

Webtechnológiák 2

Név: Demján Csongor

Neptun kód: SZ7MGG

Telepítés/Indítás

Frontend:

Lépjen be a mappába: `frontend/`

Telepítse a függőségeket: `npm install`

Indítsa el a szerveret: `npm start`

Backend:

Klónozza a repót: `git clone https://github.com/DemjanCsongor/webtech2`

Lépjen be a mappába: `cd itemregistry/backend`

Telepítse a függőségeket: `npm install`

Hozzon létre egy `.env` fájlt a gyökérmappában, és adja hozzá a MongoDB kapcsolati karakterláncát:
`MONGODB_URL=your-mongodb-connection-string`

Indítsa el a szerveret: `npm start`

Frontend implementáció

A frontend implementáció Angular keretrendszerben készült, Material Design elemekkel és a TypeScript nyelven.

Tárgyak hozzáadása (item-add)

A `item-add` komponens lehetővé teszi új tárgyak hozzáadását. Egy formban megadható a tárgy neve és leírása, majd a `Submit` gombbal a form adatait elküldi a szervernek. Ha a kérés sikeres, a felhasználó értesítést kap az új tárgy hozzáadásáról, és átirányításra kerül a tárgyak listájához.

Tárgyak részletes nézete (item-detail)

Az `item-detail` komponens egy tárgy részletes adatait jeleníti meg. A nézetben látható a tárgy neve és leírása, továbbá lehetőség van a tárgy törlésére és szerkesztésére. A törlés előtt megerősítést kér a felhasználótól. A szerkesztés gomb az adott tárgy szerkesztő oldalára navigál.

Tárgyak szerkesztése (item-edit)

Az `item-edit` komponensben egy tárgy adatai szerkeszthetők. A formban a tárgy neve és leírása módosítható, majd a `Save Changes` gombbal a változtatásokat elküldi a szervernek. Ha a kérés sikeres, a felhasználó értesítést kap a módosításról, és átirányításra kerül a tárgyak listájához.

Tárgyak listázása (item-list)

Az `item-list` komponens a tárgyak listáját jeleníti meg. Minden sor egy tárgyat reprezentál, a tárgy neve és leírása látható. A nevekre kattintva a felhasználó átirányításra kerül a tárgy részletes nézetéhez.

Bejelentkezés (login)

A login komponensben a felhasználók bejelentkezhettek. A formban meg kell adniuk a felhasználónevüket és jelszavukat. Sikeres bejelentkezés esetén a felhasználók értesítést kapnak a sikeres bejelentkezésről, és átirányításra kerülnek a tárgyak listájához.

Regisztráció (register)

A register komponens lehetővé teszi új felhasználók regisztrálását. A formban meg kell adni a teljes nevet, születési dátumot, lakóhelyet, felhasználónevet és jelszót. A jelszót meg kell erősíteni.

Útvonalmodul (app-routing.module.ts)

Ez a modul tartalmazza az alkalmazás útvonalainak konfigurációját. A Routes típusú routes tömbben határozzuk meg az egyes útvonalakat, ahol az AuthGuard biztosítja, hogy csak bejelentkezett felhasználók férjenek hozzá bizonyos útvonalakhoz.

Főkomponens (app.component.html, app.component.ts)

Az AppComponent a fő komponens, amely beágyazza az összes többi komponenset. Az app.component.html tartalmazza az eszköztárat (mat-toolbar), ahol a bejelentkezés állapota szerint változnak a megjelenített gombok. Az app.component.ts pedig maga a TypeScript osztály, ami az AuthService szolgáltatást használja, hogy ellenőrizze a bejelentkezés állapotát.

Modulok (app.module.ts)

Az AppModule az Angular fő modulja, amely importálja és deklarálja az alkalmazás összes komponensét és szolgáltatását.

Hitelesítési őr (auth.guard.ts)

Az AuthGuard egy olyan szolgáltatás, ami azt ellenőrzi, hogy a felhasználó hozzáférhet-e egy bizonyos útvonalhoz. Ha a felhasználó nincs bejelentkezve, átirányítja őt a bejelentkezési oldalra.

Hitelesítési szolgáltatás (auth.service.ts)

Az AuthService a hitelesítési szolgáltatás, amely tartalmazza a bejelentkezés és kijelentkezés logikáját, valamint a regisztrációs funkciót. A login és register metódusok HTTP kéréseket küldenek a szervernek, a válaszokat pedig kezelik.

Tárgyak szolgáltatása (item.service.ts)

Az ItemService a tárgyak kezelésére szolgál. Tartalmazza az összes tárgy lekéréséhez, valamint egy adott tárgy részleteinek lekéréséhez szükséges metódusokat. Ezek a metódusok HTTP kéréseket küldenek a szervernek, és kezelik a válaszokat.

Backend Dokumentáció

Bevezető

Ez a dokumentáció ezen része a backend API-t írja le, amely egy webalkalmazást szolgál ki. Az alkalmazás MongoDB adatbázist használ, a mongoose modullal kezeli az adatokat. Az adatbázisban két fő entitás található: a User és az Item. A kód a Node.js és az Express keretrendszert használja.

Modulok

Modell modulok

User model (User.js)

A User modell a felhasználói adatokat írja le. Egy felhasználónak a következő mezői vannak:

fullName: a felhasználó teljes neve

birthDate: a felhasználó születési dátuma

location: a felhasználó tartózkodási helye

username: a felhasználó felhasználónév

password: a felhasználó jelszó (bcrypt által hashelve)

Item model (Item.js)

Az Item modell egy tárgyat ír le. Egy tárgynak a következő mezői vannak:

name: a tárgy neve

description: a tárgy leírása

Controller modulok

UserController (UserController.js)

A UserController kezeli a felhasználói műveleteket. A következő funkciókat definiálja:

login: bejelentkezést kezel, jelszóellenőrzést végez, és JWT token-t ad vissza

register: új felhasználót regisztrál, a jelszót bcrypttel titkosítja

ItemController (ItemController.js)

Az ItemController kezeli a tárgyakkal kapcsolatos műveleteket. A következő funkciókat definiálja:

getItems: lekéri az összes tárgyat

getItem: lekéri egy adott tárgyat az ID-je alapján

addItem: hozzáad egy új tárgyat

updateItem: frissít egy tárgyat az ID-je alapján

deleteItem: töröl egy tárgyat az ID-je alapján

Middleware modulok

Passport middleware (passport.js)

A Passport middleware kezeli a JWT alapú autentikációt.

Útvonalak

Auth útvonalak (auth.js)

POST /login: bejelentkezés

POST /register: regisztráció

Item útvonalak (backend\routes\item.js)

GET /: az összes tárgy lekérése

GET /:id: egy tárgy lekérése az ID-je alapján

POST /: új tárgy hozzáadása

PUT /:id: tárgy frissítése az ID-je alapján

DELETE /:id: tárgy törlése az ID-je alapján

Fejlesztői eszközök

A projekt a következő függőségeket használja:

bcrypt: jelszavak hashelésére

body-parser: HTTP kérések törzsének elemzésére

cors: Cross-Origin Resource Sharing beállítások kezelésére

dotenv: környezeti változók kezelésére

express: a webalkalmazás keretrendszerére

jsonwebtoken: JWT tokenek kezelésére

mongoose és **mongoose:** adatbázis-kezelésre

passport és **passport-jwt:** autentikációra