

Magyar Fight Shop

1 Az Oldal és Funkciói

A projekt célja egy teljes körű nyilvántartó rendszer létrehozása, amely lehetővé teszi a felhasználók számára, hogy termékeket nyilvántartsanak, kezeljenek, és vásároljanak egy online áruházon keresztül. Az oldal fő funkciói közé tartozik a felhasználók bejelentkezése, regisztrációja, termékek megtekintése, hozzáadása, kosárba helyezése és a kosár kezelése. Az alkalmazás technológiai stackje MongoDB, Express.js és Vue.js kombinációjára épül, ami modern és skálázható megoldást biztosít.

2 Fő Funkciók

2.1 Felhasználói Regisztráció és Bejelentkezés

- A felhasználók regisztrálhatnak az oldalon egyedi felhasználónév, email és jelszó megadásával.
- Bejelentkezés után a felhasználók hozzáférhetnek a teljes funkcionalitáshoz, beleértve a termékek böngészését és a kosár kezelését.

2.2 Termékek Nyilvántartása és Kezelése

- A felhasználók megtekinthetik a termékek listáját, amely tartalmazza a termék nevét, leírását, árát és képét.
- Új termékek hozzáadása a termék nevén, leírásán, árán és képén alapul. Az új termék hozzáadása előtt az adatok validálása történik.

2.3 Kosár Funkció

- A bejelentkezett felhasználók hozzáadhatják a termékeket a kosárhoz, megtekinthetik a kosár tartalmát, és eltávolíthatják a termékeket a kosárból.

- A kosár kezelése magában foglalja a termékek hozzáadását és eltávolítását, valamint a kosár tartalmának megtekintését.

3 Fájlok és Céljaik

3.1 index.js

- **Szerver Beállítása:** Az Express.js keretrendszer használata az alkalmazás alapját képezi. A szerver beállítása, port konfigurálása és a middleware-ek betöltése itt történik.
- **Adatbázis Kapcsolat:** A MongoDB adatbázissal való kapcsolatot állítja be, és az adatbázis elérését az alkalmazás többi részéhez biztosítja.
- **Middleware-ek Betöltése:** Betölti a különböző middleware-eket, beleértve a logger-t és az API-kat (users, items, carts).

3.2 createdb.js

- **Adatbázis Inicializálás:** Kapcsolódik a MongoDB-hez, létrehozza a szükséges adatbázisokat és gyűjteményeket (users, items), és néhány előre definiált terméket szűr be az items gyűjteménybe.

3.3 Middleware Fájlok

- **middleware/api/carts.js**
 - **Kosár Kezelése:** Ez a fájl kezeli a kosárral kapcsolatos műveleteket, mint például az elemek hozzáadása, eltávolítása és a kosár tartalmának lekérdezése.
- **middleware/api/items.js**
 - **Termékek Kezelése:** Ez a fájl kezeli a termékekkel kapcsolatos műveleteket, mint például a termékek listázása és új termékek hozzáadása.
 - **Adatellenőrzés:** Az új termék hozzáadásakor ellenőrzi, hogy az adatok helyesek-e és a termék már létezik-e.
- **middleware/api/users.js**

- **Felhasználók Kezelése:** Ez a fájl kezeli a felhasználókhoz kapcsolódó műveleteket, mint például a regisztráció és bejelentkezés.
- **middleware/logger.js**
 - **Naplózás:** Ez a middleware naplózza az összes bejövő HTTP kérést, amely segíthet a hibakeresésben és az alkalmazás monitorozásában.

3.4 Model Fájlok

- **models/cart.js**
 - **Cart Schema:** Meghatározza a kosár elemek sémáját, beleértve a termék nevét, leírását, árát, kép URL-jét és a felhasználó azonosítóját.
- **models/item.js**
 - **Item Schema:** Meghatározza a termékek sémáját, beleértve a termék nevét, gyártóját, kategóriáját, árát, leírását és kép URL-jét.
- **models/user.js**
 - **User Schema:** Meghatározza a felhasználók sémáját, beleértve a felhasználónevet, jelszót, email címet, születési dátumot és a felhasználó kosarának tartalmát.

3.5 Public Mappa

- **HTML Fájlok**
 - **index.html:** Az alkalmazás főoldala, amely üdvözlí a felhasználókat és navigációs lehetőségeket biztosít.
 - **login.html:** A bejelentkező oldal, ahol a felhasználók megadhatják felhasználónevüket és jelszavukat a bejelentkezéshez.
 - **register.html:** A regisztrációs oldal, ahol az új felhasználók regisztrálhatnak.
 - **webshop.html:** A termékek listázásának oldala, ahol a felhasználók megtekinthetik a termékeket és hozzáadhatják őket a kosárhoz.
 - **cart.html:** A kosár oldal, ahol a felhasználók megtekinthetik és kezelhetik kosaruk tartalmát.

- **sikerreg.html**: A sikeres regisztrációt követő oldal.
- **hiba.html**: A hibaoldal, amely a sikertelen bejelentkezés vagy regisztráció után jelenik meg.
- **CSS Fájlok**
 - **style.css**: Az oldal stílusait tartalmazza, beleértve az alapvető layout-okat, színeket és tipográfiát.
- **JavaScript Fájlok**
 - **main.js**: A frontend logika Vue.js segítségével, amely a felhasználói interakciókat kezeli, például a termékek hozzáadása a kosárhoz és az adatok mentése.
 - **webshop.js**: A termékek listázásához kapcsolódó logika, beleértve az API hívásokat a termékek lekérdezéséhez.

4 Továbbfejlesztési Lehetőségek

4.1 Frontend Fejlesztések

- **További Interakciók**: Drag-and-drop funkciók a termékek kosárba helyezéséhez, dinamikus keresés és szűrés a termékek között.

4.2 Backend Fejlesztések

- **Adatellenőrzés Bővítése**: Részletesebb adatellenőrzés a termékek és felhasználók hozzáadásakor.
- **Skálázhatóság és Teljesítmény Javítása**: Adatbázis indexek létrehozása a gyorsabb lekérdezésekhez, és a szerver skálázhatóságának javítása.

4.3 Biztonság

- **Kétlépcsős Azonosítás (2FA)**: További biztonsági réteg bevezetése a felhasználói bejelentkezésekhez.
- **Jelszó Visszaállítás**: Biztonságos jelszó visszaállítási mechanizmus létrehozása email validációval.

4.4 Felhasználói Funkciók

- **Felhasználói Profilok:** Részletes felhasználói profilok, ahol a felhasználók szerkeszthetik adataikat és nyomon követhetik rendeléseiket.
- **Termék Értékelés és Véleményezés:** Lehetőség a felhasználók számára, hogy értékeljék és véleményezzék a termékeket.

4.5 Adminisztrációs Funkciók

- **Admin Felület:** Külön adminisztrációs felület létrehozása, ahol az adminisztrátorok kezelhetik a termékeket, felhasználókat és rendeléseket.
- **Statisztikák és Jelentések:** Részletes statisztikák és jelentések a termékek eladásairól, felhasználói aktivitásról és egyéb üzleti mutatókról.

5 Összegzés

A projekt egy jól felépített alapot biztosít egy nyilvántartó rendszerhez, amely képes kezelni a termékek nyilvántartását, felhasználók bejelentkezését és regisztrációját, valamint a kosár funkcióit. Az alkalmazás további fejlesztésekkel jelentősen bővíthető és javítható, hogy még több funkcionalitást és jobb felhasználói élményt nyújtson. A modern technológiák használatával, mint az Angular és az Express.js, az alkalmazás skálázható és könnyen karbantartható.