

# **Projektdokumentáció**

## **ZöldKosár Webáruház**

Premontrei Szakgimnázium és Technikum  
Készthely  
2025

Szoftverfejlesztő és tesztelő  
5-0613-12-03

Készítette:  
Wéber Zsombor és Kovács Tibor Benedek

# Tartalom

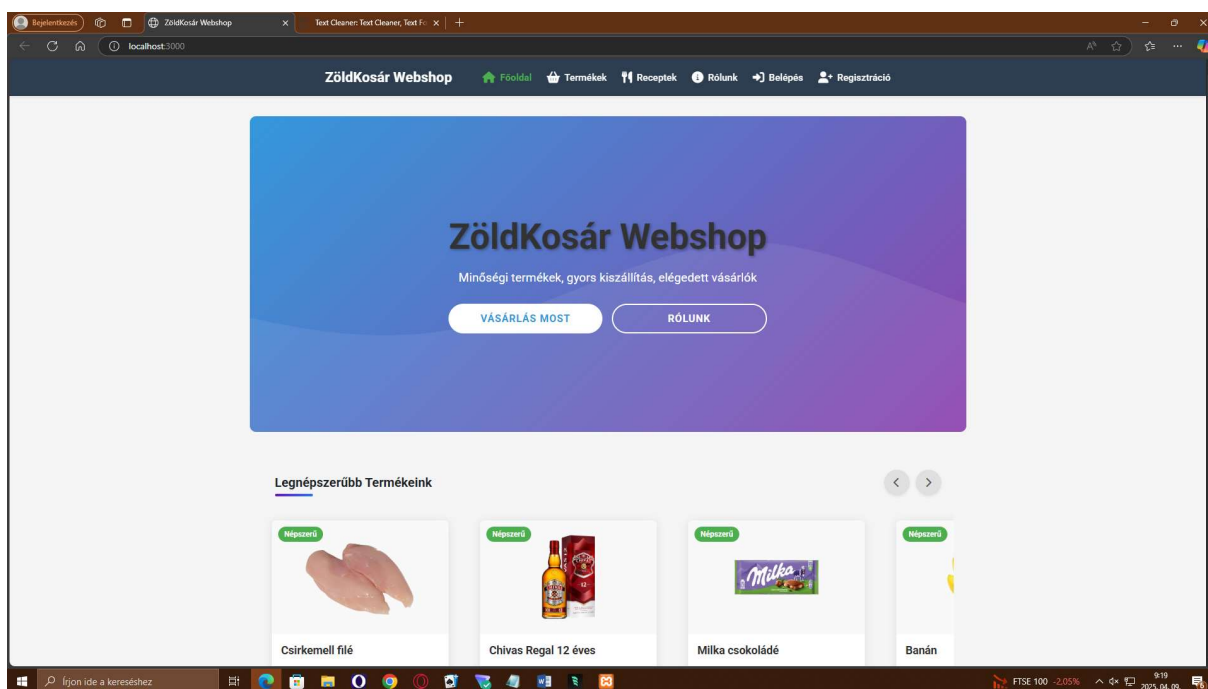
Bevezetés.....	3
A projekt célja és háttere .....	4
A webáruház fő értékei és küldetése .....	4
A dokumentáció a következő fő részekből áll: .....	6
Fejlesztőkörnyezetek .....	6
Verziókezelés .....	8
Frontend fejlesztőeszközök .....	8
Backend fejlesztőeszközök .....	9
Tervezőeszközök .....	9
Kommunikációs és projektmenedzsment eszközök .....	10
Hardvereszközök .....	11
Teszteléshez használt eszközök.....	11
Használt weboldalak és források.....	12
Dokumentációk és hivatalos források .....	12
Oktatási platformok és tutorialok .....	12
Design inspirációk és erőforrások .....	13
E-commerce referenciák.....	14
Technológiai háttér.....	15
Frontend technológiák.....	15
Backend technológiák .....	17
Adatbázis .....	20
Rendszerarchitektúra .....	22
Architektúra áttekintése.....	22
Frontend architektúra .....	22
Backend architektúra.....	24
Adatbázis architektúra.....	26
Funkcionalitás .....	28
Felhasználói felület .....	28
Vásárlási folyamat.....	30
Felhasználókezelés .....	32
Kosárkezelés.....	33
Rendeléskezelés .....	34
Összefoglalás.....	36

## Bevezetés

A ZöldKosár Webáruház projekt egy modern, felhasználóbarát online élelmiszerbolt, amely a fenntarthatóságra és a környezettudatosságra helyezi a hangsúlyt. A webáruház célja, hogy a vásárlók számára kényelmes és egyszerű módon biztosítson hozzáférést minőségi, friss élelmiszerekhez és háztartási termékekhez, miközben környezettudatos értékeket közvetít.

A projekt a Keszthelyi Premontrei Gimnázium két végzős tanulójának, Wéber Zsombornak és Kovács Tibor Benedeknek az érettségi projektmunkájaként készült. A fejlesztés során célunk volt egy teljes értékű, valós környezetben is használható webáruház létrehozása, amely nemcsak a vásárlók számára nyújt kellemes felhasználói élményt, hanem az üzemeltetők számára is hatékony adminisztrációs felületet biztosít.

A ZöldKosár Webáruház egyedisége abban rejlik, hogy a bevétel 20%-át állatmenhelyek támogatására fordítja, ezzel is erősítve a társadalmi felelősségvállalás fontosságát. Az oldal design-ja és funkcionalitása a modern webáruházak elvárásainak megfelelően került kialakításra, figyelembe véve a reszponzív megjelenítés követelményeit, hogy minden eszközön (asztali számítógép, tablet, mobiltelefon) optimális felhasználói élményt nyújtson.



## **A projekt célja és háttere**

A ZöldKosár Webáruház projekt elsődleges célja egy olyan online értékesítési platform létrehozása volt, amely:

- Lehetővé teszi a vásárlók számára, hogy kényelmesen, otthonról rendelhessenek friss élelmiszereket és háztartási termékeket
- Támogatja a fenntartható és környezettudatos életmódot a termékválasztékkal és a vállalati filozófiával
- Társadalmi felelősségvállalást mutat az állatmenhelyek támogatásával
- Modern, felhasználóbarát felületet biztosít, amely könnyen kezelhető minden korosztály számára
- Hatékony adminisztrációs rendszert nyújt a webáruház üzemeltetői számára

A projekt háttere a digitális kereskedelem növekvő jelentősége, különösen az élelmiszer-kiskereskedelem területén. A COVID-19 járvány után jelentősen megnőtt az online élelmiszer-vásárlás iránti igény, és ez a trend azóta is folytatódik. Emellett a környezettudatos és fenntartható életmód egyre fontosabbá válik a fogyasztók számára, ami új üzleti lehetőségeket teremt a környezettudatos értékeket képviselő vállalkozások számára.

## **A webáruház fő értékei és küldetése**

A ZöldKosár Webáruház küldetése, hogy összekösse a kényelmes online vásárlás előnyeit a környezettudatos és társadalmilag felelős értékekkel. A webáruház fő értékei a következők:

- Fenntarthatóság: A termékválasztékban előnyben részesítjük a helyi termelőktől származó, szezonális és környezetbarát termékeket.
- Társadalmi felelősségvállalás: A bevétel 20%-át állatmenhelyek támogatására fordítjuk, konkrétan a Zalaegerszegi Bogáncs Állatmenhely javára.
- Minőség: Csak megbízható forrásból származó, kiváló minőségű termékeket kínálunk vásárlóinknak.
- Átláthatóság: A termékek eredetéről, összetevőiről és környezeti hatásairól részletes információkat biztosítunk.

- Felhasználóbarát működés: Az oldal kialakítása során kiemelt figyelmet fordítottunk a könnyű kezelhetőségre és a pozitív felhasználói élményre.

### **Célközönség**

A ZöldKosár Webáruház elsődleges célközönsége:

- Környezettudatos vásárlók, akik fontosnak tartják a fenntarthatóságot és a környezetvédelmet
- Állatbarátok, akik szívesen támogatják az állatmenhelyeket
- Minőségi élelmiszereket kereső vásárlók, akik számára fontos a termékek eredete és összetétele
- Kényelmet kereső vásárlók, akik értékelik az online vásárlás nyújtotta időmegtakarítást
- Fiatal felnőttek és középkorú vásárlók, akik nyitottak az online vásárlásra és a környezettudatos életmódra

A webáruház ugyanakkor törekszik arra, hogy minden korosztály számára könnyen használható legyen, beleértve az idősebb generációkat is, akik esetleg kevésbé jártasak a digitális technológiák használatában.

### **A dokumentáció célja és felépítése**

A jelen dokumentáció célja, hogy átfogó és részletes képet adjon a ZöldKosár Webáruház projekt minden aspektusáról, beleértve a technológiai hátteret, a rendszerarchitektúrát, a funkcionalitást, valamint a fejlesztési folyamatot. A dokumentáció elsősorban a következő célokat szolgálja:

- A projekt teljes körű dokumentálása az érettségi projektmunka követelményeinek megfelelően
- Részletes technikai leírás biztosítása a rendszer működéséről és felépítéséről
- A fejlesztési folyamat és a felmerült kihívások bemutatása
- A rendszer továbbfejlesztési lehetőségeinek felvázolása

## **A dokumentáció a következő fő részekből áll:**

1. Bevezetés: A projekt céljának, hátterének és értékeinek bemutatása
  2. Használt programok és eszközök: A fejlesztés során használt szoftverek és hardverek ismertetése
  3. Használt weboldalak és források: A fejlesztés során felhasznált információforrások és inspirációk bemutatása
  4. Technológiai háttér: A frontend, backend és adatbázis technológiák részletes ismertetése
  5. Rendszerarchitektúra: A rendszer komponenseinek és azok kapcsolatainak bemutatása
  6. Funkcionalitás: A webáruház fő funkcióinak részletes leírása
  7. Admin panel működése: Az adminisztrációs felület funkcióinak és használatának bemutatása
  8. Adatbázis struktúra: Az adatbázis tábláinak, kapcsolatainak és mezőinek részletes ismertetése
  9. Biztonsági megfontolások: A rendszer biztonsági megoldásainak bemutatása
  10. Tesztelés: A tesztelési folyamat és eredmények ismertetése
  11. Továbbfejlesztési lehetőségek: A jövőbeli fejlesztési irányok felvázolása
  12. Összefoglalás: A projekt eredményeinek és tanulságainak összegzése
2. Használt programok és eszközök

A ZöldKosár Webáruház fejlesztése során számos modern fejlesztőeszközt és programot használtunk. Ezek kiválasztásánál fontos szempont volt, hogy naprakész, iparban is használt technológiákat alkalmazzunk, amelyek megfelelő támogatottsággal rendelkeznek és hatékony fejlesztést tesznek lehetővé.

### **Fejlesztőkörnyezetek**

#### **Visual Studio Code**

A projekt elsődleges fejlesztőkörnyezeteként a Microsoft Visual Studio Code-ot használtuk. Ez egy ingyenes, nyílt forráskódú kódszerkesztő, amely számos előnnyel rendelkezik:

- Könnyű, gyors és rugalmas
- Beépített Git integráció
- Gazdag bővítmény-ökoszisztéma
- Kiváló támogatás a JavaScript, React és Node.js fejlesztéshez
- Integrált terminál
- IntelliSense kódkiegészítés

A VS Code-hoz az alábbi bővítményeket telepítettük a hatékonyabb fejlesztés érdekében:

- ESLint: JavaScript kód minőségének ellenőrzésére
- Prettier: Kód formázásra
- React Developer Tools: React komponensek debugolására
- MySQL Extension: Adatbázis-kapcsolat kezelésére
- Live Server: Helyi fejlesztői szerver futtatására
- GitLens: Git verziókezelés kiterjesztett funkcióihoz

## **XAMPP**

A helyi fejlesztői környezet kialakításához XAMPP-ot használtunk, amely egy ingyenes, nyílt forráskódú webservert tartalmazó megoldás, amely tartalmazza a következőket:

- Apache HTTP szerver
- MariaDB (MySQL) adatbázis
- PHP interpreter
- phpMyAdmin webes adatbázis-kezelő felület

A XAMPP lehetővé tette számunkra, hogy helyi környezetben teszteljük a webáruházat, anélkül hogy külső szervert kellett volna igénybe vennünk a fejlesztés során.

## **Verziókezelés**

### **Git**

A forráskód verziókezeléséhez Git-et használtunk, amely lehetővé tette:

- A kódbázis változásainak követését
- Párhuzamos fejlesztést különböző funkciókon
- A kód biztonságos tárolását
- A fejlesztési folyamat dokumentálását a commit üzeneteken keresztül

### **GitHub**

A Git repository tárolásához és a csapatmunka koordinálásához GitHub-ot használtunk, amely:

- Webes felületet biztosít a Git repository kezeléséhez
- Lehetővé teszi a kód megosztását és a közös munkát
- Issue tracking rendszert biztosít a feladatok követéséhez
- Pull request mechanizmust kínál a kód áttekintéséhez és egyesítéséhez

## **Frontend fejlesztőeszközök**

### **Node.js és npm**

A Node.js egy JavaScript futtatókörnyezet, amely lehetővé teszi a JavaScript kód futtatását a szerveren. Az npm (Node Package Manager) pedig a Node.js csomagkezelője, amely segítségével könnyen telepíthetők és kezelhetők a különböző JavaScript könyvtárak és függőségek.

### **React Developer Tools**

A React Developer Tools egy böngésző bővítmény, amely lehetővé teszi a React komponensek vizsgálatát és debugolását a böngészőben. Segítségével könnyebben nyomon követhető a komponensek állapota és tulajdonságai.

### **Chrome DevTools**

A Google Chrome böngésző fejlesztői eszközei nélkülözhetetlenek voltak a frontend fejlesztés során. Segítségükkel:



- Debugolhattuk a JavaScript kódot
- Vizsgálhattuk és módosíthattuk a DOM-ot
- Elemezhattuk a hálózati kéréseket
- Optimalizálhattuk a teljesítményt
- Tesztelhattuk a reszponzív megjelenést különböző képernyőméreteken

## **Backend fejlesztőeszközök**

### **Postman**

A Postman egy API fejlesztő eszköz, amelyet az API-k tesztelésére és dokumentálására használtunk. Segítségével könnyen tesztelhattuk a backend API végpontjait, anélkül hogy a frontend-et kellett volna használnunk.

### **MySQL Workbench**

A MySQL Workbench egy vizuális adatbázis-tervező és adminisztrációs eszköz MySQL adatbázisokhoz. Segítségével:

- Terveztük és modelleztük az adatbázis struktúráját
- SQL lekérdezéseket írtunk és teszteltünk
- Kezeltük az adatbázis felhasználóit és jogosultságait
- Monitoroztuk az adatbázis teljesítményét

### **Nodemon**

A Nodemon egy fejlesztői eszköz, amely automatikusan újraindítja a Node.js alkalmazást, amikor változást észlel a fájlokban. Ez jelentősen felgyorsította a fejlesztési folyamatot, mivel nem kellett manuálisan újraindítani a szerveret minden változtatás után.

## **Tervezőeszközök**

### **Figma**

A felhasználói felület tervezéséhez Figma-t használtunk, amely egy webalapú UI/UX tervező eszköz. Segítségével:

- Elkészítettük a webáruház wireframe-jeit és prototípusait
- Megterveztük a felhasználói felület elemeit
- Kollaboráltunk a design folyamat során
- Exportáltuk a szükséges grafikai elemeket

## **dbdiagram.io**

Az adatbázis tervezéséhez a dbdiagram.io online eszközt használtuk, amely lehetővé teszi az adatbázis sémák vizuális tervezését és dokumentálását. Az eszköz segítségével könnyen létrehozhattuk az ER (Entity-Relationship) diagramokat, amelyek segítettek az adatbázis struktúrájának megtervezésében.

## **Kommunikációs és projektmenedzsment eszközök**

### **Discord**

A csapaton belüli kommunikációhoz Discord-ot használtunk, amely lehetővé tette:

- A valós idejű szöveges és hangalapú kommunikációt
- A képernyőmegosztást a közös problémamegoldáshoz
- A fájlok és kódrészletek megosztását
- A különböző témák szerinti csatornák létrehozását

### **Trello**

A projekt feladatainak követéséhez Trello-t használtunk, amely egy vizuális projektmenedzsment eszköz. Segítségével:

- Létrehoztuk és követtük a feladatokat
- Kiosztottuk a felelőségeket
- Nyomon követtük a projekt előrehaladását
- Határidőket állítottunk be és követtük azokat

## **Hardvereszközök**

A fejlesztés során az alábbi hardvereszközöket használtuk:

### **Teszteléshez használt eszközök**

- Különböző böngészők (Chrome, Firefox,, Edge) a keresztböngésző-kompatibilitás teszteléséhez

## **Használt weboldalak és források**

A ZöldKosár Webáruház fejlesztése során számos online forrást és weboldalt használtunk inspirációként, tanulási forrásként és problémamegoldáshoz. Az alábbiakban részletezzük a legfontosabb forrásokat kategóriák szerint.

### **Dokumentációk és hivatalos források**

#### **Frontend technológiák dokumentációi**

- React.js hivatalos dokumentáció ([reactjs.org](https://reactjs.org)): A React könyvtár hivatalos dokumentációja, amely részletes leírást és példákat tartalmaz a React komponensek, hook-ok és egyéb funkciók használatáról.
- MDN Web Docs ([developer.mozilla.org](https://developer.mozilla.org)): A Mozilla Developer Network átfogó dokumentációja a webes technológiákról, beleértve a HTML, CSS és JavaScript részletes leírását.
- React Router dokumentáció ([reactrouter.com](https://reactrouter.com)): A React Router könyvtár hivatalos dokumentációja, amely segített a kliens oldali routing megvalósításában.

#### **Backend technológiák dokumentációi**

- Node.js hivatalos dokumentáció ([nodejs.org/docs](https://nodejs.org/docs)): A Node.js részletes dokumentációja, amely segített a szerver oldali JavaScript fejlesztésben.
- Express.js dokumentáció ([expressjs.com](https://expressjs.com)): Az Express.js keretrendszer hivatalos dokumentációja, amely alapvető forrás volt a RESTful API kialakításához.
- Sequelize dokumentáció ([sequelize.org](https://sequelize.org)): A Sequelize ORM hivatalos dokumentációja, amely segített az adatbázis-műveletek implementálásában.
- MySQL dokumentáció ([dev.mysql.com/doc](https://dev.mysql.com/doc)): A MySQL adatbázis hivatalos dokumentációja, amely segített az adatbázis tervezésében és optimalizálásában.

### **Oktatási platformok és tutorialok**

#### **Online kurzusok**

- Udemy: Több kurzust is elvégeztünk a platformon, különösen a "MERN Stack Front To Back" és a "React - The Complete Guide" kurzusokat, amelyek átfogó ismereteket nyújtottak a React és Node.js fejlesztésről.

- freeCodeCamp (freecodecamp.org): Ingyenes oktatási platform, amely számos gyakorlati feladatot és projektet kínál a webfejlesztés különböző területein.
- Codecademy (codecademy.com): Interaktív programozási kurzusok, amelyek segítettek a JavaScript és React alapjainak elsajátításában.

### **Tutorial oldalak és blogok**

- CSS-Tricks (css-tricks.com): Hasznos cikkek és tutorialok a modern CSS technikákról, különösen a Flexbox és Grid layout rendszerekről.
- Medium (medium.com): Számos fejlesztői cikk és tutorial, különösen a React és Node.js témakörökben.
- Dev.to (dev.to): Fejlesztői közösségi platform, ahol számos hasznos cikket és megoldást találtunk különböző problémákra.
- Stack Overflow (stackoverflow.com): Nélkülözhetetlen forrás volt a fejlesztés során felmerülő specifikus problémák megoldásához.

### **Design inspirációk és erőforrások**

#### **Design inspirációk**

- Dribbble (dribbble.com): Design inspirációk gyűjtéséhez használtuk, különösen e-commerce felületek terén.
- Behance (behance.net): Kreatív projektek és design munkák, amelyek inspirálták a webáruház vizuális megjelenését.
- Awwwards (awwwards.com): Díjnyertes weboldalak gyűjteménye, amelyek inspirációt nyújtottak a modern webdesign trendekhez.

#### **Design erőforrások**

- Unsplash (unsplash.com): Ingyenes, magas minőségű képek forrása, amelyeket a webáruház tartalmához használtunk.
- Font Awesome (fontawesome.com): Ikonkészlet, amelyet a felhasználói felület elemeihez használtunk.
- Google Fonts (fonts.google.com): Ingyenes betűtípusok forrása, amelyeket a webáruház tipográfiájához használtunk.

## **E-commerce referenciák**

### **Hazai webáruházak**

- Tesco Online ([bevasarlas.tesco.hu](https://bevasarlas.tesco.hu)): Inspirációt nyújtott az élelmiszer kategóriák kialakításához és a termékek strukturálásához.
- Kifli.hu: Modern élelmiszer webáruház, amely inspirálta a felhasználói élmény kialakítását.
- Auchan Online ([online.auchan.hu](https://online.auchan.hu)): Referenciaként szolgált a kosár és rendelési folyamat kialakításához.

### **Nemzetközi webáruházak**

- Amazon Fresh ([amazon.com](https://amazon.com)): Inspirációt nyújtott a termékek megjelenítéséhez és a keresési funkciókhoz.
- Whole Foods Market ([wholefoodsmarket.com](https://wholefoodsmarket.com)): Környezettudatos élelmiszer webáruház, amely tartalmi és design szempontból is inspirációt jelentett.
- Ocado ([ocado.com](https://ocado.com)): Innovatív online élelmiszer-kiskereskedő, amely inspirálta a felhasználói felület kialakítását.

### **Közösségi platformok és fórumok**

- GitHub ([github.com](https://github.com)): Nyílt forráskódú projektek tanulmányozása és inspiráció gyűjtése.
- Reddit ([reddit.com/r/webdev](https://reddit.com/r/webdev), [reddit.com/r/reactjs](https://reddit.com/r/reactjs)): Fejlesztői közösségek, ahol hasznos tippeket és megoldásokat találtunk.
- Dev.to ([dev.to](https://dev.to)): Fejlesztői blogok és cikkek, amelyek segítettek a modern webfejlesztési technikák elsajátításában.

## Technológiai háttér

A ZöldKosár Webáruház fejlesztése során modern webes technológiákat alkalmaztunk, amelyek lehetővé tették egy gyors, biztonságos és felhasználóbarát webáruház létrehozását. Az alábbiakban részletesen bemutatjuk a használt technológiákat és azok szerepét a projektben.

### Frontend technológiák

#### React.js

A webáruház frontend része React.js JavaScript könyvtárral készült, amely a Facebook által fejlesztett nyílt forráskódú könyvtár a felhasználói felületek építéséhez. A React választását a következő előnyök indokolták:

- **Komponens-alapú architektúra:** A React lehetővé teszi az újrafelhasználható UI komponensek létrehozását, ami jelentősen felgyorsította a fejlesztést és konzisztens felhasználói élményt biztosított.
- **Virtuális DOM:** A React virtuális DOM-ja optimalizálja a renderelési folyamatot, ami jobb teljesítményt eredményez, különösen dinamikus tartalmak esetén.
- **Egyirányú adatáramlás:** A React egyirányú adatáramlási modellje kiszámíthatóbbá és könnyebben debugolhatóvá teszi az alkalmazást.
- **Nagy közösség és ökoszisztéma:** A React mögött álló nagy fejlesztői közösség és gazdag ökoszisztéma számos kiegészítő könyvtárat és eszközt biztosít.

#### A projektben használt főbb React funkciók és koncepciók:

- **Funkcionális komponensek:** Modern megközelítést alkalmaztunk, előnyben részesítve a funkcionális komponenseket a class komponensekkel szemben.
- **React Hooks:** Használtuk a useState, useEffect, useContext és egyéb hook-okat az állapotkezeléshez és életciklus-események kezeléséhez.
- **Context API:** A globális állapotkezeléshez Context API-t használtunk, ami lehetővé tette az adatok megosztását a komponensek között prop drilling nélkül.
- **Custom Hook-ok:** Saját hook-okat fejlesztettünk az ismétlődő logika újrafelhasználásához, például a termékek betöltéséhez vagy a kosár kezeléséhez.

## **React Router**

A kliens oldali navigációhoz React Router könyvtárat használtunk, amely lehetővé teszi a Single Page Application (SPA) navigációs struktúrájának kialakítását. Főbb előnyei:

- Deklaratív routing: Egyszerű, deklaratív szintaxis a route-ok definiálásához.
- Nested routes: Lehetőség egymásba ágyazott route-ok létrehozására, ami különösen hasznos volt az admin felület kialakításánál.
- URL paraméterek: Könnyű hozzáférés az URL paraméterekhez, amit a termékek szűrésénél és a keresési funkcióknál használtunk.
- Programozott navigáció: Lehetőség a navigáció programozott vezérlésére, például bejelentkezés után vagy kosárhoz adás után.

## **CSS és Styling megoldások**

A felhasználói felület stílusának kialakításához több megközelítést is alkalmaztunk:

- CSS modulok: Komponens-specifikus stílusok definiálásához CSS modulokat használtunk, ami megakadályozza a stílusok közötti ütközéseket.
- Flexbox és Grid: Modern CSS layout technikákat alkalmaztunk a reszponzív design kialakításához.
- Media query-k: A különböző képernyőméretekhez való alkalmazkodáshoz media query-ket használtunk.
- CSS változók: A konzisztens színpaletta és tipográfia biztosításához CSS változókat definiáltunk.

## **Axios**

A HTTP kérések kezeléséhez Axios könyvtárat használtunk, amely egy Promise-alapú HTTP kliens. Főbb előnyei:

- Egyszerű API: Könnyen használható API a HTTP kérések küldéséhez.
- Interceptorok: Lehetőség a kérések és válaszok globális módosítására, amit például az autentikációs token kezeléséhez használtunk.



- Automatikus JSON átalakítás: Az Axios automatikusan átalakítja a JSON válaszokat JavaScript objektumokká.
- Kérések megszakítása: Lehetőség a folyamatban lévő kérések megszakítására, ami hasznos volt a keresési funkciónál.

## **React-Toastify**

A felhasználói visszajelzések megjelenítéséhez React-Toastify könyvtárat használtunk, amely elegáns toast üzeneteket biztosít. Ez lehetővé tette a felhasználók tájékoztatását a különböző műveletekről, például sikeres bejelentkezésről, kosárhoz adásról vagy hibaüzenetekről.

## **React-Spring**

Az animációk megvalósításához React-Spring könyvtárat használtunk, amely fizika-alapú animációkat biztosít. Ez különösen hasznos volt a termékek betöltési animációinál és a szerencsekerék megvalósításánál.

## **Backend technológiák**

### **Node.js**

A szerver oldali alkalmazás fejlesztéséhez Node.js-t használtunk, amely egy JavaScript futtatókörnyezet, ami lehetővé teszi a JavaScript kód futtatását a szerveren. A Node.js választását a következő előnyök indokolták:

- Egységes programozási nyelv: Ugyanazt a programozási nyelvet (JavaScript) használhattuk mind a frontend, mind a backend fejlesztéséhez.
- Aszinkron I/O: A Node.js nem blokkoló I/O modellje kiváló teljesítményt biztosít, különösen I/O-intenzív alkalmazásoknál, mint egy webáruház.
- Gazdag ökoszisztéma: Az npm (Node Package Manager) hozzáférést biztosít a világ legnagyobb nyílt forráskódú könyvtár gyűjteményéhez.
- Skálázhatóság: A Node.js alkalmazások könnyen skálázhatók horizontálisan, ami fontos szempont volt a jövőbeli növekedés szempontjából.

### **Express.js**

A webszerver és API fejlesztéséhez Express.js keretrendszert használtunk, amely a Node.js legszélesebb körben használt webes keretrendszere. Főbb előnyei:

- Minimális és rugalmas: Az Express egy minimális, de rendkívül rugalmas keretrendszer, amely lehetővé tette a saját igényeinkre szabott API kialakítását.
- Middleware architektúra: Az Express middleware architektúrája lehetővé tette a kérések feldolgozásának moduláris kialakítását.
- Routing: Egyszerű és hatékony routing rendszer, amely lehetővé tette a RESTful API végpontok kialakítását.
- Integrációs lehetőségek: Könnyű integráció más könyvtárakkal és middleware-ekkel, például autentikációs megoldásokkal vagy adatbázis-kezelőkkel.

#### A projektben implementált főbb Express middleware-ek:

- cors: Cross-Origin Resource Sharing kezelése
- express.json(): JSON kérések feldolgozása
- cookie-parser: Cookie-k kezelése
- multer: Fájlfeltöltések kezelése
- Saját autentikációs middleware: JWT token ellenőrzése és felhasználói jogosultságok kezelése

#### **JSON Web Token (JWT)**

A felhasználói autentikációhoz és autorizációhoz JSON Web Token (JWT) technológiát használtunk. A JWT egy nyílt szabvány, amely lehetővé teszi a biztonságos információcsere a felek között. Főbb előnyei:

- Stateless: A JWT tokenek tartalmazzák az összes szükséges információt, így nincs szükség session tárolására a szerveren.
- Biztonságos: A tokenek digitálisan aláírtak, így biztosítható a tartalmuk integritása.
- Skálázható: A stateless természete miatt könnyen skálázható több szerver között.
- Rugalmas: Különböző jogosultsági szintek és felhasználói információk tárolhatók a tokenben.

A projektben a JWT-t a következő célokra használtuk:

- Felhasználói bejelentkezés kezelése
- API végpontok védelme
- Felhasználói jogosultságok ellenőrzése (pl. admin jogosultságok)
- Token alapú autentikáció biztosítása a frontend számára

### **Bcrypt.js**

A jelszavak biztonságos tárolásához Bcrypt.js könyvtárat használtunk, amely egy modern hash függvény kifejezetten jelszavak tárolására. Főbb előnyei:

- Sózás: Automatikus sózás a jelszavak védelméhez a rainbow table támadások ellen.
- Adaptív költség: Beállítható hash generálási költség, ami lehetővé teszi a biztonság és teljesítmény közötti egyensúly kialakítását.
- Jövőbiztos: A számítási kapacitás növekedésével a hash generálási költség növelhető.

### **Multer**

A fájlfeltöltések kezeléséhez Multer middleware-t használtunk, amely lehetővé tette a termékek feltöltését és tárolását. A Multer segítségével:

- Korlátoztuk a feltölthető fájlok típusát (csak képek)
- Beállítottuk a maximális fájlméretet
- Átneveztük a fájlokat a tároláshoz
- Strukturált mappákba rendeztük a feltöltött képeket

### **Nodemailer**

Az e-mail küldéshez Nodemailer könyvtárat használtunk, amely lehetővé tette az automatikus e-mail értesítések küldését a következő esetekben:

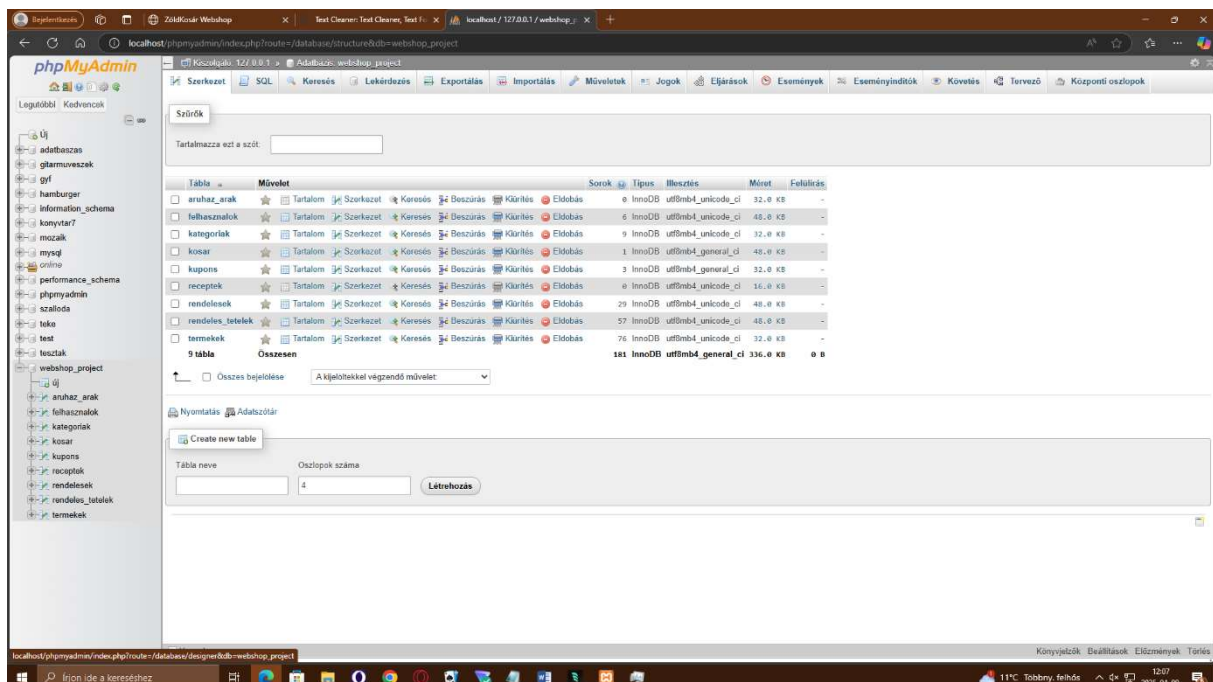
- Regisztráció megerősítése
- Jelszó visszaállítás
- Rendelés visszaigazolás
- Hírlevél küldés

## Adatbázis

### MySQL

Az adatok tárolásához MySQL relációs adatbázis-kezelő rendszert választottunk. A MySQL választását a következő előnyök indokolták:

- Megbízhatóság: A MySQL egy kipróbált, megbízható adatbázis-kezelő rendszer.
- Teljesítmény: Kiváló teljesítményt nyújt a webáruház típusú alkalmazásokhoz szükséges műveleteknél.
- Skálázhatóság: Lehetőség van a horizontális és vertikális skálázásra a növekvő adatmennyiség kezeléséhez.
- Tranzakció-kezelés: ACID (Atomicity, Consistency, Isolation, Durability) tulajdonságok biztosítása, ami különösen fontos a rendelések kezelésénél.
- Referenciális integritás: Idegen kulcs megszorítások támogatása, ami biztosítja az adatok konzisztenciáját.



### Sequelize ORM

Az adatbázis-műveletek absztrahálásához és egyszerűsítéséhez Sequelize ORM-et (Object-Relational Mapping) használtunk. A Sequelize egy Node.js alapú ORM, amely támogatja a MySQL, PostgreSQL, SQLite és más adatbázisokat. Főbb előnyei:

- Modellek definiálása: Lehetőség az adatbázis tábláinak JavaScript osztályokként való definiálására.
- Automatikus migrációk: Adatbázis séma változtatások kezelése kód alapján.
- Kapcsolatok kezelése: Egyszerű szintaxis a táblák közötti kapcsolatok definiálására (one-to-one, one-to-many, many-to-many).
- Lekérdezés építés: Komplex SQL lekérdezések építése egyszerű JavaScript szintaxissal.
- Validáció: Adatok validálása az adatbázisba írás előtt.
- Hooks: Életciklus hook-ok definiálása különböző adatbázis-műveletek előtt és után.

A projektben a Sequelize-t a következő célokra használtuk:

- Adatbázis modellek definiálása (felhasználók, termékek, kategóriák, rendelések, stb.)
- Kapcsolatok definiálása a modellek között
- Komplex lekérdezések végrehajtása (pl. termékek szűrése, rendezése)
- Tranzakciók kezelése a rendelési folyamat során
- Adatok validálása mentés előtt

## Rendszerarchitektúra

A ZöldKosár Webáruház egy modern, többrétegű architektúrával rendelkező webalkalmazás, amely a kliens-szerver modellt követi, és különböző technológiai komponensekből épül fel. Az alábbiakban részletesen bemutatjuk a rendszer architektúráját, a komponensek közötti kapcsolatokat és az adatáramlást.

### Architektúra áttekintése

A ZöldKosár Webáruház architektúrája három fő rétegből áll:

1. Frontend réteg: A felhasználói felületet biztosító React.js alapú Single Page Application (SPA).
2. Backend réteg: A Node.js és Express.js alapú RESTful API szerver, amely az üzleti logikát tartalmazza.
3. Adatbázis réteg: A MySQL adatbázis, amely az adatok perzisztens tárolását biztosítja.

Ez a háromrétegű architektúra biztosítja a rendszer modúlárisságát, skálázhatóságát és karbantarthatóságát. A rétegek közötti kommunikáció szabványos HTTP protokollon keresztül történik, JSON formátumú adatsomagokkal.

### Frontend architektúra

A frontend architektúra komponens-alapú megközelítést követ, amely a React.js könyvtár alapelveit tükrözi. A fő komponensek a következők:

### Komponens hierarchia

A React komponensek hierarchikus struktúrába szerveződnek:

- App komponens: A legfelső szintű komponens, amely tartalmazza a routing logikát és a globális állapotot.
- Oldal komponensek: Az egyes oldalakat reprezentáló komponensek (pl. Home, Products, Cart, Admin).
- Funkcionális komponensek: Újrafelhasználható komponensek, amelyek specifikus funkcionalitást biztosítanak (pl. ProductCard, CategoryFilter, Pagination).
- Közös komponensek: Az alkalmazás több részén használt általános komponensek (pl. Button, Modal, Notification).

## Állapotkezelés

Az állapotkezelés több szinten valósul meg:

- Lokális állapot: A React useState hook segítségével kezelt komponens-szintű állapot.
- Kontextus-alapú állapot: A React Context API segítségével kezelt globális állapot, amely több komponens számára elérhető.
- URL-alapú állapot: A React Router segítségével az URL-ben tárolt állapot (pl. keresési paraméterek, szűrők).

A főbb állapotkezelő kontextusok:

- useContext: A bejelentkezett felhasználó adatainak és autentikációs állapotának kezelése.
- CartContext: A bevásárlókosár tartalmának és a kapcsolódó műveletek kezelése.
- NotificationContext: A felhasználói értesítések kezelése.

## Routing

A routing a React Router segítségével valósul meg, amely lehetővé teszi a különböző oldalak közötti navigációt a böngésző történetének módosításával, de a teljes oldal újratöltése nélkül.

A fő route-ok:

- '/': Kezdőoldal
- '/products': Termékek listája
- '/login': Bejelentkezési oldal
- '/register': Regisztrációs oldal
- '/cart': Bevásárlókosár
- '/profile': Felhasználói profil
- '/admin/': Admin felület (beágyazott route-okkal)
- '/recipes': Receptkereső oldal
- '/about': Rólunk oldal
- '/privacy-policy': Adatvédelmi szabályzat

## API kommunikáció

A backend API-val való kommunikáció az Axios könyvtár segítségével valósul meg. A kommunikáció a következő módon strukturálódik:

- API szolgáltatás: Központosított API hívások az `api.js` fájlban.
- Axios konfiguráció: Alapértelmezett beállítások és interceptorok az `axiosConfig.js` fájlban.
- Hook-alapú adatlekérés: Custom hook-ok az adatok lekéréséhez és kezeléséhez.

## Backend architektúra

A backend architektúra a Node.js és Express.js alapú RESTful API szerver köré épül, amely a következő komponensekből áll:

### Szerverstruktúra

A szerver a következő fő komponensekből áll:

- Szerver konfiguráció: Az Express alkalmazás konfigurációja és middleware-ek beállítása.
- Routing: Az API végpontok definiálása és a megfelelő controller függvényekhez való irányítása.
- Controller réteg: Az üzleti logika implementációja, amely feldolgozza a kéréseket és előkészíti a válaszokat.
- Service réteg: Újrafelhasználható üzleti logika, amely a controller és a modell réteg között helyezkedik el.
- Modell réteg: Az adatbázis táblákat reprezentáló Sequelize modellek.
- Middleware-ek: Kérések feldolgozását végző köztes rétegek (pl. autentikáció, validáció).

## API végpontok

Az API végpontok RESTful elvek szerint szerveződnek, és a következő fő csoportokba sorolhatók:

- `/api/auth`: Autentikációs végpontok (regisztráció, bejelentkezés, jelszó visszaállítás).
- `/api/products`: Termékekkel kapcsolatos végpontok (listázás, részletek, keresés, szűrés).



- /api/categories: Kategóriákkal kapcsolatos végpontok.
- /api/cart: Bevásárlókosárral kapcsolatos végpontok.
- /api/orders: Rendelésekkel kapcsolatos végpontok.
- /api/users: Felhasználókkal kapcsolatos végpontok.
- /api/admin: Admin funkciókat biztosító végpontok.
- /api/upload: Fájlfeltöltési végpontok.
- /api/chat: Chat funkciókat biztosító végpontok.

## **Middleware-ek**

A backend számos middleware-t használ a kérések feldolgozásához:

- CORS middleware: Cross-Origin Resource Sharing kezelése.
- JSON parser: JSON formátumú kérések feldolgozása.
- Cookie parser: Cookie-k kezelése.
- Auth middleware: JWT token ellenőrzése és felhasználói jogosultságok kezelése.
- Error handling middleware: Hibakezelés és hibaüzenetek formázása.
- Multer middleware: Fájlfeltöltések kezelése.

## **Autentikáció és autorizáció**

Az autentikáció és autorizáció JWT (JSON Web Token) alapú megközelítést követ:

- Regisztráció: Felhasználói adatok validálása, jelszó hashelése és felhasználó létrehozása.
- Bejelentkezés: Felhasználó hitelesítése és JWT token generálása.
- Token ellenőrzés: Middleware a védett végpontok elérésének korlátozásához.
- Szerepkör-alapú hozzáférés: Admin és felhasználói jogosultságok elkülönítése.

## **Adatbázis architektúra**

Az adatbázis architektúra a MySQL relációs adatbázis-kezelő rendszerre épül, és a Sequelize ORM segítségével kapcsolódik a backend alkalmazáshoz.

## **Adatbázis modellek**

A fő adatbázis modellek a következők:

- Felhasznalo: Felhasználói adatok tárolása.
- Termek: Termékek adatainak tárolása.
- Kategoria: Termékkategóriák tárolása.
- Rendeles: Rendelések fejadatainak tárolása.
- RendelesTetelek: Rendelési tételek tárolása.
- Kosar: Bevásárlókosár tételeinek tárolása.
- Kupon: Kuponok adatainak tárolása.
- AruhazAr: Termékárak tárolása.
- Recept: Receptek tárolása.

## **Adatbázis kapcsolatok**

A modellek között a következő kapcsolatok állnak fenn:

- Kategoria - Termek: One-to-Many kapcsolat (egy kategóriához több termék tartozhat).
- Termek - AruhazAr: One-to-Many kapcsolat (egy termékhez több ár tartozhat).
- Felhasznalo - Rendeles: One-to-Many kapcsolat (egy felhasználóhoz több rendelés tartozhat).
- Rendeles - RendelesTetelek: One-to-Many kapcsolat (egy rendeléshez több tétel tartozhat).
- Termek - RendelesTetelek: One-to-Many kapcsolat (egy termék több rendelési tételben szerepelhet).
- Felhasznalo - Kosar: One-to-Many kapcsolat (egy felhasználóhoz több kosártétel tartozhat).
- Termek - Kosar: One-to-Many kapcsolat (egy termék több kosártételben szerepelhet).

## **Adatbázis indexek**

A teljesítmény optimalizálása érdekében a következő indexeket hoztuk létre:

- Elsődleges kulcsok minden táblán
- Idegen kulcs indexek a kapcsolódó mezőkön
- Keresési indexek a gyakran keresett mezőkön (pl. termék név, kategória)
- Egyedi indexek az egyedi mezőkön (pl. email cím)

## **Adatáramlás a rendszerben**

Az adatáramlás a rendszerben a következő módon történik:

1. Felhasználói interakció: A felhasználó interakcióba lép a frontend felülettel (pl. termék kosárba helyezése).
2. Frontend feldolgozás: A React komponens feldolgozza az interakciót és frissíti a lokális állapotot.
3. API kérés: A frontend API kérést küld a backend szervernek (pl. POST /api/cart).
4. Backend feldolgozás: A backend feldolgozza a kérést, végrehajtja az üzleti logikát és kommunikál az adatbázissal.
5. Adatbázis művelet: Az adatbázis végrehajtja a szükséges műveleteket (pl. új kosártétel beszúrása).
6. Válasz generálás: A backend generálja a választ és visszaküldi a frontendnek.
7. Frontend frissítés: A frontend frissíti a felhasználói felületet a kapott válasz alapján.

## **Rendszerkövetelmények**

A ZöldKosár Webáruház futtatásához a következő rendszerkövetelmények szükségesek:

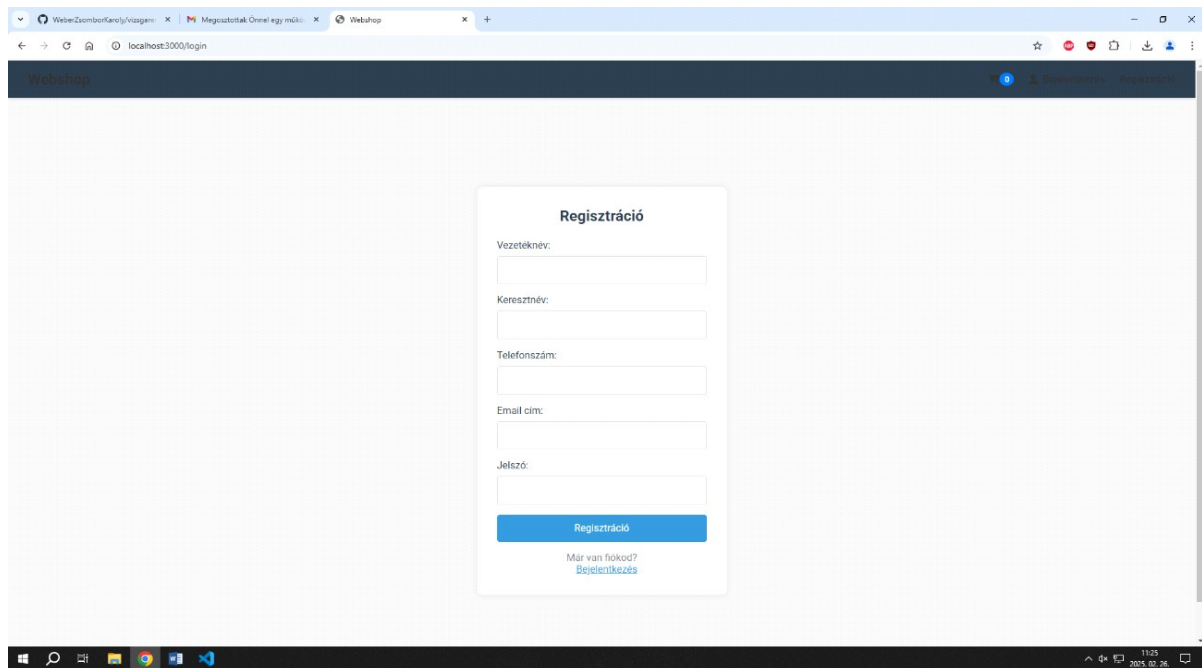
### **Frontend követelmények**

- Modern webböngésző (Chrome, Firefox, Safari, Edge legújabb verziói)
- JavaScript engedélyezve
- Minimum 1024x768 képernyőfelbontás (reszponzív design mobileszközökre is)

## Funkcionalitás

A ZöldKosár Webáruház számos funkcióval rendelkezik, amelyek együttesen biztosítják a teljes körű online vásárlási élményt. Az alábbiakban részletesen bemutatjuk ezeket a funkciókat, kategóriákra bontva.

## Felhasználói felület

A screenshot of a web browser showing the registration page of a webshop. The browser's address bar shows 'localhost:3000/login'. The page has a dark blue header with the word 'Webshop' on the left and a user profile icon on the right. The main content area is light gray and features a white registration form in the center. The form is titled 'Regisztráció' and contains five input fields: 'Vezetéknév:', 'Keresztnév:', 'Telefonszám:', 'Email cím:', and 'Jelszó:'. Below these fields is a blue button labeled 'Regisztráció'. Under the button, there is a link that says 'Már van fiókod? Bejelentkezés'. The Windows taskbar is visible at the bottom of the screen.

## Kezdőoldal

A kezdőoldal a webáruház belépési pontja, amely a következő elemeket tartalmazza:

- Üdvözlő banner: Figyelemfelkeltő banner, amely bemutatja a webáruház fő értékeit és aktuális akcióit.
- Kiemelt termékek: Válogatott termékek, amelyek különösen népszerűek vagy akciósak.
- Kategória navigáció: Gyors hozzáférés a fő termékkategóriákhoz.
- Blog szekció: Rövid kivonatok a legfrissebb blog bejegyzésekből, amelyek az egészséges táplálkozással, fenntarthatósággal és állatvédelemmel kapcsolatos témákat dolgoznak fel.
- Állatmenhely támogatás: Információk arról, hogyan támogatja a webáruház az állatmenhelyeket.
- Szerencsekerék: Interaktív játék, amellyel a felhasználók kedvezménykuponokat nyerhetnek.

A kezdőoldal reszponzív kialakítású, így különböző eszközökön (asztali számítógép, tablet, mobiltelefon) is optimális megjelenést biztosít.

### **Terméklistázó oldal**

A terméklistázó oldal a termékek böngészésére és szűrésére szolgál, és a következő funkciókat tartalmazza:

- Termékek megjelenítése: A termékek kártyás elrendezésben jelennek meg, képpel, névvel, árral és rövid leírással.
- Kategória szűrés: Lehetőség a termékek szűrésére kategória szerint.
- Keresés: Szabadszöveges keresés a termékek nevében és leírásában.
- Rendezés: Lehetőség a termékek rendezésére különböző szempontok szerint (ár, név, népszerűség).
- Lapozás: A termékek lapozható formában jelennek meg, oldalanként 12 termékkel.
- Kosárba helyezés: Lehetőség a termékek közvetlen kosárba helyezésére a listázó oldalról.
- Termék részletek: Lehetőség a termékek részletes adatainak megtekintésére.

### **Termék részletező oldal**

A termék részletező oldal egy adott termék részletes információit jeleníti meg:

- Termék képe: Nagyméretű kép a termékről.
- Termék adatai: Név, ár, kategória, leírás, készlet információk.
- Mennyiség választó: Lehetőség a kívánt mennyiség kiválasztására.
- Kosárba helyezés: Gomb a termék kosárba helyezéséhez.
- Kapcsolódó termékek: Hasonló vagy kiegészítő termékek ajánlása.
- Értékelések: Felhasználói értékelések és vélemények a termékről.

### **Navigáció**

A webáruház navigációs rendszere a következő elemekből áll:

- Fejléc: A webáruház logója, keresőmező, felhasználói menü és kosár ikon.
- Főmenü: Kategóriák, akciók, blog és egyéb fontos oldalak elérése.

- Lábléc: Kapcsolat, ÁSZF, adatvédelmi szabályzat, fizetési és szállítási információk.
- Breadcrumb navigáció: A felhasználó aktuális helyzetét mutató navigációs elem.
- Mobil menü: Hamburger menü a mobileszközökön való könnyebb navigációhoz.

## **Reszponzív design**

A webáruház rezponzív designnal rendelkezik, amely biztosítja az optimális megjelenést különböző eszközökön:

- Asztali nézet: Optimalizált megjelenés nagy képernyőkre (1200px felett).
- Tablet nézet: Átrendezett elemek közepes képernyőkre (768px - 1199px).
- Mobil nézet: Egyszerűsített elrendezés kis képernyőkre (767px alatt).
- Flexibilis képek: A képek automatikusan méretezik magukat a rendelkezésre álló helyhez.
- Adaptív betűméretek: A betűméretek alkalmazkodnak a képernyő méretéhez.

## **Vásárlási folyamat**

### **Termékek böngészése és keresése**

A termékek böngészése és keresése a következő módokon lehetséges:

- Kategória alapú böngészés: A termékek böngészése kategóriák szerint.
- Keresés: Szabadszöveges keresés a termékek között.
- Szűrés: Termékek szűrése különböző paraméterek alapján (ár, kategória, akciós termékek).
- Rendezés: Termékek rendezése különböző szempontok szerint (ár, név, népszerűség).

### **Kosárkezelés**

A kosárkezelés a következő funkciókat tartalmazza:

- Termék hozzáadása: Termékek hozzáadása a kosárhoz a kívánt mennyiségben.
- Mennyiség módosítása: A kosárban lévő termékek mennyiségének módosítása.
- Termék eltávolítása: Termékek eltávolítása a kosárból.
- Kosár tartalmának mentése: A kosár tartalma automatikusan mentésre kerül a felhasználói fiókban vagy cookie-ban.

- Kosár összesítés: A kosár tartalmának és végösszegének megjelenítése.
- Kupon beváltás: Kedvezménykuponok beváltása a kosárban.

## **Megrendelési folyamat**

A megrendelési folyamat a következő lépésekből áll:

1. Kosár áttekintése: A kosár tartalmának ellenőrzése és módosítása.
2. Bejelentkezés/Regisztráció: Bejelentkezés meglévő fiókba vagy új fiók létrehozása (vendég vásárlás is lehetséges).
3. Szállítási adatok megadása: Szállítási cím és kapcsolattartási adatok megadása.
4. Szállítási mód kiválasztása: Házhozszállítás vagy átvételi pont kiválasztása.
5. Fizetési mód kiválasztása: Online fizetés, banki átutalás vagy utánvét választása.
6. Rendelés áttekintése: A megrendelés adatainak ellenőrzése.
7. Rendelés véglegesítése: A megrendelés elküldése és visszaigazolás.

## **Fizetési módok**

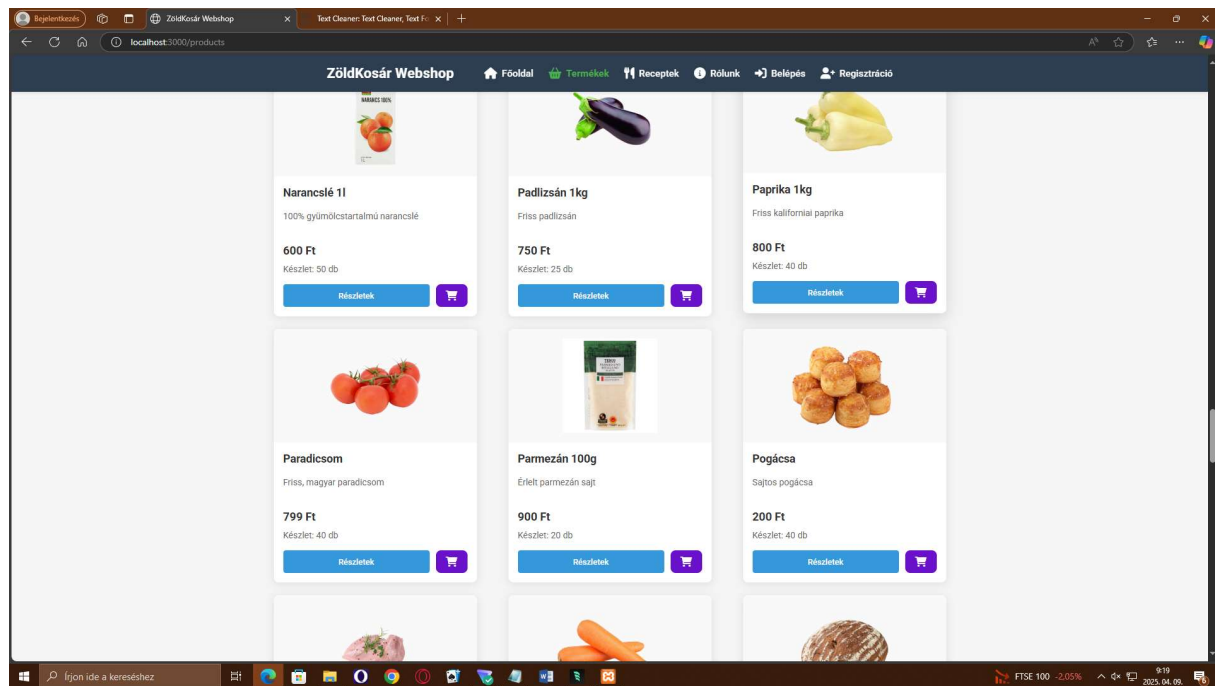
A webáruház a következő fizetési módokat támogatja:

- Online bankkártyás fizetés: Biztonságos online fizetés bankkártyával.
- Banki átutalás: Fizetés banki átutalással.
- Utánvét: Fizetés a csomag átvételekor.
- SZÉP kártya: Fizetés Széchenyi Pihenő Kártyával (vendéglátás zseb).

## **Szállítási módok**

A webáruház a következő szállítási módokat kínálja:

- Házhozszállítás: Kiszállítás a megadott címre.
- Bolti átvétel: Átvétel a webáruház fizikai üzletében.



## Felhasználókezelés

### Regisztráció

A regisztrációs folyamat a következő lépésekből áll:

1. Regisztrációs űrlap kitöltése: Alapvető adatok megadása (név, email, jelszó).
2. Email cím megerősítése: Megerősítő email küldése a megadott címre.
3. Fiók aktiválása: A fiók aktiválása a megerősítő emailben található linkre kattintva.
4. Profil kiegészítése: Opcionális adatok megadása (szállítási cím, telefonszám, stb.).

### Bejelentkezés

A bejelentkezési folyamat a következő módokon lehetséges:

- Email és jelszó: Bejelentkezés a regisztrált email cím és jelszó megadásával.
- Bejelentkezve maradás: Lehetőség a bejelentkezési állapot megőrzésére.

### Felhasználói profil

A felhasználói profil a következő funkciókat tartalmazza:



- Személyes adatok kezelése: Név, email, telefonszám módosítása.
- Jelszó módosítása: A fiókhoz tartozó jelszó megváltoztatása.
- Szállítási címek kezelése: Szállítási címek hozzáadása, módosítása, törlése.
- Rendelési előzmények: Korábbi rendelések megtekintése és nyomon követése.
- Kedvencek kezelése: Kedvenc termékek mentése és kezelése.
- Hírlevél feliratkozás: Feliratkozás és leiratkozás a hírlevélről.

## **Jogosultságkezelés**

A webáruház a következő felhasználói szerepköröket különbözteti meg:

- Vendég: Nem regisztrált felhasználó, korlátozott funkciókkal.
- Regisztrált felhasználó: Alapvető vásárlási és profilkezelési funkciók.
- Admin: Teljes hozzáférés az adminisztrációs felülethez és funkciókhoz.

## **Kosárkezelés**

### **Kosár műveletek**

A kosárkezelés a következő műveleteket támogatja:

- Termék hozzáadása: Termék hozzáadása a kosárhoz a kívánt mennyiségben.
- Mennyiség módosítása: A kosárban lévő termék mennyiségének növelése vagy csökkentése.
- Termék eltávolítása: Termék teljes eltávolítása a kosárból.
- Kosár ürítése: A teljes kosár tartalmának törlése.
- Kosár mentése: A kosár tartalmának automatikus mentése a felhasználói fiókban vagy cookie-ban.

### **Kosár perzisztencia**

A kosár tartalma a következő módokon kerül mentésre:

- Bejelentkezett felhasználók: A kosár tartalma az adatbázisban kerül mentésre, így különböző eszközökről is elérhető.

- Nem bejelentkezett felhasználók: A kosár tartalma localStorage-ban kerül mentésre, így a böngésző bezárása után is megmarad.
- Bejelentkezéskor: A nem bejelentkezett állapotban összeállított kosár automatikusan egyesül a felhasználó mentett kosarával bejelentkezéskor.

## **Kuponkezelés**

A kuponkezelés a következő funkciókat tartalmazza:

- Kupon beváltása: Kedvezménykupon beváltása a kosárban.
- Kupon érvényesség ellenőrzése: A kupon érvényességének és alkalmazhatóságának ellenőrzése.
- Kedvezmény számítása: A kupon által biztosított kedvezmény kiszámítása és alkalmazása.
- Kupon eltávolítása: A beváltott kupon eltávolítása a kosárból.

## **Rendeléskezelés**

### **Rendelés létrehozása**

A rendelés létrehozása a következő lépésekből áll:

1. Kosár véglegesítése: A kosár tartalmának ellenőrzése és véglegesítése.
2. Szállítási adatok megadása: Szállítási cím és kapcsolattartási adatok megadása.
3. Szállítási mód kiválasztása: A kívánt szállítási mód kiválasztása.
4. Fizetési mód kiválasztása: A kívánt fizetési mód kiválasztása.
5. Rendelés összesítése: A rendelés adatainak és végösszegének megjelenítése.
6. Rendelés elküldése: A rendelés véglegesítése és elküldése.

### **Rendelés nyomon követése**

A rendelés nyomon követése a következő funkciókat tartalmazza:

- Rendelés állapota: A rendelés aktuális állapotának megjelenítése (feldolgozás alatt, csomagolás, szállítás, kézbesítve).
- Rendelés részletei: A rendelés részletes adatainak megtekintése (termékek, mennyiségek, árak, szállítási és fizetési adatok).

- Szállítási információk: A szállítással kapcsolatos információk megjelenítése (várható kézbesítési idő, futár elérhetősége).
- Számla letöltése: A rendeléshez tartozó számla letöltése PDF formátumban.

### **Rendelés módosítása és lemondása**

#### A rendelés módosítása és lemondása a következő lehetőségeket kínálja:

- Rendelés módosítása: A rendelés adatainak módosítása a feldolgozás megkezdése előtt.
- Rendelés lemondása: A rendelés lemondása a feldolgozás megkezdése előtt.
- Kapcsolatfelvétel: Lehetőség a kapcsolatfelvételre a webáruház ügyfélszolgálatával a rendeléssel kapcsolatos kérdések esetén.

### **Email értesítések**

#### A rendelési folyamat során a következő email értesítések kerülnek kiküldésre:

- Rendelés visszaigazolás: Visszaigazoló email a rendelés beérkezéséről és adatairól.
- Fizetési visszaigazolás: Visszaigazoló email a sikeres fizetésről.
- Szállítási értesítés: Értesítés a rendelés szállításának megkezdéséről.
- Kézbesítési értesítés: Értesítés a rendelés sikeres kézbesítéséről.
- Rendelés módosítás: Értesítés a rendelés adatainak módosításáról.
- Rendelés lemondás: Értesítés a rendelés lemondásáról.

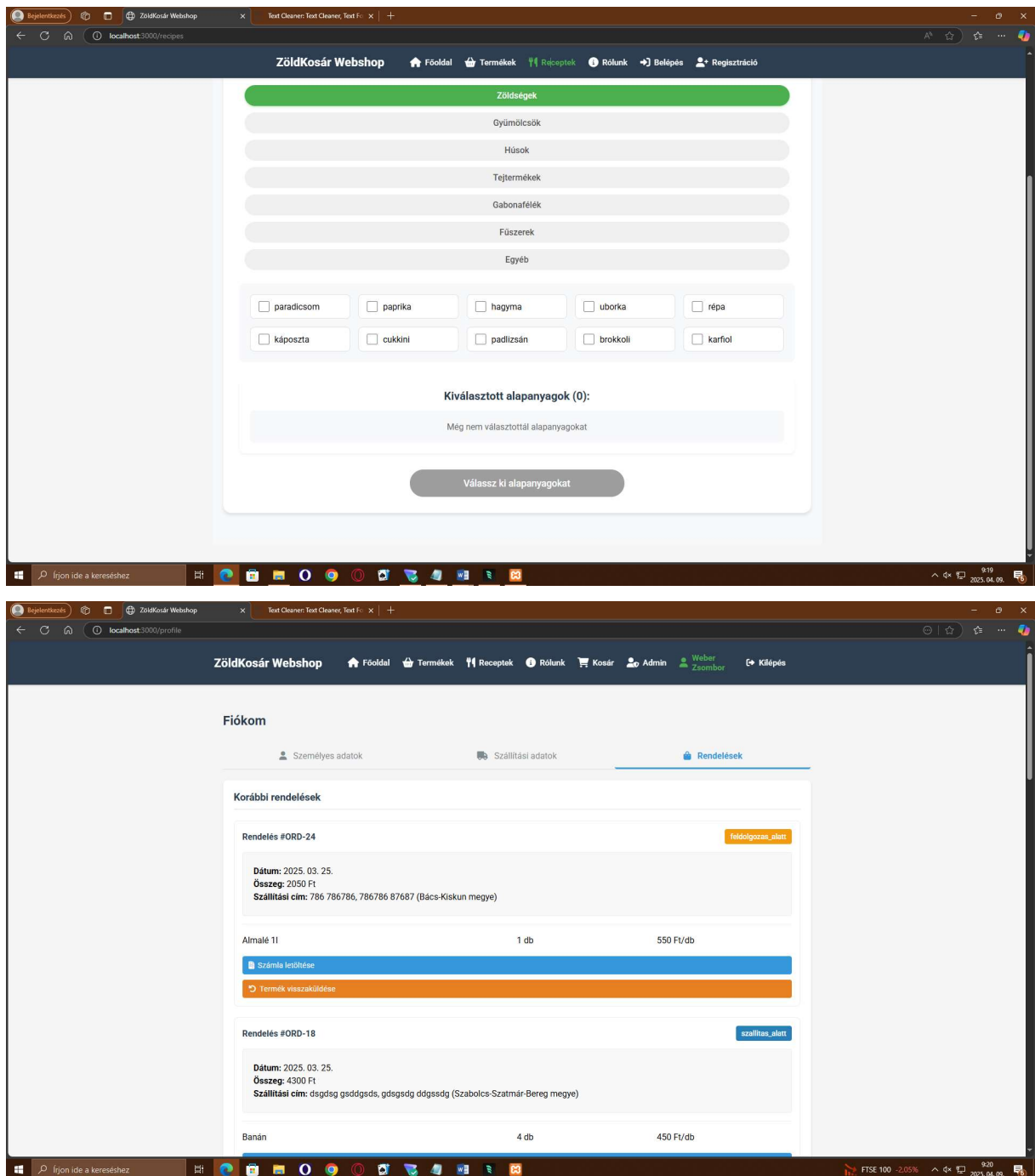
## Összefoglalás

A ZöldKosár Webáruház projekt sikeresen megvalósította a kitűzött célokat: egy modern, felhasználóbarát online élelmiszerboltot hoztunk létre, amely a fenntarthatóságra és a környezettudatosságra helyezi a hangsúlyt, miközben társadalmi felelősségvállalást mutat az állatmenhelyek támogatásával.

A projekt során számos modern webes technológiát alkalmaztunk, beleértve a React.js frontend keretrendszert, a Node.js/Express.js backend technológiákat és a MySQL adatbázist. A rendszer moduláris felépítésű, jól skálázható és könnyen karbantartható.

A webáruház teljes körű funkcionalitást biztosít a felhasználók számára, beleértve a termékek böngészését, a kosárkezelést, a megrendelési folyamatot és a felhasználói fiók kezelését. Az admin felület hatékony eszközöket biztosít a webáruház üzemeltetői számára a termékek, kategóriák, rendelések és felhasználók kezeléséhez.

A projekt során szerzett tapasztalatok és a létrehozott rendszer jó alapot biztosítanak a további fejlesztésekhez és bővítésekhez. A ZöldKosár Webáruház nemcsak egy egyszerű webshop, hanem egy olyan platform, amely ötvözi a modern e-kereskedelmi megoldásokat a környezettudatos szemlélettel és a társadalmi felelősségvállalással.



A weboldalunk nagyon felhasználóbarát könnyű eligazodni rajta mindenn korosztálynak és az adminoknak is nagyon könnyű kezelni a weboldalt egy pár kattintással bárhova el lehet jutni az oldalon hiba nélkül.

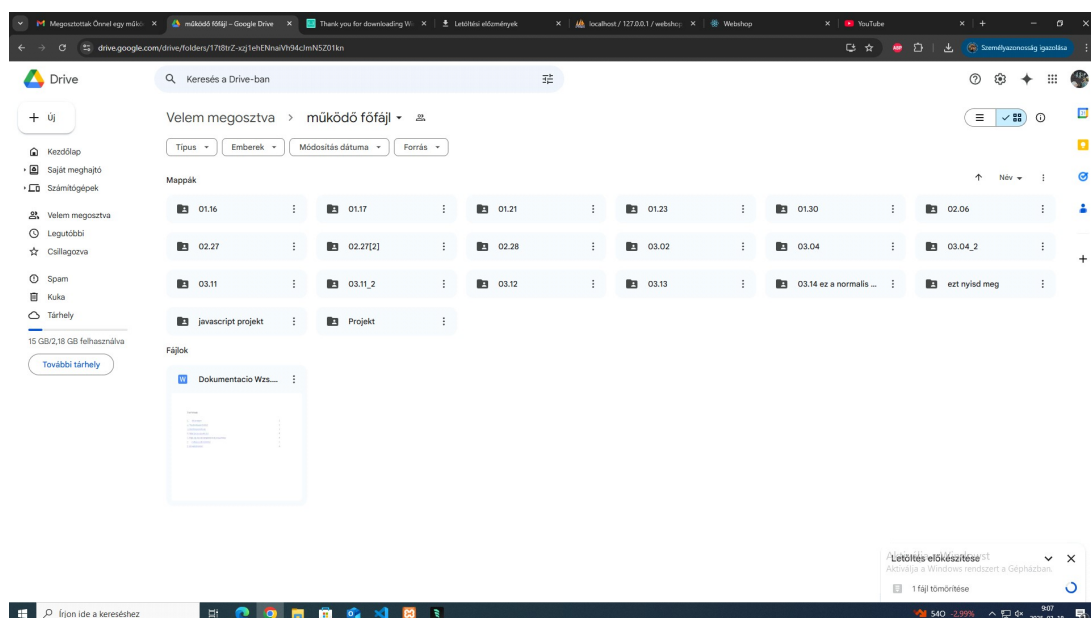
A regisztrációs felület lehetőséget biztosít az új felhasználók számára, hogy létrehozzák saját fiókjukat a webshopban. A folyamat során a felhasználók megadják alapvető adataikat, amelyeket a rendszer ellenőriz és tárol az adatbázisban.

Vezetéknév és Keresztnév - Az oldal ezek után a navbaron fogja megjeleníteni a be regisztrált felhasználó teljes nevét.

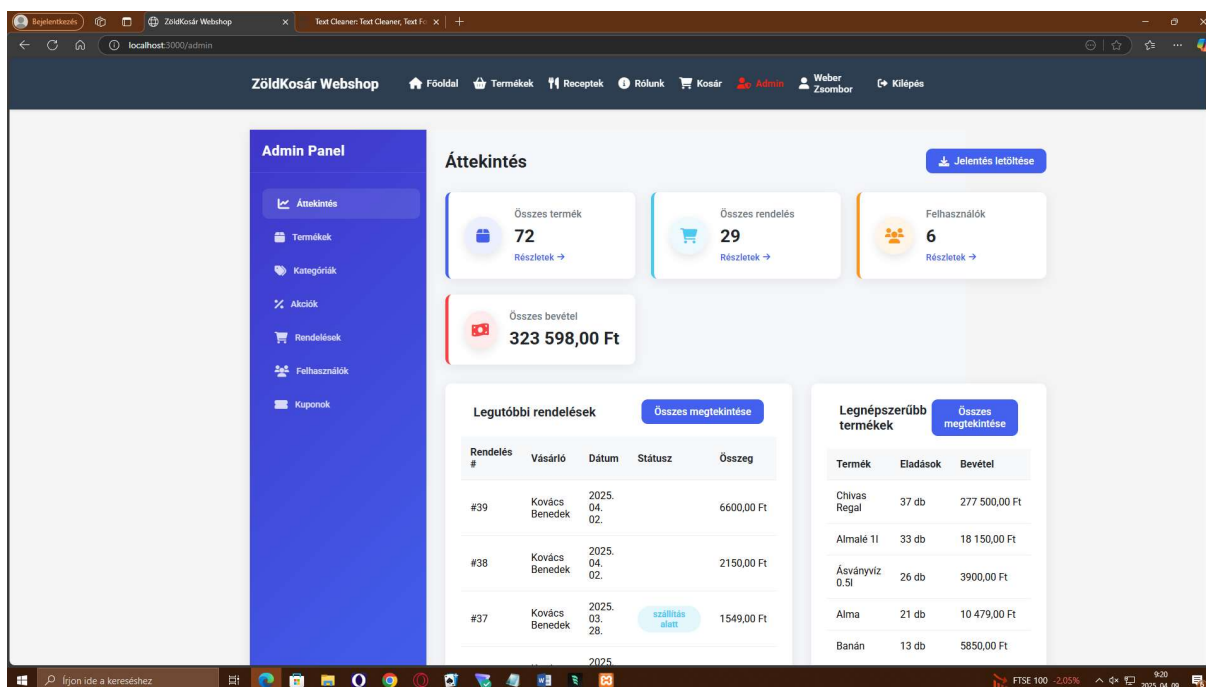
Telefonszám - Az adatbázis ezt elmenti. Ötlet volt, hogy egy 'elfelejtett jelszó'-s részt is csinálunk a közeljövőben hozzá, és ehhez szükséges lesz a telefonszáma az adott felhasználónak.

E-mail cím - a bejelentkezéshez szükséges adat.

Jelszó - megfelelően titkosítva.



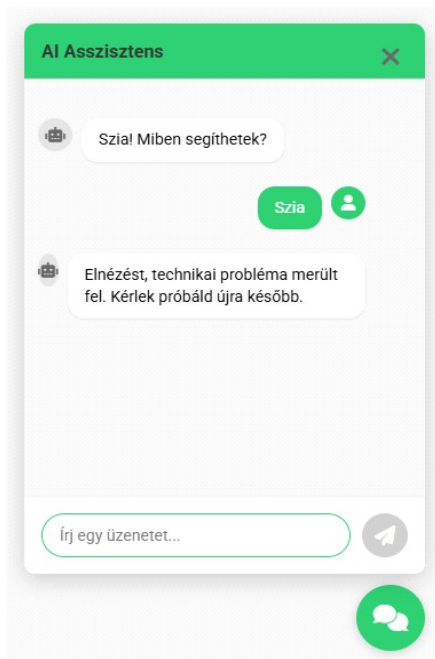
Többek között a Github mellett számunkra jobban bevált a Google Drive szolgáltatás. Mind a kettőnknek könnyebb volt ide menteni a projektet, amikor a suliban dolgoztunk rajta.



Az admin felület egy könnyen átlátható, letisztult oldal. Eléggé felhasználóbarát, így a hozzá kevésbé értő emberek is könnyen kiigazodnak rajta. Legnépszerűbb termékeknél úgy akartuk megcsinálni, hogy a legtöbbször kosárban lévő termékeket lisázza ki, de ezt nem tudtuk kivitelezni, ezért simán néhány terméket listáz ki. Az admin pannellel is sok gondunk volt, hiszen sehogysem akart jól működni. Általában a tokennel volt baj, vagy nem tudta az adatbázisból betölteni a dolgokat. Végül segítséggel, de sikerült megoldani a problémát, és teljesen jól működik az admin felület.

**Termékek** - Ezen a fülön tudjuk szerkeszteni a kiválasztott termék nevét, leírását, árát, készlet mennyiségét, kategóriáját, illetve a kép URL-jét. Új terméket is tudunk felvenni, amit elment adatbázisba a program, és kosárba helyezhető termékként fog megjelenni a products oldalon.

**Rendelések** - Ezen a fülön tudjuk kezelni a rendeléseket. Itt láthatjuk a rendelés ID-ját, dátumát, vásárló nevét és E-mailját, az összeget amennyiért vásárolt, illetve a státuszát. Manuálisan kell kezelni a rendeléseket ebben az alkalmazásban.



Az AI beépítése a projektbe egy hatalmas nehézséggel járó dolog volt, és ezzel tisztában voltunk, de ennek ellenére mi be akartuk építeni a programunkba. Nagyon sokat dolgoztunk vele, de végül nem sikerült megcsinálni, hogy jól működjön, így végül sajnálatos módon ez törlésre került.

## 11. Összefoglalás

A ZöldKosár Webáruház projekt sikeresen megvalósította a kitűzött célokat: egy modern, felhasználóbarát online élelmiszerboltot hoztunk létre, amely a fenntarthatóságra és a környezettudatosságra helyezi a hangsúlyt, miközben társadalmi felelősségvállalást mutat az állatmenhelyek támogatásával.

A projekt során számos modern webes technológiát alkalmaztunk, beleértve a React.js frontend keretrendszert, a Node.js/Express.js backend technológiákat és a MySQL adatbázist. A rendszer moduláris felépítésű, jól skálázható és könnyen karbantartható.

A webáruház teljes körű funkcionalitást biztosít a felhasználók számára, beleértve a termékek böngészését, a kosárkezelést, a megrendelési folyamatot és a felhasználói fiók kezelését. Az



admin felület hatékony eszközöket biztosít a webáruház üzemeltetői számára a termékek, kategóriák, rendelések és felhasználók kezeléséhez.

A projekt során szerzett tapasztalatok és a létrehozott rendszer jó alapot biztosítanak a további fejlesztésekhez és bővítésekhez. A ZöldKosár Webáruház nemcsak egy egyszerű webshop, hanem egy olyan platform, amely ötvözi a modern e-kereskedelmi megoldásokat a környezettudatos szemlélettel és a társadalmi felelősségvállalással.