

Szakképesítés neve: Szoftverfejlesztő és -tesztelő technikus

OKJ száma: 5-0613-12-03

SZAKDOLGOZAT

Mygamelist

Rostás Péter

Gulyás Roland, Vecsernyés Áron, Nagy István Bence

témavezető

2 / 14 E

Tartalomjegyzék

Tartalom

Tartalomjegyzék.....	1
Az oldal bemutatása.....	4
Téma választás	4
A weboldal funkciója.....	4
Az oldal kinézete	5
Az oldal jövője	5
Bejelentkezés, regisztráció.....	6
Regisztráció felhasználói útmutató.....	6
Regisztráció fejlesztői dokumentáció	9
Bejelentkezés felhasználói útmutató	12
Bejelentkezés fejlesztői dokumentáció.....	13
Profil.....	14
Profil felhasználói útmutató.....	14
Profil fejlesztői útmutató	17
Játék oldal felhasználói dokumentáció	20
Játék oldal normál felhasználóknak:	20
Játék értékelése:	21
Kedvenchez adás:.....	22
Részletek:	23
Leírás:	24
Karakterek és szinkronhangjuk:	24
Játék oldal Admin felhasználóknak:	24
Játék oldal fejlesztői dokumentáció.....	25
A weblaphoz tartozó mappaszerkezet és file-ok:	25
Játék adatainak megkapása .htaccess-el:	27
Játék adatainak megkapása Game_page.php-ban:	27
Játék értékelése:	27
Játék kedvencnek állítása:.....	27
Toplista.....	28
Rövid leírás az oldalról:	28
Toplista felhasználói dokumentáció	28
Toplista fejlesztői dokumentáció	29
A weblaphoz tartozó file-ok:.....	29

Toplista adatainak megkapása .htaccess-el:	29
Toplista adatainak megkapása Toplist.php-ban:	29
Genre lista	29
Rövid leírás az oldalról:	29
Genre felhasználói dokumentáció:	29
Genre lista fejlesztői dokumentáció	30
A weblaphoz tartozó file-ok:	30
Genre lista adatainak megkapása .htaccess-el:	30
Genre lista adatainak megkapása Genre.php-ban:	30
Fejlesztő és Kiadó lista	31
Rövid leírás az oldalról:	31
Fejlesztő és Kiadó lista felhasználói dokumentáció:	31
Fejlesztő és Kiadó lista fejlesztői dokumentáció.....	31
A weblaphoz tartozó file-ok:	31
Fejlesztő és Kiadó lista lista adatainak megkapása .htaccess-el:	32
Fejlesztő és Kiadó lista adatainak megkapása DevAndPubList.php-ban:	32
Fejlesztő és Kiadó oldal	32
Rövid leírás az oldalról:	32
Fejlesztő és Kiadó oldal felhasználói dokumentáció:	32
Fejlesztő és Kiadó oldal fejlesztői dokumentáció	33
A weblaphoz tartozó file-ok:	33
Fejlesztő és Kiadó oldal adatainak megkapása .htaccess-el:	33
Fejlesztő és Kiadó oldal adatainak megkapása Developer.php és Publisher.php-ban:	34
Sablon.....	34
Rövid leírás az oldalról:	34
Sablon felhasználói dokumentáció:	34
Sablon főbb funkciói:	35
Jobb oldali menü oszlop:	35
Keresés:	36
Sablon fejlesztői dokumentáció	36
A weblaphoz tartozó file-ok:	36
Keresés funkció:	36
Adatbázis adat feltöltés	37
A következő pár lépésben bemutatom egy játéknak az adat feltöltését:	37
1. Lépés: menüpont kiválasztása	37
2. Lépés: Adatok kitöltése	38

3. Lépés: Küldés gomb.....	39
4. Lépés (További lépésekhez admin jogosultság kell).....	40
5. Lépés: adatok ellenörzése	40
6. Lépés: Elfogadás vagy elutasítás	41
Játék Adat feltöltés	42
MyList	43
Adatküldési és MyList Fejlesztői dokumentáció	44
Weboldalon használt programok.....	44
Tesztelés.....	46

Az oldal bemutatása

Téma választás

Az ötletet az oldal elkészítéséhez a videójáték értékelő, nyomon követő oldalak hiánya ihlette. Léteznek hasonló oldalak, de egyik sem játékokkal foglalkozik, és sok olyan funkció hiányzik belőlük, amit mi szívesen látnánk (Pl.: imdb.com). Szerettünk volna egy olyan oldalt készíteni amilyen még nem létezik videójáték témában, és amit mi is szívesen használnánk minden nap.

A weboldal funkciója

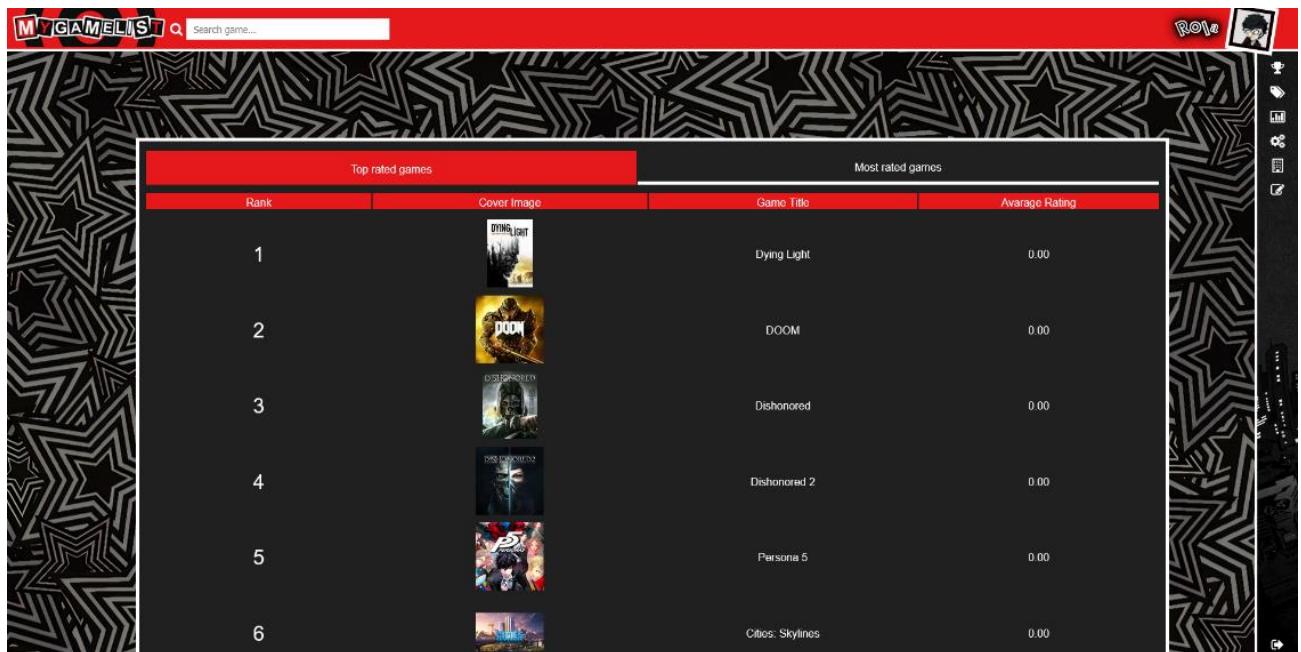
A mygamelist egy online játékkövető adatbázis, amin valós időben nyomon követhetjük saját, illetve ismerőseink játéklistáját. Az oldalon található játékok bármelyikét hozzáadhatjuk a listánhoz hozzáfűzött státusszal (kötelező kitölteni, 5db státusz van: plan to play (tervezel vele játszani), playing (éppen játszol vele), dropped (játszottál vele, de nem fejezted be, és nem is tervezed), on hold (játszottál vele, de nem fejezted be, de tervezed), completed (befejezted a játékot)), értékelléssel (nem kötelező kitölteni, 1-10-es skálán).

Az adatok feltöltése hasonló elven működik, mint a Wikipedia-nál, felhasználók feltöltési kérelmeket tudnak küldeni, amiket az adminok tudnak elfogadni (szerkeszteni is, ha esetleg kell), elutasítani. Jelenleg ez csak nem létező adatoknál működik így szerkeszteni csak adminok tudnak.

A weboldalon minden felhasználónak lehetősége van saját profil elkészítésére. Profil létrehozásakor jön létre a felhasználó listája, ezért az ahhoz tartozó funkciókat csak akkor tudja használni, ha van regisztrált fiókja és be van jelentkezve. Az oldalon található még toplisták, ami a felhasználók értékelése alapján rendezzi a játékokat. A játékokat lehet még műfaj és népszerűség szerint is rendezni, illetve a jövőben platform, kiadási év alapján.

Az oldal kinézete

A mygamerlist kinézetét a Persona 5 című játék ihlette, modern sötét témájú.



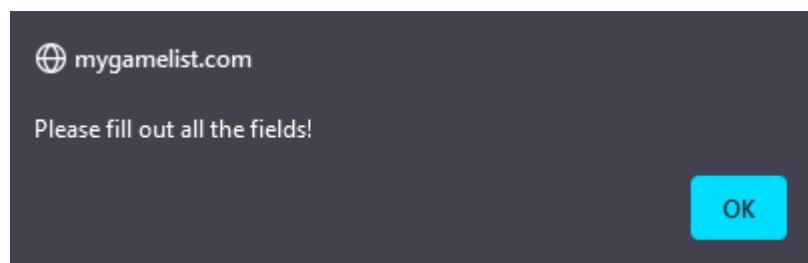
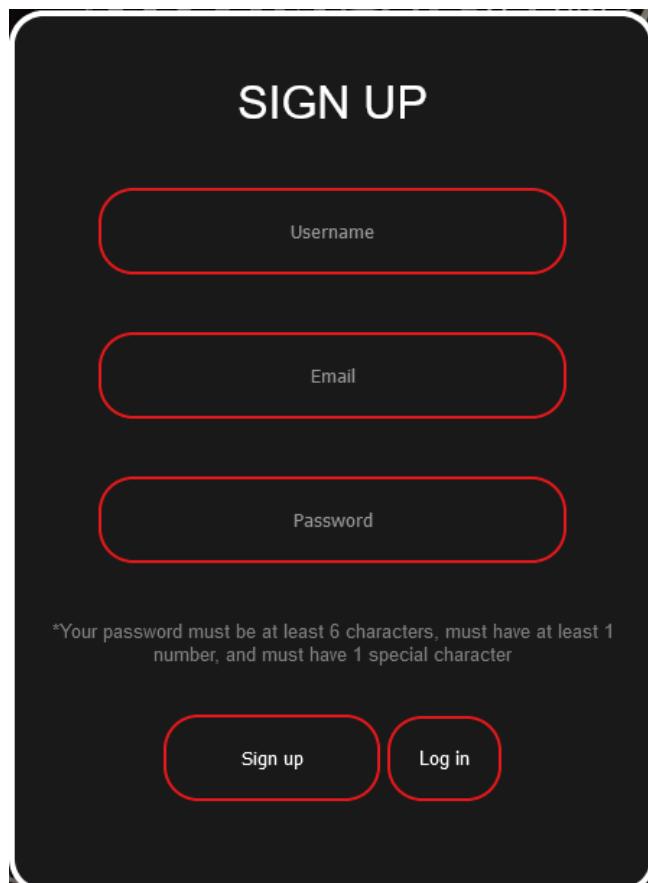
Az oldal jövője

A csoport minden tagja úgy döntött, hogy a jövőben is szeretne foglalkozni az oldallal, ezért nagyon sok tervünk van a jövőre tekintve. Szeretnénk szociális média funkciókat implementálni (barátok. fórum), kiegészíteni a meglévő oldalakat, minden oldalt rest api segítségével elkészíteni. Szeretnénk ha később az adatbázisunk felhasználók közreműködésével gyarapodna, ezért tervezzük, hogy minden olyan adat ami játékkal kapcsolatos szerkeszthető legyen midnenki számára (kérelem formájában). A célunk egy működő oldal fejlesztése, amit mi is szívesen használnánk.

Bejelentkezés, regisztráció

Regisztráció felhasználói útmutató

A regisztrációs oldal a felhasználótól három darab bemenetet kér (email, username, password), ezeknek kitöltése kötelező és szükséges a sikeres regisztrációhoz.



A felhasználónak egy olyan jelszót kell megadnia, ami megfelel az oldalon írottaknak.

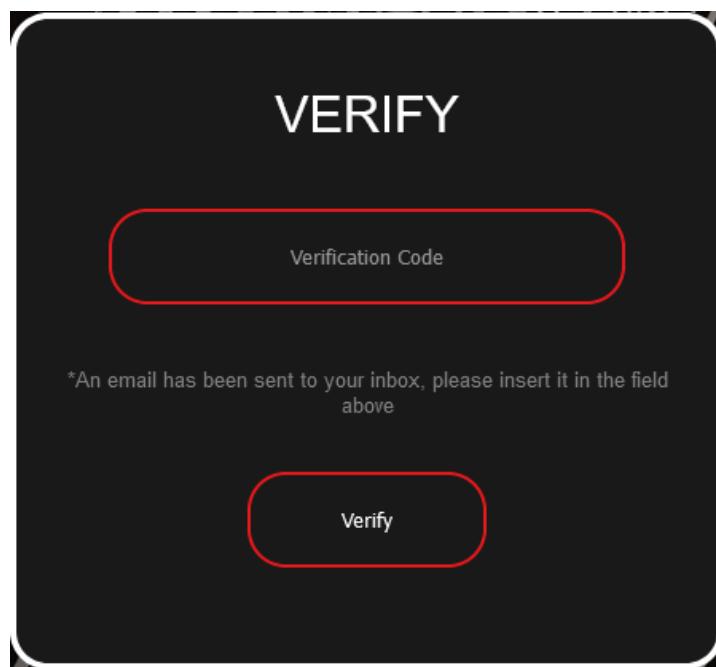
*Your password must be at least 6 characters, must have at least 1 number, and must have 1 special character

A jelszó írása közben az oldal a felhasználónak jelzi a piros, vagy zöld színnel, hogy a jelszó **gyenge** vagy **erős**.



Ha nem megfelelő formátumú jelszóval próbál regisztrálni, akkor az oldal értesíti, hogy a jelszava nem elég erős. Ha már az adatbázisban létező emaillel, vagy felhasználónével próbál regisztrálni akkor az oldal színtén értesíti a felhasználót, hogy valamelyik cím már foglalt.

Miután a felhasználó sikeresen regisztrált az oldal átirányítja a visszaigazoló weblapra.



Ezen az oldalon a felhasználótól egy bemenetet kér az oldal (Verification code), egy random generált 5 számjegyű kódot, amit a felhasználó emailben kap meg egy pár másodpercen belül.

Verifaction Code Beérkező levelek



mygamelist.com <mygamelist.help@gmail.com>

címzett: én ▾

Your Verifaction Code: 87321

 Válasz

 Továbbítás

Az oldal természetesen bemenetként csak az emailben elküldött kódot fogadja el.

Ha a felhasználó kitölti a mezőt a kapott kódjával, akkor a regisztráció sikeres és az oldal automatikusan belépteti a frissen készített profiljába, és átirányítja a kezdőlapra.

Regisztráció fejlesztői dokumentáció

A regisztráció a signUp.php-ról indul, a sablon.php include-olása után html-ben egy formot hoztam létre. A formban található minden megfelelő input mező (classsal, id-vel ellátva), gomb, ami a regisztrációhoz szükséges.

A jelszó erősségét javascript segítségével ellenőrzöm, három darab szempont alapján.

```
var number = (/[\d]/);
var alphabets = (/[\w]/);
var special_characters = (/[^~!@#$%^&*,_=,>,<]/);
```

Legalább hat karakter hosszú legyen, minimum egy szám, és minimum egy speciális karaktert kell tartalmaznia.

Ha a „password” id-jű input mezőbe nem megfelelő formátumú jelszót adunk meg akkor megváltoztatja az input mező betűszínét pirosra, ha a formátumnak megfelel, akkor pedig zöldre változtatja a betűszínt.

Csak akkor tudunk regisztrálni, ha a sikeres erős jelszót megadnunk, és kitöltöttük a többi mezőt is, ekkor javascript/ajax segítségével Postolm az adatokat a post típusával (type = mivel egy php-ban kezelem a verify.php, signup.php postjait valahogyan ellenőriznem kell melyiknek kell kezelni a postot) signUpPost.php-re, ahol az adatokat elmentem változókba (sessionbe), ezután ellenőrzöm hogy az email formai szabályosságát. Csatlakozok az oldal adatbázisához és függvény segítségével ellenőrzöm, hogy foglalt-e az email vagy a felhasználónév

```
if(filter_var($_SESSION['email'], FILTER_VALIDATE_EMAIL)){
    if (!UsernameIsTaken($_SESSION['username']) && !EmailIsTaken($_SESSION['email'])) {

function EmailIsTaken($email){
    $conn = mysqli_connect("localhost", "root", "", 'nagyprojekt');
    $sql = "SELECT email FROM users WHERE email = '$email'";
    return mysqli_num_rows($conn->query($sql));
}
```

Ha nem talál már létező felhasználót a beküldött email, fehasználónévvel, akkor elküldöm emailben a verifikációs kódot a felhasználónak és visszatérési értékként egy státuszkódot küldök vissza.

```
echo json_encode(array("statusCode"=>200));
```

Statuscode azt jelenti:

- 200 - Az adatbázisban nem voltak benne a vizsgált adatok, tovább lehet irányítani a verify.php oldalra.
- 201 - Az adatbázisban benne voltak a vizsgált adatok, másik emailt/felhasználót kell választania a felhasználónak.
- 202 - Az email nem felelt meg a formai követelményeknek, új, helyes email címet kell megadni.

```
if(dataResult.statusCode==200){
    location.href = "verify.php";
}
else if(dataResult.statusCode==201){
    alert("Email or username is already taken!");
}
else if(dataResult.statusCode==202){
    alert("Your email address is not valid!");
}
```

A verify.php oldal ugyanazon az alapelven működik, mint a signup.php, annyi különbséggel, hogy csak egy input mezővel rendelkezik.

A verification code-ot PHPmailer segítségével küldöm el a felhasználónak (<https://github.com/PHPMailer/PHPMailer>).

A verification code egy öt számjegyből álló random generált szám, amit beírva szintén javascript-el/ajax postolom a signUpPost.php-re.

Ellenőröm, hogy a kód megegyezik-e az emaliben elküldöttel, ha igen akkor csatlakozok az oldal adatbázisához és feltöltöm a user táblába a session-be elmentett adatokat (email, Felhasználónév, jelszó) ezek mellett még feltöltöm a mai dátumot (csatlakozás dátuma), az oldal alapértelmezett profilképét, és az alapértelmezett jogosultságát a felhasználónak (Basic/user).

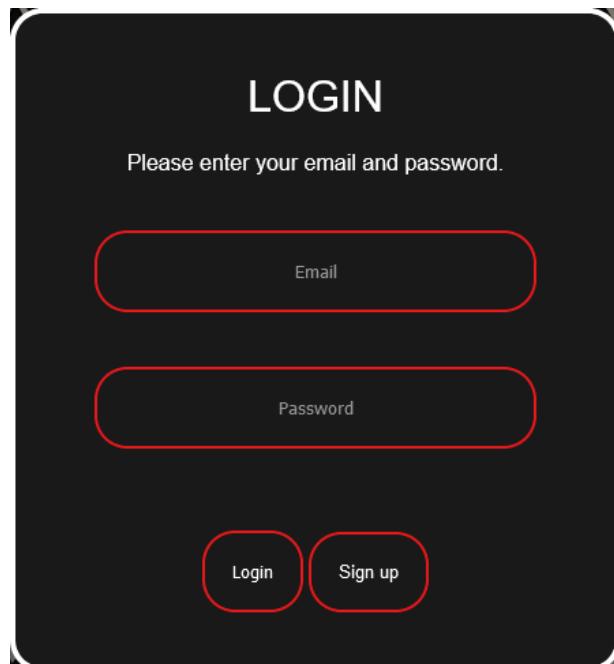
Attól függően, hogy a verification code helyes volt-e visszatérési értékként küldök státuszkódot.

Statuscode azt jelenti:

- 200 – A beküldött verification code egyezett az emailben kapott kóddal, a regisztráció sikeres volt, beléptem, értesítem, tovább irányítom a főoldalra.
- 201 - A beküldött verification code nem egyezett az emailben kapott kóddal, a regisztráció sikertelen volt, ellenőrizni kell a kódott, és újra el kell köldeni.

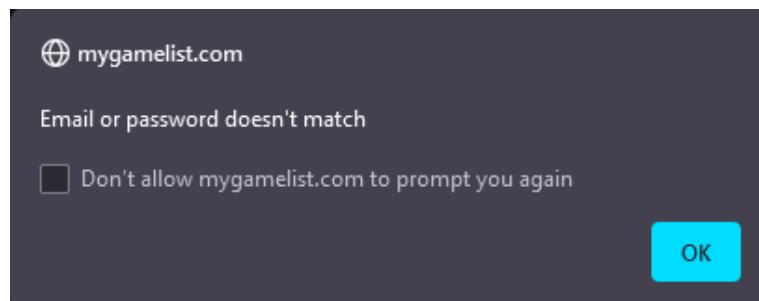
Bejelentkezés felhasználói útmutató

A bejelenkező oldal a felhasználótól kettő darab bemenetet kér (email, password), ezeknek kitöltése kötelező és szükséges a sikeres bejelentkezéshez



Ha a felhasználó már létező email és jelszó kombinációt ad meg akkor a bejelentkezés sikeres volt, és a tovább lesz irányítva a főoldalra.

Ha a felhasználó nem létező kombinációt ad meg, akkor az oldal értesíti erről.



Bejelentkezés fejlesztői dokumentáció

A login.php ugyanazon az alapelven működik mint a signup.php vagy a verify.php.

Ezen az oldalon is javascript/ajax segítségével továbbítjuk az adatokat post-al annyi különbséggel, hogy nem a signUpPost.php-ra, hanem a loginPost.php-ra küldjük az adatokat.

A loginPost.php-ban ellenőrizzük függvény segítségével a bejelentkezési adatokat.

```
if (CheckLogInInfo($_POST['email'], $_POST['password'])) {  
  
function CheckLogInInfo($email, $password){  
    $conn = mysqli_connect("localhost", "root", "", 'nagyprojekt');  
    $password = md5($password);  
    $sql = "SELECT email, passwords FROM users WHERE email = '$email' AND passwords = '$password'";  
    return mysqli_num_rows($conn->query($sql));  
}  
}
```

Ha az adatellenőrzés megtörtént, akkor eredményétől függően küldök státuszkód visszatérési értéket.

Statuscode azt jelenti:

- 200 - Az adatbázisban benne voltak a vizsgált adatok, beléptetem tovább irányítom a főoldalra.
- 201 - Az adatbázisban nem voltak benne a vizsgált adatok, nem létezik a beküldött felhasználónév és jelszó kombináció, értesítem erről.

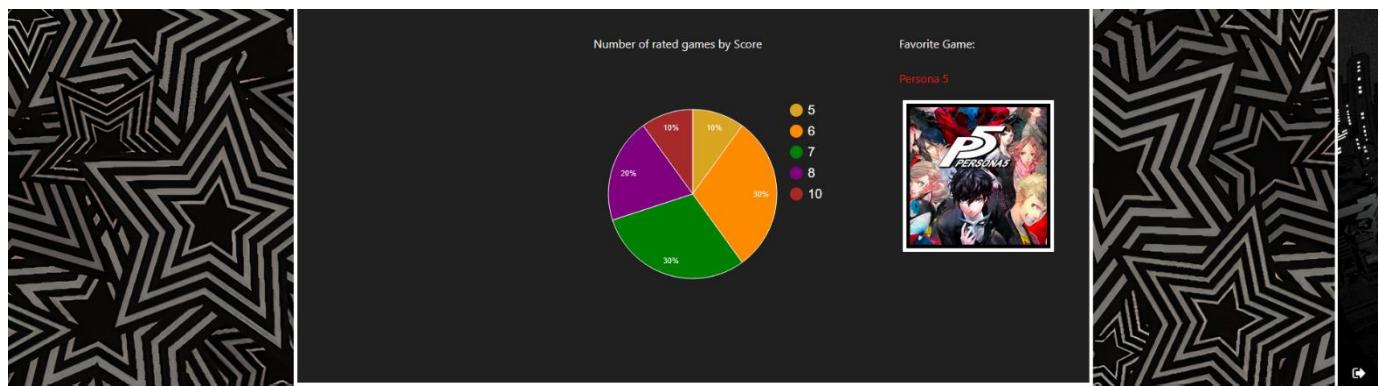
```
if (CheckLogInInfo($_POST['email'], $_POST['password'])) {  
    $_SESSION["signedUp"] = true;  
    $_SESSION["email"] = $_POST['email'];  
    echo json_encode(array("statusCode"=>200));  
}  
else{  
    echo json_encode(array("statusCode"=>201));  
}
```

```
success: function(dataResult){  
    var dataResult = JSON.parse(dataResult);  
    if(dataResult.statusCode==200){  
        location.href = "/ MainPage.php";  
    }  
    else if(dataResult.statusCode==201){  
        alert("Email or password doesn't match");  
    }  
}
```

Profil

Profil felhasználói útmutató

A profil oldalon látjuk az adott felhasználó adatait, statisztikáit, innen tudunk eljutni a listájához.

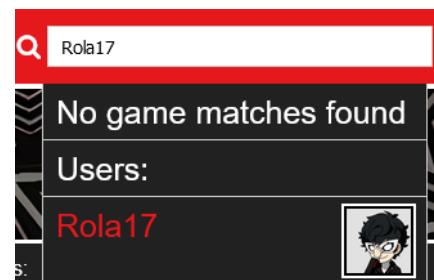


A profil oldal bármelyik jelenleg regisztrált felhasználóval működi, megjeleníti adatait. Két féle képpen tudunk eljutni egy másik felhasználó oldalára:

1. Az url átírásával minta szerint:

mygamelist.com/profile/Rola17

2. A keresés mező használatával a minta szerint:



Az oldal bal oldalán a felhasználó profilképéét, adatait láthatjuk.

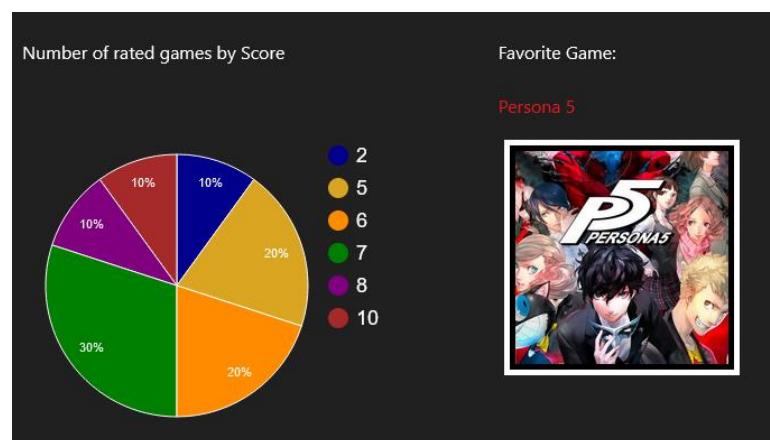
Az adatok alatt található gombok közül a weboldal jelenlegi verziójában még csak egy érhető el a „Game List”, ez a gomb a jelenlegi profil oldalon lévő felhasználó listájára irányít.

Az About Me része az oldalnak a felhasználó által megadott leírást tartalmazza, általában rövid bemutatkozás...

A játék Game Stats részének első fele a felhasználó listájához hozzáadott játékok statisztikáit mutatja.



A Mean score a felhasználó átlagértékelést jelzi, a Total games pedig az összes játék darabszáma amit a listájához hozzáadott. Öt darab státusz van felsorolva színkódolva, itt találhatóak a hozzáadott játékok státuszainak darabszámai, ezek alatt pedig egy diagram, ami a státusz darabszámok arányát jeleníti meg színkódolva. Jobbra pedig a listához legutoljára hozzáadott játkot találjuk.



A Game Stats második felében találunk szintén egy diagramot, ez listához hozzáadott játékok értékelésének darabszámának arányát jeleníti meg. Jobb oldalon pedig pedig a felhasználó kedvenc játéka található.

Profil fejlesztői útmutató

Az oldal adatait rest-api segítségével kérem le.

```
$ch = curl_init();
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_URL,"http://rest-api.com/users/".$_GET["name"]);
$output = curl_exec($ch);
$userData = json_decode($output, true);
curl_close ($ch);
```

A „Userdata” tömbben tárolom el a profil odlalon megjelenítendő felhasználó adatait.

Az oldal csak akkor tekinthető meg, ha a keresett felhasználónév benne van az adatbázisban, ha ez nem így van akkor a 404.php oldalra irányítom.

```
if(empty($userData))
{
    header("Location: 404.php");
}
```

Az oldalon található diagrammokat a google charts segítségével készítettem el.

```
google.charts.load('current', {'packages':['corechart']});
google.charts.setOnLoadCallback(drawChart);

function drawChart() {

var data = google.visualization.arrayToDataTable([
['Rating', 'Number of ratings'],
<?php
for ($i=1; $i <= 10; $i++) {
| echo("[ ".$i." ,".NumOfGivenScore($allGames, $i)."] ,");
}
?>
]);
}

var options = {
  backgroundColor: { fill:'transparent' },
  titleTextStyle: { color: "white" },
  chartArea:{left:20,bottom:100,width: '100%',height: '100%' },
  legend:{position: 'right', textStyle: {color: 'white', fontSize: 18}},
  colors:['darkred','darkblue', 'darkmagenta', 'darkgreen', 'goldenrod', 'darkorange', 'green', 'purple', 'blue', 'brown']
};

var chart = new google.visualization.PieChart(document.getElementById('piechart'));

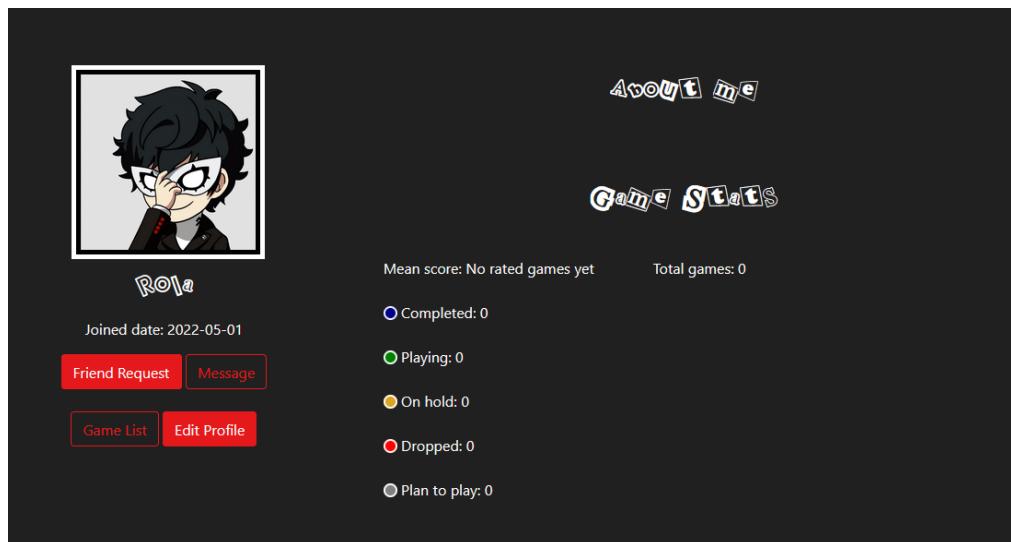
chart.draw(data, options);
}
```

A profil játék statisztikáknál egyszerű számításokat végzik el pl:

```
function MeanScore($allGames)
{
    if (!empty($allGames)) {
        $numScore = 0;
        $sumScore = 0;

        for ($i=0; $i < count($allGames); $i++) {
            if ($allGames[$i]["rating"] != 0)
            {
                $sumScore += $allGames[$i]["rating"];
                $numScore++;
            }
        }
    }
}
```

Az oldalon egyes statisztikák csak akkor jelennek meg ha a felhasználónak vannak játékok hozzáadva a listájához.



Game page vagyis a játék oldala

Rövid leírás az oldal lényegéről:

A weboldalamon a felhasználó egy jól átlátható és könnyen kezelhető felületen tekintheti meg az általa kiválasztott játék összes adatát, értékelhetjük és a kedvencek közé is rakhatjuk az adott játékot. Ezekhez, ha akarja, hozzáadhat vagy változtathat rajtuk és ezt az adminok felé kérésként elküldheti.

Játék főbb megjelenített adatai:

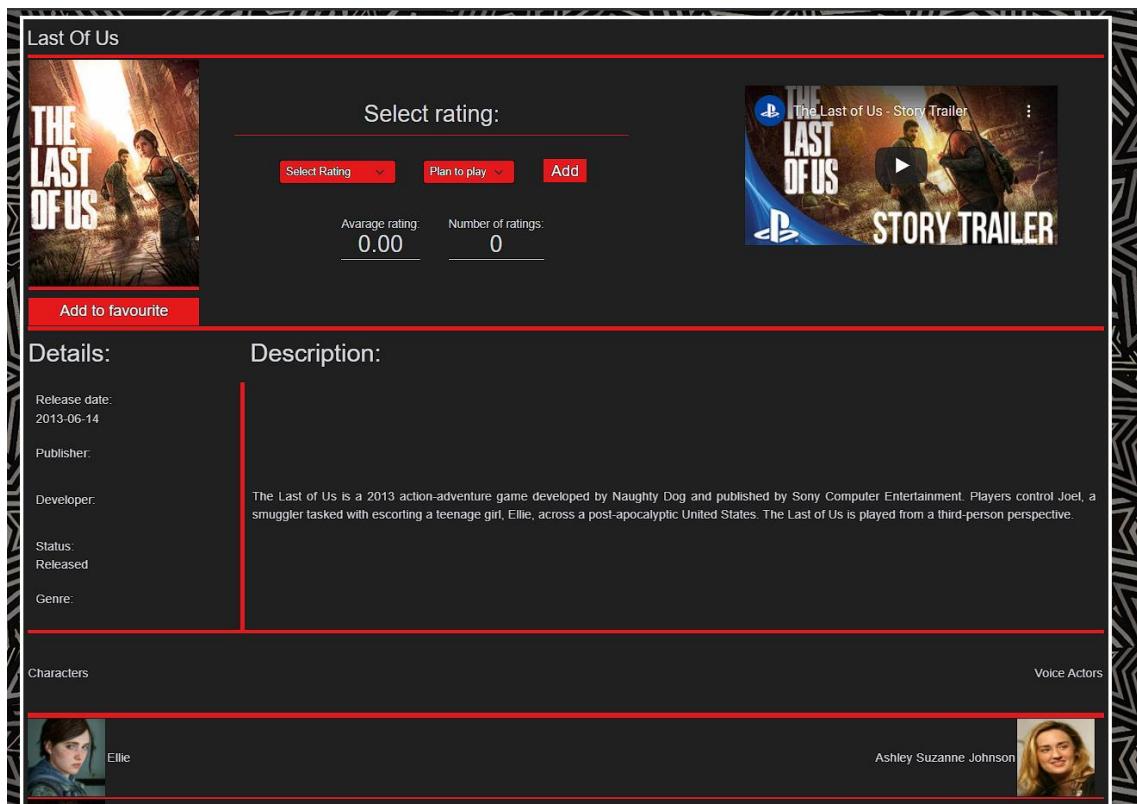
- Játék címe
- Játék borítóképe
- Játék előzetese (trailer-e)
- Játék átlag értékelése
- Játék értékeléseinek száma
- Játék műfaja
- Játék fejlesztői
- Játék kiadói
- Játékban szereplő karakterek
- Játék karaktereinek szinkronhangjai

Játék főbb funkciói:

- Játék értékelése, saját listához adása
- Játék kedvencekhez adása
- Műfaj hozzáadás
- Kiadó hozzáadása
- Fejlesztő hozzáadása
- Játék adatainak váloztatása
- Karakter hozzáadása
- Szinkronhang hozzáadása
- Játék törlése a saját listából

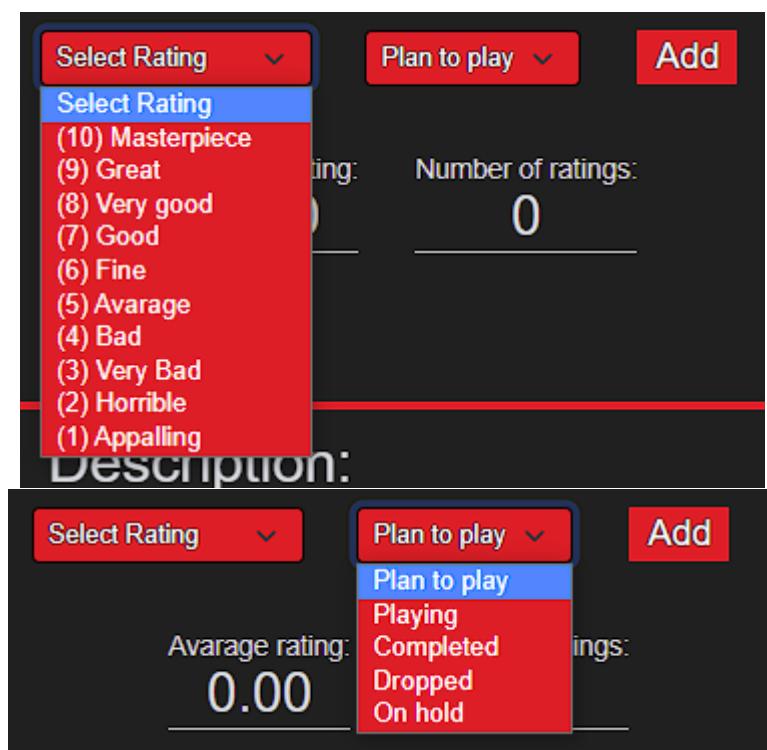
Játék oldal felhasználói dokumentáció

Játék oldal normál felhasználóknak:

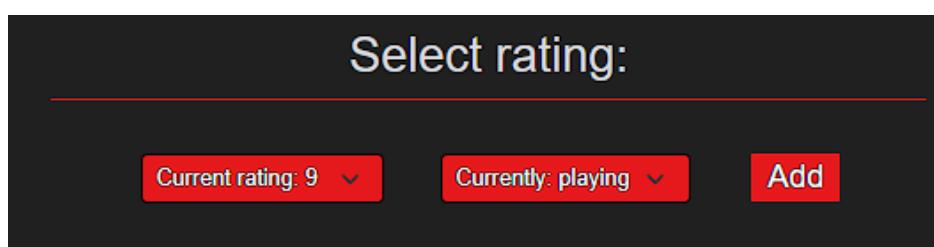


Ahogy a képen látható, az oldal megjeleníti a megadott játék adatait: cím, átlag értékelés, értékelések száma, leírás, borítókép, előzetes videó, karakterek és a hozzá tartozó szinkronszínészeket.

Játék értékelése:

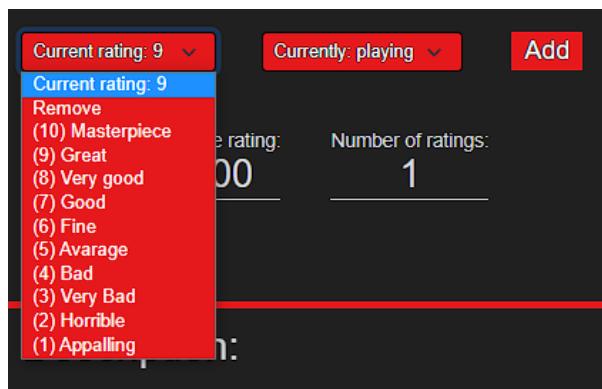


Egy játék értékeléséhez vagy listához adásához kell választanunk egy értékelést (vagy a Select Rating-et ha még nem akarjuk értékelni), az első lenyíló fülből majd a másodikból a végig játszás helyzetét kell megadni, miszerint: (Plan to play) tervezük, hogy játszunk a játékkal, (Playing) éppen játszunk vele, (Completed) végig játszottunk a játékot, (Dropped) nem játszunk vele és ezért nem tervezünk végig játszani vagy (On hold) éppen nem játszunk a játékkal, de tervezük folytatni. Ezek után az Add gombbal hozzá lehet adni a listához. Ha egy játék szerepel a listánkba akkor az értékelésnél kijelzi a mi általunk adott értékelést és helyzetet. (Bekezdés alatti kép szemléltetésképpen)



Ha esetleg a felhasználónak megváltozott a véleménye, törölni akarja a listából, más értékelt szeretne adni vagy esetleg eddig csak játszott a játékkal viszont befejezte és ezért át akarja állítani a játék értékelését ezt megteheti, ha újra rányom a lenyíló fülekre és kiválasztja a szerinte jónak talált opciókat. Értékelés után az értékelés fülben

megtalálható egy “Remove” opció, ami a nevéhez hűen a játék törlését eredményezi a saját listából. (Bekezdés alatti kép szemléltetésképpen)



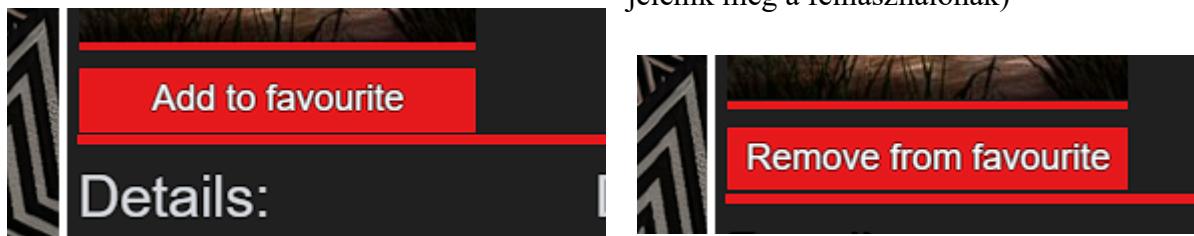
Ha a játék értékelésre került annak átlag értékelése és értékelésének száma is változik az összes felhasználó értékelése és az értékelések száma alapján. (Bekezdés alatti kép szemléltetésképpen)



Kedvenchez adás:

A felhasználónak lehetősége van a játék kedvencek-nek való tételehez. Ehez a játék borítóképe alatt megjelenő “Add to favourite” gombbal teheti meg. Ha a játék már a felhasználó kedvence akkor az “Add to favourite” helyett “Remove from favourite” gomb jelenik meg. Egy felhasználónak egyszerre egy kedvenc játéka lehet ezért, ha másik játékot állít be kedvencnek akkor az előző kedvenc törlődik a kedvencek közül.

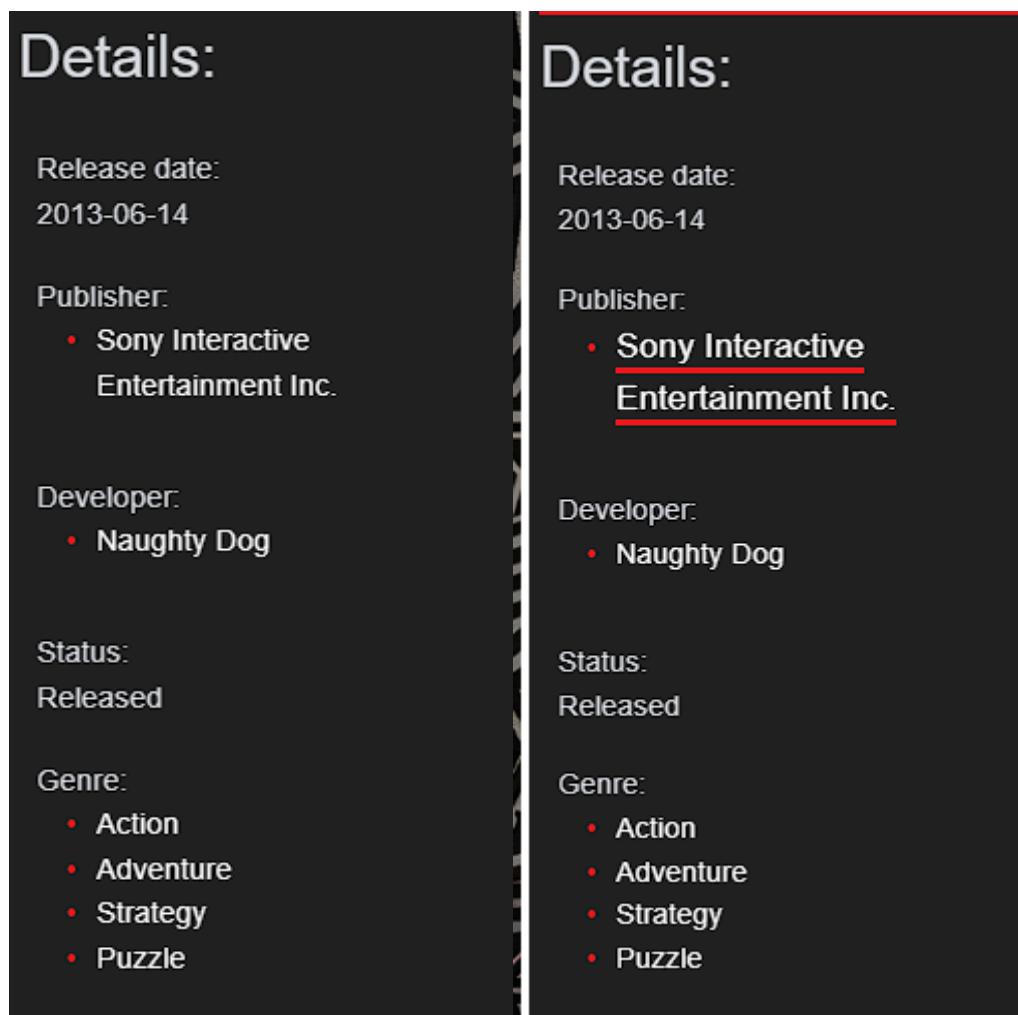
(A bekezdés alatti bal oldali kép a kedvenchez adás gombját mutatja, ha nincs kedvencnek beállítva a játék, ha kedvenc a játék akkor a jobb oldali képen látható szöveg jelenik meg a felhasználónak)



Részletek:

A részletek (Details) alatt jelenítjük meg a játékhoz kapcsolódó adatokat, mint például: megjelenési idő, kiadók, status (megjelent, nem jelent még meg, korai hozzáférés...) és a műfajokat. A kiadó, fejlesztő és műfajnál megjelenő adatok linkek, amikre rákattintva el lehet jutni a saját oldalukra, ha föléjük visszük az kurzor méretük megnő és megjelenik egy aláhúzás láthatóság miatt.

(Bal oldali kép a játék oldalon belüli részletek szemléltetésképpen, jobb oldali képen látható a változás mikor a kiadók felé vittem az kurzort)



Rájuk kattintva, kiadóknál és fejlesztőnél saját oldaluk, műfajnál az adott műfaj toplistája oldalára irányít át.

Leírás:

A leírás résznél a játék leírását láthatja a felhasználó.

Description:

The Last of Us is a 2013 action-adventure game developed by Naughty Dog and published by Sony Computer Entertainment. Players control Joel, a smuggler tasked with escorting a teenage girl, Ellie, across a post-apocalyptic United States. The Last of Us is played from a third-person perspective.

Karakterek és szinkronhangjuk:

A karakter és szinkronhang tábla bal oldalán láthatja a felhasználó a játékban szereplő karakterek képét és nevét, a tábla jobb oldalán pedig a hozzá tartozó szinkron színész nevét és képét jeleníti meg. Ha nincs a karakterhez szinkronszínész akkor nem jelenít meg semmit jobb oldalt. (Szemléltetésképp a bekezdés alatti kép)

Characters	Voice Actors
 Ellie	 Ashley Suzanne Johnson
 Joel Miller	

Játék oldal Admin felhasználóknak:

A játék oldal kibővül funkciókkal, ha a bejelentkezett fiók rendelkezik "Admin" joggal. Mint például fejlesztő, kiadó, műfaj, karakter, szinkronhang hozzá adás. A karakter és szinkronhang hozzá adás gombot kék színnel jelenítjük meg a jobb láthatóság miatt. Ezek lenyomása után az oldal átirányít egy adatfeltöltő lapra attól függően, hogy mit akartunk hozzáadni a játékhoz. (A bekezdés alatti képek szemléltetésként)

Details:

Release date:
2013-06-14

Publisher:

- Sony Interactive Entertainment Inc.
- + Add publisher

Developer:

- Naughty Dog
- + Add developer

Status:
Released

Genre:

- Action
- Adventure
- Strategy
- Puzzle
- + Add genre

Characters
Voice Actors

Character	Voice Actor
 Ellie	Ashley Suzanne Johnson 
 Joel Miller	Add voice actor

Add character

Játék oldal fejlesztői dokumentáció

A weblaphoz tartozó mappaszerkezet és file-ok:

A játék olda a Game_page.php-t jeleníti meg a .htaccess file segítségével. De közben használja a css/Gamepage.css-t és a reszt-api mappa .htaccess, Game.php, GameRestHandler.php és RestContorrel.php részét is.

(A bekezdés alatti képek szemléltetéshez, bal oldali képen a rest-api mappa látható, jobb oldali képen pedig a projekt fő mappája látható)

 .htaccess	2022. 04. 30. 8:15	HTACCESS fájl	3 KB
 Favorite.php	2022. 04. 30. 8:15	PHP fájl	1 KB
 FavoriteRestHandler.php	2022. 04. 30. 8:15	PHP fájl	1 KB
 Game.php	2022. 04. 30. 8:02	PHP fájl	3 KB
 GameRestHandler.php	2022. 04. 30. 7:45	PHP fájl	2 KB
 GetDataById.php	2022. 04. 30. 8:15	PHP fájl	3 KB
 GetDataByIdRestHandler.php	2022. 04. 30. 8:15	PHP fájl	4 KB
 Lists.php	2022. 04. 30. 10:35	PHP fájl	3 KB
 ListsRestHandler.php	2022. 04. 30. 8:15	PHP fájl	3 KB
 Mylist.php	2022. 04. 30. 8:15	PHP fájl	2 KB
 MyListRestHandler.php	2022. 04. 30. 8:15	PHP fájl	1 KB
 RelatedGames.php	2022. 04. 30. 8:15	PHP fájl	1 KB
 RelatedGamesRestHandler.php	2022. 04. 30. 8:15	PHP fájl	1 KB
 RestController.php	2022. 04. 30. 8:16	PHP fájl	8 KB
 TableRquest.php	2022. 04. 30. 8:15	PHP fájl	1 KB
 TableRquestRestHandler.php	2022. 04. 30. 8:15	PHP fájl	1 KB
 UserRestHandler.php	2022. 04. 30. 8:15	PHP fájl	1 KB
 Users.php	2022. 04. 30. 8:15	PHP fájl	2 KB
 css	2022. 05. 01. 20:10	Fájlmappa	
 html	2022. 05. 01. 18:27	Fájlmappa	
 images	2022. 05. 01. 18:27	Fájlmappa	
 javascript	2022. 05. 01. 18:27	Fájlmappa	
 PHPMailer	2022. 05. 01. 18:27	Fájlmappa	
 profil	2022. 05. 01. 18:27	Fájlmappa	
 sablon	2022. 05. 01. 18:27	Fájlmappa	
 sql	2022. 05. 01. 20:10	Fájlmappa	
 Test	2022. 05. 01. 18:27	Fájlmappa	
 .htaccess	2022. 05. 01. 18:27	HTACCESS fájl	3 KB
 404.php	2022. 05. 01. 18:27	PHP fájl	1 KB
 AdatokFeltöltése.txt	2022. 05. 01. 18:27	Szöveges dokume...	2 KB
 applications.html	2022. 05. 01. 18:27	Firefox HTML Doc...	4 KB
 asd.php	2022. 05. 01. 18:27	PHP fájl	3 KB
 bitnami.css	2022. 05. 01. 18:27	CSS fájl	1 KB
 DataSend.php	2022. 05. 01. 18:38	PHP fájl	24 KB
 DevAndPubList.php	2022. 05. 01. 18:27	PHP fájl	7 KB
 Developer.php	2022. 05. 01. 18:27	PHP fájl	3 KB
 favicon.ico	2022. 05. 01. 18:27	Ikon	31 KB
 Game_page.php	2022. 05. 01. 18:27	PHP fájl	27 KB
 Genre.php	2022. 05. 01. 18:27	PHP fájl	5 KB
 ListOfUnacceptedData.php	2022. 05. 01. 18:27	PHP fájl	6 KB
 login.php	2022. 05. 01. 18:27	PHP fájl	3 KB
 loginPost.php	2022. 05. 01. 18:27	PHP fájl	1 KB
 MainPage.php	2022. 05. 01. 18:27	PHP fájl	1 KB
 MyGameList.png	2022. 05. 01. 18:27	PNG fájl	21 KB
 MyList.php	2022. 05. 01. 18:27	PHP fájl	8 KB
 mylistanddatasend.js	2022. 05. 01. 18:27	JS fájl	1 KB
 profil.php	2022. 05. 01. 20:00	PHP fájl	13 KB
 Publisher.php	2022. 05. 01. 18:27	PHP fájl	3 KB
 search.php	2022. 05. 01. 18:27	PHP fájl	2 KB
 signUp.php	2022. 05. 01. 18:27	PHP fájl	5 KB
 signUpPost.php	2022. 05. 01. 18:27	PHP fájl	3 KB
Szakdolgozat.docx	2022. 05. 01. 21:04	Microsoft Word-d...	17 025 KB
TopList.php	2022. 05. 01. 18:27	PHP fájl	7 KB
UnAcceptedData.php	2022. 05. 01. 18:27	PHP fájl	20 KB
uploadimages.php	2022. 05. 01. 18:27	PHP fájl	2 KB
verify.php	2022. 05. 01. 18:27	PHP fájl	3 KB
VoiceActorRequest.php	2022. 05. 01. 18:27	PHP fájl	1 KB

Játék adatainak megkapása .htaccess-el:

A játék oldalnak szüksége van az adott játék Id-jére és annak segítségével kéri le az adatbázisból az adatai, hogy megkapja ezt az adatot az oldal a .htaccess file segítségével a linkből beolvassa a játék Id-jét és azt továbbadja a Game_page.php-nak.

A .htaccess file játékoldalhoz használt kódja:

```
RewriteRule ^Game/([0-9]+)$ Game_page.php?GameId=$1 [QSA]
```

Ez a kódrész felel azért az oldalon, hogy a Game_page megkapja majd az általunk beírt játéknak az Id-jét.

Játék adatainak megkapása Game_page.php-ban:

Miután a .htaccess file-ból megkaptuk a GameId-t a Game_page.php oldalon belül erre hivatkozva hívjuk meg a rest-api file-t, ami majd egy változóban visszaadja Id alapján az adott játék összes adatát, ami az adatbázisba szerepel. Miután megkaptuk a játék adatait, a \$GameData tömb segítségével a játék bármely adatára tudnunk hivatkozni a php-n belül.

Játék értékelése:

Ha a felhasználó értékel egy játékot akkor az értékelt játék bekerül a felhasználójátéklistába az adatbázison belül. Itt eseményindítókkal (trigger-ekkel) megnézi az adatbázis, hogy létezik e ennek játéknak az értékelése a felhasználó által, és ha igen akkor kicseréli azt, ezután a listába szereplő játék értékeléseket összeszámolja, átlagot von és átállítja a játék átlag értékelését és az értékelések számát a megkapott eredményekre. Ezek a triggere-k lefutnak törlésnél, beillesztésnél és az adatok frissítésénél is.

Játék kedvencnek állítása:

Ha a felhasználó értékel egy játékot akkor az értékelt játék bekerül a felhasználójátéklistába az adatbázison belül. Itt eseményindítókkal (trigger-ekkel) megnézi az adatbázis, hogy létezik e ennek játéknak az értékelése a felhasználó által, és ha igen akkor kicseréli azt, ezután a listába szereplő játék értékeléseket összeszámolja,

átlagot von és átállítja a játék átlag értékelését és az értékelések számát a megkapott eredményekre. Ezek a triggere-k lefutnak törlésnél, beillesztésnél és az adatok frissítésénél is.

Toplista

Rövid leírás az oldalról:

A toplista oldal az összes játék listában való megjelenítésére szolgál, a listát lehet értékelés és értékelés szám alapján csökkenő sorrendben rendezni.

Toplista felhasználói dokumentáció

Itt a felhasználó a játékra nyomva megnézheti az adott játékot a játék oldalán. (A bekezdés alatti képek szemléltetésképpen a legjobban értékelt játékok listájáról majd a legtöbb értékeléssel rendelkező oldalról)

Top rated games		Most rated games	
Rank	Cover Image	Game Title	Average Rating
1		Last Of Us	9.50
2		Elder Scrolls V: Skyrim	7.00

Top rated games		Most rated games	
Rank	Cover Image	Game Title	Number Of Rating
1		Last Of Us	2
2		Elder Scrolls V: Skyrim	1

A rendezésben a felhasználó változtathat a táblázat fejlécében található gombokkal, az éppen aktív rendezést piros háttérrel jelöli az oldal.

Toplista fejlesztői dokumentáció

A weblaphoz tartozó file-ok:

A topista oldal a Toplist.php-t jeleníti meg a .htaccess file segítségével. De közben használja a css/Lists.css-t és a reszt-api mappa .htacces, Lists.php, ListsRestHandler.php és RestContorrel.php részét is.

Toplista adatainak megkapása .htaccess-el:

A Toplist oldalnak szüksége van a rendezés típusára és annak segítségével kéri le az adatbázisból a játékok adatai, hogy megkapja ezt az adatot az oldal a .htaccess file segítségével a linkből beolvassa a rendezést és azt továbbadja a Toplist.php-nak. A .htaccess file Toplisthez használt kódja:

```
RewriteRule ^TopList/([A-Za-z-]+)$ TopList.php?Order=$1 [QSA]
```

Ez a kód rész felel azért az oldalon, hogy a Toplist.php megkapja majd az általunk beírt rendezési módot.

Toplista adatainak megkapása Toplist.php-ban:

Miután a .htaccess file-ból megkaptuk az Order-t a Toplist.php oldalon belül erre hivatkozva hívjuk meg a rest-api file-t, ami majd egy változóban visszaadja Order alapján az adott játékok összes adatát, ami az adatbázisba szerepel. Miután megkaptuk a játékok adatait, a \$OrderByRatingData és a \$OrderByNumberOfRatingData tömb segítségével minden adatukra tudunk hivatkozni.

Genre lista

Rövid leírás az oldalról:

A genre oldal az összes játék, listában való megjelenítésére szolgál. A lista az adott műfaj-hoz kapcsolódó játékokat jeleníti meg értékelés szerint csökkenő sorrendben.

Genre felhasználói dokumentáció:

Itt a felhasználó láthatja melyik műfaj listájának oldalán van, ezt a piros háttér jelöli, ha a játékre nyomva megnézheti az adott játékot a játék oldalán.
(A bekezdés alatti képek szemléltetésképpen az akció játékok listájáról)

Action	Adventure	Puzzle	Role-playing	Simulation	Strategy	Sport
Rank	Cover image		Title		Rating	
1			Last Of Us		9.50	
2			Elder Scrolls V: Skyrim		7.00	

Genre lista fejlesztői dokumentáció

A weblaphoz tartozó file-ok:

A genre lista a Genre.php-t jeleníti meg a .htaccess file segítségével. De közben használja a css/Lists.css-t és a reszt-api mappa .htacces, Lists.php, ListsRestHandler.php és RestContorrel.php részét is.

Genre lista adatainak megkapása .htaccess-el:

A genre lista oldalnak szüksége van az adott műfaj Id-jére és annak segítségével kéri le az adatbázisból a játékok adatai, hogy megkapja ezt az adatot az oldal a .htaccess file segítségével a linkból beolvassa az Id-t és azt továbbadja a Genre.php-nak. A .htaccess file Genre-hez használt kódja:

```
RewriteRule ^Genres/([0-9]+)$ Genre.php?Genre=$1 [QSA]
```

Ez a kódrész felel azért az oldalon, hogy a Genre.php megkapja majd az általunk beírt műfaj Id-jét módot.

Genre lista adatainak megkapása Genre.php-ban:

Miután a .htaccess file-ból megkaptuk az adott műfaj Id-jét a Genre.php oldalon belül erre hivatkozva hívjuk meg a rest-api file-okat, amik majd egy változóban visszaadják Genre alapján az adott játékok összes adatát és az összes műfajt, ami az adatbázisba szerepel. Miután megkaptuk a játékok és műfajok adatait, a

\$GenreGameData és \$GenreData tömb segítségével azok minden adatára tudunk hivatkozni.

Fejlesztő és Kiadó lista

Rövid leírás az oldalról:

Az oldalak vagy a kiadók vagy a fejlesztők, listában való megjelenítésére szolgálnak. A lista az adott fejlesztő vagy kiadó cégeket jeleníti meg munkáinak száma szerint csökkenő sorrendben.

Fejlesztő és Kiadó lista felhasználói dokumentáció:

Itt a felhasználó láthatja, hogy a kiadó vagy a fejlesztő listájának oldalán van, ezt a piros háttér jelöli, ha a játékra nyomva megnézheti az adott cég oldalát.
(A bekezdés alatti képek szemléltetésképpen a fejlesztők listájáról)

Publishers		Developers	
Rank	Logo	Company name	Works
1		Bethesda Game Studios	4
2		Naughty Dog	1

Fejlesztő és Kiadó lista fejlesztői dokumentáció

A weblaphoz tartozó file-ok:

A fejlesztő és kiadó lista a DevAndPubList.php-t jeleníti meg a .htaccess file segítségével. De közben használja a css/Lists.css-t és a reszt-api mappa .htacces, Lists.php, ListsRestHandler.php és RestContorrel.php részét is.

Fejlesztő és Kiadó lista lista adatainak megkapása .htaccess-el:

A fejlesztő és kiadó lista oldalnak szüksége van vagy a fejlesztő vagy a kiadó megnevezésre és annak segítségével kéri le az adatbázisból az adott cégek adatai, hogy megkapja ezt az adatot az oldal a .htaccess file segítségével a linkből beolvassa, hogy a fejlesztő vagy kiadókra adatait akarjuk megkapni és azt továbbadja a DevAndPubList.php-nak. A .htaccess file DevAndPubList.php-hoz használt kódja:

```
RewriteRule ^DeveloperAndPublisherList/([A-Za-z]+)$  
DevAndPubList.php?DevOrPublisher=$1 [QSA]
```

Ez a kód rész felel azért az oldalon, hogy a DevAndPubList.php megkapja majd az általunk beírt opciót.

Fejlesztő és Kiadó lista adatainak megkapása DevAndPubList.php-ban:

Miután a .htaccess file-ból megkaptuk az adott cégeket a DevAndPubList.php oldalon belül erre hivatkozva hívjuk meg a rest-api file-okat, amik majd egy változóban visszaadják kiadó vagy fejlesztő alapján az adott cégek összes adatát ami az adatbázisba szerepel. Miután megkaptuk a cégek adatait, a \$DeveloperData és \$PublisherData tömb segítségével azok minden adatára tudunk hivatkozni.

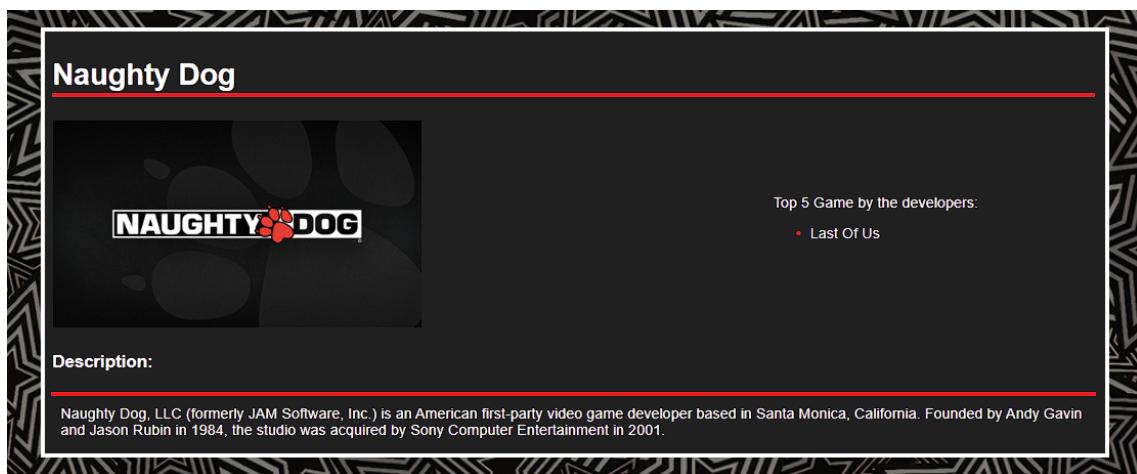
Fejlesztő és Kiadó oldal

Rövid leírás az oldalról:

Az oldal egy fejlesztő vagy kiadó adatainak megjelenítésére szolgál. Az oldal az adott cégeket jeleníti meg és a top 5 hozzá tartozó játékokat átlag értékelésük szerint csökkenő sorrendben.

Fejlesztő és Kiadó oldal felhasználói dokumentáció:

Itt a felhasználó láthatja a cég nevét, logóját, leírását, és a top5 hozzá tartozó játékokat, ezekre nyomva átirányítja az adott játék oldalára.
(A bekezdés alatti képek szemléltetésképpen egy fejlesztő oldaláról)



Fejlesztő és Kiadó oldal fejlesztői dokumentáció

A weblaphoz tartozó file-ok:

A fejlesztő oldal a Developer.php-t vagy Publisher.php-t jeleníti meg a .htaccess file segítségével. De közben használja a css/DeveloperPublisher.css-t és a rest-api mappa .htacces,GetDataByIdRestHandler.php,GetDataById.php,RelatedGamesRestHandler.php, RelatedGames.php és RestContorrel.php részét is.

Fejlesztő és Kiadó oldal adatainak megkapása .htaccess-el:

A fejlesztő és kiadó oldalnak szüksége van vagy a fejlesztő, vagy a kiadó Id-jére és annak segítségével kéri le az adatbázisból az adott cég adatait, hogy megkapja ezeket az adatokat az oldal a .htaccess file segítségével a linkből beolvassa, hogy a fejlesztő vagy kiadó adatait akarjuk megkapni és azt továbbadja vagy a Publisher.php vagy Developer.php-nak. A .htaccess file Developer.php-hoz használt kódja:

```
RewriteRule ^Developer/([0-9]+)$ RestController.php?DeveloperId=$1 [nc, qsa]
```

A .htaccess file Publisher.php-hoz használt kódja:

```
RewriteRule ^Publisher/([0-9]+)$ RestController.php?PublisherId=$1 [nc, qsa]
```

Ez a kódrész felel azért az oldalon, hogy a Developer.php vagy a Publisher.php megkapja majd az általunk beírt Id-t.

Fejlesztő és Kiadó oldal adatainak megkapása Developer.php és Publisher.php-ban:

Miután a .htaccess file-ból megkaptuk az adott cég Id-jét az adott oldalon belül erre hivatkozva hívjuk meg a rest-api file-okat, amik majd egy változóban visszaadják a kiadó vagy fejlesztő alapján az adott cégek és hozzá tartozó top 5 játéknak összes adatát, ami az adatbázisba szerepel. Miután megkaptuk a cég és a játékok adatait, a \$DeveloperData vagy \$PublisherData és \$RelatedGamesData tömb segítségével azok minden adatára tudunk hivatkozni.

Sablon

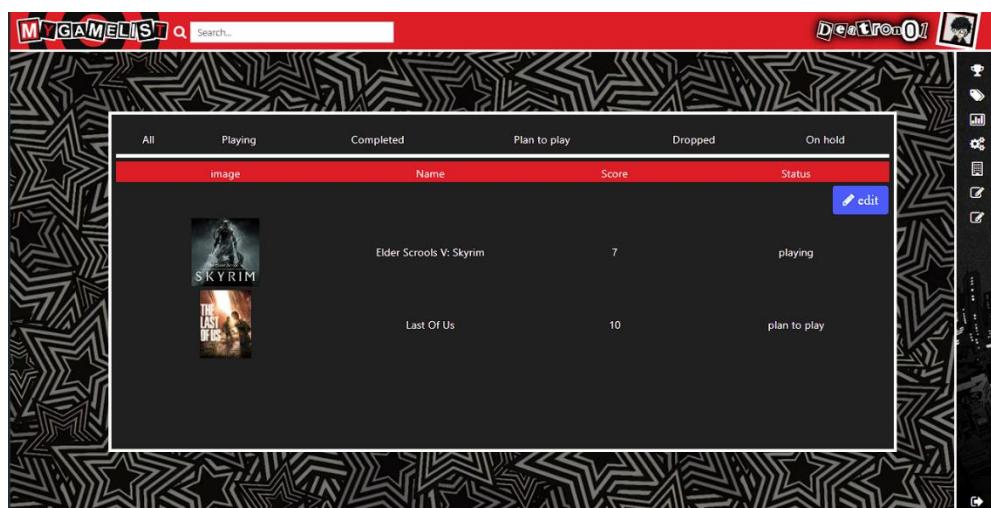
Rövid leírás az oldalról:

A sablon egy olyan oldal, amit minden másik php oldalunkon megkell hívni mivel ebben szerepel az alap weblap kinézete a keresés és a főmenük.

Sablon felhasználói dokumentáció:

A főmenün helyezkedik el az oldal logója, a keresési funkció, a bejelentkezés vagy regisztráció, ha bevan jelentkezve a felhasználó akkor a profilkép és a felhasználónév látszik helyette. A jobb oldali menün helyezkednek el az oldalaink, mint például a toplisták, saját lista, műfajok, ha a felhasználónak olyan jogai vannak akkor az adatfeltöltés és adatfeltöltés elfogadásának oldala, és a kijelentkezés a bejelentkezett felhasználóknak. Ezekre rányomva érhetik el a weboldalunk aloldalait.

(A bekezdés alatti kép szemléltetésképpen a sablonról a saját lista oldalon, bejelentkezett állapotban “Admin” jogú felhasználónál)



Sablon főbb funkciói:

- Keresés
- Profil elérése
- Navigáció az oldalak közt
- Regisztráció és Bejelentkezés elérése
- Kijelentkezés

Jobb oldali menü oszlop:

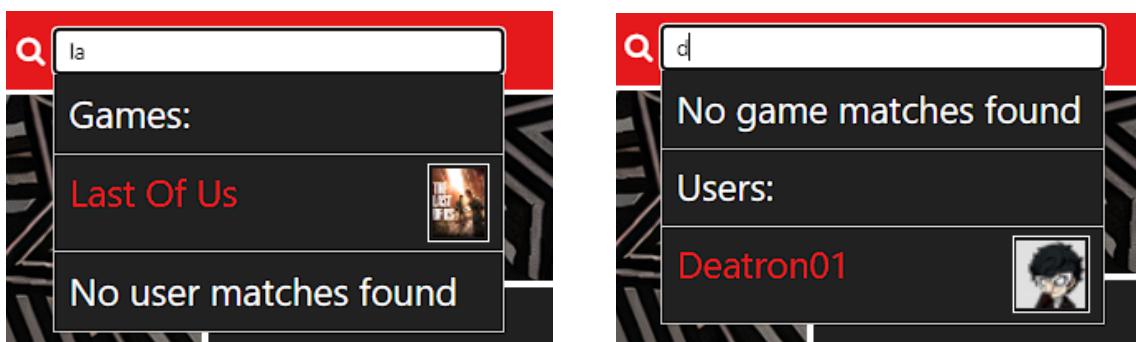
A menürész szolgál a többi oldal elérésére. Ha a felhasználó felé viszi az kurzort a menü kinyílik és átláthatóbban megtekinthetővé válik a többi oldal és azoknak gombjai.

(A bekezdés alatti kép szemléltetésképpen a sablonról a toplista oldalon, bejelentkezett állapotban “Admin” jogú felhasználónál a menü fölé vitt kurzorral)

All	Playing	Completed	Plan to play	Dropped	On hold												
	<table border="1"> <thead> <tr> <th>image</th> <th>Name</th> <th>Score</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td></td> <td>Elder Scrolls V: Skyrim</td> <td>7</td> <td>Playing</td> </tr> <tr> <td></td> <td>The Last Of Us</td> <td>10</td> <td>plan to play</td> </tr> </tbody> </table>	image	Name	Score	Status		Elder Scrolls V: Skyrim	7	Playing		The Last Of Us	10	plan to play				
image	Name	Score	Status														
	Elder Scrolls V: Skyrim	7	Playing														
	The Last Of Us	10	plan to play														

Keresés:

A keresés nevéből adódóan a keresésre szolgál, ezen belül játékok és felhasználók keressésére. Ha a felhasználó beír a kereső mezőbe egy szöveget akkor valós időben láthatja a keresésnél eredményét. Ha esetleg nem talált az adatbázisban a keresett szövegrészre eredményt akkor azt a felhasználó felé jelzi, jelen esetben a “No user matches found”-al, miszerint a keresett “la” szövegrésszel kezdődő felhasználónévre nem talált az adatbázisban eredményt. (A bekezdés alatti bal kép szemléltetésképpen a “la” szó keresésénél kiadott eredményt mutatja a jobb oldali kép a “d” szövegre kiadott eredményeket mutatja)



Sablon fejlesztői dokumentáció

A weblaphoz tartozó file-ok:

A sablon oldal a sablon/sablon.php-t jeleníti meg a .htaccess file segítségével. De közben használja a css/style.css-t és a kereséshez a search.php-t.

Keresés funkció:

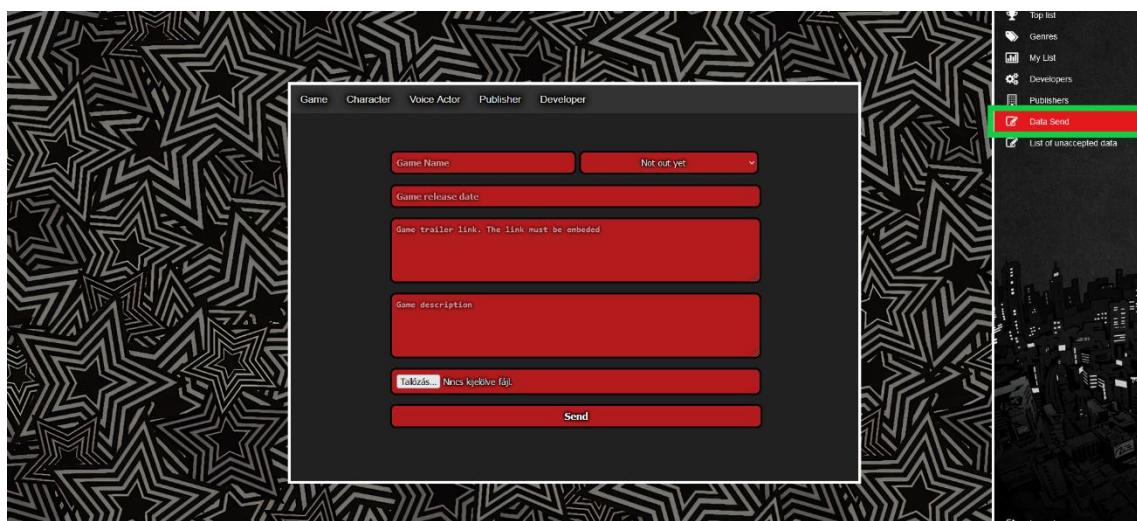
A keresési funkció-hoz javascriptet ezen belül jquery-t és php-t használunk. A jquery része felel azért, ha a keresési mezőbe írunk akkor a billentyű felengedése után a benne lévő szöveget átadja a search.php-nak ami a megkapott szöveg alapján keres az adatbázisba majd az ebből megkapott paragrafust a jquery beilleszti a keresés alá így lesz látható a keresés eredménye. Ezek a paragrafusok linket tartalmaznak, amik az adott játék vagy felhasználó oldalára irányítanak át.

Adatbázis adat feltöltés

A weboldalon a felhasználónak lehetősége van arra, hogy kedvenc játékát, szinkronhangját, karakterét feltöltsen az adatbázisba. Az oldal jobb oldalán megtalálható egy „Data send” nevezetű fül, amely az erre specifikált feladatra használható. Ezen az oldalon? számú fül található, mindegyik a nevéhez kapcsolódó adatot tölti fel az adatbázisba

A következő pár lépésben bemutatom egy játéknak az adat feltöltését:

1. Lépés: menüpont kiválasztása

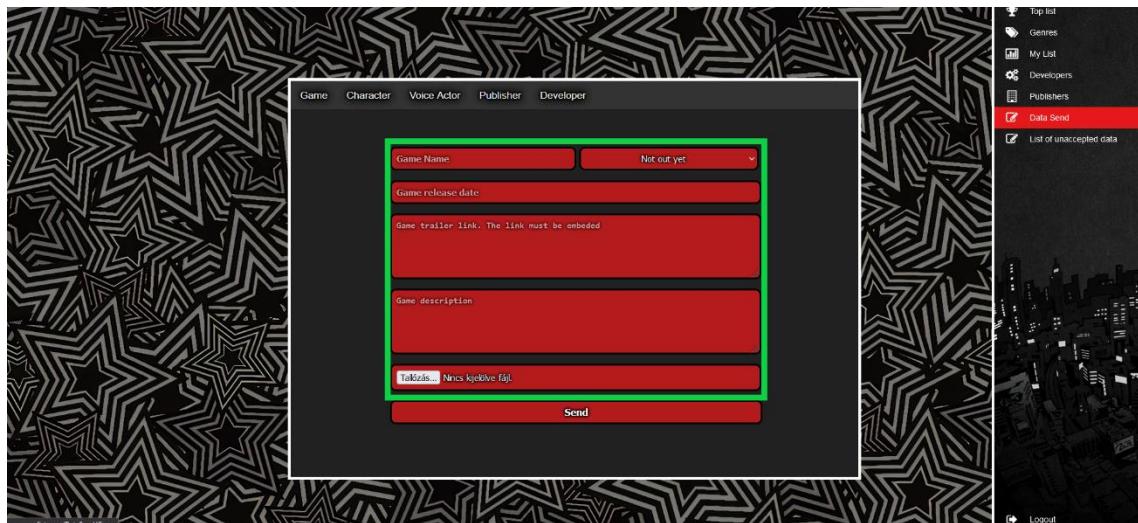


1. Data send menüpont

A egeret az oldal jobb oldalára kell helyezni ezt követően a felugró menüpontok közül ki kell választani a „Data send” menüpontot.

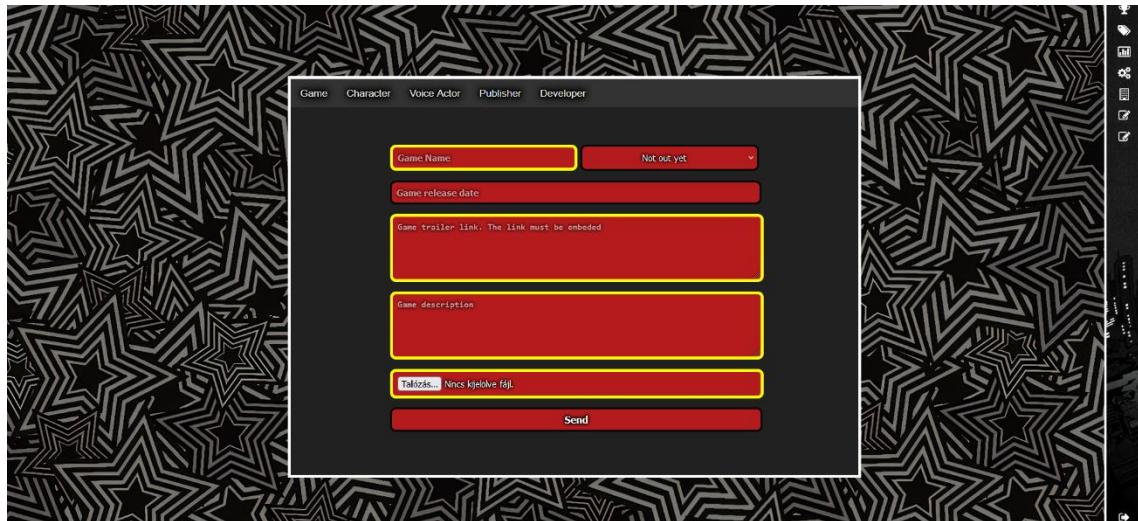
A menüpont lenyomást követően az oldal átirányítja a „Data send” oldalára (Ezen az oldalon lehet feltölteni adatot az adatbázisnak a megfelelő táblájába). A menüpont minden legelőször a „Game” menüpontra fogja átirányítani.

2. Lépés: Adatok kitöltése



2. Adat kitöltés

Tölts ki a megfelelő mezőket. Az ürlapban lévő összes adatot ki kell tölteni. Azonban van egy kivétel és ez a kivétel a „Game Status”. A „Game status”-nak a következő értékei lehetnek: **Out in early access, Not out yet, Released, Canceled** az alap értéke a „**Not out yet**”. Abban az esetben ha a felhasználó a **Not out yet**-et vagy a **Canceled** opciót választja akkor a „Game release date”-et nem kell kitölteni.

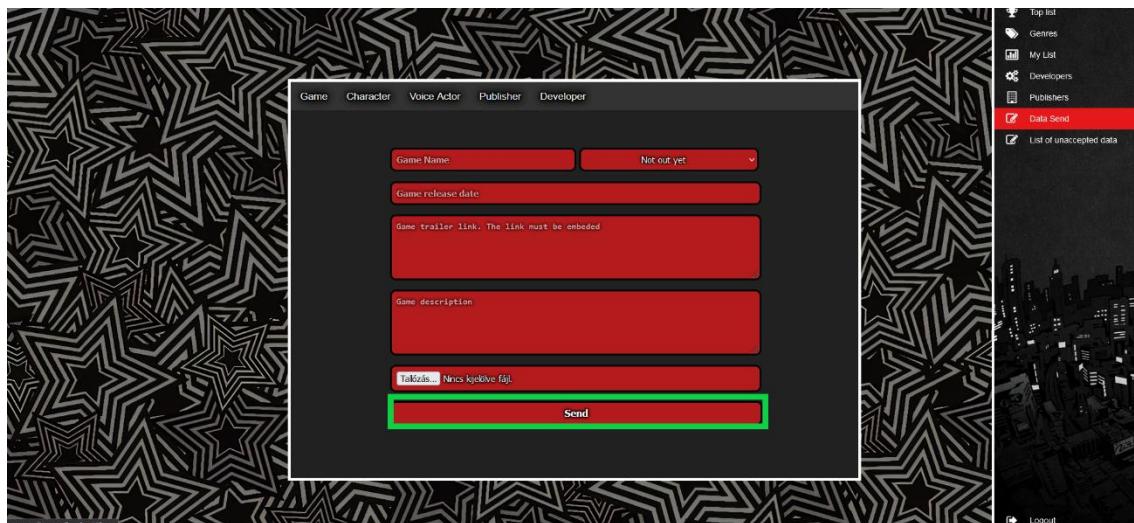


3. Adat ellenőrzés

Az űrlap képes kommunikálni a felhasználóval. Ez azt jelenti, hogy addig amíg nem töltötte ki az űrlapot helyesen addig nem engedi elküldeni a felhasználó adatait. A mezők

egy sárga keretet kapnak amennyiben nem megfelelő a mező vagy üres. A nem megfelelő itt azt jelenti, hogy például a link, amit bemásolt a felhasználó nem beágyazott link. A „Game release date” kerete akkor vált sárgára amikor a „game status” értéke **Not out yet** vagy **Canceled** és emellett a „Game release date”-nek beállítottunk dátumot. Ugyanez következik be abban az esetben is ha a „Game status” értéke **Out in early access** vagy **Released** és nem állítottunk be semmilyen dátumot.

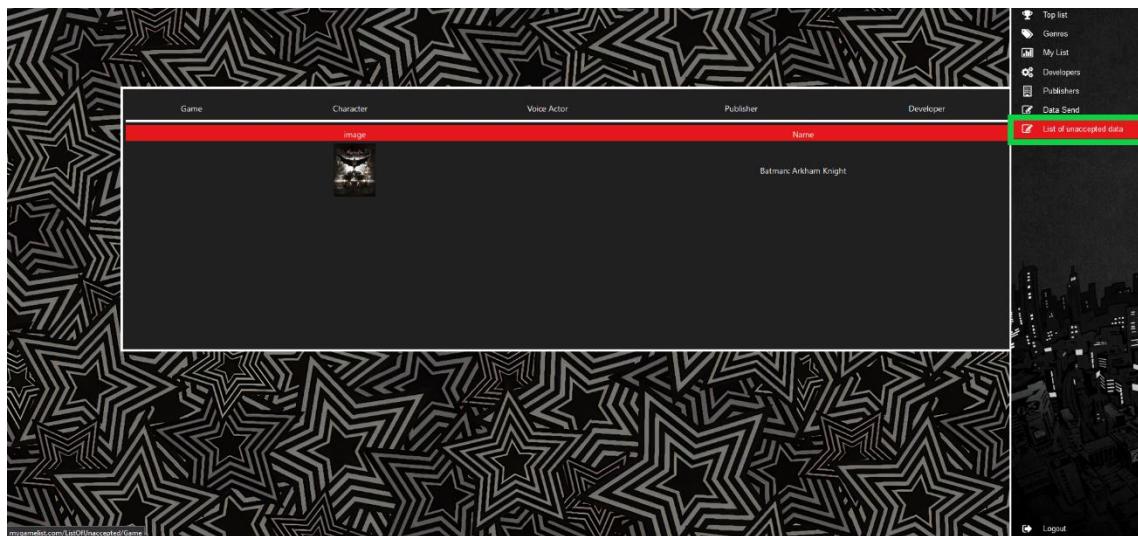
3. Lépés: Küldés gomb



4. Adat küldés

Nyomjon rá a „Send” gombra. Ezeket az adatokat egyelőre még nem lehet látni más oldalakon ameddig az admin el nem fogadja őket.

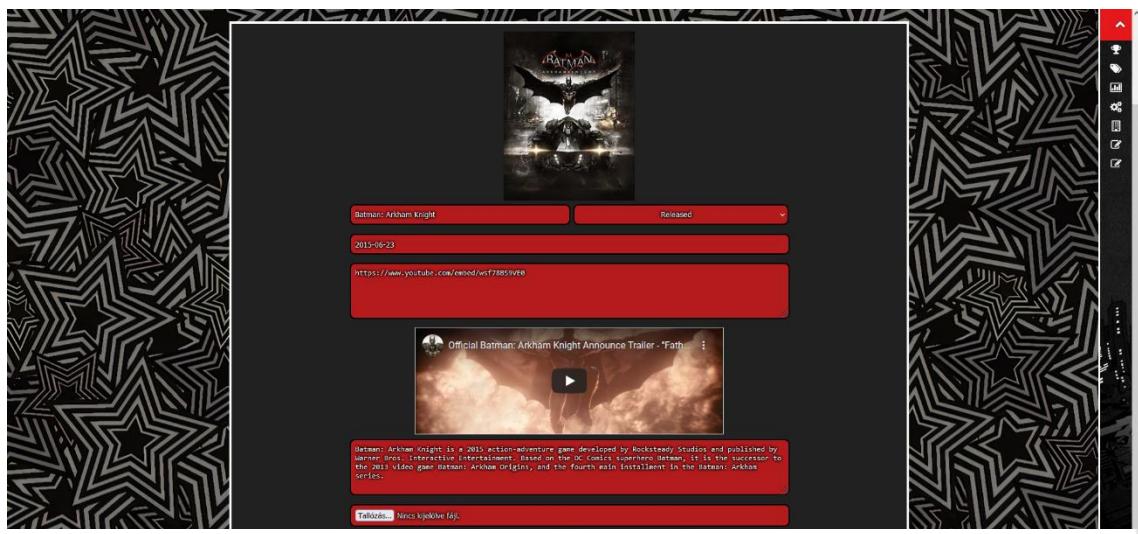
4. Lépés (További lépésekhez admin jogosultság kell)



5. Adminok által még el nem fogadott adatok listája

Hasonlóan az első lépéshoz az egeret megint az oldal jobb oldalára kell helyezni majd kiválasztani a „List of unaccepted data” menüpontot. A menüpont lenyomást követően az oldal átirányítja a „List of unaccepted data” oldalára (Ezen az oldalon az adminok fogják végig nézni a felhasználók által elküldött adatokat). A menüpont mindenlegelőször a „Game” menüpontra fogja átirányítani.

5. Lépés: adatok ellenőrzése

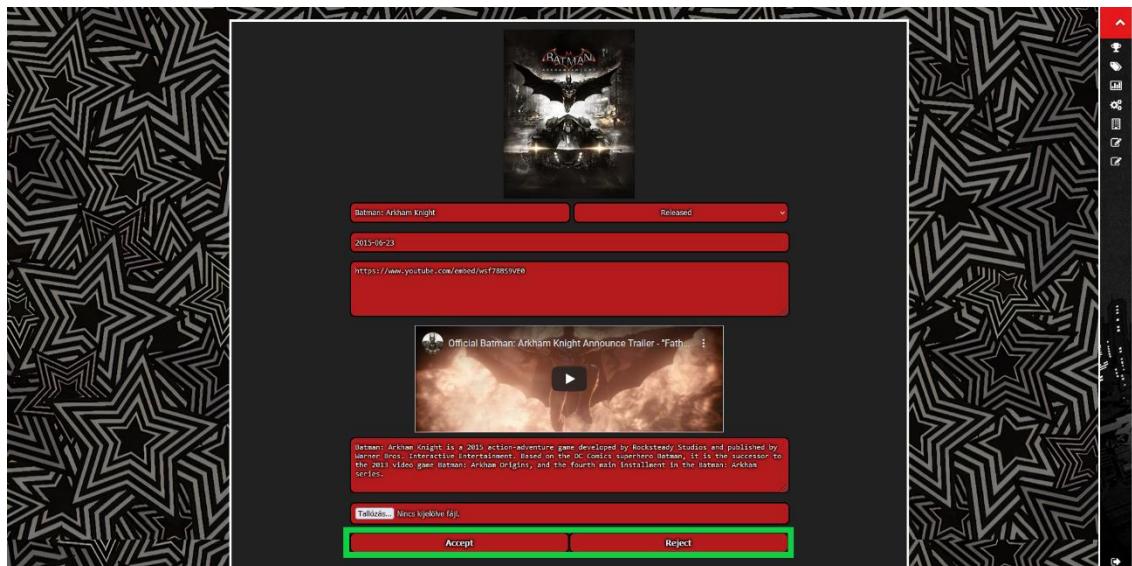


6. Admin adatai

Kattintson annak a játéknak a nevére, amivel foglalkozni szeretne. A listában a név egy linként szolgál, amelynek kattintása után az oldal átirányítja egy külalakban hasonló

oldalra, mint amely megtalálható a „Data send” menüpontban azonban itt az admin egy már adatokkal kiegészített oldalt kap meg. Ezek azok az adatok, amelyeket a felhasználó adott meg. Az űrlap itt is ellenőrzi a mezőket, hogy az admin minden helyesen töltött ki.

6. Lépés: Elfogadás vagy elutasítás

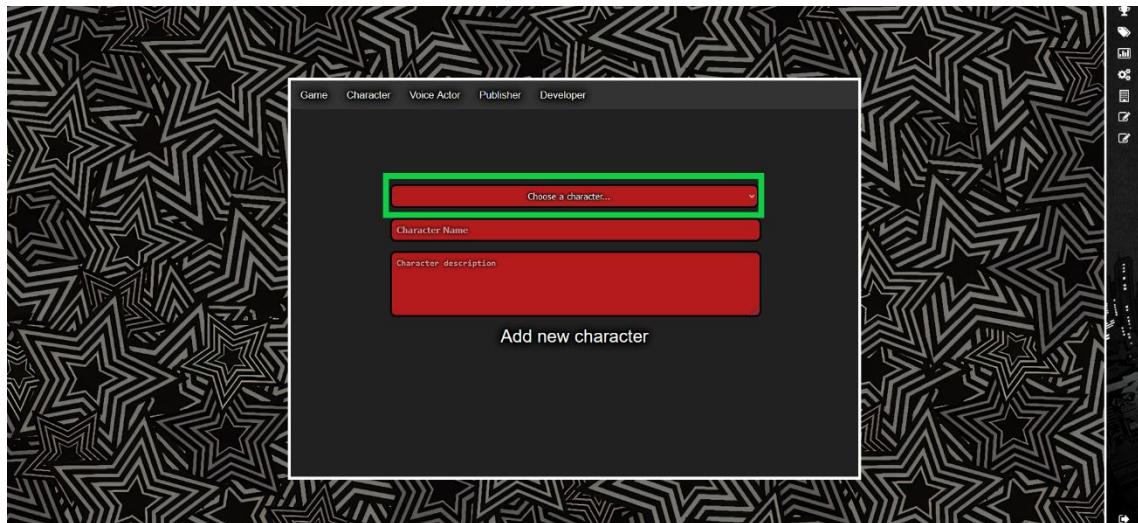


7. Admin elfogadja az adatot vagy nem

Amelyben az adatokat rendben találta kattintson az „Accept” gomb-ra, ezzel ezeket az adatokat kitöröli a „game_request” táblából és feltölti újként a „game” táblába. Amennyiben nem találta megfelelőnek az adatokat dönthet úgy, hogy megváltoztatja azt és utána megnyomja az „Accept”-et vagy ha esetleg úgy véli, hogy a felhasználó nem játskot, hanem valamilyen más adatot töltött fel ebben az esetben nyomjon a „Reject”-gombra.

Játék Adat feltöltés

Az adminoknak lehetőségeük van az általuk kiválasztott játéknak a következő típusú adatokat füldözni: **developer**, **publisher**, **character**, **genre**, **voice actor**. Ahoz hogy ezek az adatok hozzá legyenek adódva a játékhoz rá kell nyomni a megfelelő „add” gombra. Ezután a weboldal átirányítja az oldalt az ahhoz megfelelő oldalra. Ezen az oldalon a már az adminok által elfogadott adatokkal lehet dolgozni. Az oldalon legelőször

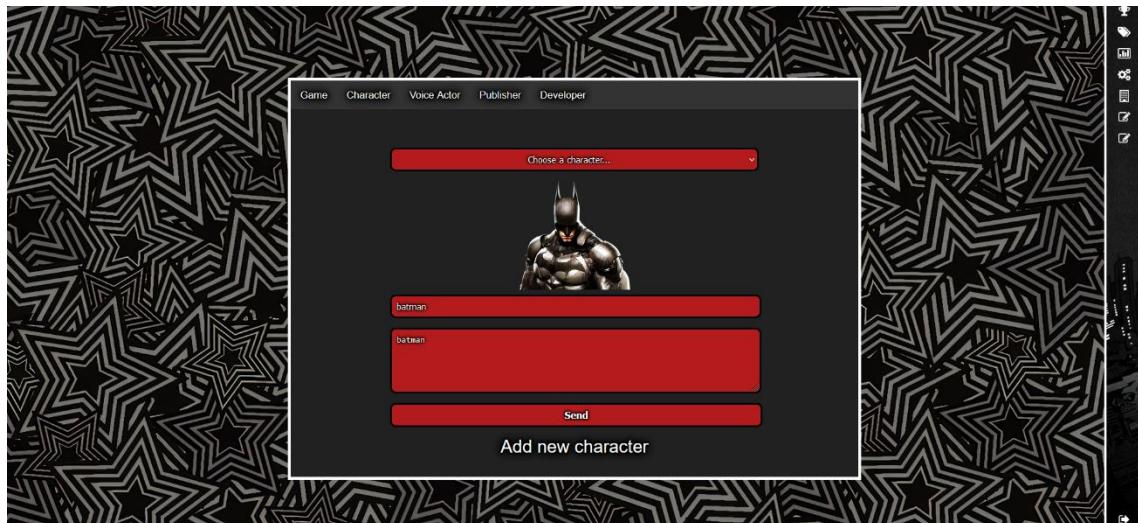


8. Karakter kiválasztása

ki kell választani a megfelelő adatot a listából.

Miután az admin kiválasztott egy karaktert azután az adott karakternek az adatai lesznek láthatók. Ezzel megbizonyosodhatnak arról, hogy biztosan azt az adott adatot szeretnék hozzáadni.

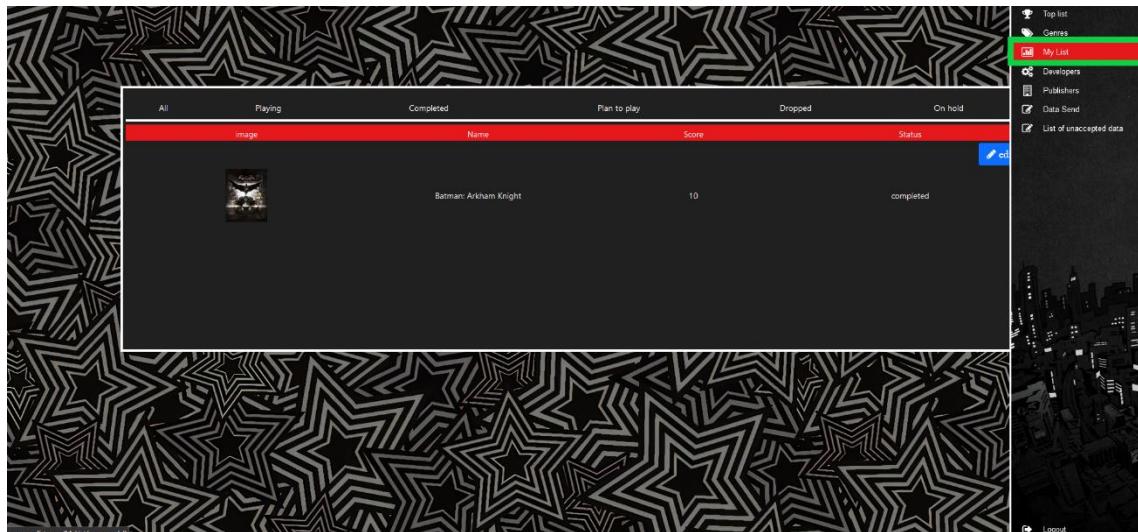
Ezután a „Send” gombra kell rányomni és utána hozzáadja az adott játékhoz.



9. Karakter adatai

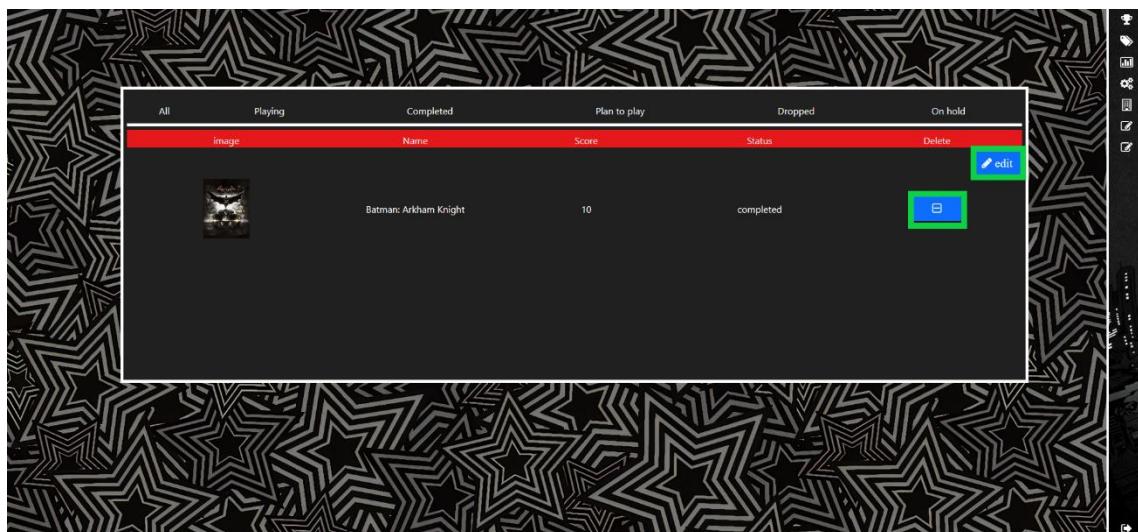
MyList

Az oldalon lehetősége van a felhasználónak arra, hogy a már értékelt játékokat egy lista formájában megtekinthesse. A listában a felhasználónak lehetősége van csoportosítani a már a felhasználó által megadott státusz alapján az értékelt játékokat. A lista arra is figyel, hogyha esetleg a felhasználó szeretné újra értékelni az adott játékot, akkor a listában lévő névre kattintva ezt megteheti. A nevek az oldalt átirányítják az adott játéakra.



10. Felhasználó listája

A Mylist-ban lehetősége van a felhasználónak a törlésre is. Ezt a funkciót akkor lehet alkalmazni, ha a felhasználó rágatott a „Delete” gombra. Ezt követően meg fog jelenni egy ikon, amelynek megnyomása követően a játék törlődni fog a listából.



11. Saját listából törlés

Adatküldési és myList Fejlesztői dokumentáció

Weboldalon használt programok

DataSend.php: Ennek a programnak a segítségével töltjük fel a felhasználó által elküldött adatokat. Ezeket az adatok először a „game_request” táblába töltjük fel. A program addig nem küldi el az adatokat, amíg a „Game name” és a „Game status” nincs kitöltve. Az oldalon használtunk htacces-t melynek segítségével az url a menüpontoknak megfelelő formában jelenik meg. Például a „Character” menüpont kattintása az url-ben a következő képpen jelenik meg: /Data_send/Character. A linkekkel \$_GET[‘dataType’]-ként tudom a programban lekérdezni. Ezeket a linkekkel felhasználva irányítom át a felhasználót egyik menüpontból a másikba.

DataSend.html: Ez a html a DataSen.php-hez tartozik. Ez jeleníti meg az alap html elemeket, mint például a menüpontok. A menüpontok egy linkként szolgálnak, amelyek a htacces-nek megfelelő formátumú linkre irányítanak.

ListOfUnacceptedData.php: Ez a program kilistázza az adminoknak azokat az adatokat, amikkel még nem foglalkoztak. A program a game_request táblából olvassa ki az adatokat. A listának a nevei a saját id-jük alapján egy linkként szolgálnak. Ezek a linkek átirányítják az adminot az UnacceptedData.php-ra. A linkek a következő url linkre mutatnak: Ha vesszünk például a Character listában egy elemet akkor UnacceptedData/CharacterId/” A character-hez tartozó id” linkre irányít.

ListOfUnacceptedData.html: Ez a html a ListOfUnacceptedData.php-hez tartozik. Ez jeleníti meg az alap html elemeket, mint például a menüpontok. A menüpontok egy linkként szolgálnak, amelyek a htacces-nek megfelelő formátumú linkre irányítanak.

UnacceptedData.php: A program az url segítségével lekérdezi az id-nek megfelelő adatokat. A lekérdezés után az űrlapot kitölti az url-ben lévő id alapján. Tehát az űrlapban lévő mezőket kiegészíti adatokkal. Az admin bármelyik mezőn képes változtatni. Amennyiben az admin az „Accept”-re nyom abban az esetben a megadott adatot kitöröli a game_request táblából és ezután feltölti a game táblába. Azonban, ha a „Reject” gombra kattint, akkor szimplán kitöröli a game_request táblából a megfelelő adatot.

UnacceptedData.html: Ez a html az UnacceptedData.php-hez tartozik.

MyList.php: Ez a program a „gamelist” táblából olvassa ki a megfelelő adatokat és abból készít egy listát. Ebben a listában megtalálható az összes olyan játék, amit a felhasználó már értékelte és vagy később értékelni akar. A lista kiírja a játéknak megadott pontszámot, amit megadott a felhasználó, a játék nevét és a játék képét. A listát lehet csoportosítani a felhasználó által megadott állapottal. Ez az állapot a következőket tartalmazhatja: **completed, playing, plan to play, dropped, on hold**. Ezekben kívül van még egy **all** amely az összes játékot kilistázza. A listában a nevek egy linkként szolgálnak, amelyek a hozzátartozó játéknak az oldalára visz.

MyList.html: Ez a html az myList.php-hez tartozik. Ez jeleníti meg az alap html elemeket, mint például a menüpontok. A menüpontok egy linkként szolgálnak, amelyek a htacces-nek megfelelő formátumú linkre irányítanak.

Tesztelés

```
C:\xampp>htdocs>hibaTesztele> Teszt.php
1 <?php
2 require '../sablon/sablon.php';
3 use PHPUnit\Framework\TestCase;
4
5 class Teszt extends TestCase
6 {
7     private $get;
8     protected function setUp(): void{
9         $this->get = new UAccData();
10    }
11    public function testGetCharacterName():void{
12        $result = $this->get->GetAll("characters_request")[0]["character_name"];
13        $this->assertEquals('asd', $result);
14    }
15    public function testGetCharactersVoiceActor():void{
16        $result = $this->get->GetAll("characters_request")[1]["voice_actor_name"];
17        $this->assertEquals('Kevin_Conroy', $result);
18    }
19    public function testGetGameName():void{
20        $result = $this->get->GetAll("game_request")[0]["game_name"];
21        $this->assertEquals('ads', $result);
22    }
23 }
24 ?>

...
3 / 3 (100%)
Time: 00:00.004, Memory: 18.00 MB
OK (3 tests, 3 assertions)
C:\xampp\htdocs\hibaTesztele>C:\xampp\php\php.exe phpunit.phar DataSendTeszt.php
```

```
<?php

require '../sablon/sablon.php';
use PHPUnit\Framework\TestCase;

class GamePageTest extends TestCase
{
    private $TestGame;
    protected function setUp() : void
    {
        $_GET["GameId"] = 1;
        $this->TestGame = new GameData();
    }

    public function testGetGameName():void
    {
        $TestValue = $this->TestGame->GetGameData("game_name");
        $this->assertEquals("Skyrim",$TestValue);
    }
    public function testGetGameID():void
    {
        $TestValue = $this->TestGame->GetGameData("game_id");
        $this->assertEquals(1,$TestValue);
    }
    public function testCheckIfItsAddedAlready():void
    {
        $TestValue = $this->TestGame->CheckIfItsAddedAlready();
        $this->assertEquals(0,$TestValue);
    }
}

?>
```

```
...
3 / 3 (100%)
Time: 00:00.008, Memory: 18.00 MB
OK (3 tests, 3 assertions)

C:\xampp\htdocs\Test>C:\xampp\php\php.exe phpunit.phar GamePageTest.php
```