

Záródolgozat feladatkiírás

Tanuló(k) neve: Iliás György Dávid, Varga Balázs Ádám,
Wittman Bence Tamás.

Képzés: nappali

Szak: 5 0613 12 03 Szoftverfejlesztő és tesztelő
technikus

A záródolgozat címe:

Salus - Healthy lifestyle

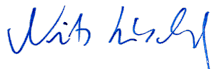
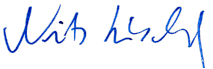
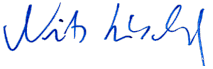
Konzulens: Nits László

Beadási határidő: 2023. 04. 28.

Győr, 2023. 04. 27

Módos Gábor
igazgató

Konzultációs lap

	A konzultáció		Konzulens aláírása
	ideje	témája	
1.	2023.02.15.	Témaválasztás és specifikáció	
2.	2023.03.14.	Zárodolgozat készültségi fokának értékelése	
3.	2023.04.28.	Dokumentáció véglegesítése	

Tulajdonosi nyilatkozat

Ez a dolgozat a saját munkánk eredménye. Dolgozatunk azon részeit, melyeket más szerzők munkájából vettünk át, egyértelműen megjelöltük.

Ha kiderülne, hogy ez a nyilatkozat valótlan, tudomásul vesszük, hogy a szakmai vizsgabizottság a szakmai vizsgáról kizár minket és szakmai vizsgát csak új zárodolgozat készítése után tehetünk.

Győr, 2023. 04. 27



tanuló aláírása



tanuló aláírása



tanuló aláírása

1. A vizsgaremekről.....	4
1.1 Alkalmazás leírása.....	4
1.2 Miért pont Salus - Healthy lifestyle?.....	5
1.3 Alkalmazás célja.....	5
1.4. Salus jövője.....	5
1.5 Elérhetőségei.....	5
2. Az adatbázis.....	7
2.1 Az "auths" táblázat:.....	7
2.2 A "comments" táblázat:.....	8
2.3 A "diets" táblázat:.....	9
2.4 A "followings" táblázat:.....	10
2.5 A "last24hs" táblázat:.....	11
2.6 Az "oils" táblázat:.....	12
2.7 A "recepieshavetags" táblázat:.....	12
2.8 A "recipes" táblázat:.....	13
2.9 A "recipesincludeingredients" táblázat:.....	14
2.10 A "tags" táblázat:.....	15
2.11 A "userprofiles" táblázat:.....	15
2.12 Az "usersprefertags" táblázat:.....	16
3. Munkamegosztás.....	17
3.1. Feladatok elosztása.....	17
3.2. Verziókezelés.....	17
3.3. A GitHub használata.....	18
3.4 Statisztikák.....	18
4. Backend.....	19
4.1 Használt technológiák.....	19
4.2 MySQL.....	20
4.3 .NET.....	21
4.4 Hosting.....	22
4.5 Tesztelés.....	23
5. Frontend.....	23
5.1 Használt Technológia.....	23
5.2 Angular.....	23
5.3 Rendszerkövetelmények.....	24
5.4 Alkalmazás szerkezete.....	25
6. Mobil.....	27
6.1 Használt Technológia.....	27
6.2 .NET MAUI.....	27
6.3 Alkalmazás szerkezete.....	28

7. Alap információk.....	29
7.1 Az alkalmazás áttekintése.....	29
7.2 Telepítési útmutató.....	30
7.3 Üzemeltetési útmutató.....	30
7.4 Bejelentkezési folyamat.....	31
7.4.1 Regisztráció.....	31
7.4.2 Hiba esetén.....	32
7.4.3 Bejelentkezés.....	33
7.4.4 Elfelejtett jelszó.....	34
8. Mobilalkalmazás.....	36
8.1 Az alkalmazás áttekintése.....	36
8.2 Telepítési útmutató.....	36
8.3 Üzemeltetési útmutató.....	36
Irodalmi jegyzék:.....	37

I. BEVEZETÉS

1. A vizsgaremekről

1.1 Alkalmazás leírása

A Salus egy olyan webalkalmazás és mobilalkalmazás, ahol kcal-t tud számolni a felhasználó, recepteket tud készíteni, mások receptjeit értékelni, követni a naponta bevitt tápanyag mennyiségét. Ezek mellett mások követésére, üzenőfalon kommunikálásra nyílik lehetősége. A funkciók használatához regisztráció szükséges, vagyis regisztráció nélkül a szolgáltatások egyike sem elérhető.

1.2 Miért pont Salus - Healthy lifestyle?

A név eredete hangzatos és sokat mondó, mégsem túl egyértelmű. Az ókori római mitológiában Salus a személyes egészség védelmezője. Emellett Latin értelmezésben a “salus” szó “egészség”-et jelent. Ezen információk tudatában már a “Salus” is eléggé beszédes név, viszont, hogy mindenki számára könnyen megtalálható legyen egy általános célú kereséssel is akár, hozzáadtuk a “Healthy lifestyle” kiegészítő alcímet.

1.3 Alkalmazás célja

A weboldalunk célja amellet, hogy a felhasználót segítse az egészséges életmód elérésében, fenntartásában, az az, hogy közösséget építsen ezen életmód köré.

1.4. Salus jövője

A jövőben szeretnénk kiterjeszteni az alkalmazás közösségi média oldalát és kidolgozni a felhasználói statisztikák elérhetőségét. Az alkalmazásunkat jövőbeli fejlesztésekre is előkészítettük, így naprakész tud maradni a koncepció.

1.5 Elérhetőségei

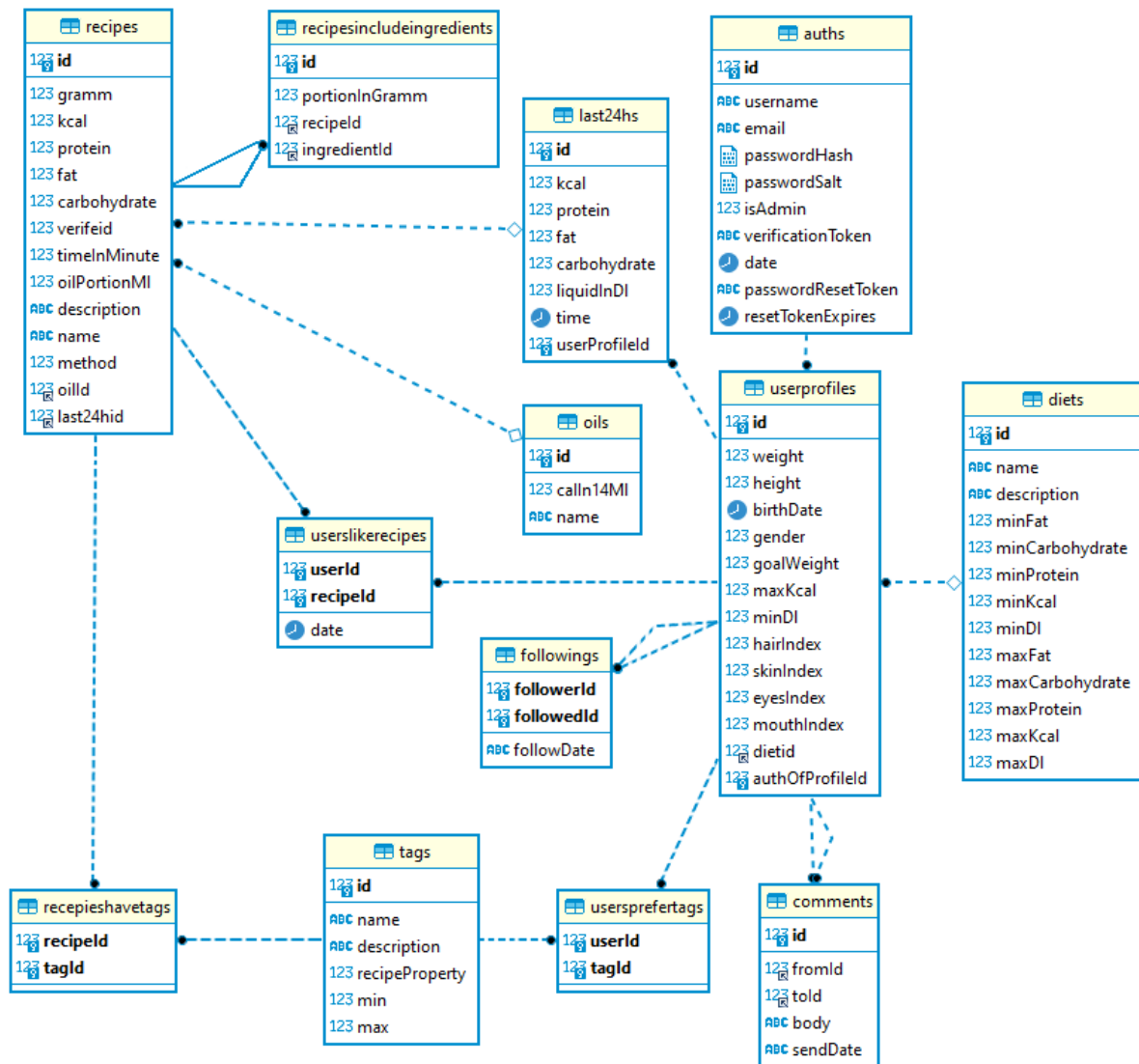
Frontend: <https://salus-healthy-lifestyle.netlify.app/>

Backend (Swagger): <https://salus.azurewebsites.net/swagger/index.html>

Mobil (APK): <https://github.com/14A-C-Salus/Others>

II. FEJLESZTŐI DOKUMENTÁCIÓ

2. Az adatbázis



1. Ábra. (Relációs modell)

2.1 Az “auths” táblázat:

- id**: A felhasználó **azonosítója**, egy egész szám (INT) típusú és nem lehet üres (NOT NULL). Az oszlop AUTO_INCREMENT tulajdonsággal rendelkezik,

ami azt jelenti, hogy az adatbázis automatikusan növeli az értékét minden új sor hozzáadásakor.

- username: A felhasználói név, hosszú szöveg (LONGTEXT) típusú és nem lehet üres (NOT NULL).
- email: A felhasználó e-mail címe, hosszú szöveg (LONGTEXT) típusú és nem lehet üres (NOT NULL).
- passwordHash: A felhasználó jelszavának hashelt verziója, hosszú bináris adat (LONGBLOB) típusú és nem lehet üres (NOT NULL).
- passwordSalt: A felhasználó jelszavának sója, hosszú bináris adat (LONGBLOB) típusú és nem lehet üres (NOT NULL).
- isAdmin: Egy logikai érték, amely azt jelzi, hogy a felhasználó rendelkezik-e adminisztrátori jogosultsággal. Egy egész szám (TINYINT(1)) típusú és nem lehet üres (NOT NULL).
- verificationToken: Az e-mail cím megerősítéséhez használt token, hosszú szöveg (LONGTEXT) típusú és alapértelmezés szerint üres (DEFAULT NULL).
- date: Az időpont, amikor a felhasználó létrejött, hosszú szöveg (DATETIME(6)) típusú és alapértelmezés szerint üres (DEFAULT NULL).
- passwordResetToken: Az jelszó-visszaállításra használt token, hosszú szöveg (LONGTEXT) típusú és alapértelmezés szerint üres (DEFAULT NULL).
- resetTokenExpires: Az jelszó-visszaállításra használt token lejárat dátuma, hosszú szöveg (DATETIME(6)) típusú és alapértelmezés szerint üres (DEFAULT NULL).

2.2 A “comments” táblázat:

- id: A komment **egyedi azonosítója**, amely egy egész szám (INT) típusú és nem lehet üres (NOT NULL). Az oszlop AUTO_INCREMENT tulajdonsággal rendelkezik, ami azt jelenti, hogy az adatbázis automatikusan növeli az értékét minden új sor hozzáadásakor.
- fromId: Az a felhasználó, aki a kommentet írta, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).

- **told:** Az a felhasználó, akihez a komment irányul, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **body:** A komment szövege, amelynek maximális hossza 1000 karakter (VARCHAR(1000)), és nem lehet üres (NOT NULL).
- **sendDate:** Az időpont, amikor a komment elküldésre került, hosszú szöveg (DATETIME(6)) típusú és nem lehet üres (NOT NULL).

Az adattáblának van egy **INDEX**-e (az "**fromId**", "**told**" és "**id**" oszlopokra), amelyek gyorsabb adatelérést biztosítanak a keresési műveletek során.

Az adattáblának van két idegen kulcsa (**FOREIGN KEY**), amelyek mind a "**userprofiles**" nevű táblára vonatkoznak, amelynek azonosító oszlopára (id) hivatkoznak. Az egyik idegen kulcs a "**fromId**" oszlopra hivatkozik, és az "**FK_comments_userProfiles_fromId**" nevet kapta, míg a másik a "**told**" oszlopra hivatkozik, és az "**FK_comments_userProfiles_told**" nevet kapta. Ezek a kulcsok biztosítják, hogy csak azon felhasználók azonosítói szerepeljenek az adattáblában, akik a "userprofiles" táblában szerepelnek.

2.3 A "diets" táblázat:

- **id:** Az étrend **egyedi azonosítója**, amely egy egész szám (INT) típusú és nem lehet üres (NOT NULL). Az oszlop AUTO_INCREMENT tulajdonsággal rendelkezik, ami azt jelenti, hogy az adatbázis automatikusan növeli az értékét minden új sor hozzáadásakor.
- **name:** Az étrend neve, hosszú szöveg (LONGTEXT) típusú és nem lehet üres (NOT NULL).
- **description:** Az étrend leírása, hosszú szöveg (LONGTEXT) típusú és nem lehet üres (NOT NULL).
- **minFat:** A minimális zsírbevitel, amelyet az étrend tartalmazhat, egész szám (INT) típusú és alapértelmezetten NULL értékű (DEFAULT NULL).
- **minCarbohydrate:** A minimális szénhidrát bevitel, amelyet az étrend tartalmazhat, egész szám (INT) típusú és alapértelmezetten NULL értékű (DEFAULT NULL).

- minProtein: A minimális fehérjebevitel, amelyet az étrend tartalmazhat, egész szám (INT) típusú és alapértelmezetten NULL értékű (DEFAULT NULL).
- minKcal: A minimális kalória bevitel, amelyet az étrend tartalmazhat, egész szám (INT) típusú és alapértelmezetten NULL értékű (DEFAULT NULL).
- minDI: A minimális folyadékbevitel, amelyet az étrend tartalmazhat, egész szám (INT) típusú és alapértelmezetten NULL értékű (DEFAULT NULL).
- maxFat: A maximális zsírbevitel, amelyet az étrend tartalmazhat, egész szám (INT) típusú és alapértelmezetten NULL értékű (DEFAULT NULL).
- maxCarbohydrate: A maximális szénhidrát bevitel, amelyet az étrend tartalmazhat, egész szám (INT) típusú és alapértelmezetten NULL értékű (DEFAULT NULL).
- maxProtein: A maximális fehérjebevitel, amelyet az étrend tartalmazhat, egész szám (INT) típusú és alapértelmezetten NULL értékű (DEFAULT NULL).
- maxKcal: A maximális kalóriabevitel, amelyet az étrend tartalmazhat, egész szám (INT) típusú és alapértelmezetten NULL értékű (DEFAULT NULL).
- maxDI: A maximális folyadékbevitel, amelyet az étrend tartalmazhat, egész szám (INT) típusú és alapértelmezetten NULL értékű (DEFAULT NULL).

2.4 A "followings" táblázat:

- followerId: Az a felhasználó, aki követi a másik felhasználót, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- followedId: Az a felhasználó, akit a követő felhasználó követ, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- followDate: Az időpont, amikor a követés létrejött, hosszú szöveg (DATETIME(6)) típusú és nem lehet üres (NOT NULL).

Az adattáblának van egy elsődleges kulcsa (**PRIMARY KEY**), amely a "**followedId**" és a "**followerId**" oszlopok kombinációjából áll. Ez biztosítja, hogy minden követés csak egyszer szerepeljen az adattáblában.

Az adattáblának van egy **INDEX**-e (az "**followerId**" oszlopra), amely segíti a gyorsabb adatelérést a keresési műveletek során.

Az adattábla két idegen kulcsa (**FOREIGN KEY**) van, mind a **"userprofiles"** nevű táblára vonatkoznak, amelynek azonosító oszlopára (id) hivatkoznak. Az egyik idegen kulcs a **"followedid"** oszlopra hivatkozik, és az **"FK_followings_userProfiles_followedid"** nevet kapta, míg a másik a **"followerid"** oszlopra hivatkozik, és az **"FK_followings_userProfiles_followerid"** nevet kapta. Ezek a kulcsok biztosítják, hogy csak azon felhasználók azonosítói szerepeljenek az adattáblában, akik a **"userprofiles"** táblában szerepelnek.

2.5 A "last24hs" táblázat:

- **id:** Az adatbázisban tárolt **egyedi azonosító**, amely egy egész szám (INT) típusú, nem lehet üres (NOT NULL), és az adatbázis automatikusan növeli az értékét minden új sor hozzáadásakor (AUTO_INCREMENT).
- **gramm:** Az elfogyasztott étel mennyisége grammokban, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **kcal:** Az elfogyasztott étel kalóriatartalma, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **protein:** Az elfogyasztott étel fehérjetartalma grammokban, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **fat:** Az elfogyasztott étel zsírtartalma grammokban, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **carbohydrate:** Az elfogyasztott étel szénhidrát tartalma grammokban, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **liquidInDI:** Az elfogyasztott folyadék mennyisége deciliterben (dl), egy egész szám (INT) típusú és alapértelmezés szerint NULL értékű (DEFAULT NULL).
- **time:** Az elfogyasztás időpontja, egy dátum és idő (DATETIME) típusú, nem lehet üres (NOT NULL), és 6 tizedesjegyet tartalmaz.
- **userProfileid:** Azon felhasználó azonosítója, aki az ételt elfogyasztotta, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **recipeid:** Az elfogyasztott étel azonosítója, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).

Az adattábla két **indexet** tartalmaz: az **"IX_last24Hs_recipeid"** és **"IX_last24Hs_userProfileid"** indexeket. Az indexek gyorsabb adatelérést biztosítanak a keresési műveletek során.

Az adattáblának van két idegen kulcsa (**FOREIGN KEY**), amelyek a **"userprofiles"** és a **"recipes"** nevű táblára vonatkozik. Az

"FK_last24Hs_userProfiles_userProfileId" idegen kulcs a **"userProfileId"** oszlopra hivatkozik, és biztosítja, hogy csak azon felhasználók azonosítói szerepeljenek az adattáblában, akik a **"userprofiles"** táblában szerepelnek. Az

"FK_last24Hs_recipes_recipId" idegen kulcs a **"recipId"** oszlopra hivatkozik, és biztosítja, hogy csak azon étel azonosítók szerepeljenek az adattáblában, amelyek a **"recipes"** táblában szerepelnek.

2.6 Az "oils" táblázat:

- **id**: Az olaj **egyedi azonosítója**, amely egy egész szám (INT) típusú és nem lehet üres (NOT NULL). Az oszlop AUTO_INCREMENT tulajdonsággal rendelkezik, ami azt jelenti, hogy az adatbázis automatikusan növeli az értékét minden új sor hozzáadásakor.
- **calln14MI**: Az olaj 14ml-ben található kalória mennyisége, amely egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **name**: Az olaj neve, amely hosszú szöveg (LONGTEXT) típusú és nem lehet üres (NOT NULL).

2.7 A "recepieshavetags" táblázat:

- **recipId**: A recept egyedi azonosítója, amely egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **tagId**: A címke (tag) egyedi azonosítója, amely egy egész szám (INT) típusú és nem lehet üres (NOT NULL).

A táblázat két oszlopból áll, amelyek a **"recipId"** és a **"tagId"**. Mindkét oszlop nem lehet üres (NOT NULL). A két oszlopból álló egyedileg azonosított elsődleges kulcs (**PRIMARY KEY**) a **"recipId"** és **"tagId"** oszlopok kombinációjából áll.

Az adattáblának van egy **indexe**, az **"IX_RecepiesHaveTags_tagId"**, amely a **"tagId"** oszlopra vonatkozik, és segíti az adatelérést a keresési műveletek során.

Az adattáblának két idegen kulcsa (**FOREIGN KEY**) van, amelyek mind a **"recipes"** és a **"tags"** táblára vonatkoznak, amelyeknek azonosító oszlopára (id) hivatkoznak.

Az egyik idegen kulcs a **"recipeId"** oszlopra hivatkozik, és az **"FK_RecepiesHaveTags_recipes_recipeId"** nevet kapta, míg a másik a **"tagId"** oszlopra hivatkozik, és az **"FK_RecepiesHaveTags_tags_tagId"** nevet kapta. Ezek a kulcsok biztosítják, hogy csak azon receptek és címkék azonosítói szerepeljenek az adattáblában, amelyek a **"recipes"** és a **"tags"** táblában szerepelnek.

2.8 A "recipes" táblázat:

- **id:** A recept **egyedi azonosítója**, amely egy egész szám (INT) típusú és nem lehet üres (NOT NULL). Az oszlop AUTO_INCREMENT tulajdonsággal rendelkezik, ami azt jelenti, hogy az adatbázis automatikusan növeli az értékét minden új sor hozzáadásakor.
- **gramm:** A recept összetevőinek mennyisége grammban, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **kcal:** A recept kalória tartalma, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **protein:** A recept fehérje tartalma grammban, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **fat:** A recept zsírtartalma grammban, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **carbohydrate:** A recept szénhidrát tartalma grammokban, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **verified:** Egy igaz-hamis érték (TINYINT(1)), amely jelzi, hogy a recept ellenőrzött-e.
- **timeInMinute:** A recept elkészítési ideje percekben, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **oilPortionML:** A receptben szereplő olaj mennyisége milliliter-ben, egy egész szám (INT) típusú, alapértelmezett értéke NULL.
- **description:** A recept leírása, hosszú szöveg (LONGTEXT) típusú és nem lehet üres (NOT NULL).
- **name:** A recept neve, hosszú szöveg (LONGTEXT) típusú és nem lehet üres (NOT NULL).
- **method:** A recept elkészítési módját reprezentáló szám, egy egész szám (INT) típusú és nem lehet üres (NOT NULL).

- **userProfileid**: Azon felhasználó azonosítója, aki létrehozta a receptet, egy egész szám (INT) típusú, alapértelmezett értéke NULL.
- **oilId**: Azon olaj azonosítója, amelyet a receptben használnak, egy egész szám (INT) típusú, alapértelmezett értéke NULL.

Az adattáblának van két **INDEX**-e (az "**oilId**" és "**userProfileid**" oszlopokra), amelyek felgyorsítják az adatelérést a keresési műveletek során.

Az adattáblának van két idegen kulcsa (**FOREIGN KEY**), amelyek a "**oils**" és a "**userprofiles**" nevű táblára vonatkozik. Az "**FK_recipes_oils_oilId**" idegen kulcs a "**oilId**" oszlopra hivatkozik, és biztosítja, hogy csak azon olajok azonosítói szerepeljenek az adattáblában, amik az "**oils**" táblában szerepelnek. Az "**FK_recipes_userProfiles_userProfileid**" idegen kulcs a "**userProfileid**" oszlopra hivatkozik, és biztosítja, hogy csak azon felhasználó azonosítók szerepeljenek az adattáblában, amelyek a "**userprofiles**" táblában szerepelnek.

2.9 A "recipesincludeingredients" táblázat:

- **id**: Az **egyedi azonosítója** annak az összetevőnek, amely az adott receptben szerepel. Ez egy egész szám (INT) típusú és nem lehet üres (NOT NULL). Az oszlop AUTO_INCREMENT tulajdonsággal rendelkezik, ami azt jelenti, hogy az adatbázis automatikusan növeli az értékét minden új sor hozzáadásakor.
- **portionInGramm**: Az adott összetevő mennyisége gramm-ban. Ez egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **recipeId**: Az azonosítója annak a receptnek, amelyben az adott összetevő szerepel. Ez egy egész szám (INT) típusú és nem lehet üres (NOT NULL).
- **ingredientId**: Az azonosítója annak az összetevőnek, amely az adott receptben szerepel. Ez egy egész szám (INT) típusú és nem lehet üres (NOT NULL).

Az adattábla két idegen kulccsal rendelkezik, mindkettő a "**recipes**" nevű táblára vonatkozik, amelynek azonosító oszlopára (**id**) hivatkozik. Az egyik idegen kulcs az "**ingredientId**" oszlopra hivatkozik, és az "**FK_RecipesIncludeIngredients_recipes_ingredientId**" nevet kapta, míg a másik a "**recipeId**" oszlopra hivatkozik, és az

"FK_RecipesIncludeIngredients_recipes_recipeld" nevet kapta. Ezek a kulcsok biztosítják, hogy csak azon receptek és összetevők azonosítói szerepeljenek az adattáblában, amelyek a "recipes" táblában szerepelnek.

Az adattáblának van két **INDEX**-e, mindkettő az adattáblában szereplő azonosító oszlopra (az "ingredientId" és "recipeld" oszlopokra) vonatkozik.

2.10 A "tags" táblázat:

- **id:** A címke **egyedi azonosítója**, amely egy egész szám (INT) típusú és nem lehet üres (NOT NULL). Az oszlop **AUTO_INCREMENT** tulajdonsággal rendelkezik, ami azt jelenti, hogy az adatbázis automatikusan növeli az értékét minden új sor hozzáadásakor.
- **name:** A címke neve, hosszú szöveg (LONGTEXT) típusú és nem lehet üres (NOT NULL).
- **description:** A címke leírása, hosszú szöveg (LONGTEXT) típusú és nem lehet üres (NOT NULL).
- **recipeProperty:** Az érték, amely azt jelzi, hogy ez a címke egy adott recept tulajdonságához tartozik-e, egy egész szám (INT) típusú és lehet üres (NULL).
- **min:** Az érték alsó határa, amely akkor hasznos, ha a címke értéke egy szám, egy egész szám (INT) típusú és lehet üres (NULL).
- **max:** Az érték felső határa, amely akkor hasznos, ha a címke értéke egy szám, egy egész szám (INT) típusú és lehet üres (NULL).

2.11 A "userprofiles" táblázat:

- **id:** Az **egyedi azonosítója** a felhasználói profilnak, amely egy egész szám (INT) típusú és nem lehet üres (NOT NULL). Az oszlop **AUTO_INCREMENT** tulajdonsággal rendelkezik, ami azt jelenti, hogy az adatbázis automatikusan növeli az értékét minden új sor hozzáadásakor.
- **weight:** A felhasználó súlya, dupla pontos szám (DOUBLE) típusú és nem lehet üres (NOT NULL).
- **height:** A felhasználó magassága, dupla pontos szám (DOUBLE) típusú és nem lehet üres (NOT NULL).

- birthDate: A felhasználó születési dátuma, időpont (DATETIME) típusú és nem lehet üres (NOT NULL).
- gender: A felhasználó nemének kódja, egész szám (INT) típusú és nem lehet üres (NOT NULL).
- goalWeight: A felhasználó cél súlya, dupla pontos szám (DOUBLE) típusú és nem lehet üres (NOT NULL).
- hairIndex: A felhasználó hajának stílusának kódja, egész szám (INT) típusú és nem lehet üres (NOT NULL).
- skinIndex: A felhasználó bőr típusának kódja, egész szám (INT) típusú és nem lehet üres (NOT NULL).
- eyesIndex: A felhasználó szemének színének kódja, egész szám (INT) típusú és nem lehet üres (NOT NULL).
- mouthIndex: A felhasználó szájának stílusának kódja, egész szám (INT) típusú és nem lehet üres (NOT NULL).
- dietid: Az étrend azonosítója, amelyhez a felhasználó tartozik, egész szám (INT) típusú és lehet üres (DEFAULT NULL).
- authOfProfileId: A felhasználói profilt létrehozó felhasználó azonosítója, egész szám (INT) típusú és nem lehet üres (NOT NULL).

Az adattábla rendelkezik egy **UNIQUE INDEX**-szel az "**authOfProfileId**" oszlopra, amely biztosítja, hogy minden felhasználó csak egy profilja legyen a rendszerben.

Az adattáblának van egy **INDEX**-e a "**dietid**" oszlopon, amely gyorsabb adatelérést biztosít a keresési műveletek során.

Az adattáblának van két idegen kulcsa (**FOREIGN KEY**), amelyek mind a "**auths**" nevű táblára vonatkoznak, amelynek azonosító oszlopára (id) hivatkoznak. Az egyik idegen kulcs az "**authOfProfileId**" oszlopra hivatkozik, és az "**FK_userProfiles_auths_authOfProfileId**" nevet kapta, míg

2.12 Az "usersprefertags" táblázat:

- userId: Az a felhasználó azonosítója, aki preferált egy bizonyos címkét. Ez egy egész szám (INT) típusú és nem lehet üres (NOT NULL).

- **tagId**: Az a címke azonosítója, amelyet a felhasználó preferált. Ez egy egész szám (INT) típusú és nem lehet üres (NOT NULL).

Az oszlopok közötti elsődleges kulcs (**PRIMARY KEY**) az **"userId"** és **"tagId"** oszlopokra utal, ami azt jelenti, hogy mindkét oszlopnak együtt kell lennie a táblában, és mindkettő egyedi azonosító. A táblának van egy **"tagId"** oszlopra készített **indexe (IX_UsersPreferTags_tagId)**, amely javítja a teljesítményt, amikor címkék alapján keresünk a táblában.

Két idegen kulcsa (**FOREIGN KEY**) van, mindkettő a **"userprofiles"** táblára vonatkozik, amelynek azonosító oszlopára (id) hivatkozik. Az egyik idegen kulcs a **"tagId"** oszlopra hivatkozik, és az **"FK_UsersPreferTags_tags_tagId"** nevet kapta, míg a másik a **"userId"** oszlopra hivatkozik, és az **"FK_UsersPreferTags_userProfiles_userId"** nevet kapta. Ezek a kulcsok biztosítják, hogy csak azon felhasználók azonosítói és címkék azonosítói szerepeljenek az adattáblában, akik a **"userprofiles"** és a **"tags"** táblában szerepelnek.

3. Munkamegosztás

3.1. Feladatok elosztása

A Salus projektet három fős csapatban készítettük. A projekt ötlet megszületése után már az elején tisztáztuk, hogy kinek mi a feladata. Alapvetően az egyéni készségek alapján osztottuk ki a feladatköröket, így a következő kiosztás született: Varga Balázs Ádám ([VargaBalazsAdam](#)) feladata volt a backend fejlesztése, Iliás György Dávid ([iliasdawe03](#), [IliasDavid03](#)) készítette a webalkalmazás frontendjét és Wittman Bence Tamás ([WittmanBence](#)) a mobil alkalmazást.

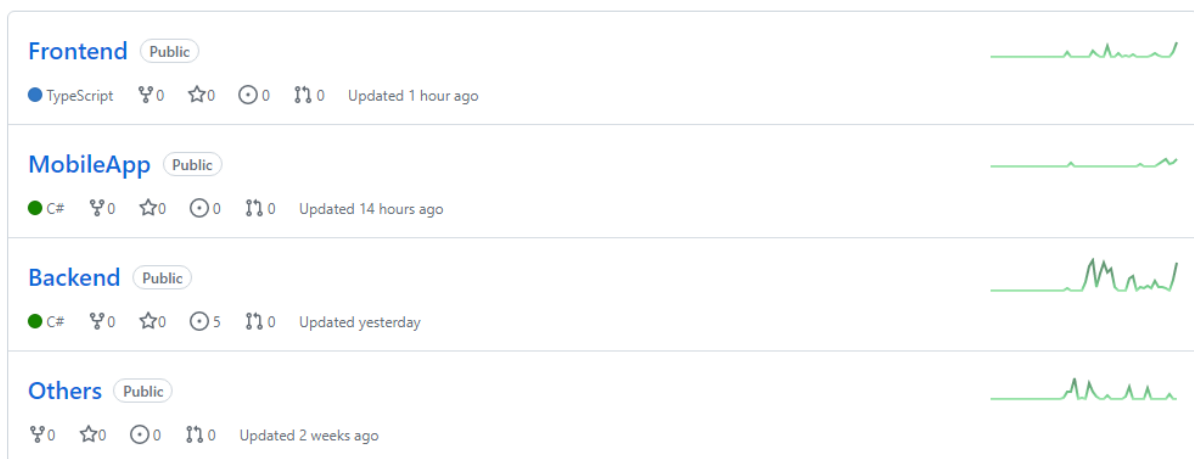
3.2. Verziókezelés

A projekt során gyakran párhuzamosan dolgoztunk, és az alkalmazás építésének folyamatát szeretnénk volna követni, ezért a GitHub nevű git verziókezelő

szolgáltatást használtuk. A Git egy nyílt forráskódú, elosztott verziókezelő rendszer, amely lehetővé teszi a fejlesztők számára a kód megosztását és verziókezelést. A GitHub egy webes Git tárhelyszolgáltatás, amelynek segítségével a fejlesztők könnyen megoszthatják és együttműködhetnek azonosított projekteken.

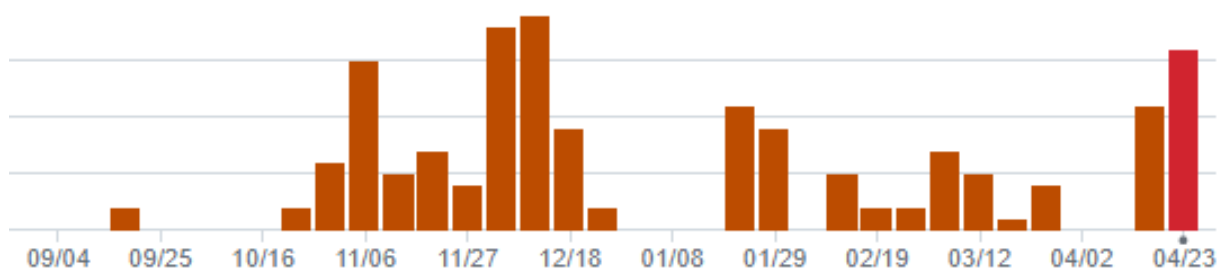
3.3. A GitHub használata

A Salus projekt fejlesztése négy GitHub repositoryban történt, amely lehetővé tette számunkra, hogy bárhol, bármilyen eszközről hozzáférve dolgozzunk az alkalmazáson.



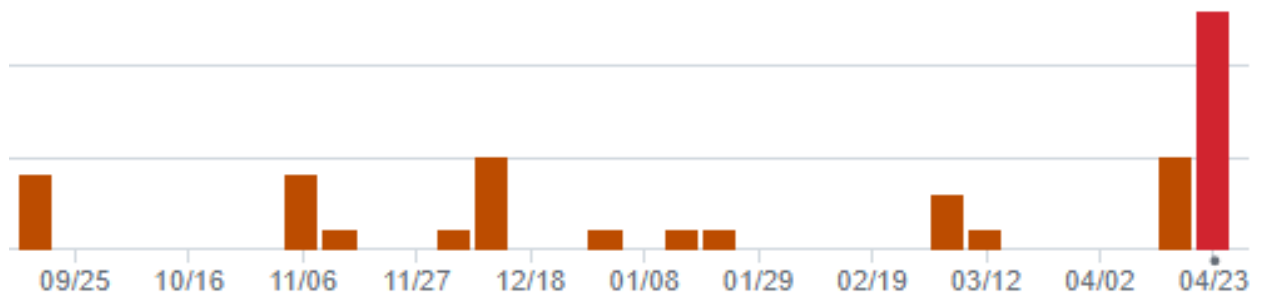
2. Ábra. (Repository-k)

3.4 Statisztikák

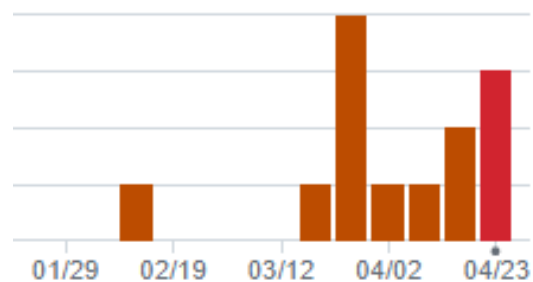


3. Ábra. (Backend commit-ok heti felbontásban 2022. szeptemberétől 2023. áprilisig)

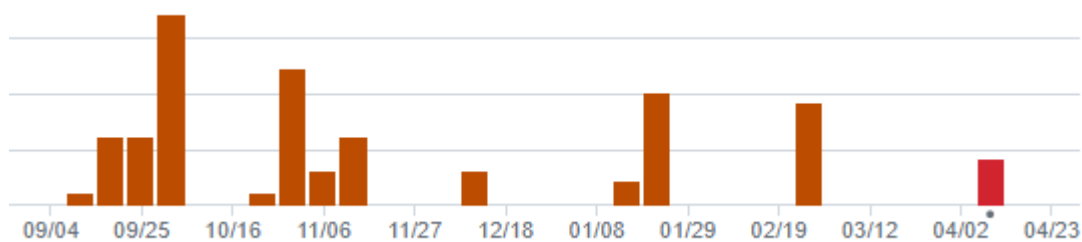
A frontendes fejlesztőnk havi sprintek alkalmazásával oldotta meg a programozás folyamatát.



4. Ábra. (Frontend commit-ok heti felbontásban 2022. szeptemberétől 2023. áprilisig)



5. Ábra. (Mobil commit-ok heti felbontásban 2023. januártól 2023. áprilisig)



6. Ábra. (Others commit-ok heti felbontásban 2022. szeptemberétől 2023. áprilisig)

4. Backend

4.1 Használt technológiák

A gyors haladás érdekében egy már a backendes által jól ismert, otthonos stack-et választottunk, a .Net Core 6.0-MySQL - t.

4.2 MySQL

A MySQL könnyen skálázható, ami azt jelenti, hogy az adatbázis méretétől és a felhasználók számától függően felfelé és lefelé skálázható. A MySQL további előnye a sebesség és a kompatibilitás. A MySQL nagyon gyors feldolgozási sebességgel rendelkezik, ami kiváló választás nagy forgalmú webhelyek és alkalmazások számára.

Előnyök:

- A MySQL tartalmazza az ACID (Atomikus, Konzisztens, Izolált és Tartósság) tranzakciókezelési tulajdonságokat, amelyek biztosítják az adatok konzisztenciáját és integritását.
- A MySQL támogatja az SQL lekérdezéseket, amelyek kényelmesek és intuitívak lehetnek azok számára, akik a relációs adatbázisokkal dolgoznak.
- A MySQL széles körben használják nagyobb vállalkozások és kormányzati szervezetek, mert megbízható és stabil adatbázis rendszert kínál.
- A MySQL több évtizedes fejlesztési tapasztalattal rendelkezik, és széles körben dokumentált. Több millió felhasználó és számos fejlesztői közösség áll rendelkezésre a problémák megoldásához és az új fejlesztések eléréséhez.

Hátrányok:

- NoSQL-hez képest bonyolult fel-, letöltés a nem JSON szerű formátum miatt.
- Szükséges előre felépíteni az adatbázis struktúráját.
- A MySQL adatbázis szigorúan meghatározott sémákkal dolgozik, ami korlátozza a rugalmasságot és a skálázhatóságot, különösen akkor, ha a sémák gyakran változnak.
- A MySQL adatbázisok nagyon jól teljesítenek kisebb és közepes méretű adathalmazok esetén. Azonban, ha az adathalmazok nagyon nagyok, és több millió dokumentumot kell kezelni, a NoSQL adatbázisok jobb teljesítményt nyújtanak, mivel könnyebb skálázni őket horizontálisan.

4.3 .NET

Az .NET egy Microsoft által fejlesztett keretrendszer, amely lehetővé teszi a fejlesztők számára a Windows operációs rendszerre szánt alkalmazások és szolgáltatások gyors és hatékony létrehozását és kezelését. Az alábbiakban összefoglaljuk az .NET előnyeit és hátrányait:

Előnyök:

- Gyors fejlesztés: Az .NET keretrendszer használata lehetővé teszi a fejlesztők számára a gyors és hatékony alkalmazásfejlesztést, mivel tartalmazza a szükséges osztályokat és könyvtárakat, amelyekre szükség lehet az alkalmazások kódolásához.
- Platformfüggetlenség: Az .NET platformfüggetlen, ami azt jelenti, hogy a kód azonosan fut a Windows, a Linux vagy a macOS rendszeren.
- Nagy biztonság: Az .NET keretrendszer beépített biztonsági funkciókat tartalmaz, amelyek lehetővé teszik a fejlesztők számára, hogy biztonságosabb alkalmazásokat hozzanak létre.
- Automatikus memóriakezelés: Az .NET automatikus memóriakezelést használ, amely megkönnyíti a fejlesztők számára a memóriakezelési problémák elkerülését.
- Nagy fejlesztői közösség: Az .NET keretrendszerhez egy nagy fejlesztői közösség tartozik, amely segítséget nyújt a fejlesztőknek, ha problémáik vannak.

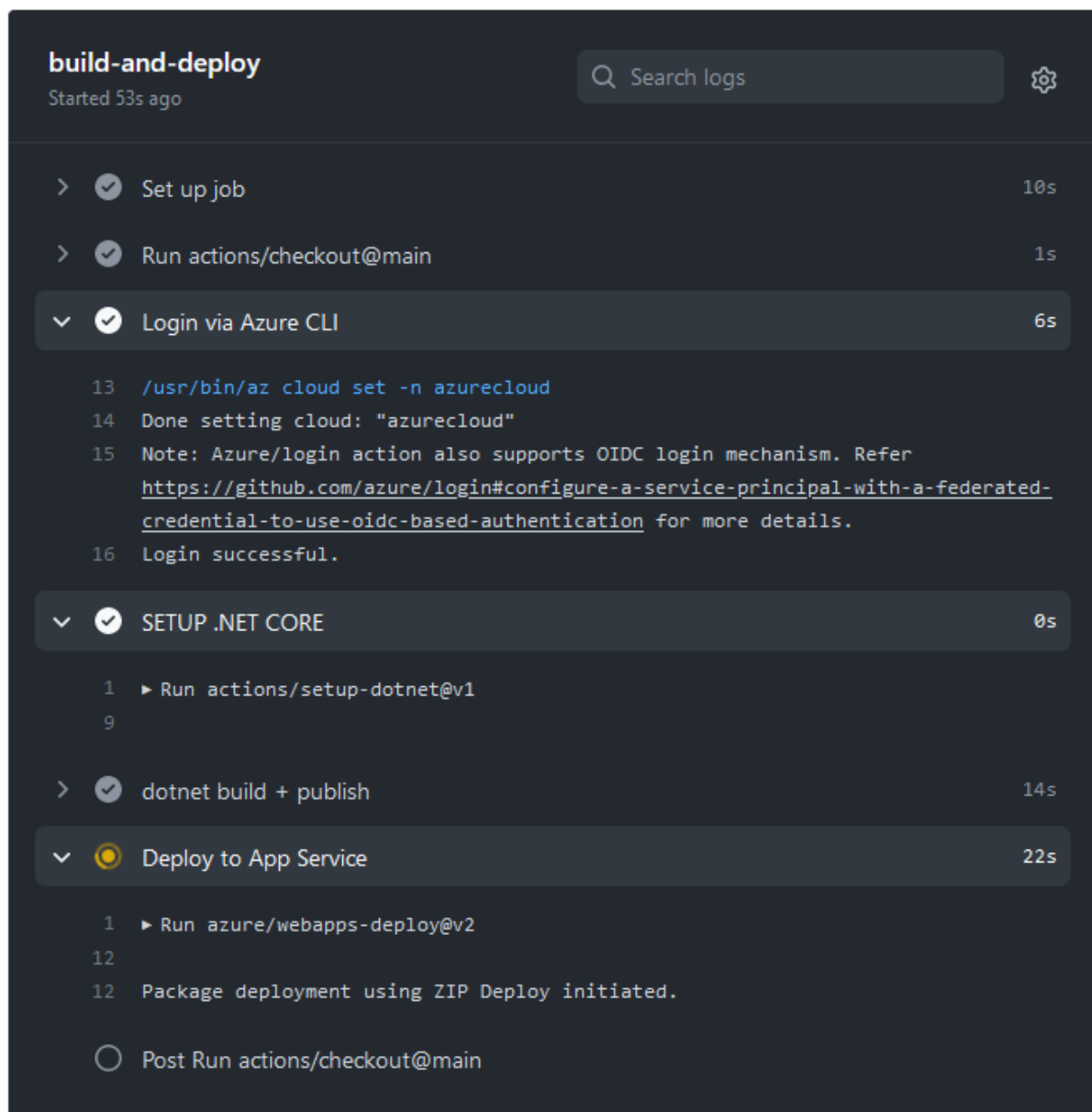
Hátrányok:

- Proprietari technológia: Az .NET egy Microsoft által fejlesztett és tulajdonolt keretrendszer, ami azt jelenti, hogy más cégek nem fejleszthetnek hozzájárulást, amely korlátozhatja a platform nyitottságát.
- Magasabb licencdíjak: Az .NET keretrendszer használata néhány licenc díjjal járhat, ami magas költségeket jelenthet a nagyobb cégek számára.
- Kisebb teljesítmény: Az .NET keretrendszer használata kisebb teljesítményt eredményezhet, mint az alacsony szintű nyelvek használata.

- Korlátozott multiplatform támogatás: Az .NET keretrendszernek bár van platformfüggetlen támogatása, de még mindig korlátozott a multiplatform támogatása.

4.4 Hosting

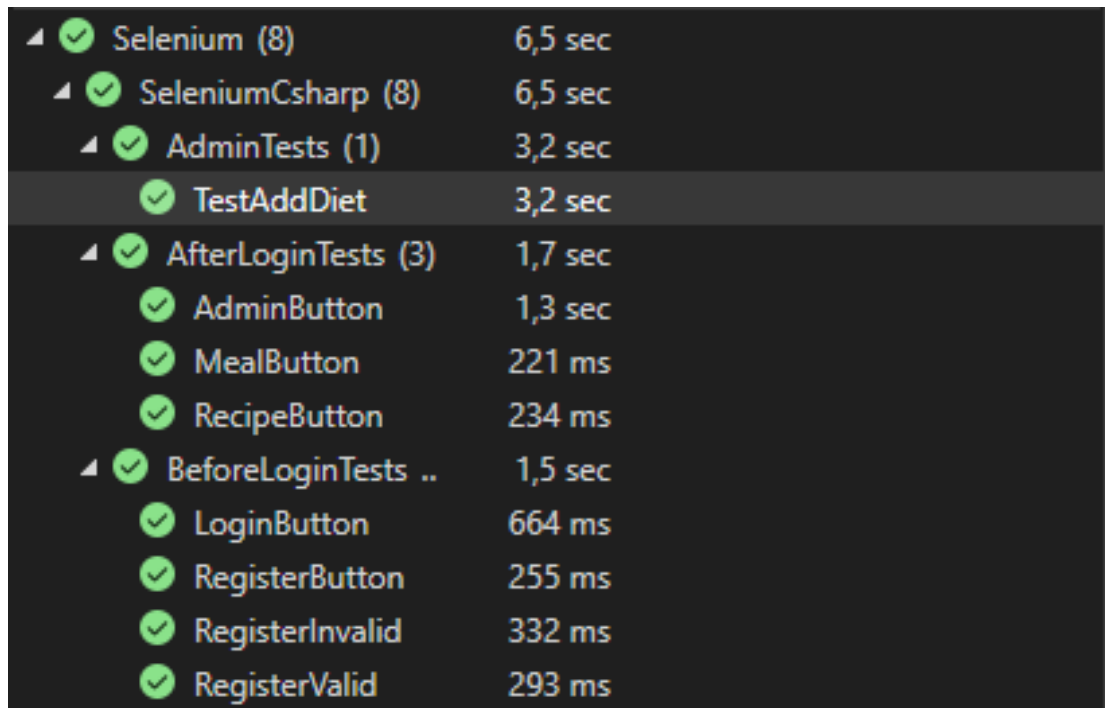
A hosting az azure students plan keretein belül Github-bal összekötve történik. Az backend automatikusan újra buildeli és publikálja magát a mainba történő merge-eket követően.



7. Ábra. (Backend host)

4.5 Tesztelés

A tesztelés C# nyelven írt Selenium tesztekkel történt, melyek képesek regisztrálni az éles, futó szerveren, bejelentkezni, ellenőrizni átirányítanak-e a kívánt oldalakra, de még admin fiókkal az admin funkciókhoz is hozzáférnek.



▲ ✓ Selenium (8)	6,5 sec
▲ ✓ SeleniumCsharp (8)	6,5 sec
▲ ✓ AdminTests (1)	3,2 sec
✓ TestAddDiet	3,2 sec
▲ ✓ AfterLoginTests (3)	1,7 sec
✓ AdminButton	1,3 sec
✓ MealButton	221 ms
✓ RecipeButton	234 ms
▲ ✓ BeforeLoginTests ..	1,5 sec
✓ LoginButton	664 ms
✓ RegisterButton	255 ms
✓ RegisterInvalid	332 ms
✓ RegisterValid	293 ms

8. Ábra. (Selenium tesztek)

5. Frontend

5.1 Használt Technológia

A választás során az Angulárra esett a választás nagy fejlesztői közössége illetve az előtte már jól ismert TypeScript miatt.

5.2 Angular

Az Angular egy TypeScript alapú, nyílt forráskódú keretrendszer, amelyet a Google fejlesztett ki a modern webes alkalmazások fejlesztéséhez. Az Angular nagyon sok hasznos funkciót és szolgáltatást kínál, amelyekkel egyszerűbbé és

hatékonyabbá tehetjük a frontend fejlesztést. Az alábbiakban összefoglalom az előnyeit hátrányait:

Előnyök:

- TypeScript alapú: Az Angular keretrendszer TypeScript nyelvet használ, amely hatékonyabb és strukturáltabb kód írását teszi lehetővé, és javítja a fejlesztői termelékenységet.
- Modularitás: Az Angular keretrendszer moduláris felépítése lehetővé teszi az alkalmazás komponenseinek és funkcióinak elkülönítését, és így egyszerűbbé teszi az alkalmazás karbantartását és bővítését.
- Adatkötés: Az Angular keretrendszer nagyon hatékony adatkötést tesz lehetővé a komponensek és a sablonok között, amely lehetővé teszi a felhasználói felület dinamikus frissítését.
- Nagy fejlesztői közösség: Az Angular keretrendszer mögött álló nagy fejlesztői közösség biztosítja, hogy a keretrendszer mindig naprakész és támogatott legyen.

Hátrányok:

- Magas tanulási küszöb: Az Angular keretrendszer viszonylag bonyolult és nagy tanulási görbével rendelkezik, ami megnehezítheti az új fejlesztők számára a kezdést.
- Nagy méretű: Az Angular keretrendszer nagyobb méretű, mint néhány más keretrendszer, és ennek következtében lassabb lehet az alkalmazás betöltése.
- Teljesítmény: Az Angular keretrendszer valamivel lassabb lehet, mint néhány más keretrendszer, különösen, ha nagy alkalmazásokat fejlesztünk vele.
- Angular verziók közötti kompatibilitás: Az Angular keretrendszer verziói közötti kompatibilitás néha problémákat okozhat az alkalmazások frissítésekor.

5.3 Rendszerkövetelmények

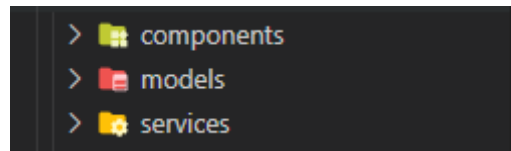
Többek között az alkalmazás futtatásához az alábbi rendszerkövetelmények szükségesek:

- Operációs rendszer: Windows, Mac vagy Linux
- Webböngésző: Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge
- Node.js: verzió 18 vagy újabb
- Angular CLI: verzió 15 vagy újabb

5.4 Alkalmazás szerkezete

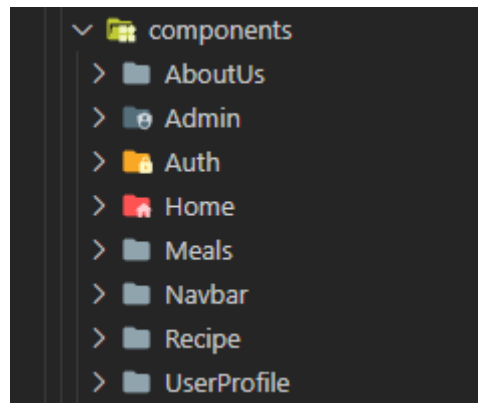
Az alkalmazás a következő főbb elemekből épül fel:

- Modulok: Az alkalmazás modulokra van bontva, amelyek az alkalmazás különböző funkcióit kezelik.



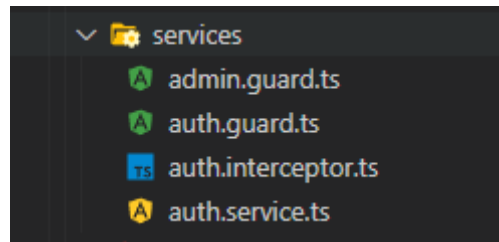
9. Ábra. (Frontend modulok)

- Komponensek: A modulokon belül találhatóak, és a felhasználói felület különböző részeit vezérlik.



10. Ábra. (Frontend komponensek)

- Szolgáltatások: A komponensek és modulok közötti kommunikációért felelős.



11. Ábra. (Frontend szolgáltatások)

6. Mobil

6.1 Használt Technológia

A választásunk az .NET MAUI keretrendszerre esett a mobil alkalmazás készítése során, mivel az iskolai tananyagunk szerves részét képezi a C# programozási nyelv és a .NET MAUI keretrendszer használata, így teljes mértékben átláttuk annak működését. Bár a választásunk előnyökkel járt, ugyanakkor nem voltak hiányosságai sem. Az .NET MAUI ugyanis egy nagyon új keretrendszer, amelynek még számos új funkciója fejlesztés alatt áll, és bizonyos funkciók esetében a dokumentáció sem áll rendelkezésre. Ez azt jelenti, hogy számos probléma merülhet fel, amelyeket nem könnyű megoldani, vagy amelyekre nincs elég információ a megoldáshoz. Ennek ellenére úgy döntöttünk, hogy felhasználjuk ezt a keretrendszert, és mindent megteszünk annak érdekében, hogy kihasználjuk előnyeit, miközben megoldjuk a felmerülő problémákat.

6.2 .NET MAUI

.NET MAUI (Multi-platform App UI) egy nyílt forráskódú, cross-platform keretrendszer, amely lehetővé teszi a fejlesztők számára, hogy hatékonyan fejlesszenek mobil, asztali és webes alkalmazásokat a .NET és C# nyelvek használatával. Az .NET MAUI keretrendszer nagyon hasonlít az elődjére, az Xamarin.Forms-ra, azonban kiterjeszti annak lehetőségeit és a felhasználói élményt.

A .NET MAUI egy magas szintű, de mégis könnyen alkalmazható felhasználói felületi keretrendszer, amely a .NET által kínált összes előnyt biztosítja, mint például a hatékony memóriakezelés, a modern és biztonságos kódolás, az erős típusosság, az egyidejűség és a hibakeresési eszközök. Az alkalmazások készítésekor az .NET MAUI felhasználók számára lehetőséget biztosít az egyszerű, platformfüggetlen megjelenési elemek használatára, amelyeket könnyen testreszabhatnak, valamint az egyéni vezérlők és sablonok készítésére.

A .NET MAUI keretrendszer támogatja a legnépszerűbb platformokat, mint az Android, az iOS, a macOS, a Windows és a Linux, és lehetővé teszi, hogy a fejlesztők egyetlen kódbázison dolgozhassanak, és mindenhol azonos felhasználói élményt nyújtsanak. Az .NET MAUI célja a hatékonyság és a termelékenység növelése, valamint az egyszerűbb, kezelhetőbb kódolás és a könnyebb karbantarthatóság.

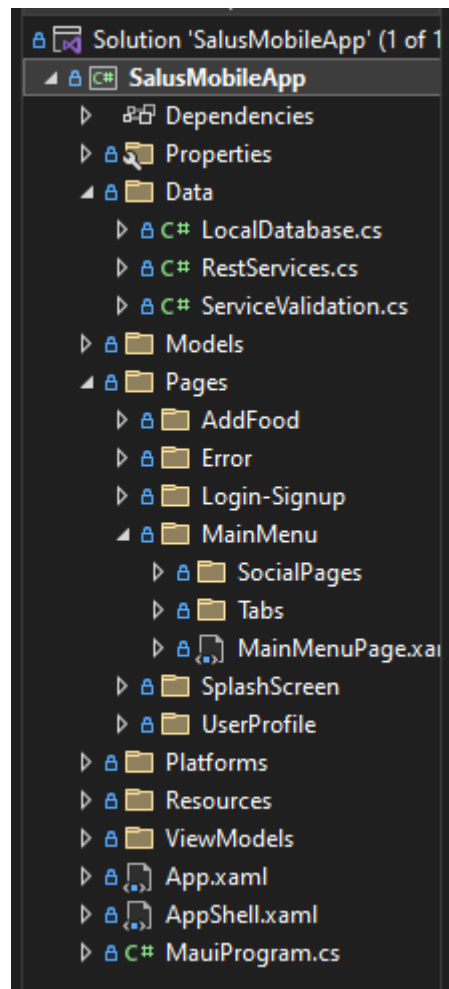
6.3 Alkalmazás szerkezete

Az mvvm minta, rövidítve a Model-View-ViewModel minta, az egyik legelterjedtebb architektúráis minta a .NET alkalmazásfejlesztés területén. Az mvvm minta használata során az alkalmazás felépítése átlátható és könnyen karbantartható lesz, hiszen az alkalmazás különböző rétegei elkülönülnek egymástól. A model réteg felelős az alkalmazásban használt adatok és üzleti logika tárolásáért, míg a view réteg a felhasználói felület megjelenítéséért és interakciójáért felelős. A viewmodel réteg azonban összeköti a két réteget, és feladata a felhasználói felület és az adatmodell közötti kommunikáció végzése.

Az mvvm minta alkalmazása lehetővé teszi az alkalmazásunk különböző részeinek egymástól független fejlesztését, amely nagyban növeli a fejlesztési folyamat hatékonyságát és rugalmasságát. Az mvvm minta segítségével a kód modularizációja javul, és így az egyes rétegek könnyen karbantarthatóak és tesztelhetőek lesznek. Az mvvm minta alkalmazása segít a kód olvashatóságának javításában is, mivel az egyes rétegek szerepei és felelősségi körük világosan el vannak választva.

A .NET MAUI keretrendszer alkalmazása során is érdemes az mvvm mintát alkalmazni, hiszen az lehetővé teszi a kód újrafelhasználását, az átláthatóbb kódstruktúrát és a könnyebb karbantarthatóságot. A .NET MAUI keretrendszer az MVVM minta támogatását is beépítette, amely biztosítja, hogy az egyes rétegek egymással jól kommunikáljanak, és az alkalmazásunk jól strukturált és könnyen bővíthető legyen. Az mvvm minta alkalmazása tehát egy hatékony és követendő megközelítés a .NET alkalmazásfejlesztés területén.

Mobilalkalmazásunk fejlesztése során az egyik legfontosabb szempont az volt, hogy a lehető legmagasabb minőségű, rugalmas és hatékony alkalmazást készítsük. Ennek érdekében a fejlesztői csapatunk úgy tervezte meg az alkalmazás szerkezetét, hogy az megfeleljen az mvvm minta követelményeinek.



12. Ábra. (Mobil könyvtárhierarchia)

III. Felhasználói dokumentáció

7. Alap információk

7.1 Az alkalmazás áttekintése

Az alkalmazás főbb célja a receptekben való kalória, zsír, szénhidrát, fehérje megismertetése, az oldal olyan embereknek szól akik szeretnék ezt követni, illetve számon tartani. Az alkalmazás hasznos lehet azoknak akik fogyni, vagy épp hogy tömegelni szeretnének.

7.2 Telepítési útmutató

Az alkalmazás weboldalának elérése semmilyen telepítést nem igényel a felhasználó részéről. A telefonra elérhető alkalmazás az alábbi címről szabadon letölthető bárki számára:

<https://github.com/14A-C-Salus/Others>,

<https://salus-healthy-lifestyle.netlify.app>

A mobilalkalmazás kezelésének leírása a 8. pontban található.

7.3 Üzemeltetési útmutató

Forráskód telepítése git verziókezelő segítségével az alábbi címről:

<https://github.com/14A-C-Salus/Frontend>.

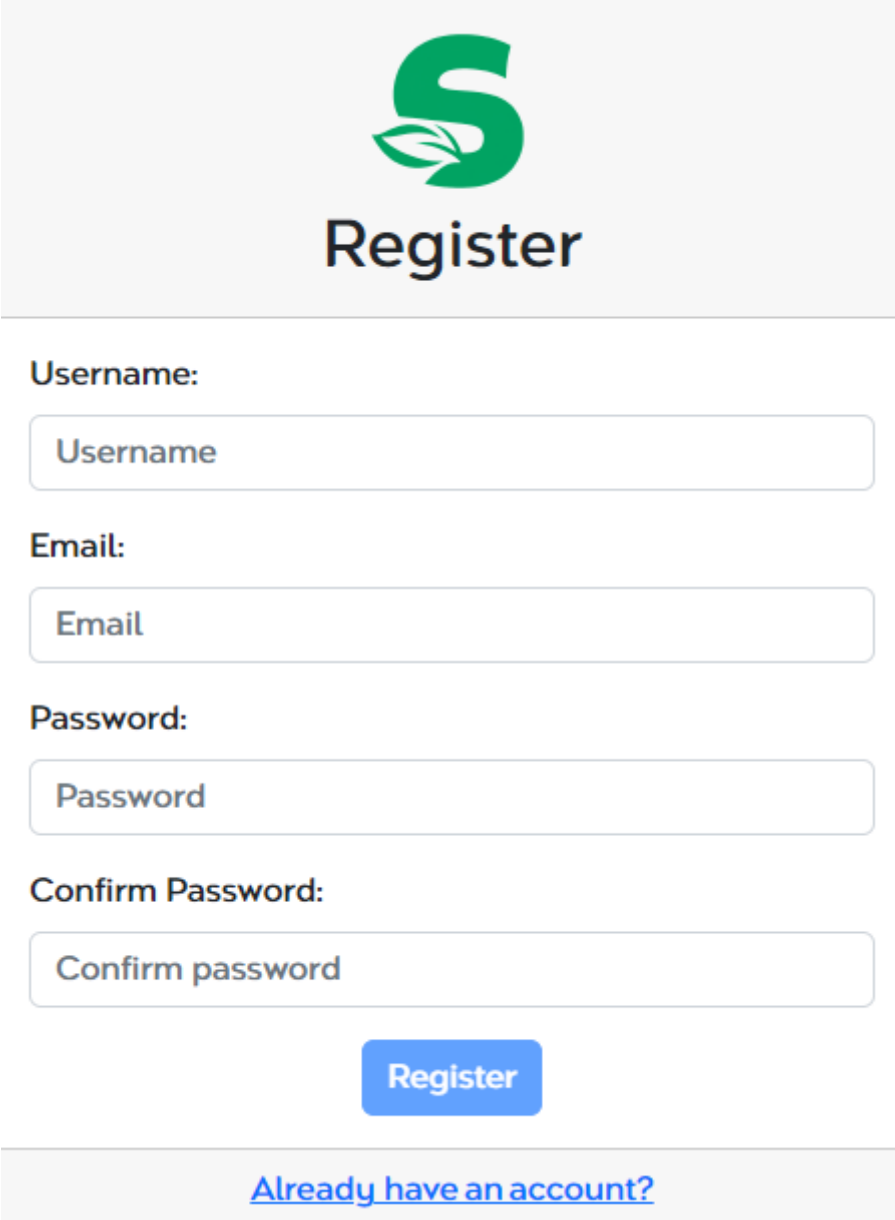
Az alkalmazás futtatásához szükséges dolgokat az 5.3 Rendszerkövetelményekben már kifejtettem.

Az alkalmazás letöltése után az “ npm i “ parancsot kell kiadni ami letölti a szükséges állományokat a futtatáshoz. Amint végzett az “ npm i “ parancs már csak az “ ng serve “ parancsot kell kiadni ami egyből elindítja az oldalt. Az oldal ezután megtalálható utána a <http://localhost:4200> címen.

7.4 Bejelentkezési folyamat

7.4.1 Regisztráció

Az oldalra mielőtt belépünk regisztrálni kell a regisztrációhoz meg kell adnunk először a felhasználó nevünket ami majd lesz a nevünk az oldalon, az email címet, a jelszavunkat és a jelszó megerősítést.



The registration form is displayed within a light gray container. At the top, there is a green logo consisting of a stylized 'S' with a leaf-like shape at its base. Below the logo, the word 'Register' is written in a large, bold, black font. The form contains four input fields, each preceded by a label: 'Username:', 'Email:', 'Password:', and 'Confirm Password:'. The input fields are white with rounded corners and a light gray border. Below the 'Confirm Password' field is a blue button with the word 'Register' in white. At the bottom of the form, there is a link that reads 'Already have an account?' in blue text.

Register

Username:

Username

Email:

Email

Password:

Password

Confirm Password:

Confirm password

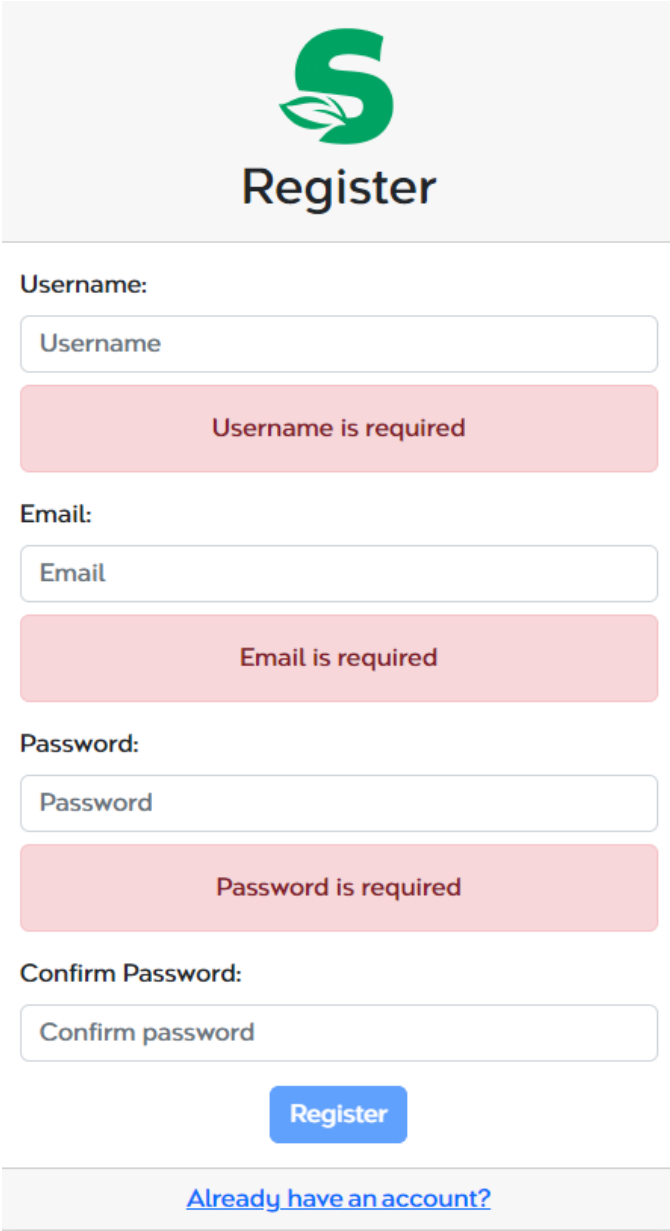
[Register](#)

[Already have an account?](#)

13. Ábra (Regisztráció)

7.4.2 Hiba esetén

Viszont hogy ha az adatokat nem adjuk meg akkor error-t kapunk



The image shows a web form titled "Register" with a green logo featuring a stylized 'S' and a leaf. The form contains four input fields: "Username:", "Email:", "Password:", and "Confirm Password:". Each input field has a corresponding red error message box below it stating "Username is required", "Email is required", "Password is required", and "Confirm password" (though the text in the image is "Confirm password" for the last one). A blue "Register" button is located below the "Confirm Password" field. At the bottom of the form, there is a link that says "Already have an account?".

Register

Username:

Username

Username is required

Email:

Email

Email is required

Password:

Password

Password is required

Confirm Password:

Confirm password

Register

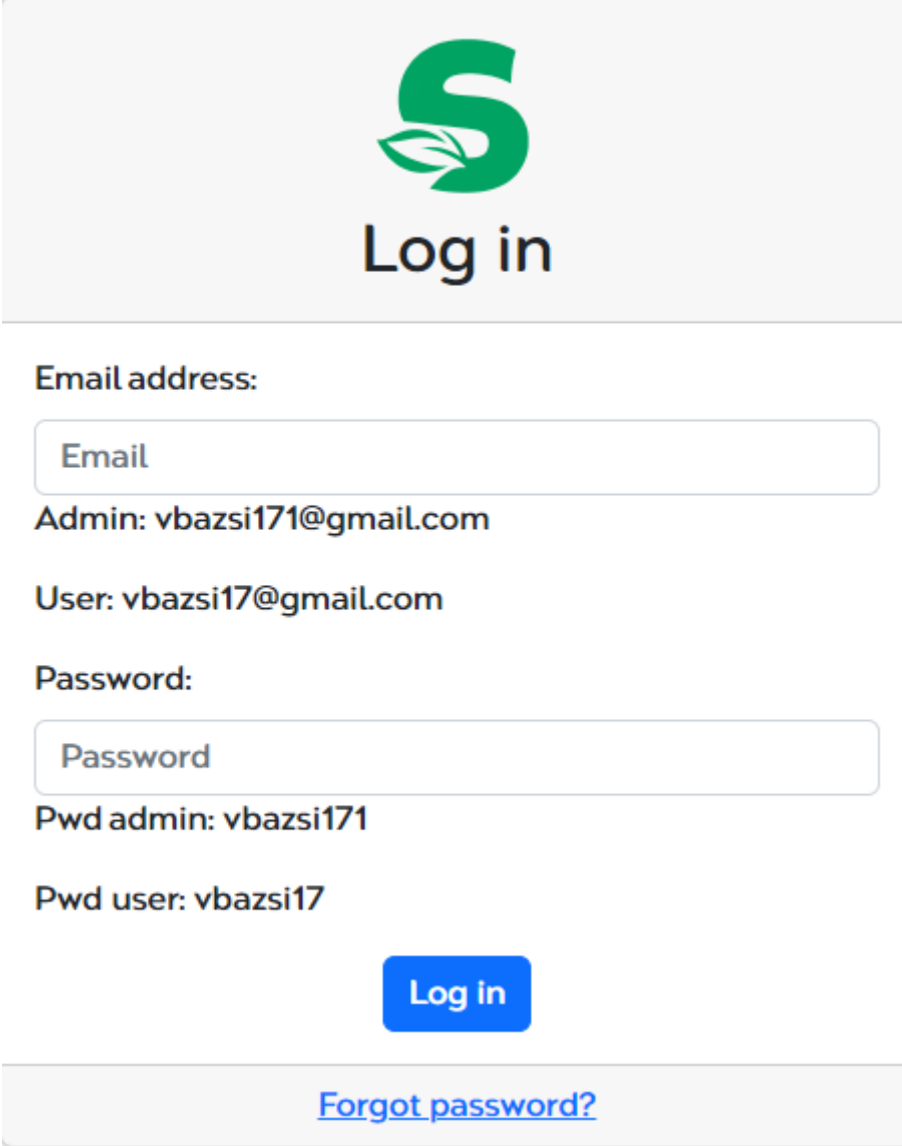
[Already have an account?](#)

14. Ábra (Hiba üzenetek)

És a Register gomb egészen addig nem lesz elérhető amíg az adatokat ki nem töltöttük.

7.4.3 Bejelentkezés

Miután regisztráltunk áttérhetünk a bejelentkezés fülre ahol is meg kell adnunk az email címet és jelszavat,



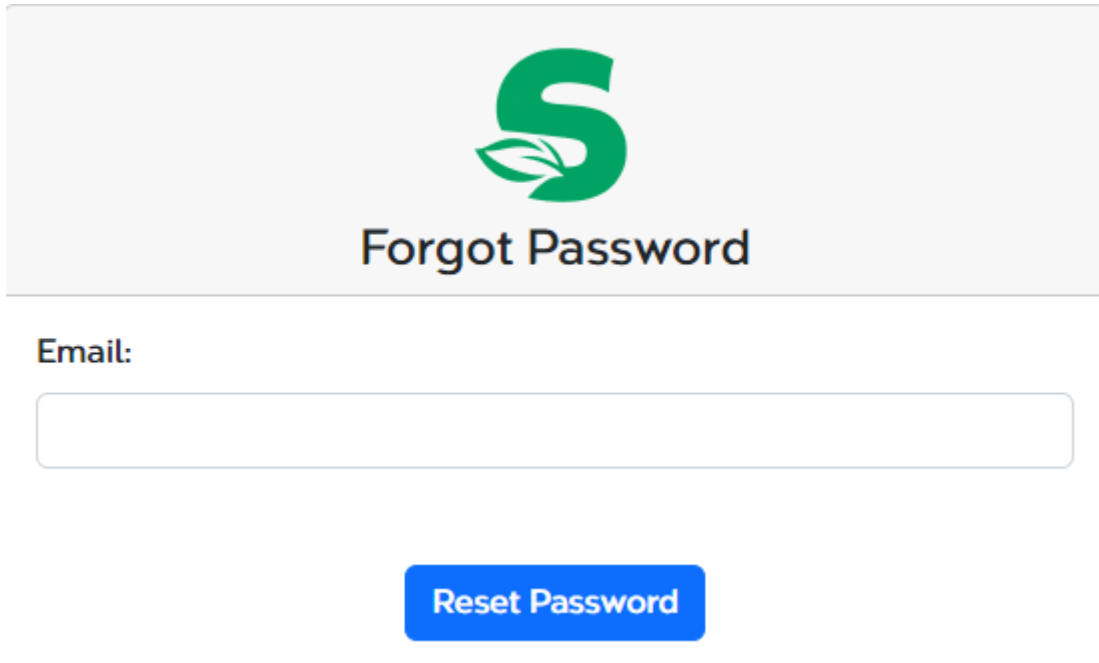
The login form features a light gray background. At the top center is a green logo consisting of a stylized 'S' with a leaf-like shape integrated into its lower left. Below the logo, the text 'Log in' is displayed in a large, black, sans-serif font. Underneath this, the label 'Email address:' is followed by a white input field with a light gray border and the placeholder text 'Email'. Below the input field, the text 'Admin: vbazsi171@gmail.com' is shown. This is followed by 'User: vbazsi17@gmail.com'. Then, the label 'Password:' is followed by another white input field with a light gray border and the placeholder text 'Password'. Below this field, the text 'Pwd admin: vbazsi171' and 'Pwd user: vbazsi17' are listed. A blue rectangular button with rounded corners and the white text 'Log in' is positioned below the password fields. At the bottom of the form, a light gray horizontal bar contains the text '[Forgot password?](#)' in blue.


15. Ábra (Bejelentkezés)

Egy admin és egy sima user fel is van sorolva annak érdekében ha ki szeretnénk próbálni az oldalt.

7.4.4 Elfelejtett jelszó

Ha elfelejtettük a jelszót, nincs gond, hisz ha az email címet tudjuk akkor lehet új jelszót csinálni.





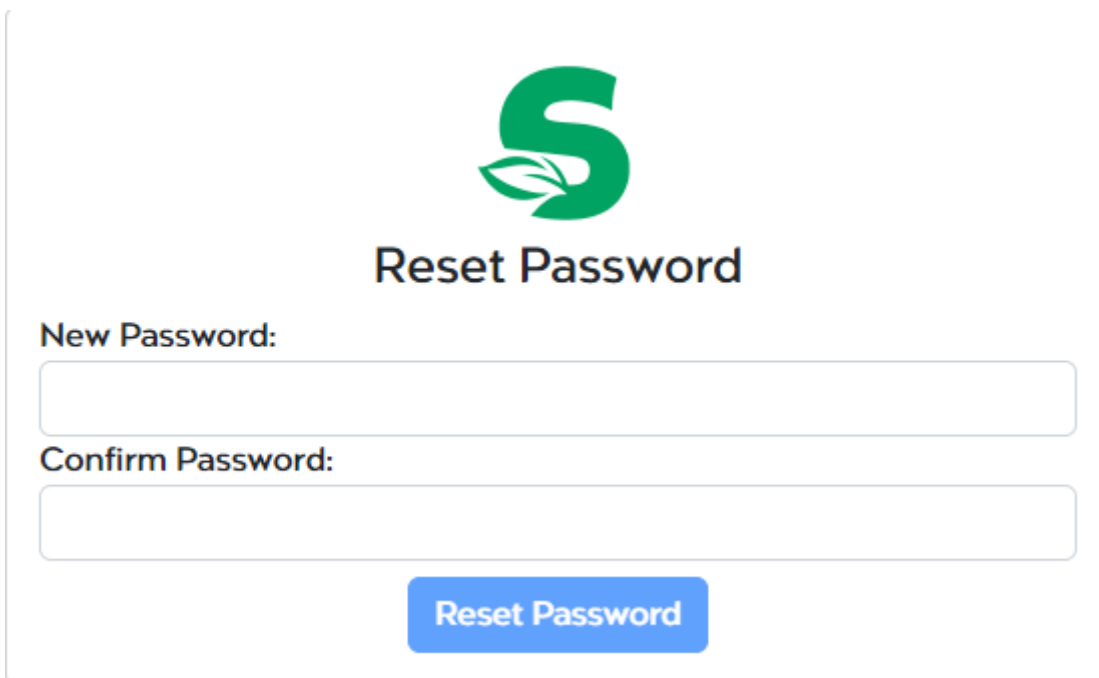
Forgot Password


Email:

Reset Password

16. Ábra (Elfelejtett jelszó)

Miután rákattintottunk a “Reset Password” gombra, egy emailt fogunk kapni. Az emailben lesz egy gomb amire ha rákattintunk a ResetPassword oldalra jutunk ahol is új jelszót tudunk csinálni.





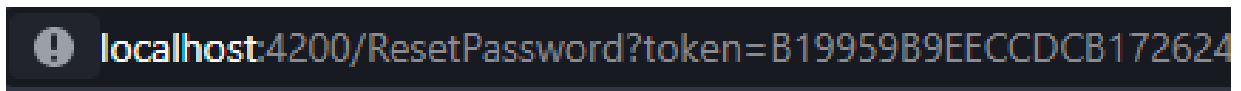
Reset Password

New Password:

Confirm Password:

Reset Password

17. Ábra. (Jelszó visszaállítás)



18. Ábra (Visszatért token)

8. Mobilalkalmazás

8.1 Az alkalmazás áttekintése

Azért döntöttünk úgy, hogy készítünk egy mobil alkalmazást a webes alkalmazásunk mellé, mert úgy véljük, hogy a mobilalkalmazások használata sokkal kényelmesebb és könnyebb, mint a webes alkalmazásoké. Emellett olyan hasznos funkciókkal is bővíthettük az alkalmazást amelyek a webes felületen megvalósíthatatlanok, mint például az alkalmazás korlátozott használata internet kapcsolat nélkül, illetve új receptek létrehozása vonalkód szkennelés segítségével.

8.2 Telepítési útmutató

A telefonra elérhető alkalmazás az alábbi címről szabadon letölthető bárki számára:

<https://github.com/14A-C-Salus/Others>

A mobilalkalmazásnak nincsenek kiemelkedő hardver igényei.

8.3 Üzemeltetési útmutató

Forráskód telepítése git verziókezelő segítségével az alábbi címről:

<https://github.com/14A-C-Salus/MobileApp>.

Amennyiben a fenti címről telepíti az alkalmazást, akkor a futtatásához szükséges a Visual Studio 2022 legfrissebb verziója, melyben egyéni igények alapján futtatható egy számítógépen illetve a számítógépre csatlakoztatott android telefonon. Ehhez szükséges engedélyezni a fejlesztői beállításokat, illetve az USB hibakeresést.

Irodalmi jegyzék:

Felhasználva	Elérhetősége
Backend	Patrick God - YouTube
Backend	ASP.NET Core Documentation — ASP.NET documentation
Mobilapp	NET Multi-platform App UI (.NET MAUI) - GitHub
Mobilapp	https://learn.microsoft.com/en-us/dotnet/maui/?view=net-maui-6.0
Frontend	Introduction to the Angular docs
Front-, Backend és Mobilapp	MySQL Documentation
Front-, Backend és Mobilapp	Introducing ChatGPT - OpenAI
Front-, Backend és Mobilapp	https://stackoverflow.com/