Tartalom

[1. Bevezetés 3](#_Toc158374777)

[2. Felhasználói dokumentáció 4](#_Toc158374778)

[2.1. Felhasználói fiók 4](#_Toc158374779)

[2.1.1. Regisztráció 4](#_Toc158374780)

[2.1.2. Bejelentkezés 7](#_Toc158374781)

[2.1.3. Vendég felhasználók 8](#_Toc158374782)

[2.1.4. Regisztrált és bejelentkezett felhasználok számára elérhető funkciók 9](#_Toc158374783)

[2.1.5. Beállítások oldal 10](#_Toc158374784)

[2.1.6. Kedvenc ételek és italok oldal 12](#_Toc158374785)

[2.1.7. Foglalások oldal 12](#_Toc158374786)

[2.1.8. Főoldal 13](#_Toc158374787)

[2.1.9. Asztalfoglalás oldal 14](#_Toc158374788)

[2.1.10. Kínálatunk oldal 16](#_Toc158374789)

[2.1.11. Termék részletei oldal 18](#_Toc158374790)

[2.1.12. Kosár oldal 19](#_Toc158374791)

[3. Fejlesztői dokumentáció 21](#_Toc158374792)

[3.1. Munkamegosztás 21](#_Toc158374793)

[3.1.1. Feladatok elosztása 21](#_Toc158374794)

[3.2. Verziókezelés 21](#_Toc158374795)

[3.3. Frontend 22](#_Toc158374796)

[3.3.1. Dizájn tervek 22](#_Toc158374797)

[3.4. Használt technológiák 23](#_Toc158374798)

[3.4.1. Angular 23](#_Toc158374799)

[3.4.2. Angular és Bootstrap 5.3 25](#_Toc158374800)

[3.4.3. Reszponzivitás 25](#_Toc158374801)

[3.4.4. Főoldal 25](#_Toc158374802)

[3.5. Backend 25](#_Toc158374803)

[3.5.1. Használt technológiák 25](#_Toc158374804)

[3.5.2. Laravel 25](#_Toc158374805)

[3.5.3. Laravel Sanctum 26](#_Toc158374806)

[3.5.4. Laravel Eloquent 26](#_Toc158374807)

[3.5.5. MySQL 26](#_Toc158374808)

[3.6. Adatbázismodell 27](#_Toc158374809)

[3.6.1. Allergens tábla 29](#_Toc158374810)

[3.6.2. Categories tábla 29](#_Toc158374811)

[3.6.3. Desks tábla 29](#_Toc158374812)

[3.6.4. Event\_logs tábla 30](#_Toc158374813)

[3.6.5. Images tábla 31](#_Toc158374814)

[3.6.6. Ingredients tábla 31](#_Toc158374815)

[3.6.7. Ingredient\_allergen kapcsolati tábla 31](#_Toc158374816)

[3.6.8. Password\_reset\_tokens tábla 32](#_Toc158374817)

[3.6.9. Personal\_access\_tokens tábla 32](#_Toc158374818)

[3.6.10. Products tábla 33](#_Toc158374819)

[3.6.11. Product\_ingredient kapcsolati tábla 34](#_Toc158374820)

[3.6.12. Product\_purchase kapcsolati tábla 34](#_Toc158374821)

[3.6.13. Product\_user kapcsolati tábla 35](#_Toc158374822)

[3.6.14. Purchases tábla 35](#_Toc158374823)

[3.6.15. Reservations tábla 36](#_Toc158374824)

[3.6.16. Users tábla 37](#_Toc158374825)

# Bevezetés

A Menu Genius egy ételrendelő program elsősorban, amely lehetőséget biztosít a felhasználó számára, hogy előre tudjon asztalt foglalni saját magának illetve a vele érkező személyeknek egyaránt. Fontos szempont volt a projekt tematikájának kiválasztását illetően, hogy sok embernek nyújtson egyszerű és korszerű megoldást. Sok étteremben, főként franchise-okban fellelhető már ez az ötlet, viszont azok az adott cégnek a saját maguk által fejlesztett egyedi, zárt rendszerek. Amit mi készítettünk az egy „template”, azaz egy sablon, amelyet az adott étteremnek a saját ételeivel, megjelenésével és egyéb paramétereivel egyetemben vagyunk képesek átalakítani.

Az általunk készített programnak az elsődleges célja, hogy bárki számára átlátható és könnyen kezelhető felületet biztosítson, az asztalnak a lefoglalásától kezdve, egészen az allergének böngészéséig és szűréséig. Mindenki számára elérhető az asztalfoglalás és böngészés funkció is. Így aki csak hirtelen pillantást szeretne vetni az adott étteremben levő kínálatra, az regisztráció nélkül is gond nélkül megteheti, ahogyan akár egy asztalnak a lefoglalását is egyszerűen megoldhatja mindenféle előzetes regisztrációt illetve bejelentkezést „átugorva”.

A könnyű használhatóságot rendkívüli mértékben segíti, hogy a felhasználói kezelőfelület letisztult és könnyen értelmezhető minden egyén számára. Felül a navigációs sávon vannak az oldalon való navigációhoz szükséges gombok nagyrésze. Ide tartozik a regisztrációs gomb, és a bejelentkezést követően a felhasználó saját adatihoz irányító gomb. Továbbá olyan beépített eszközök segítik a zökkenőmentes kezelést, mint például az ételkategória szerinti keresés, valamit allergének által szűrhető böngészés.

A MenuGenius letöltés nélkül elérhető. A program webes technológiára épül, így amennyiben az okoseszköz rendelkezik modern böngészővel, a program használható. Gyengébb rendszereken is gond nélkül futtatható és semmilyen felhasználói élmény romlás nem tapasztalható.

Kiemelt szerepet kapott a projekt megvalósításának során, hogy naprakész technológiákat használjunk, ezzel is fejlesztve a saját tudásunkat a jövőt illetően a piacon. A választásunk - a kliensoldali részt illetően - az Angular-ra esett. Az Angular egy JavaScript alapú keretrendszer, amelyet világszerte rengetegen használnak a fejlesztéseik során. Jelenleg a 2. legnépszerűbb webfejlesztői környezet.[[1]](#footnote-1) Szerver oldalon a világ egyik legkeresetteb fejlesztői környezetére, a Laravel-re hagyatkoztunk. Előszeretettel alkalmaznak startup-ok, kis- és nagyvállalatok is Laravel fejlesztőket, az igény erre a tudásra évről-évre nő. 743.470 oldal készült ennek a technológiának a segítségével 2023-ban.[[2]](#footnote-2)

Sok új dologgal találkoztunk a projekt megvalósításának során, betekintést nyertünk, hogy hogyan is kell adatbázist felépíteni egy weboldalhoz, ezekhez API erőforrásokat kell kötni, vagyis a szerveroldallal összekapcsolni. Az Angular-nak köszönhetően - fejlesztői oldalról is - egy jól áttekinthető, könnyen értelmezhető és sok komponensre bontott eredményt voltunk képesek létrehozni.

Több ötletet is merítettünk az elkészítés során, jelenleg is üzemelő nagyobb étterem láncoknak weboldalaikról.[[3]](#footnote-3) (megjelenés struktúrája, ételrendelés menete, asztalfoglalás és különböző szolgáltatások)

A MenuGenius weboldal elérhetősége:

https://github.com/Celinayy/menuGenius

# Felhasználói dokumentáció

## Felhasználói fiók

### Regisztráció

A MenuGenius weboldalon lehetőség van a regisztrációra, amelyet a bal felső sarokban található kis ikon segítségével tudunk megoldani. Ez egy felugró ablakot jelenít meg és lehetőségünk lesz a Regisztráció gombra kattintani.



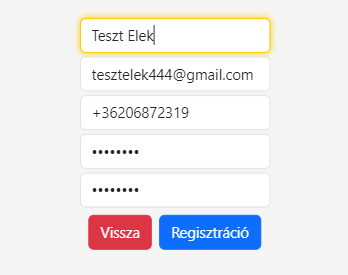
. ábra: Menüsor



. ábra: Elérhető opciók

Hibás adatok megadása nem lehetséges, mivel a regisztrációs űrlapon mindenre kiterjedő hibakezelés található. Amennyiben a felhasználó rossz adatot adna meg, azt a rendszer a megfelelő hibaüzenettel jelzi.

A regisztrációhoz a következő adatok szükségesek:

* Teljes név

3. ábra Regisztrációhoz szükséges adatok

* E-mail cím
* Telefonszám
* Jelszó
* Jelszó újra

Példa két rosszul kitöltött űrlap hibaüzenetét illetően, ahol először az e-mail címet, majd a jelszót töltöttük ki nem megfelelő adatokkal:

|  |  |
| --- | --- |
| 4. ábra: Regisztráció rossz email | . ábra: Regsiztráció rossz jelszó |

### Bejelentkezés

A weboldalra való bejelentkezéshez a bal felső sarokban levő kis ikont kell használnunk, ami aztán egy felugró ablakot jelenít meg számunkra, ahol lehetőség lesz kiválasztani a Bejelentkezés gombot.

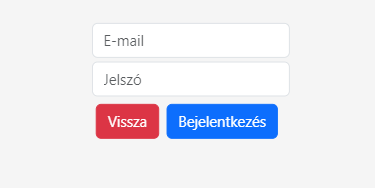


. ábra: Menüsor



. ábra: Elérhető opciók

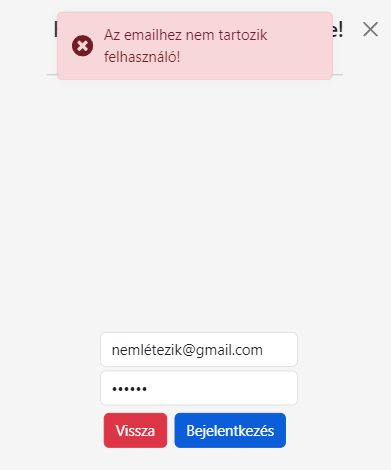
A bejelentkezéshez szükséges adatok:

* E-mail cím

8. ábra: Bejelentkezéshez szükséges adatok

* Jelszó

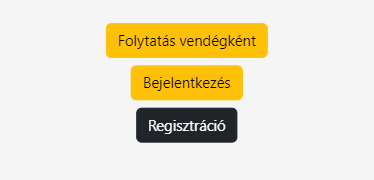
Kizárólag regisztrált felhasználók tudnak csak bejelentkezni a MenuGenius weboldalára, erről visszajelzést is kap hibaüzenet formájában, amennyiben még nem tette volna meg a regisztrációt:



9. ábra: Nem létező felhasználó

### Vendég felhasználók

Vendég felhasználóként nem kell regisztrálni illetve bejelentkezni sem a weboldalra. A főbb funkciói az oldalnak így is elérhetőek. Lehetséges az asztalfoglalás és a kínálatban levő keresés, az összes szűrési funkcióval együtt. „Folytatás vendégként” gombra kattintva tudjuk elérni ezeket a funkciókat.



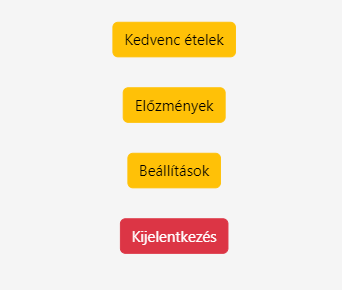
. ábra: Elérhető opciók

### Regisztrált és bejelentkezett felhasználok számára elérhető funkciók

A MenuGenius által nyújtott plusz funkciók a bejelentkezés után elérhetővé válnak a felhasználó számára, megjelenik a felső navigációs sávban, jobb felül, egy hamburger ikon, amelyen belül a *Kedvenc ételek, Előzmények, Beállítások* és a *Kijelentkezés* gomb lesz található.



. ábra: Navigációs sáv Hamburger ikon



. ábra: Elérhető plusz funkciók

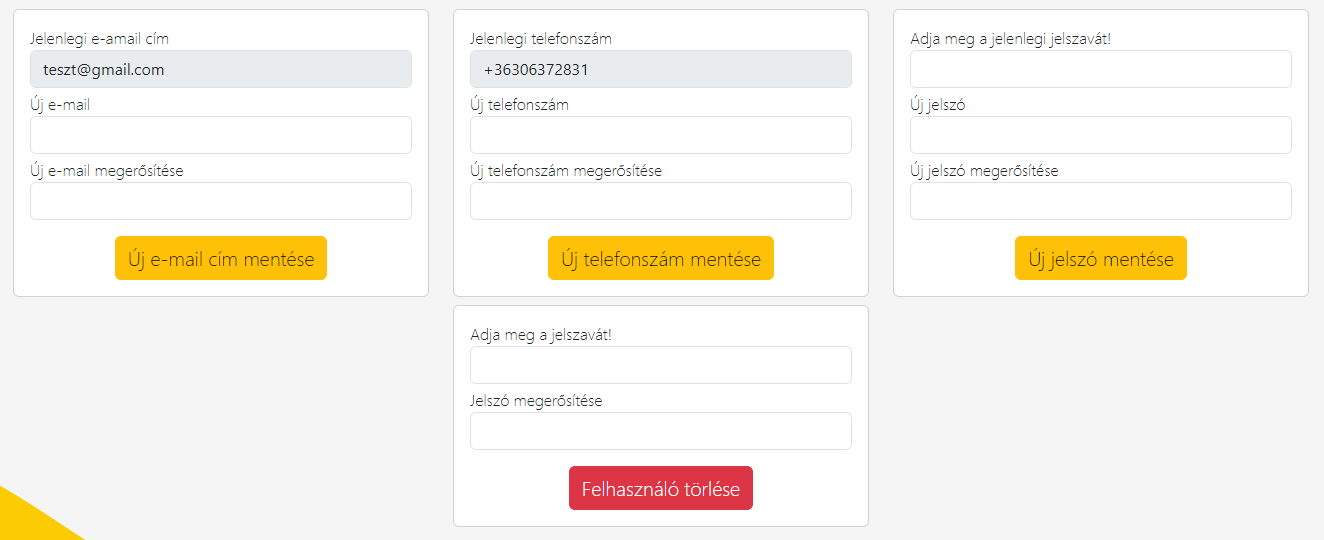
A navigációs sávon megjelenik egy bevásárlókosár ikon is, ezáltal a felhasználó tudja, hogy bejelentkezett állapotban van.



. ábra: Bejelentkezett állapot

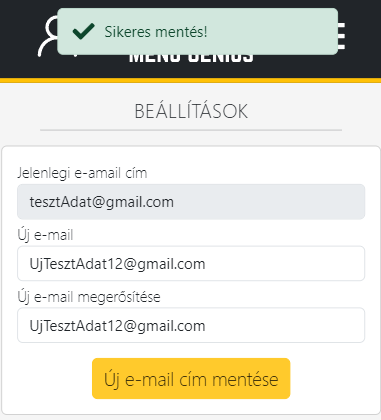
### Beállítások oldal

A *Beállítások* gombra kattintva megjelennek a regisztráció során, a felhasználó által megadott adatok automatikusan. Ezen oldalon belül a felhasználónak jogosultsága van az E-mail címének, a telefonszámának, valamint a jelszavának a megváltoztatására és még a felhasználójának a törlésére is lehetősége van.



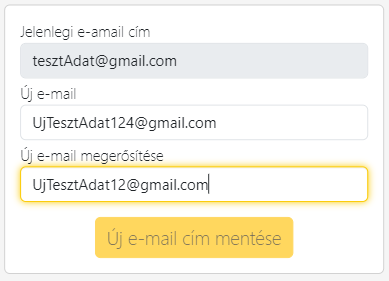
. ábra: Beállítások

Amennyiben az adott személyes adat megváltoztatása sikeresen végbe ment, arról a felhasználó egy visszajelzést is kap, amely az alábbi képernyőfotón látható:



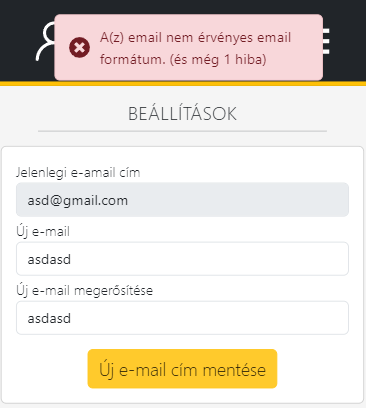
. ábra: Sikeres mentés

Nem azonos adatok megadásáról azonnali visszajelzést kap a felhasználó, azáltal, hogy nem lesz kattintható számára az új adat mentése gomb:



16. ábra: Nem azonos adatok

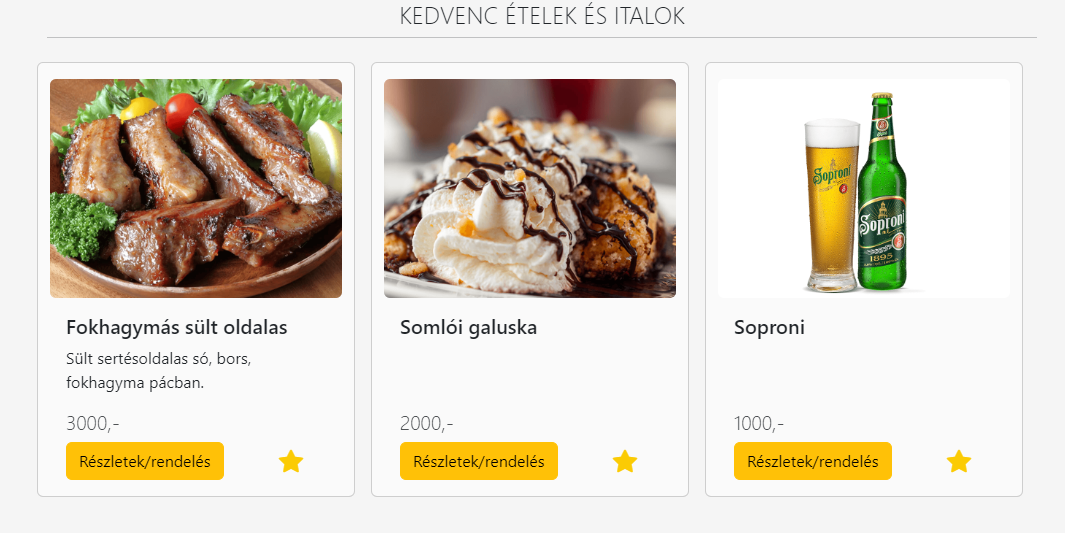
Amennyiben a felhasználó rosszul adta meg az adott adatot, arról is visszajelzés látható egy hibaüzenet formájába. Minden mezőnek egyedi hibaüzenetei vannak a hibának megfelelően, így a felhasználó egyből tudni fogja, hogy mit kell javítania:



. ábra: Egyedi hibaüzenet

### Kedvenc ételek és italok oldal

A *Kedvenc ételek és italok* gombra kattintva megjelennek a felhasználó által kiválasztott ételek és italok a kínálatból, ez segíthet a későbbiekben a gyors étel és ital rendelésben.



. ábra: Kedvenc ételek oldal

A csillag embléma segítségével egy kattintással tudja a *Kedvencekhez* adni az ételeit és italait a felhasználó, és ugyanígy eltávolítani is.

### Foglalások oldal

Amennyiben a felhasználó foglalt már asztalt, akkor az a *Foglalások oldalon* fog megjelenni az általa megadott különböző adatokkal, innen könnyen elnavigálhat a kínálathoz, ameddig várakozik.



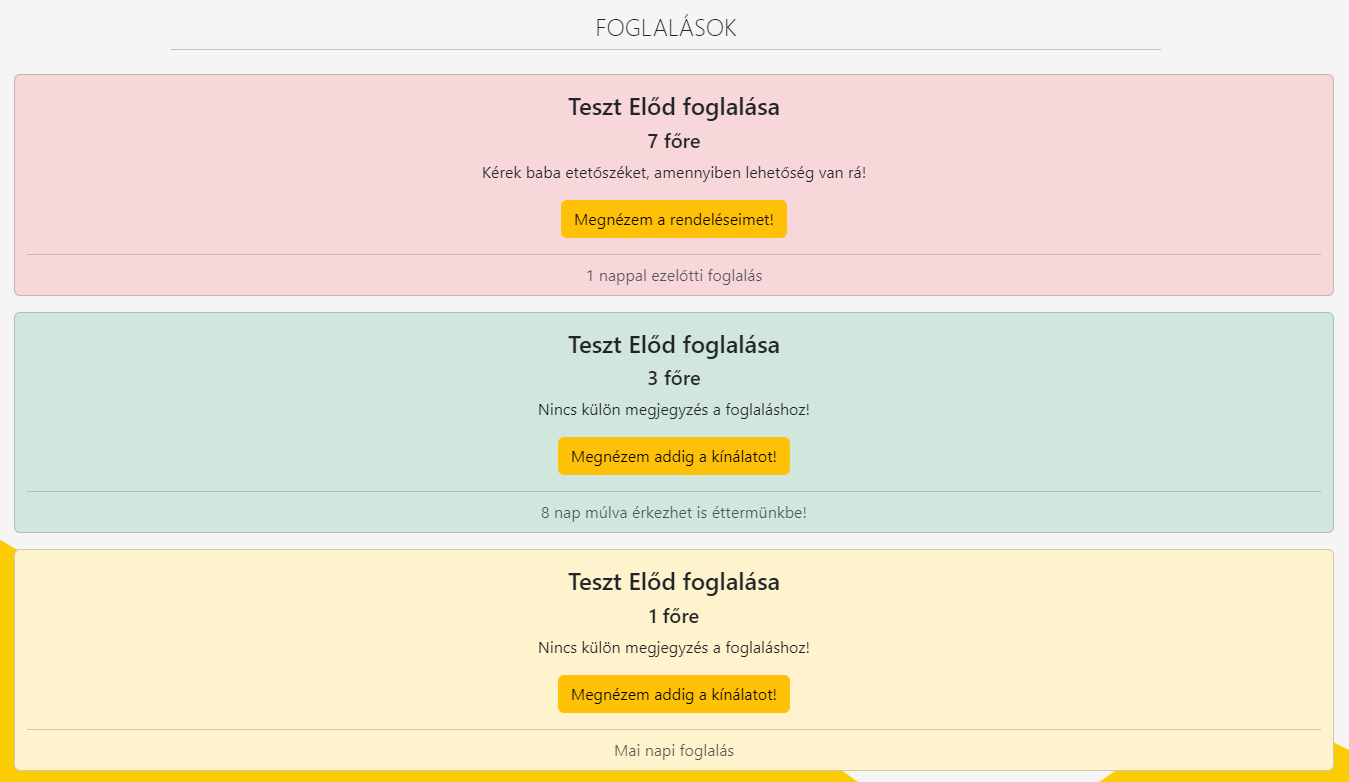
. ábra: Foglalás egyedi adatokkal

A *Foglalások oldalon* továbbra is megmaradnak az adatok, ha már elmúlt az adott időpont, így könnyen vissza lehet nézni, hogy miket rendelt azon a napon és milyen értékben fogyasztott.



. ábra: Elmúlt foglalás

A könnyű átláthatóság érdekében az elmúlt foglalások kártyáinak a háttere pirosra, a jövőben levőknek zöldre, míg az azon a napon levő foglalásoknak sárgára van állítva.



. ábra: Színjelzéses foglalások

### Főoldal

A főoldalon a megjelenés és a funkciók is mind a bejelentkezett és a nem bejelentkezett, vendég felhasználók számára is ugyanaz, a *Hamburger* és a *Bevásárlókosár* ikon kivételével.

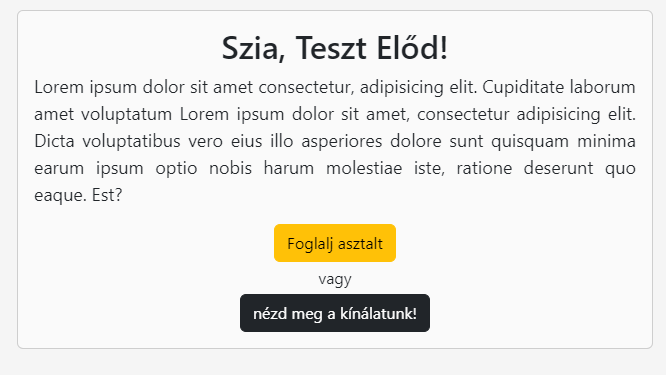


. ábra: Főöldal

A nyitóoldalon két darab kártya található, az egyik kártya egy rövid és könnyen értelmezhető leírást ad az oldalról, és a kettő gomb segítségével egyből el tudunk navigálni az *Asztalfoglaláshoz* illetve a *Kínálatot* is meg tudjuk tekinteni egy kattintás segítségével. A másik kártyában pedig véletlenszerűen jelennek meg a termékekről fotók, amelyek akár a választásban is segíthetnek, ugyanis, ha rákattint a felhasználó, akkor egyből a *Termék részleteihez* navigálja az oldal.

### Asztalfoglalás oldal

A felhasználó a ”*Foglalj asztalt”* gombra kattintva eljut az *Asztalfoglalás* oldalra, ahol asztalt tud foglalni.



. ábra: Navigációs gomb asztalfoglaláshoz

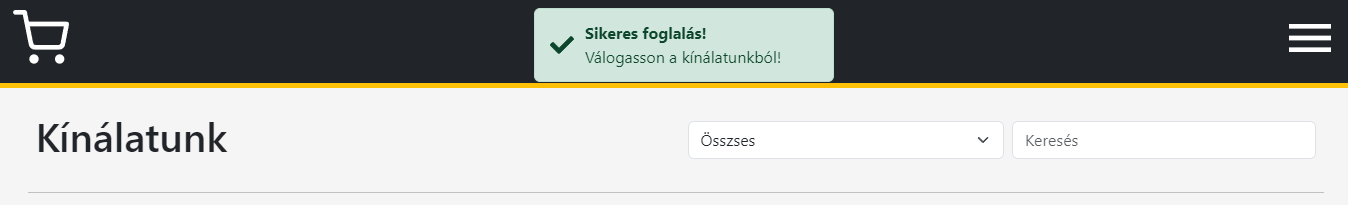
Amennyiben a felhasználó bejelentkezett állapotban van, úgy személyes adatai automatikusan kitöltésre kerülnek, valamint az adott napi dátum is betöltésre kerül, ezzel is segítve a zökkenőmentes és gyors űrlap kitöltést.

Az alábbi fotón jelölve van az automatikusan kitöltött rész:



. ábra: Automatikus mező kitöltés

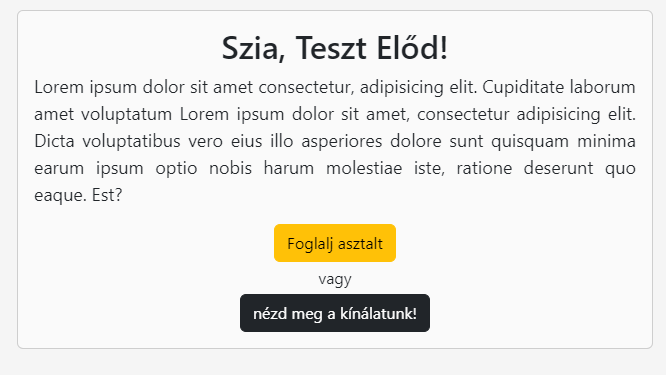
Ha a foglalás sikeres, akkor arról azonnali visszajelzést kap a felhasználó, ez egy kis felugró üzenetben jelenik meg a képernyő tetején, eközben az oldal el is irányítja az kínálathoz a felhasználót.



. ábra: Sikeres foglalás utáni állapot

### Kínálatunk oldal

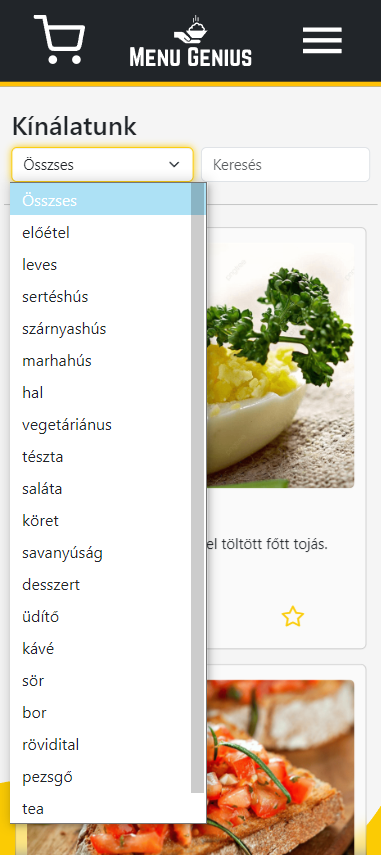
A felhasználó a főoldalon levő, *„nézd meg a kínálatunk”* gombra kattintva elnavigál a *Kínálatunk* oldalra az oldalon belül.



. ábra: Navigációs gomb a kínálathoz

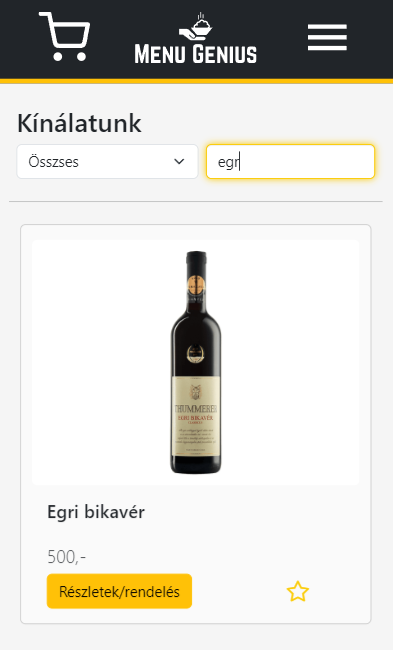
A *Kínálatunk* oldalon található az összes termék az adott éttermen belül, két lehetőség is segíti a felhasználót abban, hogy meg tudja találni a számára megfelelő terméket gyorsan és egyszerűen.

Egyik ilyen funkció, a kategóriák szerinti szűrés, amely alább látható:



. ábra: Kategória általi szűrés

A másik funkció pedig a *Keresés* mezőbe való gépeléssel érhető el, ez gomb lenyomásonként egyből frissül.



. ábra: Szűrés keresés funkcióval

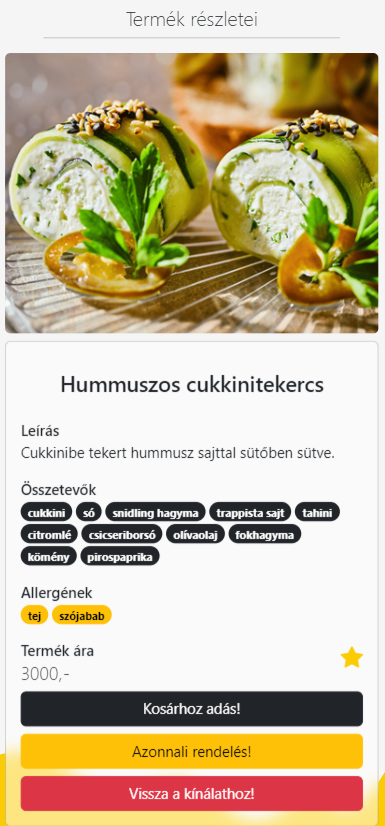
A *Kínálatunk* oldalona termékek kis kártyákba vannak kiszervezve, amelyben minden termékről egy kisebb ismertető látható, elsősorban a termék neve, leírása majd az ára. Ha a *„Részletek/rendelés”* gombra kattint a felhasználó, akkor elirányítja az oldal az adott terméknek a részletes leírásához, és plusz információkat tudhat meg az étellel kapcsolatosan, mint például az összetevőket és nem utolsó sorban azokról az allergénekről is információt szerezhet, amelyeket az adott termék tartalmaz.



. ábra: Termék kártya és a hozzátartozó gomb

### Termék részletei oldal

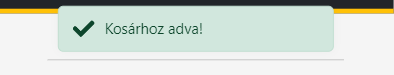
Ezen az oldalon sok új és hasznos információhoz juthat a felhasználó az adott termék részletes leírása, összetevői és allergénjei által.



. ábra: Termék részletei

További lehetőségek elérhetőek, ilyen a már ismert *Kedvencekhez* adás a csillag embléma által, valamint a „*Kosárhoz adás”* és a termék „*Azonnali rendelése”*. Ezek mellett természetesen, ha az adott termék nem nyerte el a tetszését a felhasználónak, akkor a „*Vissza a kínálathoz!”* gomb segítségével egyből újra a *Kínálatoknál* találja magát és tovább válogathat a termékek között.

A *„Kosárhoz adás!”* gombra kattintva egy üzenet fogadja a felhasználót, amennyiben sikeresen végbement a művelet. Annyiszor adódik a termék a kosárhoz, ahányszor a felhasználó rákattint a *„Kosárhoz adás!”* gombra, ez természetesen a *Kosár* megnyitásával módosítható lesz utólagosan, amennyiben mégsem kívánja az adott terméket vagy esetlegesen a termék mennyiségét szeretné módosítani.



. ábra: Sikeres kosárhoz adás visszajelzés

Amennyiben az *„Azonnali rendelés!”* gombra kattint a felhasználó, úgy az oldal egyből a *Kosárhoz* navigálja.

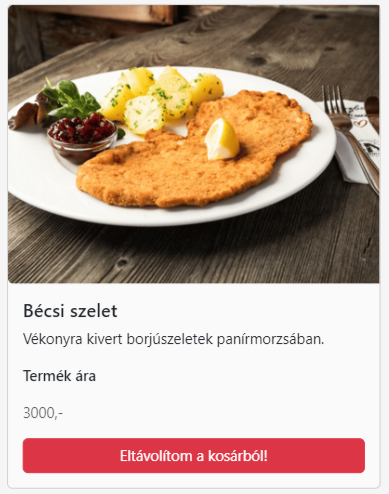
### Kosár oldal

A „*Kosár*” oldalra lépve, ha még nem adott hozzá a felhasználó terméket, akkor az annak megfelelő üzenet fogadja és egy gomb segítségével egyből a *Kínálathoz* tud navigálni.



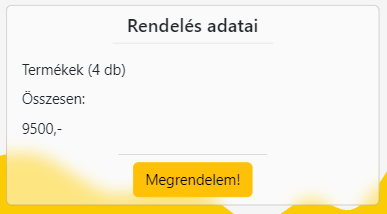
. ábra: Üres kosár

Termék hozzáadása utána *Kosárban* megjelennek a kiválasztott ételek és italok, amelyeket el tud távolítani a felhasználó az *„Eltávolítom a kosárból!”* gombbal, amennyiben mégsem kívánja azt megrendelni.



. ábra: Termék eltávolítás a kosárból

A *Kosár* oldalon a felhasználó, nyomon követheti egy kis összegző kártya által, hogy milyen mennyiségben és mekkora összegben válogatott össze termékeket.



# Fejlesztői dokumentáció

## Munkamegosztás

### Feladatok elosztása

A MenuGenius projektet első körben 3 fős csapattal terveztük elkezdeni, de végül 2 fővel mentünk tovább. A projekt legelején döntöttünk arról, hogy ki milyen szerepet fog vállalni a fejlesztés során. Letenyei Szilárd a backendért és az adatbázisért felelt. A backend Laravel-ben készült, amely egy PHP [[4]](#footnote-4)keretrendszer. Az adatbázis egy MySQL [[5]](#footnote-5)alapú relációs adatbázis, és a dbForge[[6]](#footnote-6) segítségével tudtuk vizuálisan is nyomon követni az adatokat benne. Vangel Celina Hanna felelt a frontendért, illetve a dokumentációért a projekt során. A frontendet az Angular keretrendszer 16-os verziójával valósítottuk meg. Különböző előzetes dizájn terveket szerkesztette Vangel Celina Hanna.

A használt grafikai szoftverek:

* Paint 3D
* GIMP
* Wireframe

## Verziókezelés

Párhuzamosan dolgoztunk ketten a projekt során. A munkánkat a GitHub [[7]](#footnote-7) verziókezelő szolgáltatással tudtuk zökkenőmentesen végezni. A GitHub segítségével figyelemmel tudtuk kísérni a projekt fejlődését és a módosításokat is könnyen nyomon lehetett követni.

A verziókezelés fogalma: *„A verziókezelés alatt több verzióval rendelkező adatok kezelését értjük.”* [[8]](#footnote-8)A verziókezelő programok a fejlesztendő dokumentumok, forráskódok, tervek és egyéb olyan adatok verzióinak kezelésére szolgálnak, amelyeken egyidejűleg számos fejlesztő dolgozik. A fájlok módosulását naplózza a rendszer. A nyilvántartás tartalmazza a módosító nevét, a módosítás idejét és egy „commit” üzenetet, amely egy összefoglaló leírás a módosításról. Megjegyzendő, hogy egy egyszemélyes fejlesztés során is hatalmas segítséget jelent egy verziókezelő szoftver használata.

A piacon több verziókezelő szoftver áll rendelkezésünkre, amelyek bizonyos funkciókban eltérnek, de az alapműködés és az alapfogalmak megegyeznek.

A GitHub segítségével lehetőség van több modul egyidejű fejlesztésére is, amelyeket „merge” funkció segítségével közös „branch” alá tudtunk hozni.

## Frontend

A frontend feladata az adatok megjelenítése, befogadása a felhasználó (ritkábban akár egy másik rendszer) felől. A programoknak, weboldalaknak az a része, amely a felhasználóval közvetlen kapcsolatban van.[[9]](#footnote-9)

A frontend járul hozzá a felhasználó élményéhez elsősorban. A fő cél frontend oldalról, hogy könnyen lehessen kezelni, esztétikus legyen és gond nélkül át tudja látni a felhasználó a megjelenített adatokat. Ezen séma mentén készült el a MenuGenius weboldal is.

### Dizájn tervek

Első lépés a weboldal elkészítése során a tervezés volt.

A használt grafikai szoftverek:

* Paint 3D
* GIMP
* Wireframe

Vázlatokat és rétegeket készítettünk elsősorban, hogy hogyan képzeljük el a weboldal különböző komponenseit. (1. számú melléklet) Ezeket a folyamatokat követően kezdtük meg a programozást, a dizájn tervek mellett az adatbázis is már a kezdeteken elején kidolgozásra került nagyrészt.

A weboldalon látható logó kidolgozására a NameCheap[[10]](#footnote-10) nevű oldalt választottuk, mivel ez ingyenes és az általunk megadott paraméterekkel számos variációt tudott ajánlani, amelyek közül a jelenlegire esett a választásunk. Ez a logó az adott étteremnek megfelelően változik.



. ábra: Jelenlegi logó a főoldalon

## Használt technológiák

### Angular

Az Angular egy webalkalmazás-keretrendszer, amelyet a Google fejleszt. TypeScript programozási nyelv alapú, ingyenes és nyílt forráskód segítségével érhető el a rendszer, ami meglehetősen nagy népszerűségnek örvend. Az Angular alkalmazások felépítésében az MVC[[11]](#footnote-11), vagyis Model View Controller játssza a fő szerepet. A HTML fájl tehát maga a nézet, a modellek és a kontrollerek pedig JavaScript használatával készülnek el.

Ez a frontend framework a HTML eszköztárát kiterjeszti és kiegészíti, az alkalmazás felépítésében helyet kapó elemek pedig követhetően elkülönülnek.

Ezek mellett az Angular keretrendszer alapjához különféle modulok csatolhatók, amelyek az eszköztár bővítését szolgálják, de természetesen új, saját kiegészítéseket is lehet írni és hozzákapcsolni a rendszerhez.

Az Angular előnyei összesítve:

* Típushibákra figyelmeztet
* Meghatározott keretek között zajlik a kódolás, mivel kötött felépítésű
* Jól szeparált fájlrendszerrel rendelkezik, egy adott komponens részekre van bontva, remekül strukturált a jobb átláthatóság és érthetőség szempontjából (üzleti logika, CSS, template)
* Komponensek és modulok a fő alkotóelemei, így könnyen módosítható
* A központi hibakezelés is biztosított az *Interceptor* által, az összes szerverről érkező hibaüzenet általános kezeléssel bír. (422 validációs, 403 autentikációs)

Nagy mértékben az is befolyásolta a választást frontend oldalról, hogy a tanórán ezt tanultuk, valamint a jó dokumentáltsága sem volt utolsó tényező.

### Angular és Bootstrap 5.3

Az előre definiált stílusokhoz a Bootstrap 5.3-as verzióját használtuk, mivel ez egy könnyen átlátható és kezelhető felületet tud biztosítani a felhasználóknak és a dokumentáltsága is megfelelő volt számunkra.

### Reszponzivitás

A kész projekt kiszolgálja az alapvető elvárást, a reszponzivitást, ennek megfelelően minden eszközön használható, bármilyen megjelenítő eszközt támogat. A Bootstgrap Grid System segítségével tudtuk ezt elérni a kész weboldal esetében.

### Főoldal

A főoldal célja, hogy meghozza a felhasználó kedvét és étvágyát, ezért a rövid leírás mellett egyből látható egy Bootstrap 5.3 által biztosított Carousel, amelyben a termékek fotói láthatóak véletlenszerűen megjelenítve, a felhasználó egy ilyen megjelenítésnek köszönhetően tízszer nagyobb eséllyel kattint a látott termékre[[12]](#footnote-12).

## Backend

### Használt technológiák

A backendet a nyílt forráskódú, PHP alapú Laravel keretrendszerrel valósítottuk meg. Az adatbázis felépítéséhez a MySQL-MariaDB-t használtunk. Különböző engedélyezések kezelésére pedig a Sanctum megoldásra esett a választásunk.

### Laravel

A Laravel egy ingyenes és nyílt forráskódú PHP webes keretrendszer, amelyet Taylor Otwell fejlesztett ki. Az első verziója 2011-ben jelent meg, és azóta számos verziófrissítésen és fejlesztésen esett át[[13]](#footnote-13). A projekt fejlesztése során a legfrissebb, Laravel 10-es verziójával dolgoztunk.

Az egyik legnépszerűbb, legbiztonságosabb és legkönnyebb architektúra a Laravel. Rengeteg alapvető és ismétlődő funkciót old meg a fejlesztő helyett. Akár kezdőként is használható és gyorsan megtanulható, mivel kellő támpontokat tud adni, ahhoz, hogy jól strukturált kódot kapjunk a beépített terminál (Artisan[[14]](#footnote-14)) segítségével.

A Laravel előnyei és főbb jellemzői:

* Többnelyvű alkalmazások készítésére is alkalmas, így azok könnyen lokalizálhatóak
* Remekül testreszabható és bővíthető, így az egyedi igényekhez is alkalmazkodni tud
* Optimalizált kódbázissal bír, így nagy forgalmú honlapokhoz is tökéletes
* Jól dokumentált keretrendszer, számos segítséget találni az interneten hozzá
* Rengeteg beépített biztonsági funkcióval rendelkezik, amelyek segítenek megvédeni az alkalmazásokat a támadásoktól.
* Nagyon könnyen és gyorsan lehet vele dolgozni.
* Könnyen bővíthető és fejleszthető

### Laravel Sanctum

Laravel Sanctum egy hitelesítési csomag, amelyet a Laravel keretrendszerhez fejlesztettek ki. A fő célja az API-k (alkalmazásprogramozási felületek) autentikációjának és hitelesítésének megkönnyítése a Laravel alkalmazásokban. A Sanctum lehetővé teszi a token-alapú hitelesítést az API-k számára, így azok biztonságosan kommunikálhatnak a kliensalkalmazásokkal.

A működési elve nagyon egyszerű: amikor egy felhasználó bejelentkezik az alkalmazásba, a Laravel Sanctum egy egyszerű, állapotmentes JSON Web Token (JWT) -t generál számára. Ez a token azonosítja és hitelesíti a felhasználót az API kérések során. A kliensalkalmazásoknak ezt a tokent kell használniuk az API-khoz való hozzáféréshez. Amikor a kliensalkalmazás egy kérést küld az API-hoz, a tokent az Authorization fejlécben kell elküldenie. A Laravel Sanctum ellenőrzi a tokent, és ha érvényes, engedélyezi a kérést.

Ezenkívül a Sanctum lehetőséget biztosít a tokenek és az azonosítók "tartós" tárolására is, így a felhasználók bejelentkezési állapotát hosszabb időn keresztül megőrizhetjük.

A Laravel Sanctum egy könnyen használható eszköz, hogy biztonságosan kezeljük az API-hozzáférést és a felhasználók hitelesítését.

### Laravel Eloquent

A Laravel Eloquent egy olyan adatbázis-objektum-relációs leképzési (ORM) eszköz, amely a Laravel keretrendszer része. Az ORM rendszer lehetővé teszi az adatbázis táblák és az alkalmazás modelljeinek egyszerű és intuitív összekapcsolását. Az Eloquent használata segítségével a fejlesztők objektumok formájában dolgozhatnak az adatokkal, ami könnyebb és gyorsabb fejlesztést tesz lehetővé.

Az Eloquent egy MVC (Model-View-Controller) keretrendszerben használható. A modellek reprezentálják az alkalmazásban használt adatokat, és az Eloquent segítségével azokat könnyen és elegánsan kezelhetjük. A modellek és az adatbázis táblák közötti kapcsolatot a Laravel konvenciók alapján automatikusan felismeri az Eloquent, így nem szükséges konfigurációs fájlokat írni a kapcsolatok meghatározásához és nincs szükség SQL utasítások írására sem.

Az Eloquent támogatja az adatbázis lekérdezések egyszerű kezelését és a CRUD (Create, Read, Update, Delete) műveletek könnyű végrehajtását. Például a modellek segítségével egyszerűen létrehozhatunk új rekordokat, lekérdezhetjük az adatokat a táblákból, frissíthetjük a rekordokat vagy törölhetjük azokat az adatbázisból.

Az Eloquent további előnye, hogy lehetővé teszi az adatok közötti kapcsolatok definiálását is, például egy-egy, egy-több vagy sok-több kapcsolatokat. Ezáltal könnyen és hatékonyan modellezhetjük az alkalmazásban lévő komplex adatstruktúrákat.

Az Eloquent ORM egy nagyon erőteljes és intuitív eszköz, amely lehetővé teszi az adatok könnyű és hatékony kezelését az alkalmazások fejlesztése során.

### MySQL

A MySQL egy nyílt forráskódú, többszálú, SQL (Structured Query Language) alapú relációs adatbázis-kezelő szerver. A szoftvert eredetileg a svéd MySQL AB cég fejlesztette ki. A MySQL az egyik legelterjedtebb adatbázis-kezelő, költséghatékony és egyszerűen beállítható megoldást ad dinamikus webhelyek szolgáltatására. A relációs adatbázis az egy csoportba tartozó adatokat táblákba rendezi, az adatokat mezőkben tárolja. Ahhoz, hogy az egyes táblák adatai között létrejöjjön a kapcsolat idegen kulcsokat (foreign key) kell használnunk. Tulajdonképpen az idegen kulcsok egy másik tábla elsődleges kulcsai (primary key).[[15]](#footnote-15)

A MySQL adatbázis előnyei:

* Magas biztonsági jogosultságokkal rendelkezik.
* Működéséhez nem szükséges nagy mennyiségű erőforrás
* Nagy stabilitás, mivel rétegeket és modulokat foglal magába
* Egyszerű adatimportálás és exportálás
* Nagy mennyiségű adatkezeléssel bír

## Adatbázismodell

A dizájnt megelőzően, már a projekt első körében az adatbázis megtervezésével kezdtük. Ez egy iterációs folyamat, ahol a legelső lépés az üzleti igények felmérése és megértése, ezek után több modellt hoztunk létre. A projekt során ezek a modellek folyamatos átalakításon, finomításon estek át és egyes esetekben elvetésre kerültek. Miután újra átgondoltuk az üzleti igényeket, megkapta a végleges formáját az üzleti logika.

Az üzleti logika megértése után, a logikai modell elkészítésére törekedtünk. Az adat mezők típusainak meghatározására itt került sor.

A mezők típusai:

|  |  |
| --- | --- |
| * BIGINT * DECIMAL * VARCHAR * TIMESTAMP * INT * TEXT | * DATETIME * DATE * LONGTEXT * TINYINT * ENUM * LONGLOB |

A foglalások állapotaihoz létrehoztunk egy ENUM-ot, amelynek előre meghatározott állapotai lehetnek a projekten belül: „*ordered”, „cooked”, „served”*

Az entitások kapcsolatát is meg kellett határoznunk, melyeket az alábbi táblázatban ábrázolunk is a fő kapcsolati formák alapján:

|  |  |  |
| --- | --- | --- |
| egy-az-egyhez (1-1) | egy-a-többhöz (1-N) | több-a-többhöz (N-M) |
|  |  |  |
| Egy egyettípus egy egyedéhez egy másik egyedtípus csak egyetlen egyede kapcsolódhat és fordítva is igaz.  (osztály – osztályfőnök) | Egy egyedtípus egy egyedéhez egy másik egyedtípus több egyede is kapcsolódhat, de fordítva NEM igaz.  (osztály – tanuló) | Egy egyedtípus egy egyedéhez egy másik egyedtípus több egyede is kapcsolódhat, de fordítva is igaz.  (osztály – tanár) |

Ekkor létrejöttek a kapcsolati táblák, melyek definiálják a kapcsolatokat az entitások között. Itt már egy teljes struktúra bontakozik ki, amely láttatja, leképezi az üzleti logikát. A logikai modellben létrehozott mezőket valós adattárolásra alkalmas mezőkké definiáljuk, például: purchases adatmezőt lebontottuk valós vásárlási adatok tárolására alkalmas mezőkre: (dátum, teljes összeg, státusz, kinek a rendelése, melyik asztal stb).

Utolsó lépésként a fizikai modell létrehozása volt a feladatunk, amely adatbázis specifikus elemekkel bír és már az adatbázisban kerül megalkotásra. Ez már teljesen adatbázis függő. Ennek a szakasznak az eredménye a megvalósított adattáblák a kapcsolatokkal együtt.

Az adatok tárolására a MySQL adatbázist használtuk, ahol az adatok tárolása táblákban, mezőkben történt. Ezeket a táblákat a Laravel Eloquent nevű adatbázis leképezővel hoztuk létre a backend oldalon már megalkotott modellek alapján. A migrációt minden modell változásakor le tudtuk futtatni, így tudtuk az adatbázist folyamatosan frissíteni, így lett mindig napra kész az aktuális verzióhoz.

Az alábbi alfejezetekben bemutatásra kerülnek az adatbázis táblái a jellemzőkkel együtt: (teljes adatbázismodellt a 2. számú melléklet tartalmazza)

### Allergens tábla

Az allergéneket tartalmazó tábla.

Jellemzők:

* id
  + Típus: BIGINT (20)
  + Automatikusan létrehozott azonosító (Primary Key)
* code
  + Típus: DECIMAL (8, 2)
  + Egyedi kód az allergénekhez.
* name
  + Típus: VARCHAR (255)
  + Az allergén neve.
* deleted\_at
  + Típus: TIMESTAMP

### Categories tábla

A kategóriákat tartalmazó tábla

Jellemzők:

* id
  + Típus: BIGINT (20)
  + Automatikusan létrehozott azonosító (Primary Key)
* name
  + Típus: VARCHAR (255)
  + A kategória neve.
* deleted\_at
  + Típus: TIMESTAMP

### Desks tábla

Az asztalokat tartalmazó tábla

Jellemzők:

* id
  + Típus: BIGINT (20)
  + Automatikusan létrehozott azonosító (Primary Key)
* number\_of\_seats
  + Típus: INT (11)
  + Az asztal ülőhelyének száma.
* deleted\_at
  + Típus: TIMESTAMP

### Event\_logs tábla

A felhasználó által kiváltott különböző eseményeket tárolja

Jellemzők:

* id
  + Típus: BIGINT (20)
  + Automatikusan létrehozott azonosító (Primary Key)
* event\_type
  + Típus: VARCHAR (255)
  + Az interakció típusát jelöli
* user\_id
  + Típus: BIGINT (20)
* route
  + Típus: VARCHAR (255)
  + Az adott elérési útvonal
* body
  + Típus: TEXT
  + A küldött adat
* date\_time
  + Típus: DATETIME
  + Az esemény időpontja
* deleted\_at
  + Típus: TIMESTAMP

### Images tábla

A termékek fotóit tárolja.

Jellemzők:

* id
  + Típus: BIGINT (20)
  + Automatikusan létrehozott azonosító (Primary Key)
* img\_name
  + Típus: VARCHAR (255)
  + Fotó neve
* deleted\_at
  + Típus: TIMESTAMP
* img\_data
  + Típus: LONGLOB
  + A fotó adatait tartalmazza

### Ingredients tábla

A hozzávalókat tartalmazza.

Jellemzők:

* id
  + Típus: BIGINT (20)
  + Automatikusan létrehozott azonosító (Primary Key)
* name
  + Típus: VARCHAR (255)
  + A hozzávaló neve
* deleted\_at
  + Típus: TIMESTAMP

### Ingredient\_allergen kapcsolati tábla

Kapcsolótábla, mely az adott hozzávalókhoz tartozó allergéneket tartalmazza (N:M kapcsolat)

Jellemzők:

* ingredient\_id
  + Típus: BIGINT (20)
* allergen\_id
  + Típus: BIGINT (20)
* deleted\_at
  + Típus: TIMESTAMP
  + Típus: INT (11)

### Password\_reset\_tokens tábla

A Laravel által automatikusan generált tábla, és a Laravel beépített jelszó-visszaállítási mechanizmusához kapcsolódik. Amikor egy felhasználó kéri a jelszó visszaállítását, például elfelejtette a jelszavát, akkor a Laravel ezt a táblát használja az új jelszó generálásához és az azonosításhoz.

Jellemzők:

* email
  + Típus: VARCHAR (255)
* token
  + Típus: VARCHAR (255)
* created\_at
  + Típus: TIMESTAMP

### Personal\_access\_tokens tábla

A personal\_access\_tokens tábla a Laravel Sanctum csomag része, és szintén automatikusan generálódik. Ez a tábla arra szolgál, hogy tárolja a felhasználókhoz kapcsolódó személyes hozzáférési tokeneket.

A személyes hozzáférési tokenek lehetővé teszik a felhasználók számára, hogy hozzáférjenek bizonyos védett erőforrásokhoz vagy funkciókhoz az alkalmazáson belül. Ezek a tokenek általában egy időkorlátot is tartalmaznak, így csak meghatározott ideig érvényesek, ami tovább növeli az alkalmazás biztonságát.

A personal\_access\_tokens tábla felelős az ilyen tokenek kezeléséért és tárolásáért. Amikor egy felhasználó létrehoz egy személyes hozzáférési tokent, a Laravel Sanctum egy egyedi token-t generál és tárolja ezt a táblában a felhasználóhoz kapcsolva. Ezeket a tokeneket általában a felhasználó autentikációja után kapják meg, és aztán használhatják az API-hozzáférésekhez vagy más funkciókhoz az alkalmazáson belül.

Jellemzők:

* tokenable\_id
  + Típus: BIGINT (20)
* name
  + Típus: VARCHAR (255)
* token
  + Típus: VARCHAR (64)
* abilities
  + Típus: TEXT
* last\_used\_at
  + Típus: TIMESTAMP
* expires\_at
  + Típus: TIMESTAMP
* created\_at
  + Típus: TIMESTAMP
* updated\_at
  + Típus: TIMESTAMP

### Products tábla

A termékeket tartalmazza.

Jellemzők:

* name
  + Típus: VARCHAR (255)
  + Termék neve
* description
  + Típus: VARCHAR (255)
  + A termékről való rövid leírás
* category\_id
  + Típus: BIGINT(20)
  + Category táblára mutató idegen kulcs (Foreign key)
* packing
  + Típus: VARCHAR (255)
  + Az adott termék adagolása
* price
  + Típus: INT (11)
  + A termék ára
* is\_food
  + Típus: TINYINT (1)
  + Étel vagy sem az adott termék
* image\_id
  + Típus: BIGINT (20)
  + Images táblára mutató idegen kulcs (Foreign key)
* deleted\_at
  + Típus: TIMESTAMP

### Product\_ingredient kapcsolati tábla

Kapcsolótábla, mely az egyes termékekhez tartozó hozzávalókat tartalmazza (N:M kapcsolat).

Jellemzők:

* product\_id
  + Típus: BIGINT (20)
  + Products táblára mutató idegen kulcs (Foreign key)
* ingredient\_id
  + Típus: BIGINT (20)
  + Ingredients táblára mutató idegen kulcs (Foreign key)
* deleted\_at
  + Típus: TIMESTAMP

### Product\_purchase kapcsolati tábla

Kapcsolótábla, a megvárásolt termékeket kapcsolja össze a hozzájuk tartozó vásárlással. N:M kapcsolat, tartalmaz egy plusz mezőt is (quantity), melyben a megvásárolt termék darabszáma kerül feltüntetésre, amennyiben egy vásárláskor valamely termékből többet vásároltak.

Jellemzők:

* id
  + Típus: BIGINT (20)
  + Automatikusan létrehozott azonosító (Primary Key)
* product\_id
  + Típus: BIGINT (20)
  + Products táblára mutató idegen kulcs (Foreign key)
* purchase\_id
  + Típus: BIGINT (20)
  + Purchases táblára mutató idegen kulcs (Foreign key)
* quantity
  + Típus: INT (11)
  + Adott termék mennyiségét jelöli
* deleted\_at
  + Típus: TIMESTAMP

### Product\_user kapcsolati tábla

Ez a tábla kezeli a kapcsolatokat, amikor egy regisztrált felhasználó a saját kedvencei közé tesz egy terméket. A tábla N:M kapcsolatú, külön mezőben van lehetőség a termékek 1-5 csillagig való értékelésére (amennyiben ez a funkció megvalósításra kerül a későbbiekben).

Jellemzők:

* id
  + Típus: BIGINT (20)
  + Automatikusan létrehozott azonosító (Primary Key)
* product\_id
  + Típus: BIGINT (20)
  + Products táblára mutató idegen kulcs (Foreign key)
* user\_id
  + Típus: BIGINT (20)
  + User táblára mutató idegen kulcs (Foreign key)
* favorite
  + Típus: TINYINT (1)
  + Kedvenc termék vagy sem jelölésére szolgál
* stars (ez a projekten belül nincs használatban, de a jövőben való fejlődés során igénybe vehető)
  + Típus: INT (11)
  + A felhasználó által értékelve 1-5 skálán
* deleted\_at
  + Típus: TIMESTAMP

### Purchases tábla

Az oldalon végbemenő vásárlások tárolására alkalmazzuk a projekten belül.

Jellemzők:

* id
  + Típus: BIGINT (20)
  + Automatikusan létrehozott azonosító (Primary Key)
* date\_time
  + Típus: DATETIME
  + A vásárlás időpontját jelöli
* total\_pay
  + Típus: INT (11)
  + Az az érték, amelyben a vásárlás történt
* status
  + Típus: ENUM (’ordered’, ’cooked’, ’served’)
  + Az adott étel státuszát jelöli
* paid
  + Típus: TINYINT (1)
  + A vásárlás fizetett vagy sem
* user\_id
  + Típus: BIGINT (20)
  + User táblára mutató idegen kulcs (Foreign key)
* desk\_id
  + Típus: BIGINT (20)
  + Desks táblára mutató idegen kulcs (Foreign key)
* deleted\_at
  + Típus: TIMESTAMP

### Reservations tábla

A foglalásokat tartalmazza ez a tábla.

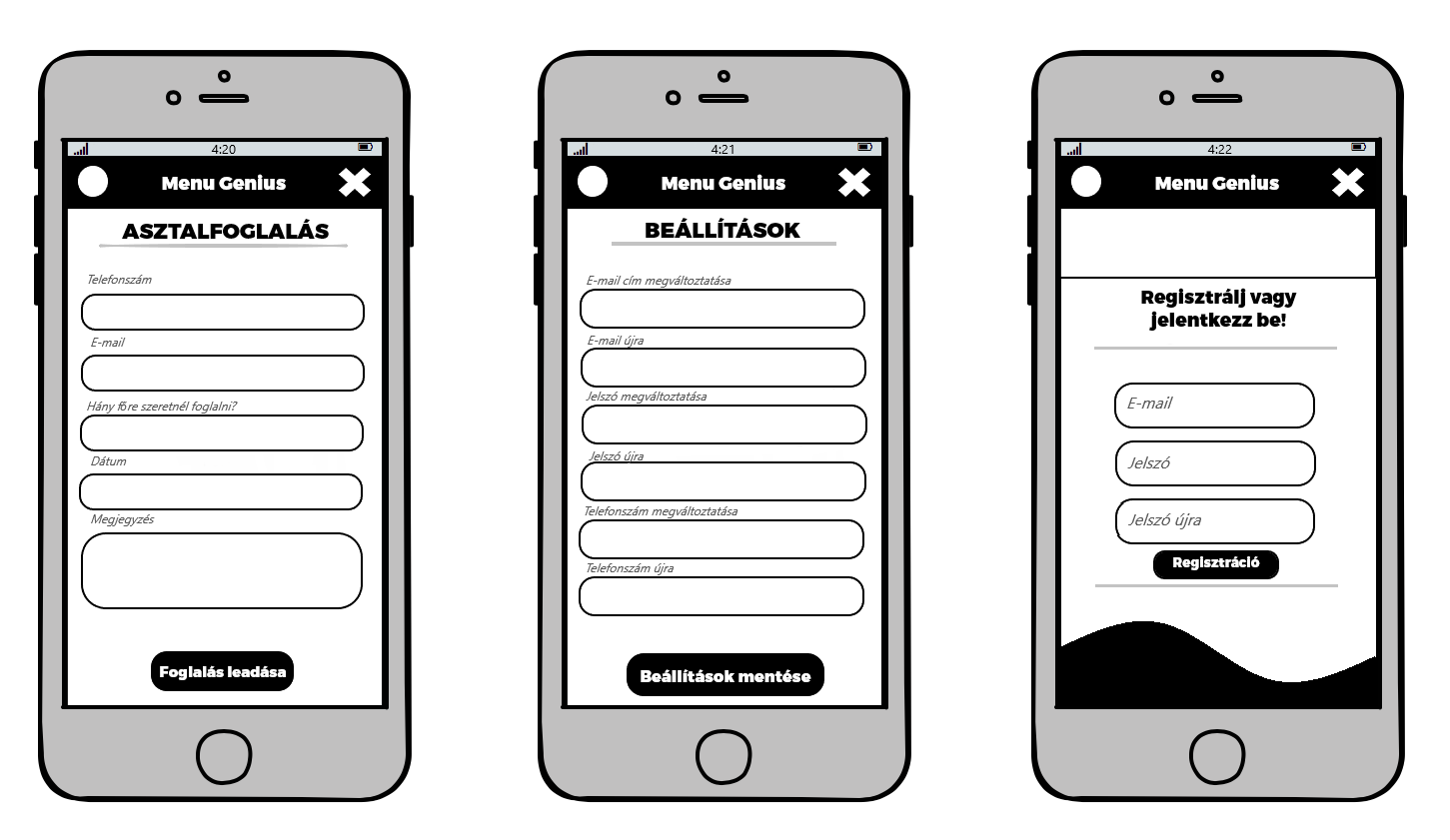
Jellemzők:

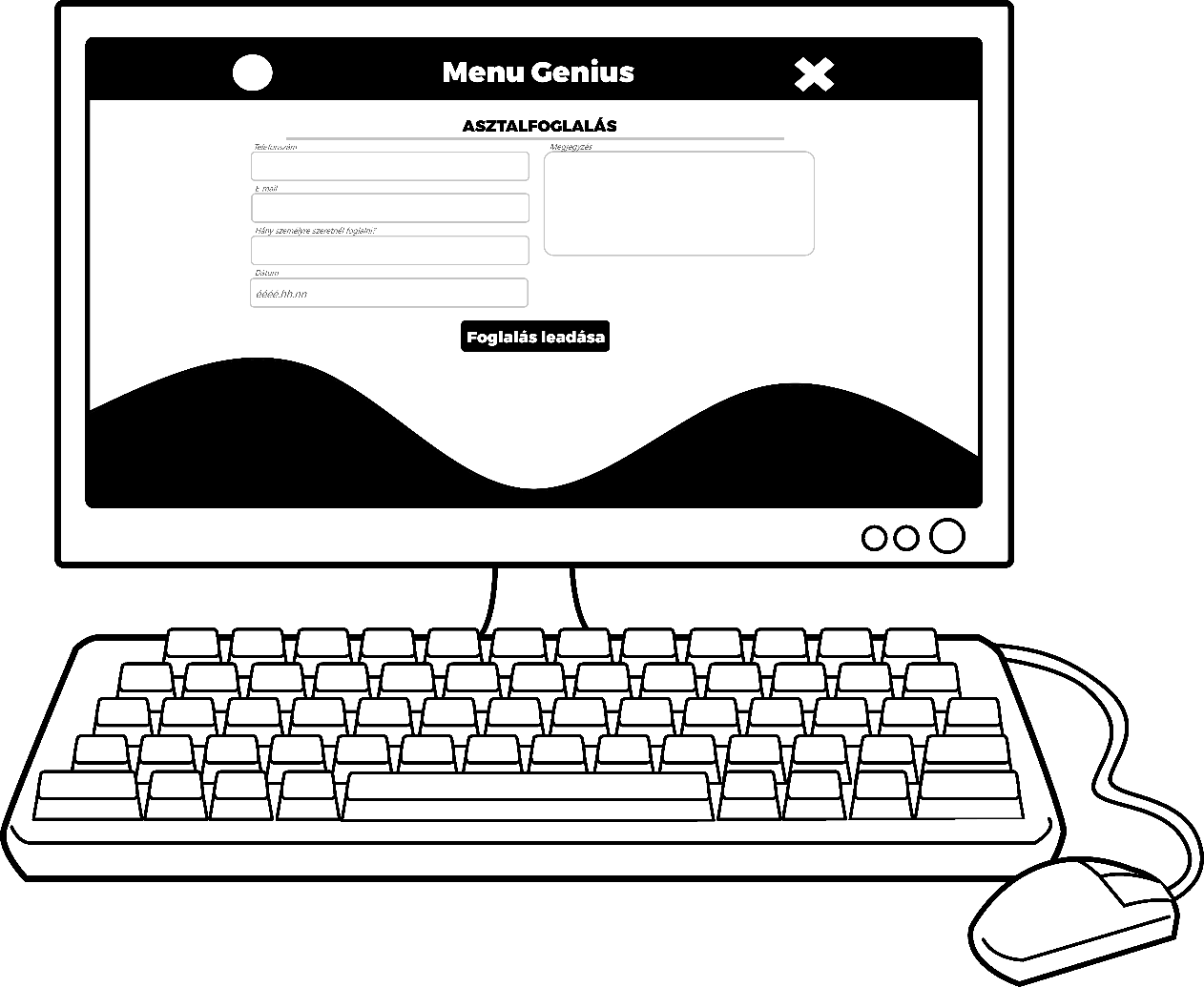
* checkout\_date
  + Típus: DATETIME
* name
  + Típus: VARCHAR (255)
  + A foglalásnál megadott nevet tárolja
* phone
  + Típus: VARCHAR (255)
  + A foglalásnál megadott telefonszámot tárolja
* desk\_id
  + Típus: BIGINT (20)
  + Desks táblára mutató idegen kulcs (Foreign key)
* user\_id
  + Típus: BIGINT (20)
  + Users táblára mutató idegen kulcs (Foreign key)
* closed
  + Típus: TINYINT (1)
  + A foglalás lezárult-e vagy sem
* comment
  + Típus: TEXT
  + A felhasználó által megadott megjegyzés a foglalás során
* deleted\_at
  + Típus: TIMESTAMP

### Users tábla

Jellemzők:

* id
  + Típus: BIGINT (20)
  + Automatikusan létrehozott azonosító (Primary Key)
* name
  + Típus: VARCHAR (255)
  + A felhasználó által megadott név
* email
  + Típus: VARCHAR (255)
  + A felhasználó által megadott email cím
* email\_verified\_at
  + Típus: TIMESTAMP
* password
  + Típus: VARCHAR (255)
  + A felhasználó által megadott jelszó
* phone
  + Típus: VARCHAR (255)
  + A felhasználó által megadott telefonszám
* admin
  + Típus: TINYINT (1)
  + Admin a felhasználó vagy sem
* remember\_token
  + Típus: VARCHAR (255)
  + Ezzel tárolható az adott felhasználó bejelentkezési adatai
* deleted\_at
  + Típus: TIMESTAMP

Mellékletek



1. számú melléklet
2. számú melléklet

1. <https://stackdiary.com/front-end-frameworks/> 2024. 01 29. [↑](#footnote-ref-1)
2. <https://techjury.net/blog/laravel-statistics/> 2024. 01 29. [↑](#footnote-ref-2)
3. <https://www.trofea.hu/>, <https://www.mcdonalds.com/hu/hu-hu.html> és <https://kfc.hu/> 2024. 01 29. [↑](#footnote-ref-3)
4. [PHP: Hypertext Preprocessor](https://www.php.net/) 2024.02.07 [↑](#footnote-ref-4)
5. [MySQL](https://www.mysql.com/) 2024.02.07 [↑](#footnote-ref-5)
6. [Database Tools for Development and Management | dbForge (devart.com)](https://www.devart.com/dbforge/) 2024.02.07 [↑](#footnote-ref-6)
7. [Git és GitHub gyorstalpaló. A fejlesztők körében leginkább… | by Laszlo Fazekas | ENVIENTA Magyarország | Medium](https://medium.com/envienta-magyarorsz%C3%A1g/git-%C3%A9s-github-gyorstalpal%C3%B3-f2d78a732deb) 2024.02.07. [↑](#footnote-ref-7)
8. [Verziókezelés – Wikipédia (wikipedia.org)](https://hu.wikipedia.org/wiki/Verzi%C3%B3kezel%C3%A9s) 2024.02.07. [↑](#footnote-ref-8)
9. [Front-end jelentése (lexiq.hu)](https://lexiq.hu/front-end) 2024.02.07. [↑](#footnote-ref-9)
10. <https://www.namecheap.com/logo-maker/> 2024.02.08. [↑](#footnote-ref-10)
11. [MVC Framework Introduction - GeeksforGeeks](https://www.geeksforgeeks.org/mvc-framework-introduction/) 2024.02.08. [↑](#footnote-ref-11)
12. <https://www.hrenko.hu/blog/a-carousel-hirdetesek-elonyei> 2024.02.08. [↑](#footnote-ref-12)
13. <https://webcapital.hu/hu/blog/laravel-bemutatasa-mukodese> 2024.02.09. [↑](#footnote-ref-13)
14. <https://laravel.com/docs/10.x/artisan> 2024.02.09. [↑](#footnote-ref-14)
15. <https://www.awh.hu/kb/webtarhely/mi-az-a-mysql> 2024.02.09. [↑](#footnote-ref-15)