



SocialCommerce

Fejlesztői dokumentáció

2024 | Budapest

A projektben résztvevők (fejlesztők)

Hajdara Patrik – BackEnd

Vinkovits János László – FrontEnd

Konzulens

Varga Zsófia - Webprogramozó

Tartalomjegyzék

A webalkalmazás célja	5
Licencek	5
Felhasznált technológiák	6
BackEnd	6
FrontEnd	6
Követelmények	7
Funkcionális	7
Nem funkcionális	8
Tervek	9
BackEnd terv	9
FrontEnd terv	10
Program leírása	15
Végpontok leírása	15
User	15
Shop	16
Product	17
Order	18
CouponCode	19
Controller-ek leírása	20
CouponCode	20
Order	21
Product	22
Shop	23
User	25
Komponens-ek leírása	28
Admin	28
Cart	29
Login	30
Payment	31
Signup	31
Shop	32
Hibajegyzék	33

Jövőbeli tervek	33
Tárgymutató	34
Időgen szavak	34
Ábra jegyzék	35
Projekthez használt képek	35
BackEnd	35
FrontEnd	36
Felhasznált irodalom	39
Kapcsolódó könyvek	39
FrontEnd	39
BackEnd	39
Köszönetnyilvánítás	40

A webalkalmazás célja

A SocialCommerce egy olyan közösségi platform, ahol bárki adhat el terméket, amit bárki megvásárolhat. Továbbá lehetőséget kínál az áruházak számára, melyek nem rendelkeznek webshoppal, hogy eladják termékeiket az online piacon gyorsan és egyszerűen. (Az eladás díjmentes.)

Licencek

A webalkalmazás zárt forráskódú. Tartalmainak felhasználása, másolása, illetve a teljes weboldal külsőleges használata (sajátként) tilos.

Az oldallal kapcsolatos változtatások jogát fenntartjuk.

Felhasznált technológiák

BackEnd:

BackEnd keretrendszer:

- ExpressJS (4.18.2)

Adatbázis kiszolgáló:

- MongoDB (6.0)

Objektumorientált programkönyvtár (ODM):

- Mongoose (7.0.0)

E-mail küldés:

- NodeMailer (6.9.1)

FrontEnd:

FrontEnd keretrendszer:

- ReactJS (18.2.0)

CSS kiszolgáló:

- Tailwind CSS (3.2.4)

JavaScript kiszolgáló:

- NodeJS (16)

Követelmények

Funkcionális:

Felhasználó (vásárló):

- A vásárlás regisztrációhoz kötött
- Regisztráció során kötelező megadni egy nevet, egy e-mailt és egy jelszót
- A jelszónak legalább 8 karakterből kell állnia
- Az első bejelentkezéshez kötelező a felhasználói fiók aktiválása a regisztráció után
- Bejelentkezéshez kötelező megadni egy e-mailt és egy jelszót
- A felhasználói fiók csak regisztráció vagy bejelentkezés után tekinthető meg
- A címek mentése (rendeléshez), értékelések írása (termékekhez) felhasználói fiókhoz kötött
- Értékelések írása csak a rendelés lezárását követően lehetséges (rendelés állapota > kiszállítva)
- Rendelés során kötelező megadni a szállítási adatokat, illetve a fizetési módot is
- A termékek fiók létrehozása vagy bejelentkezés nélkül is megtekinthetőek
- Termék mentése a kedvencekhez, illetve hozzáadása a kosárba regisztráció vagy bejelentkezés nélkül is lehetséges

Eladó:

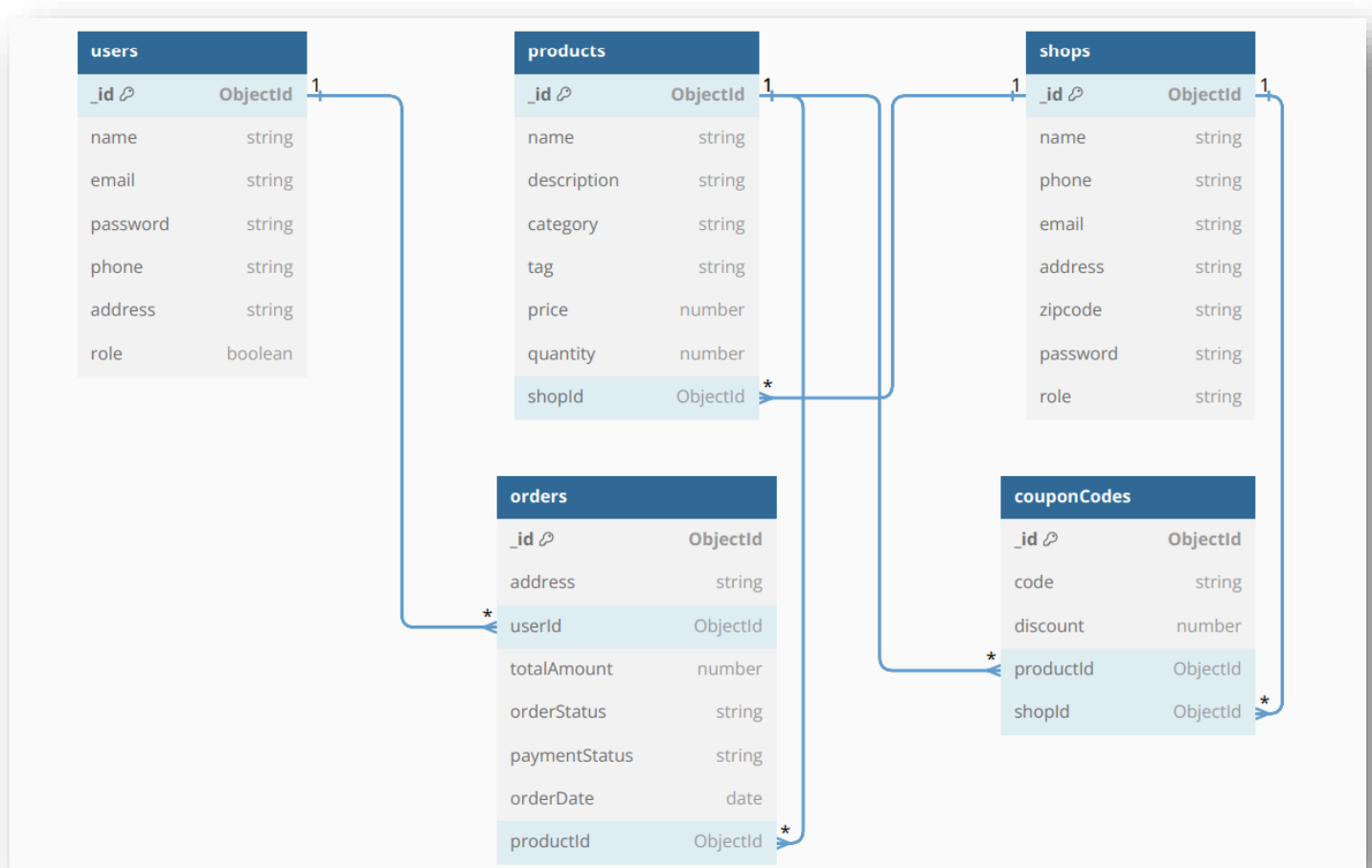
- Termékek eladásához eladói fiók regisztrálása szükséges
- Regisztráció során kötelező megadni az eladó vagy a vállalkozás nevét, egy telefonszámot, egy e-mailt, az eladó vagy vállalkozás címét, egy irányítószámot és egy jelszót
- A jelszónak legalább 8 karakterből kell állnia
- Az első bejelentkezéshez kötelező az eladói fiók aktiválása a regisztráció után
- Bejelentkezéshez kötelező megadni egy e-mailt és egy jelszót
- Az eladói profil csak regisztráció vagy bejelentkezés után tekinthető meg
- Termék feltöltése során kötelező megadni a termék nevét, kategóriáját, leírását, árát, mennyiségét, illetve 1 képet a termékről
- Kuponkód létrehozása csak regisztrált eladóként lehetséges

Nem funkcionális:

- A BackEnd az ExpressJS (4.18.2) használatával készült
- Az adatbázis kiszolgáló a MongoDB (6.0)
- Az e-mailek a NodeMailer (6.9.1) használatával kerülnek elküldésre
- Az e-mail kiszolgáló a Google 'Alkalmazásjelszavak' funkciója
- A FrontEnd a ReactJS (18.2.0) használatával készült
- A CSS kiszolgáló a Tailwind CSS (3.2.4)
- A projekt a NodeJS (16) használatával készült




Tervek

BackEnd terv:



FrontEnd terv:

- Főoldal

Logo	Termék keresése...	Eladói profil					
Kategóriák	Kezdőlap	Ajánlatok	Termékek	GYIK			
Főoldali kép							
Kategória választó							
Ajánlataink, felkapott termékek							
Footer							

- Bejelentkezés (felhasználó)

	Logo	
	Jelentkezz be a fiókodba	
	E-mail	
	Jelszó	
	Bejelentkezés	

- Regisztráció (felhasználó)

	Logo	
	Hozdd létre a fiókodat	
	Név	
	E-mail	
	Jelszó	
	Regisztráció	

- Bejelentkezés (eladó)

Logo

Jelentkezz be eladói fiókodba

E-mail

Jelszó

Bejelentkezés

- Regisztráció (eladó)

	Logo	
	Hozdd létre a fiókodat	
	Név vagy vállalkozás neve	
	Telefonszám	
	E-mail	
	Lakhely vagy vállalkozás címe	
	Írányítószám	
	Jelszó	
	Profilkép feltöltése	
	Regisztráció	

Program leírása

Végpontok leírása

User:

- Végpont neve: /create-user
- Bejövő paraméter: **name (string), email (string), password (string), avatar (string)**
- Hova mutat: **Fiók létrehozása**

- Végpont neve: /activation
- Bejövő paraméter: **activation token (string)**
- Hova mutat: **Fiók aktiválása**

- Végpont neve: /login-user
- Bejövő paraméter: **email (string), password (string)**
- Hova mutat: **Bejelentkezés**

- Végpont neve: /update-user-info
- Bejövő paraméter: **email (string), password (string), phoneNumber (string), name (string)**
- Hova mutat: **Adatok frissítése**

- Végpont neve: /update-avatar
- Bejövő paraméter: **avatar (string)**
- Hova mutat: **Avatar frissítése**

- Végpont neve: /update-user-password
 - Bejövő paraméter: **oldPassword (string), newPassword (string), confirmPassword (string)**
 - Hova mutat: **Jelszó frissítése**
-
- Végpont neve: /update-user-addresses
 - Bejövő paraméter: **addressType (string), id (string)**
 - Hova mutat: **Címek hozzáadása, törlése**
-
- Végpont neve: /admin-all-users
 - Bejövő paraméter: -
 - Hova mutat: **Összes felhasználó megtekintése (admin esetében)**
-
- Végpont neve: /delete-user
 - Bejövő paraméter: **id (string)**
 - Hova mutat: **Felhasználó törlése**

Shop:

- Végpont neve: /create-shop
 - Bejövő paraméter: **email (string), avatar (string), name (string), password (string), address (string), phoneNumber (string), zipCode (string)**
 - Hova mutat: **Fiók létrehozása**
-
- Végpont neve: /activation
 - Bejövő paraméter: **activation token (string)**
 - Hova mutat: **Fiók aktiválása**

- Végpont neve: **/login-shop**
 - Bejövő paraméter: **email (string), password (string)**
 - Hova mutat: **Bejelentkezés**
-
- Végpont neve: **/update-seller-info**
 - Bejövő paraméter: **name (string), description (string), address (string), phoneNumber (string), zipCode (string)**
 - Hova mutat: **Adatok frissítése**
-
- Végpont neve: **/update-shop-avatar**
 - Bejövő paraméter: **avatar (string)**
 - Hova mutat: **Avatar frissítése**
-
- Végpont neve: **/admin-all-sellers**
 - Bejövő paraméter: **-**
 - Hova mutat: **Összes eladó megtekintése (admin esetében)**
-
- Végpont neve: **/delete-seller**
 - Bejövő paraméter: **id (string)**
 - Hova mutat: **Eladó törlése**

Product:

- Végpont neve: **/create-product**
 - Bejövő paraméter: **shopId, images**
 - Hova mutat: **Termék létrehozása**
-
- Végpont neve: **/get-all-products-shop**
 - Bejövő paraméter: **id (string)**
 - Hova mutat: **Összes termék megtekintése (eladó esetében)**

Order:

- Végpont neve: **/create-order**
- Bejövő paraméter: **cart, shippingAddress, user, totalPrice, paymentInfo**
- Hova mutat: **Rendelés létrehozása**

- Végpont neve: **/get-all-orders**
- Bejövő paraméter: **userId (string)**
- Hova mutat: **Rendelések megtekintése (vásárló esetében)**

- Végpont neve: **/get-seller-all-orders**
- Bejövő paraméter: **shopId (string)**
- Hova mutat: **Rendelések megtekintése (eladó esetében)**

- Végpont neve: **/update-order-status**
- Bejövő paraméter: **id (string)**
- Hova mutat: **Rendelés státuszának frissítése**

CouponCode:

- Végpont neve: **/create-coupon-code**
 - Bejövő paraméter: **name (string)**
 - Hova mutat: **Kupon létrehozása**
-
- Végpont neve: **/get-coupon**
 - Bejövő paraméter: **id (string)**
 - Hova mutat: **Összes kupon megtekintése (eladó esetében)**
-
- Végpont neve: **/delete-coupon**
 - Bejövő paraméter: **id (string)**
 - Hova mutat: **Kupon törlése**

Controller-ek leírása

CouponCode:

- Végpont: **/create-coupon-code**
 - Metódus: POST
 - Működés: Ellenőrzi, hogy létezik-e már ilyen nevű kuponkód. Ha nem, létrehozza és visszaadja a létrehozott kuponkódot
 - Visszatérési érték: Létrehozott kuponkód objektum
-
- Végpont: **/get-coupon**
 - Metódus: GET
 - Paraméterek: id (az eladó azonosítója)
 - Működés: Lekéri az adott eladóhoz tartozó összes kuponkódot
 - Visszatérési érték: Eladóhoz tartozó kuponkódok listája
-
- Végpont: **/delete-coupon**
 - Metódus: DELETE
 - Paraméterek: id (kuponkód azonosítója)
 - Működés: Törli a megadott azonosítójú kuponkódot
 - Visszatérési érték: Sikeres törlés üzenet
-
- Végpont: **/get-coupon-value**
 - Metódus: GET
 - Paraméterek: name (kuponkód neve)
 - Működés: Lekéri a megadott névvel rendelkező kuponkód értékét
 - Visszatérési érték: Kuponkód objektum

Order:

- Végpont: **/create-order**
 - Metódus: POST
 - Működés: A kosár tartalmát alapján létrehozza a rendeléseket, csoportosítva azokat eladók szerint
 - Visszatérési érték: Létrehozott rendelések objektumok listája
-
- Végpont: **/get-all-orders/:userId**
 - Metódus: GET
 - Paraméterek: userId (a felhasználó azonosítója)
 - Működés: Lekéri az adott felhasználóhoz tartozó összes rendelést
 - Visszatérési érték: Felhasználóhoz tartozó rendelések listája
-
- Végpont: **/get-seller-all-orders/:shopId**
 - Metódus: GET
 - Paraméterek: shopId (az eladó azonosítója)
 - Működés: Lekéri az adott eladóhoz tartozó összes rendelést
 - Visszatérési érték: Eladóhoz tartozó rendelések listája
-
- Végpont: **/update-order-status/:id**
 - Metódus: PUT
 - Paraméterek: id (rendelés azonosítója)
 - Működés: Frissíti a megadott rendelés státuszát az eladó által megadott értékre
 - Visszatérési érték: Frissített rendelés objektum
-
- Végpont: **/admin-all-orders**
 - Metódus: GET
 - Működés: Lekéri az összes rendelést az adatbázisból
 - Visszatérési érték: Az összes rendelés listája

Product:

- Végpont: **/create-product**
 - Metódus: POST
 - Működés: Létrehozza az új terméket az adott eladóhoz. Feltölti a termékhez kapcsolódó képeket
 - Visszatérési érték: Létrehozott termék objektum
-
- Végpont: **/get-all-products-shop/:id**
 - Metódus: GET
 - Paraméterek: id (az eladó azonosítója)
 - Működés: Lekéri az adott eladóhoz tartozó üzletben található összes terméket
 - Visszatérési érték: Eladóhoz tartozó termékek listája
-
- Végpont: **/delete-shop-product/:id**
 - Metódus: DELETE
 - Paraméterek: id (termék azonosítója)
 - Működés: Törli az adott azonosítójú terméket az adatbázisból
 - Visszatérési érték: Sikeres törlés üzenet
-
- Végpont: **/get-all-products**
 - Metódus: GET
 - Működés: Lekéri az összes terméket az adatbázisból, rendezve a legfrissebbtől a legrégebbig
 - Visszatérési érték: Az összes termék listája
-
- Végpont: **/create-new-review**
 - Metódus: PUT
 - Működés: Létrehoz vagy frissít egy értékelést egy adott termékhez, amihez egy felhasználó tartozik. Az értékelés átlagát is frissíti
 - Visszatérési érték: Sikeres értékelés üzenet

- Végpont: **/admin-all-products**
- Metódus: GET
- Működés: Lekéri az összes terméket az adatbázisból, rendezve a legfrissebbtől a legrégebbig. Csak az admin jogosultsággal rendelkező felhasználók számára érhető el
- Visszatérési érték: Az összes termék listája

Shop:

- Végpont: **/create-shop**
 - Metódus: POST
 - Működés: Az alkalmazás egy aktiváló tokent hoz létre, ami tartalmazza a megadott adatokat, majd a token biztonságos aláírással ellátva kerül elküldésre a megadott e-mail címre
 - Visszatérési érték: Sikeres üzenet az aktivációs e-mail elküldéséről
-
- Végpont: **/activation**
 - Metódus: POST
 - Működés: Az eladó az e-mailben található aktiváló linkre kattintva elküldi a tokent a szervernek. A szerver ellenőrzi a token érvényességét, majd visszafejti a benne tárolt adatokat. Az adatok alapján a szerver létrehozza a fiókot az adatbázisban
 - Visszatérési érték: Aktivált eladó objektum és token
-
- Végpont: **/login-shop**
 - Metódus: POST
 - Működés: Bejelentkezés az eladói fiókba az e-mail cím és jelszó alapján
 - Visszatérési érték: Sikeres bejelentkezés üzenet és token

- Végpont: **/getSeller**
 - Metódus: GET
 - Működés: Lekéri az aktuális bejelentkezett eladó adatait
 - Visszatérési érték: Az aktuális bejelentkezett eladó adatai
-
- Végpont: **/logout**
 - Metódus: POST
 - Működés: Kijelentkezteti az aktuális bejelentkezett eladót
 - Visszatérési érték: Sikeres kijelentkezés üzenet
-
- Végpont: **/get-shop-info/:id**
 - Metódus: GET
 - Paraméterek: id (az eladó azonosítója)
 - Működés: Lekéri az adott azonosítójú eladó adatait
 - Visszatérési érték: Az adott azonosítójú eladó adatai
-
- Végpont: **/update-shop-avatar**
 - Metódus: PUT
 - Működés: Frissíti az aktuális bejelentkezett eladó avatar képét
 - Visszatérési érték: Frissített eladó objektum
-
- Végpont: **/update-seller-info**
 - Metódus: PUT
 - Működés: Frissíti az aktuális bejelentkezett eladó adatait
 - Visszatérési érték: Frissített eladó objektum

- Végpont: **/admin-all-sellers**
 - Metódus: GET
 - Működés: Lekéri az összes eladót, csak az admin jogosultsággal rendelkező felhasználók számára érhető el
 - Visszatérési érték: Az összes eladó listája
-
- Végpont: **/delete-seller/:id**
 - Metódus: DELETE
 - Paraméterek: id (az eladó azonosítója)
 - Működés: Törli az adott azonosítójú eladó profilját az adatbázisból
 - Visszatérési érték: Sikeres törlés üzenet

User:

- Végpont: **/create-user**
 - Metódus: POST
 - Működés: Az alkalmazás egy aktiváló tokent hoz létre, ami tartalmazza a megadott adatokat, majd a token biztonságos aláírással ellátva kerül elküldésre a megadott e-mail címre
 - Visszatérési érték: Sikeres üzenet az aktivációs e-mail elküldéséről
-
- Végpont: **/activation**
 - Metódus: POST
 - Működés: A felhasználó az e-mailben található aktiváló linkre kattintva elküldi a tokent a szervernek. A szerver ellenőrzi a token érvényességét, majd visszafejti a benne tárolt adatokat. Az adatok alapján a szerver létrehozza a fiókot az adatbázisban
 - Visszatérési érték: Aktivált felhasználó objektum és token

- Végpont: **/login-user**
 - Metódus: POST
 - Működés: Bejelentkezés a felhasználó fiókjába az e-mail cím és jelszó alapján
 - Visszatérési érték: Sikeres bejelentkezés üzenet és token
-
- Végpont: **/getuser**
 - Metódus: GET
 - Működés: Lekéri az aktuális bejelentkezett felhasználó adatait
 - Visszatérési érték: Az aktuális bejelentkezett felhasználó adatai
-
- Végpont: **/logout**
 - Metódus: POST
 - Működés: Kijelentkezteti az aktuális bejelentkezett felhasználót
-
- Végpont: **/update-user-info**
 - Metódus: PUT
 - Működés: Frissíti az aktuális bejelentkezett felhasználó adatait
 - Visszatérési érték: Frissített felhasználó objektum
-
- Végpont: **/update-avatar**
 - Metódus: PUT
 - Működés: Frissíti az aktuális bejelentkezett felhasználó avatar képét
 - Visszatérési érték: Frissített felhasználó objektum
-
- Végpont: **/update-user-addresses**
 - Metódus: PUT
 - Működés: Frissíti az aktuális bejelentkezett felhasználó címeit
 - Visszatérési érték: Frissített felhasználó objektum

- Végpont: **/delete-user-address/:id**
 - Metódus: DELETE
 - Paraméterek: id (a cím azonosítója)
 - Működés: Törli az adott azonosítójú felhasználó címét az adatbázisból
 - Visszatérési érték: Sikeres törlés üzenet
-
- Végpont: **/update-user-password**
 - Metódus: PUT
 - Működés: Frissíti az aktuális bejelentkezett felhasználó jelszavát
 - Visszatérési érték: Sikeres üzenet a jelszó frissítéséről
-
- Végpontok: **/user-info/:id**
 - Metódus: GET
 - Paraméterek: id (a felhasználó azonosítója)
 - Működés: Lekéri az adott azonosítójú felhasználó adatait
 - Visszatérési érték: Az adott azonosítójú felhasználó adatai
-
- Végpont: **/admin-all-users**
 - Metódus: GET
 - Működés: Lekéri az összes felhasználót, csak az adminisztrátori jogosultsággal rendelkező felhasználók számára elérhető
 - Visszatérési érték: Az összes felhasználó listája
-
- Végpont: **/delete-user/:id**
 - Metódus: DELETE
 - Paraméterek: id (a felhasználó azonosítója)
 - Működés: Törli az adott azonosítójú felhasználó profilját az adatbázisból
 - Visszatérési érték: Sikeres törlés üzenet

Komponens-ek leírása

Admin:

- AllProducts komponens:

- Meghívja a backenden a "/product/admin-all-products" végpontot GET kéréssel
- Nem vár bejövő paramétert
- Az adatokat card-ban jeleníti meg, saját CSS segítségével
- A backendtől kapott termékobjektumok listáját küldi vissza

- AllSellers komponens:

- Meghívja a backenden a "/shop/admin-all-sellers" végpontot GET kéréssel
- Nem vár bejövő paramétert
- Az adatokat card-ban jeleníti meg, saját CSS segítségével
- A backendtől kapott eladóobjektumok listáját küldi vissza

- AllUsers komponens:

- Meghívja a backenden a "/user/admin-all-users" végpontot GET kéréssel
- Nem vár bejövő paramétert
- Az adatokat card-ban jeleníti meg, saját CSS segítségével
- A backendtől kapott felhasználóobjektumok listáját küldi vissza

Cart:

- Cart komponens:

- Megjeleníti a felhasználó kosarát
- A felhasználói kosár adatait egy táblázatban jeleníti meg, amely tartalmazza a termékek nevét, mennyiségét, árát és a teljes vásárlási összeget
- Lehetőséget biztosít a termékek mennyiségének módosítására és eltávolítására a kosárból
- A vásárlás gombra kattintva lehetőség van a pénztár oldalra navigálni

- CartSingle komponens:

- Egyetlen terméket jelenít meg a kosárban
- Lehetőséget biztosít a termék mennyiségének módosítására (növelés/csökkentés) és a termék eltávolítására a kosárból
- Megjeleníti a termék nevét, árát, mennyiségét és a végösszeget
- Tartalmaz egy képet a termékről és egy gombot az eltávolításhoz

- Checkout komponens:

- A pénztár oldal, ahol a felhasználó megadhatja a szállítási információkat és érvényesíthet kuponkódokat
- Megjeleníti a felhasználó által kiválasztott termékek végösszegét, valamint a kedvezményeket
- Lehetőséget biztosít a felhasználónak, hogy megadja a szállítási címet és kuponkódokat érvényesítsen
- A fizetés gombra kattintva lehetőség van a fizetés oldalra navigálni

- ShippingInfo komponens:

- Megjeleníti a szállítási információkat, mint például a név, e-mail cím, telefonszám, ország, város, cím, stb
- Lehetőséget biztosít a felhasználónak, hogy manuálisan adjon meg a szállítási információkat, vagy válasszon egy előzőleg mentett címet

- CartData komponens:

- Megjeleníti a kosár végösszegét és a kedvezményeket.
- Lehetőséget biztosít a felhasználónak, hogy meg adjon egy kuponkódot és érvényesítse azt

Login:

- Login komponens:

- A Login komponens egy egyszerű bejelentkező űrlapot jelenít meg, ahol a felhasználó megadhatja az e-mail címét és a jelszavát
- A jelszó mezőhöz további funkciók tartoznak, mint például a jelszó megjelenítése/eltakarása ikonra kattintva
- A felhasználó bejelentkezése után sikeres bejelentkezés esetén átirányítás történik a főoldalra, egyébként hibaüzenet jelenik meg
- A regisztrációra való navigálásra is lehetőség van

Payment:

- Payment komponens:

- Megjeleníti a fizetési lehetőségeket és a kosár tartalmát.
- Nem vár bejövő paramétert
- A felhasználó által választott fizetési módnak megfelelően kezeli a fizetést
- A backend felé a /order/create-order végpontra POST kérést intéz a megrendeléssel
- A kosár és a megrendelési adatokat tárolja a helyi tárolóban (localStorage)

Signup:

- Signup komponens:

- Űrlap beküldésekor az adatokat POST kéréssel elküldi a backendnek a "/user/create-user" végpontra
- Bejövő paraméterek: name, email, password, avatar
- Az űrlap bejövő paramétereit a useState hook segítségével kezeli és tárolja
- Az avatar feltöltését is kezeli, és az input mező tartalmát base64 kódra alakítja át
- A backendtől kapott választ megjeleníti egy toast üzenetben

Shop:

- ShopCreate komponens:

- Új eladói fiók létrehozását teszi lehetővé
- Az adatokat a /shop/create-shop végpontra POST kéréssel küldi el
- Bejövő paraméterek: name (név vagy vállalkozás neve), email, phoneNumber (telefonszám), address (lakhely vagy vállalkozás címe), zipCode (irányítószám), password (jelszó), avatar (profilkép)
- Az adatokat input mezőkön keresztül lehet megadni, a profilképet pedig feltöltheti a felhasználó
- A backendtől érkező választ megjeleníti egy toast segítségével

- ShopLogin komponens:

- Meghívja a backenden a "/shop/login-shop" végpontot POST kéréssel
- Bejövő paraméterek: email, password
- A felhasználó által megadott email és jelszó alapján küld kérést a backend felé
- Sikeres bejelentkezés esetén a "/dashboard" útvonalra navigál
- Hiba esetén toast üzenet jelenik meg a kapott hibaüzenettel

Hibajegyzék

Jövőbeli tervek

- Regisztrációnál nem lesz kötelező profilképet feltölteni (minden felhasználó kap egy alapértelmezett profilképet)
- További jelszó követelmények a biztonság érdekében
- E-mail szabályozása (pl. johndoe@example.com)
- Fizetési módok bővítése, hogy a rendelések online is fizethetők legyenek bankkártya, PayPal, vagy Stripe segítségével
- Rendelések archiválása (korábbi rendelések esetében)
- Kuponkódok a készlet erejéig (csak addig legyenek felhasználhatóak, amíg érvényesek, vagy ameddig egy termék elérhető)
- Chat funkció (a felhasználó kapcsolatba tudjon lépni az eladóval közvetlenül az oldalon a chat funkciót választva)
- Felhasználók és eladók esetében korábbi adatok tárolása (bármilyen probléma esetére)
- Checkbox az admin felületen, hogy könnyedén admin jogosultságot lehessen adni, vagy megszüntetni
- Saját 404-es hibakód, ha a keresett oldal már nem érhető el
- Ha egy termék esetében nincs kép, akkor valamilyen ikon jelenjen meg a termék nevével
- Regisztráció és bejelentkezés a Google segítségével
- Modernebb, jövőbe mutató dizájn

Tárgymutató

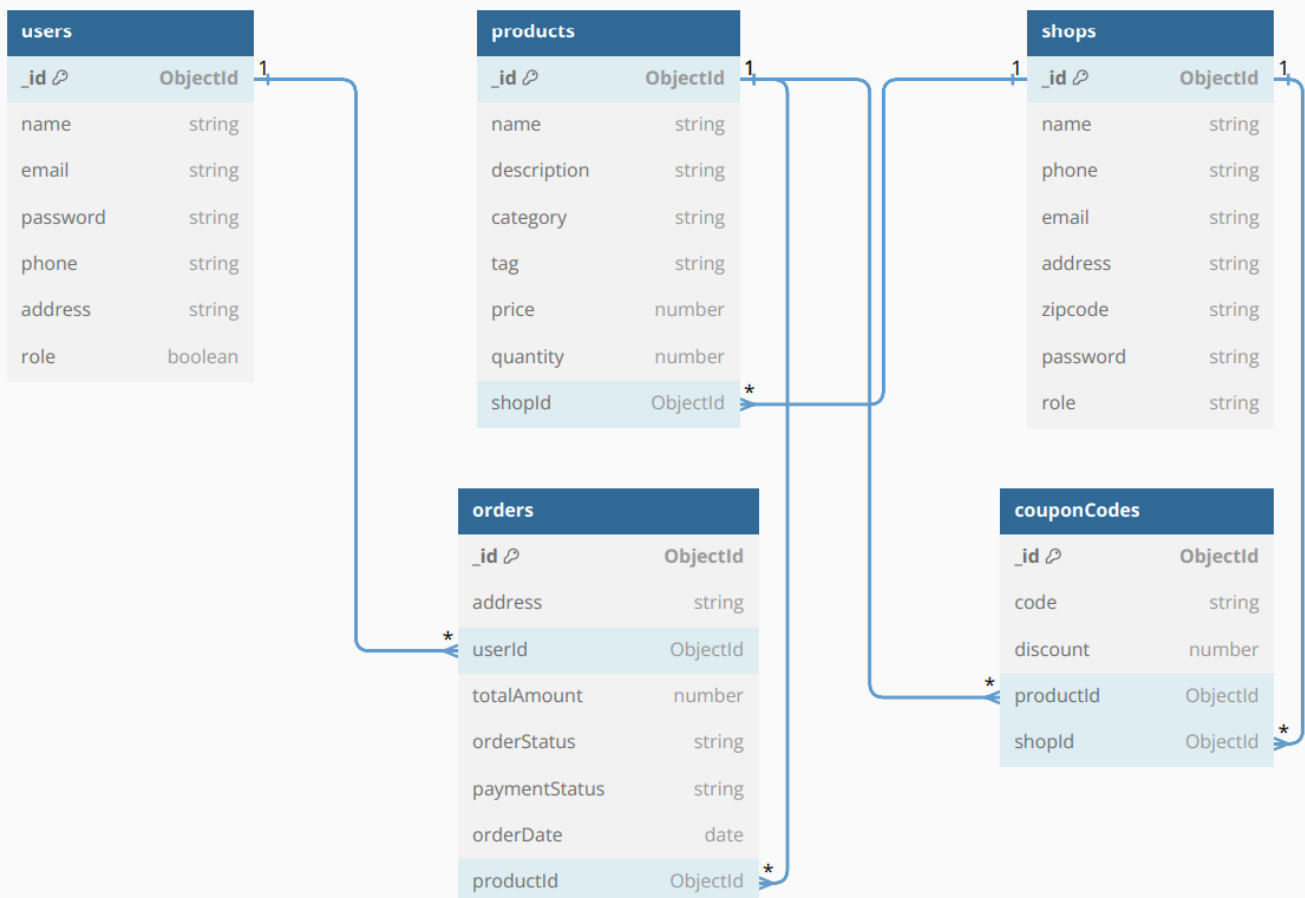
Idegen szavak

- Felhasznált technológiák: A projekt készítése során alkalmazott olyan programnyelvek, melyek lehetővé tették a webalkalmazás és a hozzá kapcsolódó szolgáltatások létrehozását és működését
- FrontEnd: Az a felhasználói felület, amelyen keresztül a felhasználók interakcióba léphetnek egy számítógépes rendszerrel vagy alkalmazással
- BackEnd: Az az alkalmazás rész, amely a háttérben fut, adatokat tárol, feldolgozza és kezeli, valamint kapcsolatot biztosít a FrontEnd és a szerver között, illetve a felhasználói kérések és válaszok között
- FrontEnd terv: A felhasználói felület és élmény kialakítását jelenti egy szoftver vagy webalkalmazás számára, beleértve a dizájnt, navigációt és felhasználói interakciókat
- BackEnd terv: A szoftver vagy webalkalmazás háttérfolyamatainak és adatkezelési stratégiájának megtervezését jelenti, beleértve az adatbázis-kezelést és a szerveroldali funkcionalitást

Ábra jegyzék

Projekthez használt képek

BackEnd



FrontEnd

Logo	Termék keresése...	Eladói profil					
Kategóriák	Kezdőlap	Ajánlatok	Termékek	GYIK			
Főoldali kép							
Kategória választó							
Ajánlataink, felkapott termékek							
Footer							

	Logo	
	Jelentkezz be a fiókodba	
	E-mail	
	Jelszó	
	Bejelentkezés	

	Logo	
	Hozdd létre a fiókodat	
	Név	
	E-mail	
	Jelszó	
	Regisztráció	

Logo

Jelentkezz be eladói fiókodba

E-mail

Jelszó

Bejelentkezés

Logo

Hozdd létre a fiókodat

Név vagy vállalkozás neve

Telefonszám

E-mail

Lakhely vagy vállalkozás címe

Irányítószám

Jelszó

Profilkép feltöltése

Regisztráció

Felhasznált irodalom

Kapcsolódó könyvek

FrontEnd:

- "Modern JavaScript fejlesztés" - Réti Zsolt – 2018
- "Az új HTML5 és CSS3 - Frontend fejlesztés alapjai" - István László – 2015
- "JavaScript & jQuery: Interaktív webfejlesztés" - Jon Duckett – 2014
- "UI és UX design alapok" - Aaron Walter - 2010

BackEnd:

- "Node.js a gyakorlatban" - Ákos Bodnár – 2015
- "Tiszta kód: Az agilis szoftverkézművesség kézikönyve" – Robert C. Martin – 2021

Köszönetnyilvánítás

Szeretnénk köszönetet mondani az alábbi tanárainknak, akik nélkül nem jöhetett volna létre a SocialCommerce projekt

Rékási József (szoftverfejlesztés)

Jáger Attila (szoftverfejlesztés)

Sallai András (szoftverfejlesztés)

Bálint Dezső (szoftverfejlesztés)

Togyerisko Erika (szakmai angol)

Horváth Boglárka (szakmai angol)

Továbbá konzulensünknek, aki véleményezte és kipróbálta a projektet,
mint felhasználó

Varga Zsófia (webprogramozó)

© 2024 | Budapest
SocialCommerce