



VAS VÁRMEGYEI SZAKKÉPZÉSI CENTRUM  
HORVÁTH BOLDIZSÁR  
KÖZGAZDASÁGI ÉS INFORMATIKAI  
TECHNIKUM

A 5 0613 12 03 számú Szoftverfejlesztő és –tesztelő vizsgaremek

**A BüféBox  
dokumentációja**

Készítették:

TAUBER MÁTYÁS  
PATAKI ANDRÁS  
NÉMETH TAMÁS

**SZOMBATHELY**

**2025**

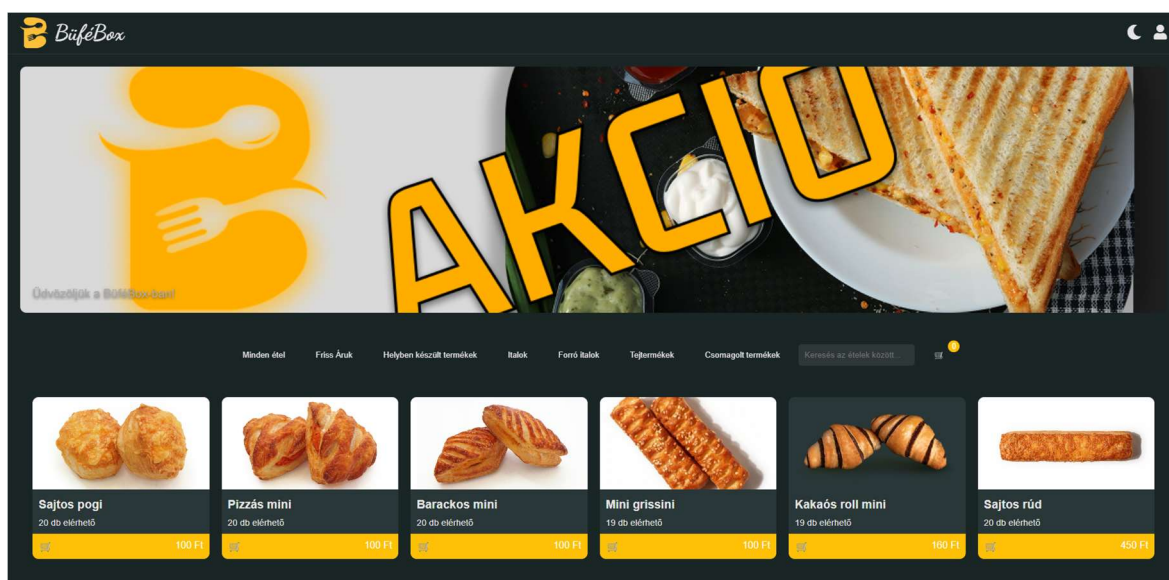
## Bevezetés

A digitális világ rohamos fejlődése megköveteli, hogy a mindennapi élet különböző területein – így például az oktatási intézményekben – is modern, hatékony megoldásokat alkalmazzunk. Egyik gyakran előforduló probléma az iskolai büfék működésében a szünetekben kialakuló hosszú sorok, a kiszolgálás lassúsága, valamint a készlet- és rendeléskezelés átláthatatlansága.

Projektünk, a BüféBox, erre a problémára kínál megoldást. A rendszer célja, hogy digitalizálja és egyszerűsítse az iskolai büfében történő rendeléseket, miközben átlátható adminisztrációs felületet biztosít az üzemeltetők számára.

### A rendszer két fő komponensből áll:

Egy webes felületből, ahol a tanulók és tanárok saját fiókokkal bejelentkezve rendelhetnek ételeket és italokat,



*Weboldal kinézete*

Valamint egy asztali adminisztrációs alkalmazásból, amely lehetővé teszi a rendelések kezelését, a készletek nyilvántartását és a felhasználói fiókok menedzselését.



*WPF Dolgozói bejelentkezés*

A projektet háromfős fejlesztőcsapat készítette: Tauber Mátyás, Pataki András és Németh Tamás, a Horváth Boldizsár Közgazdasági és Informatikai Technikumban.

A dokumentáció célja, hogy bemutassa a szoftver felépítését, működését, valamint annak műszaki és felhasználói aspektusait. A fejlesztés során külön figyelmet fordítottunk arra, hogy a rendszer egyszerre legyen felhasználóbarát, biztonságos, valamint könnyen továbbfejleszthető a jövőben.

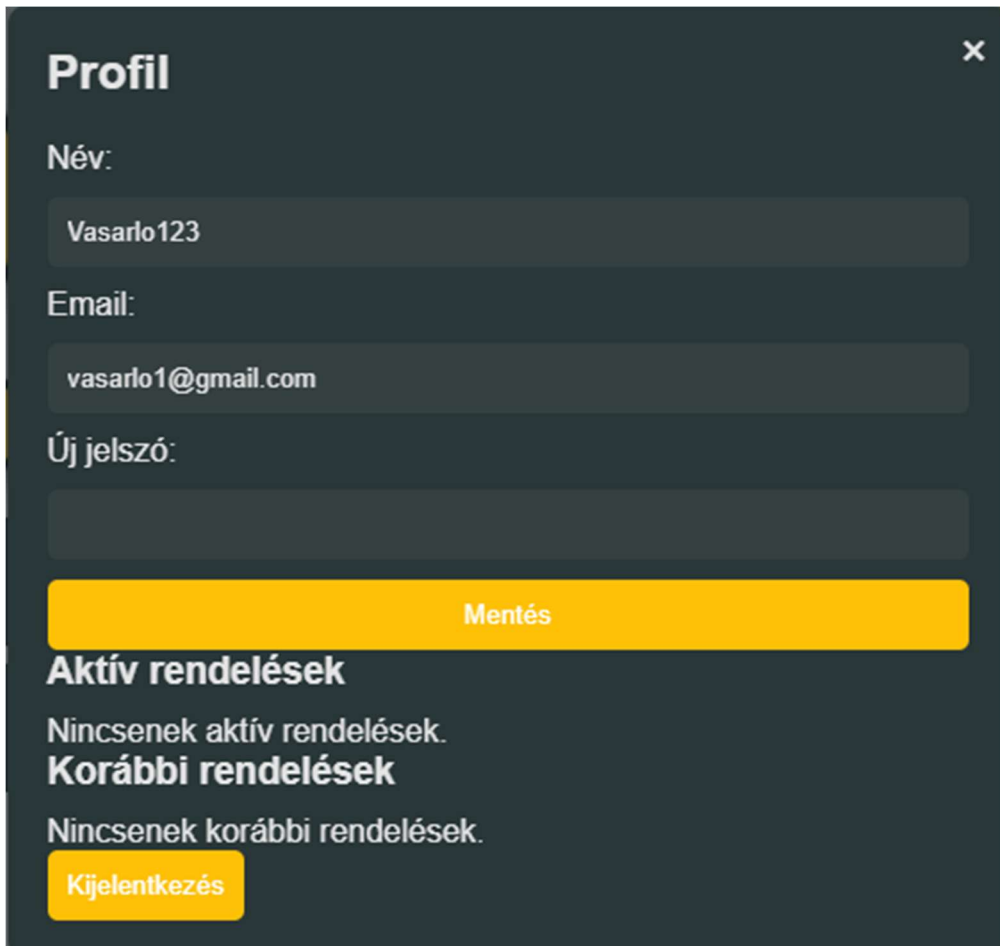
## Szoftver célját

A BüféBox szoftver célja, hogy egy modern, digitális rendelési és adminisztrációs rendszert biztosítson az iskolai büfé működéséhez. A fejlesztés kiindulópontja az volt, hogy a tanulók és tanárok ne legyenek rákényszerítve a hosszú sorban állásra, hanem kényelmesen, előre leadhassák rendeléseiket egy online felületen keresztül.

### A rendszer célkitűzései:

#### Felhasználói oldalról:

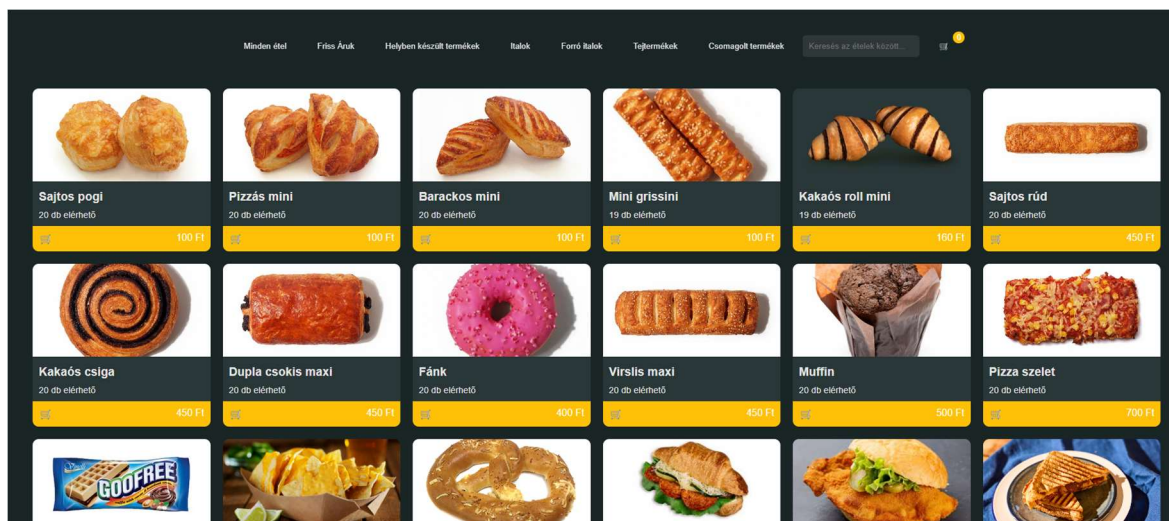
- Egyszerű és gyors online rendelési lehetőség,
- Felhasználói fiók létrehozása és kezelése,
- Elérhető termékek áttekintése,
- Rendelési előzmények megtekintése.



The screenshot shows a dark-themed modal window titled "Profil" with a close button (X) in the top right corner. The form contains the following elements:

- Név:** A text input field containing "Vasarlo123".
- Email:** A text input field containing "vasarlo1@gmail.com".
- Új jelszó:** An empty text input field.
- Mentés:** A prominent yellow button.
- Aktív rendelések:** A section header followed by the text "Nincsenek aktív rendelések."
- Korábbi rendelések:** A section header followed by the text "Nincsenek korábbi rendelések."
- Kijelentkezés:** A yellow button at the bottom left.

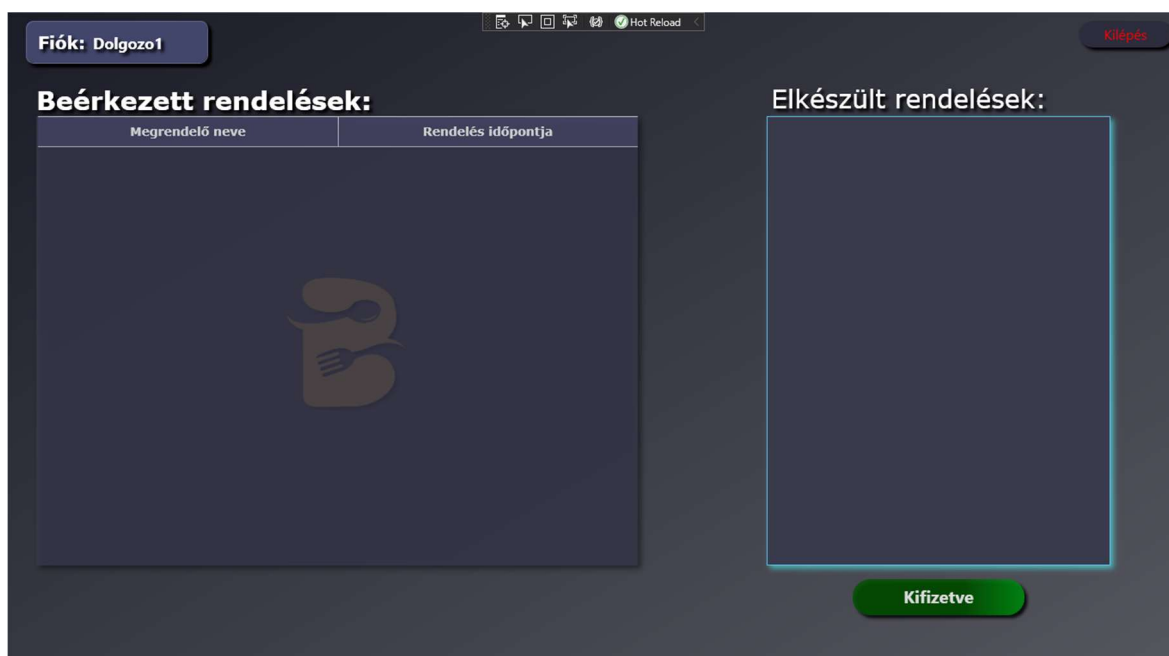
*Weboldal profiladatok módosítása*



Weboldal terméklista

## Adminisztrátori oldalról:

- Rendelések valós idejű fogadása és státuszkezelése,
- Raktárkészlet nyilvántartása és frissítése,
- Terméklista karbantartása (új termékek hozzáadása, módosítása, törlése),



WPF rendeléskezelés

Fiók: Dolgozó1

Hot Reload

Kilépés

Termékek listája:

Termék neve:

Termék kódja	Termék neve	Mennyiség
1	Sajtos pogi	20
2	Pizzás mini	20
3	Barackos mini	20
4	Mini grissini	19
5	Kakaós roll mini	19
6	Sajtos rúd	20
7	Kakaós csiga	20
8	Dupla csokis maxi	20
9	Fánk	20
10	Virslis maxi	20
11	Muffin	20

Feltöltés

Módosítás

Törlés

Termék felvétel/módosítás:

Termék neve:

Darabszám:

Kiszerelés:

Ar:

Kategóriák:

Termék képe:

Hozzáadás

WPF raktárkezelés

A szoftver egyaránt támogatja a tanulók és a büfé dolgozók igényeit: míg az előbbieket kényelmesen, gyorsan és sorban állás nélkül tudnak vásárolni, addig az utóbbiak hatékonyabban és strukturáltabban végezhetik a munkájukat.

A rendszer ezen túlmenően könnyen bővíthető, így később akár mobilalkalmazással, QR-kódos azonosítással vagy online fizetési lehetőséggel is kiegészíthető.

## Komponenseinek technikai leírása

A BüféBox rendszer több különálló, de egymással szorosan együttműködő komponensből épül fel. A rendszer két fő felhasználói csoportot szolgál ki: a diákokat/tanárokat, akik a webes rendelőfelületet használják, valamint az adminisztrátorokat, akik az asztali alkalmazáson keresztül végzik az üzemeltetési feladatokat. A háttérben közös adatbázis és szerveroldali logika biztosítja az adatok szinkronizálását és feldolgozását.

### 1. Webes rendelőfelület (Frontend)

A felhasználók ezen a felületen keresztül tudják megtekinteni a termékeket, fiókot létrehozni, bejelentkezni, valamint rendeléseket leadni. A felület letisztult, reszponzív és egyszerűen kezelhető, így ideális diákok és tanárok számára is.

#### Funkciók:

- Termékek listázása
- Keresés és szűrés
- Rendelés leadása
- Saját rendelések megtekintése
- Bejelentkezés / regisztráció

### 2. Szerveroldali logika (Backend API)

Technológia: Node.js

A backend felelős az üzleti logika kiszolgálásáért, az adatbázis műveletek végrehajtásáért és az API végpontok biztosításáért, amelyeket a frontend és az asztali alkalmazás is használ. Ezeket Swashbuckle és EntityFramework kiegészítő csomagokkal segítettük.

#### Főbb végpontok:

- Felhasználói regisztráció / bejelentkezés
- Termékek lekérdezése / módosítása
- Rendelések kezelése (leadás, státusz frissítés)
- Raktárkészlet kezelése

### 3. Asztali adminisztrációs alkalmazás

Technológia: C# (WPF)

Az iskolai büfé adminisztrátorai ezen a felületen kezelhetik a rendeléseket, láthatják azok státuszát, módosíthatják a készletet és kezelhetik a termékeket.

**Funkciók:**

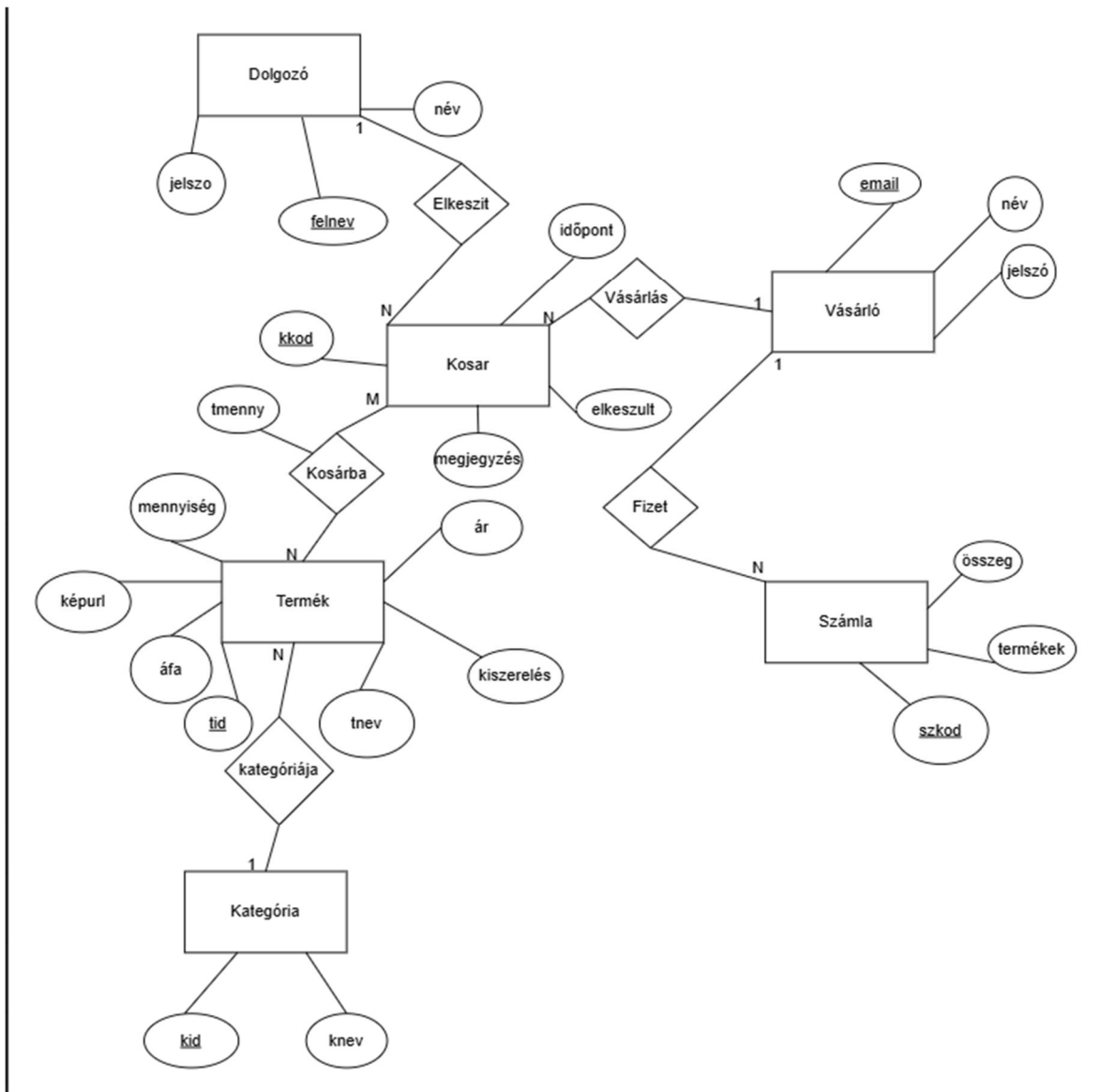
- Rendelések valós idejű fogadása és státusz frissítése (pl. „Folyamatban”, „Elkészült”)
- Terméklista karbantartása
- Raktárkészlet nyomon követése

**4. Adatbázis**

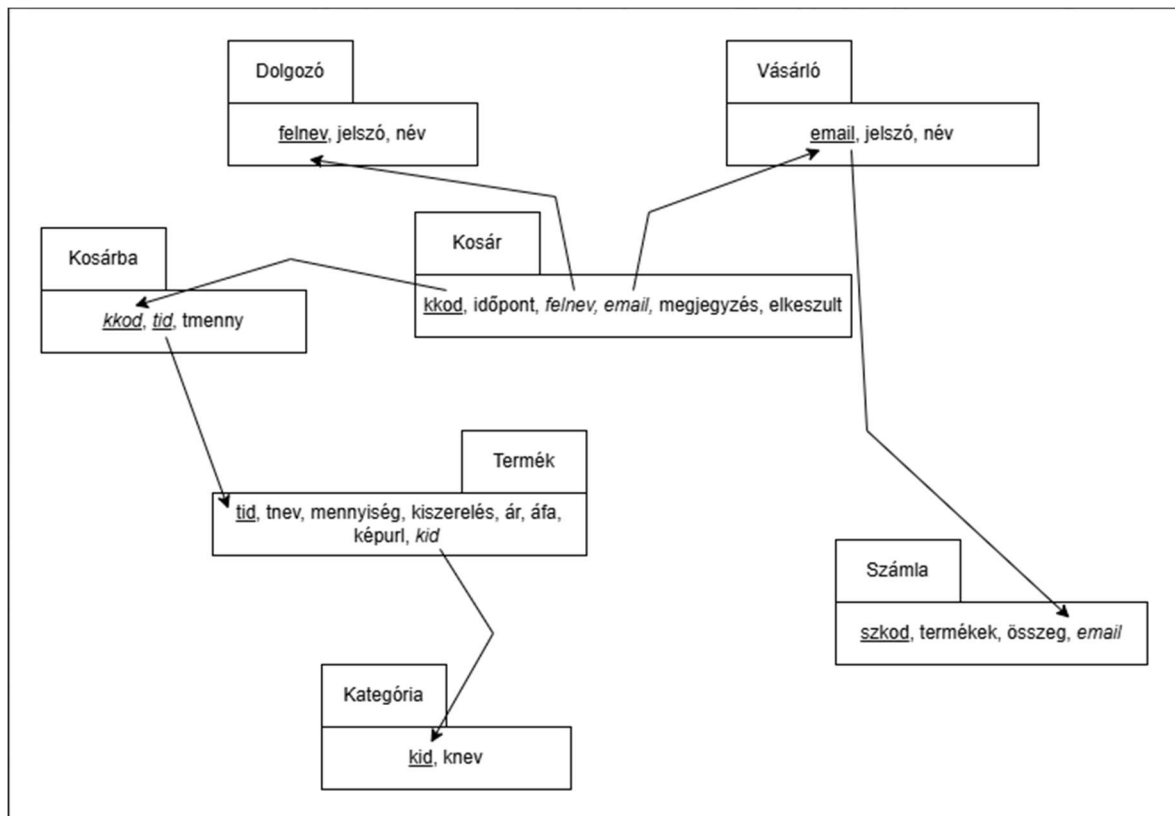
Technológia: MySQL, phpMyAdmin kezelőfelülettel

Az adatbázis tárolja a felhasználók, termékek, rendelések, jogosultságok és készletek adatait. Strukturált, normalizált adatbázistervet használunk a redundancia csökkentése és az adatok konzisztenciájának biztosítása érdekében.





ER Modell ábrája



Relációs Modell ábrája

## 5. Verziókezelés és kollaboráció

Technológia: Discord, GitHub

A csapat tagjai leginkább Discord segítségével közösen dolgoztak a projekten, nyomon követve a verziókat, hibákat és feltöltve az újdonságokat. A GitHub felületét használva publikáltuk projektünket.

## Működésének műszaki feltételei

A BüféBox rendszer működéséhez többféle eszköz és szoftverkörnyezet szükséges, mivel egy összetett, több komponensből álló alkalmazásról van szó. Az alábbiakban részletesen bemutatjuk, hogy milyen hardveres és szoftveres feltételekkel lehet a rendszert hatékonyan és megbízhatóan működtetni.

### 1. Felhasználói webalkalmazás (frontend)

**Minimális rendszerkövetelmények a kliensoldalon (pl. számítógép / telefon):**

- Modern internetböngésző (pl. Google Chrome, Mozilla Firefox, Microsoft Edge, Safari)
- Stabil internetkapcsolat

- Képernyőfelbontás: legalább 1024x768, mobilra reszponzív

#### **Ajánlott eszközök:**

- Windows, macOS vagy Linux operációs rendszerű számítógép, tablet vagy okostelefon
- RAM: 2 GB vagy több
- Processzor: legalább 1.6 GHz (dual-core)

## **2. Adminisztrációs asztali alkalmazás (WPF asztali app)**

#### **Minimális rendszerkövetelmények:**

- Operációs rendszer: Windows 10 vagy újabb
- .NET Framework 4.8 telepítve
- RAM: 4 GB
- Processzor: Intel i3 vagy AMD Ryzen 3 szintű teljesítmény
- Szabad tárhely: legalább 500 MB az alkalmazás telepítéséhez

#### **Ajánlott környezet:**

- Windows 10 Pro 64-bit
- Legalább Full HD kijelző (1920x1080)
- 8 GB RAM, SSD háttértár a gyorsabb működéshez

## **3. Backend szerver és adatbázis (üzemeltetői oldal)**

#### **Szoftverigények:**

- Node.js (ajánlott verzió: LTS, pl. 18.x)
- MySQL szerver (pl. 8.x verzió)
- phpMyAdmin a vizuális adatbáziskezeléshez
- Git és GitHub CLI a verziókezeléshez
- Szükség esetén PM2 vagy Docker a backend futtatásához stabil környezetben

#### **Hardverigény (szerver oldalon, ha lokálisan futtatjuk):**

- Operációs rendszer: Windows Server / Linux (Ubuntu 20.04+)
- RAM: 4-8 GB

- CPU: 2 magos processzor (legalább 2.0 GHz)
- Stabil hálózati kapcsolat

#### **Alternatív üzemeltetési lehetőség:**

A rendszer később áthelyezhető felhőbe (pl. AWS, Heroku, Railway.app), amely lehetővé teszi a skálázást, a magas rendelkezésre állást és távoli elérést.

#### **4. Fejlesztői környezet**

- Visual Studio 2019 – C# WPF alkalmazás fejlesztéséhez
- Visual Studio Code – Webes frontend és backend fejlesztéshez
- Node.js + npm – Backend működtetéséhez
- MySQL Workbench / phpMyAdmin – Adatbázis modellezés és kezelés
- GitHub Desktop vagy CLI – Verziókövetéshez

## Használatának rövid bemutatása

A BüféBox rendszer használata egyszerű, felhasználóbarát kialakítása miatt diákok, tanárok és adminisztrátorok számára is gyorsan megtanulható és hatékonyan használható. A szoftver két fő részre tagolódik: a webes rendelőfelületre (frontend) és az adminisztrációs asztali alkalmazásra (desktop admin).

Ahhoz hogy futtathassuk a weboldalt el kell indítanunk a XAMPP kezelőpanelen az adatbázis szerveret. Majd az sql fájlt importálva, ami magától települ, már fel is áll az adatbázisrendszer. Ez után indíthatjuk a back-end rendszert, ami összeköti a különböző részét a projektnek. Majd végül elindíthatjuk a Visual Studio Code-ban a weboldalt (ha localhost-on futtatjuk természetesen.) Ehhez egy Live Server kiegészítőt kell telepítenünk a kódszerkesztőhöz, amivel egyszerűbben tudunk futtatni webes oldalakat. Majd egy új terminált nyitva, miután a megfelelő mappát megnyitottuk amiben van a weboldal, beírjuk hogy `npx -serve`. és megnyílik a weboldal kezelőfelülete, ahol a fooldal.html oldalt megnyitva használhatjuk is a weboldalt. Majd ha pedig a dolgozói felületet szeretnénk ugyan úgy szükségünk van az sql szerver és a back-end rendszer futására és szimplán elindítjuk a Visual Studio 2019-ben a WPF alkalmazást.

## **1. Webes rendelőfelület használata (diákok és tanárok)**

A felhasználók a rendszer webes oldalát egy böngészőn keresztül érik el. A kezdőlap on egy letisztult, könnyen átlátható felület fogadja őket, ahol regisztrálni vagy bejelentkezni tudnak.

### **Használat menete:**

- Regisztráció / Bejelentkezés:
  - A felhasználó megadja nevét, e-mail címét, jelszavát és osztályát.
- A belépés után személyre szabott fiókba kerül.
- Termékek böngészése és keresés:
  - A termékek kategóriák szerint (pl. szendvicsek, italok, édességek) listázva jelennek meg.
- Beépített keresőmező és szűrők segítik a könnyebb eligazodást.
- Rendelés leadása:
  - A felhasználó kiválasztja a kívánt termékeket, mennyiséget, megjegyzést majd leadja a rendelést.
- A rendelés státusza (pl. „Feldolgozás alatt”, „Elkészült”) a fiókjában követhető.
- Rendelés átvétele:
  - A felhasználó az alkalmazásban látja, mikor készül el a rendelése, így csak akkor megy a büféhez, amikor valóban szükséges – ezzel elkerülve a sorban állást.

## **2. Asztali admin alkalmazás használata (adminisztrátorok)**

A büfé dolgozói vagy rendszergazdák egy külön, telepíthető Windows-os alkalmazást használnak. Az admin felület letisztult, és az egyes funkciók logikusan, külön szekciókban érhetők el.

### **Használat menete:**

- Bejelentkezés admin fiókkal
  - Az admin belépése után azonnal látja a beérkezett rendelések listáját.
- Rendelések kezelése
  - Az új rendelések időrendi sorrendben jelennek meg.
  - A dolgozó frissítheti a rendelés státuszát („Folyamatban”, „Elkészült”, „Átadva”).
- Raktárkészlet frissítése

- Minden termékhez rendelhető aktuális készlet. Az alkalmazás automatikusan csökkenti a készletet a rendelések után.
- Ha valamelyik termék elfogy, az automatikusan „nem rendelhetőként” jelenik meg a webes felületen.
- Termékek és árak kezelése
  - Új termékek hozzáadása, meglévők szerkesztése vagy eltávolítása néhány kattintással megoldható.

### **3. Admin és felhasználó közötti szinkron**

A két rendszer között az adatokat egy közös adatbázis és egy REST API biztosítja. Így minden módosítás – legyen szó rendelésről, termék áráról vagy készletről – azonnal megjelenik mindkét oldalon, biztosítva az élő, valós idejű működést.

## **Bejelentkezés**

A bejelentkezés funkció mind a webes, mind az adminisztrációs felületen központi szerepet tölt be. A rendszer zárt, csak érvényes felhasználói fiókkal lehet elérni a funkciókat, így biztosítható a jogosultságok szerinti hozzáférés és az adatok védelme.

### **Webes felület – Felhasználói bejelentkezés:**

- A kezdőlap on található profil ikon után a „Bejelentkezés” gombra kattintva jelenik meg a bejelentkezési űrlap.
- A felhasználónak meg kell adnia a regisztráció során használt e-mail címet és jelszót.
- Sikeres hitelesítés után a felhasználó beléphet a saját fiókjába, ahol:
- böngészheti a termékeket, leadhat új rendeléseket, valamint módosíthatja fiókadatait.
- Hibás adatok megadása esetén a rendszer figyelmeztető üzenetet jelenít meg.

### **Admin alkalmazás – Adminisztrátori bejelentkezés:**

- Az asztali alkalmazás indításakor egy bejelentkezési ablak fogadja az adminisztrátorokat.
- Itt felhasználónév és jelszó megadása szükséges.
- A sikeres bejelentkezés után az admin felület főképernyőjére jut, ahol az alábbi modulokat érheti el:
- Rendelések kezelése
- Termékek és árak kezelése
- Raktárkészlet nyilvántartás

### **Biztonsági megfontolások:**

- A jelszavak titkosított formában (hash-elve) kerülnek eltárolásra az adatbázisban.
- A bejelentkezéshez szerveroldali hitelesítési logika társul, amely megakadályozza a jogosulatlan hozzáférést.
- Az admin felület elérését csak magasabb jogosultsági szintű fiókok tehetik meg.
- A rendszer a bejelentkezési folyamat során figyelembe veszi a biztonság és a felhasználói élmény közötti egyensúlyt: gyors belépést biztosít, miközben az adatok biztonságát is garantálja.

## Tesztelések

### BackEnd rész tesztelése:

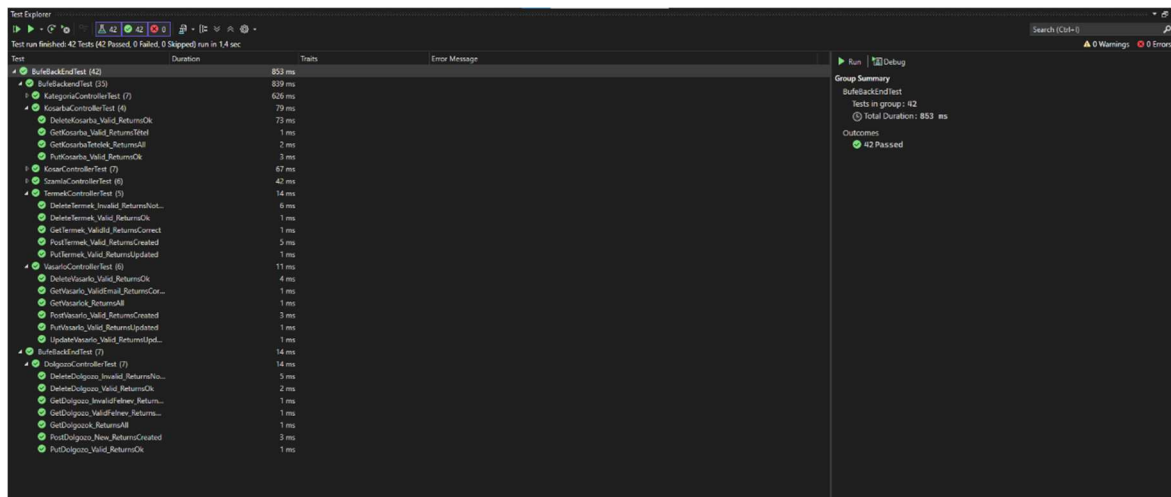
A BufoBackEndTest projekt a BufoBackEnd alkalmazás különböző részeinek egységtesztelését végzi. Az egységtesztnek célja, hogy külön-külön ellenőrizzük az egyes controllerek, metódusok helyes működését, függetlenül a valódi adatbázistól.

#### A tesztek a következő technikákat és eszközöket alkalmazzák:

**MSTest:** a Microsoft hivatalos egységteszt keretrendszere ([TestClass], [TestMethod] attribútumokkal).

**Moq:** egy népszerű mock-keretrendszer, amely lehetővé teszi az adatbázis és más függőségek szimulálását.

**Mockolt DbContext és DbSet:** nem igazi adatbázisra dolgozunk, hanem virtuális (mockolt) adatokat adunk vissza a controllereknek.



*BackEnd kód tesztelésének sikeressége*

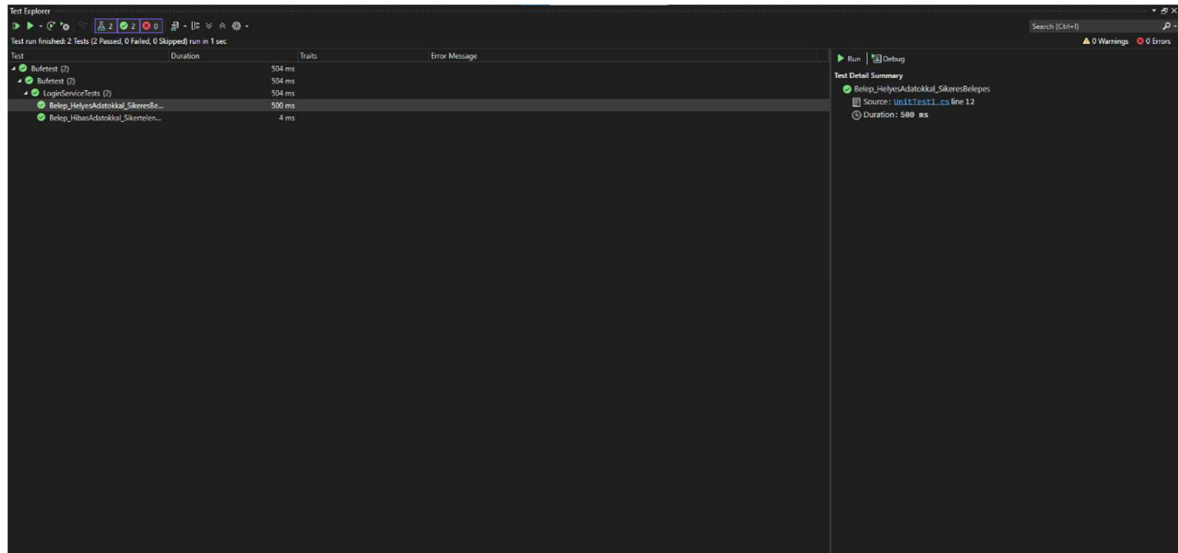
### WPF Alkalmazás tesztelése:



A projekt egy WPF alapú rendeléskezelő alkalmazás, amelyhez MSTest keretrendszerrel írt automatikus egységteszteket tartalmaz a Bufetest projekt. A tesztek célja, hogy ellenőrizzék a programban használt osztályok és metódusok helyes működését.

### Teszteléshez használt technológia:

MSTest (Microsoft.VisualStudio.TestTools.UnitTesting)



*WPF kód tesztelésének sikeressége*

## Továbbfejlesztési lehetőségek

A BüféBox rendszer jelenlegi formájában már egy teljes körű, működőképes megoldást nyújt az iskolai büfék digitális rendelésének és adminisztrációjának támogatására. Ugyanakkor a rendszer fejlesztése során nagy hangsúlyt fektettünk a jövőbeli bővíthetőségre és moduláris felépítésre, így több irányban is lehetséges további funkciók beépítése, fejlesztési lépések megtétele.

**Az alábbiakban bemutatjuk a legfontosabb és legrealisabb továbbfejlesztési lehetőségeket:**

### 1. Online fizetés integrálása

Jelenleg a rendszer csak rendelést fogad, a fizetés az átvételkor történik. A jövőben lehetőség lenne bankkártyás, mobilos (pl. SimplePay) vagy iskolai egyenlegalapú fizetési rendszer beépítésére is. Ez gyorsítaná az átvételt, és csökkentené a készpénzes fizetésből adódó hibalehetőségeket.

### 2. Mobilalkalmazás fejlesztése

A jelenlegi webes felület mobilbarát, de natív mobilalkalmazás fejlesztése (Android/iOS) még kényelmesebbé tenné a rendelést, például push értesítésekkel („A rendelésed elkészült!”), QR-kódos átvétellel vagy offline hozzáféréssel.

### 3. Admin jogosultsági szintek finomítása

A jelenlegi rendszerben az admin szerepkör egységes. A jövőben érdemes lenne bevezetni részletesebb jogosultságokat:

- Konyhai személyzet: csak a rendeléseket látja és készíti el,
- Raktáros: csak készletet kezel,
- Rendszergazda: minden modulhoz teljes hozzáférés.

### 4. QR-kódos átvétel és azonosítás

A rendelés leadása után a rendszer QR-kódot generálhatna, amelyet a diák átvételkor bemutat, így gyorsabb és pontosabb rendelésazonosítás válik lehetővé a büfénél.

### 5. Termékértékelések, visszajelzések

A felhasználók visszajelzést adhatnának az egyes termékekről, segítve ezzel a kínálat finomhangolását, és ösztönözve a minőség fenntartását.

## **6. Statisztikai modulok bővítése**

- Az admin felületen részletesebb statisztikák jeleníthetők meg:
- napi/heti/havi forgalom,
- legnépszerűbb termékek,
- egyes osztályok fogyasztási szokásai.

## **7. Felhőalapú hosztolás**

A rendszer jelenleg helyi szerveren futtatható, de a jövőben akár felhőszolgáltatásokra is áthelyezhető (pl. AWS, Azure, Railway.app), így bármikor, bárhonnán hozzáférhető lenne, stabilabb és biztonságosabb környezetet biztosítva.

# Tartalom

Bevezetés .....	1
Szoftver célját .....	3
Komponenseinek technikai leírása .....	6
Működésének műszaki feltételei .....	9
Használatának rövid bemutatása .....	12
1. Webes rendelőfelület használata (diákok és tanárok).....	13
Használat menete:.....	13
2. Asztali admin alkalmazás használata (adminisztrátorok).....	13
Használat menete:.....	13
• Raktárkészlet frissítése .....	13
• Termékek és árak kezelése .....	14
3. Admin és felhasználó közötti szinkron.....	14
Tesztelések	
Továbbfejlesztési lehetőségek .....	17