

Implementacija preporučiteljskog sustava utemeljenog na strujećim podacima

Grbeša, Tomislav

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:905382>

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

Download date / Datum preuzimanja: **2025-03-13**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 561

**IMPLEMENTACIJA PREPORUČITELJSKOG SUSTAVA
UTEMELJENOG NA STRUJEĆIM PODACIMA**

Tomislav Grbeša

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 561

**IMPLEMENTACIJA PREPORUČITELJSKOG SUSTAVA
UTEMELJENOG NA STRUJEĆIM PODACIMA**

Tomislav Grbeša

Zagreb, lipanj 2024.

DIPLOMSKI ZADATAK br. 561

Pristupnik: **Tomislav Grbeša (0036514325)**
Studij: Računarstvo
Profil: Programsko inženjerstvo i informacijski sustavi
Mentor: prof. dr. sc. Damir Pintar

Zadatak: **Implementacija preporučiteljskog sustava utemeljenog na strujećim podacima**

Opis zadatka:

U modernim informacijskim sustavima kontinuirano se prenose i prikupljaju velike količine podataka, nudeći bogat izvor informacija za personalizirane usluge i preporuke. U skladu s tim, implementacija preporučiteljskog sustava temeljenog na strujećim podacima postaje ključna kako bi se korisnicima pružila relevantna i pravovremena preporuka sadržaja, proizvoda ili usluga. Ovaj pristup omogućuje dinamično prilagođavanje preporuka na temelju trenutnih interesa i ponašanja korisnika, pružajući im personalizirano iskustvo koje odgovara njihovim jedinstvenim preferencijama i potrebama. Zadatak rada je razviti web aplikaciju koja će demonstrirati funkcionalnost preporučiteljskog sustava utemeljenog na strujećim podacima u odabranoj domeni. Potrebno je objasniti arhitekturu takvog sustava i razviti pilotsku implementaciju koja će sadržavati sve nužne segmente te na studijskom primjeru prikazati učinkovitost konačnog rješenja.

Rok za predaju rada: 28. lipnja 2024.

Sadržaj

| | |
|--|-----------|
| 1. Uvod | 3 |
| 2. Općenito o sustavima za preporučivanje | 4 |
| 2.1. Sustavi temeljeni na sadržaju | 4 |
| 2.2. Sustavi temeljeni na suradničkom filtriranju | 5 |
| 2.3. Hibridni sustavi | 5 |
| 2.4. Primjeri postojećih sustava | 5 |
| 3. Korištene tehnologije | 8 |
| 3.1. Pozadinska aplikacija poslovne logike | 8 |
| 3.1.1. Java | 8 |
| 3.1.2. Spring i Spring Boot | 8 |
| 3.1.3. Spring Web | 9 |
| 3.1.4. Spring AOP | 10 |
| 3.2. Baza podataka | 10 |
| 3.3. Kafka | 10 |
| 3.4. Python | 11 |
| 3.5. Klijentsko sučelje - React | 11 |
| 4. Arhitektura sustava | 12 |
| 4.1. Arhitektura pozadinske aplikacije poslovne logike | 12 |
| 4.2. Preporučiteljski algoritam | 13 |
| 4.2.1. Podatkovni skup | 13 |
| 4.2.2. Transformer arhitektura | 14 |
| 4.2.3. Generiranje preporuka na temelju sadržaja | 15 |

| | |
|---|-----------|
| 4.2.4. Generiranje preporuka suradničkim filtriranjem | 16 |
| 4.3. Baza podataka | 18 |
| 5. Implementacija i upute za korištenje | 23 |
| 6. Zaključak | 34 |
| Literatura | 35 |
| Sažetak | 37 |
| Abstract | 38 |

1. Uvod

Veliki rast digitalnog sadržaja i online usluga doveo je do ogromne količine izbora za potrošače, čineći zadatak pronalaženja relevantnih produkata i sadržaja sve izazovnijim. Sustavi za preporučivanje pojavili su se kao ključni alat za ublažavanje ovog preopterećenja informacijama pružajući korisnicima personalizirane prijedloge sadržaja i proizvoda. Ovi sustavi koriste različite algoritme i izvore podataka kako bi predvidjeli korisničke preference i isporučili prilagođene preporuke, povećavajući zadovoljstvo i angažman korisnika, a vlasnicima sustava donose veću zaradu i prepoznatljivost.

Tradicionalni preporučiteljski sustavi se oslanjaju na procesiranje i analizu cijelog skupa podataka za generiranje preporuka u periodičkim intervalima. Ovakvi sustavi su limitirani u obradi podataka u stvarnom vremenu i prilagodbi na korisnikove interakcije i promjene u preferencijama. Kako se korisnikovi interesi brzo mijenjaju, sve je veća potreba za preporučiteljskim sustavima koji mogu u stvarnom vremenu obraditi događaje i generirati pravovremene i relevantne preporuke.

Ovaj rad se bavi razvojem preporučiteljskog sustava koji je sposoban obrađivati strujeće podatke u realnom vremenu. Sustav je izgrađen korištenjem mikroservisne arhitekture koja se sastoji od klijentske aplikacije, pozadinske aplikacije poslovne logike, brokera poruka (u nastavku teksta broker) i dvije dodatne aplikacije zadužene za izradu preporuka. Svaki od navedenih mikroservisa se može samostalno pokrenuti i međusobno su slabo povezani (engl. loosely coupled) što olakšava implementaciju i testiranje. Rad preporučiteljskog sustava ćemo prikazati u kontekstu web trgovine elektroničkih uređaja.

U početku rada će se predstaviti preporučiteljski sustavi, tipovi preporučiteljskih sustava i prezentirati postojeće sustave za preporučivanje i način kako funkcioniraju. Nadalje, objasnit će se arhitektura sustava, arhitektura pojedinih aplikacija poslovne logike i arhitektura samog preporučiteljskog sustava. Na kraju, prikazno je implementirano rješenje s uputama za upotrebu.

2. Općenito o sustavima za preporučivanje

Sustavi za preporučivanje (engl. recommender systems) su dio mnogih digitalnih sustava i pomažu korisnicima u korištenju sustava i pronalaženju informacija ili proizvoda koji ih zanimaju dok vlasnicima sustava donose veću zaradu i prepoznatljivost. Sustave za preporučivanje dijelimo u tri kategorije: sustavi temeljeni na sadržaju (engl. content based recommender systems), sustavi temeljeni na suradničkom filtriranju (engl. collaborative filtering recommender systems) i hibridne sustave koji su kombinacija prethodna dva.

Prvi sustavi za preporučivanje nastali su 90ih godina 20. stoljeća, koristili su se za filtriranje elektroničke pošte, a tada je nastao i pojam suradničkog filtriranja [1]. Početkom 21. stoljeća su nastale naprednije tehnike preporuke gdje je veći fokus stavljen na preporuke po sadržaju. Današnji najbolji preporučiteljski sustavi u velikoj mjeri koriste duboke neuronske mreže za kreiranje modela i kombiniraju dva navedena sustava kako bi maksimizirali prednosti oba pristupa.

2.1. Sustavi temeljeni na sadržaju

Preporučiteljski sustavi temeljeni na sadržaju (engl. content based recommender systems) prikupljaju podatke o aktivnosti korisnika, a preporuku rade na temelju opisnih i metapodataka o proizvodima. Npr. korisnik je kupio određeni proizvod, a algoritam preporučivanja će preporučiti proizvode slične kupljenom proizvodu na temelju podataka o samom proizvodu. Ovaj pristup nam je pogodan ako posjedujemo velike količine podataka o proizvodima kao npr. redatelj, žanr, glumci i opis u kontekstu preporuke filmova.

2.2. Sustavi temeljeni na suradničkom filtriranju

Sustavi temeljeni na suradničkom filtriranju jednako prikupljaju podatke o aktivnosti korisnika, ali za razliku od sustava koji koriste sadržaj za kreiranje preporuke, ovi sustavi koriste same podatke o aktivnosti kako bi pronašli slične korisnike ili proizvode i tako generirali preporuke. Odlika ovih sustava je da su jednostavni za implementaciju i za njihovu implementaciju su samo potrebni podaci o aktivnosti korisnika, što je velika prednost u preporuci proizvoda koji se ne mogu dovoljno dobro opisati ili ne posjedujemo podatke o samim proizvodima.

2.3. Hibridni sustavi

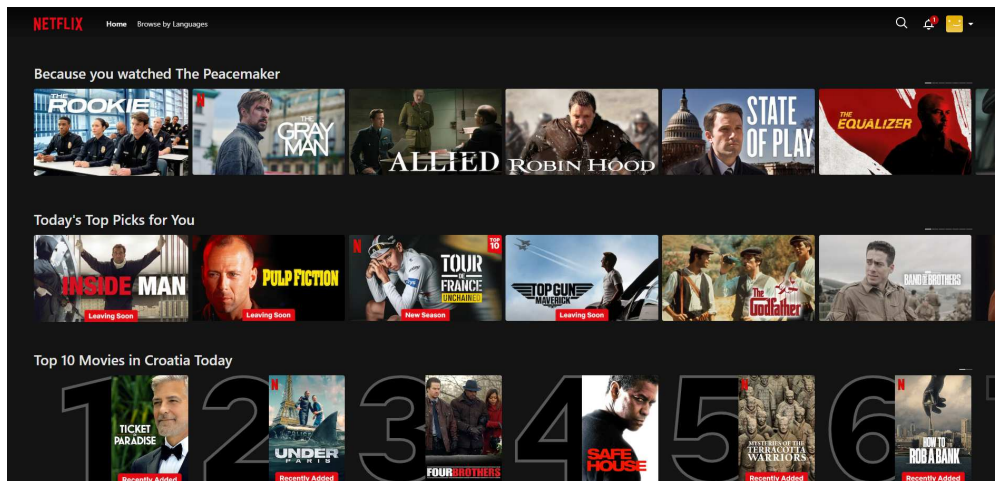
Zbog mana prethodnih načina preporuke, postoje i hibridni sustavi koji kombiniraju oba pristupa. Mane suradničkog filtriranja je U slučaju suradničkog filtriranja, ako nemamo dovoljno podataka o aktivnosti (problem hladnog starta engl. cold start problem), ne možemo kreirati adekvatnu tablicu suradničkog filtriranja jer nam previše podataka nedostaje. Nedostatak sustava temeljenih na sadržaju je tendencija preporuke jednakog tipa proizvoda s kojima je korisnik već imao interakciju, a to dovodi do smanjene diverzifikacije preporuka i nemogućnosti preporuke proizvoda van uspostavljenih preferencija. Hibridni sustav kombinacijom prethodna dva sustava umanjuje njihove nedostatke i omogućuje preporuku od korisnikove prve interakcije sa sustavom i diverzifikaciju preporuka. Implementirani preporučitelj u ovom radu koristi hibridni pristup.

2.4. Primjeri postojećih sustava

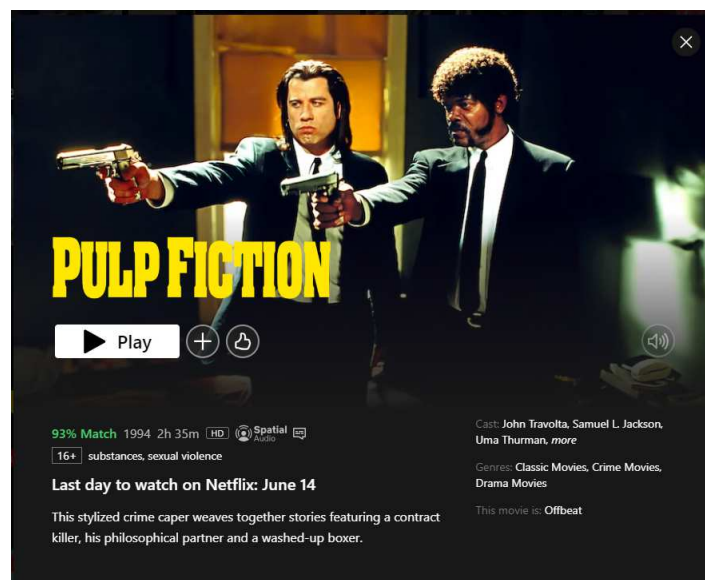
Netflix [2] je web servis čija je glavna funkcionalnost mogućnost gledanja filmova na različitim uređajima. Veliki razlog popularnosti ove aplikacije je razvijeni preporučiteljski sustav i kontinuirano poboljšavanje istog. Netflix trenutno ima najveći broj korisnika ovakvog tipa servisa [3].

Na slici 2.1. prikazan je preporučeni sadržaj u Netflix aplikaciji. Prvi preporučeni redak filmova su svi filmovi koji su sadržajem slični filmu Mirotvorac (engl. The Peacemaker). Druga prikazana kategorija su najbolje preporuke za korisnika po preporučiteljskom algoritmu koji za svaki film određuje vrijednost koliko je korisnikova povijesna aktivnost

i gledanje filmova kompatibilna s tim filmom. Primjer takve preporuke vidljiv je na slici 2.2. Navedenu kategoriju preporuka može se protumačiti kao preporuke temeljene na suradničkom filtriranju. Treća prikazana kategorija su trenutno popularni filmovi poređani po gledanosti. Prilikom prve prijave u aplikaciju, korisnik može odabrati filmove koji mu se sviđaju, a tim odabirom se rješava problem hladnog starta i algoritam može odmah stvoriti preporuke za korisnika.

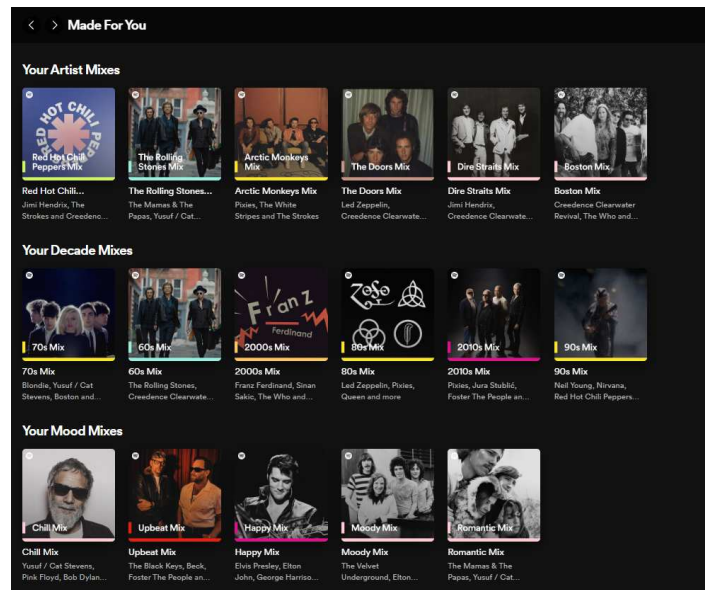


Slika 2.1. Prikaz preporučanih filmova u Netflix aplikaciji



Slika 2.2. Prikaz detalja filma s podatkom korisnikove kompatibilnosti s filmom

Spotify je platforma za slušanje i preuzimanje glazbe i podcasta. Kao i Netflix, Spotify ima preporučiteljski sustav čiji je glavni cilj kreirati jedinstveno iskustvo za svakog korisnika [4]. Na slici 2.3. prikazane su generirane preporuke u Spotify aplikaciji. Preporučiteljski algoritam u aplikaciji aktivno koristi hibridni pristup za kreiranje preporuka, a preporučene pjesme se nalaze u popisima za reprodukciju povezane određenom kategorijom npr. žanr glazbe.



Slika 2.3. Prikaz preporučениh popisa za reprodukciju u Spotify aplikaciji

3. Korištene tehnologije

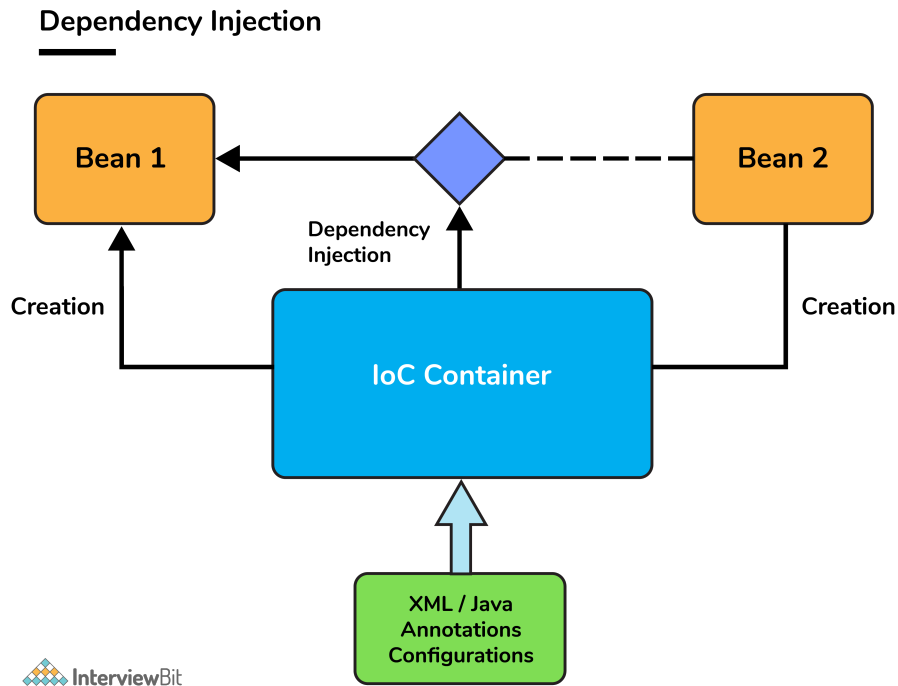
3.1. Pozadinska aplikacija poslovne logike

3.1.1. Java

Java [5] je objektno orijentirani programski jezik opće namjene nastao sa sloganom: napiši jednom, pokreni bilo gdje. Slogan označava glavnu značajku programskog jezika Java, a to je da su razvojni inženjeri mogli napisati program na bilo kojem računalu, kompajlirati kod i izvoditi ga na bilo kojem uređaju koji je imao instaliranu Javinu virtualnu mašinu (JVM). Iako je jezik nastao prije skoro 30 godina, i dalje je popularan za razvoj web aplikacija, nativnih aplikacija i aplikacija interneta stvari (engl. Internet of things - IoT). Razlog zašto se jezik zadržao je zbog zajednice i ekosustava sagrađenog oko jezika kao i velikog broja aplikacija koje su već napisane i potrebno ih je održavati.

3.1.2. Spring i Spring Boot

Spring [6] je radni okvir napisan u Javi koji olakšava razvoj Java aplikacija. Glavni mehanizam koji se koristi u Springu je smanjenje međuovisnosti komponenti korištenjem kontejnera inverzije (engl. Inversion of Control container, u nastavku teksta IoC kontejner) čija je arhitektura prikazana na slici 3.1. Zadaća IoC kontejnera je briga o životnom ciklusu i konfiguraciji objekata koje aplikacija konzumira. Objekti definiraju svoje ovisnosti mehanizmom injekcije ovisnosti (engl. dependency injection) kroz konstruktore ili kroz parametre metode tvornice koje Spring razrješava prilikom kreiranja objekata. Ovim načinom postignuta je inverzija ovisnosti gdje objekt ne mora sam direktno instancirati svoje ovisnosti nego to za njega radi IoC kontejner.



Slika 3.1. Prikaz arhitekture IoC kontejnera [7]

Spring Boot [8] je projekt unutar Springovog ekosustava koji olakšava i ubrzava razvoj aplikacija tako da pruža potrebne konfiguracije za pokretanje Spring aplikacije, ugrađeni web server i set modula koji sadrže već zadane konfiguracije i sve potrebne ovisnosti. Spring Boot je u ovom radu korišten za izradu glavne aplikacije u kojoj je sadržana poslovna logika i logika za slanje događaja prema brokeru poruka. Neki od korištenih modula su Spring Web, Spring AOP (detaljnije objašnjeni u nastavku), Spring Data JPA koji je zapravo implementacija Java objektno-relacijskog mapiranja itd.

3.1.3. Spring Web

Spring Web je skup modula koji služe za razvoj web aplikacija koji pozadini koriste Spring IoC kontejner. Navedeni razvojni okvir je iznimno popularan za razvoj mikro-servisnih aplikacija baziranih na RESTful (engl. Representational State Transfer) principima i razvoj dinamičkih web aplikacija. Razvijena aplikacija u ovom radu prati RESTful principe što nam omogućava slabu povezanost između poslužitelja i klijenta i mogućnost bržeg i paralelnog razvoja.

3.1.4. Spring AOP

Aspektno orijentirano programiranje (engl. Aspect oriented programming) je tehnika programiranja čestih rutina koje se mogu iskoristiti više puta i na više različitih mjesta u aplikaciji npr. provjera autorizacijskog tokena. Spring AOP modul komplementira Springov IoC kontejner tako da modularizira određene rutine koje su potrebne na više različitih mjesta u sustavu. U konkretnoj implementaciji aplikacije, kreiran je jedan aspekt koji se brine o slanju događaja prema brokeru. Ovakvim načinom implementacije omogućeno je jednostavno dodavanje novih tipova događaja s pomoću Spring anotacija bez izmjene starog koda.

3.2. Baza podataka

U radu je korištena PostgreSQL [9] relacijska baza podataka otvorenog koda i trenutno je jedan od popularnijih relacijskih sustava za upravljanje bazama podataka. Ovaj sustav je odabran zbog lagane integracije sa Spring aplikacijama i postojećeg skupa modula Jave za spajanje na baze podataka. Za integraciju s bazom podataka potrebno je definirati samo tri parametra: URL do baze podataka, korisničko ime i lozinku. Nakon definicije parametara, Spring i Java će u pozadini obaviti kreiranje veze na bazu podataka i uspostaviti mehanizam upravljanja transakcijama.

3.3. Kafka

Apache Kafka [10] je platforma otvorenog koda dizajnirana za procesiranje velike količine strujećih podataka i koristimo ju kao broker poruka u ovom radu. Inicijalno je razvijena za prikupljanje podataka o korisnikovoj aktivnosti prilikom korištenja web aplikacija i to je razlog zašto se platforma koristi u ovom radu, dok se trenutno puno više koristi u procesiranju drugačijih izvora podataka npr. procesiranje podataka sa senzora ili u procesiranju podataka s burze. U kontekstu korištenja, Kafka je poslužitelj dok su ostali sudionici zapravo klijenti. Klijent može biti proizvođač ili potrošač, a oni šalju ili konzumiraju podatke s različitih tema (engl. topic). Kafku u aplikaciji koristimo za prikupljanje podataka o korisnikovoj interakciji sa sustavom koje se kasnije koriste za preporuku sadržaja.

3.4. Python

Python [11] je skriptni programski jezik poznat po jednostavnosti i čitljivosti koda te podržava više različitih paradigmi programiranja. Programski jezik je trenutno popularan u zajednici znanosti o podacima jer sadrži biblioteke koje su optimizirane za treniranje modela algoritmima strojnog učenja. Python je odličan izbor za razvoj skripti za obradu podataka, vizualizaciju podataka i razvoj modela strojnog učenja zbog standardiziranih i optimiziranih biblioteka u sklopu samog programskog jezika. U implementaciji je korištena U radu su napravljene dvije Python aplikacije koje obrađuju događaje s brokera i kreiraju preporuke koje spremaju u bazu podataka.

3.5. Klijentsko sučelje - React

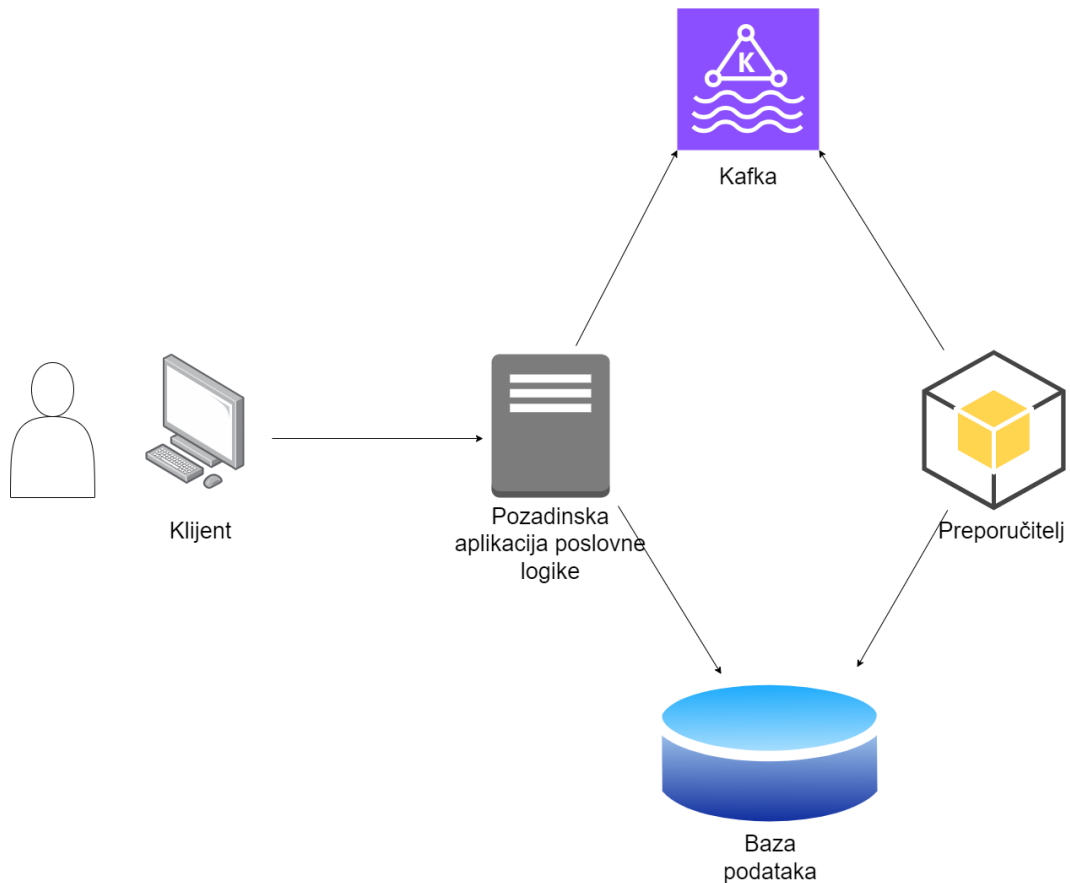
React [12] je JavaScript biblioteka otvorenog koda za kreiranje klijentskih aplikacija. React se koristi u razvoju jednostraničnih aplikacija (engl. single-page application - SPA). Jednostranična aplikacija je tip aplikacije koja dinamički mijenja izgled stranice prilikom korisnikove interakcije umjesto učitavanja cijelih novih stranica s poslužitelja. React koristi arhitekturu baziranu na komponentama (engl. component based architecture) gdje je klijentsko sučelje zapravo sastavljeno od manjih, višekratno iskoristivih komponenti. Svaka komponenta ima svoje stanje (engl. state), a komponenta se ponovno iscrtava na svaku promjenu stanja. Oko biblioteke je organizirana velika zajednica koja je razvila ogroman ekosustav modula koji su podržani unutar Reacta.

4. Arhitektura sustava

Sustav je dizajniran prateći mikroservisnu arhitekturu [13]. Mikroservisna arhitektura je arhitekturni stil koji strukturira aplikacije kao kolekcije servisa koji se mogu nezavisno pokrenuti i međusobno su slabo povezani. Skica arhitekture vidljiva je na slici 4.1. Ulazna točka za korisnika je klijentska aplikacija. Klijentsko sučelje komunicira isključivo s pozadinskom aplikacijom poslovne logike. Pozadinska aplikacija poslovne logike ima dvije ovisnosti: prema bazi podataka i prema brokeru. Pozadinska aplikacija poslovne logike koristi troslojnu arhitekturu koju ćemo više objasniti u nastavku rada. Broker i baza podataka su zasebni servisi koji nemaju ovisnosti prema drugim servisima u sustavu. Preporučiteljska aplikacija napisana u Pythonu je, kao i aplikacija poslovne logike, ovisna o bazi podataka i brokeru. Organizacija arhitekture na ovaj način omogućava jednostavno skaliranje aplikacije i sprječava postojanje kružnih ovisnosti. Dodatna prednost ovakve organizacije sustava je ta da, uz prethodno definiranu funkcijsku specifikaciju, omogućen je paralelni razvoj svakog od servisa.

4.1. Arhitektura pozadinske aplikacije poslovne logike

Razvijena aplikacija koristi troslojnu arhitekturu gdje je prvi sloj prezentacije podataka, drugi sloj poslovne logike i treći sloj pristupa podacima. Klijent vidi i konzumira prezentacijski sloj, a na tom sloju klijent ima prezentaciju podataka i mehanizme za interakciju s podacima. Implementirani prezentacijski sloj je organiziran po REST arhitekturnom stilu. Sloj poslovne logike se nalazi između prezentacijskog sloja i sloja pristupa podacima. U ovom sloju je sadržana logika i poslovna pravila aplikacije. Sloj za pristup podacima se brine o trajnom pohranjivanju podataka i dohvat samih podataka. Prikaz međusobnih odnosa slojeva i poveznice s anotacijama u Springu možemo vidjeti na slici 4.2.



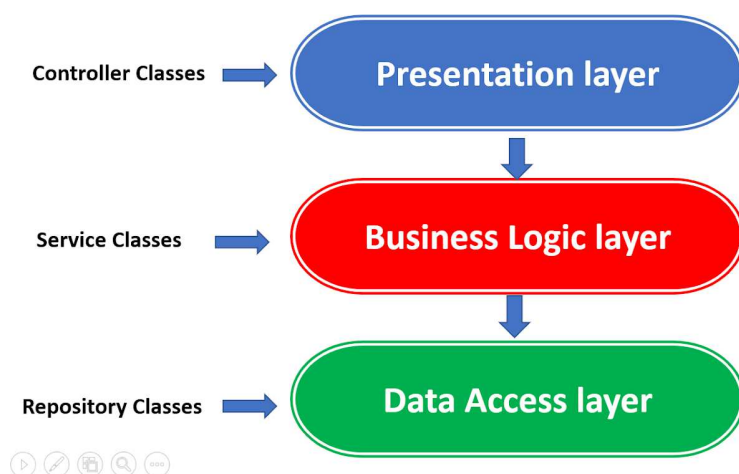
Slika 4.1. Skica arhitekture sustava

4.2. Preporučiteljski algoritam

4.2.1. Podatkovni skup

Podatkovni skup se sastoji od oko 570 milijuna recenzija i skoro 50 milijuna proizvoda prikupljenih tokom perioda od 27 godina. Zbog ogromne količine podataka, odabrani su samo proizvodi pod kategorijom elektronika. Relevantni podaci koji su iskorišteni iz podatkovnog skupa su naziv, cijena, opis i detalji proizvoda, slike proizvoda i recenzije o proizvodima. Odabrane proizvodi su dodatno filtrirani tako što su odabrani samo oni proizvodi koji imaju opis i barem 20 recenzija. Konačni podatkovni skup koji je korišten u implementaciji sadrži 3 tisuće proizvoda i oko 130 tisuća recenzija. Korišteni podaci u ovom radu su preuzeti sa službenih stranica Sveučilišta u San Diegu [15].

Three-Tier (or Three-Layer) Architecture in Spring MVC

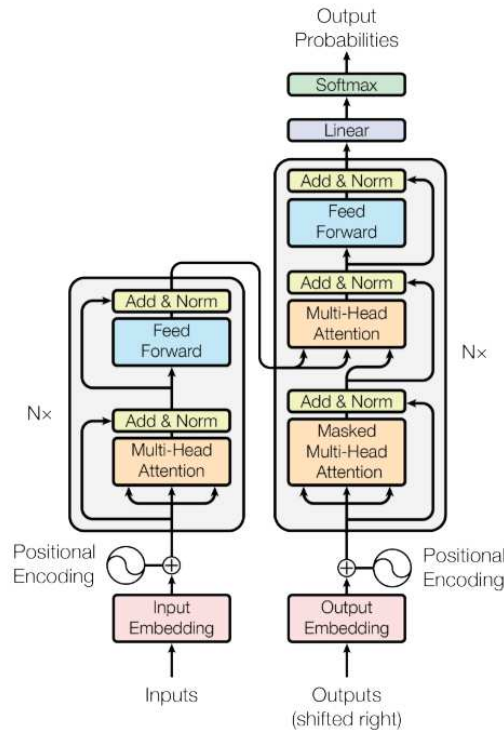


Slika 4.2. Prikaz troslojne arhitekture u Spring radnom okviru [14]

4.2.2. Transformer arhitektura

Attention Is All You Need[16] je rad iz 2017. godine kojeg su objavili zaposlenici Googlea i govori o novoj arhitekturi neuronskih mreža naziva Transformer koja je revolucionizirala područje obrade prirodnog jezika i umjetne inteligencije. Transformer arhitektura na slici 4.3., umjesto konvolucijskih ili povratnih slojeva, koristi posebne slojeve pažnje u neuronskim mrežama kojima mapira ovisnosti između ulaznih elemenata, a razvijena je s namjerom obrade sekvencijalnih podataka s fokusom na efikasnost i paralelizaciju. Model prati enkoder-dekoder strukturu gdje se enkoder sastoji od sloja pažnje i od sloja potpuno povezane mreže dok dekoder prati istu strukturu, ali na svoj ulaz dovodi trenutno generirani izlaz. U navedenom radu [16] su prikazani rezultati gdje je model ostvario puno bolje rezultate strojnog prevođenja za četvrtinu vremena treniranja od prethodno najbolje razvijenog modela. Iako je arhitektura inicijalno zamišljena kao poboljšanje strojnog prevođenja, postala je standard u području obrade prirodnog jezika. U radu je korišten model All-mpnet-base-v2 [17] koji je treniran na više od milijardu testnih primjeraka i to je zapravo nadogradnja predtreniranog BERT modela. BERT (engl. Bidirectional Encoder Representations from Transformers) je model koji koristi Transformer arhitekturu i trenira se na dva različita, ali povezana zadatka: predikcija sljedeće riječi u rečenici i predikcija sljedeće rečenice. All-mpnet-base-v2 model je specifično treniran za zadatke grupiranja tekstova i semantičkog pretraživanja. Svaki testni primjerak

je zapravo par od dva teksta koji su slični, a prilikom treninga model ima zadatak iz cijelog podatkovnog skupa za dani tekst pronaći semantički najbliži tekst. Model prima tekst od maksimalno 384 riječi, a izlaz modela je vektor od 768 varijabli koji tretiramo kao numeričku reprezentaciju proizvoda.



Slika 4.3. Prikaz transformer arhitekture[16]

4.2.3. Generiranje preporuka na temelju sadržaja

Zbog popularnosti i rezultata Transformer arhitekture, u radu je korišten navedeni model za izradu preporuka na temelju sadržaja. Modelu se predaje opis svakog proizvoda korištenog u aplikaciji, a na izlazu modela se dobije vektor kao numerička reprezentaciju opisa tog proizvoda. Dobivenim vektorom proizvode možemo uspoređivati tj. odrediti sličnost između proizvoda upotrebom kosinusne sličnosti:

$$\text{cosine similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (4.1)$$

Preporuke su generirane na temelju proizvoda s kojima je korisnik imao interakciju,

a moguće interakcije su: pregled detalja proizvoda, stavljanje proizvoda u košaricu, kupnja proizvoda i kreiranje recenzije za proizvod. Na primjer, korisnik pregledava detalje određenog produkta, taj događaj šaljemo brokeru pod temom `PRODUCT_DETAIL`, Python aplikacija osluškuje događaje s brokera i obrađuje događaj tako da za dani proizvod nalazi 10 najbližijih proizvoda po kosinusnoj sličnosti i sprema preporuke u bazu podataka rangirano po umnošku vrijednosti kosinusne sličnosti i definirane težine događaja. Razlog zašto zasebna aplikacija obrađuje događaj, a ne aplikacija u kojoj je sadržana poslovna logika, je taj da događaja može biti jako puno i želimo da se oni obrađuju asinkrono. Svakoj od interakcija je dodijeljena težina prikazana tablicom 4.1.

| Naziv događaja | Težina | Opis događaja |
|-----------------------------|--------|-------------------------------------|
| <code>PRODUCT_DETAIL</code> | 0.25 | Pregled detalja proizvoda |
| <code>ADD_TO_CART</code> | 0.4 | Stavljanje proizvoda u web košaricu |
| <code>BOUGHT_ITEM</code> | 0.65 | Kupljeni proizvod |
| <code>REVIEW_ITEM</code> | 0.85 | Postavljena recenzija za proizvod |

Tablica 4.1. Težine dodijeljene događajima u sustavu

4.2.4. Generiranje preporuka suradničkim filtriranjem

Model suradničkog filtriranja je implementiran s pomoću neuronskih mreža za suradničko filtriranje (u nastavku teksta NCF). NCF koristi interakcije između korisnika i proizvoda za izradu preporuke. Tipično je ulaz u NCF latentni vektor korisnika spojen s latentnim vektorom proizvoda. Mreža je trenirana na podacima o recenzijama, a zadatak mreže je predvidjeti ocjenu koju će korisnik postaviti za određeni proizvod prilikom recenziranja. U implementaciji mreže je korišten latentni vektor korisnika koji se dodatno mijenja tijekom procesa treninga dok je za vektor proizvoda korišten predtrenirani vektor koji je spomenut u poglavlju o preporuci na temelju sadržaja i taj vektor ne mijenjamo tijekom treniranja mreže.

Predtrenirani vektor ima 768 varijabli, a to je puno više u odnosu na latentni vektor korisnika za koji smo definirali da ima 64 varijable. Razlog odabira manjeg latentnog vektora je taj jer se mreža s većim latentnim vektorom vrlo brzo prenauci i ne postiže optimalni rezultat. Zbog toga, prije ulaza u mrežu, za predtrenirani vektor proizvoda koristimo teh-

| Hiperparametar | Vrijednost |
|---------------------------|------------|
| Veličina bloka podataka | 64 |
| Maksimalni broj epoha | 60 |
| Funkcija gubitka | MSE |
| Optimizacijski algoritam | ADAM |
| Stopa učenja | 0.0001 |
| Regularizacijski faktor | 0.005 |
| Broj slojeva | 4 |
| Veličina zasebnih slojeva | 16 8 4 1 |

Tablica 4.2. Hiperparametri neuronske mreže suradničkog filtriranja

nike reduciranja dimenzionalnosti. Tehnike reduciranja dimenzionalnosti traže reprezentativni prikaz vektora iz većih vektorskih prostora u manjim vektorskim prostorima uz maksimalno zadržavanje informacija koje smo imali u većem vektorskom prostoru. Konkretna implementacija tehnike redukcije dimenzionalnosti koju koristimo je analiza glavnih komponenti (engl. Principal component analysis).

Za odabir optimalnih hiperparametara koristi se pretraživanje po rešetci. Odabrani hiperparametri koje se optimizira su: stopa učenja, regularizacijski faktor, veličina skriveneog sloja, veličina latentnog vektora. Pretraživanjem po rešetci isprobava se svaka moguća kombinacija hiperparametara i odabire se ona kombinacija koja ostvaruje najmanju pogrešku prilikom učenja. Odabrani hiperparametri su prikazani u tablici 4.2. Istrenirani model ostvaruje srednju kvadratnu pogrešku od 1.02 na evaluacijskom podatkovnom skupu.

Generiranje preporuka s NCF mrežama ne može se izvoditi u stvarnom vremenu zbog dugotrajnog procesa treniranja mreže. Iz tog razloga, proces treniranja mreže pokreće se ručno i generira najboljih 40 preporuka koje se spremaju u bazu podataka. Prethodno generirane preporuke se brišu prilikom svakog novog pokretanja procesa treniranja.

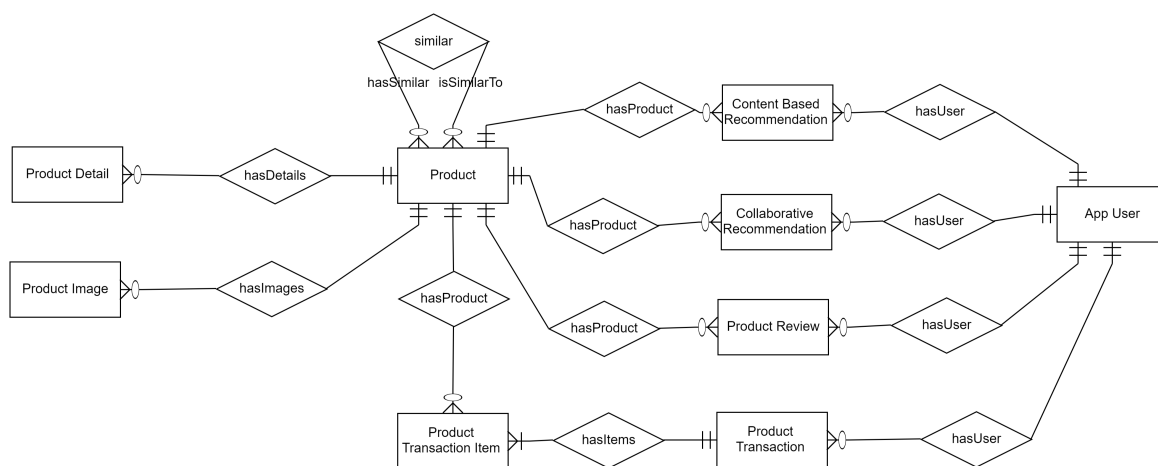
4.3. Baza podataka

U ovom poglavlju opisana je struktura baze podataka, prikazan je pripadajući ER model (4.4.) i dodatno su opisani entiteti i veze između entiteta. Struktura je osmišljena da bude fleksibilna tako da podržava buduća proširenja drugih tipova proizvoda.

Glavni entitet u pripadajućoj shemi baze na kojeg se većina drugih entiteta veže je Product (tablica 4.3.), a on reprezentira proizvode u sustavu. Svaki od proizvoda je dodatno opisan slikama (entitet Product Image, tablica 4.4.) i detaljima proizvoda (entitet Product Detail, tablica 4.5.). Tablica 4.6. predstavlja refleksivnu vezu između proizvoda gdje je za svaki proizvod definiran određeni broj sličnih proizvoda.

Entitet App User (tablica 4.7.) predstavlja korisnika u sustavu. Za svakog korisnika se bilježe njegove transakcije (entitet Product Transaction, tablica 4.8.), a svaka transakcija se sastoji od više artikala koji su modelirani entitetom Product Transaction Item koji je opisan tablicom 4.9. Product Review (tablica 4.10. predstavlja korisnikovu recenziju za određeni proizvod.

Entitetima Content Based Recommendation (tablica 4.11.) i Collaborative Recommendation (tablica 4.12.) su modelirane preporuke na temelju sadržaja i na temelju suradničkog filtriranja.



Slika 4.4. ER model baze podataka

| Naziv atributa | Podatkovni tip | Opis |
|---------------------|----------------|--|
| id | INT | Identifikator proizvoda |
| asin | TEXT | Identifikator proizvoda u Amazon sustavu |
| main_category | TEXT | Primarna kategorija proizvoda |
| title | TEXT | Naziv proizvoda |
| product_description | TEXT | Opis proizvoda |
| average_rating | FLOAT | Prosječna ocjena proizvoda |
| rating_number | INT | Broj recenzija na proizvod |
| price | TEXT | Cijena proizvoda |

Tablica 4.3. Product

| Naziv atributa | Podatkovni tip | Opis |
|----------------|----------------|---|
| id | INT | Identifikator slike proizvoda |
| product_id | INT | Identifikator proizvoda |
| hi_res | TEXT | Web link slike proizvoda u visokoj rezoluciji |
| thumb | TEXT | Web link male slike proizvoda |
| large | TEXT | Web link velike slike proizvoda |
| average_rating | FLOAT | Prosječna ocjena proizvoda |
| rating_number | INT | Broj recenzija na proizvod |
| price | TEXT | Cijena proizvoda |

Tablica 4.4. Product Image

| Naziv atributa | Podatkovni tip | Opis |
|----------------|----------------|----------------------------|
| product_id | INT | Identifikator proizvoda |
| key | TEXT | Ključ atributa proizvoda |
| value | TEXT | Vrijednost vezana uz ključ |

Tablica 4.5. Product Detail

| Naziv atributa | Podatkovni tip | Opis |
|----------------|----------------|--------------------------------------|
| id | INT | Identifikator para sličnih proizvoda |
| product_id | INT | Identifikator proizvoda |
| product_id | INT | Identifikator sličnog proizvoda |
| ranking | INT | Mjera sličnosti proizvoda |

Tablica 4.6. Content Similar Product

| Naziv atributa | Podatkovni tip | Opis |
|----------------|----------------|---|
| id | INT | Identifikator korisnika |
| username | TEXT | Korisničko ime korisnika |
| password | TEXT | SHA-256 sažetak lozinke korisnika |
| first_name | TEXT | Ime korisnika |
| last_name | TEXT | Prezime korisnika |
| email | TEXT | Račun elektroničke pošte korisnika |
| date_joined | TIMESTAMP | Vremenska oznaka registracije korisnika |

Tablica 4.7. App User

| Naziv atributa | Podatkovni tip | Opis |
|----------------|----------------|------------------------------|
| id | INT | Identifikator transakcije |
| app_user_id | TEXT | Identifikator korisnika |
| created_at | TIMESTAMP | Vremenska oznaka transakcije |

Tablica 4.8. Product Transaction

| Naziv atributa | Podatkovni tip | Opis |
|------------------------|----------------|--|
| id | INT | Identifikator kupljenog proizvoda u pojedinačnoj transakciji |
| product_transaction_id | INT | Identifikator transakcije |
| product_id | INT | Identifikator proizvoda |
| quantity | INT | Količina kupljenog proizvoda |

Tablica 4.9. Product_transaction_item

| Naziv atributa | Podatkovni tip | Opis |
|------------------|----------------|----------------------------------|
| id | INT | Identifikator recenzije |
| product_id | INT | Identifikator proizvoda |
| app_user_id | INT | Identifikator korisnika |
| rating | INT | Korisnikova ocjena za proizvod |
| review_text | TEXT | Korisnikov komentar uz recenziju |
| review_timestamp | TIMESTAMP | Vremenska oznaka recenzije |

Tablica 4.10. Product Review

| Naziv atributa | Podatkovni tip | Opis |
|----------------|----------------|--------------------------------------|
| id | INT | Identifikator preporuke po sadržaju |
| app_user_id | INT | Identifikator korisnika |
| product_id | INT | Identifikator preporučenog proizvoda |
| weight | FLOAT | Mjera relevantnosti preporuke |

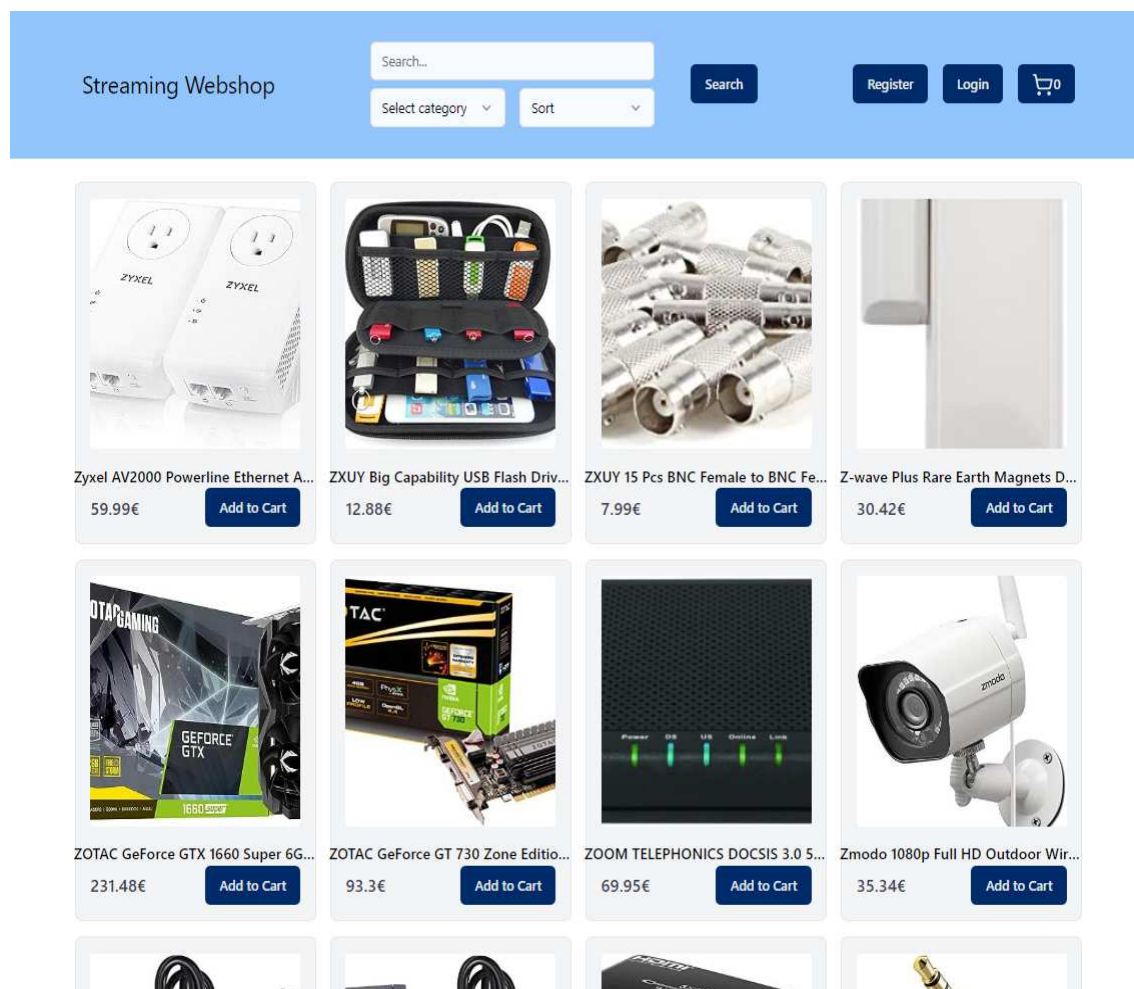
Tablica 4.11. Content Based Recommendation

| Naziv atributa | Podatkovni tip | Opis |
|-------------------|----------------|--|
| id | INT | Identifikator preporuke suradničkog filtriranja |
| app_user_id | INT | Identifikator korisnika |
| product_id | TEXT | Identifikator preporučenog proizvoda |
| ranking | INTEGER | Mjera relevantnosti preporuke |
| source_product_id | INTEGER | Identifikator proizvoda temeljem kojeg je kreirana preporuka |

Tablica 4.12. Collaborative Recommendation

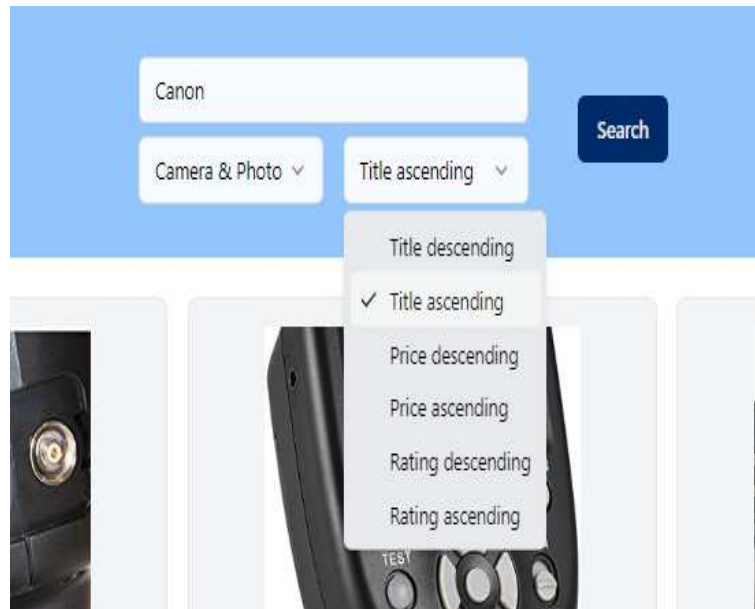
5. Implementacija i upute za korištenje

Prilikom prvog učitavanja sustava, korisniku se prikazuje početna stranica (slika 5.1.) s prikazom proizvoda u karticama i zaglavlje stranice na kojem je prikazan naziv web trgovine, tražilica i gumbi za navigaciju.



Slika 5.1. Početna stranica

Tražilica (slika 5.2.) se sastoji od sljedećih komponenti: polje za pretraživanje po nazivu proizvoda, padajući izbornik za filtriranje po kategoriji proizvoda i padajući izbornik za sortiranje proizvoda po nazivu ili cijeni. Klikom na gumb s tekстом "Search", primjenjuju se navedeni podaci za pretragu, filtriranje i sortiranje.



Slika 5.2. Polje za pretraživanje s padajućim izbornicima za filtriranje i sortiranje

Klikom na gumb na zaglavlju početne stranice s tekстом "Login", korisnika se preusmjerava na stranicu s formom za prijavu (slika 5.3.), a klikom na gumb s tekстом "Register", korisnik je preusmjeren na formu za registraciju (slika 5.4.). Svaka od formi provjerava jesu li ispunjeni svi podaci, a u slučaju neuspješne registracije ili prijave, korisniku su prikazane poruke sustava s detaljima pogreške ispod polja za unos na formama ili u obliku skočnog prozora.

Login

Username

Password

Required

Login

Don't have an account? [Register here](#)

Slika 5.3. Forma za prijavu

Register

Username

Email

First name

Last name

Password

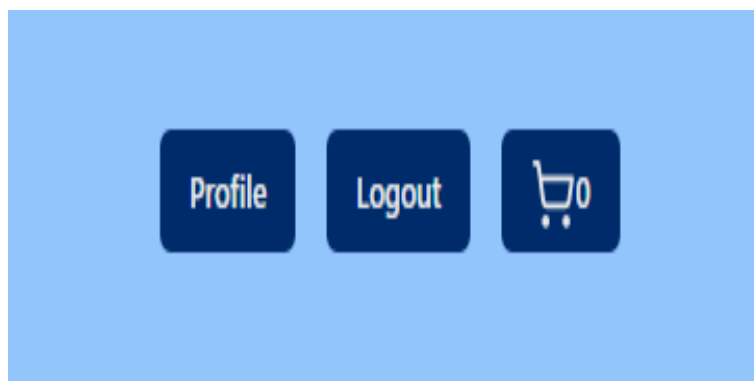
Confirm password

Register

Already have an account? [Login here](#)

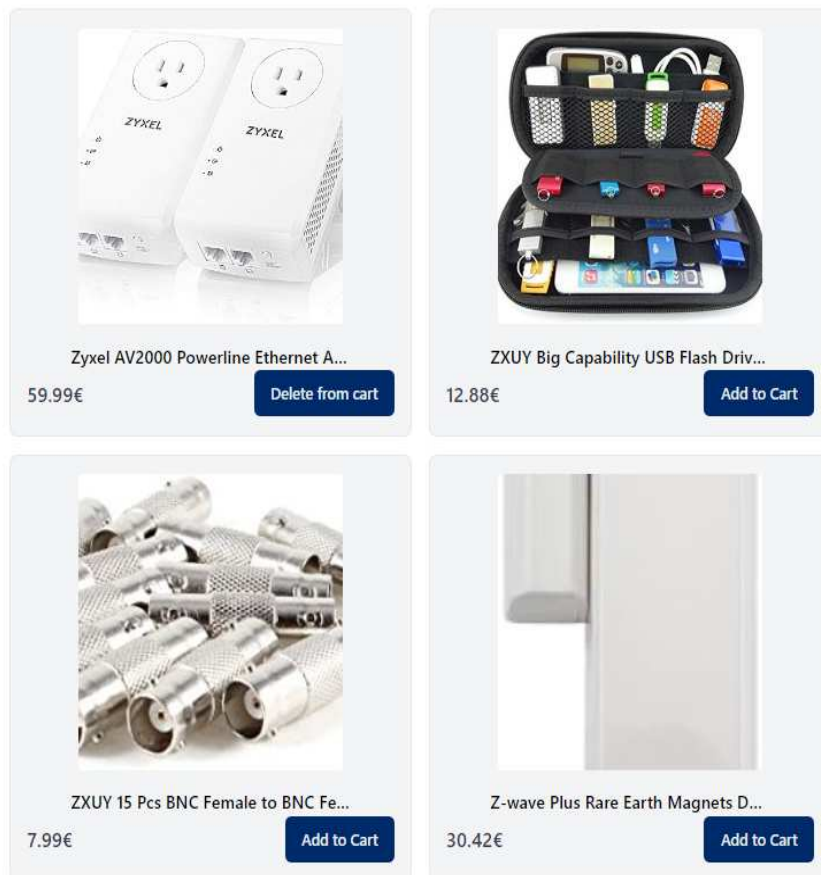
Slika 5.4. Forma za registraciju

Ako je korisnik prijavljen u sustav, umjesto gumba za registraciju i prijavu, prikazani su gumbi za prikaz korisnikova profila i za odjavu (slika 5.5.).



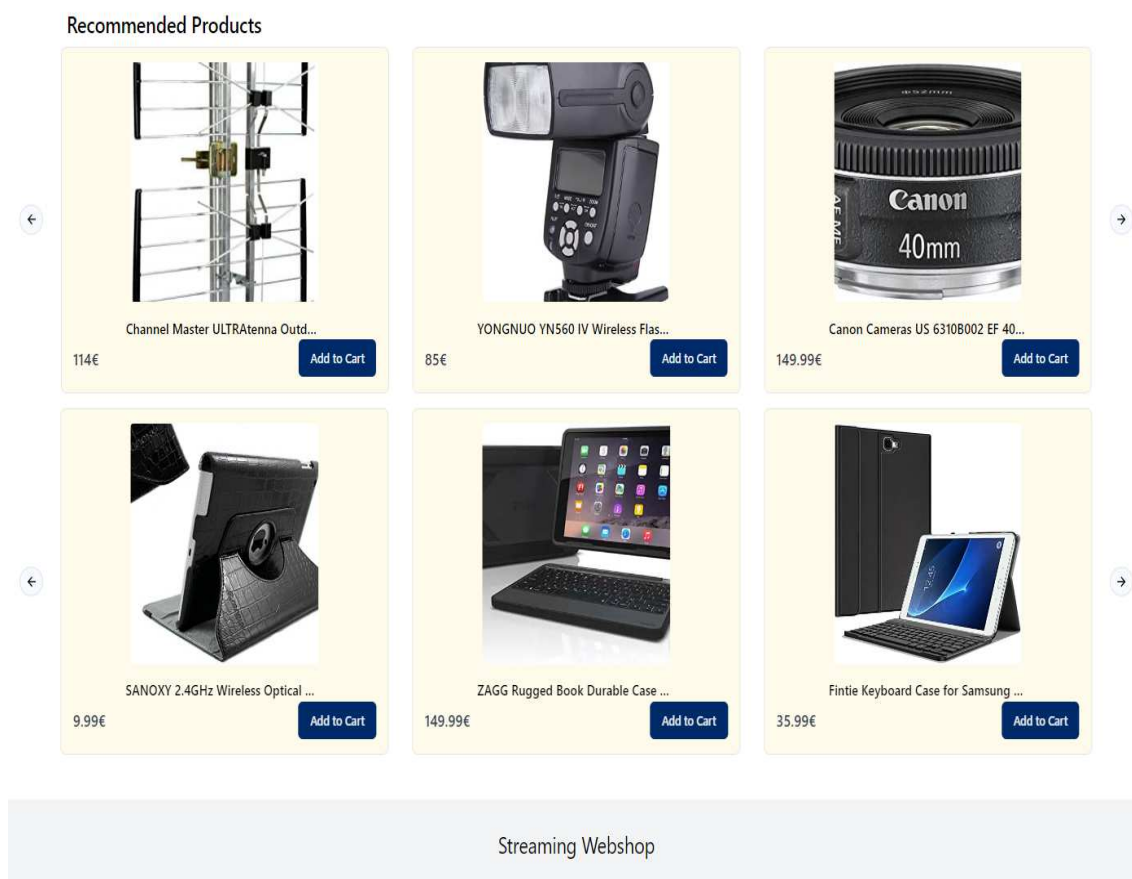
Slika 5.5. Prikazani gumbi za prijavljenog korisnika

Na svakoj kartici proizvoda na početnoj stranici je prikazana slika proizvoda, naziv proizvoda, cijena proizvoda i gumb s tekstom "Add to Cart" na čiji klik se odabrani proizvod dodaje u košaricu. Ako je proizvod već dodan u košaricu, tekst gumba je "Delete from Cart", a klikom na gumb proizvod se briše iz košarice. Klikom na tekst proizvoda ili na naziv proizvoda, korisniku se otvara stranica s prikazom detalja proizvoda (slika 5.6.).



Slika 5.6. Kartice proizvoda

Za korisnika koji je prijavljen u sustav, na dnu stranice prikazuju se preporučeni proizvodi u obliku karusela (slika 5.7.). U prvom karuselu su prikazani proizvodi preporučeni na temelju sadržaja dok su u drugom prikazani proizvodi preporučeni suradničkim filtriranjem.



Slika 5.7. Preporučeni proizvodi

Na stranici s detaljima pojedinog proizvoda, koja je prikazana na slici 5.8., nalazi se naziv proizvoda, slika proizvoda, cijena proizvoda, prosječna ocjena proizvoda, gumb s mogućnošću dodavanja proizvoda u košaricu ili brisanje iz košarice, opis proizvoda, detalji proizvoda prikazani u obliku ključ-vrijednost (slika 5.9.), recenzije proizvoda (slika 5.10.).



**Zmodo 1080p Full HD
Outdoor Wireless
Security Camera
System, Plug-In, Smart
Home Indoor Outdoor
WiFi IP Camera with
Night Vision,
Compatible with Alexa**

Price: 35.34€

Rating: 4.1 (31245) ratings

[Add to Cart](#)

← →

Product Description

Zmodo OUTDOOR WIRELESS SECURITY CAMERA · 1080P high-resolution image · Infrared Night Vision · Easy Setup · Smart motion detection · 90° viewing angle · Works with Alexa · Free 30-day Cloud Storage Please do not hesitate to contact us if you have any problem with the product.

Slika 5.8. Osnovni podaci o proizvodu

| Product Details | |
|------------------------------|---|
| Brand | Zmodo |
| Date First Available | September 4, 2019 |
| Model Name | SD-H1080P-Z |
| Recommended Uses For Product | Outdoor, Tablets |
| Item model number | SD-H1080P-Z |
| Manufacturer | Zmodo |
| Product Dimensions | 3.4 x 2.1 x 2.2 inches |
| Item Weight | 7.5 ounces |
| Best Sellers Rank | {Bullet Surveillance Cameras: 40} |
| Country of Origin | China |
| Connectivity Technology | 2.4GHz WPA/WPA2 WiFi Network with >2Mbps Upload Speed iOS (v9.0 or newer) or Android (v4.0 or newer)2.4GHz WPA/WPA2 WiFi Network with >2Mbps Upload Speed iOS (v9.0 or newer) or Android (v4.0 or newer) See more |
| Special Feature | Night Vision, Motion Sensor |

Slika 5.9. Detalji proizvoda u obliku ključ-vrijednost

Customer Reviews

A **AFGRKBHGHTQC6UQEPXLJDDXRKRZQ**
★★★★★

These cameras are high quality and have detail night vision is superb. Never had one go out on me I have the earth camera WiFi system with a 500gb hard drive hub and is outstanding I have had the cameras over 6 months and never lost contact with my home and animals cameras work great with motion recording. The pest price for hd camera owned other WiFi camera and always would connection problems

A **AGESZEGIZP66UUBTSPB7WHYU5KCA**
★★★★★

These cameras have a ton of problems but the worst one is using data all the time they are on! There is no way to see the video without a constant internet connection. They don't have video output and there is no way to connect to them with Wi-Fi, you have to use the app or website! I tried several IP camera viewer apps but none of them could connect to the cameras. If the camera and phone are both connected to my router why does it need data to view it? That will run over the data cap in no time! The description saying cloud service is optional is very misleading since they have to have a constant internet connection to work and that makes them mostly useless in my case. Also they need 120 volt outlets for power! While most security cameras have always used 12 volts these use a USB 5 volt adapter that must be plugged in but where are you going to get 120 volts at the camera mounting locations?

Slika 5.10. Recenzije proizvoda

Za prijavljenog korisnika se pri dnu stranice detalja proizvoda nalazi forma za kreiranje recenzije prikazana na slici 5.11. Na samom dnu stranice se nalazi karusel prikazan za sve korisnike sustava, a u njemu su prikazani slični proizvodi trenutno prikazanom proizvodu.

Leave a Review

Write your review here...

Rating: ★★★★★

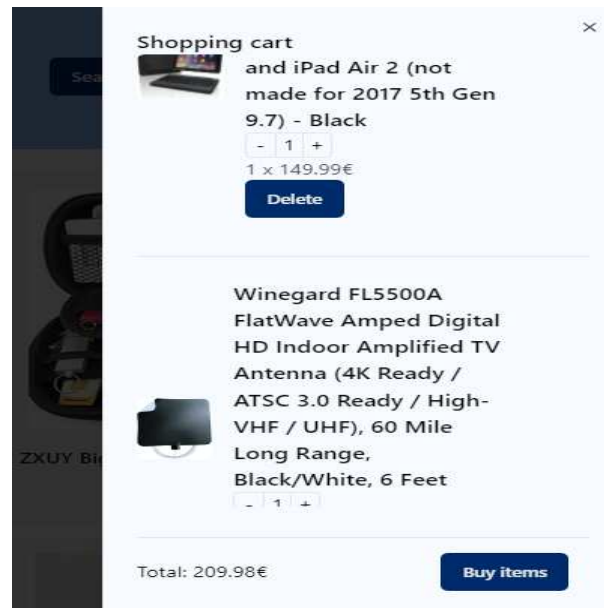
Submit Review



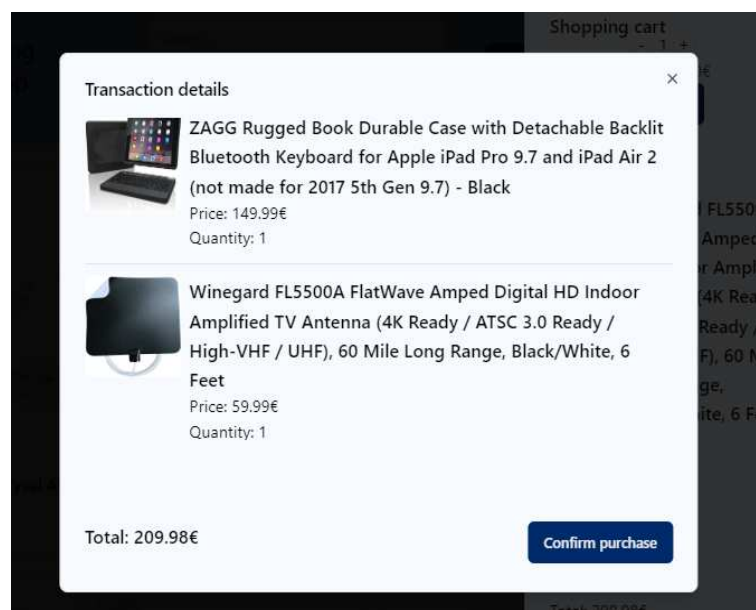
Streaming Webshop

Slika 5.11. Forma za kreiranje recenzije i karusel sličnih proizvoda

Klikom na gumb s ikonom košarice na zaglavlju početne stranice, otvara se bočna traka s prikazom proizvoda koji se trenutno nalaze u košarici i trenutna suma cijena svih proizvoda u košarici (slika 5.12.). Na bočnoj traci, korisnik može mijenjati broj artikala koje želi kupiti, brisati artikle iz košarice, a klikom na gumb s tekстом "Buy items" otvara se dijaloški prozor za potvrdu kupnje (slika 5.13.). Klikom na gumb s tekстом "Confirm purchase", kreira se transakcija s odabranim artiklima.



Slika 5.12. Prikaz proizvoda u košarici



Slika 5.13. Dijaloški prozor za potvrdu kupnje

Prijavljeni korisnik klikom na gumb s tekстом "Profile" otvara se stranica s prikazom detalja korisnika. Na lijevoj strani stranice (slika 5.14.) prikazani su detalji korisnika (korisničko ime, adresa elektroničke pošte, ime, prezime i vremenski trenutak registracije korisnika). Na desnoj strani stranice prikazane su sve korisnikove transakcije (slika 5.15.). Za svaku transakciju je istaknut identifikator transakcije, ukupna suma artikala i vremenski trenutak transakcije. Klikom na gumb s tekстом "Open transaction details" otvara se dijaloški prozor s detaljnim prikazom svih kupljenih proizvoda koji je prikazan na slici 5.16.

User profile

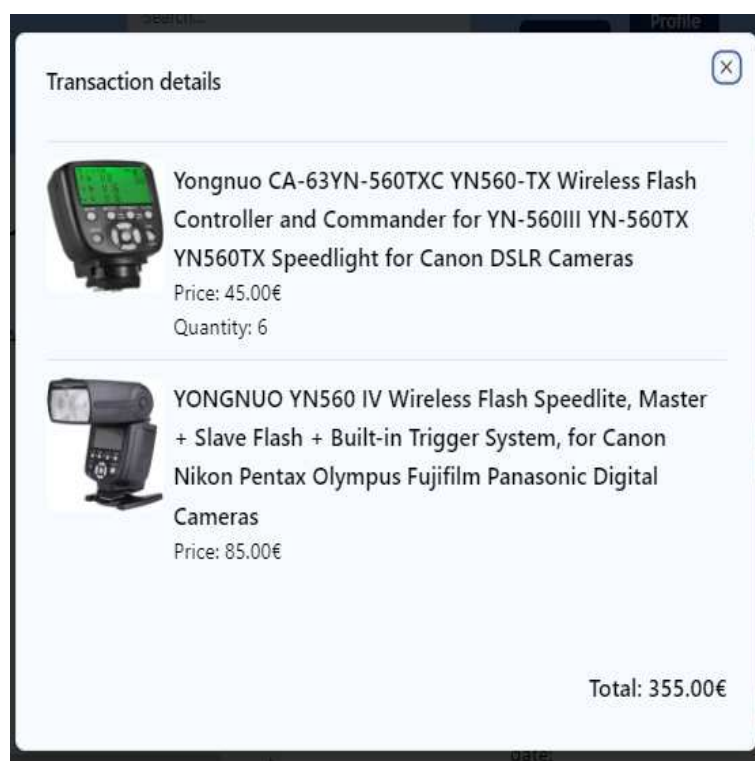
| | |
|-------------|-----------------------|
| Username | tgrbesa |
| Email | tomislav721@gmail.com |
| First name | Tomislav |
| Last name | Grbesa |
| Date joined | 23.06.2024 21:06 |

Slika 5.14. Detalji korisnika

Transaction history

| | |
|--|--|
| Transaction ID: 65 Total: 355.00€ Transaction date: 23.06.2024 23:24 | Open transaction details |
| Transaction ID: 66 Total: 209.98€ Transaction date: 24.06.2024 00:07 | Open transaction details |

Slika 5.15. Transakcije korisnika



Slika 5.16. Prikaz detalja transakcije

6. Zaključak

U ovom diplomskom radu je u konačnici prikazana arhitektura i implementacija modernog preporučiteljskog sustava koji kreira preporuke u realnom vremenu na temelju strujećih podataka. Uz obradu podataka u stvarnom vremenu, velika prednost ovakvog sustava je korištenje dubokih modela specijaliziranih za obradu teksta koji su trenirani na ogromnim skupovima podataka i postižu odlične rezultate na zadacima obrade prirodnog jezika, a vrlo lako su primjenjivi za preporučiteljske sustave, posebno za sustave generiranje preporuka na temelju sadržaja. Dodatno, ovako implementirani sustav adresira probleme pojedinačnih pristupa i daje hibridne preporuke koje su generirane na temelju sadržaja proizvoda i suradničkim filtriranjem.

Sustav ima velikog prostora za dodavanje novih funkcionalnosti poslovne logike i načina preporuke proizvoda. Primjer jedne takve funkcionalnosti je semantička pretraga proizvoda koristeći isti All-mpnet-base-v2 model koji je korišten za kreiranje preporuka na temelju sadržaja.

Nedostatak ovako implementiranog sustava koji koristi mikroservisnu arhitekturu je taj da nam je potrebno puno više resursa za održavanje za razliku od npr. monolitne arhitekture. Dodatna otežavajuća okolnost je ta da su aplikacija za kreiranje preporuka i pozadinska aplikacija poslovne logike pisane u različitim programskim jezicima za što je potrebno više razvojnih programera nego uobičajeno za razvoj i održavanje.

Literatura

- [1] D. J. et al., “Recommender systems: Past, present, future”, *AI Magazine*, 2021.
- [2] “What is netflix?” <https://help.netflix.com/en/node/412>, pristupano 1.6.2024.
- [3] “Top streaming statistics in 2024”, <https://www.forbes.com/home-improvement/internet/streaming-stats/>, pristupano 1.6.2024.
- [4] “Understanding recommendations on spotify”, <https://www.spotify.com/us/safetyandprivacy/understanding-recommendations>, pristupano 2.6.2024.
- [5] “Java documentation”, <https://www.oracle.com/java/#rc30documentation>, pristupano 10.5.2024.
- [6] “Spring framework documentation”, <https://docs.spring.io/spring-framework/reference/index.html>, pristupano 12.5.2024.
- [7] “Spring ioc container”, https://www.interviewbit.com/spring-interview-questions/?sign_up_medium=ib_article_auth_blocker, pristupano 18.5.2024.
- [8] “Spring boot”, <https://docs.spring.io/spring-boot/index.html>, pristupano 12.5.2024.
- [9] “Postgresql 16.3 documentation”, <https://www.postgresql.org/docs/current/index.html>, pristupano 14.5.2024.
- [10] “Kafka 3.7 documentation”, <https://kafka.apache.org/documentation/>, pristupano 15.5.2024.
- [11] “Python 3.12.4 documentation”, <https://docs.python.org/3.12/>, pristupano 16.5.2024.

- [12] “React reference overview”, <https://react.dev/reference/react>, pristupano 16.5.2024.
- [13] C. Richardson, “What are microservices?” <https://microservices.io/>, pristupano 20.5.2024.
- [14] “Three tier (three layer) architecture in spring mvc web application”, <https://www.javaguides.net/2020/07/three-tier-three-layer-architecture-in-spring-mvc-web-application.html>, pristupano 5.6.2024.
- [15] Y. Hou, J. Li, Z. He, A. Yan, X. Chen, i J. McAuley, “Bridging language and items for retrieval and recommendation”, *arXiv preprint arXiv:2403.03952*, 2024.
- [16] A. V. et al., “Attention is all you need”, <https://arxiv.org/pdf/1706.03762>, 2017., pristupano 19.5.2024.
- [17] “All-mpnet-base-v2”, <https://huggingface.co/sentence-transformers/all-mpnet-base-v2>, pristupano 26.5.2024.

Sažetak

Implementacija preporučiteljskog sustava utemeljenog na strujećim podacima

Tomislav Grbeša

U ovom radu prikazana je implementacija i arhitektura preporučiteljskog sustava utemeljenog na strujećim podacima. Analizirani su postojeći preporučiteljski sustavi te su objašnjene prednosti i nedostaci takvih sustava. Prikazane su korištene tehnologije, objašnjena je mikroservisna arhitektura implementiranog sustava i pojedinih aplikacija i opisan je način rada implementiranog preporučiteljskog algoritma. Na kraju je prikazana implementacija web aplikacije koja demonstrira funkcionalnosti preporučiteljskog sustava.

Ključne riječi: preporučiteljski sustav; strujeći podaci; web aplikacija; mikroservisna arhitektura

Abstract

Implementation of a recommendation system based on streaming data

Tomislav Grbeša

This paper presents the implementation and architecture of a recommendation system based on streaming data. Existing recommendation systems are analyzed, and the advantages and disadvantages of such systems are explained. The technologies used are presented, the architecture of the implemented system and individual applications are explained, and the operation of the implemented recommendation algorithm is described. Finally, the implementation of a web application demonstrating the functionalities of the recommendation system is presented.

Keywords: recommender system; streaming data; web application; microservice architecture