

Web aplikacija za organizaciju baze znanja za chatbot

Dragun, Ante

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:749222>

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

Download date / Datum preuzimanja: **2025-02-02**

Repository / Repozitorij:

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



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

Sveučilišni prijediplomski studij Računarstvo

Web aplikacija za organizaciju baze znanja za chatbot
Završni rad

Ante Dragun

Osijek, 2024.

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

Ime i prezime pristupnika:	Ante Dragun
Studij, smjer:	Sveučilišni prijediplomski studij Računarstvo
Mat. br. pristupnika, god.	R4636, 27.07.2021.
JMBAG:	0165090813
Mentor:	doc. dr. sc. Hrvoje Leventić
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Web aplikacija za organizaciju baze znanja za chatbot
Znanstvena grana završnog rada:	Programsko inženjerstvo (zn. polje računarstvo)
Zadatak završnog rada:	Napraviti web aplikaciju u Django koja omogućava organizaciju baze znanja za chatbot. Aplikacija će omogućiti upravljanje dokumentima od strane različitih korisnika, a chatbot će za generiranje odgovora koristiti te dokumente. (Predbilježbe: Ante Dragun,
Datum prijedloga ocjene završnog rada od strane mentora:	18.09.2024.
Prijedlog ocjene završnog rada od strane mentora:	Izvrstan (5)
Datum potvrde ocjene završnog rada od strane Odbora:	25.09.2024.
Ocjena završnog rada nakon obrane:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio sveučilišni prijediplomski studij:	27.09.2024.



FERIT

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O IZVORNOSTI RADA

Osijek, 27.09.2024.

Ime i prezime Pristupnika:

Ante Dragun

Studij:

Sveučilišni prijediplomski studij Računarstvo

Mat. br. Pristupnika, godina upisa:

R4636, 27.07.2021.

Turnitin podudaranje [%]:

3

Ovom izjavom izjavljujem da je rad pod nazivom: **Web aplikacija za organizaciju baze znanja za chatbot**

izrađen pod vodstvom mentora doc. dr. sc. Hrvoje Leventić

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

SADRŽAJ

1. UVOD	1
1.1. Zadatak završnog rada	1
2. ORGANIZACIJA BAZE ZNANJA ZA CHATBOT	2
2.1. Chatbot	2
2.2. Obrada prirodnog jezika	3
2.2.1. Jezični model	3
2.3. Umjetna inteligencija	4
2.3.1. Strojno učenje	5
2.3.2. Duboko učenje	6
2.4. Veliki jezični modeli	7
2.5. Baza znanja	8
2.5.1. Sigurnost baze znanja	11
2.6. Ekspertni sustavi	12
2.7. Retrieval-Augmented Generation	13
2.8. Pregled dostupnih softvera za organizaciju baze znanja za chatbot	14
3. IZRADA APLIKACIJE ZA ORGANIZACIJU BAZE ZNANJA	16
3.1. Funkcionalnost aplikacije	16
3.1.1. Baza podataka	16
3.1.2. Autentikacija korisnika	18
3.1.3. Kontroleri	18
3.2. Opis rada aplikacije	22
3.2.1. Početna stranica i korisnički pristup	22
3.2.2. Prikaz i uređivanje dokumenata	24
3.2.3. Upravljanje dokumentima i mapama	26
3.2.4. Opis administratorskih ovlasti	28
4. ZAKLJUČAK	32
LITERATURA	34
SAŽETAK	36
ABSTRACT	37

1. UVOD

U današnje vrijeme, *chatbotovi* su postali često korišteni alati tehnologije zbog pristupačnosti i širokog spektra mogućnosti koje nude. U mnogim industrijama doživljavaju nagli rast popularnosti zahvaljujući sposobnosti komuniciranja s ljudima te pružanju odgovarajućih odgovora u različitim oblicima. Međutim, kvaliteta *chatbotova* u kontekstu brzine, točnosti i prilagodljivosti značajno ovise o kvaliteti organizacije baze znanja – važnog čimbenika koji omogućuje razumijevanje i odgovaranje na korisničke upite.

U ovom završnom radu obrađena je tema *chatbotova*, polazeći od njihove osnovne strukture do uvođenja baze znanja u cijelu strukturu, kao i organiziranje baze te opis popularnih metoda organiziranja baze znanja. Praćen je povijesni razvoj programa za komunikaciju s ljudima kao i detaljan strukturalni pregled arhitekture modernih programa iste namjene. Između ostaloga, detaljno su izloženi svi koncepti i gradivni elementi potrebni za razumijevanje rada *chatbota*. Na kraju prvoga dijela, prikazani su i opisani dostupni popularni softveri za organizaciju baze znanja za *chatbot*.

Drugi je dio baziran na realizaciji web aplikacije u okruženju *Ruby on Rails*, koji za cilj ima organizirati bazu znanja za *chatbot*. U web aplikaciji omogućeno je upravljanje dokumenatima od strane različitih korisnika, njihovo skladištenje te osiguranje integriteta. Prilikom generiranja odgovora, *chatbot* koristi spremljene dokumente iz baze znanja. Na početku su prikazane osnovne funkcionalnosti web aplikacije, a nakon toga i opis rada. Na posljertku je sumiran sadržaj ovoga rada te je iznesen zaključak.

1.1. Zadatak završnog rada

Web aplikacija omogućava organizaciju baze znanja za *chatbot* u okruženju *Ruby on Rails*. Aplikacija omogućava upravljanje dokumentima od strane različitih korisnika, a *chatbot* za generiranje odgovora koristi te dokumente.

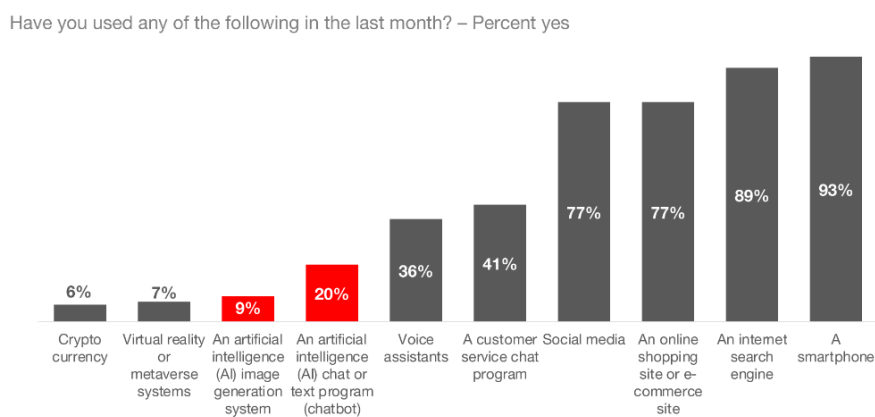
2. ORGANIZACIJA BAZE ZNANJA ZA CHATBOT

Prije nego što su se strojevi za generiranje odgovora počeli pojavljivati, prvo se o njima počelo razmišljati u teorijskim oblicima. Prema [1], jedan od prvih koji je razmišljao u tome smjeru bio je Alan Turing, koji je svojim pitanjem „Mogu li strojevi misliti?“ konceptualizirao ideju razdvajanja računala i ljudi tada inovativnom metodom – Turingovim testom. Test bi se sastojao od 3 entiteta – suca, stroja i čovjeka. Sudac bi postavljao pitanja čovjeku i stroju koji bi se pretvarao kao čovjek te ako se s obzirom na odgovore, čovjek i stroj ne mogu razlikovati, zaključak bi bio da stroj može misliti.

Tada se prvi puta govorilo o inteligenciji računala, što je pokrenulo daljnja istraživanja na tu temu, a to je dovelo do razvoja umjetne inteligencije kakvu i danas poznajemo. Prvi stroj koji je mogao generirati odgovore bio je program ELIZA koji je Joseph Weizenbaum stvorio 1966. godine. Program ELIZA imao je mogućnost identifikacije ključnih riječi u tekstu te na temelju istih generirati odgovor prema određenom skupu prethodno određenih pravila [2]. 1995. godine Richard Wallace stvorio je mnogo kompleksniji sustav od svojih prethodnika - ALICE. To je prvi program za generiranje odgovora koji je uveo bazu u koju bi spremao dokumente u obliku pitanje – odgovor. Svojevremeno, ALICE je prozvan najinteligentnijim programom za generiranje odgovora [3].

2.1. Chatbot

Prema [4], *chatbot* je računalni sustav koji omogućava interakciju tog istog sustava s čovjekom koristeći pisani jezik. Komunikacija se najčešće odvija putem sučelja u kojem korisnik unosi svoju poruku *chatbotu*, a *chatbot* mu vraća odgovor sukladno dizajnu sustava. Nekada zbog ograničenih računalnih resursa nisu bili toliko popularni, ali s vremenom računalni zahtjevi postaju veći, a samim time i računala snažnija. Optimizacijom postojećih sustava te uvođenjem novih elemenata u postojeće strukture, *chatbotovi* dobivaju široku paletu znanja koja može biti iskorištena u različitim područjima ljudskoga djelovanja. Pristupačnost i količina znanja kojom *chatbotovi* raspolažu svakako su značajno utjecali na njihovu popularnost u proteklih nekoliko godina. Iz slike 2.1. uočljivo je da je čak 20 % građana Sjedinjenih Američkih Država u ožujku 2023. godine koristilo neki oblik *chatbota*.



Slika 2.1. Postotak korištenih tehnologija u ožujku 2023. godine u SAD-u (Ipsos) preuzeto s [5]

2.2. Obrada prirodnog jezika

Prethodnici modernih *chatbotova* ALICE i ELIZA imali su značajno jednostavnije oblike obrade prirodnoga jezika. Program ELIZA generirao je odgovore u pisanom jeziku koristeći prethodno definirane uzorke koristeći riječi iz rečenice korisnika koji komunicira s programom. Dakle, program nije razumio kontekstualno značenje korisnikove rečenice. Program ALICE je, iako u nekim pogledima napredniji, i dalje koristio skup određenih pravila za generiranje odgovora i također nije mogao razumjeti značenje korisnikovih riječi te su po tom pitanju oba sustava bila poprilično ograničena.

Po [6], nastaju dva problema oslanjajući se isključivo na predodređena pravila generiranja odgovora. Program bi morao razumjeti semantičko značenje teksta korisnika te predodređena pravila jako loše obrađuju negramatični govorni jezik pa bi ih zbog toga trebalo izbjegavati. Nastala je potreba za stvaranjem takvog sustava koji bi mogao razumjeti korisnikovu poruku te ju analizirati i poslati odgovor natrag u korisniku razumljivom jeziku, a taj proces pretvorbe nestrukturiranog teksta u strukturnu reprezentaciju koja sadrži razumijevanje prirodnog jezika (engl. *Natural Language Understanding*) nazivamo obrada prirodnog jezika (engl. *Natural Language Processing, NLP*). Moderni *chatbotovi* koriste obradu prirodnog jezika, a *chatbot* koji ju ne koristi, na tržištu ne može parirati većim *chatbotovima* koji ju koriste zbog značajno koherentnijih i točnijih odgovora.

2.2.1. Jezični model

Jezični je model vrsta modela strojne obrade jezika koji koristi metode umjetne inteligencije te statističke tehnike za razumijevanju, obradu i generiranje teksta na temelju ulaznih podataka, a

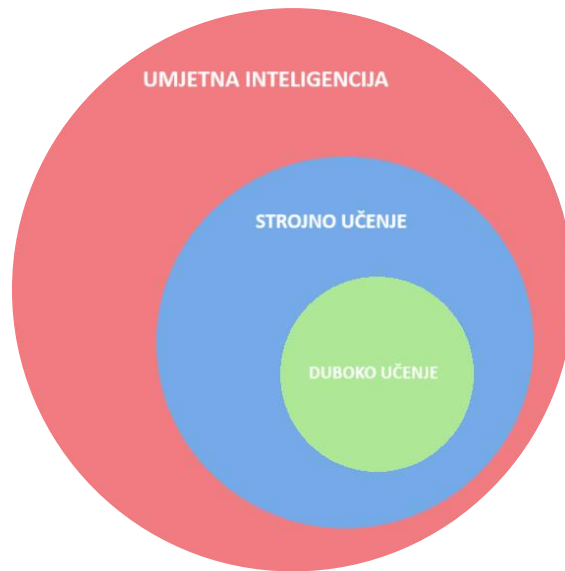
time predstavljaju temeljni element sustava za obradu prirodnog jezika. Korištenje jezičnih modela postala je osnova za stvaranje svih modernih *chatbotova*, jer se njihovim korištenjem mogu pravilno analizirati korisnički upiti i na temelju konteksta generirati odgovori, bez da se koriste definirana pravila kao što su to činili raniji sustavi. Kako bi model mogao semantički razumjeti podatke, za vrijeme treninga jezičnog modela koristi se i alat tokenizator, koji cijeli ulazni skup podataka pri ulazu u model dijeli na tokene, odnosno semantički korisne jedinice koje, ovisno o tokenizatoru, mogu biti cijele riječi ili njihovi dijelovi, a ponekada i zasebni znakovi, a taj proces pretvaranja ulaznog skupa podataka u tokene naziva se tokenizacija.

Nakon tokenizacije, zbog boljeg razumijevanja semantike korisničkog upita, koriste se vektorske reprezentacije riječi ili tokena (engl. *embedding*). Vektorske reprezentacije riječi omogućuju različitim modelima da razumiju kontekst korisničkog upita tako da se svakom tokenu iz upita dodijeli jedinstvena višedimenzionalna numerička vrijednost s kojom model može raditi. Tako modeli puno lakše mogu razumjeti i odnose među riječima, pa lakše prepoznaje sinonime i druge sličnosti između riječi, a obično se koriste i metode strojnoga učenja za bolju analizu koherentnosti između podataka.

Na primjeru gradova i država lako je razumljivo kako model prepoznaje odnose između različitih podataka koristeći vektorsku reprezentaciju. Vektor države Njemačke i vektor glavnog grada Njemačke Berlina slično su povezani kao i vektor države Italije i vektor glavnog joj grada Rima. Tako će u višedimenzionalnom vektorskom prostoru vektori za Njemačku i Berlin po pitanju udaljenosti i smjera biti slični vektorima Italije i Rima. Na takav način, model može prepoznati da je odnos između vektora Njemačke i Berlina sličan odnosu između vektora Italije i Rima, a isto tako model može prepoznati slične odnose i time shvatiti kontekst drugih podataka u vektorskom prostoru.

2.3. Umjetna inteligencija

Umjetna inteligencija (engl. *Artificial Intelligence*) grana je računarstva koja se odnosi na sustave koji simuliraju ljudsku inteligenciju te s obzirom na to da razmišljaju kao i ljudi, u stanju su razumjeti i izvršiti skup zadataka koji mogu i ljudi kao što je donošenje odluka i rješavanje problema vezanih uz specifično područje. Iz slike 2.2. uočljivo je kako se unutar umjetne inteligencije nalazi grana strojnoga učenja, a unutar strojnoga učenja duboko učenje koje je usko vezano uz proces obrade prirodnog jezika.



Slika 2.2. Prikaz unutarnje strukture umjetne inteligencije

2.3.1. Strojno učenje

Prema [7], strojno je učenje (engl. *machine learning*) takav oblik programiranja računala koji ima za cilj optimizirati određeni kriterij uspješnosti koristeći skup podataka ili prethodno iskustvo. Kod strojnog se učenja definira model koji ima određene varijabilne parametre, a proces učenja odnosi se na optimizaciju tih parametara s ciljem dobivanja novog znanja o nekom problemu ili predviđanja izlazne varijable na do tada neviđenim podacima. Dakle, cilj je strojnog učenja pronaći dovoljno dobar model koji aproksimira podatke i može se u budućnosti koristiti. Strojno se učenje dijeli na nadzirano učenje, nenadzirano učenje i podržano učenje.

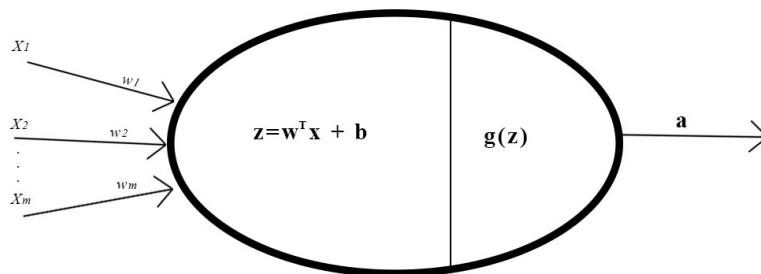
- **Nadzirano učenje (engl. *Supervised learning*)** – kod ove vrste strojnog učenja poznati su i ulazni i izlazni podaci te se s pomoću njih pokušava doći do što boljšeg modela, odnosno do modela koji najbolje aproksimira ovisnost između ulaznih i izlaznih podataka. Nakon određivanja funkcionalne ovisnosti između poznatih ulaznih i izlaznih podataka, s pomoću dobivenog modela moguće je predvidjeti izlazni skup podataka za određeni ulazni skup. U kontekstu *chatbotova*, nadzirano se učenje najčešće koristi za prepoznavanje različitih obrazaca u upitu korisnika te prepozna ključne riječi i na temelju toga olakšava generiranje odgovora.
- **Nenadzirano učenje (engl. *Unsupervised learning*)** – za razliku od nadziranog učenja, nenadzirano učenje nema znanja o izlaznom skupu podataka te ima za cilj odrediti konzistentnost između ulaznih podataka. Najčešće se pod pojmom nenadziranoga učenja konkretno misli na grupiranje podataka s obzirom na njihove zajedničke značajke ili

određivanja anomalija u podacima. Ova se vrsta učenja kod *chatbotova* koristi za grupiranje sličnih i uobičajenih upita korisnika te pomaže u generiranju sličnih odgovora s obzirom na slična pitanja.

- **Podržano učenje (engl. *reinforcement learning*)** – tip strojnog učenja u kojemu postoji posrednički element – agent, koji samostalno pokušava riješiti zadani problem koristeći povratnu informaciju od strane korisnika. Ta povratna informacija daje agentu znanje o tome je li njegovo ponuđeno rješenje ispravno te se tako optimiziraju koraci dolaska do rješenja zadanog problema. Podržano učenje između ostaloga svoju primjenu pronalazi u optimizaciji *chatbotova* te podizanje zadovoljstva korisnika tako da optimizira generirane odgovore s obzirom na tijek komunikacije s korisnikom.

2.3.2. Duboko učenje

Duboko je učenje (engl. *deep learning*) grana strojnog učenja koja koristi iznimno velike umjetne neuronske mreže, a za cilj imaju obradu podataka i učenje iz velikih količina informacija. Umjetna neuronska mreža složen je sustav umjetnih neurona, koji međusobno povezani čine oblik mreže. Umjetni je neuron funkcija koja za m ulaza i isti broj težina vraća jedan izlaz iz neurona kojeg zovemo aktivacija. Na slici 2.3. može se vidjeti uobičajena struktura umjetnog neurona. Skup spomenutih težina i pomak b parametri su umjetnoga neurona. Svaki ulaz množi se sa svojom težinom i na to se dodaje pomak b , a kada se taj izraz provuče kroz aktivacijsku funkciju, dobijemo konačan izlaz iz neurona.



Slika 2.3. Prikaz umjetnog neurona

Kako je navedeno, više umjetnih neurona moguće je spojiti u veću strukturu – umjetnu neuronsku mrežu. Umjetne neuronske mreže obično osim ulaznog i izlaznog sloja imaju i skriveni sloj umjetnih neurona. Da bi se umjetna neuronska mreža smatrala dubokom mrežom,

ona mora imati više od jednog skrivenog sloja u svojoj strukturi. Skriveno slojeve u umjetnoj neuronskoj mreži možemo dodavati, a broj umjetnih neurona u svakom sloju možemo mijenjati ovisno o modelu.

2.4. Veliki jezični modeli

Veliki jezični modeli (engl. *Large Language Models*) klasa su temeljnih jezičnih modela čiji su parametri procjenjeni na enormno velikim količinama podataka što ih čini sposobnim razumjeti i generirati odgovore u prirodnom jeziku kao i obavljanje različitih vrsta zadataka. Mnoge velike kompanije, uključujući IBM, provele su godine razvijajući velike jezične modele na različitim razinama kao i druge oblike strojnoga učenja, kako bi postali moćan alat koji se danas koristi u gotovo svim oblicima arhitekture modernih *chatbotova*. Kako su različite velike kompanije razvijale svoje velike jezične modele, tako je nastalo nekoliko zapaženih modela koji će biti uspoređeni s obzirom na njihove opsege mogućnosti i brojeve parametara modela.

- **GPT-4** – *Generative Pre-trained Transformer 4* prema [8] opisan je kao veliki multimodalni model, tj. model koji osim teksta, može za ulaz primiti i slike. Broj parametara GPT-4 jezičnog modela doseže nekoliko stotina milijardi parametara. Broj parametara nije javno dostupan, ali s obzirom na to da je GPT-3 imao 175 milijardi parametara, prognoza o broju parametara GPT-4 gotovo je sigurno ispravna. GPT-4 dominira u rasponu zadataka koje može obaviti u odnosu na druge jezične modele.
- **BERT** - *Bidirectional Encoder Representations from Transformers* jezični je model razvijen od Googlea koji, kao i GPT-4, koristi transformersku arhitekturu. BERT ima dvije verzije – *base* i *large*. *Base* verzija je model koji ima 110 milijuna parametara, dok *large* model ima 340 milijuna parametara. BERT model je specifičan po načinu na koji analizira tekst, jer koristi bidirekcijski pristup pri analizi semantičkog značenja riječi, odnosno model uzima u obzir kontekst riječi s obje strane, za razliku od tradicionalnih modela koji koriste sekvencijalni pristup analizi rečenice. Takav pristup BERT modelu omogućuje bolju semantičku analizu te zbog toga BERT model svoju primjenu pronalazi u prijevodu tekstova, pretraživanju i odgovaranju na pitanja [9].
- **LLaMA** – *Large Language Model Meta AI* model je proizveden od strane tvrtke Meta, a model dolazi u nekoliko različitih verzija među kojima najjači model LLaMa 3 ima 70 milijardi parametara. Kao i GPT-4, LLaMA model također je osmišljen s ciljem rješavanja

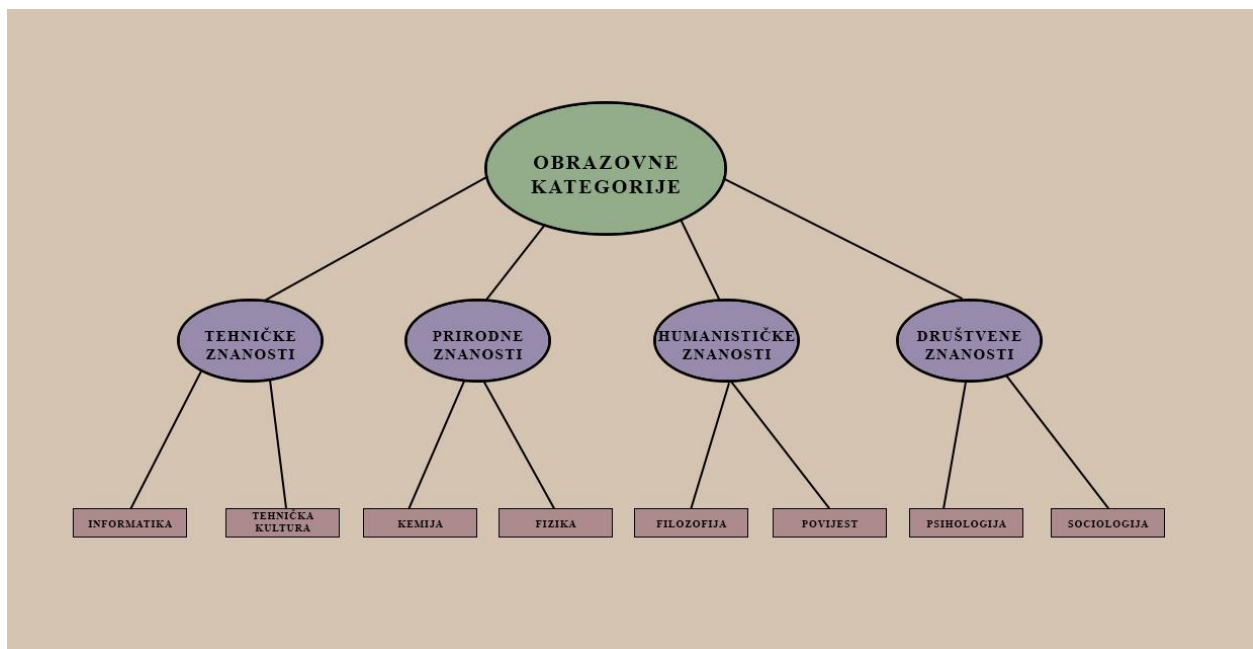
velikog opsega zadataka, a vrlo zapažene rezultate pokazuje u zadacima generiranja i prevođenja teksta. Razvojem osnovnog LLaMA modela, Prema [10], Meta je napredovala do modela LLaMA 3.1. koji ima 405 milijardi parametara te podržava komunikaciju na više govornih jezika, a nova je verzija posebno napredovala u generiranju programskoga kôda i rješavanju matematičkih zadataka.

2.5. Baza znanja

Kako je navedeno, prvi oblici *chatbotova* nisu imali nikakav oblik spremanja informacija, a program ALICE bio je pionir u korištenju baze u koju su bili spremljeni dokumenti koji su se koristili pri generiranju odgovora. U kontekstu *chatbotova*, baza znanja (engl. *knowledge base*) skup je informacija koje *chatbot* koristi kako bi došao do resursa koji mu trebaju za generiranje zadovoljavajućih odgovora. Na primjeru klijenta koji šalje pitanje *chatbotu*, on koristeći obradu prirodnog jezika obrađuje pitanje klijenta te ga uspoređuje s dokumentima iz baze znanja te pri tome koristi relevantne dokumente za kreiranje odgovora. Taj skup informacija koji se spremaju u bazu znanja nije isključivo rezerviran za tekstualne oblike podataka, između ostaloga moguće je dodati i druge oblike kao što su fotografije i videozapisi, ovisno o potrebama aplikacije. Kao i svaka baza podataka, tako i baza znanja stalno dobiva nove dokumente koje treba skladištiti i time proširiti bazu znanja. Po [11], shodno tome, za bazu znanja mora postojati sustav za dodavanje novih entiteta u bazu te dohvaćanja istih iz baze znanja. Ovisno o tome na kakva pitanja se želi da *chatbot* može odgovoriti, potrebno je kreirati bazu znanja s takvim dokumentima koji bi trebali dati odgovore na što više mogućih pitanja klijenata. Dodavanje dokumenata u bazu znanja obično se izvodi preko sučelja, koje će dodati dokument ovisno o načinu strukture baze znanja. Dobro strukturirana baza znanja omogućuje efikasno upravljanje dokumentima i njihovo korištenje čime se postiže lakše i brže pretraživanje potrebnih informacija iz baze znanja. Baza znanja najčešće je strukturirana kao kombinacija različitih organizacijskih metoda, među kojima su najpoznatije taksonomija i ontologija.

- **Taksonomija (engl. *taxonomy*)** – znanstvena je disciplina koja se bavi kategorizacijom taksonomskih jedinica s obzirom na njihove međusobne sličnosti i razlike. Kada govorimo o taksonomiji u kontekstu konstrukcije baze znanja, najčešće govorimo o tipičnoj hijerarhijskoj strukturi stabla gdje imamo 3 vrste elemenata: korijen, čvor i list. Korijen, poznat i kao glavni čvor, predstavlja prvi element u hijerarhijskoj strukturi. Korijen nema

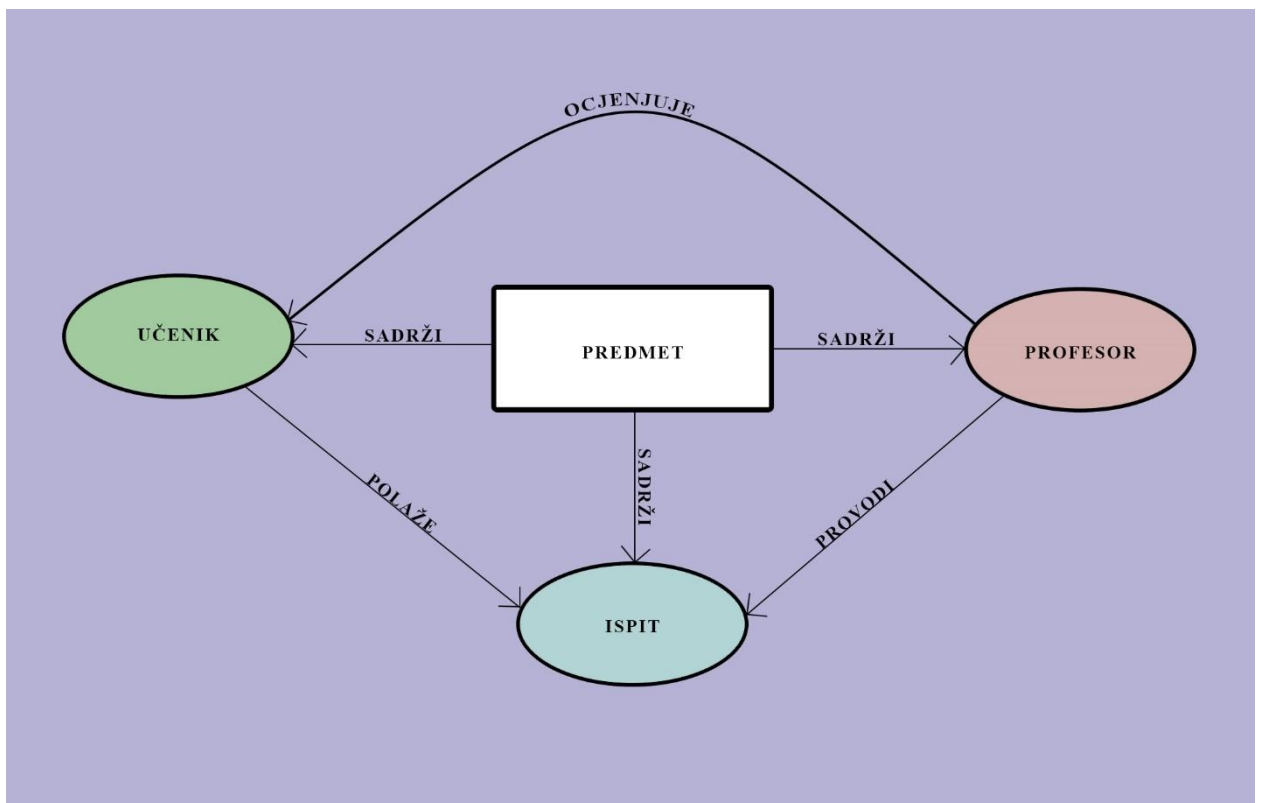
roditeljski čvor jer je po hijerarhiji na najvišoj razini. Čvor stabla odnosi se na element koji ima roditelja (čvor kojem pripada) i djecu (čvorove koji njemu pripadaju). List je element stabla koji nema djecu te se obično nalaze na najnižoj razini stabla. Svaka se struktura stabla smatra hijerarhijskom strukturom, a povećanjem dubine stabla, ono postaje kompleksnije i dobiva sve konkretnije potkategorije. Suprotno, što je manja razina stabla, kategorizacija će biti općenitija. Na slici 2.4. prikazana je taksonomija školskih predmeta. Koriijenski je čvor onaj koji obuhvaća sve discipline kojima školski predmeti pripadaju. U idućoj se razini nalaze djeca koriijenskog čvora, a svako dijete predstavlja jednu granu znanosti kojoj skupina predmeta može pripadati. U trećoj se razini nalaze listovi, to jest konkretni predmeti koji pripadaju određenom roditelju, odnosno grani znanosti. Ovakva struktura omogućava razlaganje logičkih cjelina na manje, pri tom povećavajući jasnoću podjele. Zbog lakšeg razumijevanja taksonomije, korišteno je stablo dubine 2. Povećanjem dubine stabla, povećava se njegova složenost, ali će dublje stablo preciznije kategorizirati i dati detaljniji pregled sadržaja.



Slika 2.4. Taksonomska podjela nastavnih predmeta u strukturi stabla

- **Ontologija** (engl. *ontology*) – prema [12], u aspektu informacijskih tehnologija ontologijom se smatra opisivanje koncepata i relacija koji postoje za nekog agenta ili grupe agenata. Ontologiju se smatra otvorenom, odnosno smatra ju se takvim opisivanjem koncepata, koji se može modificirati ovisno o potrebama. Za razliku od taksonomije koja koristi striktno hijerarhijske odnose (roditelj-dijete), ontologija koristi složenije vrste veza

za opisivanje odnosa između dva entiteta. Zbog ove mogućnosti, ontologija se smatra sadržajnijim opisivanjem koncepata od taksonomije, jer se više ne ograničava na jednostruko povezivanje dva entiteta vezom *pripada*, nego se uvodi i mogućnost povezivanja jednog entiteta s više drugih. Na slici 2.5. prikazana je jednostavna ontologija predmeta. Svakom predmetu pripadaju student, profesor i ocjena za svakog studenta. Svaki od tih entiteta međusobno su povezani određenim relacijama, a svi su povezani sa središnjim entitetom – predmet. S obzirom na to da se u ovoj ontologiji nalazi više različitih relacija, ova se ontologija nije mogla prikazati kao stablo, jer različite relacije ne predstavljaju strogo hijerarhijsku strukturu. U takvim slučajevima gdje se ontologija ne može prikazati kao stablo, koriste se različiti oblici mrežnih struktura za prikaz i skladištenje entiteta.



Slika 2.5. Ontologija školskog predmeta

Dakle, da bi se ontologija mogla smatrati taksonomijom, mora postojati samo hijerarhijski odnos između entiteta – roditelj i dijete, a samim time entiteti na istoj razini ne mogu biti međusobno povezani. Stoga, možemo zaključiti da je taksonomija podvrsta ontologije, ako ima samo jednu relaciju i to hijerarhijsku nad svim entitetima.

2.5.1. Sigurnost baze znanja

Kao i za svaki drugi oblik informacijskog sustava, tako i za bazu znanja valja primijeniti CIA (engl. *confidentiality, integrity, availability*) trokut, koji je temeljni koncept koji se slijedi u svrhu očuvanja sigurnosti informacijskoga sustava. Po Stallingsu [13], tri koncepta CIA trijade objedinjuju osnovne sigurnosne ciljeve za informacijske usluge. U kontekstu baze znanja, CIA trokut odnosi se na sljedeće:

- **Povjerljivost** (engl. *Confidentiality*) – odnosi se na potrebu da informacije iz baze znanja budu dostupne samo ovlaštenim korisnicima. Preciznije, niti jedna informacija iz baze znanja ne bi smjela doći do korisnika kojemu nije namijenjena. Povjerljivost se u kontekstu baze znanja najčešće implementira kao kontrola pristupa korištenjem sustava autentikacija i autorizacije.
- **Integritet** (engl. *Integrity*) – osigurava da su podaci u bazi znanja ispravni, odnosno da nisu neovlašteno izmijenjeni od korisnika koji nije trebao imati pristup toj funkcionalnosti. Tako, prilikom dizajniranja sustava za dodavanje entiteta u bazu treba uzeti u obzir da samo ovlašteni korisnici imaju pravo utjecati na strukturu baze znanja. To se najčešće postiže mehanizmima za verifikaciju podataka ili evidentiranjem svih aktivnosti mijenjanja baze znanja.
- **Dostupnost** (engl. *Availability*) – osigurava da su sve informacije iz baze znanja uvijek dostupni za korištenje ovlaštenim korisnicima. Na dostupnost baze znanja značajno utječe struktura baze koja se postiže, između ostaloga, i gore navedenim organizacijskim metodama strukturiranja podataka. Hijerarhijska struktura čini bazu podataka jednostavnom za pretraživanje i dohvaćanje podataka što pozitivno utječe na dostupnost, dok kod mrežnih struktura ontologije treba koristiti napredne algoritme pretraživanja koji su nešto kompleksniji, ali će rezultati dohvaćanja podataka biti mnogo sofisticiranije, pa će i korisnici dobiti preciznije odgovore. Također je potrebno zaštititi bazu znanja od zlonamjernih DDoS napada (engl. *Distributed Denial-of-Service*), koji bazu znanja mogu preplaviti lažnim prometom i time autoriziranim korisnicima onemogućiti dohvaćanje potrebnih im resursa iz baze znanja.

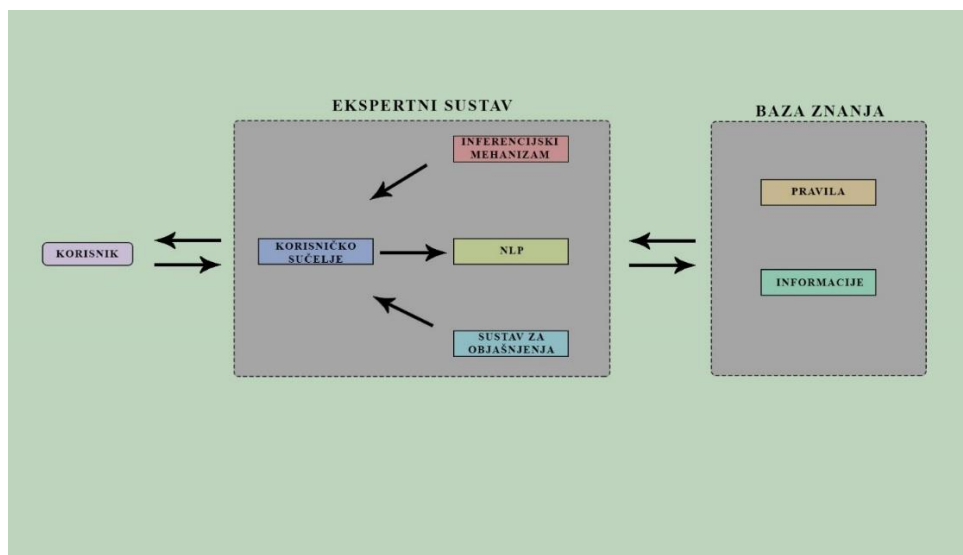
Kako bi se u konačnici zadovoljila sigurnost baze znanja, potrebno je analizirati svaki algoritam i sigurnosni mehanizam koji se odnosi na bazu znanja te istražiti moguće napade na sigurnost baze kroz njihove nedostatke. Nažalost, zbog ljudskog faktora i stalnog razvoja novih metoda napada, nikada nije moguće sa sigurnošću tvrditi da će baza znanja biti potpuno sigurna, no ako se slijede načela CIA trokuta, rizik za narušavanje sigurnosti baze znanja smanjen je na minimum.

2.6. Ekspertni sustavi

Prema Pavliču [14], ekspertni je sustav (engl. *expert system*) vrsta informacijskog sustava zasnovanog na upravljanju znanjem. To su automatizirani programi, koji korisnicima daju odgovore na pitanja vezana uz određeno strukovno polje, a pored toga imaju mogućnost zaključivanja i donošenja odluka, pri čemu se koriste duboke neuronske mreže. Znanje koje se koristi prilikom dizajniranja ekspertnih sustava modelirano je od strane stručnjaka u tom području za koje je ekspertni sustav vezan, a nakon toga znanje se unosi u računalo pri čemu se koristi baza znanja.

Baza znanja temeljni je dio svakog ekspertnog sustava, upravo iz razloga jer je u njoj pohranjeno cijelo znanje potrebno za izvođenje zaključaka od strane sustava. Baza znanja u ekspertnim sustavima mora se redovito održavati, kako bi se očuvala točnost, naročito s obzirom na nove napretke i otkrića u području znanja kojim baza raspolaže. Korištenje ekspertnog sustava izrazito je praktično, jer na brz i efikasan način omogućava pristup znanju o određenom području, a pri tom smanjuju mogućnost ljudske greške zbog pravila kojeg programi slijede. Unatoč tome, i dalje je ljudski faktor presudan u tumačenju rezultata koje daje ekspertni sustav, jer je pogreška programa uvijek moguća, u kontekstu ekspertnih sustava naročito zbog ograničenosti domene znanja kojom baza raspolaže.

Kao što je i vidljivo na slici 2.6., ekspertni se sustav sastoji od baze znanja, inferencijskog mehanizma, sustav za objašnjenja i korisničkoga sučelja. Korisnik preko korisničkoga sučelja ostvaruje komunikaciju s ekspertnim sustavom. Obično se koristi tekstualni oblik sučelja, ali ovisno o potrebama, moguće je uvesti i druge oblike interakcija. Nakon toga, obično koristeći obradu prirodnog jezika, sustav interpretira korisnikov unos i tada pristupa bazi znanja. U bazi znanja nalaze se sve potrebne informacije koje ekspertni sustav treba da bi generirao odgovor na upit korisnika, a prethodno je stručnjak za područje kojim se bavi ekspertni sustav definirao pravila i informacije. Zatim, s pomoću inferencijskog mehanizma sustav koristi pravila i informacije dohvaćene iz baze znanja i koristeći mehanizme strojnog učenja, dolazi do zaključaka koji su korisniku potrebni. Kada sustav kreira zaključak, generira se odgovor za korisnika u prigodnom obliku. Nerijetko su ekspertni sustavi opremljeni sustavom za objašnjenja koji uz generirani odgovor korisniku pružaju informaciju o tome kako je sustav došao do zaključaka, koje je informacije i algoritme koristio.



Slika 2.6. Prikaz ekspertnog sustava s NLP-om

2.7. Retrieval-Augmented Generation

Retrieval-Augmented Generation mehanizam je s pomoću kojega se povećava preciznost modela umjetne inteligencije, koristeći podatke iz vanjske baze, a ti podaci čine velike jezične modele sposobnijima generirati bogatije i preciznije odgovore, što ga čini korisnim alatom prilikom generiranja *chatbotova*. Kao što i sam naziv upućuje, *Retrieval-Augmented Generation* mehanizam sastavljen je od tri dijela: pretraživanje, proširenje i generiranje.

Pretraživanje se odnosi na pronalaženje informacija koje su značajne s obzirom na upit korisnika. U procesu pretraživanja koristi se obrada prirodnog jezika, nakon čega se identifikacijom ključnih riječi upita korisnika, određenim algoritmom pretražuju relevantne informacije iz baze znanja.

Augmentacija je druga faza *Retrieval-Augmented Generation* mehanizma u kojoj se na informacije iz prethodnog koraka procesuiraju i prilagođavaju s obzirom na korisnikov upit.

Zadnja faza *Retrieval-Augmented Generation* mehanizma jest generiranje u kojoj se stvaraju odgovori na upite korisnika temeljene na prethodna dva koraka koristeći duboke neuronske mreže za obradu prirodnog jezika. S obzirom na veliku količinu podataka kojom raspolažu ovakvi sustavi, uspješno upravljanje podacima kao i korištenje pravih optimizacijskih metoda za pretraživanje i obradu podataka od krucijalnog su značaja za generiranje odgovora kojima će korisnici biti zadovoljni.

2.8. Pregled dostupnih softvera za organizaciju baze znanja za chatbot

S obzirom na to da su *chatbotovi* postali alati koji su svakodnevno korišteni, nastala je velika potreba na tržištu za softverskim rješenjima koja omogućavaju organizaciju baze znanja specifično prilagođene za chatbotove, čime se postiže učinkovitija interakcija između korisnika i *chatbotova*. Današnji softveri nude raznolike funkcionalnosti, među kojima su upravljanje dokumentima, pretraga i kategorizacija dokumenata, kao i njihova integracija s *chatbotovima*.

Za usporedbu je analizirano pet popularnih rješenja: Zendesk, Helpjuice, Slite, Document360 i Notion.

Zendesk [15] je softver namijenjen za upravljanje korisničkim interakcijama, a baza znanja samo je jedna od usluga koje nudi. Zendesk nudi kreiranje i organiziranje dokumenata unutar kategorija i sekcija, a sve to uz podršku alata umjetne inteligencije. Velika je prednost što pri kreiranju dokumenata nudi tekst kreiran alatima umjetne inteligencije na temelju natuknica koje korisnik unese, a također nudi i mogućnost semantičkog pretraživanja (engl. *semantic search*) koje omogućava pretraživanje dokumenata s obzirom na kontekst, a ne na doslovno podudaranje upisanih riječi s onima iz dokumenata kakvo je uobičajeno kod pretraživanja. Ima mogućnost integracije s raznim drugim alatima te nudi pristup analitičkim metodama koje daju povratnu informaciju o dokumentima, kao što je broj pregleda dokumenta kroz određeni period.

Helpjuice [16] je specijalizirani alat za organizaciju baze znanja, namijenjen tvrtkama koje žele stvoriti centralizirani sustav za pohranu i pristup informacijama. Podržava kreiranje dokumenata, a organizira ih u hijerarhijskoj strukturi po kategorijama i potkategorijama. Helpjuice omogućava napredno pretraživanje i filtriranje dokumenata, ali nema specifičnih alata umjetne inteligencije za pretraživanje. Posebna je prednost mogućnost učitavanja različitih formata datoteka spremljenih na uređaju te mogućnost kontrole pristupa dokumentima te njihovom uređivanju uz praćenje verzija.

Slite [17] je alat korišten primarno za kolaboraciju u timu i stvaranje bilješki kao i njihovo dijeljenje, ali nudi i opciju organizacije baze znanja. Za organizaciju dokumenata koristi mape i podmape, a omogućuje i pregled verzija uređivanja dokumenata. Pri kreiranju dokumenata, Slite omogućuje automatsko skraćivanje teksta, prevođenje te ispravak gramatike i pravopisa, dok se za pretraživanje dokumenata koriste alati umjetne inteligencije te se korisnicima prikazuju samo oni odgovori kojima smiju pristupiti s obzirom na njihova prava pristupa.

Document360 [18] je aplikacija namijenjena upravljanju tehničkom dokumentacijom te pružanje podrške korisnicima. Omogućava organizaciju dokumenata putem hijerarhijske strukture po kategorijama, a također omogućava i učitavanje različitih formata dokumenata s uređaja. Aplikacija podržava više vrsta korisnika, a za sve korisnike i dokumente definirano je pravo pristupa. Za pretraživanje dokumenata aplikacija koristi alate umjetne inteligencije, što ubrzava proces pretrage dokumenata. Posebnost ove aplikacije je slanje upozorenja u stvarnom vremenu, pa se tako na promjenu sadržaja nekih dokumenata može poslati upozorenje nekim korisnicima kako se dokument izmijenio, što smanjuje mogućnost nesporazuma i davanja pogrešnih informacija.

Notion [19] je još jedna aplikacija koja je namijenjena za bolju timsku suradnju, a to postiže pohranjivanjem znanja u centraliziranu bazu. Omogućava stvaranje posebnih prava pristupa za stvaranje i pretraživanje dokumenata, a podržani su različiti formati dokumenata. Za pretraživanje dokumenata Notion ne koristi alate umjetne inteligencije, nego koristi oznake (engl. *tags*) i filtere koji omogućavaju jednostavniji pristup dokumentima, ali bez mogućnosti izdvajanja sadržaja.

3. IZRADA APLIKACIJE ZA ORGANIZACIJU BAZE ZNANJA

Zadatak završnog rada napravljen je na primjeru baze znanja sveučilišta. Sveučilište ima fakultete kojima pripadaju korisnici koji imaju pravo upravljati dokumentima i mapama u bazi znanja vezanim isključivo uz njihov fakultet. Web aplikacija izrađena je u programskom jeziku *Ruby* te koristeći razvojno okruženje *Ruby on Rails*. *Ruby on Rails* koristi *Model-View-Controller* arhitekturu, drugim riječima, *Rails* jasno odvaja modele, poglede i kontrolere u aplikaciji, a u konačnici odvaja implementacijske detalje od korisničkog sučelja.

Modeli (engl. *Models*) su podatkovne strukture aplikacije koje komuniciraju s bazom podataka i osiguravaju dohvaćanje, obradu i validaciju podataka te su potpuno neovisne o pogledima i kontrolerima u aplikaciji pa nemaju znanja o tome kako će se podaci modela prikazati korisnicima ili kako će podaci modela komunicirati s aplikacijom u pozadini.

Pogledi (engl. *Views*) predstavljaju korisničko sučelje, odnosno imaju zadaću prikazati podatke korisnicima na razumljiv i oku ugodan način i to samo one podatke koje im kontroleri proslijede, a to obično uključuje podatke nekih modela, pa shodno tome, pogledi ovise i o kontrolerima i o modelima.

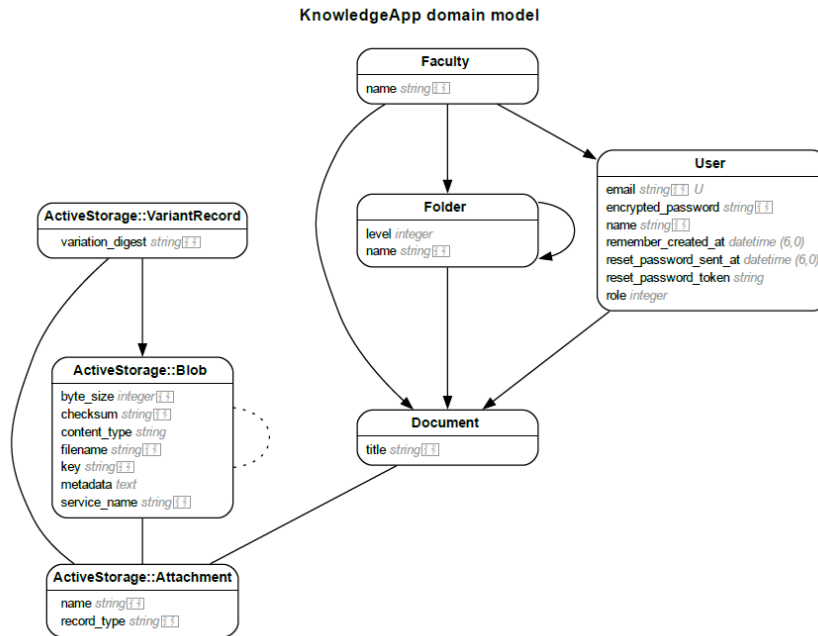
Kontroleri (engl. *Controllers*) su posrednička točka između pogleda i modela. Oni obrađuju zahtjeve korisnika unesene kroz korisničko sučelje te komuniciraju s modelima kako bi dohvatili ispravne podatke iz baze podataka te ih nakon obrade prosljeđuju korisnicima kroz pogled.

3.1. Funkcionalnost aplikacije

3.1.1. Baza podataka

U web aplikaciji korištena je *SQLite3* relacijska baza podataka u kombinaciji s alatom *Active Record* integriranog u okruženju *Rails*, koji omogućuje rad s bazom podataka kroz *Ruby* objekte, umjesto SQL upita. Uz pomoć *Active Record*a relacije između modela puno su jasnije definirane, a za bilo kakve promjene u tablicama u bazi podataka koriste se migracije.

Osnovni entiteti od kojih se sastoji baza podataka aplikacije i njihove relacije vidljive su na slici 3.1., a to su *Faculty*, *Folder*, *User* i *Document*.



Slika 3.1 Prikaz ER dijagrama za programsko rješenje

- *User* – model koji predstavlja korisnika u aplikaciji. Važniji atributi koje ima model *User* su *email*, *encrypted_password* i *name* tipa *string*, koji se koriste zbog autentikacije i razlikovanje korisnika te atribut *role* tipa *integer* koji predstavlja ulogu korisnika u aplikaciji. U aplikaciji postoje tri vrste korisnika ovisno o atributu *role*: *Admin*, *Faculty member* i *Regular user*. Ako je korisnik *Admin* ili *Faculty member* dobiva pravo upravljanja dokumentima i mapama te ima relaciju jedan naprema više s modelom dokumenta. U ovom se modelu nalazi i strani ključ koji referencira na fakultet pod uvjetom da korisnik pripada postojećem fakultetu.
- *Faculty* – model koji predstavlja fakultete u aplikaciji. Fakulteti su opisani atributom *title* koji im definira naziv po kojem će ih korisnici prepoznati. Fakulteti imaju relacije jedan naprema više s obzirom na ostale modele. Korisnici koji imaju ulogu *Faculty membera* pripadaju određenom fakultetu, u čijim mapama mogu kreirati nove mape ili dokumente.
- *Folder* – model koji predstavlja mape u aplikaciji. Mape su ključne za ostvarivanje hijerarhijske strukture baze znanja. Model je opisan atributom *name* po kojemu ga korisnici prepoznaju, a *Folderu* ga dodjeljuje *Admin* ili *Faculty Member* ovisno o tome tko ga stvara, Zbog potrebe za hijerarhijskom strukturom, svaka mapa u sebi može imati još mapa. Zbog toga postoji atribut *level*, koji predstavlja razinu svake mape u hijerarhijskoj strukturi, s tim da korijenska mapa ima *level* jednak nuli, a svaka sljedeća razina u hijerarhiji povećava razinu za jedan. *Folderi* imaju reference na fakultete kojima mogu, ali ne moraju pripadati,

te na dokumente koji se u njima nalaze. Kod *Foldera* imamo i reference na druge *Foldere*, jer svaki *Folder* može imati više djece, ali samo jednog roditelja.

- *Document* – model koji predstavlja dokumente koji se spremaju u bazi podataka aplikacije. Dokumenti su glavni dio baze znanja, jer se cijela aplikacija bazira na njihovom upravljanju i organizaciji. Ima atribut *title*, koji mu definira naziv po kojem se može pretraživati, a entiteti ovog modela opcionalno pripadaju entitetu modela *Faculty* te obavezno pripadaju određenom korisniku koji je stvorio dokument, te *Folderu* u kojem je dokument spremljen. Najvažnija stavka modela je referenca na *Attachment*, koja omogućava dodavanje datoteke modelu koristeći *Rails*ov sustav za upravljanje datotekama *Active Storage*, koji koristi modele *ActiveStorage::Blob*, *ActiveStorage::Attachment* i *ActiveStorage::VariantRecord*, koji se automatski kreiraju prilikom pokretanja migracije koja uvodi *Active Storage* u aplikaciju.

3.1.2. Autentikacija korisnika

Za autentikaciju korisnika (engl. *user authentication*) u aplikaciji korišten je *gem Devise*. *Devise* pruža jednostavnu i sigurnu autentikaciju, a funkcije koje su primarno korištene u ovoj aplikaciji su prijava i registracija. Kako bi se *Devise* mogao koristiti, potrebno ga je aktivirati. Prvi je korak dodati *Devise gem* u datoteku *Gemfile*. Nakon toga, potrebno je pokrenuti naredbu *bundle install* koja instalira sve *gemove* u datoteci *Gemfile* te generirati *Devise* konfiguraciju u terminalu. Nakon toga, moguće je kreirati modele koristeći *Devise*, a u pozadini će se same definirati rute i pogledi za potrebne funkcionalnosti koje *Devise* pruža, a koje se po potrebi mogu prilagoditi.

Registracija korisnika omogućena je samo za korisnike s ulogom *Regular usera*. Prilikom registracije korisnik u sučelje unosi podatke te ako su ispravni, stvara se novi entitet tipa *User* s unesenim podacima. *Admin* ima mogućnost dodjeljivanja drugih uloga korisniku ili stvaranja korisnika s određenom ulogom. Važan element koji pruža *Devise* je restrikcija pristupa neautoriziranim korisnicima, a radi tako da se prije određenih akcija ili pristupanja nekoj stranici provjeri je li korisnik prijavljen te ovisno o tome odlučuje hoće li prikazati stranicu ili izvršiti akciju koja se od kontrolera traži.

3.1.3. Kontroleri

U ovoj web aplikaciji, kontroleri su podijeljeni u devet datoteka, od kojih je svaka usmjerena na posebnu funkcionalnost aplikacije, koje su uglavnom vezane uz upravljanje entitetima prethodno opisanih modela na siguran i učinkovit način. U nastavku su opisani svi kontroleri u web aplikaciji.

- *ApplicationController* – predstavlja središnji kontroler kojega svi drugi kontroleri nasljeđuju. U njemu se nalaze funkcionalnosti zajedničke svim kontrolerima i metode kojima može pristupiti svaki kontroler.
- *Admin::FacultiesController* – kontroler koji se isključivo odnosi na upravljanje fakultetima od strane korisnika s ulogom administratora. Javne metode od kojih se sastoji su *index*, *new*, *create* i *destroy*.
 - *Index* – metoda koja omogućava administratorima prikaz svih postojećih fakulteta s njihovim glavnim podacima
 - *New* – metoda koja inicijalizira novi fakultet
 - *Create* – omogućava stvaranje novoga fakulteta i njegovo spremanje u bazu. U slučaju da korisnik pokuša stvoriti fakultet s nazivom koji već nosi neki fakultet, ispisan će se poruka pogreške, a pri uspješnom kreiranju i spremanju novog fakulteta, kontroler preusmjerava na popis svih fakulteta.
 - *Destroy* – metoda koja omogućava brisanje fakulteta i svih entiteta ovisnih o tome fakultetu uključujući korisnike, dokumente i mape koje su vezane uz taj fakultet. Pri neuspjehom pronalasku fakulteta, ispisan se poruka pogreške.
- *Admin::UsersController* – kontroler koji je također isključivo rezerviran za administratore u sustavu. Ovaj kontroler omogućava prikaz svih korisnika, njihovo pretraživanje te mogućnost stvaranja novih korisnika s određenim ulogama. Glavni je cilj ovog kontrolera omogućiti administratorima upravljanje korisnicima na pregledan način. Javne metode koje su definirane u ovom kontroleru su:
 - *New* – metoda koja inicijalizira novog korisnika
 - *Index* – metoda koja prikazuje popis svih korisnika. Ova metoda poziva i metode filtriranja korisnika po njihovim ulogama i fakultetima, pretraživanja po korisničkim imenima i e-mailovima te sortiranja korisnika prema datumu stvaranja tog korisnika.
 - *Create* – metoda koja stvara novog korisnika s obzirom na podatke koje je administrator unio.
- *DocumentsController* – ovaj kontroler omogućava prikaz, uređivanje, pretraživanje, brisanje i preuzimanje dokumenata koji su stvoreni.
 - *Show* – metoda koja omogućava prikazivanje pojedinačnih dokumenata. Svaki korisnik u sustavu može pristupiti sadržaju svakog postojećeg dokumenta u sustavu.

- *Index* – predstavlja stranicu koja prikazuje popis svih dokumenata u sustavu, a omogućuje pretraživanje po naslovu dokumenta, imenu korisnika koji je stvorio taj dokument te po datumu stvaranja dokumenta. Omogućuje filtriranje s obzirom na fakultet kojem dokument pripada ili s obzirom na korisnike koji su stvorili dokument. Sortiranje je omogućeno po datumu kreiranja dokumenta.
- *Update* – omogućuje uređivanje naslova i sadržaja dokumenta. Administrator ima pravo uređivati bilo koji dokument u sustavu, dok *Faculty member* može uređivati samo dokumente koji pripadaju fakultetu kojem pripada i on. Za uređivanje dokumenata koristi se *gem EasyMDE*, omogućava korisniku jednostavniji pristup uređivanju teksta.
- *Destroy* - omogućuje brisanje odabranog dokumenta. Isto kao i kod metode *update*, administrator ima pravo brisanje svih dokumenata, dok *Faculty member* ima pravo brisanja samo dokumenata koji pripadaju istom fakultetu kao i on.
- *Download* – javna metoda u kontroleru koja omogućava preuzimanje dokumenata u *markdown* formatu.
- *FacultiesController* – kontroler koji omogućava korisnicima pregled informacija o pojedinom fakultetu
 - *Show* – jedina je metoda u *FacultiesControlleru*. Omogućava prikazivanje detalja o fakultetu korisniku kao što su članovi koji pripadaju tom fakultetu i dokumenti koji pripadaju fakultetu
- *KnowledgeAppController* – kontroler koji služi kao početna stranica web aplikacije
 - *Index* – metoda koja prikazuje početni pregled aplikacije
- *RegistrationsController* – kontroler koji omogućava specifični oblik registracije korisnika, drugačiji od one koju pruža *Devise*. Sadrži metode *new* i *create*.
 - *New* – metoda koja inicijalizira *Users* i novi obrazac za registraciju
 - *Create* – metoda koja obrađuje podatke unesene u obrascu te ovisno o ispravnosti podataka, sprema novog korisnika nakon uspješne registracije.
- *UsersController* – ovaj kontroler omogućuje upravljanje korisničkim profilima. Javne metode koje ga opisuju su *edit*, *show*, *update* i *destroy*.
 - *Edit* – metoda koja omogućava prikaz stranice za izmjenu korisničkih podataka. Ako administrator pristupa uređivanju drugog korisnika, ima pravo promijeniti mu i ulogu.
 - *Show* – omogućuje prikaz detalja o svakom korisniku

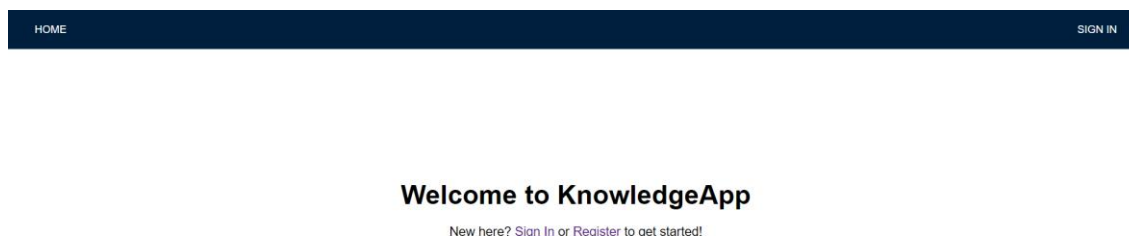
- *Update* – metoda koja omogućava provjeru i spremanje izmijenjenih podataka korisnika kojeg se pregledava
- *Destroy* – omogućava administratoru da briše korisnike iz sustava. Prilikom brisanja korisnika, svi dokumenti kojima je obrisan korisnik kreator dobivaju novog kreatora – administratora. Zbog očuvanja konzistentnosti dokumenata, administrator ne može obrisati svoj korisnički račun.
- *WorkspaceController* – predstavlja kontroler koji omogućava većinu manipulacija nad podacima koje modeli *Folder* i *Document* definiraju. Kako samo *Regular useri* nemaju pravo pristupiti akcijama kojima upravlja ovaj kontroler, za kontroler su definirane provjere autorizacije korisnika prije pokretanja akcije.
 - *New* – predstavlja javnu metodu unutar *WorkspaceControllera*, a zadužena je za pripremu svih potrebnih podataka za kreiranje novih entiteta modela *Folder* i *Document*.
 - *Create* – metoda koja obuhvaća stvaranje entiteta modela *Document* i modela *Folder*. Ovisno o odabiru korisnika, pozivaju se privatne metode *create_document* ili *create_folder*. Prilikom kreiranja *Foldera* poziva se metoda *create_folder*. Korisnik kroz sučelje odabire *Folder* koji će biti roditelj novostvorenom *Folderu* te će s obzirom na hijerarhijsku razinu roditelja, definirati vrijednost atributa *level* novostvorenog entiteta za jedan više od roditelja. Za odabir roditelja novog *Foldera*, kontroler ograničava *Faculty membera* na odabir samo onih roditelja koji pripadaju fakultetu kao i korisnik, dok *Admin* ima pravo kreiranja drugih *Foldera* u bilo koji postojeći *Folder* u sustavu. Prilikom kreiranja dokumenta poziva se metoda *create_document*. Kontroler omogućava manualno kreiranje dokumenta kroz sučelje, pri čemu se za uređivanje teksta koristi prethodno spomenuti *gem EasyMDE*, a također se može i učitati (engl. *upload*) datoteka s računala koja ima ekstenziju *.txt*, *.md*, *.pdf*, ili *.docx*, a koji se po potrebi prilikom spremanja, koristeći *gemove*, pretvaraju u dokument s *.md* ekstenzijom. Za datoteke s ekstenzijom *.docx* koristi se *Docx gem*, koji ih pretvara u *markdown*, a datoteke s ekstenzijom *.pdf* koriste *Pdf-reader gem* koji ih također pretvara u datoteke s *.md* ekstenzijom. Za datoteke s nastavcima *.md* i *.txt* nije potrebna nikakva posebna pretvorba.
 - *Delete_folder* – metoda koja omogućava brisanje postojećih *Foldera* iz sustava. *Admini* imaju pravo brisanja svakog *Foldera* u sustavu, osim korijenskog i *Foldera* koji imaju referencu na *Faculty* i atribut *level* im je jednak jedan, što predstavlja

osnovnu mapu fakulteta. *Faculty member* ima pravo brisanja samo onih *Foldera* koji se referenciraju na njegov *Faculty* i atribut *level* je strogo veći od jedan.

3.2. Opis rada aplikacije

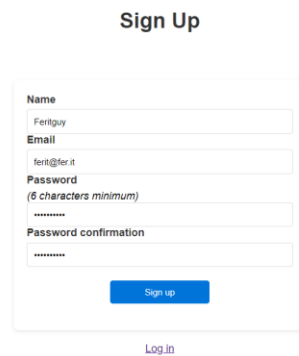
3.2.1. Početna stranica i korisnički pristup

Prilikom posjećivanja web aplikacije, korisnika se preusmjerava na početnu stranicu. Ako korisnik nije prijavljen u sustav, na početnoj stranici ima opcije za registraciju ili prijavu kako je prikazano na slici 3.2.



Slika 3.2. Prikaz početne stranice za neprijavljenog korisnika

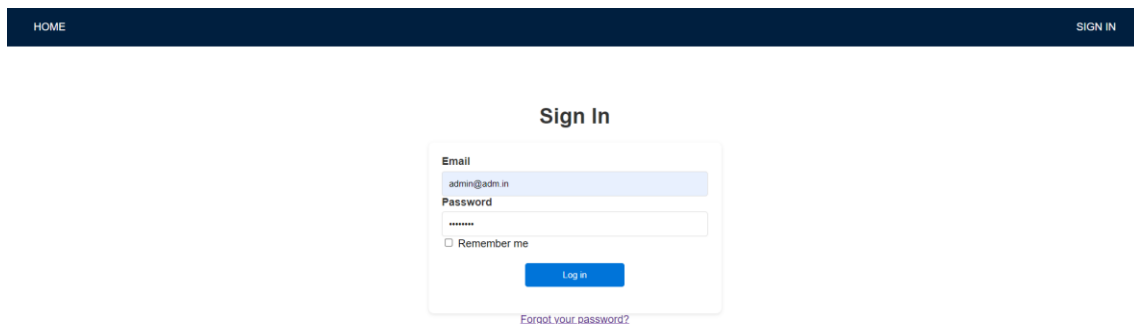
U slučaju da korisnik nema korisnički račun u sustavu, može ga kreirati tako da klikne na hipervezu *Register*, čime se na domenu URL-a dodaje *users/sign_up*, a to otvara novu stranicu vidljivu na slici 3.3, gdje se korisnik aplikacije može registrirati. Unosom podataka za kreiranje novog korisnika, a to su polja za ime, e-mail adresu, zaporku i potvrdu zaporke te pritiskom gumba *Sign up* kreira se novi korisnik te ga se, prijavljenog u sustav, preusmjerava na početnu stranicu. Registracijom, uloga korisnika automatski postaje *Regular user* te takav korisnik nema pravo uređivanja entiteta u bazi podataka, jer bi drugačiji pristup u implementaciji zadatka rezultirao mogućim neželjenim posljedicama, kao što su mogućnost bilo kojeg korisnika da upravlja tuđim dokumentima i mapama.



The image shows a 'Sign Up' form with the following fields: 'Name' (containing 'Feritguy'), 'Email' (containing 'ferit@fer.it'), 'Password' (with a note '(6 characters minimum)' and masked with dots), and 'Password confirmation' (also masked with dots). A blue 'Sign up' button is at the bottom. Below the form is a purple 'Log in' link.

Slika 3.3. Prikaz stranice za registraciju korisnika

Ako se pristupnik početnoj stranici odluči za prijavu i pritisne *Sign in* hipervezu, ona će ga preusmjeriti na stranicu za prijavu korisnika vidljivo na slici 3.4. a URL adresa će na postojeću domenu dodati *users/sign_in*. Za prijavu u sustav, korisnik jednostavno upisuje podatke o svom računu: e-mail i zaporka. Unos ispravne zaporka za e-mail adresu postojećeg korisnika u sustavu rezultira uspješnom autorizacijom korisnika od strane *gema Devise* te se korisnik preusmjerava ponovno na početnu stranicu. U slučaju pogrešnog unosa, korisnik se ne može autorizirati te se, ovisno o problemu, prikazuje poruka pogreške.



The image shows a 'Sign In' form with the following fields: 'Email' (containing 'admin@admin.in') and 'Password' (masked with dots). There is a checkbox for 'Remember me' which is unchecked. A blue 'Log in' button is at the bottom. Below the form is a purple link that says 'Forgot your password?'

Slika 3.4. Prikaz stranice za prijavu korisnika

Ako je korisnik koji pristupa početnoj stranici prijavljen, na navigacijskoj će se traci pojaviti određeni broj dodatnih opcija, ovisno o ulozi prijavljenog korisnika. Na stavci navigacijske trake *My profile*, korisnik se može preusmjeriti na stranicu koja prikazuje njegove korisničke podatke.

Na slici 3.5. vidi se stranica koja prikazuje podatke o korisniku kao što su korisničko ime, e-mail, uloga korisnika, fakultet kojemu pripada te broj dokumenata koje je korisnik kreirao kao i njihova lista. Gumb *Edit profile* omogućuje uređivanje podataka o korisničkom računu, a stranica na koju gumb preusmjerava prikazuje slika 3.6.

Title	Faculty	Path	Created At
FAQ_Courses	FAZOS	UNIOS/FAZOS/FAQ/FAQ_Courses.md	August 22, 2024 15:16:01
chatbot_openai	FAZOS	UNIOS/FAZOS/chatbot_openai.md	August 22, 2024 15:28:16

Slika 3.5. Prikaz korisničkih podataka

Edit User

Name: Petar

Email: fazos@faz.os

[Save Changes](#)

[Back to Profile](#)

Slika 3.6. Prikaz sučelja za uređivanje korisničkih podataka

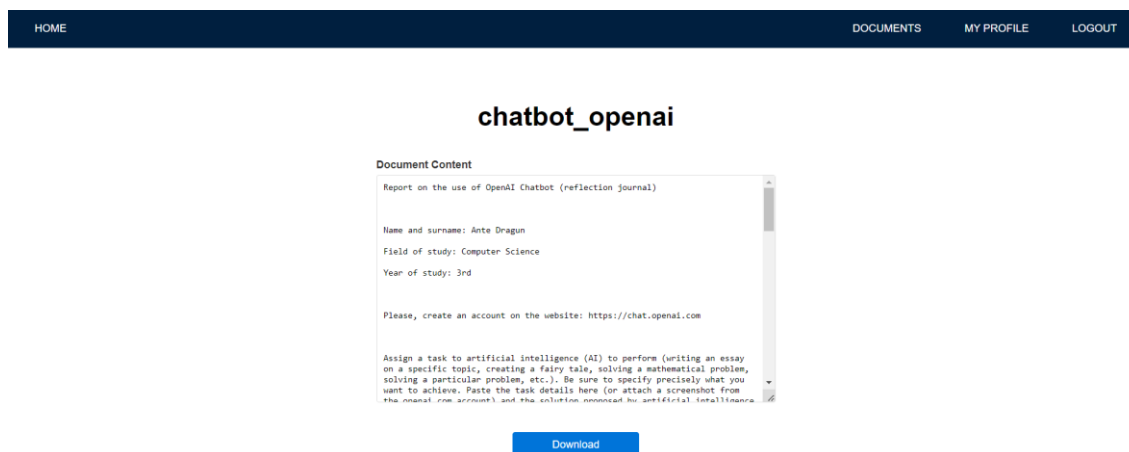
Pritiskom na gumb *Save changes* kontroler provjerava ispravnost novih unesenih podataka te ako su ispravni, korisniku se izmjenjuju podaci u bazi podataka te ga se preusmjerava na njegov korisnički profil.

3.2.2. Prikaz i uređivanje dokumenata

Svaki autorizirani korisnik može pristupiti stranici s prikazom svih spremljenih dokumenata u bazi znanja i to pritiskom na hipervezu s oznakom *Documents* na navigacijskoj traci ili ručnim

dodavanjem putanje */documents* na domenu URL adrese. Dokumenti su prikazani u tablici u kojoj je prikazan naslov dokumenta, fakultet kojemu pripada, put po *Folderima* do dokumenta, korisnika koji je kreirao dokument te datum i vrijeme zadnjeg ažuriranja dokumenta. Osim samog izlistanja svih dokumenata, stranica omogućava i sustavno pretraživanje, filtriranje i sortiranje dokumenata u tablici. Pretraživanje je omogućeno po atributima dokumenta kao što su naslov, fakultet kojemu pripada, korisniku koji je stvorio dokument i vremenu kada je nastao. Filtriranje je omogućeno po dvama kriterijima: fakultetu i korisniku, dok je sortiranje omogućeno s obzirom na datum kreiranja dokumenta i naslov uzlazno i silazno. U tablici dokumenata, moguće je pristupiti svakom dokumentu zasebno klikom na hipervezu naslova dokumenta, a moguće je pristupiti i korisničkom profilu korisnika koji je kreirao određeni dokument.

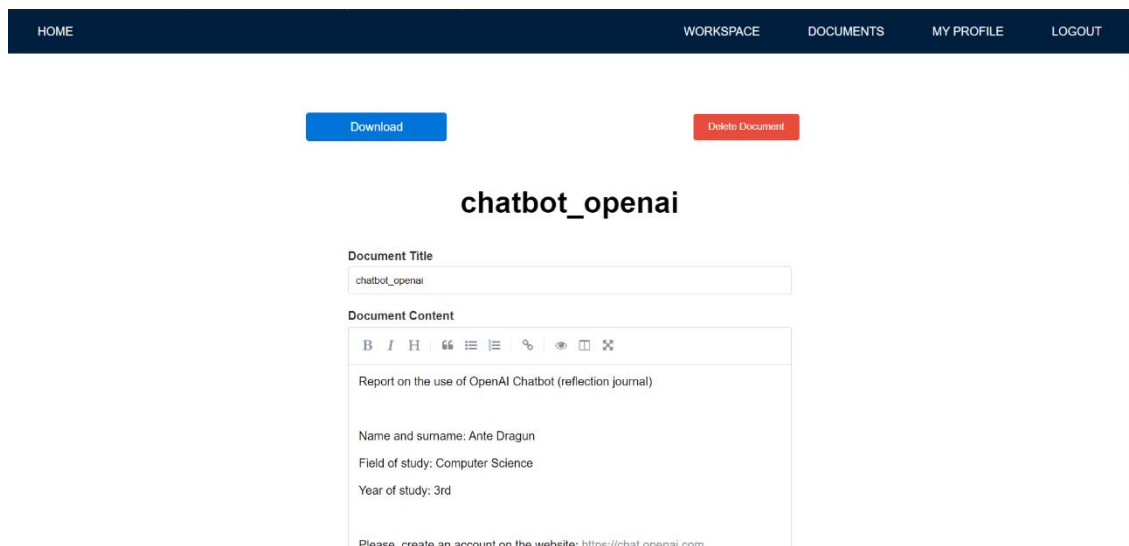
Klikom na hipervezu naslova dokumenta, prikazuje se nova stranica koja prikazuje trenutni sadržaj dokumenta kojemu korisnik pristupa. Ako korisnik ne pripada fakultetu kao i dokument ili ako korisnik nije *admin*, tada on nema pravo uređivanja dokumenta, ali na stranici i dalje može preuzeti dokument u *markdown* formatu te osim toga može u pregledniku vidjeti sadržaj trenutnog dokumenta. Na slici 3.7. vidi se prikaz stranice za korisnika koji nema pravo uređivanja dokumenta.



Slika 3.7. Stranica za prikaz i preuzimanje sadržaja dokumenta

S druge strane, ako je korisnik ovlašten izmijeniti dokument, sučelje izgleda ponešto drugačije, a vidljivo je na slici 3.8. Sada, osim suhoparnog pregleda sadržaja dokumenta i mogućnosti preuzimanja istog, korisnik ima mogućnost brisanja dokumenta iz baze znanja, kao i njegovog uređivanja kroz sučelje koje podržava i gumbе za uređivanje teksta kakvo podržavaju *markdown*

datoteke, a to su podebljanje slova, kurziv, umetanje zaglavlja (engl. *header*), umetanje citata, umetanje numeriranog popisa ili popisa s grafičkom oznakom te umetanje veza na web stranice.

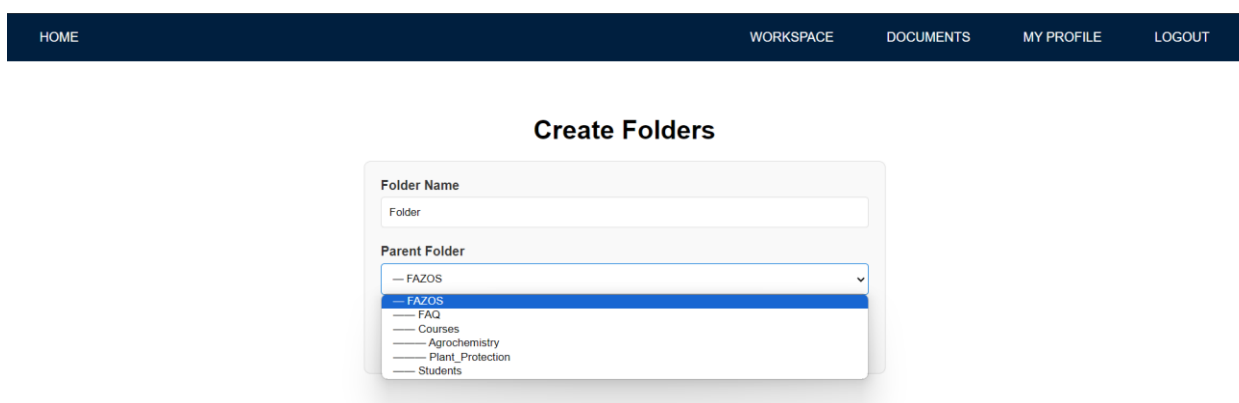


Slika 3.8. Prikaz korisničkog sučelja za uređivanje dokumenta

3.2.3. Upravljanje dokumentima i mapama

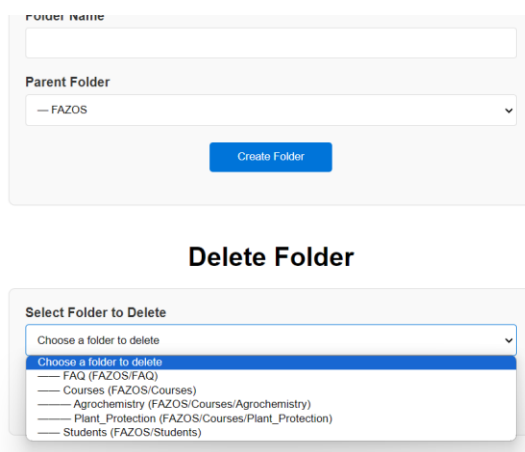
Pritiskom na hipervezu *Workspace* u navigacijskoj traci, korisnici se preusmjeravaju na stranicu gdje mogu dodavati i brisati *Foldere* te stvarati nove dokumente te odabrati mjesto gdje će stvoreni dokument spremi.

Na slici 3.9. prikazan je dio stranice koji omogućava stvaranje novog *Foldera*. Prilikom stvaranja novog *Foldera* potrebno je unijeti ime *Foldera* te postojeći *Folder* koji će mu biti roditelj. *Faculty member* za roditelja može odabrati samo one *Foldere* koji pripadaju njegovom fakultetu. Pritiskom na gumb *Create Folder*, kontroler u pozadini dodjeljuje ostale podatke entitetu te ga, ako su podaci ispravni, sprema u bazu podataka, a korisnika obavještava o uspješnosti kreiranja novog *Foldera*.



Slika 3.9. Stvaranje novog *Foldera* na stranici za upravljanje dokumentima i mapama

Stranica omogućuje i brisanje *Foldera* koje su korisnici fakulteta prethodno stvorili, a njen prikaz vidljiv je na slici 3.10. Kako bi se obrisao *Folder*, potrebno ga je odabrati u padajućem izborniku te pritisnuti gumb *Delete Folder*. Brisanjem *Foldera* brišu se i svi *Folderi* koji su ispod njega po hijerarhiji, a povezani su s njim, kao i svi dokumenti koji pripadaju takvim *Folderima*. Zbog mogućnosti da određeni *Folderi* imaju isto ime, a imaju drugačiji put od korijenskog *Foldera*, osim imena *Foldera* u padajućem je izborniku prikazana i putanja do *Foldera* počevši od glavnog *Foldera* korisnikovog fakulteta.



Slika 3.10. Brisanje *Foldera* na stranici za upravljanje dokumentima i mapama

Na slici 3.11. prikazan je dio stranice koji omogućuje stvaranje novih dokumenata. Za stvaranje novog dokumenta potrebno je upisati ispravno ime novog dokumenta, učitati datoteku dokumenta ili ručno upisati njegov sadržaj, odabrati iz padajućeg izbornika mjesto gdje će se dokument spremiti te pritisnuti gumb *Save Document*. Dokumenti se mogu učitati kao datoteka s računala pritiskom na gumb *Odaberi datoteku* ili se sadržaj dokumenta može ručno upisati u sučelje predviđeno za unos sadržaja dokumenta, koje se koristilo i na stranici za uređivanje dokumenata, a čiji je način rada prethodno objašnjen.

Slika 3.11. Stvaranje novog dokumenta unosom sadržaja u sučelje

Na posljertku, na dnu su stranice, zbog lakše navigacije, izlistani svi dokumenti koje prijavljeni korisnik ima pravo uređivati. Na slici 3.12. prikazana je tablica svih dokumenata koje korisnik s fakulteta FAZOS može uređivati. Klikom na naslov dokumenta, korisnika se preusmjerava na detaljan pregled tog dokumenta, gdje korisnik može detaljnije s njim upravljati, što je prethodno objašnjeno.

Editable Documents

Title	Path	Created By	Last Updated
FAQ_Courses	UNIOS/FAZOS/FAQ/FAQ_Courses.md	Petar	August 22, 2024 15:16:01
general_informations	UNIOS/FAZOS/general_informations.md	Petar	August 22, 2024 19:34:22
test	UNIOS/FAZOS/Students/test.md	Petar	August 22, 2024 19:31:14

Slika 3.12. Primjer tablice svih dokumenata koje korisnik može urediti

3.2.4. Opis administratorskih ovlasti

U većini opisanih funkcionalnosti aplikacije, administratori u odnosu na druge vrste korisnika, imaju veći opseg mogućnosti i veća prava pri manipulaciji podacima. Prilikom prijave u sustav, administratoru se kroz navigacijsku traku omogućava pristup stranicama koje drugim korisnicima nisu.

Za upravljanje korisnicima, administrator u navigacijskoj traci može odabrati *Users*, koji ga vodi na sučelje gdje ima prikaz svih postojećih korisnika u sustavu s mogućnostima pretraživanja,

filtriranja po korisničkim ulogama i fakultetima te sortiranja korisnika, a klikom na ime korisnika, pristupa stranici s informacijama o njegovom profilu, gdje ima gumbе za brisanje korisnika i uređivanje podataka o korisniku uključujući njegovu ulogu. Prilikom brisanja korisnika, autorstvo svih kreiranih dokumenata od strane tog korisnika kontroler prebacuje na administratora koji ga briše iz sustava. Cijelo je sučelje s navigacijskom trakom za listu svih korisnika prikazano na slici 3.13.

Name	Email	Role	Faculty	Created At
Petar	fazos@faz.os	Faculty member	FAZOS	August 22, 2024 15:15:01
Feritovic	feri@fer.it	Regular user	N/A	August 22, 2024 14:41:59
admin	admin@adm.in	Admin	N/A	August 18, 2024 08:50:44

Slika 3.13. Stranica za upravljanje korisnicima rezervirana za administratore

Kako je vidljivo na slici 3.13., u sučelju postoji i gumb *Create new user*, koje omogućava kreiranje novog korisnika od strane administratora, a stranica za kreiranje novog korisnika prikazano je na slici 3.14. Za razliku od kreiranja korisničkih računa putem registracije, na ovakav način moguće je prilikom kreiranja korisnika automatski mu dodijeliti drugačiju ulogu te ovisno o ulozi, moguće je dodati i fakultet, a pravila za kreiranje korisnika putem ovog sučelja jednaka su kao i kod registracije.

Create New User

Name
Marko

Email
marko.markovic@marko.vic

Password

Password confirmation

Role
Faculty member

Faculty
FAZOS

Create User

Slika 3.14. Stranica za kreiranje novog korisnika od strane administratora

Osim za upravljanje dokumentima, administratori imaju još jednu stranicu kojoj ekskluzivno samo oni mogu pristupiti, a to je stranica za upravljanje fakultetima prikazana na slici 3.15.

All faculties

ID	Name	Number of Members	Number of Documents	Created At
2	EERIT	0	2	August 21, 2024 09:01:54
4	FAZOS	1	3	August 22, 2024 15:14:27

New Faculty

Slika 3.15. Prikaz stranice za upravljanje fakultetima

Prikazana stranica omogućava brzi pristup svim fakultetima, a pritiskom na gumb *New Faculty*, administratora se preusmjerava na stranicu za kreiranje novog fakulteta. Kreiranjem novog fakulteta, kreira se i novi *Folder* s imenom fakulteta gdje će korisnici moći stvarati druge *Foldere* i spremati dokumente u njih. Pritiskom na ime fakulteta u tablici, administratora se preusmjerava na stranicu s detaljima o odabranom fakultetu (slika 3.16.). Na toj stranici korisnik ima prikaz o svim detaljima toga fakulteta, uključujući njegove članove i dokumente koji se u njemu nalaze.

Isključivo administratori imaju pravo obrisati fakultet, a brisanjem fakulteta brišu se i svi njegovi korisnici, mape i dokumenti.

The screenshot shows the FERIT web application interface. At the top is a dark blue navigation bar with links: HOME, USERS, FACULTIES, WORKSPACE, DOCUMENTS, MY PROFILE, and LOGOUT. Below the navigation bar, the word "FERIT" is centered, followed by a red button labeled "Delete Faculty". Underneath is the section "Faculty Members" with a table containing one row: Name (Ante), Email (ante@ant.ic), and Role (Faculty member). Below that is the section "Documents" with a table containing two rows: Title (Upute_za_zavrsni_rad), Path (UNIOS/FERIT/Courses/Upute_za_zavrsni_rad.md), Created By (admin), and Last Updated (August 22, 2024 15:13:30); and Title (listOfStudents), Path (UNIOS/FERIT/Students/listOfStudents.md), Created By (admin), and Last Updated (August 22, 2024 15:14:17).

Name	Email	Role
Ante	ante@ant.ic	Faculty member

Title	Path	Created By	Last Updated
Upute_za_zavrsni_rad	UNIOS/FERIT/Courses/Upute_za_zavrsni_rad.md	admin	August 22, 2024 15:13:30
listOfStudents	UNIOS/FERIT/Students/listOfStudents.md	admin	August 22, 2024 15:14:17

Slika 3.16. Stranica koja omogućava prikaz podataka o fakultetu

Po pitanju ostalih korisničkih funkcionalnosti, administratori, iako ne pripadaju niti jednom fakultetu, imaju pravo upravljanja svim dokumentima i mapama u bazi znanja osim korijenskog *Foldera*, imaju pristup uređivanju svakoga dokumenta te imaju mogućnost upravljati podacima svih korisnika.

4. ZAKLJUČAK

U ovom je radu detaljno objašnjen način rada suvremenih *chatbotova* s naglaskom na kvalitetnu organizaciju baze znanja. Istraženi su svi ključni elementi koji omogućuju njegovu funkcionalnost kao što su obrada prirodnog jezika, strojno učenje, veliki jezični modeli te sigurnost baze znanja, a izrađena je i web aplikacija u okruženju *Ruby on Rails* koja omogućava upravljanje dokumentima u bazi znanja.

Teorijski dio rada objedinjuje nekoliko tema potrebnih za razumijevanje rada *chatbotova*, kao i povijesni pregled razvoja *chatbota* u moćan alat kakav je danas. Obrada prirodnog jezika predstavlja temelj interakcije između *chatbota* i korisnika, a korištenje jezičnih modela istaknuto je kao ključna točka u izgradnji naprednijih sustava. Obrada prirodnog jezika omogućava *chatbotovima* da prepoznaju obrasce u jeziku te razumiju kontekst korisničkih upita te pružanja relevantnih odgovora. Analizirane su i različite grane umjetne inteligencije, s posebnim naglaskom na duboko učenje, koje koristi umjetne neuronske mreže za učenje, a one se pak koriste za stvaranje jezičnih modela. Zatim je obrađen koncept baze znanja, koja predstavlja organizirani sustav koji omogućuje učinkovito i sigurno dohvaćanje dokumenata. Objašnjene su i najčešće metode strukturiranja baze znanja – taksonomija i ontologija. Taksonomija se oslanja na strogu hijerarhijsku strukturu za organizaciju podataka, dok je kod ontologije naglasak na relaciji između entiteta čime se postiže veća preciznost u odgovaranju na specifične upite. Očuvanje povjerljivosti, integriteta i dostupnosti podataka u bazi znanja naglašeno je kao ključ za održavanje sigurnosti čitave baze. Istraženi su i ekspertni sustavi, koji primjenjujući alate umjetne inteligencije te koristeći znanje iz baze, rješavaju kompleksne probleme u specifičnim područjima. Na kraju je objašnjen i *Retrieval-Augmented Generation* mehanizam, koji povećava preciznost *chatbotova* koristeći podatke iz eksternih baza, što ih čini sposobnima generirati još preciznije odgovore.

Praktični dio rada usmjeren je na izradu web aplikacije za organizaciju baze znanja za *chatbot* u okruženju *Ruby on Rails*. Aplikacija pruža rješenje za upravljanje dokumentima unutar baze znanja sveučilišta. Sveučilište ima više fakulteta, a fakulteti imaju članove koji upravljaju mapama i dokumentima relevantnima za svoj fakultet. Omogućeno je jednostavno učitavanje dokumenata s uređaja u različitim tekstualnim formatima, kao i ručno stvaranje dokumenata kroz sučelje. Aplikacija omogućava korisnicima s različitim ulogama pravo pristupa i upravljanja određenim dokumentima i mapama unutar hijerarhijske strukture baze znanja, što uključuje dodavanje, pretraživanje, brisanje i uređivanje.

Ova se aplikacija može nadograditi dodavanjem novih funkcionalnosti kao što su napredno pretraživanje i stvaranje analitičkih izvješća. Napredno pretraživanje moglo bi biti omogućeno dodavanjem metapodataka modelu dokumenta te izmjenama metoda za pretraživanje dokumenata, dok bi stvaranje analitičkih izvješća pružilo uvid u različite aspekte korištenja sustava te bi na temelju njih mogao biti kreiran plan za poboljšano upravljanje dokumentima.

LITERATURA

- [1] A. M. Turing, Computing Machinery And Intelligence. *Mind*, LIX(236), 433–460.
- [2] J. Weizenbaum, ELIZA---a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 36–45.
- [3] R. S. Wallace, *The Anatomy of A. L. I. C. E.*, 2009.
- [4] J. Cahn, CHATBOT: Architecture, Design, & Development [online], University of Pennsylvania, 2017, dostupno na: https://www.academia.edu/37082899/CHATBOT_Architecture_Design_and_Development/ [13. 9. 2024.]
- [5] Ipsos, How do Americans feel about Generative AI? It's complicated [online], Ipsos, 2023, dostupno na: <https://www.ipsos.com/en-us/how-do-americans-feel-about-generative-ai-its-complicated> [13. 9. 2024.]
- [6] P. M. Nadkarni, L. Ohno-Machado, W.W. Chapman, Natural language processing: an introduction, *JAMIA*, 2011, dostupno na: <https://pubmed.ncbi.nlm.nih.gov/21846786/> [13. 9. 2024.]
- [7] E. Alpaydin, *Introduction to Machine Learning*, Third Edition, Massachusetts Institute of Technology, 2014.
- [8] OpenAI, GPT-4 research [online], OpenAI, San Francisco, 2023, dostupno na: <https://openai.com/index/gpt-4-research/> [13. 9. 2024.]
- [9] J. Devlin, M. W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [online], arXiv, 2019, dostupno na: <https://arxiv.org/pdf/1810.04805/> [13. 9. 2024.]
- [10] Meta, Introducing Llama 3.1: Our most capable models to date [online], Meta, Menlo Park, California, 2024, dostupno na: <https://ai.meta.com/blog/meta-llama-3-1/> [13. 9. 2024.]
- [11] S. Russel, P. Norvig, *Artificial Intelligence: A Modern Approach*, Third Edition, Pearson, 2010.
- [12] M. Žugaj, M. Schatten, Otvorena ontologija organizacijske arhitekture u funkciji upravljanja znanjem, *Ekonomski vjesnik br. 1 i 2 (20)*: 39-46, 2007.
- [13] W. Stallings, *Cryptography and Network Security Principles and Practice*, Seventh Edition, Pearson, 2017.
- [14] M. Pavlić, *Informacijski sustavi*, Školska Knjiga, Zagreb, 2011.
- [15] Zendesk, Knowledge base software [online], Zendesk, San Francisco, 2024, dostupno na: <https://www.zendesk.com/service/help-center/knowledge-base-software/> [13. 9. 2024.]

- [16] Helpjuice, Powerful and Simple is how I'd describe it [online], Helpjuice, New and Simple is how I'd describe it [online], Helpjuice, 2024, dostupno na: <https://helpjuice.com/features> [13. 9. 2024.]
- [17] Slite, Build a knowledge home that evolves with you [online], Slite, 2024, dostupno na: https://slite.com/solutions/knowledge-base?_gl=1*tzui89*_up*MQ..*_ga*NzE2OTMzNzUzLjE3MjYyNDg1MDE.*_ga_8741Z93SL4*MTcyNjI0ODUwMC4xLjAuMTcyNjI0ODUwMC4wLjAuOTk5NjI0Nzk1 [13. 9. 2024.]
- [18] Document 360, Build an AI-Powered knowledge base that delivers instant help to your customer [online], Document360, 2024, dostupno na: <https://document360.com/solutions/knowledge-base/> [13. 9. 2024.]
- [19] Notion, The wiki that redefines 'wiki' [online], Notion, 2024, dostupno na: <https://www.notion.so/product/wikis#features> [13. 9. 2024.]

SAŽETAK

U ovom je radu opisan način rada suvremenih *chatbotova*, analizirajući svaki njihov temeljni segment, s posebnim naglaskom na bazu znanja te prateći njihov povijesni napredak. Glavni zadatak ovoga rada bila je implementacija web aplikacije za organizaciju baze znanja za *chatbot* u okruženju *Ruby on Rails*. Implementacija je napravljena na konceptu baze znanja sveučilišta, koja podržava učitavanje dokumenata s uređaja u različitim tekstualnim formatima ili izravni unos sadržaja dokumenata kroz sučelje. Za organizaciju baze znanja korištena je hijerarhijska struktura, pa su dokumenti podijeljeni u mape i podmape. Uz stvaranje i organizaciju dokumenata, podržano je i njihovo pretraživanje uz mogućnost korištenja različitih filtera te uređivanje i brisanje dokumenata. Korisnici imaju različite uloge te je ovisno o njima definirano pravo pristupa određenom dijelu aplikacije.

Ključne riječi: baza znanja, chatbot, Ruby on Rails, web aplikacija

ABSTRACT

Web application for the organization of the knowledge base for chatbot

This thesis describes how modern chatbots work, analyzing each of their fundamental segments, with special emphasis on the knowledge base and following their historical progress. The main task of this thesis was the implementation of a web application for the organization of the knowledge base for a chatbot in the Ruby on Rails environment. The implementation is based on the concept of the university's knowledge base, which supports the loading of documents from the device in various text formats or the direct input of document content through the interface. A hierarchical structure was used to organize the knowledge base, so the documents were divided into folders and subfolders. In addition to the creation and organization of documents, searching is also supported with the possibility of using different filters, as well as editing and deleting documents. Users have different roles and the right to access a certain part of the application is defined depending on them.

Keywords: knowledge base, chatbot, Ruby on Rails, web application