Budapesti Műszaki Szakképzési Centrum

Verebély László Technikum

5-0613-12-03 Szoftverfejlesztő és-tesztelő szakképesítés

RushWave Retreat  
Szálloda foglaló weboldal

Készítette: Kovács Mihály Dániel, Vajda Dániel, Bárczi Bence

Konzulenstanárok: Somogyi Erika, Horváth Attila

2025. április

nyilatkozat a szakdolgozat eredetiségéről

Alulírott Kovács Mihály Dániel, Vajda Dániel és Bárczi Bence a BMSZC Verebély László Technikum 5-0613-12-03 Szoftverfejlesztő és-tesztelő képzésében részt vevő tanulók büntetőjogi felelősség tudatában nyilatkoznak és aláírásukkal igazolják, hogy a RushWave Retreat című vizsgaremek saját, önálló munkánk, és abban betartottuk az iskola által előírt, a vizsgaremek készítésére vonatkozó szabályokat.

Tudomásul veszem, hogy a szakdolgozatban plágiumnak számít:

* Szó szerinti idézet közlése idézőjel és hivatkozás nélkül,
* tartalmi idézet hivatkozás megjelölése nélkül
* más publikált gondolatainak saját gondolatként való feltűntetése.

E nyilatkozat aláírásával tudomásul veszem továbbá, hogy plágium esetén szakdolgozatom visszautasításra kerül.

Budapest, 2025. április 16 \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Hallgató 1 aláírása

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Hallgató 2 aláírása

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Hallgató 3 aláírása

Tartalomjegyzék

[1. Bevezetés 4](#_Toc195608534)

[1.1. Követelményspecifikáció: 5](#_Toc195608535)

[2. Fejlesztői dokumentáció 6](#_Toc195608536)

[2.1. Alkalmazott fejlesztői eszközök 6](#_Toc195608537)

[2.2. Alkalmazott technológiák 6](#_Toc195608538)

[2.3. Futtatási környezet 6](#_Toc195608539)

[2.4. Fejlesztői ütemterv 6](#_Toc195608540)

[2.5. Fejlesztés menete 7](#_Toc195608541)

[2.5.1. Piackutatás 7](#_Toc195608542)

[2.5.2. Adatbázis megtervezése 7](#_Toc195608543)

[2.5.3. Főbb algoritmusok 15](#_Toc195608544)

[2.5.3.1. Weboldal 15](#_Toc195608545)

[2.5.3.2. Asztali alkalmazás 22](#_Toc195608546)

[2.6. Tesztelés 27](#_Toc195608547)

[3. Felhasználói dokumentáció, felhasználói kézikönyv 29](#_Toc195608548)

[3.1. Telepítési útmutató. 29](#_Toc195608549)

[3.2. Vendég funkciók 30](#_Toc195608550)

[3.3. Felhasználói funkciók 37](#_Toc195608551)

[3.4. Alkalmazotti funkciók. 42](#_Toc195608552)

[3.5. Manageri funkciók 47](#_Toc195608553)

[3.6. Tulaj funkciók 49](#_Toc195608554)

[4. Továbbfejlesztési lehetőségek 50](#_Toc195608555)

[5. Összegzés 52](#_Toc195608556)

[6. Ábrajegyzék 53](#_Toc195608557)

1. Bevezetés

Vizsgaremekünk témájának kiválasztása egy személyes problémából adódott, hiszen egyik közeli ismerősünk megemlítette, hogy milyen nehéz manapság egy koherens szálloda foglaló weboldalt találni, ami nem komplikálja túl a foglalás menetét.

Manapság a hotellánc rengeteg mindent nyomnak a felhasználó arcába, ezzel néha átláthatatlanná teszik saját weboldalukat, ezért mi úgy döntöttünk, hogy készítünk egy letisztult és könnyen kezelhető weboldalt melynek segítségével az átlagember könnyedén tudja elintézni következő nyaralását.

Ezen felül szerettünk volna egy olyan kihívást kitűzni magunk elé, amely lehetőséget ad arra, hogy próbára tegyük saját tudásunkat és fejlesszük képességeinket. Úgy döntöttünk, hogy egy szálloda foglaló oldal erre egy tökéletes feladat, hiszen a koncepciója egyszerű, viszont számos extra kihívással bír.

Szeretnénk létrehozni egy olyan weboldalt, amely átlátható mindenki számára, miközben megőrzi a szálloda foglalás komplexitását, ezzel megnyitva a kaput különböző hotelláncoknak, hogy kérvényezzék saját szállodáikat az oldalra, valamint lehetőséget adni a felhasználóknak, hogy találjanak egy számukra kedvező szokásos nyaralóhelyet.

* 1. Követelményspecifikáció:

Célunk, hogy legyen olyan weboldalunk, amely funkcionalitás terén képes lenne megállni a helyét a modern piacon. További célunk volt, hogy a legjobb élményt nyújtsuk a felhasználónak. A weboldal elsőszámú célja hogy a felhasználó tudjon csinálni egy fiókot, ahol követni tudja saját foglalásait illetve, hogy képes legyen foglalni az oldalon jelen lévő szállodák valamelyikébe. Ezek mellett fontosnak tartottuk, hogy egy felhasználó képes legyen véleményt nyilvánítani egy értékelés formájában.

A weboldalt két féle módon lehet megtekinteni, mint vendég, és mint fiókkal rendelkező felhasználó. A vendég ugyanúgy megtudja tekinteni a különböző szállodákat illetve azoknak értékeléseit, viszont ő maga nem tud foglalni vagy értékelést írni, ellentétben egy regisztrált felhasználóval. Ezentúl, a weboldalt számos böngészőben meglehet tekinteni, illetve mobil és táblagép eszközökkel is kompatibilis.

A C sharp alkalmazás lényege, hogy a különböző szállodák alkalmazottjai tudják kezelni a szállodák különböző adatait, mint például a szolgáltatások elérhetőségét, a szobák árát vagy a foglalások különböző aspektusait. Az alkalmazottak kizárólag azokat a szállodákat tudják kezelni, ahol dolgoznak, ezzel elkerülve a lehetséges szabotálást

1. Fejlesztői dokumentáció
   1. Alkalmazott fejlesztői eszközök

* Kódszerkesztő: Visual Studio Code, Visual Studio
* Fejlesztői környezet: Laravel, .NET
  1. Alkalmazott technológiák
* Programozási nyelvek: PHP, JavaScript, C#
* Webes technológiák: HTML, CSS
* Adatbázis motor: MariaDB (MySQL)
  1. Futtatási környezet

A weboldal bármely modern böngészőben működőképes, akár mobileszközön is megtekinthető.

A C# alkalmazás asztali számítógépeken elérhető, vagy egyéb olyan eszközön, amely Windows 10 vagy afeletti operációs rendszerrel rendelkezik.

* 1. Fejlesztői ütemterv

|  |  |
| --- | --- |
| Határidő | Feladat |
| 2024.12.19 | Adatlap beadása, vizsgaremek témájának eldöntése |
| 2025.01.27 | Végleges sablon bemutatása |
| 2025.02.21 | Adatbázis terv bemutatása |
| 2025.02.24 | Működőképes Autentikáció |
| 2025.03.24 | Weboldal és asztali alkalmazás főbb funkcióinak kialakítása |
| 2025.04.10 | Stílus csiszolgatása, végső simítások |
| 2025.04.16 | Végső leadási határidő |

* 1. Fejlesztés menete
     1. Piackutatás

A projekt megtervezése előtt számos hasonló weboldalt kutattunk át, mind magyart, mind nemzetközit. Kutatómunkánk során gyűjtöttük ki hogy milyen fő funkcionalitások kellenek egy ilyen oldal működéséhez, illetve hogy mi ezeken hogyan tudnánk fejleszteni. Ennek a végeredménye az lett, hogy weboldalunk hasonlít a már piacon lévőkre, ezzel megadva a felhasználónak egy már meglévő komfortot, viszont, különbözik annyira, hogy ne vegyüljön el a már meglévő piac rengetegében

* + 1. Adatbázis megtervezése

Az adatbázisunk MySQL nyelven, MariaDB-vel működik.

Az adatbázis 12 táblával rendelkezik, viszont számos egyéb adattag kiszámítható ezekből az adatokból. Törekedtünk arra, hogy az adatbázis megfeleljen a harmadik normálformának, illetve célunk volt a redundancia (azaz a szükség nélküli adatismétlődés) elkerülése. A következő oldalakon a táblák felépítése található:

1. ’User’ tábla

Ez a tábla tárolja a regisztrált felhasználó minden adatát

* user\_id: A felhasználó egyedi azonosítója, egész szám, elsődleges kulcs
* username: A felhasználó egyedi felhasználó neve, maximum 32 karakter hosszú.
* lastName: A felhasználó vezetékneve, maximum 50 karakter hosszú.
* firstName: A felhasználó keresztneve, és amennyiben van akkor harmadik neve, maximum 50 karakter hosszú.
* birthDate: A felhasználó születési dátuma.
* phonenumber: A felhasználó telefonszáma.
* email: A felhasználó saját email címe.
* password: A felhasználó jelszavának titkosított változata.
* created\_at: A pontos időpontja, hogy egy felhasználó mikor regisztrált az oldalra.
* updated\_at: A pontos időpontja, hogy egy felhasználó mikor módosította valamelyik adatát. (például email vagy telefonszám)
* active: logikai változó, mutatja, hogy egy felhasználó fiókja törlésre került-e már.
* profilePic: A felhasználó profilképének azonosítója, egész szám.

1. ’loyaltyrank’ tábla

Ez a tábla tárolja a felhasználó hűségprogram adatait.

* rank\_id: A tábla egyedi azonosítója, elsődleges kulcs.
* rank: a hűségszint elnevezése
* discount: tört szám, amely megadja, hogy az adott rang hány százalékos kedvezményt kap a vásárlásaira.
* minPoint: A minimális pontszám, ami szükséges, hogy a felhasználó erre a szintre lépjen
* perks: A hűségszint nyújtotta előnyök

1. ’loyalty’ tábla

Ez a tábla tárolja a felhasználó hűségprogram adatait.

* loyalty\_id: A tábla egyedi azonosítója, elsődleges kulcs.
* user\_id: Idegen kulcs a ’user’ táblához.
* rank\_id: Idegen kulcs a ’loyaltyrank’ táblához.
* points: A felhasználó hűségpontjai, egész szám.
* updated\_at: A hűségpontok frissülésének pontos dátuma.

1. ’city’ tábla

Ez a tábla tárolja el azokat a városokat, ahol az oldal szállodákat üzemeltet.

* city\_id: A város egyedi azonosítója, elsődleges kulcs
* cityName: A város neve, maximum 100 karakter hosszú
* country: Az ország, amelyben a város elhelyezkedik, maximum 100 karakter hosszú
* description: A város hosszú leírása, amelyet kiírunk az oldalra.
* description\_short: A város leírásának rövid változata.

1. ’hotel’ tábla

Ez a tábla tárolja el az oldalon szereplő szállodák adatait

* hotel\_id: A szálloda egyedi azonosítója, elsődleges kulcs
* city\_id: Idegen a kulcs a ’city’ táblához, mutatja, hogy melyik városban található a város.
* hotelName: A szálloda elnevezése, maximum 150 karakter hosszú
* address: A szálloda pontos címe, szöveges.
* phoneNumber: A szálloda recepciójának telefonszáma
* email: A szálloda saját email címe, szöveges.
* description: A szálloda leírása, amelyet betöltünk az oldalra, szöveges.

1. ’reviews’ tábla

Ez a tábla tárolja el a felhasználók által írt értékeléseket.

* review\_id: Az értékelés egyedi azonosítója, elsődleges kulcs.
* user\_id: Idegen kulcs a ’user’ táblához
* hotel\_id: Idegen kulcs a ’hotel’ táblához
* rating: Az értékelés csillagértéke, egész szám.
* reviewText: Az értékelés szövegi része, szöveges.
* created\_at: Az értékelés létrehozásának/módosításának pontos időpontja.
* active: Logikai, mutatja, hogy az értékelés törlésre került-e
* edited: Logikai, mutatja, hogy az értékelés szerkesztve lett-e

1. ’employee’ tábla

Ebben a táblában vannak eltárolva azok a felhasználók, akik valamely szállodában dolgoznak.

* hotel\_id: Idegen kulcs a ’hotel’ táblához
* user\_id: Idegen kulcs a ’user’ táblához
* userType: Mutatja, hogy a felhasználó milyen pozícióban van a szállodában, értékei „employee”, azaz alkalmazott, „manager” vagy pedig „owner” aki a szálloda tulajdonos.

1. ’servicecategory’ tábla

Ebben a táblában vannak eltárolva a különböző szolgáltatásai nevei.

* serviceCategory\_id: A szolgáltatás egyéni azonosítója, elsődleges kulcs
* serviceName: A szolgáltatás megnevezése.

1. ’service’ tábla

Ebben a táblában vannak eltárolva a szálloda szolgáltatásainak pontos adatai

* service\_id: A szolgáltatás egyéni azonosítója, elsődleges kulcs.
* hotel\_id: Idegen kulcs a ’hotel’ táblához
* category\_id: Idegen kulcs a servicecategory táblához
* price: Egész szám, a szolgáltatás igénybevételének ára
* available: Logikai, mutatja, hogy a szolgáltatás az aktuális pillanatban elérhető-e
* allYear: Logikai, mutatja, hogy a szolgáltatás egész évben elérhető-e
* startDate: Amennyiben nem egész évben elérhető, a szolgáltatás kezdeti dátuma
* endDate: Amennyiben nem egész évben elérhető, a szolgáltatás végső dátuma
* openTime: Amennyiben a szolgáltatás nem érhető el egész nap, a nyitó ideje.
* closeTime: Amennyiben a szolgáltatás nem érhető el egész nap, a záró ideje.

1. ’room’ tábla

Ebben a táblában vannak eltárolva a szállodák különböző szobáinak az adatai

* room\_id: A szoba egyéni azonosítója, elsődleges kulcs
* hotel\_id: Idegen kulcs a ’hotel’ táblára.
* roomNumber: A szoba megnevezése, maximum 25 karakter hosszú
* floor: Az emelet, amelyen a szoba elhelyezkedik, egész szám
* capacity: A szoba férőhelye, egész szám.
* pricepernight: A szoba éjszakánkénti foglasának az ára
* available: logikai, mutatja, hogy a jelenlegi pillanatban elérhető-e a szoba

1. ’booking’ tábla

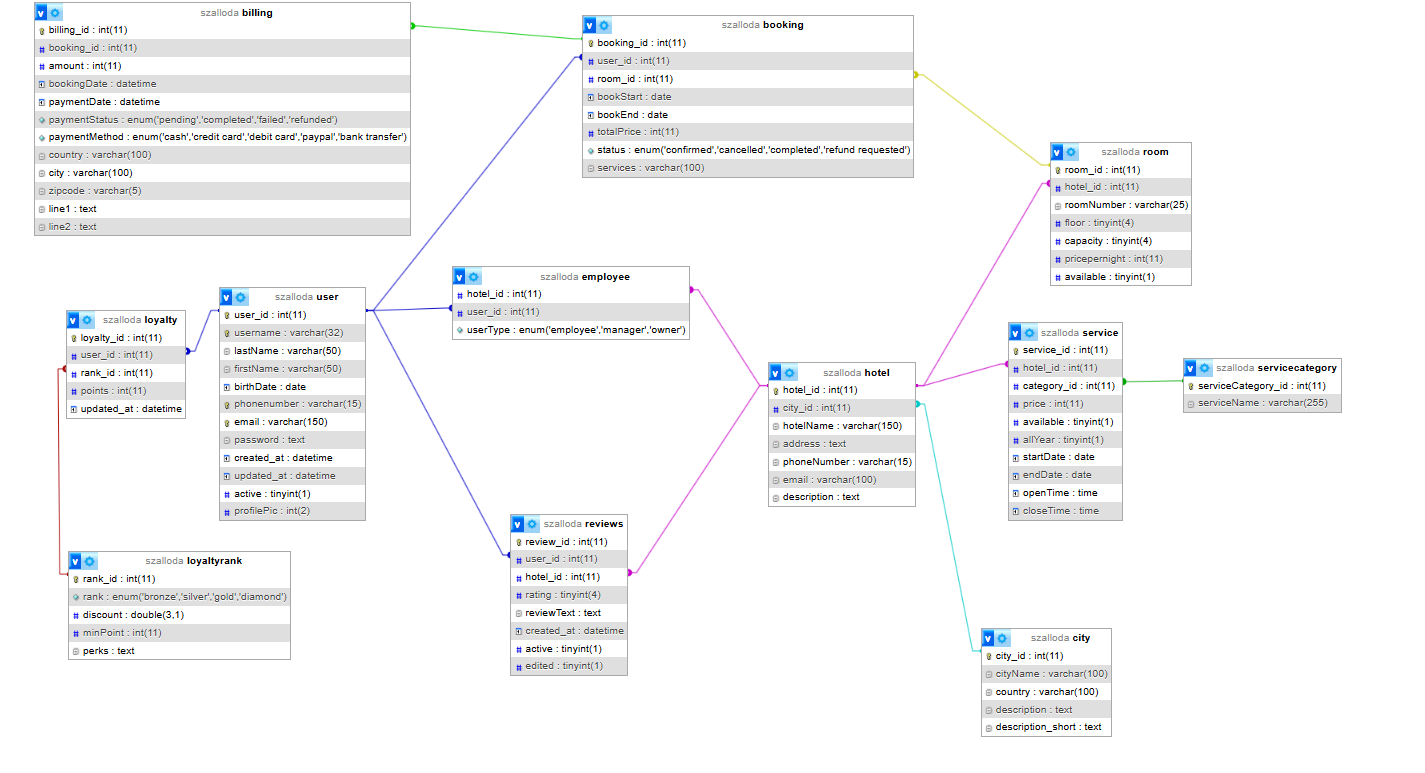
Ez a tábla tartalmazza az oldalon történt foglalásokat

* booking\_id: A foglalás egyéni azonosítója, elsődleges kulcs.
* user\_id: Idegen kulcs a ’user’ táblához
* hotel\_id: Idegen kulcs a ’hotel’ táblához
* bookStart: A foglalás kezdetét jelölő dátum
* bookEnd: A foglalás végét jelölő dátum
* totalPrice: A foglalásért fizetendő összeg forintban, egész szám
* status: A foglalás jelenlegi állapotát jelzi, értékei ’confirmed’ azaz megerősített, ’cancelled’ azaz lemondott, ’completed’ azaz befejezett és ’refund requested’ azaz visszatérést igényelt.
* services: A foglalás mellé nyújtott szolgáltatások listája.

1. ’billing’ tábla

Ez a tábla tartalmazza a foglalás számlázásával kapcsolatos adatait.

* billing\_id: A számla egyéni azonosítója, elsődleges kulcs.
* booking\_id: idegen kulcs a ’booking’ táblához.
* amount: A fizetett összeg forintban, egész szám.
* bookingDate: A foglalás pontos időpontja.
* paymentDate. A fizetés pontos időpontja.
* paymentStatus: A fizetés jelenlegi állapotát jelzi, értékei ’pending’ azaz függőben lévő, ’completed’, azaz befejezett, ’failed’ azaz sikertelen volt a fizetés és ’refunded’, azaz visszatérített
* paymentMethod: A fizetés módszerét jelzi, értékei: ’cash’ azaz készpénz, ’credit card’ azaz, hitelkártya, ’debit card’ azaz betéti kártya, ’paypal’ illetve ’bank transfer’, azaz banki átutalás.
* country: A számlázott felhasználó országa.
* city: A számlázott felhasználó városa.
* zipcode: A város irányítószáma.
* line1: A lakcím első sora, az utca és házszámot tartalmazza
* line2: A lakcím második sora, a cím egyéb adatait tartalmazza, mint például az emeletet és az ajtószámot

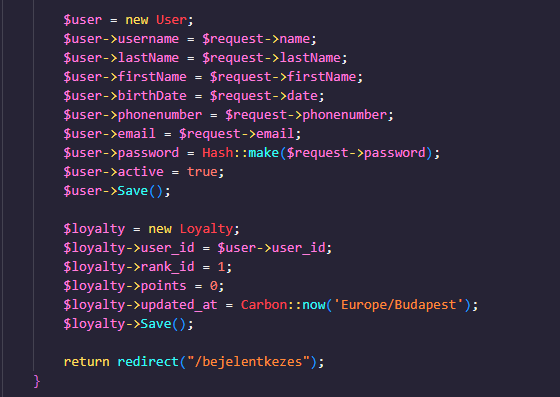


1. ábra: EER diagram a "szalloda" adatbázisról

* + 1. Főbb algoritmusok
       1. Weboldal

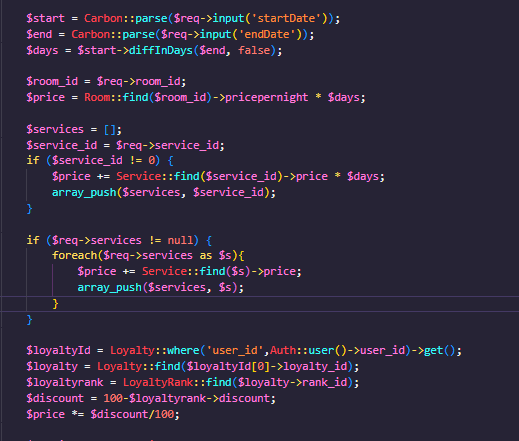
A weboldal backendje Laravelt használ, hiszen a csapat tagjainak ez volt a legkényelmesebb fejlesztői környezet, továbbá a beépített funkciók segítségével képesek voltunk többet megvalósítani, mint amit eredetileg terveztük. Erre az egyik legjobb példa a regisztráció, hiszen itt a Laravel segítségével könnyedén tudtuk validálni hogy a felhasználó megfelelően töltötte-e ki az adatait, illetve egyértelműen tudtuk jelezni, hogy mi a probléma a bevitt adatokkal.

Miután meggyőződtünk róla hogy a felhasználó helyesen vitte be az adatait, lefut a függvény további része amely létrehozza a felhasználó fiókját, illetve a hozzátartozó hűségprogram adatokat. (Lásd: 2. ábra) A felhasználó jelszava titkosítva van eltárolva, így amennyiben illetéktelen hozzáférés történik az adatbázishoz, nem fognak tudni hozzáférni a felhasználók jelszavához.



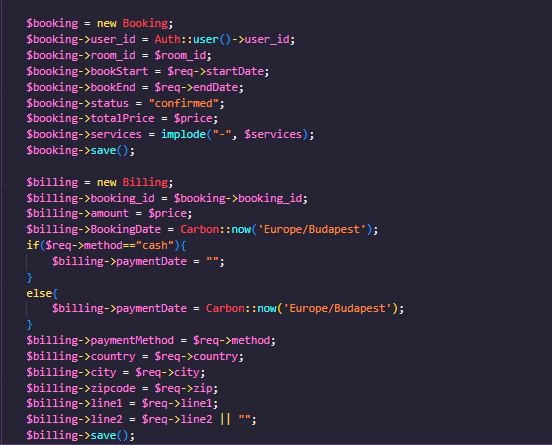
2. ábra: A regisztráció függvény adatbázisba illesztő része

Az oldal leglényegesebb funkciója a szobafoglalás, így természetesen ez a függvény az egyik leghosszabb, így ezt szeretném több részre felbontani. Elsősorban a függvény ellenőrzi, hogy a felhasználó bevitte-e a megfelelő adatokat. Majd ezután egyesével kiszámolja a költségeket.(Lásd: 3. ábra) Először kiszámolja, hogy mennyi a szoba ára az ott töltött éjszakákra, majd ezután hozzáadja a kiválasztott szolgáltatások árát. A végső árra ezután alkalmazza a felhasználó hűségprogramjából származó akciót is.



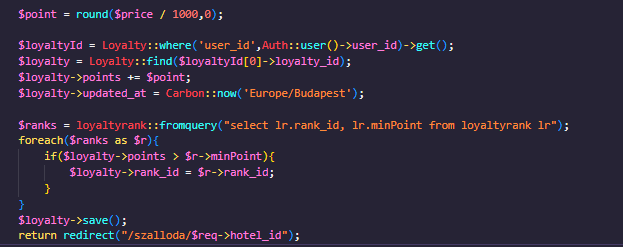
3. ábra: A foglalás függvény árat számoló része

Ezek után a függvény feltölti az adatbázisba a felhasználó által bevitt adatokat mind a foglalás, mind a számlázás táblába. (Lásd: 4. ábra)



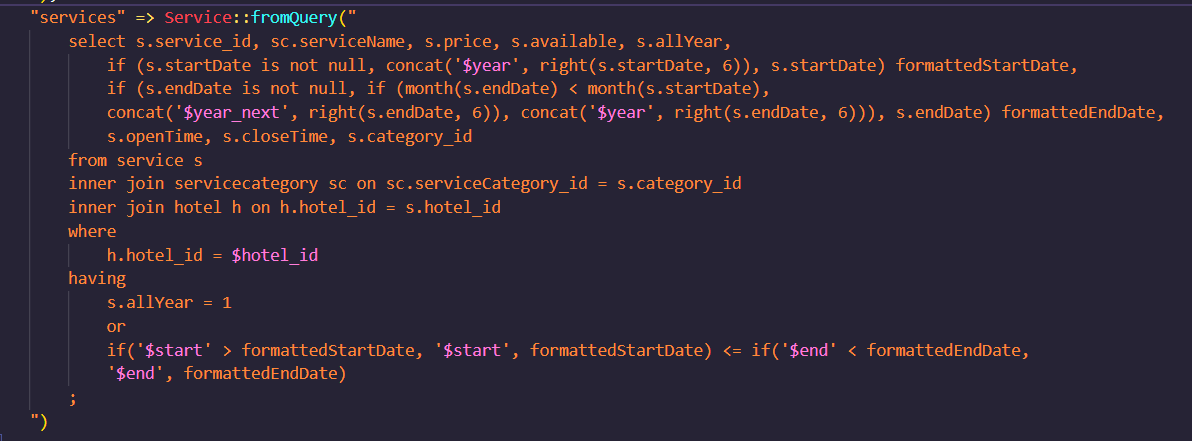
4. ábra: A foglalás függvény adatbázisba író része

Végezetül, a függvény frissíti a felhasználó hűségpontjait, illetve le ellenőrzi, hogy a felhasználó elért-e egy új hűségszintet, és amennyiben igen, annak megfelelően frissíti a felhasználó adatait. (Lásd: 5. ábra)



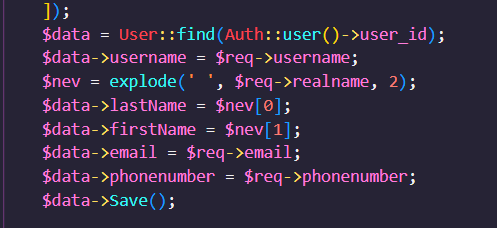
5. ábra: A foglalás függvény hűségpontszámító része

A foglaláshoz tartozik továbbá a dátum alapján betöltött adatok. Nem akartuk hogy a felhasználó képes legyen olyan szolgáltatást igénybe venni, ami nem elérhető a foglalása alatt. Ezt adatbázis lekéréssel adtuk meg, ami kizárólag azokat a szolgálatásokat adja vissza, amelyek a foglalás akár csak egy napján is elérhetőek (Lásd: 6. ábra). Hasonló lekéréssel kérjük ki a felhasználó által megadott dátumokon elérhető szobákat.

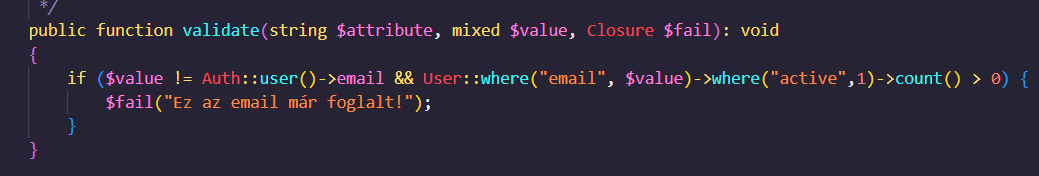


6. ábra: A szolgálatás kikéréséért felelős adatbázis lekérés

A felhasználó tudja az oldalon módosítani saját adatait, amennyiben azok változtak regisztráció óta. A felhasználó tudja módosítani a felhasználónevét, email címét, telefonszámát vagy akár a polgári nevét is. A függvény figyel az email cím helyességére, a felhasználónév foglaltságára, illetve arra hogy a felhasználó beírta-e a vezeték és keresztnevét is. (Lásd: 7. ábra) Továbbá saját validálási szabályok segítségével leellenőrizzük hogy jelenleg aktív felhasználó által használva van a megadott telefonszám vagy email. (Lásd: 8. ábra)

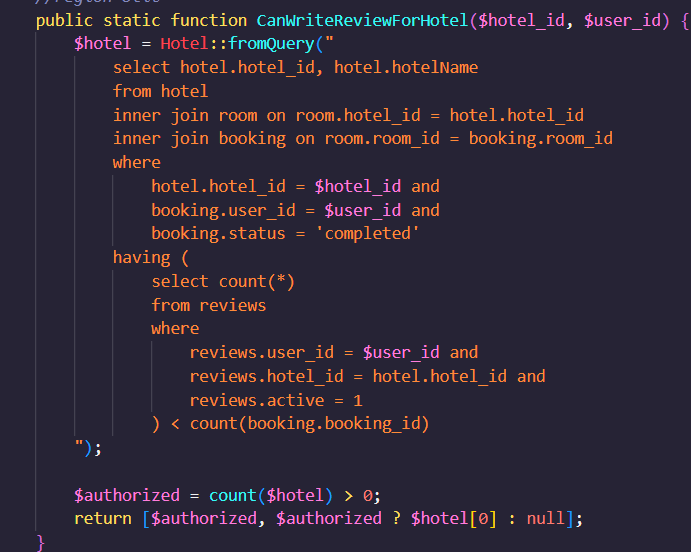


7. ábra: A felhasználói adatokat módosító függvény



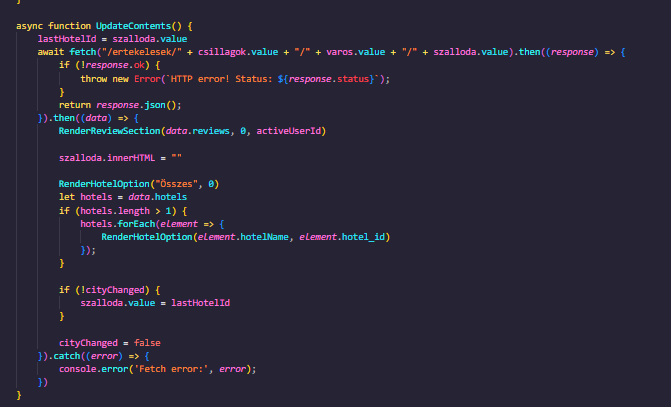
8. ábra: Az email egyéniségét ellenőrző függvény

A felhasználó nem tud bármennyi értékelést írni, kizárólag annyit amennyi befejezett foglalása van. Ezt egy adatbázis lekérdezéssel oldjuk meg, ahol megszámoljuk a felhasználó kész foglalásait, és visszaadjuk azoknak a hoteleknek a nevét amelyekhez jogosult értékelést írni. (Lásd: 9. ábra)

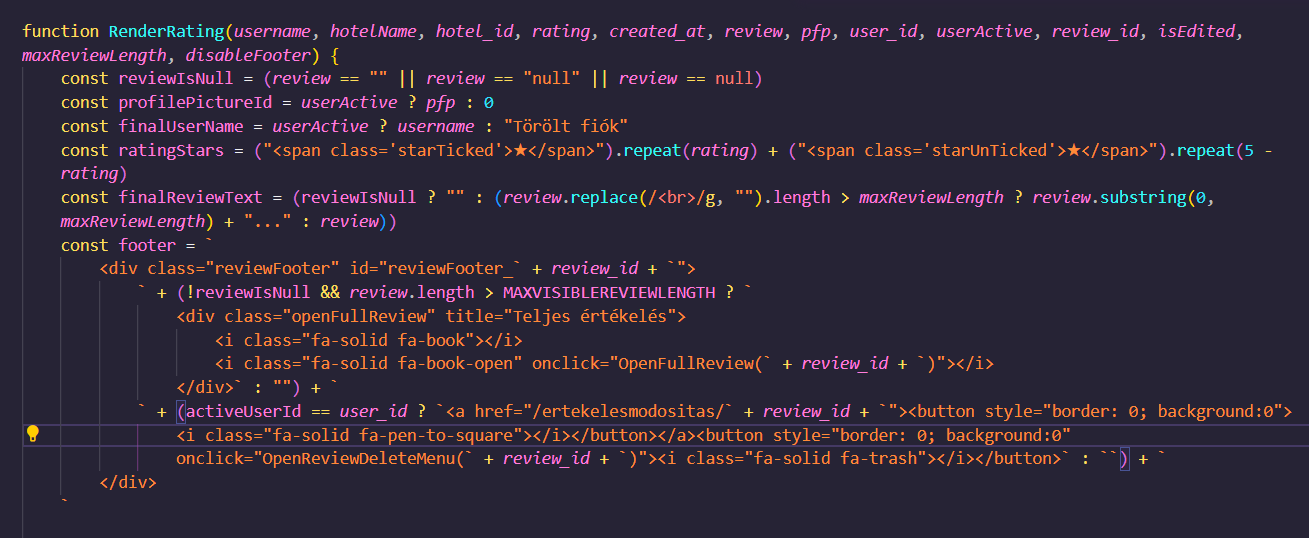


9. ábra: Az értékelés írásra engedélyt adó függvény

Sajnos, mivel a Laravel nem képes dinamikusan frissíteni az oldalt bizonyos változások bekövetkezésekor, ezért a JavaScript is fontos szerepet játszik a weboldal működésében. A JavaScript függvényeink kényelmet biztosítanak a felhasználónak, hiszen így nem kell folyamatosan újra frissíteni az oldalt, amikor valamilyen változás történik annak tartalmában. Erre a legjobb példa az értékelések szűrése. A függvény kikéri az összes értékelést amely megfelel a szűrés feltételeinek, (Lásd: 10. ábra) majd ezeket továbbítja egy másik függvénynek amely megjeleníti azokat (Lásd: 11. ábra)

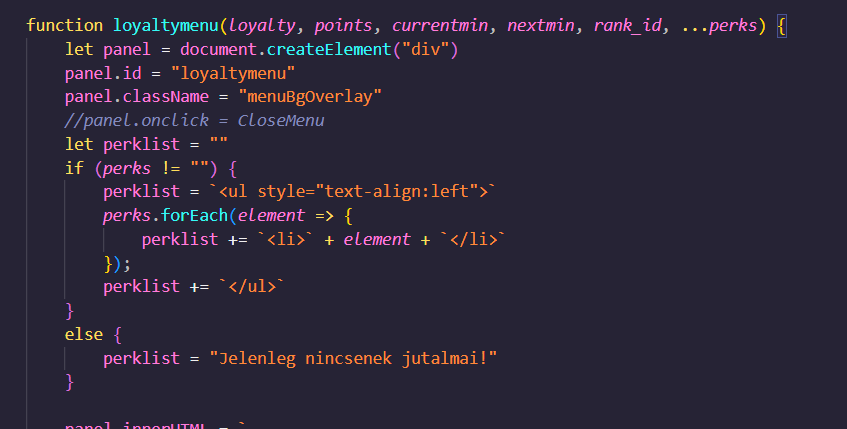


10. ábra: Az értékelés szűrést végző függvény



11. ábra: Az értékelés szekciók létrehozásáért felelős függvény

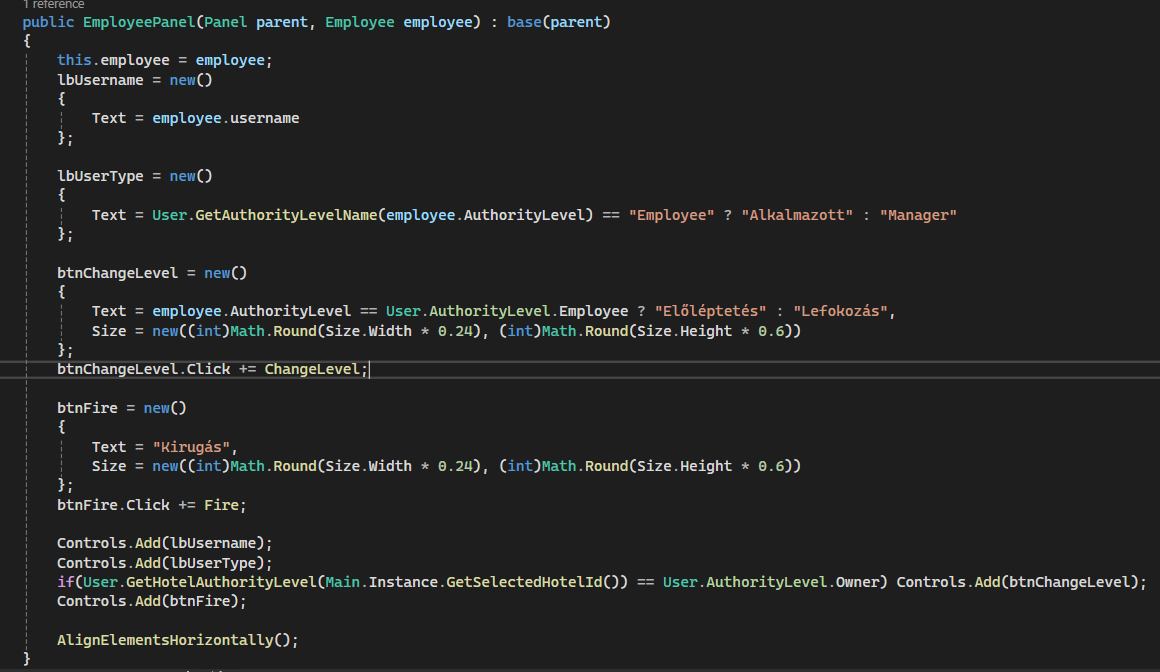
JavaScriptet viszont nem csak oldalfrissítésre használjuk, hanem arra is, hogy megjelenítsünk a felhasználónak új menüpontokat kattintásra, mint például a hűségszint vagy a profilkép változtatás menüje (Lásd: 12. ábra). Ezek a függvények hozzák létre a felhasználó által látott menüt, minden működésével együtt



12. ábra: A hűségmenü ablakát létrehozó függvény

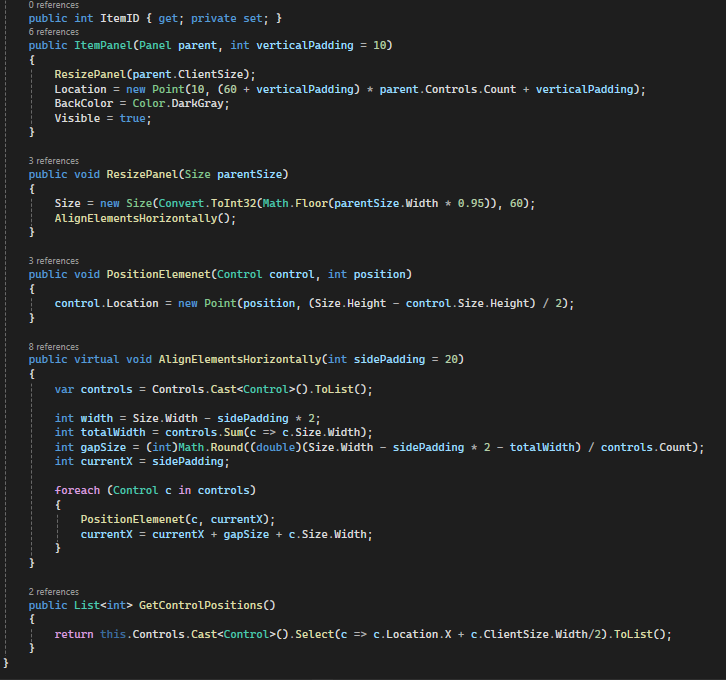
* + - 1. Asztali alkalmazás

Az asztali alkalmazás C# formsban készült, hiszen úgy gondoltuk, hogy ez tökéletesen megfelel annak, amit létre szerettünk volna hozni. Mivel nem tudtuk előre hogy mennyi adatot kell megjelenítenünk, arra a döntésre jutottunk, hogy dinamikusan fogunk paneleket létrehozni, amelyek tartalma változhat attól függően, hogy mit szeretne betölteni a felhasználó.(Lásd: 13. ábra) Minden opcióra más és más dolgokat kell betölteni, például a szolgáltatásoknál elég csak annak az elérhetőségét és egy módosítás gombot betölteni, amíg a foglalásoknál 5 különböző adatot kell.



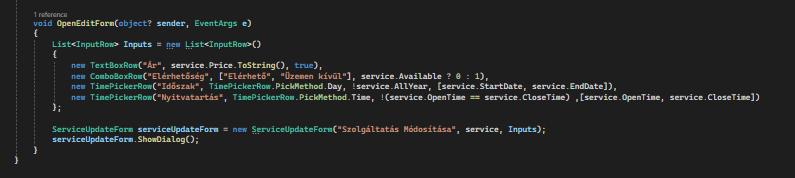
13. ábra: Példa egy panel létrehozására

Ezentúl az egyik legnehezebb része az volt, hogy megcsináljuk a paneleket illetve az egész ablakot reszponszívra, tehát méretük és helyzetük változzon, hogyha a felhasználó nagyobbítja vagy kicsinyíteni az ablakot. Ilyenkor megkell nyúlnia az alap panelnek amin minden rajta van, az elemenként betöltött paneleknek, illetve azoknak tartalmainak is úgy kell elrendeződnie hogy az illjen a panel méretére. (Lásd: 14. ábra)



14. ábra: A panelek reszponszivitásáért felelős függvény

Amennyiben a felhasználó által módosítani kívánt elemnek számos olyan eleme van, ami szerkeszthető, akkor megnyitunk egy új ablakot ennek szerkesztésére. Itt kikérjük azokat elemeket amelyeket megkell majd jelenítünk (Lásd: 15. ábra)



15. ábra: Az osztályok alapján történő form létrehozás

Meghívásra kerül számos alosztály, amelyek a beviteli mezők létrehozásáért felelnek. Itt bevitt feltételek alapján kerülnek létrehozásra az elemek (Lásd: 16. ábra), majd az adatbázisból kikért adatokkal feltölti azokat (Lásd: 17. ábra)

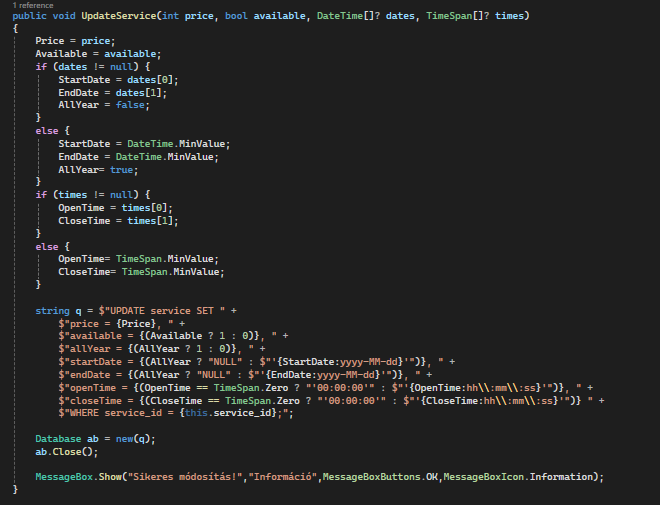


16. ábra: Első fele egy osztálynak, amely beviteli elemeket hoz létre



17. ábra: Második fele egy osztálynak amely beviteli elemeket hoz létre

Miután a felhasználó kiválasztja a módosítani kívánt elemet, lekell kezelnünk annak egyéni ablakát, majd frissíteni az adatbázis adatait annak megfelelően. Ilyenkor az adatbázisból kikeressük azt az elemet, amelyet épp módosítanak, majd minden adatát lecseréljük arra, amit a felhasználó beírt, figyelve az adatbázis különböző megkötéseire, mint például az idegen kulcsok, az adatok helyes megformázása, illetve a kötelező kitöltésű mezőkre. (Lásd: 18. ábra)



18. ábra: Példa az adatbázis frissítésére

* 1. Tesztelés

A weboldal fejlesztése alatt dinamikusan teszteltünk, ami azt takarja, hogy a kód közvetlen megírása után teszteltük a működését is. Az oldalt manuálisan volt tesztelve, azaz saját kézzel ellenőriztük az oldal megfelelő működését. Ilyenkor az összes módosított adatot egyszerre módosítjuk.

A legtöbb tesztelést a foglalási oldal igényelte, hiszen itt vannak a legtöbb adatbeviteli felület, illetve a legbonyolultabb függvények is. Eleinte a szobák elérhetőségi szűrési kizárólag azt vizsgálta, hogy a foglalás kezdeti és vég dátuma alatt foglalt-e a szoba, így amennyiben valaki úgy foglalt, hogy a foglalása közepén valaki más már kivette-e a szobát, az oldal nem jelzett volna hibát és engedte volna a kettős foglalást. Szerencsére a folytonos tesztelés miatt erre nagyon hamar rájöttünk

További nehézséget okozott az adatbázisba való helyes adat bevitel is, hiszen sokszor nem egyenesen azt visszük be az adatbázisba, amit a felhasználó begépel. Erre egy tökéletes példa a profil oldalon használt „polgári név” mező, hiszen itt a felhasználó csak beírja a teljes nevét, viszont az adatbázisban a vezeték és keresztnevek külön vannak tárolva. Sokáig nem találtunk arra megoldást, hogy hogyan tudnánk leellenőrizni azt, hogy a felhasználó legalább 2 nevet beleírt-e a mezőbe, hiszen amennyiben ez nem teljesül az adatbázis hibásan lenne feltöltve, viszont szerencsére sikerült megoldani ezt a problémát.

Sajnos az asztali alkalmazás fejlesztése se ment zökkenő mentesen. Mivel minden egyes elem saját dinamikusan létrehozott panelt kapott, ezért rengeteg munka volt azzal, hogy ezek működően, egymás alá generálódjanak, hiszen rengetegszer kilógtak a képből, egymásra generálódtak vagy a gombok nem működtek rendeltetésszerűen. Szerencsére minden hibát ezzel kapcsolatban sikerül kiköszörülni és minden kellően működik.

A folytonos tesztelésnek köszönhetően sikerült létrehoznunk egy olyan oldalt, ami javarészt hibamentes. Természetesen előfordulhat, hogy valamelyik apró zugban egy apró hiba fellelhető, viszont a csapat minden tőle telhetőt megtelt, hogy egy közel tökéletes oldalt hozzon össze.

1. Felhasználói dokumentáció, felhasználói kézikönyv

A weboldal két féle módban tekinthető meg. vendégként, ami körbetekintés céljából hasznos, hiszen a felhasználó megtudja nézni a kínálatunkat regisztráció előtt, illetve felhasználóként, akiknek a teljes oldal elérhetővé válik. Ezentúl az asztali alkalmazás szolgál az adminisztrációs felületre, ahol a szálloda alkalmazottai és managerei képesek módosítani a szállodák adatait, mint például a szolgálatásokat, szobákat és foglalásokat. A következő részek egymásra épülnek, így minden, amihez egy vendég hozzáfér, ahhoz a regisztrált felhasználók és az alkalmazottak is.

* 1. Telepítési útmutató.

Először is, bizonyosodjon meg róla, hogy az eszközén rajta van a XAMPP Control Panel, illetve a Visual Studio. Indítsa el a XAMPP Control Panelt, majd indítsa el az „Apache” és „MySQL” modulokat. Nyomja meg a „MySQL” modul „admin” feliratú gombját, és ellenőrizze, hogy létezik-e a „szalloda” című adatbázis, benne a megfelelő adatokkal. amennyiben nem, hozza létre az adatbázist és pótolja tartalmát az „sql” mappában szereplő „szalloda.sql” file segítségével. Ezután nyissa meg a projektet a Visual Studioban és futtassa le.

* 1. Vendég funkciók

A vendégeknek és alap felhasználóknak kizárólag a weboldalhoz van hozzáférésük, viszont magasabb ranggal rendelkező felhasználók hozzáférhetnek az asztali alkalmazáshoz (lásd: Alkalmazotti funkciók). Az oldal megnyitásakor, a felhasználók a főoldalon találják magukat. (Lásd: 19. ábra) Itt olvasható egy leírás a RushWave Retreat cégéről, illetve az oldal tetején található egy körhinta menü (Lásd: 20. ábra), melyben oldalaink különböző városai elérhetőek.

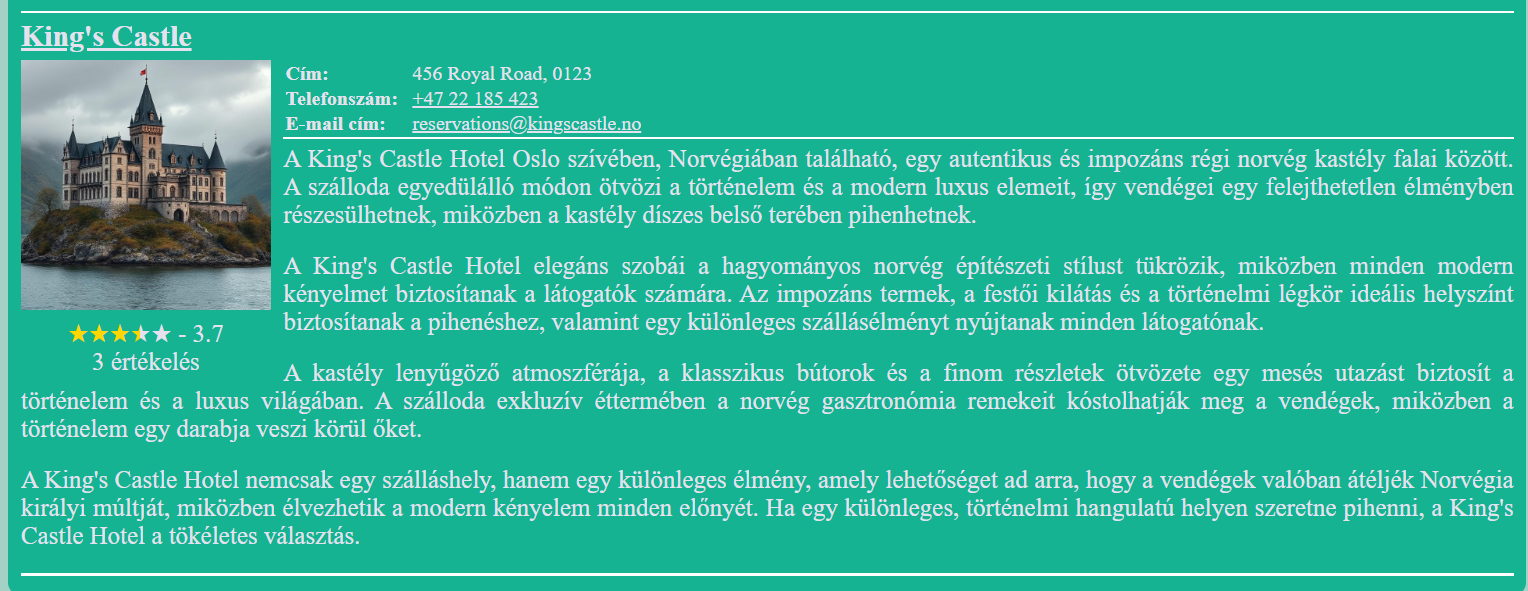


19. ábra: A főoldal



20. ábra: A körhinta menü

Innen a felhasználó számos irányba tud elindulni. A körhinta menüből ki tudja választani valamelyik várost, ahol olvashat egy turista ajánlót a helyről, illetve ki listázza azon szállodákat, amelyek abban a városban találhatóak. Minden szállodának található egy rövid leírása, illetve az átlag értékelése is. (Lásd: 21. ábra).



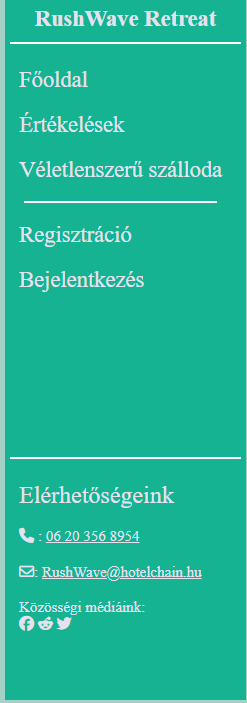
21. ábra: Egy hotel leírása

Innen a felhasználó megtudja nyitni a szálloda adatlapját. Itt találhat egy leírást a szállodáról és egy képgalériát, (Lásd: 23. ábra). Itt található továbbá a szobák listája ahol a felhasználó látja minden szoba adatát, illetve tájékoztatva van arról hogy a foglaláshoz bekell jelentkeznie (Lásd: 22. ábra). Ezentúl láthatók a szállodához írt értékelések, ezzel egy remek képet kapva arról, hogy a szálloda mit tud nyújtani. Az oldal alján található ismét egy leírás a városról, viszont ezúttal egy rövidebb és tömörebb változata. Ennek segítségével visszatud menni a város oldalára, vagy az oldalsávban (Lásd: 24. ábra) elhelyezett menüpontok segítségével a fő oldalra is. Itt található továbbá az értékelések fül, amelyre rányomva a felhasználó megtekintheti az oldal értékeléseit, illetve a bejelentkezés és regisztráció gombok is. Végezetül van egy véletlenszerű szálloda gomb, amely egy random szálloda adatlapjára irányít.

22. ábra: Egy szálloda szobáinak listája

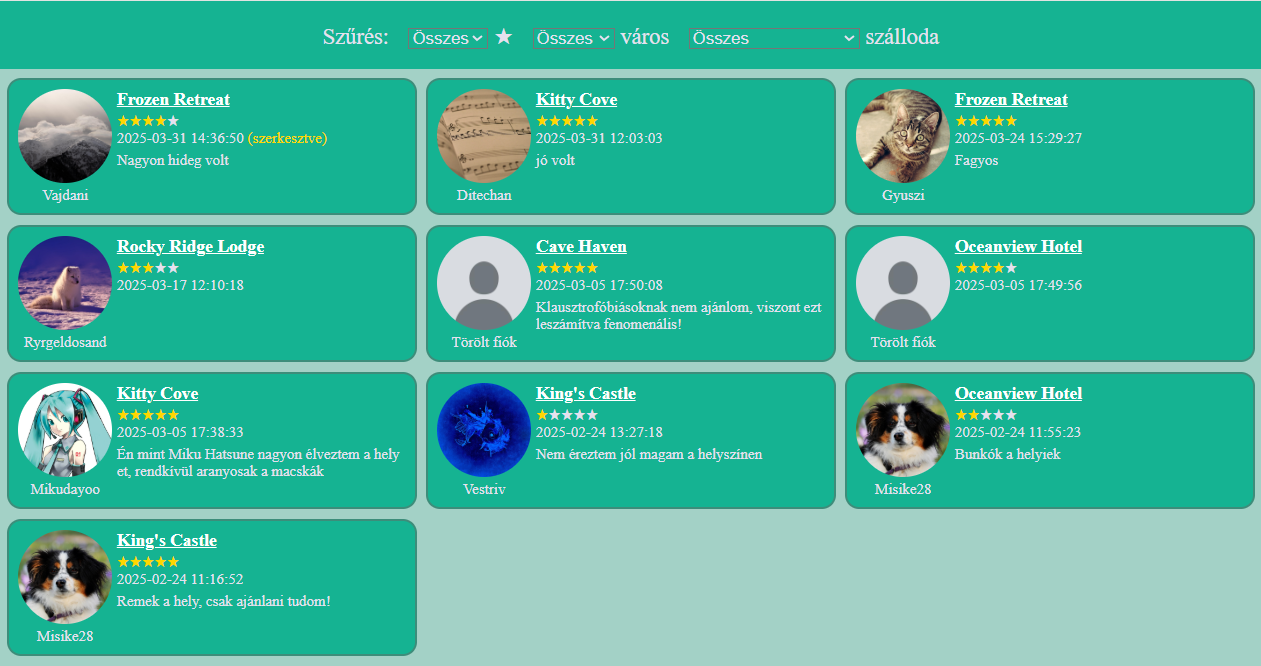


23. ábra: Egy szálloda adatlapja távoli nézetből



24. ábra: Az oldalsáv

Az értékelések oldalon megjelenik minden értékelés amely a weboldalra lett írva, (Lásd: 25. ábra) közzétételi idő szerint rendezve. Az értékelések 3 szempont szerint szűrhetőek. Az alapján hogy hány csillagosak, hogy melyik város, illetve azon belül, hogy melyik szállodához lett írva az értékelés. Természetesen nem kötelező megadni a várost ahhoz, hogy szálloda alapján lehessen szűrni, viszont amennyiben a felhasználó kiválaszt egy várost, onnantól kizárólag az abban elhelyezett szállodákat tudja kiválasztani szűrésnél, ezzel elkerülve lehetetlen szűrési kombinációkat.



25. ábra: A felhasználói értékelések

Az oldalsáv segítségével a felhasználó át tud lépni a regisztrációs oldalra, ahol létre tudja hozni a fiókját.(Lásd: 26. ábra) Itt a felhasználónak helyesen kikell töltenie a megadott adatokat, máskülönben képtelen lesz regisztrálni. A jelszónak tartalmaznia kell legalább 8 karaktert, kis és nagy betűt, számot, és speciális karaktert. Az oldal alján egy hivatkozás a felhasználói feltételekhez, melyeket a felhasználónak elkell fogadni amennyiben használni szeretné az oldalt.



26. ábra: A regisztrációs felület

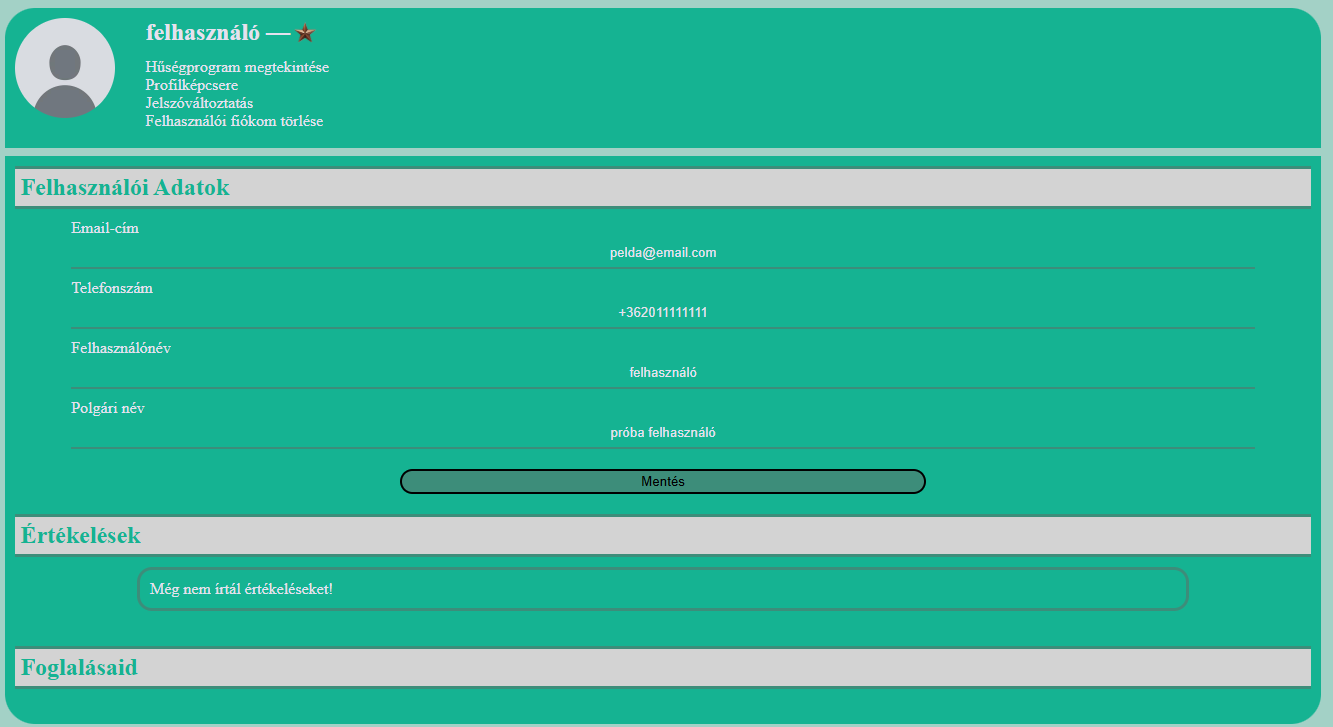
Sikeres regisztráció után a felhasználó értesítést kap, majd átirányításra kerül a belépési oldalra. Itt használhatja a felhasználónevét vagy email címét is a bejelentkezéshez. Amennyiben helyesen tölti ki a bejelentkezési adatokat, akkor hozzáférhet a felhasználói funkciókhoz.



27. ábra: A bejelentkezési felület

* 1. Felhasználói funkciók

A bejelentkezett felhasználó ismét a főoldalon találja magát pár apró változással. Az oldalsávból eltűnik a „regisztráció” és „bejelentkezés” gomb, helyükre egy „Kijelentkezés” gomb kerül, illetve hozzáadódik egy „profilom” felirat is. Erre rányomva megnyílik a felhasználó saját profilja (Lásd: 28. ábra). Itt megtekintheti a saját adatait, amiket módosítani is tud, illetve a későbbiekben itt tekintheti meg a saját értékeléseit és foglalásait. A felhasználó neve alatt található 4 további menüpont. Ezek segítségével megtudja tekinteni a hűségszintjét, megtudja változtatni a profilképét, illetve képes módosítani a jelszavát.



28. ábra: Egy újonnan létrehozott fiók profil oldal

A bejelentkezéssel a felhasználó már tud foglalni is a szálloda adatlapján. Itt kiválasztja, hogy mettől meddig szeretne foglalni, majd az ellátást és extra szolgáltatásokat. A megadott dátumok alapján az oldal automatikusan betölti azokat a szobák, amik szabadok a foglalás időszaka alatt. A szekció alján található egy összeg, ami dinamikusan változik, ahogy a felhasználó beviszi az adatokat (Lásd: 29. ábra). Amennyiben a felhasználó már elért egy magasabb hűségrankot, az abból származó akció is egyből alkalmazva van (Lásd: 30. ábra). Ezek után a felhasználónak kikell tölteni a számlázási címét illetve a fizetés módszerét (Lásd: 31. ábra). Sikeres foglalás esetén a felhasználó visszakerül a hotel adatlapjára, és értesítést kap a sikeres foglalásról.



29. ábra: Foglalás felület, hűségprogrami akció nélkül



30. ábra: Foglalási felület, hűségprogrami akcióval



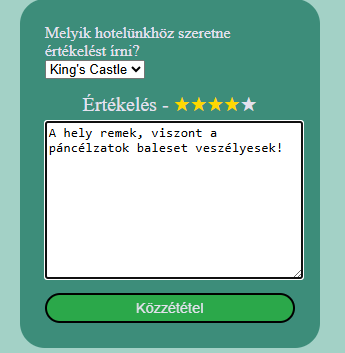
31. ábra: Számlázási adatok felülete

Foglalás után a felhasználó a profilján megtudja tekinteni a foglalásait. Itt minden korábbi foglalása megjelenik, foglalási idő szerint (Lásd: 32. ábra). A felhasználó tud visszatérítést kérvényezni amennyiben rájön, hogy mégse lenne optimális az időpont. Ennek eredményeként a foglalás állapota megváltozik, majd miután az alkalmazottak elfogadták a kérést, hivatalosan lemondottnak számít a foglalás. Amennyiben a felhasználónak van már befejezett foglalása, tud értékelést írni az adott szállodához. Foglalásonként egy értékelést tud írni a felhasználó, ezzel elkerülve a rosszindulatú értékelések tömeges beküldését.



32. ábra: Példa mind a négy foglalás típusra

Az értékelés írása egy gyors és egyszerű folyamat. Amennyiben a felhasználónak több befejezett foglalása is van, kiválasztja, hogy melyik szállodához szeretné írni. Ezután megadja, hogy hány csillagot szeretne adni, ezután amennyiben a felhasználó óhajtja, kifejtheti véleményét szöveges formában is (Lásd: 33. ábra). Közzététel után az értékelés megjelenik a profilon, ahol a felhasználó módosíthatja annak szövegét, vagy amennyiben vissza szeretné vonni, akkor törölheti is. Szerkesztett értékelés esetén megjelenik egy „szerkesztve” felirat is, ezzel jelezve a felhasználónak, hogy sikeres volt a szerkesztés. Amennyiben egy értékelés törlésre kerül, a felhasználó ismételten tud írni értékelést az adott szállodához. Az eredeti értékelés az adatbázisban tárolva marad, viszont a szálloda átlag értékelésébe nem számít bele.



33. ábra: Példa egy értékelés írására

Amennyiben a felhasználó úgy dönt, hogy profilját törölni szeretné, erre természetesen van lehetősége. A profilján található egy „felhasználói fiókom törlése” felirat. Erre rákattintva a felhasználó átkerül a törlési adatlapra, (Lásd: 34. ábra) ahol még van lehetősége meggondolnia magát. Az értékelései alapvetően továbbra is meg tekinthetőek, viszont a felhasználónév helyett csupán egy „törölt fiók” látszik (Lásd: 35. ábra). Viszont amennyiben a felhasználó szeretné, hogy az értékelései is eltűnjenek, egyszerűen betudja pipálni az adatlapon elhelyezett jelölőnégyzetet, amely gondoskodik arról, hogy a felhasználó minden értékelését törölje. A törlés után a felhasználó nem tud visszalépni a fiókjába, viszont profilja ugyanúgy megtekinthető, és amennyiben nem törölte az értékeléseit ott ugyanúgy megtekinthetőek.



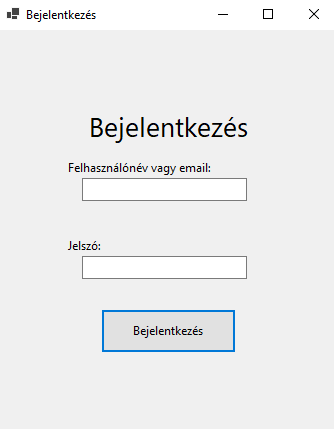
34. ábra: A fiók törlési adatlap



35. ábra: Egy törölt fiók által írt értékelés

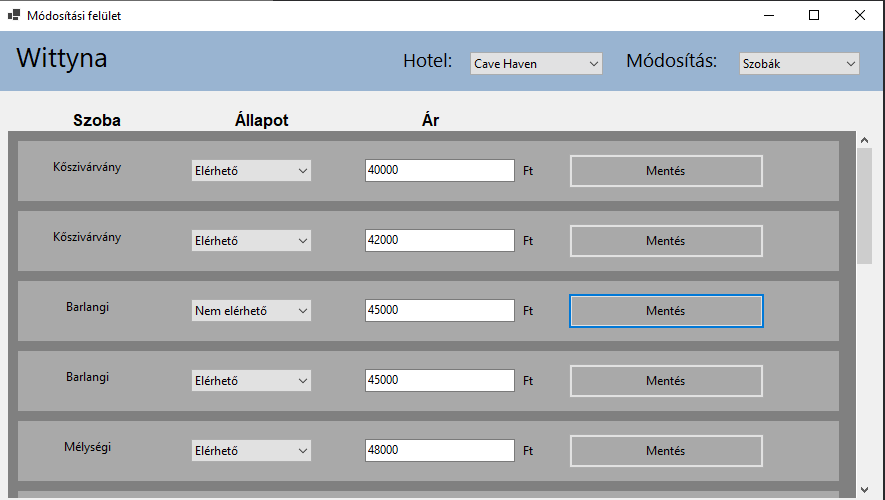
* 1. Alkalmazotti funkciók.

Az alkalmazottak engedélyt kapnak egy külön asztali alkalmazáshoz, ahol képesek hozzáférni a szállodák adatbázisához, és ennek tartalmát szerkeszteni. Természetesen minden alkalmazott kizárólag annak a szállodának az adatait tudja szerkeszteni, ahol dolgozik. Egy alkalmazott dolgozhat egy időben több különböző szállodának is, ilyenkor egy lenyíló menüből képes kiválasztani, hogy melyik szálloda adatait szeretné kezelni. Az alkalmazás először bejelentkezést igényel (Lásd: 36. ábra) ezzel leellenőrizve, hogy ténylegesen alkalmazott szeretne-e bejelentkezni. Amennyiben a megadott felhasználónév egyik hotelnek sem alkalmazottja, hibaüzenet kap. Amennyiben a megadott felhasználónév és jelszó is helyes, és a felhasználó ténylegesen egy alkalmazott, akkor a program tovább engedi.



36. ábra: Az asztali alkalmazás bejelentkező felülete

Megnyílik az adminisztrátori panel, amely alapvetően a szobák kezelését nyitja meg, (Lásd: 37. ábra) ahol a szobák esténkénti árát lehet módosítani, illetve a jelenlegi elérhetőségét. Amennyiben a szobát „nem elérhető”-re állítja a felhasználó, a szobát nem lehet lefoglalni, amíg azt vissza nem állítjuk elérhetőre, illetve a már meglévő foglalásokra visszatérítést igényel. Ez a funkció hasznos lehet mikor hosszabb takarítást, vagy egyéb hasonló ok miatt a szoba több napig nem elérhető. Változtatások után a mentés gombra kattintva tudjuk rögzíteni az új adatokat. Fontos megjegyezni, hogy ez a gomb egyéni a szobához, és nem menti el más esetlegesen módosított szobák adatait.



37. ábra: A szobák módosítására való ablak

A fejlécben található kettő legördülő lista (Lásd: 38 ábra). Az egyikkel az éppen kezelt hotelt lehet átállítani, amíg a másikkal a kiválasztott hotel adatait. Ennek értékei alkalmazottként „szobák”, „szolgáltatások” és „foglalások”. Ezekre kattintva frissül az ablak és betölti azokat az adatokat, amelyeket kiválasztott. A foglalások felépítése hasonló a szobáéhoz (Lásd: 39. ábra), azzal az eltéréssel, hogy itt több adat megjelenik a foglalásokról, viszont ezek nem módosíthatók. Itt azok a már letelt foglalások jelennek meg, amelyeket lekell validálnia az alkalmazottnak, hogy ténylegesen megtörténtek, illetve azok a foglalások, amelyekhez a felhasználó visszatérítést igényelt. Mindkettő csupán egyetlen gombnyomás, ezzel megkönnyítve az alkalmazottak dolgát. Ezek frissítik a számla állapotát is, illetve amennyiben a felhasználó úgy döntött készpénzzel fizettet, a számla fizetési dátumát is beállítja.

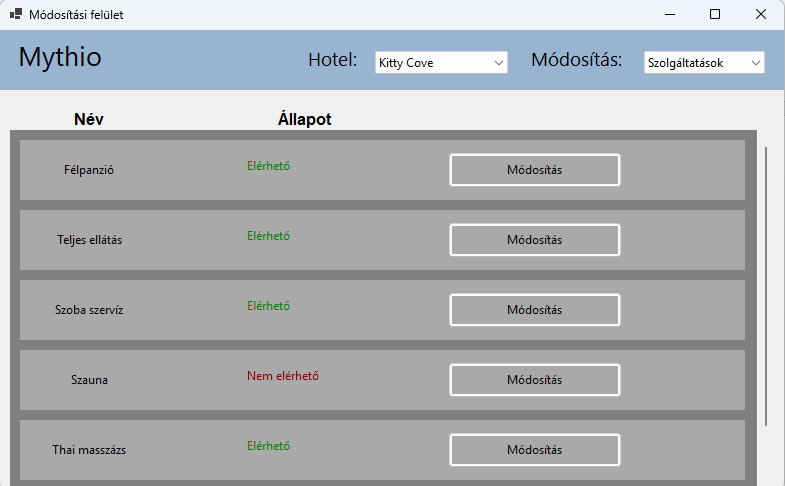


38. ábra: A legördülő listák



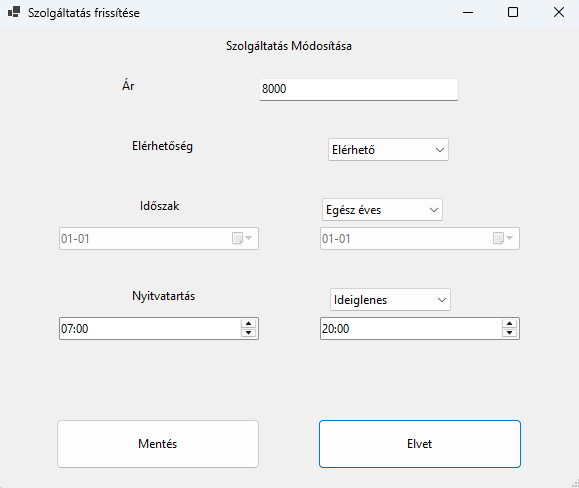
39. ábra: A foglalások levalidlására szolgáló felület

Az utolsó alkalmazottaknak szánt rész a szolgáltatások módosítása (Lásd: 40. ábra). Ide sajnos nem tudtuk megvalósítani, hogy minden módosítandó adatot kiírjunk a panelre, ezért gombnyomásra saját ablak nyílik meg a szerkesztésre.



40. ábra: A szolgálatásokat megjelenítő panel

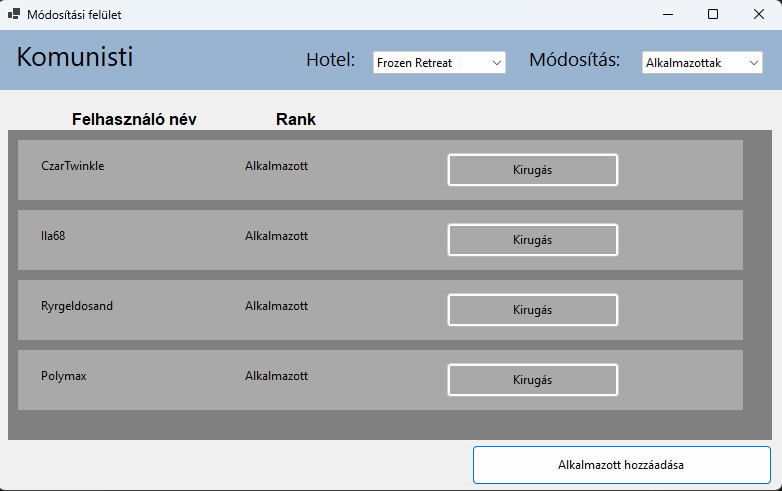
A szolgáltatást számos módon lehet módosítani (Lásd: 41. ábra). Átírható az igénybevétel ára, illetve az elérhetőség, ami ugyanúgy működik, mint a szobáknál. Ezentúl látható egy időszak, ami lehet egész évben nyitvatartott, vagy ideiglenes. Utóbbi esetben megkell adni a nyitás és zárás időtartamát. Továbbá van egy napközbeni nyitva tartás, amely szintúgy lehet egész napos vagy ideiglenes. Itt is megkell adni a nyitás és zárás idejét amennyiben ideiglenesre lett állítva



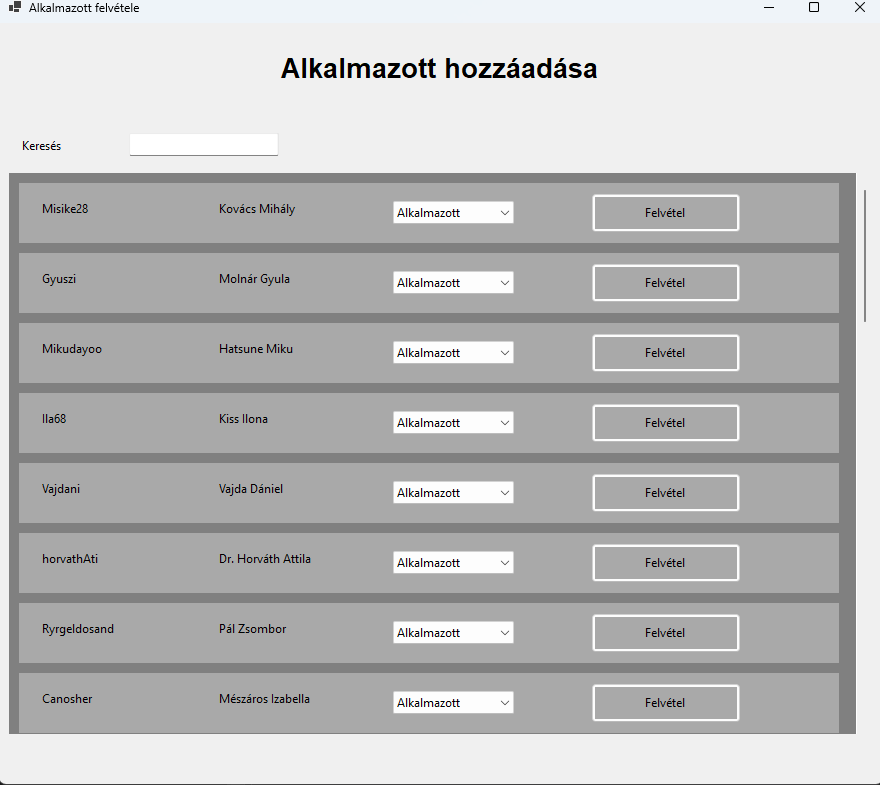
41. ábra: A szolgáltatás módosítására szolgáló ablak

* 1. Manageri funkciók

A managerek számára elérhetővé válik egy új lehetőség a legördülő listában, ami pedig az alkalmazottak kezelése. A manager itt láthatja a szállodája összes alkalmazottját, akiket ki is tud rúgni amennyiben úgy gondolja.(Lásd: 42. ábra) Továbbá található még egy alkalmazott hozzáadása gomb. Itt bármelyik regisztrált felhasználót képes felvenni a szállodához, mint alkalmazott (Lásd: 43. ábra), ezzel elérhetővé teszi számára az asztali alkalmazásba való bejelentkezést. Alkalmazott felvételnél van egy kereső mező, ahol felhasználó név alapján lehet szűrni.



42. ábra: Az alkalmazottak listája



43. ábra: Alkalmazott felvétele

Továbbá elérhető a managerek számára a szálloda statisztika oldala (Lásd: tulaj funkciók), ahol képesek megtekinteni, hogy mennyi jövedelmük volt, illetve hogy melyik hónapban foglaltak a legtöbben.

* 1. Tulaj funkciók

A tulaj birtokolja az egész szálloda láncot, így neki van a legtöbb elérhetősége a szállodákhoz. Minden szállodához hozzáfér, és mindenre képes, amit az alkalmazottak és managerek. Természetesen mivel minden szállodához hozzáfér, képes mindegyiknek a statisztikáit megnézni (Lásd: 44. ábra), illetve ő képes kinevezni a hotelekhez a managereket is.



44. ábra: A statisztikai adatlap

1. Továbbfejlesztési lehetőségek

Az projekt fejlesztése során számos olyan ötlet jutott eszünkbe, amelyeket szerettünk volna megvalósítani, viszont idő szűkében, vagy a kellő tudás hiánya miatt sajnos nem tudtuk implementálni. Vannak olyan, amik dizájn szempontból lennének jók, vagy a funkcionalitását egyszerűsítenék az oldalnak.

* Egyéni profilkép feltöltése:  
  Célunk lett volna az, hogy a felhasználó képes legyen feltölteni a saját profilképét, viszont ezt sajnos nem sikerült megvalósítani. Nem voltunk biztosak abban, hogy ilyen adatokat pontosan hogyan kellene eltárolni, illetve további problémát jelentett volna a tartalom szűrése is, hiszen lehettek volna olyan felhasználók, akik illetlen dolgokat állítottak volna be profilképüknek.
* Mások profiljának érdekesebbé tétele:  
  Törekedni akartunk arra, hogy mások profiljának böngészése jövedelmezőbb legyen, hiszen jelenleg csak az illető nevét illetve értékelései megtekinthetőek. Sajnos nem tudtuk megvalósítani, hogy a felhasználó publikussá tudja tenni bizonyos adatait, például az email címét vagy telefonszámát. Gondolkoztunk továbbá azon, hogy az alkalmazottaknak egyéni profilkép keretük lenne, ami reprezentálja a hotelt, amelynél dolgoznak.
* Hűségprogram fejlesztése  
  Szerettünk volna többet kihozni a hűségprogramból, hiszen jelenleg kizárólag akciót biztosít, illetve egy pár olyan szolgáltatást, amit kizárólag személyesen lehetne beváltani. Úgy érezzük többet kilehetett volna hozni ebből az ötletből
* Külső alkalmazások használata  
  Amennyiben egy felhasználó elfelejti a jelszavát, akkor tudjon egy validáló emailt kérni, mely segítségével visszajuthat a fiókjába. Sajnos ez egy túlságosan komplikált feladat lett volna, így ez nem került megvalósításra
* Új hotel felvitele a rendszerbe  
  Az asztali alkalmazás eredetileg tartalmazott volna a tulajnak egy egyéni szekciót, amelyben egy teljesen új hotelt tudott volna felvinni a rendszerbe, képekkel és leírással együtt. Sajnos ez túl nagy falatnak bizonyult, így a tulajnak nincs semmilyen egyéni képessége a manager felvételen kívül.
* Számlázás külön oldalon  
  Ez volt a legközelebb a megvalósításhoz, viszont sajnos nem sikerült. Eredetileg a foglalást és számlázást két külön oldalra vettük volna, majd mikor mindkettőt kitölti a felhasználó akkor rögzül az adatbázisba. Sajnos nem tudtuk úgy megoldani, hogy minden hiba mentesen működjön vagy úgy, hogy ne lehessen egyszerűen rosszindulattal manipulálni a rendszert

1. Összegzés

A vizsga remekünk célja egy felhasználó barát szálloda foglaló oldal létrehozása volt, aminek sikerült megfelelnünk. Sajnos az oldal nem tökéletes, van rajta kifogásolni való, viszont az eredeti céljának szerintünk tökéletesen megfelel. Nincsenek az oldalon könnyen kihasználható hibák, vagy olyan elemek, amik nehezítenék az oldal használatát.

Számos problémába ütköztünk fejlesztés során, mint például a határidők követése, vagy egyenlően felosztani a feladatokat. Ezeknek az akadályoknak az ellenére is sikerült megbírkóznunk a feladattal, és befejezni a projektet. A csapat minden tagja mindent megtett annak érdekében, hogy a projekt teljesen kikerekedjen.

Számos új kihívást állítottunk magunk elé a vizsga remekkel, és rengeteg újat tanultunk C sharp, Laravel, JavaScript és SQL terén is. Igyekeztünk teljesen saját stílust csinálni, és nem egy keretrendszer segítségével megírni a dizájnt. Ez nagyrészt sikerült is, az egyetlen hely ahol keretrendszert használtunk az a főoldalon lévő körhinta menü.

Sajnos mivel a csapatunk olyan emberekből állt össze, akik a programozásnak elsősorban a backend részéhez értenek, ezért a designunk eléggé minimalista, ez a legnagyobb kifogásunk az egész projekttel. Ettől eltekintve viszont büszkék vagyunk arra, amit létrehoztunk.

1. Ábrajegyzék

[1. ábra: EER diagram a "szalloda" adatbázisról 14](#_Toc195522334)

[2. ábra: A regisztráció függvény adatbázisba illesztő része 15](#_Toc195522335)

[3. ábra: A foglalás függvény árat számoló része 16](#_Toc195522336)

[4. ábra: A foglalás függvény adatbázisba író része 17](#_Toc195522337)

[5. ábra: A foglalás függvény hűségpontszámító része 17](#_Toc195522338)

[6. ábra: A szolgálatás kikéréséért felelős adatbázis lekérés 18](#_Toc195522339)

[7. ábra: A felhasználói adatokat módosító függvény 18](#_Toc195522340)

[8. ábra: Az email egyéniségét ellenőrző függvény 19](#_Toc195522341)

[9. ábra: Az értékelés írásra engedélyt adó függvény 19](#_Toc195522342)

[10. ábra: Az értékelés szűrést végző függvény 20](#_Toc195522343)

[11. ábra: Az értékelés szekciók létrehozásáért felelős függvény 20](#_Toc195522344)

[12. ábra: A hűségmenü ablakát létrehozó függvény 21](#_Toc195522345)

[13. ábra: Példa egy panel létrehozására 22](#_Toc195522346)

[14. ábra: A panelek reszponszivitásáért felelős függvény 23](#_Toc195522347)

[15. ábra: Az osztályok alapján történő form létrehozás 24](#_Toc195522348)

[16. ábra: Első fele egy osztálynak, amely beviteli elemeket hoz létre 25](#_Toc195522349)

[17. ábra: Második fele egy osztálynak amely beviteli elemeket hoz létre 26](#_Toc195522350)

[18. ábra: Példa az adatbázis frissítésére 27](#_Toc195522351)

[19. ábra: A főoldal 30](#_Toc195522352)

[20. ábra: A körhinta menü 31](#_Toc195522353)

[21. ábra: Egy hotel leírása 32](#_Toc195522354)

[22. ábra: Egy szálloda szobáinak listája 32](#_Toc195522355)

[23. ábra: Egy szálloda adatlapja távoli nézetből 33](#_Toc195522356)

[24. ábra: Az oldalsáv 34](#_Toc195522357)

[25. ábra: A felhasználói értékelések 35](#_Toc195522358)

[26. ábra: A regisztrációs felület 35](#_Toc195522359)

[27. ábra: A bejelentkezési felület 36](#_Toc195522360)

[28. ábra: Egy újonnan létrehozott fiók profil oldal 37](#_Toc195522361)

[29. ábra: Foglalás felület, hűségprogrami akció nélkül 38](#_Toc195522362)

[30. ábra: Foglalási felület, hűségprogrami akcióval 38](#_Toc195522363)

[31. ábra: Számlázási adatok felülete 39](#_Toc195522364)

[32. ábra: Példa mind a négy foglalás típusra 39](#_Toc195522365)

[33. ábra: Példa egy értékelés írására 40](#_Toc195522366)

[34. ábra: A fiók törlési adatlap 41](#_Toc195522367)

[35. ábra: Egy törölt fiók által írt értékelés 41](#_Toc195522368)

[36. ábra: Az asztali alkalmazás bejelentkező felülete 42](#_Toc195522369)

[37. ábra: A szobák módosítására való ablak 43](#_Toc195522370)

[38. ábra: A legördülő listák 44](#_Toc195522371)

[39. ábra: A foglalások levalidlására szolgáló felület 44](#_Toc195522372)

[40. ábra: A szolgálatásokat megjelenítő panel 45](#_Toc195522373)

[41. ábra: A szolgáltatás módosítására szolgáló ablak 46](#_Toc195522374)

[42. ábra: Az alkalmazottak listája 47](#_Toc195522375)

[43. ábra: Alkalmazott felvétele 48](#_Toc195522376)

[44. ábra: A statisztikai adatlap 49](#_Toc195522377)