**Fejlesztői dokumentáció**

Heaven Hotel weboldal

***Témavezetők:*** ***Készítették:***

Szabó Ákos Osikóczki Sándor Mátyás

Rehó János Soltész Mónika

Varga Tímea

NYÍREGYHÁZA  
2022

**Tartalomjegyzék**

[1.](#_heading=h.30j0zll) Feladat megadása 3

[1.1 Szereplők és igényeik 3](#_heading=h.1fob9te)

[2.](#_heading=h.3znysh7) Use-Case Funkció lista 4

[2.1 Regisztrált felhasználó 4](#_heading=h.2et92p0)

[2.2. Látogató 4](#_heading=h.tyjcwt)

[2.3. Adminisztrátor 4](#_heading=h.3dy6vkm)

[3.](#_heading=h.1t3h5sf) Magas szintű rendszerterv 5

[4.](#_heading=h.4d34og8) Screenek 6

[5.](#_heading=h.2s8eyo1) Modellek 9

[6.](#_heading=h.17dp8vu) Alkalmazások kiválasztása 10

[6.1. Frontend 10](#_heading=h.3rdcrjn)

[6.2. Backend 11](#_heading=h.26in1rg)

[6.3. Adatbázis 11](#_heading=h.lnxbz9)

[6.4. Tesztelés 11](#_heading=h.35nkun2)

[7.](#_heading=h.1ksv4uv) Routing 12

[7.1 Authentikáció 12](#_heading=h.44sinio)

[7.2 Felhasználók 12](#_heading=h.2jxsxqh)

[7.3 Foglalások 13](#_heading=h.1v1yuxt)

[7.4.Szobák 13](#_heading=h.4f1mdlm)

[8.](#_heading=h.2u6wntf) Implementáció 14

[9.](#_heading=h.19c6y18) Tesztelés 15

[10.](#_heading=h.3tbugp1) Bevezetés, éles üzemmód 17

# 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

1. ***Adminisztrátor***

* Bejelentkezett felhasználók adatainak elérése
* Bejelentkezett felhasználók adatainak módosítása
* Bejelentkezett felhasználók adatainak törlése
* Szobák módosítása
* Szobák törlése
* Szobatípusok módosítása
* Szobafoglalások törlése

# 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
* Rögzített szobák kilistázása szoba típus alapján
* Bejelentkezés
* Kijelentkezés
* Szoba lefoglalása

## 2.2. Látogató

* Böngészés
* Rögzített szobák kilistázása szoba típus alapján
* 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 foglalóként kezeljük a látogatót.

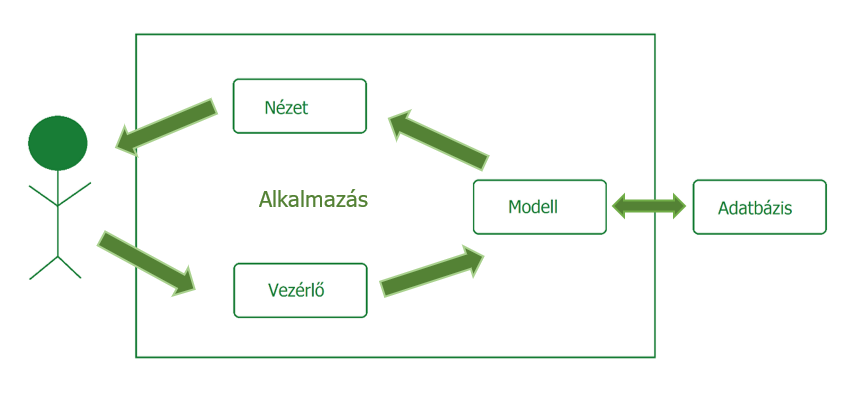
Szobát foglalni csak a bejelentkezett felhasználó tud, mivel szükségesek a regisztráláskor megadott adatok.

## 2.3. Adminisztrátor

* A szobákat az adminisztrációs felületen lehet módosítani, illetve törölni
* A szobákatípusokat 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

# Magas szintű rendszerterv

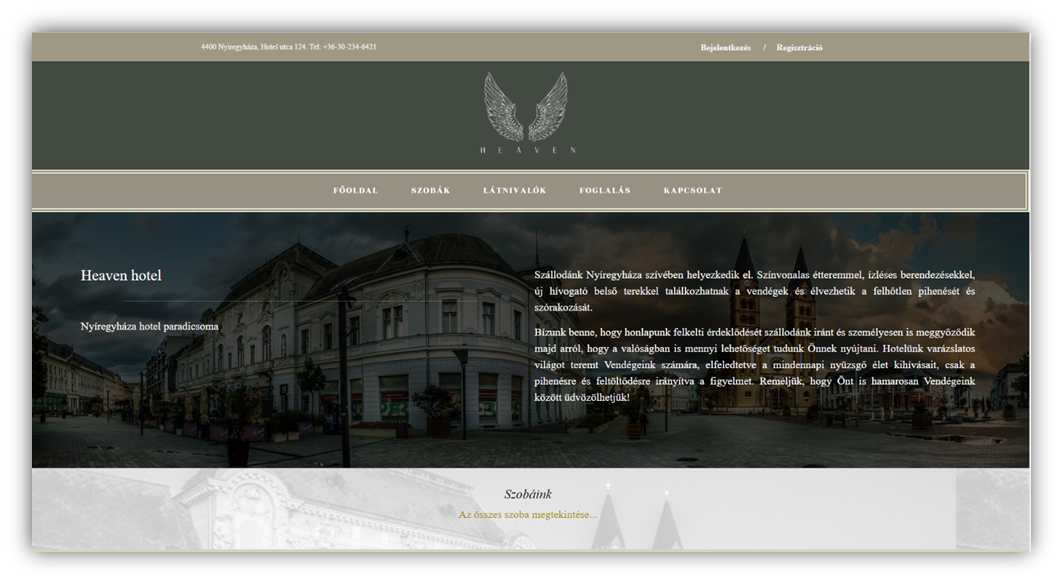
A fejlesztés során az MVC szemléletmód figyelembevételével igyekeztünk elkülöníteni az adatok elérését, az üzleti logikát, az adatok megjelenítését és a felhasználói interakciót. Ezzel biztosítjuk azt, hogy a felhasználói felület ne befolyásolja a kezelést, illetve az adatok a későbbi fejlesztések során,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



*MVC modell*

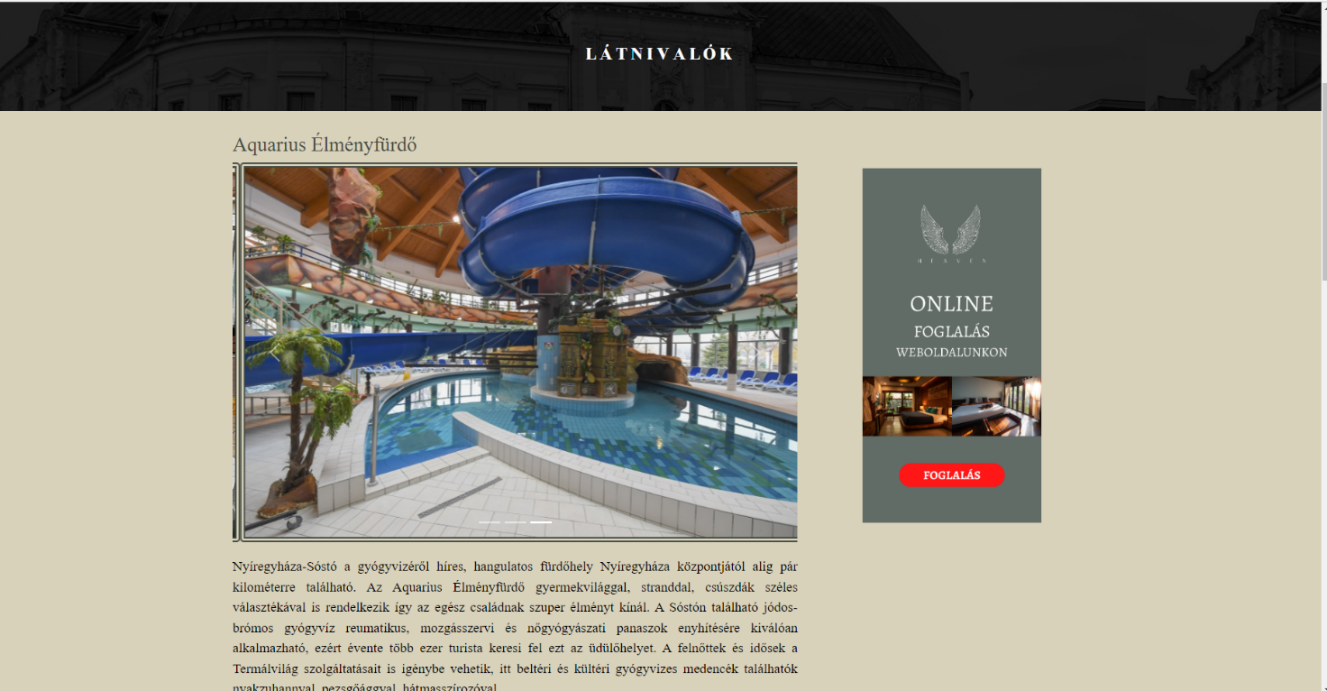
# Screenek

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 feltűntetve, vagy például az árak ÁFÁ-val terheltek, vagy sem. Az ilyen alapvető adatok hiánya miatt a vásárló más boltot választhat.



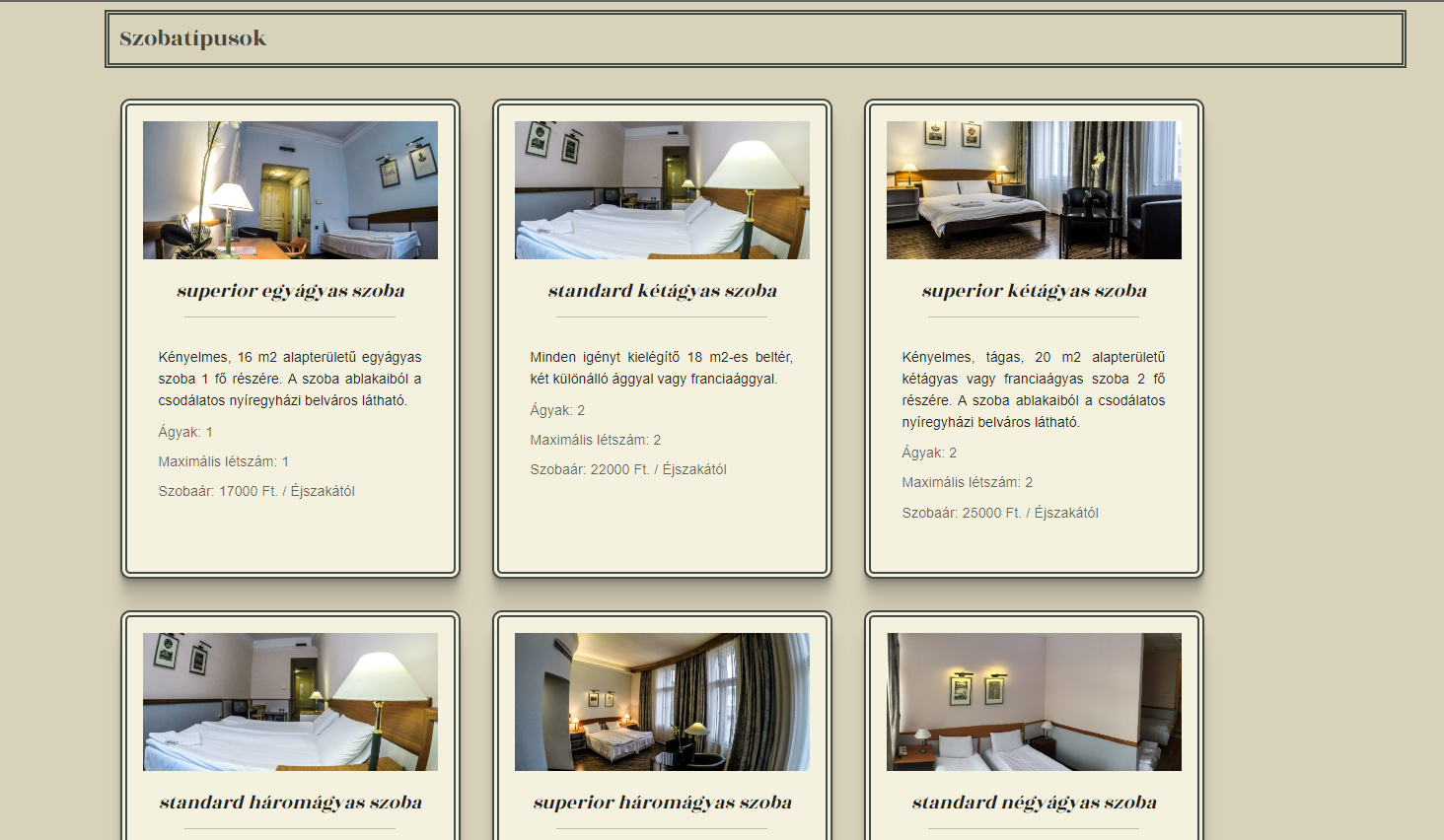
*Kezdőképernyő*

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 kérdésekre.



*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 megjelenítése*

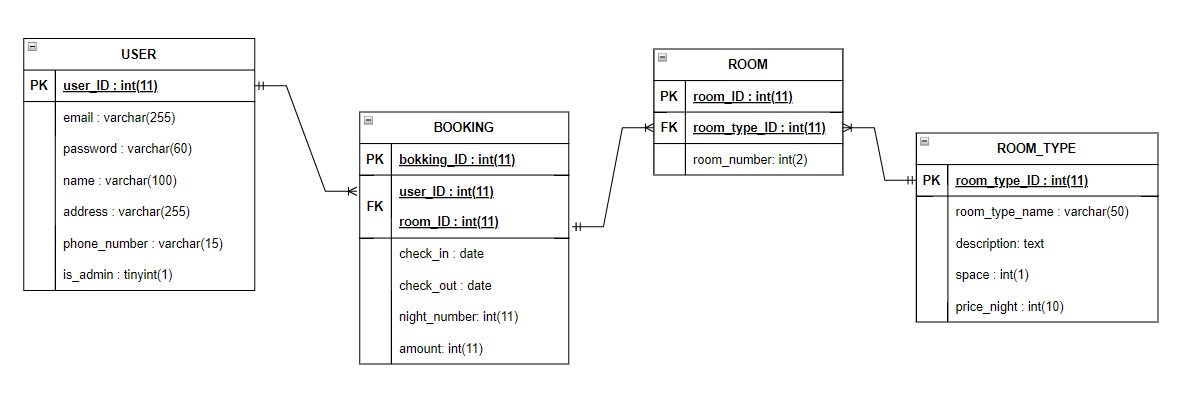


*Regisztrációs felület*



Sikeres foglalást követően a vendégek e-mailben visszajelzést kapnak.

# Modellek



# 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 keretrendszerré vált Node.js alatti fejlesztések esetében. (http://maas.hu/learning/it/prog/javascript/express/)

* ***bcrypt***– hashelés (jelszó)

A Bcrypt egy népszerű és megbízható módszer a jelszavak kivonására.

* ***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 webszerver-szoftvercsomag, azok számára, akik adatbázissal támogatott webszervert 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.

# Routing

## 7.1 Authentikáció

**POST /login**

tervezett feladat authentikáció

bemenet felhasználónév, jelszó

kimenet felhasználónév és egy jogosultságot meghatározó érték

## 7.2 Felhasználók

**GET /users**

tervezett feladat összes felhasználó kislistázása

bemenet

kimenet felhasználók listája, adataik

**GET /users/:id**

tervezett feladat adott felhasználó adatainak kislistázása

bemenet paraméterként kapott azonosító

kimenet adott felhasználó listája

**PUT /users**

tervezett feladat felhasználó rögzítése

bemenet felhasználó adatai

kimenet visszajelzés a művelet sikerességéről

**DELETE /user/:id**

tervezett feladat töröl egy felhasználót azonosító alapján

bemenet felhasználó azonosítója

kimenet a törölt felhasználó renderelt listája

## 7.3 Foglalások

**POST/reservations**

tervezett feladat szoba foglalása

bemenet a felhasználó és foglalás adatai

kimenet amennyiben sikeres a foglalás emailben visszaigazolja

**POST/ reservations/availablerooms**

tervezett feladat elérhető szabad szobák szűrése

bemenet szoba típusa, kijelentkezés/ bejelentkezés dátuma

kimenet szabad szobák listája

## 7.4.Szobák

**GET/roomtypes**

tervezett feladat Összes szobatípus kilistázása

bemenet Szobatípusa

kimenet Szobatípus lista

**GET/roomtypes/:id**

tervezett feladat Adott szobatípus kilistázása

bemenet Szoba azonosítója

kimenet Adott szoba listája

**PUT/roomtypes/:id**

tervezett feladat Adott szoba módosítása azonosító alapján

bemenet Szoba adatai

kimenet Módosított szoba

# 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/)

# Tesztelés

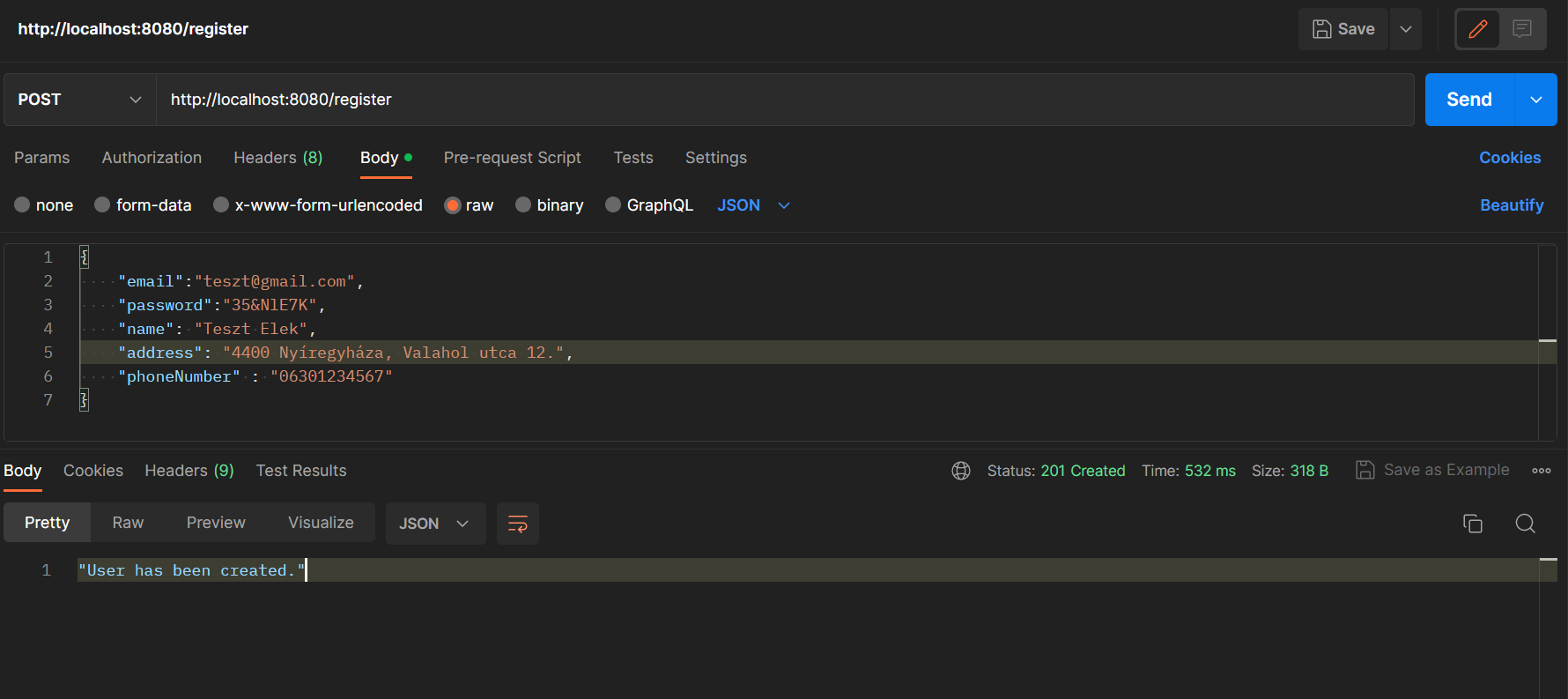
A felhasználó szempontjából a rendszer futtatásához elegendő bármely olyan operációs rendszer, amelyen futtatható HTML, CSS és JavaScript technológiával készült alkalmazás megjelenítésére képes böngésző.

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 authentikáció tesztelése Postman-ben*

A képen szöveg, képernyőkép, monitor, válaszfal látható

Automatikusan generált leírás

*Regisztráció tesztelése Postman-ben*

# Bevezetés, éles üzemmód

* Az elkészült munka megtekintéséhez szükséges a XAMPP letöltése. Látogassunk el a következő oldalra: <http://www.apachefriends.org/en/xampp.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.
* töltse 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 Front mappát nyitjuk meg, a másikban a Back-end 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.
* 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.