**Ceglédi SZC Mihály Dénes Szakképző Iskola**

**Étkezz Okosan Alkalmazás**

**Készítette:**

*Tollár András*

*Dékány Tibor*

**Szoftverfejlesztők**

Gyömrő

2022

Tartalomjegyzék:

[1. Bevezetés 4](#_Toc102738102)

[2. Felhasználói dokumentáció 4](#_Toc102738103)

[2.1. Hardware és Software igény 4](#_Toc102738104)

[2.2. Használat 5](#_Toc102738105)

[2.3. Funkciók ismertetése \*(Asztali Alkalmazás) 7](#_Toc102738106)

[2.3.1. Főoldal 7](#_Toc102738107)

[2.3.2. Kalória Kalkulátor 8](#_Toc102738108)

[2.4. Funkciók ismertetése \*(Webalkalmazás) 10](#_Toc102738109)

[2.4.1. Főoldal 10](#_Toc102738110)

[2.4.2. Kalória Kalkulátor 11](#_Toc102738111)

[3. Fejlesztői Dokumentáció 13](#_Toc102738112)

[3.1. Adatbázis táblái és azok mezői 13](#_Toc102738113)

[3.1.1. Reggeli Tábla 13](#_Toc102738114)

[3.1.2. Ebéd (Leves) Tábla 14](#_Toc102738115)

[3.1.3. Ebéd (Főétel) Tábla 15](#_Toc102738116)

[3.1.4. Vacsora Tábla 16](#_Toc102738117)

[3.1.5. Sportok Tábla 17](#_Toc102738118)

[3.1.6. Hét napjai Tábla 18](#_Toc102738119)

[3.2. Algoritmusok az alkalmazásban 19](#_Toc102738120)

[3.2.1. Asztali Alkalmazás 19](#_Toc102738121)

[3.2.1.1. Kalória kalkulátor 19](#_Toc102738122)

[3.2.1.2. Hét napjai 23](#_Toc102738123)

[3.2.1.3. Új reggeli -Új Leves -Új Főétel -Új Vacsora 24](#_Toc102738124)

[3.2.1.4. Tápérték tartalom számítás 25](#_Toc102738125)

[3.2.2. Webalkalmazás 26](#_Toc102738126)

[3.2.2.1. Hét napjai 27](#_Toc102738127)

[3.2.2.2. Új Reggeli -Új Leves -Új Főétel –Új Vacsora 28](#_Toc102738128)

[3.3. Tesztdokumentáció 29](#_Toc102738129)

[4. Összefoglalás 29](#_Toc102738130)

[4.1. Szakdolgozat célja 29](#_Toc102738131)

[4.2. Megvalósítása 29](#_Toc102738132)

[4.3. Fejlesztési lehetőségek 29](#_Toc102738133)

[5. Irodalomjegyzék 30](#_Toc102738134)

# 

# 1. Bevezetés

A digitális fejlődés és az elmúlt évek arra ösztönöztek bennünket, hogy a 2022-es szakdolgozatunk témájának a táplálkozást válasszuk. Ezért készítettünk egy programot, amely az Étkezz okosan elnevezést kapta. A program segíti a napi menünk kiválasztását, oly módon, hogy a program már a betöltéskor is random módon ajánl ételt számunkra, de ha valamely étel nem felel meg az ízlésünknek, akkor bármikor változtathatunk rajta kedvünk szerint.

# 2. Felhasználói dokumentáció

Az alkalmazás célja, hogy segítsen a napi menünk összeállításában és egyben információt ad az ételek tápértékéről.

**Lehetőségeink a következők:**

* Az Új Reggeli /Leves /Főétel /Vacsora gomb megnyomásával új ételt generálhatunk teljesen függetlenül egymástól.
* Kalória kalkuláció segítségével saját adataink alapján (pl.: testtömeg, magasság, kor stb.) megkapjuk az alap BMR értékünket, továbbá étkezésenként láthatjuk a szükséges kalória beviteleket is.

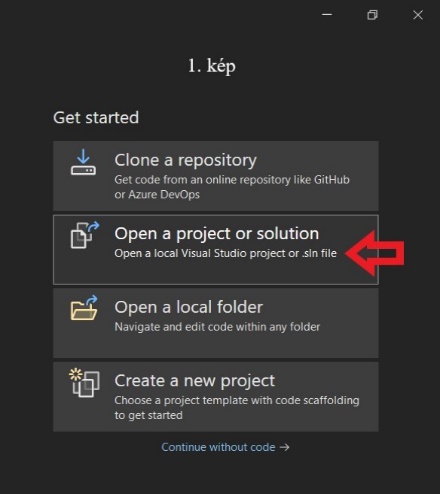
## 2.1. Hardware és Software igény

Az asztali alkalmazás használatához egy alap asztali számítógép vagy egy laptop szükséges, amelyre telepítve van a Visual Studio 2022 alkalmazás.

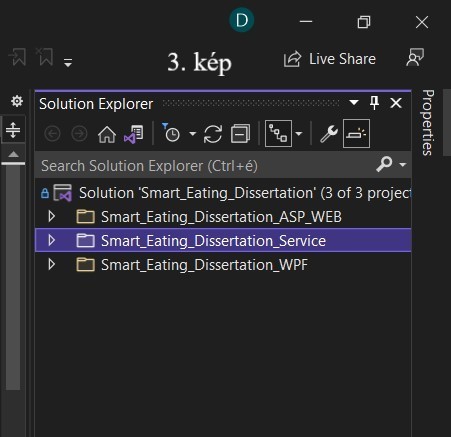
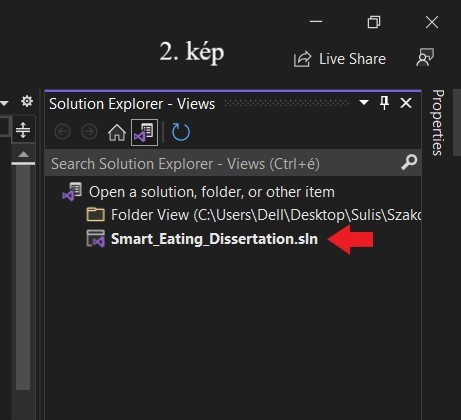
A webes alkalmazás használatához egy alap asztali számítógép vagy egy laptop szükséges, amelyre telepítve van a Visual Studio 2022, valamint egy böngésző program (mint a Chrome, vagy a Microsoft Edge).

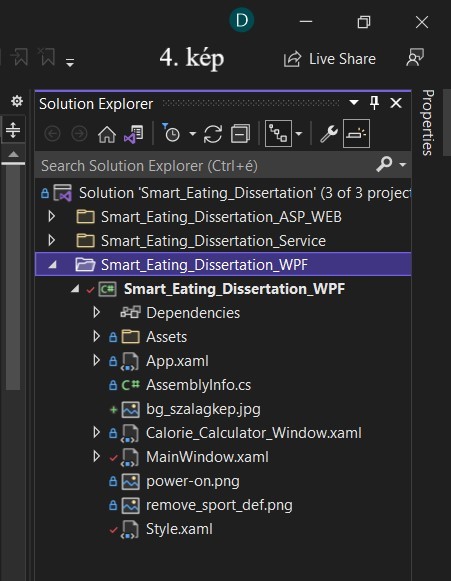
## 2.2. Használat

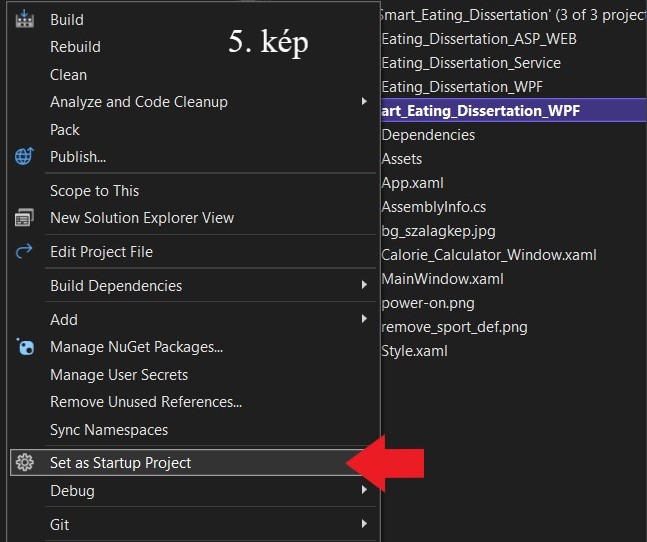
Az asztali WPF alkalmazás elindítható a Smart\_Eating\_Dissertation mappából indulva ezen az elérési úton található -*Smart\_Eating\_Dissertation\ Smart\_Eating\_Dissertation\_WPF\ bin\ Debug\ net6.0windows\ Smart\_Eating Dissertation\_WPF. exe - fájlra* kattintva. Azonban az alkalmazást el lehet indítani, a Visual Studio alkalmazással is, ha az „Open a project or solution” (lásd: 1. kép) lehetőségre kattintunk. Majd megkeressük a számítógépen a Smart\_Eating\_Dissertation mappát, azon belül a Smart\_Eating\_Dissertation.sln fájlt, amire jobb gombbal duplán rákattintunk.



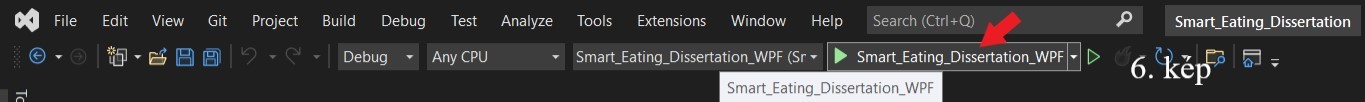
Ezt követőn a Visual Studio alkalmazáson belül a jobb oldalon automatikusan megjelenik majd a Solution Explorer-Views (lásd: 2. kép) itt a Smart\_Eating\_Dissertation.sln fájlra duplán kattintva láthatjuk a teljes alkalmazás fő mappáit (lásd: 3. kép).



A kívánt megjelenítési formára duplán kattintva ASP\_WEB vagy WPF (lásd 4. kép), majd a Smart\_Eating\_Dissertation\_WPF/\_ASP -re jobb gomb „Set as Startup Project” (lásd: 5. kép) lehetőség kiválasztása.



Ezután a programfuttatás (lásd: 6. kép) gombra kattintva elindul az alkalmazás a kiválasztott megjelenítési formában.



Az ASP nézet elindítása és futtatása csak a fent leírt módon a Visual Studio 2022 program segítségével lehetséges.

## 2.3. Funkciók ismertetése \*(Asztali Alkalmazás)

### 2.3.1. Főoldal

A főoldalon található egy szalagkép, amely fent helyezkedik el, mint az asztali alkalmazásban, mint a webalkalmazásban.

A szalagképen található egy **„Power (Lásd: zöld nyíl)”** gomb valamint egy „**kalória kalkulátor (Lásd: piros nyíl)”** gomb mellyel egy új ablakban megadhatjuk az adatainkat, ami alapján a gép számításokat végez.

A következő részben találhatóak a napok melyekből ki is választja a program az aktuális napot.

Ezután 4 gomb látható **„Új Reggeli”**, **„Új Leves”**, **„Új Főétel”**, **„Új Vacsora” (Lásd: narancssárga nyilak)** melyekkel új ételek generálhatók.

Ezek után táblázatosan megjeleníti a program az étkezéseket és a hozzájuk generált ételt valamint az ételek tápértéktartalmát **(Lásd: lila nyíl)**.

Ezután van egy összegzés a fentebb említett tápérték adatok alapján.**(Lásd: kék nyíl)**

Ha történt kalória kalkuláció, akkor a következő sorban jelennek meg a számított értékek. **(Lásd: szürke nyíl)**

Az utolsó sor információ jellegű.



### 2.3.2. Kalória Kalkulátor

A jobb felső sarokban itt is megtalálható a **„Power”** gomb. **(Lásd: piros nyíl)**

A kalória kalkulátor ablakban az alábbi beállításokat végezhetjük el:

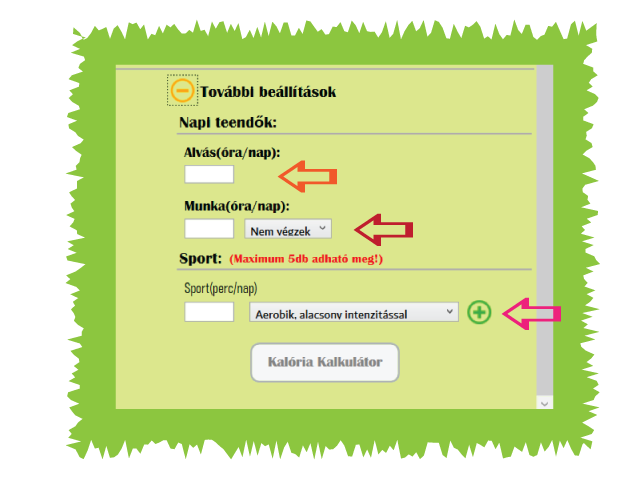
* **Nem:** Itt két lehetőségünk van Férfi, Nő. **(Lásd: zöld nyíl)**
* **Kor:** Ebben a mezőben csak szám adható meg. (Maximum 99 éves korig.) **(Lásd: kék nyíl)**
* **Magasság (cm):** Itt megadható a testmagasságunk. (Maximum: 210 cm-ig.) **(Lásd: barna nyíl)**
* **Tömeg (kg):** Itt megadható a testsúlyunk. (Maximum: 250 kg.) **(Lásd: lila nyíl)**
* **Cél:** Itt kiválaszthatjuk, hogy milyen célunk van: **(Lásd: narancssárga nyíl)**
  + **Szintentartás**
  + **Fogyás**
  + **Tömegnövelés**



* **További beállítások: (Lásd: szürke nyíl)**
  + **Napi teendők:**
    - **Alvás (óra/nap) (Lásd: narancssárga nyíl)**

**Munka (óra/nap) (Lásd: bordó nyíl) => valamint a legördülő listából az alábbiakat tudjuk kiválasztani:**

* + - * **Nem végzek**
      * **Ülőmunka/Tanuló**
      * **Átlagos aktivitású munka**
      * **Könnyű fizikai munka**
      * **Nehéz fizikai munka**
* **Sport: (Maximum 5db adható meg)**
  + **Sport (perc/nap)** és a legördülő listából kiválaszthatjuk az elvégzendő/elvégzett sportot. **(Lásd: magenta nyíl)**



## 2.4. Funkciók ismertetése \*(Webalkalmazás)

### 2.4.1. Főoldal

A kezdőképernyőn elhelyezésre került a fejlécben egy szalagkép a programunk nevével.

A fejléc alatt jobbra igazítva került elhelyezésre a Kalória kalkulátor gomb, amely egy link, de Bootstrap osztály formázással lett létrehozva. Ha rákattintunk, akkor egy új ablak nyílik meg. A Kalória kalkulátor oldal részletes bemutatása a 2.4.2. pontban történik.



A kezdőképernyőn a Kalória kalkulátor gomb alatt kerültek elhelyezésre a hét napjai. A napok jelölése automatikus. A program a számítógép által beállított dátumot figyeli, és az alapján változtatja az aktív és inaktív napok jelölését. Dátumtól függően az aktív napot fehér színnel, az inaktív napokat fekete színnel jelzi.

A hét napjait követően 4 gomb került elhelyezésre: Új reggeli, Új leves, Új főétel, Új vacsora, amelyek segítségével új ételeket generálhatunk a napi menünkbe. Az étkezések gombjai szintén linkek, amelyek Bootstrap osztály formázással lettek kialakítva. Az oldalon a betöltést követően a program Random függvény használatával minden étkezés tekintetében véletlenszerűen kiválaszt egy készételt a feltöltött adatbázisból. Ha valamelyik étel nem felel meg a felhasználó ízlésének, akkor bármelyik étkezés gomb megnyomásával új ételt generálhatunk, amely a gombok alatt egy HTML kódolással létrehozott táblázat megfelelő sorában fog megjelenni. A táblázatban, mint látható az étel nevén kívül megjelenítésre kerül az ételekben található fehérje, zsír, szénhidrát és energia tartalom mennyisége is.

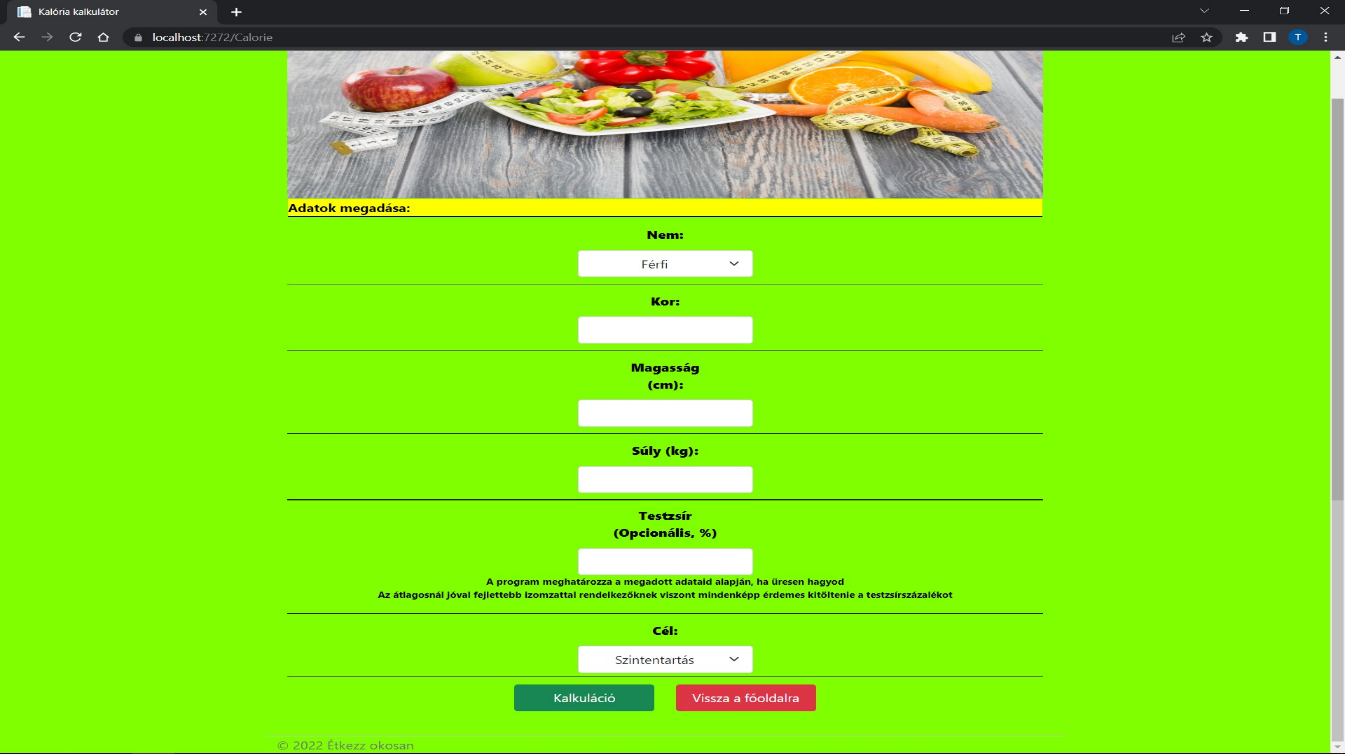
A táblázatban szerepel továbbá egy Összesen sor, amely az ételek tápértéktartalmát összesíti kategóriánként.

Amennyiben a felhasználó a Kalória kalkulátor oldalon kalkulációt végzett, annak eredménye az összesen sor alatt kerül megjelenítésre.

### 2.4.2. Kalória Kalkulátor

A főoldalon található Kalória kalkulátor gomb megnyomásával a felhasználó egy új oldalra kerül, ahol a kalória kalkulátor számára fontos adatokat (pl.: Nem, Kor, Magasság stb.) adhatunk meg és ezek alapján a program kiszámolja, hogy mennyi kalóriát fogyaszthatunk az adott napon annak függvényében, hogy fogyni, hízni vagy csak tartani szeretnénk a súlyunkat.

Az oldal fejlécben elhelyezésre került egy szalagkép, amely alá egy táblázat lett létrehozva HTML kódolással.

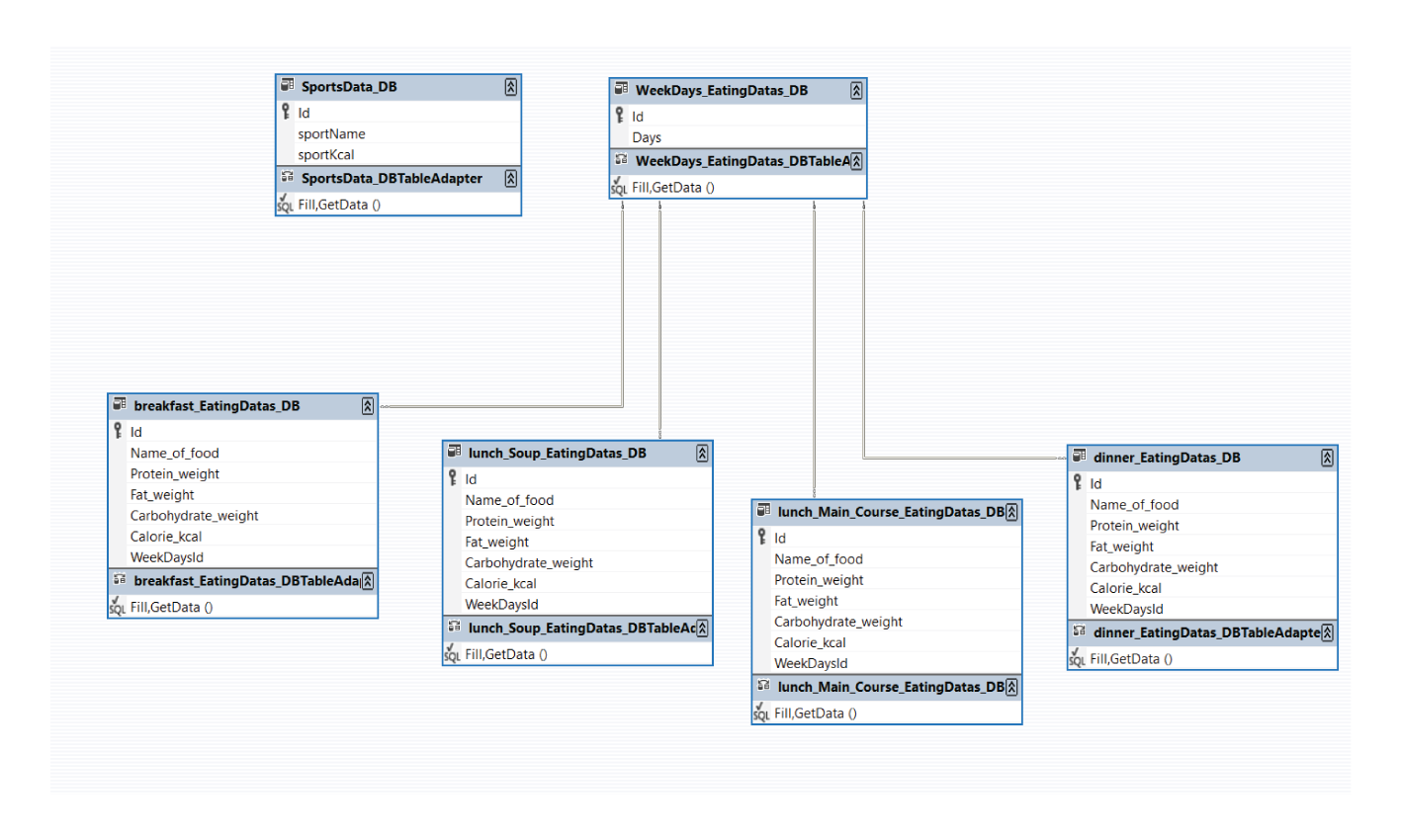


A kalkulációhoz az alábbi adatok megadása szükséges:

* **Nem:** A felhasználó egy lenyíló menüben adhatja meg a nemét, hogy Férfi vagy Nő. Ha nem választja ki a nemet, akkor alapértelmezetten a Férfi beállítás alapján számol a program.
* **Kor:** Itt a felhasználó életkorát kell megadni. Kitöltése kötelező. Amíg ez az adat nem kerül megadásra, addig a program nem hajtja végre a számítást. (Maximum 100 éves korig.)
* **Magasság:** Itt a felhasználónak meg kell adni a testmagasságát centiméterben. Kitöltése kötelező. Amíg ez az adat nem kerül megadásra, addig a program nem hajtja végre a számítást. (Maximum 250 cm-ig)
* **Súly:** Itt a felhasználónak meg kell adni a testtömegét kilogrammban. Kitöltése kötelező. Amíg ez az adat nem kerül megadásra, addig a program nem hajtja végre a számítást. (Maximum 300kg-ig)
* **Testzsír:** Kitöltése nem kötelező. A rendszer a kor, a testmagasság és a testsúly alapján automatikusan kiszámolja. (Maximum 40%-ig)
* **Cél:** Itt a felhasználó egy lenyíló menüben megadhatja, hogy mi a célja a kalkulátor eredményével. Amennyiben nem adunk meg mást akkor a program alapértelmezetten a testsúly megtartásával számol.
  + Fogyás
  + Testtömeg növelése (Tömegnövelés)
  + Testsúly megtartása (Szinten tartás)

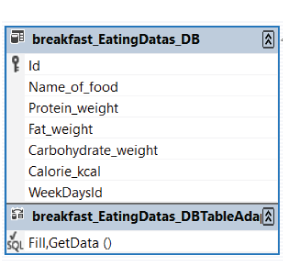
# 3. Fejlesztői Dokumentáció

## 3.1. Adatbázis táblái és azok mezői



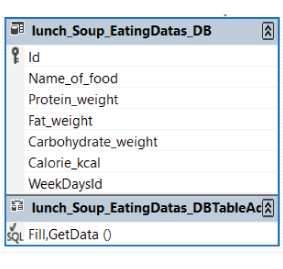
Az oldalon megjelenített adatok a fent látható táblákból érkeznek, különböző lekérések segítségével, melyet az alkalmazás algoritmusai tesznek lehetővé.

### 3.1.1. Reggeli Tábla



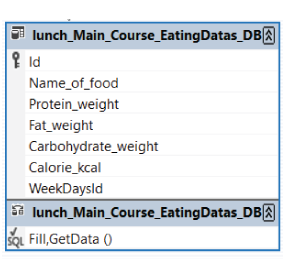
* Elsősorban a reggelik adatainak tárolására szolgál. Az első mező egy „Id” ez szám típusú mező. Ez a tábla elsődleges kulcsa, automatikusan növekszik és generálódik, ahányszor egy új sort/új reggelit hozzáadunk.
* A második mező „Name\_of\_food” varchar (max) mező ezért szöveget és számot is tartalmazhat és kötelező megadni. Ez, ahogy a neve is elárulja, az étel nevét tartalmazza.
* A harmadik mező „Protein\_weight” float típusú mező, vagyis az itt megadott érték lebegőpontos. Ezt a mezőt is kötelező kitölteni. Ebben a mezőben az étel fehérje tartalmát tároljuk.
* A negyedik mező a „Fat\_weight” ez is float típusú mező mivel itt is tizedes pontossággal kell megadni az értéket. Ezt a mezőt is kötelező kitölteni. Ebben a mezőben az étel zsírtartalmát kell megadni.
* Az ötödik mező „Carbohydrate\_weight” ez a mező is float típusú mivel úgy, ahogy az előző mezőknél itt is tizedes pontossággal kell megadni a szénhidrát tartalmát az ételnek. Ezt a mezőt is kötelező kitölteni.
* A hatodik mező „Calorie\_kcal” itt ugyan úgy float típusú mivel itt is tizedes pontossággal adjuk meg az értéket. Ez a mező a kalória tárolására szolgál. Ezt is kötelező kitölteni.
* Az utolsó mező „WeekDaysId” ez a mező szám típusú valamint másodlagos kulcs. Itt kapcsolódik a reggeli táblánk a „WeekDays” táblához.

### 3.1.2. Ebéd (Leves) Tábla



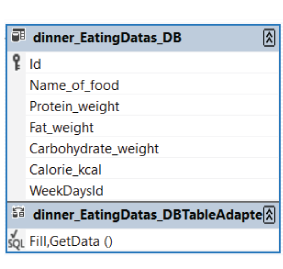
* Elsősorban az ebéd (leves) adatainak tárolására szolgál. Az első mező egy „Id” ez szám típusú mező. Ez a tábla elsődleges kulcsa, automatikusan növekszik és generálódik, ahányszor egy új sort/új ebéd (leves)-t hozzáadunk.
* A második mező „Name\_of\_food” varchar (max) mező ezért szöveget és számot is tartalmazhat és kötelező megadni. Ez, ahogy a neve is elárulja, az étel nevét tartalmazza.
* A harmadik mező „Protein\_weight” float típusú mező, vagyis az itt megadott érték lebegőpontos. Ezt a mezőt is kötelező kitölteni. Ebben a mezőben az étel fehérje tartalmát tároljuk.
* A negyedik mező a „Fat\_weight” ez is float típusú mező mivel itt is tizedes pontossággal kell megadni az értéket. Ezt a mezőt is kötelező kitölteni. Ebben a mezőben az étel zsírtartalmát kell megadni.
* Az ötödik mező „Carbohydrate\_weight” ez a mező is float típusú mivel úgy, ahogy az előző mezőknél itt is tizedes pontossággal kell megadni a szénhidrát tartalmát az ételnek. Ezt a mezőt is kötelező kitölteni.
* A hatodik mező „Calorie\_kcal” itt ugyan úgy float típusú mivel itt is tizedes pontossággal adjuk meg az értéket. Ez a mező a kalória tárolására szolgál. Ezt is kötelező kitölteni.
* Az utolsó mező „WeekDaysId” ez a mező szám típusú valamint másodlagos kulcs. Itt kapcsolódik az Ebéd (Leves) táblánk a „WeekDays” táblához.

### 3.1.3. Ebéd (Főétel) Tábla



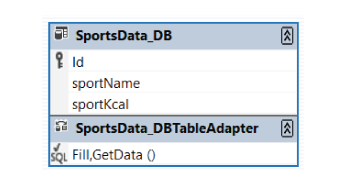
* Elsősorban az ebéd (főétel) adatainak tárolására szolgál. Az első mező egy „Id” ez szám típusú mező. Ez a tábla elsődleges kulcsa, automatikusan növekszik és generálódik, ahányszor egy új sort/új ebéd (főétel)-t hozzáadunk.
* A második mező „Name\_of\_food” varchar (max) mező ezért szöveget és számot is tartalmazhat és kötelező megadni. Ez, ahogy a neve is elárulja, az étel nevét tartalmazza.
* A harmadik mező „Protein\_weight” float típusú mező, vagyis az itt megadott érték lebegőpontos. Ezt a mezőt is kötelező kitölteni. Ebben a mezőben az étel fehérje tartalmát tároljuk.
* A negyedik mező a „Fat\_weight” ez is float típusú mező mivel itt is tizedes pontossággal kell megadni az értéket. Ezt a mezőt is kötelező kitölteni. Ebben a mezőben az étel zsírtartalmát kell megadni.
* Az ötödik mező „Carbohydrate\_weight” ez a mező is float típusú mivel úgy, ahogy az előző mezőknél itt is tizedes pontossággal kell megadni a szénhidrát tartalmát az ételnek. Ezt a mezőt is kötelező kitölteni.
* A hatodik mező „Calorie\_kcal” itt ugyan úgy float típusú mivel itt is tizedes pontossággal adjuk meg az értéket. Ez a mező a kalória tárolására szolgál. Ezt is kötelező kitölteni.
* Az utolsó mező „WeekDaysId” ez a mező szám típusú valamint másodlagos kulcs. Itt kapcsolódik az Ebéd (Főétel) táblánk a „WeekDays” táblához.

### 3.1.4. Vacsora Tábla



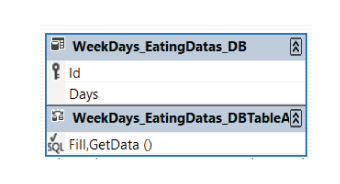
* Elsősorban a vacsora adatainak tárolására szolgál. Az első mező egy „Id” ez szám típusú mező. Ez a tábla elsődleges kulcsa, automatikusan növekszik és generálódik, ahányszor egy új sort/új vacsorát hozzáadunk.
* A második mező „Name\_of\_food” varchar (max) mező ezért szöveget és számot is tartalmazhat és kötelező megadni. Ez, ahogy a neve is elárulja, az étel nevét tartalmazza.
* A harmadik mező „Protein\_weight” float típusú mező, vagyis az itt megadott érték lebegőpontos. Ezt a mezőt is kötelező kitölteni. Ebben a mezőben az étel fehérje tartalmát tároljuk.
* A negyedik mező a „Fat\_weight” ez is float típusú mező mivel itt is tizedes pontossággal kell megadni az értéket. Ezt a mezőt is kötelező kitölteni. Ebben a mezőben az étel zsírtartalmát kell megadni.
* Az ötödik mező „Carbohydrate\_weight” ez a mező is float típusú mivel úgy, ahogy az előző mezőknél itt is tizedes pontossággal kell megadni a szénhidrát tartalmát az ételnek. Ezt a mezőt is kötelező kitölteni.
* A hatodik mező „Calorie\_kcal” itt ugyan úgy float típusú mivel itt is tizedes pontossággal adjuk meg az értéket. Ez a mező a kalória tárolására szolgál. Ezt is kötelező kitölteni.
* Az utolsó mező „WeekDaysId” ez a mező szám típusú valamint másodlagos kulcs. Itt kapcsolódik a vacsora táblánk a „WeekDays” táblához.

### 3.1.5. Sportok Tábla



* Elsősorban a különböző sportok adatainak tárolására szolgál. Az első mező egy „Id” ez szám típusú mező. Ez a tábla elsődleges kulcsa, automatikusan növekszik és generálódik, ahányszor egy új sort/sportot hozzáadunk.
* A második mező „sportName” varchar (max) mező ezért szöveget és számot is tartalmazhat és kötelező megadni. Ez, ahogy a neve is elárulja, az adott sport megnevezését tartalmazza.
* A harmadik mező „sportKcal” varchar (max) mező ezért szöveget és számot is tartalmazhat és kötelező megadni. Ebben a mezőben az adott sport kcal/kg/perc értéket tartalmazza, amelyből a kalória kalkulátor kiegészítő alkalmazás kiszámítja a megfelelő értéket, amely aztán megjelenik a főablakban.

### 3.1.6. Hét napjai Tábla



* Ez egy segéd tábla és a hét napjainak tárolására szolgál. Az első mező egy „Id” ez szám típusú mező. Ez a tábla elsődleges kulcsa, automatikusan növekszik és generálódik.
* A második mező „Days” varchar (max) mező ezért szöveget és számot is tartalmazhat és kötelező megadni. Ez a mező a hét napjainak nevét tartalmazza.

## Algoritmusok az alkalmazásban

### Asztali Alkalmazás

Az alkalmazás indulásakor minden étkezésből inicializál a program egyet random módon az adatbázisból. **(Lásd: 1. kép)**

A képen a reggeli inicializálása van megjelenítve, de az összes többi étkezés algoritmusa azonos, ezért nem tartottam érdemesnek megjeleníteni/bemutatni.



**1.KÉP**

#### Kalória kalkulátor

Feladata, hogy a felhasználó saját adatai megadásával megkapja a napi maximum bevihető kalóriát és az alapanyagcseréjét (BMR). A kalória kalkulátor algoritmusa lépésenként:

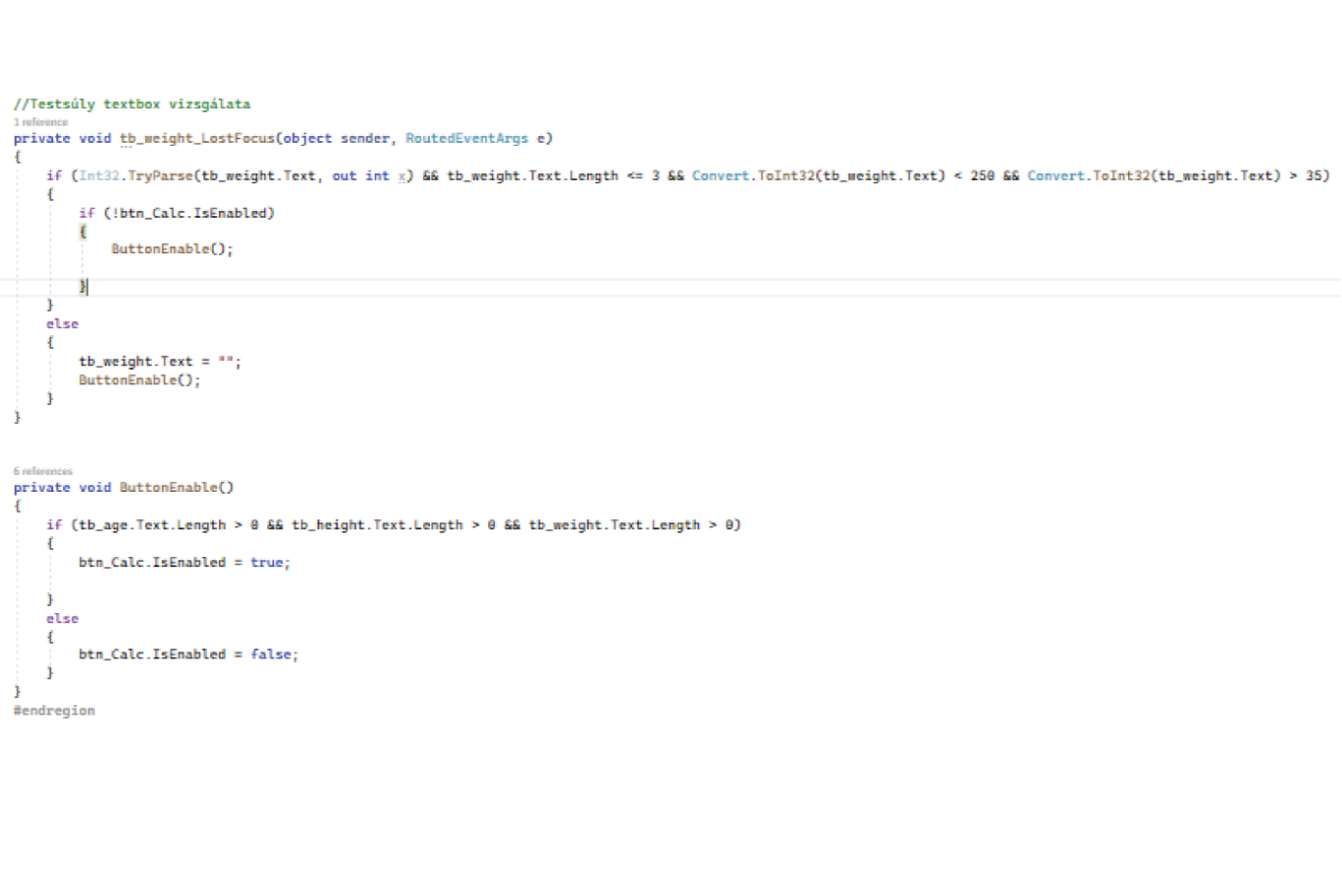
* A felhasználó kitölti mind az öt kötelezően kitöltendő/megadandó mezőt. Ha valamelyik mezőt helytelenül tölt ki, akkor az alkalmazás figyelmeztet rá, valamint ameddig az összes kötelezően kitöltendő mezőt nem tölti ki a felhasználó addig a „Kalória kalkulátor” gomb inaktív marad.
* A mezők formátumának valamint maximális értékének ellenőrzése. **(Lásd: 1/1 és 1/2. kép)**
* A további beállítások lenyíló menüben két rész tölthető ki:
  + Napi Teendők
  + Sport

Ezen részek egymástól függetlenül működnek tehát nem kötelező mind a kettő részt kitölteni mivel a „Napi Teendők” résznek van alapértelmezett értéke. A „Sport” rész kihagyásával ugyan úgy megtörténik a számítás viszont a „Sport”-ot nem veszi figyelembe a program.

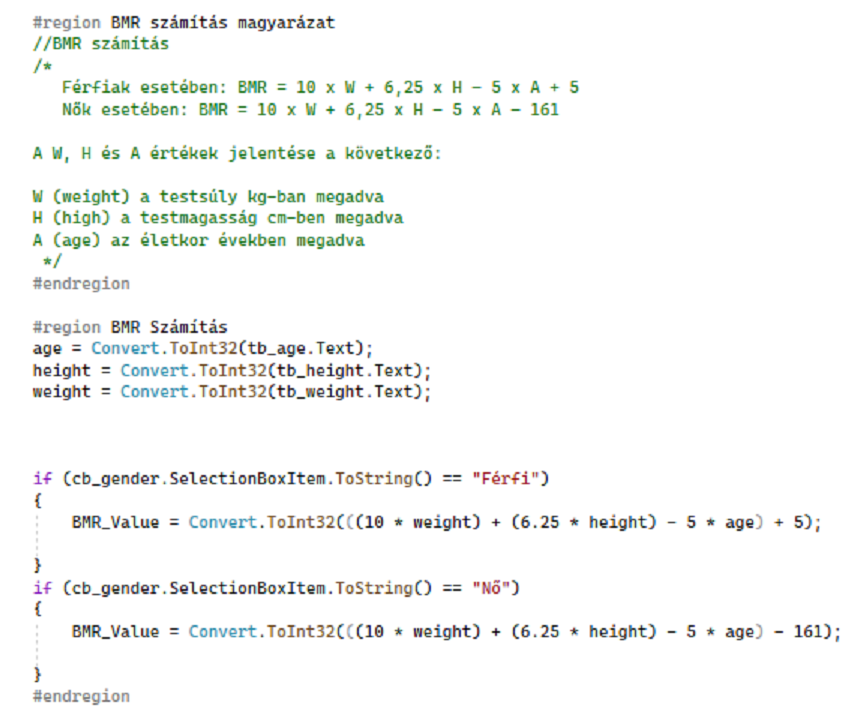
* A mezők helyes kitöltése után a „Kalória kalkulátor” gomb lenyomásával a megadott adatok alapján megtörténik a számítás:
  + BMR számítás **(Lásd: 2. kép)**
  + Kalória szükséglet számítás **(Lásd: 3. kép)**



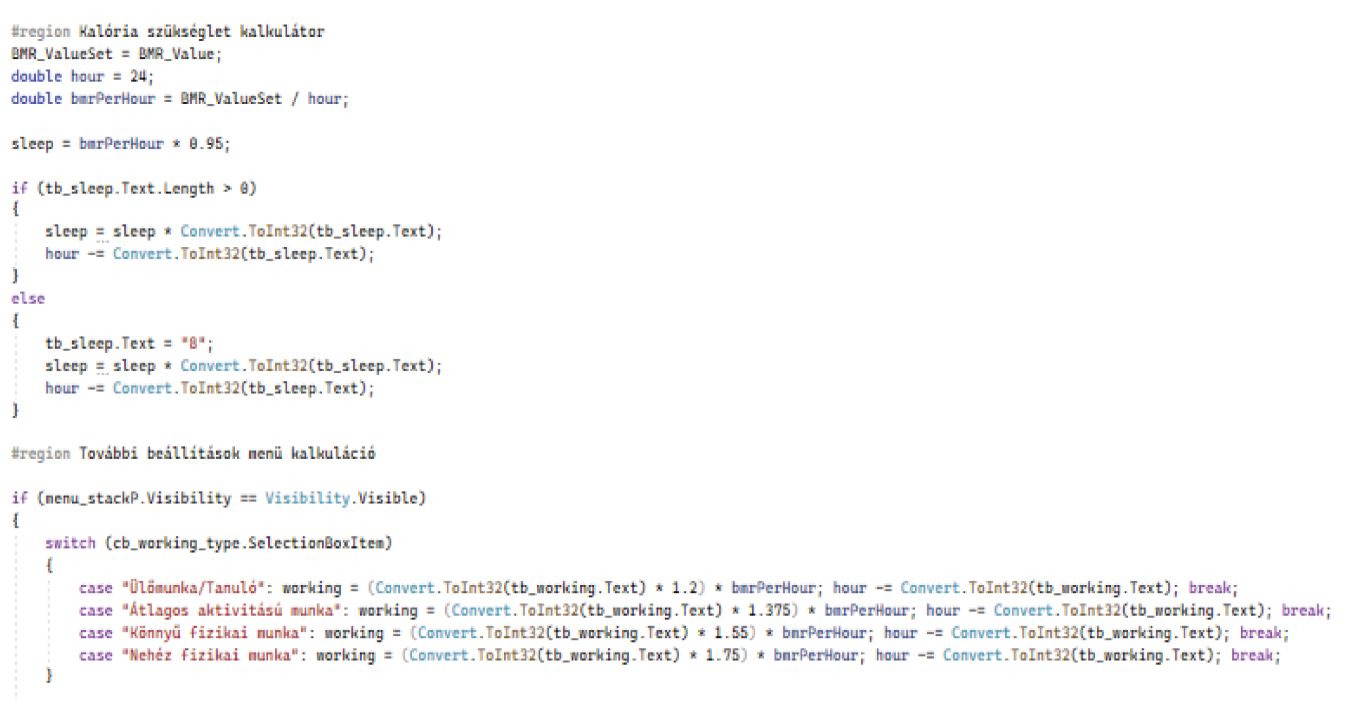
**1/1.KÉP**



**1/2.KÉP**



**2. KÉP**



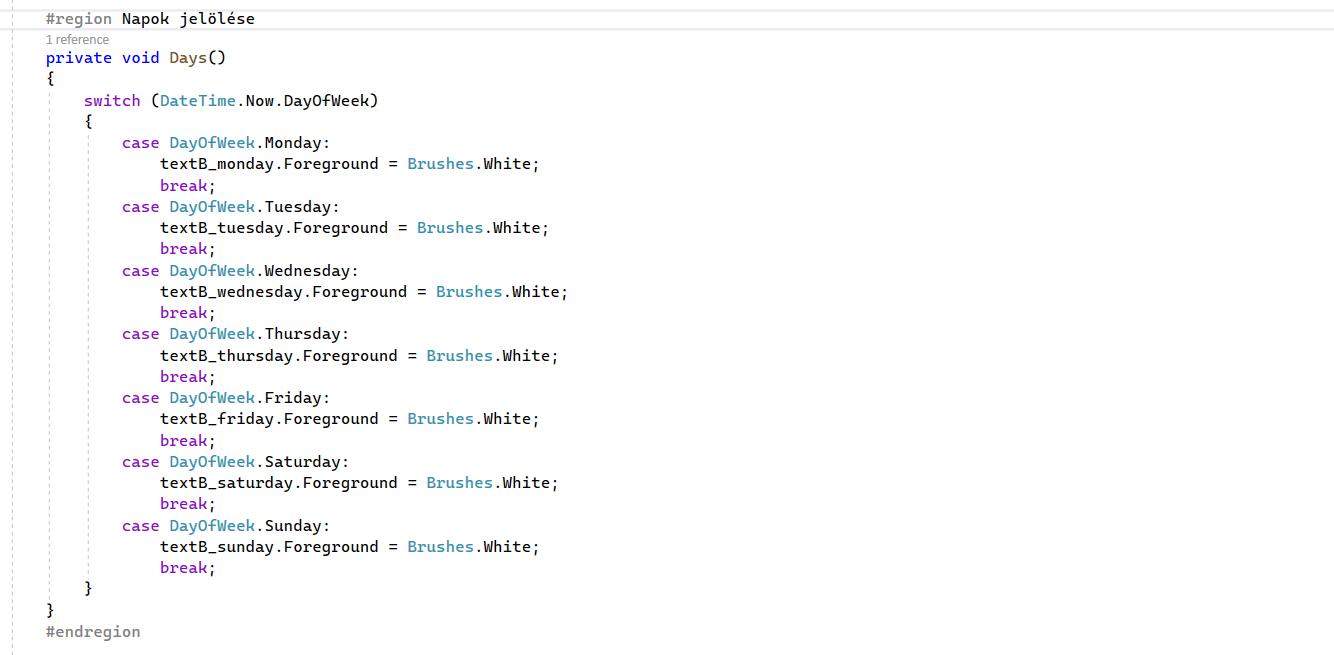
**3/1.KÉP**



**3/2.KÉP**

#### Hét napjai

A célja, hogy a felhasználóval tudassa az aktuális napot. Az algoritmusa csupán annyi, hogy egy „Switch” utasítással a „DateTime” osztály azon belül „Now. DayOfWeek” property segítségével megkapjuk az adott napot és ezzel egyidejűleg a szöveget átszínezzük, hogy látható legyen melyik az aktuális nap.



#### Új reggeli -Új Leves -Új Főétel -Új Vacsora

Ennél az algoritmusnál az étkezések inicializálása metódust használtam fel mivel a programnak ugyan azt a műveletet kell elvégeznie csupán annyi változással, hogy itt nem a program indulására kell, hogy lefusson a metódus, hanem egy gomb megnyomására. Mivel mindegyik étkezés generálása ugyan azzal a metódussal működik ezért egy képen prezentálom.

**(Lásd: 1. kép)**



**1.KÉP**

#### Tápérték tartalom számítás

Ebben az algoritmusban nem történik más, mint hogy összegezi a program az étkezések tápértékeit.

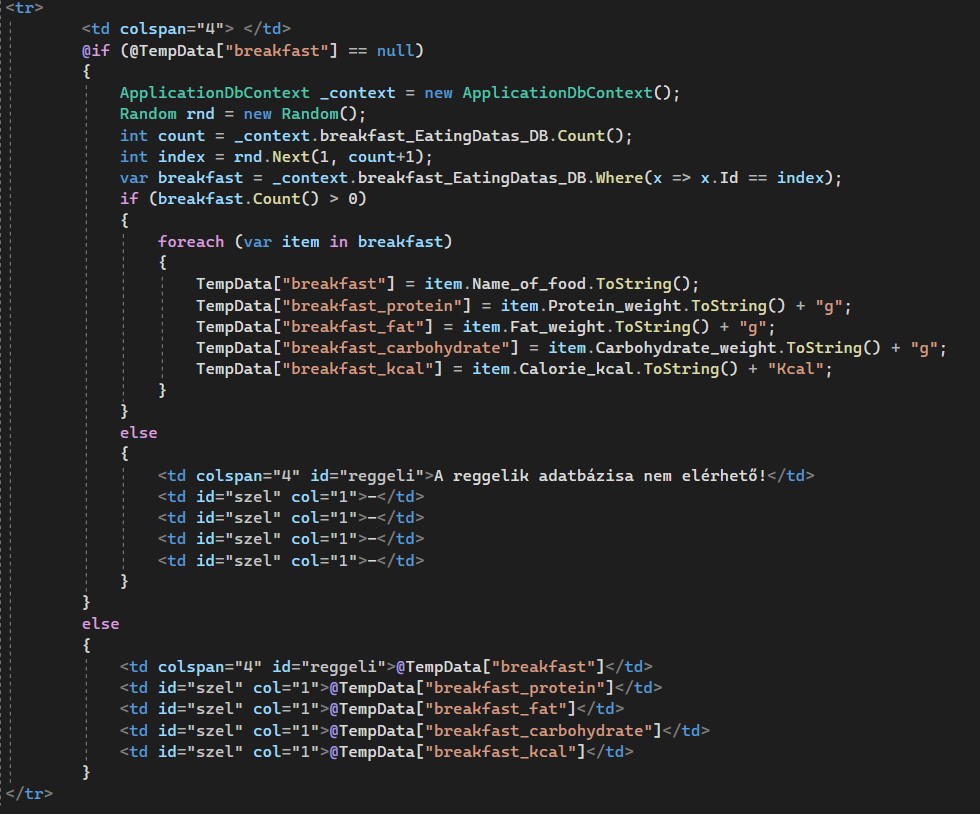
* Fehérje
* Zsír
* Szénhidrát
* Kalória

Az algoritmust prezentáló kép lentebb látható.



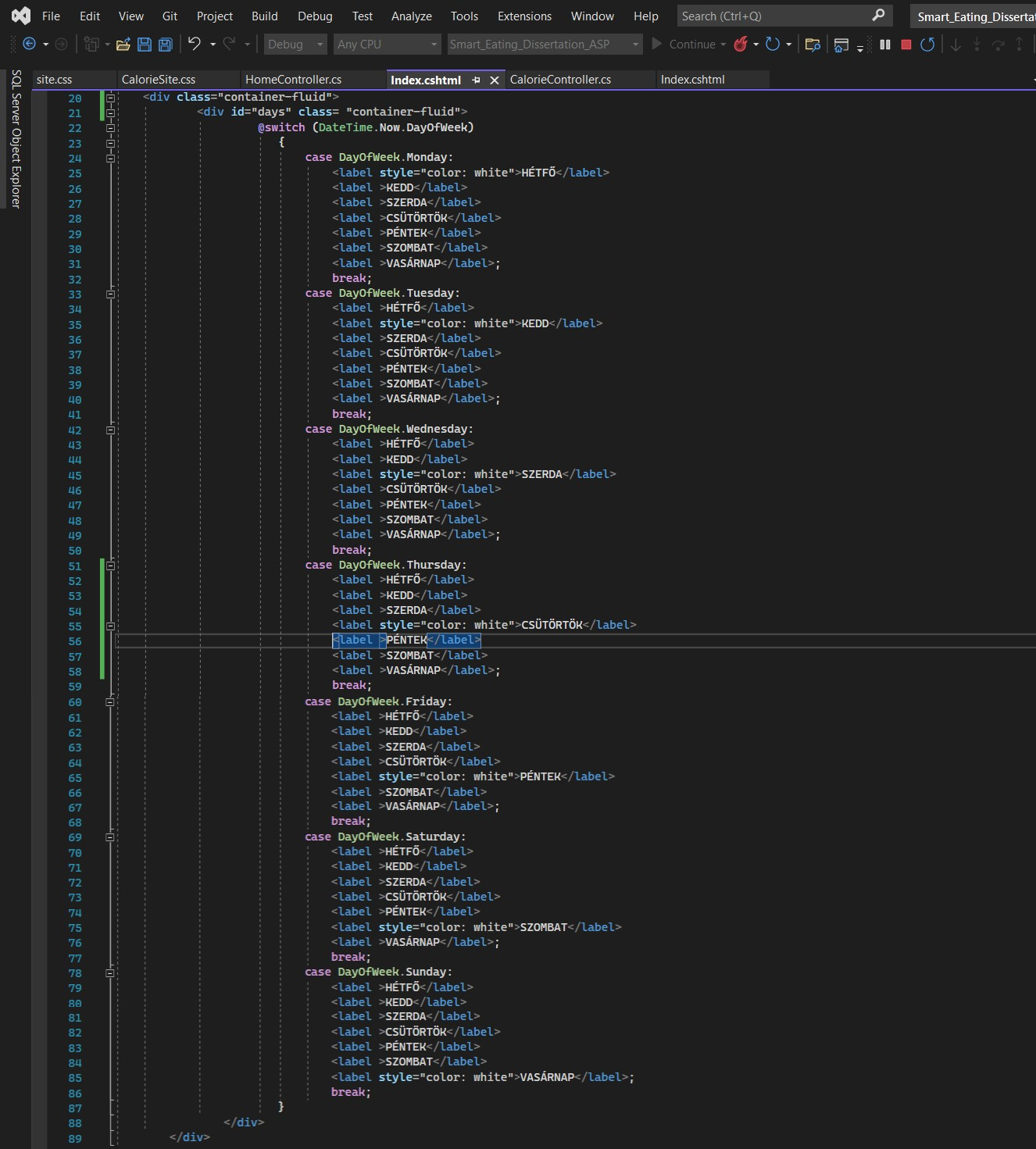
### Webalkalmazás

A webalkalmazásban létrehozott táblázat a kezdőlap betöltésekor üres, azonban az oldalt frissítve az alkalmazás minden étkezésre (Reggeli, Ebéd- ezen beül Leves és Főétel-, Vacsora) inicializál egy lehetőséget a megfelelő adatbázisból. Az alkalmazás megvizsgálja a megadott adatbázist, hogy van-e benne adat. Ha nincs, azt hibaüzenettel jelzi. Az adatbázisban, ha szerepel adat, akkor azt egy Count függvény segítségével soronként megszámolja és a véletlenszerű szám generálás maximum értékét az adatbázisban található adatsorok számával megegyezőre állítja be, amelyhez egyet hozzáadunk, hogy a sor utolsó eleme is biztosan kiválasztásra kerüljön. Ezt követően a Random osztályon belül létrehozott Random objektum segítségével a véletlenszerű számgenerálás során kapott számot az algoritmus a kiválasztott adatbázis azonosítójával (id) összehasonlítja és amennyiben megegyeznek, akkor az ahhoz az azonosítóhoz tartozó étel kerül kiírásra a táblázat megfelelő sorába. A lenti képen a reggeli étkezés randomolásának algoritmusa látható. Az alkalmazás a többi étkezés során is ugyanezt az algoritmust használja.



#### Hét napjai

Az alkalmazás az aktuális nap kijelölése során egy swicth -case utasításpár használata mellett a DateTime osztály, Now.DayOfWeek property -jének felhasználásával meghatározza az alkalmazás, futtatáskori aktuális napját és ezt követően a főoldalon megjelenített napok színét automatikusan változtatja. Fehér színnel az aktív (aktuális) napot, míg feketével az inaktív napokat jelöli.

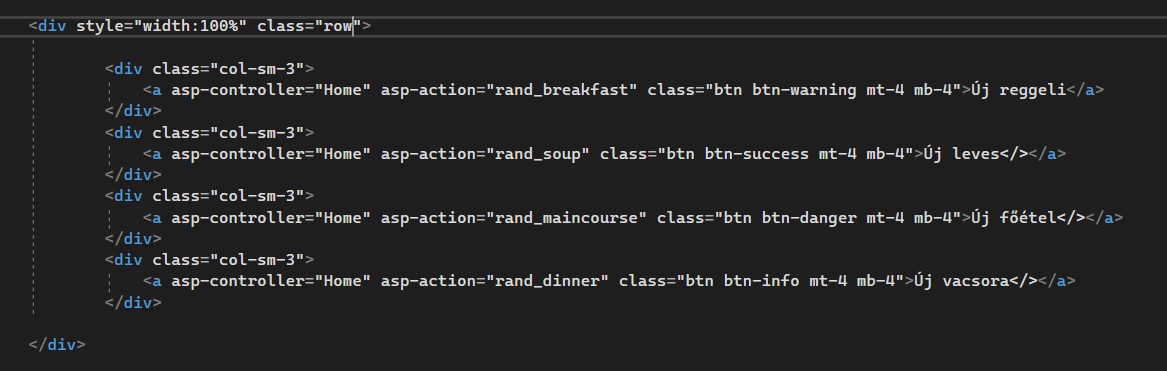


#### Új Reggeli -Új Leves -Új Főétel –Új Vacsora

Az ételek gombnyomásra történő véletlenszerű generálása hasonló módon működik, mint a betöltéskori randomolás, de ebben az esetben az adatbázis ellenőrzése, a randomolás, valamint az értékek változókba történő kiírása csak gombnyomás után fut le. Az algoritmus a HomeController vezérlőben az Új reggeli esetében az IActionResult rand\_breakfast() metódusban került meghatározásra.



Bármelyik étkezés gombra kattintás során a gomb a HomeController vezérlőben a rand\_breakfast() metódusra hivatkozva tudja, hogy melyik algoritmust vagy utasítást kell végrehajtania. Az Új reggeli gomb példájánál maradva a gomb a rand\_breakfast() metódusban meghatározottak miatt tudja, hogy a reggelik adatbázisából kell véletlenszerűen ételt generálni a főoldalon megjelenített táblázat meghatározott sorába.



## Tesztdokumentáció

* Az asztali alkalmazás Visual Studio 2022-es verzióval lett futtatva és tesztelve. Az alkalmazás az én tesztelésem alapján jól és hibamentesen működik.
* A webalkalmazás tesztelve lett Google Chrome (Verzió: 101.0.4951.54 (64 bites)) és Microsoft Edge (Verzió: 101.0.1210.32 (64 bites)) böngészőn. Az alkalmazás mindkét böngészőn hiba mentesen működik.

# 4. Összefoglalás

## 4.1. Szakdolgozat célja

A szakdolgozatunk célja, hogy segítsen azoknak a felhasználóknak, akik tudatosan és változatosan szeretnek táplálkozni. Az alkalmazás mindennapi használatával segítséget kaphatunk a változatos étkezéshez, valamint a Kalória kalkulátor kalkulációja által meghatározott értékek alapján láthatjuk, hogy mennyit fogyaszthatunk a kiválasztott ételekből, ezáltal támogatva a kitűzött célunkat (Fogyás, Tömegnövelés, Szinten tartás).

## 4.2. Megvalósítása

Az asztali alkalmazás, a webalkalmazás és az alkalmazásunkhoz kapcsolódó adatbázisok létrehozása is Visual Studio 2022 programmal készült. Az alkalmazás megtervezése után először az alkalmazások által használt adatbázisok létrehozása történt meg, amelyet Entity Framework Core keretrendszerrel készítettünk el. Az asztali alkalmazás WPF keretrendszerrel C# programozási nyelven, a webalkalmazás ASP.NET keretrendszerrel C# programozási nyelven készült. A weboldalak megjelenítése CSS stílusok használatával kerültek formázásra.

Az alkalmazás vizuális megjelenésének elkészülte után (FrontEnd) az alkalmazás által végrehajtandó algoritmusok, függvények (BackEnd) programozása következett.

## 4.3. Fejlesztési lehetőségek

* Regisztrációs felület létrehozása. Felhasználók regisztrálása, hogy a napi étrendek személyre szabhatók legyenek.
* Ételek hozzáadása a felhasználók által
* A webalkalmazásban a funkciók javítása
* Az alkalmazás elérésének bővítése (távoli hozzáférés)
* Mobilalkalmazás fejlesztése (digitális fejlődés miatt)

# 5. Irodalomjegyzék

* <https://getbootstrap.com/>
* <https://docs.microsoft.com/hu-hu/aspnet/core/>
* <https://infojegyzet.hu>
* <https://www.w3schools.com>
* <https://stackoverflow.com>