

VAS VÁRMEGYEI SZAKKÉPZÉSI CENTRUM
HORVÁTH BOLDIZSÁR
KÖZGAZDASÁGI ÉS INFORMATIKAI
TECHNIKUM

A 5 0613 12 03 számú Szoftverfejlesztő és –tesztelő vizsgaremek

A szoftveralkalmazás dokumentációja

Készítették:

JURCSÓ ANDRÁS KORNÉL
SZANYI ÁDÁM

SZOMBATHELY

2024

Tartalom

1.	Bevezetés.....	3
2.	A szoftver célja.....	4
3.	Fejlesztői környezet.....	5
3.1.	Microsoft Visual Studio.....	5
3.2.	Microsoft Visual Studio Code	5
3.3.	MySQL	5
4.	Az asztali alkalmazás bemutatása.....	6
4.1.	A program tervezése	6
4.2.	A program specifikációi.....	7
4.3.	A program telepítése, és igénybevétele	7
4.4.	A program alapértelmezett elemei.....	8
5.	Bejelentkezés a programba.....	11
6.	A főmenü és hozzá tartozó menük bemutatása	12
6.1.	Az Étlap	12
6.1.1.	Új Termék Felvitele.....	13
6.1.2.	Étlap Módosítása	14
6.1.3.	Termék Típusa.....	15
6.1.4.	Feltét Hozzáadása / Feltét Módosítása	16
6.2.	Módosítások.....	17
6.3.	Segítség.....	18
7.	A programkódokról	19
8.	Továbbfejlesztési tervek.....	20
9.	A weboldal bemutatása.....	21
9.1.	A program tervezése.....	21
9.2.	A weboldal specifikációi	21

9.3. A weboldal működtetése és felépülése	22
9.3.1. Backend	22
9.3.2. Frontend.....	23
9.3.3. Adatbázis	25
9.4. Weboldalt alkotó nyelvek.....	26
9.5. Weboldal terve röviden	27

1. Bevezetés

A dokumentáció a „Pizza_Adminisztrációs_felület” nevű asztali alkalmazásról és a hozzá készült Weboldalról szól.

A fejlesztés során próbáltunk egy olyan fejlesztői környezetet kialakítani, amellyel a weboldalunk általános információit és a hozzá tartozó kulcsfontosságú elemeit tudjuk módosítani, illetve feltölteni.

A témaválasztásunk azért esett egy ilyesfajta weboldal létrehozására, mivel az érdekeltségi körünkben megtalálható ez a fajta szolgáltatás, és szeretnénk segíteni a munkánkkal.

A weboldalunkon és az asztali alkalmazásunkban megtalálható Liget Pizzériával kapcsolatos leírások nem minden esetben pontosak és nem minden esetben tükrözik a valóságot.

Az információk tömkelege csak a programunkat és a weboldalunkat szándékoztatja bemutatni.

A következő fejezetben szemléltetem a szoftver célját, valamint a fejlesztői környezetet is, amely bemutatja milyen programokat, eszközöket és programozási nyelveket használtunk a fejlesztés során.

A dokumentáció tartalmaz képeket, kódrészleteket és példákat, illetve szó esik még a program tervezéséről is.

2. A szoftver célja

Egy olyan program létrehozása, amellyel egy átlag felhasználó is tudja módosítani a weboldalt programozói tudás nélkül is. Ezáltal pénzt és időt spórolva tudja a felhasználó úgy változtatni az oldalon található információkat, hogy azokat valós időben látja a programon belül és a weboldalon is.

3. Fejlesztői környezet

3.1. Microsoft Visual Studio

Az asztali alkalmazásunk a Microsoft által publikált integrált fejlesztői környezetben készült. A program kódolásra, a kód szerkesztésére, hibák javítására, tesztelésre és a különböző platformokra való fejlesztésre szolgál. Ezen felül a program fejlesztői környezete rendkívül fejlesztő barát, amely megkönnyítette a munkánkat. A GitHub verziókezelő támogatása és az intelligens kódkiegészítés is csak a program előnyeire szolgál.

C# Nyelv

- Egy programozási nyelv, amelyet a Microsoft a .NET keretrendszer részeként fejlesztett ki, amely egy objektumorientált programozási nyelv.

3.2. Microsoft Visual Studio Code

A Weboldalunk a Microsoft által fejlesztett, nyílt forráskódú kódszerkesztőben valósult meg, mely támogatja a GitHub-ot, rendelkezik hibakeresővel, továbbá intelligens kódkiegészítésre is képes.

HTML Nyelv

- A HyperText Markup Language az internetes tartalmak megjelenítésére és weboldalak készítésére fejlesztettek ki. A HTML strukturális nyelv, amely leírja a weboldalak szerkezetét és annak tartalmát.

CSS Nyelv

- A Cascading Style Sheets egy stílusalapú nyelv, mely weboldalak megjelenésének formázására használnak. Segíti a strukturálisabb elkülönítést, ezáltal a könnyebben karbantartható kialakítás létrehozását.

JavaScript Nyelv

- A nyelv segítségével dinamikus tartalmakat és műveleteket tudunk létrehozni a weboldalunkon. A mai modern Web-ben kulcsfontosságú szerepet vállal, hiszen segítségével interaktív és dinamikus webes élményt lehet vele létrehozni.

3.3. MySQL

A MySQL egy relációs adatbázis kezelő rendszer, mely táblákban tárolja az adatokat. A szoftver nyílt forráskódú, így ingyenesen letölthető, módosítható. További nagy előnye, hogy platformfüggetlen.

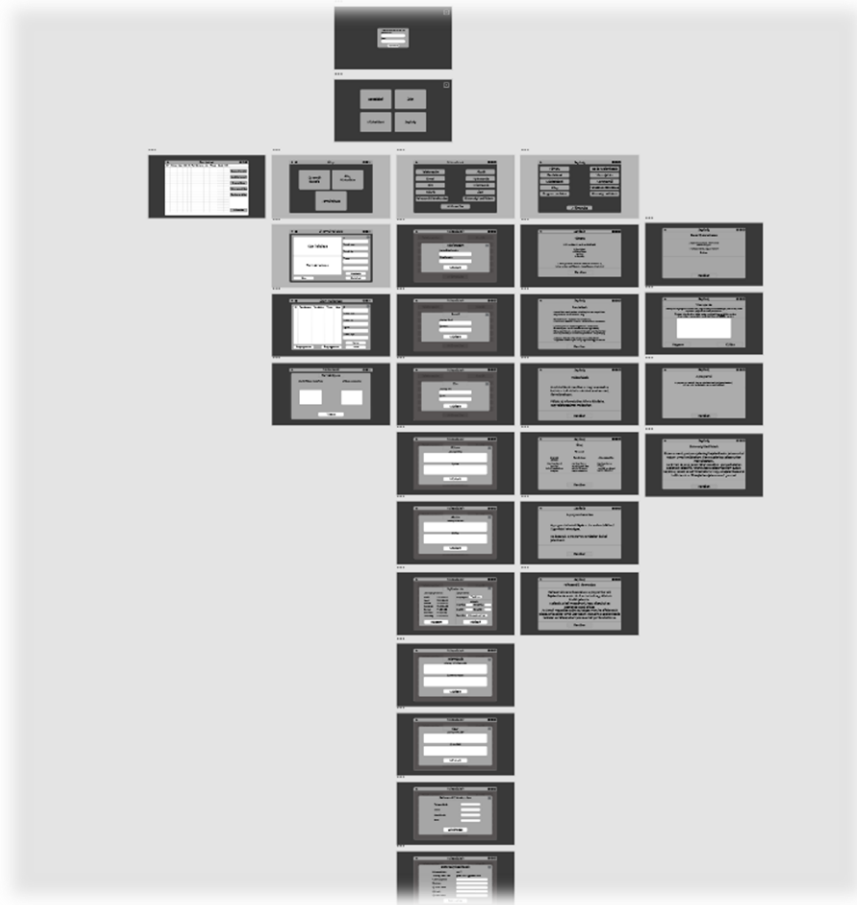
4. Az asztali alkalmazás bemutatása

Ebben a fejezetben bemutatom az asztali alkalmazás specifikációit, telepítési útmutatóját és a program készítését megelőző dizájnt. Az ismertetés jobb szemrevételezése miatt képeket és program kódokat is láthatunk.

4.1. A program tervezése

A program tervezése az Adobe XD (Experience Design) nevű programban készült, amely egy prototípuskészítő és tervező program, amelyet mobilalkalmazások, webes tartalmak és egyéb felhasználói felület tervezésére használnak.

Ezen felül az asztali alkalmazásban használt gombok átmenete az Adobe Photoshop képszerkesztővel készült.



1. ábra: A program vázlata

4.2. A program specifikációi

Az asztali alkalmazásunk a Microsoft Visual Studio (17.9) verziószámú programban, C# nyelven íródott, amely kapcsolódik egy MySQL adatbázishoz.

A program írásakor Windows10-es 64 bites operációs rendszert használtunk.

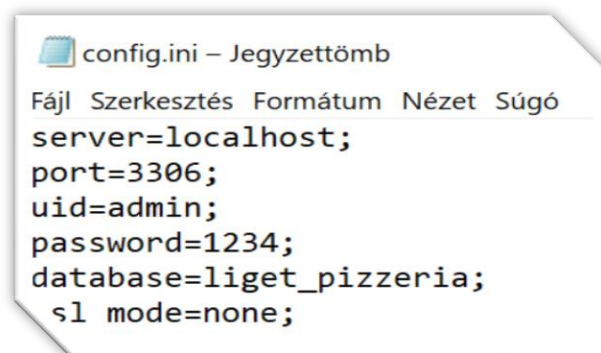
4.3. A program telepítése, és igénybevétele

A program telepítése előtt az adatbázisba töltjük fel az adatokat!

A telepítési fájlok mellett megtalálja az adatbázisba beimportálandó (liget_pizzeria.sql) fájlt. Amint az importálás megtörtént, telepítsük fel a programot „Pizza_Adminfelulet.exe”-vel történik.

A program telepítése után a program fő könyvtárában a „config.ini” fájl segítségével állítsa be igény az adatbázishoz való csatlakozás paramétereit.

Amennyiben a program indításakor összeomlást tapasztalna, kérem ellenőrizze, hogy a „config.ini” fájlban minden adat helyesen került-e megadásra.



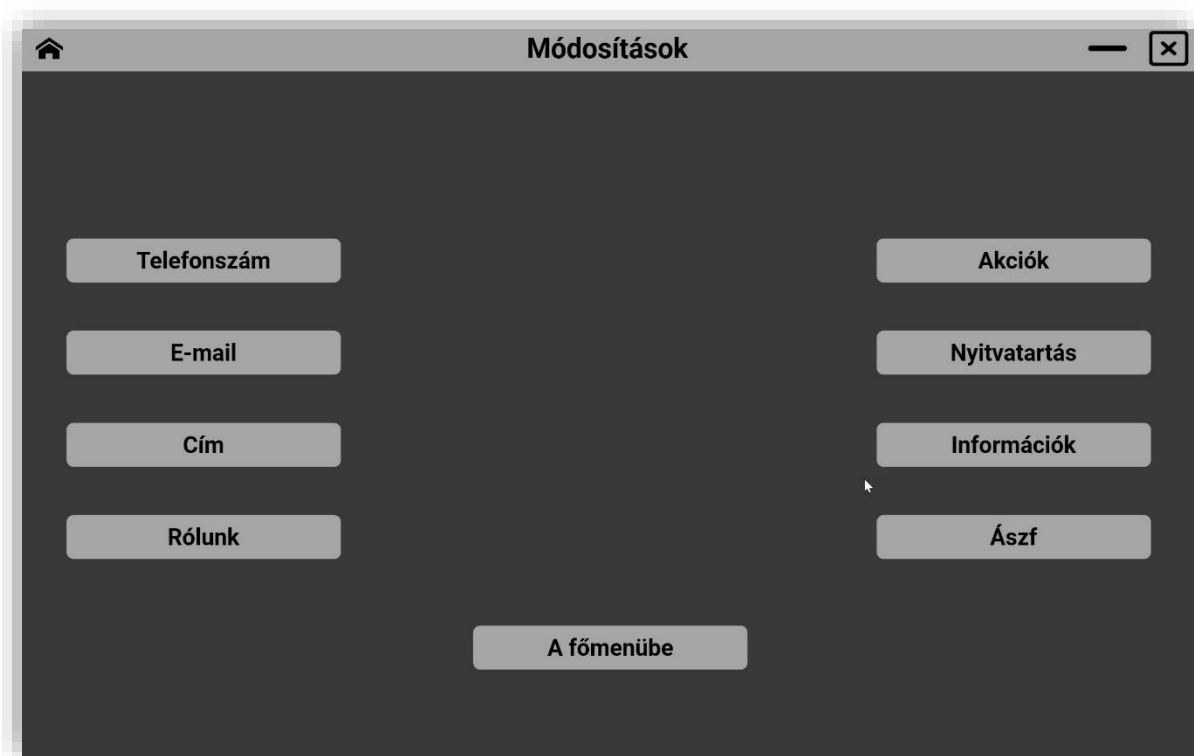
2. ábra: A config.ini fájl alapértelmezett paramétereit

4.4. A program alapértelmezett elemei



3. ábra: Adminisztrációs felület

- A fejlécen szereplő cím, amely mindig az aktuális oldalt jelöli.
- A kis méretű gomb, mely segítségével lerakhatjuk az alkalmazásunkat a tálcára.
- Kilépés a programból gombot a kis mérettel megegyezően mindig a program fejlécében találjuk meg.



4. ábra: Módosítások

- A házikó ikon a program fejlécében a bal felső sarokban található és csak akkor látjuk, amikor a programba beléptünk és nem a főmenüben vagyunk.



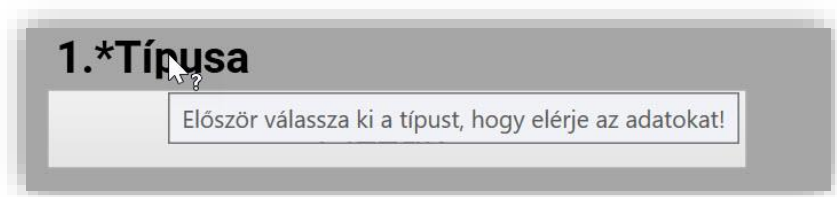
5. ábra: Bejelentkezési felület / Gomb inaktív / Eszköz tipp

- A fókusz nélküli gomboknál, ahogy az adminisztrációs felület képénél is látható, addig nem tudunk a gombra kattintani, amíg a kötelező (*) jelölésű mezőket ki nem töltöttük.
- Ezen felül gombok, vagy beviteli mezőknél is találkozhatunk, úgynevezett eszköz tippekkel. Ami segíthet, hogy mit is kéne tennünk.



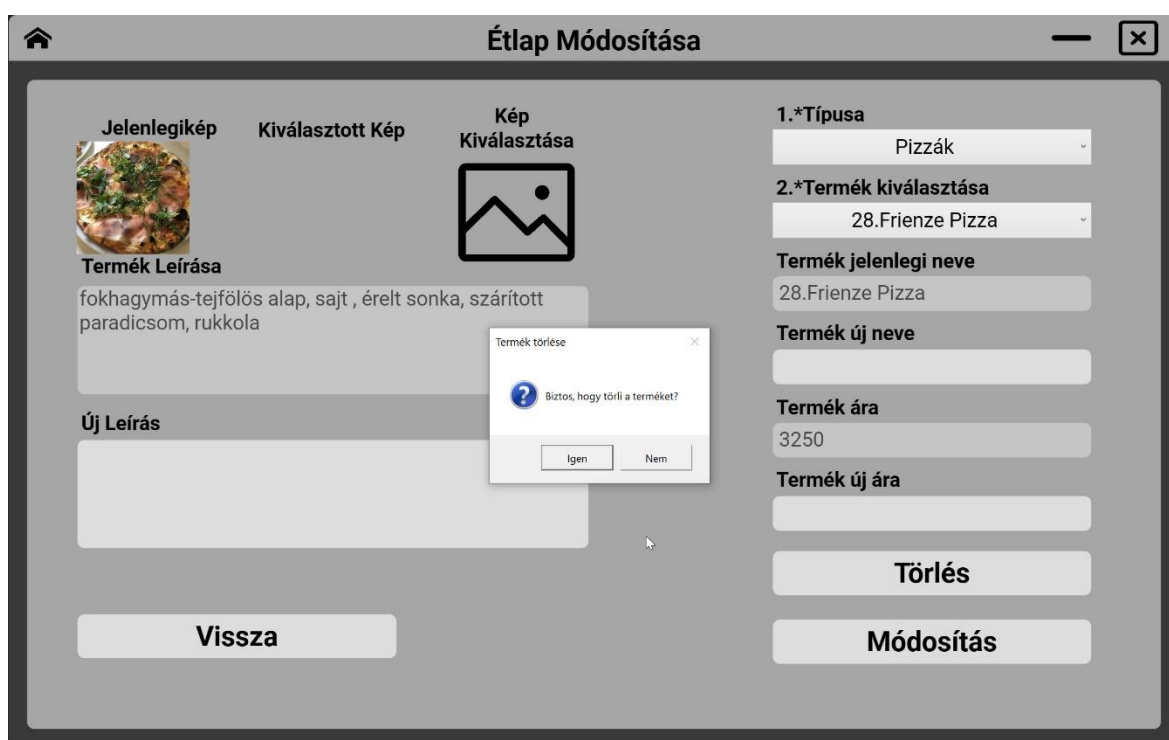
6. ábra: Bejelentkezési felület / Gomb Aktív / Egér mutató megváltozik

- Minden gomb esetében, amikor ráhúzzuk az egerünket, a gomb színe megváltozik, ez segíti a felhasználókat a könnyebb interakció teljesítéséhez.
- Továbbá, amikor egy adott gombon állunk, az egér mutatója egy mutató kézjellé változik.



7. ábra: Eszköz tippek

- A „?” elemek egy a főbb címeznél jelennek meg, ha ráhúzzuk az egeret. Amennyiben az egérmutató nem vált át „?” karakterre akkor nincs főbb leírás.



8. ábra: Visszakérdezés effektus

- A programban való módosításainknál megjelenik a visszakérdezési effektus, ami véletlen módosításokat és törléseket hivatott elkerülni.

5. Bejelentkezés a programba

A programba való bejelentkezéshez jelenleg két információra van szükségünk; a felhasználónévre és a hozzá tartozó jelszóra, amely a mi esetünkben a könnyű bejelentkezés miatt az alábbi paraméterekkel tehető meg:

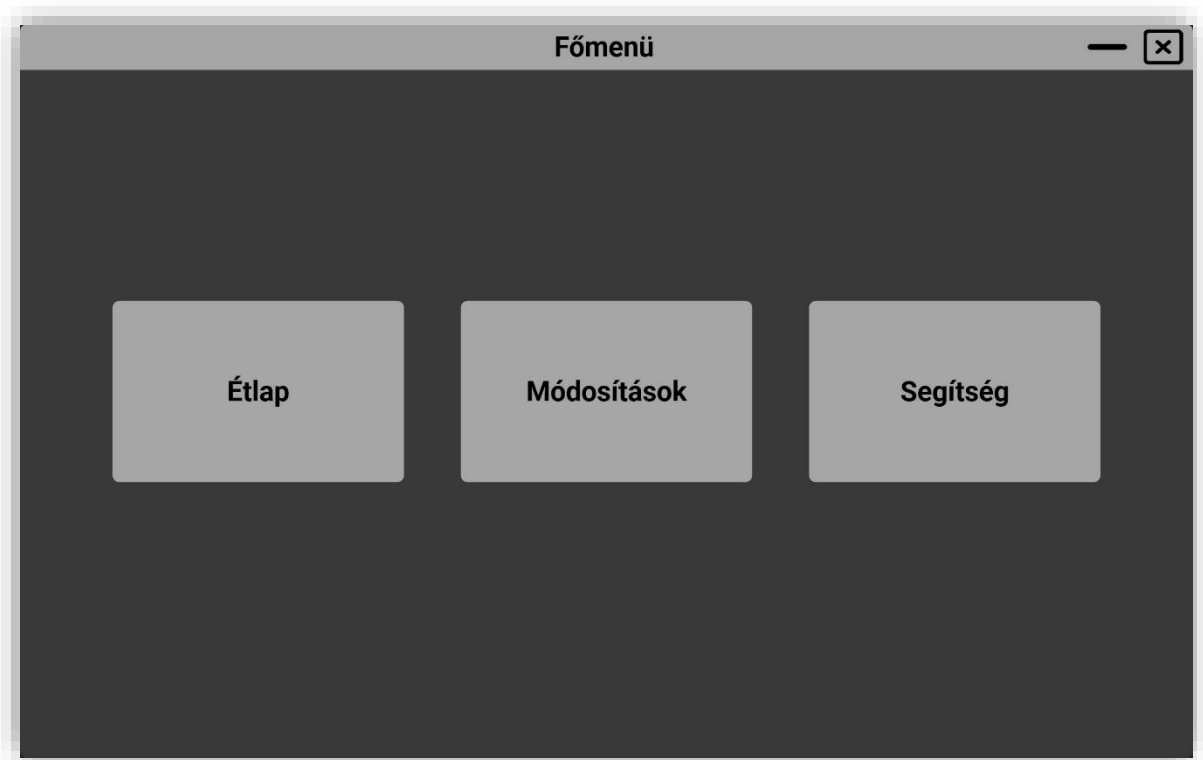
- Felhasználónév: **admin**
- Jelszó: **1234**



9. ábra: Bejelentkezési képernyő/Csak a program indításakor látható felület

6. A főmenü és hozzá tartozó menük bemutatása

Főmenü



10. ábra: A bejelentkezést követő „Főmenü”

A következő leírás bemutatja a főmenü almenüinek „füleit” és azok működését.

A főmenüben 3 darab almenüt találhatunk:

- Étlap
- Módosítások
- Segítség

6.1. Az Étlap

Az „Étlap” menüben 4 fül található:

- Új Termék Felvitele
- Étlap Módosítása
- Termék Típusa
- Feltét Hozzáadása / Feltét Módosítása

6.1.1. Új Termék Felvitele

The screenshot shows a web form titled "Új Termék Felvitele". The form is divided into two main sections. The left section contains a large box labeled "*Kép Kiválasztása" (Image Selection) and a text area labeled "*Termék Leírása" (Product Description). The right section contains three input fields: "*1. Típusa" (Type) with a dropdown menu showing "Ital", "*2. Termék Neve" (Product Name), and "*3. Termék Ára" (Product Price). Below these fields are two buttons: "Alaphelyzet" (Reset) and "Feltöltés" (Upload). At the bottom left of the form is a "Vissza" (Back) button. The form is styled with a light gray background and dark text.

11. ábra: Új termék felvitele

- Új termék felvitelekör kötelező a (*) csillaggal megjelölt elemeket kell kitölteni és kiválasztani.
- A számozott sorrend nem fontos, csak figyelemfelkeltő információ!
- Sorolja be termékét a kívánt típusba.
- Hogyha a típus mező üres, akkor hozzon létre egyet az Étlap Főmenü → Termék Típusa almenüben található típus hozzáadásával.
- Adja meg a termék nevét, árát és válassza ki a termékhez kapcsolódó képet, illetve ne feledkezzen meg a Termék leírásáról sem!
- Amennyiben bármely (*) jelölt elemet kihagyja, addig a feltöltés gomb nem jelenik meg.
- Az alaphelyzet gomb az eddigi mezőkbe beírt szövegeket és a kiválasztott képet teszi üresre, hogyha vissza szeretnénk vonni a mezők tartalmát.
- Ugyan ezt az eredményt kapjuk akkor is, ha megnyomjuk a vissza gombot is, valamint hogyha ugyanígy elhagyjuk a menüt.
- Amennyiben a termék értékénél szám helyett betűt adunk meg, akkor a termék ára 0-a lesz.
- A termék feltöltéséről visszajelzést kapunk, amely után a mezők alapértelmezetten üresre állnak, így megkönnyebbítve a további termék feltöltéseket.

6.1.2. Étlap Módosítása



12. ábra: Étlap módosítása

- Az Étlap Módosításánál is a (*) megjelölt elemeket kötelező kiválasztania, ezt követően tudja is a terméket módosítani.
- A módosítás gomb is megerősítő választ kér majd, így az esetleges félrekattintások elkerülhetők.
- Tudjuk a kiválasztott terméknek a nevét, az árát, a leírását és a képét is módosítani.
- Amennyiben nem kell Önnek az adott termék, abban az esetben a törlés gomb megnyomásával törölheti a terméket.
- A törlés csak abban az esetben végrehajtható, amennyiben a törlést a felugró értesítésben megerősíti.
- A termék utólagos visszavonására nincsen lehetőség, kérjük minden esetben legyen körültekintő, mivel mindig az az elem fog törölődni, amelyik ki van választva a „2. Termék kiválasztása” részben!

6.1.3. Termék Típusa

Módosítások

Termék Típusa

Meglévő Típus Módosítása

Jelenlegi típusok

Ital
Pizzák
Saláták
Tálak

Típus neve

Hozzáad

Módosít

Törlés

Vissza

Új Típus hozzáadása

Típus neve

Hozzáad

13. ábra: Termék típusa

- A termék típusánál tudunk új típust hozzáadni, valamint a jelenlegi típust tudjuk módosítani és törölni.
- Amennyiben a típust törölnénk, azt körültekintően tegyük meg. Amennyiben letörlünk egy típust, akkor a típushoz tartozó összes termék törlődni fog.
- A típus neve megváltoztatása nem jár ilyen hátránnyal.
- A „Módosít” mezőben mindig a jelenlegi típusokban kiválasztott elemnek a nevét tudjuk megváltoztatni.
- A típus feltöltésének (hozzáadás) és a típus módosít gombja csak akkor aktiválódik, ha a kívánt mezők nem üresek.

6.1.4. Feltét Hozzáadása / Feltét Módosítása

The screenshot shows a web application window titled "Módosítások" (Modifications). It is divided into two main sections: "Meglévő Feltétek Módosítása" (Modify Existing Conditions) on the left and "Új Feltét hozzáadása" (Add New Condition) on the right.

Meglévő Feltétek Módosítása (Left Panel):

- Jelenlegi feltétek (Current Conditions):** A dropdown menu showing "Fokhagyma", "Hagyma", "Póréhagyma", and "Gomba".
- Feltét ára (Condition Price):** A text input field containing the value "250".
- Új feltét neve (New Condition Name):** A text input field.
- Új ár (New Price):** A text input field.
- Módosít (Modify):** A button.
- Törlés (Delete):** A button.
- Vissza (Back):** A button.

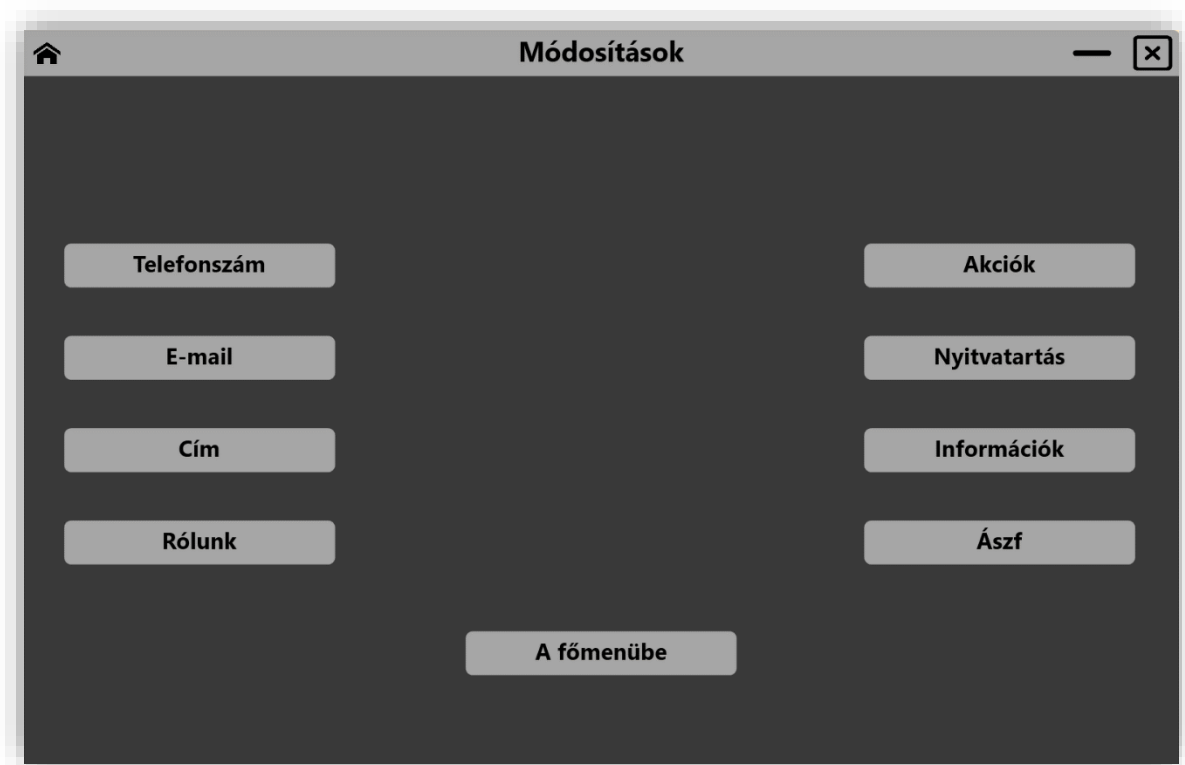
Új Feltét hozzáadása (Right Panel):

- Feltét neve (Condition Name):** A text input field.
- Feltét ára (Condition Price):** A text input field.
- Hozzáad (Add):** A button.

14. ábra: Feltét hozzáadása / Feltét módosítása

- Új feltét hozzáadásakor adjuk meg a feltét nevét és az árát.
- A mezők nem lehetnek üresek, amint beírtuk a feltét nevét és árát, a hozzáadás gomb működőképes lesz.
- Abban az esetben, hogyha a feltét áránál betűt adunk meg szám helyett, akkor 0 értéket kapunk a feltét feltöltése után.
- A jelenlegi feltétek kiválasztása után az „Új feltét neve” mezőben tudjuk beírni a kívánt új nevet, amit a „Módosít” gombbal tudunk megváltoztatni.
- A „Feltét ára” úgy módosítható, hogy kiválasztjuk „Jelenlegi feltétek” résznél a módosítandó feltétet, majd az „Új ár” mezőbe adjuk meg az új árat.
- Ezt követően a „Módosít” gomb már használható lesz és már tudunk is módosítani.

6.2. Módosítások



15. ábra: Módosítások

A „Módosítások” menüben a weboldalunkhoz tartozó elemeket tudunk módosítani.

Az adott elem megnyitása után láthatjuk a jelenlegi elem tartalmát és azt a „Módosít” gomb megnyomásával tudjuk is változtatni.

6.3. Segítség



16. ábra: Segítség

A „Segítség” menüpontban kisebb leírásokat kapunk a program elemeinek működéséről, illetve minimálisan hasonlít a program dokumentációjához.

A „Visszajelzés” ikonra kattintva látható az email-címünk, amire várjuk a programmal kapcsolatos kérdéseket, esetleges visszajelzéseket, amely segítségével még jobbá tehetjük a programunkat.

7. A programkódokról

A továbbiakban részletezünk néhány programkódot a programunkból.

A programkódot meghívjuk, amikor letörölnénk egy típust a jelenlegi típusok közül.

```
1 reference
void termék_torol() // Termék Módosítása az adatbázisban
{
    if (listBox_jelenlegitipusok.SelectedItem != null)
    {
        string kivalasztottTípus = listBox_jelenlegitipusok.SelectedItem.ToString();
        MySqlConnection con = new MySqlConnection(kapcsolat);
        con.Open();
        string sql = "DELETE FROM `tipus` WHERE `tipus_nev` = @kivalasztottTípus";
        MySqlCommand msqcl = new MySqlCommand(sql, con);
        msqcl.Parameters.AddWithValue("@kivalasztottTípus", kivalasztottTípus);
        msqcl.ExecuteNonQuery();
        con.Close();
    }
    else
    {
        MessageBox.Show("Válasszon ki egy típust a törléshez!", "Típus törlése", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

A kódrészlet a kimaradt kódok egy részét is mutatja, illetve egy elrejt / megjelenít módszert, arra az esetre, hogyha nem használunk bizonyos elemeket, akkor elrejthetjük.

```
1 reference
// void rendelesek_megjelenit_gombok_elrejt()//Segítség Rendelések előhívása,gombok elrejtése,hozzá tartozó elemek megjelenítése. ...
1 reference
void modositások_megjelenit_gombok_elrejt() //Segítség Módosítások előhívása,gombok elrejtése,hozzá tartozó elemek megjelenítése.
{
    button_fomenu.Visibility = Visibility.Hidden;
    //button_rendelesok.Visibility = Visibility.Hidden;
    button_modositasok.Visibility = Visibility.Hidden;
    button_etlap.Visibility = Visibility.Hidden;
    button_program_bezarasa.Visibility = Visibility.Hidden;
    button_bekijelentkezes.Visibility = Visibility.Hidden;
    button_visszajelzes.Visibility = Visibility.Hidden;
    button_programrol.Visibility = Visibility.Hidden;
    button_etlapmodositasa.Visibility = Visibility.Hidden;
    button_feltethozzaadasa.Visibility = Visibility.Hidden;
    button_feltetmodositasa.Visibility = Visibility.Hidden;
    button_ujtermek.Visibility = Visibility.Hidden;
    button_segitsegek.Visibility = Visibility.Hidden;
    //button_felhasznalo_letrehozasa.Visibility = Visibility.Hidden;
    //button_biztonsagi_beallitasok.Visibility = Visibility.Hidden;
    button_a_fomenube.Visibility = Visibility.Hidden;

    label_modositasok.Visibility = Visibility.Visible;
    textBlock_menusleiras_modositasok.Visibility = Visibility.Visible;
    border_fomenu.Visibility = Visibility.Visible;
    button_rendben.Visibility = Visibility.Visible;
    border_takaras.Visibility = Visibility.Visible;
}
1 reference
void modositások_elrejt_gombok_megjelenit() //Segítség Módosítások elrejtése,gombok megjelenítése,hozzá tartozó elemek elrejtése.
{
    button_fomenu.Visibility = Visibility.Visible;
    //button_rendelesok.Visibility = Visibility.Visible;
    button_modositasok.Visibility = Visibility.Visible;
    button_etlap.Visibility = Visibility.Visible;
    button_program_bezarasa.Visibility = Visibility.Visible;
    button_bekijelentkezes.Visibility = Visibility.Visible;
    button_visszajelzes.Visibility = Visibility.Visible;
    button_programrol.Visibility = Visibility.Visible;
    button_etlapmodositasa.Visibility = Visibility.Visible;
    button_feltethozzaadasa.Visibility = Visibility.Visible;
    button_feltetmodositasa.Visibility = Visibility.Visible;
    button_ujtermek.Visibility = Visibility.Visible;
    button_segitsegek.Visibility = Visibility.Visible;
    //button_felhasznalo_letrehozasa.Visibility = Visibility.Visible;
    //button_biztonsagi_beallitasok.Visibility = Visibility.Visible;
    button_a_fomenube.Visibility = Visibility.Visible;

    label_modositasok.Visibility = Visibility.Hidden;
    textBlock_menusleiras_modositasok.Visibility = Visibility.Hidden;
    border_fomenu.Visibility = Visibility.Hidden;
    button_rendben.Visibility = Visibility.Hidden;
    border_takaras.Visibility = Visibility.Hidden;
}
```

8. Továbbfejlesztési tervek

A programkódban megtalálhatóak komment formájában fel nem használt kódrészletek, amelyek a programba is beépítésre kerültek volna, de sajnos idő hiánya miatt ideglenesen kikerültek.

A jövőbeli tervek között szerepel ezeknek a kódrészleteknek a visszaépítése és továbbfejlesztése.

A továbbfejlesztett változatban szeretnénk a főmenüben egy rendelés almenüt készíteni, amely képes lenne a beérkezett rendeléseket összesíteni és azokat egy fájlba kiexportálni.

Szeretnénk továbbá a felhasználói adatokat biztonságosabbá tenni, így titkosítást alkalmaznánk a programba való belépés során.

Továbbá beépítésre kerülne, hogy tudjunk új felhasználókat adni a programhoz és a jelenlegi felhasználónk is meg tudja változtatni a hozzá rendelt e-mail címét és jelszavát.

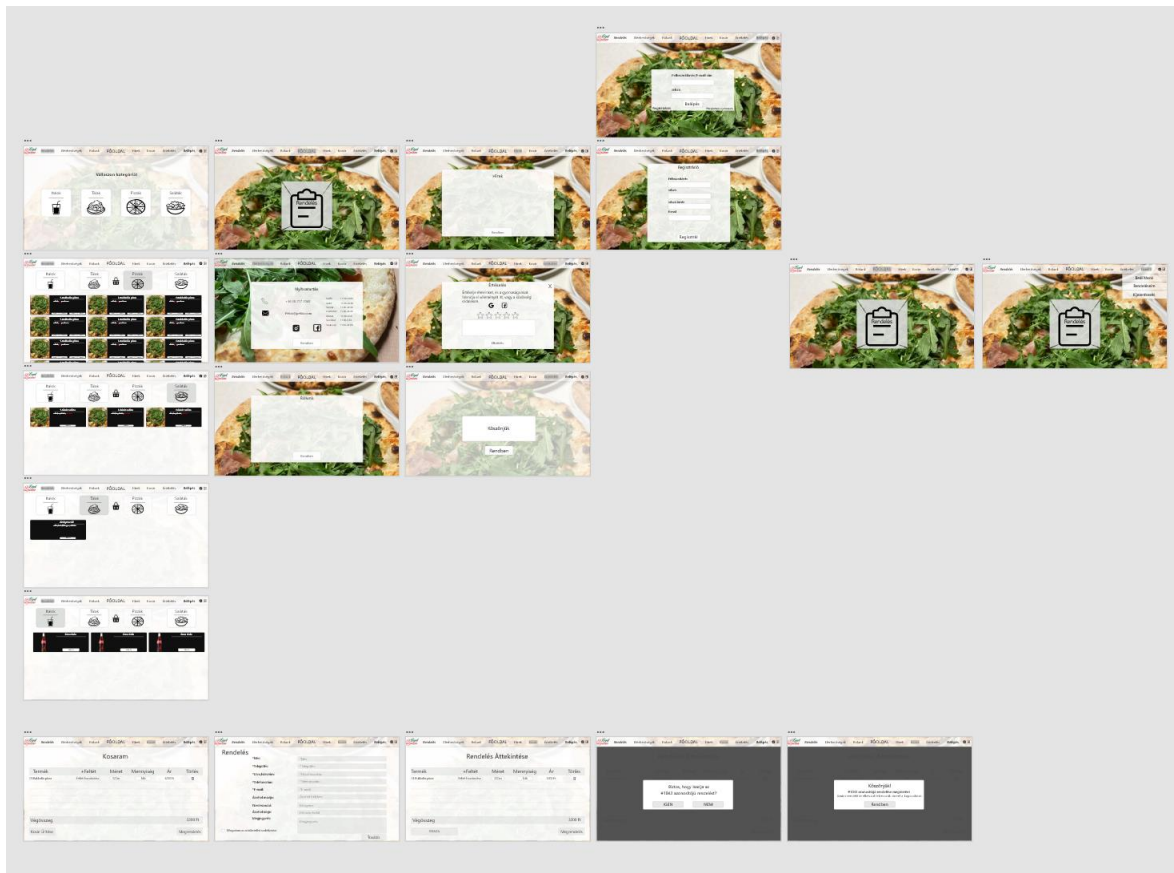
9. A weboldal bemutatása

Ebben a fejezetben bemutatom, hogy az általunk készített weboldal milyen komponensekből épül fel, milyen részeket, program nyelveket és a program készítését megelőző dizájnt tartalmaz.

Az ismertetés jobb szemrevételezése miatt képeket és program kódokat is láthatunk.

9.1. A program tervezése

Hosszas beszélgetések, Interneten való keresés weboldal dizájn ötletek után, papírra vetett rajzok, ábrák által készült el az első tervezetünk, mely a későbbiekben az Adobe XD (Experience Design) nevű programban készült, ami egy prototípuskészítő, és tervező program, amelyet mobilalkalmazások, webes tartalmak és egyéb felhasználói felület tervezésére használnak.



9.2. A weboldal specifikációi

A weboldalunk a Visual Studio Code(1.88) verziószámú programban, html, css, javascript, (JSON), sql nyelveken íródott, amely kapcsolódik egy MySQL adatbázishoz.

A program írásakor Windows10-es 64 bites operációs rendszert használtunk.

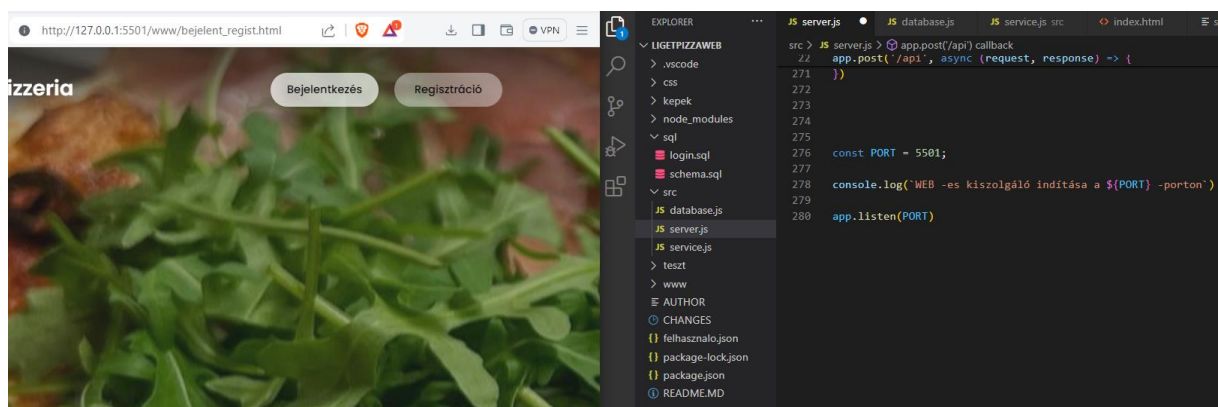
9.3. A weboldal működtetése és felépülése

A weboldalunk 3 fő részből épül fel: Backend-ből, Frontend-ből és Adatbázisból.

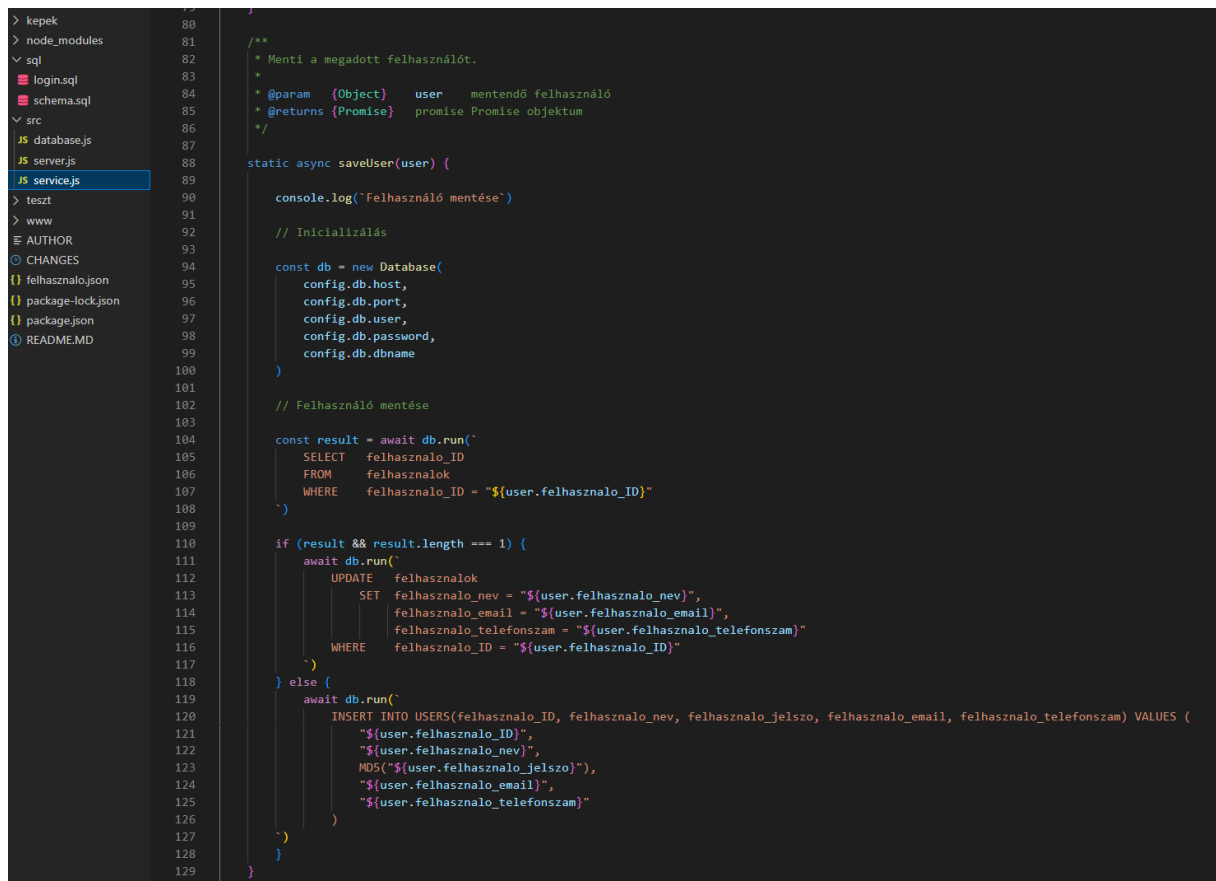
9.3.1. Backend

A mi esetünkben szerver oldalon fut és kommunikál a kliens oldallal(frontend). Felelős több fontos dologért a weboldalban, például:

- Az adatkezelésért és az adatbázissal való interakcióval: A backend feladata az adatok kezelése, lekérdezése és azoknak manipulálása az adatbázisban. Például az adatbázisból való adat lekérdezése, oda való adat beszúrása, ott lévő adat módosítása vagy törlése, valamint az adatok ellenőrzése.
- “Üzleti” logika végrehajtásáért: A backend ebben az esetben a weboldal “üzleti” logikájának végrehajtásáért felelős. Ez alatt a weboldal szabályait, folyamatait és funkcióinak megvalósítását értem.
- Felhasználó kezelés és hitelesítés: A backend azonosítja és hitelesíti a felhasználókat, valamint kezeli a felhasználói munkameneteket és engedélyeket. Ez magában foglalhatja a felhasználók bejelentkezését, kijelentkezését, regisztrációját, jelszókezelést stb.
- A kommunikáció a frontenddel: A frontend és a backend közötti kommunikáció biztosítása is a backend felelőssége. Ez magában foglalja az adatok küldését és fogadását a frontend és a backend között, például API-hívások, amiket mi is alkalmazunk.
- Biztonság: A backendnek gondoskodnia kell az alkalmazás biztonságáról, beleértve az adatbiztonságot, hozzáférési jogosultságokat, adatvédelmi szabályokat stb.



A képen látható URL: amit a Backend-ben az 5501-es porton indítottunk el, különben helyi hálózaton tudtuk volna futtatni a weboldalunkat.



A képen látható service.js (Backend része) file-amiben definiáltuk, hogy a weboldalon kitöltött felhasználónak mely adatait mentse el, tárolja el az adatbázisunk.

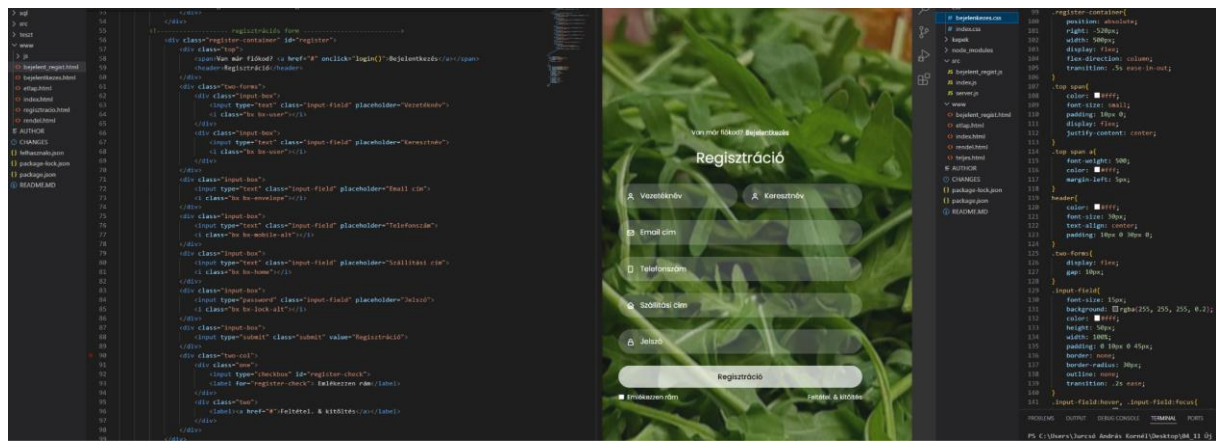
Összességében a backend egy kulcsfontosságú része az alkalmazásnak, amely gondoskodik az adatok kezeléséről, az üzleti logika végrehajtásáról és a kommunikációról a frontenddel, hogy az alkalmazás folyamatosan és hatékonyan működhessen.

9.3.2. Frontend

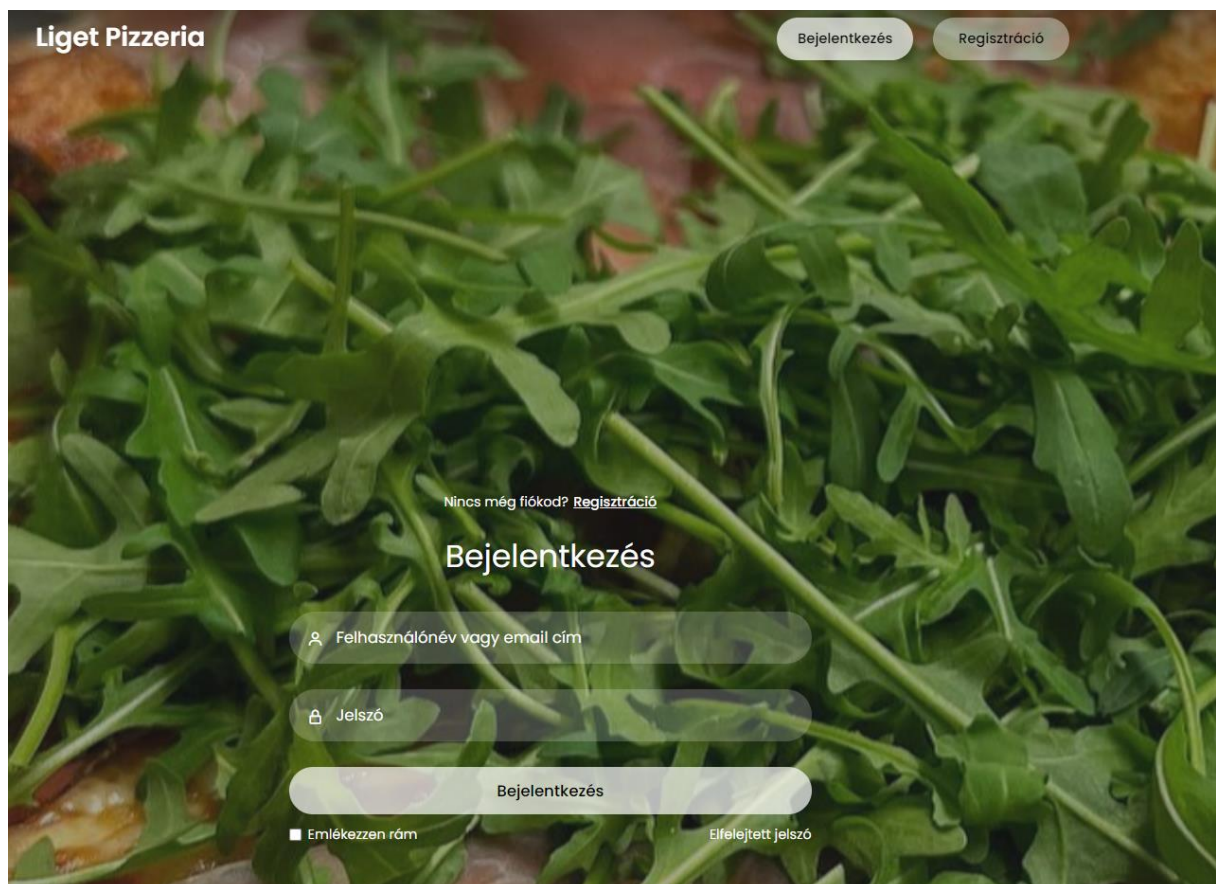
Megjeleníti az alkalmazás felhasználói felületét és kezeli a felhasználói interakciókat. A mi esetünkben kliens oldalon fut, a felhasználó böngészőjében. Felelős több fontos dologért a weboldalba, mint például:

- Felhasználói felület megjelenítése: A frontend az alkalmazás felhasználói felületét hozza létre és jeleníti meg a felhasználó számára. Ez magában foglalhatja az oldalak elrendezését, a navigációs menük megjelenítését, a felhasználói vezérlő elemeket.
- Felhasználói interakciók kezelése: amikor a felhasználó kapcsolatba lép a mi esetünkben a weboldallal és interakcióba lép vele.

- Adatok megjelenítése és frissítése: A frontend felelős a weboldalon tárolt adatok megjelenítéséért és frissítéséért. Ez lehet például adatok lekérése egy szerverről, majd ezek megjelenítése a felhasználói felületen.
- Kommunikáció a backenddel.



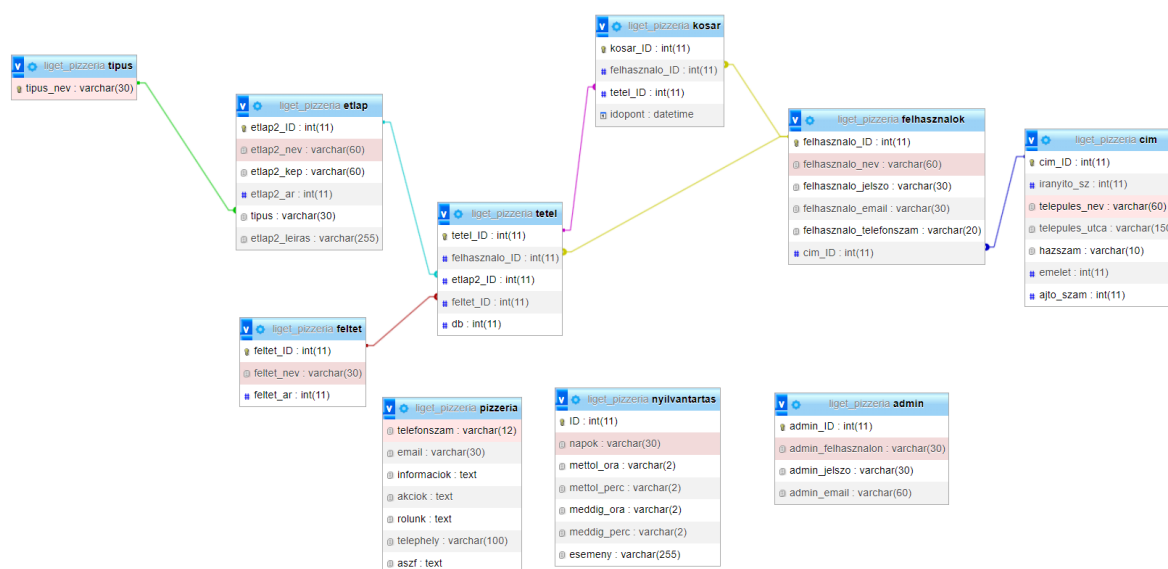
A kép közepén a (Frontend) kliens oldalon megjelenő regisztrációs felületet láthatjuk. A "szövegdobozok", szövegek és a gombot a kép bal oldalán: html nyelvben valósítottunk meg, míg ezeknek a méretét, pozicionálását, kinézetét a kép jobb oldalán: css nyelvben definiáltuk.



Összességében a frontend szerepe az, hogy gondoskodjon arról, hogy az alkalmazás felhasználói számára vonzó és könnyen kezelhető legyen, továbbá biztosítsa a kommunikációt a backenddel, hogy az alkalmazás működjön és reagáljon a felhasználói interakciókra.

9.3.3. Adatbázis

Az adatbázis egy olyan szoftverrendszer, amely strukturált adatok tárolására és kezelésére szolgál. Az adatbázis felelős az alkalmazás és weboldal adatainak tárolásáért és szervezéséért. Segít az adatok strukturált tárolásában és szervezésében. Az adatokat táblákban tárolja, és lehetőséget nyújt az adatok közötti kapcsolatok létrehozására. Lehetővé teszi az adatok lekérdezését és manipulálását.



A képen az adatbázisunk Relációs modelljét láthatjuk, amiben látjuk a táblák milyen oszlopkból állnak és mely táblák vannak kapcsolatban egymással (program:XAMPP)

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett	Megjegyzések	Extra	Művelet
1	admin_ID	int(11)			Nem	Nincs		AUTO_INCREMENT	Módosítás Eldobás Több
2	admin_felhasznalon	varchar(30)	utf8_hungarian_ci		Nem	Nincs			Módosítás Eldobás Több
3	admin_jelszo	varchar(30)	utf8_hungarian_ci		Nem	Nincs			Módosítás Eldobás Több
4	admin_email	varchar(60)	utf8_hungarian_ci		Nem	Nincs			Módosítás Eldobás Több

A képen az admin tábla Szerkezet nézetében az admin oszlopait láthatjuk. (program:XAMPP)

```
--
-- Tábla szerkezet ehhez a táblához `admin`
--

DROP TABLE IF EXISTS `admin`;
CREATE TABLE `admin` (
  `admin_ID` int(11) NOT NULL,
  `admin_felhasznalon` varchar(30) NOT NULL,
  `admin_jelszo` varchar(30) NOT NULL,
  `admin_email` varchar(60) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_hungarian_ci;

--
-- A tábla adatainak kiíratása `admin`
--

INSERT INTO `admin` (`admin_ID`, `admin_felhasznalon`, `admin_jelszo`, `admin_email`) VALUES
(1, 'admin', '1234', 'szanyi@szanyi.com');

-----
```

A képen az *admin* tábla létrehozását és annak oszlopainak definiálását láthatjuk SQL nyelven (programm Visual Studio Code)

Az adatbázisunk tehát kulcsfontosságú szerepet játszik az alkalmazásunk és weboldalunk adatainak tárolására, kezelésére és azoknak az elérhetőségére, és nélkülözhetetlen az alkalmazásban feltöltött adatok weben való megjelenítésére és azoknak megőrzésére.

9.4. Weboldalt alkotó nyelvek

Html

- A HTML (HyperText Markup Language) egy szabványosított nyelv a weboldalak strukturálására és tartalmának megjelenítésére. A HTML felelős azért, hogy az interneten megjelenített weboldalak tartalmát strukturálja és formázza úgy, hogy azokat a böngészők megfelelően tudják megjeleníteni.

Css

- A CSS (Cascading Style Sheets) egy stílusleíró nyelv a weboldalak megjelenítésének formázására és stílusának meghatározására. A CSS felelős azért, hogy az HTML-ben megadott tartalmat vizuálisan formázza és megjelenítse a böngészőben.

JavaScript

- A JavaScript az egy programozási nyelv, amelyet általában webfejlesztésre használnak, sokoldalú, amely szerintem kulcsfontosságú szerepet játszik a webfejlesztésben és a modern webes alkalmazások készítésében.

JSON

- A JSON (JavaScript Object Notation) egy adatsere formátum, az API válaszadását könnyen kezeli, ezért választottam.

SQL

- Az SQL (Structured Query Language) adatbáziskezelő nyelv, amelyet relációs adatbázis-kezelő rendszerekkel (mint például MySQL) kommunikáláshoz használnak.

9.5. Weboldal terve röviden

A weboldal áll 5 lapból: Fő oldal (index.html), bejelentkezés (bejelentkezés.html), regisztráció (regisztracio.html), Étlap (etlap.html), Rendelés (rendeles.html).

- A főoldalon az általunk kitalált vagy rokonság által érdekeltségből szerzett Liget Pizza adatait láthatjuk elrendezve, aminek a díszítését az index.css file-ban ‘készítettük’.
- A cím sorokon és gombokon kívül szinte minden szöveges adatot az adatbázisból olvassunk le, amit az alkalmazásunkkal felhasználó barát módon, ‘laikus’ ember is képes megváltoztatni, törölni vagy új szöveget alkotni.
- A regisztrációs ablakban az adatbázisba fel tudunk vinni egy felhasználót, akinek hozzáférést biztosítunk a rendelés ablakhoz, miután a Bejelentkezés ablakban bejelentkezett az általa korábban megadott belépési adatokkal.
- Az étlap ablakban az adatbázisból lekérdezett termékek képét nevét és árát láthatjuk, amiket hozzá tudjuk adni egy kosárhoz, aminek tételeit tudjuk növelni vagy törölni.
- A kosár a rendelés ablakban található.
- Kijelentkezett vagy be-nem jelentkezett állapotban csak a főoldalhoz, regisztrációs oldalhoz és a bejelentkezés oldalhoz fér hozzá a felhasználó.
- Bejelentkezett állapotban egy fülön módosíthatók a bejelentkezett felhasználó adatai, amit a regisztráció során megadott.
- Az ablakok között gombokkal tudunk navigálni.

Forrásmegjelölés

<https://boxicons.com/usage> —> Ikonok és annak képei

```
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap');
```

<https://fonts.google.com/selection/embed> —> (Css) karakter formázásra használt

https://www.facebook.com/ligetpizza/?locale=hu_HU

<https://www.svgrepo.com/svg/506663/home>

<https://www.svgrepo.com/svg/521567/close-square>

<https://www.svgrepo.com/svg/442668/info-rectangle>

<https://www.svgrepo.com/svg/509190/picture>

<https://www.svgrepo.com/svg/521971/arrow-right-square>

<https://www.svgrepo.com/svg/521961/arrow-left-square>

<https://boxicons.com/usage>

<https://learn.microsoft.com/en-us/dotnet/desktop/wpf/controls/how-to-use-the-image-element?view=netframeworkdesktop-4.8>

<https://fonts.google.com/selection/embed>

<https://www.youtube.com/watch?v=zi3tKRp0fIY>