Retro Film Kincsek

Informatikai rendszerterv

Tartalom

[1 Informatikai rendszerterv sablon 3](#_Toc132120074)

[2 Fogalmak és rövidítések 3](#_Toc132120075)

[3 Architektúra 4](#_Toc132120076)

[3.1 Áttekintés 4](#_Toc132120077)

[3.2 Választott technológiák 7](#_Toc132120078)

[3.3 Függőségek 10](#_Toc132120079)

[4 Modulok 10](#_Toc132120080)

[4.1 Áttekintés 10](#_Toc132120081)

[4.2 Modulok részletei 10](#_Toc132120082)

[5 Perzisztencia 11](#_Toc132120083)

[5.1 A tárolás választott módja 12](#_Toc132120084)

[5.2 Koncepcionális adatmodell 12](#_Toc132120085)

[6 Folyamatok 12](#_Toc132120086)

[7. Interfészek 12](#_Toc132120087)

[8. Biztonság 12](#_Toc132120088)

[9. Üzemeltetést támogató képességek 12](#_Toc132120089)

[9.1. Konfiguráció 12](#_Toc132120090)

[9.2. Monitorozás 12](#_Toc132120091)

[9.3. Logolás 12](#_Toc132120092)

[9.4. Hibaelhárítás 12](#_Toc132120093)

# Informatikai rendszerterv sablon

Verzió: 1.0

A projectmunkám egy filmajánló oldal, ami a régebbi alkotásokra specializálódott. A filmek azon rajongóinak készítettem ezt az oldalt, akik kifejezetten szeretik a régebbi filmeket, illetve azoknak, akik esetleg emlékeznek egy filmre fiatalabb korukból, viszont az idő homályában elfeledték annak a címét, esetleg csak egy színész vagy egy megjelenési évre emlékeznek. Erre a problémára hozna megoldást az oldalam, hiszen rengeteg szűrési lehetőség van, illetve az is könnyíti a keresést, hogy csak 2000 előtti filmek szerepelnek az oldalon, ezért, ha egy rendező több filmet is csinált később, már sokkal inkább leszűkült a találatok száma.

Azért választottam ezt a témát, mert régebben rendszeres program volt a filmnézés apukámmal, viszont ő sem emlékezett sokszor a film címére. Azt, hogy mikor látta a moziban vagy hogy milyen híresebb színész játszott benne azt meg tudta mondani, főleg az előbbit. Az IMDB-n viszont a rengeteg film miatt sokszor feladtuk a keresést, illetve az angol leírás sem segített minket közelebb az esti nézendő megtalálásához.

# Fogalmak és rövidítések

HTML5:

Az HTML5 egy olyan nyelv, amely lehetővé teszi a webfejlesztők számára, hogy jobban és könnyebben hozzáadjanak tartalmat és interakciókat az internetes oldalakhoz és webalkalmazásokhoz. Az HTML5 segít a fejlesztőknek abban, hogy videókat, animációkat és egyéb multimédiás tartalmakat használjanak az oldalakon, és a mobil eszközökkel is kompatibilisebbé teszi a weboldalakat. Az HTML5 sok új elemet és attribútumot vezetett be, amelyek segítségével a fejlesztők jobban és hatékonyabban tudnak dolgozni, és kreatívabb és jobban interaktív webalkalmazásokat hozhatnak létre.

CSS:

A CSS (Cascading Style Sheets) egy stílusleíró nyelv, amelyet a webfejlesztők használnak az internetes oldalak stílusának meghatározásához és formázásához. A CSS szétválasztja az oldal tartalmát és megjelenését, azaz lehetővé teszi a fejlesztők számára, hogy elkülönítsék az oldal tartalmát a formázástól.

MYSQL:

MySQL egy nyílt forráskódú, relációs adatbázis-kezelő rendszer. Az adatbázis-kezelő rendszerek olyan szoftverek, amelyek segítségével adatokat lehet tárolni, szervezni, kezelni és lekérdezni. Az adatokat táblákban tárolják, és a MySQL lehetővé teszi az adatok hatékony tárolását és keresését.

JAVAScript:

JavaScript egy szkriptnyelv, amely lehetővé teszi a weboldalak interaktív funkcióinak készítését. A JavaScript segítségével lehetőség van arra, hogy a weboldalak dinamikusabbá váljanak, többféle felhasználói interakcióval, animációkkal és más dinamikus elemekkel.

.NET:

.NET egy Microsoft által kifejlesztett szoftver platform, amely lehetővé teszi az alkalmazások fejlesztését és futtatását különböző operációs rendszereken, mint például a Windows, Linux és macOS.

# Architektúra

## Áttekintés

A rendszer két fő modulból áll

* Desktop app   
  Adminisztrációs funkciók megvalósítása
  + Műszaki környezet
* WEBkiszolgáló
  + M

**Működési környezet:** Windows környezet

* Windows .NET app
* Alkalmazásszerver - NodeJS v16.13.2
* MySQL adatbázisszerver 8.0
* Connector/NET 8.0.28
* Támogatott böngészők (Chrome, Firefox)
* Használt protokollok (http, https)
* SSL tanúsítvány
  + Teszteléshez használt: OpenSSL programmal generált önaláírt tanúsítvány
  + Az átadott vezióhoz ajánljuk valamelyik tanúsítványkiszolgálótól megvásárolni a használt domainhez (pl**. RetroFilmKincsek.hu**) tartozó tanúsítványt.
  + Tanúsítvány tároló mappa helye:

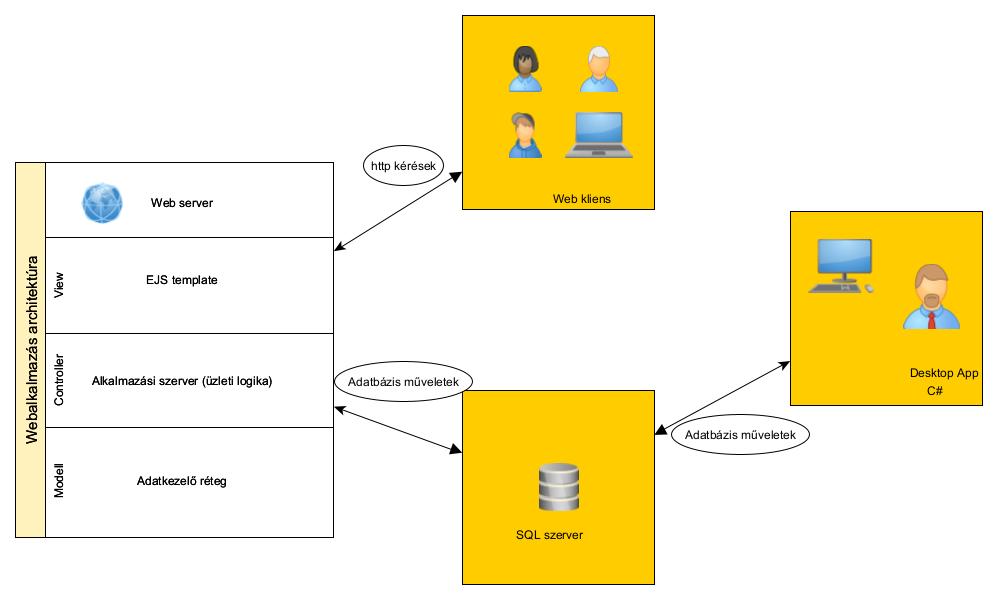
[telepítési mappa]\cert\localhost\  
\bin\bin.js file tartalmazza a használt https tanúsítvány elérhetőségét:

const options ={

key:fs.readFileSync(path.join(\_\_dirname,'../cert/localhost/localhost.decrypted.key')),

cert:fs.readFileSync(path.join(\_\_dirname,'../cert/localhost/localhost.crt'))

}



1. ábra

**Rétegek**



**2**. ábra

**A Router Layer** tartalmazza az alkalmazás programozási felületének (API) útvonalait.

**A réteg felelőssége:**

Meghívja az alkalmazás szerver megfelelő metódusát és várja a válaszát. A választ küldjön a szervertől a böngészők felé.

**A modul helye:** /db/dboperation.js

**A réteg konponensei:**

* **index.js** – általános útvonalak kezelése
  + **URL:** /kepek/:page - ***:page*** az oldalszám paramétere
    - **Metódus:** GET
    - **render:** page.ejs
  + **stb…**
* **admin.js**– adminisztátor funkciókhoz tartozó útvonalak kezelése
* **users.js -**felhasználókezeléshez tartozó útvonalak kezelése

**A Controller Layer** kezeli az alkalmazás üzleti logikáját. Ez azt jelenti, hogy az adatokat átalakítják vagy kiszámítják, hogy megfeleljenek az adatbázismodell követelményeinek, mielőtt elküldik őket a kiszolgálóra. A controller réteg a router layerbe lett integrálva.

**Adatkezelő réteg (Modells)**

Hozzáféréssel rendelkezik az adatbázishoz.

Az adatbáziskapcsolathoz szükséges paramétereket a **db/dbconfig.js** file tartalmazza:

const config = {

    connectionLimit: 10,

    password: 'root',

    user: 'root',

    database: 'keptar',

    host: 'localhost',

    port: 3306,

    dateStrings: true

  };

A telepítést követően a paramétereket helyi sajátosságoknak megfelelően kell beállítani.

A telepített verziónál ajánlatos egy korlátozott jogosultságokkal rendelkező felhasználó létrehozása a MySQL szerverben és ezzel a felhasználóval kell csatlakozni.

**host:** amennyiben az SQL szerver és a webkiszolgáló külön gépen kerül telepítésre, itt kell megadni a MySQL szervergép hostnevét

**user:** a MySQL szerverhez kapcsolódó felhasználó neve

**password:** MySQL user jelszava

**A réteg felelőssége:**

Az adatok lekérdezése, létrehozása, törlése vagy szerkesztése. Ez, az a réteg, ahol a szerver logikájából érkező összes kérést és választ kezeljük. Az adatelérési (Data Access Layer) metódusok változókat kapnak a controller rétegtől. Ezek a változók szükségesek az SQL lekérdezésekhez. Az SQL szervertől kapott válasz alapján ígéretet (Promise) küld, amely egy objektum.

Az objektum a szerver válaszának rendelkezésre állása után kerül meghatározásra. Ha a kérés sikeres, a DAL ígéretet egy objektum oldja fel, amely visszatér a szolgáltatási réteghez, amely maga is visszatér az útválasztási réteghez. Amikor az útválasztási réteg megkapja a szolgáltatási réteg által visszaadott objektumot, az útválasztási réteg JSON formátumban küldi el az objektumot a View rétegnek.

**A modul helye:** /db/dboperation.js

**View réteg:**

A html oldal előállításáért (render) **ejs template engine** a felelős.

A modul helye: **/views** mappa

A réteg felelőssége:

A JSON formátumban megkapott paraméterek és az ejs file-ban meghatározott sablon alapján előállítja a kimeneti HTML formátumot. A generált html oldal szövegét visszaadja a útválasztási rétegnek.

## Választott technológiák

Miért ezt választottam?

* Web-es technológiát azért választottam, hogy a rendszer egyszerre több száz felhasználót is ki tudjon szolgálni, probléma, vagy lassabb szolgáltatás nélkül
* Desktop alkalmazást egy szűkebb felhasználói kör éri el, pontosabban csak az adminisztrátorok. Ők akár egyszerre is elérheti az adatbázist.

**Desktop applikáció:**

* Fejlesztői környezet:
  + Visual Studio
  + MySQL for Visual Studio
* Programnyelv: C#
* Operációs rendszer: Windows
* Keretrendszer: .NET framework 4.8 Advanced Services
* ADO.NET programkönyvtár
* Connector/NET for MySQL 8.0.28 X86

**Web-es app:**

* Fejlesztői környezet:
  + NodeJS
* Programnyelv: Javascript
* Operációs rendszer: Windows / Linux

Könyvtárak:

* + bootstrap: 5.2.3
    - A Bootstrap keretrendszer nyílt forráskódú és ingyenes, segítségével a weboldalunk kinézetét egyszerűen és gyorsan elkészíthetjük. Olyan előre elkészített stílusokat, elrendezések tartalmaz, amelyeket saját CSS vagy JavaScript kód írása nélkül használhatunk.
  + cookie-parser: 1.4.4
    - A Cookie-parser egy olyan modul, amely kezeli a weboldalon használt sütiket. A sütik olyan kisméretű adatfájlok, amelyek lehetővé teszik a weboldalnak, hogy ha felhasználót bejelentkezve hagyják az eszközén, azért, mert az eszközön tárolja az ehhez szükséges adatokat.
  + corps: 0.1.0
  + cors: 2.8.5
    - Cross-Origin Resource Sharing (CORS) egy biztonsági mechanizmus, amely a webes támadásoktól védi meg a felhasználót. Egy weboldal akkor használja, ha más oldalakról is szeretne információt betölteni.
  + debug: 2.6.9
    - A Debug segít a fejlesztőknek, a hibakeresést végez az alkalmazásban. Segítségével a fejlesztők könnyen és rendszeresen megtalálhatják az általuk vétett hibákat a kódban.
  + ejs: 3.1.8
    - Az EJS (Embedded JavaScript) olyan sablonrendszer, amely lehetővé teszi az adatok dinamikus változtatását a weboldalon. Segítségével az oldalon dinamikusan lehet megjeleníteni, vagy változtatni az adatokat. Az EJS-nek köszönhetően lehet az oldal a felhasználó számára személyre szabott.
  + express: 4.16.1
    - Az Express egy olyan keretrendszer, amely rengeteg előre elkészített funkciót és modult tartalmaz. Ezek segítségével a fejlesztők egyszerűen létrehozhatnak új funkciókat saját alkalmazásukban.
  + express-session: 1.17.3
    - Az Express-session egy olyan modul, amely segít az alkalmazásoknak kezelni a felhasználói munkameneteket. Általában a felhasználói információk tárolására használják, például a bejelentkezés után elmenti a felhasználó adatait, hogy később az oldal könnyebben elérhesse azt.
  + fs: 0.0.1-security
    - Az fs a „file system” kifejezésre utal, ez a modul lehetővé teszi az alkalmazásnak, hogy fájlokhoz férjen hozzá a számítógépről.
  + http-errors: 1.6.3
    - Az http-errors modul egy Node.js könyvtár, amely lehetővé teszi a fejlesztők számára, hogy HTTP hibakódokat használjanak kivételek kezeléséhez a Node.js alkalmazásaikban.Ez azt jelenti, hogy az http-errors modul segítségével könnyedén dobhatunk kivételt a különböző HTTP hibakódokkal, például a 404-es vagy a 500-as hibakóddal kapcsolatosan.
  + https: 1.0.0
    - Az HTTPS (HyperText Transfer Protocol Secure) az HTTP protokoll biztonságosabb változata, amely titkosítja az adatokat a szerver és a kliens közötti kommunikáció során. Az HTTPS az SSL (Secure Sockets Layer) vagy a TLS (Transport Layer Security) protokollt használja a titkosított kommunikáció biztosításához. Az HTTPS titkosítja az adatokat, így a kommunikáció biztonságosabbá és védettebbé válik. Az HTTPS használata során a kliens és a szerver közötti kommunikáció titkosított csatornán keresztül történik, és csak a kommunikáló felek képesek olvasni az adatokat.
  + jquery: 3.6.3
    - A jQuery egy JavaScript könyvtár, amely megkönnyíti a webfejlesztőknek a dinamikus weboldalak fejlesztését. A jQuery segítségével a webfejlesztők egyszerűbben manipulálhatják a HTML elemeket és a CSS stílusokat, egyszerűbben hozhatnak létre animációkat, kezelhetik az eseményeket (például a kattintásokat és a beviteli mezők bevitelét), és könnyedén kommunikálhatnak a szerverrel AJAX segítségével.
  + morgan: 1.9.1
    - A morgan modul lehetővé teszi, hogy a fejlesztők az alkalmazás fejlesztése alatt naplózási információkat gyűjtsenek, mint például a beérkező kérések, azok státusza vagy időtartama.
  + multer: 1.4.5
    - A multer egy olyan eszköz, amely lehetővé teszi, hogy a felhasználó képeket, videókat vagy egyéb fájlokat töltsön fel a szerverre. A segítségével könnyedén kezelhető a fájlok mérete, típusa vagy a kiterjesztése, illetve a felmásolt fájl helyének megadása.
  + mysql: 2.18.1
    - A mysql modul lehetővé teszi a MYSQL adatbázisokkal való kommunikációt.
  + mysql2: 3.1.0
    - A mysql2 egy Node.js modul, amelyet MySQL adatbázisokkal való kommunikációhoz használnak. A mysql2 az eredeti mysql modulhoz képest hatékonyabb és több funkcionalitást kínál.A mysql2 a Node.js áramlás (stream) API-ját használja, ami hatékonyabb adatkezelést tesz lehetővé, különösen nagy adatmennyiségek esetén.
  + nodemon: 2.0.20
    - A nodemon egy Node.js alkalmazás, amely figyeli az alkalmazás fájljait, és újraindítja az alkalmazást minden alkalommal, amikor az egyik fájl módosul. A nodemon konfigurálható, és számos opciót és beállítást kínál, például a fájlok figyelési mappájának meghatározását, a figyelő ciklusának a beállítását, és az alkalmazás újraindításának módját.
  + path: 0.12.7
    - Az path modul hasznos segédfüggvényeket kínál a fájlok és mappák elérési útvonalainak kezeléséhez, fájlnevek és kiterjesztések feldolgozásához, elérési utak összekapcsolásához és ellenőrzéséhez, és sok más hasznos funkciót kínál.

## Függőségek

1. MySQL Server 8.0.31 adatbázis szerver
2. Hardver követelmény:
   1. Processzor: 1 GHz vagy gyorsabb processzor.
   2. RAM: 1 GB a 32-bites rendszerhez vagy 2 GB a 64-bites rendszerhez
   3. Háttértároló: 16 GB a 32-bites rendszerhez vagy 20 GB a 64-bites rendszerhez.
   4. Videókártya: DirectX 9 vagy újabb WDDM 1.0 driverrel.
   5. Megjelenítő: 1024 x 600 vagy nagyobb felbontás.
3. **Desktop app:**
   1. Operációs rendszer: Windows 10
   2. Keretrendszer: .NET framework 4.8 Advanced Services
4. **Web App:**
   1. Windows / Linux
   2. nodeJS
   3. package.json állományban lévő függőségek (telepítés: npm install)
   4. tanúsítvány, amennyiben van saját

# Modulok

A rendszer két modulból áll:

* Desktop alkalmazás
* Webkiszolgáló app

## Áttekintés

A szoftver architektúrája az alábbi komponensekre bomlik:

## Modulok részletei

Desktop alkalmazás – adminisztrációs modul

* Törzsadatok
  + Filmek
  + Besorolás
  + Kategóriák
  + KategóriaKapcsoló
  + Év
  + Hossz
  + Kedvenckapcsolás
  + Rendező
  + Színész
  + Színészkapcsoló
  + Felhasználók
* Kilépés

**Webes alkalmazás**

* Felhasználókezelés
  + Regisztráció
    - A felhasználók regisztrációjának kezelése.
  + Bejelentkezés
    - A regisztrált felhasználók bejelentkezésének megvalósítása.
  + Profil
    - A bejelentkezett felhasználók adatainak megjelenítése.
  + Kijelentkezés
    - A bejelentkezett felhasználók kiléptetése a weboldalról.
  + Kedvencek
    - A felhasználó hozzáadhat olyan filmeket a kedvenceinek, amelyek szimpatikusak számára, így könnyebben megtalálhatja őket, de könnyedén ki is veheti onnan. Ez a funkció csak a bejelentkezett felhasználók számára elérhető.
* Kereső
  + Ez a felső sávban található beviteli mező. Ide a felhasználható bármilyen információt megadhat a filmről, és ha azt tartalmazza az adatbázis megjeleníti.
* Szűrés
  + Ez a fajta keresés már sokkal szigorúbb. Itt a felhasználó kiválaszthatja, hogy mi alapján szeretne keresni. Ez a módszer kevésbé terheli a szervert, viszont a felhasználónak sokkal konkrétabbnak kell lennie
* Filmek megjelenítése
* Filmek adatai
  + Itt majdnem minden adatot megjelenítünk az adatbázisban szereplő filmekről.

# Perzisztencia

Adattárolás:

* MySQL adatbázis
* Képek tárolása a fájlrendszer mappájában

## A tárolás választott módja

MySQL:

MySQL egy relációs adatbázis-kezelő rendszer (RDBMS), amelyet adatok tárolására és lekérdezésére használnak. Gyakran használják webalkalmazásokban az adatok kezeléséhez és dinamikus tartalom biztosításához.

Fájlok, képek tárolása:

* Az alkalmazás erre a célra kijelölt mappákban történik:
  + Felhasználói profil képei: [alkalmazás mappája]\public\profil
  + A filmek képei: [alkalmazás mappája]\public\images

## Koncepcionális adatmodell

as

# Folyamatok

as

# Interfészek

asd

# Biztonság

as

# Üzemeltetést támogató képességek

as

## Konfiguráció

as

## Monitorozás

## Logolás

as

## Hibaelhárítás

as

Ábrajegyzék

[1. ábra 4](#_Toc129329446)

[**2**. ábra 5](#_Toc129329447)