

SZAKDOLGOZAT

Erdős Roland

2023

Magyar Agrár- és Élettudományi Egyetem

Károly Róbert Campus

**Játékalom tervező és rendező közösségi
platform társasjáték-rajongók számára**

Konzulens

Sike Zoltán

mesteroktató

Készítette

Erdős Roland

gazdaságinformatikus szak

nappali tagozat

2023

Tartalomjegyzék

1.	Bevezetés.....	4
2.	Irodalmi áttekintés.....	5
2.1	Programozás.....	5
2.1.1	Integrált fejlesztői környezet.....	6
2.1.2	Visual Studio Code	6
2.2	World Wide Web	7
2.3	Web Programozás	8
2.3.1	HTTP.....	9
2.3.2	HTML	10
2.3.3	CSS	11
2.3.4	PHP	12
2.3.5	JavaScript.....	14
2.3.6	Bootstrap	15
2.4	Adatbázisok.....	16
2.4.1	SQL.....	17
2.4.2	MySQL	18
2.4.3	phpMyAdmin.....	19
3.	Anyag és módszer	20
3.1	Anyag.....	20
3.2	Módszer.....	21
4.	Tárgyalás	24
4.1	Az adatbázis	24
4.2	A webalkalmazás	25
4.2.1	Keretrendszer	25
4.2.2	Adatbázishoz való csatlakozás.....	27
4.2.3	Kezdőlap	28
4.2.4	Játék kategóriák.....	30
4.2.5	Bejelentkezés és regisztráció	32
4.2.6	Profiloldal	35
4.2.7	Üzenetek és jegyrendszer.....	37
4.2.8	Játékmenetek.....	40
4.2.9	Admin felület	44
5.	Eredmények, következtetések, javaslatok	49
6.	Összefoglalás.....	51
7.	Irodalomjegyzék.....	52
8.	Ábrajegyzék	53
9.	Nyilatkozatok	54
10.	Melléklet	56

1. Bevezetés

Erdős Roland vagyok, a Magyar Agrár és Élettudományi Egyetem nyolcadik féléves gazdaságinformatikus hallgatója. Egyetemi tanulmányaim során elég gyorsan rájöttem, hogy számomra az informatika sokkal érdekesebb és megfoghatóbb, mint a gazdaság világa, emiatt témám választásakor igyekeztem ennek megfelelően dönteni. Választásom így a webalkalmazás készítése témára esett. Szerencsére számomra a webprogramozás egy nagyon kedvelt terület, mivel végtelenül kreatív lehet az ember, ezen kívül amellet, hogy egy hasznos alkalmazást készít a felhasználók részére, kiélheti a tervező énjét is. A 2019.-ben elkövetkezett Covid19-pandémia alatt új hobbik után kutattunk a családommal, amit közösen tudunk csinálni és segít eltölteni az időt, illetve átveszteni a világjárvány által bevezetett karantént. Gyorsan megtaláltuk a közös nevezőt a különböző kártya-, kocka- továbbá társasjátékokban, ami idővel egy fontos pontja lett a családi összejöveteleknek. A családban született siker után számos barátot sikerült bevezetni az egyszerű, illetve komplex játékok világába. A karantén feloldását követően felmerült ezen a területen egykettő jelentős probléma, amit a szakdolgozatom elkészítésével szeretnék orvosolni. Az alapvető probléma, amelyet igyekeztem megoldani az az, hogy rengeteg olyan játék van, ami az összetett szabályai és hosszú játékmenete miatt elijeszti az egyszerűbb játékokra, illetve csupán kikapcsolni vágyó játékosokat. Az emberek életében az egyik legfontosabb dolog a közösség és a szórakozás. Rengeteg különböző kártya-, vagy társasjáték van, amelyek segítségével az emberek szabadidejüket hasznosan és élményekben gazdagon tölthetik. Azonban sok esetben nehéz megszervezni ezeket az alkalmakat, és nem mindig sikerül minden résztvevőt időben és hatékonyan értesíteni a tervezett eseményekről. Ez a probléma inspirálta a szakdolgozatot, amelynek célja egy olyan webalkalmazás megtervezése és implementálása, amely lehetővé teszi az emberek számára, hogy könnyedén megszervezzék és megtalálják az általuk kedvelt játékos tevékenységeket. A webalkalmazás célja, hogy egy átfogó platformot biztosítson, ahol az emberek megtalálhatják a számukra legmegfelelőbb játékokat, és csatlakozhatnak a már meglévő csoportokhoz. Emellett a webalkalmazás lehetőséget biztosít arra is, hogy új játékokat szervezzenek, és hatékonyan kommunikáljanak egymással a tervezett eseményekről. A webalkalmazás fejlesztésével az emberek szabadidejük minőségét javíthatják, és könnyebben megszervezhetik a különböző játékokat, amelyekre vágytak.

2. Irodalmi áttekintés

2.1 Programozás

A programozás számítógépes szoftverek, alkalmazások és egyéb számítógéppel kapcsolatos rendszerek létrehozásának folyamata kódírással. Ez magában foglalja a számítógépes programok tervezését, tesztelését és karbantartását bizonyos problémák megoldására vagy bizonyos feladatok elvégzésére. A programozás kritikus készség a mai világban, mivel szinte minden iparágban használják, beleértve a technológiát, a pénzügyet, az egészségügyet, az oktatást és még sok más. A programozás első lépése a programozási nyelv kiválasztása. Több száz programozási nyelv áll rendelkezésre, mindegyiknek megvannak a maga erősségei és gyengeségei. Néhány népszerű programozási nyelv a Java, a Python, a C++ és a JavaScript. Minden nyelvnek megvan a maga szintaxisa, szabályai és konvenciói, amelyeket be kell tartani a kód helyes megírásához. Szakdolgozatom elkészítéséhez felhasznált programozási nyelvekről, illetve technológiákról később. Bár a programozás bonyolultnak és megfélemlítőnek tűnhet, ez egy olyan kritikus készség, amely forradalmasította, illetve számtalan módon megkönnyítette és hatékonyabbá tette életünket és munkánkat. Például a programozás lehetővé tette számunkra, hogy gyorsan és egyszerűen hozzáférjünk információkhoz, kapcsolatba léphessünk a világ minden tájáról érkező emberekkel, valamint automatizáljuk az unalmas és monoton feladatokat. A programozásnak gazdag története van, amely az 1800-as évek közepéig nyúlik vissza, amikor Charles Babbage megtervezte az Analytical Engine-t, egy mechanikus számítógépet, amely bonyolult számításokat tudott végrehajtani. Az első igazi programozási nyelvet azonban az 1950-es évek elején hozták létre, a Fortran (képletfordítás) kifejlesztésével. A Fortrant úgy tervezték, hogy megkönnyítse a tudósok és mérnökök számára az összetett számítások végrehajtásához szükséges kódírást. Az évek múlásával ez a terület drasztikusan fejlődött, a szakdolgozatomhoz releváns időszak az 1990-es évek, mikor kifejlődött a World Wide Web, ami új programozási nyelvek, például JavaScript és PHP létrehozásához vezetett, amelyeket webfejlesztésre terveztek. A következő években más nyelvek, például a Python, a Ruby és a Java népszerűvé váltak, és a programozás területe tovább fejlődött és terjeszkedett.

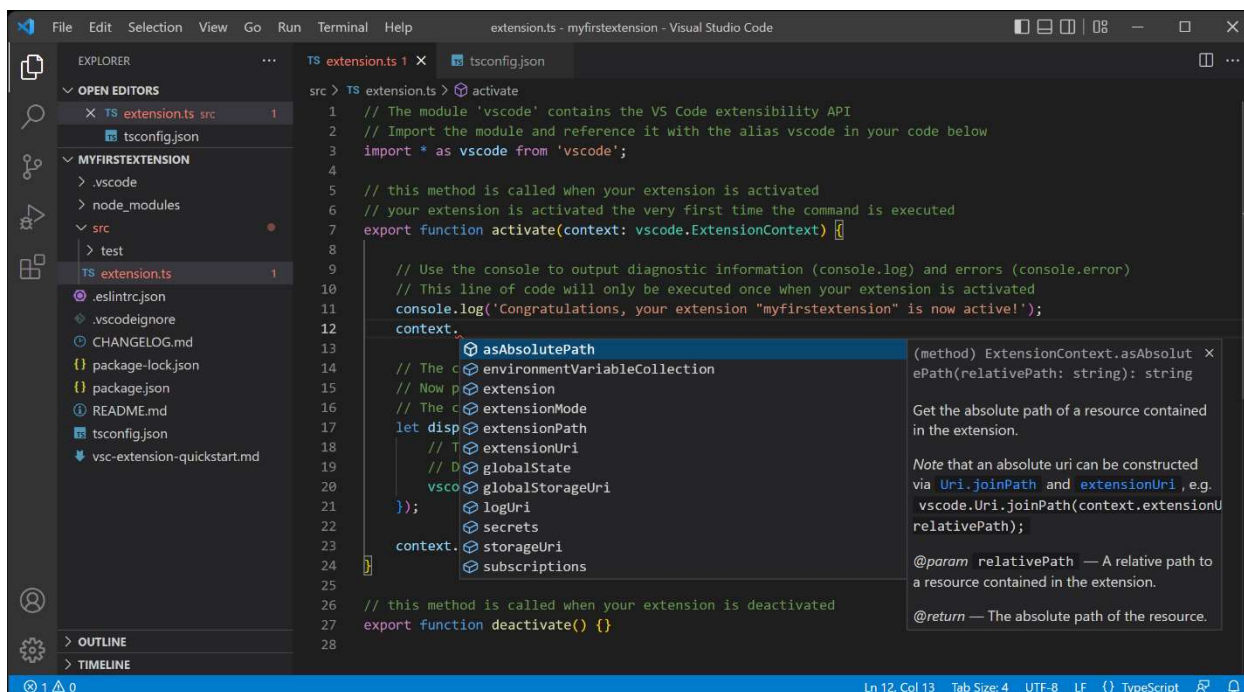
2.1.1 Integrált fejlesztői környezet

Az **Integrated Development Environment (IDE)** egy olyan szoftveralkalmazás, amely átfogó eszközöket és funkciókat biztosít a fejlesztők számára kód írásához, teszteléséhez és hibakereséséhez. Az IDE-ket úgy tervezték, hogy egyszerűsítsék a fejlesztési folyamatot, és központosított környezetet biztosítsanak a szoftverfejlesztés minden aspektusához. Általában tartalmazznak kódszerkesztőt, hibakeresőt, fordítót vagy értelmezőt, és számos egyéb szolgáltatást, például kódkiegészítőt, szintaxiskiemelést és kódrefaktoráló eszközöket. Integrált verziókezelő rendszereket, projektmenedzsment eszközöket és tesztelési keretrendszereket is biztosíthatnak. Használatának egyik legfontosabb előnye, hogy jelentős mennyiségű időt és erőfeszítést takaríthat meg a fejlesztőknek, mivel egyetlen környezetben biztosít minden szükséges eszközt. Az IDE-k emellett valós idejű visszajelzést és elemzést, valamint fejlett hibakeresési funkciókat biztosítanak a kód hibáinak és hibáinak minimalizálásában. Számos különböző integrált fejlesztői környezet érhető el a különböző programozási nyelvekhez és platformokhoz. Néhány népszerű példa a Visual Studio, IntelliJ IDEA, Eclipse vagy az Xcode.

2.1.2 Visual Studio Code

A **Visual Studio Code**, közismert nevén **VSCode**, a Microsoft által fejlesztett ingyenes és nyílt forráskódú kódszerkesztő. Ez az egyik legnépszerűbb kódszerkesztő a világon, és fejlesztők milliói használják a programozási nyelvek és platformok széles skáláján. A VSCode funkciók és eszközök széles skáláját kínálja, amelyek megkönnyítik a fejlesztők számára a kód írását, tesztelését és hibakeresését. Ezek a funkciók többek között támogatják a szintaxis kiemelését, a kódkiegészítést, a hibakeresést és a Git integrációt. Ezenkívül nagymértékben testreszabható, lehetővé téve a fejlesztők számára, hogy bővítményeket és beépülő modulokat telepítsenek a szerkesztő funkcionalitásának javítása érdekében. Az egyik legfontosabb előnye a sebesség és a teljesítmény. Úgy tervezték, hogy könnyű és gyors legyen, még akkor is, ha nagy és összetett kódbázisokkal dolgozik. Rendkívül intuitív felhasználói felületet is biztosít, amely megkönnyíti a fejlesztők számára a navigálást és a szerkesztő testreszabását. Emellett nagymértékben bővíthető, a fejlesztők nagy és aktív közössége alkotja és osztja meg a bővítményeket és beépülő modulokat. Ezek a bővítmények használhatók további programozási nyelvek támogatására, integrálhatók felhőszolgáltatásokkal, mint például az Azure és AWS, és még a rutin fejlesztési feladatokat is

automatizálhatják. Összességében a VSCode egy rendkívül sokoldalú és hatékony kódszerkesztő, amely a fejlesztők számára biztosítja a kiváló minőségű és hatékony kód írásához szükséges szolgáltatásokat és eszközöket. Népszerűsége, valamint kiterjedt bővítmények és beépülő modulok ökoszisztémája a modern szoftverfejlesztés elengedhetetlen eszközévé teszik.

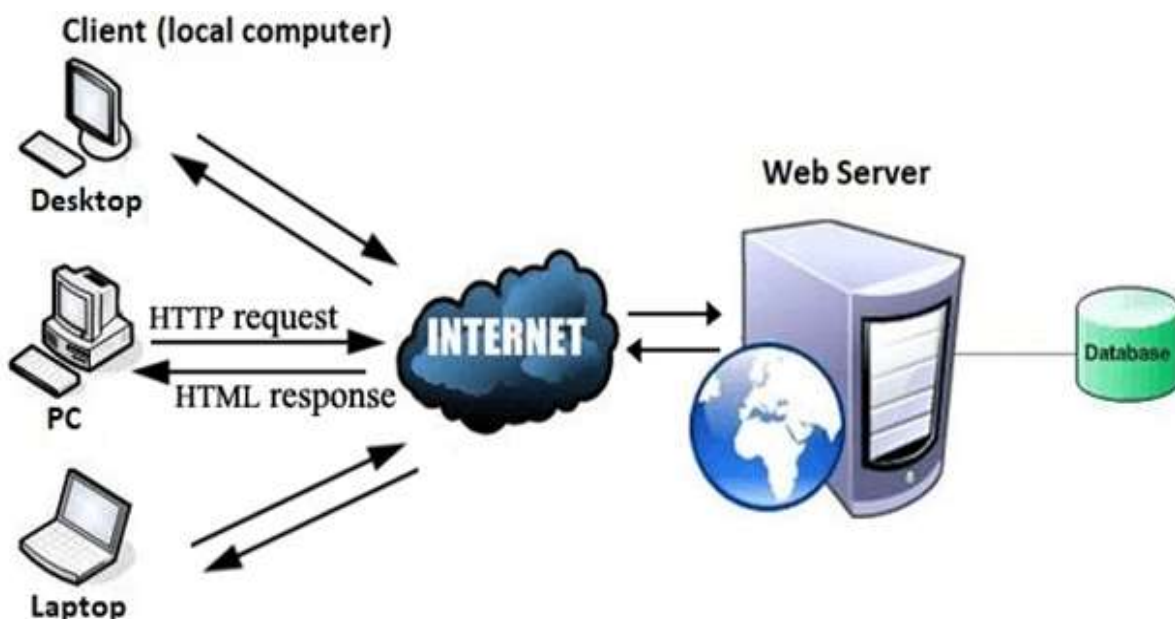


1. ábra: A Visual Studio Code felülete (VisualStudio, 2023)

2.2 World Wide Web

A **World Wide Web (WWW)** vagy egyszerűen a **Web**) dokumentumok és egyéb források összekapcsolt, hiperhivatkozásokkal és **URL**-ekkel (**Uniform Resource Locator**) összekötött rendszere, amely az interneten keresztül, webböngészővel érhető el. A webet Tim Berners-Lee találta fel az 1990-as évek elején, amikor a CERN-ben, az Európai Fizikai Kutatóközpontban dolgozott, hogy a tudósok könnyebben megosszák egymással az információkat és együttműködhessenek egymással. A Web egy globális hálózat, amely lehetővé teszi az emberek számára, hogy soha nem látott mértékben megosszák egymással az információkat és kommunikáljanak egymással. Forradalmasította azt, ahogyan élünk, dolgozunk, és hogyan kommunikálunk a minket körülvevő világgal. A web lehetővé tette az emberek számára, hogy a világ bármely pontjáról, a nap bármely szakaszában hozzáférjenek hatalmas mennyiségű információhoz. Lehetővé tette az e-

kereskedelem, a közösségi média, az online oktatás és egyebek térnyerését, és átalakította kommunikációnkat, tanulásunkat és üzletvitelünket. A web az **Internetre** épül, amely összekapcsolt számítógépek és szerverek globális hálózata. A web protokollkészletet használ, például a **HTTP-t** (**H**ypertext **T**ransfer **P**rotocol) és a **HTML-t** (**H**ypertext **M**arkup **L**anguage), hogy lehetővé tegye a webböngészők számára a weboldalak elérését és megjelenítését. A weboldalak HTML-ben íródnak, amely egy jelölőnyelv, amely lehetővé teszi a tervezők és a fejlesztők számára szövegek, képek és egyéb multimédiás tartalmak létrehozását és formázását.



2. ábra: A World Wide Web architektúrája (JavaPoint)

2.3 Web Programozás

A webfejlesztés weboldalak és webes alkalmazások létrehozásának folyamata, amely magában foglalja a weboldal működéséhez szükséges infrastruktúra tervezését, felépítését és karbantartását. A webfejlesztés története az 1990-es évek elejére nyúlik vissza, amikor a világhálót Tim Berners-Lee brit informatikus találta fel. Berners-Lee kifejlesztette az első webböngészőt, és megírta az első weboldalt, amely bevezette a hipertext fogalmát, vagyis azt a képességet, hogy egy dokumentumot a másikhoz kapcsoljanak. Ez a forradalmi ötlet fektette le a ma ismert internet alapjait. Kezdetben a webfejlesztés statikus HTML-oldalakra korlátozódott, amelyeket szövegszerkesztővel hoztak létre. Ezek az oldalak egyszerűek voltak, és nem volt interaktivitásuk, de lehetőséget biztosítottak a vállalkozások és magánszemélyek számára, hogy megjelenjenek az

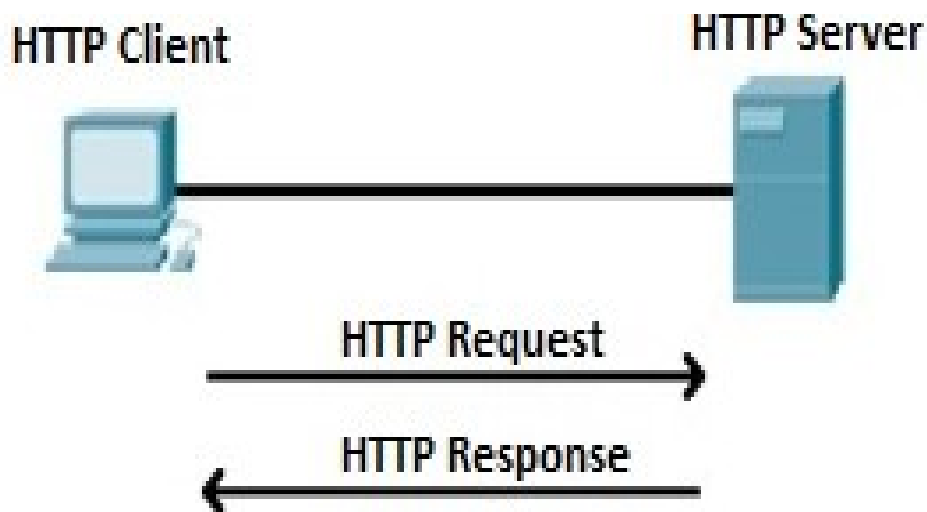
interneten. Ahogy a web fejlődött, úgy fejlődött a webfejlesztés is, és új technológiákat fejlesztettek ki, amelyek dinamikusabb és interaktívabb weboldalakat tettek lehetővé. A webfejlesztés egyik legjelentősebb fejlesztése a szerveroldali szkriptnyelvek, például a **PHP** (**H**ypertext **P**re-**p**rocessor) bevezetése volt, amely lehetővé tette a dinamikus tartalomgenerálást. Ezekkel a nyelvekkel a webfejlesztők olyan oldalakat hozhatnak létre, amelyek a felhasználói beviteltől vagy az adatbázis-lekérdezésektől függően különböző tartalmat jelenítenek meg. Ez új lehetőségeket nyitott meg az e-kereskedelmi webhelyek és webes alkalmazások előtt. Egy másik jelentős fejlemény a **JavaScript** térnyerése volt, egy kliensoldali szkriptnyelv, amely dinamikus és interaktív felhasználói felületeket tesz lehetővé. A JavaScript lehetővé tette olyan webalkalmazások létrehozását, amelyek oldalfrissítés nélkül képesek összetett feladatokat végrehajtani. Ez jelentős előrelépést jelentett a hagyományos webes alkalmazásokhoz képest, amelyekhez minden alkalommal új oldalt kellett betölteni, amikor a felhasználó valamilyen műveletet hajt végre. A 2000-es évek elején jelent meg a Web 2.0 koncepciója, amely a felhasználók által generált tartalmat, a közösségi hálózatokat és az együttműködést helyezte előtérbe. Ez olyan webes technológiák kifejlesztéséhez vezetett, mint az **AJAX** (aszinkron JavaScript és XML), amelyek valós idejű frissítéseket és gördülékenyebb felhasználói élményt tesznek lehetővé. Az AJAX lehetővé tette a webes alkalmazások számára, hogy az oldal frissítése nélkül kérjenek le adatokat a szerverről, ami olyan modern webalkalmazások létrehozásához vezetett, mint a Gmail és a Google Térkép.

2.3.1 HTTP

A **HTTP** (**H**ypertext **T**ransfer **P**rotocol) a World Wide Web által használt protokoll a webszerverek és a kliensek (webböngészők) közötti adatátvitelre. A HTTP egy alkalmazási rétegbeli protokoll, amely az **Internet Protokollt (IP)** használja az alapul szolgáló szállítási réteg protokolljaként. Így működik a HTTP:

1. A kliens (böngésző) kérést küld a webszervernek a HTTP protokoll használatával. A kérélem tartalmazza az ügyfél által lekérni kívánt weboldal URL-jét
2. A webszerver megkapja a kérést és feldolgozza azt. A szerver lekéri a kért weboldalt, és HTTP-válaszként visszaküldi az ügyfélnek.
3. Az ügyfél megkapja a HTTP-választ, és megjeleníti a weboldalt a felhasználó képernyőjén.

A HTTP kérések és válaszok fejlécekből és törzsekből állnak. A fejlécek információkat tartalmaznak a kérelemről vagy válaszról, például a kérés típusáról, a törzs hosszáról és a tartalom típusáról. A törzs tartalmazza a ténylegesen továbbított adatokat, például egy weboldal HTML-kódját. A HTTP számos kérésí módszert támogat, köztük a GET, POST, PUT, DELETE, HEAD és OPTIONS. A leggyakrabban használt kérésí módszer a GET, amely adatok lekérésére szolgál egy webszerverről. A POST arra szolgál, hogy adatokat küldjön egy webszervernek, például űrlap elküldésekor vagy fájl feltöltésekor. A HTTP támogatja a **cookie**-k használatát is, amelyek kisméretű adatok, amelyeket a webszerver tárol az ügyfél számítógépén. A cookie-k a felhasználói adatok és preferenciák, például bejelentkezési adatok és a bevásárlókosár tartalma tárolására szolgálnak



3. ábra: Hypertext Transfer Protocol működése (JavaPoint)

2.3.2 HTML

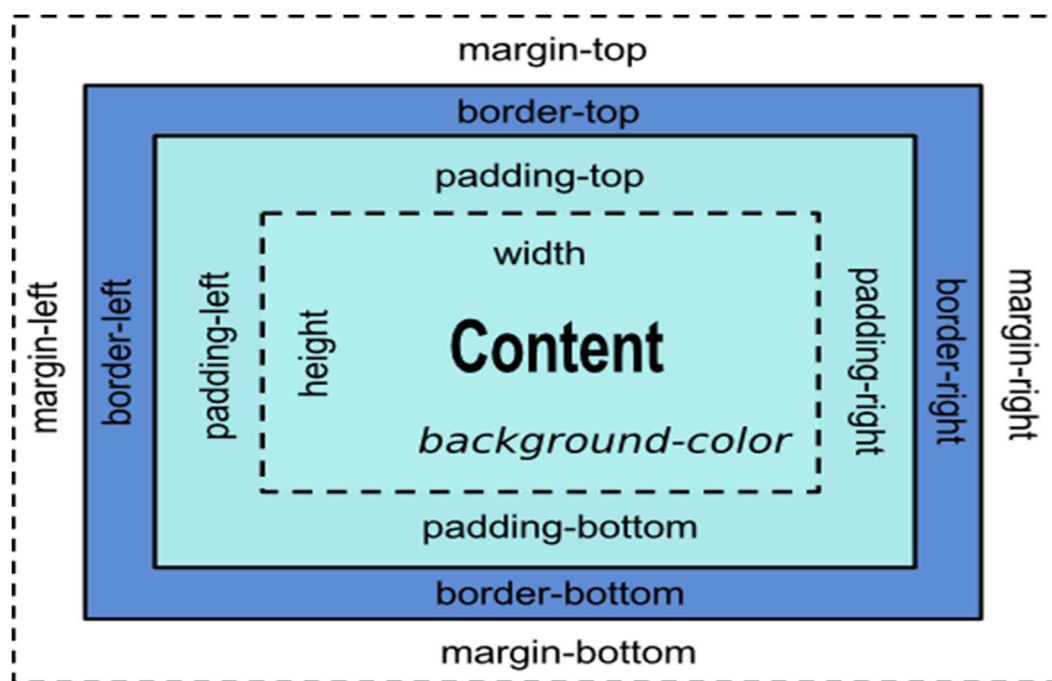
A **HTML (Hypertext Markup Language)** a szabványos jelölőnyelv, amelyet weboldalak és webes alkalmazások létrehozására használnak. Lehetővé teszi a fejlesztők számára, hogy meghatározzák a weboldal szerkezetét és tartalmát, beleértve a szöveget, képeket, videókat és egyéb multimédiás tartalmakat. A HTML úgy működik, hogy jelölőcímkéket használ a weboldal elemeinek meghatározására. A címkék szögletes zárójelben vannak, a nyitó címke tartalmazza az elem nevét és az elemet módosító attribútumokat, a záró címke pedig egy perjelet tartalmaz az elem neve előtt. Például egy bekezdésem nyitó címkéje <p>, a záró címke pedig </p> lenne. A címkéken belül a fejlesztők szöveget, képeket, videókat és más típusú tartalmat helyezhetnek el. Címkék

széles skáláját kínálja különböző típusú tartalmak meghatározásához, beleértve a címsorokat, bekezdéseket, listákat, hivatkozásokat, képeket és multimédiás elemeket, például hangot és videót. Lehetőséget biztosít a fejlesztőknek arra is, hogy az `<a>` címke használatával linkeket hozzanak létre weboldalak között. A hivatkozások használhatók ugyanazon a webhelyen belüli oldalak összekapcsolására vagy külső webhelyekre mutató hivatkozásokra. A weboldal szerkezetének és tartalmának meghatározása mellett a HTML segítségével a fejlesztők képesek arra is, hogy a `<meta>` címke használatával metaadatokat adjanak a weboldalhoz. A metaadatok tartalmazhatnak olyan információkat, mint a weboldal címe, szerzője, kulcsszavak és leírások. A HTML-dokumentum létrehozása után megjeleníthető a webböngészőben a HTTP használatával, és lekérheti a dokumentumot egy webszervertől. A webszerver elküldi a HTML-dokumentumot a kliens webböngészőjének, amely ezután rendereli a dokumentumot, és megjeleníti a képernyőn. Összefoglalva, a HTML egy jelölőnyelv, amelyet weboldalak és webes alkalmazások létrehozására használnak. Úgy működik, hogy jelölőcímkeket használ a weboldal szerkezetének és tartalmának meghatározására, és címkék széles skáláját kínálja a különböző típusú tartalmak meghatározásához. A HTML-dokumentum létrehozása után megjeleníthető egy webböngészőben a HTTP-protokoll használatával, és lekérheti a dokumentumot egy webszervertől.

2.3.3 CSS

A CSS (Cascading Style Sheets) egy stíluslapnyelv, amelyet a HTML dokumentumok megjelenítésének és stílusának leírására használnak. Lehetővé teszi a fejlesztők számára, hogy elkülönítsék a weboldal megjelenítését annak szerkezetétől és tartalmától, így nagyobb rugalmasságot és ellenőrzést tesz lehetővé a weboldal vizuális kialakítása felett. A CSS úgy működik, hogy szelektorok segítségével céloz meg adott HTML-elemeket, és stílusokat alkalmaz rájuk. A stílusok alkalmazhatók a weboldal egyes elemeire, elemcsoportjaira vagy teljes szakaszaira, és szabályozhatják az olyan tulajdonságokat, mint a betűméret, szín, háttérszín, elrendezés és elhelyezés. Számos elrendezési és pozicionáló eszközt is kínál, amelyek lehetővé teszik a fejlesztők számára, hogy összetett oldalelrendezéseket hozzanak létre, beleértve az oszlopokat, rácsokat és a különböző képernyőméretekhez és eszközökhöz alkalmazkodó reszponzív elrendezéseket. A CSS egyik legfontosabb előnye, hogy képes konzisztens, újrafelhasználható stílusokat létrehozni a webhelyen. A külső stíluslapok használatával a fejlesztők meghatározhatnak egy olyan stíluskészletet, amely több weboldalra is alkalmazható, így biztosítva, hogy a webhely kialakítása

egységes és összefüggő marad. Több speciális funkciót is kínál, például animációkat és átmeneteket, amelyek segítségével dinamikus és interaktív webes élményeket lehet létrehozni. Ezek a funkciók segíthetnek javítani a webhely felhasználói élményén, és vonzóbbá és vizuálisan vonzóbbá tenni. Az alapvető funkciókon kívül a CSS-t harmadik féltől származó könyvtárak és keretrendszerek széles skálája is támogatja, amelyek segíthetnek a fejlesztési folyamat egyszerűsítésében, és előre elkészített stílusokat és összetevőket biztosítanak a gyakori webtervezési feladatokhoz. Összefoglalva, a CSS egy stíluslapnyelv, amelyet a HTML dokumentumok megjelenítésének és stílusának leírására használnak. Lehetővé teszi a fejlesztők számára, hogy elkülönítsék a weboldal megjelenítését annak szerkezetétől és tartalmától, így nagyobb rugalmasságot és ellenőrzést tesz lehetővé a weboldal vizuális kialakítása felett. Elrendezési és pozicionálási eszközeivel, egységes stílusával és fejlett funkcióival a CSS a modern webtervezés és -fejlesztés elengedhetetlen eszköze.

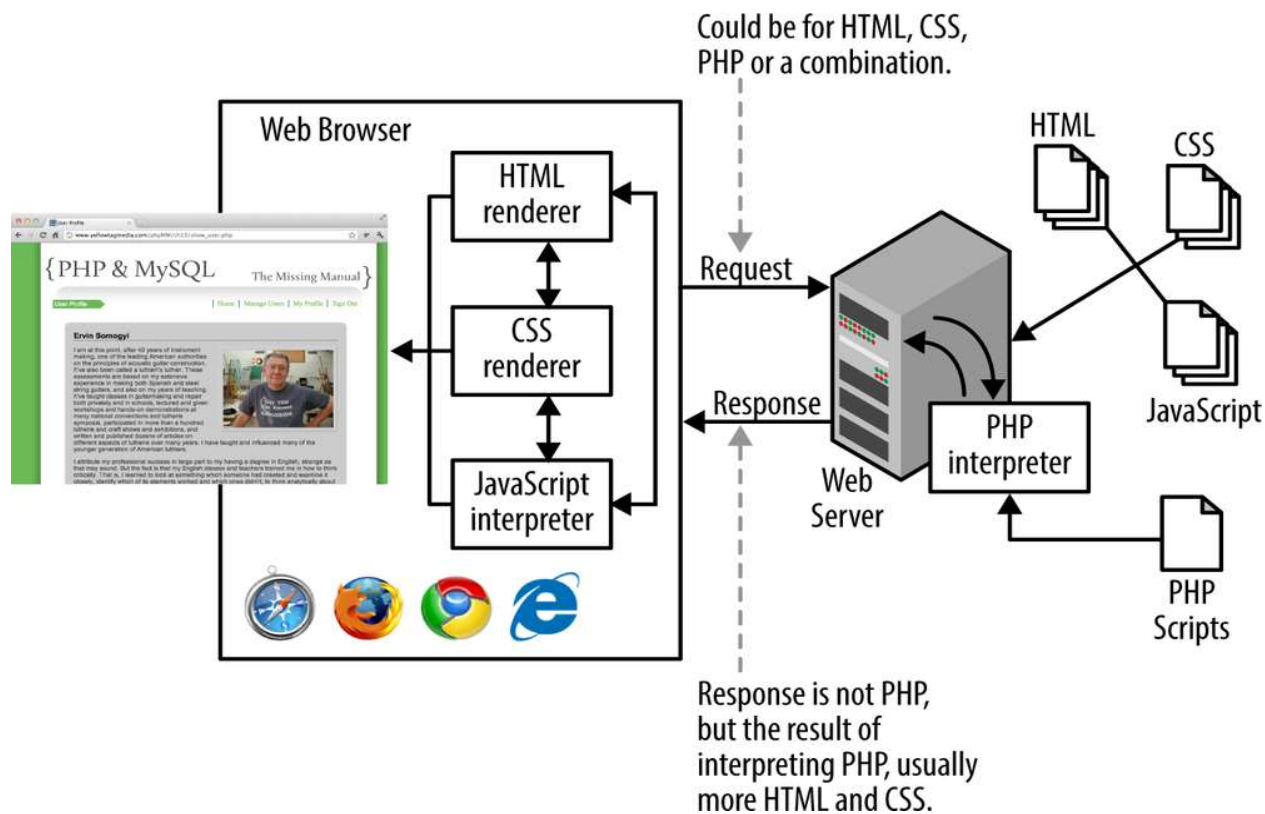


4. ábra: CSS doboz modell (OpenBookProject)

2.3.4 PHP

A **PHP** (**H**ypertext **P**reprocessor) egy szerveroldali szkriptnyelv, amelyet széles körben használnak dinamikus weboldalak és webes alkalmazások létrehozására. Ez egy nyílt forráskódú nyelv, ami azt jelenti, hogy ingyenesen használható, és a fejlesztők módosíthatják igényeiknek megfele-

lően. Úgy működik, hogy szkripteket futtat egy webserveren, hogy HTML tartalmat hozzon létre, amely megjeleníthető a webböngészőben. Ez azt jelenti, hogy a PHP szkriptek a szerver oldalon futnak le, nem pedig a kliens oldalon, ami lehetővé teszi a weboldalak tartalmának és funkcionalitásának nagyobb ellenőrzését. A PHP-szkriptek speciális címkékkel, úgynevezett PHP-címkékkel vannak beágyazva a HTML-oldalakba. Ezeket a címkéket arra használják, hogy beárják a PHP kódot, amely dinamikusan tartalom generálására, adatok manipulálására, valamint adatbázisokkal és egyéb szerveroldali erőforrásokkal való interakcióra használható. A PHP egyik legfontosabb jellemzője az adatbázisokkal való interakció képessége. Számos funkciót biztosít az adatbázisokhoz való kapcsolódáshoz, az adatok lekérdezéséhez és a rekordok frissítéséhez. Ez lehetővé teszi a fejlesztők számára, hogy dinamikus webalkalmazásokat hozzanak létre, amelyek képesek tárolni és lekérni az adatokat, például felhasználói profilokat, terméklistákat és más típusú tartalmakat. A PHP másik fontos tulajdonsága az objektum-orientált programozás támogatása. Az objektumorientált programozás egy olyan programozási paradigma, amely lehetővé teszi a fejlesztők számára, hogy összetett, moduláris kódstruktúrákat hozzanak létre osztályok, objektumok és metódusok meghatározásával. Ez a PHP kódot újrafelhasználhatóbbá, karbantarthatóbbá és méretezhetőbbé teheti. Emellett számos könyvtárat és keretrendszert támogat, amelyek segítségével a fejlesztők gyorsabban és hatékonyabban készíthetnek webalkalmazásokat. Néhány népszerű PHP-keretrendszer közé tartozik a Laravel, a Symfony és a CodeIgniter, amelyek előre beépített összetevőket biztosítanak az általános webfejlesztési feladatok kezeléséhez, mint például a felhasználói hitelesítés, az útválasztás és a bemenet ellenőrzése. Összefoglalva, a PHP egy hatékony és rugalmas szerveroldali szkriptnyelv, amely lehetővé teszi a fejlesztők számára, hogy dinamikus weboldalakot és webalkalmazásokat hozzanak létre. Úgy működik, hogy szkripteket futtat egy webserveren, hogy HTML tartalmat generáljon, amely megjeleníthető a webböngészőben. A PHP-szkriptek speciális címkék segítségével vannak beágyazva a HTML-oldalakba, és felhasználhatók adatok manipulálására, adatbázisokkal való interakcióra, valamint összetett, moduláris kódstruktúrák létrehozására objektum-orientált programozással.



5. ábra: PHP működése (McLaughlin, 2012)

2.3.5 JavaScript

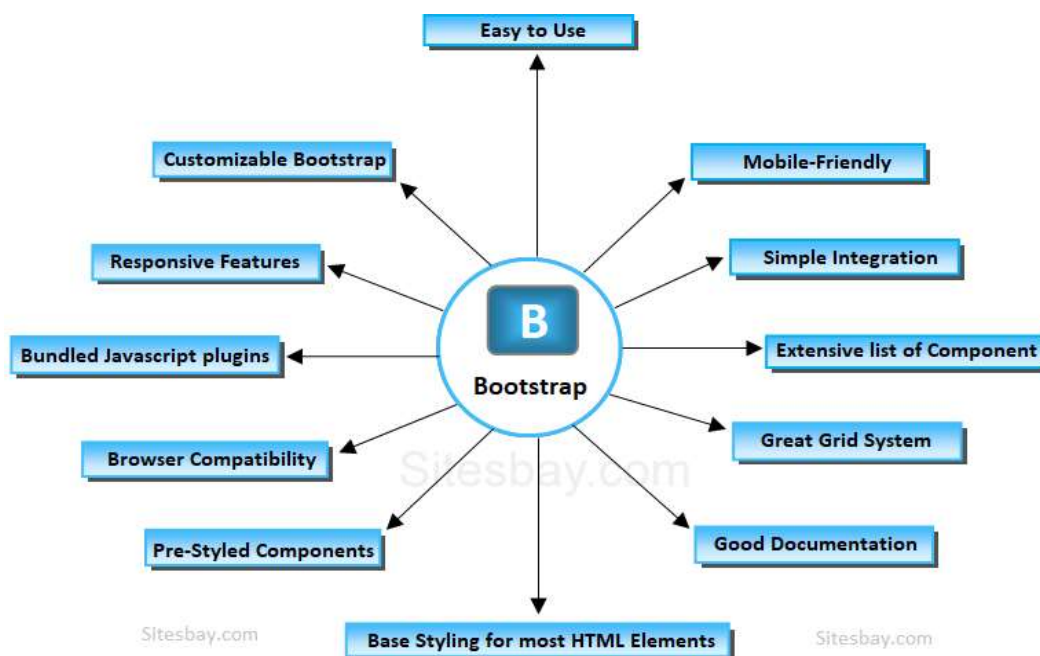
A **JavaScript** egy népszerű programozási nyelv, amelyet interaktív és dinamikus webalkalmazások létrehozására használnak. Ez egy kliensoldali szkriptnyelv, ami azt jelenti, hogy a kliens oldalon (a felhasználó webböngészőjében) fut, nem pedig a szerver oldalon. A JavaScript úgy működik, hogy manipulálja egy weboldal **dokumentumobjektum-modelljét (DOM)**. A DOM a weboldal elemeinek hierarchikus ábrázolása, és a JavaScript használható ezen elemek hozzáadására, eltávolítására, módosítására és dinamikus interakciójára. A JavaScriptet elsősorban a weboldalak interaktivitásának és válaszreakciójának növelésére használják, így a fejlesztők olyan felhasználói felületeket hozhatnak létre, amelyek valós időben reagálnak a felhasználói műveletekre, az oldal frissítése nélkül. Ez magában foglalhat olyan szolgáltatásokat, mint az animációk, az űrlapellenőrzés és a dinamikus tartalomfrissítések. Külső erőforrásokkal, például **API**-kkal (**Application Programming Interface**) és adatbázisokkal való interakcióra is használható olyan technikák használatával, mint az **AJAX**. Ez lehetővé teszi a fejlesztők számára, hogy dinamikus és interaktív webalkalmazásokat hozzanak létre, amelyek oldalfrissítés nélkül képesek lekérni és

kezelni az adatokat külső forrásokból. A JavaScript egyik legfontosabb előnye sokoldalúsága és mindenütt elérhetősége. Minden nagyobb webböngésző támogatja, és sokféle feladatra használható, az egyszerű DOM-manipulációtól a bonyolult alkalmazáslogikáig és adatvizualizációig. Valamint rendkívül moduláris, és számos könyvtár és keretrendszert támogat, ami segíthet leegyszerűsíteni és racionalizálni a fejlesztési folyamatot. Néhány népszerű JavaScript-keretrendszer közé tartozik a React, az Angular és a Vue.js, amelyek előre elkészített összetevőket és eszközöket biztosítanak a gyakori webfejlesztési feladatokhoz. Összefoglalva, a JavaScript egy erőteljes és sokoldalú programozási nyelv, amelyet dinamikus és interaktív webalkalmazások létrehozására használnak. A DOM manipulálásával és a külső erőforrásokkal való interakcióval a JavaScript lehetővé teszi a fejlesztők számára, hogy gazdag és lebilincselő felhasználói élményt hozzanak létre, amely valós időben reagál a felhasználói műveletekre. Sokoldalúságával, mindenütt jelenléte, valamint számos keretrendszer és könyvtár támogatása révén a JavaScript továbbra is a modern webfejlesztés alapvető eszköze.

2.3.6 Bootstrap

A **Bootstrap** egy népszerű front-end webfejlesztési keretrendszer, amelyet arra terveztek, hogy megkönnyítse a fejlesztők számára az érzékeny és mobil-első webalkalmazások létrehozását. A Twitter fejlesztette ki, és jelenleg a Bootstrap csapata tartja karban, a fejlesztők egy csoportja, akik folyamatosan fejlesztik és frissítik a keretrendszert. A Bootstrap előre elkészített HTML, CSS és JavaScript összetevőket biztosít, amelyek segítségével a fejlesztők gyorsan és egyszerűen hozhatnak létre reszponzív és mobilbarát weboldalakat. Ezek az összetevők olyan dolgokat tartalmaznak, mint a gombok, űrlapok, tipográfia és navigációs menük, valamint olyan összetettebb összetevők, mint a modálok, körhinták és harmonikák. Ezeknek az előre beépített összetevőknek a használatával a fejlesztők időt és erőfeszítést takaríthatnak meg a webalkalmazások készítése során, lehetővé téve számukra, hogy alkalmazásuk egyedi jellemzőire és funkcionalitására összpontosítsanak. Ezenkívül a Bootstrap tartalmaz egy hatékony rácsrendszert, amely lehetővé teszi a fejlesztők számára, hogy érzékeny elrendezéseket hozzanak létre, amelyek automatikusan alkalmazkodnak a különböző képernyőméretekhez és eszközökhöz. Az egyik legfontosabb előnye a kiterjedt dokumentáció és a közösségi támogatás. A Bootstrap csapata átfogó dokumentációt biztosít, amely a keretrendszer minden aspektusát lefedi, az alapvető használattól a speciális testreszabásig. Ezenkívül számos online forrás érhető el, beleértve a fórumokat, oktatóanyagokat,

valamint a felhasználók által létrehozott bővítményeket és beépülő modulokat, amelyek segítségével a fejlesztők gyorsan elindulhatnak.

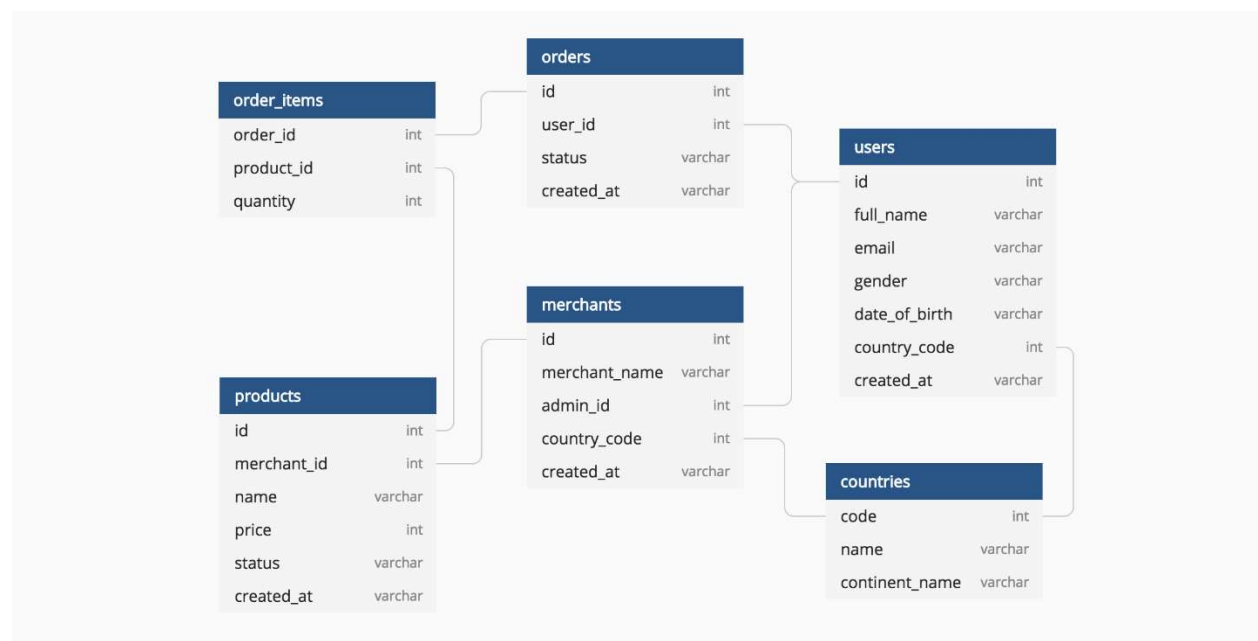


6. ábra: Bootstrap előnyei (Sitesbay)

2.4 Adatbázisok

Az adatbázis olyan rendszerezett adatgyűjtemény, amelyet elektronikusan tárolnak és hozzáférnek. Az adatbázisokat a kisméretű személyes projektektől a nagyvállalati rendszerekig számos alkalmazásban használják. Az adatbázis elsődleges feladata az adatok hatékony és pontos tárolása és visszakeresése. Ez strukturált formátum és az adatokhoz való hozzáférésre és azok manipulálására vonatkozó szabályok segítségével érhető el. A legelterjedtebb adatbázistípus a relációs adatbázis, amely sorokat és oszlopokat tartalmazó táblázatokba rendezi az adatokat. Az adatbázisokat számos különböző iparágban és alkalmazásban használják, beleértve az e-kereskedelmet, a pénzügyeket, az egészségügyet és még sok mást. Ezek nélkülözhetetlenek nagy mennyiségű adat tárolásához és rendszerezéséhez, valamint ahhoz, hogy azokat kontrollált és biztonságos módon hozzáférhetővé tegyék a felhasználók számára. Az adatbázisok legfontosabb jellemzői közé tartozik az adatintegritás, az adatbiztonság és a méretezhetőség. Az adatintegritás biztosítja az adatok pontosságát és következetességét, az adatbiztonság pedig azt, hogy az adatok védve legyenek a jogosulatlan hozzáféréstől és manipulációtól. A skálázhatóság az adatbázis azon képességére

utal, hogy nagy mennyiségű adatot és felhasználókat tud kezelni lassulás vagy összeomlás nélkül. Számos különböző típusú adatbázis áll rendelkezésre, beleértve az SQL-adatbázisokat, a NoSQL-adatbázisokat és a gráf-adatbázisokat. Minden típusnak megvannak a maga erősségei és gyengeségei, és az adatbázis kiválasztása az alkalmazás speciális igényeitől függ. Összességében az adatbázisok a modern számítástechnika kritikus alkotóelemei, és életünk szinte minden területén használatosak. Megbízható és hatékony módot biztosítanak az adatok tárolására és lekérésére, lehetővé téve a vállalkozások és szervezetek számára, hogy megalapozott döntéseket hozzanak, és kiváló minőségű termékeket és szolgáltatásokat nyújtsanak.



7. ábra: Példa adatbázis felépítésére, kapcsolataira (Thong Do, 2022)

2.4.1 SQL

Az **SQL** vagy **Structured Query Language** (strukturált lekérdezőnyelv) egy programozási nyelv, amelyet relációs adatbázisok kezelésére és manipulálására használnak. Ez egy hatékony eszköz a nagy mennyiségű adat kezeléséhez, lehetővé téve a felhasználók számára az adatok beszúrását, frissítését, törlését és adatbázisokból való lekérését. Az SQL szabványosított nyelv, ami azt jelenti, hogy különböző adatbázis-kezelő rendszerekben (**DBMS**) használható, mint például a MySQL, az Oracle és a Microsoft SQL Server. Különböző műveletek végrehajtására szolgál adatbázisokon, például táblák létrehozására, táblák módosítására és táblák törlésére. A táblákon belül

li adatok kezelésére is használható, például rekordok beszúrására, frissítésére és törlésére. Az SQL-lekérdezések meghatározott adatokat kérhetnek le az adatbázisból olyan feltételek megadásával, mint például az összes olyan rekord kiválasztásával, ahol egy adott oszlop bizonyos értékkel rendelkezik. Ez egy deklaratív nyelv, ami azt jelenti, hogy a felhasználók egyszerűen megadják, mit akarnak csinálni, és a DBMS kitalálja, hogyan kell ezt csinálni. Ez nagyon felhasználóbaráttá teszi, mivel lehetővé teszi a felhasználók számára, hogy adatbázisokkal dolgozzanak anélkül, hogy ismerniük kellene az adatbázis felépítésének vagy az adatok tárolásának minden részletét. Az SQL támogatja az összetett lekérdezéseket is, és fejlett műveleteket hajthat végre, például több tábla adatainak egyesítését vagy összesített értékek, például összegek és átlagok kiszámítását.

2.4.2 MySQL

A **MySQL** egy népszerű nyílt forráskódú relációs SQL-alapú adatbázis-kezelő rendszer, amelyet széles körben használnak webfejlesztésben és egyéb alkalmazásokban. Először 1995-ben adták ki, és jelenleg az Oracle Corporation tulajdona. A MySQL méretezhetőségéről, megbízhatóságáról és egyszerű használatáról ismert. Széles körben használják webes alkalmazásokban, és számos népszerű programozási nyelv támogatja, beleértve a PHP, Python és Java nyelveket. A MySQL néhány kulcsfontosságú funkciója magában foglalja az ACID-tranzakciók, tárolt eljárások, triggerek, nézetek és indexek támogatását. Fejlett biztonsági funkciókat is tartalmaz, mint például a titkosítás és a biztonságos jelszókivonat. Úgy tervezték, hogy nagymértékben skálázható legyen, és nagy mennyiségű adatot és felhasználót tud kezelni a teljesítmény feláldozása nélkül. Az adatok biztonsági mentésére és helyreállítására szolgáló funkciókat is tartalmaz, így megbízható választás a kritikus fontosságú alkalmazásokhoz. Alapvető funkciói mellett nagymértékben bővíthető is, hiszen a fejlesztők nagy és aktív közössége hoz létre és oszt meg bővítményeket és bővítményeket. Ezek a bővítmények további funkciók támogatására használhatók, mint például a teljes szöveges keresés és a téradattípusok. Lényegében, a MySQL egy hatékony és rugalmas adatbázis-kezelő rendszer, amely számos alkalmazáshoz jól illeszkedik. Népszerűsége, valamint kiterjedt eszközök és erőforrások ökoszisztémája teszi ezt.

2.4.3 phpMyAdmin

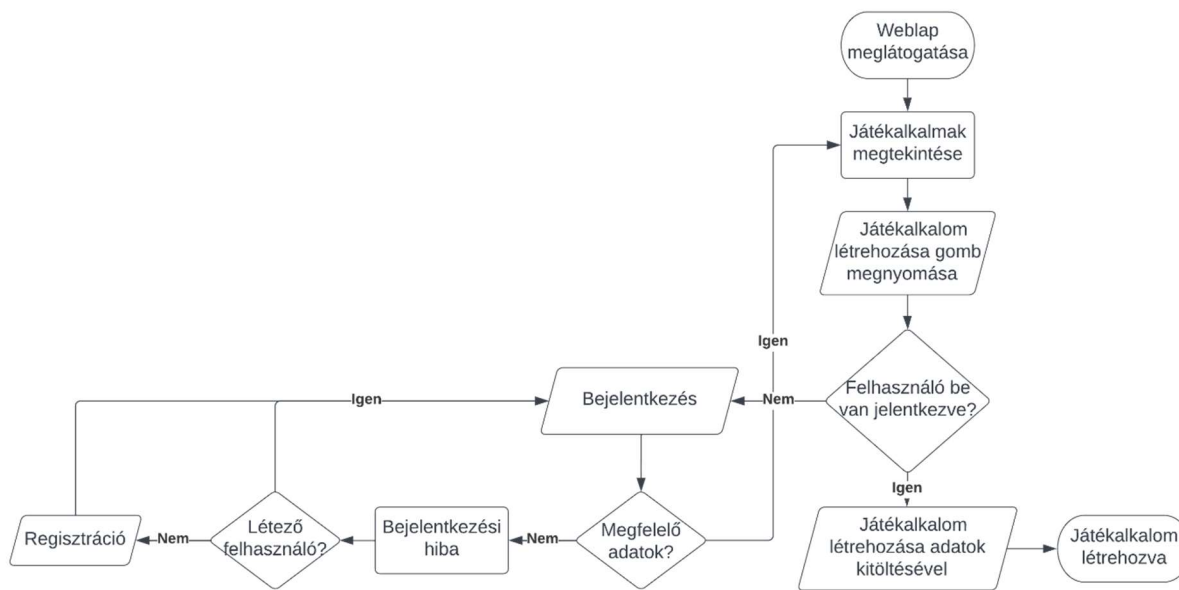
A **phpMyAdmin** egy ingyenes és nyílt forráskódú web-alapú eszköz a MySQL és MariaDB adatbázisok kezelésére. Grafikus felhasználói felületet (GUI) biztosít adatbázisok, táblák, mezők és kapcsolatok kezeléséhez, valamint SQL lekérdezések végrehajtásához. A phpMyAdmint széles körben használják a fejlesztők és rendszergazdák adatbázisok kezelésére, különösen webes alkalmazásokban. A szolgáltatások széles skáláját kínálja, beleértve a felhasználókezelést, az adatok importálását és exportálását, a tábla létrehozását és módosítását, valamint az SQL lekérdezések végrehajtását. Az egyik legfontosabb előnye az egyszerű használat. Úgy tervezték, hogy intuitív és felhasználóbarát legyen, még olyan felhasználók számára is, akik kevés vagy semmilyen tapasztalattal nem rendelkeznek az SQL vagy adatbázis kezelésében. Grafikus felhasználói felülete megkönnyíti a navigációt és a feladatok végrehajtását, míg kiterjedt dokumentációja és közösségi támogatása segítséget és erőforrásokat nyújt a segítségre szoruló felhasználóknak. A phpMyAdmin is nagymértékben testreszabható, számos téma és beépülő modul áll rendelkezésre a funkcionalitás és a megjelenés javításához. Ezenkívül számos különböző nyelven elérhető, így elérhetővé teszi a felhasználók számára szerte a világon.

3. Anyag és módszer

3.1 Anyag

Amikor létrehoztam a webhelyemet, egy olyan problémát akartam megoldani, amellyel sok ember szembesül – azt a nehézséget, hogy olyan embereket találjanak, akik hasonló érdeklődést mutatnak bizonyos társasjátékok és kártyajátékok iránt. A probléma megoldásához némi kutatást kellett végezni, és meg kellett értenem, hogy pontosan mivel is küzdenek az emberek. Rájöttem, hogy sokan arra korlátozódnak, hogy ugyanazzal a kis baráti társasággal játsszanak, és azért küzdenek, hogy új embereket találjanak, akikkel közös érdeklődési körük van. Gyakran véletlen találkozásokra vagy közösségi média bejegyzésekre kellett hagyatkozniuk, hogy új játékosokat találjanak, ami nem mindig volt eredményes. Ennek a problémának a megoldására egy webhelyet szerettem volna csinálni, mint egy platform, ahol a felhasználók könnyen megtalálhatnak és kapcsolatba léphetnek másokkal, akik megosztják érdeklődésüket bizonyos játékok iránt. A felhasználók játékmeneteket hozhatnak létre, és csatlakozhatnak hozzá hasonló gondolkodású személyekkel, tartózkodási helyüktől függetlenül. Ez a lehetőségek teljesen új világát nyitná meg a felhasználók számára, hogy más játékosokat találjanak és kapcsolatba léphessenek velük, és könnyebben találhassanak új játékpartnereket. Ennek érdekében a weblap elkészítésekor ügyelni kellett arra, hogy a létrehozási és a játékmenetekhez való csatlakozási utasítások világosak és könnyen érthetőek legyenek. Tudtam, hogy ez segíteni fogja a felhasználókat a gyors és egyszerű induláshoz. Arról is gondoskodnom kell, hogy a játékok széles választékát kínáljam, hogy a felhasználók megtalálhassák az érdeklődési körüknek megfelelőt. A felhasználói élmény fokozása érdekében engedélyeznem kell a nem regisztrált felhasználóknak, hogy megtekintsék az aktuális játékmeneteket, de regisztrációjukig nem hozhatnak létre vagy csatlakozhatnak játékalomhoz. Ez lehetővé teszi a felhasználók számára, hogy ráérezzenek a webhelyre, mielőtt a fiók létrehozása mellett döntenek. Ezenkívül minden felhasználóhoz hozzá akarok adni egy testreszabható profiloldalt. Ez lehetővé teszi a felhasználók számára, hogy személyre szabják profiljukat, és bemutassák preferenciáikat és érdeklődési köreiket. Ezenkívül egy jegyrendszert szeretnék beépíteni, hogy a felhasználók könnyen segítséget kérhessenek, ha bármilyen problémájuk van, vagy kérdésük van a webhely használatával kapcsolatban. Ezzel biztosítható, hogy a felhasználók könnyen és zökkenőmentesen használhassák a platformot. Végül meg kell bizonyo-

sodnom arról, hogy webhelyem biztonságos, és a felhasználók személyes adatai védettek. Ez kulcsfontosságú a platformom iránti bizalom kialakításához. Összességében arra szeretnék koncentrálni, hogy felhasználóbarát és élvezetes élményt teremtsék mindenki számára, aki felkeresi a webhelyemet, legyen az új vagy visszatérő felhasználó.

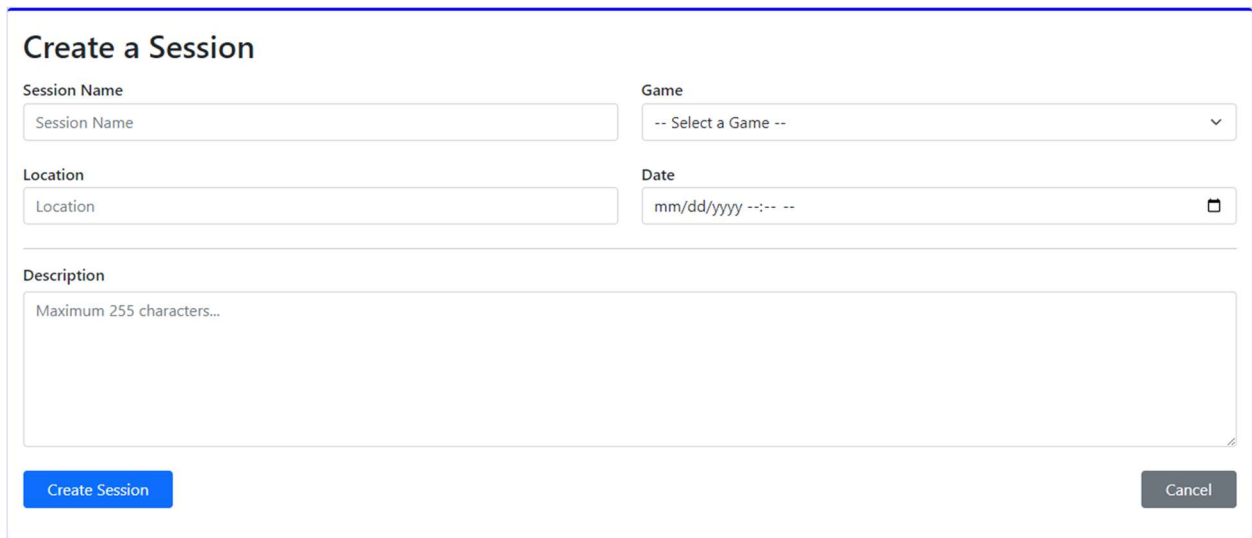


8. ábra: Játékkalkalom létrehozásának folyamata (Saját forrás, 2023)

3.2 Módszer

Különbféle eszközöket és technológiákat használtam a weboldalam létrehozásához, ahol az emberek játékmeneteket készíthetnek és csatlakozhatnak hozzájuk. A VSCode nevű integrált fejlesztői környezetet (IDE) használtam a kód megírásához és kezeléséhez. Azért használtam ezt a fejlesztői környezetet mert ebben érzem magam a legjártasabbnak, mivel egyaránt használom munkahelyi munkám és személyes projektjeim során. A VSCode egy rendkívül felhasználóbarát IDE, amely számos olyan funkciót kínál, amelyek megkönnyítik a kód írását és hibakeresését. Tartalmaz például beépített hibakeresőt, szintaktikai kiemelést, kódkiegészítést és a számomra fontos verziókezelési integrációt. Háttérfejlesztésként (backend), az XAMPP nevű programot használtam szerverként, hogy a webhelyemet helyileg futtassam a számítógépemen a fejlesztés során. Az XAMPP egy nyílt forráskódú webszerver-megoldás, amely magában foglalja az Apache-t, a MySQL-t és a PHP-t, valamint egyéb hasznos eszközöket. Lehetővé tette egy olyan helyi teszte-

lési környezet létrehozását, amely egy élő webservert működését utánozza. Az adatbázishoz a phpMyAdmin-t, egy ingyenes és nyílt forráskódú eszközt használtam a MySQL adatbázisok kezelésére. Felhasználóbarát webes felületet biztosít, amely lehetővé teszi az adatbázistáblák és rekordok egyszerű létrehozását, módosítását és törlését. Az XAMPP és a phpMyAdmin együtt hatékony és rugalmas tesztelési és fejlesztői környezetet biztosított a webhelyemhez. Lehetővé tették számomra, hogy hatékonyabban és eredményesebben dolgozhassak, és segítettek egy stabil és megbízható weboldal felépítésében, amely képes kezelni a felhasználóim igényeit. Továbbá, a webhelyem kezelőfelületének (frontend) felépítéséhez elsősorban HTML-t és CSS-t használtam. Egy kis JavaScriptet is használtam, hogy interaktivitást adjak az oldalnak, viszont a hiányos JavaScript tudásom miatt a weblap nem tartalmaz sok mennyiségű, ilyen nyelven írt kódot. A responszív és mobilbarát kialakításához a Bootstrap-et használtam, egy népszerű frontend keretrendszert, amely előre megtervezett CSS-stílusokat és JavaScript-összetevőket kínál. PHP-t használtam szerveroldali szkriptnyelvként. A PHP egy széles körben használt és népszerű programozási nyelv, amelyet kifejezetten webfejlesztésre terveztek.



Create a Session

Session Name

Game

Location

Date

Description

9. ábra: Játékkalkalom létrehozásának felülete (Saját forrás, 2023)

Find Session

[Create a Session](#)



Uno Party 🧑 1/10

Uno

Budapest 2023-03-14 18:00:00

Come over for a game of Uno!

Available

Join Session



Game Night 🧑 1/4

Pandemic

Nagykőta 2023-04-07 15:00:00

Let's try to beat this game together!

Created by you

Delete



Kitten Chaos 🧑 1/5

Exploding Kittens

Budapest 2023-04-21 21:00:00

Food & drinks included

Created by you

Delete

10. ábra: Játékkalkalmak keresésének felülete (Saját forrás, 2023)

4. Tárgyalás

4.1 Az adatbázis

A webalkalmazás SQL adatbázisa 5 különböző táblából áll össze: „users”, „games”, „sessions”, „users_sessions” és „messages”.

thesis users id : int(255) username : varchar(255) password : varchar(255) email : varchar(255) fullname : varchar(255) phone : varchar(15) address : varchar(255) bio : varchar(255) gender : int(11) birthdate : date pfp : varchar(255) facebook : varchar(255) instagram : varchar(255) twitter : varchar(255) admin : int(1)	thesis sessions id : int(255) game_id : int(255) name : varchar(255) location : varchar(255) description : text date : datetime	thesis games id : int(255) name : varchar(255) description : text max_player_number : int(255) category : varchar(255) icon : text
	thesis users_sessions session_id : int(255) user_id : int(255) role : int(255)	thesis messages message_id : int(11) user_id : int(255) date : datetime subject : varchar(255) message : longtext unread : int(1)

11. ábra: A webalkalmazás adatbázisának felépítése (Saját forrás, 2023)

A „users” tábla információkat tartalmaz a rendszer felhasználóiról, beleértve a felhasználónevüket, jelszavukat, e-mail címüket, teljes nevüket, telefonszámukat, címüket, tetszőlegesen kitölthető leírásukat, nemüket, születési dátumukat, profilképüket, pár közösségi oldalon való elérhetőségüket, valamint azt, hogy rendszergazdák-e vagy sem. Minden felhasználót egyedi azonosító mező azonosít. A „games” tábla tárolja a játszható játékok egyedi azonosító mezejét, nevét, leírását, kategóriáját, a játékosok maximális számát és a hozzárendelt képet. A „sessions” táblázat

tárolja az egyes játékmenetek adatait, mint például a játszott játékot (`game_id`), a játékmenet nevét, helyszínét, leírását, valamint az alkalom dátumát és időpontját. Ennél a táblánál is megtalálható az egyedi azonosító. A „`users_sessions`” tábla egy összekapcsolási tábla, amely összekapcsolja a felhasználókat azokkal a játékokkal, amelyeken részt vesznek. Tárolja az alkalom azonosító mezőjét (`session_id`), az ezen résztvevő felhasználók `user_id`-jét, valamint a játék során betöltött szerepüket (pl. szervező, résztvevő). Végül az üzenettábla tárolja a rendszer felhasználói között küldött üzeneteket. Minden üzenet egyedi `message_id` mezővel, a címzett felhasználói azonosítójával, az üzenet elküldésének dátumával, tárgysorral, az üzenet tartalmával, valamint egy olvasatlan jelzővel rendelkezik, amely jelzi, hogy az üzenetet elolvasták-e vagy sem.

4.2 A webalkalmazás

A webalkalmazást az igények felmérését követően, egy hosszas tervezés után elkezdtem leprogramozni. Az alkalmazás oldalai között egy navigációs sáv segítségével lehet közlekedni. Elsősorban, tartalmaz egy alapértelmezett „Home” kezdőlapot, valamint egy-egy oldal leírást a játékok különböző kategóriáiról. Ezek mellett megtalálható a játékmenet kereső oldala, illetve a bejelentkező és regisztrációs oldal.

4.2.1 Keretrendszer

Az oldal a dinamikus tartalomkezelés érdekében egy indexelő keretrendszert tartalmaz, ami lényegében azt jelenti, hogy az oldal kerete mindig betöltve marad és csak a kereten belüli tartalom változik egy PHP kódnak megfelelően. Navigáció során az indexelő megkapja az adott tartalom oldal azonosítóját és azt tölti be a kereten belül. Így a navigáció sáv használata sokkal könnyebb és dinamikusabb.

```

2      session_start();
3
4      include('php/db_connection.php');
5      include('php/jump.php');
6      include('php/renew_session.php');
7
8      if(isset($_GET['page'])) {
9          switch($_GET['page']) {
10             case 'home'           : $startalom='home.html';           break;
11             case 'card-games'      : $startalom='card-games.html';      break;
12             case 'board-games'     : $startalom='board-games.html';     break;
13             case 'dice-games'      : $startalom='dice-games.html';      break;
14             case 'ttrp-games'      : $startalom='ttrp-games.html';      break;
15             case 'other-games'     : $startalom='other-games.html';     break;

```

12. ábra: Tartalom betöltés kódrészlet 1 (Saját forrás, 2023)

A 12. ábrán látható kódblokk azzal kezdődik, hogy elindít egy böngésző munkamenetet, amely több PHP fájlt is tartalmaz (ezekről később), majd ellenőrzi, hogy be van-e állítva a „page” paraméter. Ha be van állítva, a kód egy switch utasítással határozza meg, hogy melyik oldalt jelenítse meg. Az oldalak különféle HTML és PHP fájlokat tartalmaznak, például bejelentkezést, kijelentkezést, regisztrációt és játékkezelést. Ha az „page” paraméter nincs beállítva, a kód ellenőrzi, hogy a felhasználó be van-e jelentkezve, és ha igen, akkor a tartalmat a kezdőlapra állítja. Ha a felhasználó nincs bejelentkezve, akkor a tartalmat is a kezdőlapra állítja. Végül a kód tartalmazza az „index.html” fájlt, amely tartalmazza a weboldal HTML-kódját.

```

45      }
46      } else {
47          if (isset($_SESSION['login_id'])) {
48              $startalom="home.html";
49          } else {
50              $startalom="home.html";
51          }
52      }
53
54      include("index.html");

```

13. ábra: Tartalom betöltés kódrészlet 2 (Saját forrás, 2023)

Az oldalak közötti irányítás érdekében szolgál még egy „jump” nevű PHP-függvény is, amely két paraméter vesz fel.

- \$hova: az az URL, ahová a felhasználó a megadott idő elteltével át lesz irányítva.
- \$mikor: az az idő (ezredmásodpercben), amely után a felhasználó átirányításra kerül.

Ez a függvény létrehoz egy JavaScript kódblokkot, amely a függvény meghívásakor lefut. A JavaScript kód a setTimeout függvényt használja a window.location.href utasítás végrehajtásának késleltetésére, amíg a megadott idő el nem telik. Ez az utasítás az aktuális ablak URL-jét a megadott URL-re állítja be, ami a felhasználót az új oldalra irányítja át.

```
1  <?php
   47 references
2  function jump($hova, $mikor)
3  {
4      print '
5          <script>
6              setTimeout("window.location.href=' . "$hova" . ',' . $mikor . ');
7          </script>
8      ';
9  }
10 >>
```

14. ábra: A jump függvény kódja (Saját forrás, 2023)

4.2.2 Adatbázishoz való csatlakozás

Egy PHP-szkript kapcsolatot hoz létre a MySQL adatbázishoz. Az adatbázis hitelesítő adatait a rendszer a következő változóknál tárolja: \$server, \$user, \$password és \$db_name. A mysqli_connect() függvény ezután ezekkel a hitelesítő adatokkal kerül meghívásra, hogy kapcsolatot létesítsen a MySQL adatbázissal. Ez a függvény egy kapcsolatobjektumot ad vissza, amely a \$conn változóban van tárolva. A szkript feltételezi, hogy a MySQL szerver ugyanazon a gépen fut, mint a PHP szkript (localhost), és a jelszó nélküli MySQL root felhasználónak megfelelő jogosultságai vannak a dolgozat adatbázis eléréséhez. Az adatbázis-kezelő rendszerekben, például a MySQL-ben vagy a MariaDB-ben a „root” felhasználó egy speciális felhasználó, aki teljes adminisztrátori jogosultsággal rendelkezik az adatbázis-rendszeren végzett műveletek végrehajtására.

```

1  <?php
2
3  $server = "localhost";
4  $user = "root";
5  $password = "";
6  $db_name = "thesis";
7
8  $conn = mysqli_connect($server, $user, $password, $db_name);
9
10 ?>

```

15. ábra: A kapcsolat létrehozó kód (Saját forrás, 2023)

4.2.3 Kezdőlap

A weboldal kezdőlapja a Bootstrap kártyarendszerének köszönhetően letisztult és modern dizájnnal rendelkezik, amely megkönnyíti a navigációt és a használatát. A főkártya az oldal fókuszpontja, amely két sorra van osztva. Az első sor további két oszlopra oszlik, ahol a bal oldali oszlopban egy üdvözlő üzenet található, amely üdvöзли a látogatókat, és megadja a weboldal alaphangját. A jobb oldali oszlop az aktuális játékmenetekből hármat jelenít meg (a megjelenítés kódjáról később) és tartalmazza a hozzá kapcsolatos információkat, aminek célja, hogy a látogatókat lekösse és felkeltse érdeklődésüket az oldal iránt. A főkártya második sora néhány példajátékot mutat be különböző játékkategóriákból. Ez egy nagyszerű funkció, mert segít a látogatóknak abban, hogy képet kapjanak arról, milyen típusú játékokat találhatnak az oldalon.

```

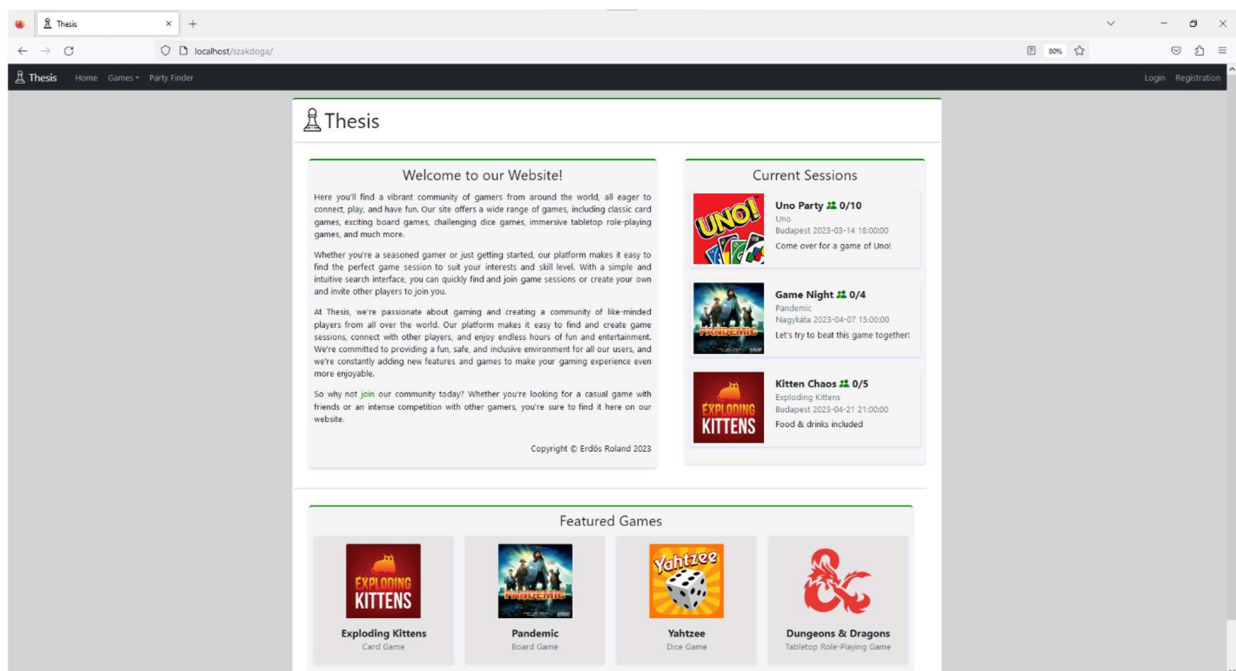
<div class='col-lg-7'>
  <div class="card-body">
    <div class='col-lg-12'>
      <div class='card wrapper' style='height: auto; background-color: whitesmoke; padding: 10px; border-top: 3px solid green;'>
        <h3 style="text-align: center; margin-bottom: 15px;">Welcome to our Website!</h3>
        <p>Here you'll find a vibrant community of gamers from around the world, all eager to connect, play, and have fun. Our site
        board games, challenging dice games, immersive tabletop role-playing games, and much more.</p>
        <p>Whether you're a seasoned gamer or just getting started, our platform makes it easy to find the perfect game session to
        search interface, you can quickly find and join game sessions or create your own and invite other players to join you.</p>
        <p>At Thesis, we're passionate about gaming and creating a community of like-minded players from all over the world. Our pl
        other players, and enjoy endless hours of fun and entertainment. We're committed to providing a fun, safe, and inclusive en
        and games to make your gaming experience even more enjoyable.</p>
        <p>So why not <a href="index.php?page=registration">join</a> our community today? Whether you're looking for a casual game
        to find it here on our website.</p>
        <p style="text-align: end; margin-top: 18px;">Copyright &copy; Erdős Roland 2023</p>
      </div>
    </div>
  </div>
</div>

```

16. ábra: Kezdőlap kódrészlet 1 (Saját forrás, 2023)

A 16. ábrán látható kódrészletben láthatjuk, hogy a Bootstrap kártyarendszerének felépítését. A `<div>` (a divízió rövidítése) egy alapvető építőelem, amelyet a weboldalak tartalmának rendezé-

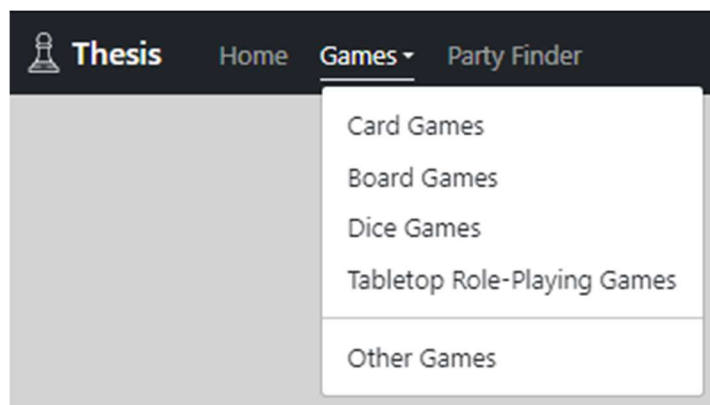
sére és stílusára használnak. Ebben a kódban hozzárendeltem a „col-lg-7” osztályt, amely az elérhető 12 oszlopból 7 szélességű oszlopot hoz létre nagy képernyőn (lg = large – azaz nagy). Ez azt jelenti, hogy a kártya a képernyő szélességének nagy részét elfoglalja a nagyobb képernyőkön. Alatta pedig az előzőleg létrehozott oszlopban létrehozásra kerül egy „card-body” osztályú <div> is, a kártya tartalma ebben az elemben fog szerepelni. Az ezt követő <div> az első logikája alapján működik, viszont említésre méltó a „wrapper” osztály, ami egy tárolóelem, mely egy weboldal teljes tartalmának becsomagolására szolgál. A wrapper segít a tartalom középpontba helyezésében az oldalon, és egységes elrendezést tart fenn a különböző képernyőméretek. Továbbá, itt megadtam a stíluselemeit is CSS segítségével. A többi kód csupán fejléc és bekezdés HTML címkékből áll. A főkártya második sorában, 4 különböző kisebb kártyát hoztam létre, amelyek tartalmazzák egyes játékok képét, nevét és kategóriáját. Annak érdekében, hogy ez a négy kártya tökéletesen elférjen egymás mellett, az előzőhöz hasonló logikát követtem. Hozzárendeltem mindegyik kártyát tartalmazó <div>-hez a „col-lg-3” osztályt, így a 12 rendelkezésre álló oszlopot fel tudja osztani 4 különböző részre, a kártyák egy sorban való elhelyezését eredményezve.



17. ábra: Kezdőlap felülete (Saját forrás, 2023)

4.2.4 Játék kategóriák

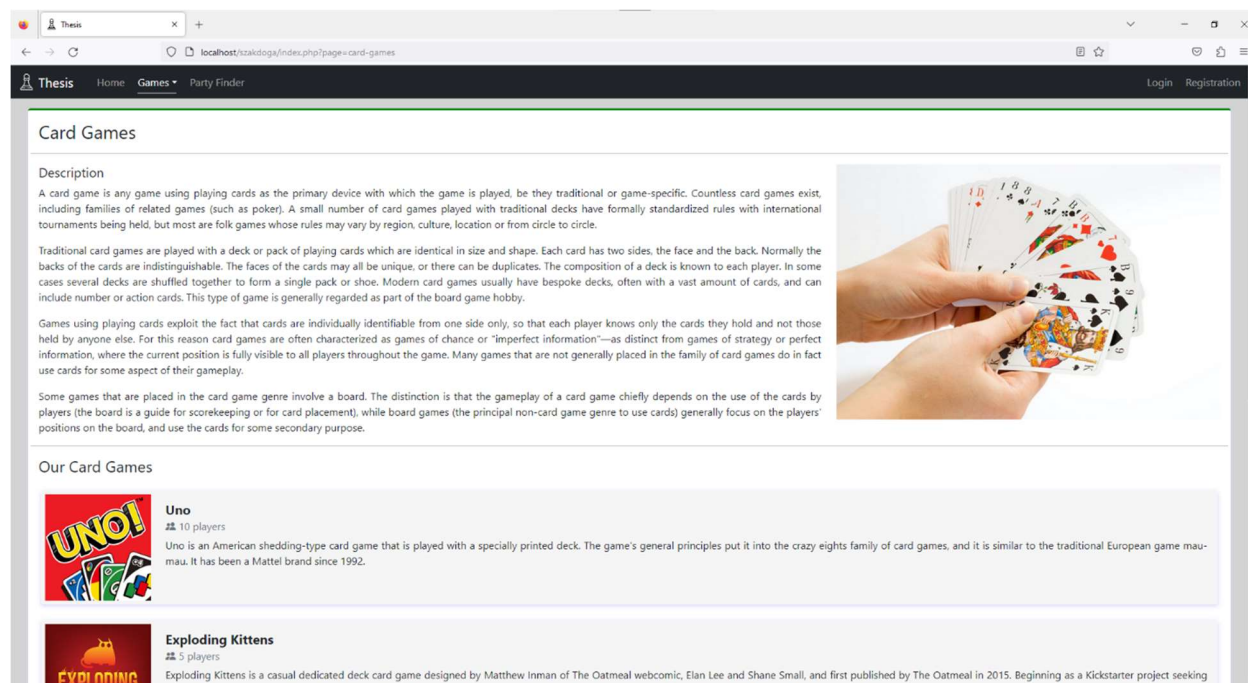
A kezdőlap után szerettem volna, ha lenne egy-egy oldal, ahol a látogatók képesek olvasni a különböző játék kategóriákról, amik jelen vannak a rendszerben. Emellett hasznosnak tartottam azt is, hogy ugyan itt láthassák, hogy milyen játékok állnak rendelkezésre ezekben a kategóriákban. Ezeket az oldalakat a felső navigációs sávon lehet elérni, ahol egy legördülő menü listájából tudjuk kiválasztani melyikről szeretnénk olvasni.



18. ábra: Legördülő menü (Saját forrás, 2023)

Annak ellenére, hogy több kategória van, az oldalak felépítése ugyan az mindegyiknél, csupán a tartalom más. Elkészítés példaként az elsőt szeretném használni, ami ebben az esetben a kártyajátékok. Az oldal elkészítésekor a fő szempontom az volt, hogy könnyen átlátható legyen és elkerüljem a fölösleges elemeket. A kód a „card-body” osztályú `<div>` címkével kezdődik, amely egy tároló létrehozására szolgál a kártyajátékok teljes szakaszához. A tároló belsejében van egy `<div>` címke a „row” osztályával, amely egy vízszintes sor létrehozására szolgál, amely tartalmazza a kártyajátékok fejlécet, azok leírását és egy képet. A fejléc egy 3. szintű `<h3>` címsor, amit egy vízszintes választóvonal követ a `<hr>` felhasználásával, amely egy sort ad a címsor alá. Ezután van egy „col-lg-8” osztályú `<div>` címke, amely a kártyajátékok leírását tartalmazza bekezdések segítségével. A másik `<div>` címke a „col-lg-4 d-md-block d-none mb-3” osztállyal rendelkezik, amely kártyajátékok képét jeleníti meg. A kép csak nagy képernyőkön jelenik meg (a „d-md-block” és a „d-none” osztályok használatával). A Bootstrap-ban a d-md-block egy segédosztály, amely egy elem láthatóságának szabályozására szolgál az eszköz mérete alapján. A mi esetünkben az elem blokkszintű elemként való megjelenítésére szolgál közepes vagy nagyobb

képernyőmérettel rendelkező eszközökön. A d-none osztály pedig egy elem elrejtésére szolgál minden eszközön.



19. ábra: Kártyajátékok leírásának felülete (Saját forrás, 2023)

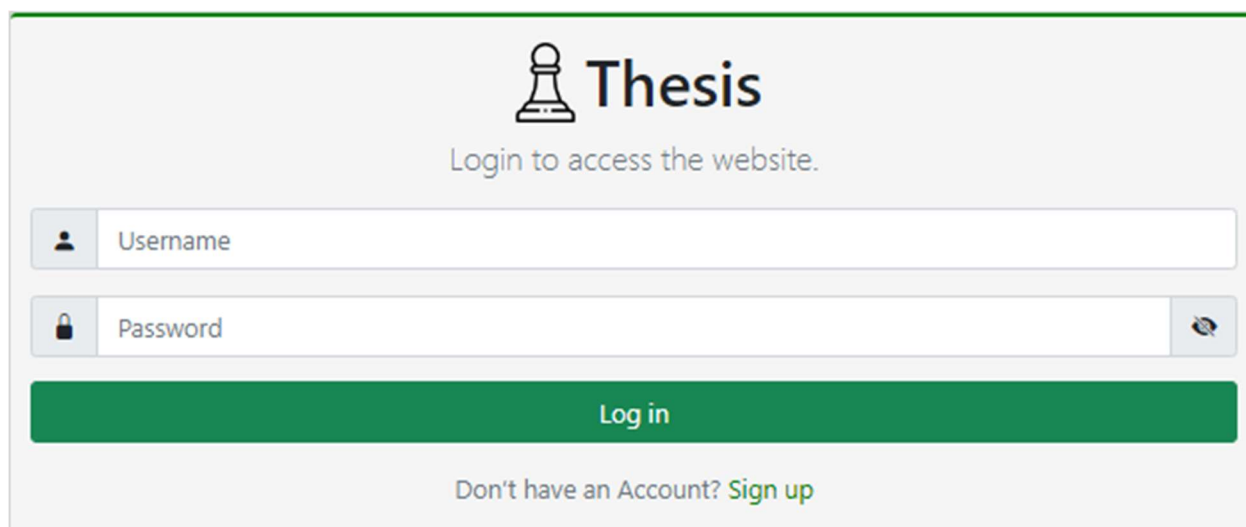
A leírás és kép után következnek a rendszerben jelenlévő kártyajátékok. Ezeknek felsorolása az „Our Card Games” fejléc után következik egy „card-body” osztályú <div> címkében. A tároló belsejében van egy PHP kód, amely egy „games” nevű adatbázistáblából kéri le az adatokat, és kártyaformátumban jeleníti meg. A kód egy while ciklust használ az adatsorok lekéréséhez és kártyaként való megjelenítéséhez. Minden kártya egy <div> címkébe van zárva, amelyen a „card” osztály szerepel. Tartalmazza a kártyajáték képét, nevét, a játékosok maximális számát, valamint a kártyajáték leírását, mindezt az adatbázisban szereplő megfelelő oszlopokból kigyűjtve.

```
<?php
$result = mysqli_query($conn,"SELECT * FROM games WHERE category = 'Card Game'");
while($row = mysqli_fetch_array($result)){
    print"
    <div class='card' style='flex-direction: row; height: auto; background-color: whitesmoke;'>
        <img src='";echo $row['icon']; print "' class='card-img-top' id='card_image' style='min-height: 175px; min-width: 175px; max-height: 175px; max-width: 175px; object-fit: fit; margin-top: auto; margin-bottom: auto; padding: 6px;'>
        <div class='card-body'>
            <h5 class='card-title'><strong>"; echo $row['name']; print"</strong></h5>
            <h6 class='card-subtitle mb-2 text-muted'><i class='bi bi-people-fill'></i> "; echo $row['max_player_number']; print " players</h6>
            <p class='card-text text-break'>"; echo $row['description']; print "</p>
        </div>
    </div>";
}
}>
```

20. ábra: Kártyajátékok lekérdezésének kódja (Saját forrás, 2023)

4.2.5 Bejelentkezés és regisztráció

A bejelentkezéshez a 21. ábrán látható űrlapot kell használatba venniük a felhasználóknak. A megfelelő adatbevitel után a „Log in” gombra kattintva tudnak belépést nyerni a látogatól. Amennyiben nem megfelelő az adatbevitel, vagy pedig nem létező felhasználó próbál bejelentkezni, akkor egy hibaüzenet jelzi azt. Amennyiben nem rendelkezik valaki felhasználóval, az vagy a gomb alatti „Sign up” szöveghivatkozásra kattintva, vagy pedig a navigációsávban található „Registration” gomb megnyomásával tud a regisztráló űrlapra jutni. Megfelelő kitöltés után felkerül a felhasználó a rendszerbe. Amennyiben már foglalt a felhasználónév, vagy használatban van a beírt e-mail, az hibás regisztrációt fog eredményezni.



21. ábra: Bejelentkezési űrlap (Saját forrás, 2023)

A „Log in” gomb megnyomását követően egy PHP kód fut le a háttérben, amely kezeli a bejelentkezési űrlap elküldését. Ellenőrzi, hogy a `$_POST` szuperglobális változó be van-e állítva, és lekéri belőle a beküldött felhasználónevet és jelszót. Ezután ellenőrzi, hogy a felhasználónév létezik-e az adatbázisban, és hogy a megadott jelszó egyezik-e az adott felhasználó tárolt jelszavával. Ha a hitelesítési adatok helyesek, különféle munkamenet-változókat állít be a felhasználó adataival és átirányítja őt a kezdőlapra. Ha a hitelesítési adatok helytelenek, a `$_SESSION['login_error']` változót igaz értékre állítja, és 5 másodperces késéssel visszairányítja a felhasználót a bejelentkezési oldalra. Ha valami kezeletlen problémába ütközik a kód, akkor kiír egy üzenetet, amely felszólítja a felhasználót, hogy kattintson a bejelentkezés menüpontra.


```

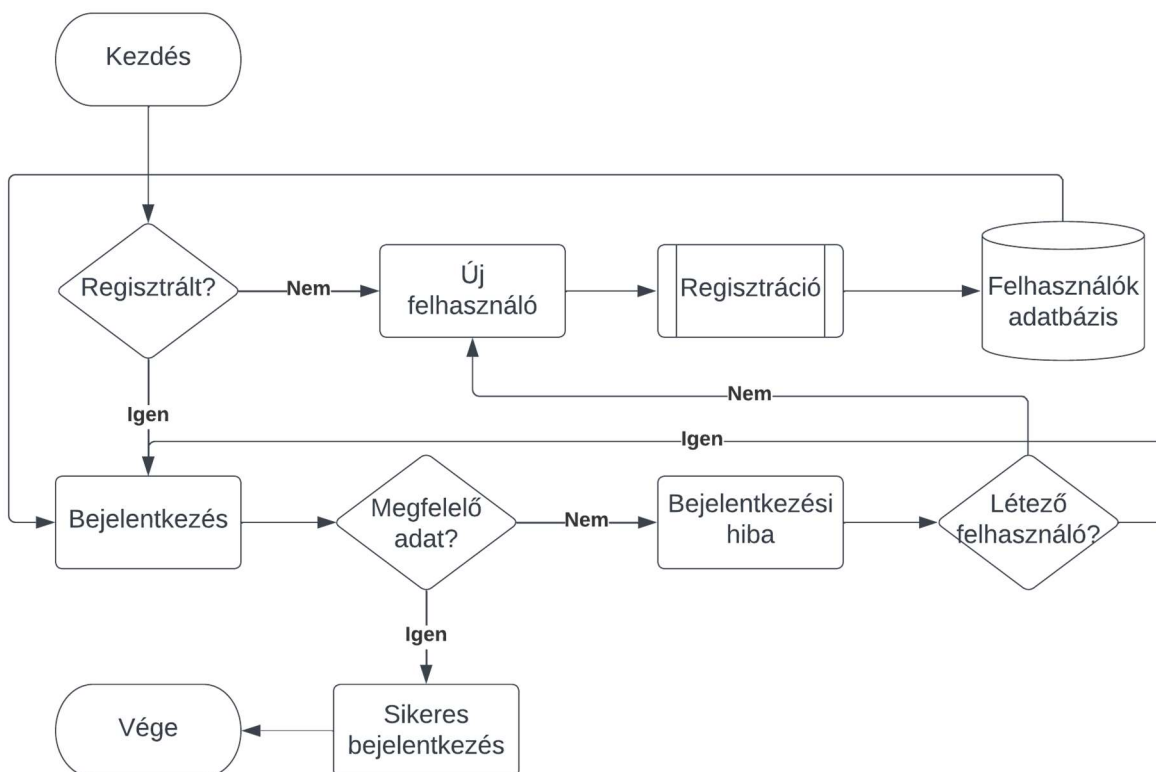
1  <?php
2      if(isset($_POST)){
3
4          $username = $_POST['username'];
5          $password = $_POST['password'];
6
7          if(isset($username)) {
8              $result=mysqli_query($conn,"SELECT * FROM users WHERE username='$username'
9              AND password='$password'");
10             $row=mysqli_fetch_assoc($result);
11             if(mysqli_num_rows($result)==1) {
12                 $_SESSION['login_id']=$row['id'];
13                 $_SESSION['login_nev']=$row['username'];
14                 $_SESSION['login_admin']=$row['admin'];
15                 $_SESSION['login_email']=$row['email'];
16                 $_SESSION['login_pfp']=$row['pfp'];
17                 $_SESSION['login_fullname']=$row['fullname'];
18                 $_SESSION['login_phone']=$row['phone'];
19                 $_SESSION['login_address']=$row['address'];
20                 $_SESSION['login_bio']=$row['bio'];
21                 $_SESSION['login_bday']=$row['birthdate'];
22                 $_SESSION['login_gender']=$row['gender'];
23                 $_SESSION['login_fb']=$row['facebook'];
24                 $_SESSION['login_ins']=$row['instagram'];
25                 $_SESSION['login_twit']=$row['twitter'];
26                 jump("index.php?page=home",1);
27
28             } else {
29                 $_SESSION['login_error']=true;
30                 jump("index.php?page=login",5);
31             }
32
33         } else {
34             print "Bejelentkezéshez kattintson a Bejelentkezés menüpontra!!!";
35         }
36     }
37
38     ?>

```

22. ábra: Bejelentkezés backend kódja (Saját forrás, 2023)

A kijelentkezést a „Logout” gomb megnyomásával tudjuk megtenni, ennek backend kódja nagyon egyszerű. Szimplán megszünteti az aktuális munkamenetet a „session_destroy()” funkcióval, valamint visszairányítja a felhasználót a kezdőlapra. Regisztráció esetén a felhasználónak ki kell töltenie a megfelelő űrlapot. Amint ez megtörtént ideális adatokkal és rányomott a „Sign up” gombra, akkor a háttérben lefut a hozzárendelt PHP kód. A regisztrációs űrlap elküldésekor a kód ellenőrzi, hogy a beküldött felhasználónév vagy email cím már létezik-e az adatbázisban. Ha igen, a kód beállít egy munkamenet-változót, jelezve, hogy regisztrációs hiba történt, és 5 má-

sodperc elteltével visszairányítja a felhasználót a regisztrációs oldalra. Ellenkező esetben a kód beilleszti az új felhasználó adatait az adatbázisba, és beállít egy munkamenet-változót, amely jelzi a sikeres regisztrációt, és 5 másodperc múlva átirányítja a felhasználót a bejelentkezési oldalra.



23. ábra: Regisztráció folyamata (Saját forrás, 2023)

A kód először azt ellenőrzi, hogy be van-e állítva a POST szuperglobális változó, ami azt jelzi, hogy a regisztrációs űrlapot elküldték. A következőleg lekéri a felhasználónevet, jelszót és e-mail címet a beküldött űrlapadatokból, és külön változókban tárolja. Majd lekérdezi az adatbázist annak ellenőrzésére, hogy a beküldött felhasználónév vagy e-mail cím már létezik-e a felhasználók táblájában. Ha létezik ugyanazzal a felhasználónévvel vagy e-mail címmel rendelkező felhasználó, a kód beállít egy munkamenet-változót, amely regisztrációs hibát jelez, és visszairányítja a felhasználót a regisztrációs oldalra. Ha nem létezik, a kód beszúrja az új felhasználó adatait a felhasználók táblájába, és lekéri az új felhasználó azonosítóját az adatbázisból. A kód beállít egy munkamenet-változót, amely jelzi a sikeres regisztráció megtörténtét, és átirányítja a felhasználót a bejelentkezési oldalra.

```

1 <?php
2 if(isset($_POST)){
3
4     $username = $_POST['username'];
5     $password = $_POST['password'];
6     $email = $_POST['email'];
7
8     $user_check_query = "SELECT * FROM users WHERE username='$username' OR email='$email' LIMIT 1";
9     $result = mysqli_query($conn, $user_check_query);
10    $user = mysqli_fetch_assoc($result);
11
12    if ($user) {
13        $_SESSION['registration_error']=true;
14        jump("index.php?page=registration",5);
15    } else {
16        $sql="INSERT INTO users VALUES ('','$username','$password','$email', '', '', '', 'New user', '', '', 'uploads/default_pfp.png', '0', '', '', '')";
17        mysqli_query($conn,$sql);
18        $sql="SELECT id FROM users WHERE username = '$username'";
19        $result = mysqli_query($conn,$sql);
20        while($row=mysqli_fetch_assoc($result)) {
21            $new_id=$row['id'];
22        }
23
24        $_SESSION['reg_suc']=true;
25        jump("index.php?page=login",5);
26    };
27 }
28
29
30 ?>

```

24. ábra: Regisztráció kódja (Saját forrás, 2023)

4.2.6 Profiloldal

A profiloldal csak akkor érhető el, ha a felhasználó bejelentkezett a weboldalra. Ez egy általános biztonsági intézkedés, amelyet a webhelyeken alkalmaznak annak biztosítására, hogy csak az arra jogosult felhasználók tekinthessék meg személyes adataikat és fiókadataikat. Amikor egy felhasználó bejelentkezik, létrejön a munkamenet, és hitelesítésre kerül. Ez azt jelenti, hogy hozzáférést kapnak a webhely bizonyos oldalaihoz és funkcióihoz, amelyek nem érhetők el mások számára. Ebben az esetben a profiloldal azon szolgáltatások közé tartozik, amelyekhez csak a weboldalra bejelentkezett felhasználók férhetnek hozzá. Az oldal felépítését és elrendezését HTML és CSS (Bootstrap) segítségével valósítottam meg. Egyszerű, de professzionális kialakítást szerettem volna, amely fontos információkat tartalmaz a felhasználóról. Ezután PHP kódot integráltam az oldalba, hogy dinamikusan jelenítsem meg a felhasználói információkat. A munkamenet-változókat használtam fel a felhasználói adatok tárolására, például a nevüket, profilképüket, nemüket, címüket, telefonszámukat és a közösségi média elérhetőségeiket. Ez lehetővé tette az adatok lekérését és az oldalon való megjelenítését, így személyre szabottabbá tettem a felhasználói profilt. Annak érdekében, hogy a felhasználók feltölthessék profilképüket, hozzáadtam egy fájlbeviteli mezőt, és létrehoztam egy PHP-szkriptet a fájlfeltöltési folyamat kezelésére. A szkript ellenőrzi a feltöltött fájlt, áthelyezi egy kijelölt mappába, és frissíti a felhasználó profilképének URL-címét a munkamenet-változóban.

```

<?php
if($_SERVER['REQUEST_METHOD'] == "POST")
{
    if(isset($_FILES['uploadPfp']))
    {
        $file_name = $_FILES['uploadPfp']['name'];
        $file_type = $_FILES['uploadPfp']['type'];
        $file_tmp_name = $_FILES['uploadPfp']['tmp_name'];
        $file_size = $_FILES['uploadPfp']['size'];
        $username = $_SESSION['login_new'];

        $target_dir = "uploads";

        if(move_uploaded_file($file_tmp_name,$target_dir."/".$file_name))
        {
            $q = "UPDATE users SET pfp = '$target_dir/$file_name' WHERE username = '$username'";
            $r = mysqli_query($conn,$q);

            if(mysqli_affected_rows($conn) == 1)
            {
                echo "<div class='col-lg-4 m-auto rounded-top' style='background-color:rgb(234, 255, 240);'>
                    <p class='text-center' id='success'><b>Profile has been successfully updated!</b></p>
                    <p class='text-center text-muted lead' style='font-size:80%;padding-bottom: 5px;'>Redirecting...</p>
                </div>";
                renew_session($conn);
                jump("index.php?page=profile",1500);
            }
            else
            {
                echo "<p>A system error has been occurred</p>".mysqli_error($conn);
            }
        }
        else
        {
            echo "File can not be uploaded";
        }
    }
}

```

25. ábra: Profilképfeltöltés kódja (Saját forrás, 2023)

Végül hozzáadtam néhány JavaScriptet az oldalhoz a felhasználói interakciók kezeléséhez, például a fénykép kiválasztása gombra kattintva a fájlbeviteli mező megnyitásához és a kiválasztott kép előnézetének megtekintéséhez az oldalon. Amennyiben a felhasználó módosítani szeretné az adatait, elhelyeztem ennek érdekében egy gombot, ami megnyomásával egy profilszerkesztő űrlaphoz kerül. Kitöltés után egy PHP kód fut le a háttérben. A szkript ellenőrzi, hogy van-e POST kérés. Ha nem, akkor nem történik semmi. A szkript adatokat gyűjt ki a POST-kérésből, beleértve a felhasználó felhasználónevét, jelszavát és új profilinformációit (teljes név, e-mail, telefon, cím stb.). A profilinformációkban szereplő egyszeres vagy kettős idézőjeleket helyettesíti azok megtisztított verzióival. A szkript ellenőrzi, hogy a felhasználó be van-e jelentkezve, és ellenőrzi felhasználónevét és jelszavát az adatbázisban. Ha nincs egyezés, a rendszer egy hibaüzenetet tárol a munkamenetben, és a felhasználót visszairányítja a profilszerkesztő oldalra. Ha a felhasználó

ló ellenőrzése megtörtént, a szkript frissíti a profilinformációit az adatbázisban, és sikeres üzenetet jelenít meg a felhasználónak. Ezután a kód megújítja a munkamenetet, és visszairányítja a felhasználót a profiloldalára.

The screenshot shows a web interface for editing a user profile. On the left is a sidebar with a profile picture placeholder (a grey circle with a white 'u' shape) and the username 'user'. Below the picture are links for 'New user' and 'Home'. Further down are social media links for Facebook, Instagram, and Twitter, each with a corresponding input field. The main content area is titled 'Your Profile' and contains several form fields: 'Full Name' (input field with 'User Name'), 'Email' (input field with 'user@gmail.com'), 'Phone' (input field with '36201234567'), and 'Address' (input field with 'Home'). Below these is a 'Bio' section with a text area containing 'New user'. Further down are 'Gender' (a dropdown menu set to 'Male') and 'Birthdate' (a date picker set to '06 / 14 / 2000'). At the bottom are 'Password' and 'Confirm Password' input fields, followed by 'Save Changes' and 'Cancel' buttons.

26. ábra: Profilszerkesztési űrlap (Saját forrás, 2023)

4.2.7 Üzenetek és jegyrendszer

Az üzenetoldal egy webalkalmazás része, ahol a felhasználók tájékoztatást kapnak munkameneveikről, és egyéb fontos értesítéseket kapnak. Az oldal úgy van kialakítva, hogy megjelenítse a felhasználónak küldött összes üzenetet, beleértve a játékmenettevékenységről szóló értesítéseket is, például amikor valaki csatlakozik vagy kilép a felhasználó által létrehozott játékmenetből, vagy ha egy rendező töröl egy létrehozott alkalmat. Az üzenetek oldalon akkor is megjelennek az értesítések, ha egy rendszergazda töröl egy játékalkalmat. Az adminisztrátor felhasználók számára itt jelennek meg a más felhasználók által beküldött leveleket is, amiket akkor írnak, ha segítségre van szükségük vagy felmerül valamilyen kérdésük. Az üzenetoldalt úgy alakítottam ki, hogy könnyű hozzáférést biztosítson a felhasználó üzeneteihez. Minden üzenet kártya formájában jelenik meg, az üzenet tárgyával, dátumával és időpontjával, valamint magával az üzenet szövegével. A még el nem olvasott üzenetek világosszürke színnel vannak kiemelve, az olvasott üzenetek pedig fehér háttérrel jelennek meg. Az üzenetoldal tetején két gomb található, amelyek segítségével a felhasználó kezelheti üzeneteit. Az egyik gomb segítségével a felhasználó törölhe-

ti az összes olvasott üzenetet, míg a másik gombbal az összes üzenetet olvasottként jelölheti meg.

```
<?php
    if(isset($_SESSION['login_id'])){
        if(isset($_POST)){

            $user_id = $_SESSION['login_id'];
            $messageid_check_query = "SELECT message_id FROM messages WHERE user_id = '$user_id' and unread = '1'";
            $result = mysqli_query($conn, $messageid_check_query);
            while($row = mysqli_fetch_array($result)){

                $message_id = $row['message_id'];
                $sql="UPDATE messages SET unread = '0' WHERE message_id = '$message_id'";
                mysqli_query($conn,$sql);
            }
            jump("index.php?page=message",5);
        }
    }
}
```

27. ábra: Összes olvasatlan üzenet olvasottként jelölő kód (Saját forrás, 2023)

Először is a kód ellenőrzi, hogy a felhasználó be van-e jelentkezve a „login_id” munkamenet-változó meglétének ellenőrzésével. Ha a felhasználó be van jelentkezve és POST kérés érkezik, a kód lekéri az összes olyan üzenetazonosítót, ahol a felhasználói azonosító megegyezik a bejelentkezett felhasználói azonosítóval, és az olvasatlan állapot 1-re van állítva (az üzenetet nem olvasták el). Ezután a kód végigfut a lekérdezés eredményének minden során a `mysqli_fetch_array()` függvény használatával, és minden üzenetazonosítónál 0-ra frissíti az olvasatlan állapotot. Végül a kód a `jump()` függvény segítségével 5 másodperces késleltetéssel átirányítja a felhasználót az üzenetek oldalra.



28. ábra: Üzenetek oldal felülete (Saját forrás, 2023)

A weboldalon található egy egyszerű támogatási jegyűrlap, amely lehetővé teszi a felhasználók számára, hogy egyszerűen elküldjék problémáikat vagy aggályaikat. Az űrlapon a felhasználónak meg kell adnia a problémájának tárgyát és részletesen leíró üzenetét. Beküldés után a jegyet elküldi az adminisztrátoroknak, akik megtekintheti ezeket, és ennek megfelelően válaszolhatnak rá. Ez az üzenetküldő rendszer előnyös a felhasználók számára, mivel közvetlen és hatékony módot biztosít számukra, hogy problémáikat közöljék vagy kérdéseiket feltegyék a weboldal adminisztrátorainak. Az űrlap egyszerűsége megkönnyíti a felhasználók számára a jegyek benyújtását anélkül, hogy bonyolult menükön vagy oldalakon kellene navigálnia. Ez az egyszerűsített folyamat segíthet a felhasználóknak kényelmesebben jelenteni problémáikat, és növelheti annak valószínűségét, hogy a rendszergazdák időben és hatékonyan válaszolnak. PHP kód felelős a weboldalon található jegyrendszer kezeléséért. Először a „login_id” session változó meglétének ellenőrzésével megnézi, hogy a felhasználó be van-e jelentkezve. Ezután a \$_POST superglobal létezésének ellenőrzésével vizsgálja, hogy elküldték-e az űrlapadatokat. Ha mindkét feltétel igaz, akkor lekéri a felhasználói azonosítót és a felhasználónevet a munkamenetből, valamint a tárgy- és üzenetmezőket az űrlapról. Ezután lekérdezi az adatbázist, hogy lekérje az összes rendszergazdai felhasználó azonosítóját. Ezután végigfut az eredménykészleten, és minden adminisztrátorhoz új üzenetet szűr be az üzenettáblázatba. Az üzenet tartalmazza a jegy tárgyát és magát az üzenetet, valamint az azt küldő felhasználó felhasználónevét. Ha az üzenet sikeresen beillesztésre került az adatbázisba, a kód a „ticket_suc” session változót IGAZ értékre állítja, je-

lezve, hogy az üzenet sikeresen elküldve. Az oldal ezután az ugrás funkció segítségével 5 másodperces késleltetéssel a támogatási oldalra kerül.

```
<?php
if(isset($_SESSION['login_id'])){
    if(isset($_POST)){
        $user_id = $_SESSION['login_id'];
        $username = $_SESSION['login_nev'];
        $subject = $_POST['subject'];
        $message = $_POST['message'];

        $admin_check_query = "SELECT id FROM users WHERE admin = '1'";
        $result = mysqli_query($conn, $admin_check_query);
        while($row = mysqli_fetch_array($result)){

            $admin_id = $row['id'];
            $sql="INSERT INTO messages VALUES ('', '$admin_id', now(), 'Support Ticket - \'$subject\'', '$message --- $username', '1')";
            mysqli_query($conn,$sql);
        }

        $_SESSION['ticket_suc']=true;
        jump("index.php?page=support",5);
    }
}
?>
```

29. ábra: Support ticket elküldő kód (Saját forrás, 2023)

4.2.8 Játékmenetek

Ez a HTML oldal a játékmenetek listáját jeleníti meg. Minden játékmenet tartalmaz információkat, például a játék nevét, a játékosok számát és az alkalom állapotát. Az oldalon található egy keresősáv is az adott munkamenetek kereséséhez, valamint egy gomb az új munkamenet létrehozásához. Fontos megjegyezni, hogy egyelőre a keresősáv szimplán koncepció, nem működőképes. A kód csatlakozik egy MySQL adatbázishoz, és kiválasztja az összes játékalkalmat a „sessions” nevű táblából. Végig pörög az egyes találatokon, információkat kap a játékról és a játékmenetben résztvevő játékosokról az adatbázis más asztalairól, és kinyomtatja az információkat HTML-kártyaként. Ha a felhasználó be van jelentkezve, akkor ellenőrzi, hogy csatlakozott-e már egyes játékmenetekhez, vagy esetleg szervezett alkalmakat.


```

<?php
if (mysqli_connect_errno())
{ echo "Failed to connect to MySQL: " . mysqli_connect_error(); }
$result = mysqli_query($conn,"SELECT * FROM sessions");

while($row = mysqli_fetch_array($result))
{
    $game = mysqli_fetch_array(mysqli_query($conn,"SELECT * FROM games WHERE id = ".$row['game_id']));
    $players = mysqli_fetch_array(mysqli_query($conn,"SELECT COUNT(user_id) FROM users_sessions WHERE session_id = ".$row['id']));
    $sessions = mysqli_query($conn,"SELECT * FROM users_sessions WHERE session_id = ".$row['id']);
    $status = 2;

    $color = "black";
    if($players['COUNT(user_id)'] < $game['max_player_number']) {
        $color = "green";
    } else {
        $color = "red";
    }

    if(isset($_SESSION['login_id'])) {
        while($session = mysqli_fetch_array($sessions)) {
            if ($_SESSION['login_id'] == $session['user_id']) {
                if ($session['role'] == 0) {
                    $status = 0;
                } elseif ($session['role'] == 1) {
                    $status = 1;
                }
            }
        }
    }
}

print "

```

30. ábra: Játékmenetek lekérdező kódja (Saját forrás, 2023)

A 30. ábrán látható kód csatlakozik egy MySQL adatbázishoz, és három táblából kér információkat: „sessions”, „games” és „users_sessions”. Ezt az információt használja fel az egyes játékmenetek állapotának megjelenítésére, amelyet a játékmenetben részt vevő játékosok száma és az aktuális felhasználó (ha van) szerepe határoz meg az adott játékmenetben. A kód először a `mysqli_connect_errno()` függvény segítségével ellenőrzi az adatbázishoz való kapcsolódást, és hibaüzenetet jelenít meg, ha kapcsolódási hiba van. Ezután a `mysqli_query()` függvény segítségével kiválasztja az összes sort a „sessions” táblából, és az eredményt a `$result` változóban tárolja. A kód ezután egy `while` ciklusba lép, amely a `mysqli_fetch_array()` függvény segítségével lekéri a `$result` változó minden sorát. A cikluson belül a kód lekéri az aktuális munkamenet játékinformációit a „games” táblából a megfelelő `game_id`-vel rendelkező sort kiválasztva. Ezenkívül lekéri a játékalkalomban részt vevő játékosok számát úgy, hogy kiválasztja a felhasználói azonosítók számát a „users_sessions” táblázatban a megfelelő `session_id`-vel. Végül kiválasztja az összes sort a „users_sessions” táblából a megfelelő `session_id` azonosítóval. A kód ezután a `$status` kezdeti értékét 2-re állítja, ami a nem csatlakozott állapotot jelenti. Ezenkívül a `$color` kezdeti értéket feketére állítja. A kód ezután ellenőrzi, hogy a játékmenetben részt vevő játékosok száma ke-

vesebb-e, mint a játékban engedélyezett játékosok maximális száma. Ha igen, akkor a \$color-t zöldre állítja, jelezve, hogy az alkalom nyitott és játékosokat fogad. Ellenkező esetben pirosra állítja, jelezve, hogy megtelt. Ha az aktuális felhasználó be van jelentkezve (amit a \$_SESSION['login_id'] változó jelenléte határoz meg), a kód egy másik while ciklusba lép be, amely lekéri az egyes sorokat a \$sessions változóból. Ezen a cikluson belül a kód ellenőrzi, hogy az aktuális felhasználói azonosító megegyezik-e az aktuális játékmenet user_id azonosítójával. Ha igen, akkor ellenőrzi a felhasználó szerepét. Ha a szerep 0, ami azt jelzi, hogy a felhasználó rendező, akkor a \$status-t 0-ra állítja, illetve, ha a szerep 1, akkor azt jelzi, hogy a felhasználó játékos, akkor 1-re állítja. Ezután a kód kártyát generál minden egyes játékmenethez. A kártya információkat jelenít meg a játékmenetről, beleértve a nevét, helyét, dátumát, leírását és a jelenleg regisztrált játékosok számát a megengedett játékosok maximális számához viszonyítva. Ha a felhasználó be van jelentkezve, a kártyán további információk és gombok is megjelennek, attól függően, hogy a felhasználó milyen állapotban van az alkalommal kapcsolatban. Ha a felhasználó a rendező, megjelenik egy törlés gomb, amivel törölheti az játékmenetet. Ha a felhasználó már csatlakozott az aktuális menethez, akkor megjelenik egy kilépés gomb, amivel elhagyhatja azt. Ha az megtelt, egy üzenetet fognak látni, amely jelzi, hogy megtelt, illetve, ha még van szabad hely, a csatlakozás gombbal tudnak jelentkezni. Ha nincsenek elérhető játékkalkalmat, a kód egy üzenetet jelenít meg, amely jelzi, hogy nincs aktuális találat.

```
print "
<div class='card' style='flex-direction: row; height: auto; background-color: whitesmoke;'>
  <img src='"; echo $game['icon']; print "' class='card-img-top' id='card_image' style='min-height: 175px; min-width: 175px; max-height: 175px; max-width: 175px; object-fit: fit; margin-top: auto;
margin-bottom: auto; padding: 6px;'>
  <div class='card-body'>
    <h5 class='card-title'><strong>"; echo $row['name']; print " <i class='bi bi-people-fill' style='color: "; echo $color; print "></i> "; echo $players['COUNT(user_id)']; print "/"; echo $game
['max_player_number']; print "</strong></h5>
    <h6 class='card-subtitle mb-2 text-muted'>"; echo $game['name']; print "</h6>
    <h6 class='card-subtitle mb-2 text-muted'>"; echo $row['location']; echo ' '; echo $row['date']; print "</h6>
    <p class='card-text text-break'>"; echo $row['description']; print "</p>
  </div>
";
```

31. ábra: Kártyageneráló kód (Saját forrás, 2023)

Játékmenet létrehozásához egy gombbal tudnak eljutni a szükséges űrlaphoz a felhasználók. Csakis regisztrált felhasználók tudnak létrehozni alkalmakat. A kód ezért egy if feltétellel kezdődik, amely ellenőrzi, hogy a munkamenetet egy bejelentkezett felhasználó hozta-e létre vagy sem. Ha nem, akkor átirányítja a felhasználót a bejelentkezési oldalra. Ellenkező esetben megjeleníti az űrlapot egy új játékmenet létrehozásához. Az űrlap létrehozásához a HTML-kód be van ágyazva a PHP-kódba. Különféle beviteli mezőkből áll, mint például az alkalom neve, a játék, hely, dátum, valamint a leírás. A „games” mező egy legördülő lista, amely dinamikusan töltődik

fel az adatbázis „games” táblájából. A „Create Session” gomb elküldi az űrlapot, és új játékal-
kalmat hoz létre. Az űrlapnak számos attribútuma van, például az enctype, a method és az auto-
complete, amelyek az űrlap viselkedésének meghatározására és biztonságosabbá tételére szolgál-
nak.

```
<?php
if(isset($_SESSION['login_id'])){
    if(isset($_POST)){
        $session_id = $_POST['join_button'];

        $mainuser_query = "SELECT * FROM users_sessions WHERE session_id = '$session_id' and role = '0'";
        $result = mysqli_query($conn, $mainuser_query);
        while($row = mysqli_fetch_array($result))
        { $main_user = $row['user_id']; }

        $session_name_query = "SELECT * FROM sessions WHERE id = '$session_id'";
        $result = mysqli_query($conn, $session_name_query);
        while($row = mysqli_fetch_array($result))
        { $session_name = $row['name']; }

        $user_id = $_SESSION['login_id'];
        $username = $_SESSION['login_new'];
        $session_check_query = "SELECT * FROM users_sessions WHERE user_id = '$user_id' and session_id = '$session_id' LIMIT 1";
        $result = mysqli_query($conn, $session_check_query);
        $session = mysqli_fetch_assoc($result);
        $relocate = $_POST['relocate'];

        if ($session) {
            $_SESSION['join_error']=true;
            if($relocate == 'party'){
                jump("index.php?page=partyhome",5);
            } elseif ($relocate == 'profile') {
                jump("index.php?page=profile",5);
            } else {
                jump("index.php?page=home",5);
            }
        } else {
            $sql="INSERT INTO users_sessions VALUES ('$session_id', '$user_id', '1')";
            mysqli_query($conn,$sql);
            $sql="INSERT INTO messages VALUES ('', '$main_user', now(), 'Party Finder', '$username has joined your Session called \"'$session_name\"!', '1')";
            mysqli_query($conn,$sql);
        }

        $_SESSION['join_suc']=true;
        if($relocate == 'party'){
            jump("index.php?page=partyhome",5);
        } elseif ($relocate == 'profile') {
            jump("index.php?page=profile",5);
        } else {
            jump("index.php?page=home",5);
        }
    }
}
```

32. ábra: Csatlakozás játékmenehez kód (Saját forrás, 2023)

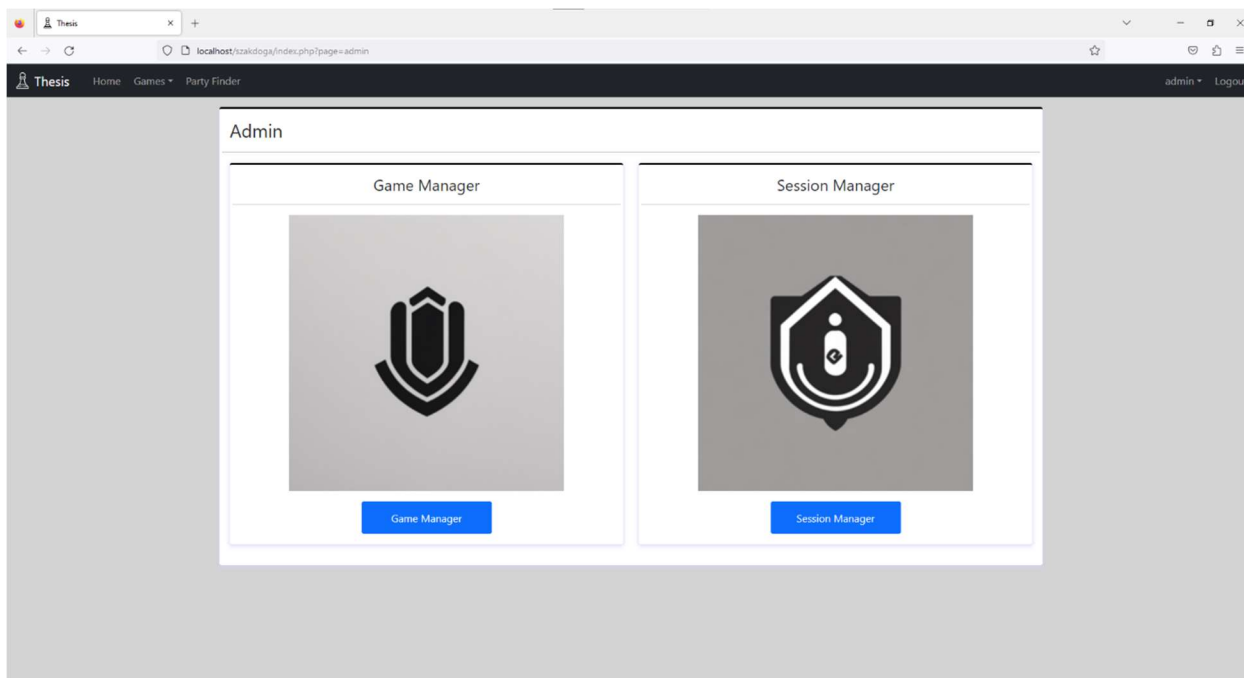
Játékmenehez való csatlakozás a következőképpen működik. Egy PHP-szkript ellenőrzi, hogy a felhasználó be van-e jelentkezve, és ha igen, akkor feldolgozza a POST metódussal elküldött űrlapadatokat. A szkript ezután a következő műveleteket hajtja végre:

1. Lekéri a játékmene-azonosítót és a rendező azonosítót az űrlapadatokból.
2. Lekéri a játékmene-azonosítóhoz társított játékmene nevét.

3. Ellenőrzi, hogy a felhasználó részt vesz-e már a játékban. Ha nem, akkor hozzáadja a felhasználót és üzenetet küld a szervezőnek, jelezve, hogy a felhasználó csatlakozott a játékalkalomhoz.
4. Beállít egy munkamenet-változót, amely jelzi, hogy az összekapcsolás sikeres volt-e vagy sem.
5. Végül átirányítja a felhasználót egy megadott oldalra, miután a csatlakozási folyamat befejeződött.

4.2.9 Admin felület

Az adminisztrációs oldal a webhely biztonságos és korlátozott része, amelyhez csak az arra jogosult adminisztrátorok férhetnek hozzá. Célja, hogy egy központi helyet biztosítson, ahol az adminisztrátor felhasználók kezelhetik a webhely tartalmát és beállításait. Az admin oldal két fő rész van: „Game Manager” és „Session Manager”. A játékkezelő rész lehetővé teszi az adminisztrátor felhasználók számára, hogy kezeljék a webhelyen elérhető játékokat. Ez magában foglalja a meglévő játékok szerkesztését, újak hozzáadását és a már nem szükséges játékok törlését. A játékmenetkezelő szakasz lehetővé teszi az adminisztrátor felhasználók számára a felhasználók által létrehozott aktív játékalkalmak kezelését. Az adminisztrátor felhasználók törölhetik azokat az alkalmakat, amelyekre már nincs szükség, vagy amelyek sértik a webhely irányelveit. Fontos megjegyezni, hogy az adminisztrátor felhasználók nem törölhetik a felhasználók által jelenleg használt játékokat a játékmenet kereső oldalon.



33. ábra: Admin felület (Saját forrás, 2023)

A játékkezelő oldal a játékok listáját jeleníti meg, és lehetővé teszi az admin felhasználó számára, hogy válasszon egy játékot, hogy megtekinthesse annak részleteit és módosítsa adatait. A kód a MySQL segítségével lekérdezi az adatbázist a játékinformációkért, és HTML és Bootstrap használatával megjeleníti az oldalon. A kód első része az adatbázisban található összes játék legördülő menüjét jeleníti meg, amelyet egy MySQL lekérdezés tölt fel. Ha a felhasználó kiválaszt egy játékot, az lekéri annak információit az adatbázisból, és egy kártyaelemben jeleníti meg a játék nevével, maximális játékoszámmal és leírásával. A kártyaelem alatt a kód egy űrlapot jelenít meg a játék profiljának szerkesztéséhez. Ez tartalmazza a játék nevének, kategóriájának, maximális játékoszámnak, leírásának és képének mezőit. A kép egy külön kártyaelemben jelenik meg az űrlap bal oldalán. Itt van lehetőségünk tovább menni a játék hozzáadó menüre, ahol a kód egy űrlapot határoz meg a játékok hozzáadásához. Az űrlap beviteli mezőket tartalmaz a játék nevéhez, a kategóriához, a játékosok maximális számához, a játék leírásához és a játék képéhez. Egy JavaScript kód felelős az „Add Game” gomb letiltásáért, amíg a legördülő listából ki nem választ egy kategóriát. A kód hozzáad egy eseményfigyelőt a legördülő listához. Amikor a legördülő lista értéke megváltozik, a kód ellenőrzi, hogy a kiválasztott érték alapértelmezett-e. Ha igen, akkor az „Add Game” gomb le lesz tiltva a „disabled” attribútum használatával. Ha más értéket választ ki a felhasználó, akkor a gomb engedélyezve lesz az attribútum eltávolításával.

```

<script>
const dropdown = document.getElementById("exampleFormControlSelect1");
const button = document.getElementById("add_button");

dropdown.addEventListener("change", function() {
  if (dropdown.value === "Default") {
    button.disabled = true;
  } else {
    button.disabled = false;
  }
});
</script>

```

34. ábra: Gomb letiltó JavaScript kód (Saját forrás, 2023)

A játék hozzáadásáért egy PHP kód a felelős, ami a következő lépéseket hajtja végre. Először ellenőrzi, hogy az űrlapot az `isset()` függvény segítségével küldték-e el. Ha igen, akkor a `$_POST` segítségével lekéri az értékeket az űrlapról, és hozzárendeli azokat a változókhoz. Ezután ellenőrzi, hogy létezik-e már azonos nevű játék az adatbázisban úgy, hogy a `mysqli_query()` függvény segítségével lekérdezi a játéktáblát, és az eredményt a `$result` változóban tárolja. Ha a lekérdezés nem ad vissza sorokat, akkor a játék hozzáadódik az adatbázishoz. Az `str_replace()` függvényt használja, hogy elkerülje a játék nevében és leírásában szereplő szimpla vagy dupla idézőjeleket. Ezután ellenőrzi, hogy a `$_FILES` szuperglobálison az `is_uploaded_file()` függvény segítségével feltöltött-e az admin egy képfájlt. Ha egy képfájlt feltöltöttek, az lekéri a fájl nevét, típusát, ideiglenes nevét és méretét a `$_FILES` szuperglobális fájlból, és áthelyezi a fájlt a feltöltési könyvtárba a `move_uploaded_file()` segítségével. Utána összeállít egy SQL INSERT lekérdezést a megfelelő értékekkel, és végrehajtja a `mysqli_query()` segítségével. Sikerüzenetet ad ki, ha a játék felkerült az adatbázisba, és rövid késleltetés után a `jump()` függvény segítségével átirányítja a felhasználót az adminisztrációs oldalra. Ha már létezik azonos nevű játék az adatbázisban, hibaüzenetet ad ki, és a `jump()` függvény segítségével átirányítja a felhasználót az adminisztrációs oldalra.

```

<?php
if(isset($_POST))
{
    $gamenname = $_POST['gamenname'];
    $maxplayers = $_POST['maxplayers'];
    $description = $_POST['description'];
    $category = $_POST['category'];

    $result=mysqli_query($conn,"SELECT * FROM games WHERE name='$gamenname'");
    $row=mysqli_fetch_assoc($result);
    if(mysqli_num_rows($result)==0) {
        $fr_desc = str_replace("'", "\'", $description);
        $fr_desc = str_replace("'", "\'", $fr_desc);
        $fr_name = str_replace("'", "\'", $gamenname);
        $fr_name = str_replace("'", "\'", $fr_name);

        if(is_uploaded_file($_FILES['gameimage']['tmp_name']))
        {
            // file name, type, size, temporary name
            $file_name = $_FILES['gameimage']['name'];
            $file_type = $_FILES['gameimage']['type'];
            $file_tmp_name = $_FILES['gameimage']['tmp_name'];
            $file_size = $_FILES['gameimage']['size'];
            $gameid = $_POST['add_button'];

            // target directory
            $target_dir = "uploads";

            // uploading file
            if(move_uploaded_file($file_tmp_name,$target_dir."/".$file_name))
            {
                $sql="INSERT INTO games VALUES ('', '$fr_name', '$fr_desc', '$maxplayers', '$category', '$target_dir/$file_name')";
            }
        } else {
            $sql="INSERT INTO games VALUES ('', '$fr_name', '$fr_desc', '$maxplayers', '$category', 'uploads/placeholder.png')";
        }
        mysqli_query($conn,$sql);

        echo "
        <div class='col-lg-4 m-auto rounded-top' style='background-color:rgb(234, 255, 240);'>
        <p class='text-center' id='success'><b>Game has been successfully added!</b></p>
        <p class='text-center text-muted lead' style='font-size:80%;padding-bottom: 5px;' >Redirecting...</p>
        </div>";
        renew_session($conn);
        jump("index.php?page=admin",2000);

    } else {
        echo "
        <div class='col-lg-4 m-auto rounded-top' style='background-color:rgb(255, 234, 234);'>
        <p class='text-center' id='failure'><b>Failed to add Game!</b></p>
        <p class='text-center text-muted lead' style='font-size:80%;padding-bottom: 5px;' >Redirecting...</p>
        </div>";
        jump("index.php?page=admin",2000);
    }
} else {
    print "How?";
}
?>

```

35. ábra: Játék hozzáadásának kódja (Saját forrás, 2023)

A munkamenet-kezelő egy PHP-szkript, amely egy webalkalmazás munkameneteit kezeli. Funkciókat biztosít a munkamenetekhez való csatlakozáshoz, kilépéshez és törléshez. A szkript lekéri az egyes munkamenetekre vonatkozó információkat egy MySQL adatbázisból, beleértve a nevet, helyet, dátumot és leírást. Ezenkívül információkat kér a kapcsolódó játékról és a játékmenetben éppen részt vevő játékosok számáról. A szkript a játékmeneteket kártyaként jeleníti meg egy

weboldalon. Minden kártya tartalmazza a játék képét, az alkalom nevét, a játékosok számát, a játék nevét, az alkalom helyét és dátumát, valamint a játékmenet leírását. A munkamenet neve melletti emberek ikon színe jelzi, hogy a munkamenet nyitott (zöld) vagy teljes (piros). Itt az adminisztrátor felhasználóknak további gomb jelenik meg, amivel kitörölheti az egyes játékmeneteket.

Session Manager

	Uno Party  0/10 Uno Budapest 2023-03-14 18:00:00 Come over for a game of Uno!	Available Delete
	Game Night  0/4 Pandemic Nagykúta 2023-04-07 15:00:00 Let's try to beat this game together!	Available Delete
	Kitten Chaos  0/5 Exploding Kittens Budapest 2023-04-21 21:00:00 Food & drinks included	Available Delete

36. ábra: Játékalcalomkezelő admin felület (Saját forrás, 2023)

5. Eredmények, következtetések, javaslatok

A webalkalmazást úgy alakítottam ki, hogy felhasználóbarát platformot biztosítson a játékrájongóknak a különböző játékkategóriákhoz tartozó játékmenetek megtalálásához, valamint létrehozásához. A webhely egy üdvözlő kezdőlappal rendelkezik, amely áttekintést nyújt a webhely szolgáltatásairól és funkcióiról, valamint példákat mutat be a platformon keresztül játszható játékokra. A különböző játékkategóriák részletes leírása külön-külön oldalakon található, az egyes kategóriákhoz tartozó rendelkezésre álló játékok listája mellett. A játékmenet kereső lehetővé teszi a regisztrált felhasználók számára, hogy megtekintsék az aktuális és aktív játékmeneteket, beleértve az olyan információkat, mint annak neve, a játék neve, helye, időpontja és dátuma, valamint leírása. A regisztrált felhasználók saját játékalkalmakat is létrehozhatnak, amelyekhez más felhasználók is csatlakozhatnak. Ezenkívül a webhely tartalmaz egy üzenetküldő és support jegyrendszert, amely lehetővé teszi a felhasználók számára, hogy kommunikáljanak egymással, és jelentsenek minden felmerülő problémát a webhely adminisztrátorainak. Mivel a webhely egy új platform, jelenleg nincs aktív felhasználó vagy játékmenet. A webhely kialakítása és funkcionalitása alapján azonban megvan a lehetőség arra, hogy sokféle felhasználót vonzzon, és egy játékrájongó közösséget hozzon létre. A weboldal felhasználóbarát felülete, valamint a játékkategóriák és játékok részletes leírása a kezdő és a tapasztalt játékosok számára egyaránt vonzó lehet. A játékmenet-kereső és az üzenetküldő rendszer egyszerű és hatékony módot is nyújthat a felhasználók számára, hogy kapcsolatba léphessenek egymással és megszervezzék a játékmeneteket. A webhely kialakítását és funkcionalitását felhasználói élmény (UX) teszteléssel teszteltem, ami annyit jelent, hogy hozzám közel álló, de a weblappal teljesen ismeretlen személyeket kértem meg, hogy konkrét feladatok végezzenek el a webhelyen, miközben visszajelzést adnak tapasztalataikról. Az UX-tesztelés során gyűjtött visszajelzéseket a webhely változtatására és fejlesztésére használtam fel, ami egy intuitívabb és felhasználóbarátabb platformot eredményezett.

Egy új weboldal egyik legkritikusabb szempontja a marketing és promóció. A webhely közösségi média platformokat és fórumokat használhatna fel a társas-, kártya- és más típusú játékokhoz a platform népszerűsítésére. A közösségi médiában nagy követőkkel rendelkező influencerekkel és játékrájongókkal való együttműködés szintén segíthetne népszerűsítésében.

Bár a webhely rendelkezik egy keresővel, amely lehetővé tenné a felhasználók számára az aktív játékok közötti specifikus és pontosabb keresést, a funkció hozzáadása nem valósult meg. Ez lehetővé tenné a felhasználók számára, hogy meghatározott, konkrét játékokat vagy játékkategóriákat keressenek, megkönnyítve számukra a releváns játékok megtalálását. A keresési funkció úgy valósítható meg, hogy a játékmenet-kereső oldalon lévő keresősávot módosítani kéne, hogy lehetővé tegye a felhasználók számára, hogy meghatározott kulcsszavak alapján keressenek játékokat. A funkció megvalósításával a felhasználók könnyen megtalálhatják az őket érdeklő alkalmakat, ami növeli az elköteleződést és az elégedettséget.

A felhasználók elégedettségi és értékelési rendszere a játékmenetekhez és a játékokhoz egyaránt segíthetne a felhasználóknak megalapozott döntések meghozatalában, mielőtt csatlakoznának az adott alkalomhoz. Ez segíthetne a népszerű játékok és alkalmak azonosításában, és segíthet a felhasználóknak megtalálni a hasonló játékokat, amelyek érdekelhetik őket. A felhasználók értékelhetik azokat a játékmeneteket és játékokat, amelyekben részt vettek, a webhely pedig megjelenítheti az értékeléseket és véleményeket, hogy segítsen másoknak megtalálni a legjobb számukra legmegfelelőbb alkalmakat, valamint játékokat. Ennek a funkciónak a megvalósításával a webhely megbízható platformmá válhatna a játékrajongók számára.

A felhasználók elköteleződésének és megtartásának növelése érdekében bevezethetnék gamifikációs elemeket, például a ranglistákat, jelvényeket vagy jutalmakat. A gamifikáció a játéktervezési elvek és mechanikák alkalmazása nem játékkontextusokban. Ez azt jelenti, hogy olyan elemeket kell venni, amelyek szórakoztatóvá teszik a játékokat, és azokat olyan tevékenységekre kell alkalmazni, amelyek általában nem minősülnek játéknak, mint például az oktatás, a munka és az egészségügy. A felhasználók például pontokat szerezhetnek a játékmenetekben való részvételért vagy azok létrehozásáért, amelyeket exkluzív funkciók feloldására használhatnak fel. Ezen elemek bevezetésével a weboldal sikerélményt és versenyt kelthetne, ami növelné a felhasználók elköteleződését és elégedettségét.

6. Összefoglalás

Összefoglalva, ez a webalkalmazás azt a problémát kívánja megoldani, hogy az emberek nehezen találjanak másokat, akikkel konkrét játékokat játszhatnak. Az eredmény egy átfogó webhely, amely lehetővé teszi a felhasználók számára, hogy különféle játékkategóriákhoz játékmeneteket találjanak és hozzanak létre. A weboldal létrehozásával olyan megoldást szeretnék kínálni, amely potenciálisan segíthet sok játékos számára, hogy csatlakozzanak és játsszanak egymással, függetlenül a tartózkodási helyüktől. A felhasználói visszajelzések beépítésével és felhasználói tesztek elvégzésével a weboldalt úgy alakítottam ki, hogy felhasználóbarát legyen és megfeleljen a célközönség igényeinek. Összességében ez téma kiemeli a technológia erejét a valós világ problémáinak megoldásában és az emberek életének javításában. Azáltal, hogy platformot biztosít a játékosok számára a kapcsolatteremtéshez és a közös játékhoz, a webhely képes előmozdítani a közösségi érzést, és összehozza az embereket a közös játékszeret révén.

Számomra ez a téma azért fontos, mert lelkes társasjáték-rajongó vagyok, akinek ritkán jut ideje bonyolult játékokat játszani másokkal. Észrevettem, hogy sokan óvakodnak a nehéz játékok eljáratásától, ezért nehéz olyan embereket találni, akikkel megvan a közös összhang. Ez arra késztetett, hogy elgondolkozzam azon, hogy a technológia miként segíthet a hasonló játékerdeklődésű és -preferenciákkal rendelkező emberek összekapcsolásában. Ahogy tovább vizsgáltam ezt az ötletet, rájöttem, hogy a webfejlesztés lehet a kulcs a probléma megoldására. A webfejlesztést izgalmas, kreatív és kielégítő területnek tartom, és szerettem volna tudásomat kamatoztatni egy olyan megoldás létrehozásában, amely segíthet másoknak, mint én. Diplomamunkám során a társasjátékok és a webfejlesztés iránti szenvedélyemet egyesítettem egy olyan weboldal létrehozására, amely lehetővé teszi a felhasználók számára, hogy különböző játékkategóriákhoz játékmeneteket találjanak és készítsenek. Úgy gondolom, hogy ez a megoldás nemcsak nekem, hanem más játékosoknak is hasznára válik, akik hasonló küzdelmekkel küzdenek. Azzal, hogy az érdeklődési köreimet folytattam, és azokat felhasználva olyan megoldást hoztam létre, amely egy valós problémát kezel, megpróbáltam bemutatni a szenvedély és a kreativitás potenciálját. Remélem, hogy diplomamunkám másokat is arra ösztönöz, hogy felfedezzék érdeklődési körük és az őket érdeklő problémák metszetét, és a technológia segítségével pozitív hatást gyakoroljanak a világra.

Irodalomjegyzék

Visual Studio Code (2023): VSCode Documentation, URL: <https://code.visualstudio.com>

Bootstrap (2023): Bootstrap Documentation, URL: <https://getbootstrap.com>

Javatpoint (2021): Learn JavaScript, URL: <https://www.javatpoint.com>

OpenBookProject (2022): Get Down with CSS, URL: <http://www.openbookproject.net/tutorials/getdown/css/>

Brett McLaughlin (2012): PHP & MySQL: The Missing Manual, 2nd Edition, O'Reilly Media Inc., ISBN 978-144-9355-54-8, URL: <https://www.oreilly.com/library/view/php-mysql/9781449355517/>

Pohl L. (2010): A programozás alapjai, Budapest, URL: https://www.eet.bme.hu/~pohl/h_jegyzet.pdf

Nagy G. (2011): Web programozás alapismeretek, Budapest, Ad Librum, ISBN 978-615-5110-26-9, URL: <https://nagygusztav.hu/web-programozas>

Timothy Berners-Lee. Fielding R. (1996): Hypertext Transfer Protocol, W3C, URL: <http://tools.ietf.org/html/rfc1945>

PHP DOCUMENTATION GROUP (2012): PHP Manual, PHP Documentation Group, URL: <https://www.php.net/docs.php>

7. Ábrajegyzék

1. ábra: A Visual Studio Code felülete (VisualStudio, 2023)	7
2. ábra: A World Wide Web architektúrája (JavaPoint).....	8
3. ábra: Hypertext Transfer Protocol működése (JavaPoint).....	10
4. ábra: CSS doboz modell (OpenBookProject).....	12
5. ábra: PHP működése (McLaughlin, 2012)	14
6. ábra: Bootstrap előnyei (Sitesbay).....	16
7. ábra: Példa adatbázis felépítésére, kapcsolataira (Thong Do, 2022).....	17
8. ábra: Játékkalkalom létrehozásának folyamata (Saját forrás, 2023)	21
9. ábra: Játékkalkalom létrehozásának felülete (Saját forrás, 2023)	22
10. ábra: Játékkalkalmak keresésének felülete (Saját forrás, 2023)	23
11. ábra: A webalkalmazás adatbázisának felépítése (Saját forrás, 2023)	24
12. ábra: Tartalom betöltés kódrészlet 1 (Saját forrás, 2023)	26
13. ábra: Tartalom betöltés kódrészlet 2 (Saját forrás, 2023)	26
14. ábra: A jump függvény kódja (Saját forrás, 2023).....	27
15. ábra: A kapcsolat létrehozó kód (Saját forrás, 2023).....	28
16. ábra: Kezdőlap kódrészlet 1 (Saját forrás, 2023)	28
17. ábra: Kezdőlap felülete (Saját forrás, 2023)	29
18. ábra: Legördülő menü (Saját forrás, 2023)	30
19. ábra: Kártyajátékok leírásának felülete (Saját forrás, 2023).....	31
20. ábra: Kártyajátékok lekérdezésének kódja (Saját forrás, 2023)	31
21. ábra: Bejelentkezési űrlap (Saját forrás, 2023).....	32
22. ábra: Bejelentkezés backend kódja (Saját forrás, 2023).....	33
23. ábra: Regisztráció folyamata (Saját forrás, 2023).....	34
24. ábra: Regisztráció kódja (Saját forrás, 2023)	35
25. ábra: Profilképfeltöltés kódja (Saját forrás, 2023)	36
26. ábra: Profilszerkesztési űrlap (Saját forrás, 2023)	37
27. ábra: Összes olvasatlan üzenet olvasottként jelölő kód (Saját forrás, 2023)	38
28. ábra: Üzenetek oldal felülete (Saját forrás, 2023)	39
29. ábra: Support ticket elküldő kód (Saját forrás, 2023)	40
30. ábra: Játékmenetek lekérdező kódja (Saját forrás, 2023)	41
31. ábra: Kártyageneráló kód (Saját forrás, 2023)	42
32. ábra: Csatlakozás játékmenethez kód (Saját forrás, 2023).....	43
33. ábra: Admin felület (Saját forrás, 2023).....	45
34. ábra: Gomb letiltó JavaScript kód (Saját forrás, 2023)	46
35. ábra: Játék hozzáadásának kódja (Saját forrás, 2023)	47
36. ábra: Játékkalkalomkezelő admin felület (Saját forrás, 2023).....	48

8. Nyilatkozatok

Hallgatói eredetiségi nyilatkozat

NYILATKOZAT

a szakdolgozat nyilvános hozzáféréséről és eredetiségéről

A hallgató neve: Erdős Roland
A Hallgató Neptun kódja: VUC48T
A dolgozat címe: Játékkalkalom tervező és rendező közösségi platform társasjáték-
rajongók számára
A megjelenés éve: 2023
A konzulens tanszék neve: Alkalmazott Informatikai Tanszék

Kijelentem, hogy az általam benyújtott szakdolgozat egyéni, eredeti jellegű, saját szellemi alkotásom. Azon részeket, melyeket más szerzők munkájából vettem át, egyértelműen megjelöltem, s az irodalomjegyzékben szerepeltettem.

Ha a fenti nyilatkozattal valótlan állítottam, tudomásul veszem, hogy a Záróvizsga-bizottság a záróvizsgából kizár és a záróvizsgát csak új dolgozat készítése után tehetek.

A leadott dolgozat, mely PDF dokumentum, szerkesztését nem, megtekintését és nyomtatását engedélyezem.

Tudomásul veszem, hogy az általam készített dolgozatra, mint szellemi alkotás felhasználására, hasznosítására a Magyar Agrár- és Élettudományi Egyetem mindenkor szellemi tulajdon-kezelési szabályzatában megfogalmazottak érvényesek.

Tudomásul veszem, hogy dolgozatom elektronikus változata feltöltésre kerül a Magyar Agrár- és Élettudományi Egyetem könyvtári repozitóri rendszerébe.

Kelt: Nagybánya 2023 év 05 hó 02 nap

Erdős Roland

Hallgató aláírása

Konzultációs nyilatkozat

KONZULTÁCIÓS NYILATKOZAT

Az Erdős Roland (név) (hallgató Neptun azonosítója: VUC48T) konzulenseként nyilatkozom arról, hogy a szakdolgozatot áttekintettem, a hallgatót az irodalmi források korrekt kezelésének követelményeiről, jogi és etikai szabályairól tájékoztattam.

A szakdolgozatot a záróvizsgán történő védeésre javaslom / nem javaslom¹.

A dolgozat állam- vagy szolgálati titkot tartalmaz: igen nem^{*2}

Kelt: Gyöngyös 2023. év 05. hó 02. nap


Belső konzulens

9. Melléklet

Melléklet 1: Az elkészített webalkalmazás, illetve szükséges adatbázis

Erdos_Roland_VUC48T_2023_B-GYO-N-HU-GAZDI_Melleklet.rar

Amely tartalma:

1. Az „Adatbazis” nevű mappa tartalmazza a webalkalmazáshoz szükséges exportált adatbázist.
2. A „Weblalkalmazas” nevű mappa tartalmazza magának a weblapnak kódját és hozzátartozó fájlok.