

SZENT ISTVÁN KATOLIKUS  
TECHNIKUM ÉS GIMNÁZIUM

**KAPITOR BALÁZS  
FRIDRIK ISTVÁN**

**RUHÁZATI WEBÁRUHÁZ**

**SZOFTVERFEJLESZTÉS ÉS -TESZTELÉS VIZSGAREMEK**

Sátoraljaújhely, 2024/25

Tartalomjegyzék

[1. Bevezetés 4](#_heading=h.3rdcrjn)

[2. Projekt jellemzése 5](#_heading=h.26in1rg)

[3. Használt csoportmunka eszközök 6](#_heading=h.lnxbz9)

[4. Felhasználói dokumentáció 7](#_heading=h.35nkun2)

[4.1. Minimális gépigény 7](#_heading=h.1ksv4uv)

[4.2. Ajánlott gépigény 8](#_heading=h.44sinio)

[4.3. Főoldal 9](#_heading=h.2jxsxqh)

[4.4. Ruhák oldalai 11](#_heading=h.z337ya)

[4.5. Regisztráció 14](#_heading=h.3j2qqm3)

[4.6. Bejelentkezés 15](#_heading=h.1y810tw)

[4.7. Elfelejtett jelszó 15](#_heading=h.4i7ojhp)

[4.8. Adományok oldal 16](#_heading=h.7b13w0hyohc1)

[4.9. Rendelések 16](#_heading=h.23ckvvd)

[4.10. Rendelés részletei 16](#_heading=h.ihv636)

[5. Fejlesztői dokumentáció 17](#_heading=h.32hioqz)

[5.1. Telepítés 17](#_heading=h.1hmsyys)

[A Forráskód Beszerzése 17](#_heading=h.28h4qwu)

[Az Adatbázis Telepítése (XAMPP) 17](#_heading=h.nmf14n)

[Lépések: 17](#_heading=h.37m2jsg)

[Backend Telepítése (Node.js) 19](#_heading=h.1mrcu09)

[Lépések: 19](#_heading=h.46r0co2)

[Frontend Telepítése (Angular 16) 20](#_heading=h.2lwamvv)

[Lépések: 20](#_heading=h.111kx3o)

[Telepítési Opciók 20](#_heading=h.3l18frh)

[Manuális Telepítés 20](#_heading=h.206ipza)

[5.2. Elindítás menete 22](#_heading=h.4k668n3)

[Megjegyzések: 23](#_heading=h.2zbgiuw)

[5.3. Backend 24](#_heading=h.1egqt2p)

[a. Fejlesztéshez használt technológiák, szoftverek, fejlesztői környezetek 24](#_heading=h.2dlolyb)

[b. Adatbázis felépítése 25](#_heading=h.sqyw64)

[c. Végpontok (táblázat) 29](#_heading=h.3cqmetx)

[d. Funkciók működésének, megvalósításának magyarázata 30](#_heading=h.1rvwp1q)

[e. Backend adatbázis kapcsolata; regisztráció, bejelentkezés, kijelentkezés működése; a funkciók megvalósításának bemutatása 33](#_heading=h.4bvk7pj)

[5.4. Frontend 35](#_heading=h.2w5ecyt)

[a. Elkészített komponensek, és működésük 35](#_heading=h.1baon6m)

[b. Routing működése, frontend végpontok felépítése 35](#_heading=h.3vac5uf)

[c. Backend adatbázis kapcsolata; regisztráció, bejelentkezés, kijelentkezés működése; a funkciók megvalósításának bemutatása 35](#_heading=h.2afmg28)

[6. Tesztelés 35](#_heading=h.pkwqa1)

[6.1. Backend 35](#_heading=h.1opuj5n)

[6.2. Frontend 35](#_heading=h.3mzq4wv)

[6.3. Terheléses teszt 35](#_heading=h.2250f4o)

[7. Továbbfejlesztési lehetőségek 36](#_heading=h.haapch)

[8. Irodalomjegyzék 37](#_heading=h.319y80a)

# Bevezetés

Az elmúlt években megnövekedtek a webáruházakon keresztül történő vásárlások száma, ezért a fő koncepciónk egy ruházati webáruház megalkotása, aminek a lényege, hogy a felhasználók azt a ruhát találják meg, amelyek tökéletesen illeszkedik stílusukhoz, egyéniségükhöz. Célunk egy olyan egyszerű, letisztult, modern webáruház megalkotása, ami segít a felhasználóknak, hogy megtalálhassák a számukra legmegfelelőbb ruhatárukat. Fontosnak tartjuk a felhasználók igényeit. Ennek érdekében a fejlesztés során elsősorban a felhasználók lehetséges igényeit, a rendszer egyszerű és egyértelmű kezelhetőségét vettük priotitásként. Ha valamilyen ruházati weboldalon veszünk termékeket, akkor a legtöbb esetben nem megfelelő a méretünkhöz, vagy nem passzol a stílusunkhoz. Ehhez segítségül szolgál a főoldalon elhelyezett mérettáblázat. Ezzel szeretnénk a rossz és helytelenül megválasztott termékek visszaküldésének mennyiségét csökkenteni. A kosárba adás opció csak a regisztrált felhasználók számára elérhető, de a termékek közötti böngészés a nem regisztrált felhasználók számára is biztosított. Így kalkulálhatóvá válik a leendő vásárlónak a költségei.

# Projekt jellemzése

A projekt célja egy modern, felhasználóbarát ruházati webáruház létrehozása, amely kifejezetten a vásárlók egyéni stílusának és igényeinek kiszolgálására fókuszál. A webáruház kialakításakor arra törekedtünk, hogy a funkcionalitás és az esztétikum tökéletes egyensúlyát teremtsük meg, miközben a felhasználói élményt helyezzük előtérbe. A rendszer lehetővé teszi a ruházati termékek széles kínálatának böngészését, szűrését és kiválasztását, különös figyelmet fordítva a méretpontosságra és a stílusos megjelenésre. A projekt ötlete az elmúlt években megfigyelhető trendekre épül, miszerint egyre többen vásárolnak online, ugyanakkor sokan szembesülnek azzal, hogy a kiválasztott termékek nem illeszkednek megfelelően méretükhöz vagy stílusukhoz. Ezt a problémát kívánjuk orvosolni egy átgondolt és könnyen használható platform megalkotásával.

A webáruház fejlesztése során egy olyan rendszert hoztunk létre, amely nemcsak a vásárlási folyamatot egyszerűsíti le, hanem a felhasználók számára átlátható és könnyen kezelhető felületet biztosít. A funkcionalitás egyik kiemelt eleme a főoldalon elhelyezett mérettáblázat, amely részletes információt nyújt a termékek méreteiről, ezzel segítve a vásárlókat a helyes méret kiválasztásában. Ez a funkció nemcsak a felhasználói elégedettséget növeli, hanem hozzájárul a visszaküldési arány csökkentéséhez is, ami mind a vásárlók, mind az üzemeltetők számára jelentős előnyt jelent. A mérettáblázat mellett a termékek részletes leírása és a szűrési lehetőségek is támogatják a döntéshozatalt, lehetővé téve, hogy a felhasználók gyorsan megtalálják a számukra leginkább megfelelő ruhadarabokat.

A rendszer regisztrációhoz kötött és a nem regisztrált felhasználók számára is eltérő funkcionalitást kínál. A regisztrált felhasználók számára biztosítjuk a kosárba helyezés lehetőségét, így ők a vásárlási folyamatot végig vihetik, míg a nem regisztrált látogatók szabadon böngészhetnek a kínálatban. Ez a megközelítés egyrészt ösztönzi a felhasználókat a regisztrációra, másrészt lehetővé teszi, hogy az érdeklődők előzetesen felmérjék a termékpalettát és kalkulálják költségeiket. A webáruházat úgy terveztük, hogy mobil- és asztali környezetben egyaránt optimális élményt nyújtson, figyelembe véve a mai vásárlók eltérő eszközeinek sokszínűségét. A projekt során kiemelt figyelmet fordítottunk a gyors betöltési időkre is, hogy a felhasználók bizalommal használhassák a platformot. Összességében a cél egy olyan webáruház létrehozása volt, amely praktikus, esztétikus és a vásárlók valós igényeire szabott megoldásokat kínál.

# Használt csoportmunka eszközök

#### Csoportmunka eszközök:

* **Trello**:
  + Feladatok rendszerezése (To Do, In Progress, Done)
  + Felelősök, határidők hozzárendelése
* **Git és GitHub**:
  + Verziókezelés, kód tárolása
  + Ágak, pull requestek, issue-k használata
* **Visual Studio Code**:
  + Kódírás, bővítmények (pl. GitLens, Prettier)
  + Beépített terminál, többnyelvű támogatás

**Programozási nyelvek és technológiák**

##### Frontend

* **Angular**:
  + Komponens-alapú keretrendszer, kétirányú adatkezelés
* **HTML**:
  + Struktúra, sablonok.
* **CSS**:
  + Stílus, reszponzivitás
* **TypeScript**:
  + Típusellenőrzés, hibacsökkentés, logika

##### Backend

* SQL:
  + Adatbázisok (pl. MySQL), lekérdezések
* **Node.js**:
  + Szerveroldali JavaScript, API-k (pl. Express.js)

# Felhasználói dokumentáció

## Minimális gépigény

A ruházati webáruház böngésző alapú alkalmazásként működik, így a gépigény a kliensoldali (felhasználói) követelményeket tükrözi. A rendszer telepítése és futtatása során az alábbi hardver- és szoftverkövetelmények szükségesek a megfelelő működéshez:

* **Processzor**: 1,5 GHz vagy gyorsabb, kétmagos processzor (pl. Intel Core i3 vagy hasonló teljesítményű AMD processzor).
* **Memória**: Minimum 4 GB RAM, hogy a böngésző és az alkalmazás stabilan fusson.
* **Tárhely**: Legalább 500 MB szabad tárhely a böngésző gyorsítótárához és a telepítési fájlokhoz.
* **Internetkapcsolat**: Stabil, legalább 5 Mbps sebességű internetkapcsolat a webáruház betöltéséhez és a képek gyors megjelenítéséhez.
* **Operációs rendszer**: Window0s 10 (64-bit), macOS 10.14 (Mojave) vagy újabb, illetve Ubuntu 18.04 vagy újabb Linux disztribúció.
* **Böngésző**: Google Chrome 90, Mozilla Firefox 88, Microsoft Edge 90 vagy Safari 14 (vagy ezek újabb verziói).
* **Kijelző**: Minimum 1024x768 felbontás, hogy a weboldal elemei torzítás nélkül megjelenjenek.
* **További szoftverek**: Aktuális böngésző frissítések és JavaScript engedélyezése a böngészőben.

Ezek a követelmények biztosítják, hogy a webáruház alapvető funkciói – mint a termékek böngészése, a mérettáblázat megtekintése és a regisztráció – zökkenőmentesen működjenek alacsonyabb teljesítményű eszközökön is.

## 

## Ajánlott gépigény

A ruházati webáruház böngésző alapú alkalmazásként működik, így az ajánlott gépigény a kliensoldali (felhasználói) követelményekre fókuszál, amelyek magasabb teljesítményt biztosítanak az optimális felhasználói élmény érdekében. A rendszer használata során az alábbi hardver- és szoftverkövetelmények javasoltak a webáruház teljes funkcionalitásának zökkenőmentes kihasználásához:

* **Processzor**: 2,5 GHz vagy gyorsabb, négymagos processzor (pl. Intel Core i5/i7 vagy ekvivalens AMD Ryzen processzor).
* **Memória**: 8 GB RAM vagy több, hogy a böngésző és az esetleges háttéralkalmazások párhuzamos futtatása ne okozzon lassulást.
* **Tárhely**: Minimum 1 GB szabad tárhely a gyorsítótárhoz, képekhez és a helyi fejlesztési környezet fájljaihoz (ha releváns).
* **Internetkapcsolat**: Stabil, legalább 20 Mbps sebességű internetkapcsolat a gyors betöltési idők és a zökkenőmentes navigáció érdekében.
* **Operációs rendszer**: Windows 11 (64-bit), macOS 12 (Monterey) vagy újabb, illetve Ubuntu 20.04 vagy újabb Linux disztribúció.
* **Böngésző**: Google Chrome 100, Mozilla Firefox 95, Microsoft Edge 100 vagy Safari 15 (vagy ezek újabb verziói).
* **Kijelző**: 1920x1080 (Full HD) felbontás vagy magasabb, a modern, letisztult dizájn és a képek optimális megjelenítéséhez.
* **További szoftverek**: Böngészőfrissítések, WebGL támogatás (a grafikai elemekhez), valamint opcionálisan Node.js 16.x vagy újabb (ha a felhasználó helyi szervert futtat teszteléshez).

Ez a konfiguráció lehetővé teszi a webáruház teljes funkcionalitásának kihasználását, beleértve a gyors oldalletöltést, a szűrési opciók hatékony használatát és a mobilbarát felület zökkenőmentes működését asztali és hordozható eszközökön egyaránt.

## Főoldal

A főoldal a DivatPalota webáruház kezdőképernyője, ahol mindent megtalál egy helyen: ruhákat, akciókat és hasznos infókat. Az oldal tetején egy menüsor van, amellyel gyorsan eljuthat a webshop többi részére. Ez a menüsor mindegyik oldal tetején megtalálható. Itt vannak például a „Kezdőlap”, a „Ruhák” (ahol egy legördülő listában pólók, pulóverek, ingek, kabátok és nadrágok közül választhat), valamint az „Adományozás” gombok. Ha még nem lépett be, a menüsor jobb oldalán a „Regisztráció” és „Bejelentkezés” gombokat látja. Ezekkel létrehozhat egy saját fiókot, vagy beléphet, ha már rendelkezik eggyel.

Amikor bejelentkezik, a menüsor jobb oldala megváltozik: megjelenik a „Kosár” és a „Fiókom” gomb. A „Kosár” mellett zárójelben látszik, hány termék van benne (például „Kosár (3 db)”). Ha az egérrel a „Kosár” fölé megy vagy kattint, egy lista ugrik fel, amelyben látja, miket tett a kosárba. Minden terméknél ott a neve, leírása, ára és hogy hány darabot választott. Ezeket szabadon változtathatja: a „+” gombbal többet adhat hozzá, a „−” gombbal kevesebbet, a szemetes ikonnal pedig kitörölheti az adott terméket. Ha mindent ki akar dobni, az „Összes törlése” gombra kattinthat, ha pedig fizetni szeretne, a „Fizetés” gombot válassza. A lista alján láthatja, összesen mennyit kell fizetnie forintban. A „Fiókom” gombra kattintva egy másik legördülő lista nyílik, ahol a „Rendelések” (korábbi vásárlások) és a „Kijelentkezés” lehetőségek vannak.

A főoldal négy nagy részre van felosztva, hogy könnyen eligazodjon:

1. **Bal felső rész – Üdvözlés**: Itt egy rövid köszöntőt találhat. Megtudhatja, hogy a DivatPalota mindig a legújabb divatot kínálja, és ha kérdése van, a csapat szívesen segít. Alatta egy kép van, ami mutatja, hogy minden évszakra vannak nálunk ruhák.
2. **Jobb felső rész – Infók a ruhákról**: Ebben a részben olvashat ruháinkról. Azt javasoljuk, hogy nézzen vissza gyakran, mert minden nap új akciók vannak. Itt vannak a mérettáblázatok is, amik segítenek kiválasztani a megfelelő méretet. Két képre kattinthat: az egyik a férfiaknak, a másik a nőknek szóló táblázat, és mindkettő új ablakban nyílik meg.
3. **Bal alsó rész – Évszakos ajánlatok**: Ez a rész az aktuális divatról szól. Mindegy, hogy tavasz, nyár, ősz vagy tél van, nálunk megtalálhatja a legjobb ruhákat. Azt javasoljuk, hogy válogasson kedvére, és ha elakad, kérjen tőlünk segítséget. Itt egy diavetítés (slideshow) is van, ami szép képeken mutatja meg a kiemelt darabokat.
4. **Jobb alsó rész – Adományozás**: Itt arról olvashat, hogyan segíthet másokon. Három lehetőség van: adhat pénzt (minden kis összeg számít), küldhet ruhát vagy játékot, vagy jelentkezhet önkéntesnek a programjainkra. Egy kép is van itt, ami az adományozás hangulatát mutatja.

Ha legörget az oldal aljára, megtalálja az elérhetőségeinket, például az e-mail címünket és telefonszámunkat, hogy felvehesse velünk a kapcsolatot. Ez a rész minden oldalon látható.

## Ruhák oldalai

A webáruházban jelenleg öt különböző ruhakategóriából válogathat: pólók, pulóverek, nadrágok, kabátok és ingek. Ebben a részben a „Pólók” kategória oldalának működését mutatjuk be részletesen, amely példaként szolgál a többi kategória hasonló felépítésére is.

A „Pólók” oldalon egyszerre 20 termék látható, amelyek két oszlopban helyezkednek el. Minden pólóhoz tartozik egy mintakép, amely alatt a következő információk találhatók:

**Márka:** A póló gyártójának neve.

**Leírás:** Rövid szöveg a termék jellemzőiről.

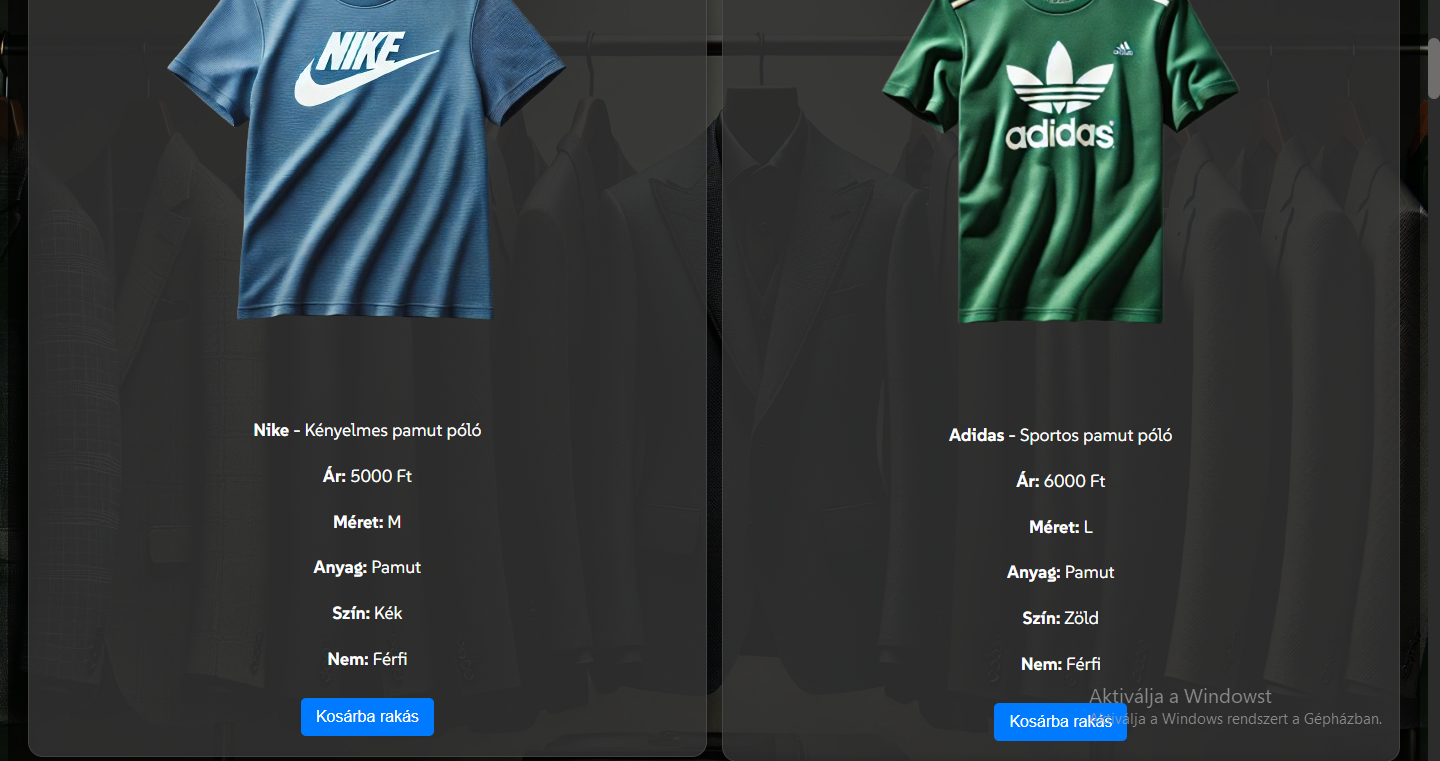
**Ár:** A póló ára forintban.

**Méret:** A rendelkezésre álló méret (pl. S, M, L, XL).

**Anyag:** A póló anyaga (pl. pamut, poliészter).

**Szín:** A póló színe.

**Nem:** Kinek ajánlott a póló (pl. férfi, női).



A termékek alatt található egy „Kosárba rakás” gomb, amelynek segítségével a kiválasztott pólót a kosarába helyezheti. Amennyiben Ön még nem jelentkezett be, akkor a gomb helyén „A vásárláshoz jelentkezzen be!” üzenet jelenik meg.  
Az oldal tetején egy szűrőpanel található, amely segít abban, hogy a pólókat az Ön igényeinek megfelelően szűkítse. A következő szűrési szempontok állnak rendelkezésre:

**Márka:** Válasszon egy konkrét márkát, vagy hagyja üresen az „Összes” opcióval, ha nem kíván márka szerint szűrni.

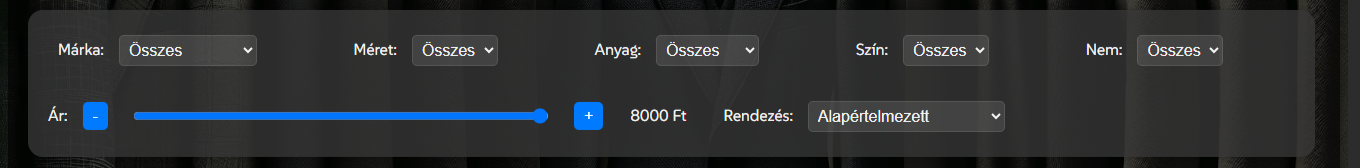
**Méret:** Szűrhet méret szerint (pl. S, M, L), vagy választhatja az „Összes” lehetőséget.

**Anyag:** Kiválaszthatja a póló anyagát (pl. pamut, poliészter), vagy az „Összes” opcióval minden anyagot megjeleníthet.

**Szín:** Szűrhet szín szerint (pl. fekete, fehér, kék), vagy az „Összes” opciónál maradhat.

**Nem:** Kiválaszthatja, hogy férfi vagy női pólókat szeretne látni, illetve az „Összes” opcióval mindet megjelenítheti.

**Ár:** Egy csúszka segítségével beállíthatja a maximum árat, amelyet hajlandó fizetni. A csúszka mellett „-” és „+” gombok találhatók, amelyekkel finomhangolhatja az értéket. Ha hosszan nyomva tartja a gombot, az ár folyamatosan csökken vagy nő, amíg el nem engedi.



A szűrők megváltoztatásakor a pólók listája automatikusan frissül, hogy csak a feltételeknek megfelelő termékek jelenjenek meg.

**Rendezési lehetőségek**

A szűrők mellett egy „Rendezés” legördülő menü található, amellyel a pólók megjelenési sorrendjét állíthatja be. Az alábbi opciók közül választhat:

**Alapértelmezett:** A pólók az eredeti sorrendben jelennek meg.

**ABC sorrend növekvő:** A márkák neve alapján ábécésorrendben, A-tól Z-ig.

**ABC sorrend csökkenő:** A márkák neve alapján fordított ábécésorrendben, Z-től A-ig.

**Ár szerint növekvő:** Az árak alapján a legolcsóbbtól a legdrágábbig.

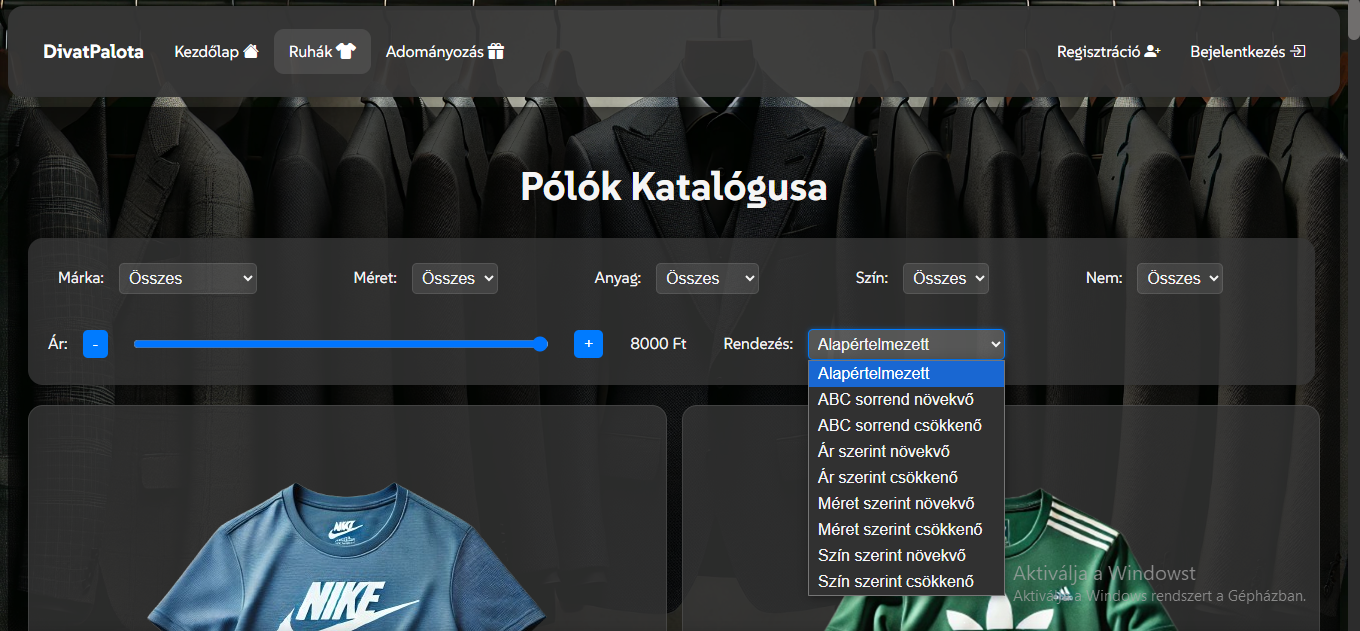
**Ár szerint csökkenő:** Az árak alapján a legdrágábbtól a legolcsóbbig.

**Méret szerint növekvő:** A méretek alapján növekvő sorrendben (pl. S, M, L).

**Méret szerint csökkenő:** A méretek alapján csökkenő sorrendben (pl. XL, L, M).

**Szín szerint növekvő:** A színek neve alapján ábécésorrendben.

**Szín szerint csökkenő:** A színek neve alapján fordított ábécésorrendben.



A rendezés kiválasztása után a lista azonnal az Ön által megadott szempont szerint frissül.

**Kosárba helyezés**

Ha egy pólót szeretne megvásárolni, kattintson a „Kosárba rakás” gombra. Ehhez azonban be kell jelentkeznie a fiókjába. Ha még nem tette meg, a rendszer figyelmeztető üzenetet jelenít meg: „Kérjük, jelentkezzen be a vásárláshoz!”. Bejelentkezés után a gomb újra aktívvá válik, és a póló a kosarába kerül. Kosárba adás után folytathatja a vásárlást, illetve további termékeket helyezhet a kosarába.

**Ha nincs találat**

Amennyiben a szűrési feltételek alapján egyetlen póló sem felel meg, az oldalon a „Nincsenek elérhető pólók!” üzenet jelenik meg. Ebben az esetben próbálja meg módosítani a szűrési feltételeket (pl. állítsa az „Összes” opciót egyes szűrőknél, vagy növelje az ár tartományt).

## Regisztráció

A regisztráció webáruházunk egyik legfontosabb funkciója, amely lehetővé teszi, hogy saját fiókot hozzon létre, és teljes körűen kihasználhassa az oldal kínálta lehetőségeket. Az oldal a „Regisztráció” gombra kattintva érhető el a főoldal menüsorából, ha még nem jelentkezett be. Itt egy egyszerű, de átlátható űrlapot talál, amelyet kitöltve gyorsan és biztonságosan létrehozhatja fiókját. A regisztrációhoz szükséges összes mező jól láthatóan jelölve van, a kötelező adatok mellett pedig egyértelmű útmutatásokat kap a kitöltéshez.

Az űrlap tetején a „Regisztráció” cím fogadja, jelezve, hogy jó helyen jár. Az űrlap több mezőből áll, amelyeket sorban kitölthet: vezetéknév, keresztnév, felhasználónév, jelszó (kétszer, az ellenőrzés miatt), születési dátum, e-mail cím, telefonszám, lakcím, nem, valamint egy biztonsági kérdés és válasz páros. Minden kötelező mezőt egy piros csillag (\*) jelöl, hogy ne maradjon ki semmi fontos. A jelszó mezőnél egy kis információs ikon (ℹ️) segít: ha az egeret fölé viszi, egy felugró ablakban láthatja, hogy a jelszónak legalább 8 karakterből kell állnia, és tartalmaznia kell kis- és nagybetűket, számot, valamint speciális karaktert (pl. @$!%\*?&). Ez biztosítja, hogy fiókja biztonságban legyen.

Miután kitöltötte az adatokat, a „Regisztrálás” gombra kattintva küldheti el az űrlapot. Ha valami nem stimmel – például üresen hagy egy kötelező mezőt, a jelszavak nem egyeznek, vagy az e-mail formátuma hibás –, egy hibaüzenet jelenik meg az űrlap tetején, például: „A \*-gal jelölt mezők kitöltése kötelező!” vagy „A jelszavak nem egyeznek!”. A rendszer emellett egy felugró ablakban is figyelmezteti a hibára, amely 3 másodperc után eltűnik. Ha minden rendben van, a gomb „Regisztrálás…” feliratra vált, jelezve, hogy a folyamat zajlik, majd sikeres regisztráció esetén egy üzenet tudatja: „Sikeres regisztráció! Most már bejelentkezhet.” Ezután átirányítjuk a bejelentkezési oldalra.

Az űrlap alján egy hasznos link is található: „Már van fiókod? Jelentkezz be!” – ha már regisztrált korábban, innen egy kattintással átléphet a bejelentkezéshez. A regisztráció célja, hogy Ön könnyedén vásárolhasson, nyomon követhesse rendeléseit, és részt vehessen az adományozási programjainkban. Ha kérdése van, az oldal alján – ahogy mindenhol a webáruházban – megtalálja elérhetőségeinket: e-mail címünket és telefonszámunkat, hogy segíthessünk. A regisztráció tehát gyors, egyszerű és biztonságos módja annak, hogy a DivatPalota közösségének tagjává váljon!

## Bejelentkezés

A bejelentkezés a webáruház kapuja, amelyen keresztül gyorsan és egyszerűen beléphet fiókjába, hogy vásárolhasson, kezelhesse rendeléseit, vagy részt vehessen programjainkban. Az oldal a menüsorban található „Bejelentkezés” gombra kattintva érhető el, ha még nem lépett be. Itt egy letisztult, felhasználóbarát űrlapot talál, amely mindössze két mező kitöltését kéri, így pillanatok alatt hozzáférhet a webáruház teljes funkcióihoz.

Az űrlap tetején a „Bejelentkezés” cím üdvözli, egyértelművé téve, hogy itt jár. Két fő mezőt kell kitöltenie: az első a „Felhasználónév vagy E-mail”, ahol megadhatja regisztrált felhasználónevét vagy e-mail címét, a második pedig a „Jelszó”, ahová a fiókjához tartozó jelszót írhatja be. Mindkét mező mellett egy piros csillag (\*) jelzi, hogy ezek kitöltése kötelező. Az űrlap alján egy kék „Bejelentkezés” gomb várja, hogy elküldje adatait. Ha a rendszer éppen dolgozik az Ön kérésén, a gomb felirata „Bejelentkezés…”-re vált, jelezve, hogy a folyamat zajlik.

Ha valami hibázik – például üresen hagy egy mezőt, vagy rossz jelszót ad meg –, egy hibaüzenet jelenik meg az űrlap tetején, például: „A \*-gal jelölt mezők kitöltése kötelező!” vagy „Rossz jelszó! Próbáld újra.”. Emellett egy felugró ablak is figyelmezteti a problémára, amely 3 másodperc után eltűnik. Sikeres bejelentkezés esetén egy vidám üzenet köszönti: „Sikeres bejelentkezés! Üdv újra! 🎉”, majd a rendszer átirányítja a főoldalra, ahol már a bejelentkezett felhasználóként folytathatja a böngészést.

Az űrlap alatt két hasznos linket talál: az „Elfelejtetted a jelszavadat? Kattints ide” opcióval jelszó-helyreállítási oldalra léphet, ha elfelejtette belépési adatait, míg a „Még nincs fiókod? Regisztrálj itt” link a regisztrációs oldalra viszi, ha új felhasználóként szeretne csatlakozni. Az oldal alján – ahogy a webáruház minden részén – elérhetőségeink (e-mail és telefonszám) is ott vannak, ha segítségre lenne szüksége. A bejelentkezés így gyors, biztonságos és kényelmes módja annak, hogy újra a DivatPalota világában találja magát!

## Elfelejtett jelszó

Az „Elfelejtett Jelszó” oldal a DivatPalota webáruház mentőöve azok számára, akik elfelejtették a jelszavukat, és szeretnék visszanyerni hozzáférésüket fiókjukhoz. Ezt az oldalt a bejelentkezési felületről, az „Elfelejtetted a jelszavadat? Kattints ide” linkre kattintva érheti el. Az oldal kétlépcsős folyamaton keresztül segít: először ellenőrzi személyazonosságát, majd lehetővé teszi az új jelszó beállítását, így gyorsan újra használatba veheti fiókját.

Az oldal tetején az „Elfelejtett Jelszó” cím fogadja, jelezve, hogy itt jár. Kezdetben egy űrlapot lát, amely három mezőt tartalmaz: „Felhasználónév vagy E-mail cím”, „Biztonsági kérdés” és „Válasz”. Mindhárom mezőt ki kell töltenie – ezt piros csillagok (\*) jelzik –, hiszen ezek alapján azonosítjuk Önt. A regisztrációkor megadott adatokat kell itt használnia: adja meg felhasználónevét vagy e-mail címét, majd a biztonsági kérdést és a hozzá tartozó választ. Miután mindent kitöltött, a „Jelszó Visszaállítása” gombra kattintva küldheti el az adatokat. Ha minden stimmel, egy üzenet jelenik meg: „Írd be az új jelszavadat!”, és az űrlap átvált az új jelszó megadására.

A második lépésben két új mező jelenik meg: „Új Jelszó” és „Új Jelszó Megerősítése”. Az „Új Jelszó” mellett egy információs ikon (ℹ️) található, amelyre az egérrel rámutatva egy felugró ablakban láthatja a követelményeket: a jelszónak legalább 8 karakterből kell állnia, kis- és nagybetűket, számot, valamint speciális karaktert (pl. @$!%\*?&) kell tartalmaznia. Ha a két jelszó megegyezik, és megfelel a feltételeknek, a „Jelszó Módosítása” gombra kattintva mentheti az új jelszót. Sikeres módosítás esetén egy üzenet jelenik meg: „A jelszó sikeresen módosítva!”, majd átirányítjuk a bejelentkezési oldalra.

Ha valami nem stimmel – például hibás adatokat adott meg, vagy a jelszavak nem egyeznek –, egy hibaüzenet jelenik meg az űrlap tetején, például: „A \*-gal jelölt mezők kitöltése kötelező!” vagy „A jelszavak nem egyeznek!”. Ezt egy felugró ablak is megerősíti, amely 3 másodperc után eltűnik. Az oldal alján egy „Vissza a bejelentkezéshez” link segít, ha inkább visszatérne a bejelentkezési felületre. Elérhetőségeink – e-mail és telefonszám – itt is megtalálhatók az oldal alján, ha további segítségre lenne szüksége. Ez az oldal gyors és biztonságos megoldást kínál jelszó problémáira!

## Adományok oldal

## Rendelések

## Rendelés részletei

# Fejlesztői dokumentáció

## Telepítés

A telepítés megkezdése előtt győződjön meg róla, hogy a következő szoftverek telepítve vannak a számítógépén:

1. **Node.js és npm**: A backend és frontend futtatásához szükséges (javasolt verzió: LTS, pl. 18.x).
2. **Angular CLI**: A frontend fejlesztéséhez és futtatásához (verzió: 16.x).
3. **XAMPP**: Az adatbázis (MySQL) futtatásához.
4. **Git**: A forráskód letöltéséhez (opcionális).
5. **Böngésző**: A webáruház teszteléséhez (pl. Chrome, Firefox).

## A Forráskód Beszerzése

Jelentkezzen be a GitHub fiókjába, és klónozza a repository-kat az alábbi parancsokkal:

git clone https://github.com/kapitorbalazs1/webaruhaz.git -b webaruhaz\_frontend

git clone https://github.com/kapitorbalazs1/webaruhaz.git -b webaruhaz\_backend

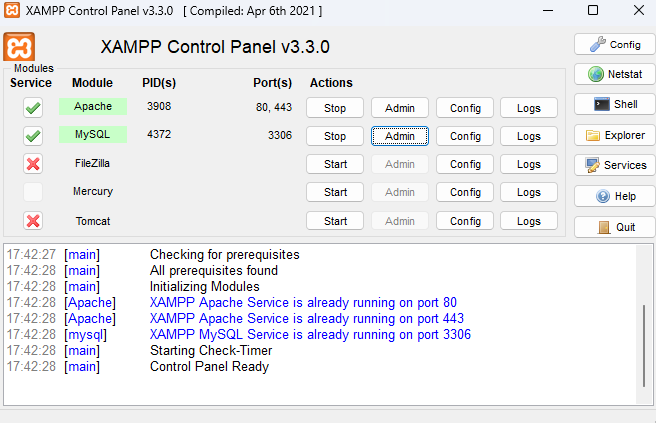
Ha nincs Git, töltse le a ZIP fájlokat, és bontsa ki őket külön mappákba (pl. webaruhaz\_frontend és webaruhaz\_backend).

## Az Adatbázis Telepítése (XAMPP)

A webáruház adatbázisa egy SQL fájlban található, amelyet XAMPP segítségével kell beállítani.

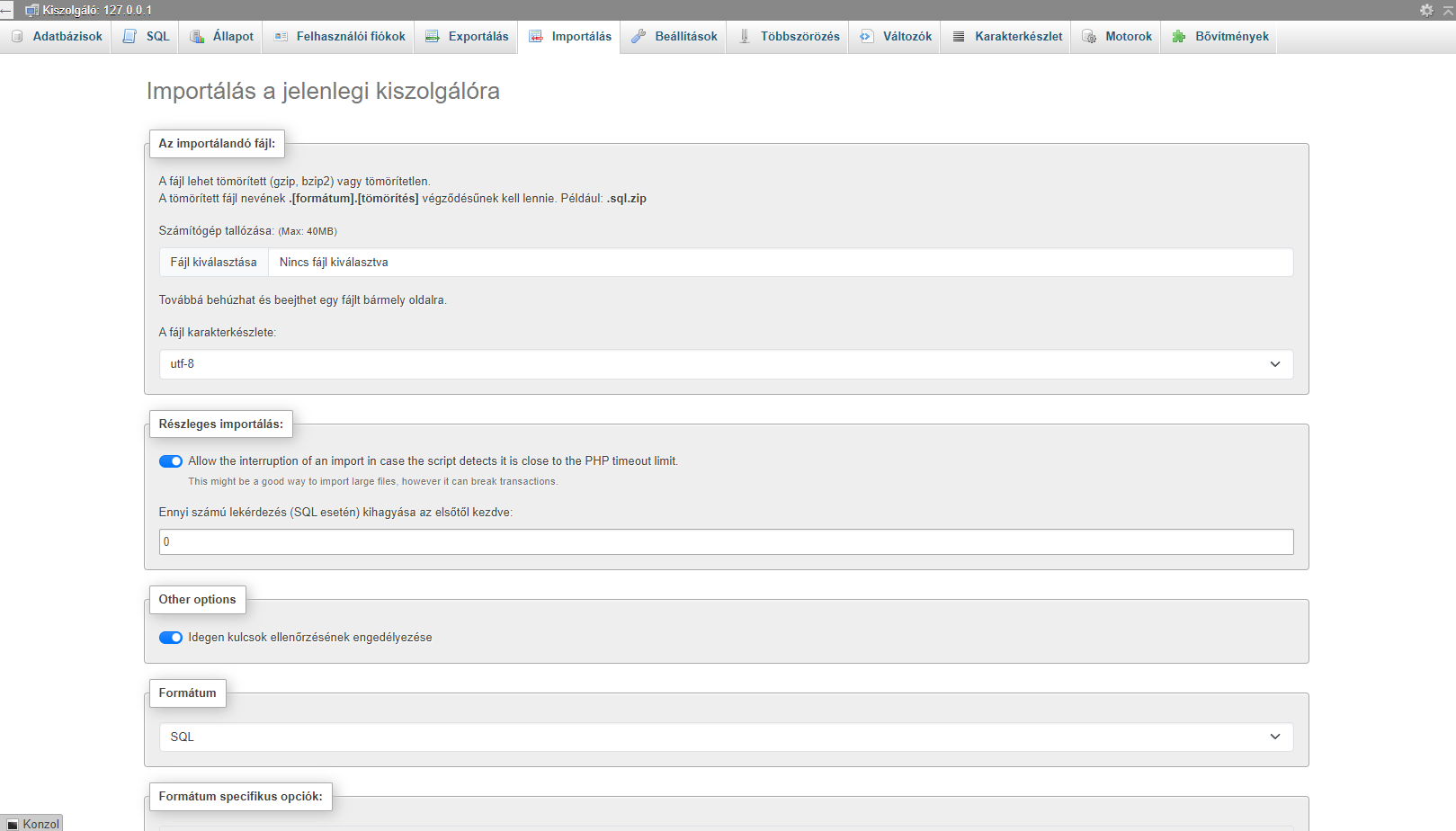
### Lépések:

1. **XAMPP indítása**:
   * Telepítse és indítsa el a XAMPP-ot. Az irányítópulton indítsa el az **Apache** és **MySQL** modulokat.



* + *(Ábra: XAMPP vezérlőpult, Apache és MySQL fut)*

1. **Adatbázis létrehozása**:
   * Nyissa meg a böngészőt, és írja be: http://localhost/phpmyadmin.
   * Kattintson az "Új" (New) gombra, és hozzon létre egy adatbázist, pl. “webaruhaz”.
2. **SQL fájl importálása**:
   * Az "Import" fülön válassza ki a repository-ban található .sql fájlt (pl. webaruhaz.sql).
   * Kattintson az "Importálás" gombra az adatbázis feltöltéséhez.



* + *(Ábra: phpMyAdmin importálás képernyő)*

## Backend Telepítése (Node.js)

A backend Node.js alapú, így a következő lépéseket kell követni:

### Lépések:

1. **Navigálás a backend mappába**:
   * Nyisson egy terminált, és lépjen a backend mappába:

cd webaruhaz\_backend

1. **Függőségek telepítése**:
   * Futtassa az alábbi parancsot a szükséges csomagok telepítéséhez:

npm install

1. **Konfiguráció**:
   * Ellenőrizze, hogy van-e .env fájl a mappában. Ha igen, szerkessze az adatbázis-kapcsolati adatokat (pl. host, user, password, database). Példa:

DB\_HOST=localhost  
DB\_USER=root  
DB\_PASSWORD=""  
DB\_NAME=webaruhaz  
PORT=3000

1. **Backend indítása**:
   * Indítsa el a szervert:

npm start

* + Ha minden rendben, a terminálban megjelenik, hogy a szerver fut (pl. "A szerver elindult a 3000-es porton!").

## Frontend Telepítése (Angular 16)

A frontend Angular 16 alapú, és a következő lépésekkel telepíthető:

### Lépések:

**Navigálás a frontend mappába**:

Nyisson egy új terminált, és lépjen a frontend mappába:

cd webaruhaz\_frontend

**Függőségek telepítése**:

Telepítse a szükséges csomagokat:

npm install

## Telepítési Opciók

* **Backend port**: Ha a 3000-es port foglalt, módosítsa a .env fájlban (pl. PORT=3001), és frissítse az environment.ts-ben is.
* **Adatbázis jelszó**: Ha a XAMPP MySQL-nek jelszava van, állítsa be a .env fájlban.
* **Production mód**: Éles környezetben az Angular build parancsát használja (ng build --prod), és a generált fájlokat másolja egy webszerverre.

## Manuális Telepítés

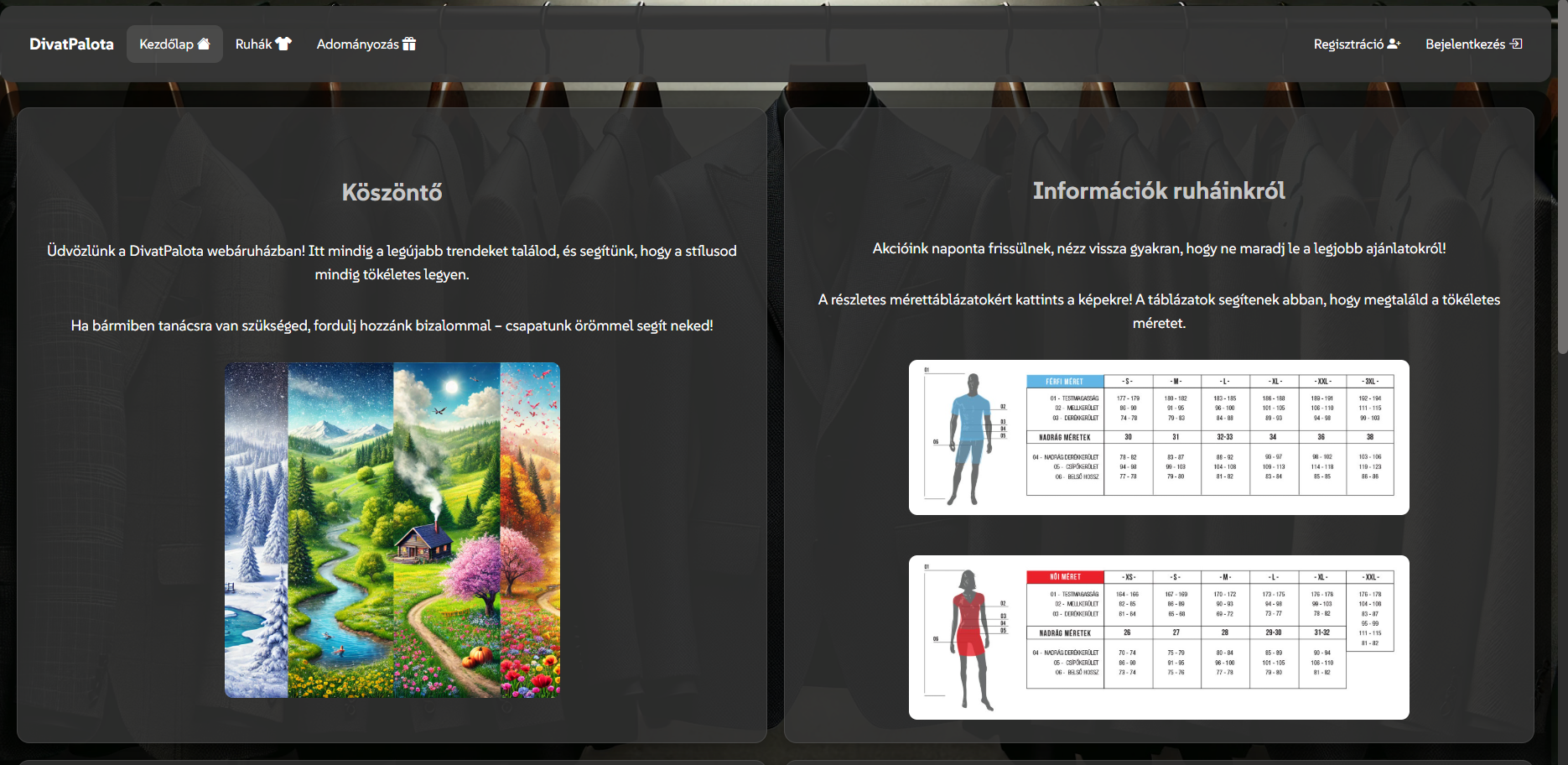
Ha a repository nem tartalmaz automatikus telepítőt, kövesse az alábbiakat:

1. **Fájlok másolása**:
   * Backend: Másolja a webaruhaz\_backend mappát egy szerverre, és futtassa az npm install és npm start parancsokat.
   * Frontend: Buildelje az Angular projektet (ng build), és a generált dist/ mappát másolja egy webszerver gyökerébe (pl. XAMPP htdocs mappájába).
2. **Adatbázis**: Importálja az .sql fájlt manuálisan a phpMyAdmin-on keresztül.
3. **Futtatás**: Győződjön meg róla, hogy a backend és az adatbázis fut, majd nyissa meg a frontend URL-jét.

## Elindítás menete

Miután a telepítés sikeresen megtörtént, a webáruház elindításához kövesse az alábbi lépéseket:

1. **XAMPP elindítása**:
   * Nyissa meg a XAMPP vezérlőpultot.
   * Indítsa el az **Apache** és **MySQL** modulokat. Ellenőrizze, hogy mindkettő zöld jelzést mutat és fut.
   * Ha szükséges, győződjön meg róla, hogy az adatbázis (webaruhaz) elérhető a http://localhost/phpmyadmin címen.
2. **Backend futtatása**:
   * Nyisson egy terminált (pl. Command Prompt, PowerShell vagy Bash).
   * Navigáljon a backend mappába: cd webaruhaz\_backend
   * Indítsa el a szervert az alábbi paranccsal: npm start
   * Ha minden rendben van, a terminálban megjelenik egy üzenet, például: *"A szerver elindult a 3000-es porton!"*. Ha a port eltérő (pl. a .env fájlban módosította), az ott megadott porton fut.
3. **Frontend futtatása**:
   * Nyisson egy új terminált (hagyja futni a backendet az előzőben).
   * Navigáljon a frontend mappába: cd webaruhaz\_frontend
   * Indítsa el az Angular fejlesztői szervert: ng serve
   * Az Angular alapértelmezett portja a 4200. Ha a terminálban megjelenik, hogy a projekt fordítása sikeres, és a szerver fut, akkor a frontend elérhető lesz.
4. **Webáruház ellenőrzése**:
   * Nyisson egy böngészőt (pl. Chrome, Firefox).
   * Írja be a címsorba: http://localhost:4200.
   * Ha a webáruház kezdőoldala betölt, az indítás sikeres.



1. *(Ábra: Webáruház kezdőoldala a böngészőben)*

## Megjegyzések:

* **Portütközés esetén**: Ha a 3000-es (backend) vagy 4200-as (frontend) port foglalt, módosítsa a .env fájlban a PORT értékét a backendhez (pl. PORT=3001), és az Angular esetében használja az ng serve --port 4201 parancsot.
* **Háttérben futás**: A fenti lépések fejlesztői környezetre vonatkoznak. Éles környezetben a frontendet buildelni kell (ng build --prod), és a generált fájlokat egy webszerverre (pl. XAMPP htdocs mappájába) kell másolni, míg a backendet külön szerveren kell futtatni.
* **Folyamatos működés**: Ha a terminált bezárja, a szerverek leállnak. Éles környezetben érdemes egy folyamatkezelőt (pl. PM2) használni a backendhez.

## Backend

### Fejlesztéshez használt technológiák, szoftverek, fejlesztői környezetek

**Alkalmazott fejlesztői eszközök**

A webáruház backend részének fejlesztéséhez és dokumentálásához az alábbi szoftvereket és technológiákat használtam fel:

* **Programozási nyelv**: JavaScript (Node.js környezetben futtatva)
* **Fejlesztői környezet**: Visual Studio Code – Könnyű, testre szabható szövegszerkesztő
* **Adatbázis-kezelő rendszer**: MariaDB 10.4.32 – Nyílt forráskódú relációs adatbázis, amely a MySQL alapjaira épül.
* **Adatbázis adminisztrációs eszköz**: phpMyAdmin 5.2.1 – Webes felület az adatbázis kezelésére, amely lehetővé tette az SQL dumpok készítését és az adatbázis szerkezetének módosítását.
* **Backend keretrendszer**: Express.js – Minimalista és rugalmas Node.js keretrendszer.
* **Adatbázis ORM**: Sequelize – Objektum-relációs leképező eszköz JavaScripthez.
* **Külső modulok**:
  + bcrypt (10-es verzió) – Jelszavak titkosítására használt könyvtár, forrás: [npmjs.com/package/bcrypt](https://www.npmjs.com/package/bcrypt), MIT licenc alatt.
  + cors – Cross-Origin Resource Sharing támogatására, forrás: [npmjs.com/package/cors](https://www.npmjs.com/package/cors), MIT licenc.
  + dotenv – Környezeti változók kezelésére, forrás: [npmjs.com/package/dotenv](https://www.npmjs.com/package/dotenv), BSD-2-Clause licenc.
  + puppeteer – PDF generáláshoz Chromium böngésző vezérlésére, forrás: [npmjs.com/package/puppeteer](https://www.npmjs.com/package/puppeteer), Apache-2.0 licenc.
  + moment-timezone – Időzónák kezelésére, forrás: [npmjs.com/package/moment-timezone](https://www.npmjs.com/package/moment-timezone), MIT licenc.
* **Verziókezelés**: Git – A forráskód változásainak követésére és tárolására.
* **Operációs rendszer**: Windows 10 – A fejlesztés és tesztelés platformja.

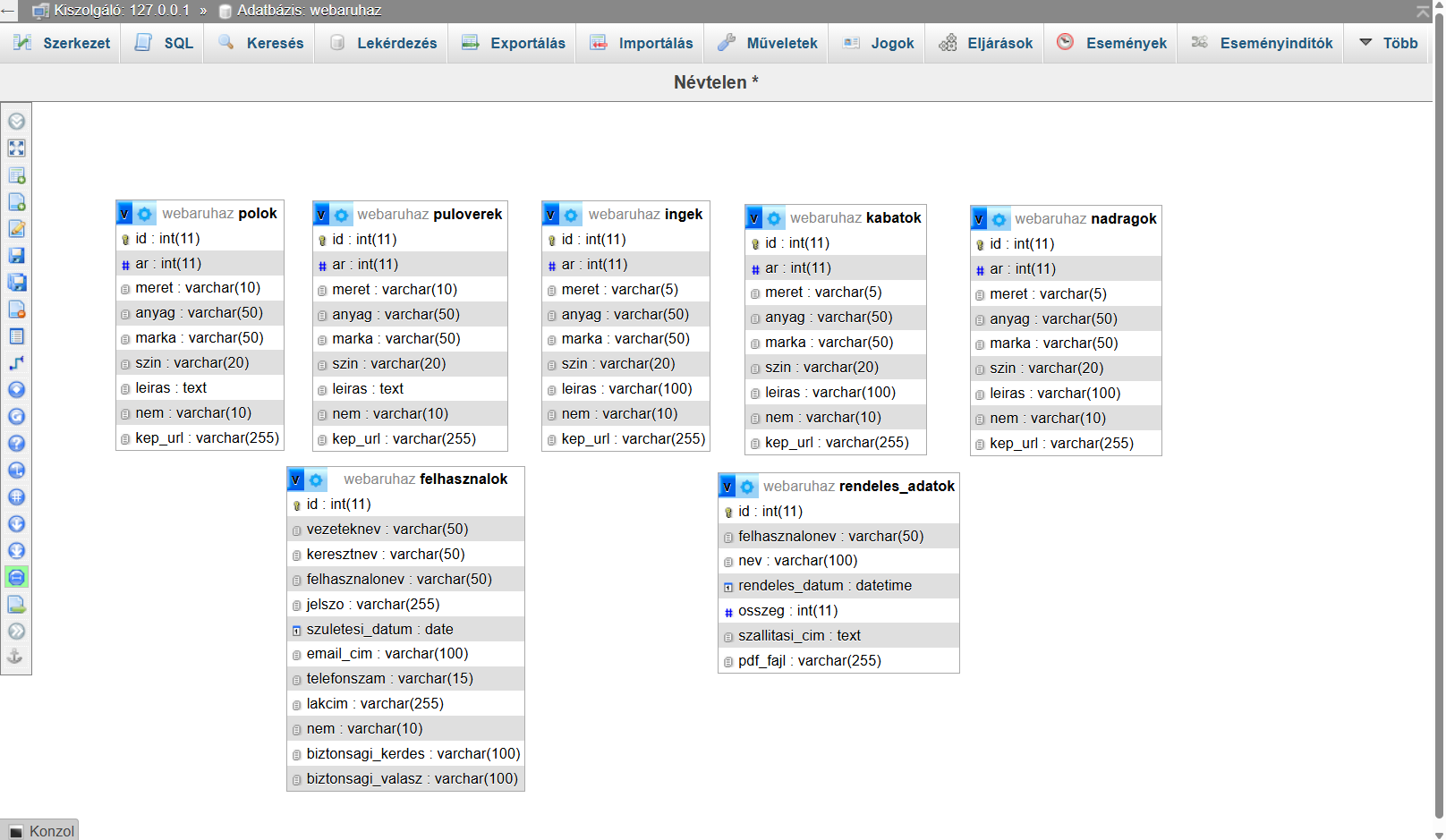
**Külső modulok jogszerű felhasználása:**

Minden külső modul nyílt forráskódú, és a licencük (MIT, BSD-2-Clause, Apache-2.0) lehetővé teszi a szabad felhasználást, módosítást és terjesztést, feltéve, hogy a licencfeltételeket betartom (pl. a szerzői jogi megjegyzések megőrzése).

### Adatbázis felépítése

**Adatmodell leírása**

A webáruház egy relációs adatbázist használ, amely a „webaruhaz” néven fut MariaDB alatt. Az adatbázis 7 táblát tartalmaz, amelyek a felhasználók, termékek és rendelések adatait tárolják. Az alábbiakban részletezem az adattáblákat és kapcsolataikat.

**Adattáblák leírása**

1. **felhasznalok**
   * **Leírás**: A regisztrált felhasználók adatait tárolja, például személyes és bejelentkezési információkat.
   * **Mezők**:
     + **id (int, PK, auto increment):** Egyedi azonosító, amely automatikusan növekszik minden új felhasználó hozzáadásakor, így biztosítva az egyedi rekordok könnyű azonosítását.
     + **vezeteknev** **(varchar(50)):** Vezetéknév, amely a felhasználó családnevét tárolja, maximum 50 karakter hosszúságban.
     + **keresztnev (varchar(50)):** Keresztnév, amely a felhasználó keresztnevét vagy adott nevét tartalmazza, szintén maximum 50 karakterig.
     + **felhasznalonev (varchar(50), egyedi):** Bejelentkezési név, amely egyedi azonosítóként szolgál a rendszerbe való belépéshez, maximum 50 karakter hosszúságban, és nem lehet két azonos érték.
     + **jelszo (varchar(255)):** Titkosított jelszó (bcrypt hash), amely a felhasználó jelszavát tárolja biztonságos, titkosított formában, bcrypt algoritmus használatával, hosszabb karakterszámra méretezve a hash miatt.
     + **szuletesi\_datum (date):** Születési dátum, amely a felhasználó születésének évét, hónapját és napját tárolja „ÉÉÉÉ-HH-NN” formátumban, például 1990-05-15.
     + **email\_cim (varchar(100)):** E-mail cím, amely a felhasználó egyedi elektronikus levelezési címét tartalmazza, maximum 100 karakter hosszúságban, és nem lehet két azonos e-mail a rendszerben.
     + **telefonszam (varchar(15)):** Telefonszám, amely a felhasználó elérhetőségét biztosítja, maximum 15 karakterben, például „+36123456789” formátumban.
     + **lakcim (varchar(255)):** Lakcím, amely a felhasználó teljes postai címét tárolja (pl. utca, házszám, város, irányítószám), maximum 255 karakter hosszúságban.
     + **nem (varchar(10)):** Nem (férfi/női), amely a felhasználó nemét jelzi, rövid szöveges formában, például „férfi” vagy „női”, maximum 10 karakterig.
     + **biztonsagi\_kerdes (varchar(100)):** Biztonsági kérdés jelszó-visszaállításhoz, amely egy felhasználó által választott kérdést tárol (pl. „Mi volt az első háziállatom neve?”), maximum 100 karakter hosszúságban.
     + **biztonsagi\_valasz (varchar(100)):** Biztonsági válasz, amely a biztonsági kérdésre adott választ tartalmazza, szintén maximum 100 karakterben, és segít a jelszó visszaállításában.
2. **ingek, kabatok, nadragok, polok, pulóverek** (terméktáblák)
   * **Leírás**: A különböző ruházati termékeket tárolják (ingek, kabátok, nadrágok, pólók, pulóverek), amelyek egy webshop kínálatát képezik. Mindegyik tábla azonos szerkezettel rendelkezik, hogy egységes legyen a termékek kezelése.
   * **Mezők**:
     + **id (int, PK, auto increment):** Egyedi azonosító, amely automatikusan generálódik minden új termék hozzáadásakor, így biztosítva az egyedi rekordok megkülönböztetését.
     + **ar (int):** Termék ára HUF-ban, amely a termék magyar forintban megadott árát tárolja egész számként, például 15000.
     + **meret (varchar(10)):** Méret (pl. S, M, L, XL).
     + **anyag (varchar(50)):** Anyagösszetétel (pl. pamut, poliészter), amely a termék gyártásához felhasznált anyagokat írja le, maximum 50 karakterben, például „100% pamut” vagy „80% poliészter, 20% elasztán”.
     + **marka (varchar(50)):** Márka neve, amely a termék gyártójának vagy forgalmazójának nevét tartalmazza, maximum 50 karakter hosszúságban, például „Nike” vagy „Adidas”.
     + **szin (varchar(20)):** Szín, amely a termék színét jelzi, maximum 20 karakterben, például „fekete”, „piros” vagy „sötétkék”.
     + **leiras (varchar(100)):** Termék leírása, amely rövid szöveges ismertetőt ad a termékről (pl. „könnyű nyári póló”), táblától függően 100 karakteres varchar vagy hosszabb szövegekre alkalmas text típusban.
     + **nem (varchar(10)):** Célzott nem (férfi/női), amely jelzi, hogy a terméket melyik nem számára tervezték, például „férfi” vagy „női”, maximum 10 karakter hosszúságban.
     + **kep\_url (varchar(255)):** Kép elérési útja, amely a termékhez tartozó kép fájlnevét vagy URL-jét tárolja, maximum 255 karakterben, például „/kepek/ing\_001.jpg”.
3. **rendeles\_adatok**
   * **Leírás**: A leadott rendelések adatait és a generált PDF-ek hivatkozásait tárolja, hogy nyomon követhetőek legyenek a vásárlások és a hozzájuk kapcsolódó dokumentumok.
   * **Mezők**:
     + **id (int, PK, auto increment):** Egyedi azonosító, amely minden új rendeléshez automatikusan generálódik, így egyedi rekordokat biztosítva a rendelésekhez.
     + **felhasznalonev (varchar(50)):** A rendelést leadó felhasználó neve (FK a felhasznalok táblára), amely a „felhasznalok” tábla „felhasznalonev” mezőjére hivatkozik idegen kulcsként, maximum 50 karakterben.
     + **nev (varchar(100)):** Vásárló neve, amely a rendelést leadó személy teljes nevét tárolja (vezetéknév és keresztnév együtt), maximum 100 karakter hosszúságban.
     + **rendeles\_datum (datetime):** Rendelés időpontja, amely a rendelés leadásának pontos dátumát és idejét rögzíti „ÉÉÉÉ-HH-NN ÓÓ:PP:MM” formátumban, például „2025-03-10 14:30:00”.
     + **osszeg (int):** Végösszeg HUF-ban, amely a rendelés teljes összegét jelzi magyar forintban, egész számként, például 25000.
     + **szallitasi\_cim (text):** Szállítási cím, amely a rendelés kiszállítási címét tartalmazza részletesen (pl. utca, házszám, város, irányítószám), hosszabb szövegekre alkalmas text típusban.
     + **pdf\_fajl (varchar(255)):** Generált PDF fájl neve, amely a rendeléshez kapcsolódó, rendszer által készített PDF dokumentum fájlnevét vagy elérési útját tárolja, maximum 255 karakterben, például „rendeles\_123.pdf”.

**Kapcsolatok**

* **felhasznalok ↔ rendeles\_adatok**: Egy felhasználó több rendelést adhat fel (1:N kapcsolat). A felhasznalonev mező köti össze a két táblát, bár nincs explicit idegenkulcs-kényszer az SQL-ben definiálva.
* A terméktáblák (ingek, kabatok, stb.) jelenleg nem állnak közvetlen kapcsolatban más táblákkal az adatbázis szintjén, de a rendelési logikában a frontend oldalon keresztül kapcsolódnak a rendeles\_adatok táblához.

### 

### Végpontok (táblázat)

### 

### Funkciók működésének, megvalósításának magyarázata

**Backend adatbázis kapcsolata**

Az adatbázis kapcsolat a Sequelize ORM segítségével történik az app.js fájlban, amely a backend belépési pontja. A kapcsolat létrehozása két fő lépésből áll:

* **Kapcsolat ellenőrzése**: Az adatbazis.authenticate() függvény aszinkron módon ellenőrzi, hogy a megadott adatbázis-hozzáférési adatok (felhasználónév, jelszó, adatbázis neve stb.) helyesek-e. A környezeti változók a .env fájlból töltődnek be a dotenv modul segítségével, így az adatbázis adatokat (pl. DB\_NAME, DB\_USER, DB\_PASSWORD, DB\_HOST) nem kell közvetlenül a forráskódba írni. Sikeres kapcsolat esetén a konzolra egy "Sikeresen csatlakoztunk az adatbázishoz!" üzenet kerül kiírásra, míg hiba esetén a folyamat kilép, és a hibát naplózza a console.error() segítségével.
* **Táblák szinkronizálása**: Az adatbazis.sync() függvény biztosítja, hogy az adatbázis sémája megfeleljen a Sequelize modellekben definiált struktúrának. Ez a függvény automatikusan létrehozza a táblákat, ha azok még nem léteznek, de alapértelmezett módban nem módosítja a meglévő táblákat. A fejlesztési környezetben célszerű az { alter: true } opció használata a táblák frissítéséhez, azonban az aktuális kódban ez az opció nincs explicit módon megadva; éles környezetben az opció kikapcsolása ajánlott a véletlen adatvesztés elkerülése érdekében.

**API végpontok működése**

Az Express.js keretrendszer használatával a backend RESTful API-t biztosít a frontend számára. Az egyes végpontok a következők szerint működnek (példák a teljesség igénye nélkül):

* **Felhasználói regisztráció (POST /api/regisztracio)**:
  + **Bemenet:** A kérés törzsében JSON formátumban érkeznek a felhasználói adatok (pl. felhasznalonev, jelszo, email\_cim, biztonsagi\_kerdes, biztonsagi\_valasz, stb.).
  + **Feldolgozás:** A jelszót a bcrypt modul titkosítja a bcrypt.hash() függvénnyel, 10-szeres iterációval, hogy biztonságos hash keletkezzen. Ezután a Sequelize.create() metódussal a felhasználói adatok a felhasznalok táblába kerülnek mentésre.
  + **Válasz:** Sikeres regisztráció esetén egy 201-es státuszkód és egy üzenet ("Sikeres regisztráció!") tér vissza, hibás bemenet (pl. hiányzó mezők vagy létező felhasználónév/email) esetén pedig 400-as vagy 409-es státusz hibakóddal.
* **Bejelentkezés (POST /api/bejelentkezes)**:
  + **Bemenet:** azonosito (lehet felhasznalonev vagy email\_cim) és jelszo.
  + **Feldolgozás:** A Sequelize findOne() metódusa megkeresi a felhasználót az azonosito alapján, amely lehet felhasználónév vagy email cím. A kapott hash-elt jelszót a bcrypt.compare() függvény összehasonlítja a bemenetként kapott jelszóval. Ha egyezik, a felhasználó adatai (jelszó nélkül) visszaadásra kerülnek.
  + **Válasz:** Sikeres bejelentkezésnél 200-as státusz és egy "Sikeres bejelentkezés!" üzenet, különben 401-es Unauthorized válasz helytelen jelszó esetén, vagy 404-es, ha a felhasználó nem található.
* **Terméklekérdezés (GET /api/ruhak)**:
  + **Bemenet:** Nincs paraméter, a végpont az összes terméktípust lekéri.
  + **Feldolgozás:** Az adatbazis.query() metódus segítségével közvetlen SQL-lekérdezésekkel gyűjti össze az összes rekordot az ingek, kabatok, nadragok, polok és pulóverek táblákból, majd egy JSON objektumba rendezi őket (ruhakLista). A Sequelize findAll() helyett itt natív SQL-lekérdezést használok.
  + **Válasz:** A termékek listája JSON formátumban tér vissza 200-as státuszkóddal, vagy üres objektum, ha nincs találat; hiba esetén 500-as státusz és hibaüzenet.

**Jelszókezelés és biztonság**

A jelszavak biztonságos tárolása érdekében a bcrypt modult használom:

* **Regisztráció és jelszó-módosítás során**: A jelszót a bcrypt.hash() függvénnyel titkosítom, amelynek során 10-szeres iterációt alkalmazok. Ez azt jelenti, hogy a hash generálása során 2^10 (1024) iteráció fut le, ami elegendő védelmet nyújt a brute-force támadások ellen, miközben nem terheli túl a szervert.
* **Biztonsági kérdések és válaszok**: Jelenleg a biztonsági kérdések és válaszok nem titkosított formában kerülnek tárolásra a felhasznalok táblában, mivel az implementáció nem tartalmazza ezek hash-elését. Ez egyszerűbbé teszi az összehasonlítást a jelszó-visszaállítási folyamat során (pl. az /elfelejtett-jelszo végpontban), de a jövőben biztonsági okokból érdemes lenne titkosítani őket, például a bcrypt.hash() segítségével 10-szeres iterációval, vagy egy szimmetrikus titkosítási módszerrel (pl. AES), hogy a jelszó-visszaállítási folyamat további védelmet kapjon.

**PDF generálás a rendelésekhez**

A rendelések PDF formátumban történő exportálásához a puppeteer modult használom:

* **Folyamat:** Amikor egy rendelés leadásra kerül, a backend a rendelési adatokat (pl. termékek, vásárló adatai) egy HTML sablonba rendereli. A puppeteer.launch() függvény elindít egy Chromium böngészőt, amely a HTML-t PDF-fé konvertálja.
* **Mentés:** A generált PDF fájl neve egyedi azonosítóval (rendeles\_${Date.now()}.pdf) kerül mentésre a szerver webaruhaz\_frontend/rendelesek mappájába, az elérési út pedig a rendeles\_adatok tábla pdf\_fajl mezőjében tárolódik.
* **Válasz**: A frontend számára a PDF relatív URL-je visszaadásra kerül (/${fajlNev}), hogy a felhasználó letölthesse.

**Rendeléskezelés**

A rendelések kezelése a következőképpen zajlik:

* **Rendelés leadása (POST /api/pdf-generalas):** A frontend elküldi a rendelési adatokat (felhasználónév, termékek, szállítási cím, stb.) JSON formátumban. A backend kiszámítja a termékek árait, majd a Sequelize.create() metódussal menti a rendeles\_adatok táblába a rendelést, miközben a PDF generálás megtörténik.
* **Időkezelés**: A moment-timezone modullal a rendeles\_datum mező pontos időzóna szerinti időt kap ("Europe/Budapest"), például YYYY.MM.DD HH:mm:ss formátumban.
* **Kapcsolat a termékekkel**: Az adatbázis szintjén nincs közvetlen idegen kulcs a terméktáblák és a rendeles\_adatok között; a rendelés adatai JSON formátumban tartalmazzák a kiválasztott termékek részleteit (pl. id, ar, mennyiseg), amelyeket a frontend biztosít.

**Hibakezelés és naplózás**

Minden végpontban try-catch blokkok biztosítják, hogy a hibák ne omoljanak össze a szerverrel:

* Hibák esetén a válasz státuszkódja (pl. 500 Internal Server Error) és egy általános hibaüzenet (pl. "Váratlan hiba történt") tér vissza a felhasználónak.
* A hibák részleteit a konzolra naplózom a console.error() segítségével, hogy a fejlesztés során könnyen azonosíthatóak legyenek (pl. "Hiba a ruhák lekérése közben")

**Környezeti változók használata**A dotenv modul lehetővé teszi, hogy az érzékeny adatokat (pl. adatbázis hitelesítő adatok, portszám) egy .env fájlban tároljuk. Ezeket a változókat a process.env objektumon keresztül érem el, ami biztosítja, hogy a kód újrafelhasználható és biztonságos maradjon.

### Backend adatbázis kapcsolata; regisztráció, bejelentkezés, kijelentkezés működése; a funkciók megvalósításának bemutatása

**Regisztráció működése (/api/regisztracio)**

* **Specifikáció**:
  + **Mit valósít meg**: Új felhasználó létrehozása az adatbázisban.
  + **Paraméterek**: felhasznalonev, jelszo, email\_cim, stb. (req.body-ból).
  + **Visszatérési érték**: JSON objektum (sikeres: { uzenet, felhasznaloAzonosito }, hiba: { hiba }).

**Algoritmus** (pszeudo-kód):  
FÜGGVÉNY regisztracio(req, res):

    HA hiányzik kötelező mező AKKOR

        VISSZA 400 "Hiányosan kitöltött mezők!"

    HA email formátuma hibás AKKOR

        VISSZA 400 "Az email cím formátuma nem megfelelő!"

    HA létezik felhasználó ezzel a névvel vagy emaillel AKKOR

        VISSZA 409 "A felhasználónév vagy email már foglalt"

    titkositottJelszo ← bcrypt.hash(jelszo, 10)

    ujFelhasznalo ← Felhasznalo.create(adatok)

    VISSZA 201 "Sikeres regisztráció!" és azonosító

VÉGE

**Bejelentkezés működése (/api/bejelentkezes)**

* **Specifikáció**:
  + **Mit valósít meg**: Felhasználó hitelesítése.
  + **Paraméterek**: azonosito (felhasználónév vagy email), jelszo.
  + **Visszatérési érték**: JSON (sikeres: { uzenet, felhasznalo }, hiba: { hiba }).

**Algoritmus**:  
FÜGGVÉNY bejelentkezes(req, res):

    HA hiányzik azonosító vagy jelszó AKKOR

        VISSZA 400 "Kötelező mezők hiányoznak!"

    felhasznalo ← Felhasznalo.findOne(felhasznalonev vagy email)

    HA nem létezik felhasznalo AKKOR

        VISSZA 404 "Felhasználó nem található!"

    HA bcrypt.compare(jelszo, felhasznalo.jelszo) = hamis AKKOR

        VISSZA 401 "Helytelen jelszó!"

    VISSZA 200 "Sikeres bejelentkezés!" és felhasználó adatai

VÉGE

**Kijelentkezés**

A backend jelenleg nem implementál kijelentkezést, mert stateless REST API-t használ (nincs munkamenet-kezelés).

**Funkciók megvalósítása**

* **Ruhák lekérése (/api/ruhak)**:
  + Öt Promise.all hívással lekéri az összes terméktábla adatait párhuzamosan.
  + Visszatér egy objektummal, amely kategóriánként tartalmazza a termékeket.
* **PDF generálás (/api/pdf-generalas)**:
  + Puppeteer segítségével HTML-ből PDF-et készít, majd menti az adatbázisba és fájlrendszerbe.

**Funkciók megvalósítása**

A teljes forráskódot nem illesztem be a dokumentációba, de a fontosabb részeket magyarázattal kiemeltem a fenti szakaszokban (pl. regisztráció, bejelentkezés). A teljes kód a mellékelt fájlokban (app.js, routes/auth.js, stb.) található.

## 

## Frontend

### Elkészített komponensek, és működésük

### Routing működése, frontend végpontok felépítése

### Backend adatbázis kapcsolata; regisztráció, bejelentkezés, kijelentkezés működése; a funkciók megvalósításának bemutatása

# Tesztelés

## Backend

## Frontend

## Terheléses teszt

## 

# Továbbfejlesztési lehetőségek

A projektet az alábbi ötletekkel bővítenénk a jövőben:

* **Token alapú hitelesítés:**
  + JSON Web Token (JWT) implementálása a biztonságos és skálázható hitelesítés érdekében.
* **Kívánságlista:**
  + Termékek mentése regisztrált felhasználók számára későbbi vásárlásokhoz.
* **Rendeléskövetés:**
  + Valós idejű rendelési státuszok (pl. "Feldolgozás alatt", "Kiszállítva").
* **Többnyelvű támogatás:**
  + Magyar és angol nyelvű felület az idegen nyelvű vásárlók eléréséhez.
* **Ajánlatok rendszer**:
  + Személyre szabott termékajánlatok a vásárlási előzmények alapján.
* **Fizetési módok integrálása**:
  + Online fizetési módok (pl. Stripe, PayPal) támogatása.
* **Mobilalkalmazás:**
  + A webshop elérése mobilalkalmazás formájában is.
* **Teljesítmény optimalizálás:**
  + Gyorsabb betöltési idők cache-eléssel és szerveroldali rendereléssel (SSR).
* **Visszajelzések és értékelések:**
  + Termékek értékelése és vélemények megosztása felhasználók által.
* **Automatizált e-mail értesítések:**
  + Rendelés-visszaigazolások és státuszfrissítések e-mailben.
* **Adminisztrátori és statisztikai felület:**
  + Eladási statisztikák és felhasználói aktivitás nyomon követése adminisztrátorok számára, valamint felhasználók kezelése.
* **Szűrők és ruhák bővítése:**
  + További ruha kategóriák (pl. zoknik, sálak, alsóneműk, sapkák)
  + Több szűrési lehetőség hozzáadása (pl. gyermek vagy felnőtt ruha szűrése, több méret hozzáadása)

# Irodalomjegyzék

Képek:  
<https://www.flaticon.com/free-icon/cursor_18356759> 2024.12.14 11:16:54

<https://www.flaticon.com/free-icon/left-click_10187229> 2024.12.14 11:17:24

<https://www.flaticon.com/free-icon/clothing_925072> 2024.12.09 8:32:35

<https://www.pexels.com/photo/close-up-of-row-325876/> 2024.12.12 11:34:07

<https://www.markasbolt.hu/heavy-tools-merettablazat> 2025.02.25 12:10:06

A többi kép a weboldalon (háttér, favicon, ruhák képei stb.) a ChatGPT képgeneráló szoftverével készült: <https://chatgpt.com>