Univerzitet Crne Gore  
ELEKTROTEHNIČKI FAKULTET  
Studije primijenjenog računarstva

**Baze podataka (napredni kurs)**  
-dokumentacija mini-projekta-

Tema: **Sound's bazaar**

Učesnici u projektu:

1. Amil Đešević 71/21 – lider grupe
2. Samed Mehmedović 107/21
3. Matija Gajović 97/21
4. Luka Zlaić 6/21
5. Damir Pilica 4/21

Podgorica 23.5.2024. godine

# **1. Uvod (o projektu)**

Ideja tima za kreiranje ove *web* aplikacije proizilazi iz ljubavi prema muzici. **Sound bazaar** je inovativan naziv za *web* aplikaciju koja će svojim funkcionalnostima podsjećati na već postojeće *web* aplikacije kao što su Spotify, Sound Cloud, Deezer, Youtube i ostale.



Slika br. 1 – Logo platforme “Sound’s Bazaar”

Ova *web* aplikacija će pružati isključivo muzički sadržaj. Osim snimljenih kompozicija u standardnim formatima (*.mp3, .mp4*) na platformi bi postojali i drugi formati muzičkih zapisa (*.midi, .wave, .adg*) koji bi bili namijenjeni za kompozitore, producente i sve ostale ljude koji žele stvarati svoja muzička djela.

Korisnici na ovoj platformi bi mogli preuzimati muziku ukoliko je admin koji je dodao tu melodiju dozvolio preuzimanje. Korisnici, *stakeholder-i* koji mogu biti dio aktivnosti na ovoj platformi su ljudi koji ažuriraju platformu, dodaju muziku i upravljaju njome (admini, kompozitori, producenti) i korisnici kojima će taj muzički sadržaj biti dostupan.

Korisnici će moći da kreiraju odgovarajuće *playlist*-e, da sačuvaju pojedinačno neku muzičku melodiju ili je preuzmu na svom uređaju. Sve melodije na ovoj *web* aplikaciji je moguće besplatno preslušati, dok je preuzimanje dozvoljeno korisnicima samo ako je admin sistema odobrio preuzimanje.

Osim ovih funkcionalnosti, sistem treba da ima opciju registracije korisnika i prijavljivanja na svoj nalog na aplikaciji gdje će svaki korisnik imati svoj nalog na kojem će se nalaziti sačuvana muzika ili *playlist*-a.

Grupa inspirisana ovim projektom ima za cilj da ponudi sličnu, ali jedinstvenu uslugu koja će kombinovati aspekte servisa vezanih za muzičke melodije i kompozicije, uz dodatne navedene funkcionalnosti.

# **2. Opis opštih funkcionalnosti**

U ovom poglavlju će biti opisane opšte funkcionalnosti sistema do kojih je došlo tokom faze planiranja, analize i dizajna sistema. Korišćenjem tekstualnih i grafičkih modela olakšaće se implementacija tj. programiranje samog sistema.

## **2.1 Funkcionalni i nefunkcionalni zahtjevi**

Funkcionalni i nefunkcionalni zahtjevi igraju ključnu ulogu u razvoju softverskih aplikacija jer definišu šta sistem treba da radi (funkcionalni zahtjevi) i kako bi sistem trebao da radi (nefunkcionalni zahtjevi).

**Funkcionalni zahtjevi** za ovaj projekat su:

* **Registracija korisnika** – Korisnici mogu kreirati (registrovati) svoj nalog putem forme za registraciju. Potrebno je validirati podatke prilikom registracije (e-mail adresa, lozinka i ostalo).
* **Prijava korisnika** – Ovaj funkcionalni zahtjev je vezan za registraciju korisnika. Nakon što se korisnik registruje na sistem, on se može prijaviti sa svojim podacima. Potrebno je implementirati autentifikaciju korisnika koristeći e-mail adresu i lozinku.
* **Stranica za muziku** – Treba da postoji stranica za muziku na kojoj će korisnici moći da pretražuju i slušaju određene numere. Pomoću modalnog prozora korisnicima se otvara odabrana pjesma i mogu je dodati na svoju *playlist*-u.
* **Kreiranje i upravljanje playlistama** – Korisnik može kreirati *playlist*-u u kojoj bi dodavao muziku. Prilikom kreiranja *playlist*-e korisnik unosi naziv liste (npr. Playlist 1) i opis (npr. žanr te liste). Svaka *playlist*-a treba da ima opciju (dugme) za dodavanje još numera unutar nje.

**Nefunkcionalni zahtjevi** za ovaj projekat su:

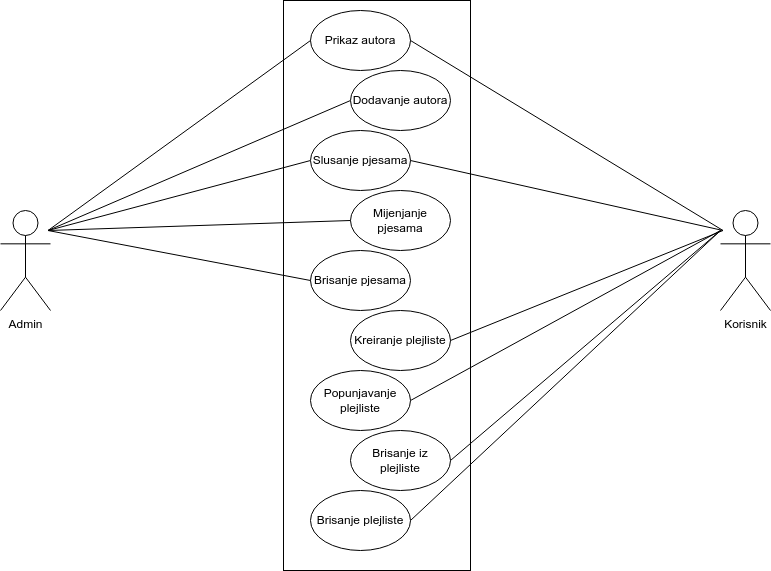
* **Performanse** – Aplikacija mora brzo reagovati na korisničke zahtjeve, sa minimalnim vremenom učitavanja. Sistem mora podržavati veliki broj ljudi istovremeno na platformi.
* **Sigurnost** – Ovaj nefunkcionalni zahtjev obuhvata implementaciju zaštite od sigurnosnih prijetnji (SQL inekcija, XSS napadi i CSRF). Prenos podataka na platformi mora biti zaštićen enkripcijom (posebno lozinke).
* **Pouzdanost** – Sistem treba da ima veliku dostupnost. Potrebno je implementirati proceduru za *backup* i *recovery* kako bi se osiguralo očuvanje podataka. Platforma treba da obuhvati SSL sertifikat kako bi se korisnici osjećali pouzdanije na platformi.
* **Prilagodljivost** – Sistem treba da bude prilagodljiv na svim uređajima (responzivnost), uključujući desktop uređaje, kao i mobilnu i tablet verziju uređaja. Aplikacija treba biti dizajnirana na način da može podržati sve veći broj korisnika i podataka u slučaju rasta platforme.

## **2.2 Analiziranje sistema**

U ovom poglavlju će se kroz *UML* dijagrame koji se koriste tokom analize sistema objasniti šta ovaj sistem treba da radi.

### **2.2.1 Use case analiza**

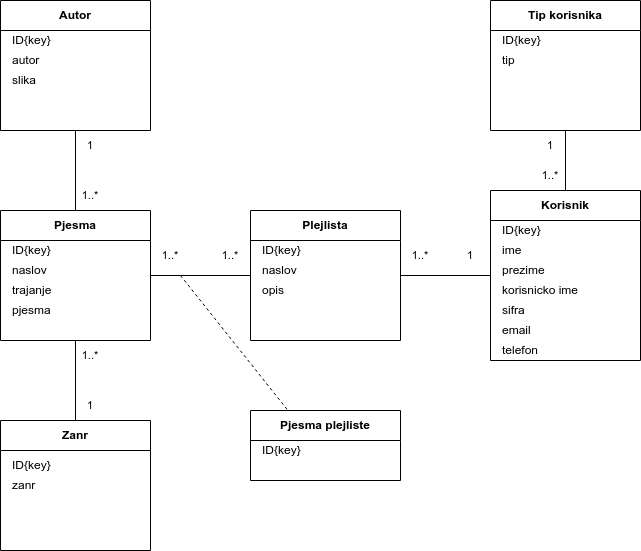
Slučaj korišćenja (*Use case*) je scenario u kojem će ovaj biti korišćen, i obično se sastoji od glagola i imenice (dodaj u korpu, naruči robu itd.). Slučajevi korišćenja u jednom sistemu se grafički prikazuju *Use Case* dijagramom.



Slika br. 2 – Use case dijagram

### **2.2.2 Dijagram klase u domenu problema**

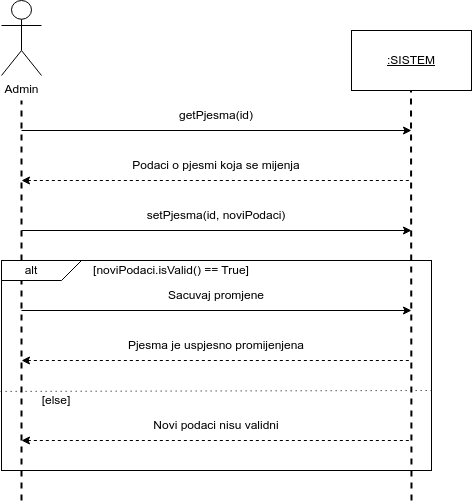
Dijagram klase u domenu problema grafički prikazuje sve klase koje će biti korišćene unutar ovog sistema. Osim klasa na ovom dijagramu su prikazani i atributi tih klasa kao i veze između klasa.



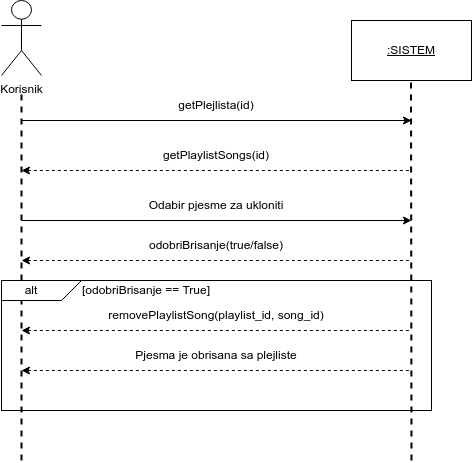
Slika br. 3 – Dijagram klase u domenu problema

### **2.2.3 *SSD (System sequence diagram)***

Dijagrami sistemske sekvence se koriste kako bi bolje objasnili komunikaciju između korisnika i sistema za pojedine slučajeve korišćenje. Najčešće se koriste za kompleksnije slučajeve korišćenja ili za one slučajeve koji nisu najjasniji. U slučaju ovog sistema napravljeni su *SSD* za dva slučaja korišćenja i to: Mijenjanje pjesama i brisanje iz plejliste.



Slika br. 4 – SSD (mijenjanje pjesama)



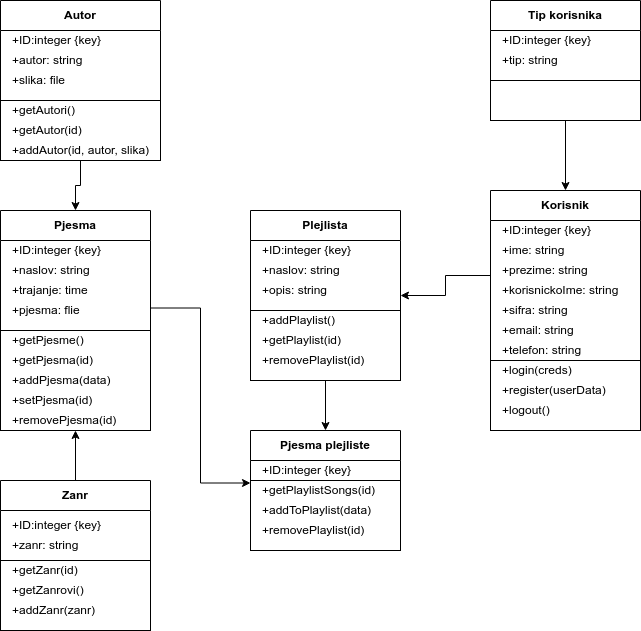
Slika br. 5 – SSD (brisanje iz playlist-e)

## **2.3 Dizajn sistema**

Dizajn informacionih sistema je faza razvoja sistema u kojoj se tektualnim i/ili grafičkim modelima opisuje kako da se sistem koji je opisan u analizi napravi.

### **2.3.1 Dijagram dizajna klase**

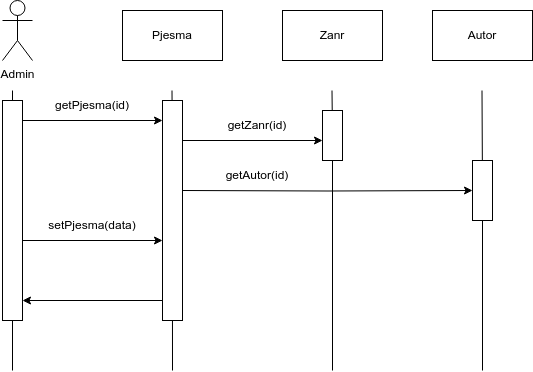
Dijagram dizajna klase je nadograđen dijagram klase u domenu problema iz analize, sa dodatkom metoda koje klase treba da implementiraju.



Slika br. 6 - Dijagram dizajna klase

### **2.3.2 Dijagram sekvence**

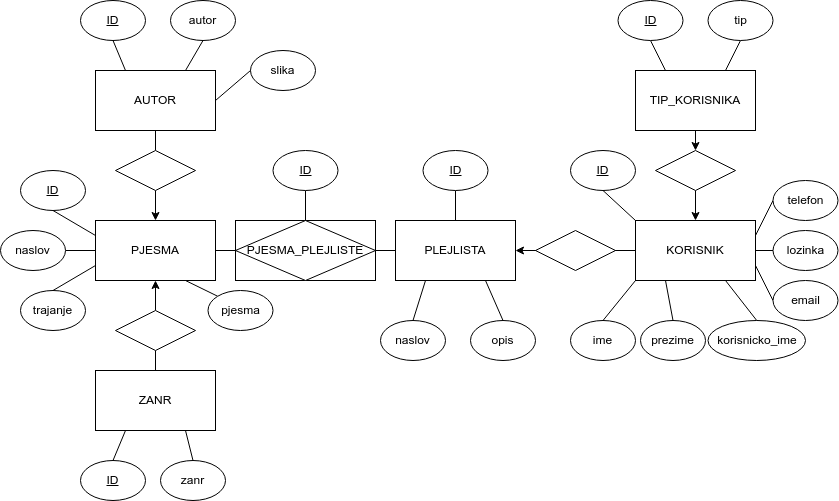
Dijagram sekvence je sličan *SSD*-u i na osnovu njega se i kreira za pojedinačni *use case*. Za razliku od *SSD*-a dijagram sekvence opisuje komunikaciju između korisnika i sistemskih klasa.



Slika br. 7 - Dijagram sekvence (mijenjanje pjesama)

### **2.3.3 Opis baze podataka**

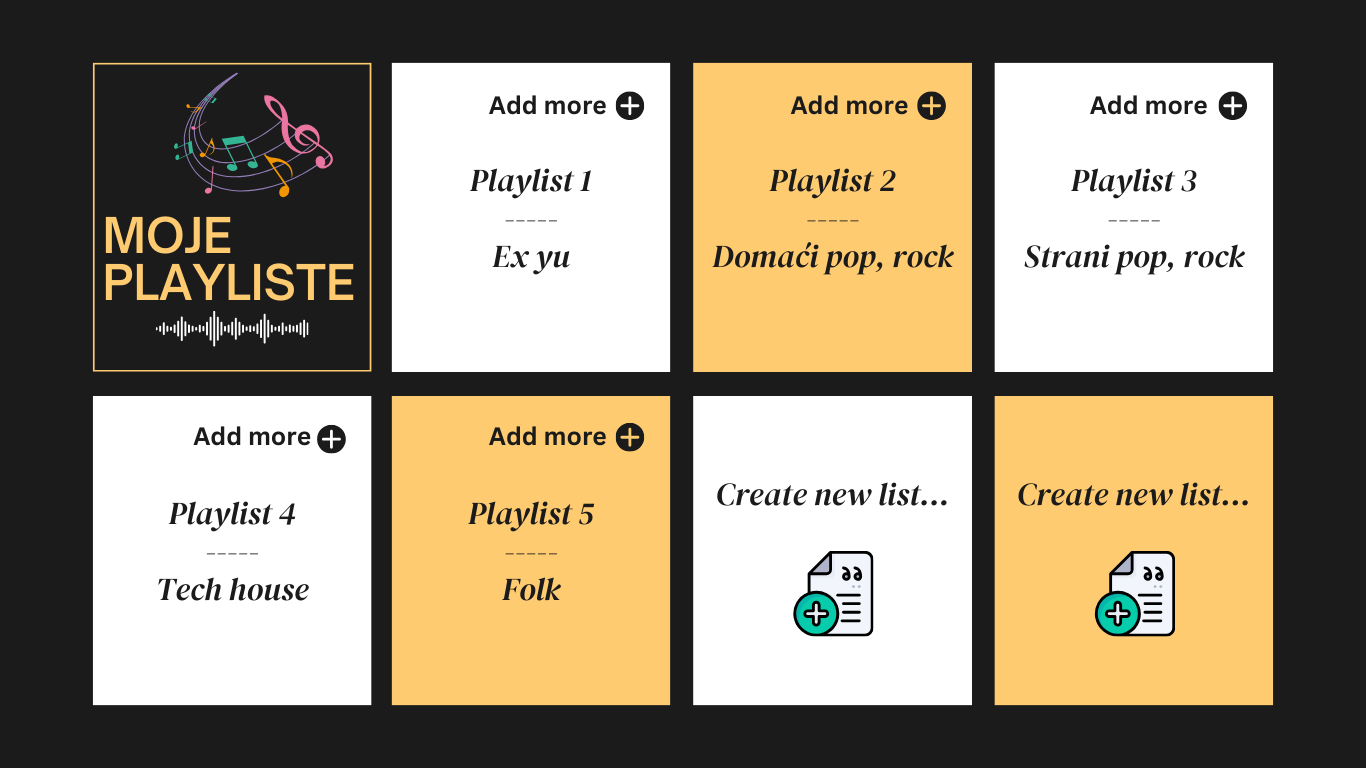
Dijagram kojim se objašnjava struktura baze podataka sa njenim tabelama, vezama i kolonama je *ERD* (*Entity relationship diagram*).



Slika br. 8 - ERD dijagram

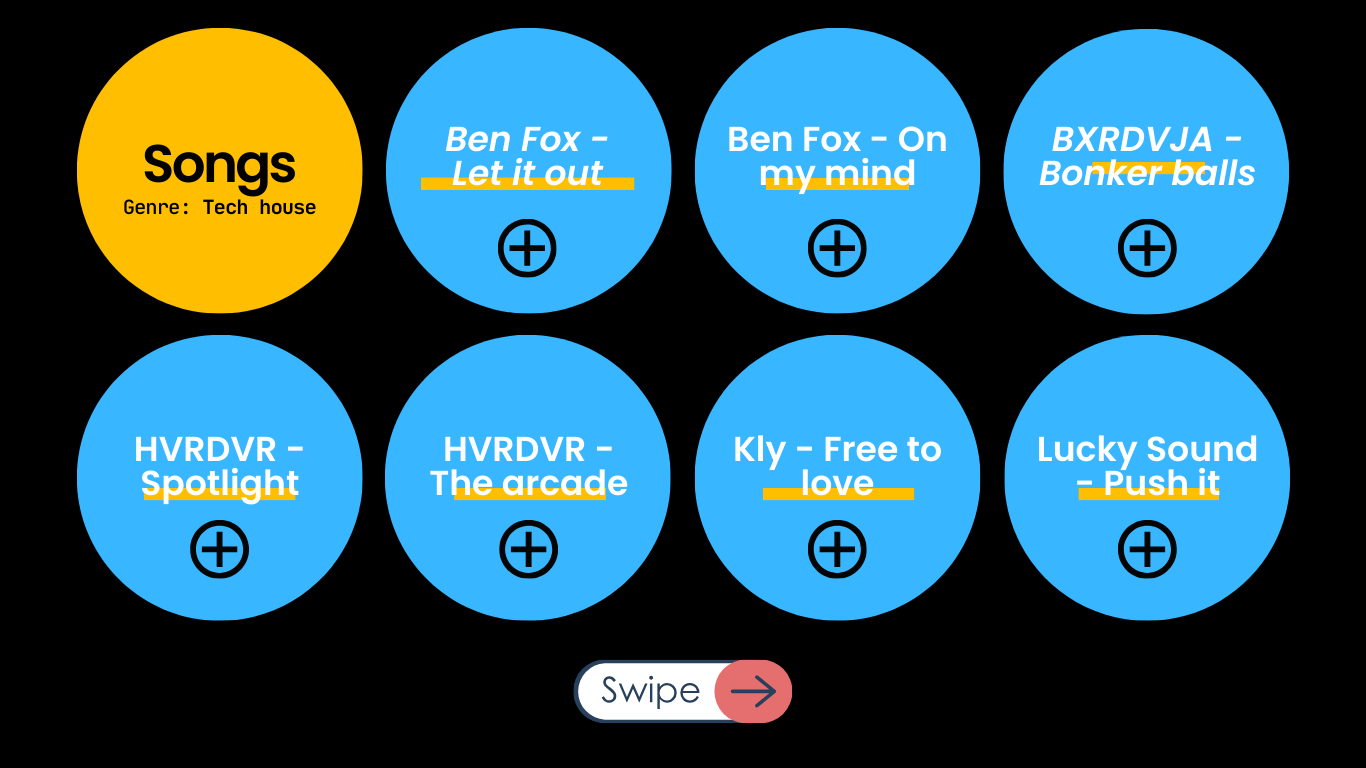
### **2.3.4 Skice interfejsa (*mokcup* ekrani)**

Na skicama interfejsa (*mokcup* ekranima) su prikazane najbitnije funkcionalnosti ovog sistema. Na prvoj skici (*slika br. 9*) se nalaze *playlist*-e koje je korisnik sistema kreirao. Korisnik sistema može dodjeljivati ime ovim listama, dodati naziv žanra kojem pripadaju, dodati novu pjesmu unutar liste pomoću „Add more“ dugmeta i kreirati novu listu. Na ovoj skici je prikazano pet kreiranih *playlist*-i kao i prostor za kreiranje dvije liste koje za sada ne postoje. Ovi *mockup* ekrani služe kao primjer za pomoć pri kreiranju korisničkog interfejsa.



Slika br. 9 - Korisničke playlist-e

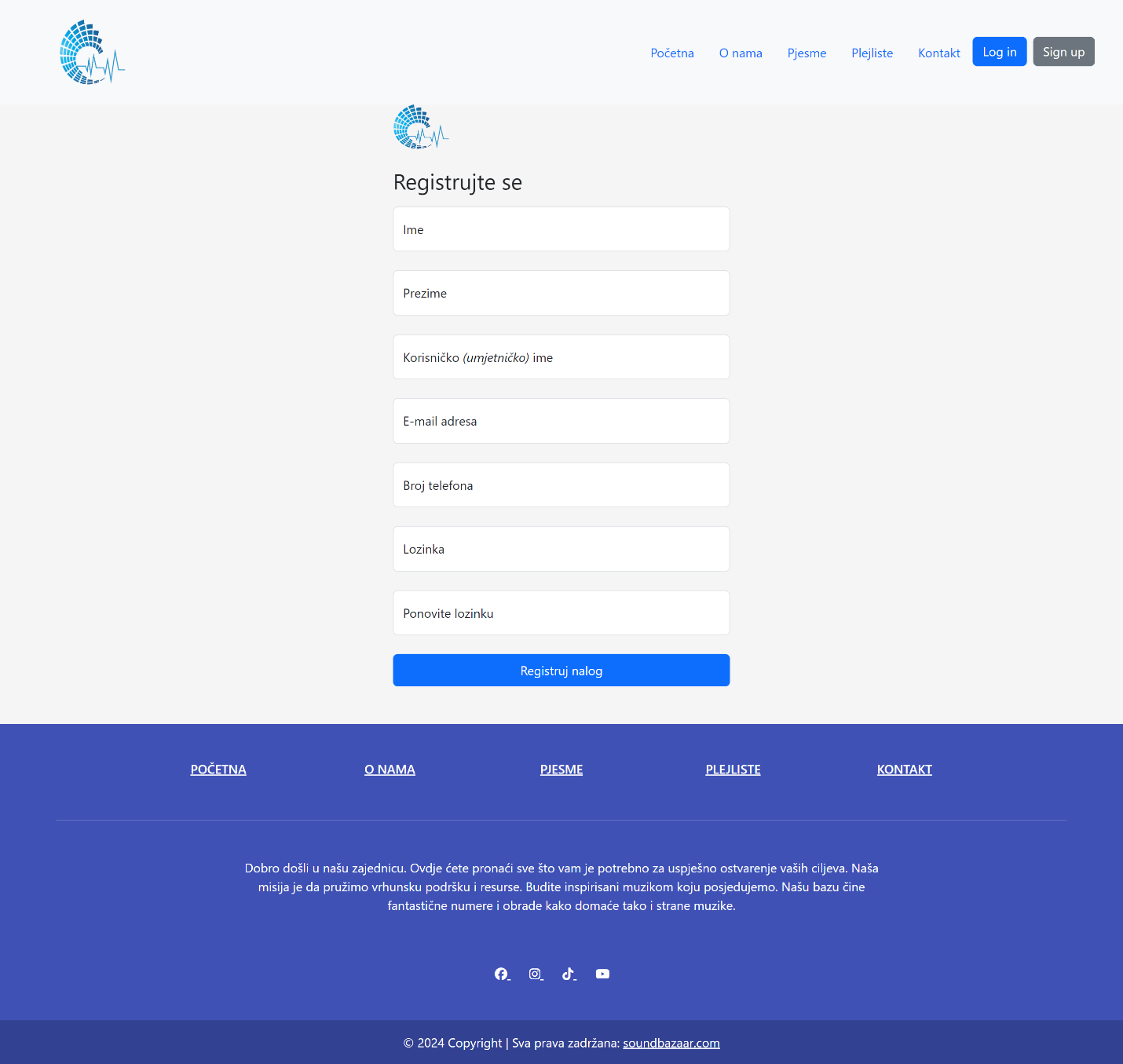
Na sljedećoj skici (*slika br. 10*) je prikazan *mockup* ekran stranice “Songs” koja sadrži pjesme koje su žanra *Tech house*. Na ovoj stranici, klikom na neku pjesmu, korisniku se otvara modalni prozor gdje može preslušati pjesmu ili je pomoću dugmeta ispod naziva dodati u svoju *playlist*-u. Dugme “Swipe” služi za prelazak na drugu stranicu gdje se nalazi još pjesama.



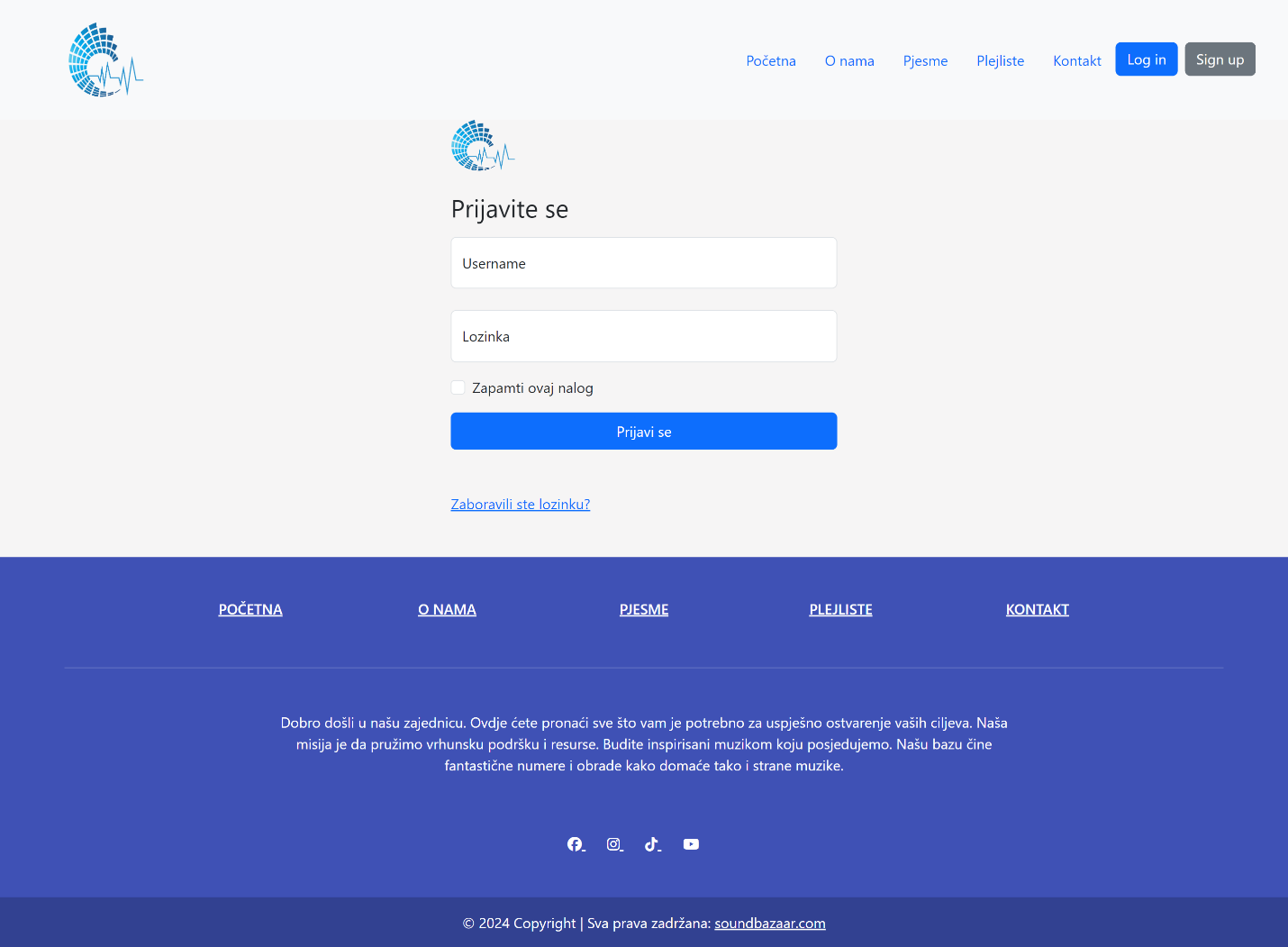
Slika br. 10 - Stranica sa pjesmama

# **3. Funkcionalnosti sistema**

Prvi korak je prijava korisnika na sistemu. Da bi se korisnik prijavio prvo se mora registrovati pomoću forme za registraciju. Nakon što je registrovan svoj nalog (*Slika br. 11*), može se prijaviti na sa tim podacima (*Slika br. 12*).

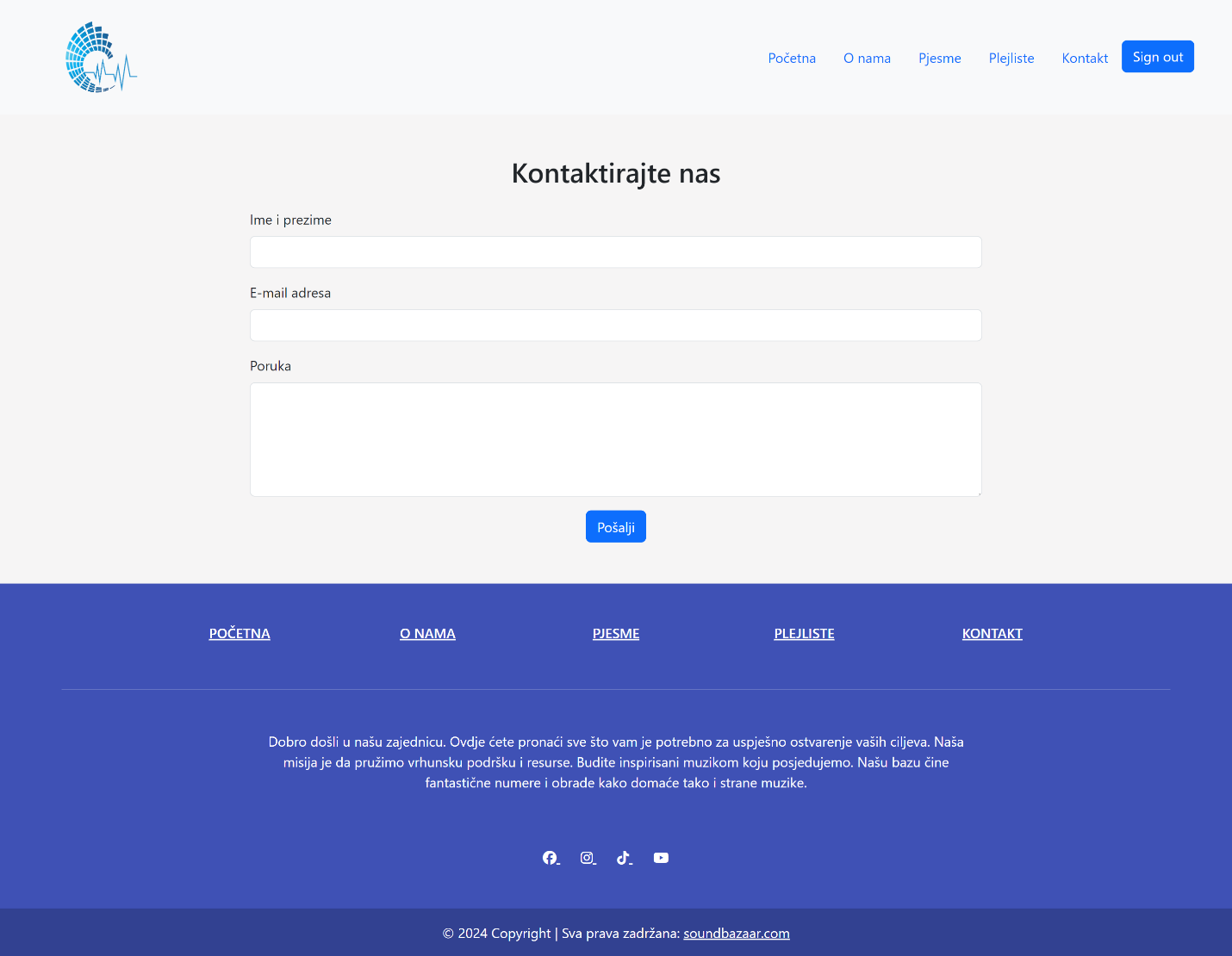


Slika br. 11 – Registracija korisnika



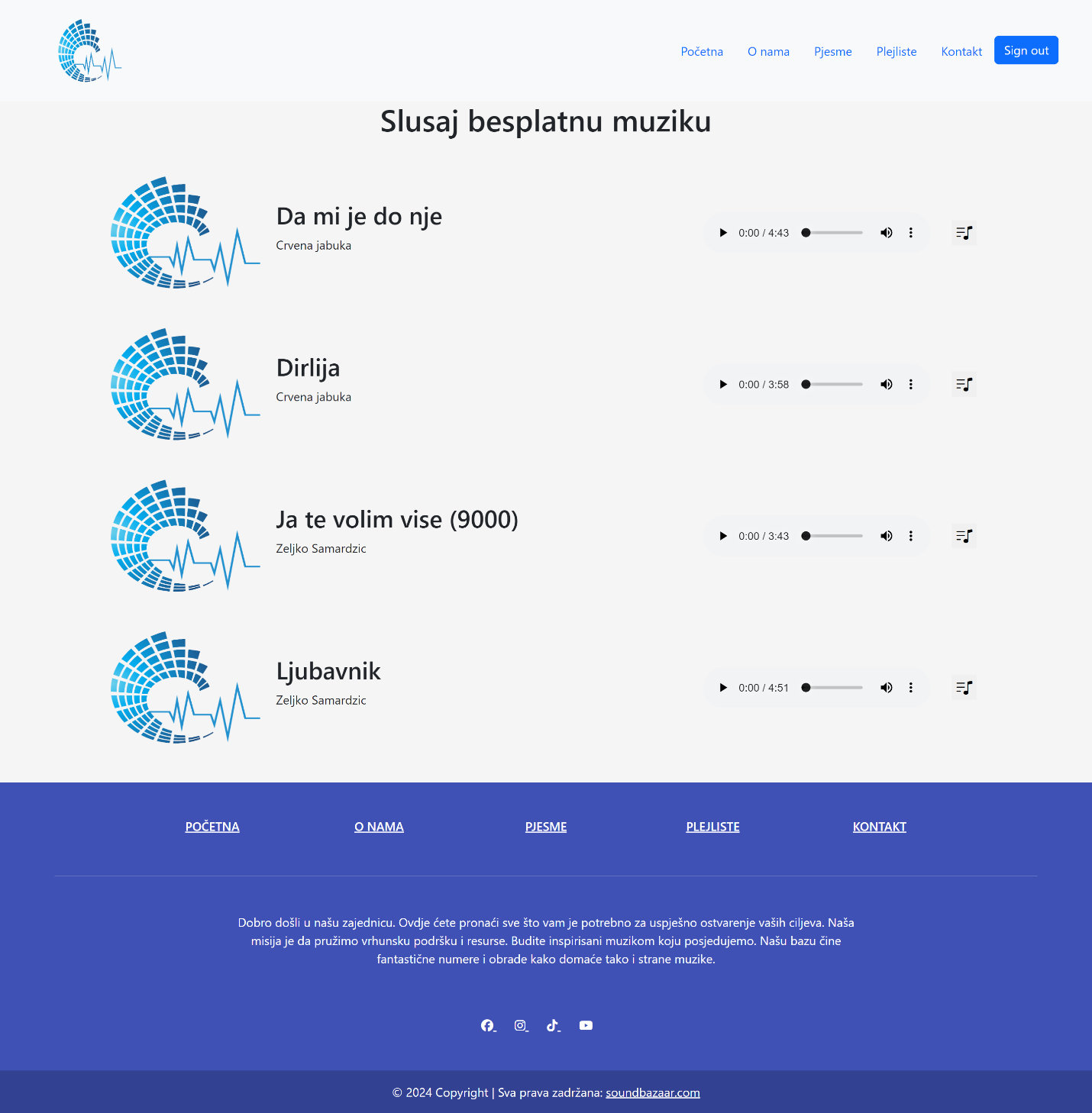
Slika br. 12 – Prijavljivanje za korisnika

Ukoliko korisnik ima dodatnih pitanja vezanih za sistem ili preuzimanje numera, postoji stranica ,,Kontakt” koja posjeduje formu preko koje korisnik može kontaktirati admina ovog sistema (*Slika br. 13*).

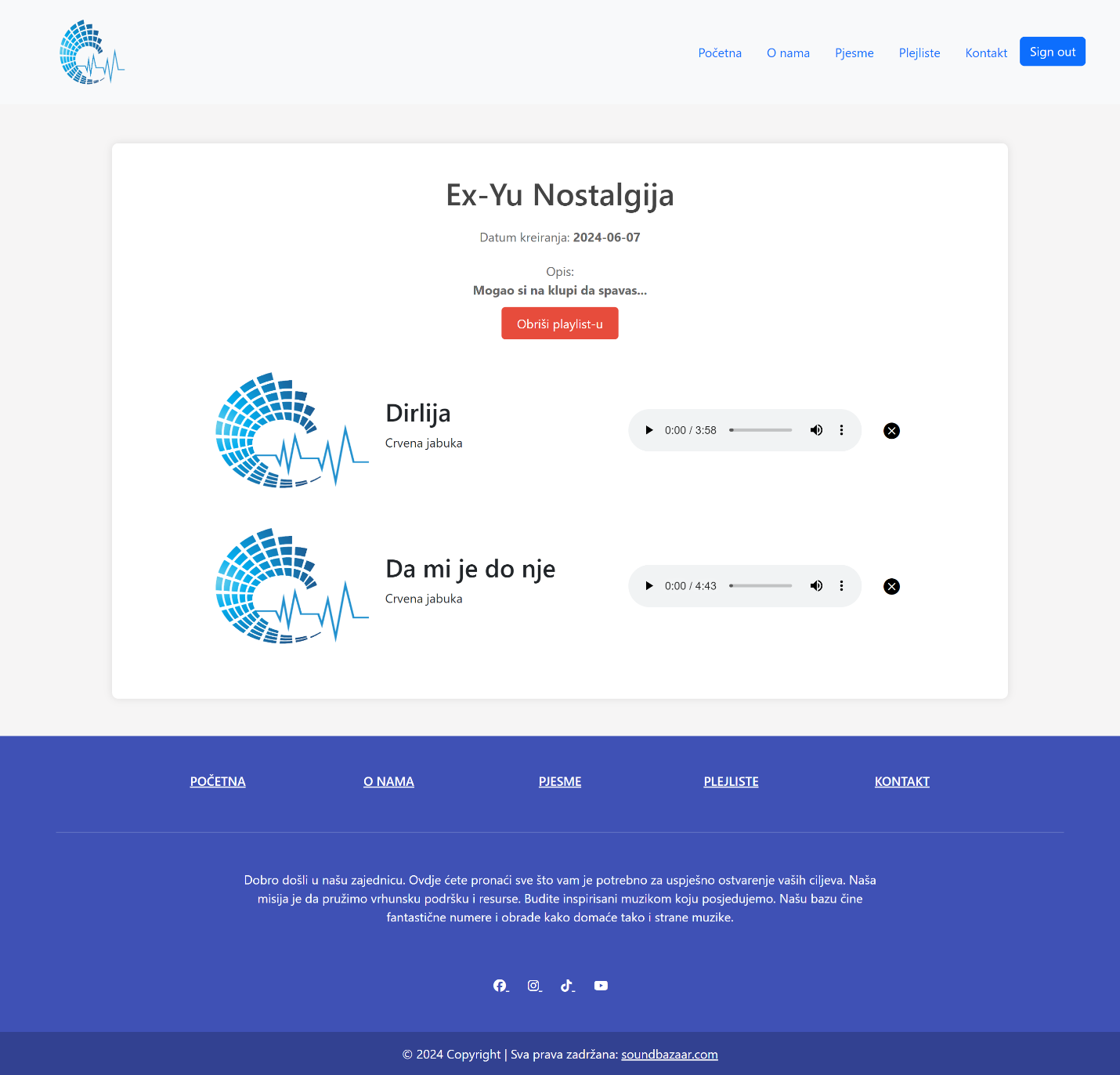


Slika br. 13 – Kontakt forma

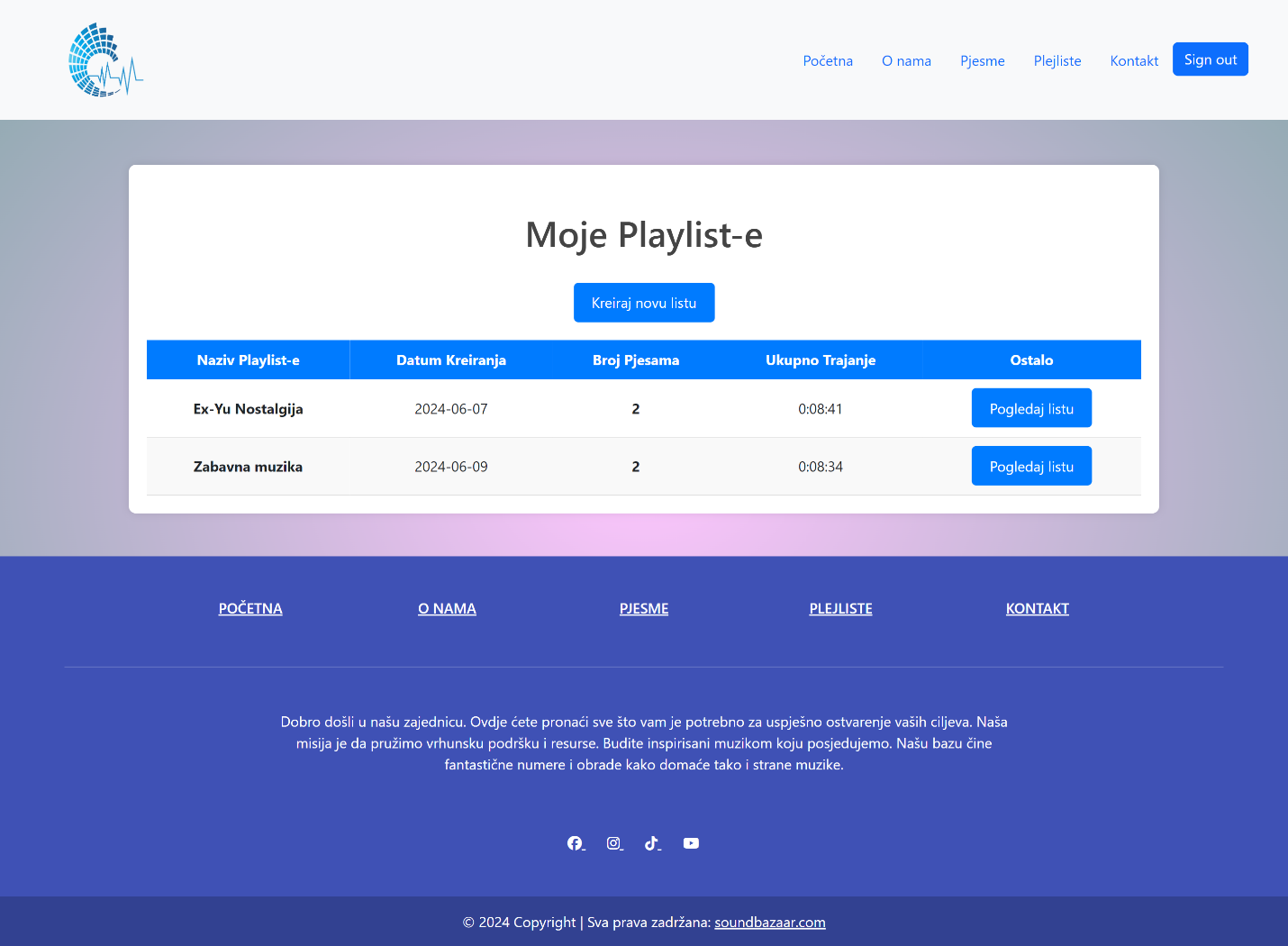
Nakon prijave, korisnik može slušati besplatnu muziku (*Slika br. 14*), kreirati *playlist*-e ili brisati postojeće, brisati pjesme iz *playlist*-e (*Slika br. 15*) ili dodavati nove, imati uvid u sve *playlist*-e (*Slika br. 16*).



Slika br. 14 – Muzika za korisnike

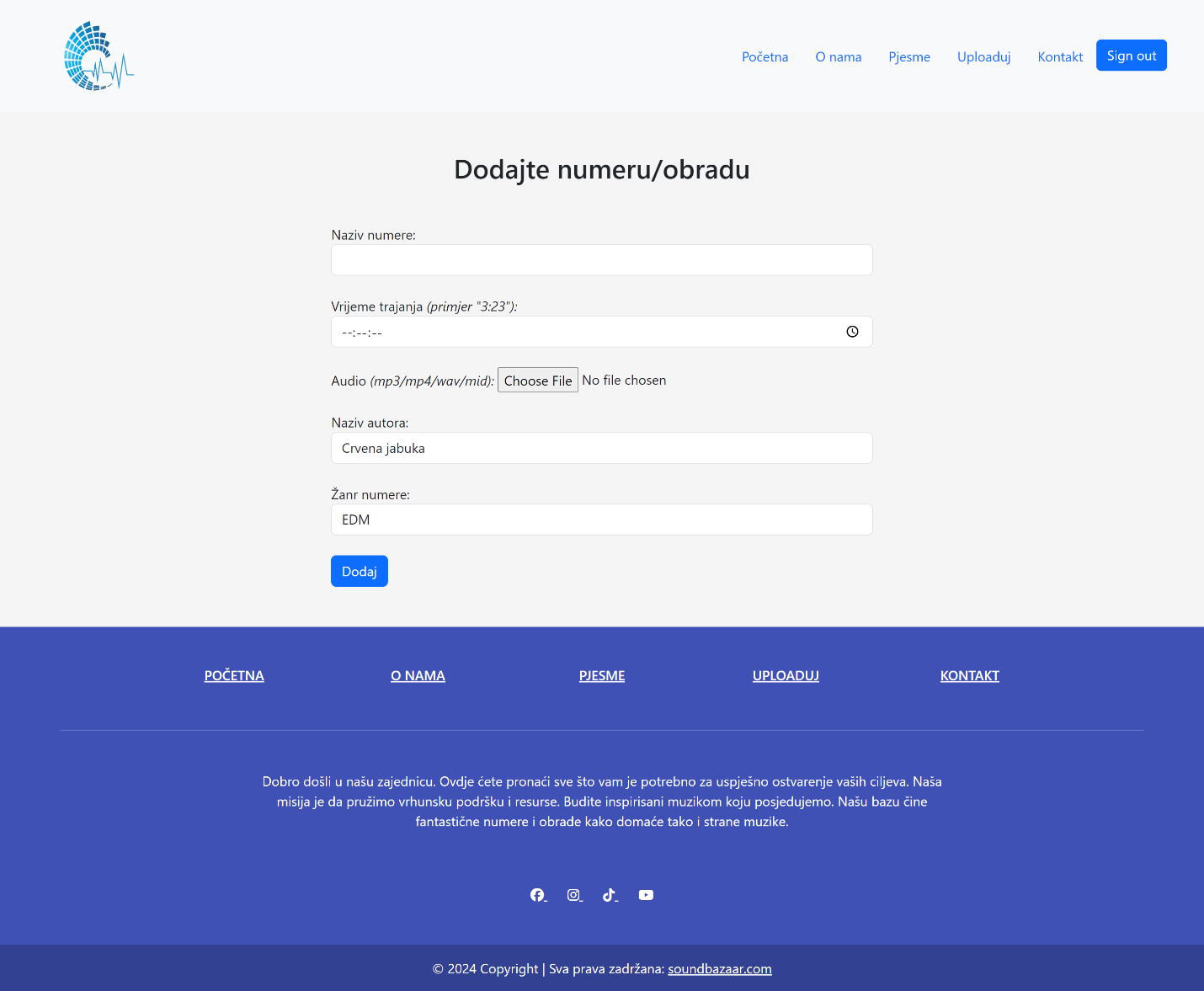


Slika br. 15 – Playlist-a “Ex-Yu Nostalgija”

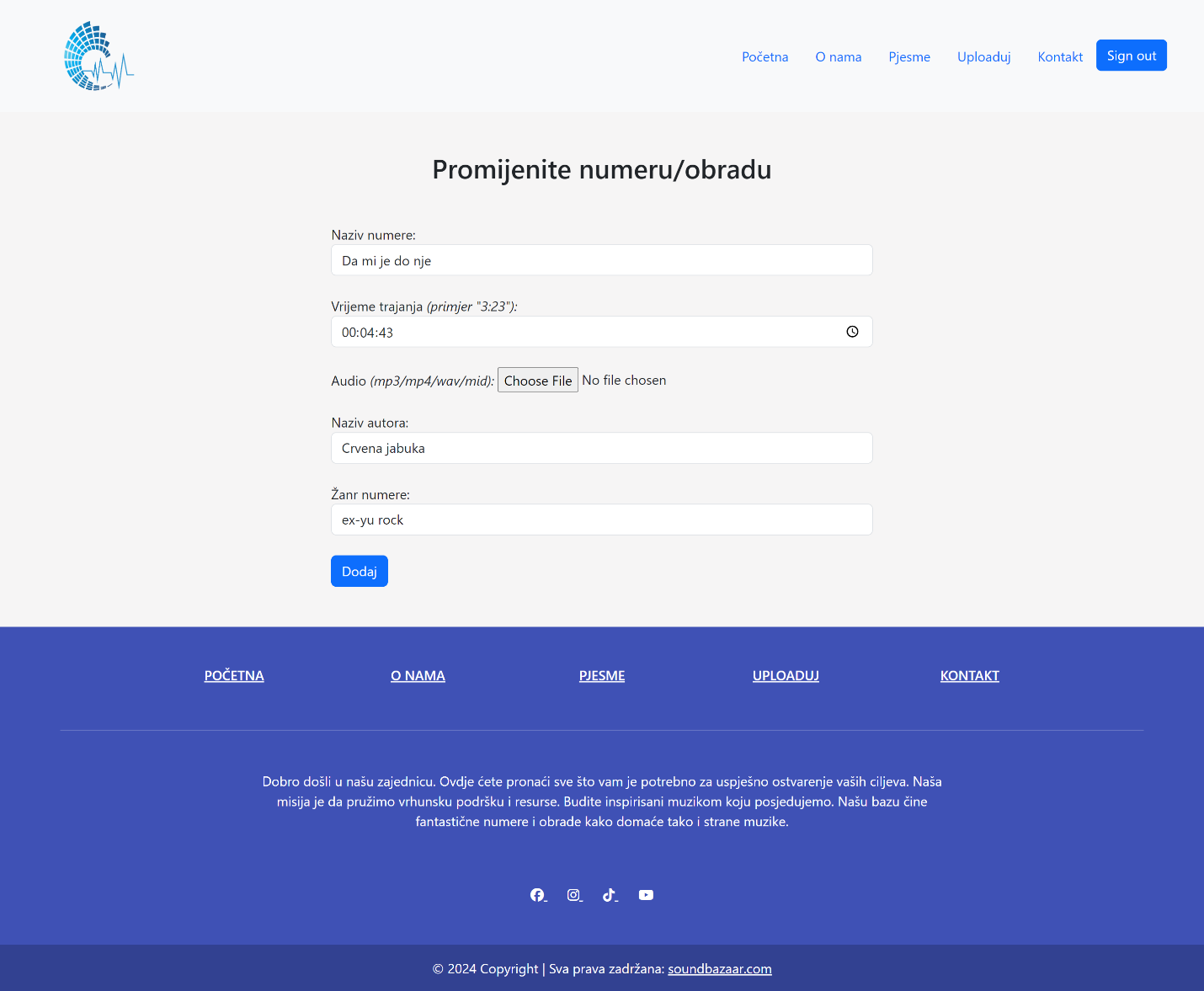


Slika br. 16 – Sve playlist-e

Osim funkcionalnosti za korisnike, postoje funkcionalnosti i za admine ovog sistema. Admin sistema ima mogućnost da odgovara korisnicima na pitanja postavljena putem kontakt forme. Osim toga, admin sistema može upravljati stranicom na kojoj se nalazi muzika. Admin može brisati pjesmu ili urediti njen naslov i opis. Admin sistema ima formu preko koje unosi naziv numere, vrijeme trajanja, audio zapis, naziv autora i žanr kojem pripada ta numera. (*Slika br. 17*) Ukoliko je došlo do greške prilikom unosa podataka, admin u svakom trenutku može ispraviti, odnosno promijeniti informacije o numeri (*Slika br. 18*).

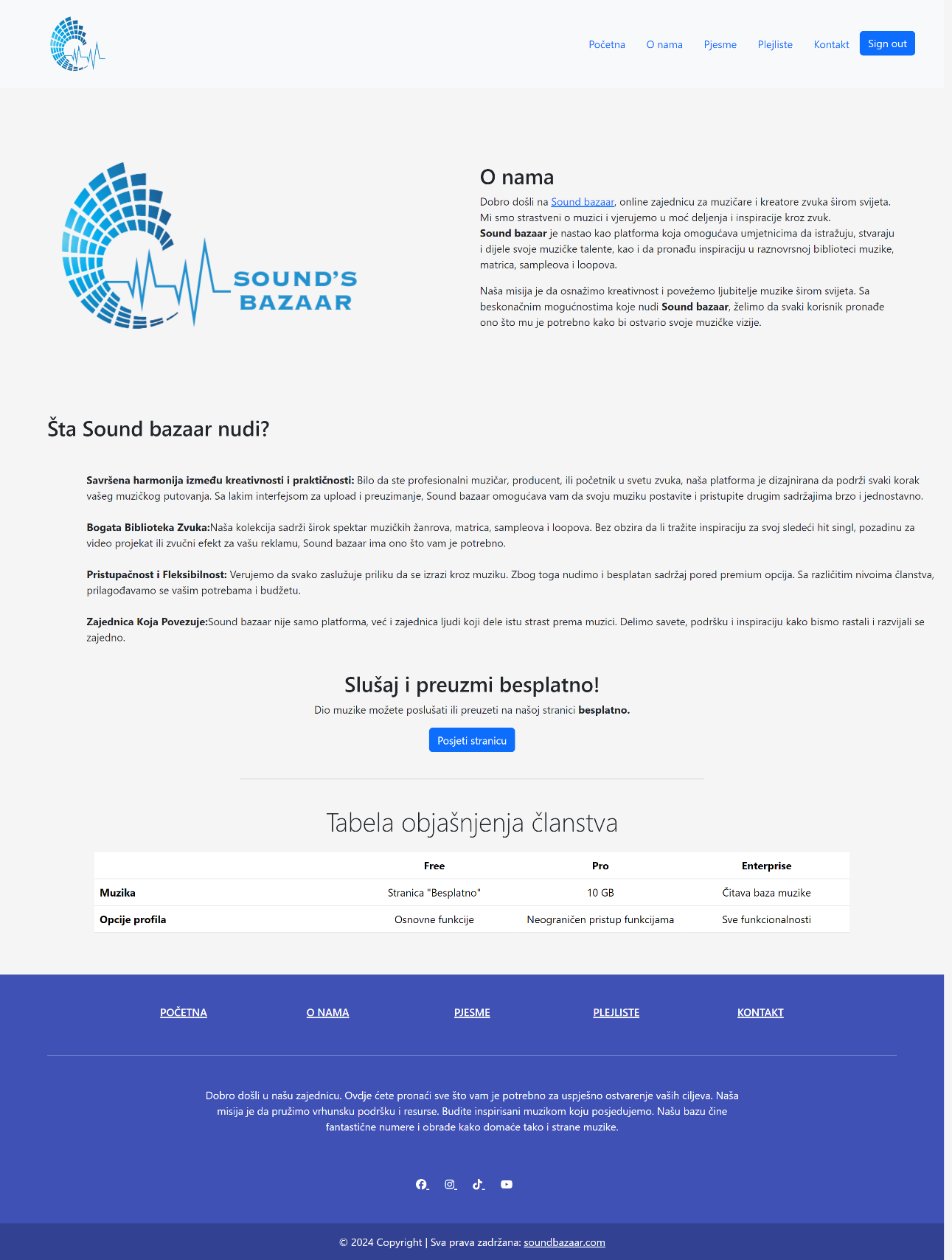


Slika br. 17 – Dodavanje numere/obrade



Slika br. 18 – Promjena informacija o numeri/obradi

Na “O nama” stranici korisnik može saznati više o sistemu i dobiti informacije o onome šta **Sound bazaar** nudi (*Slika br. 19*).



Slika br. 19 – O nama stranica

# **4. Implementacija sistema u Flask-u**

Tehnologija koja je korišćena za implementaciju backend dijela aplikacije je Flask. Flask je moćan paket napravljen za Python programski jezik kojim se kreiraju *fullstack* *web* aplikacije ili *API* (*application programming interface*) servisi. U ovom sistemu Flask služi za implementaciju logike na sajtu. On je zadužen za komunikaciju sa bazom podataka, kreiranje ruta, autentifikaciju, ograničavanje permisija korisniku itd. Zajedno sa paketom Flask-MySQLdb je omogućena konekcija sa MySQL bazom podataka sa kojom korisnici interaguju preko korisničkih ekrana na *frontend-u*. Flask je jako jednostavan i moćan *framework*, obično se koristi za razvoj manjih *web* aplikacija, ali se svakako može koristiti i na većim projektima.

# **5. Zaključak**

Kao zaključak u ovom izvještaju će biti napravljena retrospektiva urađenog tokom trajanja ovog projekta. Projekat je počeo idejom o web stranici koja će biti korišćena za dijeljenje muzike online, tj. aplikacija koja je slična aplikacijama koje su jako popularne (Spotify, Deezer). Daljim razvojem te ideje definisani su ciljevi koje treba ispuniti kako bi se do tog sistema došlo. Osmi ciljeva definisani su i zahtjevi koje sam sistem treba da ispunjava i tim je završena faza planiranja. Nakon faze planiranja dolazi faza analiziranja sistema u kojoj je potrebno definisati šta sistem treba da radi, i to predstaviti kroz tektualne i grafičke modele. U fazi dizajniranja sistema se definiše kako napraviti prethodno opisan sistem, u ovoj fazi se takođe kreiraju modeli koji treba da programerima olakšaju implementaciju traženog sistema. Završna faza ovog projekta je bilo programiranje sistema sa definisanim tehnologijama, ciljevima i na već definisan način. Testiranjem je utvrđeno da je faza programiranja prošla uspješno. Na kraju ovog projekta dobijen je kompletan i funkcionalan sistem sa dokumentovanim koracima implementacije.

## **5.1 Zaduženja članova grupe**

## **5.2 Budući koraci**

Predlog za buduće korake i unapređenje ovog sistema su kreiranje virtuelne zajednice na ovoj web aplikaciji koja će korisnicima služiti kao manja društvena mreža. Na toj zajednici, korisnici bi mogli razmjenjivati iskustva, mišljenja, ideje i ostalo. Osim toga, sistem bi trebalo unaprijediti modernijim (savremenijim) dizajnom prilagodljivim korisnicima.

Trebalo bi korisnicima pružiti mogućnost monetizacije i zarade putem aplikacije od količine dodavanja numera.

# **6. Reference**

1. .
2. .
3. .