­SVEUČILIŠTE U ZAGREBU

**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI RAD br. 1946

**Web-aplikacija za organizaciju događaja i rezervaciju termina**

Petar Kapec

# SVEUČILIŠTE U ZAGREBU

**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

Zagreb, 3. ožujka 2025.

ZAVRŠNI ZADATAK br. 1946

Pristupnik: **Petar Kapec (0036549401)**

Studij: Elektrotehnika i informacijska tehnologija i Računarstvo Modul: Računarstvo

Mentor: doc. dr. sc. Slaven Zakošek

Zadatak: Web-aplikacija za organizaciju događaja i rezervaciju termina

Opis zadatka:

Poduzeća, organizacije ili pojedinci koji na tržištu nude organiziranje i ugošćivanje događaja moraju voditi računa o mnogim pojedinostima, od rezervacije i uređivanja prostora do opskrbe hranom. Izraditi web-aplikaciju koja će podržati razne aspekte organizacije događaja. Organizatorima događaja treba omogućiti opisivanje ponude (prostori, osoblje, dekoracije i dizajn, hrana i piće, glazbenici i drugi zabavljači, sigurnost, prijevoz gostiju, slobodni termini, itd.), a zatim i upravljanje događajem (vođenje liste gostiju, slanje pozivnica, itd.) te evidentiranje plaćanja. Klijentima, odnosno korisnicima treba omogućiti efikasni pregled i pretraživanje ponuda prema različitim kriterijima, rezerviranje termina, odabir opcionalnih sadržaja i ocjenjivanje organizatora događaja.

Rok za predaju rada: 23. lipnja 2025.

Sadržaj

**Table of Contents**

[Uvod 1](#__RefHeading___Toc23263729)

[1. Pregled i analiza postojećih sustava 2](#__RefHeading___Toc23263730)

[1.1. AirBNB.com 2](#__RefHeading___Toc23263731)

[1.2. Trivago.com 2](#__RefHeading___Toc23263732)

[1.3. Njuškalo.hr/(Isključivo /nekretnine) 3](#__RefHeading___Toc23263733)

[1.4. Booking.com 3](#__RefHeading___Toc23263735)

[1.5. Eventim.hr 3](#__RefHeading___Toc23263736)

[2. Specifikacija sustava 5](#__RefHeading___Toc4101_723378845)

[2.1. Funkcionalni zahtjevi 5](#__RefHeading___Toc4103_723378845)

[2.2. Ostali zahtjevi 6](#__RefHeading___Toc4105_723378845)

[3. Arhitektura i dizajn sustava 8](#__RefHeading___Toc4107_723378845)

[3.1. Frontend sloj 8](#__RefHeading___Toc4109_723378845)

[3.2. Backend sloj 9](#__RefHeading___Toc4111_723378845)

[3.3. Model Baze podataka 10](#__RefHeading___Toc4113_723378845)

[3.3.1. Entitet Dogadjaj 10](#__RefHeading___Toc4115_723378845)

[3.3.2. Entitet Dogadjaj\_Ponuda 11](#__RefHeading___Toc4117_723378845)

[3.3.3. Entitet Gost 11](#__RefHeading___Toc4119_723378845)

[3.3.4. Entitet Korisnik 12](#__RefHeading___Toc4121_723378845)

[3.3.5. Entitet Ponuda 12](#__RefHeading___Toc4123_723378845)

[3.3.6. Entitet Prostor 13](#__RefHeading___Toc4125_723378845)

[3.3.7. Entitet Rezervacija 13](#__RefHeading___Toc4127_723378845)

[3.3.8. ENUM tablice 14](#__RefHeading___Toc4129_723378845)

[3.3.9. Restrikcije baze podataka 15](#__RefHeading___Toc4131_723378845)

[4. Implementacija i korisničko sučeljem 17](#__RefHeading___Toc4133_723378845)

[4.1. Korišteni alati i tehnologije. 17](#__RefHeading___Toc4135_723378845)

[4.2. Korisničke upute 17](#__RefHeading___Toc4137_723378845)

[4.3. ORGANIZATOR 19](#__RefHeading___Toc4139_723378845)

[4.4. KUPAC 27](#__RefHeading___Toc4141_723378845)

[5. Moguće promjene i nadogradnje sustava 31](#__RefHeading___Toc4143_723378845)

[Zaključak 32](#__RefHeading___Toc23263739)

[Literatura 33](#__RefHeading___Toc23263740)

[Sažetak 34](#__RefHeading___Toc23263741)

[Summary 35](#__RefHeading___Toc23263742)

[Skraćenice 36](#__RefHeading___Toc23263743)

[Privitak 37](#__RefHeading___Toc23263744)

# Uvod

Posljednjih godina znatno se povećalo tržište turističkih rezervacija te iznajmljivanja nekretnina. Digitalizacija je značajno olakšala procese upravljanja rezervacijama te komunikacijom između gosta i iznajmljivača. Danas su sustavi upravljanjem rezervacijama neizostavan dio svakodnevnice strankama koje se bave 3. gospodarskim sektorom. Također, sustavi za rezervaciju karata za koncerte i ostale različite događaje nikad nisu bili potrebniji.. Potrebna je dobra logistika i čin organizacije same rezervacije kako bi se događaj rasprodao u što više primjeraka i kako bi se što jednostavnije vodio zapis upravljanja gostima, to jest, u ovom slučaju publikom koncerta. Aplikacija koju opisujemo u ovom radu upravo rješava baš taj problem ali na punoj manjoj skali, na lokalnim događajima i proslavama. U ovom se radu opisuje implementacija, dizajn i arhitektura web aplikacije koja omogućuje organizatorima rasterećen posao organiziranje proslava, te korisnicima brže i preglednije vođenje proslava. U uvodnom dijelu rada, predstavljen je kontekst i motivacija za razvoj aplikacije te su definirani osnovni ciljevi projekta. U nastavku rada detaljno će se razraditi analiza zahtjeva, arhitektura sustava, implementacija te zaključak o postignutim rezultatima i mogućnostima budućeg razvoja aplikacije.

# Pregled i analiza postojećih sustava

Predstavit ćemo već postojeće i vodeće svjetske sustave u domeni rezerviranja i raspoređivanja rezervacija. Predstavit ćemo prednosti i mane svakog sustava te objasniti koji dio tržišta i tehnologije pokrivaju.

## AirBNB.com

Airbnb je globalna platforma koja se koristi za kratkoročni najam smještaja, a sve češće i za organizaciju manjih događanja poput proslava, okupljanja i team buildinga. Korištenje platforme omogućuje korisnicima velik izbor objekata poput vila i kuća koje često imaju dodatne sadržaje poput bazena, roštilja i prostranih okućnica, pogodnih za razna događanja. Komunikacija između korisnika i iznajmljivača odvija se direktno putem poruka unutar aplikacije, što olakšava dogovor oko detalja. Ipak, valja naglasiti da mnogi domaćini ne dopuštaju održavanje proslava u svojim objektima, što zahtijeva dodatnu komunikaciju i provjeru. Također, platforma nije specijalizirana za upravljanje događanjima jer ne nudi alate poput popisa gostiju ili mogućnosti angažiranja dodatnih usluga, budući da je primarno fokusirana na smještaj, a ne na event industriju. [15]

## Trivago.com

Trivago je pretraživač koji korisnicima omogućuje usporedbu cijena hotelskog smještaja s različitih internetskih stranica. Koristan je prilikom traženja smještaja za goste koji dolaze na neki događaj, osobito kada se traži najbolja cijena u određenom terminu. Njegova prednost je u brzom i jednostavnom pronalaženju smještaja putem poveznica na druge platforme kao što su Booking.com i Expedia. Međutim, Trivago ne nudi mogućnost izravne komunikacije s iznajmljivačem niti bilo kakve funkcionalnosti vezane uz organizaciju samog događanja. Fokusiran je isključivo na hotelski smještaj, bez dodatnih opcija za prilagodbu sadržaja potrebama eventa ili posebnih uvjeta. [16]

## Njuškalo.hr/(Isključivo /nekretnine)

Njuškalo je najpoznatija hrvatska oglasna platforma koja, između ostalog, nudi mogućnost pronalaska prostora za proslave, vjenčanja, rođendane i slična događanja. Prednost korištenja Njuškala je širok izbor prostora koji su većinom jasno označeni kao pogodni za proslave, te mogućnost direktnog kontakta s vlasnicima putem telefona ili e-maila. Dogovori su često fleksibilni i individualni, što korisnicima omogućuje pregovore oko cijene, termina i dodatnih sadržaja. Ipak, platforma nema ugrađen sustav za rezervacije, što otežava praćenje dostupnosti i potvrdu termina. Komunikacija se odvija izvan same aplikacije, što može dovesti do nejasnoća ili nedostatka transparentnosti, a ažurnost oglasa nije uvijek zajamčena. Na prostoru Hrvatske se prostori za proslave ne nalaze na jednom mjestu tako da korisnici često moraju „surfati” i puno izvan okvira Njuškala (Kao što je Facebook marketplace itd.) čime se znatno povećava vrijeme i trud koji korisnik mora uložiti da bi rezervirao prostor za proslavu jer je Njuškalo pretežito namijenjen za iznajmljivanje prostora za generalne potrebe a ne specifično za događaje. [17]

## Booking.com

Booking.com je svjetski poznata platforma za rezervaciju smještaja, koja uz klasične hotele nudi i vile, apartmane te druge objekte koji se mogu koristiti za organizaciju događaja. Korisnici mogu jednostavno filtrirati objekte prema lokaciji, broju osoba i dodatnim sadržajima, što olakšava odabir smještaja prilagođenog potrebama proslave. Prednost Booking.com-a je u profesionalnom pristupu, jasnim uvjetima rezervacije, mogućnostima otkazivanja i sustavu recenzija koji pomaže u procjeni kvalitete objekta. Ipak, komunikacija s iznajmljivačem često je ograničena i odvija se putem standardiziranih poruka unutar platforme, dok mnogi objekti ne navode jasno mogu li se koristiti za organizaciju proslava. Osim toga, platforma nije specijalizirana za evente, već je primarno usmjerena na turistički smještaj. [18]

## Eventim.hr

Eventim.hr je vodeća hrvatska platforma za prodaju ulaznica za koncerte, festivale, sportske manifestacije i druga javna događanja. Njezina najveća prednost je profesionalan sustav prodaje karata, velika posjećenost i pouzdana distribucija, što omogućuje široku vidljivost i kontrolu nad prodajom ulaznica. Platforma je povezana s brojnim organizatorima i lokacijama, što dodatno olakšava plasman događanja na tržište. Ipak, Eventim nije namijenjen manjim privatnim događanjima, već je fokusiran isključivo na velike javne evente. Komunikacija s organizatorom je često službena ili automatizirana, bez direktne razmjene informacija između korisnika i organizatora. Također, platforma ne nudi mogućnosti najma prostora niti upravljanja logistikom događaja, što ograničava njezinu primjenu u kontekstu sveukupne organizacije događanja. [19]

# Specifikacija sustava

U ovom uvodnom dijelu poglavlja predstavit ćemo osnovni rad aplikacije: postojali bi organizatori i kupci. Organizatori su specifične osobe koje su upoznate sa industrijom zabave i najma nekretnina pa žele koristiti svoje vrijeme kako bi organizirali razne i kvalitetne događaje te profitirali svojim znanjem i iskustvom planiranja. Kupci su osobe koje žele svoju proslavu ili zabavu ali nemaju vremena, znanja ili iskustva organizirati neki događaj. Organizatori bi postavili događaj u kojem opisuju prostor, ponudu (rekviziti, razno osoblje na zabavama i proslavama, glazbenici..), termine potencijalnih događaja te cijenu navedenog događaja u koju bi pribrojili i vlastitu komisiju. Kupci pažljivo biraju događaj po vlastitim potrebama te unose goste koje bi pozvali na isti. Nakon odabira imaju otkazni rok unutar kojeg moraju platiti rezervaciju, ako je ne plate rezervacija se otkazuje.

## Funkcionalni zahtjevi

Dionici: Organizator, Kupac

Aktori i njihovi funkcionalni zahtjevi:

* Organizator
  + Može se prijaviti, registrirati u sustav
  + Pregledavati prostore, ponude, događaje, rezervacije i termine vezane za njegove događaje
  + Može kreirati, brisati i dodavati događaje, ponude, prostore
  + Može kreirati, brisati i dodavati termine vezane za prostore
  + Može masovno dodavati termine za postojeći prostor odabirom frekvencije pojavljivanja i trajanja termina. (na primjer, sljedećih mjesec dana radnim danima prostor je slobodan od 9:00 – 17:00)
  + Može otkazati ili potvrditi rezervaciju
* Kupac
  + Može se prijaviti, registrirati u sustav
  + Pregledavati svoje prošle rezervacije i događaje koje su organizatori kreirali
  + Rezervirati događaj uz postojeće termine iz postojećih događaja
  + Dodati goste na događaj
  + Može platiti i otkazati rezervaciju

## Ostali zahtjevi

Aplikacija omogućuje paralelan rad većeg broja korisnika, čime se osigurava da više kupaca i organizatora može u isto vrijeme pristupati funkcionalnostima sustava bez usporavanja ili prekida rada. Korisničko sučelje aplikacije mora biti responzivno, odnosno automatski se prilagođavati različitim veličinama zaslona, uključujući mobilne uređaje, tablete i stolna računala, kako bi se osigurala pristupačnost i dobra korisnička iskustva.

Autentifikacija korisnika (organizatora i kupca) provodi se putem korisničkog imena i lozinke koji su individualno dodijeljeni svakom korisniku sustava, a svi zahtjevi prema aplikaciji moraju se odvijati preko sigurnog HTTPS protokola kako bi se osigurala enkripcija i zaštita prijenosa podataka. Tijekom prijave u sustav potrebno je osigurati sigurnosne mehanizme poput zaštite od pokušaja neovlaštenog pristupa i zaštite osobnih podataka korisnika. Mora se koristiti sustav zaštite pomoću JWT tokena koji se dodjeljuje korisniku unutar svake sesije.

Stabilnost sustava mora biti zajamčena i u slučajevima kada dođe do pogrešnog unosa ili prikaza elemenata korisničkog sučelja, odnosno takvi slučajevi ne smiju utjecati na funkcionalnost i stabilnost cijele aplikacije. Pri unosu podataka vezanih uz događaje (npr. datumi i termini), potrebno je poštivati standardizirani format: za datume **dd/mm/yyyy**, a za vrijeme **hh:mm:ss**, čime se izbjegavaju nesporazumi i neispravni zapisi u sustavu.

Svi pristupi bazi podataka, uključujući dohvat podataka o događajima, korisnicima i rezervacijama, moraju se izvršiti unutar zadanog vremenskog okvira koji ne smije biti dulji od 30 sekundi. Osim toga, sustav mora u potpunosti podržavati unos i prikaz sadržaja na hrvatskom jeziku, koristeći hrvatsku abecedu, kako bi korisnici mogli prirodno komunicirati sa sučeljem bez prepreka u jeziku.

Korisničko sučelje aplikacije mora biti implementirano koristeći frontend framework „**React”**, čime se osigurava modularna struktura, visoka interaktivnost i jednostavno održavanje. S druge strane, poslužiteljski dio aplikacije se mora napraviti pomoću Backend Framework-a „**Spring Boot”**, koji omogućuje robusno upravljanje pozadinskom logikom, sigurnošću i pristupom podacima.

Također, sustav mora imati implementirane sigurnosne mjere koje štite osjetljive informacije, osobito tijekom procesa prijave i autentifikacije, uključujući enkripciju lozinki i mehanizme protiv napada poput “brute-force” pokušaja pristupa. [4]

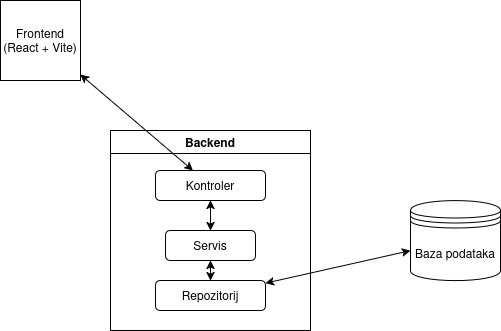
Uz to, aplikacija mora imati jasno definiranu strukturu korisničkih uloga (npr. organizator i kupac) i odgovarajuće ovlasti pristupa, kako bi se spriječilo miješanje uloga i neovlašteni pristup funkcijama koje nisu predviđene za određenu korisničku skupinu.

Vizualni identitet aplikacije trebao bi biti jednostavan, moderan i dosljedan u svim dijelovima sučelja, s fokusom na intuitivno korisničko iskustvo, posebno za korisnike koji nemaju iskustva u korištenju sličnih digitalnih alata. [5]

Aplikacija bi trebala omogućiti osnovnu analitiku i statistike za organizatore, poput broja pregleda događaja, rezervacija i povratnih informacija korisnika, čime se podržava daljnje unaprijeđenje usluga.

Sustav koristi EUR kao valutu prilikom rezervacije, mora jasno naznačiti fiksnu cijenu prilikom rezervacije, cijenu po osobi koja se mijenja u ovisnosti o broju gostiju i automatski izračunati i precizno predstaviti kupcu kako ne bi postojali nesporazumi prilikom rezervacije samih događaja.

# Arhitektura i dizajn sustava



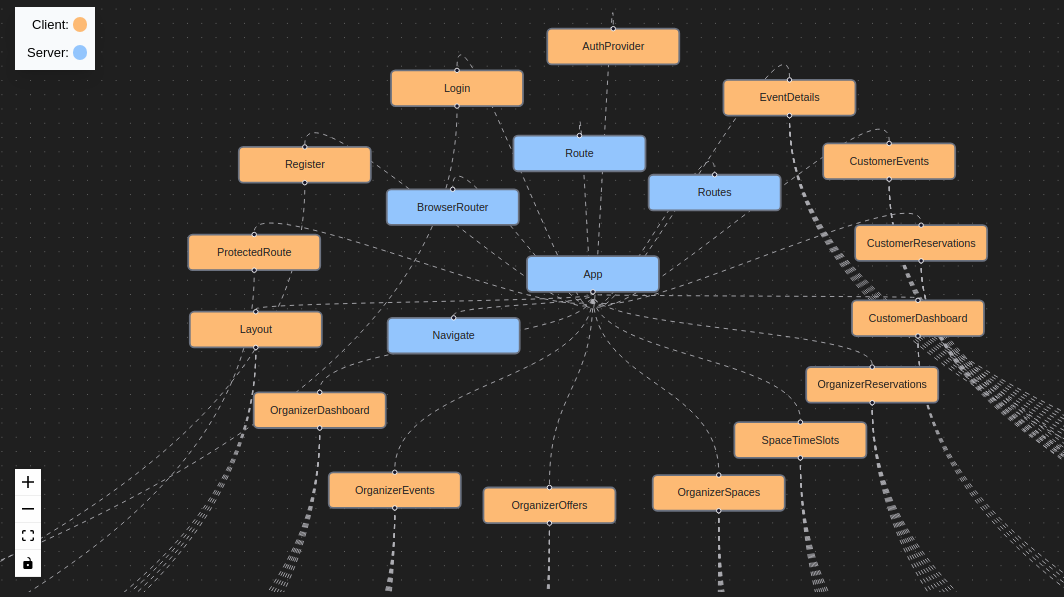
Slika 3.1 Arhitektura sustava

Web aplikacija sastoji se od korisničkog (frontend) sučelja, poslužiteljskog (backend) sučelja i baze podataka (database), koji zajedno čine cjelovit sustav za organizaciju događaja. Aplikacija je razvijena kao moderna web-aplikacija s jasno odvojenim slojevima prezentacije i logike, pri čemu se komunikacija između klijenta i poslužitelja odvija putem sigurnog **HTTPS protokola**, uz razmjenu podataka u **JSON formatu**. Svi zahtjevi prema poslužitelju koriste **Bearer token** autentifikaciju, čime se osigurava zaštita korisničkih podataka i sprječava neautorizirani pristup.

## ****Frontend sloj****

Korisničko sučelje implementirano je korištenjem **React** biblioteke u kombinaciji s **Vite** okruženjem za brzu i efikasnu izradu aplikacije. Za definiranje vizualnog izgleda korišten je **Tailwind CSS**, koji omogućuje modularno i responzivno oblikovanje sučelja. Cijeli frontend pisan je u **TypeScriptu**, čime je omogućena statička provjera tipova i smanjena mogućnost grešaka tijekom razvoja. Rad na korisničkom sučelju odvijao se u razvojnom okruženju **Visual Studio Code**, koje je omogućilo integraciju svih potrebnih alata i biblioteka. [9, 10]

Aplikacija je dizajnirana kao **Single Page Application (SPA)**, što znači da se većina sadržaja učitava dinamički bez potrebe za ponovnim učitavanjem stranice. React-ova komponentna arhitektura (slika 1.1) omogućila je jasno razdjeljivanje struktura i lakšu ponovnu upotrebu dijelova korisničkog sučelja. Podaci o trenutno prijavljenom korisniku, kao i privremeni podaci unutar interakcije s aplikacijom, pohranjuju se u **lokalnu pohranu preglednika (Local Storage)** ili **memoriju aplikacije**, ovisno o njihovoj vrsti i trajanju.

Sl. 3.2 (Vizualizacija react komponenata u frontendu koristeći React Labyrinth)

## ****Backend sloj****

Poslužiteljski dio aplikacije razvijen je korištenjem **Spring Boot** okvira, pisan u programskom jeziku **Java**, s upravljanjem projektom i ovisnostima putem **Maven** paket menađera. Backend je razvijan u **IntelliJ IDEA** razvojnom okruženju, koje je omogućilo napredno upravljanje projektima, debugiranje i integraciju s vanjskim servisima. [2, 8]

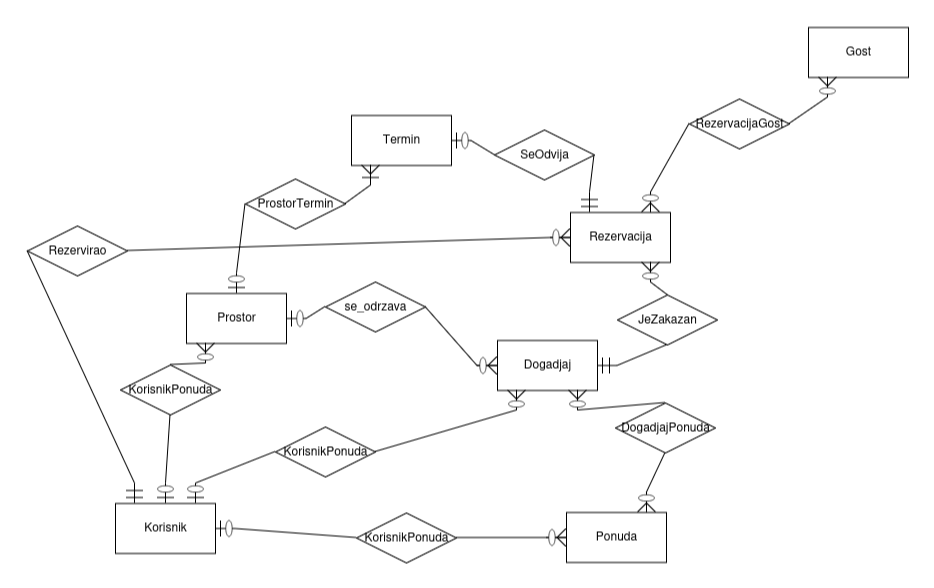
Backend arhitektura podijeljena je u tri glavna sloja:

* **Kontroleri (Controllers)**: odgovorni su za prihvat HTTP zahtjeva, definiranih putem REST API-ja. Ovi slojevi dekodiraju zahtjeve, validiraju ulazne podatke i prosljeđuju ih servisnom sloju.
* **Servisi (Services)**: u servisnom sloju nalazi se poslovna logika aplikacije. Tu se obrađuju pravila ponašanja poput validacije korisnika, provjere termina događaja ili obračuna provizije organizatora.
* **DTO objekti i pomoćne klase**: umjesto pune baze podataka, aplikacija bi koristila objekte prijenose podataka (Data Transfer Objects) i privremeno spremanje podataka u memoriju, čime se simulira protok informacija i omogućuje testiranje funkcionalnosti bez stvarnog mijenjanja objekta.

Svi zahtjevi između klijenta i servera koriste **autorizacijski token (JWT Bearer token)**, koji se šalje u zaglavlju svakog zahtjeva radi provjere identiteta korisnika i očuvanja sigurnosti. Također, svi osjetljivi podaci šalju se i primaju putem **enkriptiranih HTTPS konekcija**, čime se zadovoljavaju sigurnosni standardi suvremenih web aplikacija. [6, 12]

Aplikacija je razvijena s fokusom na modularnost i proširivost, čime se omogućava daljnja nadogradnja – primjerice, kasnije povezivanje s bazom podataka, vanjskim API-jima (npr. za plaćanje), upravljanje korisničkim ulogama i administracijom.

## Model Baze podataka

Sl 3.3 ER Model

### Entitet Dogadjaj

U entitetu Dogadjaj pohranjuju se svi potrebni podaci vezani za događaj koji organizira korisnik. Svaki događaj može imati fiksnu cijenu ili cijenu po osobi, pripada određenom prostoru i povezan je s korisnikom sustava.

PK = {dogadjaj\_id}

Tablica 1 Entitet Dogadjaj

|  |  |  |
| --- | --- | --- |
| Naziv atributa | Opis atributa | Tip podatka |
| dogadjaj\_id | Identifikator događaja | INT |
| naziv | Naziv događaja | VARCHAR(255) |
| opis | Opis događaja | TEXT |
| korisnik\_id | Organizator događaja | INT |
| prostor\_id | Prostor u kojem se održava | INT |
| uk\_cijena\_po\_osobi | Ukupna cijena po osobi | NUMERIC(10,2) |
| uk\_cijena\_fiksna | Ukupna fiksna cijena | NUMERIC(10,2) |
| otkazni\_rok | Rok za otkazivanje (u danima) | INT |
| slika\_url | URL slike događaja | VARCHAR(255) |

### Entitet Dogadjaj\_Ponuda

Entitet Dogadjaj\_Ponuda predstavlja vezu između događaja i ponude (many-to-many). Svaki zapis povezuje jedan događaj s jednom ponudom.

PK = {dogadjaj\_id, ponuda\_id}

Tablica 2 Entitet Dogadjaj\_Ponuda

|  |  |  |
| --- | --- | --- |
| Naziv atributa | Opis atributa | Tip podatka |
| dogadjaj\_id | Događaj | INT |
| ponuda\_id | Povezana ponuda | INT |

### Entitet Gost

Entitet Gost pohranjuje informacije o gostima koji su dio rezervacije za određeni događaj.

PK = {gost\_id}

Tablica 3 Entitet Gost

|  |  |  |
| --- | --- | --- |
| Naziv atributa | Opis atributa | Tip podatka |
| gost\_id | Identifikator gosta | INT |
| rezervacija\_id | Povezana rezervacija | INT |
| ime | Ime gosta | VARCHAR(255) |
| prezime | Prezime gosta | VARCHAR(255) |
| email | Email adresa gosta | VARCHAR(255) |

### Entitet Korisnik

Entitet Korisnik sadrži sve podatke o korisnicima sustava, uključujući login podatke i ulogu (organizator ili kupac).

PK = {korisnik\_id}

Tablica 4 Entitet Korisnik

|  |  |  |
| --- | --- | --- |
| Naziv atributa | Opis atributa | Tip podatka |
| korisnik\_id | Identifikator korisnika | INT |
| username | Korisničko ime | VARCHAR(255) |
| ime | Ime korisnika | VARCHAR(255) |
| prezime | Prezime korisnika | VARCHAR(255) |
| email | Email adresa | VARCHAR(255) |
| password | Lozinka korisnika | VARCHAR(255) |
| uloga | Uloga korisnika | VARCHAR(255) |

### Entitet Ponuda

Entitet Ponuda sadrži sve dostupne ponude (usluge, rekviziti itd.) koje se mogu uključiti u događaj. Svaka ponuda ima svoj tip cijene i pripada jednoj kategoriji.

PK = {ponuda\_id}

Tablica 5 Entitet Ponuda

|  |  |  |
| --- | --- | --- |
| Naziv atributa | Opis atributa | Tip podatka |
| ponuda\_id | Identifikator ponude | INT |
| naziv | Naziv ponude | VARCHAR(255) |
| opis | Opis ponude | TEXT |
| cijena | Cijena ponude | NUMERIC(10,2) |
| tip\_cijene | Tip cijene (fiksno/po osobi) | VARCHAR(255) |
| kategorija | Kategorija ponude | VARCHAR(255) |
| korisnik\_id | Korisnik koji je dodao ponudu | BIGINT |

### Entitet Prostor

Entitet Prostor sadrži podatke o prostorima u kojima se mogu organizirati događaji. Svaki prostor ima svoju cijenu, adresu i kapacitet.

PK = {prostor\_id}

Tablica 6 Entitet Prostor

|  |  |  |
| --- | --- | --- |
| Naziv atributa | Opis atributa | Tip podatka |
| prostor\_id | Identifikator prostora | INT |
| naziv | Naziv prostora | VARCHAR(255) |
| opis | Opis prostora | TEXT |
| cijena | Cijena najma | NUMERIC(10,2) |
| adresa | Adresa prostora | TEXT |
| kapacitet | Broj osoba koji stanu | INT |
| korisnik\_id | Vlasnik prostora | BIGINT |

### Entitet Rezervacija

Entitet Rezervacija pohranjuje sve podatke o rezervacijama koje korisnici izvrše za određene događaje u točno određenim terminima i prostorima. Rezervacija uključuje ukupnu cijenu, moguće dodatne zahtjeve korisnika i status rezervacije. Povezana je s korisnikom koji rezervira, događajem koji se rezervira te terminom u kojem se on održava.

**PK = {rezervacija\_id}**

Tablica 9. entitet rezervacija

| Atribut | Opis | Tip podatka |
| --- | --- | --- |
| rezervacija\_id | Jedinstveni identifikator rezervacije | INTEGER |
| korisnik\_id | Korisnik koji je napravio rezervaciju | INTEGER (FK) |
| dogadjaj\_id | Događaj koji se rezervira | INTEGER (FK) |
| prostor\_termin\_id | Termin i prostor rezervacije | INTEGER (FK) |
| ukupna\_cijena | Ukupna cijena rezervacije | NUMERIC(10,2) |
| posebni\_zahtjevi | Dodatni zahtjevi korisnika | TEXT |
| status | Status rezervacije (npr. aktivna, otkazana) | VARCHAR(255) |

### ENUM tablice

Korištene ENUM tablice u ovome radu su:

status rezervacije, tip cijene i uloga korisnika:

CREATE TYPE public.status\_rezervacije\_enum AS ENUM (

'CEKA\_POTVRDU',

'PLACENO',

'OTKAZANO'

);

CREATE TYPE public.tip\_cijene\_enum AS ENUM (

'FIKSNO',

'PO\_OSOBI'

);

CREATE TYPE public.uloga\_enum AS ENUM (

'ORGANIZATOR',

'KUPAC'

);

Kod 1.1 SQL kreiranje ENUM tablicama

One se koriste kako bi se preciziralo stanje tih vrijednosti određene u bazi podataka.

Prilikom kreiranja rezervacije, status u objektu rezervacija se postavlja status „NA\_CEKANJU”, čeka akciju kupca tako što će ili platiti pa postaviti status „PLACENO” ili otkazati pa postaviti status „OTKAZANO”. Status se sam mijenja u otkazano ukoliko je prošao otkazni rok a rezervacija nije bila plaćena. Logika upravljanja tog statusa rezervacije je osigurana u bazi sa sljedećom funkcijom koja se poziva periodički. (vidi kod 1.2) ali isto tako je i u poslužiteljskom sloju osigurana provjera otkaznog roka.

CREATE OR REPLACE FUNCTION provjeri\_i\_otkazi\_rezervacije()

RETURNS void AS $$

BEGIN

UPDATE rezervacija r

SET status = 'OTKAZANO'

FROM dogadjaj d

JOIN termin t ON t.termin\_id = r.prostor\_termin\_id

WHERE r.dogadjaj\_id = d.dogadjaj\_id

AND r.status != 'OTKAZANO'

AND CURRENT\_DATE > (t.datum\_pocetka::date - d.otkazni\_rok);

END;

$$ LANGUAGE plpgsql;

Kod 1.2 Osiguranje

### Restrikcije baze podataka

Kako bi se osigurala baza podataka potrebno je osigurati da je isključivo korisnik kojem je uloga „Organizator” omogućeno kreiranje događaja, isključivo korisnik kojem je uloga „Kupac” omogućeno kreiranje rezervacija.

To je postignuto tehnikom TRIGGER-a (vidi u nastavnku kod 1.3)

CREATE OR REPLACE FUNCTION provjeri\_ulogu\_za\_dogadjaj()

RETURNS trigger AS $$

DECLARE

uloga\_korisnika uloga\_enum;

BEGIN

SELECT uloga INTO uloga\_korisnika FROM korisnik WHERE korisnik\_id = NEW.korisnik\_id;

IF uloga\_korisnika != 'ORGANIZATOR' THEN

RAISE EXCEPTION 'Samo korisnik s ulogom ORGANIZATOR može kreirati događaj.';

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION provjeri\_ulogu\_za\_rezervaciju()

RETURNS trigger AS $$

DECLARE

uloga\_korisnika uloga\_enum;

BEGIN

SELECT uloga INTO uloga\_korisnika FROM korisnik WHERE korisnik\_id = NEW.korisnik\_id;

IF uloga\_korisnika != 'KUPAC' THEN

RAISE EXCEPTION 'Samo korisnik s ulogom KUPAC može kreirati rezervaciju.';

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

Kod 1.3 SQL ograničenje kreiranja

# Implementacija i korisničko sučeljem

## Korišteni alati i tehnologije.

Tijekom razvoja projekta korišten je niz suvremenih alata i tehnologija koji su omogućili učinkovit i strukturiran razvoj softverskog rješenja.

Za upravljanje izvornim kodom korišten je sustav za kontrolu verzija **Git**, dok se udaljeni repozitorij nalazi na platformi **GitHub**, čime je omogućena suradnja i unaprijeđivanje koda u realnom vremenu. [11]

Za razvoj poslužiteljskog dijela aplikacije korišteno je razvojno okruženje **IntelliJ IDEA**, koje nudi širok spektar značajki za rad s Java okvirom **Spring Boot**. Inicijalna konfiguracija i generiranje strukture Spring Boot projekta provedeni su pomoću alata **Bootify.io**, čime je ubrzan početni korak razvoja backenda. [13]

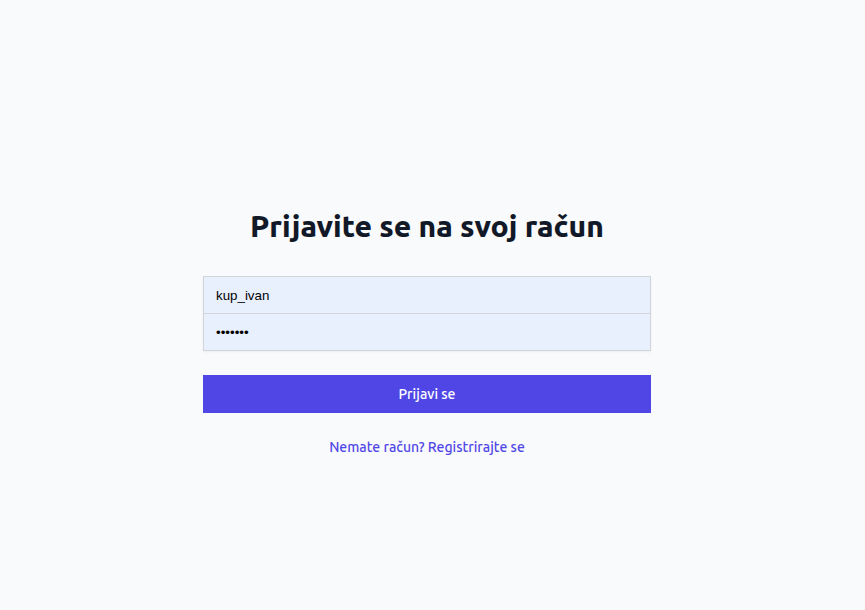
Korisničko sučelje aplikacije izrađeno je koristeći **React** biblioteku u kombinaciji s alatima **Vite** i **Tailwind CSS**. Razvoj frontenda odvijao se u integriranom razvojnom okruženju **Microsoft Visual Studio Code**, koje je dodatno prošireno korištenjem plugina **React Labirinth** za vizualizaciju komponenti React aplikacije, čime je osigurana bolja preglednost i organizacija korisničkog sučelja.

Za upravljanje relacijskom bazom podataka korišten je sustav **PostgreSQL**, dok je za administraciju i vizualizaciju baze upotrebljen alat **pgAdmin**. Modeliranje entitetsko-relacijskog modela provedeno je pomoću web alata **ERDPlus**, koji je omogućio jednostavno i pregledno definiranje odnosa među entitetima baze podataka. [7] [14] [3]

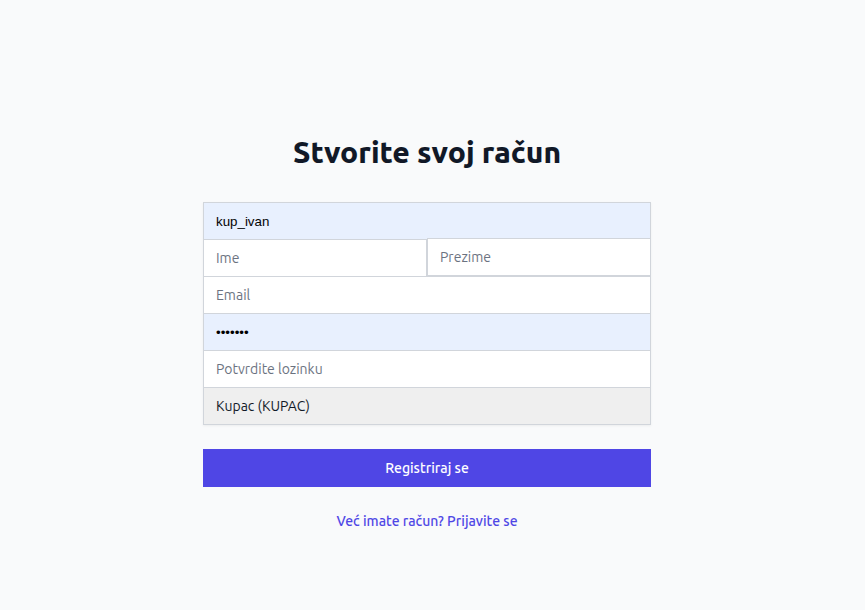
Dodatno, za generiranje vizualnih sučelja korišten je alat **Intelj Scheme**, koji je poslužio kao podrška u dizajniranju elemenata korisničkog iskustva.

## Korisničke upute

Na početnoj (login) stranici nalaze se upute za login i potencijalnu registraciju.

Slika 4.1 Login forma

ako kliknemo na „Nemate račun?, Registrirajte se” odvest će nas na ovu pod stranicu gdje se možemo registrirati i odabrati našu ulogu.

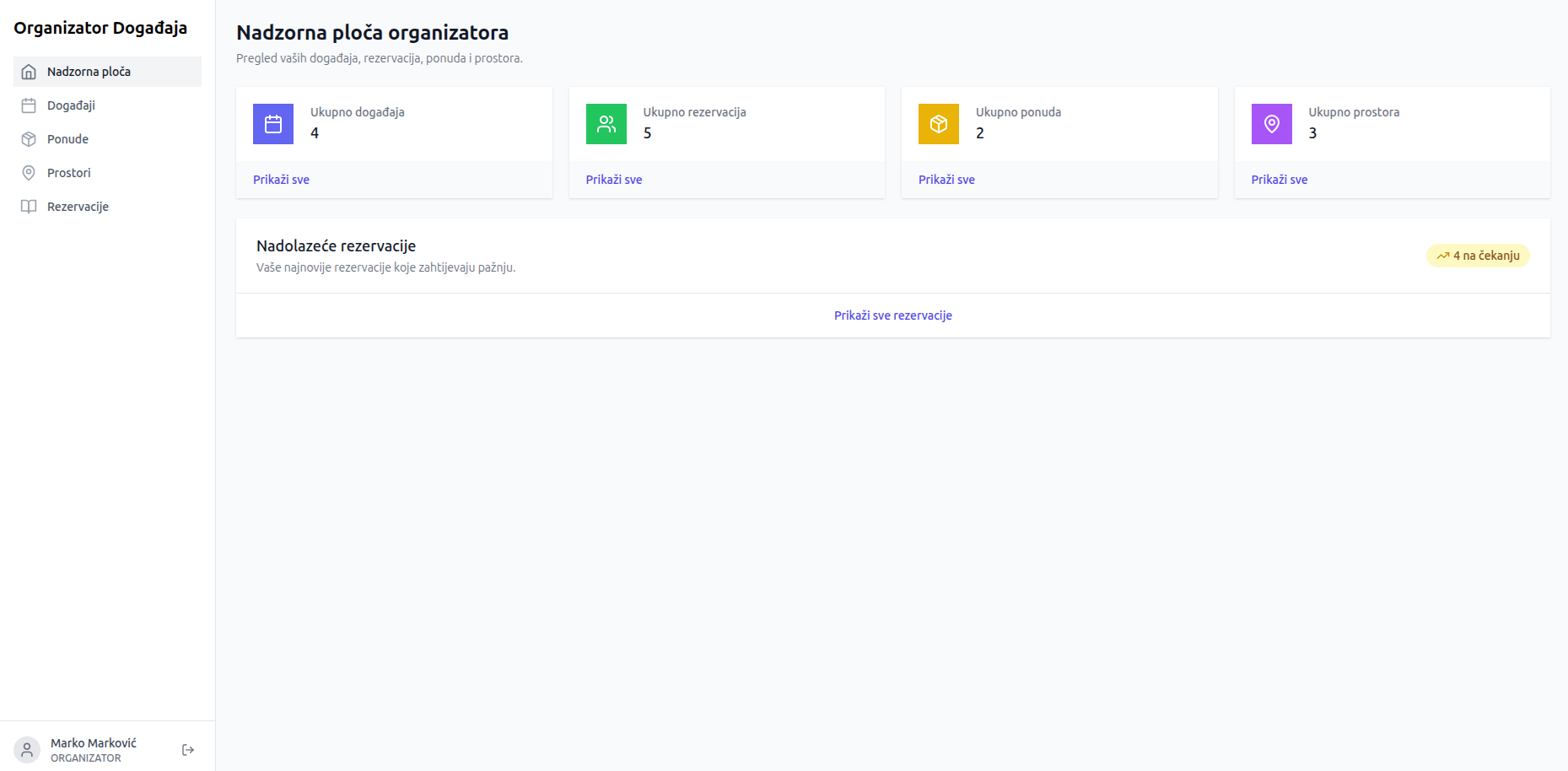


Slika 4.2 Registracijska forma

Unutar same aplikacije postoje dva sučelja to jest, dva načina rada. S obzirom na ulogu (Organizator, kupac) različit će biti prikaz stranice. Prvo ćemo pokazati upute korištenja organizatora pa onda tek kupca.

## ORGANIZATOR

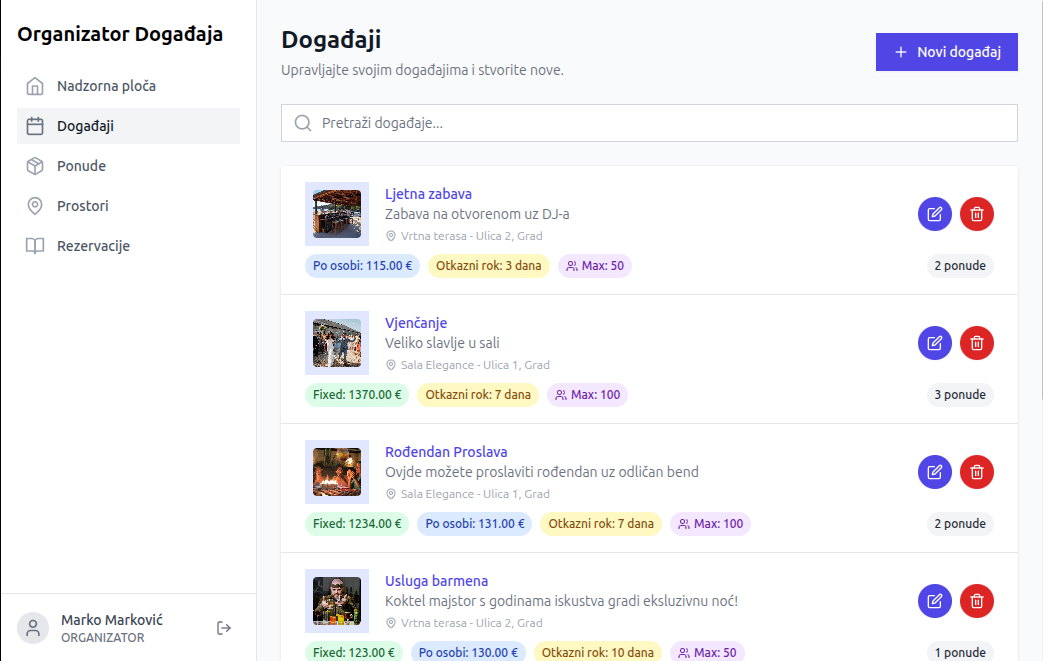
Kada se prijavimo kao organizator u sustav prvo sučelje na koje nailazimo jest nadzorna ploča (dashboard) na kojem vidimo brzu statistiku našeg korisničkog profila te osnovne podatke o količini rezervacija, ponuda, prostora te brze linkove do svakog sučelja tih elementa. Sa lijeve strane se nalazi hamburger izbornik koji pregledno sadržava sve glavne podstranice koje su potrebne organizatoru.



Slika 4.3 Organizator nadzorna ploča

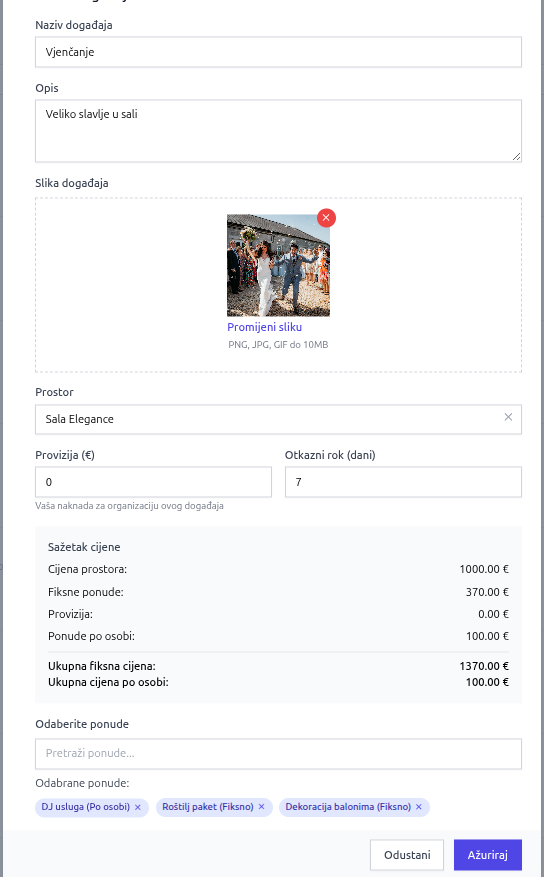
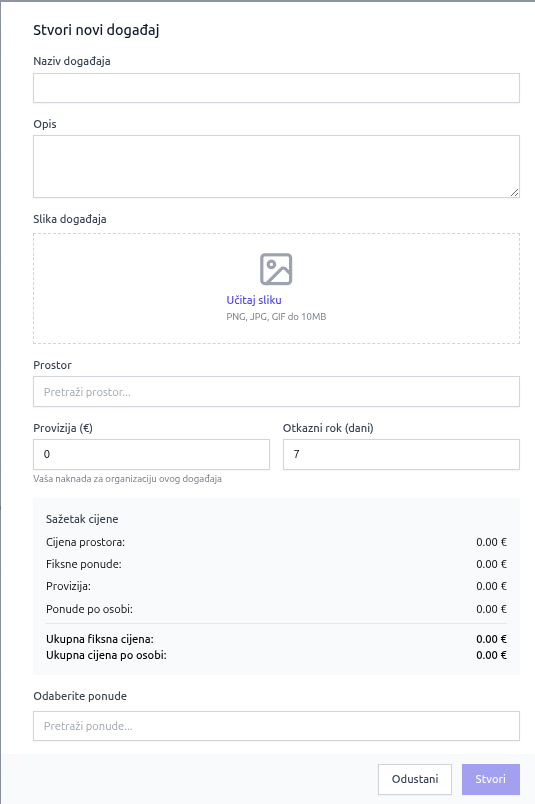
Organizator kada ode na master-detail prikaz događaja pojavi mu se popis svih nedavnih njegovih događaja te sažeti detalji o svakom na pojedinom događaju.

U gornjem desnom uglu postoji gumb sa kojim može dodavati nove događaje.

Slika 4.4 Organizator pregled događaja

Kada ide kreirati novi događaj ima pogled na formu kojom kreira događaje u kojoj mora odabrati prostor, svoju komisiju ali i može odabrati sliku, ponude i promijeniti otkazni rok kojem je zadana vrijednost postavljena na 7 dana prije termina u kojem je rezervacija kupca. Ponude i prostori se traže pomoću integrirane tražilice koja koristi **Js querry** biblioteku koja pretražuje pomoću ključnih riječi prostore i ponude. Također je na isti način moguće pretraživati događaje iz pregleda događaja (slika 4.4)

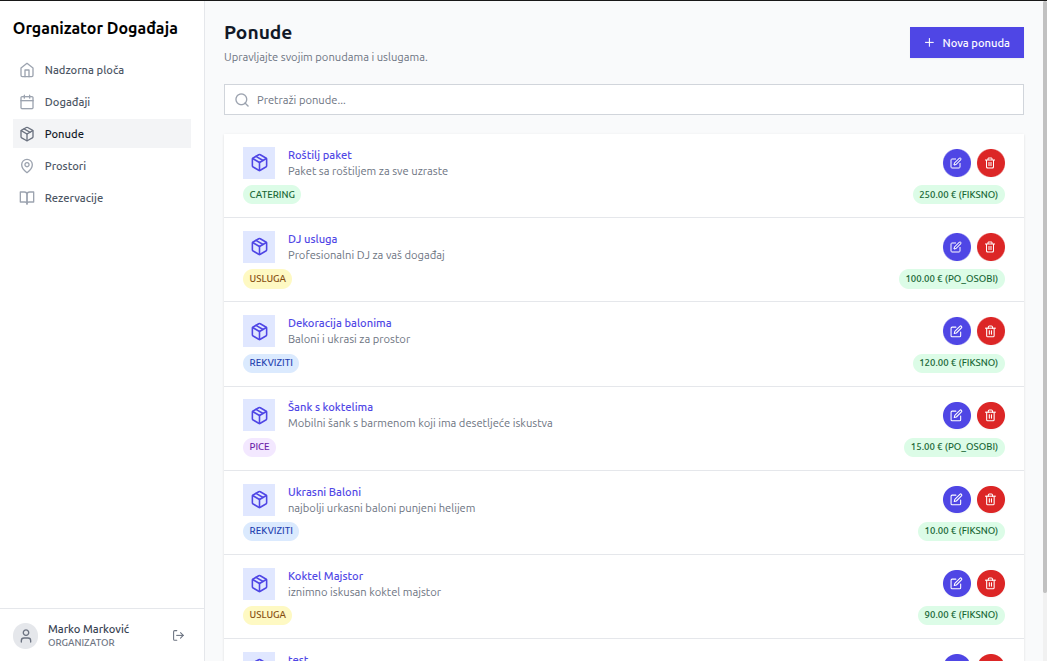
Nakon odabira svih stavki za novi događaj unutar same forme mu se automatski računa cijena koja je fiksna i cijena po osobi koja će se mijenjati ovisno o broju gostiju koje kupac prenamijeni za specifičnu rezervaciju. Također je moguće sve događaje urediti i izbrisati klikom na gumbe koji se nalaze pored svakog događaja.

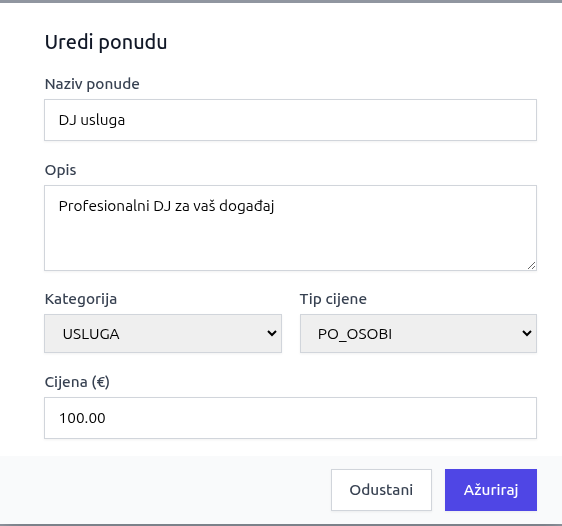
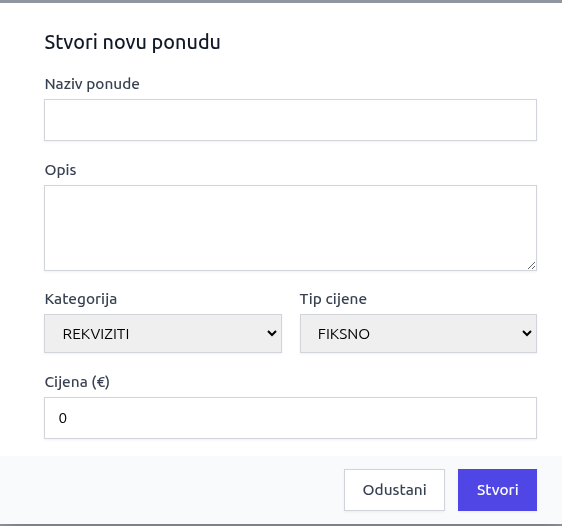


Slika 4.5, 4.6 Forma za dodavanje, uređivanje događaja (organizator)

Na sličan način moguće je dodavanje, uređivanje i brisanje ponude.

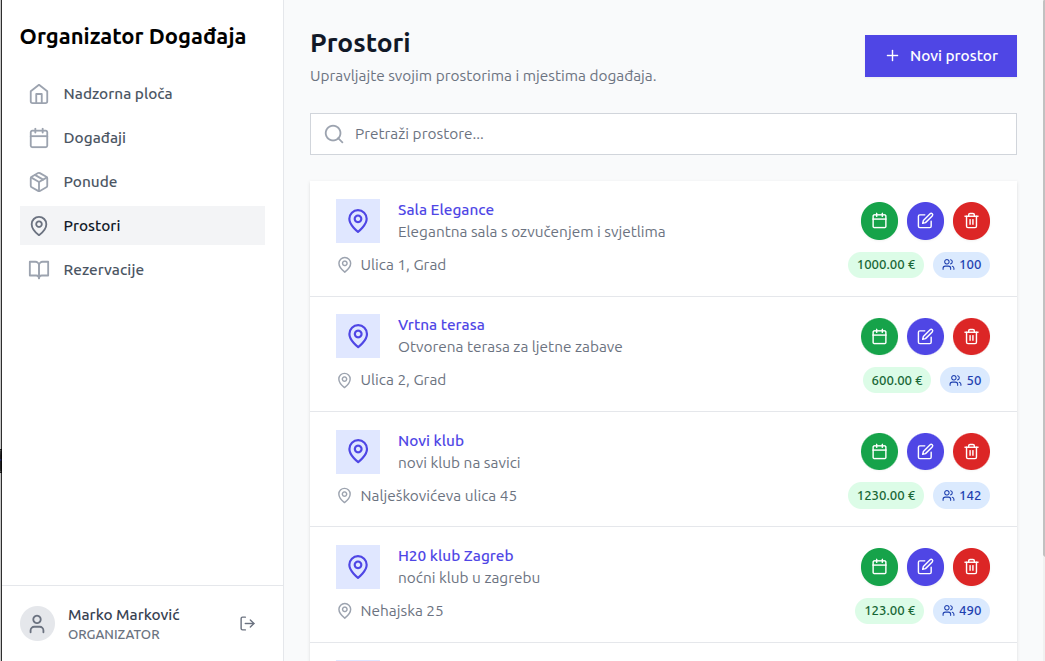
Kod ponude se bira tip cijene, odnosno naplaćuje li se navedena ponuda po osobi ili jednokratno (fiksno) padajućim izbornikom. Naravno naziv, opis i sam iznos cijene se mora također postaviti.

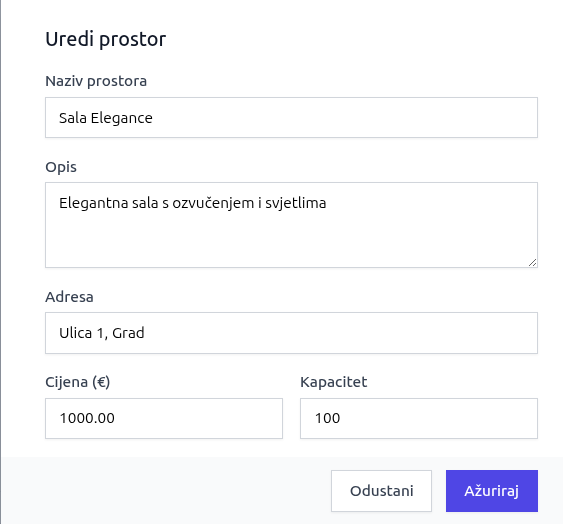
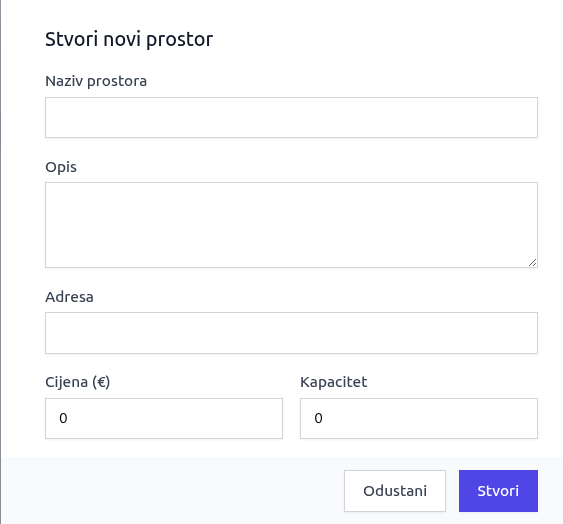
Slika 4.7 Pregled ponude



Slika 4.8, 4.9 Stvaranje ponude, uređivanje ponude

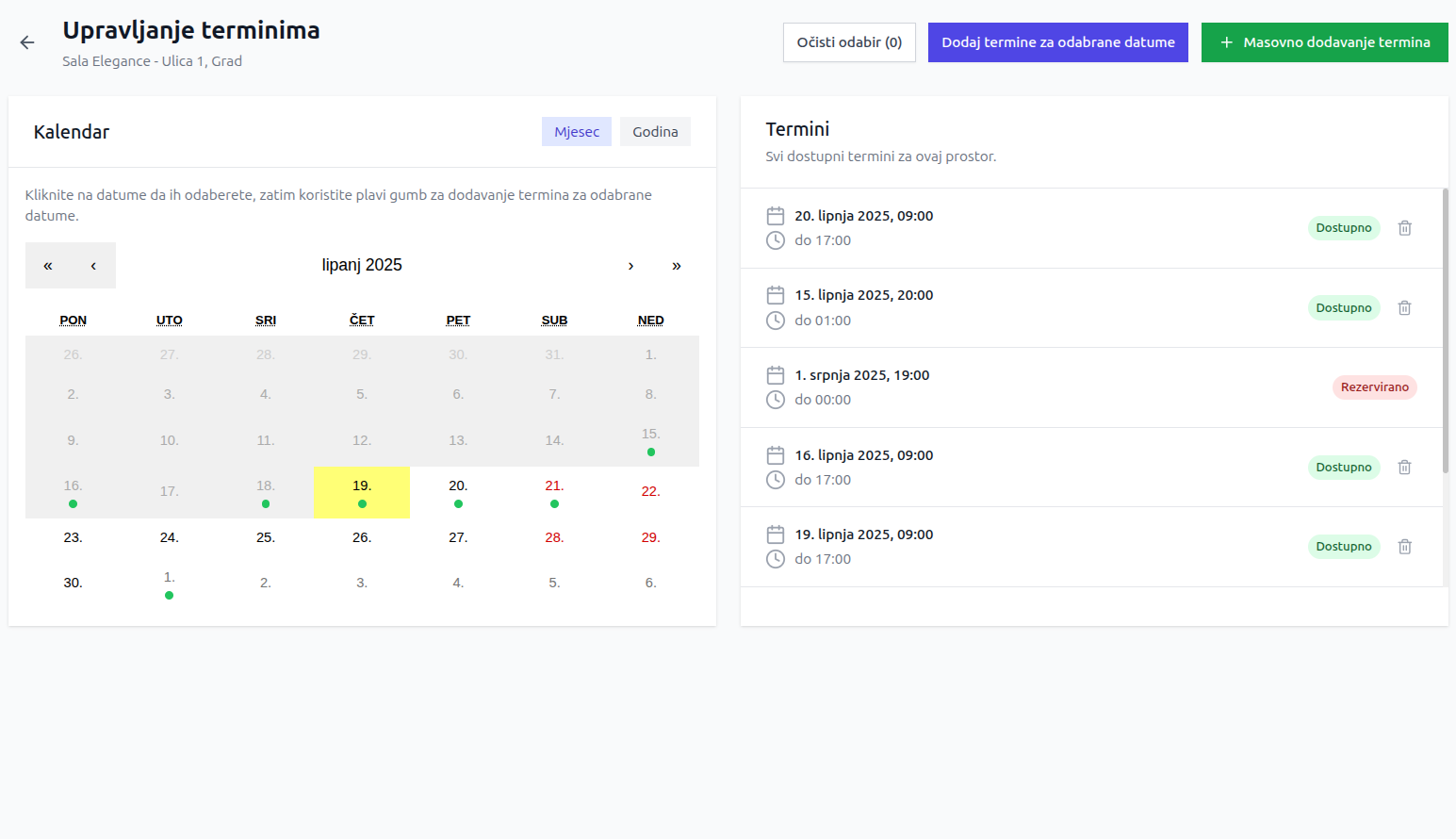
Također na sličan način se dodaju, kreiraju i brišu prostori samo će cijena uvijek biti jednokratna.

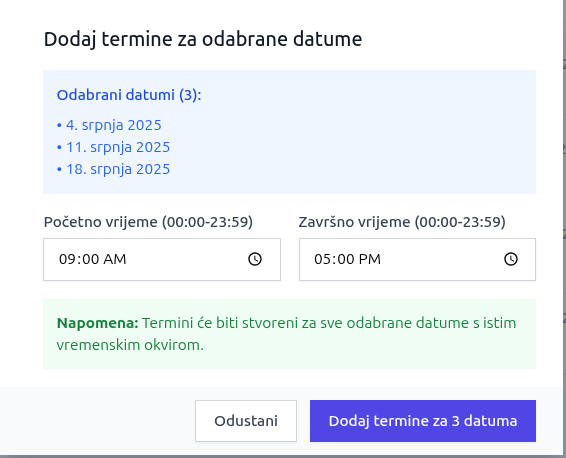
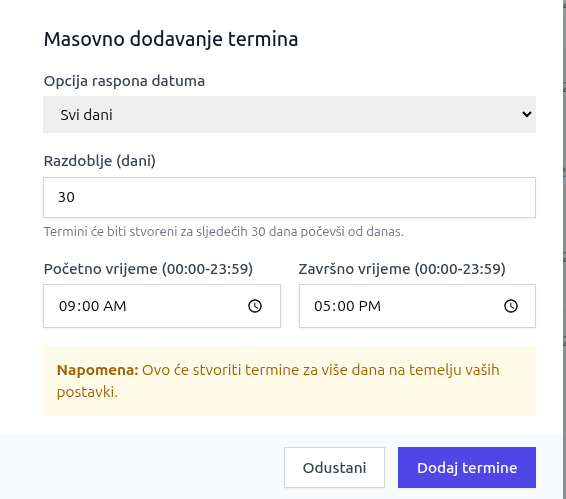
Slika 4.10 Pregled prostora



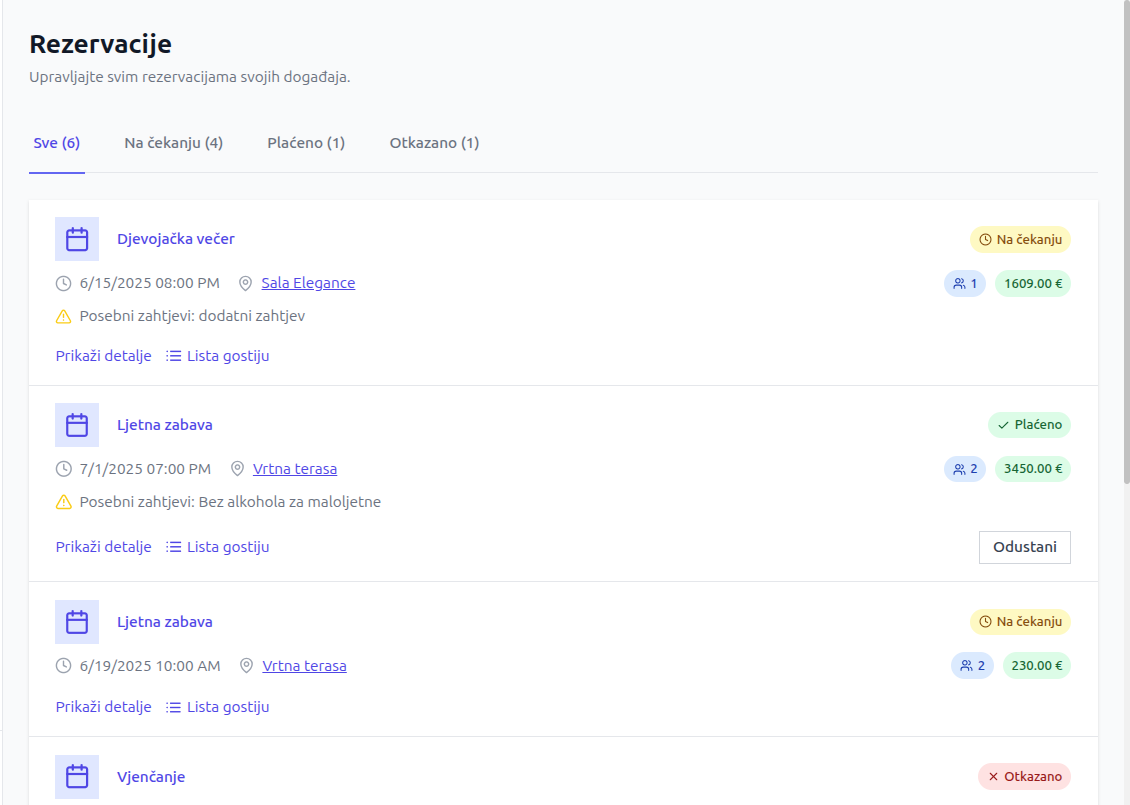
Slika 4.11, 4.12 Pregled formi za stvaranje i uređivanje prostora

Razlika između pregleda prostora i svih ostalih komponenti ovog sustava jest što je omogućeno pored svakog prostora dodavati termine za pojedini prostor. Ovaj sustav dopušta organizatoru da kreira vremenske periode unutar kojih prostori rade (moguće je kreirati rezervaciju). Unutar kreiranog događaja kupac može rezervirati taj događaj isključivo u terminima u kojima je prostor slobodan. Termini se rade tako što se selektiraju datumi unutar prikaza kalendara na sučelju za upravljanjem vremenskim periodima te se dodaju naknadno klikom na gumb u gornjem desnom uglu. S obzirom da bi jedan organizator u prosjeku htio cijeli jedan mjesec ili čak vremenski dulji period postaviti događaj omogućeno je i masovno dodavanje termina što znači da može odabrati opciju svaki dan, samo vikendi i samo radni dani te po tom predlošku kreirati više termina sa manje uloženim trudom. Vidi slike 4.13., 4.14, 4.15

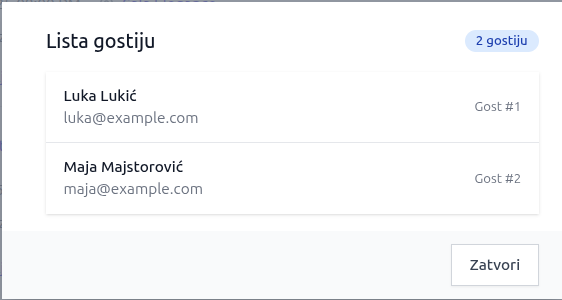
Slika 4.13 Pregled termina organizator

Slika 4.14, 4.15 Dodavanje termina unutar selektiranih datuma, masovno dodavanje termina

I zadnji prikaz organizatora jest rezervacije, odnosno sve rezervacije kupaca koje su vezane za njegove događaje.

Slika 4.16 Prikaz rezervacija organizator

Unutar svake rezervacije pišu osnovni detalji, sami termin te na desnoj strani i status same rezervacije. Gore postoji izbornik gdje organizator može vidjeti prošle rezervacije te kojima je stanje „Na čekanju, Plaćeno i Otkazano”. Također mu je dana mogućnost otkazivanja plaćenih narudžbi ukoliko se dogodi neočekivani događaj sa strane kupca, odnosno njegove strane. Pored svake rezervacije može vidjeti listu gostiju te shodno tome bolje znati kako upravljati navedenom rezervacijom.

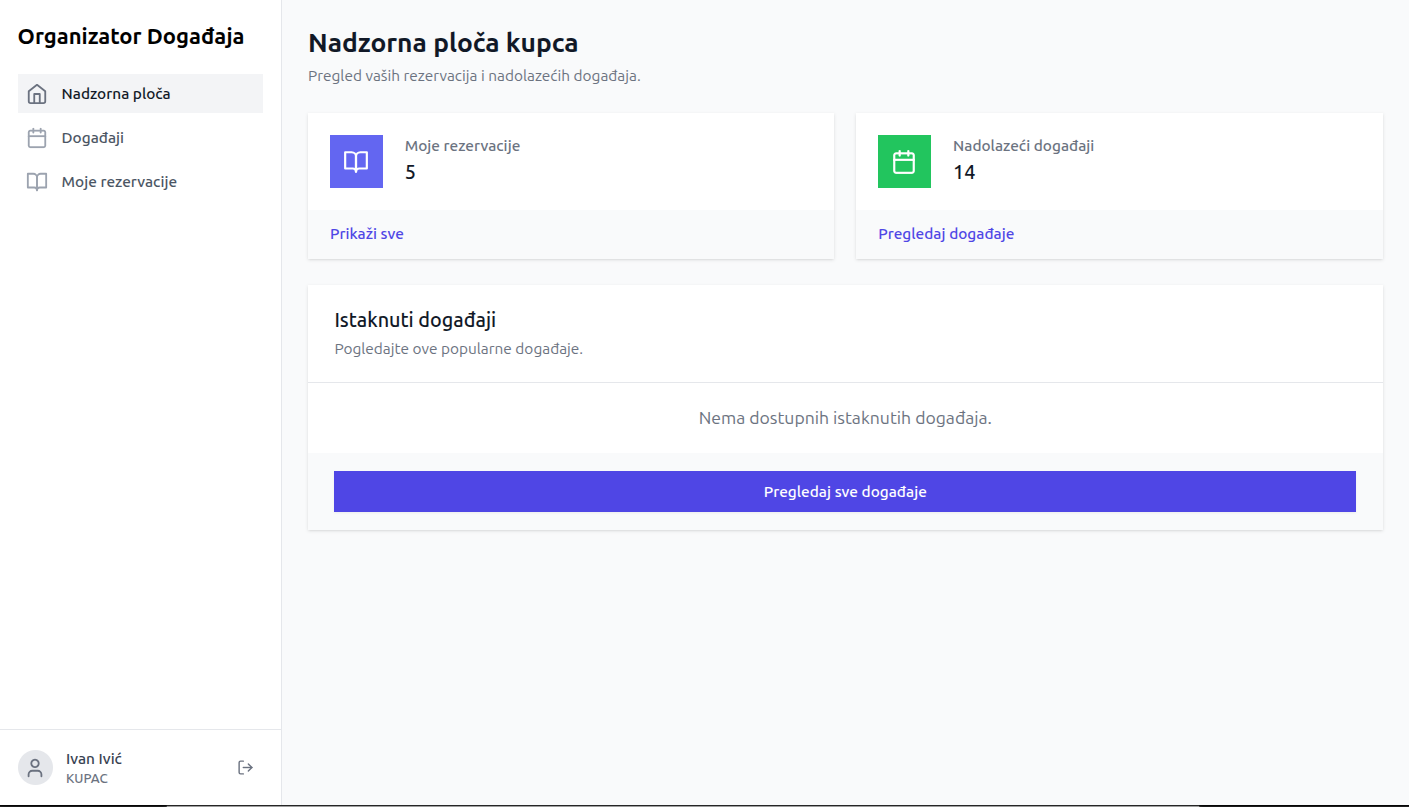


Slika 4.17 Lista gostiju

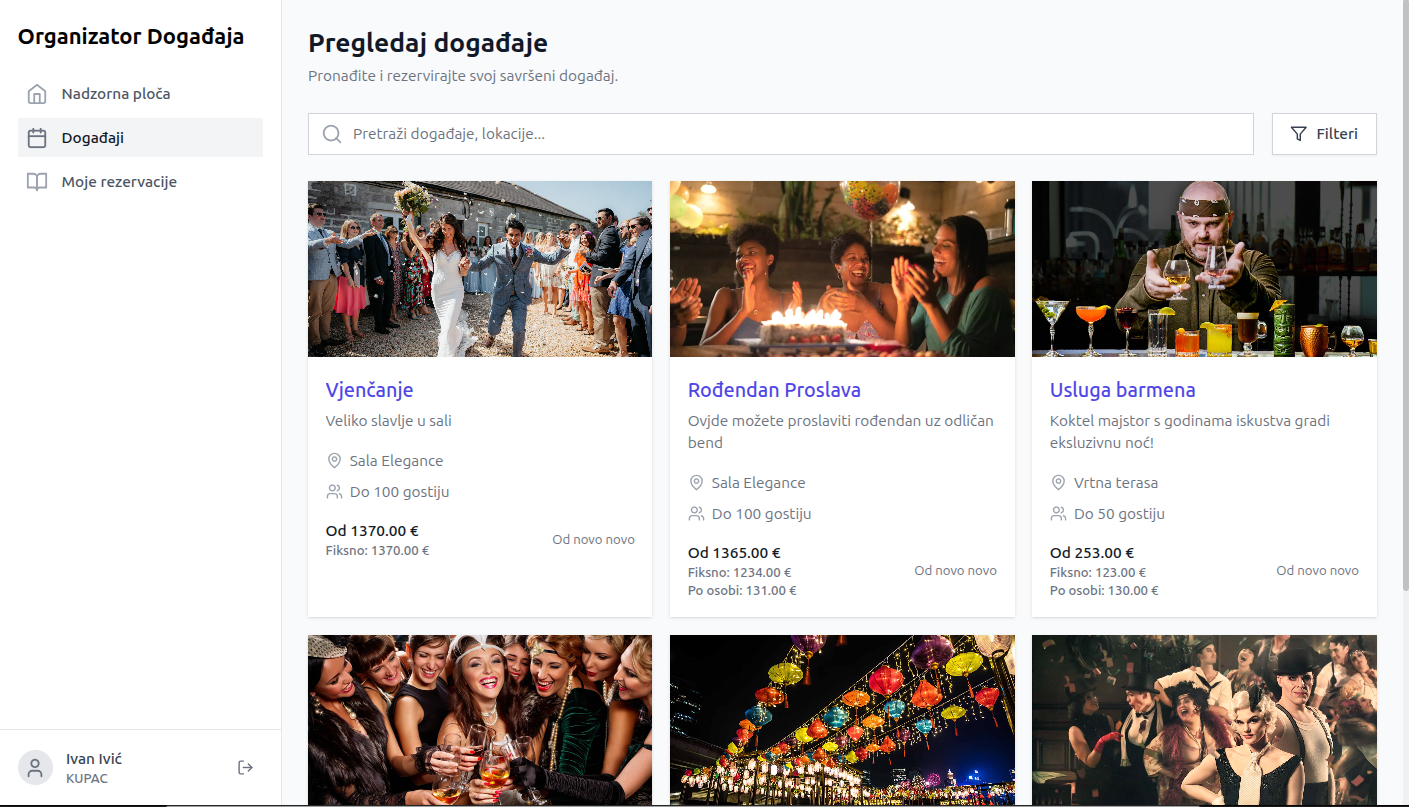
## KUPAC

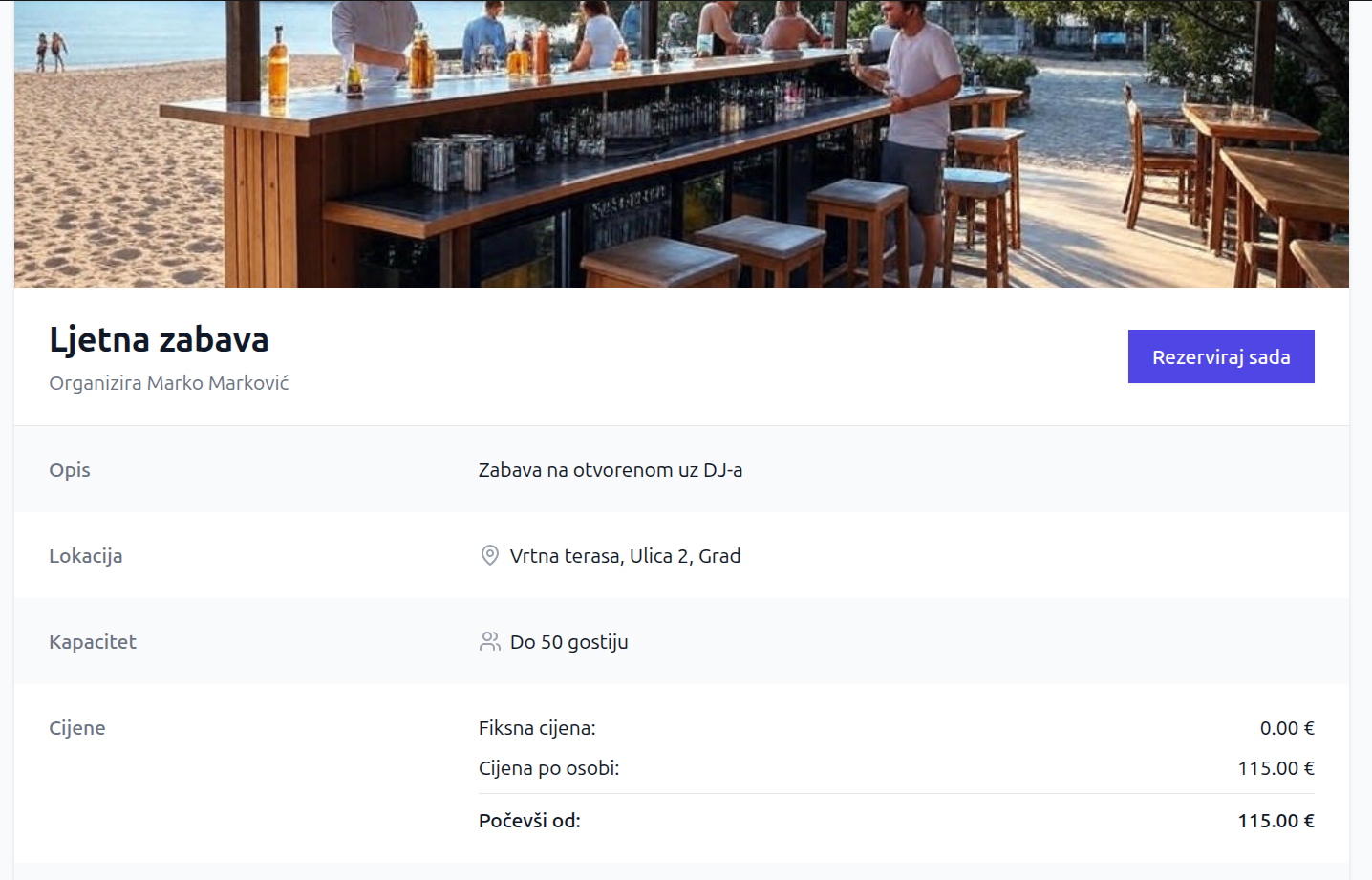
Sada ćemo predstaviti kupčev prikaz.

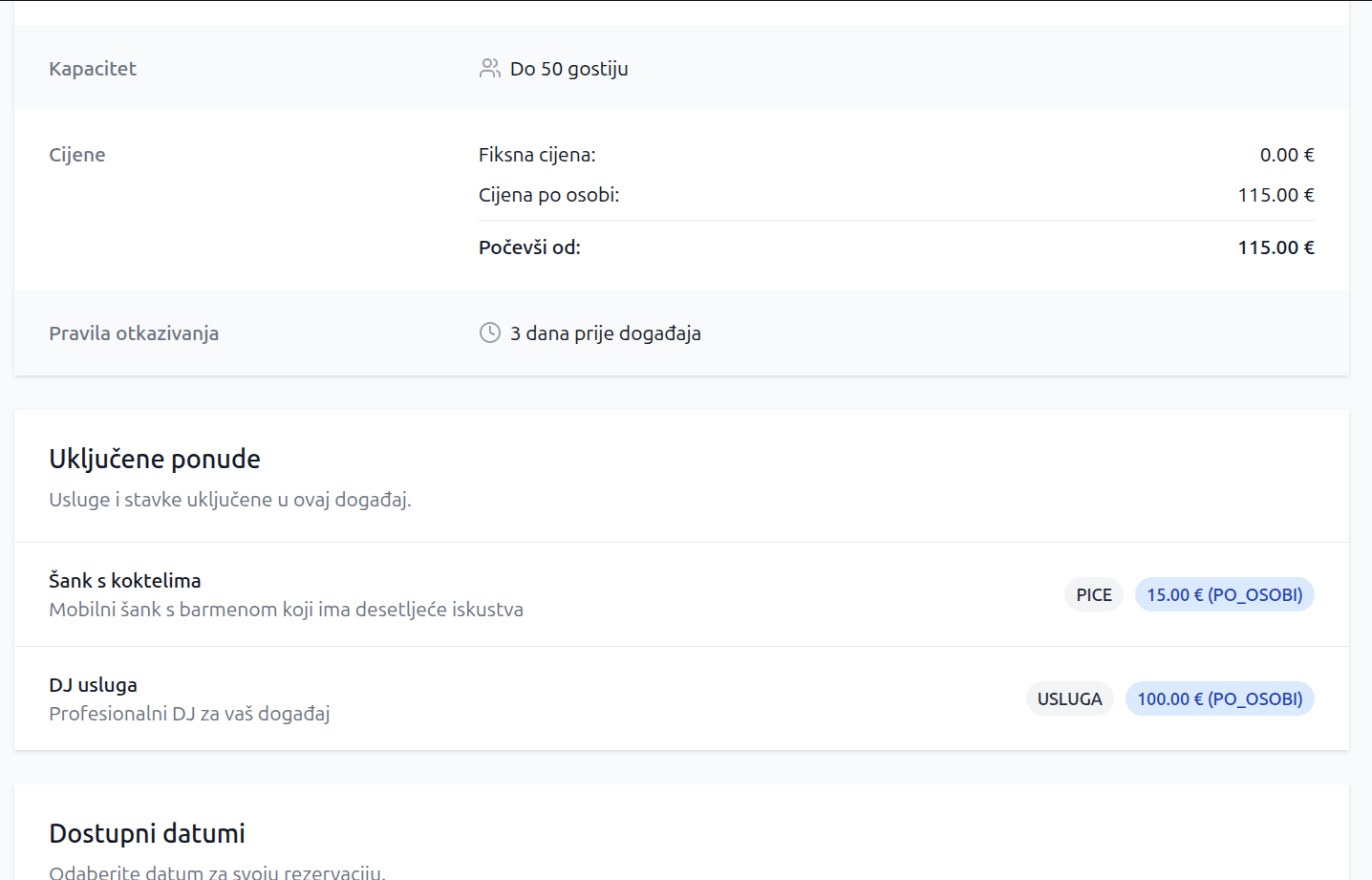
Nakon prijave u sustav dočekuje ga nadzorna ploča sa malo drugačijim izgledom u odnosu na organizatora. Početni dizajn je isti no nadzorna ploča sadrži statistiku koja je bitna kupcu te hamburger izbornik ima komponente koje su potrebne samo kupcu kao što je broj njegovih rezervacija i broj nadolazećih događaja od njegovo zadnjeg pristupa stranici, odnosno njegove zadnje prijave u sustav.

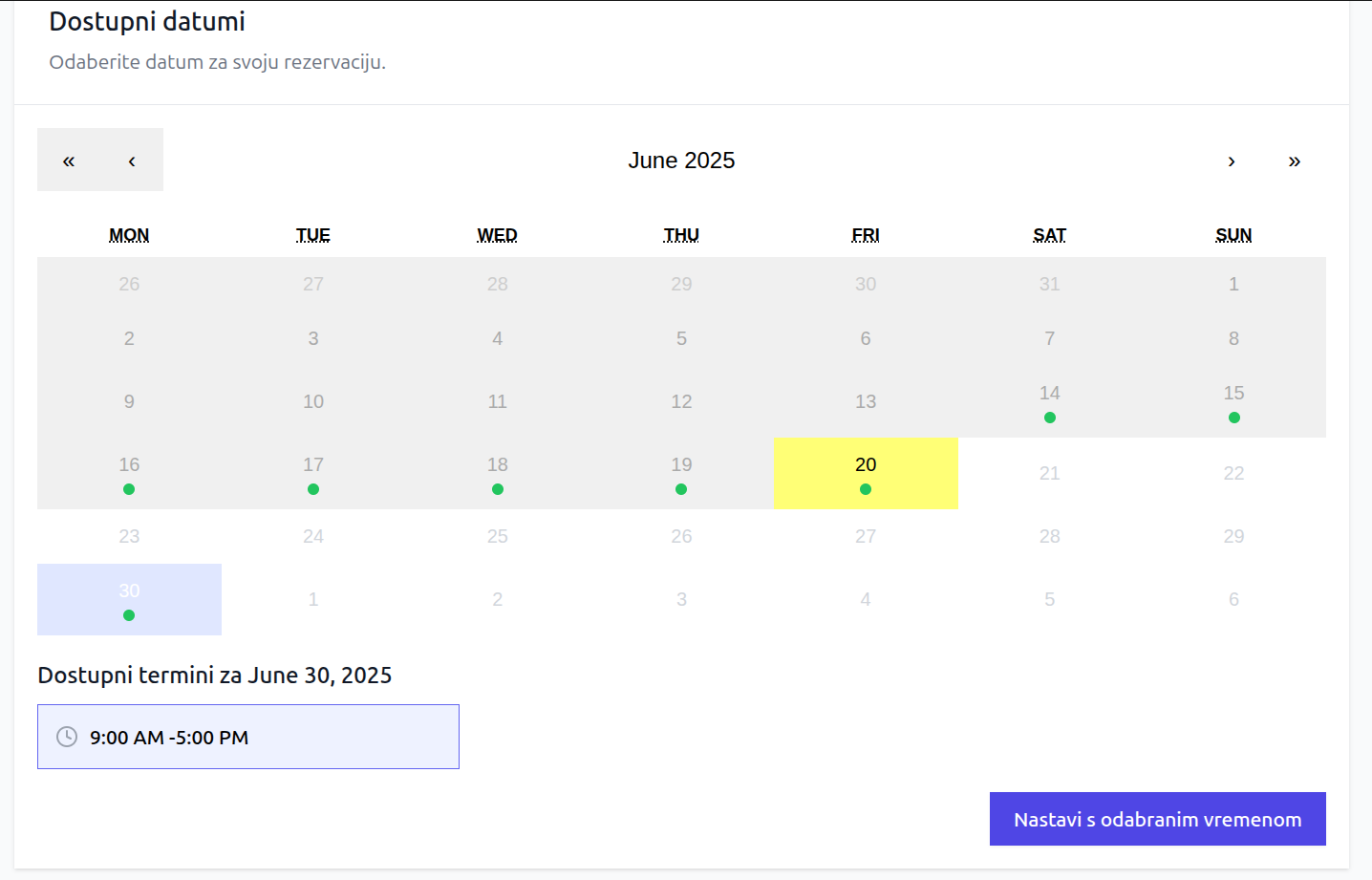
Slika 4.18 Nadzorna ploča kupca

Nakon što je preusmjeren na podstranicu prikaza događaja gumbom nadzorne ploče ili hamburger izbornikom može pregledavati događaje te rezervirati iste. Kupac također ima opciju filtriranja ako si želi ubrzati proces pronalaženja idealnog događaja po kapacitetu i cijeni događaja.

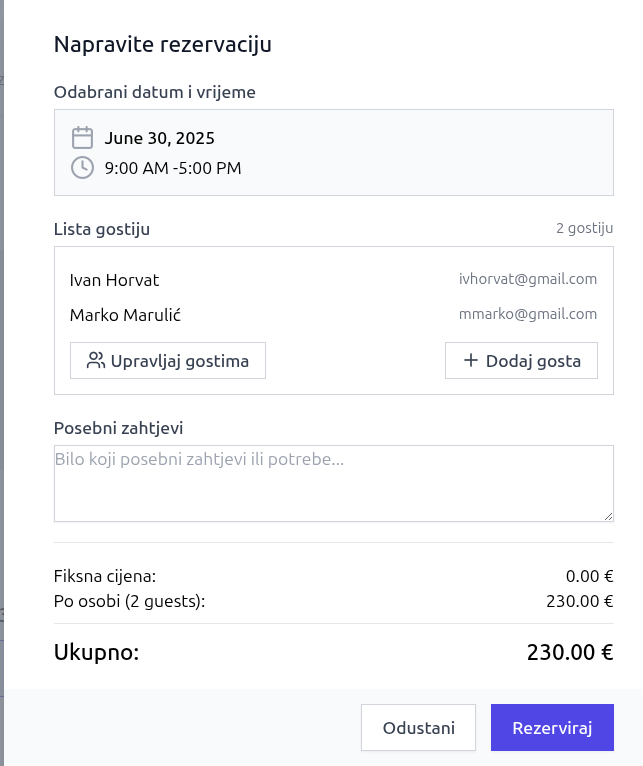
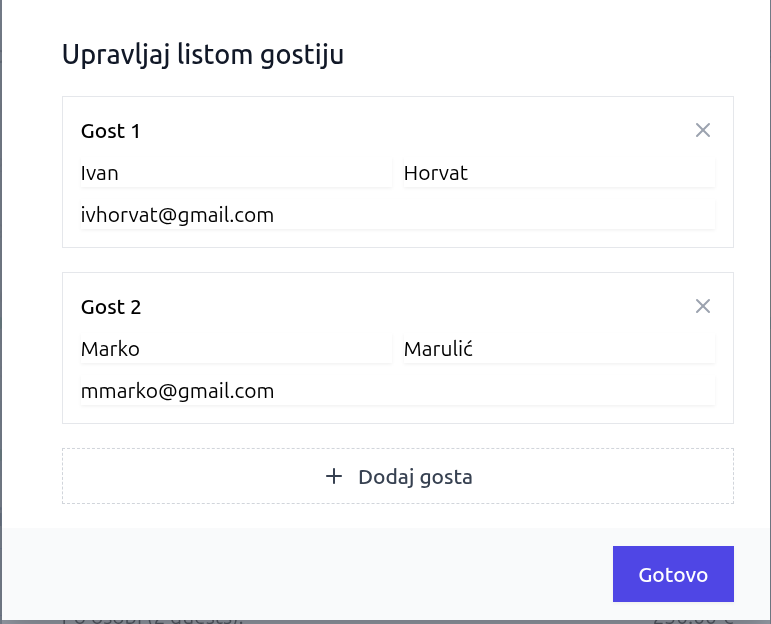
Slika 4.19 Kupčev pregled događaja

Slika 4.20 Forma za rezervaciju

Slika 4.21 Forma za rezervaciju nastavak

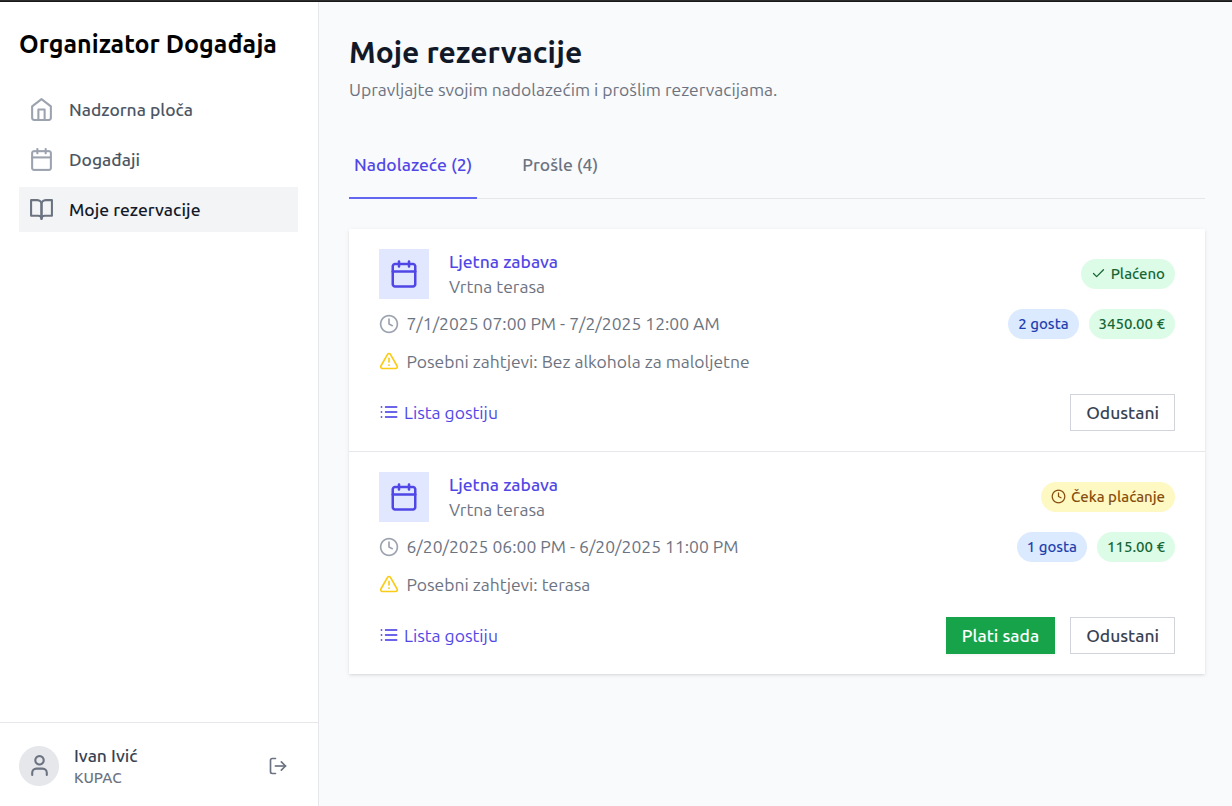
Slika 4.22 Forma za rezervaciju odabir termina

Prilikom rezerviranja korisnik određuje termin rezervacije te upisuje goste koji su pozvani na istu. Cijena se sama izračunava u ovisnosti o broju ljudi i prikazuje ažurno kupcu kako u nijednom trenutku nije neispravno informiran o cijeni rezervacije.



Slika 4.23, 4.24 Skočni prozor za dovršavanje rezervacije, Skočni prozor za odabir gosta unutar forme za rezervaciju

Nakon što je korisnik rezervirao svoju rezervaciju može pregledati sve svoje aktivne i prošle rezervacije u prikazu „Rezervacije” kojem može pristupiti klikom na odgovarajuće polje unutar hamburger izbornika.

Slika 4.25 Prikaz rezervacija kupca

Iz ovog pregleda korisnik može platiti rezervaciju pri čemu je organizator obaviješten na drugoj strani sustava.

# Moguće promjene i nadogradnje sustava

Web aplikaciju je moguće proširiti tako da pospješimo i ublažimo trenutno korisničko iskustvo te da unaprijedimo i dodamo nove funkcionalnosti. Unaprjeđivanje korisničkog iskustva bi se moglo izvesti dodavanjem većeg broja opcija filtriranja samih podataka. Kada bi kupac imao mogućnost dodavanja liste gostiju na događaj putem .csv datoteke ili nekim drugim načinom spremanja podataka o gostima značajno bi se ubrzao proces rezervacije. Dodavanjem nekakvog sučelja za upravljanjem i automatiziranjem elektroničke pošte moglo bi se ostvariti ubrzano obavještavanje gostiju o navedenom događaju. Dijeljenje ponuda i prostora između organizatora bi bila dobra nadogradnja samo bi se morala izvesti s potpunom opreznošću jer bi u isto vrijeme mogli stvoriti ekosustav organizatora koji pomažu jedni drugima i maksimiziraju si profit ali i potencijalnu kompetitivnu zonu gdje nitko ne bi htio dijeliti resurse. Najoptimalnije rješenje u tom slučaju bi bilo dodavanje atributa ponudama i prostorima – „vidljivost” gdje bi organizator odlučio želi li vlastite ponude i prostore podijeliti sa drugima. Još jedna od nadogradnji bi bio sustav recenziranja samih kupaca kako bi organizatori znali kome vjerovati. Također „ticketing sustav” odnosno sustav korisničke podrške koji bi rješavao aktualne probleme korisnika. Tu bi mogli dodati i AI bot-a koji bi značajno oslobodio i olakšao sustav korisničke podrške. Neke od funkcionalnih nadogradnji bi bile dodavanje vanjskog API-ja za procesuiranje plaćanja kako bi se na različit načine moglo plaćati preko sustava.

# Zaključak

Kroz ovaj rad razvijena je web-aplikacija koja omogućava organizaciju događaja i rezervaciju termina, ciljano prilagođena lokalnim korisnicima. Primjenom suvremenih tehnologija kao što su React, Spring Boot i PostgreSQL ostvareno je modularno, responzivno i sigurnosno robusno rješenje. Aplikacija podržava uloge organizatora i kupca, s jasnom podjelom funkcionalnosti i uloga, te obuhvaća ključne elemente korisničkog iskustva kao što su upravljanje prostorima, događajima, ponudama i rezervacijama.

Sustav omogućuje lako proširenje – primjerice dodavanjem vanjskih API-ja za plaćanje, automatizaciju obavijesti, dijeljenje resursa među organizatorima i naprednu analitiku. Nadalje, sustav osigurava sigurnost i stabilnost kroz autentifikaciju, validaciju podataka i vremenski definirane mehanizme promjene statusa rezervacija.

U budućnosti, dodatni iskoraci uključuju implementaciju sustava korisničke podrške, sustava recenzija, te boljeg sučelja za masovni unos gostiju. Time bi se dodatno unaprijedilo korisničko iskustvo, povećala konkurentnost i proširila primjenjivost aplikacije.

# Literatura

[1] Banks, A., Porcello, E. Learning React: Functional Web Development with React and Redux. 2. izdanje. Sebastopol: O'Reilly Media, 2020.

[2] Walls, C. Spring Boot in Action. Greenwich: Manning Publications, 2016.

[3] Oppel, A. J. Databases: A Beginner’s Guide. New York: McGraw-Hill Education, 2014.

[4] Stuttard, D., Pinto, M. The Web Application Hacker’s Handbook: Finding and Exploiting Security Flaws. 2. izdanje. Indianapolis: Wiley Publishing, 2011.

[5] Norman, D. The Design of Everyday Things. Revised and Expanded Edition. New York: Basic Books, 2013.

[6] Mozilla Developers. JWT Introduction – JSON Web Tokens. Poveznica: <https://developer.mozilla.org/en-US/docs/Web/Security/JWT>; pristupljeno 15. lipnja 2025.

[7] PostgreSQL Global Development Group. PostgreSQL 15 Documentation. Poveznica: <https://www.postgresql.org/docs/15/index.html>; pristupljeno 15. lipnja 2025.

[8] Spring.io. Spring Boot Documentation. Poveznica: <https://docs.spring.io/spring-boot/docs/current/reference/html/>; pristupljeno 15. lipnja 2025.

[9] React Team. React – Official Documentation. Meta Open Source. Poveznica: https://reactjs.org/docs/getting-started.html; pristupljeno 15. lipnja 2025.

[10] Tailwind CSS Team. Tailwind CSS Documentation. Poveznica: <https://tailwindcss.com/docs>; pristupljeno 15. lipnja 2025.

[11] GitHub Docs. Introduction to Git and GitHub. Poveznica: <https://docs.github.com/en/get-started/quickstart>; pristupljeno 15. lipnja 2025.

[12] JWT.io. JWT Debugger and Introduction. Auth0. Poveznica: https://jwt.io/introduction; pristupljeno 15. lipnja 2025.

[13] Bootify.io. Spring Boot Project Generator – Documentation. Poveznica: https://bootify.io/docs; pristupljeno 15. lipnja 2025. [13]

[14] ERDPlus. ER Diagram Tool for Database Modeling. Poveznica: [https://erdplus.com](https://erdplus.com/); pristupljeno 15. lipnja 2025.

**[15] Airbnb, Inc.** Airbnb – Online Platform for Short-Term Rentals and Experiences. Poveznica: [https://www.airbnb.com](https://www.airbnb.com/); pristupljeno 15. lipnja 2025.

**[16] Trivago N.V.** trivago – Compare Hotel Prices Worldwide. Poveznica: [https://www.trivago.com](https://www.trivago.com/); pristupljeno 15. lipnja 2025.

**[17] Njuškalo d.o.o.** Njuškalo – Hrvatski internetski oglasnik. Poveznica: [https://www.njuskalo.hr](https://www.njuskalo.hr/); pristupljeno 15. lipnja 2025.

**[18] Booking Holdings Inc.** Booking.com – Hotels, Homes & More. Poveznica: [https://www.booking.com](https://www.booking.com/); pristupljeno 15. lipnja 2025.

**[19] Eventimd.o.o.** Eventim.hr – Ulaznice za koncerte, festivale i događanja. Poveznica: [https://www.eventim.hr](https://www.eventim.hr/); pristupljeno 15. lipnja 2025.

# Sažetak

Web-aplikacija za organizaciju događaja i rezervaciju termina

U radu je opisana izrada web-aplikacije koja omogućuje korisnicima jednostavno upravljanje događajima i rezervacijama. Sustav uključuje dvije vrste korisnika – organizatore i kupce – te im omogućuje funkcije kreiranja i upravljanja događajima, prostorima, terminima i ponudama. Primijenjene su moderne tehnologije: React i Tailwind CSS za frontend, Spring Boot za backend te PostgreSQL za bazu podataka. Implementirana su sigurnosna rješenja poput JWT autentifikacije i HTTPS enkripcije. Aplikacija je dizajnirana za proširivost i uključuje mogućnosti budućih nadogradnji, poput integracije vanjskih servisa i dodatnih funkcionalnosti korisničke podrške. Ovaj rad prikazuje tehničku izvedbu, funkcionalne zahtjeve i arhitekturu sustava te nudi smjernice za daljnji razvoj.

**Ključne riječi:** web-aplikacija, rezervacije, organizacija događaja, React, Spring Boot, JWT, PostgreSQL

# Summary

Web Application for Event Organization and Scheduling

This thesis describes the development of a web application designed to streamline event organization and scheduling processes. The system supports two user roles – organizers and customers – enabling them to create and manage events, venues, time slots, and service offers. The application uses modern technologies: React and Tailwind CSS on the frontend, Spring Boot on the backend, and PostgreSQL as the database system. Security is ensured through JWT authentication and HTTPS encryption. The application architecture supports scalability and includes room for future improvements, such as external payment integration and customer support tools. This paper presents the implementation details, system architecture, and functional requirements while outlining potential directions for further development.

**Keywords:** web application, booking, event organization, React, Spring Boot, JWT, PostgreSQL