



Srednja šola za kemijo, elektrotehniko in računalništvo

Pot na Lavo 22

3000 Celje

PRILAGOJENO UČENJE

raziskovalna naloga

Avtorji: Filip Juršinič, Maj Krajnc, Jure Ločnikar

Mentor: Andraž Pušnik

Mestna občina Celje, Mladi za Celje

Celje, 2024

IZJAVA*

Mentor Andraž Pušnik v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom Prilagojeno učenje, katere avtorji so Filip Juršinič, Maj Krajnc in Jure Ločnikar:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 5. 4. 2024



Podpis mentorja

Podpis odgovorne osebe

ANITA
LAZNIK

Digitalno podpisal
ANITA LAZNIK
Datum:
2024.04.05
20:19:20 +02'00'

*

POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja fotografskega gradiva, katerega ni avtor raziskovalne naloge, pa hrani šola v svojem arhivu.

»Tehnologija je najboljša,

ko poveže ljudi skupaj.«

(Matt Mullenweg)

Zahvala

Za strokovno pomoč, nasvete, predloge in vso podporo se iskreno zahvaljujemo našemu mentorju gospodu Andražu Pušniku, ki nas je zanesljivo vodil in usmerjal od samega začetka, s čimer je ključno prispeval k uspehu naše raziskovalne naloge.

Povzetek

Raziskovalna naloga se osredotoča na razvoj AI asistenta za izboljšanje učnih procesov. Temelji na uporabi ChatGPT API-ja kot modela umetne inteligence za generiranje zapiskov, povzetkov in vaj. Za dostop do teh funkcionalnosti smo razvili spletno stran, ki omogoča nalaganje dokumentov. Ti dokumenti so nato preverjeni in analizirani s strani API-ja, kar uporabnikom omogoča izboljšanje učnega procesa z avtomatiziranimi povzetki in vajami.

Ključne besede: AI, ChatGPT, PHP, Visual Studio Code, zapiski, učenje, API

Abstract

The research task focuses on the development of an AI assistant to enhance learning processes. It is based on using the ChatGPT API as an artificial intelligence model for generating notes, summaries, and exercises. To access these functionalities, we have developed a website that allows the uploading of documents. These documents are then verified and analyzed by the API, enabling users to improve the learning process with automated summaries and exercises.

Keywords: AI, ChatGPT, PHP, Visual Studio Code, notes, studying, API

Kazalo vsebine

1 Uvod	11
2 Raziskava	12
2.1 Namen raziskovalne naloge	12
2.2 Cilji raziskovalne naloge	12
2.1 Hipoteze	13
3 ChatGPT	13
3.1 GPT 3	13
3.1.1 GPT 3.5-turbo	15
3.1.1.1 Razlaga izbire modela GPT-3.5-turbo	15
3.2 GPT 4	19
4 ChatGPT API	20
4.1 GPT Moderation API	21
4.2 GPT embedding models	22
4.2.1 Osnovne značilnosti GPT embedding modelov	22
5 Visual Studio Code	23
5.1 Osnove	23
5.2 Uporaba	24
6 PHP	25
6.1 Osnove	25
6.2 Spremenljivke	26
6.2.1 Konstante	26
6.2.2 Spremenljivke	27
6.3 If stavek	27
6.4 If else stavek	28
6.5 Switch case	29
6.6 Zanke	30
6.7 Funkcije	30
7 HTML	30
7.1 Osnove HTML	31
7.2 Struktura HTML dokumenta	31

8 CSS.....	32
8.1 Glavne značilnosti in prednosti CSS vključujejo:	32
8.1.1 Ločitev vsebine od oblike.....	32
8.1.2 Kaskadna pravila	33
8.1.3 Odzivno spletno oblikovanje	33
8.1.4 Ponovna uporaba in vzdrževanje.....	33
8.1.5 Napredni vizualni učinki.....	34
9 XAAMP	34
9.1 Komponente	35
9.1.1 Apache HTTP Server.....	36
9.1.2 MariaDB	36
9.1.3 PHP	37
9.1.4 Perl	38
9.2 Lastnosti	39
9.2.1 Cross-Platform Podpora	39
9.2.2 Enostavna namestitve in uporaba	40
9.2.3 Dodatna sledeča orodja	40
9.2.4 Namen in uporaba	41
10 Prilagojeno učenje	41
10.1 Tehnološko ozadje	41
10.1.1 Integracija API v spletno stran	41
10.1.2 Prednosti uporabe umetne inteligence pri generiranju zapiskov	41
10.2 Oblikovanje in funkcionalnosti spletne strani.....	42
10.2.1 Registracija in prijava uporabnikov	42
10.2.2 Vmesnik za nalaganje datotek	45
10.2.3 Implementacija kode za pretvorbo datotek v besedilni format.....	45
10.3 Proces generiranja zapiskov	46
10.3.1 Nastavitve generiranja zapiskov	46
10.3.2 Raziskovanje različnih vrst zapiskov	47
10.3.3 Uporaba API za ustvarjanje personaliziranih zapiskov	47
10.4 Uporabniška izkušnja	48

10.4.1 Pregled glavne spletne strani	48
10.4.2 Vmesnik za upravljanje nastavitev generiranja	48
10.4.3 Možnosti shranjevanja in kopiranja generiranih zapiskov	48
10.5 Evalvacija rezultatov.....	49
10.5.1 Merjenje učinkovitosti generiranja zapiskov.....	49
10.5.2 Primerjava različnih nastavitev in njihov vpliv na kakovost zapiskov	49
10.6 Prilagojeno učenje v praksi	49
10.6.1 Pogovor s profesorji.....	49
10.6.2 Testiranje dijakov.....	50
11 Zaključek	51
11.1 Povzetek doseženih ciljev	51
11.2 Razprava o rezultatih in njihov pomen	51
11.3 Prihodnje možnosti raziskovanja in nadgradnje sistema	51
11.3.1 Dodatne funkcije.....	52
12 Ovrednotenje hipotez.....	52
13 Viri in literature.....	54

Kazalo slik

Slika 1: ChatGPT 3.5 verzije	17
Slika 2: Ustvarjanje API ključa	18
Slika 3: Uporaba API ključa	18
Slika 4: Primer konstant.....	26
Slika 5: Primer spremenljivke	27
Slika 6: Primer if stavka.....	27
Slika 7: Primer if else stavka	28
Slika 8: Primer switch case stavka	29
Slika 9: Struktura html	31
Slika 10: XAMPP uporabniški vmesnik.....	34
Slika 11: XAMPP komponente	36
Slika 12: PHP v login.php	38
Slika 13: Nastavitve za podatkovno bazo	38
Slika 14: Obrazec za registracijo	43
Slika 15: Obrazec za vpis.....	43
Slika 16: Zgodovina	44
Slika 17: Uporabnikov profil	44
Slika 18: Glavna stran	45
Slika 19: Generirano besedilo iz zapiskov.....	46

Kazalo tabel

Tabela 1: Primerjava ocen	50
---------------------------------	----

1 Uvod

Učenje je lahko zelo neprijeten in dolgotrajen proces, zato ni nenavadno, da imajo dijaki pogosto probleme – urejanje zapiskov traja predolgo, večina dijakov pa se lažje uči, če imajo postavljena vprašanja iz snovi, ki pa jih profesorji navadno ne dajo.

Ker pa je trenutno ChatGPT močno v uporabi, smo prišli do ideje, da bi ga dijaki namesto za goljufanje pri ocenjevanjih lahko uporabili za pomoč pri učenju. Naredili smo spletno stran, ki vključuje ChatGPT API, v katero se mora uporabnik prijaviti z uporabniškim imenom in geslom, ki sta shranjena v podatkovno bazo, povezano s spletno stranjo. Ob prvem obisku strani se mora uporabnik registrirati. Naša spletna aplikacija je namenjena izključno uporabi za šolske namene - uporabnik se ne more direktno pogovarjati z ChatGPT-jem, lahko le naloži željene datoteke na spletno stran, ta pa jih preda umetni inteligenci. Uporabnik ima na voljo več checkbox-ov oziroma opcij, kaj naj ChatGPT API naredi z podanimi datotekami.

Pri izdelavi smo bili primorani uporabljati programske jezike, s katerimi se v šoli še nismo srečali, zato smo imeli pri programiranju kar nekaj težav. Delo je bilo zamudno in je potekalo počasneje in dlje, kot smo predvidevali. Čeprav smo vedeli, da bomo morali uporabiti nam takrat še neznane programske jezike, smo se naloge lotili zagnano in zavzeto.

V sodobnem izobraževalnem okolju se srečujemo z obilico informacij, ki jih je potrebno obvladati v relativno kratkem času. Avtomatsko generiranje zapiskov s pomočjo ChatGPT API-ja ponuja inovativno rešitev, ki olajša proces obdelave in razumevanja gradiv. Raziskujemo, kako lahko ta tehnologija prispeva k izboljšanju produktivnosti, zmanjšanju časa potrebnega za pripravo zapiskov ter povečanju kakovosti študijske izkušnje.

2 Raziskava

2.1 Namen raziskovalne naloge

Glavni namen raziskovalne naloge je razviti uporabniku prijazno platformo, ki omogoča preprosto uporabo ChatGPT API-ja za avtomatsko generiranje zapiskov iz raznolikih dokumentov. Želimo raziskati, kako učinkovita je ta tehnologija v izobraževalnem okolju in kakšen vpliv ima na proces učenja ter kvaliteto pridobljenega znanja.

2.2 Cilji raziskovalne naloge

Ustvarili smo spletišče, ki omogoča uporabnikom, to kar smo želeli. Uporabniki lahko naložijo datoteke in izberejo parametre, da dobijo personalizirane zapiske, ki jih lahko uporabljajo za pomoč pri učenju in pridobivanje novega znanja.

Cilji v teoretičnem delu:

- Predstaviti programske jezike, ki smo jih uporabili;
- predstaviti API;
- predstaviti zgradbo in delovanje asistenta;

Cilji v praktičnem delu:

- Narediti delujočega asistenta, ki pospeši učenje;
- uporabiti različne programske jezike da se čim več naučimo;
- uspešno povezati spletno stran in API;

- narediti prijetno uporabniško izkušnjo pri uporabi;

2.1 Hipoteze

1. **hipoteza:** Naše orodje izdeluje izčrpne zapiske, ki vsebujejo vse potrebne informacije.
2. **hipoteza:** Asistent bo pospešil učenje za vsaj 20%.
3. **hipoteza:** Zaradi pomankanja predznanja nam bo API predstavljal težave.
4. **hipoteza:** Asistent bo dijakom pomagal dobiti višje ocene.

3 ChatGPT

3.1 GPT 3

GPT-3, razvit s strani OpenAI in izdan leta 2020, je napreden model velikega jezika, znan po svoji izjemni sposobnosti ustvarjanja človeku podobnega besedila na podlagi prejetih vhodnih podatkov. Uporablja globoko nevronske mreže s transformatorsko arhitekturo, ki izkorišča mehanizem "pozornosti", ki modelu omogoča, da se selektivno osredotoči na relevantne dele vhodnega besedila. Ta model je opremljen s 175 milijardami parametrov in zahteva 350 GB prostora za shranjevanje, kar priča o njegovi velikosti in zmogljivosti za kompleksne jezikovne naloge. Sposobnosti GPT-3 vključujejo močne sposobnosti učenja "iz nič" in "z nekaj primeri" na številnih nalogah, kar mu omogoča, da se dobro odreže z minimalnimi ali brez specifičnih podatkov za nalogo.

Vsestranskost GPT-3 sega v širok spekter aplikacij, od generiranja kode in diagnosticiranja medicinskih stanj do ustvarjanja vsebin, kot so glasba in celo sodelovanje pri komunikacijskih nalogah. Njegova sposobnost prevajanja besedila v programerske ukaze in obratno, izvajanje analize sentimenta in generiranje besedila v različnih domenah ga naredi močno orodje za razvijalce, raziskovalce in kreativce.

Raziskave so pokazale, da se zmogljivost GPT-3 znatno izboljša z velikostjo modela, kar kaže na povečano učinkovitost pri uporabi informacij iz konteksta za učenje nalog. Pristop treniranja modela je vključeval ocenjevanje pod tremi pogoji: "učenje z nekaj primeri", kjer modelu damo več primerov; "učenje z enim primerom", z le enim primerom; in "učenje iz nič", brez primerov, le z navodili v naravnem jeziku. Te metode so pokazale, da GPT-3 lahko tekmuje z ali celo presega modele na vrhu lestvice v nekaterih primerih, brez potrebe po podatkih specifičnih za nalogo.

Kljub svojim zmogljivostim ima GPT-3 omejitve in tveganja, vključno s svojo naravo predhodnega treniranja, ki ne omogoča nadaljnjega učenja iz novih interakcij, maksimalno velikostjo vhoda zaradi transformatorske arhitekture in potencialnimi pristranskostmi, naučenimi iz njegovih učnih podatkov. Skrbi glede posnemanja in točnosti generiranega vsebina so bile prav tako izražene, kar poudarja potrebo po previdni uporabi in nadaljnjih raziskavah za omilitev teh tveganj.

Razvoj in izdaja GPT-3 s strani OpenAI, sprva kot raziskovalni projekt, predstavljata pomemben mejnik v AI in NLP, ki cilja na koristi z človeštvo z razvojem "prijazne AI". Njegova velikost in zmogljivost sta postavili nove standarde na področju, pri čemer si je Microsoft zagotovil ekskluzivne pravice do licenciranja modela, kar kaže na potencialen vpliv.

3.1.1 GPT 3.5-turbo

GPT-3.5-turbo je nadgradnja modela GPT-3.5, ki ponuja izboljšano natančnost in zmanjšane stroške. Zdaj je na voljo za fino nastavitvev, kar razvijalcem omogoča prilagajanje modela za boljše delovanje v specifičnih primerih. Fino nastavitvev omogoča izboljšave, kot so boljša vodljivost, zanesljivejše oblikovanje izhoda in prilagajanje tona izhodnih besedil, da bolje ustrezajo blagovni znamki podjetja. GPT-3.5-turbo lahko zdaj obdeluje do štiri tisoč žetonov, kar je dvakrat več kot prejšnji fino nastavljeni modeli, kar omogoča uporabnikom, da skrajšajo svoje pozive do 90% z vključitvijo navodil neposredno v model, s čimer pospešijo vsak klic API in zmanjšajo stroške.

Poleg tega, GPT-3.5-turbo uvaja nove funkcije, kot je klicanje funkcij, ki razvijalcem omogoča opis funkcij modelu in omogoča, da model izbere inteligentno, kako oddati JSON objekt, ki vsebuje argumente za klic teh funkcij. To omogoča bolj zanesljivo povezavo zmogljivosti GPT-ja z zunanjimi orodji in API-ji.

GPT-3.5-turbo je na voljo v Azure Machine Learning kot del nove zbirke modelov OpenAI, ki omogoča fino nastavitvev. Ta model, ki je optimiziran predvsem za pogovorne aplikacije, je najbolj zmogljiv in stroškovno učinkovit model v družini GPT-3.5. Podpira do 4,097 vhodnih žetonov ob času izvajanja z učnimi podatki, ki so na voljo do septembra 2021. Znanje, ki ga model GPT-3.5 vključuje, ostaja aktualno do septembra 2021.

3.1.1.1 Razlaga izbire modela GPT-3.5-turbo

OpenAI je s predstavitvijo GPT-3.5-turbo naredil pomemben korak naprej v razvoju jezikovnih modelov. Ta nadgradnja prinaša izboljšano natančnost in optimizacijo

stroškov, zaradi česar je še privlačnejša za širok spekter aplikacij. Odločitev za razvoj in uporabo GPT-3.5-turbo temelji na več ključnih prednostih:

Izboljšana natančnost in zmanjšani stroški: GPT-3.5-turbo ponuja boljšo natančnost pri generiranju besedil, hkrati pa znižuje operativne stroške, kar je ključno za podjetja in razvijalce, ki se zanašajo na visoko zmogljive jezikovne modele za svoje aplikacije.

Možnost fine nastavitve: Fina nastavitve omogoča uporabnikom, da prilagodijo model specifičnim potrebam svojih projektov. Ta prilagodljivost je bistvenega pomena za doseganje bolj relevantnih in ciljno usmerjenih rezultatov, izboljšanje vodljivosti in prilagajanje tona izhodnih besedil.

Podpora za daljše kontekste: Z zmogljivostjo obdelave do 4k žetonov, GPT-3.5-turbo omogoča obdelavo daljših kontekstov, kar znatno izboljša razumevanje in generiranje kompleksnejših besedil. To je še posebej koristno za naloge, ki zahtevajo poglobljeno analizo.

Optimizacija za hitrejša in cenejša klice API: Zmožnost skrajšanja pozivov do 90% s vključitvijo navodil neposredno v model zmanjšuje čas, potreben za vsak klic API, in optimizira stroške, kar je še posebej pomembno za aplikacije z velikim obsegom zahtev.

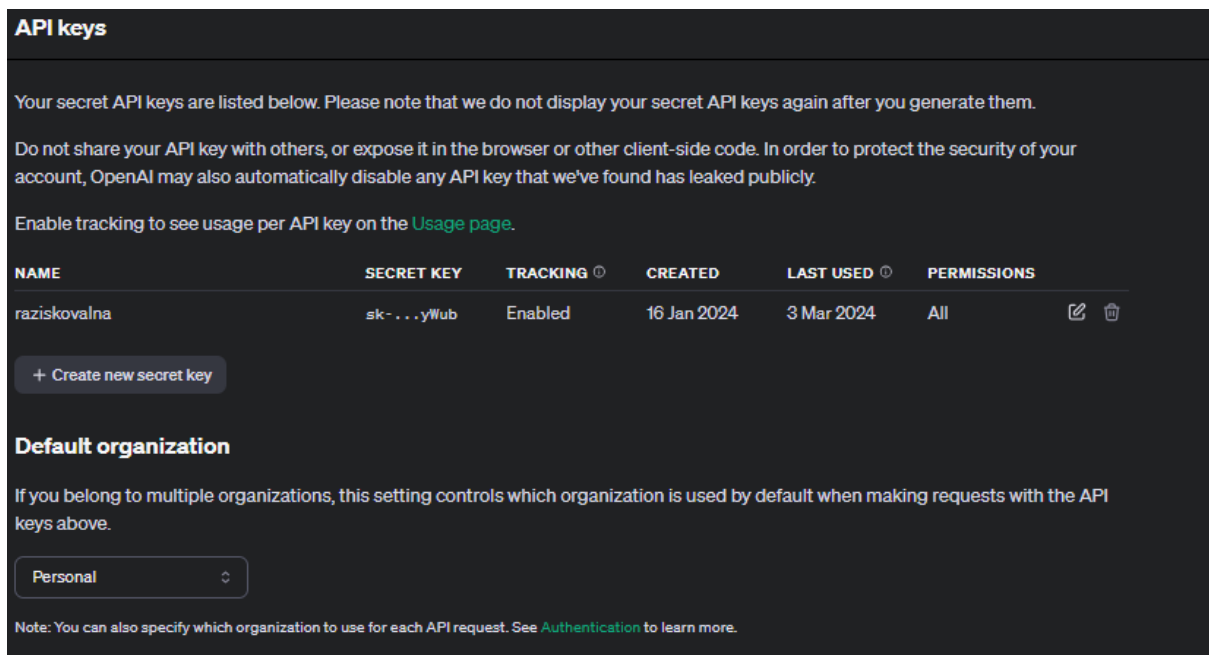
Inovativne funkcije za interakcijo z zunanjimi orodji: GPT-3.5-turbo uvaja možnost klicanja funkcij, ki uporabnikom omogoča, da modelu opišejo funkcije in kako naj model odda JSON objekt z argumenti za te funkcije. To odpira nove možnosti za integracijo GPT-ja z različnimi zunanjimi orodji in API-ji, kar širi uporabnost modela v realnih aplikacijah.

Za naš projekt je bil ta model ChatGPT API-ja najboljša izbira. Glede na to, da za obdelavo datoteke ne uporabljamo API-ja, je bila ta odločitev že kar hitro odločena. API potrebujemo samo za spreminjanje velike količine besedila v personalizirane zapiske.

MODEL	DESCRIPTION	CONTEXT WINDOW	TRAINING DATA
gpt-3.5-turbo-0125	New Updated GPT 3.5 Turbo The latest GPT-3.5 Turbo model with higher accuracy at responding in requested formats and a fix for a bug which caused a text encoding issue for non-English language function calls. Returns a maximum of 4,096 output tokens. Learn more.	16,385 tokens	Up to Sep 2021
gpt-3.5-turbo	Currently points to gpt-3.5-turbo-0125.	16,385 tokens	Up to Sep 2021
gpt-3.5-turbo-1106	GPT-3.5 Turbo model with improved instruction following, JSON mode, reproducible outputs, parallel function calling, and more. Returns a maximum of 4,096 output tokens. Learn more.	16,385 tokens	Up to Sep 2021
gpt-3.5-turbo-instruct	Similar capabilities as GPT-3 era models. Compatible with legacy Completions endpoint and not Chat Completions.	4,096 tokens	Up to Sep 2021
gpt-3.5-turbo-16k	Legacy Currently points to gpt-3.5-turbo-16k-0613.	16,385 tokens	Up to Sep 2021
gpt-3.5-turbo-0613	Legacy Snapshot of gpt-3.5-turbo from June 13th 2023. Will be deprecated on June 13, 2024.	4,096 tokens	Up to Sep 2021
gpt-3.5-turbo-16k-0613	Legacy Snapshot of gpt-3.5-16k-turbo from June 13th 2023. Will be deprecated on June 13, 2024.	16,385 tokens	Up to Sep 2021

Slika 1: ChatGPT 3.5 verzije

Slika prikazuje vse različne verzije ChatGPT 3.5. Podani so tudi kratki opisi, uporabo žetonov (to je valuta, ki jo ChatGPT uporablja za računanje, koliko uporabljaš API), podatki za treniranje umetne inteligence in ime modela.



Slika 2: Ustvarjanje API ključa

Na sliki 2 je prikazana je prikazana stran, kjer smo ustvarili naš API ključ. Proces je zelo enostaven in razumljiv. Stran ima nekaj nastavitev, s katerimi lahko personaliziramo API ključ, odvisno za kakšen namen ga bomo uporabljali.

```
function processTxtFileAndSendToOpenAI($filePath) {
    $textContent = file_get_contents($filePath);
    if ($textContent === false) {
        return "Error reading the .txt file.";
    }

    $openaiUrl = 'https://api.openai.com/v1/completions';
    $apiKey = 'sk-...lkHfFsBsuHyWub';

    $postData = json_encode([
        "model" => "gpt-3.5-turbo",
        "prompt" => $textContent,
        "temperature" => 0.7,
        "max_tokens" => 150,
    ]);
}
```

Slika 3: Uporaba API ključa

Slika 3 prikazuje del kode, ki je del »processFile.php« in iz tekstovnih datotek pretvori tekst v prilagojene zapiske. Prikazana je uporaba API ključa in nastavljanja nastavitev za izbran model.

3.2 GPT 4

GPT-4 je najnaprednejša različica OpenAI-jevih generativnih predhodno usposobljenih transformatorjev (GPT), ki prinaša znatne izboljšave v primerjavi s prejšnjimi modeli, kot sta GPT-3 in GPT-3.5. Zasnovan je bil z namenom zagotavljanja varnejših in bolj uporabnih odgovorov, hkrati pa presega zmogljivosti ChatGPT v naprednih razmišljujočih sposobnostih.

Ena od ključnih značilnosti GPT-4 je njegova več modalna zmožnost, ki omogoča modelu, da sprejema tako besedilne kot slikovne vhode, kar uporabnikom omogoča, da modelu zastavljajo vprašanja ali naloge, ki vključujejo kombinacijo besedila in slik. Ta zmožnost povečuje prilagodljivost in uporabnost GPT-4 v različnih kontekstih, od analize dokumentov do kreativnega pisanja in celo do reševanja problemov, ki zahtevajo razumevanje vizualnih informacij.

Poleg tega GPT-4 zmore obdelati več kot 25.000 besed besedila, kar omogoča dolgotrajne interakcije, ustvarjanje dolgih oblik vsebine in podrobno iskanje in analizo dokumentov. Ta zmogljivost je še posebej koristna za scenarije uporabe, ki zahtevajo obsežno obravnavo tem ali podrobnosti, kot so akademske raziskave, pisanje scenarijev ali obsežne tehnološke dokumentacije.

GPT-4 prav tako izstopa v svojih naprednih sposobnostih razmišljanja in reševanja problemov, presegajoč ChatGPT v več testnih okoljih in akademskih testih, kar dokazuje njegovo sposobnost, da deluje na človeški ravni v različnih strokovnih in akademskih nalogah. Zmožnost modela, da zagotovi natančne in ustrezne odgovore,

je bila izboljšana z obsežno uporabo povratnih informacij človeških uporabnikov in sodelovanjem z več kot 50 strokovnjaki na področju varnosti in zanesljivosti AI, kar je GPT-4 naredilo varnejšega in bolj usklajenega z etičnimi standardi.

Poleg teh izboljšav je OpenAI nadaljeval z razvojem funkcij, ki omogočajo razvijalcem in končnim uporabnikom večjo kontrolo in prilagodljivost nad obnašanjem AI, vključno z možnostjo natančnejšega usmerjanja odzivov AI prek sistemskih sporočil. To omogoča uporabnikom, da bolje določijo slog in nalogo AI, kar povečuje uporabnost GPT-4 v specifičnih uporabniških scenarijih.

4 ChatGPT API

ChatGPT API je orodje, ki razvijalcem omogoča integracijo zmogljivosti modela ChatGPT v lastne aplikacije, produkte ali storitve. Ena od ključnih značilnosti je, da modeli, kot je gpt-3.5-turbo, ki se uporablja v ChatGPT, zdaj omogočajo bolj stroškovno učinkovito uporabo z nižjo ceno na tisoč žetonov. To je desetkrat cenejše od prejšnjih modelov GPT-3.5. Modeli ChatGPT sprejemajo zaporedje sporočil skupaj z metapodatki, kar razvijalcem omogoča večjo prilagodljivost pri oblikovanju pogovorov in izboljšuje interakcijo z uporabniki.

OpenAI nenehno izboljšuje modele ChatGPT, kar vključuje tudi nadgradnje, ki so na voljo razvijalcem. To pomeni, da bodo razvijalci, ki uporabljajo model gpt-3.5-turbo, vedno imeli dostop do priporočenega stabilnega modela, hkrati pa lahko izbirajo med specifičnimi verzijami modelov. Ta pristop omogoča bolj prilagojeno uporabo API-ja glede na potrebe posameznih projektov.

Za dostop do ChatGPT API-ja se je potrebno registrirati na OpenAI platformi in pridobiti API ključ. API ima svoje cenovne modele, ki so ločeni od naročnine ChatGPT Plus in omogočajo uporabo ChatGPT v različnih aplikacijah. Pomembno je, da OpenAI od 1. marca 2023 ne uporablja več podatkov, poslanih preko API-ja, za izboljšanje svojih modelov, razen če se organizacija za to izrecno odloči. Prav tako je bila uvedena privzeta politika zadrževanja podatkov za 30 dni za uporabnike API-ja, s možnostjo strožjih pravil zadrževanja glede na potrebe uporabnikov.

V raziskavi smo prišli do nekaj težav povezanih z API-jem. Težave smo kar hitro lahko odpravili, saj je na internetu že ogromno dokumentacije glede ChatGPT API-ja.

4.1 GPT Moderation API

Moderation API analizira tekst, ki ga generira GPT-4, da prepozna morebitne neprimernosti, kot so nasilna vsebina, sovražni govor, spolna vsebina in druge vrste vsebin, ki bi lahko bile škodljive ali nezaželeni. Ko je neprimerna vsebina identificirana, lahko razvijalci ustrezno ukrepajo, na primer z blokiranjem, filtriranjem ali opozarjanjem uporabnikov.

Pomembno je poudariti, da Moderation API ni popoln in ne more zagotoviti 100% natančnosti pri identifikaciji neprimernih vsebin. Zato je priporočljivo, da se uporablja v kombinaciji z drugimi metodami moderiranja in nadzora vsebin.

OpenAI redno posodablja in izboljšuje svoje modele in orodja za moderiranje, da bi izboljšali njihovo učinkovitost in zmanjšali tveganje za generiranje neprimernih vsebin.

4.2 GPT embedding models

GPT (Generative Pretrained Transformer) embedding modeli so pomemben razvoj v področju strojnega učenja in obdelave naravnega jezika (NLP), ki temeljijo na ideji transformatorskih modelov. GPT modeli so vrsta modelov globokih nevronske mreže, zasnovanih za generiranje besedila in razumevanje naravnega jezika. Uporabljajo mehanizem pozornosti, ki omogoča modelu, da učinkovito obdeluje dolge sekvence besedila in razume kompleksne jezikovne vzorce.

4.2.1 Osnovne značilnosti GPT embedding modelov

Pretrained modeli: GPT modeli so predhodno usposobljeni na ogromnih korpusih besedila, kar jim omogoča splošno razumevanje jezika in konteksta. Usposabljanje poteka z nalogo napovedovanja naslednje besede v besedilu, kar pripomore k razvoju bogatega notranjega razumevanja jezika.

Fine-tuning: Po predhodnem usposabljanju lahko modele dodatno fino nastavimo za specifične aplikacije, kot so razumevanje besedila, prevajanje, povzemanje, generiranje besedila in še več. To omogoča, da se modeli prilagodijo konkretnim potrebam in domenam.

Velikosti modelov: GPT modeli prihajajo v različnih velikostih, od manjših variant, ki so lažje za uporabo in hitrejše, do izjemno velikih modelov, kot je GPT-3, ki ponujajo izboljšano razumevanje in generacijske zmogljivosti, a zahtevajo več virov za delovanje.

Uporabe: GPT modele lahko uporabljamo za širok spekter aplikacij vključno z avtomatskim odgovarjanjem na vprašanja, ustvarjanjem besedil, avtomatizacijo

dialoga, prevajanjem jezikov, in še mnogo več. So izredno vsestranski in najdejo uporabo v različnih industrijskih in raziskovalnih kontekstih.

Razumevanje konteksta: Eden od ključnih dosežkov GPT modelov je njihova sposobnost razumevanja in uporabe konteksta. Modeli so sposobni slediti dolgim in zapletenim razpravam ter ustvarjati koherentne in relevantne odgovore.

GPT embedding modeli so pomembno napredovali v razumevanju naravnega jezika in generiranju besedila, ki temelji na tem razumevanju. Z vsako novo iteracijo modelov, kot je GPT-4, opazamo izboljšave v njihovi zmožnosti obdelave jezika, kar odpira nove možnosti za aplikacije strojnega učenja in umetne inteligence.

5 Visual Studio Code

5.1 Osnove

Visual Studio Code je brezplačno, odprtokodno integrirano razvojno okolje (IDE) razvito s strani Microsofta. Prvič je bil izdan leta 2015 in je na voljo za Windows, Linux in macOS. Visual Studio Code je izjemno prilagodljiv in podpira različne programske jezike, kot so JavaScript, TypeScript, C++, C#, Java, Python, PHP in še mnoge druge, preko razširitev, ki jih lahko namestite iz vgrajenega tržišča razširitev.

Ponuja številne funkcije, ki olajšajo razvoj programske opreme, vključno z vgrajenim urejevalnikom kode, podporo za razhroščevanje, integracijo z Git-om za upravljanje različic kode, označevanje sintakse, inteligentno dopolnjevanje kode (IntelliSense), refaktoriranje kode in pregledovanje kode v realnem času.

IDE je zasnovan tako, da je lahek in hiter, obenem pa omogoča razvijalcem, da prilagodijo svoje delovno okolje s temami, nastavitvami in razširitvami, da izboljšajo svojo produktivnost. Prav tako podpira razvoj kontejnerskih aplikacij z Docker-jem in omogoča razvijalcem, da delajo na oddaljenih projektih preko SSH.

Za razvoj spletnih in oblakovnih aplikacij, Visual Studio Code ponuja integracijo z različnimi oblakovnimi storitvami, kot so Azure in AWS, ter orodja za razvoj in testiranje REST API-jev. Poleg tega omogoča razvijalcem, da ustvarijo in upravljajo Kubernetes konfiguracije neposredno znotraj IDE.

V skupnosti Visual Studio Code je veliko aktivnih uporabnikov in razvijalcev, ki prispevajo k bogatemu naboru razširitev in orodij, zaradi česar je Visual Studio Code eno izmed najbolj priljubljenih orodij med razvijalci za razvoj različnih vrst aplikacij za različne platforme.

5.2 Uporaba

Za Visual Studio Code smo se odločili, zaradi predznanja, ki ga že imamo in enostavnosti uporabe. Podpira vse programske jezike, ki smo jih uporabili. V pomoč nam je bila razširitev Blackbox – vsebuje umetno inteligenco, ki pomaga pri pisanju kode. Všeč nam je, da je koda pri pisanju pregledna in da lahko s pomočjo GitHub-a, vsi udeleženi na projektu delamo na istih datotekah, brez potrebe po ročnem prenašanju datotek med računalniki.

6 PHP

6.1 Osnove

PHP je skriptni jezik, ki se pogosto uporablja za razvoj spletnih aplikacij. Razvil ga je Rasmus Lerdorf leta 1995, in sicer kot orodje za vzdrževanje osebne spletne strani. Od takrat je PHP hitro pridobil popularnost med spletnimi razvijalci zaradi svoje enostavnosti, učinkovitosti in široke podpore na spletnih strežnikih po vsem svetu. PHP je interpretiran jezik, kar pomeni, da se koda izvaja v realnem času, ko spletni strežnik prejme zahtevo, kar omogoča hitro izdelavo dinamičnih spletnih strani.

Glavni cilj pri razvoju PHP je bil olajšati izdelavo dinamičnih spletnih strani. PHP je zasnovan tako, da omogoča enostavno vključevanje skriptov v HTML kodo, kar razvijalcem omogoča, da hitro izdelujejo interaktivne spletne strani. PHP podpira številne baze podatkov, kot so MySQL, PostgreSQL in MongoDB, kar razvijalcem omogoča, da gradijo kompleksne spletne aplikacije z obsežnimi podatkovnimi shrambami.

PHP je zelo prilagodljiv jezik, ki omogoča razvijalcem, da izdelujejo spletna mesta in aplikacije, ki so prilagojene specifičnim potrebam in zahtevam. Prav tako podpira številne knjižnice in okvire, kot so Laravel, Symfony in CodeIgniter, ki še dodatno poenostavijo in pospešijo razvojni proces.

Varnost je ključni vidik pri razvoju spletnih aplikacij, in PHP nudi različne mehanizme za zagotavljanje varnosti aplikacij, kot so filtriranje vhodnih podatkov, šifriranje podatkov in zaščita pred napadi, kot je SQL injekcija. To razvijalcem omogoča, da izdelujejo varne in zanesljive spletne aplikacije.

6.2 Spremenljivke

V PHP-ju se za ustvarjanje spremenljivk uporablja simbol ameriškega dolarja (\$), pred imenom spremenljivke. PHP loči med spremenljivkami in konstantami. Konstante ne morajo spreminjati tipa podatkov, ko je enkrat določen prav tako se v njih ne more spreminjati podatkov, ko so enkrat vneseni. Večinoma se konstante uporabljajo samo za vnos števil v njih, medtem ko se spremenljivke uporabljajo za vnos črk in simbolov. Prav tako pa moramo pred spremenljivkami uporabiti (\$), ki ga pred konstantami ne potrebujemo.

6.2.1 Konstante

Za ustvarjanje nove konstante uporabljamo funkciji »define()« ali ključno besedo »const«. Konstanta je vrednost, ki se, ko je enkrat določena, ne more spremeniti tekom izvajanja skripte. Konstante so globalno dostopne in ne potrebujejo simbola ameriškega dolarja (\$) pred imenom. Konstantam je treba vrednost dodeliti ob njihovi deklaraciji. Na spodnji sliki lahko vidimo tudi dva primera konstant iz naše kode.

```
const openai = new OpenAI();  
  
const rl = readline.createInterface({  
  input: process.stdin,  
  output: process.stdout,  
});
```

Slika 4: Primer konstant

6.2.2 Spremenljivke

Za ustvarjanje spremenljivke uporabimo simbol ameriškega dolarja (\$) in nato ime spremenljivke, ki jo želimo ustvariti. Spremenljivkam se lahko vrednost spreminja tekom delovanja skripte, poleg tega pa se lahko spreminja tudi podatkovni tip spremenljivke. Vrednosti jim ni treba dodeliti ob deklaraciji, saj jo lahko kasneje vnesemo ko skripta že deluje. Na spodnji sliki lahko vidimo tudi nekaj primerov spremenljivk iz naše kode.

```
$hostName = "localhost";  
$dbUser = "root";  
$dbPassword = "";  
$dbName = "data";
```

Slika 5: Primer spremenljivke

6.3 If stavek

If stavek se v računalniškem programiranju uporablja za zagon dela kode takrat, ko je izpolnjen določen pogoj. Na spodnji sliki je predstavljena sintaksa if stavka v PHP-ju.

```
if (isset($_POST["login"])) {  
    $email = $_POST["email"];  
    $password = $_POST["password"];  
    require_once "database.php";  
    $sql = "SELECT * FROM users WHERE email = '$email'";  
    $result = mysqli_query($conn, $sql);  
    $user = mysqli_fetch_array($result, MYSQLI_ASSOC);
```

Slika 6: Primer if stavka

V tem primeru smo uporabljali "if stavek", da smo preverili ali sta e-poštni naslov in geslo za prijavo pravilna. To preverimo tako, da v našo podatkovno bazo pogledamo, ali je tam shranjen takšen e-poštni naslov in če k njemu pripada geslo, ki je bilo vpisano. V primeru, da sta oba pogoja izpolnjena, nas spletna stran spusti naprej in lahko pričnemo z uporabo spletne strani.

6.4 If else stavek

If else stavek se v računalniškem programiranju uporablja za zagon dela kode takrat, ko določen pogoj v if stavku ni izpolnjen in se nato izvede koda, ki je napisana v else delu stavka. Na spodnji sliki je predstavljena sintaksa if in if-else stavka v PHP-ju.

```
if (isset($_POST["login"])) {  
    $email = $_POST["email"];  
    $password = $_POST["password"];  
    require_once "database.php";  
    $sql = "SELECT * FROM users WHERE email = '$email'";  
    $result = mysqli_query($conn, $sql);  
    $user = mysqli_fetch_array($result, MYSQLI_ASSOC);  
    if ($user) {  
        if (password_verify($password, $user["password"])) {  
            session_start();  
            $_SESSION["user"] = "yes";  
            header("Location: index.php");  
            die();  
        }else{  
            echo "<div class='alert alert-danger'>Password does not match</div>";  
        }  
    }else{  
        echo "<div class='alert alert-danger'>Email does not match</div>";  
    }  
}
```

Slika 7: Primer if else stavka

V tem primeru smo uporabljali "if-else" stavek, kjer smo v prvem koraku preverili, ali sta oba pogoja izpolnjena, nato pa v primeru, da eden izmed njih ni izpolnjen, se izpiše

napaka. Tukaj poznamo dve vrsti napak, prva je, da geslo ni pravilno, pri drugi, pa je podana e-pošta nepravilna.

6.5 Switch case

Switch case omogoča, da je izbran eden od več možnih blokov kode za izvajanje. Uporabljen je predvsem tam, kjer je možnih več pogojev. Zanke, kot so If, else if, else, se lahko uporabljajo na enak način, vendar je sintaksa switch case—a bistveno lažja za razumevanje in pisanje. Na spodnji sliki lahko tudi vidimo sintakso switch case v PHP.

```
switch ($ext) {  
    case 'txt':  
        $notes = processTxtFileAndSendToOpenAI($_FILES["fileToUpload"]["tmp_name"]);  
        break;  
    case 'docx':  
        $notes = processDocxFileAndSendToOpenAI($_FILES["fileToUpload"]["tmp_name"]);  
        break;  
    case 'pdf':  
        $notes = processPdfFileAndSendToOpenAI($_FILES["fileToUpload"]["tmp_name"]);  
        break;  
    case 'pptx':  
        $notes = processPptxFileAndSendToOpenAI($_FILES["fileToUpload"]["tmp_name"]);  
        break;  
}
```

Slika 8: Primer switch case stavka

Tu smo uporabili “switch case” stavek, kjer smo določili, kaj se zgodi, z različnimi vrstami datotek. V primeru, da uporabnik odda datoteko s končnico “.txt”, se uporabi “processTxtFileAndSendToOpenAI”. Ta proces datoteko odpre in jo posreduje našemu API-ju. Enako se zgodi za vse datoteke, le da je vrsta procesa odvisna od končnice, naprimer pri “.pdf” datoteki se uporabi proces “processPdfFileAndSendToOpenAI”.

6.6 Zanke

Zanka je niz ukazov, ki se ponavljajo, dokler ni izpolnjen določen pogoj. V zanki je običajno procedura, ki pridobiva in spreminja podatke, sledi pa ji preverjanje pogoja: na primer, ali je neki števec, ki je bil prej določen, dosegel določeno število; če ni, se bo postopek ponavljal, dokler števec ne doseže tega števila. V PHP-ju obstaja veliko vrst zank: while, do-while, for in foreach.

6.7 Funkcije

Funkcija je ponovno uporabljiv, urejen kos kode, ki izvaja eno samo povezano dejavnost. Z njo je programu zagotovljena večja modularnost, poleg tega pa se z njimi izognemo ponavljanju kode. Vsak programski jezik ima določene vgrajene funkcije, lahko pa ustvarimo tudi svoje. Funkcije, metode, procedure itd. so različna imena zanje v različnih programskih jezikih.

7 HTML

HTML, kar pomeni HyperText Markup Language, je standardni označevalni jezik za ustvarjanje spletnih strani in spletnih aplikacij. Z uporabo različnih oznak in atributov HTML omogoča strukturiranje vsebine na spletnih straneh za brskalnike. HTML je temeljna tehnologija World Wide Web-a skupaj z CSS (Cascading Style Sheets) in JavaScript-om. HTML je ključni jezik za razvoj spletnih strani, ki zagotavlja strukturo in pomen vsebine. Z razvojem standardov, kot je HTML5, in poudarkom na dostopnosti in SEO, HTML ostaja temelj uspešnega spletnega razvoja.

7.1 Osnove HTML

HTML dokument je sestavljen iz različnih elementov, ki so označeni z značkami. Te značke povedo spletnemu brskalniku, kako prikazati vsebino. Na primer:

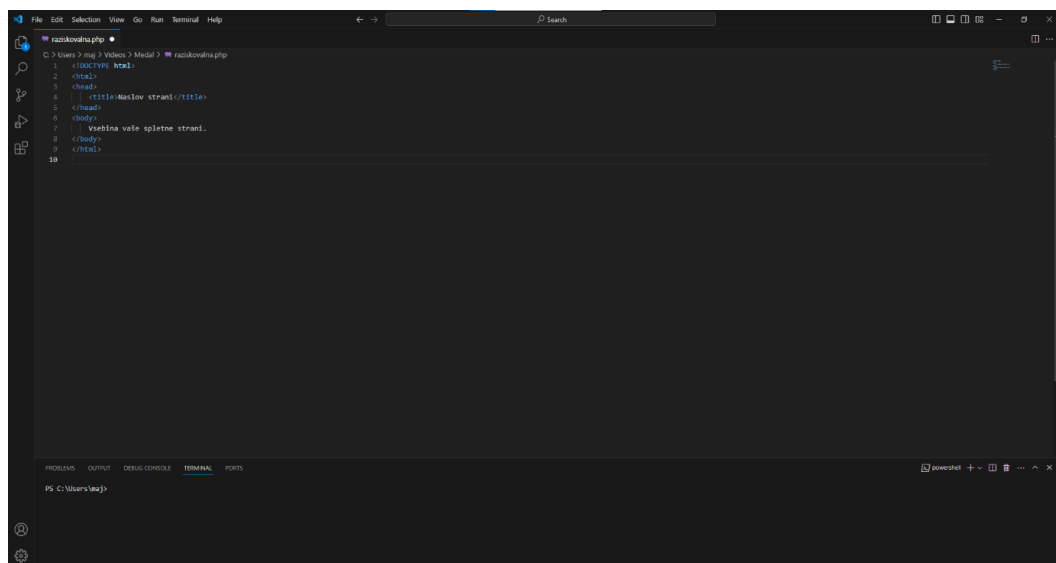
<h1> do **<h6>** so značke za naslove, pri čemer je **<h1>** največji naslov in **<h6>** najmanjši.

<p> označuje odstavek.

<a> definira hiperpovezave.

**** se uporablja za vključitev slik.

7.2 Struktura HTML dokumenta



Slika 9: Struktura html

Vsak HTML dokument ima osnovno strukturo, ki vključuje doctype deklaracijo, <html> element, ki vsebuje celoten dokument, <head> odsek za meta podatke in <body>, kjer se nahaja dejanska vsebina spletne strani.

HTML5 je zadnja večja različica HTML. Prinesla je nove značke in API-je za bolj kompleksne spletne aplikacije, vključno z boljšo podporo za multimedijske elemente kot so video in avdio, kot tudi nove oblike obrazcev, risanje z Canvasom, in podporo za lokalno shranjevanje.

8 CSS

CSS (Cascading Style Sheets) je stilski jezik, ki se uporablja za opis videza in oblikovanja dokumenta, napisanega v označevalnem jeziku, kot je HTML. CSS omogoča spletnim razvijalcem in oblikovalcem nadzor nad videzom spletnih strani, vključno z barvo, pisavo, razmikom med vrsticami, postavitvijo elementov in mnogimi drugimi vizualnimi aspekti.

CSS se nenehno razvija, z novimi specifikacijami, ki dodajajo nove funkcije in izboljšujejo obstoječe zmožnosti. CSS3, najnovejša iteracija CSS, vključuje veliko novih modulov, ki omogočajo še več nadzora nad spletnimi oblikami. Razvoj CSS je usmerjen s strani Svetovnega konzorcija za splet (W3C), ki vzdržuje standarde in specifikacije.

8.1 Glavne značilnosti in prednosti CSS vključujejo:

8.1.1 Ločitev vsebine od oblike

CSS omogoča ločitev vsebine dokumenta od njegove oblike. To pomeni, da lahko spletni razvijalci ustvarijo in vzdržujejo vsebino neodvisno od tega, kako se prikazuje, kar olajša posodobitve in zagotavlja večjo dostopnost.

8.1.2 Kaskadna pravila

CSS sledi določenim pravilom za določanje, katera stilna pravila se uporabijo v primeru konflikta (tj. ko več pravil cilja na isti element). Ta pravila vključujejo specifičnost, pomembnost (npr., pravila, označena z !important) in vrstni red pravil v CSS datoteki ali dokumentu.

8.1.3 Odzivno spletno oblikovanje

CSS omogoča razvoj spletnih strani, ki se prilagajajo različnim velikostim zaslona in ločljivostim, kar pomeni, da spletna mesta lahko izgledajo dobro in so uporabna tako na mobilnih napravah kot na namiznih računalnikih. To se doseže z uporabo medijskih poizvedb, flexbox-a, grid sistema in drugih tehnik odzivnega oblikovanja.

8.1.4 Ponovna uporaba in vzdrževanje

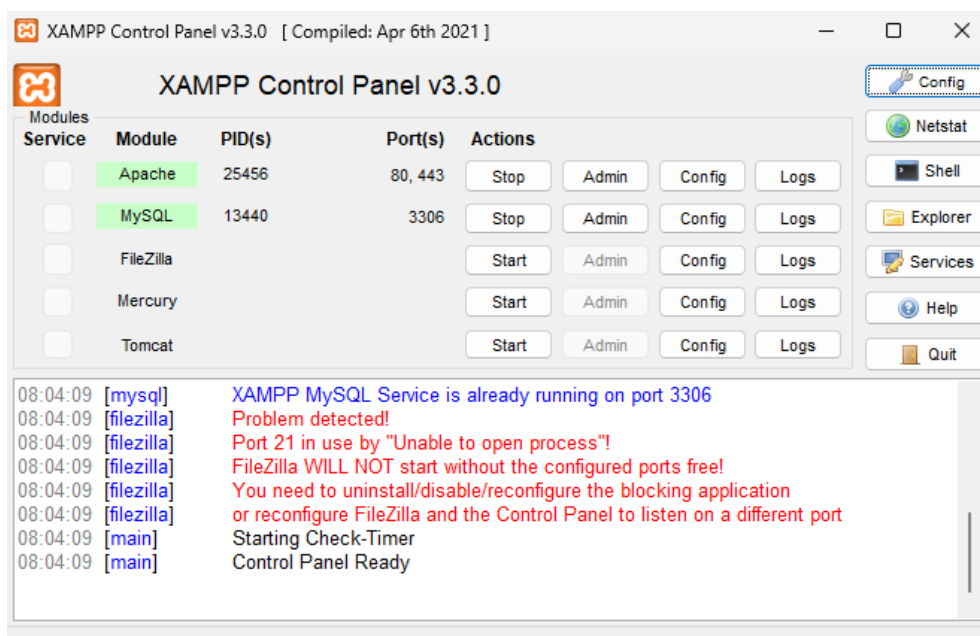
CSS omogoča definiranje stilov v ločenih datotekah, ki se lahko nato ponovno uporabijo na različnih straneh spletnega mesta, kar olajša vzdrževanje in zagotavlja doslednost oblikovanja po celotnem spletnem mestu.

8.1.5 Napredni vizualni učinki

Z uporabo CSS, spletni oblikovalci lahko ustvarijo napredne vizualne učinke, kot so prehodi, transformacije, animacije, sence, gradienti in še več, ki lahko spletnim stranem dodajo privlačnost in dinamiko.

9 XAMPP

XAMPP je brezplačen in odprtokoden preklopni paket, ki vsebuje predvsem Apache HTTP Server, MariaDB (prej MySQL) bazo podatkov, PHP in Perl. Ime XAMPP izhaja iz kratice za Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) in Perl (P), kar pomeni, da je programsko okolje, ki deluje na različnih operacijskih sistemih, kot so Linux, Microsoft Windows, macOS in Solaris.



Slika 10: XAMPP uporabniški vmesnik

Zasnovan je za enostavno namestitev in uporabo, omogoča razvijalcem, da ustvarijo spletno strežniško okolje v minutah, brez zapletenega konfiguriranja. Z njim lahko hitro vzpostavite lokalno testno okolje za razvoj in testiranje spletnih aplikacij, ne da bi morali namestiti in konfigurirati vsako komponento posebej. Zaradi te enostavnosti je postal priljubljena izbira med spletnimi razvijalci, študenti in za izobraževalne namene.

Vključuje tudi dodatna orodja, kot so phpMyAdmin za upravljanje baz podatkov, OpenSSL, Apache Tomcat za Java Servlets in JSP, FileZilla FTP strežnik, Mercury Mail strežnik in druga uporabna orodja, ki razširjajo njegovo funkcionalnost. Omogoča enostavno upravljanje s svojim kontrolnim panelom, kjer lahko zaganjate, ustavljate in upravljate storitve, spreminjate nastavitve in dostopate do log datotek.

V raziskavi, kjer smo se odločali o izbiri programa s katerim bomo lahko povezali podatkovno bazo in gostili našo spletno stran, smo se zlahka odločili za XAMPP. Ima vse funkcije, ki smo jih potrebovali in je lahek za uporabo.

9.1 Komponente

XAMPP sestavljajo glavne komponente: Apache HTTP Server, MariaDB, PHP in Perl. Vsaka od teh komponent ima ključno vlogo v spletnem razvojnem okolju, ki ga XAMPP ustvarja.

Modules							
Service	Module	PID(s)	Port(s)	Actions			
☐	Apache	25456	80, 443	Stop	Admin	Config	Logs
☐	MySQL	13440	3306	Stop	Admin	Config	Logs
☐	FileZilla			Start	Admin	Config	Logs
☐	Mercury			Start	Admin	Config	Logs
☐	Tomcat			Start	Admin	Config	Logs

Slika 11: XAMPP komponente

9.1.1 Apache HTTP Server

Apache HTTP Server je odprtokodni spletni strežnik. Je ena najbolj priljubljenih spletnih strežniških platform in služi kot hrbtenica za dostavo spletnih vsebin na internetu. Obdeluje zahteve od strank (običajno spletnih brskalnikov) in jim služi spletne strani, bodisi statične (HTML datoteke in povezane slike) ali dinamične vsebine, ki so generirane na strani strežnika (npr. s PHP).

Podpira različne module, ki omogočajo razširitev funkcionalnosti, kot so varna povezava SSL/TLS, prepisovanje URL-jev, avtentikacija uporabnikov in še več.

V naši raziskovalni nalogi uporabljamo Apache HTTP Server, zato da lahko vidimo spletno stran, ki smo jo ustvarili. V kodi smo povezali ta Apache HTTP Server z datotekami naše spletne strani.

9.1.2 MariaDB

MariaDB je odprtokodna relacijska baza podatkov, ki je nastala kot nadomestek za MySQL. Omogoča shranjevanje, upravljanje in dostop do podatkov v strukturirani

obliki. Uporablja se za shranjevanje podatkov za spletne aplikacije, od uporabniških računov do objav na blogu in transakcijskih podatkov.

Znana je po svoji visoki zmogljivosti, zanesljivosti in enostavnosti uporabe. Ponuja napredne funkcije, kot so podpora za različne shrambene mehanizme, replikacijo, particioniranje in še več.

9.1.3 PHP

PHP je priljubljen odprtokoden splošni skriptni jezik, ki je posebej oblikovan za razvoj spletnih aplikacij in lahko vgrajen v HTML. Uporablja se za ustvarjanje dinamičnih spletnih strani in aplikacij, omogoča interakcijo z bazami podatkov, upravljanje sej, obdelavo obrazcev in izvajanje kompleksne logike na strežniku.

Podpira številne baze podatkov (kot so MariaDB, MySQL, PostgreSQL...), knjižnice in orodja, kar omogoča razvoj zmogljivih spletnih aplikacij. Je enostaven za učenje za začetnike, hkrati pa ponuja napredne funkcije za izkušene razvijalce.

Za PHP smo se v naši raziskovalni nalogi odločili, ker je povezovanje podatkovne baze s spletno stranjo enostavno in hitro. Sami smo se ga morali naučiti uporabljati, ker ga v šoli še nismo spoznali. Učenje nam ni delalo velikih težav in smo hitro lahko povezali podatkovno bazo z spletno stranjo, oziroma povezali »login.php« in »registration.php« s podatkovno bazo.

Ugotovilo smo, da PHP je začetnikom dokaj prijazen jezik. Na internetu je ogromno lekcij in pomoči, ki so nam pomagale pri težavah.

```
<?php
session_start();
if (isset($_SESSION["user"])) {
header("Location: index.php");
}
?>
```

Slika 12: PHP v login.php

```
<?php

$hostName = "localhost";
$dbUser = "root";
$dbPassword = "";
$dbName = "data";
$conn = mysqli_connect($hostName, $dbUser, $dbPassword, $dbName);
if (!$conn) {
    die("Something went wrong;");
}
//nastavi nastavitve za PB
?>
```

Slika 13: Nastavitve za podatkovno bazo

9.1.4 Perl

Perl je visokonivojski, interpretiran, dinamičen programski jezik, znan po svoji moči v obdelavi besedila in uporabi v skriptah na strani strežnika, avtomatizaciji sistemskih nalog, bioinformatiki in drugih področjih. Perl omogoča razvijalcem hitro pisanje skript za različne naloge, vključno z dostopom do baz podatkov, obdelavo datotek in komunikacijo prek omrežij.

Podpira objektno usmerjeno, proceduralno in funkcionalno programiranje, kar omogoča razvijalcem veliko prilagodljivost pri ustvarjanju skript in programov. Njegova CPAN (Comprehensive Perl Archive Network) knjižnica ponuja ogromno modulov, ki razširjajo njegovo funkcionalnost na številna področja.

9.2 Lastnosti

Lastnosti XAMPP-a so oblikovane tako, da razvijalcem omogočajo hitro vzpostavitev razvojnega okolja brez zapletene konfiguracije.

XAMPP-ova oblika nam je bila lahko razumljiva in nam ni podala pretiranih težav. Njihov uporabniški vmesnik je malo zastarel, ampak še vedno naredi to, kar mora.

9.2.1 Cross-Platform Podpora

XAMPP je na voljo za več operacijskih sistemov, vključno z Windows, macOS, Linux in Solaris. To pomeni, da je razvojno okolje lahko dosledno ne glede na uporabnikov operacijski sistem.

Zaradi svoje platforme neodvisnosti je idealen za ekipne projekte, kjer člani uporabljajo različne operacijske sisteme.

9.2.2 Enostavna namestitvev in uporaba

XAMPP omogoča hitro namestitvev z eno datoteko, ki vključuje vse potrebne komponente, kar odpravlja potrebo po ločeni namestitvi in konfiguraciji Apache, PHP, MariaDB in Perl.

Vključuje tudi enostaven za uporabo kontrolni panel za zagon, ustavitvev in nadzor nad vsemi servisi, kot so Apache, MySQL, in drugi.

9.2.3 Dodatna sledeča orodja

phpMyAdmin: Za enostavno upravljanje baz podatkov preko spletnega vmesnika, kar omogoča uporabnikom, da ustvarjajo, urejajo in brskajo po bazah podatkov brez potrebe po zapletenih SQL ukazih.

SSL podpora: Omogoča enostavno nastavitvev SSL, kar je ključnega pomena za razvoj varnih spletnih aplikacij.

Apache Tomcat: Za tiste, ki razvijajo Java spletne aplikacije, XAMPP vključuje tudi Apache Tomcat.

Mercury Mail Server in **FileZilla FTP Server:** Za testiranje pošiljanja e-pošte in upravljanje datotek preko FTP.

9.2.4 Namen in uporaba

XAMPP je zasnovan kot razvojno in testno okolje, ne kot produkcijsko okolje, zaradi česar je idealen za razvoj spletnih aplikacij, kjer je enostavnost in hitrost ključnega pomena.

Odlično orodje za učenje razvoja spletnih aplikacij, saj omogoča študentom in novincem, da eksperimentirajo z različnimi tehnologijami brez skrbi zaradi kompleksne infrastrukture.

10 Prilagojeno učenje

10.1 Tehnološko ozadje

10.1.1 Integracija API v spletno stran

V procesu implementacije smo ChatGPT API smiselno integrirali v našo spletno stran. Registracija in prijava uporabnikov sta prvi koraki, ki omogočata dostop do funkcionalnosti. Ko uporabnik dostopa do glavne strani, se sreča z obrazcem, kjer lahko naloži datoteke v formatih PDF, DOCX, TXT in PPT. S pomočjo kode in drugih virov, ne da bi posegali po API-ju, pretvorimo te datoteke v čisti besedilni format.

10.1.2 Prednosti uporabe umetne inteligence pri generiranju zapiskov

Uporaba umetne inteligence, zlasti ChatGPT API, pri generiranju zapiskov prinaša številne prednosti. S sposobnostjo učenja iz velike količine podatkov model ustvarja

besedila, ki so kontekstualno povezana in bogata z vsebino. To omogoča generiranje zapiskov različnih vrst glede na uporabnikove želje, kot so dolgi, kratki, podrobni, splošni, zabavni, vprašanja, odgovori ali navadni. Prednosti vključujejo izboljšanje učne izkušnje, hitreje pripravljane zapiskov ter večjo prilagodljivost glede na individualne potrebe uporabnika.

10.2 Oblikovanje in funkcionalnosti spletne strani

10.2.1 Registracija in prijava uporabnikov

V razvoju spletne strani smo posebno pozornost namenili enostavni in varni uporabniški izkušnji. Postopek se začne s registracijo, kjer uporabniki vnašajo svoje podatke, ustvarjajo geslo in dobivajo edinstveno identifikacijo. Po uspešni registraciji se odpre možnost prijave, kjer se uporabniki lahko varno dostopajo do funkcionalnosti spletne strani.

The screenshot shows a web browser window with the title 'Registration Form'. The address bar displays 'localhost/raz/registration.php'. The main content area features a dark-themed registration form with the following elements:

- Input field for 'Name:'
- Input field for 'Email:'
- Input field for 'Password:'
- Input field for 'Repeat Password:'
- A blue button labeled 'Registriraj se'
- A link below the button: 'Ste že registrirani? Vpiši se tukaj'

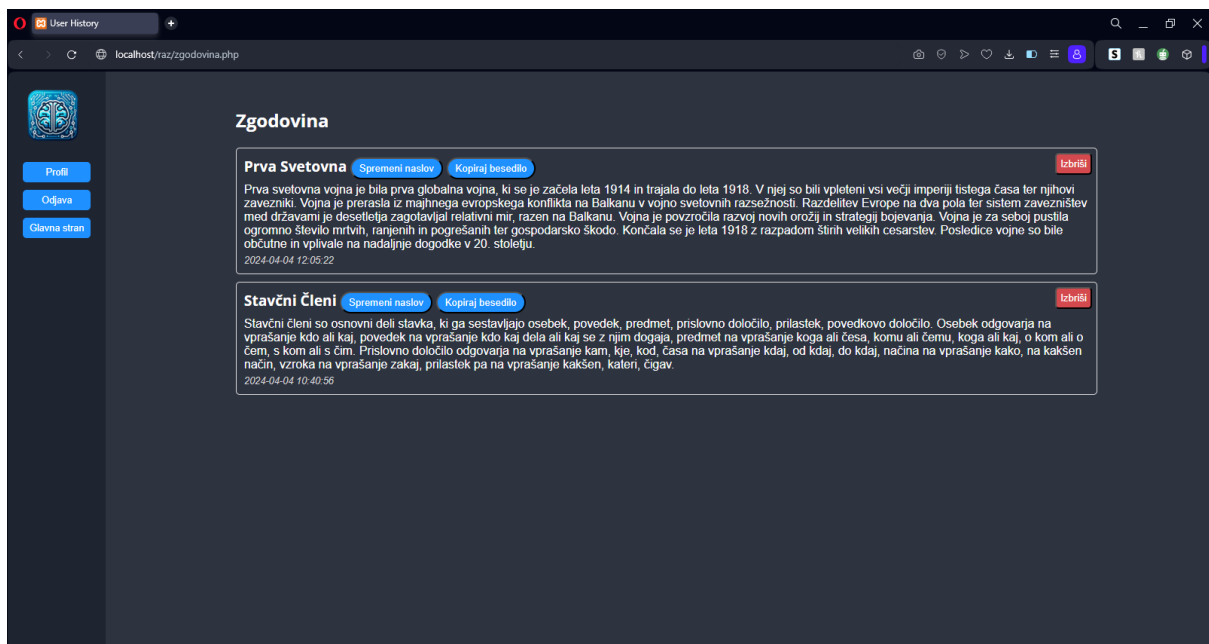
Slika 14: Obrazec za registracijo

The screenshot shows a web browser window with the title 'Login Form'. The address bar displays 'localhost/raz/login.php'. The main content area features a dark-themed login form with the following elements:

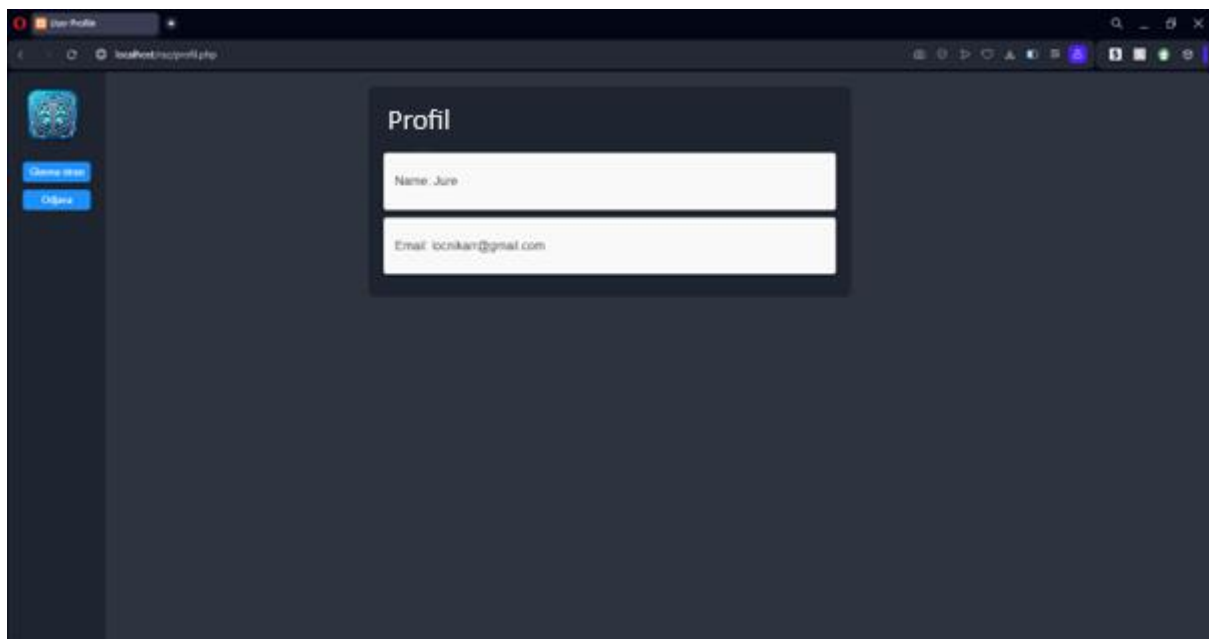
- Input field for 'Vnesi email:'
- Input field for 'Vnesi geslo:'
- A blue button labeled 'Vpiši se'
- A link below the button: 'Še niste registrirani? [Registriraj se tukaj](#)'

Slika 15: Obrazec za vpis

Za implementacijo prijave oziroma registracije v aplikacijo, smo se odločili, ker želimo uporabniku zagotoviti dostop do preteklih zapiskov, ki je mogoč pod zgodovino ter dostop do uporabnikovega profila.



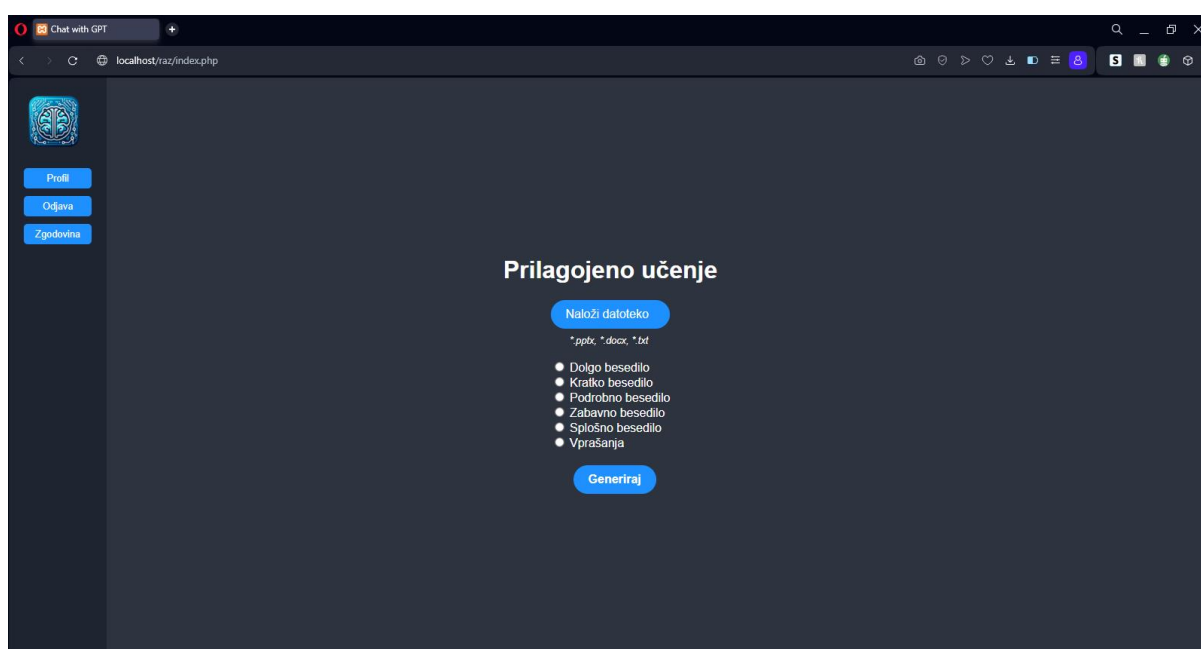
Slika 16: Zgodovina



Slika 17: Uporabnikov profil

10.2.2 Vmesnik za nalaganje datotek

Glavni vmesnik spletne strani omogoča enostavno in pregledno nalaganje datotek v različnih formatih, kot so .pdf, .docx, .txt in .ppt. Uporabniki preko intuitivnega obrazca hitro naložijo svoje dokumente, kar predstavlja začetni korak k pridobivanju zapiskov. Vmesnik je oblikovan tako, da omogoča hitro in brežhibno navigacijo.



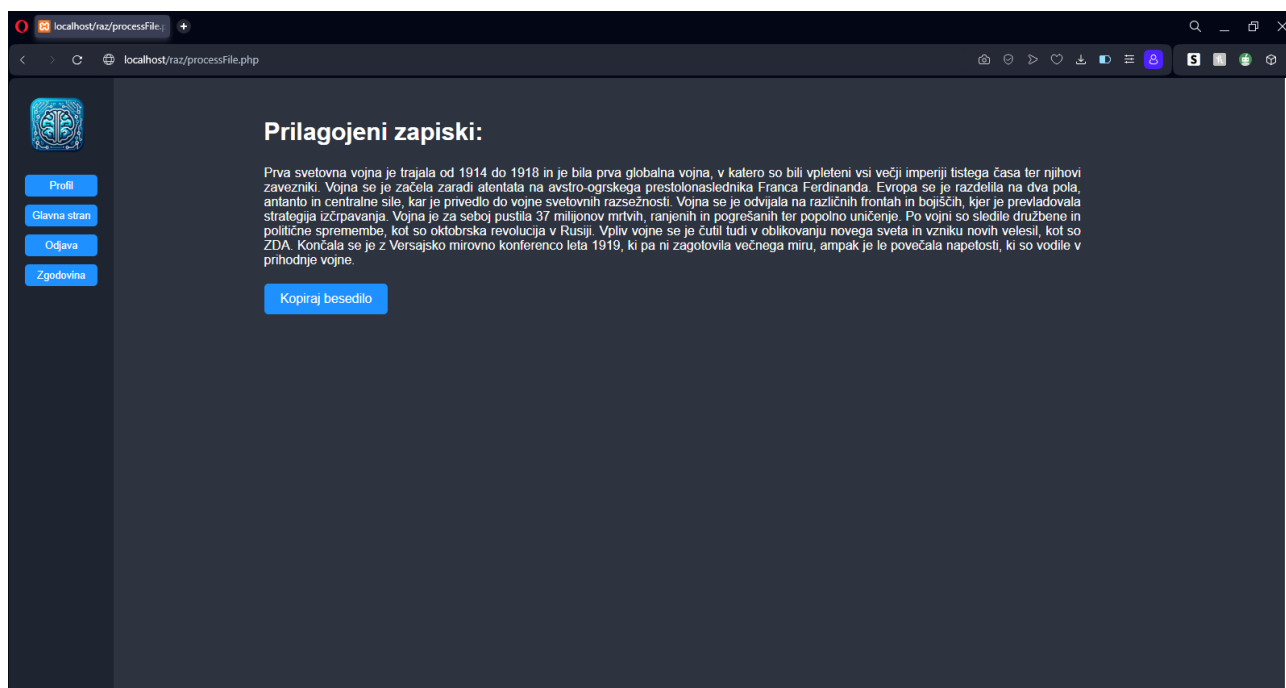
Slika 18: Glavna stran

10.2.3 Implementacija kode za pretvorbo datotek v besedilni format

Pomemben del funkcionalnosti spletne strani je implementacija kode, ki omogoča pretvorbo naloženih datotek v čisti besedilni format. Z uporabo strokovno razvite kode in drugih virov zagotavljamo, da se podatki ohranijo, medtem ko se formati spremenijo. To omogoča uporabnikom, da brez težav generirajo zapiske iz svojih dokumentov, ne da bi posegali po zunanjih orodjih ali API-jih. S tem pristopom želimo

zagotoviti uporabnikom preprosto in učinkovito uporabo spletne platforme za generiranje zapiskov.

10.3 Proces generiranja zapiskov



Slika 19: Generirano besedilo iz zapiskov

Generirano besedilo lahko uporabnik kopira, v prihodnosti pa načrtujemo, da bo lahko uporabnik zapiske shranil na računalnik kot .docx datoteko.

10.3.1 Nastavitve generiranja zapiskov

Ključni del našega raziskovalnega pristopa je prilagodljivost uporabniških nastavitev. Uporabniki imajo možnost izbire različnih parametrov, ki vplivajo na generiranje zapiskov. S tem omogočamo individualno prilagajanje izkušnje, kot so dolžina zapiskov,

stopnja podrobnosti, zabavnost ter vrsta vsebine, kot so vprašanja, odgovori ali normalni zapiski.

Ti uporabniku omogočajo personalizirane zapiske, ki jih pridobi z našo spletno aplikacijo. Za omogočanje različnih nastavitev smo v kodo implementirali spremenljivke, ki vsebujejo navodila za poizvedbo določene nastavitve.

10.3.2 Raziskovanje različnih vrst zapiskov

Raznolikost je ključna pri generiranju zapiskov. Z raziskovanjem različnih vrst zapiskov, želimo zagotoviti, da se uporabniki srečujejo z vsebinami, ki najbolj ustrezajo njihovim specifičnim potrebam. Izbira med dolgimi in kratkimi zapiski, podrobnimi ali manj podrobnimi ter zabavnimi vsebinami predstavlja raznolikost, ki odraža širok spekter uporabniških preferenc. Te preference smo pridobili z raziskovanjem. Vprašali smo učence kakšne vrste si želijo in profesorje za predloge za najpomembnejše informacije in s svojimi izkušnjami smo ugotovili in prilagodili nastavitve.

10.3.3 Uporaba API za ustvarjanje personaliziranih zapiskov

Pri implementaciji smo se osredotočili na izkoriščanje ChatGPT API za ustvarjanje personaliziranih zapiskov. S tem pridobivamo dodatno dimenzijo prilagajanja, saj model lahko bolje razume specifične zahteve uporabnikov in generira vsebine, ki so še bolj prilagojene posameznikovim potrebam.

10.4 Uporabniška izkušnja

10.4.1 Pregled glavne spletne strani

Uporabniška izkušnja se začne z jasnim in preglednim pregledom glavne spletne strani. Enostaven dostop do funkcionalnosti, kot je nalaganje datotek, omogoča hitro navigacijo in uporabnikom prijazno okolje za izvajanje njihovih nalog. Pozanimali smo se pri dijakih, kakšen uporabniški vmesnik si želijo. Te informacije smo uporabili v grajenju vmesnika za našo spletno aplikacijo.

10.4.2 Vmesnik za upravljanje nastavitev generiranja

Vmesnik za upravljanje nastavitev generiranja predstavlja ključno točko, kjer uporabniki prilagajajo parametre generiranja zapiskov. Intuitivnost in preprostost tega vmesnika povečujeta uporabniško udobje, kar omogoča hitro prilagajanje glede na želje posameznega uporabnika.

10.4.3 Možnosti shranjevanja in kopiranja generiranih zapiskov

Dodatna funkcionalnost omogoča uporabnikom, da shranjujejo in kopirajo generirane zapiske. To je ključno za trajno shranjevanje informacij ali deljenje z drugimi. S tem zagotavljamo, da so generirane vsebine enostavno dostopne in uporabne v različnih kontekstih.

Shranjevanje in kopiranje sta bili pri dijakih najbolj zaželeni funkciji.

10.5 Evalvacija rezultatov

10.5.1 Merjenje učinkovitosti generiranja zapiskov

Za merjenje učinkovitosti smo vpeljali metrike, ki ocenjujejo kakovost generiranih zapiskov. To vključuje natančnost informacij, ustreznost glede na uporabniške zahteve ter splošno jasnost in berljivost besedila.

10.5.2 Primerjava različnih nastavitev in njihov vpliv na kakovost zapiskov

S poglobljeno analizo primerjamo različne nastavitve generiranja in njihov vpliv na končno kakovost zapiskov. To nam omogoča boljše razumevanje optimalnih parametrov za različne uporabniške scenarije.

10.6 Prilagojeno učenje v praksi

10.6.1 Pogovor s profesorji

Profesorjem smo podali zapiske, ki jih je generirala naša spletna aplikacija. Med pogovorom smo jih vprašali, če zapiski vsebujejo vso potrebno snov, ki je v standardih za odlično oceno. Potrdili so, da zapiski vsebujejo dovolj informacij, ki so potrebne, da dijak doseže odlično oceno.

Pogovarjali smo se z petimi profesorji. Štirje so potrdili podrobnost naših zapiskov, eden pa je predlagal manjše izboljšave in podal informacije, katerih naša spletna aplikacija ni izluščila iz besedila.

Ob izdelavi zapiskov, smo ugotovili, da se ključne informacije začnejo izgubljati, če besedilo skrčimo na manj kot 30% podanega gradiva. Zato smo v poizvedbi, ki skrbi za izdelavo zapiskov, podali navodila, da je spodnja meja za krčenje 35%.

10.6.2 Testiranje dijakov

Šest dijakov smo prosili, če bi nam pomagali pri testiranju naše spletne aplikacije. Prvim trem smo naročili, naj se učijo snov tako, kot po navadi, drugim trem, pa da si naj s pomočjo naše spletne aplikacije generirajo zapiske in se učijo iz njih. Prišli smo do ugotovitve, da so učenci pridobili zelo podobne ocene, glavna razlika pa je bila v času, ki so ga porabili za učenje.

Upoštevali smo predznanje dijakov, zato smo izbrali dijake, ki imajo isto povprečje, v našem primeru, prav dobro.

Vsi dijaki so se učili isto snov, pod istimi pogoji. Na voljo so imeli dva dni časa, učenje pa je potekalo doma, ob večernih urah. Ocene, ki so jih dijaki pridobili so sledeče:

Tabela 1: Primerjava ocen

Klasični način	Prilagojen način
3	4
4	5
4	4

Po ocenjevanju smo se pogovorili s profesorico, pri kateri so bili dijaki ocenjeni. Večjih razlik v znanju ni zaznala, prav tako je ocenila, da imajo dijaki celovit pregled nad snovjo.

Rezultate naše raziskave je seveda potrebno vzeti z malce rezerve, saj jih je težko posplošiti, ker na same ocene močno vpliva tudi prisotnost drugih okoliščin.

11 Zaključek

11.1 Povzetek doseženih ciljev

Naša raziskovalna naloga je dosegla zastavljene cilje, ki vključujejo uspešno implementacijo ChatGPT API-ja, razvoj funkcionalne spletne strani ter zagotavljanje uporabniške izkušnje, ki izpolnjuje pričakovanja.

11.2 Razprava o rezultatih in njihov pomen

Rezultati kažejo na uspešnost uporabe umetne inteligence pri generiranju zapiskov ter pomembnost prilagodljivosti in preglednosti v spletnem okolju.

11.3 Prihodnje možnosti raziskovanja in nadgradnje sistema

Za prihodnost naše spletne aplikacije načrtujemo izboljšan uporabniški vmesnik in dodatne funkcije, ki uporabniku naše spletne aplikacije olajšajo delo in izboljšajo izkušnjo.

11.3.1 Dodatne funkcije

Uporabniku naše spletne aplikacije želimo omogočiti ne le kopiranje besedila oziroma vprašanj, temveč tudi možnost shranjevanja generiranih zapiskov na računalnik v obliki .docx datoteke.

Načrtujemo možnost ustvarjanja zaznamkov, ki jih lahko naš uporabnik doda njegovim že generiranim zapiskom in mu s tem omogočiti filtriranje po njegovih ustvarjenih zaznamkih za lažje iskanje in boljšo uporabniško izkušnjo.

V prihodnosti želimo implementirati tudi iskalno vrstico, ki bo uporabniku omogočila brskanje po njegovi zgodovini zapiskov.

12 Ovrednotenje hipotez

Hipotezo 1, ki trdi, da bo naše orodje izdelovalo izčrpne zapiske, ki vsebujejo vse potrebne informacije, lahko potrdimo.

Na podlagi povratnih informacij profesorjev smo ugotovili, da naše orodje ustvarja izčrpne zapiske, ki zajemajo vse bistvene informacije in se lahko primerjajo z njihovimi zapiski in/ali učbenikom.

Hipotezo 2, ki pravi, da bo asistent pospešil učenje za vsaj 20%, lahko potrdimo.

Testirali smo dijake, ki se učijo tako, da najprej napišejo zapiske iz katerih se učijo. Izmed teh so dijaki, ki so za ustvarjanje zapiskov uporabljali našega asistenta, končali hitreje, kot tisti, ki so zapiske pisali sami.

Hipotezo 3, ki trdi, da nam bo zaradi pomankanja predznanja API predstavljal težave, lahko potrdimo.

Iz prakse lahko potrdimo, da je bilo izdelovanje spletne aplikacije težje, kot smo pričakovali. Z API-jem se do zdaj še nismo srečali, zato smo ga morali najprej raziskati in se o njem pozanimati. Kljub temu, nam je zaradi pomanjkanja izkušenj predstavljal težave.

Hipotezo 4, ki pravi, da bo asistent pomagal dijakom dobiti višje ocene, lahko potrdimo.

Primerjali smo dijake, ki imajo isto povprečje in so do zdaj imeli vsako leto isti uspeh. Dijaki, ki so se učili s pomočjo našega asistenta, so imeli isto ali boljše povprečje kot tisti, ki so se učili na isti način kot do zdaj.

13 Viri in literature

Apache Friends (XAMPP). Pridobljeno 17. november 2023 iz

<https://www.apachefriends.org/index.html>

Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. John Wiley & Sons, Inc.

Pridobljeno 21. november 2023

Kevin Tatroe, P. M. (2013). *Programming PHP, 4th Edition*. O'Reilly Media, Inc.

Pridobljeno 21. november 2023

Lynn Beighley, M. M. (2008). *Head First PHP & MySQL*. O'Reilly Media, Inc.

Pridobljeno 20. november 2023

McFarland, D. S. (2015). *CSS: The Missing Manual, 4th Edition*. O'Reilly Media, Inc.

Pridobljeno 21. november 2023

MDN Web Docs (CSS). Pridobljeno 17. november 2023 iz

<https://developer.mozilla.org/en-US/docs/Web/CSS>

MDN Web Docs (HTML). Pridobljeno 17. november 2023 iz

<https://developer.mozilla.org/en-US/docs/Web/HTML>

OpenAI Documentation. Pridobljeno 17. november 2023 iz

<https://platform.openai.com/docs/introduction>

PHP: The Right Way. Pridobljeno 17. november 2023 iz

<https://www.phptherightway.com/>

Visual Studio Code Documentation. Pridobljeno 17. november 2023 iz

<https://code.visualstudio.com/docs>