

Sadržaj

Uvod	1
1. Pregled područja i slične aplikacije	3
1.1 Tehnologija u obrazovanju	3
1.2 Upotreba tehnologije u poučavanju pravopisa	5
1.3 Slične aplikacije i sustavi	7
1.3.1 Grammarly	8
1.3.2 Spellzone	10
1.3.3 Duolingo	12
1.3.4 Spelling Shed	13
2. Korištene tehnologije	14
2.1 React.js	16
2.2 Node.js	19
2.3 Express.js	21
2.4 PostgreSQL	23
2.5 ChatGPT API	24
2.6 ElevenLabs API	25
2.7 Ispravi.me API	27
3. Model podataka	29
3.1 Relacijski model baze podataka	29
3.2 Opis relacijskog modela podataka	29
3.2 Opisi entiteta	30
4. Opis aplikacije	34
Zaključak	44
Literatura	46

Sažetak49

Summary.....50

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 podučavanju 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 automatiziranu 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 učenje pravopisa, 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 ispravljanje pravopisa, kao što je Grammarly, nudeći funkcije kao što su evidencija čestih pravopisnih pogrešaka, 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 temeljita 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 što je integracija sa sustavom Ispravi.me. Na kraju rada daju se smjernice za daljnji razvoj i poboljšanje sustava, te zaključak.

1. Pregled područja i slične aplikacije

1.1 Tehnologija u obrazovanju

Tehnologija u obrazovanju odnosi se na upotrebu digitalnih alata, softvera i online resursa za poboljšanje i podršku obrazovnog procesa [1]. U današnje vrijeme, teško je zamisliti formalno i neformalno obrazovanje bez upotrebe računala i novih tehnologija. Računala i internet postali su neizostavan dio svakodnevnog života učenika i nastavnika, koristeći 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, obrazovne aplikacije i platforme za učenje na daljinu. Razvoj tehnologije omogućava korištenje raznih multimedijских elemenata – zvuka, videa, simulacija i animacija, teksta i slike. Važno je pažljivo odabrati alate i multimedijски softver koji se koristi u obrazovanju, 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].
2. **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 [2].
3. **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].

4. **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 [3].
5. **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 [3].

Mane tehnologije u obrazovanju:

1. **Tehnološka nejednakost:** Pristup tehnologiji nije jednak za sve učenike. Socioekonomski status može utjecati na dostupnost digitalnih uređaja i interneta [4].
2. **Distrakcije:** Korištenje digitalnih uređaja u obrazovanju može dovesti do distrakcija. Učenici mogu biti skloni odvlačenju pažnje na društvenim mrežama, igrama i sl. [4].
3. **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 [4].
4. **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 [5].
5. **Tehnički problemi:** Tehničke poteškoće, poput problema s internetskom vezom, kvarova uređaja ili bugova.

Multidisciplinarni pristup razvoju edukativnih materijala je ključan, budući da nastavnici nisu nužno informatički stručnjaci. Projektni pristup razvoju materijala i uloga koje sudjeluju u tom procesu također su važni aspekti [6]. U literaturi i javnosti su se ustalili novi termini kao što su online učenje (eng. online learning), e-učenje (eng. e-learning), učenje utemeljeno na webu (eng. web-based learning), te učenje putem mobilnih uređaja (eng. m-learning), što je trend koji se sve više širi u novije vrijeme [6].

Razvoj informacijsko-komunikacijske tehnologije mijenja paradigmu učenja i obrazovanja primjenom suvremenih metoda i oblika nastave. Suvremena se nastava temelji na didaktičkom četverokutu koji uključuje tehnologiju, učenika, nastavnika i nastavni sadržaj [7].

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 [8]. Na primjer, AI sustavi za ocjenjivanje učenika mogu preferirati određene stilove pisanja ili izraza na temelju prethodnih podataka, što može rezultirati nejednakim procjenama [8]. 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 [8]. Nedostatak transparentnosti može dovesti do nedostatka povjerenja u AI sustave i rezultate koje oni proizvode [8].

1.2 Upotreba tehnologije u poučavanju pravopisa

Većina predmeta se više ne podučava na „staromodan“ način koristeći samo kedu i ploču i repetitivno usvajanje znanja, a postoje i dobri razlozi za to. Učenje pravopisa kroz tradicionalne metode može se činiti ponavljajućim i dosadnim dok upotreba tehnologije u učenju pravopisa može poboljšati proces obrazovanja kroz interaktivno, ciljano i personalizirano podučavanje. Primjena tehnologije u podučavanju pravopisa daje brze informacije učiteljima o riješenosti zadataka, mogućnost za izmjenu opcija kod zadataka kako bi se specifično prilagodili učenicima, kao i praćenje pojedinaca tijekom određenog vremenskog perioda. Učenici se kroz igre, kvizove i digitalne diktate mogu i zainteresirati za učenje pravopisa. Osim toga, nastavnici mogu koristiti tehnologiju da bi lakše, brže i

jednostavnije ispravili zadatke. Kao rezultat toga nastavnici bi dobili više vremena za rad s učenicima, a također bili oslobođeni administrativnih poslova.

Korištenje tehnologije u podučavanju pravopisa donosi niz konkretnih prednosti koje specifično unapređuju proces učenja pravopisnih pravila. Aplikacije i softveri za učenje pravopisa kao što je recimo SpellingCity omogućuju personalizaciju zadataka prema individualnim potrebama učenika, čime se prilagođavaju različitim stilovima učenja i tempom napredovanja [9]. Interaktivne metode učenja pravopisa putem tehnologije omogućuju trenutne povratne informacije. Kada učenik napravi grešku, digitalni alat mu odmah pruža objašnjenje i ispravke, što pomaže učeniku da odmah nauči ispravan pravopisni oblik riječi. Jedna od igara u SpellingCity aplikaciji je „Hang Mouse“. Ova igra je slična klasičnoj igri „Vješala“. U njoj učenici trebaju pogoditi pravopis riječi slušajući njen izgovor i odabirući ispravna slova. Svaki netočan izbor približava miša opasnosti mačke, čime se dodaje element zabave i izazova, a i korisniku se odmah prikaže njegova greška.

Osim trenutne povratne informacije, tehnologija omogućuje široku dostupnost resursa za učenje pravopisa. Digitalni rječnici, gramatički vodiči i interaktivne lekcije pomažu učenicima da lakše savladaju pravopisna pravila. Korištenjem digitalnih rječnika koji imaju brzu pretragu i mogućnost prikaza definicija, pravopisnih varijacija i primjera upotrebe riječi omogućuje se korisnicima da brzo pronađu pravilan pravopis i značenje riječi. Također su dostupni putem interneta non-stop, što omogućuje učenicima da uče i vježbaju pravopis u bilo koje vrijeme i na bilo kojem mjestu.

Prednosti upotrebe tehnologije uključuju i podršku za učenike s posebnim potrebama, omogućavajući im pristup prilagođenim obrazovnim sadržajima koji odgovaraju njihovim specifičnim potrebama. Tako učenici s disleksijom mogu koristiti specijalizirane aplikacije koje olakšavaju učenje pravopisa putem vizualnih i auditivnih pomagala. Tehnologije kao što su softver za prepoznavanje glasa, programi za predviđanje teksta i druge asistivne aplikacije pružaju potrebnu podršku koja omogućava ovim učenicima da efikasnije savladaju pravopis i razviju vještine pisanja [10]. Na primjer, softver za prepoznavanje glasa omogućava učenicima da diktiraju tekst koji se automatski pretvara u pisani oblik, što je posebno korisno za one s motoričkim poteškoćama ili disleksijom [10]. Programi za predviđanje teksta sugeriraju riječi dok učenici pišu, smanjujući broj pravopisnih grešaka i olakšavajući ispravnu upotrebu riječi [10].

Elektronski rječnici omogućuju brzo pronalaženje ispravnog pravopisa i sinonima, što pomaže u proširenju vokabulara i poboljšanju pisanja [10].

Općenito, pravopis predstavlja ključni aspekt pismenosti koji uključuje razne kognitivne procese. Kroz godine, testirane su različite metode podučavanja pravopisa, s različitim uspjehom. Od prvog uvođenja računala u škole, jedno od popularnih pitanja je bilo je li korištenje tehnologije bolje od korištenja papira za učenje vještina kao što je pravopis [11]. Velik broj radova bavio se istraživanjem ove teme u proteklih 20 godina [11]. U novije vrijeme, provedeno je mnogo studija o upotrebi aplikacija koje pomažu u 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 [11].

1.3 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 ispravljanje 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, Spellzone se često oslanja na tradicionalne metode učenja. To znači da nema integriranu umjetnu inteligenciju za stvaranje 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.

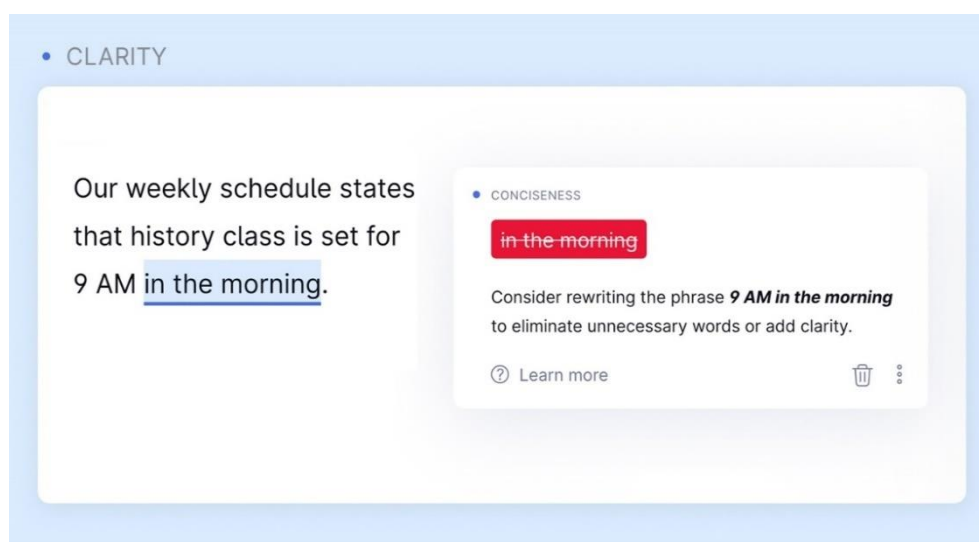
Dodatno, nedostaju im alati ili sučelje namijenjeni nastavnicima. Sučelje bi omogućilo praćenje napretka učenika, prilagođavanje lekcija prema njihovim potrebama i pružanje dodatne podrške. Isto tako, bilo bi moguće omogućiti učenicima praktičnu vježbu slušanja i pisanja kroz digitalni diktat. Ni jedna 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.

1.3.1 Grammarly

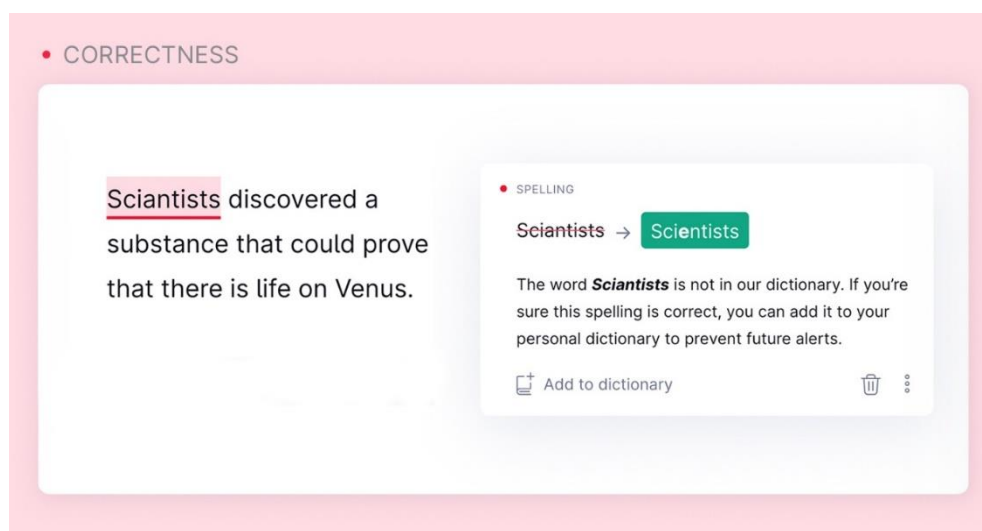
Grammarly je platforma za poboljšanje pisanja koja koristi umjetnu inteligenciju za ispravljanje pravopisnih, gramatičkih i stilskih pogrešaka. Dostupna je kao web aplikacija, proširenje za preglednike te dodatak za Microsoft Office i mobilne uređaje. Grammarly pomaže korisnicima da pišu s više točnosti i jasnoće u raznim kontekstima, uključujući poslovnu komunikaciju, akademsko pisanje i svakodnevnu upotrebu [12]. 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 [12]. 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 [12]. 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 1.2 Grammarly - Prijedlozi za nejasne rečenice i suvišne fraze

1.3.2 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 [13]. Spellzone je posebno koristan alat za nastavnike i roditelje, jer omogućuje praćenje napretka učenika i prilagođavanje sadržaja njihovim specifičnim potrebama [13]. Neke od značajki Spellzone-a su:

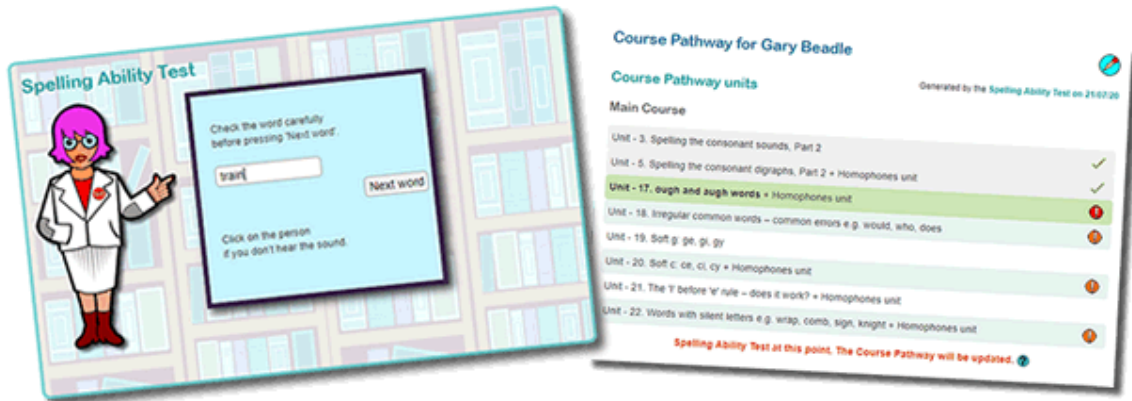
Lekcije i vježbe: struktura lekcija i vježbi su osmišljene kako bi postupno uvodile učenike u različite aspekte pravopisa, omogućujući im da napreduju vlastitim tempom.

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

Resursi za nastavnike: Spellzone pruža nastavnicima 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 pravopisnih preporuka, što može biti značajan nedostatak u odnosu na platforme poput Grammarly. Korisnici započinju s registracijom na Spellzone web stranici, gdje se mogu prijaviti kao učenici, nastavnici ili roditelji. Nakon prijave, dolaze na intuitivnu nadzornu ploču (eng. dashboard) koja omogućava jednostavnu navigaciju kroz različite module i alate. Platforma nudi razne lekcije i vježbe koje su organizirane prema razinama težine i prema 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. Nastavnici 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.



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 privlačnijom.

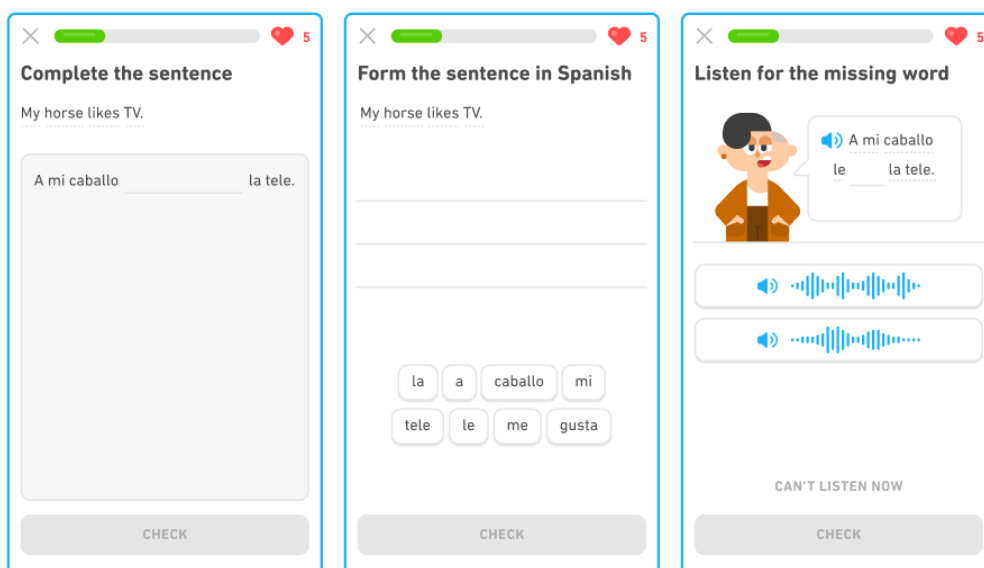


Slika 1.4 Spellzone - primjer testa temeljenog na slušanju

1.3.3 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. Umjetna inteligencija 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

1.3.4 Spelling Shed

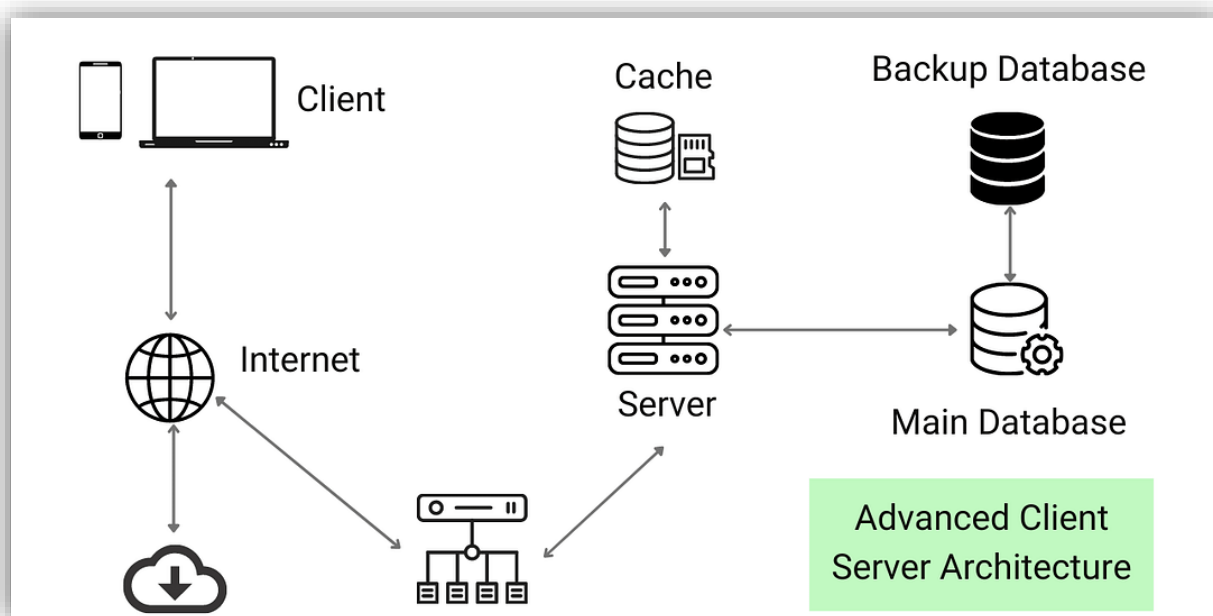
Spelling Shed je edukativna platforma namijenjena poboljšanju pravopisnih vještina kod učenika. Dostupna je kao web aplikacija i kao mobilna aplikacija za Android i iOS uređaje. Platforma nudi raznovrsne igre i zadatke koji pomažu učenicima da usvoje i vježbaju pravopis. Nastavnici mogu kreirati vlastite liste riječi ili koristiti unaprijed definirane liste koje prate odabrane važeće državne kurikulare. 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 koristi sustav bodova i rangiranja koji motivira učenike da redovito vježbaju. Učenici osvajaju bodove za točne odgovore i napredovanje kroz razine. Nastavnici mogu pratiti napredak svojih učenika putem detaljnih izvještaja koji pokazuju njihova postignuća i područja koja zahtijevaju dodatnu pažnju. Spelling Shed također omogućuje nastavnicima prilagođavanje težine zadataka prema individualnim potrebama učenika, što je ključno za adaptivnu nastavu. Nastavnici 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

Ovaj 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 [14]

Klijentska strana aplikacije razvijena je pomoću radnog okvira React. React je popularan JavaScript okvir koji omogućuje izgradnju dinamičkih i responzivnih korisničkih sučelja. Korištenjem komponenata, React omogućava modularan i efikasan razvoj aplikacije. React koristi virtualni DOM (eng. Document Object Model), što omogućuje brže ažuriranje sučelja i poboljšanu izvedbu. Njegova fleksibilnost i bogat sustav 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. On 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 [15]. Podržava jednostavno slanje GET, POST, PUT, DELETE i drugih vrsta zahtjeva, te omogućava rad s JSON podacima bez potrebe za dodatnim konfiguracijama [15]. 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, sustavu za upravljanje relacijskim bazama podataka (eng. RDBMS) koji osigurava pouzdano i učinkovito čuvanje podataka. Pouzdanost je osigurana tako što PostgreSQL podržava transakcije koje osiguravaju konzistentnost podataka, te zato što pruža ACID (Atomicity, Consistency, Isolation, Durability) svojstva transakcija. Osim toga, ima napredne mehanizme za oporavak podataka. U slučaju kvara može vratiti bazu podataka u stabilno stanje bez gubitka podataka. PostgreSQL je optimiziran za izvođenje složenih upita i podržava indekse, povećavajući tako učinkovitost pretraživanja i manipulacije podacima. Poznat je i po svojoj 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 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 (eng. SPA). 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 (eng. state) i svojstva (eng. props), čime se olakšava upravljanje podacima i ponašanje unutar aplikacije. React koristi virtualni DOM 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.

```
class Car extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      brand: "Ford",
      model: "Mustang",
      color: "red",
      year: 1964,
    };
  }
  render() {
    return (
      <div>
        <h1>My {this.state.brand}</h1>
        <p>
          It is a {this.state.color}
            {this.state.model}
            from {this.state.year}.
        </p>
      </div>
    );
  }
}
```

Slika 2.2 Primjer promjene state varijable [16]

Props (ili properties) su parametri koji se proslijeđuju 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.

```
function Car(props) {  
  return <h2>I am a {props.brand}!</h2>;  
}  
  
function Garage() {  
  const carName = "Ford";  
  return (  
    <>  
      <h1>Who lives in my garage?</h1>  
      <Car brand={carName} />  
    </>  
  );  
}
```

Slika 2.3 Primjer proslijeđivanja propsa iz roditeljske u dječju komponentu [17]

React se široko koristi u razvoju jednostranih aplikacija (eng. 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 (eng. Progressive Web Applications), koje kombiniraju najbolje značajke web i mobilnih aplikacija.

2.2 Node.js

Node.js je open-source JavaScript runtime okruženje koje omogućuje izvršavanje JavaScript koda izvan web preglednika. Razvijeno je za izgradnju skalabilnih mrežnih aplikacija s visokim performansama. Node.js koristi V8 JavaScript engine, koji je također jezgra Google Chromea, ali radi izvan preglednika i to čini Node.js izuzetno učinkovitim [18]. Node.js aplikacije rade unutar jednog procesa, bez potrebe za stvaranjem novih niti za svaki zahtjev. Node.js standardna biblioteka pruža asinkrone I/O operacije koje sprječavaju blokiranje JavaScript koda. Većina biblioteka u Node.js-u napisana je koristeći ne-blokirajuće paradigme, pa je tako blokiranje iznimka, a ne pravilo [18]. Kada Node.js izvodi I/O operaciju, poput mrežnog čitanja, pristupa bazi podataka te ne blokira nit i ne troši CPU čekajući cikluse. Umjesto toga, Node.js nastavlja s drugim operacijama i vraća se na I/O zadatak kada stigne odgovor [18]. Ovo mu omogućuje obradu tisuća istodobnih veza putem jednog poslužitelja, bez opterećenja upravljanjem nitima, što može biti značajan izvor pogrešaka. 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:** I/O operacije 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** (eng. 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.

```
const { createServer } = require('node:http');

const hostname = '127.0.0.1';
const port = 3000;

const server = createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Slika 2.4 Primjer Node.js servera koji ispisuje jednostavnu poruku [18]

2.3 Express.js

Express.js je vodeći web okvir za Node.js kreiran za razvoj web aplikacija i API-ja. To je minimalistički 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 (eng. 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 (eng. request), odgovoru (eng. 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 (eng. router) 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).

```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => res.send('Hello World!'));

app.listen(port, () => console.log(`Example app listening at http://localhost:${port}`));
```

Slika 2.5 Primjer upravljanja rutom "/" u Express.js [19]

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 (eng.

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 [19]. Bilo koji kompatibilni middleware se može ubaciti u lanac obrade zahtjeva, gotovo bilo kojim redoslijedom [19]. 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“ [19]. 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 [19].

```
var express = require('express');
var app = express();
var router = express.Router();

router.use(function (req, res, next) {
  if (!req.headers['x-auth']) return next('router')
  next()
});

router.get('/user/:id', function (req, res) {
  res.send('hello, user!')
});
```

Slika 2.6 Primjer middleware-a za GET zahtjev [19]

Za web okvire se često govori da su "opinionated" ili "unopinionated". Okviri s mišljenjem (eng. opinionated) su oni koji imaju mišljenje o "pravom načinu" rješavanja bilo kojeg određenog zadatka [20]. Često potiču brz razvoj u specifičnom području jer predstavljaju provjeren i dobro dokumentiran pristup [20]. Međutim, mogu biti manje fleksibilni u rješavanju problema izvan svoje primarne domene i često nude ograničene opcije za komponente koje se mogu koristiti [20]. 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 [20]. Oni olakšavaju programerima 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 relacijski sustav za upravljanje bazama podataka (eng. RDBMS) temeljen na POSTGRES-u, koji je razvijen na Sveučilištu Kalifornija, Berkeley [21]. POSTGRES je uveo mnoge koncepte koji su kasnije postali standardni u mnogim komercijalnim sustavima baza podataka. PostgreSQL je open-source nasljednik originalnog Berkeley koda [21]. Podržava veliki dio SQL standarda i nudi mnoge moderne značajke: složeni upiti, strani ključevi, okidači, prikazi koji se mogu ažurirati, transakcijski integritet, kontrola istodobnosti [21]. 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.

API funkcionira tako što koristi napredne tehnike obrade prirodnog jezika (eng. 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 (eng. 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 [22]. 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 (eng. attention mechanisms) [22]. 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 [22]. Proces se ponavlja dok se ne generira cijeli odgovor. Nakon što je odgovor generiran, prolazi kroz fazu postprocesiranja [22]. Ovo može uključivati dekodiranje tokena natrag u tekstualni format, normalizaciju teksta i filtriranje neprikladnog sadržaja kako bi se osigurala kvaliteta i relevantnost odgovora [22]. 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, čime se osigurava glatka izvedba tijekom razdoblja najveće upotrebe. Fleksibilnost omogućava programerima prilagodbu ponašanja modela pružanjem uputa na razini sustava ili uvjetovanjem odgovora modela određenim korisničkim uputama. Iako ChatGPT 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, generirani odgovori mogu izgubiti na važnosti. Pogrešne informacije su također moguće, jer svaki model umjetne inteligencije može generirati netočne ili pogrešne informacije.

```
const { OpenAI } = require('openai');

const openai = new OpenAI({
  apiKey: process.env.OPENAI_API_KEY,
  organization: 'org-ayfGn12VUYoLFy88Eb2QaJZ6',
});

const response = await openai.chat.completions.create({
  messages: [{
    role: "system",
    content: prompt
  }],
  model: "gpt-3.5-turbo",
});

console.log(response.choices[0]);

return res.status(200).json({
  success: true,
  result: response.choices[0].message.content,
});
```

Slika 2.7 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 (eng. Text-to-Speech) koji koristi umjetnu inteligenciju kako bi pretvorio 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 kvaliteti od 128 kbps [23]. Osigurava ultra-nisku latenciju, omogućujući vrijeme generiranja zvuka od ~400 ms pomoću Turbo modela [23]. Također, posjeduje kontekstualnu svijest, razumijevajući nijanse teksta kako bi osigurao odgovarajuću intonaciju [23].

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 bitrate-om 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, nepojačanim glasom.

2.7 Ispravi.me API

Ispravi.me je aplikacija za računalnu provjeru pravopisa, koja je do 2016. godine bila poznata kao Hascheck [24]. Hascheck je kratica za Hrvatski akademski spelling checker. Ovo je jedna od internetskih usluga u Hrvatskoj, koja od 1994. godine besplatno nudi pravopisnu provjeru tekstova na hrvatskom jeziku i to u različitim oblicima [24]. Hascheckova rječnička baza je strukturirana u tri osnovne kategorije:

- Hrvatski općejezični fond,
- Hrvatski posebnojezični fond (uključujući imena, nazive i slično),
- Engleski općejezični fond [24].

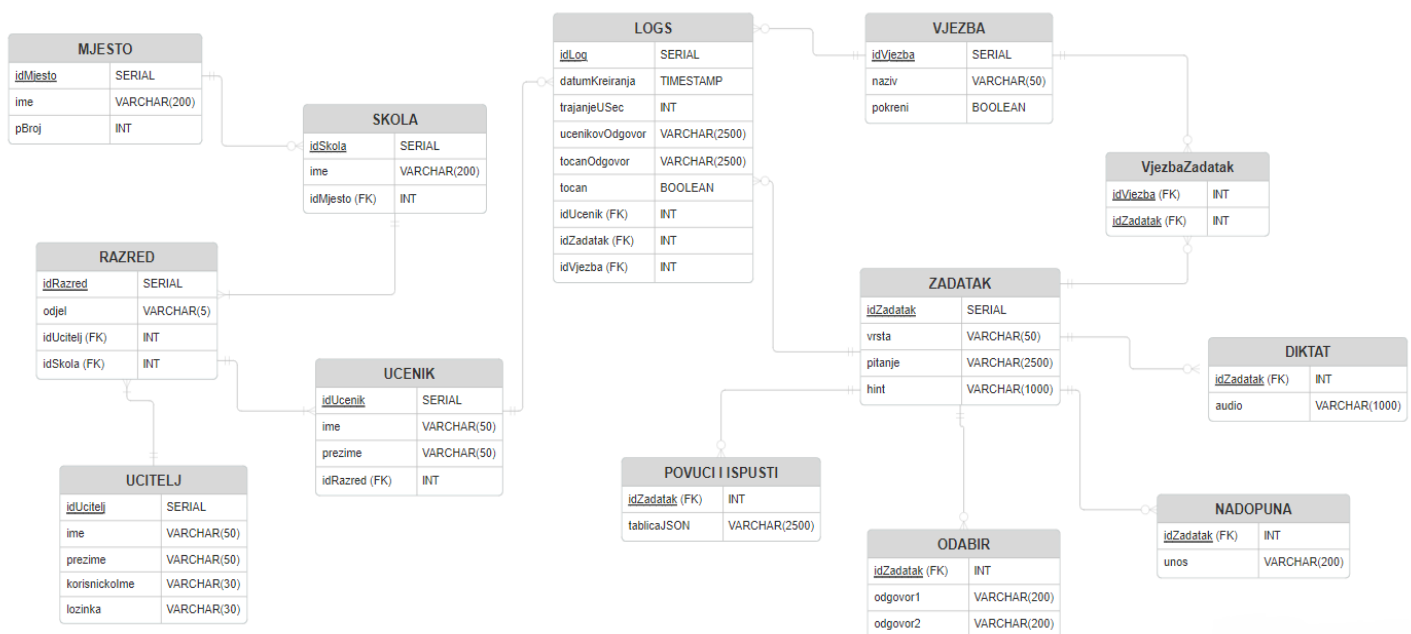
Analizom novih tekstova, a posebno onih s dotad nepoznatim riječima često iz imenskog fonda, ova baza podataka se neprestano razvija i nadopunjuje [24]. Tradicionalni sustav za provjeru pravopisa stalno se unapređuje, uključujući nove opcije za kontekstualnu analizu. Iza ove usluge stoji ekspertni sustav koji stalno uči nove riječi iz pristiglih tekstova i koji uspješno prepoznaje uobičajene pravopisne pogreške u hrvatskom jeziku, kao što su ije/je, č/ć, s/sa, te zamijenjena, dodana i izbrisana slova [24]. Koristeći napredni statistički model hrvatskog jezika, nudi popis mogućih ispravaka za sumnjive riječi. Ovaj sustav ispravlja pogreške koje standardni spellchecker pravopisa često propuštaju – Hašek prepoznaje i nudi ispravke za česte stilske greške (poput "na taj način" umjesto "tako", "obzirom na" umjesto "s obzirom na to da") i pleonazme (poput "čak štoviše", "iz razloga što", "besplatni poklon" i slične) [24]. Komunikacija između korisnika i Hašeka je šifrirana

kriptografskim ključevima, a za komunikaciju se koristi HTTPS protokol dok se odgovori Hašeka dostavljaju u JSON formatu.

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-ju 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 3.1 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 attribute 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 kojeg je potrebno nadopuniti u zadatku.
3. ODABIR, koji sadrži dva dodatna atributa – ponuđeni učeniku prilikom rješavanja zadatka.
4. POVUCI I ISPUSTI (eng. DRAG AND DROP), s atributom tablicaJSON – predstavlja generiranu tablicu u JSON formatu, sastavljenu od dva stupca zajedno sa 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 nastavnika koji vodi taj razred (FK)
- idSkola – šifra škole u kojoj se razred nalazi (FK)

UCITELJ

- **idUcitelj** – šifra nastavnika (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 – poveznica na zvučni zapis diktata koja se koristi za reprodukciju

NADOPUNA

- **idZadatak** – šifra zadatka (FK)
- unos – generirani točan odgovor kojeg je potrebno nadopunit u zadatku

ODABIR

- **idZadatak** – šifra zadatka (FK)
- odgovor1 – ponuđeni točan odgovor učeniku pri rješavanju zadatka
- odgovor2 – ponuđeni netočan 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

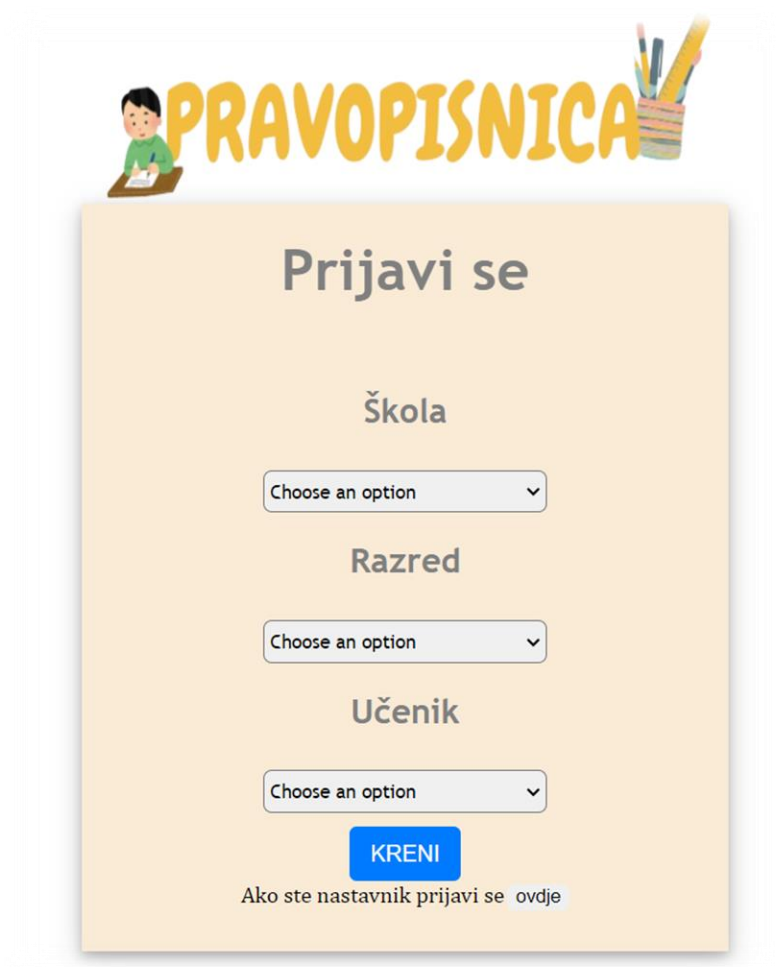
- ucenikovOdgovor – tekstualni odgovor učenika koji je unio pri rješavanju zadatka
- tocanOdgovor – ponuđeni točan odgovor za određeni zadatak
- tocan – zastavica koja određuje je li učenik određeni zadatak riješio točno ili netočno
- idUcenik – šifra učenika koji je riješio određeni zadatak na kojeg 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 četiri vrste zadataka, učenici mogu poboljšati svoje znanje pravopisa, uz pomoć sučelja koje potiče aktivno učenje. Nastavnikovo sučelje nudi niz funkcionalnosti. Nastavnici mogu kreirati nove vježbe i zadatke, prilagođavajući ih svojim specifičnim potrebama. Također mogu uređivati postojeće vježbe i zadatke, te imati pregled nad svim vježbama, zadacima i učenicima iz svojih razreda. Ova funkcionalnost omogućuje praćenje i prilagodbu sadržaja kojeg je stvorio. Posebna funkcionalnost Pravopisnice, koja je čini jedinstvenom među sličnim aplikacijama, je digitalni diktat. Ova funkcionalnost omogućuje nastavnicima da stvore zadatke sa zvučnim zapisom diktata čiji tekst je generirala umjetna inteligencija, a u zvučni zapis je pretvorio alat za pretvaranje teksta u govor. Učenici mogu slušati ove zapise i potom zapisivati tekst, čime se vježbaju ne samo pravopisu, nego i 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, što čini 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).



PRAVOPISNICA

Prijava se

Škola

Choose an option

Razred

Choose an option

Učenik

Choose an option

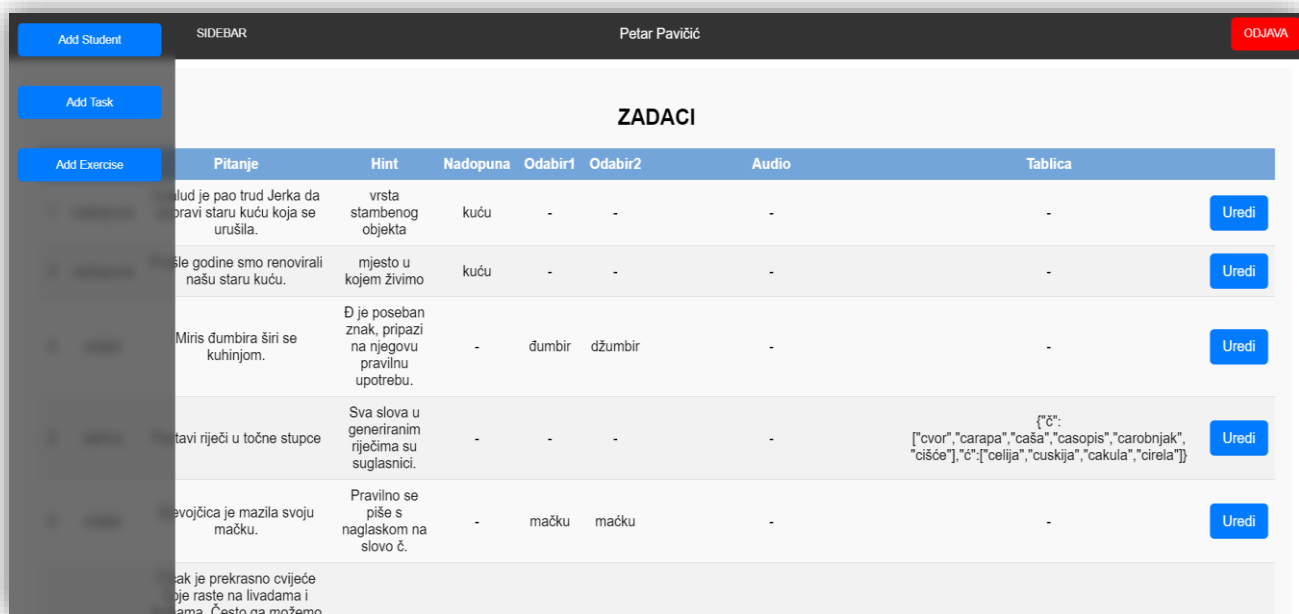
KRENI

Ako ste nastavnik prijavite se [ovdje](#)

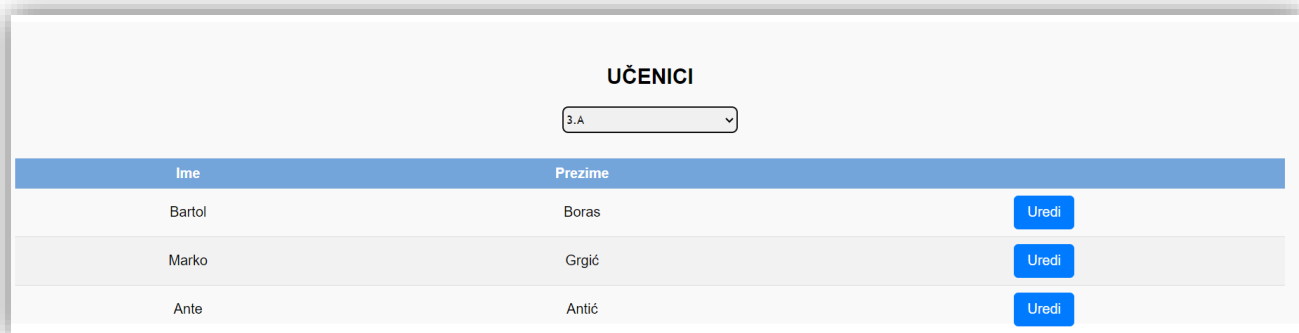
Slika 4.1 Početna stranica

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.



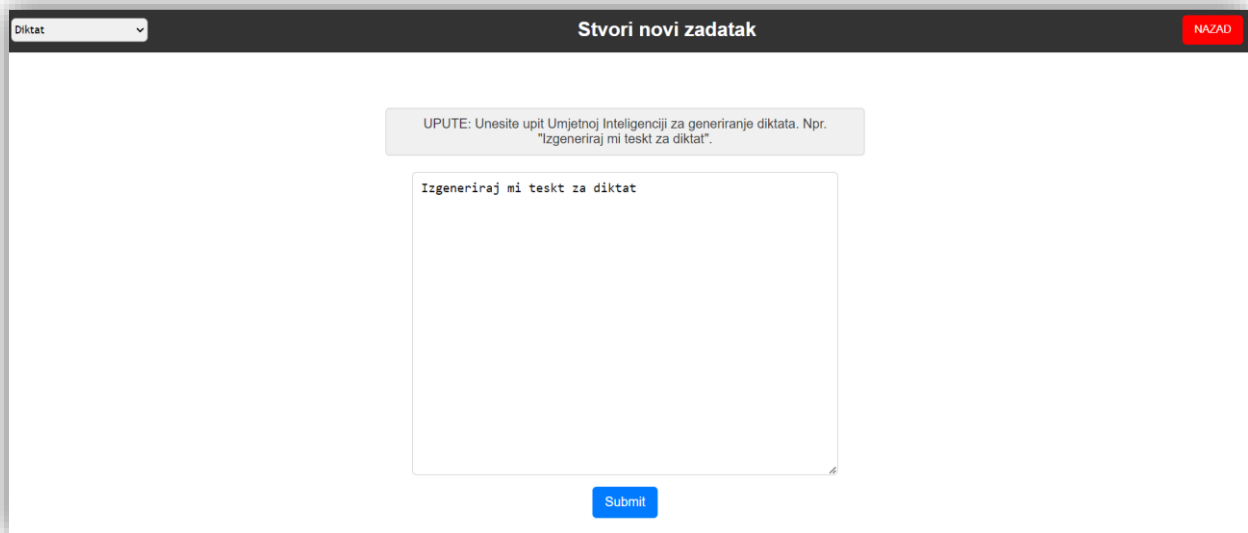
Slika 4.2 Nastavnikova profilna stranica s otvorenim sidebar-om



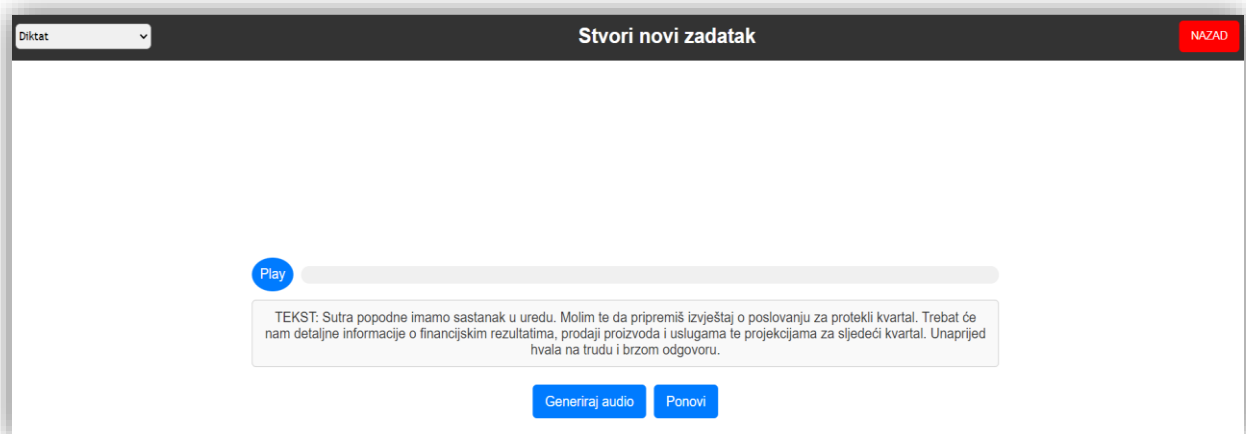
Slika 4.3 Prikaz učenika na profilnoj stranici s odabranim razredom

Pri stvaranju novih zadataka, nastavnik mora odabrati vrstu zadatka iz padajućeg izbornika (diktat, nadopuna, odabir, tablica). 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 4.4). Klikom na gumb, šalje se upit 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 4.5).

Nakon toga, nastavnik može reproducirati zvučni zapis pritiskom na gumb "Play" kako bi vidio je li sve u redu, uz opcije za premotavanje i pauziranje 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 biti preusmjeren na svoju profilnu stranicu.



Slika 4.4 Stvaranje digitalnog diktata - primjer slanja upita AI



Slika 4.5 Stvaranje digitalnog diktata - primjer generiranja zvučnog zapisa

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 je svaki stupac jedno od tih slova (Slika 4.6). Umjetna inteligencija generira riječi koje započinju s tim slovom, 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.

Tablica

Stvori novi zadatak

NAZAD

UPUTE: Unesite dva slova za koja želite da se generiraju riječi. Za ta 2 slova će se stvoriti 2 stupca u koje će učenici postavljati ponuđene riječi.

Tablica

Stupac1 (č): čokolada, čevapi, čirlica, čuturica, čubasta
Stupac2 (ć): čarapa, čizma, čvor, čičak, čaplja
HINT: Prvo slovo u riječima prvog stupca je 'č'. Prvo slovo u riječima drugog stupca je 'ć'.

Spremi Ponovi

Slika 4.6 Odgovor umjetne inteligencije na upit za tablični zadatak

Pri stvaranju zadatka vrste nadopuna, nastavniku se nudi forma za slanje upita umjetnoj inteligenciji, zajedno s uputama kako pravilno formulirati upit. Nakon slanja zahtjeva na API, umjetna inteligencija generira rečenicu u kojoj će se na prazno mjesto morati nadopuniti točna riječ. Uz rečenicu, nastavniku će biti prikazana točna riječ koja se treba nadopuniti te hint koji će pomoći učenicima pri rješavanju zadatka (Slika 4.7).

Nadopuna ▼ Stvori novi zadatak NAZAD

UPUTE: Unesite upit Umjetnoj Inteligenciji za generiranje rečenice koju će trebat nadopunit s ispravnom riječi. Npr. "Izgeneriraj mi smislenu rečenicu koja sadrži riječ kuća". Točna riječ, koju će trebat napisati, mora se nalaziti u rečenici u istom obliku.

Rečenica: Moj brat je nedavno kupio prekrasnu kuću u mirnom predgrađu.
Riječ za nadopunit: "kuću"
HINT: Objekt u kojem ljudi žive, obično građen od cigle ili drva.

Spremi Ponovi

Slika 4.7 Generirani zadatak s nadopunom od strane umjetne inteligencije

Kod zadatka vrste odabir, nastavniku se nudi forma u koju treba unijeti riječ čija se pravopisna ispravnost želi provjeriti, uz priložene upute kakav se odgovor očekuje od umjetne inteligencije. Ta riječ se šalje umjetnoj inteligenciji koja generira rečenicu s ispravnom riječju i netočnom verzijom te riječi, postavljajući ih na odgovarajuće mjesto u rečenici. Učenik će pri rješavanju ove vrste zadatka trebati odabrati jednu od te dvije riječi. Za obje vrste zadataka, nadopuna i odabir, nastavnik može ponovno generirati zadatke pomoću umjetne inteligencije ako nije zadovoljan rezultatom, ili ih može spremiti u bazu podataka ako je zadovoljan.

U slučaju kreiranja novih vježbi, nastavnik treba kroz svoj profil doći do stranice za kreiranje novih vježbi (Slika 4.8). Na toj stranici prvo će morati unijeti ime vježbe, a zatim će mu biti prikazana tablica s trenutno stvorenim zadacima. Nastavnik mora odabrati barem jedan zadatak iz te tablice, inače mu neće biti dopušteno stvoriti vježbu. Također, ako se predomisli, zadatke koje je dodao u vježbu može maknuti klikom na gumb „Makni“. Na stranici se također nalazi potvrdni okvir (eng. checkbox) koji označava želi li nastavnik tu vježbu postaviti za rješavanje, a po zadanoj nije označen. Ako ga označi, ta će vježba biti postavljena učenicima za rješavanje. Međutim, ako je već neka druga prijašnja vježba postavljena za rješavanje, sustav neće dopustiti da i trenutna vježba bude postavljena, jer u jednom trenutku samo jedna vježba može biti postavljena za rješavanje. U tom slučaju,

nastavnik će morati urediti prethodno postavljenu vježbu, odznačiti je i potom označiti željenu vježbu za rješavanje. Nakon što završi sa svim izmjenama, nastavnik može spremiti novu vježbu i ona će biti dostupna učenicima sukladno njegovim odabirima.

NAZAD Stvori novu vježbu SPREMI

Petrova Vježba

Postavi vježbu učenicima za vježbanje: ☐

Odabrani zadaci

ID	Vrsta	Pitanje	Hint	Odabir 1	Odabir 2	Nadopuna	Audio	Tablica
6	diktat	Čičak je prekrasno cvijeće koje raste na livadama i šumama. Često ga možemo naći duž staza i rubova šuma. Čuvajmo prirodu i njen raznoliki biljni svijet. Čičarija je prekrasna regija u Istri poznata po svojoj netaknutoj prirodi i bogatom biljnom i životinjskom svijetu.	none	-	-	-	blob:http://localhost:3000/906293af-01a3-4d30-a4d3-c964fb9df598	-

Makni

Svi zadaci

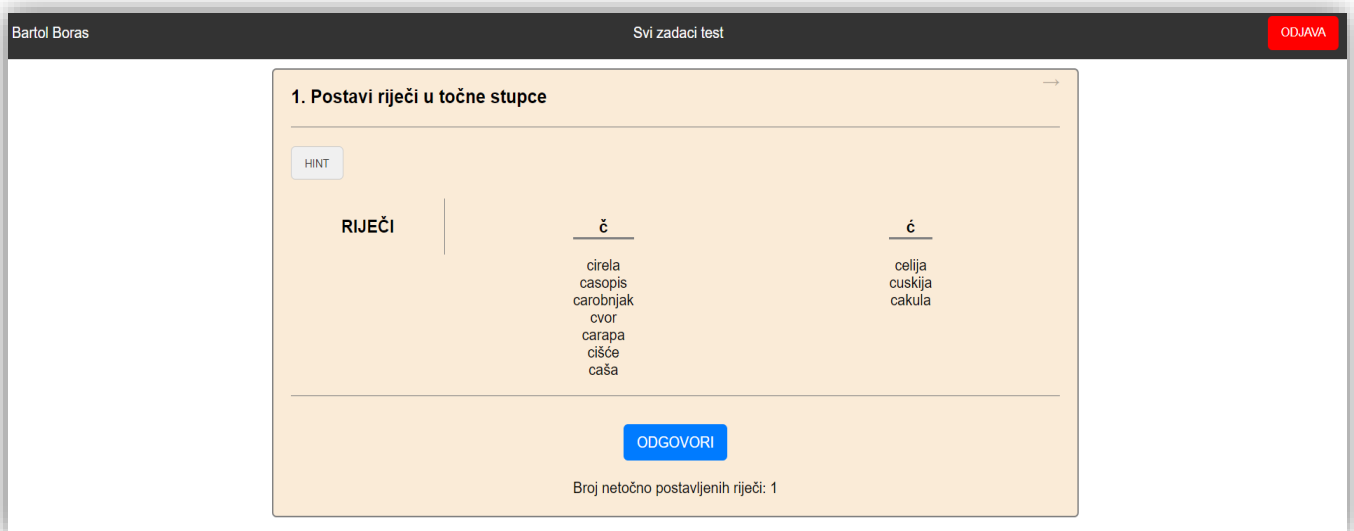
ID	Vrsta	Pitanje	Hint	Odabir 1	Odabir 2	Nadopuna	Audio	Tablica
3	nadopuna	Prošle godine smo renovirali našu staru kuću.	mjesto u kojem živimo	-	-	kuću	-	-

Dodaj

Slika 4.8 Stranica za stvaranje nove vježbe

Na početnoj stranici (Slika 4.1), učenik odabire svoje podatke, uključujući školu, razred i ime, te klikom na "KRENI" otvara novu stranicu za rješavanje vježbe koju je postavio nastavnik. Na toj stranici, unutar roditeljske komponente, ovisno o vrsti zadatka, izmjenjuju se različite dječje komponente zadataka, svaka sa svojom formom. Na vrhu na navigacijskoj traci nalaze se redom slijeva: ime učenika koji rješava vježbu, naziv vježbe, te gumb za odjavu. Svaki zadatak ima opciju preskakanja, pri čemu se zadatak automatski računa kao pogrešan. Uz to, postoji gumb "Hint" koji učeniku pruža pomoć pri rješavanju zadatka i služi kao toggle gumb za prikazivanje pomoći. Vrijeme rješavanja svakog zadatka mjeri se u pozadini, a klikom na "Odgovori" bilježe se podaci u bazu. Ovi podaci uključuju id zadatka, id vježbe, id učenika, učenikov odgovor, točan odgovor, zastavicu koja označava je li odgovor točan, te vrijeme potrebno za rješavanje zadatka. U slučaju da se učenik odjavi u bilo kojem trenutku rješavanja, vježba se prekida, što se označava preusmjeravanjem korisnika na početnu stranicu i slanjem zapisa prema bazi podataka. Sustav na taj način osigurava bilježenje svih relevantnih informacija kako bi se omogućilo praćenje napretka učenika i točnost u rješavanju zadataka.

Kod rješavanja tabličnog zadatka, odnosno zadatka vrste drag and drop, učeniku je prikazana komponenta (Slika 4.9) koja sadrži riječi koje učenik mora rasporediti, tj. povući i ispustiti u odgovarajuće stupce. Ponuđene riječi se nalaze na lijevoj strani, dok se stupci nalaze na desnoj strani. U gornjem desnom kutu nalazi se strelica koja služi za preskakanje zadatka, dok se u gornjem lijevom kutu nalazi gumb za pomoć pri rješavanju. Ako učenik ne rasporedi sve riječi točno u odgovarajuće stupce, na dnu će mu biti prikazana poruka koja ga obavještava koliko je riječi netočno rasporedio. U slučaju da su sve riječi točno raspoređene, prelazi se na sljedeći zadatak, pri čemu se učitava komponenta sljedećeg zadatka.



Slika 4.9 Drag and drop zadatak s tablicom

Kod rješavanja digitalnog diktata, komponenta prikazana učeniku se sastoji od nekoliko ključnih elemenata. Na vrhu se nalazi gumb za reproduciranje zvučnog zapisa generiranog pomoću ElevenLabs TTS API-ja na hrvatskom jeziku, uz traku koja omogućuje kontrolu reprodukcije. Traka za reprodukciju prikazuje trenutnu poziciju u zvučnom zapisu i omogućuje pauziranje te pomjeranje na željeno mjesto za nastavak reprodukcije. U prostoru ispod reprodukcije zvučnog zapisa, učenik piše diktat. Nakon što završi s pisanjem, klikom na gumb „Odgovori“ šalje se zahtjev prema Ispravi.me API-ju za pravopisnu i kontekstualnu provjeru. API vraća odgovor i aplikacija učeniku prikazuje netočno napisane riječi i njihove ispravne verzije (Slika 4.10). Ispravi.me API predlaže

točne riječi za zamjenu netočnih. Ako je diktat napisan bez grešaka, prelazi se na sljedeći zadatak uz slanje zapisa prema bazi podataka. Ova funkcionalnost ne samo da pomaže učenicima u savladavanju pravopisnih vještina, već im pruža i trenutni feedback za brže učenje i ispravljanje grešaka, te potiče samostalno učenje.



Slika 4.10 Primjer komponente za rješavanja digitalnog diktata

U zadacima vrste nadopuna, učeniku se prikazuje komponenta koja, osim standardnih gumba za preskakanje i prikazivanje hinta, sadrži rečenicu s praznom linijom na mjestu gdje se traži riječ koju je potrebno nadopisat. Učenik upisuje traženu riječ u prazno polje ispod rečenice. Ako je riječ krivo napisana, učeniku će se prikazati kratka poruka o pogrešci (Slika 4.11). Tek kada učenik ispravno unese riječ, prelazi na sljedeći zadatak. Prije prelaska na novi zadatak, učeniku se prikazuje kratka poruka koja potvrđuje da je odgovor točan.

Slika 4.11 Prikaz netočno riješenog zadatka s nadopunom

Kod zadnje vrste zadataka, s odabirom, učeniku se prikazuje rečenica s dvije ponuđene riječi koje se nalaze na pravom i smislenom dijelu rečenice, jednom ispravnom i jednom neispravnom. Kad učenik odabere riječ mora kliknuti na gumb „Odgovori“ pri čemu se šalje zapis o odgovoru na server i ako je točno odgovorio prikazat će mu se kratka poruka te automatski prijeći na sljedeći zadatak (Slika 4.12). U slučaju da je krivo odgovorio učeniku će se prikazat kratka poruka da je netočno odgovorio, a morat će ponovo odgovoriti na to pitanje. Naposljetku, nakon što je odgovorio na sva pitanja korisnik je preusmjeren na početnu stranicu.

Slika 4.12 Prikaz točno riješenog zadatka s ponuđenim odgovorima

Zaključak

U ovom radu istražena je integracija tehnologije u obrazovanje s naglaskom na podučavanje pravopisa kroz primjenu generativne umjetne inteligencije i automatiziranih alata za ispravljanje pravopisa, kao što je Ispravi.me. Tehnologija pruža brojne prednosti u obrazovnom procesu, uključujući personalizaciju učenja, povećanu interaktivnost, fleksibilnost i bolju suradnju među učenicima i nastavnicima. Analizom dostupnih aplikacija za pravopis i gramatiku utvrđeno je da većini nedostaje inovativnosti i interaktivnosti koja bi dodatno potaknula angažman učenika, kao i integracija s vanjskim alatima za olakšavanje podučavanja i ispravljanja pravopisa. Ispravi.me se ističe kao napredan alat za pravopisno i kontekstualno ispravljanje koji koristi napredne modele za provjeru pravopisa, čime značajno smanjuje administrativno opterećenje nastavnicima.

Prve postavljene ideje rada su implementirane kroz osnovne funkcionalnosti aplikacije koje integriraju nekoliko vanjskih tehnologija putem API-ja kako bi olakšale učenje i podučavanje pravopisa. Tako su korištene prednosti ChatGPT i ElevenLabs TTS API-ja za sastavljanje zadataka i Ispravi.me API-ja za ispravljanje zadataka što značajno smanjuje opterećenje nastavnicima. Aplikacija se ističe po tome što integrira funkcionalnost digitalnog diktata te omogućava prilagodbu zadataka prema potrebama učenika, što doprinosi efikasnijem učenju i razumijevanju pravopisa.

Moguća proširenja funkcionalnosti aplikacije uključuju stvaranje statistika za učenike i razrede, bilo za cjelokupno vrijeme ili po pojedinim vježbama. Također je moguće dodavanje novih vrsta zadataka, stvaranje zasebnih lekcija i grupiranje po lekcijama. Nadalje, adaptivno i individualno predlaganje zadataka korisnicima na temelju njihovih prethodnih rezultata može značajno unaprijediti korisničko iskustvo. Uvođenje različitih svakodnevnih pravopisnih savjeta prilikom posjeta početnoj stranici također može unaprijediti iskustvo korisnika kao i uvođenje elementa natjecanja odnosno skupljanja bodova ili sustava nagrada. Još jedno potencijalno proširenje koje bi značajno unaprijedilo funkcionalnost aplikacije je omogućavanje komunikacije uživo između nastavnika i učenika tijekom rješavanja vježbi. Na taj način, ukoliko učenici naiđu na poteškoće ili imaju pitanja, mogu odmah dobiti potrebnu pomoć, čime bi se pružila

podrška u stvarnom vremenu. Što se tiče proširenja na razvojnoj strani, integracija s dodatnim vanjskim tehnologijama, poput podrške za više jezika ili mogućnosti prijevoda teksta, mogla bi biti korisna. Primjerice, integracija s Google Translate API-jem za podršku višejezičnosti ili API-jima za autentifikaciju poput Google OAuth ili Facebook Login. Kontinuirano poboljšavanje korisničkog sučelja i iskustva kako bi aplikacija postala intuitivnija i privlačnija za upotrebu također je ključan aspekt daljnjeg razvoja.

Literatura

- [1] *What is Educational Technology and Why is it Important?*, Loyola University Maryland, <https://www.loyola.edu/school-education/blog/2021/what-is-educational-technology.html>, pristupljeno 15. lipnja 2024.
- [2] *7 benefits of technology in the classroom*, Adobe Blog, <https://blog.adobe.com/en/publish/2021/08/23/7-benefits-of-technology-in-the-classroom>, pristupljeno 15. lipnja 2024.
- [3] *Advantages and disadvantages of technology in education*, Allison Academy, <https://www.allisonacademy.com/students/education/technology-in-education/advantages-and-disadvantages-of-technology-in-education/>, pristupljeno 15. lipnja 2024.
- [4] *22 Advantages and Disadvantages of Technology in Education*, Future of Working, <https://futureofworking.com/10-advantages-and-disadvantages-of-technology-in-education/>, pristupljeno 15. lipnja 2024.
- [5] *What are disadvantages to technology in the classroom?*, Social Sci LibreTexts, [https://socialsci.libretexts.org/Courses/Fresno_City_College/Education_30%3A_Introduction_to_Education_\(Perez\)/09%3A_Technology/9.03%3A_What_are_disadvantages_to_technology_in_the_classroom](https://socialsci.libretexts.org/Courses/Fresno_City_College/Education_30%3A_Introduction_to_Education_(Perez)/09%3A_Technology/9.03%3A_What_are_disadvantages_to_technology_in_the_classroom), pristupljeno 15. lipnja 2024.
- [6] Matasić I., Dumić S., *Multimedijske tehnologije u obrazovanju*, Hrčak - Portal of scientific journals of Croatia, god. 18, br. 1, 2012, str. 143 - 151.
- [7] Pejić Papak P., Grubišić Krmpotić H., *Poučavanje primjenom suvremene tehnologije u obrazovanju*, Život i škola: časopis za teoriju i praksu odgoja i obrazovanja, god. 62 br. 3, 2016, str. 153 - 162.
- [8] Nguyen A., Ngo H.N., Hong Y., *Ethical principles for artificial intelligence in education*, *Educ Inf Technol*, br. 28, 2023, str. 4221–4241.
- [9] Spelling City, <https://www.spellingcity.com/>, pristupljeno 15. lipnja 2024.

- [10] Simmons K. D., Carpenter L. B., *Spelling and Assistive Technology: Helping Students with Disabilities Be Successful Writers, Physical Disabilities: Education and Related Service*, god. 29, br. 2, 2010, str. 5 - 19.
- [11] Ryder C., Giannoni D.S., *Apps-olutely Brilliant: Can Technology Improve the Spelling Ability of Primary School Children?*, *Language Studies Working Papers*, br. 7, 2016, str. 36 – 49.
- [12] *Product Features*, Grammarly, <https://www.grammarly.com/features>, pristupljeno 8. lipnja 2024.
- [13] *Spellzone – about English spelling*, Spellzone, https://www.spellzone.com/pages/about_spellzone.cfm, pristupljeno 8. lipnja 2024.
- [14] *Client Server Architecture*, EnjoyAlgorithms, <https://www.enjoyalgorithms.com/blog/client-server-architecture>, pristupljeno 12. lipnja 2024.
- [15] *Making HTTP requests with Axios*, CircleCI, <https://circleci.com/blog/making-http-requests-with-axios>, pristupljeno 15. lipnja 2024.
- [16] *React State*, W3Schools, https://www.w3schools.com/react/react_state.asp, pristupljeno 13. lipnja 2024.
- [17] *React Props*, W3Schools, https://www.w3schools.com/react/react_props.asp, pristupljeno, 13. lipnja 2024.
- [18] *Introduction to Node.js*, Node.js, <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>, pristupljeno 10. lipnja 2024.
- [19] *What is Express.js?*, Codecademy, <https://www.codecademy.com/article/what-is-express-js>, 10. lipnja 2024.
- [20] *Express/Node introduction*, MDN Web Docs, https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction, pristupljeno 10. lipnja 2024.

- [21] *What is PostgreSQL?*, PostgreSQL, <https://www.postgresql.org/docs/current/intro-what-is.html>, pristupljeno 10. lipnja 2024.
- [22] *How does ChatGPT work?*, Zapier, <https://zapier.com/blog/how-does-chatgpt-work/#transformer-architecture>, pristupljeno 12. lipnja 2024.
- [23] *API Reference Text To Speech*, ElevenLabs, <https://elevenlabs.io/docs/api-reference/text-to-speech>, pristupljeno 12. lipnja 2024.
- [24] *O usluzi*, Ispravi.me, <https://ispravi.me/info/>, pristupljeno 12. lipnja 2024.

Sažetak

Aplikacija za vježbu pravopisa uz pomoć umjetne inteligencije i usluge strojne provjere pravopisa

U današnjem digitalnom dobu, tehnologija ima ključnu ulogu u obrazovanju, posebno u unapređenju jezičnih vještina poput pravopisa. Tradicionalni pristupi u učenju pravopisa često su se oslanjali na memorizaciju pravila i praktičnu primjenu kroz vježbe pisanja. Integracija tehnologije omogućava prilagodljiv pristup učenju i podučavanju, koristeći interaktivne i vanjske alate koji se prilagođavaju potrebama učenika i olakšavaju posao nastavnicima. Ispravi.me se ističe kao napredan alat za ispravljanje pravopisnih grešaka, nudeći automatsko i kontekstualno ispravljanje riječi. Ovaj rad istražuje primjenu generativne umjetne inteligencije u obrazovanju, posebno kroz razvoj aplikacije Pravopisnica. Aplikacija je izrađena korištenjem Node.js na backendu i Reacta na frontendu, a za bazu podataka je korišten PostgreSQL, čije prednosti uključuju visoku pouzdanost, skalabilnost i podršku za složene upite, što je poželjno za obradu i pohranu podataka korisnika pri rješavanju vježbi. Aplikacija također koristi API-je poput ChatGPT-a za izradu zadataka, ElevenLabs TTS za visokokvalitetnu sintezu teksta u govor, te integrira mogućnosti ispravljanja zadataka putem Ispravi.me API-ja kako bi unaprijedila proces učenja pravopisa. Integracija tehnologije u obrazovni proces omogućava nastavnicima fokus na interakciju s učenicima i pružanje personalizirane podrške, čineći obrazovanje učinkovitijim i pristupačnijim. Kroz ovakve inovativne pristupe, nastavnici se mogu osloboditi rutinskih zadataka ispravljanja i fokusirati se na pružanje individualne podrške i feedbacka svojim učenicima.

Ključne riječi: pravopis, React, Express, Node.js, PostgreSQL, API, Ispravi.me, ChatGPT, ElevenLabs TTS

Summary

Spelling Practice Application Utilizing Artificial Intelligence and machine spell-checking services

In today's digital age, technology plays a key role in education, especially in improving language skills such as spelling. Traditional approaches to learn spelling have often relied on memorization of rules and practical application of writing exercises. The integration of technology enables an adaptive approach to learning and teaching, using interactive and external tools that adapt to the needs of students and facilitate the work of teachers. Ispravi.me stands out as an advanced tool for correcting spelling errors, offering automatic and contextual word correction. This paper explores the usage of generative artificial intelligence in education, especially through the development of the Pravopisnica application. The application was created using Node.js on the backend and React on the frontend, and PostgreSQL was used for the database, the advantages of which include high reliability, scalability and support for complex queries, which is essential for processing and storing user data when solving exercises. The application also uses APIs such as ChatGPT for task creation, ElevenLabs TTS for high-quality text-to-speech synthesis, and it integrates task correction capabilities via the Ispravi.me API to enhance the spelling learning process. Integrating technology into the educational process allows teachers to focus on interacting with students and providing personalized support, making education more efficient and accessible. Through such innovative approaches, teachers can free themselves from routine correction tasks and focus on providing individual support and feedback to their students.

Keywords: spelling, React, Express, Node.js, PostgreSQL, API, Ispravi.me, ChatGPT, ElevenLabs TTS