Általános leírás

A DrinkIsland egy komplex webalkalmazás, amelyet Angular és Laravel keretrendszerek segítségével fejlesztettek ki. Célja, hogy széles körű ismereteket nyújtson a felhasználóknak a koktélozás világáról.

A weboldal fő célja, hogy segítsen a felhasználóknak felfedezni és megismerni a különféle koktélok világát. Legyen szó klasszikus receptekről, újító ötletekről vagy érdekességekről, a DrinkIsland mindenkit inspirál és segít abban, hogy új ízeket és italokat fedezzen fel.

Fontos megjegyezni, hogy mivel az alkoholos témákat is feldolgozza, a weboldal célközönsége 18 éven felüliek. Ennek megfelelően a felhasználóknak be kell mutatniuk az életkorukat, mielőtt teljes körű hozzáférést kapnának a tartalomhoz.

A DrinkIsland egy interaktív platform, ahol a felhasználók saját koktél recepteket is megoszthatnak

s

ADMIN | Admin Blog Komponense

Az Admin Blog komponens felelős a blogbejegyzések kezeléséért az adminisztrátor számára. Ennek a komponensnek a fő feladata az adminisztrátor számára biztosítani a blogbejegyzések listázását, szerkesztését és törlését.

Az admin kezelő felülete (HTML)

A komponens HTML kódja egy div elemet tartalmaz, amelynek osztályneve "blog -component". Ez a div egy tömb ciklusban megjeleníti a blogbejegyzéseket kártya formájában. Minden kártya tartalmazza a bejegyzés címét és szövegét egy textarea elemen belül, amelyek readonly módban vannak, így nem szerkeszthetők. Emellett minden kártyán van egy "Törlés" gomb is, amely lehetővé teszi a bejegyzés törlését.

Komponens logikája (TypeScript)

A komponens TypeScript kódja felelős a blogbejegyzések lekérdezéséért, megjelenítéséért, szerkesztéséért és törléséért. A komponens inicializálásakor meghívódik a **loadPosts** metódus, amely lekéri a blogbejegyzéseket a szolgáltatástól. Az **updatePost** és **deletePost** metódusok a blogbejegyzések szerkesztését és törlését végzik el, majd frissítik a megjelenített listát.

ADMIN | Alapanyag Komponens

Az Alapanyag Komponens felelős az alapanyagok kezeléséért, amelyeket a felhasználók hozzáadhatnak, szerkeszthetnek és törölhetnek az alkalmazásban.

**Felhasználói felület (HTML)**

A komponens HTML kódja egy űrlapot és egy listát tartalmaz. Az űrlap lehetővé teszi az új alapanyag hozzáadását, amelynek meg kell adni a nevét és mértékegységét. A lista megjeleníti az összes alapanyagot, minden alapanyaghoz tartozik egy ID, név és mértékegység mező. Minden alapanyag mellett találhatóak gombok a szerkesztéshez és törléshez.

**Komponens logikája (TypeScript)**

A komponens TypeScript kódja felelős az alapanyagok lekérdezéséért, megjelenítéséért, hozzáadásáért, szerkesztéséért és törléséért. Az **ngOnInit** metódus meghívja a **loadAlapanyagok** metódust, amely lekéri az összes alapanyagot a szolgáltatástól. Az **addAlapanyag**, **updateAlapanyag** és **deleteAlapanyag** metódusok felelnek az alapanyagok hozzáadásáért, szerkesztéséért és törléséért, majd frissítik a megjelenített listát.

ADMIN | Ital Komponens

Az Ital Komponens felelős az italok kezeléséért, amelyeket a felhasználók hozzáadhatnak, szerkeszthetnek és törölhetnek az alkalmazásban.

**Felhasználói felület (HTML)**

A komponens HTML kódja egy űrlapot és egy listát tartalmaz. Az űrlap lehetővé teszi az új ital hozzáadását, amelynek meg kell adni a nevét. A lista megjeleníti az összes italt, minden italhoz tartozik egy ID és név mező. Minden ital mellett találhatóak gombok a szerkesztéshez és törléshez.

**Komponens logikája (TypeScript)**

A komponens TypeScript kódja felelős az italok lekérdezéséért, megjelenítéséért, hozzáadásáért, szerkesztéséért és törléséért. Az **ngOnInit** metódus meghívja a **loadItalok** metódust, amely lekéri az összes italt a szolgáltatástól. Az **addItal**, **updateItal** és **deleteItal** metódusok felelnek az italok hozzáadásáért, szerkesztéséért és törléséért, majd frissítik a megjelenített listát.

ADMIN | Recept Komponens

A Recept Komponens felelős az italokhoz és alapanyagokhoz tartozó receptek kezeléséért. A felhasználók hozzáadhatnak, szerkeszthetnek és törölhetnek recepteket az italokhoz és alapanyagokhoz.

**Felhasználói felület (HTML)**

A komponens HTML kódja egy űrlapot és egy listát tartalmaz. Az űrlap lehetővé teszi az új recept hozzáadását, amelyhez meg kell adni az ital ID-jét, az alapanyag ID-jét és a mennyiségét. A lista megjeleníti az összes receptet, minden recepthez tartozik egy ital ID, alapanyag ID és mennyiség mező. Minden recept mellett találhatóak gombok a szerkesztéshez és törléshez.

**Komponens logikája (TypeScript)**

A komponens TypeScript kódja felelős a receptek lekérdezéséért, megjelenítéséért, hozzáadásáért, szerkesztéséért és törléséért. Az **ngOnInit** metódus meghívja a **loadReceptek** metódust, amely lekéri az összes receptet a szolgáltatástól. Az **addRecept**, **updateRecept** és **deleteRecept** metódusok felelnek a receptek hozzáadásáért, szerkesztéséért és törléséért, majd frissítik a megjelenített listát.

ADMIN | Típus Komponens

A Típus Komponens felelős az italok típusainak kezeléséért. A felhasználók hozzáadhatnak, szerkeszthetnek és törölhetnek típusokat az italokhoz.

**Felhasználói felület (HTML)**

A komponens HTML kódja egy űrlapot és egy listát tartalmaz. Az űrlap lehetővé teszi az új típus hozzáadását, amelyhez meg kell adni az ital ID-jét és az egyes típusokhoz tartozó értékeket (téli, nyári, édes, stb.). A lista megjeleníti az összes típust, minden típus mellett találhatóak mezők az ital ID-vel és az egyes típusok értékeivel. Minden típus mellett gombok vannak a szerkesztéshez és törléshez.

**Komponens logikája (TypeScript)**

A komponens TypeScript kódja felelős a típusok lekérdezéséért, megjelenítéséért, hozzáadásáért, szerkesztéséért és törléséért. Az **ngOnInit** metódus meghívja a **loadTipusok** metódust, amely lekéri az összes típust a szolgáltatástól. Az **addTipus**, **updateTipus** és **deleteTipus** metódusok felelnek a típusok hozzáadásáért, szerkesztéséért és törléséért, majd frissítik a megjelenített listát.

ADMIN | User Komponens

A User Komponens felelős a felhasználók kezeléséért és az adminisztrátori jogosultságok kezeléséért.

**Felhasználói felület (HTML)**

A komponens HTML kódja egy listát tartalmaz, amely minden felhasználóhoz tartozik. Minden felhasználó mellett megjelenik a felhasználónév, és egy gomb, amely lehetővé teszi, hogy az adott felhasználót hozzáadjuk vagy eltávolítsuk az adminisztrátori jogosultságok listájáról. A felhasználó típusától függően a gombok szövege és funkcionalitása változik.

**Komponens logikája (TypeScript)**

A komponens TypeScript kódja felelős a felhasználók lekérdezéséért és az adminisztrátori jogosultságok kezeléséért. Az **ngOnInit** metódus meghívja a **loadUserek** metódust, amely lekéri az összes felhasználót a szolgáltatástól. A **setAdmin** metódus felelős az adminisztrátori jogosultságok beállításáért vagy eltávolításáért, majd frissíti a megjelenített listát.

Blog Komponens

A Blog Komponens felelős a blogbejegyzések megjelenítéséért.

**Felhasználói felület (HTML)**

A komponens HTML kódja egy bejegyzések listáját tartalmazza, amelyet dinamikusan generálunk az **\*ngFor** direktívával. Minden bejegyzésnek van egy címe (**post.title**) és szöveges tartalma (**post.text**), amelyeket a komponens HTML kódban jelenítünk meg.

**Komponens logikája (TypeScript)**

A komponens TypeScript kódja felelős a blogbejegyzések lekérdezéséért a szolgáltatástól, és a megfelelő adatok megjelenítéséért a felhasználói felületen. Az **ngOnInit** metódus meghívja a **getPosts** metódust, amely lekéri az összes bejegyzést a szolgáltatástól, majd eltárolja azokat a **posts** változóban, amelyet a HTML kódban használunk az adatok megjelenítésére.

Home komponens

* Az első szakasz (section class="hero") egy hős rész, amely üdvözli a felhasználót és bemutatja az oldalt egy képpel.
* A második szakasz (section class="features") bemutatja az oldal főbb jellemzőit, mint például a koktél receptek, a blog és a regisztráció lehetőségét.

Calculator Komponens

Az Alkohol kitisztulási idő kalkulátor komponens egy űrlapot biztosít a felhasználók számára, hogy kiszámítsák a becsült véralkoholszintjüket és az alkohol kitisztulásának várható idejét. A felhasználóknak lehetőségük van megadni a súlyukat, nemüket, az elfogyasztott italok mennyiségét és alkoholtartalmát, valamint az utolsó ital elfogyasztása óta eltelt időt.

**Felhasználói felület (HTML)**

Az űrlap lehetővé teszi a felhasználók számára, hogy megadják a súlyukat, nemüket, az elfogyasztott italok mennyiségét és alkoholtartalmát, valamint az utolsó ital elfogyasztása óta eltelt időt. Az "Hozzáadás" gomb segítségével új italokat adhatnak hozzá az űrlaphoz, és az "Eltávolítás" gomb segítségével törölhetik azokat. A "Számítás" gombra kattintva a rendszer kiszámítja a becsült véralkoholszintet és az alkohol kitisztulásának várható idejét, és ezeket megjeleníti a felhasználó számára.

**Komponens logikája (TypeScript)**

A komponens TypeScript kódja felelős az italok hozzáadásáért, eltávolításáért és a becsült véralkoholszint, valamint az alkohol kitisztulásának várható idejének kiszámításáért. A **CalculatorService** segítségével a komponens kiszámítja az alkohol kitisztulásának várható idejét és a becsült véralkoholszintet a megadott adatok alapján.

User-blog komponens

Az új blogkomponens segítségével felhasználók tudnak posztokat hozzáadni, frissíteni és törölni az általuk létrehozott posztokat.

**Felhasználói felület (HTML)**

Az űrlap lehetővé teszi a felhasználók számára, hogy címet és szöveget adjanak meg az új poszthoz. Az "Mentés" gombra kattintva a rendszer hozzáadja az új posztot a listához.

A már meglévő posztok listája is megjelenik az oldalon. Minden poszthoz tartozik egy cím és egy szöveg mező, amelyek szerkeszthetők. Minden poszthoz tartozik egy "Mentés" gomb a módosítások mentéséhez, valamint egy "Törlés" gomb a poszt törléséhez.

**Komponens logikája (TypeScript)**

A komponens TypeScript kódja felelős az új poszt hozzáadásáért, meglévő posztok frissítéséért és törléséért. Az adatok mentése, frissítése és törlése az **addPost**, **updatePost** és **deletePost** metódusok segítségével történik, amelyek hívják a megfelelő szolgáltatás metódusait.

A **loadPosts** metódus felelős a már meglévő posztok betöltéséért az oldal megnyitásakor.