BGSZC Logisztikai és Kereskedelmi Technikum és Szakképző Iskola

1087 Budapest, Szörény utca 2-4.

1. Záró dolgozat

MTG CardShop

Konzulens tanár: Készítették:

Boros Sándor Masznyik Márton, Berényi Benedek

Tartalom

[1 Bevezetés 3](#_Toc85723173)

[1.1 Feladat leírás 3](#_Toc85723174)

[1.2 A felhasznált ismeretek 3](#_Toc85723175)

[1.3 A felhasznált szoftverek 3](#_Toc85723176)

[2 Felhasználói dokumentáció 4](#_Toc85723177)

[2.1 A program általános specifikációja 4](#_Toc85723178)

[2.2 Rendszerkövetelmények 4](#_Toc85723179)

[2.2.1 Hardver követelmények 4](#_Toc85723180)

[2.2.2 Szoftver követelmények 4](#_Toc85723181)

[2.3 3. A program telepítése 4](#_Toc85723182)

[2.4 A program használatának a részletes leírása 5](#_Toc85723183)

[3 Fejlesztői dokumentáció 6](#_Toc85723184)

[3.1 Az alkalmazott fejlesztői eszközök 6](#_Toc85723185)

[3.2 Adatmodell leírása 6](#_Toc85723186)

[3.3 Részletes feladatspecifikáció, algoritmusok 6](#_Toc85723187)

[3.4 Tesztelési dokumentáció 7](#_Toc85723188)

[4 Összefoglalás 8](#_Toc85723189)

[4.1 Önértékelés 8](#_Toc85723190)

[4.2 Továbbfejlesztési lehetőségek 8](#_Toc85723191)

[5 Felhasznált irodalom 9](#_Toc85723192)

[6 Ábrajegyzék 10](#_Toc85723193)

# Bevezetés

## Feladat leírás

A szakmai érettségi projektünk célja egy olyan komplex alkalmazás fejlesztése volt, amely egy Magic: The Gathering (MTG) témájú kártyabolt működését támogatja, mind webes, mind asztali alkalmazás formájában. A projekt címe: MTG CardShop.

A Magic: The Gathering egy világszerte ismert és elismert gyűjtögetős kártyajáték, amely 1993 óta van jelen a piacon, és az évtizedek alatt számos különféle kiadással és játékmóddal bővült. A játékosok egyedi vagy előre elkészített paklikkal, különféle versenyformátumokban mérhetik össze tudásukat. Az MTG világában kártyák ezrei léteznek, amelyek értéke jelentősen változhat, attól függően, hogy milyen gyakoriak, mennyire erősek, vagy éppen mennyire keresettek a játékosközösségben.

Projektünk célja, hogy egy könnyen kezelhető, naprakész információkat biztosító, korszerű felhasználói felülettel rendelkező alkalmazást hozzunk létre, amely lehetőséget nyújt:

Termékek böngészésére, vásárlására és eladására

A készlet és árak kezelésére adminisztrátori oldalon

Az új kiadások naprakész nyomon követésére

A bolt háttérrendszerének átlátható és hatékony menedzselésére

A program használatához, telepítéséhez és beüzemeléséhez részletes felhasználói dokumentáció készült. Ennek célja, hogy segítséget nyújtson minden leendő felhasználónak a rendszer használatának elsajátításában, valamint megkönnyítse a telepítési és működtetési folyamatokat. A dokumentáció lépésről lépésre bemutatja a rendszer funkcióit, lehetőségeit, és választ ad a felmerülő gyakorlati kérdésekre is.

* A projekt két fő komponensből épül fel:
* Webes alkalmazás

A felhasználók ezen a felületen tudnak:

* böngészni a MTG hivatalos termékei között
* Termékeket rendelni és eseményekre jelentkezni
* értesülni az új eseményekről és esetleges árváltozásokról

A webalkalmazás frontendje modern technológiákkal (React, HTML, CSS, JavaScript) készült, a backend Node.js és Express alapokra épül. A webes felület a felhasználói élményt helyezi előtérbe, gyors betöltéssel, reszponzív designnal és logikus navigációval.

Asztali alkalmazás

Ez a rész a bolt adminisztratív tevékenységeit támogatja:

* készletkezelés (új termekek felvétele, meglévők módosítása, törlése)
* Esemenyek nyomon követése
* statisztikák és forgalmi adatok megtekintése

Az asztali alkalmazást C# nyelven, WinForms keretrendszerben fejlesztettük, amely stabil és egyszerű megoldást biztosít a helyi ügyintézéshez.

Miért ezt a feladatot választottuk?

A választásunk nem véletlen: csapattagként mindketten hosszú évek óta rajongói vagyunk az MTG-nek, így a téma nemcsak szakmai kihívást, hanem valódi érdeklődést is jelentett számunkra. Célunk volt egy olyan rendszert készíteni, amely a való életben is működőképes lenne egy MTG-specializált kártyabolt vagy magán személy számára, és amelyet akár tovább is lehetne fejleszteni egy teljes értékű, nyílt kereskedelmi platformmá.

Ezzel a projekttel nemcsak a programozási tudásunkat, hanem a csapatmunkában, verziókezelésben, adatbázis-tervezésben és felhasználói felület kialakításban szerzett ismereteinket is gyakorlatba ültethettük. Fontos volt számunkra, hogy olyan dokumentáció készüljön hozzá, amely világosan bemutatja a működést, és minden szinten támogatja a felhasználókat a rendszer megismerésében.

## A felhasznált ismeretek

A projekt elkészítése során több olyan tantárgyi és gyakorlati ismeretre is szükség volt, amelyeket az iskolai tanulmányok, valamint önálló tanulás során szereztünk meg. A program elkészítésében fontos szerepet játszott az **asztali alkalmazás fejlesztése**, a **backend és frontend programozás**, illetve az **adatbázis-kezelés** ismeretei. Ezek közül több témával már találkoztunk a korábbi gyakorlati feladatok során is.

A projekt technológiai megvalósításához használt nyelvek és keretrendszerek ismerete elengedhetetlen volt. Ezek közé tartozik:

* **C#** – az asztali alkalmazás elkészítéséhez
* **JavaScript, HTML, CSS** – a webes felülethez
* **React** – a webes frontend modern, komponens-alapú kialakításához
* **Node.js és Express** – a szerveroldali működéshez és REST API-k létrehozásához
* **MySQL** – az adatbázis kezeléséhez, a kapcsolatok és lekérdezések kialakításához
* **Bootstrap** – a reszponzív, letisztult megjelenés megvalósításához

Ezeket az ismereteket **részben az iskolai oktatás során**, részben pedig **különböző online forrásokon** keresztül sajátítottuk el, például dokumentációk, oktatóvideók és fejlesztői fórumok segítségével.

A fejlesztés során **önállóan sajátítottuk el** a **GitHub** és **Git** használatát, melyek nélkülözhetetlenek voltak a közös munka során. Továbbá fontos készség volt a **szakmai dokumentációk elkészítése**, amelyben szintén fejlődtünk a projekt alatt.

Külön figyelmet kellett fordítani az **Express** keretrendszer és a **MySQL** adatbázis-motor működésének mélyebb megértésére is, mivel ezek biztosították a backend és az adatkezelés alapját.

A csapatmunka során használtunk **GitHubot**, ahol több mint **150 commit** történt a fejlesztési időszak alatt. A hatékony együttműködés érdekében napi szintű kommunikáció zajlott **Microsoft Teamsen** és **Discordon**, amely segítette a projekt gördülékeny haladását, valamint a feladatok pontos elosztását.

## A felhasznált szoftverek

A fejlesztés során többféle szoftverkörnyezetet és eszközt alkalmaztunk annak érdekében, hogy hatékonyan tudjunk dolgozni a projekt különböző részein. A rendszer felépítése miatt mind **webes**, mind **asztali környezetekben** dolgoztunk, így szükség volt több platform és technológia összehangolt használatára.

A projekt fejlesztése **elsősorban Windows operációs rendszeren** zajlott, azonban előfordult, hogy nem állt rendelkezésre PC-s környezet, így bizonyos backend munkákat **MacOS rendszeren** végeztünk el. Ez különösen hasznos volt a platformfüggetlen fejlesztési szemlélet kialakításában.

A konkrét fejlesztői környezetek és eszközök a következők voltak:

* **Visual Studio (C# WinForms)** – az asztali alkalmazás fejlesztéséhez
* **Visual Studio Code** – a webes alkalmazás frontend és backend részének fejlesztéséhez (React, Node.js, Express)
* **XAMPP** – a helyi szerver környezet biztosításához és a MySQL adatbázis futtatásához
* **phpMyAdmin** – az adatbázis kezelőfelületének használatához, amely segítséget nyújtott a táblák, kapcsolatok és lekérdezések vizuális kezelésében

Az adatbázis-kezeléshez **MySQL-t** használtunk, amely a **XAMPP környezet részeként** biztosította a szükséges szerveres háttért. Az adatbázist phpMyAdmin segítségével adminisztráltuk, így gyorsan és átláthatóan tudtuk módosítani a szerkezetet, illetve tesztelni a lekérdezéseket.

A projekt verziókezeléséhez a **GitHub** szolgáltatását vettük igénybe. A Git-es munkafolyamat lehetővé tette, hogy a csapat minden tagja nyomon kövesse a változtatásokat, hibajavításokat és fejlesztéseket. A GitHub lehetőséget biztosított a párhuzamos munkavégzésre is, és több mint **150 commit** készült a fejlesztés során, ami jól mutatja az aktív csapatmunkát.

Ezek a szoftverek együttesen biztosították, hogy a fejlesztés folyamatos, stabil és jól szervezett legyen, valamint lehetőséget adtak a rendszer működésének gyors tesztelésére és finomhangolására.

# Felhasználói dokumentáció

## A program általános specifikációja

Az MTG CardShop egy többplatformos alkalmazás, amely a Magic: The Gathering (MTG) kártyajáték tematikájára épülő kártyabolt működését digitalizálja. A program célja, hogy modern, felhasználóbarát felületet biztosítson a kártyák böngészéséhez, vásárlásához és értékesítéséhez, valamint események kezeléséhez.

A rendszer két fő modulból áll:

* **Webes alkalmazás:** Ezen a felületen a felhasználók böngészhetnek az aktuálisan elérhető MTG kártyák között, vásárlásokat bonyolíthatnak le, valamint regisztrálhatnak különböző eseményekre (pl. versenyekre). Emellett naprakész információkat kapnak az újdonságokról, akciókról és a közelgő kiadásokról is.
* **Asztali alkalmazás:** Ez a felület az üzlet működtetéséért felelős adminisztrátorok számára készült. Lehetővé teszi az adatbázisban tárolt termékek kezelését (új kártyák felvétele, meglévők szerkesztése, törlése), az események nyomon követését, valamint a statisztikák és forgalmi adatok megtekintését.

Az alkalmazás több szempontból is törekszik a hatékonyságra és az egyszerű használhatóságra: átlátható kezelőfelülettel, naprakész adatbázissal és stabil háttértechnológiával támogatja a felhasználói élményt, legyen szó vásárlóról vagy rendszergazdáról.

## Rendszerkövetelmények

### Hardver követelmények

Minimális követelmények:

* Processzor: Dual-core 1.8 GHz
* Memória: 4 GB RAM
* Tárhely: Legalább 500 MB szabad hely az alkalmazások és adatbázis számára
* Kijelző: Minimum 1366x768 felbontás
* Internetkapcsolat: Szükséges a webes alkalmazás használatához és adatfrissítésekhez

Ajánlott konfiguráció:

* Processzor: Quad-core 2.5 GHz vagy gyorsabb
* Memória: 8 GB RAM vagy több
* Tárhely: Legalább 1 GB szabad hely
* Kijelző: Full HD (1920x1080) vagy nagyobb felbontás
* Internetkapcsolat: Stabil, szélessávú internetkapcsolat

Az alkalmazás nem igényel különleges hardverelemeket, a legtöbb mai irodai vagy otthoni gépen zökkenőmentesen futtatható.

### Szoftver követelmények

Webes alkalmazás futtatásához:

* Operációs rendszer: Windows 10/11, MacOS vagy bármilyen Linux disztribúció
* Böngésző: Legfrissebb verziójú Google Chrome, Mozilla Firefox, Microsoft Edge vagy Safari
* Frontend futtatásához: React kompatibilis környezet
* Backend kiszolgálásához:
* Node.js (14-es verzió vagy újabb)
* Express.js keretrendszer
* MySQL szerver (XAMPP csomagon belül ajánlott)

Asztali alkalmazás futtatásához:

* Operációs rendszer: Windows 10 vagy újabb
* Futtatókörnyezet: .NET Framework (ajánlott: .NET 4.7 vagy újabb)
* Fejlesztőkörnyezet (ha módosítani akarjuk a programot): Microsoft Visual Studio

Adatbázis-kezeléshez:

* MySQL
* phpMyAdmin (XAMPP csomagon belül)

## 3. A program telepítése EMPTY

Képekkel illusztrált, részletes leírás a program telepítésének a menetéről.

* A leírás alapján a felhasználónak hiba nélkül telepíteni kell tudni a programot. A leírásnak ki kell térnie a telepítés során kiválasztható opciókra is.
* Ha esetleg nincs telepítőprogram, akkor kellő részletességgel le kell írni, hogy mely fájlokat, pontosan hova kell felmásolni, és hogy lehet a programot futtatni. Módosító ablak(1. kép)
* 

1. kép Ez a módosító ablak a programban

Ajánlott terjedelem: 2 -4 oldal, ábrákkal együtt.

## A program használatának a részletes leírása EMPTY

Mindenre kiterjedő, részletes leírás a program használatáról. Alapszabályok:

* Amit leprogramoztál, azt a dokumentációban írd is le, ne legyenek eltitkolt funkciók.
* Minden pontosan, „szájbarágósan” legyen leírva. A dokumentáció alapján a teljesen kezdő, vagy laikus felhasználóknak is használniuk kell tudni a programot.
* A stílus legyen pontos és közérthető, vedd figyelembe, hogy a felhasználói dokumentáció nem szakembereknek készül.
* Ugyanakkor kerüld a laza stílust: rövidítések, smilie-k, szleng kizárva.
* Alkalmazz ábrákat, screenshot-okat , de a ne legyen túlzott a képek aránya a szöveghez képest. Kb. 2-3 oldalanként egy ábra megfelelő.

Ajánlott terjedelem: 10-15 oldal, ábrákkal együtt.

# Fejlesztői dokumentáció

A fejlesztői dokumentáció célja, hogy a segítse program logikájának, illetve a program kódjának a megértését, illetve a program továbbfejlesztését. Szakemberek számára készül, elvárás tehát a pontosság és a szakmai jellegű stílus.

## Az alkalmazott fejlesztői eszközök

A projekt során több különböző fejlesztői eszközt és környezetet alkalmaztunk a program elkészítéséhez, illetve a dokumentáció kidolgozásához. A választott eszközök és nyelvek mind ingyenesen elérhetőek, nyílt forráskódúak vagy oktatási célra jogszerűen használhatók.

Programozási nyelvek:

* C# – A WinForms-alapú asztali alkalmazás megvalósításához
* JavaScript – A webes alkalmazás dinamikus működéséhez
* HTML és CSS – A felhasználói felület szerkezetének és stílusának kialakításához
* SQL – Az adatbázis műveletekhez (lekérdezések, adatmanipulációk)

Fejlesztői környezetek és platformok:

* Visual Studio – A C# alapú asztali alkalmazás fejlesztésére
* Visual Studio Code – A frontend és backend fejlesztéshez (React, Node.js, Express)
* XAMPP – A MySQL adatbázis és a phpMyAdmin kezelőfelület helyi futtatására
* phpMyAdmin – Az adatbázis grafikus kezelőfelülete

Backend és adatbázis:

* Node.js + Express.js – A REST API-k fejlesztéséhez és szerveroldali logika megvalósításához
* MySQL – Az alkalmazás adatainak tárolásához és kezeléséhez

Frontend:

* React – A webes felület komponensalapú felépítéséhez
* Bootstrap – A reszponzív megjelenés biztosításához (nyílt forráskódú CSS keretrendszer)

Verziókezelés és csapatmunka:

* Git + GitHub – Verziókezelés és kollaborációs feladatokhoz; a projekt során több mint 150 commit készült
* Microsoft Teams + Discord – Kommunikációs és projektkoordinációs eszközök

Dokumentáció készítéséhez használt szoftverek:

* Microsoft Word / Google Docs – A dokumentáció megírására és szerkesztésére
* Greenshot / Lightshot – A képernyőképek készítéséhez, amelyek a program működését illusztrálják
* Paint / Paint.NET – Egyszerű képszerkesztési feladatokhoz

Külső modulok és csomagok:

* MySQL2 (npm csomag) – A Node.js és MySQL közötti kapcsolat biztosítására
* React Router – A frontend oldali navigáció kezeléséhez

A használt külső könyvtárak és csomagok mind nyílt forráskódúak, és a hivatalos npm registry-n keresztül letölthetők, így jogszerűen alkalmazhatók oktatási célú projektekben.

## Adatmodell leírása

A projekt során egy relációs adatbázis került kialakításra, amely a rendszer működéséhez szükséges összes entitást és azok kapcsolatát tartalmazza. Az adatbázis neve: cardshop. Az adatbázis célja, hogy hatékonyan és biztonságosan tárolja a felhasználói, termék-, esemény- és vásárlási adatokat.

Az adattáblák és funkcióik

customers  
 Ez a tábla tárolja a regisztrált felhasználók adatait. Minden felhasználó rendelkezik egyedi customer\_id-val, valamint olyan mezőkkel, mint name, email, password, address és registration\_date.

products  
 A bolt kínálatában szereplő termékeket tartalmazza. Minden termék egyedi product\_id-val azonosítható, és tartalmazza a name, price, description, stock\_quantity valamint a category\_id mezőket, amely utóbbi a termék típusát határozza meg.

categories  
 A termékkategóriák (pl. booster, display, commander deck stb.) itt kerülnek tárolásra. Az egyedi azonosító a category\_id, emellett tartalmaz category\_name és category\_description mezőket is.

orders  
 A felhasználók által leadott rendelések adatait tárolja. A customer\_id mező jelöli, hogy ki adta le a rendelést. Az order\_date, total\_amount és status mezők tartalmazzák a vásárlás időpontját, összegét és állapotát.

order\_items  
 Ez a tábla a rendelések részleteit tartalmazza, vagyis hogy egy adott rendelés során milyen termékek kerültek a kosárba. Az order\_id és product\_id mezők kapcsolják össze a többi táblával, a quantity és price\_at\_order a rendeléskori mennyiséget és árat tárolják.

events  
 Az MTG eseményeket tartalmazza, amelyeken a felhasználók részt vehetnek. Minden esemény egyedi event\_id-val azonosítható, tartalmaz nevet, leírást, időpontot és részvételi adatokat (max\_participants, current\_participants).

registrations  
 Ez a tábla a felhasználók eseményekre történő regisztrációit rögzíti. Az event\_id és customer\_id mezők együttesen jelzik, ki melyik eseményre jelentkezett. A registration\_date tárolja a jelentkezés időpontját.

A táblák közötti kapcsolatok

Az adatbázisban több idegen kulcs (foreign key) kapcsolat is létrejött:

A customers tábla kapcsolódik az orders és registrations táblákhoz.

Az orders tábla kapcsolódik az order\_items táblához.

Az order\_items tábla kapcsolódik a products táblához.

A products tábla kapcsolódik a categories táblához.

A registrations tábla kapcsolódik az events táblához.

Ez a kapcsolati struktúra biztosítja az adatok logikus szervezését, és lehetővé teszi a komplex lekérdezések egyszerű és hatékony végrehajtását (pl. egy adott felhasználó megrendeléseinek listázása vagy eseményre jelentkezők száma).

Adatkonzisztencia és adatintegritás

Az idegen kulcsos kapcsolatok biztosítják az adatintegritást és a hivatkozások érvényességét. Például ha egy termék törlésre kerülne, és az szerepel egy rendelésben, a rendszer hibaüzenetet adna vissza, vagy csak a kapcsolt adatok törlésével együtt engedélyezné a műveletet.

Az adatbázisban található trigger is, amely automatikusan frissíti az események résztvevőinek számát, ha új regisztráció történik. Ez különösen fontos az eseményeknél a maximális létszám betartása érdekében.

## Részletes feladatspecifikáció, algoritmusok EMPTY

A program lényeges függvényeinek, az osztályok metódusainak a specifikációja (mit valósít meg az adott függvény, illetve metódus, milyen paraméterei vannak, mi a visszatérési érték)  Az algoritmizálható részek leírása valamilyen algoritmus-leíró eszközzel (struktogram, pszeudo-kód, esetleg UML aktivitás-diagram) Ajánlott terjedelem: a feladat jellegétől függően 2-5 oldal. 5. Forráskód  A teljes forráskódot a nyomtatott dokumentációba nem kell beletenni!  Lehet viszont a nyomtatott dokumentációban a fontosabb kódrészeket magyarázattal szerepeltetni

## Tesztelési dokumentáció EMPTY

1. Legalább 3 különböző teszteset részletes bemutatása.

* különböző felhasználó tevékenységek esetén hogyan reagált a program
* milyen üzeneteket kaptunk
* mi a teendő az egyes üzenetek esetében

1. Normál teszteset, extrém teszteset (bolondbiztosság tesztelése)
2. A tesztelés során kiderült hibák felsorolása A tesztelési dokumentációból derüljön ki, hogy ismered a különböző tesztelési módszereket (pl. fekete doboz, fehér doboz módszer)

Ajánlott terjedelem: a feladat jellegétől függően 2-5 oldal.

# Összefoglalás

## Önértékelés

A projekt kezdeti célja az volt, hogy létrehozzunk egy teljeskörű, felhasználóbarát webes és asztali alkalmazást, amely lehetőséget ad a felhasználók számára a Magic: The Gathering (MTG) termékek böngészésére, vásárlására, eseményekre való jelentkezésére, míg az adminisztrátorok számára egy egyszerű kezelőfelületet biztosít a készlet és események nyomon követésére. Ezen célok elérése érdekében mindketten különböző feladatkörökben dolgoztunk, szorosan együttműködve, hogy a projekt minden szempontja működőképes és átlátható legyen.

A fejlesztés során számos technikai kihívással szembesültünk, mint például az adatbázis struktúrájának kialakítása, a frontend és backend közötti kommunikáció, valamint a felhasználói interakciók kezelésének megoldása. Az adatbázis kapcsolatok megtervezésében és az SQL lekérdezések optimalizálásában fontos szerepet játszottunk, hogy biztosítsuk a rendszer hatékony működését. Különösen az események és regisztrációk kezelésénél, ahol valós idejű adatfrissítésekre volt szükség, számos technikai problémával kellett szembenéznünk. Az ilyen típusú problémák megoldásában egymást segítve dolgoztunk, hogy minden funkció gördülékenyen működjön.

A saját fejlődésünket több területen is észleltük a projekt során. Az egyik legfontosabb tapasztalatunk a **backend fejlesztés** terén szerzett tudásunk mélyítése volt, különösen az Express.js és MySQL alapú adatbázis-kezelésben. A backend felépítése, a REST API-k kialakítása és a megfelelő adatkezelés technikai szempontból jelentős fejlődést hozott számunkra. A frontend fejlesztés során, amelyben React és HTML/CSS használatával készítettük el az alkalmazás felületét, sokat tanultunk az interaktív webalkalmazások fejlesztéséről és optimalizálásáról. Emellett mindketten elmélyültünk a verziókezelő rendszerek, mint a Git és GitHub használatában, melyek biztosították a projekt gördülékeny és hatékony együttműködését.

A felmerült problémák és azok megoldásai is hozzájárultak a fejlődésünkhöz. A különböző technikai kihívások, mint a **backend és frontend közötti integráció**, vagy a **rugalmas adatbázis struktúra kialakítása**, folyamatos tanulásra és kísérletezésre késztettek minket. Az ilyen típusú problémák kezeléséhez mélyebb megértést nyertünk az adatbázisok működéséről, a RESTful API-k tervezéséről és a komplex felhasználói interakciók kezeléséről is.

Összességében úgy érezzük, hogy a projekt során jelentős fejlődést értünk el mind a technikai tudásunkban, mind a projekt menedzselésében. A munkánk nemcsak a konkrét technikai megoldásokban, hanem az együttműködés és a problémamegoldás terén is sokat fejlődött, amely nagyban hozzájárult a sikeres projekt megvalósításához.

## Továbbfejlesztési lehetőségek

A projekt során rengeteg olyan ötlet merült fel, amelyek izgalmasak voltak, de sajnos nem fért bele az időnkbe, vagy technikai korlátok miatt nem valósultak meg. Ugyanakkor ezek mind olyan irányok, amelyek a jövőben továbbfejleszthetők, és amelyek jelentősen hozzájárulhatnának az alkalmazás funkcionalitásához és felhasználói élményéhez.

Nem megvalósított ötletek:

Felhasználói visszajelzés rendszer: Eredetileg terveztük, hogy a felhasználók értékelhetik a megvásárolt termékeket és az eseményeken való részvételt, így javítva a vásárlói élményt és segítve a többi vásárlót a döntéseikben. Bár a funkció elképzelése megvolt, az időhiány miatt nem sikerült implementálni.

Webshop fizetési integrációk: Terveztük, hogy a rendszerhez különböző fizetési lehetőségeket integrálunk, például PayPal-t vagy bankkártyás fizetést, hogy a felhasználók könnyebben, gyorsabban tudjanak vásárolni. A háttérrendszer alapja elkészült, de az ilyen típusú integrációk időigényesek, és nem volt rá elég időnk a fejlesztés során.

Események automatikus értesítése: Szerettük volna, hogy az eseményekről a felhasználók automatikusan értesítést kapjanak e-mailben, vagy akár SMS-ben, amint új esemény kerül meghirdetésre, vagy egy esemény már csak néhány résztvevővel elérhető. Ennek a fejlesztésére sem volt elegendő időnk.

Termékajánlók és személyre szabott ajánlatok: A felhasználói vásárlási előzmények és preferenciák alapján egy olyan ajánló rendszert szerettünk volna építeni, amely személyre szabottan ajánl termékeket. Ez a funkció különösen fontos lehet egy webshop számára, hogy növelje a vásárlói elköteleződést és a vásárlások számát, de nem volt elegendő idő az ilyen algoritmusok kidolgozására és tesztelésére.

Jövőbeli fejlesztési lehetőségek:

Reszponzív webdesign és mobilalkalmazás: A projekt jelenlegi verziója elsősorban asztali számítógépekre van optimalizálva. A jövőben azonban fontos lenne egy teljes mértékben mobilbarát verzió elkészítése, illetve egy külön mobilalkalmazás fejlesztése, amely segíthetne a felhasználóknak még kényelmesebben böngészni, vásárolni és eseményekre jelentkezni.

Részletes keresési és szűrési lehetőségek: A termékek keresése és szűrése jelenleg alapvető funkciókkal van ellátva, de érdemes lenne bővíteni a keresőt részletesebb szűrési lehetőségekkel (pl. árkategória, szín, elérhetőség, népszerűség) és a termékek összehasonlításának lehetőségével. Ezzel a vásárlók gyorsabban megtalálhatják az igényeiknek megfelelő termékeket.

Adminisztrátori elemzések és riportok: Bár már készültek statisztikai funkciók, a rendszer tovább fejleszthető lenne a backend adminisztrátorok számára hasznos riportokkal, például a termékek forgalmának részletesebb elemzésével, a vásárlói viselkedésről készült statisztikákkal, valamint az események népszerűségének és a résztvevői visszajelzések mérésével.

Termékkészlet és árképzés optimalizálása: A készletkezelés tovább fejleszthető lenne automatikus riasztó rendszerekkel, amelyek figyelmeztetik az adminisztrátort, amikor egy termék készlete alacsony, vagy ha az árat módosítani kell a piaci igények függvényében. A dinamikus árazás, amely az időszakos akciókhoz és vásárlói kereslethez igazodik, szintén érdekes lehetőség lenne a rendszer számára.

Social media integráció: A felhasználók számára lehetőséget biztosíthatnánk a termékek és események megosztására közösségi média platformokon (Facebook, Instagram, Twitter), ami segíthetne a márka ismertségének növelésében, valamint az események népszerűsítésében. Ez a funkció különösen fontos lehet, hogy növelje a rendszer elérhetőségét és a vásárlói bázist.

AI-alapú termékajánlók fejlesztése: A jövőben lehetőség lenne mesterséges intelligenciát alkalmazni a vásárlói előzmények és a termékek tulajdonságai alapján történő dinamikus termékajánlásokra. Az ilyen típusú személyre szabott ajánlórendszerek képesek lennének növelni a vásárlási hajlandóságot és a felhasználói élményt.

Összegzés

A projekt során számos ötlet merült fel, amelyek még további értéket adhatnának a rendszerhez. A jövőbeli fejlesztési lehetőségek nemcsak a rendszer funkcionalitását bővítenék, hanem jelentősen javíthatnák a felhasználói élményt is. A fent említett funkciók és ötletek mind olyan irányok, amelyek segíthetnek a rendszer folyamatos fejlődésében és piacon való versenyképességének megőrzésében.

# Felhasznált irodalom

Minden olyan forrás pontos megadása, amelyet a szakdolgozatodban felhasználtál.

A forrás lehet pl.

Könyv. Meg kell adnod a következőket: szerző(k), cím, kiadó, kiadás éve.

Weboldal. Meg kell adnod a linket, az oldal címét Mikor láttad utoljára.

Elektronikus dokumentum. Meg kell adnod a szerzőt, a letöltés helyét, idejét

Ha a szakdolgozatban valamely forrásból szó szerint idézel, akkor a megfelelő szövegrészt idézőjelbe kell tenni, és lábjegyzetben meg kell jelölnöd az idézet forrását.

Ajánlott terjedelem: ½ -1 oldal.

# Ábrajegyzék

[1. kép Ez a módosító ablak a programban 5](#_Toc63250394)