

Aplikacija za vježbu vokabulara pomoću osmosmjerke

Hrnčić, Mirta

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:168:342037>

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

Download date / Datum preuzimanja: **2025-03-21**



Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1616

**APLIKACIJA ZA VJEŽBU VOKABULARA POMOĆU
OSMOSMJERKE**

Mirta Hrnčić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1616

**APLIKACIJA ZA VJEŽBU VOKABULARA POMOĆU
OSMOSMJERKE**

Mirta Hrnčić

Zagreb, lipanj 2024.

ZAVRŠNI ZADATAK br. 1616

Pristupnica: **Mirta Hrnčić (0036538406)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: doc. dr. sc. Tomislav Jaguš

Zadatak: **Aplikacija za vježbu vokabulara pomoću osmosmjerke**

Opis zadatka:

Osmosmjerka je igra u kojoj se riječi nalaze u mreži slova, i može ih se čitati vodoravno, uspravno ili dijagonalno. Cilj igre je pronaći sve riječi koje su skrivene u mreži. U nastavi stranih jezika često se koriste za vježbanje vokabulara i pravopisa kroz igru ili timski rad. U sklopu ovog završnog rada potrebno je osmisliti i kreirati aplikaciju koja će učenicima nižih razreda osnovne škole omogućiti vježbanje vokabulara odabranog stranog jezika traženjem riječi u osmosmjerci. Potrebno je kreirati sustav koji će iz zadanog skupa riječi generirati osmosmjerke te omogućiti njihovo rješavanje. Sustav treba omogućiti održavanje natjecanja u rješavanju osmosmjerki te bilježiti povijesti aktivnosti svih učenika. Administrator ili nastavnik zadaje zadatke te kroz prikladno sučelje može vidjeti napredak svakog učenika i druge prikladne informacije. Učenici mogu tražiti pomoć (eng. hint) pri igranju, u obliku broja slova koje tražena riječ sadržava ili označavanja nasumičnog slova u rješenju.

Rok za predaju rada: 14. lipnja 2024.

Sadržaj

Uvod	1
1. Postojeće aplikacije za vježbu vokabulara	3
1.1. NY Times Games	3
1.2. StudyCat	4
2. Ideja rada	6
3. Korišteni alati i tehnologije	8
3.1. React.js	10
3.2. Node.js	11
3.3. Express	11
3.4. OpenAI API	11
3.5. PostgreSQL	13
4. Model podataka	14
4.1. Opisi relacija	16
5. Implementacijski detalji	23
5.1. Implementacija algoritma za raspored riječi	23
6. Korištenje aplikacije	25
6.1. Sučelje za nastavnike	25
6.2. Aplikacija za učenike	28
6.3. Pregled rezultata	32
7. Daljnji razvoj	34

8. Zaključak	35
Literatura	36
Sažetak	38
Summary	39

Uvod

Učenje stranih jezika danas je neizostavna vještina koja otvara vrata brojnim prilikama, bilo da je riječ o obrazovanju, karijeri ili osobnom razvoju. Vokabular pritom zauzima središnje mjesto, jer upravo kroz poznavanje riječi razvijamo sposobnost razumijevanja i izražavanja. Bez adekvatnog vokabulara, učenici teško mogu razumjeti ili učinkovito koristiti jezik, što naglašava potrebu za učinkovitim metodama učenja vokabulara. No, tradicionalni načini učenja vokabulara kroz čitanje tekstova i učenje riječi na pamet često su nedovoljno motivirajući, posebno mlađima, što smanjuje njihov interes za učenje i ponavljanje, a time i njihov uspjeh.

Istovremeno, informatička pismenost djece je u stalnom porastu te smo u vremenu uvođenja sve više tehnologije u škole. Razvojem korisnih aplikacija, osmišljenih i dizajniranih u suradnji s nastavnicima i učenicima, tehnologija može izvršno upotpuniti nastavne metode ondje gdje tradicionalni načini učenja manjkaju svojom dinamičnošću i interaktivnošću.

Igre i digitalne aplikacije pružaju inovativne načine za obogaćivanje procesa učenja. Osmosmjerke, popularna jezična igra, učinkovit su alat za poboljšanje prepoznavanja riječi i zadržavanje vokabulara. Kombinacijom zabave i edukacije, osmosmjerke mogu značajno povećati motivaciju učenika i učiniti proces učenja zanimljivijim [1]. Istraživanja pokazuju da igre temeljene na učenju ne samo da povećavaju angažman učenika, već i poboljšavaju akademske rezultate, što dodatno podupire korištenje takvih metoda u obrazovanju [2]. Igra osmosmjerke aktivno uključuje učenike u proces pretraživanja i prepoznavanja riječi, čime se pospješuje njihovo pamćenje i razumijevanje novih pojmova. Osim toga, osmosmjerke omogućuju učenicima da uče kroz igru, što doprinosi njihovom zadovoljstvu i motivaciji za daljnje usvajanje jezika.

Ovaj završni rad istražuje razvoj aplikacije za vježbu vokabulara stranih jezika pomoću osmosmjerke. Aplikacija je dizajnirana s ciljem pružanja interaktivnog i ugodnog okruženja za učenje, koje ne samo da pomaže učenicima u proširivanju njihovog vokabulara, već ih također potiče na redovitu praksu kroz igru. Uz samu igru za učenike, razvijeno je i sučelje za nastavnike koje omogućava stvaranje igara za pojedine razrede te administraciju.

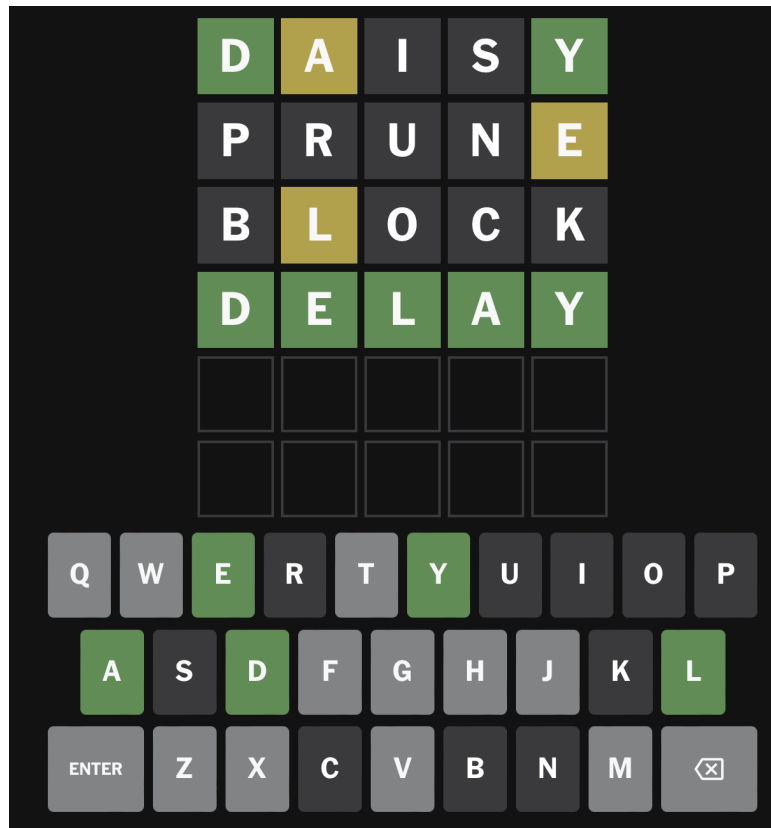
Na početku rada biti će opisane neke od postojećih aplikacija za vježbu vokabulara te ideja iza osmišljene aplikacije. Zatim će biti napravljen detaljan pregled tehnologija i alata korištenih za ostvarenje aplikacije osmosmjerke i nastavničkog sučelja te pripadni model baze podataka. Nakon toga je opisan implementirani algoritam za raspored riječi unutar osmosmjerke te su dane upute za korištenje i mogućnosti daljnjeg razvoja. Na kraju rada, izneseni su zaključak i sažetak.

1. Postojeće aplikacije za vježbu vokabulara

Korištenje igara u svrhu učenja pruža zabavan i učinkovit način stjecanja znanja, bez osjećaja da se zapravo uči. Mnoge postojeće igre nisu obrazovnog karaktera, odnosno osmišljene za korištenje u nastavi, no vrlo su djelotvorne i u tom kontekstu. Primjer takve igre je *Scrabble*, koji se već godinama koristi za zabavu i pomaže u širenju rječnika. Međutim, u današnje vrijeme sve se više koriste digitalne igre koje svojim interaktivnim sučeljima i praktičnosti motiviraju korisnike na što češću upotrebu. U nastavku je proučeno nekoliko odabranih aplikacija za vježbu vokabulara.

1.1. NY Times Games

Poznati časopis *New York Times* još od tiskanog doba sadrži igre, među najstarijima *The Crossword* i *Sudoku* te 2014. godine prelaskom na digitalni format igara šire svoju kolekciju. Najpoznatiji je dodatak napravljen 2022. godine igrom *Wordle* [3]. *Wordle* je postao globalni hit, privlačeći svakodnevno brojne igrače koji u šest pokušaja nastoje pogoditi odabranu petoslovnu englesku riječ. U svakom pokušaju ispravno smještene slova su označena zelenom bojom, a ispravna slova na pogrešnom mjestu žutom. Pokušaj rješavanja za jedan dan je prikazan slikom 1.1. Ova jednostavna, ali zarazna igra pokazala je kako kratke, svakodnevne aktivnosti mogu značajno povećati motivaciju za učenje. Također, motiviranosti dijela korisnika doprinosi kompetitivan element mogućnosti dijeljenja rezultata putem društvenih mreža. Danas, kroz svoju web platformu i mobilnu aplikaciju, *New York Times* nudi desetak igara, a njihov broj i dalje raste, prilagođavajući se različitim interesima korisnika.



Slika 1.1. najpoznatija igra *NY Times*-a: *Wordle* za datum 20.8.2024

1.2. StudyCat

StudyCat je edukativna aplikacija koja se specijalizira u učenju stranih jezika kroz interaktivne igre i aktivnosti, posebno namijenjene djeci. Moguće je odabrati između pet jezika: engleski, španjolski, njemački, francuski i kineski [4]. Aplikacija kombinira zabavne igre s obrazovnim sadržajem, pružajući djeci priliku da kroz igru razvijaju svoje jezične vještine. Na slici 1.2 prikazane su jezične vještine čiju vježbu aplikacija podržava. *StudyCat* nudi razne igre koje potiču učenje kroz zabavu. Primjeri igara uključuju povezivanje riječi s odgovarajućim slikama, slaganje rečenica u ispravan redoslijed, te prepoznavanje i ponavljanje zvukova i riječi. Djeca mogu sudjelovati u igrama poput "Memory" za učenje novih riječi ili "Puzzle" igara koje ih potiču na rješavanje zagonetki koristeći riječi i fraze iz lekcija. Osim zadataka namijenjenih vježbi vokabulara i izgovora, pojavljuju se i zadatci čitanja odnosno slušanja s razumijevanjem: interaktivne priče i dijalozi prilagođeni razini znanja djeteta. Djeca čitaju ili slušaju tekstove, a zatim odgovaraju na pitanja ili rješavaju zadatke koji zahtijevaju razumijevanje pročitanoog ili odslušanog sadržaja. Kroz ove igre, djeca ne samo da proširuju svoj vokabular, već i jačaju sposobnost

slušanja, izgovora i razumijevanja stranog jezika, sve u okruženju koje je zabavno i prilagođeno njihovom uzrastu. Aplikacija također koristi sustav nagrađivanja, gdje djeca osvajaju bodove i nagrade za uspješno završene zadatke, čime se dodatno potiče njihova motivacija za učenje.



Slika 1.2. Mogućnosti učenja unutar aplikacije *StudyCat*

2. Ideja rada

Ideja ovog rada je napraviti aplikaciju sličnu ranije navedenima s ciljem motivacije učenika na vježbu vokabulara stranih jezika kroz igru. Korištenje tehnologije u školama je sve češće, no pravilno razvijenih aplikacija napravljenih za kvalitetnu upotrebu iste tehnologije nema dovoljno. Stoga, cilj ove aplikacije nije zamjeniti uporabu klasičnih principa učenja i vježbe vokabulara poput čitanja tekstova, traženja prijevoda u riječnicima i ispisivanja riječi na papir, već nadopuniti te metode. Učenici bi igru mogli igrati na školskim tabletima te kod kuće kao vježbu ili domaću zadaću na laptopima.

Prvi je korak registracija nastavnika te dodavanje učenika u razred. Za svakog će se učenika pritom kreirati lozinka te će se ista prikazati nastavniku. Ukoliko je potrebno, nastavnik za učenika može kreirati novu lozinku. Nastavniku se nudi i mogućnost stvaranja igre. Potrebno je odabrati školu i razred, a zatim dati ime igri. Nakon toga, nastavnik unosi parove riječi: hrvatsku i stranu riječ te je time stvaranje igre završeno. Također, moguće je stvoriti igru čitanja s razumijevanjem gdje nastavnik unosi željeni tekst i odabire broj pitanja koja o njemu želi postaviti. Pitanja će za njega kreirati generativni alat umjetne inteligencije te ukoliko je njima zadovoljan, igra će se kreirati.

Ideja iza prijave učenika u sustav je maksimalno ju pojednostaviti i smanjiti mogućnost pogreške. Stoga učenici iz padajućeg izbornika biraju svoju školu, razred te ime i prezime. Nakon toga odabiru tri životinje koje su im dodijeljene kao lozinka. Potrebno je izabrati ispravne životinje ispravnim redoslijedom. Nakon prijave, učenik se nalazi na svojoj stranici te može izabrati jednu od igara koju je njegov nastavnik stvorio za razred. Unutar svake igre, riječi koje se traže su napisane na hrvatskom jeziku, a traže se u osmosmjerci na stranom jeziku. Također, u pozadini se mjeri vrijeme potrebno učeniku od trenutka kada pokrene igru do pritiska na tipku "Završi". Učenik može i odustati te se tada taj rezultat neće bilježiti. Učenik može i zatražiti pomoć te će mu se prikazati

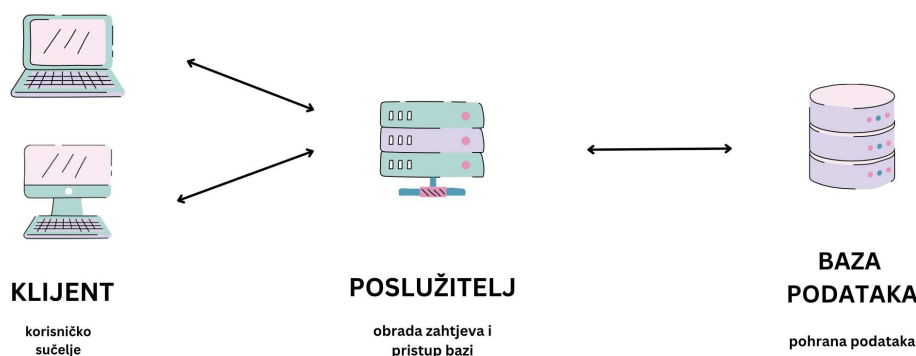
za neku od do sada nepronadenih riječi broj slova te prvo slovo i hrvatska riječ za koju je dana pomoć. Ukoliko je odlučio zaigrati igru čitanja s razumijevanjem, prvo će mu se prikazati tekst, a kada ga pročita prikazati će mu se pitanja i osmosmjerka u kojoj će tražiti odgovore. Učenik u svakom trenu ponovo može pogledati tekst, a ukoliko zatraži pomoć, prikazati će mu se pitanje za koje se pruža pomoć i rečenica ili dio rečenice u kojoj se nalazi odgovor.

Nastavnik za svaki razred i igru može vidjeti rezultate. U rezultatima mu se prikazuje zadnji rezultat za učenika (ukoliko je više puta igrao igru) odnosno vrijeme potrebno za rješavanje pojedinom učeniku i broj pronadenih riječi ili odgovorenih pitanja, ovisno o odabranoj igri. Također, nastavnik može i za cijeli razred vidjeti najčešće greške, što su u ovom slučaju riječi koje najveći broj učenika nije pronašao ili pitanja na koja je najmanji broj učenika pronašao odgovor.

Detaljnije o izgledu i funkcionalnostima razvijene aplikacije u poglavlju o korištenju aplikacije, a prije toga će biti napravljen pregled korištenih tehnologija i model podataka potreban za ostvarenje opisanog sustava.

3. Korišteni alati i tehnologije

U sklopu ovog završnog rada razvijena je web aplikacija za učenje vokabulara pomoću igre osmosmjerke. Aplikacija je implementirana kao raspodijeljeni sustav s klijent-poslužitelj arhitekturom [5]. Klijentska strana aplikacije (eng. frontend) radi interakciju s korisnikom. Služi za slanje korisničkih zahtjeva poslužitelju te prikazuje odgovore dobivene sa poslužitelja. Pozadinska strana aplikacije (eng. backend) je poslužitelj čija je zadaća komunicirati s bazom podataka. S njime korisnik ne komunicira izravno već isključivo preko korisničkog sučelja. Poslužiteljska strana sadrži logiku za obradu korisničkih zahtjeva te izvršavanja akcija sukladno tome (pohrana u bazu, slanje odgovora). Također, osmišljena je i implementirana relacijska baza podataka koja pohranjuje potrebne podatke. Pojednostavljeni prikaz arhitekture sustava je prikazan na slici (Slika 3.1).



Slika 3.1. Arhitektura sustava

Komunikacija između klijenata i poslužitelja ostvarena je HTTP protokolom. HTTP (eng. Hyper Text Transfer Protocol) je bezmemorijski protokol aplikacijskog sloja koji definira format i način razmjene poruka između poslužitelja i klijenta. Definirane su dvije vrste poruka: zahtjev i odgovor [6]. Klijent šalje poslužitelju zahtjev te je u njemu definirana metoda, resurs i parovi atribut-vrijednost potrebni za obradu zahtjeva. Najčešće su korištene HTTP metode GET za dohvaćanje resursa putem identificiranog URI-ja (eng.

Uniform Resource Identifier) te POST za slanje podataka koji će kreirati ili ažurirati resurs.

Za razvoj korisničkog sučelja korišteni su HTML (eng. HyperText Markup Language) za strukturu, CSS (eng. Cascading Style Sheets) za vizualnu prezentaciju i JavaScript za dodavanje dinamičnosti. HTML je označni jezik (eng. markup language) koji omogućava strukturiranje sadržaja pomoću oznaka (eng. tagova) [7]. CSS je jezik za stiliziranje HTML dokumenata, uključujući boju, fontove, margine i pozicioniranje [8]. JavaScript je skriptni programski jezik koji omogućava manipulaciju podacima poput promjena nad njima, izračuna ili provjera. Također, koristi se za ažuriranje i promjene HTML dokumenata [9]. Kako bi se olakšao razvoj korisničkog sučelja, korištena je React biblioteka. React je JavaScript biblioteka otvorenog koda razvijena od strane Facebooka koja služi za izradu interaktivnih korisničkih sučelja.

Za razvoj poslužiteljske strane korišten je jezik JavaScript u Node.js okruženju. Node.js omogućava izvršavanje JavaScript koda izvan preglednika, što je potrebno za izradu serverskih aplikacija. Kao radni okvir, korišten je Express. Express je minimalan i fleksibilan okvir aplikacije Node.js koji pruža jednostavan način definiranja ruta, rukovanja HTTP zahtjevima te upravljanja posredničkim funkcijama.

Za komunikaciju korisničkog sučelja s poslužiteljem te poslužitelja s OpenAI API-jem, korišten je axios. Axios je popularna JavaScript biblioteka koja omogućava slanje HTTP zahtjeva iz preglednika ili Node.js-a [10]. Pomoću axios-a, podaci se mogu dohvatiti ili poslati na server, a zatim prikazati u React komponentama, omogućavajući dinamičko i interaktivno korisničko sučelje.

Za pohranu podataka, napravljena je relacijska baza podataka. Za dohvaćanje, spremanje i promjene nad podacima, korišten je strukturni upitni jezik, SQL (eng. Structured Query Language). SQL je ANSI (eng. American National Standards Institute) i ISO (eng. International Organization for Standardization) standard koji omogućava pristup i manipulaciju bazama podataka [11]. Za upravljanje bazom podataka korišten je PostgreSQL sustav za relacijske baze podataka, zajedno s pgAdminom. pgAdmin je grafičko sučelje za upravljanje PostgreSQL bazama podataka, koje pruža korisnicima alate za pregled i manipulaciju podacima, vizualizaciju strukture baze te izvođenje naprednih SQL

upita.

Za razvoj klijentske i poslužiteljske strane aplikacija korišten je uređivač Visual Studio Code. Za potrebe testiranja, aplikacija je bila postavljena na poslužitelj Render.

U nastavku slijedi detaljniji pregled korištenih tehnologija u razvoju aplikacije.

3.1. React.js

React je popularna JavaScript biblioteka za izgradnju korisničkih sučelja na webu koju je razvio Facebook 2013. godine. Koristi se za izgradnju *single-page* stranica, odnosno web aplikacija koje se dinamički ažuriraju unutar jednog HTML dokumenta bez potrebe za ponovnim učitavanjem cijele stranice.

Jedna od ključnih karakteristika React-a je korištenje komponenata. Korisničko sučelje se sastoji od manjih, ponovljivih dijelova: komponenata. Svaka komponenta ima svoje stanje (eng. *state*) i logiku. One se mogu kombinirati i ponovno koristiti, što omogućava modularan, lako čitljiv i održiv kod. Komponente se pišu u JSX sintaksi koja spaja JavaScript i XML u jednu cjelinu [12]. Njome je omogućeno pisanje HTML elemenata izravno unutar JavaScript koda. Takva sintaksa čini komponente čitljivijima i ekspresivnijima. Druga bitna karakteristika Reacta je korištenje virtualnog DOM-a. DOM (eng. Document Object Model) je programsko sučelje za HTML i XML dokumente koje predstavlja strukturu dokumenta kao hijerarhiju čvorova. Omogućuje programerima da pomoću JavaScript-a dinamički pristupaju, mijenjaju i manipuliraju sadržajem, stilovima i strukturom web stranice u pregledniku. U standardnim aplikacijama, svaka promjena zahtjeva promjenu cijelog stvarnog DOM-a, što znači da preglednik svaki put mora ponovno izračunati izgled stranice i prikazati sve elemente. Kako bi se izbjegao taj dug i često neefikasan proces, React uvodi virtualni DOM. Kada dođe do promjene u stanju komponente, React prvo ažurira virtualni DOM, koji je mnogo brži i lakši za manipulaciju od stvarnog DOM-a. Nakon toga, uspoređuje promjene u virtualnom DOM-u sa stvarnim DOM-om i samo nužne promjene se primjenjuju na stvarni DOM [13]. Takav pristup omogućuje brže i učinkovitije ažuriranje korisničkog sučelja, što je posebno važno kod aplikacija s puno interakcija i promjena stanja.

Zahvaljujući svojim dobrim performansama i intuitivnoj sintaksi, React je postao vrlo popularna biblioteka za izgradnju korisničkih sučelja.

3.2. Node.js

Node.js je JavaScript okruženje otvorenog koda. On omogućava da se JavaScript kod izvršava izvan preglednika, što ga čini prikladnim za pisanje poslužiteljske strane aplikacija [14]. Njime je omogućeno korištenje programskog jezika JavaScript za stvaranje cijele aplikacije, poznato kao paradigma "JavaScript svugdje". Izgrađen je na V8 stroju te je višeplatformsko okruženje, odnosno kompatibilan je s različitim operacijskim sustavima. Node.js ima arhitekturu vođenu događajima i temelji se na asinkronom modelu izvršavanja. Takva arhitektura omogućava učinkovito upravljanje velikim brojem paralelnih zahtjeva, dok asinkronost osigurava mogućnost istovremenog izvršavanja više zadataka. Node.js, kojeg je 2009. godine kreirao Ryan Dahl, postao je popularan zbog svoje brzine, skalabilnosti i jednostavnosti korištenja. Kako bi se omogućilo upravljanje dodatnim programskim paketima i bibliotekama, 2010. godine uveden je npm (eng. Node Packet Manager), upravljač paketima za Node.js. On omogućava jednostavno upravljanje paketima trećih strana, uključujući njihovu instalaciju, deinstalaciju i ažuriranje te programerima olakšava razmjenu, objavljivanje i dijeljenje izvornog koda.

3.3. Express

Express je brz, minimalan i fleksibilan web okvir za Node.js. Dizajniran je kako bi olakšao pisanje poslužiteljskih aplikacija i aplikacijsko programskih sučelja, API-ja (eng. Application Programming Interface). Express pruža jednostavan način definiranja ruta, upravljanja HTTP zahtjevima i odgovorima te upravljanja *middleware* funkcijama [15]. Temeljni koncept Expressa jest usmjeravanje. Njime je definirano kako će aplikacija odgovoriti na određeni HTTP zahtjev. Ruta je kombinacija putanje i HTTP metode te je svaka ruta povezana s jednom ili više funkcija koje se nazivaju posredničkim funkcijama (eng. middleware). Zahvaljujući svojoj jednostavnosti i intuitivnosti, Express je idealan za izgradnju manjih aplikacija poput one napravljene u sklopu ovog završnog rada.

3.4. OpenAI API

Tvrtka OpenAI razvija napredne modele umjetne inteligencije za obavljanje brojnih zadataka te pruža API-je koji omogućuju programerima integraciju tih modela u aplikacije

koje razvijaju. OpenAI API-ji pružaju pristup širokom spektru modela umjetne inteligencije razvijenih za različite zadatke obrade prirodnog jezika i strojne obrade. Većinski se razvijeni modeli temelje na GPT arhitekturi (eng. Generative Pre-trained Transformer). Ta se arhitektura koristi za razne zadatke obrade prirodnog teksta, poput generiranja teksta, odgovaranja na pitanja, sažimanja dokumenata i slično. OpenAI također razvija i modele koji se ne temelje na GPT arhitekturi, od kojih su najpoznatiji DALL-E, model za generiranje slika na temelju tekstualnih opisa te Whisper, model za prepoznavanje govora koji omogućuje transkripciju audiozapisa u tekst [16].

Među GPT modelima, najpopularniji za generiranje teksta u prirodnom jeziku jest ChatGPT, koji trenutno koristi GPT-4 verziju. No u sklopu razvijene aplikacije, ostvareno je spajanje na API modela GPT-3.5-turbo. Iako je to nešto starija verzija, sasvim je dovoljna kako bi se omogućilo generiranje pitanja o određenom tekstu na odabranom stranom jeziku [17]. Ova funkcionalnost implementirana je na serverskoj strani s koje je slan zahtjev generativnom alatu koji koristi napredne mogućnosti obrade prirodnog jezika za postavljanje pitanja relevantnih za uneseni sadržaj.

Za slanje zahtjeva na OpenAI API iz Node.js okruženja, korištena je axios biblioteka, prethodno instalirana korištenjem Node Packet Managera. Kako bi se API mogao integrirati unutar aplikacije, potrebno je prethodno zatražiti tajni ključ koji će se koristiti za autentifikaciju i autorizaciju zahtjeva. Ključ je unutar koda pohranjen u *.env* datoteci kako bi se očuvala njegova tajnost prilikom javnog dijeljenja koda te njegove pohrane u repozitorij.

```
const response = await axios.post(apiUrl, {
  model: model,
  messages: [
    { role: 'user', content: prompt }
  ],
  max_tokens: 50,
  temperature: 0,
}, {
  headers: {
    'Content-Type': 'application/json',
    'Authorization': Bearer ${apiKey},
  },
});
```

Kôd 3.1: Slanje zahtjeva API-ju za generiranje teksta

Implementacija slanja zahtjeva OpenAI API-ju može se vidjeti u kôdu (Kôd 3.1). Pri-
kazan je dio asinkrone funkcije *chatCall* unutar koje je napravljen HTTP POST zahtjev
na API *endpoint* za generiranje teksta, koristeći *axios* biblioteku. Unutar poziva su defi-
nirani parametri koji se šalju:

- **model** - parametar koji specificira koji se GPT model koristi za generiranje odgo-
vora, u ovom je radu korišten "gpt-3.5-turbo"
- **messages** - polje objekata koje predstavlja tok razgovora, svaki objekt ima ulogu
(eng. role) koja je u ovom slučaju korisnik (eng. user) i sadržaj (eng. content), u
ovom radu je to upit za generiranjem željenog broja pitanja, odgovora i pomoći za
zadani tekst pohranjen u varijablu *prompt*
- **max_tokens** - ograničenje broja tokena (riječi ili dijelova riječi) koje API može
generirati u odgovoru
- **temperature** - parametar koji kontrolira kreativnost odgovora, što je bliže nuli, to
su odgovori konzervativniji, odnosno sadrže predvidljive riječi s obzirom na dani
kontekst

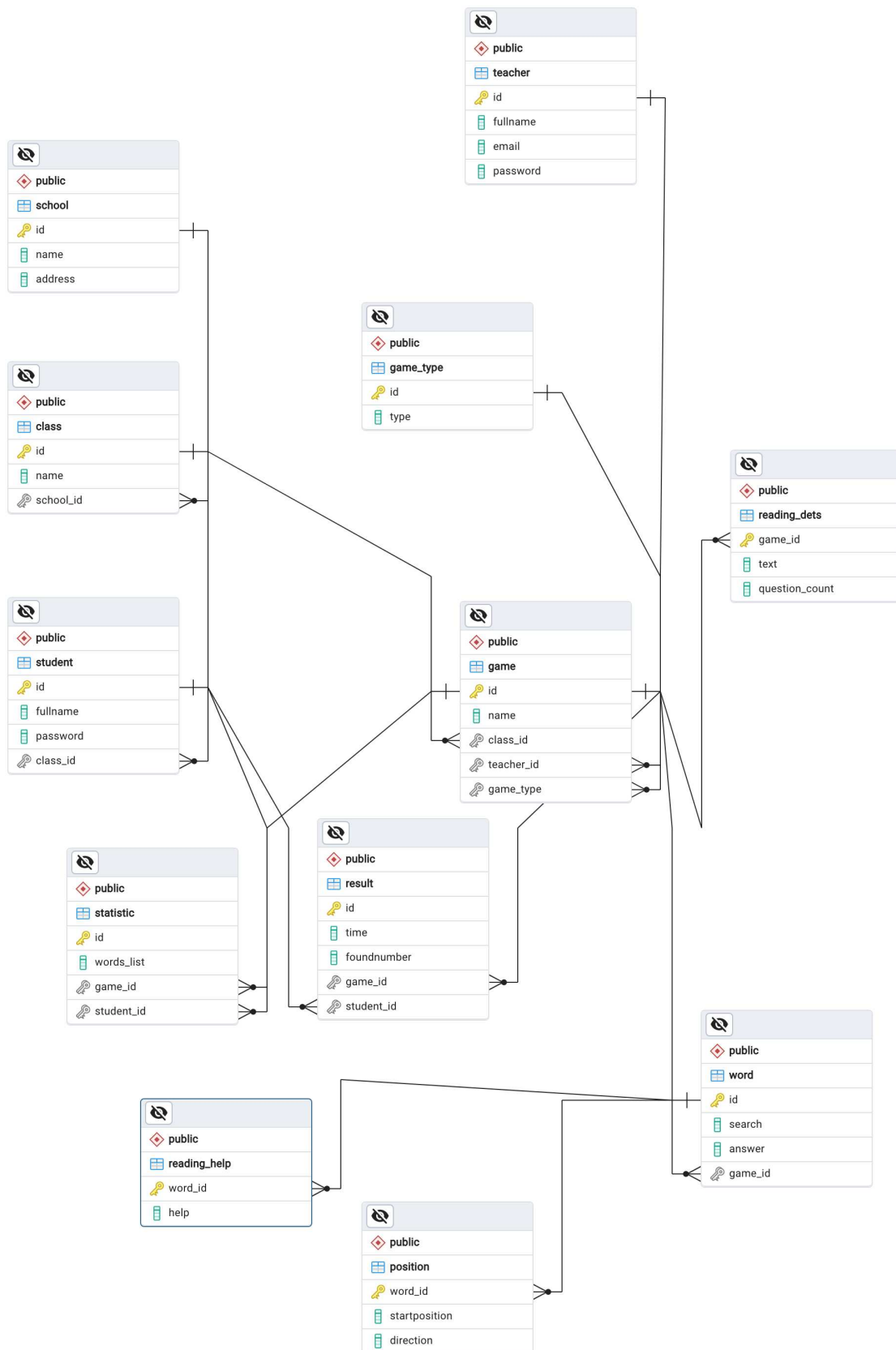
Osim parametara modela, potrebno je postaviti i zaglavlja (eng. headers) zahtjeva. U
njima se nalaze podatci o vrsti sadržaja (eng. Content Type) te autorizacijski token, od-
nosno tajni API ključ, korišten za autentifikaciju i autorizaciju pristupa API-ju.

3.5. PostgreSQL

Za upravljanje bazom podataka korišten je PostgreSQL, objektno-relacijski sustav baza
podataka otvorenog koda [18]. Napisan je u programskom jeziku C te zahvaljujući svojim
performansama postiže veliku popularnost. PostgreSQL koristi i proširuje jezik SQL te
pri provođenju transakcija poštuje ACID svojstva (atomarnost, konzistentnost, izolacija
i izdržljivost/durabilnost) [19]. Radi lakšeg upravljanja bazom, korišten je pgAdmin,
grafičko sučelje koje olakšava pregled i manipulaciju podacima te izvođenje SQL upita.

4. Model podataka

Sukladno potrebama aplikacije, osmišljen je i napravljen pripadni model baze podataka. Model je relacijski što znači da se sastoji od imenovanih dvodimenzionalnih tablica (relacija). Pristup bazi podataka imaju nastavnici i učenici. Nastavnici imaju mogućnost kreiranja zadataka: klasičnih igara osmosmjerke i čitanja s razumijevanjem, dodavanja učenika i promjene njihovih lozinki. Također mogu pregledavati igre koje su kreirali i rezultate igara po razredima. Učenici nakon prijave mogu igrati igre predviđene za njihov razred te se njihovi rezultati spremaju u bazu. Pohranjuje se najnoviji predani pokušaj rješavanja te za svakog učenika ono što nije riješio unutar pojedine igre. Trenutno postoje dvije vrste igara te su kreirane potrebne tablice za njihovo kreiranje, prikaz i rješavanje. Na slici 5.1 prikazan je relacijski model baze podataka izvezen iz pgAdmin-a.



Slika 4.1. Relacijski model baze podataka

4.1. Opisi relacija

U slijedećim tablicama prikazani su opisi pojedinih relacija iz baze podataka, zajedno s atributima, njihovim nazivima, tipovima podataka i kratkim opisima njihove uloge. Atributi označeni s PK (eng. Primary Key) predstavljaju primarne ključeve, koji služe kao jedinstveni identifikatori svake relacije. Ovi ključevi osiguravaju da su sve vrijednosti u atributu jedinstvene i neponovljive unutar tablice. Atributi označeni s FK (eng. Foreign Key) predstavljaju strane ključeve, koji povezuju vrijednosti jednog atributa u tablici s odgovarajućim atributima u drugoj tablici, omogućujući tako povezivanje podataka između različitih relacija i održavanje referencijalnog integriteta u bazi podataka.

school

Relacija school sadrži podatke vezane za škole upisane u sustav. Uz jedinstveni identifikator škole, pohranjen je njihov naziv i adresa. Jedinstveni identifikator škole je primarni ključ te se sam generira prilikom unosa zapisa u relaciju. Relacija je prikazana tablicom 4.1.

Tablica 4.1. Tablični prikaz relacije school

Varijabla	Tip podataka	Opis
id (PK)	integer	identifikator škole
name	character varying	ime škole
address	character varying	adresa škole

class

Relacija class sadrži podatke vezane uz pojedinačne razrede: njihov jedinstveni identifikator, koji je ujedno i primarni ključ, naziv te strani ključ school_id. On povezuje razred sa školom kojoj pripada. Naziv razreda je oblika "1.A", odnosno sastoji se od broja i slova. Opisana relacija je prikazana tablicom 4.2.

Tablica 4.2. Tablični prikaz relacije class

Varijabla	Tip podataka	Opis
id (PK)	integer	jedinstveni identifikator razreda
name	character varying	naziv razreda
school_id (FK)	integer	identifikator škole kojoj razred pripada

student

Relacija sadrži podatke vezane za učenike. Primarni ključ relacije je jedinstveni identifikator učenika: id. Uz to, tablica sadržava puno ime učenika, odnosno njegovo ime i prezime, lozinku potrebnu za prijavu u aplikaciju i identifikator razreda kojem učenik pripada. U bazu podataka učenike dodaje nastavnik. Lozinka je zapis koji se sastoji od naziva triju životinja odvojenih zarezom. Ona se sama generira prilikom dodavanja novog učenika u razred te je moguće izraditi novu lozinku za svakog učenika. Relacija student je prikazana tablicom (Tablica 4.3).

Tablica 4.3. Tablični prikaz relacije student

Varijabla	Tip podataka	Opis
id (PK)	integer	jedinstveni identifikator korisnika(učenika)
fullname	character varying	ime i prezime učenika
password	character varying	lozinka dodijeljena učeniku
class_id (FK)	integer	identifikator razreda koji učenik pohađa

teacher

Relacija teacher sadrži podatke značajne za nastavnike, uglavnom potrebne za njihovu prijavu u sustav. Uz email i lozinku kojima se nastavnik prijavljuje, također sadrži njihovo ime i prezime te jedinstveni identifikator. Jedinstveni se identifikator sam generira prilikom stvaranja zapisa u relaciji. Relacija je prikazana tablicom (Tablica 4.4).

Tablica 4.4. Tablični prikaz relacije teacher

Varijabla	Tip podataka	Opis
id (PK)	integer	identifikator korisnika(nastavnika)
fullname	character varying	ime i prezime korisnika (nastavnika)
email	character varying	email korisnika (nastavnika)
password	character varying	korisnikova lozinka

game

Relacija koja sadrži podatke o igri. Primarni ključ relacije je broj koji se automatski generira pri unošenju zapisa u relaciju. Uz njega, tablica sadrži naziv igre kao atribut name te tri strana ključa s kojima je povezana s drugim relacijama. Atribut class_id sadrži informaciju o razredu kojem je igra namijenjena, a teacher_id koji je nastavnik kreirao igru. Informacija o vrsti igre je pohranjena u atributu game_type, koji je strani ključ prema relaciji game_type. Te su informacije potrebne za izlist i odabir igre kod nastavnika i učenika. Relacija je prikazana tablicom 4.5.

Tablica 4.5. Tablični prikaz relacije game

Varijabla	Tip podataka	Opis
id (PK)	integer	jedinstveni identifikator igre
name	character varying	naziv igre
class_id (FK)	integer	identifikator razreda
teacher_id (FK)	integer	identifikator nastavnika
game_type (FK)	integer	identifikator vrste igre

game_type

Relacija koja služi za pohranu mogućih vrsta igara. Primarni ključ je broj koji se automatski generira prilikom unosa zapisa u relaciju te se uz njega još pohranjuje i naziv vrste igre. Prikazana je tablicom 4.6.

Tablica 4.6. Tablični prikaz relacije game_type

Varijabla	Tip podataka	Opis
id (PK)	integer	jedinstveni identifikator vrste igre
type	character varying	naziv vrste igre

reading_dets

Relacija koja pohranjuje dodatne podatke za igre čitanja s razumijevanjem. Strani ključ game_id je ujedno i primarni ključ te brisanjem zapisa iz relacije game će se obrisati odgovarajući zapis u relaciji reading_dets. U relaciji su pohranjeni još i tekst igre te broj postavljenih pitanja. Opisana relacija je prikazana tablicom (Tablica 4.7).

Tablica 4.7. Tablični prikaz relacije reading_dets

Varijabla	Tip podataka	Opis
game_id (PK, FK)	integer	identifikator igre
text	text	tekst igre
question_count	integer	broj postavljenih pitanja

word

Relacija sadrži podatke o parovima riječi koji se pojavljuju u igri osmosmjerke, odnosno parovima pitanje-odgovor u igri čitanja s razumijevanjem. Osim automatski generiranog identifikatora svakog para, relacija sadrži i sam par. Pojam na proizvoljno izabranom stranom jeziku/odgovor (atribut answer) tražiti će se u osmosmjerci, a hrvatske riječi/pitanja (atribut search) će biti na izlistane na popisu sa strane. Unutar jedne igre je moguće imati maksimalno osam parova riječi ili šest pitanja. Atribut game_id je strani ključ koji povezuje svaku zapis u relaciji sa igrom kojoj pripada. Relacija je prikazana tablicom 4.8.

Tablica 4.8. Tablični prikaz relacije word

Varijabla	Tip podataka	Opis
id (PK)	integer	jedinstveni identifikator para
search	character varying	hrvatski pojam ili pitanje
answer	character varying	strani pojam ili odgovor
game_id (FK)	integer	identifikator igre kojoj pripada

reading_help

Ukoliko se radi o igri čitanja s razumijevanjem, pohranjuje se i pomoć za svako pitanje. Ona je pohranjena u relaciji reading_help uz identifikator pojma u tablici word na koji se odnosi. Prilikom brisanja zapisa iz relacije word, obrisati će se i odgovarajući zapis u relaciji reading_help. Opisana relacija je prikazana tablicom (Tablica 4.9).

Tablica 4.9. Tablični prikaz relacije reading_help

Varijabla	Tip podataka	Opis
id (PK, FK)	integer	identifikator pitanja
help	character varying	pomoć za pitanje

position

Relacija position pohranjuje podatke o položaju riječi unutar jedne igre. Strani ključ word_id je ujedno i primarni ključ. Uz njega, relacija sadrži početnu poziciju riječi u obliku "x,y" te jedan od četiri moguća smjera: leftright, updown, diagonalup ili diagonaldown. Podatci o smještaju riječi se stvaraju i u relaciju pohranjuju pri kreiranju igre te će za svakog učenika riječi koje se traže biti na istoj poziciji unutar mreže. Prilikom brisanja zapisa u relaciji word, obrisati će se i odgovarajući zapis u relaciji position. Relacija je prikazana tablicom 4.10.

Tablica 4.10. Tablični prikaz relacije position

Varijabla	Tip podataka	Opis
word_id (PK, FK)	integer	identifikator riječi
startposition	character varying	x i y koordinata početka riječi
direction	character varying	smjer riječi

result

Relacija opisuje rezultat rješavanja određene igre za određenog studenta. Primarni ključ relacije je automatski generirani identifikator rezultata te se uz njega pohranjuje broj pronađenih riječi i vrijeme potrebno za rješavanje igre. Tablica rezultata je stranim ključem povezana s učenicom čiji je rezultat i igrom na koju se rezultat odnosi. Kombinacija učenika i igre također jedinstvena, što osigurava da je za određenu igru pojedinog učenika pohranjen samo jedan rezultat i to je posljednji rezultat. Opisana relacija je prikazana tablicom (Tablica 4.11).

Tablica 4.11. Tablični prikaz relacije result

Varijabla	Tip podataka	Opis
id (PK)	integer	jedinstveni identifikator rezultata
time	double precision	vrijeme rješavanja u sekundama
foundnumber	integer	broj pronađenih riječi
game_id (FK)	integer	identifikator igre za koju je rezultat zapisan
student_id (FK)	integer	identifikator studenta čiji je rezultat zapisan

statistic

Relacija pohranjuje podatke iz kojih će se napraviti statistika po razredima, odnosno nastavnik će dobiti podatak o riječima koje najmanji broj učenika pronađe u osmosmjerci. Uz jedinstveni identifikator same statistike te učenika i igre, pohranjuje se lista zarezom odvojenih nepronađenih riječi ili neodgovorenih pitanja. Kombinacija učenika i igre je jedinstvena čime je osigurano da se čuva samo posljednji rezultat učenika za pojedinu igru. Relacija je prikazana tablicom (Tablica 4.12).

Tablica 4.12. Tablični prikaz relacije statistic

Varijabla	Tip podataka	Opis
id (PK)	integer	jedinstveni identifikator
words_list	character varying	popis nepronadenih riječi ili neodgovorenih pitanja
game_id (FK)	integer	identifikator igre
student_id (FK)	integer	identifikator studenta

5. Implementacijski detalji

Centralni dio razvijene aplikacije jest igra osmosmjerke. Kako bi se omogućilo stvaranje osmosmjerki, nužna je implementacija algoritma za raspored riječi unutar mreže slova. Raspoređivanje riječi ostvareno je na poslužitelju te je smještaj riječi pohranjen u bazi podataka. Time je omogućeno da su zadane riječi za sve učenike na istom mjestu unutar mreže slova te uvijek na jednakom mjestu unutar jedne igre. Ostala se slova slučajnim odabirom generiraju na klijentskoj strani aplikacije te su različita za svakog učenika i drugačija u svakom pokušaju rješavanja. Algoritam radi za proizvoljne dimenzije mreže, no u ovoj je aplikaciji veličina fiksirana na dimenzije 10x10. U nastavku slijedi detaljan opis algoritma za raspored riječi unutar mreže slova.

5.1. Implementacija algoritma za raspored riječi

Funkcija koja smješta riječi u mrežu slova prima listu riječi koje je potrebno smjestiti. Nastavnicima je pri kreiranju igre naglašeno kako riječi koje zadaju ne smiju imati više od deset slova kako bi se mogle smjestiti. Algoritam prije početka smještanja riječi kreira polje dimenzija 10x10 ispunjeno nulama. Zatim uzima riječ po riječ iz liste te ih pokušava smjestiti. Ukoliko uspije, riječ se upisuje u polje umjesto nula na tom mjestu, kako bi se ubuduće mogla pratiti preklapanja slova. Pokušaj smještanja riječi započinje slučajnim odabirom smjera i početne pozicije. Za svaki smjer su definirani pomaci po x i y osi. Kodom 5.1 prikazan je odabir smjera i početne pozicije.

Nakon odabira smjera i početnog položaja, računa se pozicija na kojoj će riječ završavati ako ju se na taj način smjesti. Ukoliko kraj riječi ne prelazi granice mreže kada se postavi u odabranom smjeru, jedan je uvjet za smještaj riječi ispunjen. Drugi uvjet za smještanje riječi je provjeriti preklapanja slova s drugim riječima, ukoliko je do njih došlo. Riječ se smije s drugim riječima preklapati samo ako se radi o istom slovu.

```

const orientations = ["leftright", "updown", "diagonalup", "diagonaldown"];
for(let i = 0; i < foreignWords.length; i++) {
  const word = foreignWords[i];

  var orientation = '';
  let uspjeh = false;
  while(!uspjeh) {
    const rand = Math.floor(Math.random() * orientations.length);
    orientation = orientations[rand];
    var step_x = 0, step_y = 0;

    if(orientation === "leftright") {
      step_x = 1;
      step_y = 0;
    } else if(orientation === "updown") {
      step_x = 0;
      step_y = 1;
    } else if(orientation === "diagonalup") {
      step_x = 1;
      step_y = -1;
    } else if(orientation === "diagonaldown") {
      step_x = 1;
      step_y = 1;
    }
  }
  var start_x = Math.floor(Math.random() * n);
  var start_y = Math.floor(Math.random() * m);
}

```

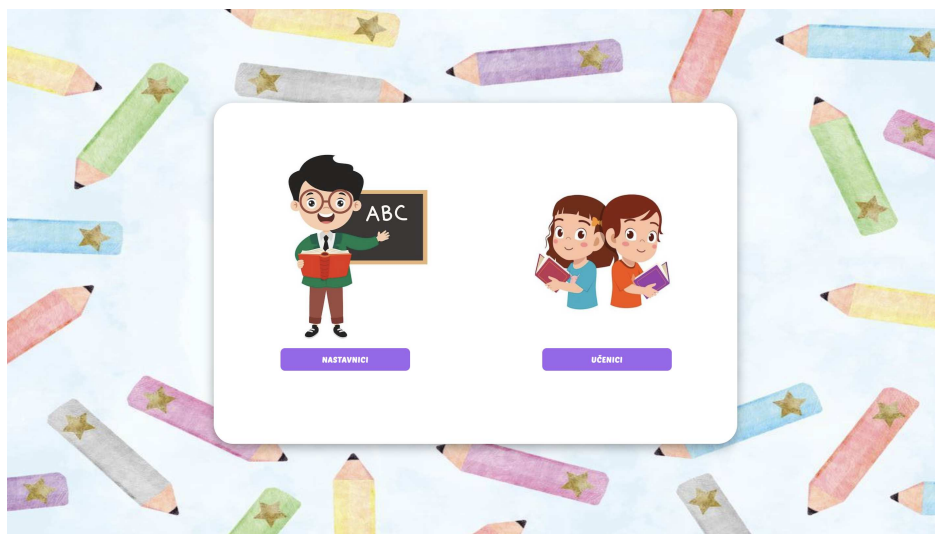
Kôd 5.1: Odabir smjera i početne pozicije

Ukoliko se riječ može smjestiti, upisuje ju se u mrežu slovo po slovo te se pamti njena početna pozicija i smjer. Ukoliko iz bilo kojeg razloga (nedozvoljeno preklapanje slova ili prekoračenje veličine mreže) riječ nije uspješno smještena, postupak će se za nju ponavljati dok ju ne smjesti uspješno. Kada su sve riječi smještene u mrežu, u bazu će se upisati njihove početne pozicije i smjer kojim su položene u mrežu.

Ovaj algoritam ne garantira da će sve riječi biti uspješno smještene. Jednom postavljena riječ ostaje na svojoj poziciji i više se ne premješta radi smještanja drugih riječi. Ako se neka riječ postavi na nepovoljno mjesto, može se dogoditi da preostale riječi ne budu mogle biti smještene, budući da algoritam ne uključuje mehanizam povratka (eng. backtracking) kako bi se razmotrile ranije opcije. Usprkos tome, algoritam je prilično uspješan stoga je odabran za implementaciju.

6. Korištenje aplikacije

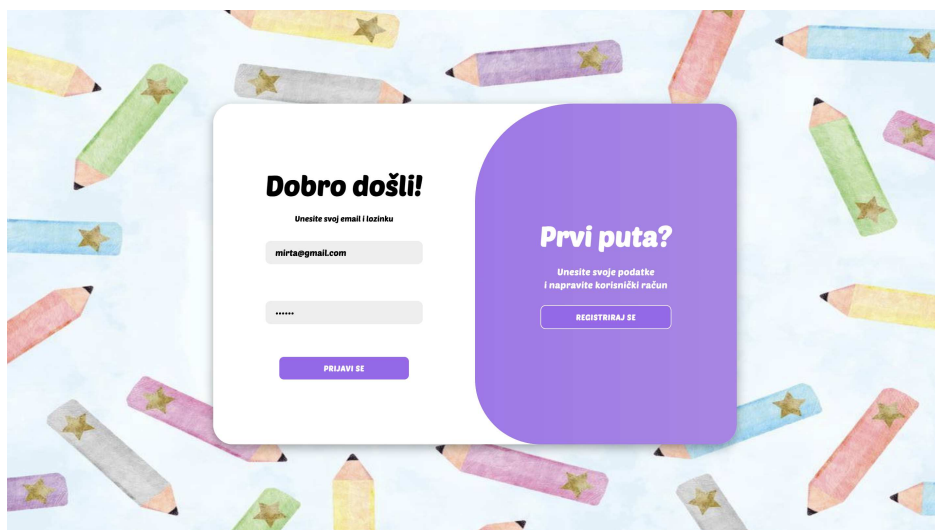
Aplikacija se sastoji od dva dijela: administratorskog, odnosno onog za nastavnike te korisničkog, odnosno onog za učenike. Nastavnici svoju aplikaciju mogu koristiti isključivo na računalu, dok je aplikacija za učenike prilagođena korištenju i na računalu i na tabletu. Na početnom zaslonu aplikacije potrebno je izabrati želite li ju koristiti kao nastavnik ili učenik te se prema odabranoj ulozi otvara stranica prijave. Početni zaslon aplikacije je prikazan na slici 6.1.



Slika 6.1. Početna stranica aplikacije s izbornikom uloge

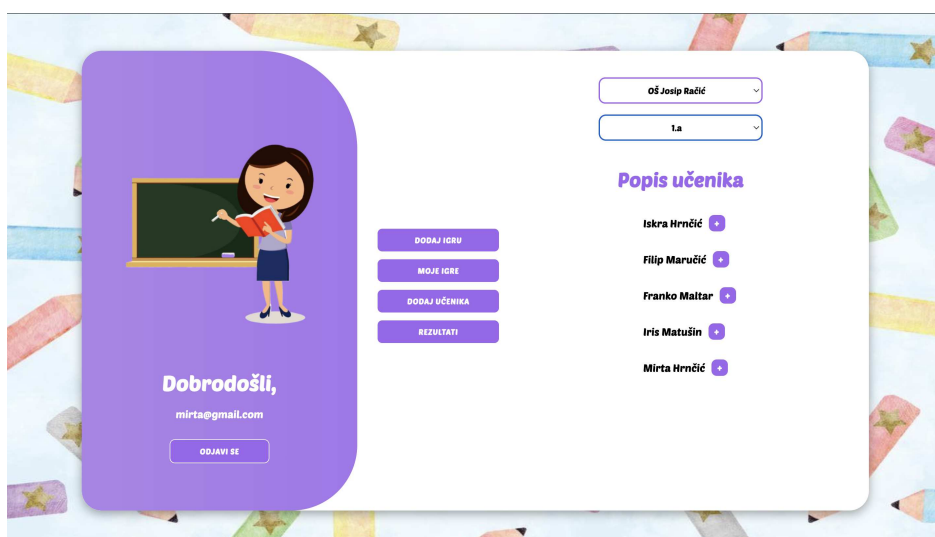
6.1. Sučelje za nastavnike

Nakon odabira uloge nastavnika, otvara se stranica prijave. Potrebno je upisati email adresu i lozinku za uspješnu prijavu. Ukoliko nisu navedeni ispravni podatci, korisnik će o tome biti obaviješten crvenim tekstom "Neispravan email ili lozinka". Stranica za prijavu nastavnika u sustav prikazana je slikom 6.2.



Slika 6.2. Prijava nastavnika u aplikaciju

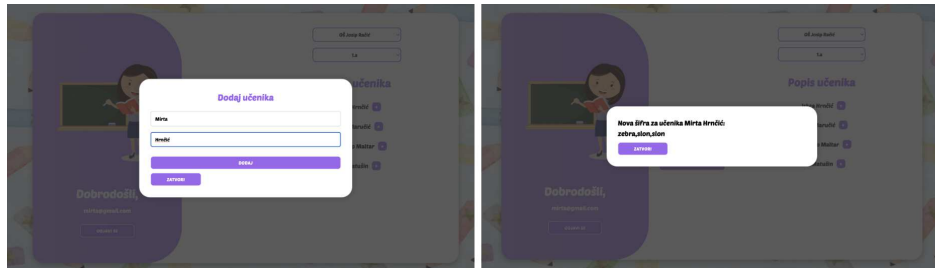
Uspješnom prijavom korisnik se preusmjerava na stranicu nastavnika. Ona je prikazana slikom 6.3. Nastavnik ima ulogu administratora i pripadajuće ovlasti. Odabirom škole i razreda, prikazuje se popis učenika u izabranom razredu. Za svakog učenika moguće je izraditi novu lozinku pritiskom na tipku "+" pored imena učenika. Ona će se zatim nastavniku prikazati u modalu kako bi ju mogao pročitati učeniku. Nakon pritiska tipke "Zatvori" nastavnik neće više moći pristupiti lozinci, no moći će za učenika zatražiti novu lozinku, ukoliko za time bude potrebe.



Slika 6.3. Nastavnikova stranica s učitanim popisom učenika u razredu

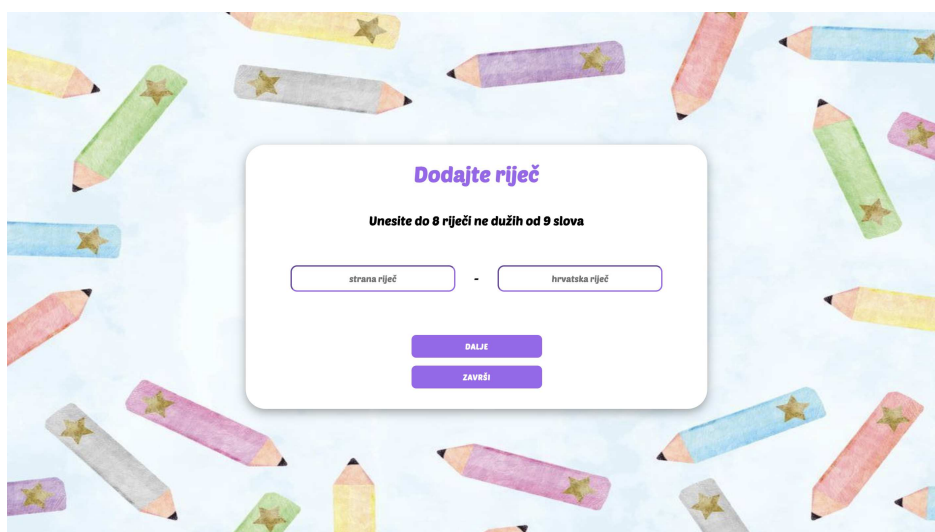
Također, nakon odabrane škole i razreda, nastavnik može dodati novog učenika u

izabrani razred. Nakon unosa njegovog imena i prezimena u otvoreni modal, nastavniku će se prikazati generirana lozinka za učenika. Dodavanje učenika u razred i generiranje lozinke prikazani su slikom 6.4.



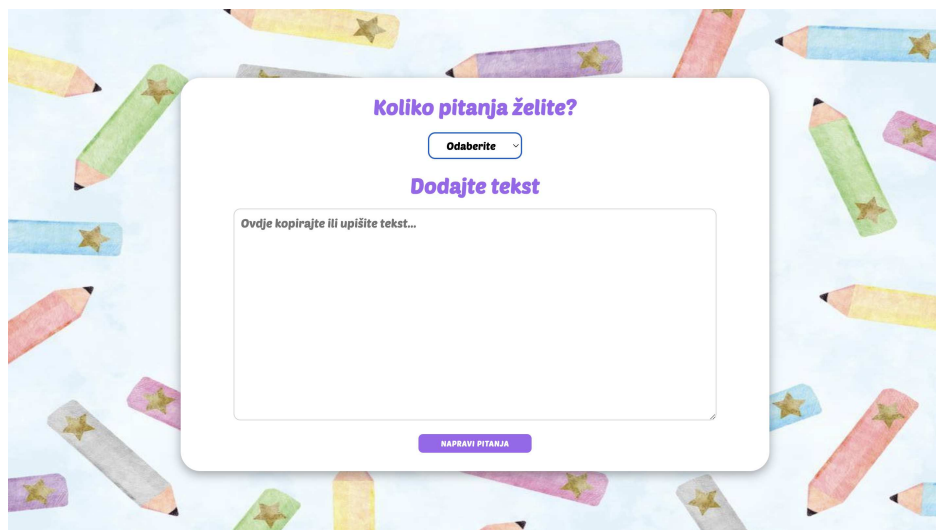
Slika 6.4. Dodavanje učenika u razred

Osim upravljanja učeničkim računima, nastavnik može kreirati igre. Odabirom opcije "Kreiraj igru", otvara se stranica s formom na kojoj se odabiru škola i razred, a zatim unese željeno ime nove igre. Zatim je potrebno odabrati vrstu igre koju se kreira: klasična osmosmjerka ili igra čitanja s razumijevanjem gdje će se odgovori na pitanja o tekstu tražiti unutar osmosmjerke. Ovisno o odabiru, otvoriti će se prikladna forma za kreiranje igre. Ukoliko se odabere prva verzija, otvara se forma u koju se unosi par: hrvatska - strana riječ. Moguće je unijeti maksimalno osam riječi te svaka riječ može imati najviše 10 slova. Ukoliko nastavnik želi stvoriti igru s manje od osam riječi, nakon željenog broja parova pritisne tipku "Završi" umjesto "Dalje". Dolaskom do osme riječi, jedino je tipka "Završi" ponuđena. Forma za unos parova riječi prikazana je slikom 6.5.



Slika 6.5. Kreiranje igre osmosmjerke

Ukoliko se odabere druga verzija igre, odnosno igra čitanja s razumijevanjem, otvara se forma s dva elementa: padajući izbornik za odabir željenog broja pitanja i polje za unos teksta. Moguće je postaviti između tri i šest pitanja. Nakon unosa potrebnih podataka i predaje forme, sa poslužitelja se šalje zahtjev generativnom alatu umjetne inteligencije da postavi određen broj pitanja o zadanom tekstu koja imaju odgovor od jedne riječi s maksimalno deset slova. Također se kreira i pomoć koju će učenici moći zatražiti: rečenica ili dio rečenice u kojoj se nalazi odgovor na pojedino pitanje. Pitanja i odgovori se zatim prosljeđuju nastavniku, te ukoliko je zadovoljan, kreira se igra. Forma za stvaranje igre čitanja s razumijevanjem prikazana je slikom 6.6.

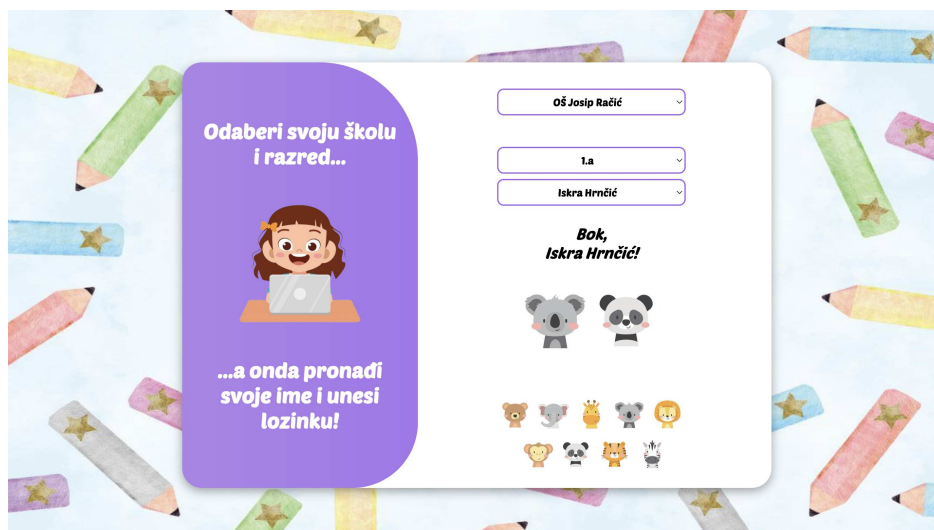


Slika 6.6. Kreiranje igre čitanja s razumijevanjem

Nastavnik također ima mogućnost pregledavanja svojih igara.

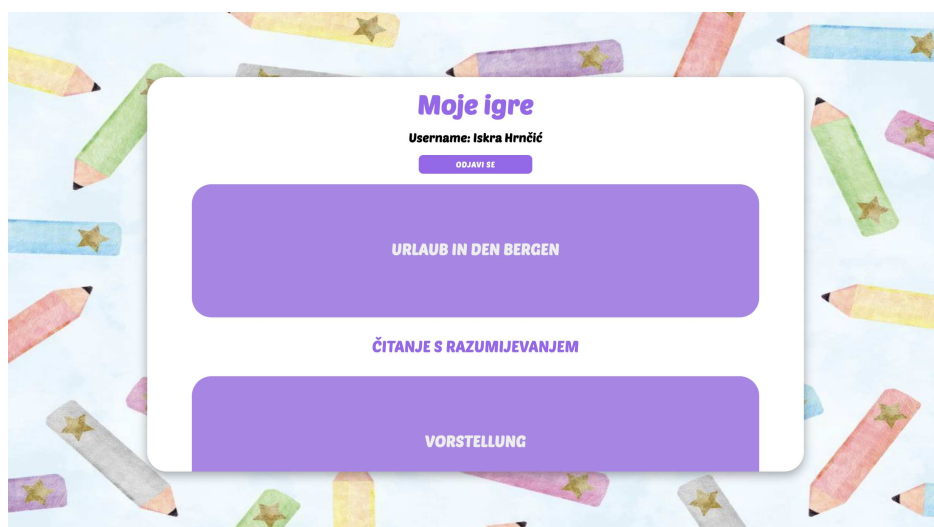
6.2. Aplikacija za učenike

Ukoliko se na početnom zaslonu odabere opcija korištenja aplikacije kao učenik, otvara se forma za prijavu učenika. Ona je prikazana slikom 6.7. S ciljem maksimalnog pojednostavljenja prijave učenicima, oni nikada ne upisuju tekst već svoju školu, razred te ime i prezime biraju iz padajućih izbornika. Nakon uspješnog odabira, moraju "upisati" svoju lozinku, odnosno ispravnim redoslijedom odabrati tri od devet životinja koje su im dodijeljene. Ukoliko pogriješe, porukom ih se obavijesti te mogu ispočetka unjeti lozinku.



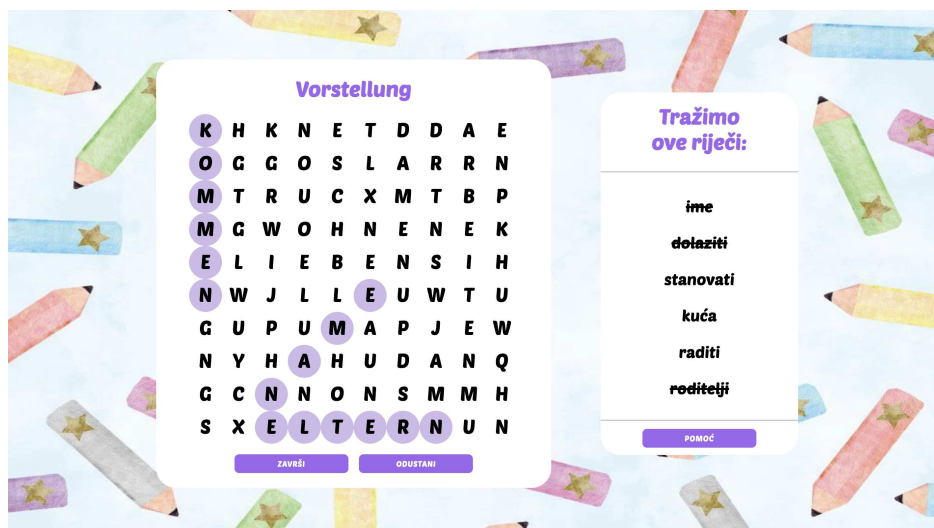
Slika 6.7. Prijava učenika

Nakon uspješne prijave, učenici se nalaze na svojoj stranici gdje su im vidljive sve igre koje je nastavnik kreirao za njihov razred. Igre su podijeljene u dvije skupine: klasične osmosmjerke i čitanje s razumijevanjem. Učeničeva stranica prikazana je slikom 6.8.



Slika 6.8. Učeničeva stranica s izlistom dostupnih igara

Nakon odabira igre klasične osmosmjerke, otvara se stranica koja sadrži mrežu slova dimenzija 10x10 u kojoj se traže riječi te desno od nje popis riječi na hrvatskom jeziku čija se strana inačica traži. Zaslone igre je prikazan slikom 6.9. Također je dostupna opcija pomoći. Kada započne igra, kreće mjerenje vremena. Kako aplikacija za učenike radi na tabletu i računalu, riječi se mogu tražiti klikom miša ili povlačenjem prsta preko slova.



Slika 6.9. Igranje igre klasične osmosmjerke

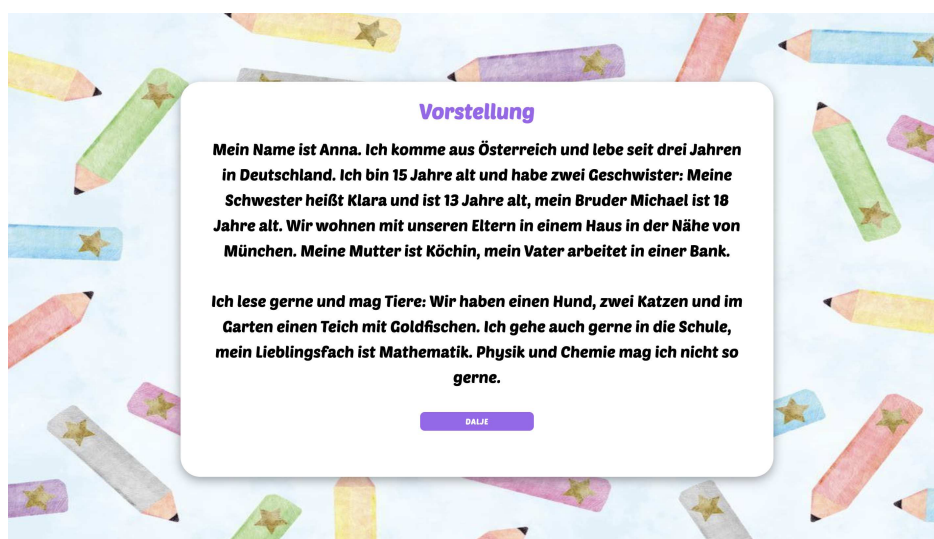
Ukoliko je pronađena neka od riječi s popisa, ona u mreži ostaje zaokružena svijetlo ljubičastim kružićima te prekrivena na popisu. Ukoliko je pri rješavanju igre zatražena pomoć, otvara se modal u kojem su napisani: hrvatska inačica riječ za koju je prikazana pomoć, prvo slovo strane inačice riječi te broj slova u njoj. Pritiskom na tipku "Zatvori", učenik se vraća igranju igre. Ukoliko učenik želi završiti igru, pritiskom na tipku "Završi" zaustavlja se mjerenje vremena i predaju njegovi rezultati: broj pronađenih riječi, potrebno vrijeme za rješavanje te popis riječi koje nije pronašao. Učenika se preusmjerava na stranicu s kratkom proslavom njegovog uspjeha (prikazana slikom 6.10) te po isteku sedam sekundi, vraća se na svoju stranicu s popisom dostupnih igara.



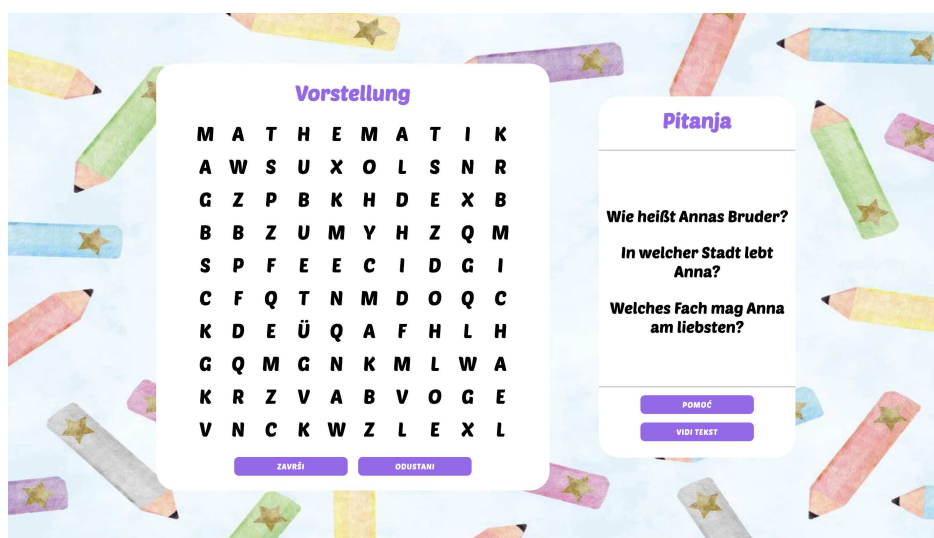
Slika 6.10. Zaslona za proslavu uspjeha

Učenik igre može igrati proizvoljan broj puta te je uvijek pohranjen samo njegov posljednji rezultat. Ukoliko učenik ne želi predati rezultate igre, pritiskom tipke "Odustani" igra završava te ga se preusmjerava na njegovu stranicu bez predaje rezultata.

Ukoliko je odabrana igra čitanja s razumijevanjem, prvo se prikazuje tekst o kojem su postavljena pitanja (prikazan slikom 6.11). Kada završi s čitanjem teksta, pritiskom na tipku "Dalje" započinje igra osmosmjerkice.



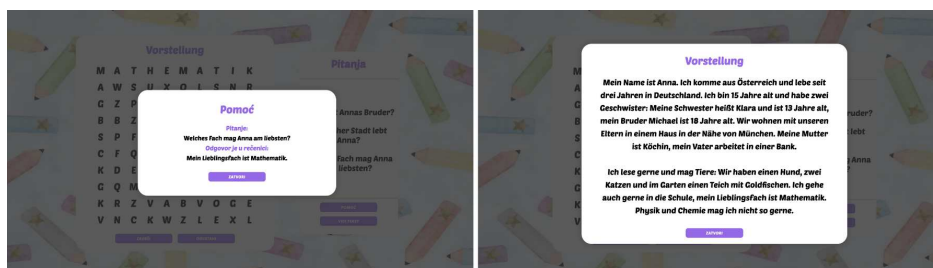
Slika 6.11. Prikaz teksta prije igre



Slika 6.12. Traženje odgovora na pitanje u osmosmjerkici

Zaslon igre prikazan je slikom 6.12. U tom trenu kreće mjerenje vremena te se prikazuje mreža slova i popis pitanja. Učenik u svakom trenu može ponovo pristupiti tekstu pritiskom tipke "Vidi tekst". Princip traženja riječi u osmosmjerkici je isti kao i u prošloj igri

te se pitanja na koja je pronađen odgovor prekrize. Pritiskom na tipku "Pomoć", otvara se modal u kojem se pojavljuje pitanje na koje će biti ponuđena pomoć te rečenica ili dio rečenice iz teksta u kojem se nalazi odgovor na pitanje. Ponovni pregled teksta i pomoć su prikazani slikom 6.13. Učenik i u ovom tipu igre ima mogućnost predaje rezultata ili odustajanja od igre.



Slika 6.13. Pregled teksta i pomoć

6.3. Pregled rezultata

Nastavnicima se također nudi mogućnost pregleda rezultata igara po razredima. Nakon odabira opcije "Rezultati" na stranici nastavnika, potrebno je u otvorenoj formi iz padajućeg izbornika odabrati školu i razred za koju rezultate žele vidjeti te zatim vrstu igre: klasičnu osmosmjerku ili igru čitanja s razumijevanjem. Kada odaberu potrebne podatke iz padajućih izbornika, moći će odabrati igru za koju žele pregledati rezultate. Za svaku su igru ponuđene dvije opcije: rezultati i najčešće greške.

Ukoliko je odabran pregled rezultata, nastavnik će moći za svakog učenika izabranog razreda vidjeti potrebno vrijeme rješavanja te broj pronađenih riječi, odnosno odgovorenih pitanja, ovisno o vrsti igre za koju se gledaju rezultati. Pregled rezultata jednog razreda za igru klasične osmosmjerke prikazan je slikom 6.14. Nastavnik vidi samo posljednji predani pokušaj rješavanja pojedinog učenika. Kada je nastavnik spreman završiti pregled, pritiskom tipke "Završi pregled", vratit će se na svoju stranicu.

Učenik	Vrijeme	Pronađene riječi
Mirta Hrnčić	42.415s	5/6
Gordana Milić	22.949s	4/6
Ivica Šipek	24.802s	4/6
Iskra Hrnčić	43.374s	4/6
Iris Matušin	49.311s	4/6
Filip Maručić	59.814s	3/6

ZAVRŠI PREGLED

Slika 6.14. Pregled rezultata osmosmjerke

1.a : Vorstellung

Najčešće pogreške

1. **Wie heißt Annas Bruder?**
2. **In welcher Stadt lebt Anna?**
3. **Welches Fach mag Anna am liebsten?**

ZAVRŠI PREGLED

Slika 6.15. Najrjeđe odgovorena pitanja

Ukoliko je odabran pregled statistike, biti će prikazane tri najrjeđe pronađene riječi u toj igri za izabran razred, odnosno najrjeđe odgovorena pitanja. Ta je opcija nastavnicima od posebne važnosti jer im ukazuje na dijelove gradiva koje učenici najteže usvajaju te bi im trebalo posvetiti dodatnu pažnju. Prikaz najčešće neodgovorenih pitanja jednog razreda prikazan je slikom 6.15.

7. Daljnji razvoj

U budućnosti, moguće je napraviti brojne izmjene i proširenja aplikacije. Razvijena aplikacija je za sada dostupna samo u web obliku. Nastavnički je dio aplikacije prilagođen korištenju na računalu, a učenički tabletima i računalima. Učenički dio aplikacije bilo bi poželjno učiniti primjerice Android aplikacijom koju mogu imati na svojim tabletima. Također, aplikacija ovakvog tipa se konstantno može proširivati novim igrama koje će poticati učenje i vježbu kod djece. Osim dodavanja novih vrsta igara, aplikaciji je moguće dodati elemente igrifikacije kako bi se motiviralo mlađe korisnike. Stvaranjem korisničkog računa izabrao bi se avatar, a rješavanjem zadataka dodjeljivale bi se nagrade. One mogu biti u obliku novčića kojim učenici mogu odabirati dodatke za svoj lik ukoliko su ih skupili dovoljno ili u obliku samih dodataka. Popravljanjem svojih rezultata, odnosno vježbom, te rješavanjem novih zadataka, odnosno učenjem, sakupljalo bi se sve više nagrada. Uz to, moguće je rezultate učenika učiniti dostupnima i drugim učenicima kako bi ih se kroz međusobno natjecanje motiviralo.

Provođenjem detaljnog i dobro osmišljenog testiranja aplikacije s nastavnicima i učenicima, dobile bi se povratne informacije o stvarnom korištenju koje su vrlo vrijedne u uvođenju izmjena u aplikaciji. Bazirano na tim informacijama, bilo bi moguće napraviti promjene koje su predložili korisnici kojima je aplikacija i namijenjena.

8. Zaključak

U sklopu završnog rada proučene su postojeće aplikacije za vježbu stranih jezika, posebice vokabulara te je razvijena i opisana aplikacija za vježbu vokabulara bazirana na igri osmosmjerke. Razvijena aplikacija sastoji se od sučelja za administratore te igre za učenike. Kao sastavni dio aplikacije, implementiran je algoritam raspodjele riječi unutar osmosmjerke dimenzija 10x10 te je osmišljen i napravljen pripadni model baze podataka za skladištenje potrebnih podataka. Napravljena je i verzija igre koja se bazira na traženju odgovora na pitanja o prethodno pročitanoj tekstu te je u sklopu te igre implementirano spajanje na generativni alat umjetne inteligencije ChatGPT.

Aplikacija za učenike isprobana je s učenicima prvog razreda osnovne škole koji uče engleski jezik, dok nastavničko sučelje nije korišteno u praksi od strane nastavnika. Na kraju godine zaigrali su dvije igre klasične osmosmjerke s riječima iz tekstova koje su obradili kroz godinu. Reakcije su bile pozitivne te su učenici bili vrlo motivirani vježbati na taj način. Kako bi se bolje procijenila sama korist i uloga aplikacije u nastavi stranih jezika, potrebno je napraviti opsežnije testiranje.

Bazirano na rezultatima takvog testiranja te povratne informacije učenika i nastavnika, aplikaciju bi bilo moguće proširiti dodatkom novih igara i mogućnosti unutar aplikacije poput elemenata igrifikacije i natjecanja. Također, aplikacija za učenike bi bila prikladnija u obliku aplikacije za tablete i računala, nego u postojećem obliku web-aplikacije.

Literatura

- [1] Mohammad Arief Wahyudi i Andini Kusumahwardani. “The Effectiveness of Word Search Puzzle Game as Media For Teaching Vocabulary”. *ELP (Journal of English Language Pedagogy)* 9.1 (siječanj 2024.), str. 43–52. DOI: 10.36665/elp.v9i1.817. URL: <http://ojs.fkipummy.ac.id/index.php/elp/article/view/817>.
- [2] Zhonggen Yu, Mingle Gao i Lifei Wang. “The Effect of Educational Games on Learning Outcomes, Student Motivation, Engagement and Satisfaction”. *Journal of Educational Computing Research* 59.3 (2021.), str. 522–546. DOI: 10.1177/0735633120969214. URL: <https://doi.org/10.1177/0735633120969214>.
- [3] The New York Times. *About New York Times Games*. <https://www.nytimes.com/crosswords>. Pristupljeno: 6.10.2024.
- [4] Studycat. *About*. <https://studycat.com/about/>. Pristupljeno: 6.10.2024.
- [5] John Terra. *What is Client-Server Architecture? Everything You Should Know*. <https://www.simplilearn.com/what-is-client-server-architecture-article>. Pristupljeno: 6.10.2024.
- [6] MDN Web Docs. *An overview of HTTP*. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>. Pristupljeno: 6.10.2024.
- [7] W3schools. *HTML Introduction*. https://www.w3schools.com/html/html_intro.asp. Pristupljeno: 6.10.2024.
- [8] GeeksforGeeks. *CSS Introduction*. <https://www.geeksforgeeks.org/css-introduction/>. Pristupljeno: 6.10.2024.
- [9] W3schools. *JavaScript Introduction*. https://www.w3schools.com/js/js_intro.asp. Pristupljeno: 6.10.2024.
- [10] Ushna Ijaz. *A brief introduction to Axios*. <https://rapidapi.com/guides/what-is-axios>. Pristupljeno: 6.10.2024.

- [11] W3 Schools. *Introduction to SQL*. https://www.w3schools.com/sql/sql_intro.asp. Pristupljeno: 6.10.2024.
- [12] GeeksforGeeks. *React Introduction*. <https://www.geeksforgeeks.org/reactjs-introduction/>. Pristupljeno: 6.10.2024.
- [13] freeCodeCamp. *What is the Virtual DOM in React?* <https://www.freecodecamp.org/news/what-is-the-virtual-dom-in-react/>. Pristupljeno: 6.10.2024.
- [14] Node.js. *About*. <https://nodejs.org/en/about>. Pristupljeno: 6.10.2024.
- [15] Codecademy. *What is Express.js?* <https://www.codecademy.com/article/what-is-express-js>. Pristupljeno: 6.10.2024.
- [16] Coursera. *What is OpenAI? Everything You Need to Know*. <https://www.coursera.org/articles/what-is-openai>. Pristupljeno: 6.10.2024.
- [17] OpenAI Platform. *Models*. <https://platform.openai.com/docs/models>. Pristupljeno: 6.10.2024.
- [18] PostgreSQL. *About*. <https://www.postgresql.org/about/>. Pristupljeno: 6.10.2024.
- [19] Aviator. *ACID transactions and implementation in a PostgreSQL Database*. <https://www.aviator.co/blog/acid-transactions-postgresql-database/>. Pristupljeno: 6.10.2024.

Sažetak

Aplikacija za vježbu vokabulara pomoću osmosmjerke

U sklopu ovog završnog rada razvijena je igra osmosmjerke za vježbu vokabulara stranih jezika te pripadajuće nastavničko sučelje. Razmotrena je upotreba igara u procesu učenja te su opisani postojeći primjeri aplikacija razvijenih u tu svrhu. Opisane su korištene tehnologije u razvoju aplikacije te osmišljen model baze podataka. Prikazane su upute za korištenje aplikacije za nastavnike i učenike. Cilj razvijene aplikacije je nadopuniti tradicionalne metode učenja i vježbe vokabulara igrom koja je primamljiva i motivirajuća učenicima te jednostavna za zadavanje i praćenje napretka nastavnicima. Uz igru klasične osmosmjerke, razvijena je i igra čitanja s razumijevanjem s ciljem poboljšanja sposobnosti učenika da prepoznaju i shvate značenje riječi u kontekstu.

Ključne riječi: React, Node.js, Express, OpenAI API, igra, osmosmjerka, učenik, nastavnik, čitanje s razumijevanjem

Summary

Vocabulary Practice Application Using Crossword Puzzle

As part of this thesis, a word search game for foreign language vocabulary practice and a corresponding teacher interface were developed. The use of games in the learning process was considered, and existing examples of applications developed for this purpose were described. The technologies used in the development of the application were outlined, and designed database model was described. Instructions for using the application for both teachers and students were provided. The aim of the developed application is to complement traditional methods of learning and practicing vocabulary with a game that is engaging and motivating for students, as well as easy for teachers to assign and track progress. In addition to the classic word search game, a reading comprehension game was also developed with the goal of improving students' ability to recognize and understand the meaning of words in context.

Keywords: React, Node.js, Express, OpenAI API, game, word search, student, teacher, reading comprehension