**ENERGETIKAI TECHNIKUM ÉS KOLLÉGIUM**

**Vizsgaremek**

**Consumption Management System**

**2025.**

|  |  |
| --- | --- |
| **Szakma:** | **Készítették:** |
| **Szoftverfejlesztő és -tesztelő** | **Dobosi Gábor** |
| **5 0613 12 03** | **Mák Luca** |
|  | **Orbán Barnabás** |

# Tartalomjegyzék

[1 Tartalomjegyzék 2](#_Toc190942122)

[2 Bevezetés 4](#_Toc190942123)

[3 Témaválasztás indoklása 4](#_Toc190942124)

[3.1 Indoklás és aktualitás 4](#_Toc190942125)

[3.2 Kutatások, követelmények 5](#_Toc190942126)

[3.3 Eszközök és eljárások 5](#_Toc190942127)

[3.3.1 Általános fejlesztéshez használt eszközök 5](#_Toc190942128)

[3.3.2 Frontend fejlesztéshez használt eszközök 6](#_Toc190942129)

[3.3.3 Backendhez használt eszközök 7](#_Toc190942130)

[3.3.4 Adatbázis fejlesztéshez használt eszközök 8](#_Toc190942131)

Alulírott **Dobosi Gábor** büntetőjogi felelősségem teljes tudatában nyilatkozom arról, hogy az itt szereplő vizsgaremek csoportmunka eredménye és sem részeiben sem egészében nem került még kereskedelmi forgalomba, ill. publikálásra, a GPL licenszelésű programrészek kivételével.

Paks, 2025. április 18.

**Dobosi Gábor**

Alulírott **Mák Luca** büntetőjogi felelősségem teljes tudatában nyilatkozom arról, hogy az itt szereplő vizsgaremek csoportmunka eredménye és sem részeiben sem egészében nem került még kereskedelmi forgalomba, ill. publikálásra, a GPL licenszelésű programrészek kivételével.

Paks, 2025. április 18.

**Mák Luca**

Alulírott **Orbán Barnabás** büntetőjogi felelősségem teljes tudatában nyilatkozom arról, hogy az itt szereplő vizsgaremek csoportmunka eredménye és sem részeiben sem egészében nem került még kereskedelmi forgalomba, ill. publikálásra, a GPL licenszelésű programrészek kivételével.

Paks, 2025. április 18.

**Orbán Barnabás**

# Bevezetés

Csapatunk egy valós problémára keresett megoldást, amely egy olyan szoftver kifejlesztéséhez vezetett, mely szállodákban, fürdőkben és egyéb komplex szolgáltatóhelyeken alkalmazható. A program célja az volt, hogy megkönnyítse az intézmény dolgozóinak munkáját, miközben a vendégek számára kényelmesebb élményt biztosít. A rendszer egy helyre gyűjti a felhasználók költéseit, melyet egy ID vagy vonalkód segítségével tudnak beazonosítani az alkalmazottak.

Példának okáért vegyünk egy fürdőt amire a programunk is alapul, de kis változtatással bármely más szolgáltatást nyújtó területre könnyen átalakítható. A vendégek belépéskor kapott karszalagjukhoz kapcsolódva igénybe vehetik a különböző szolgáltatásokat anélkül, hogy azonnali fizetésre lenne szükség. Úgy oldható meg hogyha a karszalagjukra vannak terhelve az igénybe vett szolgáltatások.

Ez több szempontból is előnyös. Először is nem kell egész nap magukkal hordani a pénztárcájukat vagy bankkártyájukat, ami kényelmi és biztonsági szempontból is praktikus. Emellett nem kell minden igénybe vett szolgáltatásnál fizetni külön-külön, hanem egyszerre kényelmesen eltudják intézni a látogatásuk végén. Fizikai eszközök is rendelkezésünkre álltak így felhasználtuk azokat is a munkához, ezzel is szimulálva az éles környezetet.

# Témaválasztás indoklása

## Indoklás és aktualitás

A témaválasztás nem volt egyszerű. Szerettünk volna egy olyan alkalmazást létrehozni, ami kihívások elé állít nem csak a minimumot teljesíteni. Számos ötlet is felmerült, például webshop, Rubik kocka időmérő. A választásunk végül egy szolgáltatás kezelő rendszerre esett. Éles környezet minél jobb reprezentációja érdekében fizikai eszközöket is bevontunk a folyamatba, így került sor a karszalagok, az RFID olvasó és a vonalkódolvasó alkalmazására. Figyelembe véve azt, hogy intézményünk területén jelenleg is alkalmaznak hasonló eszközöket a beléptetéshez, a kezdeti tesztek ezeknek a segítségével mentek végbe.

Aktualitás szempontjából is megállja a helyét, mivel mindenki egyre jobban törekszik a gyorsaságra, kényelemre, egyszerűségre. A programunk pontosan ezt kívánja megvalósítani.

A fejlesztés során a legkorszerűbb technológiákra alapoztunk, így a projekt .NET 8-ra és a React keretrendszerre épül. Ezáltal nem csak a legújabb fejlesztéseket használtuk ki, de a dinamikus, API-központú webfejlesztés irányelveit is követtük, amely a jövőbeli bővíthetőség szempontjából is kiemelten fontos.

## Kutatások, követelmények

Első lépésként áttekintettük a vizsgakövetelményeket annak érdekében, hogy azonosítsuk a kötelező elemeket. Az alkalmazásnak vagy weboldalnak tartalmaznia kellett egy frontend részt, amely a megjelenítésért és az adatok betöltéséért felelős. Emellett szükség volt egy backend rendszerre is, amely az adatok tárolását, a jogosultságkezelést, az adatbázis-táblák és kapcsolatok pontos definiálását, valamint az API végpontok meghatározását biztosítja, lehetővé téve a frontend és a backend közötti kommunikációt.

A hatékonyság érdekében frontend oldalról React keretrendszert alkalmaztunk. A backend fejlesztéséhez az ASP.NET Core Web API technológiát választottuk, amely a C# programozási nyelven alapul.

A szoftver adatbázis MSSQL alapú, amely az Entity Framework segítségével könnyen kezelhető. A programunk megfelel az alapvető adatkezelési műveleteknek (CRUD), amely az adatok létrehozását (Create), lekérdezését (Read), módosítását (Update) és törlését (Delete) foglalja magában. Ezek a backend oldalon GET, POST, PUT és DELETE HTTP-metódusokként valósultak meg.

Igyekeztünk a tiszta kód elveit alkalmazni, amelyek elősegítik a fejlesztés hatékonyságát, az átláthatóságot, valamint a későbbi módosítások egyszerűbb elvégzését. Ennek érdekében például az egyértelmű változóneveket a PascalCase elnevezési konvenció szerint definiáltuk, és logikus, jól strukturált mappaszerkezetet alakítottunk ki.

## Eszközök és eljárások

### Általános fejlesztéshez használt eszközök

#### Trello

A Trello egy vizuális projektmenedzsment eszköz, amely Kanban-alapú táblák, listák és kártyák segítségével segíti a feladatok szervezését. Fejlesztőcsapatok számára kiválóan alkalmas sprint tervezésre, backlog kezelésre és feladatkövetésre. Integrálható GitHubbal, Slackkel és más fejlesztői eszközökkel.

#### GitHub

A GitHub egy népszerű forráskód-kezelő és verziókövető platform, amely a Git rendszeren alapul. Lehetővé teszi a fejlesztők számára a kód tárolását, verziózását, csapatmunkát és CI/CD (folyamatos integráció és telepítés) folyamatok kezelését. Nyílt forráskódú és privát projektekhez egyaránt használható.

### Frontend fejlesztéshez használt eszközök

#### Visual Studio Code:

A **VSCode** egy ingyenes, könnyű, mégis erőteljes kódszerkesztő, amelyet a Microsoft fejlesztett. Beépített támogatást nyújt JavaScript, TypeScript és React fejlesztéshez, valamint számos kiegészítő (pl. ESLint, Prettier) elérhető hozzá a fejlesztési élmény javítása érdekében.

#### Vite

A Vite egy modern fejlesztési szerver és build eszköz, amely gyorsabb fejlesztői élményt nyújt a React és TypeScript alapú alkalmazások számára. Használata lehetővé teszi a gyorsabb hot-reloadot és optimalizált végső kódot.

#### Material-UI (MUI)

A Material-UI egy népszerű React UI komponenskönyvtár, amely a Google Material Design elveit követi. Segítségével modern, reszponzív és könnyen testreszabható felhasználói felületeket lehet fejleszteni.

#### Axios

Az Axios egy ígéreteken (Promises) alapuló HTTP kliens, amelyet adatok lekérésére és küldésére használnak API-khoz. Különösen népszerű a React alkalmazásokban, mivel egyszerűsíti az aszinkron hálózati kéréseket és kezelést.

#### React & React-DOM

A React egy népszerű JavaScript könyvtár interaktív felhasználói felületek fejlesztésére. A React-DOM a React komponensek DOM-ba való rendereléséért felelős, biztosítva a hatékony frissítéseket és a komponens-alapú fejlesztési modellt.

#### TypeScript

A TypeScript egy JavaScript alapú programozási nyelv, amely statikus típusellenőrzést biztosít. A React alkalmazásokban használva segít a hibák csökkentésében és a kód karbantarthatóságának javításában.

#### ESLint

Az ESLint egy JavaScript és TypeScript kódellenőrző eszköz, amely segít fenntartani a kódminőséget, betartani a kódolási szabványokat és azonosítani a lehetséges hibákat már fejlesztés közben.

#### Prettier

A Prettier egy automatikus kódformázó eszköz JavaScript, TypeScript, HTML, CSS és sok más nyelv számára. Segítségével egységes kódstílust tarthatunk fenn a projektben, anélkül hogy manuálisan kellene figyelni a formázásra.

#### Tailwind CSS

A Tailwind CSS egy utility-first megközelítésű CSS keretrendszer, amely előre definiált osztályokat biztosít a gyors és hatékony stílusozáshoz. Ahelyett, hogy kész komponenseket adna (mint pl. Bootstrap vagy Material-UI), kis, önálló osztályokat kínál, amelyekkel teljes mértékben testreszabható a dizájn.

#### Whimsical

A Whimsical egy vizuális tervezőeszköz, amely támogatja diagramok, drótvázak, folyamatábrák és gondolattérképek készítését. Ideális UX/UI tervezéshez, rendszerarchitektúrák megjelenítéséhez és csapatmunkában történő ötleteléshez.

### Backendhez használt eszközök

#### Visual Studio 2022:

A Visual Studio 2022 egy fejlett fejlesztőkörnyezet, amely támogatja az ASP.NET Web API fejlesztését C# nyelven. Beépített NuGet csomagkezelővel könnyedén hozzáadhatók és frissíthetők a szükséges csomagok, például Entity Framework Core, Swashbuckle (Swagger) és más API-fejlesztést segítő könyvtárak.

#### Általunk alkalmazott NuGet csomagok:

* Microsoft.EntityFrameworkCore (by Microsoft) – 9.0.2:
  + Entity Framework Core egy modern objektum-relációs adatbázis leképező .NET-hez, támogatja a LINQ lekérdezéseket, módosításokat, sémamigrációkat.
* Microsoft.EntityFrameworkCore.Design (by Microsoft) – 9.0.2
  + Az Entity Framework Core eszközeihez szükséges fejlesztési időben használt komponensek.
* Microsoft.EntityFrameworkCore.SqlServer (by Microsoft) – 9.0.2
  + MSSQL Server adatbázis kezelő Entity Framework Core-hoz.
* Microsoft.EntityFrameworkCore.Tools (by Microsoft) – 9.0.2
  + Az Entity Framework Core eszközeihez szükséges NuGet csomag.
* Microsoft.VisualStudio.Web.CodeGeneration.Design (by Microsoft) – 8.0.7 / 9.0.0:
  + Kódgenerálás az ASP.NET Core-hoz, tartalmazza a dotnet-aspnet-codegenerator parancsot vezérlők és nézetek generálásához.
* Swashbuckle.AspNetCore (by domaindrivendev) – 6.6.2 / 7.2.0
  + Kizárólag fejlesztői környezethez használt Swagger eszköz, mely képes a végpontok tesztelésére.

#### Postman

A Postman egy népszerű API-fejlesztő eszköz, amely lehetővé teszi HTTP-kérések küldését, API-k tesztelését és dokumentálását, támogatva az OAuth, JWT és egyéb hitelesítési módszereket.

### Adatbázis fejlesztéshez használt eszközök

#### SQLEXPRESS

A Microsoft SQL Server ingyenes, könnyített verziója, amely fejlesztési célokra ideális. Jól integrálható ASP.NET Web API alkalmazásokkal az Entity Framework Core használatával.

#### SQL Server Management Studio

Grafikus felhasználói felületet biztosító eszköz a Microsoft SQL Server kezelésére. Lehetővé teszi az adatbázisok kezelését, lekérdezések futtatását, tárolt eljárások létrehozását.

#### Dbeaver

Ingyenes, többplatformos adatbázis-kezelő eszköz, amely támogatja a SQL Server, MySQL, PostgreSQL, SQLite és más adatbázisokat, vizuális szerkesztővel és lekérdezési funkciókkal.

#### drawDB

A drawDB egy adatbázis-tervező eszköz, amely lehetővé teszi az ER-diagramok (Entity-Relationship) és adatbázisstruktúrák vizuális modellezését. Segítségével könnyedén megtervezhető egy rendszer adatmodellje, amely később SQL-kóddá alakítható.