

Movies web application design using laravel framework

Zoltan Kapas ^a

^a Máv Szolgáltató Központ Zrt., Könyves Kálmán körút 54-60, Budapest, 1087, Hungary,
zoltan.kapas25@gmail.com

Abstract

The central topic of this article is the design of a web application that offers users the opportunity to rate and like their favorite movies on a special online interface. In addition, there is an opportunity to exchange ideas with other users, as opinions about movies can be shared in a comments section. For the development of the web application, the widely recognized Laravel PHP framework was chosen, for which the Bootstrap CSS framework is used to display the user interface. Thanks to this, the website will be easy and efficient to use on all screen sizes and devices. The display of movie details will be supported by a REST API-based system on the site. In the first part of the article, the technologies and tools used for the development of the application are presented in detail. The role and benefits of using the Laravel framework and the Bootstrap CSS framework in this project will be highlighted. The second part focuses on the planning phase. In doing so, an overview is given of how they plan to implement the film evaluation and liking functions, as well as the possibility of comments among users. They also cover the design steps to be performed in terms of user experience and application functionality. This article, divided into two parts, provides a thorough insight into the process of designing a web application that provides an interactive platform for film lovers to share and live together their experiences related to their favorite films.

Keywords: programming; Laravel; REST API

Filmek webalkalmazás tervezése laravel keretrendszerrel

Kapas Zoltan ^a

^a Máv Szolgáltató Központ Zrt., Könyves Kálmán körút 54-60, Budapest, 1087, Hungary,
zoltan.kapas25@gmail.com

Absztrakt

A cikk központi témája egy olyan webes alkalmazás tervezése, mely lehetőséget kínál a felhasználóknak arra, hogy kedvenc filmjeiket értékelhessék és kedvelhessék egy speciális online felületen. Ezen túlmenően lehetőség nyílik más felhasználókkal való gondolatcserére is, mivel a filmekről alkotott vélemények megoszthatók egy hozzászólások szekciójában. A web alkalmazás fejlesztéséhez a széles körben elismert Laravel PHP keretrendszert választották, melyhez a felhasználói felület megjelenítéséhez a Bootstrap CSS keretrendszert használják fel. Ennek köszönhetően a weboldal könnyedén és hatékonyan használható lesz mindenféle kijelzőméreten és eszközön. A filmek részleteinek megjelenítését egy REST API-alapú rendszer fogja támogatni az oldalon. A cikk első részében részletesen bemutatásra kerülnek a technológiák és eszközök, melyeket az alkalmazás fejlesztéséhez alkalmaznak. Kiemelten szó lesz a Laravel keretrendszer és a Bootstrap CSS keretrendszer használatának szerepéről és előnyeiről ebben a projektben. A második rész pedig a tervezés fázisára fókuszál. Ennek során áttekintésre kerül, hogyan tervezik megvalósítani a filmértékelési és kedvelési funkciókat, valamint a hozzászólások lehetőségét a felhasználók között. A felhasználói élmény és az alkalmazás funkcionalitása terén végrehajtandó tervezési lépéseket is körüljárják. Ezzel a két részre osztott cikk alapos betekintést nyújt egy olyan webes alkalmazás tervezésének folyamatába, mely a filmkedvelők számára interaktív platformot nyújt kedvenc filmjeikkel kapcsolatos élményeik megosztására és együttélésére.

1. Bevezető

A téma választására azért került sor, hogy egy filmes webalkalmazás elkészítése történjen, mivel napjainkban egyre nagyobb népszerűségnek örvend a film készítése és a moziba járás. Mivel egyre több film kerül megtekintésre, így azok nyomon követése komoly kihívást jelenthet, és a filmek azonosításában gyakran fennáll annak a lehetősége, hogy korábban már láttuk őket. Emellett gyakran előfordulhat, hogy az adott filmről alkotott véleményünk és értékelésünk is feledésbe merül. Ezen kívül, egy olyan felületre van szükség, ahol az emberek megoszthatják tapasztalataikat, véleményüket az adott filmről, legyen szó pontozásról vagy szöveges formában történő kifejtésről. Ennek a weboldalnak a célközönsége főleg azokból áll, akik rendszeresen néznek filmeket, legyen az akár napi vagy heti rendszerességgel, és akik gyakran járnak moziba. A cikk célja, hogy az ezen webes alkalmazás fejlesztésének menetét érthető módon mutassa be.

Mivel az internet már a legtöbb ember mindennapi életének részévé vált, jelenleg szinte mindenki rendelkezik internet hozzáféréssel vagy olyan eszközzel, ami alkalmas az internet használatára, mint például okostelefon, tablet, laptop vagy számítógép. Az interneten keresztül banki tranzakciókat bonyolítunk, böngészünk híreket, videókat nézünk és kapcsolatot tartunk ismerőseinkkel. Ezért vetődött fel az az ötlet, hogy az internet kerüljön felhasználásra platformként, ahol maga a program elérhetővé válik, mivel a weboldalak könnyen hozzáférhetőek gyakorlatilag bármilyen eszközről, ami rendelkezik internet hozzáféréssel és rendelkezik egy böngésző alkalmazással.

A fejlesztés során azért esett a választás a Laravel keretrendszerre, mivel ez egy ingyenes, nyílt forráskódú PHP keretrendszer, ami viszonylag könnyen kezelhető és számos előre elkészített, hasznos funkciót tartalmaz. Továbbá, a Bootstrap 4 is beépítésre került, amely egy ingyenes front-end keretrendszer, mely az HTML, CSS és JavaScript nyelveket használja. Az adatbáziskezeléshez továbbá a MySQL adatbázis-kezelő rendszer került alkalmazásra, mivel ez az egyik legelterjedtebb adatbázis-kezelő az internetes környezetben, és grafikus felülettel rendelkezik, amely könnyen kezelhető.

2. A felhasznált technológiák

2.1. PHP keretrendszer

A jelenlegi időkben már többféle PHP keretrendszer közül választhatunk, mindegyiknek megvan saját létjogosultsága valamilyen mértékben. A Laravel azért került felhasználásra a PHP keretrendszerként, mert a nagyszerű dokumentáció és támogatás miatt könnyű az egyszerűbb hibákat kijavítani. Ezen kívül, ez a legelterjedtebb keretrendszer az emberek körében, így a legtöbb kérdés és válasz is az interneten erről található meg.

2.2. Laravel

A Laravel egy ingyenes, nyílt forráskódú PHP webes keretrendszer, amit Taylor Otwell hozott létre és a célja webes applikációk fejlesztése, amik követik a modell-nézet-vezérlő (MVC) szerkezeti mintát és a Symfony keretrendszeren alapulnak. A Laravel néhány funkciói közé tartozik a moduláris csomagoló rendszer dedikált függőségi vezetővel, különböző módok, hogy hozzáférjünk a relációs adatbázishoz, segédprogramok az alkalmazás telepítésben és karbantartásban, és az irányultság a könnyebben olvasható és kifejezhető szintaktikához. M. Bean (2015)

2.2.1. Funkciói

A Laravel főbb hasznos funkciói R. Nixon (2014):

- **Modularitás:** A Laravelt 20 különböző könyvtárra építették és magát különböző egyedülálló modulra szedték. Szorosan integrálva a Composer függőségi menedzserrel, ezek a komponensek könnyedén frissíthetők.
- **Tesztelhetőség:** Az alapjaitól úgy van felépítve, hogy segítse a tesztelést, a Laravel számos segítő funkcióval rendelkezik, amik elérhetővé teszik, hogy meggyőződjünk a metódusok bizonyos osztályokon vannak meghívva és még azt is, hogy megszemélyesítsenek bejelentkezett felhasználókat, ezáltal biztosra menjünk, hogy jó kód fut le jó időben.
- **Sablon motor:** Részben az ASP.NET-ben lévő Razor sablon motor ihlette. A Blade-el érkezik a Laravel, ami egy könnyű sablon nyelv, amelyben létrehozhatunk hierarchikus elrendezéseket előre meghatározott blokkokkal, amik dinamikus tartalmakat tárolnak.

- Hitelesítés/Bejelentkezés: Mivel a bejelentkezés egy elég gyakori funkció a webes applikációk körében, a Laravelben alaptól megtalálhatóak a regisztráció, bejelentkezés és még a jelszó emlékeztető is a felhasználók számára.
- Lekérdezés építő és ORM: A Laravel tartalmaz egy folyékony lekérdezés építőt, ami lehetővé teszi, hogy adatbázis lekérdezéseket futtassunk PHP kód segítségével, ahol szimplán egymásután futtathatunk metódusokat ahelyett, hogy SQL kódot íránk. Továbbá rendelkezik egy Objektum-relációs lekérdezővel (ORM) és ActiveRecord implementációval, amit úgy hívnak Eloquent, amely hasonló a Rubyban található Rails-el. Mind az ORM és a lekérdezés építő kompatibilis számos különböző adatbázis kezelővel, mint pl.: PostgreSQL, SQLite, MySQL és SQL szerver.

2.3. Apache

„Az Apache HTTP Server (röviden Apache) egy nyílt forráskódú webkiszolgáló alkalmazás, szabad szoftver, mely sok szabványt támogat melyeknek nagy része fordított modulok formájában áll rendelkezésre a mag kiegészítéseként. Ezek a modulok sok területet lefednek a kiszolgálóoldali programnyelvtámogatástól kezdve a hitelesítési sémákig. Az ismertebb, támogatott programnyelv modulok a a mod_perl, a mod_python, a Tcl és a PHP. A népszerűbb hitelesítési modulok a mod_access, mod_auth és a mod_digest.” „Statikus és dinamikus weboldalak közzétételére egyaránt használják. Sok webalkalmazást az Apache által nyújtott környezethez és szolgáltatásokhoz terveznek. Az Apache alkotja a webszerver komponenst a népszerű LAMP alkalmazáscsomagban, melynek további komponensei a MySQL adatbázisszerver és a PHP/Perl/Python programozási nyelvek mellett.” R. Nixon (2014)

2.4. HTML

„A Hypertext Markup Language röviden HTML, egy programozási előírás, amely azt szabályozza, hogy miként kell megírni a weblapokat ahhoz, hogy a számítógépek megértsék és megfelelő módon jelenítsék meg azokat.” J. Fielding (2014)

2.5. PHP

„A PHP hivatalosan a PHP: Hypertext Preprocessor elnevezést használja. Tulajdonképpen kiszolgáló oldali parancsnyelv, amit jellemzően HTML oldalakon használnak. A hagyományos HTML lapokkal ellentétben azonban a PHP parancsokat nem küldi el az ügyfélnek, azokat a kiszolgáló oldalán a PHP-értelmező motor dolgozza fel.” R. Nixon (2014)

2.6. CSS

A CSS az egy stílus leíró nyelv, amely lehetővé teszi a HTML oldalak megjelenésének formázását. „A stíluslapok számos formázási beállítást tesznek elérhetővé, megadhatjuk a betűtípusok különféle beállításait, a betű és sorközöket, a margókat és szegélyeket, és még sok minden mást. A méreteket különféle egységekben határozhatjuk meg, így dönthetünk a hüvelyk, a milliméter, a pont vagy éppen a pica mellett. A stíluslapok segítségével pontosan elhelyezhetjük a kívánt szöveget és grafikákat a weboldalakon, akár koordinátákkal, akár az oldal más elemeihez képest.” J. Fielding (2014)

2.6.1. Sass

A Syntactically awesome style sheets (Sass) egy CSS preprocesszor, ami annyit tesz, hogy egy érvényes Sass fájl kimenetét CSS formátumra alakítja. Ennek segítségével képesek vagyunk olyan CSS fájlokat létrehozni, amelyek sokkal átláthatóbbak és könnyebben kezelhetők.

2.6.2. Bootstrap

A Bootstrap ingyenes eszközök gyűjteménye, amik segítenek egy modern és reszponzív weboldal kidolgozásában. HTML és CSS alapú sablonokat tartalmaz gombokhoz, formokhoz, szöveghez, navigációhoz, menükhöz és egyéb hasznos kinézeti elemekhez.

2.7. MySQL

A MySQL a világ legnépszerűbb nyílt forráskódú adatbázisa. A bizonyított teljesítményével, megbízhatóságával és egyszerű használatával a MySQL lett a vezető adatbázis kezelő a webes alapú applikációk körében. R. Nixon (2014)

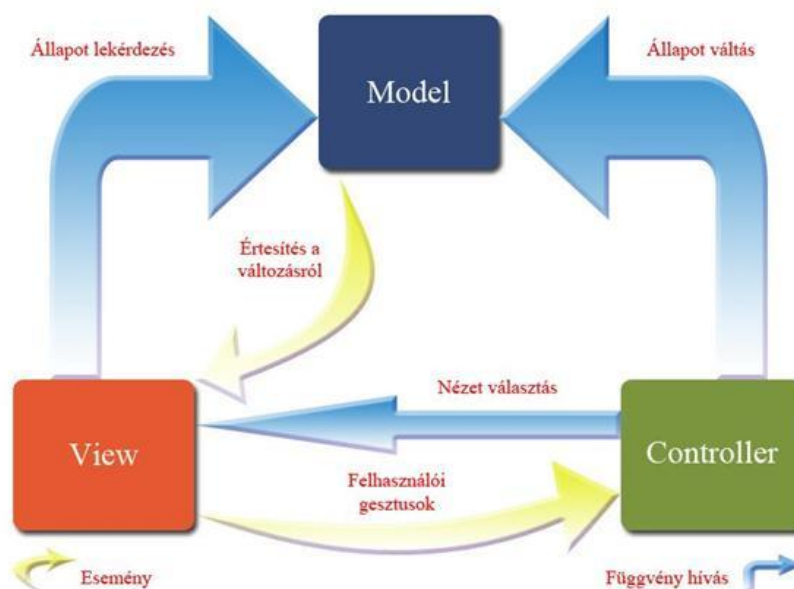
2.8. phpMyAdmin

A phpMyAdmin egy ingyenes szoftver, amit PHPban írtak. Célja, hogy megkönnyítse a MySQL adminisztrációt a Weben keresztül. Gyakran használt műveletek támogat, mint pl.: adatbázisok, táblák, kapcsolatok, indexek, felhasználók, jogok kezelése. Mindezt egy egyszerű felhasználói felületen keresztül érhetjük el. Azonban, ha szeretnénk továbbra is lehetőségünk van kézzel bevitt SQL utasítások végrehajtására is.

2.9. Az MVC architektúra

Az MVC egy olyan szoftver architektúra, ami a szerepkörök szétválasztására épül, ez kimondja, hogy egy adott programot 3 különálló részre szedünk szét, ahol minden egyes rész egy-egy adott szerepet tölt be. A három egység a következő:

- Controller (Írányítás): Irányítja az alkalmazást. Ide tartozik mind olyan dolog, ami a bemenetek feldolgozásával és fogadásával kapcsolatos. A felhasználó ennek segítségével tudja módosítani az alkalmazást.
- View (Megjelenítés): Maga a felhasználói felület (UI). Megjeleníti az alkalmazást a modell adatait felhasználva. A felhasználó ezt látja.
- Model (Modell): Ez az MVC egyik legfontosabb része mivel itt valósul meg az üzleti logikát. Kezeli az adatokat.



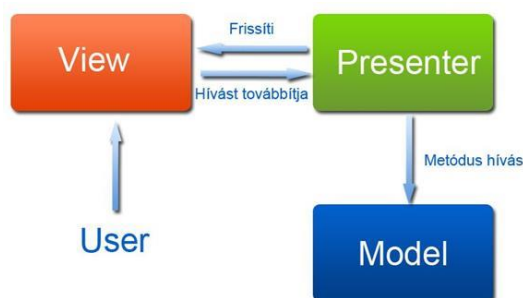
1. ábra: Az MVC architektúra

Az MVC elv szerint ezt a 3 egységet oly módon kell különválasztanunk egymástól, hogy azok csak a meghatározott szabályok szerint érintkezhetnek. Erre legfőbbképpen a hordozhatóság, információ rejtés, újra felhasználhatóság és a párhuzamos fejlesztés lehetősége miatt van szükség. Továbbá lehetővé teszi, hogy egy modellhez több nézet is tartozhat.

Az MVC mintának több továbbfejlesztése is létezik. Ezek közül a két legismertebb:

- MVP – Model View Presenter, magyarul Modell – Nézet – Megjelenítő: Itt a modell a nézet helyett a megjelenítővel áll közvetlen kapcsolatban és szolgáltat számára adatot.

- MVVM – Model View View-Model, magyarul Modell – Nézet – Nézetmodell: Ez az MVP továbbfejlesztése, itt a nézetmodell végez el minden feladatot a nézet helyett és a nézet feladata főként a megjelenítés a lehető legkevesebb logikával.



2. ábra: Az MVP architektúra

Az MVC előnyei:

- A kód sokkal könnyebben tesztelhető és karban tartható.
- Lehetővé teszi a párhuzamos fejlesztést.
- Könnyebbé teszi a komplexebb alkalmazások fejlesztést a három részre osztás miatt.

3. Fejlesztéshez szükséges eszközök

A Composer egy applikáció szintű csomag kezelő, amely a PHP függőségek és a szükséges könyvtárak kezelésére szolgál. Parancssoron keresztül futtatható és segítségével PHP függőségeket telepíthetünk. Azért van rá szükség, mert ha egynagyobb keretrendszert szeretnénk használni, mint például a Laravel, ami rengeteg függőséggel rendelkezik, és ezt mind egyesével telepítenénk fel, az egy hosszú és zavaros folyamat lenne. Azonban a Composer mindezt elvégzi helyettünk és könnyen menedzselhetővé és átláthatóvá teszi a projektünket.

3.1. XAMPP

A XAMPP teljesen ingyenes, könnyen telepíthető Apache disztribúció MariaDB, PHP és Perl komponenseket tartalmazva. A XAMPP nyílt forrású csomag úgy lett beállítva, hogy hihetetlenül egyszerű legyen a telepítése és használata.

3.2. Node.js

A Node.js egy nyílt forráskódú JavaScript alapú szoftverrendszer, amely főként kliens oldali szkriptelésben használatos. Továbbá a Node.js lehetőséget ad szerver oldali szkriptek és dinamikus weboldalak létrehozására mielőtt azok elérnék a felhasználó böngészőjét.

3.3. *Git*

A Git egy nyílt forráskódú, elosztott verziókezelő program. A célja, hogy a változásokat a forráskódban létrejött változásokat tartsa számon. Eredetileg a programozók közötti fejlesztés könnyítésére jött létre, de bármilyen fájlban létrejött változást nyomon követhetünk a segítségével. Linus Torvalds fejlesztette ki 2005ben a Linux kernel fejlesztéséhez.

3.4. *PhpStorm*

A programom a PhpStorm nevű grafikus fejlesztői környezettel (GUI) készítettem el. A PhpStorm egy kereszt platformos fejlesztő környezet, amely az IntelliJ IDEA platformon alapszik. Eleinte a Visual Studio Code-al próbálkoztam, de ebből hiányzott sok kényelmi funkció, amely a PhpStormban alaptól megtalálható. Támogatja a PHP 5.3-7.2 verzióit és valós idejű hibamegelőzést használ az általunk készített kódban. Továbbá rendelkezik automatikus kód kiegészítés funkcióval, debug móddal, amellyel megkönnyíti a hibakeresést és rendelkezik HTML, CSS és JavaScript szerkesztővel is.

4. Követelmények feltárása és az alkalmazás megtervezése

Az első dolgom a követelmények feltárása volt, mint például mit várok el magától az oldaltól, milyen keretrendszereket szeretnék használni a megvalósításához, milyen adatbázis rendszert szeretnék használni, és a kinézetét milyen módon szeretném alakítani.

4.1. *Célkitűzés*

A célom egy olyan weboldal megvalósítása, amely képes az alábbi követelményeknek megfelelni:

- Lehetősége legyen a felhasználónak létrehozni egy bejelentkező fiókot.
- Képes legyen a felhasználó böngészni a filmek között.
- Amennyiben látta az adott filmet, tudja azt kedvelni és értékelni.
- A felhasználó tudjon hozzászólásokat írni az adott filmekhez.
- Megtudja tekinteni kedvelt és értékelt filmjeit.
- Minden fajta eszközön használható és jól látható legyen a weboldal.
- Mindezeket dinamikusan megjeleníti a weboldalon.
- Olyan módon megvalósítani a weboldalt, hogy a forráskódja a későbbiekben könnyedén átlátható és fejleszthető legyen.

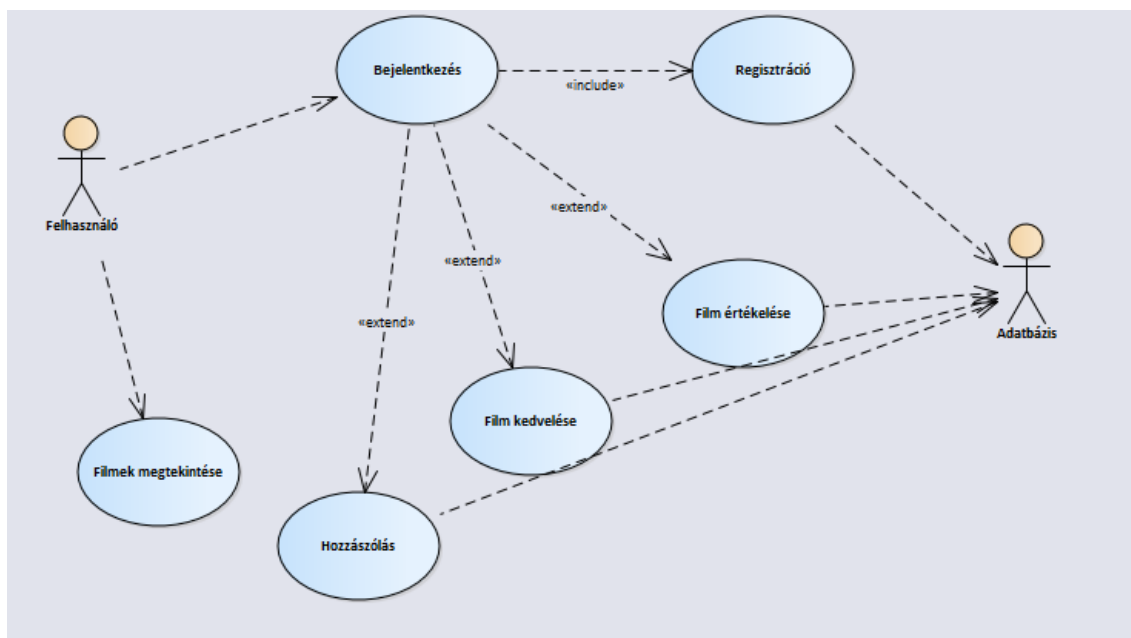
4.2. Fejlesztő környezet

Azért választottam a PhpStormot mint fejlesztői környezetet, mert ez az egyik legjobban személyre szabható GUI. Képesek vagyunk a felhasználói felület színét, a szintaxis kiemelés színét és akár a gyorsbillentyűket is átállítani. Továbbá rendelkezik beépített terminál ablakkal, ami nagyon fontos volt a program fejlesztése közben, hiszen a fejlesztés legtöbb fázisához szükségem volt rá. Lehetőségünk van további segédeszközöket úgynevezett plugin-okat telepíteni is, amelyek tovább bővíthetik a támogatási lehetőségeket.

Az egyik talán legfontosabb funkciója az integrált git kezelés. Nyomon követi a git által végrehajtott utasításokat és grafikus felületen megjeleníti azokat, így, ha valamit elrontottunk és szeretnénk visszavonni, akkor lehetőségünk van akár csak a kódrészletek megtekintésére és visszavonására is.

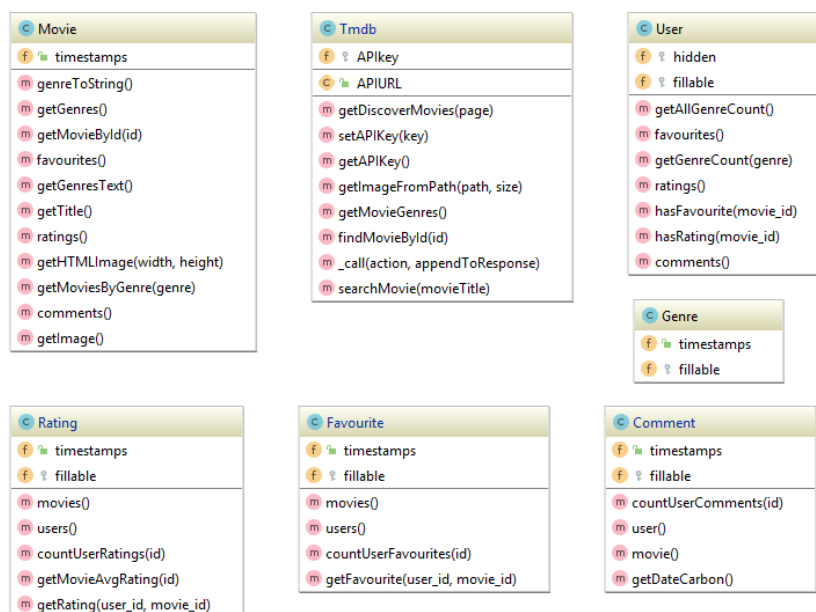
4.3. Diagramok

Szükséges volt megtervezni azt milyen módon szeretném, hogy a felhasználó az alkalmazás adott funkcióit elérje. A 3. ábrán látható diagram szemlélteti hogyan léphet kapcsolatba a felhasználó az alkalmazással és az adatbázis szerverrel.



3. ábra: Az alkalmazás használati diagramja

A tervezés során felhasználtam még egy osztály diagramot is (4. ábra), amelyben leírtam, hogy az alkalmazás adott osztályait milyen metódusokkal szükséges ellátni.



4. ábra: Az alkalmazás osztály diagramja

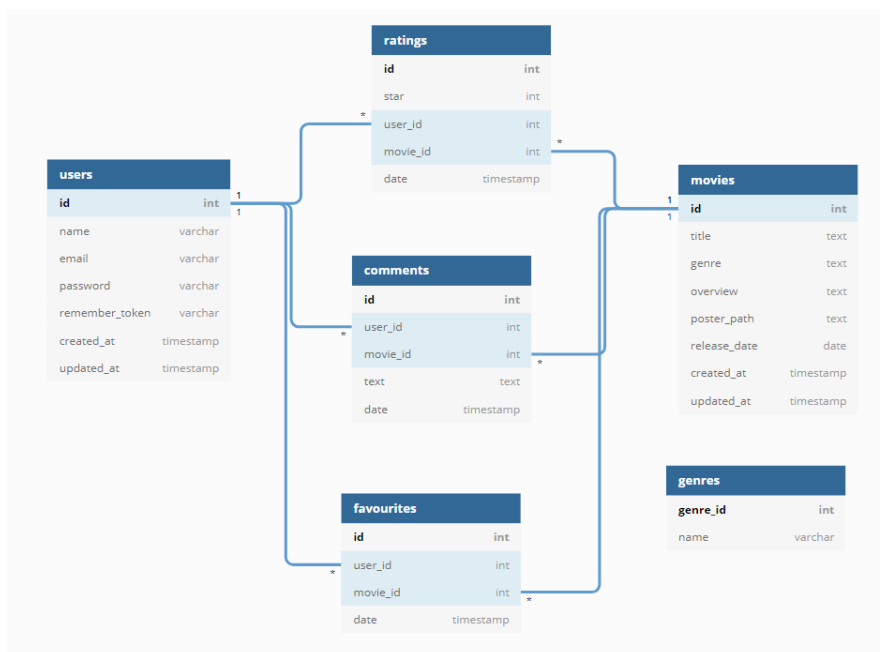
4.4. Adatbázis

Az Adatbázis megtervezésénél több pontot is figyelembe kellett vennem, mint például a későbbi bővíthetőséget, hány táblát szeretnék és ezek a táblák milyen kapcsolatban álljanak egymással. Mivel az Eloquent ORM a Laravel része ezért az adatbázis megtervezése előtt, az osztályokat kell elkészítenem. Figyelembe kell vennem, hogy a weboldalon megszeretném valósítani a felhasználók kezelését, ami a könnyebb azonosítás miatt jelentős, mivel így képes vagyok a felhasználó által végzett akciókat elmenteni. Továbbá szükségem van egy táblára, amiben a filmekhez fűzött hozzászólásokat tárolom, ennek a táblának kapcsolatban kell állnia a filmek és a felhasználók táblájával is. Értelmszerűen szükségem van egy film táblára is, amiben a filmek adatait tárolom. Mivel céljaim között szerepel az is, hogy a felhasználók képesek legyenek a filmeket kedvelni, illetve értékelni, két további táblára van szükségem, amelyek a film és a felhasználó táblához is kapcsolódnak.

Mielőtt létrehozom a táblákat, meg kell terveznem a táblában szereplő adatok típusát és azt is, hogy milyen az adott táblák milyen mezőkből épüljenek fel. Ezután meg kell határoznom az adott oszlopok feltételeit is, ezek közül a fontosabbak az elsődleges kulcs, ami alapján egyedi módon meg tudom különböztetni egymástól a rekordokat. A felhasználók táblához szükségem van még egy egyedi (UNIQUE) feltételre is, mivel nem szeretném, hogy két azonos email cím szerepeljen ugyan azon táblában. További megfontolandó feltétel a NOT NULL amennyiben nem szeretném, hogy egy rekord érték nélkül kerüljön be.

4.4.1. Az adatbázis szerkezete

Az adatbázisban összesen hat tábla szerepel, amelyekből öt kapcsolatban áll egymással.



8. ábra: Adatbázis szerkezete ábrázolva

Az adatbázis táblái a következő módon alakulnak:

- **users**: Ez a tábla tartalmazza a felhasználóval kapcsolatos összes szükséges információt. Az id a felhasználó azonosítója ez egy elsődleges kulcs ezzel könnyítve az azonosítást és ezzel a mezővel kapcsolódik össze a többi tábla. Az id mező egy a többhöz kapcsolatban áll a ratings, comments és a favourites táblával is. A name a felhasználó neve, amit regisztráció során adhat meg. Az email mező tartalmazza a felhasználó e-mail címét, amely egy UNIQUE mező, ezzel megakadályozva, hogy több azonos e-mail című felhasználó legyen. A password mező a felhasználó jelszavát tárolja, amely bcrypt titkosítással van védve. A created_at mező tárolja, hogy az adott felhasználó mikor regisztrált.
- **ratings**: Ebben a táblában találhatóak meg a felhasználókhoz és filmekhez kapcsolódó értékelések. Az id mező elsődleges kulcs, amely az adott értékelések azonosítójaként szolgál. A star mező tartalmazza az értékeléseket (1-5). A user_id mező az adott felhasználó azonosítója, amely össze van kötve több az egyhez kapcsolattal a users táblával, mivel egy felhasználónak több film is lehet értékelése. A movie_id mező az adott film azonosítója, ami szintén több az egyhez kapcsolattal van kapcsolva a movies táblához. A date mező az értékelés leadásának időbélyegét tartalmazza.

- favourites: Ez a tábla nagyon hasonló a ratings táblához, az egyetlen kivétel, hogy mivel itt a program csak azt vizsgálja, hogy van-e kapcsolat a felhasználó, illetve a film között, mert ennyiből már meg lehet állapítani, hogy az adott felhasználó kedvelte a filmet vagy sem.
- movies: Itt tárolom az összes filmekkel kapcsolatos adatot. Az id mező, mint a többi táblánál itt is elsődleges kulcs és a filmek egyedi azonosítására szolgál. A title mezőben tárolom a filmek címeit, ennek a mezőnek nemszükséges egyedinek lennie mivel léteznek filmek azonos névvel. A genre mező tárolja a filmek műfaját, egy filmnek lehet több műfaja is, ezeket vesszővel elválasztva tárolom. Az overview mező a filmek leírását tartalmazza, ami egy hosszabb szöveg formájában van jelen. Ezt követi a poster_path oszlop, amelyben a filmekhez tartozó poszterek linkjét tárolom string formájában. Szükség van a filmek megjelenését tároló mezőre is, amely tárolja a dátumot, ez a mező a release_date. Végül a created_at mező a rekordok felvitelének pontos időpontját tárolja.
- genres: Ez a tábla egyfajta gyorsítótárként szolgál. Az API segítségével töltöm a fel a műfajokat a hozzájuk tartozó azonosítókhoz. Itt megtalálható egy genre_id mező, ami a műfaj azonosítója és egy name mező, ami a műfaj nevét tárolja.

4.5. Felhasználói felület

Ez az elsődleges kapcsolódási pont a számítógép és a felhasználó között. Ezért is felhasználói felület az egyik legfontosabb része a programnak hiszen, ha ez nem lenne vagy nehéz lenne kezelni akkor senki se szeretné használni a weboldalt. Így tehát ennek tudatában kellett megterveznem a felhasználói felületet, hogy a használatához ne kelljen semmiféle leírás és magától értetődő legyen. Természetesen a megjelenés az egy szubjektív fogalom így a cél nem csak az, hogy a felület szép és letisztult legyen, de néhány alapvető tervet figyelembe kell venni, hogy jobb eredmény érjünk el. Olyan módon kell megterveznünk a felületet, hogy az illeszkedjen a felhasználó szakértelméhez és elvárásához.

Néhány nagyon fontos szempontot figyelembe kell venni:

- Az emberek rövidtávú memóriája korlátozott ezért nem ajánlatos egyidejűleg túl sok információval terhelni őket.
- A rendszert úgy célszerű tervezni, hogy azt nem a saját képességeinkre írjuk, hanem a felhasználó képességeit próbáljuk előtérbe helyezni.

Az egyik legfontosabb dolog, hogy ha valaki rátéved a weboldalunkra akkor annak a figyelmét megragadjuk, ezért az alkalmazás fő oldalán a filmeket viszonylag nagy és színes poszterrel jelenítettem meg. Továbbá a főoldalon látható filmek a jelenleg népszerű filmeket tartalmazzák így elég nagy a valószínűsége, hogy a felhasználó rá fog kattintani valamelyik filmre.

Az oldalt úgy készítettem el, hogy a navigálás az oldal részei között magától értetődő és jól átlátható legyen. Ezt a navigálást egy navigációs bár segítségével valósítottam meg. Ez minden oldalon az oldal legtetején található meg, így a felhasználó mindig tudja hol tud az oldalak között navigálni. Itt található meg továbbá a be és kijelentkezés is, ez fontos, hogy ezt nem szabad elrejteni különféle almenü pontokba, mert ha a felhasználó nem találja meg egyből, akkor ez frusztrációt okozhat.

Ami után valaki rákattint egy adott filmre fontos, hogy az információt, amelyet látni szeretne az oldal tetején helyeztem el a navigációs bár alatt közvetlenül. Így itt megtalálhatja a film címét, műfajait, megjelenését, leírását és a film poszterét. A kevésbé fontos információkat, mint például a hozzászólások és a hozzászólás írása a film információi alatt találja meg.

Arra is odakellett figyelnem, hogy a tovább fejlesztési munkák a lehető legkönnyebben kivitelezhetők legyenek, ezért a weboldal kinézetét több részre bontottam fel. A navigációs bár, a header és a footer rész beszúrását egy külön fájlban tárolom. Továbbá a CSS fájlt a lehető legtöbb helyen elláttam kommentekkel, hogy ezzel is könnyítsem a jövőbeni munkát.

5. Konklúzió

A projektben egy filmes webalkalmazás elkészítéséhez szükséges terveket készítettem el, amely Laravel keretrendszerrel kerül majd implementálásra. A tervek megvalósítása előtt betekintést nyertem a REST alapú rendszerek felépítésébe és kezelésébe, megismertem az MVC technológián alapuló alkalmazások előnyeit. A felhasználói felületet Bootstrap keretrendszer segítségével kerül majd elkészítésre.

Irodalomjegyzék

Bean, M. (2015). *Laravel 5 essentials*. Packt Publishing Ltd.

Nixon, R. (2014). *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*. "O'Reilly Media, Inc."

Fielding, J. (2014). *Beginning responsive web design with HTML5 and CSS3*. Apress.

Rövid szakmai életrajz

2020 óta alkalmazás fejlesztőként dolgozom a Máv Szolgáltató Központ Zrt.-nél. A szakterületem az automatizált tesztelés és szoftverfejlesztés amelyeket Java nyelven végzek.