Web aplikacija za studentsko istorazinsko ocjenjivanje

Anđelić, Dominik

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://urn.nsk.hr/urn:nbn:hr:200:760236

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-11-30**

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

Stručni studij

WEB APLIKACIJA ZA STUDENTSKO ISTORAZINSKO OCJENJIVANJE

Završni rad

Dominik Anđelić

Osijek, 2023.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1S: Obrazac za imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Osijek, 16.09.2023.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Ime i prezime Pristupnika:	Dominik Anđelić
Studij, smjer:	Stručni prijediplomski studij Računarstvo
Mat. br. Pristupnika, godina upisa:	AR 4770, 27.07.2020.
OIB Pristupnika:	77528248912
Mentor:	doc. dr. sc. Hrvoje Leventić
Sumentor:	,
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	doc. dr. sc. Krešimir Romić
Član Povjerenstva 1:	doc. dr. sc. Hrvoje Leventić
Član Povjerenstva 2:	dr. sc. Marija Habijan
Naslov završnog rada:	Web aplikacija za studentsko istorazinsko ocjenjivanje
Znanstvena grana završnog rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak završnog rada	Istražiti i opisati alate i metode za istorazinsko ocjenjivanje u nastavi (peer grading). U praktičnom dijelu implementirati web aplikaciju za peer grading u nastavi. Rezervirano za: Dominik Anđelić
Prijedlog ocjene pismenog dijela ispita (završnog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	16.09.2023.

Potvrda mentora o predaji konačne verzije rada:	Mentor elektronički potpisao predaju konačne verzije.
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 08.10.2023.

Ime i prezime studenta:	Dominik Anđelić
Studij:	Stručni prijediplomski studij Računarstvo
Mat. br. studenta, godina upisa:	AR 4770, 27.07.2020.
Turnitin podudaranje [%]:	3

Ovom izjavom izjavljujem da je rad pod nazivom**: Web aplikacija za studentsko istorazinsko ocjenjivanje**

izrađen pod vodstvom mentora doc. dr. sc. Hrvoje Leventić

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

1.1. Zadatak završnog rada 1

2. TEORIJSKI PREGLED RADA 2

2.1. Pouzdanost i valjanost istorazinskog ocjenjivanja 2

2.2. Usporedba značajki postojećih sustava za istorazinsko ocjenjivanje 3

2.2.1. ComPAIR 4

2.2.2. CrowdGrader 6

2.2.3. PeerStudio 7

2.2.4. Peer Review Tool – FeedbackFruits 9

2.2.5. Peergrade 10

2.3. Usporedba ostalih servisa za istorazinsko ocjenjivanje s vlastitim rješenjem . 12

3. IZRADA APLIKACIJE ZA ISTORAZINSKO OCJENJIVANJE 13

3.1. Poslužiteljski dio aplikacije 14

3.1.1. Baza podataka i modeli 14

3.1.2. Kontroleri 16

3.1.3. Algoritam za odabiranje radova za ocjenjivanje 18

3.1.4. Sigurnost i autorizacija temeljene na JWT tokenu 19

3.2. Klijentski dio aplikacije 20

3.2.1. Arhitektura 20

3.2.2. Sučelje 21

3.2.3. Ocjenjivanje 25

4. ZAKLJUČAK 27

LITERATURA 29

SAŽETAK 30

ABSTRACT 31

1. UVOD

Istorazinsko ocjenjivanje (engl. *peer grading*) je metoda ocjenjivanja u kojoj studenti ocjenjuju radove svojih kolega prema unaprijed definiranim kriterijima i skalama. Cilj istorazinskog ocjenjivanja jest poboljšati kvalitetu povratne informacije, potaknuti kritičko mišljenje i suradnju među studentima te smanjiti opterećenje nastavnika. Web aplikacija za studentsko istorazinsko ocjenjivanje je *online* platforma koja omogućava jednostavnu provedbu istog u različitim obrazovnim kontekstima. Uz uspješno izvršenje te detaljno objašnjenje zadatka završnog rada koji će biti predstavljen u podsekciji niže, u radu će se predstaviti glavne značajke i prednosti web aplikacija za istorazinsko ocjenjivanje, kao i neki izazovi i ograničenja koje takav način ocjenjivanja nosi. Također, prikazat će se i usporediti već postojeće aplikacije za istorazinsko ocjenjivanje, kao što su: ComPAIR, CrowdGrader, PeerStudio i druge.

1.1. Zadatak završnog rada

Web aplikacija omogućava nastavnicima da kreiraju kolegije i zadatke za studente te da pokrenu proces ocjenjivanja u kojem studenti rangiraju radove svojih kolega prema zadanim kriterijima. Studenti predaju svoja rješenja na zadane zadatke te ocjenjuju radove kolega. Web aplikacija također prikazuje rezultate ocjenjivanja i statističke podatke o radovima i ocjenama nastavnicima. Web aplikacija je implementirana korištenjem tehnologija Django, Next.js, SQLite te Bootstrap. U radu će biti detaljno opisane funkcionalnosti i arhitektura web aplikacije, kao i način njezinog korištenja.

2. TEORIJSKI PREGLED RADA

Aplikacije za istorazinsko ocjenjivanje (engl. *peer grading*) su *online* alati koji omogućuju studentima da ocjenjuju i daju povratne informacije o radu svojih kolega. Ove aplikacije imaju mnoge značajke i prednosti koje mogu poboljšati učenje i poučavanje. Neke od njih su:

- brže i bolje povratne informacije: studenti mogu dobiti povratne informacije od više kolega u kraćem vremenu, što im može pomoći da poboljšaju svoj rad i razumiju kriterije ocjenjivanja.
- poticanje višeg reda razmišljanja: Sadler i Good [1] u svom radu navode da učenici razvijaju sposobnost preuzimanja inicijative u ocjenjivanju vlastitog rada i korištenja vještina razmišljanja višeg reda za prosuđivanje o tuđem radu.
- suradnja i interakcija: studenti mogu surađivati i komunicirati sa svojim vršnjacima kroz proces ocjenjivanja i davanja povratnih informacija. Mogu raspravljati o problemima, pitanjima i rješenjima te dati jedni drugima podršku i poticaj.
- ušteda vremena i resursa: nastavnici mogu smanjiti svoje opterećenje ocjenjivanja i davanja povratnih informacija i usmjeriti svoju pažnju na druge aspekte poučavanja. Također, mogu pratiti napredak i učinak studenata kroz analitiku koju pružaju aplikacije za istorazinsko ocjenjivanje.

Prema Toppingu [2]*, peer grading* aplikacije mogu biti korisne za različite vrste zadataka, kao što su eseji, programski kodovi, dizajn, itd. Međutim, one također zahtijevaju određenu pripremu i podršku nastavnika da bi bile učinkovite. Nastavnici moraju osigurati jasne upute, rokove i kriterije za ocjenjivanje i davanje povratnih informacija. Također moraju osigurati pouzdanost, pravednost i anonimnost ocjenjivanja i riješiti eventualne probleme ili žalbe koje mogu nastati prilikom korištenja. Studenti moraju biti motivirani i spremni sudjelovati u ocjenjivanju svojih kolega te bi trebali razviti vještine potrebne za davanje kvalitetnih povratnih informacija.

2.1. Pouzdanost i valjanost istorazinskog ocjenjivanja

Kako navodi Topping [2], pouzdanost i valjanost ocjenjivanja nastavnika nije visoka. Procjena kolega obično je barem jednako visoka, a često i viša. Pouzdanost ima tendenciju biti veća u odnosu na: stupanj napredovanja u kolegiju, prirodu proizvoda ili učinka koji se procjenjuje, opseg u kojem se o kriterijima raspravljalo i pregovaralo, prirodu instrumentarija za ocjenjivanje, prakse i sl.

Prema istraživanjima koja su provedena u osnovnoj školi gdje su bili uključeni učenici 4. razreda i oni s posebnim obrazovnim potrebama, istorazinsko ocjenjivanje pisanog oblika radova je bilo izrazito učinkovito. U tom primjeru, istorazinsko ocjenjivanje je bilo barem jednako učinkovito u formativnom smislu kao i ocjenjivanje nastavnika, a ponekad i učinkovitije. [2]

Sadler i Good [1] u svojem radu navode da kada se govori o istorazinskom ocjenjivanju studenata, najbolje je ne razmišljati kao o izoliranoj obrazovnoj metodi, već kao o dijelu sustava učenja i ocjenjivanja koji se provodi na kolegiju. To uključuje dijeljenje sa studentima dio moći koju tradicionalno ima nastavnik, moći ocjenjivanja. Nastavnici bi trebali obučavati svoje učenike vještinama potrebnim za preciznije i točnije ocjenjivanje i trebali bi nadzirati točnost studenata u samostalnom ili istorazinskom ocjenjivanju. Kada se koristi odgovorno, ocjenjivanje studenata može biti vrlo precizno i pouzdano, štedeći vrijeme nastavnika.

Nadalje, navode kako bi nastavnici trebali biti svjesni da studenti s lošijim uspjehom imaju tendenciju napuhati vlastite ocjene, dok ocjene studenata s boljim uspjehom mogu patiti kada ih ocjenjuju drugi. Nijedna od ovih tendencija neće promijeniti poredak radova. U istraživanju koje su proveli Falchikov i Baud [3], gdje su se bavili samo-ocjenjivanjem studenata, utvrđeno je da su iskusniji studenti podcjenjivali vlastite radove, ali nije utvrđena ukupna dosljedna tendencija da se radovi podcijene ili precijene. Studente treba osposobiti za ispravno ocjenjivanje i za to ih treba nagraditi. Ako se loše provedu, ocjene mogu biti nepouzdane i učenici možda neće naučiti ništa vrijedno iz procesa. Ocjenjivanje vršnjaka može koristiti nastavnicima i studentima, ako se pažljivo provodi.

2.2. Usporedba značajki postojećih sustava za istorazinsko ocjenjivanje

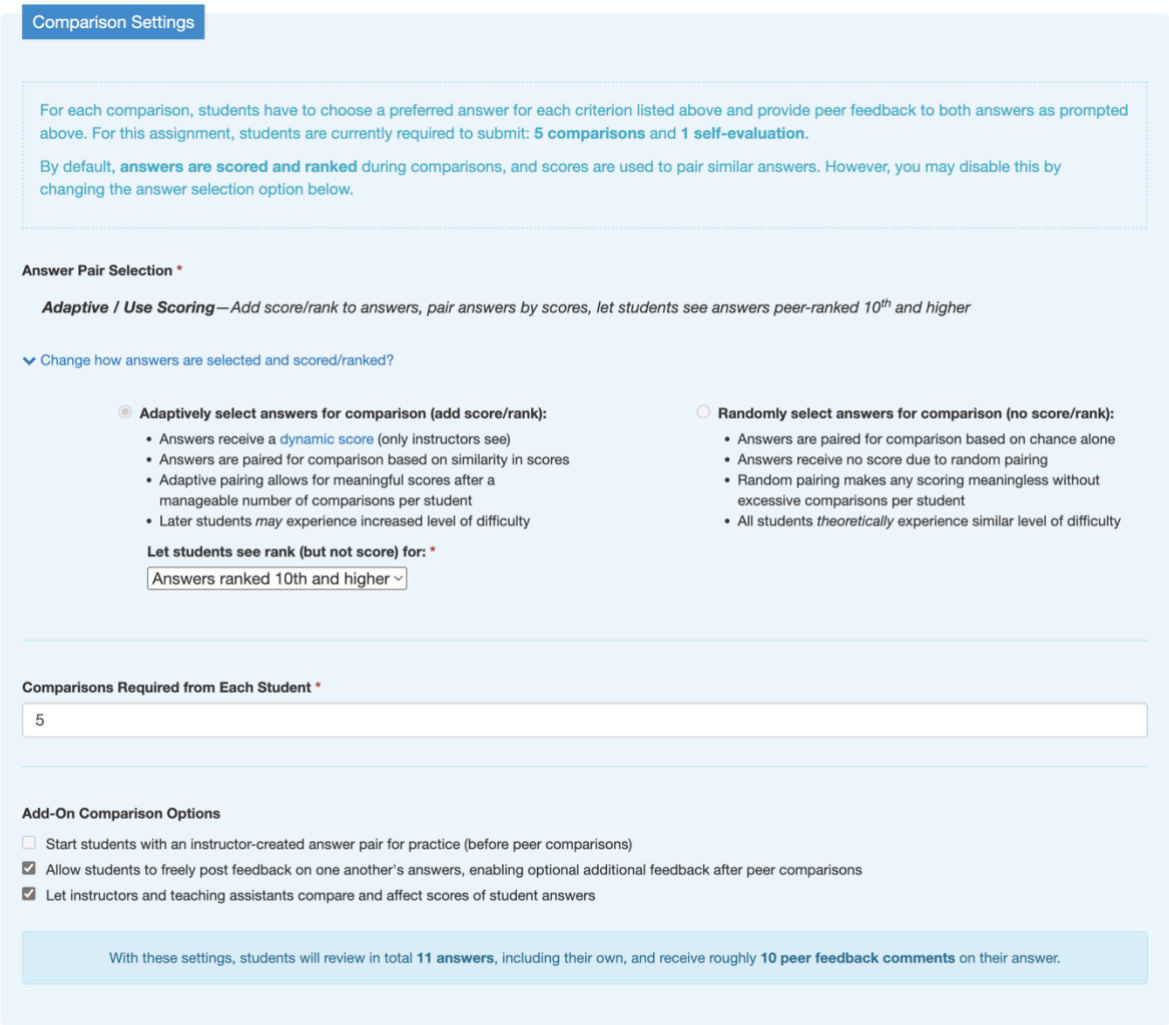
Istorazinsko ocjenjivanje je proces u kojem studenti ocjenjuju i daju povratne informacije o radu svojih vršnjaka. Ovaj proces ima brojne prednosti za učenje i poučavanje, kao što su brže i bolje povratne informacije, viša razina kritičkog razmišljanja, suradnja i interakcija te ušteda vremena i resursa [4]. Međutim, istorazinsko ocjenjivanje također zahtijeva određenu pripremu i podršku da bi bilo učinkovito. Nastavnici moraju osigurati jasne upute, rokove i kriterije za ocjenjivanje i davanje povratnih informacija. Također moraju osigurati pouzdanost, pravednost i anonimnost ocjenjivanja te riješiti eventualne probleme ili žalbe koje mogu nastati. Studenti

moraju biti motivirani i spremni sudjelovati u ocjenjivanju svojih kolega, kao i razviti vještine potrebne za davanje kvalitetnih povratnih informacije.

Da bi olakšali proces istorazinskog ocjenjivanja, postoje različiti *online* alati koji nude različite značajke i mogućnosti. Ovi alati omogućuju studentima da prenesu svoj rad na *online* platformu i da ocjenjuju i daju povratne informacije svojim vršnjacima na temelju kriterija koje je postavio nastavnik. Također omogućuju nastavnicima da prate napredak i učinak studenata kroz analitiku koju pružaju ti alati. Neki od najpopularnijih alata za istorazinsko ocjenjivanje su Peergrade [5], FeedbackFruits - Peer Review Tool [6], PeerStudio [7], CrowdGrader [8] i ComPAIR [9]. U ovom potpoglavlju ćemo usporediti značajke tih alata i istražiti njihove prednosti i nedostatke.

2.2.1. ComPAIR

ComPAIR [9] je alat za istorazinsko ocjenjivanje koji se temelji na usporednoj evaluaciji i komparaciji među radovima studenata. Studenti vide radove dva različita studenta (ili dva različita tima) i moraju procijeniti koji bolje zadovoljava kriterije postavljene od strane nastavnika. Svaki student može imati dodijeljeno više parova evaluacije. Sustav može automatski odrediti (približne) ocjene na temelju rangiranja svih usporedbi koje su napravljene, a također mogu biti automatski dodijeljeni i bodovi za sudjelovanje. Ono što ComPAIR dodatno ima jest mogućnost da studenti daju povratne informacije drugim studentima čije odgovore ocjenjuju. Isto tako, moguće je i da nastavnik objavi rješenje za zadani zadatak [9]. Na Slici 2.1 u nastavku je prikazano sučelje ComPAIR alata gdje su vidljive postavke usporedbe za jedan zadatak.



Slika 2.1: Prikaz postavki za usporedbu unutar jednog zadatka u ComPAIR alatu

Iz navedene slike može se vidjeti kako postoje dvije opcije „posluživanja“ usporedbi studentima – adaptivna i nasumična usporedba. Adaptivna usporedba uparuje one odgovore koji su slični po rangu, a nasumična uparuje i daje studentima na uspoređivanje radove bez ikakvog posebnog algoritma.

Nekoliko značajki ComPAIRa su:

- anonimnost: studenti i nastavnici ne vide identitete drugih sudionika u procesu ocjenjivanja, osim ako nastavnik ne odluči drugačije.
- prilagodljivost: nastavnik može prilagoditi postavke ocjenjivanja, kao što su broj parova evaluacije, broj kriterija, rokovi, itd.
- povratne informacije: studenti mogu davati povratne informacije jedni drugima u obliku komentara ili ocjena. Nastavnik također može davati povratne informacije studentima o njihovom radu i ocjenjivanju.

- integracija: ComPAIR se može integrirati s drugim sustavima za upravljanje učenjem, kao što su Canvas i slični.

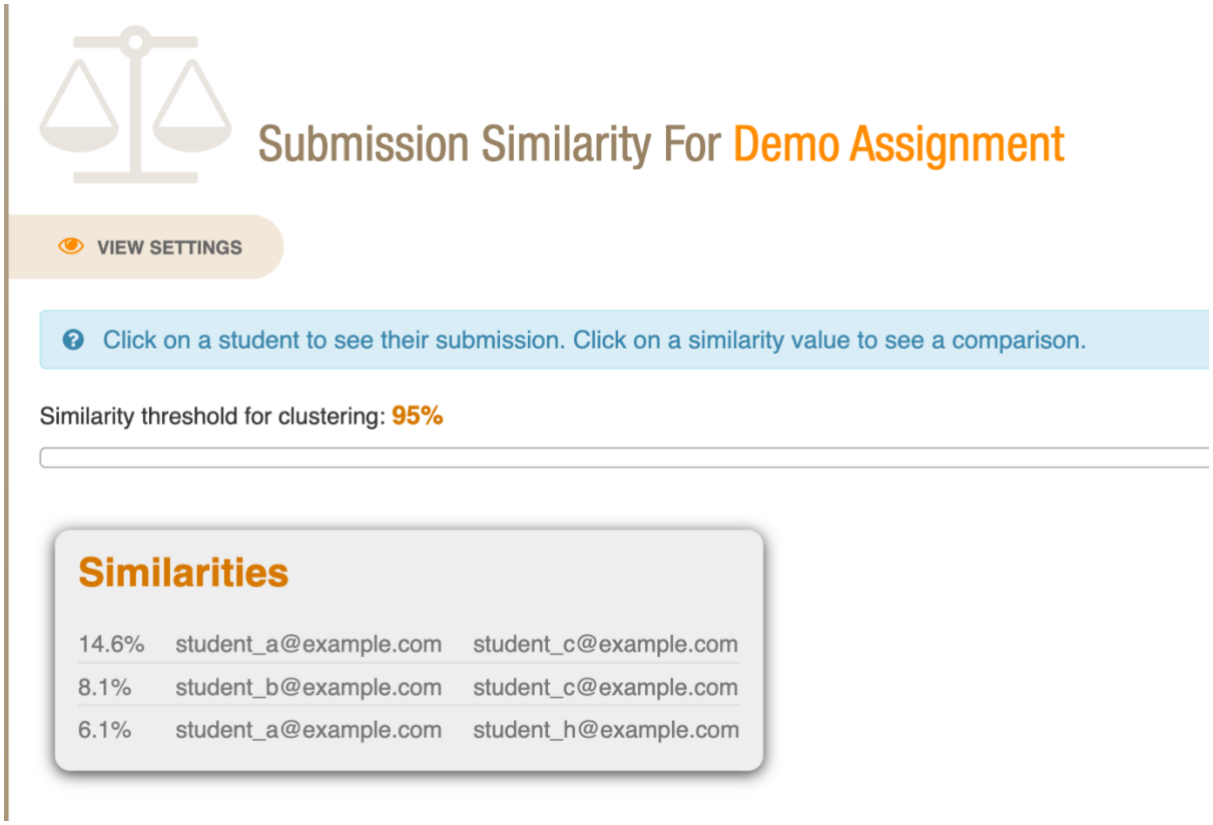
ComPAIR se razlikuje od drugih alata za ocjenjivanje studenata po tome što se temelji na usporednoj evaluaciji, koja je metoda ocjenjivanja u kojoj studenti uspoređuju rad dvoje kolega i odlučuju koji bolje zadovoljava kriterije. Ova metoda potiče kritičko razmišljanje i argumentiranje jer studenti moraju obrazložiti svoje izbore. ComPAIR također koristi adaptivni algoritam za odabir najboljih parova za usporedbu, koji uzima u obzir prethodne usporedbe i povratne informacije. Isto tako, ComPAIR omogućuje studentima da daju povratne informacije u obliku komentara ili ocjena te da vide povratne informacije koje su dobili od svojih vršnjaka i nastavnika.

2.2.2. CrowdGrader

CrowdGrader [8] je također alat za istorazinsko ocjenjivanje studenata koji omogućuje da studenti predaju i ocjenjuju svoje radove. Nekoliko značajki CrowdGradera su:

- kolaboracija: studenti mogu raditi pojedinačno ili u grupama i predavati svoja rješenja zadataka. CrowdGrader uključuje snažan uređivač (engl. *editor*) koji omogućuje nastavnicima i studentima da pišu izvještaje, matematičke formule, kod i još mnogo toga; studenti također mogu predati bilo koju vrstu datoteke (uključujući zip datoteke) na ocjenjivanje.
- ocjenjivanje: nakon što je faza predaje završena, od studenata se traži da ocijene određeni broj (obično 5) predanih rješenja drugih studenata. CrowdGrader izračunava za svakog studenta ocjenu koja ovisi o: ocjenama koje je dobilo rješenje koje je student predao; točnosti studenta u ocjenjivanju rješenja i korisnosti recenzija koje je pružio. Tako studenti imaju poticaj da budu točni u svojim ocjenama i korisni u svojim recenzijama.
- kontrola: nastavnik može koristiti ocjene koje su dodijelili studenti kao konačne ocjene ili kao polazište za dodjelu konačnih ocjena: nastavnik uvijek ima potpunu kontrolu. Nastavnik također ima pristup detaljnim informacijama o poteškoćama i problemima s kojima su se studenti susretali.

Ono što izdvaja CrowdGrader od ostalih alata za istorazinsko ocjenjivanje jest integrirana mogućnost provjere sličnosti između predanih radova, vidljiva na Slici 2.2.



Slika 2.2: Prikaz provjere sličnosti među predanim radovima u CrowdGraderu

Slika 2.2 prikazuje sličnosti u predanim radovima koje je alat uočio. S alatom je moguće pretraživati tekst kao i većinu priloga. Sličnosti su izražene u postocima.

CrowdGrader se razlikuje od drugih alata za ocjenjivanje vršnjaka po tome što koristi algoritam za izračunavanje ocjena na temelju usporedbi koje su napravili studenti. Studenti ne daju samo povratne informacije svojim vršnjacima, već i ocjenjuju kvalitetu povratnih informacija koje su primili. Sustav zatim koristi te ocjene da odredi konačnu ocjenu za svaki predani rad i za svakog studenta koji je sudjelovao u ocjenjivanju. CrowdGrader također nagrađuje studente koji daju točne i korisne povratne informacije i kažnjava one koji daju neprimjerene ili beskorisne povratne informacije.

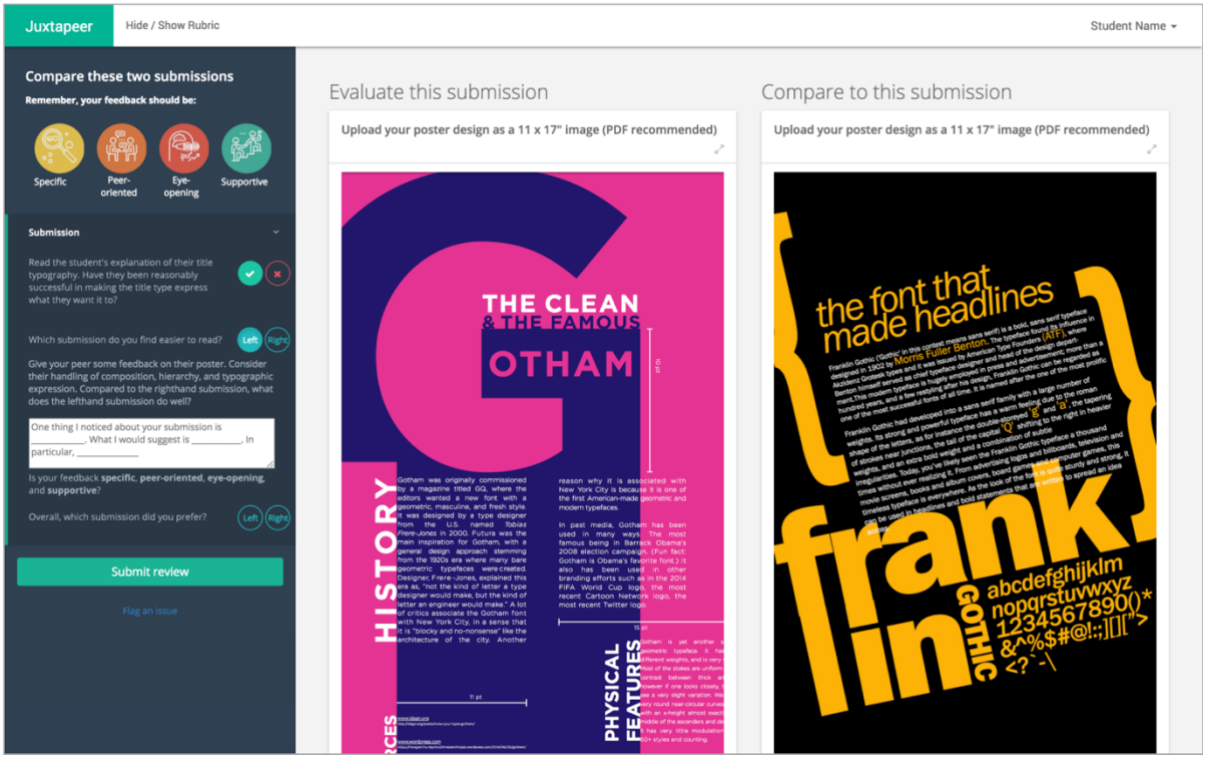
2.2.3. PeerStudio

PeerStudio [7] je alat za ocjenjivanje vršnjaka koji se temelji na usporedbi rješenja. Studenti vide rješenja dva različita vršnjaka i moraju procijeniti koja bolje zadovoljava kriterije koje je postavio nastavnik. Sustav koristi umjetnu inteligenciju da pronađe najprikladnije usporedbe za svakog studenta i rješenje. Također, sustav automatski otkriva loše ocjenjivanje i pomaže studentima da nauče kako bolje ocjenjivati.

Nekoliko značajki PeerStudia su:

- povratne informacije: studenti dobivaju povratne informacije od svojih vršnjaka, koje su često detaljnije od onih koje može pružiti nastavnik ili asistent. Također uče iz rješenja koja su predali drugi studenti i mogu usporediti svoja rješenja s njima.
- oblak (engl. *cloud*): PeerStudio je dostupan s bilo kojeg računala i omogućuje studentima da pregledavaju materijale za zadatke bez dodatnog softvera (sve je ugrađeno): PDF-ovi, videozapisi (uključujući automatsko ugrađivanje YouTubea), formatirani tekst (podebljano, kurziv, tablice, …).
- olakšano upravljanje: PeerStudio automatski dodjeljuje ocjenjivače i izračunava ocjene na temelju usporedbi. Nastavnik može koristiti te ocjene kao konačne ili kao polazište za dodjelu konačnih ocjena. Nastavnik također ima pristup detaljnim informacijama o poteškoćama i problemima s kojima su se susreli studenti.

PeerStudio se razlikuje od drugih alata za istorazinsko ocjenjivanje po tome što se temelji na teoriji kontrastnih slučajeva, koja govori da uspoređivanje sličnih radova pomaže ljudima da vide dublje i suptilnije razlike između njih. PeerStudio prilikom korištenja umjetne inteligencije pronalazi najprikladnije usporedbe za svakog studenta te stalno uči i poboljšava svoju točnost. Alat također ima interaktivne vodiče i automatsko otkrivanje lošeg ocjenjivanja koje pomažu studentima da nauče kako bolje ocjenjivati. Slika 2.3 prikazuje sučelje za usporedbu u PeerStudio aplikaciji.



Slika 2.3: Prikaz sučelja za uspoređivanje u PeerStudio aplikaciji

Student ocjenjuje samo rad koji se nalazi s lijeve strane, a rad koji se nalazi na desnoj strani mu služi kao baza usporedbe. Isto tako, uz pomoć naprednih algoritama i umjetne inteligencije studentu se poslužuju radovi za usporedbu na temelju njegovih prethodnih iskustava s ocjenjivanjem kao i njegovih kolega s kolegija.

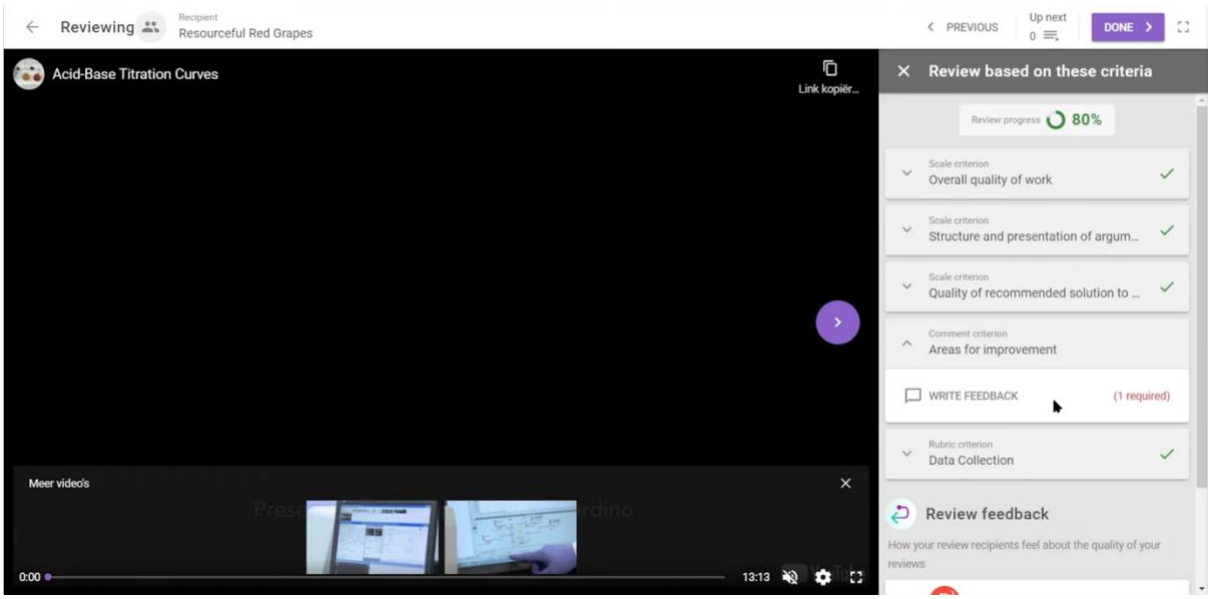
2.2.4. Peer Review Tool – FeedbackFruits

FeedbackFruits [10] je pedagoški alat koji omogućuje nastavnicima da kreiraju aktivnosti za studente koje potiču angažman, interakciju i suradnju. Jedan od alata je Peer Review [6], koji omogućuje nastavnicima da kreiraju zadatke za studente s kojim daju povratne informacije svojim vršnjacima na temelju kriterija koje je postavio nastavnik. Nekoliko značajki Peer Reviewa su:

- prilagodljivost: nastavnik može prilagoditi postavke zadatka, kao što su rokovi, kriteriji, anonimnost, broj recenzija itd. Također može odabrati između pojedinačnog ili grupnog ocjenjivanja.
- integracija: Peer Review se može integrirati s drugim alatima FeedbackFruitsa, kao što su Interactive Video, Interactive Document, Automated Feedback itd. Također se može integrirati s drugim sustavima za upravljanje učenjem, kao što su Canvas, Moodle, Blackboard itd.

- analitika: nastavnik može pratiti napredak i učinak studenata tijekom cijelog zadatka. Može vidjeti koliko su studenti sudjelovali u davanju i primanju povratnih informacija, kako su ocijenili svoje kolege i kako su se poboljšali.

FeedbackFruits se razlikuje od drugih alata za ocjenjivanje vršnjaka po tome što nudi više od jednog alata za poticanje angažmana, interakcije i suradnje među studentima. Peer Review je samo jedan od mnogih alata koji se mogu koristiti u različitim kontekstima i modalitetima učenja. FeedbackFruits također ima snažnu integraciju s drugim sustavima za upravljanje učenjem i omogućuje nastavnicima da prilagode svoje aktivnosti prema svojim ciljevima i potrebama. Taj skup alata je također zasnovan na istraživanjima i suradnji s inovativnim edukatorima koji pomažu u razvoju pedagoških rješenja. Slika 2.4 prikazuje sučelje za ocjenjivanje koje student vidi u Peer Review Toolu.



Slika 2.4: Prikaz sučelja za ocjenjivanje u Peer Review Toolu

Iz slike možemo vidjeti kako postoje kriteriji po kojima se mora ostaviti recenzija, odnosno ocjena te je moguće umetnuti i video kao predan rad. Na desnoj strani je prikazan postotak završenosti cijelog procesa ocjenjivanja, odnosno recenziranja te se može vidjeti i što ostali studenti misle o kvaliteti recenziranja tog studenta.

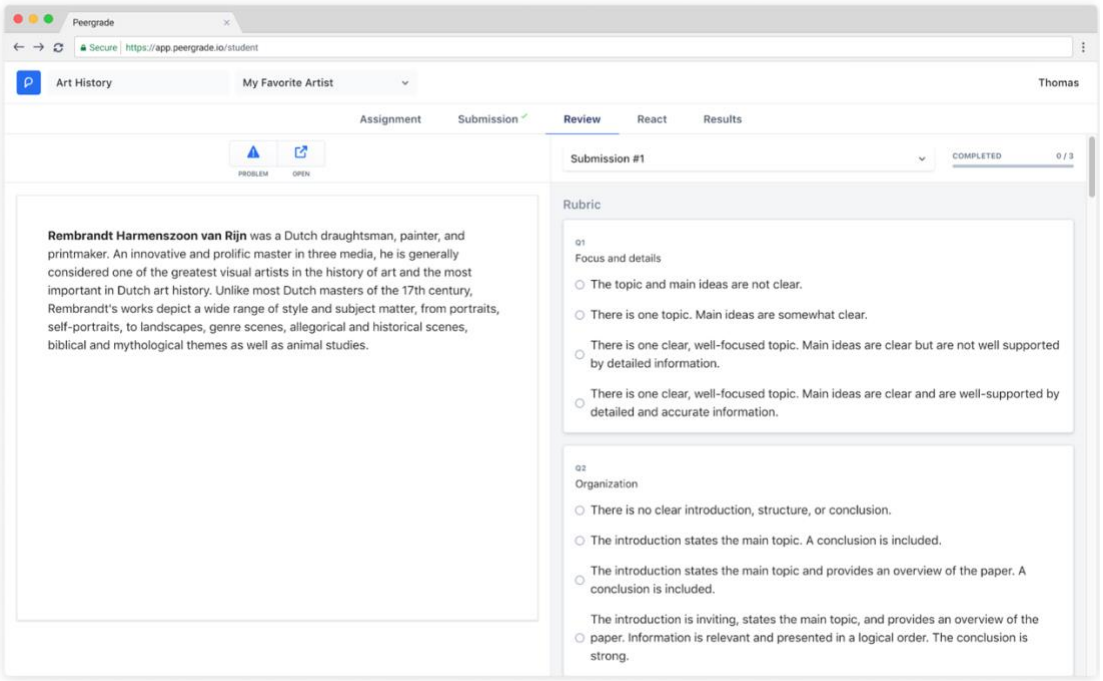
2.2.5. Peergrade

Peergrade [5] je *online* platforma koja omogućuje anonimne povratne informacije studenata. S Peergradeom studenti dobivaju brže i bolje povratne informacije. Značajke u Peergradeu kao

što su označavanje i rubrike za povratne informacije osiguravaju da studenti dobiju najbolje moguće povratne informacije i olakšavaju proces ocjenjivanja svojih kolega.
Nekoliko značajki Peergrade alata su:

- jednostavnost: nastavnik može jednostavno stvoriti zadatak za ocjenjivanje vršnjaka, postaviti kriterije, rokove i broj recenzija. Studenti mogu lako prenijeti svoj rad (pisanje, videozapisi, grafika itd.) na Peergrade i dati povratne informacije svojim kolegama na temelju kriterija koje je postavio nastavnik.
- povratne informacije: studenti dobivaju povratne informacije od svojih vršnjaka, koje su često detaljnije od onih koje može pružiti jedan nastavnik ili asistent. Također uče iz rješenja koja su predali drugi studenti i mogu usporediti svoja rješenja s njima.
- analitika: nastavnik može pratiti napredak i učinak studenata tijekom cijelog zadatka. Može vidjeti koliko su studenti sudjelovali u davanju i primanju povratnih informacija, kako su ocijenili svoje kolege i kako su se poboljšali.

Peergrade se razlikuje od drugih alata za istorazinsko ocjenjivanje po tome što nudi više mogućnosti za prilagodbu i kontrolu nad procesom ocjenjivanja. Nastavnik može odabrati između različitih načina distribucije rješenja, kao što su nasumično, ručno ili prema grupama. Također može odabrati između različitih načina ocjenjivanja, kao što su ocjene, bodovi ili zvjezdice. Nastavnik također može odlučiti hoće li povratne informacije biti anonimne ili ne, i hoće li studenti moći vidjeti ocjene i povratne informacije koje su dobili od kolega. Peergrade također ima značajku označavanja koja omogućuje studentima da označe neprimjerene ili beskorisne povratne informacije, koje nastavnik može pregledati i ispraviti. Slika 2.5 prikazuje kako to izgleda u Peergrade alatu kada student želi ocijeniti rad drugog studenta.



Slika 2.5: Prikaz sučelja za ocjenjivanje u Peergradeu

Slično kao Peer Review Tool, s lijeve strane nalazi se rad studenta kojeg se ocjenjuje te s desne strane se mogu vidjeti rubrike po kojima se ocjenjuje. U izborniku se može pogledati i koji je točno bio zadatak da se student koji ocjenjuje podsjeti. Naposlijetku se mogu vidjeti rezultati.

2.3. Usporedba ostalih servisa za istorazinsko ocjenjivanje s vlastitim rješenjem

Kada se uspoređuju konkurentski servisi s rješenjem autora ovog rada, može se doći do nekih zaključaka. Prvo, alat ComPAIR je najsličniji po značajkama onome što je napravljeno u ovome završnom radu u odnosu na ostale alate, iako svi alati generalno pokrivaju istu tematiku. Ono što je jako zanimljivo jest upotreba umjetne inteligencije za preporuku koje koristi PeerStudio te bi se to moglo smatrati kao nešto s čim bi se rješenje ovog rada moglo poboljšati. Neki od navedenih alata koriste određeni rad kao „baznu točku“ za ocjenjivanje ostalih radova, dok ostali uspoređuju radove isključivo u parovima. Vrlo dobro izvedeno rješenje je da grupe studenata se mogu međusobno ocjenjivati, što bi se moglo činiti kao neki prijelazni put do potpuno individualnog ocjenjivanja studenata. PeerGrader i PeerReview koriste sustav gdje se mora ocijeniti rad studenta na temelju predefiniranih „rubrika“ te se koristi skala koja se nakon završetka ocjenjivanja ažurira i to se prezentira nastavniku. Rješenje ovog rada trenutno je inferiornije u odnosu na već gotove alate kao što su CrowdGrader, ComPAIR, PeerStudio i drugi, no pokazalo se da tržište postoji za takve aplikacije. Pokaže li se uspješno ovaj alat u svakodnevnom radu, kod je strukturiran da se dodatne funkcionalnosti mogu lako i brzo implementirati.

3. IZRADA APLIKACIJE ZA ISTORAZINSKO OCJENJIVANJE

Aplikacija se temeljno sastoji od dva dijela – poslužiteljskog i klijentskog. Poslužiteljski dio bavi se svom pozadinskom i poslovnom logikom koja je potrebna kako bi se postiglo istorazinsko ocjenjivanje, a klijentski dio je potreban za prikaz korisničkog sučelja. S poslužiteljske strane korišten je Python [11] programski jezik te Django [12] razvojni okvir, upotpunjen moćnom bibliotekom Django Ninja [13]. Na klijentskoj strani korišten je TypeScript [14] programski jezik, Next.js [15] razvojni okvir koji „ispod haube“ ima React [16] biblioteku. Za ljepši izgled klijentske strane korištena je Bootstrap [17] biblioteka.

Budući da je riječ o klijent-poslužitelj arhitekturi, poslužiteljski dio aplikacije isključivo služi kao REST (engl. *Representational State Transfer*) API (engl. *Application Programming Interface*) prema kojem se vrše pozivi preko klijentskog dijela te se odgovori prezentiraju korisniku [18]. Sva potrebna logika izvedena je preko poslužitelja kako korisnik ne bi mogao manipulirati podacima, što je izuzetno bitno za projekt kao što je istorazinsko ocjenjivanje.

Klijentski dio aplikacije oslanja se na odgovore koje dobije od strane poslužitelja te to prezentira korisniku. Postoje tri tipa korisnika: administratori, nastavnici i studenti. Ovisno koji je tip korisnika prijavljen, različite mu se mogućnosti otvaraju ili zatvaraju.

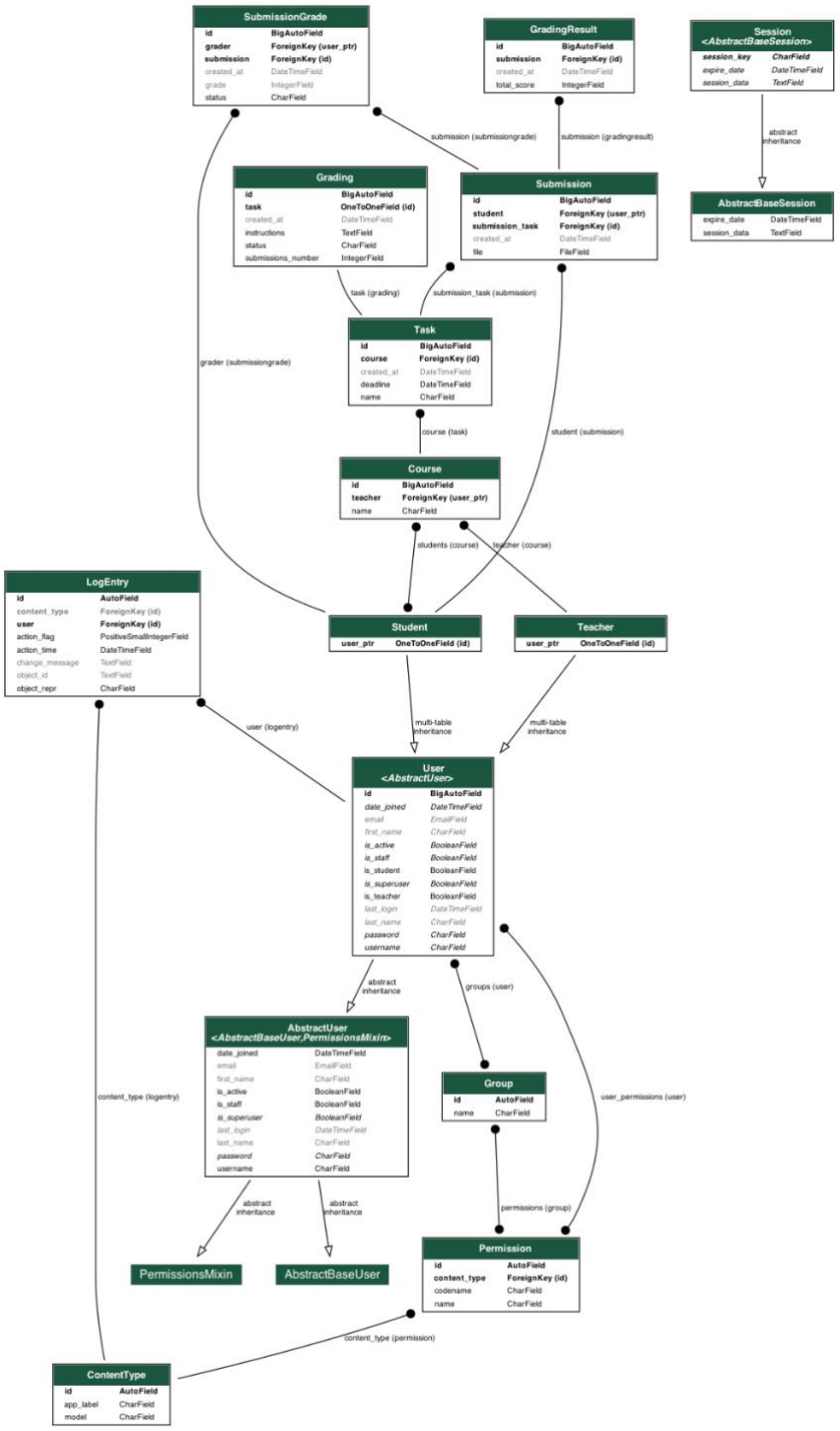
Tri tipa korisnika nose i tri različita skupa mogućnosti:

- nastavnici - mogu kreirati kolegije, upisivati studente na kolegije, dodavati zadatke za one kolegije čiji su oni nositelji, započinjati fazu ocjenjivanja, završiti ocjenjivanje i prikazati rezultate
- administratori – ponašaju se kao nastavnici s najvećim ovlaštenjima (mogu sve uređivati i brisati)
- studenti – mogu predavati svoja rješenja zadataka za one kolegije na koje su upisani, pregledavati rezultate ocjenjivanja

3.1. Poslužiteljski dio aplikacije

3.1.1. Baza podataka i modeli

U ovom radu se koristi Django razvojni okvir za izradu web aplikacije za istorazinsko ocjenjivanje. Django koristi relacijsku bazu podataka za pohranu podataka o modelima, koji su klase koje definiraju strukturu i ponašanje podataka u aplikaciji. U nastavku na Slici 3.1 slijedi prikaz relacijskog dijagrama modela.



Slika 3.1: Relacijski model baze podataka

Kao što se može vidjeti iz Slike 3.1., web aplikacija se sastoji od sljedećih modela (opisat će se modeli bitni za aplikaciju):

- **`User`**: nasljeđuje klasu **`AbstractUser`** iz Django.contrib.auth.models modula, koja pruža osnovne funkcionalnosti za autentifikaciju i autorizaciju korisnika. Dodaju se dva atributa **`is_student`** i **`is_teacher`**, koji su BooleanField tipa, kako bi se razlikovale uloge korisnika u aplikaciji.
- **`Teacher`**: nasljeđuje klasu **`User`** i ne dodaje nikakve dodatne atribute. Ovaj model služi za označavanje korisnika koji su nastavnici u aplikaciji te se u budućnosti može dodatno proširiti.
- **`Student`**: Ovaj model također nasljeđuje klasu **`User`** i ne dodaje nikakve dodatne atribute. Ovaj model služi za označavanje korisnika koji su studenti u aplikaciji te se u budućnosti može dodatno proširiti.
- **`Course`**: predstavlja kolegij u aplikaciji. Model se sastoji od dva atributa: **`name`**, koji je CharField tipa i sadrži naziv kolegija te atribut **`teacher`**, koji je ForeignKey tipa i povezuje kolegij s nastavnikom koji ga vodi. ForeignKey je vrsta relacije koja omogućava da jedan objekt pripada drugom objektu. Osim toga, ovaj model ima atribut **`students`**, koji je ManyToManyField tipa i povezuje kolegij s više studenata koji ga pohađaju. ManyToManyField je vrsta relacije koja omogućava da više objekata pripada više drugih objekata te se stvara nova tablica u bazi.
- **`Task`**: predstavlja zadatak u aplikaciji. Ovaj model sastoji se od četiri atributa: **`name`**, koji je CharField tipa i sadrži naziv zadatka; **`course`**, koji je ForeignKey tipa i povezuje zadatak s kolegijem kojem pripada; **`created_at`**, koji je DateTimeField tipa i sadrži datum i vrijeme kada je zadatak kreiran te **`deadline`**, koji je također DateTimeField tipa i predstavlja rok za predaju zadatka.
- **`Submission`**: predstavlja rješenje koje student preda za neki zadatak unutar aplikacije. Model ima četiri atributa: **`file`**, koji je FileField tipa i sadrži datoteku koju je student poslao kao rješenje zadatka (ograničenje svakog zadatka jest da rješenje treba biti predano u PDF formatu); **`submission_task`**, koji je ForeignKey tipa i povezuje rješenje sa zadatkom kojem pripada; **`student`**, koji je ForeignKey i povezuje studenta s rješenjem zadatka te **`created_at`** koji je DateTimeField tipa i sadrži datum i vrijeme kada je rješenje predano.

- **`Grading`**: predstavlja komponentu ocjenjivanja, vezano je uz zadatak. Model sadrži 5 atributa: **`task`** koji je ForeignKey tipa, povezuje postavke ocjenjivanja sa zadatkom; **`created_at`** koji je DateTimeField tipa i sadrži datum i vrijeme kad su postavke ocjenjivanja kreirane; **`instructions`** koji je CharField tipa i sadrži upute za ocjenjivanje; **`status`** koji je CharField tipa i predstavlja status ocjenjivanja te **`submissions_number`** koji je IntegerField tipa i predstavlja broj radova koje svaki student uspoređuje i ocjenjuje za taj zadatak.
- **`SubmissionGrade`**: predstavlja ocjenu koju je student dao za rad drugog kolege. Sastoji se od 5 atributa: **`grader`**, koji je ForeignKey tipa i povezuje ocjenu s ocjenjivačem; **`submission`** koji je ForeignKey tipa i povezuje ocjenu s predanim radom; **`created_at`** koji je DateTimeField tipa i predstavlja datum i vrijeme kad je ocjena kreirana; **`grade`** koji je IntegerField tipa i predstavlja ocjenu koju je student dao za rad te **`status`** koji je CharField tipa te predstavlja status ocjene.
- **`GradingResult`**: predstavlja rezultat ocjenjivanja za pojedini zadatak. Sadrži 3 atributa: **`submission`** koji je ForeignKey tipa i povezuje rezultat sa zadatkom; **`created_at`** koji je DateTimeField tipa i predstavlja datum kreiranja rezultata i **`total_score`** koji je IntegerField tipa i predstavlja sumu svih ocjena koje je rad dobio.

3.1.2. Kontroleri

U ovom radu se koristi Django Ninja biblioteka za izradu REST API-ja zajedno s Django razvojnim okvirom. Django Ninja je biblioteka koja omogućava brz i jednostavan razvoj API-ja s visokim stupnjem sigurnosti, skalabilnosti i održivosti. Django Ninja koristi Pydantic [19] biblioteku za validaciju i serijalizaciju podataka te podržava asinkrono izvršavanje zahtjeva. Django Ninja također generira OpenAPI [20] shemu za definirane operacije, koja se može pregledati i testirati putem interaktivne Swagger [21] dokumentacije.

U ovom radu se koristi NinjaExtraAPI klasa iz Django Ninja Extra [22] biblioteke, koja proširuje osnovnu NinjaAPI klasu s dodatnim funkcionalnostima, kao što su registracija kontrolera, prilagođeni rendereri i iznimke (engl. *exceptions*). Tipično u Djangu [12], pogled *(*engl. *view)* opisuje koji su podaci predstavljeni te predložak (engl. *template*) opisuje kako su podaci predstavljeni. U ovom radu su umjesto pogleda korišteni klasni kontroleri iz Django Ninja Extra biblioteke koji omogućuju enkapsulaciju povezanih funkcija i ruta u jednu klasu, što je korisno za organiziranje i čitljivost koda.

Kontroler je klasa koja sadrži metode koje obrađuju različite HTTP metode i putanje za određeni resurs. Kontroler također prima i vraća podatke u obliku Pydantic modela, koji definiraju strukturu i tipove podataka za ulazne i izlazne sheme. Sheme su definirane unutar *rest/schemas* direktorija te se sastoje od ulaznih shema i izlaznih shema. Ulazne sheme se odnose na zahtjeve od strane korisnika, a izlazne sheme su odgovori od strane poslužitelja. Ovakav pristup pomoću kontrolera iz Django Ninja Extra biblioteke je odabran budući da je autor završnog rada otprije upoznat s istima koristeći druge razvojne okvire.

U ovom radu se definira šest kontrolera: **`MyTokenObtainPairController`**, **`CourseController`**, **`SubmissionController`**, **`TaskController`**, **`UserController`** i **`GradingController`**. Svaki kontroler nasljeđuje api_controller dekorator iz ninja_extra biblioteke, koji omogućava registraciju kontrolera na API instanci. Dekorator također prima argumente kao što su tags, koji označavaju kategoriju kontrolera u dokumentaciji, i auth, koji označava način autentifikacije korisnika.

Kontroleri su grupirani u logičke cjeline i dijele se na:

- **`MyTokenObtainPairController`**: nasljeđuje klasu **`TokenObtainPairController`** iz ninja_jwt biblioteke, koja omogućava autentifikaciju korisnika putem JSON Web Tokena (JWT). JWT je sigurni način prijenosa podataka između dvije strane, koji se sastoji od tri dijela: zaglavlja (engl. *header*), tereta (engl. *payload*) i potpisa (engl. *signature*). Zaglavlje sadrži informacije o algoritmu i tipu tokena, teret sadrži podatke o korisniku i njegovim ovlastima, a potpis služi za provjeru integriteta i autentičnosti tokena [23]. Ovaj kontroler ima jednu **`POST`** metodu **`obtain_token`**, koja prima korisničko ime i lozinku korisnika, te vraća par tokena: pristupni token (engl. *access token*) i token za osvježavanje (engl. *refresh token*) kao i prijavljenog korisnika. *Access token* je kratkotrajni token koji se koristi za pristup API-ju, dok je *refresh token* dugotrajni token koji se koristi za obnavljanje *access tokena* kada on istekne. U slučaju da istekne i token za osvježavanje, korisnik se pomoću korisničkog sučelja automatski odjavljuje budući da nije moguće dobiti novi pristupni token ukoliko se ponovno ne prijavi.
- **`CourseController`**: sadrži metode koje se odnose na resurs kolegija u aplikaciji. Kontroler sadrži četiri REST metode: **`GET`**, **`POST`**, **`PUT`** i **`DELETE`**. **`GET`** metode odnose se na dohvaćanje ili liste kolegija ili pojedinačnog kolegija. **`POST`** prima podatke o novom

kolegiju u obliku **CreateCourseRequest** sheme i stvara novi objekt kolegija u bazi podataka. **PUT** metode odnose se na ažuriranje podataka o kolegiju i dodavanje studenata na kolegij. **DELETE** metoda prima parametar **course_id** i briše objekt kolegija iz baze podataka.

- **SubmissionController**: sadrži metode koje se odnose na rješenje zadatka u aplikaciji. Sadrži tri REST metode: **GET**, **POST** i **PUT**. **GET** metoda se odnosi ili na dohvaćanje pojedinačnog rješenja ili svih rješenja, primjerice za neki kolegij. **POST** metoda odnosi se na kreiranje novog rješenja zadatka u bazi, uz validaciju da je predana PDF datoteka u zahtjevu. **PUT** metoda mijenja postojeći predani rad.
- **TaskController**: sadrži metode koje se odnose na zadatak u aplikaciji. Sastoji se od 5 različitih REST metoda: **GET**, **POST**, **PUT**, **PATCH** i **DELETE**. **GET** metode odnose se na dohvat pojedinog zadatka ili liste zadataka, ovisno o potrebi. **POST** metoda se odnosi na kreiranje zadatka u aplikaciji, a **PUT** na ažuriranje istog. **PATCH** metoda je korištena za ažuriranje statusa zadatka od strane nastavnika. **DELETE** metoda koristi se za brisanje resursa zadatka.
- **UserController** sadrži metode koje se odnose na korisnika u aplikaciji. Ima 4 različite REST metode: **GET**, **POST**, **PUT** i **DELETE**. **GET** metode odnose se na vraćanje podataka o korisniku: nastavniku ili studentu. **POST** metode odnose se na registraciju nastavnika i studenata u aplikaciji. **PUT** metoda **update_profile** prima podatke o ažuriranju profila u obliku **UpdateProfileRequest** sheme i mijenja postojeći objekt korisnika u bazi podataka. **DELETE** metoda **delete_user** prima parametar **user_id** i briše objekt korisnika iz baze podataka.
- **GradingController**: najbitniji **kontroler** koji sadrži metode koje se odnose na istorazinsko ocjenjivanje u aplikaciji. Ima 2 različite REST metode: **GET** i **PUT**. **GET** metode odnose se na dohvat rješenja za ocjenjivanje, rezultata ocjenjivanja i statusa ocjenjivanja za određeni zadatak ili rješenje. **PUT** metoda se odnosi na ocjenjivanje rješenja za određeni zadatak, a prima listu zahtjeva za ocjenjivanje kao parametar. Zahtjevi za ocjenjivanje su **GradeSubmissionRequest** sheme koje sadrže identifikator rješenja i ocjenu koju je student dao tom rješenju.

3.1.3. Algoritam za odabiranje radova za ocjenjivanje

U ovom radu koristi se algoritam za odabiranje radova koje student treba ocjenjivati u okviru nekog zadatka na kolegiju. Algoritam se temelji na sljedećim pretpostavkama i ciljevima:

- svaki zadatak ima definirano ocjenjivanje, koje sadrži upute za ocjenjivanje, broj radova koje student treba ocijeniti (N) i status ocjenjivanja (započeto, na čekanju ili završeno)
- svaki student mora predati svoje rješenje zadatka prije nego što može pristupiti ocjenjivanju tuđih radova
- svaki student treba ocijeniti N radova drugih studenata za svaki zadatak, osim ako ukupan broj radova nije manji od N, u tom slučaju treba ocijeniti sve radove
- algoritam treba biti pravedan i uravnotežen, tako da nijedan rad ne bude zanemaren ili preocijenjen

Algoritam se sastoji od sljedećih koraka:

1. Provjeriti ima li student predano rješenje zadatka. Ako nema, vratiti poruku o grešci i onemogućiti pristup ocjenjivanju.
2. Provjeriti ima li student već dodijeljene radove za ocjenjivanje. Ako ima, vratiti te radove i njihove statuse (u tijeku ili završeno).
3. Dohvatiti sve radove za zadatak, isključujući rješenje studenta koji ocjenjuje.
4. Provjeriti je li broj radova manji ili jednak N. Ako jest, odabrati sve radove kao one koje student treba ocijeniti.
5. Ako nije, sortirati radove po broju dobivenih ocjena i uvesti slučajni poredak za one koji imaju jednak broj ocjena.
6. Odabrati prvih N radova iz sortirane liste kao one koje student treba ocijeniti.
7. Za svaki odabrani rad stvoriti objekt SubmissionGrade, koji sadrži podatke o ocjenjivaču, predaji, ocjeni, datumu i statusu.
8. Vratiti listu odabranih radova.

3.1.4. Sigurnost i autorizacija temeljene na JWT tokenu

JWT (engl. *JSON Web Token*) je standardni način za autentifikaciju i autorizaciju korisnika i zaštitu API ruta. JWT je kompaktan i samodostatan način za sigurno prenošenje informacija između dvije strane kao JSON objekta. Ove informacije se mogu provjeriti i može se pouzdati u njih jer su digitalno potpisane. JWT se može potpisati koristeći tajnu (s HMAC algoritmom) ili javni/privatni ključ (s RSA ili ECDSA algoritmom). Potpis garantira da je izvor podataka onaj koji tvrdi da jest i da se podaci nisu promijenili u međuvremenu.

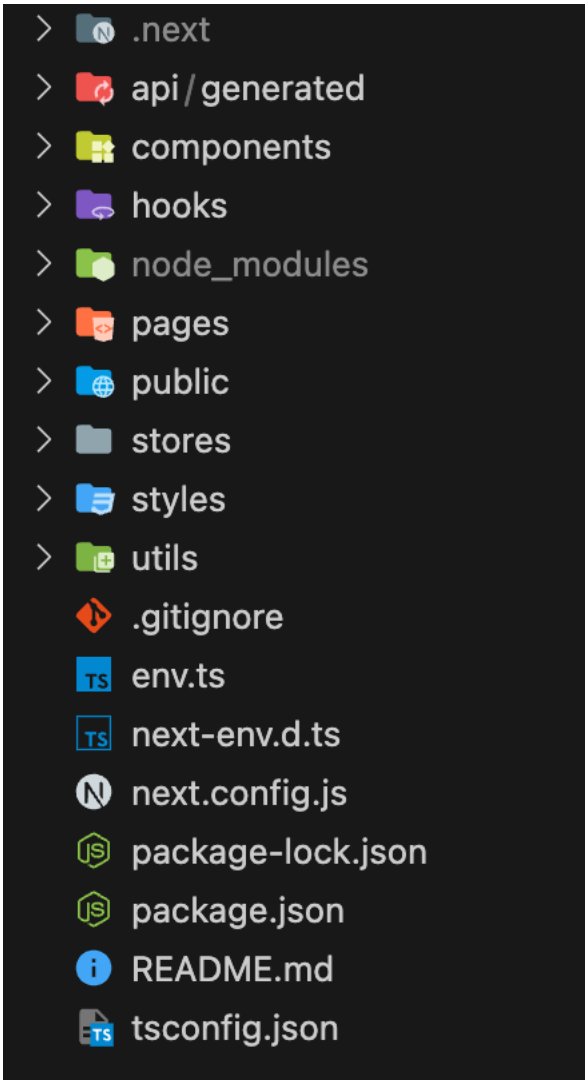
JWT se najčešće koristi za autorizaciju korisnika nakon što se prijave u aplikaciju. Svaki sljedeći zahtjev će uključivati JWT, koji omogućava korisniku pristup rutama, servisima i resursima koji su dozvoljeni s tim tokenom. Također, JWT se može koristiti za razmjenu informacija između klijenta i poslužitelja, kao što su identitet korisnika, njegove ovlasti, podaci o sesiji i druge korisne informacije.

JWT se sastoji od tri dijela odvojena točkama (.): zaglavlja, tereta i potpisa. Zaglavlje sadrži informacije o tipu tokena i algoritmu potpisivanja. Teret sadrži tvrdnje (engl. *claims*), koje su izjave o entitetu (obično korisniku) i dodatnim podacima. Potpis se računa na temelju zaglavlja, tereta i tajne ili privatnog ključa. Svaki dio je kodiran u Base64Url formatu, koji je URL-siguran način prikazivanja binarnih podataka. [23]

3.2. Klijentski dio aplikacije

3.2.1. Arhitektura

Klijentski dio aplikacije pisan je u TypeScript programskom jeziku, koristeći Next.js razvojni okvir, čija je „baza“ React. Za ljepši izgled aplikacije korištena je biblioteka Bootstrap. Svi pozivi kojima se vrše CRUD (engl. *CREATE, READ, UPDATE, DELETE*) operacije nad podacima oslanjaju se na Axios biblioteku. Sama struktura direktorija klijentskog dijela aplikacije prati smjernice koje su postavljene od strane Next.js zajednice i prikazana je na Slici 3.2.



Slika 3.2: Prikaz strukture foldera klijentskog dijela aplikacije

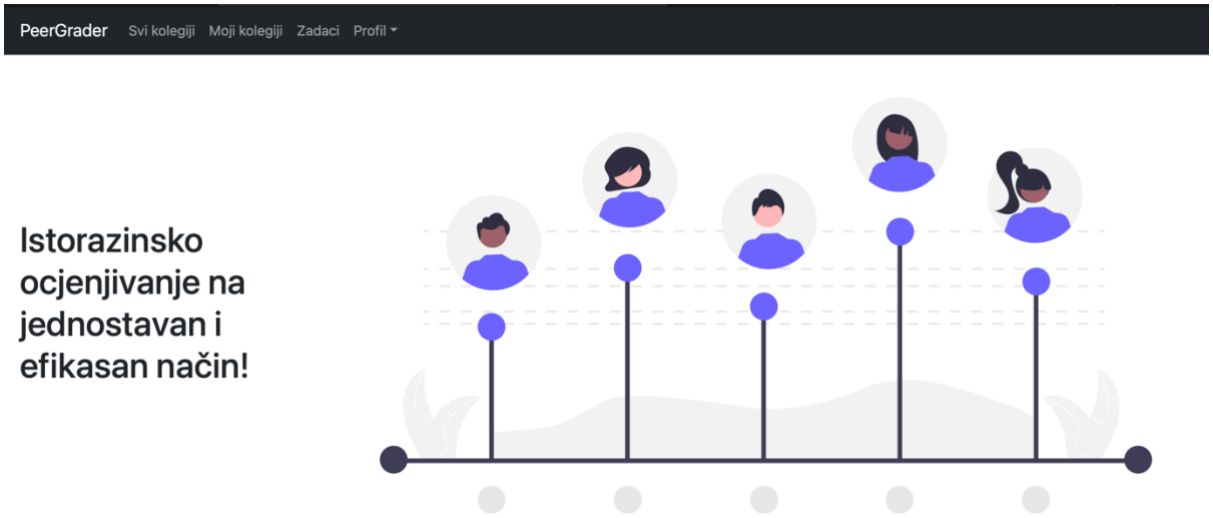
Unutar direktorija *pages* nalaze se sve stranice koje se prikazuju na klijentskom dijelu. Next.js sadrži tehniku usmjeravanja (engl. *routing)* te samim time jednostavno se mogu kreirati putanje na web stranici. Primjerice, datoteka index.tsx unutar direktorija pages označava da je to početna stranica aplikacije, kojoj se može pristupiti putem poveznice http://localhost:3000 (u slučaju da se lokalno pristupa aplikaciji).

3.2.2. Sučelje

Kako je prije spomenuto, postoje tri različite vrste korisnika: administratori, nastavnici i studenti. Sukladno tome, prikazuju se i različite mogućnosti unutar sučelja. Da bi to bilo moguće, radi se diferencijacija na temelju dohvaćenih podataka s poslužitelja.

Svaka stranica aplikacije sadrži navigacijsku traku i sadržaj, koji se mijenja ovisno o tome čemu se želi pristupiti i tko tome želi pristupiti.

Navigacijska traka sadrži akcije koje su dostupne korisniku, neovisno je li prijavljen ili ne. Kada se korisnik prijavi, navigacijska traka mijenja dostupne akcije te na Slici 3.3 može se vidjeti prikaz naslovne stranice zajedno s navigacijskom trakom na vrhu za prijavljenog nastavnika.

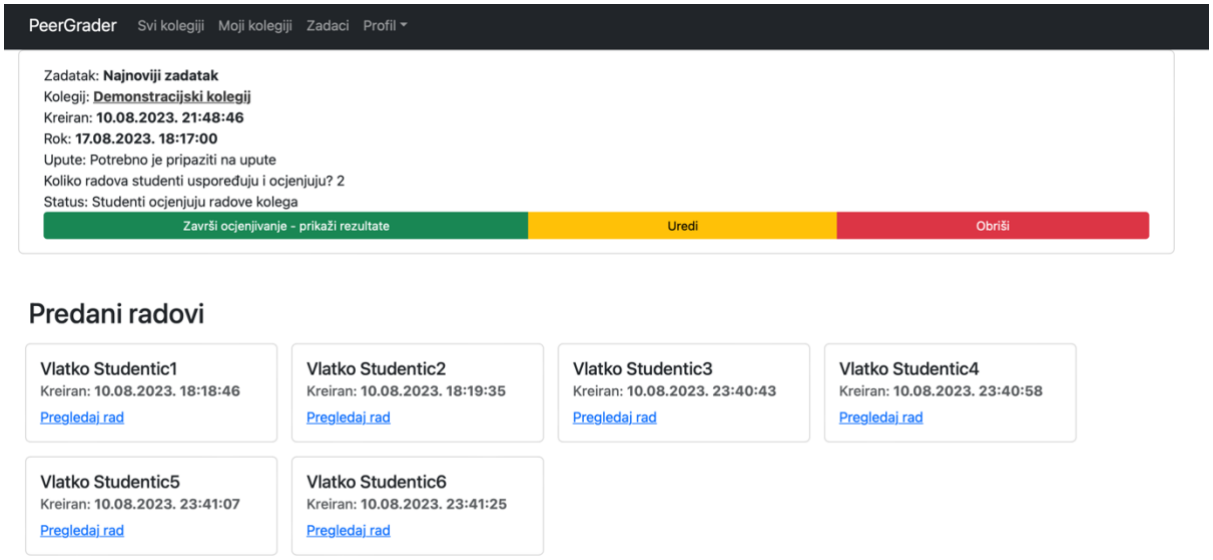


Slika 3.3: Prikaz naslovne stranice s navigacijskom trakom

Uz navigacijsku traku koja je uvijek na vrhu, na Slici 3.3 vidljiva je i naslovna stranica koja slikovito prikazuje svrhu aplikacije.

Na sličan način su raspoređene i ostale stranice aplikacije, uz napomenu da je većina stranica zaštićena te da samo prijavljeni i autorizirani korisnici mogu pristupiti istim.

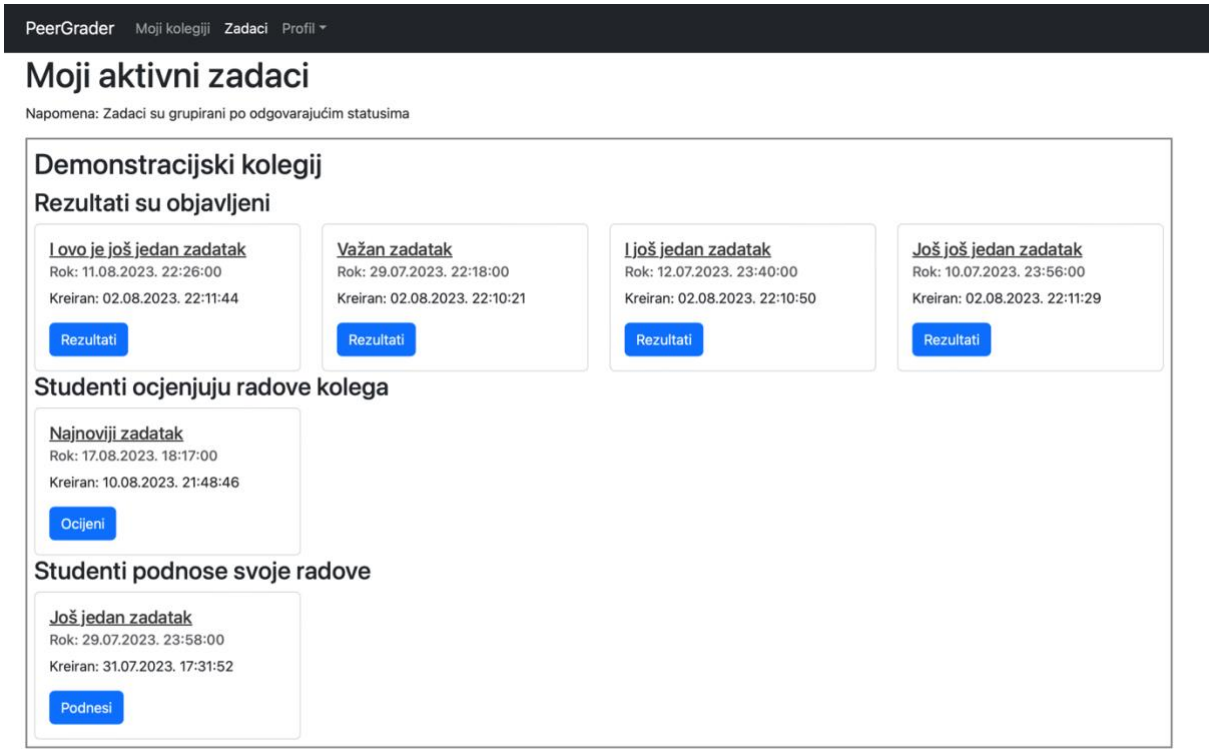
Prijavljeni nastavnici pomoću sučelja mogu raditi CRUD operacije nad kolegijima i zadacima na kolegijima čiji su oni nositelji, kao i nad vlastitim korisničkim profilom. Isto tako, mogu mijenjati statuse zadataka te pregledavati studente i njihove radove kao i rezultate ocjenjivanja za neki zadatak, zajedno s ocjenama koje su dodijeljene pojedinom radu. Administratori se u aplikaciji tretiraju kao nastavnici s mogućnostima nad svim kolegijima. Na Slici 3.4 vidljiv je prikaz zadatka i predanih rješenja studenata.



Slika 3.4: Prikaz zadatka i predanih radova studenata

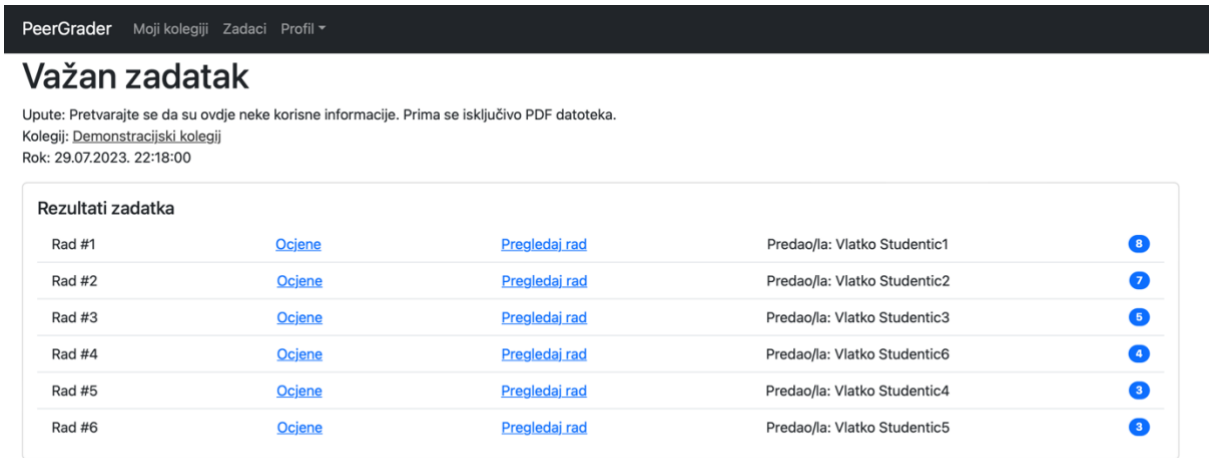
Kao što je vidljivo iz slike, uz popis predanih radova, nastavnik može urediti, obrisati ili promijeniti status zadatka.

Studenti mogu raditi CRUD operacije nad svojim profilom, vidjeti kolegije u koje su upisani te zadatke povezane s njima. Isto tako, mogu predati rješenja za zadatke, kao i ažurirati već predana rješenja. Ključna mogućnost studenata je ocjenjivanje radova koje su predali kolege u sklopu nekog zadatka. Nakon što nastavnik odredi da je kraj faze ocjenjivanja, studenti mogu vidjeti rezultate ocjenjivanja. Na Slici 3.5 prikazan je izgled sučelja gdje se nalaze aktivni zadaci studenta.



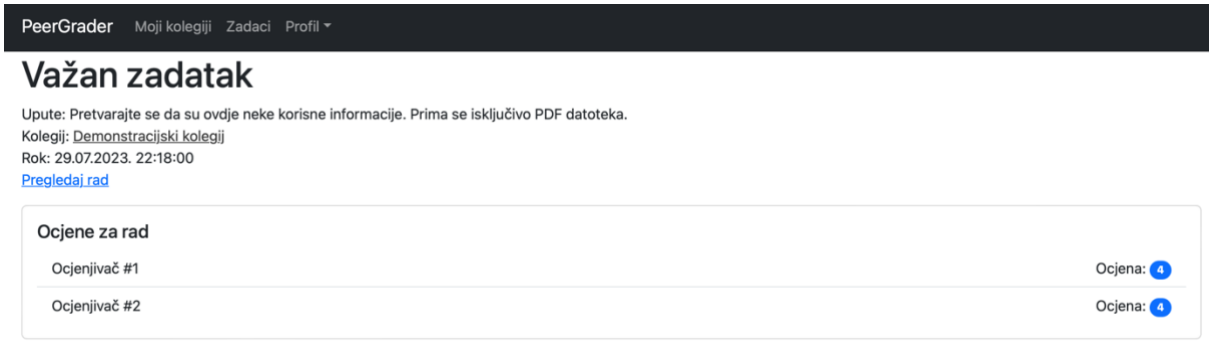
Slika 3.5: Prikaz aktivnih zadataka studenta

Budući da postoje tri različita statusa svakog zadatka, tako postoje i 3 različite akcijske tipke: *Rezultati*, *Ocijeni* i *Podnesi*. Rezultati pružaju uvid u ukupne „ocjene“ studenata za njihove radove. Na Slici 3.6 koja slijedi nalazi se primjer jednog takvog zadatka.



Slika 3.6: Prikaz rezultata zadatka

Na slici se može se vidjeti kako su radovi rangirani na temelju ukupne sume ocjena koje je svaki rad dobio, gdje veća suma predstavlja bolje plasiran rad. Odabirom poveznice Ocjene otvara se ekran gdje studenti mogu vidjeti koje su ocjene dobivene za koji rad i prikazan je u nastavku na Slici 3.7.



Slika 3.7: Prikaz ocjena za pojedini rad

Ono što se dodatno može vidjeti ako se korisnik prijavi kao administrator ili nastavnik jest i tko je tu ocjenu dao.

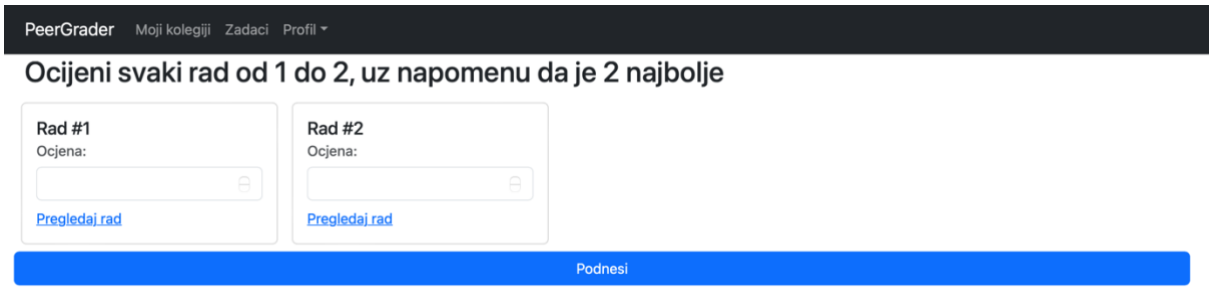
3.2.3. Ocjenjivanje

Početni stadij koji neki zadatak ima jest prijenos radova od strane studenata. Bitno je da nastavnik postavi rok u kojem studenti imaju dovoljno vremena da postave svoja rješenja. U dogovoru sa studentima, nastavnik mijenja status zadatka i zadatak prelazi u status ocjenjivanja gdje svaki student dobije na ocjenjivanje određen broj radova koje treba ocijeniti. Ocjenjivanje nije klasično u smislu od 1 do 5, već od 1 do N, gdje N predstavlja broj radova koji se dodjeljuju svakom studentu na ocjenjivanje. N je najveća „ocjena“ koja se može dati za neki rad. Student ne može ocijeniti više od jednog rada s istom ocjenom, budući da suma svih ocjena mora zadovoljavati sumu N prirodnih brojeva koristeći formulu: $očekivana\ suma = \frac{N\,(\mathrm{N}+1)}{2}$

gdje je:

- N – prirodni broj koji predstavlja broj radova koji se dodjeljuje studentu na ocjenjivanje

U nastavku je prikazano ocjenjivačko sučelje gdje studenti ocjenjuju radove od 1 do N.



Slika 3.8: Prikaz ocjenjivačkog sučelja

U ovom slučaju taj N, odnosno broj radova koje studenti uspoređuju i ocjenjuju iznosi dva. Studenti imaju mogućnost pregledati svaki rad i na temelju toga dati svoju ocjenu. Nakon što student ocijeni radove, ne može ponovno otići na ekran za ocjenjivanje tog zadatka te treba pričekati da nastavnik promijeni status istog.

4. ZAKLJUČAK

Istorazinsko ocjenjivanje je metoda koja ima brojne prednosti za studente i nastavnike, kao što su poboljšanje kvalitete povratne informacije, poticanje kritičkog mišljenja i suradnje, smanjenje opterećenja nastavnika i povećanje motivacije studenata. Međutim, istorazinsko ocjenjivanje također ima neke izazove i ograničenja u koje spadaju mogućnost nepoštenog ili nekompetentnog ocjenjivanja, nedostatak povjerenja u ocjene kolega, potreba za dodatnim vremenom i naporom za provedbu te dodatna potreba za ispravnim praćenjem procesa ocjenjivanja. Iznimno je važno da se istorazinsko ocjenjivanje provodi uz adekvatnu pripremu, podršku i nadzor od strane nastavnika. Može se reći kako je istorazinsko ocjenjivanje koncept koji je sve većim razvojem web tehnologija postao pristupačniji široj publici.

Web aplikacija za studentsko istorazinsko ocjenjivanje jedna je od brojnih aplikacija koje postoje na tržištu u ovu svrhu. U radu su uspoređene neke od najpoznatijih aplikacija za istorazinsko ocjenjivanje, kao što su ComPAIR, CrowdGrader, PeerStudio i druge. Svaka od ovih aplikacija ima svoje prednosti i nedostatke u pogledu dizajna, funkcionalnosti, korisničkog iskustva i učinkovitosti. Pregledom i usporedbom već postojećih alata za istorazinsko ocjenjivanje upućuje se na to da je potražnja za takvim načinom poučavanja i vrednovanja u porastu. Web aplikacija za studentsko istorazinsko ocjenjivanje predstavljena u ovom radu ističe se po svojoj jednostavnosti i fleksibilnosti.

Ovim radom predstavljena je web aplikacija za istorazinsko ocjenjivanje koja omogućava nastavnicima da kreiraju i upravljaju kolegijima i zadacima kao i načinom ocjenjivanja zadataka. Studentima je omogućeno da predaju i ocjenjuju radove svojih kolega. Nastavnici, administratori i studenti mogu vidjeti rezultate ocjenjivanja. Zadatak završnog rada bio je napraviti web aplikaciju za istorazinsko ocjenjivanje koja je uspješno implementirana korištenjem modernih web tehnologija kao što su Django, Next.js, TypeScript i Bootstrap.

Za budući razvoj web aplikacije za studentsko istorazinsko ocjenjivanje se predlažu neke moguće nadogradnje i poboljšanja. Neki od prijedloga su: omogućavanje komunikacije između studenata i nastavnika putem web aplikacije, davanje povratne informacije od strane studenata za pojedini rad, integracija web aplikacije s drugim obrazovnim platformama ili sustavima za upravljanje učenjem (engl. *learning management systems*), primjena naprednih algoritama ili

metoda strojnog učenja za analizu kvalitete radova i ocjena te testiranje web aplikacije na većem broju kolegija i studenata.

Web aplikacija za istorazinsko ocjenjivanje je koristan alat koji olakšava provedbu istorazinskog ocjenjivanja u različitim obrazovnim kontekstima. Istorazinsko ocjenjivanje je metoda koja ima potencijal da unaprijedi obrazovni proces i ishode za studente i nastavnike.

LITERATURA

[1] P. M. Sadler i E. Good, „The Impact of Self- and Peer-Grading on Student Learning“, *Educ. Assess.*, sv. 11, izd. 1, str. 1–31, 2006, doi: 10.1207/s15326977ea1101_1.

[2] K. Topping, *Self and Peer Assessment in School and University: Reliability, Validity and Utility*.

[3] N. Falchikov i D. Boud, „Student Self-Assessment in Higher Education: A Meta-Analysis“, *Rev. Educ. Res.*, sv. 59, izd. 4, str. 395–430, pros. 1989, doi: 10.3102/00346543059004395.

[4] M. M. Yalch, E. M. Vitale, i J. Kevin Ford, „Benefits of Peer Review on Students' Writing“, *Psychol. Learn. Teach.*, sv. 18, izd. 3, str. 317–325, stu. 2019, doi: 10.1177/1475725719835070.

[5] „Peergrade - engaging student peer review“. https://www.peergrade.io/ (pristupljeno 28. lipanj 2023.).

[6] „Peer Review Tool | FeedbackFruits“. https://feedbackfruits.com/peer-review# (pristupljeno 28. lipanj 2023.).

[7] „PeerStudio: the peer feedback and assessment tool that helps students learn“. https://www.peerstudio.org/ (pristupljeno 28. lipanj 2023.).

[8] „CrowdGrader: Peer Grading of Homework“. https://doc.crowdgrader.org/ (pristupljeno 28. lipanj 2023.).

[9] „Home | ComPAIR“. https://compair.open.ubc.ca/ (pristupljeno 28. lipanj 2023.).

[10] „FeedbackFruits | Digital Teaching Tools for Higher Ed“. https://feedbackfruits.com/ (pristupljeno 28. lipanj 2023.).

[11] „Welcome to Python.org“, *Python.org*, 06. kolovoz 2023. https://www.python.org/ (pristupljeno 11. kolovoz 2023.).

[12] „Django“, *Django Project*. https://www.djangoproject.com/ (pristupljeno 10. kolovoz 2023.).

[13] „Django Ninja“. https://django-ninja.rest-framework.com/ (pristupljeno 10. kolovoz 2023.).

[14] „JavaScript With Syntax For Types.“ https://www.typescriptlang.org/ (pristupljeno 10. kolovoz 2023.).

[15] „Next.js by Vercel - The React Framework“. https://nextjs.org/ (pristupljeno 10. kolovoz 2023.).

[16] „React“. https://react.dev/ (pristupljeno 10. kolovoz 2023.).

[17] M. O. contributors Jacob Thornton, and Bootstrap, „Bootstrap“. https://getbootstrap.com/ (pristupljeno 10. kolovoz 2023.).

[18] „What is a REST API? | IBM“. https://www.ibm.com/topics/rest-apis# (pristupljeno 10. kolovoz 2023.).

[19] „Welcome to Pydantic - Pydantic“. https://docs.pydantic.dev/latest/ (pristupljeno 11. kolovoz 2023.).

[20] „Home“, *OpenAPI Initiative*. https://www.openapis.org/ (pristupljeno 11. kolovoz 2023.).

[21] „API Documentation & Design Tools for Teams | Swagger“. https://swagger.io/ (pristupljeno 11. kolovoz 2023.).

[22] „Django Ninja Extra“. https://eadwincode.github.io/django-ninja-extra/ (pristupljeno 11. kolovoz 2023.).

[23] auth0.com, „JWT.IO - JSON Web Tokens Introduction“. http://jwt.io/ (pristupljeno 11. kolovoz 2023.).

SAŽETAK

Rad se bavi problematikom istorazinskog ocjenjivanja kao metode koja ima brojne prednosti i izazove za studente i nastavnike u obrazovnom procesu. Istorazinsko ocjenjivanje je metoda u kojoj studenti ocjenjuju radove svojih kolega prema unaprijed definiranim kriterijima i skalama. Cilj je poboljšati kvalitetu povratne informacije, potaknuti studente na razmišljanje i prosuđivanje o tuđem radu, produbiti suradnju među studentima te smanjiti opterećenje nastavnika. U radu se predstavlja i uspoređuje nekoliko web aplikacija za istorazinsko ocjenjivanje koje su dostupne na tržištu, kao što su ComPAIR, CrowdGrader, PeerStudio i druge. Svaka od ovih aplikacija ima svoje prednosti i nedostatke u pogledu dizajna, funkcionalnosti, korisničkog iskustva i učinkovitosti. U radu se također opisuje i implementira zadatak završnog rada, odnosno web aplikacija za istorazinsko ocjenjivanje koja je razvijena korištenjem modernih web tehnologija kao što su Django, Next.js, TypeScript i Bootstrap. Web aplikacija omogućava nastavnicima da kreiraju i upravljaju kolegijima i zadacima kao i načinom ocjenjivanja zadataka. Studentima je omogućeno da predaju i ocjenjuju radove svojih kolega. Nastavnici, administratori i studenti mogu vidjeti rezultate ocjenjivanja. Na kraju, u radu se daju neki prijedlozi za budući razvoj i poboljšanje web aplikacije. Web aplikacija za istorazinsko ocjenjivanje je koristan alat koji olakšava provedbu istorazinskog ocjenjivanja u različitim obrazovnim kontekstima, a sâmo istorazinsko ocjenjivanje je metoda koja ima potencijal da unaprijedi obrazovni proces i ishode za studente i nastavnike.

Ključne riječi: *istorazinsko ocjenjivanje, web aplikacija, Django, Next.js*

ABSTRACT

Web application for student peer grading

The paper deals with the issue of peer assessment as a method that has numerous advantages and challenges for students and teachers in the educational process. Peer assessment is a method in which students evaluate the works of their colleagues according to predefined criteria and scales. The goal is to improve the quality of feedback, encourage students to think and judge others' work, deepen cooperation among students, and reduce the workload of teachers. The paper presents and compares several web applications for peer assessment that are available on the market, such as ComPAIR, CrowdGrader, PeerStudio and others. Each of these apps has its pros and cons in terms of design, functionality, user experience, and efficiency. The paper also describes and implements the task of the final paper, i.e. a web application for peer assessment that was developed using modern web technologies such as Django, Next.js, TypeScript and Bootstrap. The web application allows teachers to create and manage courses and assignments, as well as how to grade assignments. Students are enabled to submit and evaluate the works of their colleagues. Teachers, administrators, and students can view assessment results. Finally, the paper gives some suggestions for the future development and improvement of the web application. A peer assessment web application is a useful tool that facilitates the implementation of peer assessment in various educational contexts, and peer assessment itself is a method that has the potential to improve the educational process and outcomes for students and teachers.

Keywords: *peer grading, web application, Django, Next.js*