

Vas Vármegyei Szakképzési Centrum  
 Horváth Boldizsár  
 Közgazdasági és Informatikai  
 Technikum

A 5 0613 12 03 számú Szoftverfejlesztő és –tesztelő vizsgaremek

**A szoftveralkalmazás dokumentációja**

Készítették:

**B**enjámin Soós

**M**áté Őri

**Zs**olt Zanati

**Szombathely**

**2025**

Tartalomjegyzék

[Bevezetés 3](#_Toc196689566)

[1. Felhasználó igényei 4](#_Toc196689567)

[2. Projekt komponenseinek technikai leírása 5](#_Toc196689568)

[2.1 Rendszerkövetelmények (minimum) 8](#_Toc196689569)

[3. Projekt használatának rövid bemutatása 9](#_Toc196689570)

[4. Szerepkörök, Feladatok felosztása 10](#_Toc196689571)

[5. Határidőnaplók 11](#_Toc196689572)

[5.1 adatbázis 11](#_Toc196689573)

[5.2 backend 11](#_Toc196689574)

[5.3 asztali alkalmazás 12](#_Toc196689575)

[5.4 frontend 13](#_Toc196689576)

[6. Adatbázis felépítése 14](#_Toc196689577)

[6.1 Relációs modell 14](#_Toc196689578)

[6.2 ER-modell 14](#_Toc196689579)

[7. Projekt használata 15](#_Toc196689580)

[7.1 Weboldal használata 15](#_Toc196689581)

[7.2 Asztali alkalmazás használata 20](#_Toc196689582)

[8. Tesztdokumentáció 25](#_Toc196689583)

[8.1 Backend tesztelés 25](#_Toc196689584)

[8.2 Frontend tesztelés 27](#_Toc196689585)

[9. Bejelentkezés 28](#_Toc196689586)

[10. Fejlesztést megkönnyítő kommunikációs eszközök 29](#_Toc196689587)

[10.1 Discord 29](#_Toc196689588)

[10.2 GitHub 29](#_Toc196689589)

[10.3 Google Drive 29](#_Toc196689590)

[11. Indítási útmutató 30](#_Toc196689591)

[12. Továbbfejlesztési lehetőségek 31](#_Toc196689592)

[14. Ábrajegyzék 32](#_Toc196689593)

# Bevezetés

A projektünk témája egy fodrászat-láncot összefogó felület. A hétköznapokban gyakran találkozunk azzal a problémával, hogy fodrászat keresés közben nem találunk egy megbízható, hatékony oldalt, ami összefoglalná a fodrászatokat egy weboldalba. Sok fodrászatnak vagy egyáltalán nincs, vagy nagyon régi, időt múlt, nem hatékony weboldallal rendelkezik. Ezért döntöttünk úgy, hogy egy olyan fodrászat franchise-t összefoglaló webes felületet építünk ki, aminek kezelése könnyű és kiépítése átlátható. Így a fiataltól az idősig bárki játszi egyszerűséggel megtudja tekinteni a fodrászatinkat, képeinket, és pár kattintással időpontot is tud foglalni.

Sok fodrászatnak a galériája nagyon kevés képet tartalmaz és legtöbbször rossz minőségben is jelennek meg, de van olyan is, hogy egyáltalán nincs galéria. Ezért az oldalunkon létrehoztunk egy galériát, ahol ki lehet szűrni a különböző fodrászatokban dolgozó fodrászok képeire, és lehetőség nyíliik egy szavazás opcióra 1 és 5 között, csillagok formájában. Ezzel az oldalunkra látogatók meg tudják tekinteni, hogy jelenleg milyen hajstílus a legfelkapottabb, és ki vágta.

A webes felület mellett készítettünk egy asztali alkalmazást is ami a fodrászokat segíti. Ki tudják listázni az adott fodrászatba szóló foglalásokat, így megkönnyítve egy munkanap lebonyolítását, illetve feltudják vinni, hogy melyik nap dolgoznak egy beosztás kezelő felületen.

Készítettünk egy konzolos „ApiStarter”-t ami elindítja az adott gépen az API-t ami kiszolgálja a webes, és asztali alkalmazást. Ez elengedhetetlen a program működéséhez.

Programjaink célja, hogy az adott fodrászat-lánc munkatársai egységesen és kooperatívan tudjanak működni, valamint, hogy a felhasználók könnyen, egy rendezett felületen tudjanak a nekik megfelelő fodrászatok, fodrászok és szolgáltatások közül válogatni.

# 1. Felhasználó igényei

A projekt tervezési szakaszában fontosnak tartottuk, hogy a felhasználó a lehető legnagyobb kényelembe tudja kezelni oldalunkat. Ezért odafigyeltünk, hogy aki nem túl jártas az internetes világban is könnyedén kiigazodjon a weboldalon. Legfőbb kiindulási pontunk a foglalás menetének a megkönnyítése volt. Sok helyen nem átlátható, nem egyértelmű, hogy mire és milyen adatokkal foglalt az oldalhoz. Mi egy saját magunk által programozott naptárral könnyítettük meg a menetet. Ezen kívül a fodrászatot, a fodrászt és a szolgáltatást a lehető legátláthatóbban oldottuk meg.

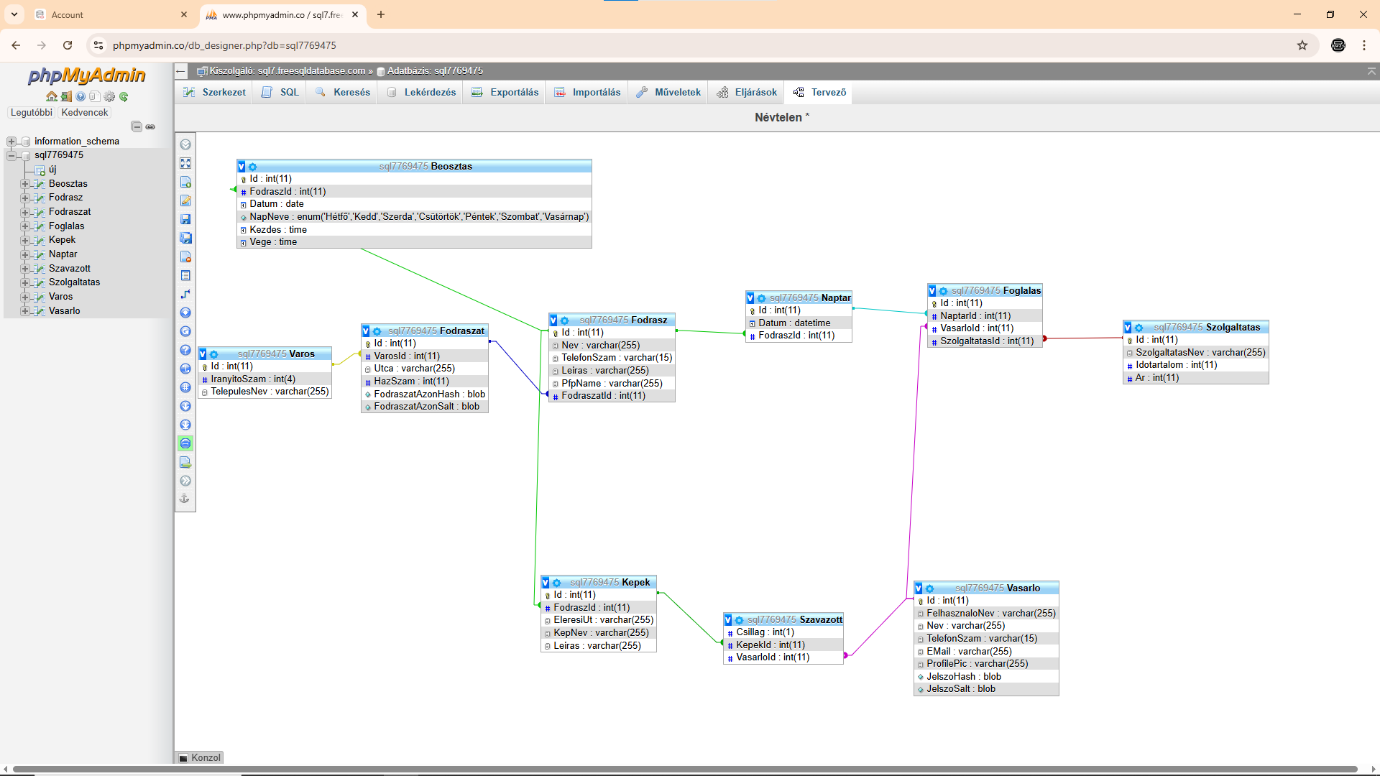
A profilján a felhasználónak megjelennek az adatai, amiket tud módosítani. Sok fodrászati oldalon ilyenekre nincs lehetőség. Hogyha ugyan azon a gépen és böngészőben lép be az oldalra, emlékezni fog rá, így nem kell minden belépésnél újra és újra bejelentkezni.

Külön odafigyeltünk a reszponzív kiépítésre, ami a különböző méretű készülékeknél is hasonlóan néz ki az oldal, mint ha teljes, 1920x1080-as méretben nyitná meg az oldalt.

A kezdőoldalon minden információt megtalál fodrászatunkról a felhasználó. Nyitvatartást, szolgáltatásokat és árait, rövid leírást, kapcsolatokat.

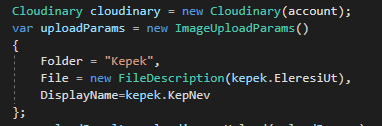
# 2. Projekt komponenseinek technikai leírása

Adatbázisunk egy MySql adatbázis, amit a könnyebb csoportmunka érdekében felraktuk egy phpMyadmin szerverre így minden csoporttag könnyedén eléri, és mindenki azonos adatokkal dolgozik. Ehhez a **freesqldatabase.com** oldalt használtuk, ami ingyen 5mb-ig enged adatokat tárolni, és egy phpMyAdmin felülettel könnyíti meg az adatbázis kezelésében.

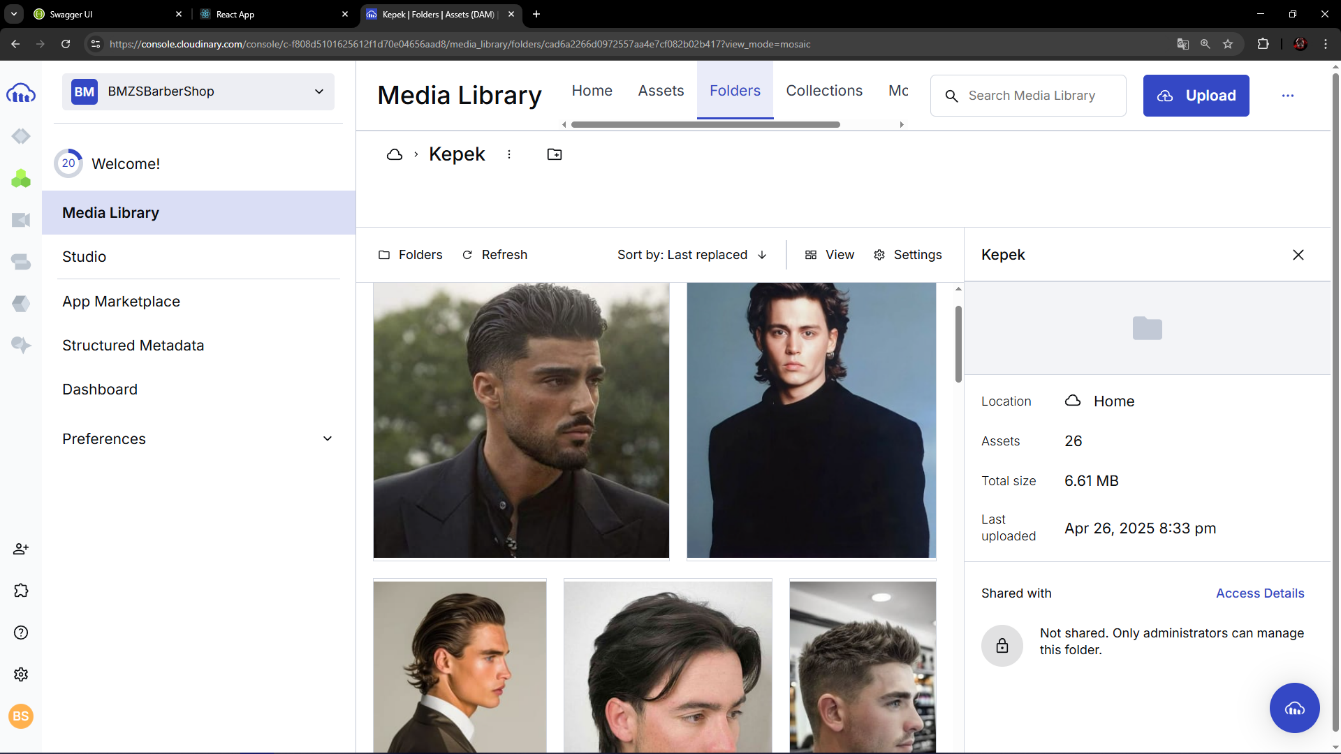


1. kép freesqldatabase.com oldalon található adatbázisunk

A weboldalunkon megjelenő képeket **Cloudinary**-ban tároljuk, ami felhőalapú médiakezelési szolgáltatásokat nyújt webhelyekhez és alkalmazásokhoz. Használata teljesen ingyenes, egy rövid regisztrálás után már használható is.



2. kép Cloudinary használata a kódban



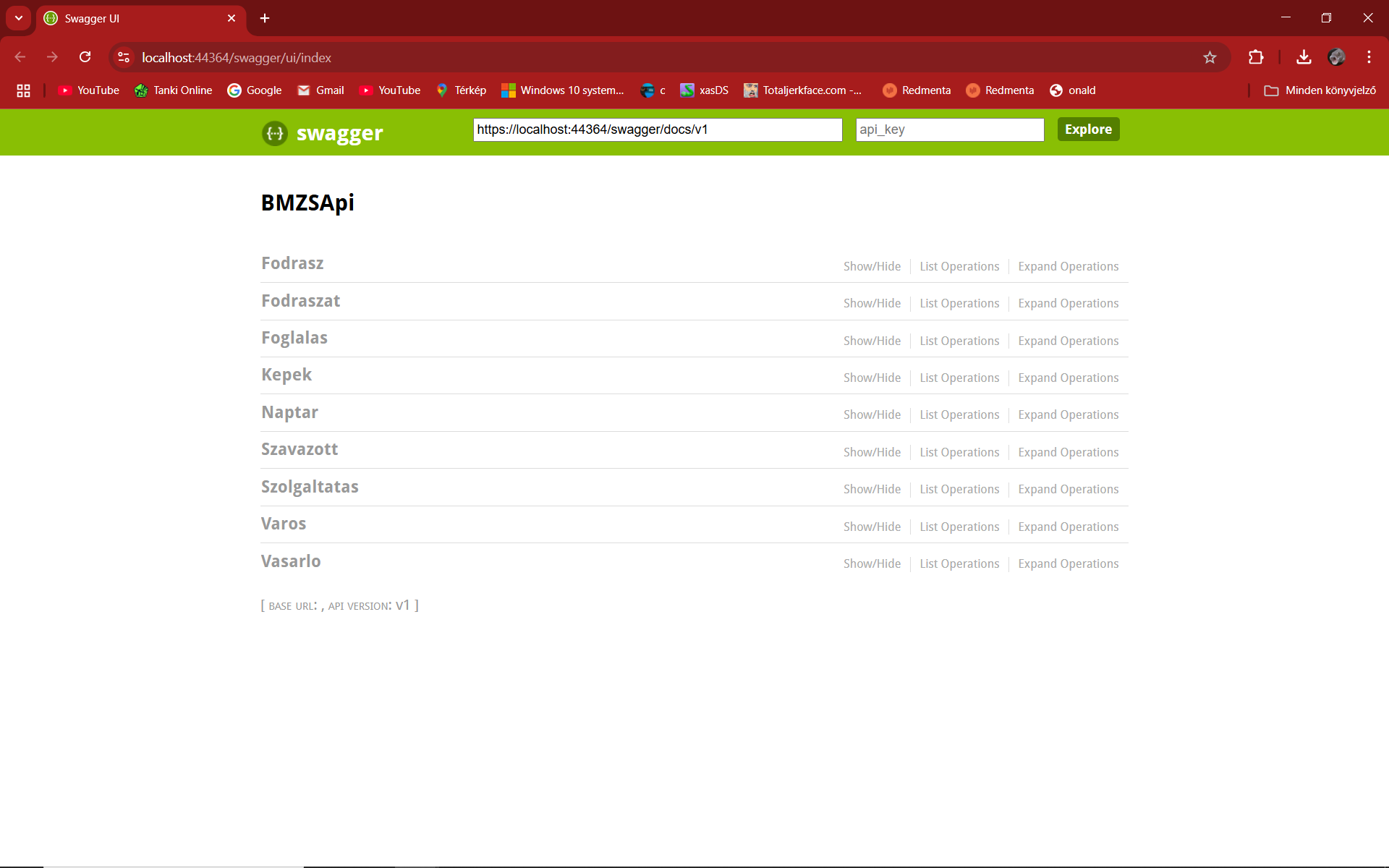
3. kép Felhőben eltárolt képek

Olyan esetekre, ha offline lenne az adatbázis, az adatbázis exportált változatát importálni tudjuk egy lokális xampp szerverre, ami ugyanúgy kiszolgálja a projektünket. Sajnos ilyen esetben a felhőben eltárolt képeket a felhasználó nem éri el.

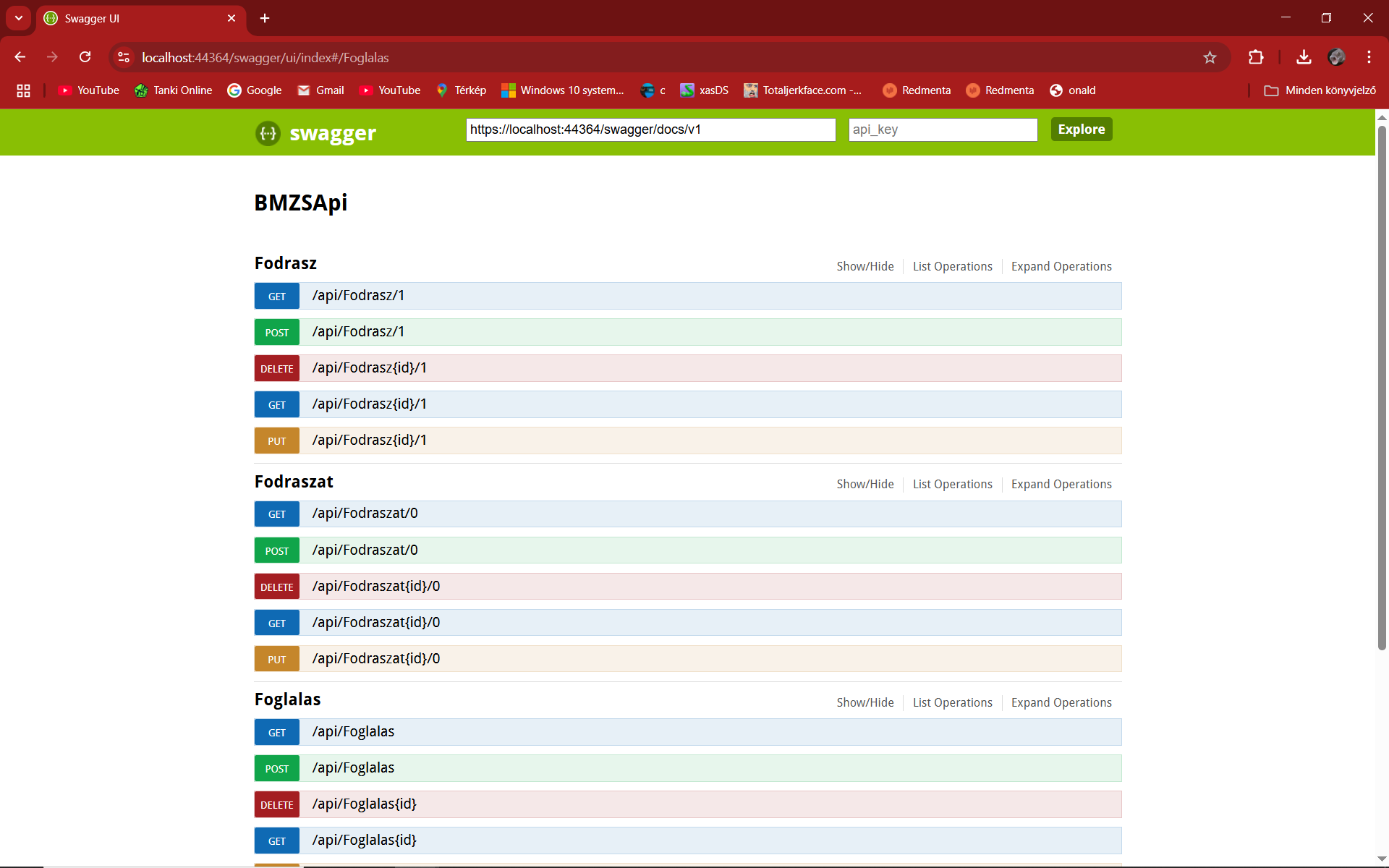
Projektünk API-ja egy ASP.NET Web Application-ben készült, .NET Framework keretrendszerrel. Ez könnyíti meg a kapcsolatot a felhasználó és az adatbázis között. Minden módosítást, új elem felvételét, elemek listázását ezen keresztül érjük el, és küldjük a felhasználónak. Adatbázisunk mind a 9 tábláját itt kezeljük.

Az átláthatóság miatt Swagger-t használunk, ami egy grafikus felületté alakítja a megírt adatbázis táblák controller-jeit, és könnyedén kezelhetjük a kontrollereké metódusait, mint a listázást (get), adatfelvitelt (post), adatmódosítást (put), adattörlést (delete) illetve az egyéni metódusok, mint az Autentikáció vagy a felhasználói adatok módosítása.

Alábbi képeken látható a Swaggernek a felülete, ahol látható a backendünk „táblái” (controllerei), illetve metódusai.



4. kép Swagger felületen megjelenő táblák



5. kép Swagger, Controllerek metódusai

Weboldalunk létrehozására React keretrendszert használtunk, aminek komponenseit visual studio code kódszerkesztőben készült.

A frontenden a foglalásnál az emailjs nevű ingyenes email-küldő szolgáltatás használtunk, míg a kapcsolattartásnál egy discord webhookot ami elküldi a felhasználó üzenetét a discord szerverünkre. A weblapon található egy csúsztatható lista és egy kereséses legördülő lista amit a react-slick és a react-select, míg az ikonokat a react-icons csomagból használtuk.

A hajakról és fodrászokról készült képeket a Pinterest nevű ingyenes és publikus képmegosztó platformról töltöttük le. <https://hu.pinterest.com>

Asztali alkalmazásunk WPF App-ban készült, .NET Framework keretrendszerrel, Visual Studio 2022-es fejlesztői környezettel.

Az alkalmazásban a NuGet Package Manager segítségével társítottunk megbízható forrásból származó bővítményeket. Használtuk például a MahApps.Metro-t, ami letisztult és modern design-t biztosít az applikáció több elemének, például gomboknak, a ComboBox-oknak és a DataGrid-nek is. Ezen kívül az ablakok fejlécét és keretét is egységesíti és különleges megjelenés biztosít nekik. Használtuk még ezenkívül az Xceed.Wpf.Toolkit bővitményt is, ami új és az alap 4.7.2-es verziójú .NET-ben nem megtalálható, könnyen kezelhető komponenseket biztosít. Ilyenek például a TimePicker-ek, amik segítségével megfelelő formátumban, egy óra megjelenítésével még nagyon felhasználóbarát módon jelenítik meg a választott időt.

## 2.1 Rendszerkövetelmények (minimum)

RAM: 4GB, Processzor: 2 vCPU, OP rendszer: Windows 10, Monitor: 1920x1080 (ajánlott), Videókártya: Nvidia 950 (nem kötelező), Böngésző: Google Chrome, .Net verzió: 4.7.2, telepített Xampp, telepített Node.js

# 3. Projekt használatának rövid bemutatása

Első lépésként a **backend**et kell elindítani, mivel ez szolgálja ki, mint a frontendet, mint az asztali alkalmazást. Ezt a projekt mappájában található „ApiStarter.exe” elindításával megtehető. Sikeres futtatás esetén a konzolban megjelenik az API url-je arra az esetre, ha böngészőből akarjuk elérni.

**Weboldalunk** megnyitásakor egy üdvözlő oldal fogad, amely röviden bemutatja a fodrászatunkat. Innen a fenti menü segítségével tovább lephetünk a **profil**, **galéria,** illetve a **foglalás** oldalra.

A **bejelentkezés** oldalon a felhasználó be tud jelentkezni amennyiben hozott már létre fiókot. Ha nem akkor regisztrációs lehetősége is van, ahol elkérjük a felhasználó E-mail címét, telefonszámát, felhasználónevét, teljes nevét, illetve egy jelszót, amit SHA256-os protokollal titkosítunk.

Egy SHA256-os protokoll által titkosított jelszó:

cAquo1ahr1sipX+tFiGHCVjqdthIlZnhvNh76zSPe1o=

A **galéria** oldalra átmenve a fenti legördülő lista segítségével ki lehet választani egy fodrászatot és alatta megjelennek a fodrászathoz tartozó fodrászok és képei. A képek alatt lehetősége van a felhasználónak szavazni a képekre 1-5-ig csillagok formájában. Csak bejelentkezett felhasználók tudnak szavazni a képekre.

A **foglalás** oldalon egy négy lépcsős foglalási rendszer fogadja a felhasználókat. Első lépésben kiválasztja a várost, második lépésben a fodrászatot, a harmadik lépésben az időpontot és a fodrászt, negyedik lépésben pedig a szolgáltatást. Végül egy összegző képet kapunk, ahol véglegesíthetjük a foglalást.

Az **asztali alkalmazás**unkban bejelentkezés után a fodrászatot kezelő felhasználó ki tudja listázni az adott fodrászatban lévő fodrászokat és beosztásaikat, a vásárlókat és foglalásaikat, a városokat, szolgáltatásokat, a naptárban eltárolt dátumokat és képeket. Az applikációban lehetőség van az előzőleg felsorolt csoportok tagok törlésére és adatainak módosítására. Új adat felvételére az ÚJ gomb megnyomása után van lehetőség a felugró ablakban.

# 4. Szerepkörök, Feladatok felosztása

|  |  |
| --- | --- |
| Szerepkör | csoporttagok |
| Backend (API) | Soós Benjámin |
| Frontend | Zanati Zsolt |
| Asztali alkalmazás | Őri Máté |
| Időfelelős | Őri Máté |

|  |  |
| --- | --- |
| FELADATOK | Csoporttagok |
| Témaválasztás | Soós Benjámin, Őri Máté, Zanati Zsolt |
| Adatbázis megtervezése | Soós Benjámin, Őri Máté, Zanati Zsolt |
| Weboldal tervezés, monitorképek rajzolása | Soós Benjámin, Őri Máté, Zanati Zsolt |
| Adatbázis fejlesztése | Soós Benjámin |
| Frontend fejlesztése | Zanati Zsolt |
| Asztali alkalmazás fejlesztése | Őri Máté |
| Backend (API) fejlesztése | Soós Benjámin |
| Backend indító konzolos alkalmazás | Soós Benjámin |
| Backend tesztelése (Postman) | Soós Benjámin |
| Frontend tesztelése (Katalon Studio) | Zanati Zsolt |
| Adatok és képek gyűjtése az adatbázisba | Soós Benjámin, Őri Máté, Zanati Zsolt |
| Dokumentáció megírása | Soós Benjámin, Őri Máté, Zanati Zsolt |

# 5. Határidőnaplók

## 5.1 adatbázis

|  |  |  |
| --- | --- | --- |
| Feladat | határidő | Csoporttag(ok) |
| Adatbázis megtervezés | 2024.10.10. | Soós Benjámin, Őri Máté, Zanati Zsolt |
| Adatbázis létrehozása | 2024.10.15. | Soós Benjámin |
| Adatbázis táblák létrehozása, kapcsolatok kiépítése | 2024.11.15. | Soós Benjámin |
| Adatbázis finomítása | 2024.11.20. | Soós Benjámin |
| Adatbázis összekapcsolása API-val | 2024.12.10. | Soós Benjámin |
| Felhő kialakítása, képek eltárolása | 2025.01.10 | Soós Benjámin |
| Adatbázis átvitele online térbe | 2025.01.30. | Soós Benjámin |
| Adatbázis teszt adatainak kiürítése | 2025.02.20. | Soós Benjámin |
| Adatbázis feltöltése | 2025.03.30. | Soós Benjámin, Őri Máté, Zanati Zsolt |
| Képek keresése, feltöltése | 2025.04.10. | Soós Benjámin, Őri Máté, Zanati Zsolt |

## 5.2 backend

|  |  |  |
| --- | --- | --- |
| Feladat | határidő | Csoporttag(ok) |
| Adatbázis teljes felépítése | 2024.11.20. | Soós Benjámin |
| api kialakítása | 2024.12.20. | Soós Benjámin |
| API kapcsolat kiépítése az adatbázissal | 2025.12.30. | Soós Benjámin |
| Api controllerek felépítése | 2025.01.10. | Soós Benjámin |
| API Controllerek metódusai kialakítása, Response messagek kialakítása | 2025.02.10. | Soós Benjámin |
| API metódusaik tesztelése | 2025.03.10. | Soós Benjámin |
| API összekapcsolása frontendel | 2025.03.15. | Soós Benjámin, Zanati Zsolt |
| API összekapcsolása az asztali alkalmazással | 2025.03.20. | Soós Benjámin, Őri Máté |

## 5.3 asztali alkalmazás

|  |  |  |
| --- | --- | --- |
| Feladat | határidő | Csoporttag(ok) |
| Alkalmazás felépítésének alapvető megtervezése | 2024. 12. 20. | Őri Máté |
| Kezdőlap felületének felépítése | 2025. 01. 31. | Őri Máté |
| Főoldal felületének felépítése | 2025. 02. 25. | Őri Máté |
| Adatfelvitel oldal felületének felépítése | 2025. 02. 30. | Őri Máté |
| Főoldalon találhato DataGrid osztályonkénti kialakítása | 2025. 03. 10. | Őri Máté |
| API imlementálása a programban | 2025. 03. 20 | Őri Máté, Soós Benjámin |
| Kezdőoldalon authentikáció bevezetése | 2025. 03. 25. | Őri Máté |
| DataGrid adatainak módosítása gomb segítségével | 2025. 03. 31. | Őri Máté |
| DataGrid adatainak törlése gomb segítségével | 2025. 04. 05. | Őri Máté |
| DataGridbe történő új adatok felvitele az Adatfelvitel oldal segítségével | 2025. 04. 20. | Őri Máté |
| WPF project design-olása | 2025. 04. 31 | Őri Máté |

## 5.4 frontend

|  |  |  |
| --- | --- | --- |
| Feladat | határidő | Csoporttag(ok) |
| A weblap dizájn megtervezése | 2024.11.22. | Zanati Zsolt |
| A weblap komponenseinek megtervezése | 2024.12.10. | Zanati Zsolt |
| Főoldal kialakítása | 2024.12.20. | Zanati Zsolt |
| Felhasználói komponensek felépítése | 2025.01.10. | Zanati Zsolt |
| Foglalófelület kialakítása | 2025.02.15. | Zanati Zsolt |
| Galéria felépítése | 2025.03.20. | Zanati Zsolt |
| API összekapcsolása frontendel | 2025.03.15. | Soós Benjámin, Zanati Zsolt |
| Elemek reszponzívvá alakítása | 2025.04.10. | Zanati Zsolt |

# 6. Adatbázis felépítése

Adatbázisunkat xampp alkalmazással, azon belül phpMyAdmin segítségével készítettünk. Eltároljuk a fodrászatok elhelyezkedését, városokat, ahol tartózkodik a fodrászat, a fodrászatokban dolgozó embereket, a fodrászokhoz irányuló foglalásokat, a foglalásban résztvevő vásárlót és a szolgálatást, amit kért az adott személy. Ezen kívül a fodrászokhoz kapcsolódnak a képek, amiket a vágás után készítettek, illetve a vásárlók tudnak szavazni a képekre, 1-től, 5-ig csillagok formájában, és a fodrászok beosztását, amit egy napra tudnak írni. Dátum, és mettől meddig fog dolgozni.

Így tehát adatbázisunk 10 táblából áll:

Város, Fodrászat, Fodrász, Naptár, Foglalás, Szolgáltatás, Képek, Vásárló, Szavazott, Beosztás.

Az átláthatóság kedvéért létrehoztunk hozzá relációs modellt, illetve ER-modellt. Itt minden tárolt attribútumot, illetve kapcsolatot megjelenítünk. Az ER-modellt **Canva** segítségével készítettük. Ez egy ingyenes, internetes tervező felület.

## 6.1 Relációs modell

**Varos** (Id, IranyitoSzam, TelepulesNev)

**Fodraszat** (Id, Utca, HazSzam, Azonosito, VarosId)

**Beosztas** (Id, Datum, NapNeve, Kezdes, Vege, FodraszId)

**Fodrasz** (Id, Nev, TelefonSzam, Leiras, ProfilKep, FodraszatId)

**Naptar** (Id, Datum, FodraszId)

**Kepek** (Id, EleresiUt, KepNev, Leiras, FodraszId)

**Szavazott** (Csillag, KepekId, VasarloId)

**Szolgaltatas** (Id, SzolgaltatasNev, Idotartalom, Ar)

**Vasarlo** (Id, FelhasznaloNev, Nev, TelefonSzam, EMail, ProfilKep, Jelszo)

**Foglalas** (Id, NaptarId, SzolgaltatasId, VasarloId)

## 6.2 ER-modell

Mellékletben a „[BMZSBarberShop ER-modell](BMZSBarberShop%20ER-modell.png)” nevű kép.

# 7. Projekt használata

Következő képekben bemutatjuk a projektünk használatának lépéseit a webes, illetve alkalmazás felületén.

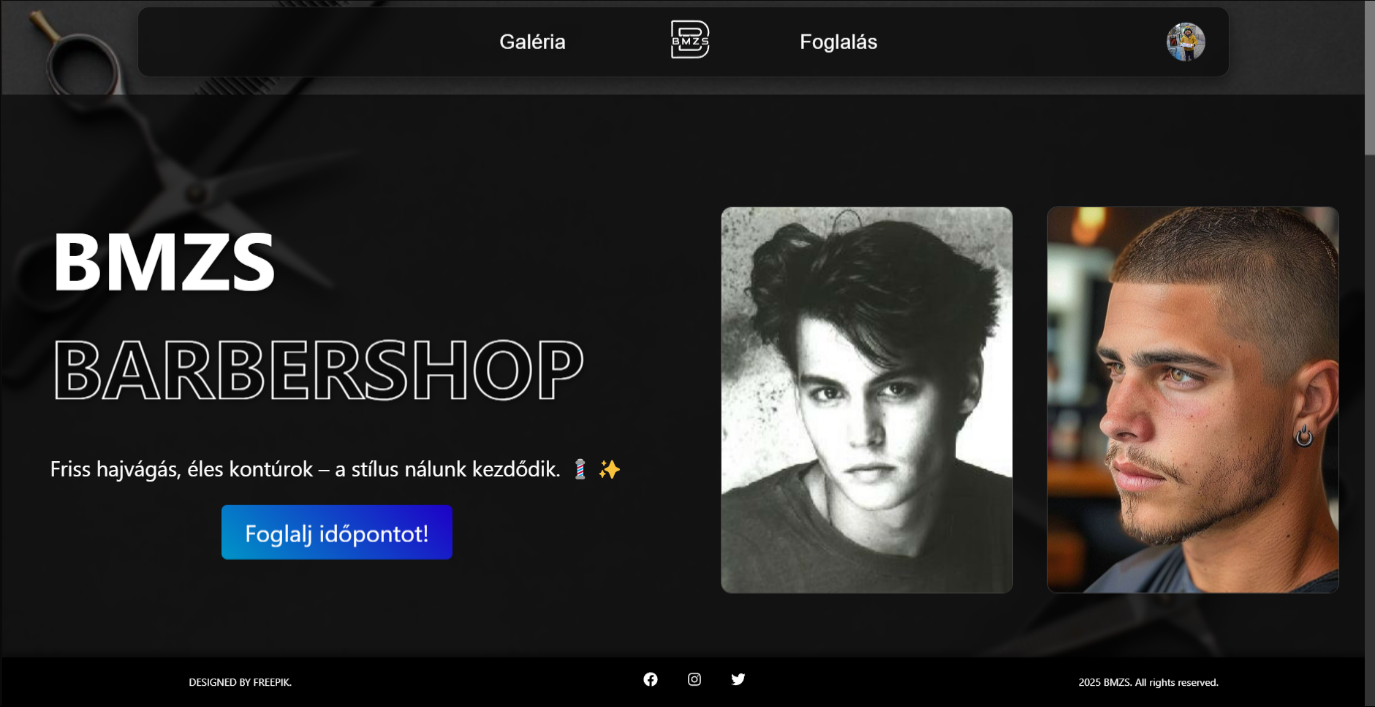
## 7.1 Weboldal használata

Az oldalra belépve a főoldal fogad minket, ahol az információkat találunk a nyitvatartásról, a szolgáltatásainkról, árakról, az elérhetőségekről, az elhelyezkedésről, illetve a közösségi média oldalainkról.



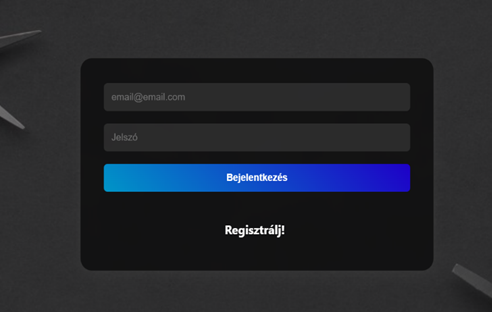
6. kép Fő oldal

A fent található navigációs bár segítségével tudunk navigálni a főoldal, a galéria, a foglalás és a felhasználó profilja között.



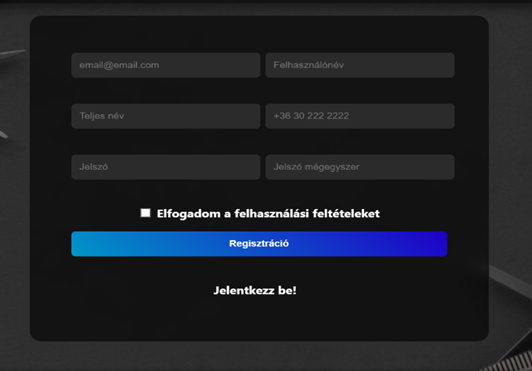
7. kép Navigációs bár

A profil ikonra nyomva (nav bár jobb szélén) ha a felhasználó nincsen bejelentkezve akkor a bejelentkezés oldalra fogja irányítani. Itt az az E-mail és jelszó megadásával betud lépni. Amennyiben a felhasználó nincs regisztrálva a „Regisztrálj!” gombra kattintva átirányítja a regisztrációs oldalra.



8. kép Bejelentkezési felület

A regisztrációs oldalon a felhasználónak megkell adnia az E-mail címét, felhasználónevét, teljes nevét, telefonszámát, jelszavát és még egyszer a jelszavát. Adatok megadása és a felhasználói feltételek elfogadása után a „Regisztráció” gombra nyomva a felhasználó sikeresen regisztrált amennyiben nincs hibás adata. Ha akármelyik adat hibás, az oldal tájékoztat róla. Sikeres regisztráció és belépés után a profil megtekintőbe fogja dobni a felhasználót.

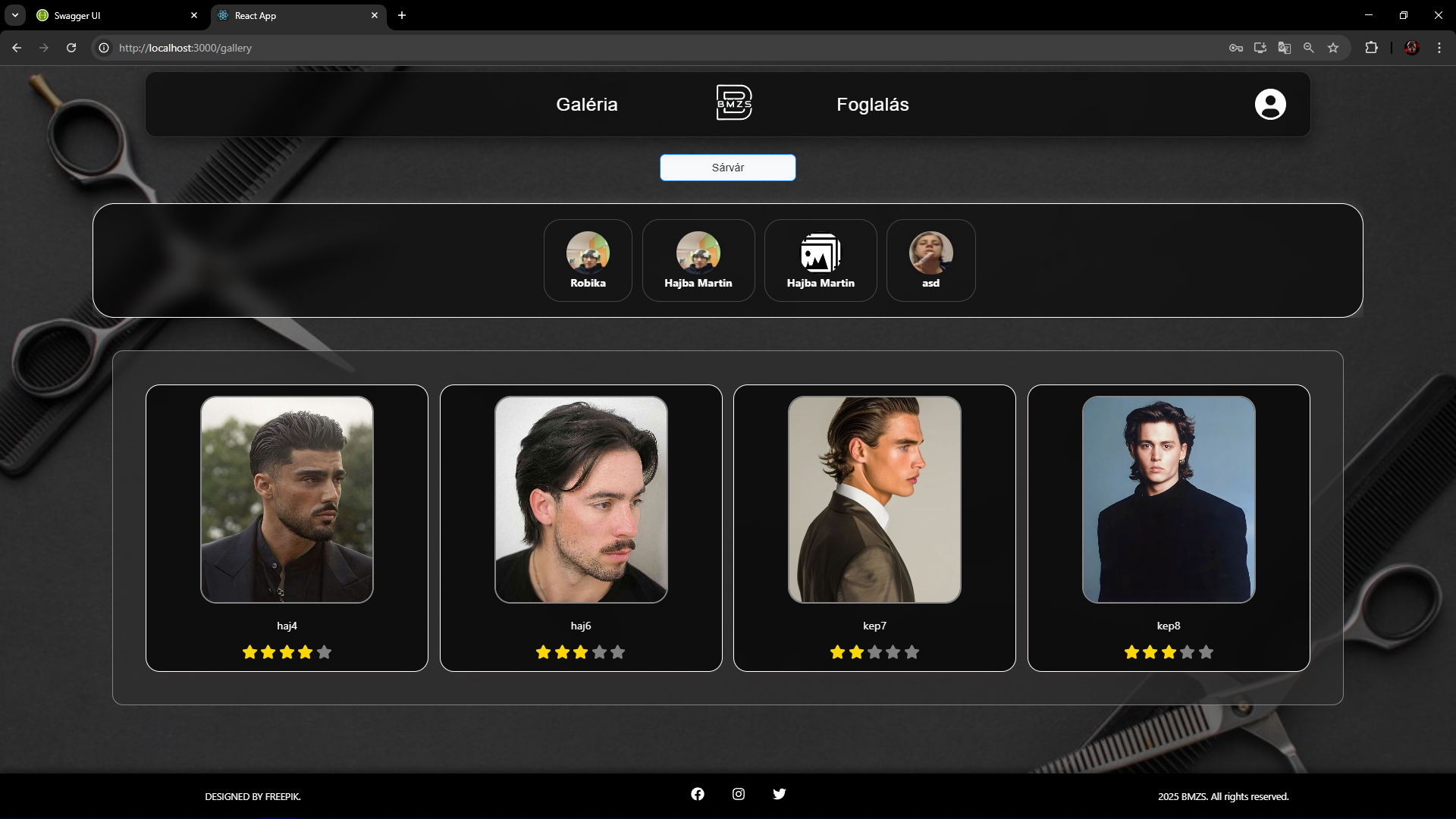


9. kép Regisztrációs felület

Itt megtalálható a felhasználónak az adatai, foglalásai. Itt van lehetősége kijelentkezni az oldalról, illetve módosítani az adatait.

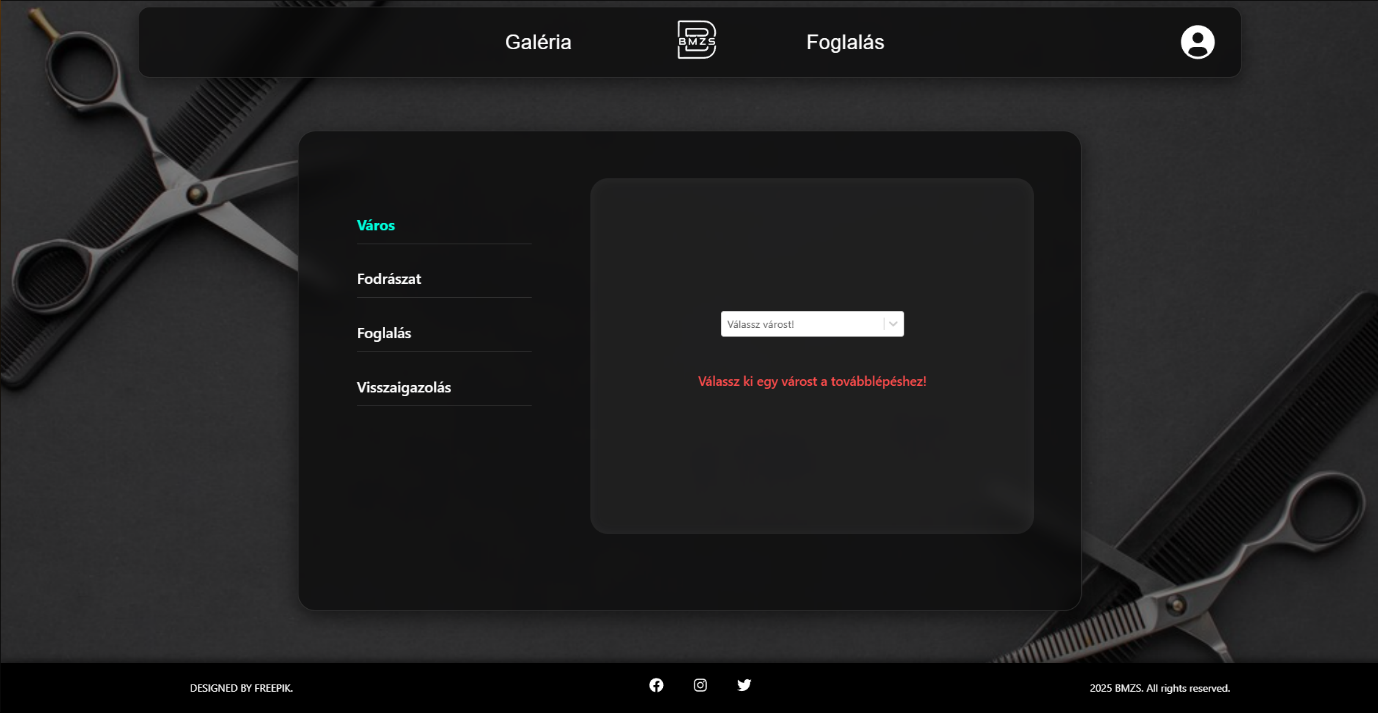
„Profil nézegető kép”

A galéria oldalra átmenve lehetőségünk van városok alapján kiválasztani a képeket. A képek alatt található csillagokkal lehet szavazni a képekre, amik a főoldalon is megjelennek, ha bent vannak a top 3 legkedveltebb képben.



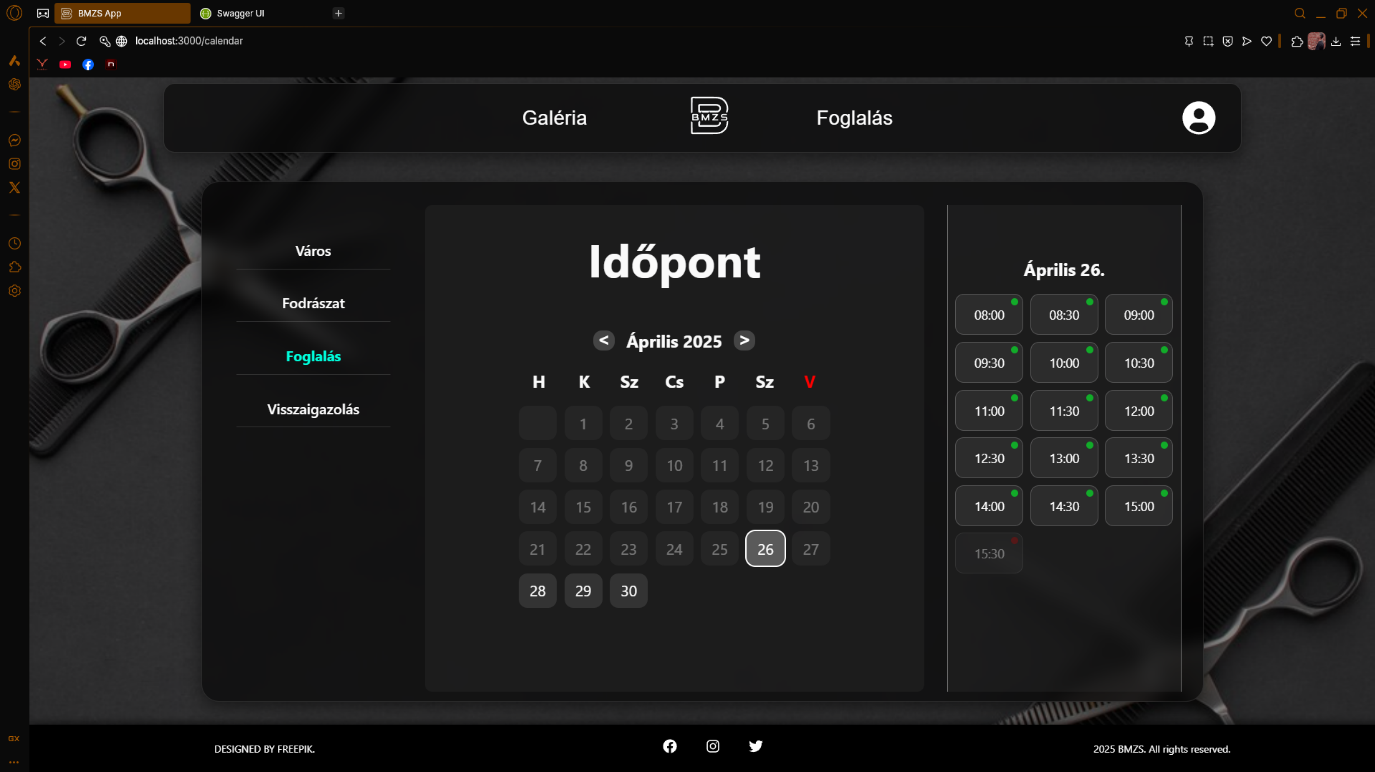
10. kép Galéria, szavazás felület

A foglalás oldalra átmenve egyesével tudjuk kiválasztani a foglalásunk paramétereit a bal oldalt található szövegek sorrendjében. Az aktuális fül mindig kék színnel jelzi az oldal. A város, illetve a fodrászat menüpontban egy legördüli lista segíti a választást.



11. kép Foglalás, város

A foglalás menüpontban egy naptárban tudjuk kiválasztani, hogy mikorra szeretnénk foglalni. Vasárnap kivételével minden nem szürke napra tud foglalni a felhasználó. A nap kiválasztása után megjelenik egy időtáblázat, ami a fodrász beosztásától változhat. Itt hasonlóan a napokhoz, a lefoglalt időpontok szürkén, illetve kis piros ponttal jelzi az oldal.



12. kép Időpont foglalás

Ha mindent kiválasztottunk akkor egy összegző oldalt kapunk, ahol letudjuk foglalni, illetve vissza tudunk menni, hogy megváltoztassuk a foglalás részleteit.

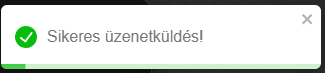


13. kép Foglalás összegző

Sikeres foglalás esetén a felhasználó egy E-mailt kap a megadott felhasználói adatokban szereplő E-mail fiókjára, illetve egy visszaigazoló zöld üzenetet.



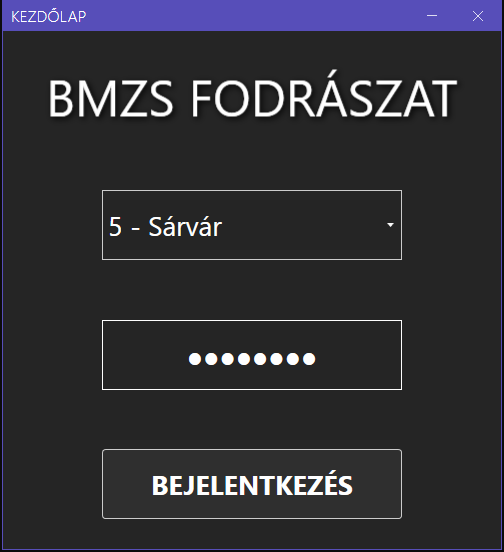
14. kép Foglalást visszaigazoló E-mail



15. kép Sikeres interakciók utáni üzenet

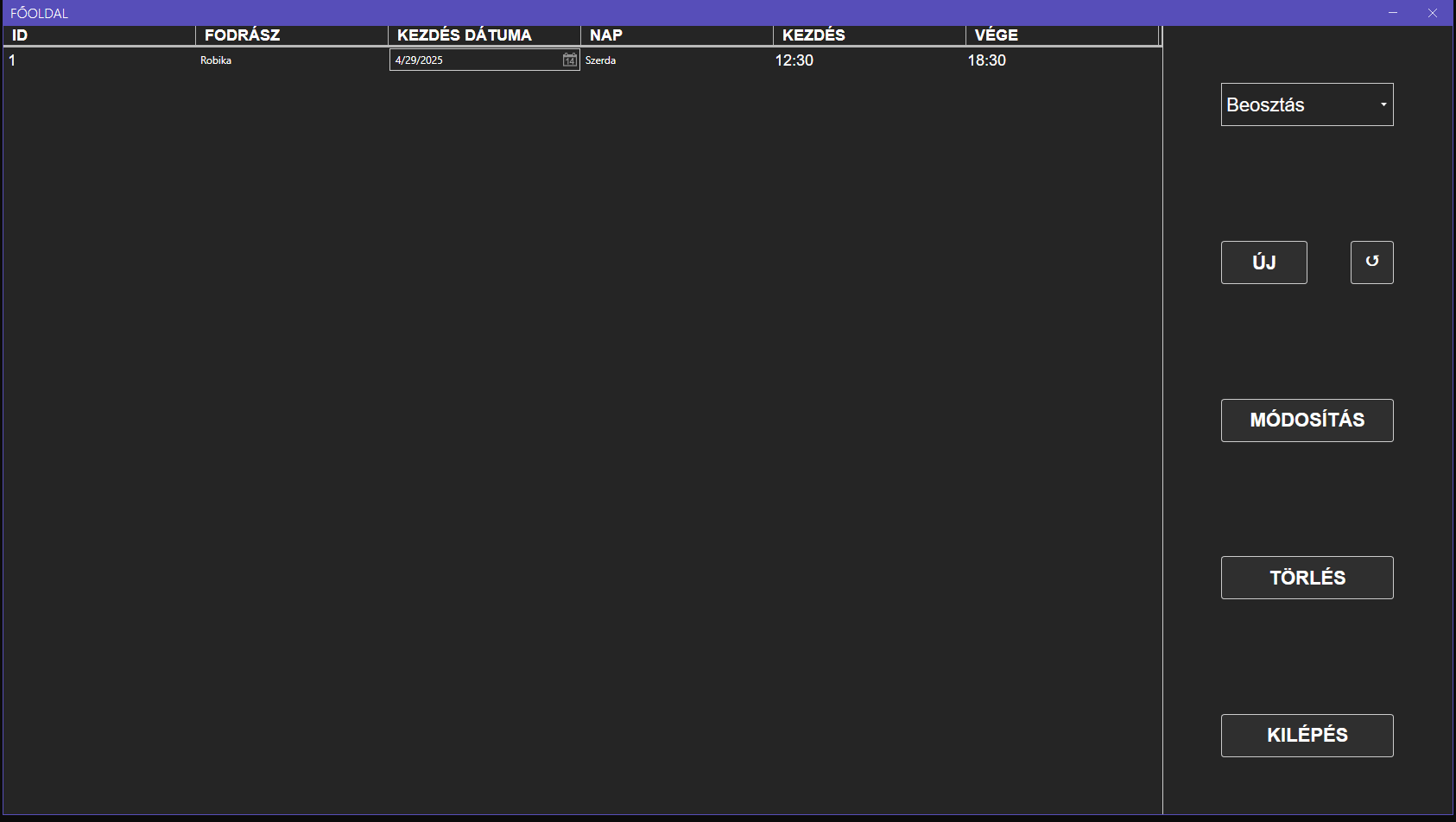
## 7.2 Asztali alkalmazás használata

Az alkalmazást elndítva a Kezdőoldal címet viselő ablak jelenik meg, itt az adatbázisban eltárolt fodrászatok közül ki tudjuk választani melyikbe szeretnénk bejelentkezni. Természetesen, csak azokba, amelyekhez ismerjük a megfelelő azonosítót.



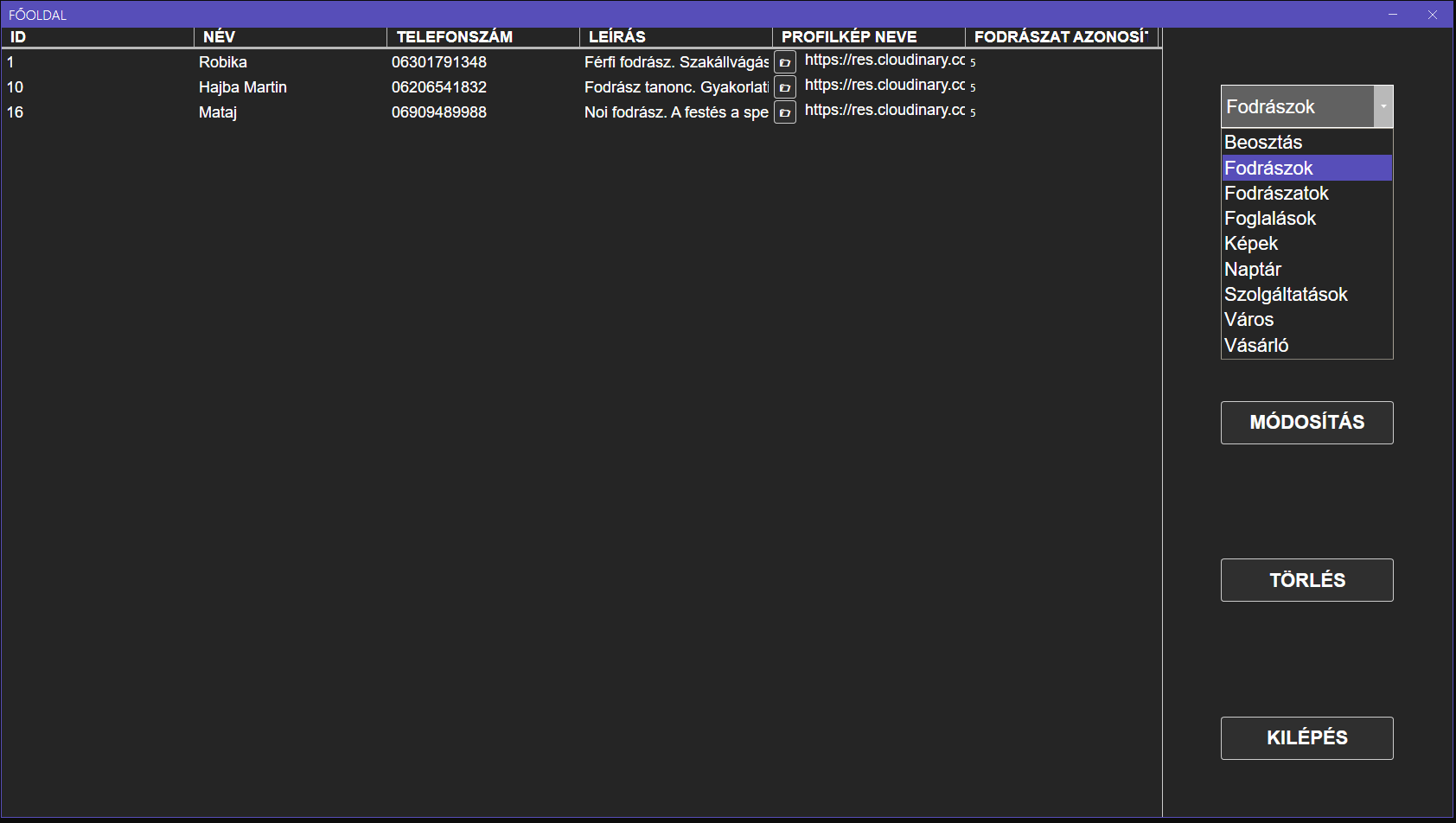
16. kép Kezdőlap

Sikeres bejelentkezés után a Főoldalra kerülünk, ahol az egyes táblák adatait láthatjuk egy DataGridben eltárolva. Ezeket az adatokat kedvünk szerint rendezhetjük a legfelül található, vastaggal kiemelt fejlécekre kattintva.



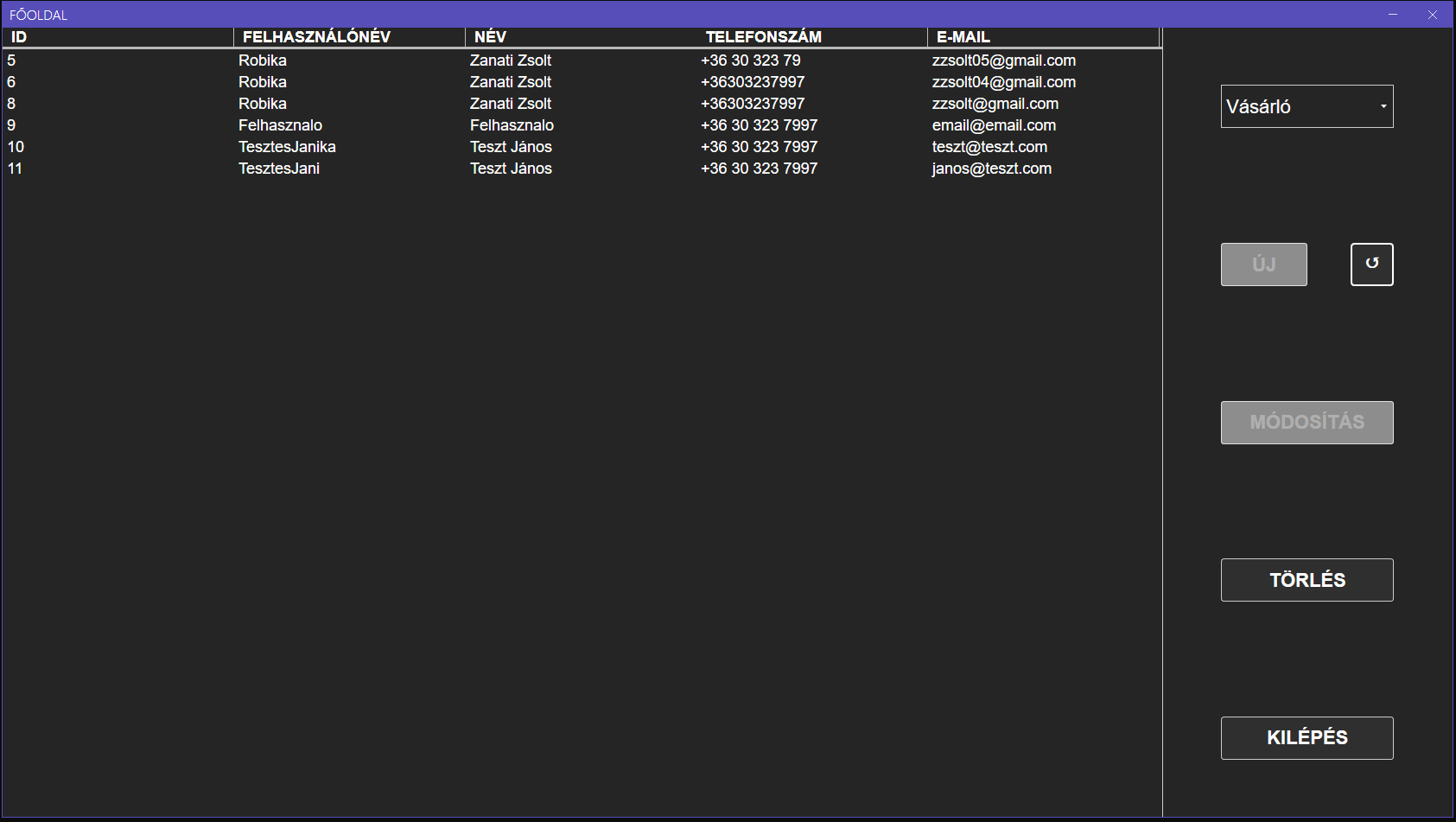
17. kép Főoldal és a Beosztások tábla adatai

A jobb felső sarokban található legördülő lista segítségével váltani tudunk a különbőző táblák között.



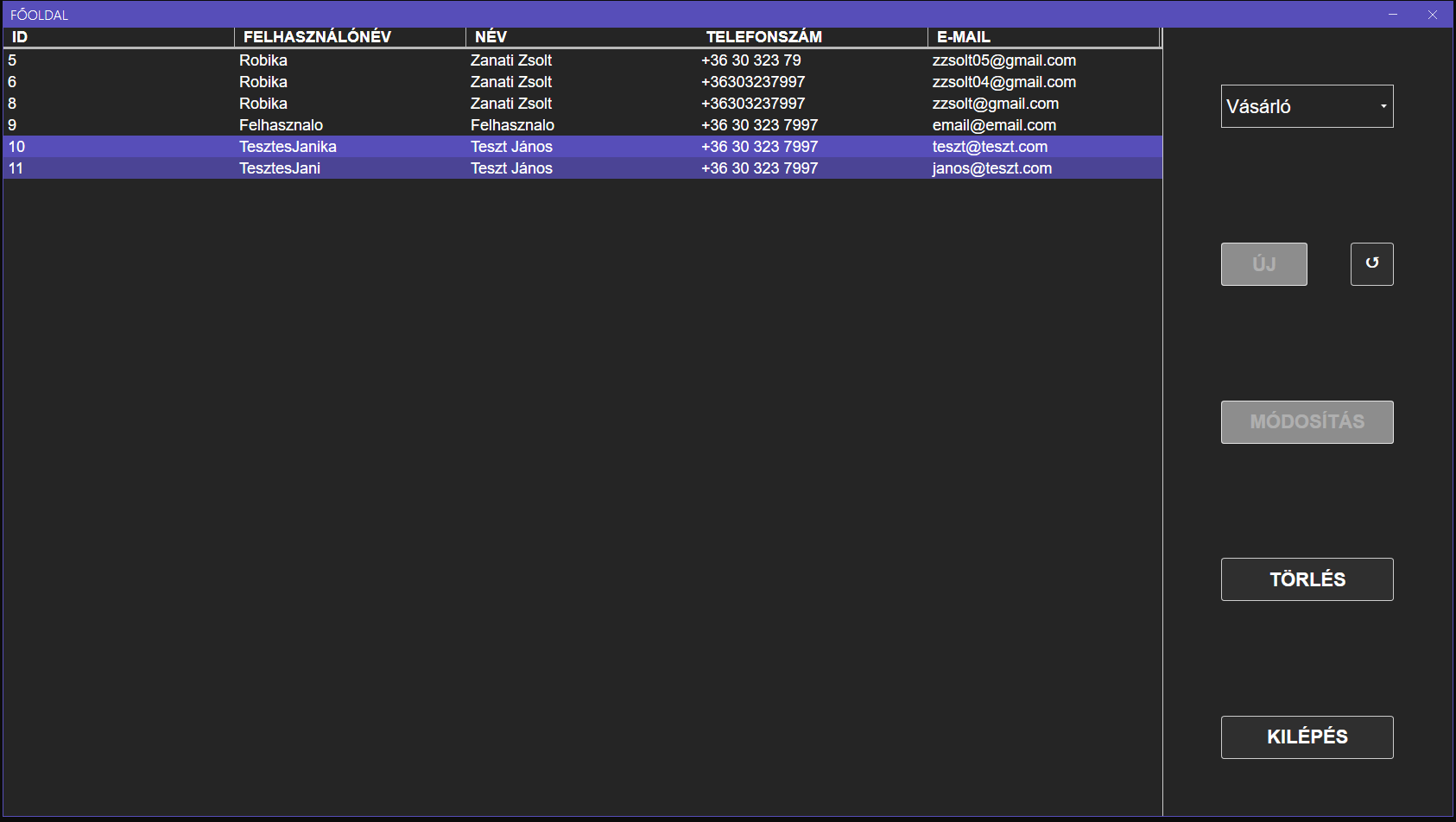
18. kép Főoldal és a Fodrászok tábla adatai

Minden adat módosítására lehetőségünk van a programban pár kivételen kívül. Az ID-kat, vagyis egyedi azonosítók módosítására nincs lehetőség, valamint a vásárlók adatainak frissítésére és új vásárló felvételére sem. A fodrászat törlésére sincsen lehetőség.



19. kép Vásárló tábla adatai és a kikapcsolt gombok

Az adatok törlése a következőképpen történik. A felhasználó kijelöl egy elemet, a sor bármely részére kattintva. Egy különböző szín ezt jelzi is. Ezután rákattint a törlés gombra és sikeresen törölte az adatot az adott táblából.



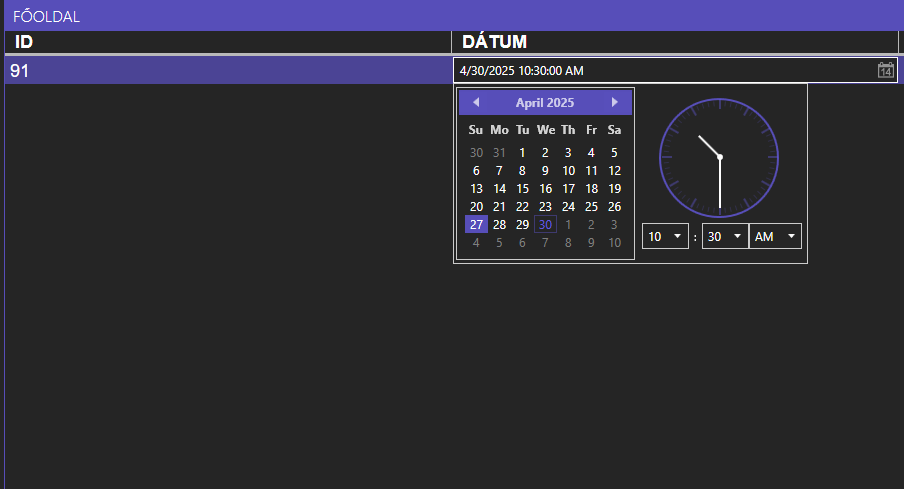
20. kép Adatok törlése a Vásárló táblában

A hibás adatok és az általános felhasználói figyelmetlenséget elkerülve bizonyos helyeket az adatbázisban eltárolt adatok közül lehet választani, például a Foglalásoknál a Szolgáltatások,



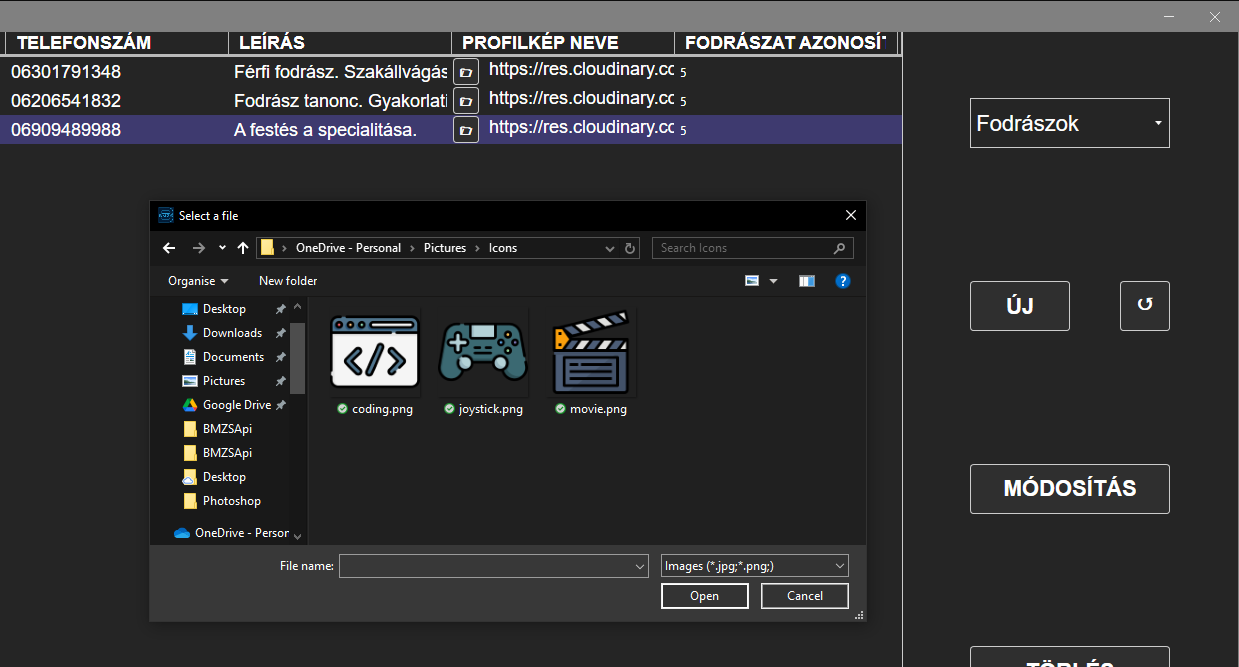
21. kép Foglalások tábla Szolgáltatás adatai

vagy mint a Naptár táblában a Dátum mezők. Dupla kattintás után meg is jelenik a kis ikon, aminek segítségével megjeleníthető a dátum és idő választó.



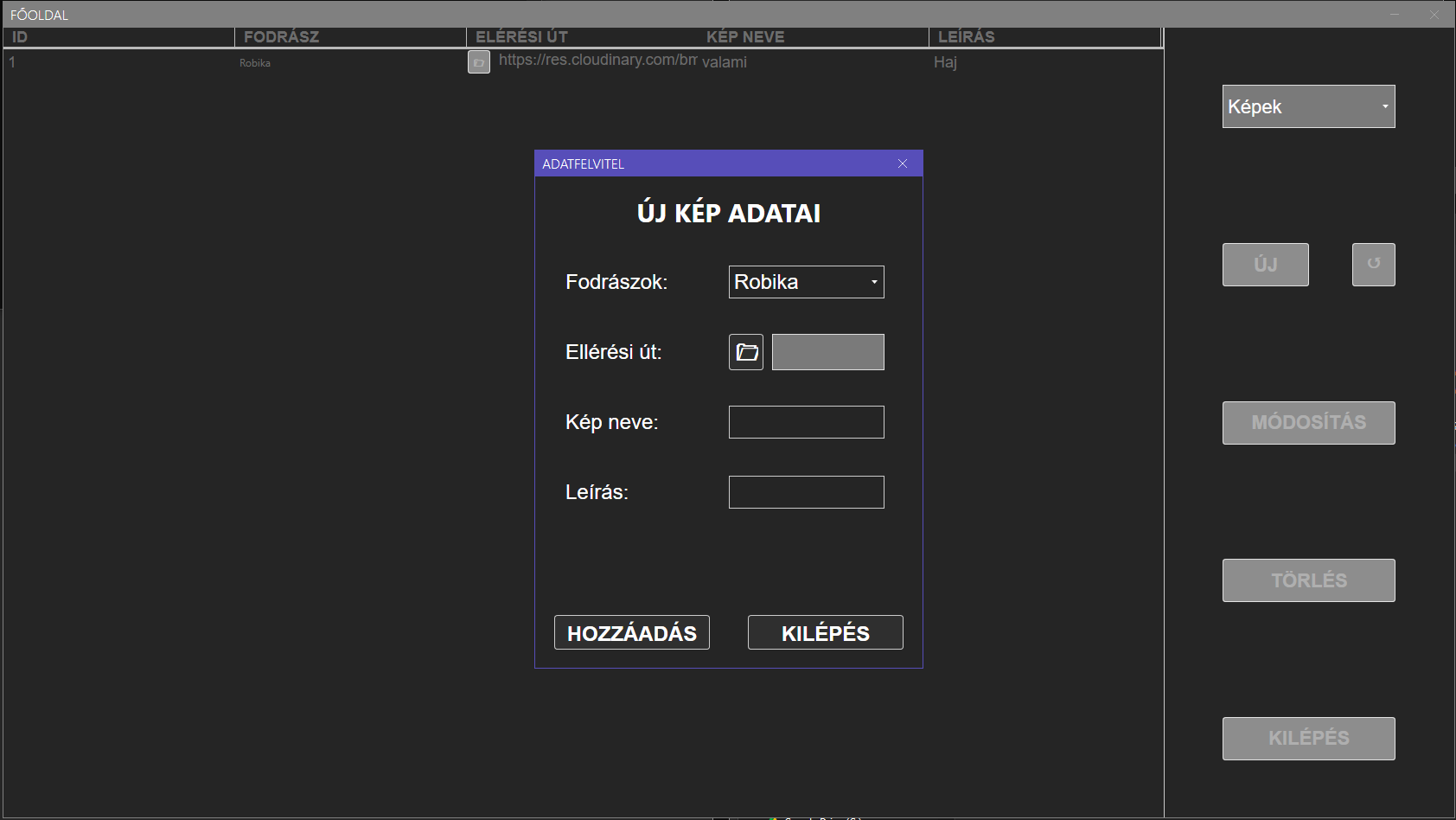
22. kép Naptár tábla Dátum mezője

A Fodrászok táblán még ilyen adatok a Profilképek, amiket a kis mappa ikonrakattinta, a felugró fájlkezelőből választhatunk ki. Ugyanígy működik a Képek táblában az Elérési Út kiválasztása is.



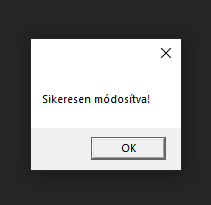
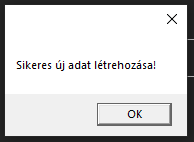
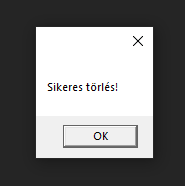
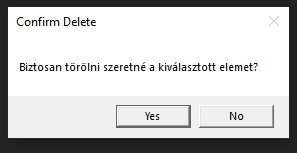
23. kép Fodrászok tábla és a fájlválasztás

Az ÚJ gombra kattintva megjelenik egy új ablak. Az ablakon a kiválasztott táblának megfelelően jelennek meg az adatok. Az adatok, ahogyan a DataGridnél, úgy itt is adott esetben választhatóak fix értékek közül. Amíg új adatot adunk hozzá, a Főoldal inaktívvá válik. Ez újra aktív lesz, miután bezártuk az oldalt.



24. kép Új adat hozzáadása a Képek táblához

Az adatok módosításáról, törléséről és újak esetén a felvitelről rendszerüzenetek tájékoztatnak.



25. kép Törlés előtti utolsó figyelmeztetés

26. kép Sikeres törlés

27. kép Sikeres adat felvitele

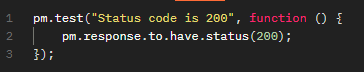
28. kép Sikeres adatmódosítás

# 8. Tesztdokumentáció

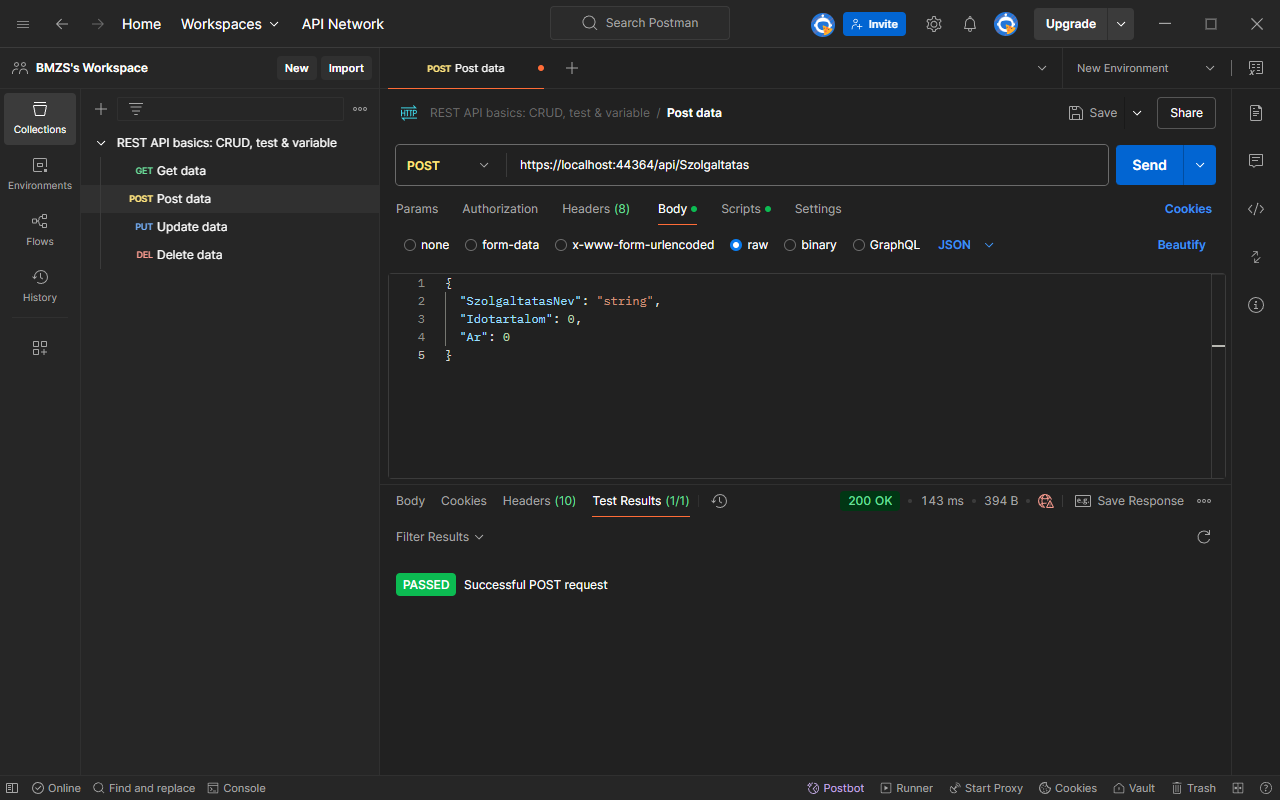
A szoftverfejlesztésben elengedhetetlen a tesztelés. Projektünk mind a 3 részét, különböző módszerekkel teszteltük a részeket. A backendet teszteket az ingyenes PostMan API tesztelővel végeztük. Itt egy test script megírása után, futtatva egy controller metódusát egyszerűen tesztelhetünk. A frontendet egy szintén ingyenes alkalmazással, a Katalon Studioval teszteltük. Az asztali alkalmazást belső, unittestekkel teszteltük. Itt minden tesztet mi írtunk egy megadott, elvárt eredményre.

## 8.1 Backend tesztelés

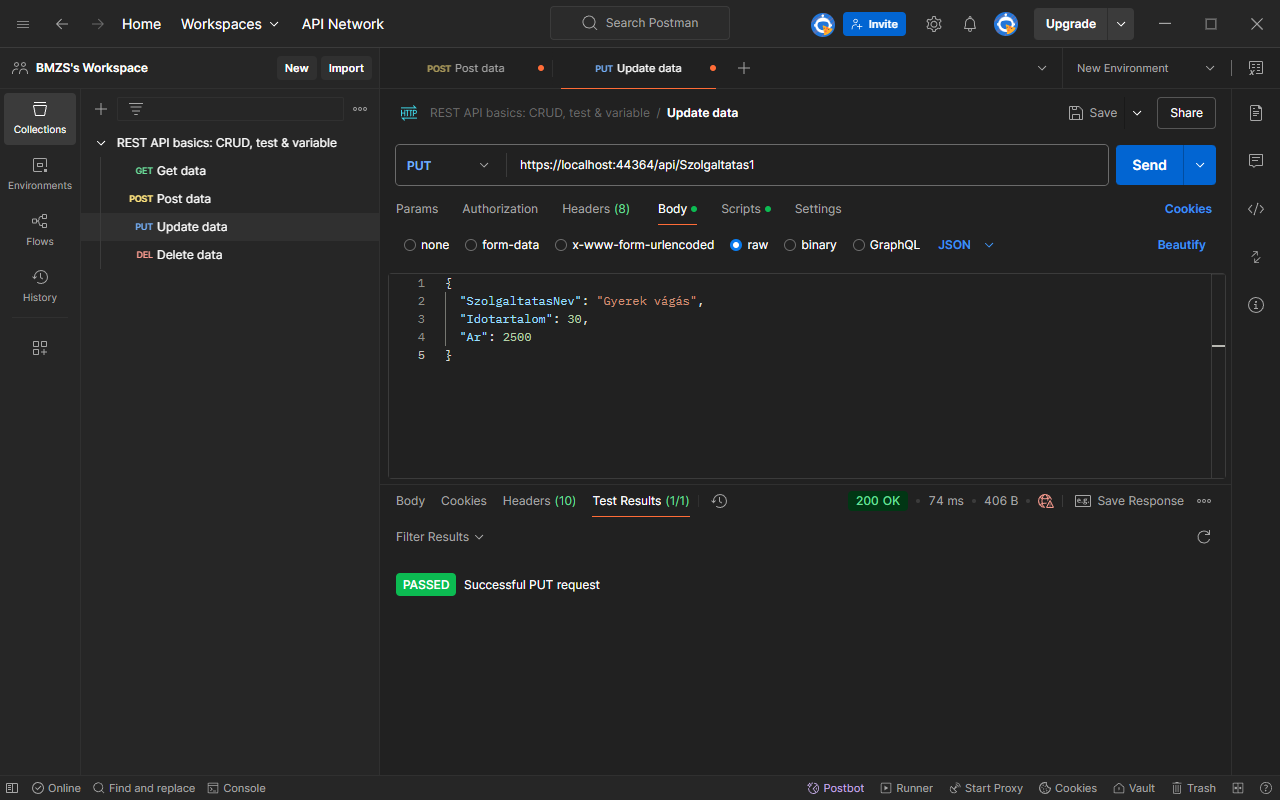
A következő példában a backendünk SzolgaltatasController metódusait teszteltük, a **PostMan** API tesztelő alkalmazással. Ebben a Controllerben egy GET, POST, PUT, DELETE metódus van. Minden esetben a visszatért adatokban a statuscodeot vizsgáltuk, hogy megfelel-e (200-as statuscode).



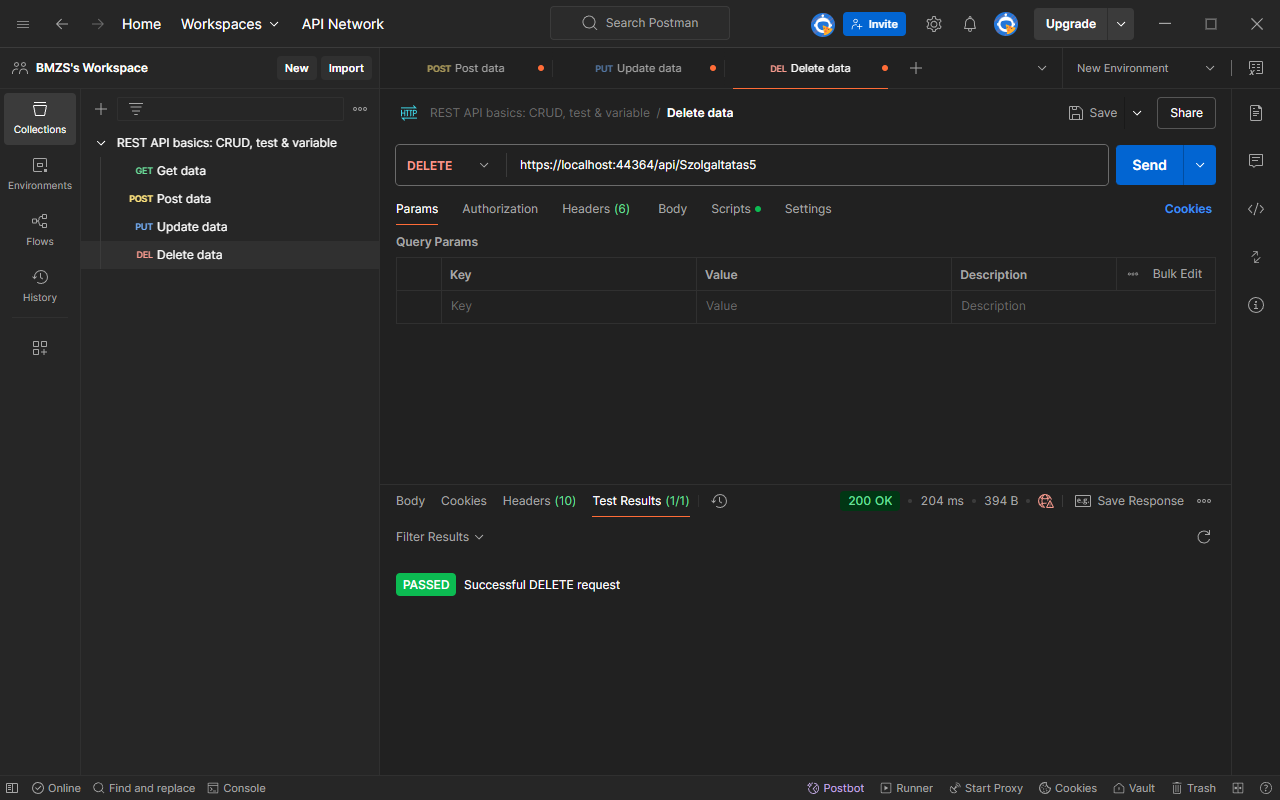
29. kép Teszt script



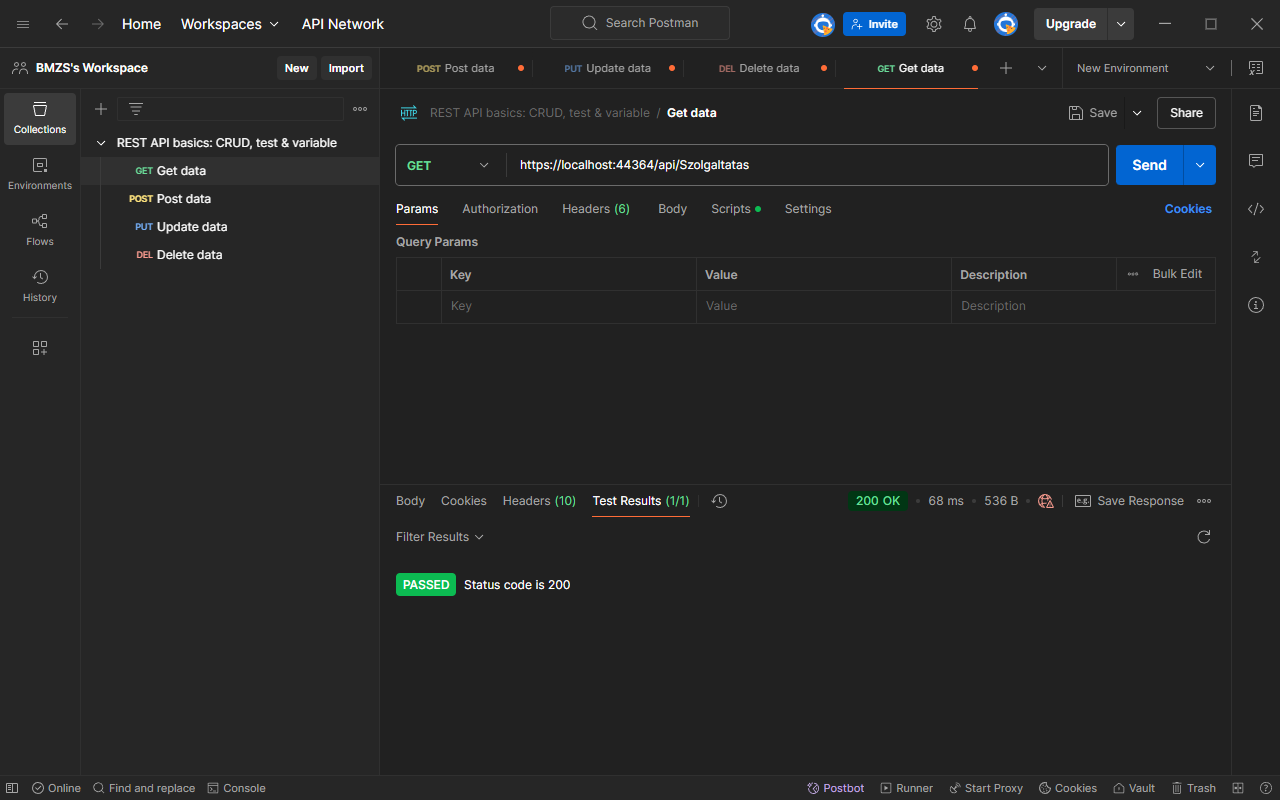
30. kép SzolgaltatasController POST metódusa



31. kép SzolgaltatasController PUT metódusa



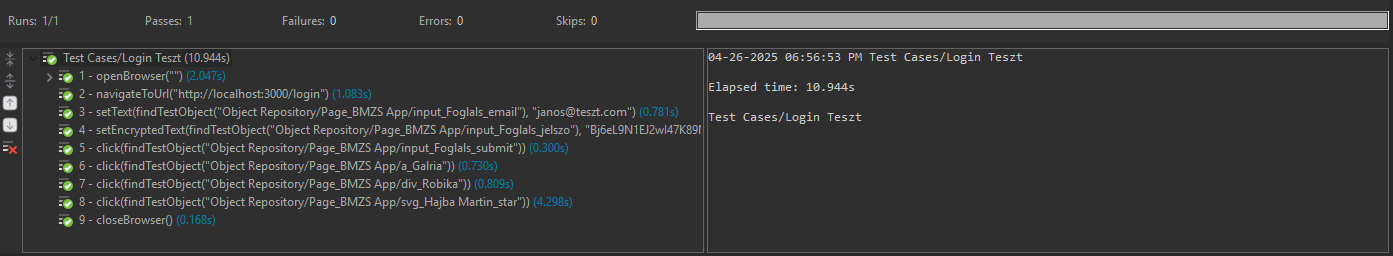
32. kép SzolgaltatasController DELETE metódusa



33. kép SzolgaltatasController GET metódusa

## 8.2 Frontend tesztelés

A weboldal teszteléséhez a **Katalon Studio** ingyenes tesztelő alkalmazást használtuk. Letöltés után egy teszteset létrehozása után a program a megadott lépéseket követve végig ment az oldalon. A képeken láthatóak a lépések. Például a kattintások, az átirányítások stb.



. kép Frontend teszt, belépés, szavazás

# 9. Bejelentkezés

Weboldalunkon a profil ikonra kattintva a következő, előre elkészített fiókba tud belépni:

E-mail cím:janos@teszt.com

jelszó: Teszt123!

(Ez az email nem létezik, a tesztelés során használtuk, mint minta felhasználó. Belépni ebbe az E-mail fiókba nem lehetséges!)

Asztali alkalmazásba való bejelentkezéshez a fodrászat azonosítója, illetve jelszavát kell megadni. Az egyszerűség kedvéért jelenleg az „Id” -t illetve az „Utca” és „Házszám” mezőket kell megadni a példához hasonlóan.

Id: 5 - Sárvár (választható)

Azonosító: Tesz123

# 10. Fejlesztést megkönnyítő kommunikációs eszközök

Mindennapos kapcsolattartáshoz elengedhetetlen volt, hogy különböző beszélgető és adatmegosztó alkalmazást használjunk. Beszélgetéshez legtöbbszőr a Discord-ot és Messenger-t használtunk, néha egypár kisebb méretű fájlt is ezeken osztottunk meg. Nagyobb méretű fájlok főleg a GitHub, illetve Google Drive felületeit használtuk.

## 10.1 Discord

A projekt kezdete óta a Discord alkalmazást használjuk, mint mindennapos kapcsolattartás. Itt minden lehetséges, ami egy projektmunkához kellhet. Beszélgető szobák, ahol akár képernyőt is meglehet osztani, chat szobák, amik fájl megosztásra is alkalmasak.

## 10.2 GitHub

A Github tökéletes volt az egymás közötti nagyobb fájlok megosztásában, illetve a fejlesztések jobb nyomon követesére a verziókezelőnek köszönhetően.

## 10.3 Google Drive

A Google Drive főleg iskolában használtuk, hogy az ottani gépeken elmentett munkát valahogy otthon is elérjük ha nem volt nálunk pendrive.

# 11. Indítási útmutató

A projekt futtatásához szükséges internet kapcsolat!

Az adatbázisunk online található, illetve az oldalon található képeket is online, felhőből tölti be a weboldal.

A webes alkalmazást és az API-t a projekt mappájában található „start.bat” script elindításával teheti meg.

Az asztali alkalmazást a BMZSApiHandler parancsikonnal lehet elindítani a projekt mappájából.

Amennyiben offline lenne az adatbázis a következő lépésekre van szükség.

Nyissa meg gépén a XAMPP Control Panel nevű alkalmazást, indítsa el a MySQL-t és Apache-t (actions: start). Ha elindultak, (Module oszlopban mind a kettő zöld) nyomjon MySQL sorban az Adminra. Egy böngészős ablakban megnyílt kezelőfelületben nyomjon az „Importálás” gombra a fenti menüben. Nyomjon a „Fájl kiválasztása” gombra és a projekt mappájában lévő „bmzsbarbershop.sql” nevű fájlt importálja. Görgessen le, és az oldal alján lévő „Importálás” gombbal importálja az adatbázist. Sikeres importálás után a bal oldalt lévő adatbázisoknál megkell jelenjen a „bmzsbarbershop”.  
Amennyiben nincs a gépén XAMPP töltse le a következő linkről: <https://www.apachefriends.org/download.html>

Sikeres importálás után a web.config, illetve az app.config fájlban vegye ki a <!-- --> jelek között található: <add name="BMZSContext" providerName="MySql.Data.MySqlClient" connectionString="server=localhost;port=3306;database=bmzsbarbershop;uid=root;password=" /> taget és a felette lévő hasonló taget tegye comment jelek közé ( <!-- --> ). Győződjön meg arról, hogy a tagen belül a connectionString port részénél megegyezik a XAMPP portjával. Általában 3306-os port, de meglehet, hogy 3307-es indult el.

# 12. Továbbfejlesztési lehetőségek

Oldalunkat továbbfejlesztve lehetőség lesz a fodrászokra véleményt írni, értékelni.

Az asztali alkalmazásban a MessageBox-ok design-jának egységesítése az oldalokkal.

# 14. Ábrajegyzék

[1. kép freesqldatabase.com oldalon található adatbázisunk 5](#_Toc196689594)

[2. kép Cloudinary használata a kódban 5](#_Toc196689595)

[3. kép Felhőben eltárolt képek 6](#_Toc196689596)

[4. kép Swagger felületen megjelenő táblák 7](#_Toc196689597)

[5. kép Swagger, Controllerek metódusai 7](#_Toc196689598)

[6. kép Fő oldal 15](#_Toc196689599)

[7. kép Navigációs bár 15](#_Toc196689600)

[8. kép Bejelentkezési felület 16](#_Toc196689601)

[9. kép Regisztrációs felület 16](#_Toc196689602)

[10. kép Galéria, szavazás felület 17](#_Toc196689603)

[11. kép Foglalás, város 17](#_Toc196689604)

[12. kép Időpont foglalás 18](#_Toc196689605)

[13. kép Foglalás összegző 18](#_Toc196689606)

[14. kép Foglalást visszaigazoló E-mail 19](#_Toc196689607)

[15. kép Sikeres interakciók utáni üzenet 19](#_Toc196689608)

[16. kép Kezdőlap 20](#_Toc196689609)

[17. kép Főoldal és a Beosztások tábla adatai 20](#_Toc196689610)

[18. kép Főoldal és a Fodrászok tábla adatai 21](#_Toc196689611)

[19. kép Vásárló tábla adatai és a kikapcsolt gombok 21](#_Toc196689612)

[20. kép Adatok törlése a Vásárló táblában 22](#_Toc196689613)

[21. kép Foglalások tábla Szolgáltatás adatai 22](#_Toc196689614)

[22. kép Naptár tábla Dátum mezője 23](#_Toc196689615)

[23. kép Fodrászok tábla és a fájlválasztás 23](#_Toc196689616)

[24. kép Új adat hozzáadása a Képek táblához 24](#_Toc196689617)

[25. kép Törlés előtti utolsó figyelmeztetés 24](#_Toc196689618)

[26. kép Sikeres törlés 24](#_Toc196689619)

[27. kép Sikeres adat felvitele 24](#_Toc196689620)

[28. kép Sikeres adatmódosítás 24](#_Toc196689621)

[29. kép Teszt script 25](#_Toc196689622)

[30. kép SzolgaltatasController POST metódusa 25](#_Toc196689623)

[31. kép SzolgaltatasController PUT metódusa 26](#_Toc196689624)

[32. kép SzolgaltatasController DELETE metódusa 26](#_Toc196689625)

[33. kép SzolgaltatasController GET metódusa 27](#_Toc196689626)

[34. kép Frontend teszt, belépés, szavazás 27](#_Toc196689627)