Szoftverfejlesztő és -tesztelő technikus záróvizsga – gyakorlati vizsgatevékenység „B”:

Asztali- és webes szoftverfejlesztés, adatbázis-kezelés feladatsor

rendelkezésre álló idő: 240 perc

**B/1/1: konzolos asztali alkalmazásfejlesztés C# nyelven (15 pont)**

Adott a **bestgames.csv** UTF-8-as kódolású, pontosvesszővel tagolt állomány, mely tartalmazza a kritikusok által legjobbnak ítélt videojátékok listáját 2017-ig.

A file egy sora például:

2010;StarCraft II: Wings of Liberty;Real-time strategy;Blizzard Entertainment;PC, Macintosh;

Ahol az adattagok jelentése a következő:

**2010** a megjelenés éve

**StarCraft II** Wings of Liberty: a játék címe

**Real-time strategy** a játék műfaja

**Blizzard Entertainment** a játék kiadója

**PC, Macintosh** azon platformok listája (vesszővel elválasztva), amikre a játék először megjelent

* hozzon létre osztályt, mely képes reprezentálni egy játékot! Az osztály konstruktora a file egy sorát kapja bemeneti paramétként!
* olvassa be a file tartalmát egy olyan összetett adatszerkezetbe, melynek segítségével meg tudja válaszolni a további kérdéseket!
* határozza meg, és írja ki a konzolra, hogy hány játék szerepe a listában!
* határozza meg, hogy mely években hány játék került fel az örökranglistára! Listázza ki azokat az évszámokat – és az abban az évben megjelent listán lévő játékok számát, amely években ez a szám több, mint 10. A felsorolás legyen a játékok száma szerinti csökkenő sorrendben!
* határozza meg, és írja ki a képernyőre, hogy hány db „First-person shooter” műfajú játék került fel az örökranglistára. A program véletlenszerűen sorsoljon egy játékot ezek közül, és írja ki ennek a címét a képernyőre!

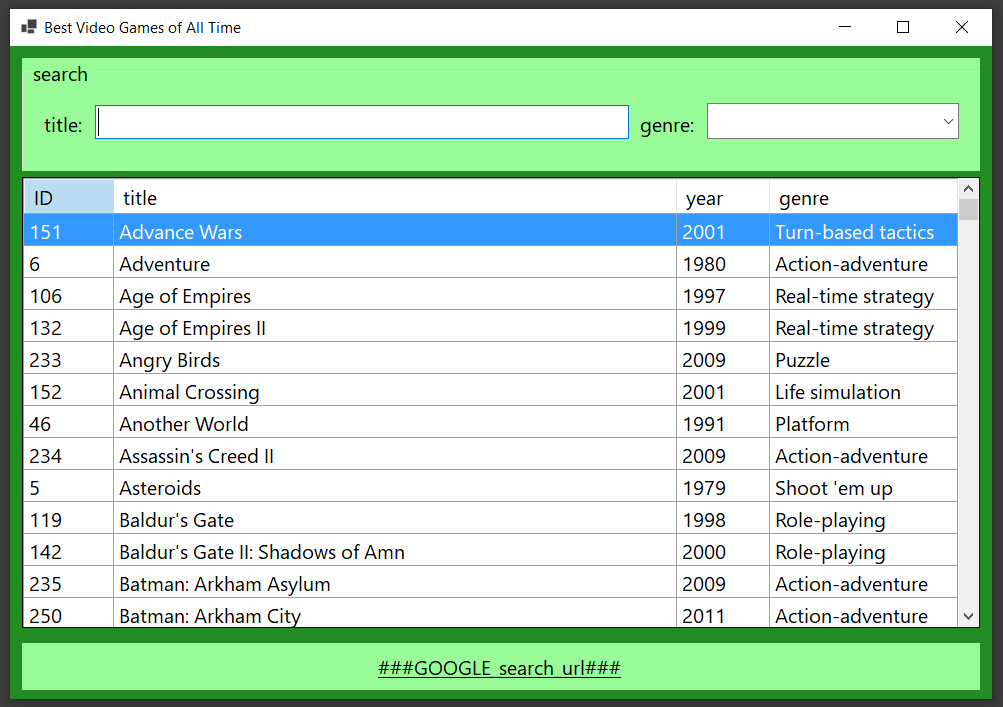
minta:

A képen szöveg látható

Automatikusan generált leírás

**B/1/2: grafikus asztali alkalmazásfejlesztés C# nyelven (10 pont)**

Hozzon létre grafikus alkalmazást az alábbi minta alapján!



Hozzon létre új adatbázist a rendelkezésére álló lokális adatbázisszerveren bestgames néven, a források közt található **bestgames\_database.sql** dump felhasználásával!

* Az ablakban látható datagrid sorait ebből az adatbázisból töltse fel!
* A ’search’ szekció ’genre:’ címkéje mellett egy legördülő lista található. A program betöltésekor, az ablak megjelenését megelőzően populálja ezen listát a genre táblában szereplő összes műfaj megnevezésével!
* Valósítsa meg azt az alkalmazásfunkciót, hogyha a ’title:’ melletti beviteli mezőbe gépel a felhasználó – vagy szűrést hajt végre a legördülő lista segítségével a műfajokra, akkor a gridben csak a feltételeknek megfelelő címek jelenjenek meg!
* Ha rákattintunk a grid egy sorára, generáljon le egy linket az ablak alján lévő szekcióba, amire ha rákattintunk megnyílik a rendszer alapértelmezett böngészője, és a google keresőmotorja segítségével keressen rá az interneten arra a játékra, melynek a címe a grid kattintásnak megfelelő sorában található. A google search query statikus részét képző substring-et a **gs\_help.txt** fileban megtalálja!

A képen szöveg látható

Automatikusan generált leírás

Klónozza le a template\_url.txt-ben található repository-t a használt IDE segítségével! elfogadhat minden tanúsítványt. Letöltés után készüljön el az első build, és győződjön meg róla, hogy hogy a template az elvárt módon jelenik meg!

A projekt tartalma egy előre konfigurált ASP.NET Framework 4.8 weboldal, melynek ’home’ aloldala tartalmazza az alapértelmezett template-en elvégzett változásokat, és az integrált NuGet-ek verziószámát. A további feladatokat ezen template módosításával valósítsa meg!

Mivel a template tartalmazza a szükséges információkat, ennek eredeti repozitory-ját nyugodtan használhatja a vizsga végéig!

A képen szöveg, képernyőkép, computer, képernyő látható

Automatikusan generált leírás

**B/2/1: reszponzív viselkedésű weboldal készítés és formázás (10 pont)**

A projekt **~/Views/Shared/\_Layout.cshtml** állományában végezze el a következő változtatásokat:

* a böngésző lapfülén megjelenő szöveg legyen a „Videójátékok örökranglista” szöveg
* az oldal láblécében cserélje le a copyright szekciót a saját nevére, valamint a „gyakorlati vizsgatevékenység B/2” szövegre.

A project **~/Content/Site.css** állományában végezzel el a következő módosításokat! Ha nem változtat az osztályneveken, előfordulhat, hogy a böngésző gyorsítótárát „hard-refresh” módban kell frissítenie, hogy a változások build után megjelenjenek (Firefoxon CTRL+F5)

változtassa meg a navigációs menü stílusát az alábbi irányelvek alapján:

* háttérszíne legyen crimson (#**DC143C**), alapértelmezett link- és szövegszíne legyen cornsilk (#**FFF8DC**). A kurzor alatti menüpontok színe legyen darkkhaki (#**BDB76B**).
* adja hozzá a projekt megfelelő mappájához a **controller.png** forrásállományt, és cserélje le a navigációs menü ikonját erre! Változtassa meg a navigációs menü feliratát és listaelemeinek szövegét a minta szerint!

Változtassa meg az **~/Views/Home/Index.cshtml** file tartalmát az alábbiak szerint! Az új stílus-definíciókat a **~/Content/Site.css** állományba vezesse fel!

* a főoldal tartalmát cserélje ki a **fooldal\_forras.txt** állományban található szövegre!
* alakítson ki 3:1 arányú rácselosztást a megfelelő bootstrap osztálykijelölők használatával a minta alapján! a két oszlop tartalmát közepes képernyőméret alatt automatikusan tördelje egymás alá!
* a bal oldali blokkban jelölje ki a minta szerint a paragrafusokat! definiáljon egy css osztályt, melynek alkalmazásával minden második paragrafus háttérszínét állítsa lightpink (#**FFB6C1**) színűre!
* a jobb oldali blokkban alakítson ki több szintes, számozatlan felsorolást a minta alapján! lássa el a bal oldali blokkot 5 pixel vastag, folytonos crimson ((#**DC143C**) színű kerettel!

A képen szöveg látható

Automatikusan generált leírás

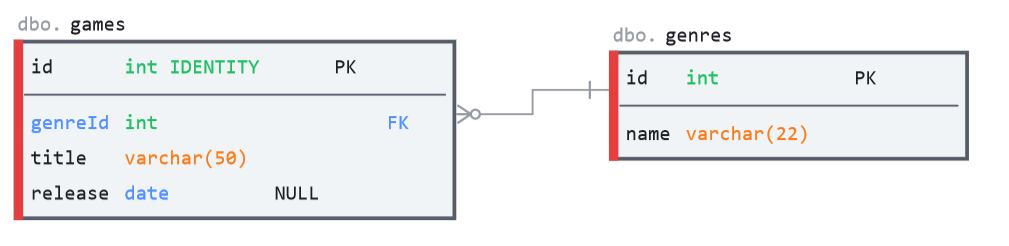
**B/2/2: „backend” programozás (15 pont)**

Engedélyezze az adatbázis migrációkat a projekthez!

Valósítsa meg, hogy legyen a weboldalon emailcím + jelszó alapú autentikáció! Inicializálja az ehhez szükséges adattáblák migrációját, és frissítse az adatbázist!

Regisztráljon **user@bestgames.vrbly** címen, **MgLKa\_nx2NZs** jelszóval tesztfelhasználót a rendszerbe!

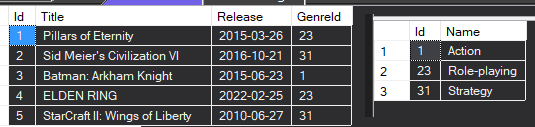
Hozzon létre a megfelelő mappába POCO modelleket. Lássa el a megfelelő annotációkkal a propertyket ahhoz, hogy az alábbi ER diagramnak megfeleltethető adatbázist tudjon belőlük származtatni code-first approach módon!



Külön migrációban generáljon adattáblákat az imént készített modellekből!

Hozzon létre migrációt az adatbázis példaadatokkal való feltöltéséhet! Táblánként 5-6 rekord elegendő. Használhatja az asztali alkalmazáshoz tartozó forrásfile-ok tartalmát, de vegye figyelembe, hogy itt dátum típussal dolgozunk! A mintaadatoknak nem kell a valóságnak megfelelniük!

néhány példa, amivel a minták készültek:

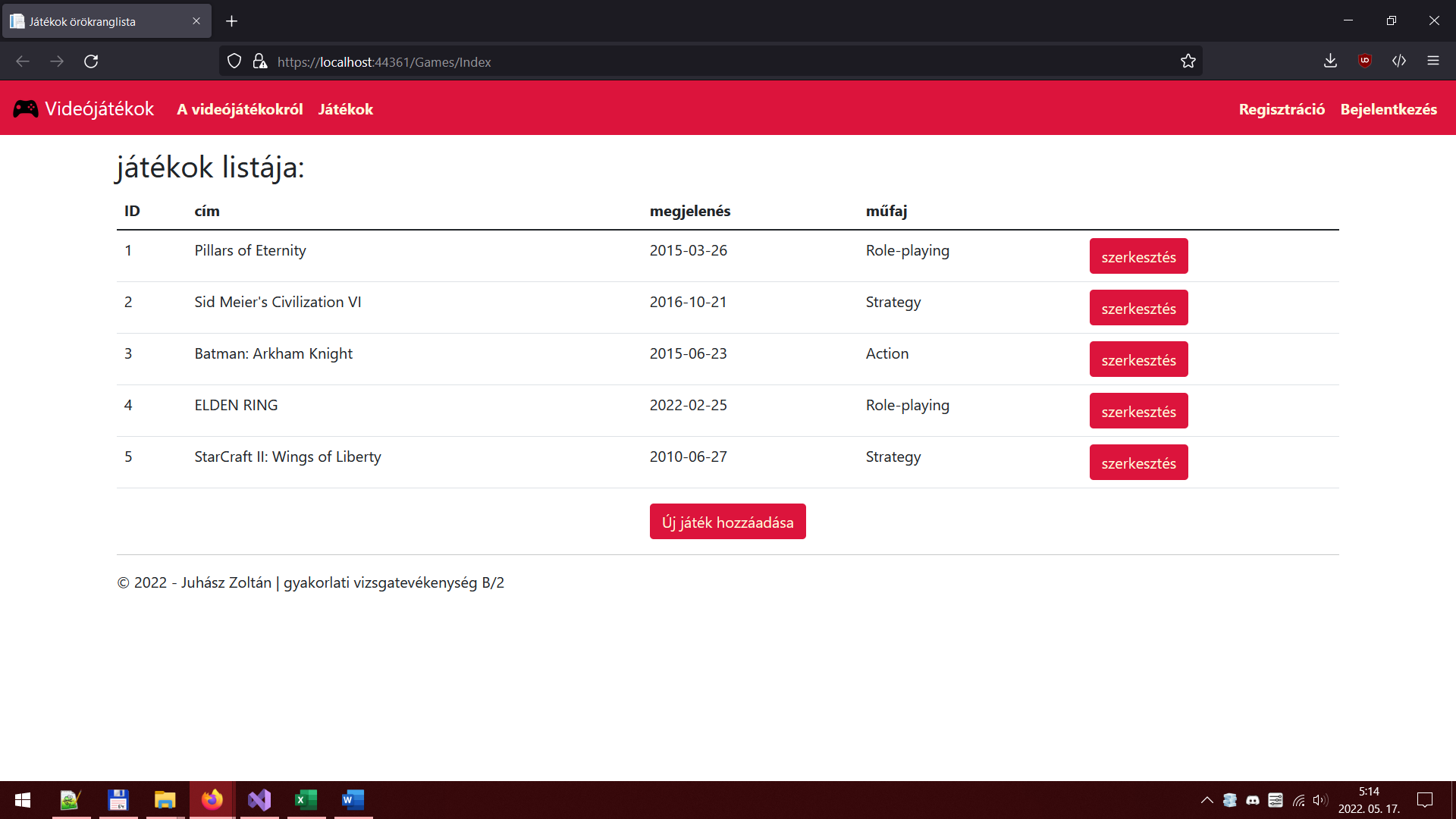


**B/2/3: „frontend” programozás (15 pont)**

Hozzon létre új controllert Games néven, melyben valósítsa meg a következő funkciókat:

**~/Games/(Index)** url-en (**http GET**):

Jelenítsen meg egy táblázatot, amiben megjelenik az összes játék összes releváns adata az adatbázisból a minta szerint!

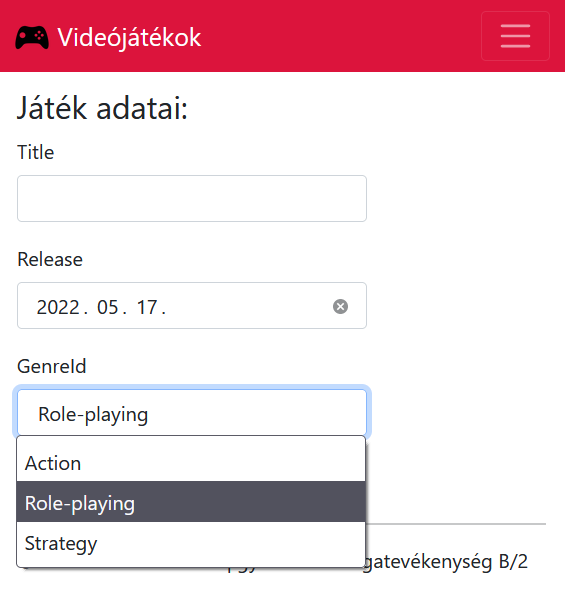


Hozzon létre új view-t GameForm néven, ami egy játék adatainak html form control-okban való megjelenítésére alkalmas!

Ha az index oldal táblázatának utolsó cellájában lévő linkre kattintunk, töltse be a GameForm view-t az **~Games/Edit/{id}** url-en úgy, hogy lehetőségünk legyen az adott játék adatainak szerkesztésére!



A **~/Games/New** url-en töltse be a GamesForm view-t úgy, hogy az legyen alkalmas egy új játék hozzáadására az adatbázishoz!



Akár meglévő rekord szerkesztésének kísérlete, akár új játék felvitele esetén, ha a form-on üresen hagyjuk a ’title’ adattagra vonatkozó mezőt, jelenjen meg hibaüzenet az oldal újratöltése nélkül!



Ha a form alján a mentés gombra kattintunk, módosítsa a megfelelő id-val jegyzett rekordot - illetve vigyen fel új rekordot az adatbázisba, és ha ez sikeresen megtörtént, költse újra az index oldalt!