

Baranya Vármegyei SzC Radnóti Miklós Közgazdasági Technikum

Szakma megnevezése: Szoftverfejlesztő és tesztelő

A szakma azonosító száma: 506131203

Vizsgaremek

Készítette:

Kátay Balázs

Kovács Máté

**Pécs
2025**

Tartalomjegyzék

Milyen Problémára Kínál Megoldást?.....	4
A projekt megvalósításához.....	4
Funkciók:	5
1. Felhasználói fiókok és regisztráció:	5
2. Menü és választék:	5
3. Rendelési folyamat:	5
4. Akciók és kuponok:.....	5
5. Admin funkciók:.....	5
6. Futár funkciók:.....	5
Eszközök, amiket a projekthez használtunk:	6
Frontend:	6
Backkend:	6
Manuális teszt:	6
Tervezés:.....	6
Automatizált teszt:.....	6
Adatbázis:	6
Telepítési dokumentáció:	7
Visual Studio Code (VS Code)	7
Letöltés és telepítés:	7
Ellenőrzés:.....	7
Szükséges bővítmények (extensions) letöltése:	7
XAMPP.....	8
Letöltés és telepítés:	8
Ellenőrzés:.....	8
Composer	10
Letöltés és telepítés	10

Ellenőrzés:.....	10
Figma:.....	11
Figma használata böngészőben (telepítés nélkül):	11
Figma letöltése Windows-ra:	12
A gombokkal való interakcióhoz szükséges információk.	12
Vizsgaprojekt futtatása:.....	12
Project felépítése	13
Az assets	13
A css.....	13
Js.....	13
Layout.....	13
Backend:	14
Controller	14
Model	14
Public.....	14
Test	14
Vendor	14
Adatbázis:.....	15
Tesztelések:	16
Manuális tesztelés:	16
Automatizált tesztelés:	16
PHPUnit:	16
Jövőbeli terveink:	17

PizzaBázis

A projekt célja egy online pizza rendelős oldal fejlesztése, amely gyors, egyszerű és kényelmes módot kínál a felhasználók számára, hogy különféle pizzákat rendeljenek otthonukba vagy munkahelyükre.

Milyen Problémára Kínál Megoldást?

Időhiány: Az embereknek gyakran nincs idejük vagy kedvük éttermekbe menni vagy főzni, és ez a projekt lehetővé teszi, hogy néhány kattintással megrendeljék kedvenc pizzájukat.

Átláthatóság és választék: A felhasználók könnyen áttekinthetik a kínálatot és annak árait, és ezek alapján tudnak rendelést leadni.

Kényelem és elérhetőség: Az oldal 24/7 elérhető, lehetővé téve a rendelést bármikor, még akkor is, amikor az étterem nem venné fel a telefont.

A projekt megvalósításához Frontend, Backend, Adatbázis, Tesztelés, Dokumentáció és Tervezés feladatokat kell teljesíteni, amit e-képpen teljesítünk.

Kátay Balázs: Frontend, Backend, Adatbázis, Automata tesztelés, Dokumentáció, Tervezés

Kovács Máté: Frontend, Manuális tesztelés, Dokumentáció, Tervezés

A project rész feladatait sprint-ekben vezettük, ahol taskokat készítettünk a **Jira**-ban. Ezeket a sprinteket havonta indítottuk, és az előző havit lezártuk.

A feladatok elvégzése után a **Git**-re töltöttük fel, ahol a csapat társunk letudta tölteni a fájlokat, és tudott velük dolgozni.

Valamint **Discord**-ot használtunk a kommunikációra, és egyéb kisebb fájlok átadására.

Funkciók:

1. Felhasználói fiókok és regisztráció:
 - Regisztráció és bejelentkezési lehetőség.
 - Profilkezelés/Profil adatok módosítása
 - Elfelejtett jelszó funkció.
2. Menü és választék:
 - Részletes menü képekkel, leírásokkal és árakkal.
3. Rendelési folyamat:
 - Kosárba helyezés.
 - Szállítási adatok megadása
4. Akciók és kuponok:
 - 24 óránként 3 különböző pizza kap 5%, 10%, 15% leárazást.
 - Kupon kódok használatával olcsóbban lehet rendelni.
5. Admin funkciók:
 - Új pizzák hozzáadása és módosítása
 - Fiókok ideiglenes vagy végleges tiltása
 - Kuponok hozzáadása és módosítása
6. Futár funkciók:
 - Bejövő rendelések elfogadása, és a rendelés leadása.

Eszközök, amiket a projekthez használtunk:

Frontend:

- HTML5.2 verzió
- CSS3 verzió
- ECMAScript 2020
- JPG fájlkiterjesztést a képekhez
- Bootstrap 5.3.0 verzió

Backend:

- PHP 8.4
- composer
- vendor
- Firebase JWT
- phpmailer

Manuális teszt:

- Excel 2016 (16.0.6366.2036)

Tervezés:

- Figma
- dbdiagram

Automatizált teszt:

- Composer
- Vendor
- PHPUnit 12

Adatbázis:

- XAMPP phpMyAdmin (mysql)

Telepítési dokumentáció:

Visual Studio Code (VS Code)

Letöltés és telepítés:

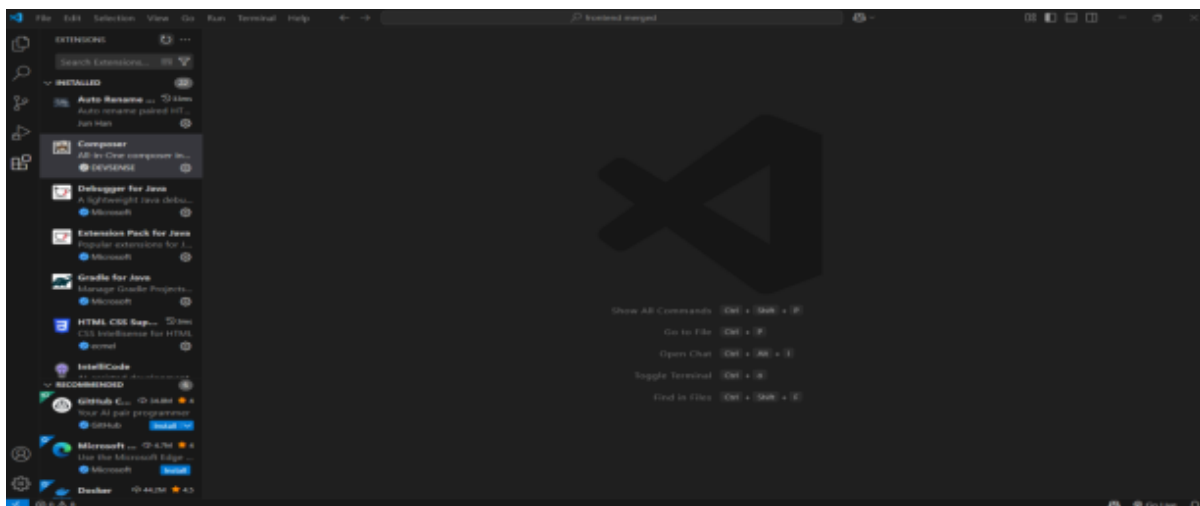
1. Nyisd meg: <https://code.visualstudio.com/>
2. Kattints a "Download for Windows" gombra.
3. Futtasd a letöltött .exe fájlt.
4. Kövesd a telepítő utasításait, pipáld be: "Add to PATH", "Open with Code".
5. Kattints "Install", majd indítsd el az alkalmazást.

Ellenőrzés:

- Ha elindul a VS Code, sikeres a telepítés.
- CMD-ben: ``code .``

Szükséges bővítmények (extensions) letöltése:

- Composer
- HTML, CSS Support
- PHP Extension Pack
- Live Server A bővítményeket itt lehet telepíteni:



A kereső mezőbe beírjuk a keresett bővítményt(extensiont).
Kiválasztjuk és az install gombra kattintva telepítjük.

XAMPP

XAMPP tartalmazza a PHP-t: A XAMPP telepítésével automatikusan telepítve lesz a PHP is, így nem kell külön PHP-t letölteni és beállítani.

Kényelmes konfigurálás: A XAMPP egyetlen telepítőcsomagban tartalmazza az Apache webszervert, MySQL adatbázist és PHP-t. Ez egy egyszerű, helyi fejlesztői környezetet biztosít, amit könnyen elindíthatsz a XAMPP vezérlőpultjából.

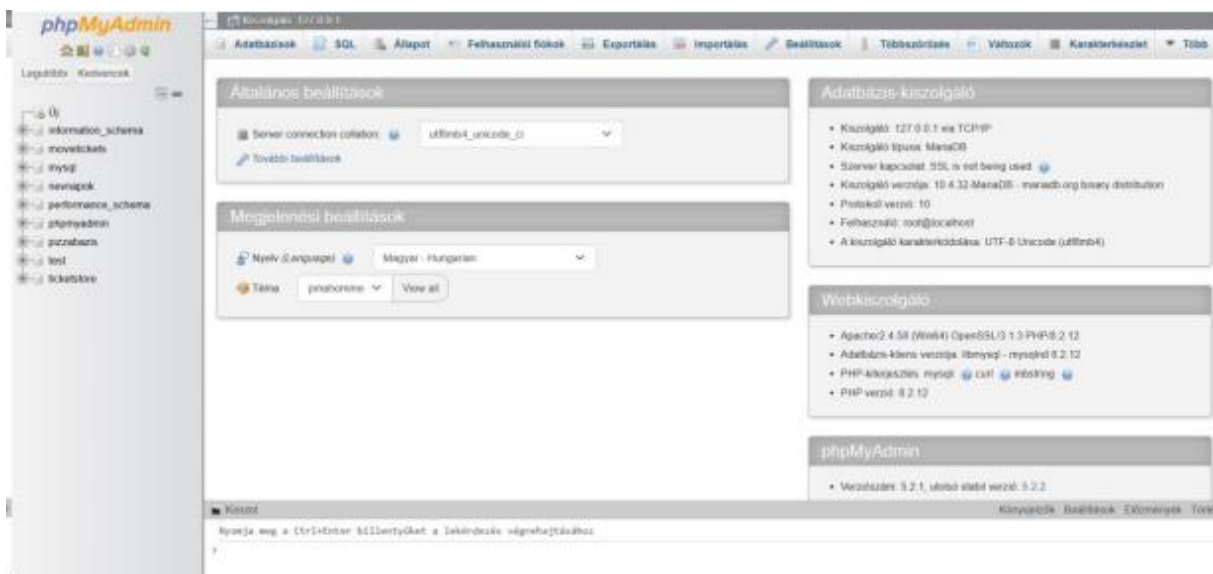
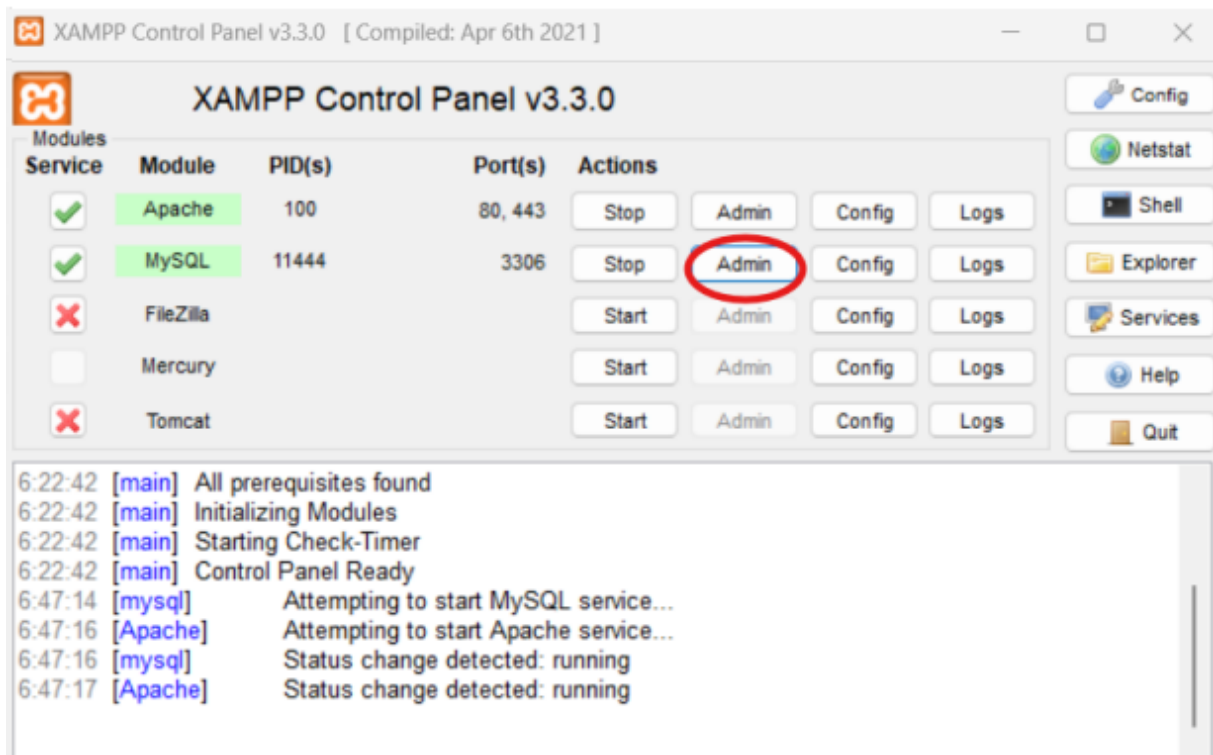
Könnyű használat: A XAMPP telepítése után nem kell manuálisan konfigurálnod a PHP-t vagy Apache-t, mindent egy helyről kezelhetsz.

Letöltés és telepítés:

1. Nyisd meg: <https://www.apachefriends.org/hu/download.html>
2. Töltsd le a xampp-windows-x64-8.2.12-0-VS16-installer verziót.
3. Futtasd a .exe telepítőt.
4. Hagyj mindent alapértelmezetten.
5. Telepítés után indítsd el a XAMPP Control Panelt.

Ellenőrzés:

- Indítsd el Apache és MySQL-t és a MySQL résznél az "Admin" gombra kattintva megnyílik a phpMyAdmin

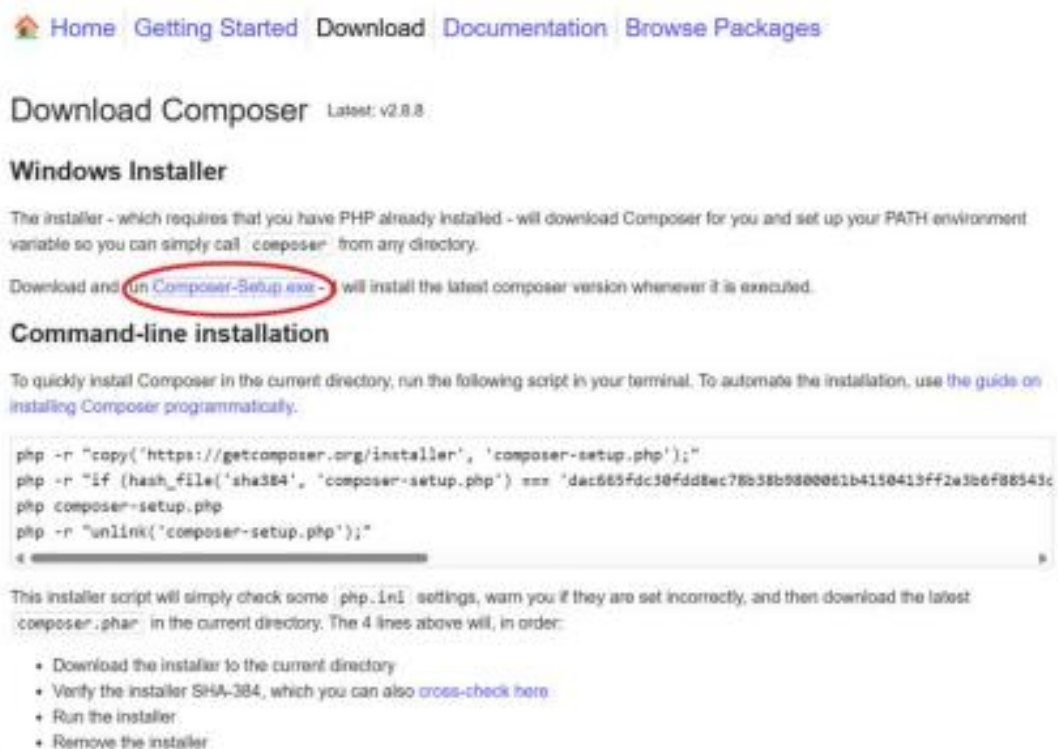


Ha egy hasonló felületet látsz mint ez akkor megfelelően működik a xampp.

Composer

Letöltés és telepítés:

1. Nyisd meg: <https://getcomposer.org/>
2. Töltsd le: <https://getcomposer.org/download/>



3. Futtasd a telepítőt.
4. Add meg a PHP elérési útvonalát, pl: C:\\xampp\\php\\php.exe
5. Telepítés után CMD-ben futtasd:

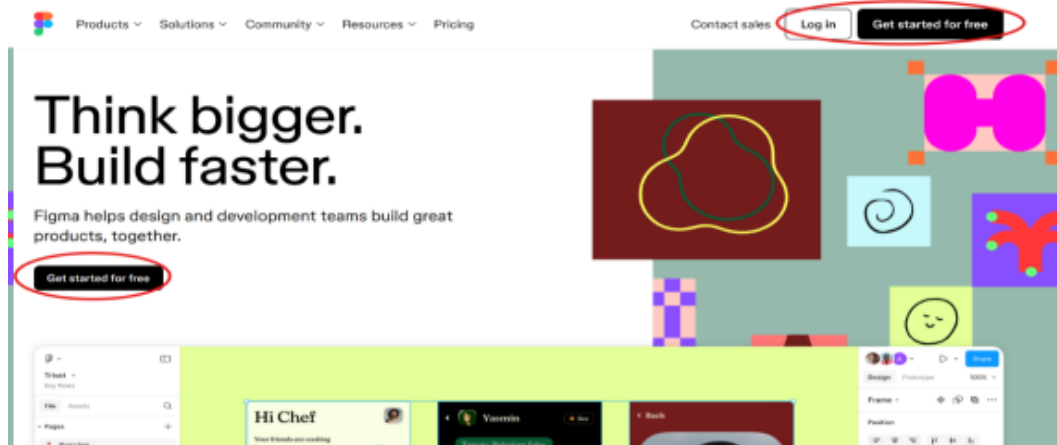
Ellenőrzés:

- `composer -V` (verzió megjelenik)

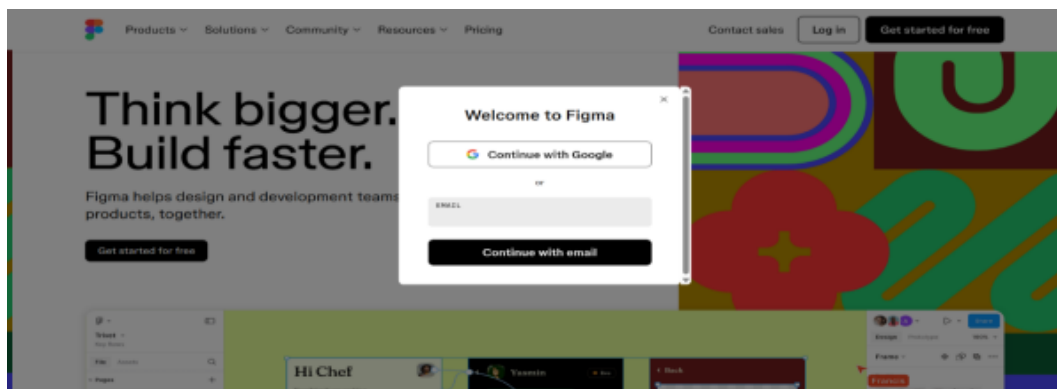
Figma:

Figma használata böngészőben (telepítés nélkül):

1. Nyisd meg a böngészőt (pl. Chrome, Firefox, Edge).
2. Látogass el a <https://figma.com> weboldalra.
3. Kattints a "Get started for free" vagy "Log in" (bejelentkezés) gombra



4. Hozd létre a Figma-fiókodat, vagy jelentkezz be Google-fiókkal.



5. Máris elkezdheted használni a Figmát a böngészőből – nincs szükség letöltésre!

Figma letöltése Windows-ra:

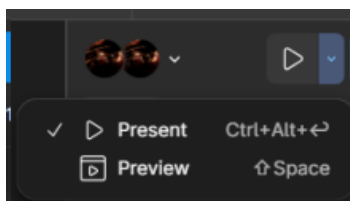
1. Nyisd meg a Figma hivatalos oldalát:
<https://figma.com/downloads>
2. Kattints a "Download for Windows" gombra.
3. Töltsd le a FigmaSetup.exe fájlt.
4. Futtasd a letöltött fájlt (dupla kattintás).
5. A Figma automatikusan települ – nem igényel bonyolult beállítást.
6. Jelentkezz be a Figma-fiókkal, és már használhatod is az asztali alkalmazást.

Bejelentkezést követően ezen a linken lehet megtekinteni a projektünk dizájn tervét:

<https://www.figma.com/design/mRnQQZhpPvH70zKPxYy1ua/PizzaB%C3%A1zis?node-id=0-1&p=f&t=DgEejwj2acnaOE3x-0>

A gombokkal való interakcióhoz szükséges információk.

Ahhoz, hogy megnyissuk a dizájn prototype előadás módját a gombok navigálásának bemutatásához jobb fent a “play gomb” lenyításakor kiválasztjuk a “present” lehetőséget.



Vizsgaprojekt futtatása:

A xampp megnyitását követően az Apach és MySQL-t elindítva a “**start**” gomb teszi lehetővé az adatbázis kapcsolatot.

A visual studio codeban a bal felső sarokban File>open folder>a projekt mappája>megnyitás

A jobb alsó sarokban a “Go live” gombra kattintva lehet a megnyitott/kiválasztott HTML fájlt futtatni.

Project felépítése (MVC model)

A Pizzabázis tartalmazza a frontend-et, backend-et, és adatbázist.

A Frontend:

Az **assets** tartalmazza két darab mappt, ami a **css** és a **js(javascript)** mappák.

A **css**-ben a projekt összes css fájlja található. Az admin mappában az admin felületek css-e található. A deliverPerson mappában a futároknak szánt felület css-e található.

Js mappában találhatóak a javascript fájlok, valamint az admin mappa, amiben meg az adminoknak szánt rész van. A js mappában még a deliveryPerson mappa is található, ahol a futárnak szánt oldal javascriptjei találhatóak.

Layout-ban található a layout html-je, css-e, javascript-je és a hozzá tartozó képek mappája design images néven. A design images mappában viszont nem csak a layout képei található, hanem a statikus, az oldalon nem változó képek is itt található.

A **view** tartalmazza a customer, az admin, és a deliveryPerson mappát. A customer mappában a vásárlók számára szánt html fájlokat lehet megtalálni, valamint az admin mappában az admin jogosultsággal rendelkező felhasználók részére szánt html fájlokat tartalmazzák. A deliveryPerson mappában a futároknak szánt oldal található, amit csak a deliveryPerson jogosultsággal rendelkező felhasználók érnek el. A futárok nem férnek hozzá a vásárlói nézetekhez.

Backend:

Controller mappa tartalmazza azokat a php fájlakat, amik az API-t szolgáltatják JSON formátumban. Ebben a mappában történik az üzleti logika.

Az **images** mappában a menu összes képe található. Ide a képek a weboldal adminja által kerülnek bele. A számozások a pizza id-ja alapján vannak megadva, és a számozott mappákban a pizza képek pizza_(id) névvel találhatóak meg.

Model mappa tartalma a Controller számára fontos adatok, amiket az adatbázisból kérdez le.

Public mappában lévő fájlok szolgálnak API-ként, amik beállítják a Cors szabályokat.

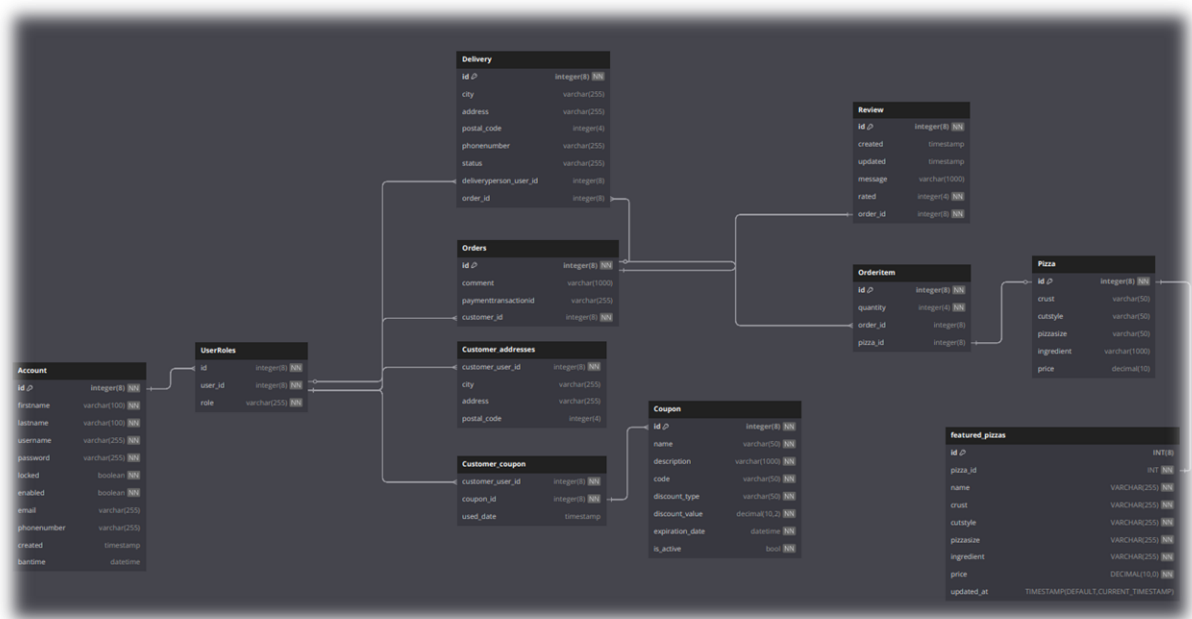
Test mappában található a phpunit által szolgáltatott unit tesztek, amik a controllert és a modelt tesztelik, ami 2 db külön mappában van elhelyezve. A model mappában található a model fájlokról szóló unit tesztek, és a controller mappában a controller fájlokról szóló unit tesztek.

Vendor mappában a phpunit működéséhez szükséges fájlok, az elfelejtett jelszóhoz szükséges phpmailer.

A **Composer** fájlok nincsenek mappában, hanem a Backend fő mappájában található

Adatbázis:

Az adatbázis egy weboldali program alapvető része, amely az adatok strukturált tárolását, rendszerezését és elérését biztosítja. Lehetővé teszi, hogy a weboldal dinamikusan működjön: például regisztráció során elmenti a felhasználók adatait, bejelentkezéskor ellenőrzi azokat, vagy egy webshopban a termékek listáját tárolja és kezeli. Az adatbázis segítségével a program gyorsan tud adatokat lekérni, módosítani vagy törölni, anélkül hogy minden adatot külön fájlokban kellene kezelnie. Emellett az adatbázis támogatja az adatok biztonságos tárolását, a jogosultságok kezelését, és a nagy mennyiségű információval való hatékony munkát is. Egy jól megtervezett adatbázis hozzájárul a weboldal stabil, gyors és megbízható működéséhez.



Tesztelések:

A különböző tesztelési fajták célja, hogy különböző szempontok szerint vizsgálják az alkalmazást, és biztosítsák annak minőségét. A manuális tesztelés, az automatizált tesztelés és a PHPUnit (unit tesztelés) mind eltérő megközelítéseket képviselnek, és mindegyiknek megvan a saját szerepe a fejlesztési folyamat során.

Manuális tesztelés:

A manuális tesztelés olyan eljárás, amikor a tesztelő saját kezűleg végzi el a tesztek, vagyis emberi erőforrás szükséges a program működésének ellenőrzéséhez. A manuális tesztelés rendkívül hasznos lehet, amikor új funkciókat vezetnek be, és fontos ellenőrizni, hogy azok a felhasználói igényeknek megfelelően működnek-e. Ilyenkor a tesztelő képes a program interaktív módon való használatával értékelni a felhasználói élményt, például a navigációt, a vizuális elemek minőségét, vagy a felhasználói interfész zökkenőmentességét.

Automatizált tesztelés:

Az automatizált tesztelés pedig egy olyan tesztelési módszer, amely különféle szoftvereszközökkel végzi el a tesztek, így csökkenti az emberi munkaerő szükségességét, és lehetővé teszi a tesztelési ciklusok gyorsabb végrehajtását. Az automatizált tesztelés különösen akkor jelentős, ha a projekt nagyméretű, és folyamatos integrációval működik. Az ilyen tesztelési módszer gyors és hatékony megoldást kínál, mivel az egyszer már megírt tesztek bármikor újra futtathatók anélkül, hogy újra kézzel kellene végrehajtani őket.

PHPUnit:

A PHPUnit tesztelés vagy unit tesztelés olyan eljárás, amely a kód egy-egy kis egységének, például egy függvénynek vagy osztálynak a helyes működését ellenőrzi. A PHPUnit egy keretrendszer, amely a PHP nyelvben elterjedt, és lehetővé teszi, hogy a fejlesztők automatikusan teszteljék a kódot az egyes funkciók szintjén.

Jövőbeli terveink:

A felhasználóknak készíteni egy felületet, ahol a korábbi rendeléseiket tudnák megtekinteni és ha szeretnék esetleg újra meg rendelni, akkor erre egy funkciót is létrehozni.

A fizetés után egy nyomon követési rész létrehozása.

A rendelés sikeres megérkezése után lehessen véleményt írni a pizzáról vagy esetleg a szállításról.

A Bank kártyás fizetés lehetőségének kialakítása.

Jelenleg csak a 375px széles telefonokig van kialakítva a reszponzivitása az oldalnak, és a kisebb képernyővel rendelkező telefonoknak is érdemes lenne ezt a lehetőséget megadni.

A futár fiók regisztrálásának külön kezelése.

Az admin felületnél az összes rendelés megtekintésére a lehetőség, és azon belül egy szűrés funkció, amivel egyszerűbben lehet benne keresni.

A kupon kódok mellet a fiókokhoz rendelt kuponok lehetőségének létrehozása.