi, ritka esetben akár új verziók is, képesek előre nem látható problémákat okozni, amelyek akadályozhatják a program futását.

A */next.config.js* állományban érdemes átírni az adatbázis hitelesítő adatait.

*databaseHost: elérés cím*

*databaseUser: felhasználói fiók*

*databasePassword: felhasználó jelszava*

*databaseDatabase: adatbázis neve*

A */scripts* mappában található telepítési és indítási scriptek Linux-ra és Windows-ra, ha a parancssor túl rémisztő.

- *./install*: telepíti a npm csomagokat, létrehozza az adatbázist és dinamikusan generál titkos szerver oldali tokeneket a kriptografikus műveleteknek.
- *./start-dev*: Elindítja a fejlesztői környezetet, ebben a módban a projectben történő változtatások egyből megváltoznak a weboldalon is, viszont ez fölösleges rendszer erőforrásokat és optimalizálatlan kódot futtat ezért a weboldal lassabbnak tűnhet.
- *./build & ./start-prod*: Ezzel a két scripttel lehet egy optimalizáltabb környezetet létrehozni a projectnek, viszont minden változtatás után újra le kell futtatni a építés parancsot.

Első indítás után érdemes belépni az „admin” fiókba és megváltoztatni a belépési adatokat.

**Alapértelmezett felhasználónév:** admin

**Alapértelmezett jelszó:** 123456789

### 3.3 Backend

#### 3.3.1 API

Az oldal API (Application Programming Interface) -ként is szolgál. Amennyiben programozóként felkarjuk használni az adatbázisban található filmeket vagy új kinézetet akarunk adni az oldalnak. A */api* elérési úton hozzáférhetünk az összes funkcióhoz, amit az oldal felajánl. A lehetőségeket, amiket az API szolgál a */dev* úton található fileok között mutatom be.

#### 3.3.2 Tervezési minta (Architektúra)

A backend kód struktúrája a Model-Service-Controller (Modell-Szolgáltatás-Vezérlő) architektúrán (röviden MSC architektúra) alapul. Minden API kérés ezen a „vezetéken” megy át. Három fontos részből áll:

1. *Controller*: A beérkező kéréseket irányítja a megfelelő irányba, ezen a rétegen történik a http kérésének szintaktikai ellenőrzése, és a http válaszok visszaküldése.
2. *Service*: Ezen a rétegen történik a komplexebb ellenőrzések (pl. felhasználó név létezik e már az adatbázisban), ez a réteg konkrétan nem foglalkozik az adatbázissal, sem a http kapcsolatokkal, de viszont egy fontos átmeneti réteg a következő rétegnek.
3. *Model*: Feladata az adatok lekérdezése az adatbázistól és annak értelmezése.

Ahogy lehet látni minden rétegnek meg van a saját felelőssége. Kisebb projektekben túlzás ilyen architektúrákat alkalmazni, lassítja a fejlesztési sebességét, viszont ezzel a tervezési mintával javítható a szoftver skálázhatósága és karbantarthatósága.

### 3.4 Adatbázis



#### 3.4.1 user tábla

Ez a tábla a felhasználók alap adatait tárolja.

- **id**: Egyedi elsődleges kulcs
- **name**: Egyedi szöveg típus, a felhasználónevet tárolja
- **created\_at**: Dátum típus, a felhasználó regisztrálásának dátumát tárolja
- **description**: Szöveg típus, nem kötelező, felhasználó által beállított leírás vagy szöveg
- **picture\_path**: Szöveg típus, nem kötelező, felhasználó profilképének a szerveren található file neve
- **permission\_id**: Idegen kulcs a permission táblába, a felhasználó jogát határozza meg

#### 3.4.2 auth tábla

Ez a tábla tartalmazza a felhasználók autentikációs adatait.

- **id**: Egyedi elsődleges kulcs
- **user\_id**: Egyedi idegen kulcs a user táblába, mely felhasználó belépési adatai az egyed
- **email**: Egyedi szöveg típus, a felhasználó email címe
- **password\_hash**: Fix hosszúságú szöveg, a jelszavakat soha nem tároljuk egy per egy az adatbázisban, hanem a jelszót odaadjuk egy egy-irányú enkriptációs algoritmusnak és csak az eredményt tároljuk.

A jelszó tárolásáról a [Algoritmusok](#)(TODO) fejlécben többet megtudhatunk.

### 3.4.3 follow tábla

Követések feljegyzésére szolgáló tábla.

- **whoUserID**: Idegen kulcs a user táblába, melyik felhasználó követ
- **whomUserID**: Idegen kulcs a user táblába, melyik felhasználót követi

### 3.4.4 permission tábla

A különbözőféle jogi szintek tárolására szolgáló tábla. Jelenleg 3 különböző szintet különböztünk meg, Felhasználó, Moderátor, Admin.

- **id**: Egyedi elsődleges kulcs
- **name**: Egyedi szöveg típus, a jog neve
- **level**: Szám típus, amely 0-255 között meghatározza a jog „szintjét”, minél magasabb annál több képessége van annak a jognak

### 3.4.5 review tábla

A felhasználók által generált értékelések tárolására szolgáló tábla

- **id**: Egyedi elsődleges kulcs
- **author\_id**: Idegen kulcs a user táblába, az értékelés szerzőjét határozza meg
- **movie\_id**: Idegen kulcs a movie táblába, mely filmnek az értékelése
- **rating**: Szám típus, 1-10 közötti érték, amely meghatározza hogyan értékelte a filmet egy tízes skálán
- **description**: Szöveg típus, nem kötelező, felhasználó által beállított leírás vagy szöveg
- **create\_date**: Dátum + idő típus, értékelés létrehozásának idejét tárolja

### 3.4.6 movie tábla

A filmek tárolására szolgál

- **id**: Egyedi elsődleges kulcs
- **name**: Egyedi szöveg típus, a film nevét tárolja
- **release\_date**: Év típus, a film kijövetelének évét tárolja
- **image\_path**: Szöveg típus, nem kötelező, a film borítóképének a szerveren található file neve

### 3.4.7 movie\_genre tábla

Mivel egy filmnek lehet több műfaja és egy műfajnak lehet több filmje, kell egy összekapcsolási tábla, ez az

- **movie\_id**: Idegen kulcs a movie táblába, melyik filmnek a támáját állítjuk most
- **genre\_id**: Idegen kulcs a genre táblába, milyen műfajt állítunk be

#### 3.4.8 genre tábla

Film műfajok tárolására szolgáló tábla

- **id**: Egyedi elsődleges kulcs
- **name**: Egyedi szöveg típus, műfaj nevét határozza meg

### 3.5 Algoritmusok

#### 3.5.1 Regisztráció

Az új felhasználó adatait a **/auth** oldalon adjuk meg. Amikor a **Regisztráció** gombra kattintunk, ellenőrizzük a szövegmezőket annak érdekében, hogy helyesen legyenek formázva: pl. a felhasználónév hossza 5 és 32 karakter között legyen, az e-mail cím pedig kövesse az „username@server.domain” formátumot stb. Ha bármilyen hiba merül fel, értesítjük a felhasználót. Ellenkező esetben HTTP kérést küldünk a **/api/user** címre.

A backend-en a Controller újra ellenőrzi a kérés testének formázását. Ezután a Service réteg ellenőrzi, hogy van-e már hasonló felhasználónév vagy e-mail cím az adatbázisban. Ha igen, akkor értesítjük a felhasználót, ha nem, akkor folytatjuk a Model réteggel. Itt a megadott jelszót **bcrypt** segítségével kódoljuk. Ezután tároljuk a **user** és az **auth** adatbázis táblában a felhasználói és az azonosítási adatokat. Amennyiben minden sikeres a felhasználót átirányítjuk a belépési oldalra.

#### 3.5.2 Belépés

Belépési adatokat a **/auth** oldalon adhatjuk meg. Itt is ellenőrizzük a felhasználónév és jelszó hosszát, hiba esetén szólunk a felhasználónak. Ezután egy HTTP kérést küldünk a **/api/auth** címre.

A backend-en ellenőrzés után, megnézzük, hogy létezik-e ilyen nevű felhasználó. Ha igen akkor **bcrypt** segítségével összehasonlítjuk az adatbázisban tárolt hash-et a felhasználó által megadott jelszóval. Hibás adatok esetén értesítjük a felhasználót. Amennyiben megfelel az adat, visszaküldünk egy **JWT** token-t, amivel a böngésző a későbbi kéréseknél be tudja bizonyítani, hogy melyik felhasználó küldte a kérelmet. Ezt a süti-ben tároljuk.



### 3.5.3 JsonWebToken / JWT

Ahogy az előző paragrafusban említettem, az autentikációhoz JSON Web Token-eket alkalmazunk. A belépés-t követően a böngésző sütik-ben tároljuk, hogy későbbi kéréseknél automatikusan elküldjük. A Session-el ellentétben a Tokeneket nem kell adatbázisban tárolni, helyette a felhasználó tárolja és egy enkriptációs algoritmussal bizonyítjuk be, hogy az a token tényleg valódi.

### 3.5.4 Képfeltöltés

Kép feltöltésre két helyen van lehetőség a honlapon, felhasználói profilkép és film borítókép, mind a kettő elérhető a beállítások fülben. A honlap csak képet enged kiválasztani. Miután kiválasztottuk a képet, megnézzük, hogy nem nagyobb-e mint 1 megabájt. Hiba esetén szólunk a felhasználónak. Ezután egy kérést küldünk a `/(movie/user)/[id]/update` címre.

A backenden megint ellenőrizzük, hogy a kép megfelel-e a kritériumoknak, amennyiben rendben van minden, generálunk egy random UUIDv4 szöveget, megnézzük, hogy létezik-e már ilyen file, ha igen generálunk tovább, ha nem akkor a generált szöveggel és a file kiterjesztésével lementjük a fájlrendszerbe a képet. Ezután az adatbázisba frissítjük a kép címét. Az előző képet töröljük a filerendszerből. Bármilyen hiba esetén értesítjük a felhasználót.

Amennyiben minden sikeres volt az oldal frissít és az új képet fogjuk látni.

### 3.5.5 Bejegyzéslista

//TODO

## 3.6 Tesztdokumentáció

Server futtatása több környezetben is volt tesztelve:

- Otthoni gépem (*Windows 10 21H2 + WSL*)
- Raspberry Pi (*Debian GNU/Linux 11 (bullseye) aarch64*) \*Jelenlegi szerver\*

A honlap is több böngészőben és felbontásban volt tesztelve:

**Asztali:**

- Böngésző

- Google Chrome *Version 111.0.5563.111*
- Firefox Developer Edition *Version 112.0b9*
- Opera GX *Version 97.0.4719.17*
- Microsoft Edge *Version 112.0.1722.39*
- Felbontás
  - 16:9 1920x1080
  - 4:3 1024 x 768

#### Telefon:

- Xiaomi Mi A1 421 x 748
- iPhone SE 375 x 667
- iPhone XR 414 x 896
- iPhone 12 Pro 390 x 844

### 3.7 TEST //TODO

### 3.8 Fejlesztési lehetőségek

- Bejegyzésekre válaszolás, nem csak a filmekre lehet válasz írni, hanem a bejegyzésekre is.
- Egyes bejegyzések értékelése, amivel a fölösleges/értelmetlen bejegyzéseket valamilyen szinten moderálni lehet.
- Moderációs rendszer bővítése: felhasználók jelentése, kitiltása.

### 3.9 Fejlesztői környezet

#### 3.9.1 Kötelező (Futtatáshoz)

A következő programok elkerülhetetlenek a webserver futtatása érdekében, elméletben minden program kompatibilis a régebbi verzióban írt kódokkal, viszont hiba esetén ajánlatos a megjelölt verziót használni.

- **Node.js v18.15.0 (LTS):** nyílt forráskódú JavaScript futtatókörnyezet, amely lehetővé teszi a JavaScript szerveroldali futtatását
- **npm v9.5.0:** JavaScript csomagkezelő, amely lehetővé teszi a különböző JavaScript modulok letöltését és telepítését
- **MariaDB v10.4.25:** nyílt forráskódú relációs adatbázis-kezelő rendszer

### 3.9.2 Ajánlott (Fejlesztéshez)

A következő programok csak a saját preferenciáim, ezeket a programokat tetszés szerint ki lehet cserélni másra.

- **git**: elosztott verziókezelő rendszer, amely lehetővé teszi a programozók számára, hogy nyomon kövessék a kódbázis változásait
- **Visual Studio Code**: kódszerkesztő
- **phpMyAdmin**: webes adatbázis-kezelő alkalmazás

## 4 Weboldal designja

A weboldal designja minimalista, és inspirációt merít a „material design” stíusból.

A „material design” stílusra jellemző az egyszerűség, az átláthatóság és a letisztultság. A design célja, hogy segítse a felhasználókat könnyedén és intuitívan navigálni az oldalon és hogy az információkat egyszerűen láthatóvá tegye.

Az oldalon erősen használom az egérkurzorral való elem kiválasztásoknak lehetőségének mutatását, egy világos árnyékkal és az elem minimális megnagyobbításával.

A CSS kód saját, nem használtam keretrendszert (pl. Bootstrap, Tailwind).

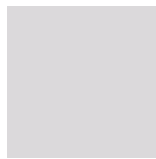
### 4.1.1 Szín paletta

A színpaletta létrehozása fontos része az oldal designjának, hiszen ez segít abban, hogy az oldalnak legyen egy saját, egyedi érzése és megjelenése.



*#252627*

Háttérszín



*#DBD9DB*

Szöveg szín



*#1A1818*

Háttér kiemelő



*#D5A021*

Állapot/kiemelő



*#048BA8*

Linkek

## 5 Összefoglalás

### 5.1 Szakdolgozat célja

## 6 Irodalomjegyzék

- <https://nextjs.org/docs>
- <https://react.dev/reference/react>
- <https://stackoverflow.com/>
- <https://www.w3schools.com/>
- <https://www.svgrepo.com/>
- <https://imdb-api.com/>