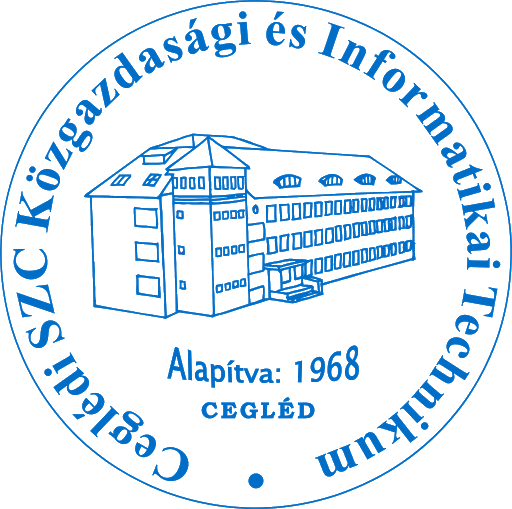
Ceglédi SZC Közgazdasági és Informatikai Technikum



# Dokumentáció a „Döner Cegléd” asztali alkalmazásához

2025

Jakab Botond, Krausz Márton Alajos, Tóth Dániel János

13.c

# Tartalom

Tartalom

[Dokumentáció a „Döner Cegléd” asztali alkalmazásához 1](#_Toc1148923290)

[Tartalom 1](#_Toc1048318116)

[Feladat/cél 2](#_Toc618492552)

[Rendelések megjelenítése és kiadása 3](#_Toc1975049771)

[Regisztrált felhasználók kezelése 3](#_Toc1356152423)

[Új termékek felvitele 3](#_Toc34923684)

[Már meglévő termékek teljes körű szerkesztése 3](#_Toc1250780695)

[Felület minta 3](#_Toc1492703303)

[Használt programok, fejlesztői környezetek és futtatási környezet 4](#_Toc1490946365)

[Használt programok: 5](#_Toc1486382561)

[Használt fejlesztői környezetek 6](#_Toc1221916098)

[https://visualstudio.microsoft.com/downloads/ 7](#_Toc137610586)

[Futtatási környezet és hardware 7](#_Toc661481810)

[A program használata 8](#_Toc1609367052)

[A program indítása 8](#_Toc1349310469)

[A rendelések megjelenítése és kiadása oldal 8](#_Toc403053701)

[Sima rendelés 8](#_Toc1632818738)

[Elvitelre szánt rendelés 9](#_Toc338404444)

[Rendelés kiadás 9](#_Toc1903151214)

[Admin bejelentkezés 9](#_Toc187430979)

[Regisztrált felhasználók kezelése oldal 11](#_Toc664581025)

[Új termékek felvitele oldal 12](#_Toc1537861813)

[A már meglévő termékek teljes körű szerkesztése oldal 14](#_Toc1751005264)

[Dokumentáció a backend tesztekről 16](#_Toc2003569359)

[Környezetleírás 16](#_Toc390464823)

[A fájl felépítése 16](#_Toc1950117225)

[Használat 17](#_Toc930496626)

[Fontosabb tesztek bemutatása 17](#_Toc1449468086)

[Middlewares tesztek 17](#_Toc719989304)

[Error Handler tesztek 17](#_Toc1971799414)

[Repository tesztek 17](#_Toc1908868073)

[User Repository tesztek 18](#_Toc767272831)

[Dish Repository tesztek 18](#_Toc903896726)

[Controller tesztek 19](#_Toc1966874585)

[User Controller tesztek 19](#_Toc1765700521)

[Dish Controller 20](#_Toc1061680140)

[Források 20](#_Toc60464253)

# Feladat/cél

Az alkalmazás célja az éttermi működés hatékonyabbá tétele, az adatok gyors elérése és módosítása, valamint a felhasználók és rendelések egyszerű kezelése. A felhasználóbarát felület és a valós idejű kommunikáció biztosítja a gördülékeny munkafolyamatokat az étterem minden területén.

## Rendelések megjelenítése és kiadása

* Az alkalmazás valós időben jeleníti meg az étteremhez beérkező rendeléseket.
* Lehetőség van a rendelések részletes megtekintésére, beleértve a termékek listáját, a mennyiségeket, az árakat és az esetleges megjegyzéseket.
* A rendelés kiadása után az állapot frissíthető az API-n keresztül.

## Regisztrált felhasználók kezelése

* Az alkalmazás lehetőséget biztosít a regisztrált felhasználók kezelésére.
* A felhasználók deaktiválhatók, ha már nem aktívak az étteremben.
* A felhasználói adatok, például a név, e-mail cím, szerepkör vagy jogosultságok szerkeszthetők.

## Új termékek felvitele

* Új termékek hozzáadása egyszerű és gyors folyamat.
* A következő adatok rögzíthetők az új termékhez:
  + Termék neve
  + Ár
  + Lehetséges módosítási opciók (pl. extra feltétek, szószok)
  + Termék kategóriája (pl. Wrap, Kebab, Drink, SideDish)
  + Termék képe feltölthető PNG formátumban
* Az új termékek mentése az API-n keresztül történik.

## Már meglévő termékek teljes körű szerkesztése

* Az alkalmazás támogatja a termékek teljes körű szerkesztését.
* A meglévő termékek adatai bármikor módosíthatók, beleértve:
  + Nevet
  + Árat
  + Módosítási lehetőségeket
  + Kategóriát
  + Termékképet
* Lehetőség van a termékek archiválására vagy inaktiválására, ha már nem elérhetők.

## Felület minta

A képen szöveg, menü, Számítógép-monitor, fedett pályás látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

# Használt programok, fejlesztői környezetek és futtatási környezet

## Használt programok:

|  |  |
| --- | --- |
| Postman Makes Building Software Accessible to All, Announces Postman Flows | Az api teszteléséhez  A kérésekre adott válaszok formátumának ellenőrzéséhez  Telepítése:  https://www.postman.com/downloads/  A Windows 64-bit-es verziót kell kiválasztani és el is indul a letöltés |
| POSTMAN használata:  1. Megadjuk az API hívás paramétereit  2. Elküldjük az API hívást |  |
| What is XAMPP? | TechABU | Helyi szerver futtatása  MySQL adatbázis futtatása a helyi szerveren  Letöltése:  <https://www.apachefriends.org/hu/index.html>  A Windows 64-bit-es verziót kell kiválasztani és el is indul a letöltés |
| XAMPP konfigurációja:  1. Megnyitjuk a MySQL my.ini konfigurációs fájlt  2. Megkeressük a max\_allowed\_packet nevű változót majd át állítjuk 1024M-re vagy nagyobbra  3. Elmentjük a konfigurácíót |  |
|  | XAMPP indítása és használata:  1. Elindítjuk az Apache szervert  2. Elindítjuk a MySQL adatbázist  3. A MySQL admin gomb megnyomásával hozzá férünk az adatbázis admin felületéhez |

## Használt fejlesztői környezetek





A fejlesztéshez használt fejlesztői környezet

Letöltés:

## <https://visualstudio.microsoft.com/downloads/>



## Futtatási környezet és hardware

Windows 11-es PC 32GB memóriával

Fejlesztői környezet: Visual Studio 2022 Community Edition

Processzor: AMD Ryzen 5600X

Memória: 2X16GB 3200Mhz DDR4 ram

Tárhely: 1TB NVME SSD

GPU: Nvidia RTX 3060Ti

Megjelenítés: 1920\*1080

Egér: Szükséges

Billentyűzet: Szükséges

# A program használata

## A program indítása

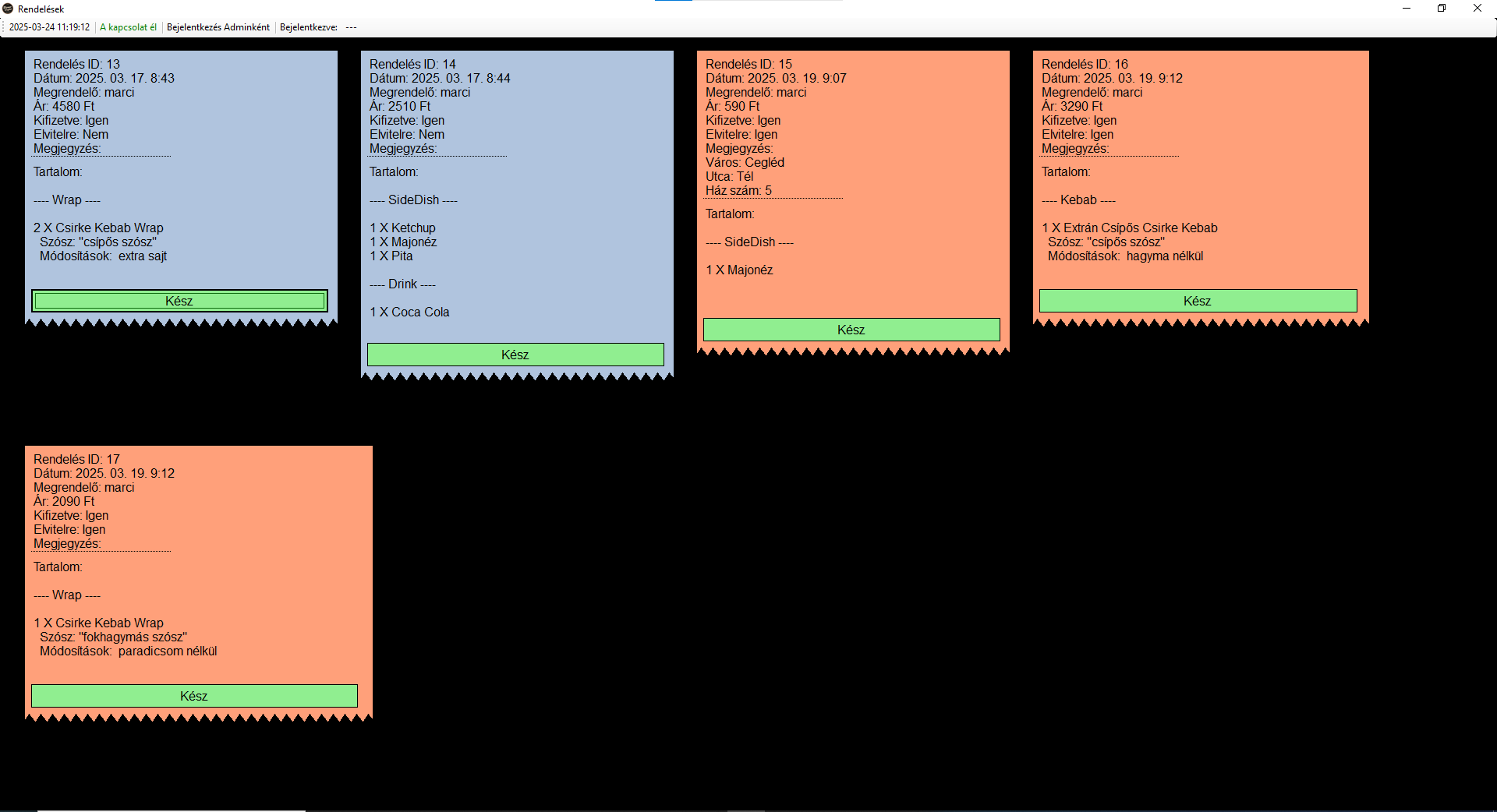
Az EtteremSideApp.sln megnyitása után à



És el is indult a program



## A rendelések megjelenítése és kiadása oldal



Az alkalmazás indítása után ez az oldal fogadja a felhasználót itt tudja megtekinteni a rendeléseket részletesen és kiadni azokat



### Sima rendelés

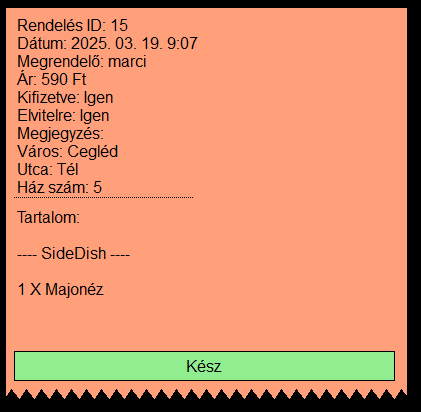
A sima rendelést jelzi a kék szín.

Tartalmazza a rendelés adatait és a megrendelt termékeket, azok esetleges módosításait.

Az alján a zöld kész gomb megnyomásával lehet kiadni a rendelést.



### Elvitelre szánt rendelés



Az elvitelre szánt rendelést jelzi a narancs szín.

Tartalmazza a rendelés adatait és a megrendelt termékeket, azok esetleges módosításait.

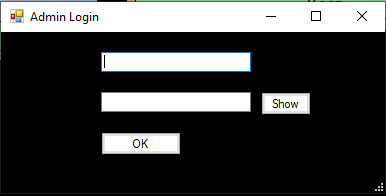
Ezen felül a kiszállítás adatait is

Az alján a zöld kész gomb megnyomásával lehet kiadni a rendelést.

### A képen szöveg, képernyőkép, Betűtípus, multimédia látható Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Rendelés kiadás

Ha a felhasználó kiadja a rendelés az eltűnik az aktív rendelések közül, inaktívvá válik

## Admin bejelentkezés



Az admin jogosultsággal rendelkező felhasználó be bír jelentkezni a fiókja emailcímével és jelszavával hogy hozzá férjen további funkciókhoz

Bejelentkezés előtt:



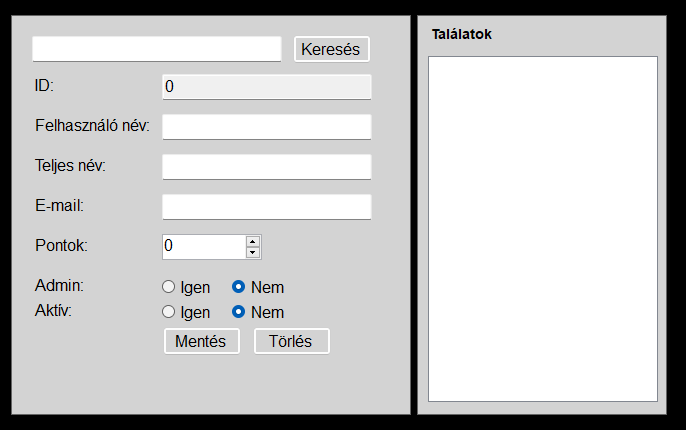


Bejelentkezés után:



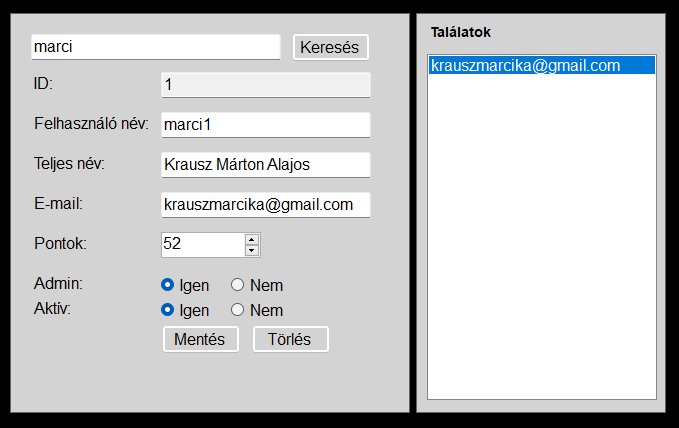
## Regisztrált felhasználók kezelése oldal

Üres felhasználó módosítás oldal:

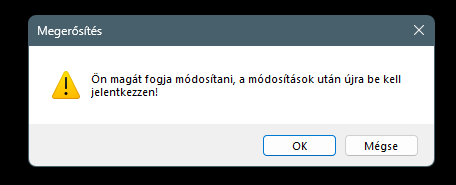




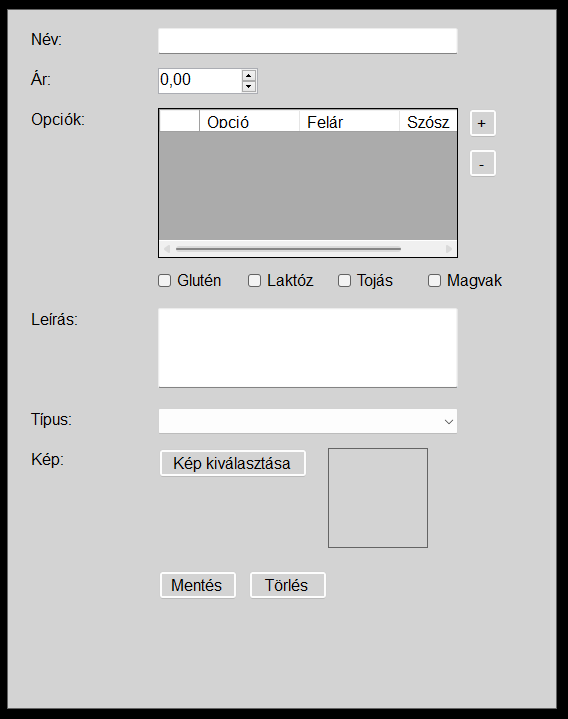
Keresés után kiválasztunk egy felhasználót és annak adatai megjelennek a mezőkben és tudjuk őket módosítani:



Ha az admin saját magát próbálja módosítani ezt az üzenetet fogja kapni:



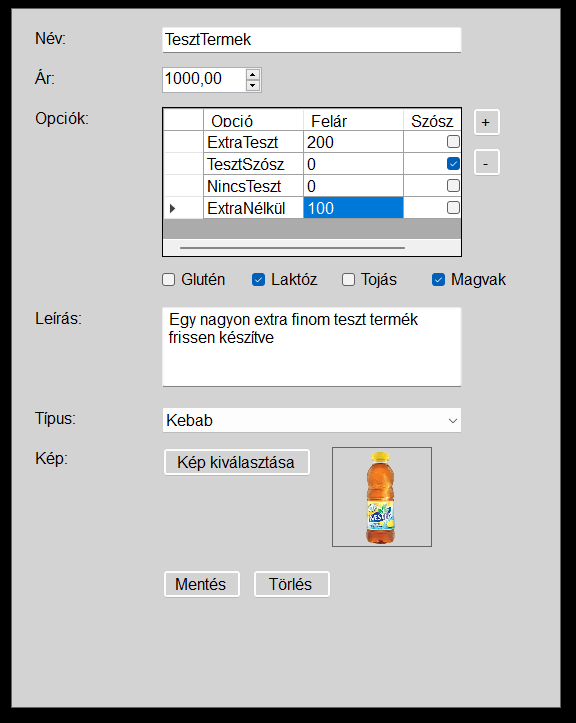
## Új termékek felvitele oldal



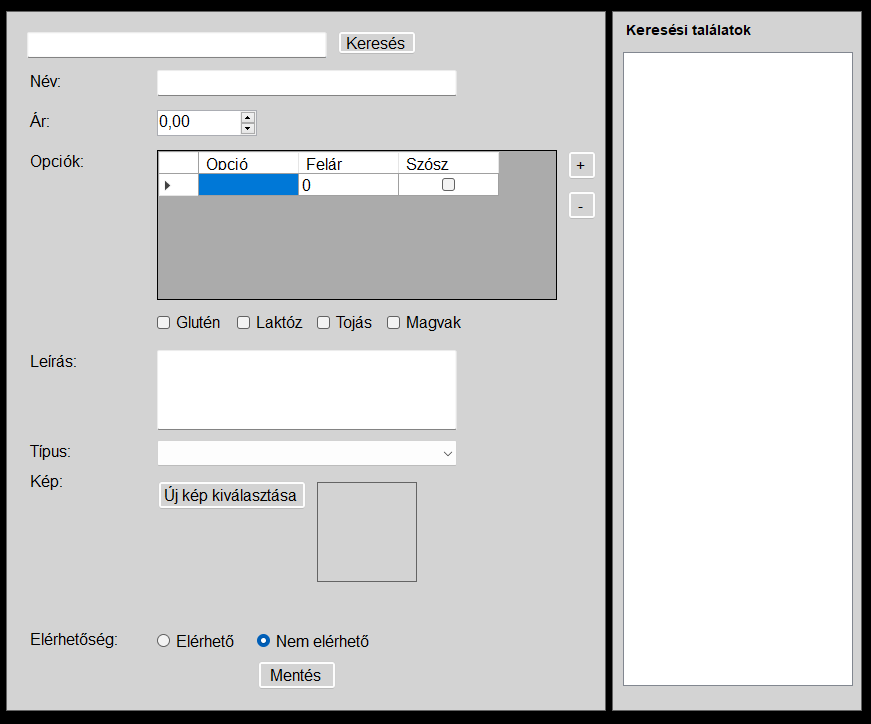
Az admin felhasználó ezen a felületen tud termékeket felvinni.

Meg kell adja a termék: nevét, árát, opcióit(név, ár, szósz e), leírását, típusát, majd ki kell választani egy PNG-t

Ugyan ez a felület adatokkal kitöltve:

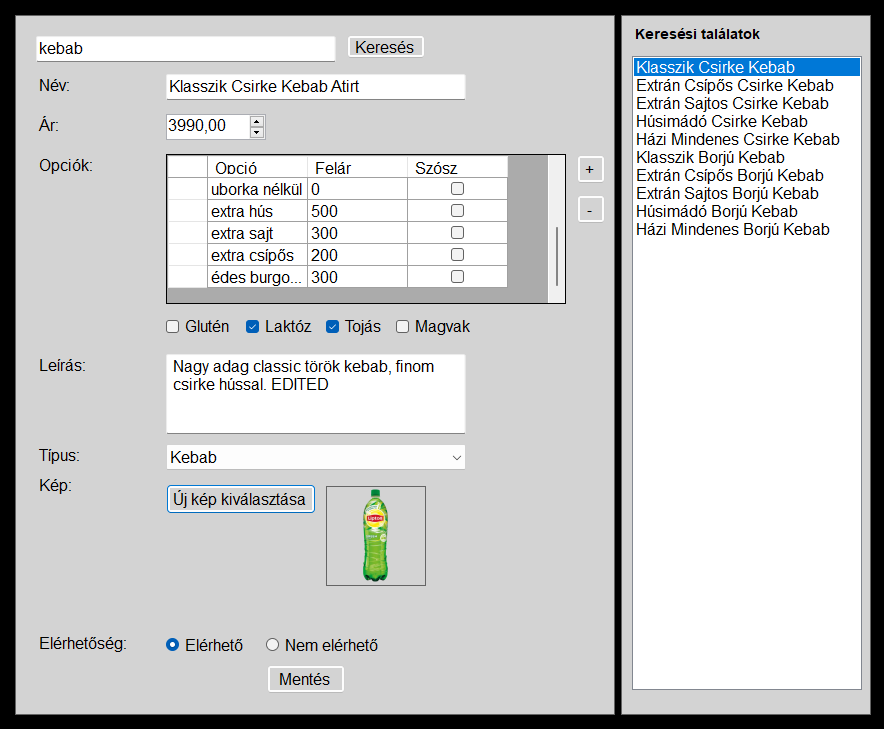


## A már meglévő termékek teljes körű szerkesztése oldal



Az admin felhasználó a keres gomb segítségével ki tudja keresni a termékeket majd a felhasználók szerkesztéséhez hasonlóan egy találat kiválasztása után a mezőkben lévő adatokat tudja módosítani.

Példa egy kitöltött felületre:



# Dokumentáció a backend tesztekről

Ez a dokumentáció részlet a main.test.js teszt fájlt mutatja be, amely a projekt backend működésének menetét teszteli, evvel biztosítva a funkciók hibamentes működését, és megfelelően kezelje a hibákat és hibaüzeneteket.

# Környezetleírás

A teszt futtatása közben használt környezet:

* Windows 10 Operációs Rendszer
* Visual Studio Code 1.98.2-as verzió
* Supertest a tesztelés lefuttatásához

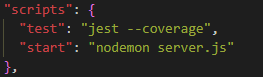
# A fájl felépítése

A main.test.js fájl struktúrája következő féleképpen néz ki:

* Backend tesztek:
  + Controller tesztek: (Például Dish Controller teszt)
  + Middleware tesztek: (Például errorHandler teszt)
  + Repository tesztek: (Például Dish Repository teszt)

# Használat

A tesztek lefuttatása elött szükségünk van néhány előkészületre:

* A helyes Node csomag letöltése, ebben az esetben supertest és jest, ezt megtehetjük a lefuttatni kívánt gyökér mappában, amely ebben a projektben “backend” néven szól.  
  A Terminált használva a megadott úton a “npm i supertest” kódot használva telepítjük a supertest-et:  
  
* A jest beállítása alapértelmezett tesztelő keretrendszernek, amelyet a “package.json” fájlban tehetünk meg a “scripts” mezőnél:  
    
  A ”--coverage” kóddal megadjuk, hogy minden teszt lefutása után készüljön egy kódlefedettségi jelentés.
* A teszt lefuttatása az említett lépések után egyszerűen megtehető a terminálban vagy konzolban a “npm test” kóddal:  
  

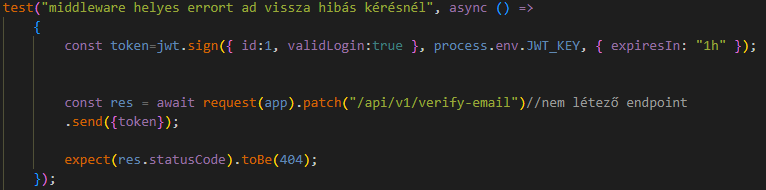
# Fontosabb tesztek bemutatása

## Middlewares tesztek

A Middlewares tesztek fő célja a hibák helyesen tovább küldése, hogy a felhasználó vagy fejlesztő felismerje a hiba eredetét.

### Error Handler tesztek

Az Error Handler, ahogy a neve is sugallja, a hibákat kezeli és küldi azokat tovább a kód biztonsága és működése érdekében, példa erre:

Ebben a példában megpróbálunk egy Controller tesztet lefuttatni, azonban egy nem létező végpontra, ami miatt hibába ütközik a folyamat és 404-es státuszkódot, hibakódot várunk vissza.

Ezt a tesztet azért kell külön “köztes réteggel” (middleware-el) letesztelni, mível a controllerben nem tudjuk azt az esetet letesztelni, ha rossz a végpont, mert nem éri el, ezért az errort tovább küldjük és visszaadja a middleware.

## Repository tesztek

A repository, “tároló” egységek tesztelésének a célja az adattárolás működésének és hibakezelésének biztosítása, emellett biztosítja, hogy az előforduló hibák megelőzhetőek legyenek és stabil maradjon a kód.

### User Repository tesztek

A User Repository tesztek főbb célja a felhasználóval kapcsolatos funkciók kezelését biztosítja, példa erre:

A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Ebben a példában leteszteljük a Felhasználó létrehozását és annak lekérését.  
Előre beégetett adatokat használunk, így könyebben ráláthatunk a lehetséges hibákra.

### Dish Repository tesztek

A Dish Repository tesztek főbb célja a fogásokkal, ételekkel kapcsolatos funkciók kezelését biztosítja, példa erre:

A képen szöveg, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Kettő példa látható, az első példában leteszteljük a fogás létrehozását és annak lekérését.  
Előre beégetett adatokat használunk, így könnyebben ráláthatunk a lehetséges hibákra.

A második példában az összes leteszteljük az összes fogás lekérését és elvárjuk, hogy a visszakapott hossz 1 legyen, mivel az előző tesztben készítettünk egyet.

## Controller tesztek

A vezérlőrétegen végzett tesztek biztosítják a kód működését és a kérések, válaszok helyességét, emellett a bemeneti adatokat is vizsgáljuk és több esetre figyelünk tesztelés során.

### User Controller tesztek

A User Controller tesztekkel biztosítjuk, hogy a felhasználóhoz kapcsolódó kérések és válaszok helyes értékkel térnek vissza és hiba nélkül mennek végbe, példa erre:

Ebben a példában kettő tesztet láthatunk, amelyek a felhasználó létrehozását kezelik, az egyiknél helyes választ várunk, míg a másiknál helytelent. Az első tesztben az ”/api/v1/register” végponton a kérés testében beküldjük a létrehozáshoz, regisztráláshoz elvárt szükséges adatokat, ami után elvárjuk, hogy a válasz státuszkódja a helyzetnek megfelelően helyesen térjen vissza.  
A második tesztben ugyanazt a kérést teszteljük, azonban ebben az esetben egy szándékosan hibás kérést küldünk be (hiányzó adat) és várjuk az adott helyzethez megfelelő választ.

### Dish Controller

A Dish Controller tesztekkel biztosítjuk, hogy a fogásokhoz kapcsolódó kérések és válaszok helyes értékkel térnek vissza és hiba nélkül mennek végbe, példa erre:

A képen szöveg, képernyőkép, Betűtípus, képernyő látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Ebben a példában egy fogás, étel létrehozását láthatjuk, ahol helyes értéket várunk vissza.  
Az ”/api/v1/dish” végponton a kérés testében beküldjük a létrehozáshoz elvárt szükséges adatokat.

# Források

McDonalds rendelés felület (inspiráció)

Postman

Xampp

Visual studio code

Microsoft Visual studio

CocaCola hu (termék fotók)