A képen embléma, szöveg, Betűtípus, Védjegy látható

Automatikusan generált leírás

ZÁRÓDOLGOZAT

Miskolc

2024.

Miskolci SZC Kandó Kálmán Informatikai Technikum

Szoftverfejlesztő és -tesztelő technikus szakképesítés

Záródolgozat

Ételrendelő Webáruház

Miskolc

2024.

Tartalom

[BEVEZETÉS: 3](#_Toc905658167)

[FELHASZNÁLT TECHNOLÓGIÁK: 4](#_Toc1037075237)

[HTML: 5](#_Toc1850359149)

[CSS: 5](#_Toc301144369)

[BOOTSTRAP: 6](#_Toc521793244)

[MYSQL: 7](#_Toc1649920105)

[REACT: 7](#_Toc664734223)

[ASP.NET CORE: 8](#_Toc721505738)

[ENTITY FRAMEWORK: 9](#_Toc1141962877)

[JWT: 9](#_Toc922208607)

[AXIOS: 10](#_Toc163816836)

[SZERKEZET: 10](#_Toc255185317)

[A weboldal szerkezete: 11](#_Toc1800679249)

[Adatbázis diagram 13](#_Toc617592910)

[FELHASZNÁLÓI DOKUMENTÁCIÓ: 13](#_Toc607649722)

[Főoldal: 14](#_Toc1450913220)

[Kapcsolat: 15](#_Toc886189744)

[Regisztráció: 16](#_Toc76835226)

[Bejelentkezés: 18](#_Toc571788872)

[Kijelentkezés: 20](#_Toc1659092190)

[Étlap: 21](#_Toc249320564)

[Kosár: 23](#_Toc330246416)

[-Vásárlás folytatása 23](#_Toc1736478257)

[-Törlés 24](#_Toc1284871146)

[Forrásjegyzék: 26](#_Toc1060034364)

# BEVEZETÉS:

Mi egy gyorsétel-rendelő webáruházat szerettünk volna megvalósítani. Azért gondoltunk erre, mivel manapság egyre kevesebb idő jut az embereknek a főzésre. A gyorséttermek és a kiszállítás hatalmas népszerűségnek örvendenek, különösen a pandémia miatt, amikor az emberek nagyon el lettek zárva a személyes kapcsolatoktól. A weboldalunk az ételek kiválasztása mellett, az ételek rendelését is tartalmazza, hogy a felhasználó minél kevesebb időt tudjon megspórolni. Hétköznapi embereknek, hétköznapi pénztárcával is meg tudják rendelni azt az ételt, amit esetleg máshol kiemelkedően magas áron akarják eladni a vásárlóknak. Ez egy jó alkalom arra, hogy betörjünk a webáruház piacára.

# FELHASZNÁLT TECHNOLÓGIÁK:

## HTML:

A HyperText Markup Language vagy HTML a szabványos jelölőnyelv a webböngészőben való megjelenítésre tervezett dokumentumokhoz. Meghatározza a webtartalom tartalmát és szerkezetét. Ezt gyakran segítik olyan technológiák, mint a Cascading Style Sheets(CSS) és a szkript nyelvek, például a JavaScript. A webböngészők HTML dokumentumokat fogadnak egy webszerverről vagy helyi tárolóról, és a dokumentumokat multimédiás weboldalakká alakítják. A HTML szemantikailag írja le a weboldal szerkezetét, és eredetileg a megjelenésére utaló jeleket tartalmazott. A HTML elemek a HTML oldalak építőkövei. A HTML-konstrukciók segítségével képek és egyéb objektumok, például interaktív űrlapok beágyazhatók a megjelenített oldalba. A HTML lehetőséget biztosít strukturált dokumentumok létrehozására a szöveg strukturális szemantikájának, például címsorok, bekezdések, listák, hivatkozások, idézetek és egyéb elemek jelölésével.



1.Ábra: HTML

## CSS:

A Cascading Style Sheets (CSS) egy stíluslapnyelv, amelyet egy jelölőnyelven, például HTML-ben vagy XML-ben (beleértve az XML-dialektusokat, például az SVG-t, MathML-t vagy XHTML-t) írt dokumentumok megjelenítésének és stílusának meghatározására használnak. A CSS a World Wide Web egyik sarokköve, a HTML és a JavaScript mellett. A CSS-t úgy tervezték, hogy lehetővé tegye a tartalom és a megjelenítés szétválasztását, beleértve az elrendezést, a színeket és a betűtípusokat. Ez a szétválasztás javíthatja a tartalom hozzáférhetőségét további magyarázat szükséges nagyobb rugalmasságot és ellenőrzést biztosít a megjelenítési jellemzők specifikációjában lehetővé teszi több weboldal formázásának megosztását a megfelelő CSS-nek egy külön .CSS fájlban történő megadásával, ami csökkenti a szerkezeti tartalom bonyolultságát és ismétlődését és engedélyezze a .CSS fájl gyorsítótárazását a fájlt megosztó oldalak közötti oldalbetöltési sebesség és a formázás javítása érdekében.



2.Ábra: CSS

## BOOTSTRAP:

A Bootstrap egy HTML, CSS és JS könyvtár, amely az informatív weboldalak fejlesztésének leegyszerűsítésére összpontosít (szemben a webalkalmazásokkal). A webes projektekhez való hozzáadásának elsődleges célja az, hogy a Bootstrap által választott színt, méretet, betűtípust és elrendezést alkalmazza az adott projektre. Mint ilyen, az elsődleges tényező az, hogy a felelős fejlesztők megtalálják-e a tetszését. Miután hozzáadta a projekthez, a Bootstrap alapvető stílusdefiníciókat biztosít az összes HTML-elemhez. Az eredmény a próza, a táblázatok és az űrlapelemek egységes megjelenése a böngészőkben. Ezenkívül a fejlesztők kihasználhatják a Bootstrapban definiált CSS-osztályokat, hogy tovább testreszabják tartalmuk megjelenését. A Bootstrap például világos és sötét színű táblázatokat, oldalcímeket, szembetűnőbb idézőjeleket és kiemeléssel ellátott szöveget biztosít.



3.Ábra: Bootstrap

## MYSQL:

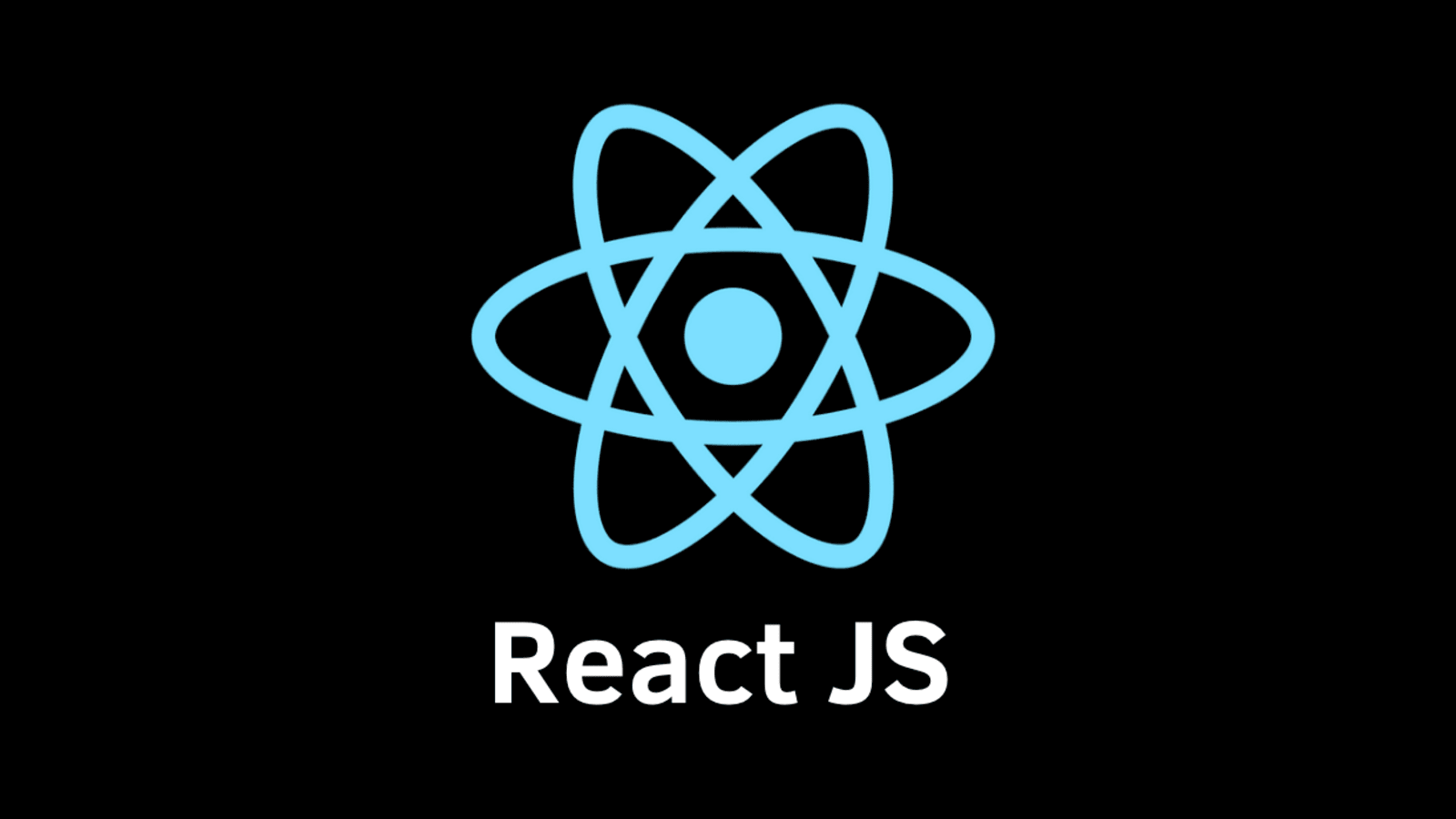
A MySQL egy nyílt forráskódú relációs adatbázis-kezelő rendszer (RDBMS). A neve a "My", Michael Widenius társalapító lánya, My nevének és az "SQL", a Structured Query Language mozaikszó kombinációja. A relációs adatbázis egy vagy több adattáblába rendezi az adatokat, amelyekben az adatok összefügghetnek egymással; ezek a kapcsolatok segítik az adatok strukturálását. Az SQL egy olyan nyelv, amelyet a programozók a relációs adatbázisból adatok létrehozására, módosítására és kinyerésére, valamint az adatbázishoz való felhasználói hozzáférés szabályozására használnak. A relációs adatbázisokon és az SQL-en kívül az RDBMS, például a MySQL, együttműködik egy operációs rendszerrel, hogy relációs adatbázist valósítson meg a számítógép tárolórendszerében, kezeli a felhasználókat, lehetővé teszi a hálózati hozzáférést, valamint megkönnyíti az adatbázis integritásának tesztelését és a biztonsági mentések létrehozását.



4.Ábra: MySQL

## REACT:

A React (más néven React.js vagy ReactJS) egy ingyenes és nyílt forráskódú előtérbeli JavaScript-könyvtár komponenseken alapuló felhasználói felületek létrehozására. Karbantartója a Meta (korábban Facebook), valamint az egyes fejlesztőkből és cégekből álló közösség. A React használható egyoldalas, mobil vagy szerver által megjelenített alkalmazások fejlesztésére olyan keretrendszerekkel, mint a Next.js. Mivel a React csak a felhasználói felülettel és a DOM -hoz való megjelenítési összetevőkkel foglalkozik, a React alkalmazások gyakran támaszkodnak könyvtárakra az útválasztáshoz és egyéb kliensoldali funkciókhoz. A React fő előnye, hogy csak az oldal megváltozott részeit jeleníti meg újra, elkerülve a változatlan DOM-elemek szükségtelen újrarenderelését.



5.Ábra: React Js

## ASP.NET CORE:

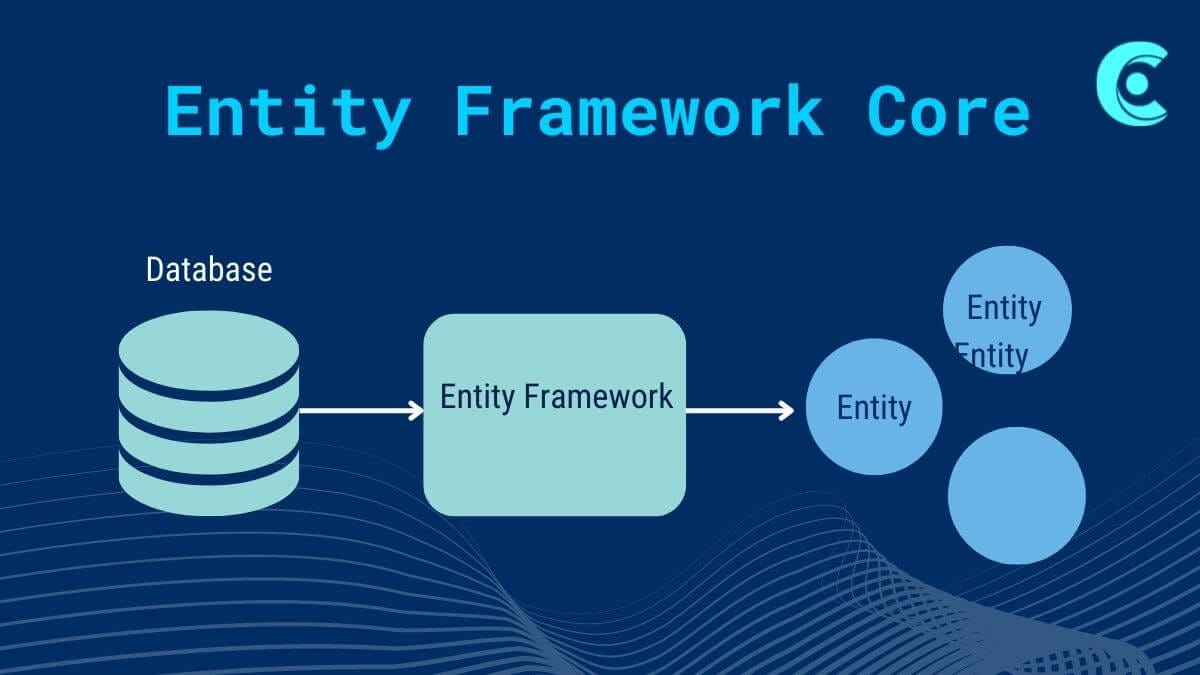
Az ASP.NET Core egy nyílt forráskódú moduláris webalkalmazás-keretrendszer. Ez az ASP.NET újratervezése, amely a korábban különálló ASP.NET MVC-t és az ASP.NET Web API-t egyetlen programozási modellben egyesíti. Annak ellenére, hogy új keretrendszerről van szó, amely egy új webveremre épül, nagyfokú koncepció-kompatibilitást biztosít az ASP.NET-tel. Az ASP.NET Core keretrendszer támogatja az egymás melletti verziókezelést, így az egyetlen gépen fejlesztett különböző alkalmazások az ASP.NET Core különböző verzióit célozhatják meg. Ez nem volt lehetséges az ASP.NET korábbi verzióival. Az ASP.NET Core kezdetben a csak Windows .NET-keretrendszeren és a többplatformos .NET-en is futott. A .NET-keretrendszer támogatása azonban az ASP.Net Core 3.0-tól kezdve megszűnt.



6.Ábra: ASP

## ENTITY FRAMEWORK:

Az Entity Framework (EF) egy nyílt forráskódú objektum-relációs leképezési (ORM) keretrendszer az ADO.NET számára. Eredetileg a .NET-keretrendszer szerves részeként szállították, azonban az Entity Framework 6.0-s verziójától kezdve a .NET-keretrendszertől külön szállították.Az Entity Framework 6.4 volt a klasszikus keretrendszer legújabb kiadása. Bár az Entity Framework 6 továbbra is támogatott, már nem fejlesztik, és csak a biztonsági problémákra kap javításokat. Az Entity Framework Core (EF Core) néven ismert új keretrendszert 2016-ban vezették be hasonló, de nem teljes funkcióparitással. A keretrendszer verziószámozása 1.0-ról indult újra, az EF Core legújabb verziója pedig a 8.0.



7.Ábra: Entity Framework

## JWT:

A JSON Web Token (JWT, javasolt kiejtés, megegyezik a "jot" szóval) egy javasolt internetes szabvány opcionális aláírással és/vagy opcionális titkosítással rendelkező adatok létrehozására, amelyek hasznos adattartalma tartalmazza a JSON-t, amely bizonyos számú állítja. A tokeneket privát titok vagy nyilvános/privát kulccsal írják alá.Például egy kiszolgáló létrehozhat egy tokent, amely a „rendszergazdaként bejelentkezett” követelménnyel rendelkezik, és ezt megadhatja egy ügyfélnek. Az ügyfél ezután ezt a tokent használhatja annak bizonyítására, hogy rendszergazdaként van bejelentkezve. A tokeneket aláírhatja az egyik fél privát kulcsa (általában a szerveré), így bármelyik fél később ellenőrizheti, hogy a token jogos-e. Ha a másik fél, valamilyen megfelelő és megbízható módon, birtokában van a megfelelő nyilvános kulcsnak, akkor ő is képes ellenőrizni a token legitimitását.

8.Ábra: JWT

## AXIOS:

Az Axios egy ígéret-alapú HTTP kliens a node.Js-hez és a böngészőhöz. Izomorf (= ugyanazzal a kódbázissal fut a böngészőben és a node.js-ben). A szerveroldalon a natív node.js http modult használja, míg a kliensoldalon (böngésző) az XMLHttpRequesteket használja.



9.Ábra: Axios

# SZERKEZET:

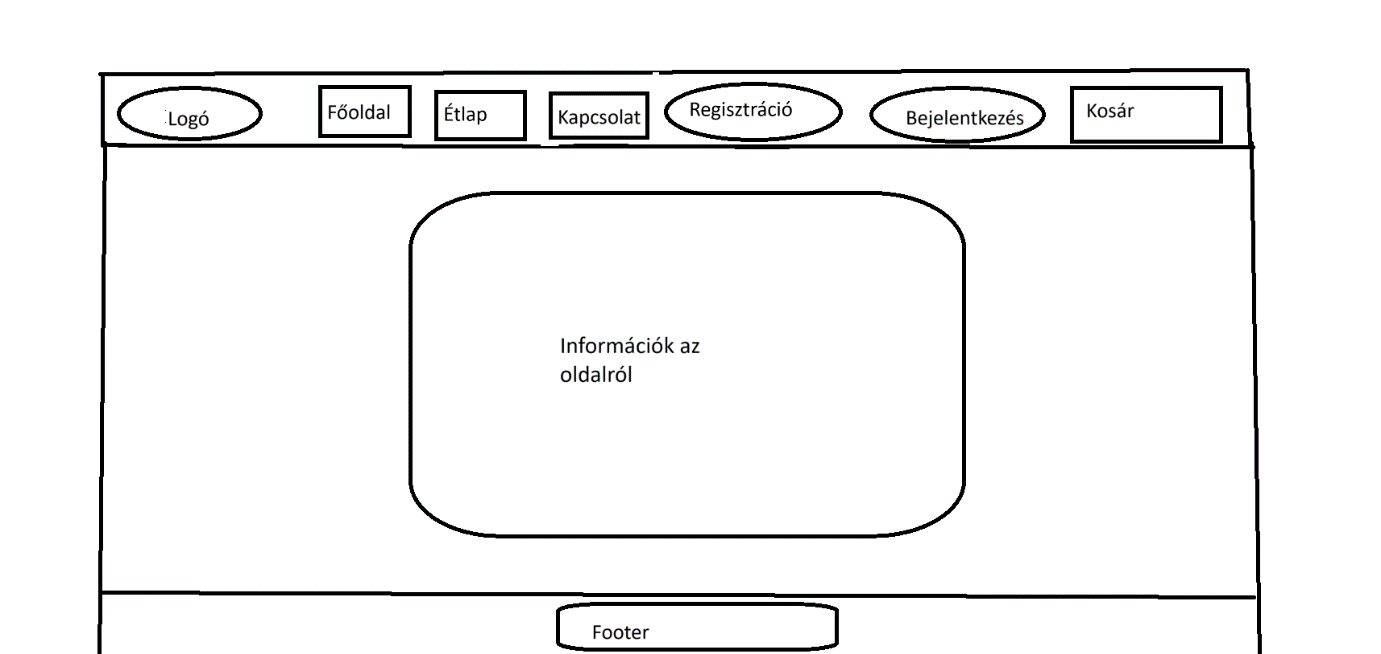
A szakdolgozat szerkezete három részből épül fel:

1.React (Frontend)

2.ASP.NET CORE WEB API (Backend)

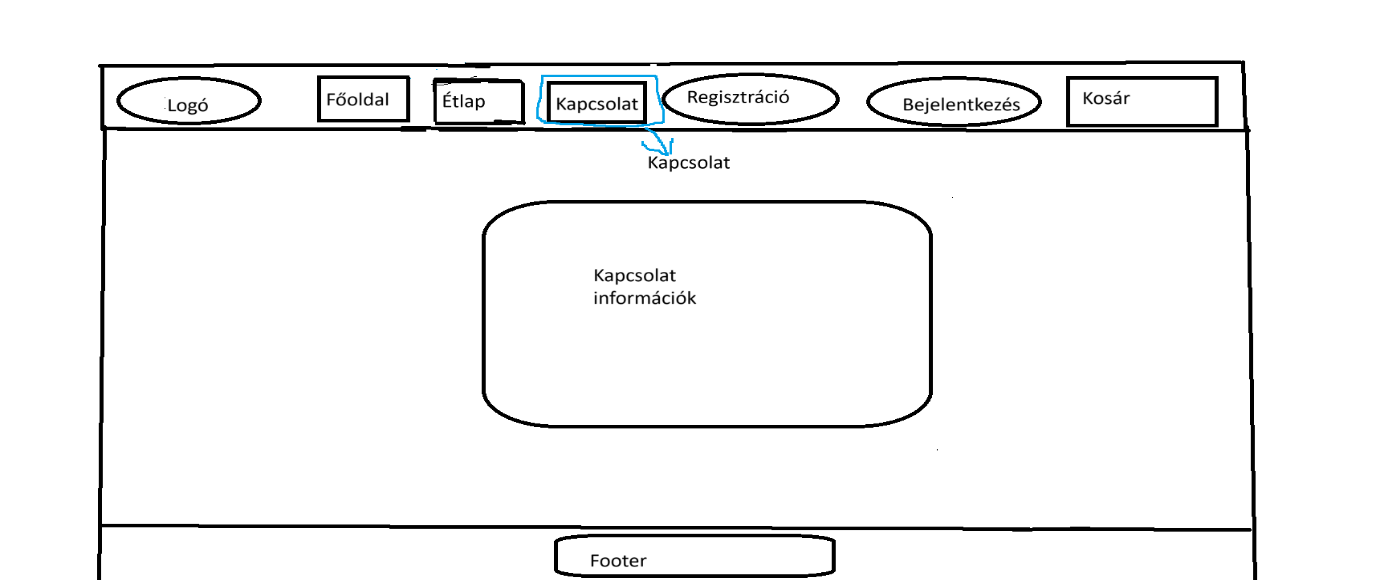
3.MySql (Adatbázis)

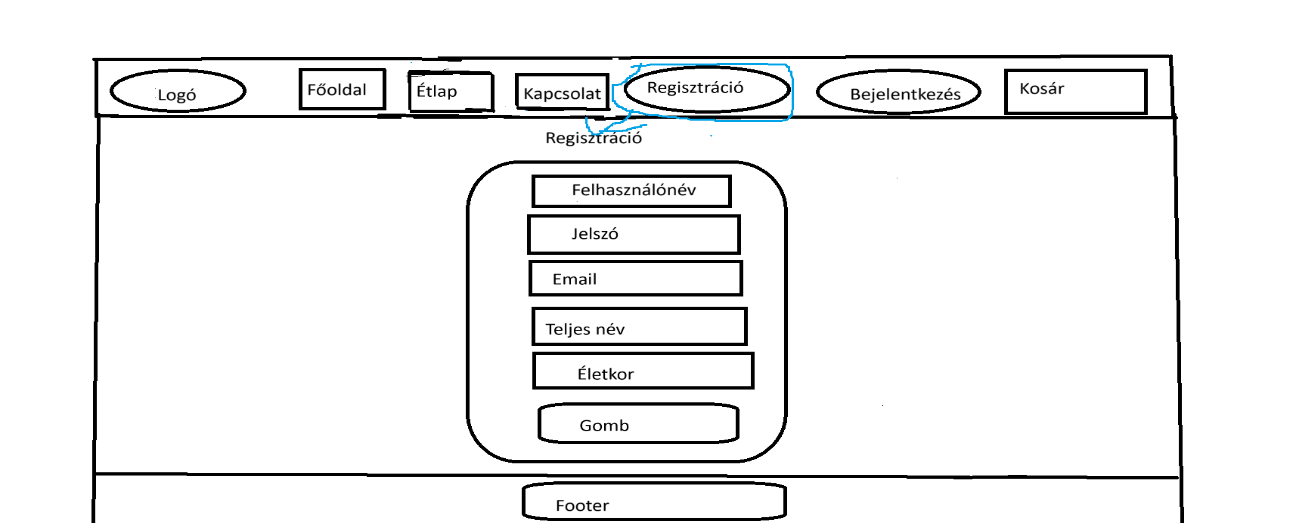
## A weboldal szerkezete:

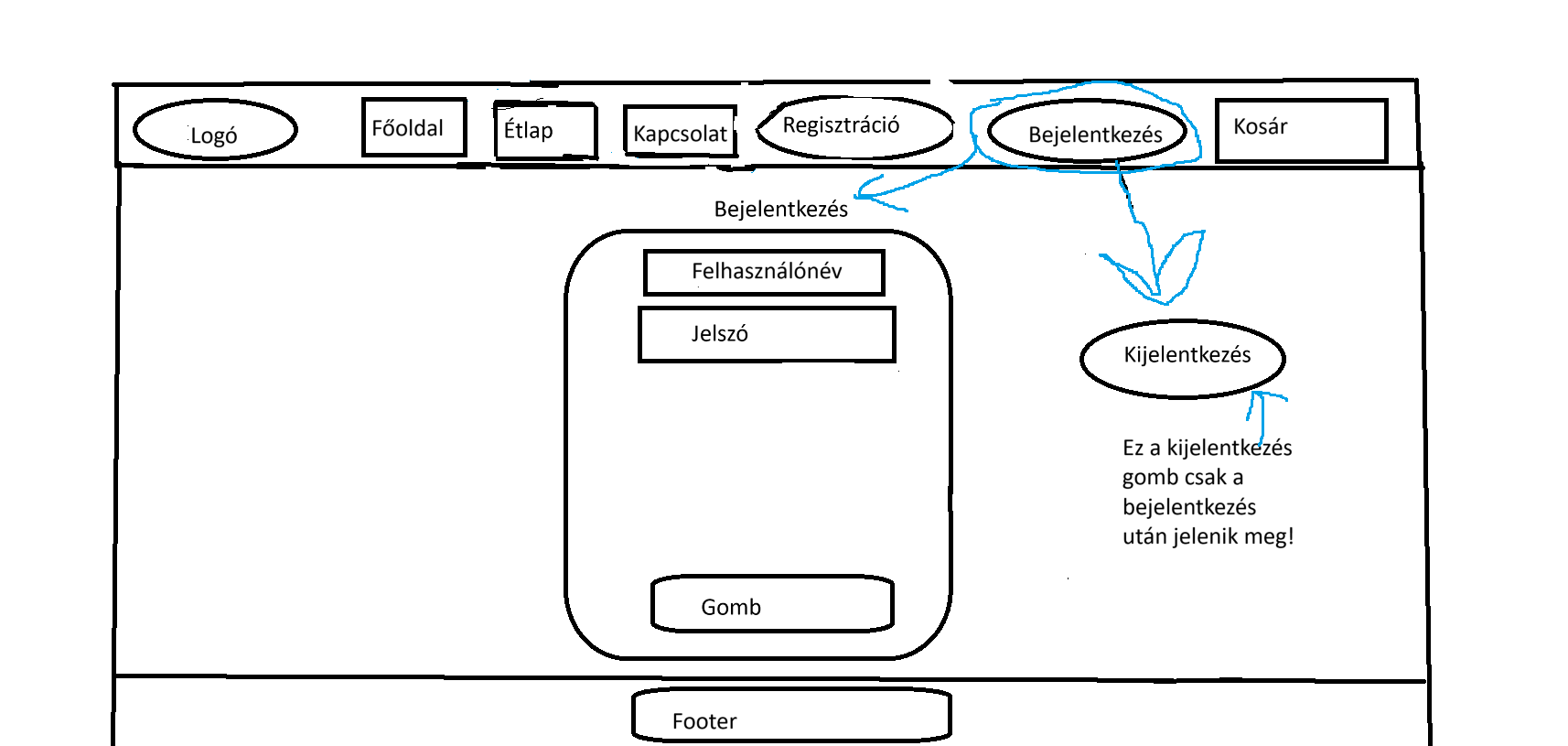
 10.Ábra: Főoldal

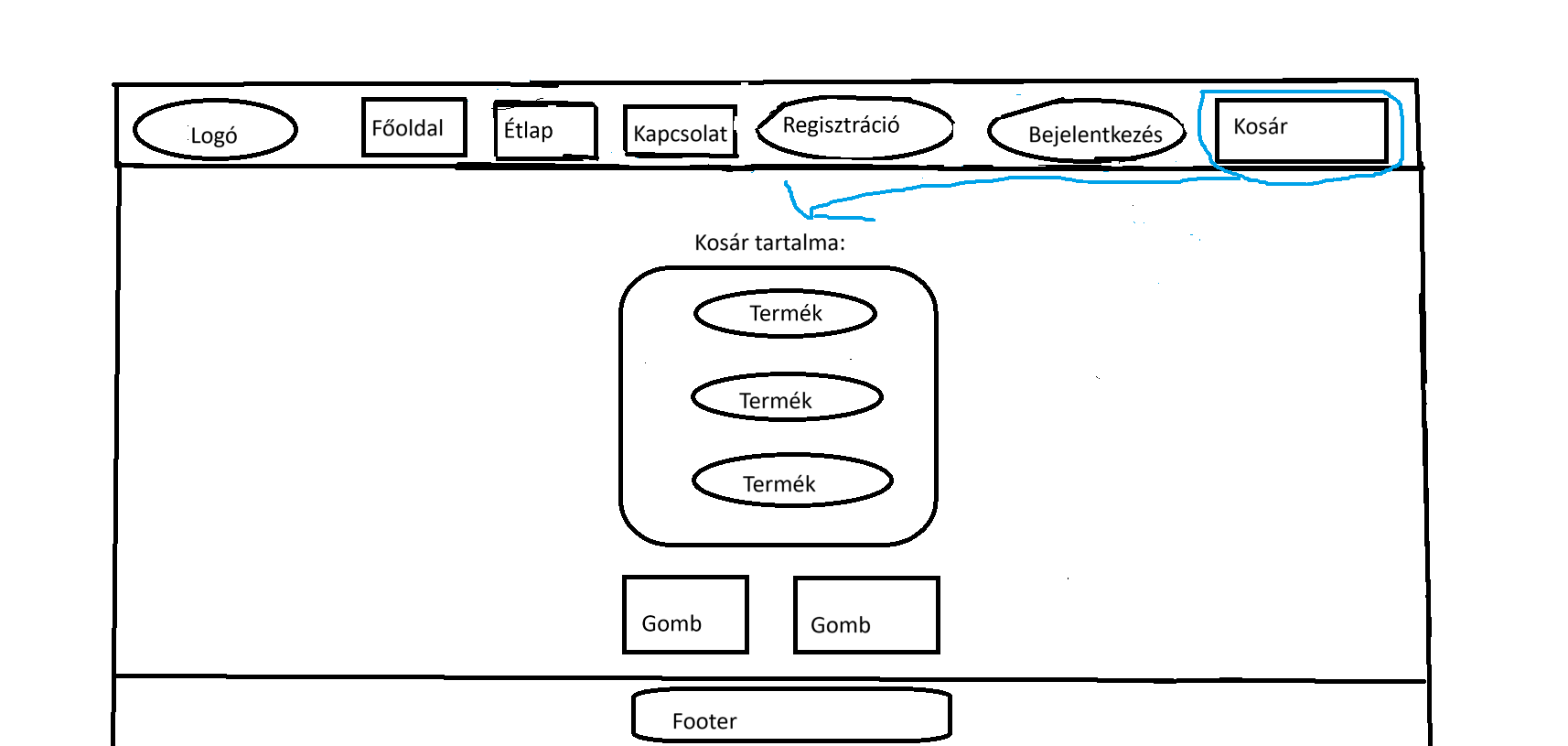
A képen képernyőkép, szöveg, diagram, Téglalap látható

Automatikusan generált leírás11.Ábra: Étlap

12.Ábra: Kapcsolat

13.Ábra: Regisztrációs felület

14.Ábra: Bejelentkezési felület

15.Ábra: Kosár

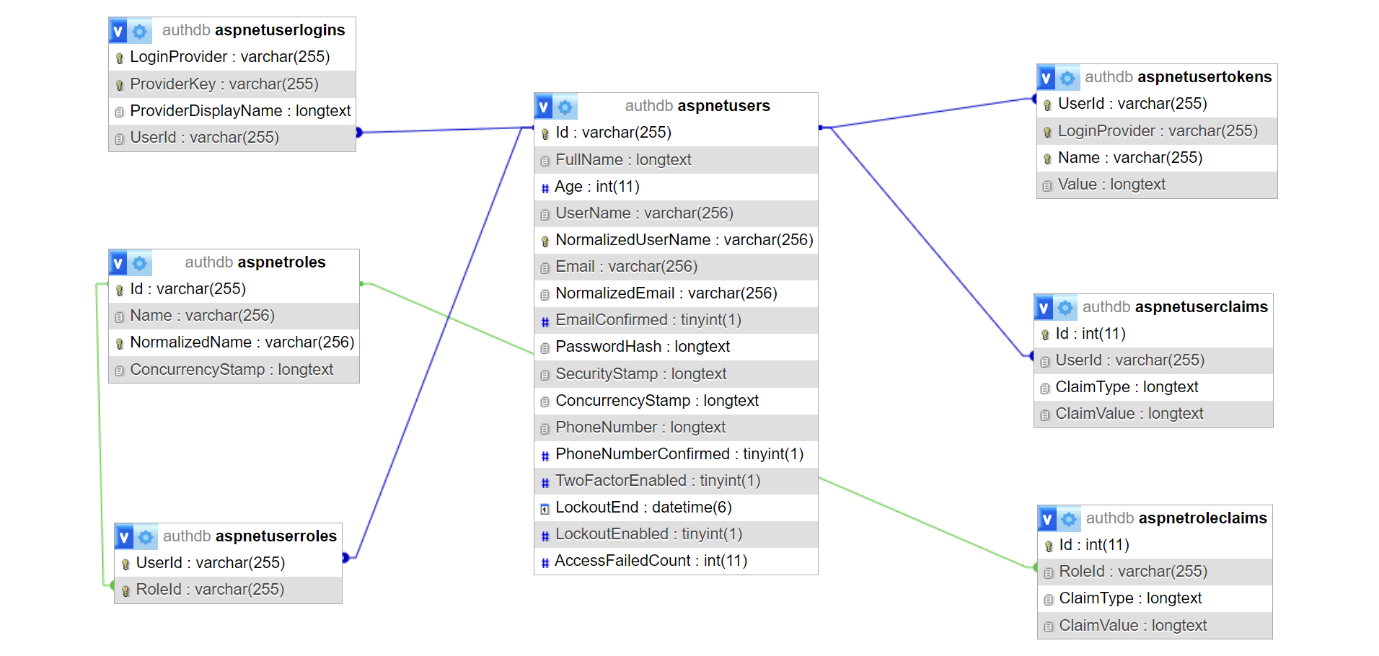
Ezen minták alapján készítettük el webáruházunkat React, illetve ASP.NET

CORE Web API segítségével.

## Adatbázis diagram

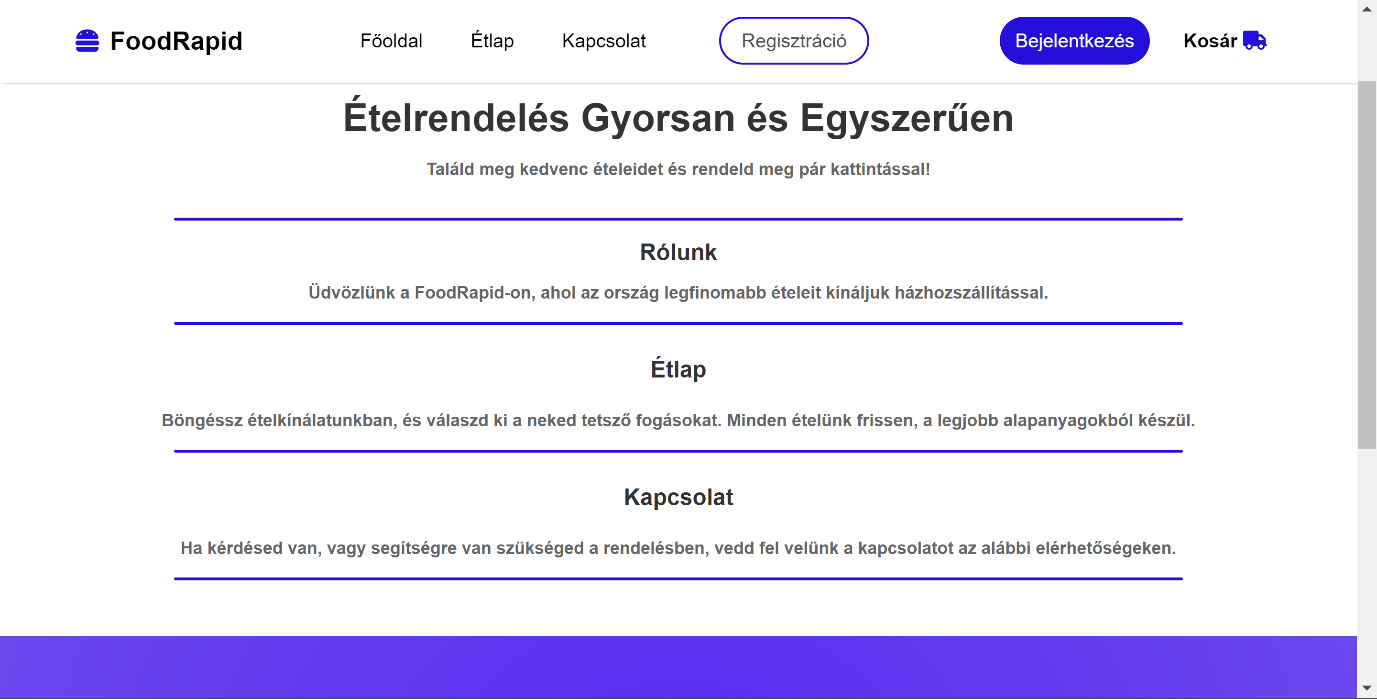
Az adatbázisunkat MySql-ben hoztuk létre, amit a XAMPP-on keresztül lehet futtatni. Az ezzel történő kapcsolatot az Entity Framework Core biztosította.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás16.Ábra: Adatbázis diagram

# FELHASZNÁLÓI DOKUMENTÁCIÓ:

A szerver elindítása után, ami közben egy alapos ellenőrzési folyamat zajlik le annak érdekében, hogy a rendszer megfelelően töltődik-e be, megjelenik a weboldal, aminek a “FoodRapid” nevet választottuk.

17.Ábra: Főoldal

## Főoldal:

A főoldal dizájnját igyekeztünk letisztult és célratörően megformázni, hogy a látogató pontos és tömör információkat kapjon a webáruházunkról, és hogy még csak véletlenül se vesszen el a részletekben. Emellett arra is gondoltunk hogy a felhasználó idejét megspórolva úgy készítjük el a főoldalt hogy a megfelelő menüpontra kattintva ellehessen jutni a cél oldalra, mindössze egy kattintásnyi idő alatt.

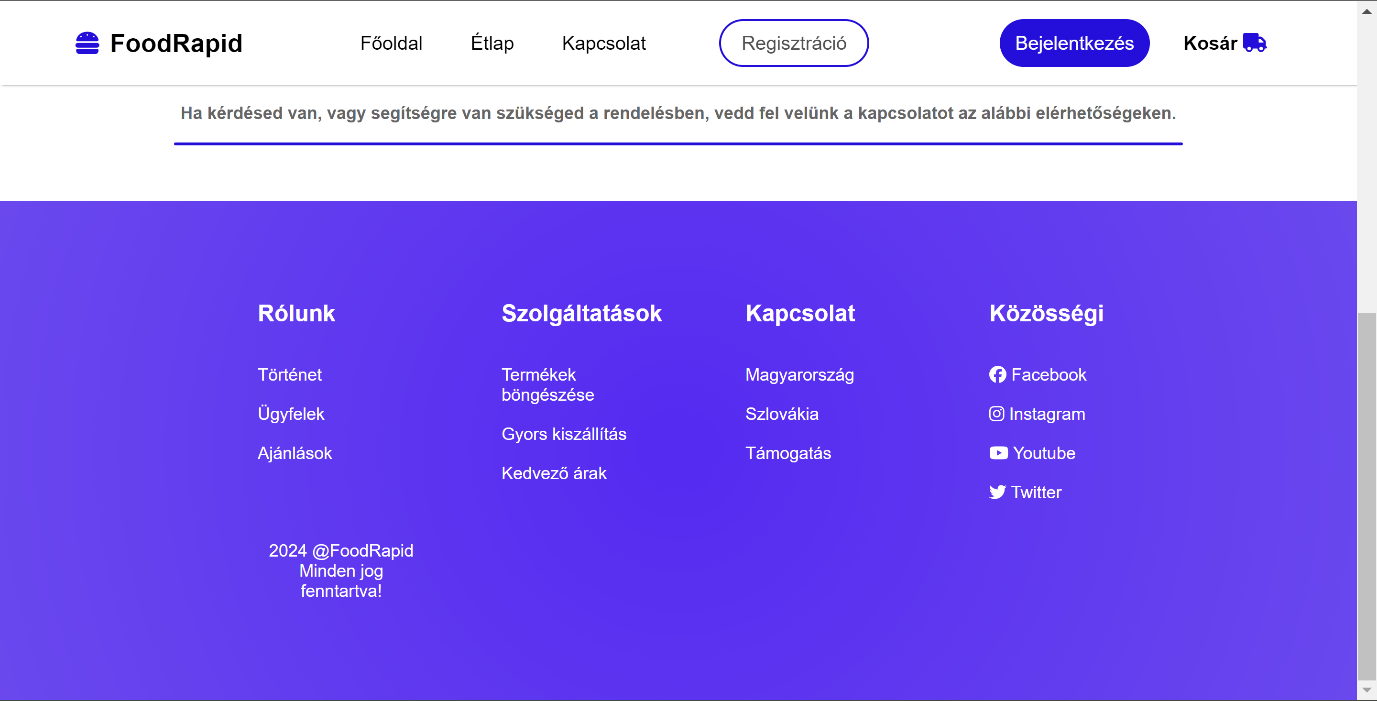


18.Ábra: Főoldal Főoldal/Navigáció

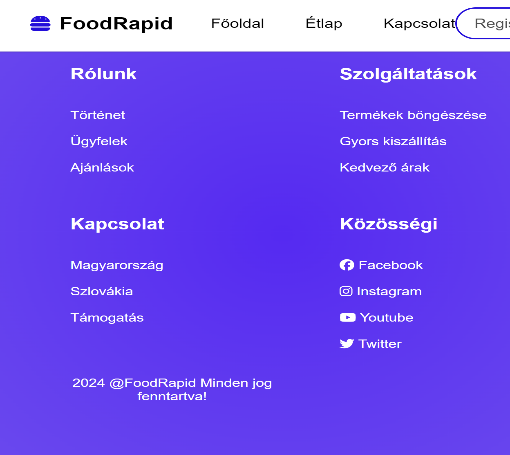
Mi esetünkben az Étlap és a Kapcsolat menüpontra esett a választás, amikor a felhasználó ezekre kattint, akkor a rendszer automatikusan a kiválasztott oldalra viszi a felhasználót.

Footer:

Mielőtt kitérnénk a talán fontosabb részekre is, előtte még szerettünk volna egy kis betekintést nyújtani a rendkívül dizájnosra sikerült footer-részre is.

19.Ábra: Footer

A footert szintén a letisztultság jellemzi, ami tartalmaz a webáruházunkról még egyéb adatokat, emellett alkalmazkodó is, mivel igyekeztünk reszponzívvá alakítani. Mint ahogyan a következő ábrán is látható:



20.Ábra: Footer/reszponzív

Természetesen minden oldalon alkalmazkodó a footer, akárcsak maguk az oldalak.

## Kapcsolat:

A webáruházunk tartalmaz egy elérhetőségek menüpontot is, amire szerintünk minden valamire való webáruháznak szüksége van a felhasználókkal való minél könnyebb kommunikáció érdekében.

21.Ábra: Kapcsolat

## Regisztráció:

A regisztráció során a felhasználó következő adatokat adhatja meg:

-Felhasználó nevét

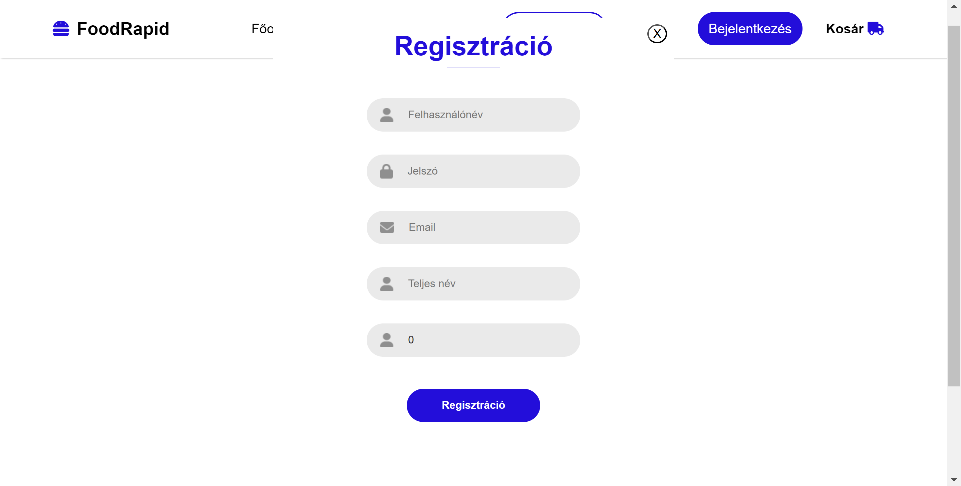
-Jelszavát

-Email-címét

- Teljes nevét

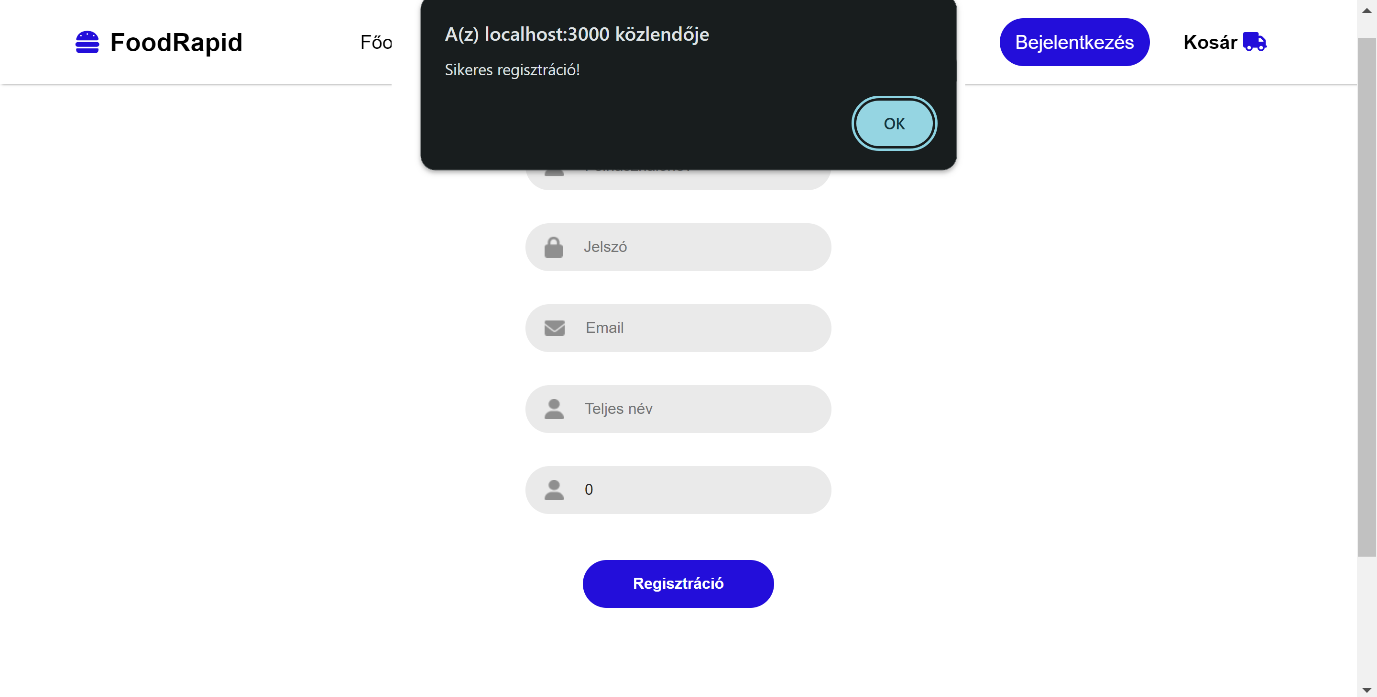
-Életkorát

Ha az adatokat az elvártnak megfelelően adja meg a felhasználó, akkor utána a Regisztráció gombra kattintva megtörténik a felhasználó adatainak elmentése, amikkel később be is jelentkezhet.



22.Ábra: Regisztráció

Ha regisztráció elkészült az elvártaknak megfelelően, akkor utána megjelenik egy üzenet ami a regisztráció sikerességéről tájékoztat.

23.Ábra: Sikeres regisztráció

Az adatok a megfelelő adatbázisba kerültek, most már a felhasználó bármikor bejelentkezhet.

Bővebben a regisztrációs folyamatokról:

A regisztrációs folyamat a következő lépéseket tartalmazza:

1.Adatbeviteli Űrlap: A frontend egy űrlapot jelenít meg, ahol a felhasználók megadhatják a szükséges adatokat.

2.Adatok Validálása: Az űrlap mezői validálják az adatokat, hogy megfeleljenek a formátumnak és a biztonsági követelményeknek.

3.Axios POST Kérés: Ha az adatok helyesek, egy Axios POST kérés indítódik a backend felé az adatokkal.

4. Backend Feldolgozás: A backend fogadja a kérést, és ellenőrzi az adatokat, például az email-cím egyediségét.

5.Adatbázisba Írás: Ha minden ellenőrzés sikeres, a backend az adatokat egy adatbázisba írja.

6. Visszajelzés a Felhasználónak: Sikeres regisztráció esetén a backend egy sikeres válaszkódot küld vissza.

7.Felhasználói Felület Frissítése: A frontend frissíti a felületet, jelezve a sikeres regisztrációt.

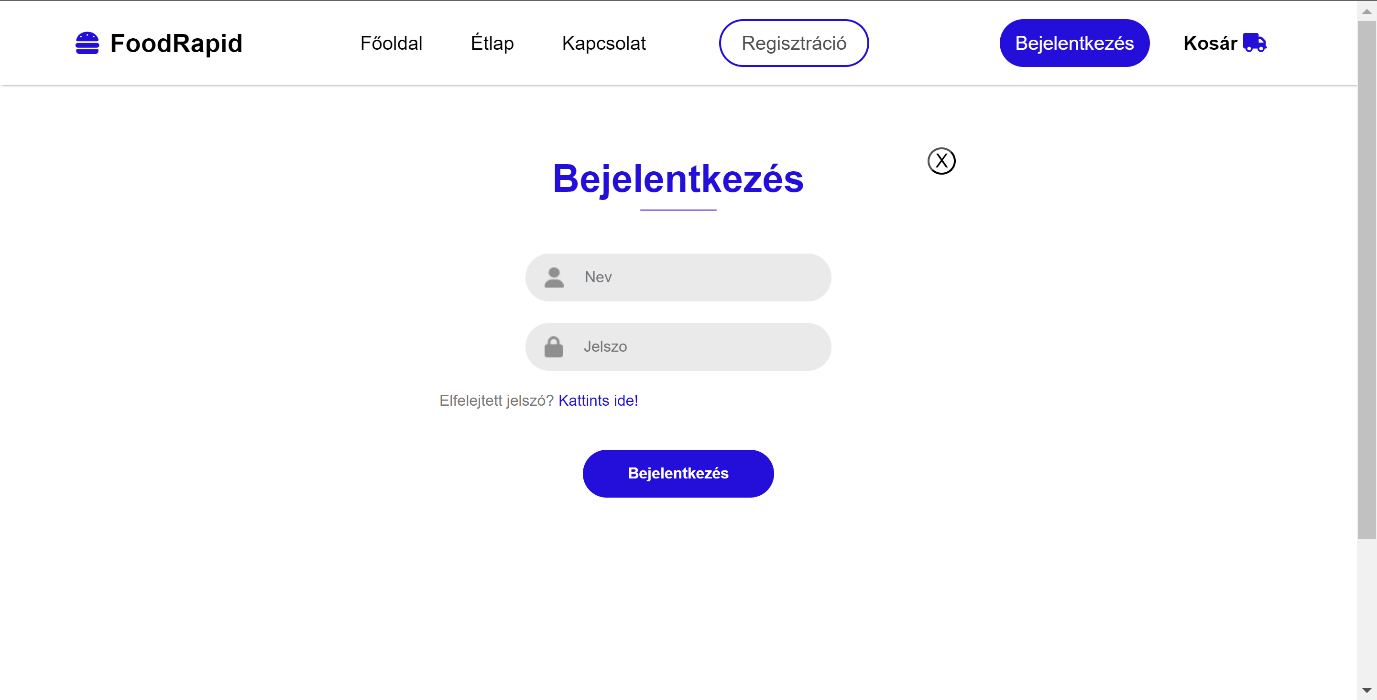
A regisztrációs folyamat során a jelszavak titkosítva kerülnek tárolásra, hogy biztosítsuk a felhasználói adatok biztonságát. Az Axios könyvtár segítségével a frontend és a backend közötti kommunikáció zökkenőmentes és biztonságos.

## Bejelentkezés:

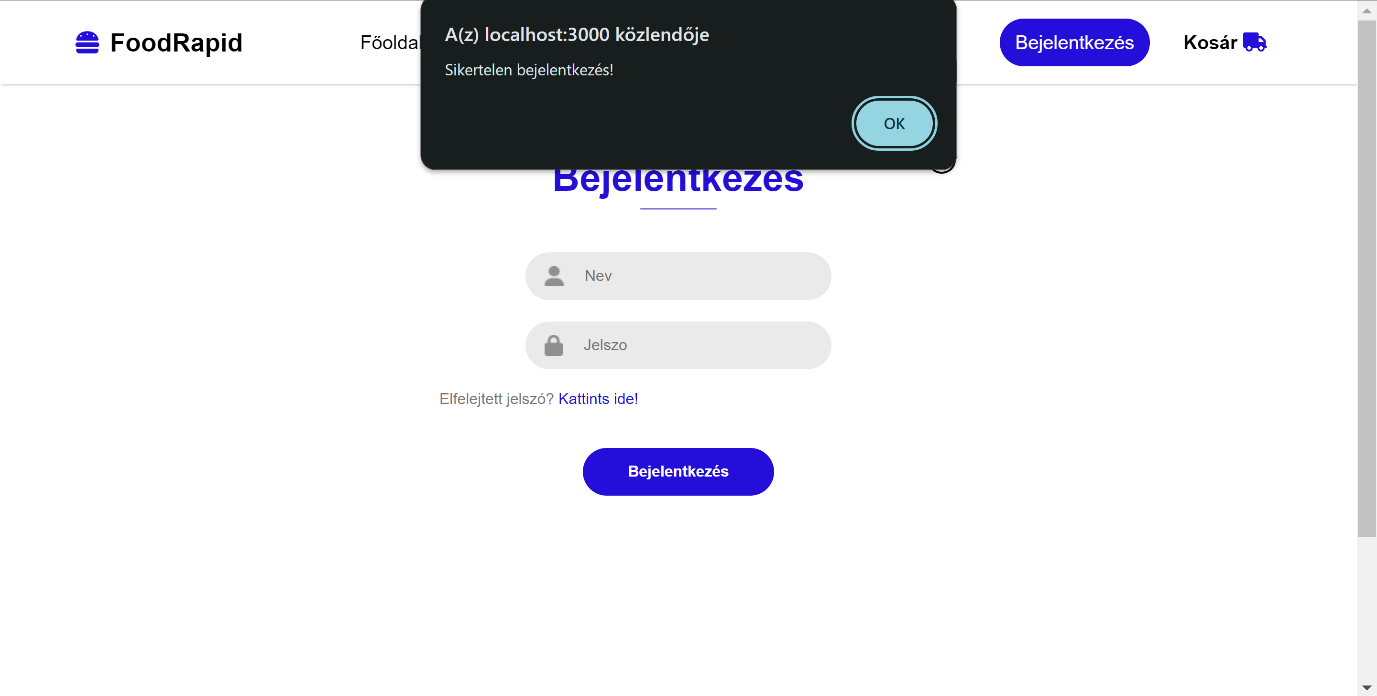
A bejelentkezéshez két adatra lesz szükség a regisztrált felhasználónak:

* Felhasználónév
* Jelszó

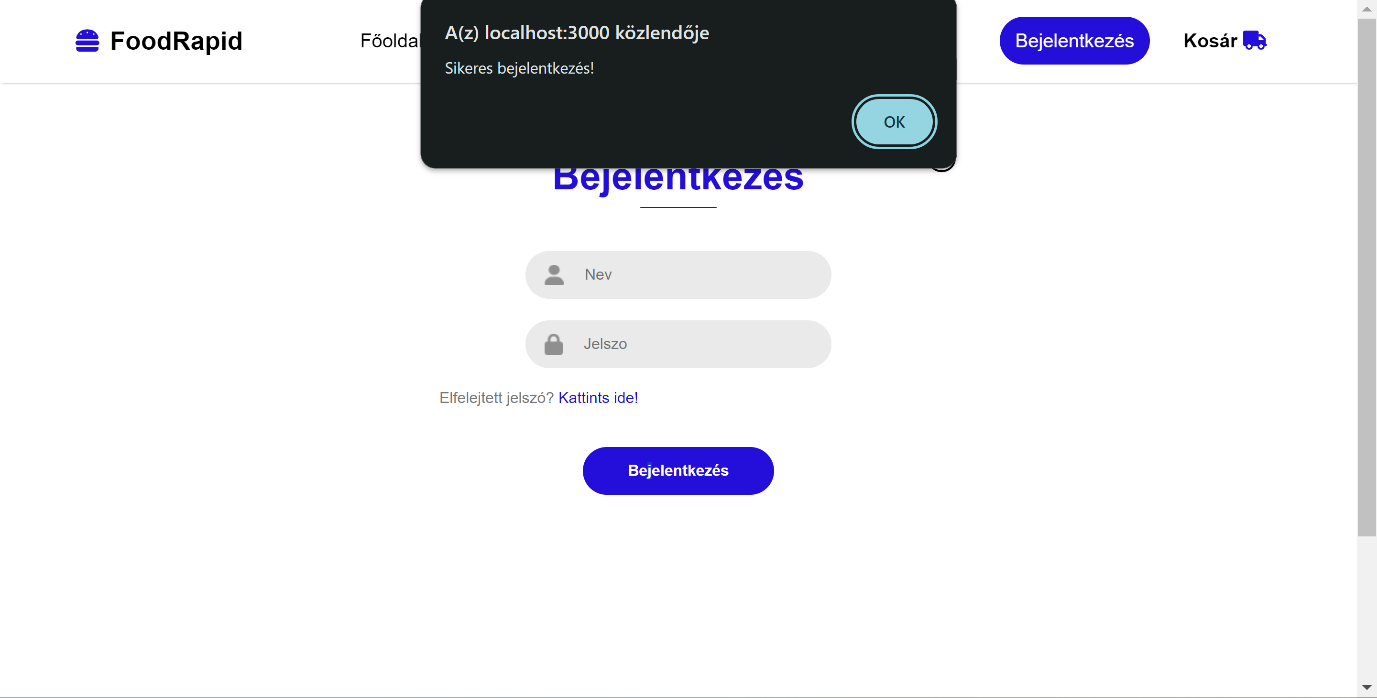
Ezeket a bejelentkezés gombra kattintás után megjelenő Login ablakban adhatja meg.

24.Ábra: Bejelentkezés

A bejelentkezést már csak a sikeresen regisztrált adatokkal rendelkező felhasználó teheti meg, máskülönben a bejelentkezés sikertelen lesz, amiről egy felugró ablak is figyelmeztetni fog.

25.Ábra: Sikertelen bejelentkezés

Ha az adatok a megfelelően lettek megadva akkor egy “Sikeres Bejelentkezés!” ablak fog megjelenni.

26.Ábra: Sikeres bejelentkezés

Bővebben a bejelentkezésről:

1. Bejelentkezési Űrlap: A felhasználói felületen egy űrlap kéri a felhasználó nevét és jelszavát.

2. Adatok Validálása: Az űrlap mezői ellenőrzik, hogy a megadott adatok megfelelnek-e a formátumnak.

3. Axios POST Kérés: Egy Axios POST kérés indítódik a backend felé a bejelentkezési adatokkal.

4. Backend Feldolgozás: A backend fogadja a kérést és ellenőrzi a felhasználói adatokat.

5. Hitelesítés: A backend összehasonlítja a megadott jelszót a titkosított jelszóval az adatbázisban.

6. Sikeres Bejelentkezés: Ha a hitelesítés sikeres, a backend egy token-t generál.

7.Token Küldése: A backend a token-t visszaküldi a frontendnek.

8. Felhasználói Állapot Frissítése: A frontend frissíti a felhasználói állapotot a token segítségével.

9. Navigáció: A felhasználó átirányítódik a kezdőlapra vagy a megadott visszatérési útvonalra.

A bejelentkezési folyamat során a következő biztonsági intézkedések kerülnek alkalmazásra:

1. Jelszó Titkosítás: A jelszavak titkosítva kerülnek tárolásra az adatbázisban.

2. Token Alapú Hitelesítés: A token-ek segítségével biztosítjuk a felhasználói munkamenetek biztonságát.

3. HTTPS Protokoll: Az adatátvitel biztonságos HTTPS kapcsolaton keresztül történik.

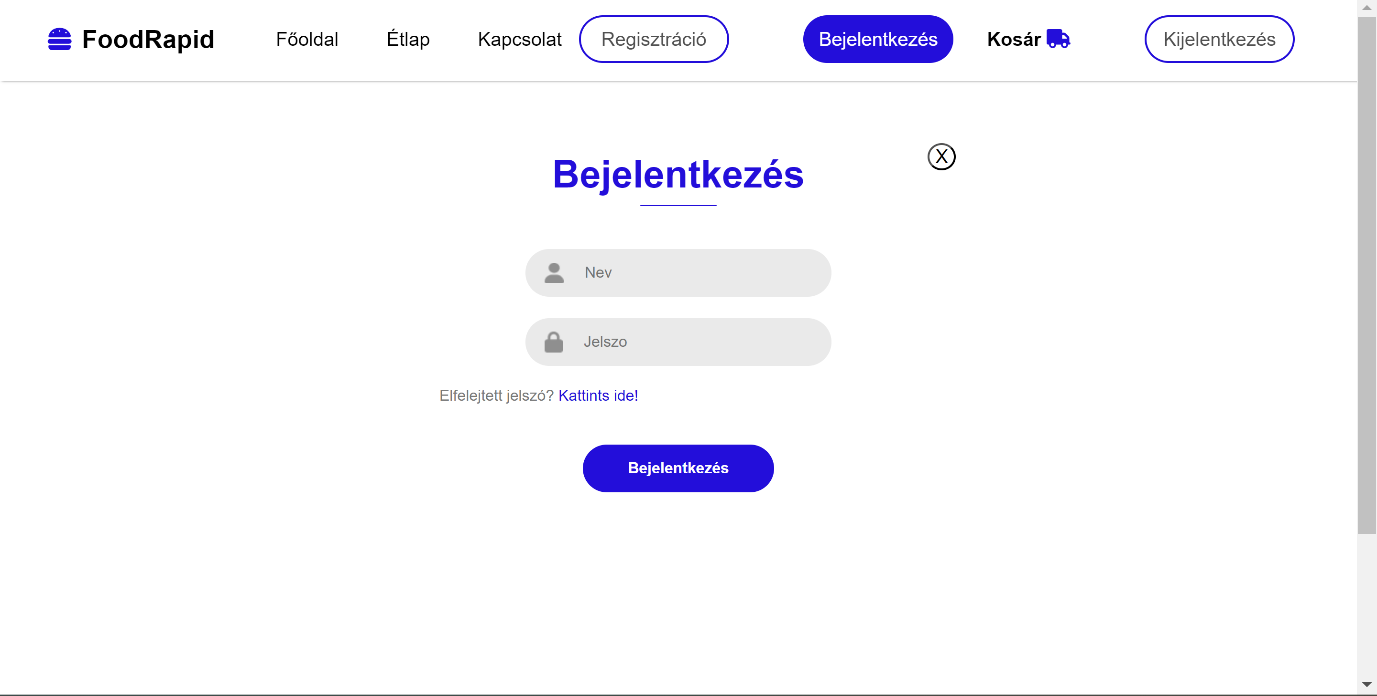
4. Csúszó Munkamenetek: A token-ek lejárati ideje biztosítja, hogy a munkamenetek ne maradjanak nyitva túl hosszú ideig.

5. Hibakezelés: A frontend megfelelő hibaüzeneteket jelenít meg, ha a bejelentkezés nem sikerül.

A bejelentkezési folyamat célja, hogy biztonságos és felhasználóbarát módon történjen. Az Axios könyvtár segítségével a frontend és a backend közötti kommunikáció zökkenőmentes. A token alapú hitelesítés modern és biztonságos módszer a felhasználói munkamenetek kezelésére.

## Kijelentkezés:

Bejelentkezés után a szerver megfelelő felhasználónév és jelszó alapján automatikusan generál egy tokent, ami lényegében egy kulcs, ami által bejelentkeztet az oldalra. Készítettünk egy kijelentkezés gombot is, amire kattintás után automatikusan kijelentkezteti a felhasználót.

27.Ábra: Kijelentkezés

Amint az látható, a bejelentkezés gombra kattintás után, és egyben sikeres bejelentkezés után kis időre rá megjelenik a jobb felső sarokban található kijelentkezés gomb is, ami ezelőtt még nem volt feltüntetve. A kijelentkezés működése egyszerű, a bejelentkezéshez szükséges adatokat (felhasználónév, token) kitörli, és ezáltal a felhasználót kijelentkezteti.

Bővebben a kijelentkezésről:

A kijelentkezés funkció biztosítja, hogy a felhasználók biztonságosan hagyják el a webáruházat. A folyamat a következő lépéseket tartalmazza:

1.Token Generálás: Bejelentkezés után a szerver egy tokent generál, ami azonosítja a felhasználót.

2. Kijelentkezés Gomb: A sikeres bejelentkezés után a felhasználói felületen megjelenik a kijelentkezés gomb.

3.Gomb Megjelenése: A gomb a jobb felső sarokban lesz látható, ami előtte rejtett volt.

4. Eseménykezelő: A gombra kattintva egy eseménykezelő aktiválódik a frontendben.

5. Axios DELETE Kérés: Az eseménykezelő egy Axios DELETE kérést indít a backend felé.

6.Token Érvénytelenítése: A backend érvényteleníti a tokent, megszakítva a munkamenetet.

7.Felhasználói Állapot Frissítése: A frontend frissíti a felhasználói állapotot, eltávolítva a hitelesítési adatokat.

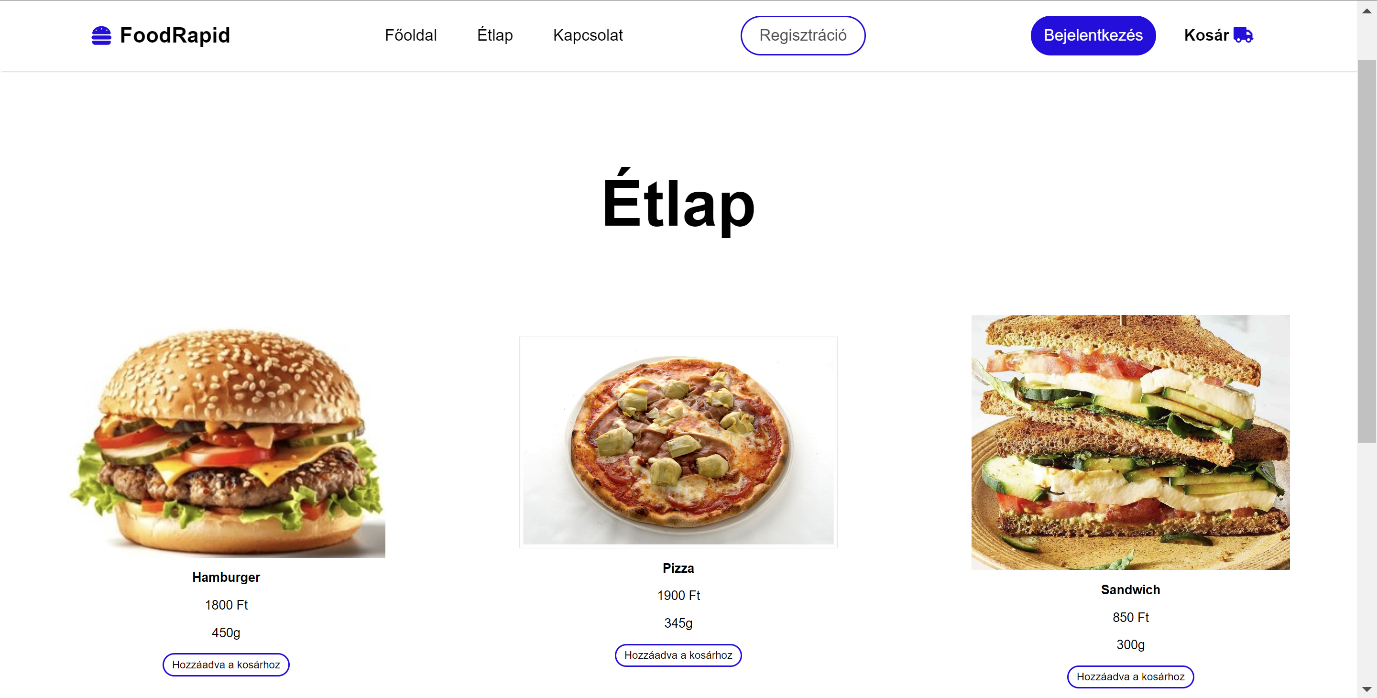
8. Navigáció: A felhasználó átirányítódik a bejelentkezési oldalra vagy a kezdőlapra.

9. Biztonság: A művelet biztonságos HTTPS kapcsolaton keresztül történik.

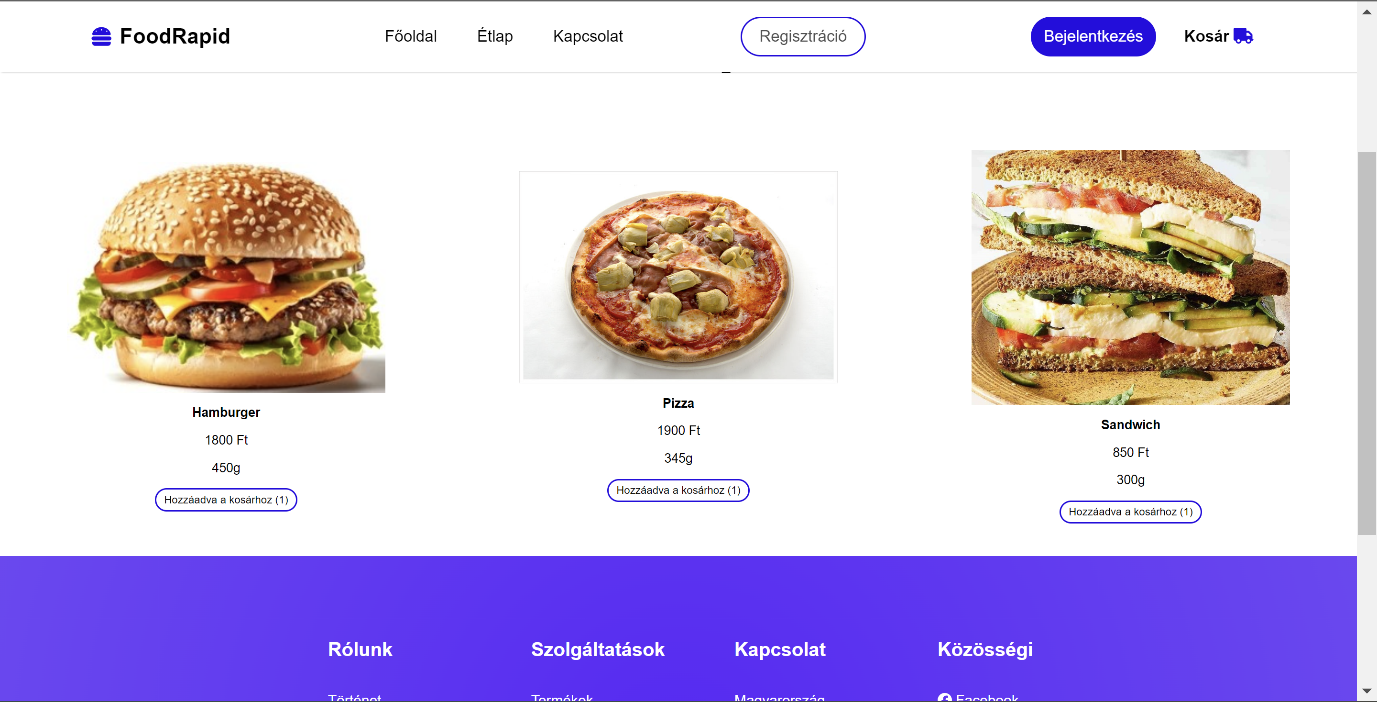
A kijelentkezés funkció működése egyszerű, de létfontosságú a webáruház biztonságának fenntartásához. A tokenek használata modern és biztonságos módszer a felhasználói munkamenetek kezelésére. A kijelentkezés gombra kattintva a rendszer automatikusan kijelentkezteti a felhasználót, kitörölve a bejelentkezéshez szükséges adatokat (felhasználónév, token), és ezzel biztosítva, hogy a felhasználói munkamenet ne maradjon nyitva a kiszolgálón. Ez a folyamat segít megvédeni a felhasználói fiókokat az illetéktelen hozzáféréstől és a biztonsági résektől.

## Étlap:

Az étlap menüpontra kattintás után megjelenik az étel választék, ami tartalmazza az ételek fényképét, nevét, árát és súlyát. Az ételek adatait, és még a hozzá tartozó fényképeket is a szerver tölti be az adatbázisban található adatokból.

28.Ábra: Étlap

Ezek mellett minden termékhez tartozik egy “Hozzá adva a kosárhoz” nevű gomb is, amire ha rákattint a felhasználó, akkor az adott termék automatikusan kiválasztásra kerül, és hozzá adódik a kosárhoz.

29.Ábra: Étlap

Mint ahogyan a képen is látható gombokban található egy számláló, ami a kosárhoz hozzá adott termékek számát mutatja. Természetesen a kosárhoz hozzá adott termékek darabszámának növelésével azonosan növekszik a megjelenő szám is.

Bővebben a "Hozzáadva a Kosárhoz" gomb működéséről:

A "Hozzáadva a kosárhoz" gomb egy kulcsfontosságú funkció, amely lehetővé teszi a vásárlók számára, hogy ételrendeléseiket gyorsan és hatékonyan adják hozzá a kosárukhoz. A folyamat a következő lépéseket tartalmazza:

1.Gomb Megjelenítése: Minden étel mellett megjelenik egy "Hozzáadva a kosárhoz" gomb a felhasználói felületen (frontend).

2.Kattintás Kezelése: Amikor a felhasználó rákattint a gombra, egy JavaScript eseménykezelő aktiválódik.

3.Étel Azonosítója: Az eseménykezelő lekéri az étel egyedi azonosítóját, amely a gombhoz van rendelve.

4.Kosár Állapotának Frissítése: A frontend frissíti a kosár állapotát, beleértve az étel azonosítóját és a kiválasztott mennyiséget.

5. HTTP Kérés: Egy HTTP GET kérés indítódik a backend felé az étel azonosítójával és a mennyiséggel.

6.Backend Feldolgozás: A backend fogadja a kérést, és ellenőrzi az étel elérhetőségét az adatbázisban.

7.Tranzakció Kezelése: Ha az étel elérhető, a backend egy tranzakciót indít a kosár adatbázisban történő frissítésére.

8.Sikeres Hozzáadás: Sikeres tranzakció esetén a backend egy sikeres választ küld vissza a frontendnek.

9. Kosár Frissítése: A frontend frissíti a kosár megjelenítését, hogy tükrözze az újonnan hozzáadott elemet.

10.Visszajelzés a Felhasználónak: A felhasználói felületen egy visszajelzés jelenik meg, amely megerősíti az étel kosárhoz adását.

A backend és frontend közötti kommunikáció Restful api technológiát használ, amely lehetővé teszi a felhasználói felület frissítését anélkül, hogy újratöltenénk az oldalt. A kosár állapotának kezelése a frontend oldalon egy globális állapotkezelő segítségével történik, mint például a Context API, ami biztosítja az állapot konzisztenciáját a felhasználói műveletek során. A backend tranzakciókezelése biztosítja, hogy az adatbázis műveletek atomiak és konzisztensek legyenek, így elkerülve az adatvesztést vagy inkonzisztenciát.

Bővebben az új termék hozzá adásáról:

1. Kép Feltöltése: Az ételhez tartozó kép egy fájlfeltöltő komponensen keresztül tölthető fel.

2. Adatok Validálása: A rendszer ellenőrzi az adatok helyességét és formátumát.

3. Axios POST Kérés: Az adatokat egy Axios POST kérés segítségével küldik a backend felé.

4. Backend Feldolgozás: A backend fogadja a kérést és validálja az adatokat.

5. Adatbázisba Írás: Az érvényesített adatok az adatbázisba kerülnek mentésre.

6. Visszajelzés a Felhasználónak: Sikeres hozzáadás esetén a rendszer visszajelzést küld.

7. Menü Frissítése: A frontend frissíti a menüt, hogy az új étel megjelenjen.

A funkció működése során a következő technikai részletek figyelembevételével történik:

1. Eseménykezelők: A frontend eseménykezelői biztosítják a felhasználói interakciók kezelését.

2. Axios Könyvtár: Az Axios segítségével kezeljük a HTTP kéréseket a frontend és a backend között.

3. Adatbázis Tranzakciók: Az adatbázis műveletek tranzakcióként kezelhetők, biztosítva az adatok integritását.

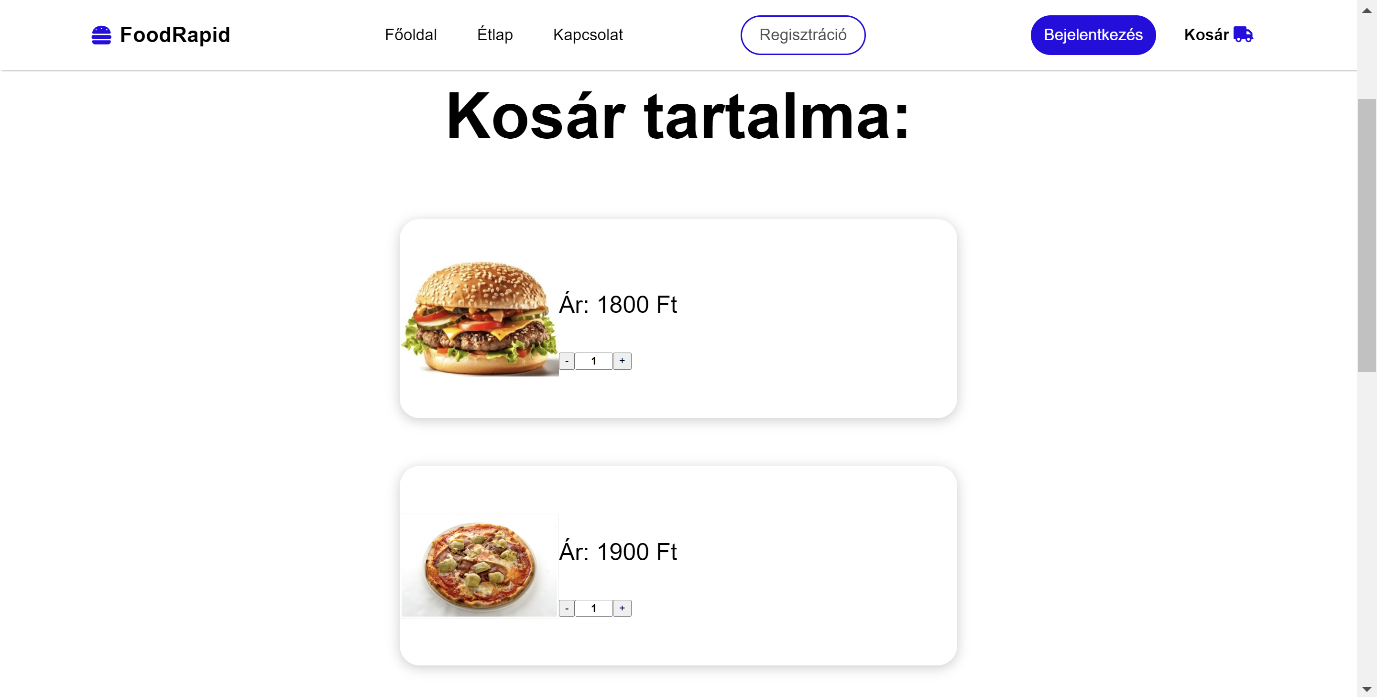
4. Hibakezelés: A rendszer hibakezelő mechanizmusokat tartalmaz, amelyek kezelik a lehetséges kivételeket.

5. Biztonság: Az adatok biztonságos átvitele HTTPS kapcsolaton keresztül történik.

6. Felhasználói Felület Frissítése: A frontend dinamikusan frissíti a menüt az új étel hozzáadása után.

Az "Új Étel Hozzáadása" funkció egyszerűsíti az étlap karbantartását és lehetővé teszi a gyors reagálást a piaci igényekre. A backend és a frontend közötti kommunikáció zökkenőmentes, ami növeli a rendszer hatékonyságát és a felhasználói élményt.

## Kosár:

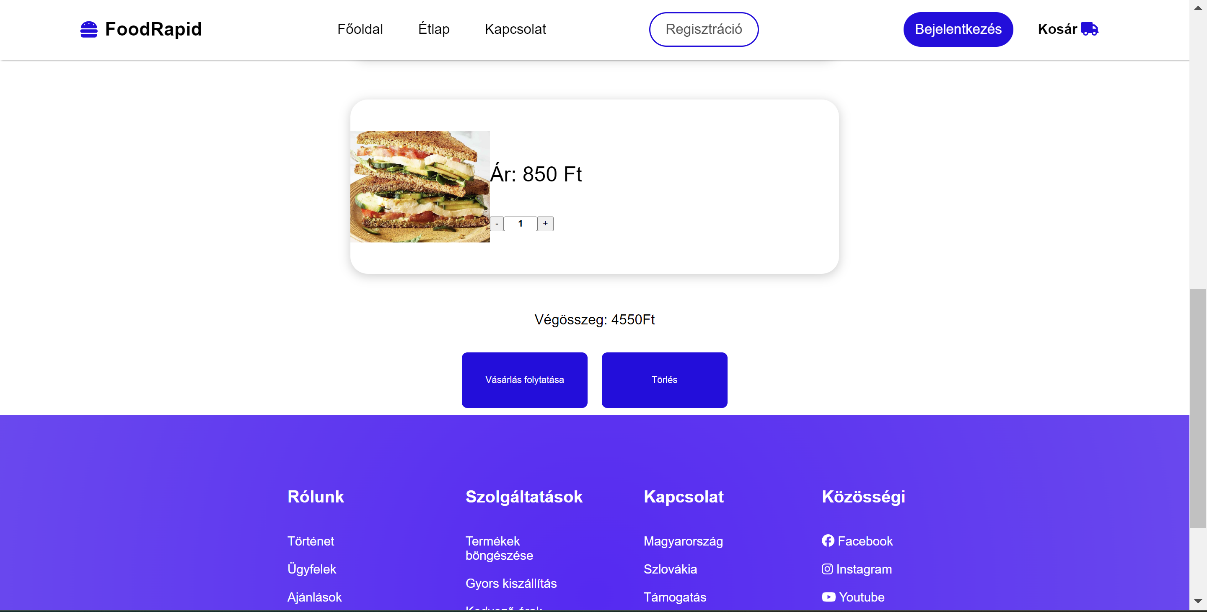
A kosárban található az összes olyan termék, ami hozzá lett adva az étlap menüpontból. 30.Ábra: Kosár

A kosárban megjelenik a termék képe, ára és mennyisége. A termékek mennyiségét a plusz és mínusz gombokkal tudjuk növelni és csökkenteni kedv szerint.

Elhelyeztünk még két gombot is a kosárban, amik felett megjelenik egy “végösszeg” felirat is. A gombok a következők:

## -Vásárlás folytatása

## -Törlés

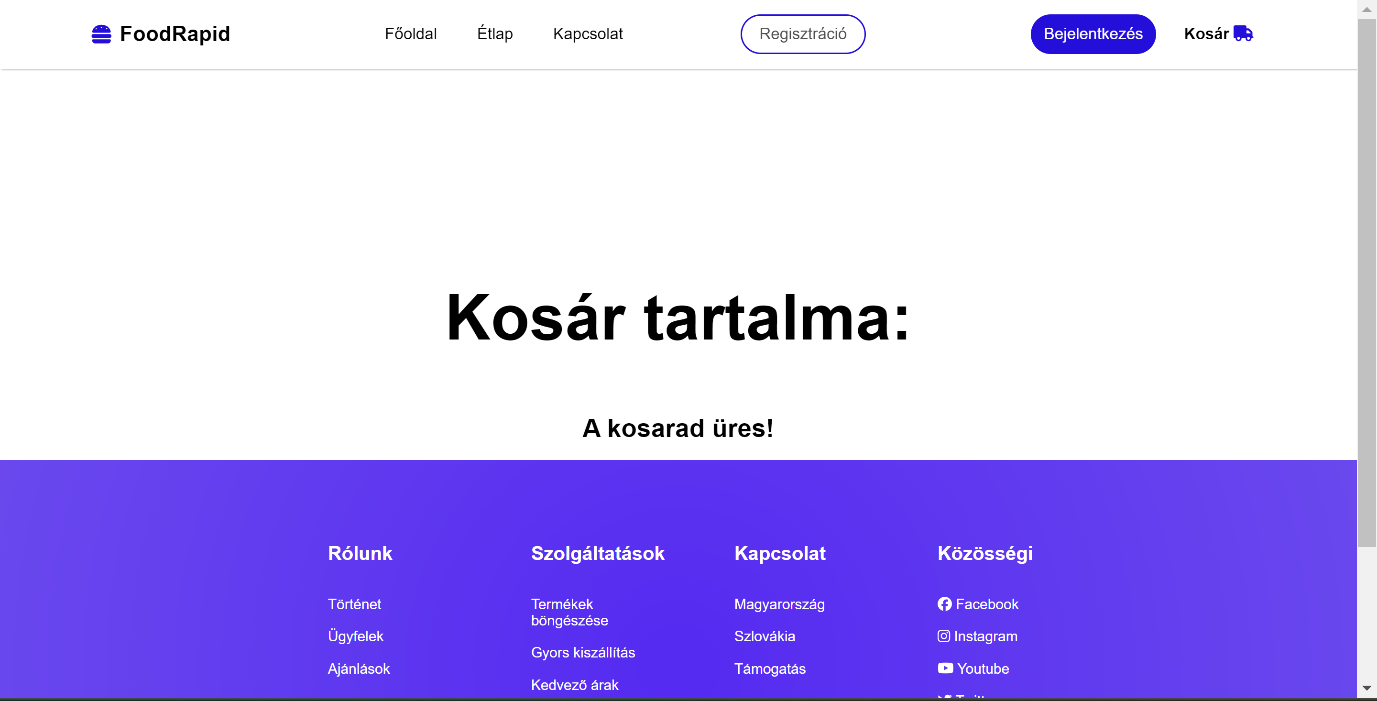


31.Ábra: Kosár

A vásárlás folytatása gombra kattintva visszatérhetünk a főoldalra, vagy a többi menüpontra anélkül, hogy a termékek eltűnnének vagy törlődnének a kosárból.

A kosár elemeit egyesével külön-külön is törölni tudja a felhasználó a mínusz gomb lenyomásával, addig amíg le nem kerül nullára a számláló.

Ha automatikusan kiakarja törölni az összes terméket, akkor ez egy nyomásra is megtehető a “Törlés” gomb lenyomásával.

32.Ábra: Törölt termékek

Ezek után megjelenik egy kiírás, hogy a termékek törlésre kerültek a kosárból.

Ha a vásárló ezek után visszatér az étlap menü pontra, hogy gombokban megjelenő számláló lenullázódott, és a termékek törlésre kerültek.

A törlés gombról bővebben:

A "Törlés" gomb lehetővé teszi a felhasználók számára, hogy eltávolítsanak egy terméket a kosárból. A folyamat az alábbi lépéseket tartalmazza:

1.Törlés Gomb Megjelenítése: Minden kosárban lévő termék mellett megjelenik egy "Törlés" gomb.

2.Termék ID Azonosítója: Minden termékhez egy egyedi ID azonosító tartozik.

3.Kattintás Kezelése: Amikor a felhasználó rákattint a "Törlés" gombra, egy eseménykezelő aktiválódik.

4.Axios DELETE Kérés: Az eseménykezelő egy Axios DELETE kérést indít a backend felé az ID azonosítóval.

5.Backend Feldolgozás: A backend fogadja a kérést és ellenőrzi a termék ID-ját.

6.Adatbázis Művelet: Ha a termék megtalálható, a backend elvégzi a törlés műveletet az adatbázisban.

7.Visszajelzés a Felhasználónak: Sikeres törlés esetén a backend egy sikeres válaszkódot küld vissza.

8. Kosár Frissítése: A frontend frissíti a kosár tartalmát, eltávolítva a törölt terméket.

9. Visszajelzés Megjelenítése: A felhasználói felületen egy megerősítő üzenet jelenik meg a sikeres törlésről.

A "Törlés" gomb működése során a következő technikai részletek figyelembevételével történik:

1.Eseménykezelők: A JavaScript eseménykezelők biztosítják a felhasználói interakciók kezelését.

2.Axios Könyvtár: Az Axios segítségével kezeljük a HTTP kéréseket a frontend és a backend között.

3.Adatbázis Tranzakciók: Az adatbázis műveletek tranzakcióként kezelhetők, biztosítva az adatok integritását.

4.Hibakezelés: A rendszer hibakezelő mechanizmusokat tartalmaz, amelyek kezelik a lehetséges kivételeket.

5.Felhasználói Felület Frissítése: A frontend dinamikusan frissíti a kosár tartalmát a törlés után.

A "Törlés" gomb használata egyszerű és intuitív, lehetővé téve a felhasználók számára, hogy gyorsan kezeljék a kosár tartalmát. Az Axios DELETE kérés biztosítja, hogy a törlési művelet gyorsan és biztonságosan történjen. A backend és a frontend közötti kommunikáció zökkenőmentes, ami növeli a felhasználói élményt. A kosárban lévő termékek kezelése kulcsfontosságú része a webáruház működésének, és a "Törlés" gomb funkciója elengedhetetlen a hatékony vásárlási folyamat biztosításához.

Tervezett funkciók:

Sajnálatos módon az időszűke, és egyéb közbejött incidens miatt elmaradt pár funkció, amit még beszerettünk volna tenni a webáruházba:

* Egy asztal foglaló rendszer a regisztrált felhasználók számára
* Külön asztal foglalás menüpont
* Kidolgozotabb dizájn a webáruház számára
* Szebb főoldal
* Tartalmasabb termék paletta
* Elfelejtett jelszó generálása

Továbbá elakartuk végezni a webáruházunk különböző részeinek tesztelését is, de sajnálatos módon az időszűke miatt erre már nem tudtunk sort keríteni.

A fejlesztések során a GitHub nevű verziókövető-szolgáltatást használtuk annak érdekében, hogy maximalizáljuk a hatékonyságunkat és elkerüljünk bármi nemű kompatibilitási problémát, ami a különböző állapotú programrészek egyesítése során felléphet.

# Forrásjegyzék:

* **HTML**: További információk: <https://en.wikipedia.org/wiki/HTML> 2024. május 2.
* **CSS**: További információk: <https://en.wikipedia.org/wiki/CSS> 2024. május 3.
* **BOOTSTRAP**: További információk: <https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)> 2024. április 23.
* **MYSQL**: További információk: <https://en.wikipedia.org/wiki/MySQL> 2024. április 30.
* **REACT**: További információk: <https://en.wikipedia.org/wiki/React_(JavaScript_library)> 2024. április 17.
* **ASP.NET CORE**: További információk: <https://en.wikipedia.org/wiki/ASP.NET_Core> 2024. február 19.
* **ENTITY FRAMEWORK**: További információk: <https://en.wikipedia.org/wiki/Entity_Framework> 2024. március 9.
* **JWT**: További információk: <https://en.wikipedia.org/wiki/JSON_Web_Token> 2024. február 27
* **AXIOS**: További információk: <https://axios-http.com/docs/intro> 2020