



**BAJAI SZAKKÉPZÉSI CENTRUM**  
**KALOCSAI DÓZSA GYÖRGY TECHNIKUM ÉS KOLLÉGIUM**  
6300 KALOCSA, Asztrik tér 5-7.  
OM azonosító: 203028  
Tel.: 36-78-461-600 Fax.: 36-78-461-178  
E-mail: [info@dozsakalocsa.hu](mailto:info@dozsakalocsa.hu) [www.dozsakalocsa.hu](http://www.dozsakalocsa.hu)

---



**Szoftverfejlesztő és -tesztelő**

## **Dokumentáció**

**Képzőhely neve és címe: Bajai SZC Kalocsai Dózsa György Technikum és Kollégium**

**Vizsgázók nevei: Gábor László Roland, Kunsági Virág, Szili Nóra**

**Vizsga helyszíne: 6300 Kalocsa, Asztrik tér 5-7.**

**2025.**

## Nyilatkozatok

Alulírottak Kunsági Virág (Kalocsa, 2005.08.10), Gábor László Roland (Budapest XIV. kerület, 2005.10.05.), Szili Nóra (Kalocsa, 2006.04.19.) jelen nyilatkozat aláírásával kijelentjük, hogy a Szoftverfejlesztő és –tesztelő szakmai vizsgára készített program és dokumentáció saját munkánk eredménye. Nyilatkozunk, hogy az egyes dokumentumokban foglalt tények és adatok a megadott forrásmunkák felhasználásán alapulnak.

Tudomásul vesszük, hogy amennyiben a dokumentációmban a vizsgabizottság tagjai olyan dokumentumot találnak, amely problémaorientált feladat esetén 20%-nál nagyobb mértékben nem önálló munkánk eredménye vagy leíró/összegző feladat esetén nem saját kutatásunk eredménye, és amelyből a források kiválasztásának gondolatmenten vagy relevanciája nem követhető, valamint nem tartalmazza a pontos forrásmegjelöléseket, abban az esetben a vizsgabizottság tagjai a dokumentációt nem fogadják el. Ezen pont alól kivételt képeznek a csoportos projektek, komplex feladatok és esettanulmányok, melyek keletkezési körülményeit a bevezető tartalmazza.

Jelen nyilatkozat aláírásával tudomásul vesszük, hogy amennyiben bizonyítható, hogy a dokumentációt nem magunk készítettük, a dokumentáció bemutatása nem kezdhető meg, így ez a vizsgarész elégtelen értékelést kap.

Hozzájárulunk ahhoz, hogy a vizsgabizottság tagjai, jegyzője és a *Vizsgaszervező* munkatársai a dokumentáció tartalmát megismerhessék.

Kijelentjük, hogy a dokumentációra kapott értékelésünket és dokumentációnkat további felhasználásra sem kinyomtatva sem pedig elektronikus úton nem továbbítom senkinek.

Dátum: 2025. március 15.

Aláírások

# Tartalomjegyzék

Nyilatkozatok.....	2
Ábrajegyzék.....	5
BEVEZETÉS .....	6
Téma ismertetése.....	6
Témaválasztás indoklása .....	6
A fejlesztőcsapat tagjainak feladatai .....	7
FEJLESZTŐI DOKUMENTÁCIÓ .....	8
Programspecifikáció .....	8
<i>A program általános leírása .....</i>	<i>8</i>
<i>Funkcionális követelmények.....</i>	<i>8</i>
<i>Nem funkcionális követelmények .....</i>	<i>9</i>
A projekthez használt szoftverek és programozási nyelvek ismertetése.....	10
MySQL - Adatbázis létrehozása és kezelése .....	10
XAMPP- Adatbázis létrehozása és kezelése .....	10
Npm - Backend programozás .....	11
Node.js - Backend programozás .....	11
Visual Studio Code - Fejlesztői környezet .....	12
JavaScript – Backend és Frontend programozás .....	12
HTML – Frontend programozás .....	13
CSS – Frontend programozás .....	13
SQL – Backend programozás .....	13
Rendszerkövetelmény .....	14
<i>Rendszeri követelmény, amelyen a weboldal stabilon fut: .....</i>	<i>14</i>
<i>Böngésző követelmény, amelyen a weboldal stabilon fut: .....</i>	<i>14</i>
<i>Szoftver követelmények, amelyek segítségével a weboldal stabilon fut: .....</i>	<i>14</i>
Telepítési útmutató .....	15
Visual Studio Code .....	15
XAMPP és adatbázis.....	15

<i>Node.js</i> .....	16
<i>Github Repository és forráskód</i> .....	16
Kódolásról – Algoritmusok a projektben .....	17
<i>A dinamikus kosár megoldása</i> .....	17
A projekt tesztelése .....	22
<i>User Acceptance Test (UAT)</i> .....	22
Fejlesztési lehetőségek - Hol fejlődhetne a projekt? .....	24
<i>Idegennyelvűség</i> .....	24
<i>Idegen pénznemek</i> .....	24
<i>Tranzakciók kezelése</i> .....	24
<i>Felhasználói profil személyreszabása</i> .....	24
<i>Mobilalkalmazás készítése iOS és Android platformokra</i> .....	25
<i>Kiegészítő funkciók</i> .....	25
Adatszerkezetek - Adatabázis terv ismertetése.....	26
FELHASZNÁLÓI DOKUMENTÁCIÓ.....	29
Telepítési útmutató .....	29
<i>Böngésző</i> .....	29
<i>Indító exe fájl</i> .....	29
<i>Felhasználói kézikönyv</i> .....	30
<i>Kereső menü</i> .....	31
<i>Rólunk menüpont</i> .....	32
<i>Hírek menüpont</i> .....	33
<i>Termékek menüpont</i> .....	34
<i>Felhasználó menüpont</i> .....	35
<i>Bejelentkezési oldal</i> .....	36
<i>Regisztrációs oldal</i> .....	37
<i>Kosár menüpont</i> .....	38
VÉGSZÓ ÉS KÖSZÖNETNYILVÁNÍTÁS.....	39

# Ábrajegyzék

1. ÁBRA ADATBÁZIS SZERKEZETE .....	26
2. ÁBRA FELHASZNÁLÓK TÁBLA .....	26
3. ÁBRA KATEGÓRIÁK TÁBLA .....	27
4. ÁBRA JÁTEKOK TÁBLA .....	28
5. ÁBRA FŐOLDAL MENÜSOR .....	30
6. ÁBRA FŐOLDAL KERESŐ MENÜ .....	31
7. ÁBRA FŐOLDAL RÓLUNK MENÜPONT .....	32
8. ÁBRA FŐOLDAL HÍREK MENÜPONT .....	33
9. ÁBRA FŐOLDAL TERMÉKEK MENÜPONT .....	34
10. ÁBRA FŐOLDAL FELHASZNÁLÓ MENÜPONT .....	35
11. ÁBRA FŐOLDAL NAVIGÁLÓ GOMBOK .....	35
12. ÁBRA BEJELENTKEZÉS OLDAL .....	36
13. ÁBRA REGISZTRÁCIÓS OLDAL .....	37
14. ÁBRA FŐOLDAL KOSÁR MENÜPONT .....	38

# BEVEZETÉS

## Téma ismertetése

Általános megfigyelés, hogy az emberek nagy része nem engedheti meg magának manapság, hogy teljes áron vegyen meg egy játékot. Emellett a szoftverek digitális megvétele egyre népszerűbb a hagyományos cd megvételével szemben. A mai világban egy szoftver- független attól, hogy milyen típusú - akkor profitál, ha digitális megvételi opcióval rendelkezik.

## Témaválasztás indoklása

Szeretnénk egy olyan platformot biztosítani a videójáték közösség számára, amely minden korosztály és digitális képességekkel rendelkező személy számára egyszerű, átlátható és nem utolsó sorban praktikus is.

Továbbá az oldal üzemeltetőinek szeretnénk egy olyan megbízható, letisztult környezettel szolgálni, ahol könnyedén képesek karbantartani a programot és kezelni az esetlegesen felmerülő problémákat.

## A fejlesztőcsapat tagjainak feladatai

### *Általánosságban a csoportról*

Természetesen meg voltak a fejlesztés során mindenkinek a szerepe és feladata, de nem ragaszkodtunk szigorúan ezekhez a szerepkörökhöz – a fejlesztés során sokszor cseréltünk szerepeket vagy dolgoztunk össze, ezzel segítve egymás munkáját.

### *Szili Nóra – Frontend és dokumentáció*

A regisztrációs, belépési, főoldali és admin oldali dizájnok elkészítése a feladatom, melyek teljes mértékben reszponzívak, dinamikus elemekkel, működő gombokkal és űrlapokkal rendelkeznek. Ezek mellett a projektet átfogó dokumentáció tervezése, szerkesztése és finomítása is az én feladatim közé tartozik.

### *Gábor László Roland - Backend*

A regisztrációs, belépési, főoldali és admin felületen az UPDATE, DELETE, CREATE parancsok összekapcsolása az adatbázissal, valamint ezek teljes működőképessé tétele, emellett a megírt API-hívások (kontrollerek) helyes útvonolainak megadása, az oldalak közötti navigálások tartozik még a feladataim közé.

### *Kunsági Virág – Dinamikusság/Backend*

A felületekhez kapcsolódó API-hívások kidolgozása, valamint a játékok létrehozásának (CREATE), módosításának (UPDATE) és törlésének (DELETE) kidolgozása. Az én feladatom még a dinamikusság megvalósítása, mint pl. a kosár összegének kiírása és annak változtatása vagy gombnyomásra a kosárba kerüljön egy termék.

# FEJLESZTŐI DOKUMENTÁCIÓ

## Programspecifikáció

### *A program általános leírása*

Szeretnénk megoldást nyújtani egy olyan platform létrehozásával ahol az eredeti ár helyett egy csökkentett ár fejében “átvehető” a szoftver egy olyan egyéntől, aki eredetileg birtokolta. Ennek egy webes felület lesz biztosítva. A weboldal kezelőinek egy másik, független felület lesz szolgáltatva A felhasználók a saját játékaikat feltölthetik, hogy mások megvásárolhassák őket, és a platform biztosítja a tranzakciók biztonságos lebonyolítását, valamint egy egyszerű, felhasználóbarát felületet.

### *Funkcionális követelmények*

#### *Felhasználói regisztráció és belépés*

A felhasználók regisztrálhatnak a weboldalon e-mail címük és jelszó segítségével. A felhasználói profilok tartalmazhatják a felhasználó e-mail címét, jelszavát.

#### *Játékok feltöltése és kezelése*

A felhasználók feltölthetik a már nem kívánt játékaikat a platformra.

A feltöltött játékok tartalmazzák a játék címét, kategóriáját, árát, valamint a játék leírását. A feltöltéshez szükséges a játék borítóképe és egyéb promóciós információk (ha elérhetők). Az adminok módosíthatják vagy törölhetik a feltöltött játékokat.

#### *Játékok keresése és böngészés*

A felhasználók a weboldalon kereshetnek a játékok között különböző szűrők segítségével:

Kategória (akció, kaland, sport, stratégia stb.)

Játék neve

Legnépszerűbb vagy legújabb játékok

A keresés és böngészés során részletes információk jelennek meg minden játékhoz: cím, leírás, ár, kategoria, feltöltő neve



### *Adminisztrációs felület*

Az adminisztrátorok kezelhetik a weboldal tartalmát, beleértve a felhasználói fiókok kezelését, a játékok moderálását, és a tranzakciók felügyeletét. Az admin felület lehetővé teszi a felhasználók jelentéseinek és problémáinak kezelését.

Az adminisztrátorok statisztikai adatokat kaphatnak a legnépszerűbb játékokról, a tranzakciók számáról, és a felhasználói aktivitásról.

### *Nem funkcionális követelmények*

#### *Teljesítmény:*

A weboldalnak biztosítania kell, hogy képes legyen kezelni a nagy forgalmat, különösen az adatfeltöltés és tranzakciók lebonyolításakor.

#### *Biztonság:*

A weboldalon tárolt összes érzékeny adat (pl. felhasználói adatok, jelszavak, tranzakciók) titkosítva kell legyenek. A tranzakciókat SSL titkosítás védi a biztonságos adatátvitel érdekében. Az adminisztrátori felületen és a felhasználói fiókokban lévő érzékeny információkhoz csak a megfelelő jogosultsággal rendelkező személyek férhetnek hozzá.

#### *Felhasználói élmény:*

A weboldalnak reszponzívnak kell lennie, hogy minden eszközön (asztali számítógép, tablet, mobil) megfelelően működjön. A felhasználói felületet úgy kell megtervezni, hogy intuitív és könnyen navigálható legyen. A vásárlási és eladási folyamatokat egyszerűsíteni kell, hogy zökkenőmentes élményt biztosítson a felhasználók számára.

## A projekthez használt szoftverek és programozási nyelvek ismertetése

### *MySQL - Adatbázis létrehozása és kezelése*

A MySQL egy ingyenes, rugalmas, nyílt forráskódú relációs adatbáziskezelő szerver (RDBMS), amely lehetőséget nyújt a felhasználói számára az SQL alapú adatbázisok létrehozására és kezelésére. Relációs szerkezetéből adódóan a rekordokat sorokban és oszlopokban (később mezőkben) tárolja el.



A tervezés alatt több különböző adatbáziskezelő szoftvert tekintettünk meg; egy olyat kerestünk, amely gyors, egyszerű, ugyanakkor nagyobb adatbázisok kezelésekor is megbízható; a MySQL tökéletes lehetőségeket nyújt ezekre. A rekordok egyszerűen változtathatók, törölhetők, hozzáadhatók.

### *XAMPP- Adatbázis létrehozása és kezelése*

Aki adatbázissal támogatott webszervert szeretne üzemeltetni, annak össze kell hangolnia a webszerver, az adatbázisszerver és a PHP szolgáltatásait, valamint a kommunikációs portokat. A XAMPP fejlesztői felismerték a lehetőséget, hogy ezeket a beállításokat mindenkinek el kell végeznie, viszont nem biztos, hogy mindenki rendelkezik adminisztrátori jogokkal és ilyen szintű ismeretekkel, ezért összeállítottak egy keretrendszert, amelyet kezdő fejlesztők is könnyedén használhatnak (elsősorban gyakorláshoz és a weboldalak prototípusainak fejlesztéséhez).<sup>1</sup>



---

<sup>1</sup> Forrás - [https://www.inf.u-szeged.hu/~gnemeth/adatbgyak/exe/MySQL\\_XAMPP\\_JDBC/a\\_xampp\\_keretrendszer\\_s\\_hasznlata.html](https://www.inf.u-szeged.hu/~gnemeth/adatbgyak/exe/MySQL_XAMPP_JDBC/a_xampp_keretrendszer_s_hasznlata.html)

## *Npm - Backend programozás*

Tanulmányaink során többféle csomagkezelő rendszerrel ismerkedtünk meg, mint pl. a NuGet, amely a .NET keretrendszer függőségeit képes kezelni, a Composer, amely egy PHP specifikus kezelőrendszer, a Gradle, amely kevésbé specifikus, de a Java és Android környezetben használt általánosságban, illetve az npm, amely egy JavaScript-orientált kezelőrendszer.



Az npm egy csomagkezelő, illetve egy központi hely a JavaScript- és Node.js-csomagok (modulok) nyilvántartására és kezelésére. Az npm lehetővé teszi a csomagok könnyű kezelését és frissítését, legyenek azok egy adott projekt helyi függőségei vagy globálisan telepített JavaScript-eszközök. Helyi projektek esetén az npm egyetlen parancs segítségével képes telepíteni az összes szükséges függőséget az általa már korábban létrehozott package.json fájl alapján. A package.json fájl egy automatikusan generált fájl, amely tartalmazza a projektünk metaadatait, mint a projektünk neve, az indítófájl neve, a telepített csomagok (modulok) verzióinak számát, stb. A package.json fájl mellett létrejön egy package-lock.json nevezetű fájl is, amely bármely új-illetve már meglévő modul változásakor szintén változik.

## *Node.js - Backend programozás*

A Node.js egy olyan keresztplatformos, nyílt forráskodú környezet, amely lehetőséget nyújt JavaScript kódok írására, szerkesztésére, futtatására. A Node.js lehetőséget biztosít ezek a kódok futtatására a számítógépen keresztül a korábbi böngészőn belüli futtatás helyett.



## Visual Studio Code - Fejlesztői környezet

A fejlesztőknek számos eszközt kell használniuk a tervezéshez, fejlesztéshez és teszteléshez. A fejlesztői környezetek olyan szoftverek, amelyek különböző funkciókat egyesítenek egyetlen alkalmazásban, megkönnyítve a fejlesztők munkáját, hatékonyabbá téve a folyamatot. A Visual Studio Code egy olyan ingyenesen használható fejlesztői környezet, amely tartalmaz kódszerkesztőt, hibakeresőt és értelmezőt (interpreter) is. Ezek alapvető elemek, de ezek mellett objektum- és adatmodellező eszközöket, forráskódkönyvtárakat is tartalmaz. A Visual Studio Code egyik legnagyobb előnye - mint ahogyan a legtöbb modern fejlesztői környezetnek is - hogy ezek a fellebb említett eszközök egy tiszta és egyszerű grafikus felhasználói felületen (GUI) keresztül érhetők el.



## JavaScript – Backend és Frontend programozás

A JavaScript egy objektum-orientált szkript alapú nyelv, mely leginkább a frontend programozásban alkalmazott. A nyelv segít dinamikus, interaktív weboldalak létrehozni, akár keretrendszerek segítségével, mint ReactJS, AngularJS, vagy VueJS. Kliens és szerver oldali fejlesztésben is helytálló; a kliens oldali fejlesztésben kezeli a felhasználó által űrlapra bevitt adatait, gombnyomásait. A szerver oldali fejlesztésben interaktálhat a kezelő az adatbázissal, módosíthatja őket és/vagy különböző fájlokat.



## HTML – Frontend programozás

A HTML a legalapvetőbb, egyszerűen kezelhetőbb frontend programozási nyelv. A HTML adja meg egy weboldal alapvető struktúráját, ezáltal olyan elemeket kínál, amelyek képesek manipulálni az oldalon megjelenített tartalom kinézetét bizonyos határokig. Ezt úgynevezett tagekkel oldja meg. Általában ezen tagek szintaxisa egy nyitó taggel kezdődik, amelynek nevét két relációs jel között adunk meg, majd a záró tagben ugyanígy, egy per jellel toldva.



## CSS – Frontend programozás

A CSS a weboldalak vizuális megjelenését segítő nyelv - Megváltoztathatod a szövegek színét, aláhúzott-e, milyen betűtípust használ.

Rengeteg előnye van CSS-t használni: Időt spórolunk - egy kijelölőt megírva egyetlen stílust több HTML elemen is használhatunk, több vizuális és kreatív lehetőséget nyújt a weboldal színesítésére, mint pl. az animációk használata, illetve a különböző okoseszközökhöz kompatibilitást nyújt. Ez azt jeletni, hogy egy számítógépes monitoron és egy mobiltelefonon más dizájn jelenik meg a könnyebb navigáció és esztétikus látvány érdekében.



## SQL – Backend programozás

Az SQL egy relációsadatbázis-kezelő lekérdezési nyelv. Bár nem tekintjük programozási nyelvnek, de mégis hasonlóan kell elképzelni, itt is vannak utasítások, amelyek segítségével műveleteket végezhetünk. Létrehozhatunk adatbázisokat, adatokat adhatunk az adatbázishoz, törölhetünk belőle, módosíthatjuk az egyes adatokat, lekérdezhetjük azokat, vagy rendezhetjük.<sup>2</sup>



---

<sup>2</sup>Forrás: <https://webiskola.hu/sql-ismeretek/relacios-adatbazis-sql-fogalmak-peldak/>

## Rendszerkövetelmény

*Rendszeri követelmény, amelyen a weboldal stabilon fut:*

### Windows PC

Operációs rendszer	Processzor	Memória
Microsoft Windows 10 Enterprise vagy magasabb	Intel(R) Core (TM) i3-8100 3.60GHz	4GB RAM minimum, vagy több

### Apple Mac

Operációs rendszer	Processzor	Memória
Mac OS X 10.9 vagy magasabb	Intel(R) Core (TM) i3-8100 3.60GHz	4GB RAM minimum, vagy több

*Mobiltelefonoknál a legtöbb eszköz kompatibilis.*

*Böngésző követelmény, amelyen a weboldal stabilon fut:*

Mozilla	Google	Opera	Edge
Mozilla Foundation Firefox 133 verzió	Google Chrome 131 verzió	Opera Software Opera 114 verzió	Microsoft Edge 131 verzió

*Szoftver követelmények, amelyek segítségével a weboldal stabilon fut:*

XAMPP	Node.js	npm	Visual Studio Code
v8.0.30 vagy magasabb	v18.20.6	v10.8.2	v1.95.3 vagy magasabb

## Telepítési útmutató

### *Visual Studio Code*

Töltsd le a Visual Studio Code telepítőt Windows-ra

Miután letöltötted, futtasd a telepítőt (VSCodeUserSetup- {verzió}.exe).

Alapértelmezetten a VS Code a következő helyre kerül telepítésre: C:\Users {Felhasználónév}\AppData\Local\Programs\Microsoft VS Code.

### *XAMPP és adatbázis*

A böngésző mellett szükség lesz még egy adatbáziskezelésre alkalmas keretrendszerre – A mi alkalmazásunk esetében ez a XAMPP. A megfelelő verziót a *Rendszerkövetelmények* pont *Szoftver követelmények, amelyek segítségével a weboldal stabilon fut* alpontjában található.

A XAMPP az alábbi linken letölthető: <https://www.apachefriends.org/hu/download.html> - kiválasztjuk a verziót, majd a kék *Letöltés (64 bit)*-es gombra kattintva a telepítőprogram letöltése elkezdődik.

Miután a telepítőprogram segítségével a program sikeresen telepítve lett, nyissuk meg, majd nyomjuk meg a megjelenő Apache és MySQL pontok melletti Start gombot. A MySQL Start pontja mellett ezek után elérhetővé válik az Admin feliratú gomb; nyomjuk meg, hogy a phpMyAdmin felülete megjelenjen.

A menüpontok közül válasszuk ki az Importálás menüpontot, ezt követően a megjelenő ablakban válasszuk ki az Fájl kiválasztása gombot. Böngésszük ki a Vizsgaremekhez csatolt .sql kiterjesztésű fájlt és importáljuk.

## Node.js

Ha az adatbázis és fejlesztői környezet sikeresen telepítve lett, szükségünk lesz a Node.js nevű keresztplatformos szoftverre. Látogass el a <https://nodejs.org/en> oldalára.

Itt két verzió közül választhatsz:

**LTS (Long Term Support):** Ez a verzió stabil, hosszú távú támogatást kap.

**Current:** A legújabb funkciókat tartalmazó verzió.

Általában az **LTS** verziót érdemes választani a stabilitás miatt, különösen, ha éles környezetben dolgozol.

Kattints a kívánt verzióra, hogy letöltsd az operációs rendszerednek megfelelő telepítőt. Miután letöltötted a telepítőt, futtasd a telepítő fájlt (.msi Windows esetén vagy .pkg macOS esetén).

Kövessd a telepítő varázsló lépéseit, és fogadd el az alapértelmezett beállításokat, ha nem vagy biztos benne, hogy mit kell választanod.

Kattints a **Telepítés** (Install) gombra, és várd meg, amíg a telepítés befejeződik.

## Github Repository és forráskód

A fejlesztéshez a forráskód is szükséges. A forráskód egy Github repository-ban van tárolva.

Látogass el a következő linkre: <https://github.com/sznora/vizsgaprojekt>, majd a zöld <Code> gombra kattintva töltsd le a zip fájlt. Csomagold ki, majd nyisd meg a Visual Studio Code-al a mappát.

Nyiss egy terminal ablakot, majd írd be az `npm update` parancsot, ez a package.json fájlban feltüntetett függőségek alapján telepíti a szükséges csomagokat (package), amelyeket a node.js használ a projekthez. A sikeres lefutás után a mappaszerkezetben egy új mappa fog megjelenni `node_modules` névvel. Ez tartalmazza az összes szükséges fájlt a csomagokhoz.

Ha ezekkel megvagy, írd be a `npm start` parancsot és már meg is jelenik a főoldal a 3000-es porton.



## Kódolásról – Algoritmusok a projektben

Kétségtelenül rengeteg klasszikus, már ismert algoritmus szerepel a projektben, de egy teljesen egyedi megoldású algoritmust szeretnénk részletesebben bemutatni.

### *A dinamikus kosár megoldása*

```
//Kosár feltöltése
let a2 = document.createElement("a");

if (user) {
  document.addEventListener("DOMContentLoaded", () => {
    osszegar = parseInt(localStorage.getItem("osszegar")) || 0;
    a2.innerHTML = `Összegár: ${összegar} Ft`;

    let storedItems = JSON.parse(localStorage.getItem("kosar")) || [];

    storedItems.forEach(item => {
      addToCart(item.title, item.price, item.summa);
    });

    if (összegar > 0) {
      ossztermek.appendChild(rendelesGomb);
    }
  });

  ossztermek.appendChild(a2);

  document.querySelectorAll(".product-card .gomb").forEach(button => {
    button.addEventListener("click", event => {
      let clickedCard = event.currentTarget.closest(".product-card");
      let cim = clickedCard.querySelector('.title').innerHTML;
      let ar = clickedCard.querySelector('.price2').innerHTML;
      let db = clickedCard.querySelector('.db').value || 1;

      // Elmentjük a terméket a LocalStorage-ba
      teljesAr = parseInt(ar) * parseInt(db);
      osszegar += teljesAr;
    });
  });
}
```

```

localStorage.setItem("összegar", osszegar);
a2.innerHTML = `Összegár: ${összegar} Ft`

let storedItems = JSON.parse(localStorage.getItem("kosar")) || [];
storedItems.push({ summa: teljesAr, title: cim, price: ar, quantity: db });
localStorage.setItem("kosar", JSON.stringify(storedItems));
addToCart(cim, ar, teljesAr);
ossztermek.appendChild(rendelesGomb);
});
});
}

```

Az fentebbi képeken a kosár feltöltése látható. Ha a felhasználó bejelentkezett, megjelenik a kosár menü, majd a LocalStorage-ból lekéri az összesített árat, majd betölti azt a kosár legelső sorába. Ha ez az összeg meghaladja a nullát, megjelenik a *“Rendelés”* gomb.

A foreach ciklusunk végigfut a termékeken (a HTML-ben ezek egy product-card osztállyal vannak ellátva) és figyeli, hogy a *“Kosárba”* nevezetű gombra rákattintott-e a felhasználó.

Ha igen, letároljuk a játék nevét, árát és hány darab került a a kosárba belőle egy clickedCard változóban. Ezek után frisstjük az összesített árat a termékek darabszáma és az egységárának szorzatával, majd letároljuk a LocalStorage-ban.

Kiíratjuk a frissített összegárat, illetve hozzátoldjuk a szükséges értékeket egy storedItems tömbhöz, amely belekerül a LocalStorage-ba, mint új termék. Az addToCart függvény a vizuális megjelenítéshez szükséges. A rendelesGomb újra hozzáadására azért van szükség, hogy mindig a kosárban lévő termékek alatt jelenjen meg.

```
// Új elem hozzáadása a DOM-hoz
function addToCart(title, price, summa) {
  const input = document.createElement("input");
  input.type = "button";
  input.value = " ✕ ";
  input.id = "trash";
  input.classList.add("torles");
  let a = document.createElement("a");
  a.innerHTML = `${title} ${summa} Ft`;
  a.classList.add("dropbtn", "termek");
  ossztermek.appendChild(a);
  a.appendChild(input);
}
```

A fentebbi képen a hozzáadott termék vizuális megjelenítésére írt algoritmus látható. Ha meghívjuk, a felsorolásban egy input elem jelenik meg, ezzel kezeltük le, hogy törölhetőek legyenek az elemek.

A szükséges értékeket megkapja, majd megjeleníti, illetve hozzáad egy *“Törlés”* osztályt, amely a későbbiekben szükséges lesz az elemek törlésének függvényéhez.

```

function Torles(event) {
  let clickedButton = event.target;
  let clickedItem = clickedButton.closest(".termek");

  if (clickedItem) {
    let priceText = clickedItem.innerHTML.match(/(\d+)\s*Ft/);
    if (priceText) {
      let price = parseInt(priceText[1]);

      osszegar -= price;
      localStorage.setItem("osszegar", osszegar);
      a2.innerHTML = `Összegár: ${osszegar} Ft`;

      let storedItems = JSON.parse(localStorage.getItem("kosar")) || [];
      storedItems = storedItems.filter(item => item.summa !== price);
      localStorage.setItem("kosar", JSON.stringify(storedItems));
    }

    clickedItem.remove();
  }
}

document.addEventListener("click", function (event) {
  if (event.target.classList.contains("torles")) {
    Torles(event);
  }
});

```

Ez a kosárelemek törlésének algoritmusa.

A függvény először létrehoz egy ClickedButton változót a törölni kívánt termék törlés ikonjához. Megnézi mi a legközelebbi letárolt játék, majd, ha ez létezik, megkeresi a sztringként tárolt korábban említett szorzatot. Ha ez létezik, kiszűri a sztringből a számokat, amely jelen esetben az ár és levonja a végösszegeből.

Megkapja egy StoredItems tömbbe a kosárban lévő jelenlegi összes terméket, kiszűri az összes terméket, amely nem egyezik meg a törölt árral és ezeket visszaadja a kosárnak. A korábban létrehozott ClickedItem változóhoz tartozó elemet kitörli.

Egy eseménykezelővel lekezeljük, hogy vizuálisan is eltűnjön a kosárból a termék.

```
//Rendelés gomb
let rendelesGomb = document.createElement("input");
rendelesGomb.type = "button";
rendelesGomb.value = "Rendelés";
rendelesGomb.setAttribute("id", "rendel");
rendelesGomb.style.paddingBottom = "10px";
rendelesGomb.style.margin = "10px";
rendelesGomb.classList.add("searchbutton");
rendelesGomb.setAttribute("onclick", "Megrendeles()");

function Megrendeles() {
    document.getElementById("rendel");
    localStorage.clear();
    location.reload();
    alert("A vásárlás sikeres volt!");
}
```

A “Rendelés” gombot input elemmel oldottuk meg, ami gomb típusú. Hozzáadunk néhány vizuális változtatást, de alapvetően a “Keresés” gomb osztályát rendeljük hozzá, hogy annak a stílusával jelenhessen meg.

Létrehozunk egy Megrendelés nevű függvényt, amely kezeli, hogy a “Rendelés” gombra kattintáskor a LocalStorage-ban tárolt kosáradatok teljesen törlődjenek és töltsse újra az oldalt, hogy ez a változás láthatóvá váljon a felhasználó számára is. Valamint egy felbukkanó értesítésként megjelenik a “A vásárlás sikeres volt!” üzenet.

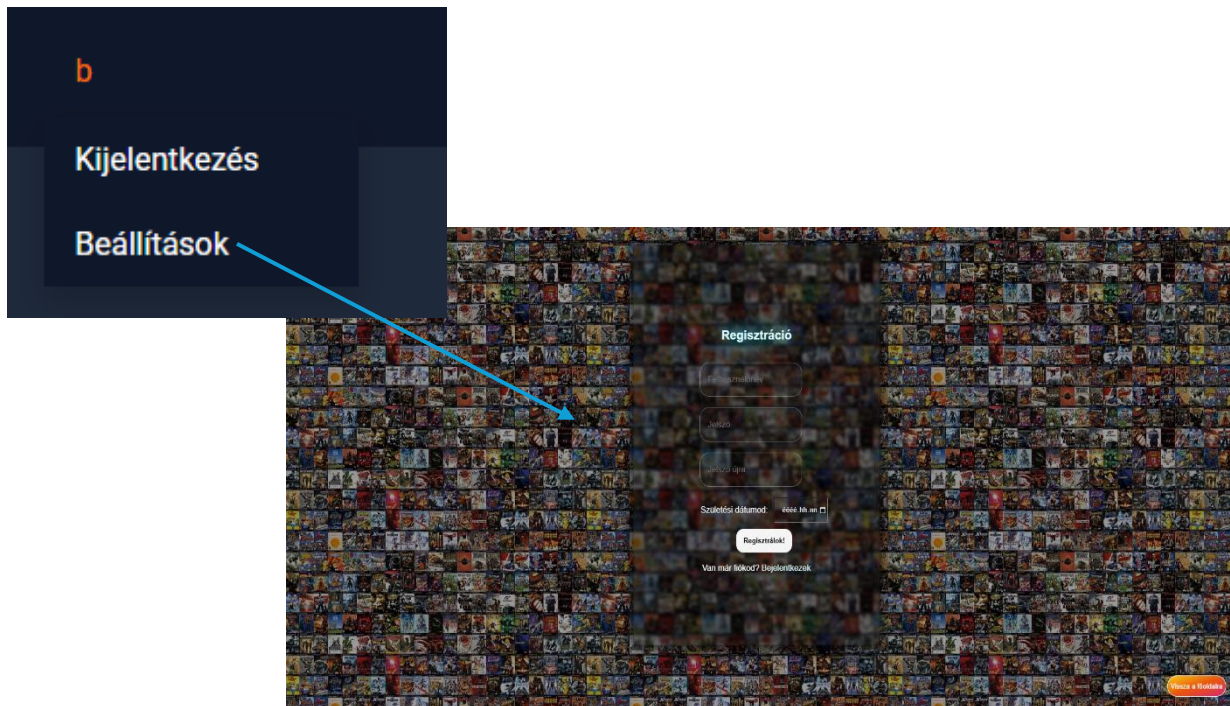
## A projekt tesztelése

### User Acceptance Test (UAT)

A User Acceptance Test (UAT) a szoftver végső tesztelési lépése, melyet a végfelhasználók vagy üzleti képviselők végeznek. Célja annak ellenőrzése, hogy a rendszer kielégíti a felhasználói elvárásokat és üzleti igényeket, valamint megfelelően működik a valós környezetben. A UAT biztosítja, hogy a szoftver készen áll a használatra és megfelel a szükséges követelményeknek.

Ezen teszt során abba a hibába futottunk, hogy a már bejelentkezett felhasználó, ha a *“Beállítások”* gombra nyom, a regisztrációs oldalra dob vissza. Ennek oka a következő volt:

A fejlesztés során ugyan a gombon szereplő szöveget lecseréltük, de nem kezeltük le, hogy funkcióját is cseréljük.



```

document.addEventListener("DOMContentLoaded", function () {

    document.getElementById("felhasz").innerHTML = user;

    if (user != "Felhasználó") {
        log.innerHTML = "Kijelentkezés";
        loggedin = true;
        document.getElementById("reg").innerHTML = "Beállítások";
        document.getElementById("reg").href = "";
        document.getElementById("reg").addEventListener("click", function () {
            alert("Fejlesztés alatt áll!");
        });

        cart.style.display = "block";
        ossztermek.style.visibility = "visible";
    }
});

```

A problémát a fentebbi kódrészlettel oldottuk meg.

Megvizsgáljuk, hogy a menüponton milyen szöveg szerepel. Ha ez egy felhasználó neve, akkor a korábbi *Regisztráció* felirat helyett, a *Beállítások* szerepeljen és ne navigáljon át sehova. A továbbfejlesztési lehetőségekre való tekintettel hozzáadtunk egy eseménykezelőt, amely egy alert-el tér vissza a gombnyomáskor.

## Fejlesztési lehetőségek - Hol fejlődhetne a projekt?

### *Idegennyelvűség*

A weboldal a felhasználóink számára szánt lapjain jelenleg csak a magyar nyelv támogatott. Ugyan egyes böngészők képesek az automatikus fordításra, de ezek sokszor pontatlanok, így frusztráló félreértések léphetnek fel - Szeretnénk a jövőben ennél nemzetközibb weboldal fenntartását biztosítani ahol különböző fordítókkal együtt dolgozva egy olyan funkciót hozhassunk létre, amely a fentebb említett problémára megoldást nyújtana.

### *Idegen pénznemek*

Jelen helyzetben a weblapon az áraink csak Forintban(*HUF*) vannak feltüntetve. A közeljövőben szeretnénk a potenciális külföldi vásárlóinknak eleget téve megváltoztathatóvá tenni hogy az árak különböző pénznemben mennyit érnek. Természetesen a valuta mindig naprakész lenne - API hívásokkal mindennap frissülne.

### *Tranzakciók kezelése*

A vásárlók egyszerűen vásárolhatnak játékokat közvetlenül a platformon.

A platform biztosítja a biztonságos fizetési lehetőségeket (pl. bankkártyás, PayPal, banki átutalás). Minden tranzakciót követően a vásárló és az eladó digitális számlát kap. A tranzakciók története és állapota (fizetés, szállítás) nyomon követhető a felhasználói profilban.

### *Felhasználói profil személyreszabása*

A felhasználói profil almenüjében a *“Beállítások”* jelenleg csak vizuális értékkel bír, még nem használható. A jövőben profilkép beállítása, jelszó vagy e-mail cserélésére is szeretnénk lehetőséget nyújtani a felhasználóinknak.



## *Mobilalkalmazás készítése iOS és Android platformokra*

A szoftverünk csak webes felülettel rendelkezik. A jövőben a könnyebb kezelhetőség érdekében akár mobilalkalmazás készíthető, természetesen figyelembe véve a különböző operációs rendszerek közötti különbségeket.

### *Kiegészítő funkciók*

Rengeteg extra funkció hiányzik az oldalról, amelyek más, konkurens oldalakon fellelhetők, mint például élő chates ügyfélszolgálat, hirdetési lehetőségek az eladók számára, illetve saját hűségprogram bevezetése. Ezekkel a kiegészítő funkciókkal az oldal modernebbé, naprakészebbé válhat, újabb vevőket és eladókat vonzva.

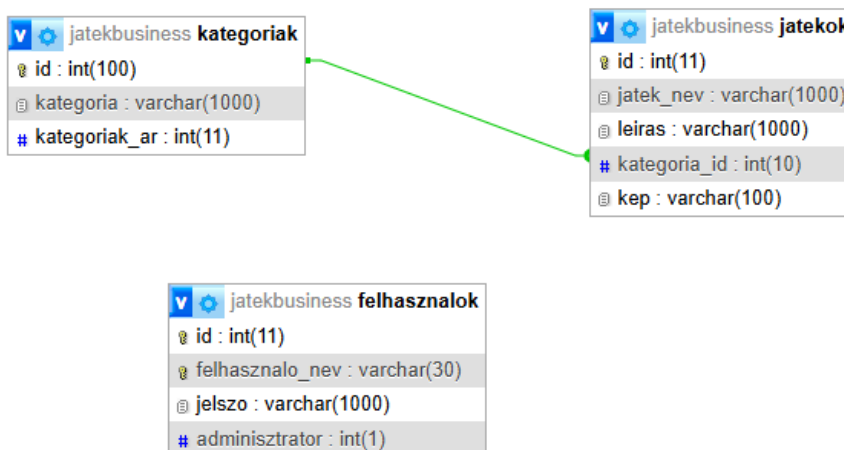
A hűségprogram bevezetése különösen sok előnnyel járhat mind az üzemeltetők, mind a felhasználók számára.

A hűségprogram ösztönözheti a felhasználókat arra, hogy rendszeresen használják a platformot, mivel jutalmakat, kedvezményeket és egyéb előnyöket kínálhat a folyamatos aktivitásért. Minden tranzakció (vásárlás, eladás) után a felhasználók hűségpontokat gyűjthetnek. A pontok később kedvezményekre vagy ajándékokra válthatók be.

A program tagjai extra előnyöket kaphatnak, mint például ingyenes szállítás, extra pontok vagy különleges ajánlatok, ha aktívan részt vesznek a közösségben (pl. értékelésekkel, visszajelzésekkel, ajánlásokkal).

## Adatszerkezetek - Adatabázis terv ismertetése

A weboldalunk egy egyszerű adatbázis szerkezettel rendelkezik. 3 tábla található benne – *felhasznalok*, *jatekok*, illetve *kategoriak*. Az elsődleges kulcsokat egy sárga kulcs ikon jelzi a mező neve mellett. Az alábbi relációs adatmodell szemlélteti ezt:



1. ábra Adatabázis szerkezete

*felhasznalok* tábla

Ebben a táblában minden rekordhoz 3 mező tartozik, itt találhatók az oldalra regisztrálók adatai:

**id** – Egy egyedi azonosító, amelyet az adatbázis a felhasználóhoz automatikusan rendel és növel eggyel minden új rekord felvételekor. A felhasználók azonosítására szolgál a felhasználónevük mellett. Egész szám típusú, elsődleges kulcs, a *jatekok* tábla ennél a mezőnél kapcsolódik a táblához.

jatekbusiness felhasznalok	
id : int(11)	
felhasznalo_nev : varchar(30)	
jelszo : varchar(1000)	
# adminisztrator : int(1)	

2. ábra Felhasználók tábla

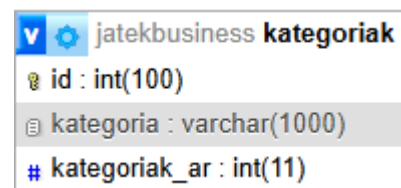
**felhasznalo\_nev** – A felhasználók által megadott (fiktív)név ebbe a mezőbe kerül rögzítésre. Azonosításra szolgál az id mellett, illetve ezen mező rekordja kerül levizsgálásra később a bejelentkezés során. Karakter típusú.

**jelszo** – A felhasználók által megadott jelszó, amely a felhasználónév párossal lehetőséget biztosít a felhasználónak később a bejelentkezésre. Minden jelszó – hasonlóan az id-hez – speciális; egyetlen rekord *jelszo* mezője sem egyezik egy másikével a táblán belül. Karakter típusú.

### *kategoriak* tábla

Ebben a táblában minden rekordhoz 3 mező tartozik, itt találhatók az oldalon megtalálható játékok kategóriái:

**id** – Egy egyedi azonosító, amelyet az adatbázis egy kategóriához automatikusan rendel és növel eggyel minden új rekord felvételekor. Egész szám típusú, elsődleges kulcs, a *jatekok* tábla ennél a mezőnél kapcsolódik a táblához.



<b>jatekbusiness kategoriak</b>
<b>id</b> : int(100)
<b>kat</b> egoria : varchar(1000)
<b>#</b> kategoriak_ar : int(11)

3. ábra Kategóriák tábla

**kat**egoria – A játék kategóriájának névszerinti megnevezése kerül ide. Karakter típusú.

**Kategoriak\_ar** – A játékok kategória szerint kapnak árat az weboldalon – Ezen árak ebben a mezőben kapnak helyet. Egész szám típusú.

### *jatekok* tábla

Ebben a táblában minden rekordhoz 4 mező tartozik, a másik két tábla ezen keresztül érheti el egymást.

**id** – Egy egyedi azonosító, amelyet az adatbázis egy kategóriához automatikusan rendel és növel eggyel minden új rekord felvételekor. Egész szám típusú, elsődleges kulcs.

**felhasznalo\_id** – Egy olyan azonosító, amelyet a tábla a *felhasznalok* tábla *id* mezőjéből kap meg. Minden játékhoz azon felhasználó egyedi azonosítója van csatolva, aki a feltöltéséért felelős. Egész szám típusú.

**jatek\_nev** – A feltöltött játék névszerinti megnevezése. Karaktert típusú.

**kategoria\_id** – Egy olyan azonosító, amelyet a tábla a *kategoriak* tábla *id* mezőjéből kap meg. Minden játék rendelkezik egy kategóriával. A mezőben tárolt adat megegyezhet más rekordok ugyanezen mezőben tárolt adatával.

**kep** – A feltöltött játékhoz tartozó borítójának url-je. Karakter típusú.



jatekbusiness jatekok	
id	int(11)
jatek_nev	varchar(1000)
leiras	varchar(1000)
kategoria_id	int(10)
kep	varchar(100)

4. ábra Játékok tábla

# FELHASZNÁLÓI DOKUMENTÁCIÓ

## Telepítési útmutató

### *Böngésző*

Az alkalmazás böngészőben fut, így szükség lesz a számítógépre feltelepíteni egy böngészőt. A Microsoft Windows az Edge nevű böngészőt kínálja alapértelmezetten, de természetesen másikkal is működik. A böngészők verziószáma a lentebbi táblázatban látható:

<b>Mozilla</b>	<b>Google</b>	<b>Opera</b>	<b>Edge</b>
Mozilla Foundation    Firefox 133 verzió	Google    Chrome 131 verzió	Opera    Software Opera 114 verzió	Microsoft    Edge 131 verzió

### *Indító exe fájl*

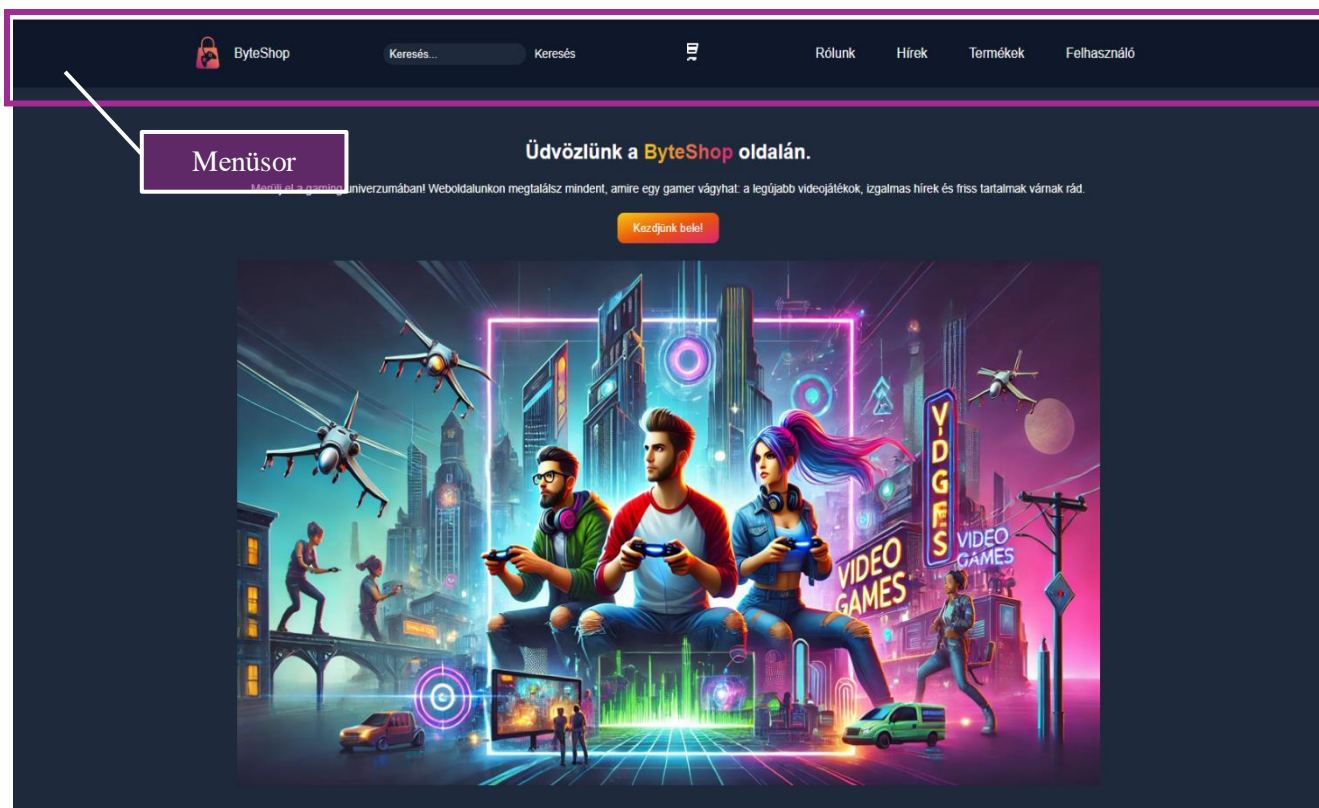
Futtassuk a telepítő exe fájlt és a programunk már készen is áll a futtatásra.

## Felhasználói kézikönyv

Az alkalmazáshoz tartozó szoftverek sikeres telepítése és a telepítő exe fájl futtatása után a program már használható is.

Ez a kézikönyv segítséget nyújthat a programban való kényelmesebb eligazodáshoz és egyes menüpontok megértéséhez. Ha használat közben bármiben elbizonytalanodik, az alábbiakban kérdésére választ kaphat:

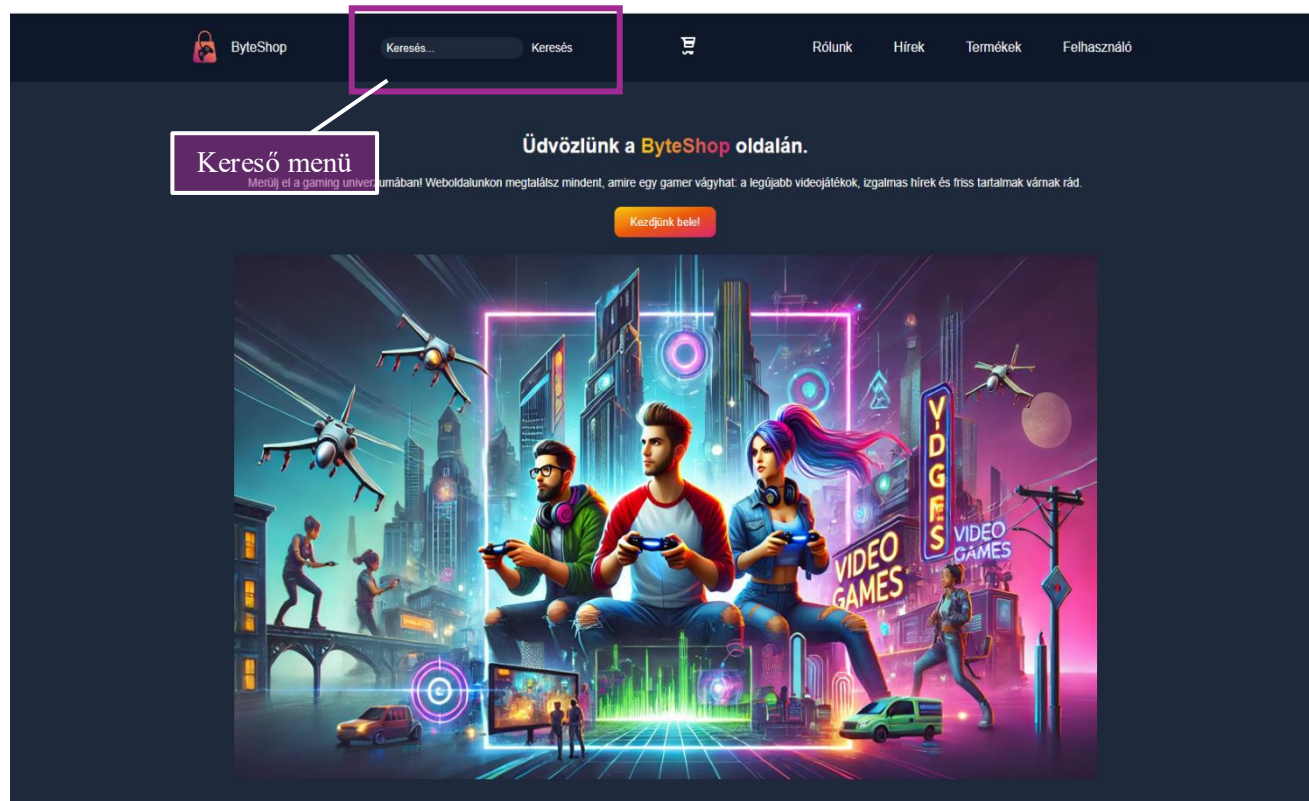
Az oldal betöltésekor a felhasználót a következő oldal fogadja:



5. ábra Főoldal menüsor

A menüsorban egy logo, egy keresőmező, illetve 4 menüponttal tájékozódhatunk az oldalon. A *Kezdjünk bele* nevezetű gomb és a *Termékek* nevezetű menüpont ugyanazon célt szolgálja.

## Kereső menü



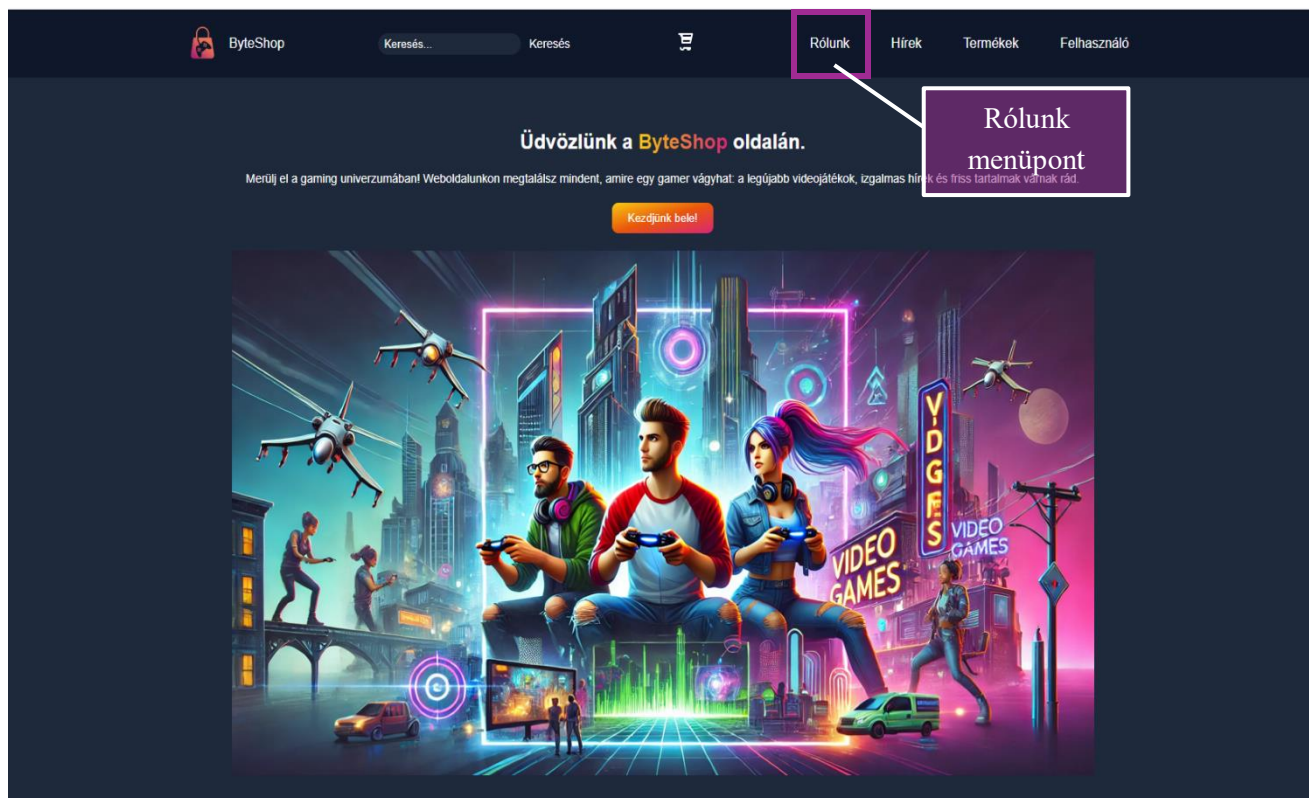
6. ábra Főoldal kereső menü

Ha esetleg felmerül egy játék amit szeretnél, de nem szeretnél böngészni a kínálatunkban, csak gépeld be a játék nevét, nyomj egy enter vagy nyomj rá a mező melletti “Keresés” gombra.

Tekerj lejjebb és minden játékot listáz az oldal számodra ami a keresésednek megfelel. Ha esetleg nem tudnád a pontos nevét, az sem probléma – Az oldal képes kilistázni az olyan játékokat is, amelyekben a megdott szó vagy szórészlet szerepel!



## Rólunk menüpont



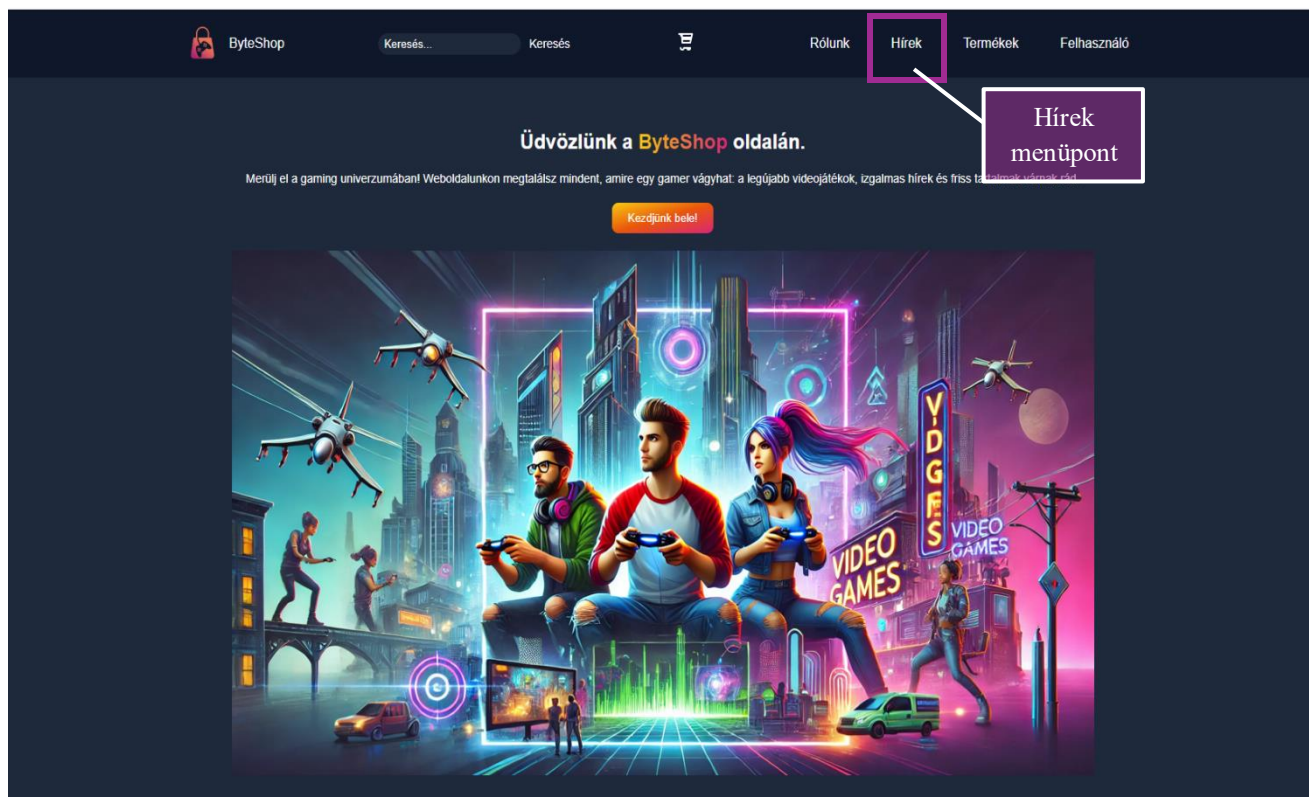
7. ábra Főoldal Rólunk menüpont

Ezen menüpont megnyomásával az oldal azon részére ugrik, ahol további információkat tudhat meg pl. az oldal üzemeltetőiről, az oldalhoz tartozó dokumentumokról és történetéről.



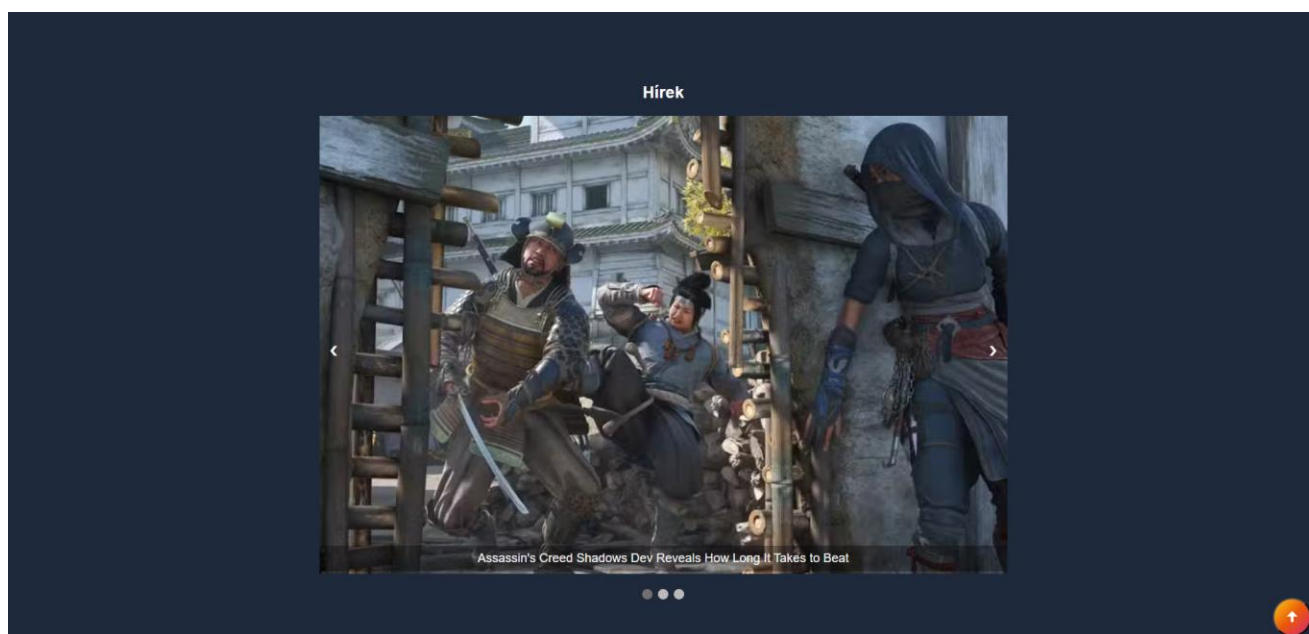


## Hírek menüpont

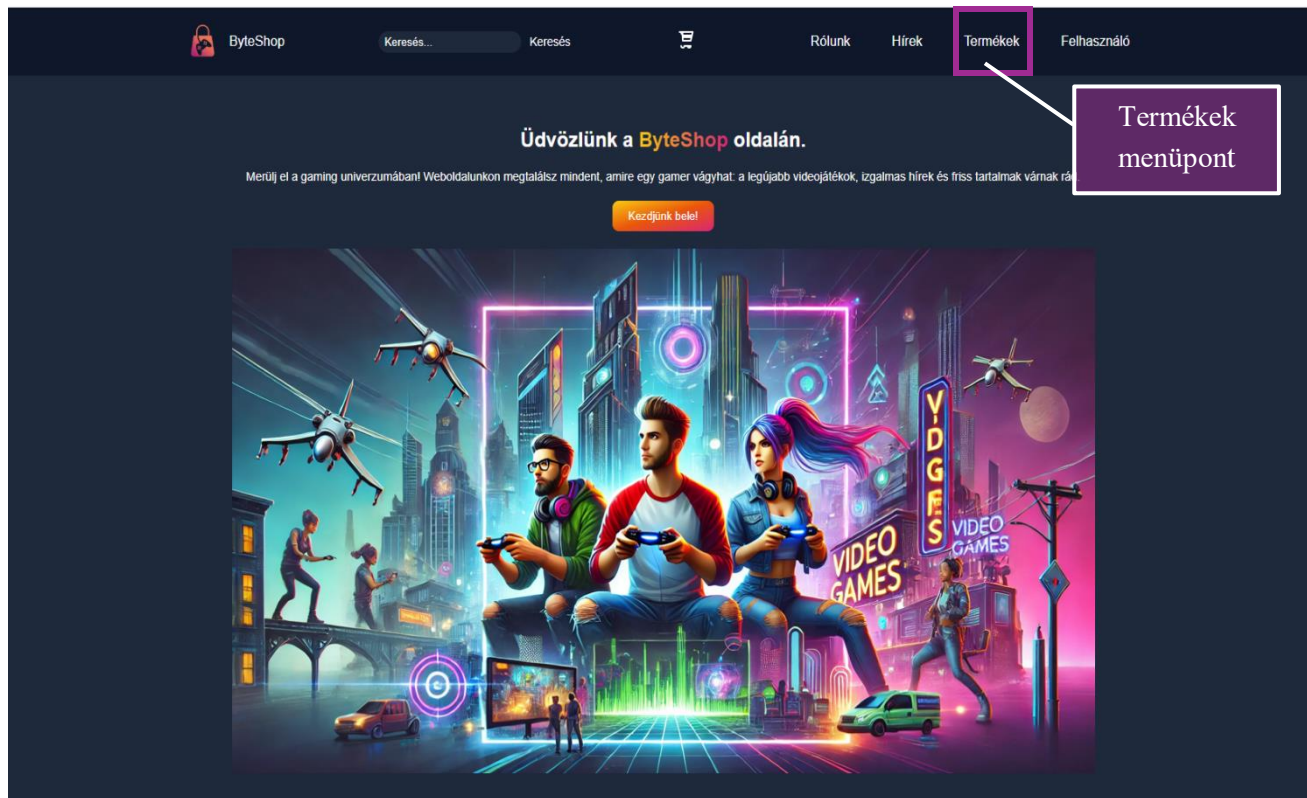


8. ábra Főoldal hírek menüpont

A hírek menüpont megnyomása a hírek szekcióra ugrást eredményezi. Itt a felhasználók megtekinthetik az aktuális híreket a videójátékok világában, egy olyan szekcióban, amely egy üdítő, modern kinézettel színesíti a honlapot.

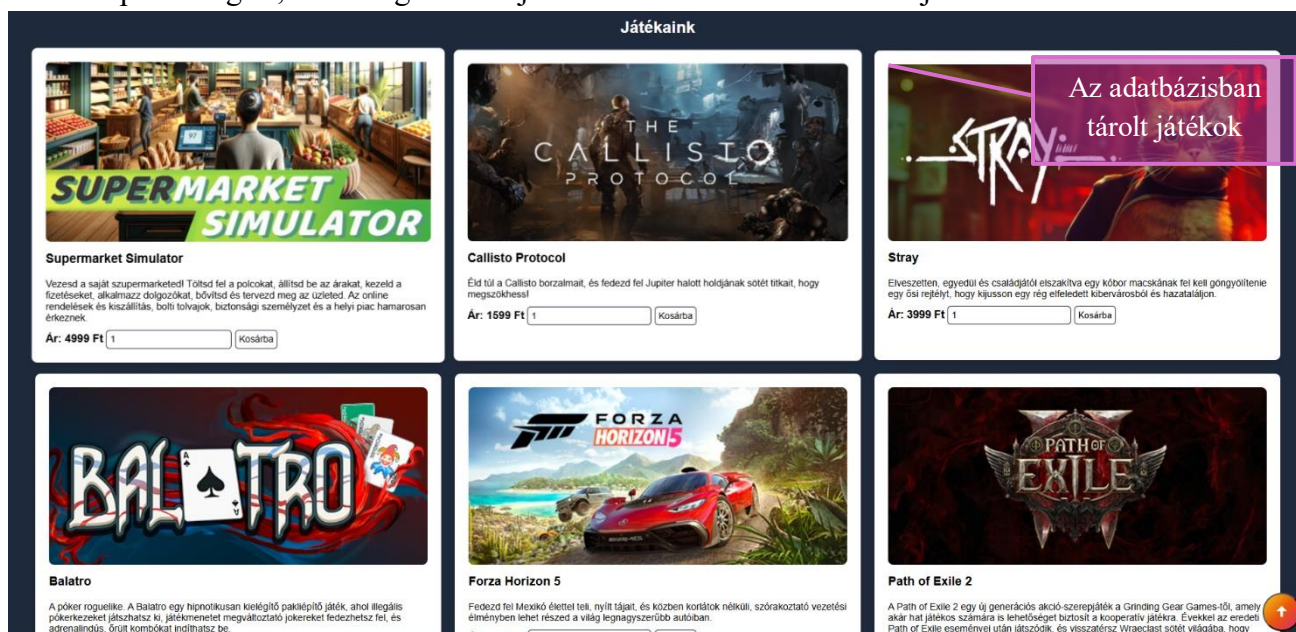


## Termékek menüpont



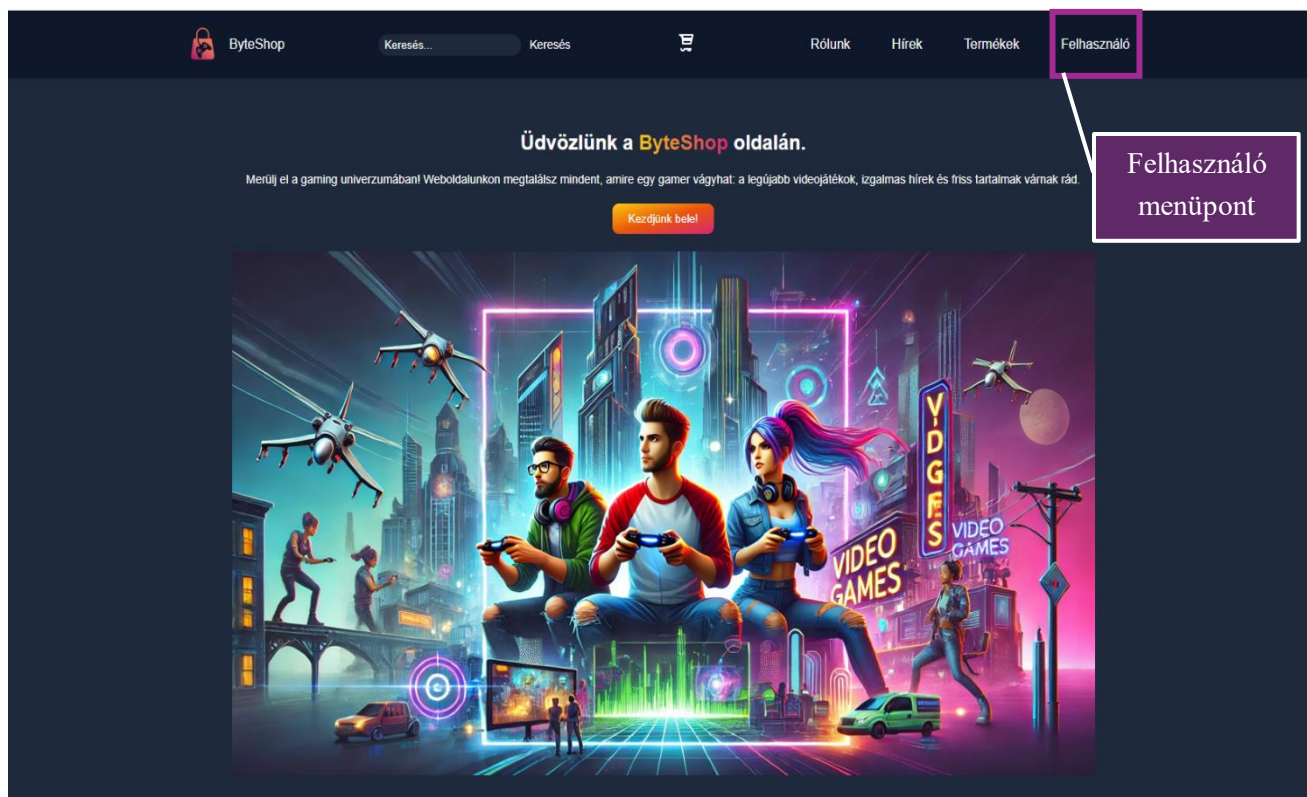
9. ábra Főoldal termékek menüpont

Ez a menüpont segít a felhasználónak a termékeinkhez navigálni. Megnyomásával az oldal azon pontra ugrik, ahol meg vannak jelenítve az adatbázisban tárolt játékok.



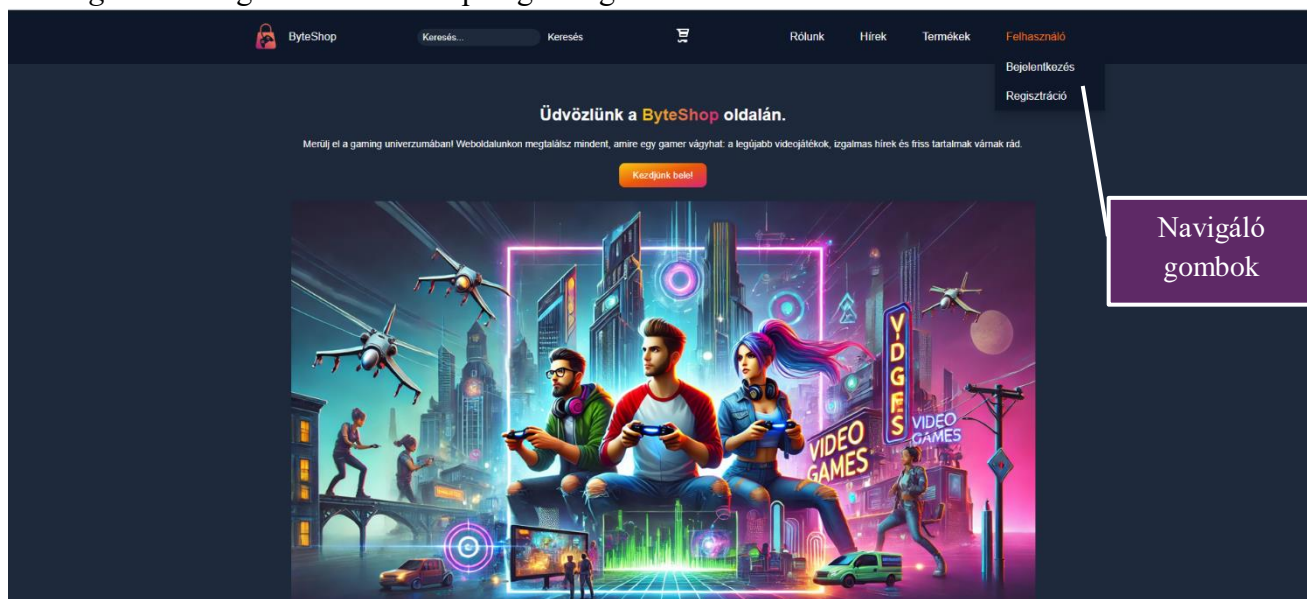


## Felhasználó menüpont



10. ábra Főoldal felhasználó menüpont

Ebben a menüpontban egy legördülő menüben 2 gombot találunk: *Bejelentkezés* és *Regisztráció*. A *Bejelentkezés* gombra kattintva az oldal átnavigál a Belépési oldalra, a *Regisztrációs* gombra kattintva pedig a Regisztrációs oldalra.



11. ábra Főoldal navigáló gombok

## Bejelentkezési oldal

Az oldal bejelentkezés nélkül is meglátogatható, de ahhoz, hogy terméket vásárolhasson, bejelentkezésre lesz szükség.

Ha a felhasználó a főoldal Felhasználó menüpontja alatti Bejelentkezés menüpontra kattint a következő oldal fogadja:

The screenshot shows a login page with a dark, textured background. The word "LOGIN" is centered at the top. Below it are two input fields: "Felhasználónév" (Username) and "Jelszó" (Password). A checkbox labeled "Jelszó megjelenítés" (Remember password) is next to the password field, with a note "Elfelejtette jelszavát?" (Forgot your password?). Below these fields is a "Bejelentkezés" (Login) button. A link "Nincs még fiókom? Regisztrálok" (Don't have an account? Register) is below the button. A "Vissza a főoldalra" (Back to home) button is in the bottom right corner. Four callout boxes provide additional information: 1. "Felhasználónév mező" (Username field) points to the username input field. 2. "Jelszó mező" (Password field) points to the password input field. 3. "Ha nem vagy biztos beírt jelszavadban, ezt a checkboxra kattintva ellenőrizheted" (If you're not sure you entered your password correctly, you can check this by clicking the checkbox) points to the "Jelszó megjelenítés" checkbox. 4. "Ha véletlen léptél az oldalra, ezzel a gombbal visszavigálhatsz a főoldalra" (If you accidentally came to this page, you can go back to the home page with this button) points to the "Vissza a főoldalra" button.

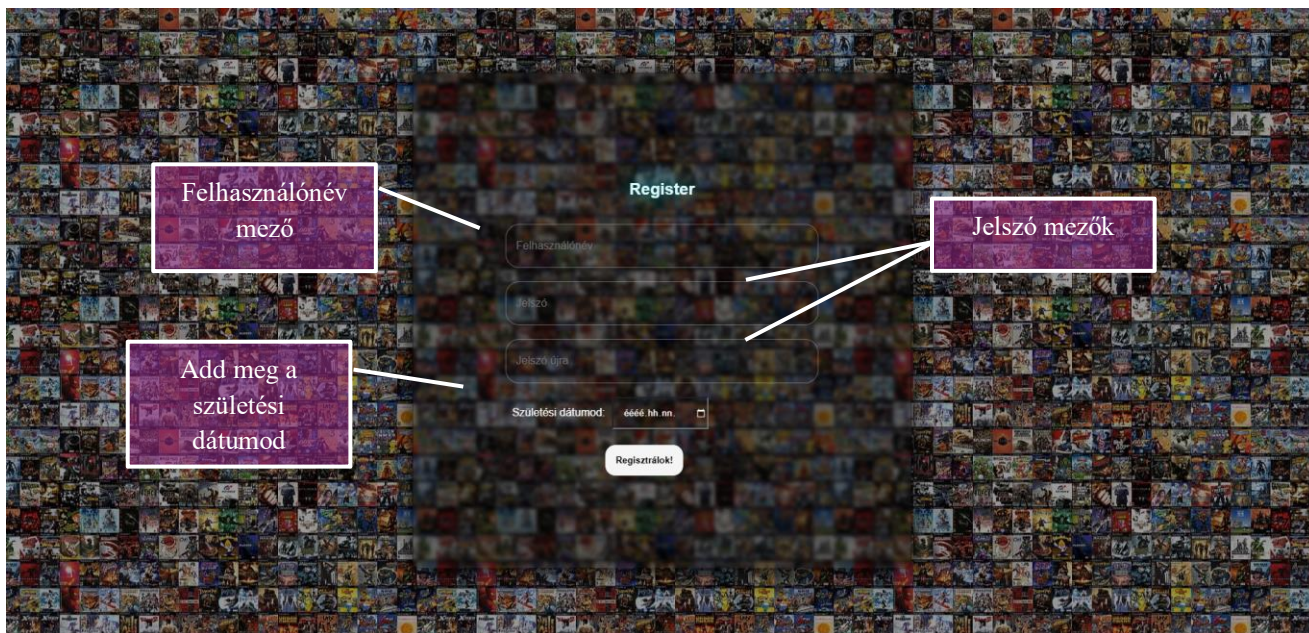
12. ábra Bejelentkezés oldal

A bejelentkezési oldal nagyon egyszerűen működik: Ha már korábban meglátogatta az oldalt és regisztrált, írja be az ott megadott felhasználónév és jelszó párost, majd a Bejelentkezés gombra kattintson. Ezekután az oldal visszavezeti a felhasználót a főoldalra, ezúttal már bejelentkezve.

Sikeres bejelentkezés után a Felhasználók fül helyén a felhasználó felhasználóneve jelenik meg, illetve az almenüpontok a Beállítások és Kijelentkezés almenüpontokra változik meg. Emellett, a menüsor bővül egy kosár menüponttal is.



## Regisztrációs oldal



The image shows a registration form titled "Register" on a background of a dense mosaic of small, colorful images. Three callout boxes with white text and arrows point to specific fields: "Felhasználónév mező" (Username field) points to the first input field, "Jelszó mezők" (Password fields) points to the second and third input fields, and "Add meg a születési dátumod" (Add your birth date) points to the date selection field. The form includes a "Regisztrálok!" button at the bottom.

Felhasználónév mező

Register

Felhasználónév

Jelszó

Jelszó újra

Születési dátumod:

Regisztrálok!

Jelszó mezők

Add meg a születési dátumod

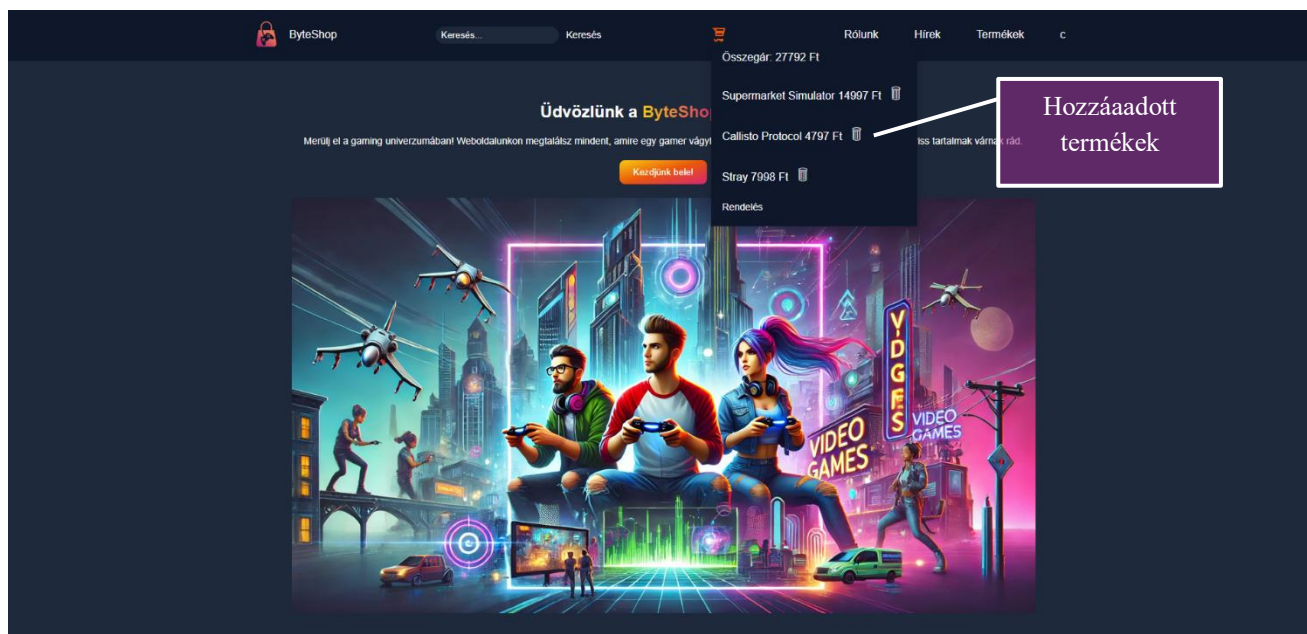
**13. ábra Regisztrációs oldal**

Ha még nem regisztrált, akkor a főoldalon Felhasználók menüpont alatt található “Regisztráció” gombba kattintva, vagy a bejelentkezési oldalon a türkizen világító “Regisztrálok” gombra kattintva, az oldal a regisztrációs oldalra navigálja át.

A regisztrálás a következő módon történik:

A felhasználó mezőben egy fiktív vagy valós nevet meg kell adni, egy jelszó és születési dátum párossal. Sikeres regisztráció után az oldal visszavigál a főoldalra, már bejelentkezve. Később, ha szeretnénk az oldalra bejelentkezni, az itt megadott név és jelszó párossal lehetséges.

## Kosár menüpont



14. ábra Főoldal kosár menüpont

A fentebbi ábrán én már adtam hozzá a kosárhoz néhány terméket szemléltetésképpen.

A legelső sorban a “Végösszeg” található, amely – ahogy a neve is jelzi, egy összegzése a kosárban lévő termékek árainak. Ez az az ár, amelyet ki kell fizetni.

Alatta az általam betett termékek vannak listázva, mellettük az árukkal. A mellettük lévő ár arányosan változik azzal, hogy hány db játékot tettem be.

Az ár mellett lévő kis szemetes ikonra kattintva törölheted a kosárból az adott terméket. Természetesen a törölt termék(ek) ára törlődik a végösszezből is.

## VÉGSZÓ ÉS KÖSZÖNETNYILVÁNÍTÁS

A projekt fejlesztése során számos kihívással szembesültünk, amelyek megoldása nemcsak szakmai fejlődésünket támogatta, hanem csapatmunkánkat is erősítette. Az elkészült weboldal funkciói és dokumentációja egyaránt a felhasználók kényelmét és az adminisztrátorok hatékony munkáját szolgálja.

Bár a jelenlegi verzió teljes funkcionalitással bír, a jövőben tovább kívánjuk fejleszteni, hogy mindig napakész legyen. A fejlesztés során szerzett tapasztalatok és elért eredményeink igazolják, hogy a csapat elhivatottsága és együttműködése kulcsfontosságú volt a projekt sikerében, és bízunk benne, hogy az elkészült rendszer hosszú távon is értékes és fenntartható megoldásként szolgál majd.

Köszönetet nyilvánítunk minden támogató embernek, aki részt vett valamilyen formában a fejlesztésben, legyen az formai visszajelzés, vagy szakmai visszajelzés. Külön köszönetet érdemelnek szakmai oktatóink, akik támogatása és segítsége nélkül ez a minőség és fejlettség nem valósult volna meg.