1. Bevezetés
   1. A vizsgaremek témája

A Danger Shop egy online cipőbolt, ahol a divat és kényelem tökéletes egyensúlyát találhatod meg. Webshopunkban széles választékban kínálunk stílusos és minőségi cipőket minden alkalomra és minden ízlésnek. Legyen szó elegáns alkalmi cipőkről, kényelmes sportcipőkről vagy divatos szandálokról, nálunk mindent megtalálsz.

* 1. A program funkciói

A weboldalunk számos praktikus funkcióval rendelkezik, hogy a felhasználók számára kényelmes és zökkenőmentes élményt nyújtson.

1. Regisztráció és Bejelentkezés:

- Regisztráció: Új felhasználók könnyedén létrehozhatnak egy fiókot, amely lehetővé teszi számukra, hogy mentse az adatokat.

- Bejelentkezés: A már regisztrált felhasználók egyszerűen bejelentkezhetnek a saját fiókjukba, így gyorsabban és könnyebben vásárolhatnak.

2. Termékek részletes bemutatása:

- Több Kép: Minden cipőhöz több részletes kép tartozik, amelyek lehetővé teszik a vásárlók számára, hogy biztosak lehessenek a választásukban.

- Több Cipőméret: A termékek oldalán a különböző méretek kiválaszthatók, így mindenki megtalálhatja a megfelelő méretet.

3. Kosár és Fizetés:

- Kosárhoz Adás: A vásárlók könnyedén hozzáadhatják a kiválasztott cipőket a kosárhoz, majd tovább böngészhetnek vagy folytathatják a vásárlást.

Ezen funkciók együttesen segítik a felhasználókat abban, hogy könnyedén böngésszenek a termékek között, kiválasszák a megfelelő cipőt, és zökkenőmentesen véglegesítsék a vásárlást.

* 1. Célközönség

A Danger Shop sokszínű kínálata lehetővé teszi, hogy széles rétegű vásárlók számára vonzó legyen. Megtalálhatóak női és férfi cipők

Férfiak:

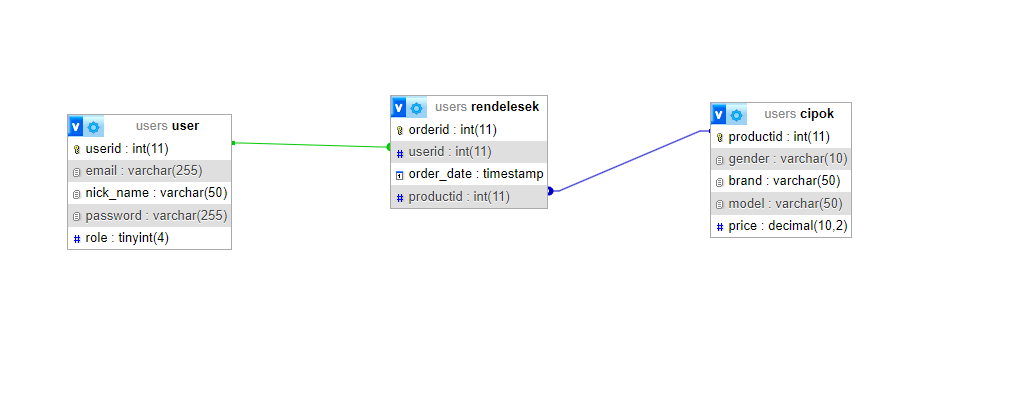
A férfiak számára kínált cipőkollekciónkban elegáns üzleti cipőktől kezdve, kényelmes sportcipőkig és divatos szandálokig minden megtalálható. Azoknak, akik értékelik a stílust és minőséget, és szeretnék kifejezni egyéni stílusukat a lábbelijükkel, számos lehetőség áll rendelkezésre.

Nők:

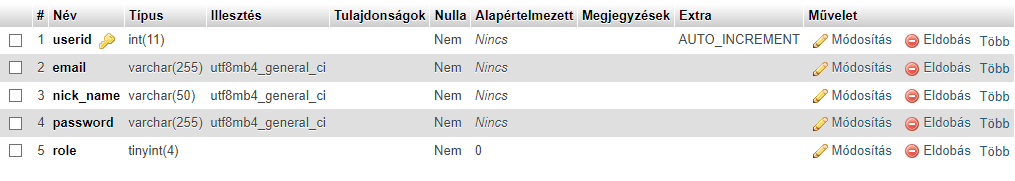
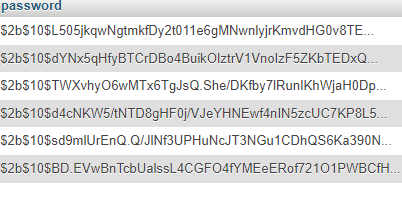
Női vásárlóink számára is széles választékban kínálunk cipőket. Legyen szó magas sarkú cipőkről elegáns alkalmi eseményekre, kényelmes balerinákról a mindennapokhoz vagy sportcipőkről az aktív életmódhoz, nálunk mindenki megtalálhatja a stílusának és kényelmi igényeinek megfelelő cipőt.

Célunk, hogy minden korosztály számára kínáljunk megfelelő cipőket, legyen szó formáról, stílusról vagy méretről. A Danger Shop abban hisz, hogy minőségi lábbeli mindenki számára elérhetővé kell, és kell is hogy legyen, és ezért folyamatosan frissítjük kínálatunkat, hogy minden vásárlónk elégedett legyen a választásával.

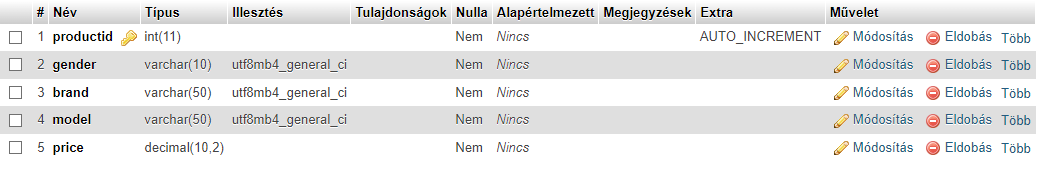
1. Fejlesztői dokumentáció
   1. Adatbázis



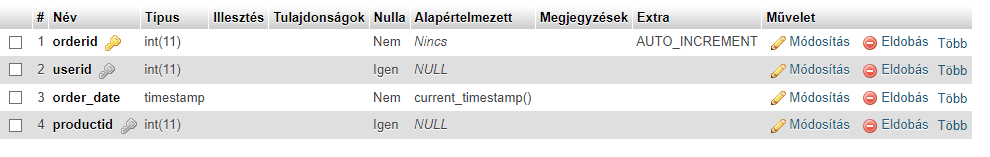
* + 1. Az adatbázisok nélkülözhetetlen eszközei a modern webalkalmazásoknak, mivel segítik az adatok hatékony tárolását, kezelését és lekérdezését. A webshopunk számára kialakított adatbázis oldalunk tervezésekor a MySQL-t választottuk, mivel az iskolában ezt tanultuk.
    2. A user tábla:  
       userid (Elsődleges int(11) AUTO\_INCREMENT): Egyedi azonosító, amely automatikusan növekszik minden új felhasználó hozzáadásakor. Az elsődleges kulcsként szolgál a táblában.  
       email (varchar(255) utf8mb4\_general\_ci): Felhasználó e-mail címe, maximális hossza 255 karakter. Az "utf8mb4\_general\_ci" karakterkészletet választottuk, ami támogatja a nemzetközi karaktereket is.  
       nick\_name (varchar(50) utf8mb4\_general\_ci): Felhasználó beceneve vagy felhasználónév, maximális hossza 50 karakter.  
       password (varchar(255) utf8mb4\_general\_ci): A felhasználó jelszavát tárolja. A "utf8mb4\_general\_ci" karakterkészletet választottuk a karakterek széles skálájának támogatásához. A felhasználók jelszavainak biztonságos tárolása kiemelt fontosságú, ehhez a bcrypt Node.js csomagot használtuk, amely hatékony jelszóhashelést és számos biztonsági intézkedést kínál.  
       role (tinyint(4)): Jogosultsági szint, amely kijelöli a felhasználó jogosultságait. A "tinyint(4)" típus kis egész számot jelöl, amely ebben az esetben a felhasználó szerepkörét tárolja.



* + 1. A cipok tábla:  
       A "cipok" tábla egy kulcsfontosságú része a webshopunk adatbázisának, amelyben rögzítjük a különböző cipőmodelleket és azok részleteit.  
       productid (Elsődleges int(11) AUTO\_INCREMENT): Egyedi azonosító, amely minden új termék hozzáadásakor automatikusan növekszik. Az elsődleges kulcsként szolgál a táblában.  
       gender (varchar(10) utf8mb4\_general\_ci): A cipő nemét tárolja, például "male(férfi)" vagy "female(női)". A "utf8mb4\_general\_ci" karakterkészletet választottuk a nemzetközi karakterek támogatásához.  
       brand (varchar(50) utf8mb4\_general\_ci): A cipő márkáját tárolja, például "Nike", "Adidas", "Puma", stb.  
       model (varchar(50) utf8mb4\_general\_ci): A cipő modelljét tárolja, például "Air Force 1", "Superstar", "Ultraboost", stb.  
       price (DECIMAL(10, 2)): A cipő árát tárolja. A DECIMAL típus lehetővé teszi a tizedes pontosságú árak kezelését.  
       A "cipok" tábla kezelése során fontos figyelembe venni a termékek egyediségét és azoknak az árukészlettel kapcsolatos változásait. Az árakat és egyéb részleteket rendszeresen frissíteni kell az aktuális állapot tükrözése érdekében. Ezen információk birtokában a webshop könnyen és hatékonyan kezelheti a cipőkkel kapcsolatos adatokat az adatbázisban, ami hozzájárulhat a felhasználókkal való sikeres és elégedett interakcióhoz.



* + 1. A rendelesek tábla:  
       A "rendelesek" táblában rögzítjük a felhasználók által leadott rendeléseket és azok részleteit.  
       orderid (Elsődleges int(11) AUTO\_INCREMENT): Egyedi azonosító, amely minden új rendelés hozzáadásakor automatikusan növekszik. Az elsődleges kulcsként szolgál a táblában.  
       userid (int(11)): Azonosító, amely a rendelést leadó felhasználóhoz kapcsolódik. Ez egy külső kulcs, amely a "user" tábla "userid" mezőjére hivatkozik.  
       productid (int(11)): Azonosító, amely a rendelt termékhez kapcsolódik. Ez egy külső kulcs, amely a "cipok" tábla "productid" mezőjére hivatkozik.  
       order\_date (TIMESTAMP DEFAULT CURRENT\_TIMESTAMP): A rendelés időpontját rögzíti. Alapértelmezett értékként a rendszer aktuális időpontját veszi fel.  
       quantity (int(11)): A rendelt termék mennyiségét rögzíti.  
       total\_amount (DECIMAL(10, 2)): A rendelés összértékét rögzíti. A DECIMAL típus lehetővé teszi a tizedes pontosságú értékek kezelését.

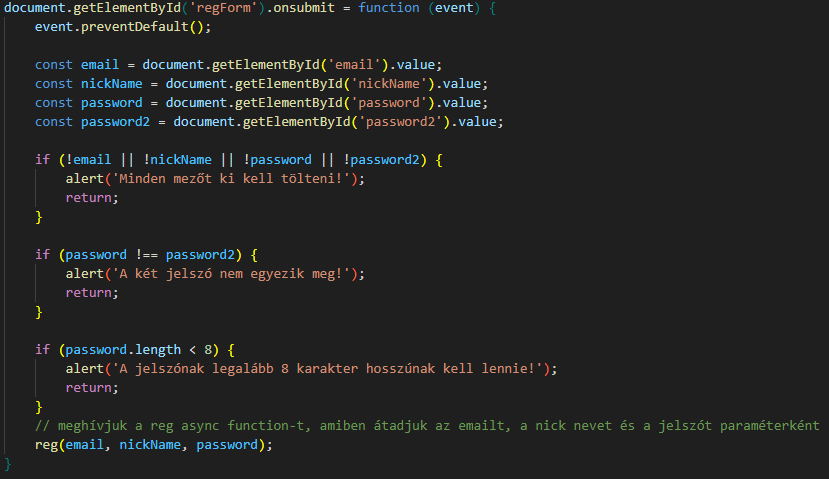
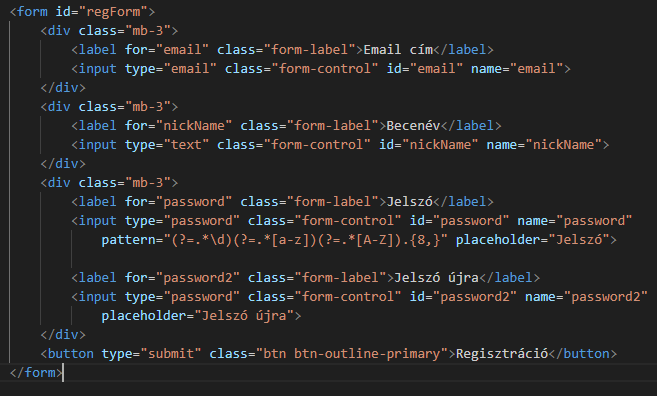


**Regisztráció a**A felhasználói regisztráció az online platformok egyik alapvető funkciója, amely lehetővé teszi az új felhasználók számára a hozzáférést és az interakciót az adott szolgáltatással vagy alkalmazással. A Danger Shop oldalán történő regisztráció is hasonló elveken alapul.Az első lépés a regisztrációs űrlap kitöltése. Ennek során a felhasználóknak meg kell adniuk néhány alapvető információt, mint például az email címüket, valamint egy becenevet és egy jelszót, amelyekkel majd be tudnak jelentkezni az oldalra. A jelszónak megfelelően kell teljesítenie a biztonsági követelményeket, például legalább nyolc karakter hosszúnak kell lennie, tartalmaznia kell kis- és nagybetűket, valamint számokat is.



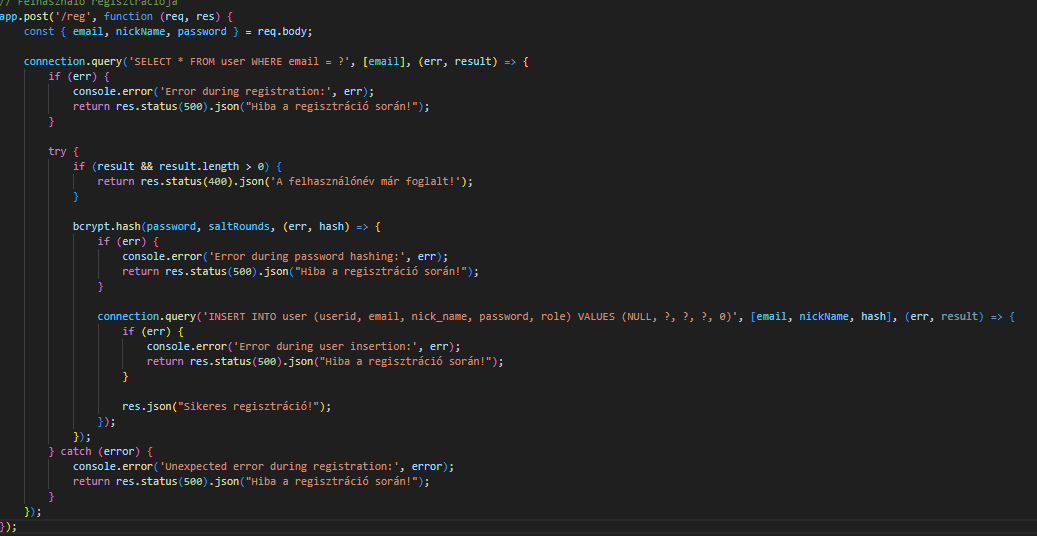
A regisztráció JavaSript által működik. Az *onkeyup* eseménykezelő figyeli a jelszómezőben történő gépelést, és ellenőrzi, hogy megfelelő-e a jelszó az elvárásoknak. Az *onsubmit* eseménykezelő megakadályozza az űrlap automatikus beküldését, és végrehajtja az összes ellenőrzést, mielőtt elküldené az adatokat a szervernek.

Ez a JavaScript kód felelős a jelszóellenőrzésért, az űrlap beküldésének kezeléséért és a regisztrációs adatok küldéséért a szervernek. Az onkeyup eseménykezelő figyeli a jelszómezőben történő gépelést, és ellenőrzi, hogy megfelelő-e a jelszó az elvárásoknak. Az onsubmit eseménykezelő megakadályozza az űrlap automatikus beküldését, és végrehajtja az összes ellenőrzést, mielőtt elküldené az adatokat a szervernek.

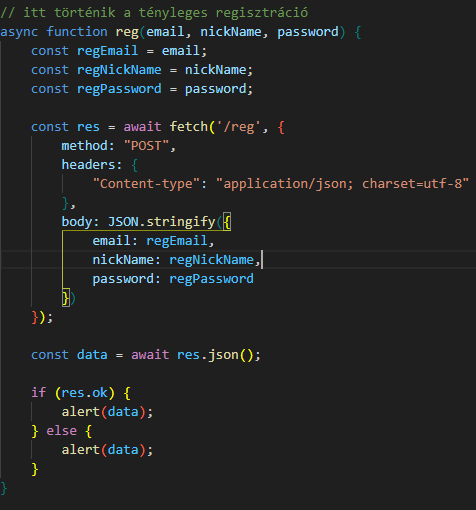


A regisztrációnál user szerepkört kapsz és ezt az admin tudja megváltoztatni.A felhasználó regisztrációja során az alkalmazás az "/reg" útvonalon figyeli a POST kéréseket, melyeket a kliens küld a regisztrációhoz. Az első lépés az adatok fogadása, melyeket a kliens a kérés testében küld. Ezek közé tartozik az email cím, a becenév és a jelszó. Ezt követően az alkalmazás ellenőrzi, hogy az adott email cím már szerepel-e az adatbázisban. Ehhez egy adatbázis lekérdezést hajt végre, és ha talál eredményt, visszaküldi a kliensnek, hogy az email cím már foglalt. Amennyiben az email cím még nem szerepel az adatbázisban, az alkalmazás hasheli a jelszót a bcrypt segítségével. Ezután az új felhasználó adatait beszúrja az adatbázisba. Minden lépés során az alkalmazás megfelelően kezeli a különböző hibákat, és visszajelzést küld a kliensnek azokról. Ezzel a folyamattal biztosítva van, hogy a felhasználó regisztrációja megbízható és biztonságos módon történjen.

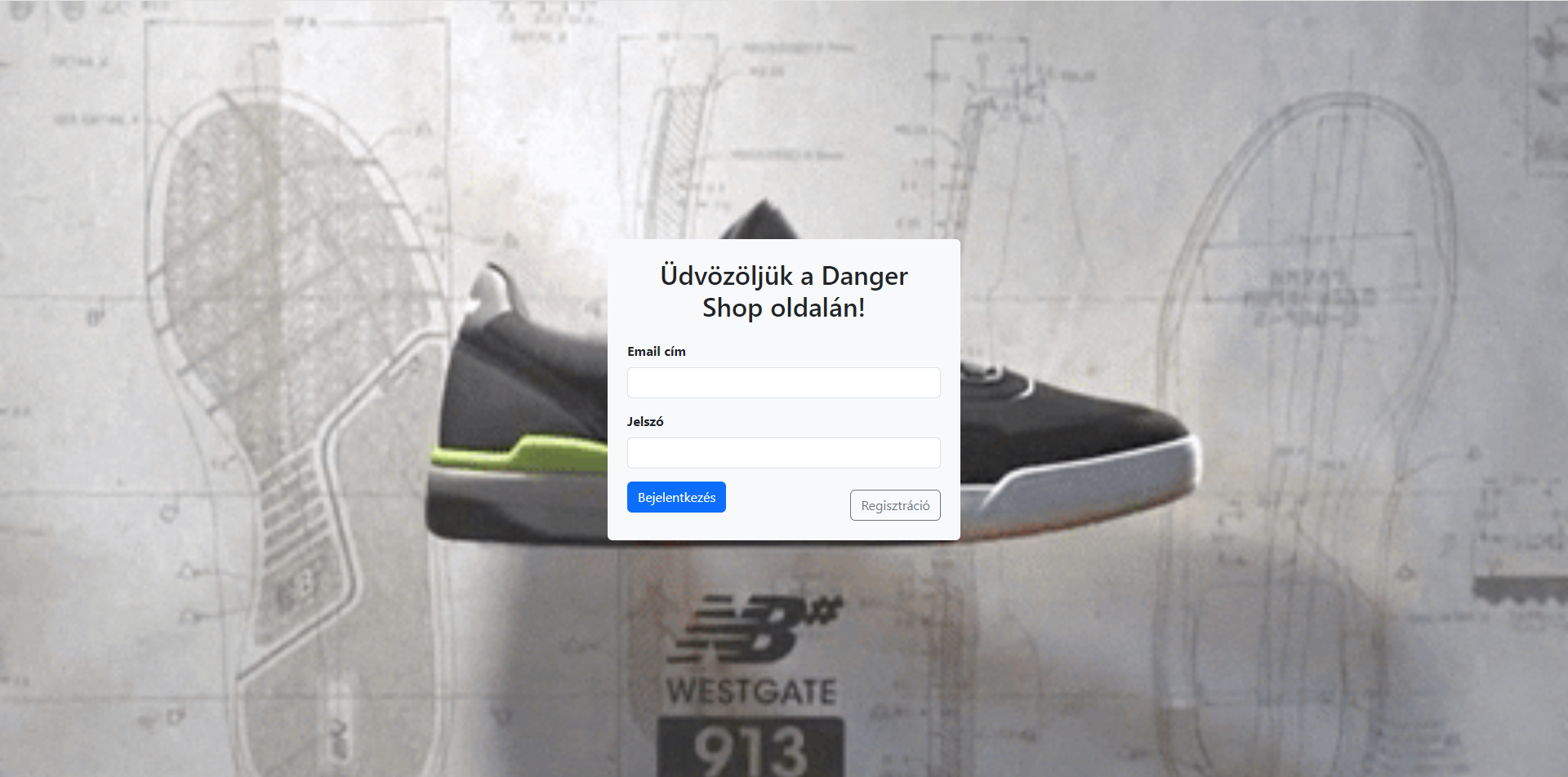
A Node.js és MySQL technológiák kombinációja lehetővé teszi az adatok hatékony kezelését és védelmét.



Az async function reg függvény felelős az új felhasználói adatok küldéséért a szervernek. A függvény az email, nickName és password paraméterekkel kapja meg a regisztrációhoz szükséges adatokat.  
A függvény először létrehozza a regEmail, regNickName és regPassword változókat, amelyek az adatokat tartalmazzák. Ezután az await fetch('/reg') segítségével HTTP POST kérést küld a /reg végpontnak a szerverrel történő kommunikációhoz.  
A kérés fejlécében beállítja a Content-type fejlécet JSON formátumra ("application/json; charset=utf-8"), majd a body-ban az adatokat JSON formátumban elküldi a szervernek.  
Miután a szerver válaszolt, az adatokat JSON formátumban várjuk és az await res.json() segítségével kinyerjük őket. Ha a válasz rendben volt (res.ok értéke igaz), akkor megjelenítünk egy üzenetet a felhasználónak, hogy a regisztráció sikeres volt (alert(data)). Ha hiba történt a regisztráció során, akkor szintén megjelenítünk egy üzenetet a hibaüzenettel.



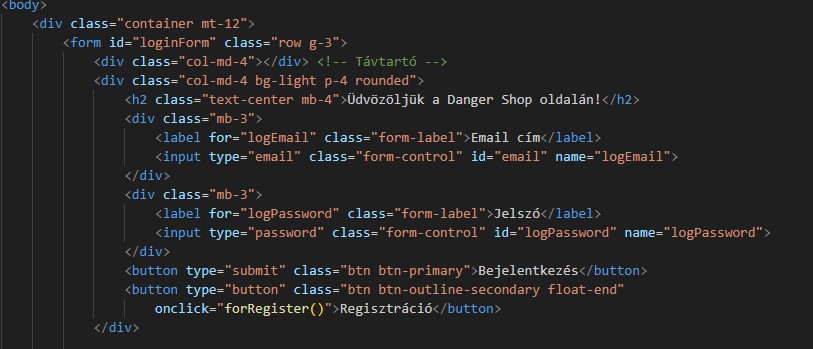
**Bejelentkezés**



A DANGER SHOP bejelentkezési oldala egyszerű és felhasználóbarát felületet kínál annak érdekében, hogy a felhasználók könnyedén hozzáférhessenek fiókjukhoz. Az oldal letisztult, ám hatékony dizájnja lehetővé teszi a gyors és zökkenőmentes bejelentkezést. Az oldal tetején található navigációs sáv segítségével könnyen lehet navigálni az oldalon, így a felhasználók könnyedén áttekinthetik a rendelkezésre álló lehetőségeket. Az oldal teljes mértékben reszponzív, így a különböző eszközökön, például asztali számítógépeken, laptopokon, tableteken és mobiltelefonokon egyaránt tökéletesen működik.

Az "Üdvözöljük a Danger Shop oldalán!" üzenet a felhasználókat barátságosan köszönti, és rögtön bevezeti őket az oldal használatába. A bejelentkezési űrlap alatt található mezők segítségével a felhasználók könnyedén megadhatják az e-mail címüket és jelszavukat, majd egyszerűen bejelentkezhetnek az oldalra.

A "Bejelentkezés" gombra kattintva a felhasználók azonnal beléphetnek fiókjukba, míg a "Regisztráció" gombra kattintva lehetőségük van új fiók létrehozására. Ez a folyamat gyors és egyszerű, így a felhasználók könnyedén regisztrálhatnak az oldalra, hogy élvezhessék a DANGER SHOP kínálta előnyöket és szolgáltatásokat. Az oldal alján található JavaScript fájl biztosítja az oldal sima működését, és lehetővé teszi a felhasználók számára a kényelmes használatot. Összességében a DANGER SHOP bejelentkezési oldala felhasználóbarát és hatékony felületet kínál annak érdekében, hogy a felhasználók könnyedén hozzáférjenek fiókjukhoz és élvezhessék az oldal nyújtotta előnyöket.



* 1. Home HTML

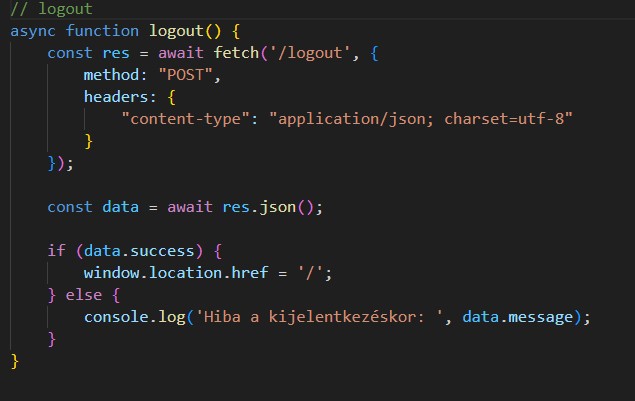
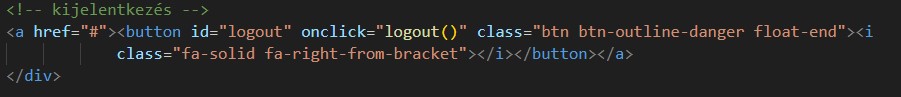
Fejléc Szekció (Header Section): A fejléc szekció az összes olyan információt tartalmazza, amelyet a böngésző a dokumentum betöltésekor elolvashat.

Ide tartoznak a meta tagok, a stíluslapok és a JavaScript fájlok hivatkozásai, valamint az oldal címe. A kódban a fejléc részben láthatók ezek a beállítások.



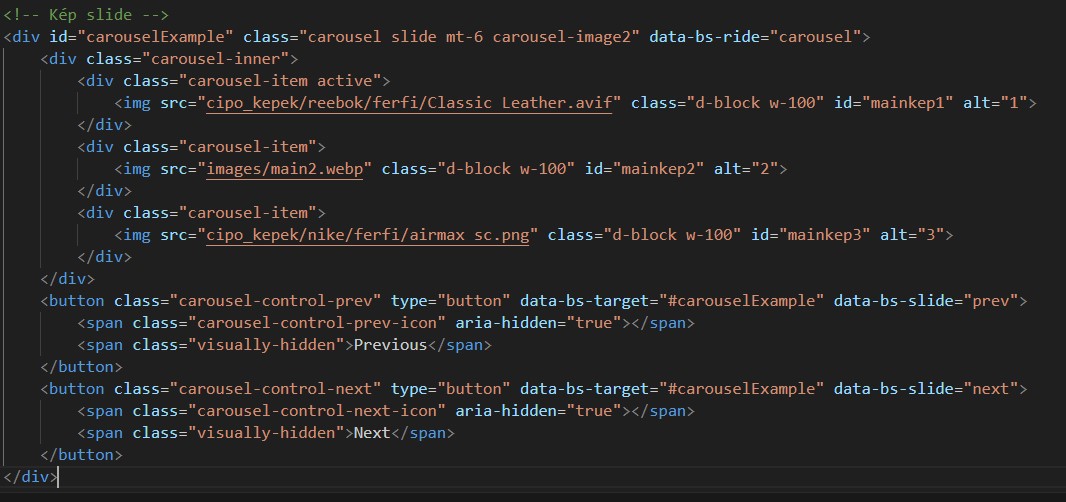
Ez a rész a HTML kódban egy kijelentkezési lehetőséget biztosít a felhasználók számára az "DANGER SHOP" online áruházban. Itt egy gomb található, amelyre kattintva a felhasználó kijelentkezhet az oldalról. A kód leírása: · <a href="#">: Ez egy hivatkozás (link), amelynek a href attribútuma üres string-re van beállítva, így ez a link nem mutat sehová. Ez azt jelenti, hogy a gomb megjelenik, de ha rákattintunk, nem történik semmi. · <button id="logout" onclick="logout()" class="btn btn-outline-danger float-end">: Ez egy gomb, amelyet "kijelentkezés" felirattal jelenít meg. A gomb stílusát a Bootstrap keretrendszer segítségével határozzák meg (btn, btn-outline-danger osztályok), és a jobb szélre igazítják (float-end osztály). · <i class="fa-solid fa-right-from-bracket"></i>: Ez egy ikon, amelyet a Font Awesome ikonkönyvtárból vesznek. Ez az ikon egy jobbra mutató nyíl, ami a kijelentkezést szimbolizálja. · onclick="logout()": Ez az attribútum hivatkozik egy JavaScript függvényre (logout()), amelynek a feladata a kijelentkezés végrehajtása.

Amikor a gombra kattintanak, ez a függvény lesz meghívva.

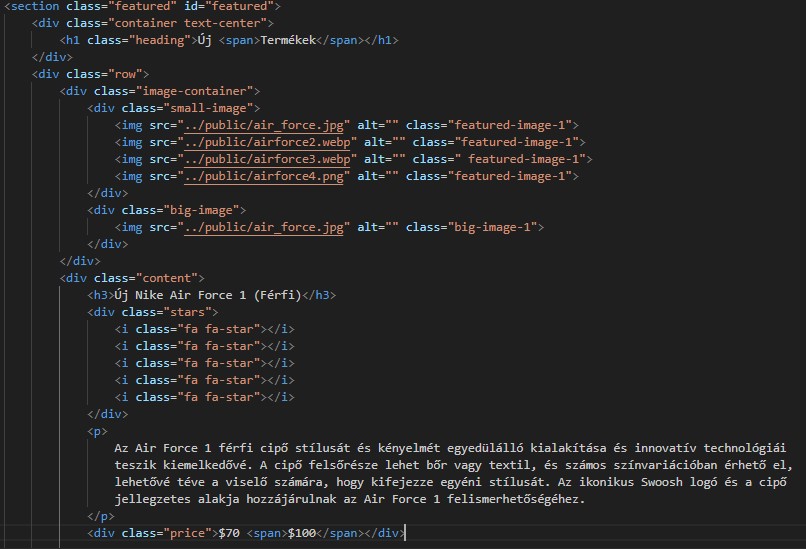


async function logout() { ... }: Ez egy aszinkron függvény definíciója, ami a kijelentkezést végrehajtja.

Az async kulcsszó jelzi, hogy ez a függvény aszinkron módon működik, és a logout nevű függvényt definiálja. const res = await fetch('/logout', { ... });: Ez a rész aszinkron módon küld egy POST kérést az '/logout' végpontra a fetch() segítségével. A kérés a szervernek jelzi, hogy a felhasználó kijelentkezett. const data = await res.json();: Ez a rész aszinkron módon várja a választ (res) a kijelentkezési kérésre, majd a választ JSON formátumban olvassa (res.json()). if (data.success) { ... } else { ... }: Ez a rész ellenőrzi a választ (data), hogy a kijelentkezés sikeres volt-e vagy sem. Ha data.success értéke true, akkor a felhasználót sikeresen kijelentkeztették, és az oldalt a főoldalra ('/') irányítja. Ha data.success értéke false, akkor a konzolra kiírja a hibaüzenetet (data.message), amelyet a szerver visszaküldött. window.location.href = '/';: Ha a kijelentkezés sikeres volt, akkor az oldalt átirányítja a főoldalra ('/').



A kép csúszka lehetővé teszi a felhasználók számára, hogy különböző képeket nézzenek meg egy adott területről, például a kiemelt termékekről vagy kategóriákról. A kód leírása: <div id="carouselExample" class="carousel slide mt-6 carousel-image2" data-bs-ride="carousel">: Ez a rész egy konténert hoz létre a kép csúszkához. Az id="carouselExample" attribútum azonosítja az elemet, amelyet később hivatkozni lehet. Az class="carousel slide mt-6 carousel-image2" osztályok a Bootstrap keretrendszer carousel komponensét alkalmazzák a dizájnhoz. A data-bs-ride="carousel" attribútum a Bootstrap által használt adattag, ami jelezni fogja a böngészőnek, hogy ez egy automatikusan járó kép csúszka. <div class="carousel-inner">: Ez a rész a kép csúszka belső tartalmát határozza meg. Ide kerülnek be a különböző képek. Kép elemek (<div class="carousel-item">): Ezek a részek tartalmazzák az egyes képeket, amelyek megjelennek a csúszkában. Minden carousel-item egy különálló képet reprezentál. Az active osztály a kezdőképet jelöli ki a csúszkában. Képek (<img>): Ezek a részek tartalmazzák magukat a képeket, amelyek megjelennek a csúszkában. Az src attribútum azonosítja a kép forrását, míg az alt attribútum a kép alternatív szövegét adja meg, ami hasznos lehet a képek nem megjelenése esetén. Navigációs gombok (<button>): Ezek a gombok lehetővé teszik a felhasználók számára, hogy lapozzanak előre vagy hátra a csúszka képei között. A carousel-control-prev gomb az előző képre lapoz, míg a carousel-control-next gomb a következőre. A data-bs-target és data-bs-slide attribútumok azonosítják, hogy melyik csúszkát kell módosítani a gomb használatakor.



A szakaszban több termék is megjelenik, mindegyiknek külön-külön van egy képe és hozzá tartozó információk, például a termék neve, leírása és ára. A kód részlet leírása: <section class="featured" id="featured">: Ez a szakasz a weboldalon a kiemelt termékeket mutatja be. Az id="featured" attribútum segítségével hivatkozni lehet erre a szakaszra más részekből. <div class="container text-center"> ... </div>: Ez a rész a szakasz közepén egy konténert definiál, amely középre igazítja a benne lévő tartalmat. <h1 class="heading">Új <span>Termékek</span></h1>: Ez a cím jelzi, hogy a szakaszban bemutatott termékek újak. A heading osztály segítségével lehet formázni a címet. Termékek bemutatása egyenkénti blokkokban (<div class="row"> ... </div>): Minden termék blokkja egy <div class="row"> elemmel kezdődik, amely egy sorba rendezett elemeket tartalmaz.

Ez a struktúra biztosítja a termékek egymástól való elválasztását. Termék képei (<div class="image-container"> ... </div>): Minden termék blokkja tartalmaz egy kép konténert, amelyben a kis és nagy méretű képek találhatók. A kis képek a termék előnézeti képeit mutatják, míg a nagy kép a kiválasztott termék részletes képe. Termék információk (<div class="content"> ... </div>): Minden termék blokk tartalmaz egy információs részt, amelyben megtalálhatóak a termék leírása, címe és ára. <h3>: A termék címe a <h3> elem segítségével jelenik meg, amely a termék nevét tartalmazza. <p>: A termék leírása egy <p> elembe van beágyazva, amely a termék tulajdonságait és jellemzőit részletezi. <div class="price">: A termék ára a <div class="price"> elembe van beágyazva, ami megjeleníti a termék aktuális árát és az esetleges kedvezményt. <div class="stars">: A termék értékelését a <div class="stars"> elem jelzi, amely csillagok formájában jeleníti meg az értékelést.

* 1. **Férfi és női HTML**

**Navigáció** (<nav>) és Menüpontok (<ul>): A navigációs sáv (<nav>) tartalmazza az oldal menüjét és a vissza gombot. A menüpontok (<ul>) a különböző oldalakra mutató linkeket tartalmazzák.

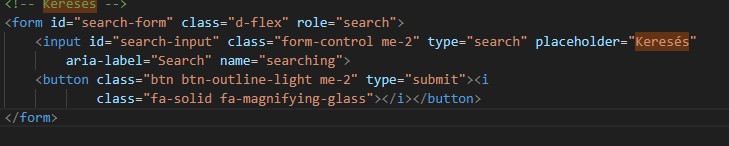
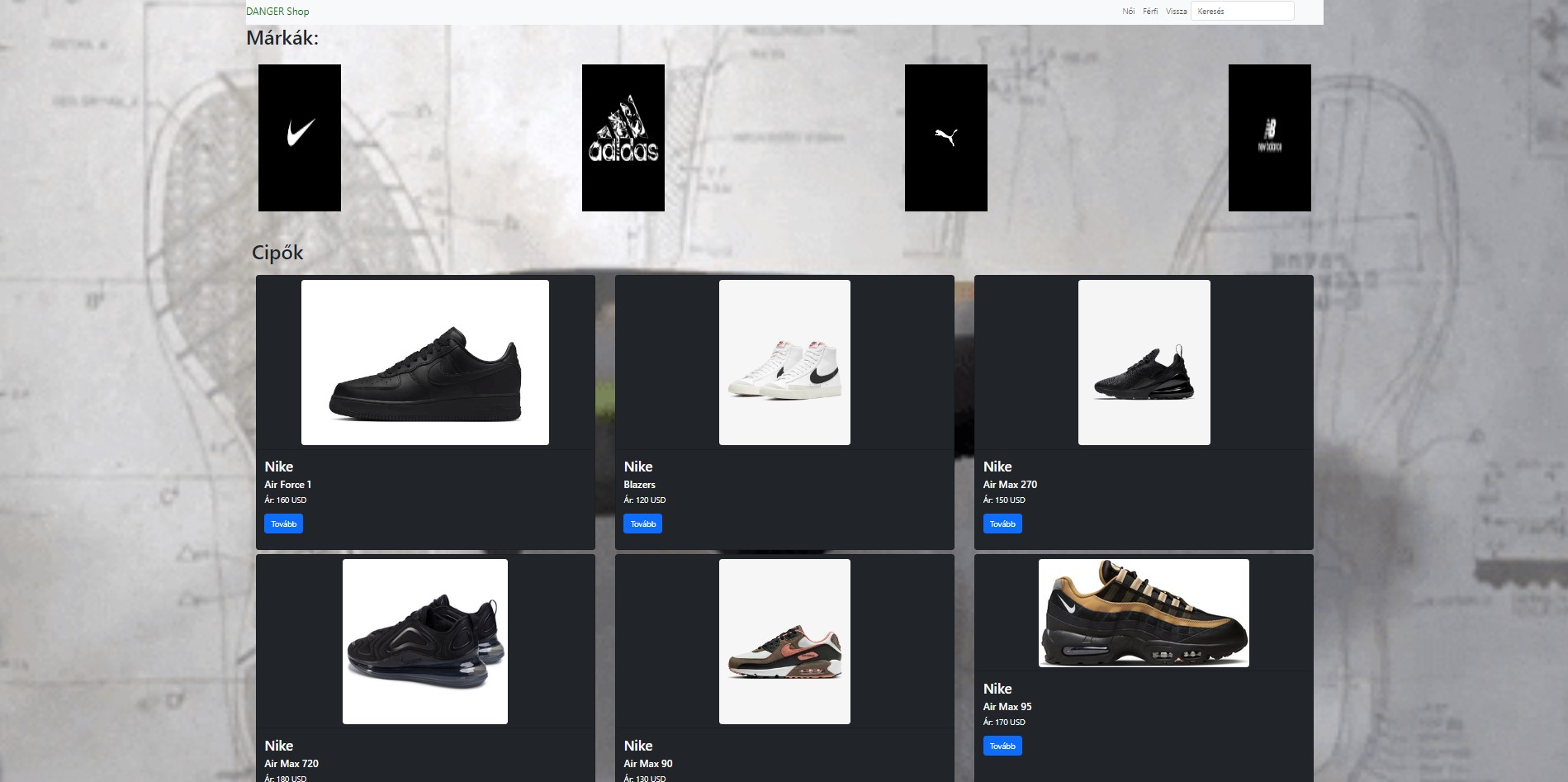
**Keresés**: A keresőmező egy form elem (<form>), amely lehetővé teszi a felhasználók számára, hogy termékeket keressenek az oldalon.

A keresőmezőt egy input mező (<input>) és egy keresés gomb (<button>) alkotja.

**Márkák**: A márka logók képeit tartalmazó rész (<div class="brand-images">), amelyeket a felhasználók megtekinthetnek az oldalon.

**Termékek lekérdezése**: Egy gomb (<button>) és egy konténer (<div id="cipo-list">) található, amely a telefonok lekérdezéséhez szolgál.

**JavaScript fájl hivatkozása**: A script.js fájl hivatkozása, amely tartalmazza az oldal interaktív funkcionalitásának definícióit.

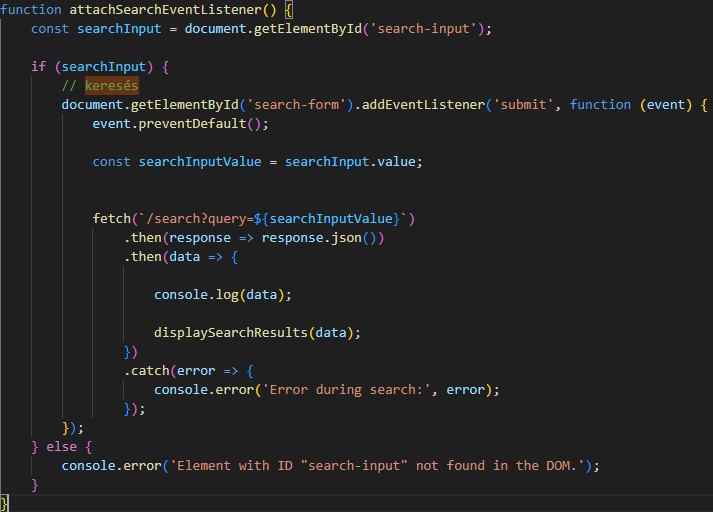


**HTML Űrlap** (<form>): Az űrlap azonosítója search-form, amelyet a JavaScript kód később referenciálhat. Az id attribútum segítségével azonosítható az űrlap.

**Keresőmező** (<input>): A <input> elem definiálja a keresőmezőt. id="search-input": Azonosító, ami lehetővé teszi a JavaScript kód számára, hogy hozzáférjen a mezőhöz. class="form-control me-2": Stíluskészlet, amely meghatározza a mező stílusát és elrendezését.

A form-control osztály például a Bootstrap keretrendszer része, ami megjeleníti a mezőt a megfelelő formázással. type="search": A mező típusának beállítása keresésre. Ez a böngészőkben speciális kezelést biztosít a keresőmező számára. placeholder="Keresés": Az üres mezőben megjelenő szöveg, ami segíti az felhasználót a keresés megkezdésében. aria-label="Search": Segíti az akadálymentesítést, azáltal, hogy jelzi a képernyőolvasóknak, hogy ez egy keresőmező.

**Keresés gomb** (<button>): A <button> elem a keresési művelet indításához szolgál. class="btn btn-outline-light me-2": Stíluskészlet a gomb megjelenésének beállításához. A btn-outline-light osztály például egy világos színű körvonalas gombot jelent a Bootstrap keretrendszerben. <i class="fa-solid fa-magnifying-glass"></i>: Egy ikon (lencse) megjelenítése a gombon, ami a keresési műveletet szimbolizálja. Ez az ikon a Font Awesome ikonkönyvtárból származik.



**attachSearchEventListener() függvény**: Ez a függvény felelős az eseménykezelő hozzáadásáért a keresőmezőhöz. Az eseménykezelő meghallgatja a keresőmezőn történő űrlap elküldését (submit esemény) vagy az "Enter" gomb lenyomását.

**const searchInput = document.getElementById('search-input')**;: Először a kód meghatározza a keresőmezőt az **id="search-input"** alapján. Ez a változó tartalmazza majd a felhasználó által bevitt keresőszót.

**Eseménykezelő hozzáadása**: A kód a search-form űrlapon történő submit eseményt figyeli meg. Amikor a felhasználó elküldi a keresőmezőben lévő adatokat, az eseménykezelő elkapja ezt az eseményt.

**event.preventDefault()**;: Ez a sor megakadályozza az űrlap alapértelmezett viselkedését, amely az oldal újratöltését eredményezné az űrlap elküldésekor. Ehelyett a JavaScript kód kezeli a keresési műveletet.

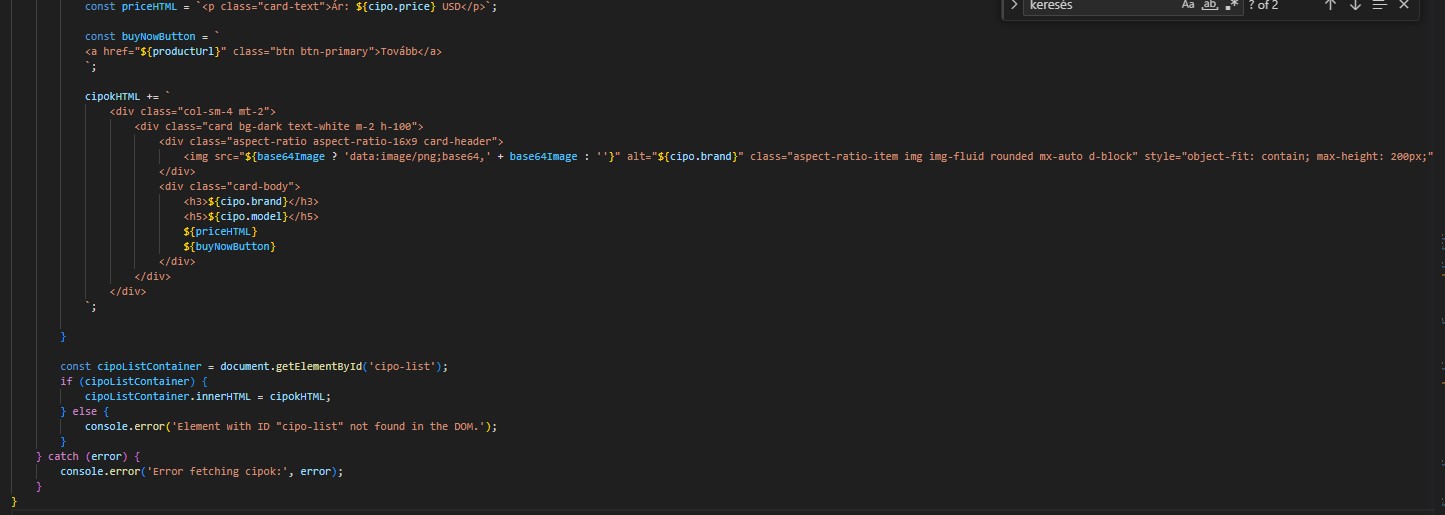
**Fetch kérés**: A kód egy GET kérést indít a /search végpont felé, amely a felhasználó által megadott keresési szót tartalmazza.

**Promise láncolás**: A kód .then() metódusokat használ a fetch kérés válaszának kezelésére. A válasz a .json() metódussal olvasható formátumba alakítódik át.

**Adatok megjelenítése**: A kapott adatokat megjeleníti a console.log() hívás és a displaySearchResults(data) függvény segítségével, amelynek a megvalósítását nem látjuk a kódrészletben.

**Hibakezelés**: A .catch() metódus a hiba esetén fut le, és a hibaüzenetet kiírja a konzolra.

**Eltérések kezelése**: Ha a keresőmező (searchInput) nem található az oldalon, a kód hibaüzenetet ír ki a konzolra. Ez azért fontos, hogy a kód hibamentesen fusson, még akkor is, ha valami váratlan dolog történik az oldalon.



Kiiratás

**priceHTML változó**: Ez a változó egy HTML szövegsorozatot tartalmaz, amely az aktuális cipő árát jeleníti meg. A cipo.price változó tartalmazza az árat, amelyet beillesztünk az HTML-be.

**buyNowButton változó**: Ez a változó egy HTML szövegsorozatot tartalmaz, ami egy "Tovább" gombot hoz létre, amelynek a linkje a productUrl változó értékére mutat. **cipokHTML változó**: Ez a változó tartalmazza az összes cipőhöz kapcsolódó HTML kódokat, amelyeket majd megjelenítünk az oldalon. Minden cipőhöz egy újabb kártya elemet hoz létre. Az ár és a "Tovább" gomb is beillesztésre kerül a kártya tartalmába a priceHTML és a buyNowButton változókból.

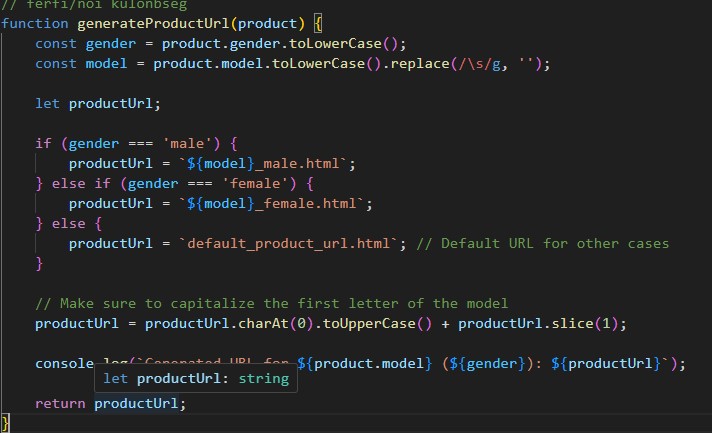
**HTML generálás**: A cipokHTML változóhoz folyamatosan hozzáadódnak a cipőkhöz tartozó kártya elemek.

Minden cipő egy külön kártya elemet kap, amely tartalmazza a már korábban említett árat és "Tovább" gombot.

**Elemek beillesztése az oldalba**: Miután az összes cipőhöz kapcsolódó HTML kód elkészült, azokat beilleszti az oldal egy adott részébe, amelynek az azonosítója "cipo-list".

Ha ez az elem nem található az oldalon, akkor hibaüzenetet ír ki a konzolra. **Hibakezelés**: A kód egy try-catch blokkot is tartalmaz, amely a kivételeket kezeli. Ha bármilyen hiba lép fel a folyamat során, azt a kód elkapja, és a konzolra kiírja a hibaüzenetet.

Ez a kód tehát dinamikusan generálja és jeleníti meg a cipőket tartalmazó kártyákat az oldalon, amelyekhez ár és további gomb is tartozik. A kód figyel arra is, hogy ha az oldalon nincs megfelelő hely a cipők megjelenítésére, akkor hibaüzenetet ír ki.



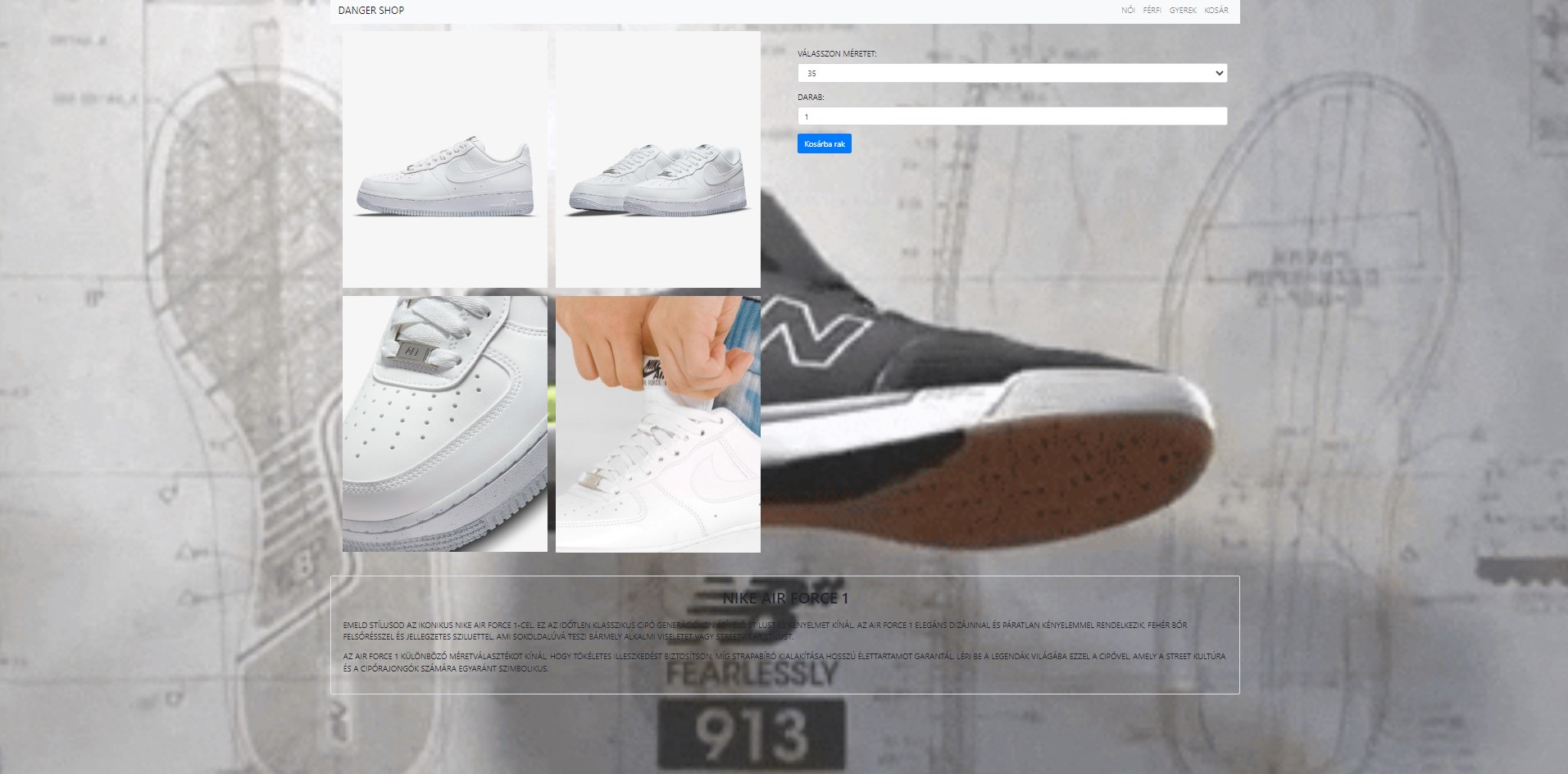
**Bemenet**: A függvény egy product objektumot vesz bemenetként, amely információkat tartalmaz a termék neméről és modelljéről.

**Feldolgozás**: A függvény először kinyeri a nem és a modell információkat a product objektumból. Ezután URL-t állít össze a nem és a modell alapján.

Férfi termékek esetén hozzáfűzi az "\_male" szót a modell nevéhez, és ".html" kiterjesztést. Női termékek esetén ugyanezt teszi, de az "\_female" szót fűzi hozzá helyette. Ha a nem sem férfi, sem nő, akkor egy alapértelmezett URL-t állít be, default\_product\_url.html néven. A függvény gondoskodik arról, hogy a modell első betűje nagybetű legyen a generált URL-ben. Végül a generált URL-t naplózza a konzolba hibakeresési célokra. **Kimenet**: A függvény visszaadja a generált termék-URL-t.

Ez a funkció rendkívül fontos, mivel az adatbázisban ugyanolyan névvel szerepelhetnek mind a férfi, mind a női cipők.

* 1. **Cipő HTML-ek**

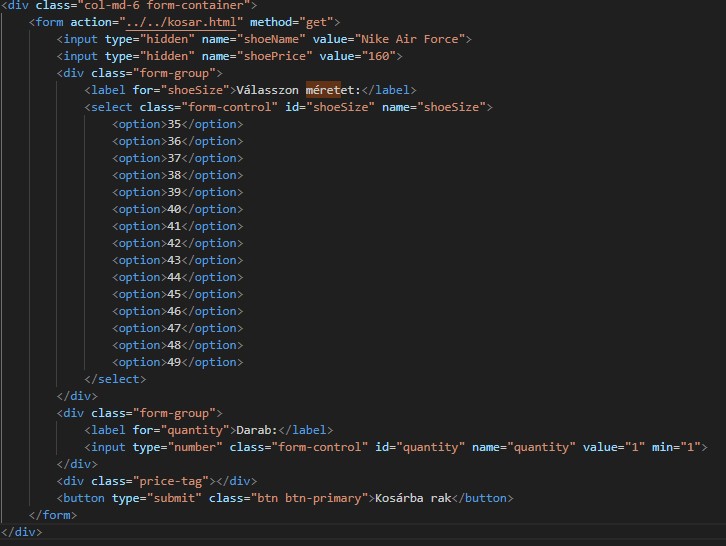


Mindegyik cipőhőz külön HTML oldal társul.

**Navigáció**: Az oldalon egy navigációs sáv található, amely lehetővé teszi a felhasználók számára a könnyű böngészést a különböző kategóriák, mint a Női, Férfi, Gyerek és Kosár között.

**Termék megjelenítése és vásárlás**: Az oldal középpontjában egy Nike Air Force 1 cipő látható, amelyhez a felhasználók kiválaszthatják a méretüket és megvásárolhatják a kosárba rakás gomb segítségével. Emellett több kép és még egy videó is látható a cipőről.

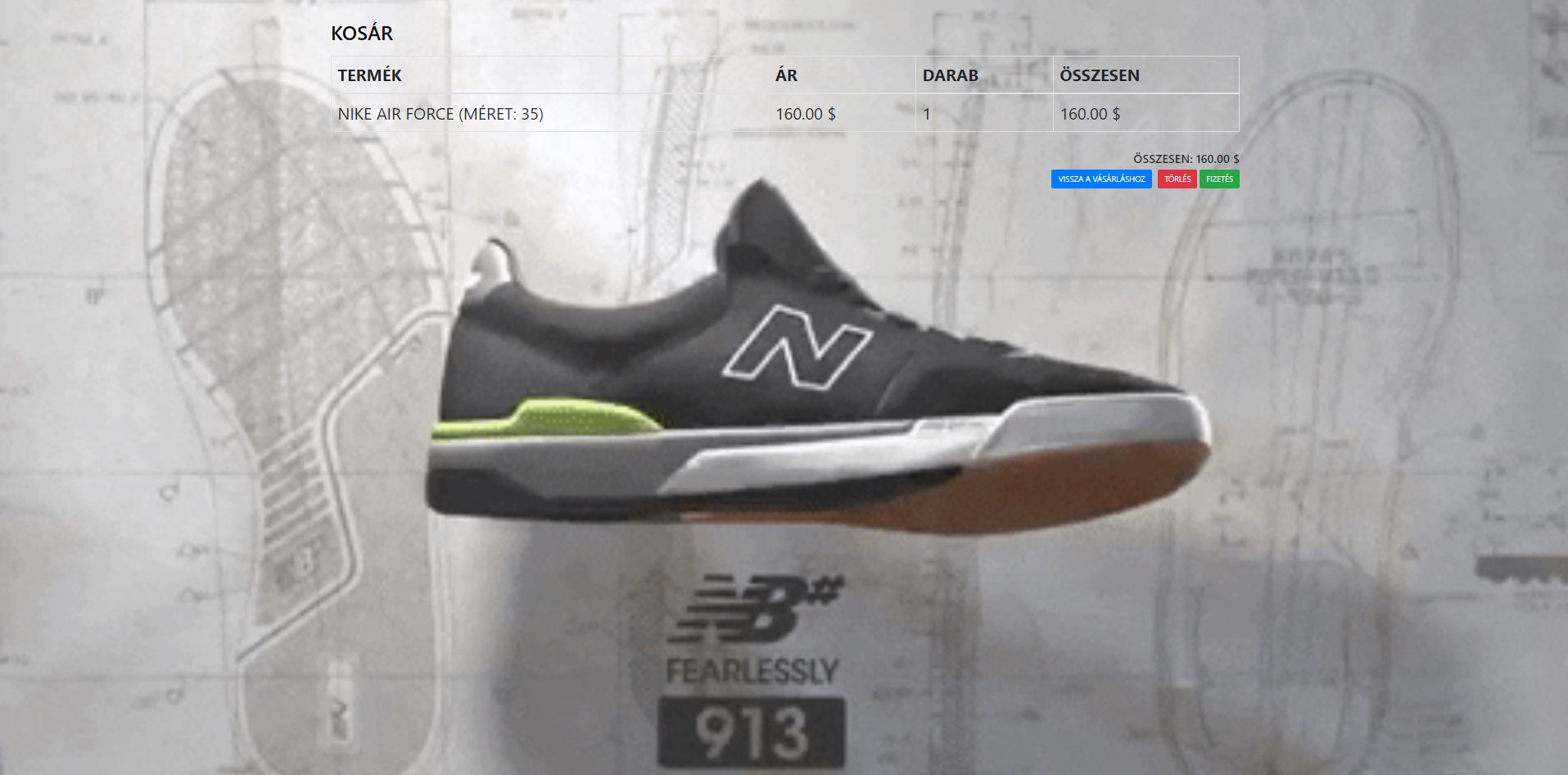
**Termék leírása**: Az oldal alján egy részletes leírás található a Nike Air Force 1 cipőről, amely bemutatja annak stílusát, kényelmét és tartósságát.



A form action attribútuma az, hogy az űrlap elküldésekor hova navigáljon az oldal.

A method attribútum meghatározza, hogy az űrlap elküldése milyen módon történjen, ebben az esetben get módszerrel. Az űrlap tartalmaz két rejtett input mezőt (input[type="hidden"]), amelyek az adott cipő nevét és árát tartalmazzák. Ezek az adatok továbbítódnak a kosár oldalra, amikor a felhasználó hozzáadja a terméket a kosárhoz.

Ezután két form elem van jelen, amelyek lehetővé teszik a felhasználó számára, hogy kiválassza a termék méretét és megadja, hogy hány darabot szeretne vásárolni belőle. A méretet egy legördülő lista (select) segítségével lehet kiválasztani, amely a lehetőségeket tartalmazza az 35-től 49-ig terjedő méretekben. A darabszámot egy szám beviteli mező (input[type="number"]) segítségével lehet megadni, ahol minimum 1 darabot kell megadni. Végül a "Kosárba rak" gombra kattintva az űrlap elküldődik a ../../kosar.html oldalra, és a kiválasztott termék a megadott mennyiségben hozzá lesz adva a kosárhoz.



**DOMContentLoaded Eseményfigyelő**: Ez a script akkor fut le, amikor a weboldal DOM tartalma teljesen betöltődik. Változókat inicializál az URL lekérdezési karakterláncából származó paraméterekkel.

Ezek a paraméterek tartalmazzák a cipő nevét (shoeName), árát (shoePrice), méretét (shoeSize) és mennyiségét (quantity). Számolja a teljes árat (totalPrice) a cipő árából és mennyiségéből.

**Elem hozzáadása a Kosárhoz**: Ellenőrzi, hogy van-e táblázat a .table-bordered osztállyal, ami feltehetően a bevásárlókosarat jelképezi. Ha talál, létrehoz egy táblasort (<tr>) és hozzáadja a tábla tartalmához (<tbody>). Ez a sor tartalmazza a cipő részleteit, többek között a nevét, méretét, egységárat, mennyiséget és teljes árat.

**Teljes Ár Frissítése**: Ha van egy elem az totalPrice azonosítóval, akkor frissíti a tartalmát, hogy megjelenítse a kosárban lévő összes tételek teljes árát.

**Kosár Ürítése**: A clearTable() függvényt definiálja annak érdekében, hogy törölje a bevásárlókosár táblázatának tartalmát. Ez eltávolítja az összes táblasort a táblából, és a teljes árat nullára állítja.

**Mennyiség Növelése**: Az increaseQuantity() függvény eggyel növeli a kiválasztott tétel mennyiségét. Megkeresi az quantity azonosítójú bemeneti elemet, lekéri annak aktuális értékét, növeli azt, majd frissíti a bemeneti elemet az új értékkel.

**Sikerüzenet Megjelenítése**: Az showSuccessMessage() függvény egy figyelmeztetést jelenít meg, ami egy sikeres vásárlást jelöl. Ezt a funkciót akkor lehet hívni, ha a vásárlás sikeresen megtörtént.

* 1. Admin HTML

Ez a HTML kód egy weboldal fejlécét és navigációs menüjét határozza meg egy Danger Shop nevű webáruházhoz, amelyet adminisztrátoroknak terveztek. Nézzük meg részletesebben:

Metaadatok és Cím: Az <head> részben beállítják a karakterkódolást, a nézetport méretét és a weboldal címét. Stílusok és Fájlok Betöltése: A fejléc a saját style.css stíluslapját és a Bootstrap keretrendszer bootstrap.min.css fájlját tölti be. Emellett ikonokat is tartalmaz a Font Awesome külső CSS fájlján keresztül.

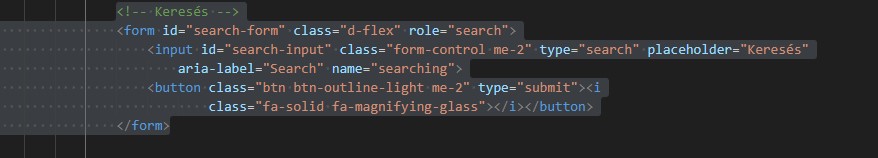
Navigációs Menü: A navigációs menü a Bootstrap keretrendszer navbar osztályát használja, ami responsív elrendezést biztosít. Tartalmaz egy "Profil" gombot, amely az adminisztrátor profilszerkesztő oldalára visz, valamint "Női" és "Férfi" linkeket az áruház különböző részeire. Egy "Felhasználók szerkesztése" ikon is található a menü jobb felső sarkában. Üdvözlő Szöveg: A <div id="szoveg"> rész tartalmazza az üdvözlő szöveget és a felhasználóknak szóló bevezetőt. Ez egy információs szakasz, amely üdvözli az adminisztrátort az admin felületen, és lehetőséget biztosít az áruház különböző részeire való navigálásra. Offcanvas Menü és Modális Ablakok: Az "editProfile" nevű offcanvas menü és a modális ablakok (editProfileUsername, editProfilePasswordModal) lehetővé teszik a felhasználói profil szerkesztését, beleértve a felhasználónév és a jelszó módosítását. Script Fájlok Betöltése: Végül a script fájlok betöltődnek, beleértve az admin.js, profile.js, és logout.js fájlokat, amelyek várhatóan a kliensoldali interakciókat és műveleteket irányítják.



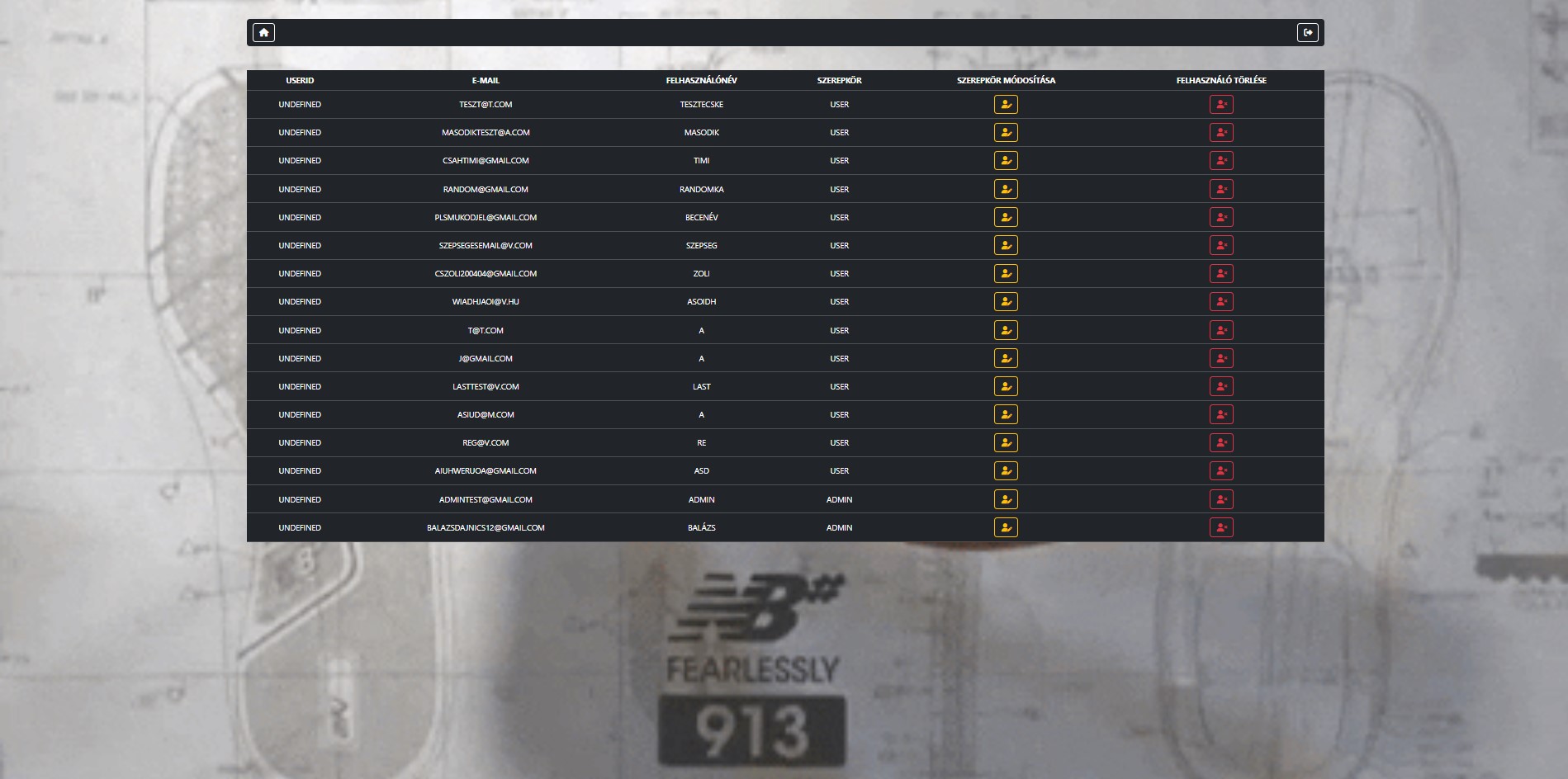
Férfi és a női felület

Metaadatok és Cím: Az oldal <head> részében beállítják a karakterkódolást, a nézetport méretét, és megadják az oldal címét. Stíluslapok és Fájlok Betöltése: A <head> rész tartalmazza a saját style2.css stíluslapját, a Bootstrap keretrendszer bootstrap.min.css fájlját, valamint a Font Awesome ikonok stíluslapját is. Szkript Fájlok Betöltése: Az oldal alján betöltődnek a JavaScript fájlok, beleértve az admin.js és a script.js fájlokat, amelyek felelősek az adminisztrációs felület működéséért. Navigációs Menü: A Bootstrap keretrendszer segítségével kialakított navigációs menü tartalmaz linkeket a női és férfi termékek adminisztrációs oldalaira, valamint egy keresőmezőt is. Új Cipő Felvétele Űrlap: Az adminisztrátorok új cipőket adhatnak hozzá az oldalhoz ezen az űrlapon. A mezők lehetővé teszik a cipő modelljének, márkanévének, árának, nemének és képének megadását. Cipők Listája: Az oldal megjeleníti a már meglévő cipők listáját egy div elem segítségével, melynek azonosítója cipo-list. Itt jelennek meg dinamikusan az adminisztrátor által hozzáadott vagy módosított cipők. Cipő Szerkesztése Modal Ablak:

Egy modális ablak jelenik meg, amikor az adminisztrátor szerkeszteni kíván egy cipőt. A modális ablakban lehetőség van a cipő modeljének, márkanévének, árának, raktárkészletének, nemének és képének szerkesztésére.



Form: A <form> elem egy űrlapot reprezentál, melynek azonosítója "search-form". Az "d-flex" osztály beállítása segítségével a form elemei vízszintesen rendeződnek el. Input Mező: Az <input> elem egy szövegbeviteli mezőt hoz létre, melynek azonosítója "search-input". A "form-control" és "me-2" osztályok segítségével a mező stílusát alakítják ki. A "type" attribútum értéke "search", ami egy keresőmezőt jelöl. A "placeholder" attribútum értéke "Keresés", ami egy segítséget nyújtó szöveget jelenít meg a mezőben. Az "aria-label" attribútum segítségével a keresőmezőhöz egy hozzáférhető címkét adnak. Gomb: A <button> elem egy gombot hoz létre a keresőmezőhöz. A "btn", "btn-outline-light" és "me-2" osztályok segítségével a gomb stílusát alakítják ki. A "type" attribútum értéke "submit", ami azt jelzi, hogy ez a gomb az űrlap elküldését indítja el. Az ikon megjelenítéséhez a Font Awesome ikonkészletből a "fa-solid fa-magnifying-glass" osztályokat használják.



Metaadatok és Cím: Az oldal <head> részében beállítják a karakterkódolást, a nézetport méretét, és megadják az oldal címét. Stíluslapok és Fájlok Betöltése: A <head> rész tartalmazza a saját style.css stíluslapját, a Bootstrap keretrendszer bootstrap.min.css fájlját, valamint a Font Awesome ikonok stíluslapját is. Szkript Fájlok Betöltése: Az oldal alján betöltődnek a JavaScript fájlok, beleértve az editUsers.js és a logout.js fájlokat, amelyek felelősek a felhasználók kezeléséért és a kijelentkezés funkcióért. Navigációs Menü: A Bootstrap keretrendszer segítségével kialakított navigációs menü tartalmaz egy "ház" ikont, amely visszavezet az adminisztrációs főoldalra, valamint egy keresőmezőt és egy "Kijelentkezés" gombot. Felhasználók Táblázata: Az oldal megjeleníti a felhasználók adatait egy táblázatban, amely tartalmazza az azonosítójukat, e-mail címüket, felhasználónevüket, szerepkörüket, valamint lehetőséget biztosít a szerepkör módosítására és a felhasználó törlésére.

Felhasználó Szerepkörének Módosítása Modal Ablak: Egy modális ablak jelenik meg, amikor az adminisztrátor módosítani kívánja egy felhasználó szerepkörét.

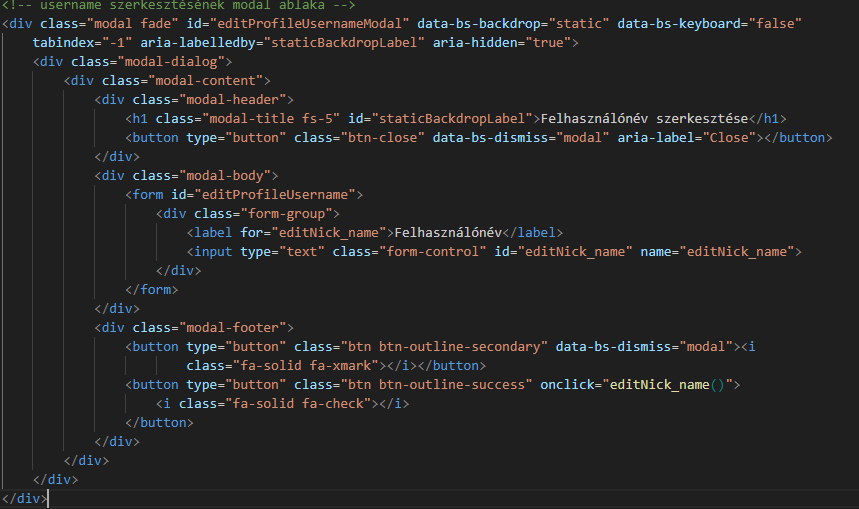
A modális ablakban lehetőség van a szerepkör megadására egy szám formájában, ahol 0 az alapértelmezett felhasználói szerepkört, míg 1 az adminisztrátori szerepkört jelöli.

* 1. Admin profil felület



<div class="modal fade" id="editProfileUsernameModal" ...>: Ez a <div> elem egy modális ablakot definiál a "modal" és "fade" osztállyal, ami azt jelzi, hogy be- és kifakadás van, amikor megnyílik/bezárul. Az "editProfileUsernameModal" azonosítóval rendelkezik, hogy hivatkozni lehessen rá JavaScriptben vagy CSS-ben. data-bs-backdrop="static" és data-bs-keyboard="false":

Ezek az attribútumok megakadályozzák, hogy a modális ablakot a külső részre kattintva vagy az Esc billentyű megnyomásával lehessen bezárni. A modális ablakon belül egy <div class="modal-dialog"> található, ami beállítja a modális tartalom méretét és igazítását. <div class="modal-content">: Ez a <div> tartalmazza a modális ablak teljes tartalmát. <div class="modal-header">: Ez a <div> tartalmazza a modális ablak fejlécét, amely tartalmaz egy "Felhasználónév szerkesztése" című fejlécet, valamint egy bezárás gombot (<button type="button" class="btn-close" ...>). <div class="modal-body">: Ebben a <div>-ben egy <form> elem van az "editProfileUsername" azonosítóval. Tartalmaz egy felhasználónév input mezőt a "Felhasználónév" címkével. <div class="modal-footer">: Ez a <div> tartalmazza a modális ablak láblécét, amely két gombot tartalmaz: Egy gomb a "btn-outline-secondary" osztállyal és a data-bs-dismiss="modal" adatattribútummal, ami bezárja a modális ablakot anélkül, hogy mentené a változtatásokat. Másik gomb a "btn-outline-success" osztállyal, amely akkor hívja meg az editNick\_name() függvényt, amikor rákattintanak. Ez a függvény valószínűleg felelős a szerkesztett felhasználónév elküldéséért.

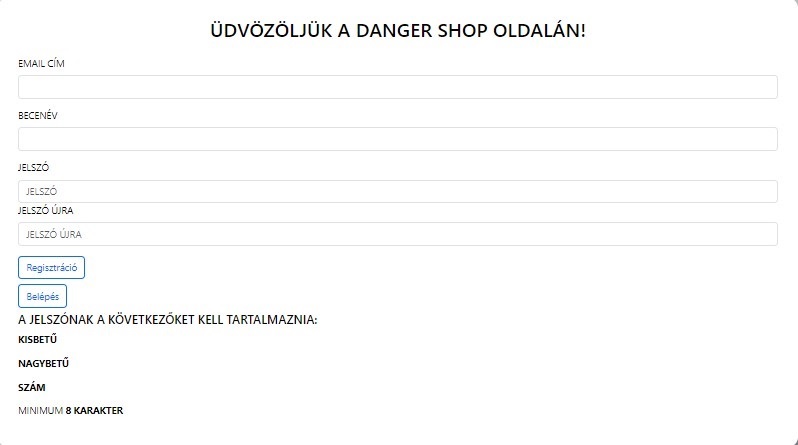


Ez a kód egy modális ablakot definiál egy "modal" és "fade" osztállyal rendelkező <div> elem segítségével, ami azt jelzi, hogy be- és kifakadás van, amikor megnyílik/bezárul.

Az "editProfilePasswordModal" azonosítóval rendelkezik, hogy hivatkozni lehessen rá JavaScriptben vagy CSS-ben. A data-bs-backdrop="static" és data-bs-keyboard="false" attribútumok megakadályozzák, hogy a modális ablakot a külső részre kattintva vagy bármely billentyű lenyomásával lehessen bezárni. A modális ablakon belül egy <div class="modal-dialog"> található, ami beállítja a modális tartalom méretét és igazítását. A modális tartalom minden tartalmát egy <div class="modal-content"> elem tartalmazza. A modális ablak fejléce egy <div class="modal-header"> elembe van helyezve, ami tartalmazza a "Jelszó módosítása" című fejlécet és egy bezárás gombot (<button type="button" class="btn-close" ...>).

A modális tartalom részét egy <div class="modal-body"> elem képezi, ami egy <form> elemet tartalmaz az "editProfilePassword" azonosítóval. Ez a forma két jelszó beviteli mezőt tartalmaz, amelyeket "Jelszó" és "Jelszó újra" címkékkel jelöltek meg. Az első beviteli mezőnek van egy minta attribútuma, amely meghatározza a jelszó követelményeit (legalább egy számjegy, egy kisbetű, egy nagybetű, és legalább 8 karakter hosszú). Az üzeneteket megjelenítő részt egy <div id="message" class="mt-5"> elem képviseli. Itt több <p> elem található az "letter", "capital", "number" és "length" azonosítókkal, amelyek mindegyike egy-egy konkrét követelményt reprezentál (például egy kisbetűt tartalmaz, egy nagybetűt tartalmaz, egy számjegyet tartalmaz, és legalább 8 karakter hosszú). A modális ablak láblécét egy <div class="modal-footer"> elem tartalmazza, amely két gombot tartalmaz: Egy gomb a "btn-outline-secondary" osztállyal és a data-bs-dismiss="modal" adatattribútummal, ami bezárja a modális ablakot anélkül, hogy mentené a változtatásokat. Másik gomb a "btn-outline-success" osztállyal, amely akkor hívja meg az editPassword() függvényt, amikor rákattintanak. Ez a függvény valószínűleg felelős az új jelszó beküldéséért.

1. Felhasználói dokumentáció

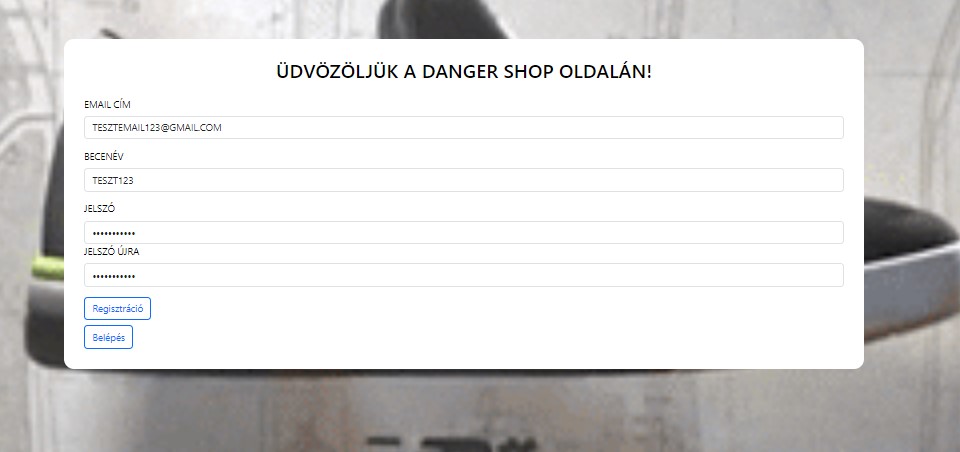


Az első adat, amit meg kell adni a regisztrációhoz az email. Azért fontos hogy email címét megadja, hogy, tájékoztassuk önt az esetleges akciókról vagy az oldal karbantartások idejéről.

A felhasználónév lehet tetszőleges. Sokan használják felhasználónévként például a saját becenevüket.

A harmadik a jelszó. Ez az adat a belépéshez szükséges, ezért ne felejtse el. A jelszót kétszer kell beírni, mert ha regisztrálásnál a jelszó kitöltésénél esetleg rossz jelszót gépel be, akkor nem tudna belépni az oldalra a későbbiekben. Amennyiben a két jelszó megadásánál megegyezik a jelszó, a regisztráció sikeres lesz és az esetleges későbbi belépés sem ütközik akadályokba.

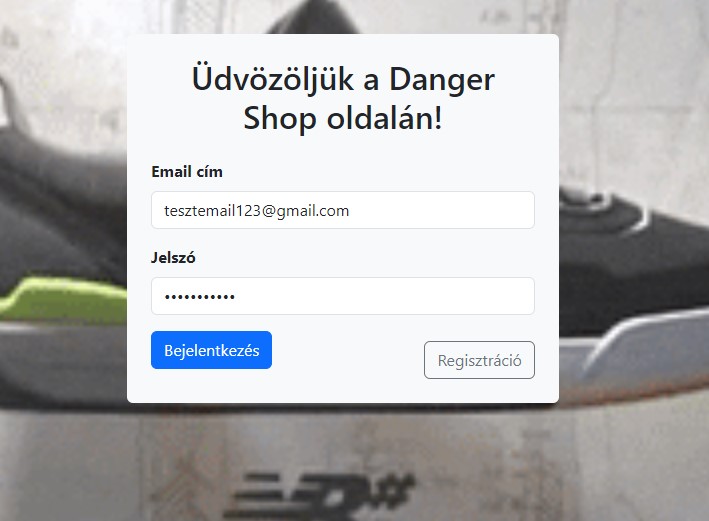
Miután minden mezőt megfelelően kitöltöttünk, kattintsunk a ’Regisztrálás’ gombra.



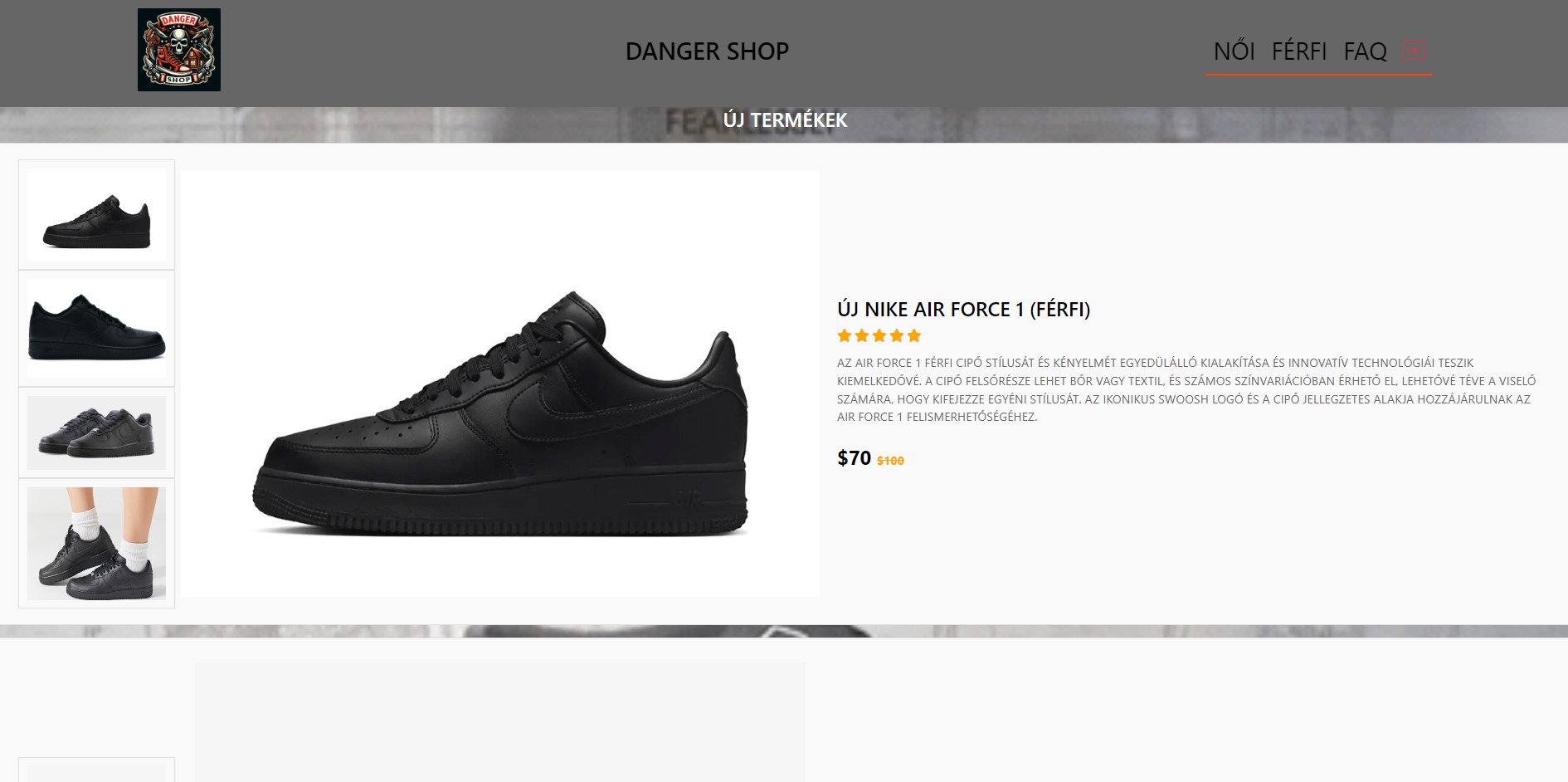
Bejelentkezés:

A bejelentkezésnél szükséges a felhasználónév és a jelszó.

Ezek megadása után a ’Bejelentkezés’ gombra kattintva történik a belépés.

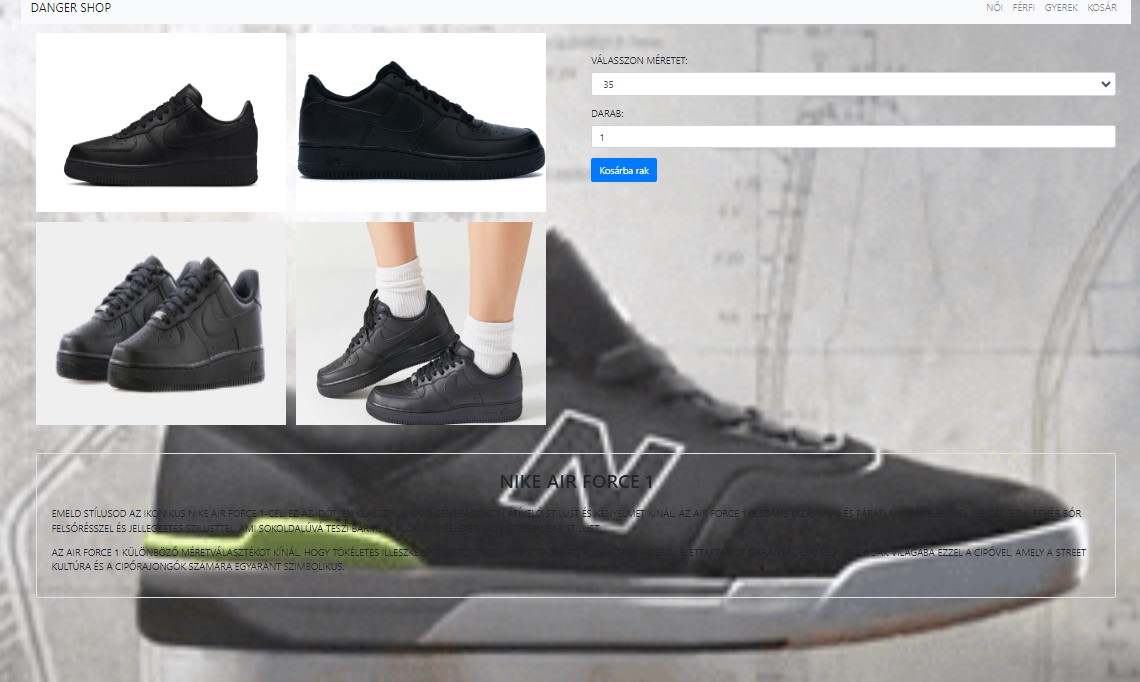


Így a következő oldal jelenik meg a képernyőnkön:



A rendeléshez a következőket kell megadni:

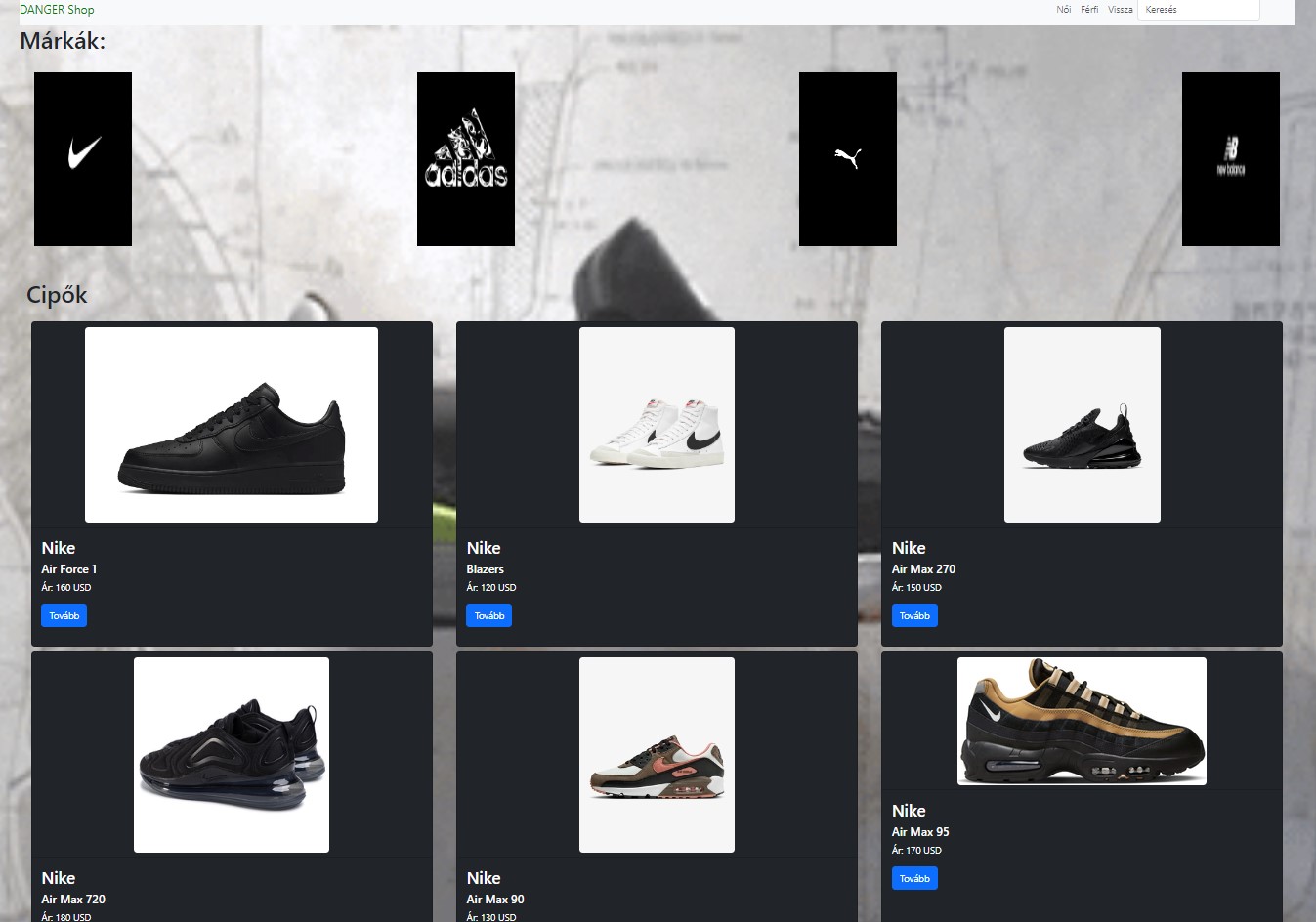
* Méret
* Darabszám



Miután kiválasztottuk a megfelelő méretet és darabszámot,utána kattintsunk a „Kosárba rak” gombra, aztán megjelenik a Kosár és itt elvégezhetjük a Fizetést.



Keresési menüpont:



Ezen az oldalon férfi cipők találhatóak meg ahol a jobb felső sarokban található egy „Keresés” menüpont amivel szabadon rákereshetünk az adott cipőre amelyiket éppen szeretnénk a Kosárba rakni.

Ezen kívül még találunk a fejlécben egy „Női” egy „Férfi” és egy „Vissza” gombot.

A „Vissza” gombbal a weboldal vissza irányít minket a Főoldalra.

A „Női” és a „Férfi” gombra kattintva pedig megjelenik az összes kívánt termék.