**ŽILINSKÁ UNIVERZITA V ŽILINE**  
 FAKULTA RIADENIE A INFORMATIKY



**Semestrálna práca**Predmet: Pokročilé objektové technológieŠtudijný program: Informačné systémy

Vypracoval: Bc. Zimen Dávid Školský rok: 2024/2025 Študijná skupina: 5ZIS12

**OBSAH**

[Úvod 3](#_Toc188172557)

[1 UML diagramy 3](#_Toc188172558)

[2 Použité technológie 3](#_Toc188172559)

[2.1 Databáza: PostgreSQL 3](#_Toc188172560)

[2.2 Backend: ASP.NET Web API 3](#_Toc188172561)

[2.3 Autentifikácia: Keycloak 3](#_Toc188172562)

[2.4 Frontend: Angular 4](#_Toc188172563)

[2.5 Kontejnery: Docker a Docker Compose 4](#_Toc188172564)

[2.6 ORM: Entity Framework Core 4](#_Toc188172565)

[3 Požiadavky na spustenie 4](#_Toc188172566)

[3.1 Pomocou Dockeru 4](#_Toc188172567)

[3.2 Lokálna inštalácia 4](#_Toc188172568)

[Záver 5](#_Toc188172569)

# Úvod

Cieľom tejto semestrálnej práce je navrhnúť a implementovať webový portál na zábavu, ktorý slúži na správu a prezentáciu informácií o filmoch, seriáloch, hercoch, režiséroch, žánroch a postavách.

Inšpiráciu som čerpal z populárnej platformy IMDb, ktorá poskytuje používateľom možnosť prehľadávať bohatú databázu audiovizuálnych diel a ich tvorcov.

Mojou prioritou bolo nielen vytvoriť funkčný a použiteľný systém, ale aj naučiť sa nové technológie, ktoré som doteraz nepoužíval. Projekt je postavený na moderných technológiách a postupoch, pričom hlavným cieľom bolo využiť pokročilé funkcie ASP.NET na backend, Keycloak na autentifikáciu a Angular na frontend. Celý systém je integrovaný do prostredia Docker, aby bolo jeho nasadenie a používanie jednoduché a konzistentné.

Projekt sa zameriava na vytvorenie robustného základu, ktorý je pripravený na budúci rozvoj a rozšírenie o ďalšie funkcionality, ako napríklad profesionálne recenzie od kritikov, systém odporúčaní, alebo možnosť vytvárať osobné zoznamy na sledovanie. Vývoj tejto aplikácie mi umožnil získať nové skúsenosti s technológiami a architektúrami, ktoré som doposiaľ nepoužíval.

# UML diagramy

# Použité technológie

V semestrálnej práci boli použité nasledovné technológie.

## PostgreSQL

PostgreSQL je objektovo-relačný databázový systém, ktorý ponúka vysoký výkon a flexibilitu. V projekte som ho použil na ukladanie informácií o filmoch, seriáloch, hercoch, režiséroch, postavách a žánroch. PostgreSQL som si vybral aj pre to, že s ním mám už bohaté skúsenosti a tak jeho integrácia do projektu nerobila žiadne problémy.

## ASP.NET Web API

ASP.NET Web API je framework vyvinutý spoločnosťou Microsoft, ktorý umožňuje vývoj REST API služieb. Pomocou ASP.NET som implementoval serverovú logiku aplikácie, ktorá spracováva požiadavky od používateľov, komunikuje s databázou a poskytuje dáta pre frontend. Jeho výhodou je jednoduchá integrácia s databázovými frameworkami, ako je Entity Framework Core a široká komunita, čo znamená, že pri akýchkoľvek problémoch človek dokáže na internete jednoducho nájsť riešenie.

## Keycloak

Keycloak je nástroj na správu identít a prístupov, ktorý poskytuje funkcie autentifikácie, autorizácie a správy používateľov. V tomto projekte som Keycloak použil na zabezpečenie prístupu k API službám a na správu používateľských práv. Keycloak podporuje štandardy OAuth 2.0 a OpenID Connect, čo zabezpečuje bezpečný a flexibilný spôsob autentifikácie. Keďže Keycloak je vyvíjaný v jazyku Java, tak aj knižnica na komunikáciu s ním je na oveľa vyššej úrovní práve v danom jazyku. Z tohto dôvodu, bola integrácia určite výzvou, ale s pomocou knižnice Keycloak.Net.Core, sa to nakoniec podarilo.

## Angular

Angular je moderný framework pre vývoj dynamických webových aplikácií. V projekte som Angular využil na tvorbu používateľského rozhrania, ktoré umožňuje prehľadné zobrazovanie údajov o filmoch a seriáloch. Angular poskytuje výkonné nástroje na správu stavov aplikácie, komunikáciu s API a tvorbu komponentov, čo umožňuje škálovateľnosť a jednoduché pridávanie nových funkcionalít.

## Docker

Docker je platforma na kontajnerizáciu aplikácií, ktorá umožňuje zabaliť softvér do prenosných kontajnerov, obsahujúcich všetky potrebné závislosti. V projekte som použil Docker na kontajnerizáciu jednotlivých komponentov (PostgreSQL, Keycloak, ASP.NET a Angular), čo zjednodušuje proces nasadzovania a spúšťania aplikácie. Pomocou Docker Compose sa zorchestrovali jednotlivé služby a zabezpečilo sa ich jednoduché spustenie pomocou jedného príkazu.

## Entity Framework Core

Entity Framework Core je framework pre mapovanie objektovo-relačných dát, ktorý umožňuje vývojárom pracovať s databázami prostredníctvom objektovo-orientovaného modelu. V projekte som ho využil na jednoduché vytváranie a správu databázových entít, čo znižuje množstvo písaného kódu a minimalizuje chyby pri práci s databázou. EF Core tiež podporuje migrácie, čo uľahčuje zmeny v databázovej štruktúre počas vývoja.

# Požiadavky na spustenie

## Pomocou Dockeru

* Spustite **docker-control.bat**, ktorý zabezpečí automatické spustenie všetkých služieb pomocou Docker Compose.
* Po úspešnom buildnutí aplikácie bude celý systém dostupný na preddefinovanej adrese, vo webovom prehliadači na adrese <http://localhost:4200>.

## Lokálna inštalácia

Nainštalujte nasledujúce komponenty:

* PostgreSQL 17
* .NET SDK (verzia 8)
* Node.js (verzia 22)
* Keycloak (verzia 26)
* Angular CLI (verzia 19)

Upravte konfiguračný súbor **appSettings.json** podľa údajov počas vašej inštalácie. Spustite jednotlivé služby ručne pomocou príkazov ./**kc.bat**, **dotnet run** a **ng serve**.

# Pužívateľská príručka

# Vzniknuté ťažkosti a ich riešenia

## Práca s ASP.NET a C#

Ako začiatočník som sa stretol s problémami pri návrhu API a validácii požiadaviek. Framework bol pre mňa úplne nový a v niektorých ohľadoch dosť odlišný od frameworku Spring Boot, ktorý je môj hlavný nástroj, To malo za následok pomalší čas vývoja.

* *Riešenie*: Štúdium oficiálnej dokumentácie a využívanie online tutoriálov najmä od youtubera Milana Jovanoviča,.

## Integrácia Keycloak

Konfigurácia Keycloaku, tak aby bola zabezpečená jeho bezproblémová integrácia s ASP.NET bola náročná, najmä z pohľadu zabezpečenia tokenov a mapovania jednotlivých oprávnení.

* *Riešenie*: Použitie príkladov z komunity a ladenie prostredníctvom debugovania.

## Dockerizácia

Orchestrácia viacerých kontajnerov bola spočiatku problematická, najmä písanie Docker súborov a synchronizácia ich spúšťania.

* *Riešenie*: Testovanie rôznych nastavení v docker-compose.yml a ladenie závislostí medzi kontajnermi.

# Záver

Projekt poskytol základné funkcie pre portál zábavy, avšak niektoré plánované funkcionality, ako správa seriálov, neboli implementované. Napriek tomu som získal cenné skúsenosti s novými technológiami, ako ASP.NET a Keycloak. Tento systém je pripravený na ďalší vývoj, ktorý by mohol zahŕňať hodnotenia, odporúčania alebo možnosť vytvárania zoznamov na sledovanie.