

Kodolányi János Egyetem

SZAKDOLGOZAT

ÓDOR MÁRK JÁNOS
ÜZEMMÉRNÖK-INFORMATIKUS
ALAPKÉPZÉSI SZAK

Budapest
2023.11.15.

**Kodolányi János Egyetem
Informatika Tanszék**

**Webfejlesztési folyamatok,
egy AI támogatott dietetikus alkalmazás számára**

*Web development processes for
an AI-supported dietetic application*

Konzulens: Dr. Pitlik László

**Készítette: ÓDOR MÁRK JÁNOS
ÜZEMMÉRNÖK-INFORMATIKUS
ALAPKÉPZÉSI SZAK**

**Budapest
2023.11.15.**

Tartalomjegyzék

RÖVIDÍTÉSEK JEGYZÉKE ANGOLUL	5
RÖVIDÍTÉSEK JEGYZÉKE MAGYARUL	6
ÁBRÁK JEGYZÉKE	7
BEVEZETÉS	9
1. Előszó	10
1.1. Kezdeti állapot	10
1.2. Motiváció	10
1.3. Célcsoport	11
1.4. Hasznosság	11
1.5 Rövid ismertetése a feladatoknak	12
2. Adatbázis	13
2.1. Kutatás	13
2.2. Adatbázis lekérdezés	15
2.3. Adatbázis feldolgozása Excelben	16
3. Szerver	19
3.1. Telepített programok	19
3.1.1. Programok rövid ismertetése	19
3.2. Szerver felállítása	20
3.2.1. VirtualBox telepítés	20
3.2.2. Debian telepítés	20
3.2.3. Apache2 Webszerver	21
3.3. PgAdmin4	22
3.3.1. Case sensitive	24
3.3.2. ASCII probléma	25
4. Kódolás	27
4.1. Kapcsolódás a szerverhez	27
4.2. Kódolási Kronológia	28
4.3. Skeleton részlet	29
4.4. Stíluselemek	31
4.4.1. Footer	31
4.4.2. Navbar	32

5. Titkosítás és Webcím	34
5.1. Tartománynév	34
5.2. Certificate.....	35
6. A weboldal tesztelés.....	38
7. ChatGPT.....	39
7.1. Támogatás	39
7.2. Ötlet szerzés	40
7.3 Kódsorok ellenőrzése és javítása	41
Tantárgyak kapcsolata a szakdolgozattal	42
ÖSSZEGZÉS.....	44
KIVONAT – HU	45
KIVONAT – EN	45
IRODALOMJEGYZÉK.....	46
HIVATKOZÁSJEGYZÉK.....	48

RÖVIDÍTÉSEK JEGYZÉKE ANGOLUL

- ODESSAI: Online dietetic expert system with AI
- AI: Artificial intelligence
- WWW: World Wide Web
- MIAÚ: Medium on Internet for Agrar / Applied Informatics in Hungary
- COCO: Component-based Object Comparison for Objectivity
- USDA: United States Department of Agriculture
- FDC: FoodData Central
- NDB: Nutrient Database
- SQL: Structured query language
- ASC: Ascending
- DESC: Descending
- VS: Visual Studio
- SSH: Secure Shell
- PHP: Hypertext Preprocessor
- TPL: Template
- ASCII: American Standard Code for Information Interchange
- HTML: Hypertext Markup Language
- SCP: Secure Copy Protocol
- IP: Internet Protocol
- JS: JavaScript
- AJAX: Asynchronous JavaScript and XML
- YAML: YAML Ain't Markup Language
- DNS: Domain Name System
- DDNS: Dynamic Domain Name System
- URL: Uniform Resource Locator
- DUC: Dynamic URL Service
- AWS: Amazon Web Services
- SSL: Secure Sockets Layer
- TLS: Transport Layer Security
- DV: Domain Validation
- OV: Organization Validation
- EV: Extended Validation
- CSR: Certificate Signing Request
- CA: Certificate Authority
- OLAP: Online Analytical Processing

RÖVIDÍTÉSEK JEGYZÉKE MAGYARUL

- ODESSAI: Online dietetikai szakértő rendszer mesterséges intelligenciával
- AI: Mesterséges intelligencia
- WWW: Világháló
- MIAÚ: Magyar Internetes Agrár / Alkalmazott Informatikai Újság
- COCO: Komponensalapú objektumösszehasonlítás az objektivitásért
- USDA: Az Egyesült Államok Mezőgazdasági Minisztériuma
- FDC: Élelmiszeradat Központ
- NDB: Tápanyag Adatbázis
- SQL: Strukturált lekérdezőnyelv
- ASC: Növekvő
- DESC: Csökkenő
- VS: Visual Studio
- SSH: Biztonságos Kapcsolat
- PHP: Hiperszöveg (hipertext) Előfeldolgozó
- TPL: Sablon
- ASCII: Amerikai Szabványos Információcsere Kódja
- HTML: Hiperszöveg (hipertext) Leíró Nyelv
- SCP: Biztonságos Másolási Protokoll
- IP: Internet Protokoll
- JS: JavaScript
- AJAX: JavaScript és XML aszinkron kommunikációja
- YAML: YAML Nem Jelölő Nyelv
- DNS: Tartománynévrendszer
- DDNS: Dinamikus Tartománynévrendszer
- URL: Egységes erőforrás azonosító (webcím)
- DUC: Dinamikus URL-szolgáltatás
- AWS: Amazon Web Szolgáltatás
- SSL: Biztonsági Aljzatréteg
- TLS: Szállítási Réteg Biztonsága
- DV: Tartomány Hitelesítés
- OV: Szervezeti Hitelesítés
- EV: Kiterjesztett Hitelesítés
- CSR: Tanúsítvány Aláírási Kérelem
- CA: Tanúsítvány Kibocsátó
- OLAP: Online Analitikai Feldolgozás

ÁBRÁK JEGYZÉKE

1. ábra: FoodData Central DB.....	13
2. ábra: Adatbázis ételcsoport	14
3. ábra: Keresztábra lekérdezés	15
4. ábra: Magvak lekérdezés	15
5. ábra: Névre szűrés	16
6. ábra: Párosítás	16
7. ábra: FKERES függvény	17
8. ábra: SORSZÁM függvény	17
9. ábra: VirtualBox telepítés.....	21
10. ábra: Magvak tábla	22
11. ábra: Oszlopadat típusok	23
12. ábra: pgAdmin oszlopnevek.....	24
13. ábra: ASCII codec error	25
14. ábra: Magyarosított táblázat	26
15. ábra: WinSCP belépés	27
16. ábra: Nyelvválasztás.....	29
17. ábra : menu2	30
18. ábra: Footer	31
19. ábra: Copyright.....	32
20. ábra: Navigációssáv.....	32
21. ábra: Navbar	32
22. ábra: Navbar menu2	33
23. ábra: Ikon.....	33
24. ábra: DDNS	34
25. ábra: Tartománynév.....	35
26. ábra: Nem privát kapcsolat.....	35
27. ábra: Útvonal	37
28. ábra: Sikertelen belépés.....	37
29. ábra: Tanúsítvány megtekintése	37
30. ábra: Webteszt	38
31. ábra: ChatGPT	39
32. ábra: ChatGPT támogatása.....	40

33. ábra: ChatGPT vonalkód.....	40
34. ábra: ChatGPT ellenőrzés.....	41

BEVEZETÉS

A következő fejezetben bemutatásra kerül a dolgozatom témája, a téma aktualitása, célcsoportok, hasznosság ismertetése, valamint az, hogy a mai világban erre miért is van szükség. A dolgozatban megemlített szoftver egy projekt keretein belül készült el, így lesznek utalások egy a projektben résztvevő szaktársam Molnár Csanád feladataira, munkáira. Ezt minden esetben hivatkozni fogom. Az alkalmazás köré épülő támogató feladatokat fogom bemutatni a következő fejezetekben.

Szeretném kifejezni hálámat mindazoknak, akik támogattak a szakdolgozatom elkészítésében, különösen Dr. Pitlik Lászlónak és Dr. Rikk Jánosnak.

1. Előszó

1.1. Kezdeti állapot

A szakdolgozat témájának az ötlete először 2023-ban látott napvilágot. Mindenképpen olyan tervvel szerettem volna előállni, ami, valamilyen formában dietetikához (A betegek helyes táplálkozásával foglalkozó tudományág) köthető, mivel számomra a családban is több ember szorul dietetikus segítségére (l. 1.2. *Motiváció*). Szaktársamnak Molnár Csanádnak is hasonló motivációs érvei voltak, így adta magát, hogy együtt, de külön részekre bontva dolgozzunk ki egy ilyen szoftvert egy projekt keretében. Az ODESSAI (online dietetic expert system with AI) csak és kizárólag ezen szakterületen lévő orvosoknak szántuk, viszont volt egy másik szempont is. Egy bárki által használható webalkalmazás, amihez plusz opciók társultak volna (l. 1.4 *Hasznosság*).

1.2. Motiváció

Mint fentebb említettem, a családomon belül van cukorbeteg és inzulin rezisztens személy is. Ezért számomra rendkívül fontos és személyes ügyként tekinthető a dolgozat, illetve a szoftver sikeressége. A mai világban rendkívül elterjedt a tudatos táplálkozás, a minél egészségesebb ételek utáni vágy ezzel szemben mégis egyre több az túlsúlyos ember és ezt sokszor (feldolgozott élelmiszer) a dezinformáció vagy a nem megfelelő információ szerzése, hiánya okozza. Olyan esetekben amikor egy embert megtévesztenek egy olyan felirattal, hogy szénhidrát csökkentett kenyér, amiből az átlagember egyből azt következteti, hogy egészséges, diétás, ettől a testzsírszázaléka csökkeni fog, holott sok esetben még több kalóriát is tartalmaz az élelmiszer, mivel ebben az esetben a szénhidrátot zsír formájában helyettesítették, így kedvező a cukorbetegnek, de kedvezőtlen azoknak, akik a testzsír-arányukon szeretnének javítani.

1.3. Célcsoport

A dolgozat célcsoportja minden olyan személy, akinek akár a legkisebb mértékben is dietetikai problémája van, vagy akárcsak elővigyázatosságból is szeretné tudni, hogy az adott táplálék, ami érdekli milyen általános tápanyagokat tartalmaz és mennyit. Illetve, össze tud hasonlítani, kettő vagy akár több terméket is különböző élelmiszer csoportokon belül. Továbbá hasznos lehet olyan orvosok/személyi edzők munkáságában, akik szeretnék a pácienseik/ügyfeleik számára felállítani a megfelelő étrendet, és pontos(összeszedett) információk alapján kidolgozni azt, amit az ODESSAI lehetővé tesz számukra.

1.4. Hasznosság

Az ODESSAI megkönnyíti és felgyorsítja az emberek információ szerzését különböző élelmiszerekről. Az ODESSAI egy könnyedén kezelhető weblapra: www.dieteticexpert.hu-ra van integrálva, ahol egy mindenki számára egyszerűen kezelhető és átlátható felületet talál. Az ODESSAI egy szakértőknek, dietetikai problémákkal küzdő személyeknek készült program, amely segíti a gyors és pontos információ átadását. Lehetőséget nyújt különböző élelmiszerek makró, illetve mikro tápanyag összetételének, ásványianyag tartalmának, vitamin bázisának megismerésére. A felhasználónak lehetősége van különböző szempontok alapján (pl. hiánybetegség, életmódbeli változások) kiválasztani és egy általunk meghatározott algoritmus alapján kiértékelni, mi lenne a megfelelő élelmiszer ezekre a szempontokra nézve.

Az ODESSAI alap koncepciója között szerepelt volna egy lehetőség arra, hogy a felhasználók saját betegségüket állíthassák be a rendszerben. Ezáltal a rendszer támogatta volna az adott betegségekhez, allergiákhoz kapcsolódó adatbázist és lekérdezési funkciókat, hogy személyre szabott tanácsokat és ajánlásokat kínáljon a felhasználóknak az egészségesebb életmód eléréséhez.

Ez a megközelítés lehetővé tette volna, hogy az emberek könnyen hozzáférjenek az egészséges táplálkozási tanácsokhoz és útmutatáshoz az egészségügyi állapotuknak megfelelően, miközben támogatást nyújtott volna azoknak, akiknek kevésbé van rálátásuk az ilyen jellegű információkra. A jelenlegi dolgozatban az egyszerűség kedvéért ezeket a formákat elvetettem,

így jelenleg nem tárolja a felhasználó adatait, egy egyszerű belépést követően bárki kedve szerint használhatja a weboldalt.

1.5. Rövid ismertetése a feladatoknak

Főbb feladatok nem tér ki minden egyes feladatrészre.

- A terv első felében szükség volt egy adatbázisra, ahol megtalálhatóak különböző élelmiszerek, egy meghatározott számú tápanyagforrással. Fontos volt, hogy az adatok hiteles forrásból származzanak.
- Meglévő adatokat szűrése, feldolgozása, feltöltésre való formára alakítás.
- Kétnyelviség témájával foglalkozni, angol szöveg magyarosítása a weblapon, adattáblában.
- Webszerver létrehozása, megvizsgálni, hogy a tovább haladás érdekében megfelelő-e.
- Adatbázis-kezelő rendszer kiválasztása, adatok végleges formájának feltöltése, a grafikus felhasználói felületre.
- Eddig keletkező problémák kezelése és javítása.
- Kódírás elkezdése, ütemterv szerint.
- Tartománynév lefoglalása, használatba léptetése.
- Weboldal titkosítása, SSL tanúsítvány révén.
- Kommunikáció COCO-val.
- Szoftver tesztelése a weboldalon.

2. Adatbázis

2.1. Kutatás

A megfelelő adatbázis kiválasztásánál az egyik legfontosabb szempont volt a hitelesség, a könnyű elérhetőség, és a folyamatos (évenkénti) adatbázis frissítése. Első körben magyar oldalakon próbálkoztam, ilyen volt a Kalóriabázis: <https://kaloriabazis.hu/>, ahol számunkra elegendő élelmiszer található, megfelelő mikro és makró tápanyag adatokkal. Első körben megpróbáltam írni egy e-mailt a weboldal Adatvédelmi tisztviselőjének, hogy lenne-e bármilyen lehetőség elkérni az adatbázisukat. Sajnos, de nem meglepő módon nem jártam sikerrel, mivel számukra maga az adatbázis volt a legnagyobb értéke a weboldalnak, így nem állt szándékukban ezt közölni velem. Ezután próbáltam olyan hivatalos forrásból származó adatbázist keríteni, ami ingyenes és/vagy a legtöbb esély van rá, hogy hozzá tudok jutni legális körülmények között. Így találtunk rá az USDA (United States Department of Agriculture – az Egyesült Államok Mezőgazdasági Minisztériuma): <https://www.usda.gov/> illetve FDC (FoodData Central – Élelmiszertár) weboldalra: <https://fdc.nal.usda.gov/>.

(1. ábra: FoodData Central DB, Forrás: <https://fdc.nal.usda.gov/fdc-app.html#/food-details/2514746/nutrients>)

Ez az oldal lehetővé tette, hogy szerezzek egy olyan adatbázist, amely rengeteg, közel 9000 élelmiszer adatát tartalmazza, mint például makró, mikro, ásványianyag, és különböző összetételek információját. Nagy segítség volt, hogy az adatbázisban részletezve volt minden, külön táblázatban voltak az élelmiszer csoportok, attribútumok (tápanyag) a hozzá tartozó mértékegységekkel.

FdGrp_CD	FdGrp_Desc
0100	Dairy and Egg Products
0200	Spices and Herbs
0300	Baby Foods
0400	Fats and Oils
0500	Poultry Products
0600	Soups, Sauces, and Gravies
0700	Sausages and Luncheon Meats
0800	Breakfast Cereals
0900	Fruits and Fruit Juices
1000	Pork Products
1100	Vegetables and Vegetable Products
1200	Nut and Seed Products
1300	Beef Products
1400	Beverages
1500	Finfish and Shellfish Products
1600	Legumes and Legume Products
1700	Lamb, Veal, and Game Products
1800	Baked Products
1900	Sweets
2000	Cereal Grains and Pasta
2100	Fast Foods
2200	Meals, Entrees, and Side Dishes
2500	Snacks
3500	American Indian/Alaska Native Foods
3600	Restaurant Foods

(2. ábra: Adatbázis ételcsoport, Forrás: fooddb.accdb)

A FDC weboldalon lehetősége van az embernek az élelmiszerek könnyű megtalálására és összehasonlítására, azok tápanyagtartalmának alapján.

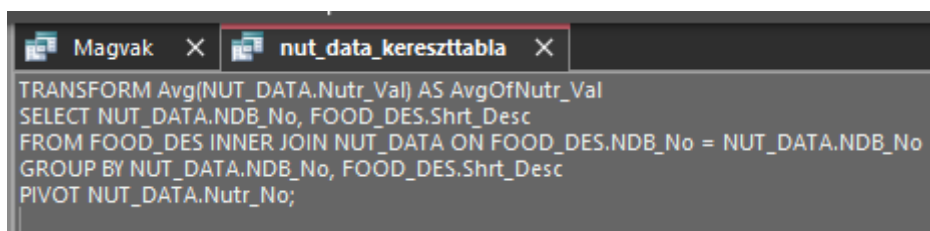
Az NDB Number (Nutrient Database Number vagyis Tápanyag-Adatbázis Szám) az élelmiszerek azonosítására szolgáló egyedi azonosító a FoodData Central adatbázisában. Az NDB szám segítségével könnyen és egyértelműen azonosíthatók az egyes élelmiszerek a táplálkozási adatbázisban.

Ez az azonosító kulcsfontosságú az élelmiszerek tápanyagtartalmának, összetételének és más részletes információinak lekérdezésében. Az NDB szám alapján lehetőség van az adott élelmiszer tápanyagtartalmának megnézésére, összehasonlítására más élelmiszerekkel, valamint az élelmiszerek táplálkozási jellemzőinek és tulajdonságainak megtekintésére.

2.2. Adatbázis lekérdezés

További lépésként szükségünk van egy kereszttáblás lekérdezésre, ami összeszedi egy táblába az NDB számot, az élelmiszer nevét, és az élelmiszerhez tartozó összetételt. Valamint

felcseréljük a Nutr_No oszlopot, sorra (fejléc) a PILOT SQL művelet segítségével. Ezt a következő SQL kóddal érhetjük el:

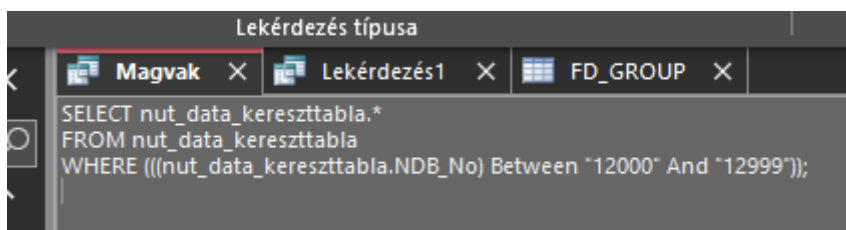


```
TRANSFORM Avg(NUT_DATA.Nutr_Val) AS AvgOfNutr_Val
SELECT NUT_DATA.NDB_No, FOOD_DES.Shrt_Desc
FROM FOOD_DES INNER JOIN NUT_DATA ON FOOD_DES.NDB_No = NUT_DATA.NDB_No
GROUP BY NUT_DATA.NDB_No, FOOD_DES.Shrt_Desc
PIVOT NUT_DATA.Nutr_No;
```

(3. ábra: Keresztábra lekérdezés, Forrás: nut_data_kereszttabla, fooddb.accdb)

Ez a táblázat tartalmaz minden számunkra fontos elemet kivéve azt a táblát, ami leírja, hogy az adott szám (Nutr_No) milyen tápanyaghoz köthető (l. 2.3. *Adatbázisból Excel fájl*).

Továbbá szükségünk van egy olyan lekérdezésre is, ami külön kigyűjti nekünk az adott élelmiszer csoportot, pl. Magvakat. Ezt könnyen megtehetjük, mivel az adatbázis már magában csoportosította az élelmiszereket, (l. 2. ábra) így egy egyszerű lekérdezéssel kigyűjthetjük ez esetben az olyan élelmiszereket, mint a magvak.



```
Lekérdezés típusa
Magvak Lekérdezés1 FD_GROUP
SELECT nut_data_kereszttabla.*
FROM nut_data_kereszttabla
WHERE (((nut_data_kereszttabla.NDB_No) Between "12000" And "12999"));
```

(4. ábra: Magvak lekérdezés, Forrás: Magvak, fooddb.accdb)

Lehetőség van arra is, hogy pl. a tejtermékek közül csak a sajtokat szűrjük ki. Ezt úgy tudjuk létrehozni, hogy név alapján szűrünk, ezt megtehetjük lekérdezés által, vagy egy szimpla szűréssel is:

Shrt_Desc	203	204	205	207	208
TER, WITH SALT	0,85	81,11	0,06	2,11	7,1
TER, WHIPPED, W/ SALT	0,49	78,3	2,87	1,62	7,1
TER OIL, ANHYDROUS	0,28	99,48	0	0	8,1
ESE, BLUE	21,4	28,74	2,34	5,11	3,1
ESE, BRICK	23,24	29,68	2,79	3,18	3,1
ESE, BRIE	20,75	27,68	0,45	2,7	3,1
ESE, CAMEMBERT	19,8	24,26	0,46	3,68	3,1
ESE, CARAWAY	25,18	29,2	3,06	3,28	3,1
ESE, CHEDDAR	22,87	33,31	3,09	3,71	4,1
ESE, CHESHIRE	23,37	30,6	4,78	3,6	3,1
ESE, COLBY	23,76	32,11	2,57	3,36	3,1
ESE, COTTAGE, CRMD, LRG OR SML CURD	11,12	4,3	3,38	1,41	1,1
ESE, COTTAGE, CRMD, W/ FRUIT	10,69				1,1
ESE, COTTAGE, NONFAT, UNCRMD, DRY, LRG OR SML CURD	10,34				1,1
ESE, COTTAGE, LOWFAT, 2% MILKFAT	10,45				1,1
ESE, COTTAGE, LOWFAT, 1% MILKFAT	12,39				1,1
ESE, CREAM	6,15	34,44	3,52	1,27	3,1
ESE, EDAM	24,99	27,8	1,43	4,22	3,1
FFSF, FETA	14,21	21,28	4,09	5,2	2,1

Egyéni szűrő ? X

Shrt_Desc tartalmazza

OK Mégse

(5. ábra: Névre szűrés, Forrás: fooddb.acddb)

2.3. Adatbázis feldolgozása Excelben

Ahhoz, hogy számunkra használható adatok legyenek PostgreSQL adatbázishoz való feltöltéshez, előbb Excelben kicsit átkell alakítani a táblázatot, amit feltöltünk. Első lépésben az oszlopneveket megfelelő párral kell azonosítani pl. 203 == Protein. Ezt az adatbázisban található definíciós táblázattal tudjuk elérni.

	Nutr_No	Units	Tagname	NutrDesc	Num_Dec	SR_Order
+	203	g	PROCNT	Protein	2	600
+	204	g	FAT	Total lipid (fat)	2	800
+	205	g	CHOCDF	Carbohydrate, by difference	2	1100
+	207	g	ASH	Ash	2	1000
+	208	kcal	ENERC_KCAL	Energy	0	300

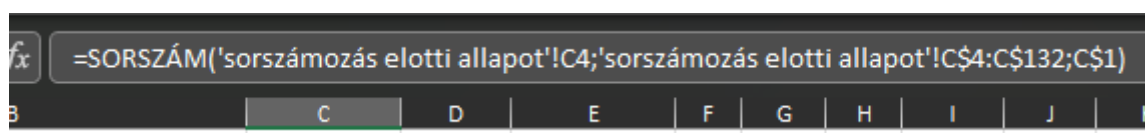
(6. ábra: Párosítás, Forrás: fooddb.acddb)

Ezt többféle módon is meg lehet oldani, én az egyszerűség fényében egy FKERES függvényt alkalmaztam, ez azért is célszerű, mivel külön munkalapon az adatbázisból kinyert táblázatban jelen volt minden tápérték párban a számával (Nutr_No):

(7. ábra: FKERES függvény, Forrás: Magvak_vegleges.xlsx)

Továbbá észrevettem, hogy vannak bizonyos élelmiszerek, amiknél a 0 érték nem feltétlenül azt jelezte, hogy nem tartalmaz az adott tápanyagból, hanem azt, hogy hiányos az adat. Ezt két módon küszöböltem ki. Az adott élelmiszer csak 1 tápanyagnál hiányos (esetek kevesebb mint 1%-a) akkor manuálisan kipótoljuk az adott értéket, ezt természetesen nem véletlenszerűen tesszük, hanem több, az ehhez hasonló weboldalon kikeressük és összevetjük az adott élelmiszer tápanyagát. A másik, ami kevésbé kedvező módszer, hogy egyszerűen kitöröljük az adott élelmiszert. Ez csak és kizárólag azért történik meg ez esetben, mert idő szűkében jelenleg nem áll módunkban minden olyan rekordot kijavítani, ami esetleg hibás lehet.

Szükségünk van továbbá még két táblára, ami azt fogja megmutatni, hogy az adott oszlop pl. Protein, sorai (élelmiszerek) az oszlopban hányadikok a rangsorban. (vö. Molnár Csanád dolgozatában, a 10. fejezetben)



(8. ábra: SORSZÁM függvény, Forrás: Magvak_vegleges.xlsx)

Ezután összegyűrjük az egészet egyetlen táblázatba, ami a következőket tartalmazza: Sorszám, egyedi azonosító szám (NDB), teljes név, tápanyag, rangsorolt tápanyag kétszer (ASC, DESC).

Valamint szükségünk van egy beállításra az Excelben mielőtt ki exportáljuk a táblának .CSV fájlba, meg kell változtatni az értékeket, hogy a tizedes elválasztó vessző helyett, pont legyen. Erre azért van szükség mert a Pgadmin4 ezt fogja tudni kezelni a vesszős változatot nem. Ezt az Excel beállítások→Speciális→Tizedesjel fülön érhetjük el. Majd ezután ki exportáljuk .CSV fájlba pontosvesszővel tagolva, a fejlécek megtartásával.

3. Szerver felépítése

3.1. Telepített programok

A szakdolgozat során több elengedhetetlen szoftver is a segítségünkre volt, ezek a következők:

- Visual Studio Code: <https://code.visualstudio.com/download>
- Microsoft 365(Office): <https://www.microsoft.com/hu-hu/microsoft-365/get-office-and-microsoft-365-oem-download-page>
- WinSCP: <https://winscp.net/eng/download.php>
- Putty: <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- Puttygen: <https://www.puttygen.com/>
- PostgreSQL: <https://www.postgresql.org/>
- VS Code extensions: PHP, Smarty Template, Composer, TPL Reader

3.1.1. Programok rövid ismertetése

- A Visual Studio Code: Egy számomra rugalmas fejlesztői eszköz, könnyű testre szabni és széles körben támogatja a programozási nyelveket.
- Microsoft 365: világszerte sokak által használt szoftver, Word, Excel, Access, PowerPoint, gyakran használt programok, sokak által használtsága miatt, könnyű szinte bármilyen segítséget megtalálni hozzá online.
- WinSCP: Ez a szoftver egy ingyenesen elérhető, nyílt forráskódú program, ami lehetővé teszi a biztonságos fájlküldést és -fogadást távoli szerverek között. Számomra ez a program egyszerre kínálta a könnyen kezelhetőséget és a sokoldalúságát.
- Putty: Szintén egy ingyenes, nyílt forráskódú SSH és telnet kliens, ami lehetővé teszi a kapcsolódást a szerverhez, könnyen használható, egyszerű felhasználói felület. Már többször használtam ezt a programot, tanulásom, munkám során, így adta magát, hogy ez mellett döntsek.
- Puttygen: Puttyhoz kapcsolódó eszköz, lehetővé teszi az SSH kulcsok létrehozását, kezelését.
- VS Code kiterjesztések: Olyan modulok vagy pluginek, amiket a VS Code-hoz hozzáadva segíti azoknak működését, ezáltal hozzájárulva a munkánkhoz.

3.2. Szerver felállítása

Az alap ötlet az volt, hogy egy VirtualBoxon: <https://www.virtualbox.org/wiki/Downloads>, fogok létrehozni egy Apache2 webszervert. Ez annyiból tűnt logikus döntésnek mivel, tanulmányaink során már használtuk, amikor Dr. Rikk János tanárral, kísérleteztünk és teszteltünk egy Raspberry Pi szimulációt VirtualBox keretein belül, ennek nagy előnye az volt, hogy ne kelljen fizikai eszközt vásárolni, hanem egy virtuális környezetben van lehetőségünk szimulálni. Ezáltal a választásom eleinte, egy VirtualBoxon futó Debian rendszer volt, ami egy ingyenes és nyílt forráskódú operációs rendszer, amely az egyik legnépszerűbb Linux disztribúciók közé tartozik.

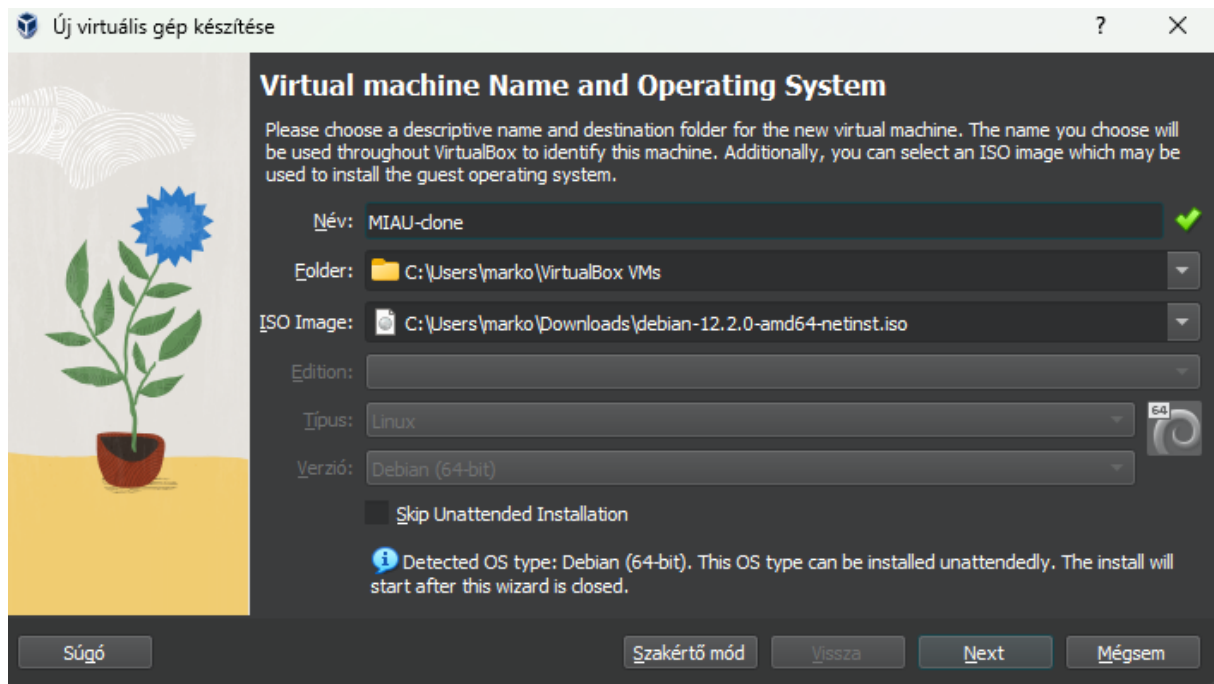
3.2.1. VirtualBox telepítése

- Meggyőződni, hogy a processzor támogatja a virtualizációt, és aktiválva van a BIOS/UEFI, ha szükséges.
- VirtualBox letöltése: <https://www.virtualbox.org/>
- VirtualBox installáció utasításainak követése

3.2.2. Debian telepítése

A Debian 5.10.179-2 számú verzió került telepítésre, lépései:

- Debian ISO letöltése: <https://www.debian.org/download>
- Virtuális gép létrehozása a Debian számára:
 - Névadás, típus és verzió kiválasztása
 - Memória méret beállítása, saját igény szerint
 - Tárhely lefoglalása
- Installáció követése
- Installáció befejezése, virtuális gép újraindítása



(9. ábra: VirtualBox telepítés, Forrás: saját készítés)

3.2.3. Apache2 Webszerver

Ezután következett, hogy a már meglévő operációs rendszerre egy Apache2 webszervert telepítünk, ennek a lépései a következők voltak:

- Megnyitjuk a terminált és a következő kódsorokat leírjuk:
 - **sudo apt update** (csomaglista frissítése)
 - **sudo apt install apache2**(telepíti az apache2-t)
 - **sudo systemctl start apache2**(webszerver futtatása a virtuális gépen)

Itt azért nem említem a további lépéseket mivel, habár alap eleme volt a dolgozatnak, hasonló, de nem ebben az irányban folytattam tovább a munkát. Sajnos több probléma is felmerült már az elején, pl. a későbbiekben megemlítésre kerülő COCO: (<https://miau.my-x.hu/myx-free/coco/index.html>) adaptálása, a nem megfelelő verziószám miatt nehézkessé vált, de úgy éreztem szükséges, ha nem is a legújabb, de viszonylag újabb verziókkal dolgozni, illetve elég nagy problémát jelentett az is, hogy ahhoz, hogy elérjük a szervert, a VirtualBoxnak 24/7 futnia kellett ami a saját számítógépemen volt így minimum az áram már költséges lett volna, illetve bármi probléma adódhat fizikailag, ami miatt nem tudunk dolgozni a szerveren vagy csak szimplán nem elérhető.

Ezért úgy döntöttünk, hogy egy Amazon által biztosított szerveren fog tovább futni, amely biztonságos és folytonosan elérhető. Ezen fejezet megtekinthető Molnár Csanád dolgozatában a 6.1. alfejezetnél.

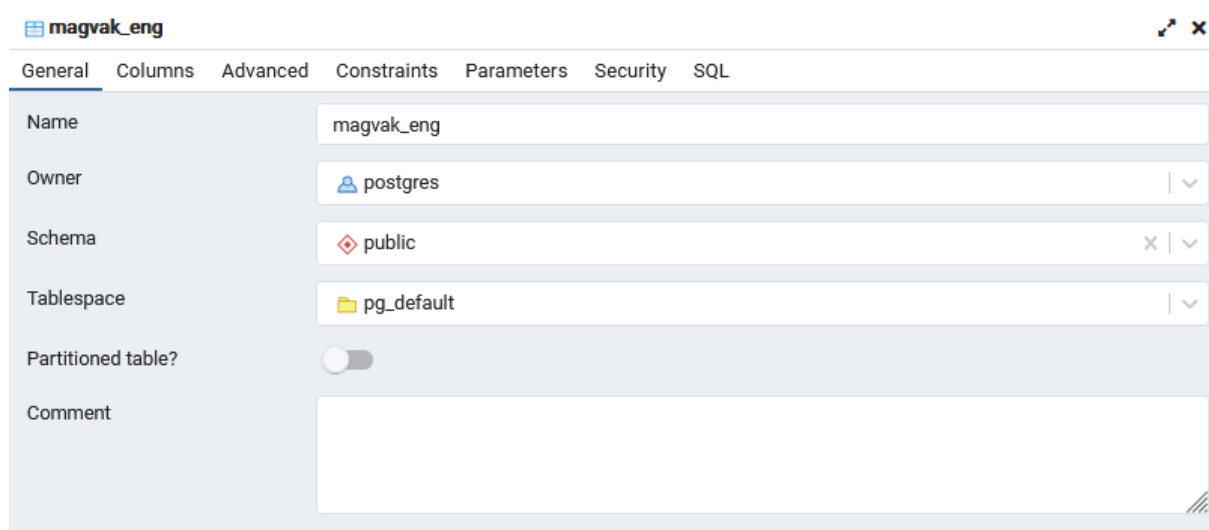
3.3. PgAdmin4

A pgAdmin a világ legfejlettebb nyílt forráskódú adatbázisának, a PostgreSQL-nek a menedzsment eszköze (grafikus felhasználói felülete). PostgreSQL adatbázisok kezelésére és adminisztrációjára tervezték. Számomra azért esett erre a választás, mert egyszerűen és hatékonyan képes kezelni az adatbázisokat, gyorsan futtat lekérdezéseket, képes táblák létrehozása után adatokat importálni/exportálni. Mivel PostgreSQL adatbázis alapján történnek a meghívások, ezért a pgAdmin mellett tettem le a voksomat.

(A PostgreSQL-ről további információ Molnár Csanád dolgozatában található, a 6.2.3. alfejezetnél)

Röviden, a pgAdmin azért volt szükséges, hogy az Excelből importált táblákat tudjuk adminisztrálni.

Ahhoz, hogy tudjak majd dolgozni az Excelből ismert táblázattal, először létre kell hoznunk pgAdmin-on belül, egy olyan táblázatot, amelyekben részletesen szerepelnek azok a fejlécek, amelyek az Excelben lévő táblázatban is szerepelnek, mint fejléc.



(10. ábra: Magvak tábla, Forrás: magvak_eng tábla, pgAdmin)

Nagyon fontos figyelni arra, hogy pontosan adjunk meg mindent ugyanis importálás után, a legtöbb esetben már nem áll mód arra, hogy ezen változtassunk. Többnyire 3 féle adat típust használtam a tábla megalkotása során ezek a következők:

- bigint (nagy egész számok tárolására szolgál)
- character varying (változó hosszúságú karakterlánc tárolására szolgál)
- real (lebegőpontos számok tárolására szolgál)

bigint: Itt megjegyezmem, hogy jelenleg elég lett volna smallint/integer adattípust használnunk, de a bigintel ki lehetett küszöbölni, ha egyszer bármilyen oknál fogva túlsordulnának a számok.

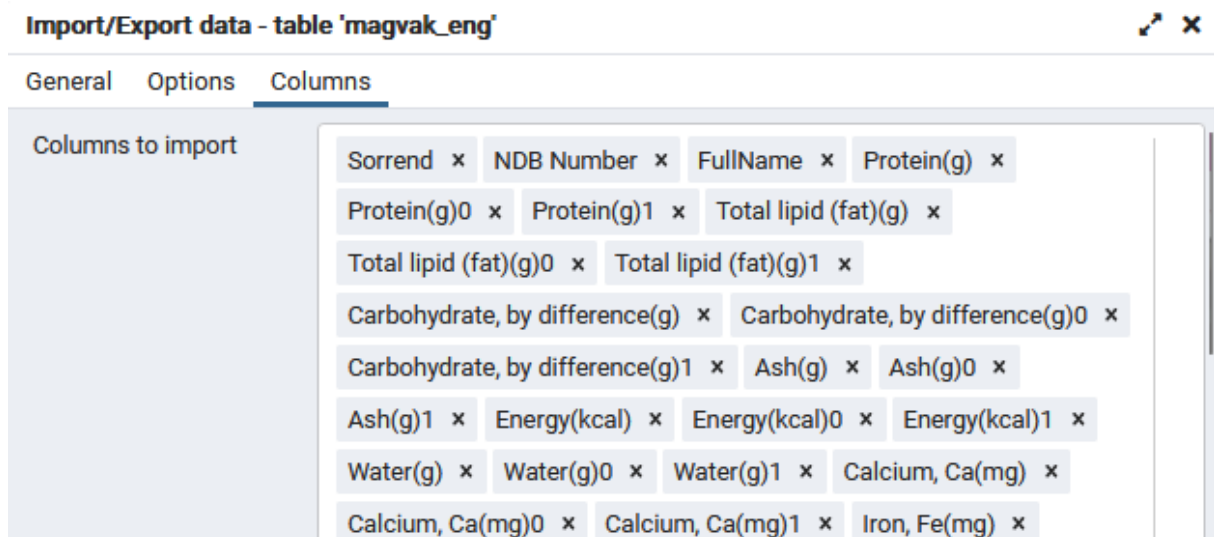
character varying: Azért használtam ezt a fajta adattípust, mivel a sima „char” adattípusnál ez egy sokkal rugalmasabb változat, a benne tárolt adatok hossza szerint változhat a tárolási területe.

Primary key: Sorrend oszlop (minden élelmiszernek egyedi sorszámozása van) itt megtehettem volna, hogy az NDB Number egyediségére hivatkozva állítom be az elsődleges kulcsot, de a hibák elkerülése végett egy külön sorrend oszlopot állítottam elő, amit kijelöltem erre a feladatra.

	Name	Data type	Length/Precisi...	Scale	Not NUL...	Primary key?	Default
	Sorrend	bigint			☒	☒	
	NDB Number	character varying			☐	☐	
	FullName	character varying			☐	☐	
	Protein(g)	real			☐	☐	

(11. ábra: Oszlopadat típusok, Forrás: magvak_eng tábla, pgAdmin)

Miután létrehoztuk a táblát és megbizonyosodtunk róla, hogy minden oszlop név és adattípus pontos, elkezdődhet az Excelből történő importálás. Ehhez az előzőleg kimentett .CSV fájlt felkell tölteni a pgAdmin-ra, majd az importálás menüponton belül, beállítani az elválasztó karakter, ami az én esetemben a pontosvessző volt (;).



(12. ábra: pgAdmin oszlopnevek, Forrás: magvak_eng tábla, pgAdmin)

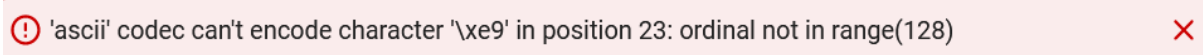
3.3.1. Case sensitive

Feltöltés során kettő problémába ütköztem. Az egyik ilyen volt a kis- és nagybetű érzékenysége. Ezért alapesetben minden táblázat nevet kis betűvel írtam, mivel többször is előfordult kódolás során, hogy emiatt rosszul vagy nem sikerült betölteni adott pontokon a táblázatot.

Az adatbázis érzékenysége a PostgreSQL adatbázisbeállításaitól és konfigurációjától függ. Alapértelmezés szerint a PostgreSQL case sensitive, de a konfigurációk változtathatják ezt a viselkedést az adatbázisban. A PostgreSQL támogatja a különféle konfigurációs beállításokat, amelyek befolyásolhatják az adatbázis érzékenységét a kis- és nagybetűkkel szemben, például a LOWER_CASE vagy UPPER_CASE paraméterek segítségével.

3.3.2. ASCII probléma

A másik nagyobb probléma, az ékezetek/különleges karakterek megjelenítése volt. A szakdolgozat része egy a weboldalon is elérhető magyar verzió is (vö. Molnár Csanád dolgozatában a 8.5. alfejezetben), ezért szükségünk volt egy olyan táblára, amelyekben az angol szövegrészeket magyar verzióra cseréljük fel.



(13. ábra: ASCII codec error, Forrás: pgAdmin)

Több ilyen hiba is volt ékezet és karakter formájában, a `\xe9` és a `\xb5` egy Unicode karakter kódját jelöli, ami nem értelmezhető ASCII kódolásban. Az ilyen hibákat a szerveren található log fájlból lehet elérni, amelynek az elérési útvonala: `/var/log/pgadmin/`

UnicodeEncodeError: 'ascii' codec can't encode character '\xb5' in position 926: ordinal not in range (128)

2023-11-12 21:47:46,589: ERROR pgadmin: 'ascii' codec can't encode character '\xb5' in position 926: ordinal not in range (128)

Itt lett volna lehetőség egy olyan opcióra, amit html entitásnak hívnak.

A HTML entítások olyan speciális karaktereket vagy szimbólumokat jelölnek meg, amelyeket a HTML kódban nem lehet közvetlenül megjeleníteni vagy használni, mert ezek részei az HTML kód struktúrájának vagy a jelölésnek. A `µ` HTML entitás például a mikrogramm karaktert ábrázolja.

pl. Valamilyen élelmiszer szelén tartalma: 10`µ`g 10 μ g

Maradtam az egyszerű ékezet elhagyás és a speciális karakterek alternatív jelölésénél, mivel ezzel is megszerettem volna spórolni az esetleges hibákat, természetesen, mint minden, ez a rész is fejleszthető, így idővel lehet, hogy áttérek a fent említett módszerre.

A dolgozat része lett volna még, hogy a magyar verziónál nem csak a fejléceket „magyarosítjuk”, hanem az élelmiszerek neveit is ezt azonban a jelen verzió még nem tartalmazza.

Mivel az ASCII kódtábla nem használ ékezeteket így, kicseréltem az angol szavakat magyar szóra, de ékezetek nélkül. A különleges karakterekkel is gond volt, pl. a mikrogramm jelölésével (µg). Ezesetben mivel az adott környezet nem támogatta, alternatív módon jelöltem a mikrogrammot: Mcg.

Az Mcg egy hivatalos rövidítés, amely a mikrogrammot jelöli.

Cink, Zn(mg)0 real	Cink, Zn(mg)1 real	Rez, Cu(mg) real	Rez, Cu(mg)0 real	Rez, Cu(mg)1 real	Szelen, Se(Mcg) real	Szelen, Se(Mcg)0 real	Szelen, Se(Mcg)1 real	A Vitamin(IU) real
52	49	0.04	17	82	14.5	39	32	721
53	47	0.024	82	17	14.5	39	32	1080
62	38	0.019	94	5	14.5	39	32	592
62	38	0.021	87	9	14.5	39	32	820
43	57	0.024	82	17	14.5	39	32	1054
16	85	0.03	51	43	28.5	4	97	1242
49	52	0.042	15	85	14.5	39	32	985

(14. ábra: Magyarosított táblázat, Forrás: sajtok_hun tábla, pgAdmin)

Még szeretném megemlíteni, hogy az élelmiszerek mértékegységeit, az egyszerűség kedvéért nem külön oszlopban tároltam, hanem már a feltöltési szakasz előtt az oszlopfejléc nevébe integráltam, zárójelben.

Ezután, ha végig ellenőriztünk mindent elkezdődhet a kódolás folyamata, amit a következő fejezetekben végig lehet kísérni. Fontos megemlítenem, hogy mivel kooperációban, külön részekre bontva történt a kódírás ezért lesznek részek, amiket csak Molnár Csanád dolgozatában tudnak megtekinteni, de ezeket jelezni fogom minden esetben.

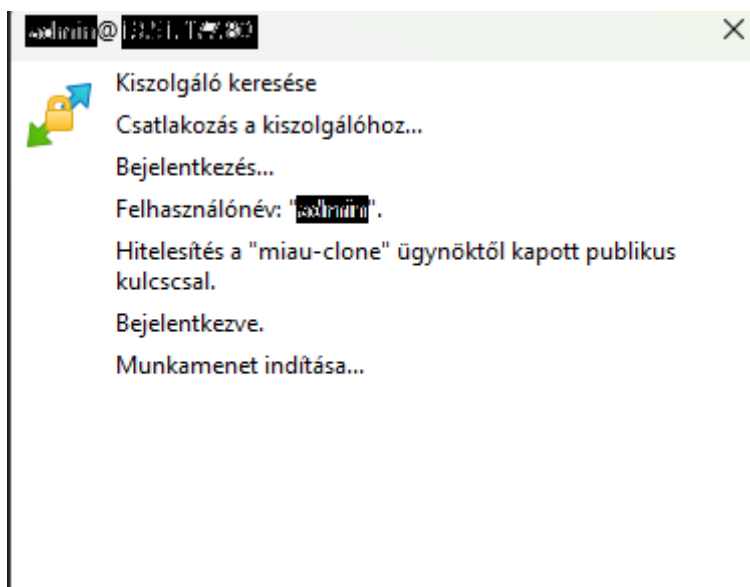
4. Kódolás

4.1. Kapcsolódás a szerverhez

Ahhoz, hogy a saját számítógépünkről feltudjunk csatlakozni a szerverhez a fent említett WinSCP programot fogjuk alkalmazni, ehhez először is meg kell nyitni egy Puttygen által készített privát kulcsot amire azért van szükség, hogy illetéktelen személyek ne tudjanak hozzáférni a rendszerhez távoli gépről. Ezután szükségünk lesz pár program béli beállításra, amelyek a következők:

- Nyissuk meg a programot, Haladó menü pontban SCP/Héj fülön a Parancsértelmező ablakban válasszuk ki a `sudo su` – parancsot, erre azért van szükség, hogy magasabb jogosultságú felhasználóvá váljunk a rendszerben, így teljes hozzáférést kapunk a rendszerhez.
- Válasszuk ki az SCP Fájlfülkölt, a magasabb biztonság érdekében
- Írjuk be az IP-címet vagy Tartománynevet és a port számát
- Írjuk be a felhasználónevet és jelszót

Ezt követően a program befog minket léptetni és kezdődhet a munka.



(15. ábra: WinSCP belépés, Forrás: saját készítés)

4.2. Kódolási kronológia

Fontos megemlítenem itt is, hogy a kódolás összes részletét, az enyém, illetve a szaktársam Molnár Csanád szakdolgozatában tekinthető meg, mivel a teljes egészet külön-külön részekre bontottuk így én csak megemlítem, illetve pár kódrészlet bemutatására törekedem. A következő pontokat részletesen, Molnár Csanád dolgozatában a 8. Fejezetnél lehet megtekinteni.

- Request Handler: bejövő kommunikációk kezelése, Ha a szerver „POST” kérést kap, ellenőrzi, hogy a beírt paraméter elő van-e konfigurálva. A "\$method" változó tárolja a kérés módját, és ha létezik a kívánt paraméter, akkor kezeli az adatokat és illeszti azokat a megfelelő „tpl” fájlokba.
- Smarty template engine: A Smarty egy template engine, amely segít a PHP-ban a megjelenítési logika és az adatok szétválasztásában. A „.tpl” kiterjesztésű fájlokban megadott vázák (skeleton) könnyen testre szabhatóak és könnyen meghívhatóak az URL cím alapján.
- Skeleton: ODESSAI projekt alapja, lehetővé teszi a könnyebb változtatások implementálását, új funkciók egyszerű hozzáadását. Az „index.php” és az „index.tpl” fájlok közötti koordináció. (l. 4.3. *Skeleton részlet*)
- Stíluselemek: (l. 4.4 *Stíluselemek*)
- Trans Handler: Az alkalmazás két nyelven történő használatát egy transhandler.inc.php nevű PHP fájl kezeli, ami tárolja a szavak magyar-angol megfelelőit YAML formátumban. A nyelv kiválasztása alapján tölti be a megfelelő változókat a YAML fájlokból, és ha nincs megfelelő nyelvi mappa, létrehozza azt.
- AJAX/JS: Az „/OLAP” menüpontok kiválasztása AJAX segítségével történik, így az élelmiszer-termék választása esetén azonnal frissülnek és jelennek meg az információk.

4.3. Skeleton részlet

```
switch ($route)
{
    //switching language will get you the preferred language
    case 'language':
        if ($method == 'get' && $request->hasParams())
            if (preg_match('/^[a-z]{2}$/', $request->getParameter(0)))
                if ($trans->isLangExists($request->getParameter(0)))
                    $_SESSION['lang'] = $request->getParameter(0);
        header("Location: /"); // újratöltjük az oldalt az új nyelvvel
        break;
```

(16. ábra: nyelvválasztás, Forrás: /var/www/html/index.php)

Ez a kódrészlet egy switch-case szerkezetet tartalmaz, amely a bejövő kérések alapján kezeli a nyelvválasztást a webalkalmazásban.

- Az első rész a „language” útvonalra irányul, ami a nyelvváltásért felelős.
- Ha a kérés „GET” módban érkezik és van paramétere, amely a nyelvet jelöli (két karakter hosszú kisbetűs angol karakterek), akkor megnézi, hogy az adott nyelv támogatott-e.
- Ha a megadott nyelv támogatott, akkor az \$_SESSION tömbbe menti ezt a nyelvet, amely azt jelzi, hogy a felhasználó preferált nyelvét használja.
- Ezután az oldalt átirányítja a főoldalra az új nyelvvel, a „Location” fejléccel, amely megújítja az oldalt a választott nyelven.

Összességében ez a kód arra szolgál, hogy a felhasználó kérése alapján lehetővé tegye a nyelvváltást, és frissítse az oldalt az új nyelvvel.

```
case 'menu2':
    if ($method == 'get')
    {
        //upon selecting "menu2", which is renamed in navigation.tpl.
        also includes a file based on the selection's value, where here it would be
        "menu2" and it would include olap2.tpl in index.tpl
        $smarty->assign('request', $route);
        $conn_str = "host=".$config['DBHostName']."
dbname=".$config['DBName']." user=".$config['DBUserName']."
password=".$config['DBPassword'];
        //connecting to postgresql and querystring building
        $dbconn = pg_connect($conn_str) or die('Could not connect: ' .
pg_last_error());
        $query= "SELECT * FROM public.usdae";
```

```

        $result=pg_query($query) or die('Query failed: '
.pg_last_error());
        $resultArr = pg_fetch_all($result,PGSQL_BOTH);
        //fetching all data from the database based on the query
        if (count($resultArr))
        { //if it has result (so not empty), assign the resulted
array for smarty to use as $resultArr, also set a variable to count the number
of columns there are, also assign it to smarty by "columnNo"
            $hasResult = true;
            $smarty->assign("resultArr",$resultArr);
            //$trimmed_array =str_replace($resultArr,";", "");
            $columnNo = count($resultArr[0]);
        }
        $smarty->assign("columnNo",$columnNo);
        $smarty->assign('request', $route);
    }
    else
        header("Location: /"); // ha nem get metódus akkor kezdőoldal
//if not get method, then homepage
    break;

```

(17. ábra: menu2, Forrás: /var/www/html/index.php)

Ez a kódrészlet egy „menu2” elágazást kezel az alkalmazásban, ami a felhasználó menüválasztására reagál. Több elágazás is van én most csak egyet mutatok be ezek közül.

- Ha a kérés „GET” módban érkezik a „menu2” útvonalon, akkor a kód csatlakozik egy PostgreSQL adatbázishoz és végrehajt egy lekérdezést a „public.usdae” táblából.
- Az eredményt egy tömbben tárolja el, majd, ha van eredmény (tehát nem üres), akkor a Smarty template motorhoz rendeli a kapott eredményeket és egy változót állít be a táblázat oszlop számainak meghatározásához.
- Ha nem „GET” a kérés metódusa, akkor visszairányítja az oldalt a főoldalra.

Összességében ez a kód a „menu2” választásra reagálva adatokat kér az adatbázisból, majd ezeket átadja a Smarty sablonnak az oldal megfelelő megjelenítéséhez. Ha nem „GET” a kérés, akkor visszatér a főoldalra.

4.4. Stíluselemek

Rövid részletben szeretnék megemlíteni egy-két stíluselemet. A következő kódok `/var/www/html/smarty/templates/` szerver mappájában találhatóak meg.

4.4.1. Footer

```
<footer class="bg-dark text-center text-white page-footer">
    <!-- Grid container -->
    <div class="container p-4">
        <div class="w3-xlarge w3-padding-16">
            <a href="/">{{ $trans['OnlineDieteticExpert'] }}</a>
            <a alt="Validated by HTML Validator" title="Validated by HTML
Validator" width="10" height="5"></a>
        </div>
        {*&p>Powered by <a
href="https://www.w3schools.com/w3css/default.asp"
target="_blank">w3.css</a></p*&}
        <a>{{ $trans['Figyelem'] }} </a>
    <!-- /Section: Social media -->
    </div>
    <!-- /Grid container -->
    <div class="container" style="text-align:right; z-index:10;
position:relative; bottom:5rem">
        <i onclick="topFunction()" class="fa-solid fa-up-long fa-2x1
pagetop-arrow"></i>
    </div>
```

(18. ábra: Footer, Forrás: `/var/www/html/smarty/templates/footer.tpl`)

Ez egy HTML kódrészlet, amely a footert (alsó lábléc) határozza meg.

- `footer class` határozza meg a lábléc stílusát, jelen esetben (sötét háttér, középre igazított, fehér szöveg.)
- A `{{ $trans['OnlineDieteticExpert'] }}` egy változót jelöl, ami az oldal fordításánál lényeges, meghívja a jelenleg használt translationst és annak tartalmában megkeresi az 'OnlineDieteticExpert' hallgató változót. (vö. `/var/www/html/translations/(hu.yaml` vagy `en.yaml)`)
- Az `<i>` tag tartalmaz egy ikont, amire kattintva a `topFunction()` JavaScript függvényt hívja meg, ami a lap tetejére vezet.

```

<!-- Copyright -->
    <div class="text-center p-3" style="background-color: rgba(0, 0, 0, 0.2)">
        @ 2023 Copyright:
        <a class="text-white"
href="/">{$trans['OnlineDieteticExpert']}</a>
    </div>
<!-- /Copyright -->

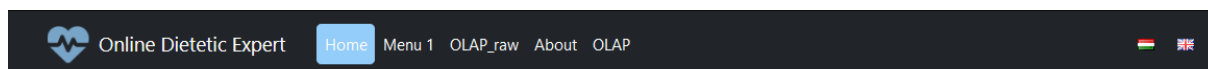
```

(19. ábra: Copyright, Forrás: /var/www/html/smarty/templates/footer.tpl)

Ez a kódrészlet a fenti kód aljához tartozik. Egy szerzői jogi védeltséget (Copyright) jelöl, amely az oldal tartalmának védeltségét hivatott jelezni. Itt is megjelenik a fent említett \$trans, „text-white osztály” segítségével a szöveg színe fehér lesz, átlátszó fekete háttérszín, szöveg középre igazított.

4.4.2. Navbar

Ez egy Bootstrap alapú navigációs sáv(navbar), interaktív mobilra is optimalizált, tartalmaz egy logót, valamint egy mobil nézetben használható lenyíló menüt.



(20. ábra: Navigációssáv, Forrás: https://www.dieteticexpert.hu/)

```

<nav class="navbar navbar-expand-lg bg-dark border-bottom border-bottom-dark sticky-top">
    <!-- Container wrapper -->
    <div class="container">
        <button class="navbar-toggler" type="button" data-mdb-
toggle="collapse" data-mdb-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
            <i class="fa-solid fa-bars fa-bounce fa-lg"></i>
        </button>
        <!-- Navbar brand -->
        <a class="navbar-brand" href="/"><i class="fa-solid fa-heart-
pulse fa-beat-fade fa-2x1" style="color: #94ccfa;"></i> &nbsp; <span>
{$trans['OnlineDieteticExpert']}</a>

```

(21. ábra: Navbar, Forrás: /var/www/html/smarty/templates/navigation.tpl)

- Ez a navigációs sávot határozza meg a Bootstrap .navbar és azok kombinációja alapján, amelyek a sötét háttérre, alulra igazított alapvonalra, és az oldal tetején rögzített elrendezésre utalnak ('sticky-top').

- '.container' meghatározza a navigációs sáv tartalmát és elrendezését a weboldalon.
- button class="navbar-toggler": Egy gomb a navigációs sáv mobil nézetéhez, amely egy hagyományos hamburger ikont használ a lenyíló menü megjelenítéséhez.
- <a class="navbar-brand": Ez a navigációs sáv logóját jelöli.

```
<div style="width:1rem;"></div>

<!-- Collapsible wrapper -->
<div class="collapse navbar-collapse" id="navbarSupportedContent">
  <ul class="navbar-nav me-auto mb-2 mb-lg-0">

    <!-- Link home, leads you to several places based on clicked objects on the
navigation bar -->
    <li class="nav-item" {include 'activator.tpl' name='home' req=$request}>
      <a class="nav-link" href="/">{$trans["home"]}</a>
    </li>

    {assign "x" "menu2" nocache}
    <li class="nav-item" {include 'activator.tpl' name=$x req=$request}>
      <a class="nav-link" href="/{$x}">{$trans[$x]}</a>
    </li>
  </ul>
```

(22. ábra: Navbar menu2, Forrás: /var/www/html/smarty/templates/navigation.tpl)

- Navigációs sávon megjelenő menüpontok, jelen esetben menu2 kezelése.

```
<!-- Icons -->

<ul class="navbar-nav d-flex flex-row me-1">
  <li class="nav-item me-3 me-lg-0">
    <a class="nav-link" href="/language/hu"><i
class="flag flag-hungary"></i></a>
  </li>
  <li class="nav-item me-3 me-lg-0">
    <a class="nav-link" href="/language/en"><i
class="flag flag-uk"></i></a>
  </li>
</ul>
```

(23. ábra: Ikon, Forrás: /var/www/html/smarty/templates/navigation.tpl)

- Ez a navigációs sáv lehetővé teszi a felhasználók számára, hogy a különböző ikonokra kattintva váltsanak a honlap különböző nyelvi verziói között. Az ikonok a flag-hungary és flag-uk osztályokkal vannak ellátva, a magyar és angol zászlók megjelenítésére.

5. Titkosítás és Webcím

5.1. Tartománynév

Már a dolgozat korai szakaszában szükségem volt egy tartománynévre, ez több okból is hasznos volt. Szinte elképzelhetetlennek találom, hogy az emberek többsége képes legyen emlékezni folyamatosan egy egyedi IP-címre, illetve mivel eleinte egy dinamikusan változó IP-cím volt és nem statikus, ezért ezt folyamatosan közölnünk kellett volna a felhasználó felé. Első lépésként egy ingyenes, dinamikus DNS (DDNS) szolgáltatás felé kezdtem el mozogni, amit meg is találtam a <https://www.noip.com> weboldalon. DDNS lehetővé teszi, hogy egy dinamikus IP-címmel rendelkező eszköz rögzített domain (tartomány) névvel rendelkezzen.

Current Hostnames: 1		
Hostname	IP/URL	Action
dieteticexpert.ddns.net	13.51.177.80	 Modify 

(24. ábra: DDNS, Forrás: <https://www.noip.com>)

Így kaptam meg a <https://dieteticexpert.ddns.net/> URL címet. Mivel dinamikus változó IP címmel kezdtünk azért kellett még egy Kliens (DUC), ami folyamatosan nézi az IP-cím béli változásokat, és automatikusan updateli azt az oldalon. Ezt a következő URL címen lehet letölteni: <https://www.noip.com/download?page=win>

Később mivel az AWS (vö. Molnár Csanád szakdolgozatában a 6.1 fejezetben) statikusan tárolta az IP címünket ezért nem volt szükségünk a továbbiakban az automatikus frissítésre. Ezután a professzionális megjelenés érdekében beszereztem egy domaint, amit később .hu végződéssel láttam el. Rövid keresés után találtam rá a <https://www.websupport.hu/> oldalra, ahol számos szolgáltatás elérhető, ezek között a domain regisztrálása is. Azért esett rájuk a választásom, mert igényes megjelenésű weboldallal és felhasználó barát admin felülettel

rendelkeznek, több mint ötszázhetvenezer regisztrált domaint tudhatnak maguk után Európában, illetve van Magyar/Angol ügyfélszolgálatuk.

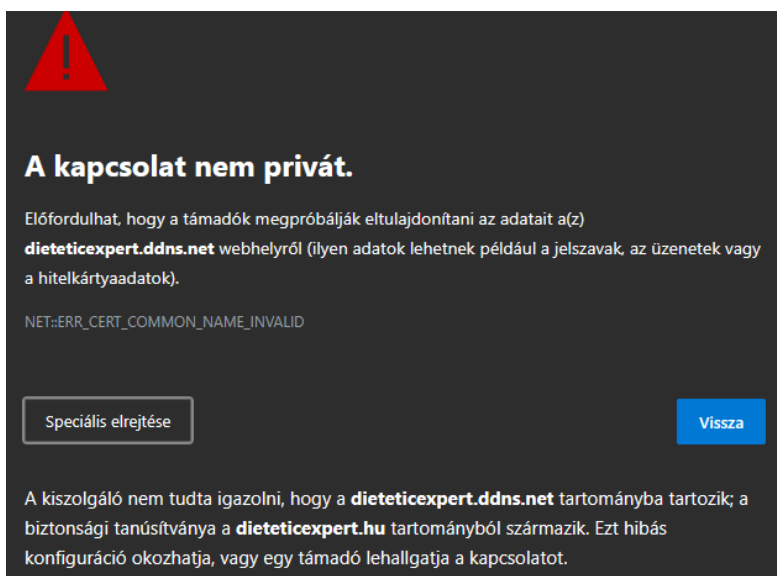
Vásárlás után az admin felületre belépve, már lehetőségem is volt testre szabni és beállítani a www.dieteticexpert.hu cél IP-jét.

☒	www.dieteticexpert.hu	13.51.177.80	600
-------------------------------------	-----------------------	--------------	-----

(25. ábra: Tartománynév, Forrás: <https://admin.websupport.hu/>)

5.2. Certificate

Ahhoz, hogy a felhasználó biztonságosan tudjon felmenni a weboldalunkra, szükségünk lesz egy hiteles tanúsítványra. Ez a tanúsítvány képes titkosítani az adatokat azáltal, hogy azokat egy titkosított csatornán, vagyis egy SSL/TLS-kapcsolaton keresztül továbbítja. Ennek eredményeként növeli az adatok biztonságát és védelmét a weboldalakon. Eleinte saját készítésű tanúsítványunk volt, de ez nem hiteles körülmények között készült.



(26. ábra: Nem privát kapcsolat, Forrás: saját készítés)

A döntésemet itt is az előző (l. 5.1. Tartománynév) feladat logikáját követve hoztam meg, mivel mint említettem több szolgáltatásuk is van ezért náluk tudtam SSL tanúsítványt is vásárolni. Alapvetően 3 féle opcióból tudok választani, röviden:

- **Domain Validated (DV) tanúsítvány:** Alapvető tanúsítvány, amely megerősíti a domain tulajdonjogát.
- **Organization Validated (OV) tanúsítvány:** Ebben az esetben azonosítják és megerősítik a vállalat adatait is.
- **Extended Validation (EV) tanúsítvány:** Ez a legmagasabb szintű SSL-tanúsítvány, ami részletesen ellenőrzi az üzleti adatokat és a vállalat hitelességét.

Mivel ez egy kisebb weboldal, számomra a DV SSL tanúsítvány biztosítja az alapvető biztonságot és megbízhatóságot.

A vásárlás után elő alítottak számomra egy privát kulcsot, tanúsítványt, CSR fájlt, illetve egy Root CA fájlt, ezekről röviden:

- **certificate.crt:** Ez a közvetlenül a SSL-tanúsítványodhoz tartozó fájl, ami tartalmazza az általam vásárolt tanúsítványt. Amikor a felhasználók a weboldaladra navigálnak, ezt a fájlt használja a böngésző a biztonságos kapcsolat kiépítéséhez.
- **csr.csr:** A Certificate Signing Request (CSR) a tanúsítvány igénylés során létrehozott fájl, ami tartalmazza a publikus kulcsot, és amelyet a tanúsítvány-kibocsátó (pl. tanúsítvány-szolgáltató) használ a tanúsítvány létrehozásához.
- **privatekey.key:** Ez a privát kulcs fájl, amely az SSL-tanúsítványhoz tartozik. Ez a kulcs szükséges a titkosításhoz és a biztonságos kapcsolat kialakításához a szerver és a felhasználók között.
- **root_ca.crt:** A tanúsítvány-kibocsátó (CA) gyökér tanúsítványát tartalmazza. Ez az a tanúsítvány, amely hitelesíti a SSL-tanúsítványt, és amelynek a segítségével a böngészők ellenőrzik a tanúsítvány hitelességét.

Ezek a fájlok együtt szükségesek ahhoz, hogy a SSL-tanúsítványt telepítsem és konfiguráljam. Ezt a szerveren található: /etc/apache2/sites-available/ mappában lévő 000-default.conf fájlban átkell írni, ez adja meg az elérési útvonalát, valamint a kapott fájlokat a /etc/apache2/certs/ mappába feltölteni.

```
</VirtualHost>
<VirtualHost *:443>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    SSLCertificateFile /etc/apache2/certs/certificate.crt
    SSLCertificateKeyFile /etc/apache2/certs/privatekey.key
    SSLCertificateChainFile /etc/apache2/certs/root_ca.crt
</VirtualHost>
```

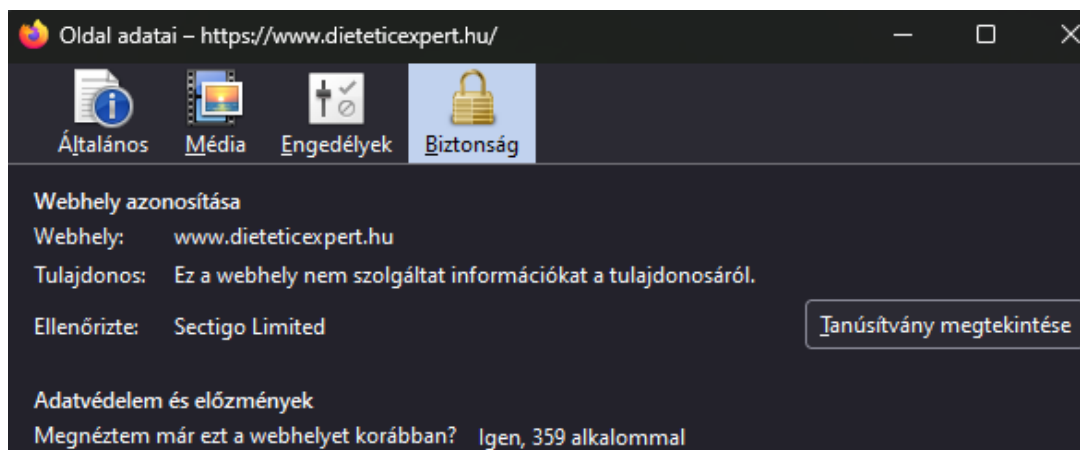
(27. ábra: Útvonal, Forrás: /etc/apache2/sites-available/000-default.conf)

Ez fontos lépés, mivel, ha az elérési útvonalon nem találja meg a fájlokat akkor nem indul el.

```
systemd[1]: Starting The Apache HTTP Server...
apachectl[695]: AH00526: Syntax error on line 39 of /etc/apache2/sites-enabled/000-default.conf:
apachectl[695]: SSLCertificateFile: file '/etc/apache2/certs/apache.crt' does not exist or is empty
apachectl[669]: Action 'start' failed.
apachectl[669]: The Apache error log may have more information.
systemd[1]: apache2.service: Control process exited, code=exited, status=1/FAILURE
systemd[1]: apache2.service: Failed with result 'exit-code'.
systemd[1]: Failed to start The Apache HTTP Server.
```

(28. ábra: Sikertelen belépés, Forrás: saját kép)

A weboldalon (Firefox böngészőben) az URL-sáv mellett lévő kis (általában) lakat ikonra kattintva, majd kapcsolat biztonságos fülön tovább haladva érhető el a további tudnivalók, ahol lehetőség van a tanúsítvány megtekintésére.



(29. ábra: Tanúsítvány megtekintése, Forrás: saját kép)

6. A weboldal tesztelése

Röviden szeretném szemléltetni a kész weboldalt és a funkcióit.

A <https://www.dieteticexpert.hu/> weboldalra fellépve, lehetőségünk van több menüsor közül választani, illetve a navigációs sávon lehetőség van jobb felső sarokban kiválasztani, hogy magyar vagy angol nyelvre van szükségünk. Ami számunkra fontos az az OLAP menü pontban található. Lehetőségünk van kiválasztani különböző ételcsoportokból, milyen tápanyagokra szeretnénk szűrni, majd a Category vagy Kategória menüpontban kiválasztott elemekkel, töltődik fel a következő két menüpont. Ezeket a felhasználó beállítja a saját igényei szerint, majd a SEARCH gombra kattintva megkapja a keresés eredményeit. A COCO kiértékelési folyamata megtalálható Molnár Csanád dolgozatában a 9 és 11. fejezetben.

Online Dietetic Expert Home Menu 1 OLAP_raw About OLAP

OLAP service

[Back](#)

Made based on sources provided by USDA, food-attribute table Source: [USDA](#)

Select a food group: Seeds

Category: Protein(g), Total lipid (fat)(g), Carbohydrate, by difference(g), Ash(g)

More: Protein(g)

Less: Total lipid (fat)(g)

SEARCH

Lehet minden másként egyforma

"NDB Number"	"FullName"	"Protein(g)"	"Total lipid (fat)(g)"	"Carbohydrate, by difference(g)"
12001	BREADFRUIT SEEDS, RAW	7.4	5.59	29.24
12003	BREADFRUIT SEEDS, BOILED	5.3	2.3	32

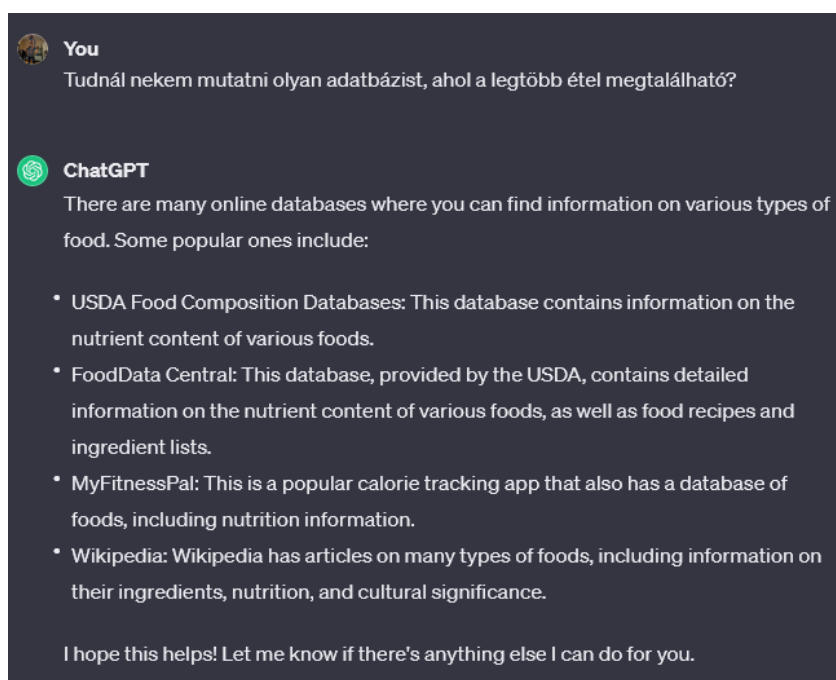
(30. ábra: Webteszt, Forrás: <https://www.dieteticexpert.hu/>)

7. Chat GPT

7.1. Támogatás

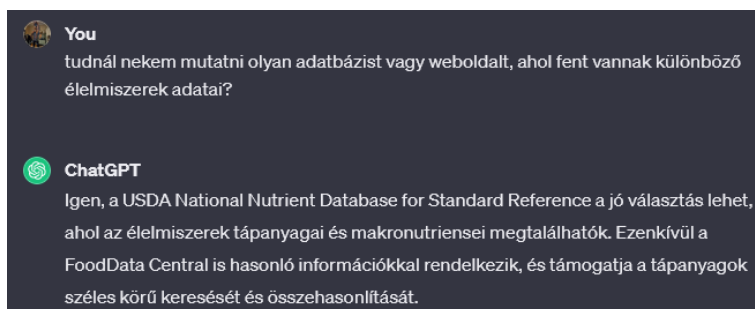
A Chatgpt-t mint támogató mesterséges intelligenciát, több helyen is alkalmaztam dolgozatom során. A mai világban, aki jól tudja használni az AI-t annak hatalmas mértékben tud segítséget, információkat, tanácsokat adni. Egy szabály van, jól kell tudni kérdezni.

Elsőkörben szerettem volna megvizsgálni, pár alapvető kérdéssel, amit én magam is letudok ellenőrizni gyorsan és hatékonyan, hogy mennyire megbízható. Mivel az AI-t folyamatosan tanítják és fejlesztik, így egyre jobb lesz, de mint minden technológia ezt sem tökéletes és előfordulnak hibák, ezért én törekedtem a számomra elfogadható módon ellenőrizni a fontosabb kérdéseimre kapott válaszokat.



(31. ábra: ChatGPT, Forrás: saját készítés)

Itt próbáltam ellenőrizni a saját munkám elejét, így megkérdeztem tőle, hogy tudna-e nekem mutatni olyan adatbázist, ahol a legtöbb étel jelen van. Ezt a kérdést utólag már pontosabban tenném fel, mivel használata során megtapasztaltam, hogy muszáj pontosan fogalmaznunk, ha pontos és érthető választ szeretnénk tőle. Külön érdekesség, hogy erre a kérdésre még angolul válaszolt, de későbbiekben már magyarul kaptam meg a válaszomat, ha magyarul kérdeztem.

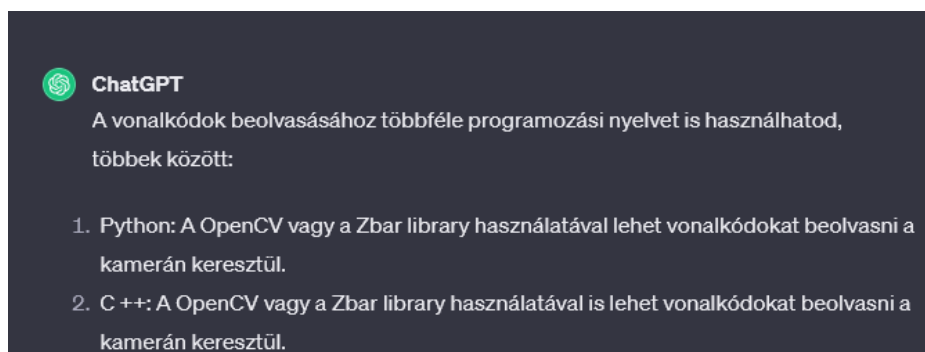


(32. ábra: ChatGPT támogatása)

Ez is része annak, hogy folyamatosan fejlődik az AI. Tapasztaltam azt is, hogy kérdésekre, egy bizonyos idő elteltével más-más választ kaptam, de ez nem fordult elő gyakran. Esetemben sokszor volt olyan válasz, amit nem fogadtam el tőle, vagy csak egyszerűen nem tetszett, ilyenkor újra feltettem, esetleg megkértem, hogy írja át a válaszát.

7.2. Ötlet szerzés

Az AI-t eleinte próbáltam ötletelésre bírni és kértem tőle tanácsokat. A dolgozat tervezési fázisában volt egy olyan ötlet, amiben a felhasználó egy vonalkód olvasó segítségével feltudja vinni az élelmiszer adatait egy adatbázisba.



(33. ábra: ChatGPT vonalkód, Forrás: saját készítés)

Majd megkérdeztem, hogy tudna-e írni egy ilyen kódot Pythonban és ő készségesen elkezdte írni a kódot.

Az ötletelésnek a ChatGPT-vel nem az volt az elsődleges célom, hogy mindent felhasználjak belőle, hanem hogy több nézőpontból új gondolataim szülessenek.

7.3. Kódsorok ellenőrzése és javítása

A kódolási folyamatom során, javarészt a legtöbb kód ellenőrzésre került a ChatGPT által, illetve sok esetben kértem tőle kód béli segítséget, javaslatot. Sok esetben kaptam tőle javításra kódot, ami nem csak szintaxis hibás volt, hanem logikai is amire javaslatokat kaptam tőle.

a kért kódsor alján legtöbb esetben még azt is leírja, hogy a kódban elnevezett változót mire cseréljem ki a saját kódomban, így is segítve az esetleges hibák elkerülését.

```
<?php if ($_SERVER['REQUEST_METHOD'] === 'POST'): ?>
    <h2>Selected Options:</h2>
    <p>Table 1 Option: <?php echo $selectedOptions['table1']; ?></p>
    <p>Table 2 Option: <?php echo $selectedOptions['table2']; ?></p>
    <p>Table 3 Option: <?php echo $selectedOptions['table3']; ?></p>
<?php endif; ?>
</body>
</html>
```

In this code:

- Replace the option values and labels inside each `**<select>**` with the actual values you have in your tables.
- The PHP code handles form submissions and stores the selected options in the `**\$selectedOptions**` array.
- After submitting the form, the selected options from each table are displayed.

(34. ábra: ChatGPT ellenőrzés, Forrás: saját készítés)

Tantárgyak kapcsolata a szakdolgozattal

- **Európai civilizáció és identitás**
Érintőlegesen szerepel a dolgozatban, a dietetikai problémákkal küzdő emberek száma növekvőben van Európa szerte.
- **Szakterületi jogi ismeretek**
Részlegesen tartalmazza, felhasználó jogai.
- **Matematikai alapok**
Részlegesen jelenik meg, a élelmiszer adatok feldolgozása során.
- **Hálózatok és számítógépes architektúrák**
l. 5. fejezet.
- **Adatszerkezetek és algoritmusok**
Kódolás során jelenik meg, pl. tömbként való adattárolás, funkciók létrehozása.
- **Operációs rendszerek**
l. 3. fejezet.
- **Programozási alapelvek és módszertanok**
l. 4. fejezet.
- **Rendszermodellezés**
Folyamatábra készítése, előre gondolkodás.
- **Felhasználói interfészek és vizualizáció**
l. 4.4.1. és 4.4.2. alfejezetekben, weboldal vizualizációja.
- **Rendszertervezés**
Fejlesztési folyamatok tervezése, alkalmazás, weboldal felépítése.
- **Programozás I,II,III**
l. 4. fejezet.
- **Vezetés és vállalkozási ismeretek**
Nem kapcsolódik a dolgozathoz.
- **Komplex társadalomtudományi ismeretek**
Nem kapcsolódik a dolgozathoz.
- **Vállalati gazdaságtan**
Részben kapcsolódik, alapesetben ingyenes, domain, tanúsítvány, tárhely leváltása.
- **Adatbázisok I,II**
l. 2. fejezet.

- **Szoftverüzemeltetés**
Weboldal biztonságos üzemeltetése, szoftver beágyazása.
- **Szoftvertesztelés**
l. 6. fejezet.
- **Elektronikus áramkörök**
Nem kapcsolódik.
- **Informatika védelem és biztonság**
l. 5. fejezet, SSL, DOMAIN.
- **Machine Learning**
l. 7. fejezet, ChatGPT beszélgetés.
- **Internet és világ**
Adatvédelmi szempontból köthető l. 5. fejezet.
- **Az elektronika fizikai alapjai**
Nem kapcsolódik a dolgozathoz.
- **IT-biztonsági fejlesztések minőség- és projektmenedzsmentje**
Részben kapcsolódik a dolgozathoz.
- **Innovatív információs és kommunikációs technológiák az IT-biztonság kapcsán**
l. 5. fejezet.

ÖSSZEGZÉS

A projektet alapvetően sikerként könyvelem el, az elején elképzelt tervek közül számos megvalósult, többek között:

- Hiteles adatbázis felállítása, elegendő adatmennyiséggel.
- Megfelelő, stabil környezet létrehozása a szervernek
- Adatok (élelmiszer) sikeres közlése, a weboldal és az ODESSAI felé.
- Két nyelvűség megoldása
- Mesterséges intelligencia bevonása
- Kódolási folyamat sikeressége

Egyéb fejlesztési lehetőség, amikről szó esett, de a jelen végtermék nem tartalmazza:

- Login system a felhasználók számára, hogy e-mail cím/jelszó-t megadva a felületre belépve saját magára szabható beállításokat kapjon, pl. életkor, testsúly, magasság, alapvető beállítások.
- A felhasználó, ha van betegsége eltudja menteni és az alkalmazás ajánlásokat dob fel neki pl. cukorbeteg(alacsony szénhidrát, magas fehérje tartalom).
- Élelmiszerek, átlagárral ellátása vagy egy vonalkód beolvasó rendszer, amivel a vonalkódot beolvasva konkrét termék felvitelére lenne lehetőség.
- Egy étrend összerakó funkció, amely lehetővé teszi, hogy összerakj magadnak egy ételt, és megmondja összesen mennyi tápértéket tartalmaz pl. sajtos csirkemell : rárakott sajt tápértéke + csirkemell tápértéke.
- Fórum rendszer, ahol a felhasználók megoszthatnak recepteket, nyithatnak topikot, ahol megbeszélhetnek bármi dietetikához kapcsolódó témát.

Kivonat– HU

A cél létrehozni egy olyan AI támogatott alkalmazást, ami segít, az ezen szakterületen lévő orvosok információ szerzését, különböző élelmiszerek széleskörű tápanyagtartalmáról. Az alkalmazás lehetővé teszi a szakértő számára, a gyors és hatékony diagnózis felállítását és tanácsadását. Az alkalmazást egy projekt keretein belül valósult meg, szaktársammal Molnár Csanáddal készítettük el. Ezáltal a dolgozatomban több helyen is hivatkozom rá, mivel több különálló feladatot dolgoztunk fel, amik szorosan kapcsolódnak egymáshoz. Az én feladataim inkább a külső megjelenésre, az adatvagyon letisztítására, karbantartására, weboldallal kapcsolatos feladatokra stb., de természetesen van átfedés pl. kódolási folyamat, ötletgyártás, hiba ellenőrzés stb.

Kivonat – EN

The goal is to create an AI-supported application that helps doctors to obtain information about the extensive nutritional content of different foods. The application enables experts to swiftly and efficiently establish diagnoses and provide guidance. The application was developed within the framework of a project, created with my colleague Csanád Molnár. As a result, I refer to it in several places in my thesis, since we worked on several separate tasks that are closely related to each other. My responsibilities primarily revolved around the application's external appearance, data refinement, maintenance, website-related tasks, etc. However, there is overlap, such as in the coding process, idea generation, error checking, etc."

IRODALOMJEGYZÉK

- **Mák Erzsébet - Étrendtervező dietetikai tanácsadó szoftver fejlesztése mesterséges intelligenciával**
Doktori értekezés, Semmelweis Egyetem, Patológiai Doktori Iskola 2010 URL:
http://old.semmelweis.hu/wp-content/phd/phd_live/vedes/export/makerzsebet.d.pdf
- **Kupcsikné Fitus Ilona - SQL-lekérdezések összeállítása: a kezdő lépések elsajátítása MS Access-ben**
E-learning tananyag / Webuni online kurzus Budapest: Gábor Dénes Főiskola 2015
URL:
<https://neptun.gdf.hu/oktato/CommonControls/SaveFileDialog.aspx?Type=&id=170505204&Func=LoginDocumentations>
- **Szabó Zsanett, Visnovitz Márton - Programozási alapfogalmak sorrendje a módszeres és funkcionális programozásban**
ELTE, Informatikai kar URL:
<http://konferenciak.inf.elte.hu/infodidact/InfoDidact21/Manuscripts/SzZsVM.pdf>
- **Bódy Bence - Az SQL példákon keresztül - (Második kiadás)**
Jedlik Oktatási Stúdió Kft. 2019 URL:
<https://docplayer.hu/4363430-Body-bence-az-sql-egy-ujabb-peldan-keresztul.html>
- **Abonyi-Tóth Andor - A weblapkészítés technikája (HTML5, CSS3) és ergonómiája**
ELTE Informatikai Kar 2012 URL:
<https://tamop412.elte.hu/tananyagok/weblapkeszites/index.html>
- **Az új diéta - A Magyar Dietetikusok Országos Szövetségének hivatalos, lektorált folyóirata.**
Magyar Dietetikusok Országos Szövetsége – MDOSZ 2019 URL:
https://mdosz.hu/hun/wp-content/uploads/2019/12/ud-2019-5_final.pdf
- **Rikk János, Pitlik László - ChatGPT-kísérletek – programozás**
Magyar Internetes Agrárinformatikai Újság 2023 KJE URL:
https://miau.my-x.hu/miau/298/chatgpt_.docx
- **Jónás Attila; Pitlik László - ChatGPT-verziókat összehasonlító elemzések I.**
Magyar Internetes Agrárinformatikai Újság 2023 KJE URL:
[https://miau.my-x.hu/miau/298/ChatGPT_GPT4%20\(1\).docx](https://miau.my-x.hu/miau/298/ChatGPT_GPT4%20(1).docx)

- **Szabó Bálint - Webprogramozás I.**

Eszterházy Károly Főiskola, Eger 2013 URL:

<https://mek.oszk.hu/14000/14068/pdf/14068.pdf>

- **Balássy György – honlapépítés a XXI. században**

Jedlik Oktatási Stúdió Kft. 2010 URL:

<https://mek.oszk.hu/10300/10392/10392.pdf>

HIVATKOZÁSJEGYZÉK

- Kész termék weblapjának URL - címe: www.dieteticexpert.hu
- A MIAU weblapja, COCO elérési útjának URL címe: <https://miau.my-x.hu/myx-free/coco/index.html>
- Magyar weboldal, élelmiszer adatbázis: <https://kaloriabazis.hu/>
- USDA (United States Department of Agriculture – az Egyesült Államok Mezőgazdasági Minisztériuma): <https://www.usda.gov/>
- FDC (FoodData Central – Élelmiszertár) weboldalra: <https://fdc.nal.usda.gov/>
- Visual Studio Code a dolgozat során használt fejlesztői környezet:
<https://code.visualstudio.com/download>
- Office 365, a dolgozat során használt WORD, EXCEL, ACCESS programokat tartalmazó csomag: <https://www.microsoft.com/hu-hu/microsoft-365/get-office-and-microsoft-365-oem-download-page>
- WinSCP egy szoftver, ami lehetővé teszi a biztonságos fájlküldést és -fogadást távoli szerverek között: <https://winscp.net/eng/download.php>
- Putty szoftver egy ingyenes és könnyen használható SSH és távoli kapcsolatok kezelője Windows operációs rendszerhez:
<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- Puttygen egy eszköz, amely kulcspárok generálását teszi lehetővé a Puttyhoz, SSH-kapcsolatokhoz és titkosított kommunikációhoz.: <https://www.puttygen.com/>
- PostgreSQL, nyílt forráskódú, erős relációs adatbázis-kezelő rendszer:
<https://www.postgresql.org/>
- VirtualBox, egy virtualizációs szoftver, amely lehetővé teszi virtuális gépek létrehozását: <https://www.virtualbox.org/>
- Debian, nyílt forráskódú rendszer: <https://www.debian.org/download>
- DDNS szolgáltató weboldala: <https://www.noip.com>
- DDNS kezelő alkalmazás: <https://www.noip.com/download?page=win>
- Tartománynév és tanúsítvány vásárlásának a weboldala: <https://www.websupport.hu/>
- Admin felület, a domain és a certificate konfigurálásához:
<https://admin.websupport.hu/>