

Budapesti Műszaki Szakképzési Centrum

Neumann János Informatikai Technikum

Szakképesítés neve: Szoftverfejlesztő és -tesztelő

száma: 5-0613-12-03

VIZSGAREMEK

ArtShop

Nyikon Zoltán Ferenc, Babik Valter

2/14. A

Budapest, 2022.

Bevezetés

Ezzel a dokumentummal szeretnénk bemutatni azt, hogy hogyan készítettük el a vizsgaremekünket, milyen problémákba ütköztünk a fejlesztés során és hogy hogyan teszteltük.

Két főből álló csapatunkkal, április elején kezdtünk neki weboldalunk aktív fejlesztésébe. Nagy vonalakban már előtte sikerült kitalálunk a projektünk tematikáját, illetve az adatbázisának felépítését, így már csak a fejlesztésre és tesztelésre kellett koncentrálnunk.

A csapatmunkát sikerült gördülékenyen megoldanunk. A munkánkat segítették a Trello, Discord és a GitHub alkalmazásai. A Trello segítségével, könnyedén tudtunk megszerezni és felosztani egymás között a feladatokat. Így mindenki tudta, hogy milyen feladatok várnak rá és elkerültük azt, hogy véletlenül ugyanazon dolgozzunk egyszerre. A kapcsolatot első sorban Discordon keresztül tartottuk egymással, ahol napi szinten megbeszéltük, hogy ki éppen mit haladt a projekttel, elakadt-e valamiben és kell-e neki segítség. A programkódot pedig GitHubon keresztül osztottuk meg egymással.

Az ArtShopot olyan otthoni művészeknek és alkotóknak terveztük, akik szeretnék a következő szintre lépni a művészi készségeikkel. Lehetőséget biztosítunk számukra, hogy értékes műveiket eladhassák. Az oldalunkon a kategóriák tágas választéka miatt, szinte bármilyen mű értékesítésére lehetőség adódik. Továbbá az oldal nagyon jó felhasználói élményt nyújt a művészet kedvelői számára, akik inkább vásárolnának.

Emellet a szoftverünk még egy asztali adminfelülettel is rendelkezik, ahol az adminisztrátorok szinte teljes körű felügyeleti jogot kaptak a rendszer felett, egy felhasználóbarát és könnyen átlátható felületen.

Tartalom

1. Az ötlet megszületése

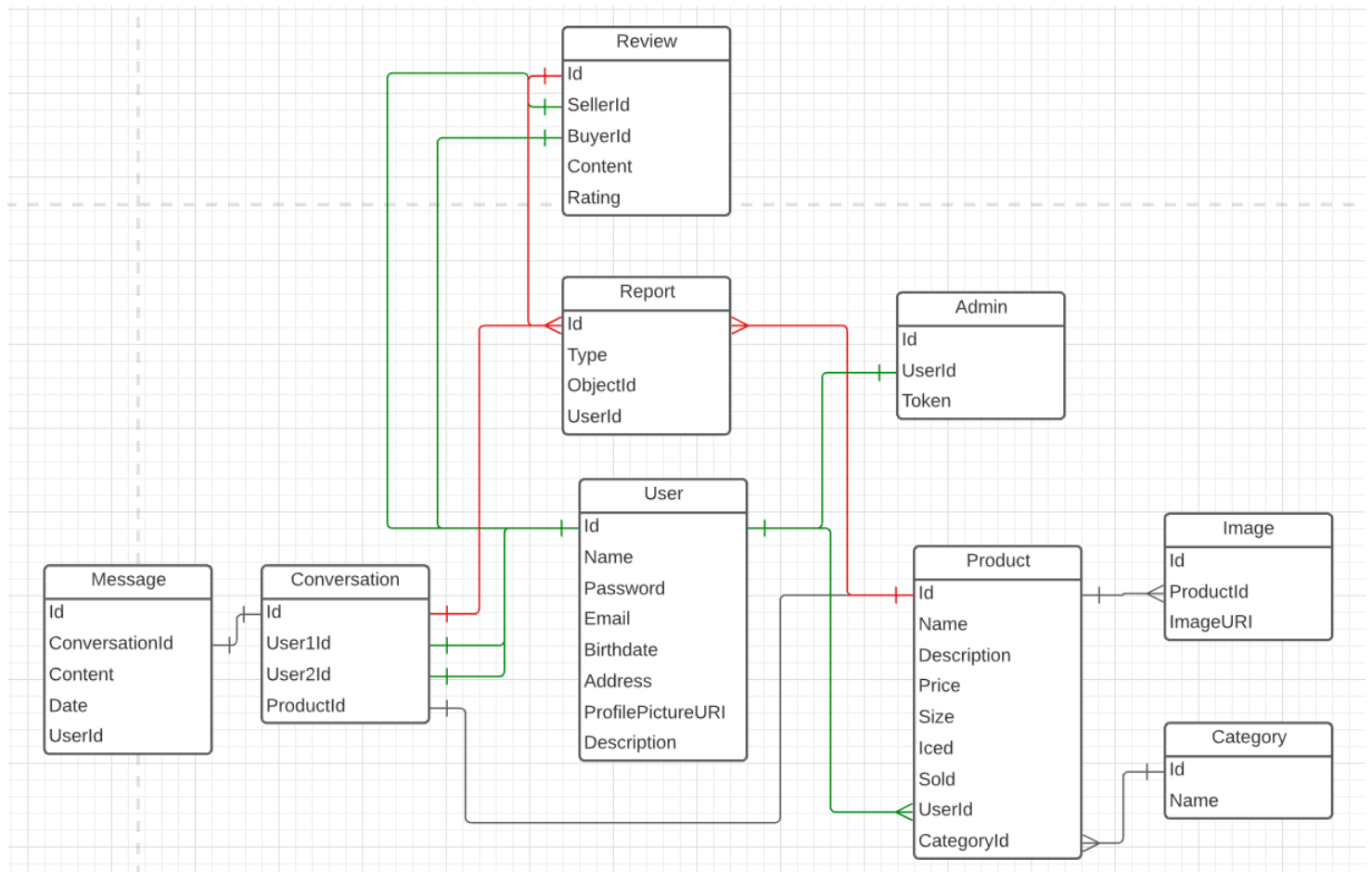
Ismerősi körünkben rengeteg ember foglalkozik otthon, hobbi szinten művészettel. Vannak akik festményeket készítenek, vannak akik ruhadarabokat de akadnak olyanok is akik apró ékszereket csinálnak.

Több embertől hallottuk azt, hogy szeretnének pénzt is csinálni, szeretett szenvedélyükből. Így nem is volt kérdés, hogy miről fog szólni a vizsgaremekünk. Ezért egy olyan szoftvert terveztünk ami lehetővé teszi számukra a termékeik eladását. Illetve ha kedvet kapnak mások műveihez, akkor azt is megvásárolhatják.

2. Adatbázis

Miután kitaláltuk, hogy mit is szeretnénk csinálni, egyből létre is hoztuk a Trello táblánkat a projekthez. Első feladatunk az adatbázis tervezése volt. Ezt a lépést gondosan kellett, megterveznünk, mert a projektünk további részéi mind erre épültek rá. Odafigyeltünk a redundanciára és figyelembe vettük a normálformákat is.

A tervezést egy Lucidchart nevű programmal vittük véghez. Ebben a szoftverben nagyon könnyen meg tudtuk tervezni adatbázisunk . Alább a Lucidchartban megtervezett ER diagram látható.



2.1. User tábla

A User táblához számos adatbázis kapcsolat tartozik. Egy a többhöz (1:N) kapcsolatban van a **Conversation** tábla *SellerId* és *BuyerId* mezőjével, a **Review** tábla *SellerId* és *BuyerId* mezőjével, a **Product** tábla *UserId* mezőjével és a **Report** tábla *UserId* mezőjével. Továbbá Egy az egy egyhez (1:1) kapcsolatban van az **Admin** tábla *UserId* mezőjével.

A User táblába a következő mezőket vettük fel:

- Id - A felhasználó egyedi azonosítója
- Name - A felhasználó neve
- Password - A felhasználó jelszava

- Email - A felhasználó email címe (egyedi érték)
- Birthdate - A felhasználó születési dátuma
- Address - A felhasználó lakcíme
- ProfilePictureURI - A felhasználó profil képének elérési útvonala
- Description - A felhasználó magáról alkotott leírása

2.2. Product tábla

A Product tábla is szintén számos adatbázis kapcsolattal rendelkezik. Egy a többhöz (1:N) kapcsolatot ápol a **Conversation** tábla *ProductId* mezőjével, a **Report** tábla *ObjectId* mezőjével és az **Image** tábla *ProductId* mezőjével. Ezen kívül még több az egyhez (N:1) kapcsolatban van a **User** táblával a *UserId* mezőjén keresztül és a **Category** táblával a *CategoryId* mezőjén keresztül..

A Product táblába a következő mezőket vettük fel:

- Id - A termék egyedi azonosítója
- Name - A termék neve
- Description - A termék leírása
- Price - A termék ára dollárban
- Size - A termék mérete. Ruhadarab esetén lehet S, M, stb...,

tárgyak esetén lehet centiméterben megadott érték is.

- Iced - Logikai érték, hogy a termék le van-e fagyasztva vagy sem
- Sold - Logikai érték, hogy a termék el van-e adva vagy sem
- UserId - A termék készítőjének egyedi azonosítója
- CategoryId - A kategória egyedi azonosítója

2.3. Category tábla

A Category tábla kizárólag egy táblával, a Product táblával van egy a többhöz kapcsolatban (1:N) a **Product** tábla *CategoryId* mezőjével.

A Category táblába a következő mezőket vettük fel:

- Id - A kategória egyedi azonosítója
- Name - A kategória neve

2.4. Conversation tábla

A Conversation tábla 4 másik táblával van kapcsolatban. Egy a többhöz (1:N) kapcsolatot ápol a **Message** tábla *ConversationId* mezőjével és a **Report** tábla *ObjectId* mezőjével. Ezekon kívül még több az egyhez (N:1) kapcsolatot ápol a **User** táblával a *SellerId* és a *BuyerId* mezőkön keresztül és **Product** táblával a *ProductId* mezőn keresztül.

A Conversation táblába a következő mezőket vettük fel:

- Id - A beszélgetés egyedi azonosítója
- SellerId - A beszélgetés résztvevői közül az eladó egyedi azonosítója
- BuyerId - A beszélgetés résztvevői közül a vevő egyedi azonosítója
- ProductId - A beszélgetéshez kapcsolódó termék egyedi azonosítója

2.5. Message tábla

A Message tábla 2 másik táblával van kapcsolatban. Több az egyhez (N:1) kapcsolatot ápol a **Conversation** táblával a *ConversationId* mezőn keresztül és a **User** táblával a *UserId* mezőn keresztül.

A Message táblába a következő mezőket vettük fel:

- Id - Az üzenet egyedi azonosítója
- ConversationId - Az üzenethez tartozó beszélgetés egyedi azonosítója
- Content - Az üzenet tartalma
- Date - Az üzenet elküldésének ideje
- UserId - Az üzenet küldőjének az egyedi azonosítója

2.6. Review tábla

A Review tábla 2 másik táblával van adatbázis kapcsolatban. Egy a többhöz (1:N) kapcsolatot ápol a **Report** tábla *ObjectId* mezőjén keresztül. Több az egyhez (N:1) kapcsolatot ápol a **User** táblával a *SellerId* és *BuyerId* mezőjén keresztül.

A Review táblába a következő mezőket vettük fel:

- Id - Az értékelés egyedi azonosítója
- SellerId - Az eladó egyedi azonosítója
- BuyerId - A vevő egyedi azonosítója
- Content - Az értékelésben meghatározott szöveg

- Rating - Az értékelésben meghatározott "pontszám", teljes szám formájában

2.7. Image tábla

Az Image tábla egyedül egy táblával van kapcsolatban. Több az egyhez (N:1) kapcsolatban áll a **Product** tábla *Id* mezőjével.

Az Image táblába a következő mezőket vettük fel:

- Id - A kép egyedi azonosítója
- ProductId - A termék egyedi azonosítója, amelyhez a kép tartozik
- ImageURI - A kép útvonala a webalkalmazás tárhelyében

2.8. Admin tábla

Az Admin táblának csak egy adatbázis kapcsolata van. Egy az egyhez (1:1) kapcsolatban áll a **User** táblával a *UserId* mezőn keresztül.

Az Admin táblába a következő mezőket vettük fel:

- Id - A rekord egyedi azonosítója
- UserId - Az admin felhasználó egyedi azonosítója
- Token - Amikor bejelentkezik a felhasználó az admin felületen akkor a szerver ezt a tokent generálja neki. Onnantól kezdve kizárólag akkor tud bármely kérést intézni a szerver felé, amennyiben ezt a token elküldi a kérésével együtt.

2.9. Report tábla

A Report táblához számos adatbázis kapcsolatot vettünk fel. Több az egyhez (N:1) kapcsolatban van a **Conversation**, **Review** és **Product** táblával az *ObjectId* mezőn keresztül. Ezekből minden rekordnál csak egy fog megvalósulni attól függően, hogy mi a *Type* mező értéke. Továbbá még a **User** táblával is több az egyhez (N:1) kapcsolatban van a *UserId* mezőn keresztül.

A Report táblába a következő mezőket vettük fel:

- *Id* - A jelentés egyedi azonosítója
- *Type* - A jelentés típusa ami azt jelzi hogy melyik táblához tartozik a jelentés. Ez lehet *product*, *conversation* vagy *review*.
- *ObjectId* - A típus által megjelölt elem egyedi azonosítója
- *UserId* - A felhasználónak, aki csinálta a jelentést, az egyedi azonosítója

3. Választott technológiák

- Laravel - az oldal a Laravel frameworkben készült
- Bootstrap - az oldalnak a frontendjéért felelős, illetve a mobil nézet megalkotásában is szerepet játszott
- C# - az asztali admin alkalmazás C# nyelvben íródott, WPF alkalmazás formájában
- Docker - az oldal Docker image-ek segítségével fut
- Intervention IO - ennek segítségével lehet képeket manipulálni
- Lightbox2 - a képgalériák megalkotásában játszott szerepet

4. Webes felület

4.1. Felhasználói felület

A weboldalon a felhasználó több felületet is elérhet, attól függően hogy be van-e jelentkezve vagy sem.

A főoldalt bárki elérheti, itt található meg az összes termék ami az adatbázisban megtalálható, "kártyák" formájában. Minden terméknek látható az ára, elnevezése, hozzácsatolt kép(ek), és ha az eladó meghatározza akkor a mérete és leírása is. A terméken minden esetben van egy "Contact seller" és "Buy" gomb, ezeknek a funkcionalitása a felhasználó vagy a termék állapotától függ. Lefagyasztott termék vagy be nem jelentkezett felhasználó esetén mindkét gomb ki van kapcsolva. Bejelentkezett felhasználók esetén ezek a gombok aktívak, és ha a felsorolt termék a bejelentkezett felhasználó terméke, akkor csak egy "Edit product" gomb jelenik meg. A Contact seller

gombbal a termék eladójával léphetünk kapcsolatba, a Buy gombbal a terméket vehetjük meg, míg az Edit product-tal a termék adatait szerkeszthetjük.

A profil oldalakon egy megadott felhasználó adatai, az általa felsorolt termékek, illetve a mások által írt értékelések olvashatóak. A felhasználó adatai az oldal tetején lévő keretben láthatóak, ahol lehetőségünk van látni a felhasználó arcképét (ha töltött fel profilképet), nevét, leírását, értékelését, illetve az általa feltöltött hirdetések számát. A felsorolt termékek ugyanolyan formában jelennek meg mint a főoldalon. Az értékeléseknél mindig látszódik a hogy ki írta az értékelést, hogy egy 1-5 skálán hány pontra értékelte az eladót, és ha bővebben ki akarja fejteni véleményét, akkor még egy egy leírás is olvasható. A felhasználó innen érheti el a személyes beállítások, illetve “dashboard” oldalakat.

A dashboard táblázatos formában sorolja fel a felhasználó listázott termékeit. Innen tudja szerkeszteni, lefagyasztani/kiolvasztani, illetve törölni azokat.

A beállításoknál (settings) megváltoztathatja egyes adatait a felhasználó (Profilkép, cím, jelszó stb...), illetve végleg törölheti a fiókját.

A messages fülben láthatóak a más felhasználókkal kezdeményezett beszélgetések, ahol minden beszélgetés egy adott termékhez van kötve.

A NAV-baron látható az oldal “logo”-ja, amire kattintva a landing page-re/feedre érkezik a felhasználó, egy searchbar, a termék feltöltésének lehetősége, illetve egy all-in-one menu, ahol a főbb funkciókat érheti el a

felhasználó. Ha nincs bejelentkezve a felhasználó, akkor itt található a belépés és regisztráció lehetősége is.

5. Admin alkalmazás

5.1. Felhasználói felület

A felhasználó felületre 2 darab ablakot terveztünk. Egy bejelentkezés és egy főoldalt.

A bejelentkezés ablakon a felhasználónak lehetősége van bejelentkezni, amennyiben jelentkezik admin joggal. Ha hibás jelszót, vagy email címet ad meg, vagy esetleg nem rendelkezik admin joggal, akkor nem engedi be a rendszer a főoldalra. Regisztrációra nincsen lehetőség.

A főoldal további 3 részre bontható. A felhasználókkal kapcsolatos oldal, a termékekkel kapcsolatos oldal és a kategóriákkal kapcsolatos oldal. A felhasználókkal kapcsolatos oldalon az adminnak számos műveletre van lehetősége.

Először is kétféleképpen végezhet keresést. Kereshet a felhasználók között a nevük, illetve az egyedi azonosítójuk alapján. Ezek közül a keresőmező melletti panelben tud választani. Amennyiben úgy dönt az adminisztrátor hogy név alapján szeretne keresni akkor egy listát fog látni a keresőmező alatt azokról a felhasználókról akiknek a nevében szerepel a keresett kifejezés. A felhasználóknak megjelenik az egyedi azonosítójuk, nevük, email címük, lakcímük, születési dátumuk és hogy rendelkeznek e admin joggal. Ezután valamelyik felhasználó rekordjának kattintásával tudja megjeleníteni a

felhasználó profilját Amennyiben a felhasználó egyedi azonosítója alapján keres, akkor rögtön meg fog jelenni a felhasználó profilja.

Amint megjelent a megtekinteni kívánt profil, újabb műveletek állnak rendelkezésre az adminisztrátor számára. Lehetősége lesz a profilkép törlésére, adminná promotálásra, vagy éppen adminról való lefokozásra normál felhasználóra, a felhasználó nevének, email címének, leírásának, lakcímének és születési dátumának módosítására, melyet az ezek alatt a mezők alatt található *“Update”* feliratú gombbal tud véglegesíteni. Közvetlenül ez alatt a gomb alatt van egy *“Delete”* feliratú gomb is amivel véglegesen lehet törölni a felhasználót. Ennek megnyomásakor egy felugró ablak megkérdezi, hogy biztosan szeretnénk e törölni a felhasználót.

Mindezek alatt számos lenyíló menüpontot is láthat a felhasználó. Ezek pedig nem mások mint a *“Products”*, *“Reviews that the user made”*, *“Reviews that the user received”*, *“Messages”*, *“Incoming reports”* és az *“Outgoing reports”* nevű menüpontok.

A *“Products”* fület lenyitva az adminisztrátornak rálátása nyílik az kiválasztott felhasználó által készített összes termékre ami még nem lett törölve. Egy hasonló lista jelenik meg mint mikor a felhasználóra keresett rá a neve alapján. Most viszont a termék egyedi azonosítója, neve, ára, eladás állapota és hogy jelentve van-e a termék mezők jelennek meg. Ezután egy termék rekordjának kattintásával jelenítheti meg az terméket.

Itt lehetőség van a termékek képeinek lapozására és törlésére, a termék nevének, leírásának, árának, méretének és kategóriájának módosítására. Ezt az ezek alatt az opciók alatt található *“Update”* feliratú gombbal teheti meg

az adminisztrátor. Amennyiben úgy kívánja, törölheti is a terméket az *“Update”* gomb alatt található *“Delete”* gombbal. Ennek megnyomásakor egy felugró ablak megkérdezi, hogy biztosan szeretnénk-e törölni a terméket.

Közvetlenül ezalatt található a lenyitható *“Reviews that the user made”* menüpont. Amint ez lenyitja az adminisztrátor, láthatja azokat az értékeléseket amiket a felhasználó írt másokról. Mindegy egyes értékelés egy kártya formájában jelenik meg a következő információkkal. Az értékelés egyedi azonosítója, az, hogy ki kiről írta, amennyiben írt véleményt akkor annak a szövege és maga az értékelés 1-től 5-ig.

Az értékelés törlésére a kártya alján egy *“Delete review”* elnevezésű gombbal van lehetőség. Ennek megnyomásakor megjelenik egy felugró ablak ami megkérdezi, hogy biztosan törölni akarja-e az értékelést.

A következő lenyitható menü szintén az értékelésekkel kapcsolatos viszont most azok az értékelések jelennek meg amiket mások írtak a kiválasztott felhasználóról. Ez a menüpont *“Reviews that the user received”* néven jelenik meg. A működése teljesen azonos a másik értékelésekkel kapcsolatos menüpontéval.

Alatta a *“Messages”* elnevezésű legördülő menüpont található mely az utolsó elem ezen a lapon. Ezt lenyitva az adminisztrátor egy listát fog látni a kiválasztott felhasználó beszélgetéseiről. Minden egye rekordban látni fogja, a beszélgetőpartner nevét, annak egyedi azonosítóját, hogy vevő-e vagy sem, illetve, hogy jelentette-valaki a beszélgetést, és ha igen akkor melyik fél.

Bármelyik beszélgetésre kattintva, megnyílik annak üzenetei közvetlenül alatta. Itt lehetősége van az adminisztrátoroknak ellenőrizni, hogy nem-e írt valaki valami nem helyénvalót. Amennyiben ez így történt törölheti a felhasználót.

A következő oldal termék oldal. Ez felépítésben nagyon hasonlít a felhasználói oldalon a termékek menüpontra. Annyiban különbözik ez, hogy itt a termék neve alapján lehet keresni az adatbázisban. A keresés gombra kattintás után megjelennek a termékek egy listában amelyek neve tartalmazza a keresett kifejezést. A termékekről a következő adatok jelennek meg. A termék neve, egyedi azonosítója, ára, az eladás állapota, illetve, hogy valaki jelentette-e a terméket. Bármely megjelent termék rekordjára kattintva, további információkat kapunk a termékről, ugyanúgy mint a felhasználói oldalon. Lehetőségünk van ezen adatok módosítására, illetve a termék törlésére is.

A 3. oldal a kategóriák oldal. Itt a termékkategóriákat lehet kezelni. Ez az oldal kettő legördülő menüpontra oszlik. Ha rákattintunk az első menüpontra ami a *“Categories”* elnevezést kapta, akkor egy legördülő listát láthatunk az összes kategóriáról. Bármelyiket kiválasztva, mellette megjelennek róla az adatok, illetve a vele végezhető adatok műveletei. Látható a kategória egyedi azonosítója, illetve a neve, melyet módosítani is lehet az *“Update”* feliratú gombbal. Alatt a *“Delete”* feliratú gombbal pedig a kategória végleges törlésére van lehetőség. Miután megnyomtuk a törlés gombot, utána egy feugró ablak megkérdezi, hogy biztosan szeretnénk-e törölni az adott kategóriát.

A kategória oldalon belül a második legördülő menü neve *“Create Category”*. Ezt lenyitva megjelenik egy szövegdoboz melybe az új kategória nevét kell beírni. A mellette lévő gomb megnyomásával létre is hozza az új kategóriát a beírt névvel.

És végül de nem legutolsó sorban az utolsó oldal az admin felületen a jelentésekről szól. Itt több féle szűrő közül is lehet választani. Az adminisztrátor kiválaszthatja, hogy mely típusú(*“product”*, *“conversation”*, *“review”*) jelentéseket szeretne látni. Továbbá azt is, hogy a felhasználó neve, vagy egyedi azonosítója alapján keressen. És ha úgy kívánja akkor a felhasználó neve vagy azonosítója nélkül és végrehajthatja a keresést, ha nem ír be semmit a szövegdobozba.

A keresés gomb megnyomása után egy lista fog megjelenni. Minden egyes rekordban látható lesz a jelentő neve, egyedi azonosítója, a jelentett neve, annak egyedi azonosítója, a rendelés típusa(*“product”*, *“conversation”*, *“review”*) és annak egyedi azonosítója, továbbá még egy gomb is, amivel törölni lehet a jelentést.

5.2. Biztonság

Az adminisztrációs rendszer mind a kettő végén ügyelnünk kellett, hogy csakis az férhessen hozzá a műveletekhez, akiknek joga van hozzá, tehát az adminoknak.

A back-enden ezt úgy valósítottuk meg, hogy létrehoztunk egy új táblát aminek egy rekordja eltárolta az felhasználó egyedi azonosítóját és számára egy token. Ezt a token minden egyes bejelentkezésnél újra generálja

teljesen random módon. Bejelentkezés után a rendszer elküldi ezt a tokent a frontendre. Ez így még önmagában semmiféle védelmet nem nyújt. Ezért hoztunk létre egy egyedi middlewaret ami minden egyes, az admin felületről érkezett kérésnél (kivéve a login kérést) megvizsgálja, hogy a kérés tartalmazza-e az adminisztrátor email címét és a generált tokent amit bejelentkezésnél kapott. Amennyiben ez nem valósul meg, 403-as hibakóddal térünk vissza, különben a szerver teljesíti a kérést.

A front-end oldalon még továbbra is szükség van némi biztonságra. Ezért ott létrehoztuk a bejelentkezés oldalt. Miután az adminisztrátor sikeresen bejelentkezett, kap egy tokent amit az alkalmazás eltárol további használatra. Ezután átirányítja az admint a főoldalra ahol bármiféle műveletet végrehajthat.

6. Back-end

6.1. Migrációk és modellek

6.2. Authentikáció

6.3. Routeok

6.3.1. Web routeok

A webes routeok a weboldalak megjelenéséért, illetve számos feltöltési feladat elvégzéséért felelősek.

- GET /register - betölti a regisztrációs oldalt
- POST /register - amennyiben megfelelőek az adatok, regisztrálja a felhasználót
- POST /login - amennyiben megfelelőek az adatok, bejelentkezteti a felhasználót
- GET /logout - tartalmazza az 'auth' middlewaret és kijelentkezteti a felhasználót
- GET /profile/id - betölti az id alapján meghatározott felhasználó profilját

(innentől az összes webes route tartalmazni fogja az 'auth' middlewaret)

- GET /profile/id/upload - betölti a termék feltöltésére szolgáló oldalt
- GET /profile/id/dashboard - betölti a kezelőfelület oldalát
- POST /products/id/edit - amennyiben megfelelnek az adatok és a felhasználónak van rá jogosultsága, frissíti az id alapján meghatározott termék adatait
- POST /products/upload - amennyiben megfelelnek az adatok, feltölt egy új terméket az adatbázisba
- GET /messages - betölti az üzenetek oldalt
- GET /settings - betölti a beállítások oldalt

6.3.2. Api routeok

Az api routeok adatok lekérdezéséért, módosításáért, feltöltéséért és törléséért lesz felelős. Ezeket az útvonalakat további kettő pontra bontanám. A felhasználók útvonalaira, amiket a sima felhasználók

érhetnek el és admin útvonalakra amiket kizárólag az admin alkalmazásról lehet elérni.

6.3.2.1. Sima felhasználói routeok

A következő routeok néhány kivétellel mind az 'auth:sanctum' middlewarrel vannak ellátva. Ahol ez nem így van, azt jelzem a route leírásában.

- GET /conversation/buys - az middleware által meghatározott felhasználó azon beszélgetéseinek adataival tér vissza, melyekben a vevő oldalon van.
- GET /conversation/buys - az middleware által meghatározott felhasználó azon beszélgetéseinek adataival tér vissza, melyekben az eladó oldalon van.
- GET /conversation/id - az id alapján meghatározott beszélgetés üzeneteivel tér vissza
- POST /conversation - amennyiben megfelelnek az adatok, létrehoz egy új beszélgetést az adatbázisban
- POST /message - amennyiben megfelelnek az adatok, létrehoz egy új üzenetet az adatbázisban, a megfelelő beszélgetéshez
- PUT /user - amennyiben megfelelnek az adatok, frissíti a felhasználó profilját
- POST /user/password - amennyiben megfelelnek az adatok, a felhasználó módosíthatja a jelszavát
- DELETE /user - a felhasználó törölheti a fiókját
- GET /products - ez a route nincs ellátva az 'auth:sanctum' middlewarrel. Visszatér az összes termék adataival

- PATCH /products/buy - egy eladott termék “sold” mezőjét frissíti igazra
- PATCH /products/freeze - egy terméknek az “iced” mezőjét állítja a jelenlegi logikai értékének ellenkezőjére
- GET /products/id - ez a route nincs ellátva az ‘auth:sanctum’ middlewarerel. Visszatér az id által meghatározott termék adataival
- DELETE /products - amennyiben az adatok megfelelnek és a felhasználónak jogosultsága van rá, törli az adott terméket
- GET /categories – ez a route nincs ellátva az ‘auth:sanctum’ middlewarrel. Visszatér az összes kategória adataival
- POST /products/id/report - amennyiben az adatok megfelelnek, létrehoz egy új jelentést az id alapján meghatározott termékről
- POST /conversation/id/report - amennyiben az adatok megfelelnek, létrehoz egy új jelentést az id alapján meghatározott beszélgetésről
- POST /reviews/id/report - amennyiben az adatok megfelelnek, létrehoz egy új jelentést az id alapján meghatározott véleményről
- POST /reviews - létrehoz egy új véleményt, amiben megvan határozva hogy ki a vevő és ki az eladó
- POST /reviews/id/update - frissít egy előzőleg létrehozott véleményt, ezzel kap a leírás és az értékelés pontszáma értéket, hogy az értékelés publikálásra kerülhessen

6.3.2.2. Admin routeok

Az admin routeok a bejelentkezés kivételével mind el vannak látva az 'AdminAuth' middlewarrel.

- POST /admin/login - amennyiben megfelel az email cím és a jelszó, bejelentkezteti az admint
- GET /admin/users/id - az id alapján meghatározott felhasználó adatait küldi vissza
- PUT /admin/users/id - ha megfelelnek az adatok, frissíti az id alapján meghatározott felhasználó profilját
- GET /admin/users - a név paraméter alapján, visszatér az összes olyan felhasználó adataival, melyek tartalmazzák a keresett kifejezést
- DELETE /admin/users/id - az id alapján meghatározott felhasználót végleg törli az adatbázisból
- POST /admin/users/id/promote - az id alapján meghatározott felhasználónak admin jogot ad
- POST /admin/users/id/demote - az id alapján meghatározott felhasználótól megvonja az admin jogot
- GET /admin/users/id/products - az id alapján meghatározott felhasználó termékeinek visszatér az összes adatával
- PUT /admin/products/id - az id alapján meghatározott termék adatait frissíti, amennyiben megfelelnek az adatok
- GET /admin/products - a paraméterben meghatározott név alapján visszatér az összes termék adataival mely tartalmazza a keresett kifejezést
- DELETE /admin/products/id - az id alapján meghatározott terméket végleg törli az adatbázisból

- GET /admin/categories - visszatér az összes kategória adataival
- PUT /admin/categories/id - az id alapján meghatározott kategóriát frissíti, amennyiben megfelelnek az adatok
- POST /admin/categories - amennyiben megfelelnek az adatok, feltölt egy új kategóriát az adatbázisba
- DELETE /admin/categories/id - az id alapján meghatározott kategóriát törli az adatbázisból
- GET /admin/users/id/reviews/made - visszatér az összes olyan értékeléssel amit az id által meghatározott felhasználó készített másokról
- GET /admin/users/id/reviews/received - visszatér az összes olyan értékeléssel amit az id által meghatározott felhasználóról készítettek mások
- DELETE /admin/reviews/id - törli az adatbázisból az id alapján meghatározott értékelést