Tartalom

[Bevezetés 2](#_Toc189822729)

[Projekttervezet 4](#_Toc189822730)

[Felhasznált programok 7](#_Toc189822731)

[Adatbázis 11](#_Toc189822732)

[Tesztelés 13](#_Toc189822733)

[Fejlesztések 14](#_Toc189822734)

[Reflektálás 15](#_Toc189822735)

[Források 17](#_Toc189822736)

**Projekttervezet**

# Bevezetés

**Előszó**

Projektünket Szentes Mirtill és Nagy Máté János közösen készítettük el, amely a Valai Pékség nevet viseli. A projekt témájának kiválasztásakor fontos szempont volt, hogy egy olyan területtel foglalkozzunk, amelyhez személyesen is kötődünk, és amelyben szívesen mélyedünk el. A pékség, mint téma közel áll hozzánk, mivel mindketten szeretjük a friss pékárukat, és érdekesnek találtuk a helyi vállalkozások digitalizációs lehetőségeit.

A név kiválasztásánál több lehetőséget is megfontoltunk, azonban a Valai Pékség név volt az, amely leginkább visszaadta a projekt által képviselt barátságos és autentikus hangulatot.

**Feladat megosztás**

A projekt megvalósítását szándékosan egyenlően osztottuk el egymás között, mivel úgy gondoltuk, hogy mindenki számára hasznos, ha részt vesz a munka minden aspektusában.

**Ezáltal:**

* Minden területen lehetőségünk volt fejlődni és új ismereteket szerezni.
* Könnyebben tudtunk segíteni egymásnak, mivel mindkettőnk pontosan ismerte a projekt összes részét.
* A folyamatos együttműködés erősítette a csapatmunkát, és hatékonyabbá tette a problémamegoldást.

A projekt feladatait közösen terveztük meg és ütemeztük be, hogy a munkafolyamatok átláthatok és jól szervezettek legyenek.

**A projekt célja**

A projekt célja egy olyan webalkalmazás elkészítése, amely a Valai Pékség online jelenlétét biztosítja. Az alkalmazás fő funkciói között szerepel a pékség termékeinek bemutatása, az online rendelési lehetőség megteremtése, valamint a vásárlók és a pékség munkatársai közötti kommunikáció elősegítése.

**A webalkalmazás elsődleges célcsoportjai:**

* Magánszemélyek, akik kényelmesen szeretnének friss pékárut rendelni otthonukból.
* A pékség munkatársai, akik az online rendelések kezeléséért és nyomon követéséért felelősek.

A projekt hosszú távú célja a digitalizáció előnyeinek bemutatása egy helyi vállalkozás esetében, valamint az online vásárlási élmény egyszerű és élvezetes kialakítása a felhasználók számára.

# Projekttervezet

**Felhasználói felület (UI) tervezés**

A felhasználói felület célja, hogy a látogatók számára egyszerű, mégis esztétikus és funkcionális élményt nyújtson. Az első benyomás kulcsfontosságú, ezért a főoldal színes, letisztult dizájnnal rendelkezik. Fontosnak tartottuk, hogy passzoljon a Pékség témájához, de ne legyen zsúfolt, így bézs-aranybarna árnyalatok kerültek előtérbe.

**A fő elemek között megtalálható:**

* Pékség logója, amely az oldal vizuális azonosítója.
* Felhasználói értékelések, hogy növeljék a bizalmat az új látogatókban.
* Termékekről készült képek, amelyek az ajánlatok minőségét és változatosságát emelik ki.

**A webalkalmazás fő funkciói:**

* **Termékek böngészése és részletezése:**
  + Az érdeklődők megtekinthetik a termékek listáját képekkel és leírásokkal.
  + A termékek nevére kattintva részletes információkat kapnak az adott áruról, például összetevőkről, árakról és elérhetőségről.
* **Kosár és rendelési folyamat:**
  + A kiválasztott termékek a kosárba kerülnek, ahol az árak automatikusan összeadódnak.
  + A rendelési folyamat végig vezeti a vásárlókat a szállítási és fizetési információk megadásán.
* **Reszponzív navigáció:**
  + A navigációs sáv a reszponzív dizájnnak köszönhetően hamburger menüként jelenik meg kisebb eszközökön.

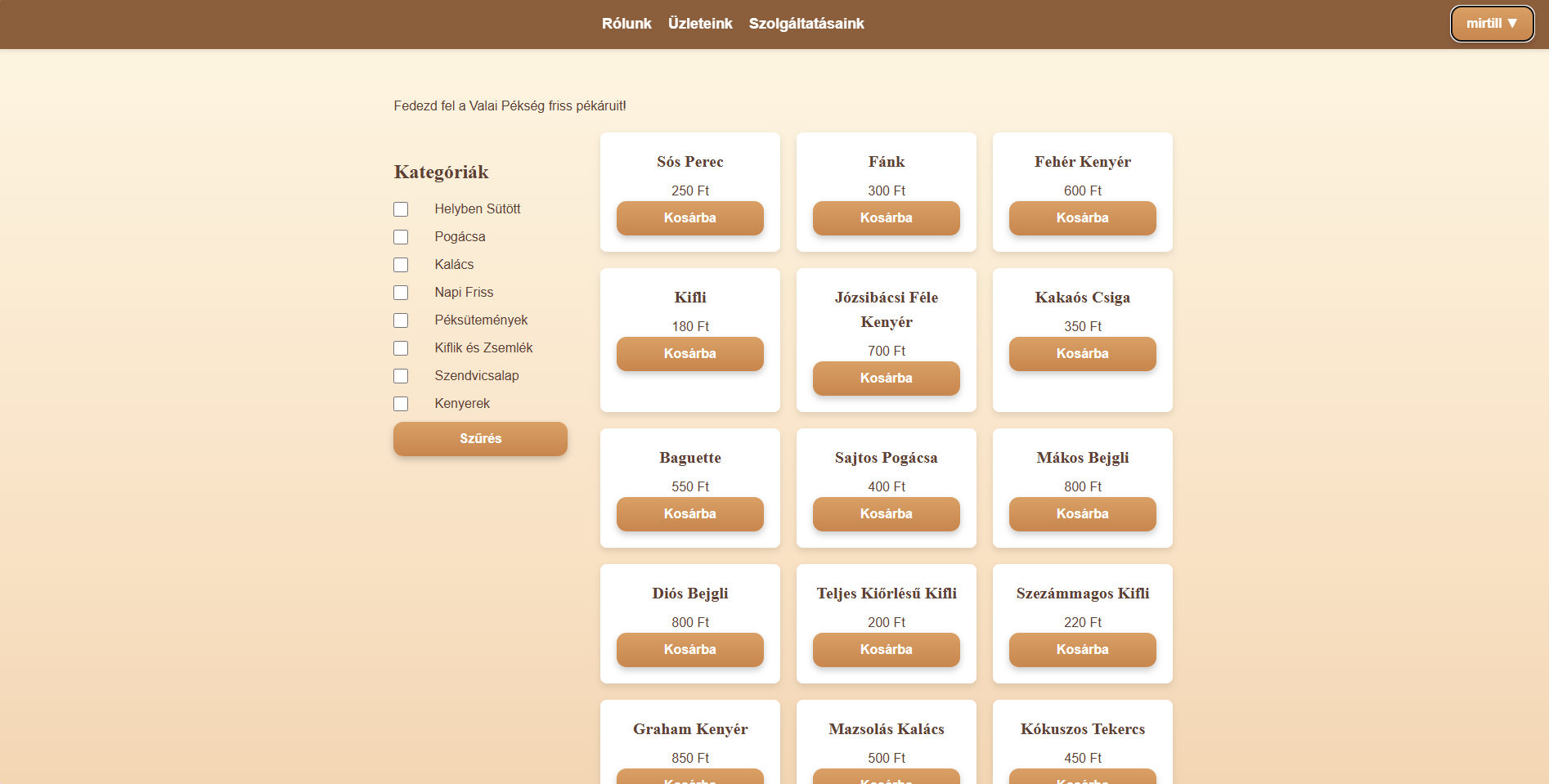
**Navigációs sáv tartalma:**

* **Rólunk:** Részletesebb bemutató a Valai Pékség történetéről és filozófiájáról.
* **Üzleteink:**
* **Szolgáltatásaink**:
* **Kapcsolat:** Információk az ügyfélszolgálat elérhetőségéről (email, telefonszám, cím).
* **Bejelentkezés és Regisztráció:** A rendeléshez szükséges, de a böngészéshez opcionális.
* **Termékek:** Visszavisz a főoldalra, ha már nem azon tartózkodunk, ahol a vevők a termékeink között válogathatnak. Addig nem látható a navigációs sávban.

**Bejelentkezés után, egy legördülő menüben:**

* **Kosár:** Áttekintést nyújt a vásárlásra kiválasztott termékekről, összegzéssel és megrendelési lehetőséggel.
* **Rendeléseim:** Megjeleníti előttünk az eddigi rendeléseinket.
* **Adatok módosítása**: Saját adatainkat tudjuk módosítani
* **Kijelentkezés**: Kijelentkezik a fiókból.

**Főoldal**

A főoldalon láthatjuk a termékeinket, melyeket képekkel egészítettünk ki. A termékekre rákattintva láthatjuk azoknak részletes leírásait is. A bal oldali sávban tudunk kategóriákra szűrni, mely megkönnyíti dolgunkat, ha egy adott terméket, vagy kategóriában keresünk.

**A fő elemek között megtalálható:**

* Pékség logója, amely az oldal vizuális azonosítója.
* Felhasználói értékelések, hogy növeljék a bizalmat az új látogatókban.
* Termékekről készült képek, amelyek az ajánlatok minőségét és változatosságát emelik ki.

**Regisztráció felület**

A regisztráció során a felhasználó egyszerűen és biztonságosan megadhatja adatait. Az adatokat a program az adatbázisba feltölti, a jelszó titkosítással. A már beregisztrált adatokkal tudunk bejelentkezni az oldalunkra. A termékeket regisztráció és bejelentkezés nélkül is látjuk, de rendelni csak ezek teljesítése után lesz lehető. Amennyiben ezt nem teljesíti, hiba üzenetet kapunk, ami felszólít a bejelentkezésre.

**A következő információkat kérjük be:**

* **Felhasználónév**
* **Email cím**
* **Jelszó:** kétszer kérjük be az egyezés ellenőrzéséhez

**Bejelentkezés felület**

**A bejelentkezés során a felhasználó az alábbi adatokat adja meg:**

* Felhasználónév
* ****Jelszó

**Hibakezelés:**

* Ha az email cím nem található az adatbázisban, figyelmeztetést jelenítünk meg: „Ez az email cím még nem került regisztrálásra.”
* Ha az email cím helyes, de a jelszó nem egyezik, egy újabb hibaüzenet tájékoztatja a felhasználót.

**Sikeres bejelentkezés után:**

* A felhasználó a kezdőlapra kerül.
* A navigációs sávban a bejelentkezés és regisztráció opció helyett egy kijelentkezési lehetőség jelenik meg.

**Rendelés felület**

A rendelési folyamatot a kosárban található „Megrendelés” gomb indítja el.

**Az alábbi elemek találhatók a rendelés oldalon:**

* **Rendelés összesítése:**
  + A termékek ára, szállítási költség, valamint a teljes fizetendő végösszeg.
* **Szállítási adatok:**
  + Név
  + Szállítási cím
  + Telefonszám
* **Számlázási adatok:**
  + Név
  + Számlázási cím
* **Fizetési lehetőségek:**
  + Bankkártyás fizetés
  + Utánvételes fizetés
* **Továbbá:**
  + Céges rendelés esetén egy külön „Kapcsolattartó” szekció jelenik meg, ahol a kapcsolattartó nevét, címét és telefonszámát kell megadni.
  + Az oldal a rendelés véglegesítése után visszaigazoló üzenetet jelenít meg, amely tartalmazza a várható szállítási időt.

# Felhasznált programok

**Discord**

A Discord egy sokoldalú kommunikációs platform, amelyet a projekt során elsősorban az egymás közötti kommunikációra használtunk.

**Az alkalmazás lehetővé tette, hogy:**

* Hívásokat bonyolítsunk le, amelyek során részletesen megbeszélhettük a projekt különböző részeit és megoldásait.
* Részfeladatok screenshot-jainak tárolását és megosztását végezzük el, így mindkét fél könnyedén hozzáférhetett a releváns információkhoz és képekhez.
* Az archiválási lehetőség révén visszakereshetővé váltak a korábbi beszélgetések és megosztott anyagok, amelyek fontos szerepet játszottak az együttműködés folytonosságában.

A Discord intuitív felülete és stabil teljesítménye nagyban hozzájárult a hatékony csapatmunkához.

**Figma**

A Figma egy online, kollaborációra optimalizált design eszköz, amelyet a projekt vizuális tervezési szakaszában használtunk.

**A Figmában:**

* Weboldalterveket készítettünk, ideértve a főoldal, a regisztrációs és bejelentkezési oldal, valamint a rendelési felület vizuális megjelenésének kidolgozását.
* Valós idejű együttműködést tett lehetővé, ahol mindketten egy időben dolgozhattunk a terven, követve egymás kurzorának mozgását és azonnal látható változtatásait.
* A kommentálási funkció segítségével könnyedén adhattunk egymásnak visszajelzéseket, ami gyorsította a döntéshozatalt és a módosítások végrehajtását.

Ez a program kiemelkedő rugalmasságot biztosított számunkra, mivel platformfüggetlen, és bárhonnan elérhető, amíg van internetkapcsolatunk.

**Visual Studio Code**

A Visual Studio Code volt az elsődleges fejlesztési környezetünk a React frontend fejlesztéséhez.

**A program több szempontból is előnyös volt számunkra:**

* Technológiai támogatás: A projekt React, TypeScript, HTML és CSS alapú, amelyekhez a VS Code hatékony eszköztárat biztosított.
* Bővítmények és integrációk: Használtunk automatikus kódkiegészítést, fejlett hibakereső eszközöket és GitHub integrációt.
* Verziókövetés: A GitHub támogatás révén a kódot közvetlenül a VS Code-on belül kezelhettük és verziózhattuk.

A Visual Studio Code intuitív felülete és könnyű használhatósága jelentősen megkönnyítette a fejlesztési folyamatot.

**Visual Studio 2022**

A Visual Studio 2022 volt az elsődleges fejlesztési környezetünk, amelyben az ASP.NET Core backend fejlesztését végeztük.

**A program előnyei:**

* Korszerű fejlesztői eszközök: Beépített hibakereső, automatikus kódkiegészítés és Entity Framework Core támogatás.
* Projektmenedzsment támogatás: Könnyen kezelhető megoldásokat biztosított az ASP.NET Core API fejlesztéséhez.
* GitHub és Git integráció: Verziókövetési és együttműködési lehetőségeket kínált közvetlenül a fejlesztői környezetben.

A Visual Studio 2022 biztosította a stabil és hatékony fejlesztési környezetet a backend számára.

**Xampp**

Az XAMPP egy könnyen telepíthető szoftvercsomag, amely a webes fejlesztésekhez szükséges szervereket és eszközöket tartalmazza.

* Adatbáziskezelés: A PhpMyAdmin felületén hoztuk létre és kezeltük az adatbázisunkat, amely tartalmazza a felhasználók, termékek és rendelések adatait.
* Helyi szerver környezet: Az XAMPP lehetővé tette, hogy helyi környezetben futtassuk és teszteljük az alkalmazást, így gyorsabban azonosíthattuk és javíthattuk az esetleges hibákat.
* Kompatibilitás: A program kiválóan működik több operációs rendszeren is, és egyszerűen kezelhető grafikus felülete megkönnyítette a használatát.

**GitHub**

A GitHub volt a projektünk verziókövetési és tárolási platformja.

**Az alkalmazás számos előnyt nyújtott:**

* Kód tárolása: A projekt forráskódja biztonságosan tárolva volt, így bármikor hozzáférhettünk.
* Verziókezelés: Az egyes fejlesztési szakaszok pontos nyomon követésére és a korábbi verziók visszaállítására is lehetőséget biztosított.
* Csapatmunka támogatása: A pull request-ek és merge request-ek segítségével egyszerűen egyesíthettük az egyéni fejlesztési munkáinkat.
* Nyilvános hozzáférés: A projekt végén a GitHub platformja biztosította, hogy az elkészült munka megosztható legyen másokkal is.

A GitHub nemcsak a kód tárolásában segített, hanem a csapatmunka szervezésében és a fejlesztési folyamat dokumentálásában is kulcsszerepet játszott.

**PowerPoint**

A Microsoft PowerPoint egy prezentációkészítő szoftver, amelyet a projekt során az eredmények és a fejlesztési folyamat bemutatására használtunk.

**A PowerPoint lehetőségei:**

* Diavetítések létrehozása: A projekt során készített tartalmak vizuális és strukturált bemutatása.
* Multimédiás elemek beágyazása: Képek, videók és animációk alkalmazása a bemutató érthetőségének növelésére.
* Együttműködés: Lehetővé teszi a csapat számára a közös szerkesztést és visszajelzések megosztását.
* Professzionális dizájn: Előre elkészített sablonok és testre szabható elrendezések segítették a vizuális megjelenés egységességét.
* Könnyű exportálás: Az elkészült bemutatók megoszthatók különböző formátumokban (PDF, PPTX, videó).

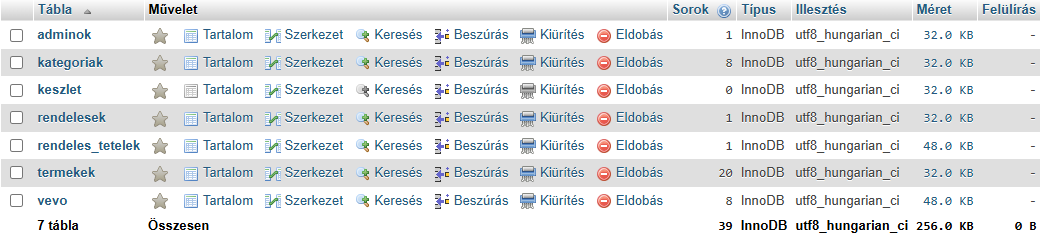
A PowerPoint segítségével a projekt összegzését és bemutatását hatékonyan tudtuk megvalósítani, biztosítva az érthetőséget és a professzionális megjelenést.

# Adatbázis

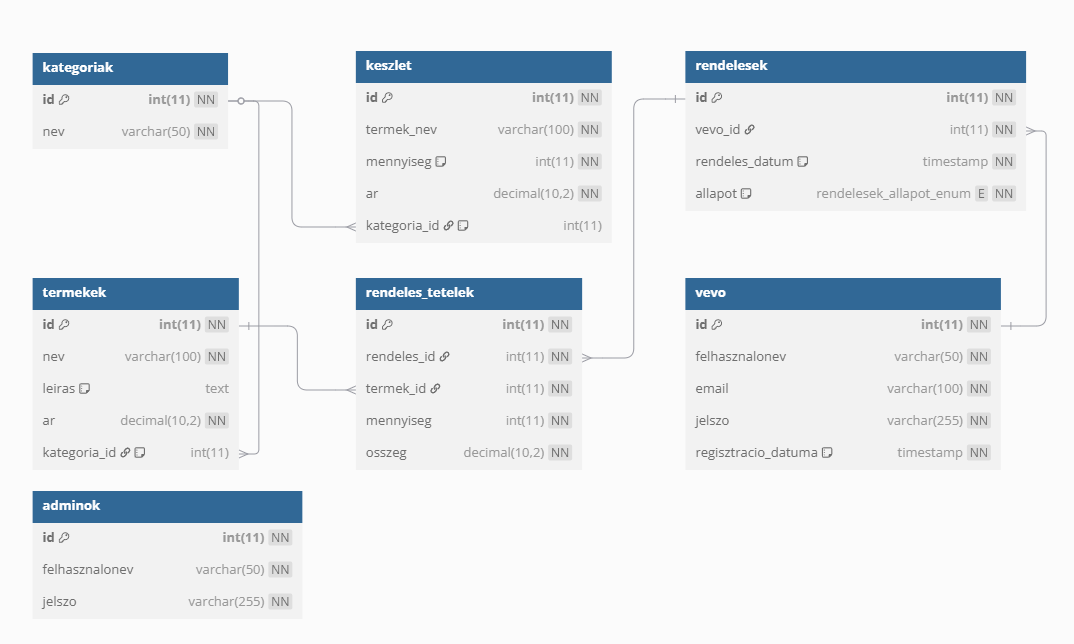
Az alkalmazásunk adatkezeléséhez MySQL-alapú adatbázist használunk, amelyet az ASP.NET backendben az Entity Framework Core ORM és a Visual Studio 2022 segítségével kezelünk. Fejlesztés során az adatbázist egy helyi MySQL szerveren futtattuk, míg az éles környezetben egy távoli adatbázis-kiszolgálóra kerül. Az adatkapcsolatot és a lekérdezések végrehajtását az ASP.NET Web API biztosítja, amely a React frontend számára továbbítja az adatokat.

**Az adatbázis főbb jellemzői:**

* Struktúra: Az adatokat külön táblákban tároljuk, amelyek a felhasználók, termékek, rendelések és kategóriák adatait kezelik.
* Kapcsolatok: Az adatbázis kapcsolatokkal van optimalizálva, így biztosítva az adatok konzisztenciáját és gyors elérését.
* Kezelés és karbantartás: Az adatbázis-műveleteket az Entity Framework Core ORM segítségével végezzük, amely egyszerűsíti a migrációkat és a lekérdezéseket.
* Biztonság: A felhasználói adatok védelméről titkosítási és jogosultsági szintek gondoskodnak.
* Hatékonyság: Az adatbázis indexelése és optimalizált lekérdezései biztosítják a gyors válaszidőt még nagy mennyiségű adat esetén is.

**Táblák**

1. Adminok tábla
   * Tárolja az adminisztrátorok adatait (id, felhasználónév, jelszó).
   * Az id mező az egyedi azonosító (PRIMARY KEY), míg a felhasznalonev UNIQUE kulcs
2. Kategoriak tábla
   * Tárolja a termékkategóriákat
   * Az id egyedi azonosító, a nev pedig UNIQUE kulcs, biztosítva, hogy ne legyenek duplikált kategórianevek.
3. Termekek tábla
   * Tárolja a termékek adatait (id, név, leírás, ár, kategória\_id, kép URL).
   * Az id mező az egyedi azonosító. A kategoria\_id itt is idegen kulcs, ami a kategoriak táblára mutat. A kategória törlésekor a mező értéke szintén NULL lesz.
4. Rendelesek tábla
   * Tárolja az egyes rendelések adatait (id, vevő\_id, rendelés dátuma, állapot).
   * Az id mező az egyedi azonosító. A vevo\_id idegen kulcs a vevo táblára. Ha egy vevőt törölnek, a hozzá kapcsolódó rendelések is törlődnek (ON DELETE CASCADE).
5. Keszlet tábla
   * Tárolja a termékek készletinformációit (id, termék neve, mennyiség, ár, kategória\_id).
   * Az id egyedi azonosító. A kategoria\_id idegen kulcs, amely a kategoriak táblára mutat. Amennyiben egy kategória törlődik, az ON DELETE SET NULL szabály érvényesül, így a kategória törlésével a kapcsolódó mezők NULL értéket kapnak.
6. Rendeles\_tetelek tábla
   * Tárolja a rendelésekhez tartozó termékeket és azok mennyiségét (id, rendelés\_id, termék\_id, mennyiség, összeg).
   * Az id egyedi azonosító. A rendeles\_id a rendelesek táblára mutat, míg a termek\_id a termekek táblára. Ha egy rendelést vagy terméket törölnek, az ide kapcsolódó sorok is törlődnek (ON DELETE CASCADE).
7. Vevo tábla
   * Tárolja a vásárlók adatait (id, felhasználónév, email, jelszó, regisztráció dátuma).
   * Az id mező az egyedi azonosító. A felhasznalonev és az email mezők UNIQUE kulcsok, amelyek megakadályozzák a duplikált regisztrációkat.

**ER Adatmodell**

A kapcsolatok kialakítása miatt, ha az egyik táblában változik valami, akkor a másik tábla is módosításra kerül. Ha például egy táblában törlünk egy rekordot, ami más táblában hivatkozásokkal rendelkezik, az adatbázis beállításaitól függően az adott hivatkozásokat is automatikusan frissíteni, törölni vagy újra értelmezni kell. Ez segít elkerülni az inkonzisztens adathalmazokat, és biztosítja, hogy az adatok közötti kapcsolatok mindig megfelelőek maradjanak.

**Kapcsolatok és integritási szabályok**:

**Kategóriák és termékek/készlet:**

A kategóriák tábla azonosítóit idegen kulcsként használják a termékek és készletek táblái. Így egy termék vagy készlet sor egy adott kategóriához tartozik. A kategória törlése esetén a kapcsolódó sorok kategória mezői NULL értéket kapnak.

**Vevők és rendelések:**

A vevő tábla azonosítóját használják a rendelések tábla idegen kulcsaként. Ha egy vevőt törölnek, az összes rendelése automatikusan törlődik (CASCADE).

**Rendelések és rendelés tételek:**

Egy rendeléshez több rendelés tétel tartozhat, amit a rendelés\_id oszlop határoz meg. Ha egy rendelést törölnek, a hozzátartozó tételek is automatikusan törlődnek (CASCADE).

**Rendelés tételek és termékek:**

A rendelés tételek tábla termék\_id oszlopa a termékek tábla azonosítójára mutat. Ha egy terméket törölnek, az ehhez tartozó rendelés tételek szintén törlődnek (CASCADE).

Ez az adatbázis-struktúra biztosítja az alkalmazás hatékony adatkezelését, lehetővé téve a termékek, rendelések és készletek precíz nyomon követését. Az adatok biztonságos tárolása és az optimalizált lekérdezések révén a rendszer gyors és megbízható működését támogatja.

# Tesztelés

**Funkcionális tesztelés**

 Ellenőrizzük, hogy az összes funkció (regisztráció, bejelentkezés, rendelés, kosár) megfelelően működik-e.

 Biztosítjuk, hogy a felhasználók minden esetben a várt visszajelzést kapják (pl. hibás jelszó esetén figyelmeztetés).

**Reszponzív tesztelés**

* Teszteljük az alkalmazást különböző eszközökön (asztali számítógép, tablet, mobiltelefon).
* Ellenőrizzük a navigáció, képek, és tartalom helyes megjelenését különböző képernyőméreteken.

**Terhelési tesztelés**

* Vizsgáljuk, hogy a rendszer hogyan kezeli a nagy mennyiségű párhuzamos felhasználói műveletet, például több rendelés egyszerre történő leadását.

**Biztonsági tesztelés**

 Teszteljük a jelszavak és személyes adatok tárolásának biztonságát.

 Megvizsgáljuk, hogy a rendszer védett-e az SQL injekció és más alapvető támadási formák ellen.

# Fejlesztések

**Kiterjesztett termékkategóriák**

További termékkategóriák hozzáadása, mint például gluténmentes vagy vegán termékek.

**Felhasználói profil**

A regisztrált felhasználók számára részletes profiloldal létrehozása, ahol láthatják korábbi rendeléseiket, módosíthatják személyes adataikat.

**Értékelések és vélemények**

Lehetőség biztosítása a felhasználók számára, hogy értékeljék a termékeket és véleményt írjanak.

**Push értesítések**

Bevezetjük az értesítések funkciót, hogy a vásárlók értesülhessenek az új termékekről, akciókról vagy a rendelésük állapotáról.

**Mobilalkalmazás**

A jövőben a webalkalmazás mellé natív mobilalkalmazás készítése iOS és Android platformokra.

# Reflektálás

**Csapatmunka és feladatmegosztás**

* Megtanultuk, hogyan lehet hatékonyan együtt dolgozni egy fejlesztői csapatban.
* A feladatok megfelelő kiosztásával és felelősségi körök meghatározásával gördülékenyebbé tettük a munkafolyamatot.
* A rendszeres kommunikáció és a közös problémamegoldás segített a hatékony együttműködésben.

**Technikai fejlődés**

* Mélyebb ismereteket szereztünk a reszponzív designról, az adatbázis-kezelésről és a biztonsági alapelvekről.
* Az Entity Framework Core és a MySQL használatával gyakorlati tapasztalatot szereztünk az adatkezelésben.
* Az ASP.NET Web API és a React integrációjának megvalósítása során elmélyítettük tudásunkat a modern webfejlesztési technológiákban.

**Problémamegoldás és rugalmasság**

* A projekt folyamán felmerült nehézségeket sikerült kreatívan és hatékonyan megoldani.
* Hibakezelési stratégiákat alkalmaztunk, hogy minimalizáljuk a rendszerösszeomlásokat és az adatvesztéseket.
* A váratlan technikai akadályok ellenére rugalmasan tudtunk reagálni, és gyorsan alkalmazkodtunk az új kihívásokhoz.

**Felhasználói élmény és teljesítményoptimalizálás**

* Törekedtünk arra, hogy a felhasználói felület intuitív és könnyen kezelhető legyen.
* Az alkalmazás teljesítményét optimalizáltuk a gyors betöltési idő és a gördülékeny működés érdekében.
* Teszteléssel és visszajelzések alapján folyamatosan finom hangoltuk a rendszert.

**Összegzés és jövőbeli fejlesztési irányok**

* A projekt során megszerzett tapasztalatok segítenek minket a jövőbeli fejlesztések során.
* Az eddigi munkánkat alapul véve tovább fejleszthetjük az alkalmazást új funkciókkal, például mobilalkalmazás-támogatással vagy mesterséges intelligencia alapú ajánlórendszerrel.
* A következő lépések között szerepel a felhasználói visszajelzések beépítése és a biztonsági rések további csökkentése.

Ez a projekt nemcsak technikai tudásunkat bővítette, hanem hozzájárult a csapatmunkában és problémamegoldásban való fejlődésünkhöz is.

# Források

**Fejlesztési dokumentációk és segédanyagok**

* HTML és CSS dokumentáció: MDN Web Docs – Részletes leírás és példák a modern webfejlesztéshez.
* JavaScript és React dokumentáció: React hivatalos dokumentáció – A React könyvtár hivatalos útmutatója és dokumentációja.
* ASP.NET Core dokumentáció: Microsoft Learn – Az ASP.NET és Entity Framework használatához szükséges források.
* MySQL dokumentáció: MySQL Reference Manual – Az adatbázis-kezelés alapjai és haladó technikák.

**Tervezési és fejlesztési eszközök**

* Figma: Weboldal tervezési eszköz – UI/UX tervezéshez és prototípus készítéshez használt eszköz.
* GitHub: Verziókövetés és projekt tárolás – A kód verziókövetése és csapatmunka támogatása.
* Visual Studio 2022: Fejlesztőkörnyezet az ASP.NET Core backend és az Entity Framework Core kezelésére.
* Visual Studio Code: Könnyűsúlyú kódszerkesztő a frontend fejlesztéshez.

**Könyvtárak és keretrendszerek**

* React: Komponensalapú JavaScript könyvtár a frontend fejlesztéshez.
* Bootstrap / Tailwind CSS: Stílusok és reszponzív megjelenés biztosítására.
* Axios: HTTP kérések kezelésére a frontend és a backend között.
* Entity Framework Core: ORM rendszer az adatbázis kezeléséhez.

**Tesztelési és hibakeresési eszközök**

* Postman: REST API tesztelésére és hibakeresésére használt eszköz.
* Google Chrome DevTools: Böngészőbe épített fejlesztői eszközök a weboldal optimalizálásához.
* Jest és React Testing Library: Egység- és integrációs tesztelések elvégzésére.

**Adatbázis és szerverkezelés**

* MySQL Workbench: Grafikus felület az adatbázis-kezeléshez.
* Docker: Konténerizált környezet létrehozására és futtatására.
* Azure / AWS szolgáltatások: Az alkalmazás üzemeltetésére és skálázására.

Ezek a források és eszközök biztosították a projektünk gördülékeny fejlesztését és megvalósítását.