VÁCI SZAKKÉPZÉSI CENTRUM   
BORONKAY GYÖRGY   
MŰSZAKI TECHNIKUM ÉS GIMNÁZIUM

**VIZSGAREMEK**

**Fedor Benjámin**

**Kertész Kornél**

**Turza Norbert**

**2024.**

**VÁCI SZAKKÉPZÉSI CENTRUM   
BORONKAY GYÖRGY   
MŰSZAKI TECHNIKUM ÉS GIMNÁZIUM**



**VIZSGAREMEK**

**Frissfutár**

|  |  |
| --- | --- |
| Konzulens: | Készítette: Fedor Benjámin  Kertész Kornél  Turza Norbert |

# **Hallgatói nyilatkozat**

Alulírottak, ezúton kijelentjük, hogy a vizsgaremek saját, önálló munkánk, és korábban még sehol nem került publikálásra.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Fedor Benjámin |  | Turza Norbert |  | Kertész Kornél |

# **Konzultációs lap**

Vizsgázók neve: Fedor Benjámin - Turza Norbert - Kertész Kornél

Vizsgaremek címe: Frissfutár

Program nyújtotta szolgáltatások:

* regisztráció
* bejelentkezés
* kijelentkezés
* rendelés
* szűrés ételfajták alapján
* adminisztrátori funkciók

|  |  |  |
| --- | --- | --- |
| Sorszám | A konzultáció időpontja | A konzulens aláírása |
| 1. | 2023.10.17. |  |
| 2. | 2023.11.21. |  |
| 3. | 2023.12.13. |  |
| 4. | 2024.01.16 |  |
| 5. | 2024.02.20. |  |
| 6. | 2024.03.12 |  |

|  |  |  |
| --- | --- | --- |
| A vizsgaremek beadható: |  | A vizsgaremek átvettem: |
| Vác, 2024. |  | Vác, 2024. |
| Konzulens |  | A szakképzést folytató intézmény felelőse |

**Tartalomjegyzék**

[**Hallgatói nyilatkozat** 3](#_Toc164941797)

[**Konzultációs lap** 4](#_Toc164941798)

[**Tartalomjegyzék** 5](#_Toc164941799)

[**Témaválasztás** 7](#_Toc164941800)

[**1.** **Fejlesztői dokumentáció** 8](#_Toc164941801)

[1.1 Ismertetés 8](#_Toc164941802)

[1.2 Fejlesztői környezet, Nyelvek és Technológiák 8](#_Toc164941803)

[1.2.1 Visual Studio Code 8](#_Toc164941804)

[1.2.2 Visual Studio 2022 8](#_Toc164941805)

[1.2.3 PHP 8](#_Toc164941806)

[1.2.4 CSS 8](#_Toc164941807)

[1.2.5 C# 9](#_Toc164941808)

[1.2.6 Javascript 9](#_Toc164941809)

[1.2.7 HTML 9](#_Toc164941810)

[1.3 Adatszerkezet 9](#_Toc164941811)

[1.3.1 Adattáblák 9](#_Toc164941812)

[1.3.2 Táblák közötti kapcsolatok 10](#_Toc164941813)

[1.3.3 A „telepulesek” tábla 11](#_Toc164941814)

[1.3.4 A „varmegyek” tábla 12](#_Toc164941815)

[1.3.5 A „szallitasi\_cim” tábla 13](#_Toc164941816)

[1.3.6 A „felhasznalo\_adatok” tábla 14](#_Toc164941817)

[1.3.7 A „rendeles” tábla 15](#_Toc164941818)

[1.3.8 A „fizetesi\_modok” tábla 17](#_Toc164941819)

[1.3.9 A „rendeles\_statusza” tábla 18](#_Toc164941820)

[1.3.10 A „rendeles\_tetelei” tábla 18](#_Toc164941821)

[1.3.11 A „termek” tábla 20](#_Toc164941822)

[1.3.12 Az „afa\_kulcs” tábla 21](#_Toc164941823)

[1.3.13 A „mennyisegi\_egysegek” tábla 22](#_Toc164941824)

[1.3.14 A „termek\_csoport” tábla 23](#_Toc164941825)

[1.3.15 A „termek\_kategoria” tábla 24](#_Toc164941826)

[1.3.16 A „termek\_csomag\_reszletei” tábla 25](#_Toc164941827)

[1.3.17 A „termek\_csomag” tábla 26](#_Toc164941828)

[1.3.18 Az „admin” tábla 27](#_Toc164941829)

[1.4 Használati esetmodell, szerepkörök 28](#_Toc164941830)

[1.5 Modulok ismertetése 29](#_Toc164941831)

[1.5.1 Webes alkalmazás 29](#_Toc164941832)

[1.5.2 Asztali alkalmazás 34](#_Toc164941833)

[1.6 Tesztelések 38](#_Toc164941834)

[1.6.1 Reszponzivitás 38](#_Toc164941835)

[1.6.2 Téma váltás 39](#_Toc164941836)

[1.6.3 Adatok kezelése 40](#_Toc164941837)

[1.6.4 Adatbázis 41](#_Toc164941838)

[1.6.5 Admin applikáció tesztelése 44](#_Toc164941839)

[1.7 Továbbfejlesztési lehetőségek 48](#_Toc164941840)

[1.7.1 Keresés funkció megvalósítása 48](#_Toc164941841)

[1.7.2 Fizetési módok kezelése 48](#_Toc164941842)

[1.7.3 Képek nem kizárólagosan az „img” mappából történő megjelenítése 48](#_Toc164941843)

[1.7.4 Rendelés visszaigazolás, email küldése 48](#_Toc164941844)

[1.7.5 Rendelések megtekintése külön webes felületen (admin) 48](#_Toc164941845)

[1.7.6 Adatbázis szerveren történő működtetése 48](#_Toc164941846)

[1.7.7 Bug-ok javítása 48](#_Toc164941847)

[1.7.8 Program feltelepítése, adatbázis mentése, frissítése 48](#_Toc164941848)

[1.7.9 Kiemelt ajánlatok kezelése 48](#_Toc164941849)

[**2** **Felhasználói dokumentáció** 49](#_Toc164941850)

[2.1 Ismertetés 49](#_Toc164941851)

[2.2 Rendszerkövetelmények 49](#_Toc164941852)

[2.3 Alkalmazás telepítése, beállítása 49](#_Toc164941853)

[2.4 Alkalmazás használata 50](#_Toc164941854)

[2.4.1 Webes alkalmazás használata 50](#_Toc164941855)

[2.4.2 Asztali alkalmazás használata 54](#_Toc164941856)

[**3** **Irodalomjegyzék** 55](#_Toc164941857)

[1.5 Internetes források 55](#_Toc164941858)

[**2.** **Mellékletek** 56](#_Toc164941859)

# **Témaválasztás**

A mesterremekünk témájául egy rendkívül egyszerű, de sokrétű és felhasználó barát online étel rendelési oldalt készítettünk.

A platformunk célja, hogy megkönnyítse az emberek számára az ételrendelést és a házhozszállítást, minimalizálva ezzel az adminisztratív terheket és a várakozási időket.

Az oldalunk intuitív felülete és felhasználóbarát tervezése lehetővé teszi, hogy a felhasználók könnyen böngészhessék az éttermek és ételek széles választékát, és egyszerűen kiválaszthassák a számukra legmegfelelőbb lehetőségeket.

A rendszerünk rugalmas és testre szabható, így mindenki megtalálhatja az igényeinek megfelelő lehetőséget, legyen szó diétás étkezésről, allergiák figyelembevételéről vagy akár különleges ételkérésekről.

A célunk, hogy a felhasználók számára kényelmes és problémamentes élményt nyújtsunk az ételrendelésben, és hozzájáruljunk a mindennapi életük egyszerűsítéséhez és élvezetesebbé tételéhez.

# **Fejlesztői dokumentáció**

## Ismertetés

A Frissfutár egy étel - és italrendelésre használatos reszponzív weboldal (PC-n tableten, telefonon egyaránt kényelmesen használható), asztali admin alkalmazással, ami teljeskörű karbantartást biztosít.

## Fejlesztői környezet, Nyelvek és Technológiák

### Visual Studio Code

A Visual Studio Code egy ingyenes, nyílt forráskódú fejlesztői környezet, amelyet a Microsoft fejlesztett ki. Számos programozási nyelvet támogat, és rendelkezik számos funkcióval, amelyek segítenek a fejlesztőknek hatékonyabban dolgozni. Ezek közé tartozik a kódszerkesztő, a beépített Git integráció, kiterjesztési lehetőségek, hibakeresési eszközök, és sok más. Az intuitív felület és a testreszabhatóság lehetővé teszi a fejlesztők számára, hogy az igényeiknek megfelelően alakítsák ki a fejlesztői környezetet.

### Visual Studio 2022

A Visual Studio 2022 egy integrált fejlesztői környezet (IDE) a Microsofttól, amely kifejezetten a fejlesztőknek készült. Ez a szoftvercsomag számos programozási nyelvet és platformot támogat, például C#, C++, JavaScript, Python és még sok más. A Visual Studio 2022 számos fejlett funkcióval rendelkezik, például kódszerkesztő, hibakeresés, verziókezelés (pl. Git), valamint kiterjesztési lehetőségek, amelyek segítségével testreszabhatod a fejlesztői környezetet az igényeidnek megfelelően. Az újabb verziók általában új funkciókat, javításokat és optimalizációkat is hoznak, hogy még hatékonyabbá tegyék a fejlesztést és a csapatmunkát.

### PHP

A PHP egy szkriptnyelv, amely kifejezetten webfejlesztésre lett tervezve. A rövidítés "PHP: Hypertext Preprocessor"-ra utal, ami egy visszafelé rekuzív akronym. Dinamikus, szerveroldali nyelvként működik, ami azt jelenti, hogy a PHP kód a webszerveren fut, és generálja a dinamikus weboldalak tartalmát.

### CSS

A CSS (Cascading Style Sheets) egy nyelv, amelyet a weboldalak stílusának meghatározására használnak. A CSS segítségével formázhatod a weboldalak HTML elemeit, például megadhatod a színeket, a betűtípusokat, a méreteket, a térközöket és az elrendezést. A CSS stílusok elválasztják a tartalmat (HTML) a megjelenéstől, ami lehetővé teszi a jobb karbantarthatóságot és a rugalmasabb tervezést. A CSS-t a böngészők értelmezik és alkalmazzák, így a weboldalakat könnyen testreszabhatod, hogy igényeidnek és tervezési elképzeléseidnek megfeleljenek.

### C#

A C# (C-sharp) egy erőteljes, modern programozási nyelv, amelyet a Microsoft fejlesztett ki. A C# a .NET platform részeként szolgál, és széles körű alkalmazások fejlesztésére alkalmas, beleértve a Windows asztali alkalmazásokat, webalkalmazásokat, mobilalkalmazásokat és játékokat is.

### Javascript

A JavaScript egy dinamikus szkriptnyelv, amelyet elsősorban webfejlesztésre használnak. Gyakran használják interaktív webes alkalmazások és weboldalak fejlesztésére, amelyeknek célja a felhasználói élmény javítása és a weboldalak dinamizálása.

Lehetővé teszi a weboldalak elemzését, módosítását és azok viselkedésének kezelését a böngészőben futtatva. Ezáltal interaktív funkciókat, animációkat, űrlapok ellenőrzését, adatkezelést és még sok másat valósíthatunk meg.

Széles körben támogatott a legtöbb böngészőben, és rendelkezik egy nagy és aktív fejlesztői közösséggel, amely folyamatosan fejleszti és bővíti a nyelvet és annak környezetét.

### HTML

A HTML (HyperText Markup Language) az internetes tartalom alapvető építőeleme, amely lehetővé teszi a strukturált szöveg megjelenítését böngészők segítségével. A HTML elemek, attribútumok és tartalom széles választékát kínálja, amelyek lehetővé teszik a tartalom testreszabását és a weboldalak funkcióinak bővítését. A HTML folyamatosan fejlődik, és újabb és újabb lehetőségeket kínál a webfejlesztőknek a weboldalak interaktívabbá és vonzóbbá tételéhez.

## Adatszerkezet

Adatbázis neve: frissfutar

Tárolómotor: InnoDB

Alapértelmezett illesztés: utf8\_hungarian\_ci

### Adattáblák

* telepulesek
* varmegyek
* szallitasi\_cim
* felhasznalo\_adatok
* rendeles
* fizetesi\_modok
* rendeles\_statusza
* rendeles\_tetelei
* termek
* afa\_kulcs
* mennyisegi\_egysegek
* termek\_csoport
* termek\_kategoria
* termek\_csomag\_reszletei
* termek\_csomag
* admin

### A képen szöveg, képernyőkép, diagram, Betűtípus látható Automatikusan generált leírásTáblák közötti kapcsolatok

Az adatbázis megtervezése: Turza Norbert, Kertész Kornél és Fedor Benjámin

Az adatbázist, a táblákat és a kapcsolatot létrehozta: Kertész Kornél

A táblák feltöltése tesztadatokkal: Turza Norbert és Kertész Kornél

A forrásgyűjtést, valamint az adatok feltöltését végrehajtotta: Turza Norbert

### A „telepulesek” tábla

Ez a tábla tartalmazza településeket.



* id: a települések azonosítója (elsődleges kulcs)
* varmegyek\_id: a vármegyék azonosítója, amelyikhez tartozik az adott település
* i\_szam: a települések irányítószáma
* telepules: a települések neve

CREATE TABLE `telepulesek` (

`id` int(11) NOT NULL,

`varmegyek\_id` int(11) NOT NULL,

`i\_szam` varchar(4) NOT NULL,

`telepules` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

ALTER TABLE `telepulesek`

ADD PRIMARY KEY (`id`),

ADD KEY `telepulesek\_ibfk\_1` (`varmegyek\_id`);

ALTER TABLE `telepulesek`

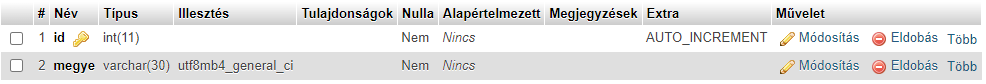
MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3579;

ALTER TABLE `telepulesek`

ADD CONSTRAINT `telepulesek\_ibfk\_1` FOREIGN KEY (`varmegyek\_id`) REFERENCES `varmegyek` (`id`);

### A „varmegyek” tábla

Ez a tábla tartalmazza vármegyéket.



* id: a vármegyék azonosítója (elsődleges kulcs)
* megye: a vármegye megnevezése

CREATE TABLE `varmegyek` (

`id` int(11) NOT NULL,

`megye` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

ALTER TABLE `varmegyek`

ADD PRIMARY KEY (`id`);

ALTER TABLE `varmegyek`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=20;

### A „szallitasi\_cim” tábla

Ez a tábla tartalmazza a szállítási címek adatait



* id: szállítási cím azonosítója (elsődleges kulcs)
* felhasznalo\_adatok\_id: a felhasználói adatok azonosítója, ez az idegen kulcs a felhasznalo\_adatok táblához
* telepules\_id: a települések id-je, ez az idegen kulcs a telepulesek táblához

CREATE TABLE `szallitasi\_cim` (

`id` int(11) NOT NULL,

`felhasznalo\_adatok\_id` int(11) NOT NULL,

`telepules\_id` int(11) NOT NULL,

`utca\_hazszam` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

ALTER TABLE `szallitasi\_cim`

ADD PRIMARY KEY (`id`),

ADD KEY `felhasznalo\_adatok\_id` (`felhasznalo\_adatok\_id`),

ADD KEY `telepules\_id` (`telepules\_id`);

ALTER TABLE `szallitasi\_cim`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

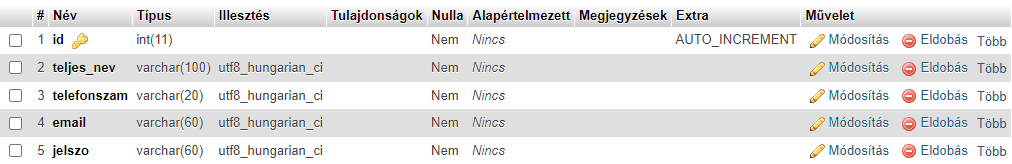
ALTER TABLE `szallitasi\_cim`

ADD CONSTRAINT `szallitasi\_cim\_ibfk\_1` FOREIGN KEY (`felhasznalo\_adatok\_id`) REFERENCES `felhasznalo\_adatok` (`id`),

ADD CONSTRAINT `szallitasi\_cim\_ibfk\_2` FOREIGN KEY (`telepules\_id`) REFERENCES `telepulesek` (`id`);

### A „felhasznalo\_adatok” tábla

Ez a tábla a tartalmazza a felhasználók adatait



* id: a felhasználó adatok azonosítója (elsődleges kulcs)
* teljes\_nev: a felhasználó teljes neve (vezetéknév, keresztnév), (regisztrációhoz)
* telefonszam: a felhasználó telefonszáma (regisztrációhoz)
* email: a felhasználó email címe (bejelentkezéshez, regisztrációhoz)
* jelszo: a felhasználó jelszava (bejelentkezéshez, regisztrációhoz)

CREATE TABLE `felhasznalo\_adatok` (

`id` int(11) NOT NULL,

`teljes\_nev` varchar(100) NOT NULL,

`telefonszam` varchar(20) NOT NULL,

`email` varchar(60) NOT NULL,

`jelszo` varchar(60) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

ALTER TABLE `felhasznalo\_adatok`

ADD PRIMARY KEY (`id`);

ALTER TABLE `felhasznalo\_adatok`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

### A „rendeles” tábla

Ez a tábla tartalmazza a rendeléshez szükséges adatokat



* id: a rendelések azonosítója (elsődleges kulcs)
* felhasznalo\_adatok\_id: a felhasználói adatok azonosítója, a regisztrált felhasználók adatainak elérésére szolgál
* teljes\_nev: a nem regisztrált felhasználó teljes neve (vezetéknév, keresztnév), bejelentkezett felhasználóknál nem kell megadni
* cim: a nem regisztrált felhasználó szállítási címe, bejelentkezett felhasználónál nem kell megadni, az eddig elmentett címekből tud választani a felhasználó(lenyíló menü, alaphelyzetben a legutóbbi címmel tölti ki, plusz gombbal lehet megadni másik címet, melyet rendelés után szintén elment)
* fizetesi\_modok\_id: a fizetési módok azonosítója
* rendeles\_statusza\_id: a rendelési státusz azonosítója, ez az idegenkulcs a rendeles\_statusza táblához
* rendeles\_datumido: egy adott felhasználó rendelés feldolgozásának időpontja
* szallitas\_datum: akkor kell megadni, ha a felhasználó máskorra kéri a szállítást
* szallitas\_megj: a felhasználó megjegyzése a rendeléshez
* brutto\_vegosszeg: a fizetendő végösszeg (mennyiseg \* netto\_egyseg\_ar \* afakulcs)
* nettó\_vegosszeg: az áfa nélküli végösszeg (mennyiseg \* netto\_egyseg\_ar)

CREATE TABLE `rendeles` (

`id` int(11) NOT NULL,

`felhasznalo\_adatok\_id` int(11) NOT NULL,

`teljes\_nev` varchar(100) NOT NULL,

`cim` varchar(150) NOT NULL,

`fizetesi\_modok\_id` int(11) NOT NULL,

`rendeles\_statusza\_id` int(11) NOT NULL,

`rendeles\_datumido` datetime NOT NULL,

`szallitas\_datum` date NOT NULL,

`szallitas\_megj` varchar(50) NOT NULL,

`brutto\_vegosszeg` int(8) NOT NULL,

`netto\_vegosszeg` int(8) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

ALTER TABLE `rendeles`

ADD PRIMARY KEY (`id`),

ADD KEY `felhasznalo\_adatok\_id` (`felhasznalo\_adatok\_id`),

ADD KEY `fizetesi\_modok\_id` (`fizetesi\_modok\_id`),

ADD KEY `rendeles\_statusza\_id` (`rendeles\_statusza\_id`);

ALTER TABLE `rendeles`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `rendeles`

ADD CONSTRAINT `rendeles\_ibfk\_1` FOREIGN KEY (`felhasznalo\_adatok\_id`) REFERENCES `felhasznalo\_adatok` (`id`),

ADD CONSTRAINT `rendeles\_ibfk\_2` FOREIGN KEY (`fizetesi\_modok\_id`) REFERENCES `fizetesi\_modok` (`id`),

ADD CONSTRAINT `rendeles\_ibfk\_3` FOREIGN KEY (`rendeles\_statusza\_id`) REFERENCES `rendeles\_statusza` (`id`);

### A „fizetesi\_modok” tábla

Ez a tábla tartalmazza a rendelésnél kiválasztható fizetési módokat



* id: a fizetési módok azonosítója (elsődleges kulcs)
* megnevezes: fizetési módok megnevezése(bankkártya, SZÉP-kártya, készpénz)

CREATE TABLE `fizetesi\_modok` (

`id` int(11) NOT NULL,

`megnevezes` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

ALTER TABLE `fizetesi\_modok`

ADD PRIMARY KEY (`id`);

ALTER TABLE `fizetesi\_modok`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;

### A „rendeles\_statusza” tábla

Ez a tábla tartalmazza a rendelés állapotát



* id: a rendelés státuszának az azonosítója (elsődleges kulcs)
* rendeles\_allapot: a rendelés állapota (megkaptuk-készítjük, elkészült-szállítjuk, kiszállítva)

CREATE TABLE `rendeles\_statusza` (

`id` int(11) NOT NULL,

`rendeles\_allapot` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

ALTER TABLE `rendeles\_statusza`

ADD PRIMARY KEY (`id`);

ALTER TABLE `rendeles\_statusza`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;

### A „rendeles\_tetelei” tábla

Ez a tábla tartalmazza a felhasználó rendelésének tételeit és azoknak a paramétereit (árát, mennyiségét, nevét) ebből lesz a blokk

A képen szöveg, Betűtípus, szám, képernyőkép látható

Automatikusan generált leírás

* id: a rendelés tételeinek az azonosítója (elsődleges kulcs)
* rendeles\_id: a rendelés azonosítója, ez az idegenkulcs a rendeles táblához
* termek\_id: a termék azonosítója, ez az idegenkulcs a termek táblához
* megnevezes: a rendelésben szereplő termékek tételenkénti megnevezése
* mennyiseg: a rendelt termékek tételenkénti mennyisége
* netto\_egyseg\_ar: a termékek tételenkénti nettó ára
* afakulcs: az adott termékekhez tartozó áfakulcs

CREATE TABLE `rendeles\_tetelei` (

`id` int(11) NOT NULL,

`rendeles\_id` int(11) NOT NULL,

`termek\_id` int(11) NOT NULL,

`megnevezes` varchar(100) NOT NULL,

`mennyiseg` int(11) NOT NULL,

`netto\_egyseg\_ar` int(11) NOT NULL,

`afakulcs` double NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

ALTER TABLE `rendeles\_tetelei`

ADD PRIMARY KEY (`id`),

ADD KEY `rendeles\_id` (`rendeles\_id`),

ADD KEY `termek\_id` (`termek\_id`);

ALTER TABLE `rendeles\_tetelei`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `rendeles\_tetelei`

ADD CONSTRAINT `rendeles\_tetelei\_ibfk\_1` FOREIGN KEY (`rendeles\_id`) REFERENCES `rendeles` (`id`),

ADD CONSTRAINT `rendeles\_tetelei\_ibfk\_2` FOREIGN KEY (`termek\_id`) REFERENCES `termek` (`id`);

### A „termek” tábla

Ez a tábla tartalmazza az oldalon lévő termékek adatait



* id: a termek azonosítója (elsődleges kulcs)
* termek\_csoport\_id: : a termék csoportok azonosítója, ez az idegenkulcs a termek\_csoport táblához
* megnevezes: az oldalon szereplő termékek teljes neve
* leiras: az oldalon szereplő termékek részletes leírása
* kep: az oldalon szereplő termékekről készült illusztráció
* netto\_egyseg\_ar: az oldalon szereplő termékek nettó ára
* mennyisegi\_egysegek\_id: a mennyiségi egységek azonosítója, ez az idegenkulcs a mennyisegi\_egysegek táblához
* afa\_kulcs\_id: az áfakulcsok azonosítója, ez az idegenkulcs a afa\_kulcs táblához

CREATE TABLE `termek` (

`id` int(11) NOT NULL,

`termek\_csoport\_id` int(11) NOT NULL,

`megnevezes` varchar(100) NOT NULL,

`leiras` varchar(100) NOT NULL,

`kep` varchar(80) NOT NULL,

`netto\_egyseg\_ar` int(11) NOT NULL,

`mennyisegi\_egysegek\_id` int(11) NOT NULL,

`afa\_kulcs\_id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

ALTER TABLE `termek`

ADD PRIMARY KEY (`id`),

ADD KEY `mennyisegi\_egysegek\_id` (`mennyisegi\_egysegek\_id`),

ADD KEY `termek\_csoport\_id` (`termek\_csoport\_id`),

ADD KEY `afa\_kulcs\_id` (`afa\_kulcs\_id`);

ALTER TABLE `termek`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=82;ALTER TABLE `termek`

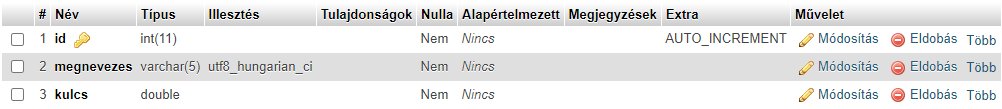
ADD CONSTRAINT `termek\_ibfk\_1` FOREIGN KEY (`mennyisegi\_egysegek\_id`) REFERENCES `mennyisegi\_egysegek` (`id`),

ADD CONSTRAINT `termek\_ibfk\_2` FOREIGN KEY (`termek\_csoport\_id`) REFERENCES `termek\_csoport` (`id`),

ADD CONSTRAINT `termek\_ibfk\_3` FOREIGN KEY (`afa\_kulcs\_id`) REFERENCES `afa\_kulcsok` (`id`);

### Az „afa\_kulcs” tábla

Ez a tábla tartalmazza a termékekhez kapcsolt áfakulcsokat



* id: az áfa kulcsok azonosítója (elsődleges kulcs)
* megnevezes: az áfakulcsok megnevezése
* kulcs: az áfakulcsok értéke (pl.: 0.27)

CREATE TABLE `afa\_kulcsok` (

`id` int(11) NOT NULL,

`megnevezes` varchar(5) NOT NULL,

`kulcs` double NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

ALTER TABLE `afa\_kulcsok`

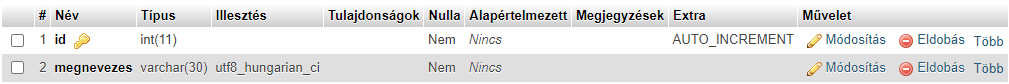
ADD PRIMARY KEY (`id`);

ALTER TABLE `afa\_kulcsok`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=6;

### A „mennyisegi\_egysegek” tábla

Ez a tábla tartalmazza a termékekhez használt mennyiségi egységeket (db, gramm, stb.)



* id: a mennyiségi egységek azonosítója (elsődleges kulcs)
* megnevezes: a mennyiségi egységek megnevezése

CREATE TABLE `mennyisegi\_egysegek` (

`id` int(11) NOT NULL,

`megnevezes` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

ALTER TABLE `mennyisegi\_egysegek`

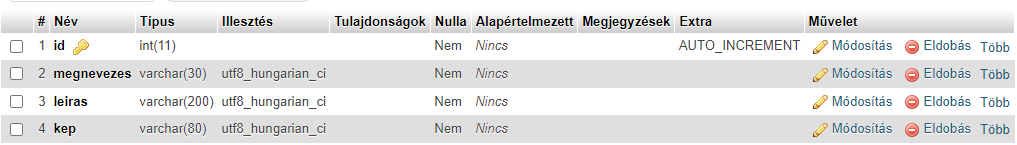
ADD PRIMARY KEY (`id`);

ALTER TABLE `mennyisegi\_egysegek`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=11;

### A „termek\_csoport” tábla

Ez a tábla tartalmazza a termékek csoportjait (étel, ital)



* id: a termék csoportok azonosítója (elsődleges kulcs)
* termek\_kategoria \_id: : a termék kategóriák azonosítója
* megnevezes: a termék csoportok megnevezése

CREATE TABLE `termek\_csoport` (

`id` int(11) NOT NULL,

`termek\_kategoria\_id` int(11) NOT NULL,

`megnevezes` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

ALTER TABLE `termek\_csoport`

ADD PRIMARY KEY (`id`),

ADD KEY `termek\_kategoria\_id` (`termek\_kategoria\_id`);

ALTER TABLE `termek\_csoport`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=17;

ALTER TABLE `termek\_csoport`

ADD CONSTRAINT `termek\_csoport\_ibfk\_1` FOREIGN KEY (`termek\_kategoria\_id`) REFERENCES `termek\_kategoria` (`id`);

COMMIT;

### A „termek\_kategoria” tábla

Ez a tábla tartalmazza a termékek kategóriáit (üdítők, szeszes italok, pizzák stb.)



* id: a termék kategóriák azonosítója (elsődleges kulcs)
* megnevezes: a termék kategóriáknak a megnevezése

CREATE TABLE `termek\_kategoria` (

`id` int(11) NOT NULL,

`megnevezes` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

ALTER TABLE `termek\_kategoria`

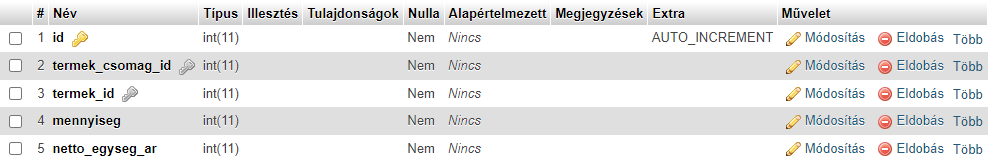
ADD PRIMARY KEY (`id`);

ALTER TABLE `termek\_kategoria`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

### A „termek\_csomag\_reszletei” tábla

Ez a tábla tartalmazza termék csomagoknak a tulajdonságait (árát, mennyiségét)



* id: a termék csomagok részleteinek az azonosítója (elsődleges kulcs)
* termek\_csomag \_id: : a termék csomagok azonosítója
* termek\_id: : a termékek azonosítója
* mennyiseg: a termék csomagban lévő termékek mennyisége
* \*\*netto\_egyseg\_ar: most ez nemtom minek a nettó ára

CREATE TABLE `termek\_csomag\_reszletei` (

`id` int(11) NOT NULL,

`termek\_csomag\_id` int(11) NOT NULL,

`termek\_id` int(11) NOT NULL,

`mennyiseg` int(11) NOT NULL,

`netto\_egyseg\_ar` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

ALTER TABLE `termek\_csomag\_reszletei`

ADD PRIMARY KEY (`id`),

ADD KEY `termek\_csomag\_id` (`termek\_csomag\_id`),

ADD KEY `termek\_id` (`termek\_id`);

ALTER TABLE `termek\_csomag\_reszletei`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

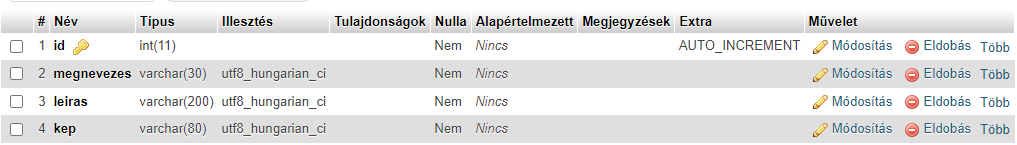
ALTER TABLE `termek\_csomag\_reszletei`

ADD CONSTRAINT `termek\_csomag\_reszletei\_ibfk\_1` FOREIGN KEY (`termek\_csomag\_id`) REFERENCES `termek\_csomag` (`id`),

ADD CONSTRAINT `termek\_csomag\_reszletei\_ibfk\_2` FOREIGN KEY (`termek\_id`) REFERENCES `termek` (`id`);

### A „termek\_csomag” tábla

Ez a tábla tartalmazza a termék csomagoknak a tulajdonságait (leírás, kép)



* id: a termék csomagok azonosítója (elsődleges kulcs)
* megnevezes: a termék csomagok megnevezése
* leiras: a termék csomagok leírása
* kep: a termék csomagról készült illusztráció

CREATE TABLE `termek\_csomag` (

`id` int(11) NOT NULL,

`megnevezes` varchar(30) NOT NULL,

`leiras` varchar(200) NOT NULL,

`kep` varchar(80) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

ALTER TABLE `termek\_csomag`

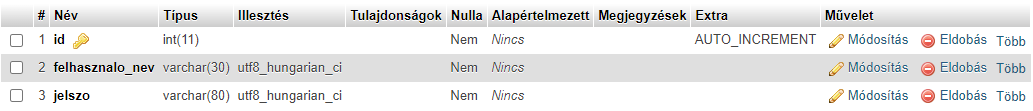
ADD PRIMARY KEY (`id`);

ALTER TABLE `termek\_csomag`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

### Az „admin” tábla

Ez a tábla tartalmazza az admin alkalmazáshoz tartozó adatokat



* id: az admin azonosítója (elsődleges kulcs)
* felhasznalo\_nev: a bejelentkezéshez szükséges felhasználónév
* jelszo: a bejelentkezéshez szükséges jelszó

CREATE TABLE `admin` (

`id` int(11) NOT NULL,

`felhasznalo\_nev` varchar(30) NOT NULL,

`jelszo` varchar(80) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_hungarian\_ci;

ALTER TABLE `admin`

ADD PRIMARY KEY (`id`);

ALTER TABLE `admin`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

## Használati esetmodell, szerepkörök

Alap felhasználói jogosultságokon felül akciós kuponokhoz is hozzáfér, szállítási adatai mentésre kerülnek.

Látogató, az oldal böngészésére és az ételek rendelésére jogosult.

Regisztrált felhasználó

Vendég felhasználó

Adatbázis karbantartása, termékek, kuponok, akciók módosítása, törlése, rendelések megtekintése.

Adminisztrátor

Webes felület

Asztali alkalmazás

## Modulok ismertetése

### Webes alkalmazás

#### Főoldal

A képen szöveg, menü, étel látható

Automatikusan generált leírás Nem bejelentkezett felhasználó számára:

A képen szöveg, étel, képernyőkép, menü látható

Automatikusan generált leírás Bejelentkezett felhasználó számára:

A termékek megjelenítésére szolgáló metódus:

A termékek megjelenítését megvalósította: Fedor Benjámin

#### Regisztráció

A regisztrálási űrlap 3 részre van bontva, melyek közt lapozni lehet, az adatok tovább lépés előtt ellenőrzésre kerülnek, és csak helyes adat esetén engedi tovább a felhasználót.

Az utolsó oldalon lévő „Regisztráció” gomb megfelelő adatok megadása esetén regisztrálja a felhasználót, és a Bejelentkezés oldalt jeleníti meg. Nem megfelelő adatok megadásakor értesíti a felhasználót a sikertelen regisztrációról és a regisztrációs űrlap elejére irányítja.

Sikeres regisztráció esetén az adatok az adatbázisban eltárolásra kerülnek, a felhasználó pedig értesítést kap a regisztráció sikerességéről.

#### Bejelentkezés

„Bejelentkezés” gomb helyes adatok megadása esetén a Főoldalra irányítja a bejelentkezett felhasználót. Helytelen adatok esetén értesíti a felhasználót a sikertelen bejelentkezésről.

A bejelentkezett felhasználó e-mail címe SESSION-ben kerül tárolásra, a továbbiakban ez alapján minden fontos adata lekérdezhető az adatbázisból. (pl: szállítási cím, telefonszám)

A regisztrációs és bejelentkezési folyamatot megvalósította: Turza Norbert és Kertész Kornél.

Az oldal megjelenéséért és használatáért felelős: Fedor Benjámin

#### Kosár:

A kosár tartalma a főoldalon és a kosár oldalon is módosítható, A kosár adatai SESSION-ben kerülnek eltárolásra.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásEgy termék hozzáadása a kosárhoz Javascriptben jQuery segítségével:

A kosár oldalt megvalósította: Fedor Benjámin

#### A képen szöveg, képernyőkép, Téglalap, szám látható Automatikusan generált leírásAdatok megadása

Ezen oldalon a felhasználó a rendeléshez szükséges adatokat tudja megadni, regisztrált és bejelentkezett felhasználó esetén a szükséges adatok automatikus kitöltésre kerülnek, ezt azonban a felhasználó szabadon megváltoztathatja.

Az adatok megadására szolgáló oldalon lehetősége van a felhasználónak a vissza nyílra kattintva visszalépni a Kosár oldalra, illetve a tovább nyílra kattintva sikeres adatmegadás esetén a Rendelés véglegesítésére szolgáló oldalra lépni.

Az adatok megadása oldal backendjét készítette: Kertész Kornél.

A Frontendet készítette: Fedor Benjámin.

#### A képen szöveg, képernyőkép, szám látható Automatikusan generált leírásRendelés véglegesítése

A rendelési folyamat szempontjából a végső oldal, a felhasználó megtekintheti az általa megadott rendeléshez kapcsolódó adatokat, valamint a rendelés tételeit, és minden hozzátartozó információt.

A „Rendelés” gombbal véglegesítheti rendelését, mely ezután feldolgozásra, és az adatbázisban való rögzítésre kerül. A felhasználó visszajelzést kap a rendelésének feldolgozásáról.

A Rendelés feldolgozásáért felelős: Kertész Kornél

A Rendelés véglegesítése weboldal megjelenéséért felelős: Fedor Benjámin

#### Felhasználói adatok módosítása

A bejelentkezett felhasználónak lehetősége van megváltoztatni telefonszámát, e-mail címét, illetve szállítási címét. Az oldal a főoldalról nyitható meg, a jobb felső sarokban található fiók ikon segítségével.

Az adatok módosításáért felelt: Kertész Kornél

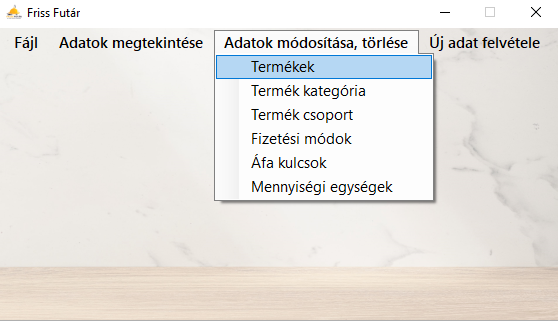
### Asztali alkalmazás

#### Bejelentkezés

#### 

A „Bejelentkezés” gomb helyes adatok megadása esetén a főablakra irányítja a bejelentkezett admint. Helytelen adatok esetén értesítést küld.

#### Főablak

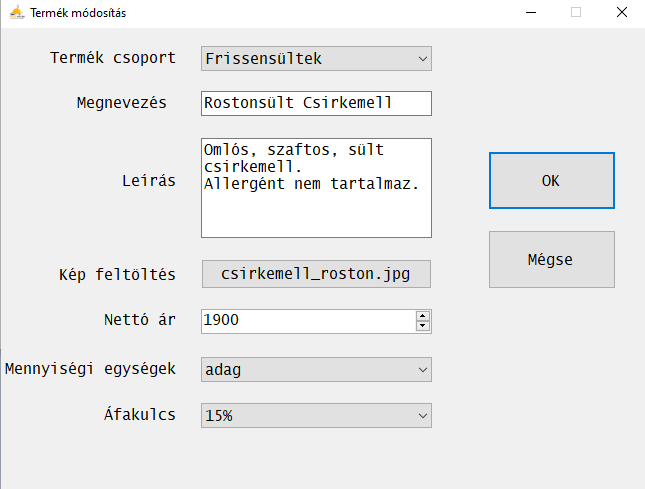


A „Fájl” menüpont alatt tálálható a „Kilépés” amivel bezárul az applikáció.

#### Adatok megtekintése

Az „Adatok megtekintése” menüpont alatt megtekinthetjük az adatbázis összes táblájának az adatait egyenként.

#### Adatok módosítása, törlése



Az „Adatok módosítása, törlése” segítségével módosíthatunk és törölhetünk a jövőben szükséges változtatásokat végezhetjük el az adatbázis bizonyos tábláin. Bármelyik lehetőségre kattintva megnyílik egy új form amin 3 gomb lesz elérhető:

„Módosít”:

A gomb megnyomásakor az előtte megadott adatok alapján egy következő formot nyit meg, ahol külön-külön változtathatók a tábla adatai, az „OK” gombbal elfogadja és feltölti az adatokat az adatbázisba, a „Mégse” gomb bezárja ezt a formot és az előző formra irányít.

„Töröl”:

A kiválasztott adatot törli az adatbázisból.

„Mégse”:

Bezárja ezt a formot és a főformra irányít.

#### Új adat felvétele

Az „Új adat felvétele” a megadott táblákat új adatokkal tölti fel. Bármelyik menüpontra kattintva megnyit egy új formot amin 2 gomb található:

„Feltölt”: Az új adatokat feltölti az adatbázisba.

„Mégse”: Bezárja a formot és visszairányít a főformra.

Az admin applikációt megvalósította: Kertész Kornél és Turza Norbert

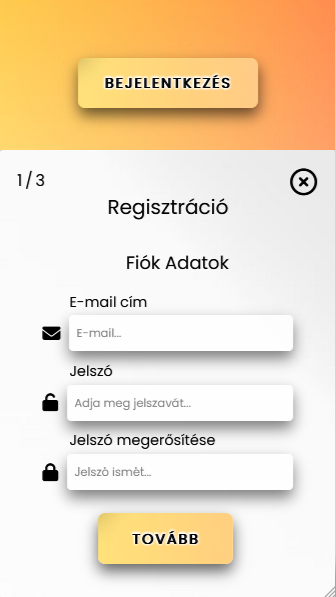
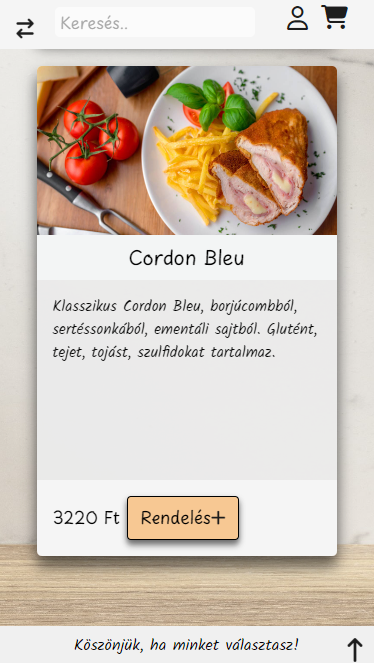
## Tesztelések

### Reszponzivitás

Minden oldal reszponzív, asztali gépen, telefonon, tableten tesztelve.

Hibák:

Néhány bug előfordul a különböző böngészőknél, amikor visszalép a felhasználó, nem mindig a kijelző méretének megfelelően jelenik meg az oldal, ezt a hibát a telefon elfordítása, majd visszafordítása helyreállítja.



Egy -a reszponzivitás megvalósításáért felelő kódrészlet javascriptben:

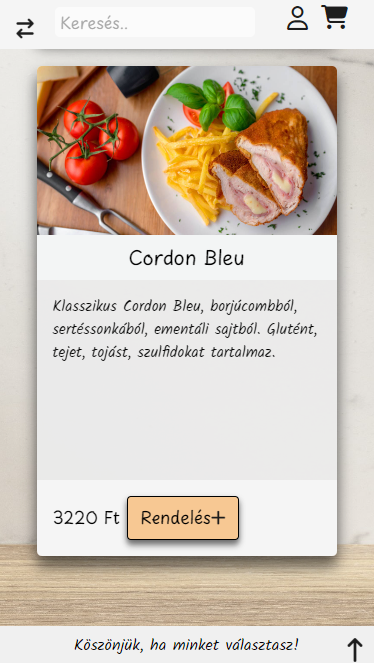
Az oldal tartalmának szélessége dinamikusan változik átméretezéskor, ha egy kritikus szélesség alá csökken, az oldalsó menü is behúzásra kerül (telefonoknál alapértelmezett)

Az oldal betöltésekor is hasonló vizsgálat zajlik le.

A weboldal reszponzivitását megvalósította: Fedor Benjámin

Tesztelte: Turza Norbert és Kertész Kornél

### Téma váltás



A főoldalon lehetősége van a felhasználónak a sötét és világos téma közt váltani, az éppen használt témát a program eltárolja, és az oldal legközelebbi látogatása esetén ezt használja, valamint a további oldalak megnyitásakor is a felhasználó által beállított témát használja.

Hibák:

Google Chrome-nál előforduló hiba, hogy az aloldalak megnyitásakor, az alapértelmezett témát használja az oldal, ennek okára még nem jöttünk rá.

A fenti hibán kívül a funkció megfelelően működik.

 Egy kódrészlet a téma mentésére, majd későbbi beállítására Javascriptben:

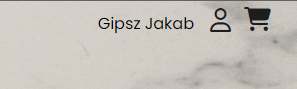
A témaváltásért, és az oldal megjelenését megvalósította: Fedor Benjámin

### Adatok kezelése

A felhasználó visszajelzést kap az általa megadott adatok sikeres feldolgozásáról, illetve arról is, ha az adatok nem megfelelőek.

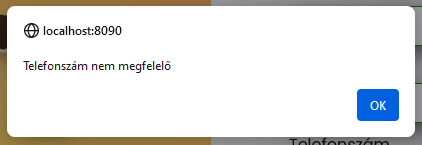
#### Bejelentkezésnél

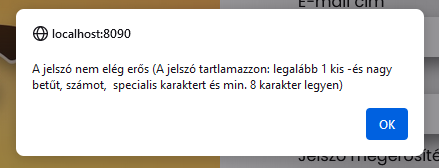
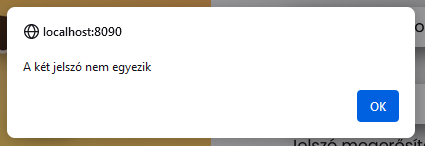
 Helytelen jelszó vagy e-mail cím megadása esetén a felhasználó értesítést kap:

Sikeres bejelentkezést követően a főoldalon többek között a felhasználó neve kerül megjelenítésre.

#### Regisztrációnál

A regisztrációs folyamat 3 részre van bontva, az adatok ellenőrzése részenként történik, ezáltal helytelen adatmegadás esetén a felhasználónak csak az aktuális modul adatait kell újra megadnia, ezzel elősegítve a gördülékenyebb felhasználói élményt.

Néhány visszajelzés a felhasználó számára, nem megfelelő adat megadása után:

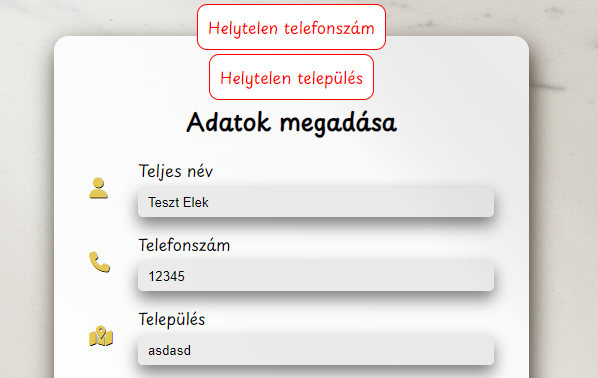


Visszajelzés sikeres regisztráció esetén:

A regisztráció és bejelentkezés backendje: Kertész Kornél és Turza Norbert

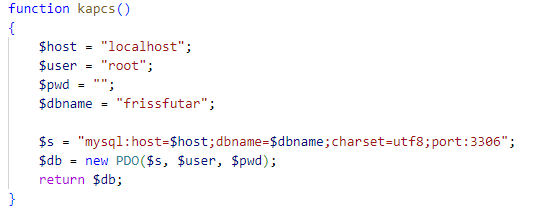
A regisztráció és bejelentkezés folyamatának tesztelése: Kertész Kornél és Fedor Benjámin

#### Rendelési adatok

A rendelési adatok megadása esetén is kap visszajelzést a felhasználó, amennyiben az általa megadott adat nem megfelelő.

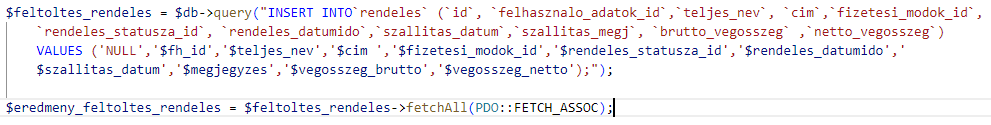
### Adatbázis

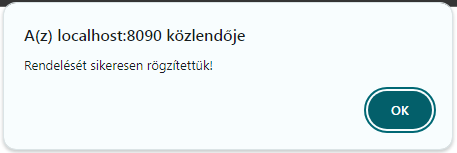
#### Kapcsolat az adatbázissal

Az adatbázishoz történő kapcsolódás php-ban PDO-val történik.

#### Adatok feltöltése az adatbázisba

 A project elején a termékek feltöltése php eljárásokon keresztül történt.

 Adatok feltöltése rendelés leadása esetén.

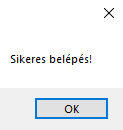


#### Adatok lekérdezése

Az adatbázisban lévő, a felhasználó számára minden lényeges adat lekérdezésre kerül a főoldal megnyitásakor, és php tömbökben kerül eltárolásra (pl. termékek, termék kategóriák stb..). Ennek segítségével jelenítjük meg a termékeket, képeket, menüpontokat az oldalon.

### Admin applikáció tesztelése

#### Bejelentkezés

 Visszajelzés sikeres és sikertelen belépés esetén:

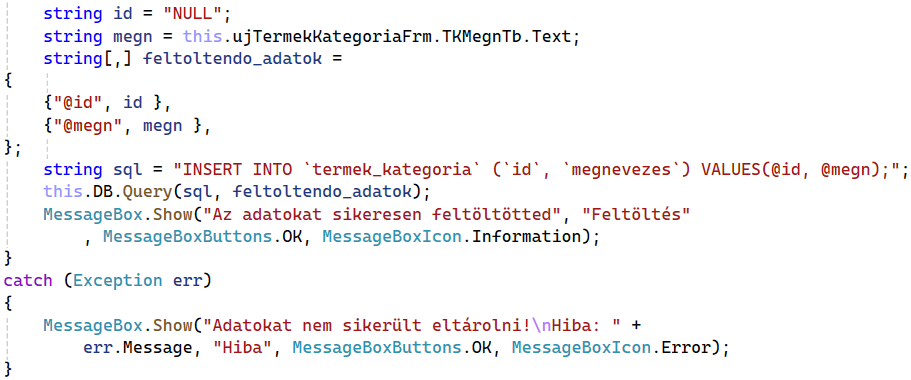
#### Adatbázis



Az adatbázishoz való kapcsolódás a connect.ini fájlban tárolt adatok segítségével történik, ami egyszerűbbé teszi a jövőben esedékes változtatások beállítását.

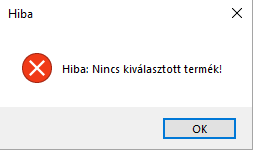
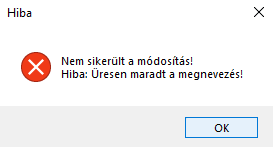
#### 

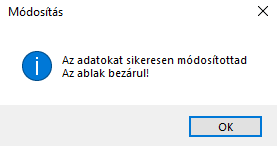
#### Adatok feltöltése adatbázisba



Az adatokat két dimenziós tömbökben tárolunk, amit később az SQL kódban felhasználunk az adatbázis feltöltéséhez.

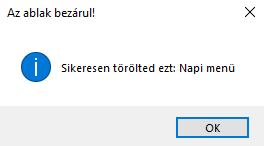
#### Módosítás

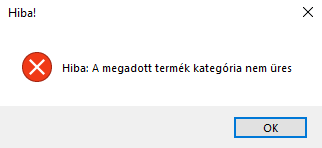


Ha vannak kötelezően kitöltendő mezők , akkor az alkalmazás hibaüzenetet küld és nem hajtja végre a változtatásokat.

Sikeres módosítás esetén visszajelzést kap a felhasználó a programtól.

#### Törlés

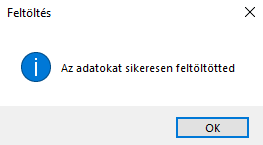
A program megerősítést kér a törlésről, ezzel elkerülve a véletlen történő adatvesztést.

Sikeres törlésnél visszajelzést küld.

A program nem minden esetben enged törölni, ilyenkor hibaüzenetet kap az admin.

#### Új adatok

Üresen hagyott mezők esetén, az alkalmazás hibaüzenetet küld és nem tölti fel az új adatot az adatbázisba.



Sikeres feltöltés esetén azonban visszajelzést küld a program a felhasználónak a végbement adatfeltöltésről.

Az admin alkalmazást készítette: Turza Norbert és Kertész Kornél

Az admin alkalmazás tesztelése: Turza Norbert

## Továbbfejlesztési lehetőségek

### Keresés funkció megvalósítása

A főoldalon a bal felső keresés modullal lehessen keresni bizonyos ételre/italra.

### Fizetési módok kezelése

Az Adatok megadásánál rendeléskor miután a megfelelő fizetési mód kiválasztásra került, ennek függvényében jelenjen is meg a megfelelő online fizetési platform.

### Képek nem kizárólagosan az „img” mappából történő megjelenítése

### Rendelés visszaigazolás, email küldése

A rendelés leadásakor küldjön visszaigazoló emailt a felhasználónak a rendelés felvételéről és a várható kiszállítási időtartamról.

### Rendelések megtekintése külön webes felületen (admin)

### Adatbázis szerveren történő működtetése

### Bug-ok javítása

### Program feltelepítése, adatbázis mentése, frissítése

### Kiemelt ajánlatok kezelése

# **Felhasználói dokumentáció**

## Ismertetés

A Frissfutár egy étel - és italrendelésre használatos weboldal, ami számítógépen, tableten, telefonon egyaránt elérhető bárki számára, aki rendelkezik internetkapcsolattal.

## Rendszerkövetelmények

Minimum:

Telefon/Tablet: Android 8.0, iOS 12

Stabil internetkapcsolat

## Alkalmazás telepítése, beállítása

## Alkalmazás használata

### Webes alkalmazás használata

#### Főoldal

A bal felső sarokban található oldalsó menü előhívó és eltüntető gomb.

Keresés modul a bal felső sarokban.

A jobb felső kosár gomb megnyitja a kosár előnézetét.

A „Kosár ürítése” gomb törli a kosár teljes tartalmát.

A „Tovább a rendeléshez” gomb átirányít a kosár oldalra.

A „Fiók” menüpont alatt található

Mindkét menüpont a bejelentkezés/regisztráció oldalra irányít át.

Az „Étlap” menüpont alatt található

„Ételek” menüpont megjeleníti az összes étel kategóriát.

„Italok” menüpont az ital típusokat jeleníti meg.

A bizonyos étel és ital kategóriákra kattintva az adott típusú ételhez vagy italhoz navigálja a felhasználót.

A „Kapcsolat” menüpont az elérhetőségi adatokhoz irányítja a felhasználót.

A „Kuponok” menüpont bejelentkezett felhasználók számára elérhető, és az akciós kuponokhoz navigál.

A kosár gomb mellett található fiók ikon bejelentkezett felhasználóknak jelenik meg, és az alábbi menüpontokat tartalmazza:

Kijelentkezés – A felhasználó kijelentkeztetése.

Adatok módosítása – Az adatmódosító aloldalra irányítja a felhasználót, ahol szállítási, illetve bejelentkezési adatainak módosítására van lehetősége.

A jobb alsó sarokban található felfelé mutató nyíl az oldal tetejére irányítja a felhasználót.

A termékekhez tartozó „Rendelés” gomb hozzáad a kosárhoz a választott termékből egyet.

Az oldalsó menü utolsó menüpontjával a sötét és világos téma közt lehet váltani, mely választott témát a későbbiekben a többi aloldal is alkalmazza.

#### Bejelentkezés és Regisztráció

Az oldal tetején található „x” ikonra kattintva a Főoldalra irányítja a felhasználót.

A „Bejelentkezés” és „Regisztráció” gombokkal a regisztráció és a bejelentkezés űrlap között lehet váltani.

A képen szöveg, képernyőkép, tervezés, Grafikus tervezés látható

Automatikusan generált leírásRegisztráció:

A „Tovább” gomb megfelelő adatok megadása esetén a regisztrációs űrlap következő oldalára lapoz.

A „Vissza” gomb az előző oldalra lapoz.

Az utolsó oldalon lévő „Regisztráció” gombbal tudja véglegesíteni a regisztrációját a felhasználó.

A képen szöveg, képernyőkép, Téglalap, Betűtípus látható

Automatikusan generált leírásBejelentkezés:

#### A képen szöveg, képernyőkép, étel látható Automatikusan generált leírásKosár oldal

A „Vissza” gomb a Főoldalra irányítja a felhasználót.

A „Kosár ürítése” gomb törli a kosár tartalmát, és visszalépteti a felhasználót a Főoldalra.

A „Tovább” gomb a rendelési adatok megadására szolgáló oldalra irányít.

A kosárban lévő termékek mennyisége mellett található „ + ” és „ - ” gombok segítségével növelhető illetve csökkenthető a megrendelni kívánt termék mennyisége.

#### A képen szöveg, képernyőkép, Téglalap, szám látható Automatikusan generált leírásRendelési adatok megadása

A rendeléshez szükséges adatok megadására szolgáló oldal.

Bejelentkezett felhasználónál az adatok automatikus kitöltésre kerülnek, azokat természetesen változtatni lehet.

#### Rendelés befejezése

*A képen szöveg, képernyőkép, szám látható

Automatikusan generált leírás*

A rendelés leadására szolgáló oldal, melyen a felhasználó áttekintheti a rendelni kívánt tételeket, és minden fontos információt.

A „Vissza” gombbal a felhasználó visszaléphet az Adatok megadása oldalra.

A „Rendelés” gombbal véglegesítheti rendelését.

#### Felhasználói adatok módosítása

A képen szöveg, képernyőkép, Betűtípus, névjegykártya látható

Automatikusan generált leírásA bejelentkezett felhasználónak lehetősége van megváltoztatni telefonszámát, e-mail címét, illetve szállítási címét. Az oldal a főoldalról nyitható meg, a jobb felső sarokban található fiók ikonra, majd a lenyíló menün belül a beállítások menüpontot választva.

### A képen szöveg, képernyőkép, Betűtípus látható Automatikusan generált leírásAsztali alkalmazás használata

A „Fájl” menüpont alatt tálálható a „Kilépés”, amivel bezárul az applikáció.

Az „Adatok megtekintése” menüpont alatt megtekinthetjük az adatbázis összes táblájának az adatait egyenként.

Az „Adatok módosítása, törlése” segítségével módosíthatunk és törölhetünk a jövőben szükséges változtatásokat végezhetjük el az adatbázis bizonyos tábláin. Bármelyik lehetőségre kattintva megnyílik egy új form amin 3 gomb lesz elérhető:

„Módosít”:

A gomb megnyomásakor az előtte megadott adatok alapján egy következő formot nyit meg, ahol külön-külön változtathatók a tábla adatai, az „OK” gombbal elfogadja és feltölti az adatokat az adatbázisba, a „Mégse” gomb bezárja ezt a formot és az előző formra irányít.

„Töröl”:

A kiválasztott adatot törli az adatbázisból.

„Mégse”:

Bezárja ezt a formot és a főformra irányít.

Az „Új adat felvétele” a megadott táblákat új adatokkal tölti fel. Bármelyik menüpontra kattintva megnyit egy új formot amin 2 gomb található:

„Feltölt”: Az új adatokat feltölti az adatbázisba.

„Mégse”: Bezárja a formot és visszairányít a főformra.

# **Irodalomjegyzék**

## Internetes források

<https://stackoverflow.com/>

<https://www.php.net/>

<https://www.w3schools.com/>

<https://codepen.io/>

<https://www.geeksforgeeks.org/>

<https://playground.com/>

[https://uxplanet.org/](https://uxplanet.org/phone-number-field-design-best-practices-23957cbd86d5)

<https://fontawesome.com/>

<https://getbootstrap.com/>

<https://learn.microsoft.com/>

[www.javascripttutorial.net](http://www.javascripttutorial.net)

<https://phppot.com/>

# **Mellékletek**

## Github

A teljes vizsgaremek publikus github linkje:

## Adatbázis

Az adatbázis exportja:

A kiürített adatbázis exportja:

## Xampp

Az adatbázissal való kapcsolathoz szükséges a xampp alkalmazás, mely az alábbi linkről tölthető le:

## Dokumentáció

Dokumentáció linkje: