

BABEŞ–BOLYAI TUDOMÁNYEGYETEM KOLOZSVÁR  
MATEMATIKA ÉS INFORMATIKA KAR  
INFORMATIKA SZAK

Szakdolgozat

# **CosmeticRadar: Mobil- és webalkalmazás a kozmetikai termékek összetételének nyilvántartására**



TÉMAVEZETŐ:

DR. SZENKOVITS ANNAMÁRIA,  
EGYETEMI TANÁRSEGÉD

SZERZŐ:

PARAJDI ZOÁRD

2022

BABEȘ–BOLYAI UNIVERSITY OF CLUJ-NAPOCA  
FACULTY OF MATHEMATICS AND INFORMATICS  
SPECIALIZATION: COMPUTER SCIENCE

Diploma Thesis

# **CosmeticRadar: Mobile and web application for keeping track of the ingredients of cosmetic products**



ADVISOR:

ANNAMÁRIA SZENKOVITS, PhD.  
ASSIST. PROF.

AUTHOR:

ZOÁRD PARAJDI

2022

UNIVERSITATEA BABEȘ-BOLYAI, CLUJ-NAPOCA  
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ  
SPECIALIZAREA INFORMATICĂ

Lucrare de licență

**CosmeticRadar: Aplicație mobilă și web  
pentru urmărirea ingredientelor produselor  
cosmetice**



CONDUCĂTOR ȘTIINȚIFIC:  
ASIST. DR. ANNAMÁRIA SZENKOVITS

ABSOLVENT:  
ZOÁRD PARAJDI

2022

BABEȘ–BOLYAI UNIVERSITY OF CLUJ-NAPOCA  
FACULTY OF MATHEMATICS AND INFORMATICS  
SPECIALIZATION: COMPUTER SCIENCE

## **Diploma Thesis**

# **CosmeticRadar: Mobile and web application for keeping track of the ingredients of cosmetic products**

### **Abstract**

Cosmetic products are used by everyone, but only a few people know about the components and different effects of this products. This app is helping us to know the products in more detail. Using the mobile application, we can learn about a product, what active ingredients it contains, what effects it has, and we can also check whether it contains any harmful, already banned, or skin-irritating substances. The application solves this with the help of a database created by the European Union. The application can also be used to check the effects in the product description, to find out what long-term effects they have. The application also includes a web interface where we can view the list of cosmetic product ingredients and modify the database if necessary.

This work is the result of my own activity. I have neither given nor received unauthorized assistance on this work.

2022

ZOÁRD PARAJDI

ADVISOR:  
ANNAMÁRIA SZENKOVITS, PHD.  
ASSIST. PROF.

# Tartalomjegyzék

<b>1. Bevezető</b>	<b>1</b>
1.1. Hasonló alkalmazások . . . . .	1
1.2. A dolgozat felépítése . . . . .	3
<b>2. Funkcionalitások</b>	<b>4</b>
2.1. Funkcionalitások a webes kliensben . . . . .	4
2.1.1. A vendég felhasználó . . . . .	4
2.1.2. Az alapértelmezett felhasználó . . . . .	4
2.1.3. Az adminisztrátor felhasználó . . . . .	5
2.2. Funkcionalitások a mobil kliensben . . . . .	5
<b>3. Az alkalmazás felépítése</b>	<b>7</b>
3.1. Fő adatbázis . . . . .	8
3.1.1. Az adatbázis kitelepítése . . . . .	8
3.2. Szerver . . . . .	9
3.2.1. Kapcsolat az adatbázissal . . . . .	9
3.2.2. API . . . . .	10
3.2.3. Biztonság . . . . .	11
3.2.4. A szerver kitelepítése . . . . .	12
3.3. Mobil alkalmazás . . . . .	14
3.3.1. Adatmodell a mobilalkalmazásban . . . . .	14
3.4. Webes kliens . . . . .	16
3.4.1. Komponensek . . . . .	16
3.4.2. API kliens . . . . .	17
3.4.3. A kliens kitelepítése . . . . .	18
<b>4. Felhasznált technológiák</b>	<b>19</b>
4.1. Szerver oldali technológiák . . . . .	19
4.1.1. Node.js . . . . .	19
4.1.2. MySQL . . . . .	20
4.2. Mobil technológiák . . . . .	20
4.2.1. React native . . . . .	20
4.2.2. Szövegfelismerés . . . . .	21
4.2.3. SQLite . . . . .	21

## TARTALOMJEGYZÉK

4.3.	Webes technológiák . . . . .	22
4.3.1.	React . . . . .	22
4.3.2.	Bootstrap . . . . .	22
4.4.	Eszközök . . . . .	23
4.4.1.	Git és GitHub . . . . .	23
4.4.2.	Docker . . . . .	23
4.4.3.	MySQL Workbench . . . . .	24
4.4.4.	Visual Studio Code . . . . .	24
<b>5.</b>	<b>Az alkalmazás működése</b>	<b>25</b>
5.1.	Mobil alkalmazás . . . . .	25
5.1.1.	Főoldal . . . . .	25
5.1.2.	Információs oldal . . . . .	25
5.1.3.	A termék elmentésére szolgáló oldal . . . . .	25
5.1.4.	A termékek oldala . . . . .	27
5.2.	Webes kliensalkalmazás . . . . .	29
5.2.1.	Főoldal . . . . .	29
5.2.2.	Bejelentkezési oldal . . . . .	31
5.2.3.	Felhasználók kezelésének oldala . . . . .	31
5.2.4.	Az adatbázis feltöltésére szolgáló oldal . . . . .	31
<b>6.</b>	<b>Összefoglaló és továbbfejlesztési lehetőségek</b>	<b>33</b>

## 1. fejezet

# Bevezető

A kozmetikai termékeket mindenki használja, de csak kevesen vannak akik ismerik az általuk használt termékek összetevőit, különböző hatásait. Ez az alkalmazás az általunk használt termékek részletesebb megismerésében segít minket. A mobilalkalmazás használatával megtudhatjuk egy termékről, hogy milyen hatóanyagokat tartalmaz, milyen hatásai vannak, és azt is ellenőrizni tudjuk hogy tartalmaz-e valamilyen káros, már betiltott, esetleg a bőrt irritáló anyagokat. Ezt az alkalmazás egy, az Európai Unió által készített adatbázis segítségével oldja meg ([Cos]). Az alkalmazás arra is alkalmas lehet, hogy a termék leírásában szereplő hatásokat leellenőrizzük, megtudjuk, hogy milyen hosszú távú hatásai vannak.

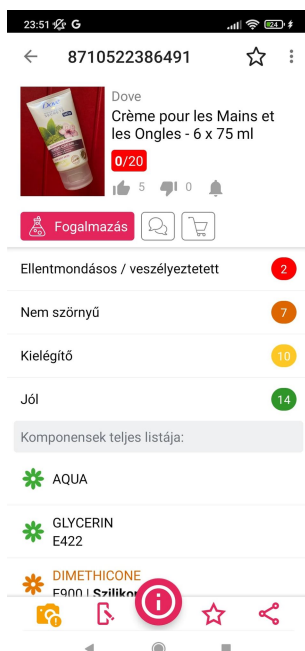
Az alkalmazáshoz tartozik egy webes felület is, ahol a kozmetikai termékek összetevőinek a listáját tudjuk megtekinteni, és szükség esetén módosítani az adatbázist.

## 1.1. Hasonló alkalmazások

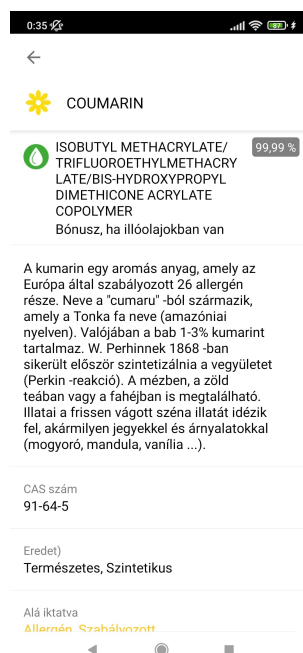
Az alkalmazás készítése során megvizsgáltunk néhány, már a piacon levő, hasonló funkcionalitású mobilalkalmazást és találtunk is egy pár hasonló androidos alkalmazást. A meglévő alkalmazások többsége egy terméket úgy ismer fel, hogy be kell írjuk a termék nevét, vagy be kell szkenneljük a termék vonalkódját, de vannak olyanok is amik kép alapján ismerik fel a terméket. A következőkben ezekből az alkalmazásokból mutatunk be két Android alkalmazást.

**INCI Beauty** Ezt a mobilalkalmazást egy francia cég készítette ([INC]). Az alkalmazásban egy terméket kétféleképpen lehet felismerni: a név beírásával, vagy a termék vonalkódjának beolvasásával. Miután felismert az alkalmazás egy terméket, az alábbi ábra fogad (1.1), ahol látnunk egy képet a termékről, írhatunk hozzászólást és megtekinthetjük a termék árát is. A termék

## 1. FEJEZET: BEVEZETŐ



1.1. ábra. INCI Beauty alkalmazás használat közben



1.2. ábra. Az összetevő részletei

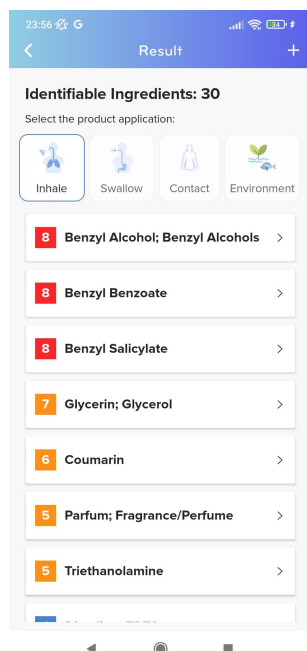
összetevőit 4 kategóriába sorolja annak függvényében, hogy mennyire egészségesek, ha rányomunk egy kategóriára, akkor láthatjuk a kategóriához tartozó összetevőket, de az oldal legalján az összetevők teljes listáját láthatjuk. Ha egy összetevőre rányomunk, akkor itt is elolvashatunk egy részletesebb leírást az összetevőről (1.2).

Az INCI Beauty azért is érdekes, mert van webes felülete is, ahol rá lehet keresni a termékekre és az összetevőkre is, hasonlóan mint az általunk készített alkalmazás webes felületén.

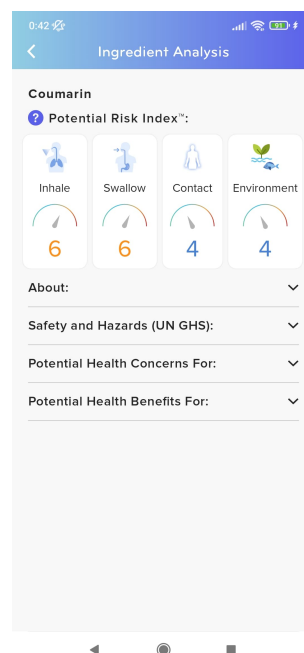
**Ingredient Scanner & Safety** Ezt az alkalmazást a 1Source Limited készítette ([1So]), ami egy Hongkongi egészségügyi cég. Az előbb bemutatott alkalmazással ellentétben itt már arra is van lehetőség, hogy egy terméket kép alapján OCR segítségével is azonosítsunk. Ennek az az előnye, hogy akkor is megnézhetjük egy termék adatait, ha a termék még ismeretlen vagy új. Miután azonosította az alkalmazás az összetevőket, láthatjuk az eredményt, ahol minden összetevő mellett egy számot láthatunk, ami az összetevő veszélyességét jelöli (1.3). Az összetevők veszélyességét négy szempontból is megvizsgálhatjuk, ha váltunk a szempontok között, mindig a legveszélyesebb összetevőket láthatjuk legfelül. Ha rányomunk egy összetevőre, akkor láthatjuk egy oldalon az összetevő veszélyességi mutatóit a különböző szempontok alapján és a leírását, hatásait is megnézhetjük (1.4).



## 1. FEJEZET: BEVEZETŐ



1.3. ábra. Az összetevők listája veszélyesség szerint



1.4. ábra. Az összetevő részletei

## 1.2. A dolgozat felépítése

A dolgozat a következőképpen van felosztva: a 2. fejezet bemutatja az elérhető funkciókat és a kliensalkalmazásban elérhető szerepköröket, a 3. fejezet részletezi az alkalmazás felépítését, a 4. fejezet összefoglalja a felhasznált technológiákat és eszközöket, a 5. fejezet bemutatja az alkalmazás működését és a 6. fejezetben megemlítsre kerül néhány továbbfejlesztési lehetőség.

## **2. fejezet**

# **Funkcionalitások**

## **2.1. Funkcionalitások a webes kliensben**

A webes kliens célja az, hogy a mobilalkalmazás által használt adatbázist felügyeljük a segítségével. A webes felületen a felhasználók meg tudják tekinteni az adatbázisban szereplő összes összetevőt. A webes felület lehetőséget nyújt az adatbázisban szereplő összetevők módosítására, törlésére, de szükség esetén az adatbázis teljes cseréjére is.

A funkcionalitások nem elérhetőek minden felhasználó számára az adatok védelme érdekében, ezért a felhasználók két csoportra oszthatók, amelyek különböző jogosultsággal rendelkeznek, amint azt a 2.1 ábra is szemlélteti. A be nem jelentkezett, vendég felhasználó és a bejelentkezett alapértelmezett felhasználó egy csoportot alkot, a második csoport a bejelentkezett adminisztrátor felhasználó. Az adatbázisban szereplő adatokat csak az adminisztrátor felhasználó módosíthatja, a többi felhasználó csak megtekintheti az adatokat és kereshet az adatok között.

### **2.1.1. A vendég felhasználó**

A vendég felhasználó megtekintheti az adatbázisban szereplő adatokat és kereshet is bennük, emellett bejelentkezhet vagy regisztrálhat is.

### **2.1.2. Az alapértelmezett felhasználó**

Regisztráció után a vendég felhasználóból egy bejelentkezett felhasználó lesz, aki még mindig ugyanazokkal a jogosultságokkal rendelkezik, mint vendégként, de egy adminisztrátor felhasználó így már bármikor jogokat adhat neki.

## 2. FEJEZET: FUNKCIONALITÁSOK



2.1. ábra. A web kliens use case diagramja.

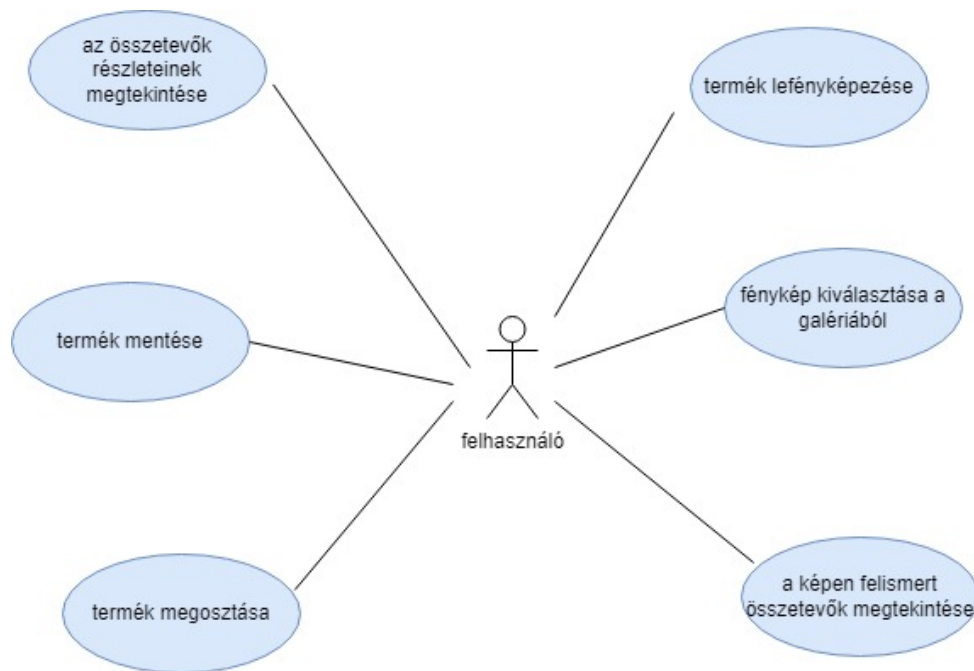
### 2.1.3. Az adminisztrátor felhasználó

Az adminisztrátor felhasználók már módosíthatják egy összetevő részleteit és ki is törölhetnek egyet az adatbázisból. Az adminisztrátor tudja lecserélni a teljes adatbázist, például ha új információk jelentek meg az összetevőkről. Ezenkívül meg tudja tekinteni a regisztrált felhasználókat és módosítani is tudja a felhasználók jogait: egy alapértelmezett felhasználót előléptethet adminisztrátor felhasználóvá, egy adminisztrátor felhasználót pedig lefokozhat alapértelmezett felhasználóvá. Ez a felhasználó továbbá törölni is tud egy felhasználót az adatbázisból.

## 2.2. Funkcionalitások a mobil kliensben

A mobil kliensben nincsenek különböző szerepkörök, mert nincs rájuk szükség, a mobil kliens nem változtatja meg a központi adatbázist, csak adatokat kér le róla (2.2). A felhasználó itt egy, a termék hátoldaláról készített fénykép alapján meg tudja tekinteni a termékében megtalálható összetevőket. A felhasználó megtekintheti az összetevők listáját, ha rákattint egy összetevőre, akkor annak a részletes leírását, tulajdonságait is megtekintheti. A terméket lehe-

## 2. FEJEZET: FUNKCIONALITÁSOK



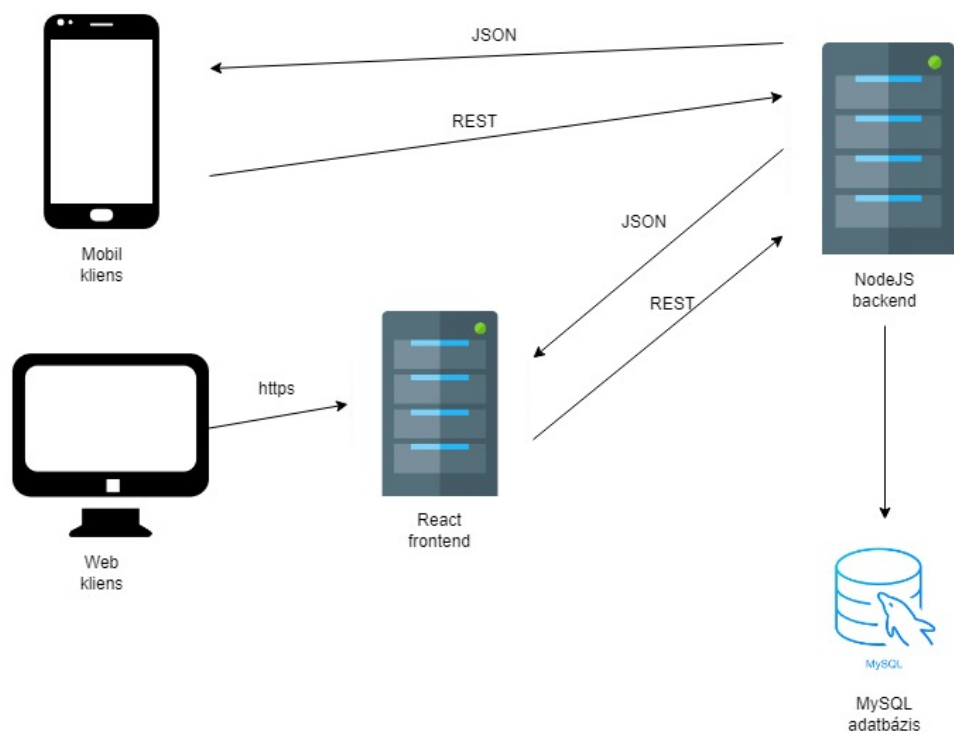
2.2. ábra. A mobil kliens use case diagramja.

tőség van elmenteni, egy leírással, értékeléssel és még egy képpel egy helyi adatbázisba. A már elmentett termékeinket újra megvizsgálhatjuk, és megoszthatjuk az ismerőseinkkel is.

## 3. fejezet

# Az alkalmazás felépítése

**Összefoglaló:** A szoftverrendszer 3 fő komponensből áll amint ez a 3.1 komponensdiagramon is látható: a mobilalkalmazásból, ami React Native-vel lett megvalósítva, egy szerveralkalmazásból, amellyel a kliensalkalmazások REST kérésen keresztül tudnak kommunikálni és egy MySQL adatbázisból. Az adatokat a mobilalkalmazás SQLite adatbázisban tárolja.



3.1. ábra. Az alkalmazás komponensei.

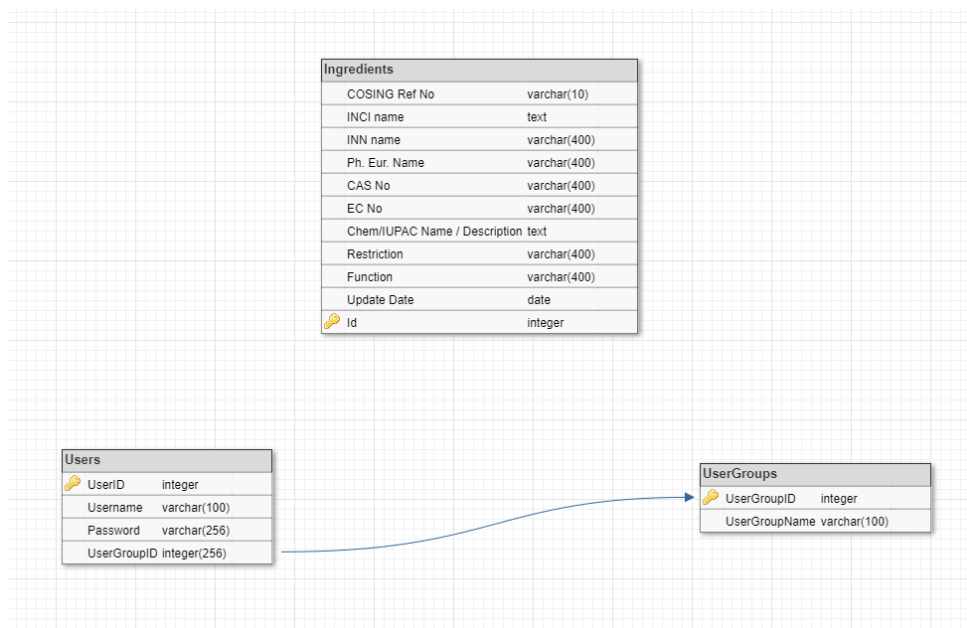
## 3.1. Fő adatbázis

A szerveroldali adatbázis az Európai Unió által létrehozott Cosmetic ingredient database (Cosing) adatbázist használja, ami tartalmazza az összes az Európai Unió területén engedélyezett kozmetikai termék összetevőit és minden összetevőről a fontosabb adatokat ([Cos]). Ezeket az adatokat az adatbázis az Ingredients nevű táblában tárolja el.

Ezenkívül még van két másik tábla, az egyik a felhasználóneveket, a hozzájuk tartozó jelszavakat és a felhasználócsoporthoz azonosítóját tárolja, ez a tábla Users néven van létrehozva, a másik tábla pedig minden felhasználócsoporthoz azonosítóhoz megfeleltet egy csoportnevet, ez a tábla UserGroups néven van jelen.

### 3.1.1. Az adatbázis kitelepítése

Az adatbázis kitelepítését a Google Cloud segítségével valósítottuk meg. A Google Cloud méretezhető számítási felhő szolgáltatásokat nyújt ügyfeleinek. A Google Cloud által nyújtott szolgáltatások ugyanazon az infrastruktúrán futnak, amelyet a Google a saját szolgáltatásaihoz is használ, mint például a Gmail, a Google Suite, vagy a Youtube. A Google Cloud SQL által lefedett három relációs adatbázismotor a következőket tartalmazza: MySQL, PostgreSQL és SQL Server. A Cloud SQL for MySQL egy teljesen felügyelt adatbázis-szolgáltatás, amely segít a



3.2. ábra. A fő adatbázis.

### 3. FEJEZET: AZ ALKALMAZÁS FELÉPÍTÉSE

MySQL relációs adatbázisok beállításában, karbantartásában, kezelésében és adminisztrálásában a Google Cloud Platformon. Általánosságban elmondható, hogy a Cloud SQL-példányok által biztosított MySQL-funkciók megegyeznek a helyileg tárolt MySQL-példányok által biztosított funkciókkal. Azonban van néhány különbség a szabványos MySQL-példány és a Cloud SQL for MySQL-példány között ([Clo] ).

## 3.2. Szerver

Az adatok hatékony elérése érdekében szükség volt egy szerveralkalmazásra is. A szerver válaszol a hozzá beérkező REST (REpresentational State Transfer) kérésekre. A REST API (más néven RESTful API) egy alkalmazásprogramozási felület, amely megfelel a REST architektúra stílusának korlátainak, és lehetővé teszi a RESTful webszolgáltatásokkal való interakciót. A REST a reprezentatív állapottranszfer rövidítése, amit Roy Fielding informatikus hozott létre ([Res] ).

A szerver csatlakozik egy MySQL adatbázishoz és ott tárolja az adatokat.

### 3.2.1. Kapcsolat az adatbázissal

A szerver a kapcsolatot az adatbázissal connection pooling használatával valósítja meg. A pooling egy tervezési minta, ami egy erőforrásból egy meghatározott számú példányt előre előkészít és készenlétbe helyez. A felhasználó kérhet egy példányt és amikor végez a használatával visszatéríti a poolba ([Poo]). Ennek a mintának az az előnye, hogy az inicializáló folyamatok nem várják meg a hívó felet és mivel az erőforrások száma előre meghatározott nem lép fel a túlterheltség veszélye. Az adatbázishoz létrejövő kapcsolatot a db/connection.js fájlban találhatjuk meg.

A kapcsolat létrejötte után a szerver létrehozza a működéshez szükséges táblákat ha azok még nem léteznének. Ezt a db/createTable.js fájlban találhatjuk meg.

Az adatbázisban szereplő adatokhoz kapcsolódó módosítások az elvégzendő műveletek szerint külön fájlokba vannak csoportosítva, ezek a fájlok mind a db mappán belül találhatóak meg.

A deleteDb.js fájlban megtalálható egy felhasználó vagy egy összetevő törlése, és az összes összetevő törlése. Az insertDb.js fájlban megtalálható egy új felhasználói csoport és egy új összetevő beszúrása, valamint egy új alapértelmezett és egy új adminisztrátor felhasználó be-

### 3. FEJEZET: AZ ALKALMAZÁS FELÉPÍTÉSE

szúrása. A request.js fájlban a különböző lekérdezések, az adatok közötti keresések találhatóak meg. Ezeken kívül még van az updateDb.js nevű fájl, amiben egy összetevő tulajdonságait változtathatjuk meg és az uploadDb.js nevű fájl, amelyben megtalálhatjuk az Ingredients tábla feltöltését egy csv fájl importálása által.

#### 3.2.2. API

A web klienst és mobilalkalmazást egy API szerver szolgálja ki, amely betartja a REST (Representational State Transfer) szabványt. Az API szerver és a webkliens, valamint az API szerver és a mobil kliens közötti adatok elküldése JSON formátumban valósul meg. A szerver elérési pontjait két csoportra oszthatjuk, az egyik ahova a bejelentkezéssel kapcsolatos kérések érkeznek és a másik amelyikhez a kozmetikai termékek összetevőivel kapcsolatos kérések. Az összetevőkhöz kapcsolódó kérések kiszolgálása a routes/api.js fájlban található meg, ebben az alfejezetben erről lesz szó.

A mobilalkalmazás a szerverrel csak egyfajta kérés segítségével kommunikál, lekéri egy összetevő minden tulajdonságát az api/ingredients/:name címen egy GET kérés segítségével. A web kliens által az api/ingredients/:id címre érkezhetsz DELETE vagy PUT kérés, amely az kiválasztott összetevőt törli az adatbázisból vagy módosítja. A api/ingredients címen GET kéréssel a kliens lekérheti az adatbázisból az összetevőket, itt tetszőlegesen megadhat szűrőfeltételeket. Az összetevők adatbázisa elég nagyméretű, 30 000 összetevőt tartalmaz, ezért a felhasználó egyszerre csak az adatok egy részét látja. Az adatok oldalankénti megjelenítése szerveroldali pagination segítségével valósul meg, a webes kliensalkalmazás elküldi, hogy hány összetevőt jelenít meg oldalanként és, hogy hányadik oldalt kéri, ezeknek az adatoknak a segítségével a szerver elküldi a kért összetevőket a kliensnek. Az összetevők között lehet név alapján keresni, és abc sorrend szerinti csökkenő vagy növekvő sorrendbe rendezni is. Az api/ingredients címre POST kérés is küldhető, ebben az esetben egy csv fájlt is el kell küldjünk, ekkor a szerver letörli az adatbázisban szereplő összetevőket és importálja az általunk feltöltött csv tartalmát. A szerver a csv fájlt ideiglenesen a tmp/csv mappában tárolja, ahonnan a sikeres adatbázisba importálás után kitörli.

Az adminisztrátor felhasználók megtekinthetik a felhasználók listáját az api/users címről egy GET kéréssel, emellett küldhetnek DELETE és PATCH kérést az api/users/:username címre, amivel törölhetnek egy felhasználót, vagy módosíthatják egy felhasználó rangját. A felhasz-



### 3. FEJEZET: AZ ALKALMAZÁS FELÉPÍTÉSE

náló rangjának módosítása kétféle lehet: lehetőség van egy egyszerű felhasználót előléptetni adminisztrátor felhasználóvá és lehetőség van egy adminisztrátor felhasználót lefokozni egyszerű felhasználóvá.

#### 3.2.3. Biztonság

A szerverre beérkező minden kérés átmegy egy autentikációs middleware-n, ami ellenőrzi, hogy a kéréshez tartozik-e egy jwt token a felhasználónévvel és, ha igen, akkor az adatbázisban ellenőrzi a felhasználóhoz tartozó jogokat és aszerint beállítja a jogosultságát. Ez a middleware-auth.js fájlban található meg. A felhasználók bejelentkezés vagy regisztráció után kapnak egy jwt tokent.

A bejelentkezéshez és a regisztrációhoz tartozó kérések kiszolgálása a routes/authRouter.js fájlban található meg. A webes kliens egy post kérést küld a szerverre az api/login vagy az api/register címre és a body-ban elküldi a felhasználónevet és a jelszavát json formátumban. Az alábbi példában láthatjuk a regisztráció menetét (3.1 ). A szerver ellenőrzi, hogy megvan-e a felhasználónév és a jelszó, létezik-e már egy felhasználó ugyanezzel a névvel valamint, hogy a megadott jelszavak egyeznek-e, ha nem, akkor egy megfelelő válaszüzenetet küld vissza 401-es státusz kóddal. A 401-es kód az Unauthorized, vagyis jogosulatlan megfelelője, amit akkor használunk ha az ügyfélkérelem nem fejeződött be, mert hiányoznak az érvényes hitelesítési adatok a kért erőforráshoz. Ezután a jelszót egy 10 karakter hosszúságú random sóval együtt hasheli a bcrypt csomagot felhasználva. A bcrypt egy Niels Provos és David Mazières által tervezett, a Blowfish titkosításon alapuló jelszó-hashelő függvény, amelyet 1999-ben mutattak be a USENIX -en. A jelszó hashelése után beszúrja az új felhasználó nevét és a jelszavát az adatbázisba. Ezután létrehozza a jwt tokent, ami a felhasználó nevét tartalmazza egy az oldalhoz tartozó egyedi karakterlánc segítségével. A szerver lekéri az adatbázisról a felhasználó jogosultság csoportjának a nevét és ha minden jól ment elküldi a választ json-ben. Ha az adatbázissal kapcsolatban valami hiba lépne fel, akkor egy 500-as hibát térít vissza, ami a belső szerver hibának a státusz kódja.

A JSON Web Token (JWT) az RFC 7519-ben definiált standard, ami két fél közötti kompakt, biztonságos információcserét biztosít JSON objektumok formájában. Az aláírt jwt annak szavatolása, hogy az elküldött adatok megbízható forrásból származnak, és nem voltak menet közben módosítva. A token 3 részből áll, amiket pontok választanak el egymástól: fejléc, pay-

### 3. FEJEZET: AZ ALKALMAZÁS FELÉPÍTÉSE

load és az aláírás. A fejléc tartalmazza a hashelő algoritmus nevét és a token típusát, a payload tartalmazza a hasznos információt, a jelen alkalmazás esetében a felhasználónevet, az aláírás generálásához pedig a fejlécben megadott algoritmust alkalmazzuk a fejléc.payload karakterláncra egy titkos kulcs segítségével. A szerver ismeri a titkos kulcsot, ennek segítségével lehet biztos benne, hogy a token valóban érvényes ([Jwt]).

#### 3.2.4. A szerver kitelepítése

A szerver kitelepítését a heroku segítségével oldottuk meg, ahol a szervert egy dockerfile segítségével állítja fel a forráskódok segítségével. A Heroku egy felhőplatform szolgáltatásként (PaaS) működik, amely számos programozási nyelvet támogat. A heroku volt az egyik első felhőplatform, a fejlesztésének 2007 júniusában kezdtek neki. A kezdeti időkben csak a Ruby programozási nyelvet támogatta, de most már egyre több programozási nyelvet támogat, például az itt használt Node.js-t is. Emiatt a Heroku-ról azt mondják, hogy egy többnyelvű platform, mivel rendelkezik fejlesztői funkciókkal alkalmazások létrehozásához, futtatásához és méretezéséhez a legtöbb nyelven hasonló módon. A Heroku-t 2010-ben vásárolta meg a Salesforce 212 millió dollárért. A Heroku hálózat virtuális konténerekben futtatja az ügyfél alkalmazásait, amelyek megbízható futási környezetben futnak. Heroku ezeket a konténereket "Dynosnak" nevezi. Ezek a Dynos Node, Ruby, PHP, Go, Scala, Python, Java vagy Clojure nyelven írt kódokat futtathatnak. A Heroku egyéni buildcsomagokat is biztosít, amelyekkel a fejlesztő bármilyen más nyelven telepíthet alkalmazásokat. A Heroku lehetővé teszi a fejlesztő számára az alkalmazás azonnali méretezését, csupán a dynók számának növelésével vagy az alkalmazás futtatható dinamó típusának megváltoztatásával ([Her]).

A dockerfile egy node 14.16.0 docker image alapján hozza létre a szervert. Kiválasztja a használt mappát, ahová bemásolja a package.json fájl tartalmát és az npm csomagkezelő segítségével telepíti a szükséges függőségeket. Ezután beállítja, hogy hányas porton működjön és elindulhat az alkalmazás. A herokunak szüksége van egy leírásra, hogy hol keresse ezt a dockerfile-t, ez a heroku.yml fájlban van pontosítva ([Doc, a]).

### 3. FEJEZET: AZ ALKALMAZÁS FELÉPÍTÉSE

```
router.post('/register', async (req, resp) => {
  try {
    const {username, password, password2} = req.body.data;
    if (username && password && password2) {
      if (password === password2) {
        const user = await db.passwordByUsername(username);
        if (!user) {
          const hashedPassword = await bcrypt.hash(
            password,
            bcrypt.genSaltSync(10),
          );
          await insertDb.insertUserUser(username, hashedPassword);
          const token = jwt.sign({username}, SECRET);

          const permission = await db.userRole(username);
          resp.json({token, permission, username});
        } else {
          const message = 'Username already exist!';
          resp.status(401).json(message);
        }
      } else {
        const message = 'The two password is not matching!';
        resp.status(401).json(message);
      }
    } else {
      const message = 'There is no username or password!';
      resp.status(401).json(message);
    }
  }
  catch (err) {
    console.log(err);
    const errorMessage = err.message;
    const obj = {};
    obj.errorMessage = errorMessage;
    const jsonString = JSON.stringify(obj);
    resp.status(500).json(jsonString);
  }
});
```

#### 3.1. kódrészlet. regisztráció

### 3.3. Mobil alkalmazás

A mobilalkalmazás react native technológiával lett megírva. A mobilalkalmazásban lefényképezhetünk egy terméket és erről a fényképről az alkalmazás optikai karakterfelismerés segítségével meghatározza a termék összetevőit. Ezután megtekinthetjük az összetevők részleteit, amit az alkalmazás a szerverről kér le. A termékeket el tudjuk menteni a helyi adatbázisba is (3.3 ).

#### 3.3.1. Adatmodell a mobilalkalmazásban

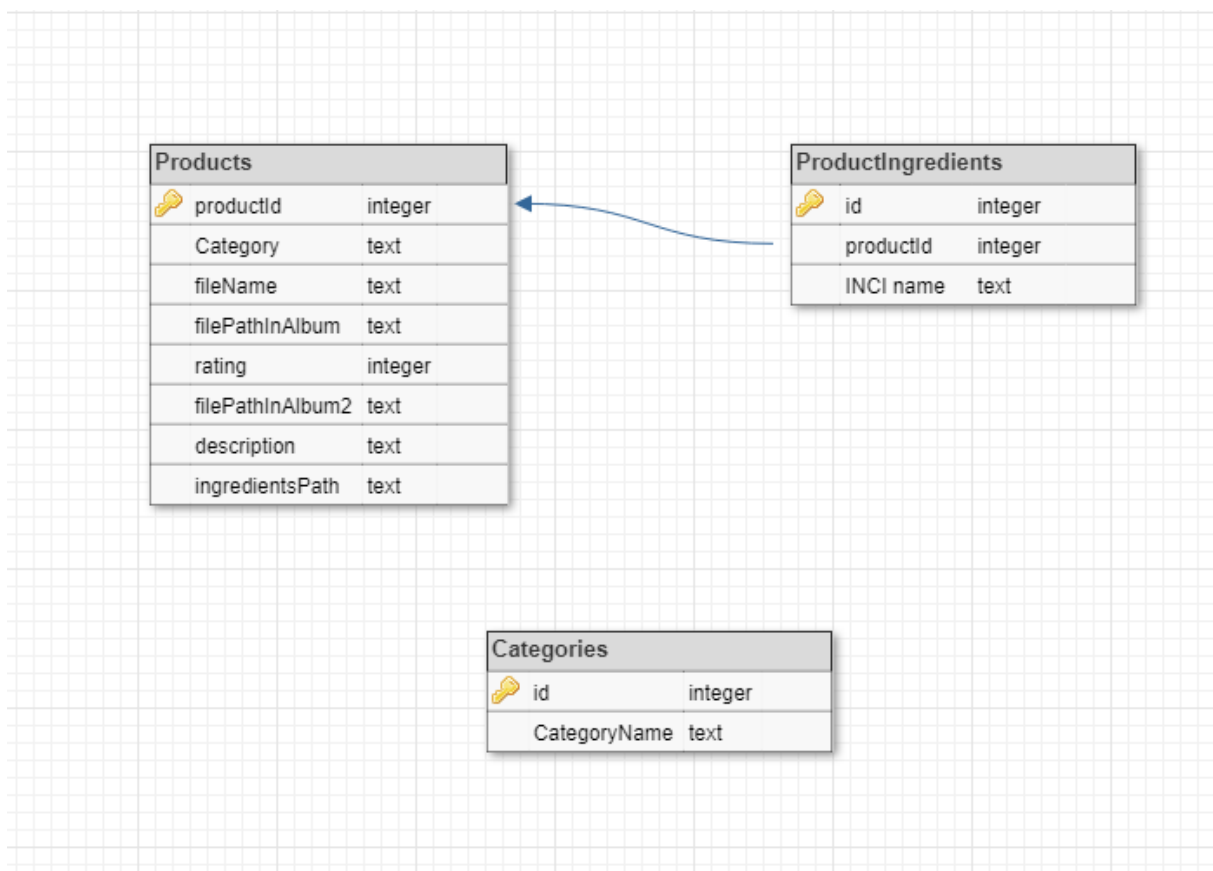
A mobilalkalmazás adatbázisa az alkalmazás telepítésekor egy üres adatbázisból jön létre. Ez az adatbázis eltárolja a felhasználó által elmentett termékeket és a hozzá tartozó adatokat, a kategóriákat, valamint a termékekhez tartozó összetevőket is. Ez az adatbázis 3 táblából áll: Products, ProductIngredients és Categories (3.3 ).

A Products táblába lesznek elmentve a felhasználó által elmentett termékek, minden termékhez tartozik egy productid, ami automatikusan létrejön, az adatbázis ez alapján tudja azonosítani a termékeket. A termékek elmentésekor a kötelezően megadott képet elmenti és az adatbázisban filePathInAlbum néven lementi az kép elérési útvonalát, a termék nevét fileName, a termék kategóriáját pedig Category néven menti el. A termékek elmentésekor a termékhez opcionálisan hozzá lehet adni: egy értékelést, 1-től 5-ig terjedő skálán, ez rating néven mentődik el, egy képet a termékről egy más pozícióban, ezt a képet is elmenti és az adatbázisban az elérési útvonalát a filePathInAlbum2 néven menti el, egy leírást az adott termékről, ez pedig description néven mentődik el. Az alkalmazás minden terméknek létrehoz egy txt fájlt, aminek az elérési útvonalát ingredientsPath néven menti el.

A ProductIngredients táblába lesznek elmentve minden termékhez az összetevők, a táblának van egy id-ja, ami az azonosításra szolgál, a termékekhez tartozó összetevőknek a hivatalos INCI nevei lesznek elmentve INCI name néven. A tábla a pructId-n keresztül több az egyhez kapcsolattal kapcsolódik a Products táblához.

A Categories táblába a felhasználó által létrehozott kategóriák nevei mentődnek el.

### 3. FEJEZET: AZ ALKALMAZÁS FELÉPÍTÉSE



3.3. ábra. A mobilalkalmazás adatbázisa.

## 3.4. Webes kliens

A webes kliens egy felület a felhasználók számára, amely az adatokat megjeleníti és a szervertel is kommunikációt folytat. A web kliens Reactban lett elkészítve. Ebben a fejezetben lesz szó a webes kliens felépítéséről és annak a megvalósításáról.

### 3.4.1. Komponensek

A React használatával a weboldalunk megjelenítését komponensekre, akár egymástól független apróbb részekre bonthatjuk fel. A komponenseket többféle módon létre lehet hozni, ebben az alkalmazásban a kliens oldal funkcionális komponensekből áll.

Minden oldal fölött elhelyezkedik egy navigációs menü, ami segít átváltani egyik oldalról a másikra. Itt található még egy gomb is, ami ha be vagyunk jelentkezve egy Logout felirattal jelenik meg és a rákattintása után kijelentkezünk a felületről, ha nem vagyunk bejelentkezve akkor egy Login/Register felirat jelenik meg, ami a bejelentkezési oldalra vezet át. Ha be vagyunk jelentkezve itt láthatjuk a felhasználónevünket is.

A főoldalon az információs oldalt láthatjuk, ami InformationPage néven található meg. Ez a komponens két részből áll, a felső részében található meg kereső, amivel név alapján kereshetünk, az adatok minden betű lenyomása után frissülnek. Attól függően, hogy be vagyunk-e jelentkezve megjelenhet egy gomb egy új összetevő hozzáadására. A gomb hatására előugrik egy ablak ahová beírhatjuk az összetevő adatait és így elmenthetjük. Ez modál segítségével van megoldva ([Mod]). A modálok külön ablakok egy alkalmazáson belül, leggyakrabban párbeszédpanelként. Ezek egy általános felhasználói felületi minta információszolgáltatáshoz vagy megerősítéshez. Az alsó komponensben található egy táblázat, itt jelennek meg a szerverről lekért adatok. Attól függően, hogy be vagyunk-e jelentkezve, minden sor végén megjelenhet két gomb, a szerkesztéshez és a törléshez, ezek hatására is előugró ablakok jelennek meg, ahol az adatokat szerkeszthetjük, vagy megerősíthetjük a szándékunkat, ezek is modálok segítségével vannak megvalósítva. Az oldal alján található az oldalváltási sáv, aminek a segítségével kiválaszthatjuk, hogy hányadik oldal tartalmát szeretnénk megnézni. Lehetőségünk van az első és utolsó oldalra ugrani, emellett választhatjuk az előző vagy a következő oldalt, de akár haladhatunk kettesével is.

Ha nem vagyunk bejelentkezve, akkor a főoldal mellett csak a bejelentkezési oldal érhető el, ez LoginPage névvel van jelen, ahol egymás mellett megtalálható a regisztrálási és a bejelentke-

### 3. FEJEZET: AZ ALKALMAZÁS FELÉPÍTÉSE

zési lehetőség. Bejelentkezés vagy sikeres regisztráció esetén átkerülünk a főoldalra, ha valami hiba adódik azt egy párbeszédablak segítségével tudjuk meg, ami itt is egy modál segítségével lett megvalósítva.

Ha egy egyszerű felhasználóval vagyunk bejelentkezve, akkor csak annyit látunk, mint vendégként, viszont ha egy adminisztrátor felhasználóval, akkor megjelenik a navigációs menü két újabb oldal. Az egyik az adatbázis cseréjére szolgáló oldal, ez az UploadPage, ahol egy csv fájlt tölthetünk fel a szerverre, ezzel lecserélve az adatbázis. Feltöltés után itt is kapunk egy válaszüzenetet.

Az utolsó oldal a felhasználók kezelésére szolgál, ennek az oldálnak UsersPage a neve, itt egy listában láthatjuk a felhasználókat és minden felhasználó esetén lehetőségünk van megváltoztatni a rangjukat: adminisztrátor felhasználó egyszerű felhasználóvá és fordítva. Emellett az egyszerű felhasználókat lehetőségünk van törölni is, minden művelet után egy párbeszédablak segítségével tudhatjuk meg, hogy a módosításunk sikerrel járt-e.

#### 3.4.2. API kliens

A szerver és a komponensek közötti kommunikáció a REST szabványnak megfelelően történik. A kliens a kommunikációt axios kérésekkel végzi ([Axi]).

A szerver többfajta HTTP kérést képes fogadni, ilyenek a GET, POST, PUT, PATCH és a DELETE. A kérések autorizációját a szerver egy jwt token segítségével valósítja meg, amit a kliens a szervertől kap meg a bejelentkezés vagy a regisztráció után és elmenti a felhasználónévvel és a jogosultsággal együtt. Azok a hívások, amelyeket csak az adminisztrátor felhasználók végezhetnek jwt token megadását igénylik, amit a kliens a kérés fejlécébe állít be, az Authorization mezőbe. A következő példában láthatjuk a regisztráció menetét kliens oldalról, amit szerver oldalon már láthattunk. A kliens küld egy post kérést az api/auth/register címre, a kérés törzsében pedig elküldi a felhasználónevet, a jelszót és a jelszó megerősítését. Sikeres regisztráció esetén a szerver egy 200-as státusz kódot térít vissza, ebben az esetben elmentődik a jwt token, a felhasználónév és a jogosultság és a felhasználót átkerül a főoldalra. Sikertelen regisztráció esetén egy hibaüzenetet kapunk, ahol megtudhatjuk a hiba okát.

### 3. FEJEZET: AZ ALKALMAZÁS FELÉPÍTÉSE

```
await axios
  .post('/api/auth/register', {
    data: {
      username: inputTextName,
      password: inputTextPasswordRegister,
      password2: inputTextPasswordRegisterRepeat,
    },
  })
  .then(resp => {
    if (resp.status === 200) {
      const token = resp.data.token;
      const permission = resp.data.permission;
      const username = resp.data.username;
      localStorage.setItem('token', token);
      localStorage.setItem('permission', permission);
      localStorage.setItem('username', username);
      setCookie('token', token, {path: '/'});
      window.location.href = '/';
    }
  })
  .catch(function (error) {
    setAction('Register');
    setMessage(error.response.data);
    setShow(true);
    setInputTextPasswordRegister('');
    setInputTextPasswordRegisterRepeat('');
  });
```

#### 3.2. kódrészlet. regisztráció

#### 3.4.3. A kliens kitelepítése

A kliens kitelepítése a szerverhez hasonlóan itt is a heroku által lett kitelepítve, egy docker-file segítségével.



## 4. fejezet

# Felhasznált technológiák

***Összefoglaló:** Ez a fejezet ismerteti a felhasznált technológiákat.*

### 4.1. Szerver oldali technológiák

A szerver oldali komponens node.js segítségével lett megírva. Az adatok tárolására pedig az SQLite adatbázis-kezelő rendszer lett használva.

#### 4.1.1. Node.js

A Node.js, rövidebben csak Node egy nyílt forráskódú fejlesztői platform JavaScript kód szerveroldali végrehajtására. A Node szerveret általában akkor ajánlott használni, amikor a böngésző és a szerver között állandó kapcsolatra van szükség, ilyenek lehetnek például a chat alkalmazások vagy a hírfolyamok ([Nod]).

A Node a Chrome V8-as JavaScript motorjára épül. A V8 a Google nyílt forráskódú, nagy teljesítményű JavaScript és WebAssembly motorja, amely C++ nyelven íródott. Az amerikai Ryan Dahl alkotta meg a Node.js-t, amit 2009 novemberében mutatott be European JSConf-on ([V8]).

A Node.js lehetővé teszi a JavaScript kód futtatását állományokból és interaktív konzolon keresztül is. A Node csak egyetlen szálát használ, emellett aszinkron módon fut és eseményvezérelt, ezért hatékony és előnyös sok bemeneti és kimeneti (I/O) művelet esetén. A Node eltér a hagyományos modelltől, egy eseményveremben dolgozza fel a bejövő kéréseket, és a saját kéréseit egymás után küldi el, a válaszok megvárása nélkül. Ezzel szemben a hagyományos modellek több szálát futtatnak párhuzamosan és minden szál a tovább lépés előtt megvárja a hozzá érkező választ.

### 4.1.2. MySQL

A MySQL az egyik legismertebb napjainkban használt adatbáziskezelő-rendszer. A gyakran a legnépszerűbb adatbázisnak nevezett és jelenleg széles körben elterjedt, hatékony felhasználást iparágtól függetlenül, egyértelmű, hogy mindenkinek, aki vállalati adatokkal vagy általános informatikával foglalkozik, törekednie kell a MySQL alapvető ismereteire.

A MySQL segítségével még a relációs rendszerekben újoncok is azonnal gyors, hatékony és biztonságos adattároló rendszereket építhetnek. A MySQL programozott szintaxisa és felületei tökéletes átjárót jelentenek más népszerű lekérdezési nyelvek és strukturált adattárak világába is.

A MySQL az Oracle által kifejlesztett relációs adatbázis-kezelő rendszer (RDBMS), amely strukturált lekérdezési nyelven (SQL) alapul ([Ora] ).

A MySQL számos legnépszerűbb szoftvercsomag szerves részét képezi az ügyfeleknek szóló webalkalmazásoktól kezdve a hatékony, adatvezérelt B2B szolgáltatásokig . Nyílt forráskódú természete, stabilitása és gazdag szolgáltatáskészlete, valamint az Oracle folyamatos fejlesztése és támogatása azt eredményezte, hogy az internet szempontjából kritikus fontosságú szervezetek, mint például a Facebook, a Flickr, a Twitter, a Wikipedia és a YouTube, mind MySQL-hátterendszereket alkalmaznak ([MyS] ).

## 4.2. Mobil technológiák

A mobilalkalmazás react native technológiával lett megírva és az adatok tárolására az SQLite adatbázis-kezelő rendszert használja.

### 4.2.1. React native

A react native egy nyílt forráskódú JavaScript keretrendszer, amelyet arra terveztek, hogy több platformra, például iOS-re, Androidra, valamint webes felületre renderelje le ugyanazt a forráskódot. A react native a React-on alapul, ami Facebook felhasználói felületek létrehozására szolgáló JavaScript-könyvtáron alapul, de a böngésző helyett a mobilplatformokat célozza meg ([Nat]).

### 4.2.2. Szövegfelismerés

Az összetevők OCR <sup>1</sup> alapú felismeréséhez a react-native-text-recognition könyvtárat használtuk. Ez a könyvtár az IOS alapú működéshez a Vision keretrendszert, az android alapú működéshez pedig a Firebase ML (Machine learning) csomagot használja fel.

Az Apple az iOS 11 részeként vezette be a Vision keretrendszert, amelynek segítségével olyan feladatokat hajthatunk végre, mint például az arcfelismerés, a vonalkódfelismerés, a képregisztráció és az általános jellemzők követése. A Vision az egyéni Core ML modellek használatát is lehetővé teszi olyan feladatokhoz, mint az osztályozás vagy az objektumok észlelése. Az iOS 15-ben az Apple bevezetett egy API-t a Vision keretrendszerben a személyszegmentálás végrehajtására, amely a portré módot is támogatja ([Vis]).

A Firebase a Google által mobil- és webalkalmazások létrehozására kifejlesztett platform. Eredetileg egy független vállalat volt, amelyet 2011-ben alapítottak. 2014 -ben a Google megvásárolta a platformot, és mára ez a zászlóshajója az alkalmazásfejlesztés terén. A Firebase Machine Learning egy mobil SDK, amely egy hatékony és könnyen használható csomag, amelynek a segítségével Android- és Apple-alkalmazásokban is elvégezhetjük az arcfelismerést, a vonalkódfelismerést, a képregisztrációt, a tájékoztató pontok felismerését és az általános jellemzők követését is. Az ML Kit olyan API-kat is biztosít, amelyek segítenek az egyéni TensorFlow Lite modellek mobilalkalmazásban való használatában ([Fir]).

### 4.2.3. SQLite

Az SQLite egy beágyazott, fájl alapú RDBMS <sup>2</sup>, amely nem igényel telepítést vagy beállítás. Ez viszont azt jelenti, hogy az alkalmazás nem fut külön szerverfolyamat alatt, amelyet el kell indítani, le kell állítani vagy konfigurálni kell. Ez a szerver nélküli architektúra lehetővé teszi, hogy az adatbázis platformok közötti kompatibilis legyen.

Az SQLite Database egy az Android rendszerben is biztosított nyílt forráskódú adatbázis, amely a felhasználó eszközén belüli adatok tárolására szolgál szöveges fájl formájában. Nagyon sok műveletet végezhetünk ezekkel az adatokkal, például új adatok hozzáadásával, frissítésével, kiolvasásával és törlésével. Az SQLite egy offline adatbázis, amely helyileg a felhasználó

---

<sup>1</sup>Az OCR az angol "Optical character recognition" vagy "optical character reader" rövidítése, ami az optikai karakterfelismerést vagy az optikai karakterolvasást jelenti

<sup>2</sup>Az RDBMS az angol "relational database management system" rövidítése, ami a relációs adatbázis-kezelő rendszert jelenti

## 4. FEJEZET: FELHASZNÁLT TECHNOLÓGIÁK

eszközén van tárolva, és nem kell semmilyen kapcsolatot létrehoznunk az adatbázishoz való csatlakozáshoz.

Az adatok az SQLite adatbázisban táblázatok formájában tárolódnak. Amikor az adatokat az SQLite adatbázisunkban tároljuk, az adatok az Excel-táblázatokhoz hasonló táblázatokba rendeződnek ([SQL]).

### 4.3. Webes technológiák

#### 4.3.1. React

A React megalkotója Jordan Walke, aki a Facebooknál fejlesztette ki, és 2013-ban tette közzé. A React egy nyílt forráskódú JavaScript könyvtár, amelynek segítségével interaktív felhasználói felületeket készíthetünk. A React egy kliens oldali technológia, tehát a Model-View-Controller (MVC) felosztásban a V, azaz View elem valósítható meg a használatával. A használatával létrehozhatunk összetettebb felhasználói felületeket is, egymástól elkülönített komponensekre, alkotóelemekre bontással. Ennek a felbontásnak az az előnye, hogy egy komponens változtatása nem érinti a teljes alkalmazást. Lehetőségünk van nemcsak egymagában, hanem más könyvtárakkal együtt is használni, mint például az Angular JS([Rea]).

#### 4.3.2. Bootstrap

A Bootstrap az egyik legnépszerűbb HTML, CSS és JavaScript keretrendszer reszponzív és mobilbarát webhelyek fejlesztésére. Ez egy frontend keretrendszer, amelyet az egyszerűbb webfejlesztéshez használhatunk. Tartalmaz HTML és CSS alapú tervezősablonokat tipográfiához, űrlapokhoz, gombokhoz, táblázatokhoz, navigációhoz, modálokhoz és sok máshoz ([Boo, a]).

A Bootstrapet Mark Otto és Jacob Thornton fejlesztette ki a Twitternél. Nyílt forráskódú termékként 2011 augusztusában adták ki a GitHubon. 2014 júniusában a Bootstrap volt az első számú projekt a GitHubon([Boo, b]).

### 4.4. Eszközök

#### 4.4.1. Git és GitHub

A verziókezelő rendszer olyan szoftver, amely segíti a szoftverfejlesztőket az együttműködésben és a munkájuk teljes előzményeinek megőrzésében. Tárolhatja a fájlok és a forráskód módosításait. Minden alkalommal, amikor a felhasználó megváltoztatja a projektet, a verziókezelő rendszer átveszi a projekt állapotát és elmenti azokat. A projekt ezen különböző mentett állapotai verzióként ismertek. Például, ha a programozó webhelyet épít, akkor az 1-es verzióként kerül tárolásra. Később, ha a programozó újabb oldalt ad hozzá az adott webhelyhez, akkor ezeket a változásokat 2-es verzióként menti. Hasonlóképpen, a módosításokat verzióként menti a verziókezelő rendszerek. A Git és a Github két kifejezés, amely a verziókezeléshez kapcsolódik. A Github egy webalapú tárhelyszolgáltatás a Git verziókezelő tárházhoz. Olyan szolgáltatásokat nyújt, mint a forráskód kezelése és az elosztott verziókezelés, mint a Git. További funkciókat is tartalmaz. Ez biztosítja az egyes projektek hozzáférés-ellenőrzését, hibakövetését, szolgáltatáskéréseit és feladatkezelését ([Git, a] és [Git, b]).

#### 4.4.2. Docker

A Docker egy platform, mint szolgáltatás (PaaS) termékkészlet, amely a virtuális gépekkel szemben (mint például a VirtualBox) nem utánozza a hardware-t, csak az operációs rendszert, csak a folyamathoz szükséges könyvtárakat és segédprogramokat, ennek köszönhetően másodpercek alatt képes elindulni. A Dockert a Docker, Inc tette elérhetővé 2013-ban, 5 évvel a cég alapítása után ([Doc, b] ) .

A tárolók egymástól elszigeteltek, és saját szoftvereiket, könyvtáraikat és konfigurációs fájljaikat csomagolják; jól meghatározott csatornákon keresztül tudnak kommunikálni egymással. Mivel az összes tároló egyetlen operációs rendszermag szolgáltatásain osztozik, kevesebb erőforrást használnak fel, mint a virtuális gépek .

A Docker egy alkalmazást és függőségeit virtuális tárolóba tudja csomagolni, amely bármely Linux, Windows vagy macOS számítógépen futhat. Ez lehetővé teszi, hogy az alkalmazás számos helyen futhasson, például helyszíni, nyilvános vagy privát felhőben. Amikor Linuxon fut, a Docker a Linux kernel erőforrás-elszigetelő szolgáltatásait (például cgroups és kernel namespaces) és egy uniószerű fájlrendszert (például OverlayFS) használja, hogy lehetővé tegye a

#### 4. FEJEZET: FELHASZNÁLT TECHNOLÓGIÁK

konténerek egyetlen Linuxon belüli futtatását, elkerülve a virtuális gépek indításával és karbantartásával kapcsolatos többletköltséget. A macOS rendszeren futó DockerLinux virtuális gépet használ a tárolók futtatásához.

Mivel a Docker-tárolók könnyűek, egyetlen szerver vagy virtuális gép több tárolót is futtathat egyszerre. Egy 2018-as elemzés megállapította, hogy egy tipikus Docker-használati eset magában foglalja állomásonként nyolc konténer futtatását, és az elemzett szervezetek negyede 18 vagy több tárolót futtat ([Doc, a] ).

##### 4.4.3. MySQL Workbench

A MySQL Workbench egy vizuális adatbázis tervező és modellező hozzáférési eszköz a MySQL szerver relációs adatbázisához. Lehetővé teszi új fizikai adatmodellek létrehozását és a meglévő MySQL adatbázisok módosítását fordított/előre irányuló tervezési és változáskezelési funkciókkal. A MySQL workbench célja, hogy felületet biztosítson az adatbázisokkal való könnyebb és strukturáltabb munkavégzéshez ([Wor]).

##### 4.4.4. Visual Studio Code

A Visual Studio Code egy hatékony forráskód-szerkesztő, amely elérhető Windows, macOS és Linux rendszereken is. Beépített JavaScript-, TypeScript- és Node.js-támogatással rendelkezik, és gazdag ökoszisztémával rendelkezik más nyelvekhez és futtatókörnyezetekhez (például C++, Java, Python, PHP, Go, .NET). A Visual Studio Code az alkalmazás mindhárom részéhez: a szerverhez, a mobil klienshez és a webes klienshez is felhasználásra került ([VSC]).

## 5. fejezet

# Az alkalmazás működése

***Összefoglaló:** Ez a fejezet bemutatja a mobilalkalmazás valamint a webes kliens-alkalmazás funkcióit, és ezek működését.*

## 5.1. Mobil alkalmazás

### 5.1.1. Főoldal

Az alkalmazás megnyitásakor a főoldal jelenik meg, ahol két opció közül választhatunk (5.1 ábra). Készíthetünk egy fényképet, vagy kiválaszthatunk egy már létező képet a galériánkból. A képen egy kozmetikai terméknek a hátsó oldala kell megjelenjen, amin kivehetőek kell legyenek a termék összetevői. Miután kiválasztottunk egy képet, az alkalmazás átvezet az információs oldalra (5.2).

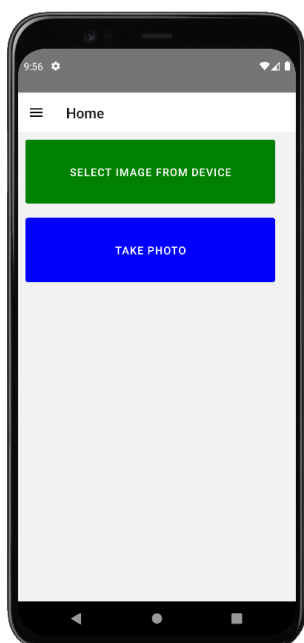
### 5.1.2. Információs oldal

Az információs oldalon az alkalmazás megjeleníti a termék képe alapján felismert összetevőket (5.2). Ha rákattintunk az egyik összetevőre, akkor előjön egy popup ablak, amin láthatjuk a kiválasztott összetevő részletes leírását (5.3). Az oldalon emellett kiválaszthatunk egy terméket, a már elmentett termékeink közül, és ebben az esetben az elmentett termékről nézhetünk meg információkat (5.4).

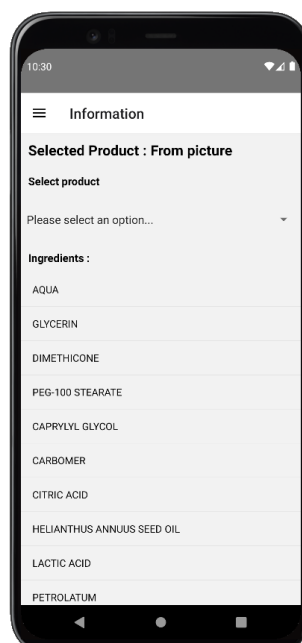
### 5.1.3. A termék elmentésére szolgáló oldal

Ezen az oldalon tudjuk elmenteni a korábban lefényképezett termékünket. Az oldal tetején láthatjuk a képet a termékről, amit el akarunk menteni, meg tudjuk adni a termék nevét és ki tudjuk választani, hogy milyen kategóriába tartozik, ha nem létezik egy nekünk megfelelő kategória, akkor hozzá tudunk adni a kategóriákhoz egy új kategóriát is (5.5). Ezenkívül meg

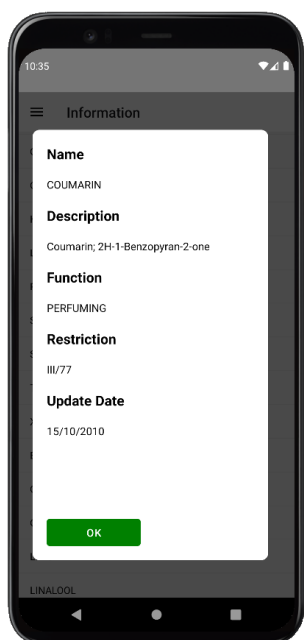
## 5. FEJEZET: AZ ALKALMAZÁS MŰKÖDÉSE



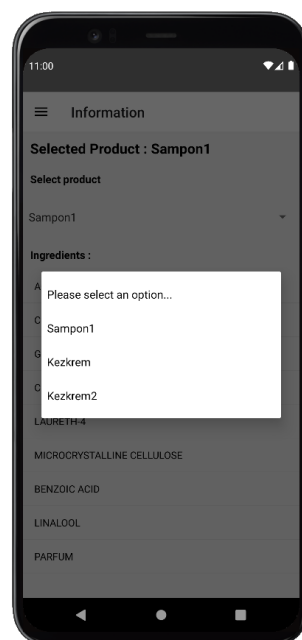
5.1. ábra. Főoldal



5.2. ábra. Információs oldal



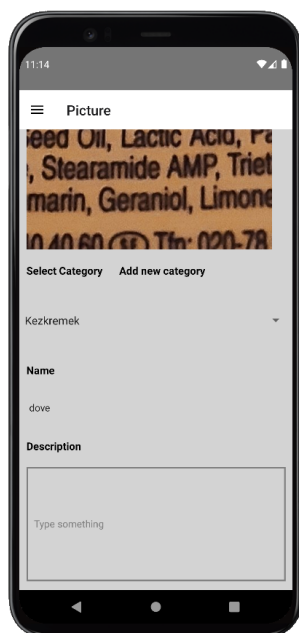
5.3. ábra. Az összetevő részletei



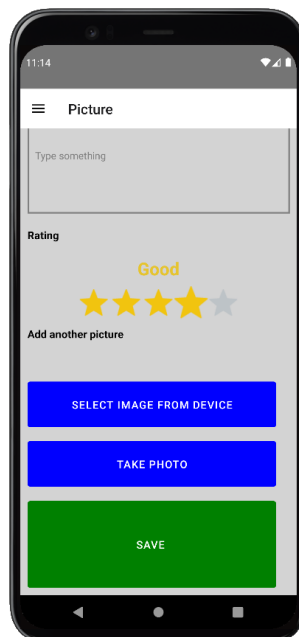
5.4. ábra. Termék kiválasztása



## 5. FEJEZET: AZ ALKALMAZÁS MŰKÖDÉSE



5.5. ábra. Termék mentése



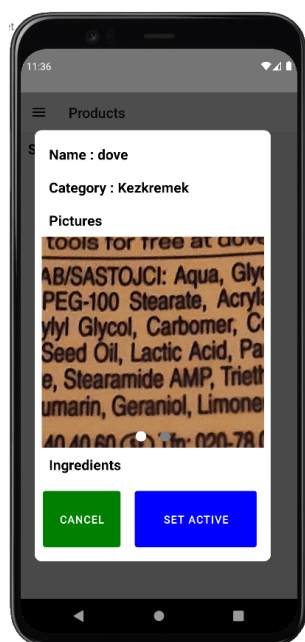
5.6. ábra. Termék mentése

tudunk adni egy leírást a termékről, tudjuk a terméket értékelni egy 1-től 5-ig terjedő skálán, hozzá tudunk adni egy újabb képet, amin a termék egy felismerhetőbb szögből van fényképezve (5.6). Ha elmentjük a terméket, az alkalmazás átvezet minket a termékek oldalára .

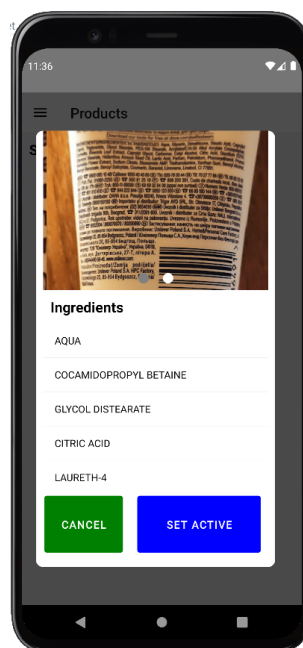
### 5.1.4. A termékek oldala

Ezen az oldalon tudjuk a már elmentett termékeinket megtekinteni (5.9). Minden termék mellett látjuk az értékelést, amit korábban kaptunk, és ha rákattintunk az egyik termékre, akkor előjön egy popup ablak, amiben meg tudjuk nézni a termékről lementett információkat (5.7 és 5.8 ). Itt ha a kiválasztott terméket aktívra állítjuk, akkor az alkalmazás átvezet az információs oldalra, és az kiválasztott termék összetevőinek a részletes információit ott tudjuk megtekinteni. Ha az egyik termékre hosszan rányomunk, akkor lehetőségünk van megosztani egy txt fájlt, ami a termék nevét és a termék összetevőit tartalmazza (5.10). Ezen az oldalon a termék oldalra húzásával kitörölhetjük azt az adatbázisból.

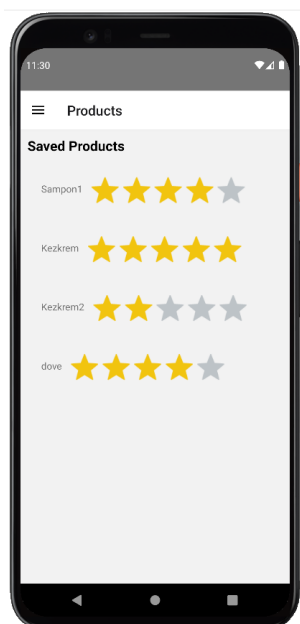
## 5. FEJEZET: AZ ALKALMAZÁS MŰKÖDÉSE



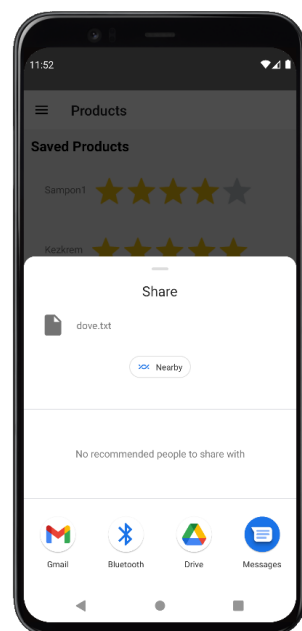
5.7. ábra. Lementett termék előnézete



5.8. ábra. Lementett termék részletei



5.9. ábra. Lementett termékek listája



5.10. ábra. Termék megosztása

5. FEJEZET: AZ ALKALMAZÁS MŰKÖDÉSE

5.2. Webes kliensalkalmazás

5.2.1. Főoldal

Ha megnyitjuk a webes alkalmazást akkor a főoldal tárul elénk (5.11), ahol egy táblázatban láthatjuk az összetevőket. Lapozhatunk az összetevők között, kereshetünk név alapján és abc szerinti sorrendbe is rendezhetjük a nevük szerint. Ugyanezen az oldalon, ha jogosultságunk engedi, megjelenik az új összetevő hozzáadása lehetőség, a törlés és a szerkesztés is (5.12). Új termék hozzáadása vagy szerkesztése esetén egy felugró ablak jelenik meg, ahova beírhatjuk az adatokat (5.13). Az oldal tetején található a navigációs menü, ahol váltogathatunk az oldalak között és a felhasználónevünket is láthatjuk.

Navbar Home Login/Register

Search

Ingredients

COSING Ref No	INCI name	INN name	Ph. Eur. Name	CAS No	EC No	Chem/IUPAC Name / Description	Restriction Function	Update Date	Id
94753	DISODIUM TETRAMETHYLHEXADECENYLCYSTEINE FORMYLPROUNATE			2040469-40-5, 2422121-34-2		Disodium Tetramethylhexadecenylcysteine Formylprolinate is the organic compound that conforms to the formula{	SKIN PROTECTING	2020-06-16	95607
99268	(ANGELICA ACUTILOBA/PAEONIA LACTIFLORA) ROOT/CNIDIUM OFFICINALE RHIZOME EXTRACT					(Angelica Acutiloba/Paeonia Lactiflora) Root/Cnidium Officinale Rhizome Extract is the extract of the roots of Angelica acutiloba and Paeonia lactiflora and the rhizomes of Cnidium officinale.	SKIN CONDITIONING - EMOLLIENT	2021-03-19	95608
99270	(ARCTIUM LAPPA/RAPHANUS SATIVUS/MALVA VERTICILLATA) SEED OIL					(Arctium Lappa/Raphanus Sativus/Malva Verticillata) Seed Oil is the volatile oil expressed from the seeds of Arctium lappa, Raphanus sativus, and Malva verticillata.	SKIN CONDITIONING - EMOLLIENT	2021-03-19	95609
94896	(LIQUIDAMBAR STYRACIFLUA/TRIBULUS TERRESTRIS) FRUIT/PAEONIA LACTIFLORA ROOT EXTRACT					(Liquidambar Styraciflua/Tribulus Terrestris) Fruit/Paeonia Lactiflora Root Extract is the extract of the fruit of Liquidambar styraciflua and Tribulus terrestris, and the roots of Paeonia lactiflora.	SKIN CONDITIONING	2017-12-01	95610
95645	ACRYLATES/AA/VINYL NEODECANOATE COPOLYMER			99728-55-9		Acrylates/AA/Vinyl Neodecanoate Copolymer is a copolymer of Vinyl Acetate, vinyl neodecanoate and one or more monomers consisting of acrylic acid, methacrylic acid or one of their simple esters.	PLASTICISER	2018-02-14	95611
96229	ASTROCARYUM VULGARE SEED BUTTER			2187449-53-0		Astrocaryum Vulgare Seed Butter is the fat obtained from the seeds of Astrocaryum vulgare, Palmae	SKIN CONDITIONING, SKIN CONDITIONING - EMOLLIENT	2018-04-06	95612
89177	BARLEY SH-POLYPEPTIDE-17					Barley sh-Polypeptide-17 is a single chain recombinant human peptide, produced by barley plants after using either a transient expression system or transgenesis (stable insertion of a copy of a human gene into its DNA). The starting gene is synthesized to be identical to a human gene which codes for Interleukin 1 Alpha. It contains a maximum of 272 amino acids, which may contain disulfide bonds and/or glycosylation. The protein consists of the proper sequence of the 20 standard amino acids.	HAIR CONDITIONING, SKIN CONDITIONING	2018-03-09	95613
100088	CITRULLUS LANATUS SEED POWDER			90244-99-8	290-802-9	Citrullus Lanatus Seed Powder is the powder obtained from the dried, ground seeds of Citrullus lanatus.	ABRASIVE, EXFOLIATING	2021-09-23	95614
98580	DAUCUS CAROTA SATIVA LEAF EXTRACT					Daucus Carota Sativa (Carrot) Leaf Extract is the extract of the leaves of Daucus carota sativa.	SKIN CONDITIONING - MISCELLANEOUS	2020-09-18	95615
89078	GOSSYPIMUM HIRSUTUM SEED EXTRACT			8001-29-4	232-280-7	Gossypium Hirsutum Seed Extract is the extract of the seeds of Gossypium hirsutum, Malvaceae.	SKIN CONDITIONING	2012-07-04	95616

First 1 2 3 -- Next Last

5.11. ábra. A főoldal

## 5. FEJEZET: AZ ALKALMAZÁS MŰKÖDÉSE

Navbar Home Upload Users Admin Logout									
Search									
Ingredients									
Add new ingredient									
COSING Ref No	INCI name	INN name	Ph. Eur. Name	CAS No	EC No	Chem./IUPAC Name / Description	Restriction Function	Update Date	Id
94753	DISODIUM TETRAMETHYLHEXADECENYL CYSTEINE FORMYLPROLINATE			2040469-40-5, 2422121-34-2		Disodium Tetramethylhexadecenylcysteine Formylprolinate is the organic compound that conforms to the formula{	SKIN PROTECTING	2020-06-16	95607
99268	(ANGELICA ACUTILOBA/PAEONIA LACTIFLORA) ROOT/CNIDIUM OFFICINALE RHIZOME EXTRACT					(Angelica Acutiloba/Paeonia Lactiflora) Root/Cnidium Officinale Rhizome Extract is the extract of the roots of Angelica acutiloba and Paeonia lactiflora and the rhizomes of Cnidium officinale.	SKIN CONDITIONING - EMOLLIENT	2021-03-19	95608
99270	(ARCTIUM LAPPA/RAPHANUS SATIVUS/MALVA VERTICILLATA) SEED OIL					(Arctium Lappa/Raphanus Sativus/Malva Verticillata) Seed Oil is the volatile oil expressed from the seeds of Arctium lappa, Raphanus sativus, and Malva verticillata.	SKIN CONDITIONING - EMOLLIENT	2021-03-19	95609
94896	(LIQUIDAMBAR STYRACIFLUA/TRIBULUS TERRESTRIS) FRUIT/PAEONIA LACTIFLORA ROOT EXTRACT					(Liquidambar styraciflua/Tribulus Terrestris) Fruit/Paeonia Lactiflora Root Extract is the extract of the fruit of Liquidambar styraciflua and Tribulus terrestris, and the roots of Paeonia lactiflora.	SKIN CONDITIONING	2017-12-01	95610
95645	ACRYLATES/VA/VINYL NEODECANOATE COPOLYMER			99728-55-9		Acrylates/VA/Vinyl Neodecanoate Copolymer is a copolymer of Vinyl Acetate, vinyl neodecanoate and one or more monomers consisting of acrylic acid, methacrylic acid or one of their simple esters.	PLASTICISER	2018-02-14	95611
96229	ASTROCARYUM VULGARE SEED BUTTER			2187449-53-0		Astrocaryum Vulgare Seed Butter is the fat obtained from the seeds of Astrocaryum vulgare, Palmae.	SKIN CONDITIONING, SKIN CONDITIONING - EMOLLIENT	2018-04-06	95612
89177	BARLEY SH-POLYPEPTIDE-17					Barley sh-Polypeptide-17 is a single chain recombinant human peptide, produced by barley plants after using either a transient expression system or transgenesis (stable insertion of a copy of a human gene into its DNA). The starting gene is synthesized to be identical to a human gene which codes for Interleukin 1 Alpha. It contains a maximum of 272 amino acids, which may contain disulfide bonds and/or glycosylation. The protein consists of the proper sequence of the 20 standard amino acids.	HAIR CONDITIONING, SKIN CONDITIONING	2018-03-09	95613
100088	CITRULLUS LANATUS SEED POWDER			90244-99-8	290-802-9	Citrullus Lanatus Seed Powder is the powder obtained from the dried, ground seeds of Citrullus lanatus.	ABRASIVE, EXFOLIATING	2021-09-23	95614
98580	DAUCUS CAROTA SATIVA LEAF EXTRACT					Daucus Carota Sativa (Carrot) Leaf Extract is the extract of the leaves of Daucus carota sativa.	SKIN CONDITIONING - MISCELLANEOUS	2020-09-18	95615
89078	GOSSYPOLIUM HIRSUTUM SEED EXTRACT			8001-29-4	232-280-7	Gossypium Hirsutum Seed Extract is the extract of the seeds of Gossypium hirsutum, Malvaceae.	SKIN CONDITIONING	2012-07-04	95616

5.12. ábra. A főoldal admin felhasználóként

Navbar Home Upload Users Admin Logout									
Search									
Ingredients									
Add new ingredient									
COSING Ref No	INCI name	INN name	Ph. Eur. Name	CAS No	EC No	Chem./IUPAC Name / Description	Restriction Function	Update Date	Id
94753	DISODIUM TETRAMETHYLHEXADECENYL CYSTEINE FORMYLPROLINATE			2040469-40-5, 2422121-34-2		Disodium Tetramethylhexadecenylcysteine Formylprolinate is the organic compound that conforms to the formula{	SKIN PROTECTING	2020-06-16	95607
99268	(ANGELICA ACUTILOBA/PAEONIA LACTIFLORA) ROOT/CNIDIUM OFFICINALE RHIZOME EXTRACT					(Angelica Acutiloba/Paeonia Lactiflora) Root/Cnidium Officinale Rhizome Extract is the extract of the roots of Angelica acutiloba and Paeonia lactiflora and the rhizomes of Cnidium officinale.	SKIN CONDITIONING - EMOLLIENT	2021-03-19	95608
99270	(ARCTIUM LAPPA/RAPHANUS SATIVUS/MALVA VERTICILLATA) SEED OIL					(Arctium Lappa/Raphanus Sativus/Malva Verticillata) Seed Oil is the volatile oil expressed from the seeds of Arctium lappa, Raphanus sativus, and Malva verticillata.	SKIN CONDITIONING - EMOLLIENT	2021-03-19	95609
94896	(LIQUIDAMBAR STYRACIFLUA/TRIBULUS TERRESTRIS) FRUIT/PAEONIA LACTIFLORA ROOT EXTRACT					(Liquidambar styraciflua/Tribulus Terrestris) Fruit/Paeonia Lactiflora Root Extract is the extract of the fruit of Liquidambar styraciflua and Tribulus terrestris, and the roots of Paeonia lactiflora.	SKIN CONDITIONING	2017-12-01	95610
95645	ACRYLATES/VA/VINYL NEODECANOATE COPOLYMER			99728-55-9		Acrylates/VA/Vinyl Neodecanoate Copolymer is a copolymer of Vinyl Acetate, vinyl neodecanoate and one or more monomers consisting of acrylic acid, methacrylic acid or one of their simple esters.	PLASTICISER	2018-02-14	95611
96229	ASTROCARYUM VULGARE SEED BUTTER			2187449-53-0		Astrocaryum Vulgare Seed Butter is the fat obtained from the seeds of Astrocaryum vulgare, Palmae.	SKIN CONDITIONING, SKIN CONDITIONING - EMOLLIENT	2018-04-06	95612
89177	BARLEY SH-POLYPEPTIDE-17					Barley sh-Polypeptide-17 is a single chain recombinant human peptide, produced by barley plants after using either a transient expression system or transgenesis (stable insertion of a copy of a human gene into its DNA). The starting gene is synthesized to be identical to a human gene which codes for Interleukin 1 Alpha. It contains a maximum of 272 amino acids, which may contain disulfide bonds and/or glycosylation. The protein consists of the proper sequence of the 20 standard amino acids.	HAIR CONDITIONING, SKIN CONDITIONING	2018-03-09	95613
100088	CITRULLUS LANATUS SEED POWDER			90244-99-8	290-802-9	Citrullus Lanatus Seed Powder is the powder obtained from the dried, ground seeds of Citrullus lanatus.	ABRASIVE, EXFOLIATING	2021-09-23	95614
98580	DAUCUS CAROTA SATIVA LEAF EXTRACT					Daucus Carota Sativa (Carrot) Leaf Extract is the extract of the leaves of Daucus carota sativa.	SKIN CONDITIONING - MISCELLANEOUS	2020-09-18	95615
89078	GOSSYPOLIUM HIRSUTUM SEED EXTRACT			8001-29-4	232-280-7	Gossypium Hirsutum Seed Extract is the extract of the seeds of Gossypium hirsutum, Malvaceae.	SKIN CONDITIONING	2012-07-04	95616

Edit

COSING Ref No

94753

INCI name

DISODIUM TETRAMETHYLHEXADECENYL CYSTEINE FORMYLPROLINATE

INN name

Ph. Eur. Name

CAS No

2040469-40-5, 2422121-34-2

EC No

Chem./IUPAC Name / Description

Disodium Tetramethylhexadecenylcysteine Formylprolinate is the organic compound that conforms to the formula{

Restriction Function

SKIN PROTECTING

Edit

Close

5.13. ábra. Egy összetevő szerkesztése

### 5.2.2. Bejelentkezési oldal

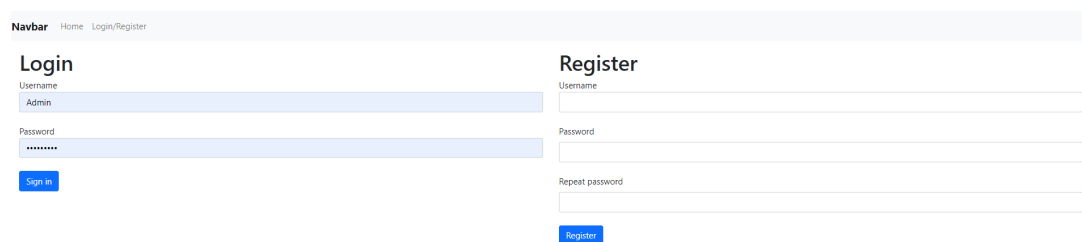
A bejelentkezési oldalon egymás mellett megtalálhatjuk a regisztrálási és bejelentkezési lehetőséget (5.14), ahonnan a sikeres bejelentkezés után a főoldalra kerülünk vissza.

### 5.2.3. Felhasználók kezelésének oldala

A felhasználók kezelésének oldalán látjuk az összes felhasználót, és lehetőségünk van a felhasználók rangjának megváltoztatására adminisztrátor felhasználóról egyszerű felhasználóra és fordítva, továbbá törölhetjük az egyszerű felhasználókat (5.15).

### 5.2.4. Az adatbázis feltöltésére szolgáló oldal

Ezen az oldalon egy gomb található, aminek a segítségével feltölthetünk egy csv fájlt a szerverre, ezáltal lecseréljük az összetevőkről szóló adatbázist. (5.16)



The image shows a web application interface with a light gray header bar. On the left, there is a 'Login' section with a 'Username' field containing the text 'Admin' and a 'Password' field with masked characters. Below these fields is a blue 'Sign in' button. On the right, there is a 'Register' section with a 'Username' field, a 'Password' field, and a 'Repeat password' field. Below these fields is a blue 'Register' button. The header bar contains the text 'Navbar' followed by links for 'Home' and 'Login/Register'.

5.14. ábra. A bejelentkezési oldal.

## 5. FEJEZET: AZ ALKALMAZÁS MŰKÖDÉSE

Navbar Home Upload Users Admin Logout

### Change user roles

Username: Zoard User role: user <input type="checkbox"/> Give admin role <input type="button" value="Delete user"/>
Username: Zoard2 User role: admin <input type="button" value="Demote user"/>
Username: Zoard3 User role: user <input type="checkbox"/> Give admin role <input type="button" value="Delete user"/>

5.15. ábra. A felhasználók kezelésének oldala. Itt listában láthatjuk rajtunk kívül az összes felhasználót. Láthatjuk a három különböző gombot, amivel a rangokat változtathatjuk és amivel felhasználót törölhetünk.

Navbar Home Upload Users Admin Logout

### Cosmetics database

Upload a csv file!

Nem lett kiválasztva fájl!

Choose before Pressing the Upload button

5.16. ábra. Az adatbázis feltöltésére szolgáló oldal. Itt két gombot láthatunk, az egyikkel kiválaszthatjuk a fájlt, amivel populálni fogjuk az adatbázist, a másikkal feltöltjük a szerverre.

## 6. fejezet

# Összefoglaló és továbbfejlesztési lehetőségek

A fejlesztés során sikerült létrehozni egy olyan szoftverrendszert, amelynek a segítségével a felhasználók a mobiltelefonjuk használatával megtekinthetik egy kozmetikai termék összetevőinek hatásait és így tudatosabban választhatják meg az általuk használt termékeket. A szoftverrendszer támogat több jogosultsági szerepkört is, ezáltal az adminisztrátoroknak lehetőségük van a felhasználók rangjának változtatására és az adatbázis módosítására.

A mobilalkalmazásban, amint a fenti fejezetben is olvasható, lehetőségünk van megtekinteni egy általunk lefényképezett termék összetevőit és az összetevők tulajdonságait, de ha változtatni szeretnénk ezen az adatbázison, akkor a webes alkalmazást kell használnunk. Egy továbbfejlesztési lehetőség lenne, ha a webes alkalmazásban elérhető funkciók, mint például a bejelentkezés és az adatbázis cseréje, változtatása a mobil alkalmazásban is elérhetővé válnának.

A webes alkalmazásban jelenleg az adatbázist csak az Európai Unió hivatalos adataival van lehetőségünk populálni ([Cos]). Egy továbbfejlesztési lehetőség lenne, ha többfajta adatot fel lehetne használni, például fel lehetne tölteni egyszerre több különböző adatot és ezeknek a kombinációjából jönne létre az adatbázisunk.

# Irodalomjegyzék

- 1source weboldala. <https://1source.com/>, utolsó megtekintés dátuma: 2022. augusztus 4.
- Axios leírása. <https://axios-http.com/docs/intro>, utolsó megtekintés dátuma: 2022. július 26.
- Bootstrap hivatalos weboldala. <https://getbootstrap.com/>, utolsó megtekintés dátuma: 2022. július 26., a.
- Bootstrap wikipédia oldala. [https://en.wikipedia.org/wiki/Bootstrap\\_\(front-end\\_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)), utolsó megtekintés dátuma: 2022. augusztus 4., b.
- Google cloud hivatalos weboldal. <https://cloud.google.com/>, utolsó megtekintés dátuma: 2022. július 25.
- Cosing adatbázis. <https://data.europa.eu/data/datasets/cosmetic-ingredient-database-ingredients-and-fragrance-inventory?locale=en>, utolsó megtekintés dátuma: 2022. július 30.
- Docker hivatalos weboldal. <https://www.docker.com/>, utolsó megtekintés dátuma: 2022. július 27., a.
- Docker wikipédia oldala. [https://en.wikipedia.org/wiki/Docker\\_\(software\)](https://en.wikipedia.org/wiki/Docker_(software)), utolsó megtekintés dátuma: 2022. augusztus 26., b.
- ML kit for firebase dokumentációja. <https://firebase.google.com/docs/ml-kit>, utolsó megtekintés dátuma: 2022. augusztus 25.
- Git hivatalos weboldala. <https://git-scm.com/>, utolsó megtekintés dátuma: 2022. július 26., a.
- Github hivatalos weboldala. <https://github.com/>, utolsó megtekintés dátuma: 2022. július 30., b.
- Heroku weboldala. <https://www.heroku.com/>, utolsó megtekintés dátuma: 2022. július 26.
- Inci beuty weboldala. <https://incibeauty.com/>, utolsó megtekintés dátuma: 2022. augusztus 4.
- Jwt hivatalos weboldala. <https://jwt.io/>, utolsó megtekintés dátuma: 2022. július 25.
- Modál leírása. <https://getbootstrap.com/docs/4.0/components/modal/>, utolsó megtekintés dátuma: 2022. július 26.
- Mysql hivatalos weboldal. <https://www.mysql.com/>, utolsó megtekintés dátuma: 2022. július 25.
- React native hivatalos weboldala. <https://reactnative.dev/>, utolsó megtekintés dátuma: 2022. július 26.
- Node hivatalos weboldala. <https://nodejs.org/en/>, utolsó megtekintés dátuma: 2022. július 26.



## IRODALOMJEGYZÉK

Oracle hivatalos weboldala. <https://www.oracle.com/>, utolsó megtekintés dátuma: 2022. július 26.

Pooling tervezési minta. <https://designpatternsphp.readthedocs.io/en/latest/Creational/Pool/README.html>, utolsó megtekintés dátuma: 2022. július 25.

React hivatalos weboldala. <https://reactjs.org/>, utolsó megtekintés dátuma: 2022. július 26.

Rest dokumentáció. <https://www.ibm.com/docs/en/control-desk/7.6.1.2?topic=apis-rest-api>, utolsó megtekintés dátuma: 2022. július 25.

Sqlite hivatalos weboldala. <https://www.sqlite.org/index.html>, utolsó megtekintés dátuma: 2022. július 26.

V8 hivatalos weboldala. <https://v8.dev/docs>, utolsó megtekintés dátuma: 2022. július 26.

Visual studio code hivatalos weboldala. <https://code.visualstudio.com/>, utolsó megtekintés dátuma: 2022. július 26.

Vision framework dokumentációja. <https://developer.apple.com/documentation/vision>, utolsó megtekintés dátuma: 2022. augusztus 25.

Mysql workbench hivatalos weboldala. <https://www.mysql.com/products/workbench/>, utolsó megtekintés dátuma: 2022. július 26.