



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

**NEUMANN JÁNOS
INFORMATIKAI KAR**



SZAKDOLGOZAT

**OE-NIK
2022**

Hallgató neve:
Hallgató törzskönyvi száma:

**Hegedűs Máté
T/007029/FI12904/N**

SZAKDOLGOZAT FELADATLAP

Hallgató neve: **Hegedűs Máté**
Törzskönyvi száma: T/007029/FI12904/N
Neptun kódja: XXXXXXXXXX

A dolgozat címe:

Reszponzív felhasználóbarát webalkalmazás tervezése és fejlesztése
Design and development of a responsive user-friendly web application

Intézményi konzulens: Sipos Miklós László, Szabó-Gali Ákos
Külső konzulens:

Beadási határidő: 2022. május 15.

A záróvizsga tárgyai: Szoftvertechnológia és grafikus felhasználói interfész
tervezése
Szoftvertervezés és -fejlesztés specializáció

A feladat

A feladat egy olyan modern felhasználóbarát webes alkalmazás készítése, amely lehetővé teszi a videójátékokról szóló fontos információk beszerzését. Vizsgálja meg, hogy mely technológiák szükségesek egy webes rendszer készítéséhez, elemezze azokat pozitív és negatív oldalról. Az adott technológia melletti döntését támassza alá. A webes rendszer mellett tervezzen egy automatizált megoldást, amely bejelentkezést követően képes a felhasználó fiókjához rendelni a számítógépének hardver komponenseit. Mai modern technológiák (pl. web scraping) segítségével gyűjtse össze a mai, korszerű hardvereket, majd biztosítson egy felületet, ahol a felhasználó ellenőrizni tudja, hogy a számítógépe képes-e a játékszoftver futtatására. A webes kommunikáció kialakításához használjon korszerű REST hívásokat. Fontos, hogy a weboldal funkciói csak beregisztrált felhasználók által legyenek elérhetőek.

A dolgozatnak tartalmaznia kell:

- a megoldandó feladat pontos leírását és részletes elemzését,
- magas szintű szerkezeti mintát az alkalmazásról,
- az egyes funkciók megvalósításához felhasználható lehetséges technológiákat,
- az elkészített rendszer tervét, blokkdiagramját,
- a megvalósítás menetét, felmerülő problémákat és az azokra adott megoldást,
- a tesztelési tervet, a tesztelés módszertanát és a tesztelés eredményeit,
- az elkészült szoftver felhasználási lehetőségeit.



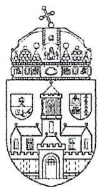

.....
Dr. Vámosy Zoltán
intézetigazgató

A szakdolgozat elévülésének határideje: **2024. május 15.**
(OE TVSz 55.§ szerint)

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens


.....
intézményi konzulens



HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 2022.05.13.


.....
hallgató aláírása



KONZULTÁCIÓS NAPLÓ

Hallgató neve:

Neptun kód:

Tagozat:

Hegedűs Máté



Nappali

Telefon:

Levelezési cím (pl: lakcím):



Szakdolgozat / Diplomamunka¹ címe magyarul:

Reszponzív felhasználóbarát web alkalmazás tervezése és fejlesztése

Szakdolgozat / Diplomamunka² címe angolul:

Design and development of a responsive user-friendly web application

Intézményi konzulens:

Külső konzulens:

Szabó-Gali Ákos

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2021.09.15	Szakdolgozattal kapcsolatban általános tudnivalók	
2.	2021.09.23	Szakdolgozat téma, cím egyeztetése	
3.	2021.10.25	Szakirodalom kutatás áttekintése	
4.	2021.12.07	Rendszerterv áttekintése, tájékoztatás a leadásról, védés menetéről.	

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplo-mamunka 2 / Diplomamunka 3 / Diplomamunka 4³ tantárgy követelményét teljesítette, beszámolóra / védésre⁴ bocsátható.

Budapest, 20 21.12.08.....

.....
intézményi konzulens

¹ Megfelelő aláhúzendő!

² Megfelelő aláhúzendő!

³ Megfelelő aláhúzendő!

⁴ Megfelelő aláhúzendő!



KONZULTÁCIÓS NAPLÓ

Hallgató neve:

Neptun kód:

Tagozat:

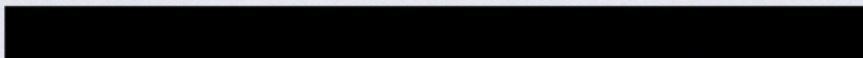
Hegedűs Máté



Nappali

Telefon:

Levelezési cím (pl: lakcím):



Szakdolgozat / Diplomamunka¹ címe magyarul:

Reszponzív felhasználóbarát web alkalmazás tervezése és fejlesztése

Szakdolgozat / Diplomamunka² címe angolul:

Design and development of a responsive user-friendly web application

Intézményi konzulens:

Külső konzulens:

Szabó-Gali Ákos

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2022.02.20	Egyeztetés a szakdolgozat további menetéről	
2.	2022.03.14	Fejlesztési feladatok áttekintése, egyeztetése	
3.	2022.04.01	Eddig elkészült munka áttekintése	
4.	2022.05.08	Szakdolgozat áttekintése, tájékoztatás a leadásról	

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4³ tantárgy követelményét teljesítette, beszámolóra / védésre⁴ bocsátható.

Budapest, 20 22.05.08

.....
intézményi konzulens

¹ Megfelelő aláhúzendő!

² Megfelelő aláhúzendő!

³ Megfelelő aláhúzendő!

⁴ Megfelelő aláhúzendő!

Absztrakt

Az informatika a világon az egyik legjobban fejlődő iparág. Évről évre hatalmasat fejlődik minden téren. Szakdolgozatom témája érinteni fogja a hardver komponenseket, melyek az elmúlt évtizedben hatalmasat fejlődtek. Minden évben újabb hardverek jelennek meg a piacon, amelyeket egy átlagember, már nehezen bír követni. A 21. században már rengeteg hardver gyártó van, az emberek számára az egyes hardverek közti teljesítmény különbségek pedig már nem egyértelműek. Ez főleg a játékosközösségnek jelent problémát, akik sokszor nem tudják, hogy a megvásárolt/megvásárlandó játékszoftverhez szükséges erőforrás megtalálható-e a számítógépükben. Szakdolgozatomban egy olyan weboldalt szeretnék létrehozni, mely segíti a játékosközösséget ebben a problémában.

Abstract

IT is one of the fastest growing sectors in the world. Year on year, it is making huge progress in all areas. The topic of my thesis will be hardware components, which have undergone a huge development in the last decade. Every year, new hardware appears on the market, which is difficult for the average person to keep up with. In the 21st century there are so many hardware manufacturers that the performance differences between different hardware are no longer obvious to people. This is a particular problem for gamers, who often do not know whether the resources needed for the gaming software they want to buy/are interested in buying are available in their computer. In my thesis I want to create a website that helps the gaming community to solve this problem.

Tartalom

1.	Bevezetés	4
2.	Irodalomkutatás	5
2.1	Információ szerzés a múltban.....	5
2.2	Hasonló rendszerek kutatása, elemzése	7
2.2.1	A legelterjedtebb magyar nyelvű alternatíva.....	7
2.2.2	Egy hasonló megközelítés	12
2.2.3	Egy automatizált megoldás.....	14
2.3	REST, SOAP összehasonlítás	16
2.3.1	REST	16
2.3.2	SOAP	17
2.3.3	REST, SOAP összehasonlítás-összegző gondolatok.....	17
3.	Követelmény specifikáció	19
4.	Tervezés.....	20
4.1	Használandó technológiák	20
4.1.1	Back-end technológiák	20
4.1.2	Választott Backend keretrendszer	21
4.1.3	Frontend technológiák	22
4.1.4	Választott Frontend technológia.....	22
4.1.5	Használandó adatbázis – Microsoft Azure SQL Database.....	23
4.1.6	Microsoft Azure SQL Database előnyei.....	23
4.2	Rendszer tervezése	23
4.2.1	Az adatbázis tervezése.....	24
4.2.2	Jogosultság RBAC vs ABAC	25
4.2.3	Jogosultsági szintek az alkalmazásban	27
4.2.4	Hardver adatok összegyűjtése-web scraping.....	28
4.2.5	Analizációs szoftver	29
4.2.3	Az alkalmazásban található Api végpontok	31
4.2.4	Fejlesztés tervezett menete	31
5.	Fejlesztési napló	33
5.1	Felhasznált csomagok	33
5.1.1	Swagger	33

5.1.2	Nswag studio	33
5.1.3	Selenium	34
5.1.4	Automapper	34
5.1.5	CloudinaryDotNet	34
5.1.6	Material UI	34
5.2	Hardverek kinyerése a számítógépből- Analizációs szoftver	35
5.3	A fejlesztés menete	36
5.3.1	Adatbázis létrehozása	36
5.3.2	Scraper megírása	37
5.3.3	Hátralévő CRUD funkciók megírása.....	38
5.3.4	Autentikáció	38
5.3.5	Összehasonlító logika	39
5.3.6	Képek feltöltése	39
5.3.7	Felhasználói felület fejlesztése	40
6.	Tesztelés	40
6.1	Egység tesztek.....	40
7.	Az elkészült alkalmazás	42
8.	Továbbfejlesztési lehetőségek	52
9.	Összegzés	53

1. Bevezetés

Szakdolgozatom célja egy olyan modern felhasználóbarát webes alkalmazás készítése, amely lehetővé teszi a videójátékokról szóló fontos információk beszerzését. A mai világban a hardverek, játékok rohamosan fejlődnek így az új videójátékoknak egyre nagyobb a gépigényük, aminek a következménye az, hogy sokszor első ránézésre nehéz megmondani, hogy az adott videójáték megfelelően működne-e a felhasználó számítógépén. Manapság a játékosközösség ezekhez az információkhoz az interneten jut hozzá.

Azonban egy létező probléma az, hogy sok felhasználó nincsen tisztában azzal, hogy milyen hardver komponensek találhatóak a számítógépében vagy ha tisztában van ezekkel, akkor sokszor nem tudja eldönteni, hogy a számítógépében található hardver erősebb-e a gépigényben megadott hardvertől.

Célom egy olyan webes alkalmazás létrehozása, ahol a játékosok képesek áttekinteni a hivatalos -fejlesztők által kiadott gépigényeket. Ezt a weboldalt pedig megtámogatom még egy automata hardver felmérő segédprogrammal. Amely képes pontosan és gyorsan felmérni, beazonosítani a felhasználó számítógépében található hardver komponenseket. Ezután pedig egy átlátható felületen megjeleníteni, hogy a számítógép képes lenne-e az adott játékszoftvert megfelelően (kompromisszumok nélkül) futtatni.

Szakdolgozatom második fejezetében elemezni fogom a hasonló rendszereket, megjelenés, illetve funkciók szempontjából. Szakdolgozatom szempontjából a legfontosabb funkció az, hogy hogyan történik a hardverek hozzárendelése a felhasználó profiljához, illetve hogyan történik a játék gépigényének és a hardver komponenseknek az összehasonlítása.

A harmadik fejezetben (Követelmény specifikáció) meghatározom a rendszeremmel kapcsolatos elvárásokat, felsorolom, hogy milyen funkciókat kell tudnia a rendszeremnek.

A negyedik fejezetben pedig felsorolom azt, hogy milyen technológiákkal fogok dolgozni a fejlesztés során, illetve megtervezem a rendszeremet.

2. Irodalomkutatás

2.1 Információ szerzés a múltban

A PC (Personal computer) a világon a legelterjedtebb platform, de a játékszoftverek szempontjából nem ez a legjelentősebb. Ha a piacra bocsájtott mennyiséget nézzük, akkor ezekből jóval több van, mint a kifejezetten játékokra szánt konzolokból, de ezeknek nagy része irodai felhasználásra készültek. Ennek köszönhető az is, hogy a mai videojátékgyártók elsősorban konzolra fejlesztenek, hiszen itt egy fix, 'kötött' hardverről van szó, ezáltal jóval egyszerűbb a fejlesztési folyamat. Azonban egy játékra szánt számítógép mellett több érv is szól.

Egyrészt a PC mindig élvonalnak tekinthető, hiszen itt gyakrabban történnek újabb fejlesztések, élen jár a technológiában. Sokszor a konzolok csak a számítógépek után jönnek teljesítményt illetően. Illetve a számítógépes játékok mindig fejlettebbnek tekinthetők (például a grafikai kinézet szempontjából), mint a konzolosok, hiszen a számítógépek jóval egyénre szabhatóak a különböző hardverek miatt. A mai modern számítógépes hardverek segítségével szinte minden esetben nagyobb felbontás, képfriessítés érhető el, viszont ezek mindegyike jobban is terheli a hardvert és emiatt kell a számítógépünket állandóan fejleszteni. Minél több extra funkciót állítunk be egy játékban, annál nagyobb lesz az adott játék gépigénye. Például: a számítógépünket rendkívül terheli az, hogy milyen felbontásban játszunk (minél nagyobb felbontást választunk, annál jobban terheli a videojáték a számítógépünket), extra grafikai funkció még az árnyék is, mely rendkívül fontos a játékok szempontjából, azonban nagyon terheli a processzorunkat, illetve a videokártyánkat.

Számítógépes hardvereket ma már nagyon sok cég gyárt. Videojáték szempontjából három fő komponens van. Processzor, Videokártya és RAM. Ezek adják egy játékra szánt számítógépnek a lelkét. A videokártya felelős a játékelemek kirendereléséért (ez jeleníti meg a látott képeket, jelenteket, animációkat). A processzor bizonyos feladatokat kap a játék futtatása közben (olyan feladatokat, amelyekben a gpu nem jelesekedik) ilyen például az NPC (nem játékos által irányított karakterek) mesterséges intelligenciája. A RAM (közvetlen hozzáférésű memória), pedig tárolja az adatokat, illetve a processzor által végrehajtandó programokat. Fontos, hogy a RAM csak addig tudja tárolni az adatokat míg feszültség alatt van az eszköz, tehát ha kikapcsoljuk, újraindítjuk az eszközünket, akkor a RAM-ban tárolt adatok elvesznek.

Gépigénynek nevezzük a számítógép alkotórészeinek (komponenseinek) összességét, amelyek a szoftver futtatásához nélkülözhetetlenek. A szoftverek gépigényét minden esetben a gyártó, fejlesztő adja ki (mindig van egy célhardver). De a gépigényeket pedig hosszas tesztelesek során határozzák meg. Ezeknek a tesztelési folyamatoknak az alapszabálya az, hogy a legnépszerűbb hardverek mindegyikén le kell tesztelni a játékot, és megvizsgálni azt, hogy egyáltalán elindul e rajta és ha elindul akkor hogyan fut, milyen

hibákat produkál a játékszoftver. Ezt nevezzük Quality Assurancenek. Minden videójátékfejlesztő és kiadó rendelkezik ilyen tesztelői csapatokkal, akiknek a munkájuk az, hogy újra és újra leteszteljék a játék egy fázisát, részét a különböző hardvereken. [1]

A gépigényt pedig minden esetben két fő részre lehet bontani. Minimum (mi a minimális követelmény a szoftver futtatásához) és ajánlott (melyen a szoftver már kompromisszumok nélkül képes működni). A játékos közösség régebben magazinokból, illetve a szoftverek dobozán (általában a hátsó oldalán) lévő információkból tudta meg a gépigényt. Azonban ez a mai modern világban megváltozott, szinte a teljes közösség ma már online felületeken jut hozzá ezekhez az információkhoz. Számtalan weboldal jött létre, amelyek a játékok gépigényeivel foglalkoznak, ebben a fejezetben pedig ezeket fogom elemezni.

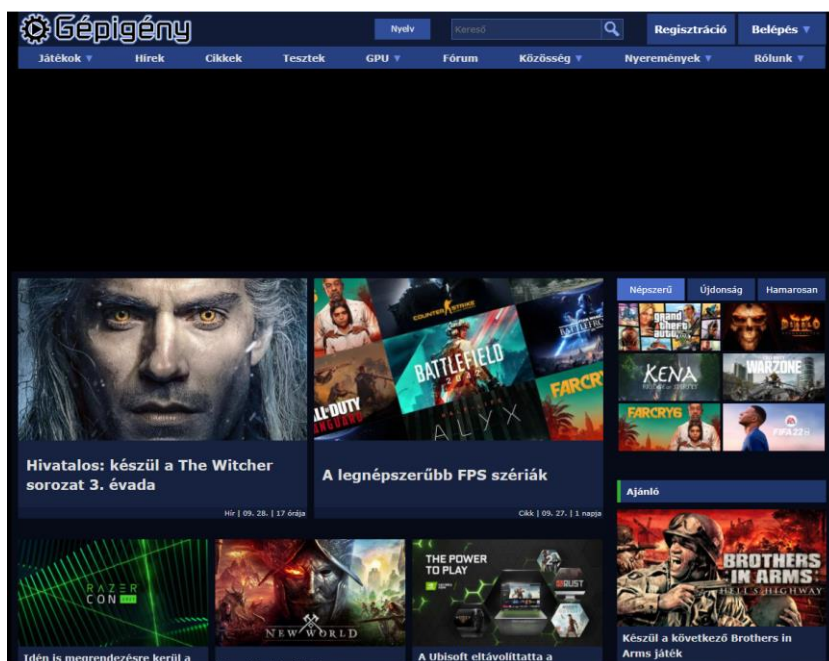
2.2 Hasonló rendszerek kutatása, elemzése

2.2.1 A legelterjedtebb magyar nyelvű alternatíva

Az első weboldal, amely szinte mindenkinek az eszébe jut az nem más, mint a 'gépígeny.hu'. Ez a weboldal a hazai viszonylatokat tekintve szinte teljesen egyedülálló a piacon.

„A Gépígeny.hu egy közösségi oldal a videojátékosok és számítógép felhasználók számára, amely már 2007 óta üzemel. Eleinte csak PC-s játékok gépígenyeinek adatbázisaként működött, de a sok látogatónak és a rengeteg regisztráló aktív tagnak köszönhetően közösségi oldallá vált. 2009-ben elkészült az oldal angol nyelvű változata, amely a Gamesystemrequirements.com címen érhető el, majd később egy lite változata, amely további 12 nyelven, a systemanforderungen.com és a systemreqs.com címen érhetőek el. Ma már több mint 200 000 regisztrálója, és havonta több mint 300 000 látogatója van a Gepigeny.hu és lokalizált oldalainak.” [2]

A weboldal számtalan funkcióval rendelkezik, mely a játékos közösség érdekeit szolgálja. Ilyen például a fórum, hírek, játék/hardver keresés, tesztek. A legtöbb funkció azonban csak regisztrálás után érhető el a felhasználó számára. A weboldal magyar változata 2007-ben készült el azóta a felhasználói felületben (User Interface) nem történt sok változás (1. ábra).



1. ábra gépígeny.hu főoldala

Az oldal megjelenése tipikusan egyáltalán nem modern, nem követ semmilyen modern dizájnt (például: playful cursors, white space, dynamic scrolling,).

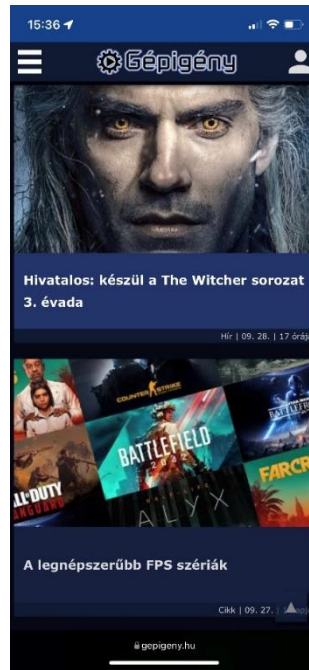
A white space a minimalista webdizájn gyakran felhasznált eleme, de ma már a modern webhelyek kritikus tulajdonsága, jellemzője. A tartalom (például: a szöveg, linkek képek) white space-el való kiegyensúlyozásával a felhasználók természetes módon figyelnek a webhely kulcsfontosságú elemeire. Bár a white space mennyiségére nincsen semmilyen szigorúan meghatározott szabály, elterjedt módszer az, hogy minden tartalom között legalább némi white space-t (fehér teret) kell hagyni. Ez azért nagyon hasznos, fontos mert a weboldal navigációja meghatározó, kritikus szerepet játszik a felhasználók elégedettségében. Ha weboldalunk navigációja, tartalma átláthatatlan, akkor a felhasználók nagyobb valószínűséggel kattintanak át egy másik weboldalra. A white space segít a felhasználók figyelmét azokra a tartalmakra, elemekre összpontosítani, melyek a rendszer szempontjából a legfontosabbak. Összegezve a white space segít abban, hogy az oldal fő funkcióit kiemeli a többi részhez képest. [3]

Viszont annak ellenére, hogy az oldal nem követi például a white space dizájnt a tartalom rendkívül egyszerűen jelenik meg a felhasználó számára, a megfelelő kategorizálás miatt a keresés pedig könnyű. A weboldalon azonban nem található semmilyen animáció. Nincs lehetőségünk megjelenítési téma kiválasztására (light, dark theme), viszont hatalmas pozitívum, hogy az oldal alap háttere sötét színű.

„A sötét mód használata csökkenti a képernyő által kibocsátott fényt, ez pedig jó hatással van mind mentális állapotunkra, mind pedig szemünk egészségére, különösen abban az esetben, ha eszközünket este használjuk. Abban az esetben ugyanis, ha körülöttünk sötét van, mi pedig a fényes képernyőre meredünk, szemünk sokkal gyorsabban károsodhat a túlzottan nagy kontrasztnak köszönhetően.” [4]

Illetve a sötét mód csökkenti az eszközünk által kibocsátott kék fényt, melynek kártékony hatását több kutatás is bizonyítja.

Az oldal viszont reszponzív, mobilos megjelenítéssel sem lesz gond (2. ábra).



2. ábra gepigeny.hu főoldala mobilon

A reszponzív design segít abban, hogy a weboldalunk rugalmasan alkalmazkodjon a különböző képernyőméretekhez, annak érdekében, hogy optimálisan jelenítse meg a weboldalunkat a felhasználó számára minden eszközön (pl: asztali számítógép, laptop, tablet). A reszponzív biztosítja azt, hogy a felhasználói élmény tökéletes maradjon, ne rontsa esetleg az élményt az, hogy kisebb kijelzőn rosszul jelenik meg a weboldalunk.

Bejelentkezést követően számtalan funkció elérhető a felhasználó számára (például: üzenetek, ismerősök, játékok, értesítések, nyereményjáték), de a szakdolgozatom szempontjából az egyik legfontosabb az eszközök hozzáadása lehetőség. Itt lehetőségünk van különböző eszközöket hozzáadni, hozzárendelni a profilunkhoz (például: laptopot, asztali számítógépet, telefont).

A játékkal való összehasonlításhoz pár komponenst kötelező megadnunk (a többi hardver, periféria csak opcionális lehetőség). Kötelező komponensek közé tartozik számítógépünk processzora, videokártyája, ram mennyisége és az operációs rendszere.

Ezeket egy legördülő listából van lehetőségünk kiválasztani (3. ábra) (ha elkezdjük gépelni például a processzorunk típusát akkor a keresést szűkíti az addigi begépett karakterek alapján).

Eszköz szerkesztése

Az eszköz neve vagy típuszáma: ?

Típus: Asztali PC

Az összehasonlításhoz szükséges főbb adatok:

CPU: Ryzen 5 3600X x	RAM: 16 GB v
GPU: GeForce GTX 1660 Ti x	OS: Windows 10 64bit v

3. ábra Eszköz hozzáadása

Ha szeretnénk megtudni, hogy egy videójátéknak milyen gépigénye van, illetve azt is, hogy számítógépünk képes-e az adott játékszoftver futtatására, akkor a weboldalon a keresési lehetőségnél megkeressük a játékot, ezt pedig képesek vagyunk 'összehasonlítani' az általunk elmentett eszközzel. Ilyenkor a háttérben a rendszer összehasonlítja (a fejlesztők által kiadott) gépigényben szereplő hardver komponenseket a mi számítógépünk hardver komponenseivel. Majd az eredményt feltünteti egy táblázatban (4. ábra). Amely a felhasználó számára könnyen áttekinthető.

Minimális gépigény:

✓ CPU: Intel Core i3-6100 @ 3.7GHz or AMD Athlon X4 880K @4GHz

✓ RAM: 8 GB RAM

✓ GPU: NVIDIA GTX 660 2GB or AMD Radeon HD 7850 2GB

✓ OS: Windows 10 - 64-Bit

Tárhely: 50 GB available space

Ajánlott gépigény:

✓ CPU: Intel i5-3550 @ 3.40GHz or AMD FX 8150 @ 3.6GHz

✓ RAM: 8 GB RAM

✓ GPU: NVIDIA GeForce GTX 670 or AMD Radeon R9 270X

✓ OS: Windows 10 - 64-Bit

Tárhely: 50 GB available space

Hálózat: Broadband Internet connection

Összehasonlítás eredménye:

Átlag fps (~):	720p (HD)	900p (HD+)	1080p (FHD)	1440p (QHD)	2160p (4K/UHD)
Nagyon alacsony	60+ fps	60+ fps	60+ fps	60+ fps	60+ fps
Alacsony	60+ fps	60+ fps	60+ fps	60+ fps	60+ fps
Közepes	60+ fps	60+ fps	60+ fps	60+ fps	60+ fps
Magas	60+ fps	60+ fps	60+ fps	60+ fps	60+ fps
Nagyon magas	60+ fps	60+ fps	60+ fps	60+ fps	60+ fps

Megosztás

Visszajelzés

Működés és általános megjegyzések

X

Összehasonlítás:

Saját Gépem

Összehasonlít

CPU: Ryzen 5 3600X

x

RAM: 16 GB

Konfig mentése

GPU: GeForce GTX 1660 Ti

x

OS: Windows 10 64bit

További beállítások

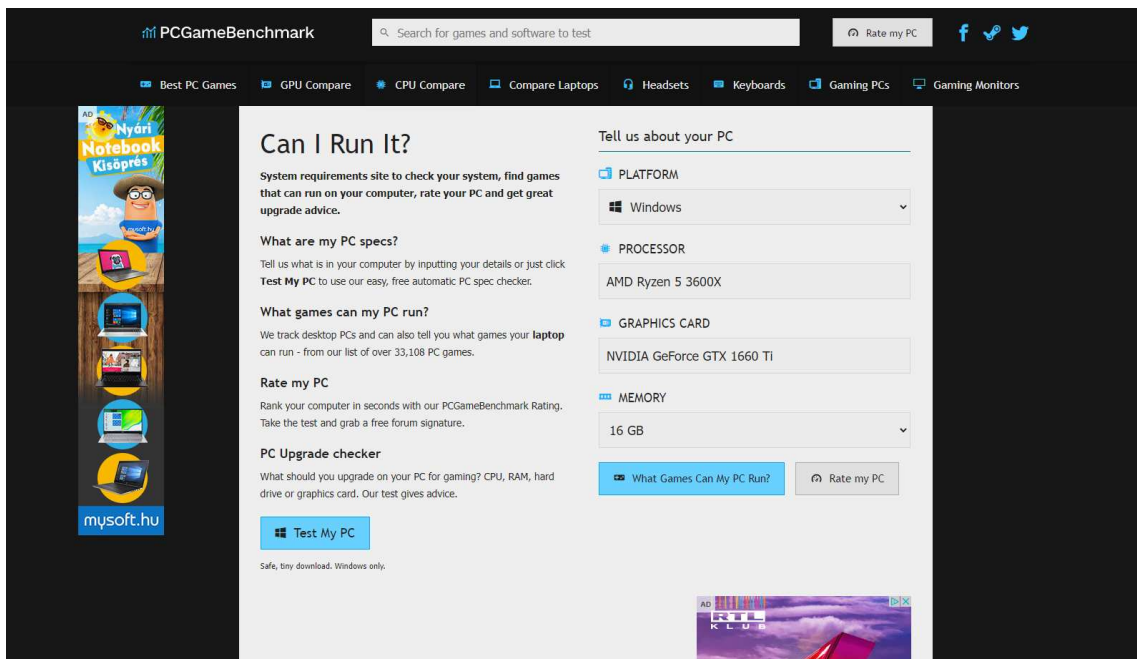
4. ábra Eredmény megjelenítés

Összegezve a gépigény.hu egy meghatározó oldal a magyar játékosközösség számára. Funkciói rendkívül fontosak, hasznosak a felhasználók számára. Negatívumokhoz tudnám sorolni az oldal megjelenítését, mert az oldalon nem találhatóak olyan animációk, egyedi ötletek, amelyek javítanák a felhasználói élményt (UX). Valamit az eszköz hozzárendelése funkciót, mert nem automatizált megoldás. A felhasználók számára sokkal egyszerűbb lenne egy automatizált megoldással hozzárendelni az eszközüket a profilukhoz. Valamint az oldal tele van hirdetéssel, ami rendkívül zavarja a felhasználói élményt.

2.2.2 Egy hasonló megközelítés

Hasonló rendszerek közé tartozik a pcgamebenchmark.com nevezetű weboldal is. Az előző oldallal ellentétben, itt egy teljesen másfajta megközelítést kapunk. Míg a [gépigény.hu](https://gepigeny.hu) egy játékosközösségi platformnak tekinthető, ez kevésbé (például ezen a weboldalon nincsen fórum, hírek kategória).

Ez egy jóval hardver specifikusabb oldal (5. ábra). Fő funkció közé sorolnám a hardver, periféria keresési, összehasonlítási lehetőséget. Számtalan cikkel, teszttel rendelkeznek a hardverekről, amely rendkívül segíti a felhasználókat, ha éppen új hardvert szeretnének vásárolni.



5. ábra pcgamebenchmark.com főoldala

A weboldal megjelenése letisztult (könnyen áttekinthető, az információ nem ömlesztve található rajta), emiatt kicsivel modernebbnek mondható a [gépigény.hu](https://gepigeny.hu)-nál (például a weboldal kinézetéhez sokat hozzáad a különböző ikonok, CTA (call to action gombok) használata. A CTA gombok, olyan gombok, amiket a weboldalon azért használnak, hogy a felhasználókat a cél felé irányítsák (jelen esetben: 'milyen játékok futtatására képes a számítógémem?'). Az oldalon a felhasználó számára rendkívül könnyű keresési lehetőség a megfelelő kategorizálás miatt. A weboldal reszponzív, tehát a különböző felbontású, méretű képernyőkön, illetve a mobilon való megjelenítéssel sincs probléma. Az oldalon nem található semmilyen animáció. Nincsen lehetőségünk megjelenítési téma kiválasztására sem (sötét, világos téma), viszont az alap megjelenés sötét ez pedig

rendkívül barátságos a szemnek. A felhasználói élményt viszont ronthatják az oldalon talált (hatalmas) hirdetések.

A hirdetések miatt a felhasználók többsége hamar frusztrálttá válik. Ez a frusztráció volt az egyik fő oka a reklámblokkolók népszerűségének. A Global Web Index kutatásai szerint [5] minden második ember használ reklámblokkolót, hogy eltüntesse a nem kívánatos hirdetéseket a képernyőjéről, mert idegesítőnek, frusztrálónak találja azokat. Ez gyorsan összekapcsolható, párhuzamba vonható a felhasználói élménnyel. A hirdetések sok esetben elvonják, elterelik a felhasználó figyelmét a weboldaltól, ezáltal nem tudják elérni a céljaikat az adott webhelyen. Minél több hirdetés van a weboldalon, a felhasználók annál nehezebben tudják elérni a céljaikat. Egyes felhasználókat pedig annyira frusztrálhatják a hirdetések, hogy keresnek egy másik weboldalt, amely ugyanazokat a funkciókat biztosítja a számukra, csak hirdetések nélkül. [6]

Az oldalon nincsen regisztrációs lehetőség. Így nincs lehetőségünk az eszközünket 'elmenteni'. A hardvereinket minden 'új látogatásnál' meg kell adnunk, a rendszer nem tudja elmenteni. Az előző képen látszik, hogy itt is (az előző megoldáshoz hasonlóan) egy legördülő listából van lehetőségünk kiválasztani számítógépünk hardver komponenseit.

Miután a felhasználó megadta a számítógépének komponenseit, két lehetősége van. Egyik az, hogy megnézi, hogy a számítógépe milyen játékszoftverek futtatására alkalmas, a másik pedig az, hogy egy elemzést kér a számítógépéről.

Az elemzés eredménye megadja, hogy a számítógépünk hány játékszoftver futtatására képes az 1000 'popular' játék közül. Valamint megadja a leggyengébb láncszemet, hardvert is a gépünkben és leírja azt is, ha ezt lecseréljük egy jobb hardverre, akkor az mekkora hatással lesz számítógépünk teljesítményére.

A másik lehetőség, hogy megnézzük, hogy a számítógépünk milyen szoftverek futtatására alkalmas. Ekkor egy felsorolást kapunk az eredményről. Ezt pedig lehetőségünk van több módon is rendezni. Rendezhetjük népszerűség, játékosszám, kiadási dátum és metascore (A metascore a Metacritic weboldalához tartozó értékelési skála) alapján. A rendszer hatalmas adatbázissal rendelkezik, a saját számítógépem konfigurációjára több mint 5287 futtatható játékot talált.

Pozitívumként tudnám felsorolni az oldal modernebb megjelenését, rendkívül javítja a felhasználói élményt. A weboldalon található funkciók hasznosak a játékosok számára, minden funkció tökéletesen működik, gyors választ kapunk. Negatívum közé sorolnám az oldalon található nagy méretű hirdetéseket, amely rontja a felhasználói élményt. Valamint negatívum az is, hogy az eszközünket nincsen lehetőségünk elmenteni, minden egyes új látogatásnál meg kell adnunk a hardvereket. Sajnos itt is egy manuális megoldásról van szó, hiszen nincsen egy automatizált folyamat, mely felmérné

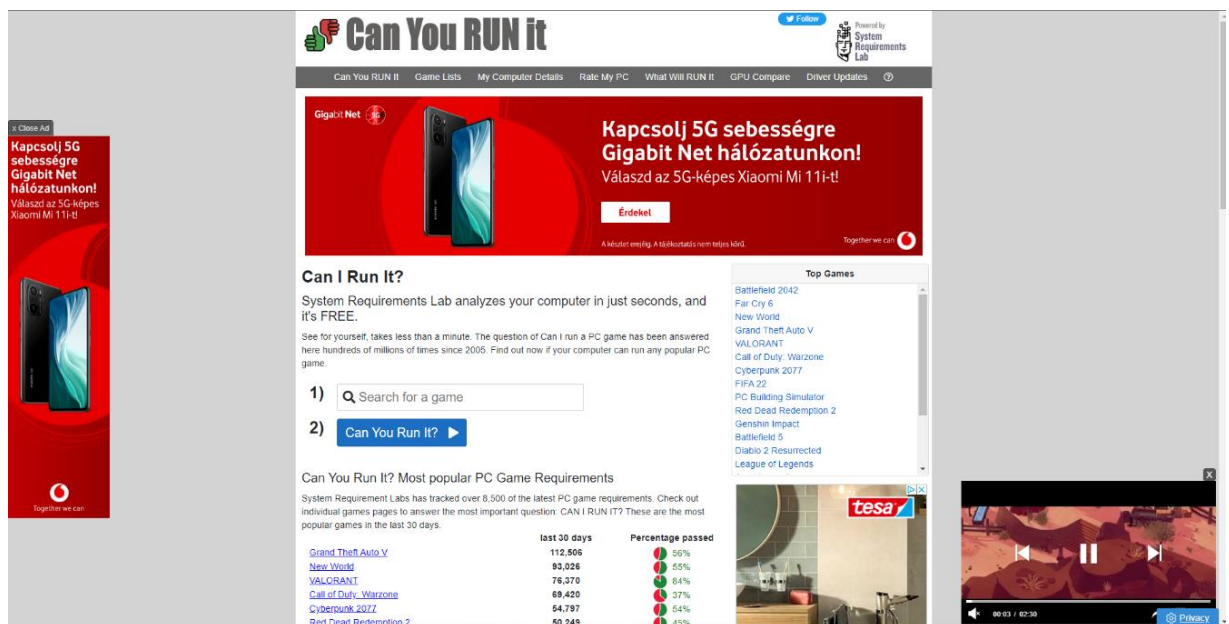
számítógépünk hardvereit, a felhasználónak a hardvereket egyenként, manuálisan kell megadnia.

2.2.3 Egy automatizált megoldás

A következő hasonló rendszer a systemrequirementslab nevezű weboldal, mely az előzőkhez képest újabb megoldásokkal áll elő. Röviden annyit érdemes róla tudni, hogy ez is a hardver specifikusabb weboldalak közé sorolható (mint a pcgamebenchmark.com).

A weboldal megjelenése az eddig elemzett rendszerek közül a legrosszabb (6. ábra). Az oldal megjelenése világos színű, ami késői órákban már rendkívül barátságtalan a felhasználó szemének. Ezen az oldalon található a legtöbb (és egyben legnagyobb) hirdetés is.

A főoldalon ömlesztve jelenik meg az információ, mely egyes felhasználók számára áttekinthetetlen is lehet. Viszont ez az első weboldal, ahol minimális effektekkel találkozunk. Az oldalon sokszor előfordul a 'hover' effekt mely kicsit színesebbé teszi a weboldalt.



6. ábra systemrequirementslab.com főoldala

Viszont az oldalon nagyon hasznos funkciók találhatóak, melyek segítik a felhasználót. Az egyik ilyen funkció a 'What Will RUN It'. Ez egy olyan funkció, amely megmutatja, hogy hány népszerű játék képes elfutni, működni az új játékokra szánt laptopokon, számítógépeken (ezeknek pedig feltünteteti az árát is).

My Computer Details funkció képes megállapítani teljesen automatikus módon azt, hogy milyen hardverek találhatóak a felhasználó számítógépében. Ehhez nincs másra szükség,

mint hogy (a weboldalon található, elérhető) exe állományt letöltjük, melyet a Rate my pc gomb lenyomásával tehetünk meg. Ezután ezt a (teljesen megbízható) exe állományt kell futtatnunk a számítógépünkön, mely automatikusan elemzi, analizálja hardvereinket.

Ha ez sikeres volt akkor a weboldalon A 'My Computer Details' fül alatt számtalan információt találhatunk meg számítógépünkéről például milyen processzor, videokártya, memória, operációs rendszer, BIOS verzió található a számítógépünkben/ön (7. ábra). Azonban ezekről még számtalan információt gyűjt a program.

Processor	
Processor	AMD Ryzen 5 3600X 6-Core Processor
Manufacturer	AMD
Speed	4.4 GHz
Number of Cores	12
CPU ID	178BFBFF00870F10
Family	17
Model	71
Stepping	0

7. ábra Processzorról gyűjtött információk

Az automata hardver felmérés pontos, több számítógépen is teszteltem. Az oldalon találhatunk egy részletes leírást, hogy ez minek a segítségével is történik.

A felhasználó szempontjából ez sokkal előnyösebb megoldás, mint az eddigiek (ahol listából kellett kiválasztani a hardvereket), mert sokkal kényelmesebb (az automatizáció miatt) illetve sok felhasználó nem tudja, hogy milyen hardverek találhatók a számítógépében, ezzel a szoftverrel pedig tényleg egy percen belül kiderül. Számítógépünk konfigurációja pedig elmentésre kerül, nincs szükség minden egyes alkalomkor lefuttatni az analízációs szoftvert.

Ezután pedig lehetőségünk van megnézni a 'Game Lists' fülnél, hogy milyen játékok képesek megfelelően működni a számítógépünkön. A játékok kilistázása rendkívül egyszerű, könnyen áttekinthető a felhasználók számára.

Összességében a weboldal számtalan hasznos, jól működő funkciót biztosít a felhasználók számára. Szerencsére az oldal működése gyors, rögtön választ kapunk minden egyes kérésre (amit a szerver felé intézünk). A hardver elemző szoftver gyors és pontos. Negatívumokhoz sorolnám, az oldal megjelenését, kinézetét (ami egyáltalán nem modern), a rajta található nagy hirdetések. A hirdetések elterelhetik a felhasználó figyelmét a weboldalunkról, valamint sok felhasználó számára frusztráló lehet. Az oldal kinézetében pedig nincsen semmi különleges, egyedi dolog, mely javítaná a felhasználói élményt. Az információk zsúfolva találhatóak meg rajta, mely nehezíti az oldal tartalmának áttekintését.

2.3 REST, SOAP összehasonlítás

A REST és a SOAP az online adatátvitel 2 különböző megközelítése. Konkrétan mindkettő azt határozza meg, hogyan kell alkalmazásprogramozási interfészeket (API-kat) létrehozni, amelyek lehetővé teszik az adatok webes alkalmazások közötti kommunikációját. A reprezentációs állapotátvitel (REST) elvek összessége. A SOAP (Simple Object Access Protocol) a World Wide Web Consortium (W3C) által fenntartott hivatalos protokoll. A fő különbség az, hogy a SOAP egy protokoll, míg a REST nem az. Általában egy API a REST vagy a SOAP szerint működik, a felhasználási esettől és a fejlesztő preferenciáitól függően.

2.3.1 REST

A REST a könnyű webszolgáltatások és mobilalkalmazások igényeihez igazított architektúráis elvek összessége. Mivel egy irányelvkészletről van szó, ezen ajánlások megvalósítását a fejlesztőkre bízta.

Amikor egy REST API-hoz adatigénylést küldenek, az általában a HyperText Transfer Protocol (közismert nevén HTTP) keresztül történik. A kérés beérkezése után a REST-re tervezett API-k (amelyeket RESTful API-knak vagy RESTful webszolgáltatásoknak neveznek) különböző formátumú üzeneteket küldhetnek vissza: HTML, XML, egyszerű szöveg és JSON. A JSON (JavaScript object notation) azért kedvelt üzenetformátum, mert bármilyen programozási nyelvvel olvasható (a neve ellenére), ember és gép által egyaránt olvasható és könnyű. Ily módon a RESTful API-k rugalmasabbak és könnyebben beállíthatók.

Egy alkalmazás akkor tekinthető RESTfulnak, ha 6 architektúráis irányelvet követ. Egy RESTful alkalmazásnak rendelkeznie kell:

1. Kliens-kiszolgáló architektúra, amely kliensekből, kiszolgálókból és erőforrásokból áll.
2. Állapotmentes ügyfél-kiszolgáló kommunikáció, ami azt jelenti, hogy a kliens tartalma a kérések között nem tárolódik a kiszolgálón. A munkamenet állapotára vonatkozó információkat ehelyett az ügyfélnél tartják.
3. Cache-elhető adatok, amelyek kiküszöbölik bizonyos ügyfél-kiszolgáló interakciók szükségességét.
4. Egységes interfész a komponensek között, így az információátvitel szabványosított formában történik, nem pedig az alkalmazás igényeihez igazodóan. Ezt Roy Fielding, a REST megalkotója úgy írja le, mint "a központi jellemzőt, amely megkülönbözteti a REST architektúráis stílust más hálózat alapú stílusoktól".
5. Réteges rendszerkényszer, ahol az ügyfél-kiszolgáló kölcsönhatásokat hierarchikus rétegek közvetíthetik.

6. Code on demand, amely lehetővé teszi a szerverek számára, hogy futtatható kód átadásával bővítsék a kliens funkcionalitását (bár a láthatóságot is csökkenti, így ez egy opcionális irányelv).

2.3.2 SOAP

A SOAP egy szabványos protokoll, amelyet először azért terveztek, hogy a különböző nyelveken és különböző platformokon készült alkalmazások kommunikálni tudjanak egymással. Mivel protokollról van szó, beépített szabályokat ír elő, amelyek növelik a bonyolultságát és a többletköltséget, ami hosszabb oldalbetöltési időkhöz vezethet. Ezek a szabványok azonban olyan beépített megfeleléseket is kínálnak, amelyek előnyösebbé tehetik a vállalati forgatókönyvek esetében. A beépített megfeleléségi szabványok közé tartozik a biztonság, az atomicitás, a konzisztencia, az izoláció és a tartósság (ACID), amely egy sor tulajdonság a megbízható adatbázis-tranzakciók biztosítására.

Amikor egy SOAP API-hoz adatigénylést küldenek, azt bármelyik alkalmazási réteg protokollon keresztül lehet kezelni: HTTP (webböngészők számára), SMTP (e-mailhez), TCP és mások. A kérés beérkezése után azonban a SOAP-üzeneteket XML-dokumentumként kell visszaküldeni - egy olyan jelölőnyelvként, amely ember és gép által egyaránt olvasható. A SOAP API-nak küldött, befejezett kérést a böngésző nem tudja gyorsítótárba helyezni, így később nem lehet hozzáférni az API-nak történő újbóli elküldés nélkül.

2.3.3 REST, SOAP összehasonlítás-összegző gondolatok

Sok régi rendszer még mindig a SOAP-ot használja, míg a REST később jött, és gyakran gyorsabb alternatívának tekintik a webalapú forgatókönyvekben. A REST egy olyan irányelvkészlet, amely rugalmas megvalósítást kínál, míg a SOAP egy olyan protokoll, amely speciális követelményekkel rendelkezik, mint az XML üzenetküldés.

A REST API-k könnyűek, így ideálisak az olyan újabb kontextusokban, mint a tárgyak internete (IoT), a mobilalkalmazások fejlesztése és a szerver nélküli számítástechnika. A SOAP webszolgáltatások beépített biztonságot és tranzakció-megfelelőséget kínálnak, amelyek számos vállalati igényhez igazodnak, de emiatt nehezebbek is. Emellett számos nyilvános API, például a Google Maps API, a REST irányelveket követi. [7]

Levont következtetések, Összegzés

Megjelenés szempontjából fontos, hogy a weboldal felhasználóbarát legyen ezt pedig különböző módszerek, dolgok segítségével érhetjük el. A legfontosabb az, hogy az oldal megjelenése egyszerű, letisztult legyen ezzel segítve annak átláthatóságát. Megkell oldani azt is, hogy a különböző méretű, felbontású kijelzőkön (esetleg mobilon is) megfelelően jelenjen meg az oldal (reszponzív legyen). Kényelmi szempontból fontos az is, hogy a weboldalon domináló színek barátságosak legyenek a felhasználó szemének (akár az esti órákban is).

Funkciók szempontjából fontos, hogy a weboldal automata hardver felmérő programmal legyen megtámogatva, mert ez egy sokkal kényelmesebb, pontosabb megoldás. Viszonylag nagy méretű adatbázissal rendelkezzen (hardverek szempontjából). Valamint gyors választ adjon, minden egyes kérésre.

3. Követelmény specifikáció

A hasonló rendszereket megvizsgálva számtalan új információra, benyomásra tettem szert. Mivel ezeket a weboldalakat én is csak egy felhasználóként teszteltem, ezáltal könnyebb volt azonosítani az olyan dolgokat, amelyek rontják, illetve javítják a felhasználói élményt. Mivel a fő célom egy felhasználóbarát weboldal létrehozása, amely jó felhasználói élményt nyújt, így a weboldal megjelenésével kapcsolatban is vannak fontos elvárások.

A pgamebenchmark.com weboldal megjelenése előnyös, hiszen letisztult (tehát az információk szépen blokkokban helyezkednek el, nem tömören, zsúfoltan jelenítődnek meg). **Fontos, hogy az oldalon található információk könnyen áttekinthetők legyenek a felhasználó számára.** Szintén fontos, hogy a felhasználót a cél felé irányítsuk ehhez pedig szükségesek a CTA gombok használata.

Mivel a felhasználói élményt rontják a weboldalon található hirdetések, ezért az oldalon nem lesznek megtalálhatóak ezek. Mivelhogy a sötét módnak számtalan előnye van (például nem erőlteti meg annyira a szemet, kevesebb kék fényt bocsájt ki eszközünk) emiatt a weboldal alap megjelenése ilyen lesz. Fontos, hogy a weboldal (az összes vizsgált hasonló rendszerhez hasonlóan) reszponzív legyen, így ne legyen probléma a különböző méretű, felbontású kijelzőkön való megjelenítéssel.

Megjelenéssel kapcsolatos dolog még az is, hogy a mai trendeknek, dizájnoknak megfelelő oldal készüljön (például: 2021-ben az egy legjobban elterjedt dizájn a minimalista, white space megjelenítés). Illetve tapasztalatom szerint rendkívül javítja a felhasználói élményt a kurzor effektek, animációk használata. Persze meg kell találni a harmóniát és nem szabad túlzásokba esni, hiszen a túl sok effekt, animáció zavarhatja egyes felhasználókat.

Funkciók szempontjából fontos az, hogy a felhasználó bejegyzés nélkül semmit ne érjen el az oldalról. Az egyes funkciók csak bejelentkezést követően lesznek elérhetőek. A weboldal terhelhetősége és a gyors válaszadás is fontos. A hardver felmérő szoftver legyen pontos és gyors. Ennek a szoftvernek nem fontos a megjelenése, hiszen ennek feladata csak felmérni a számítógépben található hardvereket majd elküldeni a szervernek (emiatt lesz konzolos alkalmazás). Viszont a hardver analízisnek (a systemrequirementslab megoldásához hasonlóan) teljesen automatikus módon kell történnie. Mivel ez egy automata megoldás, ez is javítani fogja a felhasználói élményt, hogy nem a felhasználónak kell kitöltenie különböző formokat. Fontos, hogy a felhasználó számítógépének konfigurációja a hardver felmérést követően legyen elmentve, ezzel kiküszöbölve azt, hogy ne kelljen mindig letölteni és futtatni az analízis szoftvert.

4. Tervezés

4.1 Használandó technológiák

Mivel a fejlesztendő rendszer egy weboldal, ezért nagyon sok technológia jöhet szóba. Ezért meg kell vizsgálni, hogy milyen technológiákkal érdemes elkezdni a fejlesztést. A web fejlesztés 'két fő részből áll' back-end, illetve front-end.

4.1.1 Back-end technológiák

Express

Az Express.js egy webes alkalmazás keretrendszer a Node.js számára. Számos olyan funkciót biztosít, amelyekkel a webalkalmazások fejlesztése gyors és egyszerű, ami egyébként több időt vesz igénybe, ha csak a Node.js-t használjuk. Nagyon könnyen kezelhető azok számára, akik megfelelő JavaScript tudással rendelkeznek. Fő előnyei közé tartozik, hogy könnyen konfigurálható és testre szabható. Könnyen integrálható különböző template enginekkel, mint például a Jade, Vash, EJS stb. Lehetővé teszi REST API szerver létrehozását. Valamint könnyen csatlakoztatható olyan adatbázisokhoz, mint a MongoDB, Redis, MySQL.

ASP.NET CORE

Az ASP.NET egy népszerű webfejlesztő keretrendszer webes alkalmazások .NET platformon történő építéséhez, fejlesztéséhez. Az ASP.NET Core az ASP.NET nyílt forráskódú változata, amely macOS, Linux és Windows rendszereken fut. Az ASP.NET Core először 2016-ban jelent meg, és az ASP.NET korábbi, csak Windowsra szánt verzióinak újratervezése [8]. Előnyei közé tartozik, az, hogy egyszerű a modern, kliensoldali keretrendszerek és fejlesztési munkafolyamatok integrálása. Valamint beépített dependency injectionnel rendelkezik. Fő előnye, hogy a modern webfejlesztést leegyszerűsíti. A folyamatos támogatásoknak, fejlesztéseknek köszönhetően pedig ez az egyik legmegbízhatóbb, legnépszerűbb backend keretrendszer. Ide kapcsolható a Microsoft által fejlesztett Visual Studio is, amely egy IDE (Integrated Development Environment), integrált fejlesztői környezet. Amely a C# és .NET fejlesztők közösségének készül, és évek óta jól szolgálja ezt a felhasználói bázist. Az IDE számos hagyományos funkciója mellett a Visual Studio az Azure-specifikus bővítmények választékával a Microsoft ökoszisztémára jellemző lehetőségek sokaságát is kínálja. Valamint olyan eszközkészletet kínál, amely segít a programok kódjának megírásában és módosításában, valamint a programok hibáinak felderítésében és kijavításában.

Django

A Django egy magas szintű Python webes keretrendszer, amely gyors fejlesztésre és tiszta tervezésre ösztönöz. Ingyenes és nyílt forráskódú. Fő elve, hogy ne találjuk fel újra a kereket, így számtalan funkciót biztosít a fejlesztő számára. Kompatibilis a főbb operációs rendszerekkel és adatbázisokkal.

Vizsgált keretrendszer	ASP.NET CORE	EXPRESS	DJANGO
Nyelv	C#	Javascript	Python
Open Source	igen	igen	igen
Támogatottság	Kiváló	jó	jó
Cross platform	Igen	Igen	Igen
Népszerűség	Legnépszerűbb	Népszerű	Népszerű

4.1.2 Választott Backend keretrendszer

Mindhárom vizsgált keretrendszer számtalan előnnyel rendelkezik, de személy szerint én a Microsoft által fejlesztett ASP.NET CORE-t preferálom. Egyrészt ebben van a legtöbb fejlesztői tapasztalatom. Másrészt hatalmas támogatással rendelkezik ez a keretrendszer, valamint számtalan dokumentáció segíti a használatát. Ha valami nem lenne alpból a keretrendszer része, akkor a Nuget Package Manager segítségével biztos találok olyan bővítményt, amit az adott feladat igényel.

4.1.3 Frontend technológiák

React

A React.js egy hatékony és rugalmas JavaScript könyvtár a Facebook által létrehozott felhasználói felületek építéséhez. Technikailag a React egy könyvtár, de gyakran keretrendszerként tárgyalják, és más keretrendszerekkel hasonlítják össze. A React megkönnyíti az interaktív felhasználói felületek létrehozását, mivel átlátható kóddal rendelkezik, amely könnyíti hibakeresést. Továbbá olyan komponensrendszert biztosít, ahol a kódblokkok egyszer írhatók és újra felhasználhatók. az alkalmazás különböző részeiben vagy akár más alkalmazásokban is. Más frontend keretrendszerekkel összehasonlítva a React kódja könnyebben karbantartható és moduláris felépítésének köszönhetően rugalmas. Ez a rugalmasság viszont rengeteg időt takarít meg a fejlesztőknek.

Angular

Egy Javascript keretrendszer, amelyet a Google hozott létre nagy összetett alkalmazások fejlesztéséhez. Nyílt forráskódú, TypeScriptre épülő fejlesztési platform. Jól integrált könyvtárak gyűjteménye, amelyek a funkciók széles körét fedik le. Jelenleg a Google és a Microsoft támogatja.

Vue

A Vue egy progresszív keretrendszer felhasználói felületek építésére. Más monolitikus keretrendszerekkel ellentétben a Vue-t az alapoktól kezdve úgy tervezték, hogy fokozatosan adoptálható legyen. A core library csak a nézeti rétegre összpontosít, és könnyen felvehető és integrálható más könyvtárakkal vagy meglévő projektekkel. Másrészt a Vue tökéletesen alkalmas kifinomult egyoldalas alkalmazások működtetésére is, ha modern eszközökkel és könyvtárakkal együtt használják. Nagyon jó dokumentációval rendelkezik.

Vizsgált technológia	REACT	ANGULAR	VUE
Open Source	Igen	Igen	Igen
Népszerűség	Legnépszerűbb	Népszerű	Népszerű
Támogatottság	Kiváló	Kiváló	Kiváló
Keretrendszer	Nem	Igen	Igen

4.1.4 Választott Frontend technológia

A Facebook által fejlesztett React libraryt fogom használni a fejlesztés során. Rengeteg hasznos dokumentáció található róla az interneten, melyek segítik a fejlesztést. A kódja pedig nagyon könnyen áttekinthető, karbantartható. Az átláthatósága pedig javítja a hibakeresést.

4.1.5 Használandó adatbázis – Microsoft Azure SQL Database

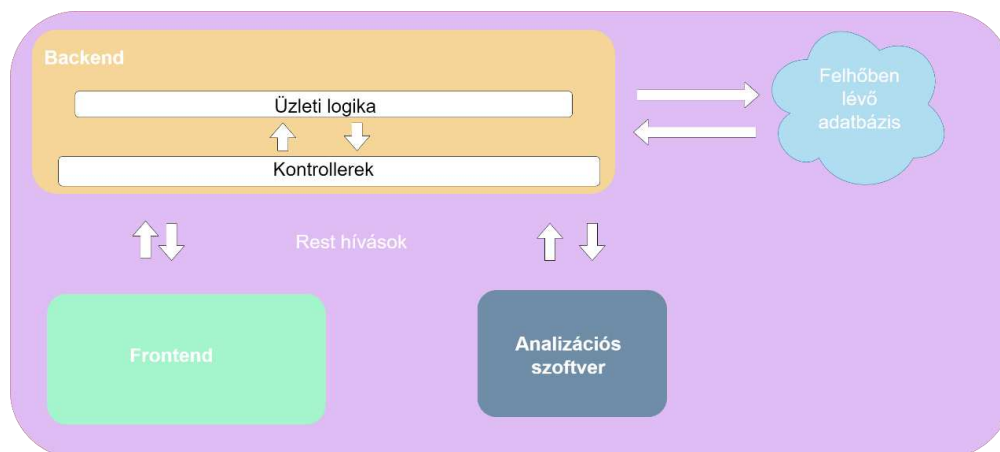
„Az Azure SQL-termékcsalád részét képező Azure SQL Database a felhőhöz készült intelligens, skálázható relációs adatbázis-szolgáltatás. Automatizált, mindig naprakész AI-alapú funkciókkal optimalizálhatja a teljesítményt és a tartósságot. A kiszolgáló nélküli számítási és rugalmas skálázású tárolási lehetőségekkel, amelyek automatikusan igény szerint skálázzák az erőforrásokat” [9]

4.1.6 Microsoft Azure SQL Database előnyei

- Microsoft Azure felhő nagyfokú rendelkezésre állást biztosít
- A Microsoft Azure nagy hangsúlyt fektet a biztonságra, és követi az észlelés, értékelés, diagnózis, stabilizálás és lezárás szabványos biztonsági modelljét.
- Skálázhatóság

4.2 Rendszer tervezése

A következő fejezetben a szakdolgozatomban elkészítendő rendszer tervezését vezetem le. Első lépésként meghatároztam a rendszerem főbb komponenseit, melyek az ábrán láthatóak (8. ábra).

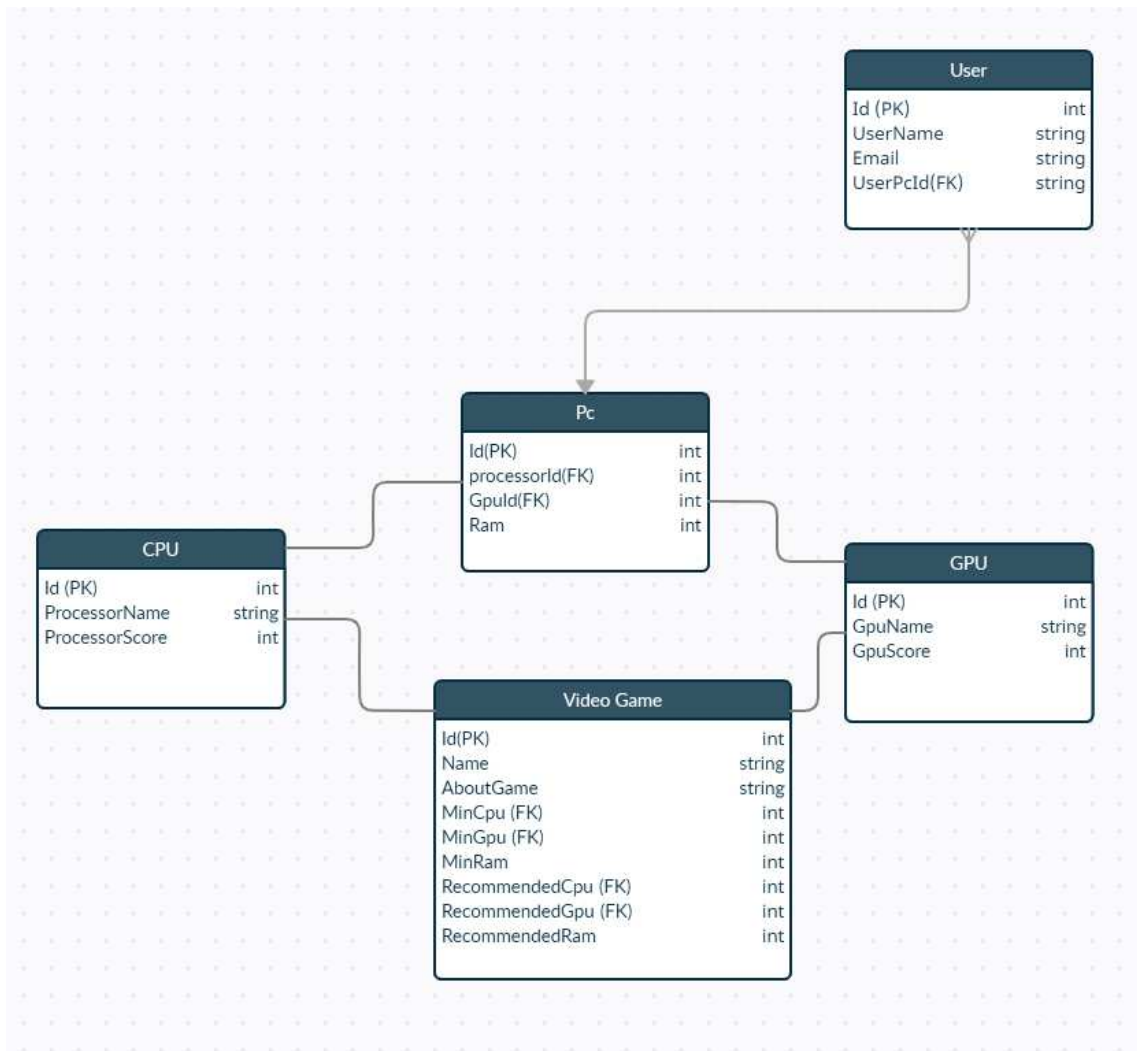


8. ábra A Rendszerterv ábrája

Az ábrán látható komponensek közül a backend, frontend és analízációs modul Rest API hívásokon keresztül kommunikál egymással. Adat módosítás esetén a szerverünk a felhőben lévő adatbázissal fog kommunikálni szintén API-n keresztül. Az Entity Framework teszi lehetővé, hogy .NET objektumok segítségével dolgozzunk egy adatbázissal.

Az Entity Framework egy objektum-relációs leképező (ORM), amely egy olyan eszköztípus, amely leegyszerűsíti a szoftverben lévő objektumok és a relációs adatbázis táblái és oszlopai közötti leképezést.

4.2.1 Az adatbázis tervezése



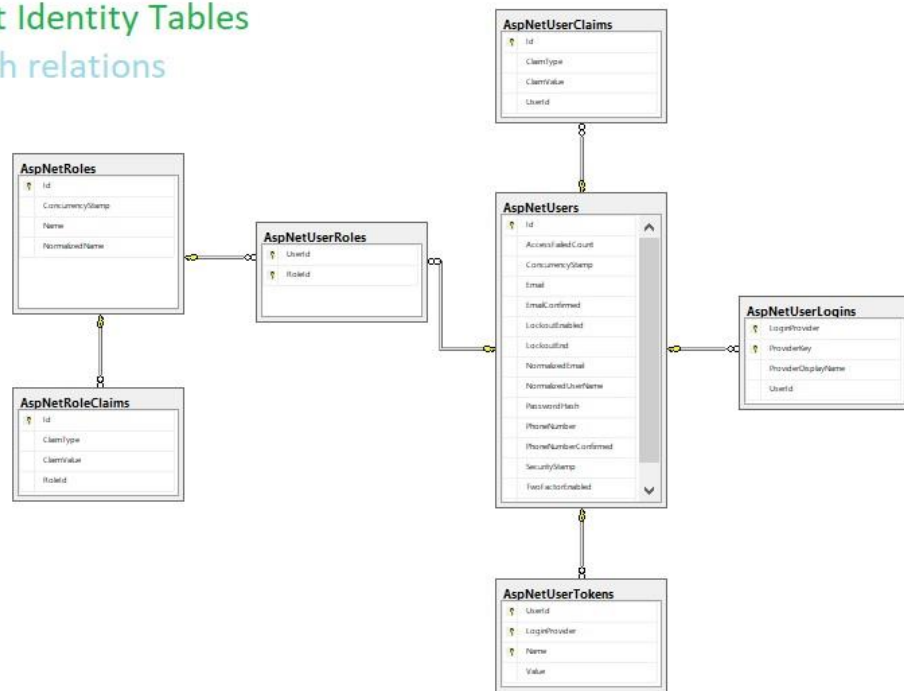
9. ábra Az adatbázis kezdetleges terve

- A User tábla az IdentityDbContext-ből származóAspNetUsers tábla kibővítése.
- A Pc tábla a felhasználók számítógépének harver komponenseit fogják tárolni, fontos, hogy ez One to One kapcsolat, egy felhasználónak egy számítógépe lehet.
- A Cpu, Gpu táblák tartalmazzák a web scraper segítségével összegyűjtött hardvereket és a hozzájuk tartozó adatokat.
- A VideoGame tábla tartalmazza az alkalmazásunkban található játékokat.

IdentityDbContext Class- Az identitáshoz használt Entity Framework adatbázis-kontextus alaposztálya.

„ASP.NET Core Identity is a membership system which allows you to add login functionality to your application. Users can create an account and login with a user name and password or they can use an external login providers such as Facebook, Google, Microsoft Account, Twitter and more.” [10]

Default Identity Tables with relations



10. ábra Alapértelmezett identity táblák

Mivel az alkalmazásunk DbContext osztályát az IdentityDbContext osztályból származtatjuk le, ezért az ábrán látható táblák automatikusan létre lesznek hozva, regenerálódnak (10. ábra).

Azért fontos, hogy az IdentityDbContextből származtassunk, mert az alkalmazásunkban különböző felhasználók lesznek, az alkalmazásunk funkció csak autentikálás, bejelentkezés után lesznek elérhetőek (a felhatalmazott userek számára).

4.2.2 Jogosultság RBAC vs ABAC

Egy RBAC-rendszerben az emberek "szerepük" alapján kapnak jogosultságokat és engedélyeket. Ezeket a szerepeket egy rendszergazda határozza meg, aki osztályokba sorolja az embereket osztályaik, felelősségi körük, szolgálati idejük és/vagy földrajzi

elhelyezkedésük alapján. A hozzáférés időtartama is eltérő lehet a különböző szerepkörök esetében. Az is lehetséges, hogy egy felhasználóhoz több szerepkör is hozzárendelhető

- Flat RBAC: Minden alkalmazottnak legalább egy szerepkör van hozzárendelve, de egyesek több szerepkörrel is rendelkezhetnek. Ha valaki egy új fájlhoz/forráshoz/szerverhez szeretne hozzáférni, először egy új szerepet kell kapnia.
- Hierarchical RBAC: A szerepek a ranglétra szintjei alapján kerülnek meghatározásra. A rangidős alkalmazottak a saját jogosultságaik mellett a beosztottjaik jogosultságaival is rendelkeznek.
- Constrained RBAC: Ez a modell bevezeti a feladatok szétválasztását (SOD). A SOD több felhasználó között osztja szét a feladat elvégzésének hatáskörét, csökkentve ezzel a csalárd és/vagy kockázatos tevékenységek kockázatát.
- Symmetric RBAC: Minden szervezeti szerepkör rendszeres felülvizsgálatra kerül. E felülvizsgálatok eredményeképpen jogosultságokat lehet kiosztani vagy visszavonni, és szerepeket lehet hozzáadni vagy eltávolítani.

Egy ABAC-környezetben a felhasználó bejelentkezésekor a rendszer különböző attribútumok alapján engedélyezi vagy elutasítja a hozzáférést. Ezek az attribútumok kapcsolódhatnak a következőkhöz:

- User- ABAC-kifejezésben a kérelmező felhasználót alanyként is ismerik. A felhasználói attribútumok közé tartozhat a kijelölés, a szokásos felelősségi körök, a biztonsági engedély, a részleg és/vagy a ranglétra.
- Accessed resource - Ez tartalmazhatja az erőforrás nevét és típusát (ami lehet fájl, kiszolgáló vagy alkalmazás), létrehozóját és tulajdonosát, valamint az érzékenységi szintet.
- Action- Mit próbál a felhasználó tenni az erőforrással? A vonatkozó attribútumok közé tartozhat az "írás", "olvasás", "másolás", "törlés", "frissítés" vagy "minden".

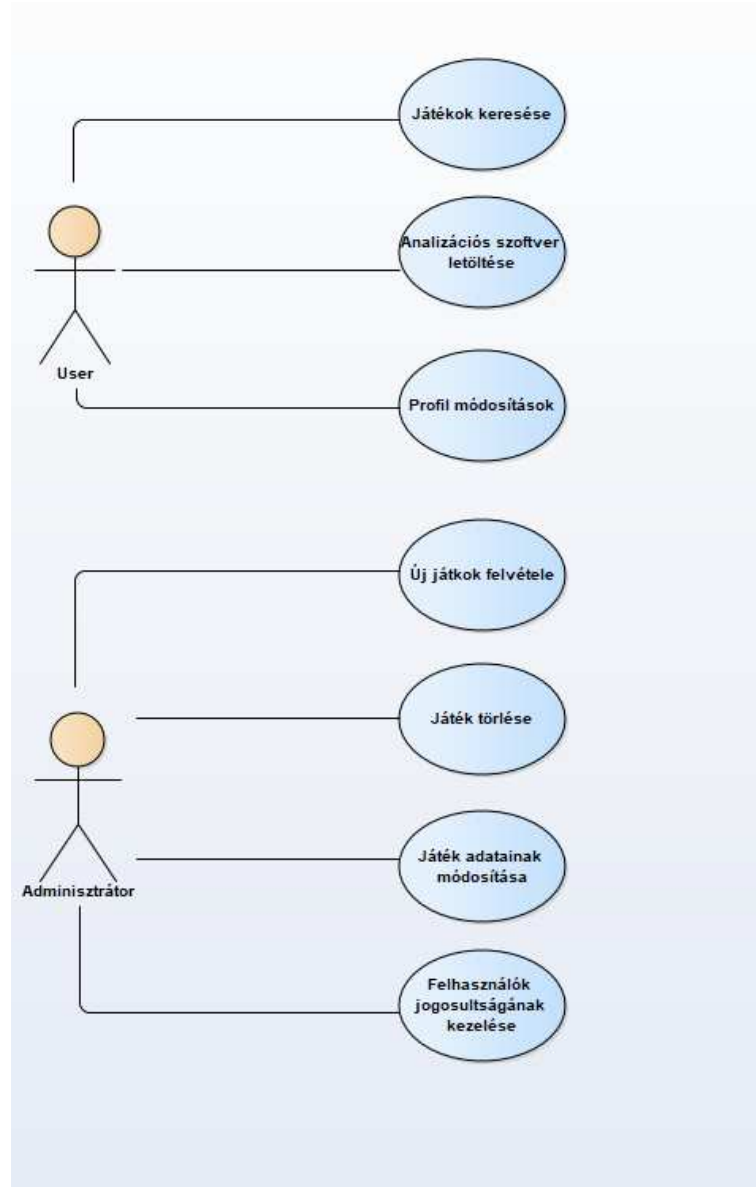
[11]

A fő érv pedig az RBAC mellett az, hogy a szerepkörök meghatározása és implementálása sokkal egyszerűbb és gyorsabb, mint az attribútumok hozzárendelése az egyénekhez. Valamint az alkalmazásban jól elkülöníthető rangok figyelhetőek meg.

4.2.3 Jogosultsági szintek az alkalmazásban

Az alkalmazásban két jogosultsági szint lesz megtalálható. Az egyik jogosultsági szint a „sima” user/felhasználó. A másik pedig az adminisztrátori jogosultsággal rendelkező felhasználó. Fontos, hogy a két rang teljesen elkülönülő, nem hierarchikus a jogosultsági szint.

Ami azt jelenti, hogy az Admin teljesen új/más jogosultságokkal rendelkezik, mint egy átlag felhasználó (tehát az átlag felhasználó jogosultságaival nem rendelkezik (11. ábra).



11. ábra Use case diagram a rendszerben található felhasználókról

Az átlag felhasználó bejelentkezést követően képes:

- Játékokra keresni az oldalon
- letölteni az analízis szoftvert
- profilját módosítani

Az adminisztrátori jogosultsággal rendelkező felhasználó pedig képes:

- Új játékot hozzáadni a weboldalhoz
- törölni játékot a weboldaltól
- a játék adatait módosítani
- felhasználók jogosultságát kezelni, módosítani

4.2.4 Hardver adatok összegyűjtése-web scraping

A Web scraping a strukturált webes adatok automatizált módon történő összegyűjtésének folyamata. Más néven webes adatszerzésnek is nevezik. A web scraping néhány fő felhasználási esete többek között az árfigyelés, az árinformáció, a hírfigyelés, a és a piackutatás.

A Web scraping (más néven web 'data extraction' vagy 'data scraping') megoldást kínál azok számára, akik automatizált módon szeretnék hozzáférni strukturált webes adatokhoz. A webes adatgyűjtés akkor hasznos, ha a nyilvános webhely, amelyről adatokat szeretne szerezni, nem rendelkezik API-val, vagy rendelkezik ugyan, de csak korlátozott hozzáférést biztosít az adatokhoz.

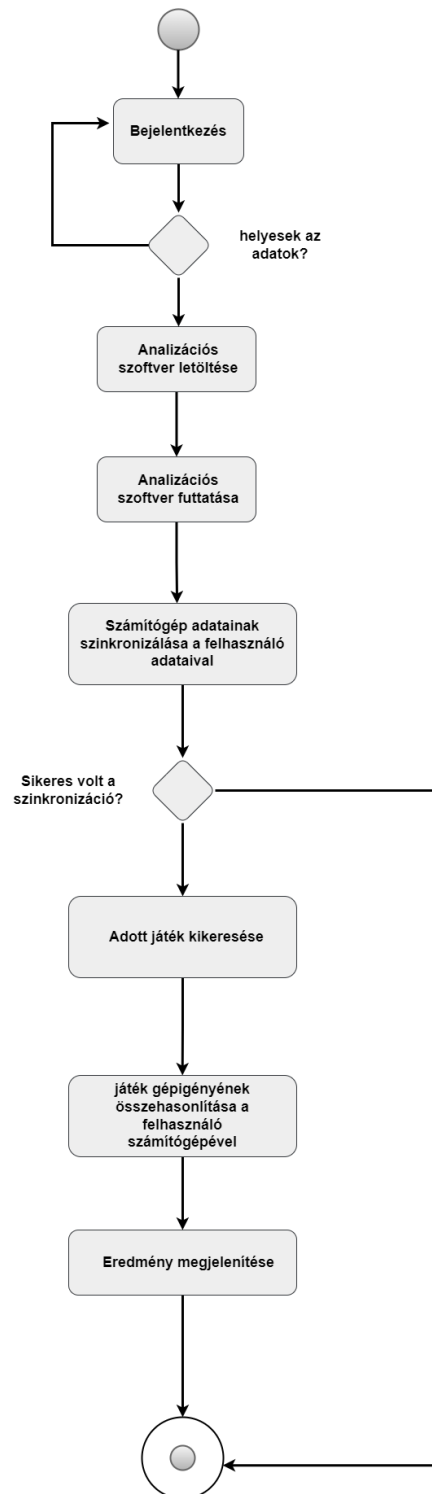
Egy web scraping szoftver segíthet automatizálni az adatok más honlapokról történő gyors és precíz kinyerésének eljárását. Emellett gondoskodhat arról is, hogy a kinyert adatok rendezettek legyenek, így könnyebben elemezhetőek és felhasználhatóak más projektekhez. Egy scraping eszköz jellemzően HTTP-kéréseket intéz a célwebhelyhez, és kivonja az adatokat egy oldalról. Általában olyan tartalmat elemez, amely nyilvánosan elérhető és látható a felhasználók számára, és amelyet a kiszolgáló HTML-ként megjelenít. [12]

Szakedolgozatom szempontjából azért fontos a web scraping technológia, mert a hardvereket így egy automatizált módon tudom összegyűjteni. Ennek a fő előnye az, hogy mivel az adatok összegyűjtése automatikus módon történik, így az adatbázist nem nekem kell manuálisan feltöltenem adatokkal, melynek több előnye is van. Egyrészt idő, energiaspórolás. Másrészt manapság évente több hardvert is piacra bocsátanak, amit a scrapinggel majd automatikusan feltudok venni az alkalmazásom adatbázisába.

4.2.5 Analizációs szoftver

Az analizációs alkalmazást a felhasználó (bejelentkezést követően) az oldalról tudja majd letölteni. Az analizációs szoftver fő célja az, hogy a számítógép hardver komponenseit gyorsan és precízen azonosítsa. Majd ezt követően a kinyert adatot visszaküldi a szervernek, ahol az feldolgozásra kerül. Mivel ennek az alkalmazásnak nem fontos a kinézete, ezért ez egy konzolos alkalmazás lesz. Ha a szerveren sikeres volt az adatfeldolgozás, azt követően a felhasználó profiljában megjelenik a sajátszámítógépének hardver komponensei.

A következő activity diagramon (12. ábra) bemutatom az analízációs szoftver letöltésének, futtatásának folyamatát a felhasználó szemszögéből.



12. ábra Activity diagram az analízációs szoftver folyamatáról

4.2.3 Az alkalmazásban található Api végpontok

Ebben a részben fogom felsorolni a backend által szolgáltatott végpontokat. Az alkalmazásban található végpontok a REST API architektúra szerint készülnek el. Az elkészítendő szoftverben számos végpont (endpoint) lesz megtalálható, azonban ezek többsége korlátozott hozzáférésű.

Publikusan elérhető végpontok a rendszerben (ezeket a végpontokat bárki elérheti):

- [PUT] /Auth/Register: Regisztráció
- [POST] /Auth/Login: Bejelentkezés
- [POST] /Auth/Logout

Végpontok melyeket a felhasználó ér el:

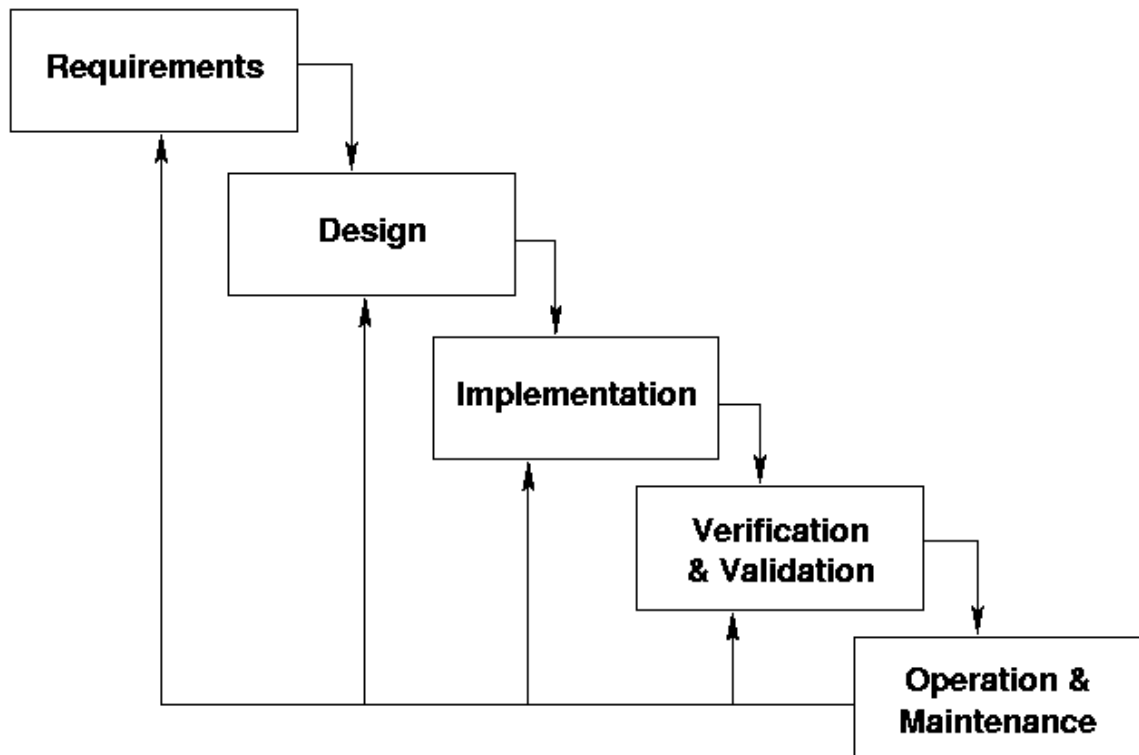
- [GET] /Game/GetGame/{GameName}: Egy játék visszaadása
- [GET] /Game/GetAllGame: Összes játék visszaadása
- [GET] /Game/Comparison: A játék gépigényének és a felhasználó számítógépének összehasonítása és ennek az eredménynek a visszaadása.
- [GET] /Profile/GetProfile: Az adott felhasználó profil adatainak visszaadása
- [PUT] /Profile/EditProfile: Profil adatok módosítása
- [GET] /Analyze/GetAnalyzeSoftware: Analizációs szoftvert letöltését biztosító endpoint.
- [POST] /Analyze/SendDetails: Az analízációs szoftver erre a végpontra küldi el a számítógépből kinyert adatokat (a számítógép hardver komponenseit).

Végpontok, melyeket az Admin jogosultsággal rendelkező felhasználók érhetnek el:

- [Post] /Admin/AddnewGame: Új játék felvételét biztosító endpoint
- [DELETE] /Admin/DeleteGame/{GameId}: Játék törlését biztosító végpont.
- [PUT] /Admin/EditGame/{GameId}: Játék szerkesztését biztosító végpont
- [PUT] /Admin/SetNewRole/{UserId}: Jogosultság változtatást biztosító endpoint

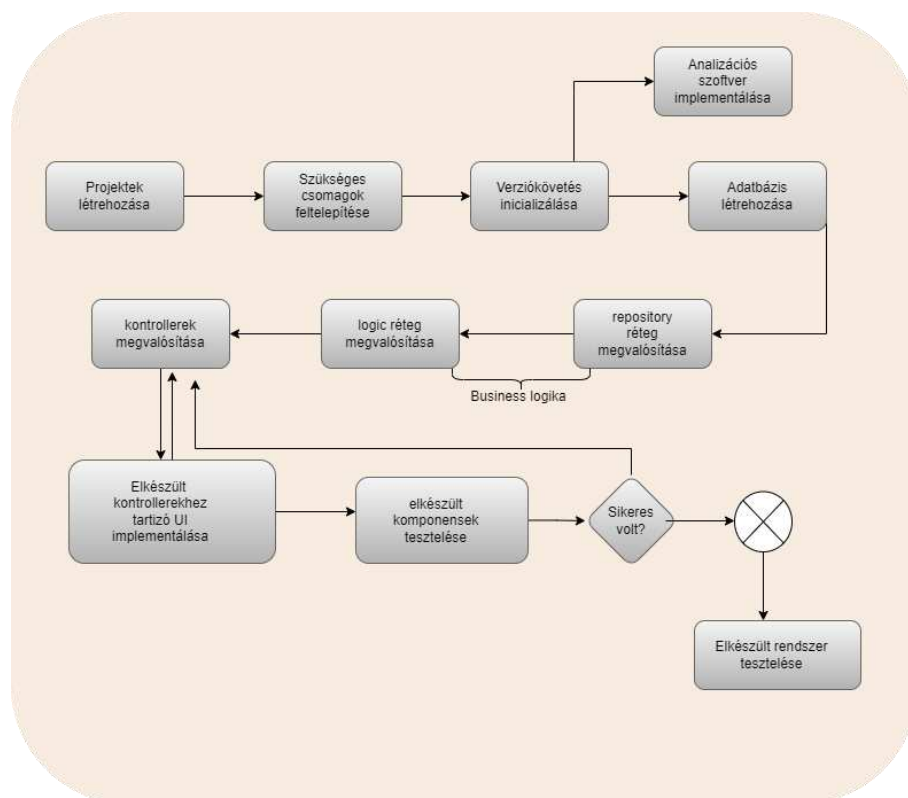
4.2.4 Fejlesztés tervezett menete

Megpróbáltam meghatározni, hogy a fejlesztés milyen folyamatokból fog állni, illetve, hogy a fejlesztési folyamatok milyen sorrendben fogják egymást követni. Fontos, hogy a fejlesztési folyamatot nem csapatban, hanem egyedül fogom végezni, így az agilis módszertanok (SCRUM, KANBAN) nem lesznek szükségesek. Tanulmányaim során megismerkedtem a waterfall(vízesés) modellel, mely tökéletesen alkalmazható erre a fejlesztési folyamatra annak ellenére, hogy ez egy régi projekt menedzsment elv (1970-ben találták ki) (13. ábra).



13. ábra Vizesés modell

A várható fejlesztés menete az alábbi ábrán (14. ábra) látható.



14. ábra A fejlesztés menete

5. Fejlesztési napló

Fejlesztési napló bevezetése

Az alkalmazásom, rendszerem fejlesztése során törekedtem, az általam meghatározott funkciók betartására. A fejlesztés során használtam külső csomagokat, melyek könnyítették a munkavégzést. Ezek között voltak olyan csomagok, amelyekkel már volt tapasztalatom, illetve voltak teljesen újak is. A következő részben pedig ezeket fogom elemezni.

5.1 Felhasznált csomagok

5.1.1 Swagger

A Swagger egy interfész-leíró nyelv a RESTful API-k JSON segítségével történő leírására. A Swagger használható RESTful webszolgáltatások tervezésére, létrehozására, dokumentálására.

A Swagger UI, amely a Swagger része, egy nyílt forráskódú eszköz, amely a Swagger specifikáció által generált API-kat dokumentáló weboldalt generál. Az API-knak ez a felhasználói felületű bemutatása felhasználóbarát és könnyen érthető, minden logikai összetettséget a képernyő mögött tartva. Ez lehetővé teszi a fejlesztők számára az általuk küldött API-kérelmek és a kapott eredmények végrehajtását és nyomon követését, így a fejlesztők, tesztelők és végfelhasználók számára kiváló eszköz a tesztelt végpontok megértéséhez. A Swagger UI az API-kat egy böngészőn belül reprezentálja, így intuitívabb, mint más eszközök, pl: Postman, SoapUI és mások. [13]

Magának a swaggernek létezik egy olyan konfigurációja is, amely lehetővé teszi az autentikációt. Ezáltal még egyszerűbb a hitelesítéshez kötött végpontoknak a tesztelése.

5.1.2 Nswag studio

Az NSwag egy nyílt forráskódú projekt a Swagger dokumentációjának előállítására és a Swagger felhasználói felület ASP.NET Core Web API-ba való integrálására. Ezenkívül az NSwag módszereket biztosít API C# klienskód generálására. Lehetővé teszi a fejlesztési ciklusok felgyorsítását és az API-változásokhoz való könnyű alkalmazkodást.

Számomra ez azért volt nagyon hasznos, mert egyetlen gombnyomás segítségével képes voltam legenerálni egy typescript fájlt, az API paraméterei és visszatérései alapján, amely tartalmazza a kliensoldali API hívásokat. Így megspóroltam ezeknek a megírását, hamar létrejöhett az API kommunikáció a backend és a frontend között.

5.1.3 Selenium

A Selenium egy nyílt forráskódú könyvtár, amely lehetővé teszi a böngészés automatizálását kód segítségével. A Selenium használatához két dologra volt szükségem: egy Selenium WebDriverre, amely kölcsönhatásba, interakcióba lép a böngészővel, és a Selenium könyvtárra, amely összeköti a kódukát a Selenium WebDriverrel. Ezt a csomagot a web scraping miatt használtam fel.

5.1.4 Automapper

Az AutoMapper a C# nyelvben egy olyan könyvtár, amelyet az adatok egyik objektumról a másikra történő leképezésére használnak. Két objektum között leképezőként funkcionál, ezen kívül az egyik objektumtípust átalakítja egy másik objektumtípussá.

Számomra ennek a csomagnak a használata azért volt fontos, mert az alkalmazásomban használok DTO-kat(Data transfer object) és ennek a csomagnak a segítségével könnyen, átláthatóan megtudtam oldani az objektumok közötti leképezést.

5.1.5 CloudinaryDotNet

A képek tárolásához szükségem volt a CloudinaryDotNet nevezetű csomagra. A Cloudinary egy SaaS (Software-as-a-Service) megoldás a webes vagy mobilalkalmazások médiakészleteinek felhőben történő kezelésére. Végponttól végpontig tartó megoldást kínál minden kép- és videóigényére, beleértve a feltöltést, a tárolást, az adminisztrációt, az átalakítást és az optimalizált kézbesítést. A média feltöltése, feldolgozása és kézbesítése a Cloudinary szerverein történik, amelyek automatikusan skálázódnak a nagy terhelés és a kiugró forgalom kezeléséhez. [14]

Így ez egy ideális szolgáltatás volt nekem a képek tárolására.

5.1.6 Material UI

A Material-UI egyszerűen egy könyvtár, amely lehetővé teszi, hogy különféle komponenseket importáljunk, valamint használjunk felhasználói felület létrehozásához a React alkalmazásainkban. Ezzel jelentős időt takaríthatunk meg, hiszen a fejlesztőknek nem szükséges mindent a nulláról megírniuk. A komponensek megfelelnek a mai trendeknek, megjelenéseknek.

5.2 Hardverek kinyerése a számítógépből- Analizációs szoftver

A hardverek kinyeréséhez egy konzolos alkalmazást fejlesztettem. Az alkalmazás felépítése egyszerű. Az adatok kiolvasása a keretrendszer által nyújtott ManagementObjectSearcher osztály segítségével történik. Az osztály segítségével képesek vagyunk management objectek gyűjteményének lekérdezésére egy megadott feltétel alapján. Ez az osztály az egyik leggyakrabban használt management objectek lekérdezéséhez. Az osztály még használható például a rendszer összes lemezmeghajtójának, hálózati adapterének, folyamatának és még sok más kezelési objektumnak a felsorolására.

Az analizációs szoftverben a lekérdezést úgy írtam meg, hogy a processzor, videokártya és RAM mennyiséget kérdezi le a felhasználó számítógépből, hiszen számunkra csak ezek az adatok a mérvadóak, az alkalmazás másik része ezekkel az adatokkal fog dolgozni. Ez egy teljesen automata folyamat, amely rögtön bejelentkezést követően elindul.

Az alkalmazásban található API kommunikáció is, hiszen elsőnek be kell jelentkeznie a felhasználónak, az adatok kinyerése csak ezután történik meg. Ha az adatok kinyerése sikeres volt, akkor pedig elküldjük a szervernek, ahol a kinyert adat feldolgozásra kerül.

A konzolos alkalmazást publisholni kellett, hiszen később a felhasználónak egy önmagában futtatható exe-t kell letöltenie. Ezért publisholásnál a deployment mode-ra self contained-et választottam. Ez egy olyan, alkalmazást hoz létre, amely tartalmazza a .NET Runtimeot és könyvtárakat, valamint az alkalmazást és függőségeit. Így az alkalmazás felhasználói futtathatják azt olyan gépen is, amelyen nincs telepítve a .NET Runtime.

Az analizációs szoftver fejlesztése közben, szerencsére nem alakult ki semmilyen implementációs nehézség, a követelményrendszerben meghatározott funkciókat sikerült teljesíteni.

5.3 A fejlesztés menete

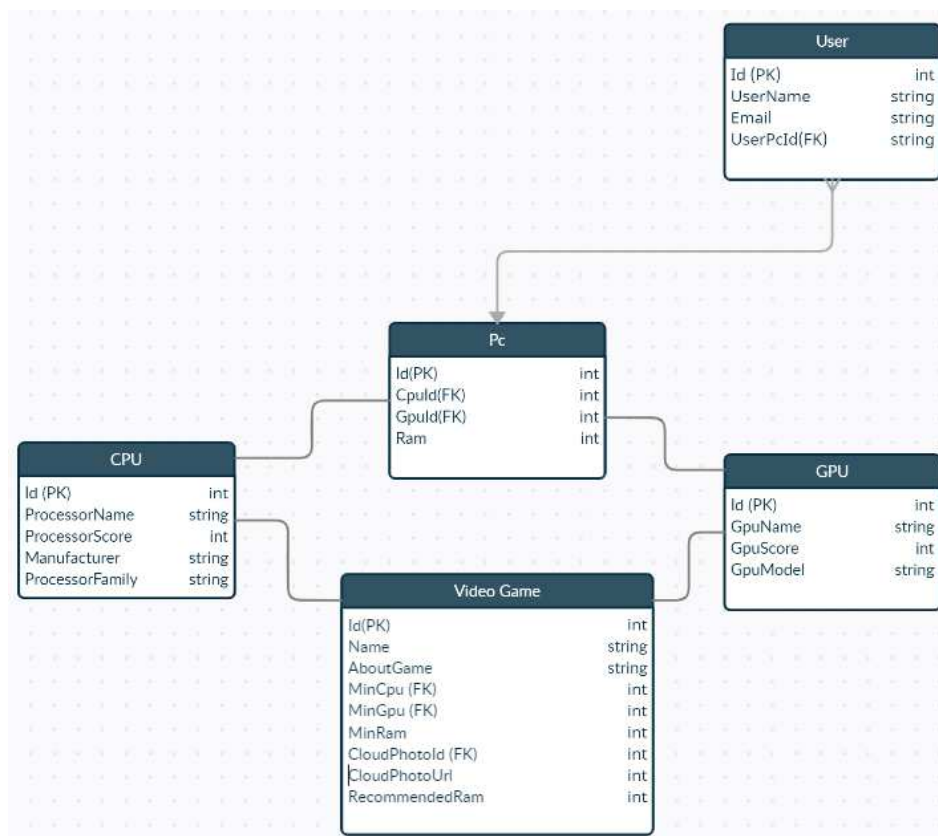
5.3.1 Adatbázis létrehozása

Szerver oldalon az első lépés az entitások létrehozása volt, a korábban megfogalmazott adatbázis követelmények alapján.

Fontos megemlíteni, hogy az alkalmazás a legújabb .NET 6 CORE-ban készült. Amely miatt néha könnyebb, néha pedig nehezebb dolgom volt. Meglepetésként ért, amikor láttam, hogy újításként már nem tartalmazza a *Startup.cs*-t, hanem már csak *Program.cs* van és ez kezdetben nehézséget okozott, a szintaktikai különbségek miatt.

Az adatbázist az *Entity-Framework* segítségével hoztam létre. Létrehoztam az adatbázis entitásokat, melyek közül párat módosítanom, bővítenem kellett. Ilyen módosítás volt például a videojáték entitásnál megjelenő photoId és PhotoURL mezők. Amint készen lettem az entitásokkal, kiadtam a megfelelő migration parancsokat és létrejött az adatbázisom. Az adatbázisom ellenőrzésére, adatainak megtekintésére, sql- el történő adat módosításokra a Microsoft által készített SQL Server Management Studiot (SSMS) használtam, amely segítette a munkámat.

Az adatbázis kiegészített, frissített verziója az alábbi képen látható:



15. ábra Az adatbázis frissített verziója

5.3.2 Scraper megírása

Következő fejlesztési feladatom a scraper megírása volt. A scraper segítségével gyűjtöttem össze a processzor, videokártya adatokat, amelyekkel később feltöltöttem az adatbázisomat. Az általam írt scraper szoftver adathalmaz forrása, amelyből dolgozik az a passmark software nevezetű oldal. Ez az oldal tartalmazott minden olyan fontos információt, amelyre szükségem volt az alkalmazásomnak. Viszonylag egyszerűen kinyerhetőek róla az adatok, illetve megbízható statisztikákkal rendelkezik. Az adatok beazonosítására XPATH-t használtam, Az XPATH segítségével navigálhatunk az XML-dokumentum elemei és attribútumai között. Ennek segítségével tökéletesen beazonosíthatóak voltak a táblázatban szereplő adatok.

Az első komoly probléma a scraper megírása során jelentkezett. Az analyzer programban felhasznált ManagementObjectSearcher osztály és az internetről származó processzor, videokártya elnevezések teljesen másképp találhatóak meg. Amely későbbiekben a rendszer működésében hatalmas problémát okozhatnak, hiszen a rendszerünk csak olyan hardverekkel tud dolgozni, amelyeket megtalál az adatbázisában.

Két megvalósítási lehetőség állt rendelkezésre a probléma megoldásához. Az egyik az, hogy a scraper által begyűjtött adathalmazt módosítom olyan módon, hogy illeszkedjen a ManagementObjectSearcher osztály által kigyűjtött adathalmazhoz vagy a későbbiekben hívom meg az összehasonlító logikát és futtatom mindig, amikor az analyzer adatokat küld a szervernek.

Végül amellet döntöttem, hogy amikor web scrapinggel kinyertem az adatokat, konvertálom őket, és így mentem el az adatbázisba. A későbbiekben az adathalmaz szűkítése ezért egyszerűbb lesz, ami később futási időben sokkal jobb eredményt produkál. Viszont negatívum az, hogy így maga a scrapelés művelet tovább tart, hiszen minden adatra el kell végezni az átalakítást, majd el kell menteni az adatbázisba. Ez processzoroknál majdnem 4000 adatot, entitást jelent, videokártyáknál pedig több mint 2300-at. Tehát viszonylag nagy adathalmazról beszélhetünk. Fontos, hogy a scrapelés többször végezhető művelet az admin által, bármikor végezhet adatgyűjtést az admin, így figyelni kellett arra, hogy duplikált adat sem kerülhet be az adatbázisunkba a scraping művelet során.

Az adatokat pedig úgy kellett konvertálni, hogy későbbiekben az adathalmaz csökkentése, szűrése egyszerűbb folyamat legyen ezért pedig logikailag daraboltam az adathalmazt.

Az adathalmaz szűkítése azért nagyon fontos, mert a több ezer adatból pontosan egy elemre kell leszűkíteni akkor, amikor a felhasználó az analyzer program segítségével hozzáadja számítógépét a rendszerünkhöz. Hiszen a rendszerünk csak olyan hardverekkel tud dolgozni, amelyek megtalálhatóak az adatbázisban. Ha esetleg olyan hardver konfiguráció érkezne az analyzer programból, ami nem található meg az adatbázisban, azt a felhasználónak jelzi a rendszer.

Az adathalmaz szűkítése logikailag történik, a processzor, videokártya elnevezési séma alapján. Például a rendszer szűr gyártó, típus és család alapján. Majd a legvégén REGEX segítségével történik a pontos megtalálás. A REGEX (reguláris kifejezés) egy olyan karaktersorozat, amely egy keresési mintát határoz meg.

Ha az analyzer program által küldött hardverek (processzor, videokártya) megtalálhatóak az adatbázisban, akkor mentésre kerül az adott számítógép konfiguráció, amellyel későbbiekben a rendszer képes dolgozni. Minden felhasználóhoz egy számítógép tartozhat, viszont bármikor futtathatja az analízis alkalmazást, duplikált adat nem keletkezik ilyen esetben sem.

5.3.3 Hátralévő CRUD funkciók megírása

Amint végeztem a scraper logika megírásával és sikeresen megvalósítottam a számítógép hozzáadása funkciót, lefejlesztettem a többi entitásra a CRUD funkciókat (létrehozás, olvasás, frissítés és törlés). Ilyen például a videojáték hozzáadása funkció, melyet az admin tud elvégezni. Videojáték hozzáadásakor kötelező megadni a minimum hardver követelményeket, melyek FOREIGN KEY hivatkozások lesznek a CPU, GPU táblákra. Ezek a táblák között tehát így fog felállni a kapcsolat, a videojátéknak lesz egy mincpu és egy mingpu mezője. Ezeknek a mezőknek a segítségével tudjuk beazonosítani a játék hardver követelményeit.

5.3.4 Autentikáció

A rendszerbe regisztráció szükséges, enélkül semmit nem ér el a felhasználó. Regisztráció folyamán kötelező megadni az email címet és a jelszót. Ha ezeket megadta a felhasználó, azután bekerül az adatbázisba, ahol az érzékeny adatokat például a jelszót biztonságos, elhashelt formában tárolom. Regisztráció közben szerver oldalon az email címre még történik egy validálás, hogy megfelelő formátumban adta-e meg a felhasználó (tényleg email címet írt e be). Ha nem megfelelő formátumban adta ezt meg, hibát jelez a rendszer.

Ha sikeres volt a regisztráció, a felhasználó ezt követően azonnal be tud jelentkezni a rendszerbe az előbb megadott email cím-jelszó párossal. A bejelentkezés során a szerver összehasonlítja a megadott adatokat az adatbázisban található adatokkal. Felhasználónévénél ez egy egyszerű ellenőrzés viszont jelszónál másképp történik a validálás. Konkrét jelszavakat (plain textben), mint fentebb is említettem nem tárolok az adatbázisban, csak azoknak a hasheit, ezzel biztosítva az érzékeny adatok biztonságát. A rendszer bejelentkezés során is hashelni fogja a megadott jelszót és a hasheket fogja összehasonlítani. Ha azok is megegyeznek, akkor a szerver visszaküld egy tokenModelt, mely tartalmazza magát a token-t illetve annak lejárat dátumát. Ezt követően a felhasználó hozzáfér a rendszer funkcióihoz, de kizárólag azokhoz, amelyekhez jogosultsága van. A token payload részét kiegészítettem úgy, hogy megtalálható legyen a felhasználó email címe, jogosultsága. Magát a token-t kliens oldalon local storageban fogja tárolni a rendszer.

5.3.5 Összehasonlító logika

A rendszer fő funkciója az, hogy képes megmondani azt, hogy mely játékok képesek futni a felhasználó számítógépén. Ehhez szükséges az, hogy a felhasználó minimum egyszer futtassa az analyzer programot a számítógépén, hiszen a számítógép konfigurációja így fog bekerülni az adatbázisba. Ha ez megtörtént akkor a főoldalon olyan játékok fognak kilistázódni a felhasználó számára, melyek minimum hardver követelményei alacsonyabbak a felhasználó számítógépében megtalálható hardverektől.

Az összehasonlítás szerver oldalon az adatbázis táblák kapcsolata miatt, egyszerűen elvégezhető. Minden játéknak van egy mincpu és egy mingpu mezője, melyek foreign key hivatkozások a cpu és gpu táblára. Az utóbbi két táblába fontos, hogy scrapelés során egy benchmark pontszámot is elmentett a szoftver, ez alapján pedig könnyen rangsorolhatóak a hardverek. A PC tábla felépítése is hasonló ebben is megtalálhatóak a foreign key hivatkozások a cpu gpu táblákra. Az adatbázis megfelelő felépítése pedig nagyban egyszerűsíti az összehasonlító logika elvégzését.

A rendszer megvizsgálja, hogy a számítógépben található processzor, videokártya pontszáma nagyobb-e a videojáték entitásnál megadott hardverek pontszámánál. Ha ez az állítás igaz, tehát nagyobb, akkor képes futni a felhasználó számítógépén a játékszoftver. A processzor és videokártya összehasonlítás mellett, még történik egy RAM mennyiség ellenőrzés is. A rendszer itt azt ellenőrzi, hogy a számítógépben található ram mennyiség meghaladja-e a videojátéknál feltüntetett RAM mennyiséget.

5.3.6 Képek feltöltése

Képek feltöltéséhez a Cloudinary szolgáltatást használom. A megvalósításhoz a ClodinaryDotNet nevezetű csomagra volt szükségem, mely tartalmazza az adat feltöltéshez szükséges metódusokat. A csomag használata nagyon egyszerű volt, a részletes dokumentációnak köszönhetően. Azért a Cloudinary-re esett a választásom, mert ez egy ingyenes szolgáltatás, amely bőven kielégíti az én alkalmazásomnak az igényeit. A Cloudinary háromféle árkategóriával rendelkezik, melyből az ingyeneset használom. Amely lehetővé teszi 25GB feltöltését. Ez a tárhely bőven elég a videojátékokhoz tartozó képek tárolásához. A Cloudinary képrekezelő API-ja nem korlátozza a feltöltések, átalakítások és letöltések (nézetek) számát. Egyetlen korlátozás a tárhely mennyisége, melyet fentebb is említettem. Cloudinarybe feltölthető képfájlok maximális mérete 10 MB.

A videojáték entitásomat ki kellett egészítenem pár mezővel, melyek szükségesek voltak a képek eltárolásához. Az egyik ilyen mező a CloudPhotoId amely egy foreign key hivatkozás, a rendszer majd ez alapján tudja beazonosítani, hogy az adott videojátékhoz melyik kép tartozik. A másik mező a CloudPhotoUrl amely a felhőben lévő kép címére mutat. A kapcsolat így fog létrejönni az én rendszerem és a külső szolgáltatás között.

5.3.7 Felhasználói felület fejlesztése

A felhasználói felület fejlesztéséhez a korábban is említett MaterialUI komponenseket használtam. Azért erre a könyvtárra esett a választásom, mert rengeteg komponenssel rendelkezik. Ezek a szegmensek mindegyike reszponzív, azaz megfelelően alkalmazkodnak a különböző képernyőméretekhez. Ez egy nagyon fontos szempont volt az alkalmazás tervezése során és a MaterialUI ezt az igényt is kielégíti. A könyvtár használata egyszerű volt, a részletes dokumentáció és interneten megtalálható segédanyagok miatt.

A felhasználói felület létrehozásánál ügyeltem a letisztult design megtartására, mely ezekkel a komponensekkel nagyon egyszerű volt. A frontend fejlesztése párhuzamosan zajlott a backend fejlesztésével, így amikor elkészültem szerver oldalon az adott funkcióval, lefejlesztettem a hozzátartozó felhasználói felületet. A frontend projektet is próbáltam minél áttekinthetőbb módon fejleszteni. Törekedtem a kód újra felhasználhatóságára így igyekeztem többször felhasználható komponenseket létrehozni, melyeket a Components mappán belül tároltam. Ilyen volt például az Admin felületeken megjelenő data grid. A követelmény szerint fontos volt a sötét megjelenítés, mely egyszerűen megvalósítható volt a MaterialUI dark mode-val.

Bejelentkezés utána felhasználó kap egy token-t. Ezt a token-t az alkalmazás local storage-ban tárolja. A localStorage egy webes tárolóobjektum, amely lehetővé teszi a JavaScript webhelyek és alkalmazások számára, hogy kulcs-érték párokat tároljanak a böngészőben lejáratí dátum nélkül. Ez azt jelenti, hogy az adatok túlélnek az oldalfrissítést (sessionStorage) és még a böngésző újraindítását is. A böngészőben tárolt adatok akkor is megmaradnak, ha a böngészőt bezárják.

Az alkalmazásban megtalálható ikonokhoz pedig a Material Icons-t használok.

6. Tesztelés

A következő fejezetben a rendszerem tesztelését fogom bemutatni.

6.1 Egység tesztek

Egység tesztelés megvalósításához szükségem volt az NUnit nevezetű keretrendszerre. Azért ezt a keretrendszert választottam, mert ebben van a legtöbb tapasztalatom, illetve .net – en belül ez az egyik legelterjedtebb választási lehetőség. A megfelelő tesztelési környezet kialakításához mockolnom kellett az adatbázist, ehhez a Moq nevezetű csomagot használtam.

A különböző tábláknak a mockolásait kiszerveztem builder osztályokba, ezek az osztályok felelősek a megfelelő mock létrehozásáért. Ezáltal sokkal könnyebb volt áttekintennem a teszt projektet.

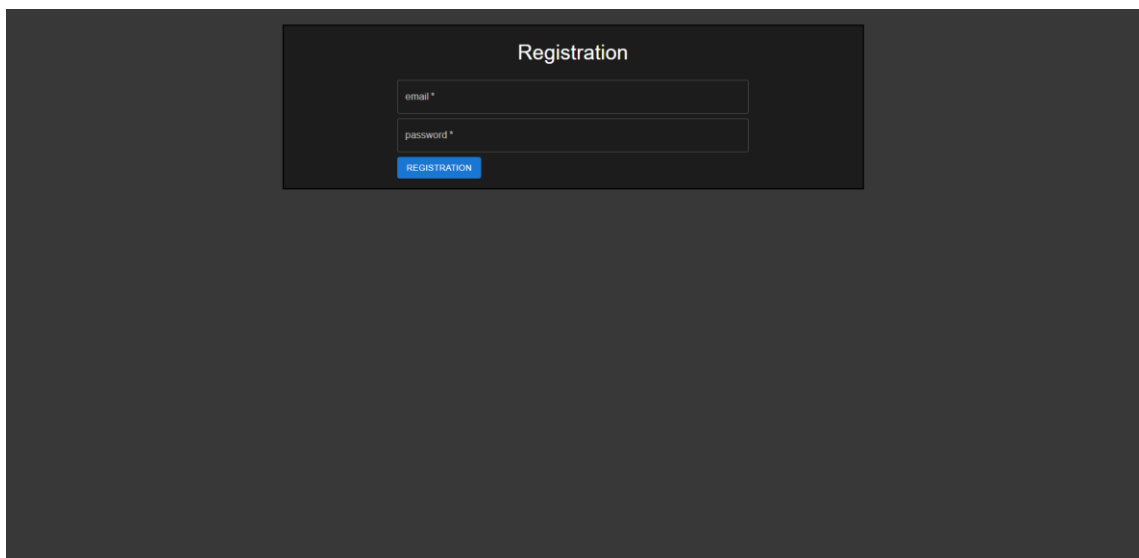
Teszt	Elvárt Kimenetel	Eredmény
Bejelentkezés	Bejelentkezés után a felhasználó kapjon egy token, a jelszó ellenőrzésért felelős logika csak egyszer fusson le	Sikeres
Felhasználó lekérdezése ID alapján	A felhasználók lekérdezése egyszer történjen meg az eredmény pedig legyen egyenlő azzal, amit elvárunk	Sikeres
Összes felhasználó lekérdezése	A felhasználók lekérdezése egyszer történjen meg, az eredményül kapott lista hossza egyezzen meg az elvárttal.	Sikeres
Processzor lekérdezése ID alapján	A GetOne metódus egyszer fusson le, az eredmény elgyen egyenlő az elvárttal.	Sikeres
Összes processzor lekérdezése	A GetAll metódus egyszer fusson le, az eredményül kapott lista hossza legyen egyenlő az elvárttal.	Sikeres
Processzorok begyűjtése(scraping)	A ScrapeCpus metódus egyszer fusson le.	Sikeres
Videokártya lekérdezése ID alapján	A GetOne metódus egyszer fusson le, az eredmény legyen egyenlő az elvárttal.	Sikeres
Összes videokártya lekérdezése	A GetAll metódus egyszer fusson le, az eredményül kapott lista hossza egyezzen meg az elvárttal.	Sikeres
Videokártya adatok begyűjtése(scraping)	A ScrapeGpus metódus egyszer fusson le.	Sikeres
Videójáték hozzáadása	Az Add metódus egyszer fusson le,	Sikeres
Videójáték lekérdezése ID alapján	A GetOne metódus egyszer fusson le, az eredmény egyezzen meg az elvárttal.	Sikeres
Összes videójáték lekérdezése	A GetAll metódus egyszer fusson le, az eredményül	Sikeres

	kapott lista hossza egyezzen meg az elvárttal.	
Videojáték módosítása	Az Update metódus egyszer fusson le.	Sikeres
Videojáték törlése	A Delete metódus egyszer fusson le	Sikeres
Videojáték többes törlés	A MultipleDelete metódus egyszer fusson le.	Sikeres
Analyzer futtatása meglévő pc esetén	Az Add metódus egyszer sem, az Update egyszer fusson le	Sikeres
Futtatható videojátékok lekérdezése	A FindByNameAsync, GetPcByUserId, GetAll metódusok egyszer fussanak le.	Sikeres

7. Az elkészült alkalmazás

A következő fejezetben szeretném bemutatni az elkészült alkalmazást és annak funkciót. Külön fogom bemutatni jogosultság szerint a funkciókat.

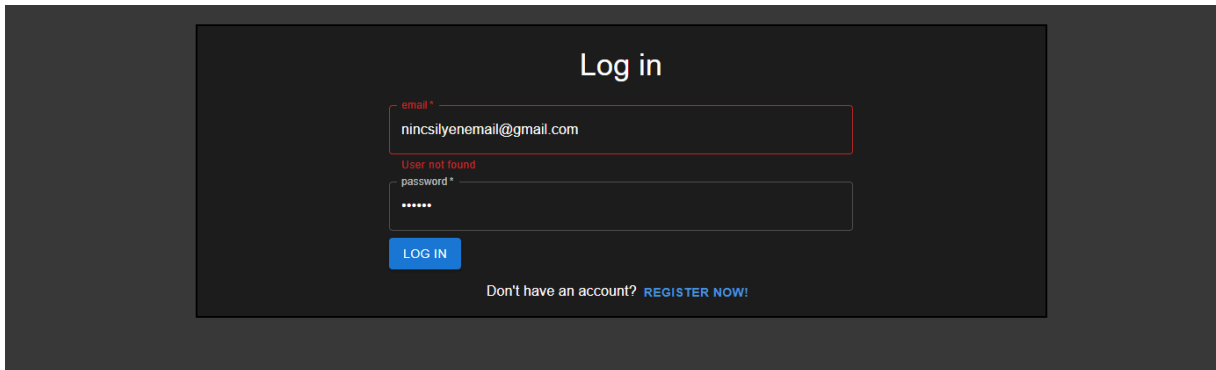
Ha nincsen fiókunk, akkor első lépésként be kell regisztrálni az alkalmazásba, ezt az alábbi felületen tudjuk megtenni.



The image shows a registration form titled "Registration" centered on a dark background. The form contains two input fields: "email *" and "password *". Below these fields is a blue button labeled "REGISTRATION".

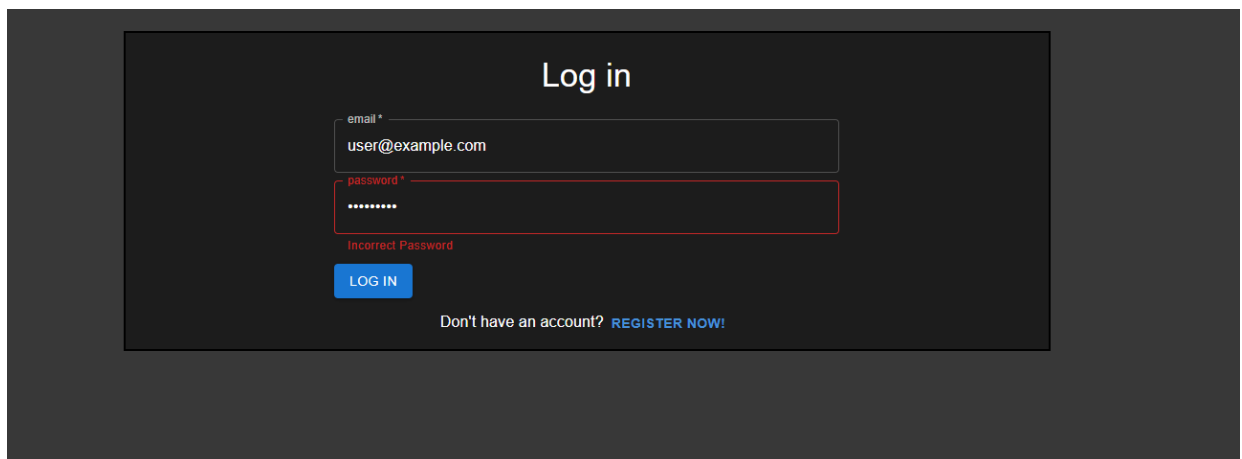
16. ábra Regisztrációs felület

Amint beregisztráltunk az alkalmazásba, képesek vagyunk bejelentkezni a megfelelő felhasználónév jelszó párossal. Ha rossz adatokkal próbálnánk belépni, arról a rendszer értesíti a felhasználót.



The screenshot shows a dark-themed login interface. At the top, the text "Log in" is centered. Below it, there are two input fields. The first field is labeled "email *" and contains the text "nincsilyenemail@gmail.com". Below this field, the error message "User not found" is displayed in red. The second field is labeled "password *" and contains six dots. Below the password field is a blue button labeled "LOG IN". At the bottom of the form, there is a link that says "Don't have an account? REGISTER NOW!".

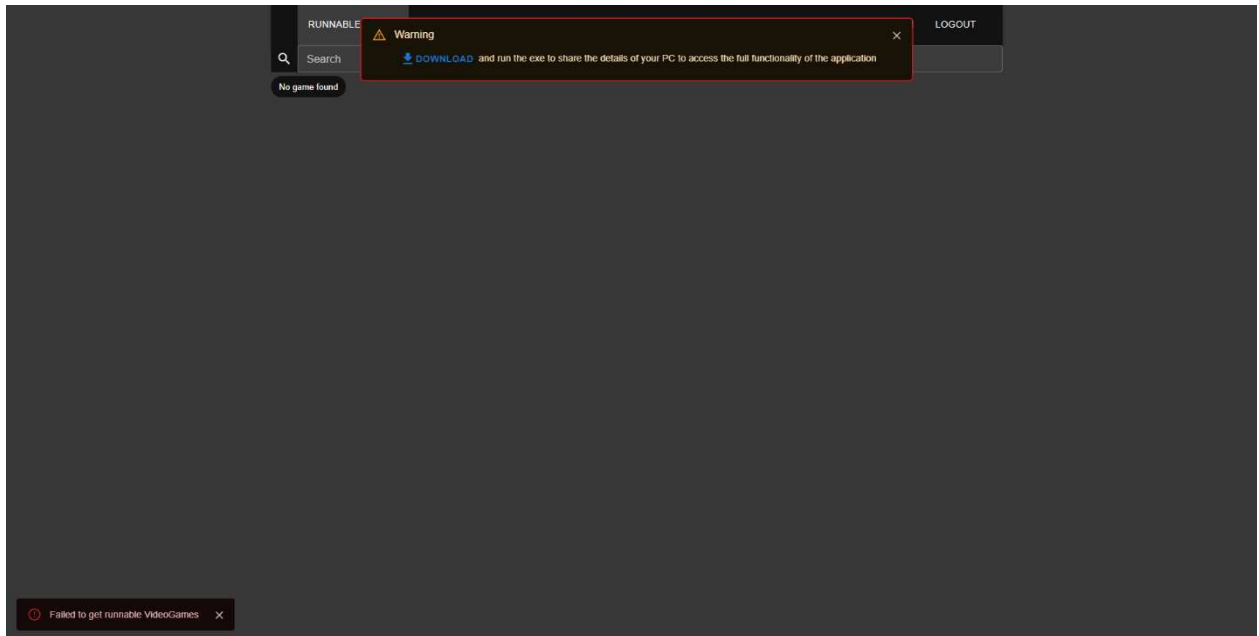
17. ábra Rendszer által nem ismert email címmel való belepés



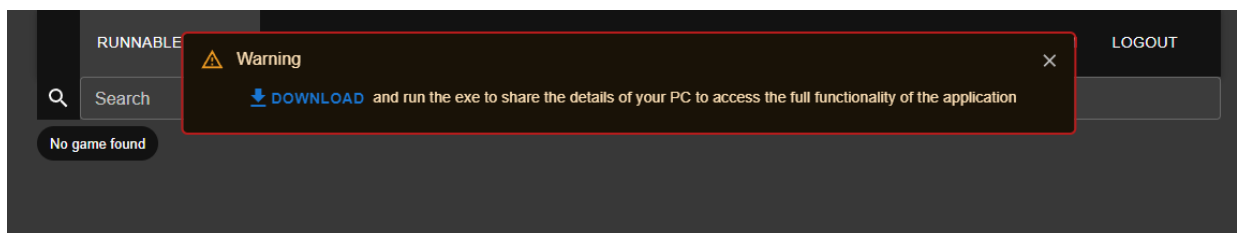
The screenshot shows a dark-themed login interface. At the top, the text "Log in" is centered. Below it, there are two input fields. The first field is labeled "email *" and contains the text "user@example.com". The second field is labeled "password *" and contains six dots. Below the password field, the error message "Incorrect Password" is displayed in red. Below the password field is a blue button labeled "LOG IN". At the bottom of the form, there is a link that says "Don't have an account? REGISTER NOW!".

18. ábra Rendszer által ismert email cím, de rossz jelszóval való belépési próbálkozás

Sikeres belépés után a felhasználót a rendszer átirányítja a főoldalra, ahol azok a játékok listázódnak ki, amelyeket képes futtatni a számítógépe. Ha új felhasználóval lépünk be, akkor nem lesz a számítógép konfigurációja megtalálható a rendszerben, erről a felhasználó kap egy figyelmeztetést. Ezt az alábbi képen ezt látjuk.

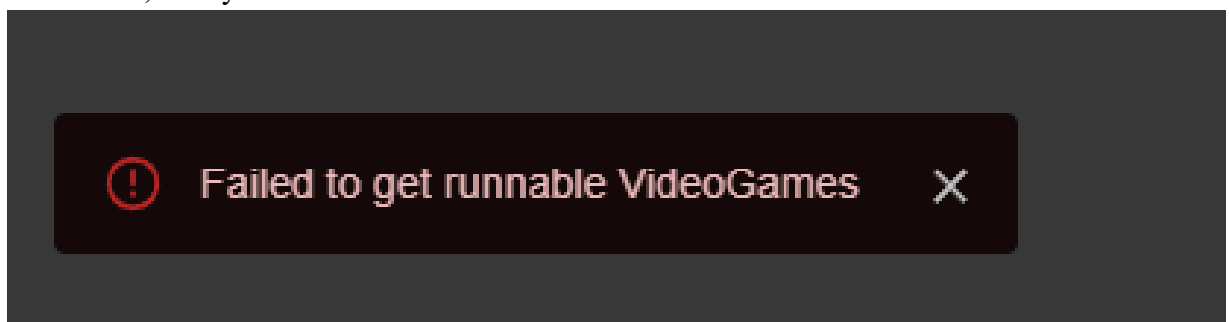


20. ábra A főoldal számítógép konfiguráció nélkül



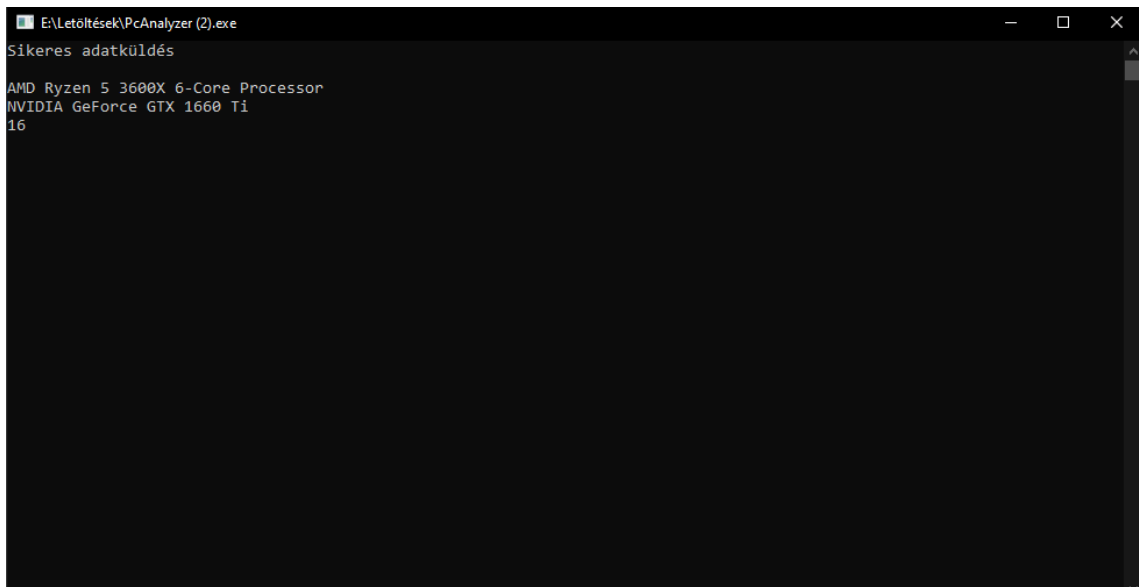
19. ábra Figyelmeztető üzenet a felhasználó számára

Ha a felhasználóhoz nincsen számítógép konfiguráció rendelve, akkor a rendszer nem tudja betölteni a számítógépén működő játékszoftvereket, erről is értesítést küld a rendszer, amelyet a mellékelt ábra mutat.



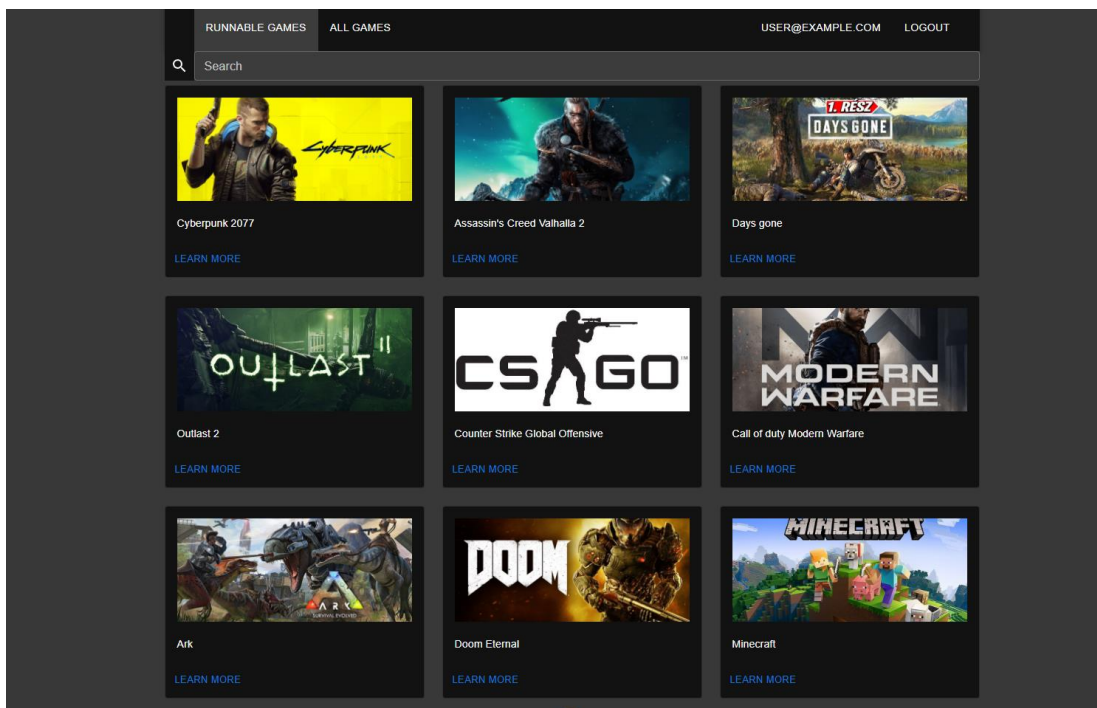
21. ábra Figyelmeztetés a felhasználó számára

A Download gombra kattintva képes a felhasználó letölteni az analyzer programot, amely bejelentkezést követően automatikusan kinyeri a felhasználó számítógépében található hardvereket és elküldi a szervernek, ahol az feldolgozásra kerül.



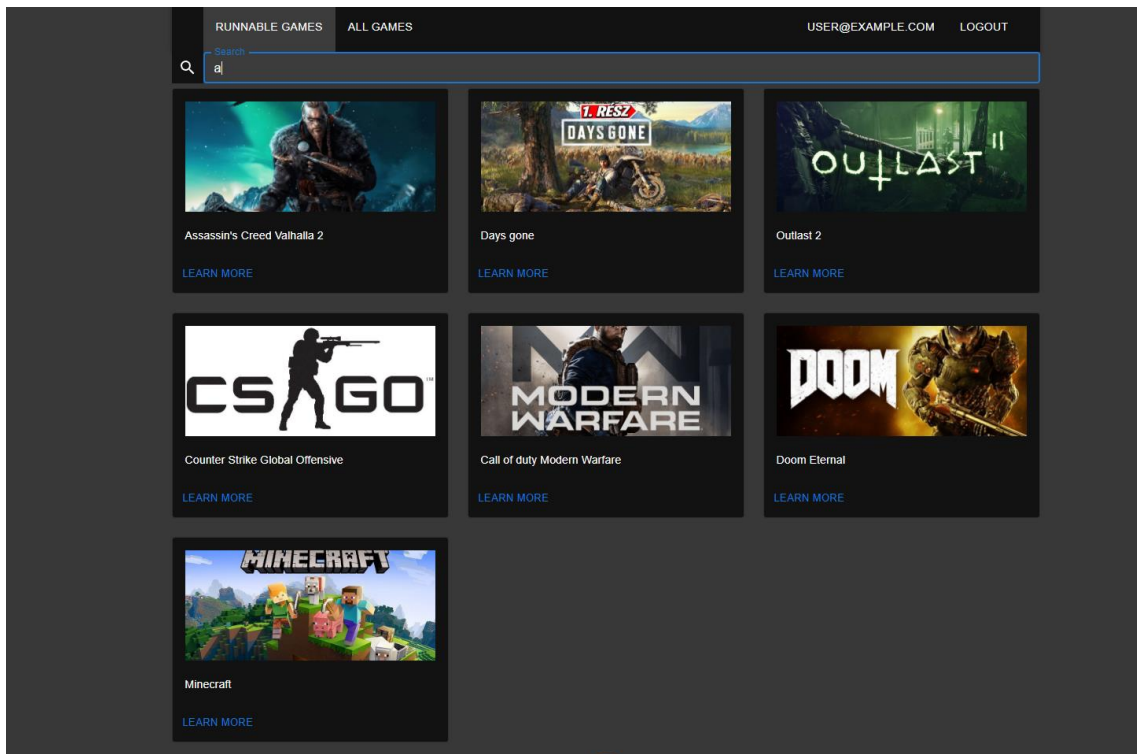
22. ábra Az analyzer alkalmazás sikeres adatküldés esetén

Miután a felhasználó az analyzer program segítségével hozzáadta a számítógépét a rendszerünkhöz, a főoldalon megjelennek azok a játékok, amelyeket képes futtatni a számítógépe.



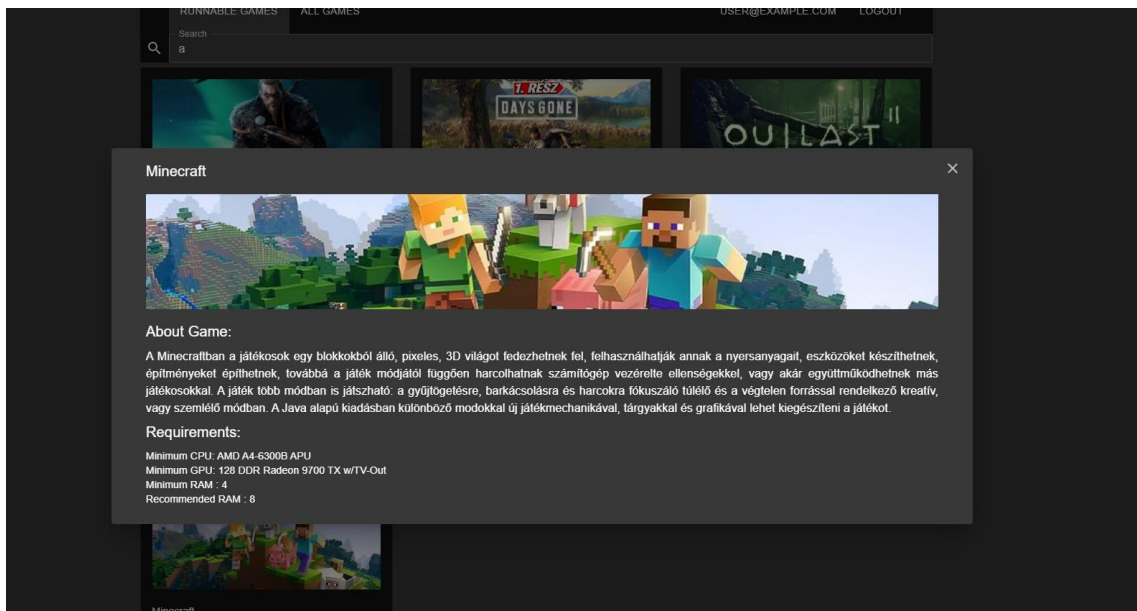
23. ábra Főoldal, ha van a felhasználóhoz számítógép rendelve

A felhasználó képes játékok neve alapján szűrni a navigation bar alatt található searchbar segítségével. Az alábbi ábrán látható a szűrés.



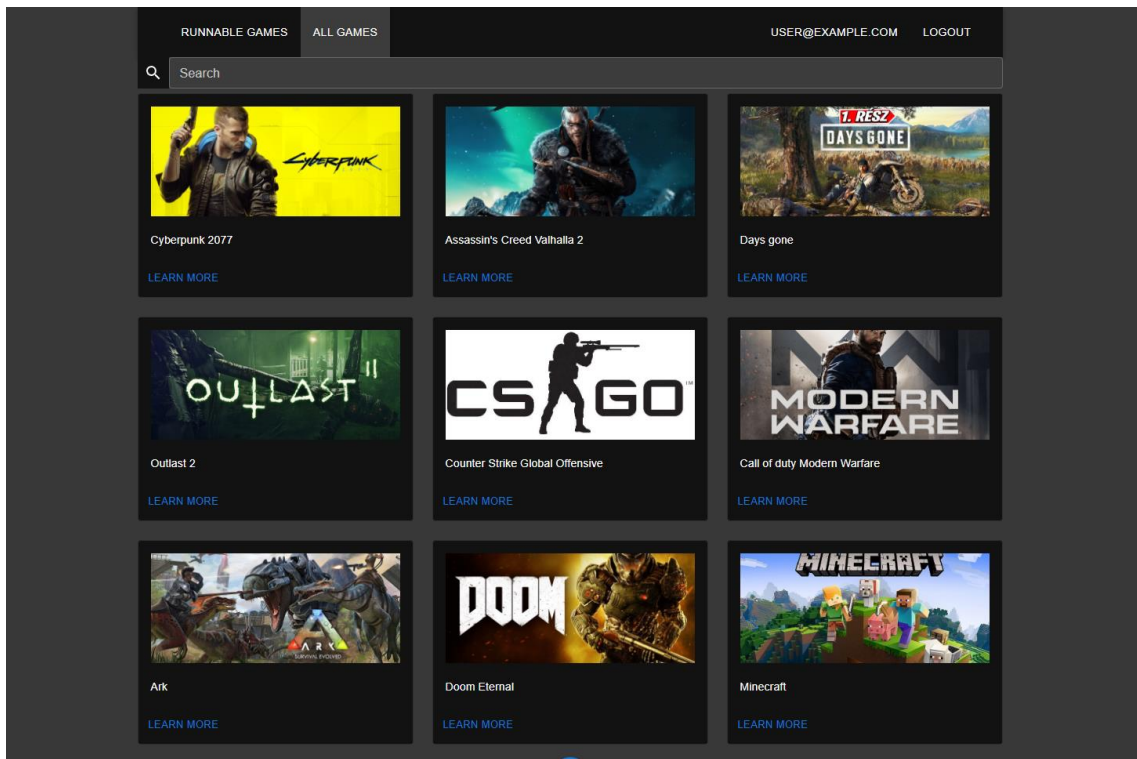
24. ábra Játék nevére való szűrés

Ha több adatot akarunk megtudni a játékról, akkor a learn more gomb megnyomásával tudjuk ezt megtenni. Ekkor feljönnek a játék részletes adatai.



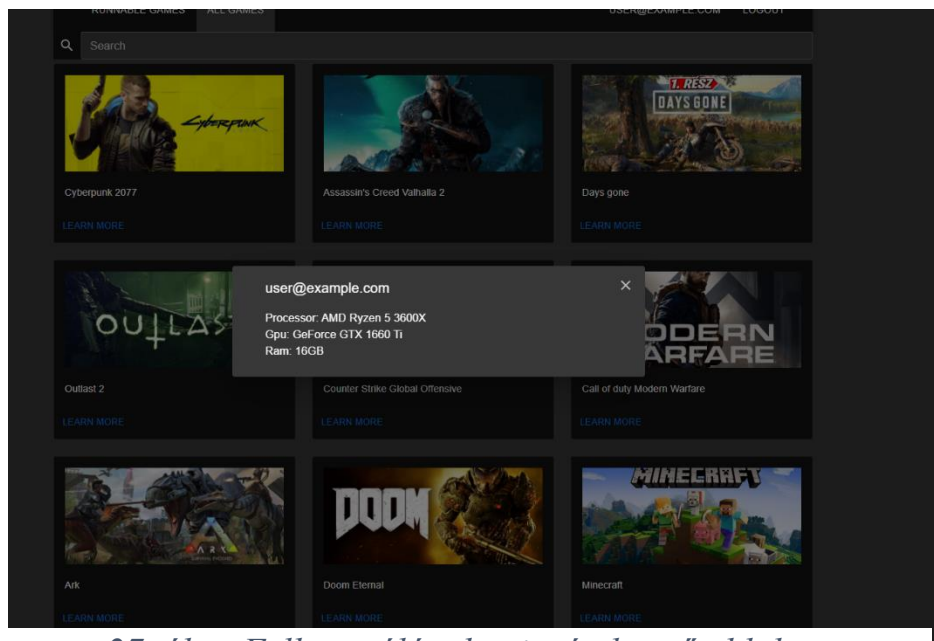
25. ábra Játék részletes adatai

A rendszerben található összes játékot az ALL GAMES menüpont alatt éri el a felhasználó. Ez bármikor elérhető a felhasználó számára, ezt az alábbi kép mutatja.



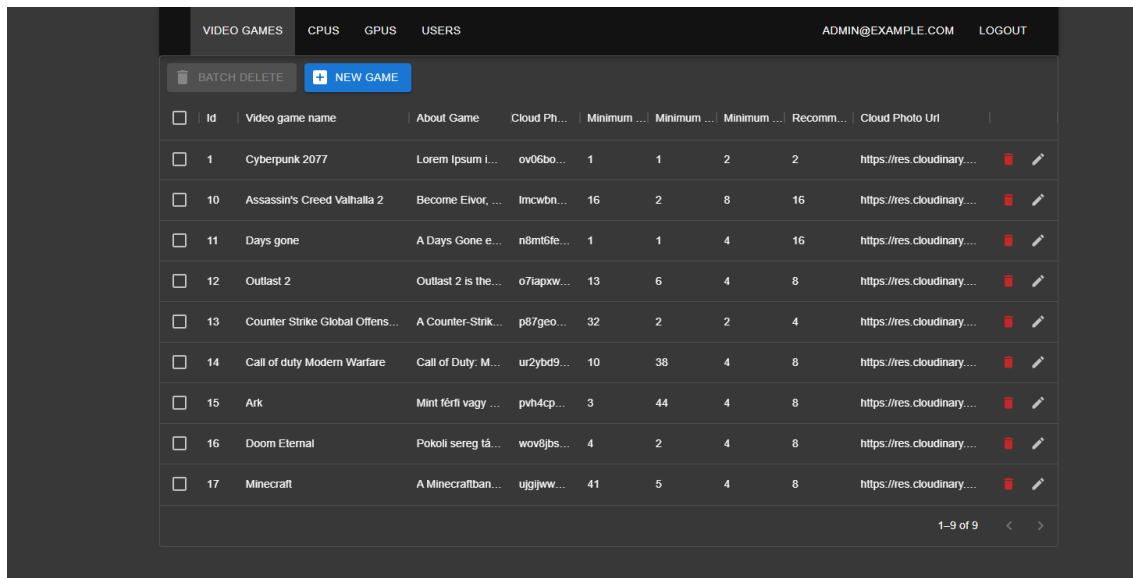
26. ábra Az összes játék megjelenítése

A felhasználó a saját adatait a saját emailcímére kattintva teheti meg. Ezt a menüpontot az alábbi képen látjuk.



27. ábra Felhasználó adatait részletező ablak.

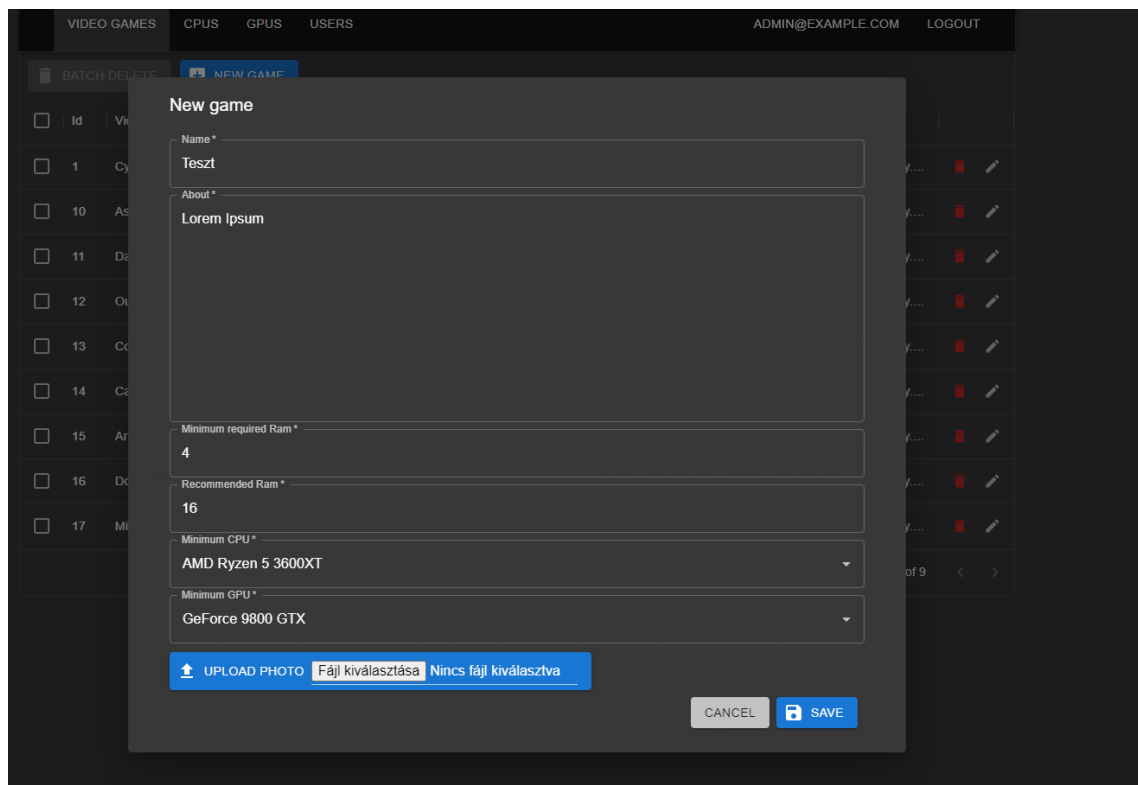
Az alábbi képen az egyik admin funkciót látjuk, ahol kilistázódik az összes játékszoftver.



	Id	Video game name	About Game	Cloud Ph...	Minimum ...	Minimum ...	Minimum ...	Recomm...	Cloud Photo Url	
☐	1	Cyberpunk 2077	Lorem Ipsum i...	ov06bo...	1	1	2	2	https://res.cloudinary...	 
☐	10	Assassin's Creed Valhalla 2	Become Elvor, ...	lmcwbn...	16	2	8	16	https://res.cloudinary...	 
☐	11	Days gone	A Days Gone e...	n8m6le...	1	1	4	16	https://res.cloudinary...	 
☐	12	Outlast 2	Outlast 2 is the...	o7lapxw...	13	6	4	8	https://res.cloudinary...	 
☐	13	Counter Strike Global Offens...	A Counter-Strik...	p87geo...	32	2	2	4	https://res.cloudinary...	 
☐	14	Call of duty Modern Warfare	Call of Duty: M...	ur2ybd9...	10	38	4	8	https://res.cloudinary...	 
☐	15	Ark	Mint férfi vagy ...	pvh4cp...	3	44	4	8	https://res.cloudinary...	 
☐	16	Doom Eternal	Pokoli sereg tá...	wov8jbs...	4	2	4	8	https://res.cloudinary...	 
☐	17	Minecraft	A Minecraftban...	ujgijw...	41	5	4	8	https://res.cloudinary...	 

28. ábra A játékokat kilistázó admin felület

Az admin képes új játékot hozzáadni a rendszerhez, módosítani a játék adatait és törölni a játékot. Ezeket az alábbi képeken látjuk.



New game

Name *
Teszt

About *
Lorem Ipsum

Minimum required Ram *
4

Recommended Ram *
16

Minimum CPU *
AMD Ryzen 5 3600XT

Minimum GPU *
GeForce 9800 GTX

 UPLOAD PHOTO | |

29. ábra Új játék hozzáadása a rendszerhez

VIDEO GAMES CPU'S GPU'S USERS ADMIN@EXAMPLE.COM LOGOUT

BATCH DE

Id: 17

Name *

Minecraft

About *

A Minecraftban a játékosok egy blokkokból álló, pixeles, 3D világot fedezhetnek fel, felhasználhatják annak a nyersanyagait, eszközöket készíthetnek, építményeket építhetnek, továbbá a játék módjától függően harcolhatnak számítógép vezérelte ellenségekkel, vagy akár együttműködhetnek más játékosokkal. A játék több módban is játszható: a gyűjtögetésre, barkácsolásra és harcokra fókuszáló túlélő és a végtelen forrással rendelkező kreatív, vagy szemléltető módban. A Java alapú kiadásban különböző modokkal új játékmechanikával, tárgyakkal és grafikával lehet kiegészíteni a játékot.

Minimum required Ram *

4

Recommended Ram *

8

Minimum CPU *

AMD A4-6300B APU

Minimum GPU *

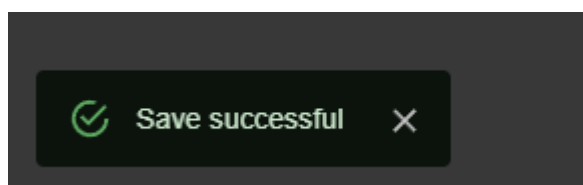
128 DDR Radeon 9700 TX w/TV-Out



CANCEL SAVE

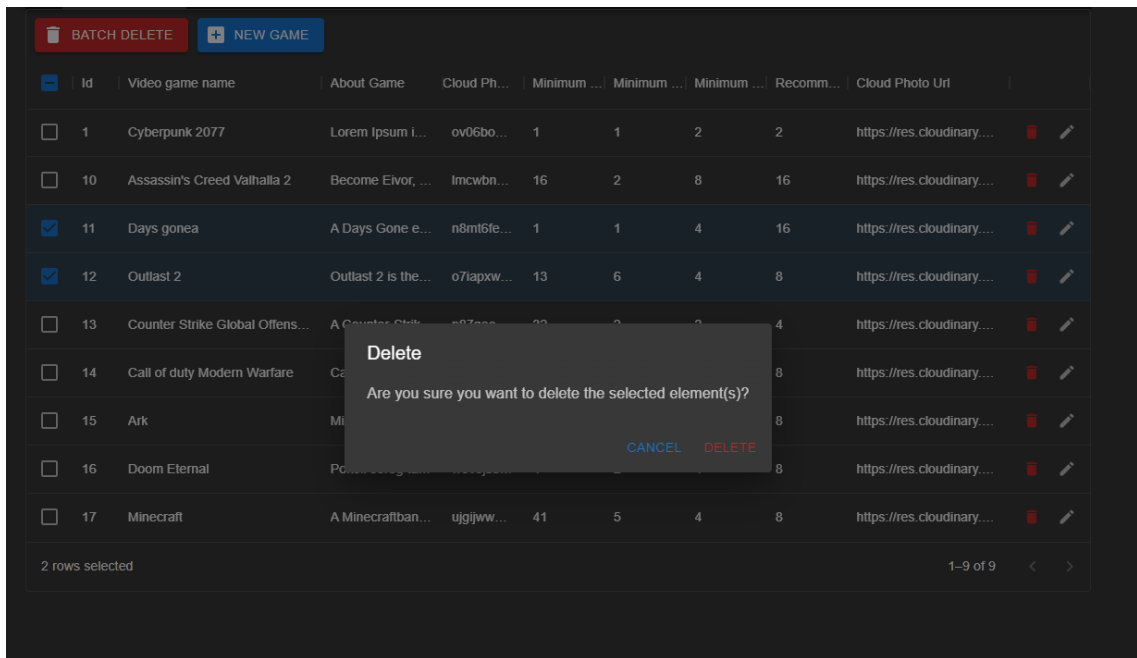
30. ábra Játék adatainak módosítása

Minden sikeres művelet után visszajelzést kap a felhasználó.



31. ábra Műveletek itáni visszajelzés

Az admin képes egyes és többes törlésre is. Mindkettővel kapcsolatban a rendszer egy megerősítést kér.



32. ábra Törlést megerősítő ablak

Processzort és videokártyát kilitázó oldal, ahol képes az admin elindítani az adatbegyűjtést (scrapelést). Ezek az alábbi képeken láthatóak.

VIDEO GAMES

CPU

GPU

USERS

ADMIN@EXAMPLE.COM

LOGOUT

SCRAPE DATAS

Id	Manufacturer	Processor Family	Processor Name	Processor score
1	06/8e		06/8e	5040
2	06/9c		06/9c	3848
3	15/70		15/70	3191
4	-1-Unisoc		-1-Unisoc	1085
5	AArch64		AArch64 rev 0 (aarch64)	2596
6	AArch64		AArch64 rev 1 (aarch64)	2656
7	AArch64		AArch64 rev 2 (aarch64)	2385
8	AArch64		AArch64 rev 4 (aarch64)	2271
9	AC825V/WAB		AC825V/WAB	1729
10	AMD	3015Ce	AMD 3015Ce	735
11	AMD	3015e	AMD 3015e	2105
12	AMD	3020e	AMD 3020e	2720
13	AMD	4700S	AMD 4700S	2495
14	AMD	A4	AMD A4 Micro-6400T APU	18212
15	AMD	A4	AMD A4 PRO-3340B	1004

33. ábra Processzorokat kilitázó oldal

VIDEO GAMES

CPUS

GPUS

USERS

ADMIN@EXAMPLE.COM

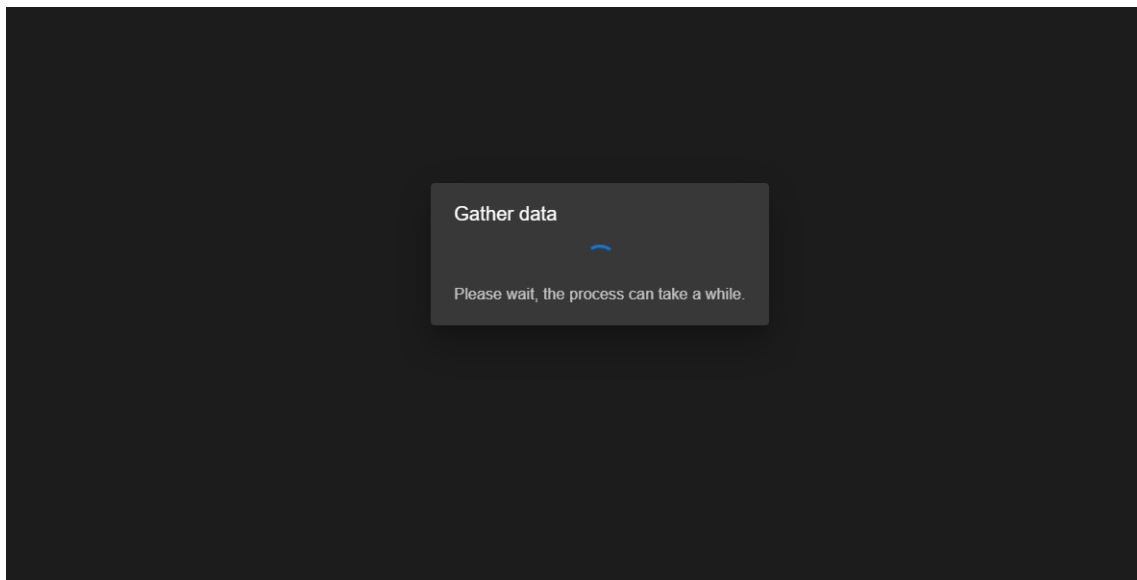
LOGOUT

SCRAPE DATAS

Id	Gpu Name	GPU Model	Gpu Score
1	9xx Soldiers sans frontiers Sigma 2	9xx	826
2	15FF	15FF	21
3	64MB DDR GeForce3 TI 200	64MB	8229
4	64MB GeForce2 MX with TV Out	64MB	5
5	128 DDR Radeon 9700 TX w/TV-Out	128	2
6	128 DDR Radeon 9800 Pro	128	44
7	128MB DDR Radeon 9800 Pro	128MB	62
8	128MB RADEON X600 SE	128MB	66
9	256MB DDR Radeon 9800 XT	256MB	49
10	256MB RADEON X600	256MB	37
11	7900 MOD - Radeon HD 6520G	7900	67
12	7900 MOD - Radeon HD 6550D	7900	610
13	A6 Micro-6500T Quad-Core APU with RadeonR4	A6	892
14	A10-8700P	A10-8700P	220
15	A16	A16	513

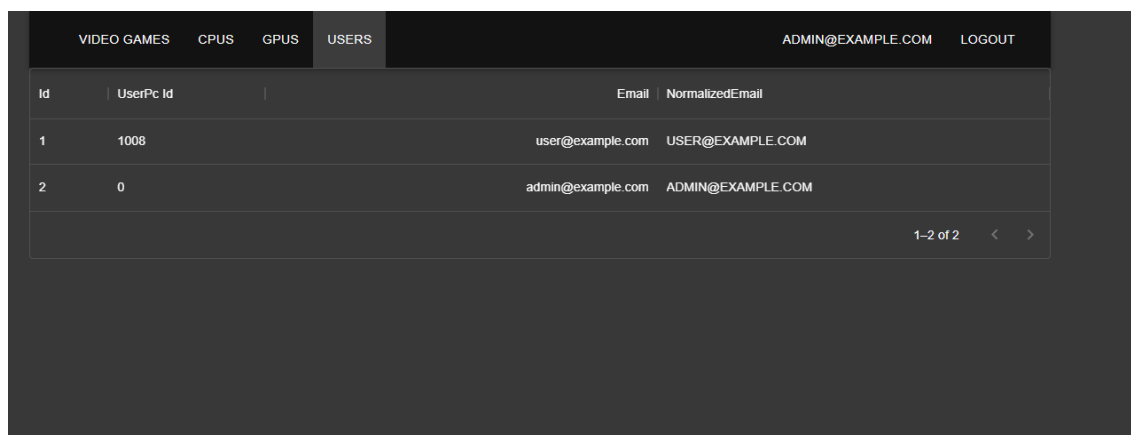
34. ábra Videokártyákat kilistázó felület

Az alábbi ábrán látható, ha az admin elindítja az adatbegyűjtést, akkor minden műveletet letilt a rendszer, amíg tart az adatok begyűjtése.



35. ábra Adatok begyűjtése

Az alábbi képen látható az összes felhasználó, aki a rendszerünkben található.



VIDEO GAMES CPUS GPUS USERS ADMIN@EXAMPLE.COM LOGOUT			
Id	UserPc Id	Email	NormalizedEmail
1	1008	user@example.com	USER@EXAMPLE.COM
2	0	admin@example.com	ADMIN@EXAMPLE.COM
1-2 of 2 < >			

36. ábra Felhasználókat kilistázó felület.

8. Továbbfejlesztési lehetőségek

Az egyetemen eltöltött éveim alatt megtanultam azt, hogy egy szoftver sosem kész, mindig vannak továbbfejlesztési lehetőségek. Ez az általam elkészített alkalmazásra is igaz, a fejlesztés során számtalan új ötlet támadt, amelyet ebben a fejezetben foglalkozok össze.

Az első továbbfejlesztési ötlet az analyzer program fejlesztése során merült fel. Sokkal kényelmesebb lenne a felhasználók számára, ha az analízis szoftverben már nem kéne bejelentkezniük, hanem teljesen automatikus módon működne a program. A felhasználó webes alkalmazásban történő bejelentkezés után tudja letölteni az analízis alkalmazást, ennek következtében megoldható az, hogy a letöltött szoftverben már ne kelljen bejelentkeznie.

Fejlesztés közben, számtalan új funkció jutott eszembe, amely könnyíti a weboldal használatát. Ilyen funkció lenne az is, hogy a sima felhasználók is tudjanak játékot hozzáadni a rendszerhez (amelyet még elbírálna, megerősítene az admin) ezzel segítve az adminok munkáját.

Hasznos funkció lenne egy adott játék értékelése és komment szekció a játékok alatt. Így a felhasználók tudnának szűrni a játékok értékelése alapján, valamint megtudnák osztani a véleményüket az adott játékkal kapcsolatban.

9. Összegzés

Szakedolgozatom során megismerkedtem számtalan új technológiával, sok esetben dokumentációból kellett dolgoznom. A követelmény specifikációban meghatározott pontokat, funkciókat sikerült lefejlesztenem. Hatalmas tanulság volt az, hogy egy jó tervezés nagyon fontos. Az adatbázis tábláimat így is bővítenem kellett, de a megfelelő kapcsolatok miatt egyszerűbb volt a funkciók megírása.

Köszönetet szeretnék nyilvánítani a konzulenseimnek Szabó-Gali Ákosnak és Sipos Miklósnak akik nagy mértékben támogatták, segítették és megfelelő irányba terelték a munkámat.

Summary

During my thesis I was introduced to many new technologies and in many cases I had to work from documentation. I managed to untangle the points and functions defined in the requirements specification. It was a huge lesson that good planning is very important. I still had to extend the database tables, but the right relationships made it easier to write the functions.

I would like to thank my advisors, Szabó-Gali Ákos and Sipos Miklós, who gave me a lot of support and help and guided my work in the right direction.

Irodalomjegyzék

- [1] „pcguru,” [Online]. Available: <https://pcguru.hu/uploads/editors/videojatek-utmutato.pdf>. [Hozzáférés dátuma: 04 12 2021].
- [2] „gepigeny,” [Online]. Available: <https://gepigeny.hu/rolunk>. [Hozzáférés dátuma: 13 10 2021].
- [3] „blog.hubspot,” 2021. [Online]. Available: <https://blog.hubspot.com/marketing/elements-of-modern-web-design-list>. [Hozzáférés dátuma: 08 11 2021].
- [4] „coderweb,” [Online]. Available: <https://www.coderweb.hu/blog/sotet-mod-elonyei>. [Hozzáférés dátuma: 05 11 2021].
- [5] „gwi,” [Online]. Available: https://www.gwi.com/hubfs/Downloads/Global_Ad-Blocking_Behavior.pdf. [Hozzáférés dátuma: 06 12 2021].
- [6] „uxdesign,” [Online]. Available: <https://uxdesign.cc/online-advertising-the-horror-of-ux-c528700db87c>. [Hozzáférés dátuma: 25 10 2021].
- [7] „redhat,” [Online]. Available: <https://www.redhat.com/en/topics/integration/whats-the-difference-between-soap-rest>. [Hozzáférés dátuma: 28 11 2021].
- [8] „dotnet.microsoft,” [Online]. Available: <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet-core>. [Hozzáférés dátuma: 25 11 2021].
- [9] „azure.microsoft,” [Online]. Available: <https://azure.microsoft.com/hu-hu/products/azure-sql/database/#overview>. [Hozzáférés dátuma: 25 11 2021].
- [10] „jakeydocs,” [Online]. Available: <https://jakeydocs.readthedocs.io/en/latest/security/authentication/identity.html>. [Hozzáférés dátuma: 04 12 2021].
- [11] „onelogin,” [Online]. Available: <https://www.onelogin.com/learn/rbac-vs-abac>. [Hozzáférés dátuma: 26 11 2021].
- [12] „zyte,” [Online]. Available: <https://www.zyte.com/learn/what-is-web-scraping/>. [Hozzáférés dátuma: 28 11 2021].
- [13] „blazemeter,” [Online]. Available: <https://www.blazemeter.com/blog/getting-started-with-swagger-ui>. [Hozzáférés dátuma: 08 01 2022].

- [14] „cloudinary,” [Online]. Available: https://cloudinary.com/documentation/cloudinary_get_started. [Hozzáférés dátuma: 20 04 2022].
- [15] „webshark,” [Online]. Available: <https://webshark.hu/hirek/reszponziv-design/#toc0>. [Hozzáférés dátuma: 28 10 2021].

Ábrajegyzék

1. ábra gepigeny.hu főoldala	7
2. ábra gepigeny.hu főoldala mobilon	9
3. ábra Eszköz hozzáadása	10
4. ábra Eredmény megjelenítés	11
5. ábra pcgamebenchmark.com főoldala	12
6. ábra systemrequirementslab.com főoldala	14
7. ábra Processzorról gyűjtött információk	15
8. ábra A Rendszerterv ábrája	23
9. ábra Az adatbázis kezdetleges terve	24
10. ábra Alapértelmezett identity táblák	25
11. ábra Use case diagram a rendszerben található felhasználókról	27
12. ábra Activity diagram az analízis szoftver folyamatáról	30
13. ábra Vízesés modell	32
14. ábra A fejlesztés menete	32
15. ábra Az adatbázis frissített verziója	36
16. ábra Regisztrációs felület	42
17. ábra Rendszer által nem ismert email címmel való beépés	43
18. ábra Rendszer által ismert email cím, de rossz jelszóval való belépési próbálkozás	43
19. ábra Figyelmeztető üzenet a felhasználó számára	44
20. ábra A főoldal számítógép konfiguráció nélkül	44
21. ábra Figyelmeztetés a felhasználó számára	44
22. ábra Az analyzer alkalmazás sikeres adatküldés esetén	45
23. ábra Főoldal, ha van a felhasználóhoz számítógép rendelve	45
24. ábra Játék nevére való szűrés	46
25. ábra Játék részletes adatai	46
26. ábra Az összes játék megjelenítése	47
27. ábra Felhasználó adatait részletező ablak	47
28. ábra A játékokat kilistázó admin felület	48
29. ábra Új játék hozzáadása a rendszerhez	48
30. ábra Játék adatainak módosítása	49
31. ábra Műveletek itáni visszajelzés	49
32. ábra Törlést megerősítő ablak	50

33. ábra Processzorokat kilistázó oldal.....	50
34. ábra Videokártyákat kilistázó felület	51
35. ábra Adatok begyűjtése.....	51
36. ábra Felhasználókat kilistázó felület.....	52