



ZÁRÓDOLGOZAT

Készítették:

Szabó Szabolcs – Jakab Balázs Erik – Puda Ádám

Konzulens:

Farkas Zoltán

Miskolc

2023

Miskolci SZC Kandó Kálmán Informatikai Technikum

Miskolci Szakképzési Centrum

SZOFTVERFEJLESZTŐ ÉS TESZTELŐ SZAK

ZÁRÓDOLGOZAT

Filmezz

online filmkereső weboldal

Szabó Szabolcs – Jakab Balázs Erik – Puda Ádám

2022-2023

Tartalomjegyzék

1.BEVEZETÉS:	4
2.Felhasznált technológiák és programok:	5
2.1 phpMyAdmin:	5
2.2 XAMPP:	6
2.3 React:	8
2.4 React axios:	10
2.5 WPF:	12
2.6 WEB API:	14
2.7 Entity Framework:	15
2.8 .NET keretrendszer:	17
2.9 Visual Studio 2022:	18
3. FELÉPÍTÉS	19
4. FELHASZNÁLÓI DOKUMENTÁCIÓ	20
4.1 Adatbázis	20
4.2 FrontEnd:	21
4.2.1 Bejelentkezési felület:	22
4.2.2 Regisztrációs felület:	22
4.2.3 Kezdőlap és kategória szerinti rendezés:	24
4.2.4 További információ egy bizonyos filmről:	25
4.2.5 Rólunk menüpont:	26
4.2.6 Kijelentkezés:	26
4.3 WPF asztali alkalmazás:	27
4.3.1 Bejelentkező felület:	28
4.3.2 Dashboard Page:	28
4.3.3 Users menüpont:	29
4.3.4 Films opció:	31
4.4.4 Kijelentkezés:	32
4.4.5 A telepítés menete:	33
5.Továbbfejlesztési opciók	35
6.Irodalomjegyzék	37

1.BEVEZETÉS:

A csoportmunka keretében csapatunk egy filmkereső weboldalt szeretett volna létrehozni és fejleszteni. Választásunk azért esett egy ilyen oldalra, mert manapság, ha egy tartalom fogyasztó filmet szeretne megtekinteni előtte informálódik az interneten.

Először megnézi a film előzetesét majd ezt követően különböző felületeken megnézi az értékeléseket. Amennyiben pozitív kritikát kapott az adott film akkor azt a felhasználó nagy eséllyel meg fogja tekinteni. Viszont, ha a kritikák és értékelések negatívan jellemeznek egy filmet, akkor annak nézőszáma jelentősen csökkenhet.

Ezen szempontokat figyelembe véve hoztunk létre egy filmkereső weboldalt, amely segít eldönteni az adott felhasználónak, hogy az esti filmezéshez milyen műsort is válasszon magának/társainak, ízlés szerint, akár kategóriánként rendezve, akár rákeresve egy olyan filmre, amit egy barát ajánlott. A weboldal azoknak is hasznos, akik egyetlen filmről szeretnének több információt megtudni.

A jövőt tekintve az általunk készített oldal tökéletes alapot nyújthat további fejlesztésekhez, amellyel a filmeket értékelni, kölcsönözni is lehetne és online streamelni bármilyen eszközön. Ennek megvalósítására viszont már több időre lenne szükség, amely sajnos nem állt csoportunk rendelkezésére.

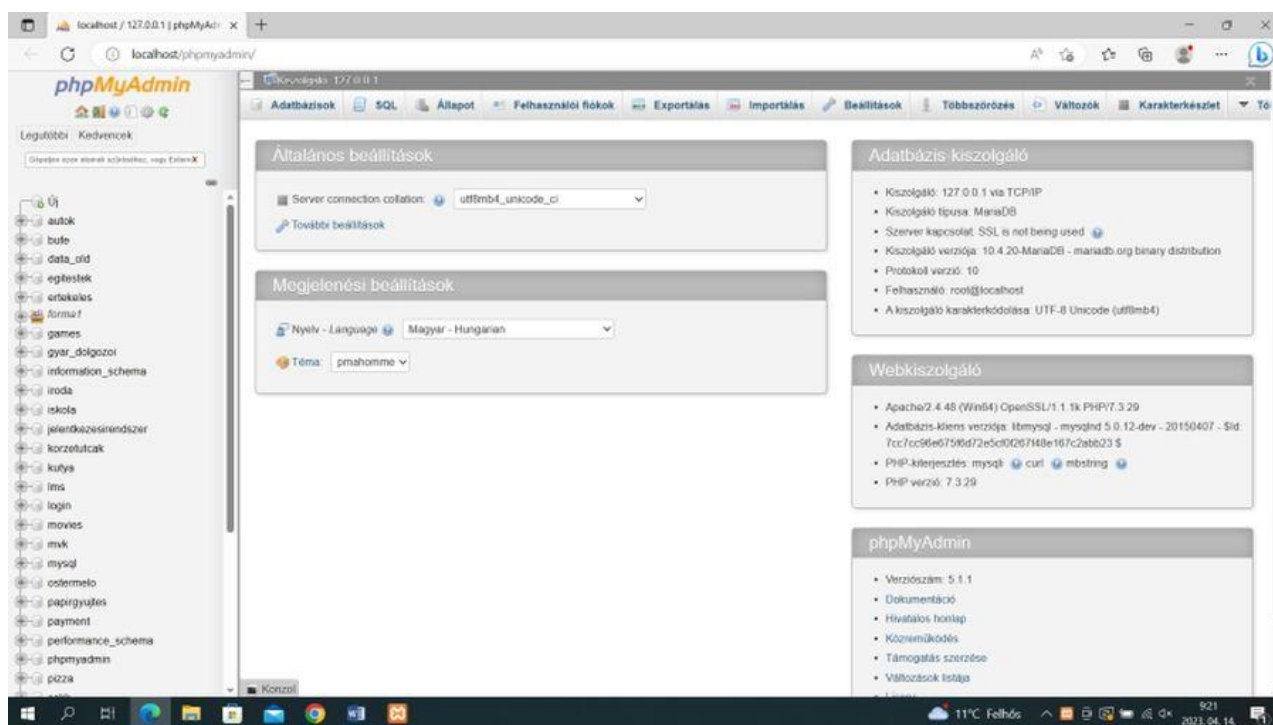
2.Felhasznált technológiák és programok:

2.1 phpMyAdmin:

A phpMyAdmin egy nyílt forráskódú szoftvereszköz, ami PHP nyelven íródott. Támogatja a MariaDB és a MySQL különböző típusú műveleteit. Fő feladata és célja a MySQL webes adminisztrációjának kezelése. Adatbázis kezeléséhez ez az egyik legnépszerűbb alkalmazás.

Ezzel a szoftverrel létrehozhatjuk, frissíthetjük, törölhetjük, módosíthatjuk, emellett importálhatjuk és exportálhatjuk a MySQL adatbázis tábláit. Ezen felül a kapcsolatokat, táblák, oszlopok, indexek, jogosultságok és felhasználók kezeléséhez szükséges műveletek széles körét támogatja, amelyeket felhasználói felületen keresztül végezhetünk el.

A phpMyAdmin 72 nyelvre van lefordítva ezáltal az emberek széles köre könnyen használhatja ezt a szoftvert. Ez egy GUI alkalmazás (felhasználói felület) továbbá webes felületet is biztosít és bármilyen szerveren futtatható. Webalapúságának révén bármilyen számítógépről hozzáférhetünk. A következő képen látható a felület szerkezete és kialakítása:



1.kép: phpMyAdmin felület

A phpMyAdmin előnyei:

- több eredménykészletet képes megjeleníteni lekérdezések és tárolt eljárások segítségével
- támogatja az idegen kulcsokat és az InnoDB táblákat
- képes nyomon követni az adatbázisokon, nézeteken és a táblákon végzett változtatásokat
- az adatbázisok elrendezéséről PDF grafikákat készíthetünk
- többféle formátumokba exportálható, mint például XML, CSV, PDF, TXT (szöveg) és táblázat
- támogatja a mysqli bővítményt
- előre definiált függvények használata, a tárolt adatokat képes bármilyen formátumra alakítani (például: BLOB adatok képként vagy letöltési hivatkozásként)
- lehetőséget biztosít arra, hogy az adatbázisokat különböző formátumokba lehessen menteni

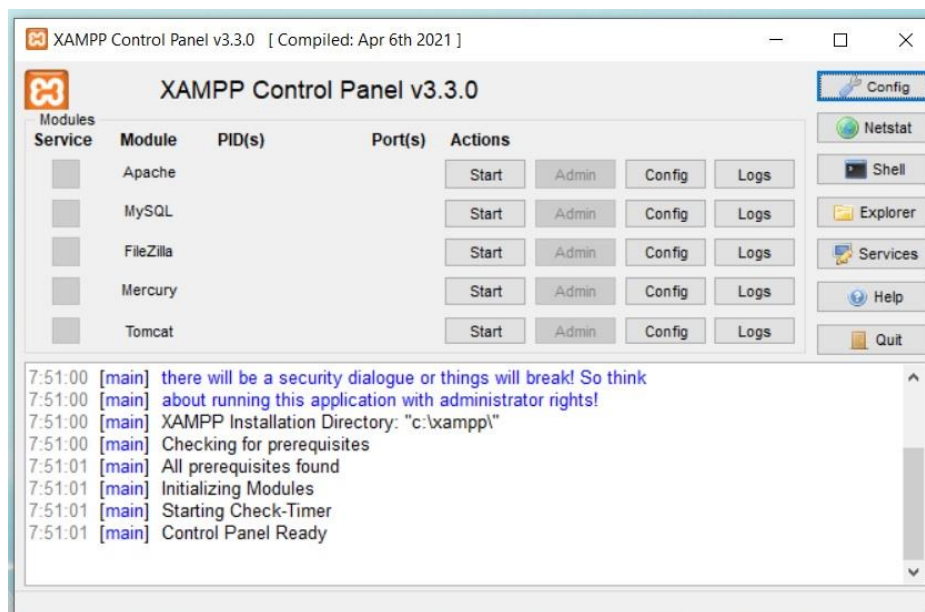
Hátrányai:

- ez egy egyszerű felület viszont a kezdők számára nehéz megtanulni, elsajátítani az oldal működését
- nehéz telepíteni, mert a telepítés előtt további három szoftvereszközre van szükség, amelyek az Apache server, PHP és a MySQL.
- az utóbb említett eszközöket egyesével kell telepíteni, míg a XAMPP már tartalmazza ezeket. A XAMPP-ról a későbbiekben lesz szó.
- nincs séma megjelenítés
- nem rendelkezik automatikus fordítással

2.2 XAMPP:

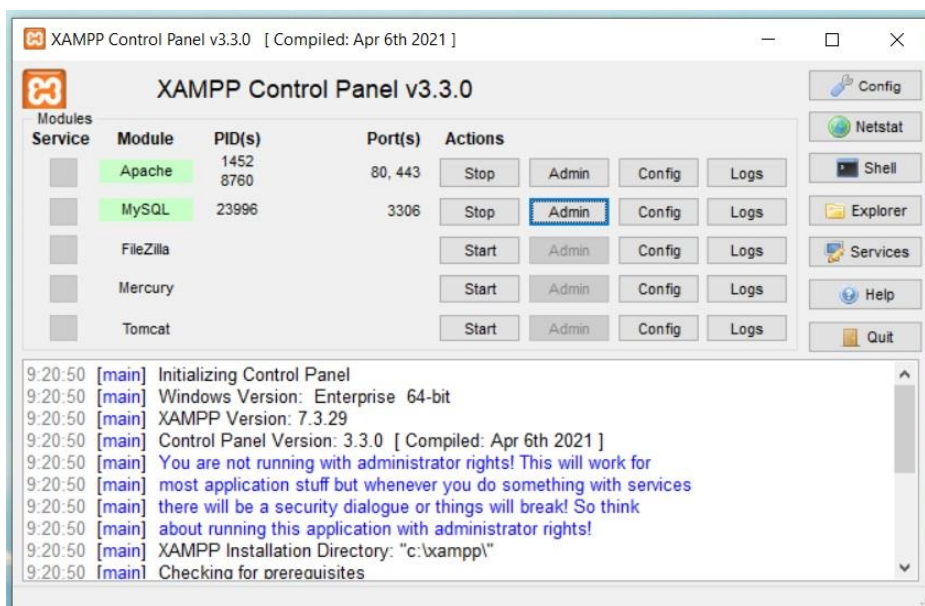
Ez egy asztali alkalmazás, amelyet szükséges telepíteni a Windowst vagy MACOS-t futtató számítógépre. Ennek a programnak az a célja, hogy az előzőekben bemutatott phpMyAdmin felülethez megteremtse a kapcsolatot a webszerver segítségével. A webszerver feladata, hogy megkeresse a kívánt oldalt, lefuttat egy PHP kódot majd ezt követően elküld egy HTML oldalt a böngészőnek.

A XAMPP egy ingyenes program, amely telepíti és konfigurálja a számítógépre az Apache webservert és a MySQL-t. Továbbá ez a felület nagyon hasznos, mert az Apache-nak és a MySQL-nek nincsen saját grafikus felülete. Az alábbi képen látható a XAMPP program:



2.kép: a XAMPP Program

Elindítást követően ez a felület nyílik meg. A Start gombra történő kattintás után a Module menüpontban az Apache és a MySQL felirat zöldre vált és a Port(s) menüpontban az előbb említett modulok portszámot kapnak. A MySQL modulnál lévő Admin gombra kattintást követően megnyílik a phpMyAdmin felülete.

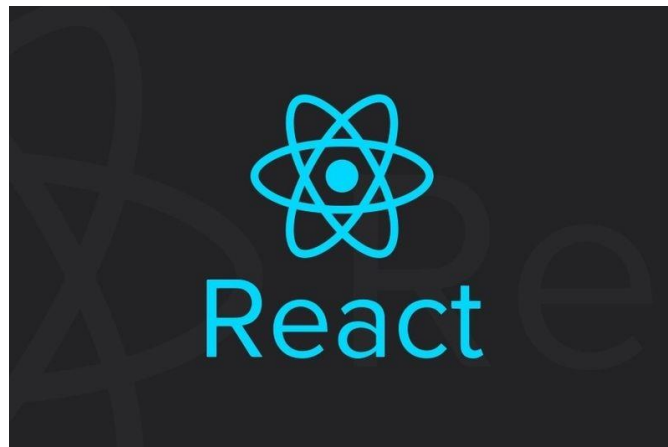


3. kép: az elindított modulok

2.3 React:

A React egy felhasználói felületeket építő könyvtár. Nem keretrendszer és nem csak kizárólag a webre korlátozódik. Más könyvtárakkal együtt használják bizonyos környezetekre való rendereléshez. A React Native például mobilalkalmazások fejlesztésére használható. A fejlesztők a Reactot a ReactDOM-mal együtt használják weboldalak építéséhez. Emellett ugyanazok a problémák megoldására alkalmazzák, mint más valódi webfejlesztő keretrendszereket. Amikor a Reactra „keretrendszerként” hivatkozunk, akkor ezzel a köznyelvi értelmezéssel dolgozunk.

Ennek a könyvtárnak az elsődleges célja, hogy minimalizálja a hibákat, amelyek a fejlesztők felhasználói felületének építése során keletkeznek. Ezt a komponensek (önálló, logikus kódrészletek, amivel a felhasználó felület egy részét írják le) alkalmazásával érik el.



4.kép: React logó¹

Felhasználási esetek

Más keretrendszerekkel ellentétben a React nem támaszt szigorú szabályokat a kódolási feltételek vagy a fájlszervezés körül. Ez lehetővé teszi a csapatok számára, hogy a számukra legjobban működő szabályokat állítsák be és a Reactot olyan módon fogadják el, ahogyan csak szeretnék. Továbbá képes kezelni egyetlen gombot, egy felület néhány részét vagy egy alkalmazás teljes felhasználói felületét.

Bár a React használható egy felület kis darabjaira, nem olyan könnyű „bedobni” egy alkalmazásba, mint egy olyan könyvtárat, mint a jQuery vagy akár egy keretrendszer, mint a Vue. Sokkal megközelíthetőbb, ha a teljes alkalmazás React-tal készül.

¹ React logo: <https://www.loginradius.com/blog/engineering/lazy-loading-in-react/> , 2023.04.17.

Ezen kívül számos fejlesztői élményt nyújtó előnye például a JSX-el történő interfészek írása fordítási folyamatot igényel. Egy olyan fordítóprogram, mint a Babel hozzáadása egy weboldalhoz lassúvá teszi a rajta lévő kódot, ezért a fejlesztők gyakran egy build-lépéssel állítják be az ilyen eszközöket. A React vitathatatlanul nagy eszközigényű, de megtanulható.

JavaScript használata a Reactban

A React a modern JavaScript jellemzőit használja fel számos mintájához. A legnagyobb eltérés a JavaScript-től a JSX szintaxis használata. A JSX kiterjeszti a JavaScript szintaxisát, így a HTML-szerű kód élhet mellette. Például:

```
const heading = <h1>Mozilla Developer Network</h1>
```

Ezt a címkonstansot JSX-kifejezésnek nevezzük. A React ezt tudja használni arra, hogy megjelenítse ezt a `<h1>` taget az alkalmazásunkban.

Természetesen a böngésző nem tudja segítség nélkül olvasni a JSX-et. Ha lefordítjuk (egy olyan eszközzel, mint a Babel vagy a Parcel), a fejléc kifejezésünk így nézne ki:

```
const header = React.createElement("header", null,  
  React.createElement("h1", null, "Mozilla Developer Network")  
);
```

A fordítási lépést kihagyhatjuk, és a `React.createElement()` segítségével magunk is megírhatjuk a felhasználói felületet. Ezzel azonban elveszítjük a JSX deklaratív előnyeit, és a kódunk nehezebben olvashatóvá válik.

A fordítás egy extra lépés a fejlesztési folyamatban, de a React közösségben sok fejlesztő úgy gondolja, hogy a JSX olvashatósága megéri. Ráadásul a modern front-end fejlesztés szinte mindig magában foglal egy build folyamatot. Az olyan népszerű eszközök, mint a Babel, már eleve tartalmazzak JSX-támogatást, így nem kell konfigurálni a fordítást.

Mivel a JSX a HTML és a JavaScript keveréke, egyes fejlesztők intuitívnak (megérzettnek) találják. Mások azt mondják, hogy a kevert jellege miatt zavaros. Ha azonban egyszer az ember megbarátkozik vele, gyorsabban építhetőek a felhasználói felületek, és másoknak is lehetővé teszi, hogy egy pillantással jobban megértsék a kódot.

2.4 React axios:

Az axios egy HTTP klienskönyvtár, ami lehetővé teszi, hogy kéréseket intézzünk egy adott végponthoz. Ez lehet egy külső API vagy egy saját Backend szerver. A kérés benyújtását követően az API ennek megfelelően kezeli az elvárt műveletet. Ez lehet GET (lekérés), POST (hozzáadás), PUT (feltöltés), DELETE (törlés).

Például, ha egy GET kérésre van szükség, akkor az a cél, hogy az alkalmazásban megjelenítendő adatokat kapjuk vissza.

Számos különböző könyvtárat használhatunk ezeknek a kéréseknek az elkészítéséhez viszont a React a következő gondolatok miatt jobb választás lehet:

- A JSON adatokkal való munkavégzéshez jó alapértelmezésekkel rendelkezik az olyan alternatívákkal ellentétben, mint a Fetch Api.
- Az axios rendelkezik olyan funkciókkal, amelyek megfelelnek bármelyik HTTP módszernek. A GET kérés végrehajtásához a `.get()` metódust használja.
- Kevesebb kóddal többet tud. A Fetch API-val ellentétben csak egy `.then()` visszahívásra van szükség a kért JSON adatok eléréséhez.
- Jobb hibakezeléssel rendelkezik. Axios 400 és 500 tartományú hibákat dob ellentétben a Fetch API-val ahol az állapotkódokat és a hibákat le kell kezelni.
- Mind a szerveren, mind a kliensen használható. Node.js alkalmazás esetén figyelembe kell venni, hogy az axios a böngészőtől különálló környezetben is használható.

A telepítés menete

Az Axios és a React használata nagyon egyszerű folyamat. Három dologra van szükség:

1. Egy meglévő React projekt
2. Az axios telepítése az npm/yarn segítségével
3. API végpont a kérelmek küldéséhez

Ha van egy meglévő React projekt, akkor csak telepíteni kell az axios-t az npm (vagy bármely másik csomagkezelő) segítségével:

```
PS C:\Users\Szabó Szabolcs> npm install axios
```

5.kép: axios telepítés

Példa egy GET kérésre:

Adatok lekérdezéséhez vagy visszakereséséhez GET kérést kell végrehajtani. Jelen példa az egyes hozzászólásokra fog kérést készíteni. A végponton látható, hogy az első bejegyzés kapható meg a /posts végpontból:

```
import axios from "axios";
import React from "react";

const baseUrl = "https://jsonplaceholder.typicode.com/posts/1";

export default function App() {
  const [post, setPost] = React.useState(null);

  React.useEffect(() => {
    axios.get(baseUrl).then((response) => {
      setPost(response.data);
    });
  }, []);

  if (!post) return null;

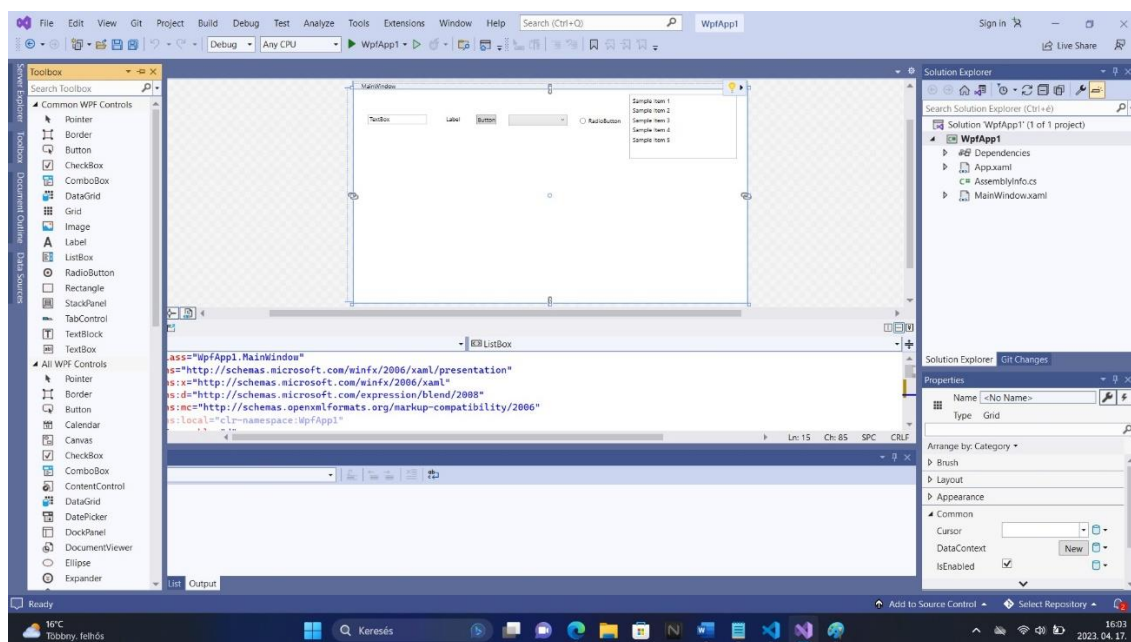
  return (
    <div>
      <h1>{post.title}</h1>
      <p>{post.body}</p>
    </div>
  );
}
```

Ahhoz, hogy a kérés lefusson a useEffect() beépített eljárást kell alkalmazni. Ez magában foglalja az axios importálását, a .get() metódus használatát egy GET kérés elkészítéséhez a végponthoz és egy .then() visszahívás alkalmazását az összes válaszadat kinyeréséhez.

A válasz objektumként kerül visszaadásra. Az adatok (amelyek ebben az esetben egy post-ot jelentenek id, title és body tulajdonságokkal) egy post elnevezésű állapotrészbe kerülnek, amely megjelenik a komponensben. A kért adatok mindig megtalálhatóak a válasz .data tulajdonságában.

2.5 WPF:

A WPF (Windows Presentation Foundation) a Microsoft által fejlesztett GUI keretrendszer, amely a .NET keretrendszerrel használható. A GUI (Graphical User Interface) rövidítése, ami a grafikus felhasználói felületet jelenti. Ilyen például a Windows operációs rendszer grafikus kezelőfelülete mely lehetővé teszi a felhasználóknak az interakciót a számítógéppel. Egy GUI framework segítségével a fejlesztők képesek grafikus alkalmazásokat készíteni a GUI elemek széles választékából ilyenek például a szövegdobozok (textbox), címkék (label), gombok (button) és további más jól ismert elemeket.



6.kép: WPF felülete

Ha nem létezne a GUI framework akkor manuálisan kellene megrajzolni, megszerkeszteni ezeket az elemeket, illetve kezelni az összes felhasználói interakciót például az egér vagy a szöveg input-okat. Ez számottevően sok munka lenne viszont a legtöbb fejlesztő ezen okból kifolyólag GUI framework-öt használ, ami az összes alapvető feladatot képes elvégezni. Ezáltal segíti a fejlesztők, programozók munkáját, hogy kizárólag az alkalmazás fejlesztésével keljen foglalkozniuk.

Manapság jó néhány GUI Framework létezik, de a .NET fejlesztők számára legfontosabb a WinForms és a WPF. Viszont meg kell említeni, hogy a WPF technológiát a Microsoft az idei éven lecseréli a MAUI-ra. A WPF továbbra is megmarad, de a Microsoft közlése alapján közösség által fenntartott projekt-té válik. Frissítéseket tekintve megkapta a .NET 6, és .NET 7 verzióit.

XAML

A XAML (eXtensible Application Markup Language) rövidítése, amely egy egyszerű és leíró programozási nyelv. Ezt a WPF-hez hasonlóan szintén a Microsoft fejlesztette ki alkalmazások felhasználói felületeinek írására. A Windows Presentation Foundation (WPF), UWP és Xamarin Forms alkalmazásokat használó Windows és Mobil appok felhasználói felületeinek készítésére szolgáló nyelv.

Célja egyszerű: felhasználói felületek létrehozása egy XML-hez hasonló jelölőnyelv segítségével. Az XML (Extensible Markup Language) egy egyszerű szövegalapú formátum több tagból felépített információk megjelenítésére.

A legtöbb esetben tervezőt kell használni a XAML létrehozásához viszont közvetlenül kézzel is manipulálható. Az XML formátumot alkalmazza az elemek és az attribútumok számára. Minden komponens egy objektumot képvisel, amely egy típus példánya. Egy típus (osztály, felsorolás stb.) hatókörét egy névtérként határozzuk meg, ami fizikailag a .NET Framework könyvtár egy összeállításában (DLL) található.

```
1 <Window x:Class="WpfAppl.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6     xmlns:local="clr-namespace:WpfAppl"
7     mc:Ignorable="d"
8     Title="MainWindow" Height="450" Width="800">
9
10    <Grid>
11        <TextBox HorizontalAlignment="Left" Margin="29,51,0,0" TextWrapping="Wrap" Text="TextBox" VerticalAlignment="Top" Width="120" Height="20"/>
12        <Label Content="Label" HorizontalAlignment="Left" Margin="183,47,0,0" VerticalAlignment="Top" Width="42" Height="28"/>
13        <Button Content="Button" HorizontalAlignment="Left" Margin="250,51,0,0" VerticalAlignment="Top"/>
14        <ComboBox HorizontalAlignment="Left" Margin="314,51,0,0" VerticalAlignment="Top" Width="120"/>
15        <RadioButton Content="RadioButton" HorizontalAlignment="Left" Margin="458,56,0,0" VerticalAlignment="Top"/>
16        <ListBox d:ItemsSource="{d:SampleData ItemCount=5}" Margin="560,10,21,294"/>
17    </Grid>
18 </Window>
19
```

7.kép: XAML környezete

Az XML-hez hasonlóan a XAML-elemek szintaxisa mindig nyitott szögletes zárójellel (<) kezdődik és zárt szögletes zárójellel (>) végződik. Minden elem tagnek van egy kezdő és egy záró tagje is. Például egy gomb objektumot a <Button> objektumelem képvisel.

Az XAML-ben egy objektumelem egy típust jelöl. A típus lehet egy vezérlőelem, egy osztály vagy más a keretrendszer könyvtárában definiált objektum.

A gyökér elemek

Minden XAML fájlnak rendelkeznie kell gyökérélemmel. A gyökérélem általában tárolóként is működik és meghatározza az elem névterét és alapvető tulajdonságait.

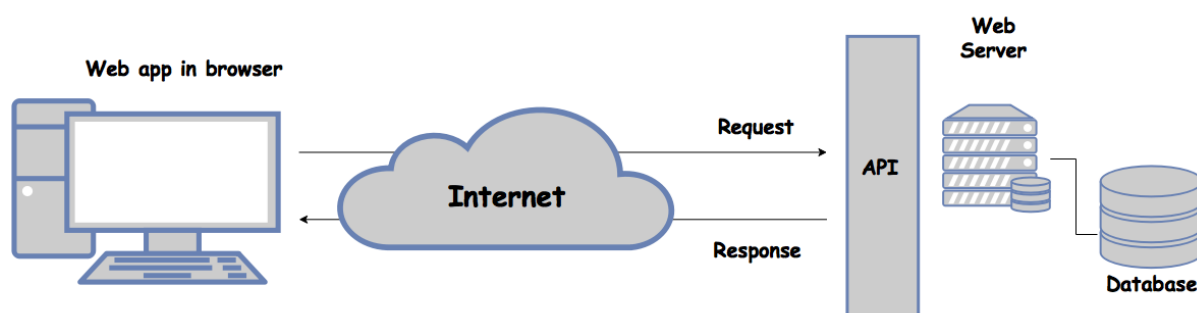
A három leggyakoribb gyökérelem a <Windows/>, <Page/>, és a <UserControl>. A <ResourceDirectory/> és az <Application/> az XAML fájlokban használható másik két gyökérelem.

2.6 WEB API:

API jelentése: Az API egy olyan interfész, amely egy sor olyan funkcióval rendelkezik, amik lehetővé teszik a programozók számára, hogy hozzáférjenek egy alkalmazás, operációs rendszer vagy más szolgáltatások bizonyos jellemzőihez vagy adataihoz.

A **webes API**, mint a nevéből is látszik, egy olyan API a weben, ami a HTTP protokoll segítségével érhető el. Ez egy koncepció(elgondolás) és nem egy technológia. WEB API-t különböző technológiákkal (például Java, .NET stb.) használatával építhetünk. Ennek apropóján a Twitter REST API-jai biztosítják a programozott hozzáférést az adatok olvasásához és írásához. Ezek felhasználásával integrálhatjuk (beépíthetjük) a Twitter képességeit a saját alkalmazásunkba.

ASP.NET Web API: Bővíthető keretrendszer HTTP alapú szolgáltatások létrehozására, amiket különböző platformokon (úgy mint web, windows, mobil stb.) alkalmazásokban lehet elérni. Működése nagyjából megegyezik az ASP.NET MVC típusú webes alkalmazásával, azzal a különbséggel, hogy HTML nézet helyett válaszként küldi az adatokat. Olyan, mint egy web vagy WCF szolgáltatás, azzal a kivétellel, hogy csak a HTTP protokollt támogatja.



8.kép: WEB API ábra²

² WEB API: <https://www.analyticsvidhya.com/blog/2022/10/most-commonly-asked-web-api-interview-questions/>, 2023.04.18.

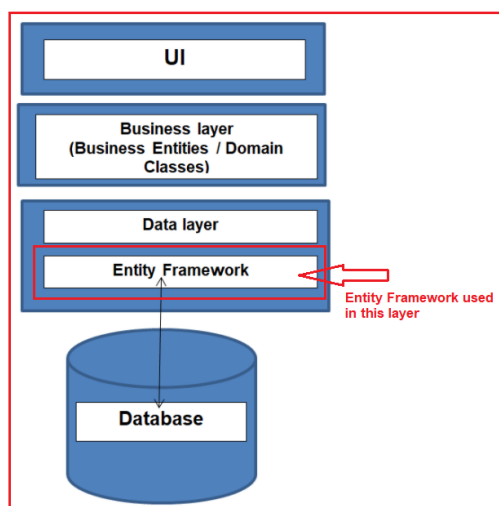
ASP.NET Web API jellemzői

- Ideális platform a RESTFUL szolgáltatások létrehozásához.
- Az ASP.NET tetejére épül és támogatja a kérés/válasz folyamatot.
- Többfajta válaszadat formátumot is engedélyez. Beépített support a JSON, XML, BSON formátumhoz.
- Saját tárhelyen vagy más a .NET 4.0+ verziót támogató webkiszolgálón lehet működtetni.
- HTTP klienst tartalmaz a WEB API kiszolgálóval való kommunikációhoz. Emellett felhasználható ASP.MVC szerveroldalon, Windows Form, konzolos alkalmazásban vagy más környezetben.

2.7 Entity Framework:

Ez egy nyílt forráskódú ORM (Object Relational Mapping) keretrendszer a Microsoft által támogatott .NET alkalmazásokhoz. Lehetővé teszi a fejlesztők számára, hogy az adatokkal a speciális domain-osztályok objektumai segítségével dolgozzanak anélkül, hogy az adatbázis tábláira és oszlopaira fókuszálnának, ahol az adatokat tárolják.

Az Entity Framework révén a fejlesztők magasabb elvonatkoztatási szinten dolgozhatnak, amikor az adatokkal foglalkoznak. Továbbá a keretrendszer segítségével a hagyományos alkalmazásokkal összehasonlítva kevesebb kóddal hozhatunk létre, illetve tarthatunk fenn adatközpontú applikációkat.



9.kép: Entity Framework elhelyezkedése³

³ Entity Framework ábra: <https://dotnettutorials.net/lesson/introduction-to-entity-framework/> , 2023.04.18.

Az előző ábra alapján az Entity Framework az üzleti egységek (domain osztályok) és az adatbázis között helyezkedik el. Az üzleti egységek tulajdonságaiban tárolt adatokat elmenti, valamint az adatbázisból is lekérdezi az adatokat és automatikusan átalakítja azokat a domain osztályok objektumaiba.

Az Entity Framework funkciói

Cross-platform támogatás: A keretrendszer futtatható Windowson, Linuxon és Macen egyaránt.

Modellezés: Az EF (Entity Framework) POCO (Plain Old CLR Object) elemeken alapuló EDM-et (Entity Data Model) hoz létre különböző adattípusok get/set tulajdonságaival. Ezt a modellt használjuk, amikor elmentjük vagy lekérjük az egységek adatait a mögöttes adatbázisba.

Lekérdezés: Lehetővé teszi, hogy LINQ lekérdezéseket használjunk (C#/VB.NET) az adatok kinyerésére a mögöttes adatbázisból. Az adatbázis-szolgáltató lefordítja ezt a lekérdezést az adatspecifikus lekérdezési nyelvre (pl. SQL egy relációs adatbázis esetében).

Változáskövetés: Nyomon követi az entitások példányaiban bekövetkezett változásokat (tulajdonságértékek), amiket el kell küldeni az adatbázisba.

Mentés: A SaveChanges() metódus meghívásakor az EF INSERT, UPDATE és DELETE parancsokat hajt végre az adatbázisban az egységeken bekövetkezett változások alapján. Biztosítja az aszinkron SaveChangesAsync() metódust is.

Tranzakciók: Az EF automatikusan tranzakciókezelést hajt végre az adatok lekérdezése vagy mentése során. Lehetőséget biztosít a tranzakciók kezelésének testre szabására is.

Tárolás: Tartalmazza az első szintű gyorsítótárazást. Így az ismételt lekérdezések az adatokat a gyorsítótárból fogják visszaküldeni az adatbázis helyett.

2.8 .NET keretrendszer:

A .NET platform részét képező keretrendszer a Microsoft által létrehozott nagy kiterjedésű környezet, amit elsősorban a vállalati szintű alkalmazások, programok Windows alapú rendszereken történő futtatására terveztek. Programozási modellt, nagy osztály könyvtárat és eszközöket biztosít alkalmazások építéséhez, telepítéséhez és futtatásához olyan platformokon, mint a Linux, a macOS, a Windows, az iOS és az Android. Továbbá ezek megvalósításához egy nélkülözhetetlen szoftverfejlesztési platform.

Több programozási nyelvet támogat, illetve lehetővé teszi a fejlesztők számára, hogy olyan kódot írjanak, amely többféle hardveres architektúrákon és eszközökön például asztali számítógépeken, szervereken és mobilkészülékeken is futtatható. A keretrendszer számos osztály könyvtárat biztosít. Ezek megkönnyítik a programozók számára, hogy az általuk előnyben részesített nyelven írjanak kódot.

2002-ben a Microsoft kiadta az első verziót és azóta számos frissítésen és fejlesztésen esett át, a legutóbbi verzió a .NET 6, ami 2021-ben jelent meg. Népszerűsége annak köszönhető, hogy képes egyszerűsíteni a fejlesztési folyamatokat, növelni a termelékenységét, valamint megbízható és biztonságos környezetet nyújt az alkalmazások futtatásához.



10.kép: .NET logo⁴

⁴ .NET logo: <https://logos-world.net/net-framework-logo/> , 2023.04.18

Felhasználásának területei:

Alkalmazások készítése: Sokféle alkalmazás (például: asztali, webes, mobil alkalmazások, játékok) készítésére használható. Emellett számos technológiát és szabványt támogat (például az ASP.NET-et a webes fejlesztéshez, a WPF-et asztali szoftverek készítéséhez, továbbá az Entity Framework-öt az adateléréshez).

Fejlesztési folyamat felgyorsítása: Olyan eszközök és erőforrások átfogó tárházát nyújtja, amivel a fejlesztők megbízható, kiváló minőségű alkalmazásokat, szoftvereket készíthetnek Windows-ra.

A .NET keretrendszer által a futó programok számára elérhető szolgáltatások közé tartoznak a következők:

- **Nyelvi átjárhatóság:** Lehetővé teszi a programozók számára, hogy alkalmazásaik kódolásakor több kompatibilis programozási nyelv közül választhassanak. Ezen felül akár különböző nyelveken épített komponensek keverékéből is összeállíthatók.
- **Közös típusrendszer:** A változótipusok ugyanazok a .NET nyelvtől függetlenül.
- **Kiterjedt osztálykönyvtár:** Ahelyett, hogy jelentős mennyiségű kódot írnának a gyakori alacsony szintű programozási műveletek kezelésére, a programozók a .NET Framework Class Library típusok és tagjaik könnyen hozzáférhető könyvtárát használhatják.
- **Verzió kompatibilitás:** Néhány ritka kivételtől eltekintve a .NET Framework egy adott verziójának segítségével fejlesztett applikációk módosítás nélkül működhetnek a későbbi verziókon is.

2.9 Visual Studio 2022:

Szintén Microsoft által támogatott és fejlesztett program egy fejlesztői környezet legújabb kiadása, amit a világ minden táján élő fejlesztő használ. Segítségével megbízhatóan, kreatívan és gyorsan tudunk számítógépes alkalmazásokat és okostelefon applikációkat létrehozni. Emellett webes szolgáltatások és internetes oldalak fejlesztésébe is bátran belevághatunk.

A szoftver több programozási nyelvet tartalmaz (például: Java, C, C++, C#, Python). Jelen kiadásba jó néhány új fejlesztést építettek be, amik közül sokat fejlődött a sokak által nagyon kedvelt kódkiegészítő funkció is. Ez megkönnyíti a munkát azzal, hogy képes akár egész kódsorokat írni, ezáltal is támogatva a gyors és korrekt fejlesztést. Valamint a testreszabhatóság is kibővült, amely segítségével a felhasználóbarát kódolási környezet fejleszthetőbbé vált. Ezáltal is megkönnyíti a felhasználhatóságot.

3. FELÉPÍTÉS

A vizsgaremek szerkezetét tekintve 4 külön egységből áll, amelyek a következők:

1. Adatbázis
2. BackEnd (WEB API)
3. FrontEnd (React)
4. Asztali Alkalmazás (WPF)

A fejlesztés folyamatát a csapatlétszámhoz igazodva 3 részegységre bontottuk. Jakab Balázs Erik feladata volt a BackEnd fejlesztése, Puda Ádám a FrontEnd-ért felelt és Szabó Szabolcs készítette a WPF alkalmazást.

Az adatbázist együtt hoztuk létre, közös megbeszélések alapján. Emellett, ha bármelyikünk a fejlesztés során nagyobb akadályba ütközött, akkor a csapat többi tagja készségesen segítséget nyújtott számára. A munka során Google Drive-on keresztül tájékoztattuk egymást az előrehaladásokról, illetve mindenki tömörítve feltöltötte egy közös Drive meghajtóra a saját fejlesztéseit.

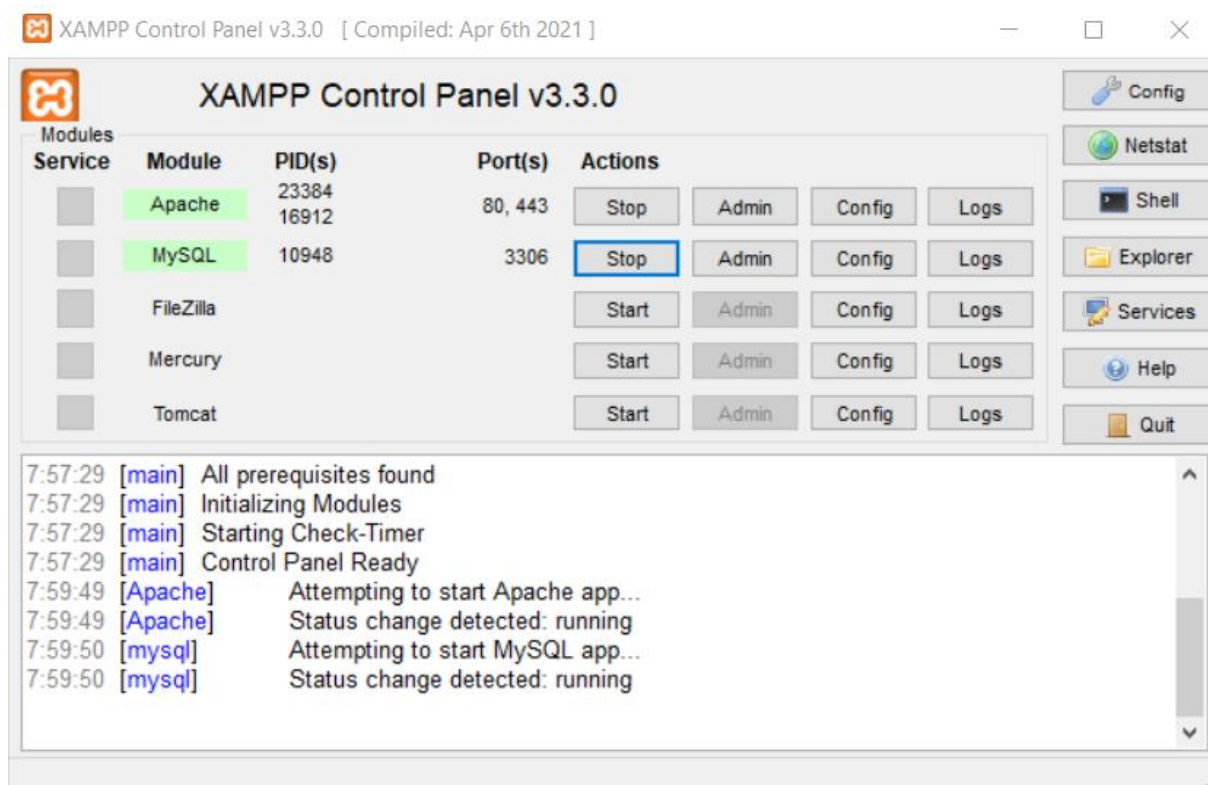
A következőkben részletesen bemutatjuk mind a négy modul felépítését és működését.

4. FELHASZNÁLÓI DOKUMENTÁCIÓ

4.1 Adatbázis

A fejlesztési folyamat alatt és a végleges verzióban is a **XAMPP** lokális hálózaton futó **phpMyAdmin** adatbázisát használtuk.

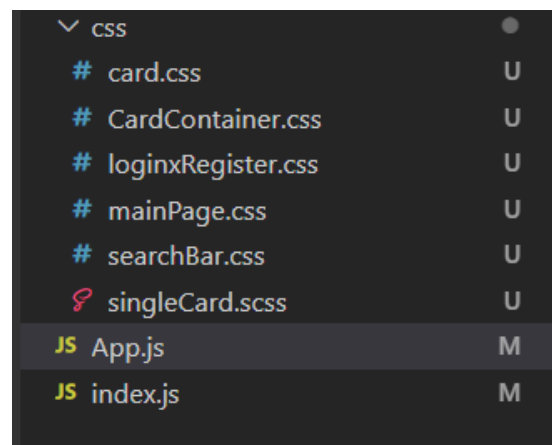
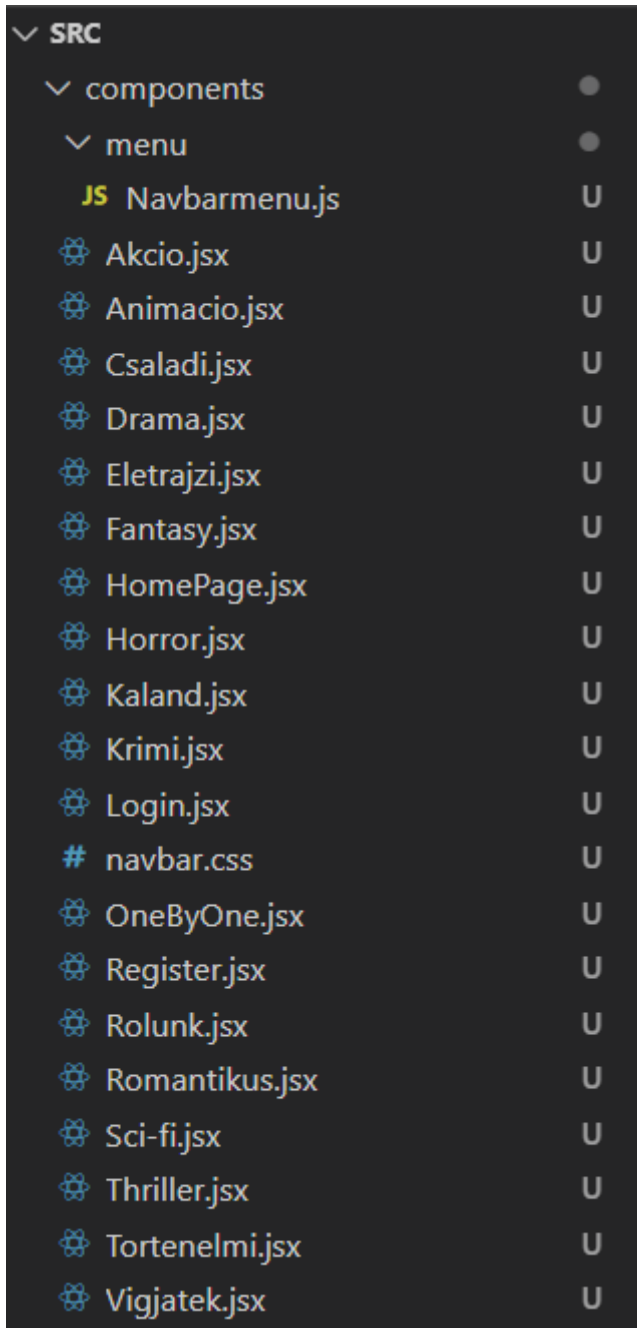
A **XAMPP** alkalmazás indítása után az **Apache** és **MySQL** modulokat indítva válik elérhetővé a lokális adatbázis, ezután lehet használni mind a **WPF**, mind a hozzá készített **React** webfelületet.



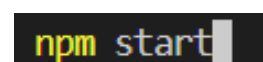
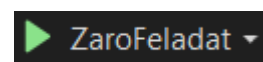
11.kép: adatbázis elindítása

4.2 FrontEnd:

A webes felület a **React 18.2.0** verziójában készült, minden hozzáadott **npm csomagnak** a lehető legfrissebb verzióját használja. Felépítését tekintve a következő mappákból, fájlokból áll:



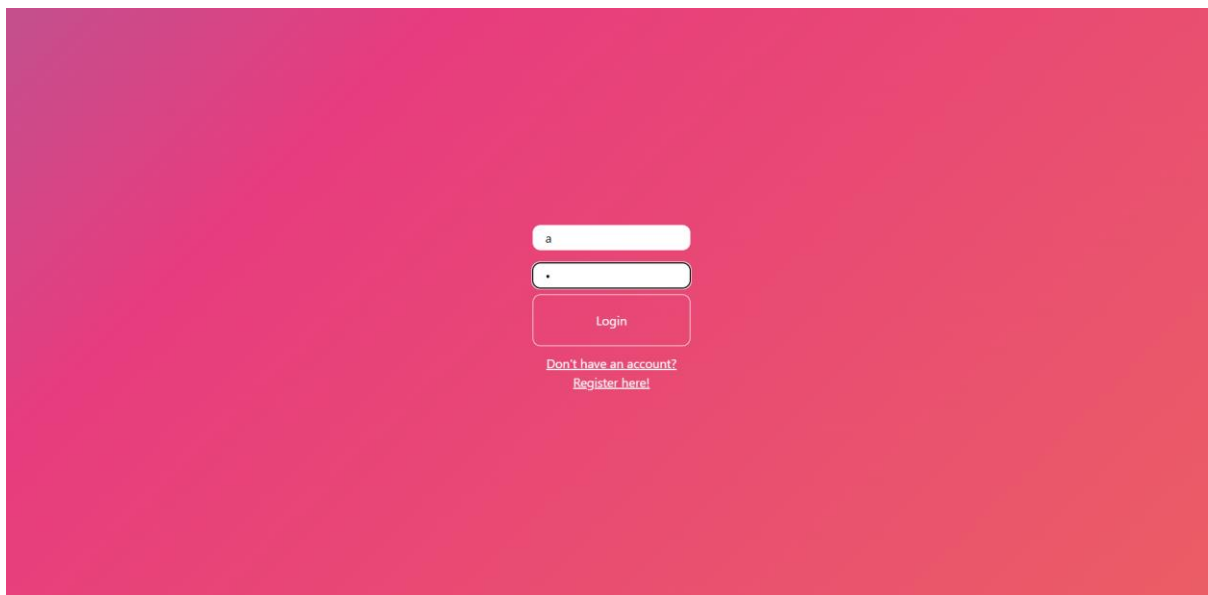
12.képek: React projekt tartalma



A felület megfelelő működéséhez futnia kell a backendnek is. A backend indítása **Visual Studio** programon keresztül történik az **F5** gomb megnyomásával, vagy a „**Zarofeladat**” gombra való kattintással. A webes alkalmazás indítása a **Visual Studio Code** terminal ablakán keresztül az „**npm start**” paranccsal történik.

4.2.1 Bejelentkezési felület:

Az alkalmazás indítását követően a bejelentkezési felületen találhatjuk magunkat. Bejelentkezés nélkül az oldal semmilyen funkciója nem érhető el, így kötelező, hogy felhasználóval rendelkezünk. További funkciók eléréséhez adjuk meg adatainkat, majd kattintsunk a „**Login**” gombra, ahogyan az alábbi példán látszódik:



13.kép: webes bejelentkezési felület

Az oldalra való bejelentkezéshez a példaként létrehozott „a” felhasználóval és „a” jelszóval léptem be. Ha nem rendelkezünk még felhasználói fiókkal, akkor kattintsunk a „**Register here!**” kiírásra és járjunk el a következő pontban leírtak szerint.

4.2.2 Regisztrációs felület:

Az előző pontban említett kiírásra kattintva megjelenik a regisztrációs felület. A regisztrációhoz minden kért adatot meg kell adnunk. Bármely adat kihagyása esetén a weboldal azzal fog visszatérni, hogy minden mezőt kötelezően ki kell töltenünk. Sikeres regisztrációhoz töltsük ki megfelelően a mezőket és kattintsunk a „**Register**” gombra. Ha mindent jól csináltunk, akkor a program visszatér egy „**Sikeres regisztráció!**” értesítéssel.

14.kép: webes regisztrációs felület

15.kép: sikeres webes regisztráció

Regisztrációs folyamatunk ezzel még nem ért véget, ugyanis az oldalon e-mail címes megerősítés működik. A sikeres regisztráció után nyissuk meg levelezési fiókunkat és kattintsunk a beérkezett levélben található linkre. Ezt követően regisztrációnk teljessé válik, bejelentkezhethetünk az oldalra.



backend.registry@kkszki.hu
címezett: én ▼

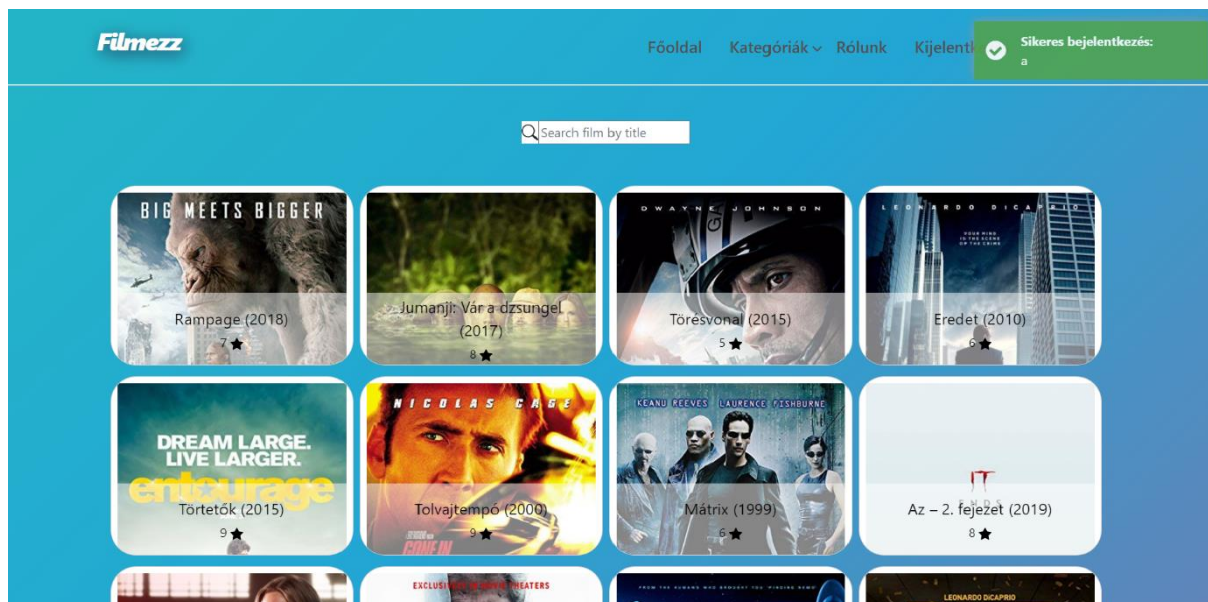
9:23 (12 perccel ezelőtt)

A regisztrációhoz kattints a következő linkre: <https://localhost:5001/Registry/ex049pDrDe0lw8AKMyRtQKXz4uciGfUEiutDdRJRCPcW4mSvxEUgiMNndFE4UJ7>

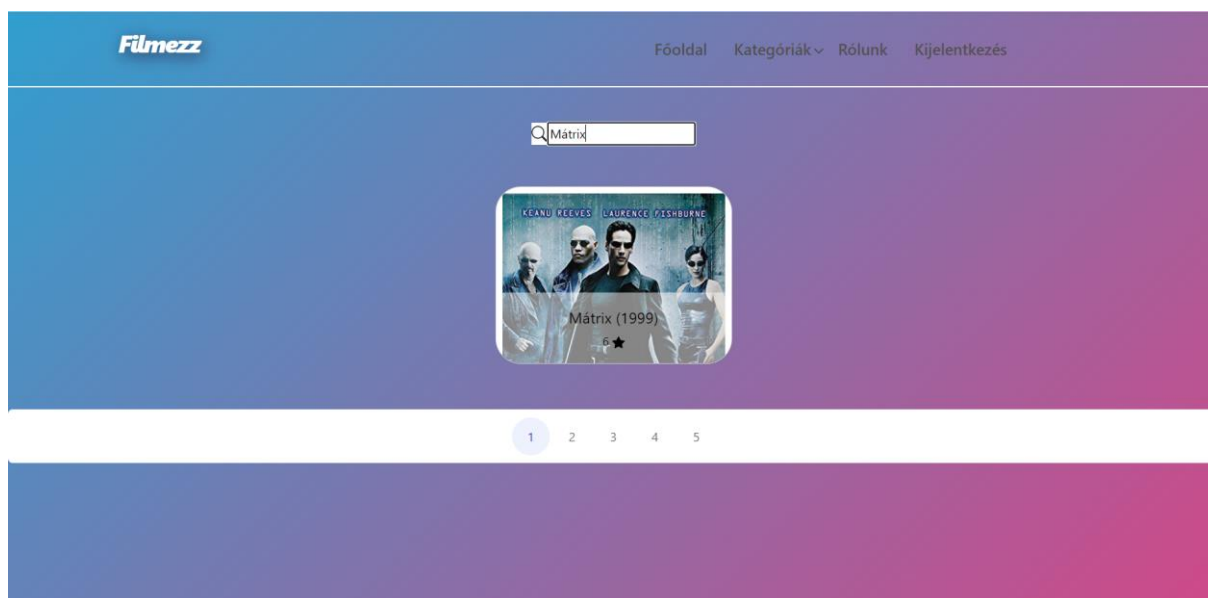
16.kép: megerősítő link

4.2.3 Kezdőlap és kategória szerinti rendezés:

A sikeres bejelentkezést követően a kezdőlapra találjuk magunkat. Ezen az oldalon pillanatok alatt betöltődnek az adatbázisból a filmek és azoknak adatai. Az oldal tetején található keresővel, akár olyan filmről is gyűjthetünk információt, melyet ismerünk.



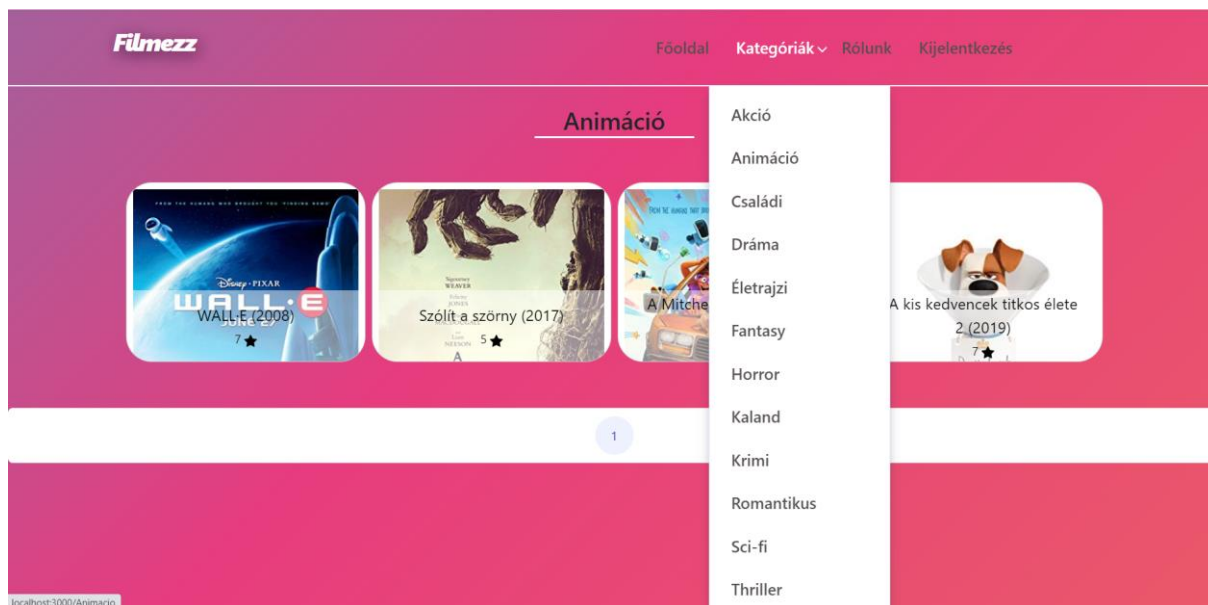
17.kép: kezdőlap



18.kép: kereső használata

A projektet úgy fejlesztettük, hogy az admin funkciókat a később említésre kerülő **WPF** alkalmazás látja el, így a webes felületen bármilyen jogosultsággal rendelkező felhasználó bejelentkezése esetén elérhetővé válik az oldal összes funkciója. A többi funkciót az oldal tetején található menüpontokra kattintva érhetjük el.

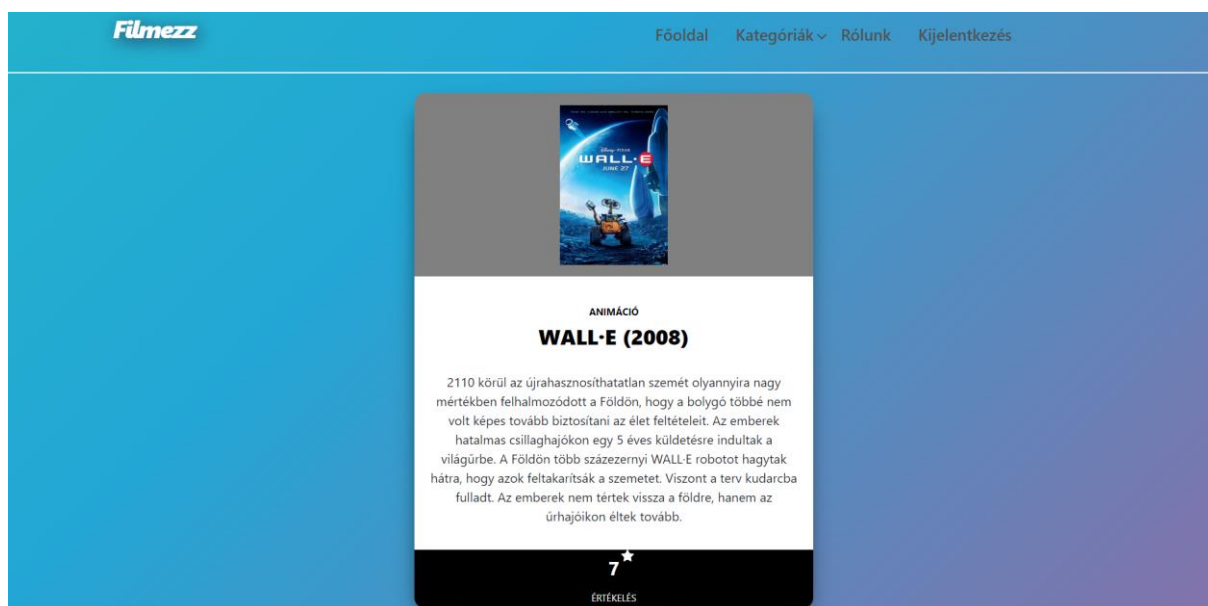
A „Kategóriák” menüpontot lenyitva kategóriák szerint listázhatjuk a filmeket. Ez a funkció megkönnyíti, hogy ízlésünk szerint böngészhessünk a tartalmak között.



19.kép: kategória szerinti rendezés

4.2.4 További információ egy bizonyos filmről:

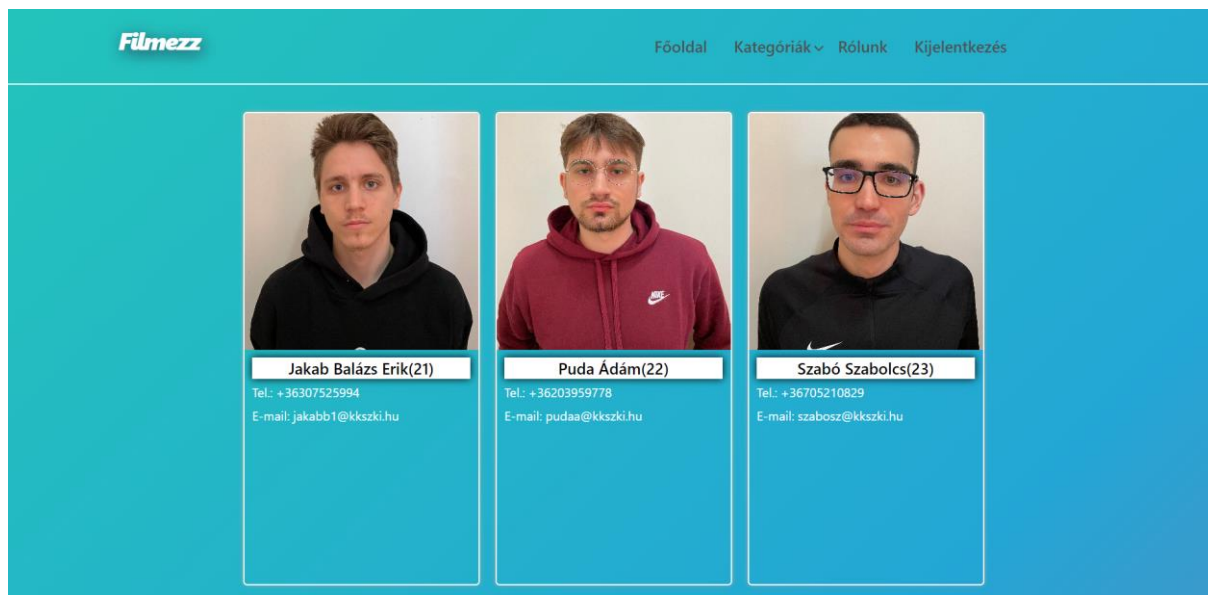
Miután kiválasztottuk a kedvelt filmet többet is megtudhatunk róla. Ha a film képére/címére kattintunk egy új oldalon találjuk magunkat, ahol megjelenik a film minden adata. Információkat szerezhetünk a kiválasztott film kategóriájáról, megjelenési évéről, a nézői értékeléséről, illetve egy rövid leírást is elolvashatunk magáról a film történetéről.



20.kép: bővebb információ egy filmről

4.2.5 Rólunk menüpont:

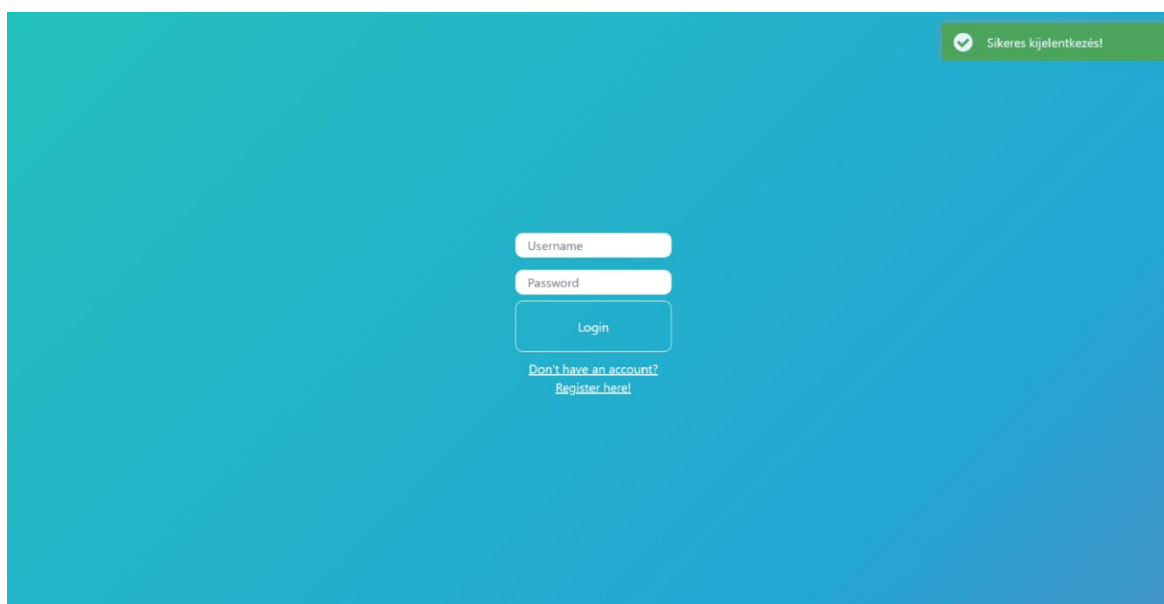
A „Rólunk” menüpontot azért hoztuk létre, hogy egy elérhetőségi felületet kínáljunk azoknak, akik fel szeretnék venni velünk a kapcsolatot. Ezen az oldalon megtalálható az oldal készítőiről egy kép, a nevük, a koruk, illetve az elérhetőségeik.



21.kép: Rólunk oldal

4.2.6 Kijelentkezés:

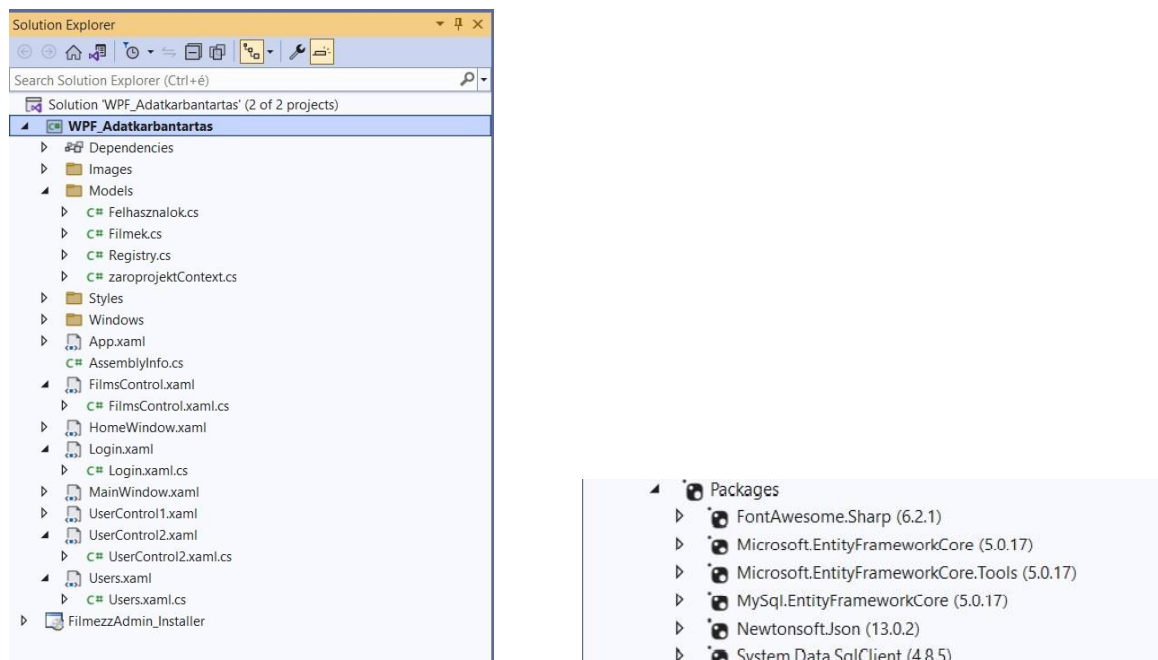
A kijelentkezés gombra kattintva elhagyjuk az oldal tartalmi részét, ismét a bejelentkező felületen találjuk magunkat. Sikeres kijelentkezés esetén jobb felső sarokban megjelenik egy üzenet, mely közli, hogy kijelentkezési próbálkozásunk sikeres volt.



22.kép: kijelentkezés

4.3 WPF asztali alkalmazás:

Az asztali applikáció (WPF Application) a BackEnd-hez hasonlóan .NET 5-tel készült azért, hogy a fejlesztés során keletkező esetleges kompatibilitási hibákat megelőzzük. Az állományok felépítését tekintve a következő mappákból és fájlokból áll:



23.kép: WPF könyvtár

A képen látható, hogy milyen összetevőkből áll egy egészzé a szoftver. A Solution az elsődleges állomány és ezekhez tartoznak további mappák és fájlok. A Dependencies-ben találhatóak a Packages (Csomagok), amelyek nagy részét egyesével kell hozzáadni a Projekthez. Újdonságnak számunkra a FontAwesome.Sharp csomag számít. Ez a Package stílusokat tartalmaz kifejezetten a WPF alkalmazások számára (például ikonok, betűstílusok, kinézetek, stb.), amely a későbbiekben megnyilvánul a bejelentkező illetve a fő felületeken. Az Images mappában találhatóak a Projekthez adott képek.

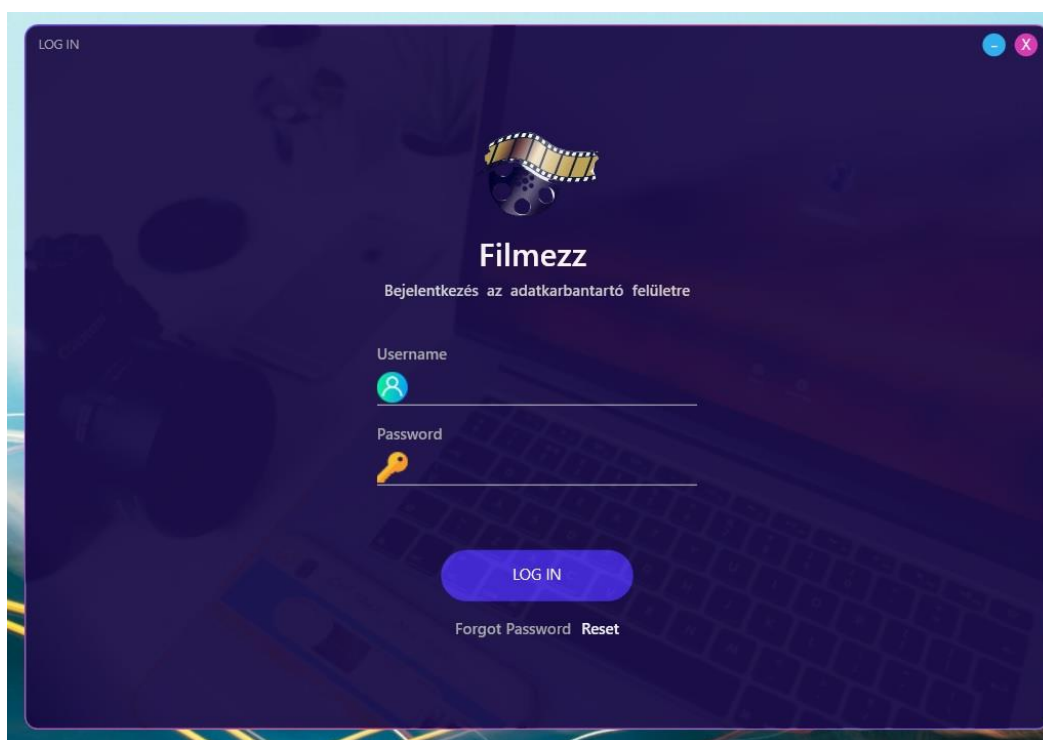
A Models-ben osztályok találhatóak melyek az adatbázisból lettek meghívva a Scaffold-DbContext tanult paranccsal. Továbbá a Windows-ban lelhetőek a későbbiekben megjelenő oldalak kódjai és eljárásai. Emellett a FilmezzAdmin_Installer-ben van egy telepítő készlet. Ennek segítségével feltelepíthető az applikáció asztali számítógépekre. Erről is lesz szó egy kicsit később.

Ugyanakkor a felületben fellelhető menüpontokat UserControl típusú ablakokkal vannak megvalósítva. Így a képen is látható HomeWindow.xaml-be lettek betöltve a UserControl fajtájú ablakok.

4.3.1 Bejelentkező felület:

Amikor elindítjuk az asztali alkalmazást a bejelentkező felület jelenik meg. Itt a korábban már regisztrált felhasználók léphetnek be felhasználónév és jelszó párossal. Fontos megjegyezni, hogy ez egy Admin felület ez által csak a magasabb jogosultságú felhasználók léphetnek be. Emellett plusz fűcsör (tulajdonság), hogy korlátozva van a belépési kísérletek száma, melyet 3-ban állapítottunk meg. A 3. sikertelen próbálkozást követően a felület bezáródik. Továbbá kidob egy figyelmeztető ablakot, hogy elfogytak a belépési lehetőségek.

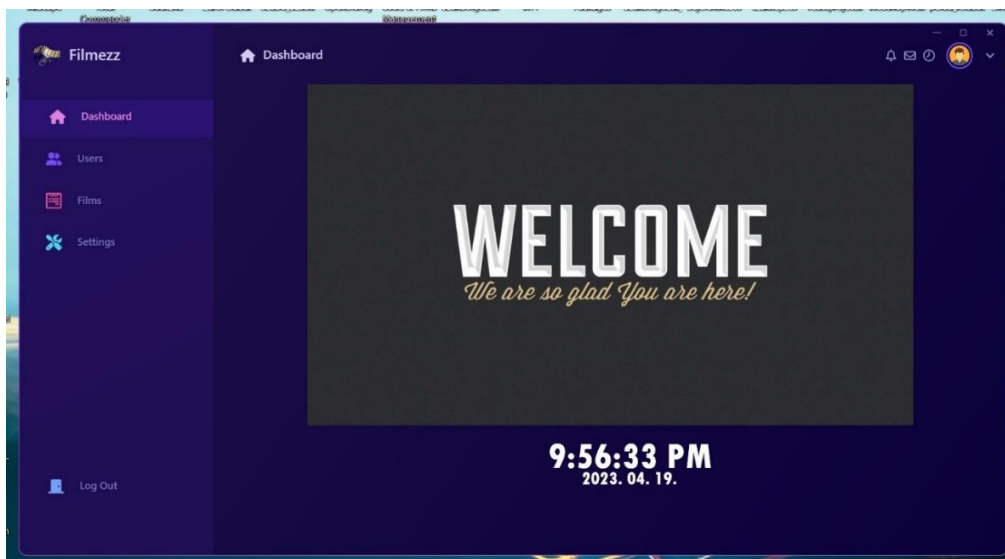
Fejlesztés közben megpróbáltunk egy minőségi dizájnt és stílust összeállítani a korábban említett Csomag segítségével. Emellett figyelmet fordítottunk arra is, hogy a FrontEnd-es felület passzoljon a WPF kinézetéhez.



24.kép: Bejelentkező felület

4.3.2 Dashboard Page:

Sikeres bejelentkezést követően a „Dashboard” az első felület, ami fogadja a felhasználókat. Itt egy kedves üdvözlő háttér és egy digitális óra lett elhelyezve dátumjelzéssel. Véleményünk szerint minden komolyabb Admin felület, ami „ad magára” ilyen menüponttal kezdődik.

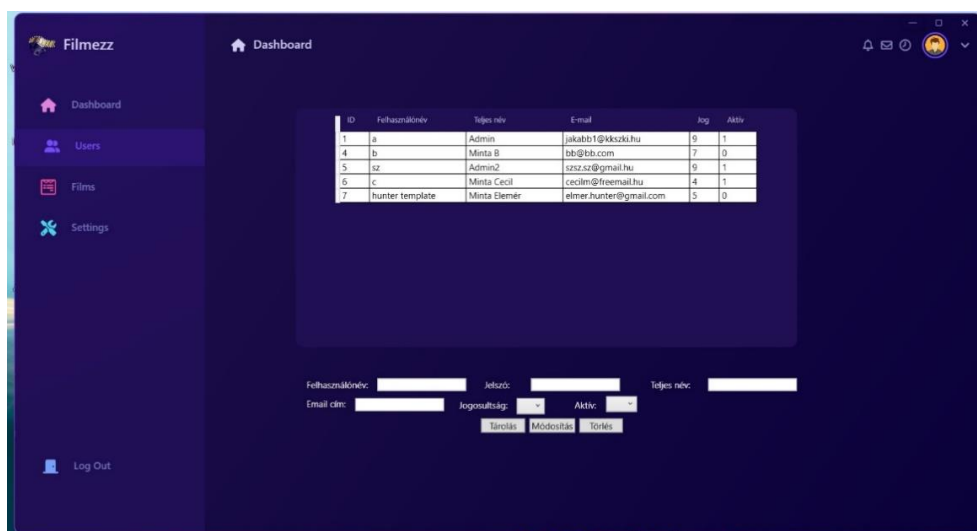


25.kép: Dashboard menüpont

Az egész ablakra különböző műveletek lettek írva ezzel megkönnyítve a felhasználói használatot. Ilyen például a teljes képernyős mód, amit ha a jobb felső sarokban kiválasztunk, akkor teljes képernyős lesz a felület. Emellett lehetőség van arra is, hogy a tálcára tegyünk a felületet, ha egy kis időre nincsen rá szükség.

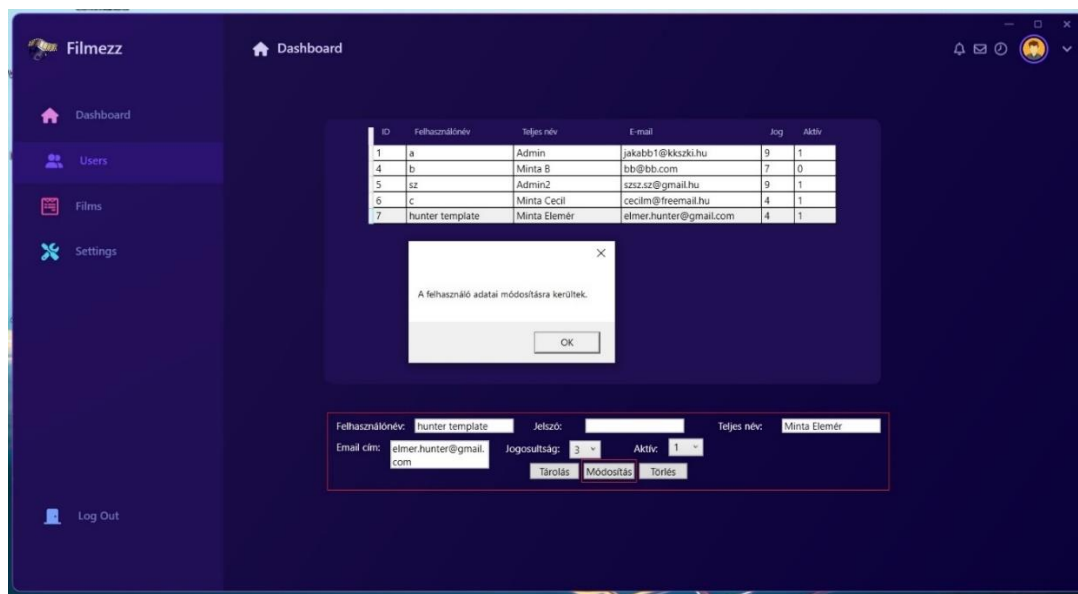
4.3.3 Users menüpont:

A következő menüpont a Users lehetőség. Ebben található az összes regisztrált felhasználó adatai. Itt lehet új embert hozzáadni, módosítani és törölni. Hozzáadás esetén meg kell adni a Felhasználónevet, jelszót, teljes nevet, email címet, jogosultságot és az aktív tulajdonságot. Ennél a résznél fontos megemlíteni, hogy a beírt jelszó sha256-tal lesz eltitkosítva, amelynek visszafejtése körülményes egy esetleges jogosulatlan behatolás esetén.



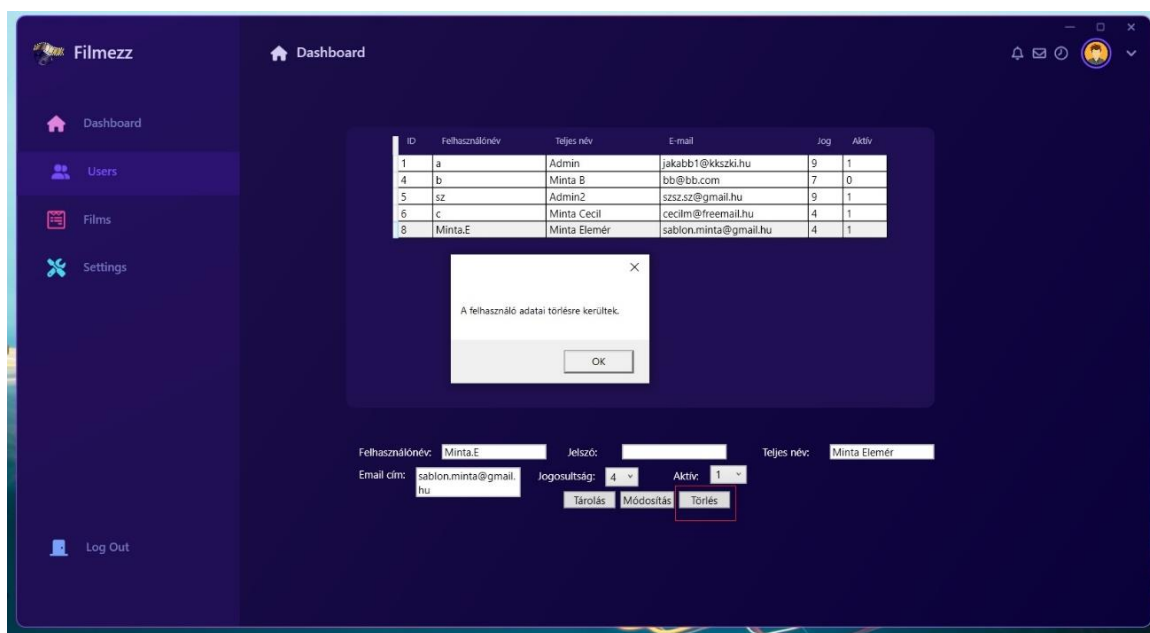
26.kép: Users

Módosítás esetén csak elég az adott ember sorára kattintani ez által az alsó textbox-ba beíródnak a jelenlegi adatok. Ezeket így már csak át kell írni és a „Módosítás” gombra kattintani.



27.kép: felhasználó módosítása

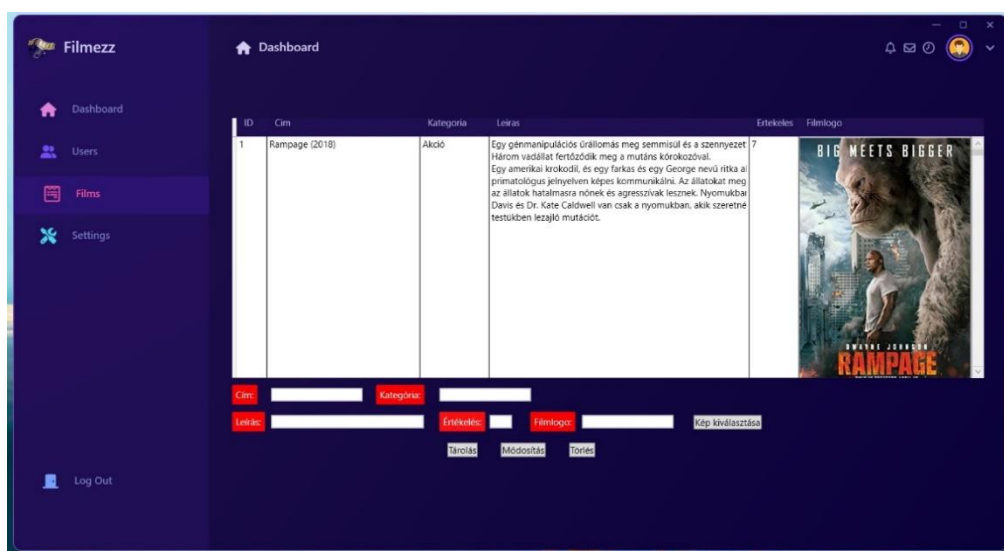
Törlésnél is hasonló a működés azzal ellentétben, hogy itt az ID alapján a „Törlés” gombra kattintást követően felugrik egy figyelmeztető ablak, ami rákérdez a felhasználótól arra, hogy biztosan szeretné-e törölni a kiválasztott ember adatait. Amennyiben az „igen”-re kattintunk, akkor a törlés végbemegy és sikeresen töröltük az adatokat. Viszont, ha a „nem” -et választjuk az ablak eltűnik és visszakerülünk a Users menü pontra.



28.kép: felhasználó törlése

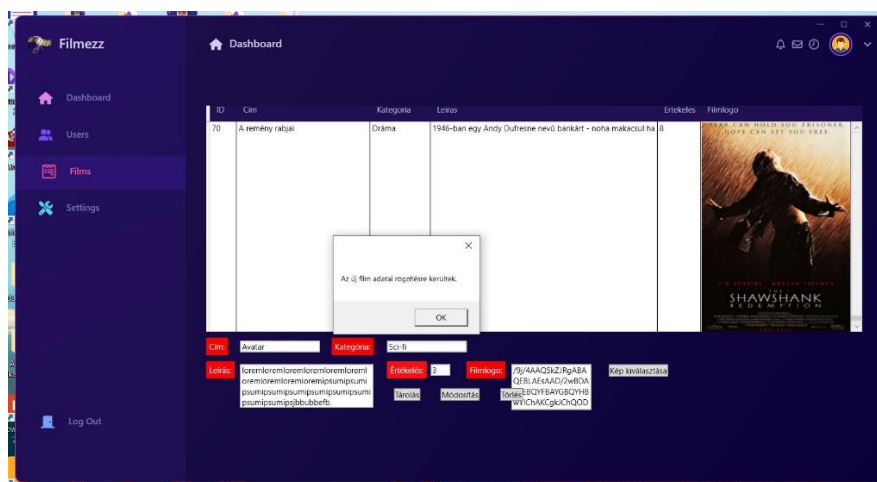
4.3.4 Films opció:

Ebben a menüpontban van elhelyezve a filmek karbantartása. A táblázatban helyet foglal számos adat hasonlóan az előzőekhez. Működésileg azonos a Users menüponttal. Itt is találhatóak textbox-ok de viszont ez ki lett bővítve egy képhozzáadás gombbal. Jelentősége azért fontos, mert a hozzáadás esetén meg kellett oldanunk, hogy a képeket hozzá lehessen adni illetve megjeleníteni a táblázatban (DataGrid-ben).



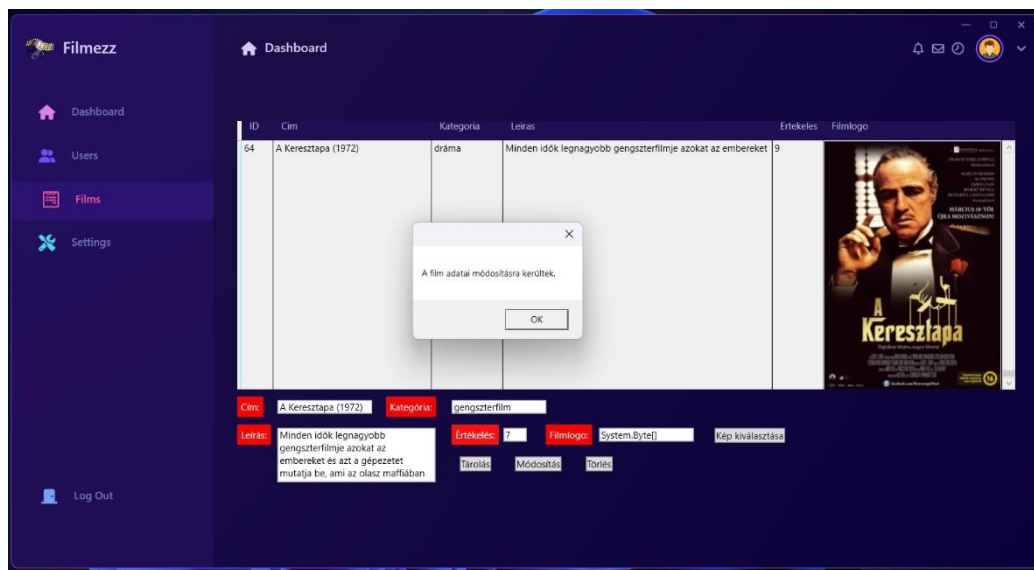
29.kép: Films

Hozzáadás estében is hasonlóképpen működik az eljárás a korábbiakhoz. A textbox-okat egyesével kitöltjük, figyelve arra, hogy minden ki legyen töltve. Például, ha nem választunk ki képet a számítógépről és ebben a formában küldjük el a program számára az adatokat hiába megjelenik a „Sikeres hozzáadás” felirat mégsem lesz jó és hibára fut az alkalmazás a leközelebbi betöltéskor. Ennek eredményeképp a felület hirtelen bezáródhat.

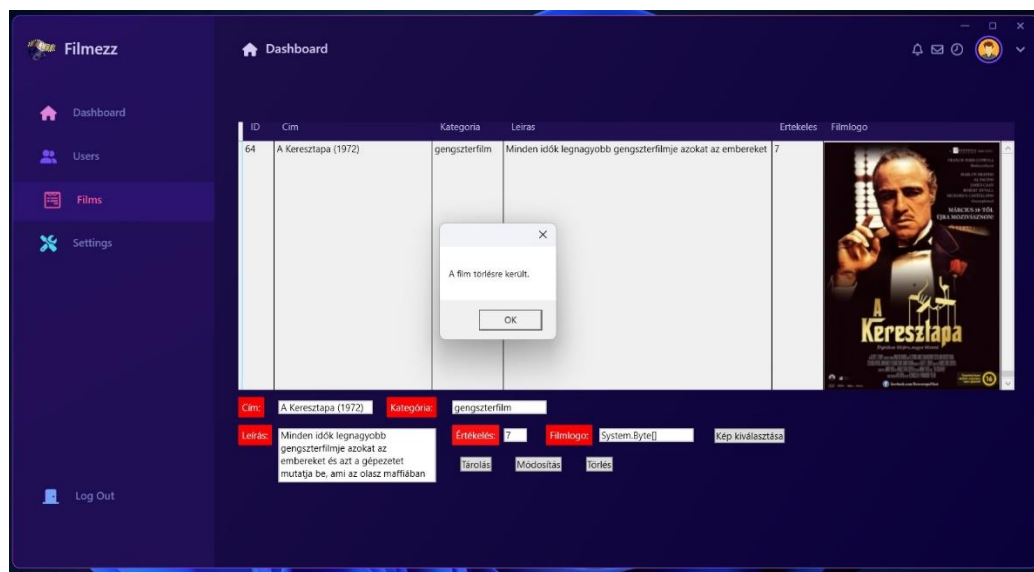


30.kép: film hozzáadás

A módosítási és törlési folyamat is hasonlóan működik az eddigiekben leírtak alapján. Ugyanúgy, ha a kurzorral rákattintunk egy sorra akkor az beíródik a textbox-okba és ott átlehet írni az adatokat. Ezen felül a törlés estében a filmeknél szintén ID alapján lett megoldva a törlés.



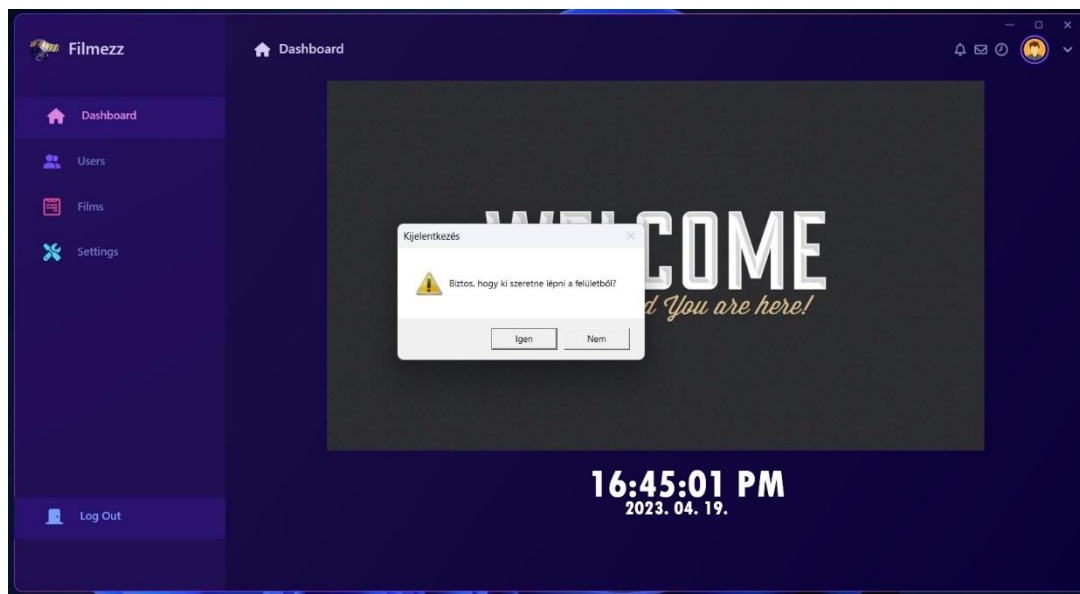
31.kép: film módosítás



32.kép: film törlése

4.4.4 Kijelentkezés:

A menüpontok legalsó részén található „Log Out” felirattal a kijelentkezés. Amennyiben erre kattintunk megjelenik egy új ablak, ami a felhasználótól megkérdezi a kilépési szándékot. Az „Igen” -re kattintás esetében a felület sikeresen kijelentkezteti a felhasználót és elköszön tőle. A másik lehetőség pedig az, hogy a felhasználó visszatérhet az alkalmazásba.

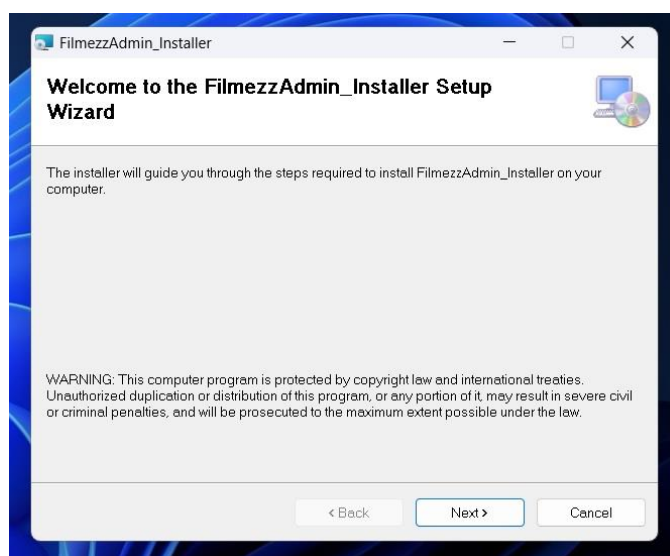


33.kép: kijelentkezés

Továbbá a jobb felső sarokban lévő X-szel is ugyanez az ablak ugrik fel, amellyel szintén kiléphetünk és bezárhatjuk az alkalmazást.

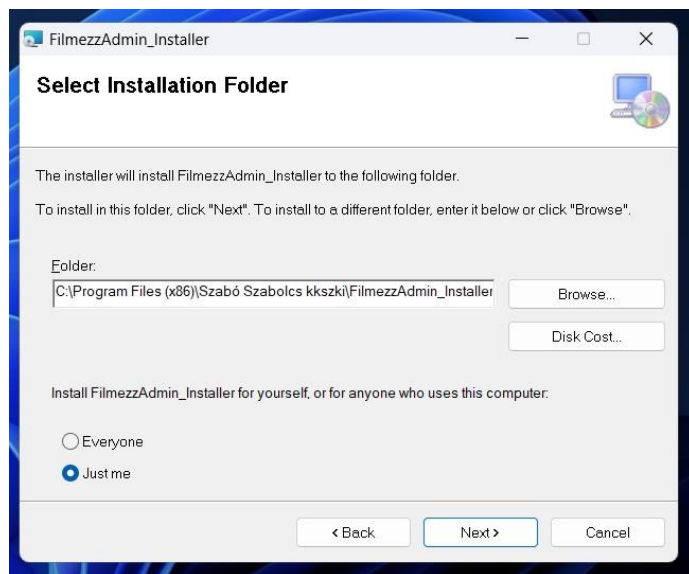
4.4.5 A telepítés menete:

A felhasználói csomaghoz tartozik egy telepítési készlet is. Első lépésként nyissuk meg a FilmezzAdmin_Installer mappát. Ezt követően lépünk be a Debug mappába. Ebben két állomány található. Egy setup és egy msi kiterjesztéssel. Nyissuk meg az egyiket és a következő ablak jelenik meg:



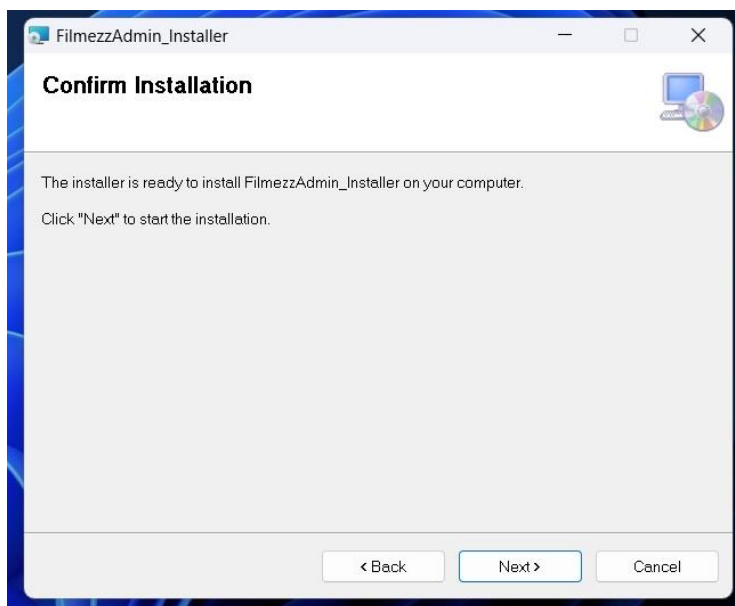
34.kép: telepítés 1.lépés

Ez beállítja az alkalmazást a Windows rendszerű számítógépünkre. Ezután kattintsunk a „Next” gombra. A kattintás után:



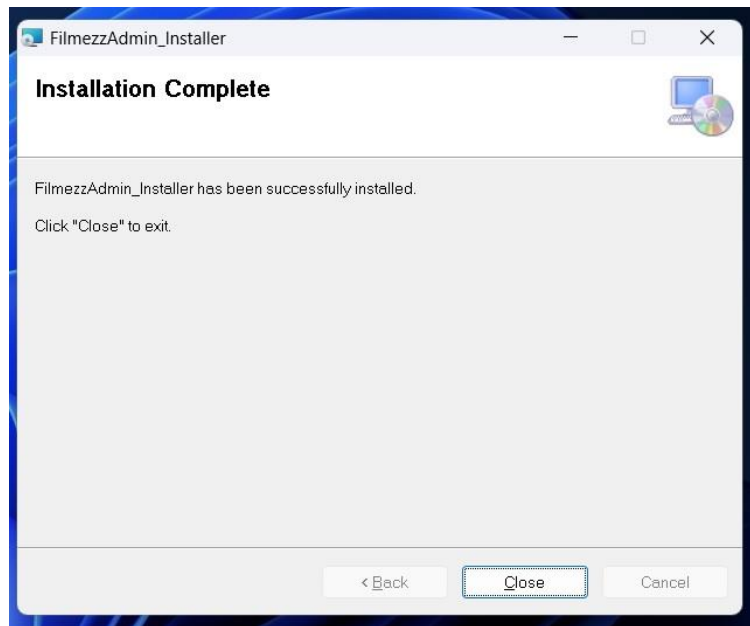
35.kép: telepítés 2.lépés

Itt kiválasztjuk azt a meghajtót és mappát ahová szeretnénk telepíteni a programot. Továbbá azt is, hogy (több felhasználó esetén) mások is elérjék az alkalmazást. Kattintsunk a „Next” gombra. Gomblenyomás után:



36.kép: telepítés 3.lépés

Itt elfogadjuk a telepítést a „Next” Gombbal. Ezt követően a telepítési varázsló végigfutott. A következő ablakot be is zárhatjuk a „Close” gombbal. Emellett keletkezett az asztalon egy ikon is, amivel megnyithatóvá vált programunk.



37.kép: 4. lépés

5. Továbbfejlesztési opciók

Csapatunk számára nem maradt több idő, illetve a mostani tudásunkkal még nem tudjuk az alábbi ficsöröket (funkciókat, képességeket) kivitelezni. Ezeket a későbbiekben szívesen beleépítenénk a vizsgaremekbe:

FrontEnd (React):

- Ha megnyitunk egy filmet akkor az megjelenik egy külön oldalon a leírással. Ezt fejlesztenénk tovább, hogy a filmhez tartozó előzetes is helyet kapna ezen az oldalon.
- Megvalósítanánk azt a funkciót, amivel a felhasználók tudják értékelni a filmeket és egymás között egy üzenőfalon megbeszélhetnék a film által nyújtott pozitív, illetve negatív tulajdonságokat és a cselekményt.
- Az emberek számára elérhetővé tettük volna azt a lehetőséget, hogy az adott filmet kikölcsönözhessek az oldalon és megtekinthessék bármilyen eszközön.

(WPF) asztali alkalmazás:

- A Settings menüpont befejezése azzal a móddal, hogy a kinézetet lehessen világos és sötét témák között kattintással változtatni.
Továbbá a már megjelenített textbox-okba a bejelentkezett felhasználó adatait betölteni és ezek módosításának a lehetőségét biztosítani a felhasználók számára.
- A login felület alján található „Reset Password” opció kivitelezése.
- A főablakokban megjelenő dropdown menü, illetve a kis avatar ikon mellett elhelyezett további ikonok jelentésének a megvalósítása.

BackEnd (WEB API)

- A FrontEnd-hez és a WPF-hez készíteni további kéréseket, metódusokat és függvényeket, amikkel teljesíthetőek lennének a korábban leírt fejlesztések.

6.Irodalomjegyzék

1. A phpMyAdmin bemutatása, <https://www.javatpoint.com/phpmyadmin> , 2023.04.14.
2. A React, https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started , 2023.04.17.
3. Axios, <https://www.freecodecamp.org/news/how-to-use-axios-with-react/#what-is-axios> , 2023.04.17.
4. WPF-ről, <https://wpf-tutorial.com/hu/1/a-wpf-rol/mi-az-a-wpf/> , 2023.04.17.
5. XAML a WPF-ben, <https://www.c-sharpcorner.com/blogs/what-is-xaml122> , 2023.04.18.
6. Mi a WEB API? , <https://www.tutorialsteacher.com/webapi/what-is-web-api> , 2023.04.18.
7. Entity Framework, <https://www.entityframeworktutorial.net/what-is-entityframework.aspx> , 2023.04.18.
8. .NET Framework, <https://codeinstitute.net/global/blog/what-is-net-framework/> , 2023.04.18.