

Aplikacija za ocjenjivanje čitljivosti teksta s obzirom na boju

Ostović, Josip

Undergraduate thesis / Završni rad

2023

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:937113>

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

Download date / Datum preuzimanja: **2024-12-28**

Repository / Repozitorij:

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



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

Sveučilišni studij

**APLIKACIJA ZA OCJENJIVANJE ČITLJIVOSTI
TEKSTA S OBZIROM NA BOJU POZADINE**

Završni rad

Josip Ostović

Osijek, 2023.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

Osijek, 15.09.2023.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Josip Ostović
Studij, smjer:	Programsko inženjerstvo
Mat. br. Pristupnika, godina upisa:	R 4411, 14.12.2020.
OIB Pristupnika:	61456795851
Mentor:	doc. dr. sc. Krešimir Romić
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Aplikacija za ocjenjivanje čitljivosti teksta s obzirom na boju
Znanstvena grana rada:	Obradba informacija (zn. polje računarstvo)
Zadatak završnog rad:	Izraditi aplikaciju pomoću koje će se moći dodijeliti ocjena za čitljivost kratkog računala prikaznog teksta s obzirom na boju samog teksta i boju pozadine. Za potrebe testiranja generirati više različitih kombinacija boje teksta i pozadine te provesti ocjenjivanje čitljivosti s više ocjenjivača. Omogućiti prikladan zapis rezultata ocjenjivanja te omogućiti analizu istih. Analizirati dobivene eksperimentalne podatke i predložiti primjenu ovakvih podataka o čitljivosti. Tema rezervirana za: Josip Ostović
Prijedlog ocjene završnog rada:	Dobar (3)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 1 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 1 bod/boda Jasnoća pismenog izražavanja: 1 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene od strane mentora:	15.09.2023.
Datum potvrde ocjene od strane Odbora:	24.09.2023.
Potvrda mentora o predaji konačne verzije rada:	<i>Mentor elektronički potpisao predaju konačne verzije.</i>
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 25.09.2023.

Ime i prezime studenta:	Josip Ostović
Studij:	Programsko inženjerstvo
Mat. br. studenta, godina upisa:	R 4411, 14.12.2020.
Turnitin podudaranje [%]:	7

Ovom izjavom izjavljujem da je rad pod nazivom: **Aplikacija za ocjenjivanje čitljivosti teksta s obzirom na boju**

izrađen pod vodstvom mentora doc. dr. sc. Krešimir Romić

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD.....	1
2. PREGLED PODRUČJA TEME.....	2
3. ČITLJIVOST RAČUNALNO PRIKAZANOG TEKSTA.....	4
3.1. Što sve utječe na čitljivost teksta.....	4
3.2. Utjecaj boje na čitljivost teksta.....	4
4. OKRUŽENJE ZA OCJENJIVANJE ČITLJIVOSTI.....	6
4.1. Opis sustava.....	6
4.2. Korištene tehnologije.....	6
4.2.1. ReactJS.....	6
4.2.2. Firebase.....	7
4.3. Implementacija rješenja	7
5. EKSPERIMENT	14
5.1. Ulazni podaci.....	14
5.2. Ocjenjivanje	14
5.3. Rezultati.....	15
6. ZAKLJUČAK.....	18
ŽIVOTOPIS.....	22

1. UVOD

U današnjem digitalnom dobu, gdje se ogroman dio informacija prenosi putem tekstualnih sadržaja na različitim medijima, pitanje čitljivosti teksta postalo je ključno za efikasnu komunikaciju. Čitljivost, odnosno sposobnost čitatelja da lako i brzo razumije tekst, ovisi o raznim faktorima, a jedan od njih je i boja pozadine na kojoj se tekst prikazuje. Ova tema postaje sve značajnijom s obzirom na raznolike digitalne platforme i uređaje koji omogućavaju personalizaciju izgleda teksta. Ovaj završni rad istražuje problem čitljivosti teksta u kontekstu boje pozadine, s posebnim fokusom na razvoj aplikacije koja će pomoći u ocjenjivanju čitljivosti teksta na različitim kombinacijama boje pozadine i teksta. Razumijevanje kako boja pozadine utječe na percepciju teksta može biti od ključne važnosti za dizajnere web stranica, mobilnih aplikacija, te svih koji žele osigurati da njihove poruke budu dostupne i razumljive svim korisnicima. Istražit će se teorijski okvir čitljivosti teksta, analizirati faktore koji utječu na čitljivost u kontekstu boje pozadine, te razviti aplikaciju koja će omogućiti korisnicima da testiraju i optimiziraju čitljivost svojih tekstova. Očekujemo da će ova aplikacija pružiti vrijedne smjernice za bolje razumijevanje kako odabrana boja pozadine može poboljšati ili smanjiti čitljivost teksta te kako ju prilagoditi za optimalno iskustvo čitatelja. Kroz daljnje istraživanje i praktične primjene rezultata ovog rada, nadamo se unaprijediti kvalitetu digitalnih sadržaja i doprinijeti boljoj komunikaciji na globalnoj razini.

2. PREGLED PODRUČJA TEME

Personalizacija teksta postala je sveprisutna u digitalnom svijetu, omogućujući korisnicima prilagodbu boje pozadine i teksta prema vlastitim preferencijama. Različite skupine korisnika, uključujući osobe s određenim vidnim ograničenjima, starije osobe s posebnim potrebama, također mogu reagirati različito na kombinacije boja pozadine i teksta. Vrlo sličan problem koji je usko vezan s ovim završnim radom je primjena boje u dizajnu i oblikovanju vizuala, gdje je paleta boja dobivena ubacivanjem fotografije u alat koji generira paletu i iz nje je dobivena odgovarajuća paleta boja [1]. Uz to, napravljeni su i uzorci odgovarajućeg teksta koji će se kombinirati s prethodno generiranom paletom boja gdje će se ispitivati različite dobne skupine koje će dati svoje mišljenje o pojedinoj kombinaciji teksta i boje. Nadalje, u drugim slučajevima kod određene skupine ljudi, zadatak je bio da na zaslonu pročitaju određeni tekst u boji s obzirom na boju pozadine te se prati akomodacijski odgovor i dinamika zjenice tijekom čitanja odlomaka s različitim kombinacijama boja teksta i pozadine [2]. Rezultati su pokazali da kombinacija boja teksta i pozadine modulira akomodativnu i zjeničnu dinamiku tijekom čitanja. Kao i na zaslonu, provjera čitljivosti boje teksta i pozadine se može izvršiti i na određenoj udaljenosti. Sukladno tome, ispituju se različiti fontovi za tekst na različitim udaljenostima [3]. U istraživanju u kojoj se ispitivalo virtualno okruženje, koristila su se četiri različita fonta u deset veličina na tri različite udaljenosti: pet, deset i petnaest metara. Na temelju rezultata, pokazalo se kako većina sudionika preferira crni tekst na bijeloj pozadini, crni tekst na sivoj pozadini te bijeli tekst na narančastoj pozadini. Također, utjecaj boje pozadine s obzirom na boju teksta se pojavljuje i u drugim slučajevima u današnjem životu. Jedno od njih su aplikacije za čitanje e-knjiga u kojoj aplikacije dopuštaju korisnicima prilagodbu boje pozadine i teksta. Ovo je osobito korisno jer čitatelji mogu odabrati kombinaciju koja im najbolje odgovara, bilo da preferiraju svijetli tekst na tamnoj pozadini ili obrnuto [4]. Nadalje, kod grafičkog dizajna i marketinga, odabir boje pozadine i teksta igra ključnu ulogu u privlačenju pažnje i stvaranju određenih emocionalnih odgovora. Boje se često koriste kako bi se izazvala određena atmosfera ili stvorila prepoznatljiva marka [5]. Utjecaj boje pozadine i teksta javlja se i kod pametnih telefona gdje je sprovedeno istraživanje u kojem su deseci studenata ispitivali i analizirali različite tipove dizajna virtualnih paketa pametnih telefona [6]. Paketi su korisnicima elektronski predstavljeni na ekranu računala. Glavni rezultati su pokazali da su muškarci više voljeli sive varijante, a djevojke ružičaste te se na temelju analize pokazalo da je boja pozadine bio glavni čimbenik koji je utjecao na preferencije muškaraca, dok kod žena to nije bilo važno pitanje. Osim grafičkog dizajna, čitanju e-knjiga i ostalih navedenih aplikacija, utjecaj boje pozadine na tekst je osobito važno i u tehničkim znanostima. Možemo napraviti

aplikaciju poput online tečajeva za obrtničke vještine. Jedan od dobrih primjera vezanog za navedenu aplikaciju je što aplikacija može nuditi svijetlu pozadinu (bijelu ili svijetlo sivu) s crnim ili tamnoplavim tekstom . Ova kombinacija pruža visoki kontrast, što olakšava čitanje i pomaže učenicima da jasno vide tekst na ekranu te također stvara profesionalan i ozbiljan dojam. Loš primjer bi bila kombinacija svijetlo žute pozadine s narančastim tekstom te samim tim može ometati čitljivost i iskustvo učenja. Ovaj primjer aplikacije naglašava važnost pažljivog odabira boja pozadine i teksta u okviru edukacijskih aplikacija kako bi se osiguralo da učenici imaju ugodno iskustvo učenja. Mogu se također usredotočiti na materijale i maksimalno iskoristiti svoje online obrazovanje. Pravilna upotreba boja može doprinijeti kvaliteti obrazovnog iskustva i potaknuti uspješno učenje.

3. ČITLJIVOST RAČUNALNO PRIKAZANOG TEKSTA

Čitljivost se odnosi na svojstva izoliranih znakova poput slova, brojeva ili drugih simboličkih znakova, kao i na svojstva većih skupina ili cjelina poput riječi i rečenica. U novije doba govori se i o čitljivosti pisanoga materijala posredovanoga drugim medijima, poput televizijskog ili računalnog zaslona.

3.1. Što sve utječe na čitljivost teksta

Veličina slova, tip slova (font), razmak između slova ili redova, veličina rubova, kontrast između slova i podloge na kojoj se nalaze samo su jedni od najčešćih svojstava koji utječu na čitljivost računalnog teksta. Velika slova vidljivija su od malih na većoj udaljenosti, dok riječi kao cjeline vidljivije su kada su pisane malim slovima. Tiskanje kurzivom (italikom) uzrokuje značajno sporije čitanje od uobičajenoga tiska. Cijeli tekst tiskan velikim slovima smanjuje brzinu čitanja dok uvlačenje prvog retka odlomka povećava čitljivost. Što se tiče boje i kontrasta, što je veći kontrast između boje slova i boje podloge, bolja je čitljivost. Tamna slova na svijetloj podlozi povećavaju čitljivost (tri četvrtine čitatelja preferira crna slova na bijeloj podlozi), dok upotreba različitih vrsta slova u istom tekstu usporava čitanje. Za uspješnu tipografiju (čitljivost), odabir pravog fonta je ključan faktor u postizanju uspješnog dizajna. Osim fonta, kao što je ranije navedeno, razmak (*engl. Leading*) je također jedan od ključnih faktora u postizanju jasne i dobre čitljivosti. Ako nismo sigurni koliki razmak trebamo koristiti, većina programa ima zadane postavke koje su napravljene da daju optimalni razmak, a profesionalni dizajneri će razmak prilagoditi viziji koju oni imaju s time da uvijek treba paziti na čitkost samog teksta. Važno je u tom slučaju pronaći ravnotežu koja djeluje prirodno. Treba još spomenuti i pojmove *kerning* i *tracking*. *Kerning* je koncept koji dodaje i oduzima razmake među slovima, gdje razmak između pojedinih slova može biti dosta različit, dok je *tracking* sličan koncept *kerning-u*, ali on dodaje i oduzima razmake između slova u cijeloj riječi.

3.2. Utjecaj boje na čitljivost teksta

Boja ima ogroman utjecaj na čitljivost teksta. Problem nastaje ukoliko trebamo pronaći odgovarajuću kombinaciju boje pozadine i teksta te iz tog razloga korisnici pokušavaju otkriti i pronaći optimalnu boju pozadine s obzirom na odabranu boju teksta. Danas se gotovo u svim okruženjima koristi bijela pozadina u kombinaciji s crnim tekstom jer je crna boja najupečatljivija ljudskom oku i nije teško zapamtiti takav tekst. Lako je kada se uzme osnovna boja pozadine i

teksta, ali problem nastaje ukoliko se treba pronaći druga boja te vidjeti kako ona utječe na odgovarajući tekst. Razni su faktori koji utječu na samu boju i njezinu prilagodbu tekstu. Jedan od njih je kontrast gdje mrežne stranice koje imaju nizak kontrast boje pozadine s tekстом mogu biti teške za čitanje i upotrebu. Stoga, crno-bijela kombinacija predstavlja i veći kontrast te je lakša za čitanje i upotrebu ukoliko imamo nekakav kraći tekst, ali onda se dolazi do problema kad se pojavljuje duži tekst. U tom slučaju, crno-bijela kombinacija gdje crna boja predstavlja pozadinu a bijela boja tekst iziskuje dodatan napor i naprezanje očiju. Iz tog razloga najbolje rješenje bi bilo pronaći nekakvu drugu kombinaciju boja pozadine i vidjeti kako ona utječe na samog korisnika i njegovo oko. Na slici 3.1. vide se razne kombinacije boje pozadine i teksta. Stupci predstavljaju boju pozadine dok redci predstavljaju boju teksta. Iz tablice se jasno vidi da kombinacija žarkih boja pozadine poput crvene, narančaste ili čak i roze s također žarkim bojama teksta uzrokuju dosta slabiju i ljudskom oku lošiju čitljivost u odnosu na druge i prikladnije boje. Stoga bi prikladnije bilo pronaći bolji kontrast boje pozadine i teksta kako bi ljudsko oko što je lakše i bolje moguće uočilo odgovarajući tekst.

		Background								
		Red	Orange	Yellow	Green	Blue	Violet	Black	White	Gray
Foreground	Red		Poor	Good	Poor	Poor	Poor	Good	Good	Poor
	Orange	Poor		Poor	Poor	Poor	Poor	Good	Poor	Poor
	Yellow	Good	Good		Poor	Good	Poor	Good	Poor	Good
	Green	Poor	Poor	Poor		Good	Poor	Good	Poor	Good
	Blue	Poor	Poor	Good	Good		Poor	Poor	Good	Poor
	Violet	Poor	Poor	Good	Poor	Poor		Good	Good	Poor
	Black	Poor	Good	Good	Good	Poor	Good		Good	Poor
	White	Good	Good	Good	Poor	Good	Good	Good		Good
	Gray	Poor	Poor	Good	Good	Poor	Poor	Poor	Good	

Sl. 3.1. Primjeri boje pozadine i teksta

4. OKRUŽENJE ZA OCJENJIVANJE ČITLJIVOSTI

U ovom poglavlju govorit će se o arhitekturi sustava i o korištenim tehnologijama pri izradi aplikacije.

4.1. Opis sustava

Web aplikacija će korisnicima prikazati nasumično generirane kombinacije boje pozadine i teksta gdje će korisnici na temelju dobivenih slika ocjenjivati odgovarajuće kombinacije koje dobiju ocjenama od jedan do pet, gdje će jedinica predstavljati lošu kombinaciju a petica najbolju, odnosno najprikladniju. Ispod slike će biti gumbovi za odabir ocjene gdje će se klikom na ocjenu korisnicima prikazati druga slika te će ih ocjenjivati dok ne ocjene sve slike. Svaka ocjena za određenog korisnika se sprema u bazu podataka gdje će se kasnije na temelju svih ocjena formirati tablica u kojoj će biti prikazani redni brojevi slike i pripadajuća prosječna ocjena za svaku pojedinu sliku. Kombinacije boje teksta i pozadine bit će opisane u JSON datoteci te se za potrebe istraživanja tamo mogu mijenjati. Broj kombinacija iz te datoteke će odrediti i koliko slika će biti u tablici rezultata.

4.2. Korištene tehnologije

Tehnologije koje se koriste u ovom završnom radu su *ReactJS* koji će služiti za kreiranje web dijela i prikaz web aplikacije i *Firebase* koji će služiti za spremanje podataka, u ovom slučaju korisnika i njihovih ocjena u bazu te pomoću dobivenih podataka analizirati slike.

4.2.1. ReactJS

ReactJS je *JavaScript* biblioteka koja se koristi za izgradnju korisničkih sučelja na učinkovit način [7]. Korisnici ga danas koriste za izradu raznih *web* aplikacija te stvara jednostavne prikaze za svako stanje u korisničkoj aplikaciji. Osim toga, kad se korisnički podaci trebaju promijeniti, biblioteka ih nakon promjene ažurira i učita pravu komponentu. Dobra stvar je što svaka komponenta u *React-u* prikazuje pojedini dio *HTML* koda koji se može ponovno koristiti. Također je dostupno i ubacivati komponente unutar drugih komponenti te se tako omogućuje izgradnja složenih aplikacija od jednostavnijih dijelova. Svaka pojedina komponenta može pratiti svoje unutarnje stanje. Prednost je što pojedini dijelovi komponenti poboljšavaju čitljivost te prema tome pomažu u održavanju većih aplikacija. Mana je fleksibilnost i nedostatak konvencija te visok tempo razvoja. U ovom završnom radu će se isključivo koristiti za izradu i dizajn *web* aplikacije i sučelja za prikaz slika korisnicima i njihovo ocjenjivanje slike.

4.2.2. Firebase

Firebase je platforma za razvoj aplikacija koja pruža razne alate i usluge za brz i jednostavan razvoj mobilnih i *web* aplikacija [8]. Nudi usluge za autentifikaciju korisnika, omogućavajući registraciju, prijavu i upravljanje korisničkim računima. Kao što je spomenuto, ima razne alate za pohranu podataka. Tako je jedan od njih *Firebase Cloud Firestore* koji se koristiti u ovom završnom radu za učinkovito spremanje i dohvaćanje podataka iz baze. Za prijavu korisnika u aplikaciju kako bi se kasnije mogao spremi u bazu, koristit će se *Firebase Authentication*.

4.3. Implementacija rješenja

Počevši od početne stranice, korisnik ima zadatak da prije svega unese svoj nadimak i klikom na gumb, ime koje je korisnik upisao u polje za unos nadimka se sprema i korisnik prelazi na ocjenjivanje određenih slika. Prije svega, metoda *useState()* korištena je za definiranje dvije lokalne varijable [9]. Na slici 4.1 prikazan je unos dvije lokalne varijable. Prva varijabla predstavlja stanje koje će sadržavati trenutno uneseni nadimak, dok druga varijabla predstavlja

```
const [nickname, setNickname] = useState('');  
const [isNicknameEntered, setIsEnteredNickname] = useState(false);
```

S1. 4.1. Kod za unos varijable nadimka

stanje je li unesen nadimak. Sa slike se jasno vidi kako na početku nadimak nije unesen. Slika 4.2 prikazuje metodu *handleChangeNickname()* koja se poziva svaki put kad se promijeni uneseni tekst u polju za unos nadimka. Koristi varijablu *setNickname* za ažuriranje stanja nadimka koja je

```
const handleChangeNickname = (e) =>{  
  setNickname(e.target.value);  
  setIsEnteredNickname(e.target.value.trim() !== '');  
}
```

S1. 4.2. Metoda handleChangeNickname()

```

return <div>
  <input
    type="text"
    value={nickname}
    onChange={handleChangeNickname}
    placeholder="Enter your nickname"
    className="nickname-input"
  />
  <button onClick={handleNicknameSubmit}
    disabled={!isNicknameEntered}
    className="button-submit">Submit Nickname</button>
</div>

```

S1. 4.3. JSX kod

```

const handleNicknameSubmit = () =>{
  if(isNicknameEntered){
    navigate('/home', {state:{nickname:nickname}});
  }
}

```

S1. 4.4. Metoda handleNicknameSubmit()

predstavljena varijablom *nickname* s trenutnom vrijednošću unosa, a zatim se provjerava je li uneseni tekst prazan. Metoda *trim()* koristi se za uklanjanje nepotrebnih razmaka. Ako polje za unos nadimka nije prazno, postavlja varijablu *isNicknameEntered* na *true*. Kada korisnik klikne na gumb za spremanje nadimka, poziva se metoda *handleNicknameSubmit()* koja je prikazana na slici 4.4. U metodi prikazanoj na gornjoj slici prvo se provjerava da li je unesen nadimak. Ako je nadimak unesen, koristi metodu *navigate()* za preusmjeravanje korisnika na stranicu gdje će moći ocjenjivati ponuđene kombinacije boje pozadine i teksta i uz to se prosljeđuje nadimak kao dio stanja (u ovom slučaju putem *state* svojstva) [10]. Unutar *return()* bloka nalazi se JSX kod koji komponira korisničko sučelje. To uključuje: *input* element koji predstavlja element za unos nadimka, zatim element koji predstavlja gumb za spremanje nadimka koji poziva funkciju sa slike 4.4 i onemogućen je ako je unos nadimka *false*. Slika 4.3 prikazuje JSX kod unutar *return()* bloka i sve pripadne elemente. U osnovi, korisnik unosi nadimak u prostor za unos nadimka, zatim klikom na gumb se preusmjerava na iduću stranicu, odnosno na stranicu za ocjenjivanje slika samo ako je uneseni nadimak valjan. Unutar *App.js* komponente koja predstavlja glavnu *ReactJS*

komponentu nalaze se poveznice i rute između svih stranica u aplikaciji. Prije korištenja pojedinih poveznica do drugih stranica, potrebno je ubaciti određene module i komponente koje će se koristiti u aplikaciji. Osim stranica, potrebni su i alati koji će služiti za povezivanje između pojedinih stranica poput *Router*, *Routes* i slično. Unutar *return()* bloka koristi se JSX kod te *React Router* za definiranje ruta i poveznica na različite stranice. Unutar glavnog dijela aplikacije nalaze se tri glavne rute: *'/home'* ruta prikazuje stranicu za ocjenjivanje pojedinih slika te će se ona prikazati kada korisnik unese svoj nadimak. Ruta koja prikazuje početnu stranicu u kojoj se unosi nadimak i zatim prelazi na ocjenjivanje je *'/'*, dok sve ostale rute i stranice koje nisu definirane unutar glavnog dijela aplikacije, odnosno stranice koje ne postoje, definirane su unutar *'*'* rute. Da bi se pojedine slike prikazale na ekranu, potrebno ih je spremiti u posebnu datoteku. U ovom slučaju, u *pictures.json* dokumentu čuvamo samo opis slike tj. njenih boja, a onda će se slika prikazati u HTML-u pomoću tih svojstava. Prikaz jednog JSON objekta prikazan je na slici 4.5.

```
{
  "id": 1,
  "Rback": 100,
  "Gback": 0,
  "Bback": 0,
  "Rtext": 0,
  "Gtext": 100,
  "Btext": 0
},
```

S1. 4.5. Prikaz jednog JSON objekta

Na slici se vidi *id* koji predstavlja redni broj slike te *RGB* komponente za boju pozadine i boju teksta. Nadalje, da bi korisnik mogao odabrati ocjenu kojom želi ocijeniti pojedinu sliku, potrebno je inicijalizirati varijablu putem *useState()* metode koja će bilježiti korisnikove ocjene. Varijabla zadužena za praćenje trenutne ocjene je *selectedOption*, gdje je početno stanje ocjene postavljeno na *null*, odnosno nije unesena trenutna korisnikova ocjena. Kada korisnik odabere jednu od ponuđenih ocjena, njegov rezultat se postavlja pomoću *setSelectedOption* metode. Indeks trenutne slike spremljen je u *currentIndex* varijablu koja je zadužena za praćenje slika koje će se prikazati na korisničkom ekranu. Referenca na kolekciju *colors* koja se nalazi unutar *Firebase* baze podataka unutar koje će se spremati potrebni podaci prilikom ocjenjivanja pojedinih slika predstavljena je varijablom *ratingRef* koja kao ulazne podatke prima komponentu koja predstavlja

bazu podataka pod nazivom *db* te drugi parametar koji predstavlja konkretnu kolekciju koja se nalazi unutar *Firebase* baze podataka. U ovom slučaju je to kolekcija *colors*. Važno je za napomenuti kako je prije korištenja baze podataka potrebno uvesti komponentu koja predstavlja samu bazu podataka, a to se ostvaruje putem ključne riječi *import* te ju nakon toga možemo koristiti u našoj aplikaciji. Kako korisnik ocjenjuje svaku pojedinu sliku, tako je i potrebno imati varijablu koja će biti zadužena za prosječne ocjene svih slika u aplikaciji te je *averageRatings* varijabla zadužena za taj zadatak. Pomoću *useLocation()* metode korisnik dobiva podatke iz trenutne lokacije i izvlači određeni nadimak iz tih podataka koji je predan preko stanja prilikom preusmjerenja s trenutne stranice [11]. Kada korisnik ocijeni svaku pojedinu sliku koja se nalazi u JSON dokumentu, potrebno se vratiti na početnu stranicu za ponovni unos nadimka, a to je ostvareno putem *navigate()* metode iz *React Router*a. Slika 4.6 prikazuje prethodno navedene varijable koje su nužne unutar aplikacije. Nadalje, metoda *handleClickNext()* se poziva kada korisnik klikne na gumb „*Slijedeći element*“. Prvo se provjerava da li je korisnik odabrao željenu ocjenu te se u toj situaciji provjerava varijabla *selectedOption*. Ako korisnik nije odabrao ocjenu, ispisuje se poruka u konzoli i vraća se iz metode. U suprotnom, povećava varijablu *currentIndex* koja je zadužena za praćenje trenutne slike za jedan, sprema odabranu ocjenu u bazu podataka zajedno s podacima o slici i nadimkom korisnika te izračunava prosječne ocjene slika. Važno je

za

```
const [selectedOption, setSelectedOption] = useState(null);
const [currentIndex, setCurrentIndex] = useState(0);
{/* const [textColor, setTextColor] = useState("blue"); */}
const ratingRef = collection(db, "colors");
const navigate = useNavigate();
const [averageRatings, setAverageRatings] = useState({});
const location = useLocation();
const { nickname } = location.state || {};
```

S1. 4.6. Prikaz varijabli unutar Home.js

naglasiti da se u bazu podataka mogu spremati bilo koji podaci koji korisnika zanimaju i koji su potrebni. Spremanje u bazu podataka se vrši pomoću metode *addDoc()*. U ovom slučaju, u bazu se spremaju *RGB* komponente pozadine koje su naznačene s *Rback*, *Gback* i *Bback*, zatim *RGB*

```

await addDoc(ratingRef, {
  Rback: ` ${PICTURES[currentIndex].Rback} `,
  Gback: ` ${PICTURES[currentIndex].Gback} `,
  Bback: ` ${PICTURES[currentIndex].Bback} `,
  Rtext: ` ${PICTURES[currentIndex].Rtext} `,
  Gtext: ` ${PICTURES[currentIndex].Gtext} `,
  Btext: ` ${PICTURES[currentIndex].Btext} `,
  rating: selectedOption,
  id: PICTURES[currentIndex].id,
  nickname: nickname
});

```

S1. 4.7. Prikaz spremanja podataka u *Firestore*

komponente teksta koje su naznačene s *Rtext*, *Gtext* i *Btext* te ocjena koju je korisnik odabrao zajedno s pripadajućim rednim brojem slike i na kraju korisnikov nadimak. Svaka kombinacija boje teksta i pozadine prikazana je pomoću komponente *PICTURES* koja predstavlja komponentu uvedenu iz *pictures.json* dokumenta koja sadrži polje objekata svih slika. U ovoj situaciji, pristupa se preko indeksa trenutne slike preko kojeg se može pristupiti svakom pojedinom podatku unutar *PICTURES* komponente. Na slici 4.7. prikazan je kod zadužen za spremanje podataka vezanih uz pojedinu sliku u *Firestore* bazu podataka. Nakon što se pripadajući podaci pohrane u *Firestore*, slijedeći korak je izračunati prosječne ocjene za svaku sliku. To se radi korak po korak kako bi se prikupili podaci o ocjenama korisnika i izračunale prosječne ocjene. Sukladno tome, na slici 4.8. prikazan je dio koda koji je zadužen za izračun prosjeka ocjena pojedine slike. Prvo, pregledavaju se svi podaci u *Firestore* bazi podataka. Svaki podatak predstavlja ocjenu koju je korisnik dao za određenu sliku. Za svaku ocjenu, bilježimo identifikator slike i ocjenu koju je korisnik dao. Ukoliko nekoj ocjeni nedostaju bitne informacije, preskače se, a ukoliko ima sve potrebne podatke, ocjene se zbrajaju i broje se za svaku pojedinu sliku. Zatim se izračunava prosječna ocjena svake pojedine slike i nakon što se sve ocjene obrade, dobit će se prosječna ocjena za svaku pojedinu sliku. Prikaz JSX koda unutar *return()* bloka zadužen za prikaz pojedine dinamički generirane slike nalazi se na slici 4.9. Pozadinska boja slike dinamički se postavlja na temelju *RGB* komponenti koje su pohranjene u objektu *PICTURES*. Slika je stilizirana tako da je centrirana na ekranu. Iznad svake slike, nalazi se tekst koji je također stiliziran i postavlja se u centar slike. Boja teksta također se određuje putem *RGB* komponenti koje su pohranjene u objektu *PICTURES*. Bitna stavka je da se slike dinamički mijenjaju kako korisnik napreduje kroz ocjenjivanje. To znači da će se svaki put kad korisnik odabere ocjenu i klikne gumb „*Slijedeći element*“, prikazati nova slika za ocjenjivanje. Prikaz jedne dinamički generirane slike nalazi se na slici 4.11.


```

querySnapshot.forEach((doc) => {
  const data = doc.data();
  if (data.id && data.rating) {
    const imageId = data.id;
    const rating = parseInt(data.rating);

    if (!ratingsByImage[imageId]) {
      ratingsByImage[imageId] = {
        totalRating: rating,
        count: 1,
      };
    } else {
      ratingsByImage[imageId].totalRating += rating;
      ratingsByImage[imageId].count++;
    }
  }
});

const imageIds = Object.keys(ratingsByImage);
const newAverageRatings = {};

imageIds.forEach((imageId) => {
  const totalRating = ratingsByImage[imageId].totalRating;
  const count = ratingsByImage[imageId].count;
  const averageRating = totalRating / count;
  newAverageRatings[imageId] = averageRating.toFixed(2);
});

```

S1. 4.8. Izračun prosjeka ocjena pojedine slike

```

return(
  <div className='container'>
    {PICTURES[currentIndex] && (
      <div>
        <div className='picture-container'
          style={{background: `rgb(${PICTURES[currentIndex].Rback},
            ${PICTURES[currentIndex].Gback},
            ${PICTURES[currentIndex].Bback})`,
          height: "200px", width: "800px"}}>
          <div className='text-container'
            style={{color: `rgb(${PICTURES[currentIndex].Rtext},
              ${PICTURES[currentIndex].Gtext},
              ${PICTURES[currentIndex].Btext})`,
            textAlign: 'center'}}>
            Ovo je neki tekst
          </div>
        </div>
      </div>
    )}
  </div>

```

S1. 4.9. JSX kod za prikaz slike na ekranu

Na slici 4.10 prikazan je JSX kod unutar *return()* bloka koji prikazuje gumbove za ocjenjivanje slike. Na temelju gore navedene slike, prvo se ispituje uvjet da li je trenutni indeks slike manji od

niza *PICTURES* i tek ako je uvjet zadovoljen, prikazuju se gumbovi za ocjenjivanje. Svaki gumb unutar sebe predstavlja jednu od opcija. Prije nego što korisnik prijeđe na slijedeću sliku, provjerava se je li odabrana neka od ponuđenih ocjena. Ukoliko korisnik pokuša prijeći na slijedeću sliku bez odabrane ocjene, prikazuje se poruka upozorenja.

```
{currentIndex < PICTURES.length ? (<div className='checkbox'>
  <>
    <input type='checkbox' checked={selectedOption === "1"}
      onChange={() => setSelectedOption("1")} />
    <label className='label-checkbox'>1</label>
    <input type='checkbox' checked={selectedOption === "2"}
      onChange={() => setSelectedOption("2")} />
    <label className='label-checkbox'>2</label>
    <input type='checkbox' checked={selectedOption === "3"}
      onChange={() => setSelectedOption("3")} />
    <label className='label-checkbox'>3</label>
    <input type='checkbox' checked={selectedOption === "4"}
      onChange={() => setSelectedOption("4")} />
    <label className='label-checkbox'>4</label>
    <input type='checkbox' checked={selectedOption === "5"}
      onChange={() => setSelectedOption("5")} />
    <label className='label-checkbox'>5</label>
    <button onClick={handleClickNext} className='button-slijedeci'>
      Slijedeci element
    </button>
  </>
</div> : null}
```

Sl. 4.10. JSX kod za prikaz ocjena na ekranu



1 2 3 4 5 Slijedeci element

Sl. 4.11. Prikaz slike na ekranu

5. EKSPERIMENT

U ovom dijelu detaljnije će se objasniti ulazni podaci aplikacije, zatim će proći kroz ocjenjivanje pojedine slike te na kraju prikaz prosječnih ocjena svake slike.

5.1. Ulazni podaci

Ulazni podaci zapisani su unutar *pictures.json* dokumenta koji predstavlja sve potrebne podatke vezane uz pojedinu sliku. Dokument se može razložiti na tri bitne stavke koje korisniku služe kao ulazni podaci. Prva od njih je *id* koja predstavlja identifikator slike. Može se koristiti kako bi se svaka slika razlikovala od drugih. U ovom završnom radu, omogućit će se jednostavno praćenje i identifikacija svake slike. Druga stavka predstavlja *RGB* komponente boje pozadine, a one su kako je navedeno ranije: *Rback*, *Gback* i *Bback*. Svaka komponenta ima vrijednost u rasponu od 0 do 255 i korisniku omogućuje postavljanje željene pozadinske boje za svaku sliku. Zadnja stavka predstavlja *RGB* komponente boje teksta koji se nalazi unutar slike. Također, kako su ranije navedene komponente boje pozadine, tako su navedene i boje teksta, a one su *Rtext*, *Gtext* i *Btext*. Opet, svaka komponenta ima također vrijednost u rasponu od 0 do 255 i korisniku omogućuje postavljanje željene boje teksta za bolju čitljivost na odabranoj pozadini. Bitno je proći kroz sve moguće kombinacije boje pozadine i teksta. Boja pozadine i teksta unutar JSON dokumenta su se odabirale na način da su se *RGB* komponente boje pozadine i teksta povećavali za 50. Drugim riječima, prvo se nasumično odabralo šest različitih boja te se unutar JSON dokumenta jedna pozadinska boja fiksirala i prolazilo se kroz sve ostale kombinacije koje predstavljaju boju teksta. Prikaz jednog JSON objekta prikazan je na slici 4.5.

5.2. Ocjenjivanje

Vrlo je važno da korisnici prođu kroz sve moguće kombinacije boje teksta i pozadine koje su dostupne unutar *pictures.json* dokumenta. Ukupno se unutar JSON dokumenta nalazi 30 različitih kombinacija boje teksta i pozadine što znači da će korisniku na raspolaganje biti ponuđeno 30 različitih slika za ocijeniti. Stoga, korisnik ocjenjuje čitljivost zadanog teksta na određenoj boji pozadine ocjenama od jedan do pet, gdje ocjena jedan predstavlja jako lošu čitljivost dok ocjena pet odličnu čitljivost. Kada korisnik odabere ocjenu klikom na odgovarajući gumb, ta ocjena se pohranjuje u odgovarajuću varijablu predstavljena *useState()* metodom. Ukoliko korisnik pokuša prijeći na slijedeću sliku bez odabrane ocjene, prikazuje se poruka upozorenja. Odabrana ocjena sprema se u bazu podataka nakon što korisnik stisne gumb „Slijedeći element“ na način da se poziva *addDoc()* metoda i unutar nje parametar *rating* predstavlja ocjenu koju je korisnik odabrao.

Ako korisnik promijeni svoj odabir ocjene, poziva se metoda *onChange()* u kojoj *setSelectedOption* metoda postavlja parametar na ocjenu koju je korisnik odabrao. Drugim riječima, ako se korisnik na početku odluči dati ocjenu dva, poziva se *onChange()* metoda i njegov odabir se sprema u varijablu zaduženu za spremanje korisnikove ocjene. Ukoliko korisnik promijeni svoj odabir, poziva se metoda *onChange()* i sprema se novo odabrana ocjena. Ukupno je u ocjenjivanju sudjelovalo šest korisnika.

5.3. Rezultati

Na temelju ocjenjenih slika, formira se konačna tablica koja prikazuje prosječnu ocjenu za svaku pojedinu sliku. Na temelju tablice 5.1. može se zaključiti kako slika pod rednim brojem 16 ima najveću prosječnu ocjenu na temelju šest korisnika, dok slika pod rednim brojem 10 ima najlošiju prosječnu ocjenu.

Tablica 5.1. Prikaz prosječnih ocjena za svaku sliku

Image ID	Average image rating
1	3.67
2	3.5
3	4.67
4	4
5	3.33
6	4
7	3.83
8	4.5
9	3.67
10	2.17
11	2.33
12	3.17
13	4.67
14	4.5
15	3.33
16	4.83
17	3.33
18	4

19	3.17
20	3.83
21	4.17
22	3
23	3.83
24	2.33
25	2.67
26	3
27	2.5
28	3.83
29	4
30	3.5

Zaključno s time, korisnici mogu vidjeti konačnu tablicu na kraju ocjenjivanja te svaki put kada bi novi korisnik ocijenio sve ponuđene slike, tablica bi se ažurirala. Shodno tome, na temelju dobivenih rezultata iz tablice, mogu se izdvojiti i prikazati slike s najboljim i najlošijim prosjekom ocjena. Tri najbolje ocjenjene slike označene s 1, 2 i 3 prikazane su ispod (Slika 5.1.).



Sl. 5.1. Tri najbolje ocjenjene slike



Sl. 5.2. Tri najbolje ocijenjene slike

S druge strane, iduće tri slike predstavljaju najbolje ocijenjene slike na temelju gore prikazane tablice (Slika 5.2).

6. ZAKLJUČAK

U ovom završnom radu, razvijena je aplikacija za ocjenjivanje čitljivosti teksta na temelju boje pozadine, istražujući kako kombinacije boja pozadine i teksta utječu na percepciju i čitljivost teksta kod korisnika. Aplikacija je omogućila korisnicima da pregledavaju i ocjenjuju slike s različitim bojama pozadine i teksta te pružila uvid u to kako te kombinacije boja utječu na njihovo iskustvo čitanja. Kroz analizu rezultata ocjenjivanja korisnika, uočeno je da odabrane kombinacije boja pozadine i teksta značajno utječu na čitljivost teksta. Visok kontrast između boja pozadine i teksta poboljšava čitljivost, dok nedostatak kontrasta može otežati čitanje. Aplikacija je također omogućila korisnicima da ocjenjuju slike s različitim kombinacijama boja, a rezultati ocjena koristili su se za izračun prosječnih ocjena za svaku kombinaciju boja. Ova analiza prosječnih ocjena pružila je dodatne uvide u preferencije korisnika i bolje razumijevanje utjecaja boje na čitljivost. Osim toga, ovaj završni rad istražuje implikacije za web dizajnere, *developere* i autore sadržaja. Naglašava se važnost pažljivog odabira boja za pozadinu i tekst kako bi se osigurala optimalna čitljivost i korisničko iskustvo. Također se razmatra potreba za alatima koji olakšavaju procjenu kontrasta boja i ocjenu čitljivosti. Zaključno s tim, ovaj završni rad pridonosi razumijevanju kako boje pozadine i teksta utječu na čitljivost teksta te pruža praktične smjernice za dizajniranje web stranica i aplikacija koje nude bolje iskustvo čitanja korisnicima. Istodobno, otvara vrata za daljnja istraživanja u području percepcije boja i čitljivost te razvoj alatima za analizu i optimizaciju boja na webu.

LITERATURA

- [1] F., Rašić, „Primjena boje u dizajnu i oblikovanju vizuala“, info:eu-repo/semantics/masterThesis, University North. University centre Varaždin. Department of Multimedia, Design and Application, 2022.
- [2] R., Jiménez, B., Redondo, R., Molina, M. Á., Martínez-Domingo, J., Hernández-Andrés, J., Vera, „Short-term effects of text-background color combinations on the dynamics of the accommodative response“, *Vision Res.*, sv. 166, str. 33–42, sij. 2020.
- [3] A., Agić, L., Mandić, N., Stanić Loknar „Legibility of typefaces and preferences of text/background color variations in virtual environment“, u *Proceedings - The Eleventh International Symposium GRID 2022*, str. 835–841, 2022.
- [4] K., Slović, „Web aplikacija za kupnju i čitanje e-knjiga“, info:eu-repo/semantics/bachelorThesis, Josip Juraj Strossmayer University of Osijek. Faculty of Electrical Engineering, Computer Science and Information Technology Osijek. Department of Software Engineering. Chair of Programming Languages and Systems, 2016.
- [5] E., Jergović, „Utjecaj boje na kvalitetu doživljaja grafičkog korisničkog sučelja“ [online], Grafički fakultet, , 27-ruj-2016. Dostupno na: <https://eprints.grf.unizg.hr/2606/>. [Pristupljeno: 7.9.2023.].
- [6] J., Grobelny, R., Michalski, „The role of background color, interletter spacing, and font size on preferences in the digital presentation of a product“, *Comput. Hum. Behav.*, sv. 43, str. 85–100, velj. 2015.
- [7] „Getting Started – React“ [online]. Dostupno na: <https://legacy.reactjs.org/docs/getting-started.html>. [Pristupljeno: 7.9.2023.].
- [8] „What is Google Firebase?“ [online]. Dostupno na: <https://www.techtarget.com/searchmobilecomputing/definition/Google-Firebase>. [Pristupljeno: 7.9.2023.].
- [9] „Using the State Hook – React“ [online]. Dostupno na: <https://legacy.reactjs.org/docs/hooks-state.html>. [Pristupljeno: 7.9.2023.].
- [10] „useNavigate v6.15.0“ [online]. Dostupno na: <https://reactrouter.com/en/main/hooks/use-navigate>. [Pristupljeno: 7.9.2023.].
- [11] Napoleon, „React Router useLocation hook - Tutorial and Examples“ [online], 26-pros-2020. Dostupno na: <https://www.kindacode.com/article/react-router-uselocation-hook-tutorial-and-examples/>. [Pristupljeno: 7.9.2023.].

SAŽETAK

Završni rad istražuje razvoj i primjenu aplikacije za ocjenjivanje čitljivosti teksta, uzimajući u obzir boju samog teksta i boju pozadine. Cilj rada je poboljšati iskustvo čitatelja na digitalnim platformama, prilagodbom kontrasta teksta i pozadinske boje radi bolje čitljivosti. Istraživanje uključuje analizu relevantnih faktora za čitljivost, kao što su tipografija, veličina fonta i kontrast. Razvijena aplikacija omogućava ocjenjivanje čitljivosti teksta od strane korisnika i automatsku analizu rezultata. Kroz testiranje na različitim uzorcima tekstova i kombinacija boja, utvrđena je optimalna kombinacija za bolju čitljivost. Rezultati ukazuju na potencijalno značajno poboljšanje čitljivosti uz pravilan odabir boja. Ova aplikacija može biti korisna za dizajnere web stranica, aplikacija i e-knjiga kako bi poboljšali korisničko iskustvo i olakšali čitanje osobama s različitim vizualnim potrebama.

Ključne riječi: boja teksta, boja pozadine, čitljivost teksta, Firebase, ReactJS

ABSTRACT

Application for evaluating the readability of colored text

This final paper investigates the development and application of an application for evaluating text readability, taking into account background color. The aim of the work is to improve the reader's experience on digital platforms by adjusting the contrast of the text and the background color for better readability. The research includes analysis of relevant factors for readability, such as typography, font size and contrast. The developed application uses algorithms for automatic evaluation of readability based on default colors of the background and text. Through testing on different text samples and color combinations, the optimal combination for better readability was determined. The results indicate a potentially significant improvement in readability with proper color selection. This application can be useful for designers of websites, applications and e-books to improve the user experience and make reading easier for people with different visual needs.

Key words: background color, Firebase, ReactJS, readability of the text, text color,

ŽIVOTOPIS

Josip Ostović rođen je 21.5.2000. godine u Slavonskom Brodu. Osnovnu školu pohađao je prvo u Zadru od 2007. do 2009. godine te se nakon toga preselio u Slavonski Brod. Osnovnu školu je nastavio i završio ju 2015. godine. Pohađao je nakon toga prirodoslovno-matematičku gimnaziju od 2015. do 2019. godine i nakon toga upisuje fakultet. Trenutno pohađa fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Student je treće godine preddiplomskog sveučilišnog studija računarstva smjera programsko inženjerstvo.

Potpis autora