Tartalom

[I. Bevezetés 2](#_Toc195257107)

[1. Áttekintés 2](#_Toc195257108)

[2. Közös munka 2](#_Toc195257109)

[II. Projektmunka általános ismeretei 4](#_Toc195257110)

[1. Felhasznált programok 4](#_Toc195257111)

[2. Program Működése 6](#_Toc195257112)

[3. Felhasználói élmény 7](#_Toc195257113)

[4. Adminisztrációs felület 9](#_Toc195257114)

[5. Célközönség 10](#_Toc195257115)

[III. Architektúra 12](#_Toc195257116)

[1. Frontend 12](#_Toc195257117)

[2. Backend 15](#_Toc195257118)

[3. Adatbázis 19](#_Toc195257119)

[IV. Biztonság 23](#_Toc195257120)

[1. Adatvédelem 23](#_Toc195257121)

[2. Felhasználói adatok védelme 23](#_Toc195257122)

[V. Tesztelés 24](#_Toc195257123)

[1. Frontend (Selinium teszt) 24](#_Toc195257124)

[5. Backend (Jest teszt) 27](#_Toc195257125)

[VI. Befejezés 28](#_Toc195257126)

[1. Fejlesztési lehetőségek 28](#_Toc195257127)

[2. Jövőbeli tervek 29](#_Toc195257128)

[3. Záró gondolatok 29](#_Toc195257129)

# Bevezetés

## Áttekintés

Célunk egy modern, felhasználóbarát szálloda kezelő weboldal megvalósítása volt, amely nemcsak a vendégek számára nyújt egyszerű és gyors foglalási lehetőséget, hanem az adminisztrátorok számára is hatékony eszközt biztosít az adott szálloda kezelésére és nyomon követésére. Az általunk tervezett rendszer átlátható és könnyen kezelhető felületet kínál, amely segíti a vendégeket abban, hogy hatékonyak tudjanak navigálni a weboldalon.

A weboldal emellett integrált szoba és étteremfoglalási és rendelési funkciókat is tartalmaz. A vendégek lehetőséget kapnak arra, hogy előre asztalt foglaljanak a szálloda éttermében, valamint kényelmesen, online leadhassák rendeléseiket. Ezáltal a vendéglátás folyamata gördülékenyebbé és rugalmasabbá válik, biztosítva a kényelmes és kellemes tartózkodást.

A vendégélmény fokozása érdekében egy innovatív pontrendszert is bevezettünk. A rendszer célja, hogy ösztönözze a visszatérő vendégeket, és még vonzóbbá tegye számukra a szálloda és annak szolgáltatásainak igénybevételét. A vendégek minden egyes foglalás, valamint éttermi fogyasztás után hűségpontokat kapnak, amelyeket később kedvezményekre vagy akár ingyenes szolgáltatásokra válthatnak be a szállodában és annak partnereinél.

A pontrendszer nemcsak a hotel szolgáltatásain belül használható fel, hanem a szálloda partnereinél is, így például közeli éttermekben, boltokban és egyéb szolgáltatóknál. A partnerhálózat folyamatosan bővül, így a vendégek egyre szélesebb körben élvezhetik a hűségprogram előnyeit. Ez a rendszer nemcsak a vendégek számára biztosít extra értéket, hanem a helyi vállalkozások számára is lehetőséget teremt arra, hogy új ügyfeleket szerezzenek és erősítsék együttműködésüket a szállodával.

## Közös munka

A projektmunkánkat ketten valósítottuk meg, Péter Kristóf és Jakab Zsombor. A projekt megvalósításában mindketten egyenlő arányban vettünk részt, a munkát előre felosztottuk, hogy hatékonyan tudjuk kivitelezni a feladatokat. A közös munka során nagy hangsúlyt fektettünk a folyamatos kommunikációra és az átláthatóságra, így a fejlesztési folyamat gördülékenyen zajlott.

A projektfeladat megvalósítása során a kommunikáció és a munka szervezése érdekében a Discordot használtuk, ahol folyamatosan egyeztettünk a fejlesztés aktuális állásáról és a felmerülő problémákról. A kódbázist GitHub-on tároltuk, hogy mindketten könnyen hozzáférjünk és szükség esetén módosíthassuk azt. A verziókezelés lehetővé tette számunkra, hogy párhuzamosan dolgozzunk a különböző funkciókon anélkül, hogy adatvesztéstől kellett volna tartanunk. Az adatok és dokumentációk megosztására a Google Drive-ot használtuk, ahol minden szükséges fájlt elérhetővé tettünk egymás számára.

A projekt első lépéseként közösen megterveztük az adatbázist az Access program segítségével. Felépítettük a szükséges táblákat, definiáltuk a mezőket és az azokhoz tartozó funkciókat, majd megteremtettük a táblák közötti kapcsolatokat. Az adatbázis struktúrájának kialakításánál ügyeltünk arra, hogy az optimálisan támogassa a későbbi fejlesztéseket és könnyen kezelhető legyen.

Ezután a backend fejlesztésére koncentráltunk, amelyet Node.js segítségével valósítottunk meg. Mindketten aktívan részt vettünk az API-k kialakításában és az adatbázis kezelésében.

A frontend fejlesztését React segítségével végeztük, ahol szintén mindketten aktívan dolgoztunk. A React komponensalapú struktúrája lehetővé tette, hogy a munkát hatékonyan osszuk fel egymás között. A felhasználói felület kialakításánál a reszponzivitás és a modern dizájn volt a fő szempont.

A dokumentáció elkészítésére is nagy figyelmet fordítottunk. Részletesen leírtuk a projekt felépítését, az alkalmazott technológiákat, az adatbázis struktúráját, valamint a frontend és backend funkcióit. A projekt sikeres megvalósításához elengedhetetlen volt a hatékony csapatmunka és a folyamatos kommunikáció, amely lehetővé tette számunkra, hogy a feladatokat egyenlően osszuk el és a kitűzött célokat időben teljesítsük.

# Projektmunka általános ismeretei

## Felhasznált programok

1. *GitHub –* A programkódót erre az alkalmazásra töltöttük fel, így közösen tudtuk szerkeszteni. A GitHub verziókezelési rendszere lehetővé tette, hogy minden módosítást nyomon kövessünk, és szükség esetén visszaállítsuk a korábbi állapotokat. Valamint A GitHub lehetővé tette, hogy az összes programkódot biztonságosan tároljuk egy központi helyen, amelyhez mindketten hozzáfértünk bármikor és bármilyen eszközről.
2. *Figma –* A design-t az asztali alkalmazásra és a telefonos applikációra is ebben a programban reszponzívan meg tudtuk tervezni.A program valós idejű kollaborációt tett lehetővé, így mindketten egyszerre dolgozhatunk ugyanazon a projekten.
3. *Google Drive –* Az egyéb dokumentumokat pl.: projekthez tartozó dokumentációt, képeket erre a felhő alapú szolgáltatáson keresztül osztottuk meg. Így minden fontos anyag egy helyen elérhető volt, akár telefonról is. A Drive együttműködik más Google szolgáltatásokkal, például a Gmail-lel, ami megkönnyítette a fájlok megosztását és az együttműködést*.*
4. *Discord –* A folyamatok kapcsolattartásunkat, a mindennapi kommunikáció használatát a legismertebb chatprogram, mely hangot, képet, videót, és dokumentációt is meg tudtunk osztani. A közös munka miatt szükségünk volt megosztani a képernyőt a program írása közben, ami nélkülözhetetlen volt számunkra a projekt megvalósítása során. Ez lehetőséget biztosított arra, hogy problémák esetén közösen elemezzük a kódot és megoldásokat találjunk, visszajelzéseket adhatunk egymásnak.
5. *Visual Studio Code –* A Microsoft programja, nyílt forráskódú kódszerkesztő ami több programnyelvet ismert illetve ingyenes, számunkra azért volt hatékony ez a program, mert GitHubbal is kompatibilis ami elkerülhetetlen volt a projekt megvalósításához
6. *Microsoft Office Programok -* A projekthez kapcsolódó dokumentációt Microsoft Word segítségével készítettük el, amely lehetőséget biztosított a szövegformázásra, kialakítására, valamint a képek beillesztésére. Az adatbázisunk tervezését és kezelését a Microsoft Access programban végeztük el, ahol létrehoztuk a kapcsolattáblákat, definiáltuk az adatkapcsolatokat, valamint különböző lekérdezéseket, szűréseket, jelentéseket és űrlapokat hoztunk létre amelyek segítették az adatkezelési folyamatainkat. A projekt bemutatásához a **Microsoft PowerPoint** programot használtuk amelyben sablonokat, képeket, áttünéseket és animációkat is hozzáadtunk hogy a prezentáció átlátható, látványos és informatív legyen.
7. *Xampp* - Egy szabad és nyílt forráskódú, platformfüggetlen webszerver-szoftvercsomag, amelyet a projekt során használtunk. A XAMPP lehetővé tette számunkra, hogy egy helyi fejlesztői környezetet hozzunk létre, ahol biztonságosan és költségek nélkül tesztelhettük a webes alkalmazásunkat. Legfőbb alkotóelemei közé tartozik az Apache webszerver, amely biztosította a helyi weboldalak futtatását, valamint a MySQLadatbázis-kezelő, amelyet az adatok tárolására és kezelésére használtunk.
8. *php Myadmin –* MySQL programnyelven keresztül itt hoztuk létre az adatbázist, valamint itt tudtunk készíteni, eldobni, módosítani, táblákat és törölni módosítani hozzáadni mezőket. A Xampp szerverre töltöttünk fel, ameddig nem volt saját/iskolai szerverünk ahova fel tudtuk tölteni a vizsgamunkát.

## Program Működése

A szállodai rendszer egy komplex webalkalmazás, amely lehetővé teszi a vendégek számára a szobafoglalást, valamint a szálloda adminisztrátorainak a foglalások és vendégadatok kezelését. A rendszer központi eleme egy SQL adatbázis, amely többek között a szobák adatait tárolja különböző kategóriákban, mint például Standard, Superior, Deluxe, Business, Luxus és Penthouse Suite.

A rendszer a szobákat részletes leírással, képekkel és árakkal együtt tárolja. Minden szobatípus különböző felszereltséggel rendelkezik, amelyek JSON formátumban vannak tárolva az adatbázisban. A szobák árai kategóriánként változnak, a Standard szoba 35.000 Ft-tól kezdődően egészen a Penthouse Suite 120.000 Ft-os áráig.

A rendszer egyik kiemelkedő funkciója a pontgyűjtési rendszer, amely a törzsvendégek számára előnyöket biztosít. A vendégek minden foglalás után pontokat kapnak, amelyek az elköltött összeg arányában kerülnek jóváírásra. Ezeket a pontokat később felhasználhatják kedvezmények igénybevételére vagy akár ingyenes éjszakák foglalására. A pontgyűjtési rendszer ösztönzi a visszatérő vendégeket, és erősíti a vendéghűséget.

Az adminisztrációs felületen keresztül a szálloda személyzete kezelheti a foglalásokat, nyomon követheti a szabad és foglalt szobákat, valamint hozzáférhet a vendégek adataihoz. A rendszer automatikus értesítéseket küld a foglalásokról, módosításokról és lemondásokról mind a vendégek, mind az adminisztrátorok számára.

A szállodai rendszer biztonságos fizetési lehetőségeket kínál, és megfelel az adatvédelmi előírásoknak, biztosítva a vendégek személyes adatainak védelmét. Az adatbázis rendszeres biztonsági mentésekkel rendelkezik az adatvesztés elkerülése érdekében.

## Felhasználói élmény

A szállodai kezelő weboldal és rendszer fejlesztése során elsődleges célunk egy olyan felhasználóbarát felület kialakítása volt, amely egyszerű, intuitív élményt nyújt minden felhasználó számára. A platformot úgy terveztük, hogy különböző eszközökön – legyen szó asztali számítógépről, tabletről vagy mobiltelefonról – egyaránt zökkenőmentesen működjön, így biztosítva a kényelmes használatot bármilyen helyzetben. A modern, letisztult dizájn és az átlátható navigáció lehetővé teszi, hogy a felhasználók gyorsan és hatékonyan elérjék a számukra fontos funkciókat, legyen az szobafoglalás, éttermi rendelés vagy pontjaik kezelése.

Az alkalmazás felhasználói felülete reszponzív, így különböző eszközökön (asztali számítógép, tablet, mobiltelefon) is kényelmesen használható. A foglalási folyamat egyszerű és átlátható, a vendégek könnyen kiválaszthatják a kívánt szobatípust, a tartózkodás időtartamát, és megtekinthetik a foglalás részleteit a véglegesítés előtt

**Szoba foglalás:**

A rendszer különböző szobatípusokat kezel, amelyek mindegyike egyedi jellemzőkkel rendelkezik. Az alábbi szobatípusok érhetők el:

* Standard: Alapvető kényelmet biztosító szoba, ideális rövidebb tartózkodásra vagy költségtudatos vendégek számára.
* Superior: Kicsit tágasabb, modernebb berendezésű szoba, extra kényelmi funkciókkal.
* Deluxe: Elegánsabb dizájn, prémium felszereltséggel, nagyobb térrel.
* Luxus: Kifinomult belső tér, exkluzív szolgáltatásokkal, például minibárral vagy panorámás kilátással.
* Business: Üzleti utazók számára optimalizált szoba, íróasztallal, gyors WiFi-vel és tárgyalási lehetőségekkel.
* Penthouse: A legmagasabb kategória, tágas térrel, luxusfelszereltséggel és különleges extrákkal (pl. privát terasz).

**Minden szobatípushoz tartozó adatok:**

* Egyedi kép: Professzionális fotó a szoba hangulatának és stílusának bemutatására, amely vizuálisan vonzóvá teszi a választást.
* Név: Egyedi, megjegyezhető elnevezés, amely tükrözi a szoba karakterét (pl. "Panoráma Deluxe" vagy "Üzleti Oázis").
* Leírás: Rövid, de informatív szöveg, amely kiemeli a szoba előnyeit és célközönségét (pl. "Tökéletes választás pihenni vágyó pároknak").

**Felhasználói fiókok:**

A screenshot of a computer

AI-generated content may be incorrect.Regisztráció és bejelentkezés: A felhasználók e-mail-cím és jelszó megadásával regisztrálhatnak A felhasználók megtekinthetik korábbi foglalásaikat. itt tudják megtekintemi a pontjaikat, kuponjaikat, valamint a foglalási előzményeiket

**Pontrendszer:**

A pontrendszer egy olyan hűségprogram, amelynek célja a vendégek visszatérésének ösztönzése, miközben személyre szabott jutalmakat kínál számukra. A rendszer alapját egy pontgyűjtő mechanizmus képezi, amely lehetővé teszi, hogy a vendégek minden vásárlás alkalmával pontokat szerezzenek. Ezeket a pontokat később különféle kedvezményekre, ajándékokra vagy exkluzív ajánlatokra válthatják be, ezáltal még vonzóbbá téve a visszatérést és a hűséget.

A pontrendszer célja, hogy a visszatérő vendégeket jutalmazza, miközben motiválja az új ügyfeleket a rendszeres foglalásra. A különböző szintek és beváltási opciók lehetővé teszik, hogy a vendégek az igényeiknek megfelelő jutalmat válasszanak, legyen szó költségcsökkentésről, extra kényelemről vagy luxusélményről. Ezáltal a rendszer egyedi, személyre szabott vendégélményt biztosít, növelve a szálloda vonzerejét és versenyképességét.

## Adminisztrációs felület

Az admin felület egy átfogó irányítópultot biztosít a szálloda menedzsment számára, ahol különböző erőforrásokat és tevékenységeket kezelhetnek.

A screenshot of a computer

AI-generated content may be incorrect.

**Irányítópult:**

Az admin felület központi eleme egy statisztikai irányítópult, amely valós idejű áttekintést nyújt a szálloda működéséről:

* Foglalt szobák aktuális száma
* Aznapi érkezések száma
* Aznapi éttermi foglalások száma
* Havi bevétel összesítése

**Szobakezelés:**

A szobákkal kapcsolatos műveletek:

* Szobák listázása: Az összes szoba megtekintése (azonosító, név, szobaszám, állapot, ár)
* Új szoba hozzáadása: Lehetőség új szoba létrehozására (szobaszám, szobatípus, ár megadásával)
* Szoba szerkesztése: Meglévő szobák adatainak módosítása (típus, ár, állapot)
* Szoba törlése: Szobák eltávolítása a rendszerből

**Foglaláskezelés**

A szállodai foglalások adminisztrációja:

* Foglalások listázása: Az összes foglalás megtekintése
* Foglalások kezelése: Lehetőség a foglalások állapotának módosítására

**Éttermi menükezelés**

Az éttermi kínálat adminisztrációja:

* Menü listázása: Az étlap összes elemének megtekintése
* Új étel hozzáadása: Lehetőség új étel felvételére (név, leírás, ár, típus)
* Étel szerkesztése: Meglévő ételek adatainak módosítása
* Étel törlése: Ételek eltávolítása az étlapról

**Éttermi foglalások kezelése**

Az éttermi foglalások adminisztrációja:

* Foglalások listázása: Az összes éttermi foglalás megtekintése
* Foglalás szerkesztése: Lehetőség a foglalások adatainak módosítására (név, telefonszám, dátum, időpont, vendégek száma, speciális kérések)
* Foglalás törlése: Éttermi foglalások törlése

**Hibaüzenetek és visszajelzések**

A rendszer minden művelet során informatív visszajelzést ad:

* Sikeres műveletek megerősítése
* Hibaüzenetek megjelenítése probléma esetén (pl. "Hiba a szobák lekérdezésekor", "Hiba az étel hozzáadásakor")

Az admin felület magyar nyelvű, így a hibaüzenetek és a visszajelzések is magyarul jelennek meg, ami megkönnyíti a hazai felhasználók számára a rendszer használatát.

## Célközönség

A szállásfoglalási weboldal fejlesztésével egy olyan platformot hozunk létre, amely minden felhasználói csoport igényeit kielégíti, legyen szó akár turistákról, üzleti utazókról vagy szállásadókról. A felhasználók egyszerűen, gyorsan és biztonságosan tudják foglalni az igényeiknek leginkább megfelelő szálláshelyeket, míg a szállásadók egy átlátható és könnyen kezelhető adminisztrációs felületen keresztül tudják kezelni foglalásaikat, naprakészen tarthatják kínálatukat, és közvetlen kapcsolatban maradhatnak ügyfeleikkel.

A szállodai kezelő weboldal és rendszer elsődleges célközönsége a modern, digitális megoldásokat előnyben részesítő vendégek és szállodaüzemeltetők széles köre. A platformot úgy terveztük, hogy különböző felhasználói igényeknek megfeleljen, így több specifikus célcsoportot is kiszolgál

A rendszer kiemelten ügyel arra, hogy minden vendégtípus számára megfelelő élményt nyújtson – a gazdaságos standard szobáktól kezdve, a kényelmet kereső üzleti utazók igényeit kielégítő business szobákon át egészen a prémium felszereltségű penthouse lakosztályokig. Minden szobatípus saját szolgáltatásokat és egyedi kényelmi funkciókat kínál.

# Architektúra

## Frontend

A frontend egy React-alapú webalkalmazás, amely egy szállodai menedzsment rendszer teljes körű felhasználói felületét biztosítja. Célja, hogy a vendégek egyszerűen foglalhassanak szobát vagy éttermi asztalt, kapcsolatba léphessenek a szállodával, míg az adminisztrátorok hatékonyan kezelhessék a rendszer adatait, például a szobák állapotát, az étlapot és a foglalásokat. A kommunikáció a backenddel Axios könyvtár segítségével történik, JSON formátumú kérésekkel, a http://localhost:3000 vagy http://localhost:3001 címen, hitelesítéshez pedig sütiket használ (withCredentials: true). A vizuális megjelenést Bootstrap keretrendszer és egyedi CSS fájlok (pl. variables.css, style.css, AdminDashboard.css) támogatják, fekete (#040303), arany (#C5B358) és fehér színpalettával, ami luxus és elegancia érzetét kelti.

Az **AdminDashboard** az adminisztrátorok központi felülete, ahol egy oldalsáv segítségével navigálhatnak az áttekintés, szobakezelés, étlapkezelés, szobafoglalások és éttermi foglalások szekciók között. Az áttekintés statisztikákat mutat (foglalt szobák, mai érkezések, havi bevétel), míg a többi szekcióban táblázatokban listázódnak az adatok, amelyeket űrlapokkal és modális ablakokkal lehet szerkeszteni vagy törölni. A BookTable oldalon a vendégek éttermi asztalt foglalhatnak egy átlátható űrlap kitöltésével, amely név, telefonszám, dátum, időpont, vendégek száma és különleges kérések megadását kéri, majd a foglalást a /api/restaurant/bookings végpontra küldi. Hasonlóan, a Booking oldal a szobafoglalást szolgálja, itt a vendégek szobát választhatnak a lekérdezett listából, megadhatják adataikat (név, email, telefonszám, dátumok stb.), és a foglalás a /api/bookings végpontra kerül.

A **Contact** oldal kétoszlopos elrendezésben mutatja a szálloda elérhetőségeit (cím, telefon, email) és egy üzenetküldő űrlapot, amely a /api/contact végponttal kommunikál. Az Index a főoldal, ahol egy videós hero szekció fogadja a látogatókat, majd kiemelt szobák (API: /api/rooms/featured), az étterem bemutatása és egy egyszerű kapcsolati űrlap következik. A Login és Register oldalak a bejelentkezést és regisztrációt kezelik, sötét kártyás dizájnnal és arany gombokkal, az autentikációt a /api/login és /api/register végpontokra küldött kérésekkel végezve. A Dashboard a bejelentkezett felhasználók felülete, ahol profiladataikat, aktív foglalásaikat, pontjaikat, jutalmaikat (beváltás: /api/rewards/redeem) és kuponjaikat láthatják, utóbbiakat vonalkódokkal (JsBarcode). A Restaurant oldal az éttermet mutatja be részletesen, étlappal (API: /api/restaurant/menu), nyitvatartással és asztalfoglalási gombbal.

Technikailag a React Router biztosítja az oldalak közötti navigációt, a JsBarcode a kuponok vonalkódjait generálja, az adatok pedig aszinkron API-hívásokkal töltődnek be. A dizájn reszponzív, a Bootstrap grid rendszerének köszönhetően mobilbarát, az átmeneti effektek és arany kiemelések pedig modern, felhasználóbarát élményt nyújtanak. Az űrlapok egyértelműek, hibajelzésekkel és visszajelzésekkel segítik a használatot, így az alkalmazás egyszerre funkcionális és esztétikus.

A kezdőlap, amelyet az Index.js fájlban definiált **Index** komponens valósít meg, a Hotel Aurum Vista webalkalmazás nyitóoldala. Ez egy React-alapú komponens, amelynek célja a szálloda bemutatása, a vendégek vonzása és a főbb funkcionalitások (szobafoglalás, éttermi foglalás, kapcsolatfelvétel) gyors elérhetőségének biztosítása. A felület modern, vizuálisan vonzó dizájnt követ, Bootstrap CSS keretrendszerrel reszponzív elrendezést kínál, és az alkalmazás globális stílusaival (variables.css, style.css) integrálódik, fekete (#040303), arany (#C5B358) és fehér színpalettát használva.

A kezdőlap három fő szekcióra oszlik: egy hero szekcióra, egy szobák szekcióra és egy kombinált étterem-kapcsolat szekcióra. Az első, **hero szekció** egy látványos, teljes képernyős videós hátteret (hero.mp4) tartalmaz, amely automatikusan lejátszódik, némítva és ciklikusan fut. Ez a vizuális elem a luxus és modernitás érzetét kelti, a konténer pedig egy üres flex elrendezést biztosít, amely később foglalási űrlappal bővíthető. A videó alatt a tartalom dinamikusan nem változik, statikusan a figyelemfelkeltést szolgálja.

A második, **szobák szekció** a kiemelt szobákat mutatja be, amelyeket az useEffect hook segítségével aszinkron módon kér le a /api/rooms/featured végpontról az Axios könyvtár használatával. A kapott adatok (szoba neve, leírása, ára, képe) kártyákban jelennek meg egy háromoszlopos Bootstrap gridben. Minden kártya tartalmaz egy képet (room.image\_path), a szoba nevét, leírását és árát (room.formattedPrice), valamint egy "Foglalás" gombot, amely a /book oldalra navigál. A szekció alján egy "További szobáink" gomb a /rooms oldalra vezet. Ha a lekérdezés sikertelen, a hibát a konzol rögzíti, de a felhasználói felületen nem jelenik meg explicit hibaüzenet.

A harmadik szekció két alrészt ölel fel: az **étterem szekciót**. Az étterem rész bal oldalon szöveges bemutatást kínál ("Aurum Vista Étterem") egy rövid leírással és egy "Asztalfoglalás" gombbal (/book-table), jobb oldalon pedig egy statikus képet (restaurant.jpg) mutat a hangulat illusztrálására.

Technikailag a komponens a React useState és useEffect hookjait használja az állapotkezeléshez és az adatok betöltéséhez. A navigációt a React Router Link komponensei helyett itt egyszerű HTML a elemekkel oldották meg, ami a jövőben egységesíthető. A Bootstrap biztosítja a reszponzivitást, a videós háttér és a képek pedig vizuális fókuszpontokat teremtenek. Funkcionálisan a kezdőlap részben statikus (hero, étterem, kapcsolat), részben dinamikus (szobák), az API-integráció csak a szobákra korlátozódik. A dizájn elegáns, a felhasználói élményt a gyors navigáció és a vizuális elemek fokozzák, bár az űrlapok interaktivitása és a hibakezelés továbbfejleszthető.

**Munkamegosztás:**

**Péter Kristóf**:  
Kristóf a projekt adminisztratív és funkcionálisabb felületeire fókuszált, amelyek komplexebb logikát és kezelést igényeltek.

* Felelős az AdminDashboard.js és AdminDashboard.css fejlesztéséért, amely egy átfogó adminisztrációs panelt biztosít szobák, étlapok, foglalások és éttermi foglalások kezelésére.
* Kialakította a modal ablakokat, táblázatok dinamikus megjelenítését és az adminisztrációs űrlapok logikáját.
* A Dashboard.js-t is ő készítette, amely a felhasználói dashboardot jeleníti meg aktív foglalásokkal, pontokkal, kuponokkal és vonalkód-generálással.
* A CSS stílusok finomhangolása az admin felületen, például a sidebar, nav-linkek és dashboard-kártyák dizájnja.

**Jakab Zsombor**:  
Zsombor a vendégek számára látható, publikus felületek fejlesztésére koncentrált, amelyek a vizuális élményre és az egyszerű használatra épülnek.

* Fejlesztette a Booking.js, BookTable.js, Contact.js, Index.js, Restaurant.js, Login.js és Register.js komponenseket, amelyek a szobafoglalást, asztalfoglalást, kapcsolatfelvételt, főoldalt, éttermi oldalt, bejelentkezést és regisztrációt fedik le.
* Bootstrap integrációval és reszponzív dizájnnal biztosította a publikus oldalak esztétikus megjelenését.
* A hero section, szobakártyák és étlap vizuális elemeinek kialakítása az ő feladata volt.
* Az űrlapok alapvető stílusait és interaktív gombjait (pl. btn-gold) ő implementálta a publikus felületeken.

## Backend

**Szerver konfiguráció:**

* **Keretrendszer:** Express.js: Kezeli az útválasztást, middleware-eket és a HTTP kérés/válasz életciklust.

**Hitelesítés:**

* Munkamenet-alapú: Az express-session-t használja a bejelentkezett felhasználók nyomon követésére.

**Middleware**:

* checkAuth: Ellenőrzi, hogy a felhasználó be van-e jelentkezve (req.session.user létezik).
* checkAdmin: Ellenőrzi, hogy a felhasználó Adminisztrátor szintű-e.

Jelszóvédelem: A jelszavakat a bcryptjs titkosítja.

**Kontrollerek:**

A backend külön kontroller modulokba van szervezve:

1. **roomController:** Szobalisták kezelése (kiemelt és összes szoba).
2. **authController**: Felhasználói regisztráció és bejelentkezés kezelése.
3. **restaurantController**: Éttermi menü lekérdezése.
4. **wellnessController**: Wellness szolgáltatások részletei.
5. **dashboardController**: Felhasználói műszerfal adatok kiszolgálása (védett).
6. **bookingController**: Szobafoglalások kezelése.
7. **restaurantBookingController**: Éttermi asztalfoglalások kezelése.
8. **shopController**: Jutalomlisták és beváltás kezelése.
9. **adminController**: Adminisztratív műveletek (pl. szobák, menü, foglalások kezelése).

A **roomController** a szálloda szobáival foglalkozik. Kezeli a kiemelt szobák és az összes szoba lekérdezését, így a vendégek láthatják a rendelkezésre álló lehetőségeket. Egyszerűen az adatbázisból gyűjti az adatokat, és JSON formátumban adja vissza őket, hitelesítés nélkül.

Az **authController** a felhasználók regisztrációját és bejelentkezését intézi. A regisztráció során titkosítja a jelszavakat, és menti az adatokat az adatbázisba, míg a bejelentkezésnél ellenőrzi a hitelesítő adatokat, majd munkamenetet hoz létre a sikeres belépéshez. Ez a kontroller kulcsfontosságú a rendszer biztonságos működéséhez.

A **restaurantController** az éttermi menüt kezeli. Egyetlen feladata, hogy lekérje a menüpontokat az adatbázisból, és visszaadja azokat a vendégeknek, akik az éttermi kínálatot szeretnék megismerni.

A **wellnessController** a wellness szolgáltatásokkal kapcsolatos információkat szolgáltatja. Bár részletek nem állnak rendelkezésre, valószínűleg egy egyszerű lekérdezést végez, és a szolgáltatások listáját adja vissza a felhasználóknak.

A **dashboardController** a bejelentkezett felhasználók számára mutat személyre szabott adatokat, például foglalásokat vagy pontokat. Ez egy védett terület, ahová csak hitelesített felhasználók léphetnek be.

A **bookingController** a szobafoglalásokért felel. Segít az elérhető szobák listázásában, és új foglalásokat hoz létre az adatbázisban, frissítve a szoba állapotát, például foglaltra állítva azt.

A **restaurantBookingController** az éttermi asztalfoglalásokat intézi. Lehetővé teszi a vendégek számára, hogy asztalt foglaljanak, és az adatokat az erre kijelölt táblában tárolja.

A **shopController** a jutalomrendszert működteti. Megmutatja az elérhető jutalmakat, és kezeli a pontokkal való beváltásukat, új kuponokat generálva a felhasználóknak, akiknek elegendő pontjuk van.

Végül az **adminController** az adminisztrátorok eszköze. Lehetővé teszi a rendszer teljes körű kezelését, például szobák, menüpontok, foglalások hozzáadását, módosítását vagy törlését. Ez a kontroller csak az adminisztrátori jogosultsággal rendelkező felhasználók számára érhető el, és átfogó ellenőrzést biztosít a szálloda működése felett.

**Modellek**:

**db.js**  
Az adatbázis-kapcsolatot állítja be (localhost, root felhasználó, szalloda adatbázis, 3306 port), és inicializálja a kapcsolatot a lekérdezésekhez.

**bookings.js (createBooking)**  
Szobafoglalást hoz létre: ellenőrzi az elérhető szobát, frissíti annak státuszát Occupied-re, majd rögzíti a foglalást tranzakcióval (commit/rollback).

**bookings.js (getActiveBookingsForUser, getBookingHistoryForUser)**  
Lekéri egy felhasználó aktív (Active) és múltbéli (Completed) foglalásait a bookings és rooms táblákból, szobanevekkel együtt.

**restaurantBookings.js**  
Éttermi foglalást rögzít a restaurant\_bookings táblában egyszerű beszúrással.

**rooms.js**  
Szobák adatait kezeli: lekéri az összes szobát, egy adott szobát ID alapján, vagy az elérhető szobákat, árakat formázva (pl. 10 000 Ft).

**menu.js**  
Az éttermi menüt (restaurant\_menu) lekéri az adatbázisból.

**shop.js**  
A boltban elérhető jutalmakat (shop tábla) listázza ki.

**coupons.js**  
Kuponokat kezel: generál egyedi kódot, új kupont ad hozzá, lekéri a felhasználó aktív kuponjait, és frissíti a felhasználói pontokat levonással.

**Biztonság**:

A backend biztonsága több rétegre épül, hogy a rendszer védett legyen a jogosulatlan hozzáférésektől és az adatkezelés biztonságos maradjon. A hitelesítés munkamenet-alapú, amit az express-session middleware kezel, így a felhasználók bejelentkezés után egy munkamenet-azonosítón keresztül azonosíthatók. A jelszavakat a bcryptjs könyvtár titkosítja, ami erős hash-elést biztosít, így az adatbázisban tárolt jelszavak nem olvashatók ki könnyen. Az útvonalak védelmét két middleware segíti: a checkAuth ellenőrzi, hogy a felhasználó be van-e jelentkezve, míg a checkAdmin csak az adminisztrátori jogosultságú felhasználókat engedi tovább, például a rendszer kezelésére szolgáló végpontokhoz. A CORS beállítása korlátozza a kéréseket egy meghatározott eredetre, jelen esetben a http://localhost:3001-re, és hitelesítő adatokkal működik, ami csökkenti a külső támadások esélyét. A sütik jelenleg nem biztonságosak a fejlesztési környezetben, de éles rendszerben HTTPS-szel együtt biztonságossá tehetők. Összességében a rendszer alapvető védelmet nyújt, de a titkos munkamenet-kulcs erősítése és a bemenetek validálása tovább növelhetné a biztonságot.

**Munkamegosztás:**

**Péter Kristóf**:  
Kristóf az adminisztrációs funkciókhoz szükséges backend logikára összpontosított, amely a rendszer kezelését és monitorozását támogatja.

* Implementálta az admin API-kat: /api/admin/dashboard, /api/admin/rooms, /api/admin/menu, /api/admin/bookings, /api/admin/restaurant-bookings, amelyek CRUD műveleteket végeznek a szobák, étlapok és foglalások kezelésére.
* Kidolgozta az admin jogosultság-ellenőrzést (checkAdminStatus) és a hitelesítési mechanizmusokat (withCredentials: true).
* A statisztikai adatok (pl. foglalt szobák, havi bevétel) kiszolgálásáért is ő felelt az admin dashboardhoz.

**Jakab Zsombor**:  
Zsombor a vendégek által használt funkcionalitások backend támogatását végezte, amely a publikus interakciókat szolgálja ki.

* Fejlesztette a publikus API-kat: /api/bookings, /api/restaurant/bookings, /api/contact, /api/rooms/featured, /api/restaurant/menu, /api/register, /api/login, amelyek a foglalásokat, kapcsolatfelvételt, kiemelt szobákat, étlapot, regisztrációt és bejelentkezést kezelik.
* A felhasználói regisztráció és bejelentkezés logikáját, valamint az űrlapok adatvalidációját ő implementálta.
* Az éttermi és szobafoglalási rendszerek adatfeldolgozását és a válaszüzenetek kezelését is ő biztosította.

## Adatbázis

**A screen shot of a computer

AI-generated content may be incorrect.Kapcsolódás az adatbázishoz:**

A program egy "szalloda" nevű MySQL adatbázist használ, amely a következő módon csatlakozik az adatbázishoz:

Kapcsolódási adatok:

* Szerver: localhost
* Felhasználó: root
* Jelszó: (nincs beállítva)
* Adatbázis neve: szalloda
* Port: 3307

**Adatbázis általános leírása:**

Az adatbázis a szállodai foglalási rendszer alapvető pillére, amelyet úgy alakítottunk ki, hogy támogassa a vendégek szobafoglalásait, éttermi szolgáltatásait, valamint a hűségprogramot. A MySQL rendszeren fut, UTF-8 karakterkódolással (utf8mb4\_hungarian\_ci kollációval), így tökéletesen kezeli a magyar nyelvű adatokat. A phpMyAdmin felületén keresztül könnyen adminisztrálható, és egy teljes körű szállodai szolgáltatás informatikai hátterét biztosítja, a szobák kezelésétől az éttermi kínálaton át a vendégélmény növeléséig.

**A screenshot of a computer

AI-generated content may be incorrect.Táblák:**

**Táblák leírása:**

1. Felhasználók (users):

A tábla a szálloda vendégeinek és törzsvásárlóinak adatait tárolja, lehetővé téve a regisztrációt, a tagsági szintek kezelését és a hűségpontok nyilvántartását. Ez a rendszer alapja a személyre szabott szolgáltatásokhoz, például a pontgyűjtéshez vagy a foglalások összekapcsolásához a vendégekkel.

2. Szobák (rooms):

Ez a tábla a szálloda különböző szobatípusait (pl. Superior, Deluxe, Luxus) részletezi, tárolva azok nevét, árát, leírását és felszereltségét. A szobák bemutatására szolgál, segítve a vendégeket a választásban, miközben az árak és a kényelmi funkciók átláthatóan kezelhetők.

3. Szobák állapota (rooms\_data):

A konkrét szobák (pl. 101-es szoba) adatait és aktuális állapotát rögzíti, például hogy foglalt, takarítás alatt van-e, vagy szabad. Ez a tábla kulcsfontosságú a foglalási rendszer valós idejű működéséhez, biztosítva, hogy mindig naprakész információ álljon rendelkezésre.

4. Foglalások (bookings):

A szobafoglalások teljes körű kezelésére szolgál, beleértve a be- és kijelentkezési dátumokat, fizetési információkat és speciális kéréseket. A tábla összeköti a vendégeket a konkrét szobákkal, és nyomon követi a foglalások állapotát a foglalástól a lezárásig.

5. Bolt (shop):

A hűségprogram részeként a vendégek pontjaikért beváltható jutalmakat kínál, például ingyenes vacsorát vagy kedvezményeket. Ez a tábla ösztönzi a visszatérő vendégeket, és egyedi élményeket biztosít a pontok felhasználásával.

6. Kuponok (coupons):

A beváltott jutalmakhoz kapcsolódó kuponokat kezeli, nyilvántartva, hogy melyik vendég milyen jutalmat kapott, és azt felhasználta-e már. Ez a tábla a hűségprogram gyakorlati megvalósítását támogatja, összekötve a pontokat a konkrét kedvezményekkel.

7. Éttermi foglalások (restaurant\_bookings):

Az éttermi asztalfoglalások adatait rögzíti, lehetővé téve a vendégek számára, hogy előre lefoglalják helyüket, akár szállóvendégek, akár külsősök. A tábla rugalmasan kezeli az időpontokat és a speciális igényeket, javítva az éttermi szolgáltatások hatékonyságát.

8. Éttermi menü (restaurant\_menu):

Az étterem kínálatát tartalmazza, kategóriákra bontva (előételek, főételek, desszertek, italok), részletes leírásokkal és árakkal. Ez a tábla az éttermi szolgáltatások alapja, átláthatóvá téve a vendégek számára a választékot, miközben az árképzés is könnyen kezelhető.

A screenshot of a computer

AI-generated content may be incorrect.

**Kapcsolatok**

A bookings tábla a users és rooms\_data táblákhoz kapcsolódik, így a foglalások a vendégekhez és a konkrét szobákhoz köthetők.

A rooms\_data a rooms táblához kapcsolódik, hogy a szobák típusa és állapota összehangolható legyen.

A coupons tábla a users és shop táblákhoz kötődik, biztosítva a hűségprogram integritását.

Az adatbázis külső kulcsokkal garantálja az adatok konzisztenciáját.

Az éttermi foglalások (restaurant\_bookings) függetlenek a felhasználóktól, így külsős vendégek is foglalhatnak.

Az étlap (restaurant\_menu) önállóan működik, nem kapcsolódik közvetlenül más táblákhoz.

# Biztonság

## Adatvédelem

Az adatbázisban lévő adatok védelmét korszerű biztonsági megoldásokkal biztosítjuk. A jelszavakat titkosított formában tároljuk. Valamintadatbázishoz való hozzáférés szigorúan korlátozott és védett.

**Technikai megoldások**

* Az adatokat SQL adatbázisban tároljuk, optimalizált indexeléssel.
* A szobák kiegészítő adatai JSON formátumban kerülnek rögzítésre.
* Biztonságos adatbázis-kapcsolatot alkalmazunk.

## Felhasználói adatok védelme

A weboldalunkon történő foglalás során az adatminimalizálás elvét követve kizárólag azokat az adatokat gyűjtjük és tároljuk, amelyek elengedhetetlenek a foglalás lebonyolításához és a kapcsolattartáshoz.

A foglalási folyamat során rendszerünk az alábbi adatokat kéri be:

* Teljes név
* E-mail-cím
* Telefonszám
* A foglalás időpontja és időtartama
* Fizetési adatok (pl. bankkártya-információk – kizárólag a tranzakció idejére, biztonságos fizetési környezetben)

Ezen adatokat kizárólag a foglalás sikeres végrehajtásához és az ügyféllel való kapcsolattartáshoz használjuk fel. Az adatokat biztonságos módon, a vonatkozó adatvédelmi előírásoknak megfelelően kezeljük és tároljuk, harmadik fél számára kizárólag jogszabályi kötelezettség esetén adjuk át.

# Tesztelés

## Frontend (Selinium teszt)

**Asztalfoglalás Teszt Leírás**

Ez a teszt a szállodai kezelő weboldal éttermi asztalfoglalási funkcionalitását ellenőrzi automatizált módon a Selenium WebDriver segítségével. A teszt célja annak biztosítása, hogy a felhasználók sikeresen tudjanak asztalt foglalni az étterembe a megadott adatokkal, és a rendszer megfelelően kezelje a foglalási folyamatot.

Teszt lépései:

1. A teszt a következő lépéseket hajtja végre:
2. Weboldal megnyitása: Megnyitja a szálloda weboldalát a http://localhost:3001/ címen.
3. Ablakméret beállítása: A böngészőablak méretét 2576x1416 pixeles felbontásra állítja, hogy a felület teljes funkcionalitása látható legyen.
4. Asztalfoglalás menüpont kiválasztása: Rákattint az "Asztalfoglalás" linkre a navigációs menüben.
5. Űrlap kitöltése:

* Név: "Horváth Béla" kerül beírásra a megfelelő mezőbe.
* Telefonszám: "06306005364" kerül beírásra a megfelelő mezőbe.
* Dátum: "2025-04-18" kerül megadásra a dátum mezőben.
* Időpont: A legördülő menüből a "18:00" opció kerül kiválasztásra.
* Vendégek száma: A legördülő menüből a "3 fő" opció kerül kiválasztásra.
* Speciális kérések: "asd" kerül beírásra a speciális kérések mezőbe.

1. Foglalás elküldése: A foglalás véglegesítése érdekében a "Foglalás" gombra kattint a teszt.

Teszt célja: A teszt ellenőrzi, hogy az éttermi asztalfoglalási folyamat zökkenőmentesen működik-e a felhasználói felületen keresztül, és hogy a rendszer helyesen kezeli-e a bevitt adatokat, beleértve a szöveges mezőket, dátumokat és legördülő menüket. A teszt sikeres lefutása azt jelzi, hogy a vendégek képesek asztalt foglalni az étterembe a megadott paraméterekkel.

**Szobafoglalás Teszt Leírás**

Ez a teszt a szállodai kezelő weboldal szobafoglalási funkcionalitását ellenőrzi automatizált módon a Selenium WebDriver segítségével. A teszt célja annak biztosítása, hogy a felhasználók sikeresen tudjanak szobát foglalni a megadott adatokkal, és a rendszer megfelelően kezelje a foglalási folyamatot.

Teszt lépései:

1. A teszt a következő lépéseket hajtja végre:
2. Weboldal megnyitása: Megnyitja a szálloda weboldalát a http://localhost:3001/ címen.
3. Ablakméret beállítása: A böngészőablak méretét 2576x1416 pixeles felbontásra állítja, hogy a felület teljes funkcionalitása látható legyen.
4. Szobafoglalás menüpont kiválasztása: Rákattint a "Szobafoglalás" linkre a navigációs menüben.
5. Űrlap kitöltése:

* Név: "Horváth Béla" kerül beírásra a user\_name nevű mezőbe.
* Email: "[H.Bela@gmail.com](mailto:H.Bela@gmail.com)" kerül beírásra a user\_email nevű mezőbe.
* Telefonszám: "06306005364" kerül beírásra a phone\_number nevű mezőbe.
* Bejelentkezés dátuma: "2025-04-14" kerül megadásra a check\_in nevű mezőben.
* Kijelentkezés dátuma: "2025-04-18" kerül megadásra a check\_out nevű mezőben.
* Vendégek száma: A guests legördülő menüből a "3 fő" opció kerül kiválasztásra.
* Fizetési mód: A pay\_method legördülő menüből a "Banki átutalás" opció kerül kiválasztásra.
* Szoba kiválasztása: A room\_id legördülő menüből a "Luxus Suite - 54000 Ft/éj" opció kerül kiválasztásra.
* Speciális kérések: "asd" kerül beírásra a special\_requests nevű mezőbe.
* További szolgáltatások: "asd" kerül beírásra az additional\_services nevű mezőbe.

1. Foglalás elküldése: A foglalás véglegesítése érdekében a .btn-gold CSS osztállyal rendelkező "Foglalás" gombra kattint a teszt.

Teszt célj: A teszt ellenőrzi, hogy a szobafoglalási folyamat zökkenőmentesen működik-e a felhasználói felületen keresztül, és hogy a rendszer helyesen kezeli-e a bevitt adatokat, beleértve a szöveges mezőket, dátumokat, legördülő menüket és a foglalás elküldését. A teszt sikeres lefutása azt jelzi, hogy a vendégek a megadott paraméterekkel képesek szobát foglalni, és a weboldal front-end funkcionalitása megfelelően működik.

**Bejelentkezés Teszt Leírás**

Ez a teszt a szállodai kezelő weboldal bejelentkezési funkcionalitását ellenőrzi automatizált módon a Selenium WebDriver segítségével. A teszt célja annak ellenőrzése, hogy a regisztrált felhasználók sikeresen be tudnak-e jelentkezni a rendszerbe a megadott hitelesítő adatokkal.

Teszt lépései:

A teszt a következő lépéseket hajtja végre:

1. Weboldal megnyitása: Megnyitja a szálloda weboldalát a http://localhost:3001/ címen.
2. Ablakméret beállítása: A böngészőablak méretét 1674x1378 pixeles felbontásra állítja, hogy a felület teljes funkcionalitása látható legyen.
3. Bejelentkezés menüpont kiválasztása: Rákattint a navigációs sáv 5. elemére, amely feltehetően a "Bejelentkezés" gomb.
4. Email mező kitöltése: Rákattint az email beviteli mezőre, de a teszt itt megszakad a további adatok megadása előtt (a dokumentumban nem szerepel további lépés).

Teszt célja: A teszt célja annak ellenőrzése, hogy a bejelentkezési folyamat elindítható-e, és hogy a felhasználói felület lehetővé teszi-e az email mező elérését. A dokumentumban szereplő kód azonban nem teljes, így a teszt jelenleg csak a bejelentkezési oldal elérését és az email mező aktiválását ellenőrzi.

**Regisztráció Teszt Leírás:**

Ez a teszt a szállodai kezelő weboldal regisztrációs funkcionalitását ellenőrzi automatizált módon a Selenium WebDriver segítségével. A teszt célja annak biztosítása, hogy az új felhasználók sikeresen regisztrálhassanak a rendszerbe a megadott adatokkal, és a folyamat végén megfelelő visszajelzést kapjanak.

Teszt lépései:

1. A teszt a következő lépéseket hajtja végre:
2. Weboldal megnyitása: Megnyitja a szálloda weboldalát a http://localhost:3001/ címen.
3. Ablakméret beállítása: A böngészőablak méretét 1674x1378 pixeles felbontásra állítja.
4. Navigáció a regisztrációhoz: Rákattint a navigációs menü 5. elemére (feltehetően a bejelentkezés/regisztráció menüpont), majd a "Regisztrálj" linkre.
5. Űrlap kitöltése:

* Vezetéknév: "Horvath" kerül beírásra a megfelelő mezőbe.
* Keresztnév: "Béla" kerül beírásra a megfelelő mezőbe.
* Email: "[Horv.Bela@gmail.com](mailto:Horv.Bela@gmail.com)" kerül beírásra a megfelelő mezőbe.
* Telefonszám: "06306005364" kerül beírásra a megfelelő mezőbe.
* Jelszó: "Selinium01" kerül beírásra a jelszó mezőbe.
* Jelszó megerősítése: "Selinium01" kerül beírásra a jelszó megerősítése mezőbe.

1. Regisztráció elküldése: Rákattint a .btn-gold CSS osztállyal rendelkező "Regisztráció" gombra.
2. Visszajelzés ellenőrzése: Ellenőrzi, hogy megjelenik-e a "Sikeres regisztráció!" szövegű felugró ablak.

## Backend (Jest teszt)

Ez a teszt a szállodai kezelő weboldal backendjének autentikációs funkcióit vizsgálja a Jest keretrendszer és a supertest könyvtár segítségével. Az Auth Controller registerUser és loginUser végpontjait teszteli sikeres műveletek és hibák esetén. A cél a regisztráció és bejelentkezés helyes működésének ellenőrzése.

Tesztkörnyezet

Jest-ben fut, supertest szimulálja a HTTP kéréseket, a db és bcryptjs mockolva van. Minden teszt előtt a mockok alaphelyzetbe állnak (jest.clearAllMocks()).

registerUser Teszt

Ellenőrzi a regisztrációt. Sikeres esetben (érvényes adatok, nincs foglalt email) 201-es státusszal "Sikeres regisztráció!" üzenetet vár, és a db.query kétszer fut. Hibák: eltérő jelszavaknál (400, "A két jelszó nem egyezik meg!"), foglalt emailnél (400, "Ez az email cím már foglalt!"), rövid jelszónál (400, "A jelszónak legalább 6 karakter hosszúnak kell lennie!").

loginUser Teszt

Ellenőrzi a bejelentkezést. Sikeres esetben (helyes email és jelszó) 200-as státusszal "Sikeres bejelentkezés!" üzenetet és felhasználói adatot vár. Hibák: nem létező felhasználónál (401, "Hibás email vagy jelszó!"), rossz jelszónál (401, "Hibás email vagy jelszó!").

Technikai részletek

A db.query és bcrypt mockolva szimulálja az adatbázist és jelszókezelést. A supertest POST kéréseket küld az Express app ellen, a Jest expect ellenőrzi az eredményeket. Időkorlát nincs explicit, a Jest alapértelmezése érvényesül.

# Befejezés

## Fejlesztési lehetőségek

A projekt jelenlegi formájában egy működőképes, teljes értékű rendszert kínál, azonban számos lehetőség áll előttünk a további tökéletesítésre. Az egyik legfontosabb célunk a felhasználói visszajelzések alapján történő finomhangolás, amely biztosítja, hogy a platform még jobban igazodjon a vendégek és az üzemeltetők valós igényeihez. További fejlesztési irányok közé tartozik a mesterséges intelligencia integrálása, például egy chatbot formájában, amely valós idejű segítséget nyújthat a foglalások során, vagy személyre szabott ajánlatokat kínálhat a vendégek preferenciái alapján.

Emellett tervezzük a rendszer skálázhatóságának növelését, hogy az ne csak egyetlen szálloda, hanem egy teljes szállodalánc kezelésére is alkalmas legyen. Ez magában foglalhatja a többnyelvű támogatás bevezetését, hogy a platform nemzetközi vendégeket is hatékonyan kiszolgálhasson, valamint a fizetési lehetőségek bővítését, például kriptovaluták vagy mobilfizetési rendszerek integrálásával.

## Jövőbeli tervek

A következő nagy lépés egy dedikált mobilalkalmazás fejlesztése lesz, amely még kényelmesebbé teszi a vendégek számára a foglalások kezelését és a szolgáltatások igénybevételét. A mobilalkalmazás push-értesítésekkel figyelmeztetné a felhasználókat a foglalási állapotokról, különleges ajánlatokról vagy a pontjaik aktuális egyenlegéről. További tervek között szerepel egy analitikai modul bevezetése az adminisztrációs felületen, amely részletes statisztikákat és előrejelzéseket kínál a foglalási trendekről, segítve a szálloda stratégiai döntéseit.

A partnerhálózat bővítése szintén kiemelt prioritás. Célunk, hogy a hűségpontok beválthatósága még szélesebb körben legyen elérhető, például kulturális programokra, turisztikai attrakciókra vagy wellness-szolgáltatásokra. Ez nemcsak a vendégek számára nyújtana további értéket, hanem a szálloda vonzerejét is növelné a versenytársakkal szemben.

## Záró gondolatok

Az általunk létrehozott szállodai kezelő weboldal és rendszer nemcsak a mai kor elvárásainak felel meg, hanem egy jövőbiztos alapot is teremt a további innovációk számára. A technológiai háttér biztosítja a gyors működést, a skálázhatóságot és a folyamatos fejlődési lehetőségeket, amelyek a weboldalt versenyképessé és időtállóvá teszik. A felhasználói élmény maximalizálása érdekében kiemelten fontos számunkra a biztonságos adatkezelés, a modern fizetési megoldások integrálása, valamint a könnyen navigálható felhasználói felület.

Ezzel a weboldallal egy olyan dinamikus és sokoldalú platformot hoztunk létre, amely képes alkalmazkodni a változó piaci igényekhez, és hosszú távon is értéket nyújt a felhasználóknak. A projekt sikerét a csapat elkötelezettsége, a gondos tervezés és a vendégközpontú megközelítés biztosította. Büszkék vagyunk arra, hogy egy olyan rendszert alkottunk, amely hozzájárulhat a szállodaipar digitalizációjához, és egy új szintre emelheti a vendéglátás élményét.