Izrada administrativne i informativne aplikacije za fiktivnu udrugu za zaštitu životinja 'Pino' u Vue.js tehnologiji

Katić, Bruno

VELEUČILIŠTE SUVREMENIH INFORMACIJSKIH TEHNOLOGIJA

STRUČNI PRIJEDIPLOMSKI STUDIJ INFORMACIJSKIH TEHNOLOGIJA

Bruno Katić

ZAVRŠNI RAD

IZRADA ADMINISTRATIVNE I INFORMATIVNE APLIKACIJE ZA FIKTIVNU UDRUGU ZA ZAŠTITU ŽIVOTINJA 'PINO' U VUE.JS TEHNOLOGIJI.

Zagreb, listopada 2024.

Veleučilište suvremenih informacijskih tehnologija
10000 Zagreb, Ulica Vjekoslava Klaića 7

Studij:	Stručni prijediplomski studij informacijskih tehnologija smjer programiranje
Student:	**Bruno Katić**
Matični broj:	2020019

Zadatak završnog rada

Predmet:	Oblikovanje Web stranica
Naslov:	**Izrada administrativne i informativne aplikacije za fiktivnu udrugu za zaštitu životinja 'Pino' u Vue.js tehnologiji.**
Zadatak:	Potrebno je ukratko opisati tehnologije za izradu mrežnih stranica, objasniti princip rada Vue.js tehnologije. Kroz praktični dio rada izraditi aplikaciju za administraciju poslovanja tvrtke.
Mentor:	Jurica Đurić, v. pred.
Zadatak uručen kandidatu:	10.9.2024.
Rok za predaju rada:	18.10.2024.
Rad predan:	_____

Povjerenstvo:

Marijan Čančarević, v. pred.	član predsjednik	_____
Jurica Đurić, v. pred.	mentor	_____
Dragana Čulina, pred.	član	_____

SADRŽAJ

POPIS SLIKA

POPIS KODOVA

1. UVOD

U današnje vrijeme, digitalizacija igra ključnu ulogu u svakodnevnom funkcioniranju organizacija različitih vrsta, pa tako i onih koje se bave zaštitom životinja. Azili te razne udruge za životinje svakodnevno se suočavaju s nizom izazova, poput učinkovitog upravljanja sa podacima o životinjama i informiranjem javnosti o svojim aktivnostima te svom radu. Ovi izazovi često predstavljaju prepreku u njihovom svakodnevnom radu, otežavajući ostvarenje ciljeva, stoga je važno razviti tehnološka rješenja koja omogućuju jednostavniji i pregledniji rad tih organizacija te efikasniju i transparentniju komunikaciju s javnošću. Cilj ovog završnog rada je prikazati implementaciju i razvoj dviju povezanih mrežnih aplikacija za fiktivnu udrugu za zaštitu životinja zvanu 'Pino', koje će omogućiti jednostavno upravljanje podacima i dati korisnicima jasan uvid u rad udruge te informacije o životinjama za udomljavanje, udomljenim životinjama te životinjama koje su uginule u udruzi.

Obje aplikacije razvijene su korištenjem Vue.js, modernog i popularnog frontend JavaScript framework-a, koji omogućava izradu responzivnih i interkativnih korisničkih sučelja te ubrzava proces razvoja mrežnih aplikacija. Prva aplikacija namjenjena je administraciji udruge. Omogućava unos, ažuriranje i brisanje podataka vezanih za pse i mačke, kao i novosti i informacije vezane za rad udruge. Ova aplikacija služi kao centralno mjesto za upravljanje svim podacima i aktivnostima unutar organizacije. Korisničko sučelje je vrlo jednostavno, s naglaskom na lakoću uporabe. Dizajn nije bio glavni prioritet, već same funkcionalnosti. Druga aplikacija je namjenjena korisnicima, pruža pregled aktualnih vijesti udruge te njezin rad. Omogućava uvid u pse i mačke koje su trenutno štičenici udruge i za udomljavanje su, ali i one koji su udomljeni i uginuli. Ključni cilj aplikacije je potaknuti na udomljavanje i promoviranje pasa i mačaka kako bi im se povećala vidljivost i s time pružila prilika za novi, trajni dom.

Posebnu vrijednost ovome projektu daje suradnja s predsjednicom Hrvatske udruge zaštitnika životinja 'Noina Arka', gospođom Helenom Fink, koja je ljubazno dozvolila korištenje stvarnih podataka iz rada udruge (slike, imena i slike životinja, vijesti..). Takva suradnja pridonijela je vjerodostojnosti aplikacija i omogućila autentično prikazivanje stvarnog rada udruge. Autor ovog rada također je dugogodišnji volonter u istoj udruzi, što je omogućilo dublji uvid i razumijevanje u rad udruge i usklađivanje tehnoloških rješenja s njezinim ciljevima.

U radu su najprije opisane tehnologije za izradu mrežnih aplikacija, uključujući analizu frontend tehnologija, backend tehnologija te alata za verzioniranje i hosting. Također, obrađena je struktura baze podataka i način komunikacije između frontend-a i backend-a te kako se podaci pohranjuju, dohvaćaju i obrađuju. Posebna pažnja posvećena je korisničkom sučelju i iskustvu, kao i izazovima koji su se pojavili tijekom razvoja obje aplikacije. Na kraju su

analizirani rezultati razvijenih aplikacija te predložene mogućnosti za njihovo buduće unapređenje i razvoj, s ciljem daljnje optimizacije rada udruge i promocije udomljavanja životinja.

2. TEHNOLOGIJE ZA IZRADU MREŽNIH APLIKACIJA

Izrada mrežnih aplikacija je složen proces koji uključuje kombinaciju različitih tehnologija, alata i metoda. Obuhvaća razvoj korisničkog sučelja (frontend), serverske strane (backend) te način na koji ti dijelovi međusobno komuniciraju. Za postizanje optimalnih performansi, sigurnosti, skalabilnosti i intuitivnog korisničkog sučelja ključno je korištenje odgovarajućih tehnologija. U daljnjem tekstu obrađuju se ključne tehnologije koje se primjenjuju u izradi mrežnih aplikacija.

2.1. Frontend tehnologije

Frontend tehnologije odnose se na sve alate i programe koji se koriste za izradu dijela aplikacije s kojim korisnici izravno komuniciraju. To su tehnologije koje omogućuju prikaz sadržaja te interakcije unutar mrežnog preglednika. Osnovne tehnologije za razvoj frontend dijela aplikacije su HTML (engl. HyperText Markup Language), CSS (Cascading Style Sheets) i JavaScript.

- *HTML:* Osnovni jezik za stukturiranje sadržaja na mrežnim stranicama. Definira osnovne elemente kao što su tekst, slike, veze i forme te osigurava hijerarhijsku organizaciju podataka na stranici.
- *CSS:* Stilski jezik koji omogućava stilizaciju HTML elemenata, upravljanje bojama, fontovima, rasporedom elemenata na stranici i prilagođavanje izgleda aplikacije različitim uređajima.
- *JavaScript:* Programski jezik za razvoj mrežnih stranica i mrežnih aplikacija. Omogućuje dodavanje dinamičkih funkcionalnosti aplikaciji. Može računati, manipulirati podacima te provjeravati podatke.

Pored ovih osnovnih tehnologija, moderne mrežne aplikacije često se oslanjaju na JavaScript framework-e i biblioteke koje omogućuju razvoj složenijih i bržih sučelja. Među najpopularnijim framework-ovima su:

- *Vue.js:* JavaScript framework poznat po svojoj fleksibilnosti i reaktivnosti, razvijen od strane Evan You-a, bivšeg zaposlenika Google-a, koji je temeljem svojih iskustava s različitim frontend frameworkovima, poput AngularJS-a, odlučio stvoriti vlastiti framework koji bi bio jednostavniji za korištenje od postojećih i istovremeno dovoljno moćan da je konkurentan sa istima. Vue.js je prvi put predstavljen javnosti u veljači 2014. godine, od tada do danas je doživio veliki rast popularnosti zbog svoje jednostavnosti i lakoće učenja [1], [2].

- *React:* JavaScript biblioteka razvijena od strane Facebook-a, prvi put predstavljen 2013. godine i od tada je postala jedna od najpopularnijih tehnologija za izradu korisničkih sučelja. Glavna karakteristika React-a je njegov pristup izradi aplikacija koristeći komponente, male, ponovno iskoristive dijelove koda koji pojednostavljuju razvoj složenih sučelja. Ima veliki ekosustav dodataka i podrške te omogućuje lako integriranje s drugim tehnologijama i framework-ovima [3].
- *Angular:* JavaScript framework kojeg je razvila tvrtka Google. Prvi put je predstavljen 2010. godine kao AngularJS, a 2016. godine je u potpunosti preoblikovan u Angular 2, čime je započela nova era razvoja mrežnih aplikacija. Angular je full-stack framework, što znači da nudi sveobuhvatna rješenja za izradu aplikacija uključujući integraciju s backend-om [4].

2.2. Backend tehnologije

Backend tehnologije pokrivaju dio aplikacije koji korisnici ne vide, a osigurava funkcioniranje sustava, obradu podataka i komunikaciju između klijentske strane i baze podataka. Serverska strana aplikacije je odgovorna za primanje zahtjeva od korisnika, obradu tih zahtjeva te slanje odgovarajućih podataka natrag na frontend.

Backend aplikacije mogu biti razvijene koristeći različite programske jezike i framework-e. Neki od najkorištenijih backend tehnologija uključuju:

- *Node.js:* JavaScript runtime okolina koja omogućuje pokretanje JavaScript koda na serveru. Prvi put je predstavljena 2009. godine od strane Ryana Dahla. Koristi arhitekturu vođenu događajima što ga čini pogodnim za izradu aplikacija koje zahtijevaju brze, asinkrone operacije kao što su aplikacije u stvarnom vremenu [5].
- *Python:* Programski jezik poznat po svojoj jednostavnosti i čitljivosti. Prvi put je objavljen 1991. godine od strane Guido van Rossum-a. Često se koristi za razvoj backend aplikacija, posebno u kombinaciji sa framework-ovima kao što su Django ili Flask [6].
- *PHP:* Programski jezik koji je prvi put predstavljen 1994. godine, a razvijen je od strane Rasmusa Lerdorf-a. Iako je stariji od mnogih modernih tehnologija, PHP je i dalje široko korišten za izradu dinamičnih mrežnih stranica i aplikacija, pogotovo u kombinaciji s bazama podataka poput MySQL-a. Popularan je zbog svoje jednostavnosti i velike zajednice [7].

Osim programskih jezika, za backend je važna i baza podataka, koja omogućava pohranu

i dohvat podataka. Baze podataka mogu biti relacijske (npr. MySQL, PostgreSQL) ili NoSql (npr. MongoDB), ovisno o zahtjevima aplikacije. Relacijske baze koriste se za pohranu podataka u strukturiranom formatu s tablicama i odnosima između njih, dok NoSQL baze pružaju veću fleksibilnost kod nestrukturiranih ili neuređenih podataka.

Za komunikaciju između frontend i backend sustava često se koriste API-ji. RESTful API i GraphQL najčešće su metode za razmjenu podataka između klijentske i serverske strane. REST koristi standardne HTTP (engl. Hyper Text Transfer Protocol) metode za prijenos podataka, dok GraphQL omogućava klijentima da specificiraju točne podatke koji su im potrebni.

2.3. Alati za verzioniranje i hosting

Osim samog razvoja, neophodno je koristiti alate za verzioniranje koda kako bi se osigurao nesmetan razvoj i implementacija aplikacija. Git je najpopularniji sustav za verzioniranje koji omogućuje praćenje promjena u kodu, suradnju među timovima i povratak na prethodne verzije koda u slučaju potrebe. Platforme poput GitHub i GitLab omogućuju hostanje repozitorija, kolaboraciju i upravljanje projektima.

Kada je aplikacija spremna za produkcijsko okruženje, potrebno je odabrati odgovarajuču hosting platformu. AWS (engl. Amazon Web Services), Google Cloud Platform, Vercel i Firebase nude različite opcije za hostanje mrežnih aplikacija, uključujući baze podataka i servere. Ovi alati omogućuju upravljanje resursima, skalabilnost aplikacija i osiguravaju da aplikacija bude dostupna korisnicima u svakom trenutku.

3. TEHNIČKA IZVEDBA APLIKACIJA

U ovom poglavlju detaljno se analiziraju ključni izazovi s kojima se suočavaju azili i udruge za zaštitu životinja te se predstavljaju rješenja kroz projekt.

3.1. Problemi u vođenju azila i udruga za životinje

Vođenje ovakvih organizacija često uključuje probleme poput upravljanja složenim podacima o životinjama, uključujući informacije o njihovom zdravstvenom stanju, lokaciji pronalaska te trenutnom statusu udomljavanja. To uključuje i vođenje evidencije o novim životinjama koje dolaze u azil ili udrugu te njihovoj povijesti udomljavanja ili vraćanja.

Još jedan od izazova je komunikacija s javnošću. Potrebno je redovito informirati javnost o aktualnim vijestima, događajima, potrebama, kao i životinje koje su dostupne za udomljavanje.

Povjerenje i transparentnost samog rada udruge ili azila je također jedan od izazova. Javnost očekuje transparentnost u vezi s radom, o tome kako se donacije koriste i kako se postupa s životinjama. Stoga je važno da udruge i azili jasno i otvoreno komuniciraju s javnošću te pružaju uvid u svoje aktivnosti.

3.2. Potrebe i zahtjevi korisnika

Prva skupina korisnika su administratori, vodeće osobe udruga i azila, čija je glavna potreba imati jednostavan i učinkovit način za upravljanje podacima. Trebaju platformu koja omogućuje jednostavno dodavanje, ažuriranje i brisanje životinja i informacija o njima poput zdravstvenog stanja, veterinarskih obrada i trenutnog statusa. Osim toga, platforma mora olakšati i objavljivanje događaja, vijesti i ostalih aktivnosti udruge ili azila.

Druga skupina su javni korisnici kojima je potrebno intuitivno korisničko sučelje za jednostavan pregled životinja dostupnih za udomljavanje te lako dostupan uvid u aktualne vijesti i sam rad udruge ili azila. Potencijalni udomitelji žele imati mogućnost detaljnog pregleda svakog ljubimca, uključujući fotografije te podatke o starosti i zdravlju kako bi lakše mogli donjeti odluku o udomljavanju.

3.3. Rješenja kroz funkcionalnosti aplikacija

Razvijene su dvije aplikacije za fiktivnu udrugu za zaštitu životinja 'Pino'. Svaka od aplikacija ima funkcionanosti dizajnirane za rješavanje konkretnih potreba i pružanja usluge.

3.3.1. Admin aplikacija

Za upravljanje podacima unutar udruge razvijena je admin aplikacija. Funkcionalnosti su sljedeće:

- *Kreiranje, ažuriranje i brisanje životinja*: Administrator može dodati nove životinje u bazu podataka, zajedno sa svim potrebitim informacijama poput imena, starosti, spola,

pasmine, mikročipa, veterinarske obrade, slika i kategorije u koju pripada. Također, dodanim životinjama se može ažurirati bilo koje od navedenih informacija te se dodane životinje i mogu obrisati.

- *Kreiranje, ažuriranje i brisanje vijesti:* Administrator može dodati nove vijesti u bazu podataka, zajedno sa svim potrebitim informacijama poput naslova, uređenog teksta sa naglašenim dijelovima i linkovima te slika. Također, dodane vijesti se mogu ažurirati i obrisati.

3.3.2. Korisnička aplikacija

Za krajnje korisnike koji traže informacije o životinjama i o radu udruge razvijena je korisnička aplikacija. Funkcionalnosti su sljedeće:

- *Pregled životinja dostupnih za udomljavanje:* Korisnici mogu pregledati životinje koje se trenutno nalaze u udruzi i dostupne su za udomljavanje te mogu filtrirat životinje po starosti ili invaliditetu. Osim toga, mogu vidjeti i uginule i udomljene životinje.
- *Pregled vijesti i aktivnosti:* Korisnici mogu pratiti vijesti i događaje udruge.
- *Interakcija sa udrugom:* Korisnici mogu ispuniti kontakt obrazac za dodatnja pitanja, informacije, prijave interesa za udomljavanje određene životinje ili druge oblike suradnje sa udrugom.

4. PRAKTIČNI RAD – "PINO - ADMIN" I "PINO – KORISNIČKA APLIKACIJA"

U ovom poglavlju detaljno se opisuje proces implementacije dviju povezanih mrežnih aplikacija za fiktivnu udrugu za zaštitu životinja 'Pino'.

4.1. Postavljanje razvojnog okruženja

Obje aplikacije kreirane su na sljedeći način:

- *Instalacija Node.js-a i npm-a (engl. Node Package Manager):* Prvi korak je instalacija Node.js-a sa službene stranice, okruženja koje omogućuje izvršavanje JavaScript koda izvan mrežnog preglednika. Potreban je za rad Vue CLI-a i drugih razvojnih alata. Njegova instalacija uključuje i npm, paket menadžer za instalaciju i upravljanje JavaScript paketima.
- *Instalacija Vue CLI*-a: Nakon instalacije Node.js-a potrebno je instalirati Vue CLI globalno putem npm-a sa komandom *npm install -g @vue/cli* [9]
- *Kreiranje Vue aplikacije:* Nakon instalacije Vue CLI-a potrebno je kreirati projekt sa komandom *vue create ime-projekta*. Prilikom kreiranja projekta, odabrane su opcije poput Babel, ESLint i Vue Router.

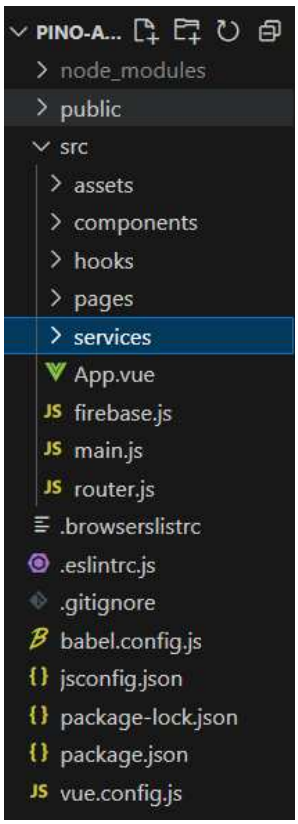
4.2. Arhitektura aplikacija

Struktura obje aplikacije, administrativne i korisničke temelji se na sljedećim datotekama i direktorijima [Slika 1.], [Slika 2.]:

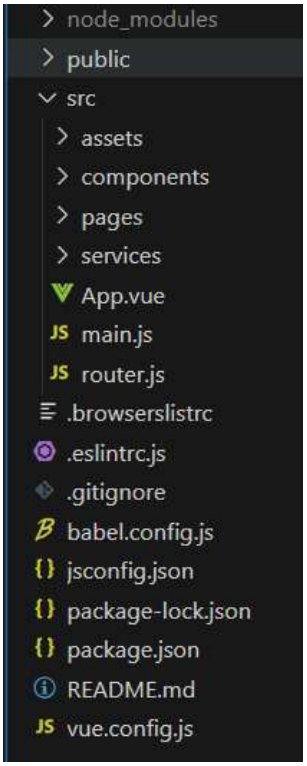
- *Root direktorij:* Korijenski direktorij u kojem se nalaze konfiguracijske datoteke i dva glavna direktorija:
    - *package.json:* Sadrži osnovne informacije o projektu poput njegovih ovisnosti i skripti za gradnju.
    - *vue.config.js:* Sadrži konfiguracije za Vue CLI, uključujući postavke za razvojni server, putanju i druge opcije koje su specifične za taj projekt.
    - *babel.config.js:* Konfiguracijska datoteka koja definira postavke za Babel, poput popisa pluginova i preseta za transformaciju koda.
    - *jsconfig.json*: Konfiguracijska datoteka za JavaScript koja definira osnovne postavke projekta, poput putanja do mapa i automatsko dovršavanje koda.
    - *node_modules:* Direktorij koji sadrži sve instalirane biblioteke i ovisnosti potrebne za rad aplikacije. Ovaj direktorij se automatski generira prilikom

instalacije paketa putem NPM-a.

- *public:* Direktorij koji sadrži statičke resurse koji su dostupni u konačnoj izgradnji aplikacije:
    - *index.html:* Glavna HTML datoteka. U njoj se nalaze osnovni HTML elementi i referenciranje glavnih JavaScript i CSS datoteka.
    - *Slike i ikone*
- *src:* Direktorij u kojem se nalaze svi izvorni kodovi aplikacije, uključujući:
    - *assets:* Direktorij za statičke datoteke poput slika i fontova
    - *components:* Direktorij za komponente koje čine korisničko sučelje aplikacije. Svaka komponenta predstavlja dio (komad) sučelja.
    - *pages:* Direktorij koji sadrži glavne stranice aplikacije. Svaka datoteka u ovom direktoriju odgovara jednoj stranici aplikacije.
    - *services:* Direktorij za upravljanje API pozivima.
    - *hooks (samo admin):* Direktorij koji sadrži datoteke za prilagođene funkcije (hooks) koje se koriste unutar Vue komponenti za ponovnu upotrebu logike.
    - *App.vue:* Glavna Vue komponenta koja postavlja osnovnu strukturu sučelja i služi kao korijen aplikacije.
    - *main.js:* Datoteka koja pokreće Vue instancu, uključujući druge pluginove i inicijalnu konfiguraciju.
    - *router.js:* Datoteka za Vue Router koja definira sve rute aplikacije i komponente vezane za iste.
    - *firebase.js (samo admin):* Datoteka koja sadrži konfiguraciju Firebase-a, omogućuje integraciju s Firebase uslugama kao što je Storage.



Slika 1. Arhitektura pino-admin aplikacije



Slika 2. Arhitektura pino-korisničke aplikacije

4.3. Osnovna struktura Vue komponente

Vue komponente su osnovni gradivni elementi svake Vue.js aplikacije. Svaka komponenta se sastoji od tri osnovna dijela [Kôd 1.]:

- *Template:* Ovaj dio komponente sadrži HTML kod. Unutar njega koriste se Vue-ove direktive i sintaksa za dinamičko povezivanje podataka te upravljanje prikaza sadržaja temeljem različitih uvjeta.
- *Script:* Dio komponente koji sadrži JavaScript kod koji upravlja funkcionalnošću komponente. Unutar njega definiraju se razne logičke operacije poput funkcija i reaktivnih varijabli.
- *Style:* Ovaj dio komponente sadrži CSS kod koji definira stil komponente. Moguće je koristiti atribut 'scoped' kako bi se osigurala izolacija stilova na razini komponente, čime se izbjegava preklapanje stilova između različitih komponenti.

```vue
<template>
 <h1>{{ counter }}</h1>
 <button @click="increaseCounter">Click me!</button>
</template>

<script setup>
import { ref } from "vue";
const counter = ref(0);
const increaseCounter = () => {
 counter.value++;
};
</script>

<style scoped>
h1 {
 color: var(--red-color);
}

button {
 padding: 2rem;
 background-color: #adad;
}
</style>
```

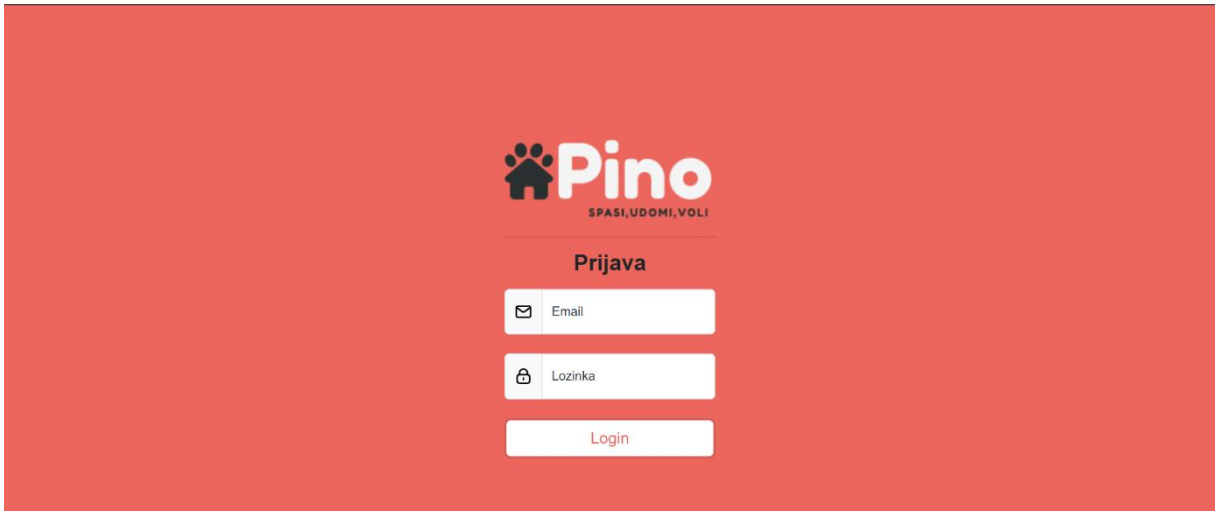
Kôd 1. Primjer osnovne strukture Vue komponente

4.4. Pino – admin

U ovom poglavlju se detaljno prikazuje izgled i funkcionalnosti administrativne aplikacije. Razrađene su različite komponente korisničkog sučelja kao i komponente glavnih funkcionalnosti.

4.4.1. Login stranica

Prijava na administrativnu aplikaciju implementirana je koristeći Firebase Authentication, što omogućava sigurnu autentifikaciju korisnika. Prijava [Slika 3.] zahtjeva unos e-mail adrese i lozinke, nakon čega aplikacija komunicira s Firebase-ovim API-em kako bi obavila autentifikaciju.



Slika 3. Login stranica

Klikom na login gumb zove se `login` funkcija [Kôd 2.] koja započinje sa postavljanjem vrijednosti isLoading na true, čime signalizira da je proces prijave u tijeku. Prikazuje se indikator učitavanja. Zatim se poziva `signInWithPasswordAndEmail` ugrađena Firebase funkcija kojoj se šalje uneseni email i password, koja te parametre šalje Firebase-u. Ukoliko je zahtjev uspješan, Firebase vraća JSON odgovor s informacijama o korisniku. U slučaju da je zahtjev neuspješan, baca se greška i korisniku je zatim prikazana poruka 'Uneseni podaci su neispravni. Molim Vas pokušajte ponovo.'. Kada se zahtjev uspješno izvrši u lokalno spremište se pohranjuje dobiveni `idToken` koji se koristi za autentifikaciju administratora te se stranica ponovno učitava, što osigurava da administrator može pristupiti dalje zaštičenom dijelu aplikacije. Svakih 50 minuta se poziva `getIdToken` funkcija [Kôd 3.] koja prilikom uspješnog odgovora vraća novi `idToken` koji se može koristiti za daljnju autentifikaciju bez potrebe za ponovnom prijavom putem email-a i passworda [10].

```
const showError = ref(false);

const isLoading = ref(false);

const email = ref("");
const password = ref("");

const login = async () => {
 try {
  isLoading.value = true;
  const auth = getAuth();
  await signInWithEmailAndPassword(auth, email.value, password.value).then(
   (response) => {
    console.log(response);
    localStorage.setItem("token", response._tokenResponse.idToken);
    location.reload();
   }
  );
 } catch (error) {
  showError.value = true;
 } finally {
  isLoading.value = false;
  email.value = "";
  password.value = "";
 }
};
```

Kôd 2. Login funkcija

```
const setTokenRefresh = () => {
 setTimeout(async () => {
  try {
   const idToken = await getAuth().currentUser.getIdToken(true);
   localStorage.setItem("token", idToken);
   setTokenRefresh();
  } catch (error) {
   console.error("Error refreshing token:", error);
  }
 }, 3000000);
};

onMounted(() => {
 if (getAuth().currentUser) {
  setTokenRefresh();
 }
});
```

Kôd 3. Inicijaliziranje token refresh intervala

4.4.2. Navigacijska traka i bočna navigacijska traka

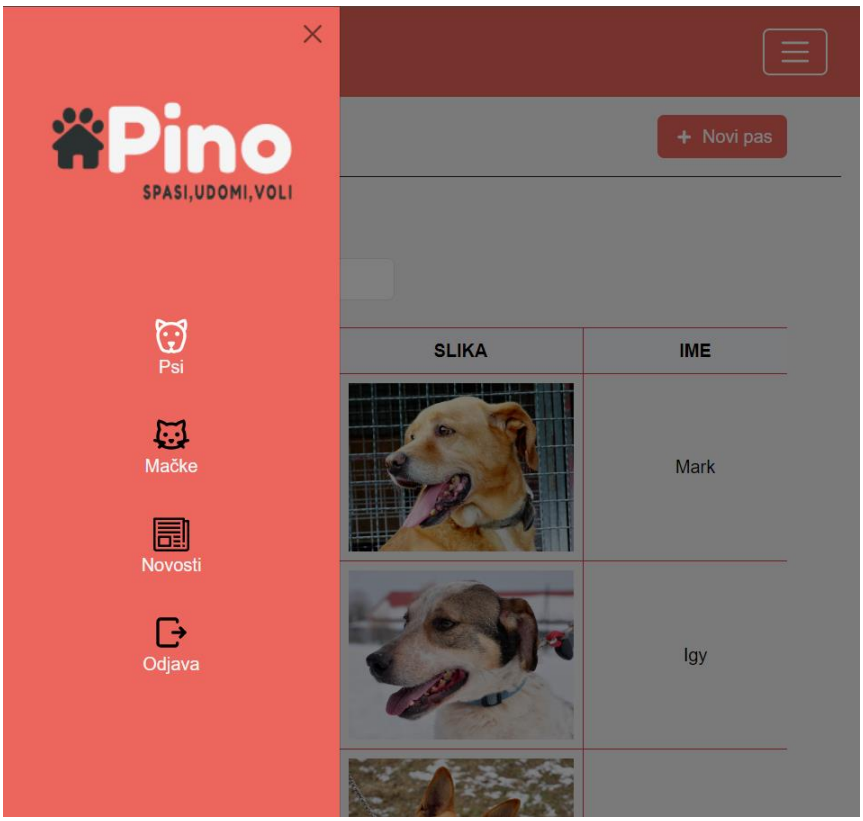
Navigacijska traka (engl. navbar) [Slika 4.] je element korisničkog sučelja koji omogućava jednostavno kretanje kroz aplikaciju. Sadrži poveznice na glavne dijelove aplikacije,kao što su:

- *Psi:* Stranica koja sadrži tablicu pasa.
- *Mačke:* Stranica koja sadrži tablicu mačaka.
- *Novosti:* Stranica koja sadrži tablicu novosti.
- *Odjava:* Odjavljivanje i povratak na login stranicu.

Implementirana je koristeći bootstrap klasu 'navbar', što osigurava responzivnost na različitim uređajima [11].

Slika 4. Navigacijska traka (navbar) za admin aplikaciju

Bočna navigacijska traka (engl. sidebar) [Slika 5.] je element korisničkog sučelja koji također omogućava jednostavno kretanje kroz aplikaciju i prisutan je samo na manjim ekranima. Implementiran je kao offcanvas bootstrap komponenta [12], što znači da se prikazuje sa strane ekrana kad korisnik klikne na hamburger gumb. Sadrži iste navigacijske linkove kao i navbar.



Slika 5. Bočna navigacijska traka (engl. sidebar) za admin aplikaciju

4.4.3. Tablični prikaz i funkcionalnosti

Stranice administrativne aplikacije koriste tablice za pregled, pretraživanje i upravljanje podacima. Svaka od stranica za upravljanje psima, mačkama i vijestima koristi tablice za prikaz na pregledan način.

Životinje ili vijesti se dohvaćaju preko `getData` funkcije [Kôd 4.] za dohvat podataka iz Firebase Realtime Database baze podataka putem HTTP GET zahtjeva. Funkcija prima dva argumenta, type koji određuje vrstu podataka koju treba dohvatiti (psi, mačke ili vijesti) i signal koji se koristi za prekidanje zahtjeva ako je potrebno (ako se brzo promjeni stranica). Ako je zahtjev uspješan odgovor se pretvara u JSON (engl. JavaScript Object Notation) format. Zatim se ti podaci transformiraju u listu objekata, pri čemu svaki objekt ima jedinstveni `id` (engl. identification) koji je ključ u bazi podataka i sve ostale podatke pridružene tom ključu. Nakon toga, oni se sortiraju prema vremenskoj oznaci (engl. timestamp) u silaznom redoslijedu, tako da su najnoviji podaci na početku odnosno vrhu tablice. Na kraju funkcija vraća sortiranu listu objekata.

```
getData(type, signal) {
  return fetch(
   `https://pino-nmpb-default-rtdb.europe-west1.firebasedatabase.app/${type}.json`,
   {
    method: "GET",
    signal: signal,
   }
  )
   .then((response) => {
    if (!response.ok) {
     throw new Error(response.statusText);
    }
    return response.json();
   })
   .then((data) => {
    const result = [];
    for (const key in data) {
     result.push({ id: key, ...data[key] });
    }
    return result.sort((a, b) => {
     const dateA = new Date(a["timestamp"]);
     const dateB = new Date(b["timestamp"]);
     return dateB - dateA;
    });
   });
 }
```

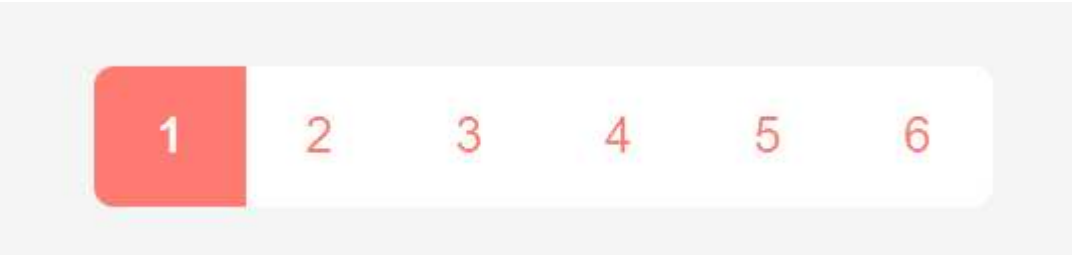
Kôd 4. Zahtjev prema Firebase-u za dohvat pasa,mačaka ili vijesti

Svaka tablica [Slika 6.] sadrži polje za pretraživanje koje omogućuje administratoru brzo pronalaženje željenih podataka. Administrator može upisati ključne riječi ili dijelove naziva kako bi suzio prikaz tablice, čime se ubrzava proces pronalaženja.

Također, stupci u tablici su sortabilni (svi osim slika), što znači da administrator može sortirati podatke prema različitim kriterijima poput imena, datuma rođenja, spola ili drugih atributa.

S obzirom da se količina podataka u tablicama s vremenom povećava implementirana je paginacija [Slika 7.], koja omogućava pregled podataka u segmentima (15 po stranici), umjesto da se svi podaci prikazuju jednom. Paginacija osigurava da sučelje ostane pregledno i administratoru omogućava lakšu navigaciju kroz veliki broj podataka [13].

Slika 6. Prikaz tablice pasa

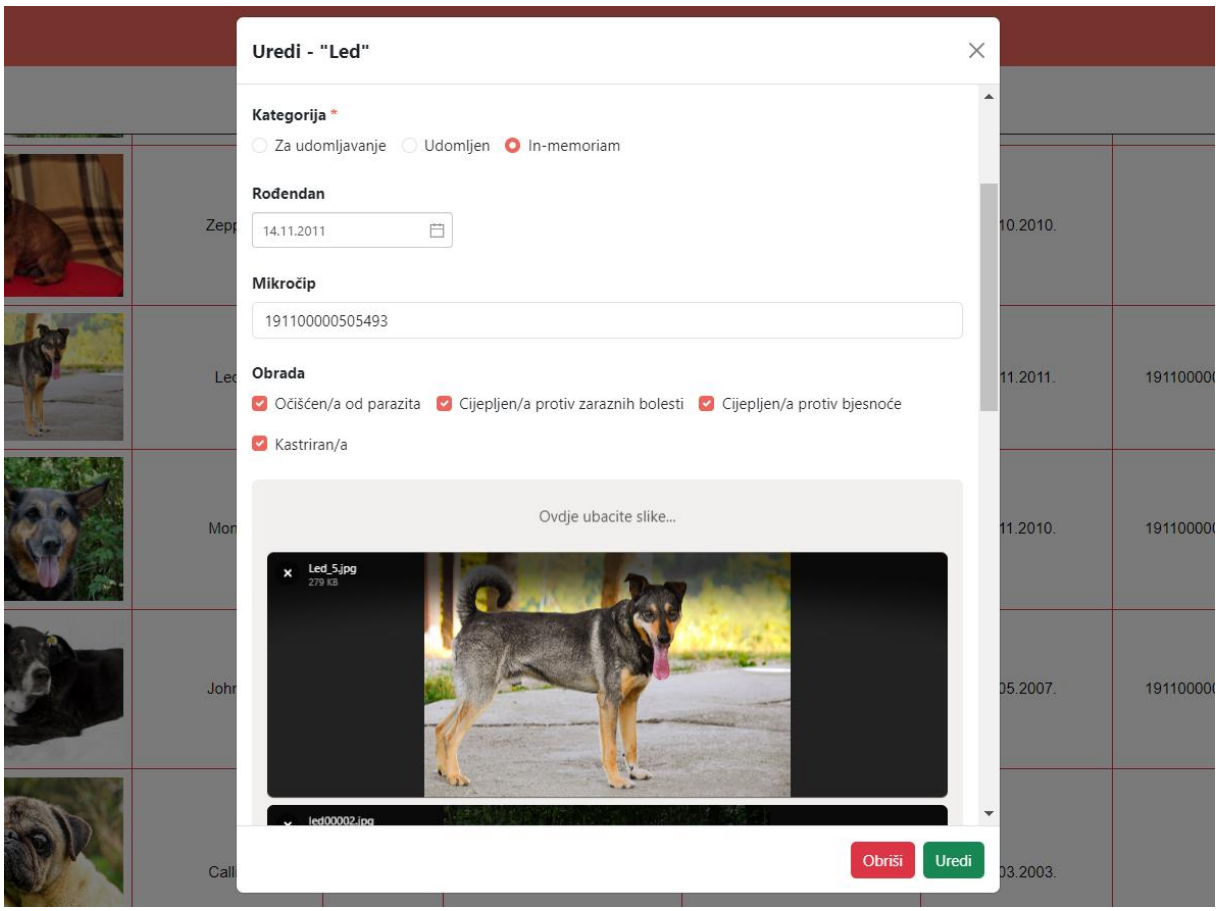


Slika 7. Paginacija

Postoji mogućnost otvaranja dijaloga za dodavanje ili ažuriranje podataka životinja ili vijesti, ovisno na kojoj stranici je administrator, putem komponente za modalni prozor.

Unutar modala se nalazi '<form>' element [Slika 8.] koji sadrži različita polja za unos podataka, kao što su ime životinje, spol, pasmina, kategorija, datum rođenja, mikročip, status obrade(npr. cijepljen, kastriran) te slike ili u slučaju vijesti naslov, tekst i slike. Sva ova polja su povezana s odgovarajućim varijablama koristeći 'v-model' što omogućava direktno ažuriranje podataka kad korisnik unese ili izmijeni vrijednost.

Forme su validirane pomoću 'novalidate' i Bootstrapovih klasa za validaciju. U slučaju krivih ili izostavljenih unosa na obaveznim poljima, administratoru se prikazuje povratna informacija putem 'invalid-feedback' elemenata.



Slika 8. Prikaz forme ažuriranja psa

Kada administrator pritisne gumb za potvrdu dodavanje ili ažuriranje unosa, validacija forme provjerava sve unesene podatke. Ako su svi podaci ispravni pokreće se odgovarajuća funkcija (add ili update) koja šalje podatke na server putem API zahtjeva.

'Obriši' gumb je prisutan u tablicama u svakom redku i u modalu ako se radi o ažuriranju postojeće životinje ili vijesti. Kada se pritisne otvara se dodatni modal za potvrdu brisanja.

Funkcije za dodavanje i ažuriranje su vrlo slične pa je dovoljno objasniti jednu od njih. `addAnimal` [Slika 9.] je asinkrona funkcija koja se koristi za dodavanje nove životinje u bazu podataka. Logika funkcije je sljedeća:

- *Provjera i inicijalizacija:* Funkcija prvo provjerava da li je već pozvana i, ako nije, onda kreće sa izvršavanjem, označava da je forma poslana kako bi se spriječilo višestruko slanje te se prikaže spinner za učitavanje.
- *Filtriranje slika:* Uklanjaju se sve slike koje imaju točno dva ključa u objektu, u kontekstu ove aplikacije to bi bile slike koje su možda nepotpuno dodane ili sadrže samo osnovne informacije.
- *Upload slika:* `firebaseStorageRef` kreira referencu na lokaciju u Firebase Storage

gdje će slika biti dodana, `uploadBytesResumable` pokreće učitavanje slika a sa `await uploadTask` se čeka završetak učitavanja. Zatim, sa `getDownloadURL` se dobiva URL (engl. Uniform Resource Locator) slike nakon što je učitana te na kraju sa `uploadedImages.push(downloadURL)` se dodaje URL slike u listu `uploadedImages`.

- *Kreiranje objekta sa podacima o životinji:* Prikupljaju se svi podaci uneseni u formu zajedno sa vremenskom oznakom dodavanja (engl. timestamp) u formatu ISO stringa.
- *Slanje podataka:* Podaci o životinji se šalju na odgovarajući API endpoint [Slika 10.] (zavisno od toga da li se radi o psu ili mački)
- *Osvježavanje stranice:* Nakon uspješnog dodavanja u lokalno spremište se pohranjuje ime akcije i ime životinje (kako bi se nakon osvježavanje stranice prikazala obavijest da je pas ili mačka sa pohranjenim imenom dodana) te se osvježi stranica.
- *Obrada grešaka:* Ako dođe do greške, funkcija resetira sve relevantne variajble i ispisuje greške u konzoli.

```
const addAnimal = async () => {
  if (numOfCalls.value === 0) {
    try {
      numOfCalls.value++;
      formSubmitted.value = true;
      const uploadedImages = [];
      images.value = images.value.filter(
        (image) => Object.keys(image).length !== 2
      );
      for (const image of images.value) {
        const storageRef = firebaseStorageRef(
          storage,
          `images/${image.filename}`
        );
        const uploadTask = uploadBytesResumable(storageRef, image.file);

        await uploadTask;

        const downloadURL = await getDownloadURL(uploadTask.snapshot.ref);

        uploadedImages.push(downloadURL);
      }
      const animalData = {
        name: name.value,
        gender: gender.value,
        breed: breed.value,
        category_query: categoryQuery.value,
        category:
          categoryQuery.value === "adopted" ||
          categoryQuery.value === "in-memoriam"
            ? [categoryQuery.value]
            : [...category.value],
        birthday: formatDate(birthday.value),
        microchip: microchip.value,
        treatment: [...treatment.value],
        images: [...uploadedImages],
        timestamp: new Date(Date.now() + 2 * 60 * 60 * 1000).toISOString(),
      };
      await apiRequests.addNew(
        route.name === "Dogs" ? "dogs" : "cats",
        animalData,
        controller.value.signal
      );
      localStorage.setItem("action", "add");
      localStorage.setItem("item", animalData.name);
      router.go(0);
    } catch (error) {
      numOfCalls.value = 0;
      formSubmitted.value = false;
      if (error.name !== "AbortError") {
        console.error(error);
      }
    }
  }
};
```

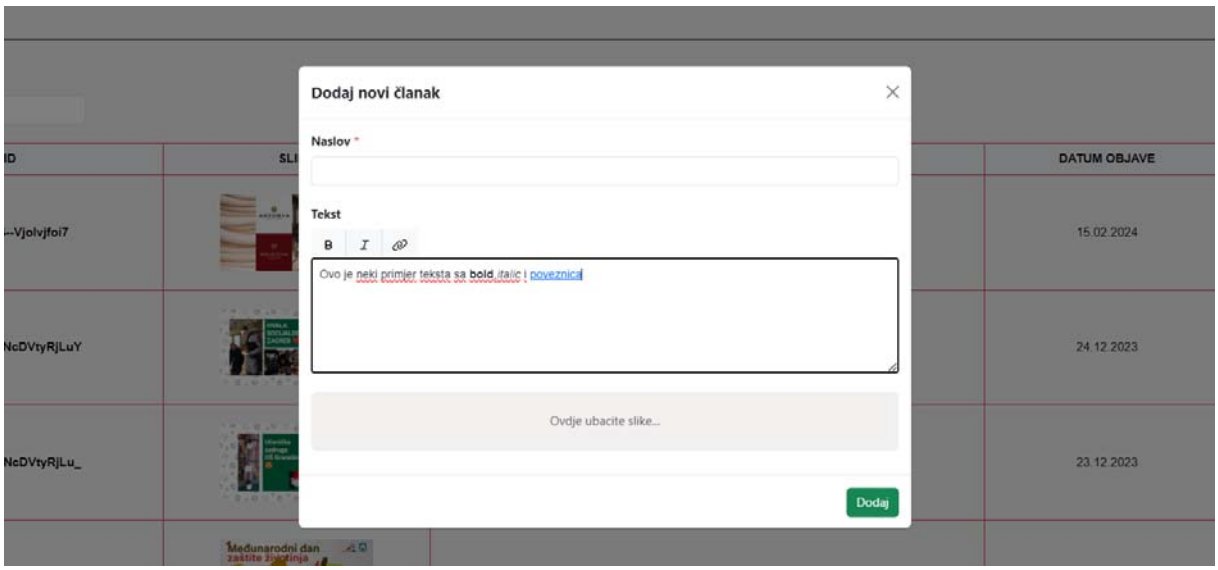
Slika 9. Zahtjev prema Firebase-u za dodavanje pasa ili mačaka

Slika 10. Struktura životinje u Firebase Realtime Database

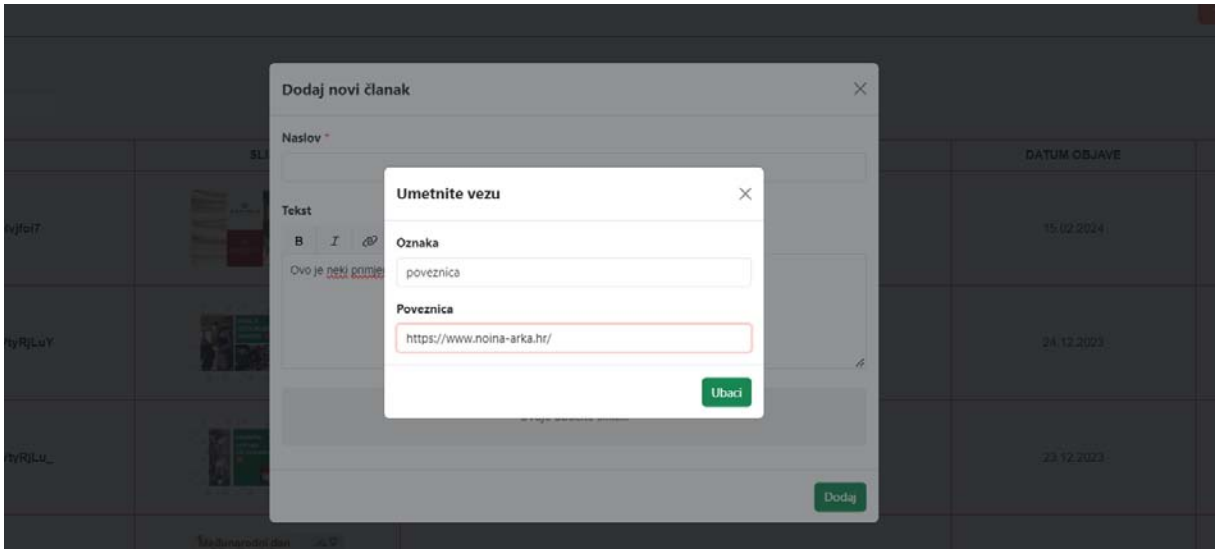
4.4.4. Uređivač teksta

Uređivač teksta je implementiran kao sastavni dio forme [Slika 11.] za tekst vijesti. Koristi `contenteditable div` element koji omogućava administratoru da unosi i uređuje tekst, bez potrebe za dodatnim bibliotekama i alatima. Ovaj uređivač teksta podržava osnovne funkcije za formatiranje teksta kao što su bold i italic te umetanje poveznica. Korištenjem `document.execCommand` funkcije, omogućeno je primjenjivanje stilova na odabrani tekst. Na primjer, klikom na gumb Bold ili koristeći prečicu Ctrl+B, aktivira se ili deaktivira bold stil na odabrani tekst. Isto tako, klikom na gumb Italic ili prečicom Ctrl+I se aktivira ili deaktivira italic stil na odabrani tekst. Stanje stilova se prati kroz `queryCommandState` funkciju, koja omogućava dinamičko osvježavanje izgleda gumba na osnovu trenutnog kursora u uređivaču [19].

Kada administrator želi umetnuti poveznicu, otvara se modal [Slika 12.] sa dva input polja, jedno za tekst koji će biti prikazan a drugo za URL adresu. Nakon validacije URL-a, poveznica se ubacuje u tekst koristeći Range API koji omogućuje pozicioniranje poveznice u tekst na poziciju kursora.



Slika 11. Prikaz forme dodavanja vijesti



Slika 12. Modal za ubacivanje poveznice

Prilikom dodavanja ili ažuriranja vijesti, tekst (sadržaj uređivača teksta) se sprema u Firebase Realtime Database [Slika 13.] kao vrijednost atributa `innerHTML`, što omogućuje spremanje svih stilova i formata koje je administrator primjenio na tekst, poput bolda, italica ili umetnutih poveznica.

Slika 13. Struktura vijesti u Firebase Realtime Database

4.4.5. Mogućnosti poboljšanja

Postoje funkcionalnosti za poboljšanje i proširenje koje mogu dodatno unaprijediti iskustvo korisnika i performanse aplikacije:

- *Više filtriranja:* Dodavanje opcije za filtriranje podataka prema više kriterija istovremeno, npr. pretraživanje životinja prema pasmini i kategoriji ili bilo kakve druge kombinacija.
- *Proširena validacija:* Složenija pravila validacija, poput provjere ispravnosti slika (npr. veličina, format).
- *Dinamične povratne informacije:* Povratne informacije koje odma obavještavaju administratora o pogreškama u unosu.
- *Role i prava:* Implementacija naprednih mehanizama za upravljanje ulogama i pravima unutar aplikacije kako bi se omogućilo različitim korisnicima različite razine pristupa i kontrole.
- *Personalizacija sučelja:* Omogućavanje administratoru da personalizira svoje sučelje odabirom između različitih tema boja ili drugačijih prikaza podataka.

Implementacija ovih poboljšanja može značajno unaprijediti korisničko iskustvo, čineći aplikaciju učinkovitijom za administratore.

4.5. Pino – korisnička aplikacija

U ovom poglavlju se opisuju sve ključne funkcionalnosti i komponente korisničkog sučelja aplikacije 'Pino', koja je namijenjena krajnjim korisnicima, odnosno potencijalnim udomiteljima i posjetiteljima udruge. Aplikacija je dizajnirana da bude intuitivna i informativna.

4.5.1. Navigacijska traka,bočna navigacijska traka i podnožje

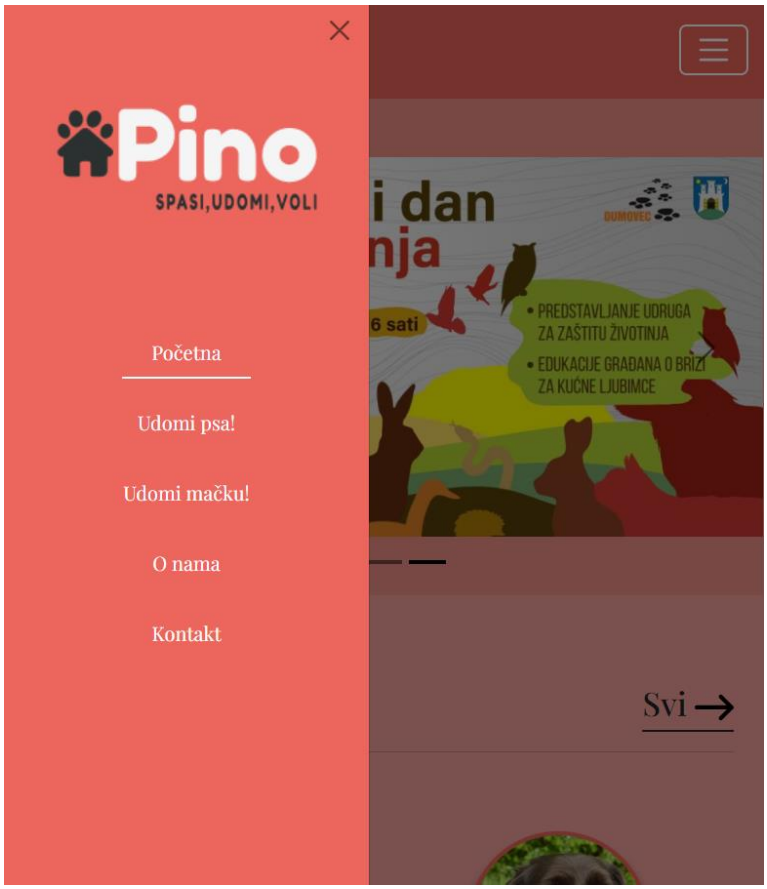
Navigacijska traka (engl. navbar) [Slika 14.] je element korisničkog sučelja koji omogućava jednostavno kretanje kroz aplikaciju. Sadrži poveznice na glavne dijelove aplikacije, kao što su:

- *Početna:* Glavna (početna) stranica aplikacije koja korisnicima pruža uvid u aktivnosti udruge te ih potakne na daljnje istraživanje i udomljavanje.
- *Udomi psa!:* Stranica koja sadrži listu pasa za udomljavanje, udomljenih pasa te uginulih pasa u udruzi.
- *Udomi mačku!:* Stranica koja sadrži listu mačaka za udomljavanje, udomljenih mačaka te uginulih mačaka u udruzi.
- *Novosti:* Stranica sa novostima i događajima u vezi s udrugom.
- *O nama:* Stranica s detaljima o povijesti i misiji udruge.
- *Kontakt:* Stranica s kontakt podacima udruge i formom za slanje upita.

Kao i u admin aplikaciji implementirana je koristeći bootstrap klasu 'navbar', što osigurava responzivnost na različitim uređajima [11].

Slika 14. Navigacijska traka (engl. navbar) za korisničku aplikaciju

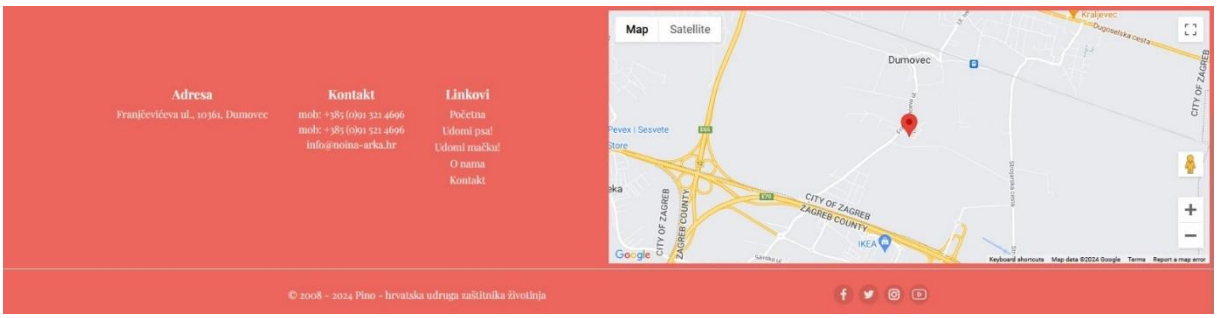
Bočna navigacijska traka (engl. sidebar) [Slika 15.] je element korisničkog sučelja koji također omogućava jednostavno kretanje kroz aplikaciju i prisutan je samo na manjim ekranima. Kao i u admin aplikaciji implementiran je kao offcanvas bootstrap komponenta [12], što znači da se prikazuje sa strane ekrana kad korisnik klikne na hamburger gumb. Sadrži iste navigacijske linkove kao i navbar.



Slika 15. Bočna navigacijska traka (engl. sidebar) za korisničku aplikaciju

Podnožje (engl. footer) [Slika 16.] je smješteno na dnu svake stranice i sadrži dodatne informacije i linkove korisne za korisnike. Sadrži sljedeće:

- *Kontakt podaci:* Detalji poput adrese, broja telefona i email adrese udruge.
- *Google map:* Ugrađena Google API mapa koja prikazuje lokaciju udruge. Korisnicima omogućava jednostavno pronalaženje adrese udruge.
- *Linkovi na društvene mreže:* Gumbovi koji vode na Facebook, Twitter, Instagram i Youtube profile udruge, omogućuje korisnicima da prate rad udruge na drugim platformama.



Slika 16. Podnožje (engl. footer)

4.5.2. Početna stranica

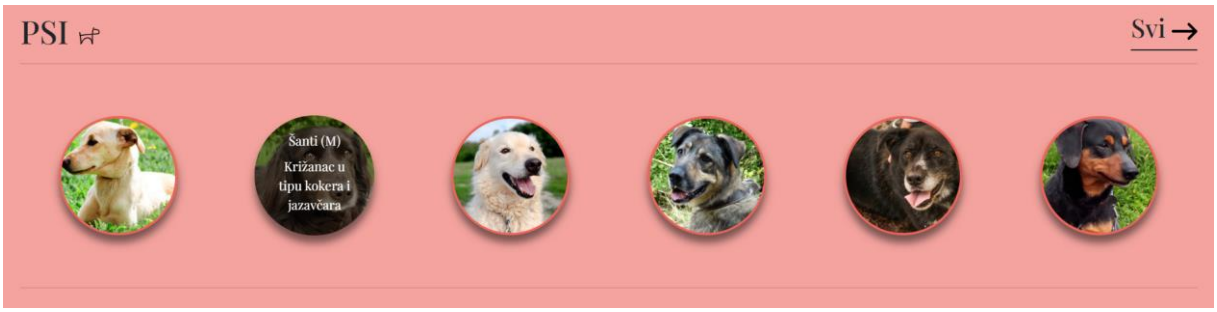
Početna stranica je dizajnirana tako da korisnicima pruža brz i pregledan uvid u sadržaje vezane za udrugu. Stranica sadrži nekoliko ključnih sekcija koje omogućuju jednostavan pristup životinjama za udomljavanje i novostima.

Na vrhu početne stranice nalazi se karusel za prikaz istaknutih novosti. Slajdovi sadrže sliku i naslov. Automatski mijenja slajdove svakih 10 sekundi, omogućujući korisnicima da bez potrebe za interakcijom pregledaju najnovije novosti, međutim korisnici mogu i ručno prebacivati između slajdova [14].

Odmah ispod karusela nalazi se sekcija 'PSI' [Slika 17.] koja prikazuje 6 pasa dostupnih za udomljavanje. Ovdje su istaknuti samo neki od pasa, kako bi korisnici mogli brzo vidjeti nekoliko pasa za udomljavanje. Za korisnike koji žele vidjeti cjelokupnu listu pasa, dostupan je gumb 'SVI' koji vodi na stranicu s pregledom svih pasa.

Sekcija 'MAČKE' se nalazi ispod pasa i funkcionira na isti način. Prikazuje 6 mačaka dostupnih za udomljavanje, s opcijom pregleda svih mačaka putem gumba 'SVE'.

Svaka kartica životinje sadrži sliku životinje te na prijelaz kursorom se prikaže ime,spol i pasmina za tu životinju.



Slika 17. Sekcija 'PSI' na početnoj stranici

Na kraju početne stranice se nalazi sekcija 'NOVOSTI' koja prikazuje 4 najnovije novosti vezane uz rad udruge. Svaka kartica novosti [Slika 18.] sadrži naslov, početak teksta i datum objave. Za korisnike koji žele pregledati sve vijesti tu je gumb 'SVE'.

Slika 18. Kartica novosti

Sve kartice, i one za životinje i za novosti imaju tranzicije za dinamičniji i zanimljiviji prikaz te klik na bilo koju karticu vodi na stranicu detalja vezano za istu.

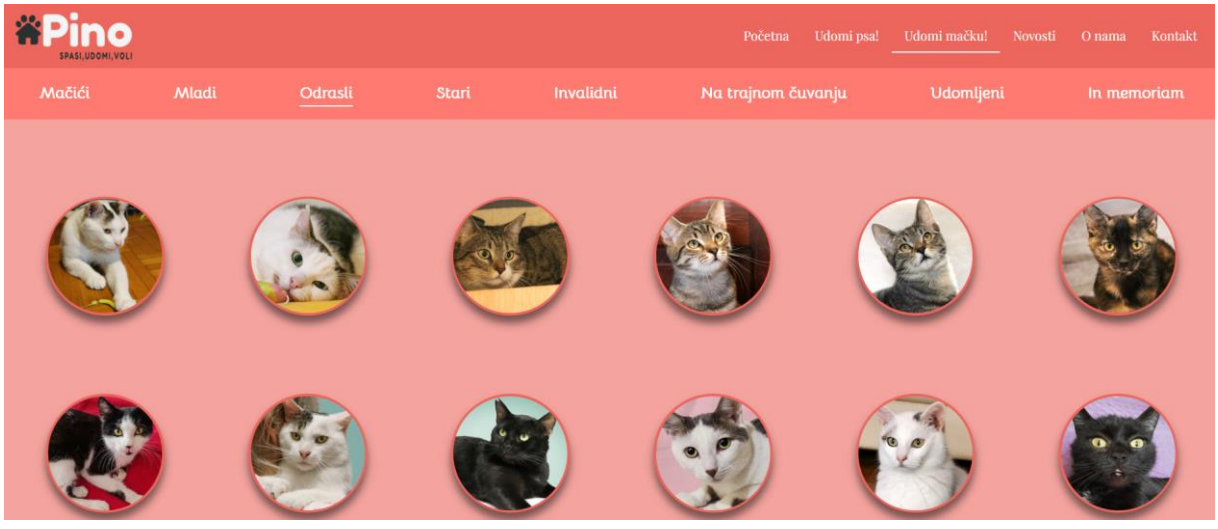
4.5.3. Stranice “Udomi psa”,”Udomi mačku” i “Novosti”

Sve 3 stranice dijele sličan raspored i funkcionalnosti te zato dijele, odnosno koriste, istu Vue komponentu, što omogućava efikasnije održavanje koda. Logika koja kontrolira prikaz sadržaja ovisi o trenutnoj ruti i odgovarajućim podacima.

Ključne karakteristike ovih stranica [Slika 19.] su prikaz svih subjekata, paginacija te podnavigacijska traka (kod životinja) koja korisnicima omogućava filtriranje sadržaja prema različitim kategorijama:

- *Štenci/Mačići:* Prikazuje tek nedavno rođene životinje.
- *Mladi:* Prikazuje mlađe životinje koje su izašle iz stadija štenaca/mačića.
- *Odrasli:* Prikazuje Odrasle životinje u punoj snazi.
- *Stari:* Prikazuje starije životinje u trećoj dobi.
- *Invalidni:* Prikazuje životinje s posebnim zdravstvenim problemima i potrebama.
- *Na trajnom čuvanju:* Prikazuje životinje koje su pod stalnom skrbi udruge te nisu dostupne za udomljavanje.
- *Udomljeni:* Prikazuje životinje koje su sretno pronašle dom.
- *In memoriam:* Prikazuje životinje koje su uginule a nalazile su se u udruzi.

Na svakoj od ovih stranica sadržaj se prikazuje u obliku kartica, gdje svaka kartica predstavlja pojedinačnu životinju ili novost. Klikom na bilo koju karticu, korisnik se preusmjerava na stranicu s detaljima, gdje korisnik može dobiti dublji uvid u odabranu životinju ili vijest.



Slika 19. "Udomi mačku" stranica sa odabranim "Odrasli" filterom

4.5.4. Stranica “O nama”

Stranica “O nama” [Slika 20.] pruža korisnicima pogled u povijest, misiju i rad udruge [8]. Dizajnirana je s naglaskom na vizualnu prezentaciju i lakoću čitanja, uz upotrebu tranzicija kako bi se poboljšalo korisničko iskustvo. Na stranici se koristi tranzicija koja se aktivira kada korisnik pregledava stranicu. Korisi se IntersectionObserver API [Kôd 5.] kako bi se postigao efekt postupnog pojavljivanja na ekranu kako korisnik pregledava sadržaj.

Slika 20. Isječak "O nama" stranice

```
<script setup>
import { onMounted } from "vue";

const isSmallViewport = window.innerWidth < 362;

const observer = new IntersectionObserver(
 (entries) => {
  entries.forEach((entry) => {
   if (entry.isIntersecting) {
    entry.target.classList.add("scroll-animation");
    observer.unobserve(entry.target);
   }
  });
 },
 { rootMargin: isSmallViewport ? "0px 0px 0px 200px" : "0px" }
);

onMounted(() => {
 const itemsToAnimate = document.querySelectorAll(".animate");

 for (let i = 0; i < itemsToAnimate.length; i++) {
  const elements = itemsToAnimate[i];

  observer.observe(elements);
 }
});
</script>

<style scoped>
.animate {
 opacity: 0;
 transform: translateX(-300px);
 transition: all 0.7s ease-out;
 transition-delay: 0.4s;
}

.scroll-animation {
 opacity: 1;
 transform: translateX(0);
}
/*other css*/
</style>
```

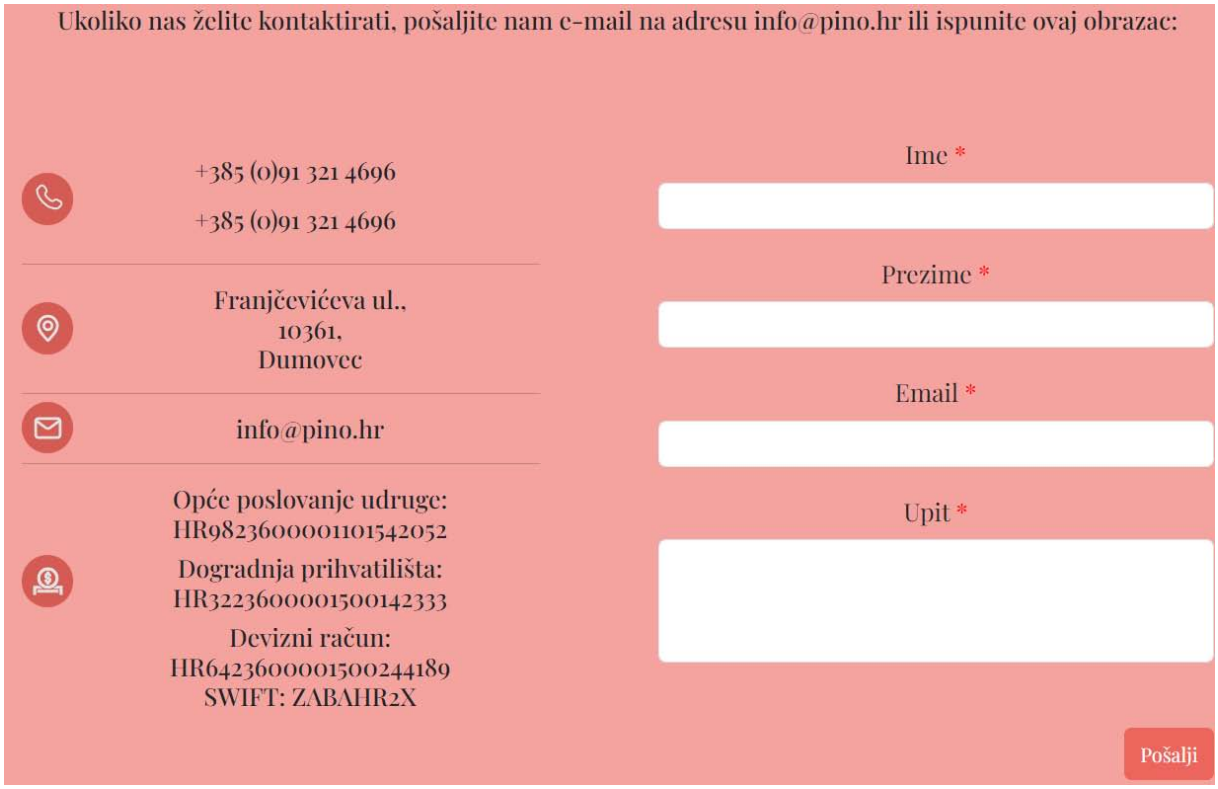
Kôd 5. Implementacija IntersectionObserver API-a sa tranzicijama

4.5.5. Kontakt stranica

Na kontakt stranici [Slika 21.] korisnici mogu brzo i jednostavno stupiti u kontakt s udrugom putem različitih kanala komunikacije. Kontakt informacije su:

- *Telefonski brojevi*
- *Adresa*
- *Email adresa*
- *Bankovni podaci*

Stranica također sadrži kontakt formu koja omogućava korisnicima slanje upita direktno putem stranice. Forma [Slika 21.] sadrži obavezno polje za ime, prezime, email adresu i poruku. Implementirana je validacija forme koja osigurava da su sva obavezna polja ispravno popunjena prije slanja. Nakon uspješnog slanja forme, prikazuje se poruka zahvalnosti uz potvrdu da je upit zaprimljen.



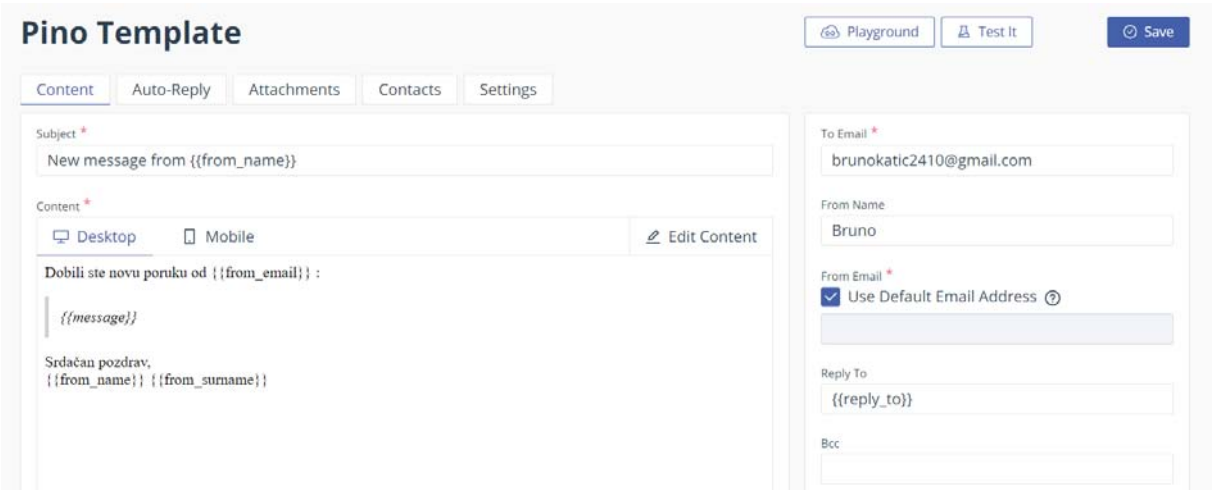
Slika 21. Kontakt informacije i kontakt forma

Proces slanja e-maila preko EmailJS-a počinje prijavom na EmailJS platformu (kreiranje računa), zatim dodavanjem email servisa (npr.Gmail, Outlook) koji će se koristiti za slanje e-mailova. Tu se dobije jedinstveni ID za taj servis. Nakon toga se kreira predložak [Slika 22.]

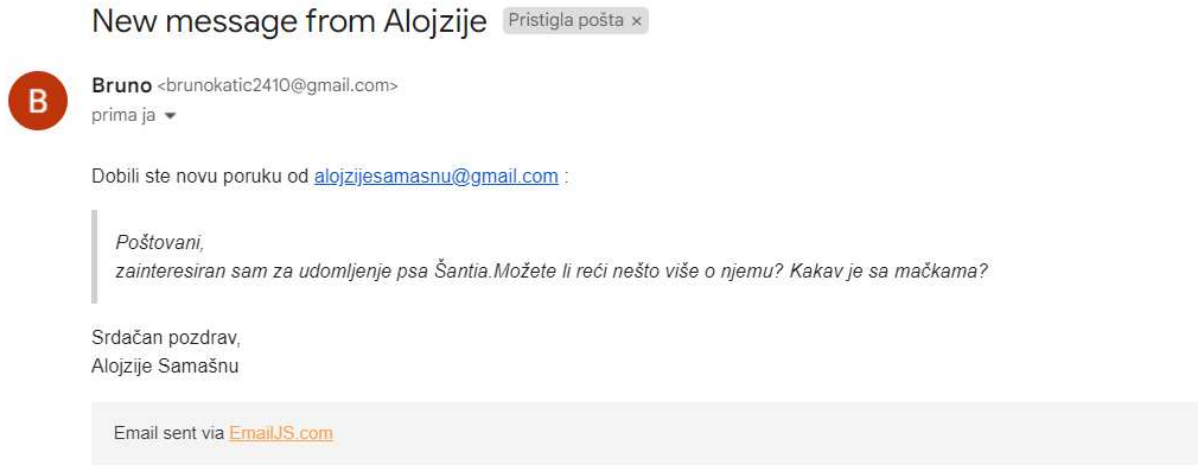
koji definira kako će mail izgledati [Slika 23.], uključujući dinamičke varijable koje će se zamjeniti stvarnim podacima [15]. Javni ključ se koristi za autentifikaciju kod slanja e-maila iz koda [Kôd 6.]

```
const sendEmail = () => {
 emailjs
  .sendForm("pino_contact", "pino_template", form.value, {
   publicKey: "SvtzovAEZMfDGJTeQ",
  })
  .then(
   () => {
    if (!isFormSuccessful.value) {
     isFormSuccessful.value = true;
    }
    const formElement = form.value;
    formElement.reset();
    formElement.classList.remove("was-validated");
   },
   (error) => {
    console.log(error.text);
   }
  );
};
```

Kôd 6. Funkcija za slanje e-maila



Slika 22. E-mail predložak



Slika 23. Primjer primljene kontakt forme

4.5.6. Stranica detalja za životinje

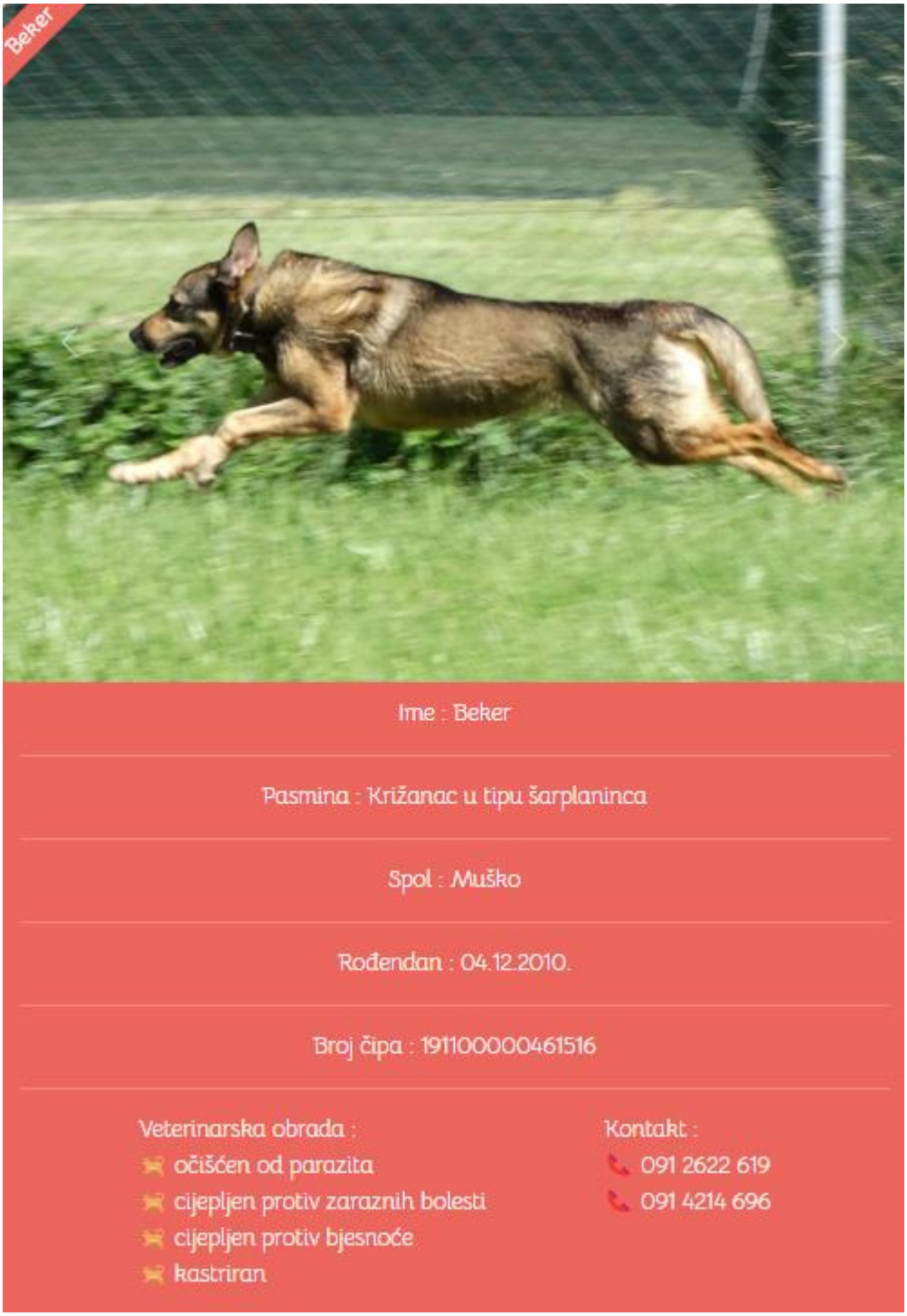
Stranica detalja za životinje [Slika 24.] omogućava pregled specifičnih informacija o životinjama. Na vrhu stranice nalazi se podnavigacijska traka koja omogućava korisnicima da prebacuju između različitih kategorija životinja.

Kada se stranica učita, u `onMounted` hook se poziva `getAnimalsById` funkcija [Kôd 7.] koja prvo određuje tip životinje na osnovu imena rute. Zatim se iz parametra URL-a izvlači ID životinje koristeći regularni izraz koji traži 19 znakova iza znaka minus. Taj ID i tip životinje se šalju API-u koji vraća podatke o životinji za taj ID te prethodnoj i sljedećoj u listi [Slika 25.]. Podaci se zatim dinamički prikazuju na stranici.

Glavni dio stranice sadrži karticu životinje koja sadrži karusel slika odabrane životinje. Karusel omogućava korisnicima pregled više slika, a kontrole za prijelazak između slika je omogućen samo ako postoji više od jedne slike.

Ispod karusela prikazane su osnovne informacije o životinji, poput njihovog imena, pasmine, spola, rođendana, broja čipa, veterinarske obrade te brojevi mobitela za ljude koji žele dodatno saznati o životinji ili su zainteresirani za udomljavanje.

Na dnu stranice, prikazane su kartice za prethodnu i sljedeću životinju (ako postoje), što omogućava jednostavno pregledavanje drugih životinja.



Slika 24. Detaljni prikaz psa

Slika 25. Zahtjev prema Firebase-u za dohvat trenutne,prethodne i sljedeće životinje

```
const getAnimalsById = async () => {
 const type = route.name === "Dog details" ? "dogs" : "cats";
 const id = route.params.slug.match(/-(.{19})$/);
 try {
  const { currentAnimal, prev, next } =
   await apiRequests.getPrevCurrentNextAnimalById(type, id[0]);
  animal.value = currentAnimal;
  prevAnimal.value = prev;
  nextAnimal.value = next;
 } catch (error) {
  console.error(error);
 }
};
```

Kôd 7. Funkcija za dohvat životinja prema ID-u

4.5.7. Stranica detalja za novosti

Stranica detalja za novosti [Slika 26.] omogućava kompletan pregled novosti, uključujući cijeli tekst i sve slike. Na vrhu stranice nalazi se podnavigacijska traka koja omogućava korisnicima da prebacuju između različitih kategorija životinja.

Kada se stranica učita, u `onMounted` hook [Kôd 8.] se poziva `getNewsById` funkcija koja iz parametra URL-a izvlači ID novosti koristeći regularni izraz koji traži 19 znakova iza znaka minus. Taj ID se šalje API-u koji vraća podatke o novosti za taj ID te prethodnoj i sljedećoj u listi. Zatim se poziva `showText` funkcija [Slika 27.] koja koristi `document.createElement` i `document.createTextNode` da bi se kreirala odgovarajuća struktura DOM-a iz HTML stringa za tekst novosti. Ova funkcija također vodi računa o formatiranju teksta, uključujući linkove, podebljane i kurzivne riječi te prijelomne linije. Na kraju se inicijalizira pravilno raspoređivanje slika na stranici koristeći Masonry layout biblioteku, ali tek nakon što su sve slike učitane koristeći ImagesLoaded biblioteku kako bi se osiguralo da su sve slike učitane prije nego Masonry izračuna i primjeni svoj raspored slaganja [16], [17]. Podaci se zatim dinamički prikazuju na stranici.

Glavni dio stranice sadrži karticu novosti koja sadrži glavnu sliku, naslov, tekst te ostale slike. Na dnu stranice, prikazane su kartice za prethodnu i sljedeću novost (ako postoji), što omogućava jednostavno pregledavanje drugih novosti.

Slika 26. Detaljni prikaz novosti

Slika 27. Funkcija za formatiranje teksta iz HTML string-a

```
onMounted(async () => {
 await getArticlesById();
 isLoading.value = false;
 nextTick(() => {
  showText();
  const imgContainers = document.querySelectorAll("[data-masonry] div");
  imagesLoaded(imgContainers, () => {
   const row = document.querySelector("[data-masonry]");
   if (row) {
    new Masonry(row, {
     percentPosition: true,
    });
   }
  });
 });
});
```

Kôd 8. Pozivanje funkcija u onMounted hook-u i kreiranje Masonry-a

4.5.8. Mogućnosti poboljšanja

Kao i za admin aplikaciju, postoje funkcionalnosti za poboljšanje i proširenje korisničke aplikacije koje bi mogle dodatno unaprijediti iskustvo korisnika i performanse aplikacije:

- *Pretraga i napredno filtriranje:* Implementacija pretrage i naprednijeg filtriranja za sekcije kao što su "Udomi psa" i "Udomi mačku". Na primjer, dodavanje filtera po veličini životinje ili specifičnim potrebama životinja bi korisnicima omogućilo lakše pronalaženje životinje koja odgovara njihovim željama.
- *Multijezična podrška:* Dodavanje podrške za više jezika kako bi aplikacija bila dostupna širem krugu korisnika.
- *Vrijeme za posjete:* Dodavanje informacije o vremenu za posjete u udrugu.
- *Anketa za volontiranje:* Implementacija forme u obliku ankete kao prijavnica za volontiranje u udruzi te informacije o samom volontiranju u udruzi.

Svaka od ovih funkcionalnosti može doprinjeti boljem korisničkom iskustvu, čime se povećava privlačnost i efikasnost za krajnje korisnike.

5. ZAKLJUČAK

Ovaj završni rad predstavlja detaljan pregled razvoja i implementacije mrežnih aplikacija Pino-admin i Pino koje su zamišljene kao digitalna rješenja za upravljanje azilima ili udrugama i procesima udomljavanja životinja. Aplikacije su izgrađene koristeći moderne tehnologije poput Vue.js framework-a i alate i biblioteke, uključujući Visual Studio Code, Vue CLI, ESLint, Vue Router i Bootstrap, čijim korištenjem je osigurana kvaliteta koda i responzivnost dizajna.

Posebna pažnja u procesu implementacije je posvećena ključnim funkcionalnostima koje odgovaraju stvarnim potrebama korisnika. U Pino-admin aplikaciji implementirane su funkcionalnosti poput prijave korisnika, tablični prikaz svih životinja i novosti te dodavanje, uređivanje i brisanje istih. S druge strane korisnička aplikacija Pino nudi intuitivno sučelje za pregled životinja za udomljavanje, udomljenih životinja te uginulih životinja udruge, kao i pregled novosti i informacija o samoj udruzi i njenom radu. Ove funkcionalnosti značajno unapređuju interakciju korisnika s aplikacijama i omogućuju bolju komunikaciju.

Rad se osvrće na konkretne probleme u vođenju azila ili udruga, kao što su potrebe za efikasnim upravljanjem podacima o životinjama, jednostavnost pristupa informacijama za potencijalne udomitelje te transparentnost u komunikaciji s javnošću. Za izgradnju sustava koji ispunjava te zahtjeve korištene su tehnologije poput Firebase-a za autentifikaciju i pohranu podataka, EmailJS za slanje upita te Vue3-google-map za integraciju karte (fizička lokacija udruge).

Osim toga, aplikacije koriste vizualne alate kao što su ImagesLoaded i Masonry, koji poboljšavaju korisničko iskustvo kroz dinamičan prikaz slika i podataka. Filepond i Vue-datepicker-next osiguravaju interakciju s datotekama i odabir datuma. Ključnu ulogu u dizajnu obje aplikacije ima Bootstrap, koji osigurava responzivnost na različitim uređajima te moderan izgled. Sučelje aplikacija je dizajnirano da bude jednostavno i lako razumljivo za korisnike, olakšavajući im pristup potrebnim informacijama i povećavajući šanse za uspješno udomljavanje životinja. Također, aplikacije omogućuju jasnu i transparentnu komunikaciju s javnošću, pružajući redovite novosti i informacije o radu udruge što je ključno za izgradnju povjerenja.

Iako su aplikacije kreirane u kontekstu fiktivne udruge, one predstavljaju funkcionalan model koji se može prilagoditi i implementirati u stvarnim organizacijama koje se bave zbrinjavanjem i udomljavanjem životinja. Podaci korišteni u aplikacijama preuzeti su iz stvarne udruge Noina Arka. Još jednom, posebna zahvalnost predsjednici udruge, Heleni Fink, koja je dozvolila korištenje podataka i podržala ovaj rad.

Rad je pokazao da je primjena modernih mrežnih tehnologija, alata i biblioteka ključna za izgradnju učinkovitih rješenja za izazove upravljanja organizacijama koje se bave

zbrinjavanjem i udomljavanjem životinja.

LITERATURA

1. Au-Yeung, John, (2021), *Vue.js 3 By Example,* Packt (pristupljeno 11. 9. 2024.)
2. https://vuejs.org/guide/introduction.html (pristupljeno 11. 9. 2024)
3. wikipedia, Vue.js – Wikipedia, https://en.wikipedia.org/wiki/Vue.js (pristupljeno 11. 9. 2024.)
4. mygreatlearning.com, History of ReactJS – Great Learning, https://www.mygreatlearning.com/react-js/tutorials/history-of-reactjs (pristupljeno 11. 9. 2024.)
5. Dave Gavigan, (2018), The History of Angular**,** https://medium.com/the-startup-lab-blog/the-history-of-angular-3e36f7e828c7 (pristupljeno 11. 9. 2024.)
6. Ogbu Uzoma, (2023), Brief history of NodeJS, https://medium.com/@ogbuuzoma413/brief-history-of-nodejs-de0cac0af448 (pristupljeno 11. 9. 2024.)
7. pythoninstitute.org, About Python, https://pythoninstitute.org/about-python (pristupljeno 11. 9. 2024.)
8. javatpoint.com, Father of PHP, https://www.javatpoint.com/father-of-php (pristupljeno 11. 9. 2024.)
9. noina-arka.hr, Noina Arka – Hrvatska udruga zaštitnika životinja, https://www.noina-arka.hr/ (pristupljeno 1. 6. 2024.)
10. cli.vuejs.org, Overview | Vue CLI, https://cli.vuejs.org/guide/ (pristupljeno 2. 6. 2024.)
11. Adekola Olawale, (2022), Firebase Authentication, https://medium.com/@Adekola_Olawale/firebase-authentication-413626c5234d (pristupljeno 4. 6. 2024.)
12. getbootstrap.com, Navbar · Bootstrap v5.3, https://getbootstrap.com/docs/5.3/components/navbar/ (pristupljeno 2. 6. 2024.)
13. getbootstrap.com, Offcanvas · Bootstrap v5.3, https://getbootstrap.com/docs/5.3/components/offcanvas/ (pristupljeno 2. 6. 2024.)
14. getbootstrap.com, Pagination · Bootstrap v5.3, https://getbootstrap.com/docs/5.3/components/pagination/ (pristupljeno 2. 6. 2024.)
15. getbootstrap.com, Carousel · Bootstrap v5.3, https://getbootstrap.com/docs/5.3/components/carousel/ (pristupljeno 2. 6. 2024.)
16. Ezekiel Lawson, (2023), Creating a Contact Form with EmailJS and Vue https://www.telerik.com/blogs/creating-contact-form-emailjs-vue (pristupljeno 4. 6. 2024.)
17. imagesloaded.desandro.com, imagesLoaded, https://imagesloaded.desandro.com/ (pristupljeno 4. 6. 2024.)
18. masonry.desandro.com, Masonry, https://masonry.desandro.com/ (pristupljeno 4. 6. 2024.)
19. developer.mozilla.org, Intersection Observer API, https://developer.mozilla.org/en-US/docs/Web/API/Intersection_Observer_API (pristupljeno 4. 6. 2024.)

20. developer.mozilla.org, Document: queryCommandState() method, https://developer.mozilla.org/en-US/docs/Web/API/Document/queryCommandState (pristupljeno 4. 6. 2024.)

SAŽETAK

Ovaj završni rad se bavi razvojem i implementacijom dviju mrežnih aplikacija, Pino i Pino-admin, koje simuliraju upravljanje udrugom/azilom za životinje. Cilj aplikacija je olakšati administrativne poslove povezane s vođenjem azila, kao i korisnicima pregled informacija i udomljavanje životinja. Rad započinje uvodom u Vue.js tehnologiju te analizira Vue 2 i Vue 3 verzije. Opisani su korišteni alati i biblioteke poput Firebase-a, Bootstrap-a i Vue Routera. Detaljno su obrađeni problemi s kojima se susreću udruge i azili koji se bave zbrinjavanjem životinja, poput učinkovitog vođenja podataka i informiranja potencijalnih udomitelja te su ponuđena rješenja kroz funkcionalnosti aplikacija. Pino-admin aplikacija služi sa upravljanje podacima o životinjama i novostima u udruzi, dok Pino aplikacija pruža intuitivno korisničko sučelje za pregled životinja, novosti i informacija o udruzi. Istaknute su mogućnosti daljnjeg poboljšanja funkcionalnosti za obje aplikacije. Rad uključuje stvarne podatke preuzete od udruge Noine Arke, što dodatno doprinosi praktičnosti i vjerodostojnosti ovog rada.

Ključne riječi: Vue, JavaScript, Firebase, Bootstrap, životinja, udomljavanje

SUMMARY

This final thesis focuses on the development and implementation of two web applications, Pino and Pino-admin, which simulate the management of animal shelter/organization. The aim of these applications is to simplify administrative tasks related to running a shelter and to provide users with easy access to information and options for adopting animals. The thesis begins with an introduction to Vue.js technology, analyzing both Vue 2 and Vue 3 versions. It describes tools and libraries used, such as Firebase, Bootstrap and Vue Router. It thoroughly addresses the challenges faced by shelters and organizations involved in animal care, such as efficient data management and informing potential adopters, and offers solutions through the functionalities of the applications. The Pino-admin application is designed for managing data on animals and news withing the organization, while the Pino application provides an intuitive user interface for viewing animals, news and information about the organization. Possible improvements to the functionality of both applications are highlighted. The work includes real data obtained from the Noina Arka organization, which further contributes to the practicality and credibility of this thesis.

Keywords: Vue, JavaScript, Firebase, Bootstrap, animal, adoption