Gyulai SZC Szigeti Endre Technikum és Szakképző Iskola

Delicious

„A finom ételek tárháza."

Készítették:

Rostás Barbara - Kasza Tünde - Kiss Lajos

2023.04.28.

**Tartalomjegyzék**

[**Bevezetés 3**](#_c8j7h0qs0pfz)

[**Fejlesztői dokumentáció 4**](#_i49e78gerap1)

[**Felhasználói dokumentáció 6**](#_r2tlc62qi7vj)

[**Kezdőkép 6**](#_abgexkvi46ii)

[**Regisztráció 7**](#_jsmkk9anlva3)

[**Bejelentkezés 8**](#_jq75f0399skg)

[**Kódrészletek az oldalakról 9**](#_j3nnrwbc0v9a)

[**Kategóriák az oldalon 11**](#_p6il052c3dxf)

[**Kód a kategóriák oldalról: 12**](#_4naang151ik6)

[**Recept a weboldalon 13**](#_jlb05yvdj8vr)

[**Kódrészlet a recept oldalról: 14**](#_59n56ihrx5gi)

[**Recept feltöltése a weboldalra 15**](#_3qpko7byqmqe)

[**Kódrészlet a recept feltöltés oldalról: 16**](#_dy18fdw6wnbu)

[**Összefoglalás 18**](#_upbbnpnxos5x)

[**Irodalomjegyzék 19**](#_qc922hwn6knj)

## **Bevezetés**

Az interneten található receptes oldalak száma rohamosan növekszik, és számos személy az online forrásokat választja, ha új ötleteket keresnek az ételkészítéshez. Az interneten található receptes oldalak fő célja, hogy segítsenek az embereknek a különféle ételek elkészítésében, legyen szó akár egyszerű ételekről, akár bonyolultabb ételekről. Ennek az oldalnak a dokumentációja bemutatja, hogy hogyan lett kialakítva és fejlesztve egy olyan receptes weboldal, amely magas minőségű recepteket és hasznos információkat tartalmaz az ételkészítéshez.

## **Fejlesztői dokumentáció**

**Hardver** – Egy Lenovo ThinkCentre M70s SFF személyi számítógépen végeztük a munkánkat.

**Specifikációk:**

⦁ CPU – intel core i5-10400 2.9Ghz

⦁ RAM - 8Gb

⦁ ROM – Kingston SSD 480Gb

**Operációs rendszer:** Microsoft Windows 10 Professional x64.

**Szoftver** – A frontendet mindhárman **Visual Studio Code**-ban írtuk. A backendet **JetBrains** átall fejlesztett **WebStorm**-ban és **Visual Studio Code**-ban írtuk.

**Adatbázis** – **Mongodb**

A **Mongodb Compass** alkalmazást használjuk az adatbázis kezeléséhez.

A táblák könnyen létrehozhatók és feltölthetők.

Egyszerűen kezelhető felülettel rendelkezik.

**Frontend – React:**

Azért választottuk, mert az oldalak egyszerűen felépíthetőek, dizájnolhatóak, hatékonyan lehet benne fejleszteni.

Rengeteg dizájn elem elérhető hozzá, sok mindennel kompatibilis.

Dizájnhoz **MUI** könyvtárát, **CSS**-t, **Bootstrap 5**-öt használtuk.

**Backend – Node:**

A node.js-ről szintén elmondható, hogy kompatibilis elég sok bővítménnyel és csomaggal amik segítségével komplexebb szerverek is írhatók vele.

**Express, Mongoose, Bcrypt, JsonWebToken** és **Multer** könyvtárakat használtuk. A node, különösen az express modullal kiegészítve a lehető legegyszerűbben teszi lehetővé az adatbázisra való kapcsolódást, az információk kinyerését abból a frontend számára.

A **mongoose**-zal pedig a mongodb adatbázisra tudtunk egyszerűen csatlakozni.

**A backend tesztelése – Postman:**

Itt az adatbázis és a backend szerver közötti adatáramlásokat ellenőriztük, hogy minden megfelelően működik e és úgy történik ahogyan azt elvárnánk.

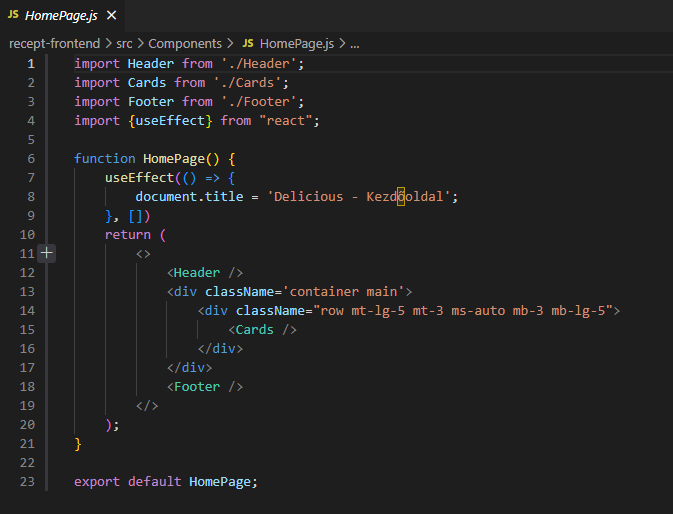
Azért ezt a programot választottuk, mert letisztult felülete és egyszerű használata miatt szimpatikusnak bizonyult számunkra és hasznos, ha látjuk, hogy már az adatbázis és a szerver között sem megy át az információ, nem kell még a frontendet is átnézni és hibát keresni benne.

## **Felhasználói dokumentáció**

### Kezdőkép

Ez itt a legelső felület, amikor fellépünk a weboldalra.

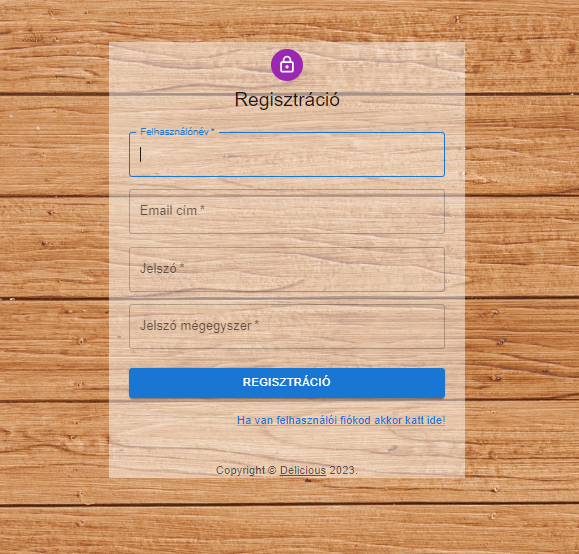
Itt véletlenszerűen jelennek meg különböző receptek.

Felül található egy navigációs sáv, amely lehetővé teszi, hogy bejelentkezzünk, kategóriánként keressünk recepteket, vagy töltsünk fel újat.

### Regisztráció

Ha szeretnénk a weboldal funkcióit használni, akkor ajánlott regisztrálni.

Regisztrációs felület:



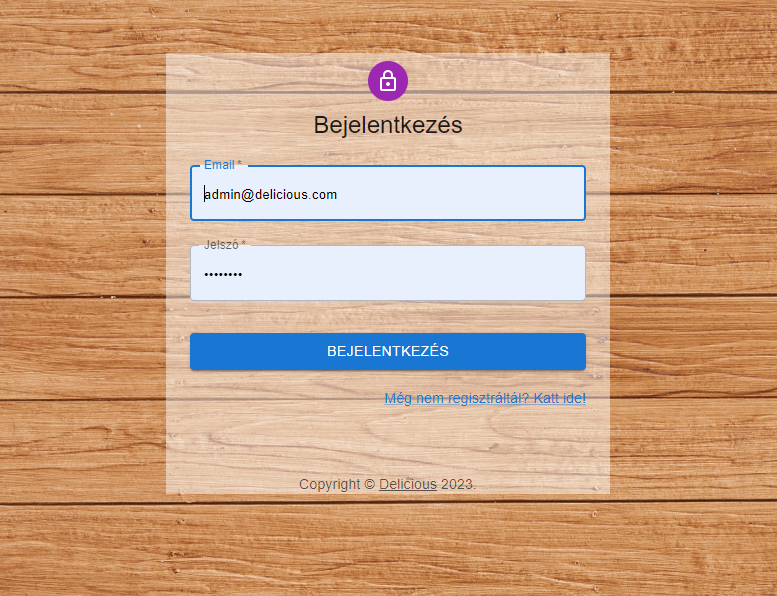
Ha már regisztráltunk, akkor bármikor bejelentkezhetünk.

### Bejelentkezés

A bejelentkezési felület egyszerű, csak a email címet és jelszó megadását kéri.

A bejelentkezési felület fontos része az adatvédelemnek és a biztonságnak, hiszen segítségével a felhasználók személyes adatait és az érzékeny információkat is védhetjük.

Bejelentkezési felület:

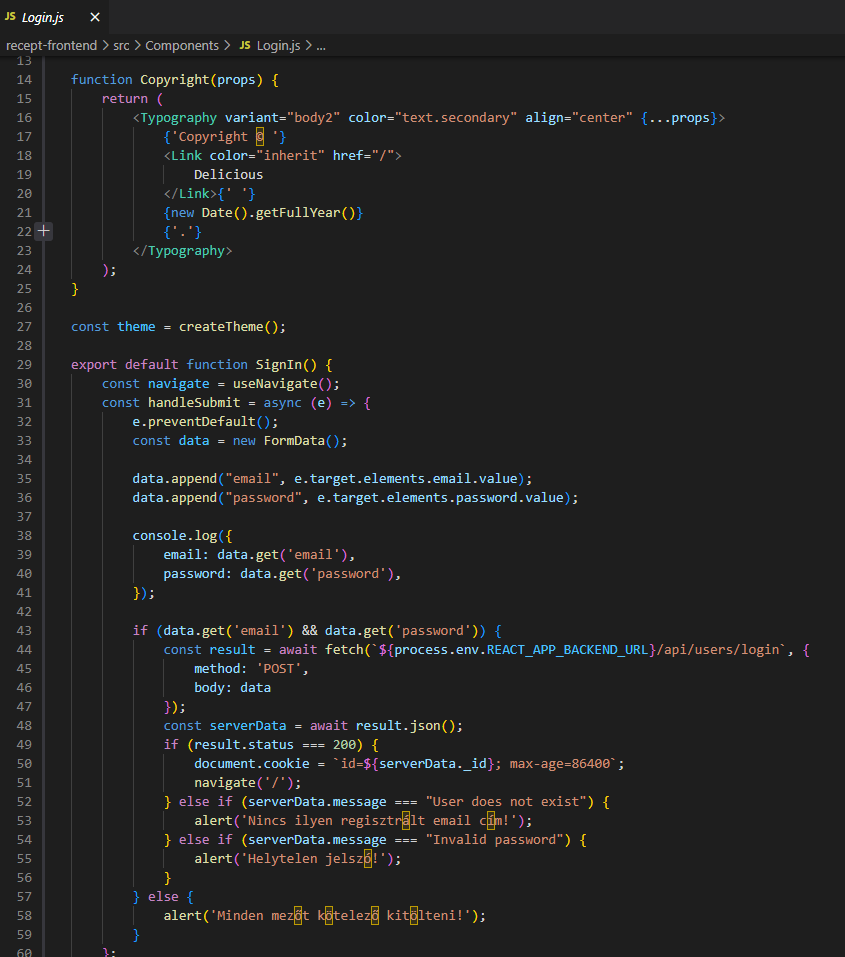


### Kódrészletek az oldalakról

Regisztrációs felületről:



Bejelentkezési felületről:

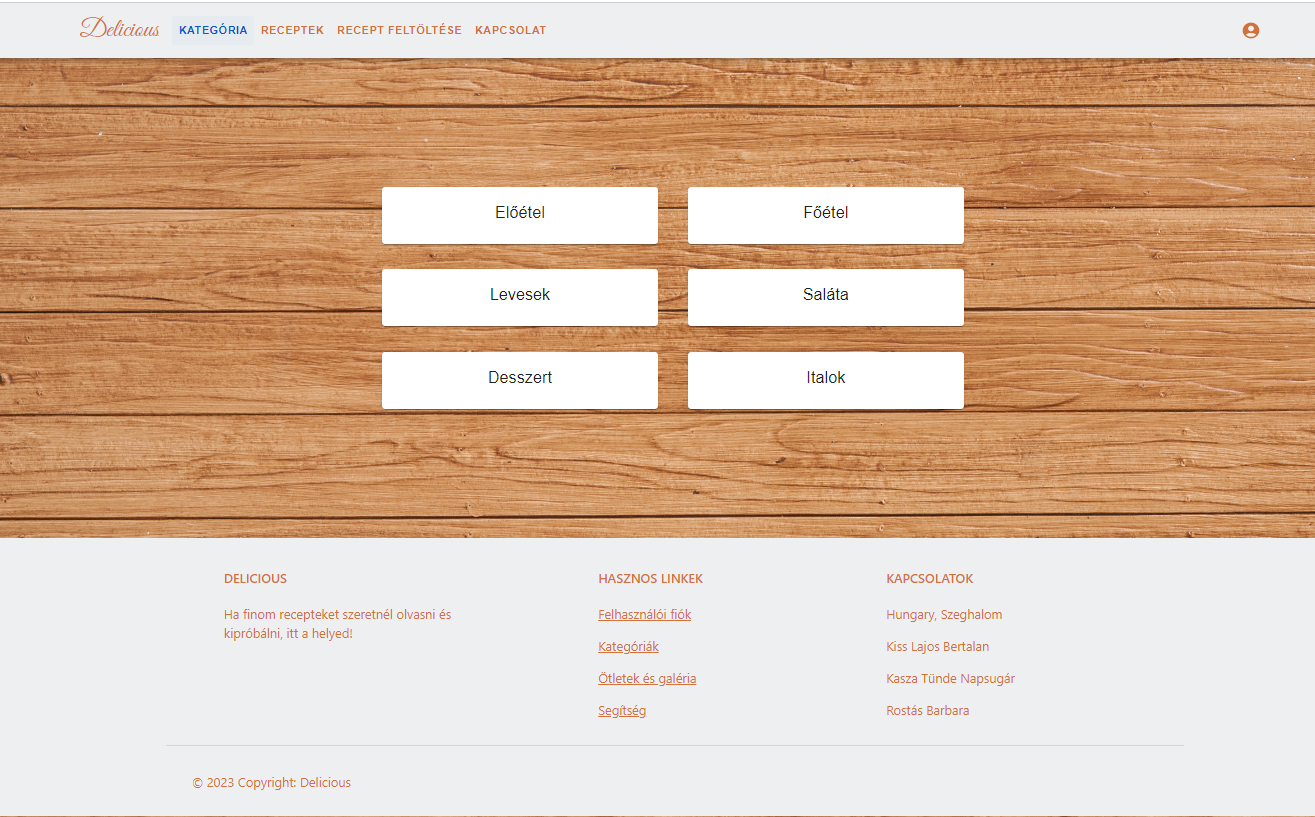


Csak azok férhetnek hozzá a szolgáltatásokhoz vagy a programhoz, akik rendelkeznek érvényes email cím és jelszóval.

### Kategóriák az oldalon

A receptes oldalakon általában kategóriák szerint csoportosítják a recepteket, hogy a felhasználók könnyen megtalálják az érdeklődési körüknek megfelelő ételeket.

A kategóriák:



A kategóriák célja az, hogy a felhasználók gyorsan megtalálják az érdeklődési körüknek megfelelő recepteket, és könnyen navigáljanak az oldalon.

### Kód a kategóriák oldalról:

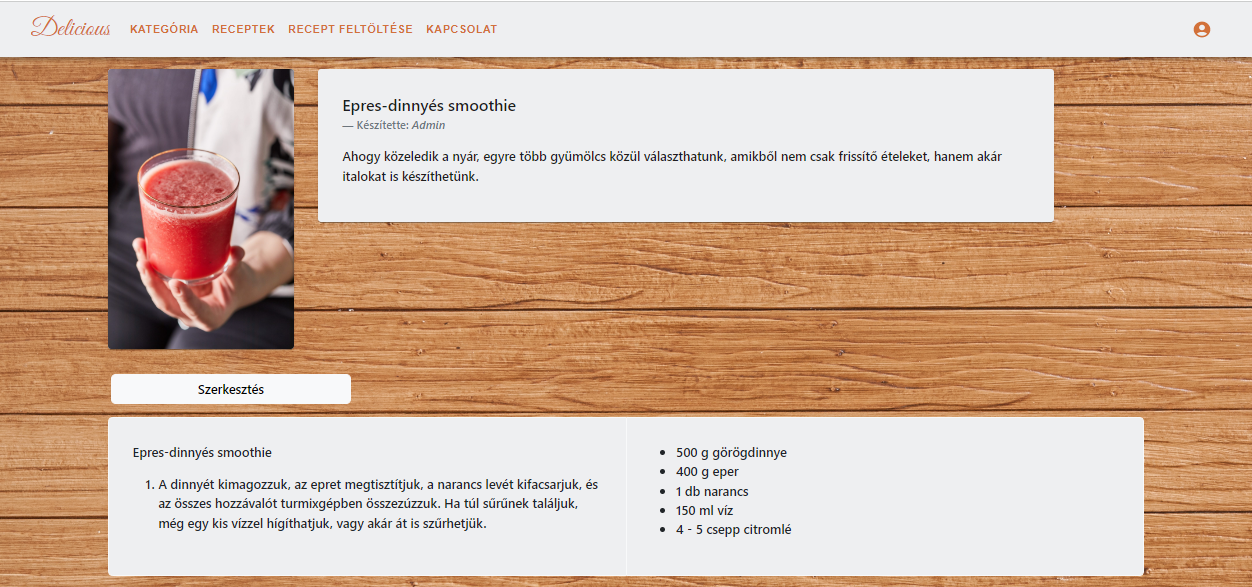


Ez egy React komponens kódja, amely az oldalon megjelenő kategóriák kártyáit állítja elő. A komponens tölti be a kategóriákat a szerverről a useEffect hook segítségével, és a useState hook használatával tartja nyilván a kategóriák állapotát. A kategóriák listáját a categories.map függvénnyel térítjük vissza, amely minden kategória elemre előállítja a kártya HTML szerkezetet, majd azokat a megjelenítési rétegbe helyezi.

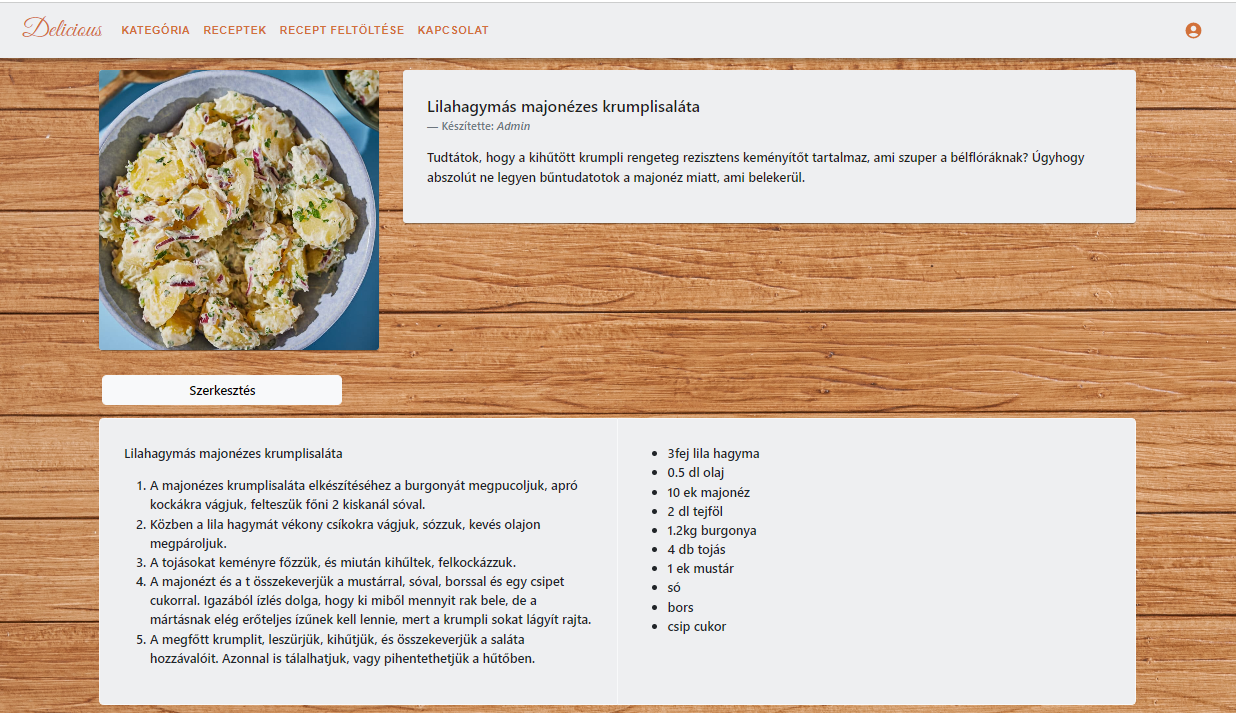
A szöveges tartalom egy Typography komponensben van beágyazva, amely a MUI (Material UI) keretrendszer része. A Typography komponens a szöveg megjelenítési tulajdonságait határozza meg, mint például a betűméret és a betűtípus.

Az export sor pedig lehetővé teszi, hogy a komponens más fájlokból importálható legyen.

### Recept a weboldalon



Megjeleníti a feltöltött képet, a hozzávalókat, az elkészítési módot és egy rövid leírást a receptről.

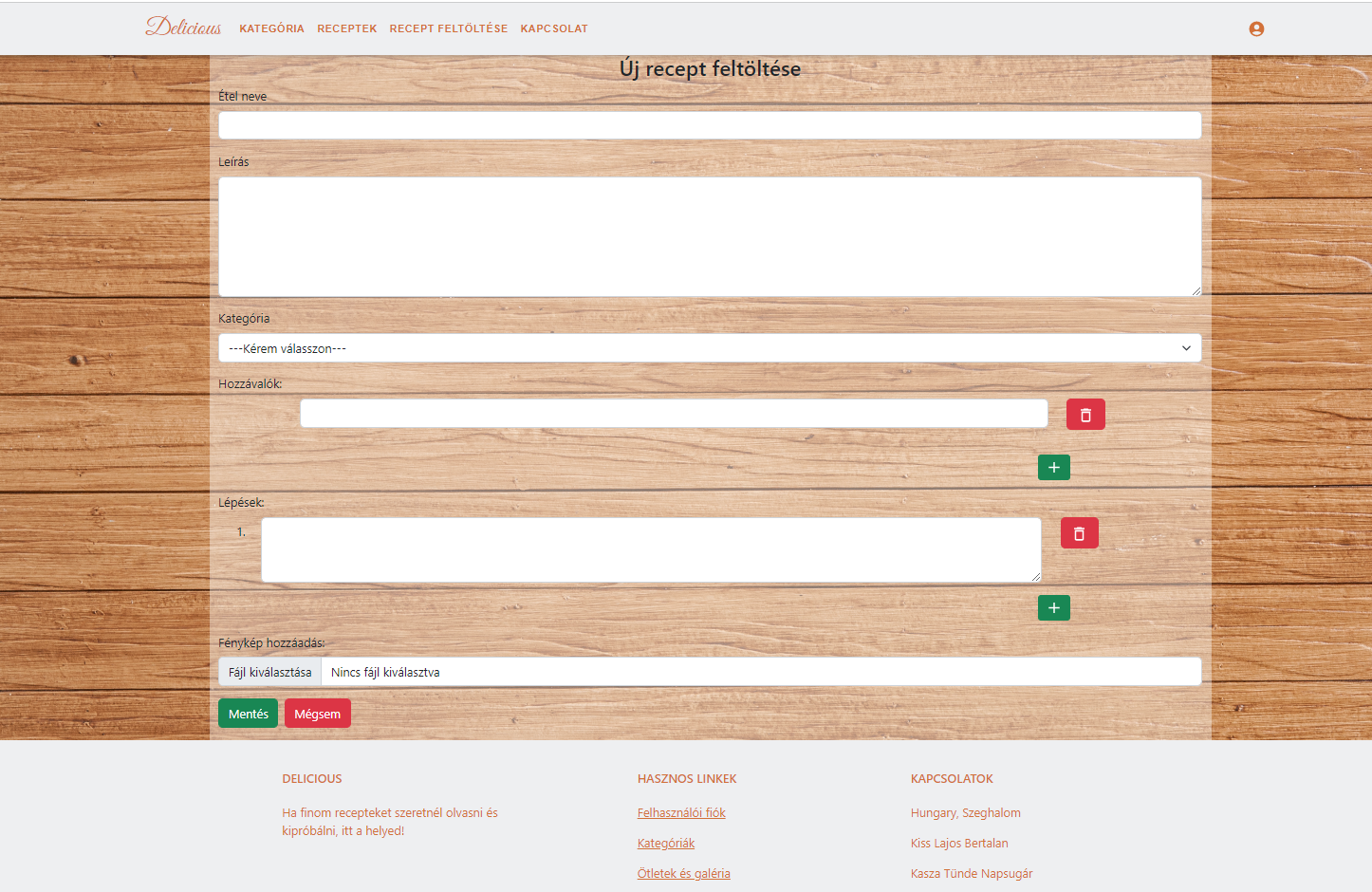


### Kódrészlet a recept oldalról:



A komponens több useEffect hook-ot is használ, amelyek segítségével adatokat kér le a backend szerverről, majd ezeket jeleníti meg a felhasználónak. A fetchCategories függvény segítségével a komponens kategóriákat kér le a szerverről, ezeket a kategóriákat kártyák formájában jeleníti meg. A fetchRecipe függvény segítségével a komponens a megadott recipeId alapján lekéri az adott receptet, majd a recept adatait megjeleníti a felhasználónak, például a nevét, leírását, összetevőit és elkészítési módját. A komponens emellett figyeli a felhasználó bejelentkezési állapotát és jogosultságait, és ha a felhasználó rendelkezik megfelelő jogosultságokkal, lehetősége van szerkeszteni a receptet.

### Recept feltöltése a weboldalra



Itt tud a felhasználó receptet feltölteni, ami azonnal a szerverre tárolódik el és majd onnan hívja meg az oldal. Itt lehet feltölteni a receptről képet, különböző kategóriába lehet besorolni, megadni az összetevőket és az elkészítési folyamatot is.

### Kódrészlet a recept feltöltés oldalról:

Ez egy React kód, amely egy recept feltöltő űrlapot reprezentál. Lehetővé teszi a felhasználók számára, hogy feltöltsék a receptjeiket, az űrlap kitöltésével a recept nevével, leírásával, kategóriájával, összetevőivel és lépéseivel. A felhasználó dinamikusan adhat hozzá és távolíthat el összetevőket és lépéseket is.

A komponens három állapotot határoz meg: összetevők, lépések és kategóriák. Az összetevők állapota egy tömb, amely tartalmazza az összetevők azonosítóját és nevét. A lépések állapota egy tömb, amely tartalmazza azonosítóját és leírását. A kategóriák állapota egy tömb, amely a backend szerverről lekért kategóriákat tartalmazza.

Az űrlap HTML elemek segítségével van megjelenítve, amelyek eseménykezelőket használnak, hogy frissítsék az állapotokat, ahogy a felhasználó a formot kitölti. Amikor az űrlapot elküldik, a komponens a fetch funkcióval küldi el az adatokat a szerverre. A válasz adatokat JSON formátumba alakítják, majd a felhasználó az átirányítás funkciójával a recept oldalra kerül.

## **Összefoglalás**

## **Irodalomjegyzék**