

# Fejlesztői dokumentáció

**Témavezető:**

Rehó János

**Készítette:**

Molnár Áron

Pelles Martin

NYÍREGYHÁZA

2024

# 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 .....	12
7.	Routing .....	13
7.1	Authentikáció .....	13
7.2	Felhasználók .....	13
7.3	Foglalások .....	14
7.4.	Szobák .....	14
8.	Implementáció .....	16
9.	Tesztelés .....	17
10.	Bevezetés, éles üzemmód .....	19
11.	Összegzés .....	20

# 1. FELADAT MEGADÁSA

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

Az alkalmazás fő feladata, 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 felhasználókat 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 bizonyos adatokat.

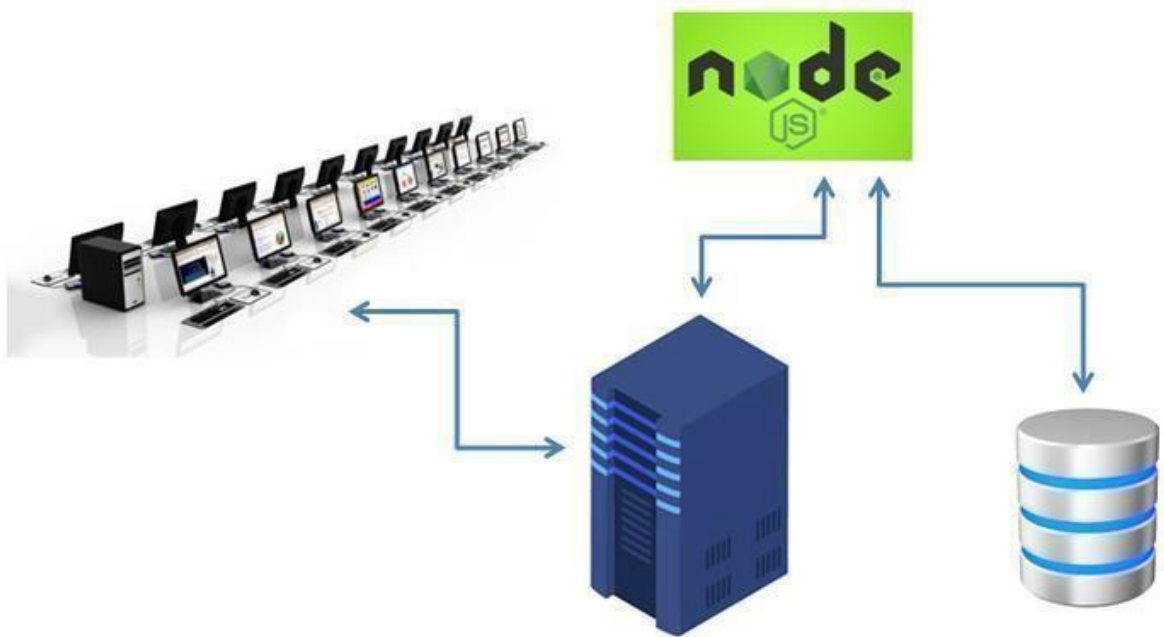
Regisztráció után a látogató bejelentkezhet. Bejelentkezés után foglalként kezeljük a látogatót.

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

### 2.3. ADMINISZTRÁTOR

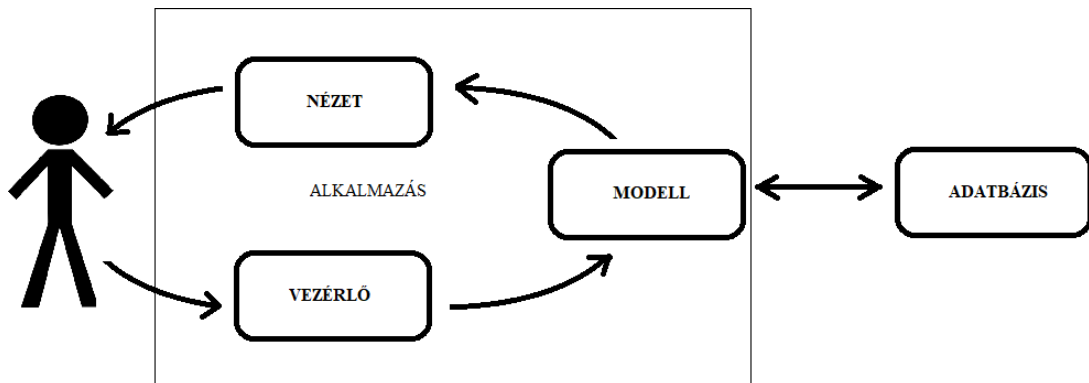
- a szobákat az adminisztrációs felületen lehet módosítani
- a szobaképeket az adminisztrációs felületen lehet módosítani
- az adminisztrációs felületet csak az adminisztrátor érheti el
- listázni tudja a regisztrált felhasználókat, illetve azokat törölni és módosítani.
- csak az adminisztrátor tud jogokat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

### 3. MAGAS SZINTŰ RENDSZERTERV



*Rendszerterv*

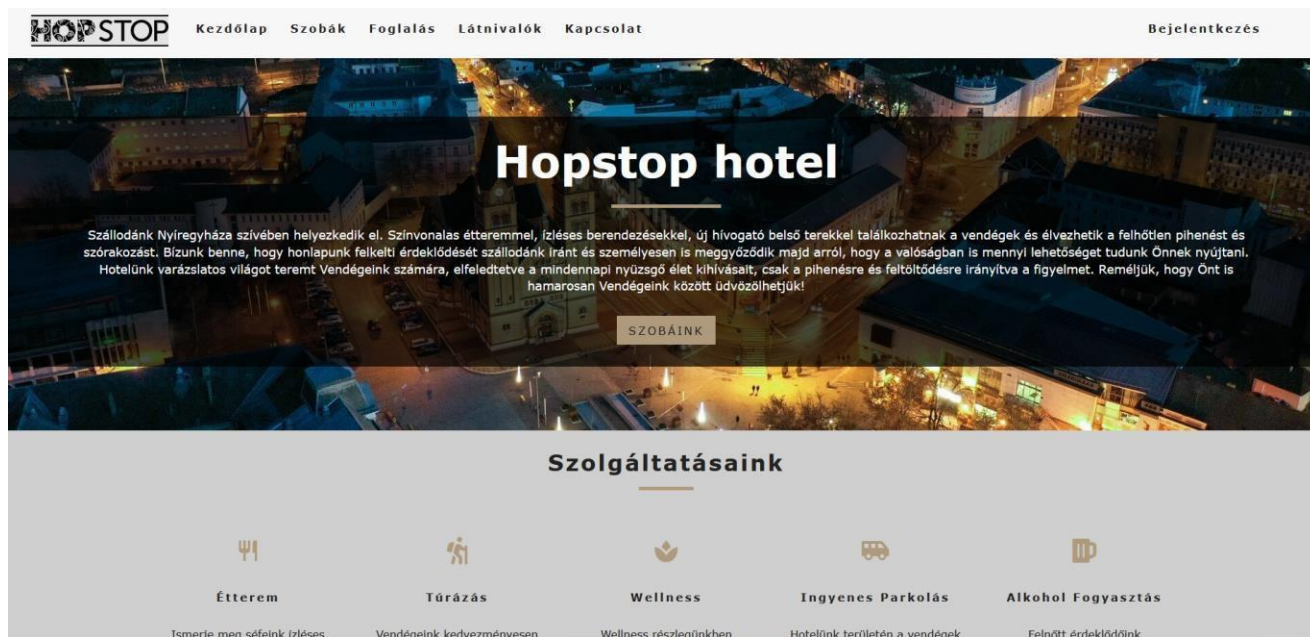
A fejlesztés során az MVC szemléletmód figyelembevételével igyekeztünk elkülöníteni az adatok elérését, az üzleti logikát, az adatok megjelenítését és a felhasználói interakciót. Ezzel biztosítjuk azt, hogy a felhasználói felület ne befolyásolja a kezelést, illetve az adatok a későbbi fejlesztések során,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



*Kezdőképernyő*

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 kérdéseikre.



## Aquarius Élményszobák



Nyíregyháza-Sóstó a gyógyvizéről híres, hangulatos fürdőhely Nyíregyháza központjától alig pár kilométerre található. Az Aquarius Élményszobák gyermekvilággal, stranddal, csúszdák széles választékával is rendelkezik így az egész családnak szuper élményt kínál. A Sóstón található jódos-brómos gyógyvíz reumatikus, mozgásszervi és nőgyógyászati panaszok enyhítésére kiválóan alkalmazható, ezért évente több ezer turista keresi fel ezt az üdülőhelyet. A felnőttek és idősek a Termálfürdő szolgáltatásait is igénybe vehetik, itt beltéri és kültéri gyógyvízes medencék találhatók nyakzuhannyal, pezsgőággal, hátmasszírozóval.

## Látnivalók

A legfontosabb látnivalók fényképeit a vendégek egy kényelmes vezérlő segítségével tudják nézegetni amely automatikusan de akár manuálisan is léptethetőek az alsó gombok segítségével, így tudnak váltogatni a képek között. Ezt a megoldást látványosnak, egyértelműnek és kellően áttekinthetőnek találtuk.

## Szobák Keresése

Szoba Típusa

Összes

Vendégek Száma

1

Szoba Ára 5499 Ft.



Szoba Mérete (m<sup>2</sup>)

0 200

☐ Reggeli

☐ Háziállatok



## Szobatípusok megjelenítése





## Regisztráció

Ilyen formában adja meg: 06301234567

Például: 1135 Budapest, Aranyosi utca 12

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

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

REGISZTRÁCIÓ

[Már van fiókja? Lépjen be itt!](#)

### Regisztrációs felület

## Sikeres foglalás Beérkező levelek x



hopstophotel@gmail.com

címzett: én ▼

Tisztelt [molnar.aron1201@gmail.com](mailto:molnar.aron1201@gmail.com)!

Köszönjük, hogy nálunk foglalt, melyet ezen levéllel visszaigazolunk.

#### Foglalás adatai:

- Érkezés: 2024. március 25.
- Távozás: 2024. március 27.
- Éjszakák száma: 2
- Fizetendő összeg: 6998 Ft

Tisztelettel:  
Hopstop hotel csapata

---

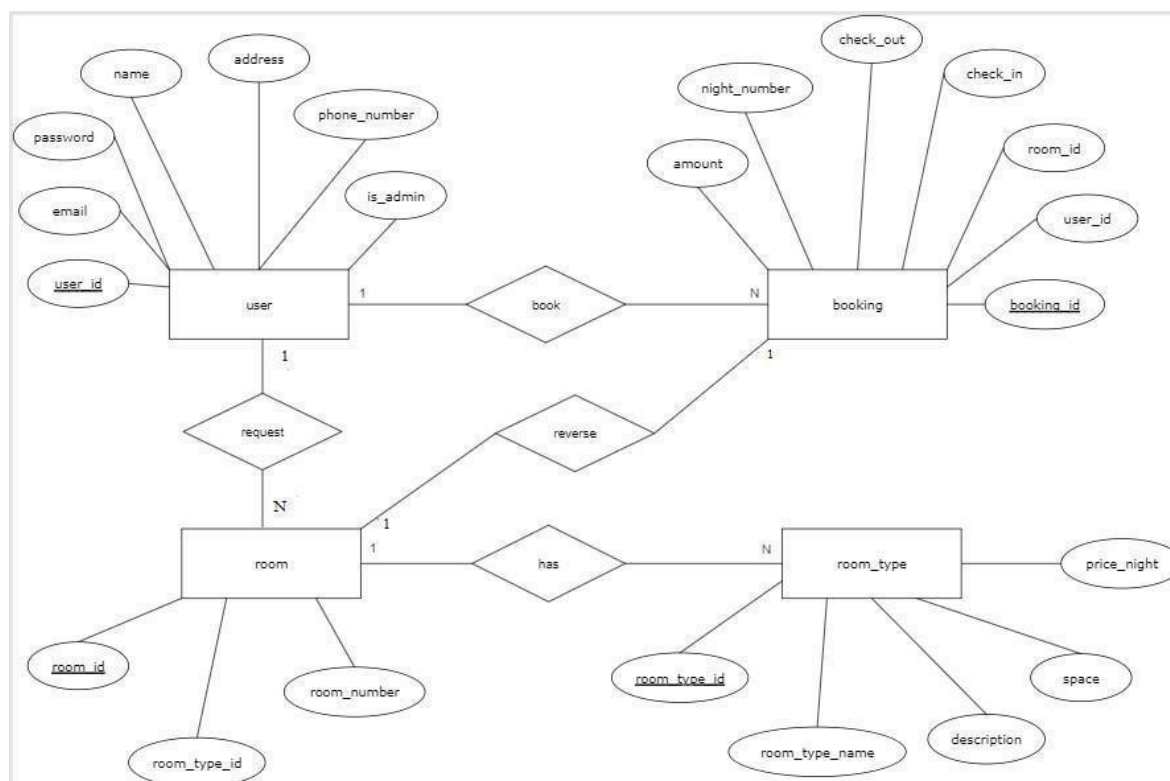
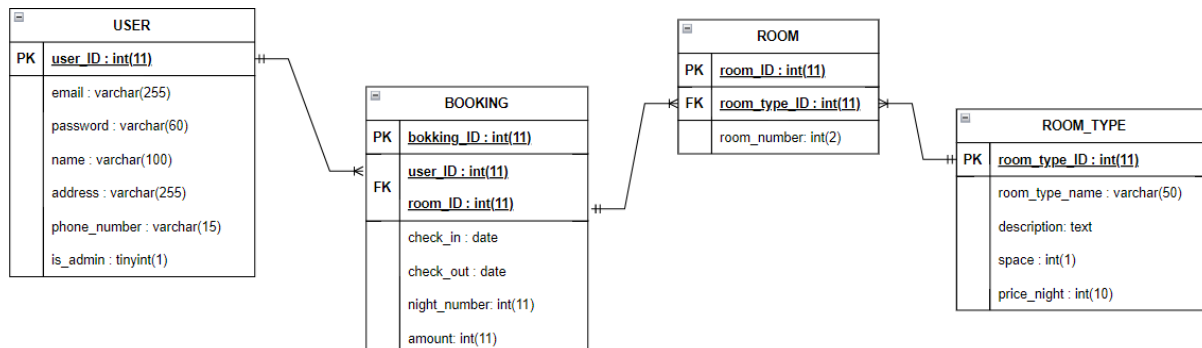
Ha bármilyen kérdése van, keresse recepciónkat az alábbi telefonszámon: +36-20-118-6170

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**

Egy nyílt forráskódú JavaScript könyvtár felhasználói felületek létrehozására, amely az egyoldalas alkalmazások fejlesztése során felmerülő weboldalak tartalmának részleges frissítési problémáinak megoldására szolgál.

- **JavaScript**

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

- **HTML5**

Az HTML (Hypertext Markup Language) egy olyan kódolási nyelv, amelyet a weboldalak létrehozására használnak. Az HTML segítségével strukturált tartalmat és formázást adhatunk a weboldalaknak, beleértve a szöveget, képeket, videókat, hivatkozásokat, űrlapokat és sok más elemet. Az HTML az internet alapvető nyelve, és az összes weboldal felépítésének alapja. Az HTML-t böngészők értelmezik, hogy a tartalom megjelenjen a felhasználóknak a weboldalon. A HTML5 a HTML ötödik verziója, amely az előző verziókhoz képest számos új funkcióval és lehetőséggel rendelkezik, amelyek lehetővé teszik a weboldalak sokkal dinamikusabb és interaktívabb kialakítását.

- **CSS3**

Az CSS3 (Cascading Style Sheets 3) egy stíuslap nyelv, amelyet weboldalak és webalkalmazások kinézetének és elrendezésének meghatározására használnak. Az első verziója, az CSS1, 1996-ban jelent meg, majd az CSS2 1998-ban. Az CSS3 az előző verziók fejlesztése, amelynek számos új funkciója és lehetősége van, mint például a jobb képkezelés, animációk, árnyékok, átlátszóság, új betűtípusok és hátterek. Az CSS3 hozzájárul a modern webfejlesztéshez, és segít a weboldalak szép és dinamikus megjelenítésében.

- **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 egy Node.js alapú keretrendszer, amely segít a webalkalmazások és API-k készítésében. Az Express egyszerű és könnyen használható, és számos olyan funkcióval rendelkezik, amelyek megkönnyítik a fejlesztést, mint például az útvonalak kezelése, a middleware-ek használata, a sablonmotorok integrációja és a HTTP kérés és válasz kezelése. Az Express rugalmas és könnyen testre szabható, így a fejlesztők nagyon sokféleképpen használhatják, attól függően, hogy milyen típusú alkalmazást szeretnének készíteni.

- **bcryptjs**

A BcryptJS egynépszerű és megbízható módszer a jelszavak kivonására.

- **JSON Web Token**

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

- **Nodemon**

A Nodemon egy olyan szoftver, amelyet a Node.js platformhoz fejlesztettek ki, és a fejlesztési folyamatok során használható. A Nodemon figyeli az alkalmazás forrásfájljait, és amikor változtatást észlel bennük, automatikusan újra indítja az alkalmazást, így nem kell manuálisan újra elindítani minden egyes módosítás után.

- **MySQL**

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

- **concurrently**

A "concurrently" npm csomag egy parancssor eszköz, amely lehetővé teszi a több parancs egyidejű futtatását egyetlen parancssoros ablakban. Ez különösen hasznos a fejlesztés során, amikor például szeretnénk egyszerre futtatni több folyamatot, mint például egy fejlesztői szerver indítása és a forráskód figyelése a változásokra. Ez teszi lehetővé projektünk során a frontend és a backend szerver egyetlen parancsban való elindítását.

## 6.3. ADATBÁZIS

- **Xampp**

Egy platformfüggetlen webszerver-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 kilistázása
bemenet	
kimenet	felhasználók listája, adataik

#### GET user/users/:id

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

#### PUT user/users

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

#### DELETE user/users/:id

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

## 7.3 FOGLALÁSOK

### **GET reservation/reservations**

tervezett feladat	összes foglalás kilistázása
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 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.

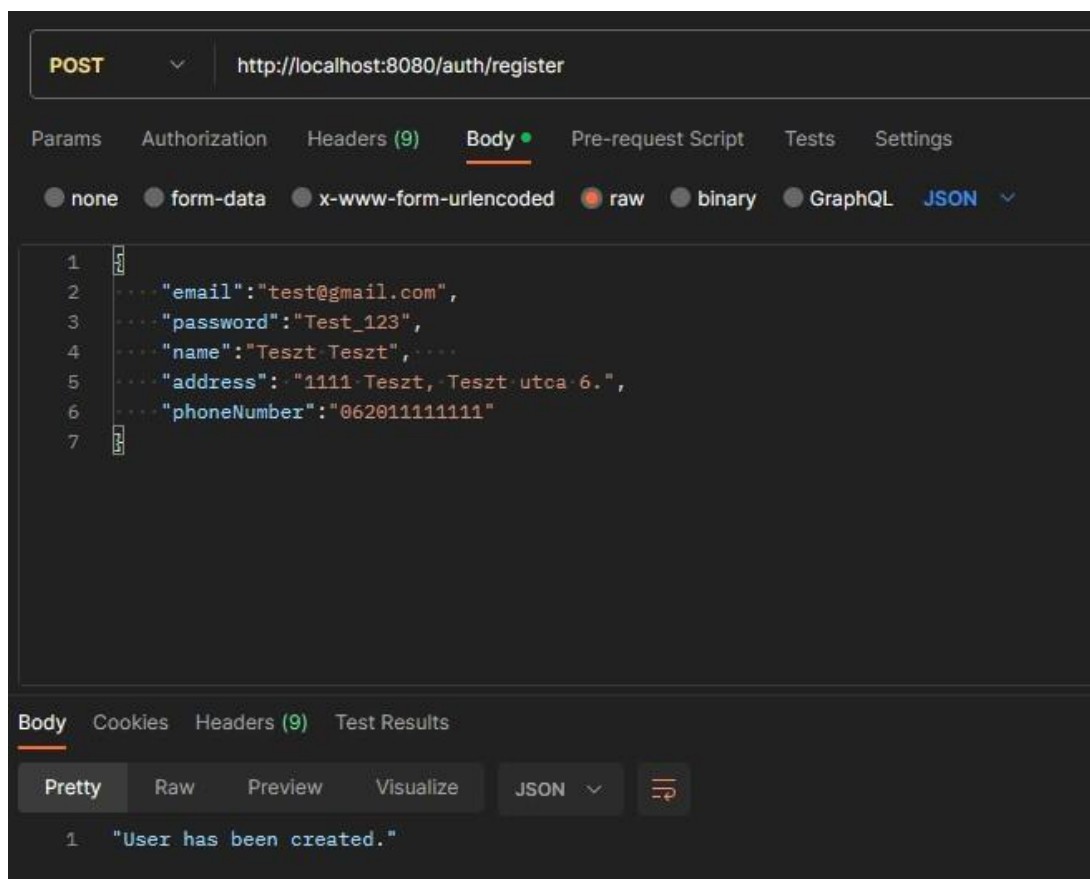
## 9. TESZTELÉS

A felhasználó szempontjából az alkalmazás használatához olyan eszköz szükséges, amely képes futtatni a mai modern böngészőket.

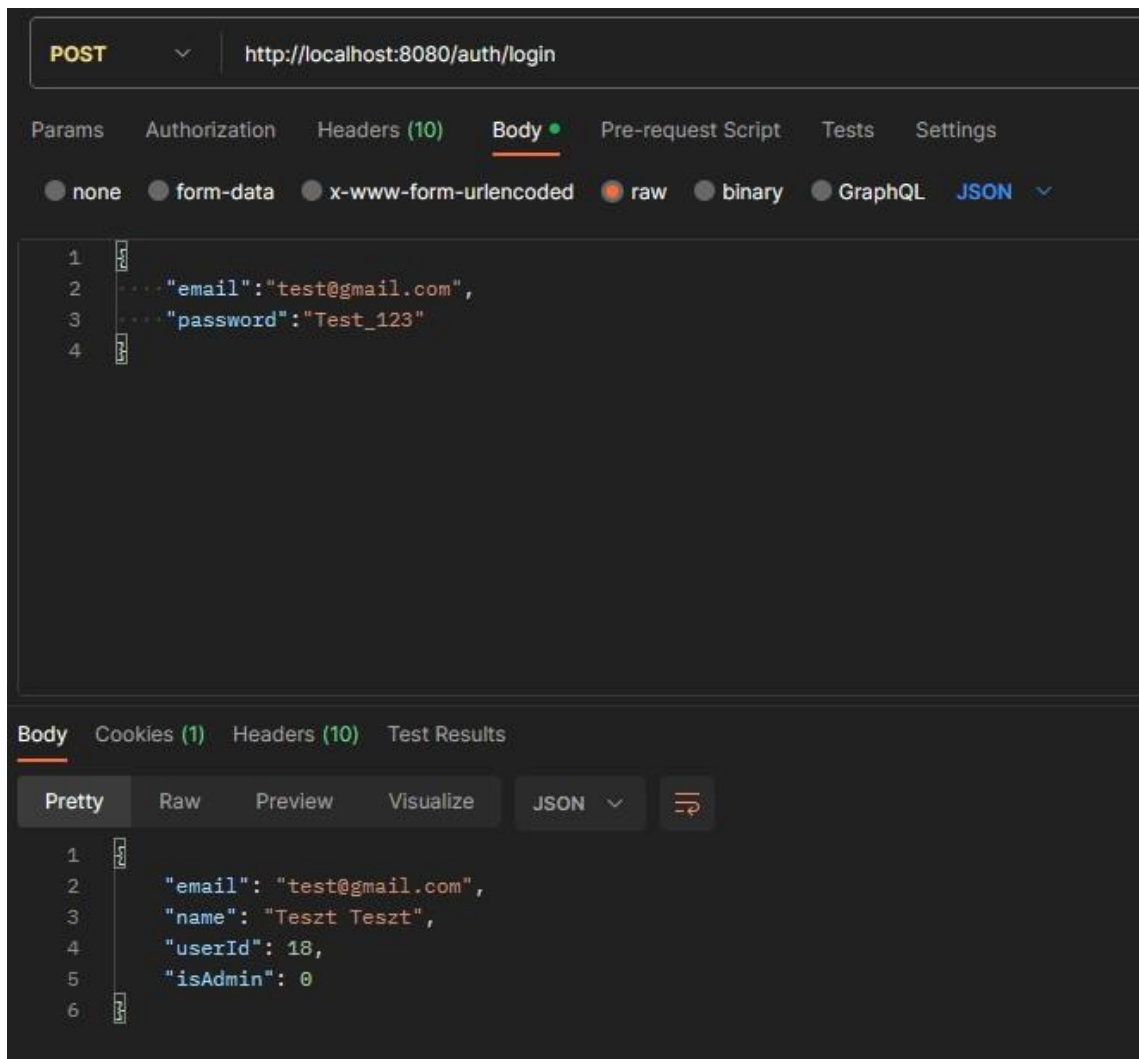
A rendszert több népszerű asztali böngészőn is teszteltünk, ezek pedig a Google Chrome (112.0.5615.138), a Microsoft Edge (112.0.1722.48) és az Opera (98.0.4759.6). Emellett Android operációs rendszerrel rendelkező okostelefonon is teszteltük az applikációt.

Az adatok mentéséhez szükségünk volt egy adatbázisra, ehhez pedig egy adatbázis szervert kellett futtatnunk. A fejlesztés során ennek megvalósításához a XAMPP beépített MariaDB szerverét használtuk.

A route -ok tesztelése legsokrétűbben a Postman szoftverrel valósítható meg, amelyet kifejezetten fejlesztőknek találtak ki API -jaik tervezésére, elkészítésére és tesztelésére.



*Regisztráció tesztelése Postman-ben*



*Login autentikáció tesztelése Postman-ben*

## **10. BEVEZETÉS, ÉLES ÜZEMMÓD**

A bevezetésről és az éles üzemmódról a felhasználói dokumentáció „Üzembe helyezés” bekezdésében olvashatunk.

## 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 nemtökéletes, további javításokra és fejlesztésekre szorul.

Bár az alkalmazás már elkészült, további finomításokra és fejlesztésekre van szüksége annak érdekében, hogy jobban működjön. Ilyen fejlesztések lehetnek például:

- Bankkártyás fizetés megvalósítása
- Új jelszó igénylése

A teljes alkalmazás Git repository -ja:

<https://github.com/Blade1201/Hopstop>