

# Szoftverfejlesztés és -tesztelés vizsgaremek

**Szegedi Szent Benedek Technikum és Gimnázium Szegedi Tagintézménye Informatika és Távközlési ágazat**

**Szoftverfejlesztő és Tesztelő technikusi szakma**

Szakmai azonosító: 5-0613-12-03

# NextStream streamingszolgáltató weboldal

|  |  |
| --- | --- |
| Témavezető | Készítők, osztály |
| Bodrogi Péter Róbert | Tóth Zétény Tibor (13. A)  Szabó Gergő (13.A) |
|  | Forgács Levente (13.A) |
|  |  |

Szeged

2025

# Tartalomjegyzék

[**Szoftverfejlesztés és -tesztelés vizsgaremek 1**](#_heading=h.1f4y0ap8qik8)

[**NextStream streamingszolgálgató weboldal 1**](#_heading=h.ijyr0jbpqba8)

[**1. Tartalomjegyzék 2**](#_heading=h.ubi3xh1l48b2)

[**2. Dokumentáció 3**](#_heading=h.roz7k54y2crl)

[**3. Bevezetés 3**](#_heading=h.bc9ngjc08bnf)

[**4. Záródolgozat témája 4**](#_heading=h.vqbshbr8168b)

[**5. Záródolgozat témája 5**](#_heading=h.vqbshbr8168b)

[**5. Felhasználói dokumentáció 5**](#_heading=h.la0ur8udj940)

[A program általános specifikációja 5](#_heading=h.ucndtpu8di24)

[1. Bevezetés 5](#_heading=h.2srwrsupqhf0)

[2. Bejelentkezés és regisztráció 6](#_heading=h.wm0f86mhpbz9)

[3. Filmek és sorozatok böngészése 6](#_heading=h.onnfukqc6dw3)

[4. Filmek és sorozatok részletes információi 6](#_heading=h.eqxlu6nc148b)

[5. Streaming folyamat 6](#_heading=h.8vp3hsm1mur7)

[6. Fizetés és biztonság 7](#_heading=h.9jax3qdnzn1r)

[Rendszerkövetelmények 7](#_heading=h.ix4cos89lvrn)

[Minimum hardver követelmény 7](#_heading=h.vnoyupoaiokf)

[Minimum szoftver követelmény 7](#_heading=h.qepeapcq6tdz)

[A program telepítése 7](#_heading=h.i6wkelicigb)

[A forráskód letöltése 8](#_heading=h.7mi82hrqwq6w)

[Szükséges szoftverek letöltése 9](#_heading=h.gwtn6o5u5mdb)

[A program használatának részletes leírása 10](#_heading=h.pllo23aikkmd)

[**6. Fejlesztői dokumentáció 13**](#_heading=h.c97qx6b7qo1o)

[Témaválasztás indoklása 13](#_heading=h.31mqpd21ex7c)

[Alkalmazott fejlesztői eszközök 14](#_heading=h.qeg1vznmadm4)

[A fejlesztőkörnyezet hardvere 15](#_heading=h.dud0wjmmnddg)

[A fejlesztés során használt szoftverek 16](#_heading=h.k4sqks1bsnez)

[Web böngésző 14](#_heading=h.gb9s8q89cy34)

[VS Code 14](#_heading=h.3p1f813jzy4s)

[Cypress 15](#_heading=h.1dwxste2roi1)

[Választott fejlesztőkörnyezet és programnyelv 15](#_heading=h.c48i0lz17f70)

[Node.js 16](#_heading=h.ymypgd17fjs)

[React 16](#_heading=h.jn6jv49lyyr5)

[Részletes feladatspecifikáció, algoritmusok, forráskód 17](#_heading=h.8xr784fx8puu)

[1. Routes 17](#_heading=h.zf06qyn6ikuq)

[2. Controllers 19](#_heading=h.6nj3lpmwm34m)

[5. Models 22](#_heading=h.wzlanekmjk8)

[6. Server.js 25](#_heading=h.rltyiff86yc1)

[Továbbfejlesztési lehetőségek 27](#_heading=h.fjfjroysrbtd)

[Irodalomjegyzék, forrásmegjelölés 27](#_heading=h.jzxfzijf736b)

# 1. Dokumentáció

Projekt neve: NextStream streamingszolgáltató

Projekt felelősök: Tóth Zétény Tibor, Szabó Gergő, Forgács Levente

Projekt kezdete: 2024. 09. 01.

Kapcsolattartó email címek: [szabo.gergo@szbiszeged.hu](mailto:szabo.gergo@szbiszeged.hu), [forgacs.levente@szbiszeged.hu](mailto:forgacs.levente@szbiszeged.hu), [toth.zeteny@szbiszeged.hu](mailto:toth.zeteny@szbiszeged.hu)   
Iskola neve: Szent Benedek Technikum és Gimnázium Szegedi Tagintézménye

Képzés neve: Szoftverfejlesztő és -tesztelő

Témavezető neve: Bodrogi Péter Róbert

Weboldal profilja: Film és Sorozatnézés.

Weboldal főtevékenysége: Filmek, sorozatok böngészése, nézése.

Weboldal altevékenysége: Reklámozás

# Bevezetés

Ez a dokumentáció azért készült, hogy bemutassa az általunk készített weboldal funkcióit, valamint segítséget nyújtson a felhasználóknak az oldal használatában.

Az film és sorozatnézés egy olyan terület, amely számos ember számára fontos és gyakran használt szolgáltatás. Ezért úgy döntöttünk, hogy az általunk készített programot ennek a területnek a megkönnyítésére tervezzük.

# Záródolgozat témája

A záródolgozatunk témája egy film és sorozatkölcsönző elkészítése, ahol ezek közül tudnak böngészni az ügyfelek.

Témaválasztás okai:

* Valós élethelyzetre kínál megoldást
* Hasznos, életszerű, sokakat érintő téma
* Közös érdeklődési kör
* Megvalósítható, megfelelő komplexitású projekt

A program funkciói:

* Film vagy Sorozat kiválasztása
* Válogatás az elérhető filmek vagy sorozatok között
* Regisztráció
* Bejelentkezés
* Oldalak közötti váltás
* Avatár és jelszó változtatás
* Kijelentkezés
* Előfizetés egy adott csomagra
* Részletek megtekintése

**Célközönség:**

A téma bárkit érdekelhet, viszont az oldalunk célközönsége kortalan, hisz a filmnézés, és a sorozatok nézése kortól független.

Az oldal egyszerű használatának köszönhetően azok a felhasználók is könnyen megtalálják az információkat, akik még nem rendelkeznek tapasztalattal a téma terén.

Ebben a dokumentációban részletesen ismertetjük az oldalunk funkcióit, így reméljük, hogy hasznos információkat nyújtunk a felhasználók számára a folyamat egyszerűsítése érdekében.

# 2. Felhasználói dokumentáció

## A program általános specifikációja

### Bevezetés

A NextStream webalkalmazás egy modern és intuitív platformot kínál a felhasználóknak, amely lehetővé teszi számukra a filmek és sorozatokról való tájékozódást. Az alkalmazás felhasználóbarát felületet és sokoldalú funkciókat kínál, hogy megkönnyítse az ügyfelek számára a megfelelő jármű kiválasztását, foglalását és bérlését. Az alábbiakban bemutatjuk a program legfontosabb jellemzőit és funkcióit, hogy a leendő felhasználók el tudják dönteni, hogy a program megfelelő-e számukra.

### Bejelentkezés és regisztráció

A webalkalmazás használata során az ügyfelek személyre szabott élményben részesülnek. Az alkalmazás lehetővé teszi a felhasználók számára, hogy regisztráljanak és bejelentkezzenek egy biztonságos fiókba. A regisztráció során a felhasználóknak meg kell adniuk alapvető személyes adataikat, mint például e-mail cím és jelszó. A sikeres regisztráció után a felhasználók bejelentkezhetnek a fiókjukba, és elkezdhetik a böngészést.

### Filmek és Sorozatok böngészése.

A webalkalmazás széles választékot kínál a különböző kategóriájú filmekből és sorozatokból, hogy minden felhasználó megtalálhassa a számára leginkább megfelelőt.

### Filmek és Sorozatok részletes információi

Az alkalmazás mindenről részletes információkat kínál, amelyek segítenek a felhasználóknak a döntésben.

Ezenkívül a felhasználók megtekinthetik a szereplőket, és olvashatnak más ügyfelek által hagyott értékeléseket. Mindez lehetővé teszi a felhasználók számára, hogy alaposan megismerjék az a filmet, vagy sorozatot amelyről érdeklődik.

### Streaming folyamat

Ha a felhasználó megtalálta a számára megfelelő filmet vagy sorozatot, a folyamat egyszerűen és gyorsan lezajlik. Az alkalmazás áttekinthető és könnyen érthető lépésekben mutatja be a folyamat menetét.:

* A felhasználó kiválasztja a számára tetsző sorozatot vagy filmet és tovább megy az oldalra.
* A felhasználó megtekinti az előfizetés oldalt, előfizet a szolgáltatásra. A rendszer visszaigazolást küld az előfizetésről, az előfizetéssel megtekintheti az adott filmet.

### Fizetés és biztonság

Webalkalmazásunk megvalósít online fizetési lehetőségeket. A fizetés minden esetben személyesen a weboldalon történik ezért kötelezettségmentesen bármikor vissza lehet mondani, ezzel a felhasználókat védve.

## Rendszerkövetelmények

### Minimum hardver követelmény

* Operációs rendszer: Windows 7, MacOS 10.12 (Sierra) vagy újabb, Linux (Ubuntu 16.04 vagy újabb) - a legfrissebb biztonsági frissítésekkel.
* Processzor: Legalább 1 GHz-es órajelű processzor (egy vagy többmagos).
* Memória (RAM): Legalább 2 GB RAM.
* Grafikus kártya: Legalább DirectX 9.0c kompatibilis integrált vagy dedikált grafikus kártya.
* Internet kapcsolat: Szélessávú internetkapcsolat (legalább 1 Mbps letöltési és feltöltési sebesség).

Ezenkívül ajánlott egy modern, frissített webböngésző használata, mint például:

* Google Chrome (legújabb verzió)
* Mozilla Firefox (legújabb verzió)
* Apple Safari (legújabb verzió)

### Minimum szoftver követelmény

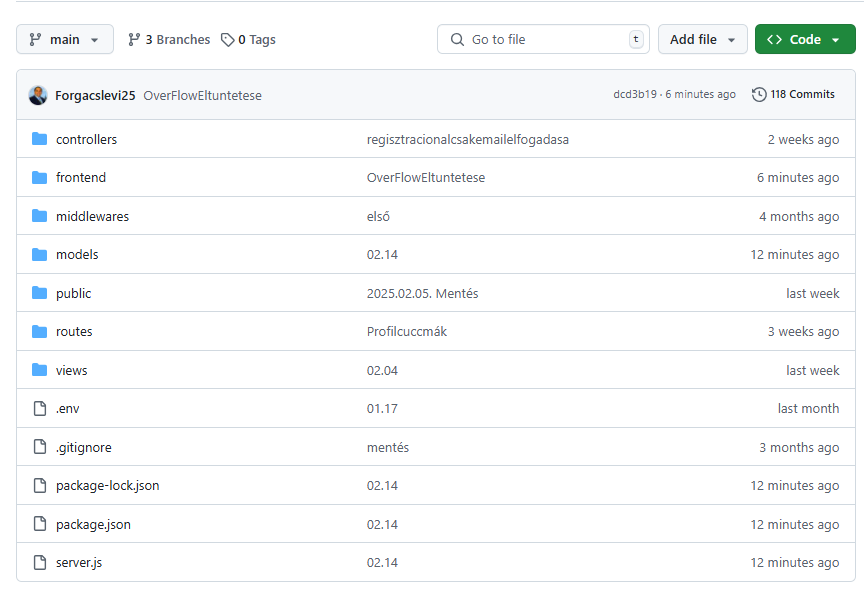
* Operációs rendszer: Windows 7, 8.0, 8.1, 10, 11, MacOS 10.12 (Sierra) vagy újabb, Linux (Ubuntu 16.04 vagy újabb)
* Fejlesztői környezet: Visual Studio Code
* Szükséges programok: AMPPS, adatbázis dump fájl
* Frontend: ReactJS
* Backend: NodeJS

## A program telepítése

### A forráskód letöltése

A program telepítésének első lépése, hogy letöltjük az összes szükséges fájl és forráskódot.

Minden szükséges fájlt ezen az oldalon találunk meg:

<https://github.com/Takigyula/Filmkolcsonzo>

Itt a zöld “code” feliratú gombra kattintva megjelenik egy lehetőség, miszerint le tudjuk tölteni a projektet zip állományban. Ezt a lehetőséget válasszuk ki és várjuk meg amíg minden letöltődik.

1. **Forráskód letöltése**

Miután letöltődött a projektünk, nyissuk meg a fájlkezelőben, a letöltések mappában lesz. Itt jobb egérgombbal tudunk a mappára kattintani, ezután a “Kibontás” lehetőséget választva meg tudjuk adni, hova csomagoljuk ki az állományt. Ha sikeresen végrehajtottuk ezeket a lépéseket, akkor a programunk készen áll a megnyitásra.

### Szükséges szoftverek letöltése

A következő lépés a Visual Studio Code fejlesztői környezet letöltése. Erről a linkről lehet letölteni: <https://code.visualstudio.com/>

Elérhető windows, max és linux rendszereken is a program. Először ki kell választanunk melyik rendszerre szeretnénk telepíteni, majd a hivatalos weboldalon lévő telepítési útmutatót kell követnünk.

A következő linkeken elérhetjük a hivatalos telepítési útmutatókat.

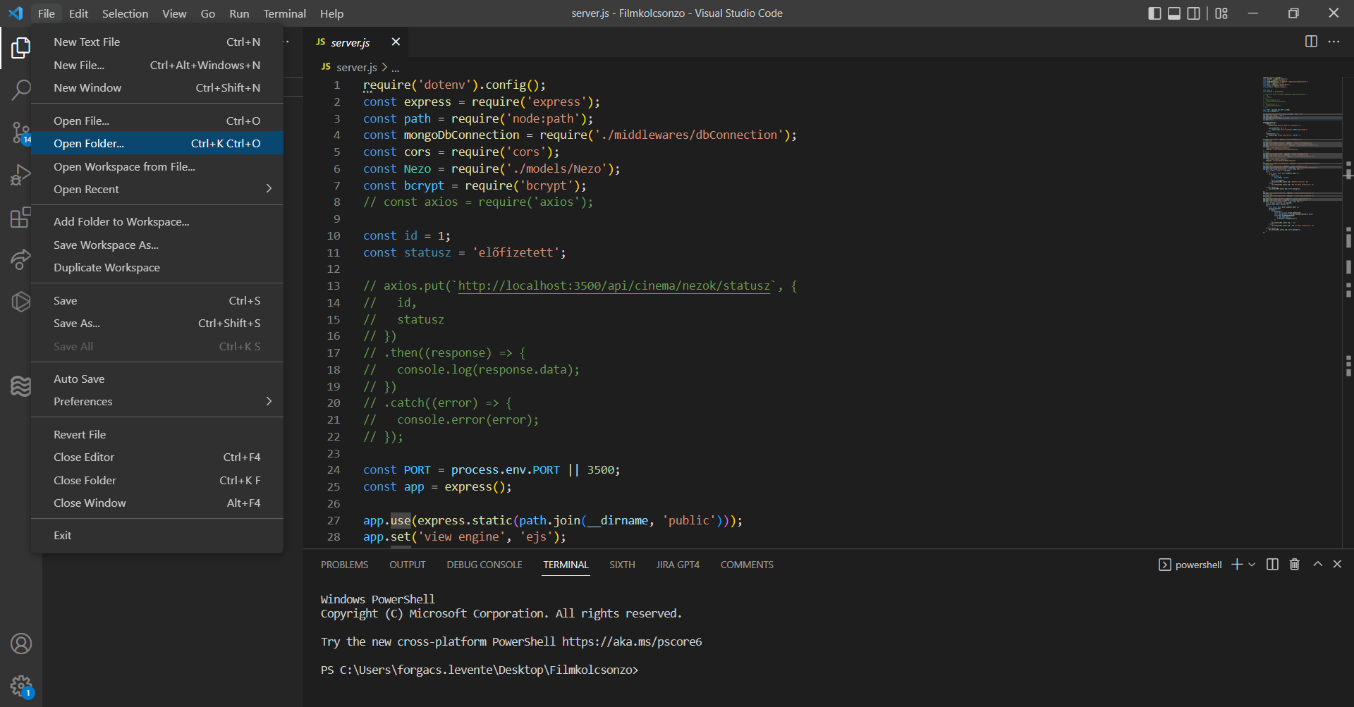
* Windows: <https://code.visualstudio.com/docs/setup/windows>
* Linux: <https://code.visualstudio.com/docs/setup/linux>
* MacOS: <https://code.visualstudio.com/docs/setup/mac>

Ha a Visual Studio Code is sikeresen feltelepült a gépre, akkor a következő lépésben a NodeJS-t kell letölteni, erről az oldalról: <https://nodejs.org/en/download>.

Itt szintén ki kell választanunk az operációs rendszerünknek megfelelő letöltési módot és aszerint eljárni.

Az utolsó lépés az adatbázist biztosító programunk a MongoDB, ez egy böngészőre alapuló adatbázis, ezért ezt nem kell letöltenünk.

Következő lépésként meg kell nyitnunk a Visual Studio Code-t a “file” menüpontban az “open folder” vagyis “fájl-mappa megnyitása” opciót válasszuk és nyissuk meg a letöltött, kicsomagolt projekt mappát.



1. **A VS Code elindítása**

Ezt követően a terminálra lesz szükség. Az alsó sávon is meg tudjuk nyitni, de ha ezt a lehetőséget nem látjuk akkor a “ctrl+ö” billentyűkombinációval meg lehet nyitni. Navigáljunk el a projekt frontend mappájába ezzel a paranccsal: “cd Filmkolcsonzo/Frontend”

Itt adjuk ki az “npm install” parancsot, ez azért szükséges hogy az összes csomagot és modult feltelepítse, ami szükséges a program működéséhez. Miután ezzel megvagyunk adjuk ki az “npm start” parancsot. Ezzel elindítjuk a weboldalt. Meg fogja kérdezni, hogy használjuk e az alapértelmezett böngészőt erre a célra, írjunk be egy “y”-t majd nyomjunk entert.

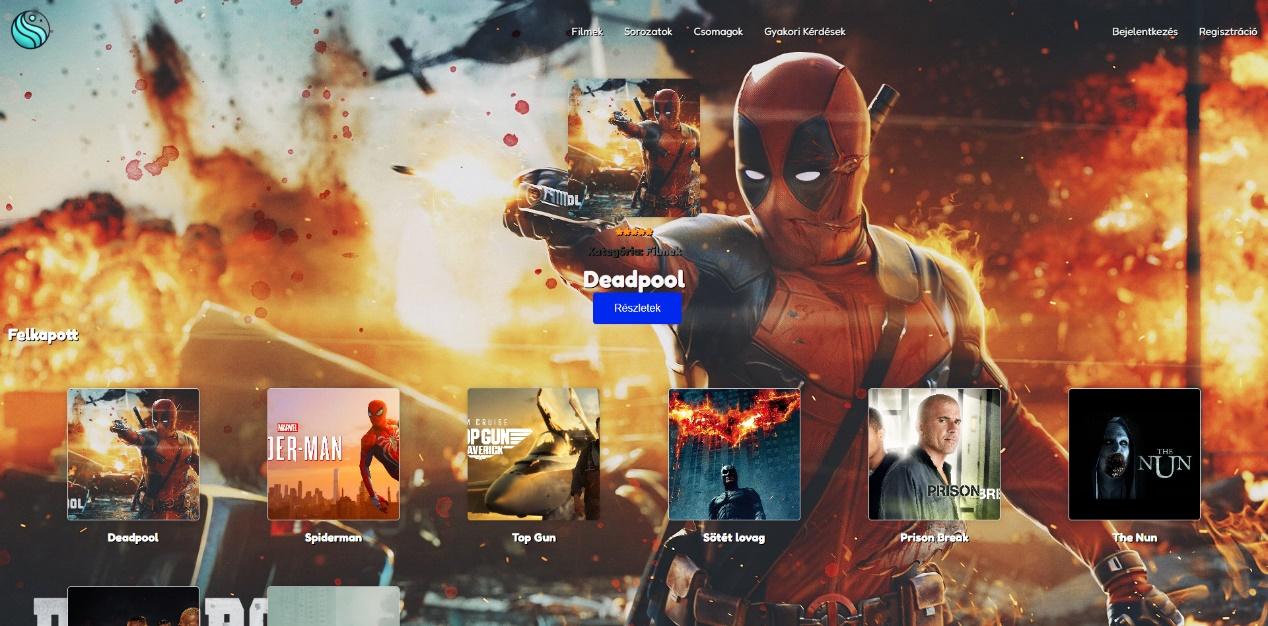
Ezt követően adjuk ki a “cd ..” parancsot, hogy vissza lépjünk egyet a mappa struktúrában. Majd kiadjuk a “cd Backend” parancsot, amivel belépünk a backend oldalra. Itt ugyanúgy szükséges az “npm install” parancs kiadása.

Ha erre hibaüzenetet kapnánk, akkor ezt írjuk be: “npm install –legacy-peer-deps” (dupla kötőjel a legacy szó előtt).

Ha ezzel megvagyunk, akkor “npm run dev” paranccsal elindul a backend szerverünk, hozzá csatlakozik az adatbázishoz és már meg is tudtuk nyitni az oldalt.

## A program használatának részletes leírása

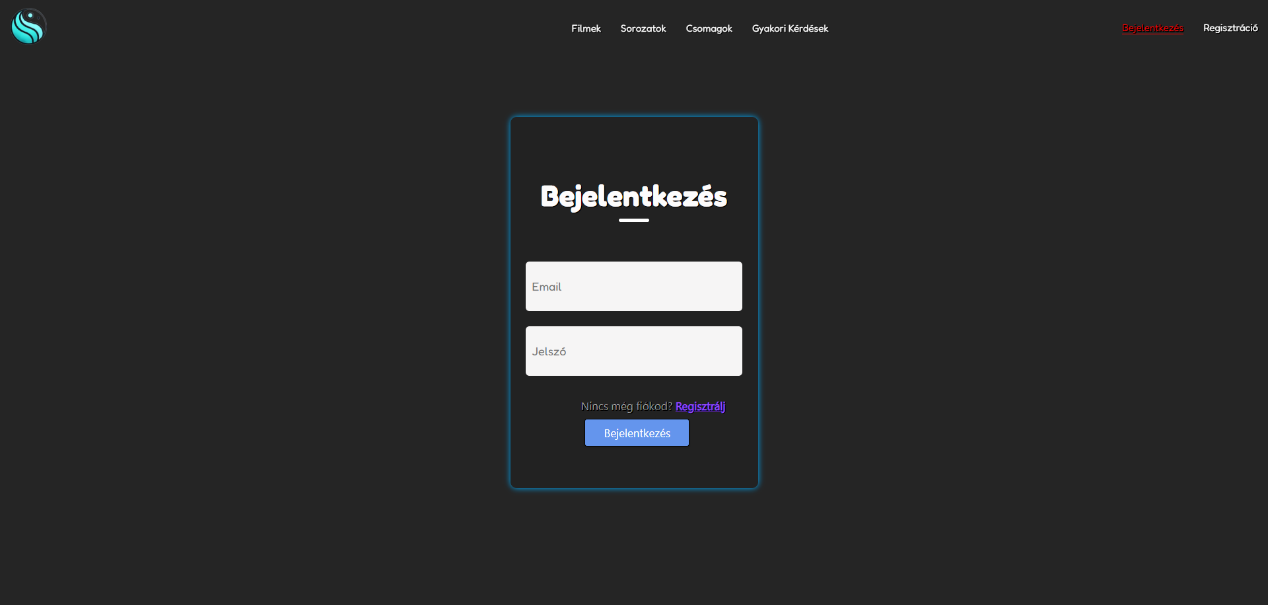
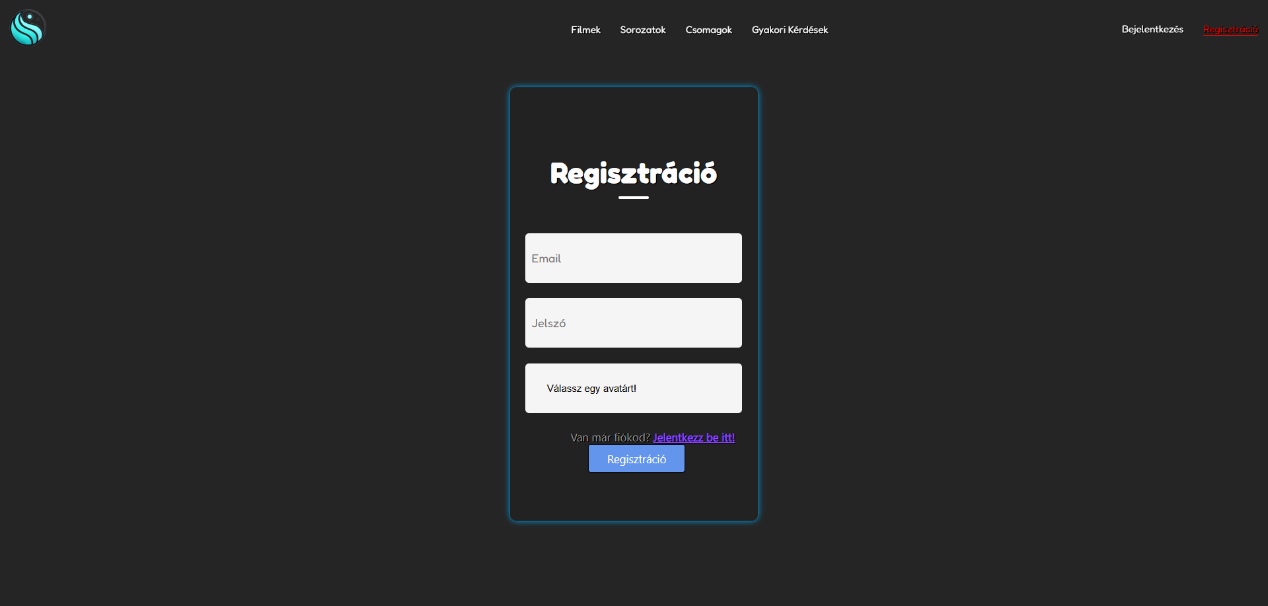
A weboldalunk megnyitása után a kezdőképernyő fogad minket. Itt láthatjuk a felső navigációs sávot, amin keresztül tudunk az oldalak között navigálni. Képünk egy korábbi verziót mutat be, a szöveg elrendezését módosítottuk azóta.

1. **Főoldal**

Ez a főoldal. Láthatunk egy alap filmet, és annak plakátját.

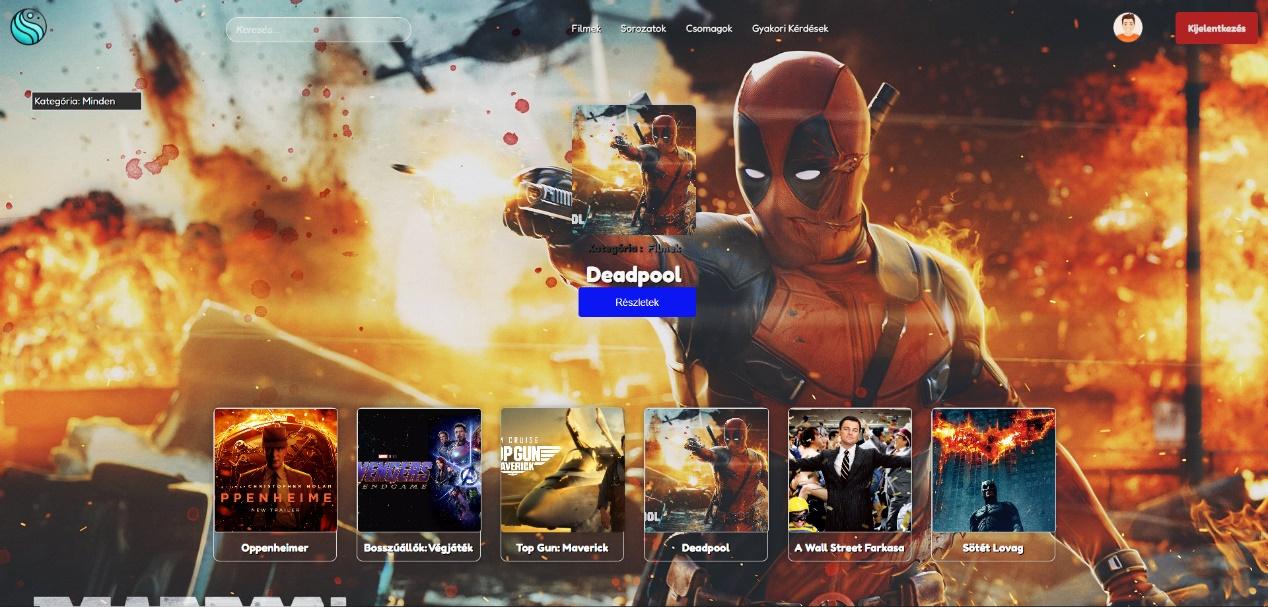
A “Gyakori Kérdések” -re kattintva egy kérdések oldalra kerül a felhasználó, ahol gyakori kérdéseket olvashat, például hogy miért nem tud belépni a fiókba.

A szolgáltatások használatához regisztráció és bejelentkezés szükséges. A felhasználó az alábbi képeken látható módon tudja ezt megtenni. A felső navigációs sávon a “Bejelentkezés” gombra kattintva be tud jelentkezni a felhasználó, és van lehetősége átmenni a regisztrációs oldalra is.

1. Bejelentkezés
2. Regisztráció

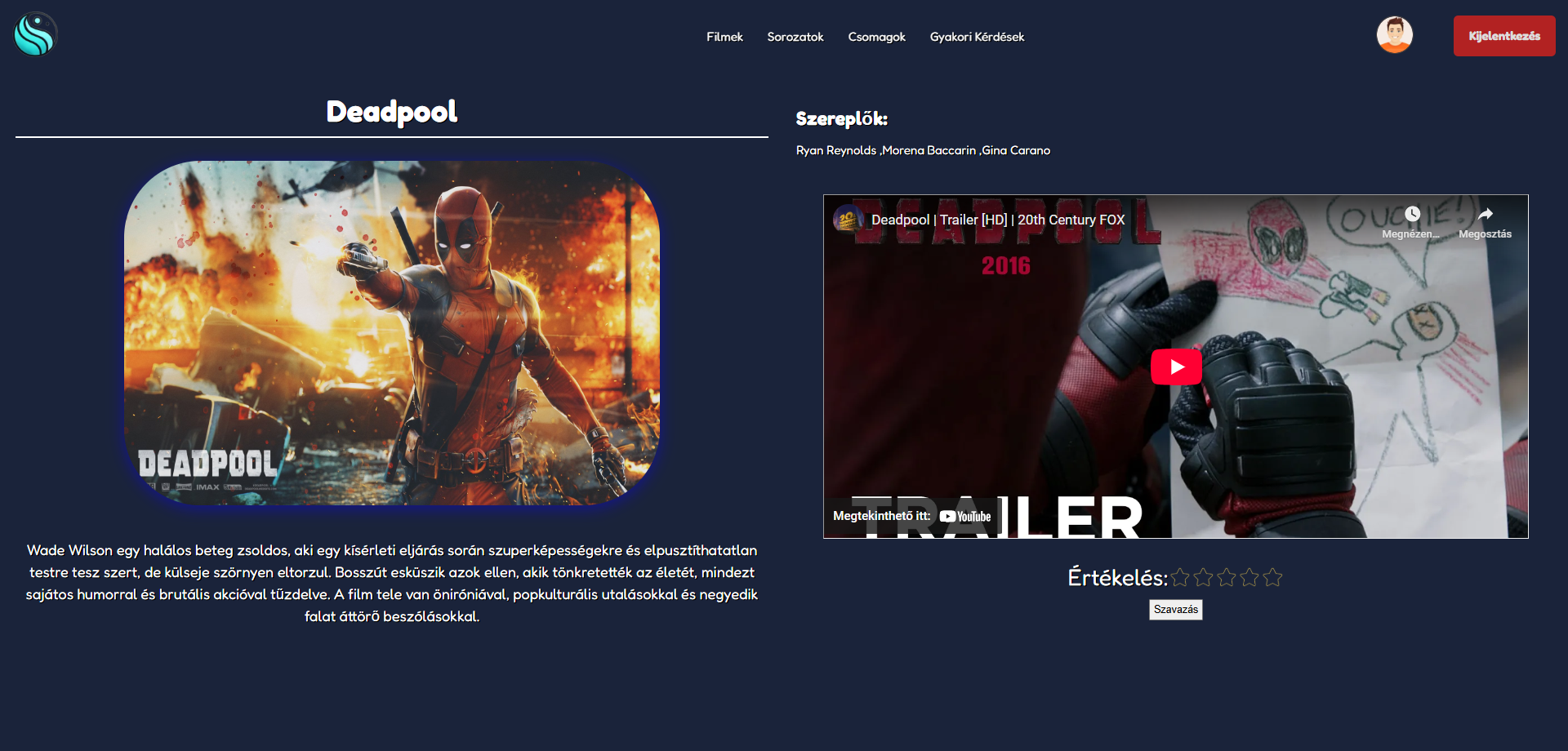
Az filmek és sorozatok böngészése bejelentkezés nélkül is lehetséges. Az alábbi képen láthatjuk, amik közül tudnak választani a felhasználók.

1. Elérhető Filmek

Ha a felhasználó úgy dönt, hogy egy adott filmet megszeretne tekinteni, a filmek gombra kattintva böngészhet meglévő filmeink között, a részletek fülre való kattintással megtekintheti annak részleteit, szereplőit, leírását, értékelheti az adott filmet, vagy sorozatot.

1. Leírás megtekintése.

Ha a felhasználó megszeretné nézni a filmet, elolvashatja annak leírását, megtekintheti a hivatalos magyar nyelven elkészített előzetest, és a végén értékelheti a filmet, amely megjelenik a főoldalon, és hozzáadódik az érték az átlaghoz, amelyet megadott.



1. Gyakori kérdések oldal

A Gyakori Kérdések oldalra kattintva általános kérdéseket vélhetünk felfedezni az adott témával kapcsolatban, a kérdésre kattintva kiadja a választ, amely megválaszolja az adott kérdést, melyre kattintottunk.

# 6. Fejlesztői dokumentáció

## Témaválasztás indoklása

A záródolgozatunk témájának egy filmekkel és sorozatokkal foglalkozó weboldalt választottuk, mivel úgy gondoljuk, hogy ez a terület számos lehetőséget nyújt a technológia, üzleti modell és felhasználói élmény vizsgálatára. A dolgozat célja, hogy bemutassa a weboldal működését, üzleti modelljét, és javaslatokat tegyen az ágazat továbbfejlesztésére.

Az iparág folyamatosan növekszik és fejlődik. A globalizáció és a technológiai fejlődés következtében az ilyesfajta cégeknek meg kell felelniük a változó ügyfél-igényeknek és piaci trendeknek. Az online jelenlét és a digitális szolgáltatások egyre fontosabbá válnak ebben a versenyképes iparágban.

Az ilyesfajta weboldalak működése számos szempontból érdekes. Az információtechnológia és a webfejlesztés területén a felhasználóbarát és hatékony felületek kialakítása kulcsfontosságú az ügyfelekkel való kapcsolattartás és a foglalások növelése érdekében. Emellett az üzleti modell elemzése lehetővé teszi, hogy feltárjuk a jövedelemszerzési és költségcsökkentési stratégiáit.

A záródolgozatunkban kiemelt figyelmet szenteltünk a felhasználói élmény elemzésének és a fejlesztési lehetőségek feltárásának. Az ilyesfajta weboldalaknak a felhasználók számára könnyen érthető, gyors és kényelmes módon kell nyújtaniuk a szolgáltatásaikat. A tervezés, a navigáció és az információstruktúra optimalizálása révén javítható a felhasználói élmény és növelhető az ügyfél-elkötelezettség.

A weboldal témaválasztása lehetővé teszi, hogy átfogó képet kapjunk az iparág működéséről és fejlesztési irányairól. A dolgozat elkészítésével célunk, hogy hozzájáruljunk az iparág fejlődéséhez, és értékes ismereteket szolgáltassunk a szakembereknek és érdeklődőknek egyaránt.

## Alkalmazott fejlesztői eszközök

## A fejlesztőkörnyezet hardvere

A fejlesztőkörnyezet hardvere a számítógépes rendszer, amelyet mi, fejlesztők használtunk a programok, alkalmazások és egyéb szoftveres projektek létrehozásához, használatához és teszteléséhez. A hardvernek a fejlesztési folyamat során a szükséges teljesítményt és stabilitást kell biztosítania, hogy a fejlesztők hatékonyan és zökkenőmentesen dolgozhassanak.

## A fejlesztés során használt szoftverek

A fejlesztés során használt szoftverek olyan eszközök, amelyek segítik a fejlesztőket a kódolás, tesztelés, hibakeresés és egyéb feladatok elvégzésében. Az alábbiakban bemutatunk néhány gyakran használt szoftvert, amelyeket a webalkalmazásunk létrehozásakor és tesztelésekor alkalmaztunk.

### Web böngésző

A webböngészők, mint például a Google Chrome, Mozilla Firefox, Safari, Brave vagy Microsoft Edge, lehetővé tették, hogy megtekinthessük és tesztelhessük a webalkalmazásunkat a felhasználók szemszögéből. A modern web böngészők fejlett fejlesztői eszközöket is tartalmaznak, amelyek hozzájárultak a hibakereséshez, teljesítményméréshez és az optimalizációhoz.

### VS Code

A Visual Studio Code (VS Code) egy népszerű, ingyenes és nyílt forráskódú szövegszerkesztő, amelyet a Microsoft fejlesztett ki. A VS Code számos kiterjesztést és beépülő modult támogat, amelyek segítségével könnyen tudtuk kezelni a különböző nyelveket és keretrendszereket. A VS Code beépített Git támogatást, hibakeresőt és intelligens kód kiegészítést is kínál.

### Cypress

A Cypress egy népszerű, modern end-to-end (E2E) tesztelési keretrendszer, amelyet webalkalmazások tesztelésére fejlesztettek ki. Különösen elterjedt a front-end fejlesztők és tesztelők körében, mivel könnyen beállítható, gyors, és hatékony eszközöket biztosít a böngészőalapú teszteléshez.

## Választott fejlesztőkörnyezet és programnyelv

A Visual Studio Code (VS Code) fejlesztői környezet és a React + Node.js programnyelvek választásának több indoka is volt a program készítése során.

### VS Code

* Könnyű használat és gyors betanulás: A VS Code egy könnyen használható és intuitív felületet kínál, amely a fejlesztők számára gyorsan elsajátítható.
* Bővítmények: A VS Code számos bővítményt támogat, amelyek segítségével a fejlesztők könnyen kezelhetik a különböző nyelveket és keretrendszereket, mint például a React és Node.js.
* Beépített Git támogatás: A VS Code beépített Git támogatást kínál, amely lehetővé teszi a fejlesztők számára, hogy könnyen nyomon követhessék a változásokat, összehasonlíthassák a kódverziókat és kezelhessék a verziókövetést.
* Hibakeresés és kódkiegészítés: A VS Code beépített hibakeresőt és intelligens kódkiegészítést is tartalmaz, amely javítja a fejlesztők hatékonyságát és produktivitását.

### Node.js

* Egyetlen nyelv a kliens- és szerver oldalon: A Node.js lehetővé teszi a fejlesztők számára, hogy JavaScriptet használjanak a szerveroldali fejlesztéshez, ami egységesíti a fejlesztési folyamatot és csökkenti a tanulási görbét.
* Aszinkron és eseményvezérelt: A Node.js aszinkron és eseményvezérelt architektúrája kiválóan alkalmas nagy teljesítményű és skálázható alkalmazások létrehozására, amelyek nagyszámú kapcsolatot és nagy adatáramlást kezelnek.
* NPM: A Node.js-hez tartozó NPM (Node Package Manager) rendelkezik a legnagyobb ökoszisztémával a nyílt forráskódú könyvtárak között, amelyek segítik a fejlesztőket a gyors és hatékony fejlesztésben.
* Könnyű integráció: A Node.js könnyen integrálható más népszerű technológiákkal és keretrendszerekkel, például a React-tel, amely tovább erősíti a teljes fejlesztési folyamatot.

Használt Node.js modulok:

* React
* Vite

### React

* Komponensalapú: A React komponensalapú architektúrája lehetővé teszi a fejlesztők számára, hogy újrafelhasználható és karbantartható kódokat hozzanak létre, amelyek elősegítik a gyors és hatékony fejlesztést.
* Gyors és optimális teljesítmény: A React virtuális DOM-mal rendelkezik, amely javítja az alkalmazás teljesítményét és optimalizálja a DOM-frissítéseket.
* Közösségi támogatás és érett ökoszisztéma: A React egy nagy és aktív közösséggel rendelkezik, valamint számos könyvtárat és eszközt kínál, amelyek segítik a fejlesztőket a gyors és hatékony fejlesztésben.

## 

## Részletes feladatspecifikáció, algoritmusok, forráskód

Ez a fejezet a program lényeges függvényeinek és az osztályok metódusainak a specifikációját fogja bemutatni. Ez a rész leginkább a backend oldali fejlesztést foglalja össze, amely Node.Js használatával valósult meg.

A backend oldali kód három fő része

* routes
* models
* controllers

### Routes

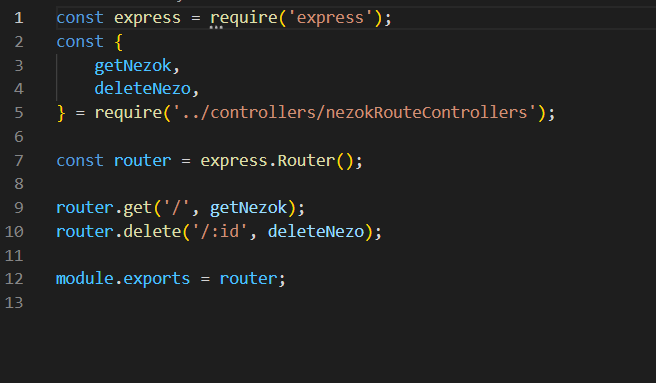
A projekt backend oldali tartalmának része a *routes* mappa. Ezen a mappán belüli fájlokkal definiáljuk az útvonalakat (routes), amelyeken keresztül az alkalmazásunk különböző kéréseket (request-eket) fogadhat el.

Egy Node.js alapú alkalmazásban az útvonalak általában a HTTP kérések kezelésére szolgálnak, és a "routes" mappában található fájlok a kérések típusaihoz (GET, POST, PUT, stb.) és a hozzájuk tartozó útvonalakhoz kapcsolódnak.

A "routes" mappa használata rendkívül fontos a fejlesztési folyamat során, mivel segít az alkalmazás funkcionalitásának szervezésében és könnyű karbantartásában. A különböző útvonalak elkülönítése a mappákban lehetővé tette számunkra, hogy könnyen megtaláljuk és frissítsük az egyes funkciókat. Ezen kívül lehetővé tette a kódbázisunk modularizálását, ami javította a fejlesztési folyamat hatékonyságát és átláthatóságát.

**A routes mappában lévő fájlok a következők**

* csillagModositRoutes.js
* egyediFilmModositRoutes.js
* egyediFilmRoutes.js
* egyediSorozatModositRoutes.js
* egyediSorozatRoutes.js
* filmekRoutes.js
* foglalasokRoutes.js
* loginRoutes.js
* mainRoutes.js
* nezokFrontRoutes.js
* nezokRoutes.js
* nezokStatuszRoutes.js
* registerRoutes.js
* sorozatokRoutes.js
* ujFilmRoutes.js
* ujsorozatRoutes.js

Ezen a képen a nezokRoutes.js route fájlját láthatjuk.  
2. nezokRoutes.js

A nezokRoutesController egy útválasztóvezérlő, amely a nézőkkel kapcsolatos kérések kezeléséért felel. Meghatározza az egyes útvonalakhoz tartozó kezelőfüggvényeket is. Például a "/nezok" útvonalon érkező GET kérés esetén a getNezok függvény fut le, amely visszaadja az összes néző adatait.

Az útvonalak dinamikussá tehetők paraméterek használatával. Például a "/nezok/:id" útvonalban az "id" egy dinamikus paraméter, amelyet a kérés során megadott azonosítóval helyettesítünk.

Az útválasztó végül exportálja az express.Router() objektumot, így más fájlokban is felhasználható. Általában a fő alkalmazásfájlban (server.js) importáljuk, hogy az útválasztó az egész alkalmazásban elérhető legyen.

### Controllers

A *controller* mappa az MVC (Model-View-Controller) tervezési minta egyik része. A minta az alkalmazás különböző részeit logikailag elkülöníti egymástól, hogy azok jól karbantarthatóak legyenek.

A *controller* az MVC modellben a vezérlő réteg, amely a HTTP kéréseket kezeli. A vezérlő réteg felelős a kérések feldolgozásáért és az adatok megfelelő megjelenítéséért a nézetben.

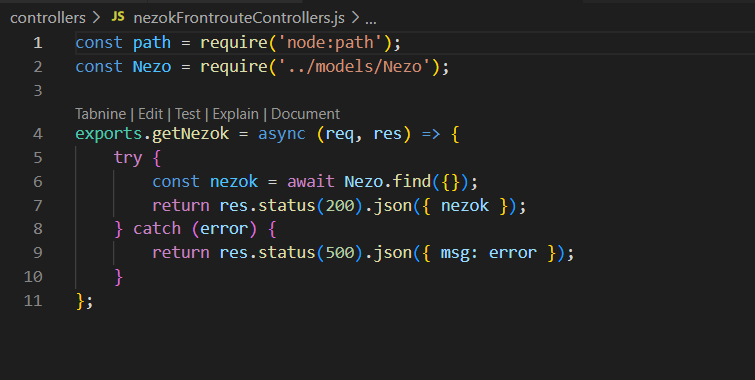
Használatával az alkalmazás vezérlő funkcióit külön fájlokban szervezzük. Ez javítja az alkalmazás karbantarthatóságát, mivel különválasztja az alkalmazás különböző funkcióit.

A vezérlők fájljai az alkalmazásban a HTTP kérések kezelő funkcióit tartalmazzák. Ezek a funkciók kapják meg a kéréseket, feldolgozzák azokat, majd visszaadják a megfelelő válaszokat. Például egy "GET" kérés esetén a vezérlő a szükséges adatokat elkéri az adatbázisból, majd visszaadja azokat json formátumban, amit később a frontenden jelenítünk meg. A *controller* mappában egy fájl van az egyes vezérlők számára. Ezek a vezérlő fájlok rendszerint az útválasztó fájlban (routes) vannak importálva, hogy azokat használni lehessen a megfelelő kérések kezelésére.

A *controller* mappa használata javítja az alkalmazás moduláris felépítését és csökkenti az egyes fájlok méretét, amelyek így egyszerűbben karbantarthatóak.

**A controller mappában lévő fájlok a következők**

* csillagModositRouteControllers.js
* egyediFilmModositRouteControllers.js
* egyediFilmRouteControllers.js
* egyediSorozatModositRouteControllers.js
* egyediSorozatRouteControllers.js
* filmekRouteControllers.js
* foglalasokRouteControllers.js
* loginRouteControllers.js
* mainRouteControllers.js
* nezokFrontrouteControllers.js
* nezokRouteControllers.js
* nezokStatuszRouteControllers.js
* RatingController.js
* registerRouteControllers.js
* sorozatokRouteControllers.js
* ujFilmRouteControllers.js
* ujsorozatRouteController.js

1. nezokRouteControllers.js - controller (1)
2. nezokFrontrouteControllers.js - controller (2) 

**A kód általános leírása**

A fenti kód egy vezérlőt (controller) definiál, amely a Nezo entitással kapcsolatos HTTP kéréseket kezeli.

Az getNezok funkció egy GET kérést kezel, amely visszaadja az összes Nezo adatot.

**Az getNezok funkció:**

Egy GET kérést kezel.

Visszaadja az összes Nezo adatot.

Egy callback függvényt használ, amely visszaküldi az eredményt a kliensnek.

Ha hiba történik, akkor egy hibaüzenetet ad vissza.

**A használt objektumok**

A res objektum a HTTP kérés válasz objektuma, amely tartalmazza a válasz szövegét, státusz kódját és egyéb információkat.

Az error objektum az esetleges hibákat tartalmazza.

**A kód összefoglalása**

Ez a kód egy adatmodellt használ, amely az adatok eléréséért és módosításáért felelős, és egy vezérlőt definiál, amely a különböző HTTP kéréseket kezeli.

### Models

A *models* mappa az alkalmazás modell rétegének része. Ez a réteg az adatokkal kapcsolatos műveleteket végzi.

A *models* mappa használata azt jelenti, hogy az alkalmazás adatmodelleit külön fájlokban szervezzük. A fájlokban a modellekhez kapcsolódó adatok elérésére és módosítására szolgáló funkciók találhatók.

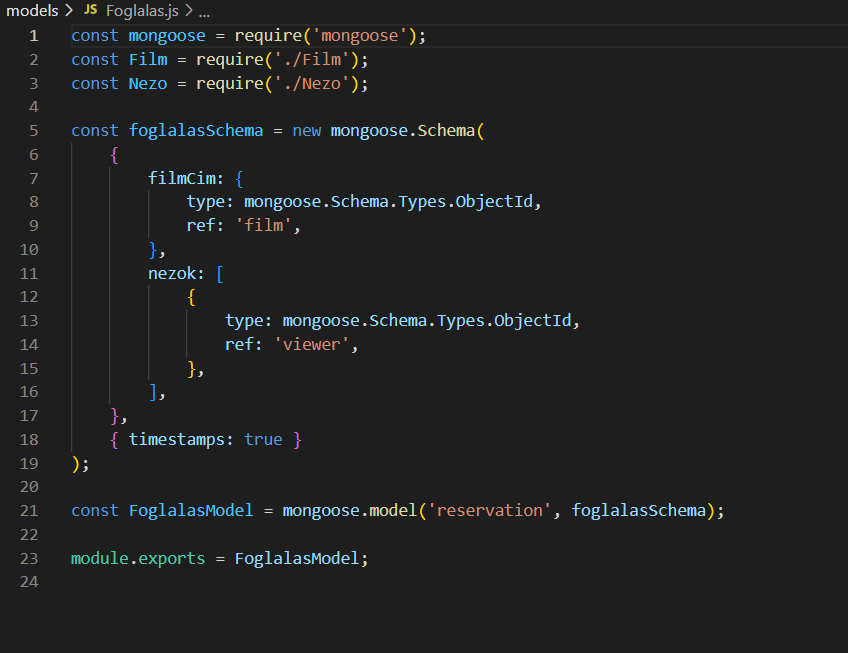
A modell fájlok általában adatbázisokhoz kapcsolódnak, esetünkben MySQL-hez. Ezek a funkciók lehetővé teszik az alkalmazásnak, hogy adatokat hozzon létre, módosítson, olvasson és töröljön az adatbázisból.

A *models* mappa használata javítja az alkalmazás struktúráját és könnyen karbantarthatóvá teszi a funkciókat.

Az adatmodell fájlok általában egyszerűen használhatók az alkalmazás többi részében, így az adatok elérésére és módosítására szolgáló funkciók szétszórt kódjait egy helyen tartjuk.

Az adatmodell fájlok gyakran importálják az adatbázis-kezelő modulokat, mint például a *mysql2*. Ezek a modulok lehetővé teszik az alkalmazás számára, hogy kapcsolatot létesítsen az adatbázissal és végrehajtsa az adatbázis-műveleteket.

A models mappában lévő fájlok a következők

* api.js
* Foglalas.js
* Film.js
* Nezo.js
* rating.js
* sorozat.js 

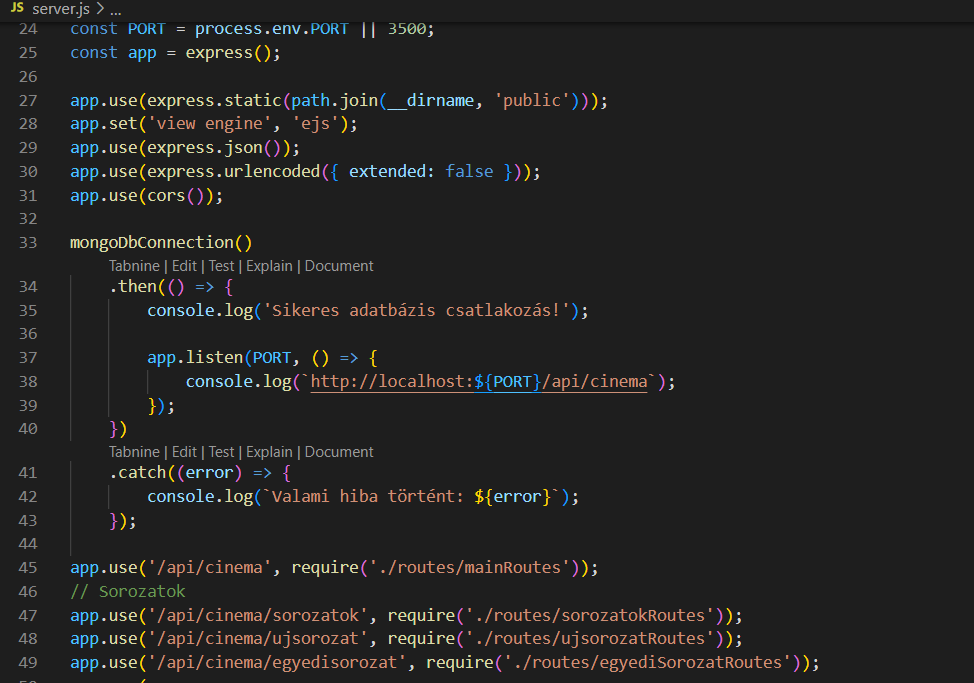
5. Foglalas.js - models (1)

A fenti képeken a Foglalas modell kódját láthatjuk. A kódunk a Mongoose kapcsolatot használja, amely az adatbázis kapcsolatot kezeli.   
Az adatbázis-kapcsolat létrehozása az alkalmazás indításakor történt, és az adatbázis-műveletek végrehajtása során használjuk.

A fájlunk egy MongoDB adatbázis-motorral dolgozik, amely az adatok eléréséért és módosításáért felelős. Az adatbázis műveletek Mongoose metódusokként jelennek meg, és a Mongoose kapcsolat biztosítja az adatbázis-kapcsolatot.

A kódunk a FoglalasModel modellt használja, amely az adatok eléréséért és módosításáért felelős. A modell a Mongoose kapcsolatot használja az adatbázis-kapcsolat létrehozásához.

### Server.js

1. server.js - cors 

A *server.js* a fő fájlunk. Ebben hívjuk meg a szükséges vezérlőket, itt állítjuk be a használt portot, és itt implementáljuk a megerősítő email küldésének funkcióját is.

A kódunk első részében a CORS (Cross-Origin Resource Sharing) engedélyezése található. Ez azt jelenti, hogy az alkalmazás szerveroldali engedélyeket biztosít a kliens oldali HTTP kérésekhez. Az "app.use()" függvény middleware funkciókat regisztrál az alkalmazásban, amelyeket az összes HTTP kérés előtt végrehajtódnak.

8. server.js - routes

A második részben a HTTP kérésekhez kapcsolódó útvonalak definiálására szolgáló függvények találhatók. A /api/cinema útvonalra számos al-útvonalat definiálunk, amelyek különböző funkciókat hajtanak végre.

**Útvonalak definiálása**

A /api/cinema/sorozatok útvonalra a sorozatokRoutes vezérlőt regisztráljuk, amely a sorozatokkal kapcsolatos funkciókat hajtja végre.

A /api/cinema/filmek útvonalra a filmekRoutes vezérlőt regisztráljuk, amely a filmekkel kapcsolatos funkciókat hajtja végre.

A /api/cinema/nezok útvonalra a nezokRoutes vezérlőt regisztráljuk, amely a felhasználókkal kapcsolatos funkciókat hajtja végre.

A /api/cinema/foglalasok útvonalra a foglalasokRoutes vezérlőt regisztráljuk, amely a foglalásokkal kapcsolatos funkciókat hajtja végre.

A /api/cinema/register útvonalra a registerRoutes vezérlőt regisztráljuk, amely a regisztrációs funkciókat hajtja végre.

A /api/cinema/login útvonalra a loginRoutes vezérlőt regisztráljuk, amely a bejelentkezési funkciókat hajtja végre.

**Egyedi útvonalak**

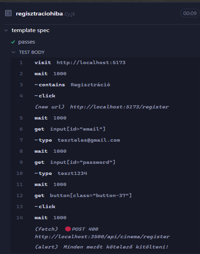
A /api/cinema/nezok/avatar útvonalra egy egyedi függvényt definiálunk, amely a felhasználók avatarját frissíti.

A /api/cinema/nezok/jelszo útvonalra egy egyedi függvényt definiálunk, amely a felhasználók jelszavát frissíti. **Megjegyzés**

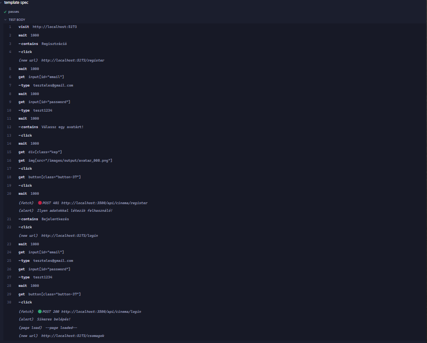
A kódunkban szerepel még a mongoDbConnection függvény, amely a MongoDB adatbázishoz való kapcsolódást hajtja végre. A cors middleware-t is használjuk, hogy engedélyezzük a keresztkérési erőforrás-megosztást (CORS).

**Tesztelés**  
  
A weboldal teszteléséhez a Cypress nevű eszközt használtuk, amellyel automatikusan ellenőriztük, hogy az oldal fő funkciói megfelelően működnek.  
  
Mit teszteltünk?  
  
- Regisztráció és bejelentkezés: működik-e hibás és helyes adatokkal is.  
- Filmek és sorozatok böngészése: betöltődnek-e rendesen, megjelennek-e a részletek.  
- Előfizetés: sikeresen le lehet-e foglalni, és visszajelzést kap-e a felhasználó.  
- Értékelés: elmenti-e a rendszer a felhasználói értékelést.  
- Felhasználói módosítások: például avatar vagy jelszó megváltoztatása.  
  
A tesztelés során néhány hibát is találtunk (pl. nem megfelelő űrlapellenőrzés), ezeket javítottuk. Miután minden teszt sikeresen lefutott, biztosak voltunk benne, hogy az oldal megbízhatóan működik.  
  
Emellett manuálisan is kipróbáltuk az oldalt különböző böngészőkben, hogy mindenki számára jól használható legyen.

Itt látható a hibás regisztráció tesztelése:



Itt látható a regisztráció illetve bejelentkezés tesztelése:



## 

## Továbbfejlesztési lehetőségek

1. Regisztrációs folyamat javítása: A regisztrációs folyamatot javíthatjuk, hogy a felhasználók egyszerűbben és gyorsabban regisztrálhassanak az oldalon. Például bejelentkezési lehetőségeket biztosíthatunk harmadik féltől származó fiókokkal, mint például a Google vagy a Facebook.
2. Mobil alkalmazás: A mobil alkalmazás bevezetése nagyszerű lehetőség az oldal bővítésére, és lehetővé teszi a felhasználók számára, hogy könnyedén böngésszenek a telefonjukon keresztül is. A mobil alkalmazásokban különböző funkciókat is elérhetünk, mint például push értesítések, valós idejű információk és még sok más.
3. Több nyelvű felület: A több nyelvű felület lehetővé teszi a felhasználók számára, hogy az oldalt az anyanyelvükön használják. Ez segít a felhasználóknak jobban megérteni az oldal funkcióit és megkönnyíti az a használatát külföldi potenciális ügyfelek számára is.

## Irodalomjegyzék, forrásmegjelölés

Jelen dokumentációban felhasznált képek forrása:  
<https://github.com/Takigyula/Filmkolcsonzo> - Github

A dokumentációban megjelölt weboldalak forrása:

* <https://code.visualstudio.com/> - Visual Studio Code
* <https://code.visualstudio.com/docs/setup/windows> - Visual Studio Code
* <https://code.visualstudio.com/docs/setup/linux> - Visual Studio Code
* <https://code.visualstudio.com/docs/setup/mac> - Visual Studio Code
* <https://nodejs.org/en/download> - Node Js

A projekt összeállításához felhasznált források:

* <https://www.w3schools.com/> - W3 Schools
* <https://nodejs.dev/en/learn/> - Node JS