

Bevezető

A projekt célja egy egyszerű, de funkcionális online ruhaáruház létrehozása, ahol a felhasználók böngészhetnek a különböző ruházati termékek között, termékeket tehetnek a kosarukba, és végrehajthatják a vásárlást. A webshopban különböző kategóriák szerint lehet keresni, mint például férfi és női ruházat, biztosítva a leginkább megfelelő termékek megtalálását.

A webshop fejlesztése során fontos szempont volt a felhasználóbarát felület, valamint a zökkenőmentes vásárlási élmény biztosítása. Az alkalmazás web alapú, és a projekt célja a webshop működésének és a felhasználói interakcióknak a bemutatása.

Technikai követelmények

A projekt működéséhez az alábbi technikai követelmények szükségesek:

- **Adatbázis-kezelő:** MySQL (egy rtmstoredb nevű adatbázisba be kell importálni a rtmstoredb.sql nevű fájlt)
- **Backend technológia:** PHP
- **Frontend technológia:** HTML, CSS, Bootstrap, JavaScript
- **Fejlesztői környezet:** A projektet a helyi gépen egy web szerver környezetben kell futtatni (pl. XAMPP, WAMP, MAMP)
- **Internetkapcsolat:** Mivel webes alkalmazásról van szó, internetkapcsolat szükséges a működéshez
- **API:** Postman az api teszteléséhez

Működés leírása

A webshop működése a következőképpen néz ki:

Főoldal: A felhasználó ezen az oldalon egy modern, képekkel ellátott **karusszel (carousel)** segítségével böngészhet különféle **férfi és női ruházati termékeket**, például kabátokat, pulóvereket, ingeket, pólókat, nadrágokat és rövidnadrágokat.

Minden dián egy **"Vásárlás" gomb** található, amely egy adott kategória oldalára vezeti a látogatót (pl. férfi pulóverek vagy női nadrágok).

A karusszel alatt két nagy kártya is megjelenik:

- az egyik a **férfi ruhák** oldalára visz,
- a másik pedig a **női ruhák** oldalára.

Az oldal alsó részén megjelennek a **legújabb férfi és női termékek**, ezzel segítve a vásárlókat az újdonságok felfedezésében.

Termékoldal: Ezen az oldalon a felhasználó egy kiválasztott termék részletes adatlapját tekintheti meg. A termékhez tartozó képeket egy lapozható karusszelben nézheti végig, amelyeket kis bélyegképeken keresztül is könnyedén elérhet. Az oldalon megjelenik a termék neve, ára, valamint az elérhető méretek és a hozzájuk tartozó raktárkészlet. A felhasználó kiválaszthatja a kívánt méretet és megadhatja, hány darabot szeretne vásárolni. Amennyiben be van jelentkezve, a „Kosárba tesz” gomb segítségével hozzáadhatja a terméket a kosarához. Ha még nem regisztrált, akkor egy gombon keresztül fiók létrehozására kap lehetőséget. A rendszer hibajelzéssel figyelmezteti, ha túl sok darabot próbálna rendelni a választott méretből, vagy ha nem adott meg mennyiséget, míg sikeres művelet esetén megerősítő üzenet jelenik meg.

Kosár: Ez az oldal a felhasználó kosarának tartalmát jeleníti meg. A belépett felhasználó itt áttekintheti, hogy milyen termékeket helyezett a kosarába, beleértve azok nevét, árát, kiválasztott méretét és mennyiségét. A termékek képekkel együtt jelennek meg, így vizuálisan is könnyen beazonosíthatók. Minden egyes tétel mellett lehetőség van annak eltávolítására is. Az oldal jobb oldalán összegzés látható a teljes vásárlási összegről, és ha a kosár nem üres, a felhasználó egy gombnyomással tovább léphet a fizetési folyamatra. Ha a kosár üres, erről egy figyelmeztető üzenet tájékoztatja. Az oldal kizárólag bejelentkezett felhasználók számára elérhető.

Vásárlás: Ez az oldal a megrendelés véglegesítésére szolgál. A felhasználó itt adhatja meg a szállítási adatait, és megadhat egy kuponkódot, amely kedvezményt biztosít. Az oldal kiszámítja a kosár összegét, levonja belőle az esetleges kedvezményt, és megjeleníti a végösszeget. Sikeres megrendelés után a rendszer elmenti az adatokat, majd a felhasználót átirányítja a „Köszönjük” oldalra.

RESTful API – Regisztráció, Bejelentkezés Működése

1. Regisztráció menete:

1. **Felhasználó kitölti a regisztrációs űrlapot**
Felhasználónév, email, jelszó, jelszó megerősítése.
2. **Űrlap elküldi az adatokat JavaScript-tel a fetch segítségével**
POST kérés megy az api.php?endpoint=register címre JSON-ben.
3. **API oldalon ellenőrzés történik:**
 - Van-e hiányzó mező?
 - Email formátum jó-e?
 - Jelszó elég hosszú-e?
 - Két jelszó megegyezik-e?
4. **Ha minden rendben van:**
 - Jelszó titkosítása (password_hash)
 - Mentés adatbázisba RegisterUser eljárással
5. **Visszajelzés:**
 - Siker: átirányítás főoldalra
 - Hiba: hibaüzenetek megjelennek az űrlapon

2. Bejelentkezés menete:

1. **Felhasználó megadja emailt és jelszót a bejelentkező űrlapon**
2. **JavaScript fetch-sel elküldi az adatokat**
→ POST kérés az api.php?endpoint=login címre
3. **API oldalon történik:**
 - Lekérdezés az adatbázisból (LoginUser eljárás) az email alapján
 - Ellenőrzi, hogy van-e ilyen felhasználó
4. **Ha talál egyezést:**
 - A jelszót összehasonlítja a tárolt titkosított jelszóval (password_verify)
 - Ha egyezik:
 - Session beállítása felhasználó ID, név, admin státusszal
 - Sikeres válasz vissza adása
5. **Visszajelzés:**
 - Siker: átirányítás a főoldalra
 - Hiba: hibaüzenet jelenik meg az űrlapon

Az API-hoz tartozó tesztek megtalálhatók a **postman_collection.json** fájlban. A fájlt egyszerűen be lehet importálni a Postman-be.

Admin Panel

Az admin felület új termék hozzáadására szolgál. Csak bejelentkezett admin használhatja. A termék nevét, árát, képeit, méretenkénti mennyiségeit, nemet és kategóriát kell megadni. A képek feltöltődnek, az adatok pedig elmentődnek az adatbázisba egy InsertProduct eljárással.

Új Termék Hozzáadása:

- Az admin a "Termék hozzáadása" gomb segítségével új terméket vihet fel a rendszerbe.
- A termékek adatainak megadásához szükséges mezők a következők:
 - **Termék neve**
 - **Kategória**
 - **Ár**
 - **Kép feltöltése** (4 darab termékhez tartozó kép feltöltése)
 - **Elérhető méretek** (pl. S, M, L, XL)
 - **Készlet mennyiség** (a termék elérhető mennyisége)

A termékek megjelenítésének folyamata az oldalon

Az oldalon található ruházati termékek egységes szerkezetben, dinamikusan kerülnek megjelenítésre. Az alábbi lépések mutatják be a megjelenítés működését:

1. Termékadatok lekérdezése

A termékek adatai – beleértve a nevet, árát, képeket és azonosítót – egy külön PHP fájl (pl. fetch_product_datas.php) segítségével kerülnek lekérésre az adatbázisból. Az adatok egy asszociatív tömb formájában kerülnek visszaadásra.

2. Kártyakomponens generálása

A generate_product_card.php fájlban található generateProductCard() függvény minden egyes termékhez HTML-alapú, Bootstrap kártyát generál. A kártya tartalmaz:

- egy képgalériát (carousel), amely maximum négy képet jelenít meg a termékről,
- a termék nevét és árát,
- egy "Megtekintés" feliratú gombot,
- valamint az egész kártya egy linkként működik, amely a termék egyedi aloldalára (productpage.php) navigál.

3. Termékek megjelenítése az oldalon

A generált kártyák HTML elemekként kerülnek elhelyezésre egy rácsszerű

elrendezésben (row), amely reszponzív megjelenítést biztosít különböző képernyőméretek esetén is. A megjelenítés egy dedikált fájl (pl. `men_products.php`) segítségével történik, amely beilleszti a termékkártyákat az adott kategóriaoldalba.

4. Oldalszerkezet

A kategóriaoldalak (például: `manpage.php`) egységes struktúrát követnek:

- A fejléc alatt egy figyelemfelkeltő banner jelenik meg az adott kategóriához igazítva.
- Ezt követi egy vízszintesen görgethető menüsáv, amely az alkategóriák közti navigációt segíti.
- A termékek ezután kártyás elrendezésben jelennek meg.

Adatbázis

Táblák:

products

- Termékek tárolása (név, ár, képek, készlet méretenként, nem, kategória, létrehozás dátuma)
- Méretek: xs, s, m, l, xl
- Kategóriák: pl. Kabátok & Pulóverek, Ingek & Pólók

cart

- Kosár tartalma: melyik felhasználónál milyen termék, milyen méretben és mennyiségben van a kosárban

orders

- Rendelések fő adatai: felhasználó, szállítási adatok, végösszeg, kuponkód, dátum stb.

order_items

- Rendelések részletezése termékszinten: mit vettek, hány darabot, milyen áron

users

- Felhasználók adatai: név, email, jelszó, admin státusz

coupons

- Kuponkódok és azok kedvezménye

Tárolt eljárások

- **Kosár műveletek**
 - AddToCart(user_id, product_id, quantity, size)
 - RemoveFromCart(cart_id)
 - GetUserCart(user_id)
- **Terméklekérdezések**
 - Különböző eljárások férfi/női és kategória szerinti terméklistázásra, pl. GetMenProducts(), GetWomenBluzokPolok()
 - GetProductById(product_id) – termék lekérdezése azonosító alapján
 - GetNewestMenProducts() / GetNewestWomenProducts() – legújabb 4 termék
- **Rendelés és fizetés**
 - PlaceOrder(...) – új rendelés leadása, ár kiszámítás, készlet csökkentése, kosár ürítése
 - ValidateCoupon(code) – kupon ellenőrzés
- **Felhasználói műveletek**
 - RegisterUser(username, email, password)
 - LoginUser(email) – visszaadja a felhasználó adatait bejelentkezéshez
- **Adminhoz hasznos**
 - InsertProduct(...) – új termék rögzítése

Befejezés

Ez a projekt egy teljes funkcionalitású, webalapú ruha webshop, amelyben képes kezelni a termékek listáját, a vásárlói kosarat és az adminisztrációs felületet. A rendszer lehetőséget biztosít a felhasználók számára, hogy kényelmesen vásárolhassanak, miközben az adminisztrátorok egyszerűen kezelhetik a webshop termékeit, az árakat, és a készletet.

A projekt során a felhasználói élményre és a könnyű kezelhetőségre helyeztük a hangsúlyt, mind a vásárlói, mind az adminisztrátori oldal szempontjából. A webáruház dinamikus és intuitív felületet kínál, miközben biztonságos és megbízható háttérrel biztosít a termékek kezelésére.

A projekt során elsajátítottuk a frontend és backend technológiák integrálását, a felhasználói interakciók kezelését, valamint az adminisztrációs felület megvalósítását, amely lehetővé teszi a webshop folyamatos karbantartását és frissítését.

Ez a projekt kiváló lehetőséget adott arra, hogy gyakorlatban is alkalmazhassuk az informatika és a webfejlesztés különböző területein megszerzett tudásunkat. A fejlesztés

során szerzett tapasztalatok hasznosak lesznek a jövőbeli projektekhez és a webes alkalmazások fejlesztéséhez.