

# IDŐPONTFOGLALÓ WEBALKALMAZÁS

## Projekt dokumentáció

**Hamvai Sándor**

**Lisovszki Fanni**

**Éles Máté**

<b>Projekt típusa</b>	WEBAPP Beadandó
<b>Frontend</b>	HTML, CSS, JavaScript, Bootstrap 5
<b>Backend</b>	Node.js, Express.js
<b>Adatbázis</b>	MySQL
<b>Hitelesítés</b>	JWT token alapú
<b>Email kezelés</b>	Nodemailer
<b>Fő funkció</b>	Időpontfoglalás 1 órás intervallumokra, minden nap 8:00 és 16:00 között

## Tartalomjegyzék

1. Bevezetés
2. A rendszer célja
3. Funkciók
4. Felhasznált technológiák
5. Projektstruktúra
6. Adatbázis
7. Backend, API-végpontok
8. Frontend oldalak
9. Hitelesítés és jogosultságkezelés
10. Email-es folyamatok

## 1. Bevezetés

Ez a dokumentáció az Időpontfoglaló webalkalmazás beadandó projektünk leírása. A rendszer célja hogy a felhasználók tudjanak időpontot foglalni egy webes felületen keresztül, az adminisztrátor pedig kezelni tudja a foglalásokat.

Az alkalmazás három rétegből áll: Bootstrap alapú frontendből, Node.js Express backendből és MySQL adatbázisból. A kommunikáció REST API-végpontokon keresztül történik, JSON adatküldéssel.

## 2. A rendszer célja

A projekt egy egyszerű időpontfoglaló alkalmazást valósít meg. A rendszer alkalmas arra, hogy a felhasználó megadja személyes kapcsolattartási adatait, kiválasszon egy dátumot, majd a szabad időpontok közül lefoglaljon egy 1 órás intervallumot.

A foglalást követően a felhasználó emailes visszaigazolást kap. Az email tartalmaz egy lemondási linket, valamint egy olyan linket, ahol a foglaláshoz tartozó megjegyzését módosíthatja.

## 3. Funkciók

### 3.1 Felhasználói funkciók

- Kezdőlap megjelenítése
- Időpontfoglalási űrlap megjelenítése név, telefonszám, email, dátum és megjegyzés mezőkkel.
- Szabad időpontok lekérése a kiválasztott dátum alapján.
- Foglalás rögzítése
- Emailes visszaigazolás foglalás után.
- Foglalás lemondása emailben kapott linken keresztül.
- Megjegyzés módosítása emailben kapott linken keresztül.

### 3.2 Adminisztrátori funkciók

- Admin bejelentkezés felhasználónévvel és jelszóval.
- JWT token alapú hitelesítés használata az admin funkciók használatához.
- Foglalások listázása az admin felületen.
- Foglalás törlése admin által.
- Védett API-végpontok használata Authorization headerrel.

## 4. Felhasznált technológiák

Terület	Használt technológia	Leírás
Frontend	HTML, CSS, JavaScript	Oldalak, űrlapok és kliensoldali működés megvalósítása.
Megjelenés	Bootstrap 5	Reszponzív, egységes és gyors felület.
Backend	Node.js	Szerveroldali JS futtatókörnyezet.
Backend keretrendszer	Express.js	REST API megvalósítása.
Adatbázis	MySQL	Adminok és foglalások tárolása.
Hitelesítés	jsonwebtoken	JWT token létrehozása és ellenőrzése.
Jelszóvédelem	bcryptjs	Admin jelszó hashelése és ellenőrzése.
Email küldés	Nodemailer	Visszaigazoló email küldése.
Konfig	dotenv	.env fájlban tárolt konfig változók betöltése.

## 5. Projektstruktúra

A projekt áttekinthető mappaszerkezetet használ. A publikus frontend fájlok a public mappában, a backend útvonalak és a függvények pedig az src mappában találhatóak.

Fájl / mappa	Leírás
server.js	Az Express alkalmazás belépési pontja, route-ok és statikus fájlok beállítása.
database.sql	MySQL adatbázis fájl.
.env	Konfigfájl.
public/index.html	Kezdőlap.
public/booking.html	Időpontfoglalási felület.
public/login.html	Admin bejelentkezési oldal.
public/admin.html	Admin foglaláskezelő oldal.
public/modify-note.html	Megjegyzés módosító oldal.
src/routes/auth.routes.js	Admin bejelentkezés API-végpontja.
src/routes/booking.routes.js	Publikus foglalási API-végpontok.
src/routes/admin.routes.js	JWT-vel védett admin API-végpontok.
src/middleware/auth.middleware.js	JWT token ellenőrzése.
src/config/db.js	MySQL pool beállítása.
src/utils/mailer.js	Email küldési logika.
src/utils/timeSlots.js	Idősávok generálása és ellenőrzése.

## 6. Adatbázis

A projekt MySQL adatbázist használ. Két fő tábla alkotja: admins és bookings. Az admins tábla az adminisztrátorokat, a bookings tábla a foglalásokat tárolja.

### 6.1 admins tábla

Mező	Típus	Leírás
id	INT AUTO_INCREMENT PRIMARY KEY	Egyedi admin felhasználó azonosító.
username	VARCHAR(100) UNIQUE	Admin felhasználónév.
password_hash	VARCHAR(255)	Bcrypt-tel hash-elt jelszó.
created_at	TIMESTAMP	Létrehozás időpontja.

### 6.2 bookings tábla

Mező	Típus	Leírás
id	INT AUTO_INCREMENT PRIMARY KEY	Egyedi foglalás azonosító.
name	VARCHAR(150)	Foglaló neve.
phone	VARCHAR(50)	Foglaló telefonszáma.
email	VARCHAR(150)	Foglaló email címe.
booking_date	DATE	Foglalás dátuma.
start_time	TIME	Foglalás kezdési időpontja.
end_time	TIME	Foglalás befejezési időpontja.
note	TEXT	Felhasználói megjegyzés.
cancel_token	VARCHAR(128) UNIQUE	Egyedi token a foglalás lemondásához.
note_token	VARCHAR(128) UNIQUE	Egyedi token a megjegyzés módosításához.
created_at	TIMESTAMP	Foglalás létrehozásának ideje.
UNIQUE KEY	(booking_date, start_time)	Megakadályozza hogy ugyanarra az időpontra több foglalás kerüljön rögzítésre.

## 7. Backend, API-végpontok

A backend Express.js -el, REST API-végpontokat biztosít. Az API JSON formátumban kommunikál a frontenddel. A rendszerben GET, POST, PUT és DELETE HTTP metódusokat használtunk fel.

HTTP metódus	Végpont	Védett?	Feladat
GET	/api/bookings/slots?date=YYYY-MM-DD	Nem	Szabad és foglalt időszavok lekérése adott dátumra.
POST	/api/bookings	Nem	Új foglalás létrehozása.
GET	/api/bookings/cancel/:token	Nem	Foglalás lemondása email token alapján.
GET	/api/bookings/note/:token	Nem	Megjegyzés módosító oldal adatainak lekérése.
PUT	/api/bookings/note/:token	Nem	Meglévő megjegyzés módosítása.
POST	/api/auth/login	Nem	Admin bejelentkezés és JWT token kiadása.
GET	/api/admin/bookings	Igen	Összes foglalás listázása admin felületen.
DELETE	/api/admin/bookings/:id	Igen	Foglalás törlése admin jogosultsággal.

## 8. Frontend oldalak

Oldal	Fájl	Feladat
Kezdőlap	index.html	A webalkalmazás bemutatása.
Foglalás	booking.html	Foglalási űrlap.
Bejelentkezés	login.html	Admin belépés felhasználónévvel és jelszóval.
Admin felület	admin.html	Foglalások listázása és törlése.
Megjegyzés módosítása	modify-note.html	Email token alapján a foglalási megjegyzés szerkesztése.

A frontend Bootstrap alapú komponenseket használ a megjelenéshez.

## 9. Hitelesítés és jogosultságkezelés

Az admin funkciók JWT token alapú hitelesítéssel védettek. Bejelentkezéskor a backend ellenőrzi az admin felhasználónevet és jelszót a MySQL adatbázisban. A jelszó összehasonlítása bcrypt segítségével történik.

1. Az admin megadja a felhasználónevet és jelszavát a bejelentkezési oldalon.
2. A frontend POST kérést küld a /api/auth/login végpontra.
3. A backend lekéri az admin rekordot az adatbázisból.
4. A bcrypt.compare ellenőrzi, hogy a megadott jelszó egyezik-e a hash-elj jelszóval.
5. Sikeres belépés után a backend JWT tokenet állít ki 2 órás lejáratú idővel.
6. A frontend a tokenet eltárolja, majd az admin kérések Authorization headerében küldi tovább.

## 10. Emailés folyamatok

A rendszer foglalás után emailt küld a felhasználónak. Az emailben szerepelnek a foglalás adatai, valamint két link: az egyik a foglalás lemondására, a másik a megjegyzés módosítására szolgál.

Email elem	Leírás
Foglalási adatok	Dátum, kezdési és befejezési idő, telefonszám és megjegyzés.
Lemondás gomb	A cancel_token alapján törli a foglalást az adatbázisból.
Megjegyzés módosító link	A note_token alapján megnyitja a módosító oldalt.
Nodemailer	SMTP szerveren keresztül elküldi az emailt.