

Projektmunka dokumentálása

Mozgóbolt webalkalmazás



Premontrei Szakgimnázium és Technikum

Keszthely

2025

Szoftverfejlesztő és tesztelő szak

5-0613-12-03

Készítette:

Igazi Kristóf Tamás, Hamza Richárd

Tartalom

Bevezetés.....	3
A Projekt célja.....	5
Adatbázis modell:.....	7
Adatbázis felépítése.....	10
Button.jsx, Badge.jsx, Input.jsx.....	13
Navigáció és keresés (page.jsx, sales.jsx).....	13
product-card.jsx	13
Backend oldalon: cartController.jsx	14
Bővíthetőség, fejlesztési javaslatok	Hiba! A könyvjelző nem létezik.
MIDDLEWARE.....	20
Backend oldalról:	21
BACKEND:	24
Környezetvédelmi hatások	28
Használati előnyök	30
Lehetséges jövőbeli fejlesztések	32
Biztonságos rendszer.....	34
Reflexió	36

Bevezetés

A vizsgafeladat keretében egy mozgóbolt rendszer fejlesztését valósítottuk meg. Projektünk egy olyan élelmiszerboltot modellez, amely könnyen elérhető, hatékony kiszállítást biztosít, és speciális igényeket is kielégít. Az elkészült rendszer célja a vásárlás egyszerűsítése, a környezeti terhelés csökkentése, valamint azoknak az embereknek a segítése, akik bármilyen okból kifolyólag nehezebben mozognak, például kisvárosokban, falvakban élők, akik számára a legközelebbi bolt nagy távolságra található.

A mozgóbolt egy különleges szolgáltatás, amely a hagyományos üzletekhez képest rugalmasabb módon juttatja el a mindennapi élelmiszereket és egyéb alapvető termékeket a vásárlókhoz. Egy ilyen bolt általában egy jól felszerelt teherautó vagy kisbusz, amely előre meghatározott útvonalon haladva több települést is kiszolgál. Elsősorban olyan helyeken működik hatékonyan, ahol kevés a helyi üzlet, vagy az emberek nehezebben jutnak el egy bevásárlóközpontba.

A rendszer működése egyszerű, mégis hatékony. A mozgóbolt meghatározott menetrend szerint közlekedik, és a kijelölt megállókon a lakosok előre tudják, mikor érkezik. Az árukínálat az alapvető élelmiszerektől kezdve a háztartási cikkekig széles skálán mozog, sőt, előrerendelési lehetőséget is biztosít, hogy minden vásárló garantáltan hozzájusson a szükséges termékekhez.

A mozgóbolt nem csupán egy áruszállító jármű, hanem egy közösségi tér is, amely hozzájárul a vidéki és kisebb településeken élők mindennapjainak megkönnyítéséhez. Egy ilyen rendszer nemcsak kényelmes és hatékony megoldást kínál a vásárlásra, hanem fenntartható alternatívát is jelent a hagyományos üzletekkel szemben, csökkentve az egyéni utazások számát és ezáltal a környezeti terhelést.

A projekt megvalósítása során lelkesen dolgoztunk, és rengeteg új dolgot tanultunk! Nemcsak a mozgóboltok működéséről szereztünk mélyebb ismereteket, hanem megtapasztaltuk, milyen izgalmas egy ilyen rendszert megtervezni és létrehozni. Összehangolt csapatmunkával, kreatív ötletekkel és sok nevetéssel haladtunk előre, miközben egyre jobban beleláttunk az élelmiszerellátás gördülékeny megszervezésébe.

Munkánk során nemcsak problémamegoldó készségeinket fejlesztettük, hanem azt is megtanultuk, milyen fontos egy olyan szolgáltatás, amely valódi segítséget nyújt az embereknek. Különösen izgalmas volt látni, hogyan lehet egy egyszerű ötletből egy hatékony, fenntartható és közösségépítő rendszer, amely megkönnyíti a mindennapokat.

Összességében elmondhatjuk, hogy ez a projekt nemcsak egy kihívás volt, hanem egy élményekkel teli, tanulságos és szórakoztató utazás is, amely során sok új ismerettel gazdagodtunk és talán egy kicsit jobban megértettük, hogy egy apró változás is hatalmas segítséget jelenthet mások számára.

A Projekt célja

A VándorBolt mozgóbolt rendszerének kifejlesztésével olyan innovatív megoldást kívánunk létrehozni, amely egyszerre szolgálja ki a társadalmi, gazdasági és környezetvédelmi igényeket. A projekt célja, hogy egy modern, felhasználóbarát online platform és egy hatékony, jól szervezett logisztikai rendszer segítségével új szintre emeljük a kereskedelmet a vidéki térségekben, különösen az elzárt, nehezen megközelíthető településeken.

Társadalmi szerepvállalás és esélyegyenlőség:

Elsődleges célunk a vidéken élők támogatása, különös tekintettel az idősekre, a mozgássérültekre, valamint azokra a személyekre, akik közlekedési lehetőségek hiányában nem tudják rendszeresen elérni a nagyobb városok boltjait vagy szupermarketjeit. A rendszer lehetőséget biztosít számukra arra, hogy méltó módon, egyszerűen és kényelmesen juthassanak hozzá az alapvető élelmiszerekhez, háztartási cikkekhez és egyéb szükséges termékekhez.

Időhatékonyság és kényelem:

A mozgóbolt rendszer célja, hogy időt takarítson meg a felhasználók számára azáltal, hogy az online rendelés lehetőségével és az ajtóig történő kiszállítással csökkenti a bevásárlásra fordított időt. Ez különösen fontos lehet a kisgyermekes családok, elfoglalt dolgozók vagy egészségügyi okok miatt otthonukhoz kötöttek számára. Az egyszerű kezelőfelületű webalkalmazás lehetővé teszi a gyors, stresszmentes vásárlást, miközben biztosítja a termékek naprakész elérhetőségét és aktuális árait.

Fenntarthatóság és környezeti tudatosság:

A VándorBolt rendszere hozzájárul a fenntartható közlekedéshez és logisztikához. Egyetlen, optimálisan megtervezett útvonalon közlekedő jármű képes egyszerre több háztartást kiszolgálni, így jelentősen csökkenti a személyes autóhasználat szükségességét a vásárlások során. Ezáltal csökken az üzemanyag-felhasználás, a károsanyag-kibocsátás, valamint a közlekedés okozta környezeti terhelés. A rendszer nemcsak gazdaságos, hanem ökológiailag is felelősségteljes alternatívát kínál.

Digitális felzárkóztatás és modernizáció:

Egy másik fontos célunk a digitális szolgáltatásokhoz való hozzáférés bővítése. A projekt ösztönzi a felhasználókat – különösen az idősebb korosztályt – az online eszközök használatára, ezáltal növeli digitális kompetenciáikat és segíti őket a modern technológiákhoz való alkalmazkodásban.

Gazdaságélénkítés helyi szinten:

A rendszer egyúttal lehetőséget teremt a helyi termelők, kistermelők és kézművesek számára is, hogy termékeiket közvetlenül eljuttathassák a vásárlókhoz, ezáltal támogatva a helyi gazdaságot és csökkentve a szállítási lánc hosszát.

Összességében a VándorBolt mozgóbolt rendszer célja egy olyan digitális és fizikai szolgáltatás integrálása, amely egyszerre hoz kényelmet, elérhetőséget és fenntarthatóságot a vidéki vásárlók mindennapjaiba.

Adatbázis modell:



User modell

Tábla: users

Mezők: id, email, jelszo, felhasznalonev, telefonszam, vezeteknev, keresztnév, szulesidatum, szerep, szamlazasi_nev, szamlazasi_irsz, szamlazasi_telepules, szamlazasi_kozterulet, szamlazasi_hazszam, adoszam, szallitasi_nev, szallitasi_irsz, szallitasi_telepules, szallitasi_kozterulet, szallitasi_hazszam, hirlevel

Kapcsolatok: kapcsolatban áll a rendeles táblával (hasMany), egy usernek több rendelése lehet

Termek modell

Tábla: termék

Mezők: azonosito, nev, ar, egysegnyiar, csoport, termekleiras, kiszereles, keszlet, akciosar, akcios_egysegnyiar, akcio_vege, akcio_eleje, hivatkozas, ajanlott_termek, tizennyolc, afa_kulcs, meret, szin, kepUrl, vonalkod

Kapcsolatok: tartozik egy csoporthoz (belongsTo), és több rendelestételhez vagy raktárkészlethez kapcsolódhat

Csoport modell

Tábla: csoport

Mezők: azonosito, nev, csoport, hivatkozas, tizennyolc

Kapcsolatok: több termék is tartozhat egy csoporthoz (hasMany)

Raktár modell

Tábla: raktár

Mezők: azonosito, rendszam, max_kapacitas, megjegyzes

Kapcsolatok: kapcsolódik a raktár_keszlet, raktár_keszlet2 és napi_fogyas táblákhoz

RaktárKészlet modell

Tábla: raktár_keszlet

Mezők: raktarId, termekId, keszlet, utolso_frissites, megjegyzes

Kapcsolatok: tartozik egy raktárhoz és egy termékhez (belongsTo), valamint a raktár és termék hasMany kapcsolatban van vele

RaktarKeszlet2 modell

Tábla: raktar_keszlet2

Mezők: raktarId, termékId, keszlet, utolso_frissites, megjegyzes

Kapcsolatok: ugyanazok mint a raktar_keszlet esetében, alternatív tároló tábla

napi_fogyas modell

Tábla: napi_fogyas

Mezők: azonosito, raktar, termék, mennyiség, datum, helyszin

Kapcsolatok: tartozik egy raktarhoz és egy termékhez (belongsTo)

Rendeles modell

Tábla: rendeles

Mezők: azonosito, rendelesAzonosito, vevoNev, vevoEmail, vevoTelefon, szallitasiCim, szallitasiVaros, szallitasiIrsz, fizetesiMod, allapot, osszeg, szallitasiDij, kedvezmeny, vegosszeg, rendelesIdeje, felhasznaloId

Kapcsolatok: több rendelesTetelek kapcsolódhat hozzá (hasMany), tartozhat hozzá user is (belongsTo)

RendelesTetelek modell

Tábla: rendeles_tetelek

Mezők: azonosito, rendelesId, termékId, termékNev, mennyiség, egysegAr, osszAr

Kapcsolatok: tartozik egy rendeléshez és egy termékhez

Adatbázis felépítése

1. csoport tábla

Ez a tábla a termékek csoportosítására szolgál. Például italok, élelmiszerek, műszaki cikkek stb.

- **azonosito:** Egyedi, automatikusan növekvő elsődleges kulcs.
- **nev:** A csoport megnevezése.
- **csoport:** Valószínűleg hierarchikus csoportosításra utal (pl. főcsoport-alkategória).
- **hivatkozas:** Egyedi URL vagy referencia azonosító.
- **tizennyolc:** Jelzi, hogy 18 éven felülieknek szóló termékcsoporthoz tartozik.

2. raktar tábla

A mozgóboltozok azonosítására szolgál.

- **azonosito:** Elsődleges kulcs.
- **rendszam:** Egyedi rendszám (pl. gépjármű rendszáma).
- **max_kapacitas:** Maximálisan szállítható kapacitás (alapértelmezett: 1000).
- **megjegyzes:** Opcionális leírás a boltról.

3. termék tábla

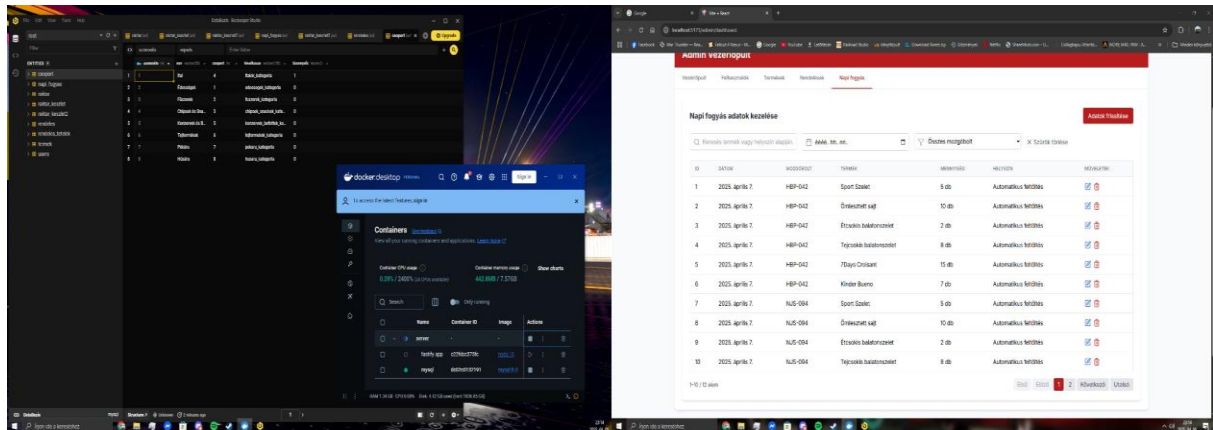
A termékek részletes leírását tartalmazza.

- **azonosito:** Egyedi termékazonosító.
- **nev, ar, egysegnyiar, kisereres:** A termék alapvető adatai.
- **csoport:** Kapcsolat a csoport táblával.
- **termekleiras:** Hosszabb szöveges leírás.
- **készlet:** Alapértelmezett készlet.
- **akciosar, akcios_egysegyiar, akcio_eleje, akcio_vege:** Akciós információk.
- **hivatkozas, ajanlott_termekkek, vonalkod:** Marketing, ajánlások, beazonosítás.
- **afa_kulcs, meret, szin:** Kiegészítő információk.
- **kepUrl:** A termékhez tartozó kép URL-je.

4. napi_fogyas tábla

A napi eladásokat és fogyást naplózza.

- **azonosito:** Elsődleges kulcs.
- **raktar, termék:** Külső kulcs a raktárra és termékre.
- **menyiseg, datum, helyszin:** Mennyi fogyott, mikor, hol.



5. raktar_keszlet és raktar_keszlet2 táblák

Mindkét tábla az egyes mozgóboltok (raktárak) aktuális készletét tárolja.

- **termekId, raktarId:** Hivatkozások a termék és raktar táblára.
- **keszlet:** Elérhető készlet adott helyen.
- **utolso_frissites:** Mikor történt az utolsó frissítés.
- **megjegyzes:** Opcionális megjegyzés.

A raktar_keszlet2 tábla redundánsnak tűnhet, de lehet, hogy külön funkciókat vagy tesztelést szolgál.

6. users tábla

A felhasználók (vevők, adminok stb.) adatait tartalmazza.

- **id:** Egyedi felhasználóazonosító.
- **email, jelszo, felhasznalonev:** Bejelentkezési adatok.
- **személyes és számlázási adatok:** Név, cím, telefonszám, adószám.
- **szerep:** admin, user, vevo szerepkör.
- **hirlevel:** Feliratkozott-e hírlevélre.

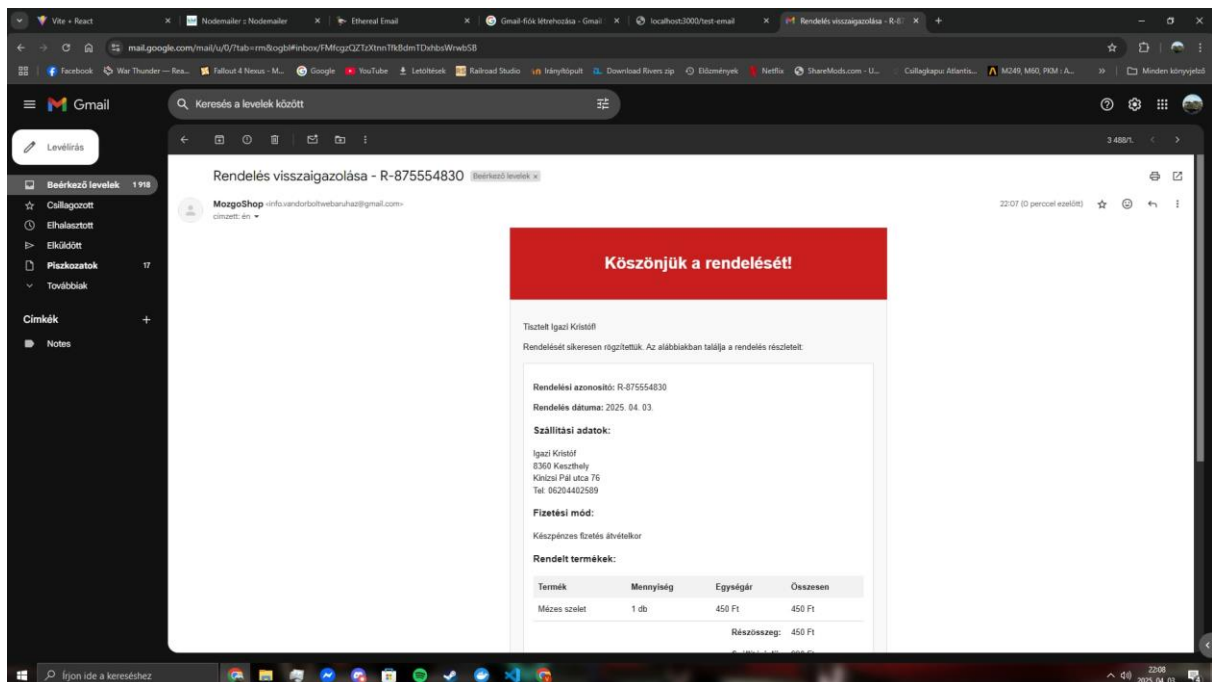
- **createdAt, updatedAt:** Regisztráció és utolsó módosítás időpontja.

Az email-címre számos egyedi megszorítás van, ezt optimalizálni lehetne.

7. rendeles tábla

A leadott rendeléseket rögzíti.

- **azonosito:** Egyedi rendelésazonosító.
- **rendelesAzonosito:** Egyedi publikus azonosító (pl. vásárlási bizonylat száma).
- **vevo adatai:** Név, email, telefon.
- **szallitasi adatok, fizetesiMod, állapot:** Szállítás és fizetés részletei.
- **osszeg, kedvezmeny, vegosszeg, szallitasiDij:** Pénzügyi adatok.
- **rendelesIdeje:** Mikor történt a rendelés.
- **felhasznaloId, termék:** Hivatkozások más táblákra.



8. rendeles_tetelek tábla

A rendeléshez tartozó terméktételeket tartalmazza.

- **rendelesId, termékId:** Rendelés és termék kapcsolata.
- **termékNev, mennyiség, egysegAr, osszAr:** A rendelés részletei.

Frontend

1. Áttekintés (Bevezetés)

- A VándorBolt célja: olyan mozgóbolt rendszer, amely egyszerre kínál online rendelést, helyi kiszállítást, és kedvezményes vásárlási lehetőségeket.
- Frontend: React + Tailwind CSS
- Backend: Node.js + Express
- Adatkezelés: REST API, localStorage (kosár) és MySQL alapú adatbázis

2. Felhasználói felület főbb elemei

Button.jsx, Badge.jsx, Input.jsx

- Újrahasznosítható UI elemek, egységes stílusban
- Button: variálható méret és stílus (default, outline, ghost)
- Badge: kiemelt információk vizuális megjelenítésére
- Input: keresőmezők és form mezők alapja

Navigáció és keresés (page.jsx, sales.jsx)

- Kategóriánként szűrhető terméklista
- Keresési funkció
- Oldalszámok közti lapozás
- Kosár ikon valós idejű számlálóval

3. Termékek kezelése

product-card.jsx

- Termékek kártyánkénti megjelenítése
- Akciós ár, kedvezmény %-ban
- „Kosárba” gomb, amely:
 - hozzáadja a terméket a localStorage-ban tárolt kosárhoz
 - elindít egy cartUpdated eseményt a globális frissítéshez

Backend oldalon: cartController.jsx

- Termék validáció és hozzáadás (adatbázisból ellenőrzi, létezik-e a termék)
- Dummy válasz, mivel a kosár helyben, kliensoldalon van tárolva
- Későbbi bővítésre alkalmas API (pl. felhő alapú tárolásra)



4. Kategóriakezelés

category-sidebar.jsx

- Dinamikus kategóriák betöltése szerverről
- Kattintásra paraméterezett URL generálás (?category=ID)

Fallback statikus kategóriák listája hálózati hiba esetén

5. Kosárkezelés (cart.jsx)

- localStorage alapú termékkosár
- Kosár tartalmának dinamikus betöltése és újrenderelése
- Termék mennyiségének növelése/csökkentése vagy törlése
- Kuponkód funkció (mozgo10 – 10% kedvezmény)

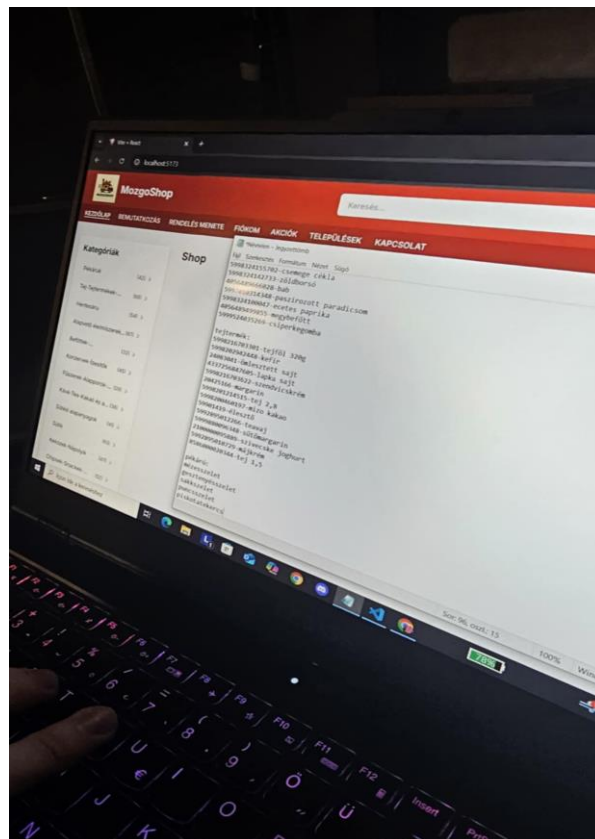
- Szállítási díj kiszámítása (10000 Ft felett ingyenes)

6. Fizetési folyamat (payment.jsx)

- Űrlap: szállítási és vásárlói adatok kitöltése
- Validálás kliensoldalon
- Kedvezmény kiszámítása és alkalmazása (százalékos)
- Megrendelés elküldése az api/rendeles endpointon keresztül (token alapú hitelesítéssel)
- Rendelés visszaigazolás oldala generált azonosítóval

7. Akciós termékek (sales.jsx)

- Backend API-ból lekért termékek szűrése: csak az akcióban lévők jelennek meg
- Kedvezmény kiszámítása, érvényességi dátum alapján
- UI és logika megegyezik a főoldali termékekkel



8. API-k és backend működés (cartController.jsx + adatbázis)

- Kosárhoz kapcsolódó műveletek REST API-n keresztül
 - addToCart, removeFromCart, updateCartItem, clearCart
- Az aktuális implementáció kliensoldalon tartja a kosarat, de elő van készítve szerveroldali kezelésre is
- Kosár szinkronizálása bejelentkezett felhasználók esetén szerverrel
- Bankkártyás fizetés integrációja
- Admin felület termékek, rendelések kezeléséhez
- Kiszállítási térkép vizualizáció
- Offline rendelés lehetősége (pl. sofőr tabletje)

10. info.jsx

Fő funkciók:

- **Oldalbetöltéskor** animált logó és cím megjelenítés.
- **Kosár státuszának figyelése:** localStorage és egyéni esemény (cartUpdated) alapján frissíti a kosár ikon melletti darabszámot.
- **Navigációs menü:** statikus linkeket jelenít meg különböző oldalakra (/info, /tutorial, stb.).
- **Bemutakozó szöveges tartalom:** rólunk, küldetés, jövőkép, miért érdemes választani őket.
- **Statikus képek betöltése:** logó, csapatkép.
- **Footer:** elérhetőségi adatok, információs linkek.

11. StaticKapcsolat.jsx – Kapcsolatfelvételi űrlap

Fő funkciók:

- **Form state kezelése:** formData objektum tartalmazza az űrlap mezőinek értékeit.
- **Submit működés:** fetch POST kérést küld a http://localhost:3000/email/contact végpontra JSON-ben az űrlapadatokkal.
- **Siker és hiba állapotkezelés:** submitStatus objektumban tárolódik, hogy sikeres, hibás vagy folyamatban van-e a beküldés.
- **Kosár ikon frissítése:** hasonlóan, mint az info.jsx-ben.
- **Statikus tartalom:** elérhetőségek, nyitvatartás, GYIK szekció.

12. route.jsx – Települések oldal

Fő funkciók:

- **Két különálló useState tömb** (towns1, towns2) kezeli a két különböző körzet településlistáját.
- Ezek az adatok **statikusan vannak kódolva**, nem API-ból érkeznek.
- **Képek** betöltése útvonal illusztrációhoz (ut1.png, ut2.png).
- **Kiszállítási információ** fix szöveggént szerepel (8:00–12:00).
- **Települések megjelenítése** táblázatszerű listákban.
- **Fejléc és lábléc**: ugyanaz a séma, mint a többi fájlban.

13. Tutorial.jsx – Rendelési útmutató

Fő funkciók:

- useState tömb (steps) tartalmaz minden egyes lépést:
 - cím, leírás, kép útvonal
- **Lépések sorrendben megjelennek**, mindegyik külön szekcióban.
- Tartalmaz egy “Vásárlás megkezdése” gombot, amely navigate('/')-tel a főoldalra visz.
- **Gyakori kérdések szekció**: statikus tartalom formájában jelenik meg.

14. Adminfelület és rendszerkezelés

AdminDashboard.jsx

- **Cél**: Admin vezérlőpult statisztikákkal és kezelőmodulokkal.
- **Funkciók**:
 - **Jogosultság ellenőrzés**: token alapján admin-e.
 - **Statisztikák betöltése**: felhasználók, termékek, rendelések, napi fogyás.
 - **Mutatók**: alacsony készlet, akciós termékek, rendelés állapotok, teljes bevétel.
 - **Navigálható panelek**: Vezérlőpult, Felhasználók, Termékek, Rendelések, Napi fogyás.

15. Adatbázis-kezelés (egyszerűbb admin nézet)

index.jsx (AdminDatabase)

- **Cél**: Másodlagos admin interfész, egyszerűbb adatlap-kezeléshez.
- **Funkciók**:
 - Token alapján azonosítja az admin jogosultságot.

- Fül-alapú navigáció UserTable, ProductTable, OrderTable között.
- Egyszerű és gyors mód az adatok kezelésére vizuális interfészen.

16. Felhasználókezelés

UserTable.jsx

- **Cél:** Felhasználók megjelenítése és adminisztrálása.
- **Funkciók:**
 - Listázás, keresés.
 - Szerkesztés (PUT metódus)
 - Új felhasználó létrehozása (POST)
 - Törlés megerősítéssel (DELETE)
 - szerep, hirlevel mezők kezelése

17. Termékek adminisztrációja

ProductTable.jsx

- **Cél:** Teljeskörű termékkezelés a backend API-n keresztül.
- **Funkciók:**
 - Termékek betöltése és frissítése (GET, PUT)
 - Új termék hozzáadása kép feltöltéssel (POST, FormData)
 - Törlés (DELETE)
 - Képek előnézete, dátumkezelés (akció időszak)
 - Kötelező mező: vonalkod

18. Rendelések kezelése

OrderTable.jsx

- **Cél:** Rendelések listázása, állapotmódosítás, részletek megtekintése.
- **Funkciók:**
 - Rendelések betöltése és frissítése (GET, PUT)
 - Állapotváltoztatás (pl. "kiszállítás alatt")
 - Rendelés részletek modalban
 - Törlés funkció

19. Napi fogyás nyilvántartás

NapiFogyasTable.jsx

- **Cél:** Mozgóbolt napi készletcsökkenésének adminisztrálása.
- **Funkciók:**
 - Adatok lekérése / szűrés helyszín, dátum, raktár szerint
 - Szerkesztés és törlés (PUT, DELETE)
 - Lapozás, keresés, szűrők
 - termékData és raktar adatkapcsolat megjelenítése

20. Hitelesítés és fiókkezelés

Login.jsx

- Bejelentkezés token alapú hitelesítéssel
- Token tárolása cookie-ban + localStorage-ban
- Hibakezelés, figyelmeztetés, átirányítás
- Sikeres bejelentkezés után profil betöltése és mentése localStorage-ba

Profile.jsx

- Felhasználói adatok megjelenítése token alapján
- Newsletter státusz
- Kijelentkezés (cookie törlése, átirányítás)
- Betöltés állapota és hibakezelés

Register.jsx

- Új felhasználó regisztrációs űrlap
- Jelszó ellenőrzés (megerősítés)
- Regisztrációs email küldése
- Sikeres regisztráció után automatikus átirányítás login oldalra

MIDDLEWARE

1. Bejelentkezés – Login.jsx

Backend kapcsolat:

- POST /user/login
- GET /user/profile (a bejelentkezés után a token ellenőrzéséhez)

Elvárt backend funkciók:

- E-mail + jelszó ellenőrzés
- JWT token generálás
- Token érvényesség ellenőrzése a /user/profile hívásban

Backend validálás:

- Hibaüzenet rossz jelszó/email esetén
- Token hiányában 401-es státuszkód

2. Regisztráció – Register.jsx

Backend kapcsolat:

- POST /user/adduser – új felhasználó létrehozása
- POST /email/registration-confirmation – regisztrációs e-mail küldése

Backend elvárások:

- E-mail egyediség ellenőrzése
- Jelszó titkosítás
- Alapértelmezett értékek (szerep: user, hirlevel)
- E-mail küldés – SMTP vagy külső API integráció

3. Fiók adatok – Profile.jsx

Backend kapcsolat:

- GET /user/profile – token alapú azonosítással

Backend oldalról:

- Token dekódolása és azonosított felhasználó visszaadása
- Biztonsági szempontból ne tartalmazzon jelszót

4. Admin jogosultság ellenőrzése – több helyen (AdminDashboard.jsx, index.jsx stb.)

Backend kapcsolat:

- GET /user/profile + token
- Jogosultság (szerep === "admin") alapján döntés

Backend funkciók:

- Token alapú hitelesítés middleware-ben
- Jogosultságellenőrzés minden admin végponton

5. Felhasználókezelés – UserTable.jsx

Backend végpontok:

- GET /user/all – összes felhasználó lekérése
- PUT /user/updateUser/:id – felhasználó frissítése
- POST /user/addUser – új felhasználó hozzáadása
- DELETE /user/deleteUser/:id – felhasználó törlése

Elvárások backend felől:

- Token hitelesítés minden kéréshez
- Validálás: kötelező mezők (email, felhasználonev)
- Ütközésellenőrzés (pl. duplikált e-mail)

6. Termékkezelés – ProductTable.jsx

Backend végpontok:

- GET /termek – termékek lekérése
- POST /termek – új termék létrehozása
- PUT /termek/:id – meglévő termék szerkesztése
- DELETE /termek/:id – törlés

Elvárt backend funkciók:

- FormData kezelés képfeltöltéssel (multer middleware)
- Akciós időszak validálása
- Vonalkód egyediség ellenőrzése
- Fájlkézelés (kép mentése, név generálása)

7. Rendeléskezelés – **OrderTable.jsx**

Backend végpontok:

- GET /rendeles – összes rendelés lekérése
- GET /rendeles/:id – egy rendelés részletei
- PUT /rendeles/:id – rendelés állapotának frissítése
- DELETE /rendeles/:id – rendelés törlése

Backend oldalról:

- Jogosultság-ellenőrzés: csak admin módosíthatja
- Védelem törölt rendelések újra módosítása ellen
- Adatbázis kapcsolatok (rendelés + rendelés tétel táblák)

8. Napi fogyás – **NapiFogyasTable.jsx**

Backend végpontok:

- GET /api/napi-fogyas – napi fogyás lista lekérése
- PUT /api/napi-fogyas/:id – bejegyzés módosítása
- DELETE /api/napi-fogyas/:id – bejegyzés törlése
- GET /api/raktar – elérhető raktárak lekérése

Backend feladatok:

- Token-alapú védelem
- termékData kapcsolat betöltése (JOIN vagy asszociációk)
- Dátumkezelés, formátum konvertálás
- Validálás (pl. raktár-azonosító, számok)

9. Admin statisztikák – **AdminDashboard.jsx**

Backend API hívások:

- GET /user/all – felhasználók számlálása
- GET /termek – termékek száma, akciós termékek
- GET /rendeles/stats – teljesített, függő, törölt rendelések és bevétel

Backend funkciók:

- Backend aggregáció (COUNT, SUM)
- Statisztikai számítások (pl. készlet < 10 db)
- recentOrders adatok visszaadása a bevételhez

10. Tokenkezelés minden adminhoz kapcsolódó fájlban

- Minden kéréshez szükséges: Authorization: Bearer <token>
- Backend oldalon middleware kell:

BACKEND:

Összegzés – Backend követelmények:

Funkció	Végpont	Művelet típus	Middleware
Bejelentkezés	<code>/user/login</code>	POST	-
Token validálás	<code>/user/profile</code>	GET	JWT
Regisztráció	<code>/user/adduser</code>	POST	-
Felhasználó kezelés	<code>/user/all</code> , <code>/user/updateUser/:id</code>	GET , PUT	JWT + role:admin
Termék kezelés	<code>/termek</code> , <code>/termek/:id</code>	GET , POST , PUT , DELETE	JWT + multer
Rendelés kezelés	<code>/rendeles</code> , <code>/rendeles/:id</code>	GET , PUT , DELETE	JWT + role:admin
Napi fogyás kezelése	<code>/api/napi-fogyas/:id</code>	GET , PUT , DELETE	JWT
Statisztika lekérés	<code>/rendeles/stats</code> , <code>/user/all</code>	GET	JWT + role:admin

Backend dokumentáció – VándorBolt Rendszer

1. Felhasználókezelés (UserController.js)

Fő funkciók:

- **Regisztráció** (createUser)
 - Bcrypt hash a jelszóhoz
 - Email + jelszó + név + hírlevél mező
- **Bejelentkezés** (authenticateUser)
 - Email/jelszó ellenőrzés
 - JWT token generálása (1 órás érvényesség)
 - role mező alapján jogosultság (pl. admin)
- **Profil lekérés** (getUserProfile)
 - Tokenből kiolvasott userId alapján

- **Admin funkciók**

- Összes felhasználó listázása (getAllUsers)
- Felhasználó frissítés/törlés
- Adatkonverziók (pl. születési dátum, irányítószám)

2. Termékkezelés (termekController.js)

Fő funkciók:

- **Termékek listázása / lekérdezése ID alapján**
- **Új termék létrehozása**
 - Multer-rel feltöltött kép kezelése (imageUrl)
 - Vonalkód egyedi validáció
- **Frissítés**
 - Részletes típusellenőrzés
 - Régi kép törlése fájlrendszerből
- **Törlés**
 - Kapcsolódó kép fájl törlése

3. Kosár kezelése (cartController.js)

Fő funkciók:

- **Kosár kezelése session alapon (nem adatbázis!)**
- Termék hozzáadása / frissítése / törlése
- Kosár teljes kiürítése
- Termék modell alapján lekérdezés + adatok csatolása a kosárhoz

4. Rendeléskezelés (rendelesController.js)

Fő funkciók:

- **Új rendelés létrehozása (createRendeles)**
 - JWT tokenből felhasználói azonosító
 - Egyedi azonosító generálása (pl. R-948237)
 - Tranzakciós kezelés: rendelés + tételek mentése egyszerre
 - Email küldés visszaigazolásként
-

- **Admin funkciók**
 - Rendelés állapot frissítése (feldolgozás, kiszállítás, stb.)
 - Rendelés törlés (tételekkel együtt)
 - Statisztikák: összes rendelés, bevétel, állapotok, legutóbbiak
- **Felhasználói funkciók**
 - Saját rendelések lekérdezése
 - Törlés "feldolgozás alatt" állapotban

5. Raktárak kezelése (raktarController.js)

Fő funkciók:

- Raktárak lekérdezése ID vagy rendszám alapján
- Új raktár létrehozása egyedi rendszámmal
- Frissítés: max. kapacitás és rendszám ütközésvizsgálat
- Törlés

6. Raktárkészlet – 2 verzióban

raktarKeszletController.js – kapacitásellenőrzéssel

- Új készlet hozzáadása mozgóbolthoz
 - Ellenőrzi: nem lépheti túl a maximális kapacitást
- Frissítés: akár raktarId/termekId módosítás, vagy csak készlet
- Összes készlet lekérdezése raktáronként / termékenként
- Validáció: vonalkód, egyedi kombináció, összKeszlet

raktarKeszlet2Controller.js – egyszerűbb logika

- Nem ellenőriz kapacitást
- Használható backupként, naplóként, vagy egyszerűbb esetekhez

7. Napi fogyás kezelése (napi_fogyasController.js)

Fő funkciók:

- Fogyások listázása termékkel és raktárral együtt
- Új napi fogyás rögzítése
- Fájl alapú feltöltés előkészítve (CSV/Excel)
- **Statisztikai funkciók:**

- Top 20 termék fogyás alapján
- Fogyások időbeni alakulása
- Raktáronkénti összesítések

8. Csoportok kezelése (csoportController.js)

Fő funkciók:

- Termékcsoportok CRUD kezelése
- Egyedi lekérdezés: termékek darabszáma csoportonként
- Hasznos statisztikai célra a főoldalon vagy adminpanelben

9. Biztonság és jogosultság

- JWT token használat
 - userId, email, role mezők a payloadban
- Token validálása middleware-en keresztül
- Szükséges védett útvonalakon: felhasználói profil, admin API-k
- A req.user objektum automatikusan elérhető a token alapján

10. Technikai alapok

Funkció	Részlet
Framework	Express.js
ORM	Sequelize
Adatbázis	MySQL vagy PostgreSQL
Hitelesítés	JWT token
Middleware	Multer (fájlkezelés), cookie-parser
Szerver oldali állapot	<code>req.session.cart</code> (kosár)

Környezetvédelmi hatások

A VándorBolt mozgóbolt rendszer nemcsak társadalmi és gazdasági szempontból jelent előrelépést, hanem a környezetvédelem területén is aktív szerepet vállal. A projekt szerves részét képezi a fenntartható fejlődés előmozdítása, amelynek során kiemelt figyelmet fordítunk a természetes erőforrások tudatos felhasználására, a károsanyag-kibocsátás csökkentésére, valamint a pazarlás visszaszorítására. Az alábbiakban részletezzük, hogyan járul hozzá a rendszer a környezeti terhelés mérsékléséhez.

1. Gázkibocsátás csökkentése – kevesebb jármű, kisebb ökológiai lábnyom

A hagyományos bevásárlási szokások során gyakran előfordul, hogy egy-egy család külön-külön, saját járművel indul el a legközelebbi boltba, gyakran jelentős távolságokat megtéve. Ezzel szemben a VándorBolt koncepciója azon alapul, hogy egyetlen mozgóbolt-jármű egyszerre több háztartást szolgál ki egyetlen útvonalon. Ez jelentősen csökkenti a forgalmat a közutakon, és mérhetően kisebb szén-dioxid- (CO_2) és egyéb károsanyag-kibocsátással jár. Azáltal, hogy egy jármű helyettesít több személygépkocsit, a rendszer közvetlenül hozzájárul az üvegházhatású gázok kibocsátásának mérsékléséhez.

2. Energiahatékonyság – optimalizált logisztika a fenntarthatóság szolgálatában

A mozgóbolt rendszer szívéét a logisztikai tervezés és az útvonaloptimalizálás képezi. A háttérrendszer figyelembe veszi a rendelések földrajzi elhelyezkedését, a kiszállítási időpontokat, valamint a forgalmi viszonyokat, hogy minden nap a lehető leghatékonyabb útvonalakon haladjanak a járművek. Ez nemcsak az üzemanyag-fogyasztást csökkenti, hanem a motorok működésének élettartamát is növeli, ezzel tovább mérsékelve a környezet terhelését. A kevesebb megtett kilométer és a kevesebb megállás–elindulás ciklus kisebb fogyasztást és kisebb mechanikai kopást is eredményez.

3. Élelmiszer-pazarlás minimalizálása – célzott és tudatos ellátás

A rendszer rendszeres ellátást biztosít a rászorulóknak számára, így megelőzhetővé válik az a gyakori probléma, hogy a háztartásokban a fel nem használt, elfelejtett élelmiszerek megromlanak, és végül a kukában végzik. A VándorBolt lehetővé teszi, hogy a felhasználók pontosan annyit rendeljenek, amennyire szükségük van, akár napi vagy heti szinten, így tudatosabb fogyasztói magatartás alakulhat ki. Emellett a szolgáltatás rugalmassága és gyakorisága csökkenti az „egyszerre sokat vásárolok” típusú pánikvásárlást is, ami szintén hozzájárul az élelmiszer-hulladék mérsékléséhez.

4. Lokális beszerzés ösztönzése – rövidebb ellátási lánc, kevesebb szállítás

A mozgóbolt rendszer célja az is, hogy a termékek egy részét lehetőség szerint helyi termelőktől, kistermelőktől szerezzük be. Ez nemcsak a helyi gazdaságot erősíti, hanem az ökológiai lábnyomot is csökkenti, mivel így elkerülhetők a több száz kilométeres szállítási útvonalak, amelyek egyébként jelentős környezeti terheléssel járnának.

5. Közlekedésbiztonsági és zajterhelési előnyök

A kevesebb járműmozgás nem csupán a levegő minőségére van pozitív hatással, hanem csökkenti a zajterhelést is a kisebb településeken. Ezen kívül a csökkenő forgalom javítja a közlekedésbiztonságot, különösen a szűkebb, gyalogosok által is sűrűn használt utcákon.

Használati előnyök

A VándorBolt mozgóbolt rendszer kialakításánál kiemelt figyelmet fordítottunk a felhasználói élményre, az egyszerű használhatóságra és a mindennapi életben való praktikus alkalmazhatóságra. A cél az volt, hogy olyan szolgáltatást hozzunk létre, amely minden korosztály és felhasználói réteg számára könnyen hozzáférhető, testre szabható, és valódi megkönnyebbülést jelent a mindennapos vásárlások terén.

1. Felhasználóbarát felület – mindenki számára könnyen használható rendszer

A mozgóbolt rendszerhez tartozó webalkalmazás modern, letisztult kialakítással rendelkezik, amelyet kifejezetten úgy terveztünk, hogy minden korosztály – a digitálisan jártas fiataloktól egészen az idősebb, kevésbé tapasztalt felhasználókig – könnyedén eligazodjon rajta. Az intuitív kezelőfelület lehetővé teszi a gyors navigálást a termékek között, a rendelés leadását néhány kattintással, valamint a korábbi rendelések áttekintését és újrarendelését is. A keresési és szűrési lehetőségek segítik a pontos, célzott vásárlást, így a felhasználók pontosan azt a terméket találják meg, amire szükségük van, idővesztés nélkül.

2. Kényelmes kiszolgálás – a bolt jön a vásárlóhoz, nem fordítva

A VándorBolt rendszer egyik legnagyobb előnye, hogy megszünteti a hagyományos bevásárlással járó fizikai erőfeszítést. A megrendelt termékek az előre egyeztetett időpontban közvetlenül a felhasználók otthonába érkeznek, ezzel különösen nagy segítséget nyújtva azok számára, akik nehezen tudják elhagyni otthonukat – például idősek, betegek, vagy kisgyermekes családok. A kiszállítás megbízható, pontos és rendszeres, így a felhasználók tervezni tudják a napi/heti vásárlásaikat anélkül, hogy a boltok nyitvatartásához kellene igazodniuk.

3. Személyre szabhatóság – rendelések az egyéni igényekhez igazítva

A rendszer rugalmas működése lehetővé teszi, hogy minden vásárló saját szokásaihoz és szükségleteihez igazítsa a rendelési gyakoriságot és a termékpalettát. Lehetőség van előre összeállított kosarak mentésére, heti ismétlődő rendelések beállítására.

A rendszer emlékeztetőket is küldhet a felhasználónak, ha elérkezik a megszokott rendelési idő, így semmiről nem feledkeznek meg.

4. Átláthatóság és nyomon követhetőség – biztonság és kontroll a vásárlás felett

A vásárlók a saját fiókjukban részletes áttekintést kapnak minden korábbi rendelésükről, a

szállítási státuszokról, valamint az aktuálisan futó akciókról és ajánlatokról. Ez fokozza az átláthatóságot, és lehetővé teszi, hogy tudatosabb döntéseket hozzanak a jövőbeli vásárlások során. Emellett a nyomon követhető kiszállítás és a digitális nyugtázási rendszer biztonságérzetet nyújt, hiszen a vásárló pontosan tudja, mikor és milyen állapotban érkeznek meg a termékek.

5. Idő- és energiatakarékosság – egyszerűség a mindennapokban

A rendszer használatával a felhasználóknak nem kell többé hosszú bevásárlólistákat írniuk, utazniuk, parkolóhelyet keresniük, vagy hosszú sorban állásokkal számolniuk. Mindezek helyett otthonról, kényelmesen, akár mobiltelefonról is intézhetik a vásárlást, így több idejük marad a családra, a munkára vagy a pihenésre.

Lehetséges jövőbeli fejlesztések

A VándorBolt mozgóbolt rendszer már jelenlegi formájában is korszerű és jól használható, azonban hosszú távon olyan új funkciók bevezetését tervezzük, amelyek tovább növelhetik a felhasználói élményt, a rendszer hatékonyságát és a szolgáltatás személyre szabhatóságát. A fejlesztések célja, hogy még rugalmasabban tudjunk alkalmazkodni a vásárlói igényekhez, miközben tovább növeljük a digitális szolgáltatásaink színvonalát.

1. Térképes nyomkövetés – valós idejű információ a szállításról

A tervek között szerepel egy integrált térképes nyomkövető rendszer bevezetése, amely lehetővé teszi, hogy a felhasználók valós időben követhessék rendeléseik helyzetét. Ez a funkció nemcsak növeli az átláthatóságot, hanem megnyugtató visszajelzést is nyújt a vásárlók számára: pontosan láthatják, hogy hol jár a mozgóbolt jármű, és mennyi idő van hátra a szállításig. Ez különösen fontos lehet olyan településeken, ahol a pontos kiszámíthatóság és az időhatékonyság kulcsfontosságú. A térképes követés a felhasználók bizalmát is növeli, hiszen az információ közvetlenül, megbízható forrásból érkezik, és bármikor elérhető az online felületen.

2. Dinamikus árazás és rugalmas kedvezményrendszer – igazodás a kereslethez

A jövőbeli fejlesztések részeként célunk egy dinamikus árazási rendszer bevezetése, amely figyelembe veszi az aktuális keresletet, a készlet szintjét, valamint a vásárlók rendelési gyakoriságát. A gyakran rendelő, hűséges vásárlók számára automatikusan alkalmazhatóak lennének egyéni kedvezmények, törzsvásárlói pontok vagy kuponrendszer. Ez nemcsak ösztönzőleg hat a vásárlásra, hanem még inkább személyre szabottá teszi a szolgáltatást.

Emellett heti szinten frissülő kínálattal és tematikus akciókkal is szeretnénk színesíteni a vásárlói élményt – például szezonális termékek kiemelése, helyi termelők külön ajánlatai, vagy időszakos csomagajánlatok formájában. A dinamikus frissítés segít abban is, hogy az árukészlet mindig naprakész és változatos legyen, elkerülve az unalmas, ismétlődő választékot, és helyet adva az új termékek kipróbálásának is.

3. Vásárlói profilok és ajánlások – intelligens rendszer személyes igényekre szabva

A felhasználói szokások alapján a rendszer képes lenne javaslatokat tenni új termékekre vagy ajánlatokra, hasonlóan a nagyobb online áruházak működéséhez. Ez mesterséges intelligencia vagy gépi tanulás segítségével automatizálható, és tovább növelheti a vásárlói élményt azáltal, hogy egyre relevánsabb ajánlatokat jelenít meg.

4. Kommunikációs fejlesztések – közvetlen kapcsolat a mozgóbolttal

Terveink között szerepel egy beépített üzenetküldő vagy visszajelző rendszer is, amellyel a vásárlók közvetlenül kapcsolatba léphetnek a mozgóbolt üzemeltetőjével. Így például kérhetnek pontosabb időpontot, visszajelezhetnek a rendelésről, vagy jelezhetnek egyedi igényeket, amiket az adott kiszállítás során figyelembe lehet venni.

Biztonságos rendszer

A VándorBolt mozgóbolt rendszer fejlesztése során kiemelt figyelmet fordítunk az információbiztonságra, valamint a felhasználók adatainak védelmére. A digitális világban egy szolgáltatás csak akkor lehet megbízható, ha a felhasználók biztosak lehetnek abban, hogy személyes adataik biztonságban vannak, és probléma esetén gyors, közvetlen segítséget kapnak. Ennek szellemében a rendszerünk több szinten is garantálja az adatok és a működés biztonságát.

1. Adatvédelem és titkosított kommunikáció – biztonság a háttérben

A webes felületen történő regisztráció és bejelentkezés során a felhasználók adatai – így például e-mail cím, név, lakcím és jelszó – titkosítva kerülnek tárolásra és továbbításra. A rendszer HTTPS-protokollt használ, amely garantálja a kommunikáció biztonságát a felhasználó böngészője és a szerver között. Emellett modern titkosítási algoritmusokat alkalmazunk a jelszavak és más érzékeny adatok kezelésére, így azok nem kerülhetnek illetéktelen kezekbe még akkor sem, ha a háttérrendszer támadás alá kerülne.

2. Felhasználói fiókvédelem és jogosultságkezelé

Minden felhasználó egyedi fiókkal rendelkezik, amelyet csak saját jelszavával érhet el. A rendszer több szintű jogosultságkezelést alkalmaz, így a vásárlók, az adminisztrátorok és a mozgóbolt üzemeltetők különböző hozzáférési jogokat kapnak. Ez biztosítja, hogy mindenki kizárólag a számára releváns adatokhoz és funkciókhoz férhessen hozzá.

3. Problémakezelés és támogatás – emberi segítség egy gombnyomásra

A VándorBolt weboldalán jól látható helyen található egy „Segítségnyújtás” menüpont, ahol a felhasználók elérhetik az ügyfélszolgálatot. A felületen megtalálható egy közvetlen telefonszám, amelyen keresztül a vásárlók azonnal kapcsolatba léphetnek a szolgáltatás munkatársaival bármilyen felmerülő kérdés vagy probléma esetén. Legyen szó technikai hibáról, elakadt rendelésről vagy kiszállítási időpont módosításáról, a gyors, személyes támogatás segít abban, hogy mindenki biztonságban és támogatva érezze magát a rendszer használata közben.

4. Rendszeres karbantartás és frissítések – naprakész védelem

A platform rendszeresen frissítésre kerül, így mindig a legújabb biztonsági protokollokat és javításokat alkalmazzuk. Ezzel minimalizáljuk a sérülékenységeket, és megelőzzük a külső támadásokat vagy rendszerhibákat. A karbantartások tervezetten, felhasználóbarát módon zajlanak, előre kiértékeléssel és minimális szolgáltatás-kieséssel.

5. Átláthatóság és adatkezelési irányelvek – felhasználóközpontú működés

A webes felületen elérhetők az adatkezelési tájékoztatók és a felhasználási feltételek, amelyek világosan és közérthetően ismertetik, hogy a rendszer milyen adatokat gyűjt, hogyan kezeli azokat, és milyen jogai vannak a felhasználóknak. Ezzel biztosítjuk az átláthatóságot és a felhasználók informált döntéshozatalát saját adataik kapcsán.

Reflexió

Ebben a projektmunkában ketten vettünk részt: Igazi Kristóf és én. Egy kiszállító mozgóbolttal kapcsolatos rendszert készítettünk, amelyhez saját kezűleg hoztunk létre egy működő weboldalt. A célunk az volt, hogy egy olyan digitális felületet alkossunk, amely bemutatja a mozgóbolt működését, és segíti annak hétköznapi használatát, például a rendeléskezelést és az adminisztrációt.

Minden részét mi ketten készítettük el a tervezéstől a megvalósításig. A weboldal felépítéséhez frontend és backend technológiákat egyaránt használtunk: HTML, CSS és JavaScript segítségével készítettük el a felhasználói felületet, a háttérrendszert pedig Node.js-sel, Express-szel és SQL adatbázissal építettük fel. A rendszer logikáját közösen dolgoztuk ki, és ügyeltünk arra, hogy átlátható, használható, és jól strukturált legyen.

A közös munka során sok új tapasztalatot szereztünk, és rengeteget fejlődtünk mind programozásban, mind csapatmunkában. Megtanultuk, hogyan osszuk meg a feladatokat hatékonyan, hogyan oldjunk meg problémákat együtt, és hogyan egészítsük ki egymás ötleteit. Sok közös emlék született, ami olykor vicces, máskor kicsit feszültebb volt, de összeségében szép emlékként gondolunk rá vissza. Jó volt látni, ahogy a saját kódjainkból egy működő, értelmes rendszer születik.

Büszkék vagyunk arra, amit ketten összehoztunk. Ez a projekt nemcsak szakmailag jelentett kihívást, hanem lehetőséget adott arra is, hogy meg tapasztaljuk, milyen az, amikor két ember közös munkájából valami valóban működő és értékes jön létre.



Forrás

<https://stackoverflow.com/questions>

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

sourcegraph.com

<https://openai.com/index/chatgpt/>

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

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

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

<https://github.com/>

<https://discord.com/>

<https://code.visualstudio.com/>