A képen clipart, macska, illusztráció, rajzfilm látható

Automatikusan generált leírás

**Széchenyi István Katolikus Technikum és Gimnázium**

**Szoftverfejlesztő és -tesztelő projektfeladat**

**PÉKSÉG**

Készítették:

Vakherda Ádám,

Bolyky Máté,

Iróczki Regina

**Ózd, 2025**

Tartalomjegyzék

[RENDSZERKÖVETELMÉNY 5](#_Toc196814635)

[Szerveroldali követelmények 5](#_Toc196814636)

[Kliensoldali követelmények 5](#_Toc196814637)

[WEBALKALMAZÁS HASZNÁLATA/INDÍTÁSA 6](#_Toc196814638)

[Alkalmazott fejlesztői és csoportmunka eszközök 7](#_Toc196814639)

[Adatbázis 8](#_Toc196814640)

[Frontend 8](#_Toc196814641)

[Backend 8](#_Toc196814642)

[ADATBÁZIS 9](#_Toc196814643)

[Modell leírása 9](#_Toc196814644)

[TÁBLÁK, KAPCSOLATOK 13](#_Toc196814645)

[ER/EK DIAGRAMM 14](#_Toc196814646)

[FRONTEND DOKUMENTÁCIÓ 15](#_Toc196814647)

[Fő navigációs menü 16](#_Toc196814648)

[Regisztráció és bejelentkezés 17](#_Toc196814649)

[Profilkezelés 19](#_Toc196814650)

[Termékek 20](#_Toc196814651)

[Kosár és rendelési folyamat 21](#_Toc196814652)

[Korai Design Koncepciók 23](#_Toc196814653)

[FŐBB KÓDRÉSZLETEK 26](#_Toc196814654)

[TESZTELÉS 38](#_Toc196814655)

[BACKEND DOKUMENTÁCIÓ 39](#_Toc196814656)

[TESZTELÉS 45](#_Toc196814657)

[getByid: 45](#_Toc196814658)

[Keresés név és típus alapján: 46](#_Toc196814659)

[Update/Destroy: 47](#_Toc196814660)

[GetHigher/Lowerthan: 48](#_Toc196814661)

[FilterByFizetesimod: 49](#_Toc196814662)

[Seeder: 50](#_Toc196814663)

[Bejelentkezési (login): 51](#_Toc196814664)

[addFelhasznalo: 51](#_Toc196814665)

[TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK 52](#_Toc196814666)

[IRODALOMJEGYZÉK 53](#_Toc196814667)

**BEVEZETÉS**

Projektmunkánk során egy olyan pékség weboldalának megtervezésén dolgoztunk, amelynek célja, hogy napi szinten friss, kiváló minőségű pékáruval lássa el az oktatási intézményeket. Az ötlet alapját saját tapasztalataink adták, hiszen diákokként sokszor szembesültünk azzal, hogy az iskolai menzán vagy büfében a kínált kenyér, kifli vagy zsemle nem volt már igazán friss, mire elfogyasztásra került.

Úgy gondoljuk, hogy egy problémát akkor lehet igazán hatékonyan megoldani, ha az ember maga is megtapasztalta azt. Ezért választottuk ezt a témát: személyesen is érezzük, mennyire fontos lenne a friss pékáru biztosítása az iskolákban, óvodákban, egyetemeken. Célunk, hogy a weboldalunk segítségével az intézmények egyszerűen és gyorsan tudjanak megrendeléseket leadni, és biztosak lehessenek abban, hogy az általuk kapott termékek mindig frissek és ízletesek lesznek.

A weboldal lehetővé teszi a rendelési folyamat digitalizálását, átláthatóságát és automatizálását. A kínálat könnyen böngészhető, a rendelés személyre szabható, a szállítás pedig megbízható és rendszeres.  
Jövőbeli terveink között szerepel, hogy a pékséget tovább fejlesztjük: bővítjük a kínálatot (pl. gluténmentes vagy vegán opciókkal), és más intézményeket – például kórházakat vagy irodaházakat – is bevonunk a rendszerbe. Hosszú távon egy olyan hálózatot szeretnénk létrehozni, amely nemcsak friss pékárut biztosít, hanem hozzájárul egy egészségesebb, élhetőbb mindennapi környezet kialakításához is.

Github elérhetőségének a linkje: https://github.com/bolykymate/Pekseg

# RENDSZERKÖVETELMÉNY

## Szerveroldali követelmények

* **Operációs rendszer:** Linux vagy Windows
* **Backend:** Node.js/PHP/Python alapú szerver
* **Adatbázis:** MySQL
* **Tárhely:** Minimum 50GB SSD tárhely
* **RAM:** Minimum 4GB RAM
* **Hálózati kapcsolat:** Minimum 100 Mbps internetkapcsolat

## Kliensoldali követelmények

* **Böngésző támogatás:** Chrome, Firefox, Edge, Safari (legújabb verziók)

**Minimális hardverigény**

* **Asztali gép/laptop:** Legalább 4GB RAM, 2 magos CPU, internetkapcsolat
* **Mobil eszköz:** Android 8.0+ vagy iOS 12.0+ támogatása

**Szoftver**

* **VS Code** v1.96.0 vagy újabb
* **Node.js** v22.13.0 vagy újabb
* **XAMPP** 8.1.12 vagy újabb
* **Composer** 2.8.4 vagy újabb

# WEBALKALMAZÁS HASZNÁLATA/INDÍTÁSA

**Előfeltételek a webalkalmazás futtatáshoz/telepítéshez:**

* XAMPP (Apache + MySQL) telepítése
* PHP (XAMPP részeként települ)
* Visual Studio Code telepítése
* Composer telepítése (Laravel függőségekhez)
* Angular CLI telepítése
* Node.js és npm telepítése (Angular futtatásához)

**XAMPP konfigurálása**:

1. Indítsa el a XAMPP Control Panelt, az Apache és MySQL szolgáltatásokat
2. Nyissa meg a http://localhost/phpmyadmin felületet

**Laravel backend indítása:**

1. Projektmappa megnyitása Visual studio Code-ban
2. Ellenőrizzük, hogy az env-ben a pekseg adatbázis van megadva
3. Új terminál nyitása
4. Adatbázis migrálása és feltöltése: „php artisan migrate –seed” paranccsal
5. Importáljuk be a projektmappában lévő sql fájlt a pékáru táblába
6. Laravel szerver indítása: „php artisan serve” paranccsal
7. A terminálban ekkor megjelenik egy visszajelzés, amely mutatja, hogy a szerver fut

Ezután a backend elérhető lesz

**Angular frontend indítása:**

1. Projektmappa megnyitása Visual studio Code-ban
2. Új terminál nyitása
3. Függőségek telepítése: npm install
4. Fejlesztői szerver indítása: ng serve
5. A terminál visszajelzést ad a sikeres indításról

Ezután a frontend a terminálban megadott címen (alapértelmezetten http://localhost:4200) lesz elérhető

# Alkalmazott fejlesztői és csoportmunka eszközök

A munkánk során többféle eszközt is használtunk, amelyek segítettek a tervezésben, a fejlesztésben és abban, hogy csapatként hatékonyan tudjunk együtt dolgozni.

* **Figma**  
  A weboldal megtervezésénél a Figma nevű online tervezőprogramot használtuk. Ebben tudtuk előre megrajzolni/megtervezni, hogyan nézzen ki az oldal. Így mindenki látta, mi, hogy fog kinézni, és könnyebb volt megbeszélni az ötleteket.
* **Trello**  
  A feladatok nyomon követéséhez a Trello-t vettük igénybe. Ebben létre tudtunk hozni listákat és kártyákat, amikre felírtuk, kinek mi a dolga, és mi hol tart. Ez különösen jól jött, mert segített rendszerezni a feladatokat/teendőket.
* **VSC**   
  A fejlesztéshez a Visual Studio Code-ot használtuk, mivel jól támogatja a különböző programozási nyelveket és keretrendszereket, mint például a HTML, CSS, JavaScript, Angular és PHP. Egyszerű kezelhetősége miatt ideális választás volt számunkra a munka során.
* **GitHub**  
  A GitHubot a forráskód tárolására és verziókezelésére használtuk. Segített abban, hogy egyszerre tudjunk dolgozni, anélkül, hogy egymás munkáját felülírtuk volna. Minden változtatás követhető és biztonságosan kezelhető volt, így átlátható maradt a fejlesztés folyamata.
* **Discord**  
  A csapat közötti kommunikációra a Discordot használtuk. A platform/felület lehetőséget biztosított a gyors üzenetváltásra, kérdések gyors megválaszolására, valamint hang- és videóhívások lebonyolítására.

## Adatbázis

Az adatbázist MySQL-ben készítettük el, amit a XAMPP-on belül a phpMyAdmin felületen keresztül kezeltünk. Itt hoztuk létre a táblákat a termékek, rendelések, iskolák és egyéb adatok számára.

## Frontend

A frontend fejlesztése Angular keretrendszerben történt. Ez lehetővé tette a komponensalapú, fejlesztést és a dinamikus felhasználói élményt.A komponenseket TypeScript-ben írtuk, a stílusok pedig CSS segítségével lettek kialakítva.

## Backend

A backend rész Laravel keretrendszerrel készült, PHP nyelven. A Laravel segítségével hoztuk létre azokat a REST API-kat, amiken keresztül az Angular frontend adatokat kér le vagy küld el. A backend feladata az adatbázis kezelése, az adatok ellenőrzése (validálása), valamint minden olyan szerveroldali működés, ami a háttérben zajlik, de fontos a rendszer működése szempontjából.

# ADATBÁZIS

Az adatbázisunk azzal a céllal készült, hogy segítsen egy online rendelési rendszer működésében, amely pékáruk árusítására lett kitalálva. Az adatbázis segítségével könnyedén nyomon követhetők a rendelések fontos részletei, például a rendelés dátuma, a kiszállítás tervezett időpontja, valamint a választott fizetési mód. Ezen kívül a Pékáru tábla információkat tárol/nyújt a termékekről, beleértve azok nevét, típusát és árait. Az adatbázis tartalmazza a felhasználók adatait is, mint például a felhasználó nevét, jelszavát és email címét, melyek segítenek a vásárlók kezelésében. A címek tábla pedig azokat az információkat tartalmazza, amelyek a kézbesítéshez szükségesek, beleértve a felhasználók címét, a telefonszámukat és a számlázási nevet is, így biztosítva, hogy a rendelések pontosan és időben eljussanak a megfelelő címre.

## Modell leírása

**Pékáru**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mezők** | **Kulcs fajtája** | **Adattípus** | **Rövid leírás** |
| **Pid** | **PRIMARY KEY** | **int** | **A pékáruink elsődleges azonosítója** |
| **nev** |  | **varchar(20)** | **A pékáruk neve** |
| **tipus** |  | **varchar(20)** | **A pékáruk fajtáját/típusát tárolja el** |
| **ar** |  | **int** | **A pékáruk ára** |

**Felhasználók**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mezők** | **Kulcs fajtája** | **Adattípus** | **Rövid leírás** |
| **Fid** | **PRIMARY KEY** | **int** | **A felhasználók elsődleges azonosítója** |
| **nev** |  | **varchar(20)** | **A felhasználók nevét tárolja** |
| **jelszo** |  | **varchar(30)** | **A felhasználók jelszavát tárolja** |
| **email** |  | **varchar(30)** | **A felhasználók email címeit tárolja** |

**Cím**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mezők** | **Kulcs fajtája** | **Adattípus** | **Rövid leírás** |
| **Cid** | **PRIMARY KEY** | **int** | **A címtábla elsődleges azonosítója** |
| **cim** |  | **varchar(30)** | **A felhasználók teljes címét tárolja el** |
| **Fid** | **FOREIGN KEY** |  | **A felhasználók tábla mezőit meghívó idegenkulcs** |
| **SzamlazasiNev** |  | **varchar(30)** | **A felhasználók számlázási nevét tünteti fel** |
| **telefonszam** |  | **varchar(15)** | **A felhasználók telefonszámait tárolja el** |

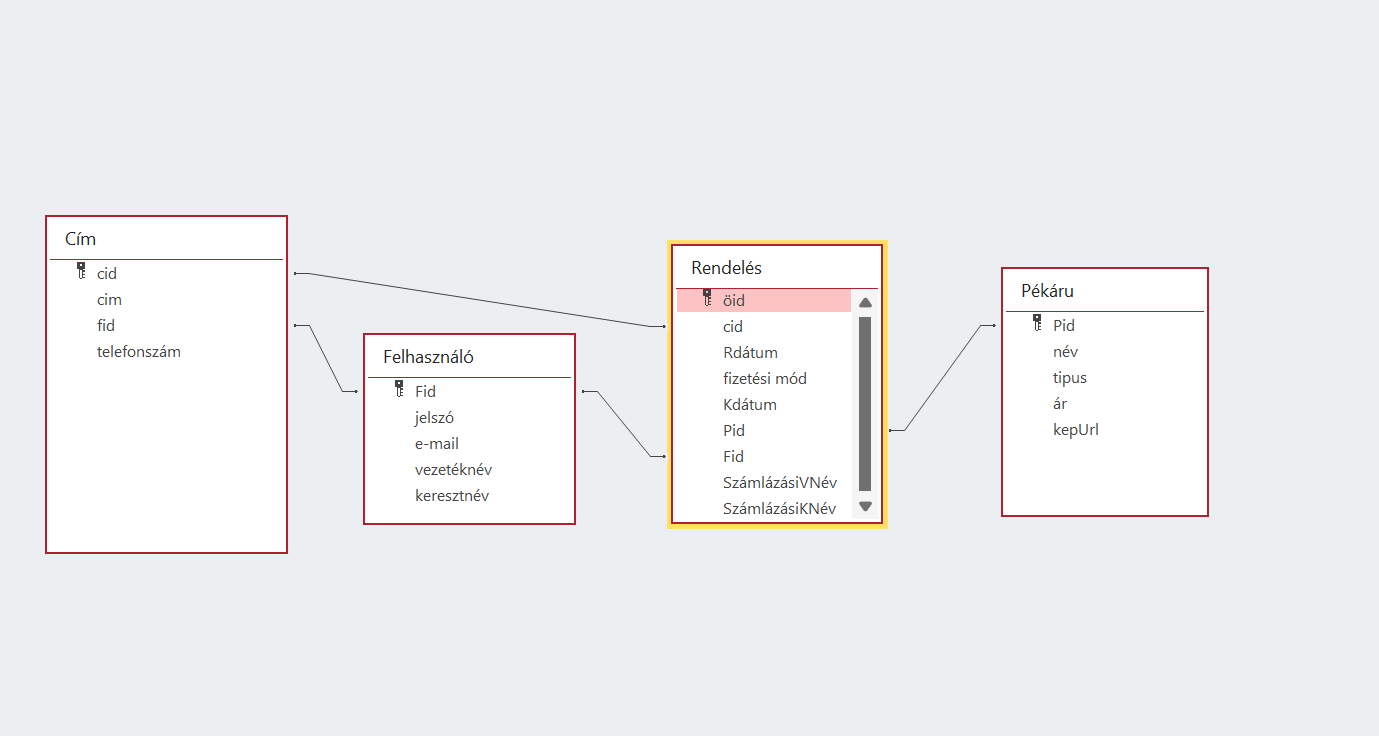
**Rendelés**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mezők** | **Kulcs fajtája** | **Adattípus** | **Rövid leírás** |
| **Rid** | **PRIMARY KEY** | **int** | **A rendelés azonosítója** |
| **Pid** | **FOREIGN KEY** | **int** | **A pékáru tábla mezőit meghívó idegenkulcs** |
| **Fid** | **FOREIGN KEY** | **int** | **A felhasználók tábla mezőit meghívó idegenkulcs** |
| **Cid** | **FOREIGN KEY** | **int** | **A címek tábla mezőit meghívó idegenkulcs** |
| **RDatum** |  | **varchar(20)** | **A rendelés dátumának ideje** |
| **KDatum** |  | **varchar(20)** | **A kiszállítás dátumának várható időpontja** |
| **FizetesiMod** |  | **boolean** | **A fizetési mód eldöntésére szolgál** |

## TÁBLÁK, KAPCSOLATOK

***SEQ ábra \\* ARABIC 1****. ábra*

Az alábbi ábra az adatbázis táblái közötti kapcsolatokat mutatja be. Itt látható, hogyan kapcsolódnak egymáshoz a Cím, Felhasználó, Pékáru és Rendelés táblák idegen kulcsokon keresztül.



1. kép: A táblák közötti kapcsolatok ábrázolása.

## ER/EK DIAGRAMM

Az alábbi ER-diagram szemlélteti, hogy a rendszer milyen adatokat tárol, ezek hogyan kapcsolódnak egymáshoz, és miként épül fel az adatbázis logikailag. Ez segít könnyebben átlátni a működését.

***SEQ ábra \\* ARABIC 2****. ábra*

A képen diagram, vázlat, rajz, Tervrajz látható

Automatikusan generált leírás

2. kép: Az entitások, azok jellemzői és a közöttük lévő kapcsolatok vizuális bemutatása.

# FRONTEND DOKUMENTÁCIÓ

A weboldal frontendjének fejlesztéséhez Angular keretrendszert használtunk, mert gyors, dinamikus és interaktív felhasználói élményt biztosít. Az Angular komponens alapú felépítése lehetővé teszi a kód újra felhasználását és könnyű karbantartását. Az Angular azért volt ideális választás a projekt számára, mivel egyszerre biztosít funkcionalitást és felhasználóbarát élményt. A webalkalmazás összesen 6 oldalból áll (Bejelentkezés/Regisztráció, Főoldal, Kosár, Rendelés véglegesítése, Termékek, és a Profil adatok), amikből elsők között a főoldalt tekinthetjük meg.

**(FŐOLDAL KINÉZETE)**

**A képen szöveg, étel, desszert, Pékáru látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.**

3. kép A főoldal felső része

**A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.**

**4**. kép A főoldal alsó része

**Navigációs rendszer**

## Fő navigációs menü

A weboldal tetején található fő navigációs sáv a következő lehetőségeket kínálja:

* Home/Logó: Visszavezet a kezdőlapra, ahol a legnépszerűbb termékek és a hirdetések láthatók
* Kapcsolat: A Bakers Room céggel kapcsolatos információk/elérhetőségek megtekintése
* Termékek: Az összes elérhető pékárú megtekintése kategóriák szerint
* Bejelentkezés/Regisztráció: Regisztrált felhasználók bejelentkezésére, vagy új felhasználók regisztrációjára szolgál
* Kosár: A felhasználó kosarában lévő termékek megtekintése és kezelése

A képen szöveg, étel, desszert, Pékáru látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A navigációs sáv minden oldalon konzisztensen megjelenik, lehetővé téve a gyors tájékozódást.

5. kép A minden oldalon feltüntetett navigációs sáv

## Regisztráció és bejelentkezés

**Regisztráció**  
A felhasználók számára elérhető egy regisztrációs űrlap, amely a következő kötelező mezőket tartalmazza:

* Vezetéknév
* Keresztnév
* E-mail cím
* Jelszó

Az űrlap alján elhelyezésre került egy **„Van már fiókja?”** link, amely a bejelentkezési oldalra navigálja a meglévő felhasználókat.

A képen szöveg, képernyőkép, étel látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.**Bejelentkezés**  
A bejelentkezéshez a felhasználóknak az alábbi adatokat kell megadniuk:

6. kép Regisztrációs felület

* E-mail cím
* Jelszó

A képen szöveg, Snack, étel, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A képen szöveg, Snack, étel, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Amennyiben a felhasználó még nem rendelkezik fiókkal, a bejelentkezési felületen található **„Nincs még regisztrálva?”** link segítségével átirányítható a regisztrációs oldalra.

8.kép Nézet, ha a felhasználó nincs bejelentkezve

7. kép Bejelentkezési felület

**Bejelentkezetlen állapotban**

**A képen szöveg, képernyőkép, szoftver, Weblap látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.**"Bejelentkezés" gomb vezet a bejelentkező felületre

"Regisztráció" lehetőség a bejelentkezési oldalon

**Bejelentkezett állapotban**

**9**. kép Nézet, ha be van jelentkezve

A "Bejelentkezés" helyén a felhasználó neve jelenik meg dropdown menüvel

5. kép Bejelentkezési felület

## Profilkezelés

A felhasználói profil oldalon a regisztrált felhasználók megtekinthetik és kezelhetik személyes adataikat.  Az oldalon az alábbi főbb funkciók érhetők el:

A rendszer lehetővé teszi a felhasználók számára, hogy megtekintsék és szükség esetén módosítsák az alapvető adataikat, beleértve például a vezeték- és keresztnevet, email címet, telefonszámot és lakcím információkat. A kötelezően kitöltendő mezőket csillag (\*) jelzi.

A felhasználók bármikor frissíthetik adataikat. A módosítások véglegesítéséhez az "Adatok mentése" gombra kell kattintani, ezzel az adatok el lesznek mentve az adatbázisba.

A képen szöveg, képernyőkép, szoftver, Weblap látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

10. kép A profil adatok kezelésére szolgáló felület

## Termékek

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A termékoldalon egyszerűen szűrhetjük a pékárukat típus szerint. Az oldal tetején lévő gombok segítségével gyorsan lehet váltani a különböző termékcsoportok között. Az "Összes" gomb minden terméket megmutat, a "Fehér" gomb csak a fehér lisztes péksüteményeket, például a fehér kenyeret. A "Finom" gombra kattintva a vajas, lágy kiflik és hasonló termékek jelennek meg. A "Fánk" gomb a süteményeket, a "Barna" gomb pedig a teljes kiőrlésű és barna lisztes pékárukat mutatja. A kategóriaválasztó felület kifejezetten felhasználóbarát kialakítású. A kiválasztott kategória mindig jól látható - külön színnel vagy kiemeléssel jelölve. Amikor a felhasználó másik kategóriára kattint, a terméklista azonnal frissül, így nincs várakozási idő.

11. kép A pékség termékei kilistázva egy külön felületen

Fontos megjegyezni, hogy a kosárba helyezés funkció csak bejelentkezett felhasználók számára érhető el. Ez biztosítja, hogy csak hitelesített felhasználók tudjanak rendelést leadni.

## Kosár és rendelési folyamat

**Kosár**

A kosár oldalon a felhasználók áttekinthetik a kiválasztott termékeket és módosíthatják a rendelés részleteit. A kosár felülete a következő funkciókat tartalmazza:

* A kosárban szereplő termékek listázása, darabszám és egységár feltüntetésével.
* A darabszám módosításának lehetősége minden termék esetében.
* Egyedi Törlés gomb minden termék mellett, a termék eltávolításához.
* Az összesített végösszeg automatikus megjelenítése és frissítése a kosár tartalmának változásakor.
* **A képen szöveg, képernyőkép, szoftver, Weblap látható

  Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.**„Rendelés folytatása” gomb, amely elindítja a rendelési folyamat következő lépését (fizetési és szállítási adatok megadása).

**12**. kép Kosár felülete

**Rendelés véglegesítése**

A képen szöveg, képernyőkép, szoftver, Operációs rendszer látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A rendelés véglegesítésekor a felhasználó egy összefoglaló oldalon ellenőrizheti, hogy milyen termékeket tett a kosárba, hány darabot választott, és mekkora az összesített ár.  
Ezen a felületen szükség van néhány számlázási adat megadására is, például a vezetéknévre és a keresztnévre. Aki már regisztrált, az egyszerűen felhasználhatja a korábban megadott adatait.  
A rendelés során ki lehet választani a kiszállítás dátumát, valamint meg kell jelölni, hogy a vásárló bankkártyával vagy készpénzzel szeretné kifizetni a rendelést.  
Végül a „Rendelés elküldése” gombbal lehet leadni a megrendelést.

13. kép A rendelés véglegesítését lebonyolító felület

## Korai Design Koncepciók

 Az alábbiakban bemutatjuk a korai design-terveket/koncepciókat, amelyekből később a jelenlegi, kiforrott felület született. Ezek a vázlatok mutatják, hogyan alakult az oldal a kezdeti ötletektől a végleges megoldásig.

**FŐOLDAL KONCEPCIÓJA**

A képen képernyőkép, szöveg, Téglalap, diagram látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

14. kép Főoldal design koncepciója

A korai vázlat egy egyszerű fejlécet (navbar) és három fő szekciót mutat be:

* HIRDETÉSEK: Akciók, újdonságok vagy promóciók számára kijelölt terület.
* LEGKELENDŐBB TERMÉKEK/TERMÉKEK: A kiemelt vagy legnépszerűbb termékek listája.
* KAPCSOLAT: Alapvető elérhetőségek vagy üzenetküldési lehetőségek.

**Bejelentkezési Felület koncepciója**

**A képen Téglalap, diagram, képernyőkép, tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.**

**15**. kép Bejelentkezés és Regisztrációs felület design koncepciója

A kezdeti tervekben a bejelentkezési és regisztrációs felület extrém minimalista megközelítést kapott.  A későbbi iterációkban ez a koncepció jelentősen átalakult. Hozzáadásra kerültek a formázott gombok, színkódolás, hover állapotok, és ezek mellett még a felhasználói élményt fokozó további funkciók.

Ez a fejlődés mutatja, hogy egy jó weboldal tervezése folyamatos finomítást igényel. A kezdeti egyszerű ötletekből alakul ki végül a kész, könnyen használható felület.

**KOSÁR KONCEPCIÓJA**

**A képen szöveg, képernyőkép, diagram, képernyő látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.**

1. kép Kosár design koncepciója

Ez a kosár oldal kezdeti vizuális koncepcióját mutatja be. A cél a funkcionalitás/használhatóság alapjainak lefektetése és az elrendezés irányának meghatározása a további részletezés előtt.

**Tartalma**  
A központi blokkban/részben jelennek meg a kosárba helyezett termékek. Minden egyes tétel három fő részből áll:

* Termékkép: Vizualizáció a gyors azonosításhoz.
* Termék neve: Rövid szöveges megnevezés a kép mellett.
* Ár és Darabszám-módosító: Jobb oldalon elhelyezve, az ár megjelenítése mellett plusz és mínusz gombok biztosítják az egyszerű darabszám-változtatást.

## FŐBB KÓDRÉSZLETEK

A képen szöveg, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

17. kép Termékek komponens html

A fenti képen a termekek.component.html kódrészlete látható, amely a webalkalmazásban a termékek listázásáért és kosárba helyezéséért felel. Az oldal elején egy fejléc található "Termék fajták" címmel, majd navigációs gombok következnek, amelyek segítségével a felhasználó az összes termék megtekintése mellett szűrhet is terméktípusok szerint. A kiválasztott típus gombja kiemelésre kerül az active-btn osztály segítségével.

A termékek kártyák formájában jelennek meg, amelyeken látható a termék képe, neve, valamint egy "Kosárba" gomb is. Ez a gomb csak akkor jelenik meg, ha a felhasználó be van jelentkezve, amit az auth.LoggedInStatus() függvény vizsgál.

**A képen szöveg, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.TERMÉKEK**

18. kép Termékek komponens ts

**METÓDUSOK**

ngOnInit()

* A komponens inicializálásakor fut le
* Betölti az összes terméket a SzolgaltatasService segítségével
* A betöltött adatokat elmenti a termekek tömbbe
* Meghívja a getTypes() metódust a terméktípusok összegyűjtéséhez

getAll()

* Újra betölti az összes terméket a szervízből
* Az adatokat a termekek3 tömbbe menti
* Beállítja a megjelenítési flag-eket:
* showProducts = true (összes termék látszik)
* showProducts2 = false (szűrt lista nem látszik)
* Nullra állítja a kiválasztott típust

getTypes()

* Kigyűjti az összes egyedi terméktípust a termekek tömbből
* Az üres értékeket kiszűri
* Az eredményt a tipusok tömbbe menti

getTermekByTypes(tipus: string)

* Szűri a termékeket a megadott típus alapján
* Az eredményt a termekek2 tömbbe menti
* Beállítja a megjelenítési flag-eket:
  + showProducts = false (összes termék nem látszik)
  + showProducts2 = true (szűrt lista látszik)

kosarba(termek: Pekaru)

* Hozzáadja a kiválasztott terméket a kosárhoz a szervíz segítségével

openSnackBar()

* Megjelenít egy értesítést (snackbar) a felhasználónak 3 másodpercig
* Egy külön SnackbarComponent-et használ ehhez

A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.**KOSÁR**

19. kép Kosár komponens ts

Ez a komponens egy bevásárlókosarat valósít meg, ahol a felhasználó megtekintheti, módosíthatja és kezelheti a kosár tartalmát.

**METÓDUSOK**

osszegSzamolas()

* Nullázza az összeget
* Végigmegy a kosár tartalmán
* Minden elemnél hozzáadja a termék árának és darabszámának szorzatát az összeghez

novel(termek: Pekaru)

* Növeli a kiválasztott termék darabszámát a kosárban
* Újraszámolja a kosár összértékét

csokkent(termek: Pekaru)

* Csökkenti a kiválasztott termék darabszámát a kosárban
* Újraszámolja a kosár összértékét

uritKosarat()

* Kiüríti a teljes kosarat
* Nullázza az összeget

torolElem(termek: Pekaru)

* Eltávolítja a kiválasztott terméket a kosárból
* Újraszámolja a kosár összértékét

**A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.FŐOLDAL**

**20**. kép Főoldal komponens html

Ez egy Angular komponens HTML szekciója/része, ami egy főoldalt valósít meg termékek megjelenítésével és egy képváltó carousellel.

Bootstrap carousel komponens automatikus lapozással (data-bs-ride="carousel")3 hirdetést/képet jelenít meg egymás után. A hirdetések után következő szekció tetején található egy link a teljes terméklistára ("/termekek" útvonalon).

.

**A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.METÓDUSOK:**

21. kép Főoldal komponens ts

openSnackBar()

* Funkció: Megjelenít egy értesítést (snackbar) 3 másodpercig
* Használat: A kosárba helyezés sikerét jelzi

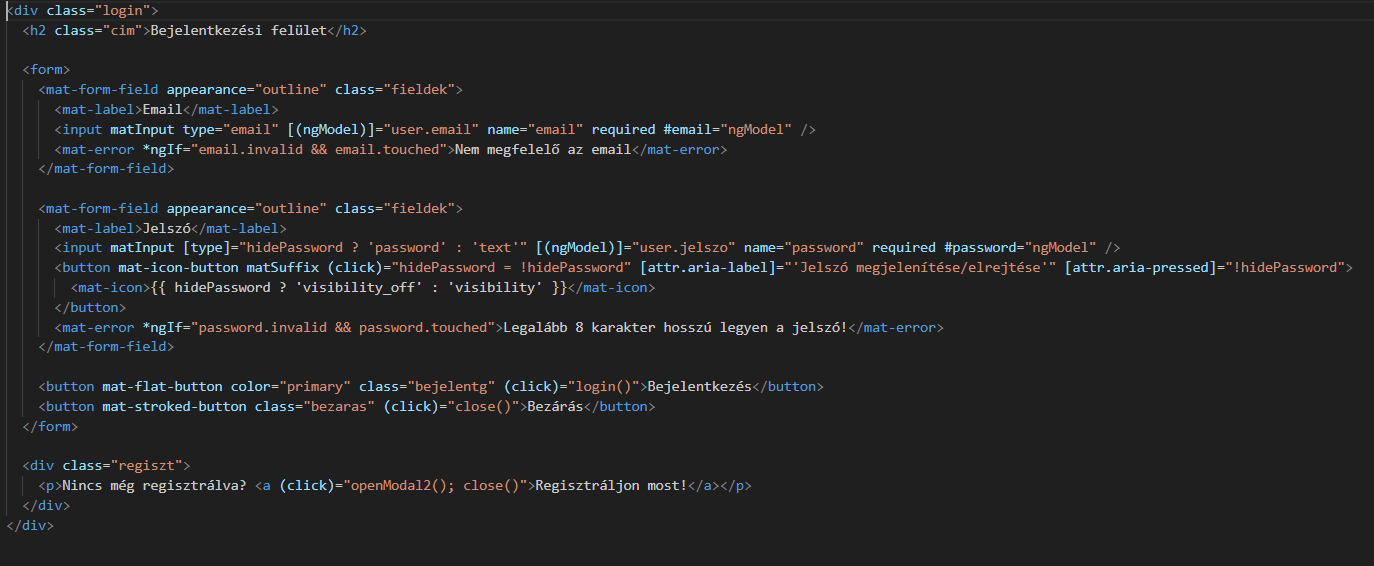
kosarba(termek: Pekaru)

* Funkció: Hozzáad egy terméket a kosárhoz a szervíz segítségével
* Paraméter: termek - A hozzáadandó termék

getFeherTypes()

* Kiszűri a 'Fehér' típusú termékeket
* Csak az első 3 ilyen terméket tartja meg
* Kigyűjti az egyedi terméktípusokat a tipusok tömbbe

**BEJELENTKEZÉS**

****

22. kép Bejelentkezési felület html kódja

A bejelentkezési oldalon két mező található: e-mail cím és jelszó. Ezek Angular Material elemek segítségével vannak megjelenítve, így az űrlap letisztult és egységes kinézetet kap. Az e-mail mező csak akkor érvényes, ha valódi e-mail címet írunk be, míg a jelszónál legalább 8 karaktert kell megadni.

A jelszó mező mellett van egy kis ikon, amivel be lehet kapcsolni a jelszó megjelenítését vagy elrejtését, attól függően, hogy mit szeretne a felhasználó.

Az űrlap alján két gomb található: a „Bejelentkezés” gomb, ami elindítja a bejelentkezési folyamatot, és egy „Bezárás” gomb, amivel az ablak bezárható.

Az oldal alján látható még egy szöveg is, ami a regisztrációra hívja fel a figyelmet. Ha valaki erre kattint, egy felugró ablakban tud új fiókot létrehozni.

A képen szöveg, képernyőkép, szoftver, Operációs rendszer látható

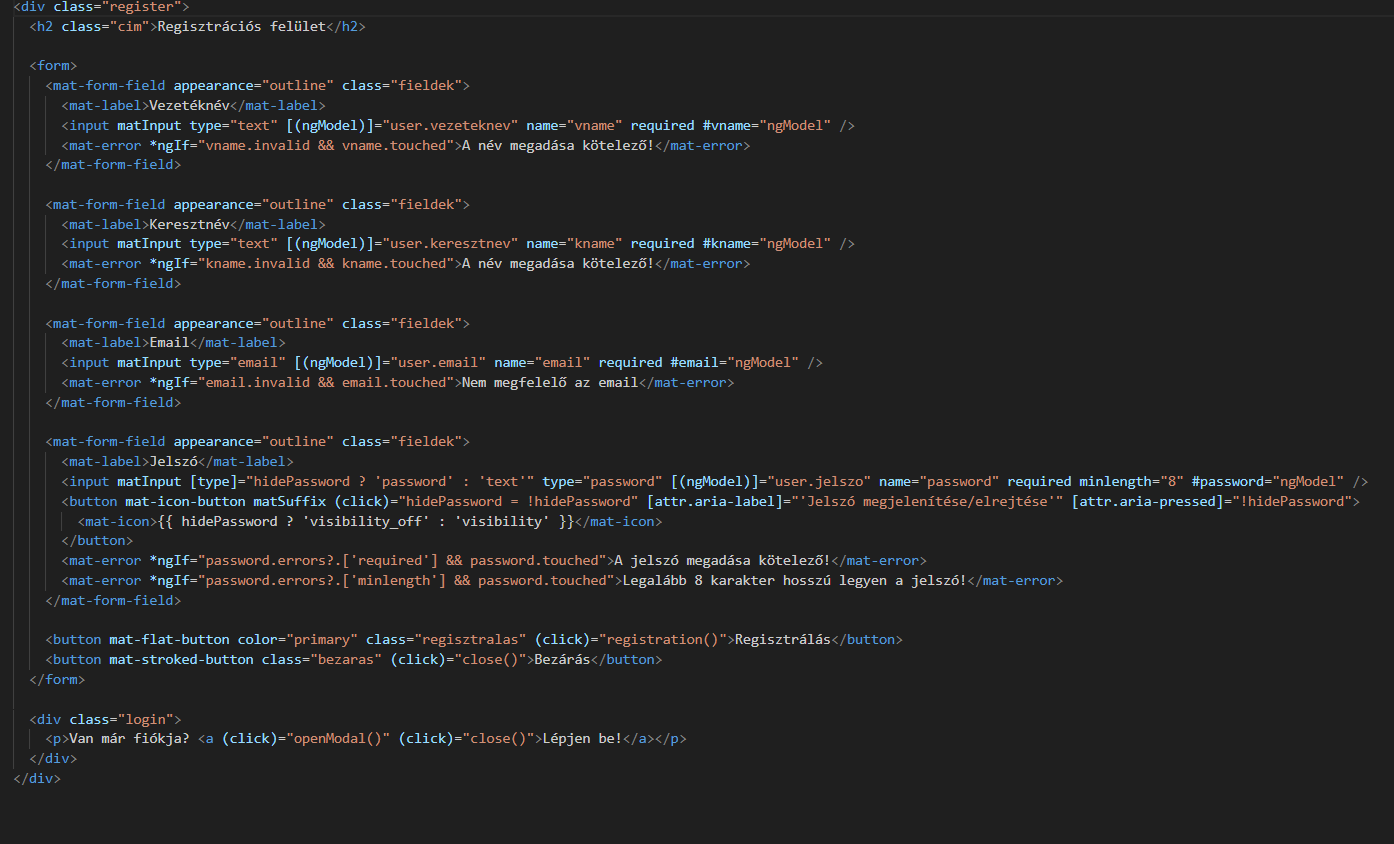
Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

23.kép A Bejelentkezési felület ts-e

A komponens felelős a bejelentkezési felület megjelenítéséért modálablakban. A felhasználó User típusú objektumba viszi be az adatait (vezetéknév, keresztnév, email, jelszó), majd a login() metódus ellenőrzi azokat. Sikeres bejelentkezés esetén bezáródik az ablak, hibás adatok esetén figyelmeztető üzenet jelenik meg.  
A hidePassword változó szabályozza, hogy a jelszó mező szövege látható legyen-e.  
A openModal2() metódus megnyitja a regisztrációs felületet egy újabb modálablakban, és kezeli a görgetés letiltását is a háttérben, amíg a modál aktív.

**23**. kép A Bejelentkezési felület ts-e

**REGISZTRÁCIÓ**

****

24. kép Regisztrációs felület html kódja

A regisztrációs felület célja, hogy a felhasználók új fiókot hozzanak létre az alkalmazásban. Az űrlap négy kötelező mezőt tartalmaz: vezetéknév, keresztnév, e-mail cím és jelszó. A mezők Angular Material formázással készültek, így egységes, kinézetet kapnak.

A névmezők (vezetéknév és keresztnév) egyszerű szövegmezők, amelyek csak akkor fogadhatók el, ha a felhasználó kitöltötte őket. Az e-mail mezőnél beépített ellenőrzés gondoskodik arról, hogy csak érvényes e-mail cím kerüljön beírásra.

A jelszó megadásánál minimum 8 karakter a követelmény és a mező mellett található egy ikon, amely lehetővé teszi a jelszó megjelenítését vagy elrejtését. Ez segíti a felhasználót a pontos beírásban. A rendszer minden esetben visszajelzést ad, ha egy mező hibásan lett kitöltve.

Az űrlap alján két gomb található: a „Regisztrálás” gomb a regisztrációs adatok beküldésére szolgál, míg a „Bezárás” gomb bezárja az ablakot. Az oldal alján elhelyezett link lehetőséget ad azoknak, akik már rendelkeznek fiókkal, hogy gyorsan átváltsanak a bejelentkezési felületre.

A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

25. kép Regisztrációs felülethez tartozó ts

A felhasználó egy User típusú objektumba adja meg adatait (vezetéknév, keresztnév, email, jelszó), ahol a jelszónak legalább 8 karakter hosszúnak kell lennie. A registration() metódus ellenőrzi ezeket az adatokat, majd sikeres validáció után elküldi a szervernek. Ha a regisztráció sikeres, automatikusan megtörténik a bejelentkezés, hiba esetén figyelmeztető üzenet jelenik meg. A jelszó láthatósága kapcsolható, a regisztrációs felület pedig bezárható vagy átváltható bejelentkezési módra.

**25**. kép Registrációs felülethez tartozó ts

## A képen szöveg, képernyőkép, szoftver, Weblap látható Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.TESZTELÉS

26.kép Frontend tesztelése

A képen egy Jasmine tesztfuttatás eredménye látható, amely 19 specifikációt (tesztesetet) tartalmaz, mindegyik sikeresen lefutott (0 hiba).

Az alkalmazás alapvető részei megfelelően működnek. (A véletlenszerű sorrend biztosítja, hogy a tesztek ne függjenek egymástól.)

# BACKEND DOKUMENTÁCIÓ

Ez a dokumentáció a pékségünkhöz tartozó API működését mutatja be, amely többek között lehetőséget biztosít új termékek hozzáadására, meglévő adatok lekérdezésére, frissítésére és törlésére is. Az API REST alapú, és JSON formátumban kommunikál. Az alábbi képen a Laravel migrációs fájlokat láthatjuk. Minden fájl egy-egy táblát hoz létre az adatbázisban pl.: Pékáruk, Felhasználók stb. A táblák közötti kapcsolatok is definiálva vannak, például a rendelések táblában a pékáru, felhasználó és cím külső kulcsként szerepel. A migrációk futtatásával (php artisan migrate --seed) az adatbázis automatikusan létrejön a megadott szerkezettel.

1.)

*SEQ kép \\* ARABIC 1. kép*

A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

27. kép: Laravel migrációs fájlok

*SEQ ábra \\* ARABIC 1. kép*

A képen képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A képen képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

28. kép A felhasználók táblájának migrációs fájlja

29. kép A rendelések táblájának migrációs fájlja

30. kép A pékáruk táblájának migrációs fájlja

*2. kép*

*2. kép*

*3. kép*

A képen képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

31. kép A címek táblájának migrációs fájlja

*4. kép*

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.2.)

Ez a kép a Laravel modell rétegét mutatja be. A $table változó meghatározza a kapcsolódó táblát, a $timestamps = false kikapcsolja az automatikus időbélyeg mezőket, míg a $guarded = [ ] engedélyezi az össze mező tömeges kitöltését. (Ez az összes táblánál el lett végezve)

*SEQ kép \\* ARABIC 3. kép*

Ez a kép a Laravel modell rétegét mutatja be.

A $table változó meghatározza a kapcsolódó

táblát, a $timestamps = false kikapcsolja

az automatikus időbélyeg mezőket,

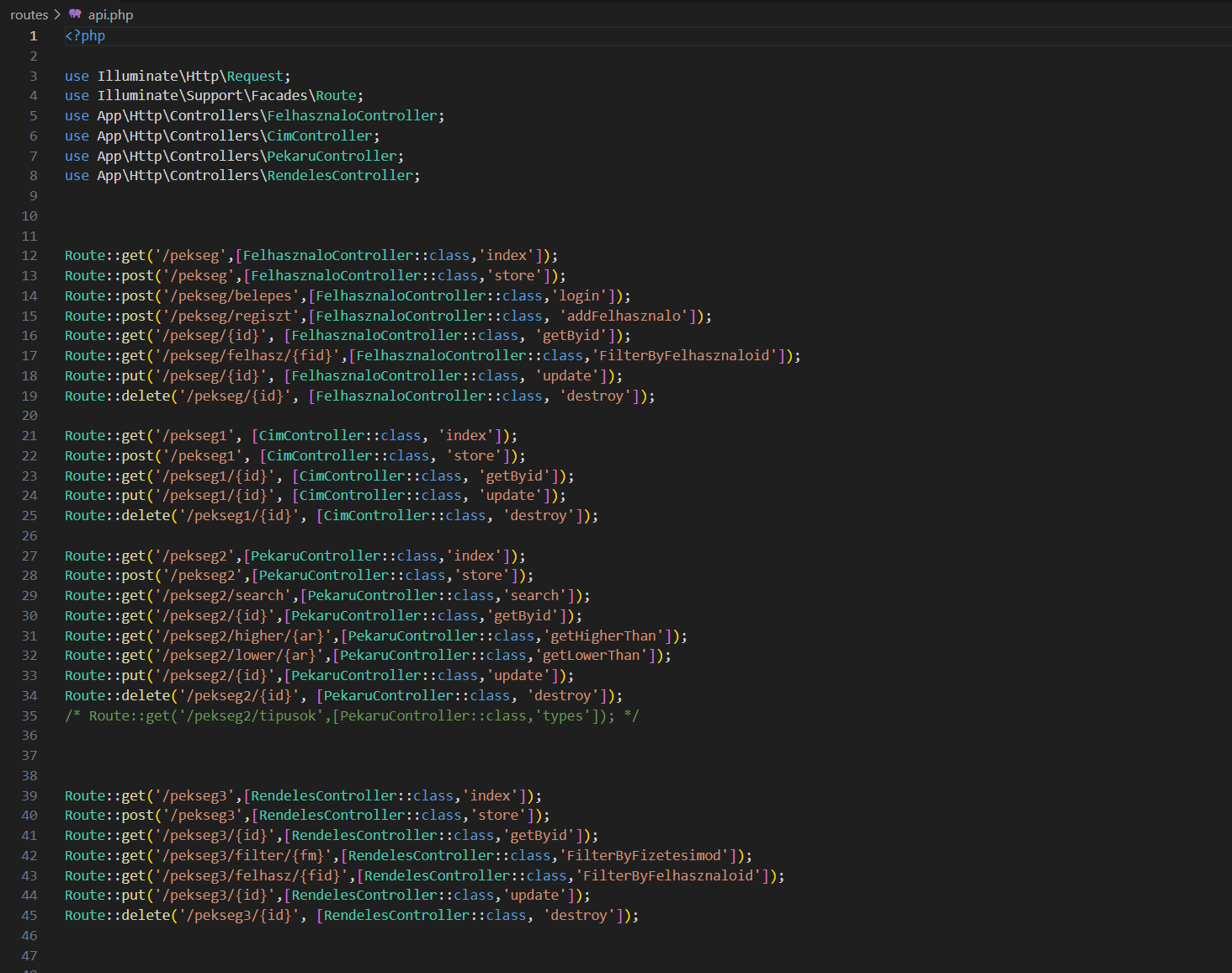
míg a $guarded = [ ] engedélyezi az összes

mező tömeges kitöltését.

(Ez az összes táblánál el lett végezve)

32. kép Laravel modell osztály

*6. kép*

3.) 

*33. kép API végpontok*

Itt a Laravel útvonalakat/útvonaldefiníciókat láthatjuk. Ezek az API végpontjait határozzák meg a különböző vezérlők számára.A **Route::get**, **Route::post**, **Route::put** és **Route::delete** metódusok különböző műveleteket végeznek, például adatokat kérnek le, hoznak létre, módosítanak vagy törölnek.

Mindegyikben egyaránt el van helyezve egy végpont a getByid, az Update, és Delete metódusoknak.

Ezeken kívül található SearchBy, Filterby, és Higher/LowerThan metódus is. Viszont ezek nem mindegyik Controllerben lettek implementálva.

## TESZTELÉS

### getByid:

A képen szöveg, Multimédiás szoftver, szoftver, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

*34. kép getById metódus sikeres tesztelése*

Reprezentáció a 4 getByid metódus egyikéről. A GET kérés a http://127.0.0.1:8000/api/pekseg1/ URL-re történik, és egy JSON-formátumú választ ad vissza az adott id-jú felhasználó, pékáru, vagy cím részleteivel/adataival.

A képen képernyőkép, szöveg, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

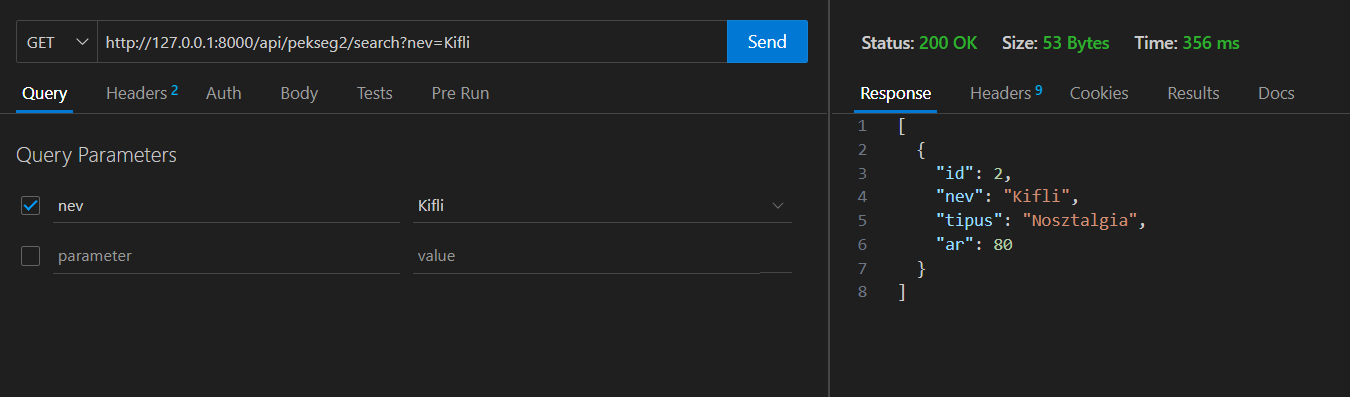
*35. kép getById metódus tesztje hibás adat/érték beírásakor*

Itt pedig az a hibaüzenet látható, státuszkóddal együtt, amit akkor kapunk vissza, ha például nem létező vagy rossz id-t adunk meg keresésre.

### Keresés név és típus alapján:

A search függvény egy request (kérés) objektumot kap bemenetként, amely alapján keresést végez az adatbázisban. Ha a kérés tartalmaz egy 'nev' (név) paramétert, akkor az adott név alapján szűri az eredményeket, míg, ha a 'tipus' (típus) paraméter szerepel benne, akkor a termék típusa szerint keres. Ha egyik paraméter sem található a kérésben, a függvény egy 400-as hibakódú válaszüzenetet küld vissza.A képen szöveg, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.



*36. kép Search metódus*

*SEQ kép \\* ARABIC 7. kép*

37. kép Search metódus sikeres tesztelése

*SEQ kép \\* ARABIC 8. kép*

*SEQ kép \\* ARABIC 11. kép*

### Update/Destroy:

Ez a kódrészlet egy API-t mutat be, amely a Pekaru tábla adatainak a frissítését (update) és törlését (destroy) végzi.

(A többi kontrollerben is megtalálható az a két metódus)

***Update:***A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

38. kép Az update metódus

*1 SEQ kép \\* ARABIC 2. kép*

Ez a függvény egy megadott id alapján keres egy Pekaru rekordot, majd a kapott adatokkal frissíti azt.

Validálja a kérést, és ha hiányzik a nev, tipus vagy ar mező, akkor 406 - Nem elfogadható hibát ad vissza.

Ha minden adat megfelelő, frissíti a rekordot.

***Destroy:***

Ez a függvény pedig egy adott id-jú Pekaru rekordot töröl az adatbázisból. Ha sikeres a törlés, akkor egy 205 – Reset Content státuszkódot küld vissza.

*1 SEQ kép \\* ARABIC 3. kép*

A képen képernyőkép, szöveg látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

39. kép A destroy metódus

### GetHigher/Lowerthan:

**A képen szöveg, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.**

40. kép A GetLower és GetHigher metódus

Ezek a függvények visszaadják azokat a pékárukat, amelyek ára magasabb és vagy alacsonyabb, mint a megadott érték ($ar).

Lehetővé teszi a pékáruk ár szerinti keresését egy weboldalon.A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen képernyőkép, Multimédiás szoftver, szoftver, szöveg látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

*SEQ kép \\* ARABIC 10. kép*

*42. kép*

*A getHigher metódus sikeres tesztelése*

*41. kép*

*A getLower metódus sikeres tesztelése*

*SEQ kép \\* ARABIC 11. kép*

### FilterByFizetesimod:

A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

43. kép A FilterByFizetesimod metódusa

*SEQ kép \\* ARABIC 12. kép*

*17. kép*

A FilterByFizetesimod lehetővé teszi a Rendelések szűrését fizetési mód alapján, és visszaadja a találatokat, ha léteznek. Két féle fizetési mód jöhet szóba: Kártyás és készpénzesHa a keresésnél 0-t adunk meg akkor a készpénzzel kívánt fizetési kérelmeket kapjuk meg. Az 1-es megjelölés pedig a kártyával való fizetést jelenti.

*SEQ kép \\* ARABIC 13. kép*

*18. kép*

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

44. kép A FilterByFizetesimod sikeres tesztelése

### Seeder:

A DatabaseSeeder osztály feladata az adatbázis feltöltése kezdeti adatokkal a Laravel keretrendszerben A seeder biztosítja, hogy a rendszer mindig rendelkezzen alapadatokkal, amelyeket a fejlesztés és tesztelés során felhasználhatunk.

Az alábbi modelleket használja:

* **Felhasznalo**: Felhasználói adatok tárolása
* **Pekaru**: Pékáruk tárolása
* **Cim**: Felhasználókhoz tartozó címek kezelése
* **Rendeles**: Megrendelések kezelése

A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

45. kép DatabaseSeeder kódja

*19. kép*

A run() metódus feltölti az adatbázist tesztadatokkal. Először létrehozza a felhasználókat névvel, jelszóval és email-címmel, majd hozzáadja a pékárukat típussal és árral. Ezután a felhasználók címei és telefonszámai kerülnek rögzítésre. Végül a rendelések kerülnek mentésre a kapcsolódó adatokkal, beleértve a pékáru azonosítót, számlázási adatokat, dátumokat és fizetési módokat. Az adatokat a megfelelő modellek create() metódusa menti el.

### Bejelentkezési (login):

A login metódus ellenőrzi a felhasználó hitelesítő adatait. Megkeresi a felhasználót az email cím alapján, majd összehasonlítja a megadott jelszót a tárolt jelszóval. Ha minden egyezik, visszaadja a felhasználó adatait.

### addFelhasznalo:

Az addFelhasznalo metódus új felhasználót regisztrál. Ellenőrzi, hogy minden kötelező mező (név, email, jelszó) ki legyen töltve. Ha valami hiányzik, hibát jelez. Ha minden rendben van, elmenti az adatokat és visszaadja az új felhasználót.

**A képen szöveg, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.**

46. kép A login és addFelhasznalo metódusok

# TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

A jelenlegi verzió megfelelően használható, de a fejlesztés során több olyan funkció ötlete is felmerült, amit a jövőben érdemes lenne beépíteni:

* A főoldalon lévő legfelkapottabb termékek megjelenítése megvásárolt mennyiség alapján
* Intézmények értékelhetik a kiszállítást és a termékeket
* Csak elérhető termékek szerepeljenek a kínálatban
* Mobilbarát felület vagy külön app fejlesztése

# IRODALOMJEGYZÉK

1. <https://angular.io/docs>
2. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Methods>
3. <https://inf.u-szeged.hu/~gnemeth/adatbgyak/exe/EK_diagram/az_egyedkapcsolat_diagram_elemei.html>
4. <https://laravel.com/docs>
5. <https://martinjoo.dev/layered-architectures-with-laravel>
6. <https://www.fornetti.hu>