**Hotel HUB**

***Dokumentáció***

***Kun Szabolcs / Fazekas Tibor***

## Telepítési/használati instrukciók

### Frontend/Backend

Bun telepítése

* A Bun telepítéséhez Powershellben kell futtatni az alábbi sort:

**powershell -c "irm bun.sh/install.ps1 | iex"**

Dependencies letöltése

* A gyökér mappát kiválasztva a következő parancs: **bun i**

letölti a gyökérmappában lévő függőségeket (concurrently).

* A gyökér mappát kiválasztva **bun ins** paranccsal letölti a használt függőségeket mind a backendhez és a frontendhez.

Szerver elindítása

* A gyökér mappát kiválasztva **bun dev** paranccsal a frontend és backend szerver elindul.

### Asztali alkalmazás

Telepítés

* A Git Repositoryban lévő **HotelAdminPanelSetup.exe** fájl automatikusan telepíti az admin panelt.
* Alapból az installer a **Dokumentumok** mappába települ, ha változtatás nélkül települ.

### Teszt felhasználók

Admin

* Email: **admin**
* Jelszó: **admin**

Staff

* Email: **staff**
* Jelszó: **staff**

Guest

* Email: **guest**
* Jelszó: **guest**

### Adatbázis

Elérhetőségek

* Az adatbázis 2 helyen elérhető, lokálisan és online.
* Online
  + host: mysql-2632cb1f-tibcso0322-7da0.c.aivencloud.com
  + user: avnadmin
  + password: AVNS\_JsONj4Ni1IT9ObgF\_z4
  + database: hotels
  + port: 27087
* Local:
  + host: localhost
  + user: root
  + password: root
  + database: hotels

## A projektről

### Bevezetés

Ez a dokumentáció a szállodafoglaló oldal fejlesztésének részleteit tartalmazza. A projekt célja egy online platform kialakítása, ahol felhasználók szállodai szobákat foglalhatnak, a háttérben pedig egy adminisztrációs felület biztosítja az adatbázis-kezelést és a foglalások menedzselését. A rendszer három fő komponensből áll: egy **React** alapú frontendből és a frontendhez kapcsolódó TSX backend, egy **WPF** alkalmazásból az adminisztrátorok számára, valamint egy **MySQL** adatbázisból, amely az adatok tárolását biztosítja.

### Projektcsapat

* **Kun Szabolcs**: Backend fejlesztésért felelős. Feladata a WPF alapú adminisztrációs panel megvalósítása és az adatbázissal való kapcsolódás kialakítása, illetve a web applikáció backend szerverének felépítése.
* **Fazekas Tibor**: Frontend és adatbázis fejlesztésért felelős. Feladata a React alapú felhasználói felület fejlesztése, designolása és a MySQL adatbázis megtervezése és karbantartása.
* **Munkamegosztás**: 50%--50%

### Rendszerkomponensek

* **Frontend**: A felhasználói felület React keretrendszerben készült, amely biztosítja a szállodafoglalási lehetőségeket és az adatok kezelését a felhasználók részére, illetve telefonos felhasználóknak is kényelmessé teszi az oldal böngészését.
* **Backend és admin panel**: A WPF technológiával készült alkalmazás az adminisztrátorok számára, mely közvetlenül az adatbázishoz csatlakozik és lehetővé teszi a foglalások, szállodai szobák és felhasználói adatok kezelését.
* **Adatbázis**: MySQL alapú adatbázis, amely a rendszerhez szükséges adatokat tárolja, beleértve a felhasználók, foglalások, szobák és szállodák adatait.

### Rendszer céljai

* Egy felhasználóbarát, online szállodafoglalási rendszer létrehozása.
* Egy átlátható adminisztrációs felület megvalósítása, amely lehetővé teszi a foglalások és szobák kezelését.
* Az adatok biztonságos tárolása és kezelése egy jól struktúrált MySQL adatbázis segítségével.

### Fejlesztési környezet

* **Frontend**: React, TypeScript, HTML, SCSS, JavaScript
* **Backend**: C#, WPF
* **Adatbázis**: MySQL
* **Verziókezelés**: Git

## Backend

### ****Bun:****

* **Keretrendszer:** Express.js
* **Jellemzők:** Egy minimalista és rugalmas webalkalmazás keretrendszer Bun-hoz, ami lehetővé teszi gyors és hatékony webszerverek készítését. Különféle funkciókat és middleware-eket kínál, mint például útvonalak kezelése, HTTP kérés- és válaszobjektumok feldolgozása, sablonmotorok integrációja és sok más.

### **Jest:**

* A végpontok teszteléséhez egy könyvtár.
* Elindítása: gyökérmappát kiválasztva: **bun test**

### Felhasznált külső modulok

Függőségek "dependencies":

* **@tanstack/eslint-plugin-query@5.67.2:** Egy ESLint plugin, amely segít a TanStack Query (React Query) használata során felmerülő hibák elkerülésében.
* **@tanstack/react-query@5.67.3:** Egy hatékony adatkezelési könyvtár React alkalmazásokhoz, amely leegyszerűsíti a szerverről származó adatok gyors és hatékony kezelését.
* **@tanstack/react-router@1.114.17:** Egy modern és rugalmas útválasztási könyvtár React alkalmazásokhoz, amely lehetővé teszi dinamikus útvonalak kezelését.
* **@tanstack/router-devtools@1.114.17:** Fejlesztői eszköz a TanStack Router számára, amely segít az útvonalak hibakeresésében és optimalizálásában.
* **@tanstack/router-plugin@1.114.17:** Egy plugin a TanStack Router számára, amely további funkciókat biztosít az útvonalak kezeléséhez.
* **axios@1.8.3:** Egy HTTP kliens könyvtár, amely egyszerűvé teszi a szerverrel való kommunikációt aszinkron kérések segítségével.
* **globals@15.15.0:** Egy csomag, amely globális változók típusdefinícióit tartalmazza a JavaScript és TypeScript projektekhez.
* **jotai@2.12.2:** Egy minimális és hatékony állapotkezelő könyvtár React alkalmazásokhoz.
* **typescript@5.8.2:** A TypeScript fordító, amely típusbiztos JavaScript kód írását teszi lehetővé.
* **zod@3.24.2:** Egy sémavalidációs könyvtár, amely segít az adatok validálásában és típusbiztos kezelésében.

### Felépítés

Könyvtárstruktúra:

* **Backend:**
  + **controllers:** A backend logikáért felelős fájlok, amelyek kezelik a kéréseket és válaszokat.
  + **models:** Az adatbázis modellek és séma definíciók helye.
  + **routes:** Az API útvonalak definíciói és kezelése.
  + **types:** TypeScript típusdefiníciók a backendhez.
  + **tests:** A teszteléshez szükséges.
  + **utils:** Segédfüggvények és segédmodulok, amelyek a backend működését támogatják.
* **Frontend:**
  + **public:** Statikus fájlok, mint például képek és vektorok, amelyek közvetlenül elérhetők a böngészőben.
    - **images:** Képek, amelyek az alkalmazásban használatosak.
      * **full:** Teljes méretű képek.
      * **thumb:** Kicsinyített (thumbnail) képek.
    - **vectors:** Vektoros grafikák és ikonok.
  + **src:** A frontend forráskódjának fő könyvtára.
    - **assets:** Erőforrások, mint például adatok és képek, amelyek a komponensekben használatosak.
      * **data:** Statikus adatfájlok (pl. JSON).
      * **images:** Képek, amelyek a komponensekben használatosak.
    - **components:** A React komponensek, amelyek a felhasználói felület építőelemei.
      * **header:** A fejléc komponens és a hozzá tartozó stílusok.
      * **styles:** Globális és moduláris stílusfájlok.
      * **svg:** SVG ikonok és grafikák.
    - **hooks:** Egyéni React hookok, amelyek újrafelhasználható logikákat tartalmaznak.
    - **routes:** Az alkalmazás útvonalai és az útvonalakhoz tartozó komponensek.
      * **booking:** A foglalással kapcsolatos útvonalak és komponensek.
      * **hotels:** A szállásokkal kapcsolatos útvonalak és komponensek.
      * **styles:** Az útvonalakhoz tartozó stílusfájlok.

Middleware:

Middleware-ek az Express.js keretrendszerben segítenek az HTTP kérések feldolgozásában és a különböző feladatok elvégzésében a kérés és a válasz között. A middleware-ek feldolgozzák a kéréseket a beérkezési sorrendben, és módosíthatják azokat vagy adhatnak hozzájuk információt.

API végpontok:

* **Nyilvános végpontok:**
* GET /hotels : Az adatbázisban lévő hoteleket listázza ki.
* GET /hotels/id/{id} : Azonosító szerint lekéri a hotelt.
* GET /bookings : Az adatbázisban lévő foglalásokat listázza ki.
* GET /bookings/{id} : Azonosító szerint lekéri a foglalást.
* GET /images : Az adatbázisban lévő képeket listázza ki.
* GET /amenities : Az adatbázisban lévő szolgáltatásokat listázza ki.
* POST /register : Felhasználó regisztráció.
* POST /login : Felhasználó bejelentkezés.

• **Email végpontok (/email alapútvonallal):**

o POST /email/send-booking-email : Foglalás email küldése.

o POST /email/send-registration-email : Regisztrációs email küldése.

o POST /email/send-contact-email : Kapcsolati email küldése.

o POST /email/send-change-email : Jelszó változásról szóló email küldése.

• **Felhasználók (/users alapútvonallal):**

o GET /users : Az adatbázisban lévő felhasználókat listázza ki

o GET /users/:userId : Azonosító szerint lekéri a felhasználót

o GET /users/:id : Alternatív útvonal azonosító szerinti felhasználó lekéréshez

o PATCH /users/:userId : Felhasználó adatainak módosítása

o DELETE /users/:userId : Felhasználó törlése

o POST /users/:userId/change-password : Jelszó megváltoztatása

* **Bejelentkezéshez kötött végpontok (érvényes token szükséges):**
* POST /bookings: Hozzáad az adatbázishoz egy foglalást, userid-hoz kötve.
* PUT /rate/{id}: Rating hozzáadása, 0-ról 1-5-ig értékelés.

## Frontend

React.js:

* **Leírás:** A React.js egy nyílt forráskódú JavaScript könyvtár, amelyet a Meta fejlesztett ki. A React célja, hogy segítsen a fejlesztőknek hatékony és dinamikus felhasználói felületeket kialakítani.
* **Fő jellemzők:**
* Komponens-alapú fejlesztési módszer: A felhasználói felület komponensekre van felosztva, ami lehetővé teszi az egyszerű és újrafelhasználható kód írását.
* Egyirányú adatáramlás: Adatai egyszerűen és hatékonyan áramolhatnak a komponensek között.

#### ****SCSS:****

* **Leírás:** Az SCSS (Sassy CSS) egy CSS preprocesszor, amely lehetővé teszi a hatékonyabb és modularizált stílusok írását. A CSS-hez képest több funkcionalitást kínál, mint például változók, beágyazás, mixinek és függvények.
* **Fő jellemzők:**
  + **Modularizálás:** Lehetővé teszi a stílusok szétválasztását és újrafelhasználását különböző fájlokban.
  + **Kompatibilitás:** Teljesen kompatibilis a hagyományos CSS-sel, így könnyen integrálható meglévő projektekbe.

#### ****DaisyUI:****

* **Leírás:** A DaisyUI egy Tailwind CSS alapú komponens könyvtár, amely előre definiált, testreszabható UI komponenseket kínál. Célja, hogy gyors és egyszerű legyen a felhasználói felületek kialakítása.
* **Fő jellemzők:**
  + **Tailwind CSS integráció:** A DaisyUI teljes mértékben integrálható a Tailwind CSS-sel, így a Tailwind utility osztályait is használhatja.
  + **Előre definiált komponensek:** Gombok, kártyák, navigációs sávok és más UI elemek gyors használatát teszi lehetővé.
  + **Testreszabhatóság:** A komponensek könnyen testreszabhatók a projekt igényei szerint.

### ****Felhasznált külső modulok****

Függőségek ”dependencies”:

* **@tanstack/react-query** - Adatlekérdezés és állapotkezelés React alkalmazásokban
* **@tanstack/react-router** - Modern, típusbiztos útválasztó Reacthez
* **react@19.0.0 / react-dom@19.0.0** - Legújabb React verzió (béta)
* **vite@5.4.14** - Gyors, modern build eszköz
* **tailwindcss@4.0.13** - Utility-first CSS keretrendszer
* **daisyui@5.0.3** - Tailwind komponenskönyvtár
* **zod@3.24.2** - Séma validáció típusbiztosan
* **react-hook-form@7.54.2** - Hatékony űrlapkezelés
* **jotai@2.12.2** - Egyszerű állapotkezelés Reactban
* **axios@1.8.3** - HTTP kliens API hívásokhoz
* **typescript@5.8.2** - Statikus típusosság JavaScripthez
* **prettier@3.5.3** - Kódformázó eszköz

Komponensek leírása:

1. **Home Page**
   * **Routing:** /
   * **Leírás:** Főoldal
2. **Hotel listázás**
   * **Routing:** /hotels
   * **Leírás:** Az elérhető hoteleket kártyaszerűen listázza, komplex szűréssel és kereséssel, a felhasználó ki tudja választani a neki tetsző hotelt.
3. **About**
   * **Routing:** /about
   * **Leírás:** Az "About" oldal információkat nyújt a cégről, az alkalmazás céljáról, valamint a fejlesztői csapatról.
4. **Booking**
   * **Routing:** /booking/id
   * **Leírás:** Ez az oldal lehetővé teszi a felhasználók számára, hogy foglalást hozzanak létre egy adott szállásra. A felhasználók kiválaszthatják a dátumokat és megadhatják a személyes adataikat. A foglalás megerősítése után a rendszer elküldi a foglalási részleteket e-mailben.
5. **Booking history**
   * **Routing:** /bookinghistory
   * **Leírás:** A felhasználók itt tekinthetik meg az összes korábbi foglalásukat. Az oldal listázza a foglalások dátumát, a szállás nevét és az árát. A felhasználók megtekinthetik a részletes információkat.
6. **Kedvencek**
   * **Routing:** /favorites
   * **Leírás:** A felhasználók itt tárolhatják és kezelhetik a kedvenc szállásaikat. Az oldalon megjelennek a felhasználó által mentett szállások, amelyeket később gyorsan megtekinthet vagy foglalhat.
7. **Bejelentkezés/Regisztráció**
   * **Routing:** /login, /register
   * **Leírás:**
     + **Login:**  
       A felhasználók itt jelentkezhetnek be a rendszerbe a regisztrált e-mail címük és jelszavuk segítségével. A bejelentkezés után hozzáférhetnek a személyes fiókhoz, a foglalási előzményekhez és a kedvencekhez.
     + **Regisztráció:**  
       Az új felhasználók itt hozhatják létre fiókjukat. A regisztráció során meg kell adniuk az alapvető adatokat, mint például nevük, e-mail címük és jelszavuk. A regisztráció után a rendszer automatikusan bejelentkezteti a felhasználót.
8. **Kapcsolat**
   * **Routing:** /contact
   * **Leírás:** A "Contact" oldalon a felhasználók kapcsolatba léphetnek az ügyfélszolgálattal vagy a támogatási csapattal. Az oldalon található egy kapcsolati űrlap, ahol a felhasználók megadhatják nevüket, e-mail címüket, üzenetüket, valamint kiválaszthatják a kapcsolati okot (pl. kérdés, probléma, visszajelzés). Az oldalon továbbá megjelenik a cég elérhetőségei, mint például telefonszám, e-mail cím és fizikai cím.
9. **Egy hotel információi**
   * **Routing:** /hotels/{id}
   * **Leírás:** A kiválasztott hotel dedikált oldala, ahol le lehet foglalni, vagy kedvencekbe rakni. A hotelről bővebb információt ad.
10. **Felhasználói Profil**

* **Routing:** /userprofile
* **Leírás:** A bejelentkezett felhasználó meg tudja tekinteni a profilját, más felhasználóknak nem lehet megtekinteni egymásét. Meg lehet tekinteni a felhasználó alapvető adatait, illetve itt lehet elérni a foglalási előzményeket és a jelszóváltoztatást.

1. **Privacy policy**

* **Routing:** /privacy-policy
* **Leírás:** Az oldalhoz kapcsolódó adatvédelmi nyilatkozat.

## ****Asztali Alkalmazás/Admin Panel****

**HotelAdminPanel** egy **C#** nyelven készült **WPF (Windows Presentation Foundation)** asztali alkalmazás, amely egy hotel adminisztrációs rendszer kezelőfelületét biztosítja.

Fő jellemzők

* **Két felhasználói szerepkör:**
  + **Admin** – Teljes hozzáférés a rendszerhez
  + **Staff** – Korlátozott jogosultságok
* **Külön XAML és kód fájlok** az egyes felületekhez:
  + **AdminPage.xaml** & **AdminPage.xaml.cs**
  + **StaffPage.xaml** & **StaffPage.xaml.cs**

Modulok

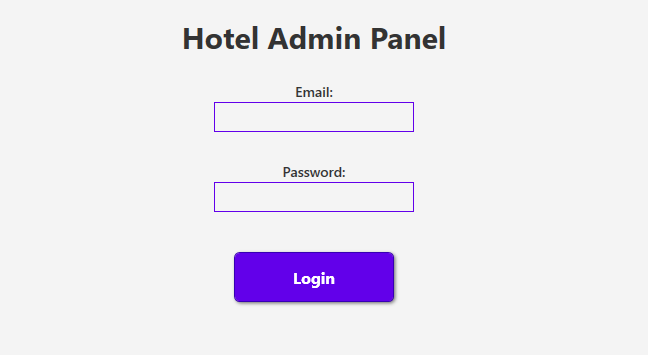
* **HotelManagement** – Szállodai egységek kezelése, Adminoknak minden jog, Staffoknak csak az olvasás funkció elérhető, nem tudnak adatot felvenni/törölni/változtatni.
* **UserManagement** – Felhasználók adminisztrációja, Adminoknak minden jog, Staffoknak csak az olvasás funkció elérhető, nem tudnak adatot felvenni/törölni/változtatni.
* **DataView** – Adatok táblázatos megjelenítése.
* **AdminPage** – Adminok felülete, ahol ki tudják választani a nézetüket.
* **StaffPage** – Staffok felülete, ahol ki tudják választani a nézetüket.
* **MainWindow** – Bejelentkezési oldal.

Technikai részletek

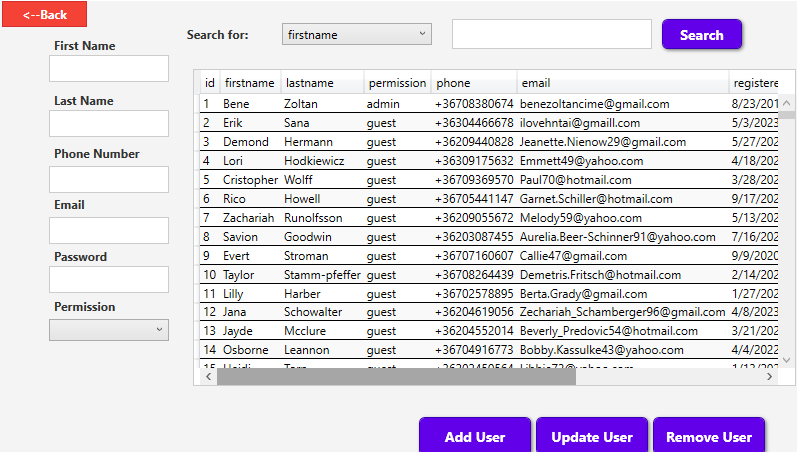
* **Adatbázis-kapcsolat:** **MySQL** extension
* **Fő ablak logikája:** **MainWindow.xaml.cs** (belépési pont)
* **Jogosultságkezelés:** Hierarchikus felépítés admin/staff különbségekkel

Az alkalmazás strukturált és hatékony kezelési lehetőséget biztosít a hotel üzemeltetéséhez, különböző jogosultsági szintek alapján.

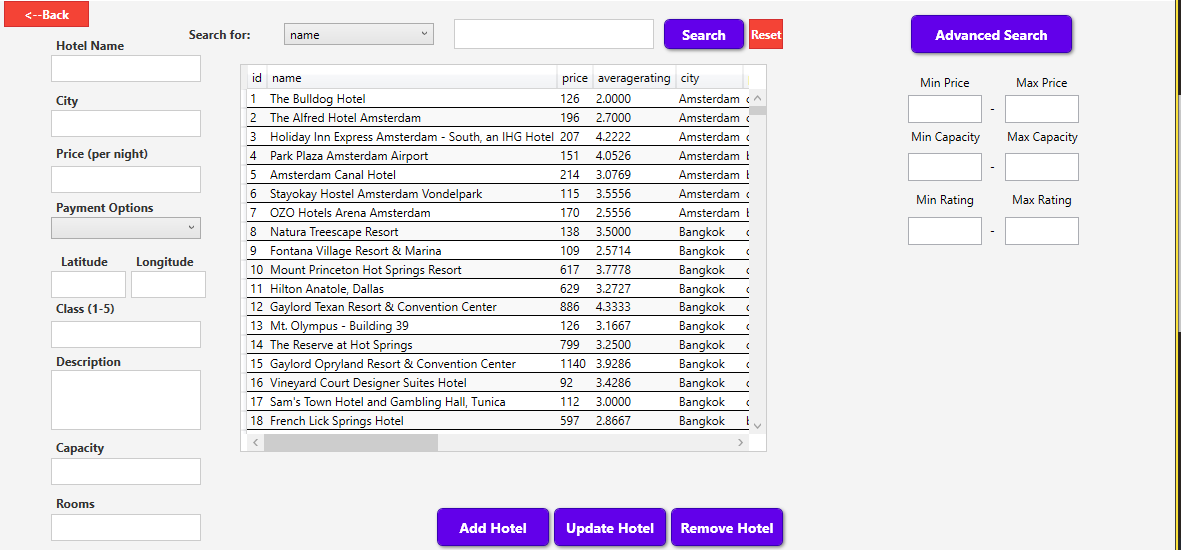
## Képek



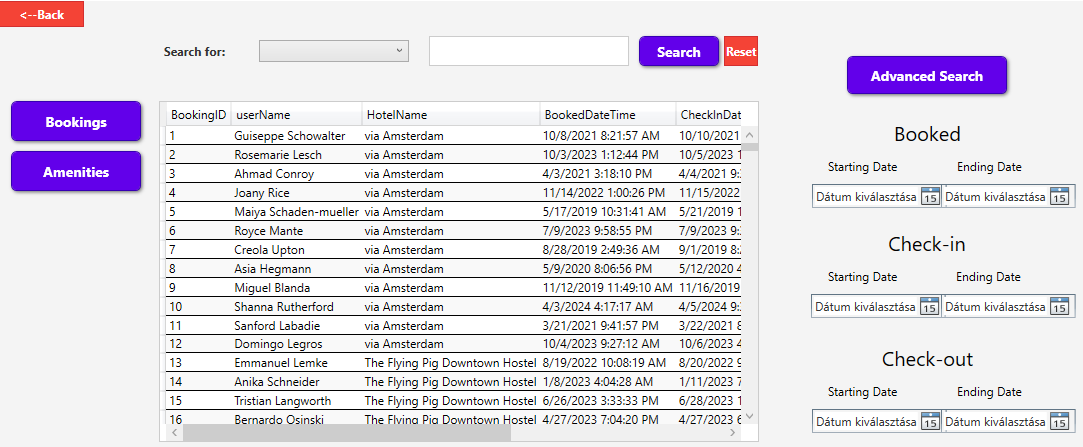
*Admin panel belépés*

**

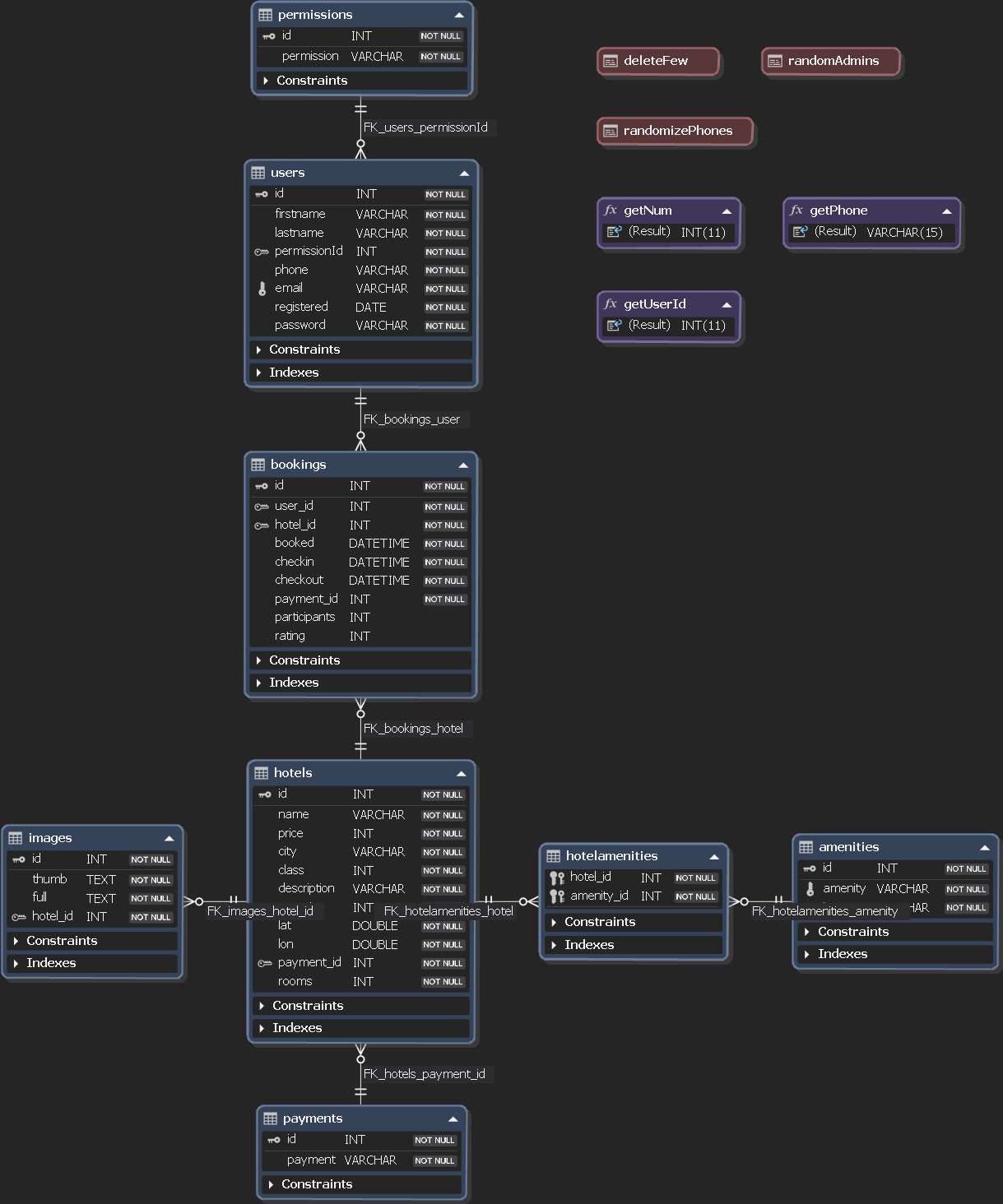
*Felhasználó Management*

**

*Hotel Management*

**

*Többi adat nézete*



*Adatbázis diagram*