

Web aplikacija za prodaju odjeće

Pejić, Mario

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:200:044760>

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

Download date / Datum preuzimanja: **2024-11-23**

Repository / Repozitorij:

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



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

Sveučilišni prijediplomski studij Računarstvo

WEB APLIKACIJA ZA PRODAJU ODJEĆE

Završni rad

Mario Pejić

Osijek, 2024.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac Z1P: Obrazac za ocjenu završnog rada na sveučilišnom prijediplomskom studiju****Ocjena završnog rada na sveučilišnom prijediplomskom studiju**

Ime i prezime pristupnika:	Mario Pejić
Studij, smjer:	Sveučilišni prijediplomski studij Računarstvo
Mat. br. pristupnika, god.	R4696, 28.07.2021.
JMBAG:	0165091691
Mentor:	izv. prof. dr. sc. Mirko Köhler
Sumentor:	mr. sc. Željko Štanfel
Sumentor iz tvrtke:	
Naslov završnog rada:	Web aplikacija za prodaju odjeće
Znanstvena grana završnog rada:	Programsko inženjerstvo (zn. polje računarstvo)
Zadatak završnog rada:	Izradi web aplikacije za prodaju odjeće koja omogućuje korisnicima registraciju, pretraživanje, pregledavanje, dodavanje artikala u košaricu, mijenjanje količine, naručivanje i pisanje komentara o artiklima. Neregistrirani korisnici mogu samo pretraživati i pregledavati artikle, dok administrator može upravljati artiklima, korisnicima, recenzijama i procesuirati narudžbe. Aplikacija pruža cjelovito korisničko iskustvo i efikasno upravljanje web trgovinom za administratore.
Datum prijedloga ocjene završnog rada od strane mentora:	18.09.2024.
Prijedlog ocjene završnog rada od strane mentora:	Izvrstan (5)
Datum potvrde ocjene završnog rada od strane Odbora:	25.09.2024.
Ocjena završnog rada nakon obrane:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio sveučilišni prijediplomski studij:	27.09.2024.

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

Osijek, 27.09.2024.

Ime i prezime Pristupnika:

Mario Pejić

Studij:

Sveučilišni prijediplomski studij Računarstvo

Mat. br. Pristupnika, godina upisa:

R4696, 28.07.2021.

Turnitin podudaranje [%]:

5

Ovom izjavom izjavljujem da je rad pod nazivom: **Web aplikacija za prodaju odjeće**

izrađen pod vodstvom mentora izv. prof. dr. sc. Mirko Köhler

i sumentora mr. sc. Željko Štanfel

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

1. UVOD	1
1.1. Zadatak završnog rada	2
2. PREGLED SLIČNIH RJEŠENJA	3
2.1. Mango.....	3
2.2. Zara	5
2.3. ABOUT YOU	7
3. KORIŠTENI ALATI I TEHNOLOGIJE	10
3.1. Django	10
3.2. SQLite	11
3.3. Python	11
3.4. HTML	12
3.5. CSS	13
3.6. Bootstrap	13
3.7. Microsoft Visual Studio Code	14
3.8. Git.....	14
4. PROGRAMSKO RJEŠENJE WEB APLIKACIJE	15
4.1. Django modeli	15
4.2. Prijava	18
4.3. Registracija	19
4.4. Početna stranica	21
4.5. Katalog proizvoda.....	22
4.6. Pretraživanje.....	24
4.7. Filtriranje i sortiranje	25
4.8. Detalji proizvoda.....	26
4.9. Košarica	28
4.10. Postavljanje narudžbe.....	31

4.11. Korisnički profil.....	32
4.12. Administrator.....	34
5. KORIŠTENJE WEB APLIKACIJE.....	36
5.1. Gost	36
5.2. Registrirani korisnik.....	40
5.3. Administrator.....	44
6. ZAKLJUČAK.....	46
LITERATURA	47
SAŽETAK	48
ABSTRACT.....	49
PRILOZI.....	50

1. UVOD

Problem, koji se rješava u ovom završnom radu, je izrada web aplikacije za prodaju odjeće, uzimajući u obzir važnost i rast popularnosti web trgovina. Prednosti web trgovina uključuju praktičnost, dostupnost širokog spektra proizvoda, te mogućnost usporedbe cijena i recenzija proizvoda. Međutim, postoje i mane poput nemogućnosti fizičkog pregleda proizvoda prije kupnje i potencijalnih problema s dostavom.

Web aplikacija ima tri tipa korisnika: gost (neregistrirani korisnik), registrirani korisnik i administrator. Registrirani korisnici imaju pristup širokom spektru funkcionalnosti, uključujući pretraživanje proizvoda, pregled detalja o proizvodima, pisanje komentara o proizvodima, dodavanje proizvoda u košaricu, mijenjanje količine proizvoda u košarici, pregled svog korisničkog računa, slanje narudžbi i pregled povijesti svojih narudžbi. Neregistrirani korisnici imaju ograničene mogućnosti, s mogućnošću pretraživanja proizvoda i pregleda njihovih detalja, ali ne mogu koristiti ključne funkcionalnosti poput naručivanja proizvoda. Administrator ima najviši stupanj pristupa i može upravljati korisničkim računima (kreiranje, uređivanje i brisanje), upravljati cijelim katalogom proizvoda (dodavanje, uređivanje i brisanje proizvoda) te upravljati komentarima. Osim toga, administrator ima ključnu ulogu u procesuiranju narudžbi, s mogućnošću postavljanja statusa narudžbi. Ova struktura korisničkih uloga osigurava učinkovito i sigurno upravljanje web trgovinom, pružajući registriranim korisnicima bogato iskustvo kupovine, dok administratorima omogućuje učinkovito vođenje i održavanje trgovine.

Za izradu ovog zadatka koristit će se razni alati i tehnologije. Django web okvir bit će osnovni alat za razvoj web aplikacije. Za izradu korisničkog sučelja aplikacije koristit će se HTML (engl. *HyperText Markup Language*) za strukturu stranica, CSS (engl. *Cascading Style Sheets*) i Bootstrap za stiliziranje i vizualni dizajn Python će biti glavni programski jezik korišten za razvoj aplikacije, dok će Microsoft Visual Studio Code služiti kao razvojno okruženje, pružajući sve potrebne alate za izradu aplikacije. Korištenjem ovih tehnologija i alata, zadatak će biti realiziran na učinkovit i suvremen način, osiguravajući kvalitetu i funkcionalnost konačne web aplikacije.

Izrada web aplikacije za prodaju odjeće opisana je u nekoliko ključnih dijelova. Prvo, daje se pregled sličnih rješenja i dostupnih web aplikacija koje obavljaju sličnu funkcionalnost. Zatim se pruža pregled korištenih alata i tehnologija koje su primijenjene u implementaciji web aplikacije. Slijedi opis procesa izrade same aplikacije, a nakon toga opis korištenja aplikacije. Na kraju rada nalazi se zaključak koji sažima postignute rezultate.

1.1. Zadatak završnog rada

Cilj završnog rada je izrada web aplikacije za prodaju odjeće koja omogućava korisnicima registraciju, pretraživanje, pregled, dodavanje artikala u košaricu, mijenjanje količine, naručivanje i pisanje komentara o proizvodima. Neregistrirani korisnici mogu samo pretraživati i pregledavati proizvode, dok administrator može upravljati proizvodima, korisnicima, komentarima i procesuirati narudžbe. Aplikacija pruža cjelovito korisničko iskustvo i učinkovito upravljanje web trgovinom za administratore.

2. PREGLED SLIČNIH RJEŠENJA

U ovom dijelu opisat će se nekoliko već postojećih web aplikacija koje služe za internetsku prodaju odjeće.

2.1. Mango

Mango [1] je globalna modna marka poznata po svojoj ponudi moderne i kvalitetne odjeće, obuće i dodataka za muškarce, žene i djecu. Osnovana je 1984. godine u Barceloni, Španjolskoj. Mango se brzo proširio na međunarodno tržište i postao jedna od najpoznatijih modnih marki u svijetu. Mango nudi širok asortiman proizvoda, uključujući žensku, mušku i dječju odjeću i obuću te razne modne dodatke i moderne proizvode za uređenje doma. Poznat je po svom elegantnom i suvremenom dizajnu koji kombinira klasične modne elemente s modernim trendovima.

Početna stranica Mango web aplikacije (vidi slike 2.1., 2.2. i 2.3.) osmišljena je tako da korisnicima pruža intuitivno i ugodno iskustvo internetske kupovine, koristeći moderan i responzivan dizajn. Gornji dio stranice (vidi sliku 2.1.) sadrži navigacijsku traku koja je prisutna na svakoj stranici unutar web aplikacije, čineći navigaciju jednostavnom i konzistentnom. Navigacijska traka uključuje sljedeće elemente:

- Logo Mango: Smješten u središnjem dijelu navigacijske trake, logo modne marke odmah privlači pažnju korisnika i omogućava povratak na početnu stranicu jednim klikom.
- Kategorije: Poveznice na glavne kategorije proizvoda kao što su *Žene*, *Muškarci*, *Teen*, *Djeca* i *Dom*. Svaka od ovih kategorija predstavlja zasebnu početnu stranicu koja korisnicima omogućuje brzo i jednostavno pretraživanje specifičnih proizvoda.
- Pretraga: Funkcionalnost pretrage omogućuje korisnicima brzo pronalaženje željenih proizvoda unutar cijele web aplikacije.
- Košarica: Poveznica na košaricu, koja prikazuje broj proizvoda i omogućuje brz pristup detaljima o odabranim proizvodima.

Na dnu svake stranice nalazi se podnožje (vidi sliku 2.3.), koje je također konzistentno prisutno na svim stranicama unutar web aplikacije. Svaka kategorija djeluje kao početna stranica sa svojim jedinstvenim sadržajem, ali zadržava isti navigacijski okvir i podnožje kako bi se osigurala konzistentnost i lakoća korištenja. Početna stranica također uključuje bogat vizualni sadržaj i poveznice na brojne potkategorije odjeće (vidi sliku 2.2.), omogućujući korisnicima brzi pristup specifičnim tipovima proizvoda unutar svake glavne kategorije. Stranica s proizvodima (vidi sliku

2.4.) na Mango web aplikaciji omogućuje korisnicima pregled proizvoda putem mrežnog rasporeda, gdje su prikazane slike, nazivi i cijene proizvoda te poveznice na detalje o svakom proizvodu. Također, korisnici mogu koristiti filtre za pretraživanje kako bi suzili izbor.



Sl. 2.1. Prvi dio početne stranice web aplikacije Mango s navigacijskom trakom [1]



Sl. 2.2. Dio početne stranice s responzivnim dizajnom web aplikacije Mango [1]

Pretplatom na newsletter dobivate popust od 10 % na vašu sljedeću kupovinu

Pretplati se

Pretplatom potvrđujete da ste pročitali
[Pravila o zaštiti privatnosti](#)

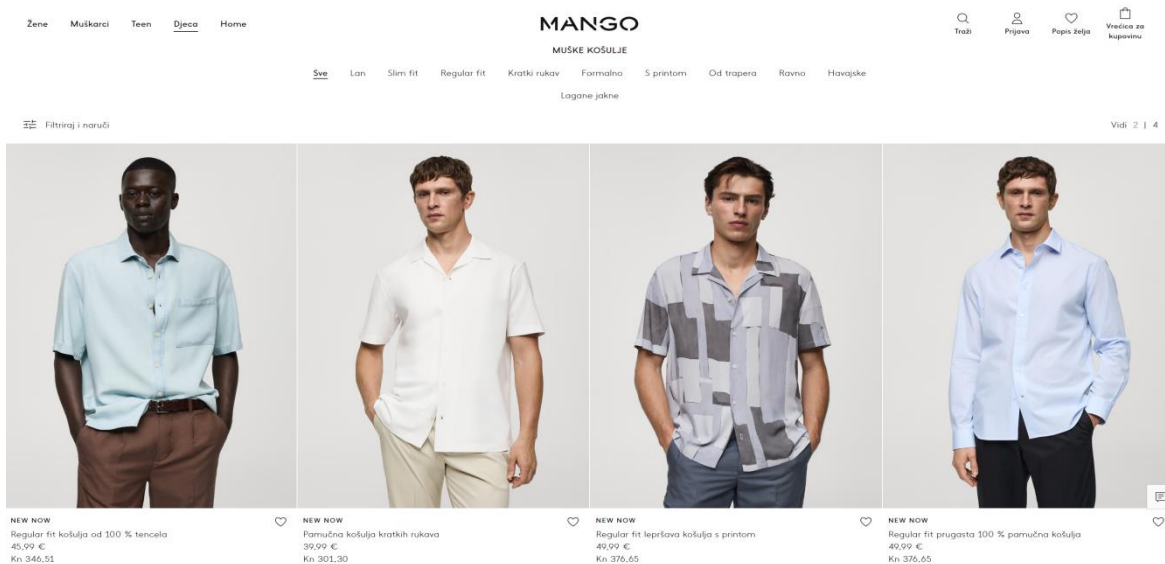
Hrvatska | Hrvatski >

Pomoć | [Moje narudžbe](#) | [Povrati](#) | [Tvrтка](#) | [Radite za Mango](#) | [Tisak](#)
[Mango Outlet](#) | [Site map](#) | [Održivost](#) | [Trgovine](#)



© 2024 MANGO Sva prava pridržana | [Pravila o zaštiti privatnosti i kolačićima](#) | [Ujeti i odredbe](#) | [Kanal Etkia](#)

Sl. 2.3. Podnožje web aplikacije Mango [1]



Sl. 2.4. Stranica muških košulja na web aplikaciji Mango [1]

2.2. Zara

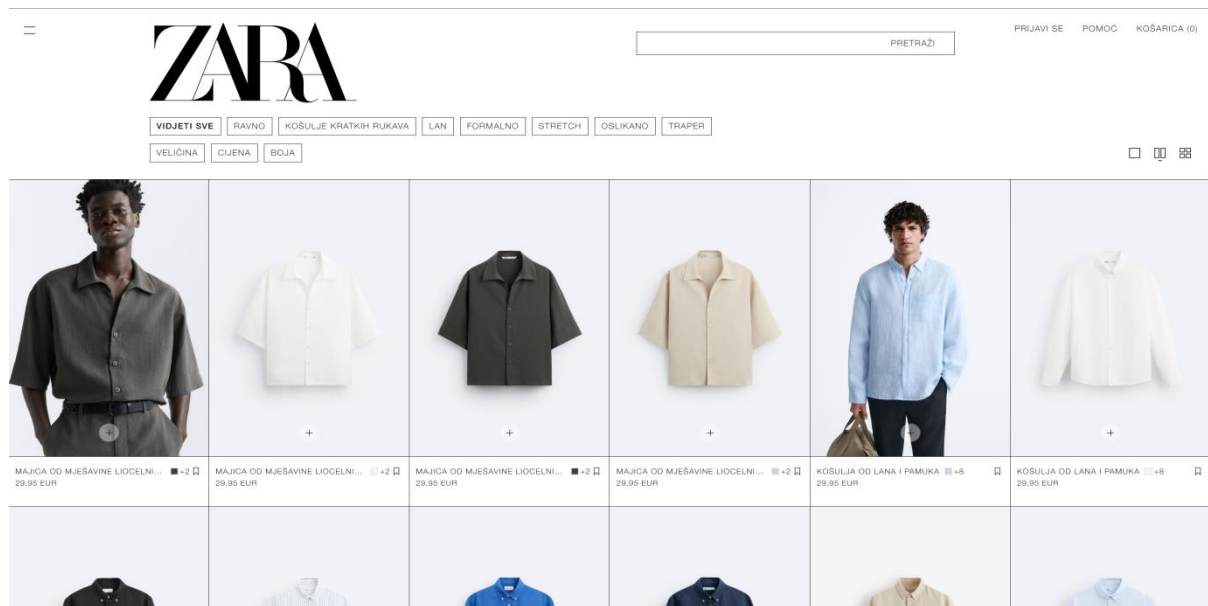
Zara [2] je globalno poznata modna marka osnovana 1974. godine u Španjolskoj. Poznata je po brzom modi koja prati najnovije trendove. Zara je prepoznata po svojoj sposobnosti da brzo uvede nove proizvode u svoje trgovine, što omogućuje kupcima pristup najnovijoj modi. Njihova filozofija kombinira kvalitetu, održivost i pristupačne cijene. S minimalističkim dizajnom, aplikacija omogućuje jednostavnu navigaciju kroz glavne kategorije proizvoda kao što su Žene, Muškarci, Djeca, Dom i Ljepota. Svaka kategorija ima svoju naslovnicu, omogućavajući korisnicima specifičan pregled proizvoda unutar te kategorije. Primjer početne stranice kategorije

Muškarci (vidi sliku 2.5.) u Zara web aplikaciji ima moderan i responzivan dizajn, pružajući korisnicima jednostavan pristup različitim potkategorijama proizvoda. Na vrhu stranice nalazi se navigacijska traka s logom tvrtke, poveznicama na glavne kategorije, opcijom za pretragu i poveznicom na košaricu. Također, tu je i opcija prijave za korisnike, omogućavajući im pristup svojim računima i povijesti narudžbi.

Stranica pregleda pojedinih kategorija proizvoda primjerice stranica s muškim košuljama (vidi sliku 2.6.), nudi detaljan prikaz svakog proizvoda. Svaki proizvod je predstavljen visokokvalitetnim slikama, nazivom i cijenom. Korisnici mogu kliknuti na poveznicu za detalje kako bi saznali više o proizvodu, uključujući dostupne veličine, boje, materijale i recenzije korisnika. Za lakše pronalaženje željenih proizvoda, korisnici mogu koristiti različite opcije filtriranja i sortiranja.



Sl. 2.5. Početna stranica kategorije Muškarci web aplikacije Zara [2]



Sl. 2.6. Stranica s muškim košuljama web aplikacije Zara [2]

2.3. ABOUT YOU

ABOUT YOU [3] je europska internetska platforma za modu koja se ističe svojim modernim pristupom kupovini odjeće i modnih dodataka. Ova platforma nudi širok spektar proizvoda za žene, muškarce i djecu, uključujući odjeću, obuću, modne dodatke i kozmetiku.

Na početnoj stranici (vidi sliku 2.7.) korisnici mogu pronaći bogatu navigacijsku traku s jasno označenim kategorijama proizvoda. Svaka kategorija ima svoju naslovnicu koja omogućava korisnicima brz pristup raznovrsnoj ponudi stilova i modni marki. Izgled početne stranice kategorije muškarci ističe se bogatim dizajnom koji omogućuje korisnicima jednostavan pristup raznim kategorijama proizvoda. Primjer za dio početne stranica s poveznicama za neke od kategorija može se vidjeti na slici 2.8.. Podnožje web aplikacije prikazano je na slici 2.9..

Svaki proizvod je predstavljen slikama, cijenom i kratkim opisom, s mogućnošću klika na detalje kako bi se saznalo više o proizvodu (vidi sliku 2.10.). Funkcionalnosti filtriranja i sortiranja proizvoda nalaze se ispod navigacijske trake. S lijeve strane stranice nalaze se poveznice na sve kategorije i potkategorije proizvoda, olakšavajući korisnicima istraživanje i navigaciju kroz različite opcije.



Sl. 2.7. Prvi dio početne stranice kategorije muškarci web aplikacije ABOUT YOU [3]

Brand Focus

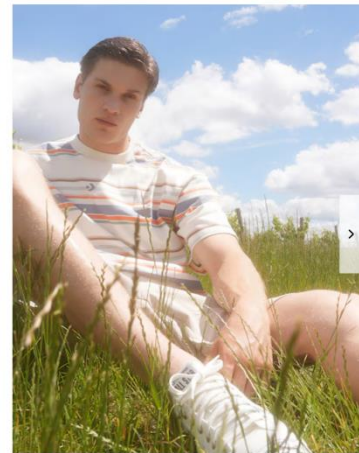
Ne smiješ propustiti ove modne marke



Ljetni stil života
Tommy Jeans



Odredište Ljeto
Tommy Hilfiger



(Ponovno) predstavljamo Star Player 76.
Converse

Sl. 2.8. Dio početne stranice kategorije muškarci web aplikacije ABOUT YOU [3]

ABOUT YOU



SLUŽBA ZA KORISNIKE

- Kontakt i pomoć
- Partnerski program
- Suradnje s kreatorima
- Područje dostave

SIGURNA KUPNJA

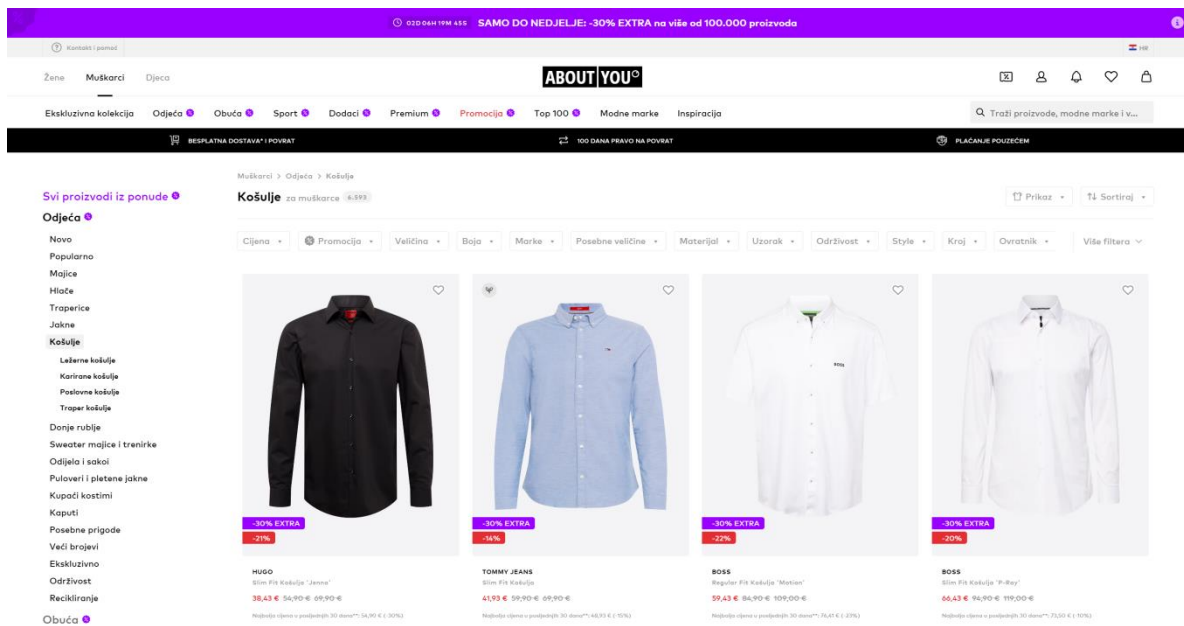
Vaši podaci su s nama sigurni

*Besplatna dostava za narudžbe od 17,90 € i iznad, u suprotnom troškovi dostave od 2,90 € će biti primijenjeni.
**Najniža ukupna cijena u zadnjih 30 dana prije sniženja cijene.
***Besplatno sa svih domaćih mreža. Za pozive iz inozemstva mogu se naplaćivati naknade.
*****Sve cijene uključuju PDV.

O nama Mediji Karijera Odnosi s investitorima Zaštita podataka Centar preferencija Opći uvjeti poslovanja Osnovne informacije

© 2024 ABOUT YOU SE & Co. KG

Sl. 2.9. Podnožje web aplikacije ABOUT YOU [3]



Sl. 2.10. Primjer stranice s muškim košuljama na web aplikacije ABOUT YOU [3]

3. KORIŠTENI ALATI I TEHNOLOGIJE

3.1. Django

Django je visokorazinski (engl. *high-level*) Python web okvir koji potiče brzo razvijanje i čist, pragmatičan dizajn. Django preuzima veliki dio opterećenja web razvoja što omogućuje pisanje aplikacija bez potrebe za ponovnim izmišljanjem već postojećih rješenja. Korištena verzija pri izradi ovog zadatka je Django 5.0.1 [4].

Razlozi zašto je korišten Django web okvir za implementaciju ovog završnog rada su:

- Django je besplatan i otvorenog je koda.
- Izuzetno je brz - pomoću Django-a možete web aplikacije dovesti od koncepta do implementacije u vrlo kratkom roku.
- Vrlo je siguran - Django ozbiljno pristupa sigurnosti i pomaže u zaobilasku brojnih sigurnosnih grešaka poput SQL injekcija i sl. Njegov sustav autentifikacije korisnika omogućuje sigurno upravljanje korisničkim računima i lozinkama.
- Izuzetno skalabilan - neke od najprometnijih stranica na webu koriste Django zbog njegove sposobnosti brzog i fleksibilnog skaliranja.
- Potpuno opremljen - Django se brine o brojnim zadacima pri web razvoju, kao što su autentifikacija korisnika, administracija sadržaja i sl. [5].

Django radi prema MVT dizajnu (engl. *Model View Template*).

Model predstavlja podatke koje želite prikazati, obično iz baze podataka. Modeli su obično smješteni u datoteci *models.py*.

View ili pogled predstavlja funkcija ili metoda koja prima HTTP zahtjeve, uvozi relevantne modele, dohvaća podatke za slanje u predložak i vraća konačni rezultat. Pogledi su obično smješteni u datoteci *views.py*.

Template predstavlja datoteka koja opisuje kako bi rezultat trebao biti predstavljen. Predlošci su često .html datoteke te su obično smješteni u mapi *templates*.

Django omogućava navigaciju po različitim stranicama na web aplikaciji. Kada korisnik zatraži URL, Django odlučuje kojem pogledu će ga poslati, što je definirano u datoteci *urls.py*. Kada preglednik zatraži URL, Django prima URL, provjerava *urls.py* datoteku i poziva pogled koji odgovara URL-u. Pogled iz *views.py* datoteke provjerava relevantne modele iz *models.py*, šalje podatke određenom predlošku u mapi *templates*, koji zatim vraća završeni HTML sadržaj pregledniku [6].

3.2. SQLite

Pri izradi ovog zadatka korištena je baza podataka SQLite. SQLite je moćna knjižnica koja omogućuje implementaciju samostalnog poslužitelja bez potrebe za konfiguracijom te pruža transakcijski SQL mehanizam za upravljanje bazom podataka. Za razliku od većine drugih SQL rješenja, SQLite ne zahtijeva zaseban poslužiteljski proces već direktno upravlja podacima putem običnih diskovnih datoteka. Sve komponente baze podataka, kao što su tablice, indeksi, okidači i pogledi, pohranjeni su u jednoj datoteci koja je interoperabilna između različitih platformi.

SQLite se ističe svojom kompaktnošću, čime je veličina knjižnice često manja od 750 KiB, ovisno o ciljanoj platformi i postavkama kompajlera. Unatoč kompaktnosti, pruža visoku razinu performansi, osobito kada se koristi s adekvatnom količinom memorije.

SQLite je *open-source* projekt s podrškom međunarodnog tima programera koji kontinuirano proširuju njegove mogućnosti, poboljšavajući pouzdanost i performanse dok održavaju kompatibilnost s objavljenim sučeljem, SQL sintaksom i formatom datoteke baze podataka. Ova popularna knjižnica koristi se u brojnim aplikacijama diljem svijeta zbog svoje jednostavnosti integracije i široke primjenjivosti u raznim industrijskim sektorima [7].

3.3. Python

Kao što je Django okvir za izradu web aplikacija temeljen na Pythonu, tako je programski jezik Python u verziji 3.12.4 korišten za razvoj funkcionalnosti aplikacije i implementaciju svih pozadinskih funkcija.

Python je visokorazinski, interpretirani, fleksibilni i jednostavan za naučiti programski jezik koji je postao izuzetno popularan u svijetu programiranja. Osnovao ga je Guido van Rossum 1991. godine.

Neke od prednosti Pythona su:

- rad na različitim platformama (Windows, Mac, Linux, Raspberry Pi)
- jednostavna sintaksa slična engleskom jeziku
- može se koristiti na proceduralan, objektno orijentiran i funkcionalan način
- podržava obilje knjižnica i modula koji olakšavaju razvoj raznih aplikacija, od web razvoja do znanstvenog računanja.
- široka i aktivna zajednica korisnika i programera koji konstantno doprinose novim alatima i rješenjima

Važno je napomenuti da je najnovija glavna verzija Python 3, dok je Python 2 i dalje u upotrebi i prima samo sigurnosna ažuriranja [8].

Python je popularan u mnogim područjima kao što su web razvoj (Django, Flask), znanstveno računanje (kroz knjižnice poput NumPy i Pandas), umjetna inteligencija i strojno učenje (s TensorFlow, PyTorch), automatizacija i još mnogo toga.

3.4. HTML

HTML je osnovni jezik za izradu web stranica. Koristi se za strukturiranje i prikaz sadržaja na webu pomoću elemenata koji se zovu oznake (engl. *tag*) koji označavaju vrste sadržaja kao što su naslovi, paragrafi, slike i veze. Uz HTML, često se koristi dodatni jezik kako bi se doradio izgled web stranice. U ovom radu, HTML služi kao temelj za definiranje izgleda i strukture web stranica koje se razvijaju. Trenutna verzija je HTML5, koja je unaprijeđena i proširena u odnosu na prethodne verzije. Pomoću HTML5 napravljena je struktura web stranica kao i način prikazivanja web sadržaja u ovom radu.

Osnovni dijelovi HTML-a uključuju:

- Oznake (engl. *Tags*): Osnovne jedinice HTML-a koje označavaju različite vrste sadržaja. Svaki tag počinje otvarajućim `<tag>` i završava zatvarajućim `</tag>` dijelom.
- Elementi (engl. *Elements*): Element je kombinacija otvarajućeg taga, sadržaja i zatvarajućeg taga. Primjer je `<p>Ovo je paragraf</p>`, gdje je `<p>` otvarajući tag, tekst "Ovo je paragraf" sadržaj, i `</p>` zatvarajući tag.
- Atributi (engl. *Attributes*): Atributi pružaju dodatne informacije o elementima . Na primjer, `src` atribut u `` označava izvor slike.
- Tekstualni sadržaj: Tekst koji se prikazuje na web stranici, obično između otvarajućeg i zatvarajućeg taga.
- Struktura: HTML struktura obično uključuje `<html>` tag koji obuhvaća cijelu stranicu, `<head>` koji sadrži metapodatke i stilove, i `<body>` koji sadrži glavni sadržaj stranice.
- Hiperlinkovi: Elementi kao što su `<a>` (engl. *anchor*) koriste se za stvaranje veza (engl. *links*) na druge web stranice ili resurse.
- Lista i tablice: HTML omogućuje kreiranje struktura poput `` za neuređene liste, `` za uređene liste, te `<table>` za tablice.
- Forme: Za prikupljanje podataka od korisnika, HTML koristi elemente kao što su `<form>`, `<input>`, `<textarea>`, `<select>`, i `<button>` [9].

3.5. CSS

CSS je jezik za stiliziranje koji se koristi za opisivanje prezentacije dokumenta napisanog u HTML-u ili XML-u. CSS opisuje kako elementi trebaju biti prikazani na ekranu, na papiru, u govoru ili na drugim medijima [10].

CSS se koristi za:

- stiliziranje teksta - promjena boje, veličine i fonta teksta
- raspored elemenata – raspoređuje elemente poput glavnog sadržaja i bočnih traka
- vizualni efekti - dodavanje animacija i prijelaza
- responzivan dizajn - prilagođava se raznim veličinama ekrana i uređaja [11]

CSS koristi jednostavnu i čitljivu sintaksu koja se sastoji od selektora, svojstava i vrijednosti. Selektor određuje koji HTML elementi će biti stilizirani. Deklaracija se sastoji od svojstva i vrijednosti. Svojstvo je karakteristika koja se stilizira (npr. boja), a vrijednost je određeni koji se primjenjuje na svojstvo. Svaka deklaracija završava točkom - zarezom. Pravilo se sastoji od selektora i jedne ili više deklaracija unutar vitičastih zagrada.

CSS se može koristiti za stiliziranje XML dokumenata na način sličan onome u HTML-u. Kao i u HTML-u, CSS definira izgled XML elemenata, uključujući boje, veličine, raspored na stranici, animacije i prilagodbu različitim uređajima. Sintaksa CSS-a za XML prati isti princip: selektori odabiru XML elemente koje želimo stilizirati, dok deklaracije unutar vitičastih zagrada definiraju njihova svojstva i vrijednosti. Svojstvo predstavlja karakteristiku koju želimo stilizirati, dok vrijednost specificira opciju za to svojstvo. Svaka deklaracija završava točkom – zarezom, a pravilo se sastoji od selektora i jedne ili više deklaracija unutar vitičastih zagrada, omogućujući prilagođeni prikaz sadržaja.

U ovom radu, CSS se zajedno s HTML-om i Bootstrap-om, koristi za implementaciju *frontend* dijela web aplikacije, pri čemu CSS pomaže u stiliziranju sadržaja i osigurava vizualno privlačan dizajn.

3.6. Bootstrap

Bootstrap je popularni *open-source* radni okvir koji omogućava brzu izgradnju responzivnih web stranica i aplikacija. Temelji se na CSS i HTML predlošcima. Bootstrap je iznimno prilagodljiv, dolazi s brojnim, gotovim komponentama kao što su forme, gumbi i sl. Dizajniran je da se automatski prilagodi različitim veličinama ekrana. Podržan je od svih modernih web preglednika.

Kao *open-source* projekt, Bootstrap se aktivno razvija uz doprinos velike zajednice programera [12]. U ovom radu koristi se inačica Bootstrap 5, čime je proširena mogućnost za izradu modernih web stranica.

3.7. Microsoft Visual Studio Code

Microsoft Visual Studio Code (VS Code) je uređivač koda koji kombinira jednostavnost s moćnim alatima za programere kao što su IntelliSense automatsko dovršavanje koda i otklanjanje pogrešaka. Dizajniran je da ne ometa rad programera, omogućavajući brzi ciklus uređivanja, izgradnje i otklanjanja pogrešaka. VS Code podržava macOS, Linux i Windows, pružajući korisnicima konzistentno iskustvo bez obzira na platformu. VS Code podržava stotine programskih jezika.

Otklanjanje pogrešaka u VS Code-u omogućuje pregled varijabli, stog poziva i izvršavanje naredbi u konzoli. Integracija s Git-om omogućuje rad s kontrolom verzija bez napuštanja uređivača, uključujući pregled promjena i razlike. VS Code također podržava Node.js razvoj, kao i druge web tehnologije poput JSX/React, HTML, CSS, SCSS, Less i JSON.

Arhitektura VS Code-a kombinira web, izvorne (engl. *native*) i jezično specifične tehnologije, koristeći Electron za brzinu i fleksibilnost nativnih aplikacija. VS Code koristi industrijski snažan uređivač temeljen na HTML-u i integrira se s tehnologijama poput Roslyn za .NET i Visual Studio motor za otklanjanje pogrešaka. Javna proširivost omogućava programerima izradu i korištenje ekstenzija te prilagođavanje iskustva uređivanja, izgradnje i otklanjanja pogrešaka [13].

Za izradu ovog zadatka korištena je verzija Microsoft Visual Studio Code 1.89.0.

3.8. Git

Za upravljanje projektom korišten je besplatni i otvoreni sustav distribuirane kontrole verzija Git. Git omogućava praćenja promjena u kodu tijekom razvoja. Jednostavan je za učenje i ima mali otisak s nevjerovatno brzim performansama. Njegove ključne funkcionalnosti uključuju jeftino lokalno grananje, podršku za praktično upravljanje fazama razvoja i podršku za višestruke tijekom rada. GitHub je korišten kao internetska platforma za *hosting* Git repozitorija [14].

4. PROGRAMSKO RJEŠENJE WEB APLIKACIJE

U ovom poglavlju bit će opisane sve ključne funkcionalnosti web aplikacije, uključujući načine na koje su implementirane i kako međusobno djeluju unutar sustava. Aplikacija je razvijena kao Django projekt pod nazivom `zavrzni_rad` unutar kojeg je kreirana aplikacija *webshop*. Unutar datoteke *models.py* predstavljena je strukturu podataka projekta; definirano je kako se modeli spremaju u bazu podataka. Datoteka *urls.py* definira kako se URL obrasci povezuju s odgovarajućim pogledima. U ovom radu, funkcije koje se nalaze u datoteci *views.py*, bit će nazivane "pogledima" kako bi se uskladile s Django terminologijom. Ovi pogledi upravljaju logikom aplikacije i određuju što će se prikazati korisnicima na temelju njihovih akcija i zahtjeva.

4.1. Django modeli

Web aplikacija koristi nekoliko modela u Django radnom okviru za organizaciju podataka. Svaki model predstavlja određeni entitet u bazi podataka, a njegovi atributi definiraju karakteristike tog entiteta. Modeli s atributima koji su potrebni za web aplikaciju su sljedeći:

Account (Korisnički račun)

- *user*: Povezan je s ugrađenim *User* modelom iz Django okvira (1:1 relacija). Ovaj atribut povezuje korisnički račun s osnovnim korisničkim podacima.
- *name*: Ime korisnika.
- *surname*: Prezime korisnika.
- *phone*: Telefonski broj korisnika.
- *email*: Jedinstvena elektronska pošta korisnika.
- *city*, *postal_number*, *street*, *house_number*: Adresa korisnika.
- *iban*: Opcionalni IBAN korisnik za povrate.
- *bank*: Opcionalna banka korisnika za povrate.

Category (Kategorija proizvoda)

- *name*: Naziv kategorije.
- *slug*: Jedinstveni identifikator u URL-u.

SubCategory (Potkategorija)

- *name*: Naziv potkategorije.
- *slug*: Jedinstveni identifikator u URL-u.

- *category*: Veza s *Category* modelom (strani ključ).
- *image*: Slika potkategorije.

Color (Boja proizvoda)

- *name*: Naziv boje.
- *slug*: Jedinstveni identifikator u URL-u.

Size (Veličina)

- *name*: Naziv veličine.

Product (Proizvod)

- *category*: Veza s *Category* modelom (strani ključ).
- *subcategory*: Veza s *SubCategory* modelom (strani ključ).
- *name*: Naziv proizvoda.
- *slug*: Jedinstveni identifikator u URL-u.
- *price*: Cijena proizvoda.
- *details, composition, care_instructions*: Opis proizvoda, njegov sastav i upute za održavanje.
- *created_at*: Datum kada je proizvod kreiran.
- *reference_number*: Jedinstveni referentni broj proizvoda.

ProductVariation (Varijacija proizvoda)

- *product*: Veza s *Product* modelom (strani ključ).
- *color*: Veza s *Color* modelom (strani ključ).
- *sizes*: Povezane veličine varijacije proizvoda (preko *ProductVariationSize* modela) (1:više relacija).
- *main_image*: Glavna slika proizvoda.

ProductVariationImage (Slika varijacije proizvoda)

- *product_variation*: Veza s *ProductVariation* modelom (strani ključ).
- *image*: Ostale slike varijacije proizvoda.

ProductVariationSize (Veličina varijacije proizvoda)

- *product_variation*: Veza s *ProductVariation* modelom (strani ključ).

- *size*: Veza s *Size* modelom (strani ključ).
- *is_available*: Označava dostupnost veličine.

Cart (Košarica)

- *user*: Veza s *Account* modelom (1:1 relacija).
- *items*: Veza s *CartItem* modelom (1:više relacija).

CartItem (Stavka u košarici)

- *cart*: Veza s *Cart* modelom (strani ključ).
- *product*: Veza s *ProductVariation* modelom (strani ključ).
- *size*: Veličina varijacije proizvoda (preko *ProductVariationSize* modela).
- *quantity*: Količina stavke.

Order (Narudžba)

- *account*: Veza s korisničkim računom (strani ključ).
- *date_ordered*: Datum narudžbe.
- *status*: Status narudžbe (npr. "U pripremi", "Poslano").
- *payment_method*: Metoda plaćanja.
- *order_number*: Jedinstveni broj narudžbe.
- *items*: Veza s *OrderItem* modelom (1:više relacija).

OrderItem (Stavka u narudžbi)

- *order*: Veza s *Order* modelom (strani ključ).
- *product*: Varijacija proizvoda koji je naručena; veza sa stranim ključem.
- *size*: Veličina proizvoda.
- *quantity*: Količina proizvoda.

Comment (Komentar)

- *product*: Varijacija proizvoda na koji se komentar odnosi; veza sa stranim ključem.
- *poster*: Korisnik koji je postavio komentar; veza sa stranim ključem.
- *text*: Tekst komentara.
- *created_at*: Vrijeme kada je komentar postavljen.

Account model predstavlja korisnika aplikacije i pohranjuje njegove osobne podatke. Svaki korisnik može imati jednu košaricu (engl. *Cart*) i može napraviti više narudžbi (engl. *Order*). Košarica se isprazni nakon uspješne narudžbe. Ove narudžbe povezuju se s korisničkim računom putem veze sa stranim ključem, što omogućuje praćenje svih narudžbi koje je korisnik napravio.

Modeli kategorija i potkategorija pomažu u organizaciji proizvoda. Kategorija (engl. *Category*) predstavlja glavne skupine proizvoda, dok potkategorija (engl. *SubCategory*) omogućuje detaljnije kategoriziranje unutar svake glavne kategorije. Svaki proizvod (engl. *Product*) pripada određenoj kategoriji i potkategoriji, što olakšava pretraživanje, filtriranje i sortiranje proizvoda.

Product model sadrži osnovne informacije o proizvodu, dok *ProductVariation* model omogućuje različite varijacije tog proizvoda (npr. različite boje i različite slike). Svaka varijacija može imati više veličina (engl. *ProductVariationSize*), a *ProductVariationImage* model omogućuje pohranu slika za svaku varijaciju. Boje su definirane u *Color* modelu i povezane su s *ProductVariation*.

Cart model predstavlja košaricu korisnika i povezan je s *CartItem* modelom koji predstavlja pojedinačne stavke u košarici. Svaka stavka u košarici povezana je s varijacijom proizvoda i s modelom *ProductVariationSize* kako bi se precizno prikazali odabrani proizvodi i veličine.

Order model predstavlja narudžbe koje korisnici postavljaju. Svaka narudžba može sadržavati više *OrderItem* (stavki narudžbe), koje su povezane s *ProductVariation* modelom. Stavke narudžbe sadrže informacije o varijaciji proizvoda.

Comment model omogućuje korisnicima da ostave komentare na varijacije proizvoda. Svaki komentar povezan je s *ProductVariation* i korisnikom (engl. *Account*) koji je napisao komentar [6].

4.2. Prijava

Pogled *login_view* (vidi sliku 4.1.) služi za omogućavanje prijave korisnika u sustav, dok pogled *logout_view* (vidi sliku 4.1.) omogućuje korisnicima odjavu. Prijava zahtjeva od korisnika da unese svoju elektronsku poštu i lozinku i pošalje formu. Funkcija *authenticate* provjerava korisničke vjerodajnice (elektronsku poštu i lozinku) i pokušava pronaći korisnika u bazi podataka. Django koristi elektronsku poštu (engl. *email*) kao *username* kako bi provjerio je li korisnik registriran. Ako je unos ispravan i korisnik postoji u bazi podataka, funkcija vraća korisnički objekt. Ako unos nije točan, vraća *None*. Ako autentifikacija uspije i korisnik postoji, poziva se pogled *login_view*, koji prijavljuje korisnika i postavlja potrebne podatke u sesiju. Nakon uspješne prijave, korisnik se preusmjerava na stranicu profila. Ako autentifikacija nije uspješna, korisnik

ostaje na istoj stranici i prikazuje se poruka o pogrešnim vjerodajnicama. Pogled *logout_view* odjavljuje trenutno prijavljenog korisnika tako što briše sve podatke iz sesije povezane s korisnikom. Nakon uspješne odjave, korisnik se preusmjerava natrag na stranicu za prijavu.

```
def login_view(request):
    if request.method == 'POST':
        email = request.POST.get('email')
        password = request.POST.get('password')
        user = authenticate(request, username=email, password=password)
        if user is not None:
            login(request, user)
            return redirect('webshop:profile')
        else:
            messages.error(request, 'Pogrešan email ili lozinka.')
    return render(request, 'webshop/login.html')

def logout_view(request):
    logout(request)
    return redirect('webshop:login')
```

Sl. 4.1. Prikaz pogleda prijave i odjave

4.3. Registracija

Registracija korisnika u ovoj aplikaciji omogućava neregistriranim korisnicima (gostima) da kreiraju korisnički račun, što im omogućava pristup dodatnim funkcionalnostima, kao što su dodavanje proizvoda u košaricu, stvaranje narudžbi, pregleda korisničkog računa, pregled povijesti narudžbi te pisanje komentara. Registracija se provodi putem pogleda *register_view* (vidi sliku 4.2.).

Kada neregistrirani korisnik pristupi stranici za registraciju i pošalje ispunjenu formu putem metode POST, pregledava se nekoliko ključnih koraka:

- Unos podataka: Korisnik mora unijeti svoje osobne podatke, uključujući ime, prezime, elektronsku poštu, grad, poštanski broj, ulicu, kućni broj, broj telefona te lozinku s potvrdom lozinke.
- Provjera postojanja računa: Prije kreiranja novog korisnika, sustav provjerava postoji li već korisnik s unesenom elektronskom poštom u bazi podataka, i ako postoji, sustav vraća poruku o grešci i omogućuje korisniku da ispravi svoje podatke. Provjera elektronske pošte vrši se kako ne bi postojalo više korisnika s istim korisničkim računom.

- Provjera lozinke: Sljedeći korak je provjera valjanosti lozinke. Ako je lozinka kraća od osam znakova, sustav vraća poruku o grešci i omogućuje unos ispravne lozinke. Također, lozinka mora biti identična potvrđenoj lozinki. Ako se lozinke ne podudaraju, sustav javlja grešku korisniku.
- Kreiranje korisnika: Ako su svi uvjeti zadovoljeni, kreira se novi korisnički račun.
- Nakon toga, dodatni korisnički podatci, poput imena, adrese i broja telefona, pohranjuju se u povezani *Account* model kako bi se proširila osnovna funkcionalnost Django *User* modela.
- Uspješna registracija: Kada je proces registracije uspješno završen, korisniku se prikazuje poruka o uspjehu i on se preusmjerava na stranicu za prijavu, gdje se može prijaviti s upravo kreiranim podacima.

```

def register_view(request):
    if request.method == 'POST':
        name = request.POST.get('name')
        surname = request.POST.get('surname')
        email = request.POST.get('email')
        city = request.POST.get('city')
        postal_number = request.POST.get('postal_number')
        street = request.POST.get('street')
        house_number = request.POST.get('house_number')
        phone = request.POST.get('phone')
        password = request.POST.get('password')
        confirm_password = request.POST.get('confirm_password')

        if User.objects.filter(email=email).exists():
            messages.error(request, 'Račun s ovim emailom već postoji!')
            return render(request, 'webshop/register.html', {
                'name': name,
                'surname': surname,
                'email': email,
                'city': city,
                'postal_number': postal_number,
                'street': street,
                'house_number': house_number,
                'phone': phone,
            })

        if len(password) < 8:
            messages.error(request, 'Lozinka mora sadržavati najmanje 8 znakova!')
            return render(request, 'webshop/register.html', {
                'name': name,
                'surname': surname,
                'email': email,
                'city': city,
                'postal_number': postal_number,
                'street': street,
                'house_number': house_number,
                'phone': phone,
            })

        if password != confirm_password:
            messages.error(request, 'Lozinke se ne podudaraju!')
            return render(request, 'webshop/register.html', {
                'name': name,
                'surname': surname,
                'email': email,
                'city': city,
                'postal_number': postal_number,
                'street': street,
                'house_number': house_number,
                'phone': phone,
            })

        user = User.objects.create_user(username=email, email=email, password=password)
        Account.objects.create(
            user=user,
            name=name,
            surname=surname,
            email=email,
            phone=phone,
            city=city,
            postal_number=postal_number,
            street=street,
            house_number=house_number
        )
        messages.success(request, 'Registracija je uspješna! Sada se možete prijaviti.')
        return redirect('webshop:login')

    return render(request, 'webshop/register.html')

```

Sl. 4.2. Prikaz pogleda za registraciju

4.4. Početna stranica

U aplikaciji postoje dvije početne stranice, jedna za žensku odjeću, a druga za mušku odjeću. Prikaz na ženskoj početnoj stranici (pogled *female_homepage_view*, vidi sliku 4.3.) i muškoj početnoj stranici (pogled *male_homepage_view*, vidi sliku 4.3.) uključuje:

- Potkategorije proizvoda: Korisnici mogu birati između različitih potkategorija odjeće, koje se dinamički dohvaćaju i prikazuju na temelju odabrane kategorije (žene ili muškarci).
- Najnoviji proizvodi: Na svakoj početnoj stranici prikazuju se najnoviji proizvodi iz odabrane kategorije, s limitiranim brojem varijacija proizvoda (najviše osam).

```
def female_homepage_view(request):
    category = Category.objects.get(name='zene')
    subcategory = SubCategory.objects.filter(category=category).first()

    latest_products = ProductVariation.objects.filter(
        product__category=category
    ).order_by('-product__created_at')[:8]

    context = {
        'subcategories': SubCategory.objects.filter(category=category),
        'latest_products': latest_products,
        'category_name': category.slug,
        'subcategory_name': subcategory.slug if subcategory else 'sve',
    }
    return render(request, 'webshop/homepage.html', context)

def male_homepage_view(request):
    category = Category.objects.get(name='muskarci')
    subcategory = SubCategory.objects.filter(category=category).first()

    latest_products = ProductVariation.objects.filter(
        product__category=category
    ).order_by('-product__created_at')[:8]

    context = {
        'subcategories': SubCategory.objects.filter(category=category),
        'latest_products': latest_products,
        'category_name': category.slug,
        'subcategory_name': subcategory.slug if subcategory else 'sve',
    }
    return render(request, 'webshop/homepage.html', context)
```

Sl. 4.3. Prikaz pogleda za stvaranje početnih stranica

4.5. Katalog proizvoda

U aplikaciji je implementiran detaljan katalog proizvoda, koji omogućava korisnicima pretragu i filtriranje proizvoda unutar specifičnih kategorija i potkategorija. Funkcionalnost kataloga proizvoda implementirana je putem pogleda *articles_display_view* (vidi sliku 4.4.), koji omogućuje prikaz varijacija proizvoda unutar odabrane potkategorije, zajedno s mogućnošću sortiranja i filtriranja po različitim atributima.

```

def articles_display_view(request, category_name, subcategory_name):
    sort = request.GET.get('sort', 'price_asc')
    selected_colors = [color for color in request.GET.getlist('color') if color.isdigit()]
    selected_sizes = [size for size in request.GET.getlist('size') if size.isdigit()]
    category = get_object_or_404(Category, slug=category_name)
    subcategory = get_object_or_404(SubCategory, slug=subcategory_name, category=category)
    products = Product.objects.filter(subcategory=subcategory)
    product_variations = ProductVariation.objects.filter(product__in=products)

    if selected_colors:
        product_variations = product_variations.filter(color__id__in=selected_colors)

    if selected_sizes:
        product_variation_sizes = ProductVariationSize.objects.filter(
            size__id__in=selected_sizes,
            is_available=True
        ).values_list('product_variation_id', flat=True)
        product_variations = product_variations.filter(id__in=product_variation_sizes)

    if sort == 'newest':
        product_variations = product_variations.order_by('-product__created_at')
    elif sort == 'oldest':
        product_variations = product_variations.order_by('product__created_at')
    elif sort == 'price_asc':
        product_variations = product_variations.order_by('product__price')
    elif sort == 'price_desc':
        product_variations = product_variations.order_by('-product__price')

    filtered_products = Product.objects.filter(variations__in=product_variations).distinct()

    context = {
        'category_name': category_name,
        'subcategory_name': subcategory_name,
        'products': filtered_products,
        'filtered_product_variations': product_variations.distinct(),
        'colors': Color.objects.all(),
        'sizes': Size.objects.all(),
        'selected_colors': selected_colors,
        'selected_sizes': selected_sizes,
        'sort': sort,
    }

    return render(request, 'webshop/articles_display.html', context)

```

Sl. 4.4. Pogled za prikaz kataloga proizvoda

Pogled koristi naziv kategorije parametar iz URL-a kako bi pronašao odgovarajuću kategoriju. Nakon toga, pogled traži *Category* objekt u bazi podataka koji odgovara *slug* vrijednosti. Ako kategorija s tim *slug*-om ne postoji javlja se greška koja osigurava da korisnik ne može pristupiti nepostojećim kategorijama. Nakon što je kategorija pronađena, pogled koristi naziv potkategorije kako bi pronašao odgovarajuću potkategoriju unutar te kategorije. Na kraju, pogled priprema kontekst za predložak, uključujući sve filtrirane proizvode, varijacije proizvoda, boje i veličine. Ovaj kontekst omogućuje prikaz relevantnih informacija korisnicima, omogućavajući im vidjeti proizvode koji odgovaraju njihovim kriterijima.

4.6. Pretraživanje

Pogled `search_view` omogućava korisnicima pretraživati varijacije proizvoda na temelju različitih kriterija. Također, omogućuje filtriranje sortiranje rezultata pretrage. Proces pretrage izgleda ovako:

- Dohvaćanje podataka iz URL-a *Query* (vidi sliku 4.5.): *query* parametar iz URL-a sadrži tekst koji korisnik unosi u pretraživač. Ovaj parametar koristi se za pretraživanje varijacija proizvoda na temelju naziva proizvoda, kategorije, potkategorije ili boje.

```
query = request.GET.get('query', '')
```

Sl. 4.5. URL s *query* parametrom

- Dohvaćanje svih objekata varijacija proizvoda (vidi sliku 4.6.).

```
product_variations = ProductVariation.objects.all()
```

Sl. 4.6. Dohvaćanje objekata varijacija proizvoda

- Pretraživanje prema ključnim riječima (vidi sliku 4.7.): Ako korisnik unese tekst u pretraživač (*query* nije prazan), pretražuju se objekti varijacija proizvoda prema ključnim riječima. Koristi se *Q* objekt za sastavljanje složenih upita koji pretražuju varijacije proizvoda na temelju njihovih imena, naziva kategorija, potkategorija i boja.

```
if query:
    keywords = query.split()
    q_objects = Q()
    for keyword in keywords:
        q_objects &= (
            Q(product__name__icontains=keyword) |
            Q(product__category__name__icontains=keyword) |
            Q(product__subcategory__name__icontains=keyword) |
            Q(color__name__icontains=keyword)
        )
    product_variations = product_variations.filter(q_objects).distinct()
```

Sl. 4.7. Pretraživanje prema ključnim riječima

4.7. Filtriranje i sortiranje

Pogledi *articles_display_view* i *search_view* omogućuju korisnicima da na temelju različitih kriterija filtriraju i sortiraju rezultate.

Unutar pogleda *articles_display_view* omogućeno je je filtriranje prema boji i veličini, a unutar pogleda *search_view* samo po veličini. Slijedi opis procesa filtriranja:

- Filtriranje po bojama (vidi sliku 4.8.): Korisnici mogu odabrati boje koje žele vidjeti, a rezultati se filtriraju prema ID-ovima tih boja. Ova funkcionalnost omogućuje prikaz samo onih varijacija proizvoda koje odgovaraju odabranim bojama.
- Filtriranje po veličinama (vidi sliku 4.8.): Korisnici mogu odabrati veličine, a filtriranje se provodi na temelju dostupnosti tih veličina u *ProductVariationSize* modelu. Varijacije proizvoda koje nemaju odabrane veličine ili koje nisu dostupne se isključuju iz rezultata.

```
if selected_colors:
    product_variations = product_variations.filter(color__id__in=selected_colors)

if selected_sizes:
    product_variation_sizes = ProductVariationSize.objects.filter(
        size__id__in=selected_sizes,
        is_available=True
    ).values_list('product_variation_id', flat=True)
    product_variations = product_variations.filter(id__in=product_variation_sizes)
```

Sl. 4.8. Logika filtriranja

Proces sortiranje je opisan u nastavku:

- Prvotno se dohvaća opcija sortiranja (vidi sliku 4.9.).

```
sort = request.GET.get('sort', '')
```

Sl. 4.9. Dohvaćanje opcije sortiranja

- Zatim slijedi sortiranje prema odabranoj opciji (vidi sliku 4.10.):

```
if sort == 'newest':
    product_variations = product_variations.order_by('-product__created_at')
elif sort == 'oldest':
    product_variations = product_variations.order_by('product__created_at')
elif sort == 'price_asc':
    product_variations = product_variations.order_by('product__price')
elif sort == 'price_desc':
    product_variations = product_variations.order_by('-product__price')
```

Sl. 4.10. Logika sortiranja

Mogućnosti načina sortiranja su:

- Sortiranje prema najnovijem: Rezultati se sortiraju prema datumu kreiranja proizvoda, od najnovijeg do najstarijeg.
- Sortiranje prema najstarijem: Rezultati se sortiraju prema datumu kreiranja proizvoda, od najstarijeg do najnovijeg.
- Sortiranje prema cijeni (najpovoljnije): Rezultati se sortiraju prema cijeni proizvoda, od najniže do najviše.
- Sortiranje prema cijeni (najskuplje): Rezultati se sortiraju prema cijeni proizvoda, od najviše do najniže.

4.8. Detalji proizvoda

Pogled *articles_details_view* (vidi sliku 4.11.) pruža korisnicima detaljne informacije o određenoj varijaciji proizvoda. Na stranici su uključeni brojni detalji o proizvodu, uključujući naziv, cijenu, boju, opis i sastav varijacije proizvoda. Također, prikazane su sve slike povezane s odabranom varijacijom proizvoda, omogućavajući korisnicima da pregledaju vizualne prikaze različitih aspekata varijacije proizvoda.

Na stranici su dostupne poveznice koje omogućavaju korisnicima pregledati sve ostale varijacije proizvoda prema bojama. Ovo olakšava korisnicima pregled različitih opcija boja i odabir željene varijacije. Korisnici mogu odabrati dostupne veličine za odabranu varijaciju proizvoda. Funkcionalnost dodavanja proizvoda u košaricu je omogućena putem gumba. Ovaj gumb omogućuje korisnicima da lako dodaju proizvod u svoju košaricu s odabranom varijacijom i veličinom.


```

def articles_details_view(request, category_name, subcategory_name, product_slug, color):
    category = get_object_or_404(Category, slug=category_name)
    subcategory = get_object_or_404(SubCategory, slug=subcategory_name, category=category)
    product = get_object_or_404(Product, slug=product_slug, subcategory=subcategory)

    if color:
        try:
            selected_variation = product.variations.get(color__slug=color)
        except ProductVariation.DoesNotExist:
            selected_variation = product.variations.first()
    else:
        selected_variation = product.variations.first()

    variation_images = selected_variation.images.all()
    available_sizes = selected_variation.variation_sizes.filter(is_available=True)

    if request.method == 'POST':
        text = request.POST.get('text')
        if text:
            account = get_object_or_404(Account, user=request.user)
            Comment.objects.create(
                product=selected_variation,
                poster=account,
                text=text,
                created_at=timezone.now()
            )
            return redirect('webshop:articles_details', category_name=category_name, subcategory_name=subcategory_name,
                            product_slug=product_slug, color=selected_variation.color.slug)

    context = {
        'category_name': category_name,
        'subcategory_name': subcategory_name,
        'product': product,
        'selected_variation': selected_variation,
        'variation_images': variation_images,
        'variations': product.variations.all(),
        'available_sizes': available_sizes
    }
    return render(request, 'webshop/articles_details.html', context)

```

Sl. 4.11. Pogled za prikaz detalja varijacije proizvoda

Unutar predloška za prikaz detalja varijacije proizvoda nalaze se i komentari koje su ostavili prethodni korisnici (vidi sliku 4.12.). Ova sekcija omogućava korisnicima pročitati komentare drugih kupaca i ostaviti vlastite komentare, ako su prijavljeni.

```

<div class="comments">
  <h2>KOMENTARI</h2>
  <div class="comments-list">
    {% for comment in selected_variation.comments.all %}
      <div class="comment">
        <p>{{ comment.text }}</p>
        <small>{{ comment.poster.name }}, {{ comment.created_at|date:"d.m.Y" }}</small>
      </div>
    {% endfor %}
  </div>

  {% if user.is_authenticated %}
  <form method="post" action="{% url 'webshop:articles_details' category_name subcategory_name product.slug selected_variation.color.slug %}">
    {% csrf_token %}
    <textarea name="text" rows="4" placeholder="Ostavite komentar..."></textarea>
    <br>
    <button type="submit">Pošalji komentar</button>
  </form>
  {% endif %}
</div>

```

Sl. 4.12. Prikaz komentara i mogućnosti komentiranja

4.9. Košarica

Košarica je ključni dio procesa narudžbe jer omogućuje korisnicima pregledati i upravljati proizvodima koje su odabrali prije nego što izvrše narudžbu. Svaki proizvod u košarici predstavljen je ključnim informacijama poput naziva proizvoda, boje, veličine, količine i cijene. Korisnici mogu prilagoditi količinu pojedinog proizvoda, ukloniti proizvode i pregledati ukupne troškove, uključujući cijenu dostave. Pogled *cart_view* (vidi sliku 4.13.) omogućava korisnicima pregled sadržaja njihove košarice. Ključni elementi ovog pogleda su:

- Dohvaćanje košarice i stavki: Uvijek se prvo dohvaća košarica korisnika ili stvara ako korisnik nije imao prethodnih kupnji, a zatim sve stavke koje su u njoj.
- Izračun ukupne cijene: Za svaku stavku u košarici izračunava se ukupna cijena na temelju cijene proizvoda i njegove količine. Zbroj svih stavki daje ukupnu cijenu prije dostave.
- Kontekst za predložak: Informacije koje se šalju predlošku uključuju košaricu, stavke u košarici, ukupnu cijenu prije dostave, trošak dostave i konačnu ukupnu cijenu. Ovo omogućava korisnicima vidjeti sve stavke koje su dodali i koliko će platiti.

```
def cart_view(request):
    cart, created = Cart.objects.get_or_create(user=request.user.account)
    cart_items = cart.items_in_cart.all()

    subtotal = 0
    for item in cart_items:
        product_price = item.product.product.price
        item_total = item.quantity * product_price
        subtotal += item_total

    delivery_cost = 0 if subtotal >= 50 else 5
    total_price = subtotal + delivery_cost

    context = {
        'cart': cart,
        'cart_items': cart_items,
        'subtotal': subtotal,
        'delivery_cost': delivery_cost,
        'total_price': total_price,
    }

    return render(request, 'webshop/cart.html', context)
```

Sl. 4.13. Pogled na stvaranje i pregled košarice

Pogled `remove_cart_item` (vidi sliku 4.14.) omogućava korisnicima ukloniti stavke iz svoje košarice. Logika uklanjanja stavke ima nekoliko koraka:

- Pristup stavci: Dohvaća se specifična stavka iz košarice na temelju ID-a stavke i provjerava se pripada li korisniku.
- Obrada uklanjanja: Ako stavka ima više od jedne jedinice, količina se smanjuje za jedan. Ako količina dostigne nulu, stavka se potpuno uklanja iz košarice.
- Preusmjeravanje: Nakon obrade zahtjeva, korisnik se preusmjerava na stranicu košarice kako bi vidio ažurirane stavke.

```
def remove_cart_item(request, item_id):
    cart_item = get_object_or_404(CartItem, id=item_id, cart__user=request.user.account)

    if request.method == 'POST':
        if cart_item.quantity > 1:
            cart_item.quantity -= 1
            cart_item.save()
        else:
            cart_item.delete()

    return redirect('webshop:cart')
```

Sl. 4.14. Pogled za uklanjanje stavke košarice

Unutar prikaza košarice nalazi se forma za uklanjanje stavke košarice (vidi sliku 4.15.).

```
<form action="{% url 'webshop:remove_cart_item' item.id %}" method="post" class="remove-item-form">
    {% csrf_token %}
    <button type="submit" class="remove-item-btn">Ukloni</button>
</form>
```

Sl. 4.15. Forma za uklanjanje stavke košarice

Proces dodavanja proizvoda u košaricu omogućava korisnicima odabrati varijaciju proizvoda (boju, veličinu) te da je zatim pohrane u svoju košaricu za kasniju kupnju.

- Odabir varijacije proizvoda: Na stranici s detaljima proizvoda, korisnici mogu birati između različitih boja i veličina. Nakon odabira željene kombinacije, sustav provjerava dostupnost te varijacije proizvoda u skladištu. Forma za dodavanje proizvoda u košaricu nalazi se na slici 4.16..

```

<form method="post" action="{% url 'webshop:add_to_cart' category_name subcategory_name product.slug selected_variation.color.slug %}">
  {% csrf_token %}
  <h3>Izaberite veličinu:</h3>
  <select name="size" id="size">
    {% for size in available_sizes %}
      <option value="{| size.id |}">{{ size.size.name }}</option>
    {% empty %}
      <option disabled>Nema dostupnih veličina</option>
    {% endfor %}
  </select>
  <input type="hidden" name="quantity" value="1">
  <input type="hidden" name="product_variation_id" value="{{ selected_variation.id }}">
  <input type="hidden" name="color" value="{{ selected_variation.color.slug }}">
  {% if user.is_authenticated and not user.is_superuser %}
  <button type="submit">Dodaj u košaricu</button>
  {% endif %}
</form>

```

Sl. 4.16. Forma za dodavanje artikla u košaricu

- Spremanje u košaricu: Kad korisnik potvrdi odabir, pogled *add_to_cart* (vidi sliku 4.17.) preuzima podatke o varijaciji proizvoda (boji, veličini) te stvara ili ažurira stavku u košarici.

```

def add_to_cart(request, category_name, subcategory_name, product_slug, color=None):
    if request.method == 'POST':
        quantity = int(request.POST.get('quantity', 1))
        product_variation_id = request.POST.get('product_variation_id')
        size_id = request.POST.get('size')

        product_variation = get_object_or_404(ProductVariation, id=product_variation_id)

        size = get_object_or_404(ProductVariationSize, id=size_id) if size_id else None

        cart, created = Cart.objects.get_or_create(user=request.user.account)

        cart_item, created = CartItem.objects.get_or_create(
            cart=cart,
            product=product_variation,
            size=size,
            defaults={'quantity': quantity}
        )

        if not created:
            cart_item.quantity += quantity
            cart_item.save()

        return redirect('webshop:articles_details', category_name=category_name,
            subcategory_name=subcategory_name, product_slug=product_slug, color=color)

```

Sl. 4.17. Pogled za dodavanje artikla u košaricu

- Ako korisnik već ima isti proizvod u košarici, količina se povećava.
- Ako je proizvod novi u košarici, stvara se nova stavka s odgovarajućim atributima.
- Ažuriranje košarice: Nakon dodavanja stavke u košaricu, korisnik se preusmjerava natrag na stranicu s detaljima proizvoda.

4.10. Postavljanje narudžbe

Proces stvaranja narudžbe započinje nakon što korisnik završi s dodavanjem proizvoda u košaricu i odluči dovršiti kupnju. Ovaj proces prolazi kroz nekoliko ključnih koraka:

- Učitavanje detalja za dostavu: Potrebni detalji za dostavu već pohranjeni u korisničkom računu preuzimaju se procesu stvaranja narudžbe.
- Odabir načina plaćanja: Prije stvaranja narudžbe, korisnik mora odabrati način plaćanja. Korisnik ne može nastaviti bez odabira načina plaćanja.
- Izračun troškova dostave i ukupne cijene: Na temelju ukupne vrijednosti artikala u košarici, izračunavaju se eventualni troškovi dostave. Ako je iznos u košarici iznad 50 EUR, dostava može biti besplatna. Ukupna cijena, koja uključuje proizvode i dostavu, jasno se prikazuje korisniku prije konačne potvrde narudžbe.
- Potvrda narudžbe: Nakon što su svi detalji uneseni i potvrđeni, korisnik ima opciju da potvrdi narudžbu. Klikom na "Potvrdi narudžbu", sustav stvara narudžbu i sprema je u bazu podataka. Pogled za stvaranje narudžbe prikazan je na slici 4.18..

```

def create_order_view(request):
    if request.method == 'POST':
        cart = get_object_or_404(Cart, user=request.user.account)
        cart_items = cart.items_in_cart.all()

        payment_method = request.POST.get('payment_method')

        if not payment_method:
            messages.error(request, 'Molimo vas da odaberete način plaćanja.')
            return redirect('webshop:cart')

        order = Order.objects.create(
            account=request.user.account,
            status='IP',
            payment_method=payment_method
        )

        for cart_item in cart_items:
            size_name = cart_item.size.size.name if cart_item.size else ''
            OrderItem.objects.create(
                order=order,
                product=cart_item.product,
                size=size_name,
                quantity=cart_item.quantity
            )
        cart.items_in_cart.all().delete()

        return redirect('webshop:order_details', order_id=order.id)

    return redirect('webshop:cart')

```

Sl. 4.18. Pogled za stvaranje narudžbe

4.11. Korisnički profil

Registrirani korisnici mogu pristupiti i ažurirati svoje osobne podatke unutar svoga korisničkog profila. Na slici 4.19. prikazan je pogled za prikaz i ažuriranje osobnih podataka.

```

def personal_data_view(request):
    user_account = Account.objects.get(user=request.user)

    if request.method == 'POST':
        user_account.name = request.POST.get('name')
        user_account.surname = request.POST.get('surname')
        user_account.phone = request.POST.get('phone')
        user_account.email = request.POST.get('email')
        user_account.city = request.POST.get('city')
        user_account.postal_number = request.POST.get('postal_number')
        user_account.street = request.POST.get('street')
        user_account.house_number = request.POST.get('house_number')
        user_account.iban = request.POST.get('iban')
        user_account.bank = request.POST.get('bank')

        user_account.save()
        return redirect('webshop:personal_data')

    return render(request, 'webshop/personal_data.html', {'user_account': user_account})

```

Sl. 4.19. Pogled za prikaz i mijenjanje osobnih podataka

Osim pregleda svojih podataka, korisnici mogu pregledati povijest svojih narudžbi, kao i detalje o svakoj narudžbi. Na slici 4.20. prikazan je prikaz povijesti narudžbi, dok je na slici 4.21. prikazan prikaz detalja o narudžbi. Također, korisnici unutar svoga profila imaju poveznice na informacije o dostavi, povratima, zamjenama i načinima plaćanja.

```

def orders_history_view(request):
    orders = Order.objects.filter(account=request.user.account).order_by('-date_ordered')

    for order in orders:
        order.total_price = sum(item.total_price() for item in order.items_in_order.all())

    context = {
        'orders': orders
    }

    return render(request, 'webshop/orders_history.html', context)

```

Sl. 4.20. Pogled za prikaz povijesti narudžbi

```

def order_details_view(request, order_id):
    order = get_object_or_404(Order, id=order_id)
    order_items = order.items_in_order.all()

    total_price = sum(item.total_price() for item in order_items)

    context = {
        'order': order,
        'order_items': order_items,
        'total_price': total_price,
    }
    return render(request, 'webshop/order_details.html', context)

```

Sl. 4.21. Pogled za prikaz detalja o narudžbi

4.12. Administrator

Administratorski panel u aplikaciji omogućava administratorima jednostavno upravljanje sadržajem i korisnicima kroz web sučelje. Django generira administratorski panel na temelju modela definiranih u aplikaciji, a *admin.py* datoteka omogućava prilagodbu i konfiguraciju ovog panela. Na slikama 4.22. i 4.23. prikazana je *admin.py* datoteka. Ključne funkcionalnosti administratora su:

- Registracija modela: Modeli se registriraju u administratorskom panelu koristeći dekorator (engl. *decorator*) *@admin.register* ili funkciju *admin.site.register()*.
- Prilagodba prikaza: Kroz *list_display* određuju se polja koja će se prikazivati u popisu modela.
- Pretraživanje i filtriranje: Pomoću *search_fields* i *list_filter* omogućeno je poboljšanje pretraživanja i filtriranja podataka.
- *Inline* modeli: Omogućuju upravljanje povezanim modelima unutar glavnog modela.
- Postavke samo za čitanje: *readonly_fields* omogućuju postavljanje polja koja nisu dostupna za uređivanje.


```

class ProductVariationImageInline(admin.TabularInline):
    model = ProductVariationImage
    extra = 1

class ProductVariationSizeInline(admin.TabularInline):
    model = ProductVariationSize
    extra = 1

@admin.register(Account)
class AccountAdmin(admin.ModelAdmin):
    list_display = ('user', 'name', 'surname', 'email', 'phone')
    search_fields = ('name', 'surname', 'email')

@admin.register(Category)
class CategoryAdmin(admin.ModelAdmin):
    list_display = ('name', 'slug')
    prepopulated_fields = {'slug': ('name',)}

@admin.register(SubCategory)
class SubCategoryAdmin(admin.ModelAdmin):
    list_display = ('name', 'category', 'slug')
    list_filter = ('category',)
    prepopulated_fields = {'slug': ('name',)}

@admin.register(Color)
class ColorAdmin(admin.ModelAdmin):
    list_display = ('name', 'slug')
    search_fields = ('name', 'slug')

@admin.register(Size)
class SizeAdmin(admin.ModelAdmin):
    list_display = ('name',)
    search_fields = ('name',)

@admin.register(Product)
class ProductAdmin(admin.ModelAdmin):
    list_display = ('name', 'category', 'subcategory', 'price', 'created_at', 'reference_number')
    search_fields = ('name', 'reference_number')
    prepopulated_fields = {'slug': ('name',)}
    list_filter = ('category', 'subcategory')

@admin.register(ProductVariation)
class ProductVariationAdmin(admin.ModelAdmin):
    list_display = ('product', 'color', 'main_image')
    list_filter = ('product', 'color')
    search_fields = ('product__name', 'color__name')
    inlines = [ProductVariationImageInline, ProductVariationSizeInline]

```

Sl. 4.22. Prvi dio *admin.py* datoteke

```

class CommentAdmin(admin.ModelAdmin):
    list_display = ('poster', 'product', 'text', 'created_at')

admin.site.register(Comment, CommentAdmin)

class OrderItemInline(admin.TabularInline):
    model = OrderItem
    extra = 1
    fields = ('product', 'size', 'quantity', 'total_price')
    readonly_fields = ('total_price',)

class OrderAdmin(admin.ModelAdmin):
    list_display = ('order_number', 'account', 'status', 'date_ordered', 'payment_method')
    list_filter = ('status', 'payment_method', 'date_ordered')
    search_fields = ('order_number', 'account__name', 'account__surname')
    readonly_fields = ('order_number', 'date_ordered', 'payment_method')
    inlines = [OrderItemInline]

admin.site.register(Order, OrderAdmin)

```

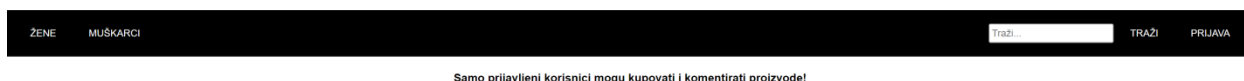
Sl. 4.23. Drugi dio *admin.py* datoteke

5. KORIŠTENJE WEB APLIKACIJE

U ovom poglavlju detaljno će biti objašnjene mogućnosti i funkcionalnosti dostupne različitim vrstama korisnika u web aplikaciji. Objasniti će se uloga gosta, registriranog korisnika i administratora, naglašavajući specifične opcije i prava koja su dostupna svakom tipu korisnika. Također ćemo prikazati sučelja aplikacije, kako bismo vizualno prikazali kako se korisnici mogu kretati kroz aplikaciju i koristiti njene funkcionalnosti.

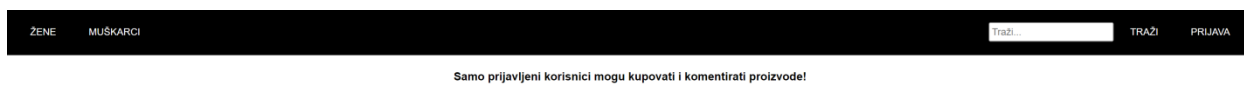
5.1. Gost

Kao gost, korisnik ima pristup osnovnim funkcionalnostima web aplikacije, ali s ograničenjima u usporedbi s registriranim korisnicima. Navigacijska traka gosta (vidi sliku 5.1.) pruža manje mogućnosti od navigacijske trake registriranih korisnika. Gost ima poruku unutar svake stranice aplikacije da se mora prijaviti kako bi mogao naručivati i komentirati. Gost nema poveznice na profil niti na košaricu.



Sl. 5.1. Navigacijska traka gosta

Gost ima poveznicu za prijavu, ali se ne može prijaviti dok se ne registrira. Prijava za gosta može se vidjeti na slici 5.2..



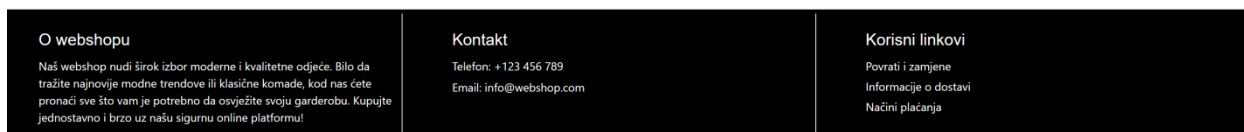
Prijava

Email:

Lozinka:

Prijavi se

Nemate račun? [Registrirajte se](#)



Sl. 5.2. Prikaz stranice za prijavu

Prikaz stranice za registraciju prikazan je na slici 5.3. .

ŽENE MUŠKARCI TRAJI PRIJAVA

Samo prijavljeni korisnici mogu kupovati i komentirati proizvode!

Registracija

Ime:

Prezime:

Email:

Grad:

Poštanski broj:

Adresa:

Kućni broj:

Telefon:

Lozinka:

Potvrdite lozinku:

O webshopu

Naš webshop nudi širok izbor moderne i kvalitetne odjeće. Bilo da tražite najnovije modne trendove ili kvalitetne bazične stvari, mi ćemo vam pružiti sve što vam je potrebno da svojih želja i potreba. Kupujte jednostavno i brzo uz našu sigurnu online platformu!

Kontakt

Telefon: +123 456 789
Email: info@webshop.com

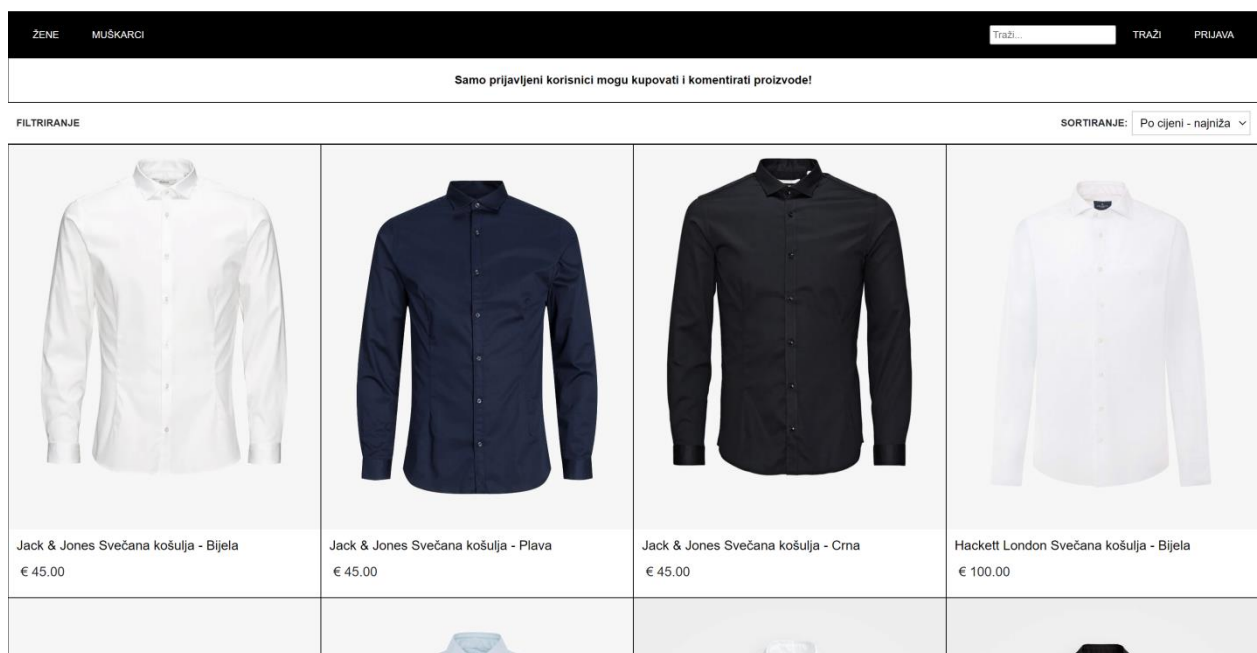
Korisni linkovi

Povratni i zamjene
Informacije o dostavi
Našim plaćanjima

Sl. 5.3. Prikaz stranice za registraciju

Gost može raditi sljedeće stvari:

- Pregled artikala: Gost može pregledavati sve dostupne proizvode u web aplikaciji. To uključuje pristup glavnim kategorijama i potkategorijama proizvoda, kao i mogućnost pregledavanja liste proizvoda unutar svake kategorije. Na slici 5.4. prikazan je katalog proizvoda za gosta.



Sl. 5.4. Prikaz kataloga za goste

- Pretraživanje: Gost može pretraživati proizvode u aplikaciji.

- Filtriranje i sortiranje: Mogućnostima filtriranja i sortiranja dostupne su gostu: Na slici 5.5. vide se mogućnosti filtriranja, dok su na slici 5.6. prikazane mogućnosti sortiranja.

FILTRIRANJE

Boje

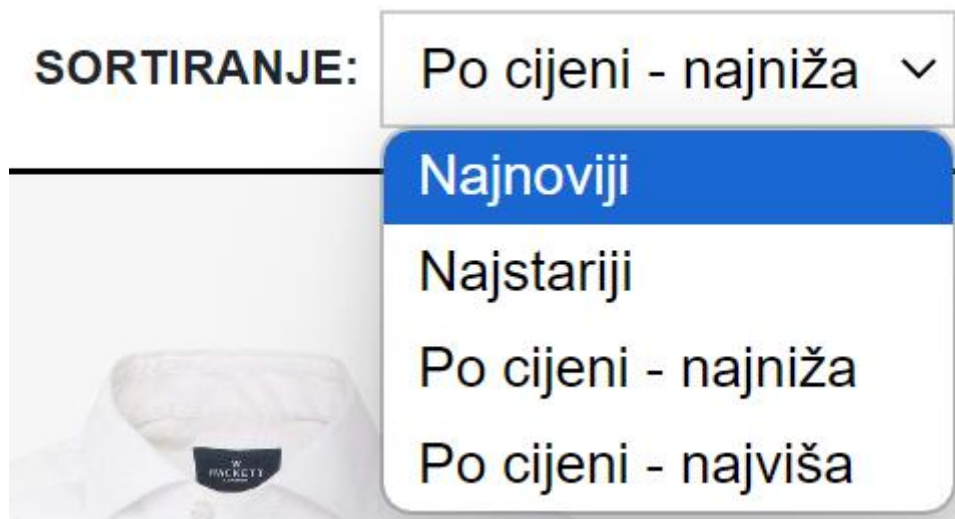
- Bijela
- Plava
- Crna
- Zelena
- Bež
- Sky Plava
- Khaki

Veličine

- XS
- S
- M
- L
- XL
- XXL

Primijeni

Sl. 5.5. Prikaz filtra

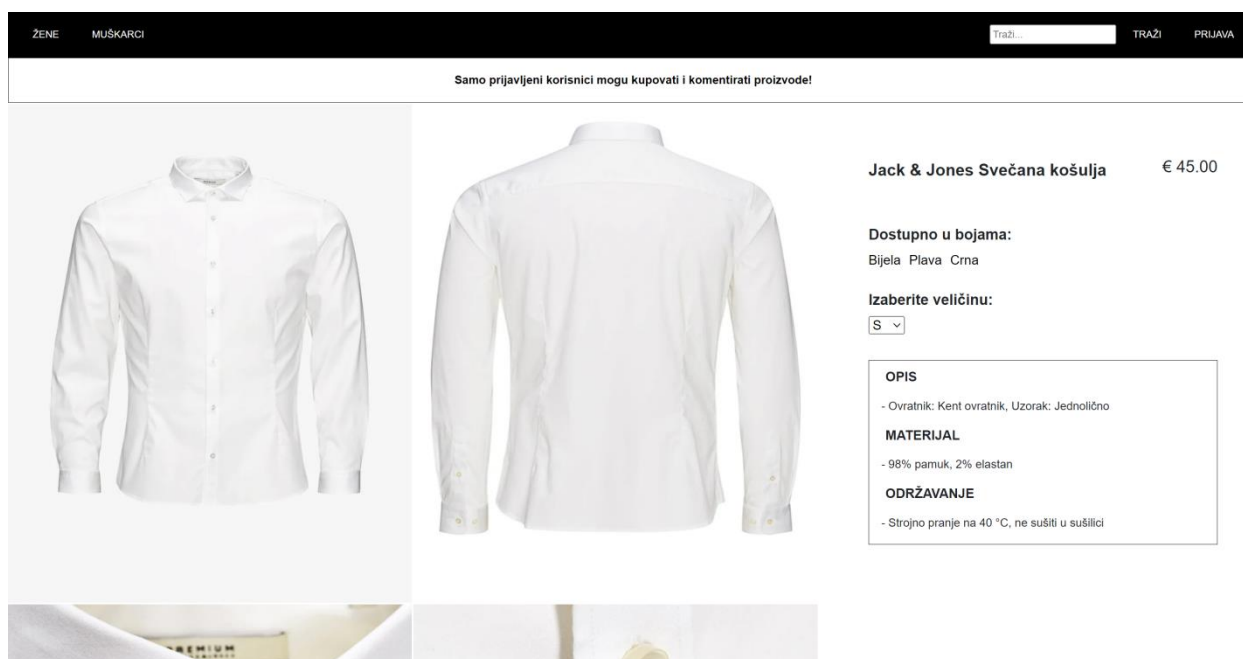


Sl. 5.6. Prikaz mogućnosti sortiranja

- **Detalji o artiklima:** Kada gost klikne na određeni proizvod, može vidjeti detaljne informacije o tom proizvodu. Ove informacije obično uključuju slike proizvoda, naziv, cijenu, opis i druge važne karakteristike.
- **Pregled komentara:** Gost može čitati komentare koje su ostavili drugi korisnici na stranici s detaljima proizvoda.

Ograničenja gosta su:

- **Bez košarice:** Gost ne vidi gumb za dodavanje proizvoda u košaricu (vidi sliku 5.7.) niti ima košaricu.



Sl. 5.7. Prikaz detalja artikla za gosta

- Nema mogućnosti komentiranja (vidi sliku 5.8.): Gost ne može ostaviti komentare na proizvode. Samo registrirani korisnici imaju ovu mogućnost.

KOMENTARI

Odlično!

Ivo, 14.09.2024

Jako udobno.

Marko, 14.09.2024

Sl. 5.8. Prikaz komentara za gosta

5.2. Registrirani korisnik

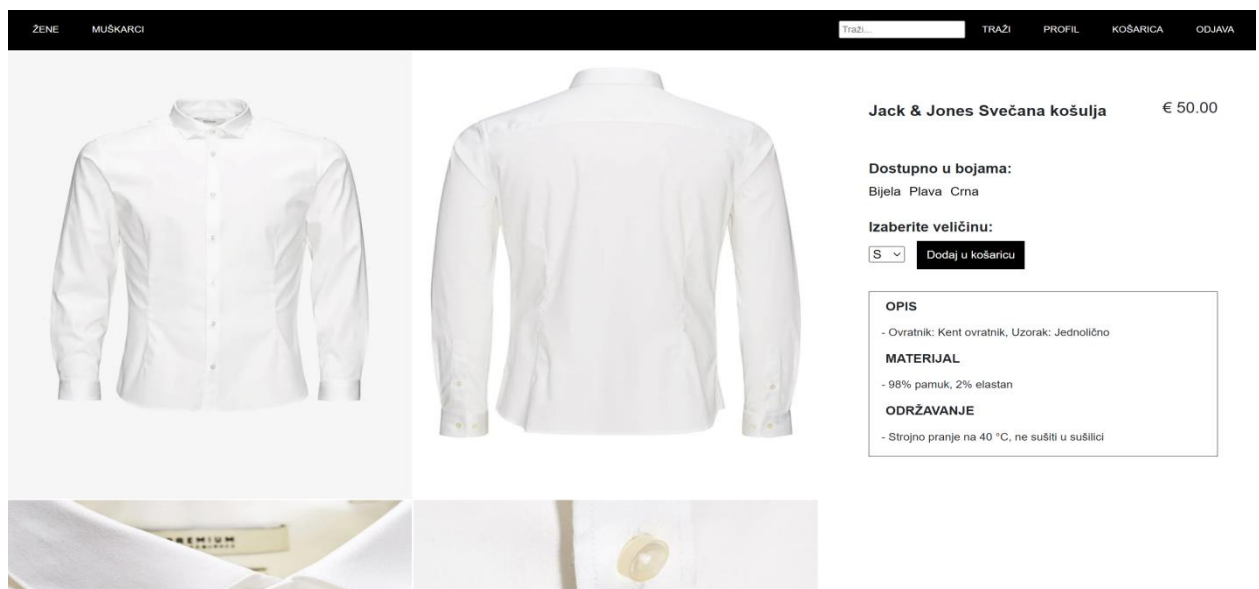
Registrirani korisnici imaju proširene mogućnosti u web aplikaciji u odnosu na goste. U odnosu na gosta, navigacijska traka registriranog korisnika (vidi sliku 5.9.) ima i poveznica na osobni profil i košaricu.



Sl. 5.9. Navigacijska traka registriranog korisnika

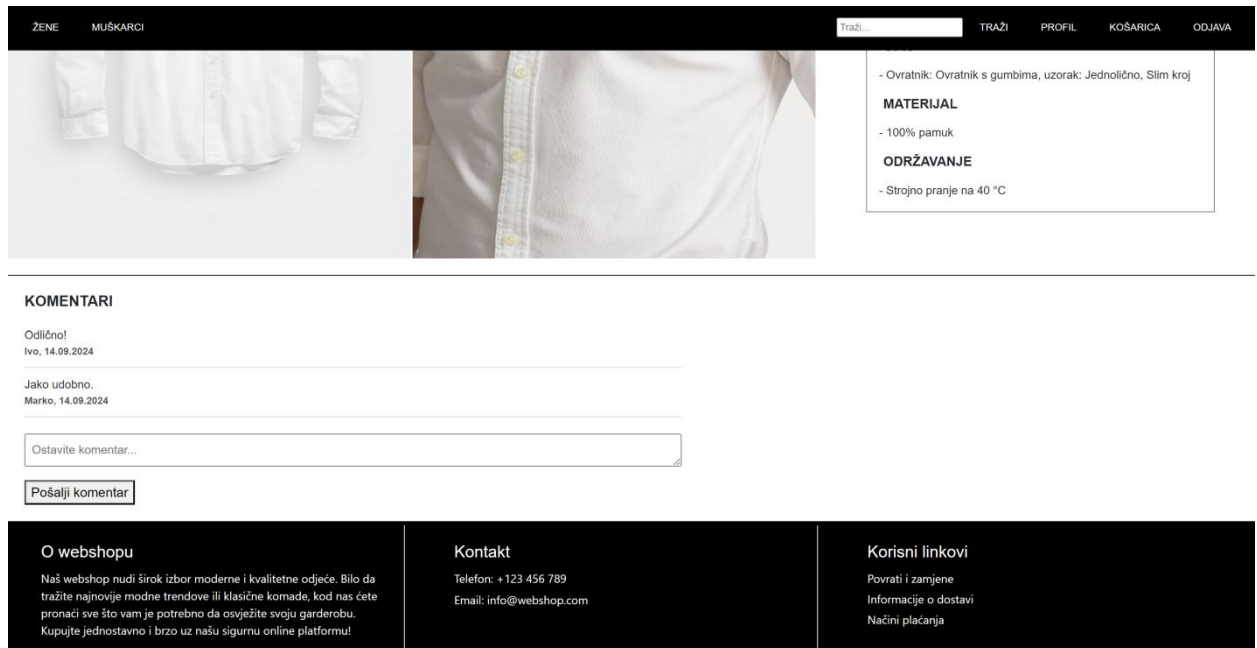
Osim pristupa svim funkcijama dostupnim gostima, registrirani korisnici mogu:

- Pregled proizvoda: Kao i gosti, registrirani korisnici mogu pregledavati sve dostupne proizvode i njihove detalje u web aplikaciji.
- Dodavanje proizvoda u košaricu (vidi sliku 5.10.): Registrirani korisnici mogu dodavati proizvode u svoju košaricu, pregledavati svoju košaricu te istoj mijenjati sadržaj.



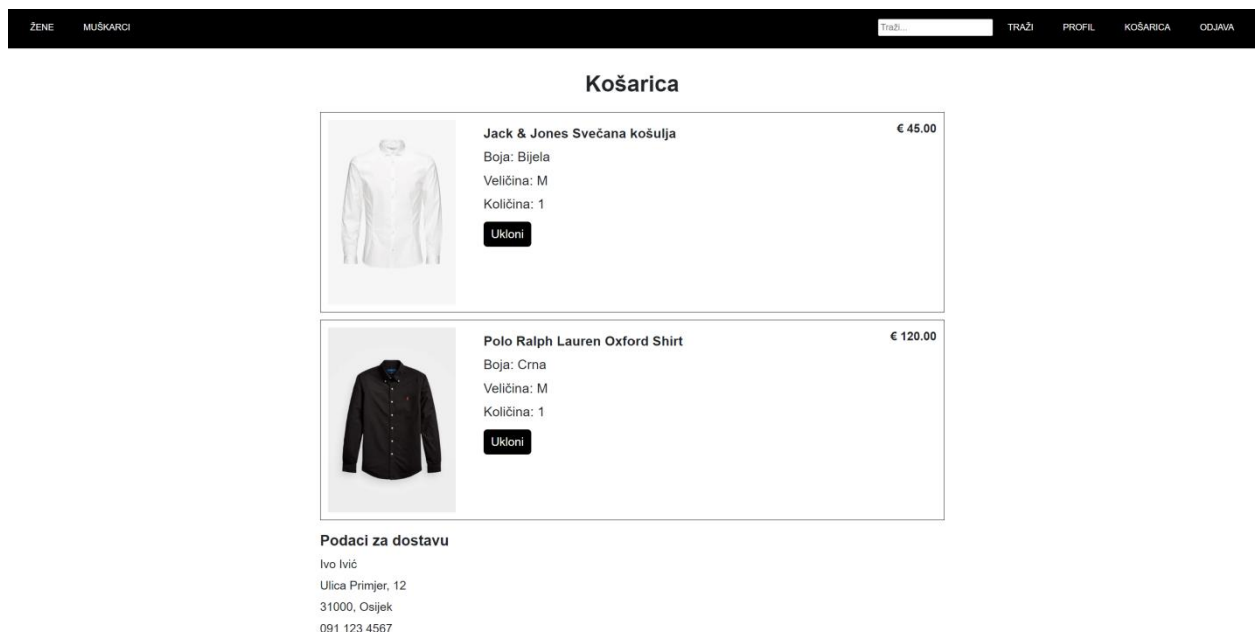
Sl. 5.10. Prikaz detalja o proizvodu za registriranog korisnika s mogućnošću dodavanja artikla u košaricu

- Komentiranje proizvoda (vidi sliku 5.11.): Registrirani korisnici mogu ostaviti komentare na proizvode.



Sl. 5.11. Komentari za registriranog korisnika

Primjer popunjene košarice može se vidjeti na slikama 5.12. i 5.13..



Sl. 5.12. Prvi dio slike košarice registriranog korisnika

ivo@gmail.com

Detalji
 Međuzbroj: € 165.00
 Dostava: € 0
Ukupno: € 165.00

Metoda plaćanja

Plaćanje pouzecom

- Internetsko bankarstvo (onemogućeno)

Naruči

<p>O webshopu</p> <p>Naš webshop nudi širok izbor moderne i kvalitetne odjeće. Bilo da tražite najnovije modne trendove ili klasične komade, kod nas ćete pronaći sve što vam je potrebno da osvježite svoju garderobu. Kupujte jednostavno i brzo uz našu sigurnu online platformu!</p>	<p>Kontakt</p> <p>Telefon: +123 456 789 Email: info@webshop.com</p>	<p>Korisni linkovi</p> <p>Povrati i zamjene Informacije o dostavi Načini plaćanja</p>
---	---	--

Sl. 5.13. Drugi dio slike košarice registriranog korisnika

- Profil i osobni podatci (vidi slike 5.14 i 5.15.): Registrirani korisnici mogu pregledavati i uređivati svoje osobne podatke.

ŽENE MUŠKARCI TRAŽI PROFIL KOŠARICA ODJAVA

Osobni podatci

Ime:

Prezime:

Telefon:

Email:

Podaci za dostavu

Poštanski broj:

Mjesto/Grad:

Ulica:

Kućni broj:

Sl. 5.14. Prvi dio prikaza osobnih podataka korisnika

Kućni broj:

Bankovni podatci

Unesite ili uredite svoje bankovne podatke kako biste mogli primiti povrat za narudžbe plaćene pouzecom.

IBAN:

Banka:

Spremi promjene

<p>O webshopu</p> <p>Naš webshop nudi širok izbor moderne i kvalitetne odjeće. Bilo da tražite najnovije modne trendove ili klasične komade, kod nas ćete pronaći sve što vam je potrebno da osvježite svoju garderobu. Kupujte jednostavno i brzo uz našu sigurnu online platformu!</p>	<p>Kontakt</p> <p>Telefon: +123 456 789 Email: info@webshop.com</p>	<p>Korisni linkovi</p> <p>Povrati i zamjene Informacije o dostavi Načini plaćanja</p>
---	---	--

Sl. 5.15. Drugi dio prikaza osobnih podataka korisnika

- **Povijest narudžbi:** Registrirani korisnici mogu pregledavati povijest svojih narudžbi (vidi sliku 5.16.). Ovdje mogu vidjeti sve prethodne narudžbe, uključujući datume narudžbi, stavke, količine i ukupne iznose.

ZENE MUŠKARCI
Traži:
TRAŽI
PROFIL
KOŠARICA
ODJAVA

Povijest narudžbi

Broj narudžbe: IIVEUCJJ 14. 09. 2024 U pripremi Detalji € 100.00	Broj narudžbe: TU4G7KTY 14. 09. 2024 Poslano Detalji € 45.00
Broj narudžbe: 29GQE6D2 14. 09. 2024 Dostavljeno Detalji € 90.00	Broj narudžbe: W4J5NYMQ 14. 09. 2024 Dostavljeno Detalji € 120.00
Broj narudžbe: DUKWM97L 14. 09. 2024 Dostavljeno Detalji € 220.00	Broj narudžbe: MGJB3R62 14. 09. 2024 Odbijeno Detalji € 45.00

O webshopu

Naš webshop nudi širok izbor moderne i kvalitetne odjeće. Bilo da tražite najnovije modne trendove ili klasične komade, kod nas ćete pronaći sve što vam je potrebno da osvežite svoju garderobu. Kupujte jednostavno i brzo uz našu sigurnu online platformu!

Kontakt

Telefon: +123 456 789
Email: info@webshop.com

Korisni linkovi

[Povrati i zamjene](#)
[Informacije o dostavi](#)
[Načini plaćanja](#)

Sl. 5.16. Prikaz povijesti narudžbi

- **Detalji narudžbe:** Kada pregledavaju povijest narudžbi, registrirani korisnici mogu pregledati detalje svake narudžbe. Ovo uključuje sve informacije o naručenim proizvodima, njihovim količinama, cijenama, statusima isporuke i drugim važnim podacima. Primjer narudžbe može se vidjeti na slici 5.17..

ZENE MUŠKARCI
Traži:
TRAŽI
PROFIL
KOŠARICA
ODJAVA

Broj narudžbe: T2BYBIBA

Status: U pripremi

Datum kupnje

14.09.2024

Podaci za dostavu

Ivo Ivić
Ulica Primjer 12, 31000 Osijek
091 123 4567
ivo@gmail.com

Način plaćanja:

Plaćanje pouzećem

Ukupno: € 100.00

Stavke narudžbe

Hackett London Svečana košulja **€ 100.00**

Bijela
M
Količina: 1

O webshopu

Naš webshop nudi širok izbor moderne i kvalitetne odjeće. Bilo da tražite najnovije modne trendove ili klasične komade, kod nas ćete pronaći sve što vam je potrebno da osvežite svoju garderobu. Kupujte jednostavno i brzo uz našu sigurnu online platformu!

Kontakt

Telefon: +123 456 789
Email: info@webshop.com

Korisni linkovi

[Povrati i zamjene](#)
[Informacije o dostavi](#)
[Načini plaćanja](#)

Sl. 5.17. Prikaz narudžbe

5.3. Administrator

Uloga administratora u aplikaciji ima ključnu važnost jer omogućava upravljanje svim aspektima aplikacije i osigurava pravilno funkcioniranje aplikacije. Administrator ima pristup različitim funkcionalnostima koje omogućavaju učinkovito upravljanje sadržajem i korisnicima. Evo što administrator može raditi:

- Dodavanje, uređivanje i brisanje proizvoda: Administrator može dodavati nove proizvode, mijenjati postojeće proizvode ili ih ukloniti iz baze podataka. To uključuje unos naziva proizvoda, cijene, opisa, kategorije, potkategorije, boje i veličine.
- Upravljanje varijacijama proizvoda: Administrator može dodavati različite varijacije proizvoda, boje i veličine te upravljati slikama koje se prikazuju za svaku varijaciju.
- Dodavanje i uređivanje kategorija i potkategorija: Administrator može dodavati nove kategorije i potkategorije proizvoda te ih uređivati ili brisati. Ovo omogućava organiziranje proizvoda na logičan način i olakšava korisnicima pretraživanje.
- Pregled i uređivanje podataka korisnika: Administrator može pregledavati informacije o korisnicima, uključujući njihove osobne podatke i povijest narudžbi. Također može uređivati podatke korisnika i, ako je potrebno, brisati korisničke račune.
- Praćenje i uređivanje statusa narudžbi: Administrator može pregledavati sve narudžbe koje su napravljene kroz aplikaciju te ih sortirati prema njihovom statusu . Također može mijenjati status narudžbi.
- Manipulacija komentara: Administrator može pregledavati komentare koje su korisnici ostavili na proizvodima. Također može odobravati ili uklanjati komentare prema potrebama i pravilima aplikacije.
- Admin sučelje (vidi sliku 5.18.): Django omogućava kreiranje admin sučelja koje je jednostavno za korištenje i pruža pristup svim gore navedenim funkcionalnostima. Administrator može koristiti ovo sučelje za obavljanje svih navedenih zadataka bez potrebe za direktnim radom s bazom podataka ili drugim složenim alatima.

AUTHENTICATION AND AUTHORIZATION	
Groups	+ Add
Users	+ Add
WEBSHOP	
Accounts	+ Add
Categorys	+ Add
Colors	+ Add
Comments	+ Add
Orders	+ Add
Product variations	+ Add
Products	+ Add
Sizes	+ Add
Sub categorys	+ Add

Sl. 5.18. Prikaz svih modela kojima administrator može manipulirati

6. ZAKLJUČAK

Ovim završnim radom izrađena je potpuno funkcionalna web aplikacija za prodaju odjeće koja zadovoljava sve funkcionalnosti opisane u postavljenom zadatku. Web aplikacija omogućava različite tipove korisnika – goste, registrirane korisnike i administratora – pri čemu su njihove mogućnosti prilagođene njihovim potrebama i razinama pristupa.

Gost može pregledavati proizvode, pregledavati detalje o proizvodima, čitati komentare, ali ne može dodavati proizvode u košaricu niti komentirati proizvode. Registrirani korisnik ima proširene mogućnosti u odnosu na gosta. Osim pregleda proizvoda i njihovih detalja, može dodavati proizvode u košaricu, komentirati proizvode, mijenjati količinu proizvoda u košarici te izvršiti narudžbu. Također, korisnik može pregledavati i uređivati svoj profil, pratiti povijest svojih narudžbi i dobivati obavijesti o stanju svojih narudžbi. Ovim funkcionalnostima, aplikacija omogućava registriranim korisnicim cjelokupni proces kupnje od odabira proizvoda do završetka narudžbe, sve unutar korisničkog računa. Administrator upravlja svim modelima u bazi podataka. Ključnu ulogu administrator ima u procesiranju narudžbi zato što može vidjeti sve nove i završene narudžbe te im može mijenjati status (npr. poslano, isporučeno). Ova razina pristupa omogućava učinkovito vođenje webshopa i osigurava kontrolu nad cijelim poslovnim procesom.

Aplikacija je izrađena korištenjem Django web okvira temeljenog na Pythonu. Za izradu korisničkog sučelja aplikacije korišten je HTML za strukturiranje stranica, dok su CSS i Bootstrap korišteni za stiliziranje stranica i organizaciju elemenata na stranici. Baza podataka aplikacije oslanja se SQLite bazu podataka, u koju su proizvodi, korisnici i narudžbe pohranjeni. Administrator putem Django administratorskog sučelja može dodavati nove proizvode, što omogućava fleksibilno i jednostavno upravljanje sadržajem aplikacije.

Trenutna aplikacija može biti unaprijeđena brojnim načinima, kao što su dodavanje novih kategorija i potkategorija, podrška za više načina plaćanja (kartice, PayPal), te omogućavanje gostima naručivati uz ručni unos podataka za svaku narudžbu.

Web aplikacija pruža kompletno rješenje za prodaju odjeće putem interneta, omogućavajući korisnicima jednostavan proces kupnje od pregleda proizvoda, kreiranja narudžbi, do završetka procesa plaćanja. Na taj način, aplikacija nudi bogato korisničko iskustvo te učinkovito upravljanje web trgovinom za administratore.

LITERATURA

- [1] Mango, dostupno na: <https://shop.mango.com/hr/hr/h/muskarci> [Datum posjete: 28.06.2024.]
- [2] Zara, dostupno na: <https://www.zara.com/hr/> [Datum posjete: 28.06.2024.]
- [3] ABOUTYOU, dostupno na: <https://www.aboutyou.hr/?gender=male> [Datum posjete: 28.06.2024.]
- [4] Django: The web framework for perfectionists with deadlines, dostupno na: <https://www.djangoproject.com/> [Datum posjete: 28.06.2024.]
- [5] Django: The web framework for perfectionists with deadlines, Overview, dostupno na: <https://www.djangoproject.com/start/overview/> [Datum posjete: 28.06.2024.]
- [6] W3Schools, dostupno na: https://www.w3schools.com/django/django_intro.php [Datum posjete: 28.06.2024.]
- [7] SQLite, dostupno na: <https://www.sqlite.org/about.html> [Datum posjete: 29.06.2024.]
- [8] W3Schools, Python Introduction, dostupno na: https://www.w3schools.com/python/python_intro.asp [Datum posjete: 29.06.2024.]
- [9] MDN Web Docs, HTML: HyperText Markup Language, dostupno na: <https://developer.mozilla.org/en-US/docs/Web/HTML> [Datum posjete: 29.06.2024.]
- [10] MDN Web Docs, dostupno na: <https://developer.mozilla.org/en-US/docs/Web/CSS> [Datum posjete: 29.06.2024.]
- [11] MDN Web Docs, What is CSS?, dostupno na: https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS [Datum posjete: 29.06.2024.]
- [12] W3Schools, Bootstrap Get Started, dostupno na: https://www.w3schools.com/bootstrap/bootstrap_get_started.asp [Datum posjete: 29.06.2024.]
- [13] Visual Studio Code, Why Visual Studio Code?, dostupno na: <https://code.visualstudio.com/docs/editor/whyvscode> [Datum posjete: 29.06.2024.]
- [14] Git, dostupno na: <https://git-scm.com/> [Datum posjete: 29.06.2024.]

SAŽETAK

Ovaj rad se bavi izradom web aplikacije za prodaju odjeće, koristeći Django, okvir temeljen na Pythonu, za implementaciju poslovne logike i upravljanje korisnicima, proizvodima i narudžbama. HTML, CSS i Bootstrap korišteni su za dizajn i stilizaciju aplikacijskog sučelja. Cilj aplikacije je osigurati jednostavno i intuitivno korisničko iskustvo te omogućiti učinkovit proces internetske kupovine. Aplikacija omogućuje tri razine korisničkog pristupa: gosti mogu pregledavati proizvode, registrirani korisnici imaju pristup potpunom procesu kupovine, dok administratori upravljaju proizvodima, korisnicima i narudžbama. Kroz ovaj rad, čitatelj će moći upoznati razvoj i funkcioniranje aplikacije, kao i način njenog korištenja, od prijave korisnika i pregleda proizvoda do cjelokupnog procesa kupovine i upravljanja narudžbama. Postignuti rezultati uključuju funkcionalnu web aplikaciju s intuitivnim sučeljem i jednostavnim i pouzdanim procesom naručivanja, s mogućnostima daljnjeg unaprjeđenja.

Ključne riječi: baza podataka, Django, odjeća, postavljanje narudžbe, web trgovina

ABSTRACT

Title: Web application for selling clothes

This paper focuses on developing a web application for selling clothing, utilizing Django, a Python-based framework, for implementing business logic and managing users, products, and orders. HTML, CSS, and Bootstrap were used to design and style the application's user interface. The application aims to ensure a simple and intuitive user experience and facilitate an efficient online shopping process. The application provides three user access levels: guests can browse products, registered users can access the complete purchasing process, and administrators manage products, users, and orders. Through this paper, readers will gain insight into the development and operation of the application, including how it is used from user registration and product browsing to the entire purchasing process and order management. Development results include a functional web application with an intuitive interface and a straightforward and reliable ordering process, with opportunities for further enhancement.

Keywords: database, Django, clothing, placing an order, online store

PRILOZI

1. Izvorni kod web aplikacije, dostupno na: <https://github.com/mariopejic1/ZavrzniRad-Web-aplikacija-za-prodaju-odjece>