

Dokumentáció

Webtechnológiák II.

Beadandó

Készítette: **Görög Krisztina Erzsébet**

Neptunkód: **MPW46D**

Dátum: 2024. 05. 21.

Tartalom

| | |
|-------------------------------|---|
| Bevezetés..... | 3 |
| Felület..... | 3 |
| Az alkalmazás felépítése..... | 5 |
| Model mappa | 5 |
| App komponens | 5 |
| Add komponens | 6 |
| List komponens | 6 |
| Login komponens | 6 |
| Register komponens..... | 6 |
| Backend: db.js | 7 |

Bevezetés

Ez a dokumentum azt a célt szolgálja, hogy bemutassam a webtechnológia tantárgyra készült beadandómat. Egy olyan weboldalt készítettem, amely parfümök raktáron való nyilvántartását teszi lehetővé.

Felület

Ha megnyitjuk az alkalmazást, akkor először a bejelentkező felület fogad bennünket. Ha már regisztráltunk, akkor itt tudunk belépni.





Ha még nem regisztráltunk, akkor azt a regisztráció gombra kattintva a regisztrációs felületen megtehetjük. Itt a felhasználónévnek minimum 3, a jelszónak minimum 6 betűsnek kell lennie.



Belépés után a parfüm hozzáadása felületre navigál az alkalmazás. Itt az adatok megfelelő megadásával hozzá is adhatunk egy parfümöt a listához. A parfüm kódjának egyedinek kell lennie, ez az azonosítója, valamint ez biztosítja, hogy két egyforma parfümöt nem adhatunk meg. Minden mezőt ki kell tölteni.

Listázás

Kijelentkezés

Parfüm hozzáadása

Parfüm neve

Parfüm kódja

Típus

Neme

Megjelenés időpontja

éééé. hh. nn.

Megjelenés időpontja



éééé. hh. nn.

Leírás

Ár

Mennyiség

Hozzáadás

A parfüm hozzáadása után a listázó felületre kerülünk. Itt láthatjuk az eddigi parfümjeinket minden adatukkal kiírva. A parfümök raktáron lévő mennyiségét a + és – gombokkal növelhetjük, illetve csökkenthetjük. A piros szemetesre kattintva törölhetjük a parfümet a listából.

Hozzáadás

Kijelentkezés

Parfümök

| Parfüm neve | Parfüm kódja | Típus | Nem | Megjelenés dátuma | Leírás | Ár(Ft) | Mennyiség(db) | Törlés |
|----------------------|--------------|----------------|---------|-------------------|---|--------|---------------|---|
| Ariana Grande Cloud | 23232 | Eau de Parfume | Női | 2024-03-02 | Ariana Grande új illata. | 27800 | – 4 + |  |
| Tom Ford Lost Cherry | 456 | Eau de Parfume | Uniszex | 2023-11-11 | Cseresznye illat. | 85000 | – 6 + |  |
| Rabanne 1 Million | 565 | Parfume | Férfi | 2022-07-09 | 1 Million Parfum, a bőr vakmerő illata. | 40000 | – 7 + |  |
| Giorgio Armani | | Fau de | | | Akárcsak az | | | |

A lista alatt láthatjuk a parfümök raktáron lévő mennyiségét név szerint egy kördiagrammon. Az alkalmazás random színt rendel az egyes szeletekhez. Ha a kurzorunkat az egyik szelet felé irányítjuk, akkor ott is kiírja a szelet nevét. A kördiagramm a Chart.js segítségével jelenik meg.



Az alkalmazás felépítése

Model mappa

A model mappában találhatjuk a felhasználó és parfüm modelleket.

```
export class User {  
  uname: string;  
  password: string;  
}
```

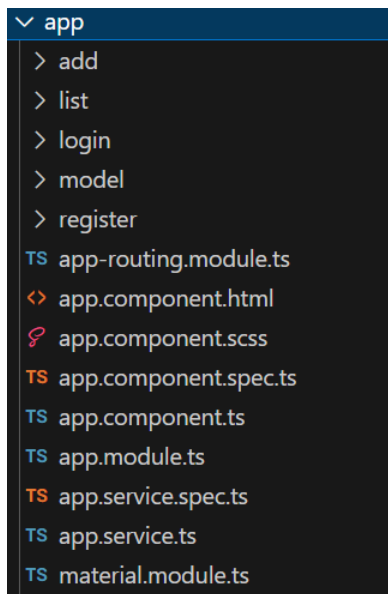
A User osztály a felhasználót definiálja. Adattagjai a felhasználónév és a jelszó.

```
export class Perfume {  
  perfumeName: string;  
  perfumeCode: string;  
  type: string;  
  perfumeFor: string;  
  releaseDate: Date;  
  description: string;  
  price: number;  
  amount: number;  
}
```

A Perfume osztály a parfümöt definiálja. Az adattagok: a parfüm neve, kódja, típusa, neme, megjelenési dátuma, leírása, ára, raktáron lévő mennyisége.

App komponens

Ebbe a komponensbe van beágyazva a többi komponens, gyakorlatilag ez az alkalmazás gyökere. Az index.html fájl gyakorlatilag ehhez fér hozzá először. Benne importáljuk a szükséges modulokat is. Az app.service.ts-ben pedig a backenddel szükséges kommunikációt hajtjuk végre HTTP requestekkel.



Add komponens

Ez a komponens a parfüm hozzáadást teszi lehetővé, azt kezeli. A fájlokat az add mappa tartalmazza.

A html fájl a megjelenítésért felel, a formázását pedig az scss fájl határozza meg. Először megjeleníti a két felső gombot, ami a listázásra navigál, vagy kijelentkeztet. Alatta megjelenik a parfüm hozzáadása form.

Az add.component.ts fájl az oldal viselkedését írja le. Itt implementáljuk a form ellenőrzést. Minden mezőt ki kell tölteni, valamint a parfüm kódnek egyedinek kell lennie.

List komponens

A list komponens a parfümök listázásáért felel.

A html fájl lesz megjelenítve, az scss-ben lévő formázás szerint. Felül megjelenik a hozzáadás gomb, ami visszavezet a parfüm hozzáadása felülethez, valamint a kijelentkezés gomb. A parfümöket egyesével kilistázzuk egy ciklus segítségével. A lista alatt megjelenítjük a kördiagrammot.

A list.component.ts a list komponens viselkedését írja le. Például a getParfume visszaadja a parfümöket. A deleteParfume törli az adott parfümöt. A prepareChartData előkészíti a kördiagramm megjelenítéséhez szükséges adatokat, valamint beállítja kinézetét.

Login komponens

A bejelentkezéshez szükséges komponens.

A html fájl szerint fog az oldal megjelenni az scss-ben megadott beállításokkal. A bejelentkezési form mellett megjelenik egy regisztrációs gomb is, amivel a regisztrációs mezőre navigálhatunk.

A login.component.ts az oldal viselkedését írja le. Kezeli, hogy mi történjen bejelentkezéskor, tehát hogy helyes-e a felhasználónév és jelszó.

Register komponens

A regisztrációhoz szükséges komponens.

A html fájl szerint fog az oldal megjelenni az scss-ben megadott beállításokkal. A regisztrációs mező mellett megjelenik egy bejelentkezési gomb is, amivel a bejelentkezési mezőre navigálhatunk.

A register.component.ts az oldal viselkedését írja le. Kezeli, hogy mi történjen regisztrációkor, tehát hogy megfelelő hosszúságú-e a felhasználónév és jelszó.

Backend: db.js

A db.js felelős az adatbáziskezelésért. Egy szerver segítségével kommunikál az adatbázissal. Definiálja a User és Perfume modelleket.

Először a szerver segítségével csatlakozik az adatbázishoz. HTTP request esetén végrehajtja a megfelelő utasítást. Például a getPerfume esetén visszaadja az összes adatbázisban lévő parfümöt.

```
perfumeRoute.route('/getPerfume').get((req, res) => {  
  perfumeModel.find((error, data) => {  
    if (error) {  
      return next(error)  
    } else {  
      res.json(data)  
    }  
  })  
})
```