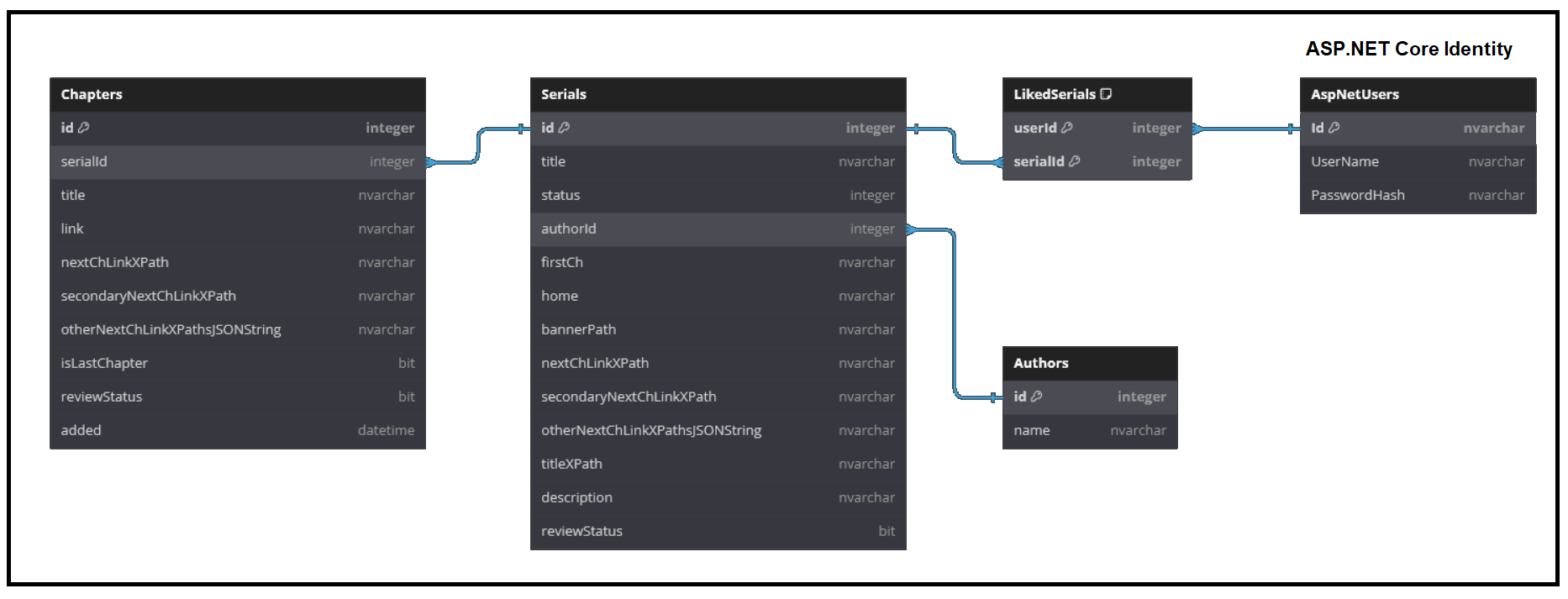
**Webserial-Tracker**

Mi az a webserial? A webserial, vagy webnovel egy regény, amit a szerző nem könyv formában ad ki, hanem az interneten keresztül. Ez történhet az írót saját blogján, valamilyen CMS (Content Management Software) - például WordPress - segítségével készítve, vagy különböző kifejezetten ezt a célt szolgáló weboldalon. Ezek ismertebb példái közé tartozik a WattPad (<https://www.wattpad.com/>), ScribbleHub (<https://www.scribblehub.com>), Royal Road (<https://www.royalroad.com>), stb. Ezek a sorozatokra talán úgy érdemes gondolni, mint regényekre, amiket nem egyszerre adnak ki, hanem változó gyakorisággal teszenk közzé új fejezeteket. Ebben hasonlít, és talán átvette a helyét a régi, újságokban fejezetenként publikált történeteknek. Az hogy milyen gyakran jön ki új része, az írónként és sorozatonként változik, van hogy havonta egy, van hogy egy héten néhány, vagy akár naponta egy fejezet. Mindezek ellenére ezek a web serial-ok egy egyedi műfaj, és ezzel együtt járnak különböző erősségek és hátrányok is. Mivel sokkal kisebb pénz/idő/energia befektetés regisztrálni egy oldalra, mint kiadni egy könyvet - akár magánszemélyként, és főleg kiadóval kiadatni - sokkal többen kezdenek hozzá. Megnézve csak az egyik oldalt, ennek a dokumentációnak az írása pillanatában 72.804 különböző sorozatot tart nyilván, csak a Royal Road. Egy másik fontos tulajdonságuk ezeknek a webserial/webnoveleknek, ami releváns ezzel a projekttel kapcsolatban, hogy egy-egy ilyen sorozat meglepően hosszú tud lenni. Az egyik legnépszerűbb, és egyben leghosszabb ilyen történet, a “The Wandering Inn”, az írás pillanatában nagyjából 12.4 millió szóból áll. Összehasonlításképp, a három Gyűrűk Ura könyv együtt egy kicsit kevesebb mint fél millió. Ez azzal is jár, hogy ezek a sorozatok/regények éveken át futnak, például az előbb említett 12.4 millió szavas művet 2016-ban kezdte el az író, és a mai napig rendszeresen jönnek ki újabb és újabb részek.

Manapság amilyen széles körben elérhetők weboldal készítő platformok, amik mindenki számára lehetővé teszik saját honlapok készítését, talán nem meglepő, hogy nincs egy centralizált gyűjtőhelye ezeknek a sorozatoknak. A legnépszerűbbek közül több az író saját WordPresses blogján lett csak publikálva, és bár nem ritka, hogy valaki a nagyobb oldalak közül többön is egyszerre közzéteszi a művét, ritka, hogy mindegyiken publikálná. Nincs egy oldal, amit meglátogatva, felhasználók elérhetnének minden webserialt.

Ebből a hiányosságból született meg az ötlet ehhez a projekthez. Fejleszteni szerettünk volna egy oldalt, ahova bárki felmehet, és egy helyen tudja követni az összes kedvenc sorozatát.

Az volt a célunk mikor elkezdtük, hogy egy helyen elérhető legyen minél több web serial/webnovel. Fejleszteni akartunk egy olyan felületet, a felhasználók nem csak követni tudják saját kedvenc sorozataikat, de felfedezhetnek újakat, attól függetlenül, hogy valamelyik ismertebb, népszerű oldalon lett publikálva, vagy az író saját kézzel létrehozott blogján.

Adatbázis

1. ábra - Az adatbázis felépítése (users tábla nem teljes)

Adabázisnak, mivel a projekt backend részét C#-ban terveztük írni, a Microsoft SQL Server-t használtuk a Visual Studio-n keresztül, amivel Entity Framework segítségével interaktáltunk. Mivel ASP.NET Core Identity-t használtuk a felhasználók kezelésére, a felhasználó adatokat tartalmazó táblákat a Visual Studio automatikusan generálta. Ezeken kívül 4 táblás adatbázist terveztünk.

Ezekből a legegyszerűbb az **Authors** tábla volt, amiben a sorozatok íróit tároltuk el. Sorozatok hozzáadásakor, módosításakor és törlésekor ezt a táblát is ennek függvényében módosítjuk, figyelve rá, hogy ha egy írónak nincs egyáltalán műve az adatbázisban, akkor ebből a táblából is törölve legyen:

-id – a tábla Primary Key-e, az adatbázis által automatikusan generált egy egyedi integer azonosító

-name – szöveg típusú adat, az író neve

Következő, és a legtöbb különböző oszlopot tartalmazó táblába maguk a sorozatokat tároltuk, ez a **Serials**:

-id – Primary Key, automatikusan generált integer, egyedi azonosító

-title – nvarchar típusú, a sorozat címét tartalmazza

-status – a sorozat állapotát jelöljük itt, már befejeződött, még készül, szünetel, vagy félbe van hagyva, vagyis “Ongoing”, “Completed”, “Hiatus”, vagy “Abandoned”. Létrehoztunk egy enum objektumot, ezekkel az értékekkel, így az adatbázisban egy integerként vannak tárolva

-authorId – ez az oszlop egy Foreign Key, és az authors tábla id integer mezőjével lett összekapcsolva, hogy kifejezhessük hogy egy író gyakran több sorozatot is írthatott, ez egy “many-to-one” kapcsolat

-home – több oldal úgy mutatja be a sorozatokat, hogy van egy oldal ahol a felhasználó láthat adatokat róla, egy fülszöveget, stb., és innen vezet egy link az első fejezethez, ahol lehet kezdeni az olvasást. Itt ezt a linket tároljuk. Ez a mező eredetileg nem volt része a táblának mikor még e fejlesztés megkezdése előtt gondolkodtunk az adatbázisról, de végül hasznosnak ítéltük hogy a felhasználó elérhesse a sorozat bemutatkozó oldalát

-firstCh – a sorozat első fejezetének linkje, ezt használja a webscraping servíz kezdőoldalként amikor először hozzáadja a sorozat fejezeteit az adatbázishoz

-nextChLinkXPath – annak az elemnek az XPath-ként meghatározott helye az oldalon, ami a következő fejezetre mutató linket tartalmazza. Gyakorlatban ez sokszor egy “Next Chapter” vagy hasonló feliratú gomb. A webscraping servíz ezt használja hogy a következő fejezet oldalára navigáljon.

-secondaryNextChLinkXPath – a legtöbb sorozatnak van egy “Next Chapter” gombja a fejezet elején, és a végén, ezt a redundancia érdekében tároljuk el, hogy ha esetleg az elsődleges link nem működik valamiért, ez át tudja venni a szerepét a webscraping során

-otherNextChLinkXPathsJSONString – itt tároljuk JSON array formátumban, (amit egy wrapper propertyvel érünk el a kódban, ami JSON formátumból és JSON formátumba alakítja az adatokat) az egyéb következő fejezet linkjeinek XPath-ját. Továbbá a webscraping servíz hozzá tud adni dinamikusan talált linkek XPathját, ha szükséges.

-titleXPath – ezt a mezőt a fejezetek (chapters) adatbázishoz adásánál használjuk, ez az Xpath kifejezés ami kifejezi hol van a honlapon a fejezet címe

-description - rövid ismertető leírás a sorozatról, egy könyv fülszövegének felelne meg

-reviewStatus – ez egy bit értékű oszlop, ami alapvetően 0, amíg egy admin nem ellenőrzi a sorozat adatait, és amit a frontend (booleanként) arra használ, hogy eldöntse melyik sorozatot jelenítsen meg, illetve a webscraping servíz ellenőriz, mielőtt elkezdi a fejezetek adatbázishoz adását

A **Chapters** tábla az, ahol a különböző sorozatok fejezeteit tároljuk. Ezt a táblát automatikusan feltölti a backend, mikor egy admin hozzáad egy új sorozatot, vagy egy admin kiegészít egy felhasználók által javasolt sorozatot a megfelelő adatokkal.

-id – egyedi azonosító szám, elsődleges kulcs

-serialId – Foreign Key, annak a sorozatnak az Id-ja, amelyikhez ez a fejezet tartozik, a **Serials** id mezőjével van összekötve

-title – a fejezet címe, a webscraping metódus a **Serials** titleXPath mezője alapján rendeli hozzá

-link – az url, ahova a fejezetet feltöltötték, azért tároljuk, hogy a felhasználó meg tudja látogatni, ne kelljen a sorozat honlapjáról elnavigálni a legutóbbi fejezethez

-nextChLinkXPath =>

-secondaryNextChLinkXPath =>

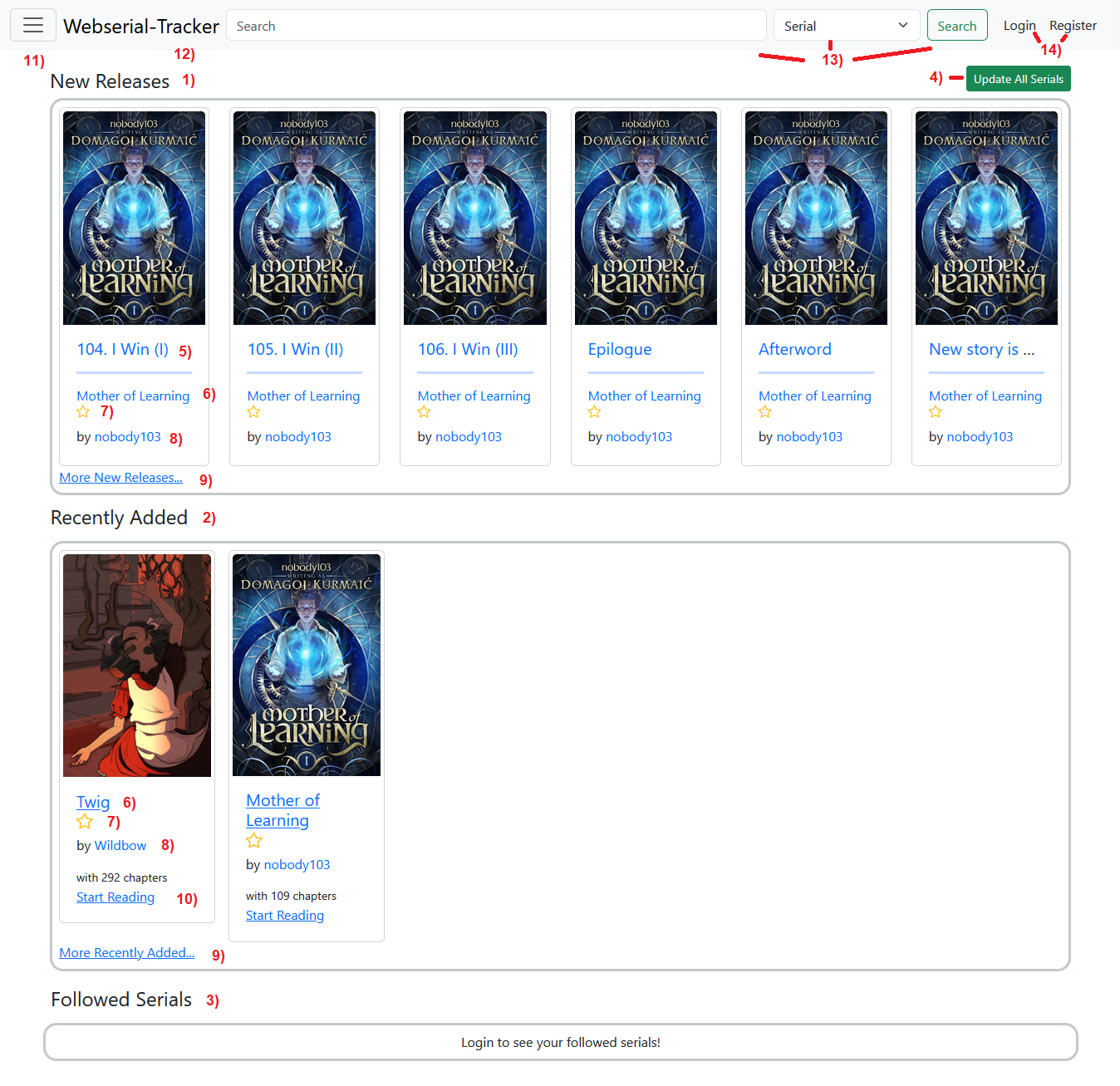
-otherNextChLinkXPaths – ez a három mező ugyan azt tárolja mint a **Serials** táblában, a következő fejezet gombjának XPath kifejezésként meghatározott helyzetét az oldalon, és azért vannak itt is külön eltárolva, hogyha a webscraping logika valami nem várt hibába ütközik, például ha az összes link más helyen van, mert például megváltozott az egész oldal felépítése, egy admin meg tudja változtatni, és tudjon folytatódni a fejezetek felvétele

-isLastChapter – egy bit oszlop, alapértéke 0, amit aztán a ScrapingService-ben meghatározott metódus, mikor hozzáadja az adatbázishoz a fejezetek, az utolsó fejezetnél 1-re változtat. Ezt az értéket használja a frontend, hogy figyelmeztetni tudja az adminokat, hogy ez a fejezetet ellenőrizni kell, hogy valóban ez-e az utolsó, vagy hibába ütközött a ScrapingService

-reviewStatus szintén egy bit érték, és szintén alapvetően 0 értékű, akkor válik 1-é, ha egy admin ellenőrizte, és vagy megváltoztatta az adatait, vagy elfogadta. Mindemellet az “isLastChapter” és a Serial/status értékkel értékkel együtt arra használjuk a sorozatok update-lésénél (mikor ellenőrizzük, jött-e ki új fejezete a sorozatnak), hogy gyorsabb végrehajtást biztosítsunk, kihagyva a már befejezett vagy félbehagyott sorozatokat.

Felhasználói Dokumentáció:

Mikor valaki először megnyitja az oldalt, a Home oldal fogadja.



A főoldal 3 fő részre van osztva:

1) az adatbázisban rögzített sorozatok legújabb fejezetei, a “New Releases” címszó alatt,

2) a legújabban felvett sorozatok, a “Recently Added” részben,

3) illetve a “Followed Works” alatt, ha a felhasználó bejelentkezett, azok a sorozatok amiket követett külön meg vannak jelenítve.

A New Releases szekció fölött jobb oldalt 4) látható az “Update All Serials” gomb, amire kattintva le lehet kérdezni a webscraping szervíztől, hogy jött-e ki újabb része az adatbázisba felvett sorozatoknak.

5) A fejezetcím egy link, ami a fejezet oldalára mutat, hogy a felhasználó egy kattintással el is kezdhesse olvasni.

6) Itt látható a különböző sorozatok címe, ami elvisz a SerialDetails komponensre, ahol a sorozat részletes adatait lehet megtekinteni.

7) A sorozat címe után, a sárga csillagra kattintva tudja a felhasználó követni a sorozatot/abbahagyni a sorozat követését.

8) Mindenhol fel van tüntetve a sorozat vagy a fejezet írója, akinek a nevére kattintva megtekinthetünk minden sorozatot attól a bizonyos írótól az adatbázisban. A Recently Added, és a Followed Serials alatt, ha a “Start Reading” gombra kattintunk, annak a sorozatnak az első fejezetének oldalán fogjuk találni magunkat.

9) A főoldalon a legújabbban megjelent fejezetkből, és a legutóbb hozzáadott sorozatokból is csak maximum 6-ot láthatunk, de ha ennél többre kíváncsi a felhasználó, a “More New Releases/Recently Added” gombokra kattintva láthatjuk az összes tárolt sorozatot a megfelelő szempont alapján sorba rendezve.

10) Mind a Recently Added és a Followed Serials pontok alatt, a “Start Reading” linkre kattintva eljuthatunk annak a sorozatnak az első fejezetének oldalára.

Navbar

Az oldal tetején található a navigációt elősegítő úgynevezett “navbar”, ami a honlapon mindenhol elérhető, és amin keresztül el tudunk érni néhány alapvető funkció.

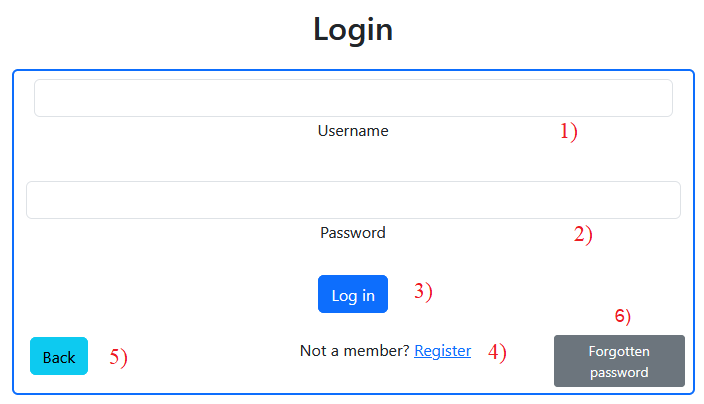
12) A honlap nevére kattintva visszatérhet erre az oldalra bárhonnan máshonnan.

13) A search mezőbe írva tud keresni, a jobb oldalán lévő legördülő menü segítségével pedig ki tudja választani, hogy a sorozatok címében vagy az írók között szeretne keresni, mi után ezt a keresést a “Search” gombra kattintva kezdheti meg a felhasználó.

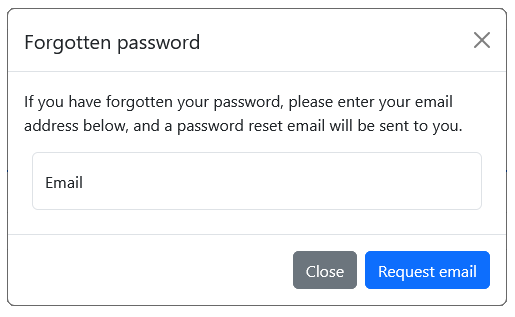
14) Az utolsó két gomba navbar-on a “Login” és a “Register” gombok. Értelemszerűen a “Login” a bejelentkezési oldalra vezet, míg a “Register” a regisztrációhoz. Ez megváltozik, ha a felhasználó bejelentkezett. Ilyenkor a Login helyén a felhasználó nevét látnánk, amire kattintva lehetőségünk lenne, a “Change Password” opcióra kattintva megváltoztatni a jelszavunkat, vagy a “Logout”-ra kattintva kijelentkezni, illetve, a Register opció helyett egy szintén egy Logout gombot látnánk.

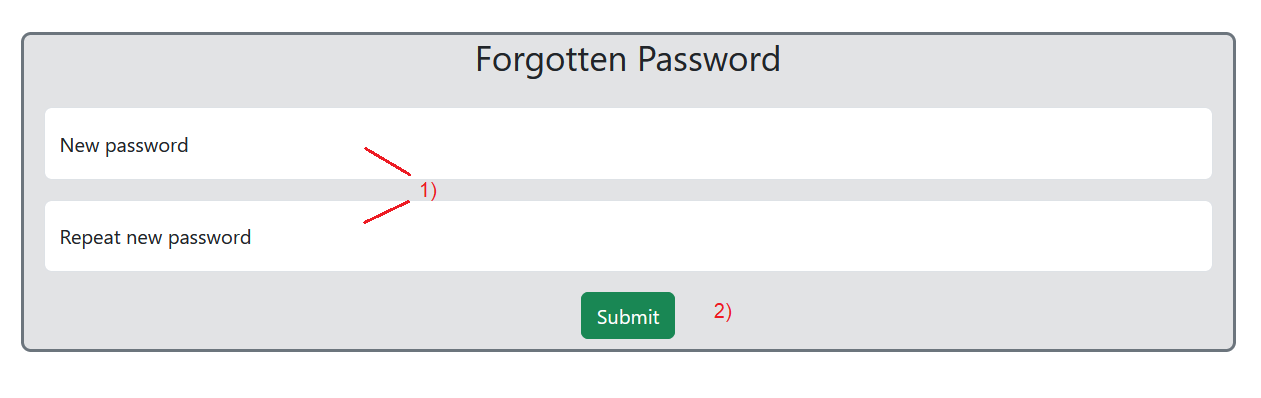
11) A még nem lett említve a három vízszintes vonalból álló (“hamburger”) gomb, amire kattintva láthatóvá válik az oldalsó navbar. Az ábra egy olyan állapotot mutat, mikor az adminként jelentkezett be valaki, mivel így látható a legtöbb elem. A honlap neve 1) mindig ott lesz, és hasonlóan a oldal tetején található navbar-hoz, visszavisz a főoldalra. Attól függően, hogy a felhasználó be van-e jelentkezve, legalul 7) vagy egy Login, vagy egy Logout gombot láthatunk.

Ha a felhasználó bejelentkezett, viszont nincs admin hozzáférése, csak a “Propose Serial” linket 3) fogja látni itt, amire kattintva eljut a felületre ahol javasolhat új sorozatot hozzáadni az adatbázishoz. Ha a felhasználó bejelentkezett, és van admin jogosultsága, elérhető válik 4 újabb link.Az első, az “Add Serial” arra az oldalra irányít át bennünket, ahol adminként új sorozatot addhatunk az adatbázishoz, annak minden szükséges adatát megadva. Természetesen adminként is látjuk, a Propose Serial menüpontot. Ez után a következő, a “Manage Serials” arra az oldara visz, ahol adminként módosítani tudjuk a sorozatok adatait, illetve meg tudjuk adni az ahhoz szükséges adatokat, hogy egy a Proposed Serial oldalon keresztül javasolt sorozat ellenőrzött státuszát igazra módosítsuk, és így a fejezeteit hozzáadjuk az adatbázishoz, illetve felhasználók számára elérhetővé tegyük. Következő, a “Manage Users” link 3), egy olyan felületre visz, ahol az adminok tudják kezelni, és módosítani a felhasználók adatait. Végül pedig a “Manage Chapters” link 5), ahol minden egyes chapter adatai külön megtekinthetőek és módosíthatók az admin által, amire akkor lehet szükség, ha a fejezetek hozzáadása közben a webscraping szervíz hibába ütközött.

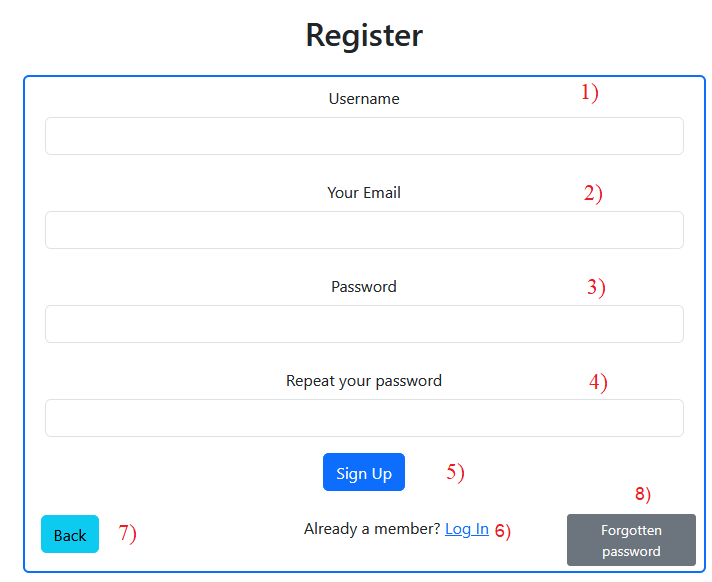
Login - Bejelentkezés

Itt tud a felhasználó, ha már regisztrált, bejelentkezni. Két adatra van szükség a bejelentkezéshez, a felhasználónévre 1) és a jelszóra 2). A “Login” gomb 3) megnyomásakor, ha helyes, létező adatokat adott meg a felhasználó, be lesz léptetve, és az automatikusan visszairányítódik a főoldalra. Ha nem helyes adatokat adott meg, akkor az alkalmazás értesíti, hogy ilyen adatokkal nem létezik felhasználó. A “Register” gomb 4) célja, hogy ha valaki még nincs regisztrálva, ezzel egyszerűen át tudjon váltani az oldalra, ahol lehetősége lesz rá. A “Back”, Vissza gomb 5) egyszerűen visszaviszi a felhasználót a főoldalra.

Végül, a “Forgotten Password” gomb 6), arra az esetre, ha valaki elfelejtette a jelszavát. Ide kattintva megjelenik egy ablak, ahol ha beírjuk a felhasználónk email címét, majd a “Request email” gombra kattintunk, kapni fogunk egy emailt, ami tartalmaz majd egy linket, ahol be tudunk állítani egy új jelszót magunknak.

Password Reset

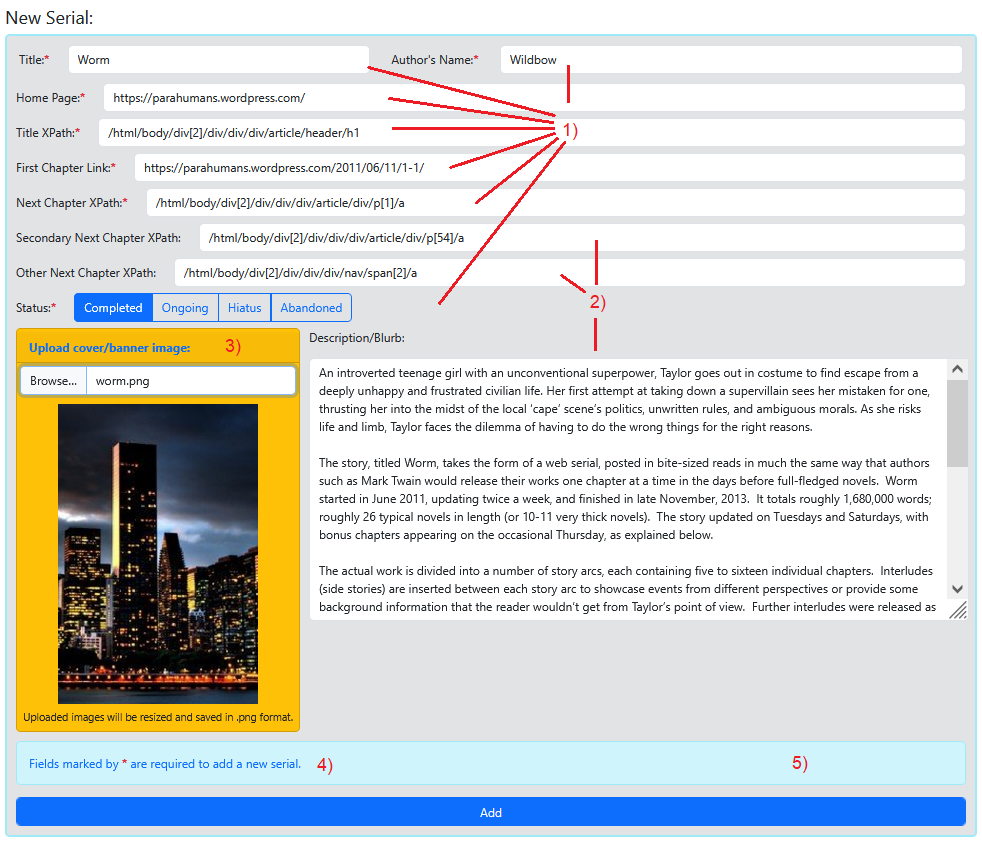
Erre az oldalra akkor jutunk, ha követjük az emailben kapott linket. Itt be tudunk írni egy új jelszót a felhasználónknak 1), és ha a két jelszó megegyezik, a “Submit” gombra kattintva meg tudjuk változtatni az elfelejtett jelszavunkat. Ha ez sikeres, át leszünk irányítva a Login oldalra.

Register - Regisztráció

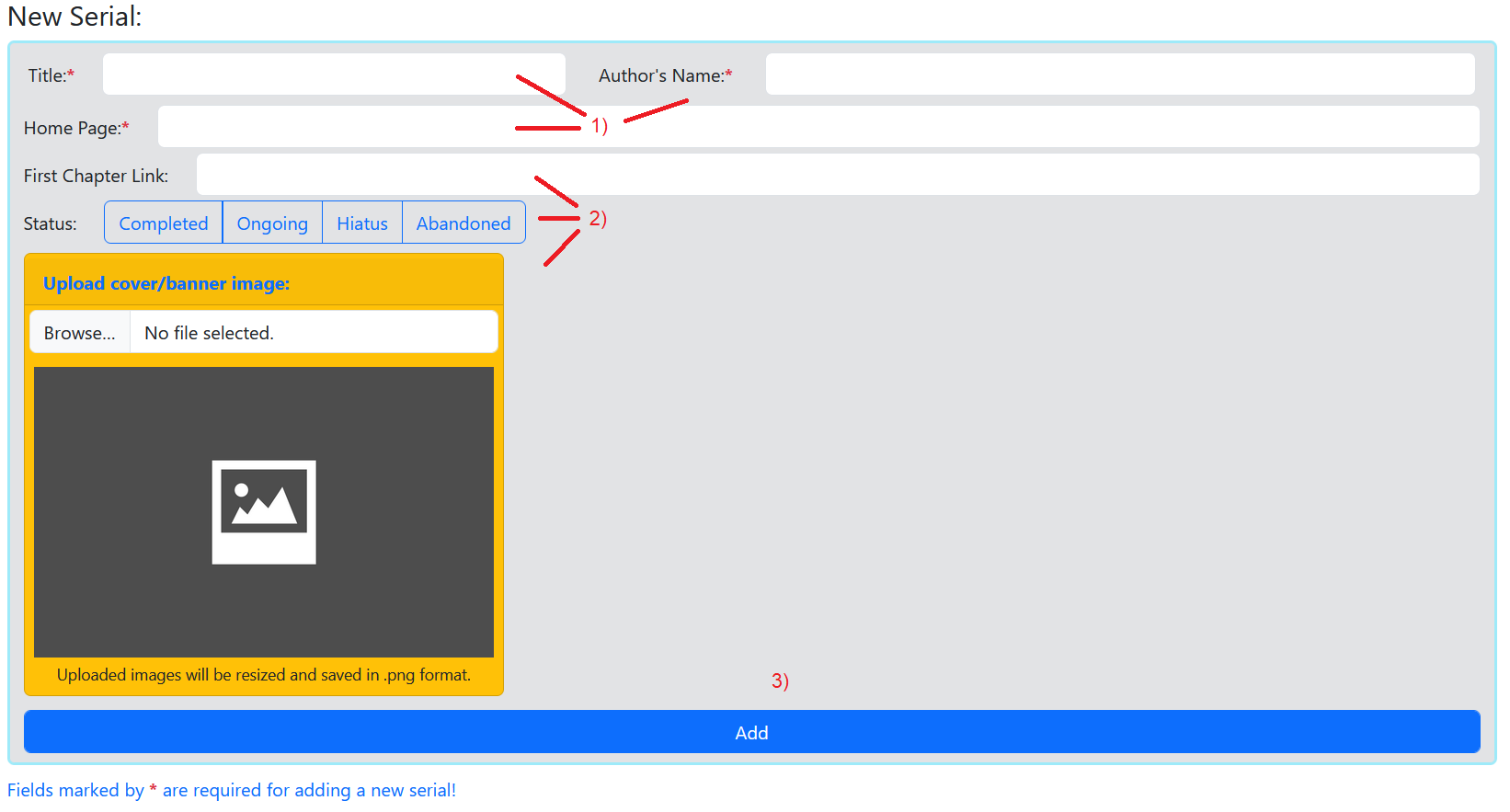
Ha valakinek nincs még felhasználói fiókja, de szeretne regisztrálni, itt teheti meg. Ez a komponens elérhető egyenesen a főoldalról, vagy a Login komponensen keresztül. Három adatot kell a felhasználónak megadnia, hogy regisztrálni tudjon: felhasználónevet 1), email címet 2), és egy jelszót 3) & 4). A regisztrációhoz kétszer kéri el a jelszót, hogy növelje az esélyét, hogy nem fogja azonnal elfelejteni. Mind a négy mező kitöltése szükséges, és a két jelszónak egyeznie kell hozzá, hogy sikeresen regisztrálni tudjon a felhasználó. Ha helyesen van kitöltve az űrlap, és a felhasználó rákattint a “Register” gombra 5), az alkalmazás elmenti az adatbázisba a felhasználó adatait, és átirányítja a Login oldalra, ahol értesíti hogy sikeresen regisztrált. Itt is lehetőség van átirányítódni a bejelentkezés oldalra, a “Login” 6) link segítségével, illetve visszatérni a főoldalra a “Back” gombbal 7). Itt is láthatjuk a “Forgotten password” gombot 8), aminek itt is ugyanaz a funkciója, megjeleníteni a ablakot, amin keresztül egy emailt kérhetünk ha elfelejtettük a jelszavunk.

Add Serial - Sorozat Hozzáadása

Ha a bejelentkezett felhasználónak admin jogosultságai vannak, el tudja érni az “Add Serial” oldalt, aminek a linkjére kattintvaa következő ábrát láthatjuk:

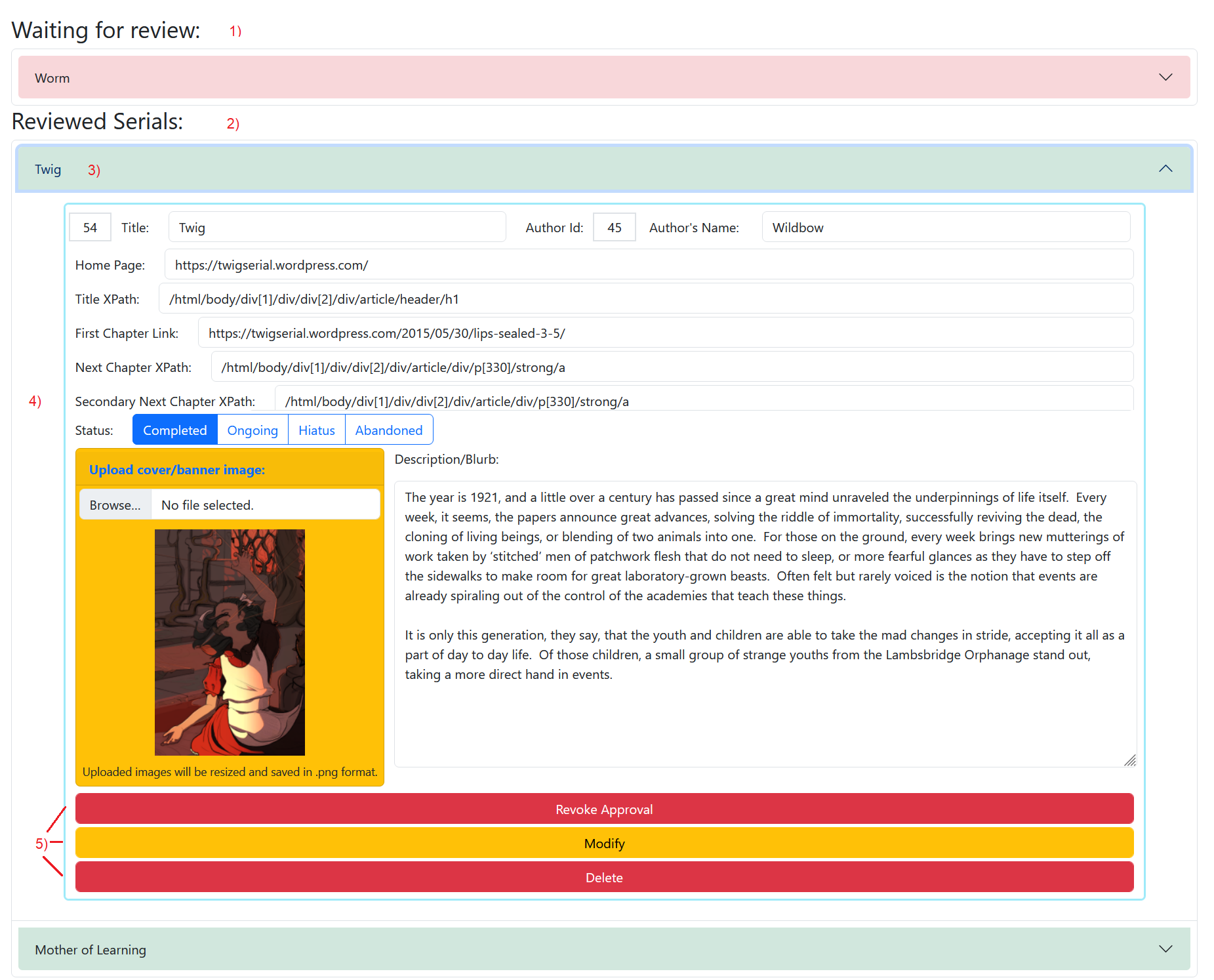


A fenti példán egy már kitöltött, és elküldésre váró példát látunk. Az 1)-el jelölt mezők kitöltése kötelező új sorozat hozzáadásához, amire a 4)-es számú pont alatt látható figyelmeztetés is felhívja a figyelmet. A 2)-es számmal jelölt mezők kitöltése opcionális. Miután a kötelező mezők ki lettek töltve, fel lehet venni a sorozatot az adatbázisba az ‘Add’ gombbal 5). Ilyenkor ha az adatok jól lettek megadva, nem csak a sorozat lesz felvéve az adatbázisba, hanem a sorozat fejezetei is, a webscraping szervíz által.

Propose Serial - Sorozat Javaslása

Ha bejelentkeztünk, de nincsen admin jogosultságunk, akkor is van lehetőségünk javasolni sorozatokat, amiket szeretnénk látni az oldalon. Ezt a “Propose Serial” oldalt akkor tudjuk elérni ha már bejelentkeztünk. Az ábrán az 1)-es számmal jelölt mezők kitöltése kötelező jó sorozat javaslásakor, míg az adatok amiket az ábrán 2)-es számmal jelöltünk megadása opcionális. Ha a minden kötelező mezőt kitöltöttünk, az “Add” gombbra kattintva tudjuk javasolni a sorozatot, és miután ezt egy admin ellenőrizte, és hozzáadta a szükséges adatokat, már láthatjuk is majd a többi sorozat között.

Admin-Serials



Egy csak adminok(‘Admin’, vagy ‘SAdmin’ jogosultság) számára látogatható oldal, ahol a tárolt sorozatok adatait tudjuk módosítani hibás adatok esetén, illetve ki tudjuk egészíteni a felhasználók által javasolt sorozatokat a megfelelő adatokkal.

Itt láthatjuk egy példát arra hogyan nézhet ki az oldal. Két részre van osztva alapvetően.

1) Felül láthatjuk a piros háttér színnel jelölve az admin elismerésre váró sorozatokat. Ide kerülhetnek sorozatok ha egy admin visszavonta a jóváhagyását, illetve a felhasználók által javasol sorozatok automatikusan ide leszek besorolva.

2) Itt láthatjuk zöld háttérrel a már jóváhagyott sorozatokat. Ezek, ellentétben a még jóvá nem hagyottakkal, meg lesznek jelenítve felhasználók számára, és a keresési eredmények között is láthatóak lesznek.

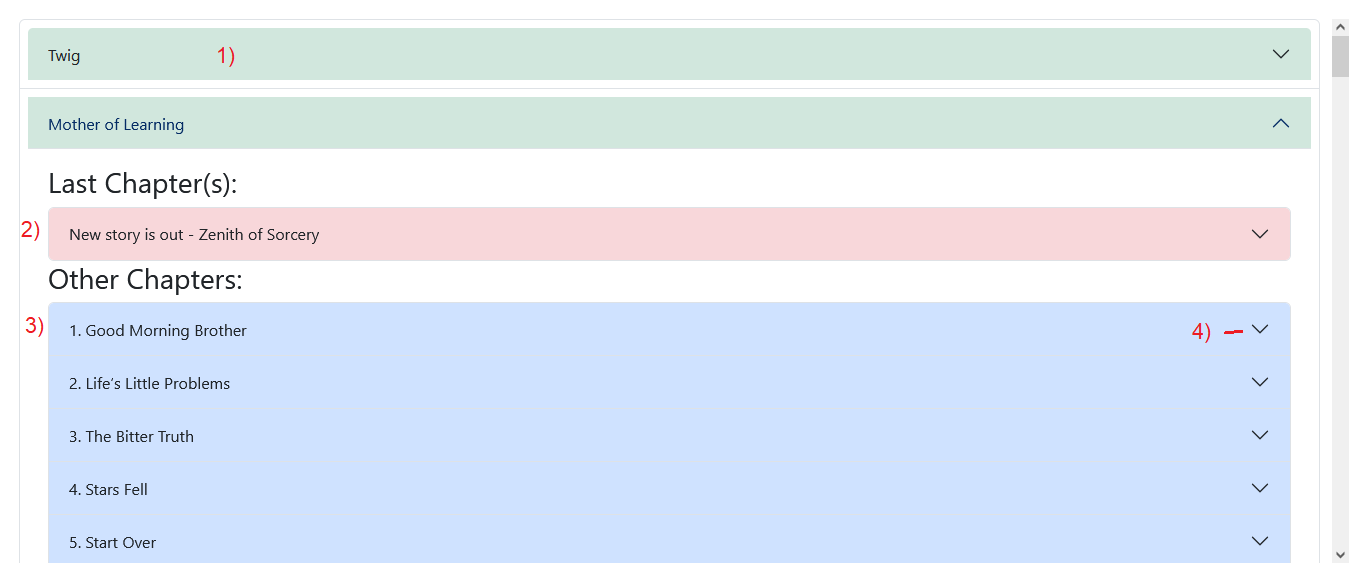
3) Akárhova tartozik a sorozat,a címére kattintva le lehet nyitni a fület, ahol meg tudjuk tekinteni az adatait.

4) Itt láthatjuk a sorozat minden adatát, és meg tudjuk változtatni őket. Ez alól kivétel az “id”, amit nem lehet megváltoztatni.

5) Ez a három gomb a különböző, kizárólag adminok által végrehajtható műveletet jelöli. Először is lehetőségünk van jóváhagyni, vagy visszavonni a jóváhagyást egy sorozatot, (ilyenkor sem a sorozat, sem fejezetei nem fognak megjelenni se a főoldalon, se a keresési eredmények között). Ha egy még jóvá nem hagyott sorozat részleteit néznék meg, ott a ‘Revoke Approval’ gomb helyett egy zöld ‘Approve’ gombot találnánk. Tudjuk módosítani a sorozat adatait a “Modify” gombbal, és ha ezt egy jóváhagyott sorozattal tesszük, akkor annak a fejezetei automatikusan felvevődnek az adatbázisba. Végül pedig lehetőségük van törölni a sorozatot az adatbázisból.

Admin-Chapters

A harmadik csak adminok számára elérhető oldal, itt tudjuk szükség szerint módosítani az adatbázisban eltárolt fejezetek adatait.

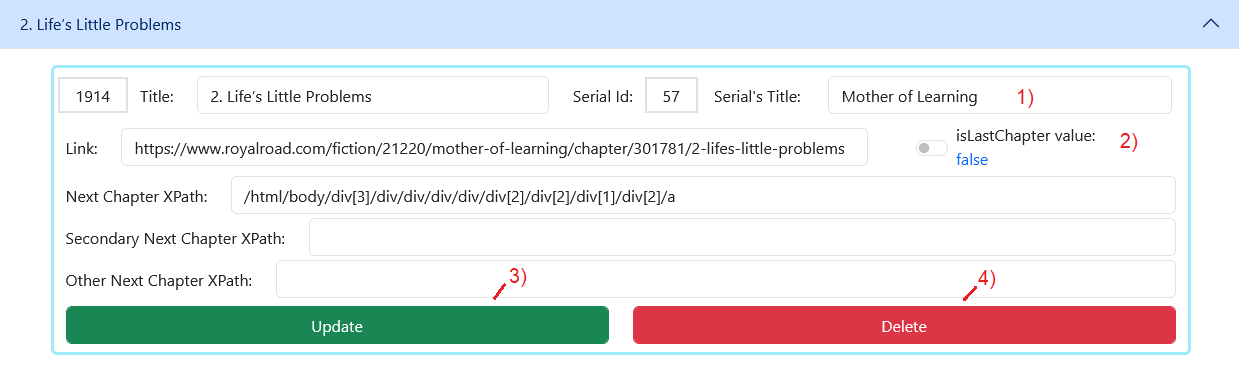
8. ábra

A fenti ábrán láthatjuk, hogy ezen az oldalon a tárolt fejezetek sorozatokra vannak osztva, a könnyebb átláthatóság kedvéért.

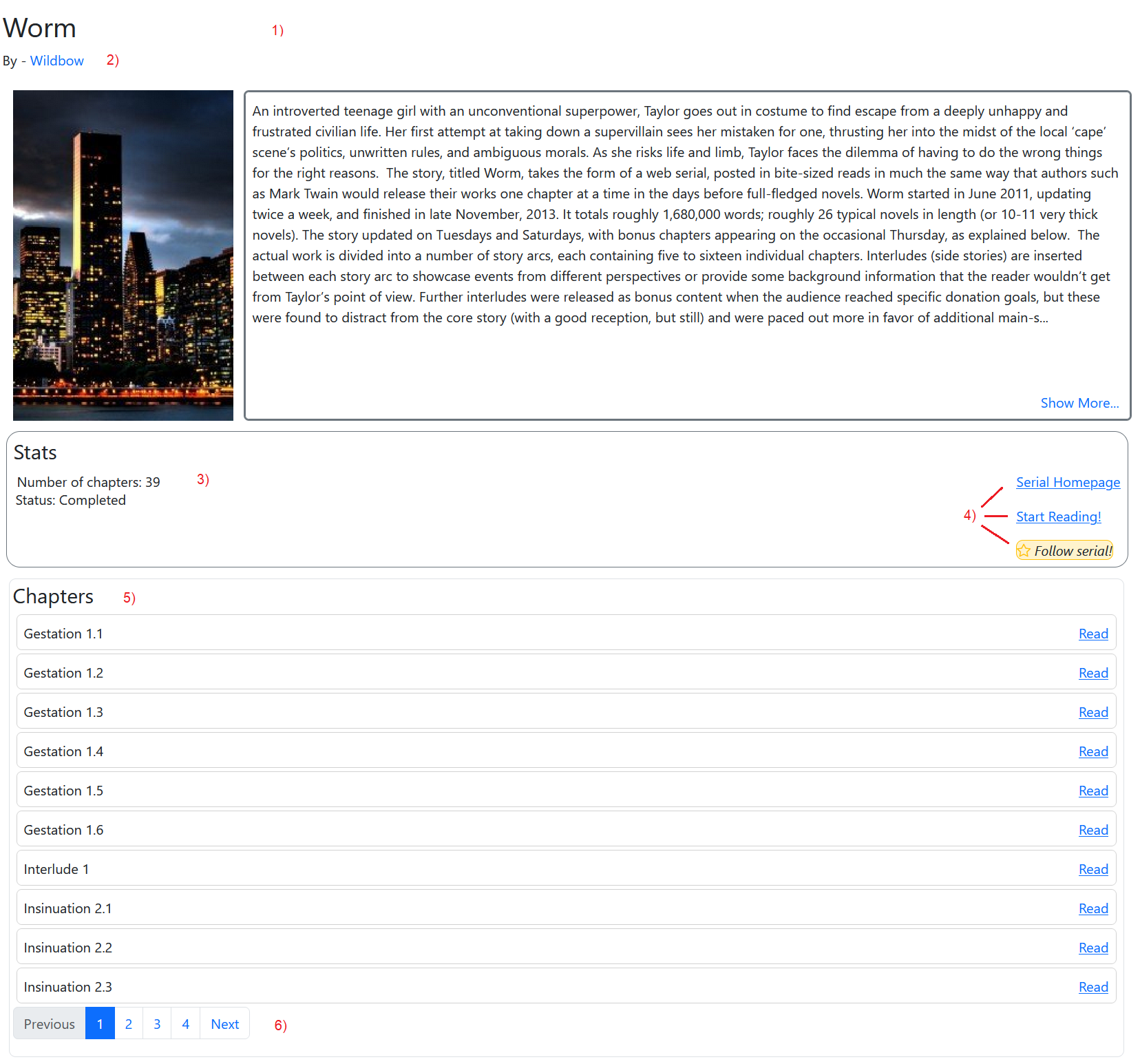
1) A sorozatok címeit látjuk itt zöld háttérrel, amit legyitva megtekinthetjük ennek sorozatnak a fejezeteit.

Ezek két részre vannak bontva, felül a Last Chapter(s) 2) alatt látjuk az utolsó fejezetét a sorozatnak, míg alul sorrendben a többi fejezetet, az “Other Chapters” 3) alatt. Ez a felbontás abban segít, hogy ha valami hiba lépett fel a sorozat fejezeteinek felvételekor, akkor a legutoljára sikeresen felvett fejezet adatainak megváltoztatásával az könnyen javítható lesz.

4) A fejezetek címére kattintva, a leggödülő részben tekinthetjük meg ezeket az adatokat.



Itt látjuk egy fejezet adatait, amiket, az id, a Serial id, és a Serial’s title, kivételével, módosítani is tudunk. A módosításainkat az “Update” 3) gombbal tudjuk menteni, míg a “Delete” 4) gomb segítségével tudjuk törölni a fejezetet.

Serial Details - Sorozat részletek

Több link is vezethet erre az oldalra minket, ahol meg tudjuk nézni egyes sorozatok részletes adatait felhasználóbarát módon.

1) - 2) Látjuk a sorozat címét, és íróját. Az író nevére kattintva megtekinthetjük a az író minden művét.

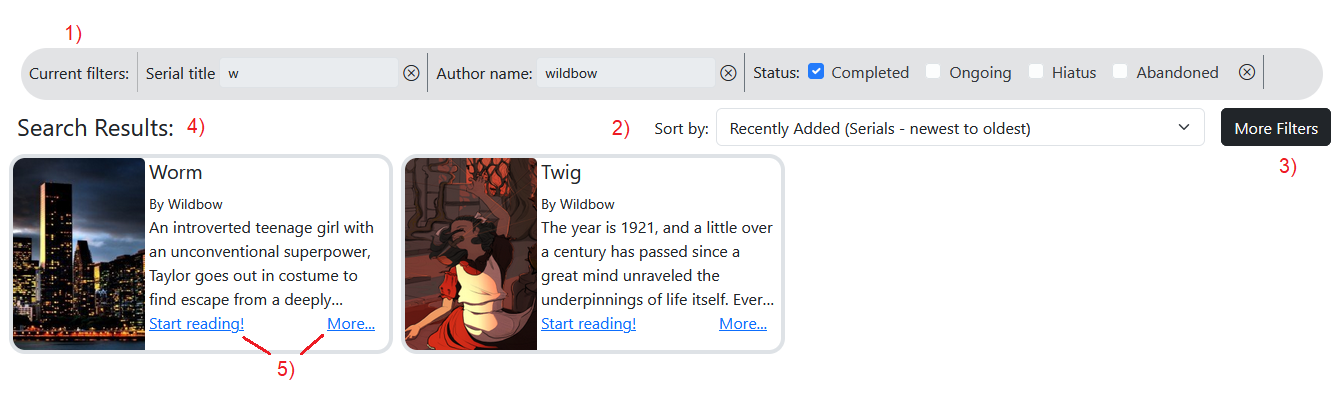
Láthatjuk a sorozat borítóképét, illetve leírását a következő részen, ahol a leírás ha túl hosszú, le lesz rövidítve, amit a “Show More…” gombbal tudunk teljes hosszában elolvasni.

3) Itt van néhány adata a sorozat, hogy hány fejezetből áll, illetve mi a státusza:

* Be van fejezve - “Completed”
* Folyamatban van - “Ongoing”
* Szünetel - “Hiatus”
* Félbehagyott - “Abandoned”

4) Ezekkel a linkekkel a tudja a felhasználó meglátogatni a sorozat kezdőoldalát (“Serial Homepage”), elkezdheti olvasni azonnal az első fejezettől (“Start Reading”), vagy, ha be van jelentkezve hozzáadhatja a követett sorozataihoz.

5) - 6) Végül pedig lehetőségünk van elkezdeni olvasni, a “Read” linkkel, a sorozat fejezeteit, és böngészni közöttük, az oldalszámokra kattintva, vagy a “Next” és “Previous” gombokat használva.

Results - Keresési eredmények

Ezen az oldalon tudunk különböző szűrők segítségével rákeresni sorozatokra.

1) Itt látjuk a jelenlegi filtereket, amik épp használatban vannak. Ezek függőleges vonalakkal vannak elválasztva, és az ‘x’-re kattintva törölhetjük őket.

2) Itt tudjuk állítani, mi szerint vannak az eredmények sorba rendezve. Hat féle sorrend lehet, 3 szempont, és azok fordítottja alapján:

* A sorozatok az adatbázishoz adásának sorrendje, a legutóbbitól a legelsőig, és fordítva
* Új fejezet az adatbázishoz adásának sorrendje, és fordítottja
* Sorozat fejezeteinek száma szerint, növekvő és csökkenő sorrendben

3) Ezzel a gombbal tudunk megjeleníteni egy oldalsó menüt, ahol beállíthatjuk a használni kívánt filtereket.

4) Itt látjuk, a jelenleg használt filterek (“Current filters”) alapján talált keresési eredményeket.

5) Minden megtalált sorozatról megtekinthetünk további részleteket a “More…” linkkel, ami annak a sorozatnak a Serial Details oldalára vezet, vagy a “Start reading!”-re tudunk kattintani, ami sorozat kezdőoldalára vezet.



Ezen az ábrán láthatjuk azt a menüt, amit az előzőleg a 3-as pontban említett “More Filters” gombbal tudunk megjeleníteni. Ha vannak jelenleg használt filterek, azok itt is ki lesznek választva, hogy módosítani tudjuk őket.

1) Ide beírva tudunk a sorozatok között keresni cím alapján.

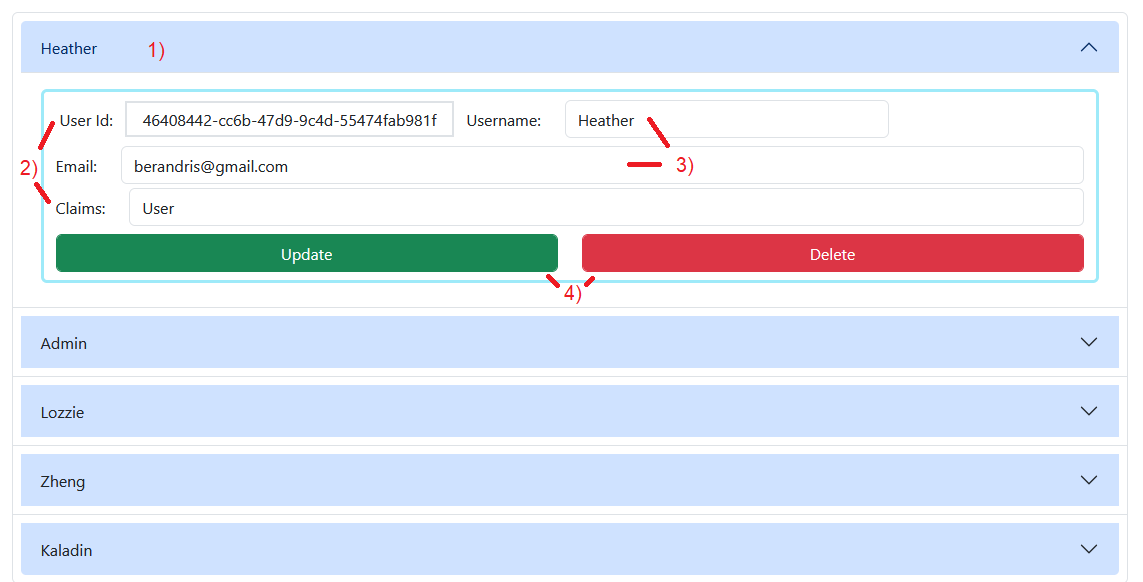
2) Itt tudunk írók neve alapján keresni.

3) Ha fejezet számok alapján szeretnénk keresni, itt tudjuk megtenni, akár egyedi intervallumot is választhatunk, a “Custom” opció kiválasztásával.

4) Itt a 4 lehetséges sorozat státusz alapján szűrhetünk, ezek közül akármennyit bejelölhetünk.

5) Ha be vagyunk jelentkezve, az is opció, hogy csak a követett sorozatok között keressünk.

6) Az “Apply Filters” gombbal indíthatjuk a keresést a beállított filterek alapján, míg a “Reset Filters” töröl minden eddig beállított filtert.

Manage Users - Felhasználók kezelése

Szintén egy csak adminként elérhető oldal. Ezen a felületen tudjuk kezelni a felhasználókat és néhány adatukat.

1) Felhasználók nevére kattintva, a legördülő részen láthatjuk részletesebb adataikat.

2) Ezeket az adatokat nem tudjuk módosítani

3) A felhasználónevet és az email címet az admin, ha szükséges meg tudja változtatni.

4) Az “Update” gombbal a változtatásokat tudjuk menteni, míg a “Delete” gomb törli a felhasználót.

**Backend**

A projekt backend részét C#-ban készítettük el, és az Entity Framework segítségével interaktáltunk az adatbázissal.

Ebben a részben végig fogok menni az API végpontjain, csoportosítva aszerint, hogy melyik controller fileban vannak definiálva, és leírom a funkciójukat.

Chapters

A ChaptersController fájl végpontjain keresztül tudjuk módosítani a fejezeteket tartalmazó adatbázis tábla adatait. A fejezetek lekérésére nincs itt végpont, mivel az a sorozatokkal együtt történik.

PUT - /Chapters/{id}

Itt tudjuk módosítani egy fejezet adatait.

Felhatalmazás: Admin - SAdmin

Paraméterek:

* Query - id - Módosítani kívánt fejezet id-ja
* Body - valid Chapter objektum

Válaszok:

* 204 No Content - sikeres módosítás
* 400 Bad Request - a paraméter id nem egyezik a bodyban megadott chapter id-jával
* 404 Not Found - nem létezik fejezet ezzel az id-val

DELETE - /Chapters/{id}

Töröl egy fejezetet

Felhatalmazás: Admin - SAdmin

Paraméterek:

* Query - id - Törölni kívánt fejezet id-ja

Válaszok:

* 204 No Content - sikeres törlés
* 404 Not Found - nem létezik fejezet ezzel az id-val

PATCH - /Chapters/{id}/approve

Itt tudunk jóváhagyni egy fejezetet, vagy visszavonni a jóváhagyását.

Felhatalmazás: Admin - SAdmin

Paraméterek:

* Query - id - jóváhagyni kívánt fejezet id-ja
* Body - valid Review objektum

Válaszok:

* 202 Accepted - sikeres jóváhagyás/visszavonás
* 404 Not Found - nem létezik fejezet ezzel az id-val

Serials

A Serials táblához való hozzáférést a SerialsController fájlban meghatározott végpontokon keresztül biztosítja a backend.

GET - /Serials

Minden tárolt sorozat adatainak lekérése

Válaszok:

* 200 Ok - sikeres lekérés - bodyban SerialDTO array
* 404 Not Found - nem létezik a Serials tábla

GET - /Serials/{id}

Egy sorozat adatainak lekérése

Paraméterek:

* Query - id - sorozat id-ja

Válaszok:

* 200 Ok - sikeres lekérés - bodyban SerialDTO objektum
* 404 Not Found - nem létezik sorozat ezzel az id-val

GET - /Serials/update

Minden sorozat frissítése, új fejezetek lekérdezése

Válaszok:

* 200 Ok - bodyban hozzáadott fejezetek száma
* 404 Not Found - nem létezik a Serials tábla

GET - /Serials/update/{id}

Egy bizonyos sorozat frissítése

Paraméterek:

* Query - id - frissítendő sorozat id-ja

Válaszok:

* 200 Ok - bodyban hozzáadott fejezetek száma
* 404 Not Found - nem létezik sorozat ezzel az id-val

PUT - /Serials/{id}

Egy sorozat adatainak módosítása

Felhatalmazás: Admin - SAdmin

Paraméterek:

* Query - id - sorozat id-ja
* Body - from-data - valid CompleteSerial objektum

Válaszok:

* 204 No Content - sikeres módosítás
* 404 Not Found - nem létezik sorozat ezzel az id-val

POST - /Serials/addSerial

Új sorozat és sorozat fejezeteinek hozzáadása

Felhatalmazás: Admin - SAdmin

Paraméterek:

* Body - from-data - valid CompleteSerial objektum

Válaszok:

* 200 Ok - sikeres hozzáadás - body-ban sorozat címe, és hozzáadott fejezetek száma
* 404 Not Found - nem létezik sorozat ezzel az id-val
* 417 Expectation Failed - már létezik sorozat ezzel a névvel

POST - /Serials/proposeSerial

Új sorozat javaslása

Felhatalmazás: Bejelentkezett felhasználó

Paraméterek:

* Body - from-data - valid PropsedSerial objektum

Válaszok:

* 200 Ok - sikeres hozzáadás
* 404 Not Found - nem létezik sorozat ezzel az id-val
* 417 Expectation Failed - már létezik sorozat ezzel a névvel

DELETE - /Serials/{id}

Sorozat törlése

Felhatalmazás: Admin - SAdmin

Paraméterek:

* Query - id - törlendő sorozat id-ja

Válaszok:

* 204 No Content - sikeres törlés
* 404 Not Found - nem létezik sorozat ezzel az id-val, vagy nem létezik a Serials tábla

PATCH - /Serials/{id}/approve

Itt tudunk jóváhagyni egy sorozatot, vagy visszavonni a jóváhagyását.

Felhatalmazás: Admin - SAdmin

Paraméterek:

* Query - id - jóváhagyni kívánt sorozat id-ja
* Body - valid Review objektum

Válaszok:

* 202 Accepted - sikeres jóváhagyás/visszavonás
* 404 Not Found - nem létezik sorozat ezzel az id-val

POST - /Serials/like

Sorozat hozzáadása a felhasználó követett sorozatokhoz, vagy törléslése közülük.

Felhatalmazás: Bejelentkezett felhasználó

Paraméterek:

* Body - valid LikedSerial objektum

Válaszok:

* 202 Accepted - sikeres követés / követés törlése
* 404 Not Found - nem létezik sorozat, vagy felhasználó a LikedSerial objektum-beli id-val

GET - /Serials/images/{serialTitle}

Egy sorozat borítóképének lekérése - a böngésző beépített caching módszerének használatával.

Paraméterek:

* Query - serialTitle - sorozat címe

Válaszok:

* 200 Ok - létezik a kép - bodyban a sorozat borítóképe, megfelelő content-type-al
* 304 Not Modified - legutóbbi lekérdezés óta nem változott a kép
* 404 Not Found - nem létezik kép ehhez a sorozathoz

Mikor sorozat módosítva lesz, a program ellenőrzi, hogy ha az író neve megváltozott, létezik-e az új névvel író az adatbázisba, és ha nem hozzáadja, illetve, a régi írónak van-e sorozata az adatbázisban, és ha nincs, törli az írót.

Hasonlóan, sorozat törlésekor is ellenőrizzük, hogy az írónak van-e másik sorozata az adatbázisban.

Users

A felhasználók adatainak elérését a UserController, fájlban leírt végpontok teszik lehetővé. Ez nem foglalja magába a bejelentkezést és a regisztrációt, amit egy külön AuthenticationController kezel.

GET - /api/user/userlist

Minden felhasználó adatának lekérése

Felhatalmazás: Admin - SAdmin

Válaszok:

* 200 Ok - sikeres lekérés - bodyban UserWithClaims objektumok listája

DELETE - /user/{id}

Felhasználó törlése

Felhatalmazás: SAdmin

Paraméterek:

* Query - id - törlendő felhasnzáló id-ja

Válaszok:

* 200 Ok - sikeres törlés
* 400 Bad Request - nem létezik felhasználó ezzel az id-val, vagy sikertelen a törlés

PUT - /api/user/{id}

Egy felhasználó adatainak módosítása

Felhatalmazás: Admin - SAdmin

Paraméterek:

* Query - id - felhasnzáló id-ja
* Body - valid UpdateModel objektum

Válaszok:

* 200 Ok - sikeres törlés
* 400 Bad Request - nem létezik felhasználó ezzel az id-val, vagy nincs felhatalmazása, vagy invalid az UpdateModel
* 500 Internal Server Error - sikertelen törlés

HEAD - /api/user/password/resetReq/{email}

Elfelejtett jelszó esetén, új jelszó beállításához, email kérése.

Felhatalmazás: Bejelentkezett felhasználó

Paraméterek:

* Query - email - felhasználó email címe

Válaszok:

* 200 Ok - email sikeresen elküldve
* 500 Internal Server Error - felhasználó nem létezik, vagy email szervíznek nem sikerült elküldenie az emailt

POST - /api/user/password/reset/{id}

Emailben kapott token segítségével új jelszó beállítása.

Felhatalmazás: Bejelentkezett felhasználó

Paraméterek:

* Query - id - felhasználó id-ja

Válaszok:

* 200 Ok - jelszó sikeresen módosítva
* 400 Internal Server Error - felhasználó nem létezik, vagy nem sikerült jelszót módosítani

PUT - /api/user/changePassword

Felhasználó saját jelszavának módosítása

Felhatalmazás: Bejelentkezett felhasználó

Paraméterek:

* Body - valid ChangePasswordModel objektum

Válaszok:

* 200 Ok - sikeres jelszó módosítás
* 400 Bad Request - invalid ChangePasswordModel, vagy nincs jogosultsága, vagy sikertelen jelszó módosítás
* 500 Internal Server Error - sikertelen jelszó módosítás

Authorization

POST - /api/Authentication/login

Bejelentkezés felhasználónévvel és jelszóval

Paraméterek:

* Body - valid LoginModel objektum

Válaszok:

* 200 Ok - bejelentkezés - bodyban LoggedUser - tartalmazza követett sorozatokat, és a felhasználó szerepeit
* 400 Bad Request - hibás jelszó vagy felhasználónév, vagy invalid LoginModel
* 500 Internal Server Error - ismeretlen hiba

POST - /api/Authentication/register

Regisztráció új felhasználóként

Paraméterek:

* Body - valid RegistrationModel objektum

Válaszok:

* 201 Created - sikeres regisztráció- bodyban új felhasználó neve
* 400 Bad Request - felhasználónév vagy email foglalt, sikertelen regisztráció, vagy invalid RegistrationModel
* 500 Internal Server Error - ismeretlen hiba

Scraping Service

Ez az a szervíz, ami a fejezetek automatikus adatbázishoz adásáért felelős, és az egyik legfontosabb része a backendnek. Cél volt, hogy minnél több felmerülő problémával automatikusan meg tudjon bírkózni, hogy, minél robosztusabb legyen. A tesztelés során legtöbbet felmerült akadályokat sikerült kiküszöbölni, de még sokat lehetne fejleszteni rajta.

Nagyrészt egy metódus hajtja végre a web scraping-et, az AddSerialChapters. Külön metódus csak a fejezetek adatbázishoz adására, illetve az egyik oldalról a következőre való sikeres navigálás biztosítására van. Az utóbbi, a CheckUrl, biztosítja, hogy a következő fejezet linkjére való navigáláskor felmerülő problémákat kezelni tudjuk. A két legrelevánsabb probléma mibe ütköztünk, az oldal elköltözése, illetve a rate-limiting voltak. Az első, az elköltözés probléma megoldása viszonylag egyszerű volt, hiszen ilyenkor a HTTP response tartalmaz egy “Location” header-t, aminek segítségével folytathatjuk a folyamatot. A második probléma ritkán jelentkezett, de ez lehet a teszteléskor használt hardwer limitációja miatt volt csak, úgyhogy írtunk rá egy megoldást. Ha a navigáció közben egy 429 Too Many REquests választ kaptunk, ellenőriztük, hogy a válasz tartalmaz-e “Retry-After” header-t, amit aztán hasnzáltunk, ha igen. Ha nem találtunk ilyen header-t, vártunk egy fokozatosan növekvő mértékű időt, mielőtt újra próbálkoztunk. Ez megoldotta a problémát, bár lehet hogy egy valós környezetben egy ennél robosztusabb megoldásra lenne szükség.

A fő metódus, az AddSerialChapters, több különböző módszerrel próbálja biztosítani, hogy ne akadjon meg a fejezetek beolvasása. Egy adott sorozat objektumnak több mezője is van, ahol a következő fejezet linkjének a helyét XPath-ban kifejezve tároljuk. Ezeket természetesen a metódus mind ellenőrzi, ha egyik helyen sem talál valid linket. Ez jelentheti azt, hogy a megadott helyen már nincs semmi, de akár azt is, hogy ott egy olyan link van, ami az előző 3 felvett fejezet valamelyikének linkjével azonos. Ez tesztelés során többször előfordult, hogy az oldal úgy változott meg, hogy attól a ponttól kezdve, ahol eddig a következő fejezet linkje volt, most az előzőé lett. Ha minden a sorozat objektumban megadott lehetséges XPath-ot ellenőrzött, és még mindig nem talált megfelelő linket, a módszer amit leghatékonyabbnak találtunk, a legutoljára működő link szövegének tárolása, és az oldal szövegén belül való keresése volt, vagy, azt tapasztaltuk, hogy a következő fejezetre mutató linknek gyakran volt “next” értékű “rel” attribútuma, úgyhogy ha szöveg alapján sem sikerült, akkor a metódus megpróbálja az oldalon található linkek között ez alapján megtalálni a következő fejezetre mutatót. Ha az utóbbi két fajta keresés is sikertelen, akkor feltétlezzük, hogy annyira megváltozott az oldal, hogy emberi beavatkozásra van szükség, vagy ez volt a sorozat utolsó fejezete. Mindkét esetben a webscraping befejeződik, és a hozzáadott fejezetek mentésre kerülnek. Ha vagy a szöveg alapú, vagy a “rel” attribútumra való keresés sikeres, míg az elmentett XPath-ok nem voltak azok, a webscraping befejezésekor a sorozathoz hozzáadjuk az így talált linkek XPath-ját.

Ez a metódus nem tökéletes, de lévén hogy potenciálisan több százszor le kell futni egy sorozat hozzáadásakor, igyekeztünk a teljesítmény optimalizálásra is törekedni.