Soproni SZC Vas- és Villamosipari Technikum

54 213 05 Szoftverfejlesztő szakképesítés

2023/2024 tanév

13.A osztály

FitnessForge  
Záró dolgozat

Horváth Olivér

2024

[1. Bevezetés 3](#_Toc164938978)

[2. Fejlesztői dokumentáció 4](#_Toc164938979)

[2.1 Fejlesztőkörnyezet 4](#_Toc164938980)

[2.2 Adatszerkezet 6](#_Toc164938981)

[2.2.1 Az egész adatbázis 6](#_Toc164938982)

[2.2.2 Étkezéssel kapcsolatos táblák 7](#_Toc164938983)

[2.2.3 Edzéssel kapcsolatos táblák 8](#_Toc164938984)

[2.2.4 Fontosabb táblák 9](#_Toc164938985)

[2.3 Program sajátosságai 12](#_Toc164938986)

[2.3.1 UserService 12](#_Toc164938987)

[2.3.2 UnitConverter 13](#_Toc164938988)

[2.3.3 Controller-ek 14](#_Toc164938989)

[2.3.4 Fontosabb View-k 20](#_Toc164938990)

[2.4 Tesztelés 25](#_Toc164938991)

[2.5 Fejlesztési lehetőségek 25](#_Toc164938992)

[3. Felhasználói dokumentáció 26](#_Toc164938993)

[3.1 Program célja és funkciója 26](#_Toc164938994)

[3.2 Szükséges hardvereszközök 26](#_Toc164938995)

[3.3 Telepítés és indítás 27](#_Toc164938996)

[3.3.1 Telepítési lépések 27](#_Toc164938997)

[3.3.2 Program indítása 27](#_Toc164938998)

[3.4 Részletes bemutatás 27](#_Toc164938999)

[3.5 Hibajelzések 27](#_Toc164939000)

[3.6 Információkérési lehetőségek 27](#_Toc164939001)

[4. Összefoglalás, köszönetnyilvánítás 27](#_Toc164939002)

[5. Irodalomjegyzék 27](#_Toc164939003)

# 1. Bevezetés

A FitnessForge nevű programom egy táplálkozás felügyelő és segítő program, továbbá egy edzések rögzítésére és áttekintésére készült program. A program azért készült el mert én gyakran szoktam ilyen alkalmazásokat használni, de mindig is hiányoltam a kevés információt és tápanyaginformációt az étkezésemből ezért egy olyan programot próbáltam meg létrehozni, ami ezt pótolja. A programban diagrammok és részletes táblázatok segítik a felhasználót. Továbbá azt is hiányoltam a programokban, amiket a múltban használtam, hogy nem volt megfelelő módja nagyobb ételek kényelmes felvételének a napi étkezések közé, mindig minden terméket külön ki kellett keresni és felvenni a naphoz. A programomban lehetőség van akármennyi étel létrehozására, így könnyen létre lehet hozni azt az ételt, amit a felhasználó nap mint nap szokott enni és így elég csak egyszer kikeresni és felvenni a naphoz, ezzel időt és energiát spórolva. Az edzés része a programnak pedig csak szimplán egy kiegészítése az táplálkozás részének a programnak, témába illett és úgy gondoltam ezt is érdemes beépíteni az alkalmazásomba. Nem szerettem, hogy mindig egy másik alkalmazás használatával tudtam csak edzéseket keresni, amiket az nap szerettem volna csinálni, illetve a programban a saját edzés létrehozása is egy egyedi funkció amivel jobban testre szabhatóbb egy edzés.

# 2. Fejlesztői dokumentáció

## 2.1 Fejlesztőkörnyezet

Visual Studio 2022

A Visual Studio 2022 a fejlesztők egyik legkedveltebb eszköze világszerte. Számos funkciót kínál, mint például az intelligens kódkiegészítés vagy a könnyen kezelhető projekt- és fájlkezelés. Támogatja a különféle nyelveket, például a C#, C++, JavaScript vagy Python. Emellett lehetőséget biztosít a platformok közötti fejlesztésre, mint például a Windows, Android vagy iOS.

C# és ASP.NET Core MVC

A C# és az ASP.NET Core MVC együtt egy erős és rugalmas fejlesztői környezetet alkotnak webalkalmazásokhoz. A C# lehetővé teszi a hatékony és strukturált kódolást, míg az ASP.NET Core MVC az MVC tervezési minta alkalmazásával segít a könnyebb alkalmazásfejlesztésben. Ez az alapja az alkalmazásomnak, a C#-ot és az MVC struktúrát az átláthatóság és a könnyű kezelhetőség miatt választottam.

Razor

Az ASP.NET Core Razor egy olyan sablonmotor, amely segíti a dinamikus weboldalak és webalkalmazások létrehozását. Könnyen kombinálható a C# és HTML, így könnyen generálható és megjeleníthető dinamikus tartalom. A Razor használatával az egyszerűség és hatékonyság mellett lehetőség van a személyre szabott kódolásra is. A Razor használatával az egyszerűség és hatékonyság mellett lehetőség van a személyre szabott kódolásra is. Nekem kényelmesebb volt ezt a megjelenítést válsztani mint bármi más frontend keretrendszer.

CSS és Bootstrap

A CSS stíluslapnyelv lehetővé teszi az HTML elemek megjelenésének testre szabását, míg a Bootstrap egy előre elkészített front-end keretrendszer, amely segít gyorsan és könnyen fejleszteni reszponzív weboldalakat. A programom nagyrészt Bootstrap-en alapul de tartalmaz egész sok CSS formázást is.

JavaScript és jQuery

A JavaScript segítségével adhatsz hozzá interaktivitást és dinamikusságot az oldalakhoz. Különféle funkciókat valósíthatsz meg, mint például űrlapellenőrzés vagy animációk. Dinamikus elemeknél és AJAX-al való dinamikus keresésnél használtam. A jQuery egy könnyűsúlyú JavaScript könyvtár, amely lehetővé teszi az HTML dokumentumok manipulálását és különféle interaktív elemek hozzáadását.

Chart.js

A Chart.js egy ingyenes JavaScript könyvtár, amely segít könnyen és rugalmasan létrehozni különféle típusú diagramokat és grafikonokat weboldalakon. 2 kördiagram-nál használtam könnyű kezelhetősége miatt.

Python

A Python sokoldalú programozási nyelv, amely széles körben használható különféle területeken. Könnyen tanulható és olvasható szintaxisa miatt ideális választás mind kezdők, mind tapasztalt fejlesztők számára. A projectemben egy külön alkalmazást hoztam létre ami Python segítségével küldd API kéréseket amiknek az eredményét az adatbàzisomba küldi.

MySQL

A MySQL egy megbízható relációs adatbázis-kezelő rendszer, amelyet sokan alkalmaznak a szoftverfejlesztés és a webalkalmazások háttértárolójaként. Összességében egy erős eszköz az adatok tárolásához és kezeléséhez.

## 2.2 Adatszerkezet

### 2.2.1 Az egész adatbázis

27 tábla

A képen szöveg, képernyőkép, Párhuzamos, diagram látható

Automatikusan generált leírás

9 tábla az étkezzéssel kapcsolatos

9 tábla az edzéssel kapcsolatos

7 az Identity által generált

### 2.2.2 Étkezéssel kapcsolatos táblák

A képen szöveg, képernyőkép, Betűtípus, diagram látható

Automatikusan generált leírás

### 2.2.3 Edzéssel kapcsolatos táblák

A képen szöveg, képernyőkép, Betűtípus, diagram látható

Automatikusan generált leírás

### 2.2.4 Fontosabb táblák

AspNetUsers

A képen szöveg, képernyőkép, szám, Betűtípus látható

Automatikusan generált leírásNapi kalóriához használt táblák:

Születési dátum

Nem

Súly

Magasság

Cél súly

Heti cél súly

AktivitásId 🡪Idegen kulcs az Aktivitási szint táblához

Az aktivitási szint tartalmazza, hogy a felhasználó egy héten

nagyjából mennyit mozog és ezáltal egy szorzót ad vissza a

napi kalória számolásához

NutirentId 🡪Idegen kulcs a Tápanyag táblához

A tápanyag tábla pedig különböző tápanyag százalék

eloszlásokat tartalmaz a 3 fő tápanyagból: Szénhidrát, Fehérje

és Zsír

Meal (Kapcsoló tábla)

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásÉtkezés típus megadja, hogy az 5 étkezés közül melyikhez tartozik

Az étet azonosító megadja, hogy melyik étel tartozik ide

És az Bevitel azonosító megadja, hogy melyik Felhasználó mai beviteléhez tartozik

Product

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásA termék összes fontos adatát tárolom

A Mértékegység azonosító megadja a termék mértékegységét

A ProductStatus oszlop eltárolja, hogy a termék jóvá van e hagyva

vagy nem

Food

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásA Mértékegység azonosító az étel mértékegységét tárolja

Az étel és a termék táblát egy kapcsolótábla köti össze

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásExercise

A TypeId megadja, hogy az adott gyakorlat melyik gyakorlat típushoz tartozik

Instructions a feladat elvégzését tartalmazza

Workout\_Has\_Exercise

A képen szöveg, Betűtípus, képernyőkép, szám látható

Automatikusan generált leírásEz a kapcsolótábla kapcsolja össze az Exercise és a Workout

táblákat.

A SetCount azt jelzi, hogy az adott feladatból hány kört kell

elvégezni

A RepetitionCount az elvégzendő darabszámot tartalmazza

Workout

A képen szöveg, Betűtípus, szoftver, képernyőkép látható

Automatikusan generált leírásA CreatorUserId jelzi, hogy az adott edzést melyik felhasználó hozta

létre, ezt az értéket szűrésnél használom.

## 2.3 Program sajátosságai

### 2.3.1 UserService

UserDailyCalorie() A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Automatikusan generált leírás

Ez a függvény az ún. Harris Benedict egyenlet alapján számolja ki egy felhasználó napi kalória bevitelét. Ennek a képletnek a legfrissebb verzióját használom.

A férfiaknál így néz ki:

(10 × súly kg-ban) + (6.25 × magasság cm-ben) – (5 × életkor évben) + 5

A nőknél pedig így:

(10 × súly kg-ban) + (6.25 × magasság cm-ben) – (5 × életkor évben) – 161

Ezeket a képleteket a függvény még megszorozza a felhasználó aktivitási szintjének a szorzójával majd hozzáadja a felhasználó heti súly változás célját szorozva ezerrel hogy így a megfelelő kalória mennyiséget adja hozzá vagy vegye el a napi kalória bevitelhez.

UserAge()

Kiszámolja, hogy a felhasználó mennyi idős a születési dátuma alapján.

UserDailyNutrients()

Kikeresi és visszaadja a felhasználó által kiválasztott tápanyag mennyiségek alapján, hogy hány gramm-ot lehet a különböző tápanyagokból fogyasztani.

A képlet, amit használok:

felhasználó napi kalóriája \* (tápanyag százalék / 100) / tápanyag kalória 1 gramm-ban

### 2.3.2 UnitConverter

A képen szöveg, képernyőkép, szoftver, Operációs rendszer látható

Automatikusan generált leírás

A tömeg mértékegységek úgy vannak kigyűjtve egy szótárba, hogy a mértékegyég neve és hogy hány gramm az adott mértékegység. Az űrmértékegységek pedig úgy, hogy a mértékegység neve és hogy az hány milliliter. Két függvénye van ennek az osztálynak Egy, ami a tömeg átváltására szolgál egy pedig ami az űrmérték átváltására szolgál. Mindegyik függvény paraméterül vár egy értéket, amit át kell váltani, a mértékegységet amiben az érték van és a cél mértékegység nevét. A függvények ezután megnézik, hogy léteznek-e a mértékegységek, ha igen akkor tovább megy különben hibát dob. Ezután megnézik, hogy a kívánt mértékegység az nagyobb e mint amiből át kell váltani, ez alapján kiderül, hogy felfele vagy lefele kell váltani.

### 2.3.3 Controller-ek

Ennek az 5 Controllernek a segítségével működik aA képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás progran. A HomeController egy darab Action-t tartalmaz, ez a főoldal. A ManagementController a felhasználók és a jogok kezelését végzi. A UserController az adott felhasználóhoz tartozó Action-ket tartalmazza. Itt van a felhasználó Home oldala, ami a mai étkezési adatait tartalmazza. Illetve a felhasználói adatok változtatása is ebben a Controllerben van. A MealController az összes olyan Action-t tartalmazza, ami az ételek kezeléséhez, és az oldalak használatához tartozik. A WorkoutController pedig az összes olyan Action-t, ami az edzések kezeléséhez tartozik.

#### 2.3.3.1 A MealController fontosabb kódjai:

[HttpGet] EditIntake(string mealType)

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

Az Action fontosabb része az, hogy a visszaküldésre kerülő model milyen adatokat fog tartalmazni. A model.userMeals az összes a felhasználóhoz tartozó mai étkezést tartalmazza. A userMealsFoodHasProducts az összes kapcsolótábla bejegyzést tartalmazza, ahol a felhasználó étkezéseinek ételei benne vannak. Az Action ezután átváltja a mennyiségeket a kapott listában milliliterbe és grammba, hogy a View-ban könnyen lehessen számolni ezekkel az adatokkal.

[HttpPost] Add(List<int> foods, string mealType)

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

Az Action kap egy listát, ami azokat az étel azonosítókat tartalmazza, amik már benne vannak az aktuális étkezéstípusban és egy étkezést típust amihez az ételeket hozzá lehet adni. Az allFoodHasProducts az összes olyan ételt tartalmazza, amiben csak Jóváhagyott termékek vannak. Ezután csoportosítom a listát foodId alapján.

[HttpPost] CreateFood(IFormCollection form)

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

Ez az Action egy IFormCollection-t kap paraméterül, aminek az adatai utána eltárolja változókba. Ezután az adatokból létrehoz egy Étel-t majd a View-ban kiválasztott termékeket (receiptProductIds) kikeresi az adatbázisból és hozzáadja a saját termék listájához. Ezután végigmegy az összes terméken és beállítja a FoodHasProduct kapcsolótábla bejegyzések mennyiség (Amount) értékét.

[HttpGet] SearchFood(string searchString, string mealType)

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

Ennek a keresés függvénynek a viewModel.userMeals listája alapjárat megkapja az összes olyan étkezést, ami aznap volt és ahhoz az étkezés típushoz tartozik. Ezt, ha keresési paraméter lett megadva leszűri arra a listára ahol az étel neve tartalmazza azt a szöveget. Ezután ugyan az az eljárás következik, mint korábban az EditIntake függvénynél. A keresés visszatérési értéke egy PartialView és a hozzátartozó viewModel.

#### 2.3.3.2 A WorkoutController fontosabb kódjai:

List()

A képen szöveg, Betűtípus, képernyőkép látható

Automatikusan generált leírás

Ez a függvény visszaadja azokat az edzéseket, amiket a jelenlegi felhasználó készített.

UpdateList(int selectedValue)

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

Ez a függvény pedig frissíti a listát annak megfelelően, hogy a View szűrési lehetőségei közül melyik lett kiválasztva. A felhasználó, vagy csak a saját maga által létrehozott edzéseket látja vagy a bárki által létrehozottakat.

Details(int id, string returnAction)

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

Ez az Action visszadja a kiválasztott edzés részletes adatai. Ebbe beletartozik, hogy melyik feladatot hányszor és hány körben kell elvégezni. A viewModel.ReturnAction eltárolja, hogy a Details View-ról melyik oldalra kell majd visszatérni, mivel ez az Action több oldalról is elérhető.

Create(IFormCollection form)

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

Az Action a kapott IFormCollection-ből kiszedi az adatokat egy változóba, amik alapján létrehozza az edzést. Ezután végigmegy a feladatok számán és létrehozza a Kapcsolótábla adatait, itt kerül beállításra, hogy melyik feladatból hány kört és hány ismétlést kell megcsinálni.

### 2.3.4 Fontosabb View-k

User/Home

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

A lap első fele tartalmazza a két diagrammot és a középső táblázatot a napi összegzéssel. A lap második fele tartalmazza az 5 étkezés típus részletes tápanyagtartalmát a mai napra. A gombra kattintva lehet megnézni az adott étkezéshez tartozó összes ételt.

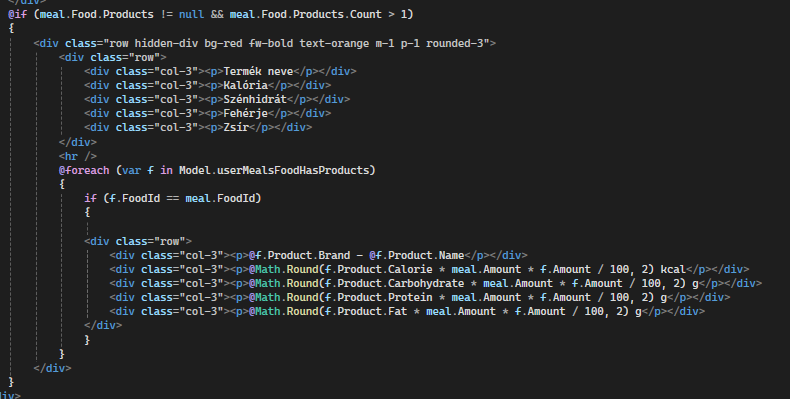
Meal/\_FoodPartial

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Automatikusan generált leírás

Ez a Partial View az EditIntake.cshtml-be töltődik be, ahol az összes ahhoz a naphoz és ahhoz a étkezés típushoz tartozó ételeket és adatait.

Ha az étel több mint 1 termékből áll akkor egy gombbal meg lehet jeleníteni az összes termék részletét.



Meal/\_FoodAddPartial

A képen szöveg, képernyőkép, szoftver, Operációs rendszer látható

Automatikusan generált leírás

Ez az Partial View az Add.cshtml-ben van benne, ahonnan ételeket lehet felvenni a napra és azon belül egy étkezéshez. Az oldal 100g vagy 100ml-ra írja ki az étel adatait.

Ha több mint 1 terméket tartalmaz az étel akkor a Viewmodel 1 függvényét használva kell lekérni az adatokat mivel a kapcsolótábla több sora is szükséges.

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

A GetFoodDatas visszaadja az ételhez tartozó összes termék tápanyagainak összegét. A GetProductDatas függvény egy étel és egy termék azonosító segítségével kikeresi a megfelelő bejegyzést a kapcsolótáblában és kiszámolja az arra az egy termékre való tápanyag adatokat.

Ezt a függvényt az oldal akkor használja amikor az ételhez tartozó termékek részleteit írja ki.

A képen szöveg, képernyőkép, szoftver, Betűtípus látható

Automatikusan generált leírás

Workout/UserWorkouts

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

Ez a View dátumonként csoportosítva kiírja a felhasználó összes eddigi edzéseit. Erről az oldalról megtekinthető a részletes leírása az edzésnek.

Workout/\_WorkoutPartial

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

Erről a Parital View-ról lehet hozzáadni a mai naphoz az edzéseket.

Workout/\_ExercisePartial

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Ez a Partial View egy adott edzésnek a gyakorlatait írja le részletesen.

## 2.4 Tesztelés

A Program 2 tesztet tud futtatni. A tesztosztály felépítése:

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

Az egyik le ellenőrzi az adatbázis működését pár adat feltöltésével. A másik teszt pedig egy Controller Action működését ellenőrzi le.

A képen képernyőkép, szöveg, szoftver, Multimédiás szoftver látható

Automatikusan generált leírás

## 2.5 Fejlesztési lehetőségek

A program több helyen is tovább fejleszthető. Elsősorban az edzés és a mozgás része az, ami nagyon kevés figyelmet kapott és rengeteg apró fejleszteni való dolog van benn. Például vannak olyan gyakorlatok, amiket nem csak kör és ismétlés számmal lehet végezni pl.: futás. A feladatok között a program jelenleg nem tesz különbséget. Továbbá a program jelenleg nem rendelkezik a legjobb designnal ez többet között annak köszönhető, hogy rengeteg oldalt kellett elkészíteni ezért csak egy egyszerű designt hoztam létre. Az projectben továbbá hasznos lenne majdnem az összes View-hoz egy ViewModel-t létrehozni mivel sokkal több adat lehet benne, mint jelenleg és ezeket nem a legjobb döntés ViewData-ban szállítani. Külön Service-ek kialakítása is segíthet a program átláthatóságában és kezelésében. Egy teljes mobilalkalmazás lefejlesztése is jó lenne mivel ott szebben és kényelmesebben ki lehetne alakítani a felhasználói felületet. A mobilalkalmazás mellett ki lehetne alakítani, hogy a projectnek legyen teljesenen kettéválasztott frontend és backend része, C# ASP.NET CORE Api mint backend és egy javascript keretrendszer (Angular, React) mint a frontend része, így gyorsulna a szerver válaszideje. A programban továbbá hasznos lenne egy teljes admin felület kialakítása, de mivel jelenleg csak én vagyok az admin ezért nem lettek a legszebben és leghasznosabban kialakítva ezek a felületek. Továbbá egy kommunikációs felület kialakítása, ahol a felhasználók tudnak egymással beszélgetni és tippeket megosztani egymásnak. Emellett személyi edzőknek is lehetne felületet csinálni, ahol tudnak maguknak választani személyeket akiknek javasolnak étrendet és edzéseket.

# 3. Felhasználói dokumentáció

## 3.1 Program célja és funkciója

A program célja, hogy a felhasználók könnyen és egyszerűen tudjanak az étrendjükre figyelni és követni, a program grafikonokkal és tápanyag ajánlásokkal próbálja meg elősegíteni a célok elérését. A program legnagyobb előnye, hogy bármennyi és bármilyen ételeket létre lehet hozni ezáltal könnyen be lehet írni az összes terméket és összetevőt, amit a nap során elfogyaszt a felhasználó. A program továbbá egy egyszerű edzés készítési és követési rendszerrel is rendelkezik.

## 3.2 Szükséges hardvereszközök

A program futtatásához szükséges egy ASP.NET futtatását támogató operációs rendszer, ehhez egy Windows megfelel. Hardverkövetelmények terén elgének kell lennie 2Ghz vagy gyorsabb processzornak 4GB vagy több RAM memóriának és nagyjából 10Gb szabad helynek. Egy adatbázis szerverre is szükség van, aminek egy MySQL szervernek kell lennie. A programhoz továbbá szükséges internetkapcsolat és egy böngésző, ahonnan el lehet érni a programot.

## 3.3 Telepítés és indítás

### 3.3.1 Telepítési lépések

Ellenőrizni kell, hogy az operációs rendszer támogatja-e az ASP.NET futtatását. Ha nem telepíteni kell a .Net keretrendszert és az ASP.NET keretrendszer. Ellenőrizni kell továbbá, hogy van e MySQL szerver az operációs rendszeren, ha nincs akkor egyet telepíteni kell. Ezután a mellékelt tömörített fájl kicsomagolásával vagy a GitHub fiókomról a project letöltésével telepíthető a program. GitHub elérés: <https://github.com/Ol1v3r09/FitnessForge>

### 3.3.2 Program indítása

Meg kell győződni róla, hogy megfelelően van konfigurálva a webszerver. Ezután egy fejlesztői környezet pl.: Visual Studio segítségével indítható a program. Ezután ellenőrizni kell, hogy elérhető-e az alkalmazás a böngészőből. Ha a programot csak használni szeretné a program ezen a webhelyen megtalálható.: <https://ho.m108.eu>

## 3.4 Részletes bemutatás

A program összes funkciója elérhető a felső menüsorból.



Ebből a 6 menüből egy felhasználónak a „Termék Jóváhagyása” és a „Kezelés” fül nem elérhető mert ezek a funkciók csak egy Manager vagy egy Admin ranggal rendelkező felhasználónál elérhető.

A Napi fogyasztás fülre kattintva ez az oldal fog megjelenni.

A képen szöveg, képernyőkép, menü látható

Automatikusan generált leírás

Ez az oldal egy áttekintés az összes mai napi tápanyagbevitelre. Az alsó + gombokra kattintva az adott étkezések részletes információit lehet megtalálni.



Ezen az oldalon a mai nap reggeliéhez felvett ételek láthatóak, ha az étel több mint 1 termékből áll akkor a Név és tömeg alatti nyíl gombra kattintva megtekinthetőek a termékek részletei. A törlés gombra kattintva eltávolítható az étel az adott napi étkezésből. A kereső pedig név alapján szűri az ételeket. Az étel felvétele gombra kattintva ez az oldal fog megjelenni.

A képen képernyőkép, szöveg, szoftver, tervezés látható

Automatikusan generált leírás

Ezen az oldalon az összes jóváhagyott étel megtalálható, kivéve azok, amik már szerepelnek abban az étkezésben, ahonnan erre az oldalra jöttünk. Itt egy darab beviteli mező van a mennyiségnél, ahova ha hozzászeretnénk adni az adott ételt az étkezésünkhöz ki kell tölteni és utána lehet csak a + gombbal hozzáadni az ételt. A kereső itt is az ételek neve alapján szűr.

A termék felvétele menü után ez az oldal fog megjelennni.

A képen szöveg, képernyőkép, Betűtípus, Téglalap látható

Automatikusan generált leírás

Itt egy három lapból álló űrlap található, ha az összes adatot kitöltöttük, ami egy termékhez szükséges akkor a 3. lap végén a sárga pipára kattintva felvehető az termék. Ezt a terméket még egy Manager vagy Admin-nak jóvá kell hagynia ahhoz, hogy elérhető legyen.

Az új étel készítése menüre kattintva ez az oldal jelenik meg.

A képen szöveg, képernyőkép, tervezés látható

Automatikusan generált leírás

Itt az étel neve és receptje megadása után ki kell választani a bal oldali táblázatból azokat a termékeket, amiket az étel tartalmaz, ha egy terméket áthelyezünk a jobb oldali oszlopba, amit a termék melletti + gombbal lehet megtenni, a beviteli mezőt ki kell tölteni mielőtt elmentjük az ételt. A mennyiség az a tömeg, ami az egész étel elkészítése után lett, illetve az a mértékegység, amiben ez le lett mérve.

Az edzéseim menüre kattintva ez az oldal jelenik meg.

A képen szöveg, képernyőkép, sárga, Betűtípus látható

Automatikusan generált leírás

itt látható az összes eddigi edzése a felhasználónak, évszámonként csoportosítva.

A részletek gombra kattintva megtekinthető az adott edzés feladatsora.:

A képen képernyőkép, sor látható

Automatikusan generált leírás

A + gombra kattintva már a meglévő edzések közül lehet felvenni a mai naphoz egy edzést.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Automatikusan generált leírás

Itt látható az összes edzés, ezeket két szűrő alapján lehet szűrni, vagy csak a felhasználó által készített edzéseket lehet megtekinteni vagy az összes felhasználó által készítetteket. A részletek gomb az előbb bemutatott oldalra visz, a + gomb pedig hozzáadja az edzést a mai naphoz.

Az új edzés létrehozása gombra kattintva ez a felület jelenik meg.

A képen szöveg, képernyőkép, Betűtípus, tervezés látható

Automatikusan generált leírás

Az oldal ugyanazon az elven működik, mint a korábban bemutatott étel létrehozása, a név kitöltése után a jobb oldali oszlopba tudjuk kigyűjteni a gyakorlatokat, ha a jobb oszlopban van egy feladat akkor a létrehozás gomb lenyomása előtt ki kell tölteni a két beviteli mezőt.

A felhasználó részletes adatai oldal.:

A képen szöveg, képernyőkép, Betűtípus, tervezés látható

Automatikusan generált leírás

Itt lehet a felhasználó részletes adatait megadni, ez az oldal a regisztráció után jelenik meg illetve a beállítások menüpontban is elérhető ez az oldal. Ezen az oldalon a Heti cél súlyváltozáshoz csak pozitív számot lehet felvinni, de a program magától át fogja írni negatívra, ha a cél súly kevesebb mint a jelenlegi súly. Ez az érték azt jelzi, hogy a felhasználó hány kilót szeretne fogyni egy hét alatt.

## 3.5 Hibajelzések

### 3.5.1 Bejelentkezés oldalon a lehetséges hibák:

A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Automatikusan generált leírás

Sikertelen bejelentkezés:

Elrontotta a felhasználónevet vagy a jelszavát. Ellenőrizze le ezeket és próbálkozzon újra.

The UserName field is required:

Üresen hagyta a felhasználónév mezőt. Töltse ki a felszanálónév mezőt.

The Password field is required:

Üresen hagyta a jelszó mezőt. Töltse ki a felhasználónév mezőt.

3.5.2 Regisztráció oldalon előfordulható hibák:

A képen szöveg, képernyőkép, Betűtípus, tervezés látható

Automatikusan generált leírás

The UserName field is required:

Üresen hagyta a felhasználónév mezőt. Töltse ki a felszanálónév mezőt.

The Email field is required:

Üresen hagyta az email mezőt. Töltse ki az email mezőt.

The Password field is required:

Üresen hagyta a jelszó mezőt. Töltse ki a felhasználónév mezőt.

A képen szöveg, képernyőkép, Betűtípus, tervezés látható

Automatikusan generált leírás

A felhasználónév és a jelszónál előforduló hiba esetén teljesítse az ott leírt utasítást.

The Email field is not a valid e-mail address:

A megadott email cím nem megfelelő. Javítsa ki a hibát, rakjon @ karaktert és megfelelő cím legyen az email vége például: [teszt@gmail.com](mailto:teszt@gmail.com)

Ezeken kívül még előfordulhatnak az oldal tetején megjelenő angol nyelvű hibaüzenetek. Ezek a jelszó helytelensége miatt jelenhetnek meg. A jelszónak tartalmaznia kell legalább 1 nagybetűt, legalább 1 kisbetűt, és legalább 1 számot.

Ezenkívül a programban előfordulhat, hogy a nem kitöltött beviteli mezőknél megjelenik egy „Kérjük, töltse ki ezt a mezőt” nevű felirat, ilyenkor ki kell tölteni azt a mezőt is a továbbhaladáshoz.

## 3.6 Információkérési lehetőségek

A programban van lehetőség visszajelzés küldeni a programmal kapcsolatban. Minden oldal alján továbbá megtalálható egy pár elérhetőségem. További kérdések és információkérésre pedig az email-re lehet küldeni levelet: [horvatholiver0929@gmail.com](mailto:horvatholiver0929@gmail.com).

# 4. Összefoglalás, köszönetnyilvánítás

Összefoglalva ez az első nagyobb alkalmazásom, amit egyedül írtam, ehhez képest szerintem elég jól sikerült. A program nagysága jóval nagyobb, mint amennyi idő volt az elkészítésére, de így is a legtöbbet hoztam ki belőle véleményem szerint. Több apróbb a funkciót nem befolyásoló hiba van még a programban, amit a jövőben tervezek kijavítani. Sőt a program nagyrészét újra tervezem írni mivel a program írása közben is többször átírtam volna sok mindent, de ha ezt tettem volna nem készültem volna el a projekttel. A program alapjain minimális változtatást tennék, de több különálló Service-t és több ViewModel-t is tennék a programba a jobb átláthatóság és a könnyebb fejlesztés miatt. Továbbá tervezem a program átírását egy ASP.NET Api backend és egy Angular vagy React frontend átírására. Ezután egy mobil alkalmazást is írnék mellé. A program írása közben a legnagyobb kihívás az volt, hogy voltak olyan pillanatok főleg a cshtml-ek írása közben amikor nem tudtam, hogy pontosan mit is akarok megvalósítani és ezáltal kicsit átláthatatlan lett a program, erre a megoldás az lett volna, ha több kicsi részre, service, viewmodel stb. használatával készítettem volna a programot. Köszönettel tartozom az összes szakmai tanáromnak, akik megtanítottak programozni és akik segítsége nélkül jónéhány helyen megakadtam volna a program készítésénél. Továbbá a szüleimnek is köszönettel tartozom, akik támogattak tanulmányaim során és segítettek tanácsaikkal.

# 5. Irodalomjegyzék

Microsoft Learn: <https://learn.microsoft.com/en-us/>

Stackoverflow: <https://stackoverflow.com/>

ChatGPT: <https://chat.openai.com/>

Chart.js: <https://www.chartjs.org/docs/latest/>

W3schools: <https://www.w3schools.com/>