

Sustav za personalizirano preporučivanje video sadržaja

Đurđević, Vladimir

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:168:346323>

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

Download date / Datum preuzimanja: **2025-03-27**



Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 608

SUSTAV ZA PERSONALIZIRANO PREPORUČIVANJE VIDEO
SADRŽAJA

Vladimir Đurđević

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 608

SUSTAV ZA PERSONALIZIRANO PREPORUČIVANJE VIDEO
SADRŽAJA

Vladimir Đurđević

Zagreb, lipanj 2024.

DIPLOMSKI ZADATAK br. 608

Pristupnik: **Vladimir Đurđević (0036522404)**
Studij: Računarstvo
Profil: Programsko inženjerstvo i informacijski sustavi
Mentor: doc. dr. sc. Tomislav Jagušć

Zadatak: **Sustav za personalizirano preporučivanje video sadržaja**

Opis zadatka:

Kako raste količina materijala dostupnih na internetu, sustavi koji omogućavaju personalizirano preporučivanje sadržaja sve više dobivaju na važnosti. U sklopu ovog diplomskog rada potrebno je razviti sustav koji roditeljima daje prilagođene i primjerene preporuke video sadržaja za njihovu djecu, temeljene na dječjim preferencijama i interesima. Prvi korak u implementaciji sustava uključuje prikupljanje podataka o korisničkim preferencijama i interesima kroz interaktivni korisnički upitnik. Nakon toga, sustav treba analizirati prikupljene informacije i generirati personalizirane preporuke. Potrebno je usporediti postojeće biblioteke za preporučivanje s vlastitim AI modelom temeljenim na podacima videa (naslov, opis, oznake itd.). U sklopu rada, potrebno je detaljno istražiti dostupne biblioteke i alate za preporučivanje sadržaja te usporediti njihovu učinkovitost i prilagodljivost. Sustav treba omogućiti pohranu povijesti korisničkih interakcija te evaluaciju učinkovitosti preporuka. Osim toga, potrebno je razviti korisničko sučelje za praćenje i usporedbu preporuka, pružajući korisnicima transparentnost i kontrolu nad preporukama koje dobivaju.

Rok za predaju rada: 28. lipnja 2024.

Zahvala

Zahvaljujem svima koji su na bilo koji način pridonijeli nastanku i dovršetku ovog diplomskog rada.

Prije svega, iskreno zahvaljujem svom mentoru doc. dr. sc. Tomislavu Jaguštu, na njegovoj stručnosti, podršci i strpljenju tijekom cijelog procesa izrade rada. Njegovi savjeti i smjernice bili su neprocjenjivi i omogućili su mi da unaprijedim svoje znanje i vještine.

Posebnu zahvalnost dugujem svojim roditeljima, koji su mi pružili neizmjernu podršku i ljubav kroz cijelo školovanje. Njihovo razumijevanje i ohrabrenje bili su ključni za moje uspjehe.

Također želim izraziti veliku zahvalnost svojoj baki i djedu, koji su uvijek vjerovali u mene i svojim primjerom pokazali važnost upornosti, truda i rada.

Veliko hvala mojoj djevojci, čija je podrška, ljubav i razumijevanje bila neprocjenjiva tijekom cijelog procesa. Njena vjera u mene pomogla mi je da ostanem motiviran i predan radu.

Na kraju, zahvaljujem se i svojim prijateljima, koji su uvijek bili tu za mene, pružali podršku i pomagali mi da se opustim i zadržim ravnotežu između rada i odmora.

Bez svih vas, ovaj rad ne bi bio moguć. Hvala vam od srca.

Sve vas volim!

Uvod	1
1. Pregled područja	3
1.1. Uvod u sustave za preporučivanje sadržaja	3
1.2. Tipovi sustava za preporučivanje sadržaja	3
1.2.1. Kolaborativno filtriranje	3
1.2.2. Filtriranje temeljeno na sadržaju	4
1.2.3. Hibridni sustavi	4
1.3. Prikupljanje i obrada podataka	5
1.3.1. Interaktivni korisnički upitnici	5
1.3.2. Analiza korisničkih podataka	5
1.4. Evaluacija učinkovitosti preporuka	6
1.5. Postojeće aplikacije slične namjene	6
1.5.1. YouTube Kids	6
1.5.2. PBS Kids Video.....	7
1.5.3. Khan Academy Kids.....	8
2. Korištene tehnologije, alati i programski jezici.....	10
2.1. Next.js.....	10
2.1.1. Tailwind CSS i DaisyUI.....	11
2.2. Node.js.....	12
2.2.1. Express.js.....	12
2.3. MongoDB	13
2.4. Python.....	14
2.4.1. Flask	15
2.4.2. SentenceTransformer.....	16
2.4.3. Faiss.....	16
3. Model baze podataka	17

3.1.	Opis kolekcija	17
4.	Opis korištenja aplikacije	22
4.1.	Prijava i registracija	22
4.2.	Pregled sučelja s video sadržajem	25
4.2.1.	Rješenje za desktop računala	25
4.2.2.	Rješenje za mobilne uređaje	27
4.3.	Pregled aplikacije za unos i izmjenu video sadržaja	29
5.	Usporedba implementiranih sustava.....	34
5.1.	Amazon Personalize	34
5.2.	Recombee	35
5.3.	Izrada vlastitog rješenja	37
6.	Učinkovitost sustava.....	41
	Zaključak	45
	Literatura	46
	Sažetak.....	49
	Summary.....	50

Uvod

U današnje digitalno doba, sustavi za preporučivanje video sadržaja postaju sve važniji. S obzirom na količinu dostupnog sadržaja, ovi sustavi igraju ključnu ulogu u pomaganju korisnicima da pronađu relevantne i zanimljive videozapise. Na platformama poput YouTube-a, Netflix-a i sličnih, sustavi za predlaganje sadržaja koriste složene algoritme kako bi analizirali korisničke preferencije i ponudili personalizirane prijedloge. Ovakav pristup ne samo da povećava zadovoljstvo korisnika, već i povećava vrijeme provedeno na platformama, čime se ostvaruju poslovni ciljevi ovih kompanija.

Međutim, iako su postojeći sustavi za preporučivanje iznimno sofisticirani i efikasni, često zanemaruju važnost kvalitete sadržaja koji preporučuju, osobito kada je riječ o djeci. Djeca su posebno ranjiva skupina korisnika jer su u procesu razvoja i oblikovanja svojih stavova, vrijednosti i interesa. U današnjem informacijskom okruženju, djeca lako mogu biti izložena neprimjerenom, nasilnom ili besmislenom sadržaju koji može negativno utjecati na njihovo emocionalno i intelektualno zdravlje. Stoga je izuzetno važno razvijati sustave za preporučivanje video sadržaja koji će biti prilagođeni potrebama i interesima djece, a istovremeno osigurati da su ti sadržaji obrazovni, inspirativni i prikladni njihovoj dobi.

Postoji nekoliko već razvijenih sustava koji nude roditeljsku kontrolu i filtriranje sadržaja, ali mnogi od tih sustava nisu dovoljno učinkoviti. Na primjer, algoritmi mogu pogrešno procijeniti prikladnost određenih videozapisa ili jednostavno nemaju dovoljno preciznosti da bi isključili sav nepoželjan sadržaj. Kao rezultat toga, javlja se potreba za razvojem sustava koji će osigurati sigurnu digitalnu okolinu za djecu. Takvi sustavi trebali bi biti sposobni prepoznati ne samo očite neprikladnosti poput nasilja ili nepristojnog jezika, već i suptilnije aspekte poput negativnih stereotipa ili sadržaja koji potiče pasivno gledanje umjesto aktivnog učenja.

Za roditelje je posebno važno imati kontrolu nad sadržajem koji njihova djeca konzumiraju. Roditelji žele biti sigurni da njihova djeca provode vrijeme gledajući sadržaje koji će pozitivno utjecati na njihov razvoj. U ranijim godinama, kada djeca najviše uče i razvijaju svoje osobne interese, važno je da sadržaji budu pažljivo odabrani kako bi podržavali njihov intelektualni, emocionalni i socijalni razvoj. Integracija roditeljskih kontrola unutar sustava

za preporučivanje omogućava roditeljima da prilagode iskustvo gledanja svojim vrijednostima i standardima, čime se stvara sigurnija i poučnija digitalna okolina za djecu.

Unatoč značajnim prednostima koje sustavi za preporučivanje video sadržaja donose korisnicima i platformama, postoji potreba za razvojem specijaliziranih, dječjih algoritama koji će ponuditi kvalitetan i siguran sadržaj. Ovime bi se osiguralo da digitalni mediji budu alat za učenje i razvoj, a ne izvor potencijalne štete. Također, omogućavanje roditeljima veće kontrole nad sadržajem koji je dostupan njihovoj djeci doprinosi općem osjećaju sigurnosti i povjerenja u digitalne platforme.

U ovom radu ću najviše truda, vremena i znanja uložiti u kreiranje vlastitog sustava za preporučivanje video sadržaja i usporediti ga s gotovim rješenjima (Recombee [1] i Amazon Personalize [2]). Cilj ovog rada je isprobati i približiti se rješenjima velikih korporacija i tvrtki. Isto tako, cilj je evaluirati korisnost gotovih rješenja, usvojiti što više znanja i iskustva te procijeniti isplati li se kreirati vlastito rješenje.

1. Pregled područja

1.1. Uvod u sustave za preporučivanje sadržaja

Sustavi za preporučivanje sadržaja postali su ključna stavka za upravljanje informacijama dostupnima na internetu. Koristeći korisničke preferencije i povijest interakcija pružaju se personalizirane preporuke. Od svojih početaka, sustavi za preporučivanje sadržaja doživjeli su značajan razvoj usvajajući različite pristupe kako bi poboljšali točnost i relevantnost preporuka. U radu *Introduction to Recommender Systems Handbook* [3] pružen je sveobuhvatan uvod u ovu temu, obuhvaćajući povijest, osnovne pojmove i razne tehnike koje su korištene u sustavima za preporučivanje sadržaja.

Sustavi za preporučivanje sadržaja dijele se u tri glavne skupine: kolaborativno filtriranje, filtriranje temeljeno na sadržaju i hibridne metode [4]. Svaka skupina razlikuje se po konceptima i metodologijama. Kolaborativno filtriranje koristi podatke o interakcijama korisnika s različitim sadržajima kako bi pronašlo sličnosti između korisnika i predložilo nove stavke koje bi ih mogle zanimati. S druge strane, filtriranje temeljeno na sadržaju analizira karakteristike sadržaja kako bi preporučilo stavke koje su slične onima koje je korisnik prethodno visoko ocijenio. Prethodne dvije metode sadrže određene nedostatke koje se nastoje nadoknaditi hibridnim sustavima.

1.2. Tipovi sustava za preporučivanje sadržaja

1.2.1. Kolaborativno filtriranje

Kolaborativno filtriranje jedna je od najčešće korištenih tehnika u sustavima za preporučivanje sadržaja. Vrlo je učinkovita i jednostavna. Postoje dvije glavne vrste kolaborativnog filtriranja, jedna se temelji na korisnicima, a druga na sadržaju. U kolaborativnom filtriranju temeljenom na korisnicima, preporuke se generiraju pronalaženjem korisnika sa sličnim interakcijama i preporučivanjem sadržaja koje su ti slični korisnici ocijenili visoko. U drugoj metodi, temeljenoj na sadržaju, slične stavke se identificiraju na temelju korisničkih ocjena, a korisniku se preporučuje sadržaj sličan onome koji je već pozitivno ocijenio [5].

Jedan od ključnih izazova kolaborativnog filtriranja je problem hladnog starta (eng. *cold start*) koji se javlja kada postoji nedostatak podataka o novim korisnicima ili sadržaju. Za rješavanje ovog problema predložene su različite metode, uključujući i korištenje vanjskih podataka ili implementaciju hibridnih sustava koji nadopunjavaju nedostatke kolaborativnog filtriranja [4].

1.2.2. Filtriranje temeljeno na sadržaju

Filtriranje temeljeno na sadržaju koristi karakteristike sadržaja, kao što su tekstualni opisi, metapodaci (eng. *metadata*), oznake i sl., za generiranje preporuka. Ovakav pristup se oslanja na analizi značajnih stavki koje su korisnici prethodno pozitivno ocijenili kako bi pronašao slične stavke koje bi ih mogle zanimati [6]. Prednost ovakvog pristupa je njegova sposobnost preporučivanja sadržaja novim korisnicima. Novim korisnicima se na temelju njihovih početnih preferencija preporučuje sadržaj, čime se smanjuje problem hladnog starta.

Filtriranje temeljeno na sadržaju također ima određena ograničenja. Na primjer, postoji mogućnost poteškoća s preporukom sadržaja koji je različit od onoga koji je korisnik već pogledao ili ocijenio [6]. Na taj način smanjuje se inovativnost što je vrlo važna stavka sustava za preporučivanje sadržaja. Kako bi se ublažili ovi problemi, razvijene su razne tehnike, uključujući integraciju vanjskih izvora podataka i korištenje algoritama za obradu prirodnog jezika (eng. *Natural Language Processing* – NLP) za bolje razumijevanje sadržaja.

1.2.3. Hibridni sustavi

Hibridni sustavi kombiniraju različite metodologije i koncepte kako bi poboljšali točnost i povećali raznovrsnost preporuka. U radu *Hybrid recommender systems: Survey and experiments* [7] pruža se pregled različitih metoda i pristupa te njihovo kombiniranje. Neki hibridni sustavi koriste sekvencijalne pristupe gdje se preporuke generiraju korištenjem jedne metode, a zatim se samo filtriraju korištenjem neke druge metode. Drugi pristupi koriste paralelnu integraciju u kojoj se istovremeno kombiniraju različite metode kako bi se generirale preporuke.

Glavna prednost hibridnih sustava je njihova sposobnost i fleksibilnost da nadoknade nedostatke pojedinih metoda. Na primjer, kombinacijom kolaborativnog filtriranja i

filtriranja temeljenog na sadržaju hibridni sustav može pružiti bolje preporuke u slučajevima gdje nema dovoljno podataka o korisnicima ili o stavkama koje se preporučuju [7]. Hibridni sustavi su posebno korisni u velikim i složenim scenarijima gdje je potrebno uzeti u obzir razne faktore i kombinirati ih u generiranju preporuka.

1.3. Prikupljanje i obrada podataka

1.3.1. Interaktivni korisnički upitnici

Ključni korak u razvoju personaliziranih sustava za preporučivanje sadržaja je prikupljanje podataka o korisničkim preferencijama. Interaktivni korisnički upitnici su često korišteni alat za prikupljanje podataka. U radu *Usability Engineering for the Adaptive Web* [8] govori se o različitim metodama za dizajn i implementaciju interaktivnih upitnika. Također, naglašavaju važnost prilagodljivost i upotrebljivosti upitnika kako bi se osigurala točnost prikupljenih podataka i zadovoljstvo korisnika.

Upitnici mogu biti kreirani na različite načine, kao otvorena pitanja, pitanja višestrukog izbora ili ljestvice ocjena. Kvalitetno dizajnirani upitnici mogu pomoći u preciznom prikupljanju podataka o preferencijama i interesima korisnika. Precizni podaci ključni su za generiranje točnih i relevantnih preporuka [8].

1.3.2. Analiza korisničkih podataka

Analiza podataka prikupljenih od korisnika važna je za razvoj učinkovitih sustava za preporučivanje sadržaja. U radu *Collaborative Filtering Recommender Systems* [9], autori daju pregled metoda za analizu korisničkih podataka u kontekstu kolaborativnog filtriranja. Autori također govore o različitim tehnikama za obradu i analizu podataka, uključujući prepoznavanje obrazaca i identifikaciju sličnosti između korisnika

Jedan od glavnih izazova u analizi korisničkih podataka je upravljanje velikim količinama podataka i osiguravanje točnosti analize. Različite metode strojnog učenja, uključujući udruživanje (eng. *clusterization*) i klasifikaciju, mogu se koristiti za prepoznavanje obrazaca i generiranje preporuka na temelju korisničkih podataka [9]. Također, integracija podataka iz različitih izvora može poboljšati točnost i relevantnost preporuka.

1.4. Evaluacija učinkovitosti preporuka

Evaluacija učinkovitosti preporuka koje daju sustavi za preporučivanje sadržaja ključna je za razumijevanje njihove točnosti i korisnosti.

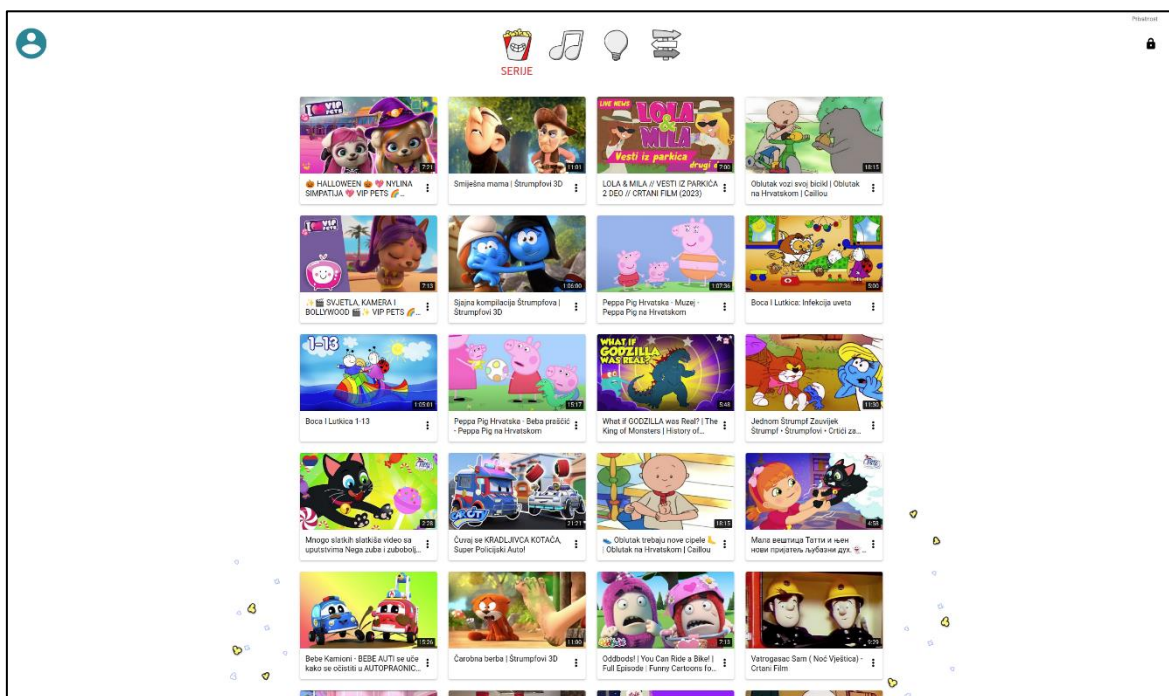
U radu *Evaluating collaborative filtering recommender systems* [10], autori pružaju sveobuhvatan pregled metodologija za evaluaciju kolaborativnih sustava preporučivanja. Autori detaljno govore o različitim mjerama učinkovitosti kao što su preciznost, pokrivenost, inovativnost i zadovoljstvo korisnika. Preciznost se odnosi na točnost preporuka, dok pokrivenost predstavlja mjeru sustava u preporučivanju širokog spektra sadržaja. Inovativnost se odnosi na sposobnost sustava da preporuči nove stavke koje korisnici nisu ranije vidjeli, dok zadovoljstvo korisnika mjeri koliko su korisnici zadovoljni preporukama koje dobivaju. Neke od prethodno spomenutih mjera učinkovitosti korištene su i na kraju ovog rada.

Kombinacija različitih mjera evaluacije i kontinuirano praćenje performansi sustava za predlaganje sadržaja ključni su za njihovo poboljšanje i prilagodbu korisničkim potrebama. Osim toga, uključivanje povratnih informacija korisnika u proces evaluacije može pomoći boljem razumijevanju njihovih preferencija i poboljšanju sustava [10].

1.5. Postojeće aplikacije slične namjene

1.5.1. YouTube Kids

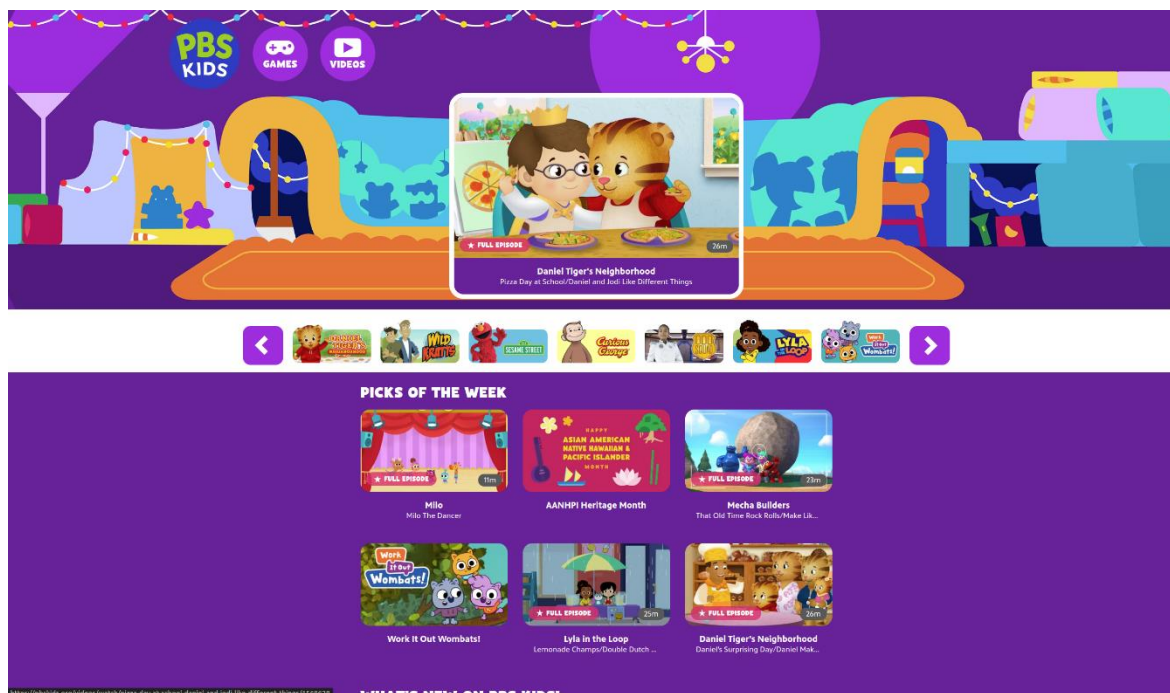
Aplikacija YouTube Kids [11] je razvijena od strane YouTube-a specifično za djecu. Pokrenuta je s ciljem pružanja kontroliranog i kvalitetnog sadržaja za djecu, primjerenog njihovoj dobi. Aplikacija nudi mogućnost roditeljskog nadzora kroz postavljanje raznih ograničenja te kreiranja profila. Moguće je kreirati poseban profil za svako dijete te kontrolirati koliko vremena djeca provode na aplikaciji. Također, postoji mogućnost filtriranja sadržaja koje roditelj smatra neprikladnim. Prednosti korištenja ovakve aplikacije su sigurno okruženje za djecu uz kontrolu i nadzor roditelja, jednostavnost te kontrola vremena. Neki od nedostataka su: ponekad neprikladan videozapis može proći kroz filtre, ograničena količina sadržaja, i ovisnost o tehnologiji. Slika 1 prikazuje izgled aplikacije YouTube Kids.



Slika 1. Aplikacija YouTube Kids [11]

1.5.2. PBS Kids Video

Aplikacija PBS Kids Video [12] razvijena je od strane Public Broadcasting Service (PBS) kako bi pružila sigurno i edukativno okruženje za djecu. Slika 2 prikazuje izgled početnog sučelja aplikacije PBS Kids Video. Namijenjena je za djecu u dobi od dvije do osam godina te uz video sadržaj nudi i različite igre. Kao i ostale aplikacije slične namjene, uključuje mogućnost roditeljskog nadzora te sigurno okruženje. Neke od glavnih prednosti ove aplikacije su integracija obrazovnog kurikuluma, pristup bez reklama i cijena (aplikacija je besplatna). Glavni nedostaci ove aplikacije su ograničenje sadržaja i kurikuluma na prostor SAD-a, a mogućnost personalizacije je sužena.



Slika 2. Aplikacija PBS Kids Video [12]

1.5.3. Khan Academy Kids

Khan Academy Kids [13] je besplatna obrazovna aplikacija namijenjena djeci u dobi od 2 do 8 godina. Razvijena je od strane neprofitne organizacije Khan Academy. Cilj pokretanja je pružanje visokokvalitetnog obrazovnog sadržaja kroz razne interaktivne aktivnosti, lekcije i igre. Sve navedene kategorije potiču razvoj raznih vještina, kreativnost i socio-emocionalni razvoj. Sadržaj koji aplikacija nudi je širokog spektra, što znači da pokriva razne skupine, kao što su STEM, aktivnosti društvenog karaktera i sl.. I ova aplikacija kao i sve prethodne ima mogućnost roditeljskog nadzora. Neke od glavnih prednosti ove aplikacije su: besplatna je, fokus na sveobuhvatni razvoj djeteta i personalizacija učenja. Od nedostataka ističe se ograničenost obrazovnih predmeta za djecu iznad osam godina te potreba posjedovanja Android ili iOS uređaja za pristup aplikaciji, budući da ne postoji web inačica. Slika 3 predstavlja aplikaciju Khan Academy Kids.

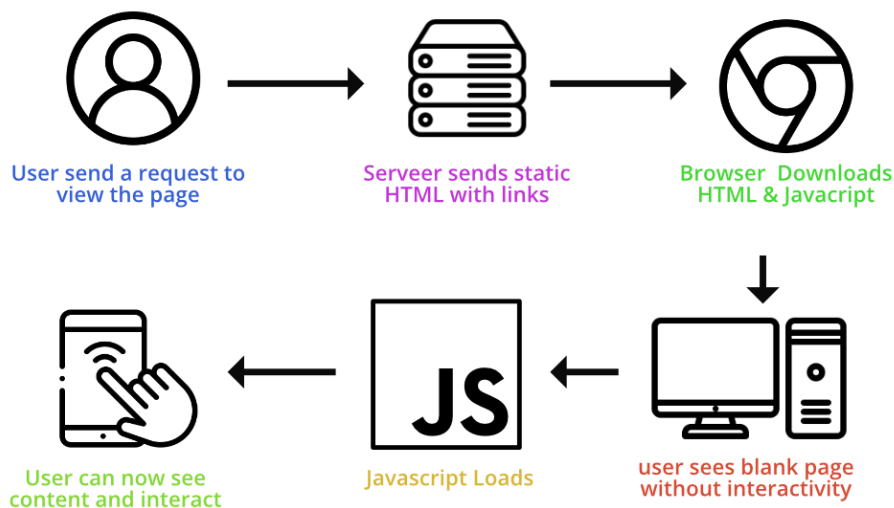


Slika 3. Aplikacija Khan Academy Kids [13]

2. Korištene tehnologije, alati i programski jezici

2.1. Next.js

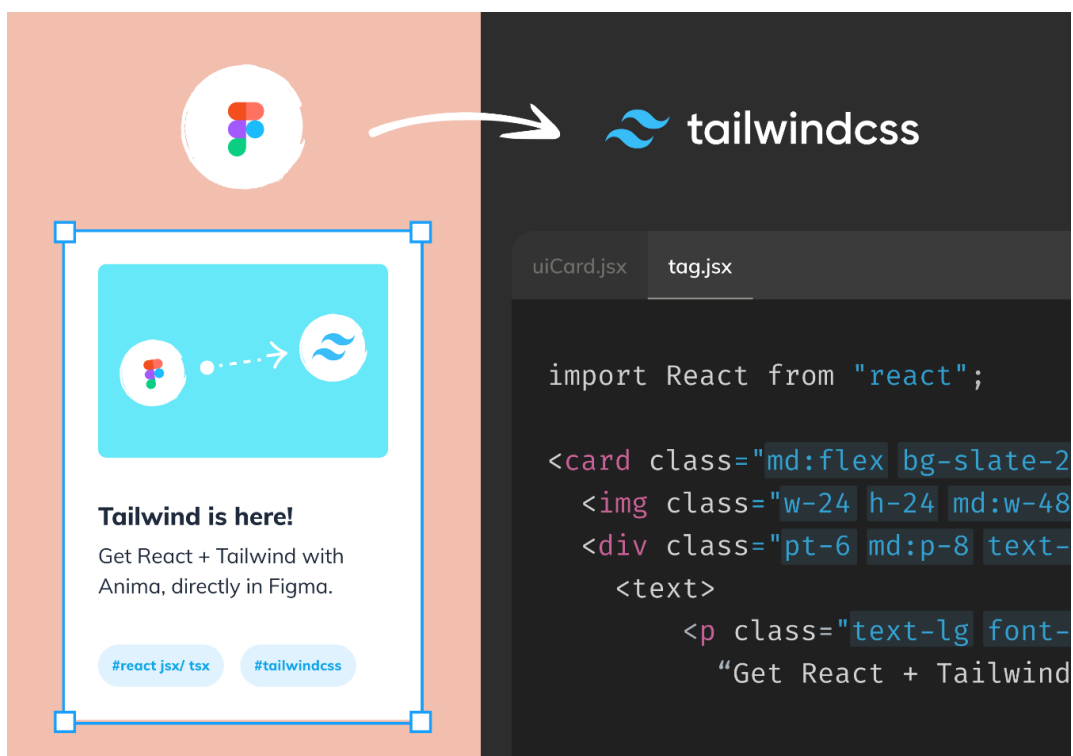
Next.js [14] je radni okvir otvorenog koda za React. Omogućava generiranje stranica i na serveru (eng. *backend*) i na klijentu te generiranje statičkih stranica. Next.js rješava mnoge nedostatke React-a, a neki od njih su: automatsko generiranje ruta i datotečnog sustava, podrška za statičko generiranje (eng. *static site generation* - SSG) i serversko iscrtavanje (eng. *server side rendering* – SSR). Također, nudi optimizaciju performansi pomoću učitavanja po potrebi (eng. *lazy loading*) i optimizacije slika. Next.js nudi mogućnost integracije serverskih usluga i API-ja (Application Programming Interface), postižući time mogućnost kreiranja vrlo kompleksnih aplikacija s boljim optimizacijama za internetske pretraživače (eng. *Search Engine Optimization* – SEO) i bržim učitavanjem stranica. Next.js je moguće razvijati kako u JavaScript-u tako i u TypeScript-u. Slika 4 prikazuje tijek podataka i radnji koje se izvršavaju kako bi se od korisničkog zahtjeva došlo do konačnog vizuala.



Slika 4. Next.js - od korisničkog zahtjeva do vizuala [15]

2.1.1. Tailwind CSS i DaisyUI

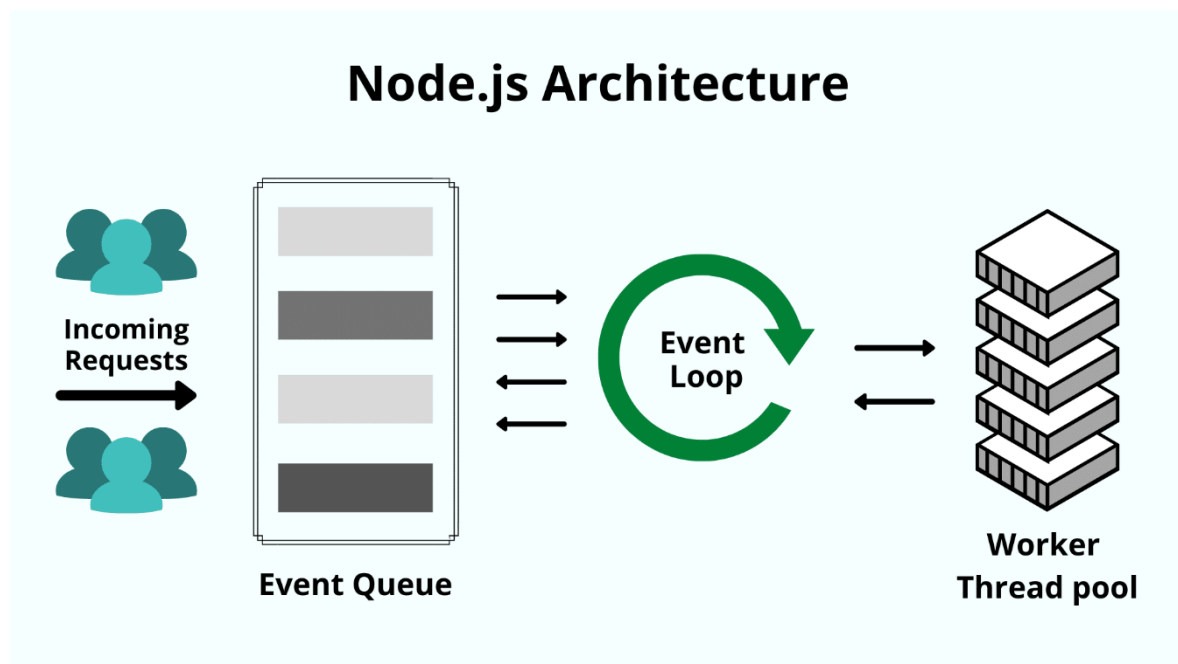
Tailwind CSS [16] je Cascading Style Sheets (CSS) radni okvir koji omogućava brže i efikasnije stiliziranje web stranica. Usmjeren je na korištenje predefiniranih klasa. Na Slika 5 prikazan je primjer korištenja Tailwind CSS-a. Tailwind CSS ima predefinirane klase čijim korištenjem se direktno definiraju stilovi kao što su font, margine, pozadinske boje i slično. Uvelike ubrzava razvoj jer programeri ne moraju pisati vlastita CSS pravila. Cjelokupan dizajn je prvenstveno usmjeren na mobilne uređaje (eng. *mobile first*), a korištenjem klasa kao što su “md” “lg” i ostalih, vrlo lako je moguće definirati @media [17] upite koji utječu na izgled ovisno o veličini uređaja. Dodatak Tailwind CSS-u je DaisyUI [18]. DaisyUI pruža gotove komponente kao npr. kartice, modalne prozore, gumbove i slično. Svi elementi su stilizirani prema Tailwind CSS principima. DaisyUI pojednostavljuje izradu modernih i dinamičkih korisničkih sučelja kombinirajući prednosti Tailwind CSS-a s praktičnošću unaprijed definiranih komponenti.



Slika 5. Tailwind CSS i DaisyUI [19]

2.2. Node.js

Node.js [20] je radno okruženje za JavaScript temeljeno na V8 JavaScript mehanizmu (eng. *engine*). On omogućava izvođenje JavaScript programskog koda izvan bilo kojeg preglednika. Node.js je uvelike poznat po svojoj arhitekturi vođenoj događajima (eng. *event-driven*) te asinkronoj paradigmi zahvaljujući čemu može rukovati s velikim brojem zahtjeva i veza. Node.js aplikacije rade u jednom procesu te se za nove zahtjeve ne stvaraju nove dretve, već se iz bazena dretvi (eng. *Thread pool*) uzimaju prema potrebama. Node-ov ekosustav sadrži velik broj paketa koji su dostupni putem sustava npm (Node Package Manager) te olakšavaju i ubrzavaju razvoj aplikacija. Slika 6 prikazuje arhitekturu Node.js-a.

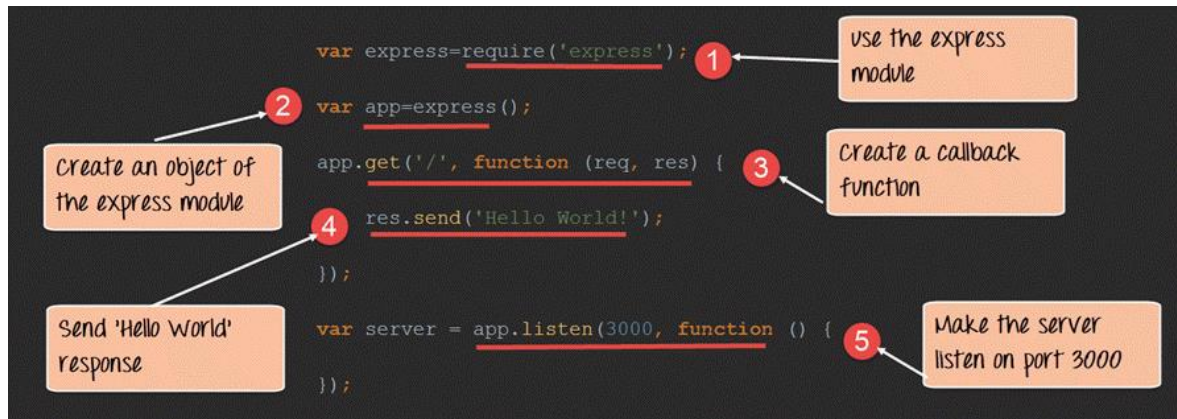


Slika 6. Node.js arhitektura [21]

2.2.1. Express.js

Express.js [22] je najpoznatiji radni okvir za Node.js. Pruža robustan set značajki za izradu sveobuhvatnih web i mobilnih aplikacija. Definira tablicu usmjeravanja koja se koristi za izvođenje različitih radnji na temelju HTTP metode i URL-a. Express.js podržava postavljanje međuprograma (eng. *middleware*) koji omogućava proširenje funkcionalnosti aplikacija bez potrebe za izmjenama osnovnog koda. Međuprogram je obično potprogram, primjerice za provjeru ovlaštenja korisnika za određenu radnju. Express.js se najčešće koristi

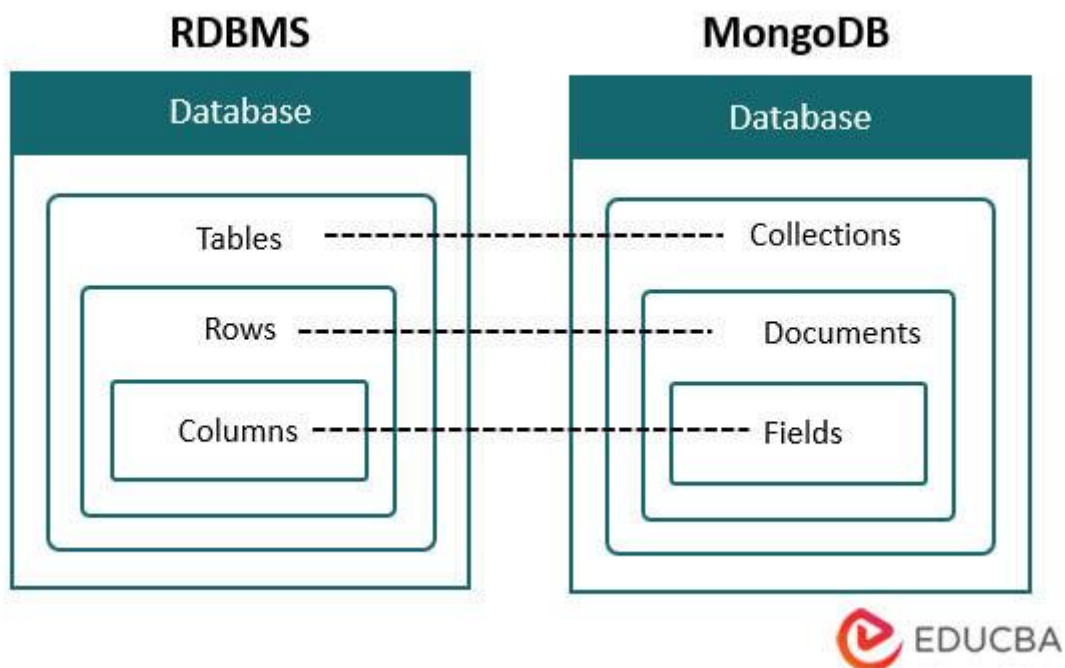
za izradu REST (eng. *Representational State Transfer*) API-ja zbog svoje jednostavnosti i ekstenzibilnosti. Na Slika 7 prikazan je primjer programskog koda. U samo nekoliko linija moguće je kreirati osnovnu verziju web aplikacije.



Slika 7. Primjer jednostavne Express.js aplikacije [23]

2.3. MongoDB

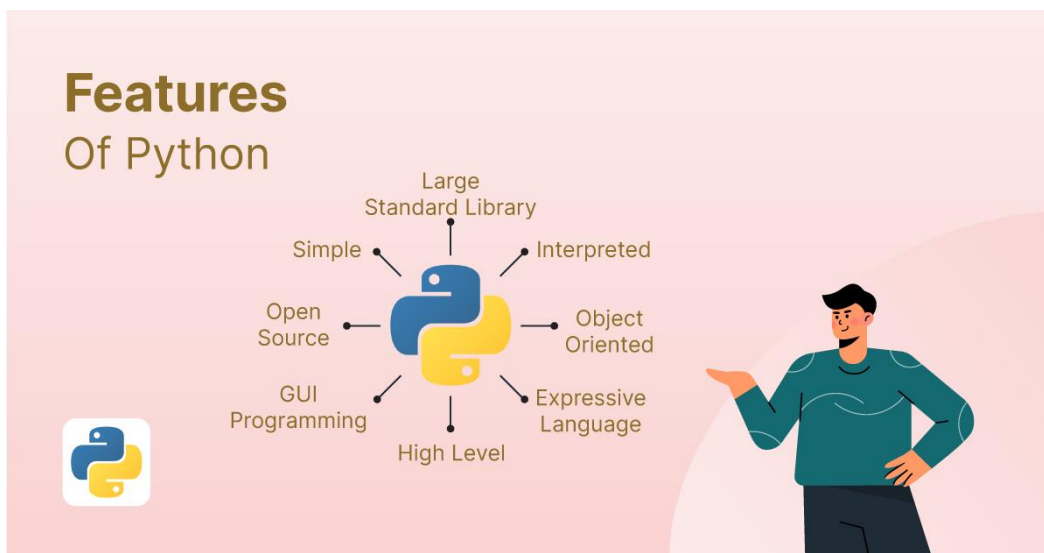
MongoDB [24] je jedna od najpoznatijih i najpopularnijih NoSQL baza podataka. Pohranjuje podatke u formatu sličnom JSON-u, nazvanom BSON (Binary JSON). Namijenjen je za pohranu velikih količina podataka i podložan je horizontalnom skaliranju. Iznimno je poznat po fleksibilnosti i jednostavnosti korištenja, omogućavajući programerima da brzo iteriraju i mijenjaju strukturu podataka bez potrebe za složenim migracijama podataka i slično. Jedan zapis odnosno dokument u MongoDB kolekciji je gotovo identičan zapisu u nekoj relacijskog bazi podataka te je danas korištenjem raznih programa vrlo jednostavno i bezbolno moguće migrirati podatke u oba smjera (iz NoSQL baze u SQL i obrnuto). Podaci su u MongoDB bazi otporni na kaskadna brisanja. Slika 8 prikazuje paralelu između relacijske i MongoDB baze podataka.



Slika 8. Poveznica SQL i noSQL baza podataka [25]

2.4. Python

Python [26] je interpretirani programski jezik, pri čemu se programski kod izvršava liniju po liniju koristeći interpreter, a ne prevodi se u strojni kod kao prevođeni jezici poput C-a. Postao je jedan od najpopularnijih jezika zbog svoje jednostavnosti i široke primjene. Podržava više paradigmi programiranja kao što su proceduralno, objektno orijentirano, funkcionalno i drugo. Shodno tome, može se koristiti za različite vrste programiranja, uključujući web razvoj, znanstvene izračune, analizu podataka, umjetnu inteligenciju (eng. *Artificial Intelligence* – AI) i sl.. Python dolazi s velikim brojem standardnih biblioteka koje olakšavaju razvoj. Koristi automatsko upravljanje memorijom putem sakupljača smeća (eng. *Garbage Collector*) što programerima olakšava upravljanje resursima. Slika 9 prikazuje glavne značajke Python-a.



Slika 9. Python značajke [27]

2.4.1. Flask

Flask [28] je minimalni i lagani web radni okvir za Python. Omogućava brzo i jednostavno razvijanje web aplikacija. Stvoren je za izradu web aplikacija bez nametanja previše ograničenja ili složenosti. Podržava razna proširenja koja omogućuju dodavanje raznih funkcionalnosti, a neke od njih su: autentifikacija, objektno-relacijsko preslikavanje (eng. *Object-Relational Mapping* - ORM) i upravljanje obrascima. Zbog svoje jednostavnosti, pogodan je za male do srednje velike projekte. Slika 10 prikazuje glavne značajke Flask-a.



Slika 10. Flask značajke [29]

2.4.2. SentenceTransformer

SentenceTransformer [30] je Python biblioteka za pretvaranje tekstualnih podataka u numeričke reprezentacije (eng. *embedding*). Dobivene numeričke reprezentacije se mogu koristiti za razne obrade prirodnog jezika (NLP). Neki od načina korištenja *embedding*-a su: pretraživanje semantičke sličnosti, *clusterization* i klasifikacija teksta. SentenceTransformer koristi modele koji su unaprijed trenirani na velikim korpusima teksta. Prolazeći kroz razne slojeve transformera, model koristi informacije iz cijele rečenice uzimajući u obzir kontekst svake riječi. Konačni *embedding* se dobiva agregiranjem reprezentacije svakog tokena u rečenici.

2.4.3. Faiss

Faiss [31] (Facebook AI Similarity Search) je biblioteka otvorenog koda razvijena od strane Facebook AI tima. Omogućava efikasno pretraživanje i povrat visoko dimenzioniranih vektora ovisno o modelu. Faiss je dizajniran za rad s velikom količinom podataka omogućavajući brze i skalabilne operacije pretraživanja sličnosti. Posjeduje podršku za grafičku jedinicu za obradu (eng. *graphics proccesing unit* – GPU) što je vrlo korisno za velike skupove podataka. Za efikasan rad su zaslužne različite vrste indeksa za pretraživanje spremljenih vektora. Dvije glavne kategorije indeksa su egzaktni i aproksimativni indeksi. Egzaktni indeksi su značajno sporiji jer imaju visoku stopu preciznosti dok aproksimativni indeksi rade različite metode kao što su kvantizacija i hashiranje kako bi se ubrzalo pretraživanje. Za pretraživanja se najčešće koristi kosinusova sličnost ili euklidska udaljenost.

3. Model baze podataka

3.1. Opis kolekcija

Sljedeće tablice prikazuju i detaljno opisuju attribute i njihove tipove.

Kolekcija: **Korisnik**

Kolekcija Korisnik sadrži podatke vezane za korisnike registrirane u aplikaciji. Shema kolekcije sadrži atribut *email* koji je obavezan i jedinstven te *categories* koji predstavlja kategorije koje je korisnik označio omiljenima. Atribut *password* predstavlja hashirani zapis lozinke dok atributi *createdAt* i *updatedAt* predstavljaju datume i vremena kreiranja odnosno ažuriranja korisnikovih podataka. Slika 11 prikazuje izgled kolekcije Korisnik.

Atribut	Tip podatka	Opis
_id	ObjectID	ID korisnika
email	String	Email korisnika
categories	[ObjectID]	Popis kategorija
password	String	Lozinka korisnika
createdAt	Date	Datum i vrijeme kreiranja
updatedAt	Date	Datum i vrijeme ažuriranja

Slika 11. Kolekcija Korisnik

Kolekcija: **Kategorija**

Kolekcija Kategorija sadrži podatke vezane za kategorije videozapisa. Shema kolekcije sadrži attribute *name* koji predstavlja naziv kategorije, *url* koji predstavlja link na sliku kategorije te attribute *createdAt* i *updatedAt* koji predstavljaju datume i vremena kreiranja odnosno ažuriranja korisnikovih podataka. Slika 12 prikazuje izgled kolekcije Kategorija.

Atribut	Tip podatka	Opis
<code>_id</code>	ObjectID	ID kategorije
<code>name</code>	String	Naziv kategorije
<code>url</code>	String	Link na sliku
<code>createdAt</code>	Date	Datum i vrijeme kreiranja
<code>updatedAt</code>	Date	Datum i vrijeme ažuriranja

Slika 12. Kolekcija Kategorija

Kolekcija: **Videozapis**

Kolekcija Videozapis sadrži podatke vezane za videozapise koje korisnici mogu gledati. Shema kolekcije sadrži atribut *title* koji predstavlja naslov videozapisa, *description* koji predstavlja opis videozapisa, *youtubeURL* koji predstavlja link videzapisa koji vodi na YouTube platformu i atribut *tags* koji predstavlja oznake povezane s videzapisom. Osim spomenutih, kolekcija sadrži atribut *category* koji predstavlja svojevrsni strani ključ, odnosno poveznicu između videzapisa i kategorije kojoj pripada, atribut *duration* koji predstavlja ukupno trajanje videozapisa, *thumb* koji predstavlja link na sličicu videozapisa te atribut *channelId* koji predstavlja identifikator kanala kreatora i vlasnika videozapisa. Također sadrži i atribut *youtubeVideoId* koji predstavlja identifikator videozapisa na platformi YouTube, *numberOfTimesRecommended* koji predstavlja ukupan broj preporuka tog videozapisa, *numberOfTimesWatched* koji predstavlja ukupan broj klikova odnosno gledanja videozapisa te attribute *createdAt* i *updatedAt* koji predstavljaju datume i vremena kreiranja odnosno ažuriranja korisnikovih podataka. Slika 13 prikazuje izgled kolekcije Videozapis.

Atribut	Tip podatka	Opis
_id	ObjectId	ID videa
title	String	Naslov videa
description	String	Opis videa
youtubeURL	String	URL videa na YouTube-u
tags	String	Oznake povezane s videom
category	ObjectId	ID kategorije
duration	Number	Trajanje videa u sekundama
thumb	String	URL sličice videa
channelId	String	ID kanala
youtubeVideoId	String	ID videa na YouTube-u
numberOfTimesRecommended	Number	Broj preporuka videa
numberOfTimesWatched	Number	Broj pregleda videa
createdAt	Date	Datum i vrijeme kreiranja
updatedAt	Date	Datum i vrijeme ažuriranja

Slika 13. Kolekcija Videozapis

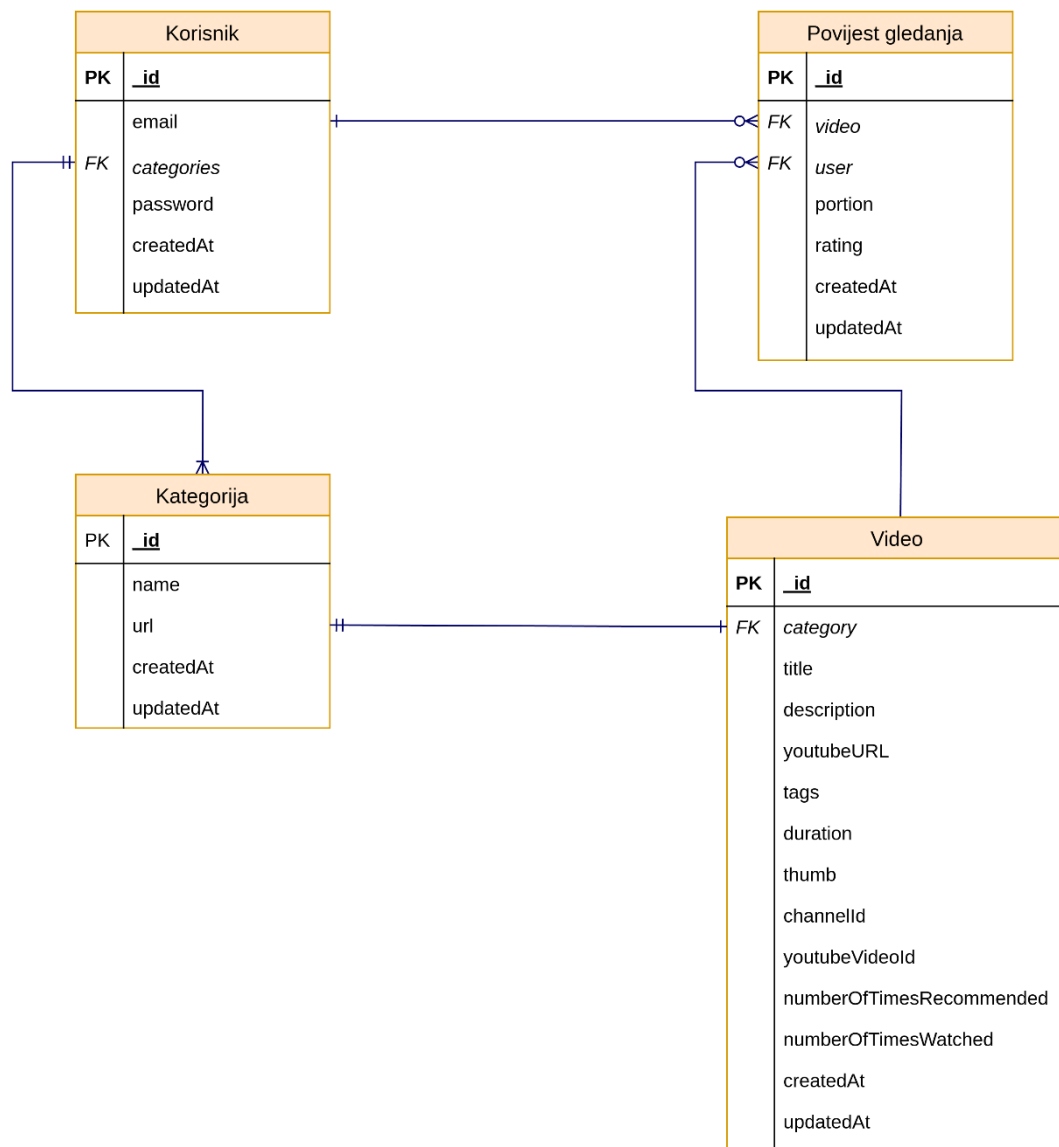
Kolekcija: **Povijest gledanja**

Kolekcija Povijest gledanja predstavlja podatke odnosno zapise svakog pojedinog gledanja videozapisa. Shema kolekcije sadrži atribut *video category* koji predstavlja svojevrsni strani ključ, odnosno poveznicu između zapisa gledanja i videozapisa koji je gledan te atribut *user* koji predstavlja svojevrsni strani ključ, odnosno poveznicu između zapisa gledanja i korisnika koji je taj video gledao. Također sadrži atribut *portion* koji predstavlja postotak videozapisa koji je pogledan, *rating* koji predstavlja ocjenu kojom je korisnik ocijenio video te attribute *createdAt* i *updatedAt* koji predstavljaju datume i vremena kreiranja odnosno ažuriranja korisnikovih podataka. Slika 14 prikazuje izgled kolekcije Povijest gledanja.

Atribut	Tip podatka	Opis
_id	ObjectId	ID pregleda
video	ObjectId	ID videa
user	ObjectId	ID korisnika
portion	Number	Dio videa koji je pregledan
rating	Number	Ocjena videa
createdAt	Date	Datum i vrijeme kreiranja
updatedAt	Date	Datum i vrijeme ažuriranja

Slika 14. Kolekcija Povijest gledanja

Slika 15 prikazuje relacijski dijagram (eng. *Entity Relationship Diagram – ER Diagram*) baze podataka, poznat i kao ER dijagram. Na dijagramu su prikazani entiteti (tablice) kao pravokutnici, a veze između njih kao linije koje ih povezuju. Entiteti su označeni imenima. Veze između entiteta mogu biti unarne (gdje sudjeluje jedan entitet), binarne (gdje sudjeluju dva entiteta) ili n-arne (gdje sudjeluje više od dva entiteta).

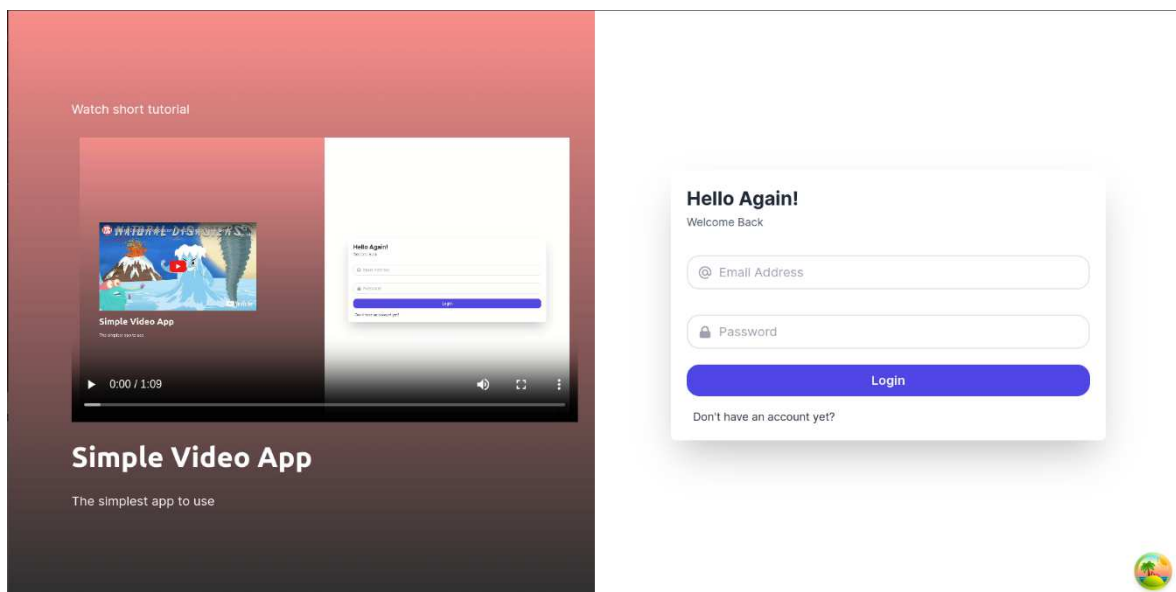


Slika 15. ER dijagram baze podataka

4. Opis korištenja aplikacije

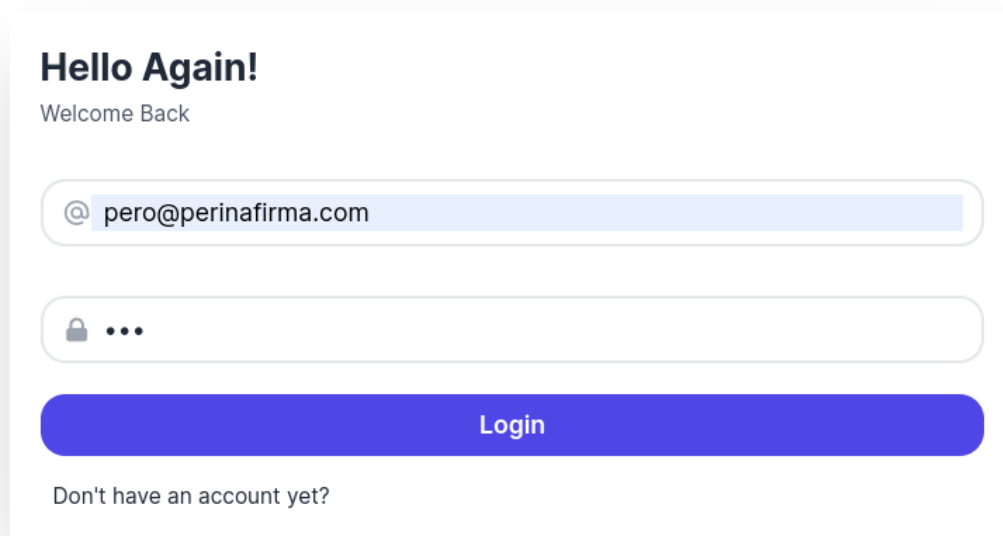
4.1. Prijava i registracija

Aplikacija razvijena u okviru ovog diplomskog rada na svojoj početnoj stranici od korisnika traži da se prijavi ili registrira ako ne posjeduje korisnički račun. Početna stranica također nudi kratak video koji daje pregled i upute za korištenje aplikacije. Na desnoj strani Slika 16 vidi se forma za unos email-a i lozinke. To su podatci koji se na serverskoj strani koriste za autentifikaciju korisnika.



Slika 16. Izgled stranice za prijavu u aplikaciju

U situaciji kada korisnik nema korisnički račun, moguće je klikom na poveznicu "Don't have an account yet?" korisnika odvesti na formu za registraciju što je vidljivo na Slika 17 i Slika 18. Korisnik mora unijeti email te lozinku i ponovljenu lozinku. Nakon toga, na red dolazi izbor kategorija koje se korisniku sviđaju. Slika 19 prikazuje prvi korak u registraciji, a Slika 20 prikazuje odabir kategorija odnosno korisnikove preferencije.

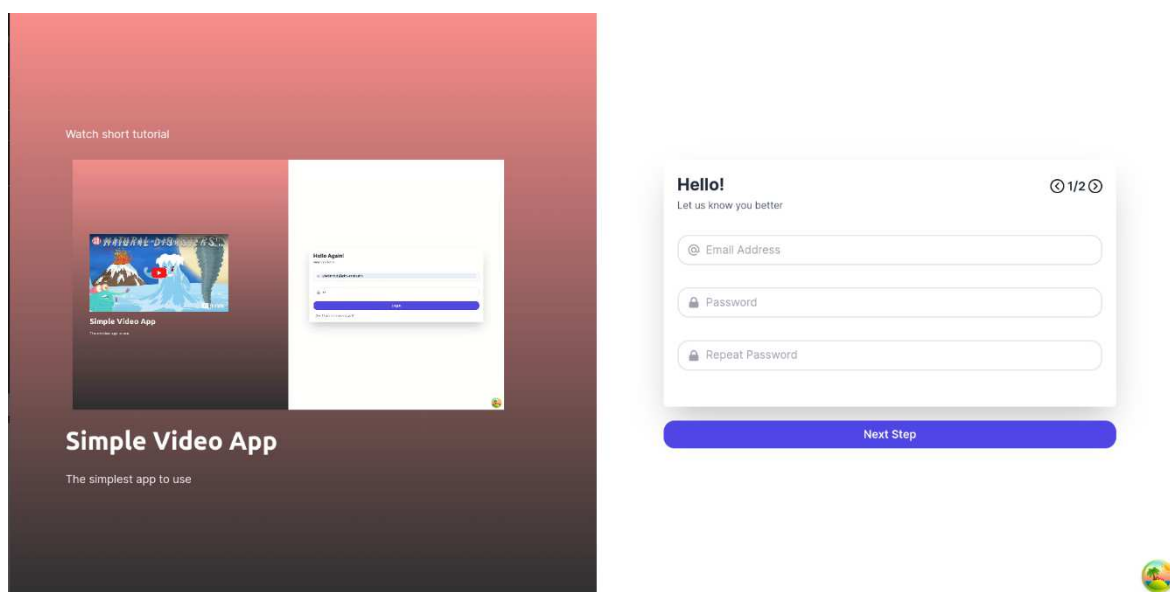


Hello Again!
Welcome Back

Login

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

Slika 17. Unos podataka za prijavu



Watch short tutorial

Simple Video App
The simplest app to use

Hello!
Let us know you better

Next Step

1/2

Slika 18. Izgled stranice za registraciju

Hello! ⏪ 1/2 ⏩

Let us know you better

@ test@test.com

•••••

•••••

Next Step

Slika 19. Unos podataka za registraciju – 1. korak

Hello! ⏪ 2/2 ⏩

Pick your favorite categories

History

Art & Crafts

Sports

Science

Nature

Register

Slika 20. Unos podataka za registraciju – 2. korak

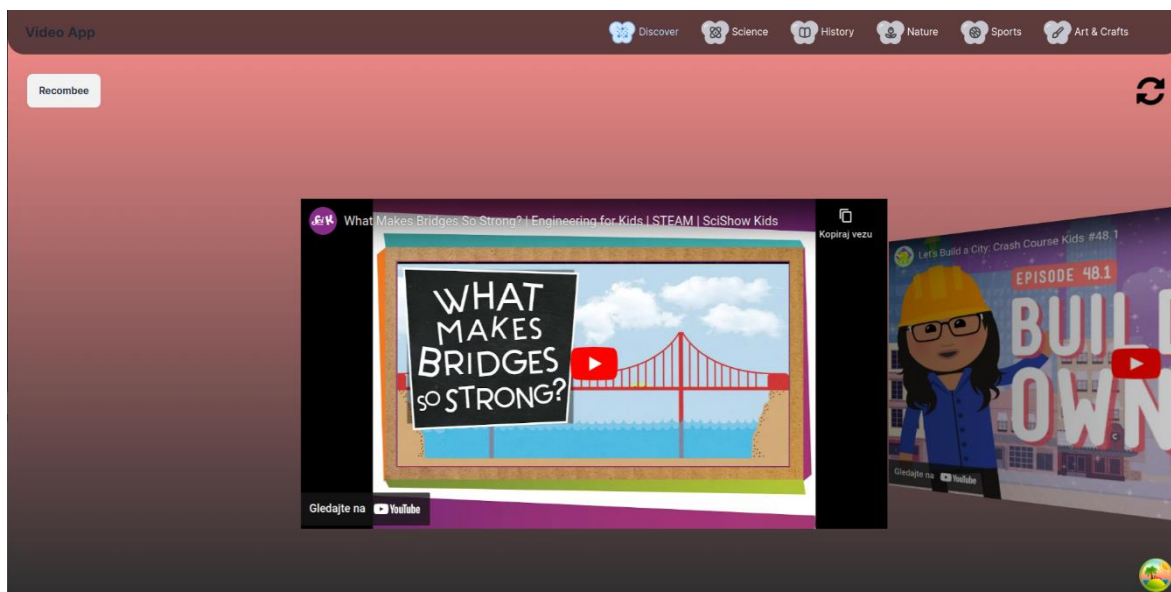
Kvalitetan izbor kategorija vrlo je bitan, on se koristi u hladnom startu aplikacija ako je aktivan neki od sustava Recombee ili Amazon Personalize. Sustav za preporuke Recombee

nudi mogućnost stvaranja raznih “filtera”. Filteri svoj najveći potencijal ostvaruju upravo u hladnom startu kada se po nekim parametrima moraju izabrati videozapisi. U slučaju ovog sustava, s obzirom na to da korisnik bira kategorije koje mu se sviđaju te da svaki video pripada nekoj kategoriji, u početku se najveću pažnju stavlja na to da korisnik dobiva samo videozapise koji su u odabranim kategorijama. ReQL je naziv jezika kojim se definiraju pravila filtriranja kod sustava Recombee. Sustav Amazon Personalize također nudi mogućnost filtriranja po parametrima, a upiti uvelike podsjećaju na SQL upite. Sustav vlastite izrade u hladnom startu daje slučajno odabrane videozapise te svakom idućom interakcijom kreće s personaliziranim preporukama koje bivaju bolje što se sustav više koristi. Bitno je napomenuti kako svi sustavi bilježe međusobno jednake interakcije. Primjerice, interakcija koja se koristi za spremanje postotka pogledanog videozapisa bilježi se istovremeno u sve tri inačice sustava za preporuku.

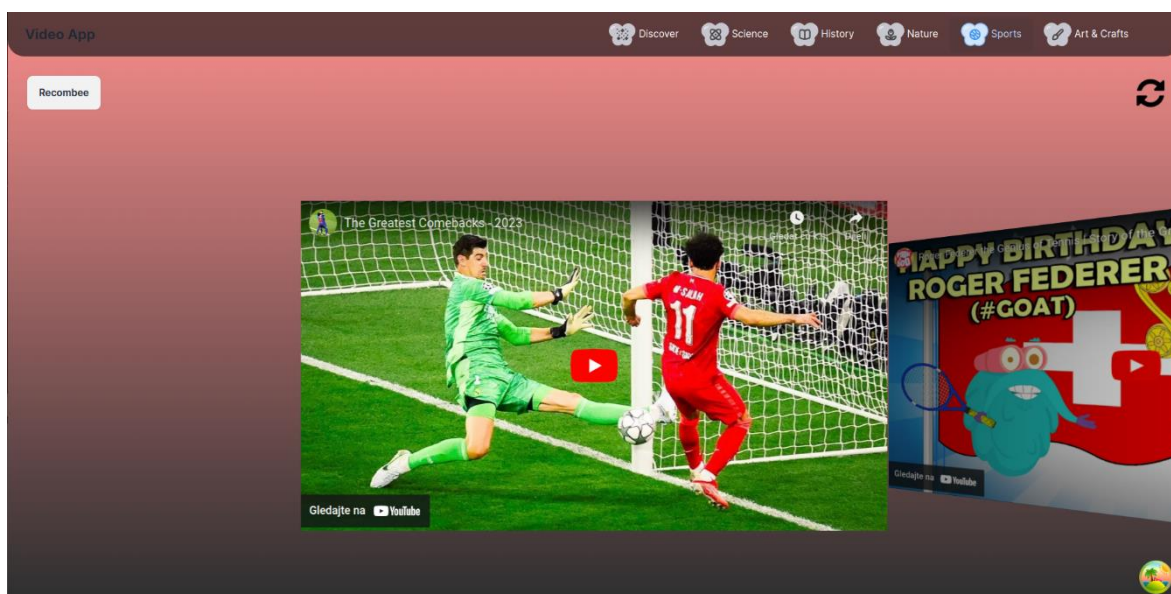
4.2. Pregled sučelja s video sadržajem

4.2.1. Rješenje za desktop računala

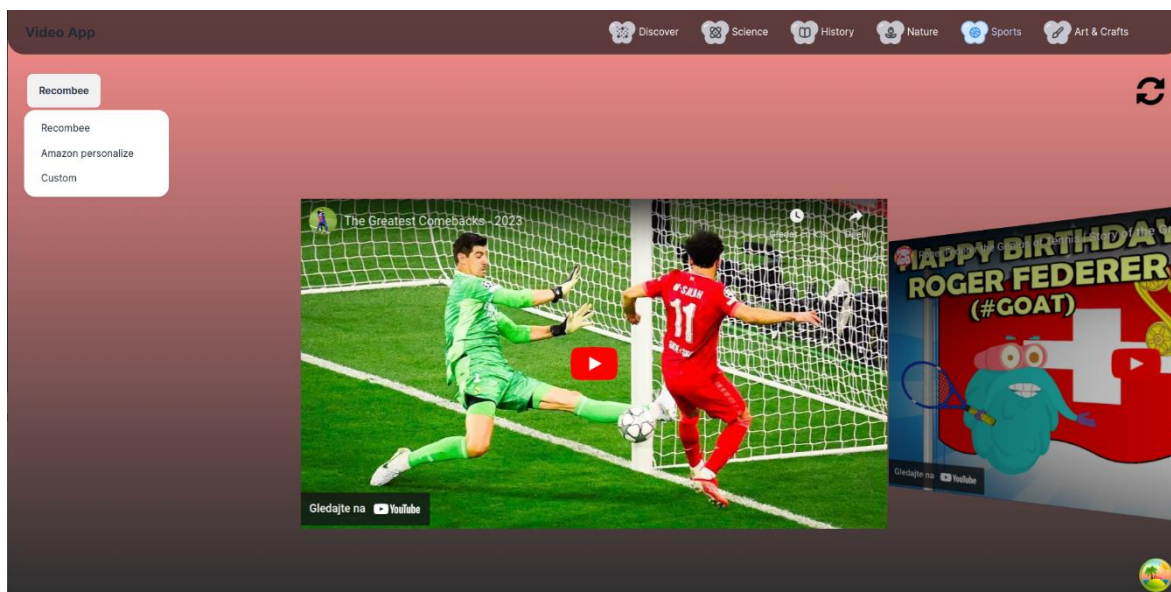
Slika 21 prikazuje izgled početnog sučelja aplikacije. Na početnom sučelju se nalazi lista videozapisa koje je moguće pogledati, gumb za osvježavanje liste, niz kategorija koje je moguće odabrati te niz sustava za predlaganje videozapisa koji se temelje na različitim algoritmima. Slika 22 i Slika 23 prikazuju mogućnost odabira drugih kategorija kao i listu sustava za preporučivanje video sadržaja.



Slika 21. Izgled stranice nakon prijave u aplikaciju



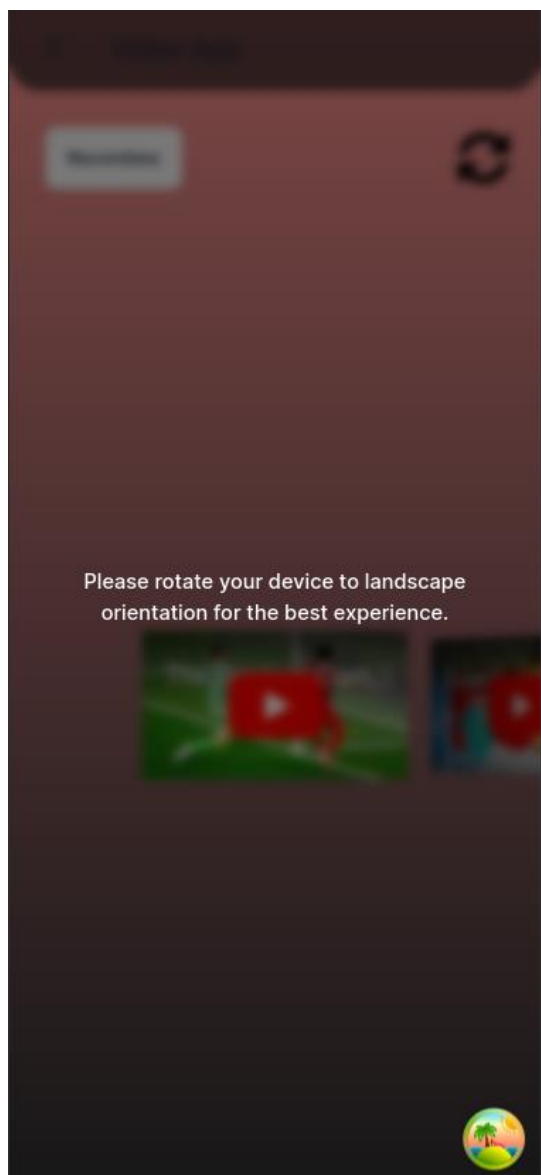
Slika 22. Izgled stranice nakon odabira kategorije



Slika 23. Izgled stranice nakon otvaranja izbornika sustava za predlaganje sadržaja

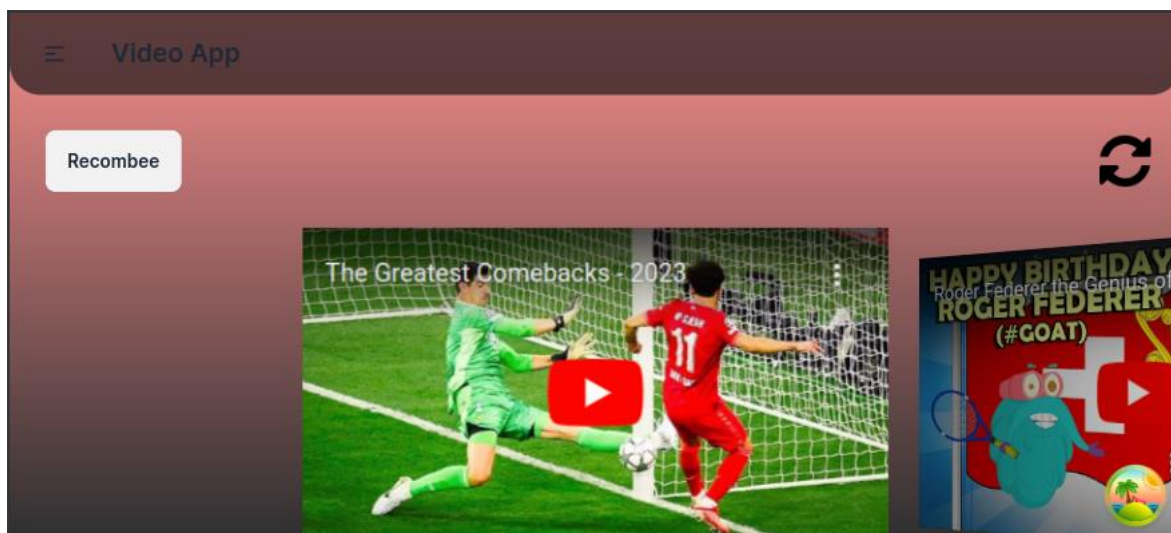
4.2.2. Rješenje za mobilne uređaje

Danas je nužno kreirati aplikacije responzivnog dizajna (prilagodljive različitim veličinama ekrana). S obzirom na to da je u nekim slučajevima udio mobilnih korisnika značajno veći u odnosu na “web desktop” korisnike, ponekad je važnije kreirati *mobile first* aplikacije. U slučaju ove aplikacije, korisničko iskustvo u pejzažnom načinu rada značajno je bolje i reprezentativnije te se iz tog razloga korisnika forsira i upućuje na rotaciju mobilnog uređaja. Slika 24 prikazuje uputu korisniku za rotaciju mobilnog uređaja.



Slika 24. Uputa korisniku za rotaciju mobilnog uređaja

Na Slika 25 i Slika 26 moguće je vidjeti izgled aplikacije na mobilnom uređaju u pejzažnom načinu rada. Prethodno spomenute slike također demonstriraju i prikazuju responzivnost te pristupačnost korištenja na raznim platformama i ekranima.



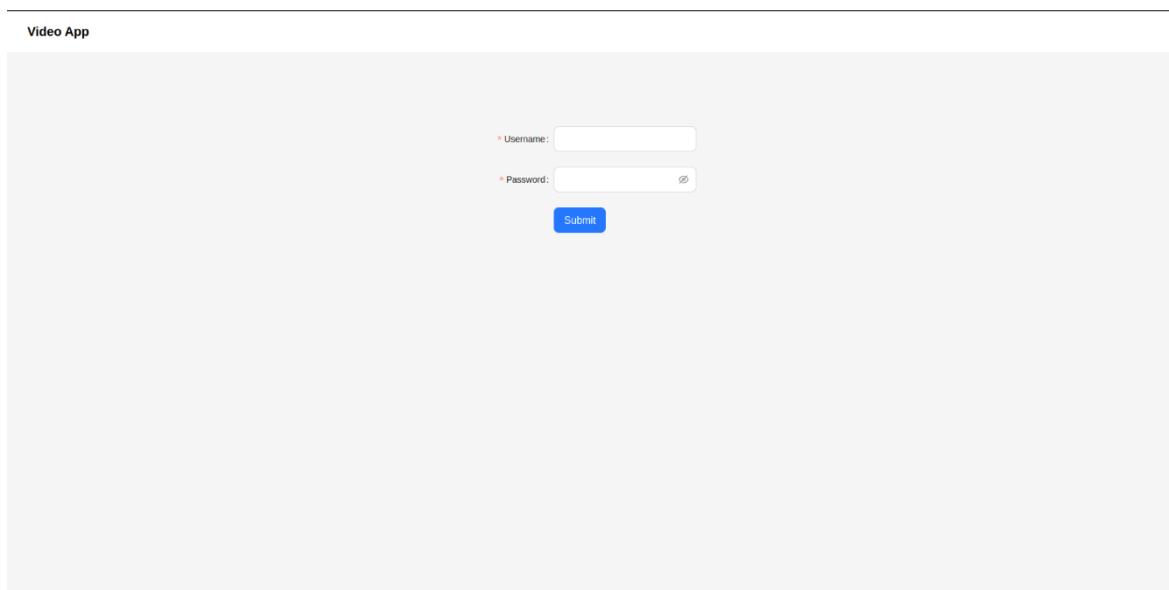
Slika 25. Izgled aplikacije na mobilnom uređaju u pejzažnom načinu rada



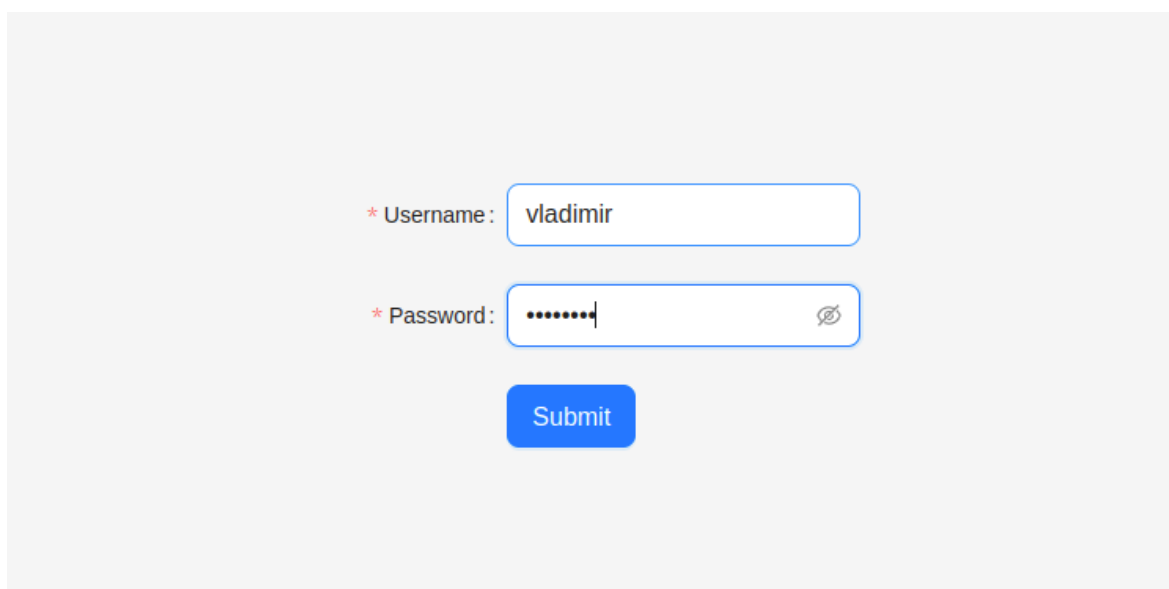
Slika 26. Izgled aplikacije na mobilnom uređaju u pejzažnom načinu rada s otvorenim izbornikom kategorija

4.3. Pregled aplikacije za unos i izmjenu video sadržaja

Za upravljanje bazom videozapisa te organizaciju istih u kategorije zaslužna je administratorska aplikacija. Aplikacija služi za unos, uređivanje, brisanje i detaljan pregled videozapisa. Početna stranica aplikacije koja je prikazana na Slika 27 prikazuje formu za unos korisničkog imena i lozinke. Formu za unos moguće je dodatno vidjeti na Slika 28.



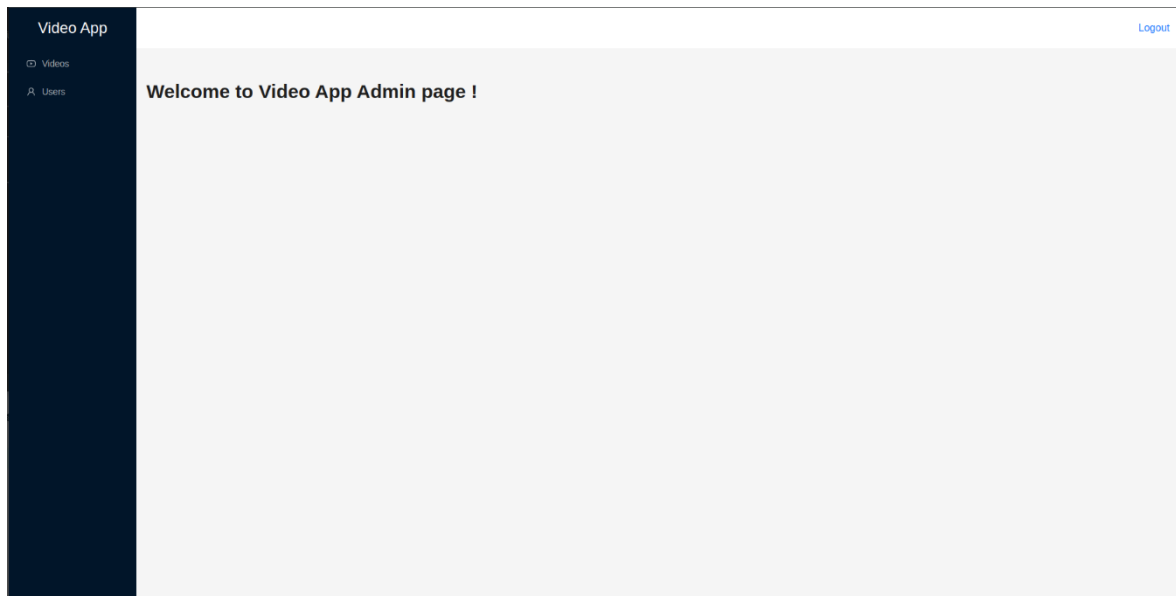
Slika 27. Izgled početne stranice aplikacije za administratore



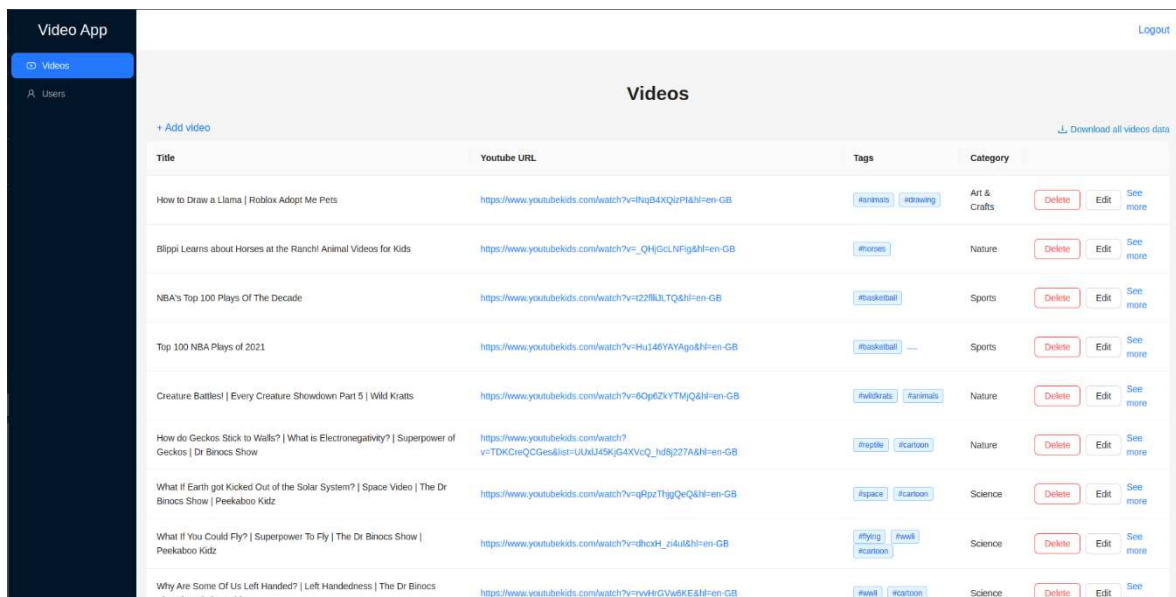
Slika 28. Forma za unos podataka za prijavu

Unosom točnih podataka za prijavu aplikacija preusmjerava korisnika na sučelje za upravljanje podacima u bazi podataka. Početno sučelje na koje je korisnik preusmjeren nakon prijave moguće je vidjeti na Slika 29. Klikom na poveznicu "Videos" aplikacija preusmjerava korisnika na stranicu, prikazanu na Slika 30, koja sadržava listu videozapisa i razne mogućnosti za manipulaciju istima. Klikom na gumb "+ Add video" otvara se prozor, prikazan na Slika 31, za unos podataka o videozapisu. S obzirom na to da je aplikacija povezana s YouTube API-jem korisniku se nudi mogućnost dohvaćanja

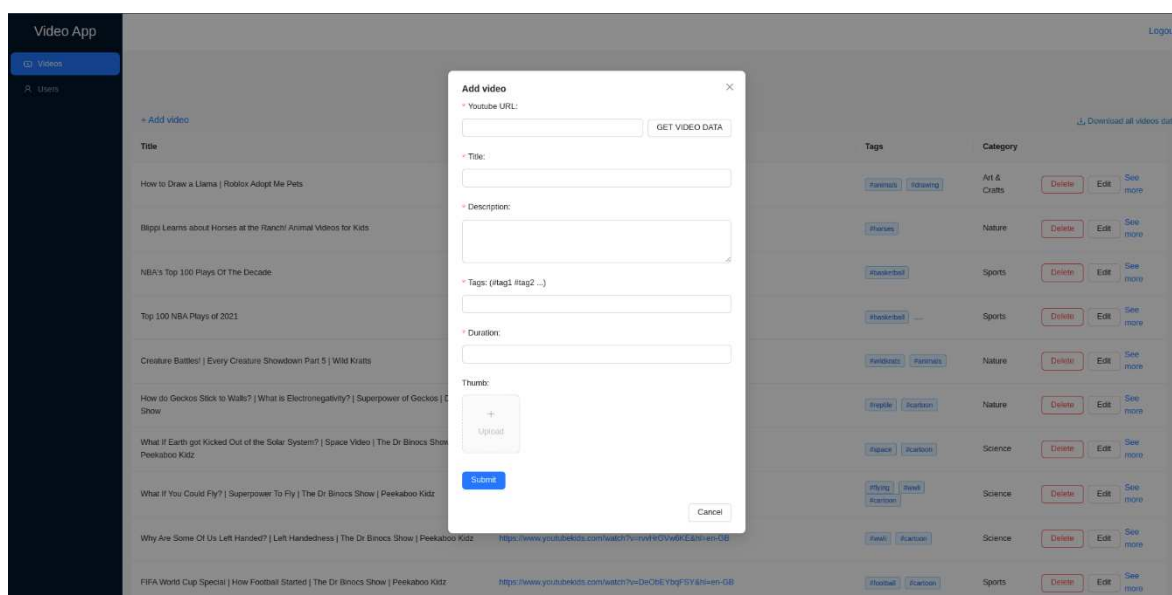
podataka o videozapisu ako on to želi. Dohvaćeni podaci se po želji mogu mijenjati. Klikom na gumb "Submit" videozapis se sprema u bazu.



Slika 29. Početno sučelje aplikacije za administratore nakon uspješne prijave



Slika 30. Izgled sučelja nakon klika na poveznicu "Videos"



Slika 31. Dodavanje novog videozapisa

Add video

* Youtube URL:

<https://www.youtube.com/watch?v=UBVV8pch1>
GET VIDEO DATA

* Title:

The Science of Thinking

* Description:

Nathan Hansen, Donal Botkin, Tony Fadell, Zach Mueller, Ron Neal
Support Veritasium on Patreon: <http://bit.ly/VePatreon>

* Tags: (#tag1 #tag2 ...)

* Duration:

730

Thumb:

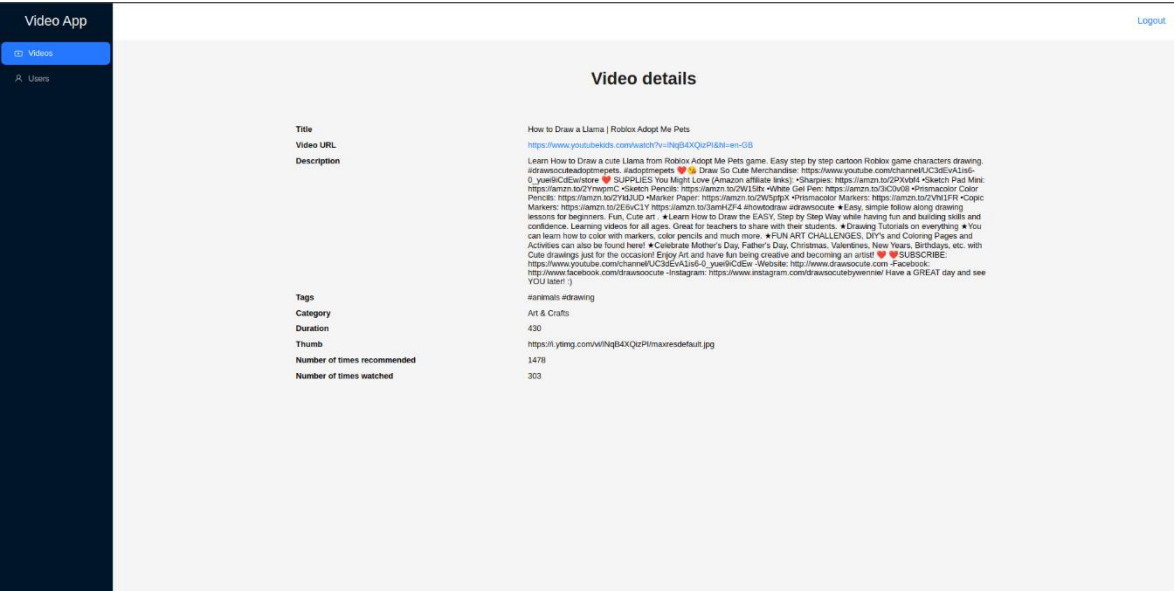


Submit

Cancel

Slika 32. Ispunjena forma za dodavanje novog videozapisa

Lista videozapisa prikazuje samo najvažnije podatke za pojedini videozapis, ali klikom na gumb “See more” otvara se nova stranica koja prikazuje detalje videozapisa. Izgled stranice prikazan je na Slika 33.



Slika 33. Detalji videozapisa

5. Usporedba implementiranih sustava

Algoritmi za preporučivanje sadržaja postali su ključni alat u digitalnom svijetu, posebno u platformama za emitiranje video sadržaja. Neke od izabranih gotovih rješenja su Recombee i Amazon Personalize. U okviru ovog rada je kreirana i vlastita inačica sustava za preporučivanje koristeći programski jezik Python te razne biblioteke.

Polazišna točka cjelokupne aplikacije je bazična serverska aplikacija u koju su integrirani različiti sustavi za preporučivanje sadržaja (Recombee, Amazon Personalize i vlastita inačica). O pojedinom sustavu će više riječi biti u nastavku. Baza serverska aplikacija sprema podatke o korisnicima, njihove preferencije, povijest gledanja, interakcije i drugo, koji se potom koriste u podsustavima za preporuke video sadržaja. Cilj je bio stvoriti aplikaciju koja nudi izbor raznih podsustava za preporuke te mogućnost lake promjene okoline, odnosno podsustava za preporuke koji je aktivan u trenutku.

5.1. Amazon Personalize

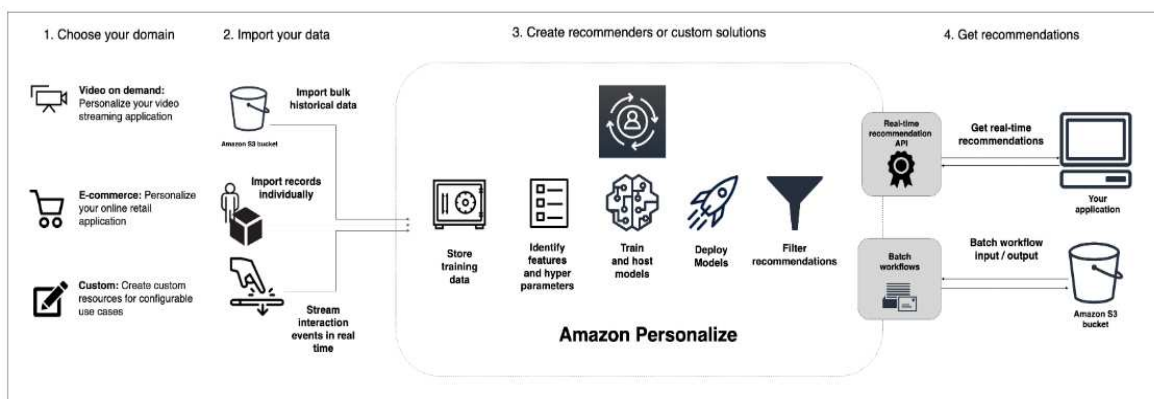
Amazon Personalize je upravljana usluga preporučivanja razvijena od strane Amazon Web Services (AWS) [32]. Zahvaljujući iscrpno velikoj dokumentaciji, moguće je stvoriti rješenje potpuno personalizirano i prilagođeno poslovnoj potrebi bez opsežnog znanja strojnog učenja. Jedna od ključnih prednosti Amazon Personalize-a je u mogućnosti integracije s drugim AWS uslugama kao što su S3 [33], Redshift [34] i sl.. On tada postaje idealan izbor jer omogućuje jednostavan pristup svim podacima servisa bez dodatnih napora. Amazon Personalize nam također nudi mogućnost automatskog upravljanja modelima tj. automatsko treniranje, optimiziranje i implementiranje modela. Mogućnost preporuka u stvarnom vremenu je još jedna od opcija koje nudi ova usluga. Točni algoritmi programerima nisu dostupni, no za implementaciju raznih rješenja i poslovnih potreba postoji izbor između raznih “recepata”. Neki od recepata su “popular items”, “related items”, “personalized actions” i drugi. Kako bi algoritam odnosno rješenje bilo optimizirano i točno, potrebno je sustavu dostaviti određene podatke tj. EVENT TYPE-ove i EVENT VALUE-e. Slika 34 prikazuje slijed integracije sustava Amazon Personalize.

Prednosti:

- Skalabilnost: Kada broj korisnika i podataka naglo poraste, Amazon Personalize je u mogućnosti prilagoditi se broju zahtjeva i količini podataka kako bi odziv bio konstantno visok.
- Automatizacija: Programer ne mora imati znanja o strojnom učenju, može se potpuno posvetiti implementaciji ostalih funkcionalnosti.
- Integracija s ostalim elementima AWS-a.

Nedostaci:

- Trošak: Može biti prilično skup, pogotovo za manje tvrtke ili sustave s ograničenim budžetom. Vrlo ograničen probni period koji ne uključuje sve osnovne funkcionalnosti za isprobavanje i testiranje sustava.
- Vrlo kompliciran sustav s vrlo velikim brojem značajki i elemenata.
- Dokumentacija bez strukture i slijeda - loše za učenje kroz isprobavanje okoline.



Slika 34. Amazon Personalize [2]

5.2. Recombee

Recombee je platforma u oblaku (eng. *cloud based*) koja pruža usluge preporučivanja koristeći napredne algoritme strojnog učenja, uključujući kolaborativno i sadržajno filtriranje te hibridne pristupe. Nudi mogućnost preporuke različitih vrsta sadržaja, uključujući videozapise, glazbu, proizvode i članke. Vrlo je pogodna za integraciju s već postojećim sustavima putem REST API-ja. Recombee dokumentacija točno specificira

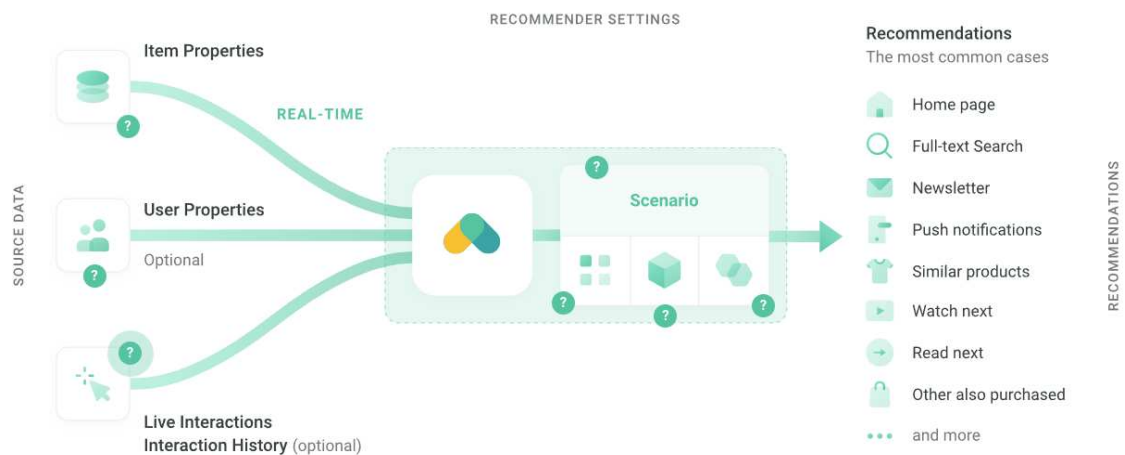
kojim slijedom i koje podatke je sustavu potrebno dostavljati kako bi preporuke bile maksimalno ispravne i personalizirane. Slika 35 prikazuje postavke sustava Recombee.

Prednosti:

- Detaljna, strukturirana i opsežna dokumentacija.
- Korisnička podrška uvijek dostupna.
- Analitika i uvidi u stvarnom vremenu.
- Cijena se mijenja ovisno o potrošnji (eng. *pay as you go*) te postoji besplatna inačica do određenog broja preporuka, korisnika i artikala.
- Jednostavna integracija putem API-ja i skupa pribora za razvijanje aplikacija (eng. *software development kit – SDK*).

Nedostaci:

- Može biti prilično skup za vrlo velike sustave.
- Potrebna komunikacija sa korisničkom podrškom za dodatnu prilagodbu modela (eng. *fine-tuning*) u slučaju nestandardnih ideja korištenja



Slika 35. Recombee [1]

5.3. Izrada vlastitog rješenja

U ovom poglavlju bit će predstavljena i detaljno opisana implementacija vlastitog rješenja. Ispod se nalazi programski kod koji je zaslužan za odabir najboljih preporuka za korisnika. Nakon programskog koda slijedi objašnjenje implementacije i teorijska podloga.

```
class VideoRecommender:
    def __init__(self, model_path, df_videos):
        self.model = SentenceTransformer(model_path)
        video_pairs = [(x,y) for x,y in
            zip(df_videos.tags.values, df_videos.title.values)]

        encoded_videos = self.model.encode(video_pairs)
        self.num_docs, self.dim = encoded_videos.shape

        index = faiss.IndexFlatL2(self.dim)
        index.add(encoded_videos)
        self.index = index
        self.id_2_index = {str(video_id):faiss_index for
            faiss_index, video_id in
            enumerate(df_videos._id.values)}

        self.index_2_id = {faiss_index:str(video_id) for
            video_id, faiss_index in self.id_2_index.items()}

    def reccomend_function(self, video_ids):
        liked_videos_index = [self.id_2_index[video_id] for
            video_id in video_ids]

        saved_embeddings =
            faiss.rev_swig_ptr(self.index.get_xb(),
                self.num_docs*self.dim).reshape(self.num_docs,
                self.dim)

        liked_embeddings = saved_embeddings[liked_videos_index]
        D, I = self.index.search(liked_embeddings, 5)
        index_dist = sorted([(dist, index) for dist_list_i,
            index_list_i in zip(D,I) for dist, index in
            zip(dist_list_i, index_list_i)])
```

```

indexes = [index for dist, index in index_dist if index
not in liked_videos_index]

indexes_cleaned = list(dict.fromkeys(indexes))
recommended_video_ids = [self.index_2_id[ind] for ind
in indexes_cleaned] return recommended_video_ids

```

Nalaže se mogućnost kreiranja vlastitog rješenja odnosno vlastitog sustava za preporučivanje video sadržaja. Korištenjem programskog jezika Python te pomoćnih biblioteka kreirano je vlastito rješenje.

Kreirani sustav za preporučivanje video sadržaja temeljen je na sličnosti sadržaja (eng. *content based*). Preporuke za korisnika se generiraju prema zadnjim uspješnim interakcijama, odnosno video sadržajima koje je korisnik označio sa “sviđa mi se”. Kada ne postoji dovoljan broj videozapisa označenih sa “sviđa mi se”, u obzir se uzimaju zapisi o gledanju s visokim postotkom preglednosti. Na temelju tih videozapisa traže im se oni najslbližnji. Svi videozapisi su reprezentirani u vektorskom obliku i pohranjeni su u vektorsku bazu Faiss. Vektorska reprezentacija svakog videozapisa se kreira temeljem oznaka (eng. *tag*) i naslova videozapisa. Oznake i naslov videozapisa su ulaz u neuronski jezični model koji na izlazu daje vektorsku reprezentaciju odnosno videozapis zapisan kao vektor. Ideja za kreiranje ovakvog sustava za preporučivanje sadržaja bazirana je na sličnosti vektora među sadržajem sličnih videozapisa. Prethodno predstavljena ideja ispostavila se kao prilično dobra što se i vidi kroz funkcioniranje sustava.

Kao neuronski jezični model koristi se sentence-transformers/paraphrase-MiniLM-L6-v2 koji se temelji na arhitekturi transformera. Arhitektura transformera je prvi put predstavljena u radu *Attention Is All You Need* [35] i temelji se na principu pozornosti (eng. *attention*). Isto tako, donijela je veliki napredak u području obrade prirodnog jezika [36]. Ulaz u model su identifikatori riječi pomoću kojih se riječi mapiraju na vektorsku reprezentaciju koristeći potpuno povezani sloj neuronske mreže. Dodatno se na vektorsku reprezentaciju dodaje pozicijsko kodiranje koje modelu daje informaciju o poziciji dane riječi u tekstu. Reprezentacije svih riječi mapiraju se na tri vektora: vektor ključa, vektor upita i vektor vrijednosti. Mapiranje se događa pomoću naučenih potpuno povezanih slojeva kojih kod ovog modela ima šest. Mapiranje vektora ključeva, upita i vrijednosti provodi se sljedećim formulama:

1. Mapiranje na upit (Q):

$$Q_i = W_Q \cdot x_i$$

gdje je W_Q matrica težina koja transformira ulazni vektor x_i u upitni vektor Q_i

2. Mapiranje na ključ (K):

$$K_i = W_K \cdot x_i$$

gdje je W_K matrica težina koja transformira ulazni vektor x_i u ključni vektor K_i

3. Mapiranje na vrijednost (V):

$$V_i = W_V \cdot x_i$$

gdje je W_V matrica težina koja transformira ulazni vektor x_i u vrijednosni vektor V_i

Važno je napomenuti kako su W_Q , W_K i W_V matrice težina koje se nauče tijekom treniranja modela.

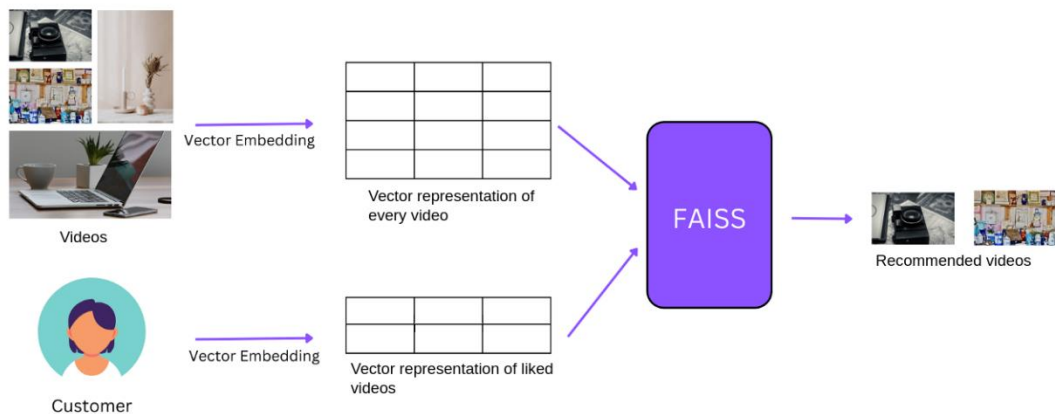
Princip pozornosti omogućava prilagodbu vektorske reprezentacije riječi ovisno o kontekstu u kojem se riječ našla oslanjajući se pritom na sličnost između upita dane riječi i ključeva drugih riječi. Nova vektorska reprezentacija nakon principa pozornosti se dobije težinskom sumom u kojoj su faktori određeni sličnošću upita dane riječi i ključeva drugih riječi.

Model sentence-transformers/paraphrase-MiniLM-L6-v2 prethodno je treniran na zadatku predviđanja sličnosti dviju rečenica. Rezultat takvog treniranja je upravo ono što je bilo potrebno za ostvarivanje ovog sustava, a to je da model za slične tekstove stvara vektorski slične reprezentacije. U radu *Pre-training of Deep Bidirectional Transformers for Language Understanding* [37] govori se o koristi transformera te o unaprednom treniranju na raznim zadacima. Isto tako, govori se i o mogućnostima prilagođavanja na nove specifične zadatke.

Na zahtjev za dohvaćanjem novih preporuka za nekog korisnika, iz Faiss baze podataka se dohvaćaju vektori koji reprezentiraju videozapise koje je korisnik prethodno pozitivno ocijenio ili pogledao u visokom postotku. Tim vektorima potom tražimo najbližije koristeći L2 normu, odnosno udaljenost, među svim vektorima u bazi. Dobivene preporuke je potrebno filtrirati i izbaciti one sa 100%-tnom sličnosti s obzirom na to da su to upravo oni videozapisi kojima tražimo slične.

Prilikom dodavanja novog videozapisa u bazu, potrebno je kreirati vektorsku reprezentaciju tog videozapisa te ga pohraniti u Faiss bazu podataka.

Sustav za preporučivanje video sadržaja je pokrenut kao manja web aplikacija koristeći radni okvir Flask te je preporuke moguće dohvatiti API pozivom. Dodavanja novog videozapisa se također vrši API pozivom uz slanje podataka o videozapisu u *body*-ju zahtjeva. Slika 36 ilustrira tijek (eng. flow) rada sustava.



Slika 36. Vlastito rješenje

6. Učinkovitost sustava

Tablica 1 prikazuje podatke o povezanosti pojedinih preporuka s preferiranim kategorijama, raznolikost videozapisa koji su preporučeni, relevantnost preporuka s obzirom na prethodne interakcije i inovativnost preporuka. Koriste se kriteriji i pristupi iz rada *Evaluating collaborative filtering recommender systems* [10]. Skala ocjenjivanja je od 1 do 5 za svaki kriterij pri čemu viši broj predstavlja bolju ocjenu.

Početna točka procjene učinkovitosti sustava je registracija korisnika te odabir kategorija. Za korisnika sam izabrao kategorije “**Sports**”, “**Science**” i “**Nature**”. Inicijalno sam napravio nekoliko interakcija s raznim videozapisima te nakon nekog vremena uzeo uzorak preporuke svakog pojedinog sustava i napravio evaluaciju. Zbog jednostavnosti i lakšeg pregleda, u obzir se uzima samo prvih 5 predloženih videozapisa.

Tablica 1. Evaluacija učinkovitosti preporuka

Sustav Preporuka	Naslov videozapisa	Povezanost s kategorijama	Releva ntnost	Raznoli kost	Inovativ nost	Ukupna ocjena
Recombee	Fractions! Mini Math Movies Scratch Garden	5 - Science	5	5	5	51
Recombee	The Freshwater Paradox	5 - Nature	4			
Recombee	How To Draw A Dragon	1 – Art & Crafts	2			
Recombee	The Greatest Comebacks - 2023	5 - Sports	4			
Recombee	Math Antics - Telling Time	5 – Science	5			
Amazon Personalize	How An Igloo Keeps You Warm	5 - Science	3	5	4	43

Amazon Personalize	Math Antics - Telling Time	5 - Science	5			
Amazon Personalize	Could I Dig a Hole Through the Earth?	5 - Science	4			
Amazon Personalize	How To Draw Meowth Pokemon	1 – Art & Crafts	1			
Amazon Personalize	Explorers in History - Exploring the Globe - Lesson Preview	2 - History	3			
Vlastito rješenje	Addition and Subtraction with Dinosaurs - Math for Kids - Math Operations	5 - Science	5	4	3	51
Vlastito rješenje	Deadliest Sea Creatures Underwater Killers Shark vs Giant Squid The Dr. Binocs Show	5 - Science	3			
Vlastito rješenje	Fractions! Mini Math Movies Scratch Garden	5 - Nature	5			
Vlastito rješenje	Top 100 NBA Plays of 2021	5 - Sport	3			
Vlastito rješenje	NBA's Top 100 Plays Of The Decade	5 - Sport	3			

Evaluacija sam proveo na primjeru jednog korisnika korištenjem aplikacije na standardno očekivani način.

Pri evaluaciji, koristio sam pet metrika: povezanost s kategorijama, relevantnost, raznolikost, inovativnost i ukupna ocjena.

- **Povezanost s kategorijama** mjeri koliko su preporučeni videozapisi povezani s interesima i kategorijama koje je korisnik prethodno iskazao i izabrao.
- **Relevantnost** mjeri koliko su preporuke relevantne za korisnika u smislu sadržaja i interesa.
- **Raznolikost** ocjenjuje koliko su preporuke raznolike.
- **Inovativnost** ocjenjuje koliko su preporuke nove i svježije za korisnika.
- **Ukupna ocjena** je zbroj svih prethodnih ocjena i predstavlja ukupnu učinkovitost preporuka sustava.

Sustav Recombee pokazuje prilično visoku inovativnost i raznolikost što proizlazi iz rezultata da dva videozapisa, odnosno 40%, dolaze iz područja matematike za koje je korisnik najviše zainteresiran. Isto tako, dva videozapisa, odnosno 40%, dolaze iz kategorija koje je korisnik označio preferabilnima prilikom registracije. Videozapis “How To Draw A Dragon” apsolutno je nepovezan s kategorijama i nije relevantan za ovog korisnika te su i ocjene u skladu s tim.

Sustav Amazon Personalize također pokazuje prilično visoku raznolikost, no nešto nižu inovativnost u odnosu na sustav Recombee. Tri videozapisa pristižu iz korisnikove omiljene kategorije, no greške nisu toliko relevantne jer nisu iz područja matematike. Dva videozapisa su iz kategorija koje uopće ne zanimaju korisnika pa je zbog nižih ocjena u tim slučajevima i ukupna ocjena nešto niža.

Sustav vlastite izrade postiže niže ocjene za mjere raznolikosti i inovativnost. Razlog tome je ograničenje u preporučivanju sadržaja koji nije usko vezan uz prethodne interakcije. U Tablica 1 je vidljivo kako je povezanost s kategorijama izrazito visoka, no relevantnost i nije u potpunosti. Neki videozapisi imaju nižu ocjenu iz mjere relevantnosti jer nisu povezani s matematikom.

Jednako dobre ocjene su ostvarene za sustave Recombee i vlastito rješenje. Važno je napomenuti kako sustav Recombee ipak ima dodatnu prednost jer za razliku od vlastitog rješenja sustav prilično dobro radi i u hladnom startu. Važno je za sustav da su preporuke dovoljno dobre i u samom početku korištenja kako bi se zadržalo korisnika i zainteresiralo za daljnje korištenje. Vlastito rješenje je postiglo visoku ocjenu jer su svi videozapisi došli iz kategorija koje je korisnik označio željenima dok je kompromis u tome što je inovativnost prilično loša. Isto tako, važno je za istaknuti kako su sustavi Recombee i

Amazon Personalize dali jednak videozapis u listi preporuka. Riječ je o videozapisu *Math Antics - Telling Time*. Također, sustavi Recombee i vlastito rješenje su dali jednaku preporuku, a to je videozapis *Fractions! Mini Math Movies | Scratch Garden*. Sustav Amazon Personalize ostvaruje nešto lošiju ukupnu ocjenu, no neovisno o tome, radi se o sustavu koji nudi visoke mogućnosti i performanse. Za pravilno funkcioniranje takvog sustava potrebno je jako puno znanja i utrošenog vremena kako bi ga se ispravno postavilo i pustilo u rad.

Prema ovoj usporedbi, sustav Recombee kompletno se pokazao kao potpuno rješenje koje je vrlo pristupačno cijenom i potencijalno najbolji izbor za aplikacije ovog tipa. Isto tako, može se zaključiti da se uz ulaganje određene količine vremena može kreirati vlastito rješenje koje također funkcionira zadovoljavajuće, a potencijalno i jednako dobro kao i drugi sustavi.

Zaključak

U suvremenom digitalnom dobu, tehnologija je uključena u svaki aspekt čovjekova života. Problem predstavlja nekvalitetan i štetan sadržaj koji se plasira, a gotovo je nezaobilazan. U okviru ovog diplomskog rada, razvijena je sveobuhvatna aplikacija. Aplikacija ima cilj pružiti roditeljima sigurno okruženje za njihovu djecu. Velik problem sveobuhvatnih sustava i aplikacija za gledanje video sadržaja je u tome što taj materijal možda nije prilagođen za djecu ili jednostavno nije u skladu s interesima kako roditelja za njihovu djecu tako i same djece. Ova aplikacija nudi izbor kategorija te je u svega nekoliko minuta moguće gledati potpuno personaliziran sadržaj u kojem je mogućnost pojave lošeg i štetnog sadržaja sveden na minimum.

Uz razvoj aplikacije, napravljen je detaljan pregled područja za preporuku video sadržaja. Razni sustavi su napravljeni, integrirani i testirani te su istaknute i prezentirane prednosti i nedostaci svakog. Uz integraciju dva gotova sustava (Recombee i Amazon Personalize), koje je bilo samo potrebno prilagoditi potrebi ove aplikacije, stvoreno je i vlastito rješenje. Vlastita inačica sustava stvorena je s ciljem da zaobiđe sve nedostatke koje gotova rješenja nose u sebi. Sustav koristi model za procesiranje prirodnog jezika koji je prethodno treniran na općenitim stvarima i cilja na široku primjenu. S obzirom na to da je aplikacija široke namjene po pitanju kategorija i video sadržaja koje nudi, takav model pokazao se savršenim izborom.

Razvijeni sustav predstavlja savršeni temelj za daljnji razvoj i proširenja. Jedna od mogućnosti daljnjeg rada je iskorištavanje posjedovanja mogućnosti rada mobilnih uređaja u ograničenom načinu rada (eng. *guided access mode*). Ako bi se ova aplikacija napravila za mobilne uređaje i koristila u ograničenom načinu rada roditelji bi mogli biti posve sigurni kako njihova djeca koriste tehnologiju na ispravan način. S obzirom na sve štetniji video materijal koji se prezentira djeci, potreba za aplikacijama ovog tipa će bivati sve veća.

Literatura

- [1] Recombee, *Artificial Intelligence Powered Recommender as a Service*, Poveznica: <https://www.recombee.com/>; pristupljeno 4. lipnja 2024.
- [2] Amazon Personalize, Poveznica: <https://aws.amazon.com/personalize/>; pristupljeno 4. lipnja 2024.
- [3] Ricci, F., Rokach, L., & Shapira, B., *Introduction to Recommender Systems Handbook*. Springer, (2018), Poveznica: https://link.springer.com/chapter/10.1007/978-0-387-85820-3_1, pristupljeno 25. lipnja 2024.
- [4] Adomavicius, G., & Tuzhilin, A., *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions*, (2005), Stranice 734-749, Poveznica: <https://ieeexplore.ieee.org/document/1423975>, pristupljeno 25. lipnja 2024.
- [5] Koren, Y., Bell, R., & Volinsky, C., *Matrix factorization techniques for recommender systems*, (2009), Stranice 30-37, Poveznica: <https://ieeexplore.ieee.org/document/5197422>, pristupljeno 25. lipnja 2024.
- [6] Lops, P., Gemmis, M. D., & Semeraro, G., *Content-based Recommender Systems: State of the Art and Trends*, (2011), Stranice 73-105, Poveznica: https://link.springer.com/chapter/10.1007/978-0-387-85820-3_3, pristupljeno 25. lipnja 2024.
- [7] Burke, R., *Hybrid recommender systems: Survey and experiments*, (2002), Stranice 331-370, Poveznica: <https://link.springer.com/article/10.1023/A:1021240730564>, pristupljeno 25. lipnja 2024.
- [8] Gena, C., & Weibelzahl, S., *Usability Engineering for the Adaptive Web*, (2007), Stranice 720-762, Poveznica: https://link.springer.com/chapter/10.1007/978-3-540-72079-9_24, pristupljeno 25. lipnja 2024.
- [9] Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S., *Collaborative Filtering Recommender Systems*, (2007), Stranice 291-324, Poveznica: https://link.springer.com/chapter/10.1007/978-3-540-72079-9_9, pristupljeno 25. lipnja 2024.
- [10] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T., *Evaluating collaborative filtering recommender systems*, (2004), Stranice 5-53, Poveznica: <https://dl.acm.org/doi/10.1145/963770.963772>, pristupljeno 25. lipnja 2024.
- [11] YouTube Kids, Poveznica: <https://www.youtube.com/kids/>; pristupljeno 4. lipnja 2024.
- [12] PBS Kids Video, Poveznica: <https://pbskids.org/videos/>; pristupljeno 4. lipnja 2024.
- [13] Khan Academy Kids, Poveznica: <https://learn.khanacademy.org/khan-academy-kids/>; pristupljeno 4. lipnja 2024.
- [14] Next.js, *Next.js - The React Framework for the Web*, Poveznica: <https://nextjs.org/>; pristupljeno 4. lipnja 2024.

- [15] Ksantiani13, *Different forms of Pre-rendering in Next.js*, Poveznica: <https://www.geeksforgeeks.org/different-forms-of-pre-rendering-in-nextjs/>; pristupljeno 4. lipnja 2024.
- [16] TailwindCSS, *TailwindCSS – Rapidly build modern websites without ever leaving you HTML*, Poveznica: <https://tailwindcss.com/>; pristupljeno 4. lipnja 2024.
- [17] Media Queries, *Responsive Web Design - Media Queries*, Poveznica: https://www.w3schools.com/css/css_rwd_mediaqueries.asp/; pristupljeno 4. lipnja 2024.
- [18] DaisyUI, *DaisyUI – The most popular component library for Tailwind CSS*, Poveznica: <https://daisyui.com/>; pristupljeno 4. lipnja 2024.
- [19] Yarden Katz, *Introduce Tailwind CSS Support in Anima*, Poveznica: <https://www.animaapp.com/blog/product-updates/introducing-tailwind-css-support-in-anima/>; pristupljeno 4. lipnja 2024.
- [20] Node.js, *Node.js - Run JavaScript Everywhere*, Poveznica: <https://nodejs.org/en/>; pristupljeno 4. lipnja 2024.
- [21] Kinsta, *What Is Node.js and Why You Should Use It*, Poveznica: <https://kinsta.com/knowledgebase/what-is-node-js/>; pristupljeno 4. lipnja 2024.
- [22] Express.js, *Express.js – Node.js web application framework*, Poveznica: <https://expressjs.com/>; pristupljeno 4. lipnja 2024.
- [23] James Hartman, *Node.js Express Framework Tutorial*, Poveznica: <https://www.guru99.com/node-js-express.html>, pristupljeno 4. lipnja 2024.
- [24] MongoDB, *MongoDB – The developer Data Platform*, Poveznica: <https://www.mongodb.com/>; pristupljeno 4. lipnja 2024.
- [25] Priya Predamkar, *What is MongoDB?*, Poveznica: <https://www.educba.com/what-is-mongodb/>; pristupljeno 4. lipnja 2024.
- [26] Python, Poveznica: <https://www.python.org/>; pristupljeno 4. lipnja 2024.
- [27] Vikram Singh, *Top 10 Features of Python*, Poveznica: <https://www.shiksha.com/online-courses/articles/top-10-features-of-python/>; pristupljeno 4. lipnja 2024.
- [28] Flask, *Flask – The Python micro framework for building web applications*, Poveznica: <https://flask.palletsprojects.com/en/3.0.x/>; pristupljeno 4. lipnja 2024.
- [29] Jeel Patel, *Flask vs Django: A Detailed Comparison of Python Web Frameworks*, Poveznica: <https://www.monocubed.com/blog/flask-vs-django/>; pristupljeno 4. lipnja 2024.
- [30] Sentence Transformers, Poveznica: <https://sbert.net/>; pristupljeno 4. lipnja 2024.
- [31] Faiss, Poveznica: <https://faiss.ai/index.html/>; pristupljeno 4. lipnja 2024.
- [32] Amazon Web Services, Poveznica: <https://aws.amazon.com/>; pristupljeno 4. lipnja 2024.
- [33] Amazon Web Services S3, Poveznica: <https://aws.amazon.com/s3/>; pristupljeno 4. lipnja 2024.
- [34] Amazon Web Services Redshift, Poveznica: <https://aws.amazon.com/redshift/>; pristupljeno 4. lipnja 2024.

- [35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, *Attention Is All You Need*, (2017, lipanj) Poveznica: <https://arxiv.org/abs/1706.03762>, pristupljeno 25. lipnja 2024.
- [36] Prakash M Nadkarni, Lucila Ohno-Machado, Wendy W Chapman, *Natural language processing: an introduction*, Journal of the American Medical Informatics Association, Volume 18, Issue 5, (2011, rujan), Stranice 544–551, Poveznica: <https://doi.org/10.1136/amiajnl-2011-000464>, pristupljeno 25. lipnja 2024.
- [37] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, (2018, listopad) Poveznica: <https://arxiv.org/abs/1810.04805>, pristupljeno 25. lipnja 2024.

Sažetak

U okviru ovog diplomskog rada uspoređivani su različiti sustavi za predlaganje video sadržaja. Uz integraciju i testiranje gotovih rješenja, Recombee i Amazon Personalize, implementirana je i vlastita inačica sustava za preporučivanje video sadržaja. Također, kreirana je klijentska aplikacija koja nudi mogućnost testiranja i korištenja implementiranih sustava. Za manipulaciju baze podataka, odnosno skupa videozapisa koje sustavi preporuka koriste, kreirana je još jedna klijentska aplikacija namijenjena administratorima. U radu je detaljno opisana razlika pojedinih sustava, prednosti i nedostaci te su tekstualno i slikovno opisane sve funkcionalnosti i izgled aplikacija.

Summary

In this thesis, many different recommendation systems are tested and compared. In addition to the integration and testing of ready-made solutions, Recombee and Amazon Personalize, my own version of recommendation system is also implemented. Also, a client application with possibility to test and use integrated systems is created. Another client application which is created is used for data manipulation by admins. The idea is to use that application to manage set of videos, ie. add, update or delete videos. The paper describes in detail differences between individual systems, their advantages and disadvantages. Also, all the functionalities and appereance of the application are described in text and shown in pictures.