

Záródolgozat

készítette:
Ladmaczi Linda

2023

Tartalomjegyzék

Bevezetés.....	3
Témaválasztás indoklása.....	3
Fejlesztői Dokumentáció.....	3
Fejlesztői környezet.....	4
Weboldal telepítése és elindítása.....	5
Adatbázis szerkezet felépítése.....	6
User_categories tábla.....	6
Users tábla.....	6
Users_adress tábla.....	7
Orders tábla.....	7
Categories tábla.....	8
Products tábla.....	8
Order_products tábla.....	8
Főbb kódrészek.....	9
Termékek kosárba mentése, megadott mennyiség alapján.....	9
Kosár tartalmának lekérése és ellenőrzése.....	10
Kosár tartalmának törlése.....	10
Felhasználói dokumentáció.....	11
Regisztráció.....	11
Bejelentkezés.....	11
Termék rendelésének menete.....	12
Checkout oldal.....	13
Felhasználói fiók.....	14
Admin oldal.....	15
Tesztelési dokumentáció.....	16
Fejlesztési lehetőségek.....	16
Irodalomjegyzék.....	17

Bevezetés

Témaválasztás indoklása

Ebben a dokumentációban áttekintést fogsz találni a weboldalamról, amely szépségápolási termékeket kínál. Célom, hogy a látogatók számára kellemes és informatív élményt nyújtsak, valamint segítsek nekik megtalálni azokat a minőségi termékeket, amelyek könnyebbé teszik a mindennapi rutinjaikat.

Miért éppen a szépségápolási termékek világában merültem el? Ennek egyszerű oka van. Hiszek abban, hogy az egészséges, ápolt bőr a szépség és magabiztosság alapja. A szépségápolási termékek iránti kereslet folyamatosan nő, és egyre többen keresik az online vásárlás lehetőségeit. Szükségessé vált tehát egy olyan oldal, ahol az emberek könnyen hozzáférhetnek és rátalálhatnak a számukra legmegfelelőbb termékekre.

Nagyon sok termék sajnos nem elérhető Magyarországon, ezért mindenképpen egy olyan oldalt szerettem volna készíteni, ami elérhetővé teszi számukra ezeket a termékeket. A témaválasztásom indoklása tehát az, hogy segítséget és inspirációt nyújtsak az embereknek egyszerűen az otthon kényelméből.

Fejlesztői Dokumentáció

Fejlesztői környezet

Az weboldalam JavaScript és PHP nyelven íródott, a Visual Studio Code (VS Code) fejlesztői környezet segítségével.

- Visual Studio Code: Fejlesztői környezet, amelyben a weboldal teljes kódját írtam és szerkesztettem. Ez egy ingyenes nyílt forráskódú kódszerkesztő, rengeteg beépített, felhasználóbarát támogatással.
- JavaScript: A kliensoldali logika megvalósításához, például az interaktivitásért és az űrlapkezelésért felelős.
- HTML/CSS: Az alapvető webes szerkezet és megjelenítés leírásához használtam.
- PHP: A szerveroldali logika megvalósításához, például az adatbáziskezelésért és a dinamikus tartalomért felelős.
- XAMPP v3.3.0: Ingyenes, nyílt forráskódú szoftvercsomag, ami könnyen kialakítható szerverkörnyezetet kínál. Illetve a XAMPP tartalmazza a PHP szerverkomponenst, amely lehetővé tette az oldalam futtatását.

Weboldal telepítése és elindítása

1. Visual Studio Code telepítése és futtatása
2. XAMPP telepítése és futtatása
3. Apache és MySQL futtatása
4. Kód bemásolása a XAMPP/htdocs mappájába
5. Kódot tartalmazó mappa átnevezése **vizsga_project** -re
6. A kód megnyitása Visual Studio Code-ban
7. <http://localhost/phpmyadmin/> megnyitása
8. Új tábla létrehozása **vizsgaproject_ladmaczilinda** néven
9. SQL fájl importálása az újonnan létrehozott táblába
10. http://localhost/vizsga_project/ elérési út megadása a böngészőben

Tesztelési felhasználó bejelentkezési adatai:

felhasználónév: test

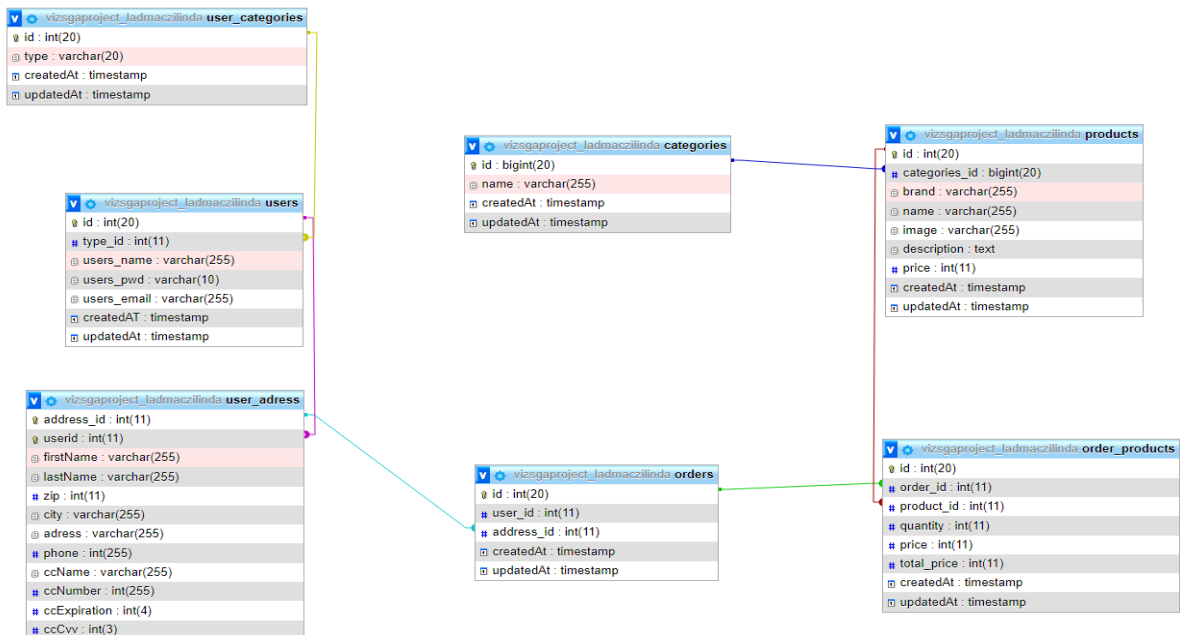
jelszó: test

Tesztelési admin bejelentkezési adatai:

felhasználónév: admin

jelszó: admin

Adatbázis szerkezet felépítése



User_categories tábla

vizsgaproject_ladmaczilinda user_categories
id : int(20)
type : varchar(20)
createdAt : timestamp
updatedAt : timestamp

Ebben a táblában a felhasználók típusait tárolom el. Jelenleg két fajta felhasználó létezik. Admin és sima felhasználó. A tábla “type” mezőjében tárolok el 0-ás és 1-es értéket. Csak az admin férhet hozzá az admin felülethez, amin lehetősége van törölni a beérkező megrendeléseket.

Users tábla

vizsgaproject_ladmaczilinda users
id : int(20)
type_id : int(11)
users_name : varchar(255)
users_pwd : varchar(10)
users_email : varchar(255)
createdAt : timestamp
updatedAt : timestamp

A users tábla fő feladata, hogy eltárolja a regisztráció során megadott információkat. Az első mező az id, amely szám alapú adatokat tárol. Valahányszor új felhasználó jön létre, automatikusan generálódik hozzá egy id, ami alapján

egyszerűen szűrhető a felhasználó. A type_id mező szintén szám alapú adat. Itt határozzuk meg, hogy a regisztrált felhasználónk admin-e vagy sem. Mindenki automatikus 0 type_id értékkel indít. A rendszergazda adhat admin jogosultságot, azzal, hogy 1-es értéket ad meg. A users_name, users_pwd és users_email mezők szöveg alapú adatokat tárolnak, melynek feladatuk a felhasználók által megadott adatok tárolása. Ide kerül be a felhasználónév, email cím és jelszó.

Users_adress tábla

vizsgaproject_ladmaczilinda user_adress	
🔑	address_id : int(11)
🔑	userid : int(11)
📄	firstName : varchar(255)
📄	lastName : varchar(255)
#	zip : int(11)
📄	city : varchar(255)
📄	adress : varchar(255)
#	phone : int(255)
📄	ccName : varchar(255)
#	ccNumber : int(255)
#	ccExpiration : int(4)
#	ccCvv : int(3)

A tábla célja a regisztrált felhasználók rendelési címének eltárolása. A felhasználó megadhatja a vezetéknévét, keresztnévét, az irányítószámot, várost, pontos címet, telefonszámot és a kártya adatait. Az address_id egy automatikusan generált szám típusú adatot tárol el, melyet utána átadok és összekötök az orders táblával. A userid mező pedig a users táblából áthozott id-t kapja meg.

Orders tábla

vizsgaproject_ladmaczilinda orders	
🔑	id : int(20)
#	user_id : int(11)
#	address_id : int(11)
📄	createdAt : timestamp
📄	updatedAt : timestamp

Az orders tábla célja a felhasználók és a megrendelt termékek összekapcsolása. Eltárolja felhasználói id-t és a rendelési címének id-át.

Categories tábla

vizsgaproject_ladmaczilinda categories	
id	bigint(20)
name	varchar(255)
createdAt	timestamp
updatedAt	timestamp

A categories tábla feladata a weboldalon található termékek kategorizálása. Minden kategória saját id-val és névvel rendelkezik. Az itt létrehozott id-t fogom a későbbiekben felhasználni arra, hogy leszűrjem a termékeket.

Products tábla

vizsgaproject_ladmaczilinda products	
id	int(20)
categories_id	bigint(20)
brand	varchar(255)
name	varchar(255)
image	varchar(255)
description	text
price	int(11)
createdAt	timestamp
updatedAt	timestamp

A products tábla rendelkezik saját id-val, amit a későbbiekben az order_products tábla kap meg. A categories_id mező tárolja el a categories tábla egyedi azonosítóját. A tábla többi részében eltárolom a márkát, a termék megnevezését, egy képet, részletes leírást a termékről és az aktuális árát.

Order_products tábla

vizsgaproject_ladmaczilinda order_products	
id	int(20)
order_id	int(11)
product_id	int(11)
quantity	int(11)
price	int(11)
total_price	int(11)
createdAt	timestamp
updatedAt	timestamp

Az order_products tábla eltárolja a rendelés azonosítóját, a megrendelt termék azonosítóját, a termék mennyiségét, a megrendelt termék árát és az összesített árát, ha többet rendelt.

Főbb kódrészletek

Szeretném kiragadni a SESSION-t, mint főbb részét a weboldalamnak. A SESSION több fő funkciót is ellát a weboldalon

Termékek kosárba mentése, megadott mennyiség alapján

```
public function setCartSession($product, $qty)
{
    $isExistProduct = false;

    if (!empty($_SESSION["cart"])) {
        foreach ($_SESSION['cart'] as $product_value) {
            if ($product_value['id'] == $product[0]['id']) {
                $isExistProduct = true;
            }
        }
    }

    if ($isExistProduct == true) {
        $this->updateProductQuantitySession($product[0]['id'], $qty);
    } else {
        $price = $product[0]['price'] * $qty;
        $_SESSION['cart'][] = $this->getProductStructure($product, $qty, $price);
    }
}
```

Létrehozok egy `setCartSession` metódust. Ez a metódus egy termék és egy mennyiség paramétert vár és hozzáadja a terméket a kosárhoz a ‘\$_SESSION’ változóban tárolt adatok alapján. Az if feltételben először is ellenőrzöm, hogy a kosaram üres-e, azért, hogy ne fusson hibába, ha a kosár nem tartalmaz semmilyen elemet. Aztán a foreach ciklus végigiterál a \$_SESSION[“cart”] tömbön és ellenőrzi, hogy a hozzáadni kívánt termék szerepel-e már a kosárban úgy, hogy összehasonlítja a termék id-ját a már létező termékkel. Ha a termék már megtalálható a kosárban, akkor meghívódik az `updateProductQuantitySession` metódus és csak frissíti a már meglévő termék mennyiségét.. Ha nem található termék a kosárban, akkor kiszámolom a termék árát, majd meghívom a `getProductStructure` metódust, a termék adatainak előkészítéséhez, majd hozzá adom a terméket a kosárhoz.

Kosár tartalmának lekérése és ellenőrzése

```
public function getCartSession()
{
    if( empty($_SESSION['cart']))
    {
        echo 'There are no items in your cart';
    }

    if(!isset($_SESSION['cart']) && empty($_SESSION['cart']))
    {
        $_SESSION['cart'] = [];
    }
    return $_SESSION['cart'];
}
```

Egy if feltétellel ellenőrzöm, hogy a `$_SESSION['cart']` változó üres-e. Ha nem tartalmaz semmit, akkor jelzem, hogy a kosár üres. Egy másik if feltétellel ellenőrzöm, hogy létezik-e és üres-e. Ha a változó nem létezik vagy üres, akkor biztosítva van, hogy üres tömbként tér vissza. Végül vissza adja a `$_SESSION['cart']` változót, hogy a későbbiekben felhasználhassam.

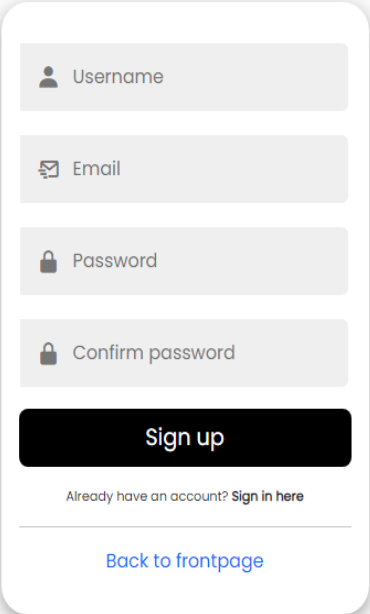
Kosár tartalmának törlése

```
public function removeProductSession($productId)
{
    foreach($_SESSION['cart'] as $key => $product_value){
        if($product_value['id'] == $productId) {
            unset($_SESSION['cart'][$key]);
        }
    }
    return $_SESSION['cart'];
}
```

Létrehoztam a `removeProductSession` funkciót, amely azért felel, hogy a kosárból termékeket törölhessek. Egy foreach ciklusban körbejárom a kosár tartalmát és minden egyes terméket leellenőrzök, hogy a `$product_value["id"]` megegyezik-e a bemeneti paraméterrel(`$productId`). Ha megtalálom a megfelelő terméket, akkor az `unset` függvény segítségével eltávolítom a terméket a kosárból. Végül visszatérési értéként megadjuk a `'$_SESSION["cart"]` tömböt, hogy a frissített állapotban elérhető legyen a kosár.

Felhasználói dokumentáció

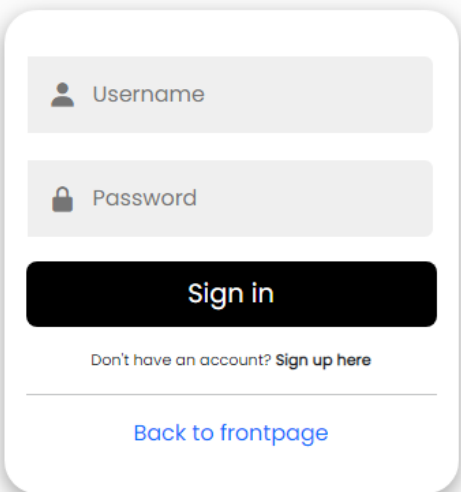
Regisztráció



A registration form with a white background and rounded corners. It contains four input fields: 'Username' with a person icon, 'Email' with an envelope icon, 'Password' with a lock icon, and 'Confirm password' with a lock icon. Below these fields is a black 'Sign up' button. Under the button, there is a link 'Already have an account? Sign in here'. At the bottom, there is a blue link 'Back to frontpage'.

A regisztráció során a felhasználónak meg kell adnia a felhasználónevét, email címét, jelszavát és a jelszó ellenőrző mezőt is ki kell töltenie. Sikeres regisztráció után, az oldal átnavigál a bejelentkezés oldalra és felugró ablakkal jelzi a sikeres regisztrációt. Ha már van felhasználói fiókunk, akkor a “Sign in here” hivatkozással vissza térhetünk a bejelentkezés fülhöz vagy a “Back to frontpage” hivatkozással újra megnyithatjuk a főoldalt.

Bejelentkezés

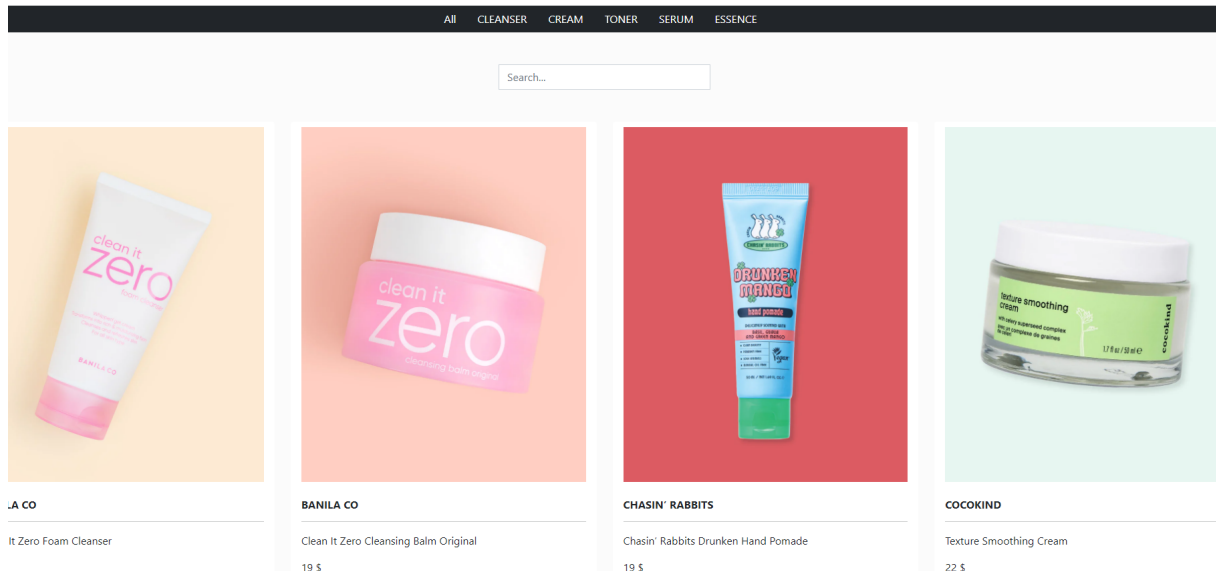


A login form with a white background and rounded corners. It contains two input fields: 'Username' with a person icon and 'Password' with a lock icon. Below these fields is a black 'Sign in' button. Under the button, there is a link 'Don't have an account? Sign up here'. At the bottom, there is a blue link 'Back to frontpage'.

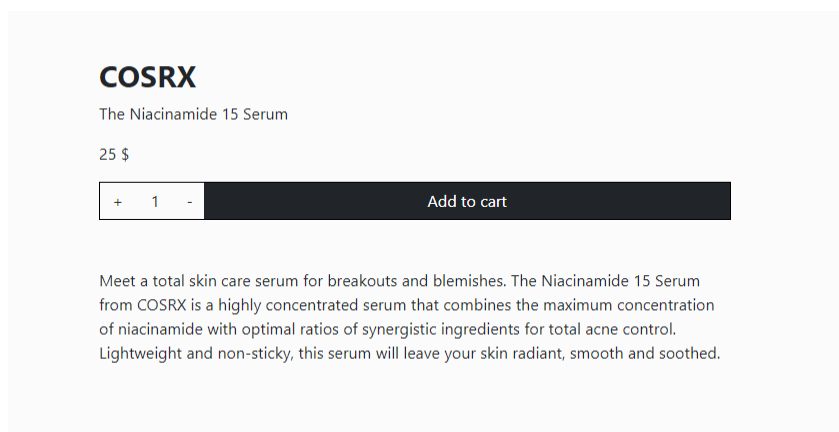
A bejelentkezéshez meg kell adnia a felhasználónak a regisztráció során megadott felhasználónevét és jelszavát. Ha nincs felhasználói fiókja a “Sign up here” hivatkozással átmehet a regisztrációs fülre, ha pedig a főoldalra szeretne visszamenni, akkor a “Back to frontpage” hivatkozással ezt is megteheti.

Termék rendelésének menete

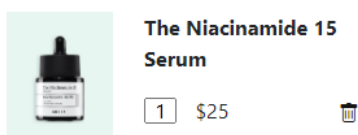
Termék kereső



A navigációs sávon a “products” fülre kattintva megjelenik az oldalon rendelhető termékek listája. A termékeket kategória szerint le lehet szűrni, illetve a kereső mezőbe is meg lehet adni a keresett termék nevét vagy bármilyen tetszőleges szót, amire szűrni szeretne. A kiválasztott termék képére kattintva kiválaszthatja a terméket és átnavigálhat az oldalra, ahol bővebb információt kaphat a termékről.



A felhasználónak lehetősége van megadni a kívánt mennyiséget, ezután az “Add to cart” gombra kattintva a kosárba helyezni.



Az “Add to cart” gombra kattintás után felugrik a jobb oldalon a kosár, ami tartalmazza a kosárba helyezett

termékeket, képekkel, mennyiséggel és árral ellátva. A termékeket lehetőség van tételesen törölni.

Subtotal: \$25

CHECKOUT

A kosár alján a Subtotal jelzi a kosárba helyezett termékek végösszegét. A “Checkout” gombra kattintva mehet át a rendelés leadása oldalra.

Checkout oldal

Please log in to your account to finish the checkout! [Log in.](#)

Shipping address

First name

Last name

Zip code

City

1234 Main St

Phone

☐ Agree to Term & Conditions

A rendelés leadásához szükséges a felhasználói fiók létrehozása, a bejelentkezés és a felhasználói fiókon belüli rendelési cím megadása. Amíg az adatok nincsenek megadva, a rendelés leadás inaktív.

Ha a felhasználó megadta az adatait, akkor azok a Checkout oldalon automatikusan kitöltődnek, és a kártya adatok megadása után, leadhatja a rendelését a “Continue to checkout” gombra kattintva.

Felhasználói fiók

Profile Settings

First name:

Last name:

Phone Number:

Zip:

City:

Address:

[Back](#) [Save Profile](#)

Bejelentkezés után a főoldalon megjelenítésre kerül a felhasználói név. A névre kattintva átnavigálhatunk a felhasználói fiókra, ahol a felhasználó megadhatja a rendelési címét. Az adatok helyes kitöltése után a “Save Profile” gombra kattintva elmentheti az adatait. A “Log out” gombra kattintva a felhasználó ki jelentkezhet az oldalról.

Profile Settings

First name:

Last name:

Phone Number:

Zip:

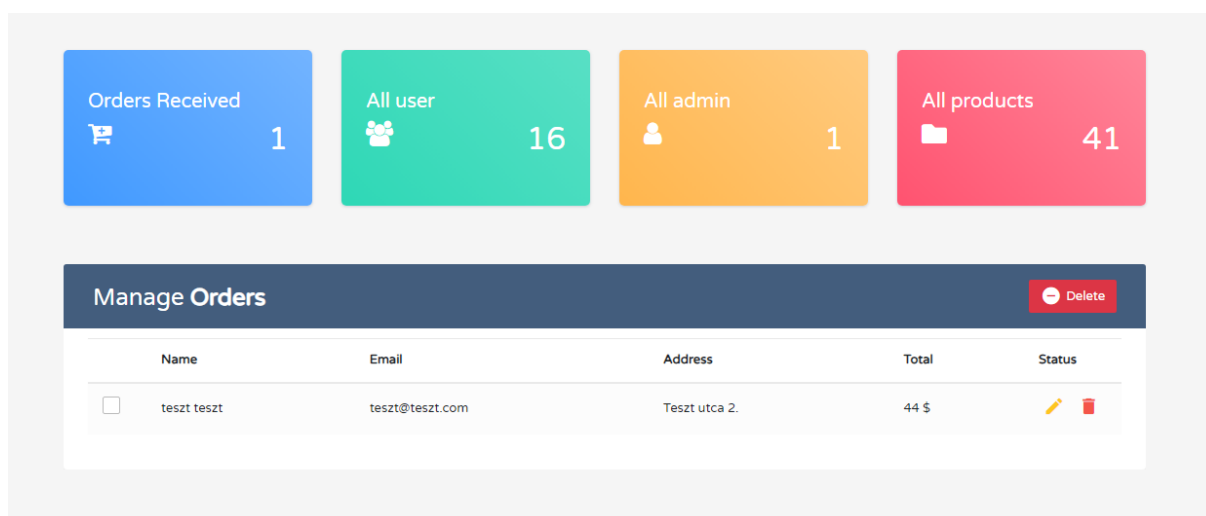
City:

Address:

[Back](#) [Update Profile](#)

Ha a felhasználó megadta az adatait de a későbbiek során változtatni szeretne rajta, akkor azt is megteheti az adatok javításával és az “Update Profile” gombbal.

Admin oldal



Az admin jogosultsággal ellátott felhasználók bejelentkezés után, a felhasználó nevükre kattintva a főoldalon átnavigálhatnak az admin oldalra. Az oldalon kilistázva megtekintheti a az összes leadott rendelés számát, a regisztrált felhasználók számát, az adminok számát és a weboldalon található összesített termékek számát. A táblázatban kilistázva megjelenik az összes leadott rendelés és tetszőlegesen törölheti azokat. Az oldal jobb sarkában lévő “Log out” gombra kattintva az admin ki jelentkezhet és visszakerül a főoldalra.

Tesztelési dokumentáció

Az oldalt teszteltem Google Chrome, Microsoft Edge és Opera böngészőben is. Eltérést nem találtam.

Fontosnak tartottam, hogy egyszerű felhasználók milyen életszerű problémákat generálnak, amik az én figyelmemet esetleg elkerülhették. 4 ismerősömet kértem meg, hogy teszteljék az oldal működését. Végig mentek a regisztráció, bejelentkezés, termék keresés, megrendelés és felhasználói profil szerkesztése pontokon. Többször generáltak olyan felmerülő problémát, amit sikeresen javítani tudtam.

- regisztráció után a főoldalra vitt vissza az oldal, nem pedig a bejelentkezéshez - javítva
- nem volt egyértelmű, hogy a rendelés leadásához a profilon belül meg kell adni az adatokat - javítva
- kevés volt az input feedback - javítva

Emellett figyeltem hogyan navigálnak el az oldalon és végeztem fejlesztéseket, hogy áttekinthető és igazán felhasználó barát legyen.

- több/jobban elhelyezett gombok
- kattintható képek
- színebb/interaktívabb oldalak

Fejlesztési lehetőségek

- A főoldalon a legújabb termékek listázása
- Contact oldal létrehozása a főoldalon, ahol az elérhetőség megadása után üzenetet küldhetünk az oldal tulajdonosának
- A termékek oldalán több kép megjelenítése és részletes összetevőinek listázása
- Promo code létrehozása, eltárolása és rendeléskor annak felhasználása
- Az admin oldalon a megrendelések csoportos törlése és szerkesztése

Irodalomjegyzék

- W3Schools - <https://www.w3schools.com/>
- Stack Overflow - <https://stackoverflow.com/>
- Freepik - <https://www.freepik.com/>
- Bootstrap - <https://getbootstrap.com/docs/5.0/>
- Sokoglam - <https://sokoglam.com/>