

**Trefort Ágoston Technikum,
Szakképző Iskola és Kollégium
Szoftverfejlesztő – 54 213 05**

Záródolgozat

CinemaSzofisztikáció

Készítette: Karlovszki Laura 13/A

Konzulens: Szabó Richárd

Békéscsaba 2024.

Tartalomjegyzék

Tartalomjegyzék	2
A felhasználói dokumentáció	1
A program általános specifikációja	1
Rendszerkövetelmények.....	2
Szoftver követelmények	2
A program telepítése	3
XAMPP:	3
Node.js:.....	6
A program használatának a részletes leírása	9
A regisztráció oldala:.....	9
A Főoldal.....	11
PAGEREF _Toc163124294 \h Kívánság lista	16
Leírás és Kritika	19
Értékelés	19
Kép	20
A fejlesztői dokumentáció.....	21
Témaválasztás indoklása	21
Az alkalmazott fejlesztői eszközök	22
Adatmodell leírása.....	23
Részletes feladat-specifikáció, algoritmusok.....	26
Forráskód.....	31
Tesztelési dokumentáció	33
Továbbfejlesztési lehetőségek	34
Irodalomjegyzék, forrásmegjelölés	35

A felhasználói dokumentáció

A program általános specifikációja

A CinemaSzofisztikáció egy olyan szoftver, amely lehetővé teszi a felhasználók számára a látott műsorok tárolását és rendszerezését. Célja, hogy egy személyes filmes adatbázist biztosítson minden felhasználó számára, hogy könnyen nyomon követhessék és megosszák saját filmélményeiket magukkal a családjukkal vagy éppenséggel a barátaikkal.

Lehetőséget nyújt a filmek a mesék és a sorozatok szűrésére ezért több korosztály is kitudja használni az alkalmazást. A felhasználók képesek lesznek hozzáadni a saját látott műsoraikat a rendszerhez, megadva a film címét, keletkezését, kiadási idejét, kategóriáját és további olyan adatokat, amik az adott illetőnek fontos lehet. A felhasználók címkéket és kategóriákat adhatnak hozzá a műsorokhoz, azért, hogy egyszerűen lehessen külön válogatni az összes kategóriát, a meséknek és a sorozatoknak is van saját oldaluk, ami alapján könnyebb megtalálni. A programnak felhasználóbarát felülettel kell rendelkeznie, amely lehetővé teszi a felhasználók számára, hogy könnyen kezelhessék és személyre szabhassák saját filmes adatbázisukat. A programnak felhasználóbarát felülettel kell rendelkeznie, amely lehetővé teszi a felhasználók számára, hogy könnyen kezelhessék és személyre szabhassák saját filmes adatbázisukat. A programnak lehetőséget kell biztosítania a későbbi funkciók és fejlesztések könnyű integrálására, hogy az alkalmazás tovább fejleszthető legyen a felhasználói igényeknek megfelelően. Azért ezt a címet kapta, mert összehozza a filmes világot a szofisztikált, átgondolt megközelítéssel. A cím nemcsak leírja a szoftver célját, hanem egyedi és figyelemfelkeltő is. A "Szofisztikáció" szó a magasabb szintű gondolkodást és kidolgozottságát sugallja kifinomultság. A "Cinema" Ez egy univerzális szó, amelyet a filmkedvelők mindenhol könnyen azonosíthatnak, angolul a mozi jelent és igazából ezt egy zene miatt is gondoltam jónak. A cím egyszerű, közérthető és könnyen megjegyezhető, ami fontos a szoftver terjesztése szempontjából. Öszintén mindenkinek csak ajánlani tudom bármelyik korosztálynak mivel ténylegesen családbarát és személyreszabható tehát semmi féle olyan tartalom nem tud felkerulni ide ami esetleg a fiatalabbaknak kellemetlenséget rémálmokat vagy barmifele agressziót okozhat és így a meséjüket se felejtene el, idősebb korosztálynak is szánom mivel barmiféle filmet feltudnak tölteni maguknak.

Rendszerkövetelmények

Hardverkövetelmények

Minimum:

1. táblázat

CPU	GPU	RAM	OS	Tárhely	Internet
Intel Core I3-530	Nvidia GeForce 9600GT 1GB Vram	2GB+	Win 10 +	1GB+	Szükséges

Ajánlott

CPU	GPU	RAM	OS	Tárhely	Internet
Intel Core i5-3300	Nvidia GeForce 560	4GB+	Win 10 +	1GB+	Szükséges

Szoftver követelmények

Szükséges szoftverek a program működéséhez: XAMPP, NODEJS. Illetve bármilyen Windows rendszer képes a működésére, amire elérhető a XAMMP.

A program telepítése

Helyezze be a CinemaSzofisztikáció telepítő CD-jét a számítógép CD/DVD meghajtójába.

Átalában a számítógép fel ajánlja, hogy telepítse-e az adott alkalmazás meg a fájlt is megnyitja, ha engedélyezzük neki, de ha ezek a lehetőségek nem jelennek meg akkor Nyissa meg a Fájlkészlőt, és válassza ki a CD-t. Minden szükséges komponenst kijelöl, majd kattintson a "Next" gombra.

A visual studio letöltése is ajánlott igazából mivel azon belül lehet úgymond elindítani magát az alkalmazást----- ezután a mappából nyissuk meg a az alkalmazást a visual studioba a következő lépés a XAMPP telepítése

XAMPP:

Leírás: XAMPP egy ingyenes és nyílt forráskódú szoftvercsomag, amely tartalmazza az Apache HTTP szerver, MySQL adatbázis-kezelőt és PHP-t.

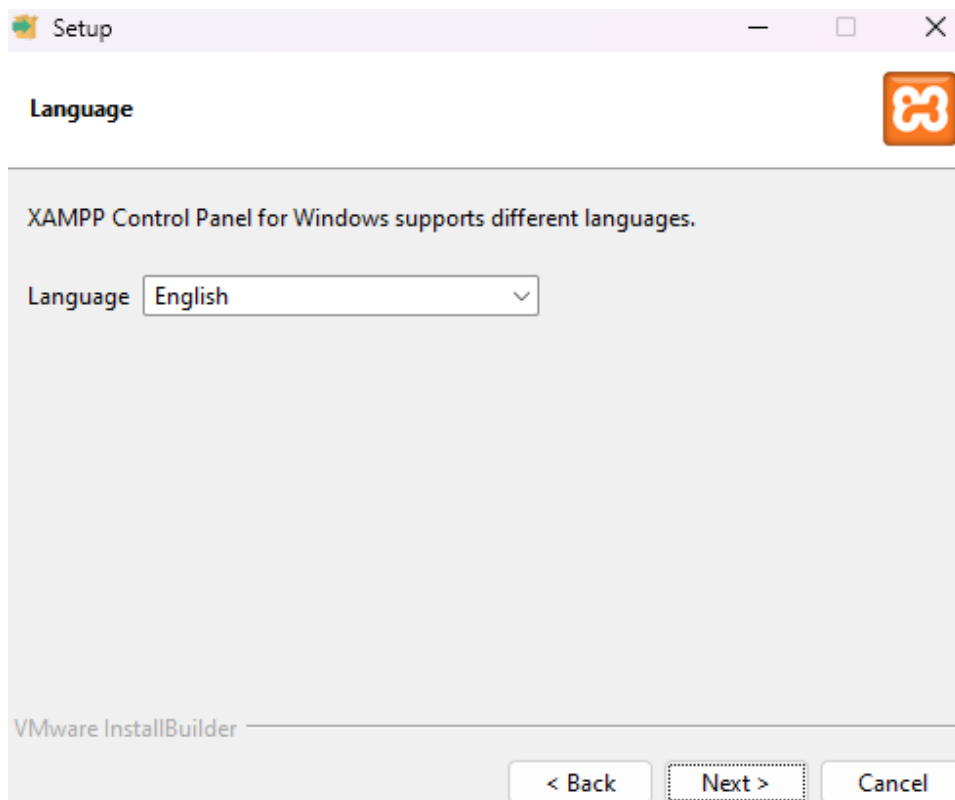
De igazából a CDn rajta van szóval elég csak onnan letölteni

Verzió: A legfrissebb stabil verzió ajánlott.

Letöltés és telepítés: A szoftver letölthető és telepíthető a hivatalos weboldaltól:
<https://www.apachefriends.org/index.html>

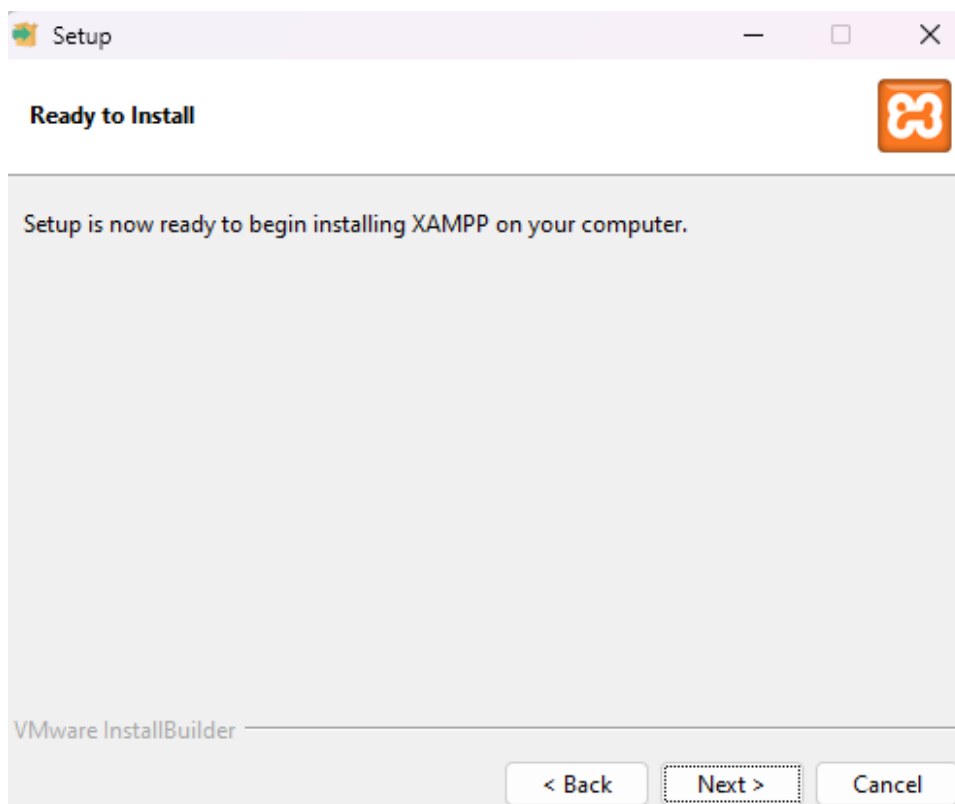
Letöltés folyamata:

- A letöltés oldalon kattintson a letöltés gombra.
- Miután befejezte a letöltés nyissa meg a mappát, amit kiválasztott a fájl telepítésére.



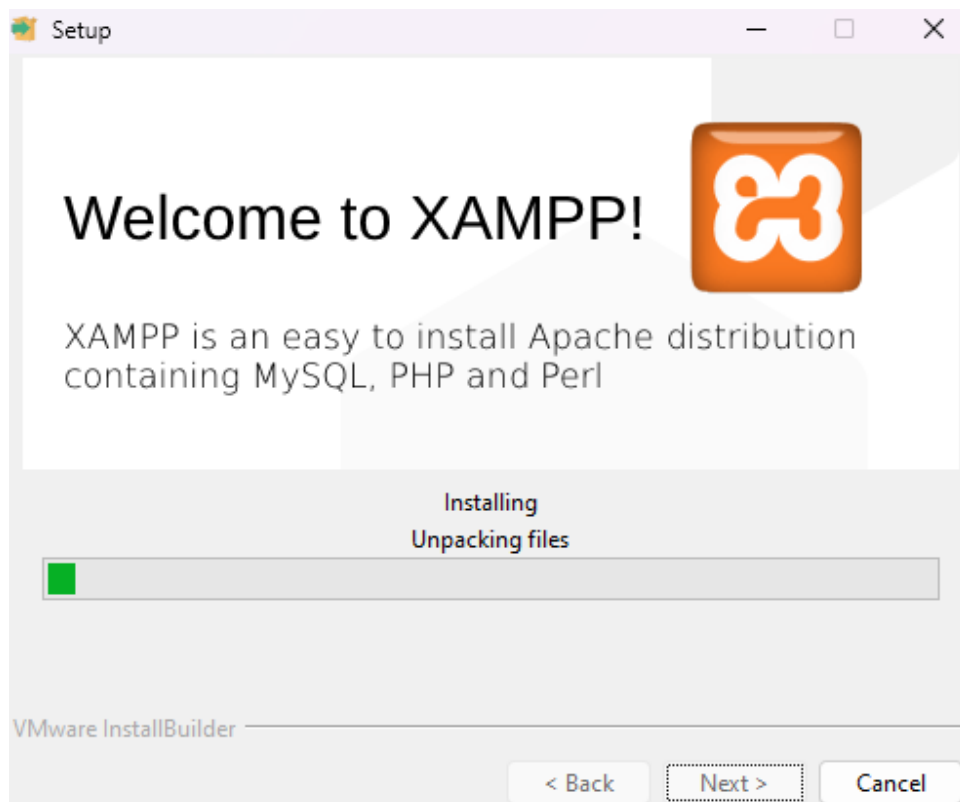
1. ábra

- Válassza ki azt a nyelvet, amelyiket beszélni vagy érteni, majd nyomjon a "Next" gombra



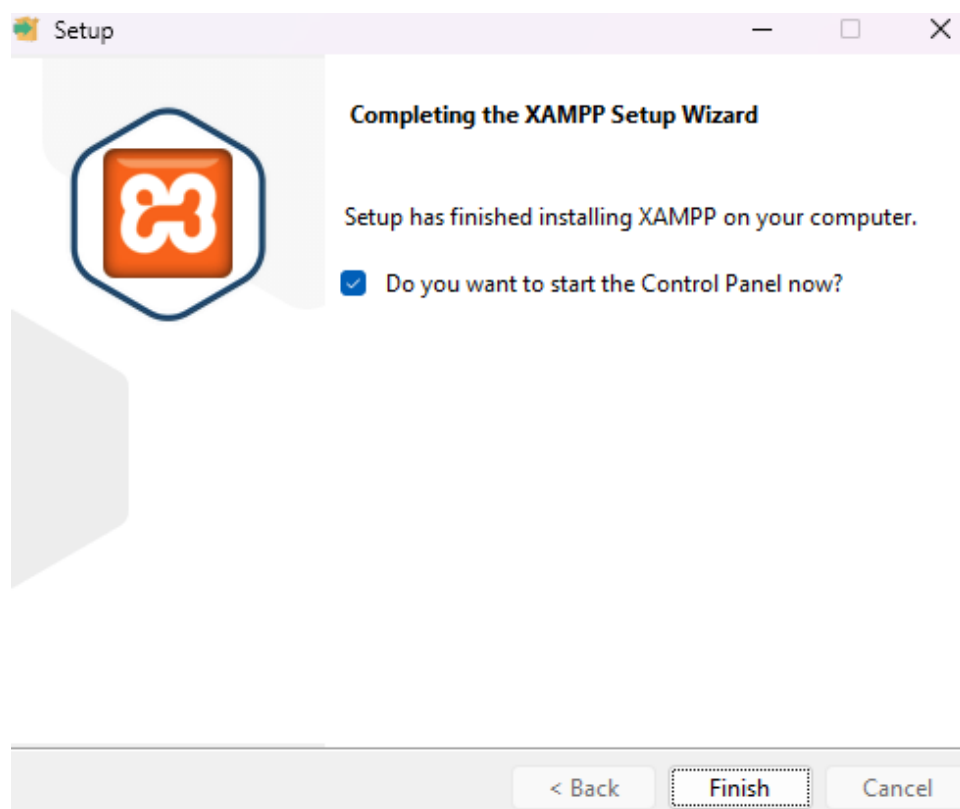
2. ábra

- Ha a továbbiakban is folytatni szeretne, ismét nyomja meg a "Next" gombot.



3. ábra

- A folyamat megismétlődő, ha befejezte a letöltés folytassa a “Next” gomb megnyomásával.



4. ábra

- Majd végezetül nyomjuk meg a “Finish” gombot es készen is van a XAMPP letöltése.
- Majd, ha ez is megvan jöhet a node js mivel az is szükséges a programunk mukodtetesehez

Node.js:

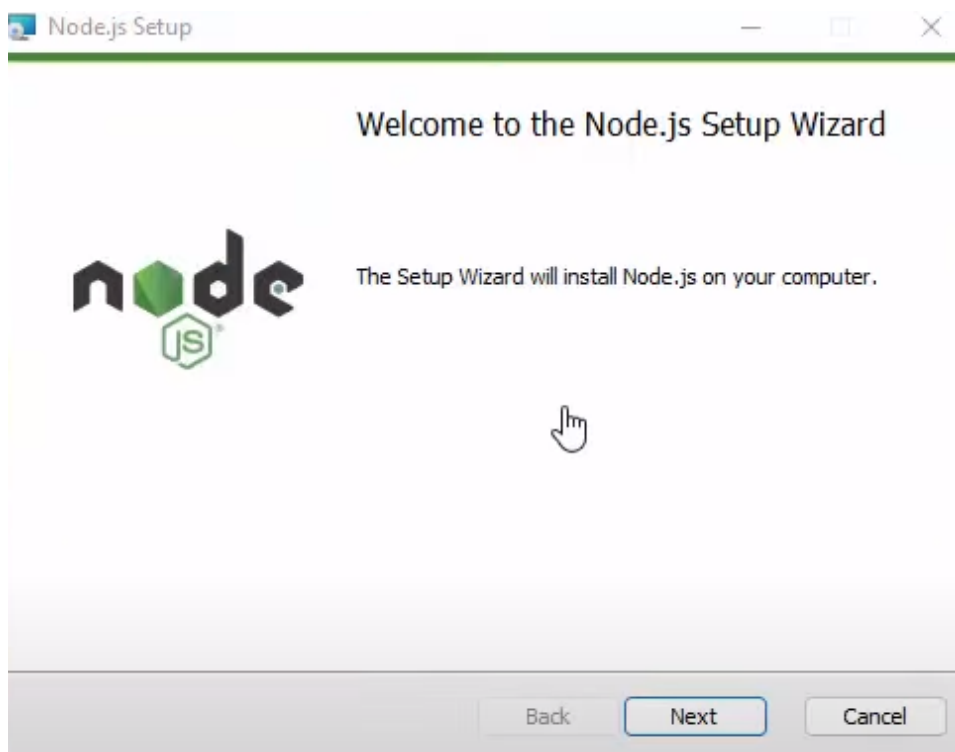
Leírás: Node.js egy JavaScript runtime környezet, amely lehetővé teszi a szerveroldali JavaScript kód futtatását.

Verzió: A legfrissebb stabil verziót ajánlott használni.

Letöltés és telepítés: A Node.js letölthető és telepíthető a hivatalos weboldaltól: <https://nodejs.org/>

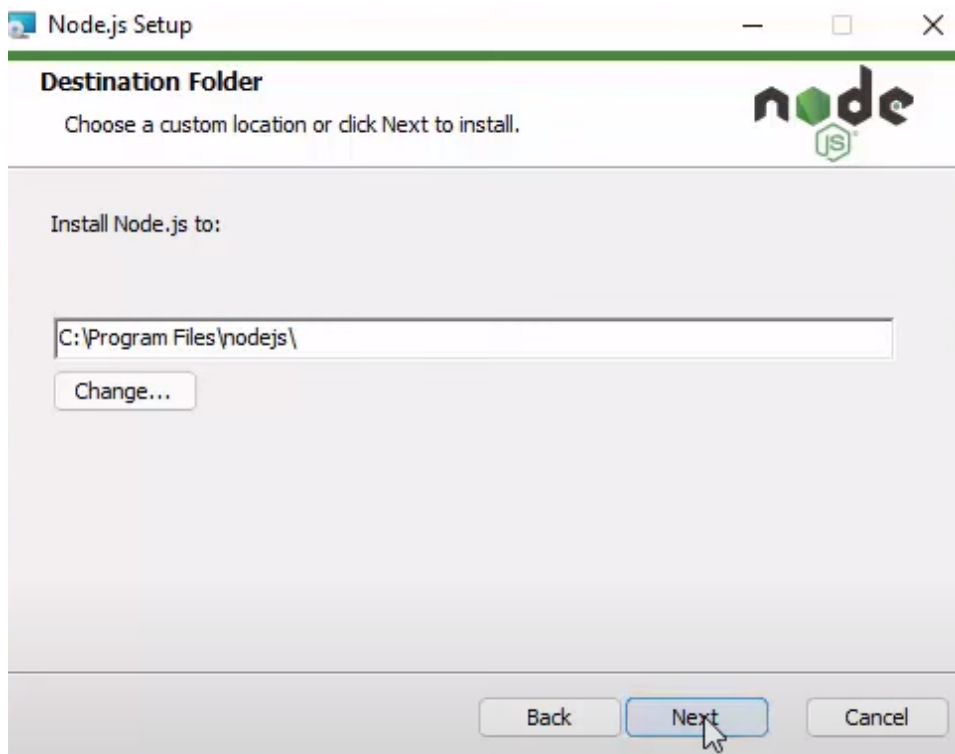
Letöltés folyamata:

- A letöltés oldalon kattintson a letöltés gombra.
- Miután befejezte a letöltés nyissa meg a mappába, amit kiválasztott a fájl telepítésére.



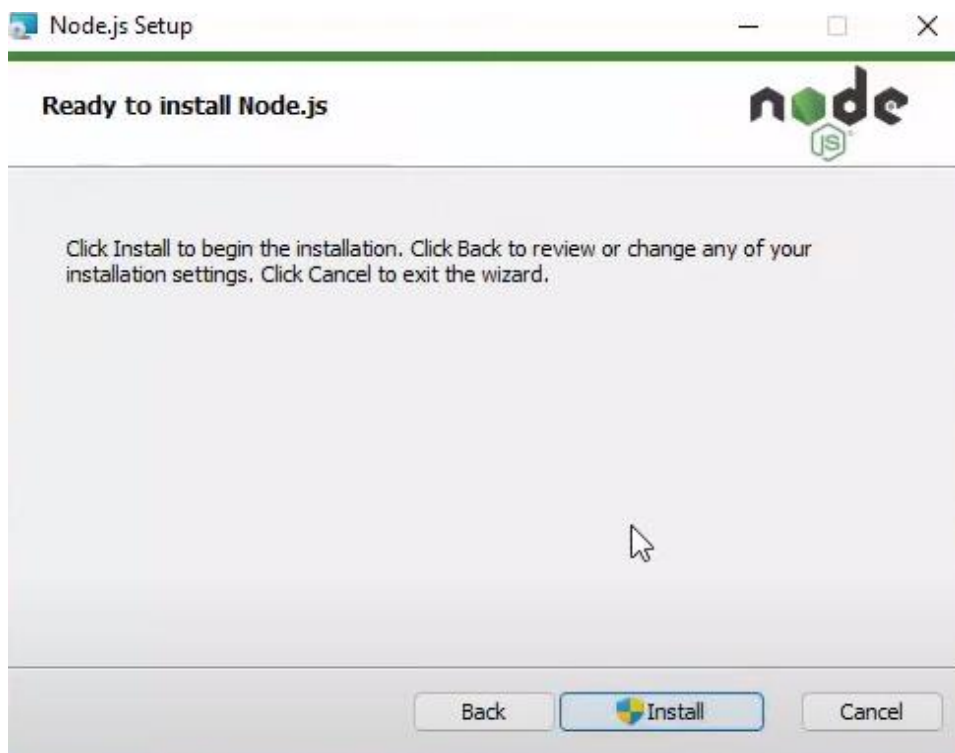
5. ábra

- Nyomjon a "Next" gombra, ha folytatni akarja.



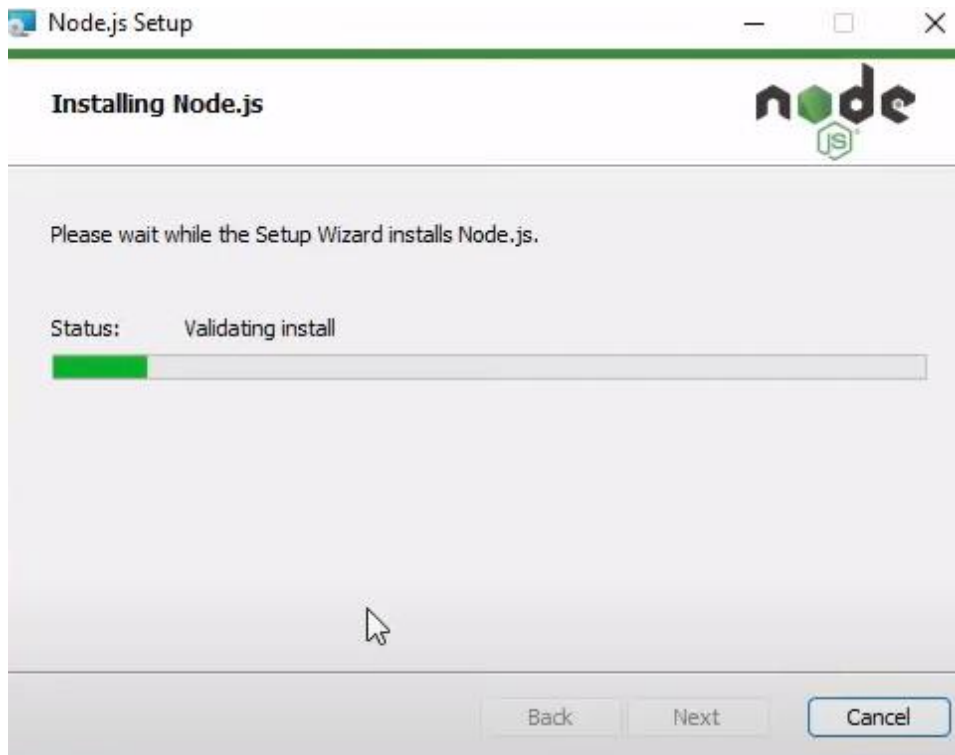
6. ábra

- Ha esetleg változtatni szeretne a fájlnek a metésének a helyen itt megteheti, ha meg így tökéletes számára nyomjon a “Next” gombra.



7. ábra

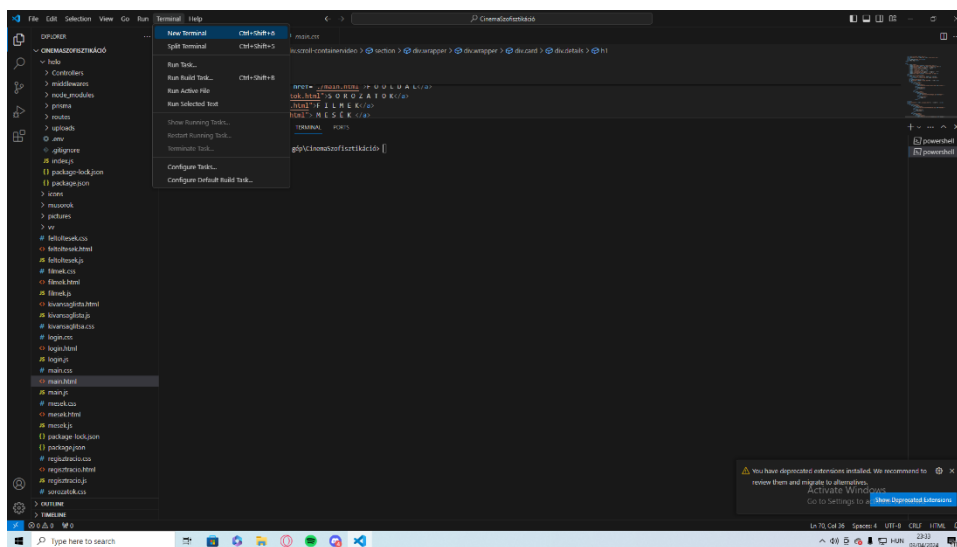
- Itt már készen áll a Node.js-unk a telepítésre, folytassuk is az “Install” gomb megnyomásával.



8. ábra

Megvárjuk, ameddig be nem fejeződik a telepítés majd a "Next" gomb megnyomásával készen is leszünk.

És sikeresen telepítve van az összes alkalmazás, ami szükséges a szoftver felhasználásához. Ha mind ezzel megvagyunk, jöhet is a Terminal megnyitása a Visual Studio-ban ---->



9. ábra

Es el is kezdhetjük a az alkalmazásunk üzemeltetését annak érdekében, hogy működjön.

Első sorban a be kell lépniünk a backendbe a következő paranccssorral

----->**Cd helo**

Ha ez megvan, el kezdhetjük a Node js telepítését. -----> **npm install.**

Ha ezt sikeresen letöltötte telepítenünk kell meg a prizrát, ami az adatbázisunkat kezeli a következő kód sorral-----> **npm install prisma --save-dev--> npx prisma generate.**

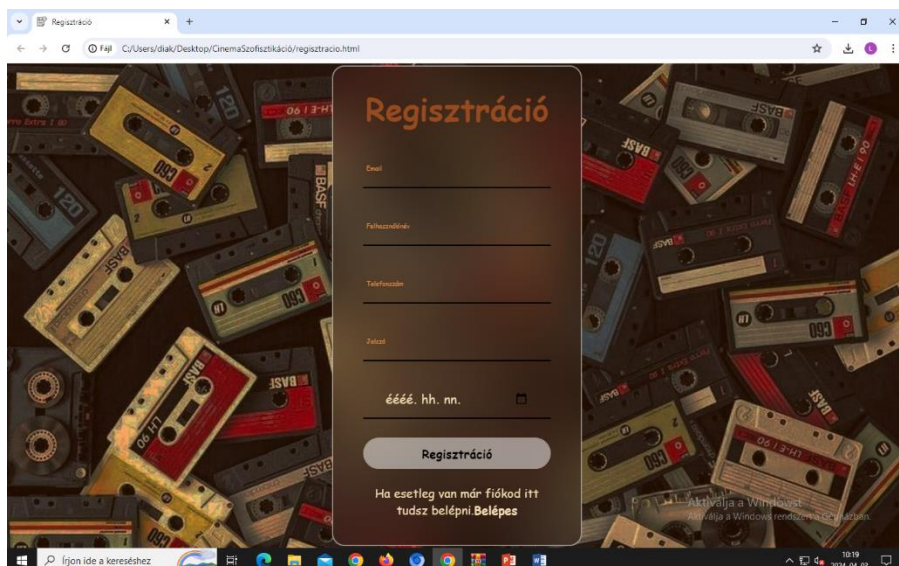
Ha ez si megvan meg kell nyitnunk a XAMPPunkat es elindítania az APACHET és a MYSQLt is.

A weblapot nyissuk meg mappából és már tudjuk is hasznain.

A program használatának a részletes leírása

Az alkalmazás használata alapvetően elég egyszerű. Főleg, amiatt mert nincsen külön admin felhasználói felület mivel mindenki saját magának tudja személyre szabni
Második sorban is a felhasználónak regisztrálnia kell annak érdekében, hogy láthassa a fő oldalt.

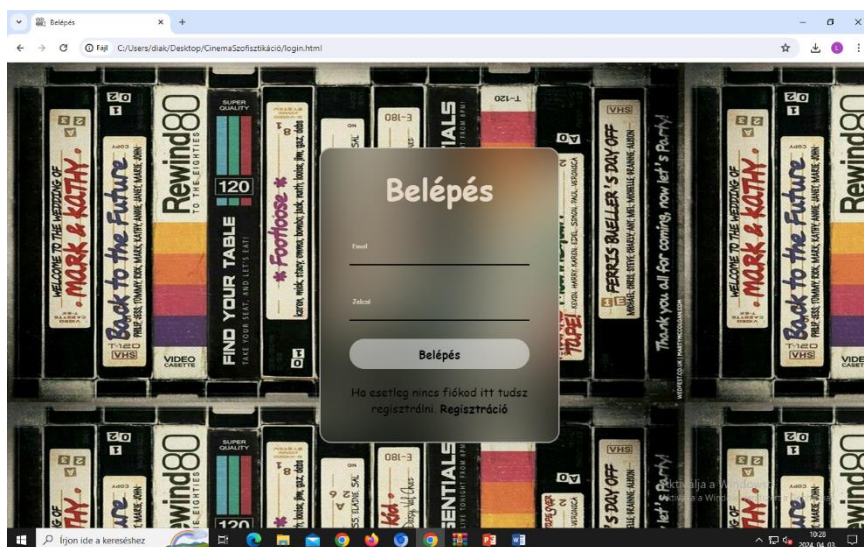
A regisztráció oldala:



10. ábra

Letisztult a felülete egyszerű a regisztrálás akar fiatalabb személyek számára is. Ha a felhasználó már rendelkezik egy adott fiókkal, akkor a belépés nevezetű szövegre nyomva

Ezt a felületet látja és egyszerűen az Email-címmel es a jelszavával be tud a felhasználó jelentkezni es tökéletesen tudja használni a sajátosan személyre szabott szoftverét.



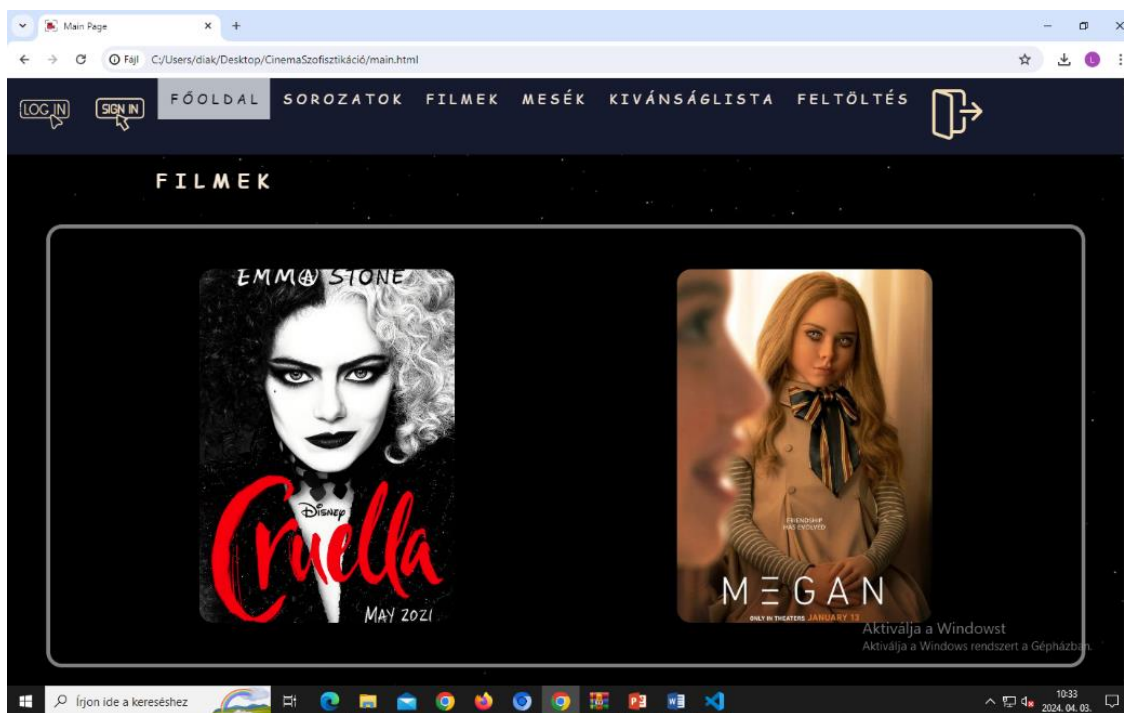
11. ábra

Amint belépett lathatja a szoftver fő oldalát ahol több navigációs gomb is található, amit szinten a letisztultsága és egyszerű működési elve miatt bárki könnyen tudja kezelni, illetve használni.

De igazából a regisztrációnál a felhasználó névhez a saját nevet vagy esetleg valamifele fantázia nevet használhat. Email cím mezőben bármiféle email cím elfogadható viszont ez az egyik legfontosabb adatunk mivel ezzel tudunk majd a későbbiekben bejelentkezni, illetve a

jelszóval, amit titkosít az oldalunk, de persze a többi mezőt is ki kell töltenie máskülönben nem, tudunk regisztrálni erre a szuper szoftverre

A Főoldal

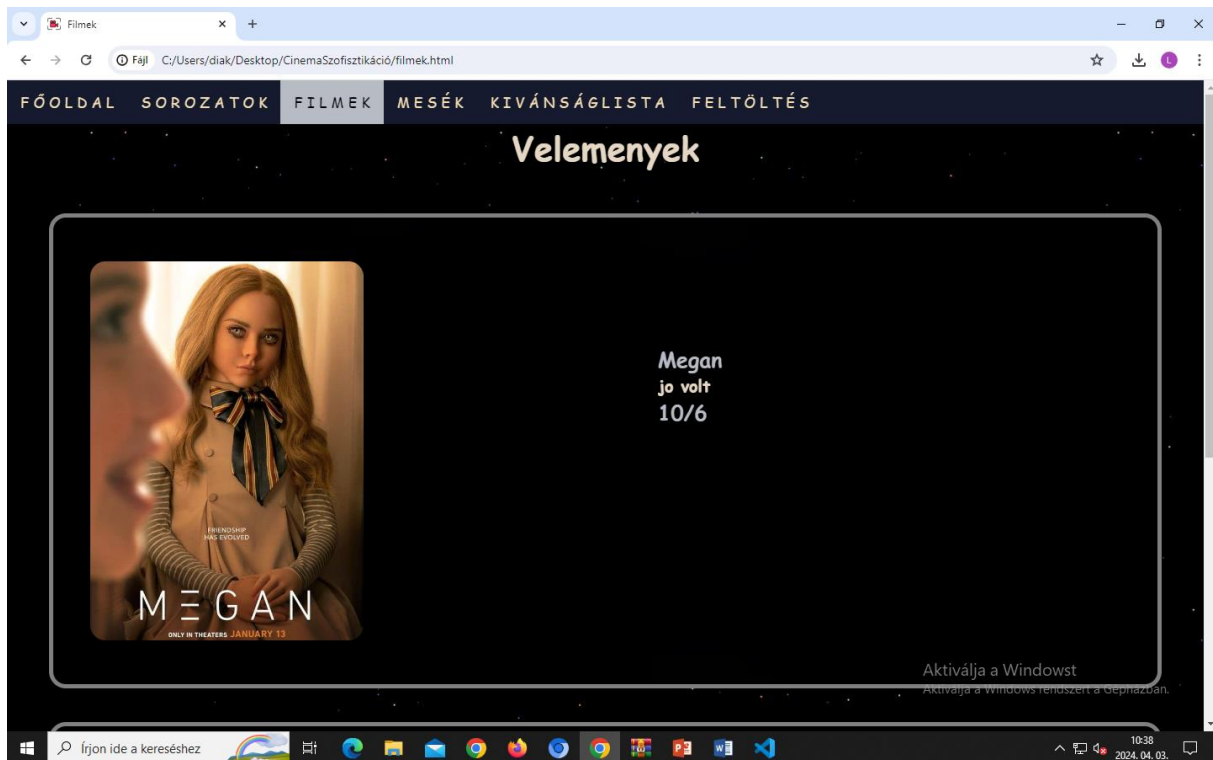


12. ábra

Könnyen olvasható és átlátható felületet kapunk meg ahol a filmeket lathatjuk elősorban. Ahogyan rávisszük az egerünket a filmekre esetleg mesék illetve filmekre lathatjuk az adott műsornak a címet. A képek, illusztrációk hogy hogyan is nézne, ki ahogy saját műsorokat toltunk fel magunknak. Az itt található menüsorban látható singin login illetve a kilépést jelző ikon csak itt látható mivel ez az oldal a legpraktikusabb a tájékozás szempontjából.

A kilépést szimbolizáló ikon vissza vezet a belépés oldalra. Illetve onnan eltudunk jutni a regisztrációs felületre. Az összes oldal mutatja, hogy éppen melyiken is tartózkodunk. Tehát felhasználó barát es nem tudunk belezavarodni. Ezek az oldalak nem nyílnak meg új lapon, hanem az adott lap léptet át oda minket.

A Filmek címre kattintva átvezényel minket a Filmek nevezetű oldalra, ahogy itt láthatjuk is:

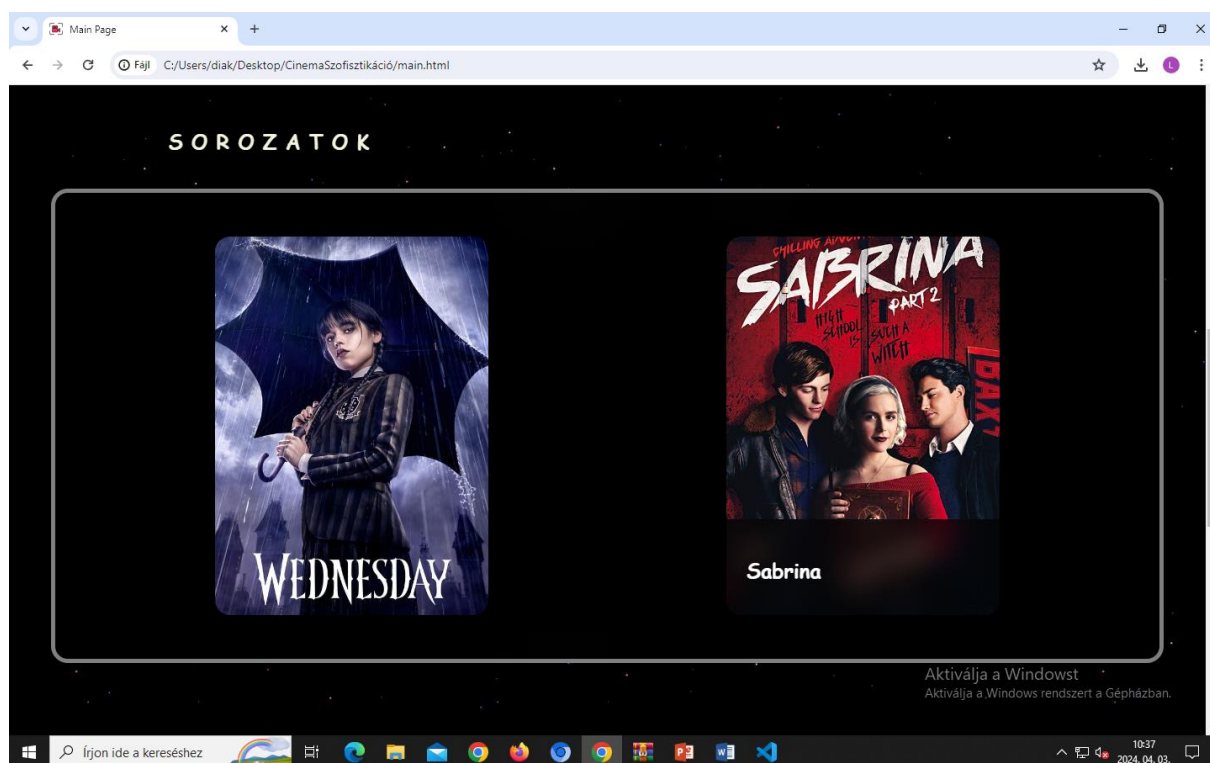


13. ábra

Ezen az oldalon is van, egy funkció amint rávisszük az egeret az adott filmre megjelenik egy gomb, amivel hozzá, tudjuk adni a kívánság lista nevezetű oldalhoz, ami azért is van, hogy esetleg azokat a filmeket, amiket újra szeretnénk nézni el lehessen menteni de a későbbiekben kiterек részletesebben az oldal lényegere.

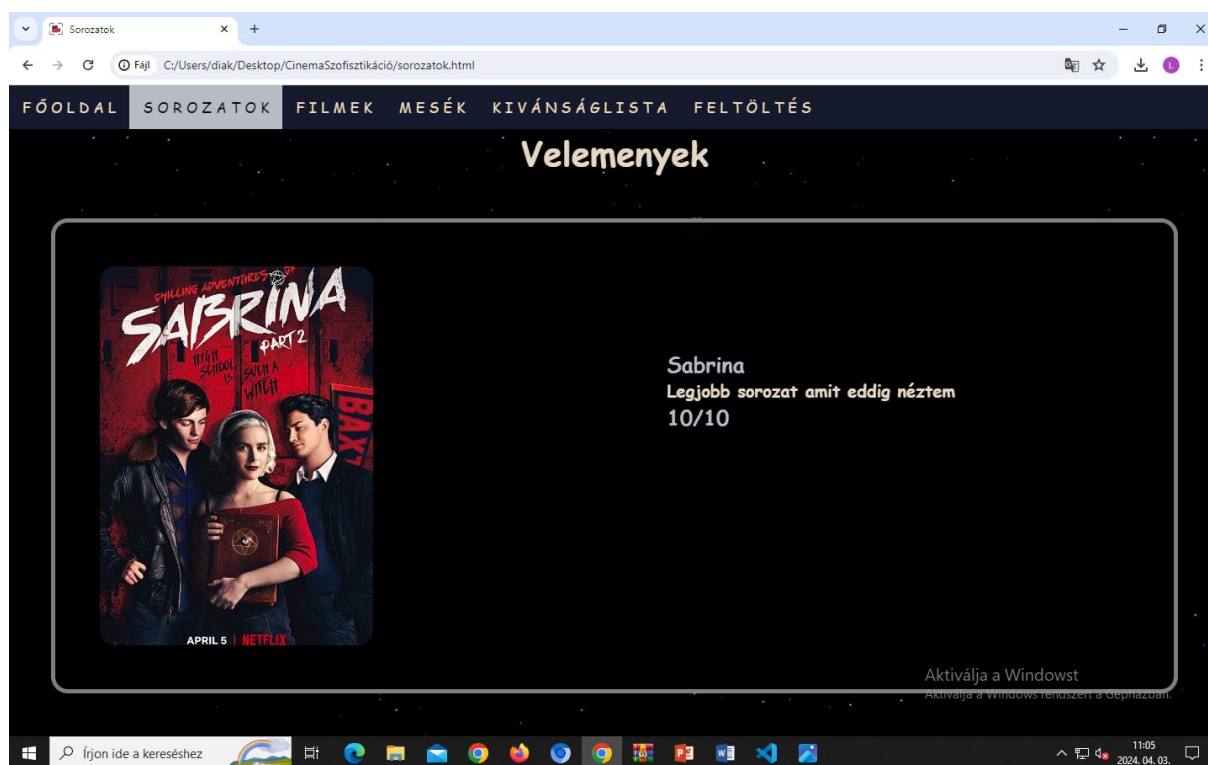
Ahogy haladunk lefelé a fő oldalon láthatóvá válik számunka a másik kettő kategória is, ami hasonló felépítéssel van kiírtatva. Ahogy görgetünk lefele a többi Filmet is láthatjuk szépen egymás alá felsorakoztatva az összes adattal, amit megadunk, a feltöltések nevezet oldalon erről a későbbiekben többet fogok beszélni.

A következő témakör a műsorokat tekintve a sorozatok:



14. ábra

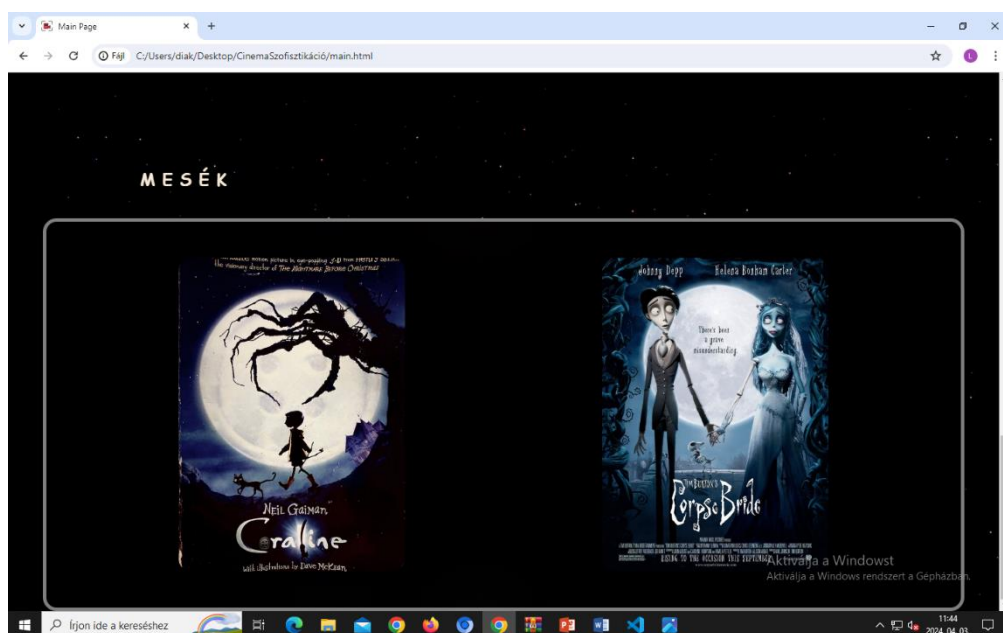
A működési elve ugyan az, mint a Filmeknél hogy a Sorozatokra rá nyomva átvezényel a sorozatok nevezetű fültre ahol szinten az adatbázisból elérhető sorozatok láthatóak, amelyeket mi töltöttünk fel a feltöltés nevezetű oldalon.



15. ábra

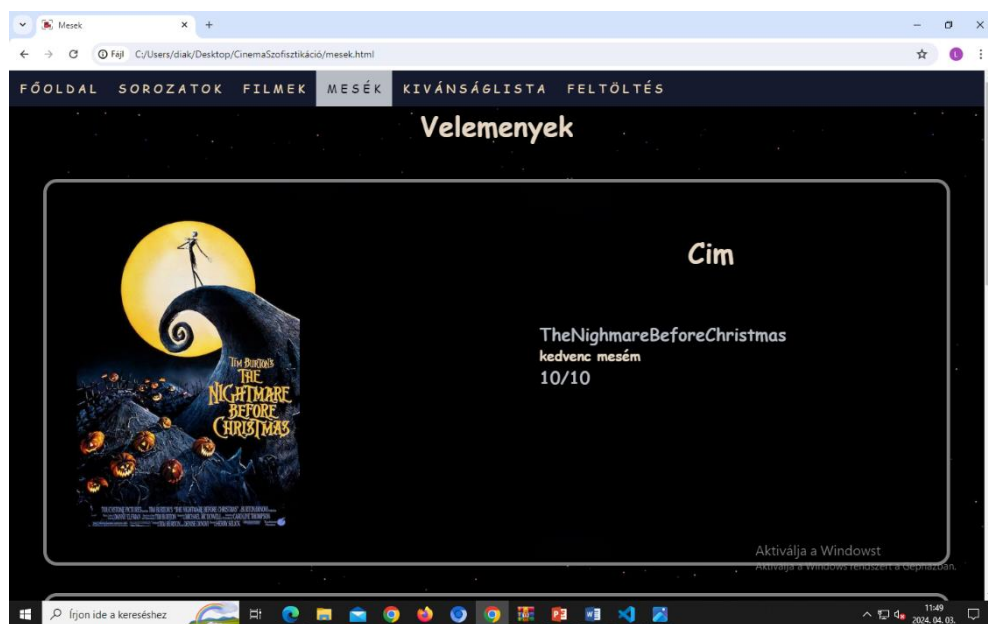
Ezen az oldalon is van, egy funkció amint rávisszük az egeret az adott Sorozatra megjelenik egy gomb, amivel hozzá, tudjuk adni a kívánság lista nevezetű oldalhoz, ami azért is van, hogy esetleg azokat a sorozatokat, amiket újra szeretnénk nézni el, lehessen menteni, és ne keljen felesleges keresgélésbe kezdeni, hogy vajon mi is lehetett az adott sorozatnak a címe, ez is igazából egyszerűsíti az adott felhasználónak az élményét mivel így nincs arra szükség hogy órákon keresztül keressenek egy adott sorozatot csak mert nem emlékezne esetleg a címére.

Majd a legutolsó kategóriánk a mesék ez azért is pozitív, mert lényegében fiatalabb személyek is tudják használni vagy esetleg szülők, ha volt olyan mese, ami nagyon tetszett a gyerekeiknek vagy esetleg saját maguknak.



16. ábra

Ahogy az eddigiekben is tapasztalhattuk ugyan úgy működik, mint a másik kettő, ha rávisszük az egeret lathatjuk a címet illetve a mesék című feliratra rákattintunk, láthatóvá válik számunka a Mesék nevezetű oldal mivel ugye ez is ezzel a mozdulattal átvezet minket a mesék oldalára



17. ábra

Ezen az oldalon is ugyanúgy található egy gomb a képén, ahogy rávisszük az egeret es itt is ahogy a sorozatoknál vagy a filmeknél a kívánság listához lehet hozzá adni ennyit szeretnek mondani a main oldal felépítéséről.



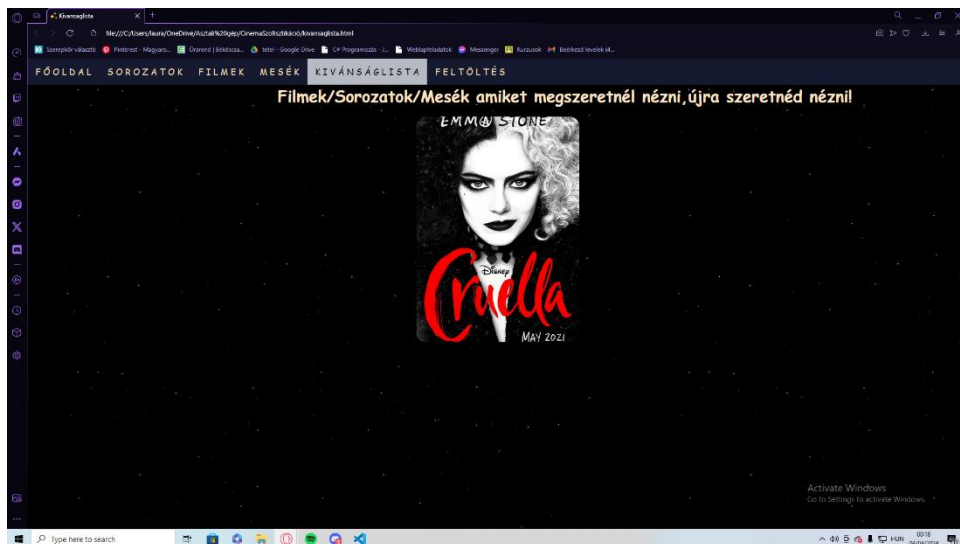
18. ábra

A menüsorban így megtalálható az összes oldalunk, amit az eddigiekben felsoroltunk. A Login nevezetű gombbal át tudunk jelentkezni esetleg egy másik fiókba vagy egy teljesen másik ember a saját fiókjába. A Sign In nevezetű gombbal regisztrálni tudunk egy új fiókot. Főoldal, amit a fentiekben mutattam be.

Kívánság lista

A kívánság lista azt a célt szolgálja, mint ahogy a cím sorban is olvasható azok a filmek, mesék, sorozatok, amiket újra vagy először szeretnénk megnézni.

Itt is hasonló az eset a film címet lehet leolvasni illetve a kritikát, amit az adott felhasználó írt róla. Egymás alatt helyezkednek el, a képek tehát nem kell aggódnunk annak érdekében h esetleg nem tette át az egyik oldalról a számunkra megfelelő műsort Lényegében ennyit szeretnék erről az oldalról beszélni.



19. ábra

Feltöltés nevezetű oldal picit összetettebb mivel több mezőt is ki kell töltenünk, hogy sikeres legyen a feltöltés viszont ez sem bonyolult az alkalmazásunk segít nekünk az mivel

minden kitöltendő mező felett oda van írva szép nagy betűkkel hogy hova milyen adatot kell írunk

20. ábra

A név arra szolgál, hogy mi is az adott filmek mesének vagy sorozatnak a címét kell oda beírni, ahogyan ezt kitöltöttük jöhet is a következő lépés a kiadásának az időpontja ezt elég egyszerűen kilehet választani a naptárból vagy esetleg saját magunk írjuk le. Akar, olyan filmeket is feltudunk tölteni, ami még esetleg, nincsen kiadva tehát a naptarba igazából jövő idejű dátumot is fel lehet vinni, ami elég hasznos, lehet annak érdekében, hogy esetleg feltesszük a kívánság lista nevezetű oldalunkra és bármikor láthatjuk, illetve eszünkbe juthat,

kevesebb eséllyel felejtjük el, ha mondjuk 2 évvel később fog kijönni.

21. ábra

A műsoroknak a leírása a tartalmát jeleníti, meg hogy miről is szól az adott tartalom. A kategóriák nevezetű fül arra szolgál, hogy ki lehessen szűrni, úgymond hogy a mesék meg a filmek és a sorozatok ne egybe ömlesztve jelenjen meg, hanem szépen kiválogatva azt.

Feltöltések

C:/Users/diak/Desktop/CinemaSzofisztikáció/feltoltesek.html

Név

Kiadás

6666. hh. nn.

Kategóriák

Film
Mese
Sorozat

Leírás

Kritika

Értékelés

Kép

Fáj kiválasztása

Küldés

Aktiválja a Windowst
Aktiválja a Windows rendszert a Gépházban.

Írjon ide a kereséshez

13:22
2024. 04. 03.

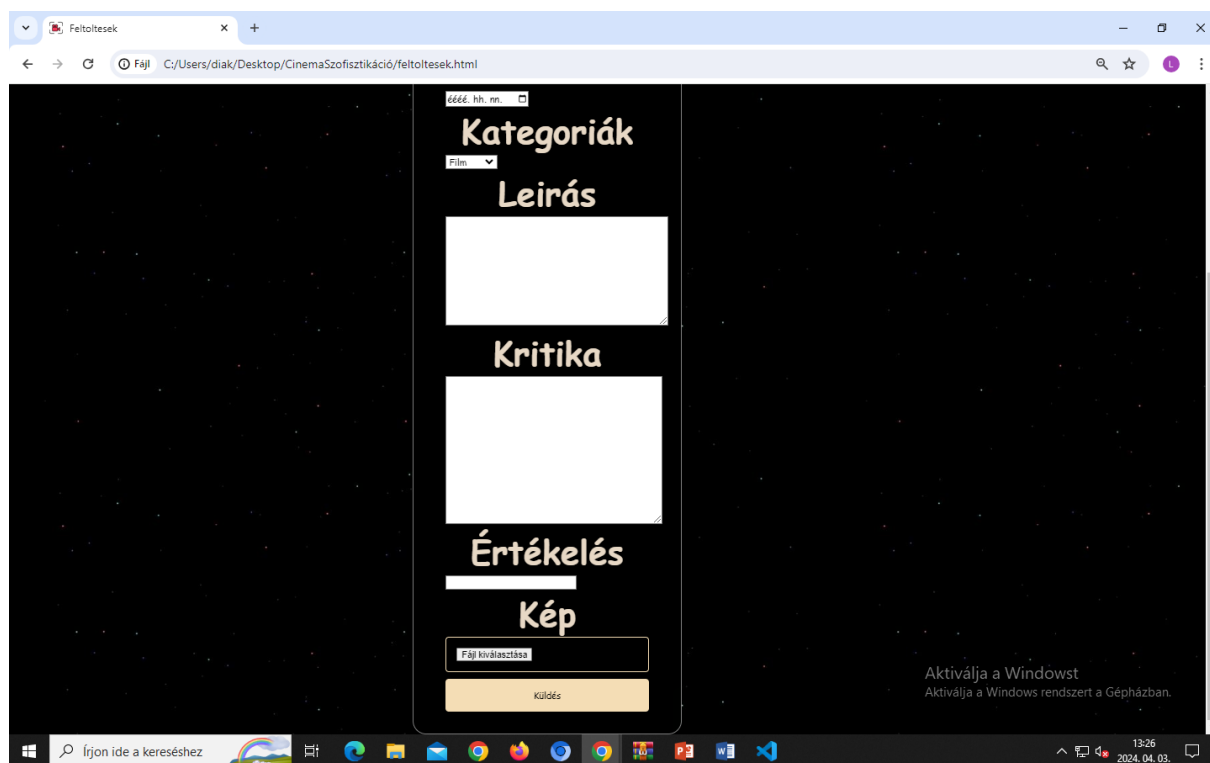
22. ábra

Ezekre az oldalakra menti el az összesre rá tudunk kattintani, hogy láthassuk eddig mit is töltöttünk fel, külön - külön új lapokon is meglehet nyitni, hogy egyszerre láthassuk, hogy milyen filmek mesék, illetve sorozatok, vannak eddig feltöltve.

SOROZATOK FILMEK MESÉK

23. ábra

Leírás és Kritika



24. ábra

A leírás szolgálja azt a funkciót, hogy az adott Műsorról mit is kell tudni tehát röviden a tartalma miatt van fenn tartva, el tudja mesélni mit látott az adott műsorban és mennyire is volt emlékezetes illetve tartalmaz rövid elbeszélés az adott műsorról.

A kritika a másik fontos eleme a feltöltésnek mivel itt letudjuk, írni mi tetszett benne illetve mi nem el tudjuk mesélni melyik részét hogyan éltük meg milyen érzelmeket generált bennünk hogyan is hatott ránk olyan részletesen tudjuk kifejezni amilyen részletesen csak, szeretnénk mivel korlátlan a karakter száma.

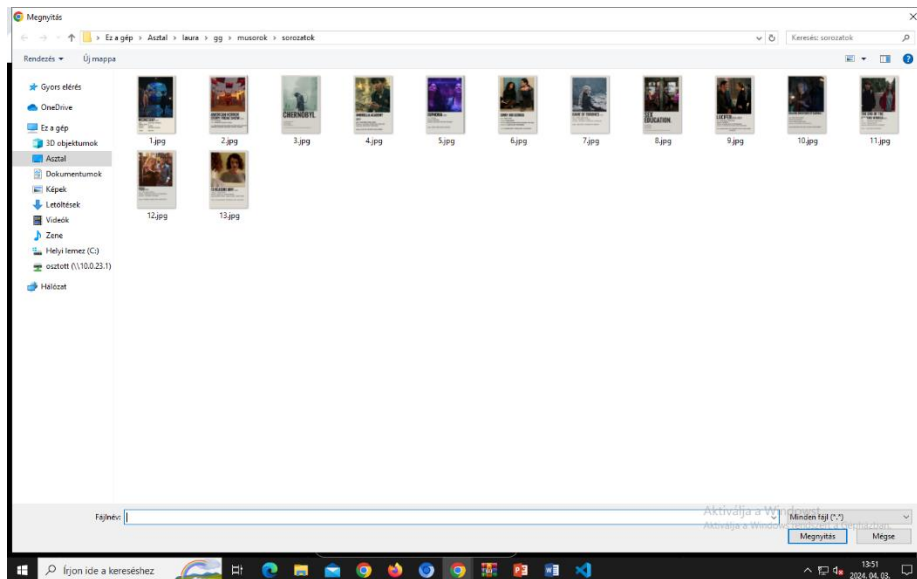
Értékelés

Az értékelés azt a célt szolgálja, hogy 10/ mennyire tetszett nekünk a film ide számokat tudunk beréni, ami a filmek, mesék, sorozatok oldalon megjelenik.

Kép

Itt a műsoroknak a borító képét tudjuk feltölteni a fájl kiválasztása nevezetű gombbal.

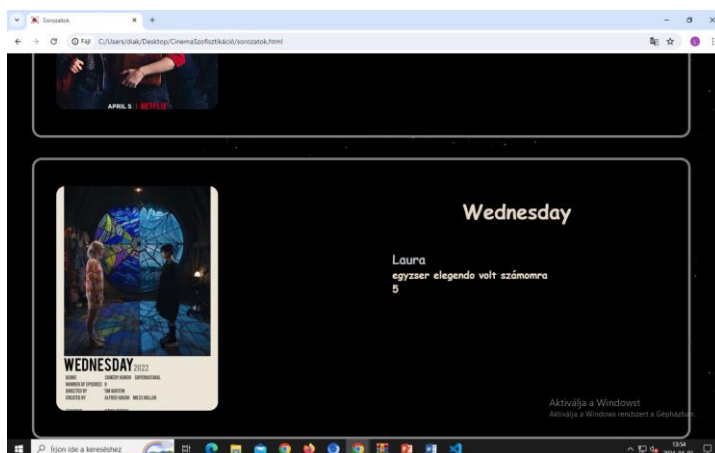
Jpeg
png



vagy

25. ábra

formátumban tudunk képeket feltölteni. Majd ha ezekkel mind megvagyunk meg kell nyomni a küldés nevezetű gombot és készen is van. De a szoftverünk úgyis jelzi, hogy sikeres volt-e a feltöltése, vagy esetleg ha hiányoznak adatok is megmondja nekünk. De viszont akkor újra kell kezdenünk az egész oldal és mező kitöltését illetve a kép feltöltését.



26. ábra

És ezt láthatjuk is a megfelelő oldalon. A Laura, mint a felhasználó, azért látszódig hogy tudja, éppen melyik fiókjába van belépve majd az összes többi adat, amit a feltöltések oldalán láthatunk.

A fejlesztői dokumentáció

Témaválasztás indoklása



27. ábra

Azért ezt a témát választottam mivel én személy szerint aránylag elég sok filmet, sorozatot vagy éppen mesét nézek az öcsémmel ezért arra gondoltam hogy záródolgozat témájának tökéletes lenne egy olyan alkalmazás ahol számon tudom tartani azokat a műsorokat amiket valaha is láttam, ez elég nagy segítség lehet azok számára akik szintén sok hasonló tartalmat fogyasztanak mivel így nem zavarodnak bele hogy látták-e már vagy sem, tudnak maguknak értékelni és véleményt is írni az adott műsorról hogy érdemes-e megnézni újra vagy sem esetleg ajánlanák-e barátoknak vagy családtagoknak. Ezért gondolom azt, hogy szükség volt a programra illetve hasonló szoftvert nem igazan lattam szóval ezért is érzem egy jó választásnak. Meg igazából saját felhasználásra is tökéletes lesz.

Nemcsak szakmai, Hanem személyes szempontból is jó választásnak tűnt. Személyes szempont miatt azért gondoltam jónak mivel eddig a jegyzetemben, ami a telefonomon található írtam fel a filmek címét és értékeléseit ez az alkalmazás meg áltathatóbb, letisztultabb, mint egy jegyzetomb. Ez az egyik indok, ami miatt nagyon fontosnak tartom ezt az alkalmazást mivel sokat jelent számomra a minőségi időtöltés társaságban akár egyedül

Szakmai szempontból meg elég nagy gyakorlást és szakmai fejlődésben segített nekem mivel elég sok újdonságot tanulhattam az alapján, hogy utána kellett néznem vagy esetleg segítséget kérnem. Évelejéhez képest sokkal többet tudok és ezt ennek a szoftvernek is köszönhetem. Szóval hálás vagyok az összes tanáromnak, aki segített, illetve osztálytársaimnak.

Az alkalmazott fejlesztői eszközök

Az alkalmazáshoz felhasznált fejlesztői eszközök: Az alkalmazásom alapköve a HTML.

A HTML minden létező weboldal létfontosságú eleme tehát erre épül fel.

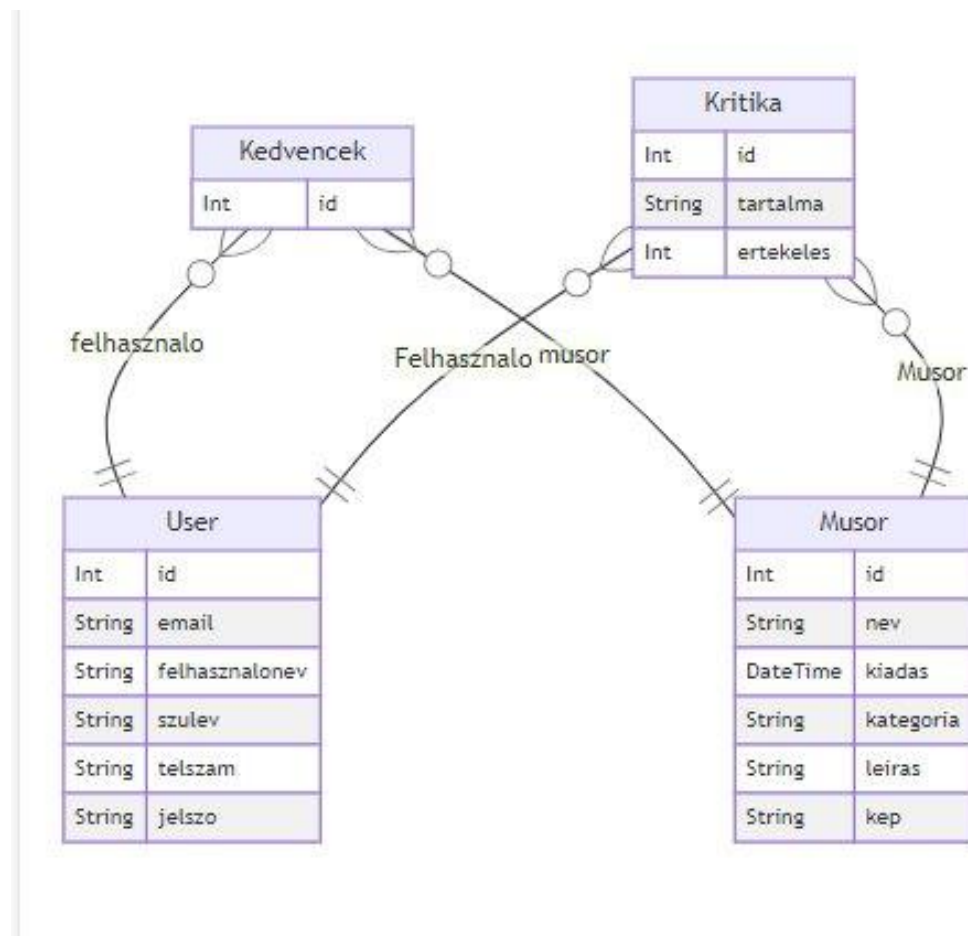
Erre jön rá a CSS ami a formai megjelenésért szolgál azéért használtam CSS-z mivel számomra ez az egyik legátláthatóbb és egyszerűbb stílusért felelő nyelv. A weblapom CSS-ét nem támogatja semmiféle keretrendszer. Funkcionalitását támogatta a JavaScript azért ezt választottam mivel a JavaScript alapvetően kompatibilis a Backendemmel is. Azért Node.js lett a választott nyelv, mert sokkal egyszerűbb számomra, mint a PHP átláthatóbb és kevesebb a hiba lehetőség mivel nincsen szükség minden kód sorom után \$-re

- prisma/client: Ez felel az adatbázishoz való hozzáféréshez.
- argon2: A jelszó titkosítása érdekében használandó.
- cors: Feloldja az Internethez való csatlakozást.
- express: Backend működése miatt hasznos.
- jsonwebtoken : Az adott felhasználónak a sajátos és egyedi tokent készíti el.
- Multer: Kép feltöltés miatt lett használatba véve.

Magát az alkalmazást Visual Studio Code-ban hoztam létre. Mert a Visual Studio Code egyszerűen lehet használni, segítséget is nyújt számunkra. A dokumentáció Wordben íródott, amíg a prezentáció PowerPointba. Jegyzetömböt is használtam a forrás kódok felírásához mivel elégsok film mese vagy sorozat címet rendezőjét feljegyezni. Képek szerkesztése érdekében csak a Paintet használtam, mert az alapvetően van a számítógépemen.

Alapvetően az alkalmazásomnak szüksége van a XAMPPra tehát azt is használtam hozzá mivel az adatbázisom ebben működtethető.

Adatmodell leírása



28. ábra

Zárójelben megszeretnem jegyezni, hogy (felhasznalonev,telszam>User,szulev) nem helyesírási hibák vagy rövidítések, hanem az adatbázisomban szereplők mezőinek a neve.

A kedvencek nevezetű tábla amiatt jött létre, hogy ne legyen redundancia. Egy a többhöz kapcsolat van a Kedvencek és az User tábla között. A User nevezetű táblában megtaláljuk a fontos adatok helyét és annak az adott adatnak a típusát.

Email mező tartalmazza, a felhasználók email címét, felhasznalonev nevezetű mező az adott email címhez tartozó felhasználó nevét tarolja, el az adatbázisban ezek az adatok nem folyhatnak össze ezért egyetlen egy fióké sem fog keveredni a másik felhasználó adataival, a telszam mezőben az adott felhasználó telefonszamat taroljuk el a szulev mezőben a születési dátumot tehát evet hónapot és napot taroljuk el. A jelszo mezőben található meg az argon2 által titkosított jelszó formátum, amit nem lehet feltörni ezért biztonságát nyújt a felhasználók számára és bármiféle félelem nélkül tudjak használni a CinemaSzofitkicaio nevezetű szoftvert. Ezek az adatok nagyon fontosak az alkalmazás működése és a felhasználói élmény

szempontjából, és a megfelelő adatbázisstruktúra segít abban, hogy hatékonyan kezeljük és biztonságban tudjuk azokat. Az User tábla Egy a többhöz kapcsolatban van a Kritika nevezetű táblánkkal, ahol igazából csak kettő tartalom látható, ami a tartalom és egészen idáig csak Stringeket tehát szöveges változókat használtunk az ID-változókon kívül, de ebben a táblában már Int tehát számos változó is található, ami az értékelés névre hallgat ezzel a műsoroknak a 10/ valahányas értékelést kaphat és ezt számok alapján határozzuk meg. Majd az utolsó táblánk a műsorok miatt felel, ez a táblánk kapcsolatban van egy a többhöz kapcsolattal a kritikákhoz illetve szinten egy a többhöz kapcsolattal van a kedvencek nevezetű táblával.

Ezek alapvetően elegendők ahhoz, hogy az egész alkalmazásunk működjön a céljának megfelelően, az egész adatbázis szerkeszthető egyszerűen, ha a későbbiekben esetleg többfelé adatot lehessen feltölteni vagy a saját profilunkat láthassuk az adatok szerkesztésére és saját adataink módosítására. Az adatmodellem Prismában készítettem el és a kapcsolatok következő képpen lett megoldva

```
model User {
  id          Int          @id @default(autoincrement())
  email       String       @unique
  felhasznalonev String
  szulev      String
  telszam     String
  jelszo      String
  kedvencek   Kedvencek[]
  kritikak    Kritika[]

  @@map("users")
}
```

```
model Kedvencek {
  id          Int          @id @default(autoincrement())
  felhasznaloId Int
  musorId     Int
  felhasznalo User @relation(fields: [felhasznaloId], references: [id])
  musor       Musor      @relation(fields: [musorId], references: [id])
}
```

@relation. ez az idegen kulcs, amely összeköti őket.

```
model Musor {
  id          Int          @id @default(autoincrement())
  nev         String
  kiadas      DateTime
  kategoria   String
  leiras      String
  kep         String
  kedvencek   Kedvencek[]
  kritikak    Kritika[]
}
```

@id @default meghatározza az ID-t tehát a fő kulcsot

```
model Kritika {
  id          Int          @id @default(autoincrement())
  felhasznaloId Int
  musorId     Int
  tartalma    String
  ertekeles   Int
  Felhasznalo User @relation(fields: [felhasznaloId], references: [id])
  Musor       Musor      @relation(fields: [musorId], references: [id])
}
```

@relation. ez az idegen kulcs, amely összeköti őket.

Prismában az adott kód sorokkal bármikor lehet változtatni a felépítésen csak ahhoz törölni fogja az eddigi összes adatot, amit valaha is feltöltöttünk az adatbázisunkba. **npx prisma migrate dev --name init**

- Létrehozott egy új SQL
- Végrehajtotta az SQL migrációs fájlt

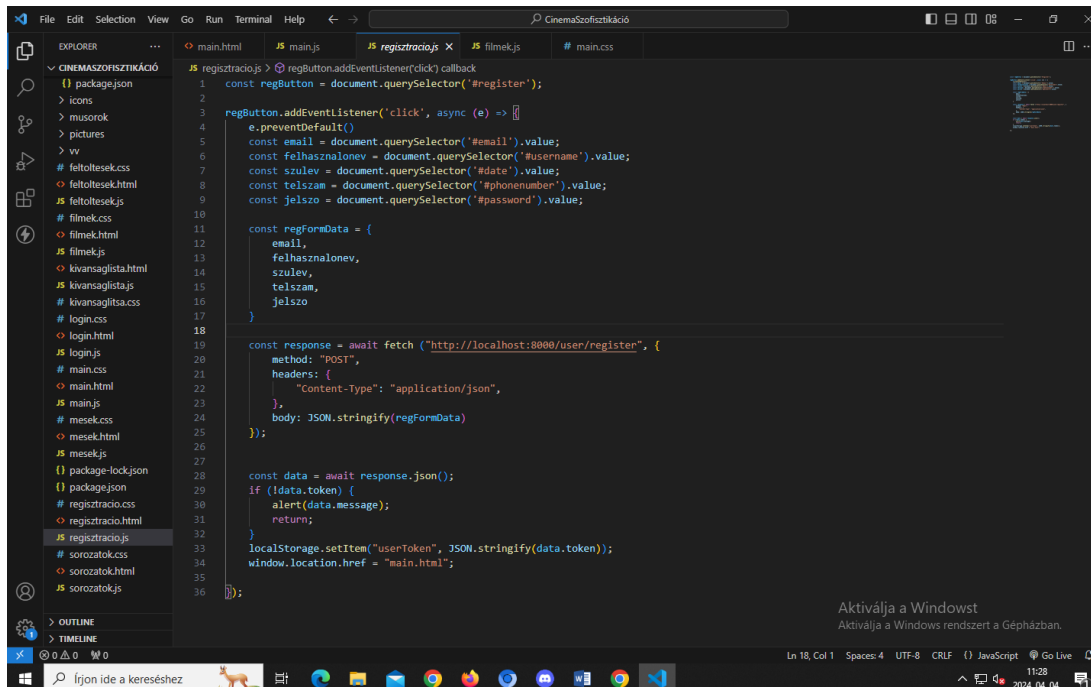
- Személyre szabott Prisma Client API-t hozott létre

Részletes feladatspecifikáció, algoritmusok

```
if(!token || token === undefined) {  
  window.location.href = "login.html";  
}
```

1. egyenlet

Az egyik legfontosabb If utasítás az egész alkalmazásban, mivel ha az nincsen tokenünk tehát nem található a felhasználónk, akkor alap értelmezett Belépés oldalra átdobni minket tehát így is úgys kötelező a regisztráció es a belépés, mert máskülönbem nem látható az alkalmazás többi része.



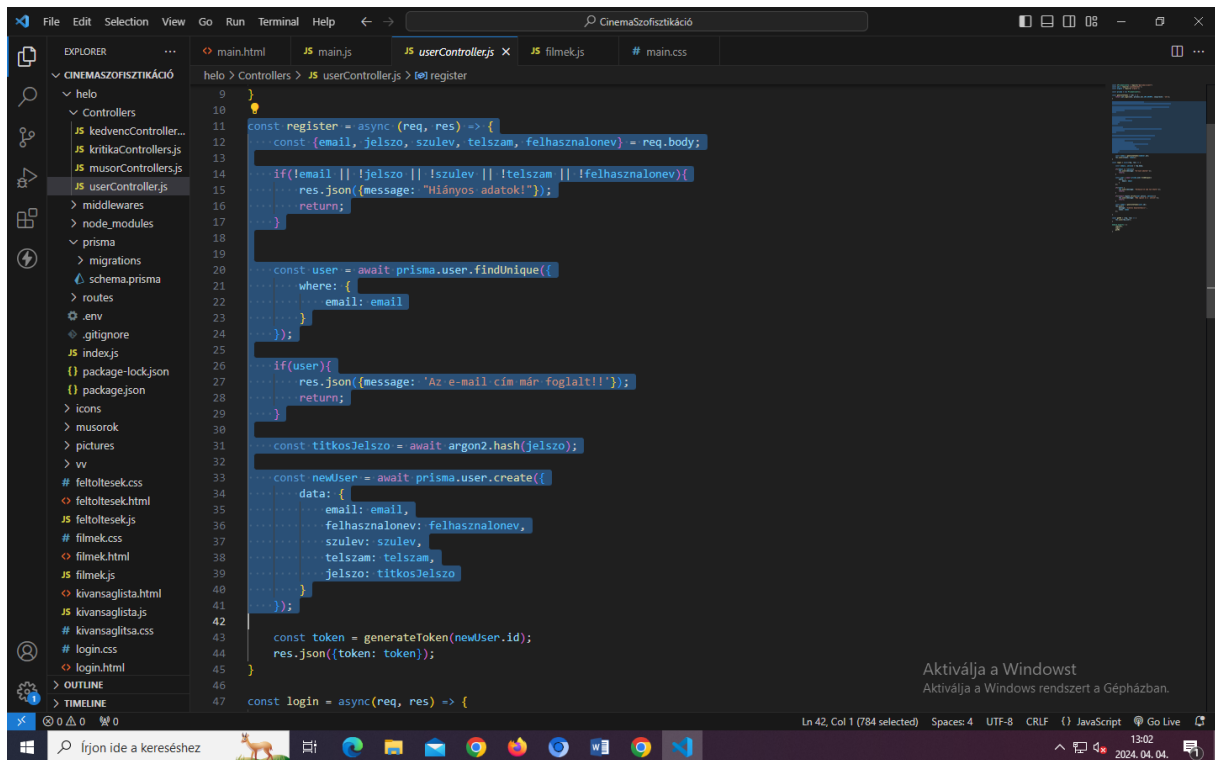
```
1 const regButton = document.querySelector("#register");
2
3 regButton.addEventListener('click', async (e) => {
4   e.preventDefault()
5   const email = document.querySelector("#email").value;
6   const felhasznalonev = document.querySelector("#username").value;
7   const szulev = document.querySelector("#date").value;
8   const telszam = document.querySelector("#phonenumber").value;
9   const jelszo = document.querySelector("#password").value;
10
11   const regFormData = {
12     email,
13     felhasznalonev,
14     szulev,
15     telszam,
16     jelszo
17   }
18
19   const response = await fetch("http://localhost:8080/user/register", {
20     method: "POST",
21     headers: {
22       "Content-Type": "application/json",
23     },
24     body: JSON.stringify(regFormData)
25   });
26
27   const data = await response.json();
28   if (data.token) {
29     alert(data.message);
30     return;
31   }
32   localStorage.setItem("userToken", JSON.stringify(data.token));
33   window.location.href = "main.html";
34
35 });
```

29. ábra

Regisztráció. js szolgál arra, hogy regisztrálni tudjunk, elsősorban lekérjük a HTML oldalunkról a regisztrálásra szolgáló gombot. Ha ezzel megvagyunk, jöhetnek a személyes adatokra fenntartott mezők. Szükséges ezekhez is a html oldalon található nevek.

Utána jöhetnek az adatok lekérései, és az alkalmazásunk eltudja majd küldeni az adatbázisunknak egy új felhasználó adatait minden felhasználónak van egy sajátos Tokenje ami által tudja használni az alkalmazást, és ezáltal tudunk regisztrációt elvégezni majd az adtok

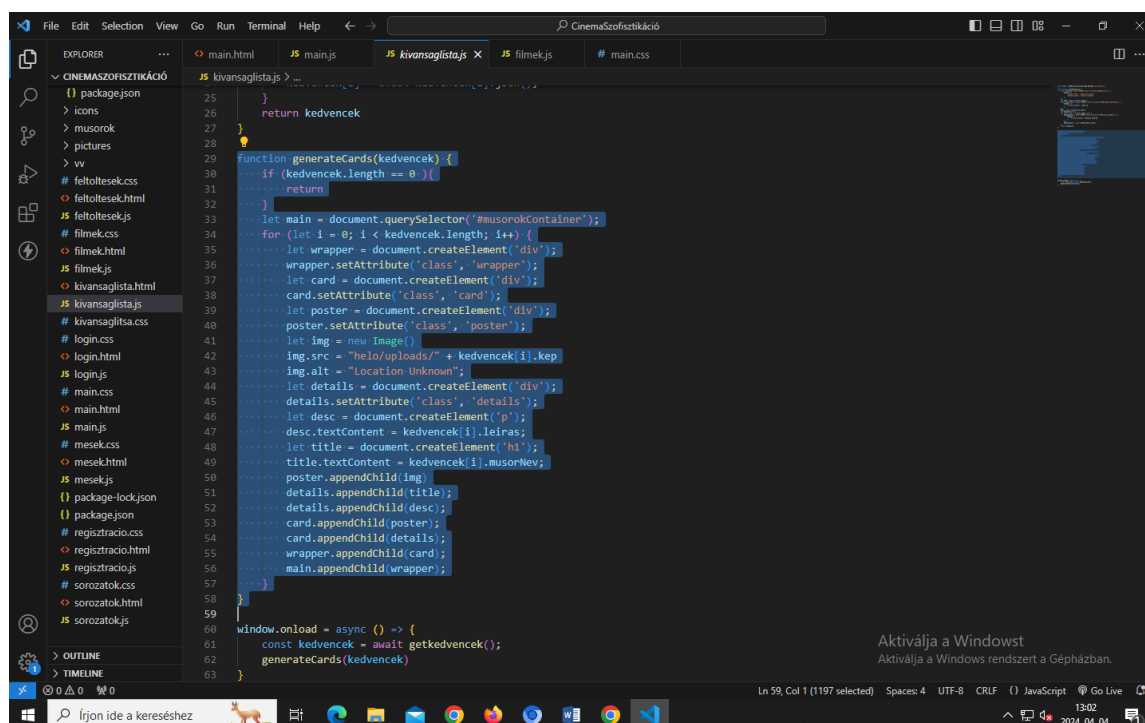
megmaradnak mert el raktározódnak és emiatt fog a belépésünk is működni.



```
9 }
10
11 const register = async (req, res) => {
12   const {email, jelszo, szulev, telszam, felhasznalonev} = req.body;
13
14   if(!email || !jelszo || !szulev || !telszam || !felhasznalonev){
15     res.json({message: "Hiányos adatok!"});
16     return;
17   }
18
19   const user = await prisma.user.findUnique({
20     where: {
21       email: email
22     }
23   });
24
25   if(user){
26     res.json({message: 'Az e-mail cím már foglalt!'});
27     return;
28   }
29
30   const titkosJelszo = await argon2.hash(jelszo);
31
32   const newUser = await prisma.user.create({
33     data: {
34       email: email,
35       felhasznalonev: felhasznalonev,
36       szulev: szulev,
37       telszam: telszam,
38       jelszo: titkosJelszo
39     }
40   });
41
42   const token = generateToken(newUser.id);
43   res.json({token: token});
44 }
45
46 const login = async (req, res) => {
```

2. egyenlet

Szintén a regisztrációhoz tartozik, lekérjük aza adatokat mikre is van szükségünk, hogy regisztrálni tudjon a felhasználó. Az IF águnkban is láthatjuk hogy ha esetleg nincsen meg minden megfelelő adat tehát az email címe a jelszava a sültetési éve telefonszáma vagy esetleg a felhasználó név, akkor ki jelzi számunkra a problémát és üzenetet is küld mellé, hogy tudjuk is hogy az a probléma, hogy valamelyik adatunk hiányzik. ha az az email cím foglalt is jelzi nekünk. Az argon2 titkosítja a felhasználónak a sajátos egyedi jelszavát, hogy ne lehessen feltörni a fiókját, az argon2 a titkosított adatot küldi fel az adatbázisba, hogy ténylegesen biztonságos legyen. Illetve itt található meg az új felhasználó létrehozása is.



30. ábra

Kívánság lista funkciója hogy ide menthessük el a kedvenceink, amiket láttunk és újra szeretnénk nézni vagy bármi ilyesmi. Ez a kódrészlet szolgálja azt a célt, hogy legenerálja a kártyáimat. A kódnak az első felében a kártyának a helyére hivatkozunk, amiket a HTML oldalunkon csináltuk meg es az ott elnevezett elemeknek az divben tartózkodó osztályukat mivel itt fognak elhelyezkedni a kártyaink, más nevén a műsorok, amiket a kedvencekhez, de itt csak az adatok róluk ---> kívánság listához akarunk, hozzá rendelni. Majd a „let img = new Image()” –tol kezdődő kódolás miatt jelenik meg az oldalon a feltöltött filmek vagy mesék, vagy sorozatok oldalról a kívánság lista oldalára az oda kívánt műsor. A feltöltött tartalmak közül egy gomb segítségével működik csak!!!

```
const getMesek = async (req, res) => {
  const mesek = await prisma.musor.findMany({
    where: {
      kategoria: "mese"
    }
  })
  res.json(mesek);
}
```


Ez a függvényünk szolgálja azt a célt, hogy az összes mesét kikeresse számunkra az egész műsor prismabol tehát egy szűrőként funkcionál.

Ezzel a legegyszerűbb a feltöltésnél külön válogatni mind a három műfajt mivel a json Könnyű adatsere-formátum

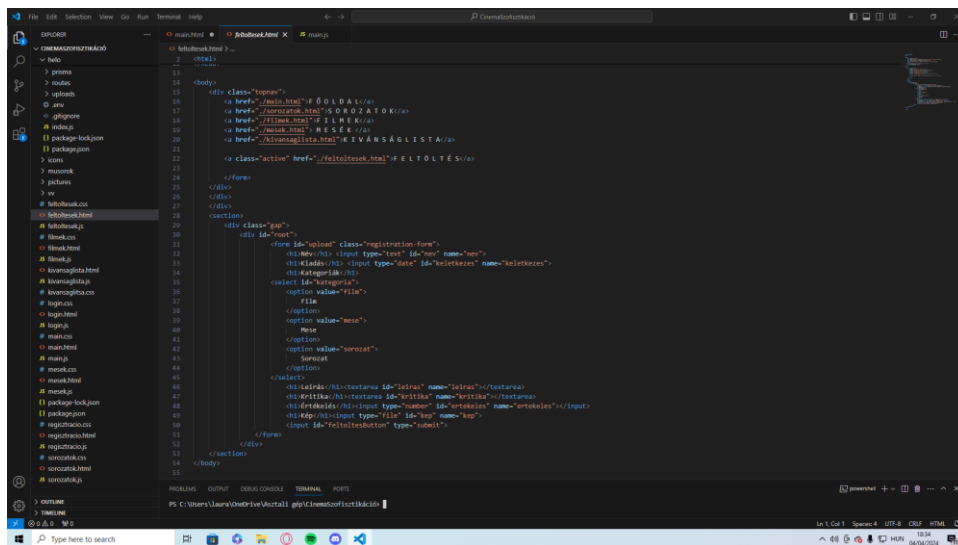
```
const getFilmeK = async (req, res) => {
  const filmek = await prisma.musor.findMany({
    where: {
      kategoria: "film"
    }
  })
  res.json(filmeK);
}
```

```
const getSorozat = async (req, res) => {
  const sorozatok = await prisma.musor.findMany({
    where: {
      kategoria: "sorozat"
    }
  })
  res.json(sorozatok);
}
```

Mind a három kategóriában ugyan úgy kell hivatkozni csak a változó neve fog megváltozni mivel ugyan olyan el nevezéssel nem igazan működne az alkalmazásunk illetve meg zavaró tényező is lenne. A where nevezetű kóddal tehát magyarul a hol nevezetű kóddal a kategórián belül tudjuk kiválasztani, hogy éppenséggel melyikre is lenne szükségünk és ezeket a változókat meg kell hívnunk pontosabban exportálnunk, hogy minden mással is kompatibilis legyen, mert máskülönben teljesen feleslegesen csináltuk ezt meg.

```
module.exports = {
  getMusor,
  getFilmeK,
  getMesek,
  getSorozat,
}
```

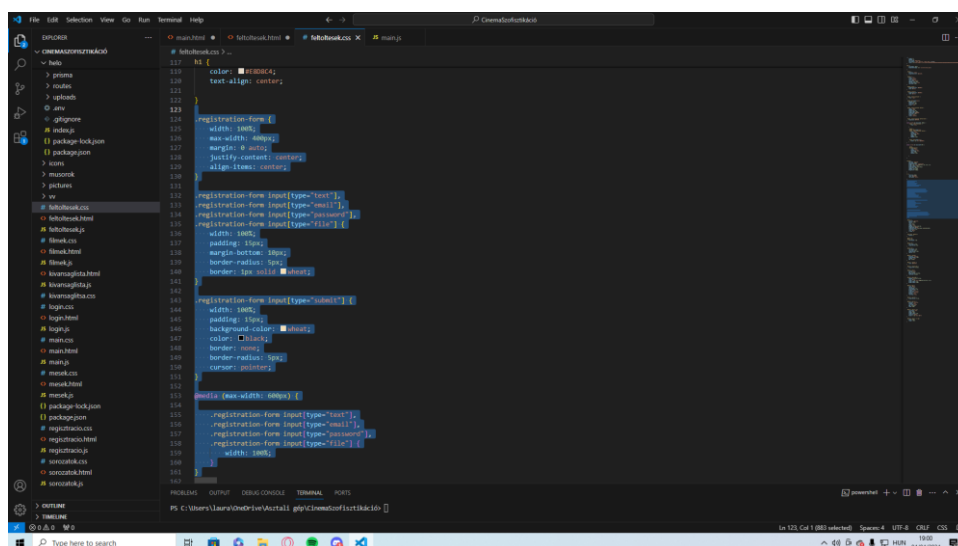
Forráskód



A weblapom html kódolásának az egyik részlete, ami a feltöltések nevezetű oldal.

Az oldal felépítése alapvetően egy html űrlap

- A name attribútumok használata lehetővé teszi az űrlapadatok azonosítását a szerver oldalon.
- Az <input> elemek különböző típusokat tartalmaznak, például text, date, number és file.
- A select elem a kategóriák kiválasztására szolgál egy legördülő listából.
- A <textarea> elem a hosszabb szöveges adatok bevitele céljából van jelen.
- Az ID az a JavaScript kódban vagy a CSS stílusokban használatos esetleg meg Class lehetne.



Ez itt a feltöltések oldalnak a CSS verziója is, ahol az ID felhasználása történt meg.

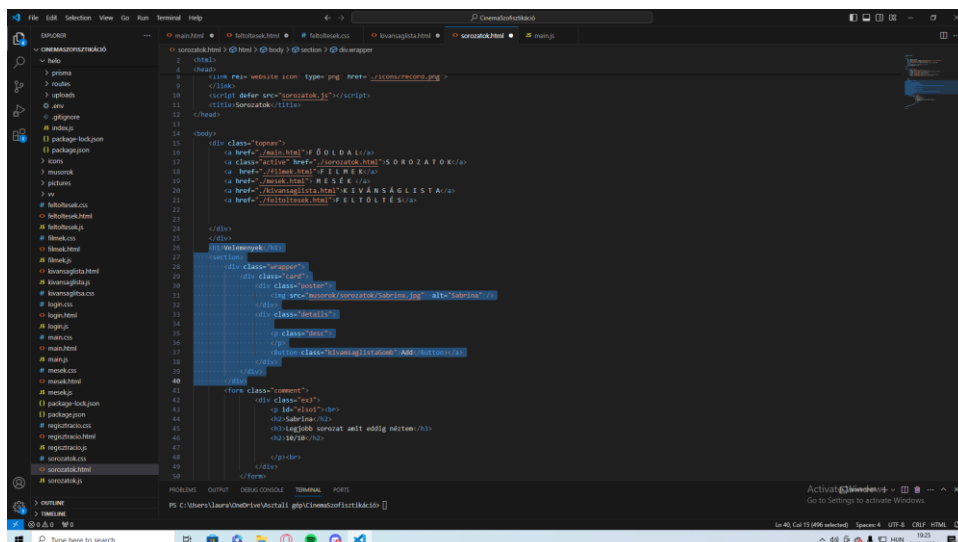
```

<div class="topnav">
  <a href="/main.html">F Ő O L D A L</a>
  <a href="/sorozatok.html">S O R O Z A T O K</a>
  <a href="/filmek.html">F I L M E K</a>
  <a href="/mesek.html"> M E S É K </a>
  <a class="active" href="/kivansaglista.html">K I V Á N S Á G L I S T A</a>
  <a href="/feltoltesek.html">F E L T Ő L T É S</a>
</div>

```

navigációs sáv, amely linkeket tartalmaz más oldalakra való kattintással vehető használatba.

href megadja az adott oldal URL-jét. class="active" azt jelenti, hogy ez a link aktív, vagyis az aktuális oldalra mutat. Ez az összes html oldalunkon megtalálható mivel egységesen nézki és ugyan az a működése elve, mint a többinek.



<h1>Velemenyek</h1> Témáját jelzi.

<div class="wrapper">: Ez egy külső konténer

<div class="card">: Ez egy kártya

<div class="poster">: képét tartalmazó rész.

Tesztelési dokumentáció

Az alkalmazáson regisztrációs felületét láthatjuk először.

Minden mezőt ki kell töltenünk vagy máskülönben hiba jelentkezi, amit nagyon szépen feldob és tökéletesen elmondja, hogy mi is a problémája hiányosak-e az adatok vagy esetleg az email cím használatban e van.

Belépés oldalon is hasonló problémákba ütközhetünk, ha hibás a jelszó vagy az email cím.

Miután ezekkel megvagyunk igazából a feltöltések oldalra nyomva láthatjuk az összes mezőt, amit ki kell töltenünk mivel máskülönben “Hiányzó adatok! Szöveggel rendelkező hiba üzenet jelenik meg ennek az orvoslását igazából annyival tudjuk kijavítani, hogy minden mezőt a nevének, illetve céljának megfelelően kitöltjük.

Oda kell figyelni arra is, hogy a kép feltöltéséről se feledkezünk meg mivel az is fontos adat es az adja ki a szoftverünkön lényegesebb személyre szabottabb részét, de ha ezzel mind megvagyunk akkor szinten egy felrúgó ablakocska jelzi, hogy az adott műfajban levő műsort sikeresen feltöltöttük es ezt szimplán csak le kell okozunk!

Mivel az alkalmazásomnak nincsen adminja mivel ez minden felhasználónak a sajátos személyisége szerint formázható ezért nincs szükségé a weblapnak adminra. Az argon2 miatt biztonságosan kódoltak a jelszavak, ezt az adatbázisba láthatjuk. Ez is egy tökéletes definíció A tesztelés alátámasztására mivel bármikor megnézhetjük a hashelt jelszavakat viszont semmit sem fogunk belőle érteni. Fontos még megjegyezni, hogy ezek bizalmas adatok evégett van erre szükség.

A fejlesztés közben adódott hibákat orvosolni tudtam.

Problémás volt az elején az adatbázisban a dátum es az Intek tehát a számokkal való adatbevitel, mivel valahol át kellett konvertálni stringbol tehát szöveges formátumból ahogyan itt is látható "formData.append('keletkezes', keletkezes.toString());". A XAMMP-ban a hostolási problémák is akadtak mivel nem teljesen mindegy, hogy 33000 vagy éppen 8000.

A kódolás es fejlesztés folyamatában a fetchel is adódtak problémák, amik igazából, ha picit figyelmesebb vagyok akkor nem történt volna meg mivel egy adott adatbázisból kellett szűrőzni a műsorok kategóriáját ezért sokszor crashelt ki és az egész szoftvert megölte.

Illetve a saját hibáim, hogy a backendemet nem indítottam el és evégett se működött tehát figyeljünk oda arra, hogy ténylegesen mindent letelepítettünk és elindítottunk, illetve összekapcsoltunk.

Továbbfejlesztési lehetőségek

A későbbiekben, ha szükség lenne rá szívesen csinálnék egy olyan opciót, ahol a felhasználók tudnak ajánlatokat tenni arra, hogy miként is lehetne fejlesztéseket behozni a szoftverre ahogy nekik az tetszene.

De a saját ötleteimet is megszeretném osztani, ami első sorban a személyes fiokunk ki bővítése. Lényegében arról szólna, mint spotify-on a közös lejátszólista ez azt jelenti, hogy megtudnának maguknak hívni a barátaik közül vagy bármiféle kapcsolatukból adott személyeket és oda közösen tudnának műsorokat feltölteni. A következő pedig az értékelések megosztása több felhasználó között tehát látnánk, hogy ki mit nézett meg és milyen véleménnyel volt az adott tartalomról. Több adatot lehetne feltölteni és részletesen dokumentálni, hogy kik szerepeltek benne, ki rendezte, írta az adott műsort. Külön oldal az újdonságokra, mint fejlesztés. Viszont műsorok szempontjából is lenne egy ilyen oldal, hogy milyen műsorok jelentek meg az adott napon ezt naptárból lehetne nyomon követni.

Esetleg a filmeket, meséket, sorozatokat részletesebb elosztásban is ezalatt arra gondolok, hogy a műfajok alapján is lehetne őket számontartani.

Irodalomjegyzék, forrásmegjelölés

Prisma alapok:

<https://www.prisma.io/docs>

Argon2:

<https://www.npmjs.com/package/argon2>

Cors:

<https://www.npmjs.com/package/cors>

Nodemon:

<https://www.npmjs.com/package/nodemon>

Multer:

<https://www.npmjs.com/package/multer>

CSS

<https://www.w3schools.com/css/>

HTML

<https://www.w3schools.com/html/default.asp>

JavaScript

<https://www.w3schools.com/js/default.asp>

Plágium-nyilatkozat

Alulírott (név)
..... (szül.hely)
..... (szül.idő)
..... (anyja neve)

jelen nyilatkozat aláírásával kijelentem, hogy a
című záródolgozat (a továbbiakban: dolgozat) önálló munkám, a dolgozat készítése során
betartottam a szerzői jogról szóló 1999. évi LXXVI. tv. szabályait, valamint a Békéscsabai
Szakképzési Centrum által előírt, a dolgozat készítésére vonatkozó szabályokat.

Tudomásul veszem, hogy a dolgozat esetén plágiumnak/szerzői jogsértésnek számít:

- szó szerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más szerző publikált gondolatainak saját gondolatként való feltüntetése

Kijelentem továbbá, hogy a dolgozat készítése során az önálló munka kitétel tekintetében a
konzulenszt, illetve a feladatot kiadó oktatót nem tévesztettem meg.

Jelen nyilatkozat aláírásával tudomásul veszem, hogy amennyiben bizonyítható, hogy a
dolgozatot nem magam készítettem vagy a dolgozattal kapcsolatban szerzői jogsértés ténye
merül fel, a Békéscsabai Szakképzési Centrum a dolgozatot elégtelennek minősíti.

Békéscsaba, 20hónap

Tanuló aláírása

