

# **Fejlesztői dokumentáció**

**Szoftverfejlesztő és –tesztelő technikus szakma**

**2025**

**Készítette: Sövény Benjámin, Tóth Dániel**

# Tartalomjegyzék

Tartalomjegyzék .....	2
Bevezetés .....	6
Technológiák és készítéshez használt programok .....	7
Backend .....	7
Frontend .....	7
Adatbázis .....	7
Kódolási környezet .....	8
Webböngészők .....	8
Tesztelés .....	9
Felhőalapú média-kezelő szolgáltatás .....	9
E-mail kiszolgáló .....	9
Kártyás fizetés .....	10
Deploy .....	10
Kódolási konvenciók .....	11
Alapkönyvtárak .....	11
Frontend .....	11
Backend .....	13
Admin .....	14
Felülettervek .....	15
Adatmodell .....	15
Végpontok: Összes fájl .....	16
Backend .....	16
Végpontok: cartRoute.js .....	16
Végpontok: orderRoute.js .....	16
Végpontok: productRoute.js .....	17
Végpontok: userRoute.js .....	17
Általános működés .....	18
Osztályok .....	18
Frontend .....	18
Komponensek .....	19

BestSeller.jsx-----	19
CartTotal.jsx-----	19
Footer.jsx -----	19
Hero.jsx-----	19
LatestCollection.jsx -----	19
Navbar.jsx-----	19
NewsletterBox.jsx -----	20
Policy.jsx -----	20
ProductItem.jsx-----	20
RelatedProducts.jsx -----	20
ScrollToTop.jsx-----	20
SearchBar.jsx -----	20
Title.jsx -----	20
Kontextus -----	21
ShopContext.jsx -----	21
Állapot kezelés: -----	21
Kosár Műveletek: -----	21
Kosár Számítások: -----	21
Backend Kommunikáció: -----	21
Felhasználói bejelentkezés és kijelentkezés: -----	22
Pages/Oldalak -----	22
About.jsx -----	22
Cart.jsx -----	22
Contact.jsx -----	22
DataProtection.jsx -----	22
Delivery.jsx -----	22
Home.jsx -----	22
Login.jsx -----	23
Orders.jsx -----	23
PlaceOrder.jsx -----	23
Product.jsx-----	24
Products.jsx -----	24

ResetPassword.jsx-----	24
Verify.jsx-----	24
Backend -----	25
Controller-ek: -----	25
cartController.js: -----	25
orderController.js: -----	25
productController.js: -----	25
userController.js:-----	25
Middleware-k:-----	25
adminAuth.js: -----	25
auth.js:-----	26
multer.js:-----	26
Modell-ek: -----	26
orderModel.js: -----	26
productModel.js: -----	26
userModel.js:-----	26
Route-ok: -----	26
cartRoute.js:-----	26
orderRoute.js:-----	26
productRoute.js: -----	27
userRoute.js: -----	27
Admin -----	28
Komponensek -----	28
Login.jsx-----	28
Navbar.jsx-----	28
Sidebar.jsx-----	28
ScrollToTop.jsx-----	28
Pages/Oldalak -----	29
Add.jsx-----	29
List.jsx -----	29
ManageUsers.jsx -----	29
Orders.jsx -----	30

PopUp.jsx -----	30
Update.jsx -----	30
Tesztelés -----	31
Ismert hibák és hiányosságok -----	55
Felhasználó oldali hibák-----	55
Admin oldali hibák-----	55
Feljesztési lehetőségek -----	56

# Bevezetés

A projekt egy esküvői csokrokat és egyéb, kerti és dekorációs növényeket árusító magán vállalkozásnak készült, amire az igény az ismeretségi körünkön belül jelentkezett. A cél az volt, hogy egy olyan komplett webshopot készítsünk, amiben van egyaránt felhasználói és admin felület is. A felhasználók szemszögéből a legfontosabb és egyben elvárt funkció az volt, hogy az oldalon lévő termékek között lehessen válogatni, kategóriák és típusok szerint egyben lehetőség nyíljon a rendelésre, az üzlet és meghirdetett termékek adatainak a megismerésére. Az Admin oldali funkciók lényege az volt, hogy termékeket lehessen feltölteni, szerkeszteni, a felhasználókat és a rendeléseket kezelni.

A Virág Webshop projektünkben olyan alap funkcióknak kellett megvalósulniuk, mint a CRUD műveletek (Create, Read, Update, Delete) amelyek működése egy MongoDB alapú adatbázis segítségével zajlik. A Webshop egy Mern Stack Developing néven elhíresült módszerrel lett létrehozva, ahol a backend Node.js keretrendszer alapján, az adatbázis MongoDB, a frontend pedig React keretrendszerrel dolgozik és működik. A módszer kiválasztásának fő oka a fejlesztő környezetek nyújtotta egyszerűség, gyorsaság, kompaktság és az ezen rendszer által alkotható modern szép kinézetű weboldalakról eredeztethető. Fejlesztésünk során ezen előfeltevéseink be is bizonyosodtak. Arra jöttünk rá, hogy a főként JavaScript alapú rendszerekkel való munka tényleg egy sokkal egyszerűbb és modernebb fejlesztési környezetet adott.

A projektünkhöz sok segítséget sikerült találni különböző weboldalakról, így, ha időközben bármilyen probléma is ütköztünk, akkor általában sikerült olyan módszereket találni amiket felhasználva viszonylag gyorsan orvosolni sikerült az adott problémát.

# Technológiák és készítéshez használt programok

## Backend

A **Node.js** egy aszinkron JavaScript futtatókörnyezet, amely lehetővé teszi a szerveroldali fejlesztést. Az **Express.js** pedig egy könnyen használható webes keretrendszer, amely egyszerűsíti a webalkalmazások és API-k fejlesztését. Együtt gyors és hatékony megoldásokat kínálnak.



## Frontend

A **React** egy népszerű JavaScript könyvtár, amelyet a felhasználói felületek (UI) építésére használnak. Komponens-alapú megközelítést alkalmaz, lehetővé téve az újrafelhasználható UI elemek létrehozását. A React gyors renderelést biztosít a virtuális DOM használatával, ami növeli az alkalmazások teljesítményét. Kiválóan alkalmas dinamikus, interaktív webalkalmazások fejlesztésére.



## Adatbázis

### MongoDB

A **MongoDB** egy dokumentum-orientált NoSQL adatbázis, amely rugalmas és skálázható tárolást kínál. JSON-szerű adatstruktúrával dolgozik, így könnyen kezelhetők a változó adatmodellek. Nagyszerű választás nagy mennyiségű adat kezelésére és dinamikus alkalmazásokhoz.

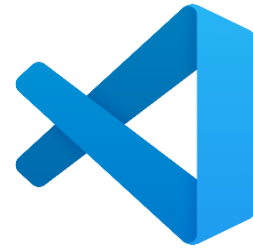


## Kódolási környezet

### Microsoft Visual Studio Code

A **Visual Studio Code** egy ingyenes, gyors és testreszabható kódszerkesztő, amely számos nyelvet támogat. Hibakeresést, verziókezelést és bővítményeket kínál.

Ideális webfejlesztőknek és szoftverfejlesztőknek.



## Webböngészők

### Google Chrome

A **Google Chrome** egy gyors, fejlesztők számára számos hasznos eszközt kínáló böngésző. Beépített fejlesztői eszközei (DevTools) lehetővé teszik a kód hibakeresését, a teljesítmény optimalizálását és a webalkalmazások tesztelését. Támogatja a legújabb webes technológiákat és könnyen bővíthető különböző kiegészítőkkel.




### Firefox

A **Firefox** egy fejlesztők számára is kiemelkedő böngésző, amely erőteljes beépített fejlesztői eszközöket (DevTools) kínál. A JavaScript hibakeresés, a CSS szerkesztés és a teljesítményfigyelés könnyedén elérhetők benne. Támogatja a legújabb webes szabványokat, és nagy hangsúlyt fektet a biztonságra és a magánélet védelmére.






## Tesztelés

<p><b>Postman</b></p> <p>A <b>Postman</b> egy népszerű API-fejlesztő eszköz, amely lehetővé teszi API-k tesztelését, dokumentálását és monitorozását. Könnyen készíthetők HTTP kérések, és az eszköz segítségével ellenőrizhetők a válaszok és hibák. Támogatja az automatizált tesztelést és a munkafolyamatok optimalizálását API fejlesztéskor.</p>	
<p><b>Insomnia</b></p> <p>Az <b>Insomnia</b> egy API-fejlesztő eszköz, amely lehetővé teszi RESTful és GraphQL API-k könnyű tesztelését és hibakeresését. Támogatja a HTTP kérések gyors küldését, válaszok megjelenítését és a környezeti változók kezelését. Felhasználóbarát felülete és fejlett funkciói segítik az API fejlesztést és integrációt.</p>	

## Felhőalapú média-kezelő szolgáltatás

<p><b>Cloudinary</b></p> <p>A <b>Cloudinary</b> egy felhőalapú média kezelő szolgáltatás, amely lehetővé teszi képek és videók tárolását, optimalizálását és kezelését. Automatikusan méretezhetők, formátumra konvertálhatók és tömöríthetők a fájlok. Fejlesztők számára API-kat kínál a médiafájlok egyszerű integrálásához web- és mobilalkalmazásokba.</p>	
---	--

## E-mail kiszolgáló

<p><b>G-mail</b></p> <p>A <b>Gmail API</b> lehetővé teszi a fejlesztők számára, hogy integrálják a Gmail szolgáltatásait saját alkalmazásaikba. Az API segítségével automatizálhatók az e-mailek küldése, olvasása, címkék kezelése és a fiók kezelésének különböző aspektusai. Biztonságos hozzáférést biztosít OAuth2 autentikációval, és rugalmas lehetőségeket kínál a Gmail fiókokkal való interakcióra.</p>	
---	--

## Kártyás fizetés

### Stripe

A **Stripe** egy fizetési platform, amely egyszerűsíti online tranzakciók kezelését web- és mobilalkalmazásokban. Támogatja a kártyás fizetéseket, előfizetéseket és egyéb módszereket. Az API-k könnyen integrálhatók és biztonságosak.



## Deploy

### Vercel

A **Vercel** egy felhőalapú platform, amely lehetővé teszi webalkalmazások és statikus oldalak gyors telepítését és skálázását. Különösen jól integrálódik a Next.js-el, de más front-end keretrendszerekkel is használható. Az automatikus deploy, a gyors betöltési idő és a globális CDN segíti az alkalmazások teljesítményét.



# Kódolási konvenciók

A kódot Git verziókezelő rendszerrel kezeltük, amely lehetővé tette a fejlesztés során történő változások nyomon követését, visszakereshetőségét és a hatékony csapatmunkát. A projekt során következetesen alkalmaztuk a Clean Code alapelveit, amelynek célja az átlátható, olvasható és karbantartható kód írása volt. Külön figyelmet fordítottunk a változónévadásra, a függvények egyszerűségére és a redundancia elkerülésére és külön magyarázattal is láttunk el egyes részeket a kódban, hogy könnyebben érthetőbb legyen a fejlesztés menete. Igyekeztünk, hogy a kód minden egyes része világosan tükrözze annak célját, így esetleg, ha más külső fejlesztő akarna dolgozni a projekten akkor könnyen megértheti és módosíthatja azt. Rendszeresen végeztünk kódfelülvizsgálatot, hogy biztosítsuk a minőségi követelmények betartását és a közös kódstílus érvényesülését főleg fejlesztés közben figyeltünk arra, hogy minden egyes módosítás, amit a kódban írunk az működjön még mielőtt a következő lépést kezdtük volna meg, hozzájárulva a megbízható és stabil működéshez. A célunk az volt, hogy a kódunk nemcsak technikailag legyen működőképes, de hosszú távon is fenntartható legyen.

## Alapkönyvtárak

### Frontend

- |— App.jsx
  - |— /src
    - | |— /components
      - | | |— BestSeller.jsx
      - | | |— CartTotal.jsx
      - | | |— Footer.jsx
      - | | |— Hero.jsx
      - | | |— LatestCollection.jsx
      - | | |— Navbar.jsx
      - | | |— NewsletterBox.jsx
      - | | |— Policy.jsx
      - | | |— ProductItem.jsx
      - | | |— RelatedProducts.jsx
      - | | |— SearchBar.jsx
      - | | |— ScrollToTop.jsx
      - | | |— Title.jsx

- | | — /context
- | | | — ShopContext.jsx
- | | — /pages
- | | | — About.jsx
- | | | — Cart.jsx
- | | | — ResetPassword.jsx
- | | | — Contact.jsx
- | | | — DataProtection.jsx
- | | | — Delivery.jsx
- | | | — Home.jsx
- | | | — Login.jsx
- | | | — Orders.jsx
- | | | — PlaceOrder.jsx
- | | | — Product.jsx
- | | | — Products.jsx
- | | | — Title.jsx
- | | | — Verify.jsx
- | | — /assets
- | | | — assets.js
- | | | — about\_img.png
- | | | — bin\_icon.png
- | | | — cart\_icon.png
- | | | — contact\_img.png
- | | | — cross\_icon.png
- | | | — dropdown\_icon.png
- | | | — exchange\_icon.png
- | | | — hero\_img.png
- | | | — logo.png
- | | | — menu\_icon.png
- | | | — profile\_icon.png

- | | |— quality\_icon.png
- | | |— search\_icon.png
- | | |— star\_dull\_icon.png
- | | |— star\_icon.png
- | | |— stripe\_logo.png
- | | |— support\_img.png

## Backend

- |— /config
  - | |— cloudinary.js
  - | |— mongodb.js
- |— /controllers
  - | |— cartController.js
  - | |— orderController.js
  - | |— productController.js
  - | |— userController.js
- |— /middleware
  - | |— adminAuth.js
  - | |— auth.js
  - | |— multer.js
- |— /models
  - | |— orderModel.js
  - | |— productModel.js
  - | |— userModel.js
- |— /routes
  - | |— cartRoute.js
  - | |— orderRoute.js
  - | |— productRoute.js
  - | |— userRoute.js

## Admin

- |— /src
  - | |— /pages
    - | | |— Add.jsx
    - | | |— List.jsx
    - | | |— Orders.jsx
    - | | |— Update.jsx
    - | | |— PopUp.jsx
    - | | |— ManageUsers.jsx
- |— /components
  - | |— Login.jsx
  - | |— Navbar.jsx
  - | |— ScrollToTop.jsx
  - | |— Sidebar.jsx
- |— /assets
  - | |— add\_icon.png
  - | |— assets.js
  - | |— logo.png
  - | |— order\_icon.png
  - | |— parcel\_icon.svg
  - | |— upload\_area.png
  - | |— user\_icon.png

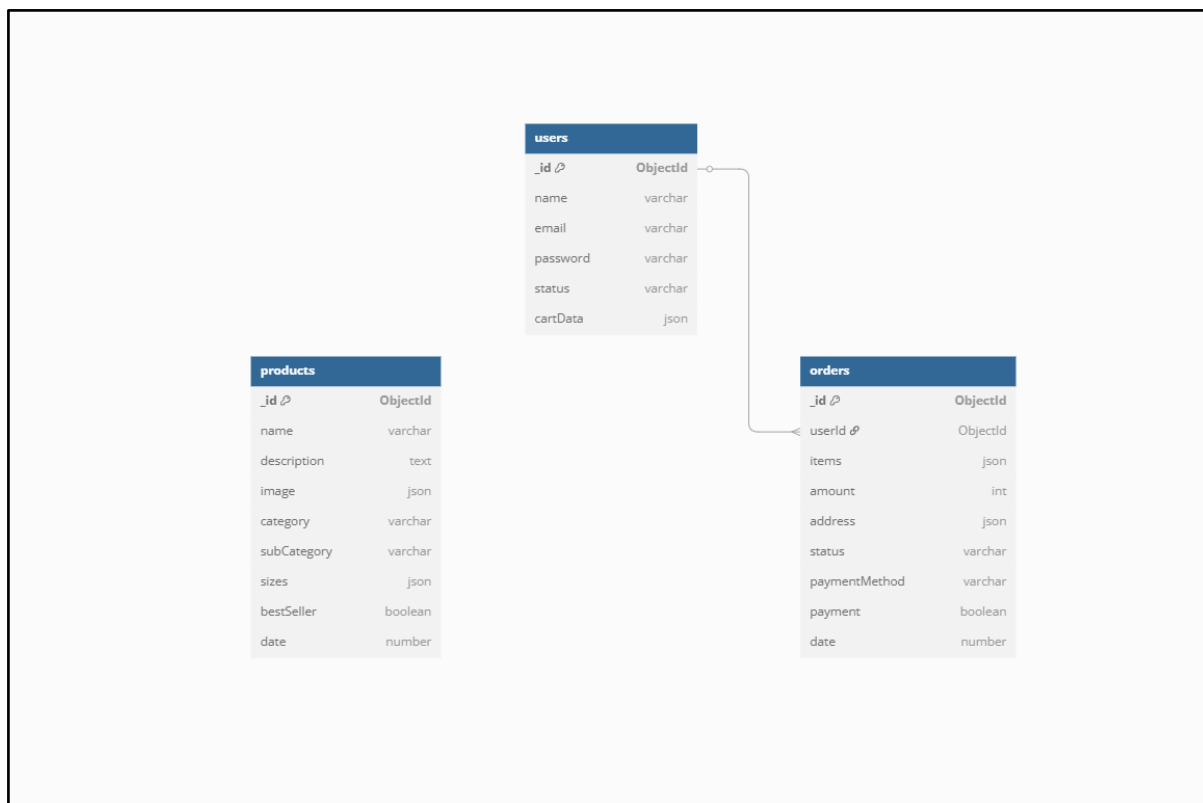
# Felülettervek

## Adatmodell

Az adatbázis a MongoDB nevű rendszerrel készült, ami egy modern, rugalmas adatkezelési megoldás. Itt az adatokat nem hagyományos táblázatokban tároljuk, mint egy Excel-ben, hanem úgynevezett dokumentumokban – hasonlóan ahhoz, ahogy egy Word-fájl tartalmazhat többféle információt egyszerre. Ez azt jelenti, hogy minden rendelés, felhasználó vagy termék külön kis csomagban van elmentve, és ezek a csomagok könnyen kezelhetők, kereshetők és bővíthetők. Ez a módszer különösen jól működik olyan webáruházaknál, ahol sokféle adatot kell gyorsan és hatékonyan tárolni.



*A kép a dbdiagram.io által készült*



*A kép a dbdiagram.io által készült*

## Végpontok: Összes fájl

### Backend

- ├── /routes
  - | ├── cartRoute.js
  - | ├── orderRoute.js
  - | ├── productRoute.js
  - | └── userRoute.js

### Végpontok: cartRoute.js

Végpont	Metódus	Azonosítás	Leírás
/get	post	authUser	getUserCart
/add	post	authUser	addToCart
/update	post	authUser	UpdateCart

### Végpontok: orderRoute.js

Megjegyzés	Végpont	Metódus	Azonosítás	Leírás
------------	---------	---------	------------	--------



Admin funkciók	/list	post	adminAuth	allOrders
	/status	post	adminAuth	updateStatus
Fizetési funkciók	/place	post	authUser	placeOrder
	/stripe	post	authUser	placeOrderStripe
Felhasználó lehetőségei	/userorders	post	authUser	userOrders
Fizetési authentikáció	/verifyStripe	post	authUser	verifyStripe

### Végpontok: productRoute.js

Végpont	Metódus	Azonosítás	Leírás
/add	post	adminAuth	addProduct
/remove	post	adminAuth	removeProduct
/list	get	nincs	listProducts
/single/:productId	get	nincs	singleProduct
/update	put	adminAuth	updateProduct

### Végpontok: userRoute.js

Végpont	Metódus	Azonosítás	Leírás
/register	post	nincs	registerUser
/login	post	nincs	loginUser
/admin	post	nincs	adminLogin

# Általános működés

A REST API http kéréseket fogad, melyek tartalmazzák a műveletekhez szükséges megfelelő adatokat. A kényes műveletek végpontjai védettek, autentikáció szükséges a használatukhoz. Az adatokat JSON formátumban fogadja és feldolgozza. A vezérlést kontrollerek valósítják meg, minden adatkezelési csoportnak külön kontrollere van, itt történik az adatfeldolgozás.

A kontrollerek modellekkel vannak kapcsolatban, amelyek az adatkezelésért felelősek. Minden adatkezelési csoportnak külön modellje van, itt történik az adatok adatbázisból kiolvasása, illetve az adatok kiírása adatbázisba.

A modellek adatbázis táblákkal vannak kapcsolatban, melyek az adatok tárolásáért felelősek. Az adatbázis táblák adatait a modellek kezelik.

## Osztályok

### Frontend

A frontend felület az az oldal, amit a felhasználók látnak. Az Admin oldali működés is nagyon hasonló képen van felépítve mint a Frontend mivel ott és itt is találhatóak komponensek és oldalak. Lényegében a komponensek olyan elemek, amiket könnyen lehet importálni az oldalakra. Ezt a módszer azért láttuk jobbnak mert így nem kellett minden egyes oldalon külön saját kinézetet értem itt azt, hogy minden oldalon a Footert megírni vagy a Headert hanem egyszerűen elég volt csak megírni egyszer és csak egy hivatkozás alapján az oldalakra importálni. A módszer ihletét számomra a Wordpress-ben is általában használatos Elementor plugin adta, aminek működése nyilván sokkal egyszerűbb mert a felhasználó úgy építhet oldalt, hogy egyes komponensekből építheti össze az oldalát hasonló képen, mint ha egy Lego várat akarna építeni. Én ezt a módszert akartam volna valahogy átültetni a projektbe és így jött az az ötlet, hogy sokkal jobb lenne, ha lennének olyan komponensek, amiket szabadon bele lehet építeni az egyes oldalakra így megkönnyítve a munkát és rengeteg időt és energiát megtakarítva sokkal átláthatóbb fejlesztői környezetet alkotva. Természetes törekedtünk, hogy minden egyes komponens reszponzívan a használt készülékhez képest megfelelően jelenjen meg.

## **Komponensek**

### **BestSeller.jsx**

A komponens lényege, hogy ide kerülnek azok a termékek, amiket az Admin felkapottnak jelölt a termék hozzáadásakor. A komponens csak a menü oldalon a /home útvonalon található. A célja, hogy tájékoztassa a felhasználókat, hogy mik azok a termékek, amik felkapottak.

### **CartTotal.jsx**

A komponens a placeOrder és a Cart oldalon van jelen. A lényege, hogy a kosárba került termékek összességét és a hozzáadott szállítási költséget mutatja meg. A komponens olyan tekintetben lehetne fejleszthető, hogy a szállítási érték dinamikusan változzon attól függően, hogy milyen szállítási módot választ a felhasználó, de jelenleg csak személyes átvétel létezik az oldalon ezért nyilván mindig nulla értéket fog mutatni.

### **Footer.jsx**

A footer komponens az összes oldalon megtalálható. Lényege amellet, hogy minden egyes weboldalnál szinte elvárt elem és a kinézethez is nagyban hozzájárul, hogy az oldal alján található kiegészítő információkat megtalálják a felhasználók, mint a szállítás, adatvédelmi tájékoztató és a kapcsolat felvételi elérhetőségek.

### **Hero.jsx**

A felhasználó ezzel a komponenssel találkozik először amikor az oldalra látogat mivel csak a menü oldalon található meg (/home) a funkciója inkább látványelem mint funkcionalitás. Itt csak egy képet lát a felhasználó és egy címet, amin a Vásárlás gombra kattintva átnavigál a termékek oldalra.

### **LatestCollection.jsx**

A komponens a legfrissebben feltöltött termékeket mutatja meg és abból is csak az első tíz darab terméket. A komponens hasonló képen, mint a Hero.jsx csak a menü (/home) oldalon található. A termékeket balról jobbra sorolja fel baloldalt a legújabb termékek vannak és onnantól jobbra haladva a feltöltés szerint a régebbieket mutatja meg.

### **Navbar.jsx**

A Navbar az összes oldalon alapelem. Itt vannak mondhatni az irányításra szolgáló információk, mint hogy a termékek oldalt a kosarat a rendeléseket is ebben a komponensben lehet megnyitni. A termékek (/products) oldalon megjeleníthető keresés mező ikonja is itt található. A fő funkciója, hogy a felhasználót elirányítsa a kívánt oldalakra. Az webshop ikonjára való kattintással a menübe irányít vissza.

### **NewsLetterBox.jsx**

A komponens jelenleg csak látvány elem mivel nem működik, de a koncepciója az, hogy ha valaki beírja az e-mail címét akkor arra kap időszakosan olyan e-maileket, amivel tájékoztatják az esetleges eseményekről vagy akciókról. A feliratkozott e-mail címeket az admin láthatná a saját oldalán és onnan írhatna e-maileket és küldhetné el az összes felhasználónak, de nyilván ez még egy nagyobb fejlesztési lehetőség lenne. A komponens a menü, rólunk, szállítás és a kapcsolat oldalon található meg.

### **Policy.jsx**

A komponens csak a menüben a (/home) oldalon található meg és szimplán csak tájékoztató funkciója van. A lényege, hogy tájékoztassa a felhasználókat, hogy milyen szolgáltatásokat nyújt a cég a termékek megvásárlásakor.

### **ProductItem.jsx**

A komponens a termékek egységes megjelenéséért felel, ami a (/product) termék, a menü (/home), termékek (/products) oldalon és a RelatedProducts, LatestCollection és a BestSeller komponensekben is megtalálható. Itt a termék első képe az ára és a termék neve található meg és célja az egységes termékek megjelenítése a felhasználó számára.

### **RelatedProducts.jsx**

Ez a komponens a termék oldalon (/product) található és az adott éppen megnyitott termékhez hasonló termékek javaslatáért felel. A javaslatokat az alapján adja meg hogy az éppen megnyitott termék milyen kategóriába és típusba tartozik és azt figyelve az összes eddig feltöltött termékből ajánl hasonlókat.

### **ScrollToTop.jsx**

Ez a komponens azért felel, hogy az éppen használt oldalak mindig az oldal elejétől induljon és oda töltsön be ne pedig az adott oldalról való átlépésnél hagyott magasságtól nyissa meg a következő oldalt. A komponens egy hibát hivatott kijavítani és erre a célra lett létrehozva.

### **SearchBar.jsx**

A SearchBar komponens csak a termékek oldalon (/products) oldalon található mert a lényege az, hogy a felhasználók a termékek oldalon meg tudják keresni azokat vagy azt az egy terméket, amit meg akarnak vásárolni. A termékek keresésénél csak a termék nevét figyeli. Az összes többi oldalon le van tiltva ez a komponens.

### **Title.jsx**

A komponens lényege, hogy minden egyes oldalon található címek, mint a /products oldalon a *minden termék* a /about oldalon a *rólunk* felirat és többi helyen is hasonlóan egységesen nézzenek ki. Mondhatni a komponens lényegi funkciója látvány elemként szolgál.

## Kontextus

A kontextus file lényege (másnéven a ShopContext) hogy az oldalon található összes és szükséges adatot, műveletet az alkalmazás különböző részein hasznosítsa, mint például a kosár kezelése a termékek lekérése a backendből vagy a felhasználói autentikáció.

### ShopContext.jsx

#### Állapot kezelés:

**currency & symbol:** A pénznemet és szimbólumot tárolja (HUF és Ft).

**cartItems:** A kosár tartalmát tárolja, amely egy objektumban van, ahol a termékek azonosítói, azok mennyiségei és méretei vannak.

**products:** A termékeket tárolja, amelyeket a backendről tölt le.

**token:** A felhasználói autentikációhoz használt token tárolása.

**search & showSearch:** A keresés állapota és annak megjelenítése.

**delivery\_fee:** A szállítási díjat tárolja (jelenleg 0).

#### Kosár Műveletek:

**addToCart:** Ez a funkció hozzáadja a kosárhoz a kiválasztott terméket, méretet és mennyiséget. Ha a termék már a kosárban van, frissíti a mennyiséget. Ha nem elérhető a kívánt mennyiség a raktárban, hibát ad vissza. Az adatokat a localStorage-ban is eltárolja.

**updateQuantity:** A kosárban lévő termékek mennyiségének frissítése. Ha van érvényes autentikációs token, a backend-en is frissíti a kosár állapotát.

**clearCart:** Kiüríti a kosarat amikor a rendelés befejeződik.

#### Kosár Számítások:

**getCartCount:** Megszámolja a kosárban lévő összes termék mennyiségét.

**getCartAmount:** A kosár összes értékét (árát) számítja ki, figyelembe véve az egyes termékek méretét és azok árait.

#### Backend Kommunikáció:

**getProductsData:** A termékeket lekéri a backendről, és frissíti a termékek állapotát. Ha a termékek lekérdezése sikeres, az adatokat felhasználja a front-end oldalon is.

**AddToCart:** Ha a felhasználó be van jelentkezve (a token létezik), akkor a kosárba tett termékekről információt küld a backendnek, hogy az adatbázisban is frissítse a kosarat.

## **Felhasználói bejelentkezés és kijelentkezés:**

**logout:** A felhasználót kijelentkezteti a rendszerből és törli a token-t és a kosár adatait, majd átnavigál a bejelentkezési oldalra.

## **Pages/Oldalak**

### **About.jsx**

Az /about oldalon magáról a cégről lehet információkat találni és egyéb információkat, mint hogy miért érdemes a céget választani és hasonló dolgok. Az oldalon használjuk a NewsletterBox komponenst is.

### **Cart.jsx**

Az oldalon a felhasználó által megvásárolni kívánt termékek találhatóak és a CartTotal komponenst is használjuk, amin a termékek összesített értéke látható a szállítási díjjal együtt. Az oldalon a rendelés leadása gombra kattintva a /placeOrder oldalra navigál, ahol a személyes adatok megadása után megtörténik a rendelés leadása. Itt lehet a rendelés leadása előtt a kívánt mennyiséget frissíteni vagy esetleg törölni. Fontos, hogy a rendelés leadása gomb addig inaktív amíg nincs valami a kosárban amint van valami a kosárban aktív lesz. Mivel a rendelés bejelentkezéshez van kötve csak akkor tud rendelni a felhasználó, ha ennek a kritériumnak eleget tett ellenkező esetben a bejelentkezés oldalra navigál.

### **Contact.jsx**

Az oldal hasonlóképpen, mint a /about csak adatokkal tájékoztatja a felhasználót. Itt van az üzlet nyitvatartása és címe, valamint az elérhetőségei, mint telefonszám és e-mail cím. Az oldal egyik különös funkciója az útvonal gomb amire kattintva a felhasználó jelenlegi tartózkodási helyétől elnavigál az előre a kódban beállított címre. A térkép a google maps-et használja és azt nyitja meg. Az oldalon helyet kapott még a NewsletterBox komponens is.

### **DataProtection.jsx**

Az oldalon egy előre a kódban megírt statikus (beégetett) adatvédelmi tájékoztató van. Komponenseket nem használtunk és az oldalon semmi olyan funkció nincsen, ami a tájékoztató információkon túl mutatna.

### **Delivery.jsx**

Hasonló képen, mint a DataProtection-ban vagy az About oldalon itt is csak tájékoztatjuk a felhasználókat azokról a lehetőségekről, amik a szállítást illetik. Itt egy lenyitható fül van, amiben benne van a szállítási információ, amiből tájékozódhat a felhasználó.

### **Home.jsx**

Ez az első oldal, amivel a felhasználó találkozik és ez a legtöbb komponenst használó oldal. Itt a Hero, LatestCollection, BestSeller, Policy, NewsletterBox komponenseket használjuk és itt tudják a felhasználók tájékoztatni magukat az oldal által kínált népszerű és

legújabb termékeivel kapcsolatban és mellette azokról a szolgáltatásokról, amiket a rendelések mellett tanúsít a cég.

### **Login.jsx**

Itt történik a felhasználók regisztrálása és bejelentkeztetése, amik egy `currentState` vagy állapotváltozó alapján működik. Itt olyan funkciók vannak, mint hogy a regisztrációkor a felhasználók által megadott e-mail címet ellenőrzi, hogy létezik-e és ha igen akkor nem engedi az újra regisztrációt vagy esetleg a jelszavak megadásánál az előre kijelölt kritériumoknak megfelelő jelszót adjanak a felhasználók. A bejelentkezési oldalon figyelve van, hogy a jelszó és az e-mail cím egyezzen és csak úgy engedjen belépni az oldalra vagy azt is, hogy csak olyan e-mail címet fogadjon el, ami már regisztrált az oldalra.

### **Orders.jsx**

Az oldal csak tokennel rendelkező eleve bejelentkezett felhasználóknak elérhető csak mivel itt tudják a rendeléseiket megtekinteni és az esetleg rendelési állapotokat figyelemmel kísérni. Az állapotoktól függően az átvehető státuszban lévő termékeknél az a hasonló funkció elérhető, mint a `/contact` oldalon, hogy a státusz állapotában megjelenik egy gomb, hogy útvonal és a felhasználó tartózkodási helytől elnavigál a boltba. Ezzel egyetemben e-mailt is kap a felhasználó arról, hogy a rendelés létrejött és arról is, hogy ha átvehető vagy esetleg lezárult, vagy nem teljesített.

### **PlaceOrder.jsx**

Itt történik a rendeléseknek az utolsó fázisa, amikor is a felhasználó megadhatja a személyes adatait a rendelés leadásához. Ez is csak a bejelentkezett felhasználóknak elérhető. Az adatok megadásánál a teljes nevet és az e-mail címet nem lehet megadni mert azt a regisztrációkor megadott adatokat felhasználva automatikusan helyettesíti és használja fel. Az ország a megye és a telefonszám mezők összekötve frissülnek, amikor is az ország kiválasztásakor a telefonszám mezőben automatikusan frissül az előhívó szám a megyéknél meg csak az adott országhoz tartozó megyéket lehet kiválasztani. A többi adatot, mint az utca és az irányító szám minden egyes rendelésnél kézzel kell megírni. A rendelést két opcióval lehet fizetni. Alapértelmezetten az utánvételes fizetés van megjelölve, amikor is a boltban személyes átvétellel történik a termék átvétele és a termék ellenértékének a megtérítése is személyesen történik meg. A másik opció a bankkártyás fizetés, amihez a Stripe külső szolgáltatását használjuk, amikor is átirányít a szolgáltató oldalára a bankkártyás fizetés lebonyolításához. A kódban azt a `verify.jsx` kezeli. A rendelés gomb addig inaktív amíg a mezők nincsenek kitöltve miután ennek eleget tett a felhasználó a rendelés leadható és a gomb aktív állapotba kerül.

### **Product.jsx**

Itt az egyes termékek adatait lehet megtekinteni és tájékozódni róluk. Itt történik a termékek kosárba adása és az opciók, mennyiségek kiválasztása. Az oldalon a RelatedItems komponenst is használjuk, hogy az adott termékekhez hasonló termékeket is meglehessen nézni esetleg ezzel is ajánlásokat tegyünk a felhasználóknak. Az oldalon egyik hiba a vélemény mező használhatatlansága, ami teljes mértékben csak egy beégetett értéket mutat és semmi hitelessége nincsen. A hiányosságot mi inkább fejlesztési javaslatnak fogalmazzunk meg a projektünkben. Az oldalon külön a termékek leírása és az opciók leírása is jelen van ezzel segítve a felhasználók tájékoztatását a termék megvételével kapcsolatban.

### **Products.jsx**

Itt az összes termék megtalálható, amit az oldalon feltöltött az admin. A termékeket lehet szűrés alapján, mint kategória és típusok keresni és válogatni belőlük. A termékek keresését az is megkönnyíti, hogy a searchBar komponens is csak ezen az oldalon érhető el, ami a termékek neve alapján tudja a keresett termékeket megjeleníteni. A szűrés és keresés mellett egy rendezési funkció is jelen van, ahol a termékeket lehet újdonság és ár szerint növekvő vagy csökkenő sorrendbe rendezni. Az oldalon az egységes termék ablakok megjelenésért a ProductItem felel.

### **ResetPassword.jsx**

Ez csak egy koncepció. Ez egy olyan oldal lenne, ahol az új jelszó megadását lehetne elérni, de mivel ezt nem fejeztük be nagyvonalakban befolyásolta volna a működését az oldalnak ezért teljes ki kommentelésre került.

### **Verify.jsx**

Az oldal a bankkártyás fizetés miatt lett létrehozva, ahol is a rendelés leadásánál a kártyás opció választva a Stripe oldalára navigál és az ott az adatok megadása után és a fizetés véglegesítése után ellenőrzi, hogy ez a rendelés sikeresen létrejött-e vagy nem.



## Backend

### Controller-ek:

A controller-ek az "irányítók", más néven a vezérlők, akik eldöntik, hogy egy adott kérésre hogyan reagáljon az alkalmazás. Mi 4 db controllert használtunk a backendben.

#### **cartController.js:**

Ez felelős a kosárral kapcsolatos műveletek elvégzéséért. Itt történik a termékek hozzáadása a kosárba, a felhasználó adatainak lekérése az adatbázisból, amikor szükséges, az üres kosár létrehozása, a kosár frissítése, amikor szükséges, a kosár adatainak elmentése az adott felhasználóhoz és a kosár adatainak lekérése is.

#### **orderController.js:**

Itt történnek a rendeléssel kapcsolatos műveletek, úgy, mint az utánvételes és a kártyás rendelés implementálása. A felhasználó rendelései, a rendelési státuszok frissítése, illetve az összes rendelés megjelenítése az admin felületen is.

#### **productController.js:**

Ebben a controllerben zajlik az összes termékkel kapcsolatos művelet. Ilyen a termék hozzáadása a webshophoz, a képek kezelése, a képek feltöltése a Cloudinary-ba, a termékadatok összegyűjtése, valamint a termékek listázása, törlése és egyetlen terméklekérése is. Ide tartozik még a termék frissítése, a termék ID alapú keresése, az új képek kezelése és azok feltöltése, és a bestseller termékek frissítése.

#### **userController.js:**

Az utolsó controller a userController, ami a felhasználóval kapcsolatos műveleteket végzi el. Például a felhasználó belépéséért ez felelős, továbbá a felhasználó regisztrálásáért is. Itt történik a user email címének validálása és hogy a jelszó elég erős legyen hozzá. Itt ellenőrizzük azt is, hogy a felhasználónak van-e már fiókja, vagy nincs. Itt történik az admin regisztrációja is.

### Middleware-k:

A backendben a middleware-ek egyfajta köztes szoftverként működnek, amelyek a kérések és válaszok áramlását irányítják, valamint különböző funkciókat valósítanak meg a folyamat közben. Ilyen például az autentikáció és a jogosultságkezelés és a hibakezelés, meg a logolás is. Mi 3 db middleware-t használtunk a backendben.

#### **adminAuth.js:**

Ez azért felelős, hogy ellenőrizze, a beérkező kérés jogosult-e egy adminisztrátori művelet végrehajtására. Itt vannak tokennel kapcsolatos műveletek is. Ez a token kinyerése a fejlécből és a token dekódolása, ellenőrzése. Van hibakezelés is itt.

Ez a middleware tehát egy biztonsági réteggént működik, amely biztosítja, hogy csak azok a felhasználók férjenek hozzá az adminisztrációs funkciókhoz, akik érvényes

adminisztrátori tokent küldenek be. Ez segít megvédeni az érzékeny funkciókat az illetéktelen hozzáféréstől.

#### **auth.js:**

Ez a middleware a felhasználói autentikáció alapját képezi. Itt történik a felhasználó azonosítása. Ez a rendeléshez szükséges (elengedhetetlen) bejelentkezést biztosítja.

#### **multer.js:**

Ez a middleware a fájlok feltöltését kezeli a multer nevű könyvtár segítségével. A multer egy népszerű Node.js middleware, amely a fájlok feltöltését egyszerűsíti.

#### **Modell-ek:**

A backendben a models (modellek) az alkalmazás adatstruktúrájának meghatározásáért és kezeléséért felelnek. Ez különösen fontos, ha egy adatbázist használunk. A modellek segítségével hajtunk végre CRUD műveleteket. A models tehát kulcsszerepet játszik a backendben az adatstruktúrák és adatbázisműveletek kezelésében.

#### **orderModel.js:**

Ez a modell meghatározza a rendeléshez szükséges adatstruktúrákat, elmenti és kezeli az adatokat az adatbázisban és bizonyos mezőknél alapértelmezett értékeket is megad.

#### **productModel.js:**

Ez a modell a termékek adatstruktúráját határozza meg.

#### **userModel.js:**

Ez a modell pedig a felhasználóhoz tartozó adatok struktúrájáért felelős (név, email, jelszó, kosár adatok).

#### **Route-ok:**

A routes (útvonalak) az alkalmazás végpontjainak logikai szervezéséért felelnek. Ezek határozzák meg, hogy az adott URL-eken keresztül érkező kéréseket milyen logika kezelje, például milyen adatokat kapjon a felhasználó, vagy milyen műveletek hajródjának végre.

Ide tartozik tehát az útvonalak definiálása. Ide integrálhatunk middleware-eket is az autentikáció miatt például. Mi a backendben 4 db route-ot használunk.

#### **cartRoute.js:**

Itt történik a kosár tartalmának lekérése, a kosárhoz adás és a kosár módosítása is. A jogosultságot pedig az authUser middleware ellenőrzí.

#### **orderRoute.js:**

Itt vannak a rendeléssel kapcsolatos útvonalak. Itt találjuk az admin funkciókat, mint a rendelések listázása és státuszának módosítása. Itt vannak a fizetési funkciók is, ami az utánvétes és a Stripe-on keresztüli fizetés. Valamint itt találjuk a felhasználó lehetőségeit is, ami a saját rendelésének megjelenítéséből áll. Továbbá van még itt egy fizetési autentikáció is.

**productRoute.js:**

Itt találhatóak a termékekkel kapcsolatos útvonalak. Van itt hozzáadás, törlés és módosítás, amit csak az admin tud megtenni. Van még listázás és a termék keresése ID alapján.

**userRoute.js:**

Itt vannak a felhasználóval kapcsolatos útvonalak. Ide tartozik a regisztráció, a bejelentkezés és az adminként való bejelentkezés is.

## Admin

Az admin felület az az oldal, amit csak az admin lát. Az Admin oldali működés is nagyon hasonló képen van felépítve mint a Frontend mivel itt is találhatóak komponensek és oldalak. Lényegében a komponensek olyan elemek, amiket könnyen lehet importálni az oldalakra. Ezt a módszer azért láttuk jobbnak mert így nem kellett minden egyes oldalon külön saját kinézetet értem itt azt, hogy minden oldalon a Footert megírni vagy a Headert hanem egyszerűen elég volt csak megírni egyszer és csak egy hivatkozás alapján az oldalakra importálni. A módszer ihletét számomra a Wordpress-ben is általában használatos Elementor plugin adta, aminek működése nyilván sokkal egyszerűbb mert a felhasználó úgy építhet oldalt, hogy egyes komponensekből építheti össze az oldalt hasonló képen, mint ha egy Lego várat akarna építeni. Én ezt a módszert akartam volna valahogy átültetni a projektbe és így jött az az ötlet, hogy sokkal jobb lenne, ha lennének olyan komponensek, amiket szabadon belehet építeni az egyes oldalakra így megkönnyítve a munkámat és rengeteg időt és energiát megtakarítva sokkal átláthatóbb fejlesztői környezetet alkotva. Természetes törekedtünk, hogy minden egyes komponens reszponzívan a használt készülékhez képest megfelelően jelenjen meg.

## Komponensek

### Login.jsx

A komponens lényege, hogy az admin bejelentkezzen és hogy ne tudjon senki enélkül az egész oldalhoz hozzáférni. A komponens mindig megjelenik amikor nincs bejelentkezve az admin ezzel is megakadályozva azt, hogy bárki is hozzá férjen az oldalhoz. Az oldalon egy e-mail mező és egy jelszó mező kitöltésére szolgáló rész van és mind a két helyen ellenőrizni a formátumot és csak az admin által használható jelszót és e-mail címet engedi elfogadni.

### Navbar.jsx

A komponensben összesen egy gomb és egy oldal által használt logó található. A kilépés gomb megnyomásakor az admin kilépteti magát az oldalról. A komponens az összes oldalon alapelem így lehetővé téve, hogy az összes oldalról képes legyen magát kijelentkeztetni.

### Sidebar.jsx

A SideBar az összes oldalon alapelem amikor a bejelentkezés megtörtént. Itt vannak mondhatni az irányításra szolgáló információk, mint hogy a termékek feltöltéséhez, a leadott rendelések megtekintéséhez és a felhasználók kezelése mellett az összes terméket is ezen a fülön tudja irányítani az admin.

### ScrollToTop.jsx

Ez a komponens azért felel, hogy az éppen használt oldalak mindig az oldal elejétől induljon és oda töltsön be ne pedig az adott oldalról való átlépéskor hagyott magasságtól nyissa

meg a következő oldalt. A komponens egy hibát hivatott kijavítani és erre a célra lett létrehozva hasonló képen, mint a frontend oldalon.

## **Pages/Oldalak**

### **Add.jsx**

Az oldal a termékek hozzáadásáért felel és célja az, hogy az admin a termékeit feltudja tölteni. Az oldalon a 4 kép feltöltésére szolgáló hely van és emellett a termék nevére, leírására, a mérethez tartozó árak és a készletek megadása mellett felkapottság jelölésére szolgáló helyek vannak. A kategóriák és típusok kiválasztása a kódban van beégetve és új kategóriákat nem tud az admin hozzáadni. Fejlesztési javaslatként ezt is megfogalmaztuk, hogy miért nem lenne ez a funkció jelenleg olyan szinten helytálló, hogy az admin adjon hozzá új kategóriákat. A hozzáadás gomb csak addig inaktív amíg a négy kép nincs feltöltve, de amúgy nem is engedi a termékek feltöltését anélkül, hogy az összes mező valamilyen értékkel ne lenne kitöltve. Csak innen lehet termékeket feltölteni és csak is az admin képes erre.

### **List.jsx**

Az oldal a termékek kilistázásáért felel és itt lehet megtekinteni minden egyes adatot a termékekről. Itt nyílik lehetőség arra is, hogy a termékeket törölni tudjuk vagy esetleg az egyes termékek szerkesztési felületére átléphessünk. A termékeket név alapján lehet megkeresni és erre az oldal tetején lévő searchbar ad lehetőséget. A termékek adataiba tartozik, hogy itt lehet a termékekről a feltöltött első képet, a termék nevét, kategóriáját, méretét, amit változtatás hatására a mellette lévő árat is megváltoztatja, a felkapottságát megfigyelni, hogy az vagy nem és a legvégén a műveletek végrehajtására használatos ikonok megnyomására végrehajtani a törlést vagy a módosítást.

### **ManageUsers.jsx**

Az oldal lényege, hogy itt tudja az admin látni azokat a felhasználókat, akik regisztráltak az oldalra és itt tudja állítani a státuszukat, hogy aktívak vagy esetleg inaktívak. Itt is hasonló képen, mint a termékek oldalán tudja a felhasználók neve vagy e-mail címe alapján megkeresni a felhasználókat. Az inaktíválás gomb megnyomására egy pop up ablak jelenik meg aminél megkérdezi, hogy tényleg inaktíválja a felhasználót vagy mégse és nyilván, ha inaktíválja akkor megváltozik a gomb kinézete és a felhasználó nem tud az oldalra belépni ennek hatására viszont, ha inaktív állapotból helyezi vissza aktív állapotba akkor hasonló képen visszaállítja a felhasználót.

## Orders.jsx

Az oldal a rendeléseket mutatja meg. Itt találhatóak a rendelés adatai, mint a megvásárolt termék a cím, ahonnan rendelték és a felhasználó neve e-mail címe fizetés típusa, dátuma, értéke. A rendelés állapotát az admin tudja állítani és ez alapján a felhasználó orders oldalán is változik a rendelés állapota. A rendelés mellett az állapotát is az admin tudja irányítani, de mindegyik rendelési formához alapjáraton tartozik egy státusz. A bankkártyás fizetésnél automatikusan teljesített a fizetés állapota utánvételes fizetés esetén van folyamatban. A rendelés állapotától függően változik a rendelések információit tartalmazó cella keretének színe, ami sokkal jobban átláthatóvá teszi a rendelések kezelését. Három külön rész van a rendelések átláthatósága miatt az egyikben az aktív hirdetések, a másikkban a már teljesített rendelések vannak míg a harmadikban a nem teljesített vagy a nem átvett rendelések vannak. A rendelések átláthatósága miatt még egy teljes mértékű keresési lehetőséget is bevezettünk, aminél a felhasználó e-mail címe vagy simán a teljes neve alapján lehet a rendeléseket megjeleníteni. Olyan funkciók is vannak, mint hogy csak egy bizonyos dátumtól mutassa meg a rendeléseket és azon belül lehessen keresni, vagy csak a fizetés vagy a rendelés állapotától függően mutassa meg a termékeket. Rendezni is lehet, hogy a legújabb rendelések kerüljenek előre vagy azok, amik a legrégebben érkeztek be az oldalra. ó

## PopUp.jsx

Az PopUp egy olyan oldal, ami csak a ManageUsers oldalon használatos mert itt csak egy olyan funkciót lát el, hogy egy bizonyos gomb megnyomásának hatására megjelenik ez az oldal, aminek *igen* gomb lenyomására megváltozik a kívánt esemény és ellenkező esetben a *mégse* megnyomására, mint ha meg se nyomta volna a gombot. Az oldal tulajdonképpen lehetne komponens is csak mivel csak a ManageUsers oldalon használjuk így a komponens, importálás megkönnyítése miatt hagytuk végül az oldalak között.

## Update.jsx

Az update lényege, hogy ezen az oldalon lehet az elve feltöltött termékeket módosítani. Itt lehet új árat állítani ez egyes méretekhez, esetleg új nevet leírást vagy mennyiséget (készletet), kategóriát vagy típust is szerkeszteni. A képeket egyesével lehet módosítani és a legelső képen lévő jelenik meg fő termék képnek az oldalon.

## Tesztelés

Felhasználói teszt				
Dátum:	2025.04.10			
Tesztelő:	Sövér Benjámin			
Tesztkörnyezet:	Windows 10			

Komponensek				
Footer				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A footerben minden egyes gomb, ami kattintható az megfelelő oldalra irányít	OK		
2	A Footerben a kapcsolat felvételnél feltüntetett e-mail cím és telefonszám helyes külső funkciót akar megnyitni	OK		
3	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
4	A Footer és Navbarban a logó kattintható és a főmenübe irányít vissza	OK		
5	Betűtípus	OK		

Hero				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A HERO komponens a termékekre irányít a vásárlás kattintáskor	OK		
2	A megfelelő képet mutatja	OK		

3	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
4	Betűtípus	OK		

BestSeller				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	Azokat a termékeket mutatja meg amiket az admin népszerűnek jelölt a termék hozzáadásnál	OK		
2	Ha nincs ilyen termék akkor kiírja, hogy nincs népszerű termék	OK		
3	Megfelelő címet jelenít meg (NÉPSZERŰEK)	OK		
4	A termékek a saját oldalukra irányítanak	OK		
5	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
6	Képek egységes mérete	OK		
7	Helyes árak megjelenítése	OK		
8	A termékek a saját oldalukra irányítanak	OK		
9	Kattintásra és a kurzor rávitelénél előrébb ugrik a termékek képe	OK		
10	A megfelelő képet mutatja	OK		
11	A megfelelő termék nevét mutatja	OK		
12	Betűtípus	OK		



CartTotal				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A megfelelő címet mutatja (KOSÁR ÉRTÉKE)	OK		
2	Rendelés teljes értékét mutatja	OK		
3	A szállítás értékét mutatja	OK		
4	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
5	A teljes értéket mutatja mind a szállítást és a termékek mennyiségét figyelembe véve	OK		
6	Betűtípus	OK		

LatestCollection				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A megfelelő címet mutatja (Legújabb termékeink)	OK		
2	Azokat a termékeket mutatja meg amiket az admin legfrisseben feltöltött	OK		
3	Csak az első 10 legújabb terméket mutatja	OK		
4	A legújabban feltöltött termékek kerülnek a sorrendben előre	OK		
4	A termékek a saját oldalukra irányítanak	OK		
5	Méretarányos betöltése (Reszponzívan minden méretben)	OK		

6	Képek egységes mérete	OK		
7	Helyes árak megjelenítése	OK		
8	A termékek a saját oldalukra irányítanak	OK		
9	Kattintásra és a kurzor rávitelénél előrébb ugrik a termékek képe	OK		
10	A megfelelő képet mutatja	OK		
11	A megfelelő termék nevét mutatja	OK		
12	Betűtípus	OK		

NavBar				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A megfelelő oldalt mutatja meg amikor az adott gombra kattint (Menü, Rólunk stb.)	OK		
2	Egyező mérete van a szavaknak	OK		
3	A Headerben a felhasználó ikon és a kosár ikon helyes oldalra irányít	OK	Ha bevan jelentkezve akkor is	
4	Az felhasználói ikon megnyomásakor vagy amikor a kurzort rávisszük megjelenik a rendeléseim és kilépés funkció	OK	És ezek a funkciók megnyomások helyesen működnek és irányítanak át az adott oldalra	
5	A keresési ikon megnyomására nem nyílik meg a keresés mező azon az oldalon, ahol le lett tiltva	OK	Csak a termékeknel engedélyezük a megnyitását (/products)	
6	Méretarányos betöltése (Reszponzívan minden méretben)	OK		

7	A Footer és Navbarban a logó kattintható és a főmenübe irányít vissza	OK		
8	Betűtípus	OK		

NewsLetterBox				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A megfelelő formátumot fogadja csak be	OK		
2	A megfelelő címet mutatja	OK		
3	Kap e-mailt a feliratkozásról	NEM	Mivel ez a funkció nem lett megcsinálva így nem kap e-mailt	H1
4	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
5	Üres mezőt hibának érzékeli	OK		
6	Betűtípus	OK		

Policy				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A megfelelő címeket mutatja meg	OK		
2	A megfelelő képeket mutatja	OK		
3	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
4	Betűtípus	OK		

ProductItem				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy

1	Megfelelő méretben jelennek meg a termékek	OK	Ezek a kis termék boxok a menüben és a termékek oldalon	
2	A megfelelő termék nevét mutatja	OK		
3	A megfelelő méretben jelennek meg a képek	OK		
3	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
4	Betűtípus	OK		

RelatedProducts				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
2	A megfelelő méretben jelennek meg a képek	OK		
3	A megfelelő termék nevét mutatja	OK		
4	A megfelelő termék árát mutatja	OK		
5	A megfelelő helyeken jelenik meg	OK	A termék oldalán jelenik meg csak	
6	Betűtípus	OK		

ScrollToTop				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	Az összes oldal mindig a tetjéről indul és onnantól görgethető	OK	Ez a komponens azért lett megírva	

SearchBar				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	Csak akkor elérhető amikor a termékek oldalán (/products) van.	OK		
2	Amikor más oldalon akar rákattintani akkor nem működik	OK		
3	A bezárás gomb megnyomására a searchbar bezáródik	OK		
4	A termékeket figyeli csak és azon belül is a címüket vagy a termék nevét	OK		
5	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
6	Betűtípus	OK		

Title				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A cím minden egyes helyen azonos	OK		
2	A betűtípus azonos minden egyes címnél	OK		
3	A cím alatti csík mindenhol ugyanolyan széles	OK		

Oldalak				
About				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A komponensek helyesen jelennek meg	OK	Használt komponensek: Title, NewsLetterBox	
2	A beállított képek és szövegek helyesen jelennek meg	OK		
3	Méretarányos betöltése (Reszponzívan minden méretben)	OK		

Cart				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A komponensek helyesen jelennek meg	OK	Használt komponensek: Title, CartTotal	
2	A beállított szövegek és címek helyesen jelennek meg	OK		
3	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
4	Rendelés gomb inaktív amíg nincs a kosárban semmi	OK		
5	Rendelés gomb aktív, ha van a kosárban valami	OK		
6	Nem lehet negatív mennyiséget megadni	OK		
7	Nem lehet több mennyiséget megadni, mint amennyi készleten van	OK		
8	Termék törlése működik	OK		
9	Termék törlésénél a kosárban lévő szám is változik	OK		

10	Mennyiség módosítására az összértékek változnak	OK		
11	A rendelési gomb csak akkor továbbít a placeorder oldalra amikor be van lépve a felhasználó az oldalra	OK		
12	Ha nincs belépve a felhasználó az oldalra és rendelni akar akkor átirányít a belépés oldalra	OK		
13	Minimum egy darab mennyiséget rendel az a minimum ellenkező esetben nem működik a rendelés	OK		

Contact				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A komponensek helyesen jelennek meg	OK	Használt komponensek: Title, NewsLetterBox	
2	A beállított képek és szövegek helyesen jelennek meg	OK		
3	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
4	Az útvonal gomb megnyomására átirányít a google maps-re és a jelenlegi helyzettől navigál a kódban megadott célhoz	OK		

DataProtection				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A komponensek helyesen jelennek meg	OK	Használt komponensek: Title	
2	A beállított képek és szövegek helyesen jelennek meg	OK		
3	Méretarányos betöltése (Reszponzívan minden méretben)	OK		

Delivery				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A komponensek helyesen jelennek meg	OK	Használt komponensek: Title, NewsLetterBox	
2	A beállított képek és szövegek helyesen jelennek meg	OK		
3	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
4	A személyes átvétel lehetőség rendesen lenyílik és csukódik	OK		
5	A lenyitható fülben minden információ helyesen és szimmetrikusan fel van tüntetve	OK		



DataProtection				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A komponensek helyesen jelennek meg	OK	<b>Használt komponensek:</b> Hero, LatestCollection, BestSeller, Policy, NewsLetterBox	
2	A beállított képek és szövegek helyesen jelennek meg	OK		
3	Méretarányos betöltése (Reszponzívan minden méretben)	OK		

Login				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A komponensek helyesen jelennek meg	OK	<b>Használt komponensek:</b> NINCS	
2	A beállított képek és szövegek helyesen jelennek meg	OK		
3	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
4	A mezők kitöltése figyelembe van véve mind a bejelentkezés és a regisztráció oldalán	OK		
5	Az email mezők csak email formátumot fogadnak el	OK		
6	A jelszó mező csak a kritériumoknak megfelelő jelszavakat engedi regisztrálni	OK		
7	A jelszóban a szem megnyomására ki be kapcsolható a jelszó láthatósága	OK		
8	A regisztráció során a jelszó mezőbe írásakor	OK		

	automatikusan megjelennek a jelszó követelmények			
9	A mezők alatt lehet váltogatni a regisztráció és a bejelentkezés között	OK		
10	A gombok megnyomásakor, mint a Bejelentkezés vagy Regisztráció a megfelelő hiba és sikeres toast messagest küldik vissza	OK		
11	Már eleve regisztrált email címmel nem lehet újra regisztrálni	OK		
12	Regisztrációkor e-mail érkezik	OK		

Orders				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A komponensek helyesen jelennek meg	OK	Használt komponensek: Title	
2	A beállított képek és szövegek helyesen jelennek meg	OK		
3	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
4	A rendelés oldal csak akkor elérhető, ha a felhasználó be van lépve	OK		
5	Minden termék, amit rendelt meg lehet találni	OK		
6	A két kategória, mint a folyamatban lévő és teljesített rendelések lenyithatóak és teljes mértékben áttekinthetőek	OK		
7	A rendelt termékek adatai teljesen látszódnak	OK		
8	A rendelésről kapok e-mailt	OK		
9	A rendelés atvehető állapotáról kapok e-mailt	OK		
10	A státusz frissítése jól működik	OK		

PlaceOrder				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A komponensek helyesen jelennek meg	OK	Használt komponensek: Title, CartTotal	
2	A beállított képek és szövegek helyesen jelennek meg	OK		
3	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
4	A oldal csak akkor elérhető ha a felhasználó be van lépve	OK		
5	A név és az e-mail cím mező automatikusan ki van töltve a felhasználó adataival	OK		
6	A cégnév opcionális nem figyel, hogy ki van töltve	OK		
7	Az országok kiválasztása után az ahhoz tartozó megyék jelennek meg	OK		
8	Az országok kiválasztása után automatikusan bekerül az előhívó szám a telefonszám mezőbe	OK		
9	A rendelés gomb addig inaktív amíg a kötelező mezőkben nincs valami beírva	OK		
10	Az utánvételes vagy bankkártyás fizetést jól tudja feldolgozni és ha kártyával fizet akkor a megfelelő tranzakciós oldalra irányít	OK		
11	Az üzlet nyitvatartása információkban minden helyesen megjelent és a telefonszám és az e-mail cím kattintható és azokat a külső hivatkozásokat hívják meg amiket kell	OK		

Product				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A komponensek helyesen jelennek meg	OK	Használt komponensek: RelatedProducts	
2	A beállított képek és szövegek helyesen jelennek meg	OK		
3	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
4	A méret opciók választás hatására automatikusan változik a termék ára	OK		
5	A megfelelő termék nevét mutatja	OK		
6	A megfelelő képet mutatja	OK		
7	A kosárba gomb akkor inaktív amikor a készlet nulla	OK		
8	A készlet a méretek szerint változik	OK		
9	Nem lehet több mennyiséget hozzáadni a kosárhoz, mint amennyi a készlet	OK		
10	Ha nulla a mennyiség, amit a kosárba akarok tenni akkor inaktív lesz a gomb automatikusan	OK		
11	Az opciókról bővebben rész minden terméknel azonos	OK		
12	A képeknél a megfelelő kerül a nagy kép helyére amikor a kicsi képekre kattintok	OK		
13	A megfelelő képeket mutatja és megfelelő sorrendben az adott termékhez	OK		

Products				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A komponensek helyesen jelennek meg	OK	Használt komponensek: ProductItem, Title	
2	A beállított képek és szövegek helyesen jelennek meg	OK		
3	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
4	A szűrők rendesen lenyílnak és csukódnak	OK		
5	A termékek rendezése is megfelelően működik	OK		
6	A megfelelő képet mutatja a termékekhez igazítva	OK		
7	Amikor rámegyek a képre vagy az cursort ráteszem akkor a képek menőnek	OK		
8	A szűrők rendesen szűrik a termékeket típus és kategória alapján	OK		
9	A searchbar rendesen nyílik és záródik	OK		

Verify				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A komponensek helyesen jelennek meg	OK	Használt komponensek: NINCS	
2	A beállított képek és szövegek helyesen jelennek meg	OK		
3	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
4	A stripe oldala megnyílik	OK		

5	A bankkártya adatok megadása után visszatér a rendelések oldalra	OK		
6	Ha a bankkártya adatok megadása közben meggondolom magam akkor visszatérek az oldalra és a kosárhoz dob vissza	OK		

<b>Admin teszt</b>				
<b>Dátum:</b>	2025.04.10			
<b>Tesztelő:</b>	Sövér Benjámin			
<b>Tesztkörnyezet:</b>	Windows 10			

<b>Komponensek</b>				
<b>Login</b>				
<b>Sorszám</b>	<b>Leírás</b>	<b>Eredmény</b>	<b>Megjegyzés</b>	<b>Hibajegy</b>
1	Figylei a beviteli mezőket	OK		
2	Méretarányos betöltése (Reszponzíván minden méretben)	OK		
3	A megfelelő hiba üzeneteket adja vissza	OK		
4	A belépés gomb megnyomására a termék hozzáadás oldalra navigál	OK		
5	Csak a belépett admin tud használni mindent csak úgy tud megnyitni valamit, ha be van lépve	OK		
6	Csak az admin email és jelszó kombinációt fogadja el	OK		

Navbar				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	A kilépés gomb kattintásra kiléptet az oldalról és újra belépést kér	OK		
2	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
3	Csak a belépett admin tud használni mindent csak úgy tud megnyitni valamit, ha be van lépve	OK		

ScrollToTop				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	Az összes oldal mindig a tetejéről indul és onnantól görgethető	OK	Ez a komponens azért lett megírva	

SideBar				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
2	Csak a belépett admin tud használni mindent csak úgy tud megnyitni valamit, ha be van lépve	OK		
3	A megfelelő oldalra navigál	OK		
4	A megjelenése egységes és a rövidítés is megfelelő szélességnél lesz rövidítve	OK		



Oldalak				
Add				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
2	Csak a belépett admin tud használni mindent csak úgy tud megnyitni valamit, ha be van lépve	OK		
3	A képek hozzáadása egyesével történik	OK		
4	A kategória és típus a három eredetileg beégetett elemet tartalmazza	OK		
5	A kategória és típus megfelelően lenyílik és ki lehet választani a megfelelő elemet	OK		
6	A méret/ár mezőbe nem lehet negatív karaktert belevinni	OK		
7	A készlet mezőbe nem lehet negatív karaktert belevinni	OK		
8	A méret/ár mezőbe nem lehet speciális karaktert bevinni	OK		
9	A készlet mezőbe nem lehet speciális karaktert belevinni	OK		
10	Addig inaktív a hozzáadás gomb amíg nem töltött fel 4 képet az admin	OK		
11	Addig nem lehet feltölteni terméket amíg nincs neve	OK		
12	Addig nem lehet feltölteni terméket amíg nincs leírás róla	OK		

13	Addig nem lehet feltölteni terméket amíg nincs ára	OK		
14	Úgyis fel lehet tölteni terméket amíg a készlet nulla	OK		
15	Felkapottakhoz adás gomb hatására a felkapottak arra a helyre kerülnek az oldalon	OK		

List				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
2	Csak a belépett admin tud használni mindent csak úgy tud megnyitni valamit, ha be van lépve	OK		
3	Minden termék, amit az admin feltöltött itt van	OK		
4	Minden termék első képe, neve, Kategória, Méret, Ár, Felkapott, Művelet látszódik és jó értéket mutat	OK		
5	A méretnél lehet változtatni, hogy kicsi közepes vagy nagy	OK		
6	A méret/ár mezőbe nem lehet negatív karaktert belevinni	OK		

7	A kiválasztott méret alapján változik az ár is	OK		
8	Kijelzi, hogy felkapott vagy nem	OK		
9	Műveleteknél a törlés funkcióra kattintva kitörli a terméket	OK		
10	A ceruza ikonra kattintva módosítani lehet a terméket és átdob a termék módosítás oldalra	OK		
11	A keresés a termékek nevére működik	OK		

Orders				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
2	Csak a belépett admin tud használni mindent csak úgy tud megnyitni valamit, ha be van lépve	OK		
3	Minden termék, amit a felhasználók rendeltek itt van	OK		
4	A keresés névre vagy e-mail címre működik	OK		
5	A rendezés működik a legújabb előre a legrégebbi hátulra	OK		
6	A rendelés állapota alapján jeleníti meg a rendeléseket	OK		
7	A fizetés állapota alapján mutatja meg a termékeket	OK		
8	A dátumtól rész onnantól mutatja a termékeket amelyiktől be van állítva	OK		
9	A rendelés állapota szerint változik a rendelés kerete	OK		

10	A termékek rendelési állapota alapján a felhasználónál is változik a rendelés státusza	OK		
11	Az aktív és teljesített rendelések megfelelően vannak elosztva és teljesen átlátható és összezsukható	OK		
12	A termékek fizetési állapota alapján a felhasználónál is változik a fizetés státusza	OK		

ManageUsers				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
2	Csak a belépett admin tud használni mindent csak úgy tud megnyitni valamit, ha be van lépve	OK		
3	Az összes felhasználó megtalálható	OK		
4	A keresés névre vagy e-mail címre működik	OK		
5	Inaktiválás/visszaállítás gombra kattintva megjelenik a popup	OK		
6	Amikor aktív a felhasználó akkor az inaktiválást jelzi ki, amikor meg inaktív akkor a visszaállítást	OK		
7	A popup message igen gomb lenyomásakor változik az állapot	OK		
8	A popup message mégse gomb lenyomásakor visszadob az oldalra	OK		

PopUp				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
2	Csak a belépett admin tud használni mindent csak úgy tud megnyitni valamit, ha be van lépve	OK		
3	Megjelenik a popup message a várt kinézetben	OK		
4	Megjelenik a popup message a várt állapothoz illően	OK	Amikor inaktív a felhasználó akkor helyesen kérdezi, hogy visszaállítani akarja amikor meg aktív a felhasználó akkor meg helyesen kérdezi, hogy inaktiválni akarod	
5	A popup message igen gomb lenyomásakor változik az állapot	OK		
6	A popup message mégse gomb lenyomásakor visszadob az oldalra	OK		

Update				
Sorszám	Leírás	Eredmény	Megjegyzés	Hibajegy
1	Méretarányos betöltése (Reszponzívan minden méretben)	OK		
2	Csak a belépett admin tud használni mindent csak úgy tud megnyitni valamit, ha be van lépve	OK		
3	A módosítani kívánt termékek eredeti adatai megjelennek	OK		

4	Név módosítás működik mentés után	OK		
5	Leírás módosítás működik mentés után	OK		
6	Árak különböző méretekben módosítása működik mentés után	OK		
7	Készlet különböző méretekben módosítása működik mentés után	OK		
8	Képek módosítása működik mentés után	OK		
9	A képek módosulnak egyesével is	OK		
10	A kategóriák módosulnak	OK		
11	A típusok módosulnak	OK		
12	A felkapottság módosítása	NEM	Nincs ez a funkció beleépítve sajnos a projektünkbe ezt elfelejtettük és kicsit felesleges is tartjuk	H2

## Ismert hibák és hiányosságok

A jelenlegi projektünkben a sok tesztelés miatt mondhatni minimális az ismert hibák száma, legfőbb képen az olyanoké, amik nagy arányban befolyásolhatják a weboldal zökkenő mentes működését. Ezek a hibák inkább fejlesztésre szorulnak, mint hiba kezelésre. Azonban akadnak olyan hibák is, amelyekre nem tudtunk egyszerűen megoldást találni ezért ezeket megpróbáltuk úgy megtervezni, hogy az esetleges alap funkcióját ellássa és más már működő funkciót ne befolyásoljon. Akadnak azonban olyan hibák is az oldalon, amik inkább kompromisszumot igényelnek, mint Admin és Ügyfél (Végfelhasználó) szemszögből. Ami a hibakezelést illeti, a legtöbb a tesztelési folyamatok alatt ki is derült. Mivel nagy vonalakban csak manuális tesztelést végeztünk így akadhatnak esetleges hibák, amikről sajnos nincs tudomásunk, mint a terhelést tekintve. Összesítve viszont a legtöbb hiba, amit az oldalon találtunk legfőbb kép abból ered, hogy ha a célunk az lett volna, hogy minden egyes hibát elhárítsunk akkor nem lehetett volna olyan funkciókat megvalósítani az oldalon, amik elengedhetetlenek ahhoz, hogy a célnak kitűzött alap funkciók tökéletesen működjenek. A célunk az volt, hogy alapjaiban egyszerű és abban hibamentes webshopot alkossunk.

### Felhasználó oldali hibák

#### Termékek értékelése és véleménye:

A termékek értékelése minden egyes feltöltött termékénél ugyanaz és egy teljes mértékben beégetett értéket mutat. Az értékelés a vélemények alapján tevődne össze és működne megfelelően (1-5 csillagos értékelést lehetne adni az adott termékekre) de mivel a felhasználók funkciója jelenleg nem olyan fejlett, hogy a fiókkal rendelkezők ezt megtehessek.

#### Felhasználói profil hiánya:

Az oldalon a felhasználónak nincs saját profilja, ahol megjelenhetnek az adatai, mint pl. E-mail cím, teljes név, amit regisztrációkor megad vagy esetleg profilképet sem tud beállítani magának a felhasználó. Ez a mai weboldalaknál elengedhetetlen, hogy ilyen funkció rendelkezésre álljon, de ez is egy hamar beépíthető funkció lehet. Ami viszont a hiba ellenére mondható, hogy a webshop legfontosabb funkcióját, hogy a saját rendeléseit látni tudja a felhasználó az tökéletesen működik.

#### Hírlevél feliratkozás

Az oldalon jelenleg a hírlevélre való feliratkozás az teljes mértékben működés képtelen. Egy szimplán beégetett komponens, aminek célja az lenne, hogy valaki, aki lehet olyan felhasználó is, aki nem regisztrált az oldalra beírhat egy e-mail címet és arra folyamatosan kaphat aktuális hírlevelet esetleges akciókról és új termékekről. Ezeket az e-mail címeket az admin láthatná és kezelhetné.

### Admin oldali hibák

#### A kategóriák és típusok hozzáadása:

Jelenleg három kategória és három típus létezik az oldalon és ezek be vannak égetve az oldalra szóval minden egyes termék, amit feltölt az Admin az oldalra az ebbe a három kategóriába és típusba kell, hogy tartozzon. Az Admin csak ebből a három kategóriából és típusból választhat viszont a kódon belül korlátlan lehetőség nyílik ezeknek a kibővítésére.

#### **A termékek frissítésénél a felkapottság kezelése:**

Ez a hiba talán az egyik legnagyobb az oldalon. A lényege, hogy azt az adatot nem tudja módosítani a termékeken, hogy a termék felkapott vagy nem, ami nyilván idő közben dől el miközben az eladások történnek, hogy az vagy nem. Ez egy inkább kompromisszumot igénylő megoldás, hogy ennek kiküszöbölése miatt újra kell feltölteni a terméket és ott beállítani, hogy a termék felkapott vagy már nem.

## **Feljesztési lehetőségek**

A mai modern virtuális világban már a webshopok fejlettsége is már olyan szintre emelkedett, hogy nem csak tájékoztató jellegű kell legyen, hanem már sokszor szórakoztató is, hogy a felhasználók az oldalon maradjanak. Az amatőr kinézetű oldalak sokszor azt az érzetet is kelthetik a felhasználókban, hogy hamis vagy csaló oldalon esetleg egy webshop esetében bankkártya adatok halászatára létrehozott oldal. Ezek kiküszöbölése érdekében igyekeztünk mi is egy kinézetében és funkciójában egy szép és megbízható webshopot létrehozni. Viszont, mint minden weboldal szinte a végtelenségig fejleszthető, de a mi oldalunk vanna olyan dolgok, amik, ha fejlesztve lennének akkor sokkal több értéket tulajdoníthatna magának.

#### **Az elfelejtett jelszó:**

Ezt a funkciót sajnos nem sikerült megírni. A legfőbb oka ennek, hogy rengeteg megoldást próbáltunk ki arra, hogy hogyan lehetne az elfelejtett jelszót kezelni, mint példaként az OTP alapú e-mail hitelesítést vagy a teljesen link alapú visszaállítást. A projektünkben van egy tervezet, ami végül ki lett kommentelve, hogy a működést ne befolyásolja, de legfőbbként amiért abba maradt ennek a fejlesztése az azért történt mert sok nekifutásunk ellenére nem sikerült megoldást találni erre. Ezért inkább fejlesztési javaslatként említenénk meg a projektünkben.



### **AI alapú chatbot vagy tényleges 24/7 ügyfélszolgálat:**

Egyre nagyobb népszerűségnek örvendő mesterséges intelligencia sok weboldalon már, mint ügyfélszolgálati asszisztensként működik. Ez a funkció nagyon előnyös lehet olyan kis vállalkozásoknál, akik nem tudnak erre a pozícióra külön embert vagy személyzetet alkalmazni viszont tény, hogy az AI jelenleg nem képes teljes mértékben megoldani olyan problémákat amire nem programozták be vagy mondták meg előre, hogy így kell végrehajtása. Jelen projektünkben azért lenne hasznos, hogy az esetleges felhasználók letiltása vagy rendelési problémák kiküszöbölése miatt teljes felvilágosítást tudjon nyújtani a felhasználónak.

### **A felhasználói és Admin oldali hibák kiküszöbölése:**

Minden olyan funkció, ami hibaként jelentkezett a projekt folyamán azt nyilván, ha kijavításra fogunk akkor az is hatalmas tekintetben fejlesztené az oldal működését. Ilyenek például az Admin oldaláról a termékek felkapottságának szerkesztése, az új típusok hozzáadása amit nem a kódon belül lehetne ki egészíteni hanem az Admin adhatna hozzá új típusokat, utóbbi személyes véleményünk szerint azért nem egy esetlegesen megfelelő megoldás mert ha az Admin nem teljesen fejlett a számítógépes ismereteit tekintve nagyon könnyen előfordíthatja azt az esetőséget hogy egy majdnem teljes mértékben azonos elnevezésű kategóriát többször ad az oldalhoz hozzá pl. Cserepes és Cserép, Csokor és Csokros. Nyilván az egyik legégetőbb fejlesztési lehetőség a felhasználók oldalán, hogy minden egyes felhasználó tudja a saját adatait kezelni, mint az e-mail címet módosítani, jelszót módosítani, a saját fiókját törölni és esetleges extraként profilképet is állíthat be magának.

### **Termékek értékelése és egy kisebb közösségi funkció:**

A termékek értékelése is jelenleg csak egy beégetett értéket mutat minden egyes terméknel így a felhasználókban itt jöhet elő a legnagyobb bizonytalanság az oldal hitelességével kapcsolatban mert általában a vásárlók a termék értékelések alapján választják meg hogy mit választanak. Egy fajta közösség építési funkciót is elláthatna ez a fejlesztés mert egyes értékelésekre mások is reagálhatnak vagy alátámaszthatják az esetleges negatív vagy pozitív értékeléseket megoszthatják személyes véleményüket a termékekkel kapcsolatban.