**HaLáli Kft. – dokumentáció**

Ez a dokumentáció a HaLáli Kft. webshop alkalmazásának főbb PHP, JavaScript és CSS fájljainak működését írja le.

**1. Szerveroldali Fájlok (PHP)**

**adatkezeles.php**

* **Leírás:** Ez a fájl jeleníti meg az Adatkezelési Tájékoztató oldalt. Statikus HTML tartalmat tartalmaz, amely ismerteti a cég adatkezelési gyakorlatait a GDPR rendeletnek megfelelően.
* **Függőségek:**
  + ./nav.php: Navigációs menü beillesztése.
  + ./footer.php: Lábléc beillesztése.
  + ./style/style.css: Stíluslap.
  + Bootstrap CSS, Google Fonts.
* **Működés:**
  + Beilleszti a navigációs menüt és a láblécet.
  + Megjeleníti a HTML struktúrában definiált adatkezelési információkat (adatkezelő adatai, cél, kezelt adatok köre, jogalap, tárolási időtartam, érintettek jogai, adattovábbítás, panasztételi lehetőség).
  + Formázást Bootstrap és egyedi CSS (style.css) segítségével kap.

**imp.php**

* **Leírás:** Az Impresszum oldalt jeleníti meg. Statikus HTML tartalom, amely a cég alapvető adatait, bemutatkozását, tevékenységi köreit és elérhetőségeit tartalmazza.
* **Függőségek:**
  + ./nav.php: Navigációs menü beillesztése.
  + ./footer.php: Lábléc beillesztése.
  + ./style/style.css: Stíluslap.
  + Bootstrap CSS, Google Fonts.
* **Működés:**
  + Beilleszti a navigációs menüt és a láblécet.
  + Megjeleníti a HTML struktúrában definiált impresszum információkat (cégnév, székhely, képviselők, elérhetőségek, cégbemutató, tevékenységi körök, nyitvatartás).
  + Formázást Bootstrap és egyedi CSS (style.css) segítségével kap.

**db.php**

* **Leírás:** Ez a fájl felelős az adatbázis-kapcsolat létrehozásáért.
* **Függőségek:** Nincs (de sok más fájl függ tőle).
* **Működés:**
  + Definiálja az adatbázis kapcsolódási paramétereket (host, adatbázis neve, felhasználónév, jelszó).
  + Létrehoz egy PDO (PHP Data Objects) kapcsolatot a MySQL adatbázishoz.
  + Beállítja a PDO hibakezelési módját PDO::ERRMODE\_EXCEPTION-re, ami kivételeket dob adatbázis hiba esetén.
  + Ha a kapcsolat sikertelen, kiír egy hibaüzenetet és leállítja a script futását (exit()).
  + A sikeresen létrehozott $pdo objektumot használják más fájlok az adatbázis műveletekhez.

**bejelentkezes.php**

* **Leírás:** Kezeli a felhasználói bejelentkezési folyamatot.
* **Függőségek:**
  + db.php: Adatbázis kapcsolat.
  + nav.php: Navigációs menü.
  + footer.php: Lábléc.
  + style/style.css: Stíluslap.
  + Google reCAPTCHA API.
* **Működés:**
  + Elindítja a munkamenetet (session\_start()).
  + Beállítja a Cache-Control és Pragma HTTP fejléceket a gyorsítótárazás tiltására.
  + Ha POST kérés érkezik (az űrlap elküldésekor):
    - Ellenőrzi a Google reCAPTCHA választ a Google szerverén.
    - Ha a reCAPTCHA érvényes:
      * Lekéri a felhasználó által megadott felhasználónevet és jelszót.
      * Lekérdezi az adatbázisból a felhasználót a megadott felhasználónév alapján (PDO).
      * Ha a felhasználó létezik:
        + Ellenőrzi a megadott jelszót a password\_verify() függvénnyel az adatbázisban tárolt hash-hez képest.
        + Sikeres jelszó ellenőrzés esetén:

Beállítja a session változókat: $\_SESSION['felhasznalo'] (felhasználói adatok tömbje), $\_SESSION['jogosultsag'] (pl. 'admin' vagy 'user'), és $\_SESSION['admin\_logged\_in'] (ha admin).

Átirányítja a felhasználót a kosár oldalra.

* + - * + Hibás jelszó esetén hibaüzenetet ($error) generál.
      * Ha a felhasználó nem létezik, szintén hibaüzenetet generál.
    - Ha a reCAPTCHA érvénytelen, visszairányít a bejelentkezési oldalra.
  + Ha GET kérés érkezik:
    - Megjeleníti a bejelentkezési űrlapot.
    - Megjeleníti a kijelentkezés utáni sikerüzenetet, ha a logout=1 paraméter szerepel az URL-ben.
    - Megjeleníti a $error változóban tárolt hibaüzenetet, ha van.
    - Beilleszti a reCAPTCHA komponenst az űrlapba.

**fooldal.php**

* **Leírás:** Az webshop főoldalát (kezdőlapját) jeleníti meg.
* **Függőségek:**
  + nav.php: Navigációs menü.
  + footer.php: Lábléc.
  + style/style.css: Stíluslap.
  + Bootstrap CSS/JS, Google Fonts, Font Awesome.
* **Működés:**
  + Elindítja a munkamenetet (session\_start()).
  + Beállítja a Cache-Control és Pragma HTTP fejléceket.
  + Beilleszti a navigációt és a láblécet.
  + Megjelenít egy Bootstrap képváltót (carousel).
  + Megjelenít több statikus tartalmi blokkot (Rólunk, Kínálatunk, Kalkulátor, Ügyfélszolgálat) képekkel, szövegekkel és gombokkal, amelyek a weboldal más részeire mutatnak (termekek.php, kalkulator.php, kapcsolat.php).

**footer.php**

* **Leírás:** A weboldal láblécét tartalmazó, újrafelhasználható komponens.
* **Függőségek:** Nincs.
* **Működés:**
  + HTML <footer> elemet definiál.
  + Egy táblázatban megjeleníti a logót, az alkotók nevét, email címet, céget.
  + Linkeket tartalmaz az Impresszum (imp.php), Adatkezelési tájékoztató (adatkezeles.php) és Kapcsolat (kapcsolat.php) oldalakra.
  + Megjeleníti az aktuális évet (<?= date("Y"); ?>) és a copyright szöveget.

**kalkulator.php**

* **Leírás:** Egy oldalt biztosít különböző villamossági számítások elvégzésére (feszültségesés, ellenállás).
* **Függőségek:**
  + nav.php: Navigációs menü.
  + footer.php: Lábléc.
  + ./js/kalkulator.js: A számítások logikája és UI kezelése.
  + ./style/style.css: Stíluslap.
  + Bootstrap CSS, Google Fonts.
* **Működés:**
  + Elindítja a munkamenetet.
  + Beállítja a cache kontroll fejléceket.
  + Beilleszti a navigációt és a láblécet.
  + HTML űrlapokat tartalmaz a feszültségesés és ellenállás számításához szükséges adatok (vezeték hossz, keresztmetszet, áramerősség, fázisok száma, feszültség) bekérésére.
  + Rádiógombokkal lehet váltani a kalkulátorok között.
  + A tényleges számításokat és az eredmények megjelenítését a kalkulator.js végzi.

**kapcsolat\_kuldes.php**

* **Leírás:** Feldolgozza a kapcsolat.php oldalon található kapcsolatfelvételi űrlap elküldött adatait és e-maileket küld.
* **Függőségek:**
  + vendor/autoload.php: Composer által generált autoload fájl (PHPMailer és Dotenv betöltéséhez).
  + PHPMailer library.
  + Dotenv library.
  + .env fájl (SMTP adatokhoz).
* **Működés:**
  + Betölti a .env fájlban tárolt környezeti változókat (pl. SMTP felhasználónév, jelszó).
  + Elindítja a munkamenetet.
  + Csak POST kéréseket dolgoz fel.
  + Megszerzi és tisztítja (trim) a beküldött adatokat (név, email, üzenet).
  + Ellenőrzi, hogy minden mező ki van-e töltve. Hiba esetén session hibaüzenetet állít be és visszairányít a kapcsolat.php-ra.
  + Ha az adatok rendben vannak:
    - Létrehoz egy PHPMailer példányt az ügyfélszolgálatnak szánt emailhez.
    - Beállítja az SMTP hitelesítést és kapcsolatot a .env fájl adatai alapján (Gmail SMTP).
    - Beállítja a feladót (a látogató email címe és neve) és a címzettet (a cég email címe, pl. halalikft@gmail.com).
    - Beállítja a tárgyat és a HTML formátumú email törzset a látogató adataival és üzenetével.
    - Elküldi az emailt a cégnek.
    - Létrehoz egy másik PHPMailer példányt a látogatónak szánt visszaigazoló emailhez.
    - Beállítja az SMTP-t ugyanúgy.
    - Beállítja a feladót (a cég email címe) és a címzettet (a látogató email címe).
    - Beállítja a tárgyat és a HTML email törzset (köszönet, üzenet idézése).
    - Elküldi a visszaigazoló emailt a látogatónak.
    - Sikeres küldés esetén session sikerüzenetet állít be.
    - Hiba esetén (pl. email küldés sikertelen) session hibaüzenetet állít be a kivétel üzenetével.
  + Minden esetben visszairányít a kapcsolat.php oldalra.

**kapcsolat.php**

* **Leírás:** Megjeleníti a kapcsolatfelvételi oldalt űrlappal, elérhetőségekkel és térképpel.
* **Függőségek:**
  + nav.php: Navigációs menü.
  + footer.php: Lábléc.
  + ./style/style.css: Stíluslap.
  + Bootstrap CSS, Google Fonts, Font Awesome.
* **Működés:**
  + Elindítja a munkamenetet és beállítja a cache kontroll fejléceket.
  + Beilleszti a navigációt és a láblécet.
  + Megjeleníti a cég logóját és elérhetőségi adatait (cím, telefon, email, nyitvatartás).
  + Megjelenít egy kapcsolatfelvételi űrlapot (név, email, üzenet mezőkkel), amely a kapcsolat\_kuldes.php-nak küldi az adatokat POST metódussal.
  + Megjeleníti a kapcsolat\_kuldes.php által beállított siker- vagy hibaüzeneteket a session-ből, majd törli őket.
  + Beágyaz egy Google Maps térképet a cég címével.
  + Megjeleníti a közösségi média ikonokat (Facebook, Instagram, Twitter, LinkedIn) linkekkel.

**kijelentkezes.php**

* **Leírás:** Kezeli a felhasználó kijelentkeztetését.
* **Függőségek:** Nincs.
* **Működés:**
  + Ellenőrzi, hogy a session el van-e indítva, ha nincs, elindítja.
  + Nullázza a $\_SESSION["cart\_szamlalo"] változót (ennek hatása kérdéses a session\_destroy előtt/után).
  + Törli az összes session változót (session\_unset()).
  + Megsemmisíti a munkamenetet (session\_destroy()).
  + Átirányítja a felhasználót a bejelentkezes.php oldalra egy logout=1 paraméterrel, jelezve a sikeres kijelentkezést.

**kosar\_muveletek.php**

* **Leírás:** Feldolgozza a termékek kosárba helyezését.
* **Függőségek:**
  + db.php: Adatbázis kapcsolat (PDO).
  + sql\_fuggvenyek.php: Adatbázis segédfüggvények (mysqli - *ellentmondás!*).
* **Működés:**
  + Elindítja a munkamenetet.
  + Ellenőrzi, létezik-e a $\_SESSION['kosar'], ha nem, létrehozza üres tömbként (bár a kód nem használja közvetlenül a session kosarat itt).
  + Csak POST kérést dolgoz fel, amely tartalmazza az add\_to\_cart paramétert.
  + Lekéri a termék ID-t és a mennyiséget a POST adatokból.
  + Lekéri a bejelentkezett felhasználó nevét (fh\_nev) a session-ből. Ha nincs bejelentkezve, átirányít a kosar oldalra (ami valószínűleg továbbirányít a loginra).
  + **Adatbázis műveletek:**
    - Megkeres egy "nyitott" (fizetesi\_mod üres, osszeg 0, szallitas üres) rendelést (megrendeles tábla) az adott felhasználóhoz (PDO).
    - Ha nincs ilyen rendelés, létrehoz egy újat a felhasználó nevével és az aktuális dátummal (PDO). Lekéri az új rendelés ID-jét.
    - Ha van nyitott rendelés, annak az ID-jét használja.
    - Beilleszt egy új tételt a tetelek táblába, vagy ha már van ilyen termék az adott rendeléshez és felhasználóhoz, növeli a mennyiségét (tetelek\_mennyiseg) az ON DUPLICATE KEY UPDATE segítségével (PDO).
  + **Kosárszámláló frissítése:**
    - Lekérdezi a legutóbbi rendelés ID-t a felhasználóhoz (mysqli, adatokLekerdezese).
    - Megszámolja a 'kosárban' státuszú tételeket az adott felhasználóhoz és rendelés ID-hez a tetelek táblában (mysqli, adatokLekerdezese).
    - Beállítja a $\_SESSION["cart\_szamlalo"] értékét a kapott darabszámra.
  + Beállít egy session üzenetet a sikeres hozzáadásról.
  + Átirányítja a felhasználót a termekek oldalra.
* **Megjegyzés:** Keveri a PDO és a mysqli adatbázis-kezelést, ami komoly inkonzisztencia és potenciális hibaforrás. A kosárszámláló logikája a mysqli függvényeket használja, míg a fő tételműveletek PDO-t.

**kosar.php**

* **Leírás:** Megjeleníti a felhasználó kosarának tartalmát, lehetővé teszi a mennyiségek módosítását, tételek törlését, a kosár ürítését és a rendelés leadását (fizetési folyamat indítása).
* **Függőségek:**
  + db.php: Adatbázis kapcsolat (PDO).
  + sql\_fuggvenyek.php: Adatbázis segédfüggvények (mysqli).
  + nav.php: Navigációs menü.
  + footer.php: Lábléc.
  + style/style.css: Stíluslap.
  + rendeles\_visszaigazolo\_mailer.php (feltételezve, hogy ez a helyes név a rendeles\_visszaigazolas.php helyett): Email küldéshez.
* **Működés:**
  + Elindítja a munkamenetet, beállítja a cache kontrollt.
  + Inicializálja a $\_SESSION['kosar']-t, ha szükséges.
  + Lekéri a felhasználónevet a session-ből.
  + **Kosár betöltése:** Meghívja a betolt\_kosar\_adatbazisbol függvényt, ami lekérdezi a 'kosárban' státuszú tételeket a tetelek táblából a felhasználóhoz, összekapcsolva a termek táblával a részletekért (név, ár, kép, készlet), és betölti az eredményt a $\_SESSION['kosar']-ba (PDO).
  + **Kosárszámláló:** Frissíti a $\_SESSION['cart\_szamlalo'] értékét adatbázis lekérdezéssel (mysqli, adatokLekerdezese).
  + **Segédfüggvények:**
    - frissit\_session\_adatbazisbol: (Opcionálisnak jelölt) frissítené a felhasználói adatokat a sessionben az adatbázisból (nincs meghívva a kódban).
    - teljes\_e\_a\_profil: Ellenőrzi, hogy a sessionben lévő felhasználói adatok ($\_SESSION['felhasznalo']) tartalmazzák-e a kötelező számlázási (és opcionálisan szállítási, céges) adatokat.
    - osszegzo: Kiszámolja a kosárban lévő termékek összértékét (figyelembe veszi az akciós árat).
    - szallitas\_dij: Kiszámolja a szállítási díjat (ingyenes 25000 Ft felett, egyébként 1690 Ft).
  + **POST Kérések Kezelése:**
    - delete\_item: Törli a megadott termek\_id-jű tételt a tetelek táblából a felhasználóhoz (PDO). Frissíti a kosárszámlálót (mysqli). *Nem módosítja a session kosarat közvetlenül, a következő oldalbetöltéskor a betolt\_kosar\_adatbazisbol frissíti.* Átirányít önmagára.
    - update\_cart: Végigmegy a beküldött mennyisegek tömbön. Ha az új mennyiség 0 vagy kevesebb, törli a tételt (PDO). Ha több, frissíti a tetelek\_mennyiseg értékét az adatbázisban (PDO). Frissíti a kosárszámlálót (mysqli). *Hibásan próbálja frissíteni a termék készletét.* Átirányít önmagára.
    - empty\_cart: Törli az összes 'kosárban' státuszú tételt a felhasználóhoz a tetelek táblából (PDO). Frissíti a kosárszámlálót (mysqli). Üríti a $\_SESSION['kosar']-t. *Hibásan próbálja frissíteni a termék készletét.* Átirányít önmagára.
    - fizetes: Ez a rendelés véglegesítése.
      1. **Készletellenőrzés:** Lekérdezi a kosárban lévő tételekhez tartozó termékek aktuális készletét és a rendelt mennyiséget (mysqli, adatokLekerdezese). Ellenőrzi, hogy minden termékből van-e elég raktáron.
      2. **Készlet Levonás:** Ha van elég készlet, végigmegy a tételeken és csökkenti a termek táblában az elerheto\_darab értékét a rendelt mennyiséggel (PDO).
      3. **Hiba Kezelés:** Ha nincs elég készlet vagy a készletlevonás sikertelen, átirányít a rendeles\_sikertelen.php-ra.
      4. **Rendelés Frissítése:** Ha a készletkezelés sikeres, lekéri a szállítási és fizetési módot a POST adatokból. Kiszámolja az összeget, szállítási díjat, végösszeget. Lekérdezi a legutóbbi (még nyitott) rendelés ID-jét és dátumát (mysqli, adatokLekerdezese). Frissíti a megrendeles tábla megfelelő rekordját a kiszámolt összegekkel, szállítási/fizetési módokkal (PDO).
      5. **Tételek Státusz Frissítése:** Frissíti a kosárban lévő tételek státuszát 'leadva'-ra a tetelek táblában, hozzárendelve a véglegesített rendelés ID-jét (PDO).
      6. **Kosár Ürítése:** Üríti a $\_SESSION['kosar']-t és frissíti a kosárszámlálót (ami 0 lesz).
      7. **Email Küldés:** Beinclude-olja és futtatja a rendeles\_visszaigazolo\_mailer.php-t (a megadott rendeles\_visszaigazolas.php fájlt) az email elküldéséhez.
      8. **Átirányítás:** Átirányít a rendeles\_sikeres.php oldalra.
  + **Megjelenítés:**
    - Ellenőrzi, be van-e jelentkezve a felhasználó ($bejelentkezve).
    - Ellenőrzi, ki van-e töltve a profilja ($profil\_teljes). Megfelelő üzeneteket jelenít meg (pl. "Jelentkezzen be!", "Töltse ki a profilját!").
    - Ha a kosár üres, üzenetet jelenít meg.
    - Ha van termék a kosárban és a felhasználó be van jelentkezve/profilja teljes:
      1. Kilistázza a kosár tartalmát ($\_SESSION['kosar'] alapján) képekkel, nevekkel, árakkal, mennyiség beviteli mezőkkel és törlés gombokkal egy form-on belül (update\_cart, empty\_cart, delete\_item céljára).
      2. Megjeleníti a rendelési összesítőt (szállítási mód, fizetési mód választók, összegek).
      3. A "Fizetés" gomb elküldi a rendelést (fizetes POST).
      4. JavaScript kódot tartalmaz a bankkártya adatokat bekérő mezők dinamikus megjelenítésére/elrejtésére a fizetési mód választása alapján.
* **Megjegyzések:**
  + Jelentős inkonzisztencia a PDO és mysqli használata között.
  + A készletkezelés (update\_cart, empty\_cart) hibásnak tűnik.
  + A delete\_item nem frissíti a session kosarat, csak az adatbázist, ami zavart okozhat az oldal újratöltése előtt.

**nav.php**

* **Leírás:** A weboldal felső navigációs menüjét tartalmazó, újrafelhasználható komponens.
* **Függőségek:**
  + kosarszamlalo.php: (Nincs mellékelve) Valószínűleg lekérdezi és kiírja a $\_SESSION['cart\_szamlalo'] értékét.
* **Működés:**
  + Egy Bootstrap navbar komponenst definiál.
  + Megjeleníti a logót.
  + Tartalmazza a fő menüpontokat (Kezdőlap, Termékek, Kalkulátor, Kapcsolat).
  + Jobb oldalon megjeleníti a kosár ikont, mellette a kosarszamlalo.php által biztosított darabszámmal egy piros jelvényben (badge).
  + Megjelenít egy profil ikont, amely egy legördülő menüt nyit:
    - Ha a felhasználó be van jelentkezve (isset($\_SESSION['felhasznalo'])):
      * Profil (profil.php)
      * Admin dashboard (admin/dashboard.php), ha a jogosultság 'admin'.
      * Rendelések (rendeles\_kovetes.php)
      * Kijelentkezés (kijelentkezes.php)
    - Ha nincs bejelentkezve:
      * Regisztráció (regisztracio.php)
      * Bejelentkezés (bejelentkezes.php)
  + Reszponzív: Kisebb képernyőkön hamburger menüvé alakul, és az ikonok (kosár, profil) elrendezése változik.
  + Tartalmazza a Bootstrap JS bundle-t és egy kis JavaScriptet a hamburger ikon animációjához.

**profil.php**

* **Leírás:** Lehetővé teszi a bejelentkezett felhasználó számára személyes adatainak és jelszavának módosítását.
* **Függőségek:**
  + db.php: Adatbázis kapcsolat (PDO).
  + nav.php: Navigációs menü.
  + kijelentkezes.php: A profil mentése után meghívódik.
  + ./js/inputmask.min.js, ./js/inputmask.js: (Nincs mellékelve) Telefonszám formázásához.
  + ./style/style.css: Stíluslap.
* **Működés:**
  + Elindítja a munkamenetet, beinclude-olja az adatbázis kapcsolatot.
  + Ellenőrzi, hogy a felhasználó be van-e jelentkezve ($\_SESSION['felhasznalo']). Ha nem, átirányítja a bejelentkezes.php-ra.
  + Lekérdezi a bejelentkezett felhasználó (fh\_nev) összes adatát az adatbázisból (PDO).
  + **POST Kérések Kezelése:**
    - Ha a password\_change POST paraméter létezik (jelszóváltoztatás):
      * Ellenőrzi a jelenlegi jelszót (password\_verify).
      * Ellenőrzi, hogy az új jelszavak megegyeznek-e.
      * Ellenőrzi az új jelszó minimális hosszát (6 karakter).
      * Ha minden rendben, hasheli az új jelszót (password\_hash) és frissíti az adatbázisban (PDO).
      * Siker vagy hiba esetén üzenetet ($message) állít be.
    - Ha a password\_change nincs (profiladatok módosítása):
      * Lekéri az űrlapról a személyes adatokat, számlázási címet.
      * Ellenőrzi a same\_address checkboxot, és ennek megfelelően állítja be a kézbesítési címet (vagy veszi az űrlapról).
      * Ellenőrzi a ceg checkboxot, és ennek megfelelően állítja be a cégnevet/adószámot (vagy null-ra).
      * Frissíti a felhasználó összes adatát az adatbázisban (PDO).
      * Üzenetet ($message) állít be a sikerességről/hibáról.
      * **Fontos:** A profiladatok sikeres (vagy sikertelen) mentése után **meghívja a kijelentkezes.php-t**, így a felhasználót kijelentkezteti a módosítás után. Ez kényszeríti az újbóli bejelentkezést a frissített adatokkal.
  + **Megjelenítés:**
    - Beilleszti a navigációs menüt.
    - Megjeleníti a felhasználónevet a címben.
    - Megjeleníti a $message változó tartalmát, ha van.
    - Megjelenít egy különálló űrlapot a jelszó módosításához.
    - Megjelenít egy fő űrlapot a többi adat módosításához, előre kitöltve a felhasználó aktuális adataival. Külön kártyákba (card) rendezi a személyes, számlázási, kézbesítési és céges adatokat.
    - Tartalmazza a "Számlázási címmel megegyezik" és a "Céges vásárlás" checkboxokat.
    - Figyelmeztetést jelenít meg, hogy mentés után kijelentkeztetés történik.
    - JavaScriptet használ a telefonszám formázásához (Inputmask) és a kézbesítési cím mezők engedélyezéséhez/tiltásához/kitöltéséhez a same\_address checkbox alapján.

**regisztracio.php**

* **Leírás:** Kezeli az új felhasználók regisztrációs folyamatát.
* **Függőségek:**
  + db.php: Adatbázis kapcsolat (PDO).
  + nav.php: Navigációs menü.
  + footer.php: Lábléc.
  + style/style.css: Stíluslap.
  + Google reCAPTCHA API.
  + Bootstrap JS (a sikerességi modálhoz).
* **Működés:**
  + Elindítja a munkamenetet, beállítja a cache kontrollt, include-olja az adatbázist.
  + **POST Kérés Kezelése:**
    - Ellenőrzi a Google reCAPTCHA választ.
    - Ha a reCAPTCHA érvényes:
      * Lekéri a beküldött adatokat (vezeték-/keresztnév, felhasználónév, email, jelszó, jelszó újra).
      * Ellenőrzi, hogy a felhasználónév vagy email cím létezik-e már az adatbázisban (PDO). Hiba esetén $error üzenetet állít be.
      * Ellenőrzi, hogy a két megadott jelszó megegyezik-e. Hiba esetén $error üzenetet állít be.
      * Ha nincs hiba:
        + Hasheli a jelszót (password\_hash).
        + Beilleszti az új felhasználó adatait a felhasznalo táblába (PDO).
        + Sikeres beszúrás esetén beállítja a $\_SESSION['reg\_success'] = true; változót.
        + Sikertelen beszúrás esetén $error üzenetet állít be.
    - Ha a reCAPTCHA érvénytelen, visszairányít a regisztrációs oldalra (nincs explicit hibaüzenet a felhasználónak).
  + **Megjelenítés (GET kérés vagy POST után):**
    - Beilleszti a navigációt és a láblécet.
    - Megjeleníti a regisztrációs űrlapot (név, felhasználónév, email, jelszavak, adatkezelési checkbox).
    - Megjeleníti a $error változó tartalmát, ha van.
    - Beilleszti a reCAPTCHA komponenst.
    - Ha a $\_SESSION['reg\_success'] be van állítva (sikeres regisztráció után):
      * Megjelenít egy Bootstrap modális ablakot (felugró ablak) a sikeres regisztrációról.
      * JavaScript segítségével az "OK" gombra kattintva vagy 5 másodperc után automatikusan átirányítja a felhasználót a bejelentkezes.php-ra.
      * Törli a $\_SESSION['reg\_success'] változót.

**rendeles\_kovetes\_fetch.php**

* **Leírás:** API végpontként szolgál a rendeles\_kovetes.php oldal számára, hogy lekérdezze a felhasználó rendelési adatait AJAX segítségével.
* **Függőségek:**
  + db.php: Adatbázis kapcsolat (PDO).
  + sql\_fuggvenyek.php: Adatbázis segédfüggvények (mysqli - *ellentmondás!*).
* **Működés:**
  + Elindítja a munkamenetet, beállítja a cache kontrollt, include-olja az adatbázis fájlokat.
  + Lekéri a bejelentkezett felhasználó nevét.
  + Feldarabolja a kérés URI-ját, hogy meghatározza a kért erőforrást (pl. 'rendelesek', 'rendeles\_reszletek').
  + A kért erőforrás (switch alapján) és a HTTP metódus ($\_SERVER['REQUEST\_METHOD']) alapján hajt végre műveleteket:
    - **GET /rendelesek**: Lekérdezi a felhasználóhoz tartozó összes rendelés alapadatait (ID, végösszeg, dátum, státusz) a megrendeles táblából, ID szerint csökkenő sorrendben (mysqli, adatokLekerdezese). JSON formátumban visszaadja az eredményt. Hiba esetén hiba JSON-t és 400-as státuszkódot küld.
    - **POST /rendeles\_reszletek**: Lekéri a POST kérés törzséből (JSON) a rendelés ID-t és státuszt. Lekérdezi a megadott rendeléshez tartozó tételek részletes adatait (termék név, kép, ár, mennyiség) és a rendelés egyéb adatait (szállítási/fizetési mód, végösszeg, dátum, státusz) a termek, tetelek, megrendeles táblák összekapcsolásával (PDO). JSON formátumban visszaadja az eredményt. Hiba esetén hiba JSON-t és 400-as státuszkódot küld.
    - **Egyéb kérések**: Hiba JSON-t és 400-as státuszkódot küld.
* **Megjegyzés:** Keveri a PDO és mysqli használatát.

**rendeles\_kovetes.php**

* **Leírás:** Megjeleníti a felhasználó korábbi rendeléseinek listáját és azok állapotát. Lehetővé teszi a rendelések részleteinek megtekintését.
* **Függőségek:**
  + db.php: Adatbázis kapcsolat.
  + sql\_fuggvenyek.php: Adatbázis segédfüggvények.
  + nav.php: Navigációs menü.
  + footer.php: Lábléc.
  + ./js/rendeles\_kovetes.js: Az adatok lekérését és megjelenítését végző JavaScript.
  + ./style/style.css: Stíluslap.
* **Működés:**
  + Elindítja a munkamenetet, beállítja a cache kontrollt, include-olja az adatbázis fájlokat, navigációt, láblécet.
  + HTML struktúrát biztosít a rendelések megjelenítéséhez (pl. rendelesTable, rendeles, hibauzenet ID-jű divek).
  + Beinclude-olja a rendeles\_kovetes.js fájlt, amely AJAX hívásokkal lekéri az adatokat a rendeles\_kovetes\_fetch.php-től és dinamikusan feltölti a HTML elemeket a rendelési listával, állapotsávval és részletekkel.

**rendeles\_sikeres.php**

* **Leírás:** A sikeres rendelés leadása után megjelenő oldal.
* **Függőségek:**
  + db.php: Adatbázis kapcsolat (bár itt nem használja).
  + nav.php: Navigációs menü.
  + style/style.css: Stíluslap.
* **Működés:**
  + Elindítja a munkamenetet, include-olja az adatbázist (feleslegesen?), navigációt.
  + Megjelenít egy "Sikeres megrendelés!" üzenetet és tájékoztatást a visszaigazoló emailről.
  + JavaScript segítségével egy 10 másodperces visszaszámlálást indít, majd automatikusan átirányítja a felhasználót a főoldalra (fooldal.php).
  + Tartalmaz egy gombot is a főoldalra való azonnali visszatéréshez.

**rendeles\_visszaigazolas.php**

* **Leírás:** Felelős a rendelés-visszaigazoló email elküldéséért. Ezt a fájlt a kosar.php hívja meg a sikeres rendelés (fizetes POST) végén.
* **Függőségek:**
  + vendor/autoload.php: Composer autoload (PHPMailer, Dotenv).
  + PHPMailer library.
  + Dotenv library.
  + .env fájl (SMTP adatokhoz).
  + A kosar.php kontextusából elérhető session adatok ($\_SESSION['felhasznalo']) és rendelési adatok ($ID\_megrendeles, $vegosszeg, $szallitasi\_mod, $fizetesi\_mod).
* **Működés:**
  + Betölti a .env változókat.
  + Definiál egy sendOrderConfirmation függvényt, amely:
    - Létrehoz egy PHPMailer példányt.
    - Beállítja az SMTP kapcsolatot a .env adatok alapján (Gmail).
    - Beállítja a feladót (cég email címe) és a címzettet (a vásárló email címe a sessionből).
    - Beállítja a tárgyat ("Rendelés visszaigazolás").
    - HTML formátumú email törzset hoz létre, amely tartalmazza a rendelés alapadatait (ID, összeg, szállítási cím, fizetési/szállítási mód) a függvénynek átadott $orderDetails tömbből.
    - Elküldi az emailt.
    - Visszaad egy siker- vagy hibaüzenetet.
  + Létrehozza az $orderDetails tömböt a kosar.php kontextusában lévő változókból (rendelés ID, végösszeg, szállítási cím a sessionből, szállítási/fizetési mód).
  + Meghívja a sendOrderConfirmation függvényt a vásárló email címével és az $orderDetails tömbbel.
* **Megjegyzés:** A fájl neve a kódban (kosar.php) rendeles\_visszaigazolo\_mailer.php-ként szerepel, de a megadott fájlnév rendeles\_visszaigazolas.php. Feltételezzük, hogy ez ugyanaz a fájl.

**sql\_fuggvenyek.php**

* **Leírás:** Két segédfüggvényt tartalmaz az adatbázis-műveletek egyszerűsítésére, **mysqli** kiterjesztést használva.
* **Függőségek:** Nincs.
* **Működés:**
  + adatokLekerdezese($muvelet):
    - Létrehoz egy új mysqli kapcsolatot a halaliweb adatbázishoz (a kapcsolat adatai fixen vannak beégetve).
    - Ellenőrzi a kapcsolat sikerességét.
    - Végrehajtja a paraméterként kapott SQL lekérdezést ($muvelet).
    - Ellenőrzi a lekérdezés sikerességét.
    - Ha sikeres és van találat (num\_rows != 0), visszaadja az összes sort asszociatív tömbként (fetch\_all(MYSQLI\_ASSOC)).
    - Ha sikeres, de nincs találat, a 'Nincs találat!' stringet adja vissza.
    - Hiba esetén a mysqli hibaüzenetét adja vissza.
  + adatokValtoztatasa($muvelet):
    - Létrehoz egy új mysqli kapcsolatot.
    - Ellenőrzi a kapcsolat sikerességét.
    - Végrehajtja a paraméterként kapott SQL módosító utasítást ($muvelet, pl. INSERT, UPDATE, DELETE).
    - Ellenőrzi a művelet sikerességét.
    - Ha sikeres és volt érintett sor (affected\_rows > 0), a 'Sikeres művelet!' stringet adja vissza.
    - Ha sikeres, de nem volt érintett sor (affected\_rows == 0), a 'Sikertelen művelet!' stringet adja vissza.
    - Hiba esetén a mysqli hibaüzenetét adja vissza.
* **Megjegyzés:** Ez a fájl a mysqli kiterjesztést használja, míg az alkalmazás többi része (ahol a db.php-t include-olják) a PDO-t. Ez jelentős inkonzisztencia és kerülendő gyakorlat. A kapcsolat adatai ismétlődnek itt és a db.php-ban.

**termek\_reszletek.php**

* **Leírás:** Egy adott termék részletes adatlapját jeleníti meg.
* **Függőségek:**
  + db.php: Adatbázis kapcsolat.
  + nav.php: Navigációs menü.
  + footer.php: Lábléc.
  + ./js/termek\_reszletek.js: Az adatok lekérését és megjelenítését végző JavaScript.
  + ./style/style.css: Stíluslap.
* **Működés:**
  + Elindítja a munkamenetet, beállítja a cache kontrollt, include-olja az adatbázist, navigációt, láblécet.
  + HTML struktúrát biztosít a termékadatok megjelenítéséhez (egy reszletek ID-jű div).
  + Beinclude-olja a termek\_reszletek.js fájlt, amely:
    - Kiolvassa a termek URL paramétert (ami a termék urlnev-e).
    - AJAX POST kérést küld a termekek\_adatok.php/termek\_url végpontra a urlnev-vel.
    - A kapott JSON adat (a termék adatai) alapján dinamikusan feltölti a #reszletek divet a termék nevével, képével, gyártójával, típusával, raktárkészletével, árával (akciósan is), leírásával és egy kosárba helyező űrlappal (mennyiség választóval).

**termekek\_adatok.php**

* **Leírás:** API végpontként szolgál a termekek.php és termek\_reszletek.php oldalak számára, hogy termékadatokat, kategóriákat, gyártókat és árinformációkat kérdezzen le.
* **Függőségek:**
  + sql\_fuggvenyek.php: Adatbázis segédfüggvények (mysqli).
* **Működés:**
  + Include-olja az sql\_fuggvenyek.php-t.
  + Feldarabolja a kérés URI-ját.
  + A kért erőforrás (switch) és a HTTP metódus alapján hajt végre műveleteket (mindegyik a mysqli függvényeket használja):
    - **GET /kategoriakleker**: Lekérdezi az összes kategóriát (kategoria tábla). JSON választ ad.
    - **GET /gyartokleker**: Lekérdezi az összes egyedi gyártót a termek táblából. JSON választ ad.
    - **POST /termek**: Lekérdezi egy termék adatait ID alapján (a POST body-ban kapott ID). JSON választ ad.
    - **POST /termek\_url**: Lekérdezi egy termék adatait urlnev alapján (a POST body-ban kapott urlnev). JSON választ ad.
    - **GET /arakleker**: Lekérdezi a minimális és maximális termékárat a termek táblából. JSON választ ad.
    - **POST /szures**:
      * Lekéri a szűrési feltételeket (kategória, gyártó, min/max ár, keresőszó) és a rendezési módot a POST body-ból.
      * Dinamikusan összeállít egy SQL lekérdezést a termek és kategoria táblák összekapcsolásával.
      * Hozzáadja a WHERE feltételeket a kapott paraméterek alapján (ha nem üresek vagy alapértelmezettek).
      * Hozzáadja az ORDER BY feltételt a rendezési mód alapján. Speciális feltételt alkalmaz az 'akcio' rendezésre (csak akciós, készleten lévő termékek).
      * Végrehajtja a lekérdezést.
      * Visszaadja a szűrt és rendezett termékek listáját JSON tömbként (vagy üres tömböt, ha nincs találat).
    - **Egyéb kérések**: Hiba JSON-t és 400-as státuszkódot küld.
* **Megjegyzés:** Csak mysqli-t használ az sql\_fuggvenyek.php-n keresztül.

**termekek.php**

* **Leírás:** Megjeleníti a webshop termékeinek listáját, szűrési és rendezési lehetőségekkel.
* **Függőségek:**
  + db.php: Adatbázis kapcsolat (PDO).
  + nav.php: Navigációs menü.
  + footer.php: Lábléc.
  + ./js/termekek.js: A szűrők feltöltését, termékek lekérését és megjelenítését végző JavaScript.
  + ./style/style.css: Stíluslap.
* **Működés:**
  + Elindítja a munkamenetet, beállítja a cache kontrollt, include-olja az adatbázist (PDO).
  + **Fontos:** Az oldal betöltésekor **futtat egy SQL UPDATE parancsot (PDO)**, amely nullázza az akcios\_ar, akcio\_kezdete, akcio\_vege mezőket azoknál a termékeknél, ahol az akció vége a múltban van, vagy ahol nincs kezdeti és végdátum sem megadva. Ezzel biztosítja, hogy csak az érvényes akciók jelenjenek meg.
  + Include-olja a navigációt és a láblécet.
  + HTML struktúrát biztosít a szűrő elemeknek (kategória select, gyártó select, ár range csúszkák, kereső input, rendezés select). Ezek egy szures\_ ID-jű divben vannak, amelyre a JS figyel.
  + HTML helyőrzőket biztosít a találatok számának (talalatokSzama), a termékek listájának (termekekTartalom), és az esetleges "nincs találat" üzenetnek/képnek (valaszSzoveg, hibaKepDiv).
  + Beinclude-olja a termekek.js fájlt, amely:
    - Betöltéskor lekéri és feltölti a kategória és gyártó legördülő menüket, valamint beállítja az ár csúszkákat a termekek\_adatok.php végpontok segítségével.
    - Betöltéskor lekéri és megjeleníti az összes terméket.
    - Figyeli a szűrő elemek változását (input esemény a #szures\_ diven), és minden változáskor újra lekéri és megjeleníti a termékeket a termekek\_adatok.php/szures végpont segítségével, átadva az aktuális szűrő/rendezési beállításokat.

**2. Kliensoldali Fájlok (JavaScript)**

**kalkulator.js**

* **Leírás:** A kalkulator.php oldal funkcionalitását biztosítja.
* **Függőségek:** A kalkulator.php HTML struktúrája (ID-k).
* **Működés:**
  + OldalBetolt(): Az oldal betöltésekor alapértelmezetten a feszültségesés kalkulátort jeleníti meg, a többit elrejti.
  + kalkValaszt(): Figyeli a rádiógombok változását, és megjeleníti a kiválasztott kalkulátor divjét, a többit elrejti. (Az árkalkulátor logikája hiányzik).
  + Szamolas(): A "Feszültségesés kalkulátor" számítás gombjára kattintva fut le.
    - Beolvassa a beviteli mezők értékeit (hossz, keresztmetszet, áram).
    - Ellenőrzi, hogy a fázis rádiógomb ki van-e választva és az értékek ki vannak-e töltve, pozitívak-e. Hibaüzenetet jelenít meg a #hiba divben.
    - A kiválasztott fázis (1 vagy 3) és keresztmetszet alapján kiválasztja a megfelelő impedancia (Z) értéket (ezek fixen vannak beégetve).
    - Kiszámolja a feszültségesést (V) és a százalékos értéket a megfelelő képlettel (1 vagy 3 fázisra).
    - Megjeleníti az eredményt a #eredmeny divben, két tizedesjegyre kerekítve.
    - Az eredményt a százalékos érték alapján színezi (zöld < 3%, narancs 3-7%, piros > 7%).
  + ellenallas\_szamolas(): Az "Ellenállás kalkulátor" számítás gombjára kattintva fut le.
    - Beolvassa a feszültséget és áramot.
    - Ellenőrzi, hogy ki vannak-e töltve a mezők. Hibaüzenetet jelenít meg a #hiba\_ellenallas divben.
    - Kiszámolja az ellenállást (R = V / I).
    - Megjeleníti az eredményt a #eredmeny\_ellenallas divben "Ω" jellel, zöld színnel, nagy méretben.
  + Eseményfigyelőket rendel a számítás gombokhoz (click) és az oldalhoz (load, change).

**rendeles\_kovetes.js**

* **Leírás:** A rendeles\_kovetes.php oldal dinamikus működését biztosítja AJAX segítségével.
* **Függőségek:** A rendeles\_kovetes.php HTML struktúrája (ID-k), rendeles\_kovetes\_fetch.php API végpont.
* **Működés:**
  + rendelesek(): Aszinkron függvény, amely az oldal betöltésekor lefut.
    - fetch segítségével lekéri a rendelési listát a rendeles\_kovetes\_fetch.php/rendelesek GET végpontról.
    - Meghívja a print\_() függvényt az eredménnyel és a státuszkóddal.
  + print\_(eredmeny, FetchStatus): Megjeleníti a rendelési listát.
    - Ha a státuszkód 400 (vagy hiba történt), hibaüzenetet és képet jelenít meg.
    - Ha sikeres, inicializálja az állapotsávot (rendelesTable).
    - Végigmegy a kapott eredmeny tömbön (rendelések).
    - Minden rendeléshez létrehoz egy kártyát (card) a #rendelesek\_ divben, amely megjeleníti az ID-t, dátumot, végösszeget és státuszt. A kártya hátterét a státusz alapján színezi.
    - A kártya egy kattintható <button> elem, amely meghívja a rendeles\_allapot() függvényt a rendelés ID-jével és státuszával.
    - Számolja és kiírja a rendelések számát (rendeles\_db).
  + rendeles\_allapot(rend\_id, allapot): Egy rendeléskártyára kattintva fut le.
    - Frissíti az állapotsávot (rendelesTable) úgy, hogy a kiválasztott rendelés státuszát tükrözze (színes sávok).
    - Frissíti az oldal fejlécét (rendelesCount), hogy megjelenítse a kiválasztott rendelés ID-ját, és hozzáad "Részletek" és "Vissza" gombokat.
  + rendeles\_reszletek(rendeles\_statusz, rendeles\_id): A "Részletek" gombra kattintva fut le.
    - Aszinkron fetch POST kérést küld a rendeles\_kovetes\_fetch.php/rendeles\_reszletek végpontra a rendelés ID-jével és státuszával.
    - Meghívja a Rendeles\_adatok\_megjelenit() függvényt a kapott részletes adatokkal.
  + Rendeles\_adatok\_megjelenit(rendReszletek): Megjeleníti a kiválasztott rendelés részleteit.
    - Kiüríti a rendelési lista helyét (rendelesek\_).
    - Létrehoz egy táblázatot a rendelés tételeiről (kép, név, mennyiség, ár, összesen).
    - Létrehoz egy másik táblázatot a rendelés általános adatairól (státusz, dátum, szállítás, fizetés, végösszeg).
  + vissza(): A "Vissza" gombra kattintva fut le.
    - Kiüríti a részletek nézetet (rendelesek\_).
    - Meghívja újra a rendelesek() függvényt, hogy visszatérjen a rendelési lista nézethez.
  + Eseményfigyelőt rendel az oldal betöltéséhez (load), ami elindítja a rendelesek() lekérdezést.

**termek\_reszletek.js**

* **Leírás:** A termek\_reszletek.php oldal működését biztosítja, lekéri és megjeleníti az adott termék adatait.
* **Függőségek:** A termek\_reszletek.php HTML struktúrája (ID: reszletek), termekek\_adatok.php API végpont.
* **Működés:**
  + termek\_reszletei(): Aszinkron függvény, amely az oldal betöltésekor lefut.
    - Kiolvassa a termek nevű URL paraméter értékét (urlnev).
    - fetch POST kérést küld a termekek\_adatok.php/termek\_url végpontra a urlnev-vel.
    - A kapott JSON (termékadatok tömbje) alapján:
      * Generálja a HTML kódot a #reszletek divbe.
      * Megjeleníti a termék nevét, képét, gyártóját, típusát.
      * Kiszámolja és megjeleníti az akciós százalékot, ha van (akcioSzazalek).
      * Megjeleníti az árat (áthúzva a régit, pirossal az akcióst, ha van) (akciosAr).
      * Megjeleníti a raktárkészletet (raktaronSzoveg), zölddel, ha van, pirossal, ha nincs.
      * Megjeleníti a termék leírását.
      * Létrehoz egy form-ot, amely a kosar\_muveletek.php-nak küld adatokat.
      * Tartalmaz rejtett mezőket (termek\_id, kep, ar).
      * Tartalmaz egy mennyiség beviteli mezőt (input type="number") +/- gombokkal. A gombok a mennyisegValtoztat() függvényt hívják. A max attribútum a raktárkészlethez van igazítva.
      * Tartalmaz egy "Kosárba" gombot (add\_to\_cart), amely le van tiltva (disabled), ha a termék nincs készleten.
  + mennyisegValtoztat(valtozas, maxErtek, inputId): A +/- gombokra kattintva módosítja a megadott inputId-jű beviteli mező értékét a valtozas értékével (+1 vagy -1). Biztosítja, hogy az érték 1 és a maxErtek (raktárkészlet) között maradjon.
  + Eseményfigyelőt rendel az oldal betöltéséhez (load), ami elindítja a termek\_reszletei() lekérdezést.

**termekek.js**

* **Leírás:** A termekek.php oldal dinamikus működését biztosítja: szűrők feltöltése, termékek lekérése és megjelenítése AJAX segítségével.
* **Függőségek:** A termekek.php HTML struktúrája (ID-k), termekek\_adatok.php API végpont.
* **Működés:**
  + **Szűrő Feltöltő Függvények:**
    - kategoriakFeltolt(adatok): Feltölti a #kategoriaSzures select elemet a kapott kategória adatokkal.
    - gyartokFeltolt(gyartok): Feltölti a #gyartoSzures select elemet a kapott gyártó nevekkel.
    - arakFeltolt(arak): Beállítja a #minRangeAr és #maxRangeAr range csúszkák min, max és value értékeit a kapott minimális/maximális árak alapján, és megjeleníti az aktuális értéket mellettük.
  + **Adat Lekérő Függvények:**
    - kategoriakLeker(): Aszinkron, fetch-csel lekéri a kategóriákat a termekek\_adatok.php/kategoriakleker-ről, majd meghívja a kategoriakFeltolt-ot.
    - gyartokLeker(): Aszinkron, fetch-csel lekéri a gyártókat a termekek\_adatok.php/gyartokleker-ről, majd meghívja a gyartokFeltolt-ot.
    - arakLeker(): Aszinkron, fetch-csel lekéri a min/max árakat a termekek\_adatok.php/arakleker-ről, majd meghívja az arakFeltolt-ot.
    - termekekLeker(): Aszinkron, ez fut le a szűrők változásakor.
      * Beolvassa az összes szűrő és rendezési beállítás aktuális értékét.
      * Összeállítja a bodyAdatok objektumot ezekkel az értékekkel.
      * fetch POST kérést küld a termekek\_adatok.php/szures végpontra a bodyAdatok-kal.
      * A kapott JSON (terméklista) alapján meghívja a termékeket megjelenítő logikát (lásd osszesTermekekLeker leírását).
    - osszesTermekekLeker(): Aszinkron, ez fut le az oldal betöltésekor.
      * Hasonló a termekekLeker-hez, de fix, alapértelmezett szűrőbeállításokkal hívja a /szures végpontot.
      * **Termék Megjelenítés (közös logika a termekekLeker-rel):**
        + Kiüríti a #termekekTartalom divet.
        + Ha a kapott terméklista üres, hibaüzenetet és képet jelenít meg.
        + Ha vannak termékek:

Végigmegy a listán.

Minden termékhez létrehoz egy Bootstrap kártyát (card).

Megjeleníti a termék képét (kattintható, a termekMegnyit függvényt hívja). Ha akciós, egy piros -X% jelvényt helyez el a képen.

Megjeleníti a termék nevét.

Megjeleníti az árat: Normál ár, vagy áthúzott normál + piros akciós ár. Ha nincs készleten, piros üzenetet ír ki ár helyett. Kiírja a raktárkészletet.

Tartalmaz egy "Részletek" gombot (termekMegnyit hívása).

Tartalmaz egy kosárba helyező form-ot (kosar\_muveletek.php cél) rejtett mezőkkel (ID, kép, ár) és egy mennyiségválasztóval (+/- gombokkal, mennyisegValtoztat hívása) és egy "Kosárba" gombbal (letiltva, ha nincs készleten).

* + termekMegnyit(urlnev): Átirányítja a böngészőt a termek\_reszletek.php?termek=<urlnev> oldalra.
  + mennyisegValtoztat(valtozas, maxErtek, inputId): Ugyanaz, mint a termek\_reszletek.js-ben.
  + **Eseménykezelés:**
    - window.addEventListener("load", ...): Az oldal betöltésekor meghívja a szűrőket feltöltő és az összes terméket lekérő függvényeket.
    - document.getElementById("szures\_").addEventListener("input", termekekLeker): Figyeli a szűrőket tartalmazó div (#szures\_) bármelyik input eseményét (pl. select változás, input gépelés, range csúszka mozgatás), és ilyenkor meghívja a termekekLeker függvényt, hogy frissítse a terméklistát.

**3. Stíluslap (CSS)**

**style.css**

* **Leírás:** Az alkalmazás egyedi megjelenését definiáló CSS szabályokat tartalmazza. Kiegészíti vagy felülírja a Bootstrap stílusait.
* **Működés:**
  + Beállítja az alapvető body stílusokat (háttér, szín, betűtípus, padding).
  + Stílusokat definiál a navigációs sáv (navbar) elemeihez (logó mérete, linkek színe, hover effektusok, ikonok mérete, reszponzív viselkedés, hamburger menü animációja).
  + Megadja a kiválasztás (::selection) stílusát.
  + Stílusokat ad a kalkulátor (kalkulator.php) elemeihez (eredmény megjelenítése).
  + Általános konténer és kártya (card) stílusok.
  + Űrlap elemek (form-label, form-control) és fókusz állapotuk stílusa.
  + Különböző div blokkok (pl. bemutatkozasDiv, termekekDiv a főoldalon) háttérszínét és paddingját állítja be.
  + Gombok (navbarGomb, szamitasBtn, kapcsolatBtn, termekekKartyaGomb, profilMentGomb, stb.) egységes megjelenését és hover effektusait definiálja.
  + A lábléc (footer) stílusát (háttér, szín, linkek).
  + Termék kártyák (termekek.php) elemeinek stílusa (kép mérete, középre igazítás).
  + Kosár (kosar.php) elemeinek stílusa (ár, felirat mérete, kuka ikon).
  + Kapcsolat (kapcsolat.php) oldal közösségi média ikonjainak színe.
  + A kosárszámláló jelvény (#cart-count) pozícióját és megjelenését.
  + Profil (profil.php) oldal kártyáinak fejléc hátterét (profilBG).
  + Mennyiség növelő/csökkentő gombok (mpGomb) stílusa.
  + Ár csúszkák (range) megjelenésének testreszabása.
  + Google reCAPTCHA elem középre igazítása.
  + Reszponzív szabályok (pl. lábléc kép elrejtése kisebb képernyőn).

Ez a dokumentáció lefedi a megadott fájlok főbb funkcióit és működését. Fontos megjegyezni a PDO és mysqli keveréséből adódó inkonzisztenciát és a kosárkezelésben észlelt potenciális hibákat.