

# WEB APLIKACIJA ZA PRAĆENJE RADA NA PROJEKTIMA U OKVIRU NASTAVNIH KOLEGIJA

---

**Biskupić, Luka**

**Undergraduate thesis / Završni rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:190:614411>

*Rights / Prava:* [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

*Download date / Datum preuzimanja:* **2025-03-14**



*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI  
TEHNIČKI FAKULTET

Prijediplomski sveučilišni studij računarstva

Završni rad

**WEB APLIKACIJA ZA PRAĆENJE RADA NA PROJEKTIMA U  
OKVIRU NASTAVNIH KOLEGIJA**

Rijeka, rujan 2024.

Luka Biskupić

0069089082

SVEUČILIŠTE U RIJECI  
TEHNIČKI FAKULTET

Prijediplomski sveučilišni studij računarstva

Završni rad

**WEB APLIKACIJA ZA PRAĆENJE RADA NA PROJEKTIMA U  
OKVIRU NASTAVNIH KOLEGIJA**

Mentor: Izv. prof. dr. sc. Marko Gulić

Rijeka, rujan 2024.

Luka Biskupić

0069089082

**SVEUČILIŠTE U RIJECI**  
**TEHNIČKI FAKULTET**  
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 8. ožujka 2024.

Zavod: **Zavod za računarstvo**  
Predmet: **Razvoj web aplikacija**  
Grana: **2.09.06 programsko inženjerstvo**

## ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Luka Biskupić (0069089082)**  
Studij: Sveučilišni prijediplomski studij računarstva

Zadatak: **Web aplikacija za praćenje rada na projektima u okviru nastavnih kolegija**

Opis zadatka:

Razviti web aplikaciju za praćenje rada na projektima u okviru nastavnih kolegija. Aplikacija će omogućiti dodjeljivanje, ažuriranje i brisanje projekata od strane profesora, prijavu studenata na projekt te dodjeljivanje voditelja (studenta) svakom projektu. Potrebno je podržati predaju projektne dokumentacije te izradu obrazaca za vrednovanje, kao i samo provođenje vrednovanja i samovrednovanja. Za razvoj poslužiteljskog dijela web aplikacije treba koristiti Laravel radni okvir uz proizvoljno odabran sustav za upravljanje bazama podataka. Također, za realizaciju autentifikacije s klijentskom stranom, koristiti AAI@EduHr infrastrukturu.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 22. ožujka 2024.

Mentor:

Predsjednik povjerenstva za  
završni ispit:

---

doc. dr. sc. Marko Gulić

---

prof. dr. sc. Miroslav Joler

---

prof. dr. sc. Mladen Tomić (komentor)

## **Izjava o samostalnoj izradi rada**

Izjavljujem da sam samostalno izradio ovaj rad

Rijeka, rujan 2024.

-----

Luka Biskupić



# SADRŽAJ

<b>1 UVOD</b>	<b>1</b>
<b>2 TEHNOLOGIJE</b>	<b>3</b>
2.1 HTML	3
2.2 CSS	3
2.3 Blade	3
2.4 Laravel	4
2.4.1 Laravel Pint	5
2.4.2 Spatie uloge i dozvole	5
2.4.3 Eloquent	6
2.5 Sustav za upravljanje bazama podataka	7
2.6 Filament	7
2.6.1 Filament panel	7
2.6.2 Filament resurs	8
2.6.3 Filament obrasci	8
2.6.4 Filament tablice	8
2.6.5 Filament widgets	9
2.6.6 Filament plugins	10
2.6.7 Filament zajednica i podrška	10
2.7 Docker	11
2.7.1 Docker compose	12
2.7.2 Laravel Sail	13
2.8 Git	13
2.8.1 Github	13
<b>3 APLIKACIJA ZA PRAĆENJE RADA NA PROJEKTIMA</b>	<b>14</b>
3.1 Student	14
3.2 Voditelj kolegija	24
3.3 Super admin	39
<b>4 RAZVOJ I IMPLEMENTACIJA SPECIFIČNIH APLIKACIJSKIH FUNKCIONALNOSTI</b>	<b>41</b>
4.1 Dodavanje studenta na projekt	41
4.2 Prijenos csv datoteke sa podacima studenata	46
<b>5 ZAKLJUČAK</b>	<b>51</b>
<b>LITERATURA</b>	<b>52</b>
<b>SAŽETAK</b>	<b>55</b>

# 1 UVOD

U ovom radu napravljena je web aplikacija za praćenje rada na projektima u okviru nastavnih kolegija. Aplikacija omogućuje dodjeljivanje, ažuriranje i brisanje projekata od strane profesora, prijavu studenata na projekt i označavanje voditelja svakog projekta. Aplikacija omogućuje predaju projektne dokumentacije. Predaja projektne dokumentacije podrazumijeva predaju pdf dokumenta timske deklaracije, dokumenta aktivnosti te dokumenta u kojem su zapisani detalji sastanaka. Aplikacija također omogućuje izradu obrazaca za vrednovanje. Obrasci za vrednovanje omogućuju studentima da ocjenjuju ponašanje i aktivnosti drugih studenata s kojima rade na projektu. Obrasci za samovrednovanje omogućuju studentu da ocjeni svoj doprinos projektu.

Za razvoj poslužiteljskog dijela web aplikacije korišten je Laraven radni okvir [1]. Laravel je popularan PHP radni okvir koji se koristi za izradu web aplikacija. Pruža elegantnu sintaksu koja pojednostavljuje zadatke kao što su autentifikacija, rutiranje i validacija. Laravel koristi MVC arhitekturu koja omogućuje jasnu organizaciju koda i olakšava održavanje aplikacija. Na klijentskoj strani su korištene sljedeće tehnologije: HTML, CSS, Javascript i Blade [2]. Blade sustav za rad s predlošcima je jednostavan, ali moćan alat za izradu dinamičkih pogleda (Views) [3] u Laravelu. Omogućuje korištenje lagane i intuitivne sintakse za ugnježđivanje PHP koda unutar HTML-a. Za izradu ove aplikacije najviše je korišten Filament paket [4]. Filament je paket za Laravel koji omogućuje brzu i jednostavnu izradu administratorskih panela i kontrolnih ploča za web aplikacije. Njegova glavna svrha je pojednostaviti i ubrzati razvoj složenih backend sustava.

Rad se sastoji od sljedećih poglavlja: u poglavlju 2 opisane su sve tehnologije koje su korištene. Opisane su tehnologije koje su korištene na klijentskoj i poslužiteljskoj strani, sustav za upravljanje bazom podataka kao i zasebne biblioteke koje su korištene za izgradnju ove aplikacije. U poglavlju 3 objašnjene su funkcionalnosti aplikacije te je detaljno objašnjeno kako koristiti aplikaciju iz perspektive studenta i iz perspektive profesora odnosno voditelja kolegija. U poglavlju 4 objašnjene su funkcionalnosti aplikacije ali na razini koda. U tom poglavlju je objašnjeno kako se kroz korištenje aplikacije šalje zahtjev na poslužitelja, kako poslužitelj obrađuje taj zahtjev, kako pristupa bazi podataka te kako se vraća odgovor klijentu. Cijeli postupak neke funkcionalnosti bit će objašnjen kroz kod. Zaključak je iznesen u posljednjem poglavlju.



## 2 TEHNOLOGIJE

U ovom poglavlju se nalazi opis svake tehnologije koja je korištena u razvoju aplikacije. Redom će biti objašnjene tehnologije korištene za klijentski dio, poslužiteljski dio, sustav za upravljanje bazom podataka, Filament paket, Docker [5] i Git [6].

### 2.1 HTML

HTML [1] (HyperText Markup Language) osnovni je jezik za izradu web stranica. Kao standardni označni jezik, HTML omogućuje web dizajnerima da strukturiraju sadržaj na web stranici koristeći različite elemente kao što su naslovi, paragrafi, slike i linkovi. Svaki HTML dokument sastoji se od niza elemenata, označenih oznakama (tagovima), koje preglednici koriste kako bi prikazali stranice korisnicima na razumljiv i organiziran način. Njegov razvoj nadzire World Wide Web Consortium [8] (W3C) kako bi se osiguralo da standardi ostaju relevantni i usklađeni s tehnološkim napretkom.

### 2.2 CSS

CSS [9] (Cascading Style Sheets) je stilski jezik koji se koristi za definiranje vizualnog prikaza i dizajna web stranica. CSS omogućuje web dizajnerima i programerima da odvoje sadržaj stranice (npr. pisani u HTML-u) od njegovog stila, što uključuje boje, fontove, raspored i druge vizualne aspekte. CSS standardi razvijaju se i održavaju od strane World Wide Web Consortiuma (W3C), osiguravajući tako njihovu konzistentnost i kompatibilnost s različitim uređajima i preglednicima.

### 2.3 Blade

Blade je sustav za upravljanje predlošcima (eng. templating engine). Blade dolazi integriran unutar Laravela, popularnog PHP web okvira. Ovaj sustav predložaka omogućuje jednostavno kreiranje korisničkog sučelja kroz nasljeđivanje i komponiranje predložaka, što pridonosi efikasnijem i organiziranijem razvoju aplikacija. Blade predlošci se mogu sastojati od običnog PHP koda, ali mogu sadržavati i specifične Blade direktive [10] koje su čitljive i koje olakšavaju ispis podataka i kontrolu toka podataka između predložaka.

Jedna od prednosti korištenja Blade sustava je čista sintaksa. Blade omogućuje korištenje kontrolnih struktura kao što su uvjeti i petlje. Korištenje tih struktura je intuitivno i elegantnije od korištenja PHP oznaka. Te direktive odnosno strukture se kasnije prevedu u PHP kod. Korištenje Blade sustava značajno ubrzava razvoj aplikacije zbog njegove integracije s Laravelom. Unutar Laravel pogleda [11], u kojima se koristi Blade sustav, lako se može koristiti i pristupiti funkcionalnostima kao što su sesije [12], *cache* [13] i autentikacija. Nasljeđivanje predložaka također je jedna od prednosti korištenja ovog sustava. Nasljeđivanje predložaka omogućava da se stvori osnovni predložak koji definira zajednički izgled dok se specifični dijelovi mijenjaju odvojeno koristeći sekcijske direktive [14]. Bitno je istaknuti da Blade omogućuje laku implementaciju obrane od *CSRF* (Cross-Site Request Forgery) napada [15].

*CSRF* napad je vrsta sigurnosne prijetnje u kojoj napadač prisiljava korisnika da izvrši neželjene akcije na web stranici na kojoj je trenutno autentificiran. Tipično, napadač koristi hiperlink ili skriptu koja automatski šalje zahtjev web aplikaciji koristeći korisnikove autentifikacijske podatke, kao što su kolačići sesije, bez korisnikova znanja ili odobrenja. Ovaj zahtjev može uključivati razne akcije poput promjene email adrese, lozinke, ili čak iniciranja transakcija, što može dovesti do ozbiljnih sigurnosnih problema.

Da bi se obranila od *CSRF* napada, web aplikacija mora implementirati mehanizme koji provjeravaju svaki zahtjev kako bi potvrdili da nije generiran od strane treće strane. Jedna od uobičajenih metoda zaštite je korištenje tokena koji se generira na strani poslužitelja i šalje se kao dio forme ili URL-a na klijentsku stranu. Kada korisnik pošalje formu, aplikacija provjerava da li se poslani token podudara s onim koji je generiran na poslužitelju, čime se osigurava da je zahtjev legitimno poslan od strane autentičnog korisnika. U kontekstu Lavela, zaštita od *CSRF* napada ostvaruje se korištenjem Blade direktive *csrf*, koja automatski ubacuje *CSRF* token u formu kao skriveno polje. Ovo osigurava da svaki *POST* zahtjev koji dolazi iz forme sadrži valjani token, čime se efikasno sprečavaju *CSRF* napadi. Implementacija ove zaštitne mjere je vitalna za sigurnost bilo koje web aplikacije koja omogućava korisnicima izvršavanje akcija koje mijenjaju stanje na poslužitelju, čime Laravel pomaže u održavanju visokog nivoa sigurnosti bez kompliciranog procesa razvoja.

## 2.4 Laravel

Laravel je moderni PHP okvir za izradu web aplikacija. Dizajniran je s ciljem pojednostavljenja razvojnog procesa pružanjem niza pomoćnih funkcija i klasa koje ubrzavaju izradu kompleksnih aplikacija. Laravel nudi ugrađenu podršku za rutiranje [18], autentikaciju, sesije, *cache* i mnoge druge ključne aspekte web razvoja. Prednost korištenja Laravela za razvoj aplikacija je njegova velika i aktivna zajednica. Velika i aktivna zajednica omogućava lak i brz pronalazak rješenja problema s kojima se programer susreće tijekom razvoja aplikacija. Isto tako, zbog velike i aktivne zajednice postoji veliki broj dostupnih obrazovnih resursa i paketa trećih strana. Zahvaljujući svojoj MVC arhitekturi, Laravel omogućava programerima da održe jasan odnos između logike aplikacije i njezine prezentacije, čineći kod lako održivim i nadogradivim. Laravel implementira veliki broj različitih obrazaca dizajna (eng. design patterns) [18], što omogućuje programerima da pišu organiziraniji i modularniji kod. Neki od njih su: kontejner usluga (eng. service container) [19], fasadni obrazac [20] (facade pattern), promatrački obrazac [21] (observer pattern) itd.

#### **2.4.1 Laravel Pint**

Laravel Pint [22] je alat za automatizirano formatiranje koda, koji je automatski instaliran sa svim novim laravel aplikacijama. Korištenje Laravel Pinta može značajno poboljšati kvalitetu koda tako što uniformno primjenjuje stilsko formatiranje na cijeli *codebase*, osiguravajući da sav kod slijedi iste stilističke smjernice. To ne samo da poboljšava čitljivost i održivost koda, već također olakšava suradnju među programerima na projektu, smanjujući mogućnost sintaktičkih grešaka i nekonzistentnosti.

#### **2.4.2 Spatie uloge i dozvole**

Spatie je razvojni tim iz Belgije koji je poznat po stvaranju kvalitetnih paketa za Laravel, koji su široko prihvaćeni u Laravel zajednici. Njihovi paketi omogućujući programerima da brzo implementiraju složene funkcionalnosti bez potrebe za pisanjem koda od nule. Jedan od tih paket je i *laravel-permission* [34]. Ovaj paket omogućuje jednostavno upravljanje ulogama (roles) i dozvolama (permissions) unutar Laravel aplikacija. Koristeći ovaj paket, programeri mogu lako dodijeliti dozvole direktno korisnicima ili dodijeliti direktno neku ulogu koja podrazumijeva skup određenih dozvola. Paket dolazi s nizom ugrađenih metoda koje olakšavaju provjeru dozvola i

implementaciju složenih pravila pristupa, što uvelike pojednostavljuje upravljanje sigurnošću aplikacije.

### 2.4.3 Eloquent

Laravel Eloquent ORM (Object-Relational Mapping) je napredan alat unutar Laravel okvira koji omogućava jednostavnu i intuitivnu interakciju s bazama podataka. Korištenjem Eloquent, programeri mogu izvršavati upite na bazu podataka koristeći PHP sintaksu bez potrebe za ručnim pisanjem SQL koda. Ovaj pristup ne samo da ubrzava proces razvoja već i smanjuje mogućnost grešaka koje su česte pri direktnom manipuliranju SQL upitima.

Eloquent koristi model kao glavni način za interakciju s bazom podataka. Svaki model u Laravelu odgovara tablici u bazi podataka, a atributi modela odgovaraju stupcima u tablici. Na primjer, ako imamo tablicu *users* u bazi podataka, kreirat ćemo model *User* u Laravelu. Ovaj model će automatski naslijediti sve funkcionalnosti Eloquent ORM-a, omogućujući manipulaciju podacima korisnika.

Jedna od ključnih prednosti Eloquent je njegova sposobnost da automatizira većinu CRUD (Create, Read, Update, Delete) operacija. Na primjer, stvaranje novog korisnika u bazi podataka može se izvesti stvaranjem nove instance *User* modela, postavljanjem atributa i pozivanjem metode *save* na toj instanci. Eloquent će se pobrinuti za generiranje odgovarajućeg SQL upita koji će novi zapis spremiti u bazu.

Eloquent također podržava napredne funkcije kao što su odnosi među tablicama (eng. relationships). Ovo omogućava definiranje veza kao što su *hasOne*, *hasMany*, *belongsTo*, i *belongsToMany* između modela. Uz pomoć ovih odnosa, programeri mogu lako manipulirati povezanim podacima i izvršavati složene upite. Na primjer, ako model *User* ima mnogo modela *Post*, možemo jednostavno dohvatiti sve objave korisnika pristupom atributu *posts* modela *User*.

Eloquent pruža i mogućnost automatske validacije i konverzije podataka prilikom njihova dohvata ili spremanja, kao što su datumi ili serijski podaci. Ovo značajno olakšava rad s podacima i povećava čitljivost koda. Korištenje Eloquent također donosi prednosti u smislu sigurnosti, posebno u kontekstu SQL injekcija. Eloquent automatski koristi pripremljene iskaze (eng. prepared statements), što čini aplikaciju otpornijom na ovu vrstu napada.

Laravel Eloquent ORM predstavlja snažan alat koji pojednostavljuje rad s bazama podataka u PHP aplikacijama. Svojom elegantnom sintaksom, bogatim setom funkcionalnosti i integracijom s Laravel frameworkom, Eloquent pomaže u brzom i sigurnom razvoju aplikacija, što ga čini preferiranim izborom za mnoge programere.

## 2.5 Sustav za upravljanje bazama podataka

MySQL [23] je popularni sustav za upravljanje bazama podataka i to je sustav koji je korišten za izgradnju aplikacije koja je opisana u ovom radu. MySQL nudi moćne opcije za pohranu, manipulaciju i dohvat podataka, što ga čini idealnim rješenjem za aplikacije koje zahtijevaju visoku dostupnost i skalabilnost. Podržava transakcije [24], što omogućava sigurno i pouzdano upravljanje podacima, kao i kompleksne upite koji poboljšavaju efikasnost pristupa podacima. Korištenje MySQL-a u Laravel aplikacijama omogućava brzu integraciju s Eloquent ORM-om [25], Laravelovim ugrađenim mehanizmom za rad s bazama podataka, što dodatno pojednostavljuje razvoj i održavanje aplikacija. Zahvaljujući svojoj snažnoj zajednici i širokoj podršci, MySQL ostaje jedan od vodećih izbora za razvojne timove širom svijeta.

## 2.6 Filament

Filament [26] je paket za Laravel koji programerima aplikacija omogućava brzu i jednostavnu izgradnju administrativnih sučelja. Filament je paket koji sadrži set alata, komponenti i integrira razne funkcionalnosti kao što su upravljanje formama, tablicama i resursima. Pruža robusnu platformu za učinkovito upravljanje podacima unutar Laravel aplikacije. Ovaj paket je veoma intuitivan i lako prilagodljiv te omogućava programerima da kreiraju složena korisnička sučelja s minimalnim kodiranjem. Još jedna od prednosti koju vrijedi spomenuti je da Filament pruža komponente s visokim standardom estetike. To omogućava programerima koji nisu vješti u dizajniranju vizualnih komponenti da stvore estetski privlačne i vizualno upečatljive aplikacije.

### 2.6.1 Filament panel

*Filament Panel* [27] je centralizirano mjesto aplikacije koje omogućava administratoru da upravlja sadržajem i funkcionalnostima aplikacije. Neke od glavnih značajki administratorskog sučelja su: upravljanje korisnicima, sadržajem, pregled statistika te upravljanje resursima.

Upravljanje korisnicima podrazumijeva dodavanje, brisanje i izmjenu korisničkih podataka, korisničkih profila te upravljanje njihovim ulogama i pravima pristupa ostalim resursima. Na primjer, u aplikaciji imamo korisnike s ulogama: *super-admin*, voditelj kolegija i student. *Super admin* ima najveće privilegije i može mijenjati sve unutar aplikacije, dok voditelj kolegija ima najveće privilegije samo unutar kolegija kojeg vodi. Student je uloga korisnika s koja ima najviše ograničenja. *Super admin* može upravljati s voditeljima kolegija i studentima, izmjenjivati njihove podatke, podatke kolegija i slično.

*Filament Widgets* su korisni za pregled statistika u administratorskom sučelju. *Stats overview widgets* [28] mogu se koristiti za prikaz različitih statistika u brojčanom obliku i bez potrebe za stvaranje i kodiranje prilagođenog pogleda. Stvaranje *filament widgeta* je uvelike olakšano *filament* komandama tj. *artisan* komandama [29]. Za vizualni prikaz može se koristiti *chart widget* [30].

### **2.6.2 Filament resurs**

Filament resurs je komponenta unutar Filament paketa za Laravel koja omogućava jednostavno upravljanje podacima modela kroz grafičko sučelje. Svaki resource u Filamentu povezan je s određenim Laravel modelom i služi kao most između modela i administrativnog sučelja, omogućujući krajnjim korisnicima da pregledavaju, uređuju, dodaju i brišu podatke u bazi. Resursi su dizajnirani da automatski prepoznaju strukturu povezanog modela, generirajući prikladne forme i tablice za interakciju s podacima.

### **2.6.3 Filament obrasci**

Filament obrasci su komponenta unutar Filament paketa, dizajnirana da omogućava izgradnju prilagodljivih obrazaca za unos i obradu podataka unutar web aplikacija. Programer aplikacije ima veliku kontrolu nad funkcionalnostima i izgledom obrasca. Obrasci mogu imati razne vrste polja, kao što su: tekstualna polja, izbornici, datumski odabiri i ostala. Svako polje može imati pridružena pravila za validaciju kako bi se osiguralo da uneseni podaci zadovoljavaju određene kriterije. Osim toga, forme podržavaju napredne funkcije kao što su kondicionalni prikazi, grupiranje podataka i prilagođavanje izgleda pomoću Tailwinda [31].

### **2.6.4 Filament tablice**

Filament tablice [30] omogućuju automatsko dohvaćanje podataka iz baze, uz minimalnu potrebu za ručnim kodiranjem. Tablice su ključna komponenta Filament paketa jer omogućavaju učinkovito prikazivanje, filtriranje, sortiranje i paginaciju velike količine podataka. Filament tablice se najbolje mogu objasniti na primjeru Student modela i *StudentResourcea* u predstavljenoj aplikaciji. Pomoću graditelja Filamentovih tablica (Filament's Table Builder) paketa može se prikazati sve studente u jednoj preglednoj tablici. Ta tablica, uz minimalnu potrebu za pisanjem koda, omogućuje i filtriranje i pretraživanje studenata iz baze podataka.

Jedna od prednosti Filament tablica je njihova visoka prilagodljivost. Programeri mogu lako definirati stupce tablice i prilagoditi način na koji se podaci prikazuju korisnicima. Uz to, tablice podržavaju dodavanje prilagođenih akcija [33], kao što su brisanje ili uređivanje podataka, što korisnicima omogućava da interaktivno upravljaju sadržajem direktno iz tablice. Filament tablice su dizajnirane da budu vizualno privlačne i u skladu sa suvremenim web standardima.

### 2.6.5 Filament widgets

Filament paket uključuje raznolike *widgete* koji značajno doprinose funkcionalnosti i interaktivnosti korisničkog sučelja administratorskih panela. Widgeti u Filamentu omogućavaju prikaz dinamičkih informacija poput statistika, najnovijih aktivnosti, grafičkih prikaza podataka i drugih relevantnih sadržaja koji pomažu korisnicima u efikasnijem upravljanju i nadzoru aplikacija.

Jedna od ključnih prednosti korištenja *widgeta* u Filamentu jest njihova visoka prilagodljivost. Programeri mogu lako stvoriti prilagođene *widgete* koji odgovaraju specifičnim potrebama projekta. Ovo uključuje mogućnost integracije s vanjskim izvorima podataka, prikazivanje podataka u realnom vremenu te implementaciju interaktivnih elemenata koji poboljšavaju korisničko iskustvo. Primjeri *widgeta* uključuju kartice za brzi pregled ključnih performansi aplikacije, liste zadataka, kalendar događaja, i grafikone koji prikazuju trendove ili analitiku. Svaki *widget* može se konfigurirati tako da prikazuje podatke relevantne za određene korisnike ili uloge unutar aplikacije, omogućavajući time personalizirani pristup informacijama.

Uvođenje *widgeta* u administrativno sučelje ne samo da povećava njegovu funkcionalnost, već i olakšava dnevne operacije korisnika, pružajući im brz i jasan uvid u najvažnije aspekte

sustava koji upravljaju. Korištenjem *widgeta*, Filament aplikacije postaju moćniji alat za upravljanje, sposoban za pružanje ključnih informacija na intuitivan i pristupačan način.

### 2.6.6 Filament plugins

Filament plugins su proširenja koja omogućuju dodavanje specifičnih funkcionalnosti u okviru Filament administrativnog sučelja, pružajući dodatne mogućnosti koje nisu obuhvaćene osnovnim instalacijama. Ovi dodaci olakšavaju prilagođavanje i proširenje standardnih mogućnosti, čineći Filament još moćnijim i fleksibilnijim alatom za izgradnju web aplikacija. Plugins u Filamentu omogućuju integraciju novih komponenti, poput prilagođenih formi, tablica, *widgeta*, ili čak novih vrsta autentifikacija i korisničkih interakcija.

Na primjer, programeri mogu instalirati *plugin* koji dodaje podršku za naprednije upravljanje medijima unutar aplikacije, omogućujući korisnicima da na efikasan način upravljaju slikama, videima i drugim medijskim datotekama. Drugi popularni *plugin* može uključivati alate za SEO optimizaciju, koji korisnicima omogućuju da izravno kroz administrativno sučelje upravljaju SEO postavkama svojih stranica.

Jedna od prednosti korištenja Filament plugins jest što oni omogućuju programerima da se usredotoče na specifične zahtjeve projekta bez potrebe za razvijanjem kompleksnih rješenja od nule. *Pluginsi* su često razvijeni od strane zajednice i dostupni su kroz Filament trgovinu ili na GitHubu, gdje zajednica može doprinijeti svojim kodom, što dodatno potiče suradnju i dijeljenje resursa među programerima.

Zaključno, upotreba Filament plugins poboljšava učinkovitost razvojnog procesa i pruža programerima snažan alat za prilagođavanje svojih aplikacija prema konkretnim potrebama korisnika. Uz stalne inovacije i doprinose zajednice, Filament plugins ostaju ključan element za razvoj visoko prilagodljivih i funkcionalno bogatih web aplikacija.

### 2.6.7 Filament zajednica i podrška

Filament zajednica i podrška ključni su elementi koji doprinose uspjehu i popularnosti ovog alata među programerima. Zajednica oko Filamenta se sastoji od programera i dizajnera koji aktivno doprinose razvoju i poboljšanju ovog paketa kroz različite platforme kao što su GitHub,



Discord kanali, i forumi. Zajednica nije samo mjesto za razmjenu koda i rješavanje tehničkih problema, već i prostor gdje članovi mogu dijeliti najbolje prakse, korisnička iskustva i inovativne ideje za korištenje Filamenta u različitim projektima. Aktivni članovi zajednice redovito organiziraju webinare, radionice i online sastanke, gdje se novi korisnici mogu upoznati s osnovama Filamenta, a iskusniji programeri mogu produbiti svoje znanje o naprednim funkcionalnostima. Podrška za Filament proširuje se i na službene dokumentacije koje su detaljno napisane i redovito ažurirane, pružajući korisnicima sve potrebne informacije za početak rada i rješavanje potencijalnih problema. Osim toga, dokumentacija nudi i vodiče koji korisnicima olakšavaju implementaciju složenijih funkcionalnosti. Važno je naglasiti da Filament teži inkluzivnosti i otvorenosti, potičući članove da doprinose bez obzira na njihovu razinu znanja ili iskustva. Ovaj pristup ne samo da unapređuje sam paket, već i gradi snažnu i suradničku atmosferu unutar zajednice. Zahvaljujući ovom dinamičnom ekosistemu, Filament se kontinuirano razvija i prilagođava potrebama svojih korisnika, čvrsto se uspostavlja kao jedan od ključnih alata u razvoju web aplikacija.

## **2.7 Docker**

Docker [35] je platforma koja omogućava programerima da jednostavno kreiraju, distribuiraju i izvršavaju aplikacije koristeći kontejnere. Kontejnerizacija omogućava programerima da pakiraju aplikaciju sa svim njenim ovisnostima u standardizirani paket, poznat kao kontejner, koji može biti efikasno premješten između različitih okruženja koja imaju instaliran Docker. Korištenjem Dockera, aplikacija se izvodi unutar kontejnera koji dijeli resurse operativnog sustava domaćina, ali je izoliran od drugih procesa. Time se osigurava da aplikacija radi jednako u svakom okruženju. Docker također pruža mogućnost verzioniranja kontejnera putem Docker Huba, gdje korisnici mogu pohraniti i dijeliti kontejnere sa zajednicom. Ova funkcionalnost omogućuje programerima da pristupe već izgrađenim kontejnerskim konfiguracijama, što dodatno ubrzava razvoj i implementaciju aplikacija. Dodatno, Docker olakšava skaliranje aplikacija kroz upravljanje višestrukim kontejnerima, što omogućava jednostavnu distribuciju opterećenja i upravljanje resursima. Dockerfile, tekstualna datoteka s konfiguracijskim naredbama, koristi se za automatizaciju procesa izrade Docker slike. Kroz Dockerfile, programeri definiraju korake potrebne za izgradnju slike, što može uključivati instalaciju softvera, konfiguraciju postavki i kopiranje potrebnih datoteka u kontejner. Kada je Dockerfile spreman, Docker može automatski

izgraditi sliku kontejnera koja je spremna za pokretanje, osiguravajući dosljednost i smanjujući mogućnost ljudske pogreške.

Ove funkcionalnosti čine Docker izuzetno popularnim alatom u svijetu razvoja softvera, jer pruža brzinu, sigurnost i fleksibilnost potrebnu za suvremene razvojne i produkcijske cikluse. Docker ne samo da pojednostavljuje razvoj aplikacija već i podiže efikasnost testiranja i *deploymenta*, što je čini idealnim rješenjem za mikroservisne arhitekture i *cloud* bazirane aplikacije.

### 2.7.1 Docker compose

*Docker compose* [36] je alat za definiranje i upravljanje aplikacijama koje imaju više docker kontejnera. Aplikacija koja je tema ovog rada ima dva kontejnera: kontejner za laravel aplikaciju i kontejner za bazu podataka. S *Docker Composeom*, korisnici mogu koristiti YAML datoteku za konfiguriranje usluga svoje aplikacije, što olakšava pokretanje više kontejnera kao jedinstvenu uslugu. Ovaj pristup je izuzetno koristan kada aplikacije zahtijevaju više odvojenih usluga koje trebaju raditi zajedno.

Jedna od ključnih prednosti Docker Composea je njegova sposobnost da automatski rješava i upravlja zavisnostima između kontejnera. Na primjer, ako aplikacija zahtijeva da baza podataka bude dostupna prije nego što se web server može pokrenuti, Docker Compose može konfigurirati da se kontejner baze podataka pokrene prvi. Ova funkcija osigurava da se aplikacija uvijek ponaša očekivano, bez obzira na okruženje u kojem se izvodi.

Uz to, Docker Compose olakšava upravljanje okruženjima i verzijama aplikacije kroz različite faze razvoja softvera, od razvoja, preko testiranja, do produkcije. Programeri mogu definirati različite konfiguracije za svako okruženje koristeći odvojene YAML datoteke ili specifične sekcije unutar jedne datoteke, što omogućava brz prijelaz između različitih faza bez rizika od konfiguracijskih grešaka.

Ove napredne mogućnosti čine Docker Compose nezaobilaznim alatom za timove koji traže efikasno rješenje za orkestraciju kontejneriziranih aplikacija, omogućavajući im da se fokusiraju na razvoj umjesto na upravljanje infrastrukturom.

## 2.7.2 Laravel Sail

Laravel Sail [36] je alat za upravljanje Docker okruženjem koji je posebno dizajniran za Laravel aplikacije. Sail omogućuje jednostavan način za pokretanje Laravel projekta koristeći Docker, bez potrebe za ručnim konfiguriranjem Docker kontejnera. Sail omogućuje programerima aplikacija da s nekoliko jednostavnih naredbi postave sve potrebne usluge, uključujući PHP, MySQL, Redis [37] i više. Sail se koristi u razvojnom okruženju. To znači da Docker treba posebno konfigurirati za pripremno i produkcijsko okruženje. Pomoću Saila, programeri mogu lako započeti s novim Laravel projektima ili integrirati Sail u postojeće projekte bez dubokog poznavanja Docker tehnologije. To čini Sail izvrsnom opcijom za sve koji žele iskoristiti prednosti kontejnizirane razvojne okoline, ali nemaju iskustva s kompleksnom Docker konfiguracijom.

## 2.8 Git

Git [38] je sustav za upravljanje verzijama koji programerima omogućava efikasno praćenje promjena u izvornom kodu tijekom razvoja aplikacije. Jedna od ključnih značajki Gita je njegova sposobnost da omogućava rad s više verzija projekta istovremeno. To programerima omogućuje eksperimentiranje i razvijanje novih funkcionalnosti bez rizika da će to utjecati na glavnu verziju *softwarea*. Svaka zabilježena promjena se zove *commit* koji je obilježen autorom, opisom i vremenom kad je promjena napravljena. Time se olakšava praćenje povijesti izmjena. Git podržava upravljanje neovisnim granama koda koje se kasnije mogu sigurno integrirati u glavni projekt. Zbog svoje svestranosti i moćnih funkcionalnosti, Git je neophodan alat za timove koji žele efikasno upravljati svojim projektima.

### 2.8.1 Github

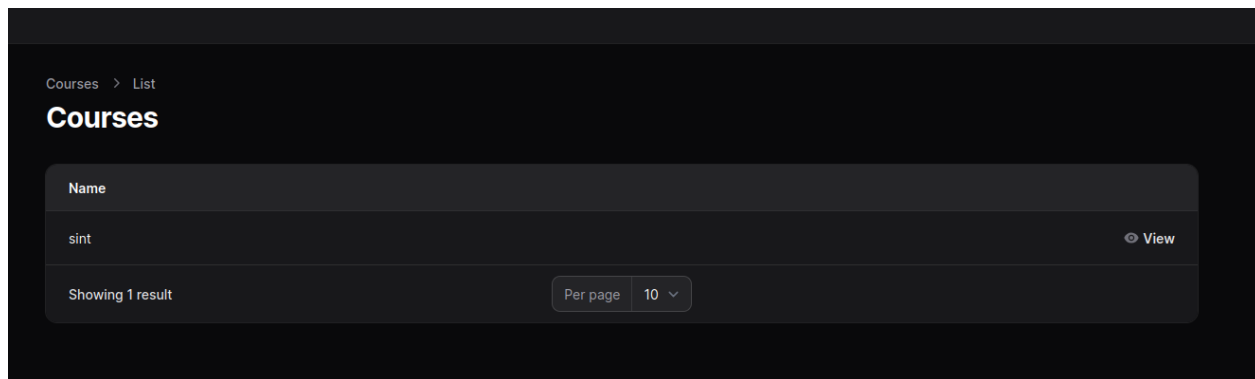
GitHub je platforma za smještaj (hosting) koda koji koristi Git. Omogućuje programerima i timovima da surađuju na projektima koristeći alate kao što su upravljanje granama, pregled promjena (eng. commits), povijest izmjena i *pull requests* za integraciju promjena u neku granu. GitHub također nudi niz dodatnih funkcionalnosti, uključujući *issue tracking*, *wikis* za projekte, kao i alate za reviziju koda. Ove funkcionalnosti pomažu timovima da organiziraju i prioritetiziraju svoj rad, dokumentiraju projekte i učinkovito komuniciraju o problemima i značajkama koje razvijaju.

### 3 APLIKACIJA ZA PRAĆENJE RADA NA PROJEKTIMA

Aplikacija omogućuje voditeljima kolegija da kroz ovu aplikaciju imaju uvid u rad studenata na projektima kolegija koje vode. Isto tako, aplikacija omogućuje studentima da se zabilježe na projekt na kojem žele raditi te da kroz aplikaciju unesu dokumente, ocjene i slično. U ovom poglavlju će se detaljno predstaviti funkcionalnosti ove aplikacije. Aplikacija ima korisnike koji su podijeljeni po ulogama: student, voditelj kolegija i *super admin*. Ovo poglavlje će biti podijeljeno upravo po tim ulogama. Prvo će biti predstavljene funkcionalnosti iz perspektive studenata zatim iz perspektive voditelja kolegija i *super admina*.

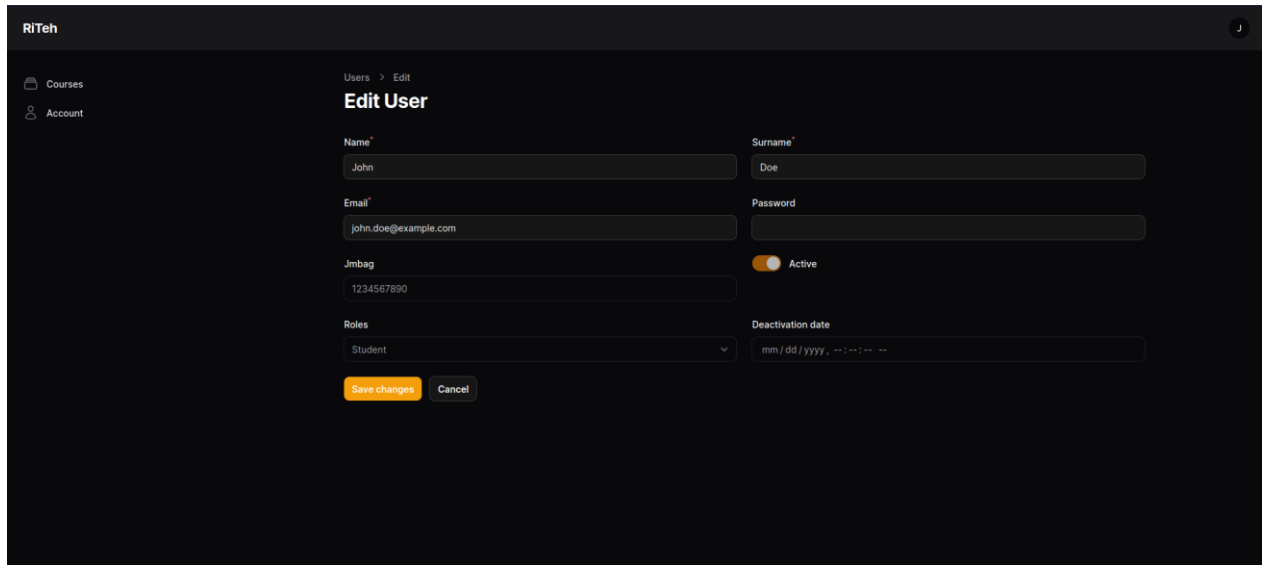
#### 3.1 Student

Nakon što se student prijavi u aplikaciju, prvo što vidi je stranica s tablicom na kojoj su prikazani svi kolegiji koje taj student pohađa. Na slici 3.1 se vidi ta tablica i u ovom slučaju student pohađa samo jedan kolegiji.



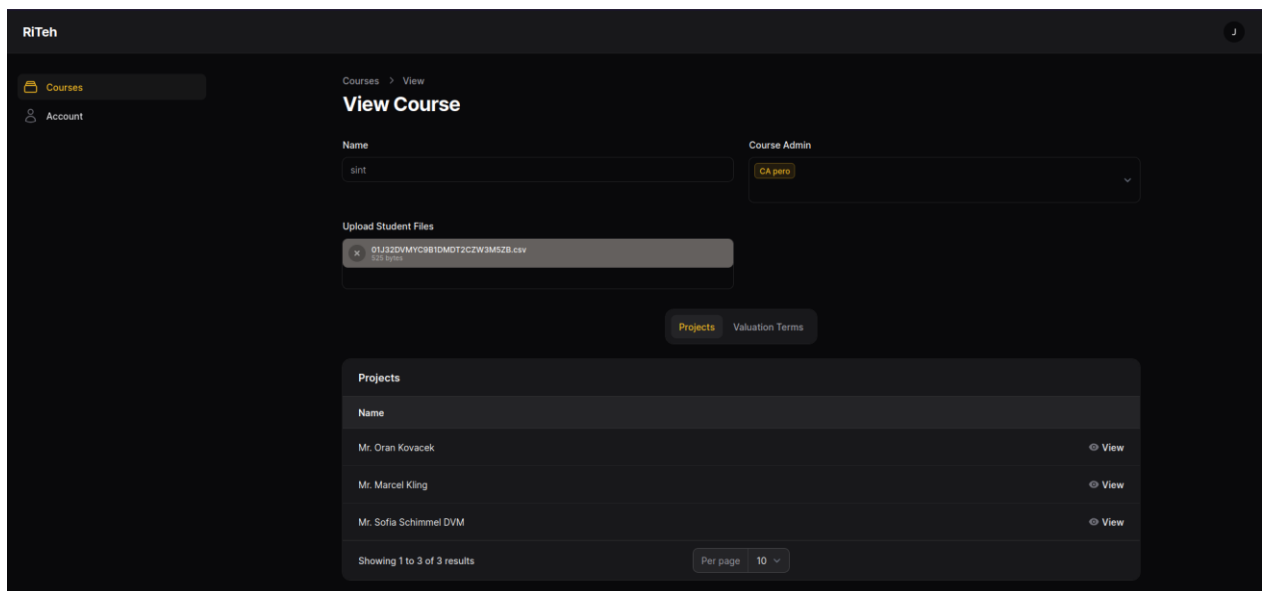
Slika 3.1 – Primjer tablice kolegija

Student ima mogućnost da pogleda detalje svog profila, od kojih samo neke može mijenjati. Na slici 3.2 prikazani su podaci studenta do kojih se može doći klikom na gumb sa tekстом: *Account*. Student može promijeniti svoje ime, prezime, e-adresu i šifru, ali ne može promijeniti jmbag, aktivnost, ulogu i datum deaktivacije.



*Slika 3.2 - Stranica sa podacima students*

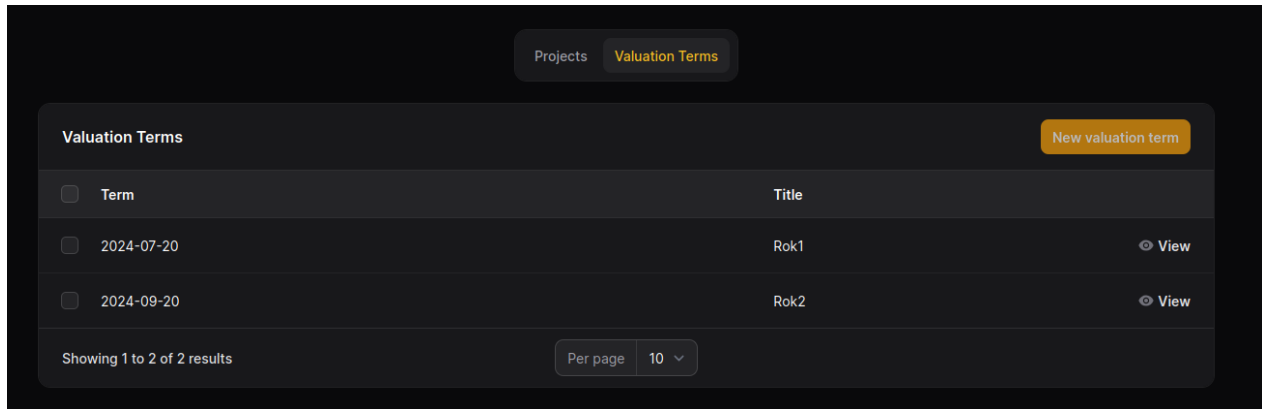
Da bi student pristupio kolegiju i vidio detalje nekog kolegija, mora odabrati kolegij u već spomenutoj tablici kolegija. Nakon toga, student će vidjeti stranicu kolegija, koja izgleda kao što je to prikazano na slici 3.3.



*Slika 3.3 - Primjer stranice kolegija*

Na stranici kolegija student može vidjeti ime kolegija i imena voditelja kolegija. Oba polja student ne može mijenjati jer nema te privilegije. Ispod spomenutih polja student vidi listu projekata na tom kolegiju. U tablici projekata vidi se ime svakog projekta te gumb sa tekstom: *View* koji studenta

vodi na stranicu projekta. Osim liste projekta student može vidjeti i listu rokova za vrednovanja (eng. valuation terms). Na slici 3.4 možemo vidjeti kako izgleda ta tablica.

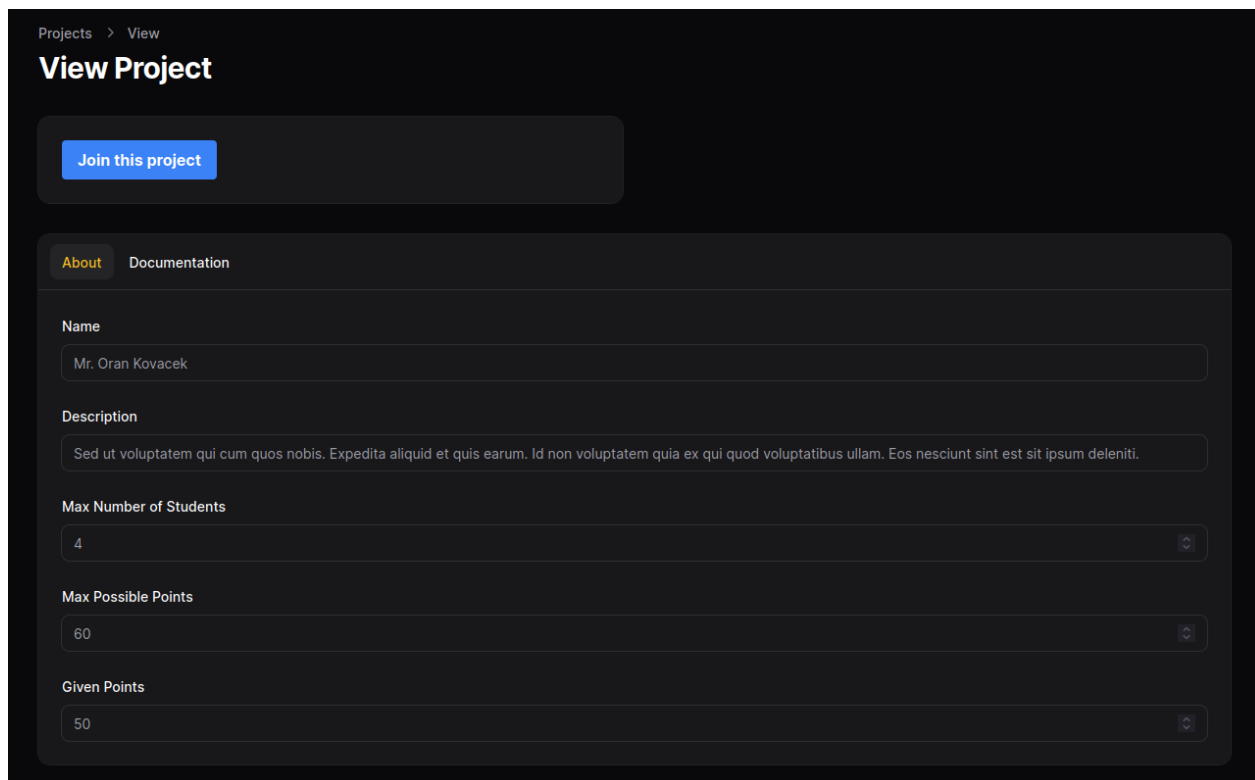


Term	Title	
<input type="checkbox"/> 2024-07-20	Rok1	<a href="#">View</a>
<input type="checkbox"/> 2024-09-20	Rok2	<a href="#">View</a>

Slika 3.4 - Tablica rokova vrednovanja

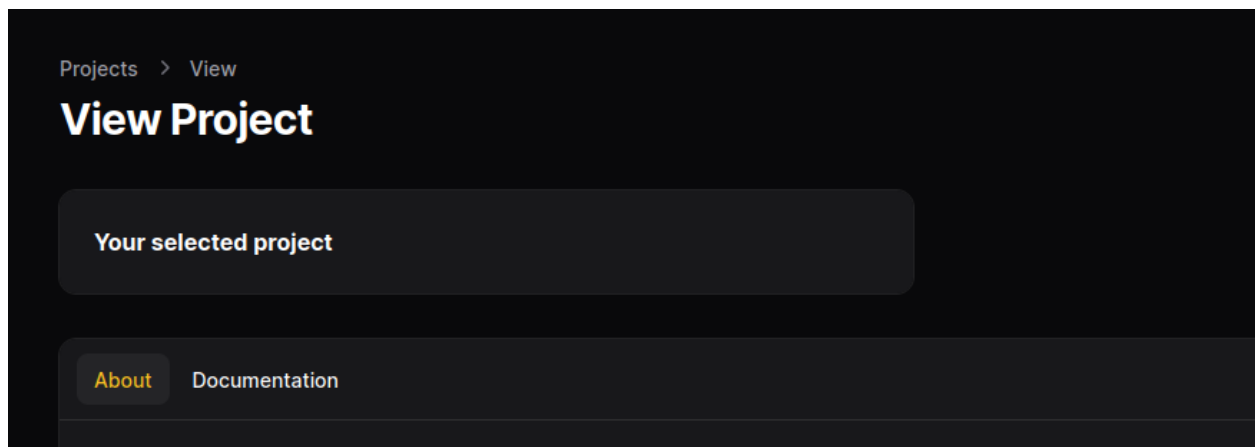
Naravno, kad student pritisne na gumb sa tekстом: *Projects*, prikazuje se tablica projekata, a kad se pritisne gumb s tekстом: *Valuation Terms* prikazuje se tablica s rokovima vrednovanja. Na tablici rokova vidljiva su tri stupca. Prvi stupac prikazuje datum roka, drugi stupac prikazuje naslov ili ime dodijeljeno tom roku. Zadnji stupac sadrži gumb koji vodi studenta na stranicu pojedinog roka.

U tablici projekata su izlistani svi projekti dostupni za taj kolegij. Student može odabrati projekt na kojem želi raditi. Kad student odabere jedan od projekata, preusmjeren je na stranicu tog projekta gdje može vidjeti: ime projekta, opis projekta, maksimalan broj studenata na tom projektu i najveći mogući broj bodova koji se može ostvariti na tom projektu. Ta stranica projekta i spomenuta polja se mogu vidjeti na slici 3.5.



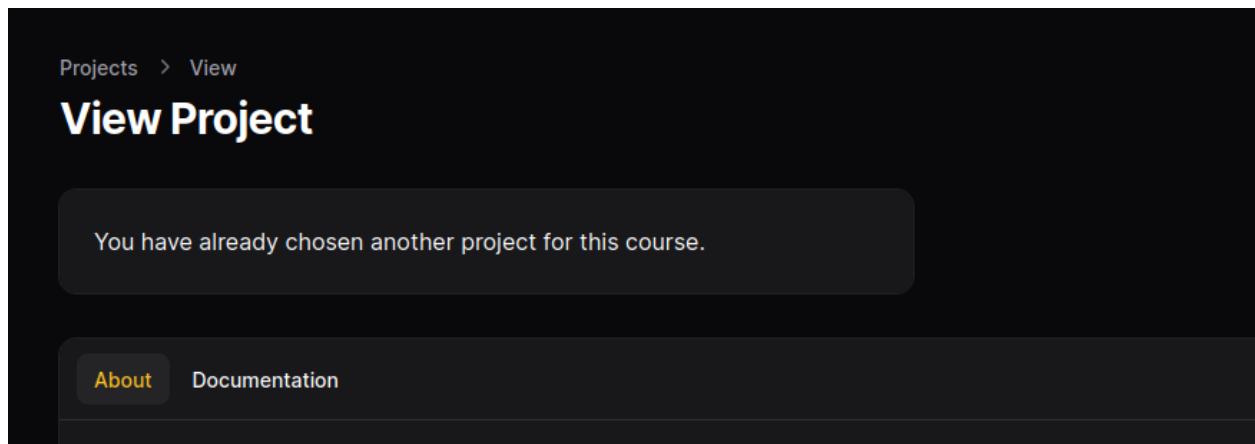
*Slika 3.5 - Stranica projekta*

Na vrhu stranice projekta nalazi se gumb sa tekstom: *Join this project*. Taj gumb student može pritisnuti ukoliko se odluči pridružiti i raditi na tom projektu. Nakon što student pritisne taj gumb, na stranici projekta prikazuje se tekst: *Your selected project* kako je vidljivo na slici 3.6.



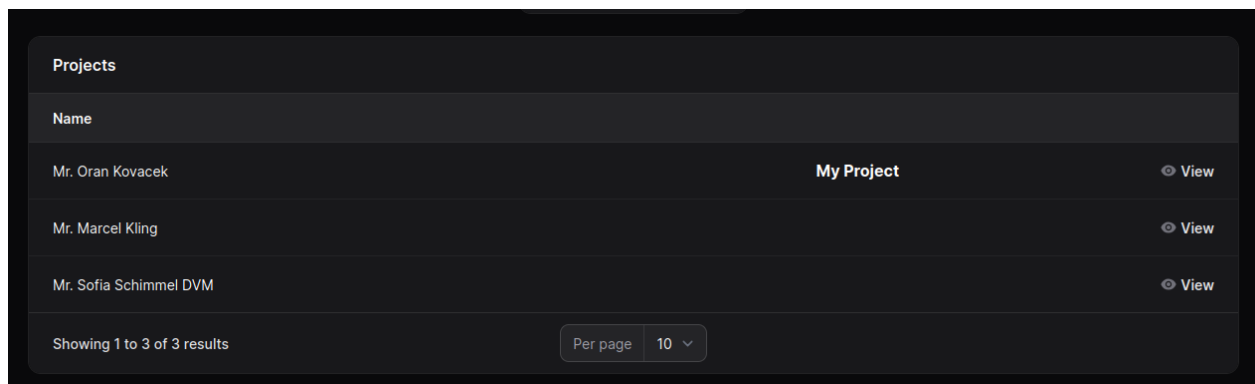
*Slika 3.6 - Oznaka za odabrani projekt*

Student se može prijaviti samo na jedan projekt. Stoga, kada odabere drugi projekt u tablici projekata i pristupi stranici tog projekta, pojavit će se obavijest koja ga informira da je već izabrao svoj projekt te da prikazani projekt više ne može odabrati. Tekst te obavijesti prikazan je na slici 3.7.



*Slika 3.7 - Oznaka da je projekt već izabran*

U tablici projekata postoji oznaka za odabrani projekt kako bi se student lakše snalazio te kako bi aplikacija bila intuitivnija. Ta oznaka je tekst: *My Project* i može se vidjeti na slici 3.8.



Projects	
Name	
Mr. Oran Kovacek	<b>My Project</b> <a href="#">View</a>
Mr. Marcel Kling	<a href="#">View</a>
Mr. Sofia Schimmel DVM	<a href="#">View</a>

Showing 1 to 3 of 3 results

Per page 10

*Slika 3.8 - Oznaka za odabrani projekt u tablici*

Na stranici projekta vidljiva je tablica u kojoj su studenti koji su na tom projektu. Kao što se vidi na slici 3.9 ta tablica ima stupce za ime, elektroničku adresu, jmbag te oznaku za timskog vođu kojeg označava isključivo voditelj kolegija.



Name	Email	Jmbag	Team lead
John	john.doe@example.com	1234567890	<input checked="" type="checkbox"/>
Laura	laura.garcia@example.com	7890123456	<input type="checkbox"/>
Jessica	jessica.anderson@example.com	9012345678	<input type="checkbox"/>

Showing 1 to 3 of 3 results

Per page 10

Slika 3.9 - Popis studenata na projektu

Kroz ovu aplikaciju, student može unijeti pdf dokument u kojem su opisani sastanci koje je imao tim koji radi na projektu. Popis svih dokumenata koje je student unio u aplikaciju vidljiv je pritiskom na karticu sa tekстом *Meetings* kao što je to vidljivo na slici 3.10.

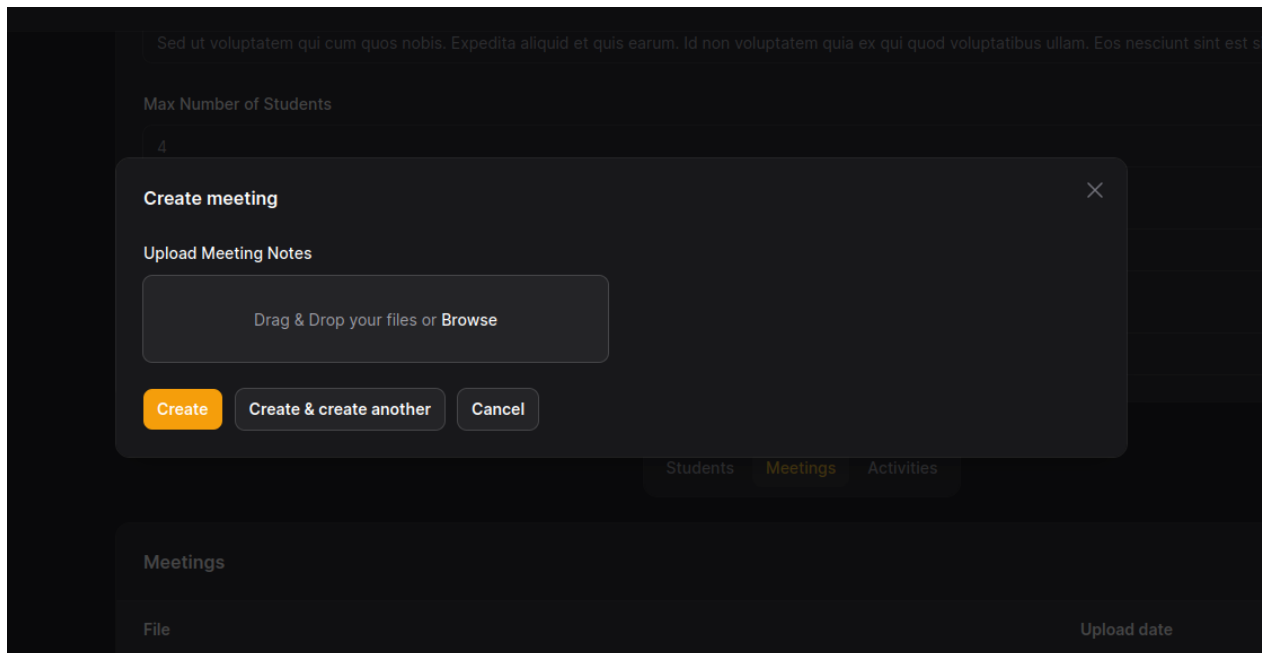
File	Upload date	
testing-fileanme.pdf	Jul 18, 2024	<a href="#">View</a>
testing-fileanme.pdf	Jul 18, 2024	<a href="#">View</a>
testing-fileanme.pdf	Jul 18, 2024	<a href="#">View</a>

Showing 1 to 3 of 3 results

Per page 10

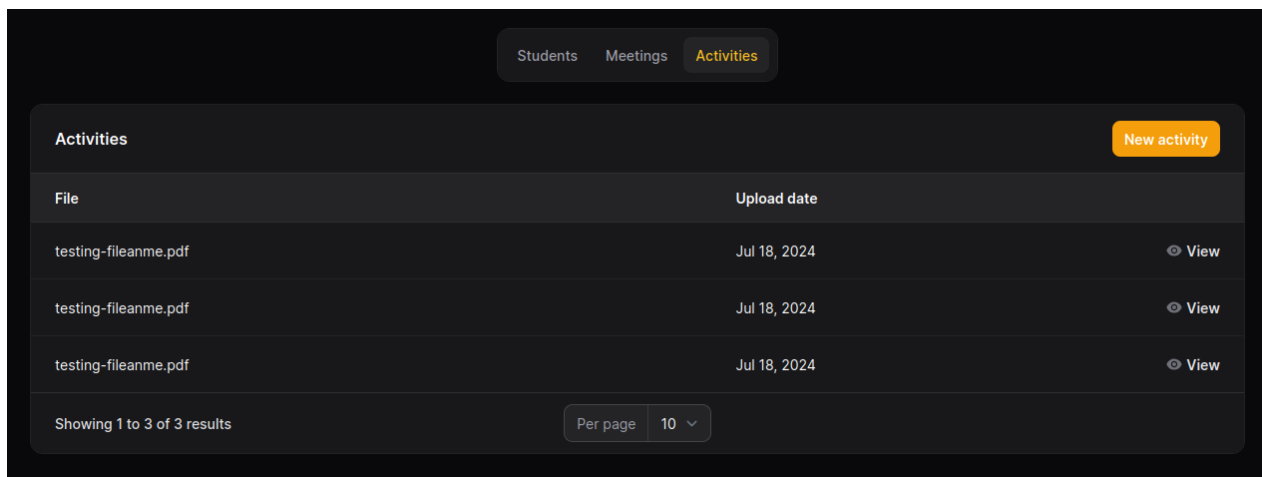
Slika 3.10 - Popis pdf dokumenata za sastanke

U spomenutoj tablici vidimo 3 stupca: ime datoteke, datum unošenja dokumenta u aplikaciju te gumb koji omogućava da se dokument pogleda u novom *tabu*. Na vrhu tablice se vidi gumb sa tekстом: *New meeting* koji omogućava studentu da unese novi pdf dokument. Pritiskom na taj gumb otvara se novi obrazac koji služi sa unos pdf dokumenta kao što je to vidljivo na slici 3.11.



*Slika 3.11 - Obrazac za unos novog pdf dokumenta*

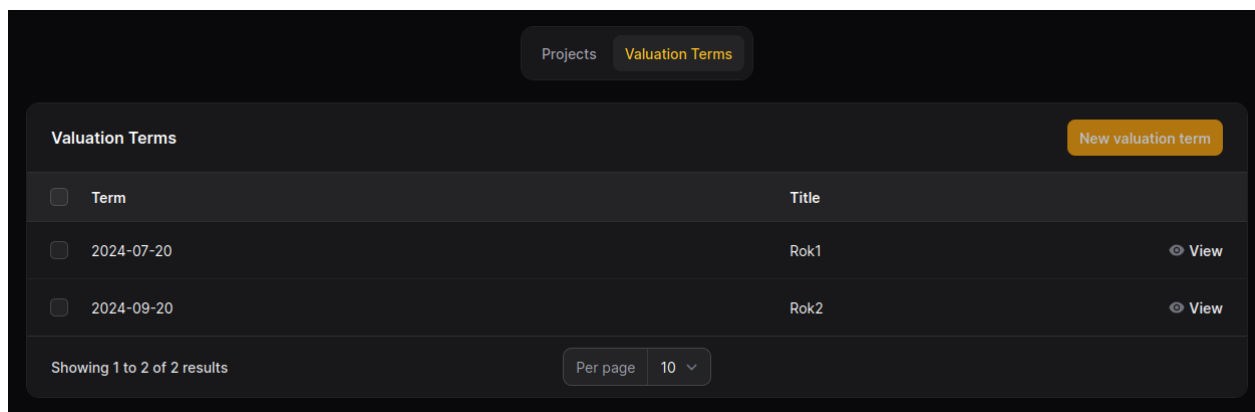
Ista tablica sa istim obrascem postoji i za aktivnosti. Student može unijeti pdf dokument u kojem se nalazi popis aktivnosti i listu tih pdf dokumenata može vidjeti u tablici kao što je to vidljivo na slici 3.12.



*Slika 3.12 - Tablica aktivnosti*

Tijekom rada na projektu, studenti ocjenjuju svoj rad te rad i aktivnost svojih kolega s kojima surađuju na projektu. Kako bi ocijenili sebe i svoje kolege, studenti moraju ispuniti obrasce za ta vrednovanja, ali ti obrasci moraju biti ispunjeni prije isteka roka. Voditelj kolegija postavlja rokove

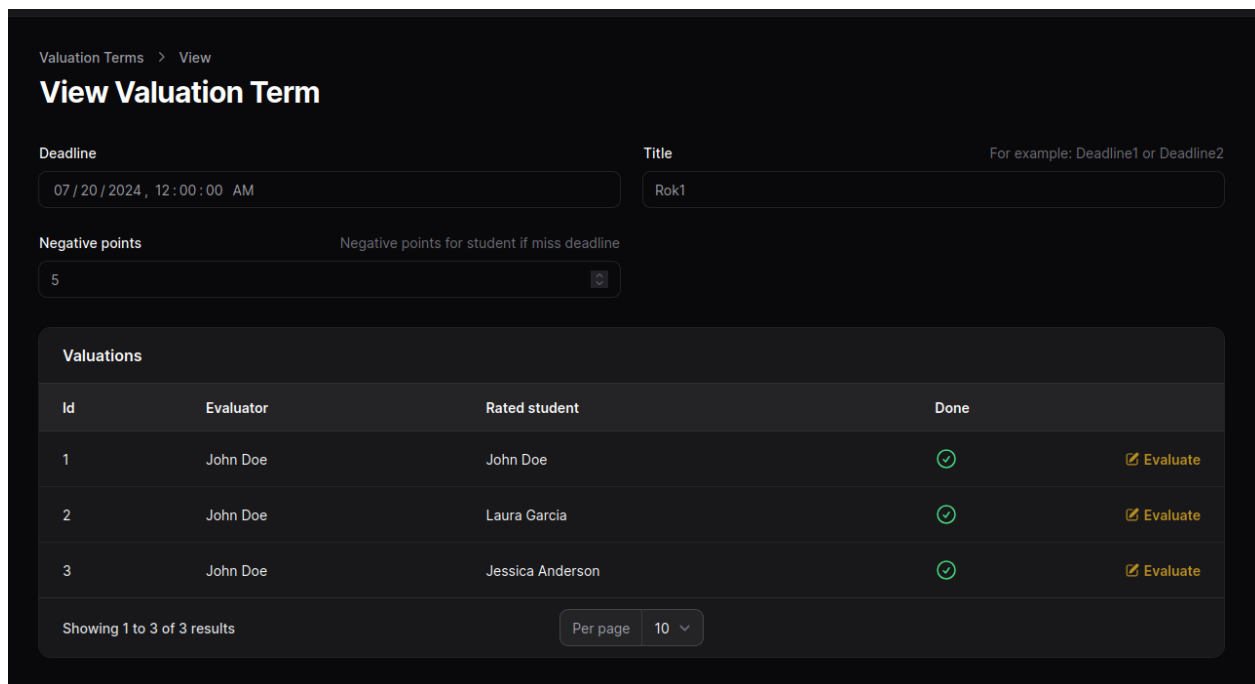
za vrednovanja. Na primjer, voditelj može postaviti dva roka: Rok1 i Rok2, a studenti moraju ispuniti obrasce vrednovanja prije isteka postavljenih rokova. Na stranici kolegija postoji kartica sa tekstom: *Valuation Terms* i pritiskom na tu karticu studentu postane vidljiva tablica sa rokovima vrednovanja kao što se to može vidjeti na slici 3.13.



Term	Title	
<input type="checkbox"/> 2024-07-20	Rok1	<a href="#">View</a>
<input type="checkbox"/> 2024-09-20	Rok2	<a href="#">View</a>

*Slika 3.13 - Tablica rokova vrednovanja*

Na tablici se vidi datum roka i ime roka. Na vrhu tablice je gumb za kreiranje roka, ali je taj gumb onemogućen za studenta i može ga koristiti samo voditelj kolegija. Pritiskom na gumb sa tekstom: *View* student je preusmjeren na stranicu roka vrednovanja gdje su vidljivi detalji o tom roku: datum, naslov i broj negativnih bodova koje student može dobiti ako ispuni obrasce za vrednovanje nakon specificiranog datuma. Stranica s detaljima o roku vrednovanja prikazana je na slici 3.14.



*Slika 3.14 - Stranica roka vrednovanja*

Na toj stranici je vidljiva i tablica vrednovanja. Budući da student treba ocijeniti svakog člana tima, u tablici se nalazi poveznica za obrazac ocjenjivanja svakog studenta u timu. U tablici su vidljiva imena studenata koji ocjenjuju i imena studenata koji se ocjenjuju. Predzadnji stupac sadrži oznaka koja sugerira da li je obrazac ispunjen ili ne. Ako je student ispunio obrazac, u predzadnjem stupcu nalazi se zelena kvačica, a u protivnom crveni križić. Nakon što student ispuni obrazac, više mu nije omogućen pristup istom. Pristup obrascu za vrednovanje nekog studenta omogućen je klikom na gumb s tekстом Evaluate, a sam obrazac prikazan je na slici 3.15.

Architecto earum qui et voluptatem impedit.\*

A  
Odio est eligendi nisi ducimus id quibusdam consequatur. Nihil voluptatem nam ut repellendus quaerat odio molestias. Facilis quis et dolorum rem minima.

B  
Dolores veritatis et illo dolorum. Voluptatum veniam molestiae laboriosam quaerat voluptatem. Nihil sunt nam ipsum blanditiis voluptates ut itaque.

C  
Ut ea molestiae in repellendus quae nihil ab voluptas. Saepe est maxime eligendi doloremque excepturi. Rerum esse dolores provident velit.

D  
Atque voluptas perferendis et nostrum. Sed qui ipsam ut odio. Veniam labore nostrum sit perferendis dignissimos voluptatum in.

Omnis itaque sint dolor accusamus.\*

A  
Aut atque corporis repellat illo pariatur sequi sunt. A facere eos adipisci. Inventore distinctio ipsum molestiae expedita laboriosam quaerat saepe. Veritatis et dignissimos doloribus sed.

B  
Ut itaque iste qui vel harum. Iusto ab quasi perferendis quae et ducimus veritatis. Ex est sequi dolor deleniti consectetur. Et consequatur error corporis necessitatibus asperiores soluta quae ut.

C  
Quia ipsa sit nihil praesentium. Est ea quia eum est voluptas voluptatem suscipit. Rerum animi ut expedita itaque fugit.

D  
Quisquam sapiente est modi est quaerat reiciendis molestiae. Illo et commodi harum omnis quia.

Extra comment

Done

*Slika 3.15 - Obrazac vrednovanja*

U obrascu za vrednovanje nalaze se pitanja o angažmanu, ponašanju i aktivnostima studenta. Za svako pitanje, student odabire jednu od ponuđenih ocjena. Na slici 3.15 to su ocjene: A, B, C i D. Svaka ocjena nosi određeni broj bodova koje određuje voditelj kolegija. Kao što je vidljivo na slici 3.15, na kraju obrasca, nakon svih pitanja, student može ostaviti i neki svoj komentar. Svaki student ima uvid u to koliko je bodova ostvario na svakom roku i to je vidljivo na dnu stranice kolegija. Primjer takve tablice prikazan je na slici 3.16.

Full name	Rok1 (2024-07-20)	Rok2 (2024-09-20)	Average	Rok1 (negative points)	Rok2 (negative points)	Plocica deadline (negative points)	Total negative points	Final
John Doe	48.795180722892	51.612903225806	50.204041974349	5	0	0	5	45.20404197

*Slika 3.16 - Tablica bodova*

Na tablici bodova student može vidjeti koliko je bodova ostvario na svakom roku, prosjek bodova ostvarenih na svim rokovima, negativne bodove koji mogu biti dodijeljeni u slučaju da je student zakasnio ispuniti obrasce za vrednovanje. U tablici je vidljiv stupac koji pokazuje ukupan broj negativnih bodova i konačan broj bodova koje je student ostvario.

### **3.2 Voditelj kolegija**

U ovom poglavlju su predstavljene sve funkcionalnosti koje voditelj kolegija može koristiti, a student ne može. Prvo su predstavljene funkcionalnosti vezane za kolegije, zatim funkcionalnosti koje se odnose na proces vrednovanja, a na kraju je prikazano kako voditelj kolegija upravlja podacima studenata i projekata.

Voditelj kolegija može mijenjati ime kolegija i može dodati novog voditelja kolegija. Kao što je vidljivo na slici 3.17, postoji i polje za prijenos csv datoteke u kojoj se nalaze podaci svih studenata koji pohađaju taj kolegij.

Slika 3.17 - Obrazac za podatke kolegija

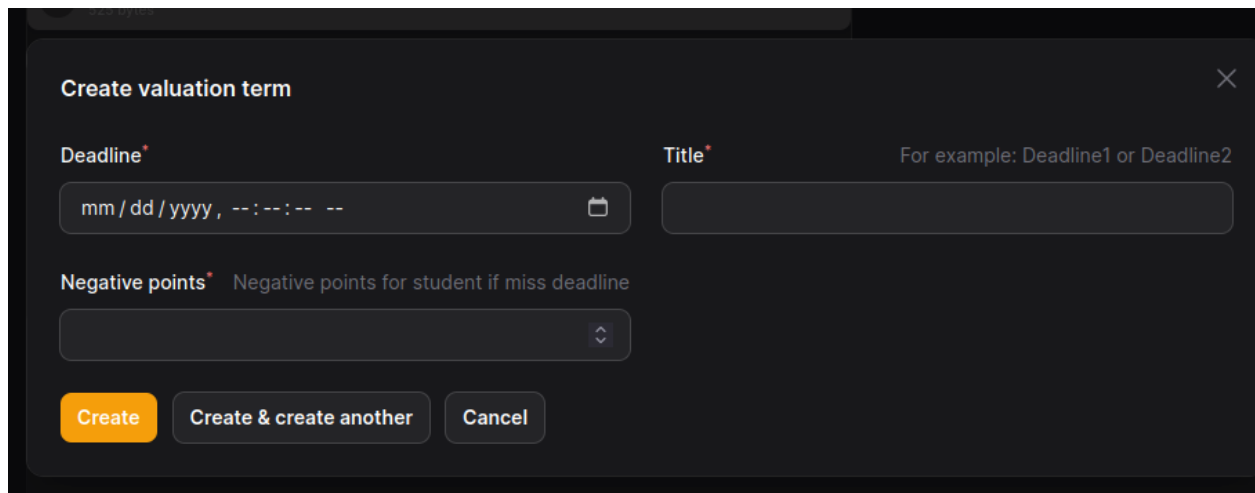
Na slici 1.17 vidi se obrazac za kolegij. Iako student može vidjeti ovaj obrazac, ne može mijenjati vrijednosti tih polja. Vrijednosti može mijenjati samo voditelj kolegija. Voditelj kolegija ima CSV datoteku u kojoj su podaci studenata: ime, prezime, elektronička adresa i JMBAG. Kada voditelj prenese tu datoteku u aplikaciju, stvoreni su studenti čija je inicijalna šifra: 12345678. Pri prvoj prijavi, studenti mogu promijeniti svoju šifru, kako je to opisano u prethodnom poglavlju.

Studentima mora biti omogućeno da vrednuju jedni druge, stoga voditelj kolegija mora stvoriti rokove za vrednovanje. To se može postići pritiskom na karticu s tekстом: *Valuation Terms*, a zatim na gumb s tekстом: *New valuation term*, kako je prikazano na slici 3.18.

Term	Title	
<input type="checkbox"/> 2024-07-20	Rok1	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">View</a>
<input type="checkbox"/> 2024-09-20	Rok2	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">View</a>

Slika 3.18 - Tablica sa rokovima vrednovanja

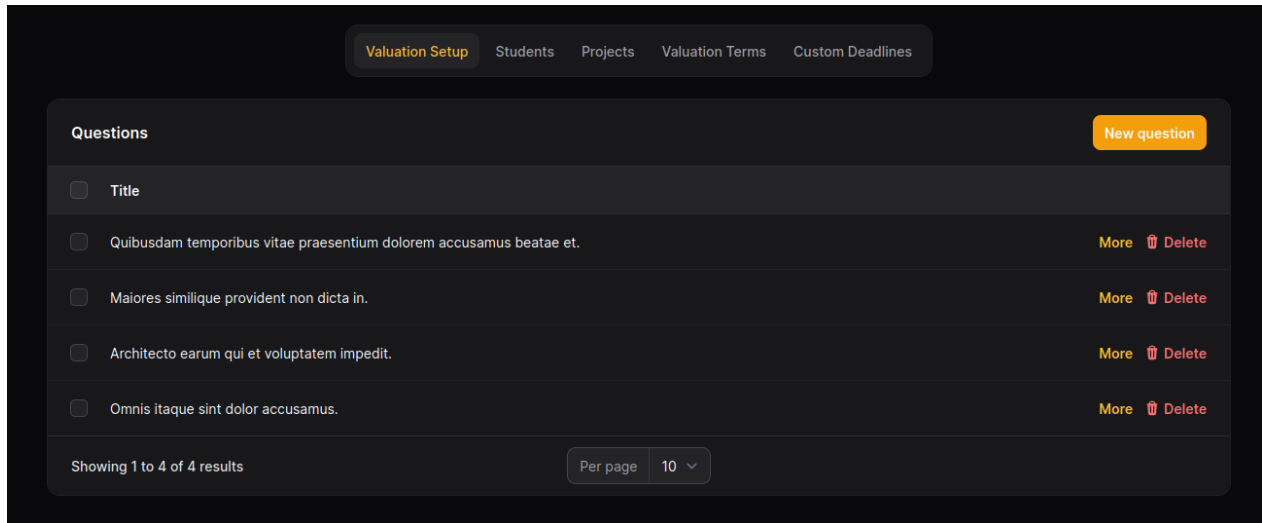
Obrazac za stvaranje novog roka izgleda kao na slici 3.19. Voditelj kolegija treba odrediti datum roka, ime za taj rok te mora specificirati koliko negativnih bodova će biti dodijeljeno studentu ukoliko zakasni sa vrednovanjima svojih kolega.



*Slika 3.19 - Obrazac za kreiranje novog roka*

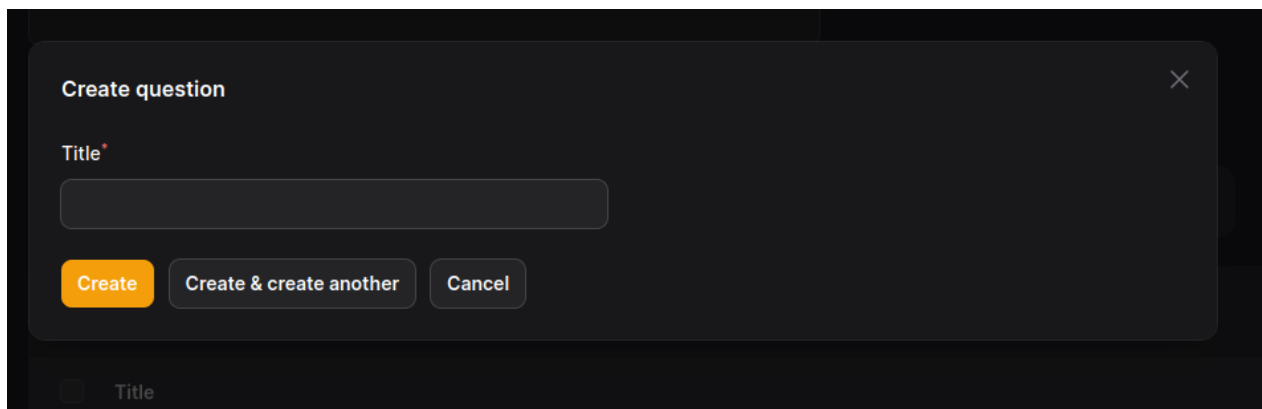
Voditelj kolegija je dužan odrediti i pitanja koja će biti postavljena u obrascima za vrednovanje. Studenti ocjenjuju svoje kolege tako da ocjenom odgovore na pitanja koja su postavljena u obrascima za vrednovanje, a ta pitanja određuje voditelj kolegija tako da pritisne na karticu sa tekстом: *Valuation Setup*, a zatim na gumb s tekстом: *New question*, kako je to prikazano na slici 3.20.





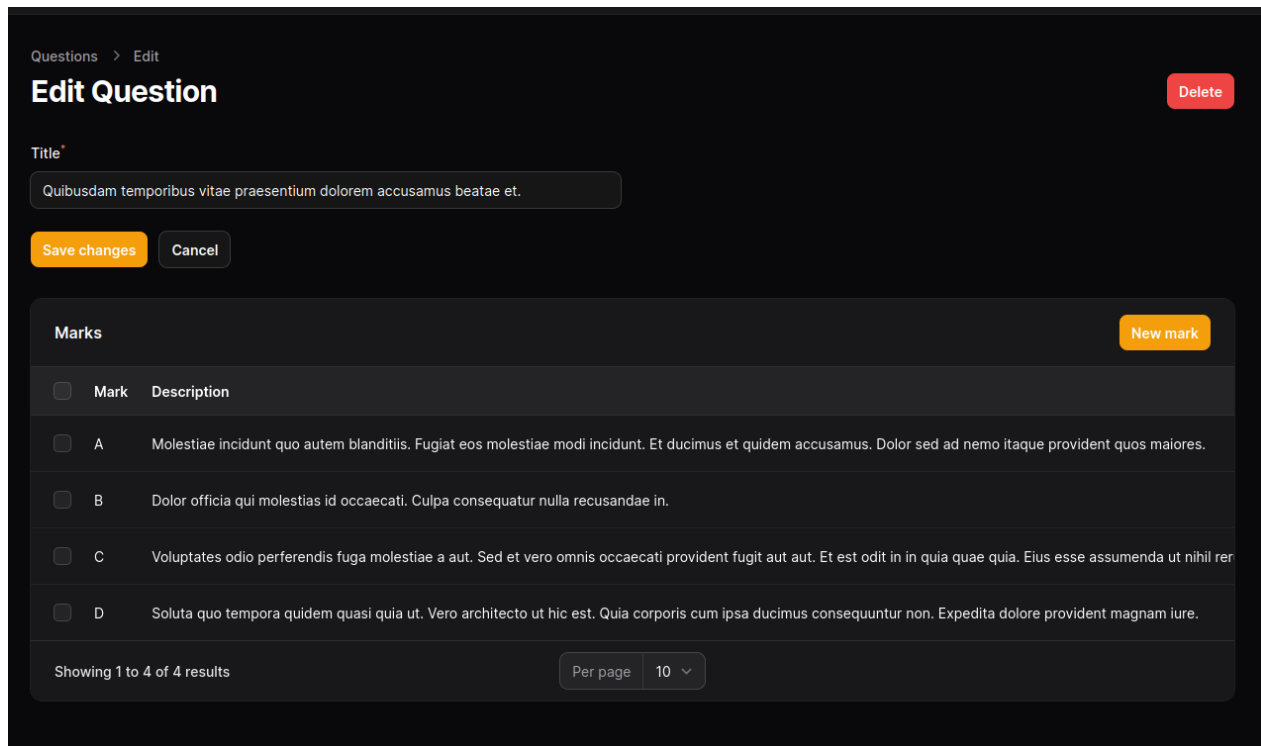
*Slika 3.20 - Tablica sa pitanjima za obrasce vrednovanja*

Obrazac za stvaranje novog pitanja je jednostavan. Sadrži samo jedno polje u kojem voditelj kolegija mora samo unijeti tekst. Obrazac je predstavljen na slici 3.21.



*Slika 3.21 - Obrazac za stvaranje pitanja*

Zadnji obavezan korak u procesu postavljanja vrednovanja je određivanje ocjena za svako pitanje. Kako bi se odredile ocjene za svako pitanje, voditelj kolegija mora prvo pritisnuti na tekst *More* u tablici pitanja kao što je to prikazano na slici 3.20. Nakon toga, voditelj će biti preusmjeren na stranicu pitanja koja izgleda kao na slici 3.22.



*Slika 3.22 - Stranica jednog pitanja*

Na stranici pitanja, voditelj kolegija može izmijeniti tekst pitanja i može upravljati ocjenama dostupnim za to pitanja. Na slici 3.22, prikazana je tablica s ocjenama. Svaka ocjena ima svoje slovo. Na primjer, ocjena može bit označena sa slovom A ili B. Svaka ocjena ima i svoj opis. Ocjena se dodaje pritiskom na gumb sa tekстом: *New mark*. Obrazac za stvaranje nove ocjena prikazan je na slici 3.23.

Mark \*

Description \*

Points \*

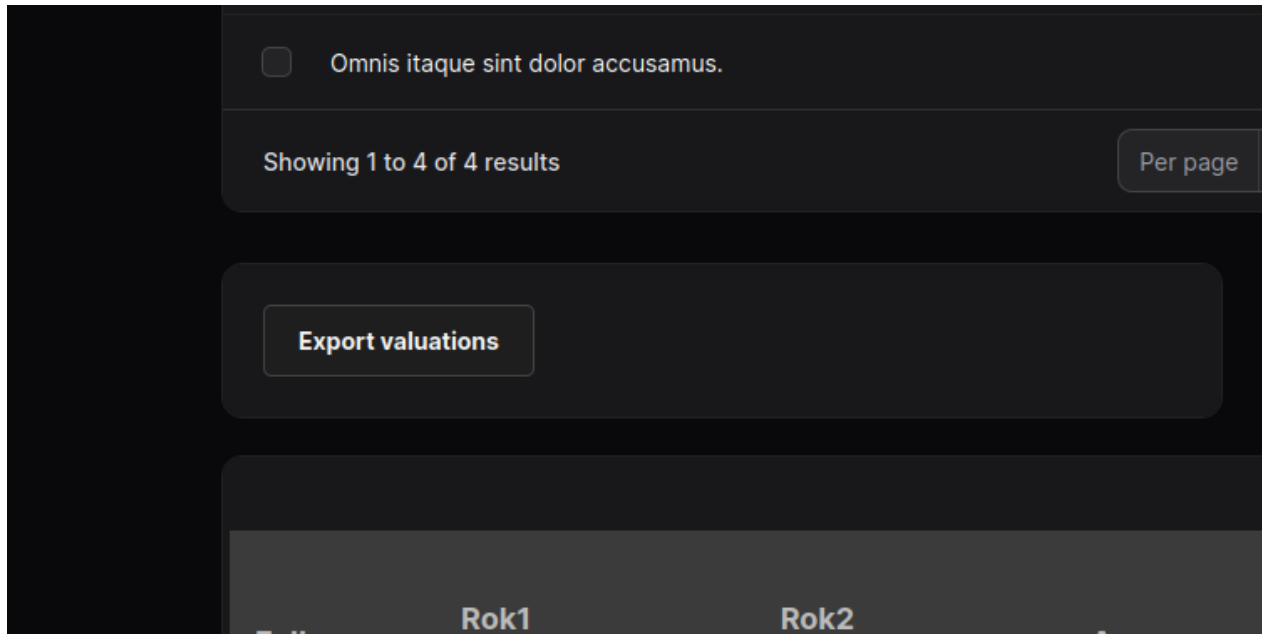
Create Create & create another Cancel

A Molestiae incidunt quo autem blanditiis. Fugiat eos molestiae modi incidunt. Et ducimus et quidem accusamus. Dolor sed

*Slika 3.23 - Obrazac za stvaranje nove ocjena*

U prvom polju tog obrasca, voditelj određuje slovo asocirano s tom ocjenom. To polje prima samo jedno slovo i korisniku je onemogućeno da unese više od jednog slova. Drugo polje je tekst koji služi za opis te ocjene. Posljednje polje prima samo numeričke vrijednosti i odnosi se na broj bodova koji nosi ta ocjena.

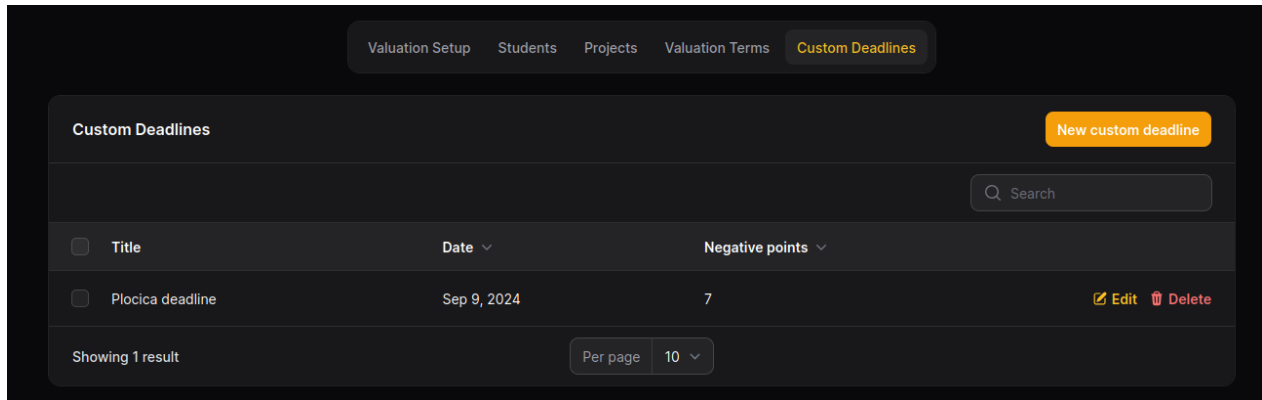
Na stranici kolegija nalazi se gumb koji je vidljiv na slici 3.24 i služi da voditelj kolegija može preuzeti csv datoteku sa svim vrednovanjima.



*Slika 3.24 - Gumb za preuzimanje datoteke s vrednovanjima*

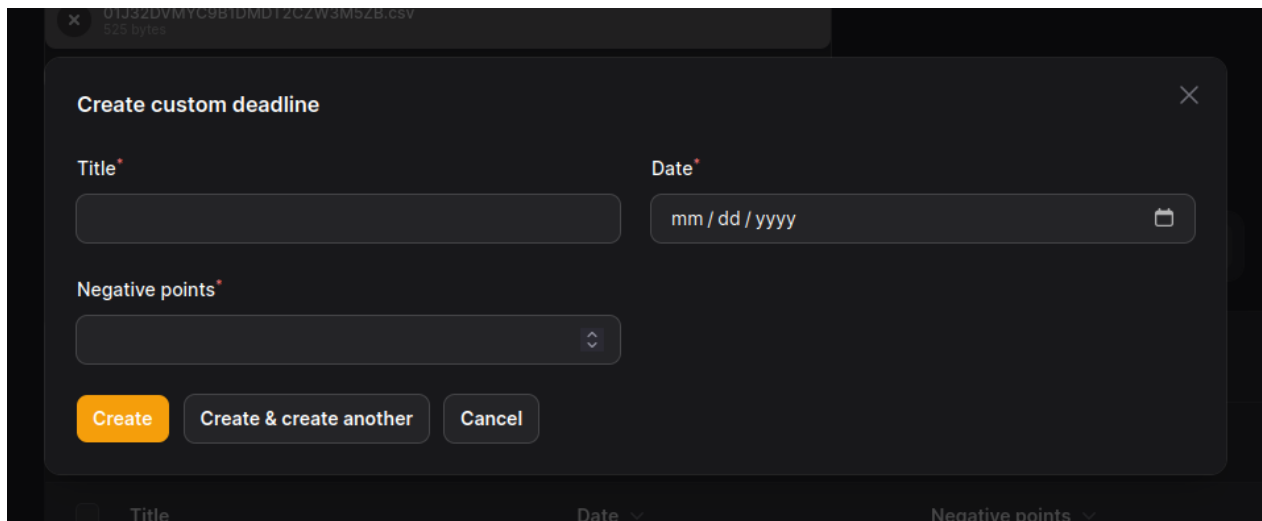
U preuzetoj csv datoteci nalaze se sve vrijednosti za svako vrednovanje, a stupci u toj datoteci su: ime roka, ime studenta koji ocjenjuje, ime studenta koji je ocjenjen, ime projekta, dodatan komentar, datum stvaranja vrednovanja te tekst svakog pitanja.

Voditelj kolegija može stvoriti i prilagođene rokove (eng. custom deadlines). Do sad su bili opisani samo rokovi za vrednovanja, ali u sklopu kolegija postoji mogućnost da će biti potrebno definirati rokove za neke druge obveze. Na primjer, rok za predaju PCB [39] sheme. U tom slučaju, voditelj može stvoriti prilagođeni rok kojem će dati ime, datum te broj negativnih bodova koji će biti dodijeljeni studentu ukoliko obvezu ne obavi na vrijeme. Na slici 3.25 prikazana je tablica prilagođenih rokova.



Slika 3.25 - Tablica prilagođenih rokova

Pritiskom na gumb sa tekстом: *New custom deadlines*, voditelju se otvara obrazac za stvaranje novog roka. Obrazac ima polje za naslov, datum i broj negativnih bodova. Obrazac je prikazan na slici 3.26.



Slika 3.26 - Obrazac za stvaranje prilagođenog roka

Kada voditelj želi dodijeliti negativne bodove nekim studentima onda mora pritisnuti gumb sa tekстом: *Edit* na tablici prilagođenih rokova. Nakon toga je preusmjeren na stranicu roka gdje se nalazi tablica sa studentima. Kada želi dodijeliti negativne bodove nekom studentu onda pritisne gumb u stupcu *Applied* kako je to prikazano na slici 3.27.

Custom Deadline		
Student full name	Student email	Applied
John Doe	john.doe@example.com	<input type="checkbox"/>
Jane Smith	jane.smith@example.com	<input type="checkbox"/>
Emily Johnson	emily.johnson@example.com	<input type="checkbox"/>
Michael Brown	michael.brown@example.com	<input type="checkbox"/>
Sarah Davis	sarah.davis@example.com	<input type="checkbox"/>
David Wilson	david.wilson@example.com	<input type="checkbox"/>
Laura Garcia	laura.garcia@example.com	<input type="checkbox"/>
James Martinez	james.martinez@example.com	<input type="checkbox"/>
Jessica Anderson	jessica.anderson@example.com	<input checked="" type="checkbox"/>
Thomas Hernandez	thomas.hernandez@example.com	<input type="checkbox"/>

Showing 1 to 10 of 10 results

Per page 10

*Slika 3.27 - Tablica studenata na stranici prilagođenog roka*

Na stranici kolegija nalazi se tablica vrednovanja. Ta tablica omogućuje da voditelj kolegija ima uvid u sve bodove koje su studenti postigli. Ta tablica ima sljedeće stupce: ime studenta, bodovi postignuti na svakom roku, prosjek bodova postignutih na svim rokovima, negativni bodovi dodijeljeni na svakom roku, ukupni negativni bodovi te konačni bodovi. Tablica je prikazana na slici 3.28. Ako je red u tablici crvene boje, to upućuje na to da je taj student neaktivan.

Full name	Rok1 (2024-07-20)	Rok2 (2024-09-20)	Average	Rok1 (negative points)	Rok2 (negative points)	Plocica deadline (negative points)	Total negative points	Final
John Doe	48.795180722892	51.612903225806	50.204041974349	5	0	0	5	45.204
Jane Smith	0	0	0	0	0	0	0	0
Emily Johnson	0	0	0	0	0	0	0	0
Michael Brown	0	0	0	0	0	0	0	0
Sarah Davis	0	0	0	0	0	0	0	0
David Wilson	0	0	0	0	0	0	0	0
Laura Garcia	50.602409638554	0	0	0	0	0	0	0
James Martinez	0	0	0	0	0	0	0	0
Jessica Anderson	50.602409638554	48.387096774194	49.494753206374	0	0	7	7	42.494
Thomas Hernandez	0	0	0	0	0	0	0	0

*Slika 3.28 - Tablica bodova sa vrednovanja*

Omogućeno je horizontalno pomicanje (eng. horizontal scroll) za tablicu što je jako korisno u slučaju da je uneseno puno rokova. Kad je puno rokova tablica može postat jako široka i horizontalno pomicanje čini tablicu preglednom i jednostavnom za koristiti.

Na stranici kolegija nalazi se tablica studenata koja je prikazana na slici 3.29. U toj tablici, voditelj kolegija ima uvid u sve studente na tom kolegiju.

<input type="checkbox"/>	Name	Surname	Email	Active	
<input type="checkbox"/>	John	Doe	john.doe@example.com	✓	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	Jane	Smith	jane.smith@example.com	✓	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	Emily	Johnson	emily.johnson@example.com	✓	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	Michael	Brown	michael.brown@example.com	✓	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	Sarah	Davis	sarah.davis@example.com	✓	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	David	Wilson	david.wilson@example.com	✓	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	Laura	Garcia	laura.garcia@example.com	✗	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	James	Martinez	james.martinez@example.com	✓	<a href="#">Edit</a> <a href="#">Delete</a>

*Slika 3.29 - Tablica studenata*

Tablici se pristupi tako da se klikne na karticu sa tekstom: *Students*. U tablici su prikazani podaci studenata: ime, prezime, elektronička adresa, aktivnost. Samo voditelj kolegija može označiti studenta kao neaktivnog. To se može napraviti tako da se pritisne na gumb sa tekstom: *Edit* čime se pristupa obrascu studenta koji je prikazan na slici 3.30.

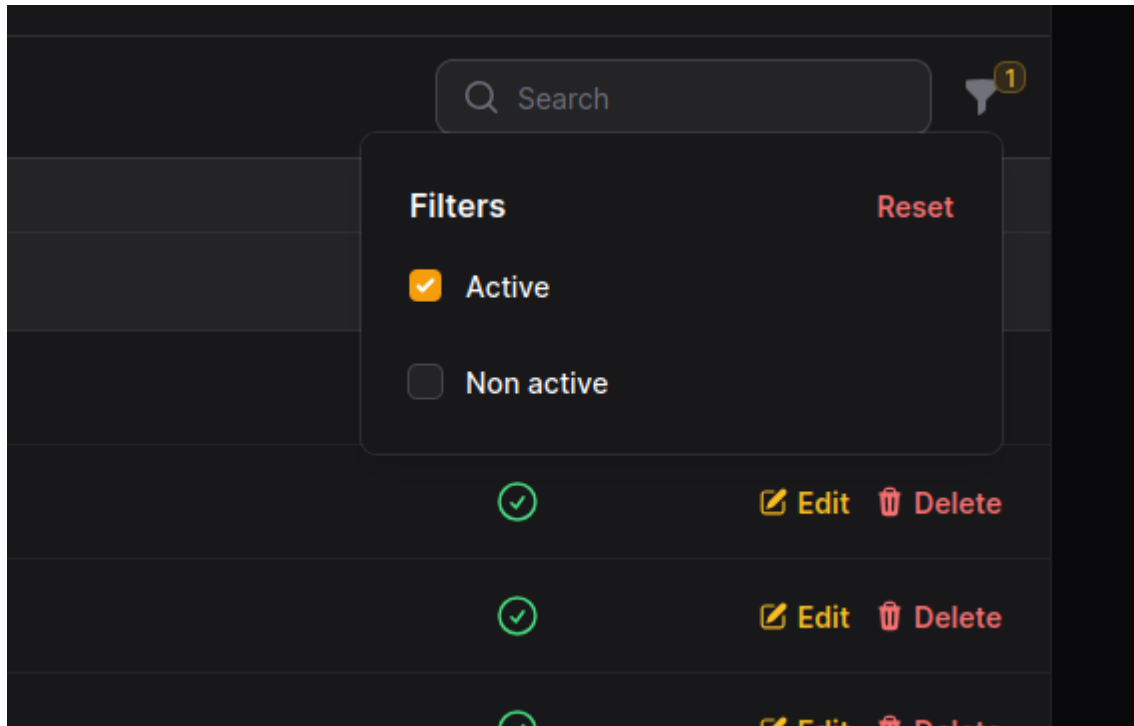


The image shows a dark-themed modal window titled "Edit John" with a close button in the top right corner. The form contains the following fields and controls:

- Name\***: Text input containing "John".
- Surname\***: Text input containing "Doe".
- Email\***: Text input containing "john.doe@example.com".
- Password**: Empty text input field.
- Jmbag**: Text input containing "1234567890".
- Active\***: A toggle switch that is currently turned on (orange).
- Roles**: A dropdown menu showing "Student".
- Deactivation date**: Text input with a date format mask "mm / dd / yyyy , --:--:-- --".
- Buttons**: "Save changes" (orange) and "Cancel" (grey).

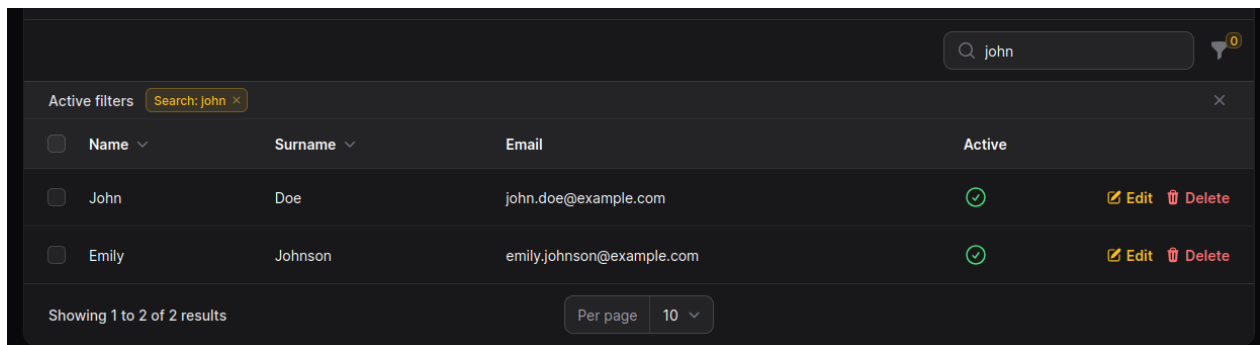
*Slika 3.30 - Obrazac studenta*

Pomoću tog obrasca, voditelj može upravljati podacima studenta i putem polja *Active* mijenjati aktivnost istog. Na vrhu tablice se nalazu dva gumba sa tekstovima: *New user* i *Attach*. Prvi gumb omogućuje stvaranje novog studenta, a drugi gumb omogućuje dodjeljivanje već postojećeg studenta u trenutni kolegij. Uz to, omogućeno je filtriranje studenata po njihovoj aktivnosti. To se može pomoću gumba za filtriranje koji je prikazan na slici 3.31.



*Slika 3.31 - Gumb za filtriranje studenata*

Pored gumba za filtriranje nalazi se i polje za pretraživanje studenata po njihovom imenu i elektroničkoj adresi. Primjer takvog pretraživanja prikazan je na slici 3.32.



*Slika 3.32 - Primjer pretraživanja studenta*

Voditelj kolegija ima uvid u sve projekte na svom kolegiju. U tablici projekata, koja je prikazana na slici 3.33, voditelj ima popis imena svih projekata i gumb koji mu omogućuje stvaranje novog projekta.

Projects		New project
Name		
Mr. Oran Kovacek		More
Mr. Marcel Kling		More
Mr. Sofia Schimmel DVM		More

Showing 1 to 3 of 3 results      Per page 10

*Slika 3.33 - Tablica projekata*

Pritiskom na gumb sa tekstem: *More* voditelj je preusmjeren na stranicu projekta koja je prikazana na slici 3.34. Za razliku od studenta, voditelj kolegija može mijenjati podatke projekta na toj stranici. Voditelj postavlja ime i opis projekta, najveći dopušteni broj studenata na tom projektu, najveći mogući broj bodova koji se može ostvariti na tom projektu te broj postignutih bodova.

Projects > Edit

## Edit Project

About Documentation

Name\*

Mr. Oran Kovacek

Description\*

Sed ut voluptatem qui cum quos nobis. Expedita aliquid et quis earum. Id non voluptatem quia ex qui quod voluptatibus ullam. Eos nesciunt sint est sit ipsum deleniti.

Max Number of Students\*

4

Max Possible Points\*

60

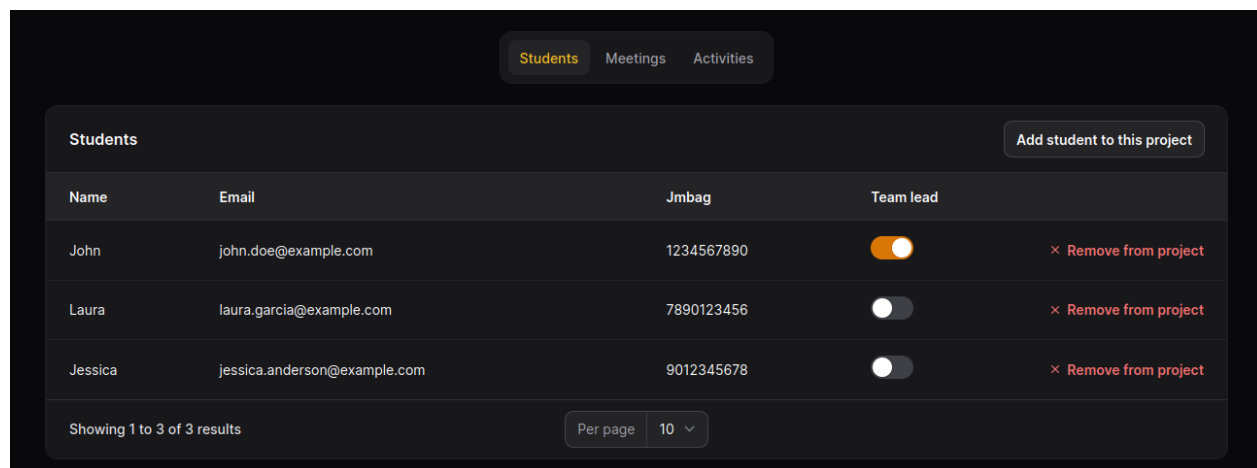
Given Points\*

50

Save changes Cancel

*Slika 3.34 - Stranica projekta*

Ispod obrasca za projekt, nalazi se tablica studenata koji rade na tom projektu. Tablica je prikazana na slici 3.35 i prikazani su stupci: ime, elektronička adresa, jmbag, timski vođa.



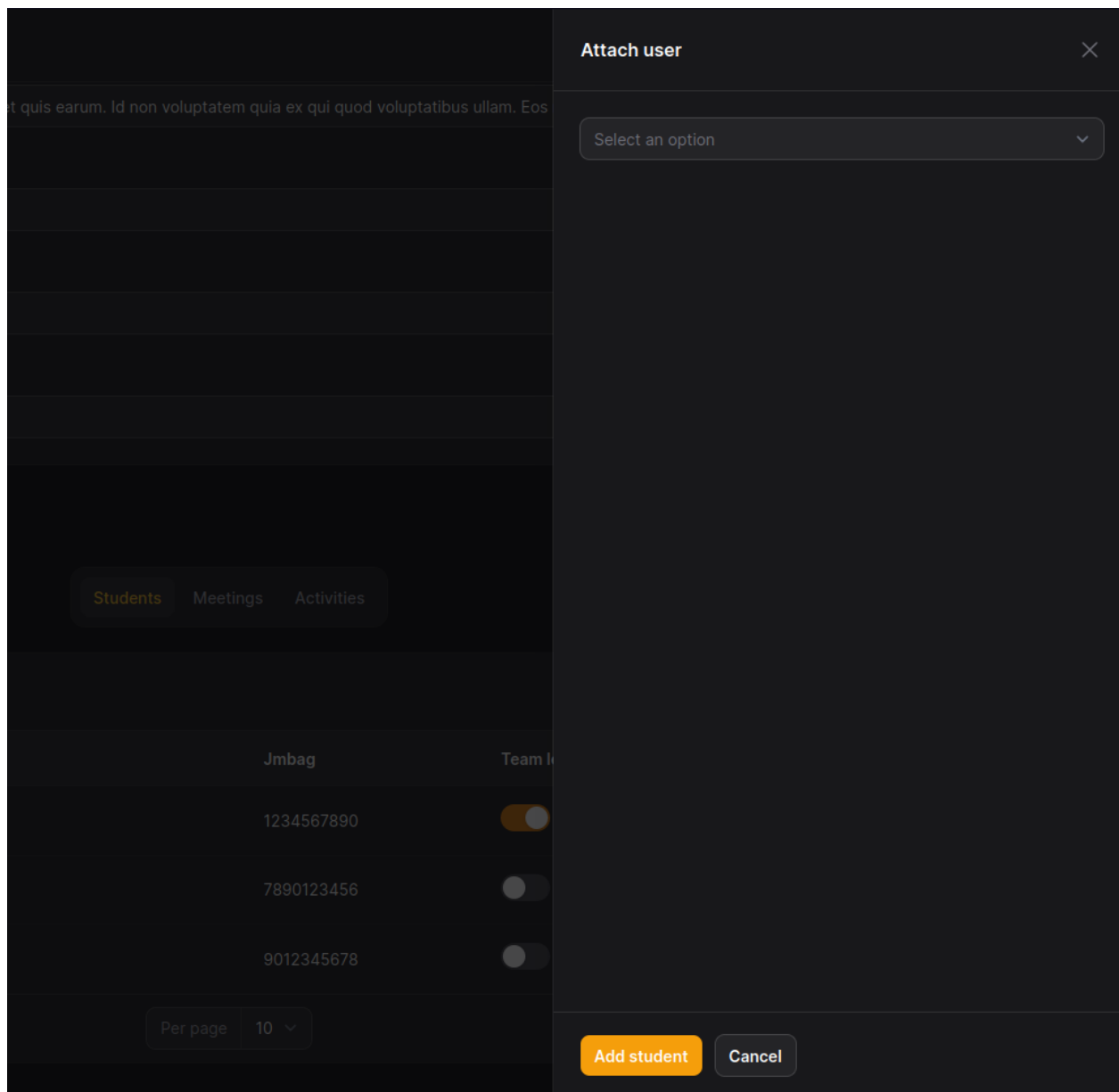
Name	Email	Jmbag	Team lead	
John	john.doe@example.com	1234567890	<input checked="" type="checkbox"/>	<a href="#">× Remove from project</a>
Laura	laura.garcia@example.com	7890123456	<input type="checkbox"/>	<a href="#">× Remove from project</a>
Jessica	jessica.anderson@example.com	9012345678	<input type="checkbox"/>	<a href="#">× Remove from project</a>

Showing 1 to 3 of 3 results

Per page 10

Slika 3.35 - Tablica studenata na projektu

Voditelj označuje vođu tima pomoću gumba u zadnjem stupcu tablice. Voditelj može dodati novog studenta u projekt tako da pritisne gumb sa tekстом: *Add student to this project*. Pritiskom na taj gumb, korisniku se omogućuje da odabere studenta kojeg želi dodati u projekt kako je to pokazano na slici 3.36.



Slika 3.36 - Primjer dodavanja korisnika u projekt

Pritiskom na gumb sa tekstom: *Select an option* otvara se *dropdown* s popisom imena studenata koje voditelj može odabrati i dodati u projekt.

### 3.3 Super admin

*Super admin* je uloga koja je dodijeljena korisniku koji nema nikakvih ograničenja i može upravljati bilo kojim sadržajem u aplikaciji. Većina funkcionalnosti je već spomenuta, ali vrijedi

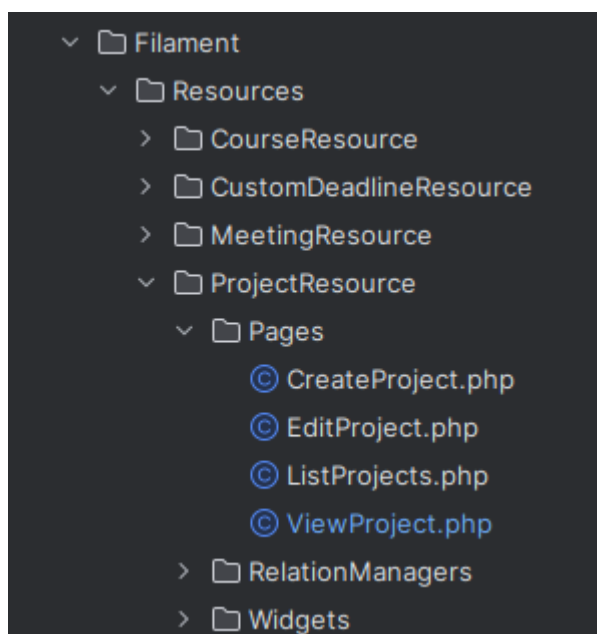
istaknuti da *super admin* ima uvid u sve kolegije i da je on taj koji postavlja početne voditelje kolegija.

## 4 RAZVOJ I IMPLEMENTACIJA SPECIFIČNIH APLIKACIJSKIH FUNKCIONALNOSTI

### 4.1 Dodavanje studenta na projekt

U ovom poglavlju će se objasniti dodavanje studenta na projekt, ali na razini koda. Bit će predstavljen i objašnjen kod sa klijentske i poslužiteljske strane.

Pošto u aplikaciji postoji *Project* model [40] napravljen je i Filament resurs pod imenom *ProjectResource*. Za svaki filament resurs postoje četiri stranice, pa tako za *ProjectResource* postoje sljedeće stranice: *CreateProject*, *EditProject*, *ListProject* i *ViewProject*. Za svaku stranicu postoji specifična klasa. Struktura datoteka koje sadrže te klase je prikazana na slici 4.0.



Slika 4.0 - Struktura datoteka za *ProjectResource*

Unutar datoteke *ViewProject.php* nalazi se klasa *ViewProject*. Pomoću metode *getHeaderWidgets* u toj klasi možemo definirati koji *widget* želimo da bude vidljiv na stranici projekta. Na slici 4.1 je prikazano da metoda *getHeaderWidgets* vraća polje s *AssignToProjectWidget* što znači da će taj *widget* bit vidljiv na stranici projekta.

```

class ViewProject extends ViewRecord
{
    protected static string $resource = ProjectResource::class;

    no usages new *
    public function getHeaderWidgets(): array
    {
        $isStudent = auth()->user()->hasRole( roles: 'Student');
        $projectSelected = false;
        $hasProjectInThisCourse = auth()->user()->projects()->where('course_id', $this->record->course_id)->count() > 0;
        $maxNumberOfStudents = $this->record->max_students <= $this->record->students()->count();

        if ($isStudent) {
            $projectSelected = auth()->user()->projects()->get()->pluck('id')->contains($this->record->id);
        }

        return [
            AssignToProjectWidget::make([
                'studentId' => auth()->user()->id,
                'projectId' => $this->record->id,
                'projectSelected' => $projectSelected,
                'displayWarning' => $hasProjectInThisCourse,
                'maxNumberOfStudents' => $maxNumberOfStudents,
            ]),
        ], You, 05. 08. 2024. 16:37 • Uncommitted changes
    ];
}
}

```

Slika 4.1 - ViewProject klasa

U metodi *getHeaderWidgets* su definirane neke varijable. Varijable *\$isStudent* pohranjuje *boolean* vrijednost. Ako trenutno autentificirani korisnik ima ulogu studenta onda će u toj varijabli biti pohranjena vrijednost *true*, a u protivnom *false*. Varijabla *\$projectSelected* također pohranjuje *boolean* vrijednost. Ta varijabla će poprimiti vrijednost *true* samo u slučaju ako je trenutno autentificirani student već prijavljen na taj projekt. Varijabla *\$hasProjectInThisCourse* će poprimiti vrijednost *true* u slučaju da je trenutni korisnik prijavljen na bilo koji projekt kolegija. Posljednja definirana varijabla je *\$maxNumberOfStudent* koja će poprimiti vrijednost *true* u slučaju da je na projektu prijavljen najveći dopušteni broj studenata. U posljednjem dijelu *getHeaderWidgets* metode se vidi da su vrijednosti svih tih varijabli prenesene *widgetu* koje ta metoda vraća. Vrijednosti tih varijabli će se koristiti kasnije u Blade kodu koji je asociran sa tim *widgetom*. Taj Blade kod je prikazan na slici 4.2.



```

<x-filament-widgets::widget>
  <x-filament::section>
    @if($projectSelected)
      <h3>
        <b>Your selected project</b>   Luka, 23. 06. 2024. 17:26 • assign me to the project button done
      </h3>
    @elseif($displayWarning)
      <p>
        You have already chosen another project for this course.
      </p>
    @elseif($maxNumberOfStudents)
      <p>
        Project has reached maximum number of students.
      </p>
    @else
      <form action='{{ "/projects/{ $projectId }/users/{ $studentId }" }}' method="post">
        @csrf
        <input type="hidden" name="studentId" value="{{ $studentId }}">
        <input type="hidden" name="projectId" value="{{ $projectId }}">
        <button style="background-color: #3b82f6; color: white; font-weight: bold; padding: 8px 16px; border-
          onmouseover="this.style.backgroundColor=' #1e40af'"
          onmouseout="this.style.backgroundColor=' #3b82f6'"
          type="submit">
          Join this project
        </button>
      </form>
    @endif
  </x-filament::section>
</x-filament-widgets::widget>

```

Slika 4.2 - AssignToProjectWidget

U kodu za *AssignToProjectWidget* se vidi da će se na stranici projekta pokazati tekst: „*Your selected project*” kako bi se korisniku sugeriralo da je na stranici onog projekta na kojem radi. U slučaju da varijabla *\$displayWarning* ima vrijednost *true*, a *\$projectSelected* je *false*, to znači da je student izabrao neki drugi projekt te da više ne može odabrati projekt koji je trenutno prikazan. Iz tog razloga će se na stranici ispisati tekst: *You have already chosen another project for this course*. U slučaju da varijabla *\$maxNumberOfStudents* ima vrijednost *true* to znači da je projekt već dosegao najveći dopušteni broj studenata i da se zbog toga korisnik više ne može pridružiti prikazanom projektu. Posljednji slučaj je slučaj u kojem se student može pridružiti projektu i u tom slučaju je prikazan gumb sa tekstom: *Join this project* Taj gumb je, u kodu, dio *form* elementa. *Form* element omogućuje da se definira ruta na koju će se poslati zahtjev, omogućuje da se definira tip zahtjeva koji se šalje i u ovom slučaju, unutar *forma* je Blade direktiva koja služi za zaštitu od CSRF napada. U ovom slučaju, kad korisnik stisne gumb, šalje se POST zahtjev na rutu: */projects/{projectId}/users/{studentId}*.

Rute su definirane u *routes* mapi laravel projekta, a ruta na koju dolazi zahtjev za dodavanje studenta na projekt nalazi se u datoteci *web.php* kao što je to prikazano na slici 4.3.

```
Route::middleware('auth')->group(function () {  
    Route::post('projects/{project}/users/{user}', [ProjectController::class, 'assignStudentToProject'])->name('assign-student-to-project');  
});
```

*Slika 4.3 - Ruta za dodavanje studenta na projekt*

U kodu, sa slike 2.2, se vidi da je ruta osigurana od strane *auth middlewarea*. To znači da su za tu rutu dopušteni samo oni zahtjevi koji dolaze od strane autenticiranih korisnika. Kad je zahtjev poslan na tu rutu onda se poziva metoda *assignStudentToProject*. Ta metoda se nalazi u *ProjectController* klasi koja je prikazana na slici 4.4.

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Project;
6  use App\Models\User;
7
8  class ProjectController extends Controller
9  {
10     public function assignStudentToProject(Project $project, User $user)
11     {
12         $project->students()->attach($user);
13
14         return redirect()->back();
15     }
16 }
17

```

Slika 4.4 - ProjectController

Metoda *assignStudentToProject* prima dva argumenta: *Project* model i *User* model. ID projekta i ID studenta definirani su u zahtjevu, a Laravel u pozadini pronalazi projekt i korisnika s odgovarajućim vrijednostima ID polja. Zatim se taj projekt i taj korisnik ubacuju kao argumenti pri pozivu metode *assignStudentToProject*. U tijelu *assignStudentToProject* metode, projektu se dodjeljuje student tako što se nad instancom *Project* modela poziva metoda *students*. Ta *students* metoda označava Eloquent vezu [41] između modela. Ta veza je prikazana na slici 4.5, označava *many to many* oblik veze [42] i definirana je unutar *Project* modela.

```

class Project extends Model
{
    use HasFactory;

    no usages
    protected $guarded = [];

    public function course(): BelongsTo
    {
        return $this->belongsTo(related: Course::class, foreignKey: 'course_id');
    }

    public function students(): BelongsToMany
    {
        return $this->belongsToMany(related: User::class, table: 'project_user', foreignPivotKey: 'project_id', relatedPivotKey: 'user_id')->withPivot( columns: 'team_lead');
    }

    no usages
    public function meetings(): HasMany
    {
        return $this->hasMany(related: Meeting::class, foreignKey: 'project_id');
    }

    no usages
    public function activities(): HasMany
    {
        return $this->hasMany(related: Activity::class, foreignKey: 'project_id');
    }
}

```

Slika 4.5 - Project model

U klasi *ProjectController*, unutar metode *assignStudentToProject*, student se projektu dodjeljuje na vrlo čitljiv način pozivom metode *attach*. Posljednja linija koda u *assignStudentToProject* metodi, odgovorna je za preusmjeravanje korisnika natrag na prethodnu stranicu nakon što se izvrši akcija dodavanja studenta projektu. Metoda *redirect* je dio Laravelovog mehanizma za upravljanje HTTP odgovorima i koristi se za stvaranje odgovora koji preusmjerava korisnika na drugu URL adresu. Dodatna metoda *back* specificira da se korisnik treba vratiti na stranicu s koje je poslao zahtjev. Ovo je korisno u formama gdje se želi, nakon što se podaci uspješno obrade, omogućiti korisniku nastavak s prethodnim aktivnostima bez gubitka konteksta navigacije.

## 4.2 Prijenos csv datoteke sa podacima studenata

Aplikacija omogućuje da voditelj kolegija prenese csv datoteku koja sadrži sve podatke o studentima koji pohađaju njegov kolegij. Nakon prijenaosa te datoteke, ti studenti su stvoreni u bazi podataka i omogućen im je pristup aplikaciji. U ovom poglavlju će ta funkcionalnost biti objašnjena na razini koda.

Kako bi korisnicima bilo omogućeno da upravljaju sa podacima kolegija, stvoren je Filament resurs za kolegije *CourseResource*. Svaki resurs ima *form* metodu u kojoj se definiraju polja obrasca. Polja za obrazac kolegija je prikazan na slici 4.6.

```
public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\TextInput::make('name')
                ->required()
                ->disabled(auth()->user()->hasRole(roles: 'Student')),
            Forms\Components\Select::make('admins')
                ->label('Course Admin')
                ->searchable()
                ->preload()
                ->disabled(auth()->user()->hasRole(roles: 'Student'))
                ->relationship(name: 'admins', titleAttribute: 'name')
                ->multiple(),
            Forms\Components\FileUpload::make('student_file')
                ->disk(name: 'student-files')
                ->label('Upload Student Files')
                ->acceptedFileTypes(['text/csv', 'text/plain'])
                ->disabled(auth()->user()->hasRole(roles: 'Student')),
        ]);
}
```

Slika 4.6 - form metoda u *CourseResource*

U kodu je definirano tekstualno polje za unos imena kolegija, izbornik za odabir voditelja te polje za prijenos CSV datoteke. Korištenjem metode *disabled* na tim poljima, ona su blokirana za korisnike s ulogom studenta. Pri prijenosu, datoteka se sprema u mapu *storage*, dok se ime datoteke pohranjuje u bazi podataka, unutar tablice *courses*. *CourseObserver* je klasa stvorena za promatranje događaja poput stvaranja novog kolegija, izmjene postojećeg, brisanja kolegija i slično. U ovoj aplikaciji, *observer* se koristi za kreiranje novih studenata iz CSV datoteke. Taj proces kreiranja studenata pokreće se stvaranjem novog kolegija s pridruženom CSV datotekom ili promjenom imena CSV datoteke u postojećem kolegiju, što implicira da je voditelj kolegija prenio

novu datoteku. Na slici 4.7 su prikazane metode koje se pokreću kada se novi kolegij kreira, odnosno kada se postojeći mijenja.

```
class CourseObserver
{
    /**
     * Handle the Course "created" event.
     */
    no usages  📄 Luka
    public function created(Course $course): void
    {
        if ($course->student_file === null) {
            return;
        }

        $this->createNewStudents($course);
    }

    /**
     * Handle the Course "updated" event.
     */
    no usages  📄 Luka
    public function updated(Course $course): void
    {
        if (! $course->isDirty( attributes: 'student_file')) {
            return;
        }

        $this->createNewStudents($course);
    }
}
```

Slika 4.7 - CourseObserver metode

U *updated* metodi, pomoću metode *isDirty*, provjerava se mijenja li se *student\_file* polje, jer se kreiranje novih studenata pokreće samo ako dođe do promjene tog specifičnog polja. Ako se mijenja vrijednost nekog drugog polja, proces kreiranja novih studenata neće biti pokrenut. Metoda koja je zadužena za kreiranje novih studenata je prikazana na slici 4.8.

```

public function createNewStudents(Course $course): void    Luka, 17. 06. 2024. 01:27 *
{
    if (! Storage::disk( name: 'student-files')->exists($course->student_file)) {
        return;
    }

    $fileContents = Storage::disk( name: 'student-files')->get($course->student_file);

    $csvService = new CsvService($fileContents);
    $csvService->validate();

    foreach ($csvService->getItems() as $student) {

        if ($course->students()->count() === $course->max_students) {
            return;
        }

        $student = User::create([
            'name' => $student['name'],
            'surname' => $student['surname'],
            'email' => $student['email'],
            'jmbag' => $student['jmbag'],
            'password' => Hash::make(config( key: 'students.default-password')),
        ]);

        $student->assignRole('Student');
        $student->attendingCourse()->attach($course);
    }
}

```

Slika 4.8 - createNewStudents metoda

U toj metodi prvo se dohvaća sadržaj CSV datoteke, a zatim se taj sadržaj predaje kao argument klasi *CsvService*. Ova servisna klasa zadužena je za validaciju prisutnih polja u CSV datoteci te omogućuje elegantno dohvaćanje svakog retka iz sadržaja CSV datoteke pomoću metode *getItems*. U konfiguracijskoj datoteci *students.php*, koja je prikazana na slici 4.9, definirana su sva polja koja su obavezna da budu prisutna u CSV datoteci. *CsvService* provjerava jesu li ta specifična polja zaista prisutna u sadržaju datoteke.

```
return [  
  
    /*  
    |-----  
    | Student attributes from student csv file  
    |-----  
    |  
    | This attributes should be present in csv file when course manager imports  
    | new students. Keys represent the name of the attribute and values are actual  
    | titles in the first row of csv file.  
    |  
    */  
  
    'csv-attributes' => [  
        'name' => 'ime',  
        'surname' => 'prezime',  
        'email' => 'email',  
        'jmbag' => 'jmbag',  
    ],  
],
```

Slika 4.9 - Sadržaj datoteke *students.php*

Unutar *createNewStudents* metode, nakon validacije sadržaja datoteke, u petlji se prolazi kroz svaki redak CSV datoteke i za taj redak se kreira novi korisnik. Tom korisniku se dodjeljuje uloga studenta te ga se dodjeljuje na kolegij pozivom metode *attach* koja kao argument prima instancu *Course* modela.



## 5 ZAKLJUČAK

U ovom završnom radu predstavljena je *web* aplikacija namijenjena upravljanju projektima za kolegije unutar akademskih ustanova. Aplikacija omogućava studentima da se prijavljuju na projekte, vrednuju svoj rad i rad svojih kolega, te prate napredak kroz različite faze projekta. Za izradu aplikacije korištene su moderne tehnologije kao što su Laravel za poslužiteljski dio i HTML, CSS te Blade za klijentski dio. Također, korišten je Docker za olakšavanje razvoja i implementacije, te Git za upravljanje verzijama koda.

Ove tehnologije su se pokazale kao izuzetno efikasne u razvoju *web* aplikacija. Laravel, s njegovim bogatim ekosustavom i jasnom dokumentacijom, omogućio je brz i organiziran razvoj, dok su HTML, CSS i Blade pružili fleksibilno i prilagodljivo korisničko sučelje koje poboljšava iskustvo korisnika. Docker je značajno pojednostavio distribuciju aplikacije, a Git je omogućio lako upravljanje izmjenama i verzijama koda.

Trenutno, aplikacija sadrži stranice za pregled na kojima se nalaze obrasci s poljima koja su zaključana za uređivanje korisnicima. U budućnosti, izgled tih stranica mogao bi se dodatno unaprijediti implementacijom Filament infolista [43]. Time bi stranice za pregled postale preglednije, a kod bi bio elegantniji i učinkovitiji. Također, implementacija autentikacije pomoću edu.hr računa mogla bi dodatno olakšati proces prijave i povećati sigurnost aplikacije. Ove nadogradnje bi značajno doprinijele funkcionalnosti aplikacije, čineći je još pristupačnijom i sigurnijom za krajnje korisnike.

## LITERATURA

- [1] „Laravel 10“, s interneta, <https://laravel.com/docs/10.x>, 21. srpnja 2024.
- [2] „Blade“, s interneta, <https://laravel.com/docs/10.x/blade>, 21. srpnja 2024.
- [3] „Dinamički pogledi (Views)“, s interneta, <https://laravel.com/docs/11.x/views>, 21. srpnja 2024.
- [4] „Filament“, s interneta, <https://filamentphp.com/docs>, 21. srpnja 2024.
- [5] „Docker“, s interneta, <https://docs.docker.com/guides/docker-overview/>, 23. srpnja 2024.
- [6] „Git“, s interneta, <https://git-scm.com/>, 23. srpnja 2024.
- [7] „HTML“, s interneta, <https://developer.mozilla.org/en-US/docs/Web/HTML>, 28. srpnja 2024.
- [8] „World Wide Web Consortium“, s interneta, <https://www.w3.org/>, 28. srpnja 2024.
- [9] „CSS“, s interneta, <https://developer.mozilla.org/en-US/docs/Web/CSS>, 28. srpnja 2024.
- [10] „Blade direktive“, s interneta, <https://laravel.com/docs/11.x/blade#blade-directives>, 28. srpnja 2024.
- [11] „Laravel pogledi“, s interneta, <https://laravel.com/docs/11.x/views>, 28. srpnja 2024.
- [12] „Sesija“, s interneta, <https://laravel.com/docs/11.x/session>, 28. srpnja 2024.
- [13] „Cache“, s interneta, <https://laravel.com/docs/11.x/cache>, 28. srpnja 2024.
- [14] „Sekcijske direktive“, s interneta, <https://laravel.com/docs/11.x/blade#section-directives>, 28. srpnja 2024.
- [15] „CSRF napad“, s interneta, <https://owasp.org/www-community/attacks/csrf>, 28. srpnja 2024.
- [16] „CSRF direktiva“, s interneta, <https://laravel.com/docs/11.x/blade#csrf-field>, 28. srpnja 2024.
- [17] „Rutiranje“, s interneta, <https://laravel.com/docs/11.x/routing>, 28. srpnja 2024.
- [18] „Design patterns“, s interneta, <https://designpatternsphp.readthedocs.io/en/latest/>, 28. srpnja 2024.
- [19] „Kontejner usluga“, s interneta, <https://laravel.com/docs/11.x/container>, 29. srpnja 2024.

- [20] „Fasadni obrazac", s interneta, <https://designpatternsphp.readthedocs.io/en/latest/Structural/Facade/README.html>, 29. srpnja 2024.
- [21] „Promatrački obrazac", s interneta, <https://designpatternsphp.readthedocs.io/en/latest/Behavioral/Observer/README.html>, 29. srpnja 2024.
- [22] „Laravel Pint", s interneta, <https://laravel.com/docs/11.x/pint>, 29. srpnja 2024.
- [23] „MySQL", s interneta, <https://www.oracle.com/mysql/what-is-mysql>, 29. srpnja 2024.
- [24] „MySQL transakcije", s interneta, <https://www.tutorialspoint.com/mysql/mysql-transactions.htm>, 29. srpnja 2024.
- [25] „Laravel Eloquent", s interneta, <https://laravel.com/docs/11.x/eloquent>, 29. srpnja 2024.
- [26] „Filament", s interneta, <https://filamentphp.com/docs/3.x/panels/installation>, 29. srpnja 2024.
- [27] „Filament panel", s interneta, <https://filamentphp.com/docs/3.x/panels/getting-started#overview>, 29. srpnja 2024.
- [28] „Stats overview widgets", s interneta, <https://filamentphp.com/docs/3.x/widgets/stats-overview>, 29. srpnja 2024.
- [29] „Artisan komande", s interneta, <https://laravel.com/docs/11.x/artisan>, 29. srpnja 2024.
- [30] „Chart widget", s interneta, <https://filamentphp.com/docs/3.x/widgets/charts>, 30. srpnja 2024.
- [31] „Tailwind", s interneta, <https://tailwindcss.com/>, 30. srpnja 2024.
- [32] „Filament tablice", s interneta, <https://filamentphp.com/docs/3.x/tables/getting-started>, 30. srpnja 2024.
- [33] „Filament akcije", s interneta, <https://filamentphp.com/docs/3.x/actions/installation>, 30. srpnja 2024.
- [34] „laravel-permission", s interneta, <https://spatie.be/docs/laravel-permission/v6/introduction>, 30. srpnja 2024.
- [35] „Docker", s interneta, <https://aws.amazon.com/docker/>, 30. srpnja 2024.
- [35] „Docker compose", s interneta, <https://docs.docker.com/compose/>, 30. srpnja 2024.

- [36] „Laravel Sail", s interneta, <https://laravel.com/docs/11.x/sail#introduction>, 30. srpnja 2024.
- [37] „Redis", s interneta, <https://laravel.com/docs/11.x/redis>, 30. srpnja 2024.
- [38] „Git", s interneta, <https://git-scm.com/about>, 30. srpnja 2024.
- [39] „pcb", s interneta, [https://en.wikipedia.org/wiki/Printed\\_circuit\\_board](https://en.wikipedia.org/wiki/Printed_circuit_board), 11. kolovoza 2024.
- [40] „Model", s interneta, <https://laravel.com/docs/11.x/eloquent#generating-model-classes>, 11. kolovoza 2024.
- [41] „Eloquent veza", s interneta, <https://laravel.com/docs/11.x/eloquent-relationships#defining-relationships>, 11. kolovoza 2024.
- [42] „Many to many", s interneta, <https://laravel.com/docs/11.x/eloquent-relationships#many-to-many>, 11. kolovoza 2024.
- [43] „Filament infolist", s interneta, <https://filamentphp.com/docs/3.x/infolists/getting-started>, 11. kolovoza 2024.

## SAŽETAK

U ovom radu razvijena je web aplikacija koja omogućava praćenje rada i aktivnosti studenata na projektima unutar različitih kolegija. Aplikacija omogućuje studentima da se prijavljuju na projekte, prate svoje zadatke, kao i ocjenjuju rad i doprinos svojih kolega kroz interaktivne obrasce. Implementacija ove aplikacije doprinosi boljoj organizaciji i transparentnosti u akademskim projektima, te omogućava voditeljima kolegija efikasnije upravljanje studentskim aktivnostima i projektima. Za izgradnju aplikacije korištene su tehnologije kao što su Laravel za poslužiteljski dio aplikacije, Blade za klijentski dio aplikacije i Filament za upravljanje administrativnim sučeljem, što je osiguralo pouzdanu i skalabilnu platformu. Korištenje ovih tehnologija pruža fleksibilnost, visoku razinu prilagodljivosti i olakšava daljnji razvoj aplikacije.

Ključne riječi: Web, aplikacija, praćenje projekata, akademski projekti, Laravel, Blade, Filament, Docker

## ABSTRACT

This thesis develops a web application designed to monitor the work and activities of students on projects within various courses. The application enables students to enroll in projects, track their tasks, and evaluate the work and contributions of their peers through interactive forms. The implementation of this application contributes to better organization and transparency in academic projects and allows course leaders to manage student activities and projects more effectively. Technologies such as Laravel for the backend, Blade for the frontend, and Filament for administrative interface management were used in building the application, ensuring a reliable and scalable platform. The use of these technologies provides flexibility, a high level of customizability, and facilitates further development of the application.

Keywords: web, application, project tracking, academic projects, Laravel, Blade, Filament, Docker