Tartalom

[*Bemutatás* 2](#_Toc132287471)

[Specifikációk 4](#_Toc132287472)

[Rendszerkövetelmények 5](#_Toc132287473)

[Felhasználói kézikönyv 6](#_Toc132287474)

[Program használata 8](#_Toc132287475)

[Bevezető 8](#_Toc132287476)

[Regisztráció 8](#_Toc132287477)

[Navigálás 9](#_Toc132287478)

[Bejelentkezés 9](#_Toc132287479)

[Főoldal 9](#_Toc132287480)

[Étlap 10](#_Toc132287481)

[Rendelés 10](#_Toc132287482)

[Fizetés 10](#_Toc132287483)

[Kosár 11](#_Toc132287484)

[Fejlesztői dokumentáció 11](#_Toc132287485)

[Témaválasztás 11](#_Toc132287486)

[Fejlesztői eszközök 12](#_Toc132287487)

[Tervezés 13](#_Toc132287488)

[Adatmodell 16](#_Toc132287489)

[Feladatspecifikációk 19](#_Toc132287490)

[Tesztelés 26](#_Toc132287491)

[1. Regisztrációs űrlap 26](#_Toc132287492)

[2. Rendelési űrlap 26](#_Toc132287493)

[3. Étlap kiíratása 26](#_Toc132287494)

[4. Adatbázis kapcsolat lezárása 27](#_Toc132287495)

[4. SQL query 27](#_Toc132287496)

[Továbbfejlesztési lehetőségek 28](#_Toc132287497)

[Összegzés 30](#_Toc132287498)

[Forrásmegjelölés 31](#_Toc132287499)

[Köszönetnyilvánítás 31](#_Toc132287500)

# *Bemutatás*

A FastFoodKing fejlesztői vagyunk, Hegedüs Zsolt, Raffai Ferenc és Kisfali Krisztián. A web alkalmazásunk célja, hogy a felhasználók egyszerűbben tudjanak ételt rendelni. Az ötlet inspirációja a Wolt alkalmazásból származik.

Alkalmazásunk időszerűségét alátámasztja az, hogy az emberek jelentős része nagyon szeretnek ételeket rendelni. Ezért hoztuk létre ezt weboldalt. Sok hasonló oldal található ételrendelésre, amik bonyolultabb megoldásokkal rendelkeznek, ezáltal megnehezítik a rendelést. Mi egy sokkal egyszerűbb, dinamikusabb ötlettel álltunk elő. A weboldalunk felhasználó barátibb, jobban átlátható, egyszerű dizájnnal készült, de annál hatékonyabb.

Az oldalunkat, egy egyszerű regisztrálás után bárki könnyen tudja kezelni. A felhasználó profil résznél megadja az elérhetőségeit, amit ezek után a rendszer az adatbázisból, bármikor elér, így nem kell többet telefonálni, hanem csak rendelnie kell az ételt. Az elérhetőség adatai bármikor módosíthatók.

A fizetésre egy praktikus ötlettel álltunk elő. Egy virtuális pénzt programoztunk le, amit megtud venni fizikai fizető eszközzel. Ez miért jó? Esetleg, ha van gyermeke, aki tegyük fel kollégiumba lakik, akkor tud neki creditet venni, ami úgy működik, mint egy virtuális pénztárca, amit csak nálunk tud felhasználni. Ezért a szülőknek nem kell attól félniük, hogy a gyermekük másra költi el a pénzt, amit kapott.

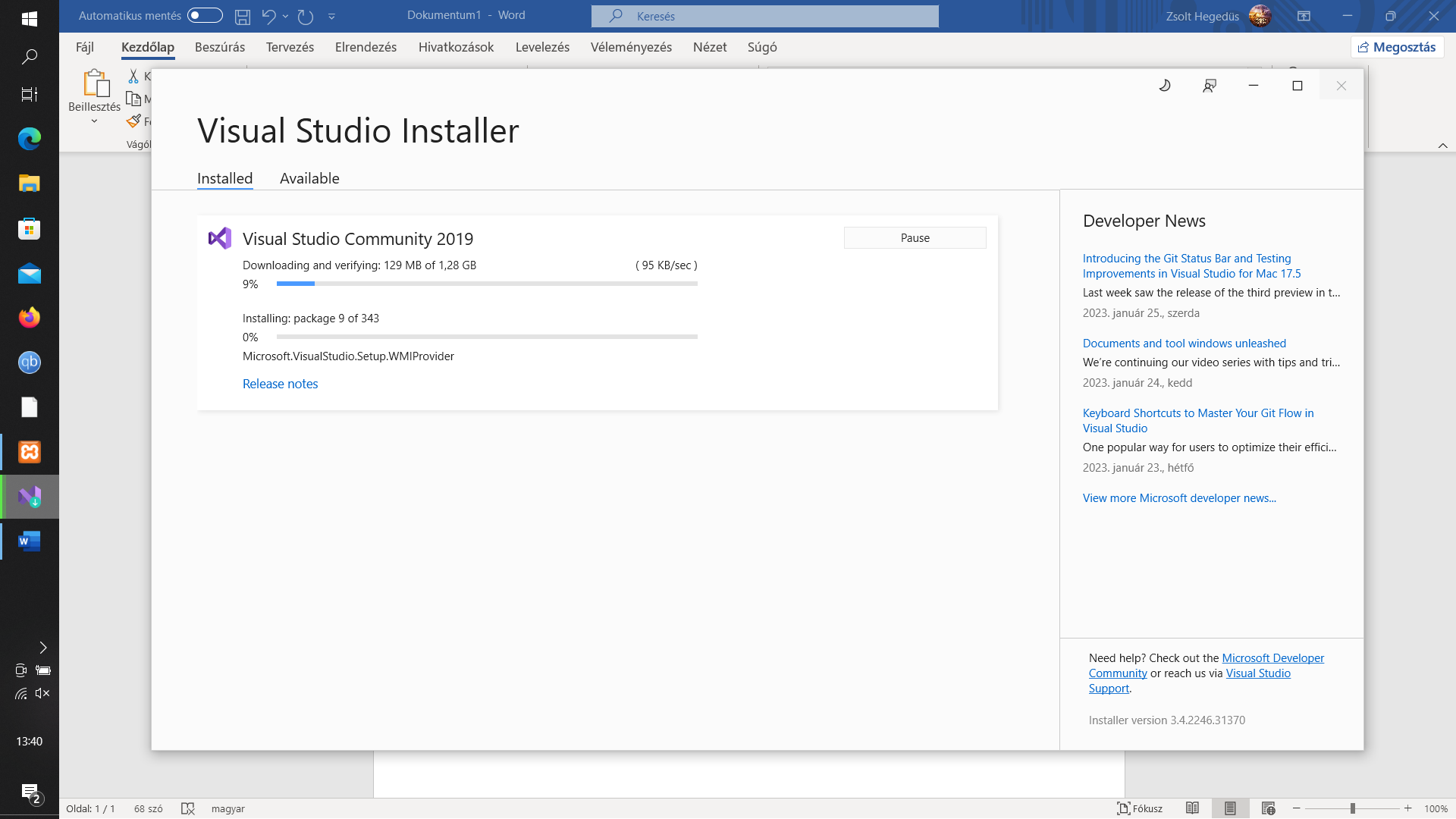
A projektmunka alatt, számos problémával szembesültünk, mint csapatszinten és egyénileg is egyaránt. Az egyéni problémákat elég hamar tudtuk orvosolni, saját kutatások, önfejlesztések útján. Csapatként az elején több akadályba is ütköztünk, mivel nem voltunk még össze szokva, de ahogy haladtunk a projektmunkával a csapatmunkánk nagyon sokat változott. Számos dolgot tudtunk egymástól elsajátítani, illetve sok egyéni problémában is tudtunk egymásnak segíteni az idő múlásával.

Új ismeretek elsajátítása, főként backend oldalon történt. Mivel mi főként frontend részről voltunk jobbak, ezért a backend rész igazából új dolog volt számunkra. Új ismeretek terén megtanultuk, hogyan lehet az adatbázist összekötni a weboldallal. Új dolog volt még a GitHub, mivel előtte nem sokat foglalkoztunk vele.

Feladat egyes részterületinél is inkább a csapatmunka volt a jellemző. Ugyanakkor célszerűnek tartottuk, hogy 1-1 témarészben 1-1 csapattag mélyüljön el jobban.

Ezek alapján: Hegedüs Zsolt főként a frontend-el foglalkozott, de egy két dologba a backendbe is segített. Kisfali Krisztián készítette az adatbázist, illetve frontend oldalon volt még aktív. Raffai Ferenc a backand részétnél munkálkodott valamint az adatbázisban vett még részt aktívabban. A dokumentációt együtt írtuk meg, mindenki a saját részterületénél volt hangsúlyosabb ebben a munkában.

A projektfeladat alatt használt eszközeink közé tartozott a visual studio code. Kezdetben javát, illetve netbeanst használtunk a backend résznél, de azzal sok gond adódott, ezért úgy döntöttünk, hogy visual studio-ra és c#-ra váltunk a számunkra jobb kezelhetőség, illetve a több segítség miatt, viszont a c# se akart rendes működni ebből adódóan úgy döntöttünk, hogy php-ba folytatjuk a munkálatokat. Adatbázis terén mySql workbench-et használtunk mivel átláthatóbb és egyszerűbb, mint a többi adatbázis kezelő, amit ismerünk.

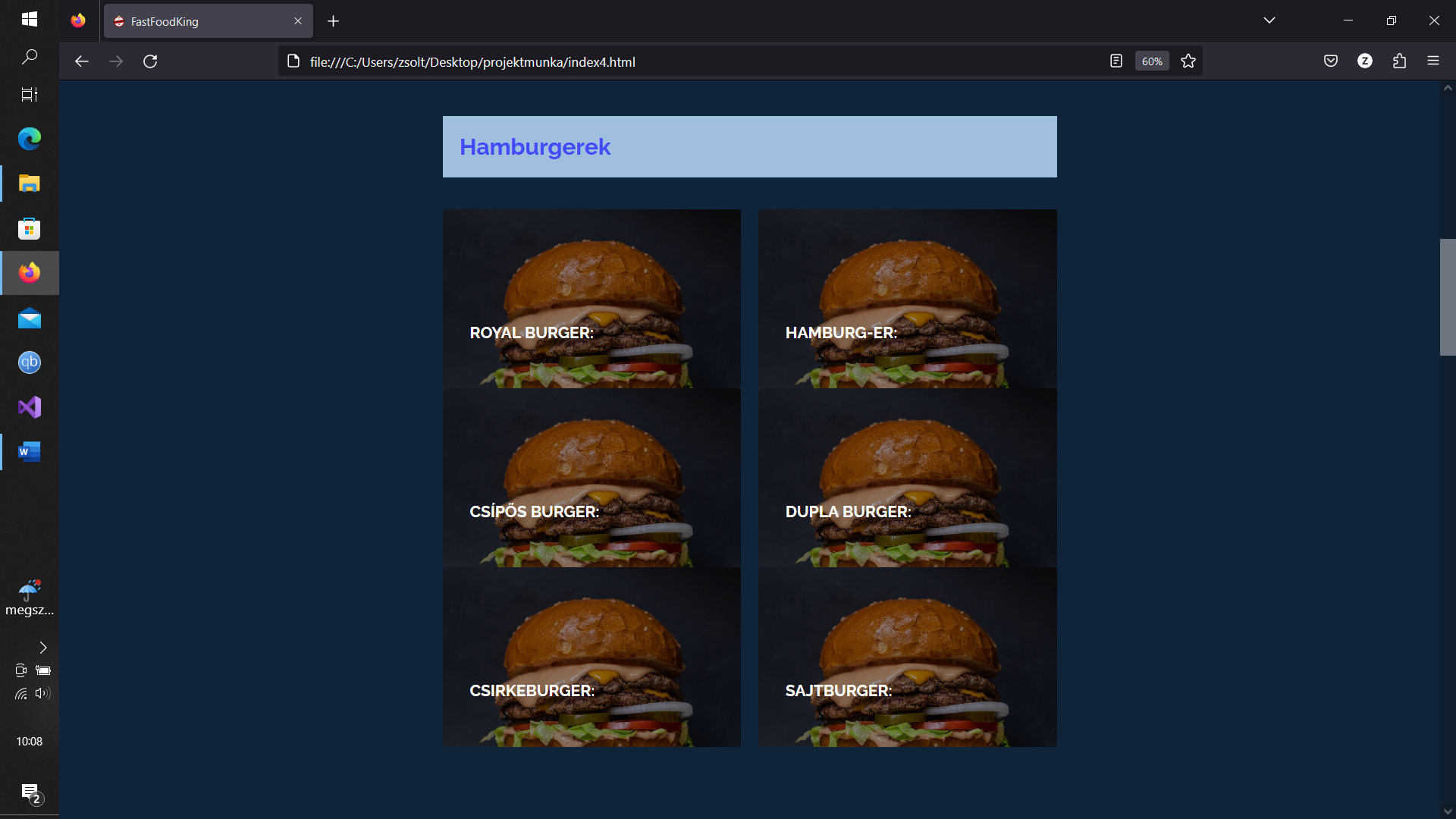


1. ábra:Munkafolyamat

A képen *(1. kép)* az látható, ahogy töltődik le a vizuális stúdió 2019, mert a 2022 annyira új, hogy amit hozzá csináltak alkalmazást az nem működik.

# Specifikációk

Ahogyan a bemutatóban már leírtuk, a projekt egy ételrendelő alkalmazás. A specifikációit tekintve, ebben az alkalmazásban a felhasználó saját profilt tud létre hozni, ahol meg tudja adni az általános adatait, ezzel megkönnyítve számára az online rendelést. Sokkal letisztultabb és egyszerűbb használatot biztosít a vásárló számára. Specifikációk tekintetében a felhasználó egy átláthatóbb étlapot kapott.



2. ábra: Étlap

A képen *(2. kép)* a hamburgerek menü pont látható. Itt látható, hogy az összes étel egy pont alá van véve. Ezek a képek kártyaként működnek, amire, ha a vásárló rá megy egérrel, akkor az megfordul és láttja, hogy mik az összetevői az ételnek.

A fizetéshez virtuális egyenleget hoztunk létre, amit a felhasználó felt tud tölteni valós fizetőeszközzel. Ettől kezdve ezzel a virtuális pénztárcával tud fizetni az alkalmazásunkba. Ez azért lehet egyszerűbb a vásárlónak, mert ezt bármikor fel tudja használni, függetlenül attól, hogy van-e a kártyáján pénz vagy nincsen. Illetve, ha egy szülőnek van egy kollégista gyermeke és pénz helyett ilyen kreditet venne a gyermeknek, akkor tudja, hogy nem másra fogja elkölteni a pénzt, és a szülő is látja, hogy mennyi van még a gyerekének.

Mivel kisebb a vásárlói kör, ezért gyorsabb kiszállítással tudunk szolgálni. Az ételek árán kívül más kezelési, illetve szállítási költséget nem számolunk fel, ezzel bíztatva a vásárlókat arra, hogy a mi szolgáltatásunkat válasszák.

Az oldalunk célja, hogy a vásárlók egy megbízható szolgáltatást kapjanak a pénzükért.  
A vásárlóink adatait bizalmasan kezeljük, a jelszavak biztonságban vannak nálunk az elérhetőségek, illetve az adatok semmilyen körülmények között nem kerülnek nyilvánosságra.

## Rendszerkövetelmények

A webalkalmazásunk futtatására bármilyen Androidos, illetve bármilyen Apple termék képes. Mivel a weboldal reszponzívan épült fel, hogy telefont jobban preferálók is ugyan olyan élményben részesüljenek mind a pc-s felhasználók. Számítógépen a rendszerkövetelmény Windows 7-től fölfele bármilyen eszköz, amin fut valamilyen böngésző képes futtatni az oldalunkat. Ezen kívül az oldalunkhoz nem szükséges más egyéb követelmény.

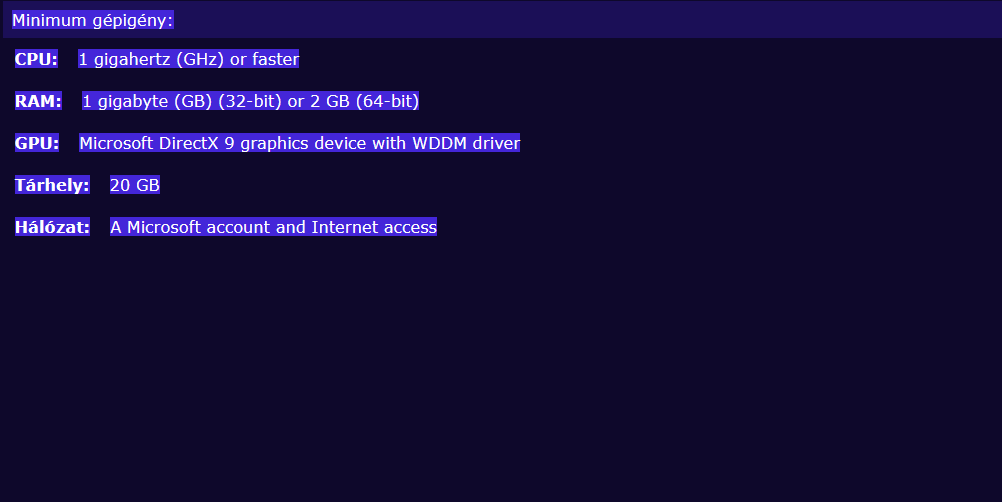
Processzor: Minimum ajánlott két magos processzor a számítógépbe, mert az egyes weboldalakhoz magasabb számítás szükséges. Mivel a mostani operációs rendszerek legtöbbje már nem működik két magos processzor nélkül, ebből adódóan sajnos már a mi webalkalmazásunk is ezt igényli.

Memória: A webalkalmazások általában adatokat tárolnak, ebből adódóan legalább már 4 gb rammra szükség van, hogy a weboldalunk akadozás nélkül működjön.

Tárhely: A webalkalmazásoknak szükségük van tárhelyre, hogy tárolják az adatokat és a felhasználói fájlokat. Az alkalmazás típusától függően a tárhelyméret változhat.

Hálózati kapcsolat: A webalkalmazásokhoz szükség van egy megbízható hálózati kapcsolatra, hogy a felhasználók által küldött adatok és információk gyorsan és hatékonyan továbbítójának.

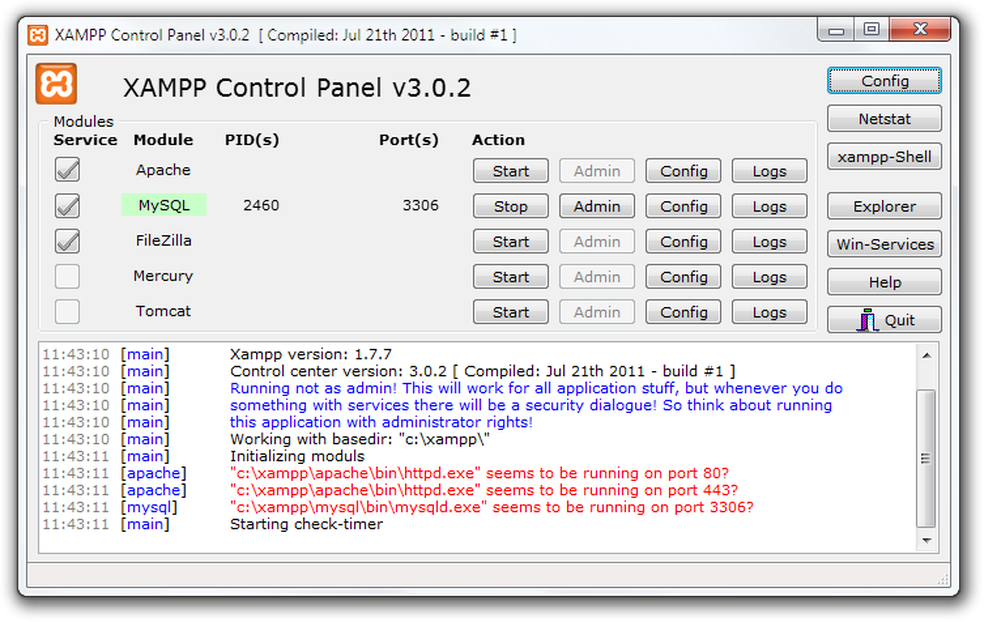
Böngészőkompatibilitás: A webalkalmazásoknak többféle böngészővel kompatibilisnek kell lenniük, hogy a felhasználók bármilyen eszközről használhassák őket.



3. ábra: Rendszerkövetelmény

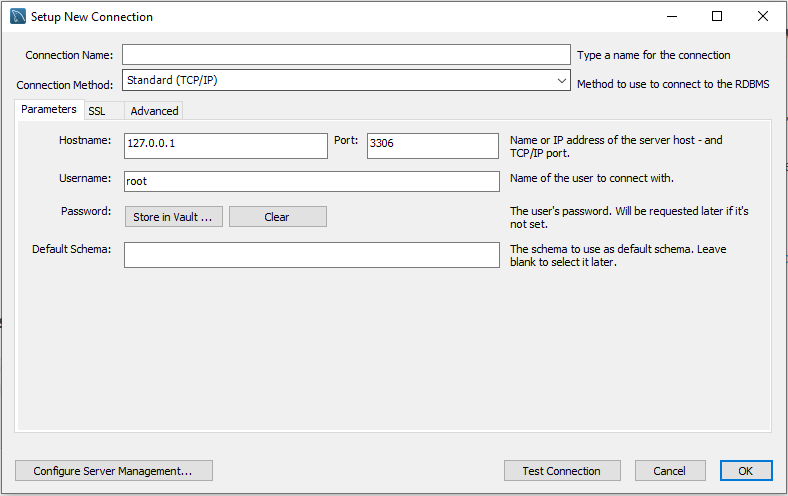
# Felhasználói kézikönyv

1. Az első lépés a mysql alkalmazás telepítése lesz. A myspl workbench alkalmazás telepítőre kétszer rá kattint a felhasználó. Itt meg fog nyílni a telepítő ablaka. A felhasználó rá nyom a next gombra itt feljön a telepítőmappa kiválasztása, mi azt ajánljuk, amit alapból ajánl a gép. Ezután ismét a next gombra megy. A következő ablakba feljön, hogy a complete vagy custom telepítést szeretne. Ha a felhasználó a complete lehetőséget használja, akkor a programhoz lévő összes tulajdonságot beállít magától, míg a custom lehetőségnél a felhasználónak kell mindent beállítania, illetve neki kell személyre szabnia az egész alkalmazást. Mi a program hibátlan működése érdekében a complete lehetőséget ajánljuk. A felhasználó eldöntötte, hogy melyiket választja akkor itt rá nyom a next gombra, ami után feljön az install lehetőség erre rá kattintva az alkalmazás mindent feltelepít már magától.
2. A második lépés a xampp telepítése lesz. Itt a felhasználó ugyan úgy rákattint a telepítőre, ami, ha megnyílt szintén ki lehet választani, hogy hova szeretné telepíteni a felhasználó itt is az ajánlott telepítési útvonalat válassza ki, ezután next gomb, majd install és az alkalmazás magától feltelepít mindent. Ezután a felhasználó a Fastfoodking nevű mappát a c meghajtón lévő xampp mappán belüli htdocs nevű mappába. Ahhoz, hogy az oldal működjön a xampp alkalmazást rendszergazdaként elkell indítani, amit a felhasználó úgy tud meg csinálni, hogy a windows keresőbe beírja, hogy xampp jobb egérgombbal rákattintunk és a lehetőségek közül ki választjuk, hogy futtatás rendszergazda ként. Ezután feljön egy ablak, ahol az apache és a mysql lehetőséget ki kell pipálni és ezután rá kell nyomni a start gombra (4.*kép*).



4. ábra: Xampp

1. Ezután elkell indítani a mysql workbench: Ezen belül lesz egy olyan rész, hogy mysql connections, itt mellet lesz plusz jel, arra rákattint a felhasználó, akkor bejön egy olyan ablak, hogy setup new connection. A connection névnél meg kell adni, hogy localhost a többi beállítást úgy kell hagyni, ahogy bevan állítva majd rányomni, hogy ok így elkészült a kapcsolat. Ezután rá kell nyomni a localhost ablakra, itt rá kell menni a file nevű funkcióra, ezen belül a run spl script, ha ezt meg nyitja a felhasználó akkor feljön a fájlkezelő, itt ki kell a választani a FastFoodKing mappából a projekt nevű sql fájl, majd a visible szavakat ki kell törölni, a villám jellel lefuttatjuk és kész az adatbázis beolvasás. Ezutan a schema résznél jobb egérgomb refresh és ott lesz az adatbázis.



5. ábra: MYSQL

1. Negyedik lépés, ha nincs böngésző a gépen, akkor a google böngésző telepítőt futtatod a cd-ről. Itt feljön az ablak ugyan úgy, mint a többi telepítőnél. Kiválaszthatod, hogy te hova szeretnéd telepíteni a böngészőt, vagy hagyod az alapbeállításon. Ez után a next gombra kattintva előjön a telepítés funkció, ha a telepítés elkészült, akkor rá kell nyomni kétszer a google képre. Ha elindult a böngésző, akkor a keresőbe be kell írni, hogy localhost/FastFoodKing és már működik is az oldal.

# Program használata

## Bevezető

Az ételrendelős weboldalunk azért jött létre, hogy megkönnyítsük az emberek életét, és lehetővé tegyük, hogy kényelmesen rendelhessenek otthoni ételt. Az oldalunkon megtalálhatóak a legjobb ételek, ízek és ételkülönlegességek, amelyek kielégítik az ízléseket és igényeket. Az oldalunk használata rendkívül egyszerű és gyors, és minden egyes részletet leírunk, hogy a felhasználók könnyedén tudjanak navigálni az oldalon. A navigációt egy menüpontokból álló rész segíti a felhasználónak. Ezek a menüpontok a regisztráció, ahol a felhasználó pár egyszerű lépéssel betud regisztrálni. Van a főoldal, ahol az kínálatunkat láthatja a felhasználó, van a bejelentkezés, ahol a felhasználó a korábbi adataival be tud jelentkezni, van a rendelés oldal, ahol a vásárló meg tudja adni, hogy miket szeretne rendelni és van az étlap, ahol az összes kapható ételünk látszódik és a leírása is, hogy miből áll az étel, illetve van még a kosár oldal.

## Regisztráció

Az ételrendelés előtt minden felhasználónak regisztrálnia kell az oldalon. A regisztrációs folyamat gyors és egyszerű, és minden szükséges információt tartalmaz, mint például a nevét, e-mail címét a jelszavát, a várost, ahol él, az irányítószámot, az utcát és a házszámot, hogy a dolgozók tudják, hogy hova kell szállítani a rendelést. A regisztráció során a felhasználó a lehető legpontosabban adja meg az adatait, mert később a bejelentkezésnél adódhatnak problémák ebből. Főként a lakcím megadásánál, illetve a jelszó megadásánál kell figyelni a legjobban.

6. ábra: Regisztráció

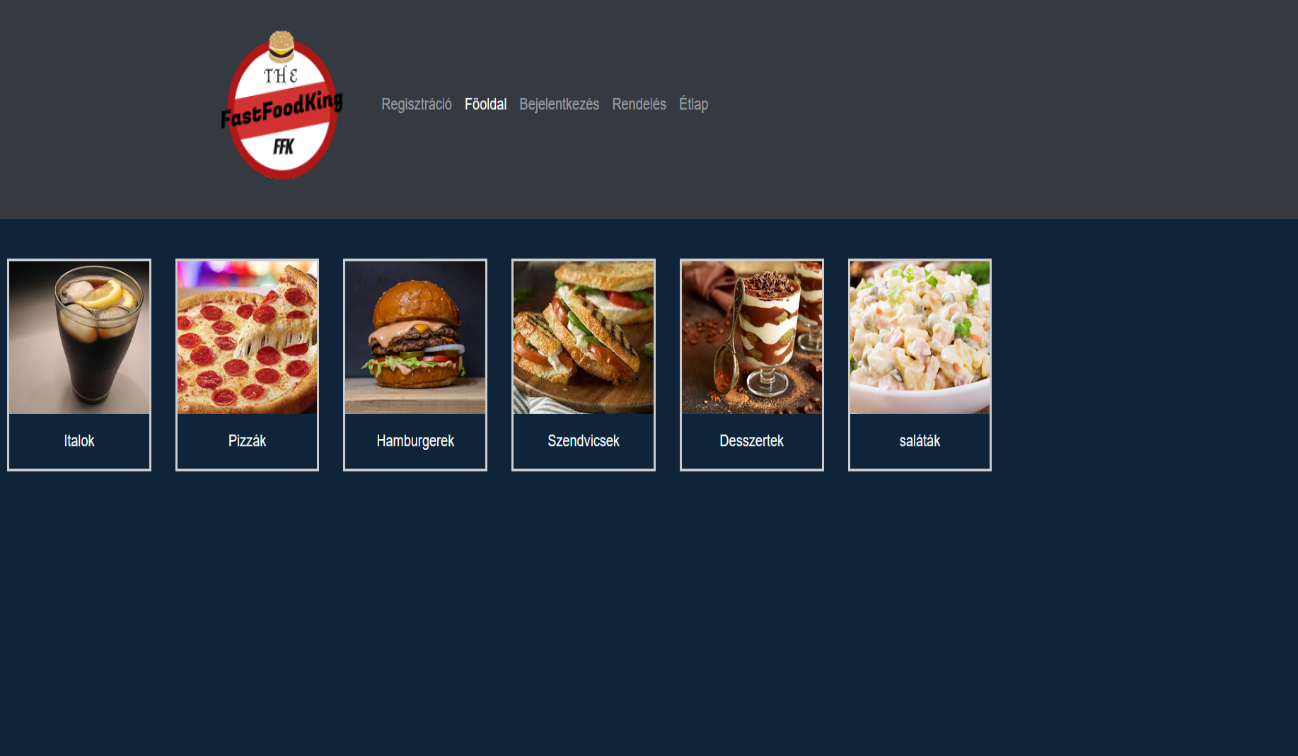
Navigálás

Az oldalunkon a felhasználók, egy navbar segítségével eltudnak jutni, az oldal bármelyik pontjára ezzel is könnyítve a felhasználói élmény. A navbaron öt oldal elérhet, van a regisztráció, a bejelentkezés, a főoldal, a rendelés és az étlap rész. Bármelyik oldalra rá nyom a vásárló akkor arra az oldalra foglya átnavigálni a weboldal.

# Bejelentkezés

A vásárló a regisztráció során a felhasználó nevével és jelszójával könnyedén be tud már jelentkezni az oldalra és innentől kezdve bármikor tud rendelni amikor csak szeretne.

## Főoldal

A főoldalon látható az összes kapható termékfajta. Ezek egy kisképek, amik keretben vannak. Ezek a képek úgy működnek, hogy ha a felhasználó rá kattint bármelyik képre, ami neki tetszik akkor az átdobja őt az étlap oldalra, pont ahhoz a ponthoz, amit választott.

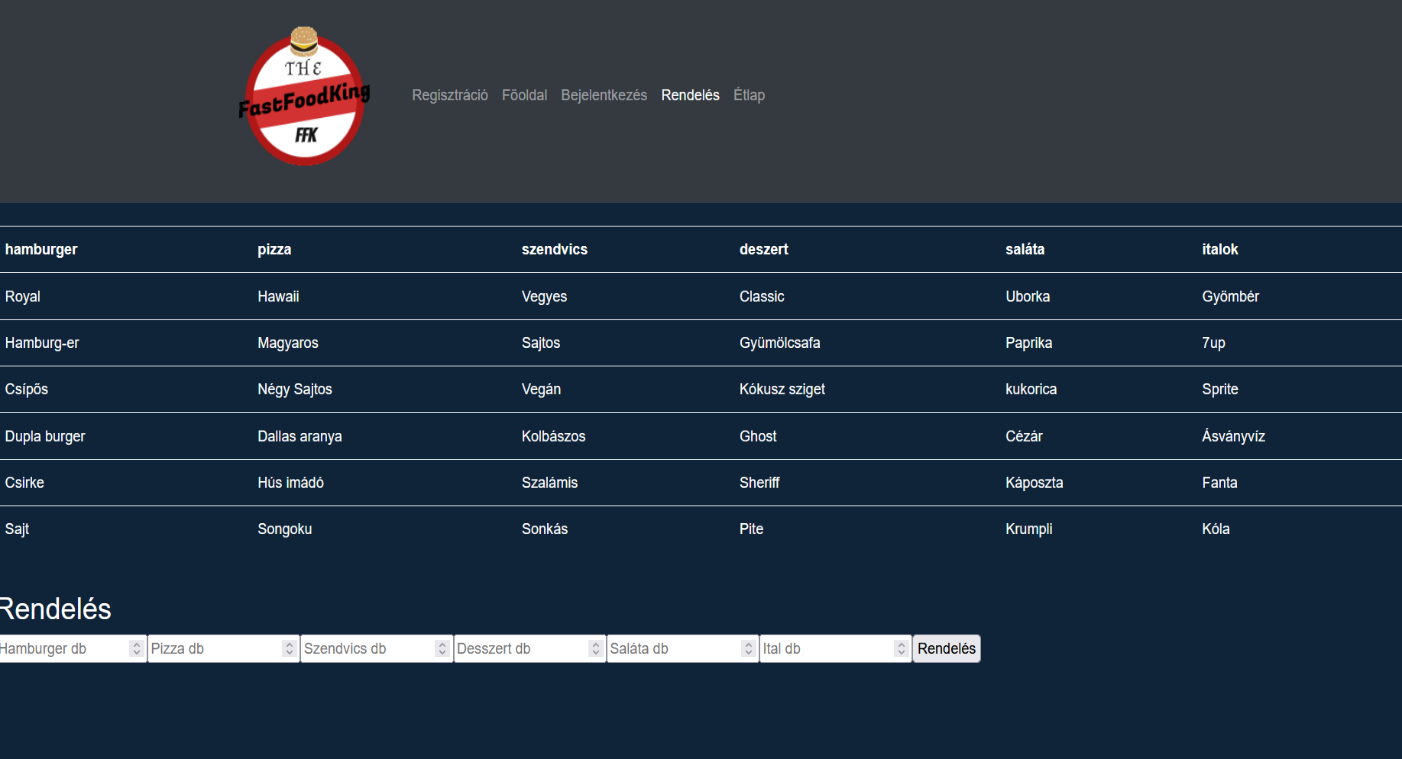
7. ábra: Főoldal

## Étlap

Az étlapon az összes étel külön pontokra van szedve. Az termékek kártya ként működnek. Az első oldalán az étel maga látható, ha a felhasználó rá viszi az egerét akkor feljön, hogy miből áll az étel maga. Az ételeink a következőkből állnak. Van hamburger, pizza, szendvicsek, desszertek, saláták. és italok.

## Rendelés

Az étel rendelése egyszerű folyamat, amelyet pár kattintással el lehet végezni. Az oldalon kiválaszthatja a kívánt ételeket és mennyiséget. A rendelés oldalon egy táblázatban össze van szedve, hogy miket lehet rendelni. Ezután meg tudja adni, hogy melyikből melyiket szeretné majd a rendelés gombra rá kattintva a felhasználót át küldi a kosár oldalra, ahol összegezve van a rendelése és hogy mennyit kell majd fizetnie.

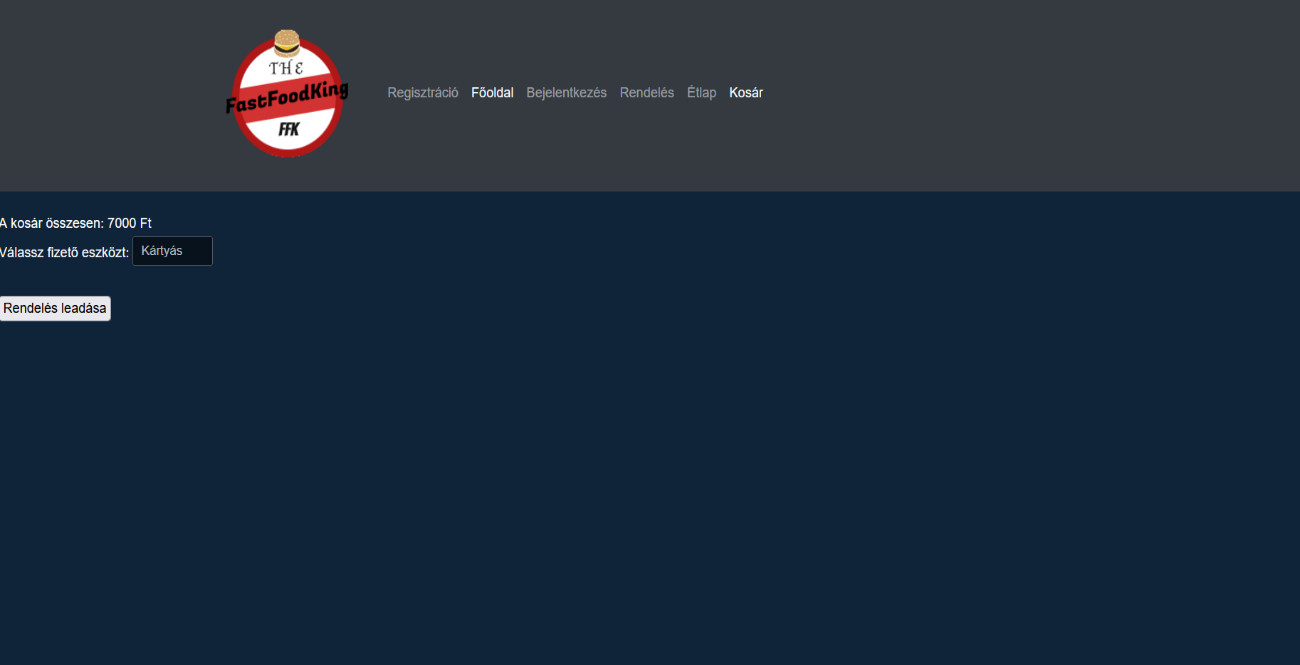


8. ábra: Rendelés

## Fizetés

A fizetés jelenleg úgy működik, hogy a vásárló háznál tud fizetni készpénzzel, vagy bankártyával.

## Kosár

 A felhasználó, amint rá ment a rendelés gombra, át lesz irányítva a kosár oldalra, ahol összegezve lesz, hogy mennyit kell fizetnie, illetve, hogy kártyásan vagy készpénzzel szeretne fizetni a futárnak. Ezután rá nyom a submit gombra és a rendelése véglegesítve lesz.

9. ábra: Kosár

# Fejlesztői dokumentáció

## Témaválasztás

A választott téma azért volt fontos számunkra, mert az emberek nagy része nap mint nap fogyaszt gyorséttermi ételeket. A gyorséttermi ételek és italok különösen elterjedtek a világban, széles körben elérhetők a közösségi médiában, az online fórumokon és más platformokon. Az online ételrendelés egyre népszerűbbé válik, mivel lehetővé teszi az emberek számára, hogy kényelmesen és könnyedén rendeljenek ételt a telefonjukon vagy a számítógépükön keresztül. A téma fontosságát növeli még az is az is, hogy az online ételrendelés és az ételfutárszolgálatok piaca rendkívül dinamikus és fejlődőképes a technológiai újítások további lehetőségeket nyújtanak a szolgáltatás javítására és a vásárlói élmény növelésére.

A felhasználók szeretik az egyszerűséget, a kényelmet és a gyorsaságot, amelyet az online rendelés nyújt számukra. Az éttermek számára pedig az online rendelési rendszerek lehetőséget biztosítanak a bevétel növelésére, a folyamatok hatékonyságának javítására és a vásárlói elégedettség növelésére.

A különböző online ételrendelő platformok és szoftverek alkalmazása egyre elterjedtebbé válik, azonban a saját, egyedi online rendelési rendszer kialakítása lehetőséget ad az éttermeknek a személyre szabásra, valamint az üzleti folyamatok teljes körű ellenőrzésére. Ezért fontos, hogy az éttermek olyan online rendelési rendszert használjanak, amely alkalmas a saját igényeiknek és szolgáltatásaiknak.

Az online ételrendelési rendszer fejlesztése azonban nem könnyű feladat, és szakértelemmel kell rendelkezni ahhoz, hogy hatékonyan működjön. Fontos, hogy az adatok biztonságosan kezeljék, a rendszer felhasználóbarát és könnyen kezelhető legyen, valamint képes legyen integrálódni más szoftverekkel, mint például a pénztárgépekkel vagy a raktárkezelő rendszerekkel.

Összefoglalva, azért döntöttünk úgy, hogy egy ételrendelő webalkalmazást hozunk létre, mert a mai társadalomban nagy az az igény az ilyen fajta weboldalakra, illetve alkalmazásokra. Az online ételrendelési rendszerek lehetőséget biztosítanak az éttermeknek a hatékonyság növelésére és a bevétel növelésére, valamint az ügyfelek számára a kényelem és a gyorsaság élményét nyújtják.

## Fejlesztői eszközök

1. A Visual Studio Code egy nyílt forrású, szabadon letölthető és keresztplatformos kódszerkesztő eszköz, amelyet a Microsoft fejlesztett ki. Az eszköz több programozási nyelvet támogat, és számos kiegészítővel bővíthető. Főként ez volt az oka annak, hogy ezt az alkalmazást használtuk a kódolásra.
2. A MySQL egy nyílt forrású, relációs adatbázis-kezelő rendszer, amelyet a Oracle fejlesztett ki. Az eszköz széles körben használt webes alkalmazások, szerveralkalmazások és egyéb programok által adatok tárolására. Mivel a MySQL az egyik legelterjedtebb adatbázis kezelő program és ehhez sok segítséget találtunk az interneten, főként ezért választottuk ezt a programot.
3. Mivel saját logót akartunk az oldalhoz ezért számos ingyenes online logó tervezőt használtunk. A logó megtervezéséhez a Shopify nevű oldalt használtuk. Végül az Adobe photoshopban szerkesztettük meg a hátterét, hogy át látszó legyen.
4. Az XAMPP egy ingyenesen letölthető, keresztplatformos szoftvercsomag, amely magában foglalja az Apache webszervert, a MySQL adatbázis-kezelőt, valamint a PHP és a Perl programozási nyelveket. Az XAMPP segítségével lehetőség nyílik a helyi számítógépen történő weboldal fejlesztésre és tesztelésre. A php és az adatbázis ennek köszönhetően tud hibátlanul futni és ez köti össze az adatbázist a php-al, éppen ezért választottuk ezt a programot, illetve ehhez találtunk a legtöbb segítséget.

10. ábra: Xampp2

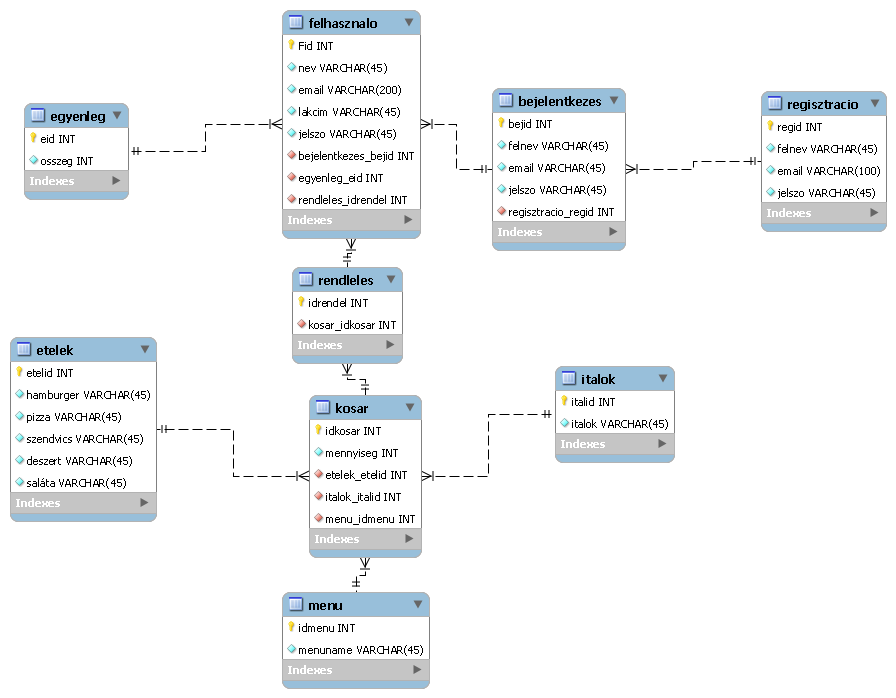
1. A PHP egy nyílt forráskódú, szerveroldali programozási nyelv, amely dinamikus weboldalak fejlesztésére használható. A PHP nyelv nagyon könnyen tanulható és nagyon széles körben elterjedt a webfejlesztők között. Az elején javával és c#-al próbáltuk meg csinálni a backend részt, de ott sok hibába futottunk és végül így kezdtünk le php-ba dolgozni, mert sokkal egyszerűbben ment, mint a többibe.

# Tervezés

A tervezés során az elején a csapat azon gondolkozott, hogy milyen fajta irányba vigyük a fejlesztést, milyen programokat használjunk, mi legyen a témája a projektnek és milyen fajta alkalmazást szeretnék csinálni. Amint a téma és az, hogy milyen alkalmazást csináljunk meglett, elkezdtük ki dolgozni az erdiagram plus segítségével az adatbázistervezést, ez alatt közbe megbeszéltük az alap funkciókat, amiket meg akartunk csinálni elsődlegesen, ilyen volt például a regisztráció, bejelentkezés és a rendelés. Majd ez után következett a frontend dizájnjának megbeszélése, hogy milyen oldalakat hozzunk majd létre, milyenek legyenek a szín kombinációk. Majd jött az OOP megtervezése, hogy milyen osztályokat szeretnénk létre hozni.

Az alkalmazás a következő főbb osztályokat tartalmazza:

* Felhasználó: Ez az osztály a felhasználókat reprezentálja az alkalmazásban. Az osztály tartalmazza a felhasználó azonosítóját, nevét, e-mail címét, jelszavát stb.
* Termék: Ez az osztály a termékeket reprezentálja az alkalmazásban. Az osztály tartalmazza a termék azonosítóját, nevét, hogy milyen termékeink vannak stb.
* Kosár: Ez az osztály a kosarat reprezentálja az alkalmazásban. Az osztály tartalmazza a kosár azonosítóját, a kosárba helyezett termékek listáját, a felhasználó azonosítóját stb.
* Rendelés: Ez az osztály a rendeléseket reprezentálja az alkalmazásban. Az osztály tartalmazza a rendelés azonosítóját, a rendelt termékek listáját, a rendelés összegét, a felhasználó azonosítóját stb.

A fejlesztés során törekedtünk arra, hogy az osztályok a tiszta kód alapján legyenek elnevezve ne legyenek össze zavarható osztály nevek. Figyeltünk arra, hogy fájlok nevei ne indexek legyenek, hanem legyenek meghatározva az adott oldal célja alapján.

11. ábra: Kezdetleges adatbázis

A (*10.kép) -en* látható a kezdetleges adatbázisunk, itt sok még a logikai buktató, ezt leegyszerűsítettük egy három táblás modellre, ahol vannak a termékek, van a kosár, ahol rendelési adatok a termék adatok és felhasználó azonosítója található, és van még egy, ami a regisztrációs tábla, ahol a felhasználó adatai szerepelnek.

Tervezési módszerként az inkrementális rendszert tudnám mondani még, mert a programot nem az elejétől terveztük meg, hanem egy alap terv alapján fejlesztés közben döntöttünk arról, hogy mi kerüljön be a webalkalmazásba. Ezzel a módszerrel, könnyedén tudtuk szűrni azt, hogy mi kerüljön bele a programba és mi nem.

Ezzel a fejlesztési módszerrel a célunk az volt, hogy minél gyorsabban és egyszerűbben tudjuk megcsinálni a programot. Illetve, hogy a számunkra fölösleges tervezési módszerekkel ne veszítsünk időt a fejlesztés során.

Azért használtuk még ezt a fejlesztést, mivel úgy gondoltuk, hogy ez a módszer lehetővé teszi számunkra az alkalmazás folyamatos fejlődését és javulását, valamint lehetővé teszi számunkra, hogy közelebb kerüljünk a végfelhasználók igényeihez.

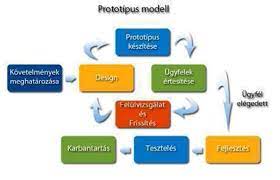
Az alkalmazás fejlesztése során az egyes funkciókat kis részekre bontottuk, és ezeket fokozatosan integráltuk a rendszerbe. Ez lehetővé tette számunkra, hogy az alkalmazás legfontosabb funkciói gyorsan elkészüljenek, és azokat az ügyfelek is használni tudják. A felhasználói visszajelzések alapján azonban folyamatosan finomítottuk és bővítettük az alkalmazást, amely így a mi igényeinknek és az ügyfelek igényeinek is megfelel.

Főként ezek az előnyök miatt, választottuk ezt a fejlesztési módszert, mert figyelembe vettük azt, hogy a későbbiekben egyszerűen tudjuk majd fejleszteni az oldalt, minden probléma és gond nélkül. Illetve több előnye van ennek a fejlesztő módszernek, mint hátrányai. A következő bekezdésben bemutatok pár előnyt, amit mi felfedeztünk a fejlesztés során. Illetve néhány hátrányt, ami még jött a fejlesztés során.



12. ábra: Fejlesztési folyamat

Előnyök:

* Gyorsabb eredményeket lehet elérni, hiszen az alkalmazás fokozatosan fejlődik és folyamatosan javításra kerül.
* Rugalmasabb, könnyebben lehet alkalmazkodni a változó ügyfél igényekhez és a piaci változásokhoz.
* Fokozottabb ügyfél elégedettség, mivel az ügyfél folyamatosan látja az eredményeket és lehetősége van visszajelzést adni.
* Kisebb kockázat a hibák és problémák kezelése során, mivel kisebb méretű, egyszerűbb részekre van bontva az alkalmazás.

13. ábra: Fejlesztési folyamat2

Hátrányok:

* Nehézkes lehet a tervezés és dokumentáció, mivel az inkrementális fejlesztés során nem feltétlenül egyértelmű, hogy milyen funkcionalitást kell implementálni a következő iterációban.
* A kód bonyolultabb lehet, mivel a különböző iterációk során implementált részek összehangolása kihívást jelenthet.
* A megfelelő tesztelés biztosítása nehéz lehet, mivel a folyamatos fejlesztés során változhatnak az alkalmazás funkciói és a hibák kezelése is változó lehet.
* Az iterációk során a szükséges funkcionalitások egyre összetettebbé válhatnak, és a későbbi iterációk időigényesebbek és költségesebbek lehetnek.

# Adatmodell

Az adatbázisban három táblát hoztunk létre az ételek táblát, a kosár táblát, és a regisztrációs táblát. Az ételek táblába létre hoztuk a termékek id-t az összes elérhető terméket. A kosár táblához hozzá kötöttük az ételek táblát ebből ki szedtük a termékek azonosítóját a kosár táblába, hogy a rendelés során az ételek azonosítóját elmentse a kosár táblába. A regisztrációs táblába megadtunk a felhasználók adatait például jelszó, felhasználó, név, lakcím.

Attribútumok:

* regid: azonosító szám típusú, elsődleges kulcs
* felnev: felhasználónév, karakter típusú
* email: email cím, karakter típusú
* jelszo: jelszó, karakter típusú
* kosarid\_kosar: azonosító, szám típusú, külső kulcs, amely kapcsolódik a kosár nevű tábla azonos nevű attribútumához
* varos: város neve, karakter típusú
* irszam: irányítószám, karakter típusú
* utca: utca neve, karakter típusú
* hazszam: házszám, karakter típusú

Kapcsolatok:

A regisztráció nevű tábla kapcsolódik a kosár nevű táblához az azonosítók alapján. Az egyik oldalon a regisztráció táblában található kosarid\_kosar attribútum, a másik oldalon pedig a kosár táblában található kosarid attribútum jelenti a kapcsolatot.

Táblanév: etelek

Attribútumok:

* etelid: azonosító, szám típusú, elsődleges kulcs
* hamburger: karakter típusú, mely jelzi, hogy a konkrét étel a hamburger kategóriába tartozik-e
* pizza: karakter típusú, mely jelzi, hogy a konkrét étel a pizza kategóriába tartozik-e
* szendvics: karakter típusú, mely jelzi, hogy a konkrét étel a szendvics kategóriába tartozik-e
* deszert: karakter típusú, mely jelzi, hogy a konkrét étel a desszert kategóriába tartozik-e
* saláta: karakter típusú, mely jelzi, hogy a konkrét étel a saláta kategóriába tartozik-e
* italok: karakter típusú, mely jelzi, hogy a konkrét étel az italok kategóriába tartozik-e

Kapcsolatok:

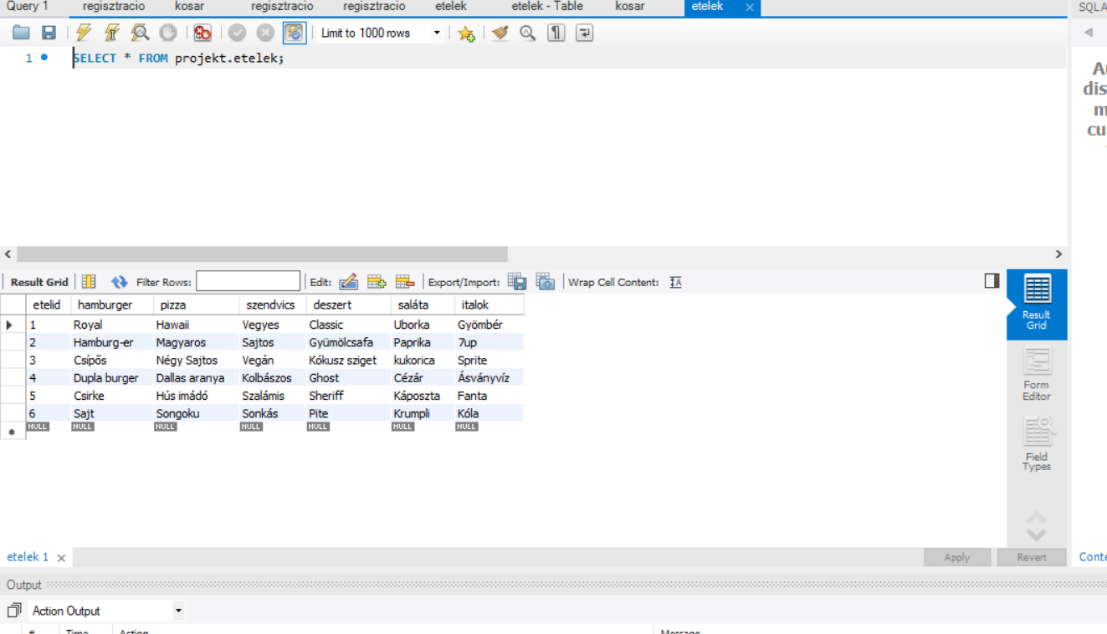
Az etelek tábla nem rendelkezik közvetlen kapcsolattal más táblákkal.

Táblanév: kosar

Attribútumok:

* idkosar: azonosító, szám típusú, elsődleges kulcs
* mennyiseg: egész szám típusú, a kosárban lévő adott étel mennyiségét jelzi etelek\_etelid: szám típusú, idegen kulcs, amely az etelek táblában lévő etelid attribútumra utal, és jelzi, hogy az adott kosárban található étel melyik étel azonosítóhoz tartozik
* hamburger: karakter típusú, mely jelzi, hogy a kosárban található adott étel a hamburger kategóriába tartozik-e
* pizza: karakter típusú, mely jelzi, hogy a kosárban található adott étel a pizza kategóriába tartozik-e
* szendvics: karakter típusú, mely jelzi, hogy a kosárban található adott étel a szendvics kategóriába tartozik-e
* deszert: karakter típusú, mely jelzi, hogy a kosárban található adott étel a desszert kategóriába tartozik-e
* saláta: karakter típusú, mely jelzi, hogy a kosárban található adott étel a saláta kategóriába tartozik-e
* ital: karakter típusú, mely jelzi, hogy a kosárban található adott étel az italok kategóriába tartozik-e regid\_regid: szám típusú, idegen kulcs, amely a regisztráció nevű táblában lévő
* regid attribútumra utal, és jelzi, hogy a kosár melyik felhasználóhoz tartozik

Kapcsolatok: Az etelek\_etelid attribútum az etelek táblához kapcsolódik, és az adott kosárban található étel azonosítójára mutat. A regid\_regid attribútum a regisztráció táblához kapcsolódik, és jelzi, hogy a kosár melyik felhasználóhoz tartozik.



14. ábra: Adatbázis

# Feladatspecifikációk

Az első fontosabb kód egy regisztrációs űrlap feldolgozását végzi el, amely a felhasználó által megadott adatokat hozzáadja egy adatbázishoz. A feladat specifikációja a következő:

Feladat leírása:

* Az űrlap a felhasználó nevét, e-mail címét, jelszavát, városát, irányítószámát, utca nevét és házszámát tartalmazza.
* Az űrlap elküldésekor a szerver oldali PHP kód ellenőrzi a bevitt adatokat a test\_input() függvény segítségével.
* Az adatok a regisztracio táblához adódnak hozzá, ha az adatok megfelelnek a követelményeknek.
* Az adatok hozzáadása után a felhasználó egy üzenetet kap a sikeres regisztrációról.
* Ha az adatok hozzáadása nem sikerült, akkor a felhasználó egy hibaüzenetet kap.
* A visszatérési érték az üzenet szövege.

Funkciók:

* function test\_input($data): a bemeneti adatokat ellenőrzi és formázza, majd visszaadja a formázott adatot.
* $nev, $email, $jelszo, $varos, $irszam, $utca, $hazszam: a felhasználó által megadott adatokat tárolják.
* $hashelt\_jelszo: a felhasználó által megadott jelszó hashelt verziója, amely biztonságosabbá teszi a jelszótárolást.
* $sql: az SQL lekérdezés, amely hozzáadja az űrlapból kapott adatokat az adatbázishoz.
* mysqli\_query($conn, $sql): az adatok hozzáadása az adatbázishoz.
* mysqli\_error($conn): hibaüzenetet ad vissza, ha az adatok hozzáadása nem sikerült.
* header("Location: fooldal.php"): átirányítja a felhasználót egy másik oldalra.

A fenti kód tehát egy regisztrációs űrlap feldolgozását végzi el, és biztosítja, hogy az adatok megfelelően kerüljenek hozzáadásra az adatbázishoz. A visszajelzések segítségével a felhasználók tájékoztatást kapnak arról, hogy a regisztráció sikeres volt vagy sem.

// Űrlap adatok kezelése

if ($\_SERVER["REQUEST\_METHOD"] == "POST")

{ // Felhasználói adatok ellenőrzése

$nev = test\_input($\_POST["nev"]);

$email = test\_input($\_POST["email"]);

$jelszo = test\_input($\_POST["jelszo"]);

$varos = test\_input($\_POST["varos"]);

$irszam= test\_input($\_POST["irszam"]); $utca = test\_input($\_POST["utca"]);

$hazszam= test\_input($\_POST["hazszam"]);

// Jelszó hashelése

$hashelt\_jelszo = password\_hash($jelszo, PASSWORD\_DEFAULT); /

/ SQL lekérdezés futtatása $sql = "INSERT INTO regisztracio (felnev, email, jelszo, varos, irszam, utca, hazszam) VALUES ('$nev', '$email', '$hashelt\_jelszo', '$varos', '$irszam', '$utca', '$hazszam')";

if (mysqli\_query($conn, $sql)) {

echo "Sikeres regisztráció!";

header("Location: index7.php"); }

else { echo "Hiba a regisztráció során: " . mysqli\_error($conn); } }

// Adatbázis kapcsolat lezárása

mysqli\_close($conn);

// Segédfüggvény a felhasználói adatok ellenőrzéséhez

function test\_input($data) { $data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data); return $data; }

A következő kód a felhasználói bejelentkezési form kezelését valósítja meg. Az if(isset($\_POST['login'])) feltétel azt ellenőrzi, hogy az űrlap elküldése (a "login" gombra kattintás) megtörtént-e. Ha igen, akkor az űrlap által megadott felhasználónév és jelszó adatokat ellenőrzi.

Először az mysqli\_real\_escape\_string() függvénnyel "eltávolítja" a speciális karaktereket a felhasználónév és jelszó adatokból, hogy elkerülje a SQL injection támadást. Ezután az SQL lekérdezés az adatbázisból kiválasztja az összes rekordot, ahol a felhasználónév vagy az email cím egyezik a megadott felhasználónévvel.

Ha a lekérdezés eredménye legalább 1 sor (rekord), akkor a mysqli\_fetch\_assoc() függvénnyel az első rekordot asszociatív tömbként visszaadja. Ezután a password\_verify() függvény segítségével ellenőrzi a megadott jelszót a tárolt hash jelszóval. Ha a jelszó egyezik, akkor a felhasználói munkamenetet (session) elindítja és átirányítja a felhasználót a rendelés oldalra. Ha a jelszó nem egyezik, akkor a "Hibás felhasználónév vagy jelszó!" üzenetet jeleníti meg.

Ha a lekérdezés eredménye 0 sor (rekord), akkor ugyanazt az üzenetet jeleníti meg, hogy "Hibás felhasználónév vagy jelszó!".

A visszatérési értékek nem explicit módon vannak definiálva, hanem a felhasználót átirányítja a rendelés oldalra vagy kiírja a hibaüzenetet.

if(isset($\_POST['login'])) { $username = mysqli\_real\_escape\_string($conn, $\_POST['username']);

$password = mysqli\_real\_escape\_string($conn, $\_POST['password']);

$sql = "SELECT \* FROM regisztracio WHERE (felnev='$username' OR email='$username')"; $result = mysqli\_query($conn, $sql);

if(mysqli\_num\_rows($result) >= 1) { $row = mysqli\_fetch\_assoc($result);

if(password\_verify($password, $row['jelszo'])) { $\_SESSION['username'] = $row['felnev']; header("location: rendeles.php");

} else { echo "Hibás felhasználónév vagy jelszó!"; } }

else { echo "Hibás felhasználónév vagy jelszó!"; } }

A következő kód egy webalkalmazásban kezeli egy kosár tartalmának lekérdezését, majd annak árait összeadja és kiírja a képernyőre. Ha a kosár üres, akkor egy üzenetet jelenít meg erről.

Ezen kívül, ha a felhasználó elküldi a rendelést a "submit" gomb megnyomásával, akkor a kód rögzíti a rendelést az adatbázisban és kiírja egy üzenetben, hogy a rendelés sikeresen rögzítve lett.

A kód visszatérési értékei csak a kiírt üzenetek, vagy hibaüzenetek lehetnek, de az árak és a kosár tartalmának összegzése csak egy megjelenített értéket ad vissza a felhasználónak. A rendelés rögzítésekor semmilyen értéket nem tér vissza a kód, csak kiír egy üzenetet a sikeres rögzítésről.

if ($result->num\_rows > 0) { $row = $result->fetch\_assoc();

// Az árakat összeadjuk

$osszeg = 0; $osszeg += $row['hamburger'] \* 1000;

$osszeg += $row['pizza'] \* 1200;

$osszeg += $row['szendvics'] \* 1200;

$osszeg += $row['deszert'] \* 1200;

$osszeg += $row['salata'] \* 1200; $osszeg += $row['ital'] \* 1200;

// Kiírjuk az összeget

echo "A kosár összesen: " . $osszeg . " Ft"; }

else { echo "A kosár üres"; }

if(isset($\_POST['submit'])) { // rendelés rögzítése az adatbázisban // ... // üzenet kiírása a felhasználónak echo "Rendelését rögzítettük!"; }

A kód egy PHP fájl, amely az adatbázisba történő rendelések rögzítéséért felelős. Az első sorban hivatkozás található egy másik PHP fájlra, amelyben az adatbázis kapcsolat és a szükséges függvények definiálva vannak.

A második sorban egy feltétel vizsgálja, hogy a "submit" gombra kattintottak-e a felhasználók, ha igen, akkor az űrlapon megadott adatokat (étel id, hamburger, pizza stb.) elmenti az adatbázisba.

A harmadik sorban a mysqli\_real\_escape\_string() függvény használata miatt biztosítjuk, hogy az adatok biztonságosak legyenek a MySQL adatbázisban, és ne legyenek kihasználhatók adathalászati ​​támadásokra.

A negyedik és ötödik sorban létrehoz egy SQL lekérdezést az adatok rögzítésére az adatbázisban, majd az adatokat a lekérdezés segítségével hozzáadja a rendelés táblához.

A hatodik sorban egy feltétel vizsgálja, hogy az adatok sikeresen hozzáadódtak-e az adatbázishoz, ha igen, akkor a rendelés rögzítése sikeres volt, ha nem, akkor a hibaüzenetet jeleníti meg a felhasználónak.

include 'rendeles1.php';

if(isset($\_POST['submit'])) { $etelid = mysqli\_real\_escape\_string($conn, $\_POST['etelid']); $hamburger = mysqli\_real\_escape\_string($conn, $\_POST['hamburger']);

$pizza = mysqli\_real\_escape\_string($conn, $\_POST['pizza']);

$szendvics = mysqli\_real\_escape\_string($conn, $\_POST['szendvics']);

$deszert = mysqli\_real\_escape\_string($conn, $\_POST['deszert']);

$salata = mysqli\_real\_escape\_string($conn, $\_POST['salata']);

$ital = mysqli\_real\_escape\_string($conn, $\_POST['ital']);

$sql = "INSERT INTO rendeles (etelid, hamburger, pizza, szendvics, deszert, salata, ital) VALUES ('$etelid', '$hamburger', '$pizza', '$szendvics', '$deszert', '$salata', '$ital')"; if(mysqli\_query($conn, $sql)) { echo "Sikeres rendelés!"; } else { echo "Hiba: " . mysqli\_error($conn); } }

Az első blokk egy űrlapot dolgoz fel. Ha az űrlap elküldésekor a 'submit' gombra kattintunk, akkor az űrlapban megadott adatokat egy adatbázisba menti el. A második blokkban az adatok lekérdezése történik egy adatbázisból, majd egy táblázatban megjelenítjük az ételek adatait. Az utolsó blokkban pedig egy űrlapot jelenítünk meg, ahol a felhasználó megadhatja, hogy milyen ételeket szeretne rendelni.

A $conn változó egy adatbáziskapcsolatot jelöl, amit az index5.php fájlban lehet definiálni. Az SQL lekérdezésekhez az adatbáziskapcsolatot használjuk.

Az első blokkban lévő SQL INSERT INTO utasítás az adatbázisba helyezi el az űrlapban megadott adatokat. Az SQL utasításban az értékek előtt elhelyezett mysqli\_real\_escape\_string függvény a bemeneti adatokat "tisztítja", hogy megakadályozza az SQL injection támadásokat.

A második blokkban a lekérdezett adatokat egy táblázatban jelenítjük meg. A táblázat az etelek adatbázistábla elemeit tartalmazza, és a while ciklussal végigmegyünk az összes táblázatsorán.

Az utolsó blokkban a felhasználó megadhatja az ételek darabszámát, majd a "Rendelés" gombra kattintva az űrlap elküldi a rendelést az adatbázisba. A min='0' attribútumokkal biztosítjuk, hogy a felhasználók csak pozitív számokat adhassanak meg.

if(isset($\_POST['submit'])) { $hamburger = $\_POST['hamburger']; $pizza = $\_POST['pizza']; $szendvics = $\_POST['szendvics']; $deszert = $\_POST['deszert']; $salata = $\_POST['salata']; $ital = $\_POST['ital'];

$sql = "INSERT INTO kosar (hamburger, pizza, szendvics, deszert, salata, ital) VALUES ('$hamburger', '$pizza', '$szendvics', '$deszert', '$salata', '$ital')"; if ($conn->query($sql) === TRUE) { header('Location: kosar.php'); exit(); }

else { echo "Hiba a rendelés feldolgozásakor: " . $conn->error; } } if (isset($\_POST['submit'])) {

// rendelés feldolgozása } // adatok lekérdezése

$sql = "SELECT \* FROM etelek"; $result = $conn->query($sql);

// táblázat kiírása

echo "<form method='post'>"; echo "<table class='table'>"; echo "<tr><th>hamburger</th><th>pizza</th><th>szendvics</th><th>deszert</th><th>saláta</th><th>italok</th></tr>"; if ($result->num\_rows > 0) { while($row = $result->fetch\_assoc()) { echo "<tr><td>" . $row["hamburger"]. "</td><td>" . $row["pizza"]. "</td><td>" . $row["szendvics"]. "</td><td>" . $row["deszert"]. "</td><td>" . $row["saláta"]. "</td><td>" . $row["italok"]. "</td></tr>"; } } else { echo "Nincs találat"; } echo "</table>"; echo "<br>"; echo "<h2>Rendelés</h2>"; echo "<input type='number' name='hamburger' placeholder='Hamburger db' min='0'>"; echo "<input type='number' name='pizza' placeholder='Pizza db' min='0'>"; echo "<input type='number' name='szendvics' placeholder='Szendvics db' min='0'>"; echo "<input type='number' name='deszert' placeholder='Desszert db' min='0'>"; echo "<input type='number' name='salata' placeholder='Saláta db' min='0'>"; echo "<input type='number' name='ital' placeholder='Ital db' min='0'>"; echo "<input type='submit' name='submit' value='Rendelés'>"; echo "</form>"; // adatbázis kapcsolat bezárása $conn->close();

# Tesztelés

## 1. Regisztrációs űrlap

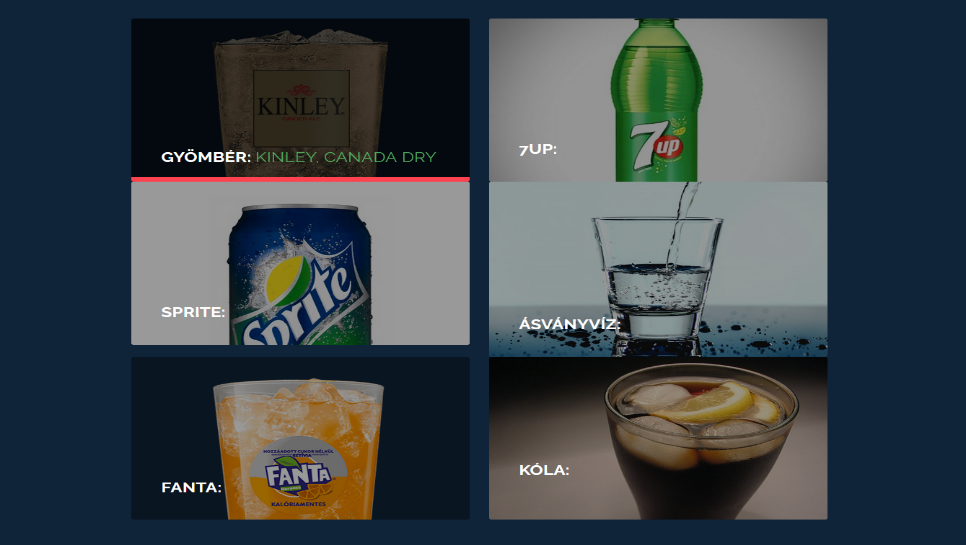
1. Az elején mi azt teszteltük, hogy a regisztrációs oldal rendesen megjelenik e a szöveg dobozok nem túl hosszúak e.
2. Második lépésként mi teszteltük, hogy a megadott adatokat a kód elmenti e az adatbázisba, ez több próbálozás után működött.

## 2. Rendelési űrlap

1. Elsőnek megnyitottuk a weboldalt, hogy a táblázat rendesen van e megjelenítve.
2. Utána ellenőriztük, hogy a megjelenő űrlap tartalmazza-e a hamburger, pizza, szendvics, desszert, saláta és ital mezőket.
3. Majd megnéztük, hogy rendelni kívánt gomb működik e.
4. Végül megnéztük, hogy a rendelés sikeresen felvételre került-e az adatbázisban.

# 3. Étlap kiíratása

1. Itt az első lépésként ellenőriztük, hogy az étlap megfelelően jelenik-e meg a felhasználó számára a kártyák rendesen néznek e ki.
2. Ellenőriztük, hogy a képek rendesen jellenek e meg nem csúsznak e el vagy nem pixelesek a képek.
3. Majd meg néztük, hogy a kártyák megfordulnak e, ahogy annak működnie kell.
4. Ezután meg néztük, hogy a leírások jól jellenek e meg, nincs e hiba a dizájnban.



15. ábra: étlap

# 4. Adatbázis kapcsolat lezárása

1. Itt megnéztük, hogy az adatbázis kapcsolat sikeresen lezáródik-e.
2. Megnéztük, hogy az adatbázisból minden adathoz hozzá fér e az oldal. Ezt lekérdezésekkel oldottuk meg.
3. Majd ismét meg néztük, hogy a kapcsolat sikeresen lezáródik-e.

# 4. SQL query

1. Ellenőriztük kell, hogy a SQL query sikeresen lefut-e és az adatok megfelelően kerülnek-e beillesztésre az adatbázisba.

# Továbbfejlesztési lehetőségek

Az oldal további funkciói az alábbiak szerint bővíthetők:

1. Felhasználói fiókok: A felhasználói fiókok lehetővé teszik a felhasználók számára, hogy saját profilokat hozzanak létre, személyre szabják a beállításaikat, kövessék más felhasználókat és így tovább. Ez a funkció lehetővé teszi az oldal személyre szabottabbá tételét és a felhasználói élmény fokozását.
2. Keresési funkció: Az oldal keresési funkcióval bővíthető, amely lehetővé teszi a felhasználók számára, hogy gyorsan megtalálják azokat az elemeket, amelyeket keresnek. Ez különösen hasznos lehet, ha az oldal nagy adatbázissal rendelkezik, vagy ha sok különböző típusú adatot kell kezelni.
3. Hozzászólás funkció : Az oldal hozzászólás funkcióval bővíthető, amely lehetővé teszi a felhasználók számára, hogy elmondják a véleményüket az adott ételről vagy magáról az étteremről az oldalon belül. Ez növeli az interakciót az oldalon, és javítja a felhasználói élményt.
4. Fórum funkció: Az oldal fórum funkcióval bővíthető, amely lehetővé teszi a felhasználók számára, hogy témákhoz kapcsolódó vitákat folytassanak, kérdéseket tegyenek fel, és megoszthassák az ismereteiket. Ez javítja az oldal közösségi aspektusát, és segíti a felhasználókat abban, hogy segítséget kérjenek és adhassanak.
5. Hirdetési funkció: Az oldal hirdetési funkcióval bővíthető, amely lehetővé teszi a felhasználók számára, hogy hirdetéseket adjanak fel az oldalon. Ez javítja az oldal bevételi forrásait, és lehetővé teszi az oldal tulajdonosainak, hogy újabb szolgáltatásokat nyújtsanak a felhasználóknak.
6. Mobilalkalmazás: Az oldal mobilalkalmazással bővíthető, amely lehetővé teszi a felhasználók számára, hogy ne csak böngészőből, hanem egy alkalmazásból is tudjanak rendelni. Ez gyorsabb lehetőséget biztosít a felhasználók számára. illetve több vásárlóhoz tud eljutni az adott weboldal.
7. Szállítási lehetőségek hozzáadása: Ha az oldalt ételrendelési szolgáltatásnak szánjuk, akkor érdemes lehet olyan funkciókat hozzáadni, mint például a szállítási lehetőségek, a szállítási díjak, valamint a szállítási idők beállítása.
8. Értékelési rendszer: Az oldalra fel lehetne venni egy értékelési rendszert, amely lehetővé tenné a felhasználók számára, hogy értékeljék az éttermeket, az ételeket, valamint a szolgáltatást. Ezáltal az oldal megbízhatósága növekedne, és a felhasználók könnyebben dönthetnének, hogy hol szeretnének ételt rendelni.
9. Kuponok és promóciók: Az oldalra lehetne felvenni kuponokat és promóciókat, amelyeket a felhasználók felhasználhatnak az ételrendelés során. Ezáltal növelhető a felhasználók aktivitása az oldalon, és ösztönözhetők az ismételt rendelések.
10. Rendelési előzmények: Az oldalra fel lehetne venni egy rendelési előzmények funkciót, amely lehetővé teszi a felhasználók számára, hogy megtekinthessék korábbi rendeléseiket. Ezáltal a felhasználók könnyebben újrarendelhetik kedvenc ételeiket.
11. Integráció a közösségi médiával: Az oldalt lehetne integrálni a közösségi médiával, amely lehetővé teszi a felhasználók számára, hogy megoszthassák rendeléseiket és értékeléseiket barátaikkal és családtagjaikkal. Ezáltal növelhető az oldal láthatósága és népszerűsége.
12. Kedvenc éttermek: Az oldalon lehetőség lenne felvenni egy kedvenc éttermek funkciót, amely lehetővé teszi a felhasználók számára, hogy könnyen megtalálják és újra rendelhessék kedvenc éttermeiket.

# Összegzés

Az általunk tervezett weboldal egy étterem rendelési platformot foglal magában, amely lehetővé teszi a felhasználók számára, hogy az oldalon keresztül ételt rendeljenek és fizessenek online.

A weboldal tartalmazni fog egy adatbázist, amelyben tárolni lehet az ételeket, azok árát és a rendelések adatait.

A felhasználók által leadott rendelések adatait egy űrlap segítségével rögzítjük, és az adatbázisba mentjük.

Az oldalon található étlapot dinamikusan generáljuk az adatbázisban tárolt adatok alapján.

A weboldal jelenlegi verziójában a felhasználóknak nincs lehetősége fiókot létrehozni, de lehetőség van a jövőbeni fejlesztésekre, például a felhasználói fiók létrehozására.

További funkciók közé tartozik például a mobilalkalmazás, a promóciós kódok, az értékelések és a vélemények, a szállítási idő becslése és az ételekhez kapcsolódó diétás információk megjelenítése.

A weboldal fejlesztése során fontos szerepet kell kapnia a tesztelésnek is, amelynek célja az oldal megbízhatóságának és funkcionalitásának ellenőrzése.

Végül, a weboldal egy biztonságos és kényelmes módja lehet az ételrendelésnek az ügyfelek számára, amely az étteremnek is előnyös, mivel az online rendelések hatékonyabbá és átláthatóbbá tehetik az üzletmenetet.

Főbb nehézségeink, igazából a backend résznél voltak, az elején, amit ki választottunk backend nyelvet nem igazán tudtunk benne haladni, itt főképp olyan problémáink voltak, hogy a verziók eltértek egymástól, vagy nem támogatták már az kódrészt.

Ezen kívül a githubbal volt még kisebb fennakadásunk mivel nem nagyon használtunk még githubot, és nem tudtuk, hogy hogy lehet feltölteni a projekt mappát.

# Forrásmegjelölés

<https://github.com/Zsolt0508/projekt2.git>

# Köszönetnyilvánítás

Szeretnénk köszönetet mondani Illésné Kincsei Valériánák, aki mindig tudott nekünk bármilyen problémára megoldást találni, alternatívákat javasolni, megkérdezni egy olyan tanárt, aki esetleg tud nekünk segítséget nyújtani, vagy csak éppen érdeklődött, hogy pillanatnyilag, hogy áll a projektünk.

Szeretnénk még köszönetet mondani Illés Attila tanárnak, aki sok remek ötlettel látott el minket a projekt haladása alatt.

[1. ábra:Munkafolyamat 3](#_Toc132283378)

[2. ábra: Étlap 4](#_Toc132283379)

[3. ábra: Rendszerkövetelmény 5](#_Toc132283380)

[4. ábra: Xampp 6](#_Toc132283381)

[5. ábra: MYSQL 7](#_Toc132283382)

[6. ábra: Regisztráció 9](file:///C:\Users\zsolt\Downloads\FastFoodKing_1.docx#_Toc132283383)

[7. ábra: Főoldal 10](file:///C:\Users\zsolt\Downloads\FastFoodKing_1.docx#_Toc132283384)

[8. ábra: Rendelés 11](file:///C:\Users\zsolt\Downloads\FastFoodKing_1.docx#_Toc132283385)

[9. ábra: Kosár 11](file:///C:\Users\zsolt\Downloads\FastFoodKing_1.docx#_Toc132283386)

[10. ábra: Xampp2 13](file:///C:\Users\zsolt\Downloads\FastFoodKing_1.docx#_Toc132283387)

[11. ábra: Kezdetleges adatbázis 14](file:///C:\Users\zsolt\Downloads\FastFoodKing_1.docx#_Toc132283388)

[12. ábra: Fejlesztési folyamat 15](#_Toc132283389)

[13. ábra: Fejlesztési folyamat2 16](file:///C:\Users\zsolt\Downloads\FastFoodKing_1.docx#_Toc132283390)

[14. ábra: Adatbázis 19](file:///C:\Users\zsolt\Downloads\FastFoodKing_1.docx#_Toc132283391)

[15. ábra: étlap 27](file:///C:\Users\zsolt\Downloads\FastFoodKing_1.docx#_Toc132283392)

**Baranya Megyei Szakképzési Centrum**

**Simonyi Károly Technikum és Szakképző Iskola**

Szakma megnevezése: Szoftverfejlesztő és –tesztelő

A szakma azonosító száma: 5 0613 12 03

**Vizsgaremek**

FastFoodKing

Készítették: Hegedűs Zsolt 2/14-4

Kisfali Krisztián 2/14-4

Raffai Ferenc 2/14-4

Konzulens: Dávid Balázs

Pécs

2023

**Baranya Megyei Szakképzési Centrum**

**Simonyi Károly Technikum és Szakképző Iskola**

Szakma megnevezése: Szoftverfejlesztő és –tesztelő

A szakma azonosító száma: 5 0613 12 03

**Vizsgaremek**

Készítették: Hegedűs Zsolt

Kisfali Krisztián

Raffai Ferenc

Konzulens: Dávid Balázs

Pécs

2023