Tartalom jegyzék

[1. Bevezetés 3](#_Toc194053646)

[1.1 A téma részletes ismertetése 3](#_Toc194053647)

[1.2 A témaválasztás részletes indoklása 3](#_Toc194053648)

[1.3 Szakmai célkitűzések részletes bemutatása 4](#_Toc194053649)

[2. Fejlesztői dokumentáció 4](#_Toc194053650)

[2.1 Fejlesztőkörnyezet részletes ismertetése 4](#_Toc194053651)

[2.1.1 Fejlesztéshez használt technológia, választás indoklása 4](#_Toc194053652)

[2.1.2 Fejlesztés során használt szoftverek ismertetése 5](#_Toc194053653)

[2.1.3 Felhasznált külső modulok ismertetése és választás indoklása 6](#_Toc194053654)

[2.1.4 Fejlesztéshez használt hardverkörnyezet bemutatása 6](#_Toc194053655)

[2.2 A kialakított adatszerkezet részletes bemutatása 7](#_Toc194053656)

[2.2.1 Adatbázis kiválasztása és indoklása 7](#_Toc194053657)

[2.2.2 Adatszerkezet és komponensek 7](#_Toc194053658)

[2.2.3 Az adatbázis táblái, mezői részletesen 8](#_Toc194053659)

[2.2.4 ER-diagram ismertetése (szöveges leírás) 10](#_Toc194053660)

[2.3 Program tipikus, egyedi algoritmusai 10](#_Toc194053661)

[2.4 Tesztelés leírása, tesztdokumentáció 11](#_Toc194053662)

[2.5 Fejlesztési lehetőségek 12](#_Toc194053663)

[3. Felhasználói dokumentáció 12](#_Toc194053664)

[3.1 A program céljának és lényeges funkcióinak összefoglalása 12](#_Toc194053665)

[3.2 Szükséges hardvereszközök és szoftverek felsorolása 12](#_Toc194053666)

[3.3 Telepítés és indítás lépéseinek részletes ismertetése 13](#_Toc194053667)

[3.4 A program részletes bemutatása 13](#_Toc194053668)

[3.4.1 Webes felület használata: 13](#_Toc194053669)

[3.4.2 Admin WPF alkalmazás használata: 13](#_Toc194053670)

[3.5 Helytelen használatból adódó hibajelzések magyarázata 14](#_Toc194053671)

[3.6 Információkérés lehetőségeinek megadása 14](#_Toc194053672)

[3.7 Admin felhasználói felület lehetőségei (WPF): 14](#_Toc194053673)

[3.8 Szükséges fejlesztői szoftverek és telepítési útmutató 14](#_Toc194053674)

[3.8.1 Visual Studio telepítése (Backend fejlesztéshez) 14](#_Toc194053675)

[3.8.2 Visual Studio Code telepítése (Frontend fejlesztéshez) 15](#_Toc194053676)

[3.8.3 XAMPP telepítése (Apache, MySQL szerver futtatásához) 15](#_Toc194053677)

[3.8.4 Swagger használata (Backend API tesztelése) 15](#_Toc194053678)

[3.8.5 DbDiagram.io használata (Adatbázis ER-diagram készítéséhez) 16](#_Toc194053679)

[3.8.6 Figma használata (Felhasználói felület tervezéséhez) 16](#_Toc194053680)

[3.8.7 Trello használata (Projektmenedzsmenthez) 16](#_Toc194053681)

[3.8.8 Backend fájlok kezelése és parancsok (Visual Studio terminálban) 16](#_Toc194053682)

[3.8.9 Backend projekt indítása Visual Studioból 17](#_Toc194053683)

[4. Összefoglalás, köszönetnyilvánítás 17](#_Toc194053684)

[4.1 A projekt értékelése 17](#_Toc194053685)

[4.2 A program további hasznosulása, jövőbeli tervek 17](#_Toc194053686)

[4.3 Köszönetnyilvánítás 18](#_Toc194053687)

[5. Irodalomjegyzék 18](#_Toc194053688)

[5.1 Felhasznált szakmai könyvek 18](#_Toc194053689)

[5.2 Felhasznált weboldalak 19](#_Toc194053690)

# 1. Bevezetés

## 1.1 A téma részletes ismertetése

A **BuliHub** egy innovatív, webalapú közösségi platform, mely lehetővé teszi, hogy a felhasználók egyszerűen, gyorsan, átlátható módon böngészhessenek, keresgélhessenek különféle események, rendezvények, bulik között, és regisztrációt követően saját eseményeket is létrehozhassanak, szerkeszthessenek.

A platformon két különböző felhasználói csoport különböztethető meg: az átlagos felhasználók és a hitelesített felhasználók. Átlagos felhasználóként bárki regisztrálhat, eseményeket hozhat létre, böngészhet, jelentkezhet eseményekre. A hitelesített felhasználók azonban ennél magasabb szintű jogosultsággal rendelkeznek. Ők elsősorban olyan szervezők, akik szórakozóhelyeket, klubokat, rendezvénytermeket képviselnek, és hitelesítési folyamaton esnek át a platform adminisztrátorai által. Ezáltal megbízhatóbbá, professzionálisabbá válnak a felhasználók szemében. A későbbiekben a hitelesített funkció lehetősége további prémium lehetőségekkel is bővülhet, például kiemelt események hirdetése vagy fizetős események kezelése.

A platformhoz egy dedikált, desktop alapú adminisztrációs felület is kapcsolódik, amely WPF technológián alapul. Ez az adminisztrációs felület lehetőséget biztosít a rendszer adminisztrátorainak, hogy részletesen kezelhessék mind a regisztrált felhasználókat, mind pedig az eseményeket. Az adminisztrátorok számára elérhetővé válik a felhasználók jogainak szabályozása, inaktiválása, törlése, valamint az események adatainak módosítása, törlése vagy új események létrehozása. Az adminfelület célja, hogy a platform működése folyamatosan zavartalan, átlátható és biztonságos maradjon.

A rendszer főbb funkciói közé tartozik még a biztonságos felhasználói hitelesítés, amely JWT (JSON Web Token) alapú megoldással történik, biztosítva ezzel a felhasználók adatainak védelmét, valamint a platform biztonságát. A programban SMTP protokollt alkalmazunk automatizált email értesítések kiküldésére, például regisztráció után visszaigazoló email küldése, elfelejtett jelszavak visszaállítása, vagy eseményekkel kapcsolatos automatikus értesítések esetén

## 1.2 A témaválasztás részletes indoklása

A BuliHub platform elkészítésének ötlete személyes tapasztalatokra és piackutatásra épül. Magyarországon több különféle közösségi média platform létezik, amelyeken lehetőség van események meghirdetésére, ám ezek általában vagy túl általános jellegűek (például Facebook események), vagy túlzottan specializáltak és korlátozottak funkcionalitás szempontjából. Hiányzott azonban egy olyan átfogó, egyszerűen kezelhető platform, amely kifejezetten a rendezvényszervezés, buli szervezés igényeit szolgálja ki. A piaci felmérések során egyértelművé vált, hogy lenne igény egy olyan dedikált platformra, amely egyszerűsíti az események szervezését és a közönség elérését, miközben megbízható hátteret nyújt a hitelesített szervezők számára is.

A saját élethelyzetemet tekintve is relevánsnak tartom a témát, mivel magam is számos alkalommal szembesültem azzal a problémával, hogy események szervezése során több különálló eszközt, platformot kellett használnom párhuzamosan, ami jelentősen megnehezítette a kommunikációt és az adminisztrációs teendőket. Ennek a nehézségnek a kiküszöbölésére, valamint saját tapasztalataim alapján olyan megoldás kidolgozására vállalkoztam, amely minden szükséges funkciót integráltan, egy helyen kínál.

Továbbá, a fejlesztés szakmai oldalát tekintve is kiváló lehetőségnek tartottam egy ilyen komplex webes alkalmazás elkészítését, amely során lehetőségem nyílik a React keretrendszer, a TypeScript nyelv, valamint a .NET backend, Entity Framework Core, SQL Server adatbázis-kezelés mélyebb megismerésére és gyakorlatias alkalmazására. A technológiai tudás fejlesztésének lehetősége további motivációként szolgált, mivel ezek a technológiák a jelenlegi fejlesztői munkaerőpiacon rendkívül keresettek, így komoly szakmai előnyt jelenthetnek számomra a későbbiekben.

## 1.3 Szakmai célkitűzések részletes bemutatása

A BuliHub platform megalkotásakor több szakmai célkitűzést is meghatároztam magamnak, amelyeket a projekt során sikeresen teljesíteni kívántam. Elsődleges célom egy modern technológiai alapokra épülő, stabil, biztonságos, jól skálázható webalkalmazás elkészítése volt. Ehhez a frontend oldalon a React keretrendszert választottam, amely tökéletes eszköz az interaktív és felhasználóbarát webes felületek létrehozására, a TypeScript nyelv pedig segített abban, hogy a kód jól strukturált, könnyen kezelhető és hibamentes legyen.

A backend fejlesztés során a választásom a Microsoft által kínált .NET 7 platformra esett, amely kimagasló teljesítményével, biztonságával és széles körű támogatottságával megfelelt minden elképzelésemnek. Az Entity Framework Core ORM és az SQL Server adatbázis használata pedig lehetővé tette, hogy gyorsan, hatékonyan tudjam megoldani az adatkezelési feladatokat, valamint biztonságosan tárolni és lekérdezni az adatokat.

Külön célként jelent meg a biztonságos felhasználókezelés, amely során JWT autentikációs technológiát alkalmaztam, továbbá figyeltem a megfelelő validációs és hibakezelési mechanizmusok implementációjára is.

További szakmai célként tűztem ki a program alapos dokumentálását, a tesztdokumentáció elkészítését, valamint a szoftver alapos tesztelését többféle környezetben, különböző terhelések alatt. Ezek a lépések biztosították, hogy az elkészült alkalmazás valóban megbízható, jól működő, valós körülmények között is használható szoftver legyen.

Végül pedig fontos célként szerepelt a program továbbfejleszthetőségének biztosítása. A fejlesztés során úgy építettem fel az alkalmazás architektúráját, hogy lehetőség legyen a későbbi új funkciók integrálására, mint például mobil applikáció fejlesztése, online fizetési rendszerek integrációja vagy akár analitikai funkciók beépítése is.

# 2. Fejlesztői dokumentáció

## 2.1 Fejlesztőkörnyezet részletes ismertetése

### 2.1.1 Fejlesztéshez használt technológia, választás indoklása

A BuliHub fejlesztése során gondosan választottam ki azokat a technológiákat, melyek biztosítják a projekt stabilitását, könnyű továbbfejleszthetőségét, valamint a fejlesztési folyamat gördülékenységét.

**Frontend technológiák**

A frontend fejlesztéshez a React keretrendszert választottam, amely világszerte az egyik legnépszerűbb JavaScript-alapú UI-készítő könyvtár. React előnye, hogy komponens-alapú architektúrát kínál, amely segítségével újrahasznosítható komponensekből építhetjük fel alkalmazásunkat, ezzel jelentősen csökkentve a kódismétlődést és növelve a karbantarthatóságot. Az alkalmazás állapotának kezelése egyszerűvé válik az állapotkezelő eszközökkel, mint például a Redux vagy React Context. A React további előnye, hogy nagy közösséggel rendelkezik, amely támogatja a gyors tanulást és a problémák hatékony megoldását.

A JavaScript helyett a TypeScript programnyelvet választottam, amely egy erősen típusos JavaScript kiterjesztés. TypeScript segítségével jelentősen csökken a fejlesztési hibák száma, mivel a típusellenőrzés már a fejlesztés idején megtörténik, így gyorsan azonosíthatók a hibás értékadások és a típusokkal kapcsolatos problémák. A fejlesztőkörnyezet integrált támogatása révén könnyebben követhető, strukturáltabb, karbantarthatóbb kód jön létre.

**Backend technológiák**

A backend oldalon a .NET 7 keretrendszer mellett döntöttem, amely egy rendkívül robusztus, megbízható és jól skálázható környezet, és a Microsoft támogatása garantálja a folyamatos fejlődést és frissítéseket. Ezáltal a projekt hosszútávon is fenntartható és könnyen bővíthető marad. A .NET 7 hatékonyan támogatja a RESTful API-k létrehozását, ami pontosan megfelelt a projekt követelményeinek.

Az adatok tárolásához relációs adatbázist, konkrétan Microsoft SQL Server-t használtam, mivel ez megbízható, nagy teljesítményű adatkezelést kínál. A táblák és adatok közötti relációkat könnyedén kezelhettem az Entity Framework Core ORM használatával, ami drasztikusan leegyszerűsíti a lekérdezések megírását, az adatbázis-kezelést, valamint biztonságosabbá teszi az adatkezelési folyamatokat.

**Admin felület technológiái**

Az admin felülethez WPF (Windows Presentation Foundation) technológiát választottam. A WPF egy igen rugalmas, Windows-alapú alkalmazások fejlesztésére alkalmas keretrendszer, mely lehetőséget ad egyedi, intuitív, látványos felhasználói interfész létrehozására, ugyanakkor natívan támogatja a .NET keretrendszert, így kiválóan illeszkedett a projekt backend részéhez.

### 2.1.2 Fejlesztés során használt szoftverek ismertetése

A fejlesztői munkát az alábbi eszközök segítették, jelentősen növelve a munka hatékonyságát:

* **Visual Studio 2022**  
  Az elsődleges fejlesztői környezet, amelyben a backend, valamint a WPF admin kliens fejlesztése zajlott. A Visual Studio támogatja a .NET 7, C#, Entity Framework Core technológiákat, valamint integrált hibakereső és unit tesztelő eszközöket tartalmaz.
* **Visual Studio Code**  
  Frontend fejlesztéshez használt könnyűsúlyú, gyors és flexibilis kódszerkesztő, amely kiváló TypeScript és React támogatással rendelkezik, továbbá számos kiegészítő bővítménnyel (pl. ESLint, Prettier) teszi hatékonyabbá a fejlesztési folyamatot.
* **Postman**  
  Az API tesztelésére használt eszköz, amely segítségével gyorsan, egyszerűen tudtam tesztelni az egyes végpontokat, ellenőrizni azok működését és válaszidejét. Postman-ben automatizált tesztelési folyamatokat is kialakítottam, melyek jelentősen leegyszerűsítették a hibakeresést.
* **SQL Server Management Studio**  
  Az adatbázis közvetlen kezelésére és lekérdezések optimalizálására használt eszköz, amely lehetőséget nyújt az adatbázis struktúrájának könnyű áttekintésére, az SQL lekérdezések fejlesztésére és a teljesítmény optimalizálására.
* **Git, GitHub**  
  Verziókövetési és kollaborációs eszközök, melyek lehetővé tették a verziók közötti egyszerű navigálást, kód változások nyomon követését, valamint biztosították a projekt folyamatos biztonsági mentését.

### 2.1.3 Felhasznált külső modulok ismertetése és választás indoklása

A fejlesztés során néhány harmadik féltől származó modul is felhasználásra került, melyek segítségével gyorsabban és biztonságosabban tudtam implementálni egyes funkciókat:

* **Axios (Frontend)**  
  HTTP kérések egyszerű kezelésére alkalmaztam, amely nagyban megkönnyítette az API kommunikációt, valamint egyszerűbbé tette a hibaüzenetek és válaszok kezelését.
* **Framer Motion (Frontend)**  
  Interaktív és vizuálisan vonzó felhasználói interfész animációk létrehozására szolgáló modul, amely jelentősen javítja a felhasználói élményt.
* **ASP.NET Identity (Backend)**  
  A felhasználók biztonságos azonosítására és jogosultságok kezelésére használt modul, amely előre elkészített megoldásokkal támogatja a regisztrációt, bejelentkezést és jogosultságkezelést.
* **JWT - JSON Web Token (Backend)**  
  Biztonságos autentikáció és API kérés-hitelesítés megvalósítására alkalmaztam. A JWT segítségével biztosítottam a felhasználói session-ök biztonságát és a backend szolgáltatások védelmét.
* **SMTP Email modul (Backend)**  
  Az alkalmazás automatikus email-küldési képességeit biztosította. Olyan értesítésekhez használtam, mint például regisztrációs visszaigazolás, esemény értesítés, jelszó visszaállítás.

### 2.1.4 Fejlesztéshez használt hardverkörnyezet bemutatása

A fejlesztési folyamat során két eszközzel dolgoztam, amelyek mindegyike megfelelő erőforrást biztosított a projekt hatékony kivitelezéséhez. Az első eszköz egy Lenovo ThinkPad T480s laptop volt, amely kiváló mobilitását és megbízhatóságát ötvözi a modern üzleti laptopok kategóriájában. Ennek a laptopnak a konfigurációja a következő volt:

• **Processzor:** Intel® Core™ i5-10210U CPU @ 1.60GHz, 2112 MHz, 4 mag, 8 logikai processzor  
• **Memória:** 8,00 GB DDR4 RAM  
• **Tárhely:** 238,47 GB (256 052 966 400 bájt)  
• **Operációs rendszer:** Windows 10 64bit

Ez az eszköz kiválóan alkalmas volt a mindennapos fejlesztési feladatokhoz, gyors fordításhoz és a komplexebb tesztelési műveletek végrehajtásához, különösen mobil környezetben.

Emellett dolgoztunk egy másik számítógépen is, amely hasonló konfigurációval rendelkezett, bár itt némileg eltérő hardverparaméterek voltak érvényben. Ez a számítógép a következő adatokkal bírt:

• **Processzor:** Intel® Core™ i3-6100 CPU @ 3.7GHz  
• **Memória:** 8 GB RAM  
• **Tárhely:** 256 GB  
• **Operációs rendszer:** Windows 10 64bit

Ez az asztali gép is elegendő teljesítményt biztosított a fejlesztési feladatokhoz, különösen a stabil környezet és a Windows 10 rendszer által nyújtott megbízhatóság révén. Mindkét eszköz összhangban működött, így a fejlesztési és tesztelési folyamat zökkenőmentesen zajlott, függetlenül attól, hogy mobil vagy asztali környezetben dolgoztunk. Ezek az eszközök lehetővé tették, hogy rugalmasan alkalmazkodjunk a különböző feladatokhoz, és hatékonyan kezeljük a projekt különböző aspektusait.

## 2.2 A kialakított adatszerkezet részletes bemutatása

### 2.2.1 Adatbázis kiválasztása és indoklása

Az alkalmazás adatainak tárolására relációs adatbázist, konkrétan Microsoft SQL Server-t választottam, amely egy kiforrott, stabil és nagy teljesítményű relációs adatbáziskezelő rendszer. A választás oka, hogy a Microsoft SQL Server tökéletesen illeszkedik a választott backend technológiához (.NET 7), és teljes integrációt biztosít az Entity Framework Core ORM keretrendszerrel, amely gyors és biztonságos fejlesztést tesz lehetővé. Az SQL Server stabilan kezeli a nagy mennyiségű adatot, támogatja a bonyolultabb relációkat és a tranzakciókezelést, valamint rendelkezik megfelelő biztonsági eszközökkel is (pl. jogosultságok kezelése, titkosítás).

### 2.2.2 Adatszerkezet és komponensek

A rendszer adatszerkezete két fő rétegből áll: a backend adatszolgáltatási logikájából és a frontend felhasználói felületből.  
Backend Entitások és Modellek  
  
    Event.cs:  
    Az események modellje, mely tartalmazza az esemény nevét, leírását, kezdési és befejezési időpontját, helyszín adatait (LocationName, Address, Equipment), szervező nevét (OrganizerName), státuszát, vendégek számát és témáját.  
    Fontos megjegyzés: Az entitás opcionális kapcsolatban áll a ServiceProvider entitással, melyet az esetleges szolgáltató felhasználási esetekhez használunk.  
     
  
Party.cs:  
Egyszerűbb eseménymodell, GUID alapú azonosítóval, melyet elsősorban a gyors prototípusok és egyszerűbb buli regisztrációk kezelésére használtunk.  
  
  
Report.cs:  
A felhasználók által leadott jelentéseket tartalmazza, melyben a jelentő felhasználó, az érintett esemény, a jelentés tartalma, a jelentés dátuma és státusza található meg.  
  
  
User.cs, ApplicationUser.cs, CertifiedUser.cs:  
A rendszer felhasználóit több szinten kezeljük: a hagyományos User entitás, az [ASP.NET](http://asp.net/) Core Identity alapú ApplicationUser, valamint a hitelesített felhasználókat reprezentáló CertifiedUser entitás.  
  
  
  
  
  
Ticket.cs:  
Jegyvásárlások adatait tartalmazza, összekötve az eseményekkel és a felhasználókkal.  
  
  
Role.cs:  
Egyedi szerepek és jogosultsági szintek kezelése, melyek a felhasználók jogosultságait szabályozzák.  
  
  
ServiceProvider.cs:  
Olyan entitás, amely a szolgáltatókat reprezentálja, összekapcsolva a felhasználói adatokkal.  
  
  
Admin.cs:  
Az adminisztrátorok adatait tartalmazza, beleértve a felhasználónevet, hash-elt jelszót és regisztrációs dátumot.  
  
DTO-k (Data Transfer Objects)  
  
    CreateEventDto.cs:  
    Az esemény létrehozásához szükséges adatokat tartalmazza (partyName, date, time, locationName, address, equipment, guests, theme, description).  
     
  
UpdateEventDto:  
Az esemény frissítéséhez használt DTO, melyet az EventsController implementációjában alkalmazunk.  
  
RegisterDto, LoginDto, ForgotPasswordDto:  
A felhasználói regisztráció, bejelentkezés és jelszó visszaállítás során használt adatok.

### 2.2.3 Az adatbázis táblái, mezői részletesen

A BuliHub alkalmazás adatbázisa a következő fő táblákból áll:

**Users (Felhasználók)**

* **Id (int)** - Egyedi azonosító, elsődleges kulcs
* **Username (nvarchar(50))** - Felhasználónév, kötelező, egyedi
* **Email (nvarchar(100))** - E-mail cím, kötelező, egyedi
* **PasswordHash (nvarchar(max))** - Titkosított jelszó
* **RegistrationDate (datetime)** - Regisztráció dátuma
* **IsActive (bit)** - Aktív státusz jelölője
* **Role (nvarchar(20))** - Felhasználó szerepköre (user, admin, certified)
* **Address (nvarchar(200))** - Lakcím (opcionális)

**Events (Események)**

* **EventId (int)** - Egyedi eseményazonosító, elsődleges kulcs
* **Title (nvarchar(150))** - Esemény címe
* **Description (nvarchar(max))** - Részletes eseményleírás
* **Location (nvarchar(200))** - Esemény helyszíne
* **EventDate (datetime)** - Az esemény időpontja
* **MaxParticipants (int)** - Maximális résztvevők száma
* **OrganizerUserId (int)** - Szervező felhasználó azonosítója (FK Users.Id)
* **IsCertified (bit)** - Hitelesített szervező által létrehozott esemény-e

**Admins (Adminisztrátorok)**

* **AdminId (int)** - Admin azonosító, elsődleges kulcs
* **UserId (int)** - Felhasználói azonosító (FK Users.Id)
* **Permissions (nvarchar(max))** - Jogosultságok (esemény kezelés, felhasználó kezelés)

**Tickets (Jegyek)**

* **TicketId (int)** - Egyedi jegyazonosító, elsődleges kulcs
* **EventId (int)** - Kapcsolódó esemény azonosítója (FK Events.EventId)
* **UserId (int)** - Vásárló felhasználó azonosítója (FK Users.Id)
* **PurchaseDate (datetime)** - Vásárlás dátuma
* **TicketPrice (decimal(10,2))** - Jegy ára
* **IsPaid (bit)** - Fizetés státusza (későbbi fizetési rendszer integrációhoz)

**ServiceProviders (Hitelesített szervezők/szolgáltatók)**

* **ProviderId (int)** - Egyedi azonosító, elsődleges kulcs
* **UserId (int)** - Kapcsolódó felhasználó azonosítója (FK Users.Id)
* **CompanyName (nvarchar(100))** - Szervezet neve
* **TaxNumber (nvarchar(20))** - Adószám
* **IsApproved (bit)** - Hitelesítés státusza

**Reports (Felhasználói jelentések)**

* **ReportId (int)** - Jelentés azonosítója, elsődleges kulcs
* **UserId (int)** - Jelentő felhasználó azonosítója (FK Users.Id)
* **EventId (int)** - Kapcsolódó esemény azonosítója (FK Events.EventId)
* **Message (nvarchar(max))** - Jelentés tartalma
* **ReportDate (datetime)** - Jelentés beküldési dátuma

### 2.2.4 ER-diagram ismertetése (szöveges leírás)

Az adatbázis táblái a következő kapcsolatokkal rendelkeznek:

* **Users** tábla egy-egy kapcsolatban állhat az **Admins** és a **ServiceProviders** táblával.
* **Users** és **Events** táblák egy-a-többhöz kapcsolatban állnak (egy felhasználó több eseményt hozhat létre, de egy eseménynek csak egy szervezője lehet).
* **Events** és **Tickets** egy-a-többhöz kapcsolat (egy eseményhez több jegy kapcsolódhat, egy jegy egy eseményhez kapcsolódik).
* **Users** és **Tickets** egy-a-többhöz kapcsolat (egy felhasználó több jegyet vásárolhat, de egy jegy egy felhasználóhoz tartozik).
* **Reports** tábla mind a **Users** és **Events** táblákkal egy-a-többhöz kapcsolódik (egy felhasználó több jelentést tehet, és egy eseményre több jelentés érkezhet).

## 2.3 Program tipikus, egyedi algoritmusai

***Esemény Létrehozása***  
  
A buli létrehozásának folyamata a következő lépésekből áll:  
  
 ***Időpont Feldolgozás:***  
    A felhasználó által megadott dátum és „óra:perc” formátumú string alapján a rendszer az idő adatokat feldolgozza. A stringet a Split metódussal bontjuk szét, majd az értékeket egy DateTime objektumba integráljuk. A rendszer alapértelmezettként 4 órás eseményt tartalmaz, melyet az EndDate kiszámítása során adunk hozzá.  
  
    ***Adatátvitel a DTO-n keresztül:***  
    A CreateEventDto objektum tartalmazza az összes szükséges adatot, melyet a backend az esemény entitás létrehozásához használ. Az átvitel során JSON formátumban kerülnek továbbításra a paraméterek, melyeket a rendszer validál.  
  
    ***Jogosultság Ellenőrzése:***  
    A JWT tokenből kinyert felhasználói név alapján ellenőrizzük, hogy a bejelentkezett felhasználó jogosult-e az esemény létrehozására. Ezt követően a rendszer az esemény adatait az adatbázisba menti, és visszaadja a létrehozott esemény adatait.  
  
***Animációk a Frontenden***  
  
A Framer Motion könyvtár segítségével valósítottuk meg az interaktív animációkat, különösen a ***statisztikai kártyák esetében:***  
  
    ***Mobil vs. Asztali Nézet:***  
    Mobil nézetben a kártyák egymás alatt jelennek meg, míg asztali nézetben az animációk segítségével pozícionáljuk és mozgatjuk őket.  
  
    ***Animációs Változatok:***  
    Különböző motion variant-ek (például initial, animate, exit) segítségével biztosítottuk, hogy a kártyák be- és kilépése zökkenőmentes legyen, illetve reagáljon az oldalsó görgetésre és az intersection observer eseményekre.  
  
***Autentikáció és Jogosultság***  
    ***JWT Token Generálás:***  
    A bejelentkezés során a backend ellenőrzi a felhasználó hitelességét, majd a token generálása során a felhasználó emailje, neve és egyéb releváns adatai kerülnek a tokenbe. A token 7 napos érvényességgel kerül kiadásra, melyet minden későbbi kérés hitelesítéséhez használunk.  
  
    ***Jelszó Hash-elés:***  
    A regisztráció során a felhasználók jelszavait hash-eljük, hogy az adatbázisban ne tárolódjanak egyszerű szövegként. Ez növeli a biztonságot, és megakadályozza a jelszavak illetéktelen kiszivárgását.  
  
***Email Értesítések***  
  
***A jelszó visszaállítási funkció során a rendszer:***  
  
    Generál egy ideiglenes jelszót egy véletlenszerű karakterkészletből.  
  
    A generált token alapján visszaállítja a felhasználó jelszavát.  
  
    Az új ideiglenes jelszót emailben elküldi a felhasználónak, a SMTP protokoll segítségével, megfelelő biztonsági beállításokkal (SSL, port 587).

## 2.4 Tesztelés leírása, tesztdokumentáció

A tesztelést az alábbi körülmények között végeztem:

1. **Böngésző-kompatibilitás:** Chrome, Firefox, Edge böngészőkkel teszteltem a frontend helyes megjelenését és működését.
2. **Hibás adatkezelés:** Regisztráció és eseménylétrehozás során invalid adatokat (például hibás e-mail, múltbeli esemény dátum) adtam meg, a rendszer megfelelő hibajelzéseket adott.
3. **Terhelési teszt:** Egyszerre több ezer eseményt töltöttem be, az alkalmazás teljesítménye stabil maradt.
4. **Jogosultságkezelés:** Jogosulatlan felhasználókkal próbáltam admin műveleteket végezni, az alkalmazás megfelelően blokkolta a hozzáférést.
5. **Email küldés:** SMTP email-ek kézbesítési idejét ellenőriztem különböző szolgáltatók felé (Gmail, Outlook).
6. **WPF admin-felület teszt:** Felhasználók és események kezelését teszteltem, módosításokat végeztem, majd ellenőriztem az adatbázisban.

## 2.5 Fejlesztési lehetőségek

* **Mobil alkalmazás fejlesztése:** Android és iOS platformokra React Native segítségével.
* **Online fizetési rendszer integrációja:** Például SimplePay, Stripe segítségével, amely lehetővé tenné fizetős események bevezetését.

# 3. Felhasználói dokumentáció

## 3.1 A program céljának és lényeges funkcióinak összefoglalása

A **BuliHub** egy webes platform, amely a rendezvények, események és bulik átfogó kezelésére szolgál. A platform lehetőséget nyújt bárki számára arra, hogy különféle eseményeket fedezzen fel, böngésszen közöttük, továbbá regisztrációt követően akár saját eseményeket hozzon létre és szervezzen.

A rendszer felhasználóbarát módon kezeli az eseményadatokat, biztosítva a résztvevők számára a könnyű tájékozódást, a szervezők számára pedig az egyszerű, gördülékeny eseményszervezést. A platformon hitelesített és nem hitelesített felhasználók is megtalálhatók. A hitelesített felhasználók olyan szórakozóhelyek vagy eseményszervező szervezetek, amelyek már bizonyították hitelességüket és professzionalizmusukat.

A rendszerhez kapcsolódik egy adminisztrációs felület is, amely lehetővé teszi a regisztrált felhasználók és az események hatékony, gyors kezelését, módosítását és ellenőrzését. Az adminisztrációs eszköz egy önálló WPF alapú alkalmazás.

## 3.2 Szükséges hardvereszközök és szoftverek felsorolása

**Hardver követelmények:**

A webes felület használatához nincs különösen magas hardverigény, az alábbi konfiguráció vagy annál jobb bármely számítógépen, tableten, vagy okostelefonon elegendő:

* Processzor: Intel Core i3 vagy ezzel egyenértékű
* Memória: 4 GB RAM
* Tárhely: legalább 50 MB szabad hely
* Internet kapcsolat: stabil internet-hozzáférés

Az adminisztrációs WPF alkalmazás (csak adminok számára):

* Processzor: Intel Core i5 vagy jobb
* Memória: 8 GB RAM ajánlott
* Tárhely: legalább 100 MB szabad hely
* Operációs rendszer: Windows 10 vagy újabb verzió

**Szoftver követelmények:**

* Webes felülethez ajánlott böngészők: Chrome, Firefox, Microsoft Edge, Safari
* JavaScript támogatás engedélyezése szükséges
* Adminisztrációs WPF alkalmazáshoz: Microsoft .NET 7.0 futtatókörnyezet telepítése szükséges

## 3.3 Telepítés és indítás lépéseinek részletes ismertetése

**Webes alkalmazás használata:**

A webes alkalmazás telepítést nem igényel, közvetlenül böngészőből érhető el az alábbi címen:  
[**https://www.bulihub.hu**](https://www.bulihub.hu)

* Nyissa meg böngészőjét.
* Írja be a weboldal címét a címsorba.
* Az oldal automatikusan betöltődik.
* Használhatja az oldalt vendégként, vagy regisztrálhat/beléphet felhasználóként.

## 3.4 A program részletes bemutatása

### 3.4.1 Webes felület használata:

* **Kezdőlap:** Itt láthatóak a legnépszerűbb események, valamint egy keresősáv található, amelyben város, dátum vagy esemény nevére kereshet.
* **Esemény létrehozása:** Regisztráció után kattintson az „Esemény létrehozása” gombra. Töltse ki az esemény adatait: név, dátum, helyszín, résztvevők maximális száma, részletes leírás. Minden mező kitöltése kötelező.
* **Események böngészése:** Egyszerűen görgessen az oldalon, vagy használja a keresőmezőt, illetve a szűrőket.
* **Profil kezelése:** Bejelentkezés után a „Profilom” menüpontban tudja módosítani személyes adatait, mint pl. jelszó módosítás.

### 3.4.2 Admin WPF alkalmazás használata:

* **Bejelentkezés:** Indítás után adja meg adminisztrátori email-címét és jelszavát.
* **Felhasználók kezelése:** Az „Users” menüpontban lehetősége van a felhasználók státuszának megváltoztatására (aktív/inaktív), törlésére vagy jogosultság módosítására.
* **Események kezelése:** Az „Events” menüpont alatt új eseményeket hozhat létre, módosíthat vagy törölhet meglévőket.
* **Jelentések kezelése:** A „Reports” menüpont alatt felhasználói jelentéseket láthatja, válaszolhat rájuk, vagy jelölheti őket megoldottként.

## 3.5 Helytelen használatból adódó hibajelzések magyarázata

Példák a rendszer hibajelzéseire:

* **„E-mail formátuma nem megfelelő!”**: Regisztrációkor, ha hibás email címet ad meg.
* **„Ez az email cím már regisztrálva van!”**: Már létező email-címmel történő regisztráció esetén.
* **„Kötelező mezők hiányoznak!”**: Esemény létrehozásakor, ha kihagy valamilyen kötelező adatot.
* **„Az esemény dátuma nem lehet múltbéli!”**: Ha hibás időpontot ad meg esemény létrehozásakor.

## 3.6 Információkérés lehetőségeinek megadása

Ha kérdése merülne fel, segítségre van szüksége, vagy hibát tapasztal, kérjük vegye fel velünk a kapcsolatot az alábbi módokon:

* **Kapcsolatfelvételi űrlap:** A weboldal „Kapcsolat” menüpontján keresztül.
* **E-mail:** support@bulihub.hu
* **Telefonszám:** +36 1 234 5678 (munkaidőben)

## 3.7 Admin felhasználói felület lehetőségei (WPF):

* **Felhasználói státusz módosítása:** Aktív, inaktív állapotok váltása, jogosultság beállítása.
* **Új esemény hozzáadása:** Admin jogosultsággal esemény hozzáadása közvetlenül.
* **Jelentések kezelése:** Felhasználói panaszok kezelése, megválaszolása.

## 3.8 Szükséges fejlesztői szoftverek és telepítési útmutató

Ebben az alfejezetben részletesen bemutatjuk, hogy a fejlesztési és telepítési folyamat során milyen szoftvereket használtunk, és hogyan kell azokat telepíteni, konfigurálni, illetve milyen parancsokat szükséges kiadni a terminálban.

### 3.8.1 Visual Studio telepítése (Backend fejlesztéshez)

A **Visual Studio** egy fejlesztőkörnyezet (IDE), amelyet a backend fejlesztéséhez (.NET alapú projektekhez) használtunk.

* Töltsd le innen: [Visual Studio Community](https://visualstudio.microsoft.com/downloads/)
* Indítsd el a telepítőt, válaszd ki az „ASP.NET and web development” opciót.
* Kattints az „Install” gombra, várd meg a telepítés végét.
* Indítsd el a programot, majd nyisd meg benne a backend projektet (Bulihub\_Backend.sln).

### 3.8.2 Visual Studio Code telepítése (Frontend fejlesztéshez)

A **Visual Studio Code** egy könnyen használható, egyszerű kódszerkesztő, a frontend fejlesztéshez használtuk.

* Töltsd le innen: [Visual Studio Code](https://code.visualstudio.com/)
* Indítsd el a telepítőt, kattints többször a „Next” gombra, majd „Install”.
* Indítás után nyisd meg benne a frontend mappát.

**Szükséges package-ek telepítése terminálban (frontend)**:

Nyisd meg a terminált a Visual Studio Code-ban (Ctrl + `):

Írd be a parancsokat sorban, minden parancs után nyomj Enter-t:

npm i

npm i motion

npm i react-router-dom

npm i react-leaflet@4.2.1

npm run dev

Az utolsó parancs (npm run dev) után a weboldal automatikusan elindul és elérhető lesz böngészőben a http://localhost:5173 címen.

### 3.8.3 XAMPP telepítése (Apache, MySQL szerver futtatásához)

A XAMPP biztosítja a szükséges Apache és MySQL környezetet az adatbázis kezeléséhez.

* Töltsd le innen: XAMPP
* Telepítés után nyisd meg a XAMPP Control Panel-t.
* Kattints a „Start” gombra az Apache és MySQL sorában is.
* Ha elindult mindkettő, kattints a MySQL sorban található „Admin” gombra. Ez megnyitja a PhpMyAdmin felületet a böngésződben.
* Itt importáld be az adatbázist: kattints az „Import” fülre, majd válaszd ki az adatbazisletrehozas.sql fájlt, kattints az „Importálás” gombra.

### 3.8.4 Swagger használata (Backend API tesztelése)

A **Swagger** egy webes felület az API-végpontok tesztelésére.

* Indítsd el a backend projektet Visual Studio-ból (F5 gomb vagy zöld „Run” gomb).
* Automatikusan megnyílik a Swagger felület böngészőben: https://localhost:5001/swagger.
* Itt láthatod és tesztelheted az összes API-végpontot.

### 3.8.5 DbDiagram.io használata (Adatbázis ER-diagram készítéséhez)

A **DbDiagram.io** weboldalt az adatbázis-terv elkészítéséhez használtuk.

* Nyisd meg ezt a weboldalt: [DbDiagram.io](https://dbdiagram.io/)
* Regisztrálj ingyenes fiókot.
* Hozz létre új diagramot, vagy importáld az SQL sémát.
* Készítsd el, módosítsd, exportáld az ER-diagramot (pl. PNG vagy PDF formátumban).

### 3.8.6 Figma használata (Felhasználói felület tervezéséhez)

A **Figma** egy online UI-tervező alkalmazás, amellyel megterveztük az oldal megjelenését.

* Nyisd meg a [Figma.com](https://figma.com) weboldalt.
* Regisztrálj ingyenesen, majd hozz létre új projektet.
* Készítsd el a felület terveit (wireframe-ek, mockupok).
* Exportáld vagy oszd meg a terveket a fejlesztőkkel.

### 3.8.7 Trello használata (Projektmenedzsmenthez)

A **Trello** egy egyszerű projektmenedzsment-eszköz.

* Nyisd meg: [Trello.com](https://trello.com)
* Regisztrálj ingyenesen.
* Hozz létre új táblát, adj hozzá listákat, kártyákat a feladatok kezeléséhez.
* Itt rendszerezd és kövesd nyomon a projekt feladatait.

### 3.8.8 Backend fájlok kezelése és parancsok (Visual Studio terminálban)

Nyisd meg a terminált (PowerShell vagy Command Prompt):

Navigálj a backend projekt mappájába:

cd Bulihub\_Backend

Telepítsd az Entity Framework eszközöket:

dotnet tool install --global dotnet-ef

Frissítsd az adatbázist az Entity Framework segítségével:

dotnet ef database update

Ez létrehozza vagy frissíti az adatbázis struktúráját az SQL szerveredben a Bulihub\_Backend projekt alapján.

### 3.8.9 Backend projekt indítása Visual Studioból

* Nyisd meg a backend megoldást (solution fájlt) Visual Studioban.
* Kattints a zöld „Run” gombra vagy nyomd meg az F5 billentyűt.
* Ez elindítja a backend szervert, a szolgáltatás automatikusan futni fog a háttérben.

# 4. Összefoglalás, köszönetnyilvánítás

## 4.1 A projekt értékelése

A BuliHub projekt egy hosszú és rendkívül tanulságos folyamat eredménye, amelynek során szakmailag és személyesen is jelentős tapasztalatokra tettem szert. A fejlesztés elején megfogalmazott szakmai célokat, a modern technológiák hatékony alkalmazását, valamint egy jól skálázható és könnyen karbantartható alkalmazás létrehozását sikerült teljes mértékben megvalósítani.

Szakmailag sikerült alaposan megismernem és alkalmaznom olyan széles körben elismert technológiákat, mint a React, TypeScript, .NET 7, SQL Server és a WPF. Ez a technológiai stack rendkívül erős alapot adott a projektnek, amely nemcsak a fejlesztést könnyítette meg, hanem a későbbi továbbfejlesztések és módosítások lehetőségét is biztosítja. Úgy érzem, hogy a szakmai fejlődésem szempontjából óriási előrelépést jelentett a projekt sikeres végigvitele.

A projektet értékelve kiemelkedő jelentőséget tulajdonítok annak, hogy olyan komplex feladatokat sikerült teljesítenem, mint például egyedi algoritmusok kidolgozása (JWT-alapú autentikáció, eseménykezelő rendszer, SMTP email küldési logika), valamint az adatbázis megfelelő megtervezése és implementálása. Ezen feladatok sikeres elvégzésével jelentősen bővült a szakmai tudásom, és magabiztosabbá váltam a jövőbeli fejlesztések terén is.

A program elkészítése során figyeltem arra is, hogy megfelelően kezeljem a hibás bemeneteket, robusztus és felhasználóbarát rendszert alakítsak ki. A tesztdokumentáció elkészítése során megtanultam, hogyan lehet különböző felhasználói viselkedéseket, nem ideális körülményeket és hibás használati eseteket megfelelően kezelni, és az ezekhez kapcsolódó hibajelzéseket világosan megfogalmazni.

A dokumentáció elkészítése önmagában is nagy kihívás volt, hiszen célom volt, hogy mindenki számára érthető, részletes, pontos útmutatót és leírást biztosítsak, amely szakmailag és felhasználói szemmel is megfelelő minőséget képvisel. Véleményem szerint sikerült ezt az elvárást is teljesíteni.

## 4.2 A program további hasznosulása, jövőbeli tervek

A BuliHub nemcsak egy vizsgafeladatként, hanem egy valós életben is hasznosuló alkalmazásként lett elkészítve. Úgy látom, hogy van valós piaci igény egy ilyen típusú platformra, ezért a vizsgát követően további terveim vannak az alkalmazással.

Elsődlegesen szeretném továbbfejleszteni a mobilalkalmazás irányába, amely lehetővé tenné, hogy még szélesebb körben elérhetővé váljon a felhasználók számára. A mobil app egy React Native alapú fejlesztéssel valósulhat meg, lehetőséget nyújtva Android és iOS platformokon való futtatásra. Ez nagyban növelheti az alkalmazás elérhetőségét és felhasználhatóságát, hiszen manapság egyre többen használják mobiltelefonjukat események keresésére és szervezésére.

Továbbá tervezem az online fizetési rendszerek integrációját, amely lehetőséget adna arra, hogy fizetős események kezelése is megvalósuljon a platformon belül. Ez egy olyan irány, amely jelentősen növelheti a platform gazdasági hasznosságát is.

Szeretnék még részletesebb analitikai eszközöket is integrálni a platformba, hogy a szervezők könnyebben és hatékonyabban tudják mérni eseményeik sikerességét. Ez egy további értékes funkció lehet a hitelesített felhasználók számára.

Úgy látom, hogy a platform jövője nagyon ígéretes, és az eddig elért eredmények megalapozták azt a bizalmat, hogy érdemes energiát és időt fordítani a további fejlesztésekre és piaci bevezetésére. Már most érzem, hogy a projekt komoly lehetőségeket rejt magában, és hosszabb távon is szeretném fenntartani, fejleszteni és továbbgondolni a BuliHub lehetőségeit.

## 4.3 Köszönetnyilvánítás

Ezúton szeretném kifejezni hálámat mindazoknak, akik segítettek, támogattak és inspiráltak a BuliHub projekt létrejöttében és sikeres végrehajtásában.

Először is köszönet illeti a mentoromat, aki szakmai tanácsaival, folyamatos támogatásával és iránymutatásával végig segítette a fejlesztési folyamatot. Nélküle nem lett volna lehetséges ilyen minőségben és hatékonysággal elkészíteni az alkalmazást.

Köszönöm a családomnak és barátaimnak is, akik végig mellettem álltak, türelmesek és támogatóak voltak, miközben hosszú órákat töltöttem a fejlesztéssel és teszteléssel. Különösen hálás vagyok a türelmükért és megértésükért, valamint azért, hogy folyamatosan bíztattak és motiváltak a nehezebb időszakokban is.

Végül köszönetet mondok minden tesztelőnek és segítőnek, akik visszajelzéseikkel, javaslataikkal, észrevételeikkel hozzájárultak a projekt sikeréhez. Az ő munkájuk kulcsfontosságú volt ahhoz, hogy a BuliHub valóban felhasználóbarát, jól működő és valós igényeket kielégítő alkalmazás lehessen.

# 5. Irodalomjegyzék

## 5.1 Felhasznált szakmai könyvek

Az alábbi könyvek és szakirodalmak voltak nagy segítségemre a BuliHub projekt fejlesztése során. Ezekből a forrásokból nemcsak alapos elméleti tudásra tettem szert, hanem a gyakorlatban is sikeresen alkalmaztam az ezekben ismertetett módszereket és technikákat.

1. **Adam Freeman: Pro ASP.NET Core 7**  
   Apress, 2023, ISBN: 978-1484286184  
   Ez a könyv átfogó képet ad az ASP.NET Core 7 keretrendszerről, segítséget nyújtott számomra a backend architektúra kialakításában, valamint a hatékony és biztonságos API fejlesztésben. A benne található gyakorlati példák és részletes magyarázatok elengedhetetlennek bizonyultak a fejlesztés során.
2. **Robin Wieruch: The Road to React with Firebase**  
   Independently Published, 2021, ISBN: 978-1720043997  
   Bár közvetlenül Firebase-t nem használtam, ez a könyv nagyon jól bemutatja a React alapú frontend fejlesztés alapjait, különösen a komponens-alapú tervezést, a hatékony állapotkezelést, és a TypeScript integrálását. Ebből a könyvből sikerült megismernem a React részletesebb működését, és több gyakorlati megoldást is átvettem belőle.
3. **Christian Nagel: Professional C# and .NET**  
   Wrox, 2022, ISBN: 978-1119797203  
   Ez a könyv segített mélyebben megérteni a .NET platform működését, különösen az Entity Framework Core ORM használatát, illetve az adatbázis-kezelési módszereket, melyek kulcsfontosságúak voltak a projekt adatkezelési részének fejlesztésekor.

## 5.2 Felhasznált weboldalak

Az alábbi weboldalak is kiemelkedő segítséget nyújtottak a fejlesztési folyamat során. Ezeken az oldalakon találtam példákat, magyarázatokat, és hasznos tanácsokat, melyek gyorsították és hatékonyabbá tették a fejlesztést.

**React hivatalos dokumentáció**  
Teljes mélységű URL: <https://react.dev>  
Ezen az oldalon pontos információkat találtam a React keretrendszer használatáról, állapotkezelési technikákról, valamint a komponensek életciklusáról. Gyakran nyúltam vissza hozzá, amikor részletes magyarázatra vagy példára volt szükségem.

**TypeScript dokumentáció**  
Teljes mélységű URL: https://www.typescriptlang.org/docs  
TypeScript fejlesztés során a hivatalos dokumentáció volt a legmegbízhatóbb forrásom, különösen a típusdefiníciók, generikusok és interface-ek használata kapcsán.

**Microsoft .NET dokumentáció**  
Teljes mélységű URL: <https://learn.microsoft.com/dotnet>  
A .NET dokumentáció átfogó, részletes információkat nyújtott számomra a keretrendszer architektúrájáról, az API-k létrehozásáról, az autentikáció megvalósításáról, és számos más fontos témakörről.

**Entity Framework Core dokumentációja**  
Teljes mélységű URL: <https://learn.microsoft.com/ef/core>  
A hivatalos dokumentáció rendkívül részletesen ismerteti az Entity Framework Core használatát, segítséget nyújtott adatbázis migrációk készítésében, a relációk kezelésében és a hatékony lekérdezések optimalizálásában.

**JWT.io - JSON Web Tokens**  
Teljes mélységű URL: https://jwt.io/introduction  
Ez az oldal kulcsfontosságú volt az autentikációs megoldásom elkészítésénél. Megismertem itt a JWT működését, biztonsági megfontolásokat, valamint kódpéldákat találtam, amelyek jelentősen könnyítették a JWT implementációját.

**Axios hivatalos dokumentáció**  
Teljes mélységű URL: https://axios-http.com/docs/intro  
Az Axios dokumentációja segített megérteni az API kommunikáció működését, hibakezelést és a hatékony HTTP-kérések létrehozását a frontend oldalon.

**Framer Motion dokumentáció**  
Teljes mélységű URL: https://www.framer.com/motion/  
Az animációk készítéséhez használt Framer Motion dokumentációja részletes példákkal és útmutatókkal támogatott, amelyekkel interaktív, látványos UI elemeket tudtam létrehozni.

**SMTP email küldés (Microsoft dokumentáció)**  
Teljes mélységű URL: <https://learn.microsoft.com/dotnet/api/system.net.mail.smtpclient>  
Az SMTP protokollal történő email-küldéshez használt Microsoft-dokumentáció pontos leírást és kódmintákat tartalmazott, ami nagyban megkönnyítette az automatikus email értesítések implementálását.

**Űrlap-elemek megvalósítása HTML-ben**  
Teljes mélységű URL: <http://infojegyzet.hu/webszerkesztes/html/urlapok/>  
Innen hasznos ötleteket és magyarázatokat vettem át az űrlapok kialakítására, validációjára és optimalizált kezelésére.

**Dinamikus képgaléria**  
Teljes mélységű URL: <http://infojegyzet.hu/webszerkesztes/php/kepgaleria/>  
A képgaléria kezeléséhez ötleteket merítettem ebből a cikkből, még ha konkrétan PHP-t nem is használtam, az itt ismertetett logikai megoldások hasznos inspirációt jelentettek számomra.

**Záródolgozat témák, szempontok, ötletek**  
Teljes mélységű URL: <http://infojegyzet.hu/webszerkesztes/zarodolgozat/>  
Ez az oldal segített nekem eligazodni a záródolgozat strukturálásában, tartalmi és formai követelményeiben, valamint számos hasznos tanácsot kaptam, amelyeket felhasználtam a dokumentáció elkészítése során.