

# Fejlesztői dokumentáció

Heaven hotel weboldal

## **Témavezetők:**

Szabó Ákos

Rehó János

## **Készítették:**

Osikóczy Sándor Mátyás

Soltész Mónika

Varga Tímea

NYÍREGYHÁZA  
2023

# Tartalomjegyzék

1.	Feladat megadása .....	3
1.1	Szereplők és igényeik .....	3
2.	Use-Case Funkció lista .....	4
2.1	Regisztrált felhasználó .....	4
2.2.	Látogató .....	4
2.3.	Adminisztrátor .....	4
3.	Magas szintű rendszerterv.....	5
4.	Screenek.....	6
5.	Modellek .....	9
6.	Alkalmazások kiválasztása .....	10
6.1.	Frontend.....	10
6.2.	Backend .....	11
6.3.	Adatbázis .....	11
6.4.	Tesztelés .....	11
7.	Routing .....	12
7.1	Authentikáció .....	12
7.2	Felhasználók .....	12
7.3	Foglalások.....	13
7.4.	Szobák .....	13
8.	Implementáció.....	15
9.	Tesztelés .....	16
10.	Bevezetés, éles üzemmód .....	18

# 1. FELADAT MEGADÁSA

Feladatunk egy alkalmazás készítése, amelyet több felhasználó fog használni. Online alkalmazásunkban a felhasználók szállodai szobákat foglalhatnak, amelyhez regisztrálniuk kell a felhasználónak. Regisztráció hiányban csak böngészni tud a felhasználó az oldalon.

Az alkalmazásnak fő feladata az, hogy a regisztrált felhasználók képesek legyenek szobát foglalni.

## 1.1 SZEREPLŐK ÉS IGÉNYEIK

### 1. *Bejelentkezett felhasználó*

- legyen képes szobát foglalni

### 2. *Látogató*

- tudjon böngészni
- legyen képes regisztrálni, majd bejelentkezni a fiókjába

### 3. *Adminisztrátor*

- regisztrált felhasználók adatainak elérése
- regisztrált felhasználók adatainak módosítása
- regisztrált felhasználók adatainak törlése
- szobák módosítása
- szobatípusok módosítása
- szobafoglalások törlése

## **2. USE-CASE FUNKCIÓ LISTA**

Az alkalmazásban a felhasználók két csoportra tudjuk osztani. Azokra, akik regisztráltak és azokra, akik nem. A két csoport között a különbség mindössze annyi, hogy a még nem regisztrált felhasználó nem tud szobát foglalni.

### **2.1 REGISZTRÁLT FELHASZNÁLÓ**

- böngészés
- szabad szobák kilistázása szobatípus, érkezés és távozás megadása után
- bejelentkezés
- kijelentkezés
- szoba lefoglalása

### **2.2. LÁTOGATÓ**

- böngészés
- szabad szobák kilistázása szobatípus, érkezés és távozás megadása után
- legyen képes regisztrálni (név, email, jelszó, telefonszám, lakcím)
- legyen képes bejelentkezni

A látogató és a regisztrált/bejelentkezett felhasználó is tud az oldalon a szobák és szolgáltatások között böngészni. Mindketten képesek szobatípus alapján szűrni a felvett szobákat.

A látogatónak, ha szobafoglalási szándéka van, előbb regisztrálnia kell, majd bejelentkezni a fiókjába, hogy a foglalási folyamatot megkezdhesse. A regisztrációkor meg kell adni a bizonyos adatokat.

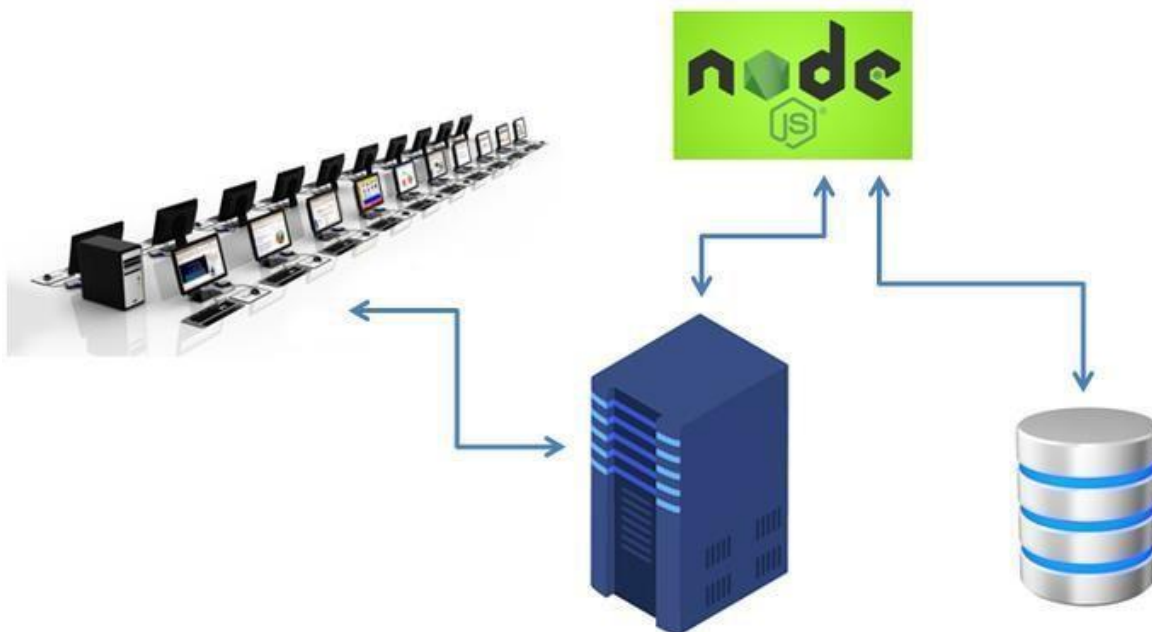
Regisztráció után a látogató bejelentkezhet. Bejelentkezés után foglalként kezeljük a látogatót.

Szobát foglalni csak a bejelentkezett felhasználó tud, mivel szükségesek a regisztráláskor megadott adatok.

### **2.3. ADMINISZTRÁTOR**

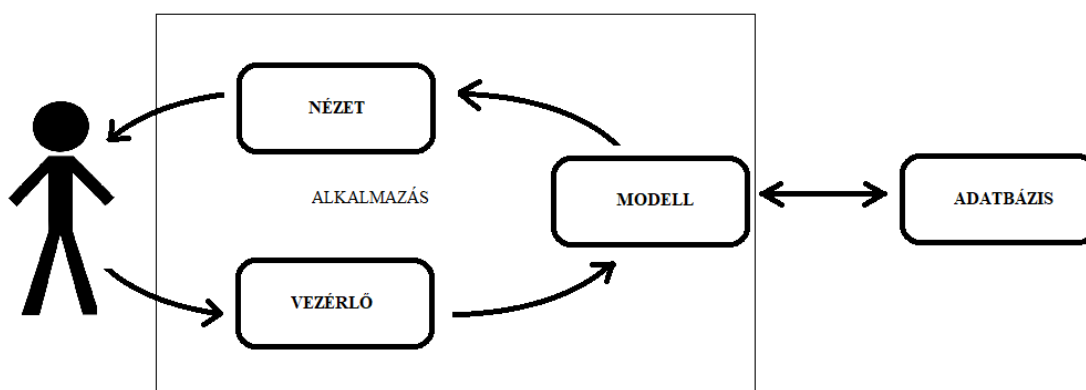
- a szobákat az adminisztrációs felületen lehet módosítani
- a szobákátípusokat az adminisztrációs felületen lehet módosítani
- az adminisztrációs felületet csak az adminisztrátor érheti el
- listázni tudja a regisztrált felhasználókat, illetve azokat törölni és módosítani.
- csak az adminisztrátor tud jogokat kiosztani

### 3. MAGAS SZINTŰ RENDSZERTERV



*Rendszerterv*

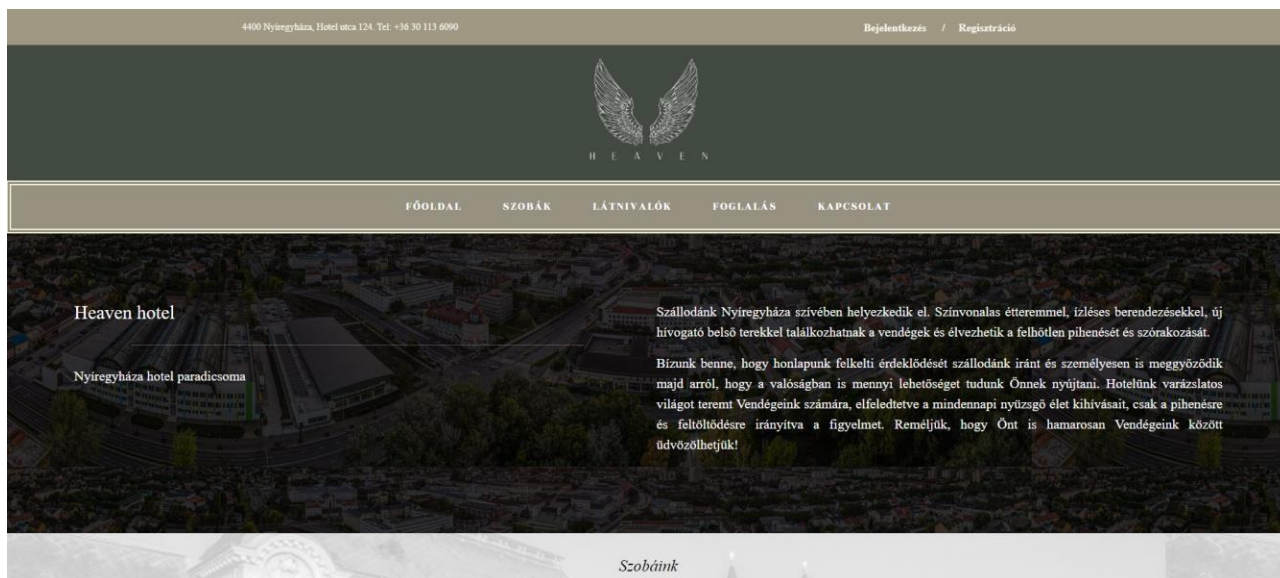
A fejlesztés során az MVC szemléletmód figyelembevételével igyekeztünk elkülöníteni az adatok elérését, az üzleti logikát, az adatok megjelenítését és a felhasználói interakciót. Ezzel biztosítjuk azt, hogy a felhasználói felület ne befolyásolja a kezelést, illetve az adatok a későbbi fejlesztések során, ha kell, könnyen átszervezhetőek legyenek a felhasználói felület változtatása nélkül.



*MVC modell*

## 4. SCREENEK

Egy internetes alkalmazás készítésekor fontos több szempontot is figyelembe venni. A végtermék olyan felhasználóknak készül, akikkel közvetlen kapcsolat nem áll fenn, így, ha a felhasználó úgy érzi nem kapott kellő információt az oldalon, akkor lehet nem fogja az általunk nyújtott szolgáltatást igénybe venni, valamint az oldalra sem fog visszatérni. Ennek elkerülése érdekében kiemelkedően fontos az oldal kinézetével és tartalmával kapcsolatban, hogy a látogatók bizalmát elnyerje, könnyen kezelhető és logikus felépítésű legyen. Minden információ egyértelmű legyen, például az elérhetőségek feltüntetése, a termékekről elég információ legyen feltüntetve, vagy például az árak ÁFÁ-val terheltek, vagy sem. Az ilyen alapvető adatok hiánya miatt a vásárló más boltot választhat.



### *Kezdőképnyő*

A használhatóság érdekében az információk és funkciók könnyen elérhetőek a felhasználók számára, biztosítjuk a logikus navigációt. Az elérhetőséget feltüntettük, a gyakori kérdések oldalon böngészve pedig a vendégek gyorsan választ kaphatnak a legfontosabb kérdésekre.

## LÁTNIVALÓK

### Aquarius Élményfürdő



Nyíregyháza-Sóstó a gyógyvizéről híres, hangulatos fürdőhely Nyíregyháza központjától alig pár kilométerre található. Az Aquarius Élményfürdő gyermekvilággal, stranddal, csúszdák széles választékával rendelkezik így az egész családnak szuper élményt kínál. A Sóstón található jódos-brómos gyógyvíz reumatikus, mozgásszervi és nőgyógyászati panaszok enyhítésére kiválóan



## Látnivalók

A legfontosabb látnivalókról a képeket a vendégek Bootstrap Carousel segítségével tudják megnézni, amelyen előző/következő vezérlők találhatóak, így tudnak váltogatni a képek között. Ezt a megoldást látványosnak és kellően áttekinthetőnek találtuk.

### Szobatípusok



#### superior egyágyas szoba

Kényelmes, 16 m<sup>2</sup> alapterületű egyágyas szoba 1 fő részére. A fürdőszobában ülökád található. Bekészítések: törölköző, hajszárító. A szoba ablakából a csodálatos nyíregyházi belváros látható. A szobában minibár is a vendégek rendelkezésére áll.

Ágyak: 1

Maximális létszám: 1

Szobaár: 17000 Ft. / Éjszakától



#### standard kétágyas szoba

Minden igényt kielégítő 18 m<sup>2</sup>-es beltér, két különálló ágygal vagy franciaágygal. A szobában minibár és LCD TV áll a vendégek rendelkezésére. A fürdőszobában kényelmes ülökád található. Bekészítések: törölköző, hajszárító.

Ágyak: 2

Maximális létszám: 2

Szobaár: 22000 Ft. / Éjszakától



#### superior kétágyas szoba

Kényelmes, tágas, 20 m<sup>2</sup> alapterületű kétágyas vagy franciaágyas szoba 2 fő részére. A fürdőszobában ülökád található. Bekészítések: törölköző, hajszárító. A szoba ablakából a csodálatos nyíregyházi belváros látható. A szobában minibár is a vendégek rendelkezésére áll.

Ágyak: 2

Maximális létszám: 2

Szobaár: 25000 Ft. / Éjszakától



## Szobatípusok megjelenítése



## Regisztráció

Név \*

Telefonszám \*

Ilyen formában adja meg: 06301234567

E-mail cím \*

Lakcím \*

Jelszó \*

min 8 karakter, kis- és nagybetű, szám, speciális karakter

☐ Elfogadom az [ÁSZF-et](#) és az [adatvédelmi tájékoztatót](#).

REGISZTRÁCIÓ

[Már van fiókja? Lépjen be itt!](#)

### Regisztrációs felület

Sikeres foglalás  Beérkező levelek x


**saleshotelheaven01@gmail.com**  
 címzett: én ▼

**Tisztelt [vargatimi505@gmail.com!](#)**

Köszönjük, hogy nálunk fogalt, melyet ezen levéllel visszaigazolunk.

**Foglalás adatai:**

- Érkezés: **2023. április 16.**
- Távozás: **2023. április 18.**
- Éjszakák száma: **2**
- Fizetendő összeg: **44000 Ft**

Tisztelettel  
Heaven Hotel csapata

---

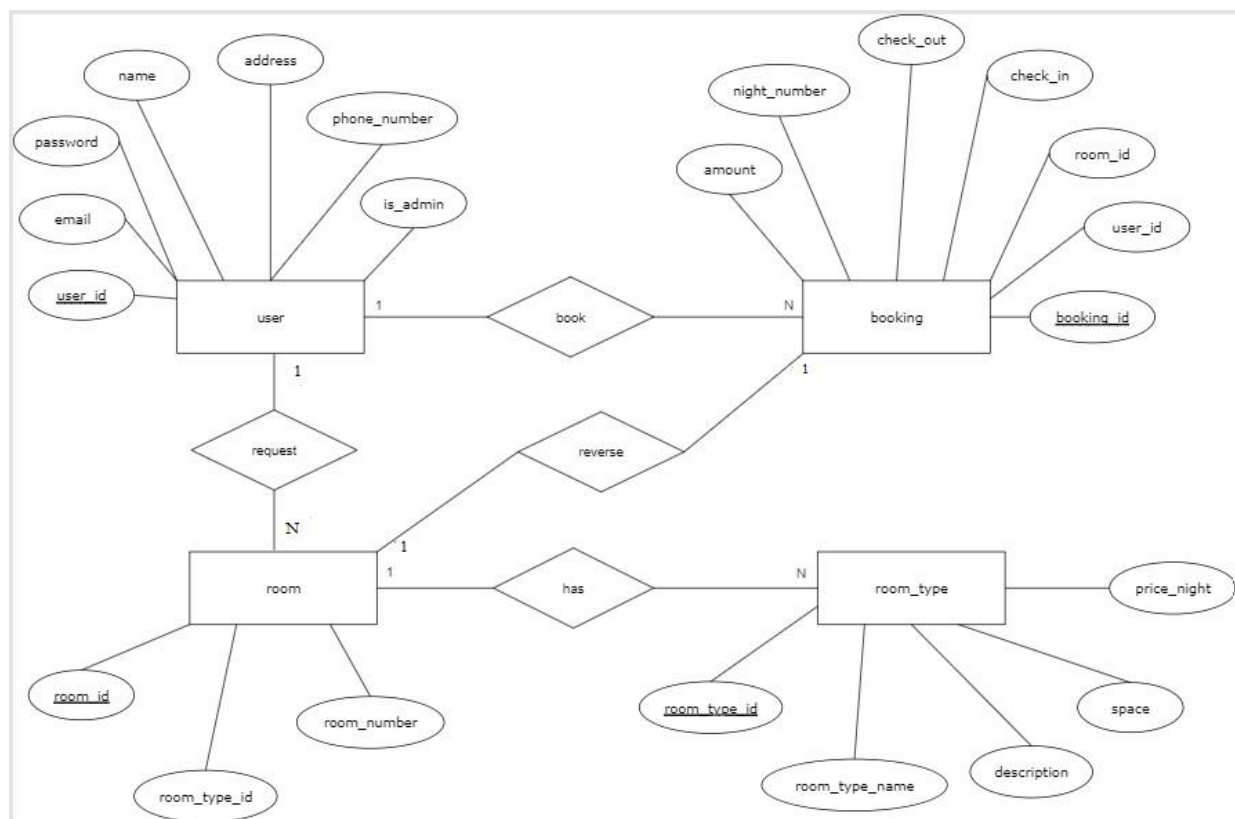
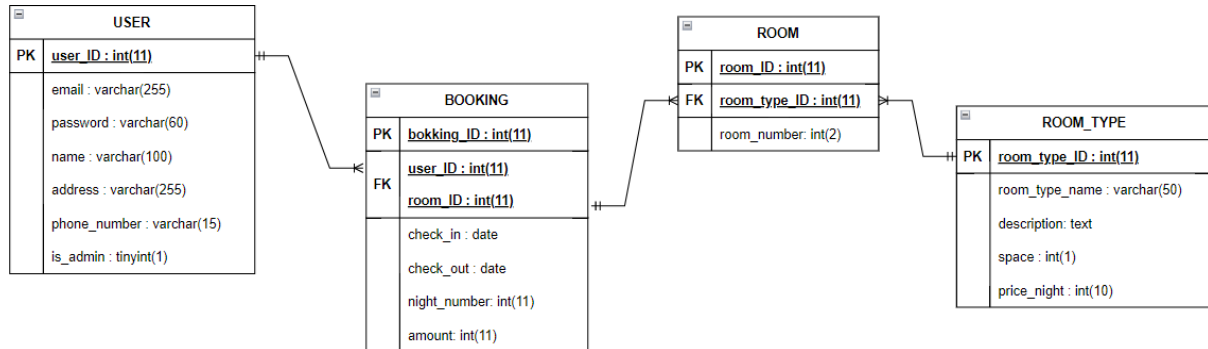
Ha bármilyen kérdése van, keresse recepciónkat az alábbi telefonszámon: +36-30-234-6421

Sikeres foglalást követően a felhasználók e-mailes visszaigazolást kapnak.



## 5. MODELLEK

Az adatok tárolására több modellre is szükség van, mint regisztrált felhasználók, foglalások, szobák, szobatípusok.



## 6. ALKALMAZÁSOK KIVÁLASZTÁSA

### 6.1. FRONTEND

- **React**

A React az egyik nyílt forráskódú JavaScript könyvtár. Interaktív felhasználói felületek felépítésére szolgál. Ez egy hatékony, deklaratív és rugalmas könyvtár. A Model-View-Controller (MVC) V, azaz View elemével foglalkozik. Ez nem egy teljes keret, hanem csak egy előlapi könyvtár. Lehetővé teszi összetett felhasználói felületek létrehozását vagy elkészítését, összetevőkként ismert, elkülönített és apró kóddarabok felhasználásával. (<https://hu.education-wiki.com/9050114-what-is-react>)

- **JavaScript**

A JavaScript (röviden JS) egy objektumalapú scriptnyelv, amelyet elterjedten használnak weboldalakon, a web programozási nyelve. A JavaScript futási ideje egyszálú, ami azt jelenti, hogy egyszerre csak egy darab kódot képes végrehajtani. Ennek a megoldására használunk async függvényeket.

- **HTML**

A HTML (Hypertext Markup Language) egy leíró nyelv, melyet weboldalak elkészítésére használhatunk.

- **Bootstrap**

A Bootstrap egy nyílt forráskódú keretrendszer (framework), mely **HTML, CSS, JavaScript technológiákat használ**. Nagyon könnyen, és minimális energia befektetéssel tudunk jól kinéző, bármilyen képernyőméreten szépen megjelenő weboldalakat készíteni a Bootstrap segítségével.

- **Material Ui**

A MUI vagy Material UI a felhasználói felület tervezési összetevőinek hatalmas könyvtára, amelyet a fejlesztők könnyen importálhatnak a React alkalmazások létrehozásához. Ez egy nyílt forráskódú projekt, amely követi a Google alkalmazás- vagy webhely-összetevők létrehozására vonatkozó irányelveit.

- **axios**

Az Axios egy könyvtár, amelyet HTTP -kérések küldésére használnak a böngészőből Node és Express.js platformon keresztül.

## 6.2. BACKEND

- **Node.js**

A Node.js egy nyílt forrású, többplatformos, szerver oldali JavaScript futtatókörnyezet a *Google Chrome V8 JavaScript*-motoron alapulva. Leginkább webes applikációk, dinamikus weboldalak fejlesztésére használják, de akár szerveroldali feldolgozó scripteket készítésére is kiválóan alkalmas.

- **express**

Az **Express** (amit gyakran **Express.js**-nek, vagy **ExpressJS**-nek is hívnak) egy szerver oldali (back end) applikáció fejlesztői keretrendszer Node.js-hez. Használata és terjesztése ingyenes, szabad szoftver az MIT licensze alatt. Elsősorban webes applikációkat és API-kat fejleszthetünk vele, és rendkívül sikeres életútja során viszonylag hamar az általános webes applikációs keretrendszeré vált Node.js alatti fejlesztések esetében. (<http://maas.hu/learning/it/prog/javascript/express/>)

- **bcryptjs** – hashelés (jelszó)

A BcryptJS egy népszerű és megbízható módszer a jelszavak kivonására.

- **JSON Web Token** – autentikáció

A JWT vagy JSON Web Token egy nyílt szabvány, amelyet két fél – egy ügyfél és egy kiszolgáló – közötti biztonsági információk megosztására használnak.

- **nodemon** – no restart

A nodemon egy olyan eszköz, amely segít a node.js alapú alkalmazások fejlesztésében azáltal, hogy automatikusan újraindítja a csomópont-alkalmazást, ha a rendszer fájl változásokat észlel a könyvtárban.

- **MySQL** – node module

A MySQL egy rendkívül népszerű *relációs adatbázis-kezelő rendszer*. A MySQL-t gyakran használják adatbázisként a Node-dal épített projektekhez és alkalmazásokhoz.

## 6.3. ADATBÁZIS

- **Xampp**

Egy platformfüggetlen webservert-szoftvercsomag, azok számára, akik adatbázissal támogatott webservert szeretnének üzemeltetni, a webes alkalmazások készítését, tesztelését és futtatását oldja meg.

- **MariaDB**

A MariaDB Server az egyik legnépszerűbb nyílt forráskódú relációs adatbázis.

## 6.4. TESZTELÉS

- **Postman**

A Postman egy komplett eszköztár API fejlesztők számára. Olyan eszköz, amelyet az API-k megbízható teszteléséhez használhatunk. A Postman leegyszerűsíti az API életciklusának minden lépését, és egyszerűsíti az együttműködést.

## 7. ROUTING

### 7.1 AUTHENTIKÁCIÓ

#### POST auth/login

tervezett feladat	authentikáció
bemenet	felhasználónév, jelszó
kimenet	felhasználónév és egy jogosultságot meghatározó érték

#### POST auth/register

tervezett feladat	authentikáció
bemenet	email, jelszó, név, lakcím, telefonszám
kimenet	felhasználónév és egy jogosultságot meghatározó érték

### 7.2 FELHASZNÁLÓK

#### GET user/users

tervezett feladat	összes felhasználó kislistázása
bemenet	
kimenet	felhasználók listája, adataik

#### GET user/users/:id

tervezett feladat	adott felhasználó adatainak kislistázása
bemenet	paraméterként kapott azonosító
kimenet	adott felhasználó adatai

#### PUT user/users

tervezett feladat	felhasználó módosítása
bemenet	felhasználó adatai a módosítottokkal
kimenet	visszajelzés a művelet sikerességéről

#### DELETE user/users/:id

tervezett feladat	töröl egy felhasználót azonosító alapján
bemenet	paraméterként kapott azonosító
kimenet	visszajelzés a művelet sikerességéről

## 7.3 FOGLALÁSOK

### GET reservation/reservations

tervezett feladat	összes foglalás kislistázása
bemenet	
kimenet	foglalások listája, adataik

### POST reservation/reservations

tervezett feladat	szoba foglalása
bemenet	a felhasználó és foglalás adatai
kimenet	amennyiben sikeres a foglalás emailben visszaigazolja

### POST reservation/reservations/availablerooms

tervezett feladat	elérhető szabad szobák szűrése
bemenet	szoba típusa, kijelentkezés/ bejelentkezés dátuma
kimenet	szabad szobák listája

### DELETE reservation/reservations/:id

tervezett feladat	töröl egy foglalást azonosító alapján
bemenet	paraméterként kapott azonosító
kimenet	visszajelzés a művelet sikerességéről

## 7.4. SZOBÁK

### GET room/roomtypes

tervezett feladat	összes szobatípus kilistázása
bemenet	
kimenet	szobatípusok lista

### GET room/roomtypes/:id

tervezett feladat	adott szobatípus adatainak kilistázása
bemenet	paraméterként kapott azonosító
kimenet	adott szobatípus adatai

### **PUT room/roomtypes/:id**

tervezett feladat	adott szobatípus módosítása
bemenet	paraméterként kapott azonosító, szobatípus adatai a módosítottokkal
kimenet	visszajelzés a művelet sikerességéről

### **GET room/rooms**

tervezett feladat	összes szoba kilistázása
bemenet	
kimenet	szobák lista

### **GET room/rooms/:id**

tervezett feladat	Adott szoba adatainak lekérése
bemenet	paraméterként kapott azonosító
kimenet	adott szoba adatai

### **PUT room/rooms/:id**

tervezett feladat	adott szoba módosítása
bemenet	paraméterként kapott azonosító szoba adatai a módosítottokkal
kimenet	visszajelzés a művelet sikerességéről

## 8. IMPLEMENTÁCIÓ

Az implementáció a tervezést követő folyamat. A tervezés során megtörténik a feladatokra bontás, valamint az egyes feladatokban érintett funkcionalitás és szerkezet modellezése. Az implementáció során a modellekből készítünk működő, futtatható kódot.  
(<https://okt.sed.hu/rf2/gyakorlat/tananyag/implementacio/implementacio/>)

## 9. TESZTELÉS

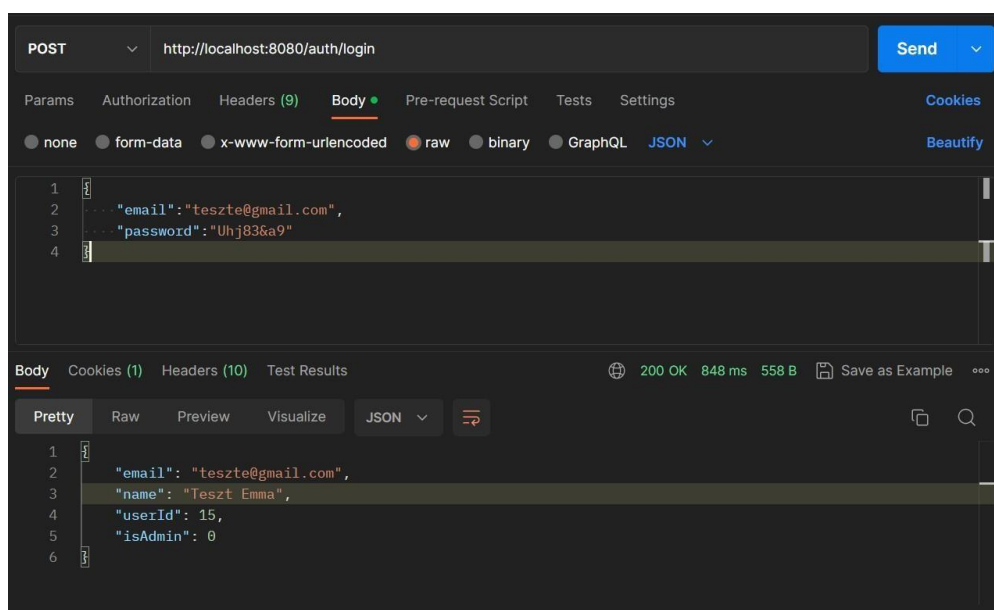
A felhasználó szempontjából a rendszer futtatásához elegendő bármely olyan operációs rendszer, amelyen futtatható HTML, CSS és JavaScript technológiával készült alkalmazás megjelenítésére képes böngésző.

A fejlesztés alatt a rendszer tesztelése desktop eszközön böngészőben (Google Chrome 100.0.4896.127) Windows 10 Enterprise operációs rendszer alatt történt. Továbbá mobil eszközön Android operációs rendszer alatt is teszteltük.

A szerver oldalról nézve a rendszernek szüksége van egy olyan platformra, ami MySQL szervereket képes futtatni, a fejlesztés során egy helyi localhost-tal oldottuk meg.

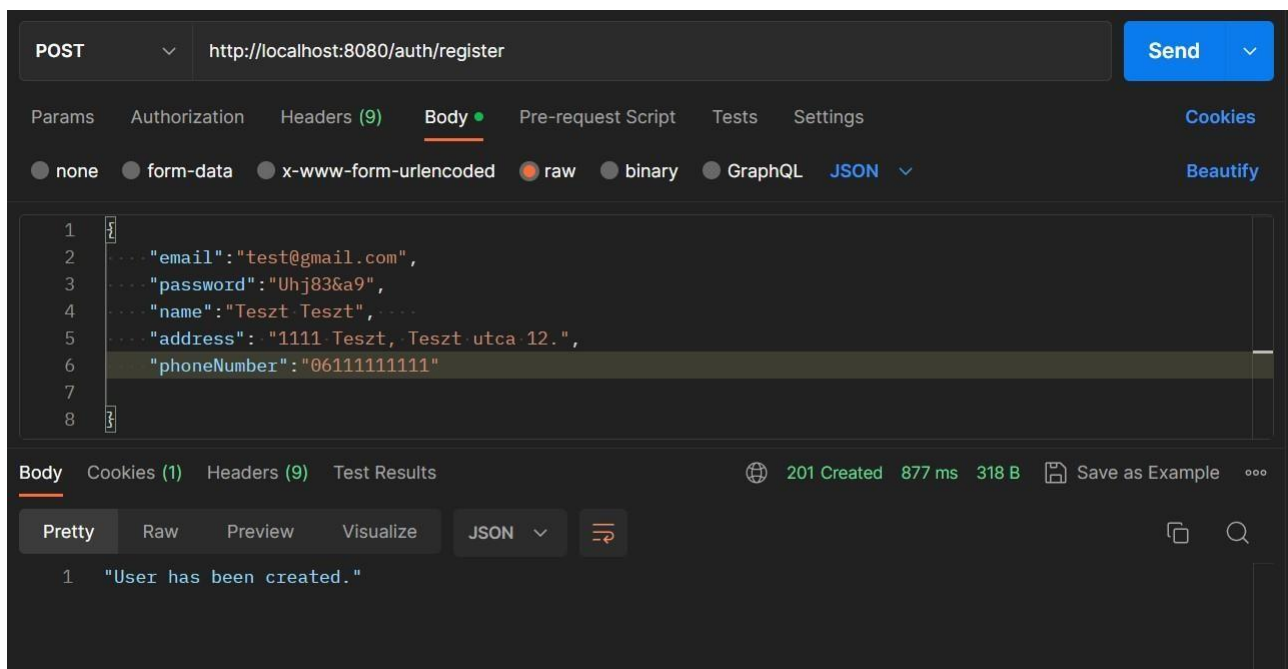
A feladat végrehajtása során folyamatosan jelen volt a tesztelés. Már a statikus oldal megtervezésekor és készülésekor tesztelve volt a megjelenés responsivitása, az olvashatóság, a láthatóság.

Az adatbázis tartalommal feltöltése után az oldal tartalmi része is tesztelve, ellenőrizve volt. Ezt Postman-ban valósítottuk meg, amely azonnali visszajelzést adott a fejlesztés sikerességéről, esetleges hibákról.



*Login autentikáció tesztelése Postman-ben*





### *Regisztráció tesztelése Postman-ben*

A fejlesztés során JEST unit tesztek is készültek, melyekkel a Kapcsolat komponens lett tesztelve.

```
import { screen, render } from "@testing-library/react";
import { Kapcsolat } from "../pages/Kapcsolat";

test("test1", () => {
  render(<Kapcsolat/>);

  const buttonElement = screen.queryByRole('button', { innerText: "Küldés" });
  expect(buttonElement).toBeInTheDocument();
});

test("test2", () => {
  render(<Kapcsolat/>);

  const buttonElement = screen.queryByText(/Név/i);
  expect(buttonElement).toBeInTheDocument();
});

test("test3", () => {
  render(<Kapcsolat/>);

  const h5Element = screen.queryByText(/Elérhetőségeink/i);
  expect(h5Element).toBeInTheDocument();
});

test("test4", () => {
  render(<Kapcsolat/>);

  const hElements = screen.queryAllByRole("heading");
  expect(hElements).toHaveLength(3);
});

test("test5", () => {
  render(<Kapcsolat/>);

  const h3Element = screen.queryByRole("heading", { level: 3 });
  expect(h3Element).toBeInTheDocument();
});
```

## 10. BEVEZETÉS, ÉLES ÜZEMMÓD

- Az elkészült munka megtekintéséhez szükséges a XAMPP letöltése. Látogassunk el a következő oldalra: <https://www.apachefriends.org/hu/download.html>, és onnan töltsük le az operációs rendszerünknek megfelelő XAMPP-verziót.
- Telepítsük fel. A letöltött futtatható állományt indítsuk el, és kövessük a képernyőn megjelenő utasításokat. Windows alatt a javasolt telepítési könyvtár `c:\xampp`.
- töltsse le a Windows telepítőt a Node.js webhelyről. Miután kiválasztotta az igényeinek megfelelő verziót, futtassa a telepítőt.
- Indítunk kettő Visual Studio Code alkalmazást, egyikben a client mappát nyitjuk meg, a másikban a backend mappát kell megnyitni az Open Folder menüponttal.
- mindkét Visual Studio Code-ban indítsunk egy Terminált a Terminal menu/New Terminal-lal. (Vagy a Command Line is megfelel)
- Sikeres telepítések után a xampp-ot indítsuk el és az Apache-t és MySQL-t is Startoljuk, és a MySQL Admin gombját válasszuk.
- A phpMyAdmin-ba importáljuk be a kapott adatbázist. (backend/databases)
- Mindkét Visual Studio Code-ban a Terminal-ban `npm -i` paranccsal telepítsük az összes modult.
- Mindkét Visual Studio Code-ban `npm start` paranccsal indítsuk az alkalmazást.

## Összegzés

Az általunk fejlesztett REST Api alkalmazás minden olyan CRUD funkciót kihasználva, amely szükséges volt a projekt megvalósítása során, mutatja be a kliens és szerver oldal közötti http kommunikációt.

A fejlesztés során fontos szempont volt, hogy a jogkörök legyenek kezelve, adminisztrátor és regisztrált felhasználó, valamint látogató jogosultságainak elkülönítése az alkalmazás különböző rétegeihez.

Az elkészített alkalmazás nem tökéletes, további javításokra és fejlesztésekre szorul.

Sajnálatos módon néhány funkció a fejlesztési idő rövideje miatt nem kerülhetett bele a szoftverbe. Ilyen hiányosságok például a pénzüsszegre való szűrés a szabad szobák keresésénél.

A teljes alkalmazás Git repository-ja:

<https://github.com/timi15/HeavenHotel>