# **Uvod**

U današnjem digitalnom dobu, tehnologija igra ključnu ulogu u obrazovanju, pružajući inovativne načine za poboljšanje jezičnih vještina. Pravopis je jedna od temeljnih vještina u učenju jezika, a njegovo pravilno usvajanje ključno je za uspješnu komunikaciju. Primjena tehnologije u obrazovanju omogućava prilagodljiv pristup učenju i podučavanju pravopisa kroz interaktivne alate koji su sposobni prilagoditi zadatke prema potrebama kako učenika, tako i nastavnika. Ovakav pristup ne samo da poboljšava učinkovitost učenja, već i povećava angažman učenika, čineći proces učenja pravopisa dinamičnijim i zanimljivijim, a smanjuje angažman nastavnika pri sastavljanju i ispravljanju zadataka čineći proces podučavanja olakšanim i manje stresnim. Integracija tehnologije u podučavanje pravopisa omogućava nastavnicima automatizirano sastavljanje i ispravljanje zadataka, kao i praćenje i analizu napretka učenika. Korištenje generativne umjetne inteligencije za sastavljanje zadataka, zajedno s automatiziranim alatima za ispravljanje poput Ispravi.me, značajno olakšava administrativno opterećenje nastavnika. Time im se pruža mogućnost da više pažnje posvete interakciji s učenicima i pružanju dodatne podrške. Kroz automatizirano analizu učeničkih postignuća, nastavnici mogu dobiti detaljan uvid u područja koja zahtijevaju dodatnu pažnju te prilagoditi svoje metode podučavanja kako bi zadovoljili potrebe svakog učenika. Uporaba tehnologije u obrazovanju tako značajno olakšava rad nastavnicima, omogućujući im da se usmjere na pružanje kvalitetnijeg i učinkovitijeg obrazovanja.

Većina trenutačnih sustava i aplikacija za pravopisno učenje, ne koriste generativnu umjetnu inteligenciju za stvaranje zadataka. Ona donosi niz koristi, uključujući raznolikost zadataka, kontinuiranu dostupnost novog materijala te automatizaciju procesa. Iako su ove prednosti očite, većina pravopisnih sustava još uvijek ne integrira generativnu umjetnu inteligenciju niti funkcionalnosti ispravljanja pravopisa koje nudi sustav Ispravi.me. Ispravi.me se ističe u odnosu na druge aplikacije i sustave za pravopisno ispravljanje, kao što je Grammarly, nudeći funkcije kao što su evidencija čestih pravopisnih grešaka, pozadinsko učenje novih riječi te kontekstualno ispravljanje.

U sklopu ovog diplomskog rada proučit će se literatura iz područja upotrebe tehnologije u obrazovanju, kao i postojeće aplikacije za vježbanje pravopisa i gramatike. Obavit će se temeljna analiza arhitekture sustava, korištenih tehnologija, alata i programskih jezika te pregled modela podataka. Potom slijedi opis razvijenog rješenja koje, za razliku od drugih aplikacija za vježbanje pravopisa, uključuje funkcionalnost digitalnog diktata. Posebna pažnja bit će usmjerena na proces izrade zadataka, uključujući korištenje API-ja kao i integraciju sa sustavom Ispravi.me. Na kraju, bit će predložene smjernice za daljnji razvoj i poboljšanje sustava, te će bit iznesen zaključak.

# **1. Pregled područja i slične aplikacije**

## **Tehnologija u obrazovanju**

Tehnologija u obrazovanju odnosi se na upotrebu digitalnih alata, softvera i online resursa za poboljšanje i podršku obrazovnog procesa. U današnje vrijeme, gotovo je nemoguće zamisliti formalno i neformalno obrazovanje bez korištenja računala i novih tehnologija. Računala i internet postali su sastavni dio svakodnevnog života učenika i nastavnika, koriste se prije, poslije, ali i tijekom nastavnog procesa. Integracija tehnologije u obrazovni sustav danas uključuje širok raspon alata kao što su e-udžbenici, online tečajevi, edukativne aplikacije i platforme za učenje na daljinu. Razvoj tehnologije omogućava korištenje raznih multimedijskih elemenata – zvuka, videa, simulacija i animacija, uz tekst i slike. Važno je pažljivo odabrati alate i multimedijski softver koji se koristi u edukaciji, te osigurati da različiti elementi uspješno funkcioniraju u zadanom kontekstu obrazovanja. Multimedija i nove tehnologije pružaju nastavnicima niz novih alata i inovativnih pristupa podučavanju, dok učenicima omogućuju brže, funkcionalnije, raznovrsnije i pristupačnije učenje.

**Prednosti tehnologije u obrazovanju:**

1. **Personalizacija učenja:** Tehnologija omogućuje prilagodbu nastavnih sadržaja individualnim potrebama i tempu učenika. Adaptivni sustavi za učenje mogu identificirati slabosti učenika i pružiti ciljane materijale za poboljšanje.
2. **Interaktivnost:** Digitalni alati i resursi mogu učiniti učenje interaktivnijim i zanimljivijim. Kvizovi, igre, simulacije i virtualni laboratoriji omogućuju učenicima da aktivno sudjeluju u procesu učenja, što može poboljšati njihovu motivaciju i angažman.
3. **Povećana dostupnost resursa:** Tehnologija omogućuje pristup širokom spektru obrazovnih resursa, uključujući e-udžbenike, online tečajeve, video predavanja i interaktivne alate. To može posebno koristiti učenicima u udaljenim ili nedovoljno opremljenim područjima.
4. **Fleksibilnost učenja:** Online i hibridni modeli obrazovanja omogućuju učenicima da uče vlastitim tempom i prema vlastitom rasporedu. To može biti posebno korisno za učenike s različitim stilovima učenja i obavezama izvan škole.
5. **Poboljšana suradnja:** Digitalne platforme za učenje omogućuju lakšu suradnju među učenicima i nastavnicima, bez obzira na njihovu lokaciju. Alati poput foruma, chatova i zajedničkih dokumenata olakšavaju razmjenu ideja i timski rad.

**Mane tehnologije u obrazovanju:**

1. **Tehnološka nejednakost:** Pristup tehnologiji nije jednak za sve učenike. Socioekonomski status može utjecati na dostupnost digitalnih uređaja i internetske veze, što može dovesti do digitalnog jaza.
2. **Distrakcije:** Korištenje digitalnih uređaja u obrazovanju može dovesti do distrakcija. Učenici mogu biti skloni odvlačenju pažnje na društvenim mrežama, igrama i sl.
3. **Tehnički problemi:** Tehničke poteškoće, poput problema s internetskom vezom, kvarova uređaja ili bugova.
4. **Privatnost i sigurnost:** Korištenje tehnologije u obrazovanju postavlja izazove u pogledu zaštite privatnosti i sigurnosti učenika. Prikupljanje podataka o učenicima mora biti u skladu s pravilima o zaštiti privatnosti i sigurnosti podataka.
5. **Potreba za obukom:** Nastavnici trebaju adekvatnu obuku kako bi učinkovito koristili tehnologiju u nastavi. Nedostatak vještina ili znanja o digitalnim alatima može ograničiti njene prednosti.

Multidisciplinarni pristup razvoju edukativnih materijala je ključan, budući da nastavnici nisu nužno informatički stručnjaci. Projektni pristup razvoju materijala i uloge koje sudjeluju u tom procesu također su važni aspekti [1]. U literaturi i javnosti su se uvriježili i novi termini, pa se tako govori o on-line učenju (online learning), e-učenju (e-learning) te o učenju utemeljenom na web-u (web-based learning), a u novije vrijeme i učenju putem mobilnih uređaja (m-learning) [1]. Razvoj informacijsko-komunikacijske tehnologije mijenja paradigmu učenja i obrazovanja primjenom suvremenih metoda i oblika nastave. Istaknuta je didaktička dimenzija primjene suvremenih tehnologija u procesu nastave kao i promjena didaktičke perspektive dionika obrazovanja. Suvremena se nastava temelji na didaktičkom četverokutu koji uključuje tehnologiju, učenika, učitelja i nastavni sadržaj [2].

Kada se tehnologija poveže s učenjem pravopisa, može značajno unaprijediti proces kroz interaktivne i prilagodljive metode učenja. Digitalni alati omogućuju trenutne povratne informacije, prilagodbu zadataka individualnim potrebama učenika i kontinuirano praćenje napretka. Korištenje tehnologije za učenje pravopisa omogućuje lakše savladavanje pravopisnih pravila kroz igre, kvizove i digitalne diktate, čime se potiče motivacija i angažman. Također, učitelji mogu koristiti tehnologiju za učinkovito ispravljanje zadataka i pružanje dodatne podrške, smanjujući administrativno opterećenje i omogućujući im više vremena za interakciju s učenicima. Općenito, pravopis je temeljni aspekt pismenosti koji uključuje niz kognitivnih procesa. Tijekom godina, isprobane su mnoge metode podučavanja pravopisa, s različitim stupnjevima uspjeha. Otkako je prvo računalo bilo uvedeno u škole, jedno popularno područje rasprave bilo je je li korištenje tehnologije bolje od korištenja papira za učenje vještina, uključujući pravopis [3]. Mnogo je istraživanja u proteklih 20 godina to istraživalo [3]. U novije vrijeme, bilo je mnogo studija o korištenju aplikacija za pomoć razvoju pismenosti. Međutim, još uvijek nije postignut konsenzus o tome koja je metoda najkorisnija. Na kraju se ispostavi da je tehnologija poželjnija ponajprije radi njenih ranije spomenutih prednosti, iako je korištenje papira potencijalno korisnije zbog učinaka pamćenja i motoričkih sposobnosti [3].

Tehnologija u obrazovanju nosi sa sobom niz etičkih i moralnih izazova koji zahtijevaju pažljivo razmatranje i upravljanje. To je važno istaknuti jer se u današnjem obrazovanju, uključujući i ovaj rad, koristi umjetna inteligencija. AI sustavi često donose odluke na temelju složenih algoritama koji se temelje na podacima. Međutim, moguće je da ovi algoritmi mogu reflektirati predrasude i pristranosti prisutne u podacima na kojima su trenirani, što može rezultirati nepravednim odlukama. Na primjer, AI sustavi za ocjenjivanje učenika mogu preferirati određene stilove pisanja ili izraza na temelju prethodnih podataka, što može rezultirati nejednakim procjenama. Osim toga, postavlja se pitanje transparentnosti i odgovornosti u korištenju AI u obrazovanju. Kako ovi sustavi često donose odluke na temelju kompleksnih algoritama, korisnici, uključujući učenike, nastavnike i roditelje, možda ne razumiju uvijek kako su donesene te odluke. Nedostatak transparentnosti može dovesti do nedostatka povjerenja u AI sustave i rezultate koje oni proizvode.

## **Slične aplikacije i sustavi**

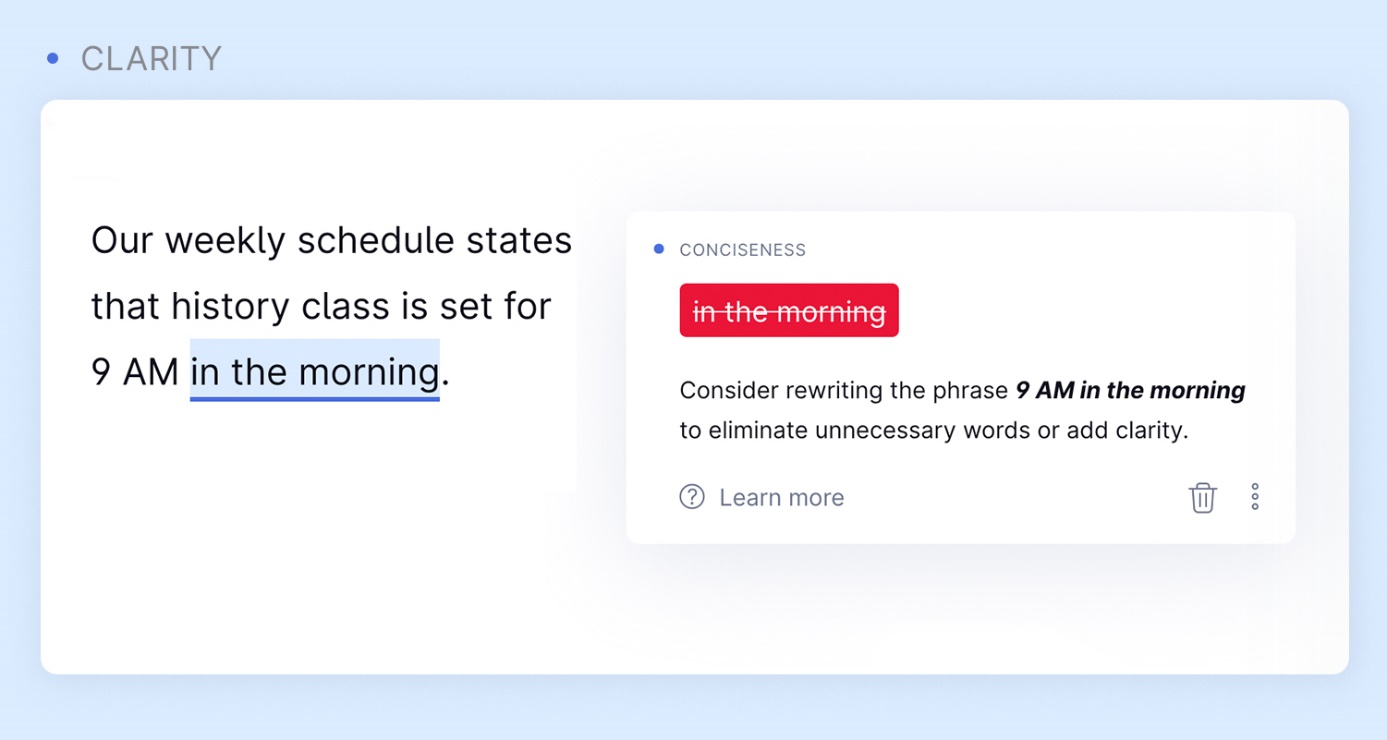
Većina postojećih aplikacija za učenje pravopisa uglavnom se fokusira na osnovne funkcionalnosti poput vježbi i lekcija, ali im često nedostaje inovativan pristup i interaktivnost. Na primjer, aplikacije poput Grammarly i SpellCheckPlus omogućuju korisnicima provjeru pravopisa i gramatike, ali im nedostaje element igre i motivacije koji bi dodatno potaknuli angažman učenika. To su aplikacije više usmjerene na analizu i ispravak teksta, pružajući korisnicima povratne informacije o pravopisnim i gramatičkim greškama. Slično tome, aplikacije poput Duolingo, koje nude vježbe iz različitih jezika, ne nude specifične alate ili lekcije posvećene samo pravopisu. Spellzone je još jedna aplikacija za učenje pravopisa koja pruža lekcije, vježbe i testove za poboljšanje pravopisa. Međutim, u usporedbi s aplikacijama koje koriste napredne tehnologije poput umjetne inteligencije (AI), Spellzone se često oslanja na tradicionalne metode učenja. To znači da nema integriranu AI za personalizaciju lekcija ili ispravljanje grešaka. Iako ove aplikacije pružaju korisne funkcionalnosti, nedostaje im personalizirani pristup i interaktivni elementi koji bi učenje pravopisa učinili automatiziranim i lakšim za nastavnike, a zabavnijim i učinkovitijim za učenike.

Dodatno, nedostaju im alati ili sučelje namijenjeni nastavnicima. Sučelje bi omogućilo praćenje napretka učenika, prilagođavanje lekcija prema njihovim potrebama i pružanje dodatne podrške. Isto tako, bilo bi moguće omogućiti učenicima praktičnu vježbu slušanja i pisanja kroz digitalni diktat. Ni jedna od navedenih aplikacija ne nudi ovu funkcionalnost koja bi mogla biti izuzetno korisna za učenje pravopisa i bila bi izvrsna vježba za primjenu pravopisnih pravila u stvarnom vremenu.

### **Grammarly**

Grammarly je platforma za poboljšanje pisanja koja koristi umjetnu inteligenciju za ispravljanje pravopisnih, gramatičkih i stilskih pogrešaka. Dostupna je kao web aplikacija, proširenje za preglednike te dodatak za Microsoft Office i mobilne uređaje. Grammarly pomaže korisnicima da pišu s više točnosti i jasnoće u raznim kontekstima, uključujući poslovnu komunikaciju, akademsko pisanje i svakodnevnu upotrebu [4]. Grammarly analizira tekst u stvarnom vremenu, podcrtavajući greške i nudeći prijedloge za ispravke. Korisnici mogu unositi tekst izravno u Grammarly-jev uređivač ili kopirati i zalijepiti sadržaj za provjeru. Kao proširenje za preglednike, omogućuje automatsku provjeru teksta dok korisnici pišu online, dok dodatak za Microsoft Office omogućuje provjeru dokumenata u Wordu i Outlooku [4]. Korištenjem Grammarly-ja korisnici mogu značajno unaprijediti svoje pravopisne vještine zbog nekoliko ključnih funkcionalnosti:

* **Ispravak pravopisa i gramatike:** Automatski prepoznaje i ispravlja pravopisne i gramatičke pogreške, osiguravajući točnost u pisanju.
* **Stilske preporuke:** Pruža savjete za poboljšanje stila pisanja, jasnoće i konzistentnosti, prilagođene korisnikovom stilu.
* **Provjera plagijata:** nudi alat za provjeru plagijata, uspoređujući tekst s milijunima web stranica i akademskih radova.
* **Kontekstualne preporuke:** Koristi naprednu umjetnu inteligenciju za davanje kontekstualnih preporuka koje poboljšavaju kvalitetu pisanja.
* **Personalizirani pristup:** Uči iz korisnikovih prethodnih ispravaka i prilagođava prijedloge prema njihovom stilu pisanja.

Grammarly također nudi mogućnost timskog rada i kolaboracije u poslovnim okruženjima, omogućujući korisnicima zajedničko uređivanje dokumenata i dijeljenje prijedloga za ispravke [4]. Integracija naprednih tehnologija poput umjetne inteligencije čini Grammarly moćnim alatom za poboljšanje pravopisa i učinkovitosti pisanja, no nedostaje mu element igre i mogućnost sastavljanja provjera.

Slika 1.1 Grammarly - Provjera osnovnih gramatičkih grešaka

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, dizajn

Opis je automatski generiran

Slika 1.2 Grammarly - Prijedlozi za nejasne rečenice i suvišne fraze

### **Spellzone**

Spellzone je online platforma za učenje pravopisa, osmišljena da pomogne učenicima u poboljšanju pravopisnih vještina. Dostupna je putem web aplikacije, koja nudi niz interaktivnih lekcija, vježbi i testova prilagođenih različitim razinama znanja [5]. Spellzone je posebno koristan alat za nastavnike i roditelje, jer omogućuje praćenje napretka učenika i prilagođavanje sadržaja njihovim specifičnim potrebama [5]. Neke od značajki Spellzone-a su:

**Lekcije i vježbe**: struktura lekcija i vježbi su osmišljene kako bi postupno uvodile učenike u različite aspekte pravopisa, omogućujući im da napreduju vlastitim tempom.

**Testovi i kvizovi**: Redoviti testovi pomažu učenicima da prate svoj napredak i identificiraju područja koja zahtijevaju dodatnu pažnju.

**Resursi za učitelje**: Spellzone pruža učiteljima alate i materijale koji im pomažu integrirati platformu u nastavni plan i program, uključujući mogućnost prilagodbe lekcija specifičnim potrebama učenika i praćenje njihovog napretka.

Unatoč svojim prednostima, Spellzone ima ograničenja u usporedbi s nekim naprednijim aplikacijama. Na primjer, ne koristi umjetnu inteligenciju za generiranje personaliziranih zadataka niti za davanje kontekstualnih preporuka, što može biti značajan nedostatak u odnosu na platforme poput Grammarly.

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, web-stranica

Opis je automatski generiranKorisnici započinju s registracijom na Spellzone web stranici, gdje se mogu prijaviti kao učenici, učitelji ili roditelji. Nakon prijave, dolaze na intuitivan dashboard koji omogućava jednostavnu navigaciju kroz različite module i alate. Platforma nudi razne lekcije i vježbe koje su organizirane prema razinama težine i temama. Svaka lekcija sadrži detaljne upute, primjere i objašnjenja kako bi korisnici mogli razumjeti pravila pravopisa. Nakon svake lekcije, korisnici rješavaju interaktivne vježbe koje im pomažu primijeniti naučeno. Ove vježbe uključuju razne aktivnosti poput upisivanja riječi, povezivanja, slaganja rečenica i više, što omogućava učenicima da na praktičan način utvrde svoje znanje. Platforma također omogućuje redovite testove koji korisnicima pomažu procijeniti njihov napredak i identificirati područja koja zahtijevaju dodatnu pažnju. Učitelji mogu koristiti alate za praćenje napretka svojih učenika, prilagođavati lekcije njihovim potrebama i kreirati specifične zadatke kako bi poboljšali učenje. Roditelji također mogu pratiti napredak svoje djece i pružiti im dodatnu podršku.

Slika 1.3 Spellzone - pravopisni zadatak s unosom i sučelje sa sistematiziranim prikazom lekcija

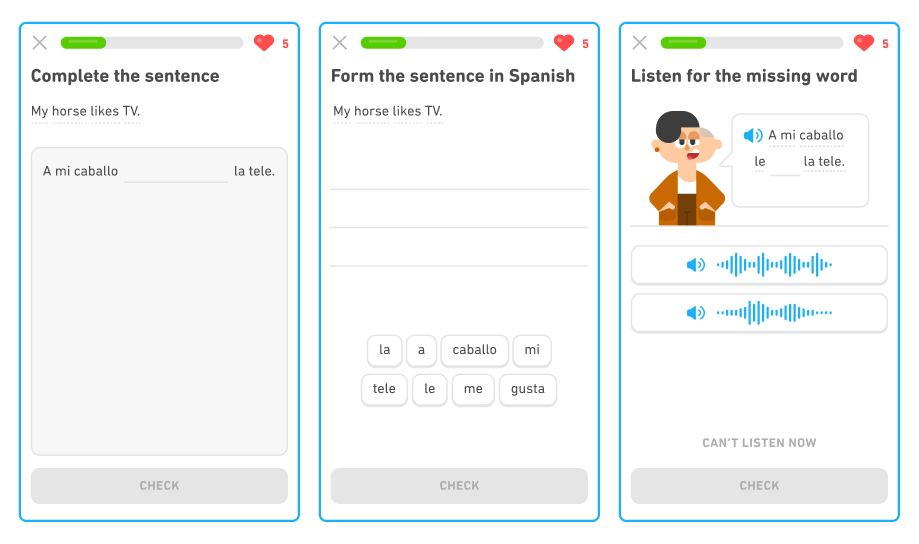
U cjelini, Spellzone je vrlo koristan alat za učenje pravopisa, posebno za obrazovne ustanove i individualne korisnike koji traže strukturirani pristup učenju. Međutim, integracija naprednih tehnologija poput umjetne inteligencije i dodatnih interaktivnih elemenata mogla bi značajno unaprijediti ovu platformu i učiniti je još učinkovitijom.



Slika 1.4 Spellzone - primjer testa temeljenog na slušanju

### **Duolingo**

Duolingo je popularna online platforma za učenje jezika koja je dostupna kao web aplikacija i kao mobilna aplikacija za Android i iOS uređaje. Platforma je dizajnirana da bude intuitivna i jednostavna za korištenje, s naglaskom na igrifikaciju kako bi proces učenja bio zabavan i motivirajući. Kroz Duolingo, korisnici prolaze kroz niz lekcija koje pokrivaju različite aspekte jezika, uključujući vokabular, gramatiku, izgovor i slušanje. Svaka lekcija sastoji se od različitih zadataka kao što su prevođenje rečenica, slaganje riječi, ponavljanje izgovora i odgovaranje na pitanja s višestrukim izborom. Jedna od ključnih značajki Duolinga je sustav bodova i nagrada. Korisnici osvajaju bodove za točne odgovore i postizanje određenih ciljeva, što im omogućuje napredovanje kroz razine i otključavanje dodatnih sadržaja. Ovakav pristup potiče učenike da redovito vježbaju i ostanu motivirani.

Duolingo koristi umjetnu inteligenciju za prilagođavanje lekcija individualnim potrebama korisnika. AI analizira njihove odgovore i prilagođava težinu zadataka, fokusirajući se na područja gdje korisnici imaju poteškoće. Iako Duolingo pruža sveobuhvatan i interaktivan pristup učenju jezika, nedostaju mu specifični alati za učenje pravopisa. Za razliku od aplikacija koje su specifično dizajnirane za poboljšanje pravopisa, Duolingo se fokusira na cjelokupno učenje jezika, što znači da pravopis nije u prvom planu.

Slika 1.5 Duolingo - primjeri zadataka vezanih uz učenje pravopisa

### **Spelling Shed**

Spelling Shed je edukativna platforma namijenjena poboljšanju pravopisnih vještina kod učenika. Dostupna je kao web aplikacija i kao mobilna aplikacija za Android i iOS uređaje. Platforma nudi raznovrsne igre i zadatke koji pomažu učenicima da usvoje i vježbaju pravopis. Učitelji mogu kreirati vlastite liste riječi ili koristiti unaprijed definirane liste koje prate nacionalne kurikulume. Spelling Shed koristi različite metode za vježbanje, uključujući slaganje slova, ispunjavanje praznina i prepoznavanje riječi, čime osigurava sveobuhvatno pokrivanje pravopisnih pravila i izuzetaka. Spelling Shed-a koristi sustav bodova i rangiranja koji motivira učenike da redovito vježbaju. Učenici osvajaju bodove za točne odgovore i napredovanje kroz razine. Učitelji mogu pratiti napredak svojih učenika putem detaljnih izvještaja koji pokazuju njihova postignuća i područja koja zahtijevaju dodatnu pažnju. Spelling Shed također omogućuje učiteljima prilagođavanje težine zadataka prema individualnim potrebama učenika, što je ključno za adaptivnu nastavu. Učitelji mogu postaviti izazove i organizirati natjecanja unutar razreda ili škole, čime dodatno potiču učenike na vježbanje.



Slika 1.6 Spelling Shed - primjer „drag and drop“ zadatka za učenje pravopisa

# **2. Korištene tehnologije**

Slika na kojoj se prikazuje tekst, dijagram, snimka zaslona

Opis je automatski generiranOvaj rad obuhvaća razvoj aplikacije koja se oslanja na klijent-poslužitelj arhitekturu (Slika 2.1). Uvođenje ove arhitekture omogućuje efikasno rukovanje velikim količinama podataka i pruža robusno rješenje za različite korisničke zahtjeve. Primarni cilj rada je demonstrirati kako se moderne tehnologije mogu integrirati kako bi se postigla visoka razina funkcionalnosti i interaktivnosti. Odabrana je klijent-poslužitelj arhitektura zbog svoje pouzdanosti i skalabilnosti. Kroz ovo poglavlje, detaljno će se analizirati korištenje različitih tehnologija koje su međusobno integrirane kako bi se osigurala optimalna izvedba aplikacije. Također, rad će obuhvatiti i prikaz načina na koji su klijentska i poslužiteljska strana aplikacije povezane te kako baza podataka igra ulogu u pohrani i upravljanju podacima.

Slika 2.1 Arhitektura klijent - poslužitelj

Klijentska strana aplikacije razvijena je pomoću radnog okvira React. React je popularan JavaScript okvir koji omogućuje izgradnju dinamičnih i responzivnih korisničkih sučelja. Korištenjem komponenata, React omogućava modularan i efikasan razvoj aplikacije. React koristi virtualni DOM, što omogućuje brže ažuriranje sučelja i poboljšanu izvedbu. Njegova fleksibilnost i bogat ekosustav dodataka i alata čine ga idealnim izborom za razvoj modernih web aplikacija. HTML se koristi za strukturiranje sadržaja, dok CSS osigurava stilizaciju i vizualnu prezentaciju. JavaScript omogućava interakciju i dinamiku unutar aplikacije, stvarajući bogato korisničko iskustvo.

Poslužiteljska strana implementirana je pomoću Express-a, fleksibilnog web okvira za Node.js, koji omogućuje jednostavnu izradu robusnih aplikacija. Express je dizajniran za izradu robusnih i skalabilnih web aplikacija. Omogućava brzu i jednostavnu kreaciju RESTful API-ja i web poslužitelja. Express nudi niz gotovih funkcionalnosti i middleware-a, što pojednostavljuje rukovanje zahtjevima, odgovorima, sesijama i statičkim datotekama. Njegova jednostavnost i proširivost čine ga popularnim za razvoj poslužiteljskih aplikacija.

Za komunikaciju između klijentske i poslužiteljske strane koristi se Axios, HTTP klijent poznat po svojoj jednostavnosti i efikasnosti. Axios je promise-based HTTP klijent za JavaScript koji se koristi za slanje asinkronih HTTP zahtjeva. Podržava jednostavno slanje GET, POST, PUT, DELETE i drugih vrsta zahtjeva, te omogućava rad s JSON podacima bez potrebe za dodatnim konfiguracijama. On olakšava komunikaciju između klijentske i poslužiteljske strane aplikacije, omogućujući razmjenu podataka na brz i efikasan način.

Baza podataka aplikacije bazirana je na PostgreSQL-u, moćnom sustavu za upravljanje relacijskim bazama podataka koji osigurava pouzdano i učinkovito čuvanje podataka. PostgreSQL je moćan, open-source sustav za upravljanje relacijskim bazama podataka (RDBMS). Poznat je po svojoj pouzdanosti, skalabilnosti i podršci za napredne SQL funkcionalnosti. PostgreSQL, koji se temelji na ACID svojstvima, podržava kompleksne upite, transakcije, replikaciju i puno drugih naprednih značajki.

Osim osnovne funkcionalnosti, aplikacija je integrirana s nekoliko vanjskih API-ja kako bi se proširile njezine mogućnosti. Jedan od njih je ChatGPT API. To je usluga koju pruža OpenAI, omogućujući pristup moćnom modelu za obradu prirodnog jezika. API omogućava aplikacijama integraciju naprednih funkcionalnosti obrade jezika, poput generiranja teksta, odgovaranja na pitanja, vođenja dijaloga i još mnogo toga. Integracija s ChatGPT API-jem u sklopu ovog rada omogućava automatizirano generiranje različitih vrsta zadataka. Osim njega koriste se još ElevenLabs API i Ispravi.me API. ElevenLabs API pruža napredne zvučne funkcionalnosti, uključujući pretvaranje tekstualnih podataka u prirodan zvuk i analizu zvučnih podataka. Integracija ovog API-ja u radu omogućuje kreiranje digitalnih diktata, tj. pretvaranje teksta generiranog umjetnom inteligencijom u zvučni oblik koji se potom reproducira tijekom rješavanja digitalnih diktata. Ispravi.me API predstavlja alat za jezičnu provjeru koji automatizira ispravljanje pravopisnih i gramatičkih pogrešaka u tekstovima na hrvatskom jeziku. Uključivanje Ispravi.me API-ja omogućuje automatsku provjeru točnosti diktata te detekciju i ispravljanje eventualnih grešaka.

## **2.1 React.js**

React.js je popularna open-source JavaScript knjižnica za izgradnju korisničkih sučelja, prvenstveno za jednostrane aplikacije. Omogućuje programerima da kreiraju interaktivne i dinamične web stranice koristeći komponentni pristup, gdje se korisničko sučelje dijeli na male, ponovo upotrebljive dijelove koji se nazivaju komponente. Svaka komponenta može imati svoje stanje (state) i svojstva (props), čime se olakšava upravljanje podacima i ponašanjem unutar aplikacije. React koristi virtualni DOM (Document Object Model) kako bi optimizirao ažuriranja korisničkog sučelja. Kada se stanje aplikacije promijeni, React najprije ažurira virtualni DOM, zatim izračunava minimalni skup promjena potrebnih za stvarni DOM, i na kraju ažurira stvarni DOM s tim promjenama, što rezultira bržim i efikasnijim prikazom stranica.

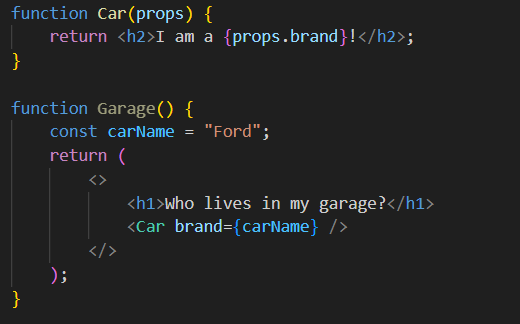
Jedna od glavnih prednosti React-a je njegova komponentna arhitektura koja omogućuje ponovnu upotrebu koda, što skraćuje vrijeme razvoja i olakšava održavanje aplikacija. Također, virtualni DOM osigurava visoke performanse i brze reakcije na promjene u korisničkom sučelju. React koristi JSX (JavaScript XML), sintaksu koja omogućuje pisanje HTML-a unutar JavaScript koda, što čini kod čitljivijim i lakšim za pisanje. Također, React se lako može integrirati s drugim bibliotekama i okvirima, kao što su Redux za upravljanje stanjem aplikacije ili React Router za upravljanje navigacijom unutar aplikacije.

U Reactu, State varijable i Props igraju ključne uloge u upravljanju podacima i komunikaciji između komponenti. State varijable su interno održavane varijable unutar komponente koje se koriste za spremanje i upravljanje podacima koji se mogu promijeniti tijekom vremena. One predstavljaju stanje komponente u određenom trenutku. Kada se state varijable promijene, React automatski re-renderira komponentu kako bi se reflektirale te promjene. State se može inicijalizirati u konstruktoru komponente i mijenjati pomoću setState() metode (Slika 2.2). To je posebno korisno za spremanje podataka kao što su korisnički unosi, stanja forme, ili bilo kakvih dinamičkih promjena u UI-u koji treba reflektirati stanje aplikacije.



Slika 2.2 Primjer promjene state varijable

Props (ili properties) su parametri koji se prosljeđuju komponenti i koriste se za prijenos podataka od roditeljske komponente prema djetetu. Props su read-only, što znači da se ne mogu mijenjati iz same komponente, već ih komponenta prima od svog roditelja i koristi ih za konfiguriranje svog ponašanja i izgleda (Slika 2.3). Props se koriste za dinamičko renderiranje komponenti na temelju vanjskih podataka ili stanja aplikacije. To omogućava komponentama da budu modularne i ponovno upotrebljive, jer mogu primiti različite podatke i ponašati se na različite načine ovisno o tim podacima. Primjerice, ako imamo roditeljsku komponentu koja predstavlja listu elemenata, ona može proslijediti svaki element kao prop svom djetetu (komponenti za prikaz pojedinog elementa), a zatim ta dječja komponenta koristi te propse za prikazivanje informacija o svakom pojedinom elementu.



Slika 2.3 Primjer prosljeđivanja propsa iz roditeljske u dječju komponentu

React se široko koristi u razvoju jednostranih aplikacija (SPA), gdje omogućuje dinamičko i interaktivno korisničko sučelje bez potrebe za osvježavanjem cijele stranice. Također se koristi u razvoju mobilnih aplikacija putem React Native-a, koji omogućuje razvoj mobilnih aplikacija koristeći istu React sintaksu. Osim toga, React se često koristi u razvoju PWA (Progressive Web Applications), koje kombiniraju najbolje značajke web i mobilnih aplikacija. Facebook je izdao proširenje za Chrome koje se može koristiti za uklanjanje pogrešaka u React aplikacijama. To čini proces otklanjanja pogrešaka React web aplikacija bržim i lakšim.

## **2.2 Node.js**

Node.js je open-source JavaScript runtime okruženje koje omogućuje izvršavanje JavaScript koda izvan web preglednika razvijeno za izgradnju skalabilnih mrežnih aplikacija s visokim performansama. Node.js pokreće V8 JavaScript mehanizam, jezgru Google Chromea, izvan preglednika [6]. To omogućuje Node.js da bude vrlo učinkovit. Aplikacija Node.js radi u jednom procesu, bez stvaranja nove niti za svaki zahtjev. Node.js pruža skup asinkronih I/O operacija u svojoj standardnoj biblioteci koje sprječavaju blokiranje JavaScript koda i općenito u Node.js su biblioteke napisane korištenjem ne blokirajućih paradigmi, čineći ponašanje blokiranja iznimkom, a ne normom [6]. Kada Node.js izvodi I/O operaciju, poput čitanja s mreže, pristupa bazi podataka ili datotečnom sustavu, umjesto blokiranja niti i trošenja CPU ciklusa na čekanje, Node.js će nastaviti s operacijama kada se vrati odgovor [6]. To mu omogućuje da obrađuje tisuće istodobnih veza s jednim poslužiteljem bez uvođenja tereta upravljanja istovremenošću niti, što bi moglo biti značajan izvor grešaka. Ključne komponente Node.js-a su:

* **Event-driven architecture**: Node.js koristi event petlju za upravljanje asinkronim operacijama. To omogućuje rukovanje mnogim istovremenim vezama bez blokiranja niti.
* **Non-blocking I/O**: Operacije ulaza/izlaza (I/O) u Node.js su asinkrone i ne blokiraju izvršavanje koda, što omogućuje efikasno rukovanje velikim brojem zahtjeva.

Node.js je stvorio Ryan Dahl 2009. godine. Razlog za njegovo stvaranje bio je pružiti bolje rješenje za razvoj aplikacija koje trebaju rukovati velikim brojem simultanih veza, posebno web aplikacija. Dahl je koristio Googleov V8 JavaScript engine zbog njegove brzine i efikasnosti. Neke od prednosti Node.js-a su:

* **Visoke performanse**: Zahvaljujući V8 engine-u i asinkronoj prirodi, Node.js može rukovati velikim brojem simultanih veza s malim opterećenjem.
* **Jednostrani jezik**: Programeri mogu koristiti JavaScript za razvoj cijelog projekta, kako na klijentskoj, tako i na poslužiteljskoj strani.
* **Velika zajednica i ekosustav**: Node Package Manager (npm) omogućuje pristup velikom broju modula i paketa koje je lako integrirati.
* **Skalabilnost**: Jednostavno je skalirati aplikacije horizontalno i vertikalno pomoću Node.js-a.

Node.js se koristi u različitim vrstama aplikacija, uključujući:

* **Web aplikacije**: Poslužiteljski dio aplikacija kao što su Express.js koristi Node.js za rukovanje HTTP zahtjevima.
* **API serveri**: Node.js je odličan za izgradnju RESTful API-ja zbog svoje brzine i efikasnosti.
* **Mikroservisne arhitekture**: Zbog svoje sposobnosti da skalira i upravlja mnogim simultanim vezama, Node.js je idealan za mikroservise.
* **IoT** (Internet of Things): Node.js se koristi za razvoj aplikacija koje komuniciraju s IoT uređajima.

Velika razlika između Node.js-a i preglednika je u tome što u Node.js-u programer kontrolira okolinu. To znači da se može napisati sav moderni ES2015+ JavaScript koji podržava verzija Node.js. Budući da se JavaScript razvija brzo, a preglednici mogu biti malo spori za nadogradnju, ponekad se na webu zaglavi s korištenjem starijih izdanja JavaScripta / ECMAScripta.

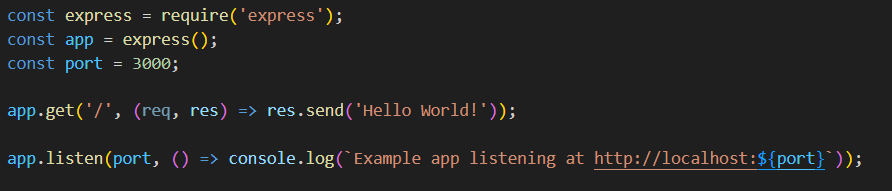
Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

Slika 2.4 Primjer Node.js servera koji ispisuje jednostavnu poruku

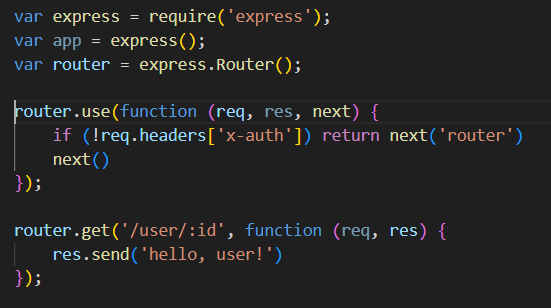
## **2.3 Express.js**

Express.js je vodeći web okvir za Node.js kreiran za razvoj web aplikacija i API-ja. Minimalistički je i fleksibilni web aplikacijski okvir za Node.js, koji pruža robusni set značajki za izgradnju web i mobilnih aplikacija. Pomaže u upravljanju različitim aspektima razvoja web aplikacija, kao što su usmjeravanje (routing), middleware, HTTP zahtjevi i odgovori. Express.js funkcionira kao sloj koji nadograđuje Node.js, omogućujući programerima da pišu čisti i organizirani kod za rukovanje zahtjevima i odgovorima. Nudi jednostavno API sučelje za definiranje ruta aplikacije i korištenje middleware funkcionalnosti. Middleware su funkcije koje imaju pristup zahtjevu (request), odgovoru (response) i sljedećoj middleware funkciji u aplikaciji, omogućujući modularnost i ponovno korištenje koda.

Express.js je razvio TJ Holowaychuk 2010. godine kao odgovor na potrebu za jednostavnim, ali moćnim okvirom za razvoj web aplikacija na Node.js platformi. Jedna od značajnih prednosti Express-a je rukovanje zahtjevima. U tradicionalnim web aplikacijama, web poslužitelj čeka na HTTP zahtjeve od klijenata. Kada primi HTTP zahtjev, poslužitelj odabire odgovarajući router i prenosi mu daljnju obradu tog zahtjeva. Pisanje upravljača ruta od nule u Node.js-u može biti prilično složeno, ali kod Express-a postoje metode za određivanje koja se funkcija poziva za određeni HTTP zahtjev (GET, POST, PUT itd.) i obrazac URL-a (Ruta).

Slika 2.5 Primjer upravljanja rutom "/" u Express.js

Express ovdje (Slika 2.5) određuje da će svi GET zahtjevi upućeni ruti "/" biti obrađeni funkcijom koja klijentu vraća odgovor "Hello World!". Osim rukovanja zahtjevima, odlična funkcionalnost Express-a je primjena međuprograma (middleware). Middleware omogućuje izvođenje operacija na zahtjevima i odgovorima koji se kreću kroz rute. Middleware se može primijeniti i na razini aplikacije i na razini rute, kao i biti povezan u lance. Bilo koji kompatibilni middleware se može ubaciti u lanac obrade zahtjeva, gotovo bilo kojim redoslijedom [7]. Tipična ruta obično se sastoji od nekog middleware-a i funkcije rukovanja rutom. Prikazan je Express router (Slika 2.6) koji primjenjuje middleware na svaki HTTP GET zahtjev upućen na rutu „/user/:id“ [7]. Zahtjev i odgovor prolaze middleware i na kraju dolaze do funkcije rukovanja rutom, koja završava ciklus HTTP zahtjev-odgovor vraćajući "hello, user!" prema HTTP klijentu [7].



Slika 2.6 Primjer middleware-a za GET zahtjev

Za web okvire se često govori da su "opinionated" ili "unopinionated". Okviri s mišljenjem (opinionated) su oni koji imaju mišljenje o "pravom načinu" rješavanja bilo kojeg određenog zadatka. Često podupiru brz razvoj u određenoj domeni jer je pravi način da se bilo što učini obično dobro shvaćen i dobro dokumentiran [8]. Međutim, oni mogu biti manje fleksibilni u rješavanju problema izvan svoje glavne domene i imaju tendenciju ponuditi manje izbora za komponente koje mogu koristiti [8]. Okviri bez mišljenja, nasuprot tome, imaju mnogo manje ograničenja u vezi s najboljim načinom spajanja komponenti kako bi se postigao cilj ili čak koje komponente treba koristiti [8]. Oni programerima olakšavaju korištenje najprikladnijih alata za dovršenje određenog zadatka, iako programeri sami moraju pronaći te komponente. Express je ''unopinionated'' što znači da se aplikaciju može strukturirati u jednoj datoteci ili u više datoteka, koristeći bilo koju strukturu direktorija, a ''unopinionated'' znači i korištenje middleware-a.

## **2.4 PostgreSQL**

PostgreSQL je objektno-relacijski sustav za upravljanje bazom podataka (ORDBMS) temeljen na POSTGRES-u, razvijen na Sveučilištu Kalifornija u Odsjeku za informatiku Berkeley [9]. POSTGRES je uveo mnoge koncepte koji su tek mnogo kasnije postali dostupni u nekim komercijalnim sustavima baza podataka. PostgreSQL je open-source potomak originalnog Berkeley koda [9]. Podržava veliki dio SQL standarda i nudi mnoge moderne značajke: složeni upiti, strani ključevi, okidači, prikazi koji se mogu ažurirati, transakcijski integritet, kontrola istodobnosti [9]. Osnovana na principima ACID (Atomicity, Consistency, Isolation, Durability), ova baza podataka osigurava pouzdanost, integritet i dosljednost podataka čak i u složenim okruženjima. PostgreSQL podržava širok raspon tipova podataka, naprednih SQL funkcionalnosti, pohrane proceduralnih jezika poput PL/pgSQL, Python, Perl, i mnoge druge mogućnosti koje omogućuju fleksibilno modeliranje podataka i implementaciju složenih poslovnih logika unutar baze podataka. PostgreSQL funkcionira kao server koji prima SQL upite od klijentskih aplikacija i izvršava ih na njegovim bazama podataka. Postavljanje i konfiguriranje PostgreSQL-a je relativno jednostavno, a korisnici mogu pristupiti bazi podataka putem različitih sučelja poput naredbenog retka, grafičkih korisničkih sučelja ili kroz aplikacijski kod putem raznih biblioteka.

Prednosti PostgreSQL-a uključuju visoku pouzdanost i robusnost, bogatstvo značajki za upravljanje podacima, proširivost, i podršku za kompleksne upite i transakcije. Njegova arhitektura omogućuje horizontalno i vertikalno skaliranje, što ga čini prikladnim za širok raspon aplikacija i poslovnih scenarija. Popis značajki PostgreSQL-a obuhvaća širok raspon funkcionalnosti, uključujući različite vrste podataka poput primitivnih, strukturiranih, dokumenata i geometrije, te mogućnosti osiguravanja integriteta podataka putem ograničenja i indeksiranja. Također, nudi napredne mogućnosti za upravljanje transakcijama, pouzdanost i oporavak od katastrofe, visoku razinu sigurnosti te proširivost putem pohranjenih funkcija, proceduralnih jezika i podrške za međunarodne skupove znakova. PostgreSQL se široko koristi, uključujući web aplikacije, mobilne aplikacije, poslovne sustave, geoprostorne informacijske sustave (GIS), analitiku podataka, e-trgovinu, financije i još mnogo toga. Koristi se kao centralno skladište podataka za web aplikacije, kao i za analitičke i izvještajne svrhe.

## **2.5 ChatGPT API**

ChatGPT API, sučelje za programiranje aplikacija koji je razvio OpenAI, omogućuje programerima integraciju naprednih mogućnosti ChatGPT-a u njihove aplikacije ili usluge. API omogućuje aplikacijama da koriste napredne funkcije poput razumijevanja jezika, generiranja teksta, odgovaranja na upite, vođenja razgovora i pružanja kontekstualnih informacija.

On funkcionira tako što koristi napredne tehnike obrade prirodnog jezika (NLP) i strojno učenje kako bi generirao odgovore na korisničke upite. Kada korisnik pošalje POST zahtjev putem API-ja, taj zahtjev obično uključuje tekstualni upit (prompt). API prima ovaj tekstualni unos i započinje proces obrade. Prije nego što se tekst proslijedi modelu, prolazi kroz fazu pretprocesiranja, što može uključivati normalizaciju teksta, tokenizaciju (razbijanje teksta na manje dijelove) i kodiranje [10]. Tokenizacija i kodiranje transformiraju tekst u format razumljiv modelu. Model je obučen na velikim količinama tekstualnih podataka kako bi razumio kontekst i generirao odgovarajuće odgovore. Model koristi transformersku arhitekturu koja se sastoji od slojeva mehanizama pažnje (attention mechanisms) [10]. Ovi slojevi omogućuju modelu da obrađuje informacije paralelno i efikasno. Mehanizmi pažnje omogućuju modelu da se fokusira na relevantne dijelove ulaznog teksta prilikom generiranja odgovora. Na temelju ulaznog teksta i konteksta, model generira odgovore predviđanjem sljedećih tokena u rečenici [10]. Proces se ponavlja dok se ne generira cijeli odgovor. Nakon što je odgovor generiran, prolazi kroz fazu postprocesiranja. Ovo može uključivati dekodiranje tokena natrag u tekstualni format, normalizaciju teksta i filtriranje neprikladnog sadržaja kako bi se osigurala kvaliteta i relevantnost odgovora [10]. Konačni odgovor se zatim vraća korisniku putem API-ja u JSON formatu.

Jednostavnost integracije je ključna prednost, jer je Chat GPT API dizajniran za jednostavnu integraciju u postojeće aplikacije. Primjer integracije je prikazan na slici 2.7. Skalabilnost je također značajna, budući da je API izgrađen za obradu velike količine zahtjeva, osiguravajući glatku izvedbu tijekom razdoblja najveće upotrebe. Fleksibilnost omogućava programerima prilagodbu ponašanja modela pružanjem uputa na razini sustava ili uvjetovanjem odgovora modela određenim korisničkim uputama. Iako Chat GPT API ima ogroman potencijal, pristrani odgovori mogu biti prisutni zbog pristranosti u podacima o obuci. Još jedno od ograničenja je i osjetljivost na kontekst što znači da odgovori modela uvelike ovise o kontekstu razgovora, a ako je kontekst izgubljen ili krivo shvaćen, generiranim odgovorima može nedostajati koherentnosti ili važnosti. Pogrešne informacije su također mogućnost, jer svaki model umjetne inteligencije može generirati netočne ili pogrešne informacije.

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

Slika 3.7 Primjer povezivanja Node.js servera s ChatGPT API-jem

## **2.6 ElevenLabs API**

Eleven Labs TTS API, sučelje za programiranje aplikacija, je sustav za sintezu govora (Text-to-Speech) koja koristi umjetnu inteligenciju kako bi pretvorila tekstualni sadržaj u prirodni, realistični govor. API se odlikuje glasovima visoke kvalitete, s preko 1000 glasova dostupnih na 29 jezika, prilagođenih za razne slučajeve upotrebe, pri brzini od 128 kbps [11]. Osigurava ultra-nisku latenciju, omogućujući vrijeme generiranja zvuka od ~400 ms pomoću Turbo modela [11]. Također, posjeduje kontekstualnu svijest, razumijevajući nijanse teksta kako bi osigurao odgovarajuću intonaciju i rezonanciju [11].

Da bi se koristio API, prvo se mora dobiti xi-api-ključ. Programer se registrira na platformi Eleven Labs i dobiva API ključ koji će koristiti za autentifikaciju. Programer šalje POST zahtjev na Eleven Labs TTS API krajnju točku, uključujući tekstualni sadržaj i potrebne parametre. Neki od tih posebnih parametara uključuju:

* **voice\_id**: Identifikator glasa koji će se koristiti. Može se koristiti <https://api.elevenlabs.io/v1/voices> za popis svih dostupnih glasova.
* **enable\_logging**: Kada je enable\_logging postavljen na false, koristi se potpuni način privatnosti za zahtjev. To znači da funkcije povijesti nisu dostupne za ovaj zahtjev, uključujući i sastavljanje zahtjeva. Potpuni način privatnosti mogu koristiti samo poslovni korisnici.
* **optimize\_streaming\_latency**: Omogućuje optimizaciju latencije uz određeni gubitak kvalitete. Ovo se može postaviti na različite razine od 0 do 4, pri čemu veće vrijednosti pružaju smanjenje latencije, ali s potencijalno većim gubitkom kvalitete.
* **output\_format**: Predstavlja format generiranog zvuka. Može biti jedan od navedenih formata, poput mp3\_44100\_128 koji je zadani format sa sample rate-om od 44.1kHz i brzinom prijenosa od 128kbps. Postoje i drugi formati, a dostupnost ovisi o pretplati.
* **voice\_settings.stability**: Ovaj parametar određuje stabilnost glasa. Stabilnost se odnosi na konzistentnost intonacije, brzine govora i izraza glasa. Veće vrijednosti ovog parametra rezultirat će stabilnijim glasom, dok će manje vrijednosti možda dovesti do varijacija u izgovoru.
* **voice\_settings.similarity\_boost**: Ovaj parametar kontrolira sličnost glasa s predloženim glasom. Povećanje ove vrijednosti rezultirat će većom sličnošću između generiranog glasa i odabranog glasa iz biblioteke glasova.
* **voice\_settings.style**: Omogućuje prilagodbu stila izgovora glasa. Može se postaviti na različite vrijednosti kako bi se postigao određeni ton, naglasak ili emocionalni izraz u glasu.
* **voice\_settings.use\_speaker\_boost**: Ovaj parametar određuje hoće li se koristiti dodatno pojačanje glasa kako bi se poboljšala jasnoća i snaga govora. Kada je postavljen na true, koristi se pojačanje glasa, dok će postavljanje na false rezultirati prirodnijim, ne pojačanim glasom.

## **2.7 Ispravi.me API**

Ispravi.me je korisničko sučelje aplikacije za računalnu provjeru pravopisa koja je do 2016. godine pod imenom Hascheck [Hašek] posluživala korisnike na adresi hascheck.tel.fer.hr [12]. Hascheck je kratica za Hrvatski akademski spelling checker. Radi se o jednoj od najstarijih internetskih usluga u Hrvatskoj koja u različitim oblicima djeluje kao javna i besplatna usluga pravopisne provjere teksta pisanog hrvatskim jezikom od proljeća 1994. godine do danas [12]. Hascheckova rječnička baza je strukturirana u tri osnovne kategorije:

* Hrvatski općejezični fond,
* Hrvatski posebnojezični fond (uključujući imena, nazive i slično),
* Engleski općejezični fond [12].

Ova baza podataka nije statična; kontinuirano se širi i unapređuje obradom novih tekstova, osobito onih s do tada nepoznatim riječima, često iz imenskog fonda. Klasični sustav za provjeru pravopisa stalno se poboljšava, uključujući nove mogućnosti kontekstualne analize. Iza ove usluge stoji ekspertni sustav koji stalno uči nove riječi iz pristiglih tekstova. Ovaj sustav učinkovito prepoznaje standardne pravopisne pogreške u hrvatskom jeziku, poput ije/je, č/ć, s/sa, zamijenjenih, ubačenih i izbrisanih slova. Na temelju naprednog statističkog modela hrvatskog jezika, nudi popis mogućih ispravaka za sumnjive riječi.. Nudi ispravljanje pogrešaka koje konvencionalni spellcheckeri ne prepoznaju – Hašek prepoznaje i nudi ispravke za česte stilske pogreške (na taj način vs. tako, obzirom na vs. s obzirom na to da) i pleonazme (čak štoviše, iz razloga što, besplatni poklon i slične) [12]. Sva komunikacija između korisnika i Hašeka šifrirana je kriptografskim ključevima. Za komunikaciju se koristi protokol HTTPS, a odgovor Hašeka dostavlja se u formatu JSON.

Ispravi.me API je aplikacijsko programsko sučelje koje omogućuje integraciju svih prethodno navedenih funkcionalnosti u vlastitu aplikaciju, čime omogućuje korištenje njegovih funkcionalnosti provjere pravopisa hrvatskog jezika izravno unutar aplikacije. Korisnik koji integrira Ispravi.me API trebao bi izvršiti nekoliko koraka kako bi uspješno koristio ovu uslugu. Prije nego što korisnik može koristiti API, trebao bi preko platforme Ispravi.me zatražiti i dobiti pristupne podatke odnosno API ključ. Ovi pristupni podaci omogućuju API-u identificiranje i praćenje korisnika prilikom korištenja usluge. Korisnik bi trebao implementirati funkcionalnost slanjem običnog POST zahtjeva prema resursu <https://ispravi.me/api/ispravi>, koristeći klasični POST umjesto AJAX-a. Ovaj zahtjev treba sadržavati tekst koji se provjerava i dodatne parametre. Ti dodatni parametri koje zahtjev sadrži su "context", koji može biti postavljen na "on" za kontekstualnu provjeru teksta ili na "off" za standardnu provjeru teksta (zadana vrijednost); "punctuation", koji može biti postavljen na "on" za provjeru razmaka ispred i iza znakova interpunkcije (zadana vrijednost) ili na "off" ako provjera razmaka nije potrebna; "showinfo", koji može biti postavljen na "on" za ispis dodatnih informativnih poruka vezanih uz svaku pogrešku, poput relevantnih pravopisnih savjeta (zadana vrijednost), ili na "off" ako ispis dodatnih poruka nije potreban; te "app", koji predstavlja jedinstveni identifikator korisnika usluge, odnosno API ključ. Nakon što obradi tekst, API vrati korisniku odgovor u formatu JSON.

# **3. Model podataka**

## **3.1 Relacijski model baze podataka**

Slika 4.1 Relacijski model baze podataka

## **3.2 Opis relacijskog modela podataka**

Baza podataka je strukturirana u trećoj normalnoj formi i sastoji se od 13 entiteta. Kako bi se omogućilo odvajanje uloga i prava pristupa, koriste se dva entiteta za korisnike. Za učenike postoji entitet UCENIK, koji pohranjuje osnovne informacije kao što su ime, prezime te šifra razreda kojeg pohađaju. Učenici pripadaju jednom razredu, a svaki razred je dio određene škole. Nastavnici su zabilježeni u entitetu UCITELJ, gdje se čuvaju podaci potrebni za prijavu u sustav, poput korisničkog imena i lozinke. Nastavnici mogu predavati u više razreda unutar jedne škole. Svaka škola je povezana s jednim mjestom, dok jedno mjesto može imati više škola, čime se osigurava konzistentna struktura podataka.

Nastavnici su odgovorni za kreiranje zadataka koji se mogu koristiti u različitim vježbama. Entitet ZADATAK sadrži ključne atribute kao što su vrsta zadatka, hint koji služi kao pomoć učeniku pri rješavanju zadatka te pitanje koje zadatak postavlja. Zadatak je specijaliziran, što znači da iz njega proizlaze podtipovi zadataka sa specifičnim svojstvima i atributima. Zadatak je specijaliziran na četiri podtipa:

1. DIKTAT, koji uključuje atribut audio – predstavlja zvučni zapis diktata koji se reproducira tijekom rješavanja zadatka.
2. NADOPUNA, s atributom unos – predstavlja generirani točan odgovor na pitanje kojeg je potrebno nadopunit.
3. ODABIR, koji sadrži dva dodatna atributa – ponuđeni učeniku prilikom rješavanja zadatka.
4. POVUCI I ISPUSTI (DRAG AND DROP), s atributom tablicaJSON – predstavlja generiranu tablicu u JSON formatu, sastavljenu od dva stupca s točnim riječima generiranim pomoću umjetne inteligencije.

Entitet VJEZBA ima atribut pokreni, tipa boolean, koji određuje hoće li vježba biti dostupna učenicima za rješavanje. Samo jedna vježba može biti aktivna u bilo kojem trenutku, što je označeno atributom postavljenim na true. Budući da se jedan zadatak može pojaviti u više vježbi, koristi se entitet VjezbaZadatak, koji povezuje zadatke s vježbama i omogućuje određivanje sastava svake vježbe.

Tijekom rješavanja vježbi i unosa odgovora, podaci se pohranjuju u tablicu LOGS. Ova tablica bilježi učenikove odgovore za određene zadatke i vježbe, kao i točne odgovore. Također se bilježe podaci o datumu i vremenu kreiranja zapisa te vremenu potrebnom učeniku za unos odgovora, izraženom u sekundama. Ova evidencija omogućuje praćenje napretka učenika i analizu učinkovitosti zadataka.

## **3.2 Opisi entiteta**

**MJESTO**

* **idMjesto** – šifra mjesta (PK)
* ime – naziv mjesta
* pBroj – poštanski broj

**SKOLA**

* **idSkola** – šifra škole (PK)
* ime – naziv škole
* idMjesto – šifra mjesta u kojem se škola nalazi (FK)

**RAZRED**

* **idRazred** – šifra razreda (PK)
* odjel – naziv odjeljenja
* idUcitelj – šifra učitelja koji vodi taj razred (FK)
* idSkola – šifra škole u kojoj se razred nalazi (FK)

**UCITELJ**

* **idUcitelj** – šifra učitelja (PK)
* ime – ime nastavnika
* prezime – prezime nastavnika
* korisnickoIme – korisničko ime za prijavu
* lozinka – lozinka za prijavu

**UCENIK**

* **idUcenik** – šifra učenika (PK)
* ime – ime učenika
* prezime – prezime učenika
* idRazred – šifra razreda kojeg učenik pohađa (FK)

**ZADATAK**

* **idZadatak** – šifra zadatka (PK)
* vrsta - vrsta zadatka (nadopuna, odabir, diktat, povuci i ispusti)
* pitanje – tekst pitanja zadatka
* hint – pomoć pri rješavanju zadatka

**VJEZBA**

* **idVjezba** – šifra vježbe (PK)
* naziv – naziv vježbe
* pokreni – zastavica koja određuje je li vježba postavljena za rješavanje

**VjezbaZadatak**

* **idVjezba** – šifra vježbe (PK)
* **idZadatak** – šifra zadatka (PK)

**DIKTAT**

* **idZadatak** – šifra zadatka (FK)
* audio – link na zvučni zapis diktata koji se koristi za reprodukciju

**NADOPUNA**

* **idZadatak** – šifra zadatka (FK)
* unos – generirani točan odgovor na pitanje kojeg je potrebno nadopunit

**ODABIR**

* **idZadatak** – šifra zadatka (FK)
* odgovor1 – ponuđeni točni odgovor učeniku pri rješavanju zadatka
* odgovor2 – ponuđeni netočni odgovor učeniku pri rješavanju zadatka

**POVUCI I ISPUSTI**

* **idZadatak** – šifra zadatka (FK)
* tablicaJSON – tablica sa 2 stupca i riječima za svaki stupac generirana za rješavanje zadatka u JSON formatu

**LOGS**

* **idLog** – šifra zapisa (PK)
* datumKreiranja – datum i vrijeme kreiranja zapisa u bazi podataka
* trajanjeUSec – vrijeme koje je bilo potrebno učeniku za rješavanje zadatka izraženo u sekundama
* ucenikovOdgovor – tekstualni odgovor učenika koji je unio pri rješavanju zadatka
* tocanOdgovor – ponuđeni točan odgovor za određeni zadatak
* tocan – zastavica koja određuje je li učenik određeni zadatak riješio točno ili netočno
* idUcenik – šifra učenika koji je riješio određeni zadatak na kojeg se zapis odnosi (FK)
* idZadatak – šifra zadatka na kojeg se zapis odnosi (FK)
* idVjezba – šifra vježba na koju se zapis odnosi (FK)

# **4. Opis aplikacije**

Pravopisnica je web aplikacija za učenje i vježbanje pravopisa, namijenjena učenicima nižih razreda osnovne škole. Aplikacija se sastoji od dva glavna sučelja: učenikovog i nastavnikovog. Učenikovo sučelje omogućuje učenicima rješavanje zadanih vježbi na interaktivan i pristupačan način. Kroz različite vrste zadataka, učenici mogu poboljšati svoje znanje pravopisa, uz pomoć sučelja koje potiče aktivno učenje. Nastavnikovo sučelje nudi niz funkcionalnosti za učitelje. Nastavnici mogu kreirati nove vježbe i zadatke, prilagođavajući ih specifičnim potrebama svojih učenika. Također mogu uređivati postojeće vježbe i zadatke, te imati pregled nad svim vježbama, zadacima i učenicima iz svojih razreda. Ova funkcionalnost omogućuje praćenje napretka učenika i prilagodbu sadržaja. Posebna funkcionalnost Pravopisnice, koja je čini jedinstvenom među sličnim aplikacijama, je digitalni diktat. Ova funkcionalnost omogućuje učiteljima da stvore zadatke sa zvučnim zapisom diktata čiji tekst je generirala umjetna inteligencija, a alat za pretvaranje teksta u govor pretvorio u zvučni zapis. Učenici mogu slušati ove zapise i potom zapisivati tekst, čime se vježbaju ne samo u pravopisu, nego i u slušnom razumijevanju.

Aplikacija je integrirana s tri različita vanjska API-ja, što značajno poboljšava njezinu učinkovitost. Ovi API-ji omogućuju automatizirano generiranje i ispravljanje zadataka, čime se štedi vrijeme nastavnicima i pruža preciznija povratna informacija. Integracija API-ja osigurava da aplikacija uvijek koristi najnovije tehnologije i metode za učenje pravopisa, čineći Pravopisnicu korisnim alatom u obrazovanju.

Prije početka korištenja aplikacije, potrebno je unijeti u bazu podataka informacije o mjestima, školama unutar tih mjesta, razredima unutar škola te nastavnicima koji predaju u tim razredima. Na početnoj stranici se nalazi forma za prijavu učenika pomoću padajućih izbornika (Slika 4.1). Nakon što učenik odabere svoje podatke tj. svoju školu, razred, i ime te potom klikne na „KRENI“ otvorit će mu se nova stranica za rješavanje vježbe koju je postavio nastavnik.

Slika na kojoj se prikazuje tekst, snimka zaslona, ptica, crtić

Opis je automatski generiran

Slika 5.1 Početna stranica

Ispod gumba "KRENI" nalazi se gumb koji vodi na stranicu za prijavu nastavnika. Nakon uspješne prijave, nastavnik je preusmjeren na svoju profilnu stranicu, gdje može vidjeti dosad izrađene zadatke, vježbe i popis učenika iz svojih razreda (Slika 4.2). Na lijevoj strani navigacijske trake nalazi se gumb "SIDEBAR". Klikom na njega, nastavniku se otvaraju mogućnosti za stvaranje novog zadatka, vježbe ili unosa novog učenika. Klikom na bilo koji od tih gumba, nastavnik će biti preusmjeren na odgovarajuću stranicu. Kako bi nastavniku bili prikazani njegovi učenici, mora izabrati jedan od svojih razreda iz padajućeg izbornika (Slika 4.3). Nakon odabira razreda, prikazat će se učenici iz tog razreda. Za svaki zadatak, vježbu i učenika dostupna je funkcionalnost uređivanja istog.

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, broj

Opis je automatski generiranSlika na kojoj se prikazuje tekst, snimka zaslona, softver, broj

Opis je automatski generiran

Slika 4.4 Prikaz učenika na profilnoj stranici s odabranim razredom

Slika 4.3 Nastavnikova profilna stranica s otvorenim sidebar-om

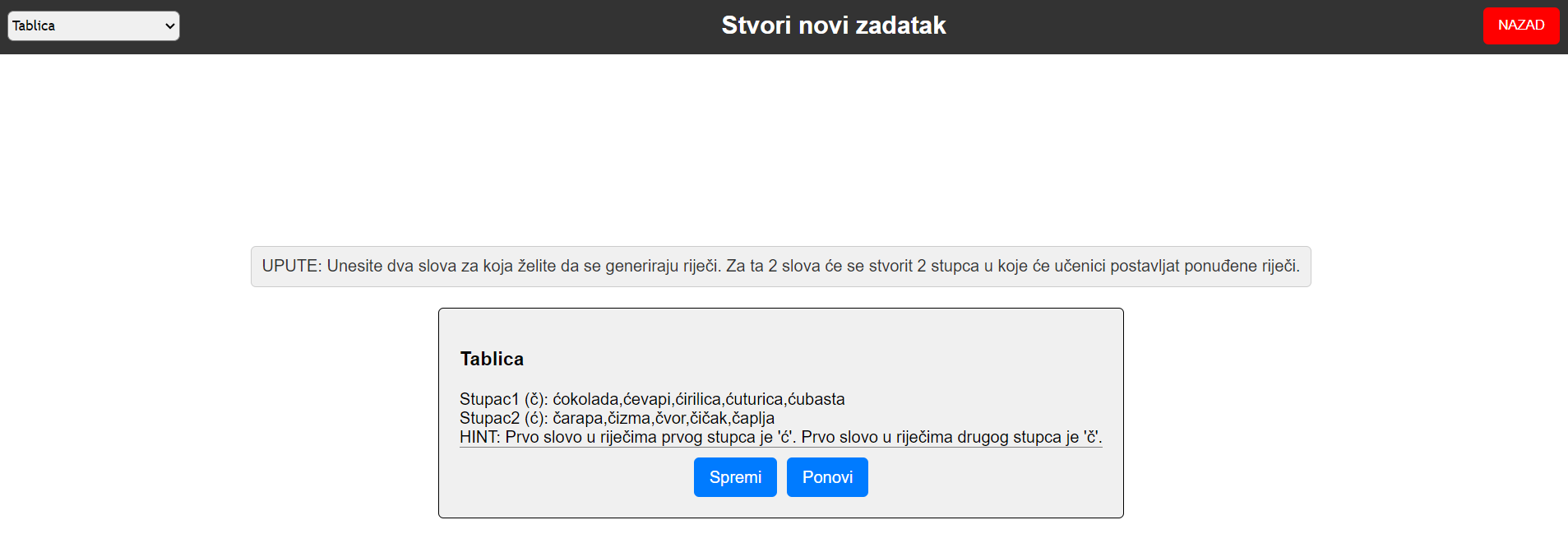
Pri stvaranju novih zadataka, nastavnik mora odabrati vrstu zadatka iz padajućeg izbornika (diktat, nadopuna, odabir, drag and drop). Ovisno o odabranoj vrsti zadatka, prikazat će se odgovarajuća komponenta. Za zadatke vrste diktat, tekst za diktat generira umjetna inteligencija. Nastavnik treba unijeti upit u predviđeno polje i kliknuti na gumb "Submit" (Slika 4.5). Klikom na gumb, upit se šalje umjetnoj inteligenciji putem API-ja, koja generira tekst i vraća ga korisniku. Nakon vraćenog odgovora, prikazuje se gumb "Generiraj audio". Klikom na taj gumb, generirani tekst se šalje na ElevenLabs API, koji pretvara tekst u zvučni zapis i vraća ga korisniku u traženom formatu (Slika 4.6). Nastavnik zatim može kliknuti na gumb "Play" kako bi reproducirao zvučni zapis kako bi provjerio je li sve u redu, s mogućnošću premotavanja i pauziranja zvučnog zapisa. Ako nastavnik nije zadovoljan s generiranim diktatom može krenuti ispočetka klikom na gumb „Ponovi“. U slučaju da je sve ispravno, klikom na gumb "Spremi" zadatak će biti pohranjen u bazu podataka, a nastavnik će Slika na kojoj se prikazuje tekst, softver, Font, web-stranica

Opis je automatski generiranSlika na kojoj se prikazuje tekst, snimka zaslona, softver, web-stranica

Opis je automatski generiranbiti preusmjeren na svoju profilnu stranicu.

Slika 4.6 Stvaranje digitalnog diktata - primjer generiranja zvučnog zapisa

Slika 4.5 Stvaranje digitalnog diktata - primjer slanja upita AI

Pri stvaranju zadatka vrste tablica, nastavniku će se prikazati komponenta koja traži da unese dva slova u dva prazna polja. Umjetna inteligencija zatim generira tablicu s dva stupca, pri čemu svaki stupac je jedno od tih slova (Slika 4.7). Umjetna inteligencija generira riječi koje sadrže ta slova, a učenici će kasnije moći te riječi razvrstavati u odgovarajuće stupce koristeći funkcionalnost drag and drop. Nastavniku će biti prikazani stupci s odgovarajućim riječima, kao i tekstualni hint koji pomaže učeniku pri rješavanju zadatka. Ako nastavnik nije zadovoljan generiranom tablicom ili hintom, može kliknuti na gumb "Ponovi" kako bi ponovo poslao upit umjetnoj inteligenciji za novu generaciju sadržaja. Ako je zadovoljan, klikom na gumb "Spremi" zadatak će biti pohranjen u bazu podataka, a nastavnik će biti preusmjeren na svoju profilnu stranicu.

Slika 4.7 Primjer generirane tablice od strane umjetne inteligencije