

# Web aplikacija za planiranje događaja s automatskim odabirom kandidata

---

**Kiš, Petra**

**Undergraduate thesis / Završni rad**

**2023**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:321041>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-02-18**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

**Sveučilišni studij**

**WEB APLIKACIJA ZA PLANIRANJE DOGAĐAJA S  
AUTOMATSKIM ODABIROM KANDIDATA**

**Završni rad**

**Petra Kiš**

**Osijek, 2023.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

Osijek, 19.09.2023.

Odboru za završne i diplomske ispite

**Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju**

<b>Ime i prezime Pristupnika:</b>	Petra Kiš
<b>Studij, smjer:</b>	Računalno inženjerstvo
<b>Mat. br. Pristupnika, godina upisa:</b>	R4509, 27.07.2020.
<b>OIB Pristupnika:</b>	25083773653
<b>Mentor:</b>	izv. prof. dr. sc. Zdravko Krpić
<b>Sumentor:</b>	,
<b>Sumentor iz tvrtke:</b>	
<b>Naslov završnog rada:</b>	Web aplikacija za planiranje događaja s automatskim odabirom kandidata
<b>Znanstvena grana rada:</b>	<b>Programsko inženjerstvo (zn. polje računarstvo)</b>
<b>Zadatak završnog rad:</b>	Zadatak rada je istražiti problematiku izrade web aplikacija koje omogućuju organizatorima događaja stvaranje i upravljanje događajima, praćenje registracije sudionika i automatizaciju procesa odabira najprikladnijih kandidata za događaj na temelju predefiniраних kriterija. U praktičnom dijelu rada potrebno je izraditi jednu takvu web aplikaciju s potporom za događaje poput studentskih praksi, ljetnih škola te
<b>Prijedlog ocjene završnog rada:</b>	Vrlo dobar (4)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 2 razina
<b>Datum prijedloga ocjene od strane mentora:</b>	19.09.2023.
<b>Datum potvrde ocjene od strane Odbora:</b>	
Potvrda mentora o predaji konačne verzije rada:	Mentor elektronički potpisao predaju konačne verzije.
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****IZJAVA O ORIGINALNOSTI RADA**

Osijek, 20.09.2023.

**Ime i prezime studenta:**

Petra Kiš

**Studij:**

Računalno inženjerstvo

**Mat. br. studenta, godina upisa:**

R4509, 27.07.2020.

**Turnitin podudaranje [%]:**

7

Ovom izjavom izjavljujem da je rad pod nazivom: **Web aplikacija za planiranje događaja s automatskim odabirom kandidata**

izrađen pod vodstvom mentora izv. prof. dr. sc. Zdravko Krpić

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

# SADRŽAJ

<b>1. UVOD .....</b>	<b>1</b>
1.1. Zadatak završnog rada .....	1
<b>2. POSTOJEĆA RJEŠENJA.....</b>	<b>2</b>
2.1. Automatski odabir kandidata .....	2
2.2. Eventbrite .....	3
2.3. MeetUp.....	4
2.4. EventZilla.....	5
2.5. Cvent .....	7
2.6. Bizzabo .....	8
<b>3. ZAHTJEVI NA PROGRAMSKO RJEŠENJE .....</b>	<b>10</b>
3.1. Popis korisničkih priča .....	10
<b>4. KORIŠTENE TEHNOLOGIJE.....</b>	<b>12</b>
4.1. HTML i CSS .....	13
4.2. Bootstrap.....	14
4.3. TypeScript .....	14
4.4. Angular .....	15
4.5. Java.....	15
4.6. Spring Boot .....	16
4.7. SQL.....	16
4.8. PostgreSQL.....	16
<b>5. IMPLEMENTACIJA WEB APLIKACIJE.....</b>	<b>18</b>
5.1. Baza podataka .....	18
5.2. Izrada korisničkog sučelja.....	19
5.3. Izrada poslužiteljskog dijela aplikacije.....	34
<b>6. VREDNOVANJE APLIKACIJE IZ PERSPEKTIVE KORISNIKA APLIKACIJE .....</b>	<b>38</b>
6.1. Odgovori s višestrukim izborom.....	38

<b>6.2. Odgovori na pitanja po uzoru na Likertovu skalu .....</b>	<b>41</b>
<b>6.3. Kratki odgovori.....</b>	<b>42</b>
<b>7. ZAKLJUČAK.....</b>	<b>44</b>
<b>LITERATURA .....</b>	<b>45</b>
<b>SAŽETAK.....</b>	<b>47</b>
<b>ABSTRACT .....</b>	<b>48</b>
<b>PRILOZI.....</b>	<b>49</b>
<b>ŽIVOTOPIS.....</b>	<b>50</b>

# 1. UVOD

Kvalitetna organizacija događaja postupak je koji zahtijeva veliki trud i uključuje mnogo parametara, predstavlja intrigantan i dinamičan proces koji ima sposobnost spajanja ljudi, poticanja komunikacije i stjecanja znanja. Prijave na događaje u prošlosti su se provodile putem telefonskih poziva, fizičkih pozivnica i osobnih susreta, što je zahtijevalo utrošak vremena i financijskih resursa. Sudjelovanje na više različitih događaja podrazumijevalo je odlaske na različite lokacije, što je i danas izazovno s obzirom na prometnu situaciju i troškove prijevoza. Zahtjevi i proces zapošljavanja osoba koje bi provodile razgovore za prijavu na događaje ili pregled papira su se značajno promijenili tijekom vremena. U prošlosti, ovakvi zadaci su često podrazumijevali ručno pretraživanje papirnatih prijava i osobne komunikacije s kandidatima. Ovo je zahtijevalo znatno više vremena i resursa u odnosu na današnje digitalno optimizirane procese. Kroz povijest, organizacija događaja često je bila povezana s nizom izazova, uključujući prikupljanje podataka o sudionicima, ograničenu personalizaciju iskustva sudionika i ograničene mogućnosti praćenja i analize. Digitalna rješenja omogućuju uštedu vremena i čine organizaciju događaja jednostavnijom, omogućavaju stvaranje brojnih događaja čiji će kandidati biti automatski izabrani na temelju kriterija koji su postavljeni.

## 1.1. Zadatak završnog rada

Zadatak rada je istražiti problematiku izrade web aplikacija koje omogućuju organizatorima događaja stvaranje i upravljanje događajima, praćenje registracije sudionika i automatizaciju procesa odabira najprikladnijih kandidata za događaj na temelju predefiniраниh kriterija. U praktičnom dijelu rada potrebno je izraditi jednu takvu web aplikaciju s potporom za događaje poput studentskih praksi, ljetnih škola te konferencija. U radu je potrebno obuhvatiti sve važne korake pri izradi programskog projekta, kao što su postavljanje zahtjeva na aplikacija, odabir tehnologija, dizajn aplikacije (arhitekturni i/ili detaljni), implementaciju te vrednovanje (verifikacija ili validacija). U samoj aplikaciji kriteriji za odabir kandidata trebali prilagođavaju se od strane organizatora događaja omogućujući mu postavljanje specifičnih zahtjeva poput obrazovanja, iskustva, vještina i drugih relevantnih faktora. Izraditi i implementirati algoritam izbora kandidata prema najboljim praksama u postojećoj literaturi i sličnim programskim rješenjima uz prilagodbu potencijalnim specifičnostima aplikacije. Osim toga, aplikacija bi također trebala uključivati sljedeće značajke: Registracija i prijava za događaj, Automatizirano slanje e-pošte sudionicima i kandidatima.

## **2. POSTOJEĆA RJEŠENJA**

Postojeće aplikacije za planiranje događaja se mogu podijeliti u četiri glavne kategorije. Prva kategorija su platforme za registraciju događaja koje omogućuju organizatorima da stvore događaje, postavljaju datume i lokacije te prikupljaju prijave i registracije sudionika. Druga kategorija su kalendari i planeri događaja koji pomažu korisnicima da pregledavaju i planiraju događaje, obavještavaju se o budućim događajima i prate svoj raspored. Treća kategorija su aplikacije za upravljanje logistikom događaja koje pomažu organizatorima u planiranju i upravljanju rasporeda događanja, hranu i piće, tehničku opremu i slično. Posljednja, četvrta kategorija su aplikacije za prodaju karata koje omogućuju organizatorima prodaju ulaznica ili rezervaciju mjesta za svoje događaje putem interneta. Web aplikacija koja će biti izrađena u okviru ovog završnog rada spada u prvu kategoriju, stoga je istraživanjem i pregledavanjem područja rada izabrano pet programskih rješenja koja su po funkcionalnostima slična ili ista kao spomenuta web aplikacija. Svaka od ovih aplikacija ima mogućnost izrade raznih vrsta događaja, prijave kandidata na njih, automatski odabir kandidata na temelju predefiniраних kriterija koje je organizator postavio za vlastiti događaj i upravljanje tim događajem.

### **2.1. Automatski odabir kandidata**

Automatski odabir kandidata je proces u kojem se koriste algoritmi i metode kako bi se automatski rangirali potencijalni kandidati koji su se prijavili na određeni događaj. Ovaj proces omogućuje organizacijama da učinkovito procijene i izaberu najprikladnije kandidate za otvorene pozicije, smanjujući pritom subjektivne ili pristrane odluke. Postoje razne metode koje se koriste pri automatskom odabiru kandidata. Neke od metoda su analiza ključnih riječi u prijavi, ocjena kompetencija, analiza društvenih profila i ocjenjivanje prijavnice na događaj.

Analiza ključnih riječi je pristup koji uključuje pretraživanje životopisa kandidata kako bi se identificirale ključne riječi koje odgovaraju zahtjevima stvorenog događaja. Kandidati se rangiraju na temelju podudaranja ključnih riječi u njihovim životopisima s ključnim riječima zadanih u stvorenom događaju. Ocjena kompetencija je pristup u kojem se koriste ocjene i testiranja kako bi se procijenile tehničke vještine i kompetencije kandidata. Algoritmi uzimaju u obzir rezultate testova, obrazovanje i iskustvo kako bi rangirali kandidate prema njihovim sposobnostima, ovu metodu koristiti će web aplikacija koja će biti izrađena u sklopu završnog rada. Analiza društvenih profila je pristup koji uzima u obzir javno dostupne informacije s društvenih mreža i profesionalnih platformi, algoritmi mogu ocijeniti aktivnosti i interakcije kandidata. Ocjenjivanje prijavnice je pristup koji pruža mogućnost dodavanja ili oduzimanja bodova kandidatima nakon čitanja



prijavnice na događaj. Ocjenjivanje prijavnice će također sadržavati web aplikacija koja će biti izrađena u sklopu završnog rada.

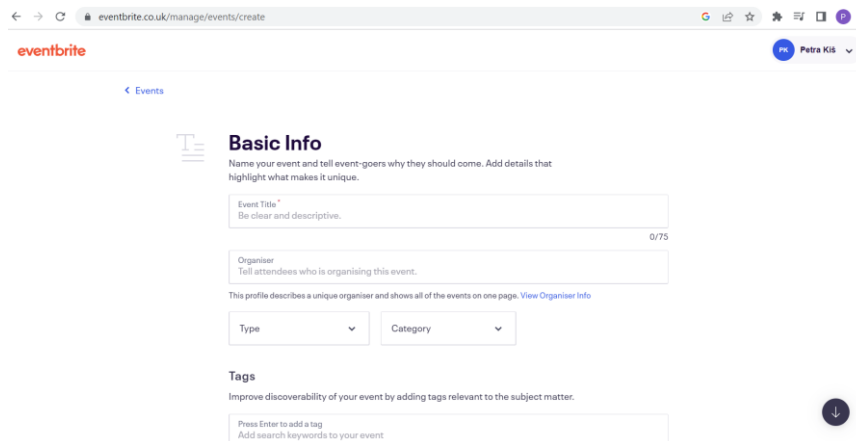
## 2.2. Eventbrite

Prema [1] , "Eventbrite" je široko korištena platforma, dostupna putem weba i mobilnih uređaja, koja pruža mogućnost organizatorima da izrade razne vrste događaja kao što su prakse, seminari, edukacije i brojni drugi događaji u svim područjima. Cilj aplikacije je omogućiti organizatorima lakše steći popularnost događaja preko promocija i reklama te omogućiti lakšu organizaciju uz manje posla. Ova platforma također nudi korisne alate za prijavu na događaje i automatizirano biranje kandidata. Korisničko sučelje aplikacije Eventbrite prikazano je na Slika . i dodavanje događaja na Slika .2.

Aplikacija EventBrite ima brojne funkcionalnosti, od kojih su neke mogućnost stvaranja događaja, prijava za događaj, automatski odabir kandidata za događaj po predefiniranim kriterijima, obavještavanje izabranih sudionika, upravljanje događajima, promocija događaja i prodaja ulaznica. Eventbrite i aplikacija koja će biti izrađena u okviru ovog rada imaju jako slične funkcionalnosti. Kao i u završnom radu, događaji se mogu izraditi i zainteresirane osobe se mogu prijaviti na događaj te se izabrani kandidati obavještavaju automatski. Web aplikacija Eventbrite ima probni period od četrnaest dana besplatnog korištenja, nakon čega se njezino korištenje naplaćuje, za razliku od aplikacija koja će biti izrađena u okviru ovog rada koja je besplatna. Također, nije često korištena na području Republike Hrvatske jer nije prilagođena na naše tržište i nije stekla veliku popularnost. Za potrebe istraživanja u ljetnom periodu na području Osijeka nije bilo niti jednog aktualnog događaja.



Slika 2.1. Početni izgled Eventbrite aplikacije



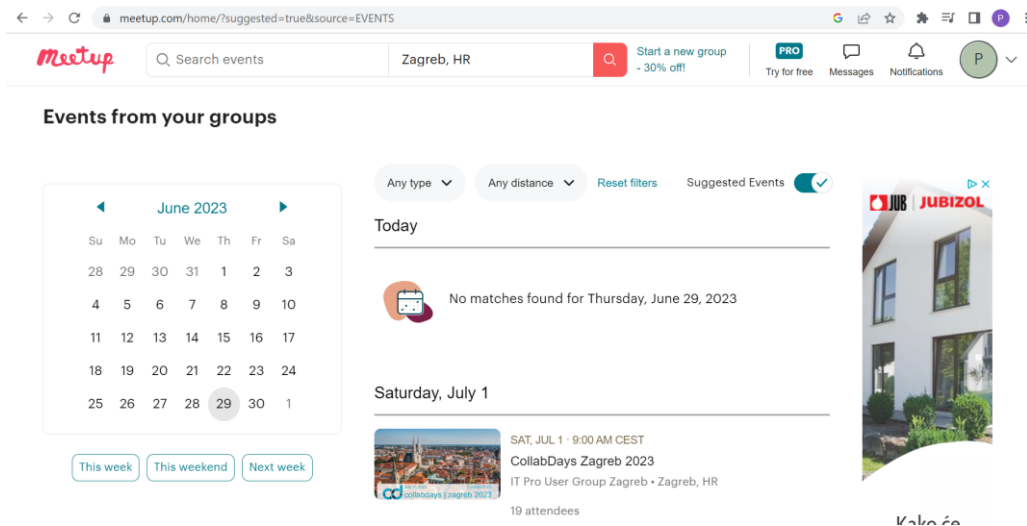
Slika 2.2. Prikaz dodavanja događaja

## 2.3. MeetUp

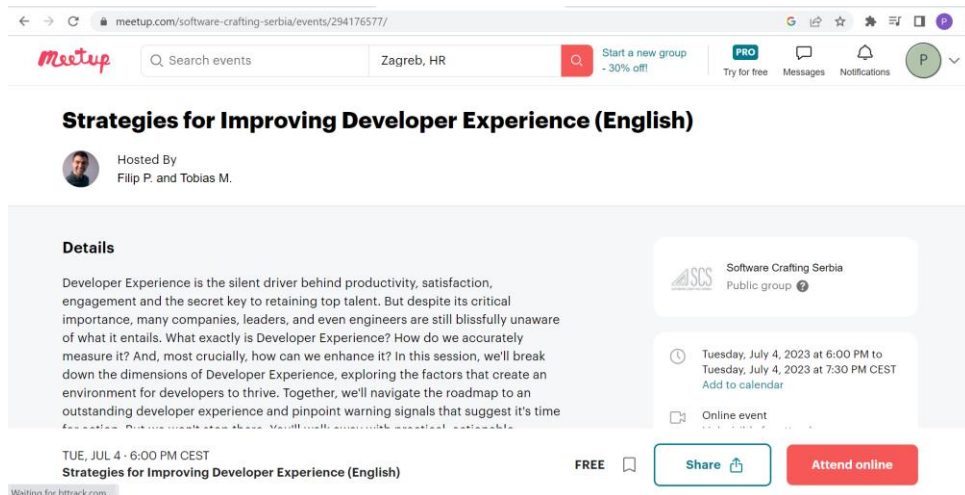
Meetup je prema [2] popularna platforma koja spaja ljude s istim interesima i omogućava im da se sastanu i sudjeluju u raznovrsnim stvarnim događajima. Ova aplikacija stvara mogućnosti za otkrivanje novih hobija, razvijanje vještina i upoznavanje istomišljenika. Aplikacija Meetup organizatorima omogućava jednostavno stvaranje događaja u raznim područjima kao što su tehnologija, umjetnost, sport, edukacija, putovanja i mnoge druge. Imaju fleksibilnost odabira datuma, vremena i lokacije događaja te mogu pružiti sve potrebne informacije sudionicima. To omogućava organizatorima da ciljano privuku ljude koji su zainteresirani za određenu temu ili aktivnost. Mogu se jednostavno prijaviti za sudjelovanje na događajima i povezati s drugim sudionicima koji dijele iste interese.

Slika 2.3. prikazuje naslovnu stranicu koja ima mogućnost pretraživanja događaja po nazivu, lokaciji i vremenu održavanja, dok Slika . prikazuje određeni događaj s mogućnošću prijave na njega. Aplikacija MeetUp omogućava stvaranje događaja, pretraživanje događaja, prijavu za događaje, primanje obavijesti, odabir interesa, odabir lokacije i vremena održavanja

Iako MeetUp i aplikacija koja će biti izrađena u okviru ovog rada imaju istu funkcionalnost stvaranja i upravljanja događajima, prijavu sudionika, postoje i neke ključne razlike. MeetUp nema mogućnost automatskog odabira kandidata. Ograničenja aplikacije MeetUp su što korisnik mora biti pripadnik neke određene skupine kako bi se mogao prijaviti na neki od događaja, postoji mjesečno članstvo za neke funkcionalnosti i nije toliko popularan kao Eventbrite.



Slika 2.3. Naslovna stranica aplikacije MeetUp



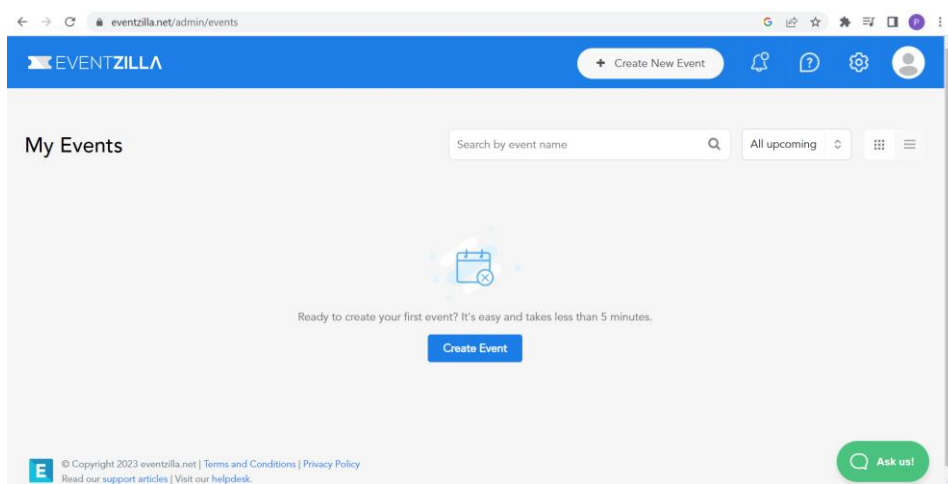
Slika 2.4. Prikaz jednog od događaja

## 2.4. EventZilla

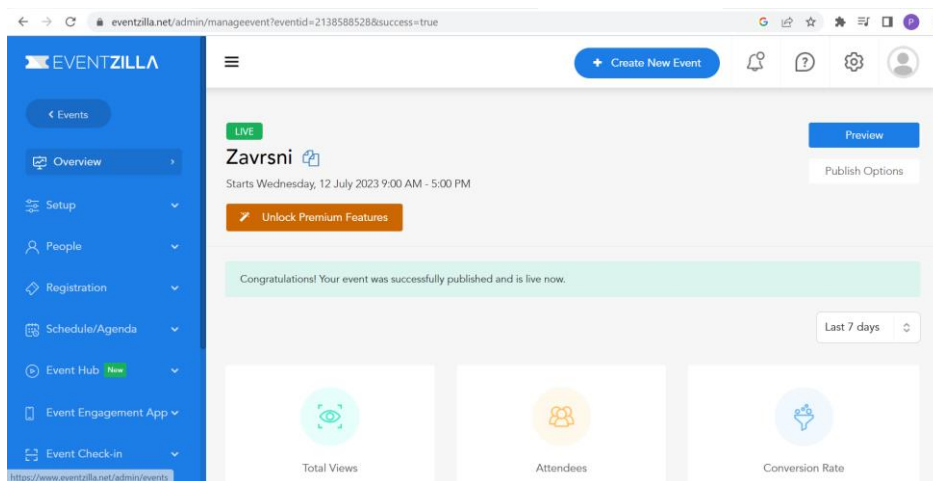
Prema [3], Eventzilla je platforma za upravljanje događajima koja omogućava organizatorima da izrade različite vrste događaja, poput konferencija, seminara, radionica i ostalih sličnih aktivnosti, također ima mogućnost automatskog odabira kandidata. Kada organizator koristi Eventzilla platformu za stvaranje događaja, moguće je dodati prilagođeni obrazac za prijavu sudionika u obliku bodovanja i kriterija koje organizator postavi. Organizator ima mogućnost postavljanja određenih kriterija i zahtjeva za prijavu, kao što su iskustvo, vještine ili interesi. Kada pristignu prijave, Eventzilla platforma filtrira prijave na događaj prema algoritmu za automatski odabir kandidata koji najbolje odgovaraju postavljenim kriterijima. Ova funkcionalnost olakšava

organizatorima proces odabira kandidata za događaj, štedeći vrijeme i trud koji bi inače bili potrebni za ručno pregledavanje i odabir prijava. Funkcionalnosti aplikacije EventZilla su stvaranje događaja, automatski odabir kandidata, prodaja ulaznica, analitika događaja, prilagođeni obrazac za prijavu koji sadrži kriterije koje organizator može postaviti i broj bodova koji će svaki kriterij nositi pri automatskom računanju bodova. EventZilla ima slične funkcionalnosti kao i aplikacija koja će biti izrađena u okviru ovog rada, a to su automatski odabir kandidata i prilagođeni obrazac za prijavu na temelju kojega se boduju kandidati (i na temelju kojeg se šalje prijava). EventZilla naplaćuje mjesečnu članarinu kada istekne probni period od četrnaest dana što nekim korisnicima može biti izazov i postoji mogućnost traženja alternativne web aplikacije za korištenje.

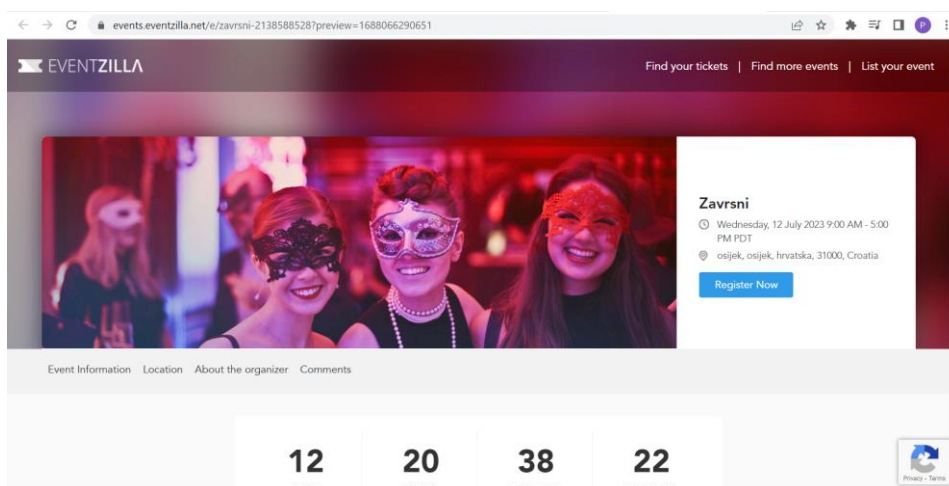
Slika . prikazuje naslovnu stranicu koja daje mogućnost dodavanja novih događaja i pregled postojećih, nakon izrađivanja događaja prikazuju se njegove karakteristike gdje je vidljiva i analitika događaja na Slika ., Slika . prikazuje izgled događaja koji korisnici mogu vidjeti.



Slika 2.5. Naslovna stranica EventZille



Slika 2.6. Izrađeni događaj



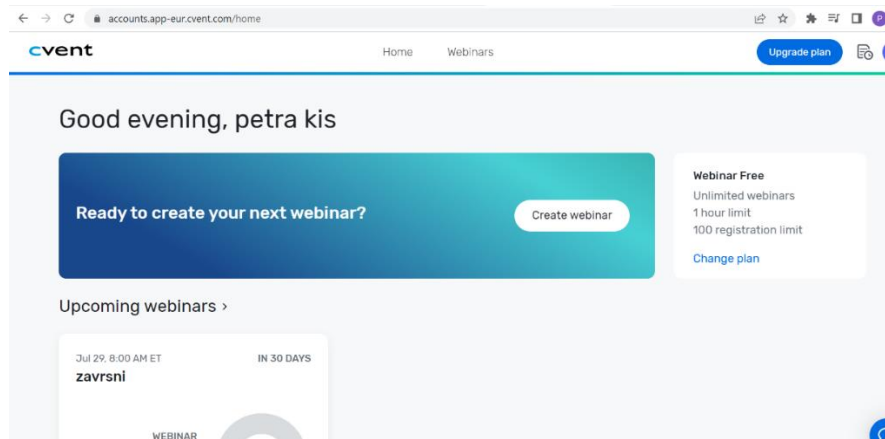
Slika 2.7. Izgled događaja za klijente

## 2.5. Cvent

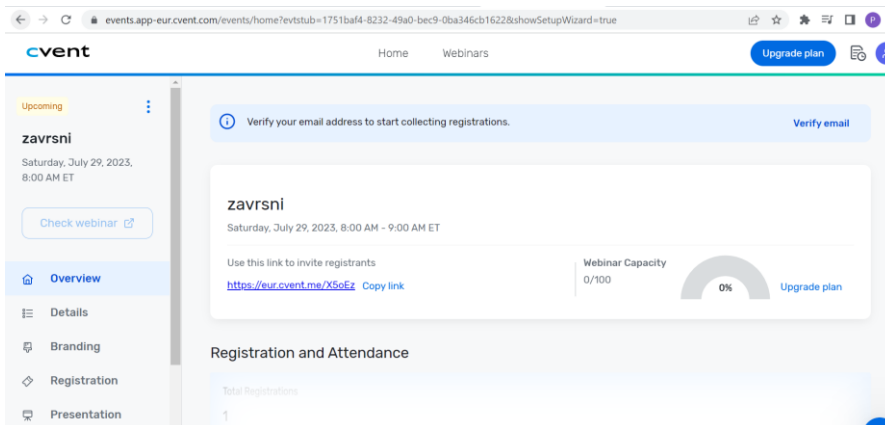
Cvent je, kao što je spomenuto u [4], platforma za upravljanje događajima koja pruža organizatorima sve potrebne alate i značajke za uspješno planiranje, promociju i upravljanje njihovim događajima. Jedna od glavnih značajki Cventa je automatski odabir kandidata. Ova značajka koristi algoritam za računanje ukupnog broja bodova i filtriranje kandidata kako bi organizatorima pomogla u automatskom odabiru najprikladnijih kandidata na temelju postavljenih kriterija.

Funkcionalnosti koje ima platforma Cvent su stvaranje događaja, automatski odabir kandidata, analitika događaja, pristup događaju preko URL-a i praćenje prisutnosti sudionika. Sličnosti aplikacije koja će biti izrađena u okviru ovog rada i Cvent aplikacije su u automatskom odabiru kandidata te stvaranju i upravljanju događajem. Prednost web platforme Cvent je dobra analitika

događaja i automatski odabir kandidata. Web platforma Cvent naplaćuje korištenje usluga koje pruža web platforma nakon probnog korištenja i sadrži limit sudionika u događajima u probnom periodu. Slika . prikazuje naslovnu stranicu koja ima mogućnost izrađivanja događaja i pregled nadolazećih, dok Slika . prikazuje analitiku izrađenog događaja i njegove detalje.



Slika 2.8. Naslovna stranica Cvent platforme



Slika 2.9. Pregled izrađenog događaja

## 2.6. Bizzabo

Bizzabo je platforma za upravljanje događajima koja omogućava organizatorima da stvaraju, promoviraju i upravljaju raznim vrstama događaja. Popularan izbor među organizatorima događaja zbog bogatih značajki. Prema [5], korisnici Bizzabo web platforme imaju mogućnost bavljenja marketingom i promocijom događaja na prilagođen, intuitivan i jednostavan način kako bi događaj imao veći odaziv. Bizzabo omogućava stvaranje događaja, analitiku i izvješća, marketing događaja i prijavu sudionika na događaj. Negativne karakteristike Bizzabo platforme su cijena i nemogućnost pristupa izrađivanju događaja ni tijekom probnog perioda. Slika 2. prikazuje primjer događaja napravljen u suradnji s Bizzabo platformom.



*Slika 2.10. Primjer Bizzabo događaja*



### 3. ZAHTJEVI NA PROGRAMSKO RJEŠENJE

Nakon analize postojećih web aplikacija za planiranje događaja koje omogućuju organizatorima da stvore događaje, postavbe datume i lokacije te prikupljaju prijave i registracije sudionika, u ovom poglavlju bit će definirani zahtjevi na web aplikaciju koja će biti izrađena u okviru ovog rada. Zahtjevi su oblikovani prema iskustvima iz spomenutih web aplikacija i njihovim funkcionalnostima u obliku korisničkih priča. Prema [6], korisničke priče (engl. *user stories*) su jedan od načina za bilježenje zahtjeva na aplikaciju i fokusirane su na pogled na aplikaciju iz korisničke perspektive, čime opisuju funkcionalnost ili mogućnosti aplikacije koje korisnik vidi. Koriste jezik koji je razumljiv svima uključenima u razvoj i korištenje programskog rješenja, uključujući razvojni tim, dizajnere i korisnike, čime se olakšava komunikacija između članova tima i smanjuje pojava nesporazuma. Općenito, korisnička priča je u formatu:

"Kao [tip korisnika], želim [funkcionalnost] kako bih [cilj]".

#### 3.1. Popis korisničkih priča

Pomoću korisničkih priča navedeni su zahtjevi koje mora ispuniti aplikacija koja će biti izrađena u okviru ovog rada.

PRIČA 1	Kao korisnik, želim imati uvid u sve događaje koji se nalaze u bazi podataka aplikacije kako bih imao sve potrebne informacije.
PRIČA 2	Kao korisnik, želim imati mogućnost pretraživanja događaja po nazivu, kako bih pronašao događaje koji su u mojoj zoni interesa.
PRIČA 3	Kao korisnik, želim imati mogućnost detaljnog pregleda svakog događaja posebno, kako bih bio ustanovio želim li se prijaviti na događaj.
PRIČA 4	Kao korisnik, želim mogućnost prijave na događaj bez izrade korisničkog računa.
PRIČA 5	Kao korisnik, želim imati jednostavnu i brzu prijavu u sustav web aplikacije.
PRIČA 6	Kao korisnik, želim da mi aplikacija pošalje potvrdu o prijavi na događaj u obliku elektroničke pošte.
PRIČA 7	Kao korisnik koji želi stvoriti vlastiti događaj, želim imati uvid u sve događaje koje sam stvorio, kako bi mi praćenje više događaja bilo jednostavnije.
PRIČA 8	Kao korisnik koji želi stvoriti vlastiti događaj, želim da stvaranje događaja bude jednostavno i intuitivno.



PRIČA 9	Kao korisnik koji je izradio vlastiti događaj, želim mogućnost biranja vještina koje želim bodovati kod registracija korisnika s brojem bodova koje odredim za svaku vještinu posebno.
PRIČA 10	Kao korisnik koji je izradio vlastiti događaj, želim imati mogućnost uređivanja osnovnih podataka o događaju i brisanje događaja.
PRIČA 11	Kao korisnik koji je izradio vlastiti događaj, želim imati uvid u svaku registraciju u obliku tablice sortirane od registracije s najvećim brojem bodova do registracije s najmanjim brojem bodova.
PRIČA 12	Kao korisnik koji je izradio vlastiti događaj, želim imati mogućnost dodavanja bodova ako mi se registracija korisnika sviđa ili ne sviđa.
PRIČA 13	Kao korisnik koji je izradio vlastiti događaj, želim da sustav automatski rangira registracije i ima mogućnost pozivanja kandidata koji su iznad granice maksimalnog broja sudionika
PRIČA 14	Kao korisnik koji je izradio vlastiti događaj, želim da aplikacija automatski pošalje pozivnicu sudionicima u obliku elektroničke pošte.
PRIČA 15	Kao korisnik koji je izradio vlastiti događaj, želim imati uvid u svakog sudionika i bilježiti njegov napredak po tjednima.

U daljnjem koraku korisničke priče bit će detaljnije analizirane, kako bi se omogućila pravilna interpretacija i specifikacija zahtjeva te kako bi se definirali ciljevi za daljnje planiranje razvoja aplikacije. Karakteristike aplikacije dobivene iz korisničkih priča:

- Pregled i pretraživanje događaja: Korisnici žele sposobnost pregleda svih događaja dostupnih u aplikaciji i mogućnost pretraživanja po nazivu. Ovo zahtijeva sustav za pohranu događaja i sposobnost pretraživanja.
- Detaljan pregled događaja: Korisnici trebaju mogućnost detaljnog pregleda svakog događaja kako bi donijeli odluku o prijavi. To uključuje prikaz osnovnih informacija, poput datuma i vremena, lokacije i opisa događaja.

- Jednostavna prijava: Korisnici žele jednostavnu prijavu na događaj bez potrebe za stvaranjem računa ili prijavom u sustav. Ovo zahtijeva sustav za brzu prijavu.
- Potvrda prijave na događaj: Nakon uspješne prijave, korisnici očekuju potvrdu putem elektroničke pošte kako bi bili sigurni da su se prijavili.
- Upravljanje vlastitim događajima: Korisnici koji organiziraju događaje trebaju mogućnost stvaranja, uređivanja i brisanja svojih događaja.
- Bodovanje i brisanje registracija: Organizatori događaja žele mogućnost bodovanja korisnika prijavljenih na događaj i brisanje prijave korisnika ako dođe do duplikacije.
- Pozivanje kandidata koji ispunjavaju kriterije: Sustav treba pozivati kandidate po automatskom bodovanju.
- Prikaz registracija i sudionika: Organizatori događaja trebaju popis svih registracija i sudionika, uključujući detaljne informacije o registraciji.
- Jednostavna i brza registracija te prijava u sustav: Korisnici ističu važnost jednostavne, brze i sigurne registracije i prijave.
- Prilagodljivo bodovanje vještina: Organizatori događaja trebaju dinamički model koji pruža mogućnost postavljanja različitih vještina i povezivanje bodova s njima za registraciju korisnika.

## 4. KORIŠTENE TEHNOLOGIJE

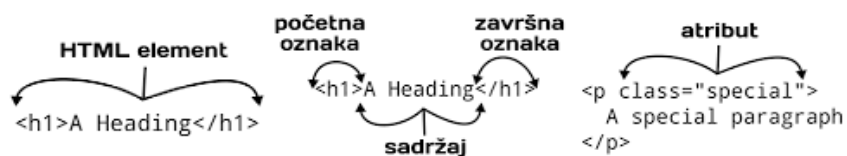
Aplikacija izrađena u okviru ovog rada sastoji se od dva dijela, korisničkog i poslužiteljskog. Preko korisničkog sučelja klijent komunicira direktno s web aplikacijom, a poslužiteljski dio obrađuje poslovnu logiku i upravlja podacima. Korisničko sučelje aplikacije razvijeno je koristeći programski okvir Angular. Angular se koristi za izgradnju reaktivnih komponenti i usmjeravanje na različite stranice, služi za razvoj dinamičkih korisničkih sučelja web aplikacija. Aplikacija koristi programski jezik TypeScript, prezentacijski jezik za izradu web stranica (engl. *HyperText Markup Language*, HTML) i stilski jezik CSS (engl. *Cascading Style Sheets*, CSS) koji su opisni jezici korišteni za strukturiranje i stiliziranje elemenata, dok je programski okvir (engl. *Framework*) Bootstrap pružio osnovne dizajnerske komponente i responzivnost web aplikacije. Poslužiteljski dio aplikacije je implementiran pomoću idućih tehnologija i alata: Spring Boot, Java,

SQL i PostgreSQL. Spring Boot je korišten za izgradnju poslužiteljskog okvira. Programski jezik Java se koristi za razvoj poslovne logike i manipulaciju podacima, dok se standardni jezik sustava za upravljanje relacijskim bazama podataka SQL (engl. *Structured Query Language*, SQL) koristi za dohvaćanje, ažuriranje i brisanje podataka iz PostgreSQL baze podataka. PostgreSQL je sustav za upravljanje bazama podataka i njen univerzalni alat PgAdmin koji za funkciju ima upravljanje tom bazom podataka.

## 4.1. HTML i CSS

HTML je osnovni jezik za izradu web stranica. Prema [7], hipertekstualni dokumenti elektronički su dokumenti ili datoteke koje sadrže tekstualne informacije, poveznice ili hiperveze koji omogućuju korisnicima da brzo i jednostavno prelaze s jednog dijela web stranice na drugi. Sadržaju hiperveze kao poveznice s drugim web stranicama. HTML pruža web preglednicima podatke o sadržaju i strukturi učitane Internet stranice. HTML5 je najnovija verzija koja donosi brojne mogućnosti koje su naprednije od prethodnih verzija. HTML daje osnovnu strukturu web stranice.

Svaki HTML dokument sastoji se od niza oznaka (engl. *tag*) koji definiraju različite elemente i uloge tih elemenata. *Slika 4.1.* vizualno prikazuje primjere nekih od oznaka u HTML jeziku. Prema slici 4.1. svaki HTML element se sastoji od sadržaja koji se nalazi između početne i završne oznake, unutar početne oznake se mogu dodati atributi, koji web pregledniku daju više informacija o sadržaju i njegovom prikazivanju.



Slika 4.1. Sadržaj HTML dokumenta

CSS je stilski jezik, koji se koristi se za prikazivanje dokumenta napisanog pomoću HTML jezika. Njime se uređuje raspored i izgled internet stranice. CSS, kao što je spomenuto u [7], nastao je kako bi omogućio odvajanje logike prikazivanja sadržaja i njegovog oblikovanja. Logika prikazivanja sadržaja odnosi se na način na koji se informacije ili elementi prikazuju kako bi se korisnicima prenijela određena poruka. Rezultat toga je manje linija programskog koda u HTML jeziku i samim time programski kod postaje pregledniji što omogućava njegovu lakšu izmjenu i lakše ispravljanje pogrešaka. CSS se sastoji od stilskih obrazaca koji su sastavljeni od stilskih pravila. Svako stilsko pravilo ima dva dijela: selektor i deklaracijski blok. Selektorom je određen

element na koji će se odnositi stilsko pravilo. Deklaracijski blok određuje izgled sadržaja opisanog pomoću CSS stilskog koda, dijeli se na svojstva i vrijednosti. *Slika 4.2.* prikazuje ranije objašnjene elemente: selektor i deklaracijski blok.



*Slika 4.2. Elementi CSS-a*

Ukratko, strukturni jezik HTML se koristi za strukturiranje sadržaja web aplikacija, definirajući osnovnu građu i organizaciju elemenata na web aplikacijama, stilski jezik CSS koristi se za stiliziranje izgleda web aplikacija. Navedene tehnologije zajedno čine temelj modernog razvoja web stranica i omogućuju stvaranje bogatih i funkcionalnih web sučelja.

## 4.2. Bootstrap

Prema [8], Bootstrap je CSS programski okvir koji se koristi za izradu responzivnih klijenskih sučelja na web stranici. Bootstrap je okvir koji automatski prilagođava responzivnost korisničkog sučelja različitim zaslonima i uređajima. Izradio ga je razvojni tim Twitter web platforme. Osnovna komponenta u Bootstrapu, koja se naziva 'spremnik', predstavlja strukturalni okvir u kojem se organiziraju i raspoređuju svi elementi web stranice kako bi se postigla responzivnost i jednostavan izgled. Bootstrap je jednostavan za korištenje, njegov CSS stilski kod omogućava lako prilagođavanje svim vrstama zaslona.

Programski okvir Bootstrap je kompatibilan sa svim internet preglednicima. Zahvaljujući Bootstrap programskom okviru, web programeri mogu fokusirati svoje napore na funkcionalnost i sadržaj web stranice, dok se za stilizaciju i responzivnost brine Bootstrap. Osim svoje responzivnosti, Bootstrap se odlikuje i prilagodljivošću. Web programeri mogu lako prilagoditi Bootstrap CSS stilski kod kako bi odgovarao određenim potrebama projekata, bez potrebe za pisanjem složenih stilova samostalno što iziskuje puno vremena.

## 4.3. TypeScript

Programski jezik visoke razine TypeScript, kao što je spomenuto u [9], predstavlja nadogradnju skriptnog programskog jezika JavaScript i donosi ključnu značajku - tipizaciju. Tipizacija omogućava precizno definiranje tipova podataka za varijable i funkcije, što čini

TypeScript strožim jezikom u usporedbi s JavaScriptom. Ovaj naglasak na tipovima pomaže u ranom otkrivanju potencijalnih grešaka u programskom kodu prije njegovog izvršavanja, što značajno poboljšava stabilnost i pouzdanost aplikacija. Također, TypeScript pruža bolju podršku za razvoj većih i složenijih aplikacija jer omogućava veću preciznost u deklaracijama tipova podataka i strukturi koda te olakšava suradnju među razvojnim timovima. Uz to, TypeScript se prevodi u čisti JavaScript, što znači da se može izvršavati u svim modernim web preglednicima i okolinama koje podržavaju izvođenje JavaScripta. Pomoću njegovog programskog koda moguće je raditi promjene na web stranicama koje su izrađene pomoću HTML-a i CSS-a kako bi se postigla dinamičnost i interaktivnost.

#### **4.4. Angular**

Prema [10], Angular je razvojni okvir otvorenog izvornog koda koji je razvila korporacija Google. Angular se koristi za razvoj dinamičnih i responzivnih web aplikacija pomoću HTML i TypeScript jezika. Prvo izdanje objavljeno je 14. rujna 2016. godine dok je zadnje stabilno izdanje dovršeno 7 godina nakon, 21. lipnja 2023. godine. Struktura Angular aplikacije se temelji na modulima, komponentama, direktivama, servisima, rutama i reaktivnim obrascima. Moduli su organizacijske jedinice koje grupiraju srodne komponente, usluge i druge resurse, definiraju se kao klase s dekoratorom `@NgModule` i omogućuju organiziranje modularnost aplikacije. Osim toga, Angular koristi koncept komponenti, koji omogućava izradu web aplikacija kao skupa reaktivnih komponenata koje olakšavaju organizaciju i održavanje koda. Svaka komponenta ima svoj HTML predložak koji definira izgled korisničkog sučelja, TypeScript koji opisuje programsku logiku komponente i CSS predložak za stiliziranje HTML strukturnog koda.

#### **4.5. Java**

Prema [11], Java je jedan od najpopularnijih i najkorištenijih objektno orijentiranih programskih jezika, razvijen je u tvrtki Sun Microsystems pod vodstvom inženjera Jamesa Goslinga i Patricka Naughtona. Ovaj jezik je postao ključan u svijetu razvoja programske podrške zbog svoje neovisnosti o platformi, što znači da se Java programi mogu izvršavati na različitim operacijskim sustavima bez potrebe za prevođenjem. Java se široko primjenjuje u razvoju raznovrsnih desktop, web, poslužiteljski i mobilnih aplikacija. Posebno je važna za razvoj mobilnih aplikacija jer postala osnovni jezik za programiranje Android operacijskog sustava. To je rezultiralo ekosustavom bogatih Android aplikacija koje se koriste na milijunima uređaja diljem svijeta. Sintaksa Java je slična programskim jezicima C, C++ i C#, što olakšava programerima s iskustvom u tim jezicima lak prelazak na Javu. Osim toga, Java je omiljeni jezik za razvoj

poslužiteljskih aplikacija, posebno uz korištenje Spring Boot programskog okvira, koja omogućava izgradnju skalabilnih i sigurnih web aplikacija.

## 4.6. Spring Boot

Spring Boot je programski okvir za razvoj Java aplikacija koji je dizajniran kako bi olakšao stvaranje samostalnih aplikacija, kao što je spomenuto u [12]. Ovaj okvir se ističe po svojoj sposobnosti brzog pokretanja i automatskog konfiguriranja mnogih dijelova aplikacije, što značajno smanjuje kompleksnost razvoja iste. Za početak rada sa Spring Bootom programeri često koriste web stranicu Spring Initializr, gdje mogu odabrati pakete i ovisnosti potrebne za njihovu Spring Boot aplikaciju, a sve ovisnosti se lako upravljaju putem datoteke pom.xml. Struktura koda u Spring Bootu obično slijedi arhitekturni obrazac Model-Pregled-Kontroler (engl. *Model-View-Controller*, MVC) gdje se aplikacija organizira u slojevima za bolje snalaženje i preglednost. Osim toga, Spring Boot pruža standardiziranu strukturu projekta koja uključuje pakete za upravljače, servise, repozitorije i konfiguriranje modela. Servisi u Spring Bootu su komponente koje sadrže funkcionalnost aplikacije i poslovnu logiku. Repozitoriji su sučelja koja omogućuju komunikaciju s bazom podataka i izvršavanje upita. Upravljači upravljaju hipertekstualni protokol prijenosa (engl. *HyperText Transfer Protocol*, HTTP) zahtjevima i odgovorima te preusmjeravaju ih prema odgovarajućim servisima. Preko HTTP zahtjeva web preglednik šalje web poslužitelju zahtjev za određenom stranicom ili objektom.

## 4.7. SQL

SQL je standardni jezik sustava za upravljanje relacijskim bazama podataka. Služi za pristup, rukovanje, pohranjivanje i dohvaćanje podataka iz baza podataka. Kao što je spomenuto u [13], postoje različite vrste softverskih sustava za upravljanje bazama podataka (engl. *Relational Database Management System*, RDBMS) kao što su PL/SQL, MySQL, SQLite i PostgreSQL, koji se koristio u i izradi programskog rješenja u okviru ovog završnog rada. Kako bi međusobno ispunjavali SQL standard, svi podržavaju glavne naredbe na sličan način. Relacijska baza podataka koristi tablice za pohranu podataka, koje se sastoje od stupaca i redaka u kojima se nalaze podatci. Stupac predstavlja atribut, a redak predstavlja slog.

## 4.8. PostgreSQL

Prema [14], PostgreSQL je objektno-relacijski sustav za upravljanje bazama podataka otvorenog koda koji koristi i proširuje SQL jezik. PostgreSQL podržava i SQL (relacijske) i JSON (nerelacijske) upite. Koristi se za upravljanje i pohranjivanje podataka. Također, ističe se svojom

robusnom i skalabilnom arhitekturom koja ga čini popularnim izborom za složene aplikacije i velike sustave. Osim toga, PostgreSQL ima brojna proširenja i podržava dodatne jezike za pisanje pohranjenih procedura, što ga čini prilagodljivim za različite potrebe razvoja aplikacija.

## 5. IMPLEMENTACIJA WEB APLIKACIJE

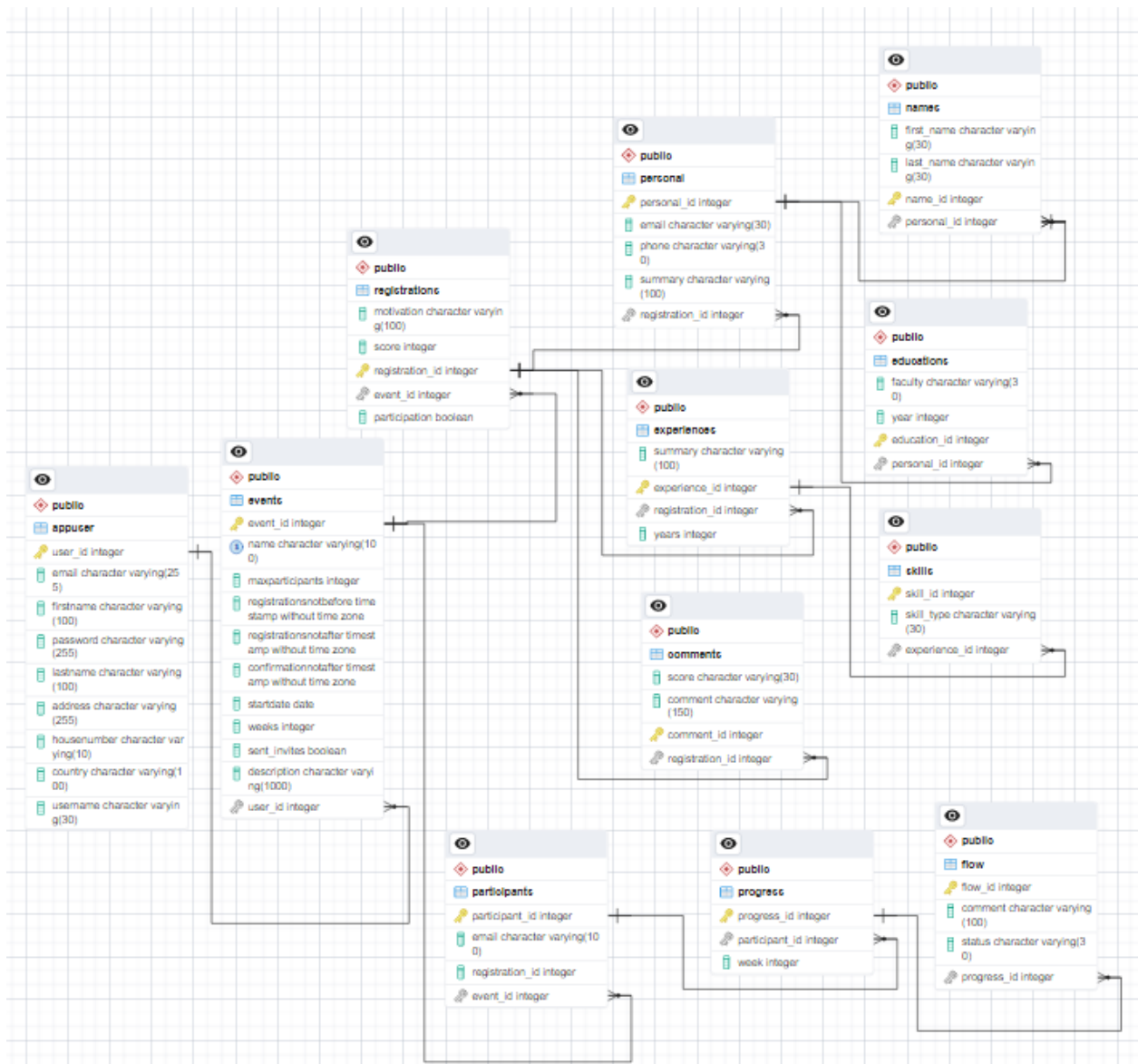
Programsko rješenje web aplikacije koja je izrađena u okviru završnog rada sastoji se od korisničkog sučelja, odnosno „prednje strane“ (engl. *frontend*), poslužiteljskog sučelja, odnosno „stražnje strane“ (engl. *backend*) i baze podataka.

### 5.1. Baza podataka

Baza podataka je strukturirano skladište podataka koje omogućava pohranjivanje, organizaciju i upravljanje podacima, čineći ih dostupnima za razne aplikacije i analize. Ključan je dio mnogih informacijskih sustava i računalnih aplikacija. Zbog jednostavnosti, baza podataka korištena u programskom rješenju izrađenom u okviru ovog završnog rada prikazana je dijagramom entiteta i veza (engl. *entity relationship diagram*, ER diagram). Prema [15], ER dijagram je grafički prikaz entiteta, atributa i relacija u kontekstu baze podataka, pomaže razumjeti strukturu podataka i odnose među njima, što je ključno za dizajn i upravljanje bazama podataka.

*Slika* prikazuje ER dijagram baze podataka završnog rada. Baza podataka sastoji se od tablica: Appuser, Events, Registrations, Personal, Names, Educations, Skills, Experiences, Comments, Participants, Progress i Flow. Primarni ključevi tablica su jedinstveni identifikator za svaki redak u tablici baze podataka, ključna vrijednost koja omogućava precizno identificiranje podataka. Grafički prikaz baze podataka dobiven je iz PgAdmin4 alata i njegove funkcionalnosti „Generate ERD“.

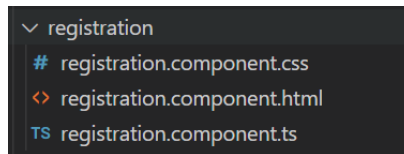




Slika 5.1. ER dijagram baze podataka

## 5.2. Izrada korisničkog sučelja

Korisničko sučelje je dio web aplikacije koji korisnici vide i s kojim imaju direktnu interakciju. Za izradu korisničkog sučelja korišteni su strukturni jezik HTML, stilski jezik CSS i programski jezik TypeScript. Razvojni okvir Angular, ranije spomenut i objašnjen, korišten je za razvoj korisničkog sučelja. U ovom poglavlju pobježe je prikazan i objašnjen izgled korisničkog sučelja uz njegove mogućnosti. Struktura Angular komponente Registration prikazana je na slici 5.2., dok i ostale komponente slijede istu strukturu.



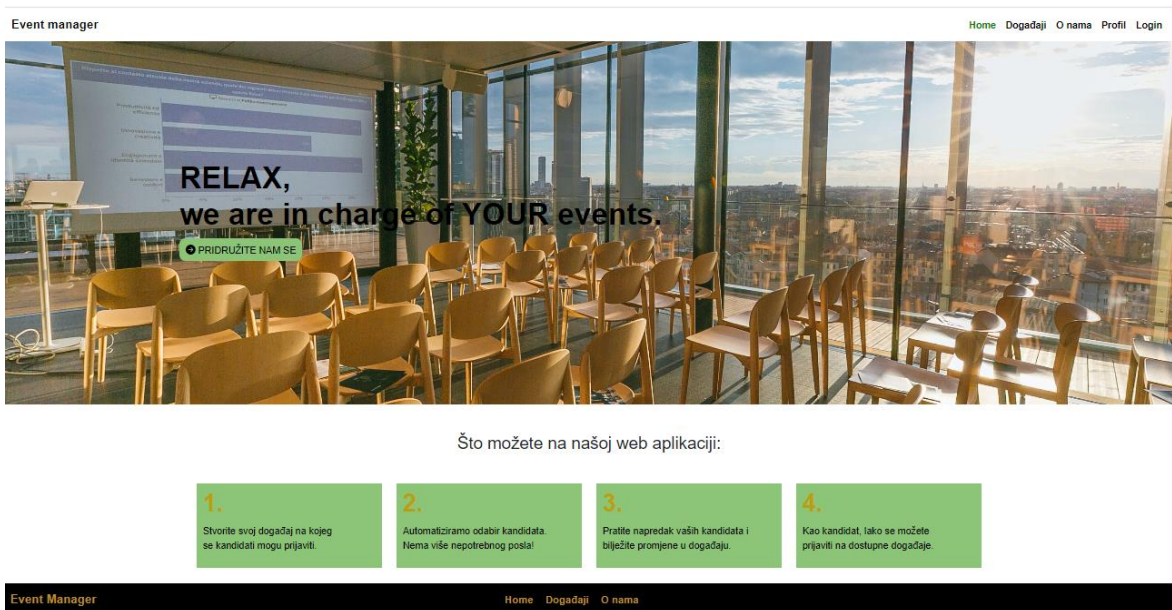
Slika 5.2. Dijelovi Angular komponente

Svaka stranica u web aplikaciji ima navigacijsku traku (engl. *Navbar*) na vrhu, na slici 5.3. prikazan je određeni sadržaj koji je implementiran pomoću Angular usmjeravanja (engl. *Routing*). Svaka od navedenih ruta na slici 5.3. vodi do specifične komponente, a detaljna objašnjenja ovih komponenti biti će navedena kasnije u ovom poglavlju. Također, na dnu stranice nalazi se zaglavlje (engl. *Footer*)

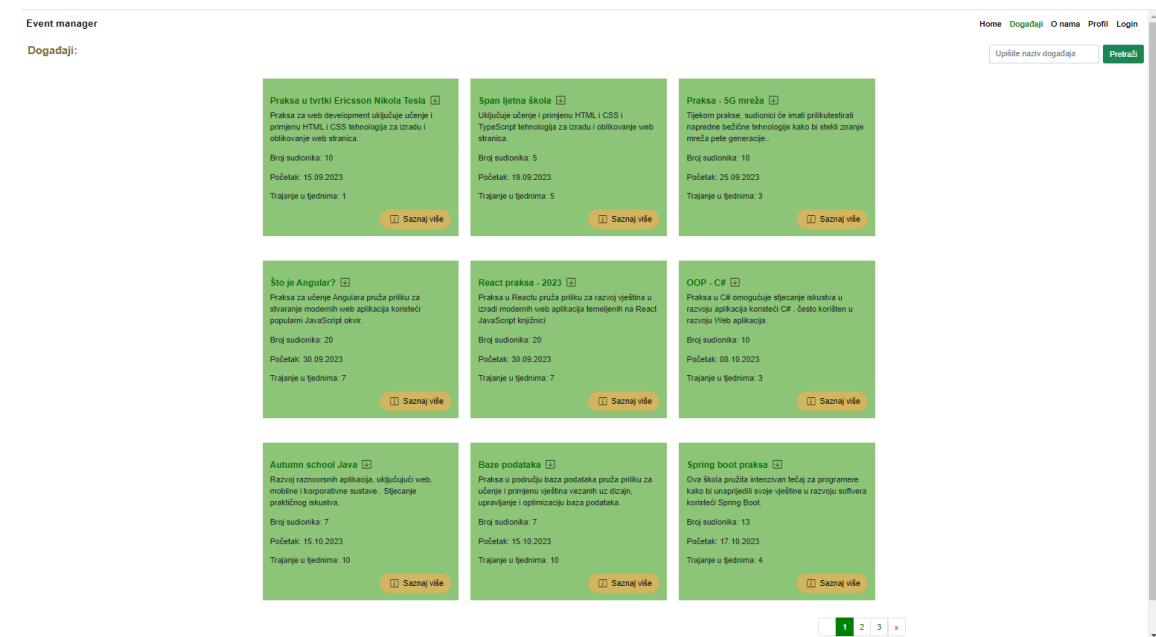
```
const routes: Routes = [
  {path: '', component: HomeComponent},
  {path: 'events', component: EventsComponent},
  {path: 'events/:eventId', component: EventComponent},
  {path: 'about', component: AboutComponent},
  {path: 'login', component: ContentComponent},
  {path: 'events/:eventId/registrations', component: RegistrationComponent},
  {path: 'profile', component: ProfileComponent },
  {path: 'create-event', component: MakeEventComponent },
  {path: 'user-event/:eventId', component: EventDetailsComponent },
  {path: 'event-edit/:eventId', component: EventEditComponent },
  {path: 'user-event/:eventId/registration/:registrationId', component:
RegistrationDetailsComponent },
  { path: 'participation-details/:participantId', component:
ParticipationDetailsComponent },
];
```

Slika 5.3. Prikaz ruta

Ulaskom u web aplikaciju prikazana je naslovna stranica sa Slike 5.4., sadrži gumb „Pridruži nam se“, klikom na gumb, korisnici će biti preusmjereni na stranicu *EventsComponent* koja se naziva „Događaji“ u navigacijskoj traci. Ovo olakšava korisnicima pristup i sudjelovanje u događajima. Komponenta *EventsComponent* prikazana je na slici 5.5., sastoji se od događaja poredanih po datumu, a na jednoj stranici moguće je prikazati najviše devet događaja.



Slika 5.4. Naslovna stranica



Slika 5.5. Komponenta EventsComponent

U *EventsComponent* komponenti postoji traka za navigaciju među stranicama (tj. straničenje) koja se u ovisnosti o broju događaja proširuje ili sužava. Kada korisnik klikne na posljednju numeriranu stranicu, strelica koja označava navigaciju prema naprijed nestaje, a gumb postaje neaktivan, kako je prikazano na slici 5.6. Ista situacija se ponavlja i kada korisnik klikne na prvu stranicu na traci za navigaciju između stranica. Slika 5.7. prikazuje mogućnost

pretraživanja događaja po bilo kojoj riječi iz imena događaja, također se traka za navigaciju među stranicama prilagođava broju pronađenih događaja.



Slika 5.6. Posljednja numerirana stranica



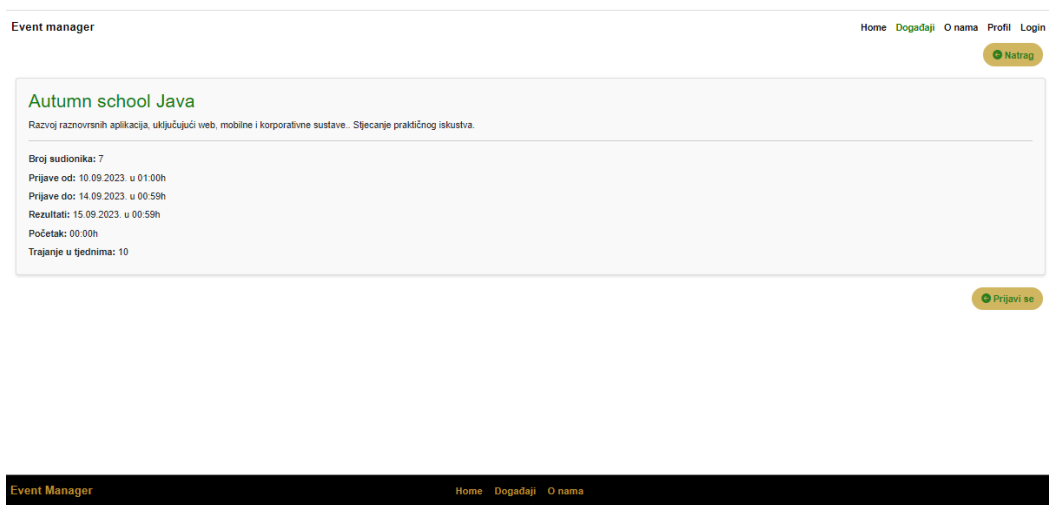
Slika 5.7. Funkcionalnost Pretraži

Svaki događaj na web stranici ima svoju ikonu u obliku strelice prema dolje. Ova ikona ima funkciju detaljnijeg prikaza osnovnih informacija događaja. Osim ikonice strelice, postoji i gumb pod nazivom „Saznaj više“. Klikom na ovaj gumb korisnik će biti preusmjeren na stranicu komponente *EventComponent*, koja je prikazana na slici 5.8. Na stranici *EventComponent* korisnik može pronaći dodatne informacije o tom događaju.

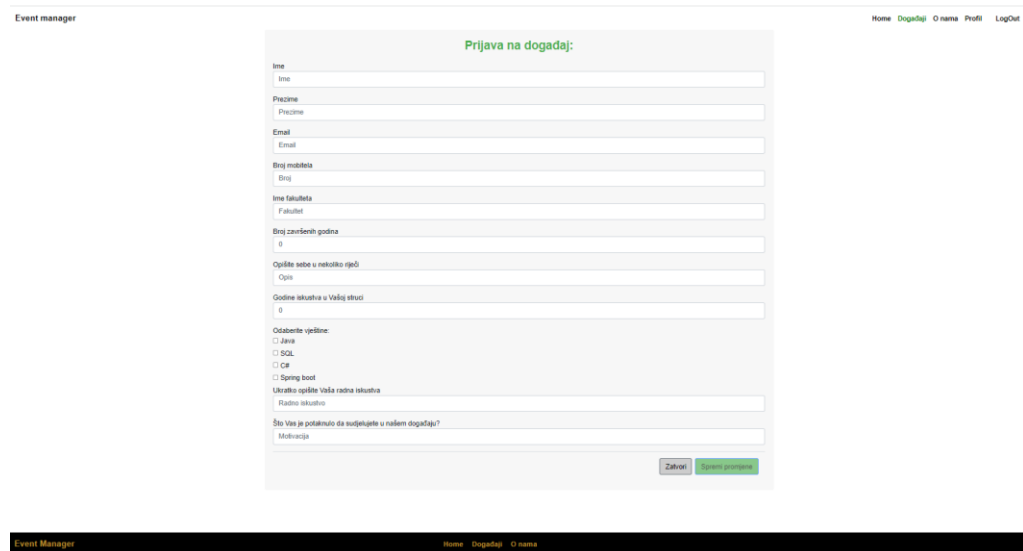
Također, na stranici *EventComponent* postoji i gumb „Prijavi se“. Klikom na ovaj gumb, korisnik će biti preusmjeren na stranicu komponente *RegistrationComponent*, koja je prikazana na slici 5.9.

Ova stranica predstavlja formu za popunjavanje i prijavu na taj određeni događaj. Korisnici mogu unijeti svoje informacije i potvrditi svoju prisutnost na događaju putem ove forme."

Svaki korisnik je obavezan upisati podatke u svako polje forme za prijavu na događaj. Primjer pravilnog ispunjavanja forme nalazi se na Slici 5.10. Na svaki događaj postoji mogućnost prijave bez izrađenog korisničkog računa. Informacija o uspješnosti prijave dolazi korisnicima u obliku elektronične pošte, primjer koje se nalazi na Slici 5.11.



Slika 5.8. Komponenta EventComponent



Slika 5.9. Komponenta RegistrationComponent

**Prijava na događaj:**

Ime  
Davor

Precizno  
Gota

Email  
davor.gota@gmail.com

Broj mobitela  
091567345

Ime fakulteta  
Feri Osijek

Broj završenih godina  
3

Opišite sebe u nekoliko riječi  
Zabavljenik u nove tehnologije, spretni saru u programiranju i backend je nešto u čemu se vidim

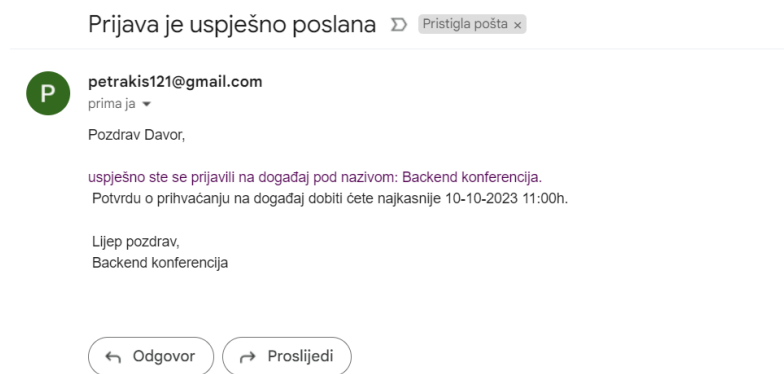
Godine iskustva u tojnoj stvari  
1

Odaberite vještine:  
 Java  
 SQL  
 C#  
 Spring boot

Ukratko opišite Vaša radna iskustva  
Radim godinu dana u Gideon brothers i sada sam počeo raditi u Ericsson Nikola Tesla u 5G odjelu

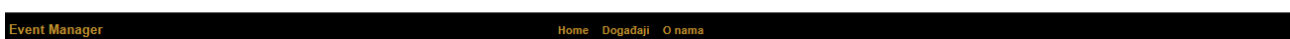
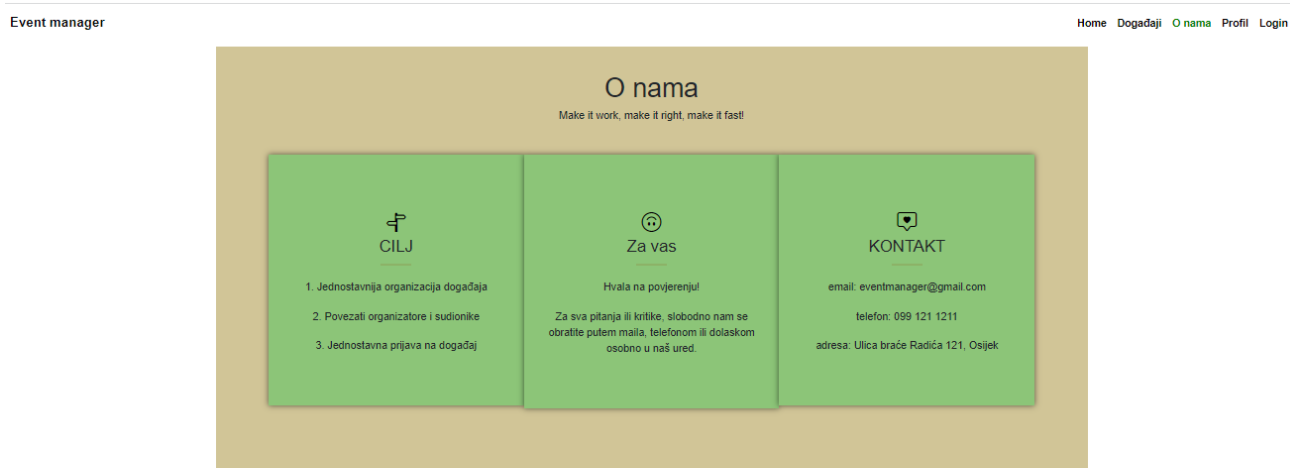
Što Vas je potaknulo da sudjelujete u našem događaju?  
Želja za naučiti backend još više.

Slika 5.10. Ispunjena prijava u komponenti `RegistrationComponent`



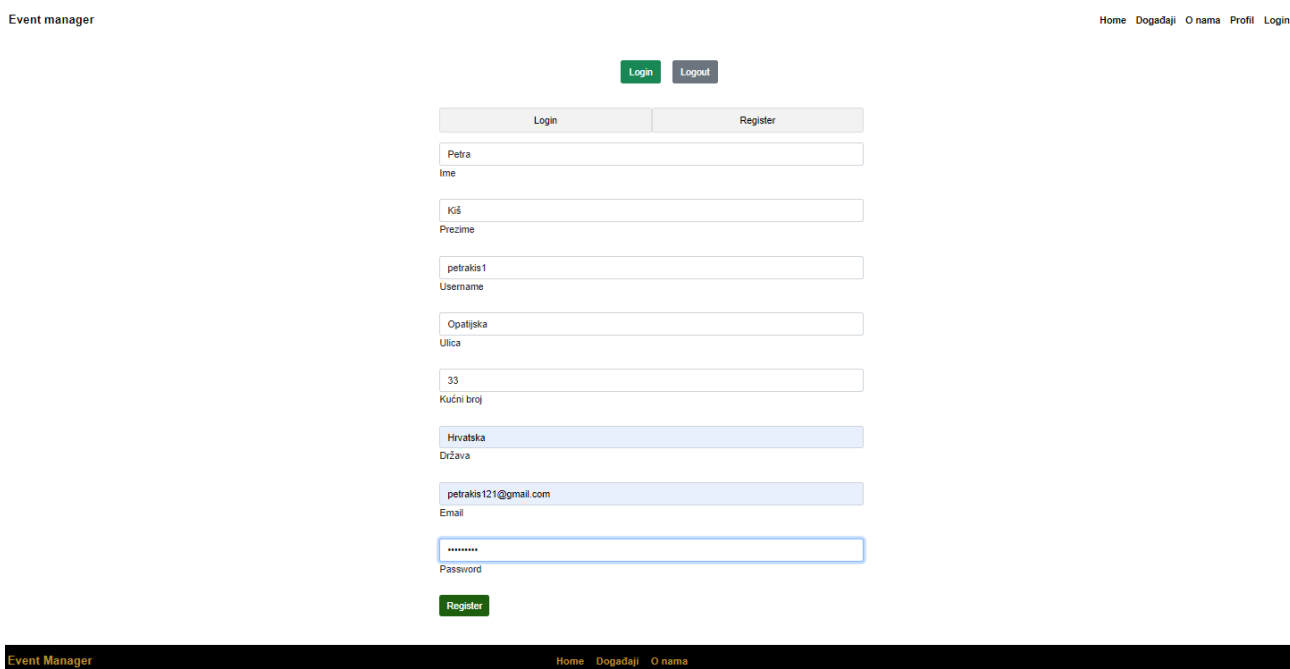
Slika 5.11. Elektronička pošta poslana korisniku nakon prijave na događaj

Klikom na „O nama“ u navigacijskoj traci aktivira se `AboutComponent` komponenta koja prikazuje osnovne informacije o web aplikaciji, prikazane na slici 5.12.

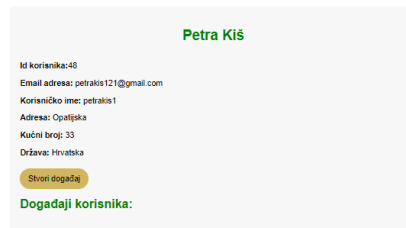


Slika 5.12. Komponenta AboutComponent

Nakon što su sve funkcionalnosti koje ne zahtijevaju registraciju ili prijavu obrađene, obrađuje se drugi dio web aplikacije. Na slici 5.13. prikazana je forma za registraciju korisnika u sustav, a nakon uspješne registracije, sustav automatski izvršava prijavu i prelazi na komponentu *ProfileComponent*, prikazanu na slici 5.14. Također, primjetno je da se u navigacijskoj traci mijenja naziv s „Login“ na „Logout“



Slika 5.13. Registracija u sustav, komponenta RegistrationComponent



**Petra Kiš**

Id korisnika:48  
Email adresa: petra.kis121@gmail.com  
Korisničko ime: petra.kis1  
Adresa: Opaljska  
Kućni broj: 33  
Država: Hrvatska

[Stvori događaj](#)

**Događaji korisnika:**

Slika 5.14. Komponenta ProfileComponent

Nakon što se korisnik uspješno registrira, otvara se mogućnost stvaranja događaja klikom na gumb „Stvori događaj“. Klikom na ovu tipku otvara se *MakeComponent*, što je prikazano na slici 5.15. U ovom koraku, korisnik može definirati vještine koje želi vrednovati u svom događaju. Uspješni prikaz stvaranja događaja je prikazan na slici 5.16.



**Stvorite događaj:**

Naziv događaja

Opis događaja

Maksimalan broj sudionika

Prijave počinju

Prijave završavaju

Potvrde o upisu na događaj

Datum početka događaja

Trajanje u tjednima

[Vještine koje očekujete od kandidata](#)

[Zatvori](#) [Spremi promjene](#)

Slika 5.15. Prikaz MakeEventComponent komponente



**Stvorite događaj:**

Naziv događaja  
Backend konferencija

Opis događaja  
Konferencija o backend developerima okuplja stručnjake za razvoj serverne strane aplikacija radi razmjene najnovijih tehnologija i praksi.

Maksimalan broj sudionika  
100

Prijave počinju  
09/12/2023

Prijave završavaju  
09/30/2023

Potvrde o upisu na događaj  
10/10/2023

Datum početka događaja  
10/15/2023

Trajanje u tjednima  
1

Vještine koje očekujete od kandidata

Ime vještine  
Java  
Broj bodova  
10

Ime vještine  
Spring boot  
Broj bodova  
20

Ime vještine  
SQL  
Broj bodova  
5

Slika 5.16. Prikaz popunjene forme u *MakeEventComponent* komponenti

Postoji opcija automatskog bodovanja registracija na događaj, organizatori imaju mogućnost postavljanja vlastitih kriterija za bodovanje sudionika. Ova opcija omogućava organizatorima da precizno definiraju na temelju kojih parametara žele ocjenjivati i rangirati prijave kandidata za svoj događaj. Na taj način, organizator ima potpunu kontrolu nad ocjenjivanjem kandidata za sudjelovanje u događaju

Nakon stvaranja događaja, taj događaj se dodaje u *ProfileComponent* komponentu, čije se ažurirano stanje mož vidjeti na slici 5.17. Također, događaj se pojavljuje i u *EventsComponent* komponenti, primjer ažurirane *EventsComponent* komponente može se vidjeti na slici 5.18.

**Petra Kiš**

Id korisnika:48  
 Email adresa: petrakis121@gmail.com  
 Korisničko ime: petrakis1  
 Adresa: Opaljska  
 Kućni broj: 33  
 Država: Hrvatska

[Svi događaji](#)

**Događaji korisnika:**

**Naziv događaja:** Spring konferencija  
 Opis: Spring konferencija okuplja stručnjake iz područja razvoja softvera kako bi razmijenili najnovija znanja i iskustva vezana za Spring ekosustav.  
**Maksimalni broj sudionika:** 35  
**Početak prijave:** 12.09.2023  
**Završetak prijave:** 14.09.2023  
**Početak:** 23.10.2023

[Detalji](#)

**Naziv događaja:** Scala praksa za studente  
 Opis: Scala je višenamjenski programski jezik koji kombinira funkcionalno i OOP, često korišten u razvoju skalabilnih i izdržljivih aplikacija.  
**Maksimalni broj sudionika:** 10  
**Početak prijave:** 10.09.2023  
**Završetak prijave:** 25.10.2023  
**Početak:** 29.10.2023

[Detalji](#)

**Naziv događaja:** Backend konferencija  
 Opis: Konferencija o backend developerima okuplja stručnjake za razvoj serverske strane aplikacija radi razmjene najnovijih tehnologija i praksi.  
**Maksimalni broj sudionika:** 100  
**Početak prijave:** 10.09.2023  
**Završetak prijave:** 01.10.2023  
**Početak:** 15.10.2023

[Detalji](#)

Slika 5.17. ProfilComponent komponenta nakon dodavanja događaja

Događaji:

Upišite naziv događaja

[Pretraži](#)

**Vue.js** ✖

Praksa u Vue.js omogućuje stjecanje iskustva u razvoju modernih web aplikacija koristeći brzi i jednostavan JavaScript okvir Vue.js.

Broj sudionika: 10  
 Početak: 20.10.2023  
 Trajanje u tjednima: 4

[Saznaj više](#)

**VHDL PRAKSA** ✖

VHDLje opisi jezik za digitalni dizajn koji omogućuje precizno modeliranje i simulaciju kompleksnih elektroničkih sklopova.

Broj sudionika: 12  
 Početak: 21.10.2023  
 Trajanje u tjednima: 6

[Saznaj više](#)

**Spring konferencija** ✖

Spring konferencija okuplja stručnjake iz područja razvoja softvera kako bi razmijenili najnovija znanja i iskustva vezana za Spring ekosustav.

Broj sudionika: 35  
 Početak: 23.10.2023  
 Trajanje u tjednima: 1

[Saznaj više](#)

**Scala praksa za studente** ✖

Scala je višenamjenski programski jezik koji kombinira funkcionalno i OOP, često korišten u razvoju skalabilnih i izdržljivih aplikacija.

Broj sudionika: 10  
 Početak: 29.10.2023  
 Trajanje u tjednima: 2

[Saznaj više](#)

**Backend konferencija** ✖

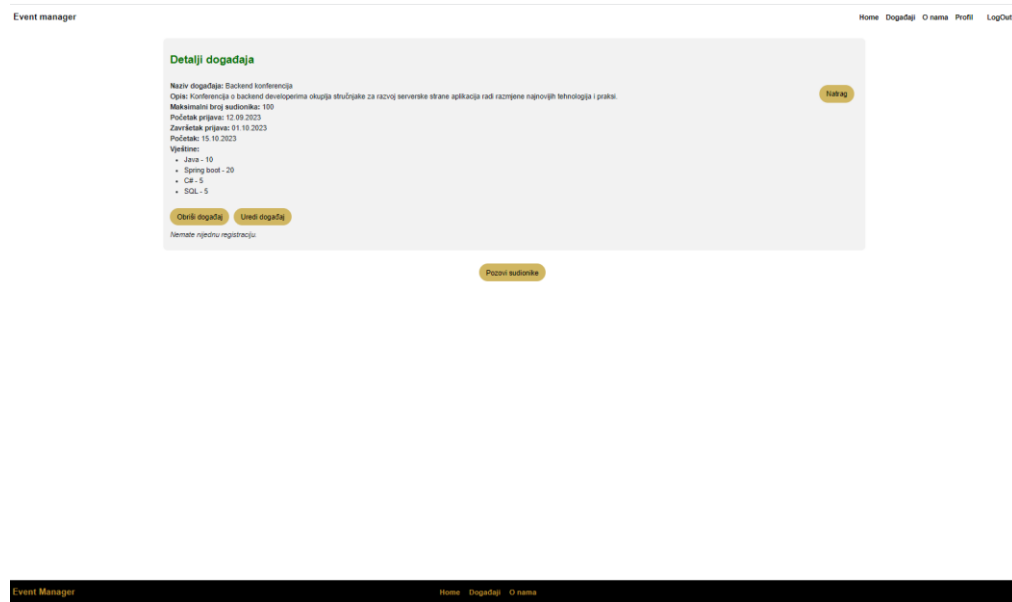
Konferencija o backend developerima okuplja stručnjake za razvoj serverske strane aplikacija radi razmjene najnovijih tehnologija i praksi.

Broj sudionika: 100  
 Početak: 15.10.2023  
 Trajanje u tjednima: 1

[Saznaj više](#)

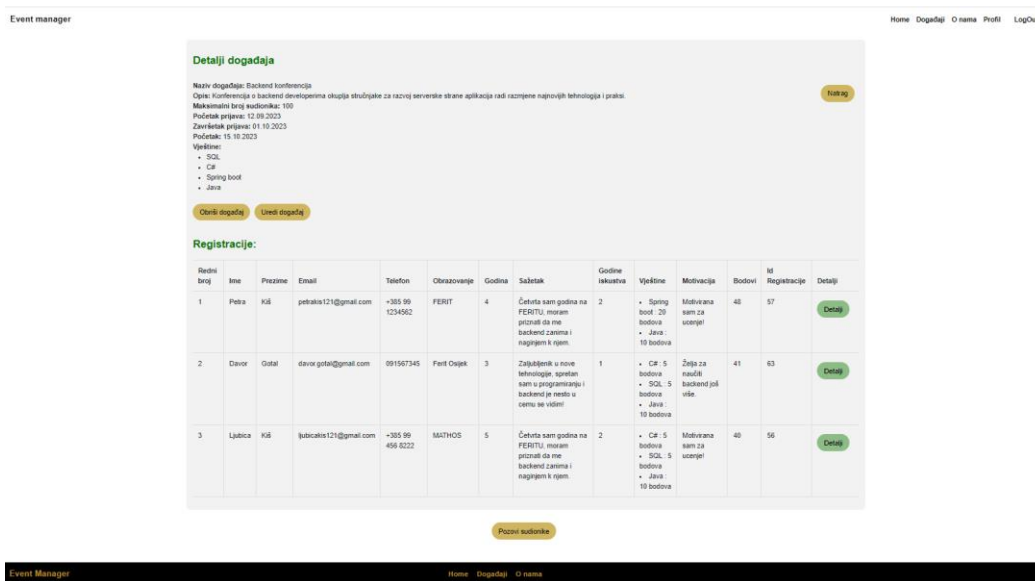
Slika 5.18. Ažurirana komponenta EventsComponent nakon dodavanja događaja

Klikom na gumb „*Detalji*“ otvara se *EventDetailsComponent* komponenta, prikazana na slici 5.19., koja sadrži opis događaja, mogućnost uređivanja i brisanja tog istog događaja. Ako se na događaj nije registroao niti jedan korisnik, u korisničkom sučelju biti će napisano „Nimate nijednu registraciju“, prikazano na slici 5.19.



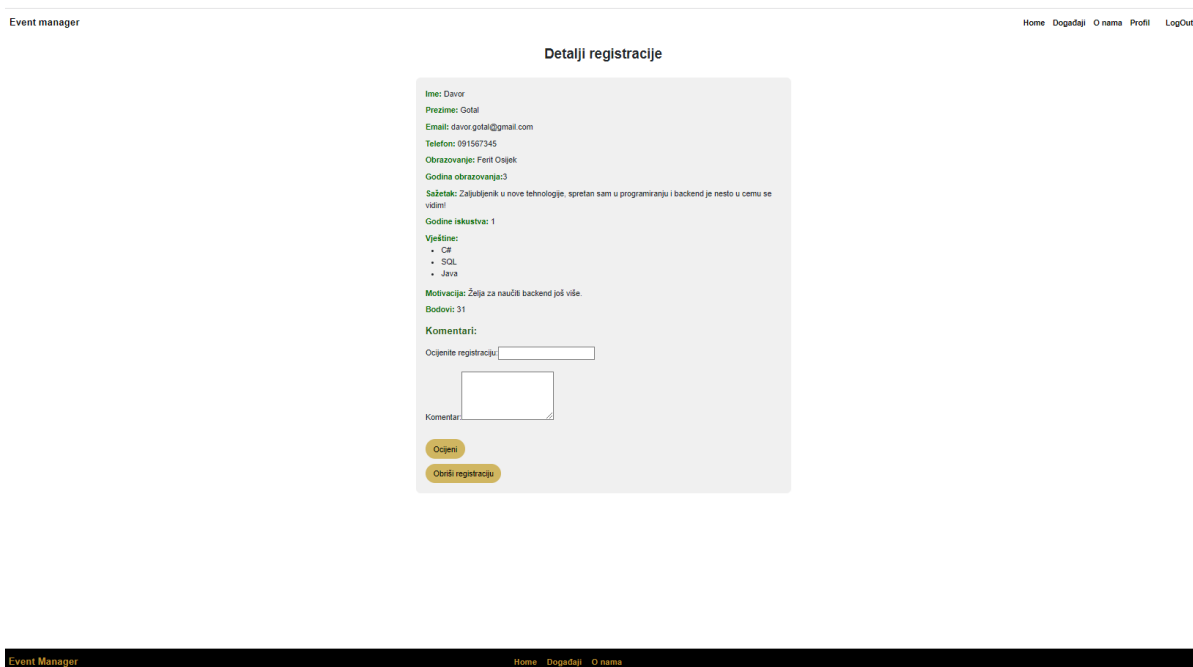
Slika 5.19. *EventDetailsComponent* komponenta bez registracija kandidata

Kada u događaju postoje registrirani korisnici, stvara se tablica koja sadrži popis kandidata rangiranih po broju bodova, od registracije korisnika s najviše bodova do registracije korisnika s najmanje bodova, slika 5.20.

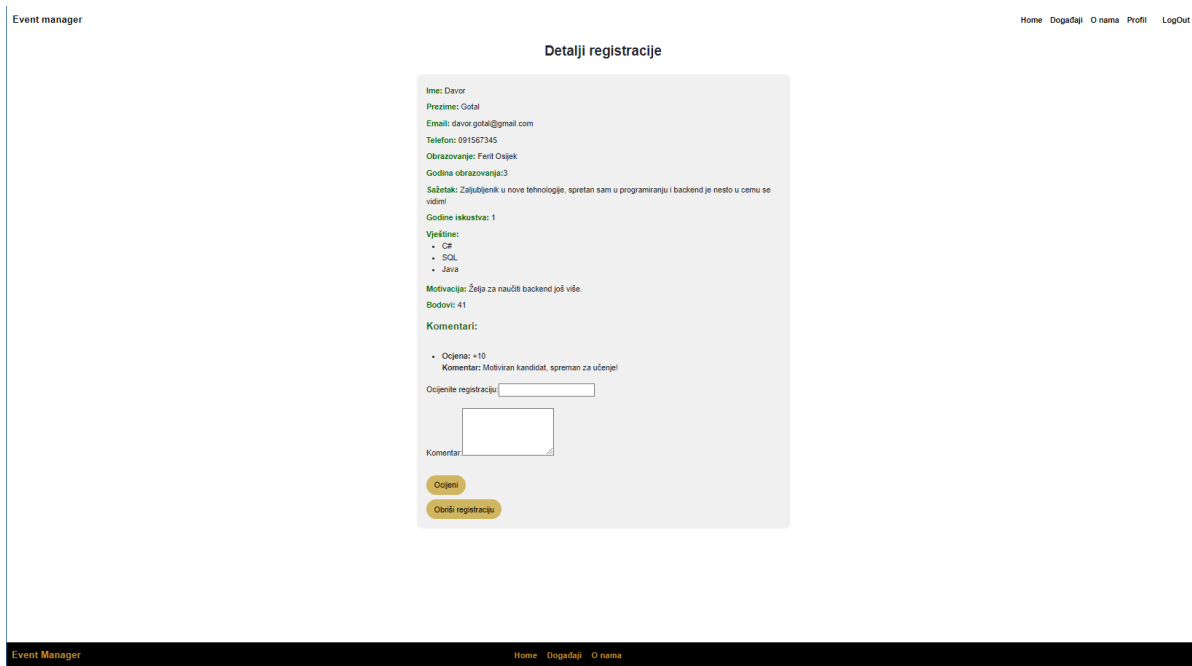


Slika 5.20. EventDetailsComponent komponenta s registracijama kandidata

Gumb „Detalji“ vodi do komponente *RegistrationDetails*, prikazane na slici 5.21, gdje su dostupni svi detalji o registraciji pojedinog kandidata. Osim toga, korisnik ima mogućnost dodavanja komentara i broja bodova, bilo pozitivnih ili negativnih, kako bi izrazio svoje mišljenje o registraciji. Mogućnost dodavanja pozitivnih bodova uz određeni komentar i uspješnost postavljanja komentara prikazani su na slici 5.22. Uz to postoji mogućnost brisanja određenih registracija, klikom na gumb „Obriši registraciju“.

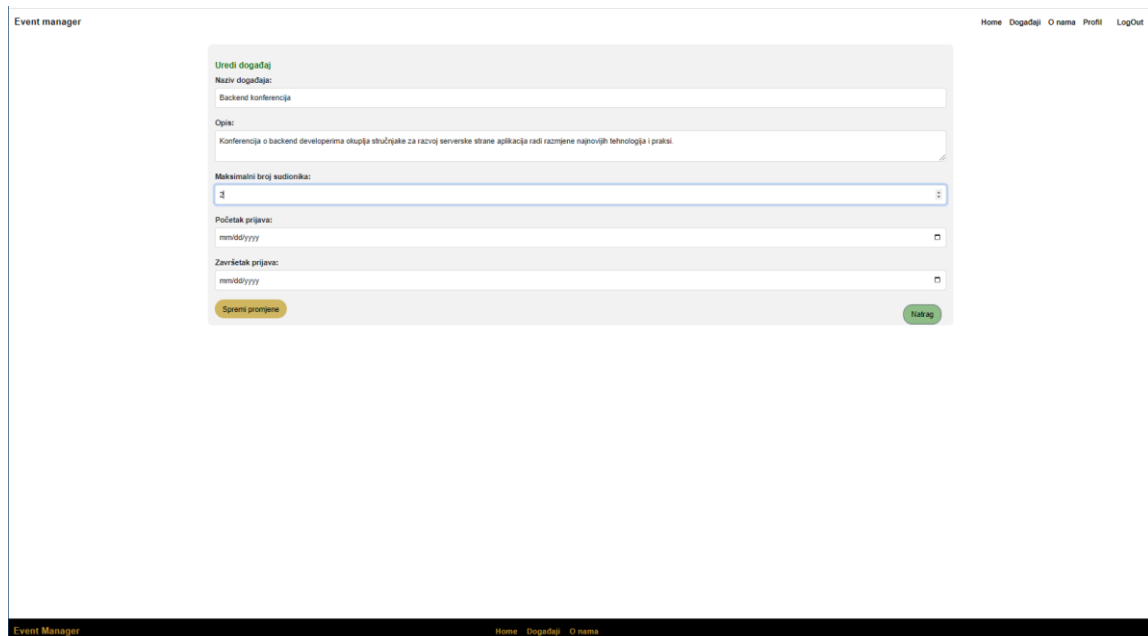


Slika 5.21. RegistrationDetailsComponent komponenta



Slika 5.22. RegistrationDetailsComponent komponenta s komentarom

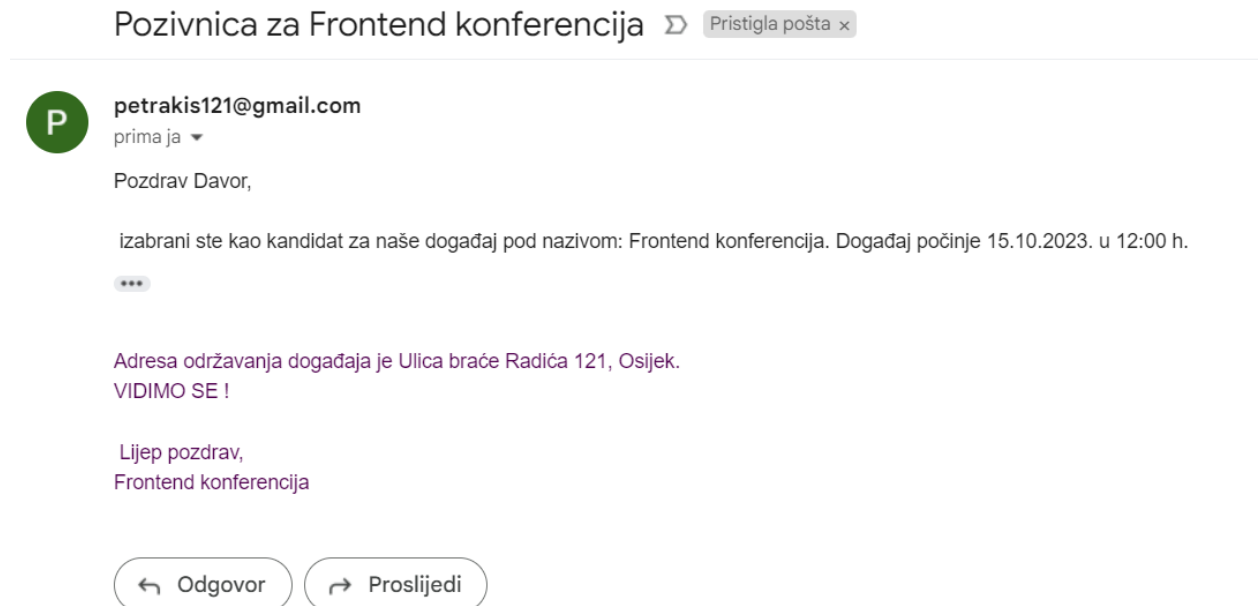
Stranica *Detalji događaja*, ranije prikazana na slici 5.20., sadrži gumb „*Uredi događaj*“. Zbog jednostavnosti, broj sudionika se uređuje kako bi se lakše objasnila funkcija „*Pozovi sudionike*“ koja se, također, nalazi na stranici *Detalji događaja*. Uređivanje broja sudionika prikazano je na slici 5.23.



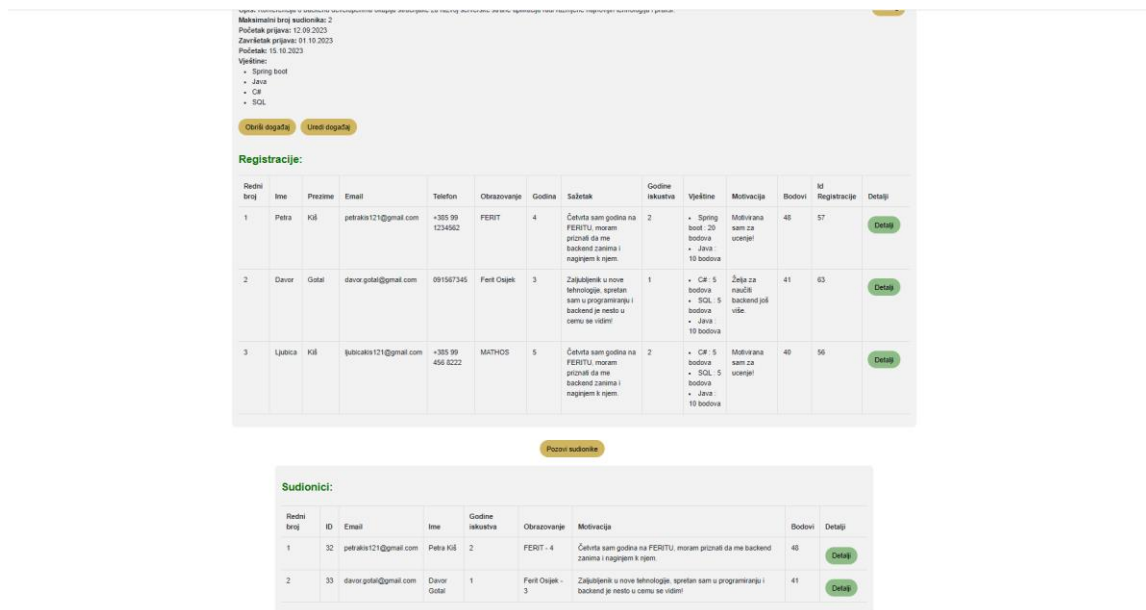
Slika 5.23. EventEditComponent komponenta

Klikom na gumb „*Pozovi sudionike*“ prvih N registriranih sudionika po broju bodova, gdje je N prag koji je postavio organizator događaja, dobiva pozivnicu (u ovom slučaju postoje dva

takva sudionika), pozivnica dolazi u obliku elektroničke pošte prikazane na slici 5.24. Tablica *Sudionici* se izrađuje nakon klika na gumb „Pozovi sudionike“, prikazana na slici 5.25., ima dva sudionika koji su po broju bodova u registracijama bili na vrhu.



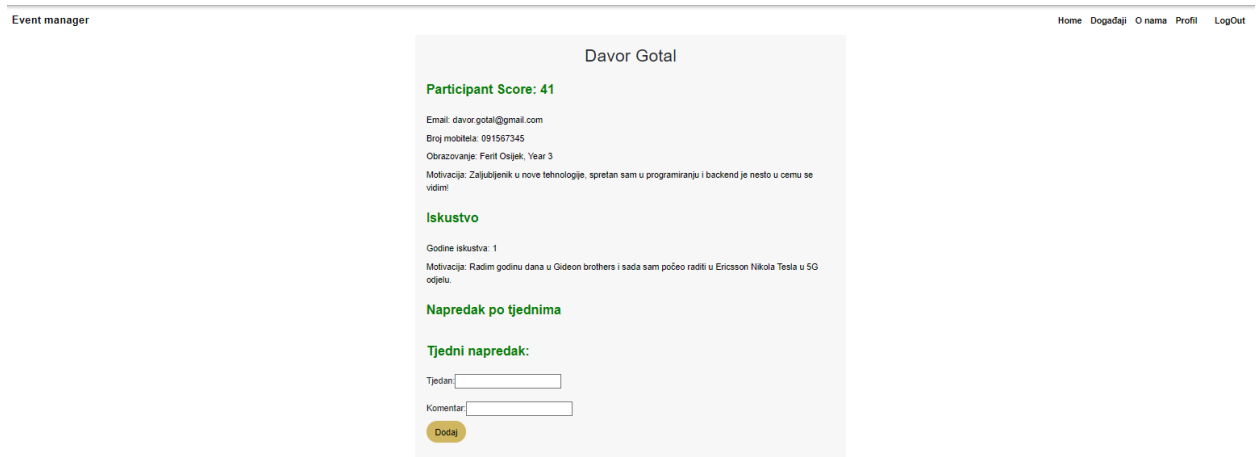
5.24. Primjer elektroničke pošte jednog od sudionika



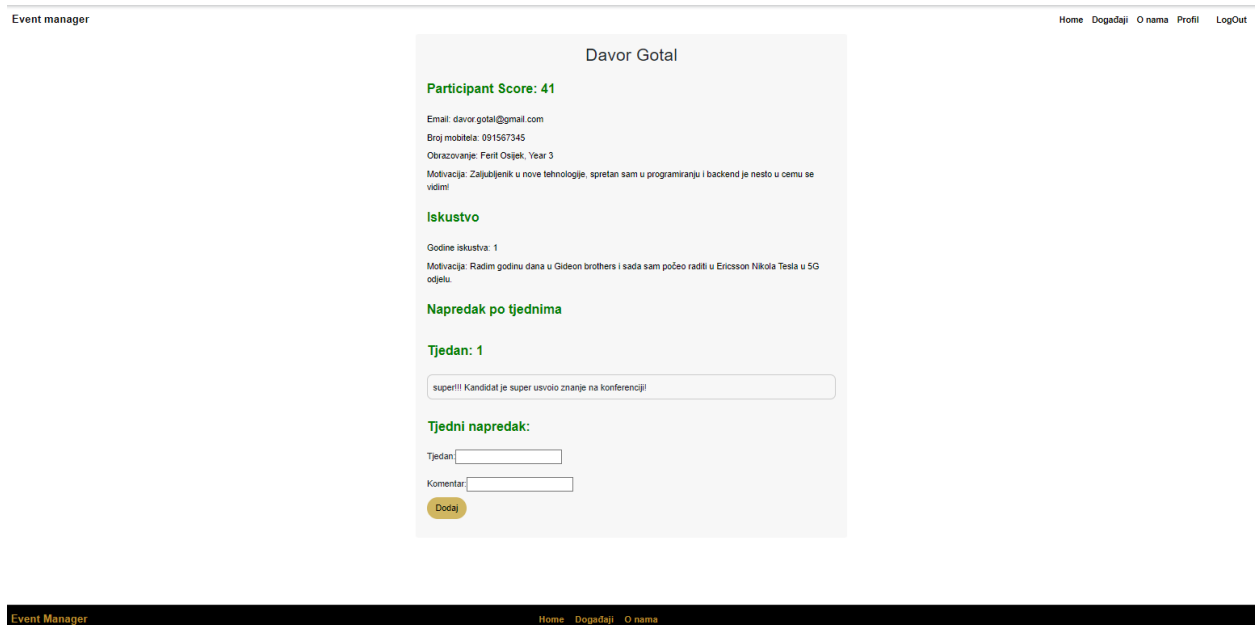
Slika 5.25. Tablica *Sudionici*

Slika 5.26. prikazuje *ParticipantComponent* komponentu koja sadrži informacije o svakom sudioniku događaja, uključujući njegovu e-mail adresu, obrazovanje, motivaciju i slične detalje

važne za samog sudionika. Organizator događaja ima mogućnost pažljivog praćenja napretka sudionika kako bi osigurao njihovo kvalitetno iskustvo na događaju. Napredak sudionika u događaju, ispunjavanje zadataka, sudjelovanje i aktivnost, prati se tjedno, pri čemu svaki tjedan može imati više komentara. Također, organizator pruža dodatne resurse i podršku sudionicima temeljem tih informacija. Slika 5.27. prikazuje uspješno dodijeljen tjedni napredak.



Slika 5.26. ParticipantComponent komponenta



Slika 5.27. ParticipantComponent komponenta

### 5.3. Izrada poslužiteljskog dijela aplikacije

Prvi korak pri izradi poslužiteljskog dijela aplikacije je u Spring Initializr web aplikaciji stvoriti Spring Boot strukturu projekta sa svim ovisnostima koje su potrebne u projektu. Neke od ovisnosti koje su uključene u poslužiteljski dio završnog rada su Spring Web, Spring Data JPA i Spring Security. Spring Web je ovisnost koja omogućava izgradnju web aplikacija pomoću Spring MVC-a opisanog u poglavlju 4. Spring Data JPA omogućava integraciju s JPA (engl. *Java Persistence API*) za pristup relacijskim bazama podataka. Spring Security omogućava dodavanje sigurnosnih značajki, poput autentikacije i autorizacije korisnika. Nakon toga potrebno je spojiti ranije napravljenu bazu podataka s poslužiteljskim dijelom aplikacije na način prikazan na slici 5.28.

```
spring.datasource.url=jdbc:postgresql://127.0.0.1:5432/Hackathon
spring.datasource.username=postgres
spring.datasource.password=password
spring.datasource.driver-class-name=org.postgresql.Driver
```

Slika 5.28. Spajanje s bazom podataka

Svaka Spring Boot aplikacija mora sadržavati anotaciju `@SpringBootApplication` u glavnoj klasi aplikacije koja omogućava automatsko skeniranje putanja do glavnih dijelova aplikacije. Organizacija poslužiteljskog dijela aplikacije strukturirana je u tri glavna dijela: repozitorij, servis i upravljač. Pomoću paketa repozitorij (engl. *Repository*) s anotacijom `@Repository` aplikacija komunicira s bazom. Servis (engl. *Service*) s anotacijom `@Service` sadrži logiku i funkcionalnost. Upravljač (engl. *Controller*) s anotacijom `@RestController` koristi se zbog obrade HTTP zahtjeva. Upravljači ostvaruju komunikaciju s korisničkim sučeljem, primaju podatke od klijenta te usmjeravaju komunikaciju do odgovarajućih servisa i vraćaju odgovore klijentima, usmjeravaju i obrađuju HTTP zahtjeve. Servisi predstavljaju logiku aplikacije, obrađuju zahtjeve kontrolera, izvršavaju logiku i komuniciraju s bazom podataka. Repozitorij je sučelje koje se također koristi za interakciju s bazom podataka, služi kao most između aplikacije i baze podataka. Repozitorij ima mogućnost izvođenja operacija brisanja, dohvaćanja i spremanja podataka u bazu podataka. Poslužiteljski dio aplikacije ostvaruje komunikaciju s bazom podataka i klijentima putem REST aplikacijskog sučelja (engl. *Representational State Transferful Application Programming Interface*, REST API). Ovo sučelje omogućava klijentima da na poslužitelj šalju zahtjeve za stvaranje, čitanje, ažuriranje i brisanje podataka. Poslužiteljski API podržava sve četiri osnovne operacije izradi, čitaj, uredi i obriši (engl. *Create, read, update, delete*, CRUD), što znači da klijenti mogu stvarati nove zapise u bazi podataka preko HTTP metode POST, primjer metode se nalazi na slici 5.29. i prikazuje metodu koja sprema događaje koje klijent upisuje putem servisa. Također, klijenti mogu dohvaćati postojeće podatke preko HTTP metode



GET. Podaci se mogu ažurirati na dva načina preko HTTP metode PUT i HTTP metode PATCH. HTTP metoda PUT ažurira cijeli objekt na web poslužitelju, dok HTTP metoda PATCH koristi parcijalno ažuriranje postojećeg objekta na web poslužitelju. HTTP metoda DELETE briše nepotrebne podatke. Ovo pruža potpunu funkcionalnost za upravljanje podacima putem API poziva.

```
@PostMapping("/event")
public ResponseEntity<String> save(@RequestBody EventRequestDto
eventRequestDto) {
    return eventService.save(eventRequestDto);
}
```

Slika 5.29. Metoda u EventControlleru

Primjer povezivanja servisa sa sučeljem repozitorija je prikazan na slici 5.30. u kojem se vidi ranije spomenuta manipulacija bazom podataka preko sučelja *eventRepository* repozitorija. Prikazana metoda sprema događaj koji je poslan na servis, provjerava postoji li već u bazi podataka događaj s istim imenom i ako postoji vraća odgovor.

```

public ResponseEntity<String> save(EventRequestDto eventRequestDto) {

    Event event = eventMapper.toEntity(eventRequestDto);
    List<Event> events = eventRepository.findAll();

    for(Event e: events) {
        if (e.getName().equals(event.getName())) {
            return ResponseEntity.badRequest().body("An event with the same
name already exists.");
        }
    }

    eventRepository.save(event);
    URI locationUri= ServletUriComponentsBuilder
        .fromCurrentRequest()
        .path("/{eventId}")
        .buildAndExpand(event.getEventId())
        .toUri();

    return ResponseEntity.created(locationUri).build();
}

```

Slika 5.30. Primjer metode u implementaciji Event servisa

Na slici 5.31. prikazano je sučelje repozitorij naziva *EventRepository* koji se proširuje s *JpaRepository*, *JpaRepository* u sebi sadrži osnovne metode za manipulaciju bazom podataka. Dodatno su implementirane metode *findByUserId* koji pretražuje događaje po korisničkom id podatku i *findByNameContaining* koji pronalazi događaj po imenu.

```

public interface EventRepository extends JpaRepository<Event, Long>,
PagingAndSortingRepository<Event, Long> {

    @Query("SELECT e FROM Event e WHERE e.confirmationNotAfter < CURRENT_DATE
AND e.invitesSent = false")
    List<Event> findAllWhereConfirmationDateExpiredAndInvitesSentFalse();
    @Query("SELECT e FROM Event e WHERE e.appUser.userid = :userId")
    List<Event> findByUserId(@Param("userId") Long userId);

    Page<Event> findByNameContaining(String name, Pageable pageable);
}

```

Programski kod 5.31. Prikaz sučelja repozitorij

Automatski odabir kandidata, kako je opisan u klasi *ScoringEngine* i prema slici 5.32., predstavlja proces ocjenjivanja korisnika na temelju različitih čimbenika kako bi se utvrdilo tko će biti odabran za sudjelovanje u određenom događaju. Ovaj proces koristi bodovanje kako bi svakom korisniku dodijelio odgovarajući broj bodova na temelju njihovog obrazovanja, godina iskustva i vještina. Ključna komponenta ovog procesa je klasa *ScoringEngine*. Prvi korak u

procesu bodovanja je uzimanje podataka o godinama obrazovanja koje je korisnik naveo tijekom registracije.

Ovaj proces bodovanja može se prikazati formulom kako slijedi:

$$\begin{aligned} \text{Bodovi za obrazovanje} &= (\text{Broj godina obrazovanja}) * 2 \\ \text{Bodovi za iskustvo} &= (\text{Broj godina iskustva}) * 5 \end{aligned}$$

*Primjer:*

*Ako korisnik ima 5 godina obrazovanja, bodovi za obrazovanje = 5 \* 2 = 10 bodova.*

*Ako korisnik ima 3 godine iskustva, bodovi za iskustvo = 3 \* 5 = 15 bodova.*

*Ukupni bodovi korisnika = Bodovi za obrazovanje + Bodovi za iskustvo*

Kako bi se dodatno ocijenio korisnik, uzima se u obzir broj bodova za vještine koje je korisnik odabrao tijekom prijave. Ovaj broj bodova određuje organizator događaja i temelji se na tome koliko su te vještine relevantne za taj specifičan događaj. Nakon što su sve ove komponente bodovanja izračunate, zbrajaju se kako bi se dobila ukupna ocjena korisnika. Korisnik s višim brojem bodova ima veću šansu biti automatski odabran kao kandidat za događaj.

```
@RequiredArgsConstructor
public class ScoringEngine {
    private final SkillRepository skillRepository;
    private Integer yearsOfEducationMultiplier = 2;
    private Integer yearsOfExperienceMultiplier = 5;

    public Integer CalculateScore(RegistrationRequestDto registrationRequest) {
        Integer yearsOfEducationPoints =
registrationRequest.getPersonal().getEducation().getYear() *
yearsOfEducationMultiplier;
        Integer yearsOfExperiencePoints =
registrationRequest.getExperience().getYears() * yearsOfExperienceMultiplier;
        int sum=0;
        for( SkillNameRequestDto skillDto :
registrationRequest.getExperience().getSkills()) {
            Skill skill =
skillRepository.findFirstByName(skillDto.getName());
            int point = skill.getPoints();
            sum += point;
        }

        return yearsOfEducationPoints + yearsOfExperiencePoints + sum;
    }
}
```

*Programski kod 5.32. Klasa ScoringEngine*

## 6. VREDNOVANJE APLIKACIJE IZ PERSPEKTIVE KORISNIKA APLIKACIJE

Korisničko iskustvo igra ključnu ulogu u oblikovanju uspješnih i korisnih proizvoda ili usluga. Razumijevanje kako korisnici doživljavaju web aplikaciju za planiranje događaja iznimno je važno zbog pružanja bolje usluge i poboljšavanja korisničkog zadovoljstva. U tu svrhu provedena je usluga za anketiranje Google Forms među petnaest studenata Fakulteta elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Anketa obuhvaća različite aspekte korištenja web aplikacije završnog rada, uključujući registraciju, funkcionalnosti i sve ostalo što se odnosi na korisničko iskustvo. Pitanja su u obliku odgovora s višestrukim izborom, kratkim odgovorom i ocjenjivanja po Likertovoj skali.

### 6.1. Odgovori s višestrukim izborom



Slika 6.1. Analiza odgovora na prvo pitanje

Iz prvog pitanja sa slike 6.1. vidi se da bi većina korisnika intuitivno pretražila rješenja problema organiziranja događaja na web preglednicima.



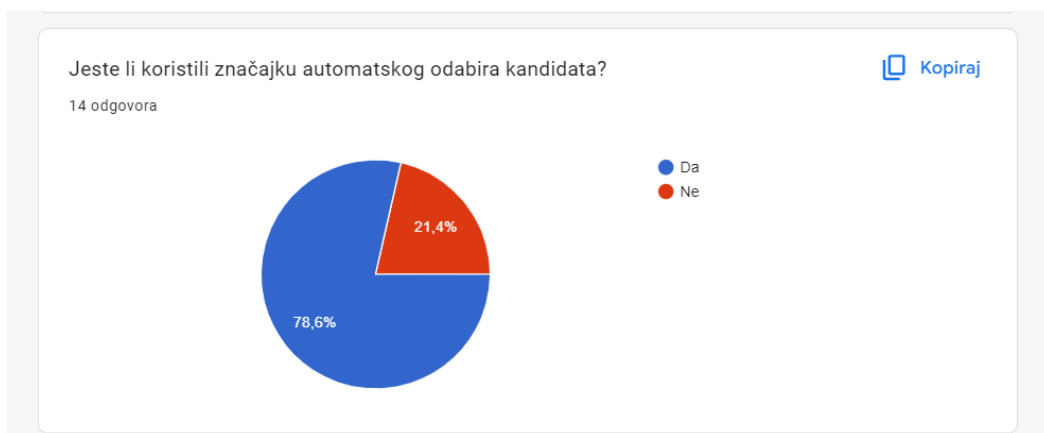
Slika 6.2. Analiza odgovora na drugo pitanje

Iz rezultata odgovora na pitanje prikazano na slici 6.2. evidentna je jednostavnost registracije za većinu korisnika te izostanak većih problema prilikom iste. U svakoj aplikaciji bitno je imati sigurnu i jednostavnu prijavu.



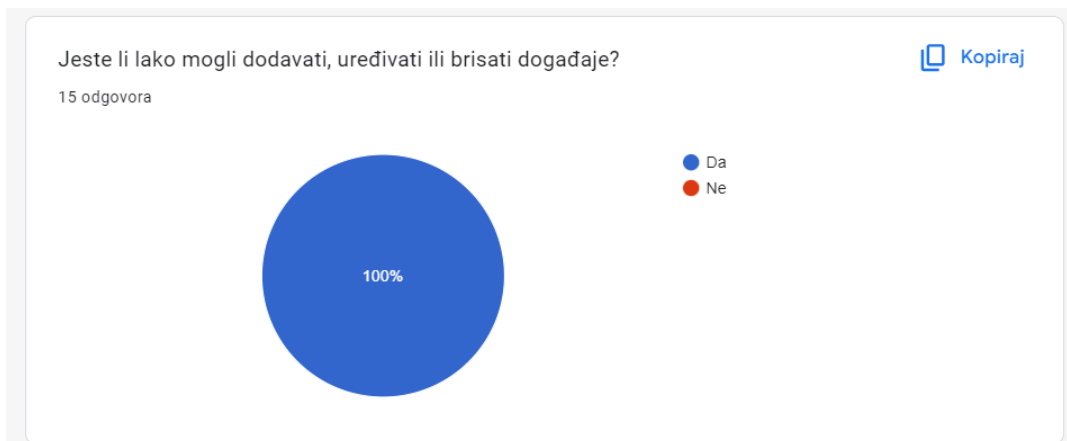
Slika 6.3. Analiza odgovora na treće pitanje

Slika 6.3. prikazuje odgovor o jasnoći korištenja web aplikacije. Korisnici web aplikacije završnog rada shvaćaju način rada web aplikacije, iz toga se zaključuje da je sučelje aplikacije jednostavno napravljeno.



Slika 6.4. Analiza odgovora na četvrto pitanje

Iz slike 6.4. zaključuje se da je većina korisnika izradila svoj događaj i pozivala kandidate na njega. Mali broj korisnika je aplikaciju koristio anonimno i samo za prijavu na događaj.



Slika 6.5. Analiza odgovora na peto pitanje



Slika 6.6. Analiza odgovora na šesto pitanje



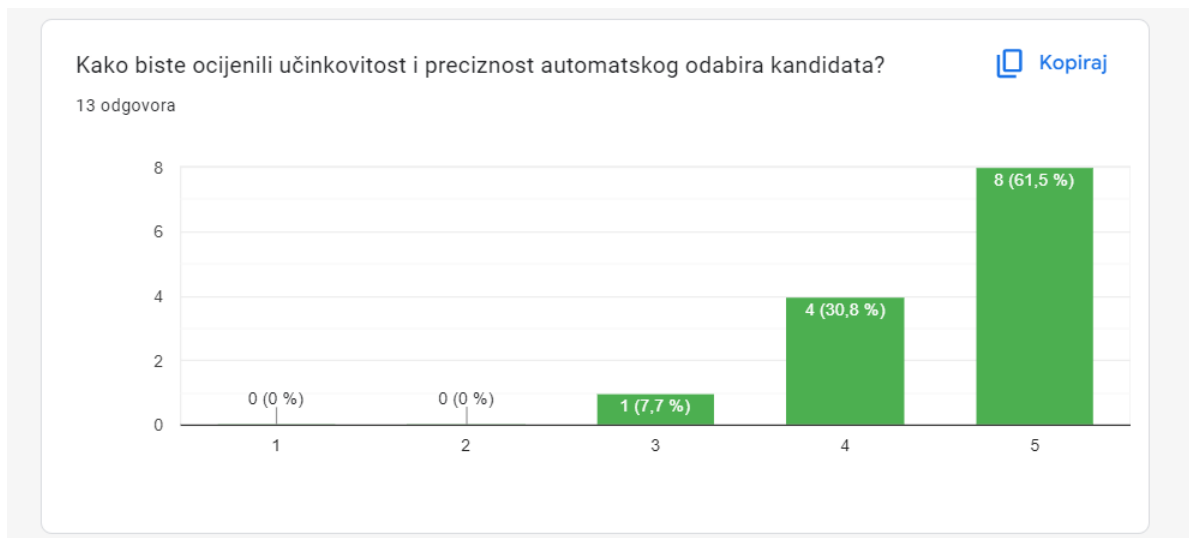
Slika 6.7. Analiza odgovora na sedmo pitanje

Iz analize odgovora vidljivih na slikama 6.5., 6.6. i 6.7. može se zaključiti da su korisnici mogli lako uređivati, dodavati i brisati događaje, da su svi primili obavijest o uspješnosti prijave ili sudjelovanja na događaju na elektroničku postu i da bi preporučili web aplikaciju završnog rada i

drugim korisnicima. Rezultati ova tri pitanja su svi jednoglasni i puno znače za vrednovanje korisničkog iskustva.

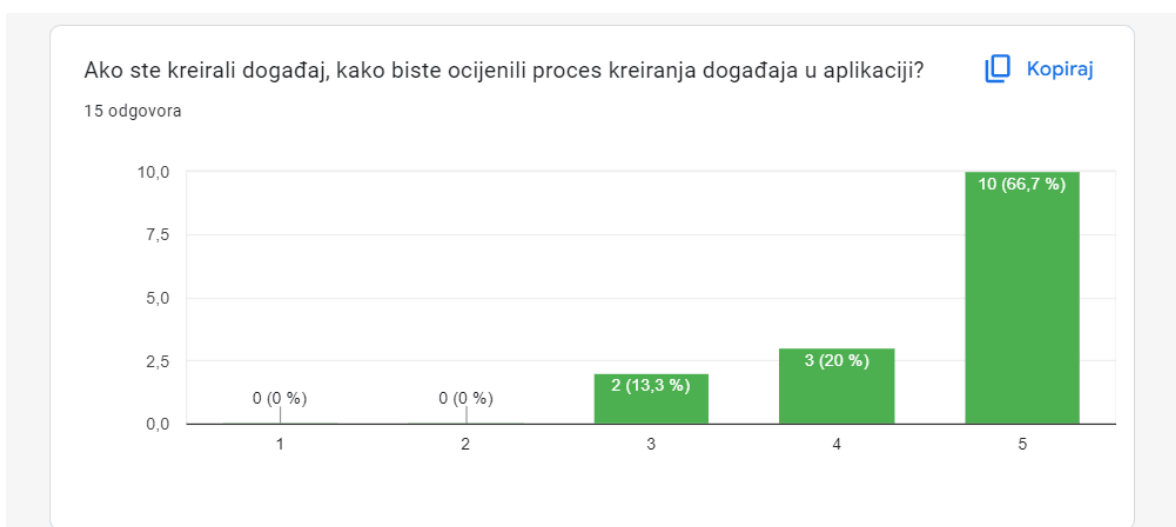
## 6.2. Odgovori na pitanja po uzoru na Likertovu skalu

Odgovori s linearnim mjerilom omogućuju kvantitativno procjenjivanje odgovora sudionika na određena pitanja, olakšava analizu korisničkih preferencija i rangiraju različite stavove. U anketi su postavljena tri takva pitanja.



Slika 6.8. Graf analiza prvog pitanja

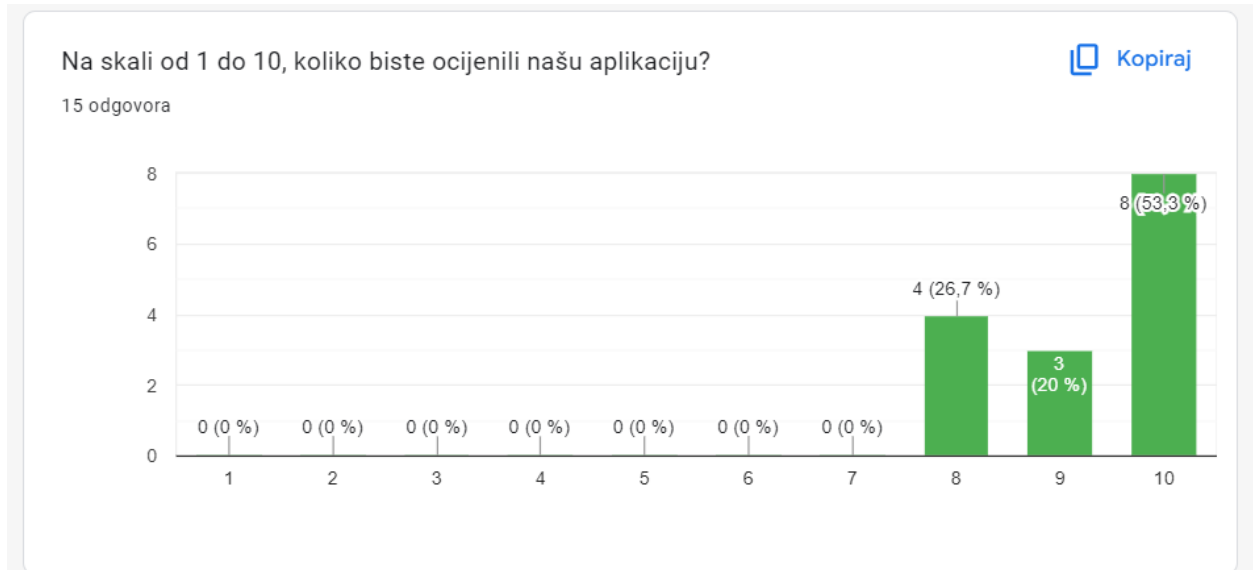
Na slici 6.8. vidljivo je da su korisnici automatskom odabiru kandidata pretežito dali ocjene odličan ili vrlo dobar. Nakon računanja prosjeka, ukupna ocjena iznosi 4.54 što je odličan.



Slika 6.9. Graf analiza drugog pitanja

Korisnici koji su izradili događaj u web aplikaciji završnog rada, ocijenili su ga pretežito ocjenom odličan, analitika je prikazana na slici 6.9.. Prosječna ocjena je 4,53.

Pitanje 3:



Slika 6.10. Graf analiza trećeg pitanja

Generalno iskustvo korisnika web aplikacije završnog rada prikazano na slici 6.10. je 9.27 u prosjeku. Iz toga se zaključuje da je aplikacija jednostavna, intuitivna i korisna.

### 6.3. Kratki odgovori

U anketi postoje dva pitanja iz kategorije kratkih odgovora.

Pitanje 1: Napišite svoju kratku recenziju na iskustvo korištenja web aplikacije

Neki od odgovora su:

- „Jednostavno i pristupačno, ne naplaćuje se za razliku od drugih.“
- „Potrebno mi je uvesti još jedan način sortiranja događaja, uz pretraživanje i po datumu, inače mi se aplikacija sviđa i lagana je za koristiti.“
- „Mislim da je sama ideja aplikacije inovativna i da uz ostale ponuđene izbore ova najbolje odgovara mojim potrebama.“
- „Registracija je jednostavna i prijava nakon nje je automatska, odabir kandidata je točan i po broju bodova.“



Pitanje 2: Koje biste nove značajke ili poboljšanja željeli vidjeti u ovoj web aplikaciji?

Neki od odgovora su:

- „Izgled bi mogao biti moderniji.“
- „Više kriterija za automatski odabir.“
- „Kategorizacija događaja po naprimjer vrsti događaja“

Iz svih odgovora na pitanja, zaključuje se da su korisnici pretežito zadovoljni web aplikacijom završnog rada, postoji prostora za promjenom i nadogradnjom aplikacije. Moguće promjene mogu biti uređenje korisničkog sučelja, kategorizacija događaja, poboljšavanje automatskog bodovanja.

## 7. ZAKLJUČAK

U okviru ovog završnog rada razvijena je web aplikacija za planiranje događaja s automatskim odabirom kandidata. Cilj završnog rada bio je stvoriti web aplikaciju koja će imati mogućnost digitalne organizacije događaja, automatsko rangiranje kandidata i pozivanje na taj događaj, praćenje registracija i sudionika i slanje potvrda o uspješnosti prijave putem elektroničke pošte. Istraženo je područje digitalnih rješenja za organizaciju raznih događaja, zatim su istaknute prednosti i mane svake od navedenih web aplikacija uz usporedbu s web aplikacijom završnog rada. Postojeća rješenja poslužila su i postavljanju zahtjeva na programsko rješenje izrađeno u okviru ovog rada, i to u obliku korisničkih priča. Na osnovu specifikacije zahtjeva izabrane su tehnologije koje su se koristile u izradi web aplikacije završnog rada: HTML, CSS, TypeScript, Angular, Java, Spring Boot, SQL, PostgreSQL.

Značajke kao što su automatski odabir kandidata, jednostavno planiranje događaja i fleksibilnost u postavljanju kriterija ističu se kao najvažniji rezultati ovog rada. Osim toga, uspješno ispitivanje aplikacije nad stvarnim korisnicima ukazuje na njen potencijal za praktičnu primjenu u organizaciji događaja različitih vrsta.. Uz pozitivne stavke koje su naveli, korisnici su imali i primjedbe. Predloženo je uređivanje korisničkog sučelja kako bi bilo modernije, mogućnost pretraživanja događaja po kategoriji u koju događaji pripadaju i dodatni alati za analizu rezultata pri odabiru kandidata. Web aplikacija završnog rada pruža osnovu za daljnje istraživanje i razvoj aplikacije za planiranje događaja s automatskim odabirom kandidata te potvrđuje njenu potencijalnu vrijednost u pojednostavljenju procesa organizacije događaja. Kroz kontinuirani dijalog s korisnicima i prilagodbu njihovim potrebama, moguće je unaprijediti korisničko iskustvo i osigurati uspješnu primjenu ovakvih rješenja u stvarnom svijetu.

## LITERATURA

- [1] Bringing the world together through live experiences, Eventbrite 2023, dostupno na: <https://www.eventbrite.com/about/> [datum zadnjeg pristupa: 14.9.2023.]
- [2] What you can do?, Meetup LLC 2023. dostupno na: <https://www.meetup.com/about/> [datum zadnjeg pristupa 14.9.2023.]
- [3] About EventZilla, Eventzilla, dostupno na: <https://www.eventzilla.net/us/about> [datum zadnjeg pristupa 14.9.2023.]
- [4] About Cvent, Cvent Inc 2023. dostupno na: <https://www.cvent.com/en/company-overview> [datum zadnjeg pristupa 14.9.2023.]
- [5] Build better events with Bizzabo, Bizzabo 2023. dostupno na: <https://www.bizzabo.com/our-team> [ datum zadnjeg pristupa 14.9.2023.]
- [6] User stories, Mountain Goat Software, dostupno na: <https://www.mountaingoatsoftware.com/agile/user-stories> [datum zadnjeg pristupa 10.9.2023.]
- [7] T. A. Powell, HTML & CSS: The Complete Reference, Fifth Edition, McGraw Hill, 2010.
- [8] About, Bootstrap Authors 2023. dostupno na: <https://getbootstrap.com/docs/4.1/about/overview/> [datum zadnjeg pristupa 10.4.2023.]
- [9] What is TypeScript?, Microsoft 2023 , dostupno na: <https://www.typescriptlang.org/> [datum zadnjeg pristupa 10.9.2023.]
- [10] What is Angular?, Google 2023. dostupno na: <https://angular.io/guide/what-is-angular>. [datum zadnjeg pristupa 12.9.2023.]
- [11] What is Java technology and why do I need it?, Oracle 2023. dostupno na: [https://www.java.com/en/download/help/whatis\\_java.html](https://www.java.com/en/download/help/whatis_java.html) [datum zadnjeg pristupa 8.9. 2023.]
- [12] Spring boot Overview, Vmware, Inc 2023. dostupno na: <https://spring.io/projects/spring-boot> [datum zadnjeg pristupa 14.9 .2023.]
- [13] WHAT IS SQL?, SQLCourse 2022. dostupno na: <https://www.sqlcourse.com/beginner-course/what-is-sql/> [datum zadnjeg pristupa 14.9.2023.]
- [14] What is PostgreSQL?, The PostgreSQL Global Development Group 2023. dostupno na: <https://www.postgresql.org/about/> [datum zadnjeg pristupa 14.9.2023.]

[15] What is ER Diagram?, Guru99 2023. dostupno na: <https://www.guru99.com/er-diagram-tutorial-dbms.html> [datum zadnjeg pristupa 4.9.2023.]

## SAŽETAK

Povijesno gledano, organizacija događaja uključivala je ručno vođenje dokumenata o događajima i često složene postupke za prikupljanje informacija o sudionicima. Zahvaljujući digitalizaciji, organizatori mogu brzo i efikasno upravljati događajima putem web aplikacija, pojednostavljajući korisničko iskustvo organizacije i prijave na događaje.

Web aplikacijom za planiranje događaja s automatskim odabirom kandidata postignut je cilj pojednostavlivanja organizacije događaja. Organizatori događaja sami postavljaju tražene vještine sudionika, po kojima se tada prijavljeni sudionici automatski rangiraju. Zahtjevi na navedenu web aplikaciju prikazani su u obliku korisničkih priča, a ona je implementirana pomoću programskih tehnologija Angular, Spring boot i PostgreSQL. Anketa o korisničkom iskustvu provedena je nad petnaest korisnika u kojoj su se otkrile prednosti i mane rješenja web aplikacije završnog rada.

Zaključno, implementacija web aplikacije završnog rada ispunila je glavne ciljeve kao što je automatski odabir kandidata i fleksibilnost odabira kriterija, ali ima mjesta iza poboljšanja i izmjene.

**Ključne riječi:** Angular, događaji, Spring boot, web aplikacija

## **ABSTRACT**

Title: Event planning web application with automated candidate selection

From a historical perspective, event organization involved manual record-keeping of event-related documents and often complex procedures for gathering participant information. Thanks to digitalization, organizers can now efficiently manage events via web applications, thus simplifying the user experience for event organization and registration.

The goal of simplifying event organization has been achieved through a web application for event planning with automatic candidate selection. Event organizers set the desired participant skills, based on which registered participants are automatically ranked. Requirements for this web application are presented in the form of user stories and the application itself has been implemented using programming technologies such as Angular, Spring Boot, and PostgreSQL. A user experience survey was conducted with fifteen users, revealing the strengths and weaknesses of the web application.

Web application has met its main objectives, such as automatic candidate selection and flexibility in criteria selection. However, there is room for improvement and modification.

**Keywords:** Angular, events, Spring Boot, web application

## **PRILOZI**

Sav programski kod korišten za izradu web aplikacije nalazi se na poveznicama:

Backend: <https://github.com/kispetra/Event-manager-backend>

Frontend: <https://github.com/kispetra/Event-manager-frontend>

## **ŽIVOTOPIS**

Petra Kiš rođena je 12.1.2002. godine u Osijeku. Osnovnu školu je pohađala u Tenji, nakon nje upisuje Opću gimnaziju Osijek i maturira 2020. godine. Iste godine upisuje sveučilišni preddiplomski studij računarstva, smjer računalno inženjerstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

---

Potpis autora