Aplikacija za upravljanje rezervacijama

Sadržaj

[1. Uvod 4](#_Toc193831907)

[2. Baza podataka 5](#_Toc193831908)

[3. Backend 5](#_Toc193831909)

# Uvod

Cilj razvoja ove aplikacije je stvoriti univerzalni sustav za upravljanje rezervacijama, primjenjiv u različitim kontekstima. Mnogi korisnici koje se oslanjaju na zakazivanje termina i dalje koriste neučinkovite i zastarjele sustave, što otežava rad i smanjuje zadovoljstvo korisnika. Ideja je ponuditi pouzdano, intuitivno i fleksibilno rješenje koje se lako prilagođava specifičnim potrebama, bez potrebe za izradom zasebnih sustava za svakog korisnika. Time se standardizira proces rezervacija i podiže kvaliteta korisničkog iskustva.

Aplikacija je zamišljena kao backend implementacija koja pruža sve ključne funkcionalnosti i jednostavno se integrira s različitim sučeljima i poslovnim sustavima. Služi kao temeljni servis koji se koristi putem API-ja u web i mobilnim aplikacijama ili drugim poslovnim alatima.

Kroz jasno definirano RESTful sučelje omogućuje se potpuno upravljanje rezervacijama – od prikaza slobodnih termina do kreiranja i otkazivanja. Jednom razvijen, sustav može služiti velikom broju korisnika bez potrebe za dodatnim prilagodbama.

U praksi, klijenti razvijaju vlastita sučelja i povezuju ih s ovim backendom, koristeći ga kao centralni alat za upravljanje rezervacijama. Takva arhitektura omogućuje fleksibilnost, modularnost i jednostavno proširivanje funkcionalnosti. Aplikacija predstavlja standardiziranu osnovu za brzu i učinkovitu digitalnu implementaciju rezervacijskog sustava.

# Baza podataka

Za potrebe ove aplikacije koristi se MongoDB kao baza podataka. Glavni razlog za taj izbor jest fleksibilnost i skalabilnost koju MongoDB nudi, što se savršeno uklapa u zahtjeve sustava koji mora podržavati različite korisničke scenarije.

S obzirom na to da aplikacija mora biti univerzalno rješenje primjenjivo u različitim primjenama, struktura podataka može se značajno razlikovati od slučaja do slučaja. MongoDB, kao dokumento orijentirana NoSQL baza, omogućuje pohranu podataka u obliku fleksibilnih JSON dokumenata, bez stroge sheme.

Osim toga, MongoDB omogućuje brzo čitanje i pisanje podataka, što je ključno za sustave rezervacija koji zahtijevaju visoku dostupnost i responzivnost. MongoDB je jednostavan za horizontalno skaliranje, što osigurava spremnost sustava za veću količinu korisnika.

Iako relacijske baze poput PostgreSQL-a nude napredne mogućnosti za rad s kompleksnim odnosima, u ovom slučaju važnije su prednosti fleksibilnosti, brzine razvoja i lakše prilagodbe koje nudi MongoDB.

# Backend

## 3.1. Endpoints

**GET** /api/reservations

**Opis:** Dohvaća sve postojeće rezervacije.

**Prima:** Nema parametara.

**Vraća:** Popis svih rezervacija s informacijama o korisniku, terminu, statusu i vremenu kreiranja

**POST /api/reservations/{id}**

**Opis:** Kreira novu rezervaciju za željeni termin, ako je on još dostupan. Rezervacija se tada sprema i označava kao aktivna.

**Prima:**

* id – ID željenog termina

**Vraća:**

* Potvrdu o uspješnom kreiranju rezervacije s pripadajućim podacima
* Ako termin nije dostupan, vraća poruku o grešci

**PATCH /api/reservations/{id}**

**Opis:** Ažurira (npr. otkazuje) postojeću rezervaciju.

**Prima:**

* id – ID rezervacije
* **Body:** novi status rezervacije (npr.  status : *cancelled*)

**Vraća:**  
– Potvrdu o ažuriranoj rezervaciji  
– Ako rezervacija nije pronađena, vraća poruku o grešci

**GET /api/reservations/{id}**

**Opis:** Dohvaća detalje pojedine rezervacije prema njenom ID-u

**Prima:**

* id rezervacije

**Vraća:**

* Detaljne informacije o rezervaciji
* Ako rezervacija nije pronađena, vraća poruku o grešci

## 3.2. Modeli

User (Korisnik)

Predstavlja korisnika sustava – može biti običan korisnik, poslovni korisnik ili administrator.

Polja:

* Id- user id
* name – Ime korisnika (tekst, obavezno)
* email – Email adresa korisnika (tekst, jedinstvena, obavezna)
* role\_id – id role korisnika u sustavu
* createdAt - automatski zapis vremena kreiranja
* updatedAt – automatski zapis vremena zadnje izmjene

Appointment (Termin)

Predstavlja termine koji korisnici mogu rezervirati.

Polja:

* id – id termina
* time – Datum i vrijeme termina (format: datum/vrijeme, obavezno)
* available – Dostupnost termina (true ako je slobodan, false ako je zauzet; zadano: true)
* owner\_id – id korisnika koji je vlasnik termina (tip: User, obavezno)
* createdAt - automatski zapis vremena kreiranja
* updatedAt – automatski zapis vremena zadnje izmjene

Reservation (Rezervacija)

Predstavlja rezervaciju termina od strane korisnika.

Polja:

* id - id rezervacije
* user\_id – Referenca na korisnika koji je napravio rezervaciju (tip: User, obavezno)
* Appointment \_id – id termina
* appointment – Referenca na termin koji je rezerviran
* status – Status rezervacije (active ili cancelled; zadano: active)
* createdAt - automatski zapis vremena kreiranja
* updatedAt – automatski zapis vremena zadnje izmjene

Role

Sadrži role koje user može imati

Polja:

* id - id role
* name - ime role
* createdAt - automatski zapis vremena kreiranja
* updatedAt – automatski zapis vremena zadnje izmjene