Beadandó feladat dokumentáció

Web technológiák 2. tárgyból

2025.

Név: Lengyel Gábor
Neptun-kód: GKIU70

Tartalomjegyzék

1. Bevezetés

A feladat célja egy egyszerű webalkalmazás prototípusának készítése Angular, Node.js és MongoDB alkalmazásával.

A weboldal lehetővé teszi a felhasználók számára, hogy felhasználói fiókat hozzanak létre, és bejelentkezzenek. Bejelentkezés után elérhetővé válik a terméknyilvántartás, amelyet a bejelentkezett felhasználó feltölthet termékekkel, valamint módosíthat vagy törölhet meglévő termékeket.

A prototípus számítógép alkatrészek nyilvántartására lett kialakítva.

2. Technológiák

2.1. Frontend

2.1.1. Angular

Az Angular egy népszerű, TypeScript-alapú frontend keretrendszer, amely komponensalapú felépítésével elősegíti jól strukturált, moduláris és komplex webalkalmazások fejlesztését. A projekt főként Angular segítségével valósítja meg a frontendet.

2.1.2. Angular Material

Az Angular hivatalos komponenskönyvtára, amely előre definiált, reszponzív és modern megjelenésű UI-elemeket biztosít (például űrlapok, gombok). Ezek használatával könnyebben kialakítható egy letisztult és felhasználóbarát felület.

2.1.3. TypeScript

A TypeScript a JavaScript típusos, objektumorientált kiterjesztése, amely segíti a nagyobb, jól karbantartható alkalmazások fejlesztését. Az Angular teljes mértékben TypeScript-re épül, így a projektben is elengedhetetlen a használata.

2.2. Backend

2.2.1. Node.js

A Node.js egy eseményvezérelt, aszinkron JavaScript futtatókörnyezet, lehetővé teszi szerveroldali alkalmazások és API-k fejlesztését JavaScript nyelven. A projekt backendje Node.js segítségével biztosítja az adatok elérését.

2.2.2. MongoDB

A MongoDB egy dokumentumalapú NoSQL adatbázis, amely JSON-szerű dokumentumokat tárol. Rendkívül rugalmas és jól alkalmazható dinamikus, változó szerkezetű adatok kezelésére. A projekt a MongoDB-t használja a felhasználók és a termékek adatainak tárolására.

3. Követelmények

3.1. Funkcionális követelmények

- **Regisztráció és bejelentkezés**
  - A felhasználók regisztrálhatnak felhasználói fiókot, amely során meg kell adniuk a felhasználónevet, e-mail címet és jelszót.
  - A rendszer ellenőrzi az adatok helyességét és nem enged már létező felhasználónévvel vagy e-mail címmel regisztrálni.
  - A regisztrált felhasználók be tudnak jelentkezni a rendszerbe felhasználónév és jelszó megadásával.
- **Termékek megtekintése**
  - A bejelentkezett felhasználók megtekinthetik a termékek listáját.
  - A termékek szűrhetők keresőszó, kategória és márka alapján is.
  - A termékek rendezhetők név vagy ár szerint növekvő vagy csökkenő sorrendben.
  - Limitálható az egy oldalon megjelenített találatok száma.
- **Termékek hozzáadása**
  - A bejelentkezett felhasználók új termékeket adhatnak hozzá az adatbázishoz.
  - A terméknek kötelező megadni a nevét, márkáját, kategóriáját és árát, továbbá opcionálisan leírás adható meg.
  - Az új termékek adatai tárolásra kerülnek az adatbázisban.
- **Termékek módosítása**
  - A bejelentkezett felhasználók módosíthatják a már meglévő termékek adatait.
- **Termékek törlése**
  - A bejelentkezett felhasználók törölhetik a termékeket.
  - Törlés megerősítésére szolgáló párbeszédablak megjelenítése a véletlen törlés elkerülése érdekében.
  - Csak létező termék törlése engedélyezett, egyébként hibaüzenet jelenik meg.

3.2. Nem funkcionális követelmények

- **Teljesítmény**
  - A rendszernek gyors válaszidőt kell biztosítania a felhasználók számára.
  - Az adatbázis lekérdezések optimalizáltak legyenek, hogy a nagyobb adatmennyiség esetén is elfogadható sebességet nyújtsanak.
- **Biztonság**
  - Csak hitelesített felhasználók férhessenek hozzá a termékek szerkesztési funkcióihoz.
- **Használhatóság**
  - A felhasználói felület legyen egyszerű, intuitív és könnyen kezelhető.
  - A rendszer biztosítson visszajelzést a felhasználónak a műveletek eredményéről (sikeres vagy sikertelen).

- A felület legyen reszponzív, különböző kijelzőméreteken is jól használható és áttekinthető legyen.

- **Karbantarthatóság**
  - A rendszer kódja legyen átlátható, jól dokumentált és modulárisan felépített, hogy a jövőbeni módosítások, bővítések és hibajavítások könnyen elvégezhetők legyenek.
  - Az alkalmazás felépítése támogassa a verziókezelést és a csapatmunkát.

4. Adatmodell

- Felhasználók (Users)
  - _id (ObjectId): Egyedi azonosító.
  - username (string): Egyedi felhasználónév.
  - email (string): Egyedi e-mail cím.
  - password (string): Jelszó.
  - createdAt (Date): A regisztráció időpontja.

- Termékek (Products)
  - _id (ObjectId): Egyedi azonosító.
  - name (string): A termék neve.
  - category (string): A termék kategóriája.
  - brand (string): A termék márkája.
  - price (number): A termék ára.
  - description (string): Részletes leírás a termékről.
  - createdAt (Date): A termék létrehozásának időpontja.

5. Backend

A backend feladata a kliens által küldött kérések kiszolgálása és az adatbázis kezelése. A frontend és a backend közötti kommunikáció egy API-n keresztül történik, ami az adatok továbbítását HTTP-kéréseken keresztül végzi.

- GET: Adatok lekérdezésére szolgál.
- POST: Új adat létrehozására szolgál.
- PUT: Létező adat módosítására szolgál.
- DELETE: Létező adat törlésére szolgál.

5.1. Felhasználókezelés

5.1.1. Regisztráció

POST /api/register

Ellenőrzésre kerül, hogy minden adat megadásra került-e, továbbá korábban még nem regisztrált felhasználónév és e-mail cím került-e megadásra.

Válaszok:

- 201 Created – Sikeres regisztráció
- 400 Bad Request – Hiányzó adatok
- 409 Conflict – Létező felhasználónév vagy email
- 500 Internal Server Error – Szerverhiba

5.1.2. Bejelentkezés

POST /api/login

Bejelentkezés a megadott felhasználónév és jelszó alapján. Hibajelzés hiányzó vagy hibás adatok esetén.

Válaszok:

- 200 OK – Sikeres bejelentkezés
- 400 Bad Request – Hiányzó adatok
- 401 Unauthorized – Hibás felhasználónév vagy jelszó
- 500 Internal Server Error – Szerverhiba

5.2. Termékkezelés

5.2.1. Termékek lekérdezése

GET /api/products

Termékek listázása szűréssel, rendezéssel.

Paraméterek:

- search: név szerinti keresés
- category: kategória szerinti szűrés
- brand: márka szerinti szűrés

- sort: price_asc, price_desc, name_asc, name_desc
- limit: oldalanként megjelenített elemek száma
- page: oldalszám

Válaszok:

- 200 OK – Sikeres lekérés, visszaadja a találatok számát és azok tömbjét
- 500 Internal Server Error – Szerverhiba

5.2.2. Új termék hozzáadása

POST /api/products

Új terméket ad hozzá az adatbázishoz, a következő adatokkal:

- name
- brand
- category
- price
- description (opcionális)

Válaszok:

- 201 Created – Termék sikeresen hozzáadva
- 400 Bad Request – Hiányzó kötelező mezők
- 500 Internal Server Error – Szerverhiba

5.2.3. Termék módosítása

PUT /api/products/:id

Egy létező termék adatait módosítja azonosító alapján.

Válaszok:

- 200 OK – Termék frissítve
- 404 Not Found – Nincs ilyen termék
- 500 Internal Server Error – Szerverhiba

5.2.4. Termék törlése

DELETE /api/products/:id

Termék törlése azonosító alapján.

Válaszok:

- 200 OK – Termék törölve
- 404 Not Found – Nincs ilyen termék
- 500 Internal Server Error – Szerverhiba

6. Frontend

A frontend felület egy egyszerű felhasználói felület, amely lehetővé teszi a felhasználók regisztrációját és bejelentkezését, valamint a termékek megtekintését, létrehozását, módosítását és törlését.

Egy komponens alapvetően egy HTML, egy CSS és egy TypeScript fájlból áll. A HTML a felhasználói felület struktúráját és tartalmát határozza meg, a CSS fájl a megjelenést és stílust szabályozza, a TypeScript pedig az adott komponens logikáját, adatkezelését és eseménykezelését valósítja meg.

A weboldal széles körben alkalmazza az Angular Material komponenskönyvtárát. Megtalálhatók többek között a form-elemek (például mat-form-field, mat-input, mat-select és mat-option), amelyek dinamikus és felhasználóbarát űrlapok készítését teszik lehetővé. Ezen kívül a dialógusablakok (MatDialog), gombok (mat-button, mat-raised-button), kártyák (mat-card), értesítések (MatSnackBar) megjelenítésében is segítséget nyújtott.

6.1. Navbar

A navigációs sáv az oldal tetején helyezkedik el fixen. Ezen keresztül navigálhatunk el a különböző oldalakra:

- Főoldal
- Termékek
- Elérhetőségek
- Bejelentkezés
    - Regisztráció

A bejelentkezési felület egy lenyíló űrlap, ahonnan a felhasználó elnavigálhat a regisztrációs felületre is. Bejelentkezett állapotban egy üdvözlő szöveg jelenik meg a Bejelentkezés gomb helyett, amihez tartozik egy lenyíló Kijelentkezés gomb.

6.2. Home

A kezdőoldal. A navigációs sávon a logóra kattintva is vissza lehet ide térni.

6.3. Products

Ez a komponens a terméklista megjelenítéséért felel. Biztosítja a bejelentkezett felhasználók számára a termékek böngészését, szűrését és szerkesztését.

Ha a felhasználó nincs bejelentkezve, egy hibaüzenetet lát. Bejelentkezett felhasználók számára megjeleníti a termékek listáját, valamint a szükséges szűrő és elemeket, lapozó gombokat és a termékek szerkesztéséhez szükséges gombokat.

6.4. ProductForm

Ez a komponens felel a termékek létrehozásához vagy módosításához szükséges űrlap megjelenítéséért és működtetéséért. A form validációt végez, és kommunikál a backenddel az adatok mentéséhez.

6.5. AuthService

Ez kezeli a felhasználó hitelesítésének állapotát, valamint a bejelentkezett állapotot és felhasználónév tárolását, elérhetőségét az alkalmazás egészében.

7. Beüzemelés

A projekt elérhető az alábbi GitHub repositoryban:

https://github.com/Eldzsi/web-technologies-2

Backend beüzemelése:

cd backend
npm install: Függőségek telepítése.
node server.js: Szerver elindítása.

Frontend beüzemelése:

cd frontend
npm install: Függőségek telepítése.
ng serve: Szerver elindítása.

A weboldal ezután elérhető böngészőben a következő címen:

http://localhost:4200

8. Összegzés

A dokumentáció egy webalkalmazás prototípusát mutatja be, amely Angular, Node.js és MongoDB technológiákra épül. A rendszer lehetővé teszi a felhasználók számára, hogy regisztráljanak, bejelentkezzenek, valamint termékeket kezeljenek.

A frontend Angular keretrendszerre és az Angular Material komponenskönyvtárra épül, amely modern, reszponzív és felhasználóbarát kezelőfelületet biztosít. A backend Node.js környezetben fut, és MongoDB adatbázissal kommunikál.

A prototípus kiindulópontként szolgálhat egy komplexebb, éles környezetben is használható webalkalmazás fejlesztéséhez.