

# Web trgovina za ljubimce

---

Šuljug, Katarina

Undergraduate thesis / Završni rad

2024

*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:200:065961>

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

*Download date / Datum preuzimanja:* **2025-04-03**

*Repository / Repozitorij:*

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



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

**Stručni studij Računarstvo**

**WEB TRGOVINA ZA LJUBIMCE**

**Završni rad**

**Katarina Šuljug**

**Osijek, 2024.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMATIJSKIH TEHNOLOGIJA OSIJEK**Obrazac Z1S: Obrazac za ocjenu završnog rada na stručnom prijediplomskom studiju****Ocjena završnog rada na stručnom prijediplomskom studiju**

Ime i prezime pristupnika:	Katarina Šuljug
Studij, smjer:	Stručni prijediplomski studij Računarstvo
Mat. br. pristupnika, god.	AR 4823, 27.07.2020.
JMBAG:	0165085297
Mentor:	Marina Peko, dipl. ing. el.
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	doc. dr. sc. Ivana Hartmann Tolić
Član Povjerenstva 1:	Marina Peko, dipl. ing. el.
Član Povjerenstva 2:	Robert Šojo, univ. mag. ing. comp.
Naslov završnog rada:	Web trgovina za ljubimce
Znanstvena grana završnog rada:	<b>Procesno računarstvo (zn. polje računarstvo)</b>
Zadatak završnog rada:	Izraditi web trgovinu koja prodaje proizvode za ljubimce. Korisnik može biti registriran prilikom kupovine i skupljati bodove/ostvariti popuste ili biti gost. Uloga prodavača na stranici omogućava editiranje i postavljanje novih proizvoda za prodaju te brisanje istih. Naplata se vrši putem uplatnice dostavljene na e-mail. Tema rezervirana za: Katarina Šuljug
Datum ocjene pismenog dijela završnog rada od strane mentora:	23.09.2024.
Ocjena pismenog dijela završnog rada od strane mentora:	Izvrstan (5)
Datum obrane završnog rada:	30.09.2024.
Ocjena usmenog dijela završnog rada (obrane):	Izvrstan (5)
Ukupna ocjena završnog rada:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio stručni prijediplomski studij:	30.09.2024.

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

Osijek, 30.09.2024.

<b>Ime i prezime Pristupnika:</b>	Katarina Šuljug
<b>Studij:</b>	Stručni prijediplomski studij Računarstvo
<b>Mat. br. Pristupnika, godina upisa:</b>	AR 4823, 27.07.2020.
<b>Turnitin podudaranje [%]:</b>	11

Ovom izjavom izjavljujem da je rad pod nazivom: **Web trgovina za ljubimce**

izrađen pod vodstvom mentora Marina Peko, dipl. ing. el.

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 pristupnika:

# SADRŽAJ

<b>1. UVOD</b> .....	1
<b>1.1. Zadatak završnog rada</b> .....	2
<b>2. POSTOJEĆE WEB STRANICE</b> .....	2
<b>2.1. Pet Network International d.o.o. za trgovinu i usluge – pet-centar.hr</b> .....	2
<b>2.2. Zoocity.hr</b> .....	3
<b>2.3. Pet Home Shop</b> .....	4
<b>2.4. PetSmart</b> .....	5
<b>2.5. Šapica d.o.o.</b> .....	6
<b>3. TEHNOLOGIJE KORIŠTENE ZA IZRADU APLIKACIJE</b> .....	8
<b>3.1. HTML</b> .....	8
<b>3.2. CSS</b> .....	9
<b>3.3. Bulma</b> .....	10
<b>3.4. Django</b> .....	10
<b>3.5. Vue.js</b> .....	12
<b>4. FUNKCIONALNOST APLIKACIJE</b> .....	14
<b>4.1. Baza podataka</b> .....	14
<b>4.2. Registracija i prijava korisnika</b> .....	14
<b>4.3. Administratorske funkcije</b> .....	20
<b>4.4. Glavna stranica</b> .....	23
<b>5. KORIŠTENJE APLIKACIJE</b> .....	28
<b>5.1. Početna stranica</b> .....	28
<b>5.2. Proizvod</b> .....	30
<b>5.3. Košarica</b> .....	31
<b>5.4. Povijest narudžbi</b> .....	34
<b>5.5. Sučelje administratora</b> .....	36
<b>6. ZAKLJUČAK</b> .....	39
<b>LITERATURA</b> .....	40
<b>SAŽETAK</b> .....	41
<b>ABSTRACT</b> .....	42
<b>ŽIVOTOPIS</b> .....	43

# 1. UVOD

U suvremenom digitalnom dobu, internetske trgovine postale su sastavni dio svakodnevice, pružajući korisnicima brz i praktičan način kupovine, neovisno o njihovoj lokaciji. Web aplikacije poput ove omogućuju jednostavan i učinkovit pregled širokog asortimana proizvoda u kratkom vremenu, eliminirajući potrebu za fizičkim posjetima trgovinama. Ova pogodnost posebno dolazi do izražaja kod vlasnika egzotičnih kućnih ljubimaca, jer proizvodi specifični za njihove potrebe često nisu dostupni u standardnim trgovinama. Zahvaljujući ovakvim aplikacijama, željeni proizvodi dostavljaju se izravno na kućnu adresu, olakšavajući brigu o njihovim ljubimcima.

Tema ovog završnog rada je razvoj specijalizirane web trgovine namijenjene vlasnicima kućnih ljubimaca, koja korisnicima pruža intuitivno iskustvo pregleda i kupnje širokog spektra proizvoda za njihove ljubimce. Aplikacija omogućava korisnicima registraciju i prijavu, otvarajući vrata jednostavnoj i sigurnoj narudžbi željenih artikala, uz mogućnost pregleda povijesti kupnji. Dodatno, korisnici imaju priliku koristiti kupone za popuste, što dodatno smanjuje cijenu proizvoda. Aplikacija je opremljena funkcionalnošću pretraživanja specifičnih proizvoda te pregledom po kategorijama, što znatno olakšava i ubrzava pronalaženje željenih artikala.

Osim korisničkih funkcionalnosti, aplikacija nudi i napredno administratorsko sučelje za upravljanje korisnicima i proizvodima, oslanjajući se na Django administracijski sustav. Administratori imaju mogućnost dodavanja novih proizvoda, uređivanja postojećih, upravljanja kuponima, te praćenja i izmjene narudžbi, uključujući promjenu statusa narudžbe (naručeno, poslano, dostavljeno). Također, mogu pregledavati narudžbe, brisati proizvode koji više nisu dostupni te ažurirati količinu trenutno dostupnih proizvoda.

Dokument je podijeljen u pet poglavlja koja prikazuju razvoj web trgovine za kućne ljubimce. Prvo poglavlje daje opći uvod u temu i objašnjava problematiku koju ovaj rad adresira. Drugo poglavlje analizira postojeće web stranice slične funkcionalnosti te ih uspoređuje s ovom aplikacijom. Treće poglavlje posvećeno je tehnologijama korištenim u izradi web aplikacija. Četvrto poglavlje opisuje glavne funkcionalnosti web trgovine, dok peto poglavlje prikazuje praktičnu demonstraciju upotrebe same aplikacije.

## **1.1. Zadatak završnog rada**

Web aplikacija omogućuje vođenje trgovine za kućne ljubimce, s opcijom kupnje kao prijavljeni korisnik ili gost, pri čemu su potrebni korisnički podaci. Administrator upravlja proizvodima, analitikom i kuponima. Registrirani korisnici imaju dodatne pogodnosti, uključujući pregled povijesti kupnji.

## **2. POSTOJEĆE WEB STRANICE**

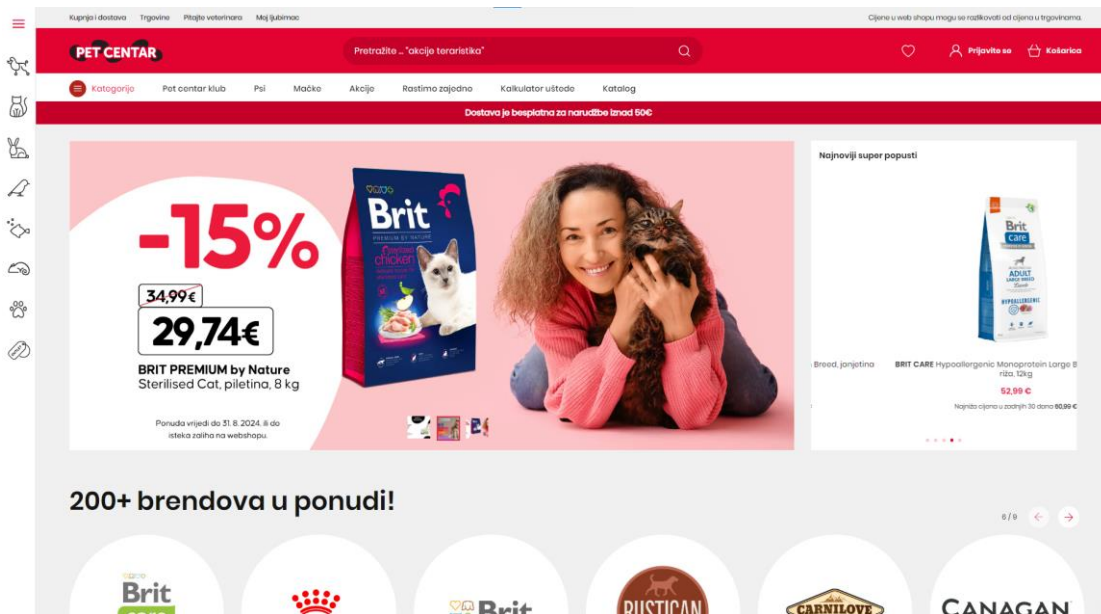
U ovom poglavlju prikazane su postojeće web stranice za prodaju proizvoda namijenjenim kućnim ljubimcima. Neki od primjera trgovina specijaliziranih za ovu vrstu asortimana su:

- Pet Network International d.o.o. za trgovinu i usluge – pet-centar.hr
- zoocity.hr
- pethomeshop.hr
- petsmart.com
- Šapica d.o.o.

### **2.1. Pet Network International d.o.o. za trgovinu i usluge – pet-centar.hr**

Pet centar je osnovan 1999. godine s ciljem postati vodeće mjesto za nabavku asortimana za kućne ljubimce u Hrvatskoj. Danas zapošljava oko 250 djelatnika i upravlja s 20 trgovina diljem zemlje, uključujući Zagreb, Rijeku, Split i Dubrovnik. Razvili su vlastite robne marke hrane za pse i mačke te unutar trgovina nude veterinarske ljekarne s dostupnim stručnim savjetima.

Na početnoj stranici prikazane su informacije o trenutnim popustima i najtraženijim proizvodima. Izbornik na lijevoj strani omogućuje pretraživanje po kategorijama, dok je traka za pretraživanje specifičnih proizvoda smještena na vrhu stranice. Kontakt informacije nalaze se na dnu stranice. Početna stranica pet centra prikazana je na slici 2.1. [1]



Slika 2.1. Isječak početne stranice pet-centar.hr.

## 2.2. Zoocity.hr

ZOOCITY je vodeća mreža specijaliziranih trgovina za kućne ljubimce u Hrvatskoj, s 50 lokacija diljem zemlje i bogatim asortimanom koji obuhvaća više od 6000 artikala. Njihov *webshop* nudi širok izbor hrane, opreme i potrepština iz preko 140 renomiranih brandova. ZOOCITY se ističe ne samo kvalitetom proizvoda nego i brigom za ljubimce, nudeći mogućnost povrata hrane te aktivno surađujući s udrugama i azilima kroz donacijske programe.

*Web* stranica omogućuje jednostavnu komunikaciju s korisnicima pomoću kontakt obrasca na početnoj stranici u donjem desnom kutu. Stranica nudi intuitivan odabir kategorija proizvoda i traku za pretraživanje. Prilikom otvaranja stranice predlaže se digitalni katalog i prikaz novih proizvoda u ponudi. Prikaz početne stranice zoocity.hr vidljiv je na slici 2.2 [2].

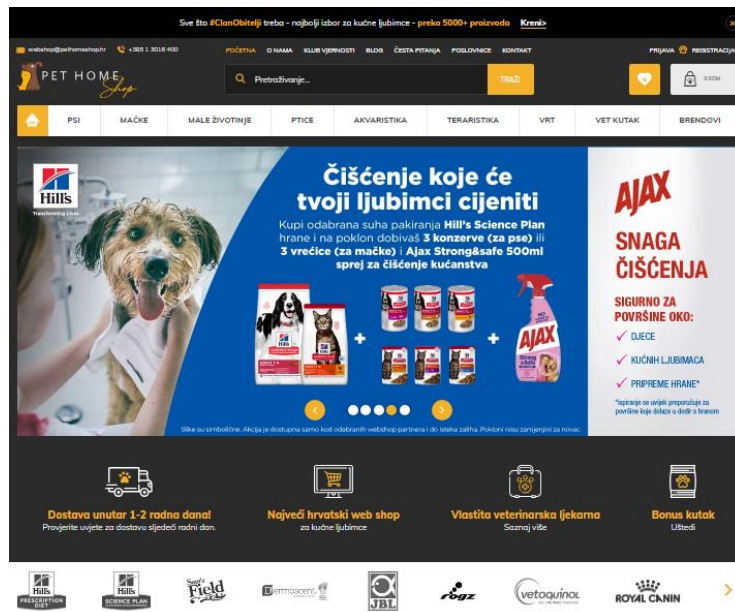




Slika 2.2. Isječak početne stranice zoocity.hr.

### 2.3. Pet Home Shop

Pet Home Shop započeo je s radom kao web trgovina 2020. godine i od tada svake godine osvaja „Web Retailer Award“ u različitim kategorijama. Njihova web aplikacija organizirana je tako da početna stranica odmah ističe klub vjernosti, mjesečni katalog i nove proizvode u ponudi. Navigacija kroz stranicu omogućena je pomoću trake s kategorijama ili trake za pretraživanje, olakšavajući korisnicima brzo pronalaženje željenih proizvoda. Prikaz početne stranice Pet Home Shop-a možemo vidjeti na slici 2.3. [3].

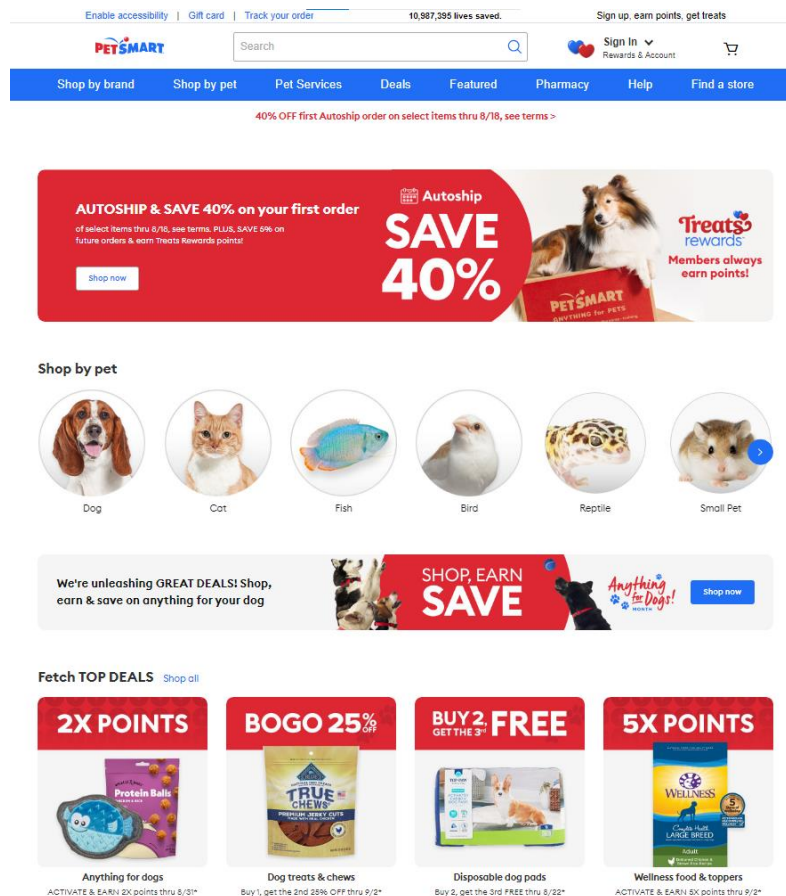


Slika 2.3. Isječak početne stranice Pet Home Shop.

## 2.4. PetSmart

PetSmart najveći je specijalizirani trgovac na malo u Sjevernoj Americi, s više od 1660 trgovina i 7 distributivnih centara. U radu s udrugama u Americi i Kanadi pomogli su s više od 10 milijuna udomljenih kućnih ljubimaca.

Pet Smart *web* stranica nudi prikaz trenutno najboljih ponuda, praćenje narudžbi, kupovanje poklon kartica, pretraživanje po kategorijama ili pomoću trake za pretraživanje. Slika 2.4. prikazuje početnu stranicu Pet Smart-a [4].



**Slika 2.4.** Isječak početne stranice Pet Smart.

## 2.5. Šapica d.o.o.

Šapica d.o.o. je profesionalni uslužni centar za kućne ljubimce iz Osijeka, osnovan 2005. godine, koji nudi usluge šišanja i njege kućnih ljubimaca, trgovinu hranom te savjetovalište.

Web stranica Šapice pruža informacije o plaćanju, dostavi, općim uvjetima i kontaktima putem trake na vrhu stranice. Također je omogućeno pretraživanje po kategorijama vrsta kućnih ljubimaca jednostavnim klikom na odgovarajuću sliku. Početna stranica Šapice prikazana je na slici 2.5 [5].



**Slika 2.5.** *Isječak početne stranice trgovine Šapica.*

### 3. TEHNOLOGIJE KORIŠTENE ZA IZRADU APLIKACIJE

U ovom poglavlju objašnjene su tehnologije i alati korišteni za izradu ovog završnog rada.

#### 3.1. HTML

HTML (engl. *Hypertext Markup Language*) jezik je koji se koristi za stvaranje web stranica. Kombinacija je hiperteksta koji definira vezu između web stranica i *Markup* jezika kojim se definiraju tekstualni dokumenti unutar oznaka za strukturiranje [6]. Logo HTML 5 prikazan je na slici 3.1.1.



Slika 3.1.1. Logo HTML-a 5.

Slikom 3.1.2. je prikazan jedan od načina korištenja oznaka pisanih HTML 5 jezikom.

```
24 <div id="main-navbar" class="navbar-menu" :class="{ 'is-active' : menuClass}">
25   <div class="navbar-start">
26     <div class="navbar-item">
27       <form method="get" action="{% url 'search' %}">
28         <div class="field has-addons">
29           <div class="control">
30             <input type="text" class="input" placeholder="Search..." name="query">
31           </div>
32           <div class="control">
33             <button class="button is-success">Search</button>
34           </div>
35         </div>
36       </form>
37     </div>
38 </div>
```

Slika 3.1.2. Primjer oznaka i njihovih akcija u HTML-u.

## 3.2. CSS

CSS (engl. *Cascading Style Sheets*) je jednostavan stilizacijski jezik osmišljen kako bi olakšao proces izrade vizualnog prikaza web stranica. Njegova glavna svrha je razdvajanje sadržaja dokumenta od prezentacije, uključujući elemente poput fontova, izgleda i boja. Ova podjela omogućava veću čitkost koda, fleksibilnost u programiranju i bolju pristupačnost. CSS je neizostavan za dizajn web stranica jer olakšava održavanje i ponovnu upotrebu koda te omogućava prilagodljiv dizajn usklađen s različitim veličinama ekrana [7].

Pomoću CSS-a moguće je stvoriti konzistentan, vizualno privlačan izgled, te dodavati animacije i prijelaze. Na taj način CSS značajno doprinosi poboljšanju korisničkog doživljaja i dostupnosti. Logo CSS-a prikazan je na slici 3.2.1



**Slika 3.2.1.** Slika loga CSS-a.

Na slici 3.2.2. možemo vidjeti jedan od načina upotrebe CSS jezika.

```
1  form .helptext {
2      font-size: 12px !important;
3      color: #ccc !important;
4  }
5
6  form ul {
7      display: none;
8  }
```

**Slika 3.2.2.** Prikaz oznaka u CSS-u.

### 3.3. Bulma

Bulma je moderni CSS okvir za stiliziranje koji web programeri koriste za jednostavno kreiranje responzivnih i estetski privlačnih *web* stranica. Zasnovan je na Flexboxu, što omogućuje fleksibilan i učinkovit način izrade složenih rasporeda. Bulma pruža unaprijed definirane klase za tipografiju, raspored, forme i različite komponente poput navigacijskih traka, kartica i modala, što ubrzava razvoj i smanjuje potrebu za prilagođenim CSS kodom[8].

Bulma se koristi zajedno s HTML-om i omogućuje web stranicama da budu prilagođene različitim veličinama ekrana, što je ključno u razvoju modernih web aplikacija. Korištenjem Bulme, programeri mogu kreirati kompleksne korisničke sučelja uz minimalan napor, dok istovremeno održavaju čist i održiv kôd.

Slika 3.3.1. prikazuje dio koda u kojem se koristi Bulmin *grid* sustav, implementiran kroz klase „*columns*“ i „*column*“, za organizaciju elemenata u responzivne stupce.

```
68     <footer class="footer">
69         <div class="columns">
70             <div class="column is-4">
71                 <h2 class="subtitle">Pet Store</h2>
72             </div>
73
74             <div class="column is-4">
75                 <a href="{% url 'contact' %}" class="navbar-item">Contact</a>
76                 <a href="{% url 'about' %}" class="navbar-item">About</a>
77             </div>
78         </div>
79
80     </div>
81
82 </div>
83 </footer>
```

Slika 3.3.1. Primjer upotrebe Bulma grid sustava u HTML kodu.

### 3.4. Django

Django je razvojni okvir izveden u programskom jeziku Python koji pruža brz i učinkovit razvoj pouzdanih i fleksibilnih web aplikacija. Za razliku od klijent-*side* tehnologija poput JavaScript-a, Django obrađuje zahtjeve na poslužitelju, generirajući dinamički sadržaj koji se šalje klijentu u obliku HTML-a. Django posjeduje mnoštvo ugrađenih značajki,

uključujući autentifikaciju korisnika, ORM (*Object-Relational Mapping*) za upravljanje bazama podataka, i napredno administracijsko sučelje.

Django je dizajniran s naglaskom na sigurnost i jednostavnost korištenja, čime olakšava zaštitu od uobičajenih sigurnosnih prijetnji poput SQL injekcija i XSS (engl. *Cross-Site Scripting*). Također, pruža alate za razvoj skalabilnih web aplikacija, omogućujući programerima da se fokusiraju na logiku aplikacije dok razvojni okvir brine o infrastrukturnim aspektima. Na slici 3.4.1. prikazan je logo Django razvojnog okvira [9].

The image shows the Django logo, which consists of the word "django" in a lowercase, bold, sans-serif font. The letter 'j' is stylized with a dot that is slightly offset to the right.

**Slika 3.4.1.** Logo Django-a.

Slikom 3.4.2. prikazuje Django model „*OrderItem*“ koji koristi „*ForeignKey*“ odnose za povezivanje narudžbi i proizvoda.

```
42 class OrderItem(models.Model):
43     order= models.ForeignKey(Order, related_name='items', on_delete=models.CASCADE)
44     product = models.ForeignKey(Product, related_name='items', on_delete=models.DO_NOTHING)
45     price= models.FloatField()
46     quantity = models.IntegerField(default=1)
47
48     def __str__(self):
49         return '%s' % self.id
```

**Slika 3.4.2.** Primjer Django modela.

Slika 3.4.3. prikazuje inicijalizaciju klase „*Cart*“ dohvaćanjem korisničke sesije pomoću „*request.session*“, što je Django specifična metoda za rad s podacima pohranjenim na strani poslužitelja tijekom korisničke sesije. Ako košarica ne postoji, kreira se kao prazan rječnik i pohranjuje u sesiji, što omogućuje trajno čuvanje podataka o košarici između korisničkih zahtjeva.



```

4 class Cart(object):
5     def __init__(self, request):
6         self.session = request.session
7         cart = self.session.get(settings.CART_SESSION_ID)
8
9         if not cart:
10            cart = self.session[settings.CART_SESSION_ID] = {}
11
12        self.cart = cart
13

```

**Slika 3.4.3.** Prikaz inicijalizacije klase „Cart“.

### 3.5. Vue.js

Vue.js je napredni JavaScript razvojni okvir koji se rabi za izgradnju dinamičnih korisničkih sučelja. Vue je poznat po svojoj prilagodljivosti i jednostavnosti učenja, što ga čini idealnim za projekte svih veličina, od manjih komponenti do kompleksnih *frontend* aplikacija. Slika 3.5.1. prikazuje logo Vue.js razvojnog okvira [10].



**Slika 3.5.1.** Logo Vue.js-a.

Vue.js omogućuje deklarativno povezivanje podataka s HTML elementima kroz reaktivni sustav podataka, što znači da se promjene podataka automatski odražavaju na korisničkom sučelju. Osim toga, Vue podržava modularnu arhitekturu aplikacija pomoću komponentata, koje se lako mogu kombinirati i ponovno koristiti. Ova fleksibilnost omogućava laku izgradnju dinamičkih i responzivnih *web* aplikacija.

Vue.js se često koristi zajedno s Django kao bi se stvorila snažna kombinacija za razvoj *web* aplikacija. Dok Django osigurava robusnu *backend* infrastrukturu i upravljanje podacima, Vue.js preuzima ulogu na *frontend* strani, pružajući dinamično i interaktivno korisničko sučelje. Ova kombinacija omogućuje gradnju modernih aplikacija gdje se Django koristi za obradu podataka i API-je, a Vue.js za stvaranje reaktivnih korisničkih sučelja.

Slika 3.5.2. prikazuje inicijalizaciju Vue.js instance unutar Django aplikacije. Vue.js se koristi za upravljanje interaktivnim dijelom korisničkog sučelja, posebno košaricom za kupovinu. Definirana su podatkovna polja koja upravljaju stanjem aplikacije, te se koriste izračunata svojstva za automatsko izračunavanje broja artikala i ukupnih troškova, uključujući popuste, što omogućuje dinamično ažuriranje korisničkog sučelja u stvarnom vremenu.

```
132 {% block scripts %}
133 <script type="application/javascript" src="https://js.stripe.com/v3/"></script>
134 <script>
135     var productapp = new Vue({
136       el: '#cartapp',
137       delimiters: ['[[', ']]'],
138       store: store,
139       data () {
140         return {
141           errors: [],
142           first_name: '{{ first_name }}',
143           last_name: '{{ last_name }}',
144           email: '{{ email }}',
145           address: '',
146           zipcode: '',
147           place: '',
148           products: [{{ productsstring|safe }}],
149           coupon_value: 0,
150           coupon_code: ''
151         }
152       },
153       computed: {
154         numItems: function() {
155           return store.state.numItems
156         },
157         totalCost: function() {
158           return store.state.totalCost
159         },
160         totalCostWithCoupon: function() {
161           if (this.coupon_value > 0) {
162             return store.state.totalCost * (parseInt(this.coupon_value) / 100);
163           } else {
164             return store.state.totalCost;
165           }
166         }
167       }
168     });
169 }
```

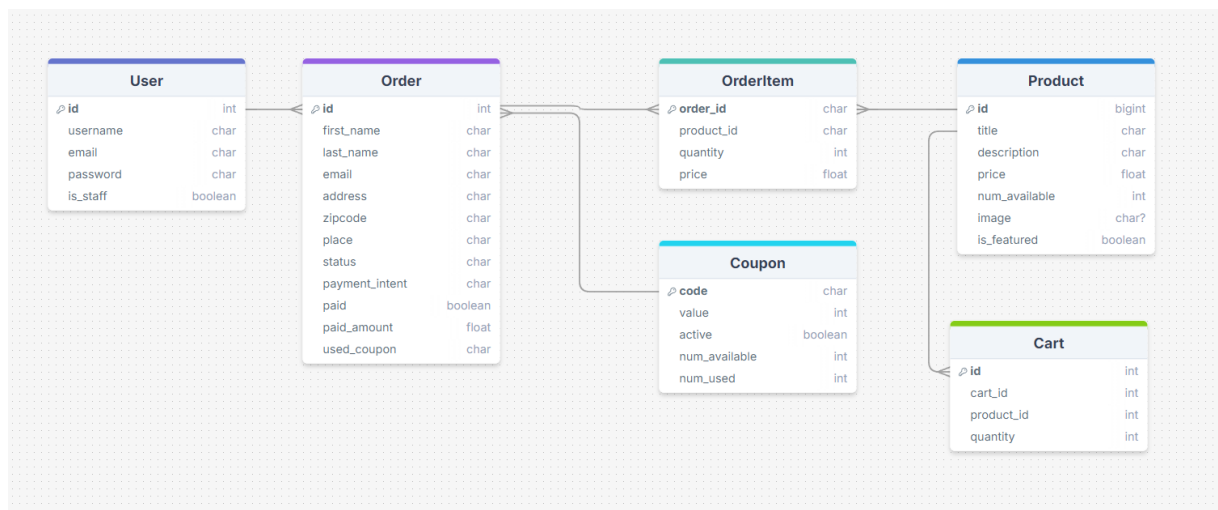
Slika 3.5.2. Primjer korištenja Vue.js za upravljanje košaricom i dinamičkim podacima .

## 4. FUNKCIONALNOST APLIKACIJE

U ovom poglavlju opisane su funkcije koje se koriste u aplikaciji web shop za ljubimce.

### 4.1. Baza podataka

Baza podataka za ovu web aplikaciju sastoji se od sedam ključnih tablica koje pohranjuju sve relevantne podatke vezane za korisnike, narudžbe, košarice i kupone. Te tablice i njihove međusobne veze prikazane su ER (engl. *Entity–relationship*) dijagramom, koji jasno ilustrira strukturu i povezanost podataka unutar aplikacije, kao što je prikazano na slici 4.1.



Slika 4.1. ER dijagram baze podataka.

### 4.2. Registracija i prijava korisnika

Korisnik ima mogućnost registracije te to može napraviti prije ili tijekom procesa naplate proizvoda. Programski kôd koji opisuje postupak registracije opisan je u ovom poglavlju.

Proces registracije uključuje unos osnovnih informacija kao što su korisničko ime, *email* adresa i lozinka. Nakon uspješne registracije, korisnici mogu koristiti te podatke za prijavu u aplikaciji.

Autentifikacija korisnika bazirana je na ugrađenom Django „*User*“ modelu, koji osigurava sigurnu pohranu lozinki pomoću *hashinga* i pruža potrebne mehanizme za provjeru vjerodostojnosti korisnika. Prijavljeni korisnici mogu pristupiti dodatnim funkcionalnostima aplikacije, kao što su pregled povijesti narudžbi, automatsko popunjavanje dijela podataka

potrebnih za izvršavanje narudžbe na temelju već unesenih informacija, i upravljanje košaricom.

Slika 4.2.1. prikazuje sve varijable koje će biti unesene u bazu podataka pri kreiranju korisničkog računa, pod uvjetom da su podaci uneseni u ispravnom formatu. Ova polja uključuju korisničko ime, ime, prezime, email adresu i zaporku. Pravila validacije osiguravaju da ime i prezime sadrže samo slova, email adresa mora sadržavati znak "@" i točku ".", dok se provjera zaporke temelji na duljini i provjeri podudarnosti dvaju unosa zaporke.

```
4
5 class SignUpForm(UserCreationForm):
6     first_name = forms.CharField(max_length=50, required=True)
7     last_name = forms.CharField(max_length=50, required=True)
8     email = forms.EmailField(max_length=255, required=True)
9
10    def __init__(self, *args, **kwargs):
11        super(SignUpForm, self).__init__(*args, **kwargs)
12
13        self.fields['username'].widget.attrs['class'] = 'input'
14        self.fields['email'].widget.attrs['class'] = 'input'
15        self.fields['password1'].widget.attrs['class'] = 'input'
16        self.fields['password2'].widget.attrs['class'] = 'input'
17        self.fields['first_name'].widget.attrs['class'] = 'input'
18        self.fields['last_name'].widget.attrs['class'] = 'input'
19
20
21    class Meta:
22        model = User
23        fields = ['username', 'first_name', 'last_name', 'email', 'password1', 'password2']
24
```

**Slika 4.2.1.** Polja za upis podataka prilikom registracije.

Na slici 4.2.2. prikazana je funkcija „*signup*“ koja omogućuje korisnicima registraciju na aplikaciju, uključujući validaciju podataka, spremanje korisnika u bazu podataka, automatsku prijavu i preusmjerenje na početnu stranicu. Funkcija „*myaccount*“ prikazuje stranicu korisničkog računa samo prijavljenim korisnicima, osiguravajući da neovlašteni korisnici nemaju pristup ovoj stranici.

```

4
5 from .forms import SignUpForm
6
7 def signup(request):
8     if request.method == 'POST':
9         form = SignUpForm(request.POST)
10
11         if form.is_valid():
12             user = form.save()
13
14             login(request, user)
15
16             return redirect('frontpage')
17
18     else:
19         form = SignUpForm()
20
21     return render(request, 'signup.html', {'form': form})
22
23 @login_required
24 def myaccount(request):
25     return render(request, 'myaccount.html')

```

**Slika 4.2.2.** Prikaz funkcija za registraciju korisničkih računa.

Prilikom prikaza detalja košarice i završetka kupnje, funkcija „*cart\_detail*“ provjera je li korisnik prijavljen. Ako jest, automatski dohvaća i popunjava polje s njegovim imenom, prezimenom i email adresom, ako korisnik nije prijavljen i ne želi se prijaviti onda ta polja ostaju prazna, te će ih korisnik morati ručno ispuniti. Ova provjera i automatsko popunjavanje vidljivo je na slici 4.2.3.

```

15
16     if request.user.is_authenticated:
17         first_name = request.user.first_name
18         last_name = request.user.last_name
19         email = request.user.email
20     else:
21         first_name = last_name = email = ''
22

```

**Slika 4.2.3.** Automatsko popunjavanje korisničkih podataka za prijavljene korisnike.

Slikom 4.2.4. prezentirano je povezivanje narudžbe s registriranim korisnikom. Ako je korisnik prijavljen, što se provjerava koristeći „*request.user.is\_authenticated*“, narudžba se povezuje s

korisničkim računom. Ovo povezivanje omogućuje povezivanje narudžbe s korisnikom koji je izvršio kupnju, što olakšava kasnije preglede povijesti narudžbi i druge korisničke funkcionalnosti. Ako korisnik nije prijavljen, narudžba će biti zabilježena bez povezivanja s korisničkim računom, ali svi podaci će biti pohranjeni kao samostalna narudžba.

```
9
10 def checkout(request, first_name, last_name, email, address, zipcode, place):
11     order = Order(first_name = first_name, last_name = last_name, email = email, address = address, zipcode = zipcode, place = place)
12
13     if request.user.is_authenticated:
14         order.user = request.user
15
16
17     order.save()
18
19     cart = Cart(request)
20
21     for item in cart:
22         OrderItem.objects.create(order=order, product=item['product'], price=item['price'], quantity=item['quantity'])
23
24     return order.id
```

**Slika 4.2.4.** Funkcija za kreiranje narudžbe i pohranu stavki iz košarice.

Obrazac za registraciju novog korisnika prikazuje se na stranici „*signup.html*“. U obrascu se koristi `{{ form.as_p }}`, što koristi Django metode za automatsko generiranje HTML-a za unos podataka u obliku paragrafskih oznaka (`<p>`). `{% csrf_token %}` uključuje CSRF token za zaštitu od napada Cross-Site Request Forgery (CSRF), a pritiskom na gumb „*Sign up*“ podaci se šalju putem POST metode na istu stranicu za obradu.

```
2
3     {% block content %}
4     <div class="columns">
5         <div class="column is-6 is-offset-3">
6             <h1 class="title">Sign up</h1>
7
8             <form method="post" action=".">
9                 {% csrf_token %}
10
11                 {{ form.as_p }}
12
13                 <div class="field">
14                     <div class="control">
15                         <button class="button is-success">Sign up</button>
16                     </div>
17                 </div>
18             </form>
19         </div>
20     </div>
21     {% endblock %}
```

**Slika 4.2.5.** Prikaz stanice sa obrascem za registraciju korisnika.

Kao što je vidljivo na slici 4.2.6. prije prijave na stanici se prikazuju gumbi „Sign up“ i „Log in“, no nakon uspješne prijave na njihovom mjestu će se nalaziti gumb „My account“.

```
47     {% if request.user.is_authenticated %}
48         <div class="navbar-item">
49             <a href="{% url 'myaccount' %}" class="button is-light">My account</a>
50         </div>
51     {% else %}
52         <div class="navbar-item">
53             <div class="buttons">
54                 <a href="{% url 'signup' %}" class="button is-primary"><strong>Sign up</strong></a>
55                 <a href="{% url 'login' %}" class="button is-light">Log in</a>
56             </div>
57         </div>
58     {% endif %}
```

**Slika 4.2.6.** Prikaz različitih opcija u navigaciji ovisno o prijavi korisnika.

Na „My account“ stranici prikazuje se gumb za odjavu, podaci o korisničkom računu (slika 4.2.7.) i podaci o korisnikovim narudžbama (slika 4.2.8.)

```
3     {% block content %}
4         <h1 class="title">My account</h1>
5
6         <div class="navbar-item">
7             <form method="post" action="{% url 'logout' %}">
8                 {% csrf_token %}
9                 <button type="submit" class="button is-danger">Log out</button>
10            </form>
11        </div>
12
13        <hr>
14
15        <h2 class="subtitle">Account information</h2>
16
17        <p><strong>Name:</strong>{{ request.user.get_full_name }}</p>
18        <p><strong>E-mail:</strong>{{ request.user.email }}</p>
19    {% endblock %}
```

**Slika 4.2.7.** Prikaz „Log out“ gumba, imena i e-maila korisnika.

```

22 <h2 class="subtitle">My orders</h2>
23
24 {% for order in request.user.orders.all %}
25   <article class="message is-light">
26     <div class="message-header">
27       <p>{{ order.created_at }}</p>
28       <p><strong>Status: </strong>{{ order.status }}</p>
29     </div>
30
31     <div class="message-body">
32       <table class="table is-striped is-fullwidth">
33         <thead>
34           <tr>
35             <th>Title</th>
36             <th>Quantity</th>
37             <th>Price</th>
38           </tr>
39         </thead>
40
41         <tbody>
42           {% for item in order.items.all %}
43             <tr>
44               <td>{{ item.product.title }}</td>
45               <td>{{ item.quantity }}</td>
46               <td>{{ item.price }}</td>
47             </tr>
48           {% endfor %}
49         </tbody>
50
51         <tfoot>
52           <tr>
53             <td>Total</td>
54             <td>{{ order.get_total_quantity }}</td>
55             <td>{{ order.paid_amount }}</td>
56           </tr>
57         </tfoot>
58       </table>

```

Slika 4.2.8. Prikaz korisnikovih narudžbi.



### 4.3. Administratorske funkcije

Prijavom kao administrator dobiva se sučelje sa različitim funkcijama za upravljanje bazom podataka. Administrator može unositi nove proizvode, dodavati kupone te pregledavati i mijenjati informacije o narudžbama. Početna stranica administrativnog sučelja organizirana je u tri glavne kategorije: kuponi, narudžbe, i trgovina, koja uključuje upravljanje proizvodima, kategorijama i dodatnim slikama proizvoda.

Slikom 4.3.1. prikazano je registriranje modela koji upravljaju dodavanjem, uređivanjem i brisanjem proizvoda. Model „*Category*“ koristi se za kategorizaciju proizvoda u aplikaciji, te sadrži polja: *title*, *slug* i *ordering*. Ovaj model vidljiv je na slici 4.3.2. i omogućuje administratorima dodavanje, uređivanje i brisanje kategorija, a polje „*ordering*“ omogućava administratorima određivanje i mijenjanje redoslijeda prikazivanja kategorija.

```
1 from django.contrib import admin
2
3 from .models import Category, Product, ProductImage
4
5 admin.site.register(Category)
6 admin.site.register(Product)
7 admin.site.register(ProductImage)
```

Slika 4.3.1. Administracija modela u admin sučelje.

```
6 class Category(models.Model):
7     title = models.CharField(max_length=225)
8     slug = models.SlugField(max_length=225)
9     ordering = models.IntegerField(default=0)
10
11     class Meta:
12         verbose_name_plural = 'Categories'
13         ordering = ('ordering',)
14
15     def __str__(self):
16         return self.title
17
```

Slika 4.3.2. Definiranje modela „*Category*“.

Model „*Product*“ sadrži attribute poput naziv proizvoda, opis, cijena, slike, a s kategorijom je povezan pomoću „*ForeignKey*“. Model koristi „*Meta*“ klasu za prikaz

proizvoda prema datumu dodavanje. Slika 4.3.3. prikazuje model „*Product*“, a na slici 4.3.4. prikazane su funkcije za generiranje sličica i spremanje proizvoda.

```
17
18 class Product(models.Model):
19     category = models.ForeignKey(Category, related_name='products', on_delete=models.CASCADE)
20     title = models.CharField(max_length=225)
21     slug = models.SlugField(max_length=225)
22     description = models.TextField(blank=True, null=True)
23     price = models.FloatField()
24     is_featured = models.BooleanField(default=False)
25     num_available = models.IntegerField(default=1)
26
27     image = models.ImageField(upload_to='uploads/', blank=True, null=True)
28     thumbnail = models.ImageField(upload_to='uploads/', blank=True, null=True)
29     date_added = models.DateTimeField(auto_now_add=True)
30
```

Slika 4.3.3. Prikaz modela „*Product*“.

```
32 class Meta:
33     ordering = ('-date_added', )
34
35 def __str__(self):
36     return self.title
37
38 def save(self, *args, **kwargs):
39     print('Save', self.image.path)
40     self.thumbnail = self.make_thumbnail(self.image)
41
42     super().save(*args, **kwargs)
43
44 def make_thumbnail(self, image, size=(300, 200)):
45     img = Image.open(image)
46     img.convert('RGB')
47     img.thumbnail(size)
48
49     thumb_io = BytesIO()
50     img.save(thumb_io, 'JPEG', quality=85)
51
52     thumbnail = File(thumb_io, name=image.name)
53
54     return thumbnail
55
```

Slika 4.3.4. Funkcija za spremanje proizvoda i dodavanje sličica - thumbnaila.

Dio koda prikazanog na slici 4.3.5. omogućuje administratorima upravljanje narudžbama. Admin ima uvid u sve izvršene narudžbe i može mijenjati status narudžbe ovisno o njenom trenutnom stanju. Konkretno, postoji mogućnost promjene statusa narudžbe u poslano - *shipped* putem akcije u administrativnom sučelju, što automatski postavlja datum slanja narudžbe i ažurira status narudžbe na *shipped*.

```

8  def order_name(obj):
9  |     return '%s %s' % (obj.first_name, obj.last_name)
10 order_name.short_description = 'None'
11
12
13 def admin_order_shipped(modeladmin, request, queryset):
14 |     for order in queryset:
15 |         order.shipped_date = datetime.datetime.now()
16 |         order.status = Order.SHIPPED
17 |         order.save()
18 |     return
19 admin_order_shipped.short_description = 'Set shipped'
20
21 class OrderItemInline(admin.TabularInline):
22 |     model = OrderItem
23 |     raw_id_fields = ['product']
24
25
26 class OrderAdmin(admin.ModelAdmin):
27 |     list_display = ['id', order_name, 'status', 'created_at']
28 |     list_filter = ['created_at', 'status']
29 |     search_fields = ['first_name', 'address']
30 |     inlines = [OrderItemInline]
31 |     actions = [admin_order_shipped]
32
33
34 admin.site.register(Order, OrderAdmin)
35 admin.site.register(OrderItem)

```

**Slika 4.3.5.** *Prikaz koda za pregled narudžbi i promjena statusa narudžbi.*

Administratori mogu kreirati i upravljati kuponima. Model „*Coupon*“, prikazan na slici 4.3.6., ima polje za kod kupona, vrijednost popusta, broj dostupnih i korištenih kupona, taktivni status kupona.

```

2
3 class Coupon(models.Model):
4     code = models.CharField(max_length=50, unique=True)
5     value = models.IntegerField()
6     active = models.BooleanField(default=True)
7     num_available = models.IntegerField(default=1)
8     num_used = models.IntegerField(default=0)
9
10    def __str__(self):
11        return self.code
12
13    def can_use(self):
14        is_active = True
15
16        if self.active == False:
17            is_active = False
18
19        if self.num_used >= self.num_available and self.num_available != 0:
20            is_active = False
21
22        return is_active
23
24    def use(self):
25        self.num_used = self.num_used + 1
26
27        if self.num_used == self.num_available:
28            self.active = False
29
30        self.save()

```

**Slika 4.3.6.** Prikazani model „Coupon“ upravlja kuponima, provjerava dostupnost i ažurira korištenje.

#### 4.4. Glavna stranica

Na naslovnoj stranici prikazani su istaknuti proizvodi (*is\_featured*) koji uključuju sliku, naziv i cijenu proizvoda, uz link „View“, vidljiv na slici 4.4.2., koji vodi na detalje proizvoda i dodavanje u košaricu.

```

5    def frontpage(request):
6        products = Product.objects.filter(is_featured = True)
7        context = {
8            'products': products
9        }
10       return render(request, 'frontpage.html', context)

```

**Slika 4.4.1.** Prikazuju se istaknuti proizvodi.

```

4
5 {% block content %}
6   <div class="columns is-multiline">
7     {% for product in products %}
8       <div class="column is-3">
9         <div class="card">
10          <div class="card-image">
11            <figure class="image">
12              
13            </figure>
14          </div>
15          <div class="card-content">
16            <div class="media">
17              <div class="media-content">
18                <p class="title is-4">{{ product.title }}</p>
19                <p class="subtitle is-6">€{{ product.price }}</p>
20              </div>
21            </div>
22          </div>
23          <footer class="card-footer">
24            <a href = "{% url 'product_detail' product.category.slug product.slug %}" class="card-footer-item">View</a>
25          </footer>
26        </div>
27      </div>
28    {% endfor %}
29  </div>
30 {% endblock %}
31

```

**Slika 4.4.2.** HTML kôd prikazuje proizvode s naslovom, cijenom, slikom i detaljima.

Stranica također sadrži traku za pretraživanje, kategorije proizvoda, gumb za pristup košarici te link na „My account“ stranicu ili stranice za registraciju. U *footeru* stranice nalaze se linkovi za „Contact“ i „About“ stranice.

Slika 4.4.3. prikazuje rješenje pretraživanja proizvoda. Funkcija prima GET zahtjev, pretražuje proizvode prema naslovu ili opisu koji sadrži taj upit, i vraća rezultate na stranicu *search.html* koristeći rezultate pretraženih proizvoda.

```

7 def search(request):
8     query = request.GET.get('query')
9     products = Product.objects.filter(Q(title__icontains=query) | Q(description__icontains=query))
10
11     context = {
12         'query': query,
13         'products': products
14     }
15
16     return render(request, 'search.html', context)
17

```

**Slika 4.4.3.** Funkcija za pretraživanje proizvoda.

Proces dodavanja proizvoda u košaricu opisan je slikama 4.4.4. i 4.4.5. Slika 4.4.4. prikazuje funkcionalnost unutar Vue.js metode, pilikom klika na gumb „Add to cart“, šalje POST zahtjev na „/api/add\_to\_cart/“ sa specifičnim podacima o proizvodu. Nakon uspješne obrade, prikazuje se poruka o uspjehu.

```

59     methods:{
60         changeMainImage(image) {
61             this.mainImage=image;
62         },
63         addToCart(product_id){
64             console.log('Product_id:', product_id);
65
66             var data = {
67                 'product_id': product_id,
68                 'update': false,
69                 'quantity': 1
70             };
71
72             fetch('/api/add_to_cart',{
73                 method:'POST',
74                 headers:{
75                     'Content-Type': 'application/json',
76                     'X-CSRFToken': '{{ csrf_token }}'
77                 },
78                 credentials:'same-origin',
79                 body: JSON.stringify(data)
80             })
81             .then((response) => {
82                 console.log(response)
83
84                 this.showMessage = true
85
86                 store.commit('increment', 1);
87
88                 setTimeout(() => {
89                     this.showMessage = false
90                 }, 2000)
91             })
92             .catch(function(error) {
93                 console.log('Error 2');
94                 console.log(error);
95             })
96         }
97     }
98 }
99
100 });

```

**Slika 4.4.4.** Vue.js metoda za dodavanje proizvoda u košaricu koristeći POST zahtjev.

Slika 4.4.5. prikazuje kako se taj zahtjev obrađuje na serverskoj strani. Funkcija „*api\_add\_to\_cart*“ prima podatke, dodaje proizvod u košaricu, i vraća JSON odgovor natrag na klijentsku stranu.

```

126 def api_add_to_cart(request):
127     data = json.loads(request.body)
128     jsonresponse = {'success': True}
129     product_id = data['product_id']
130     update = data['update']
131     quantity = data['quantity']
132     cart = Cart(request)
133     product = get_object_or_404(Product, pk=product_id)
134     if not update:
135         cart.add(product=product, quantity=1, update_quantity=False)
136     else:
137         cart.add(product=product, quantity=quantity, update_quantity=True)
138
139     return JsonResponse(jsonresponse)

```

**Slika 4.4.5.** Django metoda 'api\_add\_to\_cart' za dodavanje proizvoda u košaricu putem API zahtjeva.

U završnom procesu narudžbe stvara se sesija za košaricu i stripe, prikazano slikom 4.4.6. Proizvodi u košaricu sastavljaju se u popis proizvoda, a zatim se kreira *Stripe payment intent* i sesija za plaćanje prikazana na slici 4.4.7.

```
26 # Cart and Stripe session creation
27 cart = Cart(request)
28 stripe.api_key = settings.STRIPE_API_KEY_HIDDEN
29 items = []
30 total_price = 0.00 # Initialize total_price before the loop
31
32 for item in cart:
33     product = item['product']
34     price = int(product.price * 100)
35     if coupon_value > 0:
36         price = int(price * (1 - (coupon_value / 100)))
37
38     total_price += price * item['quantity'] / 100 # Convert back to original currency
39
40     obj = {
41         'price_data': {
42             'currency': 'usd',
43             'product_data': {
44                 'name': product.title
45             },
46             'unit_amount': price
47         },
48         'quantity': item['quantity']
49     }
50     items.append(obj)
```

Slika 4.4.6. Izvršavanje narudžbe i kreiranje Stripe sesije.

```
# Create PaymentIntent
payment_intent = stripe.PaymentIntent.create(
    amount=int(total_price * 100), # amount in cents
    currency='eur'
)

session = stripe.checkout.Session.create(
    payment_method_types=['card'],
    line_items=items,
    mode='payment',
    success_url='http://127.0.0.1:8000/cart/success/',
    cancel_url='http://127.0.0.1:8000/cart/'
)
```

Slika 4.4.7. Generiranje Stripe PaymentIntent-a i kreiranje Stripe sesije za plaćanje.

Narudžbe se stvaraju koristeći podatke iz forme i pohranjuju u bazu podataka, kao što je prikazano na slici 4.4.8.

```
66     # Create order
67     first_name = data['first_name']
68     last_name = data['last_name']
69     email = data['email']
70     address = data['address']
71     zipcode = data['zipcode']
72     place = data['place']
73
74     orderid = checkout(request, first_name, last_name, email, address, zipcode, place)
75
76     order = Order.objects.get(pk=orderid)
77     order.payment_intent = payment_intent.id # Use the payment intent ID
78     order.paid_amount = total_price
79     order.used_coupon = coupon_code
80     order.save()
```

**Slika 4.4.8.** Prikaz spremanja narudžbi u bazu podataka.

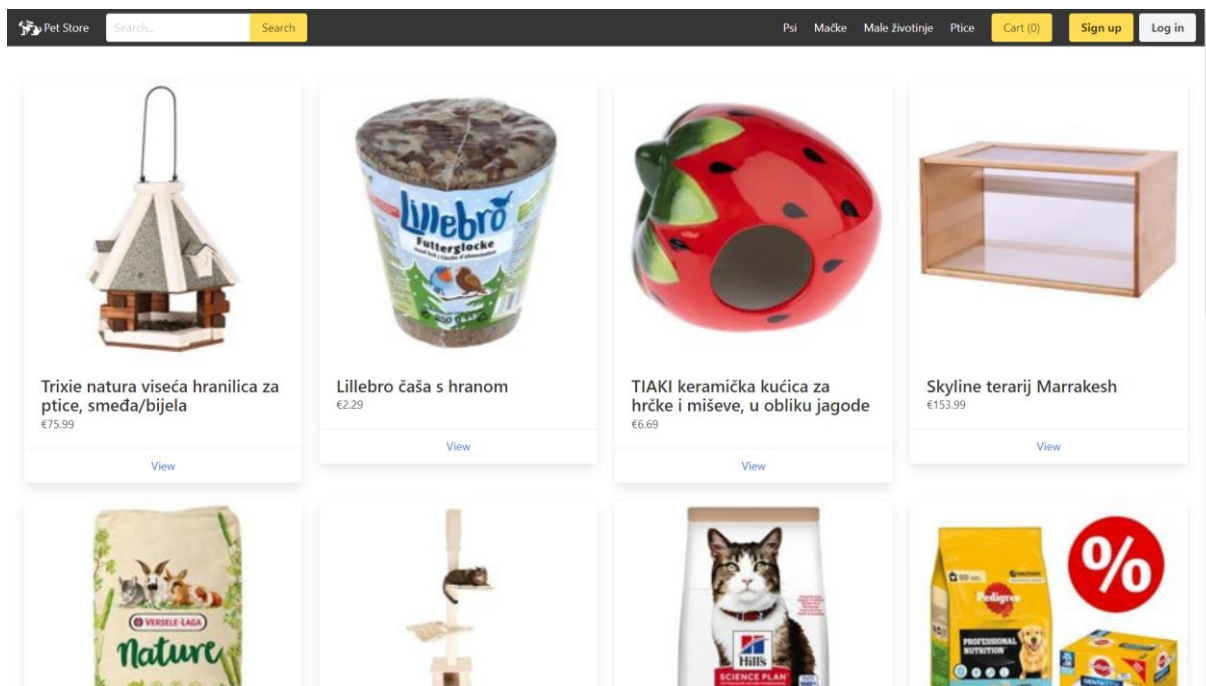


## **5. KORIŠTENJE APLIKACIJE**

Web trgovina za ljubimce sastoji se od nekoliko različitih stranica. Aplikacija može biti korištena kao administrator, neprijavljeni korisnik, ili registrirani korisnik. Prijavljeni korisnici ove aplikacije imaju mogućnost kupnje željenih proizvoda, pregledavanja povijesti prethodnih narudžbi, a dio potrebnih podataka za izvršavanje narudžbi automatski se popunjava. Administrator ima najviše ovlasti prilikom korištenja aplikacije. Na stranici s popisom korisnika mogu se vidjeti svi članovi i njihovi podaci, koje se po potrebi može uređivati ili brisati. Osim popisa korisnika, administrator može dodavati nove korisnike. Na stranici s popisom proizvoda prikazani su svi trenutno dostupni artikli, a njihovo uređivanje i brisanje također je omogućeno. Uz mogućnost uređivanja popisa proizvoda, administrator može dodavati nove proizvode, mijenjati status narudžbi, uređivati kategorije i prilagođavati redoslijed prikaza kategorija, kao i stvarati kupone. Istovremeno, ima uvid u popis svih do sada realiziranih narudžbi s podacima o kupcu i kupljenim proizvodima. Neprijavljeni korisnici mogu pregledavati proizvode, dodavati ih u košaricu i izvršiti kupnju, ali ne mogu vidjeti svoje prijašnje narudžbe.

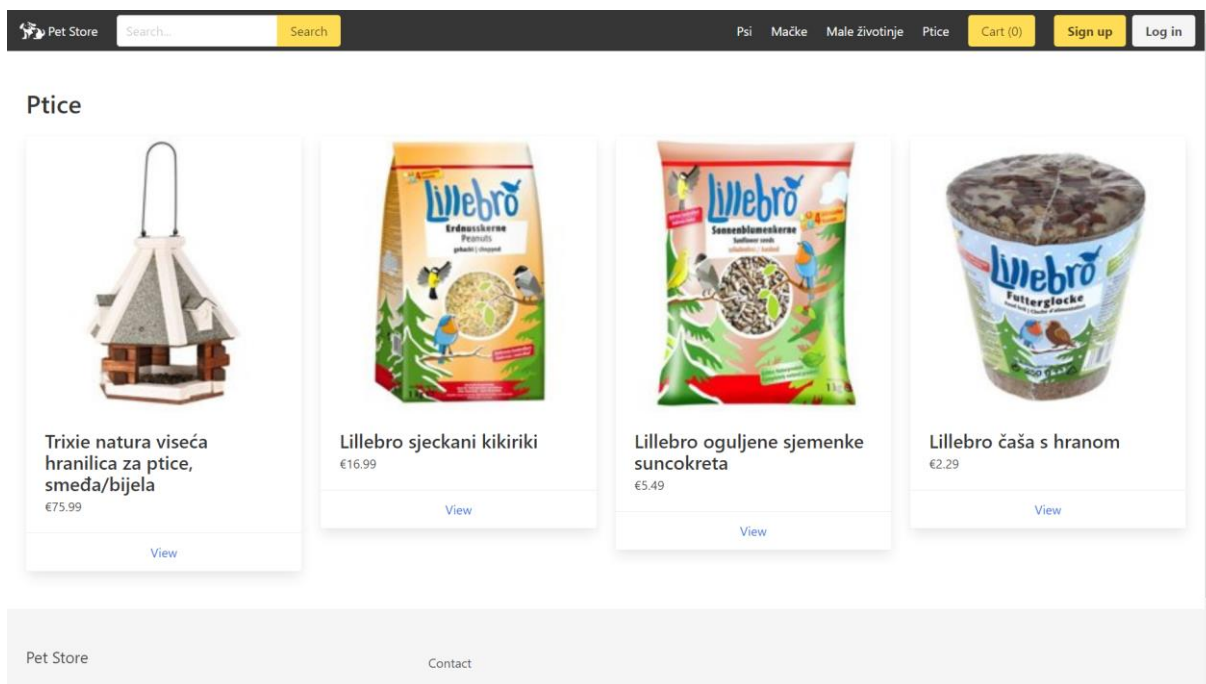
### **5.1. Početna stranica**

U gornjem dijelu početne stranice nalaze se naziv web stranice, traka za pretraživanje, gumbi za kategorije, gumb za košaricu, te gumbi za prijavu i registraciju korisnika. Navigacijska traka omogućava odabir traženog proizvoda, razvrstanog prema kategorijama, ili pregled istaknutih proizvoda na početnoj stranici. Klikom na proizvod, kupci se preusmjeravaju na stranicu s detaljima o tom proizvodu i gumbom za dodavanje u košaricu. Primjer ovog koncepta, koji obuhvaća sve navedene elemente i njihov raspored, prikazan je na slici 5.1.1.



Slika 5.1.1. Početna stranica.

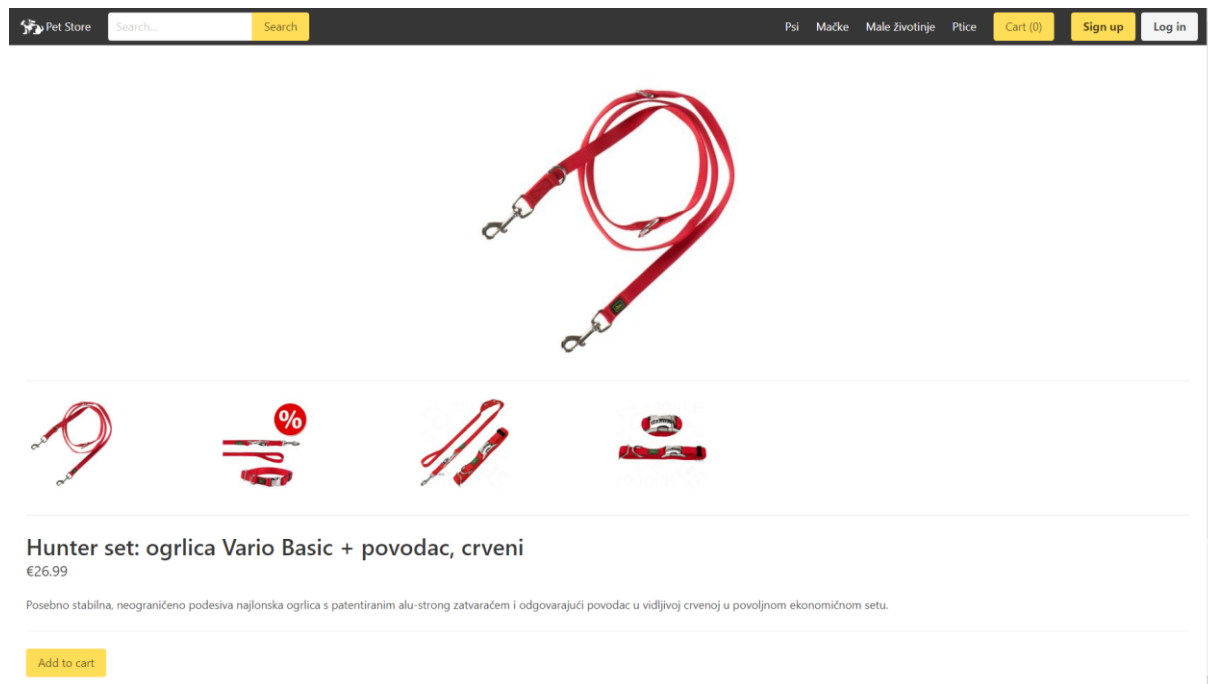
Na početnoj stranici web aplikacije klikom na jednu od kategorija prikazivat će se proizvodi vezani samo uz tu kategoriju, kao što je prikazano na slici 5.1.2.



Slika 5.1.2. Prikaz proizvoda kategorije „Ptice“.

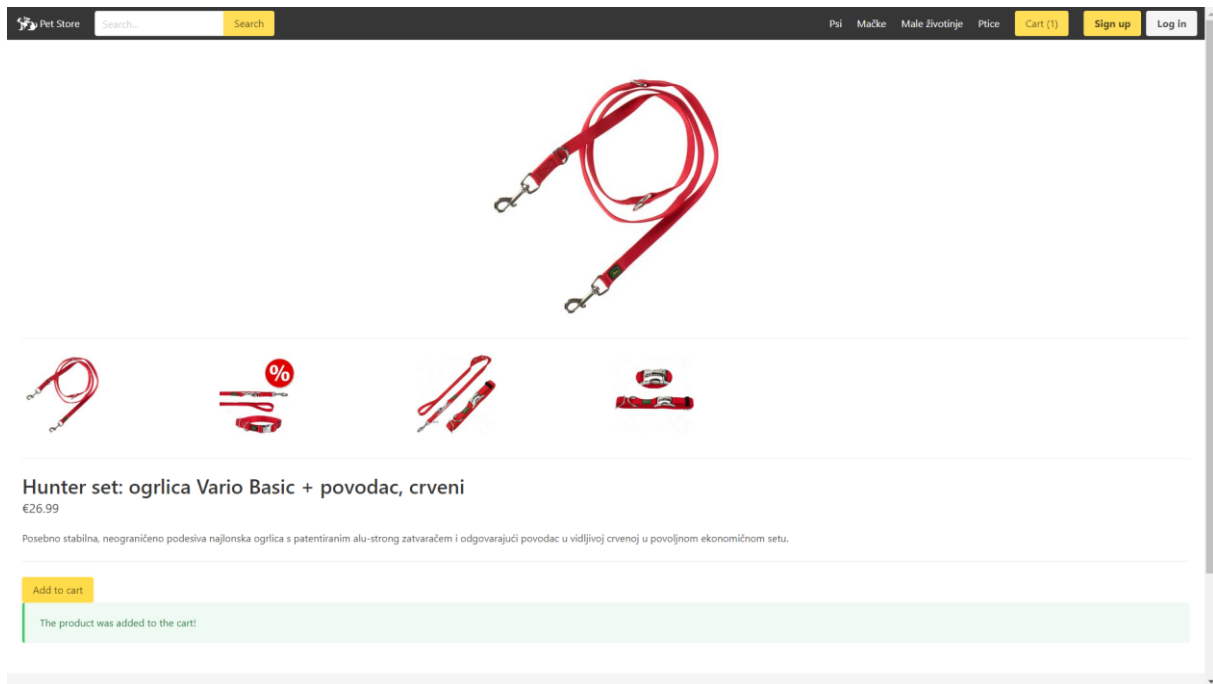
## 5.2. Proizvod

Na slici 5.2.1. prikazana je stranica odabranog proizvoda koja kupcu pruža uvid u dodatne detalja. Kupac može vidjeti i kliknuti slike vezane uz proizvod, pregled opisa samog proizvoda, cijenu, te dodavanje proizvoda u košaricu.



**Slika 5.2.1.** *Stranica pojedinog proizvoda.*

Prilikom dodavanja proizvoda u košaricu, ako je dodavanje uspješno, korisniku će se kratko prikazati poruka koja potvrđuje da je proizvod dodan u košaricu, vidljivo na slici 5.2.2.




**Slika 5.2.2.** *Uspješno dodan proizvod u košaricu.*

### 5.3. Košarica

Unošenjem proizvoda u košaricu i klikom na ikonicu košarice pokazuje se njezin trenutni sadržaj. U košarici kupac može pregledavati artikle koje je odabrao, zajedno s osnovnim informacijama o tim proizvodima, kao što su: naziv, količina, cijena svakog proizvoda te ukupni iznos. Ako se kupac odluči nastaviti s kupnjom tih artikala, mora popuniti podatke o narudžbi poput: imena, prezime, e-mail, adrese, poštanskog broja i mjesta, te kliknuti na gumb „Checkout”. Ako je kupac prijavljen, podaci za ime, prezime i e-mail automatski će se popuniti. Ako neko od polja nije ispravno popunjeno, polje će se zacrveniti kao što je prikazano na slici 5.3.1. Kupci također mogu iskoristi kupon kod za smanjivanje ukupne cijene proizvoda.

**Cart**

Product	Quantity	Price
 Hunter set: ogrlica Vario Basic + povodac, crveni	1	26.99
Total cost:		26.99
Total cost with coupon:		13.495

Coupon code:  
50code

Continue as guest or [log in / sign up.](#)

First name:  Last name:

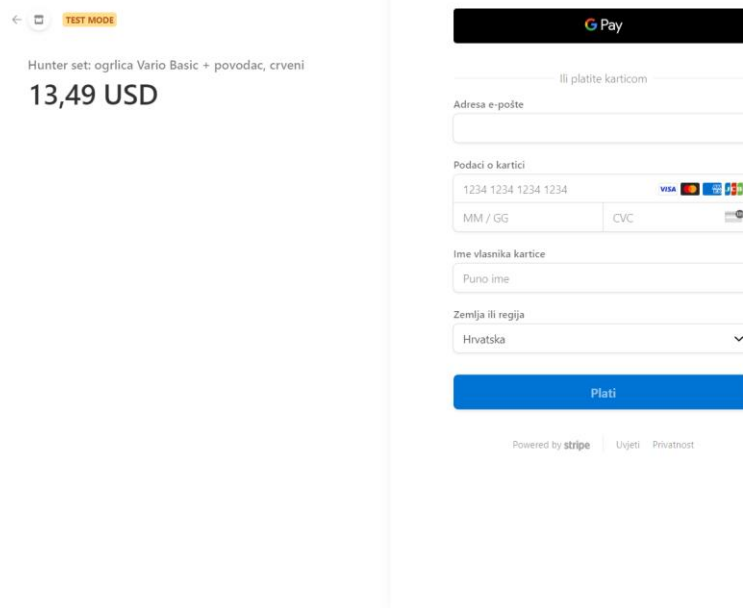
E-mail:

Address:

Zip code:  Place:

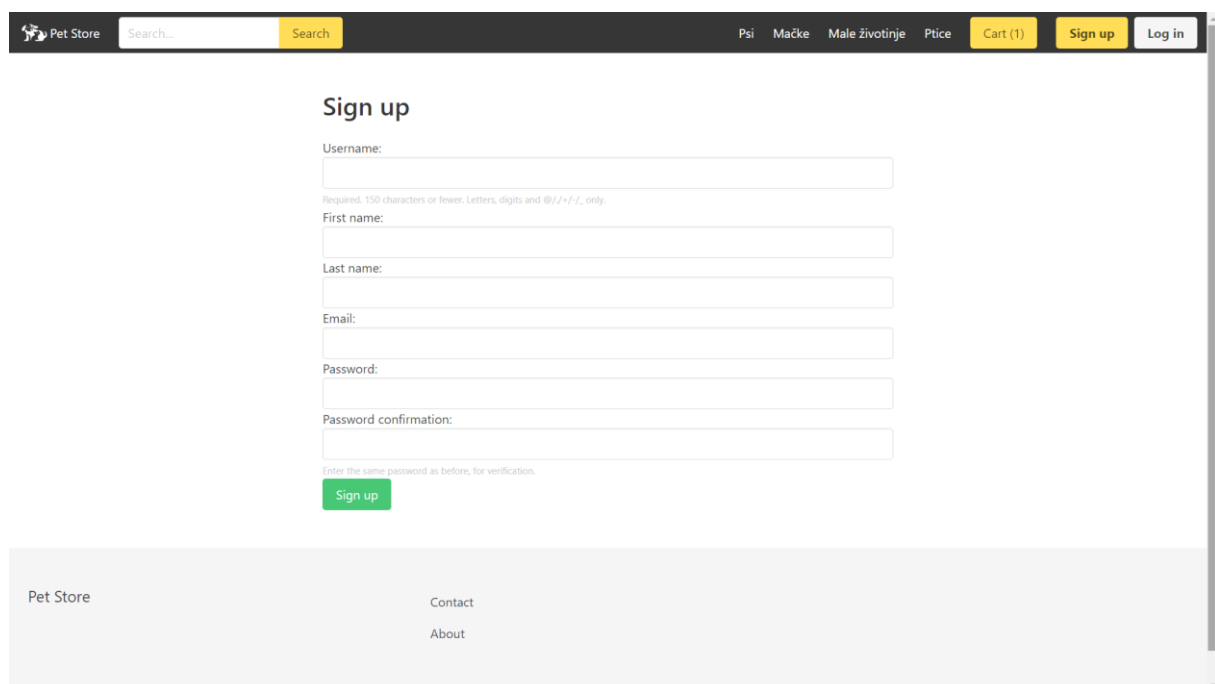
**Slika 5.3.1.** *Prikaz proizvoda u košarici u slučaju netočnog ispunjavanja i korištenje koda za popust.*

U završnom dijelu kupovine, kupac će biti preusmjeren na Stripe stranicu i popunit podatke. Od kupca se traže adresa e-pošte, podaci o kartici, ime vlasnika kartice i zemlja ili regija. Tek nakon ispunjavanja svih obaveznih polja moguće je, klikom na gumb „Plati“, završiti proces naručivanja. Narudžba se pohranjuje u bazu podataka, a kupac se preusmjerava na stanicu košarice, koja je sada ispražnjena. Slikom 5.3.2. je predstavljen završni dio kupovine.



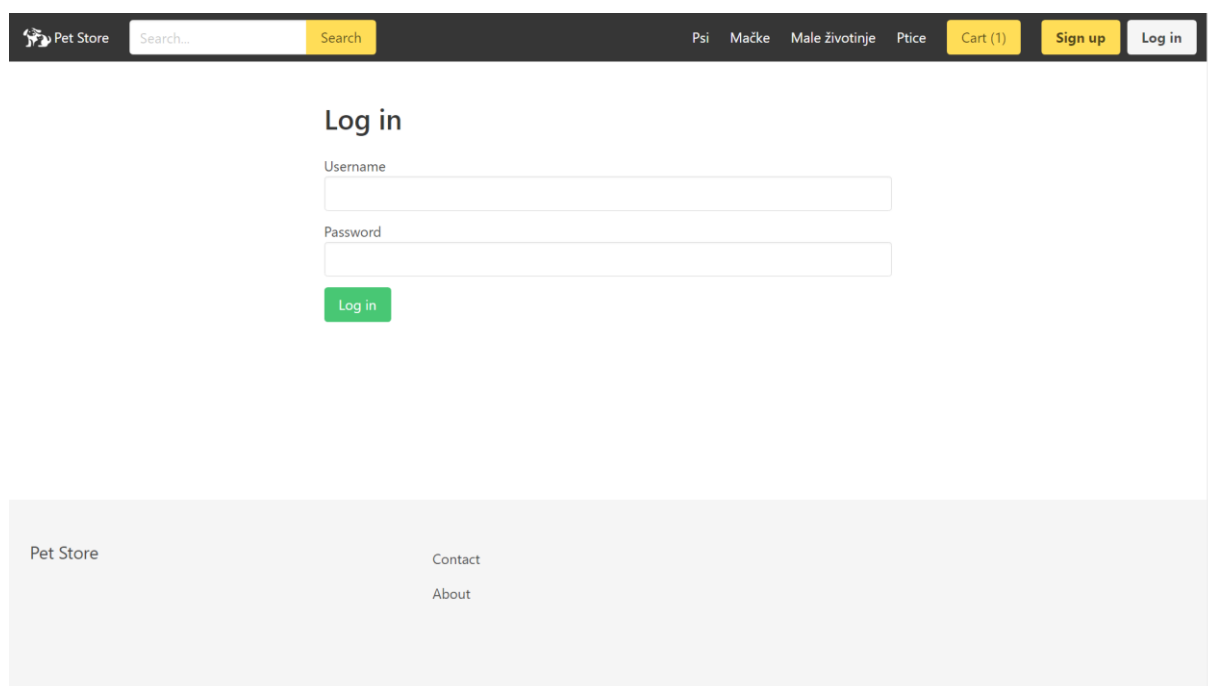
**Slika 5.3.2.** Prikaz stripe checkout stranice za konkretnu narudžbu.

Korisnik mora biti registriran u sustavu kako bi mogao pristupiti pregledavanju svojih prethodnih narudžbi. To može učiniti klikanjem gumba za registraciju „*Sign up*“. Korisnika se preusmjerava na stranicu za registraciju, gdje treba unijeti potrebne informacije. Klikanjem na gumb „*Sign up*“ uneseni podaci se pohranjuju u bazu podataka. Na slici 5.3.3. nalazi se prikaz stranice za registraciju korisnika.



**Slika 5.3.3.** Registracija.

Ako je korisnik prethodno registriran, može se prijaviti u aplikaciju, kao što je prikazano na slici 5.3.4. unosom svog korisničkog imena i lozinke. Klikom na gumb „Log in“ korisnika se preusmjerava na početnu stranicu i sada je prijava je vidljiva pojavljivanjem novog gumba „My account“ u gornjem desnom kutu.

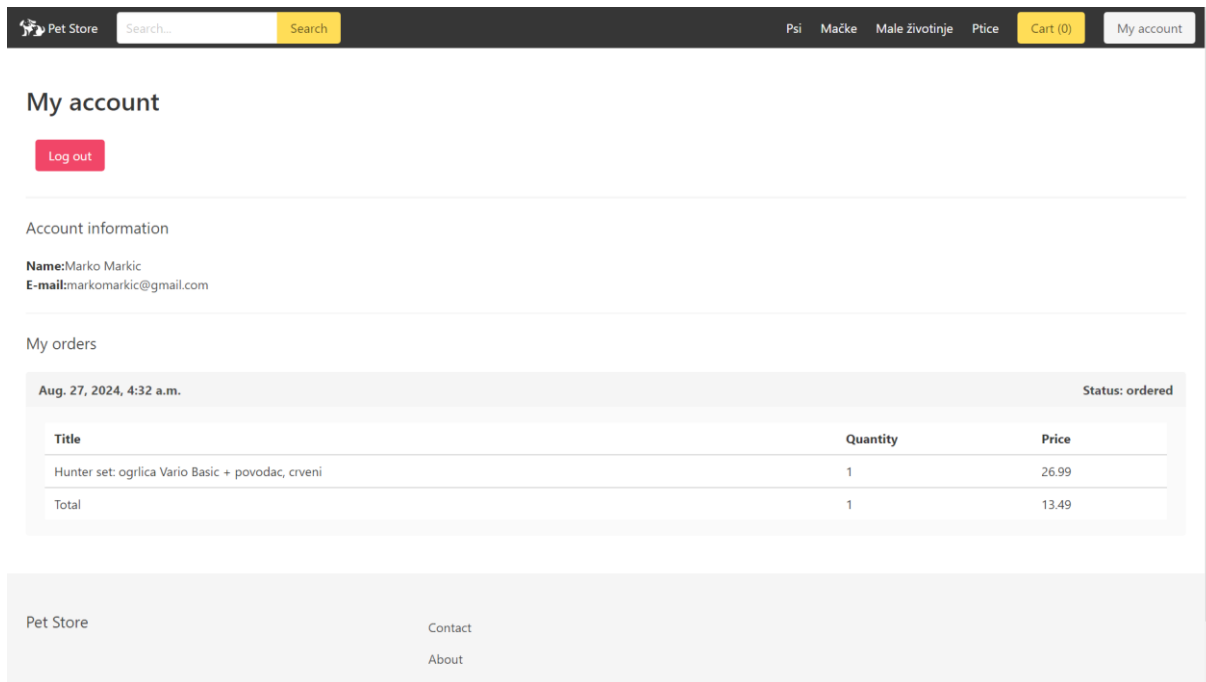


The screenshot shows a web application interface for a pet store. At the top, there is a dark navigation bar with the 'Pet Store' logo on the left, a search bar with a 'Search' button, and several menu items: 'Psi', 'Mačke', 'Male životinje', and 'Ptice'. On the right side of the navigation bar, there are three buttons: 'Cart (1)', 'Sign up', and 'Log in'. Below the navigation bar, the main content area is titled 'Log in'. It features two text input fields, one for 'Username' and one for 'Password', both with placeholder text. Below these fields is a green button labeled 'Log in'. At the bottom of the page, there is a light gray footer containing the text 'Pet Store' on the left and 'Contact' and 'About' on the right.

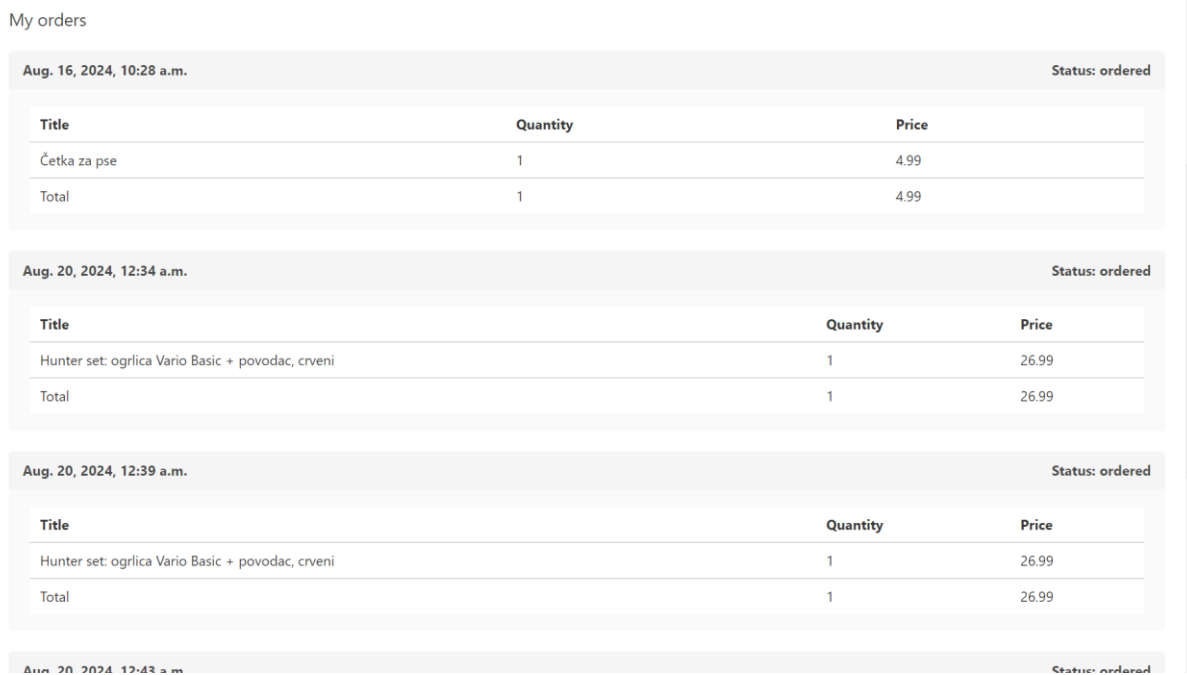
**Slika 5.3.4. Prijava.**

## 5.4. Povijest narudžbi

Ako je korisnik nekada ranije izvršio narudžbu, može je pregledati na stanici svog profila. Slikom 5.4.1. i 5.4.2. je prikazana stranica korisnika, njegovih osnovnih podataka i prijašnjih narudžbi te trenutno stanje tih narudžbi.



**Slika 5.4.1.** Pregled korisničkog računa.



**Slika 5.4.2.** Pregled prijašnji narudžbi računa i njihov trenutni status.



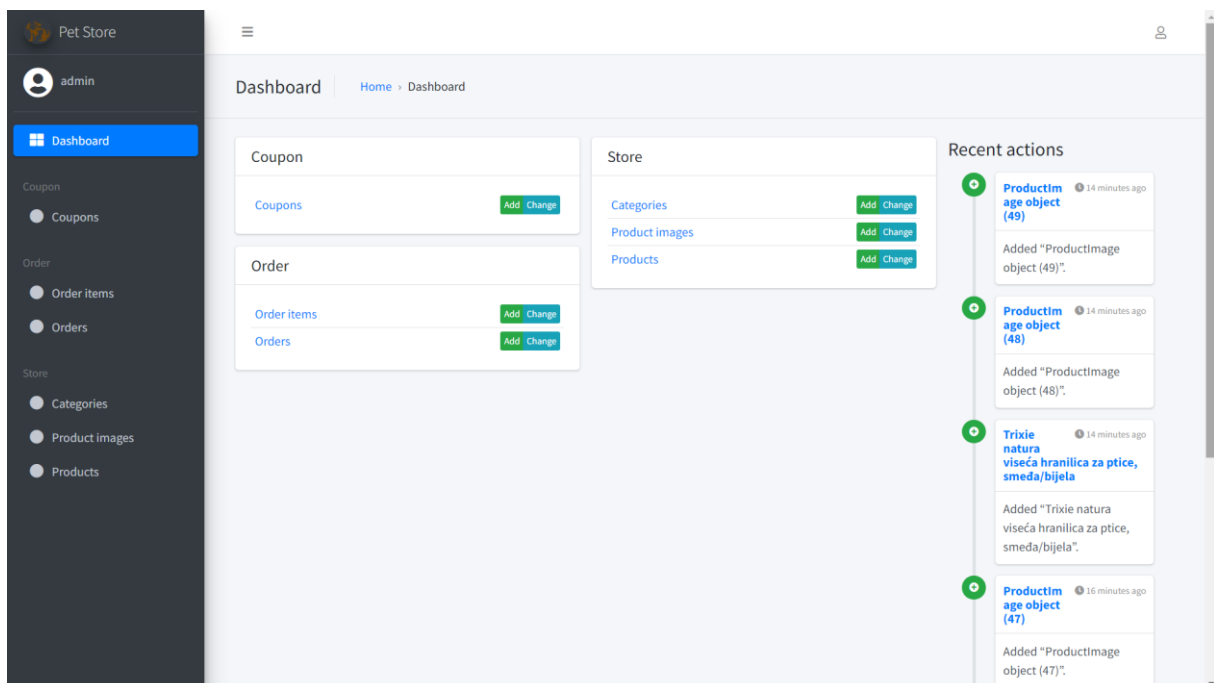
## 5.5. Sučelje administratora

Ako prijavljeni korisnik ima ulogu admina na navigacijskog traci će se stvoriti gumb na navigacijskog traci koji nas vodi na administratorsko sučelje.



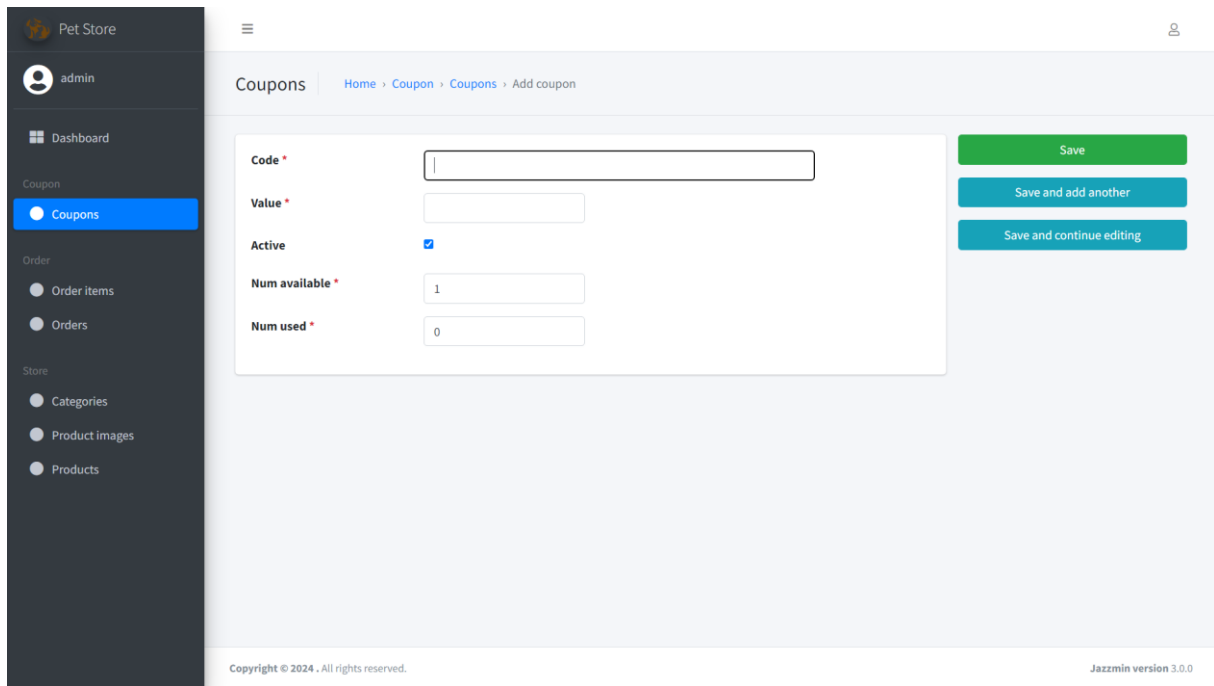
**Slika 5.5.1.** Navigacijska traka sa gumbom za administratorsko sučelje.

Administratorsko sučelje, prikazano na slici 5.5.2., podijeljeno je u tri grupe: kuponi, narudžbe i trgovina, te prikazuje kratki opis svih nedavnih radnji.



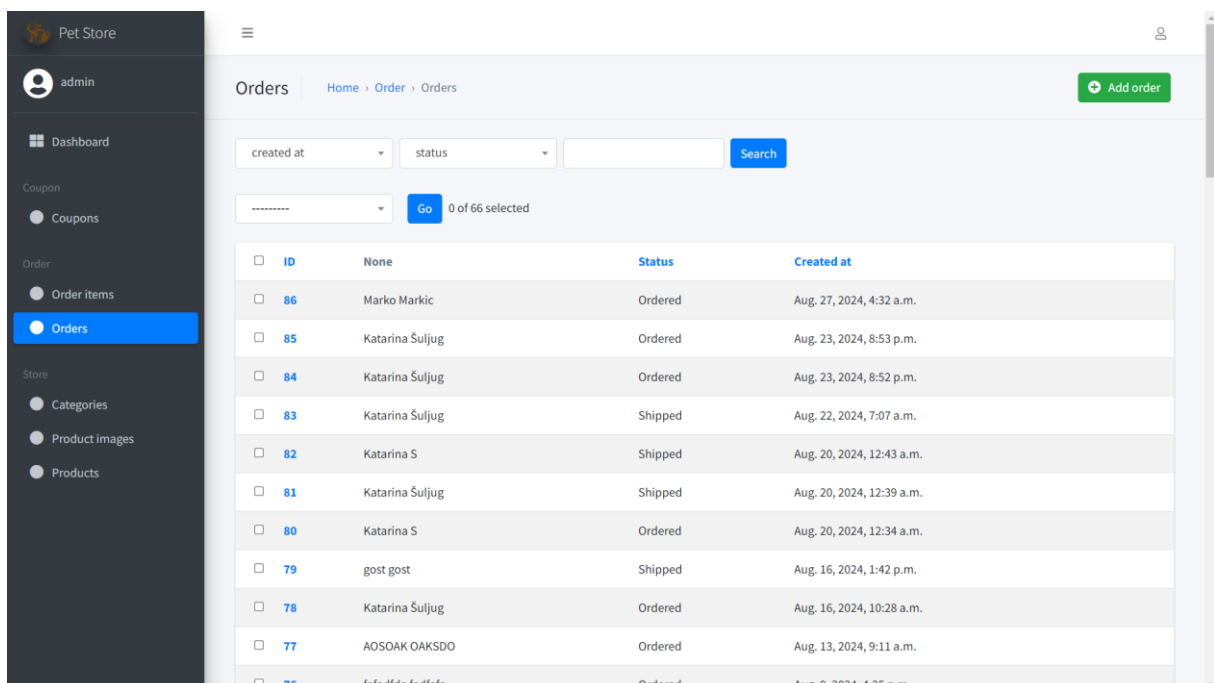
**Slika 5.5.2.** Prikaz početne stranice administratorskog sučelja.

Klikom na „*Coupon*”, administratoru je vidljiv popis svih trenutnih kupona, te mogućnost stvaranja novih. Sučelje za dodavanje novih kupona vidljivo je na slici 5.5.3.



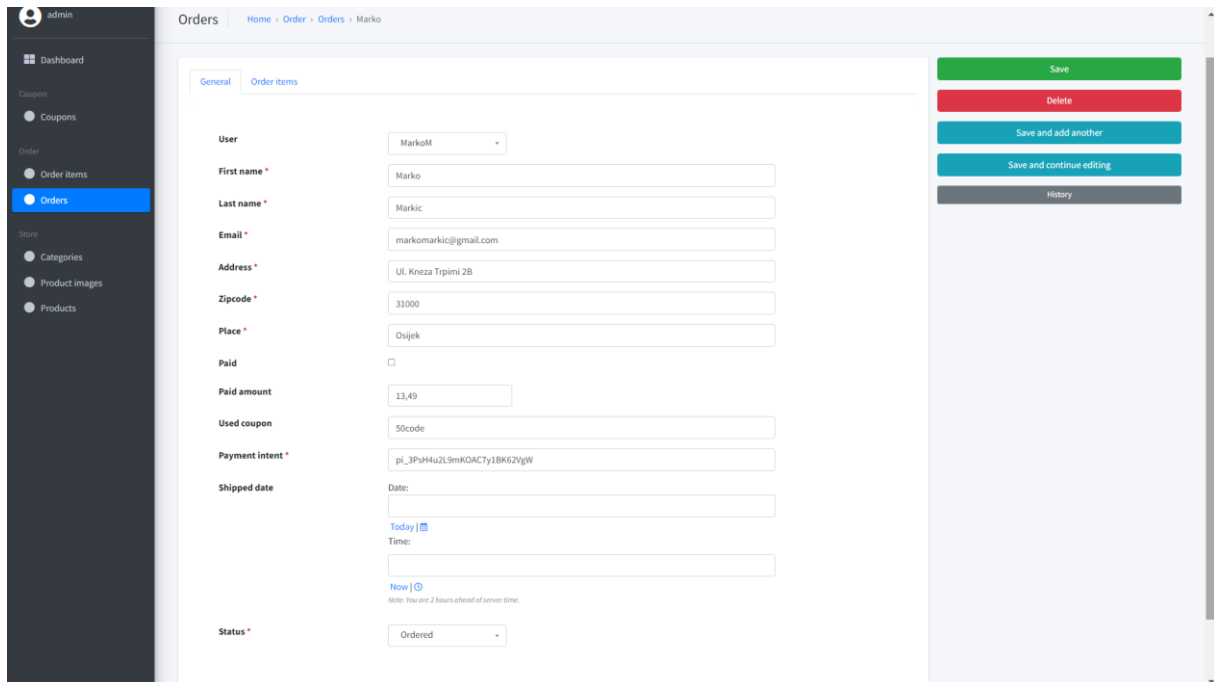
**Slika 5.5.3.** Prikaz stranice za dodavanje kupona u administratorskom sučelju.

Administratorima je također vidljiva lista svih narudžbi, ime i prezime kupca, trenutni status i vrijeme kreiranja narudžbe, ovaj prikaz vidljiv je na slici 5.5.4.



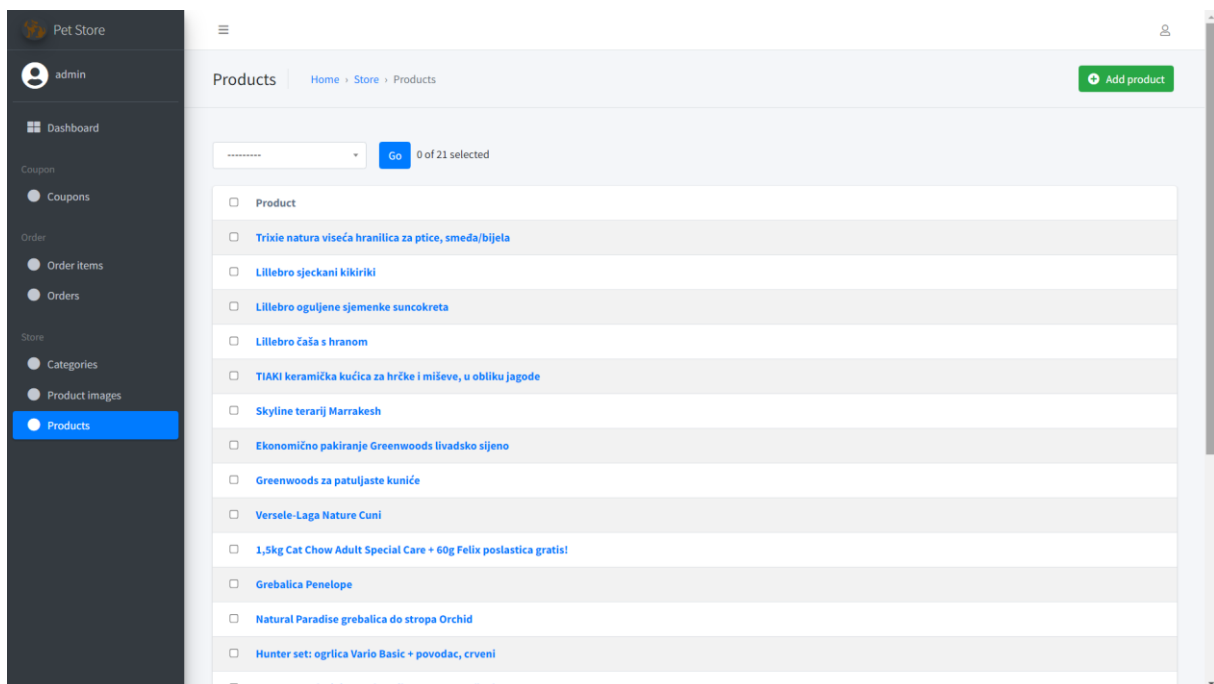
**Slika 5.5.4.** Prikaz liste narudžbe u administratorskom sučelju.

Klikom na specifičnu narudžbu prikazuju se detalji te narudžbe te polja za izmjenu trenutnog statusa narudžbe, koji su proizvodi u narudžbi, podaci o kupcu i korišteni kuponi.



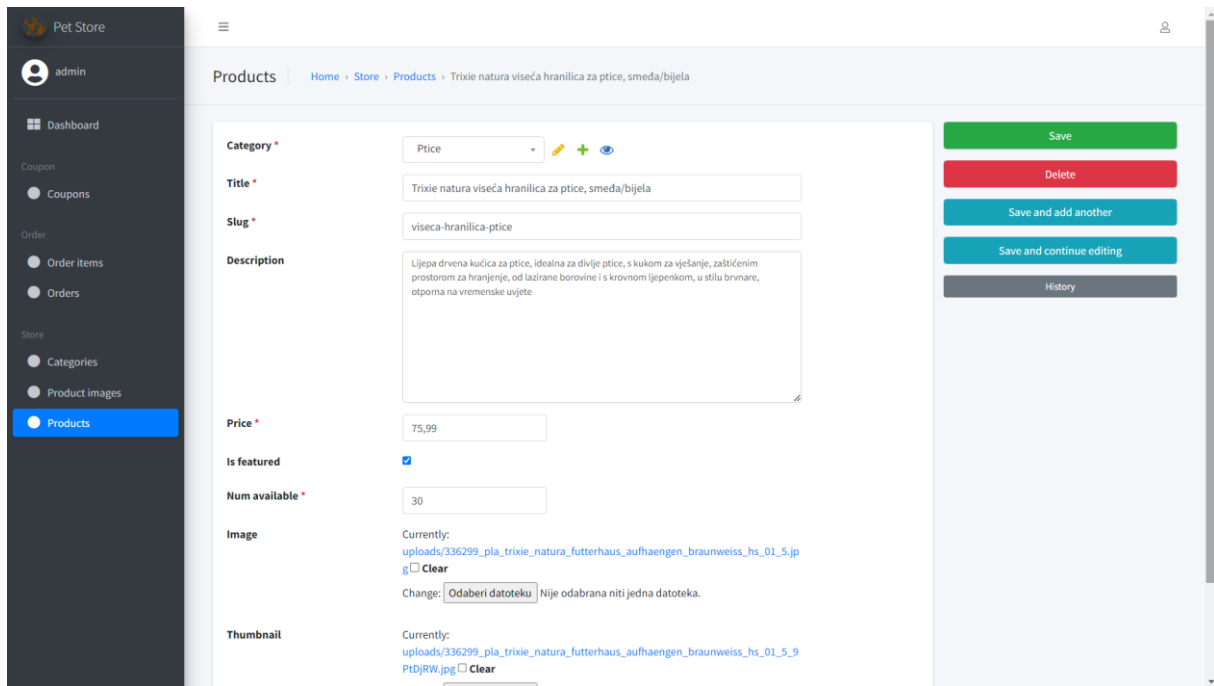
Slika 5.5.5. Prikaz detalja o specifičnoj narudžbi u administratorskom sučelju.

Slika 5.5.6. predstavlja popis svih trenutnih proizvoda prikazanih u administratorskom sučelju.



Slika 5.5.6. Prikaz proizvoda u administratorskom sučelju.

Klikom na ime proizvoda, admin je preusmjeren na detalje proizvoda gdje je prikazana kategorija, ime, opis, cijena, polje za isticanje proizvoda na početnoj stranici, trenutni broj dostupnih proizvoda, te slika u originalnoj kvaliteti i *thumbnaila* - sličica.



**Slika 5.5.7.** Prikaz detalja o proizvodima.

## 6. ZAKLJUČAK

Kupnja putem web trgovina za ljubimce postala je ključan dio svakodnevice za mnoge vlasnike ljubimaca, omogućujući im pristup širokom asortimanu proizvoda bez obzira na vrijeme i lokaciju. Ove trgovine nude praktičnost, od 24/7 dostupnosti do jednostavne usporedbe proizvoda, što ih čini sve popularnijima u odnosu na tradicionalne trgovine. Kroz inovacije poput personaliziranih preporuka, korištenje umjetne inteligencije i tehnologija poput strojnog učenja, web trgovine za ljubimce mogu dodatno unaprijediti korisničko iskustvo, pružajući bržu i jednostavniju kupovinu.

Pravilno vođenje web trgovine pruža mogućnost širenja proizvoda i usluga koje trgovci mogu ponuditi, pa se ne pružaju samo proizvodi, već i vrijedni savjeti i podrška vlasnicima ljubimaca. Takav pristup, zajedno s pažnjom prema kvaliteti usluge i proizvodima, može značajno povećati povjerenje kupaca i osigurati dugoročni uspjeh na tržištu.

## LITERATURA

- [1] Pet Network International d. o. o. - <https://www.pet-centar.hr/> [16.08.2024.]
- [2] Zoocity.hr - <https://www.zoocity.hr/> [16.08.2024.]
- [3] Pet Home Shop - <https://pethomeshop.hr/> [16.08.2024.]
- [4] PetSmart - <https://www.petsmart.com/> [16.08.2024.]
- [5] Šapica d.o.o. - <https://www.sapica-osijek.hr/> [16.08.2024.]
- [6] HTML - <https://www.geeksforgeeks.org/html-introduction/> [17.08.2024.]
- [7] CSS - <https://www.simplilearn.com/tutorials/html-tutorial/html-vs-css> [17.08.2024.]
- [8] Bulma - <https://bulma.io/documentation/> [17.08.2024.]
- [9] Django - <https://docs.djangoproject.com/en/5.1/> [17.08.2024.]
- [10] Vue.js - <https://vuejs.org/guide/introduction.html> [17.08.2024.]

## SAŽETAK

**Tema:** Web trgovina za ljubimce

U ovom završnom radu opisana je Web aplikacija trgovine za ljubimce. Web aplikacija napravljena je pomoću tehnologija HTML-a, CSS-a, JavaScripta, Django, Vue.js i Stripe, koje su detaljno opisane su u radu. Funkcionalnosti aplikacije uključuju registraciju korisnika, naručivanje proizvoda, pregled povijesti narudžbi i upravljanje proizvodima. Registrirani korisnici mogu kupovati proizvode i pregledavati narudžbe, dok administrator ima mogućnost upravljanja korisnicima, proizvodima, narudžbama i kuponima. Web aplikacija sadrži stranice kao što su početna stranica, stranica proizvoda, registracija, prijava, profil stranica korisnika i košarica. Registrirani i prijavljeni korisnici imaju pristup stranici „*My account*“, a korisnici koji nisu prijavljeni imaju pristup „*Sign in*“ i „*Login*“ stranicama.

**Ključne riječi:** online kupovina, trgovina za ljubimce, web aplikacija, Django, Vue.js

## **ABSTRACT**

**Title:** Web application for music equipment store

This final paper describes a web application for a pet store. The application is built using HTML, CSS, JavaScript, Django, Vue.js, and Stripe, all of which are detailed in the work. The application's functionalities include user registration, product ordering, viewing order history, and product management. Registered users can purchase products and view their orders, while administrators can manage users, products, orders, and coupons. The web application includes pages such as the homepage, product page, registration, login, user profile, and cart. Registered users have access to the "My account" page, while non-registered users can access the "Sign in" and "Login" pages.

**Key words:** online shopping, pet store, web application, Django, Vue.js

## ŽIVOTOPIS

Katarina Šuljug rođena je 28.09.2001. u Osijeku. Odrasla je u Osijeku i pohađala Osnovnu školu Vladimir Nazor Čepin. Nakon završetka osnovne škole 2016. godine, upisala je smjer Upravni referent u Ekonomskoj i upravnoj školi Osijek. Nakon toga, nastavila je svoje obrazovanje na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, gdje je upisala prijediplomski stručni studij Računarstvo 2020. godine. Godine 2023. odradila je praksu vezanu uz *frontend* tehnologije i korištenje Vue.js razvojnog okvira.