

# Web aplikacija za upravljanje završnih i diplomskih radova (e-radovi)

---

**Domić, Filip**

**Undergraduate thesis / Završni rad**

**2019**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Pula / Sveučilište Jurja Dobrile u Puli**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:137:124935>

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

*Download date / Datum preuzimanja:* **2025-03-20**



*Repository / Repozitorij:*

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli  
Fakultet informatike u Puli

**FILIP DOMIĆ**

**Web aplikacija za upravljanje završnim i diplomskim radovima  
(e-radovi)**

Završni rad

Pula, rujan, 2019.

Sveučilište Jurja Dobrile u Puli  
Fakultet informatike u Puli

**FILIP DOMIĆ**

**Web aplikacija za upravljanje završnim i diplomskim radovima  
(e-radovi)**

Završni rad

**JMBAG: 0303063538, izvanredni studij**

**Studijski smjer: Informatika**

**Predmet: Napredne tehnike programiranja**

**Znanstveno područje: Društvene znanosti**

**Znanstveno polje: Informacijske i komunikacijske znanosti**

**Znanstvena grana: Informacijski sustavi i informatologija**

**Mentor: doc. dr. sc. Tihomir Orehovački**

Pula, rujan, 2019.



## IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani FILIP DOMIĆ, kandidat za prvostupnika INFORMATIKE ovime izjavljujem da je ovaj Završni rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

---

U Puli, \_\_\_\_\_, \_\_\_\_\_ godine



## **IZJAVA o korištenju autorskog djela**

Ja, FILIP DOMIĆ dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom Web aplikacija za upravljanje završnim i diplomskim radovima(e radovi) koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, \_\_\_\_\_ (datum)

Potpis

---

# SADRŽAJ

<b>1. Uvod.....</b>	<b>1</b>
<b>2. Use-case dijagram aplikacije .....</b>	<b>2</b>
2.1. Studenti .....	2
2.2. Profesori .....	3
2.3. Referada .....	3
<b>3. Planiranje projekta .....</b>	<b>4</b>
3.1. Planiranje prije početka .....	4
3.2. Praćenje razvoja projekta.....	5
<b>4. Tehnologije razvoja .....</b>	<b>6</b>
4.1. Klijentski razvoj.....	6
4.2. Serverski razvoj.....	6
<b>5. Produkcijaska okolina.....</b>	<b>7</b>
<b>6. Ovlasti u aplikaciji .....</b>	<b>8</b>
6.1. Referada .....	8
6.2. Profesor .....	8
6.3. Student .....	8
<b>7. Pristupni podaci.....</b>	<b>9</b>
<b>8. Aplikacija.....</b>	<b>10</b>
8.1. Prijava na stranicu.....	10
8.2. Upravljanje fakultetima .....	11
8.3. Upravljanje odjelima .....	14
8.4. Upravljanje djelatnicima .....	16
8.5. Upravljanje studentima.....	19
8.6. Upravljanje ponuđenim temama.....	21
8.7. Upravljanje radovima.....	23
8.8. Rezervacija ponuđene teme .....	26
8.9. Upravljanje odlukama .....	28
8.10. Upravljanje kronologijom.....	30

<b>9. Hodogram kroz aplikaciju .....</b>	<b>34</b>
<b>9.1. Kreiranje studenta (Referada) .....</b>	<b>34</b>
<b>9.2. Kreiranje ponuđenih teme (Profesor) .....</b>	<b>35</b>
<b>9.3. Prijava rada (Student) .....</b>	<b>36</b>
9.3.1. Ponuđena tema .....	36
9.3.2. Unošenje vlastite teme.....	36
<b>9.4. Odobravanje rada (Profesor).....</b>	<b>37</b>
<b>9.5. Komentiranje rada (Student).....</b>	<b>38</b>
<b>10. Programski isjetci.....</b>	<b>39</b>
<b>10.1. Autetifikacija.....</b>	<b>39</b>
<b>10.2. Dodaj Fakultet.....</b>	<b>40</b>
<b>10.3. Kronologija.....</b>	<b>42</b>
<b>11. Prijedlozi za poboljšanja .....</b>	<b>44</b>
<b>12. Zaključak .....</b>	<b>45</b>
<b>13. Popis literature.....</b>	<b>46</b>

# 1. Uvod

Dizajn i kreacija ovoga rada rezultat su ideje o olakšanju i ubrzanju procedure prijavljivanja i primopredaje završnih i diplomskih radova. U svijetu aplikacija studenti i nastavnici lako se nalaze, komunikacija do sada već se obavlja putem elektroničke pošte, ova aplikacija nije ništa kompliciranija. Tradicionalno postupak odabira teme rada, pisanje formulara i ponovno popunjavanje istog, u slučaju da je odbijen, čini se kao faktor koji usporava sam početak pisanja rada. Prilikom odabira teme rada student ima izbor odabira ponuđenih tema ili smišljanja svoje teme. Ako se student odluči potražiti teme među ponuđenima, rezervacija teme je potrebna. Informacije o ponuđenim temama nalaze se na stranici fakulteta. Student u tom trenutku nije jedini koji je napravio rezervaciju, ali sigurno treba biti prvi. Nakon uspješnog odabira kreće sljedeća procedura. Formular za prijavu teme rada nalazi se u skriptarnici i potrebno ga je pokupiti. Odabranu temu treba odobriti profesor putem predanog i ispunjenog formulara od strane studenta, koji zatim referada provlači kroz sustav. Nije rijedak slučaj da je postupak potrebno ponoviti. Ako je tema rada odbijena od strane profesora. Student je smislio ili odabrao temu, obavijestio profesora te mu je profesor odobrio temu, zatim je formular preuzet u skriptarnici, ispunjen od strane studenta, pregledan od strane profesora i provučen kroz sustav od strane referade. Student sada može početi pisati rad. Aplikacije na mobitelima su uvijek ažurne, postavlja se pitanje može li biti i procedura potrebna za pisanje završnog i diplomskog rada u korak s vremenom. Može, ako se stavi u okruženje u kojem će uvijek biti ažurna, pretvori u aplikaciju koja znatno skraćuje potrebno vrijeme za odabir teme, pisanje rada i svake popratne procedure.<sup>1</sup> Aplikacija koja je osmišljena da bude glavna tema ovog rada mogla bi biti moguće rješenje. Tema kao i sam završni ili diplomski rad može se slati putem aplikacije. Profesor i student imaju mogućnost komentiranja raznih verzija poslanog završnog rada. Referada, kao administrator aplikacije ima pristup svim radovima unutar aplikacije, a ako je potrebno sve izmjene za dodatno ubrzanje ovog procesa lagano su izvedive u obliku sljedećih ažuriranja aplikacije. Detaljnije objašnjen postupak rada s aplikacijom i njezine mogućnosti nalaze se u nastavku rada.

---

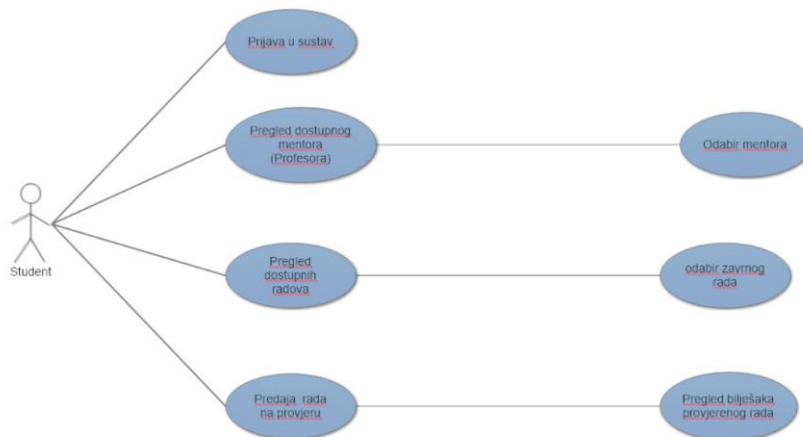
<sup>1</sup> Barać, A. Razvoj dinamičkih web aplikacija, 2014. diplomski rad, Zagreb, dostupno na: <https://www.pfri.uniri.hr/knjiznica/NG-dipl.EITP/219-2014.pdf> (03.09.2019)



## 2. Use-case dijagram aplikacije

### 2.1. Studenti

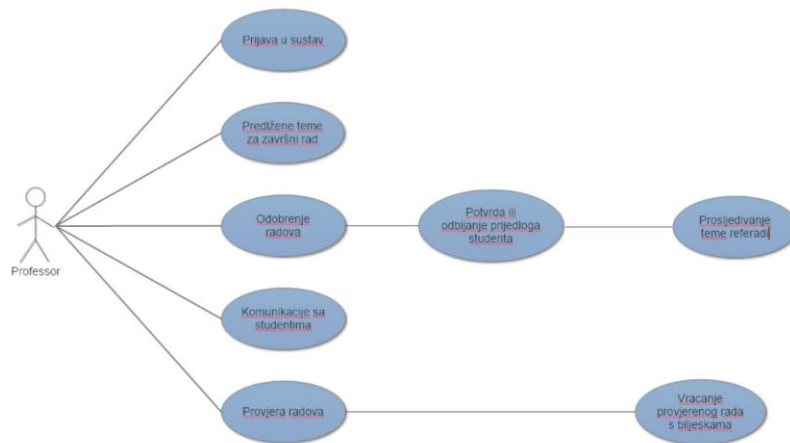
Slika 1. prikazuje use-case za studenta. Student ima mogućnost prijaviti se u sustav, student može vidjeti dostupne mentorove teme odnosno kako se u sustavu zovu ponuđene teme ili predložiti mentoru neku svoju temu. Najznačajnija funkcionalnost aplikacije sa studentove strane je predaja rada na provjeru i komunikacija s profesorom.



Slika 1. Use-case Student

## 2.2. Profesori

Slika 2. prikazuje use-case za profesora. Profesor ima mogućnost prijaviti se u sustav, te može napraviti predložke za radove. Također profesor ima mogućnost prihvaćanja ili odbijanja teme koju je predložio neki student. Omogućena je komunikacija sa studentom preko aplikacije i profesor ima mogućnost preuzimati verzije završnih radova, te ima mogućnost ostaviti neki svoj komentar.



Slika 2. Use-case Profesor

## 2.3. Referada

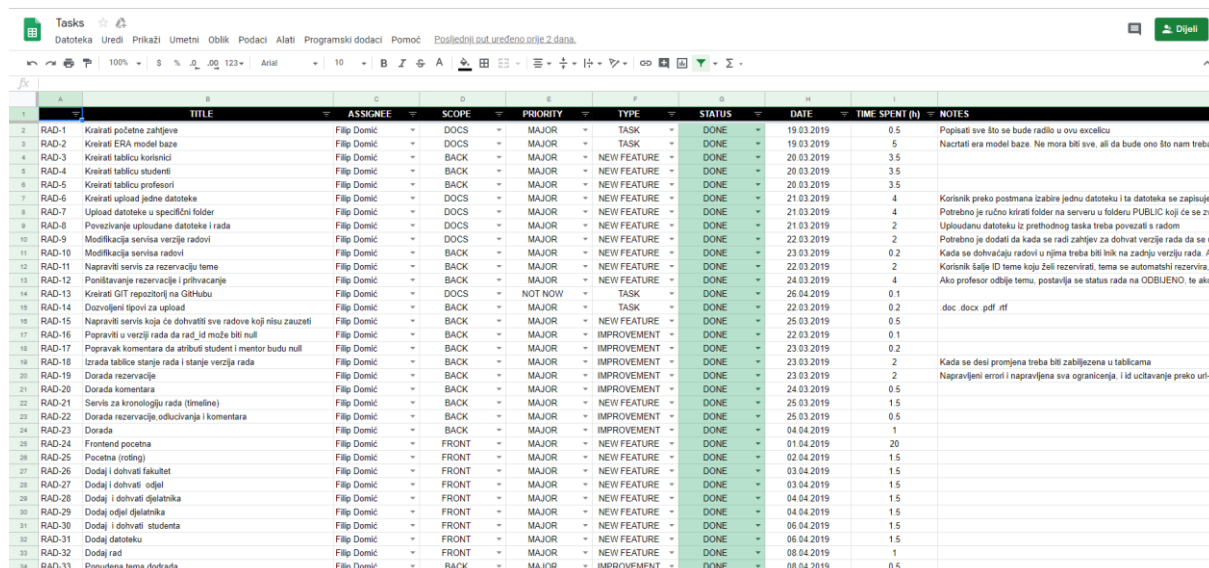
Use-case dijagram za referadu nije zasebno kreiran jer referada se smatra super korisnikom koji ima kombinaciju svih mogućnosti koje su dostupne profesorima i studentima.

### 3. Planiranje projekta

Prva faza izrada bilo kojeg projekta je samo planiranje projekta. Vrlo je važno u samome početku napraviti dobar plan kako će taj sami projekt izgledati i koje ciljeve želimo postići. Vrlo je bitno postaviti si neka od jednostavnijih pitanja kao npr. što je potrebno napraviti? Na koji način ćemo to napraviti? Koliko će koštati projekt? Koliko je vremena potrebno da se projekt realizira? I dali je uopće moguće taj projekt napraviti. Ako zanemarimo neke od ovih pitanja, ako loše isplaniramo sam koncept projekta ili pak u potpunosti preskočimo fazu planiranja, ovaj projekt zasigurno ulazi u postotak projekata koji se započnu a nikada ne završe. <sup>2</sup>

#### 3.1. Planiranje prije početka

Projekt „Web aplikacija za prijavu završnih i diplomskih radova“ u fazi planiranja započinje sa sa raspisivanjem zadataka. Za potrebe raspisa zadataka korištena je excel tablica unutar koje se program dijelio na faze razvoja. Kao što je i prikazano na slici 3. Podjela projekta na faze razvoja.



ID	TITLE	ASSIGNEE	SCOPE	PRIORITY	TYPE	STATUS	DATE	TIME SPENT (h)	NOTES	
2	RAD-1	Kirati početne zahtjeve	Filip Domic	DOCS	MAJOR	TASK	DONE	19.03.2019	0.5	Popisati sve što se bude radilo u ovoj excelici
3	RAD-2	Kirati ERA model baze	Filip Domic	DOCS	MAJOR	TASK	DONE	19.03.2019	5	Nacrtati era model baze. Ne mora biti sve, ali da bude ono što nam treba
4	RAD-3	Kirati tablicu korisnici	Filip Domic	BACK	MAJOR	NEW FEATURE	DONE	20.03.2019	3.5	
5	RAD-4	Kirati tablicu studenti	Filip Domic	BACK	MAJOR	NEW FEATURE	DONE	20.03.2019	3.5	
6	RAD-5	Kirati tablicu profesori	Filip Domic	BACK	MAJOR	NEW FEATURE	DONE	20.03.2019	3.5	
7	RAD-6	Kirati upload jedne datoteke	Filip Domic	DOCS	MAJOR	NEW FEATURE	DONE	21.03.2019	4	Korisnik preko postama izabire jednu datoteku i ta datoteka se zapisuje
8	RAD-7	Upload datoteke u specifični folder	Filip Domic	DOCS	MAJOR	NEW FEATURE	DONE	21.03.2019	4	Potrebno je ručno kirirati folder na serveru u folderu PUBLIC koji će se zv
9	RAD-8	Povezivanje uploadane datoteke i rada	Filip Domic	DOCS	MAJOR	NEW FEATURE	DONE	21.03.2019	2	Uploadanu datoteku iz prethodnog taska treba povezati s radom
10	RAD-9	Modifikacija servisa verzije radovi	Filip Domic	DOCS	MAJOR	NEW FEATURE	DONE	22.03.2019	2	Potrebno je dodati da kada se radi zahtjev za dohvat verzije rada da se u
11	RAD-10	Modifikacija servisa radovi	Filip Domic	BACK	MAJOR	NEW FEATURE	DONE	23.03.2019	0.2	Kada se dohvaćaju radovi u njima treba biti link na zahtjev verzije rada. A
12	RAD-11	Napravit servis za rezervaciju teme	Filip Domic	BACK	MAJOR	NEW FEATURE	DONE	22.03.2019	2	Korisnik šalje ID teme koju želi rezervirati, tema se automatski rezervira
13	RAD-12	Ponikavanje rezervacije i prihvatanje	Filip Domic	BACK	MAJOR	NEW FEATURE	DONE	24.03.2019	4	Ako profesor odbije temu, postavlja se status rada na ODBIJENO, te ako
14	RAD-13	Kirati GIT repozitorij na GitHubu	Filip Domic	DOCS	NOT NOW	TASK	DONE	26.04.2019	0.1	
15	RAD-14	Dovravljeni tipovi za upload	Filip Domic	BACK	MAJOR	TASK	DONE	22.03.2019	0.2	doc docx pdf rtf
16	RAD-15	Napravit servis koja će dohvatiti sve radove koji nisu zauzeti	Filip Domic	BACK	MAJOR	NEW FEATURE	DONE	25.03.2019	0.5	
17	RAD-16	Popraviti u verziji rada da rad_id može biti null	Filip Domic	BACK	MAJOR	IMPROVEMENT	DONE	22.03.2019	0.1	
18	RAD-17	Popravlak komentara da atribut student i mentor budu null	Filip Domic	BACK	MAJOR	IMPROVEMENT	DONE	23.03.2019	0.2	
19	RAD-18	Izrada tablice stanje rada i stanje verzije rada	Filip Domic	BACK	MAJOR	IMPROVEMENT	DONE	23.03.2019	2	Kada se desi promjena treba biti zabilježena u tablicama
20	RAD-19	Dorada rezervacije	Filip Domic	BACK	MAJOR	IMPROVEMENT	DONE	23.03.2019	2	Napravljeni error i napravljena sve ogracenja. I id uclavanje preko url-
21	RAD-20	Dorada komentara	Filip Domic	BACK	MAJOR	IMPROVEMENT	DONE	24.03.2019	0.5	
22	RAD-21	Servis za kronologiju rada (timeline)	Filip Domic	BACK	MAJOR	IMPROVEMENT	DONE	25.03.2019	1.5	
23	RAD-22	Dorada rezervacije odlucivanja i komentara	Filip Domic	BACK	MAJOR	IMPROVEMENT	DONE	25.03.2019	0.5	
24	RAD-23	Dorada	Filip Domic	BACK	MAJOR	IMPROVEMENT	DONE	04.04.2019	1	
25	RAD-24	Frontend pocetna	Filip Domic	FRONT	MAJOR	NEW FEATURE	DONE	01.04.2019	20	
26	RAD-25	Pocetna (rošnja)	Filip Domic	FRONT	MAJOR	NEW FEATURE	DONE	02.04.2019	1.5	
27	RAD-26	Dodaj i dohvati fakultet	Filip Domic	FRONT	MAJOR	NEW FEATURE	DONE	03.04.2019	1.5	
28	RAD-27	Dodaj i dohvati odjel	Filip Domic	FRONT	MAJOR	NEW FEATURE	DONE	03.04.2019	1.5	
29	RAD-28	Dodaj i dohvati djelatnika	Filip Domic	FRONT	MAJOR	NEW FEATURE	DONE	04.04.2019	1.5	
30	RAD-29	Dodaj odjel djelatnika	Filip Domic	FRONT	MAJOR	NEW FEATURE	DONE	04.04.2019	1.5	
31	RAD-30	Dodaj i dohvati studenta	Filip Domic	FRONT	MAJOR	NEW FEATURE	DONE	06.04.2019	1.5	
32	RAD-31	Dodaj datoteku	Filip Domic	FRONT	MAJOR	NEW FEATURE	DONE	06.04.2019	1.5	
33	RAD-32	Dodaj rad	Filip Domic	FRONT	MAJOR	NEW FEATURE	DONE	08.04.2019	1	
34	RAD-33	Penudena tema dodrada	Filip Domic	BACK	MAJOR	IMPROVEMENT	DONE	08.04.2019	0.5	

Slika 3. Excel tablica podjela projekta na zadatke

<sup>2</sup> Vučković, A. Faze upravljanja projektom na primjeru projekta rekonstrukcija he zakučac, 2016. završni rad, Split, dostupno na : <https://repozitorij.efst.unist.hr/islandora/object/efst%3A935/datastream/PDF/view> (03.09.2019)

## 3.2. Praćenje razvoja projekta

Upravljanje izvornim kodom (ili često se naiđe na naziv verzioniranje izvornog koda) je postupak pri kojem izvorni kod nekog softvera spremamo u središnju bazu podataka i tako bilježimo svu povijest promjena izvornog koda. Stupanj razvoja nekog softvera nazivamo inačica. Prilikom promjene neke inačice, tu promjenu spremamo u bazu te joj dajemo neki naziv koji može biti napisan brojevima, slovima ili kombinacija. Tako i svaka inačica ima i oznaku osobe koja je napravila promjenu na inačici. Prednost ovakvih sustava je da na ovaj način se vrlo jednostavno u projekt uključi više programera i svatko može raditi na svojem modulu i poslije ga vrlo jednostavno spojiti u jednu cjelinu. Danas najpoznatiji sustavi na tržištu za upravljanje izvornim kodom je GIT. GIT je također korišten u ovom projektu. Projekt smo spremali u besplatno GIT spremište na stranici GitHub.<sup>3</sup>

Na sljedećim poveznicama je izvorni kod na GitHub-u:

### Klijentski kod

<https://github.com/fdomic/eradovi-frontend>

### Serverski kod

<https://github.com/fdomic/eradovi-backend>

---

<sup>3</sup> GitHub, dostupno na : <https://github.com/> (03.09.2019)

## 4. Tehnologije razvoja

Radi lakšeg razvoja sustava programiranje je razdvojeno na dva sloja, a to su klijentski sloj i serverski sloj.

### 4.1. Klijentski razvoj

Klijentski razvoj pisan je u klijentskim tehnologijama za web programiranje, a to su: HTML, CSS i JavaScript. <sup>4</sup> Kako bi se olakšao i ubrzao sam razvoj programiranja korišten je razvojni okvir Angular<sup>5</sup> zajedno s bibliotekom NG-Zorro.<sup>6</sup>

### 4.2. Serverski razvoj<sup>7</sup>

Serverski razvoj napravljen je u programskom jeziku PHP s podrškom za MySQL bazu podataka. Prilikom serverskog programiranja također je korišten razvojni okvir PHP Laravel.<sup>8</sup>

---

<sup>4</sup> Maltar, J. Angular (2017.), diplomski rad, Osijek, dostupno na :

<https://zir.nsk.hr/islandora/object/mathos:142/preview> (03.09.2019)

<sup>5</sup> Angular (online) , dostupno na; <https://angular.io/guide/http> (03.09.2019)

<sup>6</sup> NgZorro (online), dostupno na : <https://ng.ant.design> (03.09.2019)

<sup>7</sup> Dumbovic, V. Razvoj aplikacije u lavarel okruženju ,diplomski rad, Zagreb , dostupno na:

<https://repozitorij.pmf.unizg.hr/islandora/object/pmf%3A631/datastream/PDF/view> (03.09.2019)

<sup>8</sup> Larvel (online), dostupno na: <https://laravel.com/docs/6.0> (03.09.2019)

## 5. Produkcijaska okolina

Kako bi se omogućilo korištenje sustava nakon faze programiranja potrebno je bilo uspostaviti produkcijsku okolinu kojoj će moći pristupiti svaki korisnik.<sup>9</sup>

Za potrebe produkcije zakupljen je server u oblaku kod davatelje usluge DigitalOcean te je na njemu podignuta okolina LAMP. LAMP označava akronim programa koje su potrebni za pokretanje same aplikacije s to su: LINUX kao operativni sustav, APACHE kao aplikativni server, MySQL kao baza podataka, te PHP kao programski jezik.<sup>10</sup>

Za pristup do produkcijske okoline koristi se sljedeća adresa:

<https://e-radovi.com>

Detaljni podaci za prijavu dani su u poglavlju 7.

---

<sup>9</sup> Valjak, S, Mysql sustav za upravljanje bazama podataka otvorenog koda,2015, završni rad, Gospić, dostupno na ; <https://zir.nsk.hr/islandora/object/velegs%3A487/datastream/PDF/view> (03.09.2019)

<sup>10</sup> Krečić,Ž. Primjer baze podataka u sustavu mysql,2013, završni rad ,Varaždin, dostupno na : <https://docs.microsoft.com/hr-hr/dynamics365/customer-engagement/admin/security-roles-privileges> (03.09.2019)

## **6. Ovlasti u aplikaciji**

Ovlasti u aplikacijama su vrlo bitne jer svaki softver želi zaštititi korisnike. Ne može svaki korisnik imati sve informacije i pristupe koje softver nudi. Svaki sustav je podijeljen na korisnike koji mogu imati različite ovlasti, i na administratora koji ima pristup svemu i ima sve ovlasti. U slučaju ovog softvera sama aplikacija je podijeljena na tri skupine korisnika (ovlasti): Referadu, Profesore i Studente.<sup>11</sup>

### **6.1. Referada**

Referada ima najveće ovlasti u ovom softveru. Referada predstavlja administratora te ona ima mogućnost dodavati korisnike odnosno profesore i studente, također ima mogućnost dodati neki novi fakultet, odjel i mogućnost djelatniku pridružiti odjel.

### **6.2. Profesor**

Profesor je drugi po jačini ovlasti koji ima u samom sustavu. Profesor ima mogućnost dodati neku od svojih ponuđenih tema, ima mogućnost odbiti ili prihvatiti rad studenta koji je napravi rezervaciju za neku od predloženih tema. Također je profesor u mogućnosti koristiti glavnu mogućnost programa a to je komunikacija sa studentom, preuzimanje rada od studenta i ostavljanje komentara.

### **6.3. Student**

Student ima ovlasti pribilježiti se na neku od ponuđenih tema od bilo kojeg profesora ili ima mogućnost predložiti neku od svojih tema. Također ima mogućnost komunicirati s profesorom, te mu poslati rad na pregled. Student ima također mogućnost ostaviti neki komentar profesoru.

---

<sup>11</sup> loc.cit.

## 7. Pristupni podaci

### Referada

Email: ref@unipu.hr

Lozinka: ref

### Profesor

Email: prof@unipu.hr

Lozinka: prof

### Student

Email: filip@unipu.hr

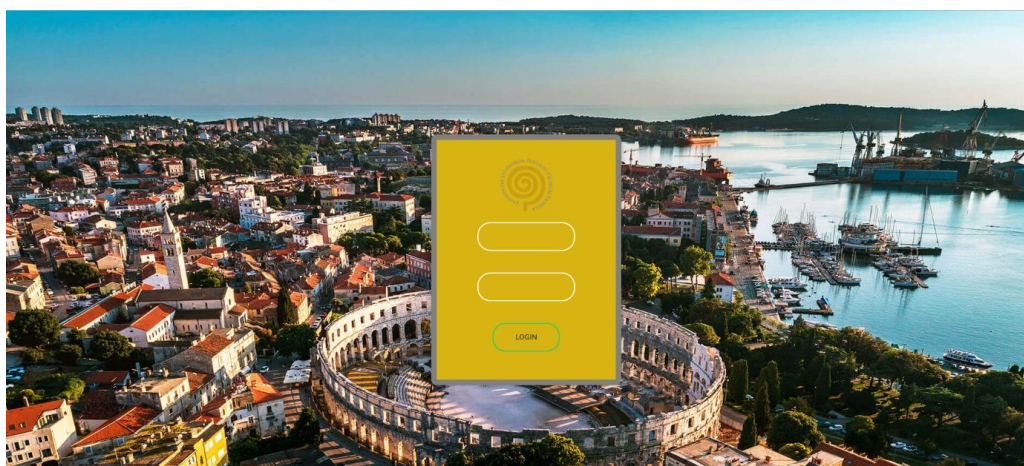
Lozinka: filip



## 8. Aplikacija

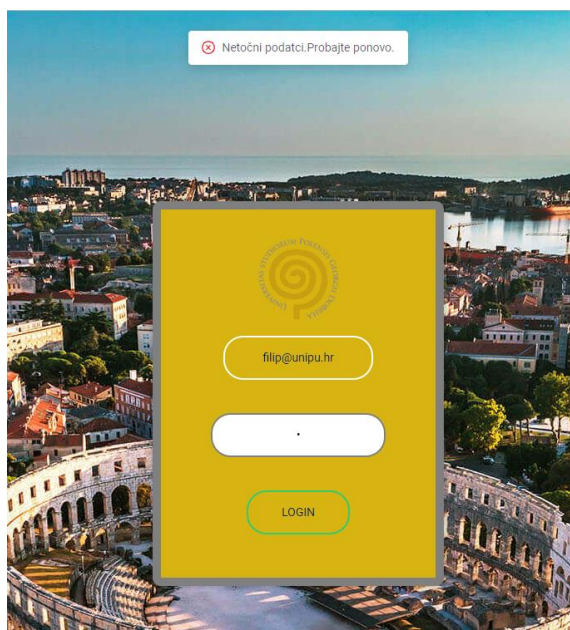
### 8.1. Prijava na stranicu

Kod stranice za prijavu provjeravamo osobne podatke. Napravljena je tako da pri samom kreiranju korisnika, njegova zaporka se kriptira i kada se korisnik prijavljuje, sustav se kreira sa web tokenom koji je vremenski ograničen. Slika 4. prikazuje stranicu za prijavu u aplikaciju.



Slika 4. Stranica za prijavu

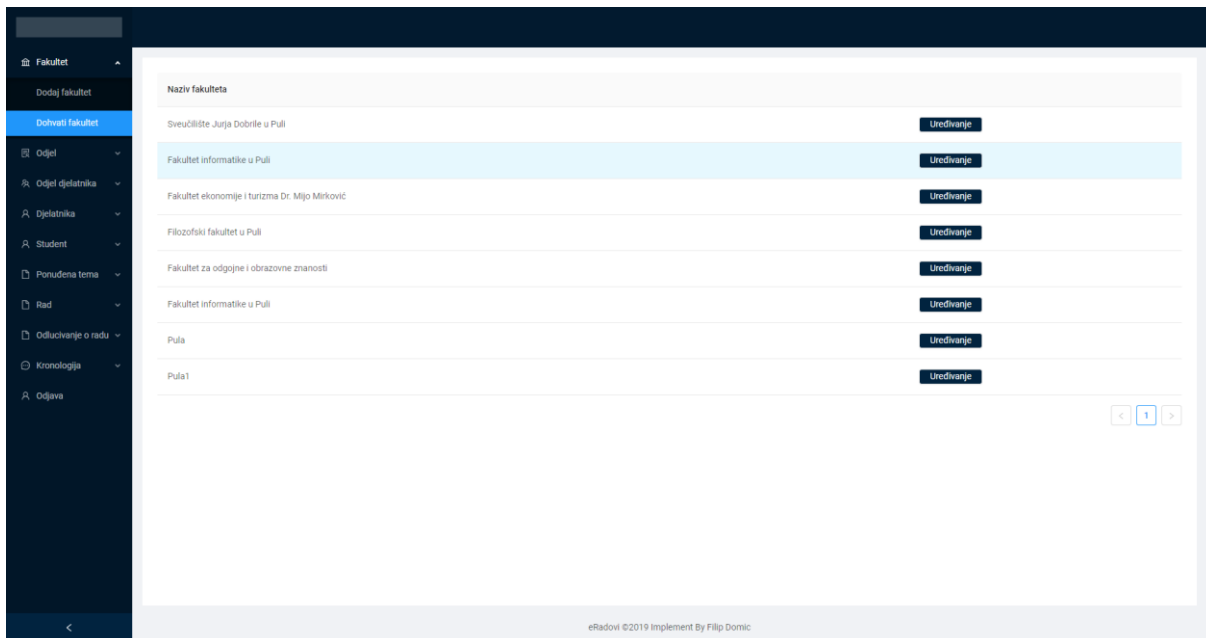
U slučaju da je korisnik poslao krive podatke, sustav javlja grešku da zaporka ili email nisu pravilno uneseni (Slika 4a.). Grešku javlja pomoću skočnog prozorčića u kojem piše „netočni podatci.“



Slika 4a. Greška prilikom prijave

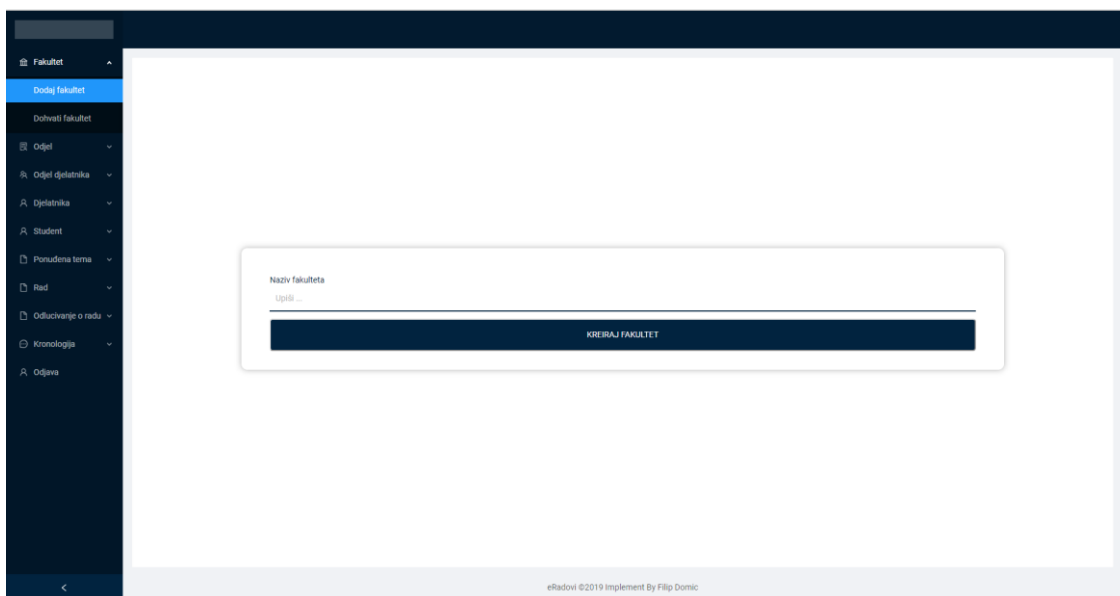
## 8.2. Upravljanje fakultetima

Prva mogućnost u samoj aplikaciji je dodavanje fakulteta. Otvaranjem dohvata fakulteta prikazuje nam se prozor odnosno lista svih fakulteta koje smo dodali. Izgled navedenog ekrana prikazuje Slika 5.



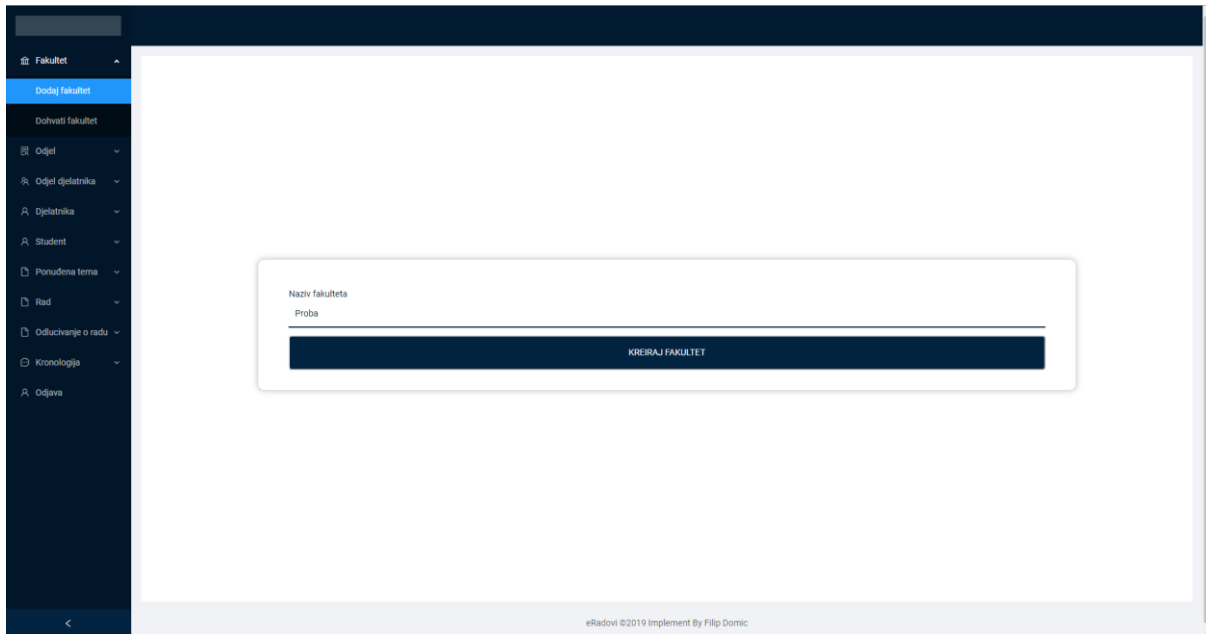
Slika 5. Dohvat fakulteta

U daljnjem prikazivanju mogućnosti prikazat ćemo na koji način dodajemo fakultete. Izgled navedenog ekrana prikazuje Slika 6.



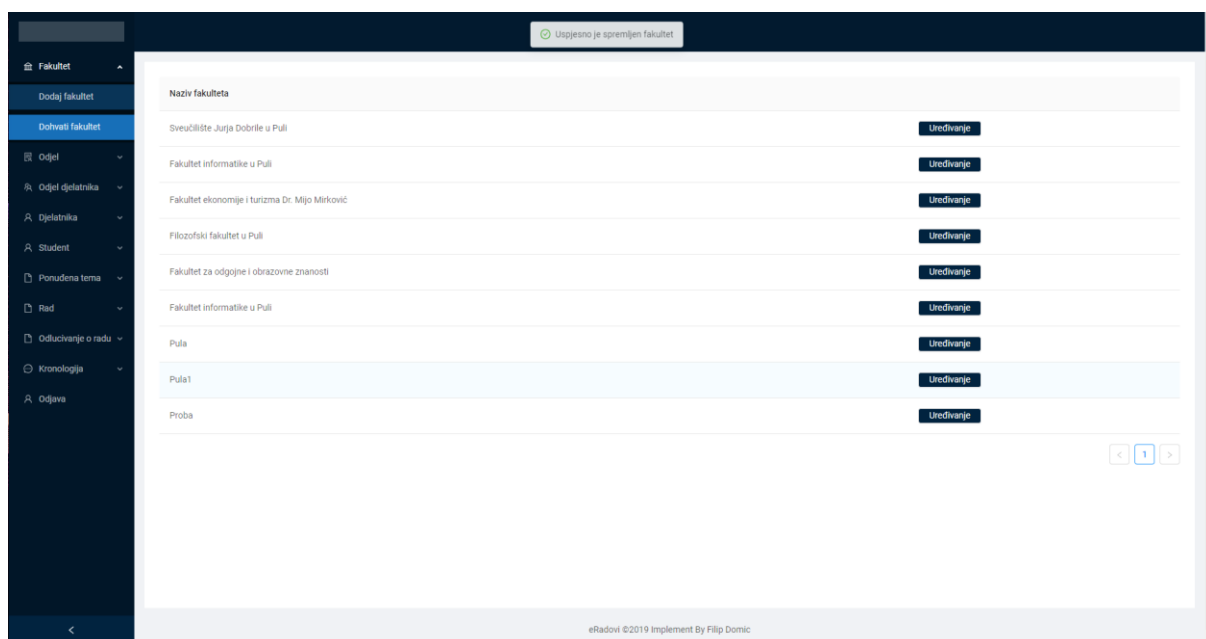
Slika 6. Dodaj fakultet

Na ovom ekranu imamo mogućnost kreiranja novog fakulteta ili uređivanje postojećeg fakulteta. Klikom na uređivanje fakulteta otvaramo prozor koji nam nudi mogućnost promjene pojedinog fakulteta. Mijenjati možemo samo naziv fakulteta ili ako se kreira neki novi fakultet upisujemo samo jedan atribut koji je ime fakulteta. Izgled navedenog ekrana prikazuje slika 7.



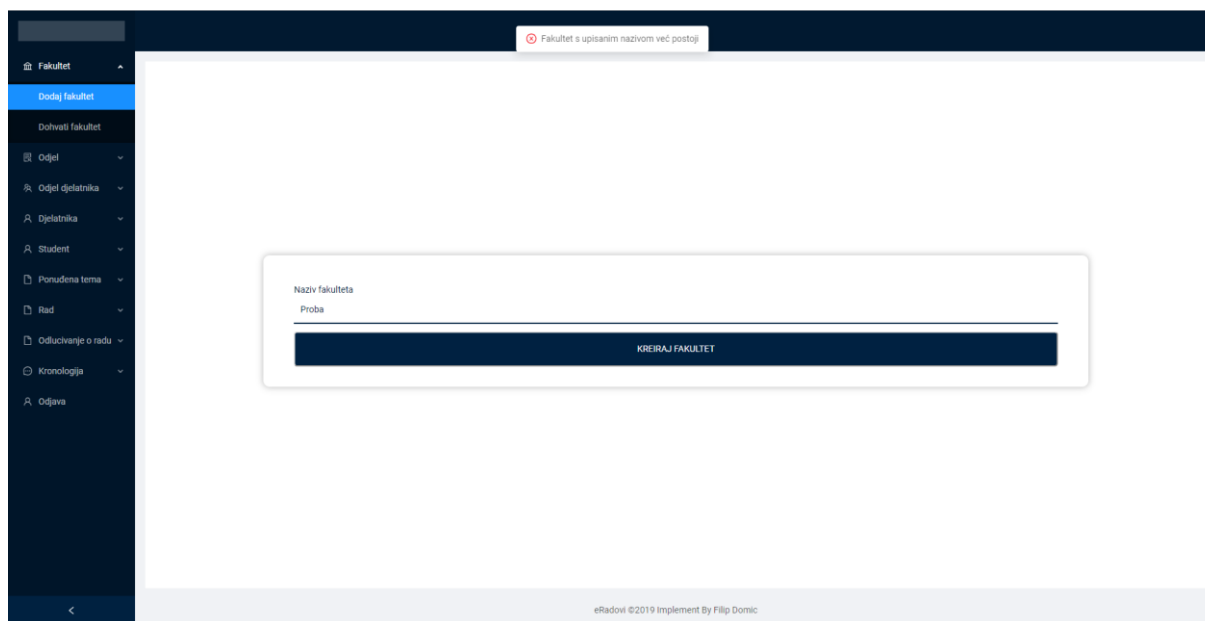
**Slika 7. Dodavanje novog fakulteta**

Ako je fakultet uspješno spremljen ispisat će se u tablici i ispisuje se obavijest da je fakultet dobro spremljen. Izgled navedenog ekrana prikazuje slika 8.



**Slika 8. Potvrda spremanja fakulteta**

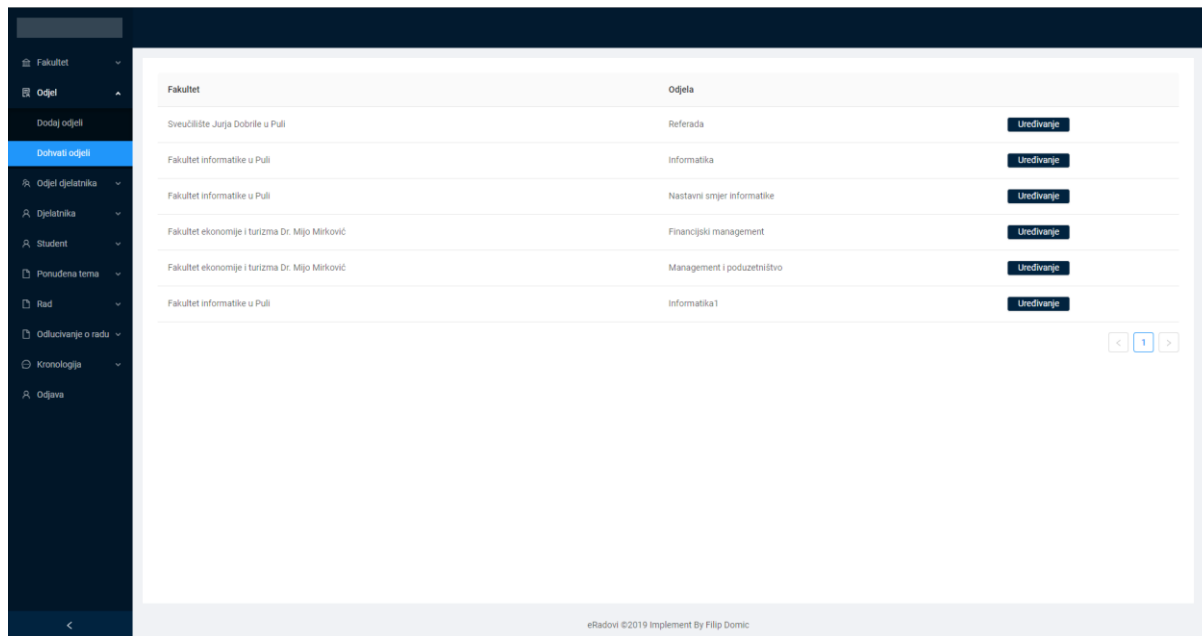
U slučaju da korisnik pokuša upisati postojeći fakultet aplikacija će ispisati grešku, odnosno ispisat će da taj fakultet s tim imenom već postoji. Izgled navedenog ekrana prikazuje slika 9.



**Slika 9. Dodaj postojeći fakultet**

### 8.3. Upravljanje odjelima

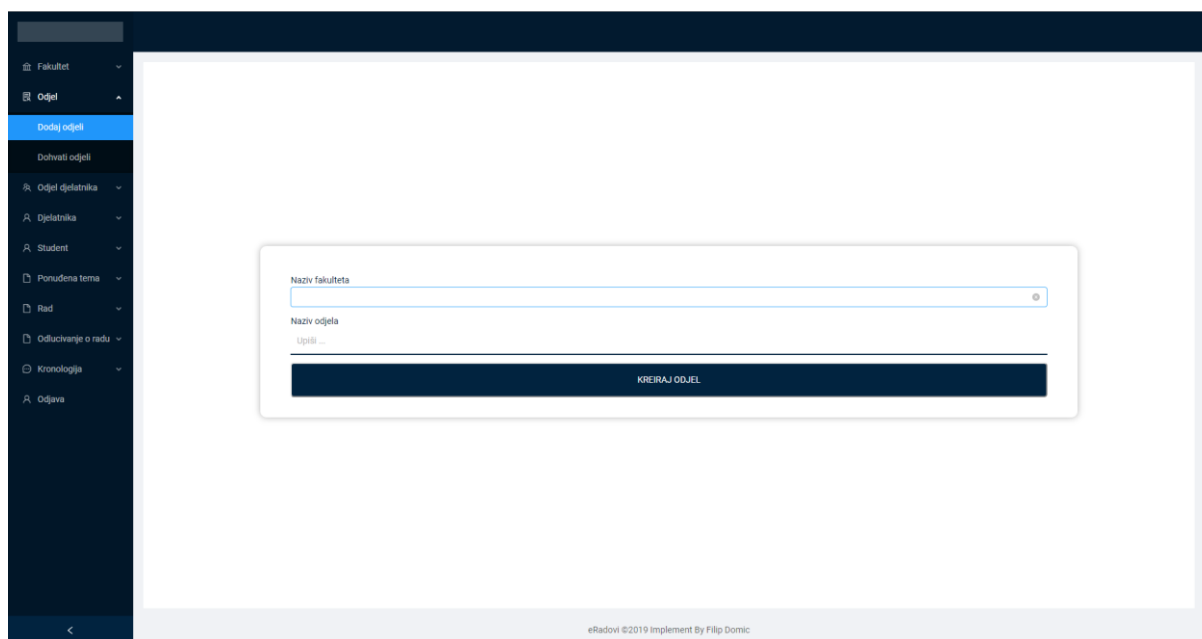
Nakon opcija za fakultete imamo također opciju za odjele gdje nam se prikazuje prozor odnosno lista svih odjela koje smo dodali, to je prikazano na slici 10.



Fakultet	Odjela	
Sveučilište Jurja Dobrića u Puli	Referada	Uredi
Fakultet informatike u Puli	Informatika	Uredi
Fakultet informatike u Puli	Nastavni smjer informatike	Uredi
Fakultet ekonomije i turizma Dr. Mijo Mirković	Financijski management	Uredi
Fakultet ekonomije i turizma Dr. Mijo Mirković	Management i poduzetništvo	Uredi
Fakultet informatike u Puli	Informatika1	Uredi

Slika 10. Tablica odjela

U daljnjem prikazivanju mogućnosti prikazat ćemo na koji način dodajemo odjele. Izgled navedenog ekrana prikazuje slika 11.



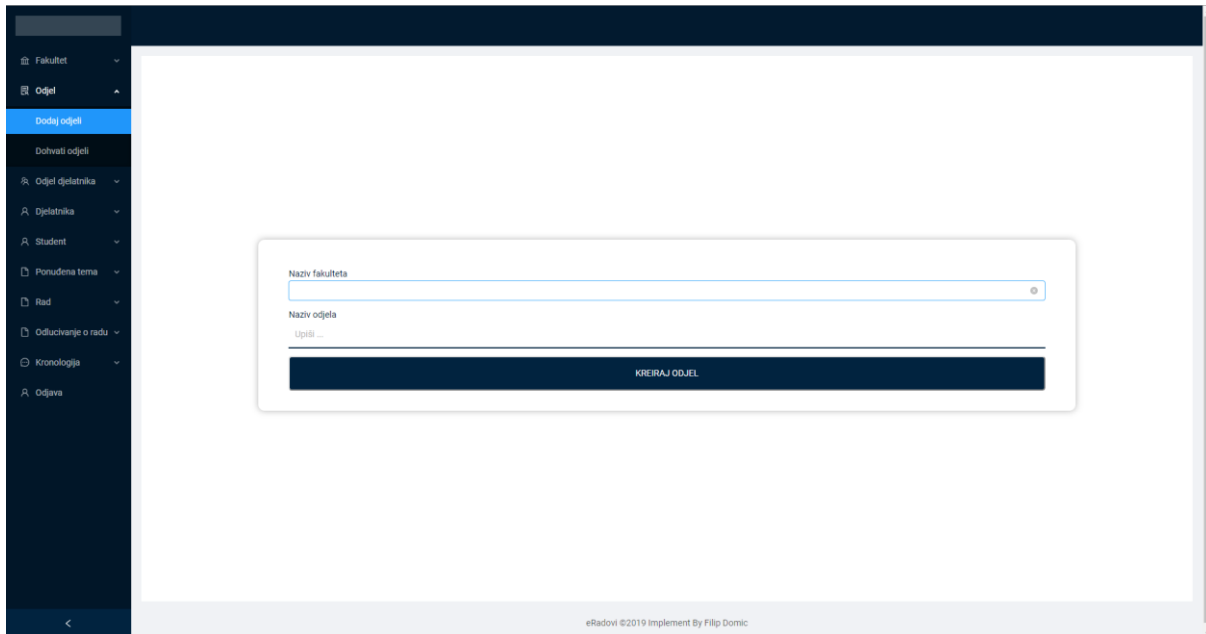
Naziv fakulteta

Naziv odjela

KREIRAJ ODJEL

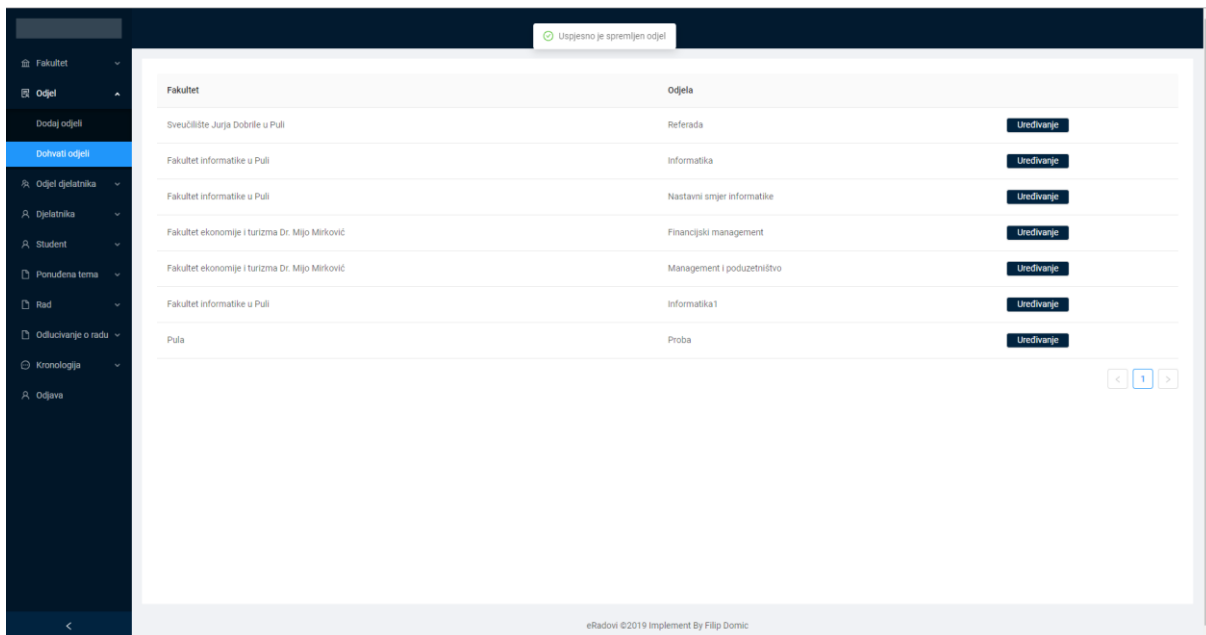
Slika 11. Dodaj odjel

Na ovom ekranu imamo mogućnost kreiranja nekog novog odjela ili uređivanje postojećeg odjela. Klikom na uređivanje odjela otvaramo prozor koji nam nudi mogućnost promjene pojedinog odjela. Mijenjati možemo naziv fakulteta i ime odjela ili ako se kreira neki novi odjel upisujemo ime odjela i pridružujemo ga nekom fakultetu. Izgled navedenog ekrana prikazuje slika 12.



**Slika 12. Dodavanje novog odjela**

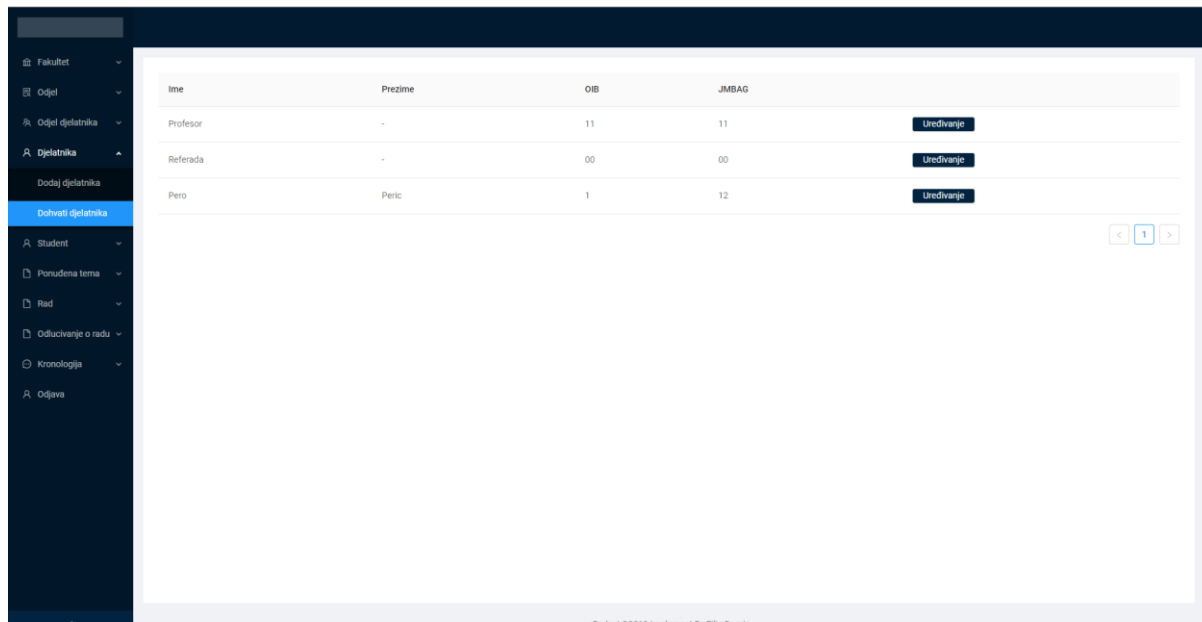
Ako je odjel uspješno spremljen ispisat će se u tablici i ispisuje se obavijest da je odjel dobro spremljen. Izgled navedenog ekrana prikazuje slika 13.



**Slika 13. Odjel uspješno spremljen**

## 8.4. Upravljanje djelatnicima

Sljedeća od mogućnosti je dodavanje djelatnika. Otvaranjem dohvata djelatnika prikazuje nam se prozor odnosno lista svih djelatnika koje smo dodali. Izgled navedenog ekrana prikazuje slika 14.



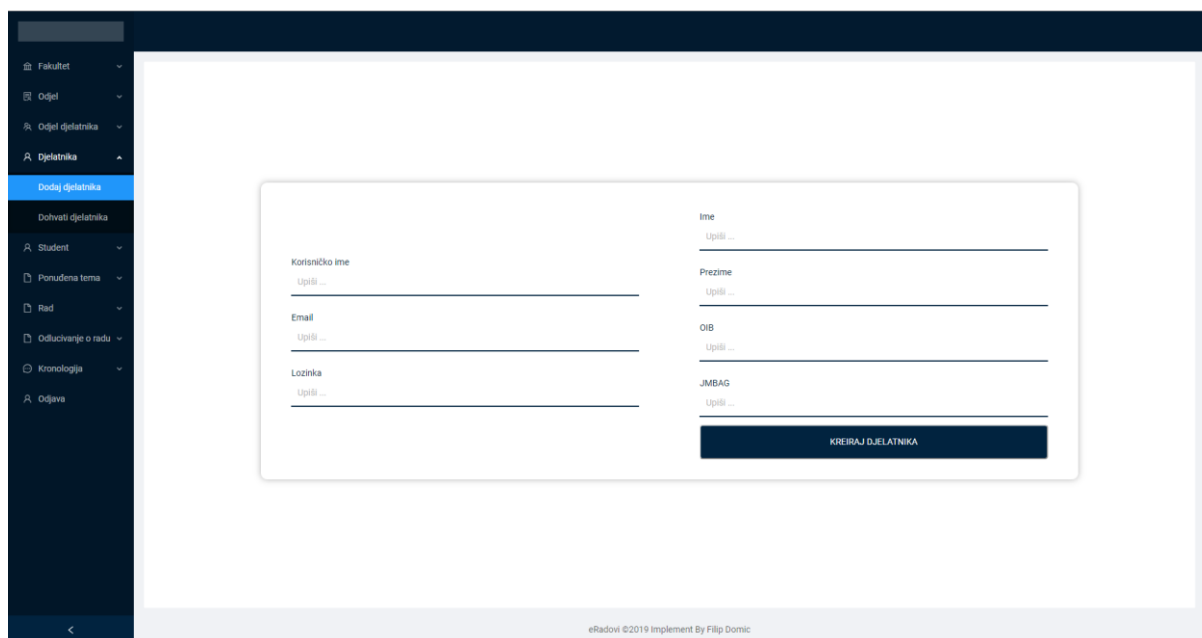
The screenshot shows a web application interface with a dark blue sidebar on the left containing navigation options like 'Fakultet', 'Odjel', 'Djelatnika', 'Dodaj djelatnika', 'Dohvati djelatnika', 'Student', 'Ponuđena tema', 'Rad', 'Odlučivanje o radu', 'Kronologija', and 'Odjava'. The main content area displays a table with the following data:

Ime	Prezime	OIB	JMBAG	
Profesor	-	11	11	<a href="#">Uredi</a>
Referada	-	00	00	<a href="#">Uredi</a>
Pero	Peric	1	12	<a href="#">Uredi</a>

At the bottom right of the table, there are navigation controls: '< 1 >'. The footer of the application reads 'eRadovi ©2019 Implement By Filip Domic'.

Slika 14. Tablica djelatnika

U daljnjem prikazivanju mogućnosti prikazat ćemo na koji način dodajemo djelatnika. Izgled navedenog ekrana prikazuje slika 15.



The screenshot shows the 'Dodaj djelatnika' (Add employee) form. The sidebar is the same as in Slika 14. The main content area contains a form with the following fields:

- Korisničko ime (User name):
- Ime (Name):
- Prezime (Surname):
- Email:
- OIB:
- Lozinka (Password):
- JMBAG:

At the bottom right of the form is a dark blue button labeled 'KREIRAJ DJELATNIKA'. The footer of the application reads 'eRadovi ©2019 Implement By Filip Domic'.

Slika 15. Dodaj djelatnika

Na ovom ekranu imamo mogućnost kreiranja nekog novog djelatnika ili uređivanje postojećeg djelatnika. Klikom na uređivanje djelatnika otvaramo prozor koji nam nudi mogućnost promjene pojedinog djelatnika. Mijenjati možemo ime, prezime, OIB i JMBAG ili ako se kreira neki novi djelatnika upisujemo attribute koje smo naveli u prijašnjoj rečenici te još dodatno email, lozinku i korisničko ime koji se mogu mijenjati samo prilikom kreacije. Izgled navedenog ekrana prikazuje slika 16.

**Slika 16. Dodavanje novog djelatnika**

Ako je djelatnik uspješno spremljen ispisat će se u tablici i ispisuje se obavijest da je djelatnik dobro spremljen. Izgled navedenog ekrana prikazuje slika 17.

Ime	Prezime	OIB	JMBAG	
Profesor	-	11	11	Uređivanje
Referada	-	00	00	Uređivanje
Pero	Peric	1	12	Uređivanje
proba	proba			Uređivanje

**Slika 17. Djelatnik uspješno spremljen**



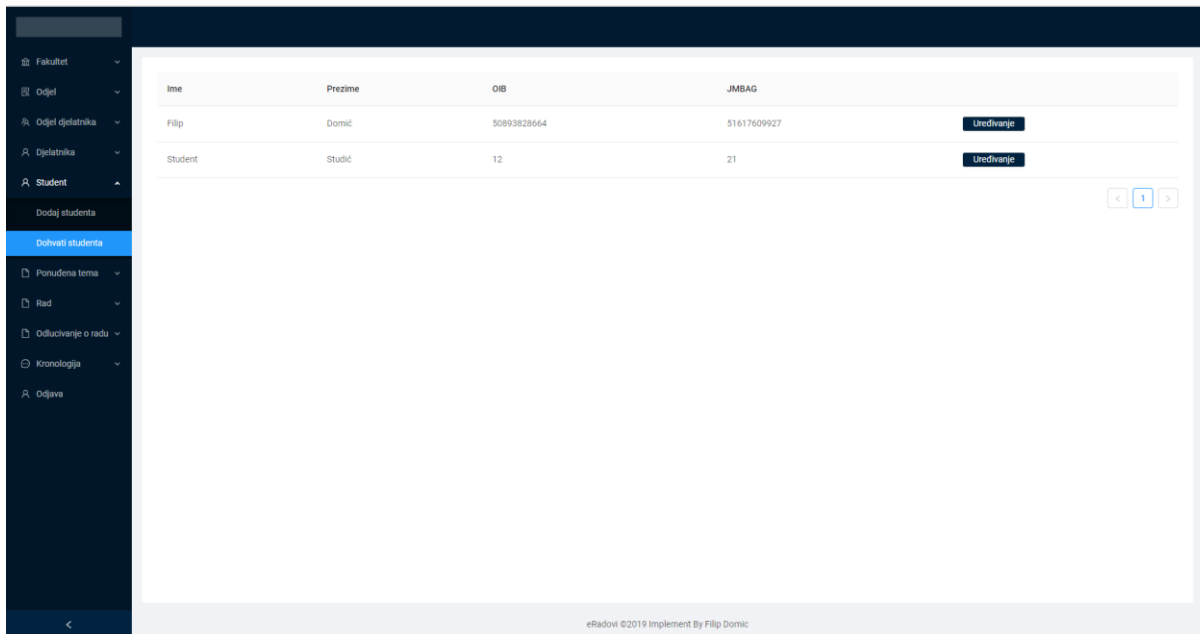
U slučaju da korisnik pokuša upisati postojeći OIB ili JMBAG aplikacija će ispisati grešku, odnosno ispisat će da taj djelatnik već postoji. Izgled navedenog ekrana prikazuje slika 18.

The screenshot shows a web application interface with a dark blue sidebar on the left containing a menu with items like 'Fakultet', 'Odjel', 'Odjel djelatnika', 'Djelatnika', 'Dodaj djelatnika', 'Dohvati djelatnika', 'Student', 'Ponuđena tema', 'Rad', 'Odlučivanje o radu', 'Kronologija', and 'Odjava'. The main content area is white and features a form for adding a user. At the top of the form, a red error message reads 'Djelatnik već postoji'. The form has two columns of input fields: 'Korisničko ime' (Username) with 'proba', 'Email' with 'proba', and 'Lozinka' (Password) with 'Upiši ...'. The second column contains 'Ime' (Name) with 'proba', 'Prezime' (Surname) with 'proba', 'OIB' with '1', and 'JMBAG' with 'Upiši ...'. A dark blue button labeled 'SPREMI PROMJENE' is at the bottom right of the form. At the bottom of the page, there is a small copyright notice: 'eRadovi ©2019 Implement By Filip Domic'.

**Slika 18. Greška krivi unos podataka**

## 8.5. Upravljanje studentima

Sljedeća od mogućnosti je dodavanje studenata. Otvaranjem dohvata studenata prikazuje nam se prozor odnosno lista svih studenata koje smo dodali. Izgled navedenog ekrana prikazuje slika 19.



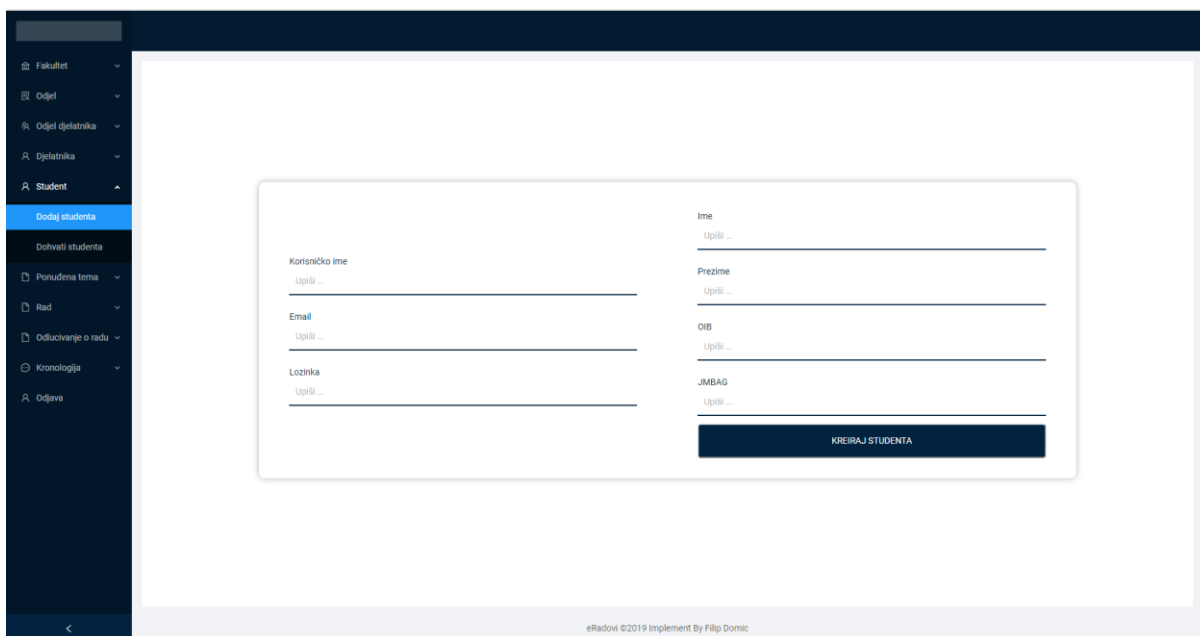
The screenshot shows a web application interface for managing students. On the left is a dark sidebar with a menu containing options like 'Fakultet', 'Odjel', 'Odjel djelatnika', 'Djelatnika', 'Student', 'Dodaj studenta', 'Dohvati studenta', 'Ponuđena tema', 'Rad', 'Odlučivanje o radu', 'Kronologija', and 'Odjava'. The main content area displays a table with the following data:

Ime	Prezime	OIB	JMBAG	
Filip	Domac	50893828664	51617609927	<a href="#">Uređivanje</a>
Student	Studić	12	21	<a href="#">Uređivanje</a>

At the bottom right of the table, there are navigation controls: '< 1 >'. At the bottom center of the page, there is a footer: 'eRadovi ©2019 Implement By Filip Domic'.

Slika 19. Tablica studenata

U daljnjem prikazivanju mogućnosti prikazat ćemo na koji način dodajemo studenata. Izgled navedenog ekrana prikazuje slika 20.



The screenshot shows the 'Dodaj studenta' (Add student) form. The sidebar is the same as in the previous image. The main content area contains a form with the following fields:

- Korisničko ime (Username):
- Email:
- Lozinka (Password):
- Ime (Name):
- Prezime (Surname):
- OIB:
- JMBAG:

At the bottom right of the form is a dark blue button labeled 'KREIRAJ STUDENTA'. At the bottom center of the page, there is a footer: 'eRadovi ©2019 Implement By Filip Domic'.

Slika 20. Dodaj studenta

Na ovom ekranu imamo mogućnost kreiranja nekog novog studenata ili uređivanje postojećeg studenta. Klikom na uređivanje studenata otvaramo prozor koji nam nudi mogućnost promjene pojedinog studenata. Mijenjati možemo ime, prezime, OIB i JMBAG ili ako se kreira neki novi studenata upisujemo attribute koje smo naveli u prijašnjoj rečenici te još dodatno email, lozinku i korisničko ime koji se mogu mijenjati samo prilikom kreacije. Izgled navedenog ekrana prikazuje slika 21.

**Slika 21. Dodavanje novog studenta**

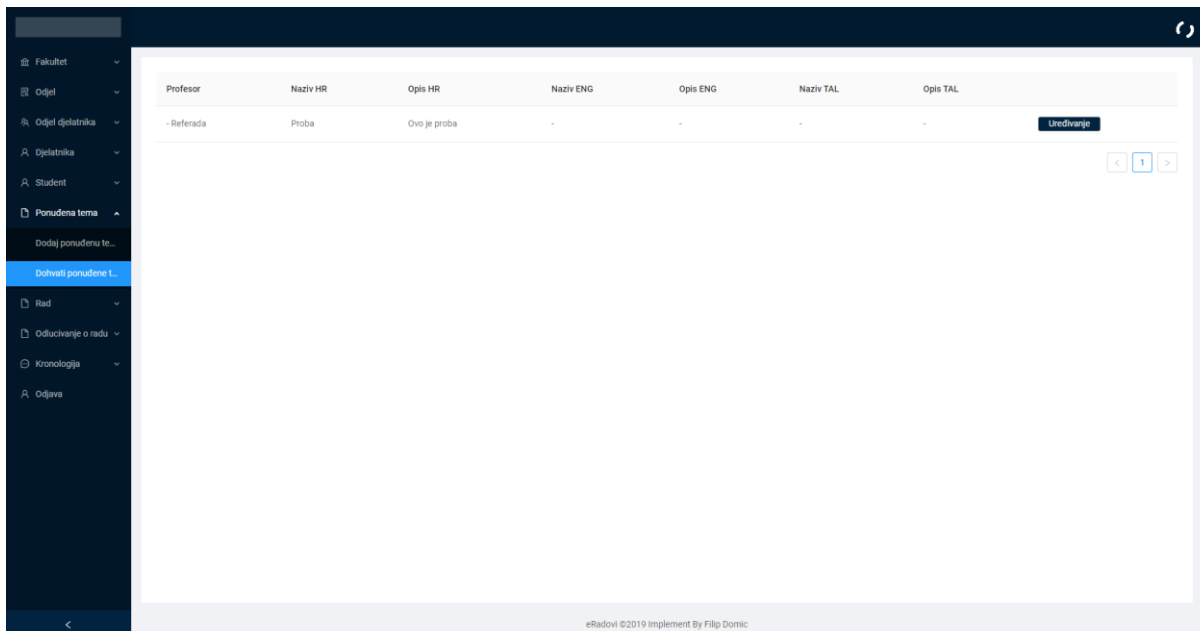
Ako je student uspješno spremljen ispisat će se u tablici i ispisuje se obavijest da je student dobro spremljen. Izgled navedenog ekrana prikazuje slika 22.

Ime	Prezime	OIB	JMBAG	
Filip	Domac	50893828664	51617609927	Uređivanje
Student	Studic	12	21	Uređivanje
proba	proba	1	1	Uređivanje

**Slika 22 Student uspješno spremljen**

## 8.6. Upravljanje ponuđenim temama

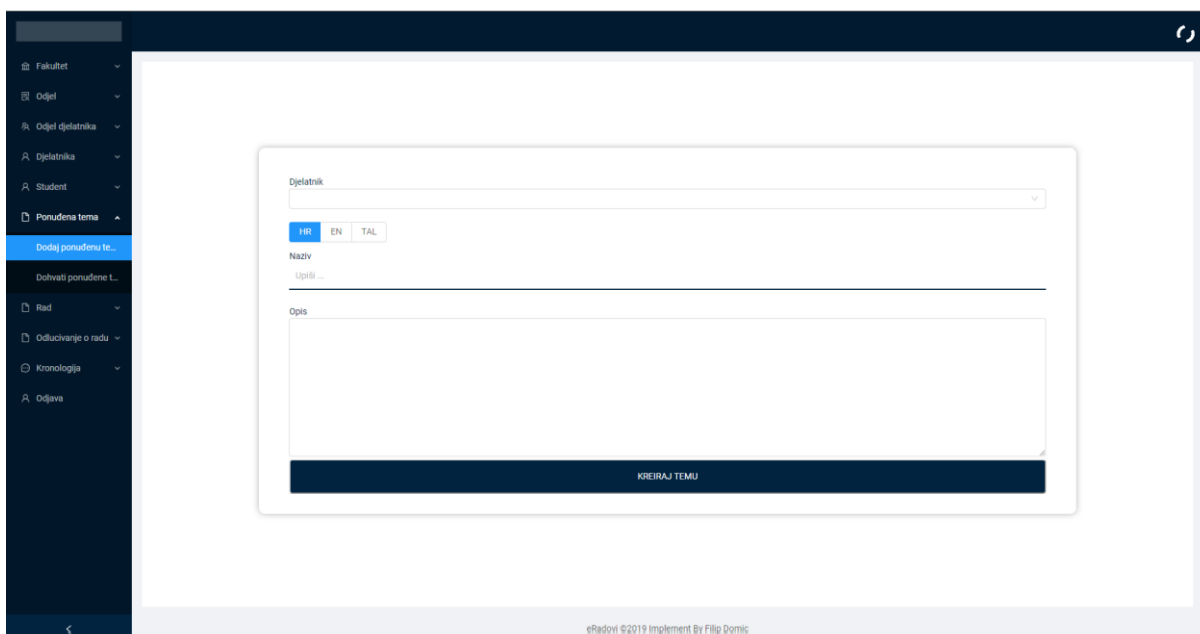
Sljedeća od mogućnosti je dodavanje ponuđene tema. Otvaranjem dohvata ponuđenih tema prikazuje nam se prozor odnosno lista svih predloženih tema koje smo dodali. Izgled navedenog ekrana prikazuje slika 23.



Profesor	Naziv HR	Opis HR	Naziv ENG	Opis ENG	Naziv TAL	Opis TAL
- Referada	Proba	Ovo je proba	-	-	-	-

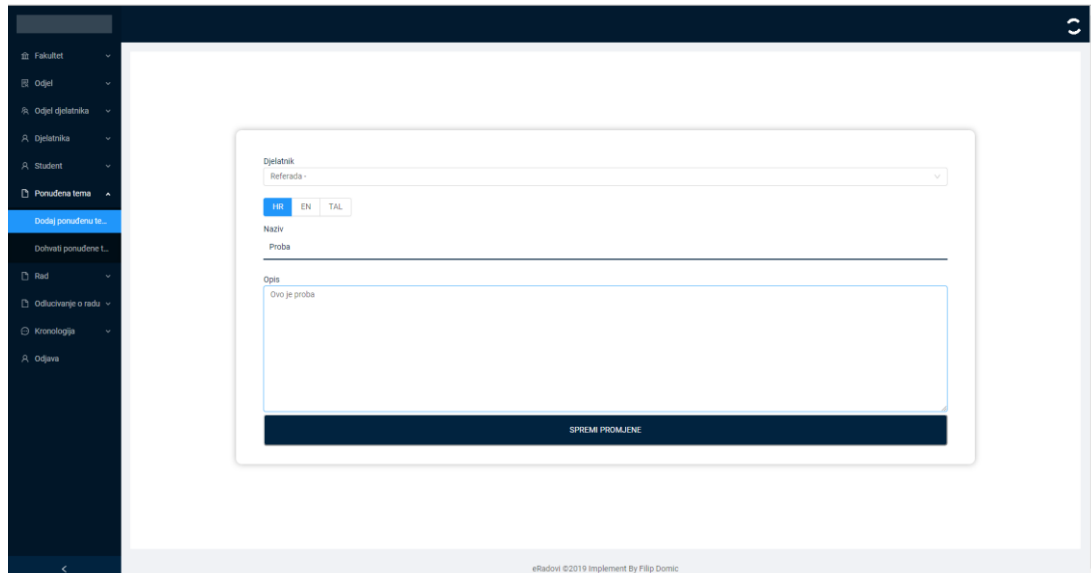
Slika 23. Tablica ponuđene teme

U daljnjem prikazivanju mogućnosti prikazat ćemo na koji način dodajemo ponuđene tema. Izgled navedenog ekrana prikazuje slika 24.



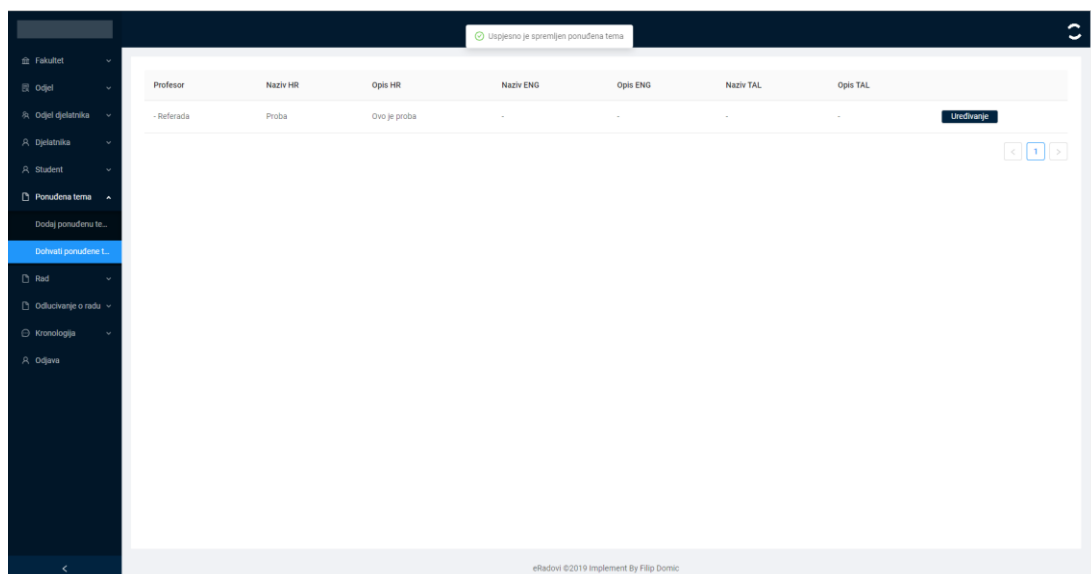
Slika 24. Dodaj ponuđenu temu

Na ovom ekranu imamo mogućnost kreiranja neke nove ponuđene teme ili uređivanje ponuđene teme. Klikom na uređivanje ponuđene teme otvaramo prozor koji nam nudi mogućnost promjene pojedine ponuđene teme. Mijenjati možemo djelatnika, naziv na hrvatskom, engleskom i talijanskom, tako i opise na ta tri jezika ili ako se kreira neka nova ponuđena tema upisujem attribute koje smo naveli u prijašnjoj rečenici. Izgled navedenog ekrana prikazuje slika 25.



**Slika 25. Dodavanje ponuđene teme**

Ako je ponuđena tema uspješno spremljen ispisat će se u tablici i ispisuje se obavijest da je ponuđena tema dobro spremljen. Izgled navedenog ekrana prikazuje slika 26.

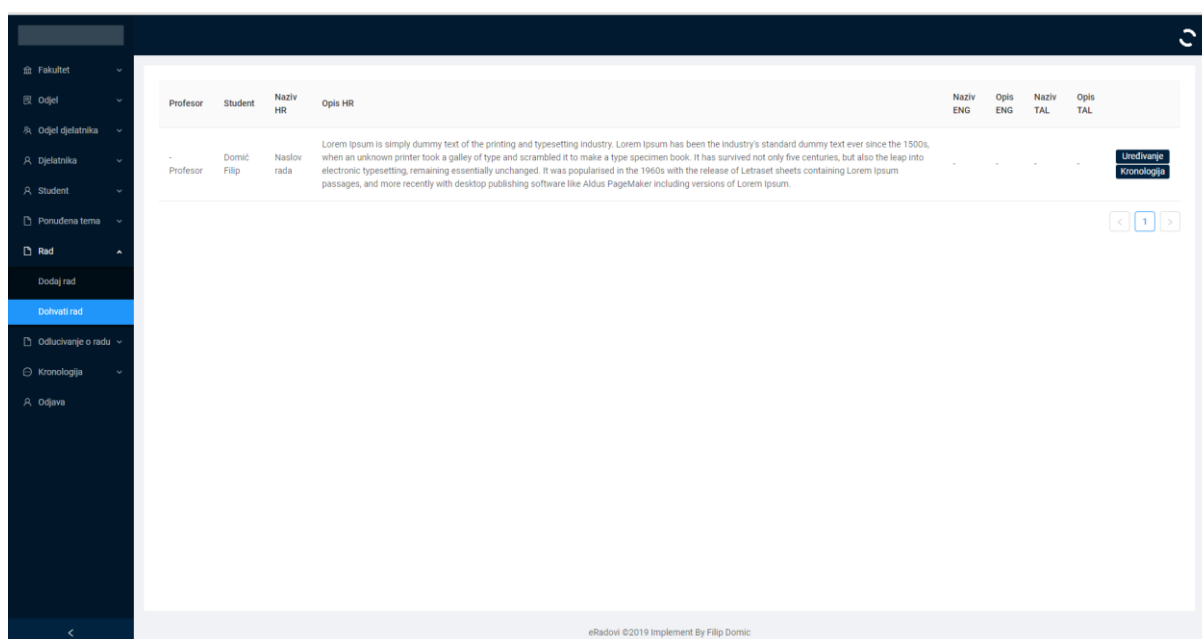


**Slika 26. Ponuđena tema prihvaćena**

Valja napomenuti da polje za izbor djelatnika se prikazuje samo korisniku koji je prijavljen kao referada. Ako je u sustavu prijavljen profesor prikazuje se ista forma bez padajućeg izbornika za djelatnike. Studenti ne vide ekran za uređivanje ponuđenih tema.

## 8.7. Upravljanje radovima

Sljedeća od mogućnosti je dodavanje rada. Otvaranjem dohvata radova prikazuje nam se prozor odnosno lista svih radova koje smo dodali. Izgled navedenog ekrana prikazuje slika 27.



Profesor	Student	Naziv HR	Opis HR	Naziv ENG	Opis ENG	Naziv TAL	Opis TAL
-	Domić Filip	Naslov rada	Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.	-	-	-	-

Slika 27. Tablica radova

U daljnjem prikazivanju mogućnosti prikazat ćemo na koji način dodajemo radove. Izgled navedenog ekrana prikazuje slika 28.

Student

Profesor

HR EN TAL

Naziv  
Upiši ...

Opis

KREIRAJ RAD

eRadovi ©2019 Implement By Filip Domic

**Slika 28. Dodaj rad**

Na ovom ekranu imamo mogućnost kreiranja nekih novih radova ili uređivanje samog rada. Klikom na uređivanje rada otvaramo prozor koji nam nudi mogućnost promjene pojedinog rada. Mijenjati možemo djelatnika, naziv na hrvatskom, engleskom i talijanskom, tako i opise na ta tri jezika ili ako se kreira neka nova ponuđena tema upisujemo attribute koje smo naveli u prijašnjoj rečenici. Izgled navedenog ekrana prikazuje slika 29.

Student  
Filip

Profesor  
Profesor

HR EN TAL

Naziv  
Proba

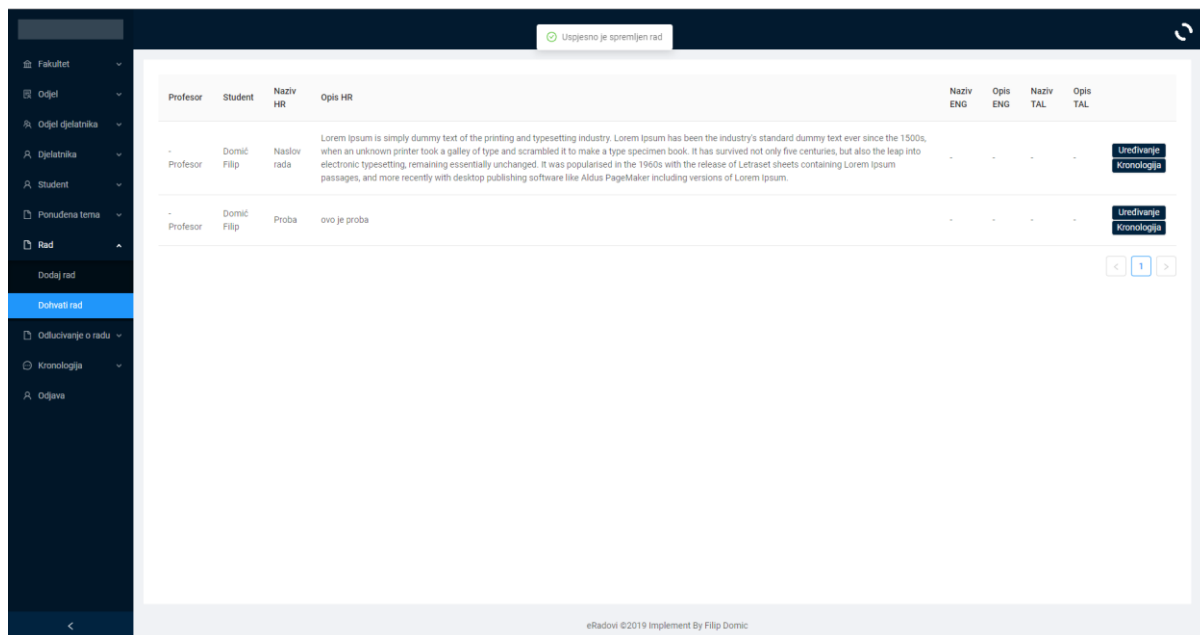
Opis  
ovo je proba

KREIRAJ RAD

eRadovi ©2019 Implement By Filip Domic

**Slika 29 Dodavanje rada**

Ako je rad uspješno spremljen ispisać će se u tablici i ispisać se obavijest da je rad dobro spremljen. Izgled navedenog ekrana prikazuje slika 30.



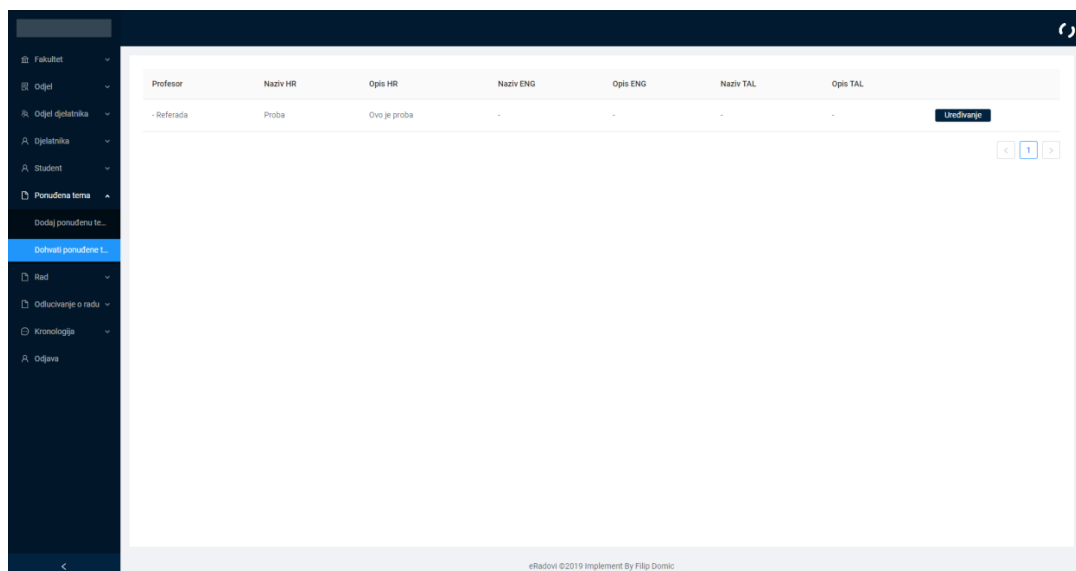
Slika 30. Rad prihvaćen

Valja napomenuti da polje za izbor djelatnika i studenta se prikazuje samo korisniku koji je prijavljen kao referada. Ako je u sustavu prijavljen profesor prikazuje se ista forma bez padajućeg izbornika za djelatnika, ali s padajućim izbornikom za izbor studenta. Studenti ne vidi niti jedan padajući izbornik, nego samo ostatak forme.



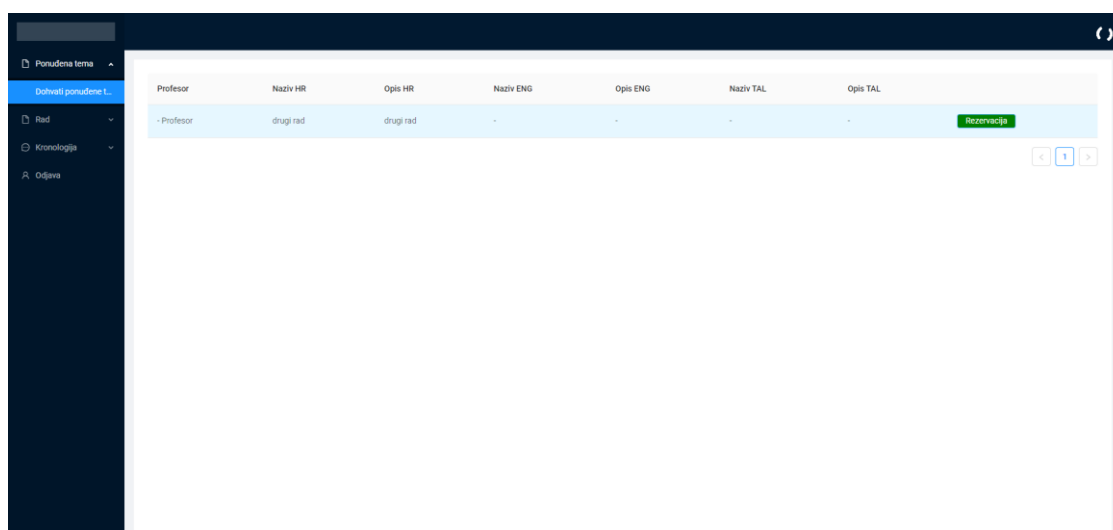
## 8.8. Rezervacija ponuđene teme

Sljedeća od mogućnosti je rezervacija ponuđene teme. Otvaranjem dohvata ponuđenih tema prikazuje nam se prozor odnosno lista svih predloženih tema koje smo dodali i s desne strane ekrane u istom redu gdje je i naziv teme nalazi se gumb za rezervaciju. Izgled navedenog ekrana prikazuje slika 31.



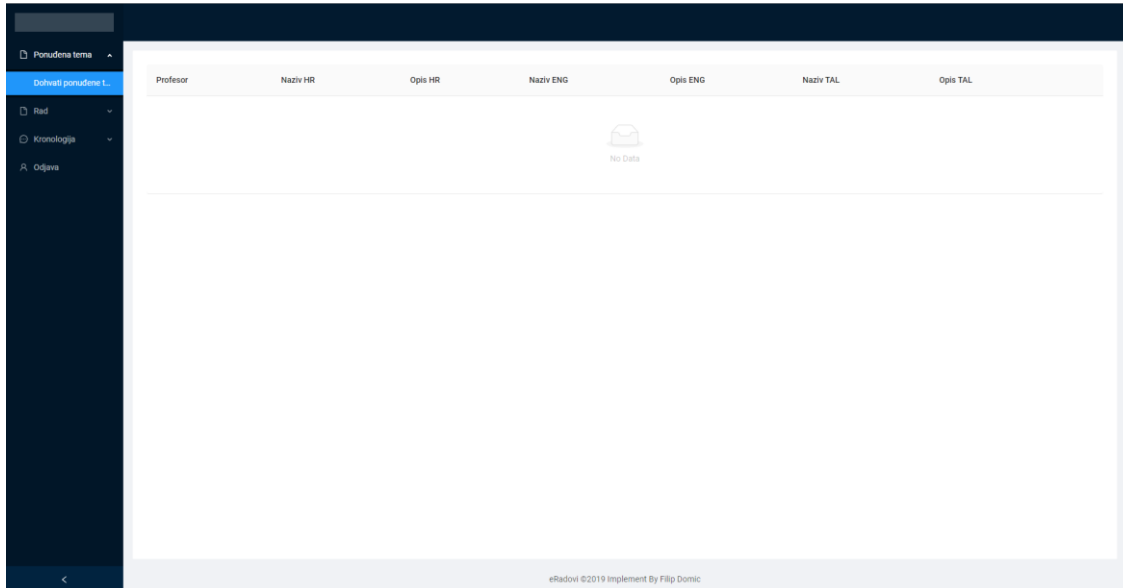
Slika 31. Tablica ponuđene teme

U daljnjem prikazivanju mogućnosti prikazat ćemo na koji način napravimo rezervaciju. Izgled navedenog ekrana prikazuje slika 32.



Slika 32. Rezervacija

Kao što se vidi na prethodnoj slici 32. prilikom klika na gumb rezervacija on se zazeleni i tako je student poslao upit na tu temu ta se tema automatski miče s liste ponuđenih tema i šalje se na odluku profesoru. Izgled navedenog ekrana prikazuje slika 33.

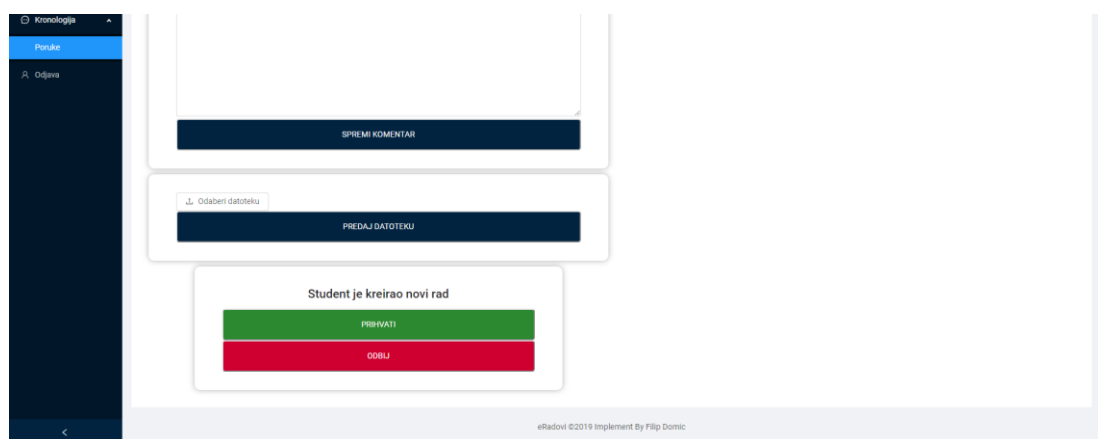


**Slika 33. Rezervirana tema ne prikazuje se više u tablici**

Možemo uočiti da slika 31. i 32. imaju različit izgled sučelja, a to je zato što je jedan pogled s prijavljenom referatom, a drugi s prijavljenim studentom. Osim same razlike u sučelju ovdje možemo uočiti i razlike u funkcionalnosti. Ako je prijavljen korisnik referada ili profesor u tablici ponuđenih tema bit će gumb za uređivanje teme, a ako je prijavljen student bit će prisutan gumb za rezervaciju teme.

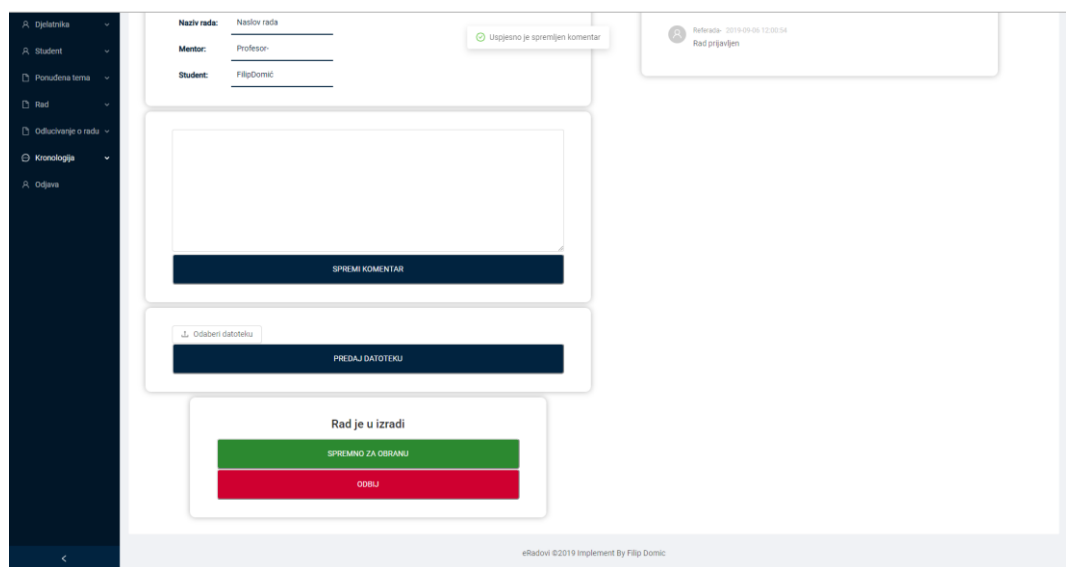
## 8.9. Upravljanje odlukama

Od sljedećih mogućnosti za profesora i referadu je mogućnost donositi odluke da li je rad pravovaljan i dali zadovoljava sve uvjete, te dali su pravilno nazivi i opisi napisani. Otvaranjem kronologije profesor ili referada prvo trebaju potvrditi rad ili ponuđenu temu. Izgled navedenog ekrana prikazuje slika 34.



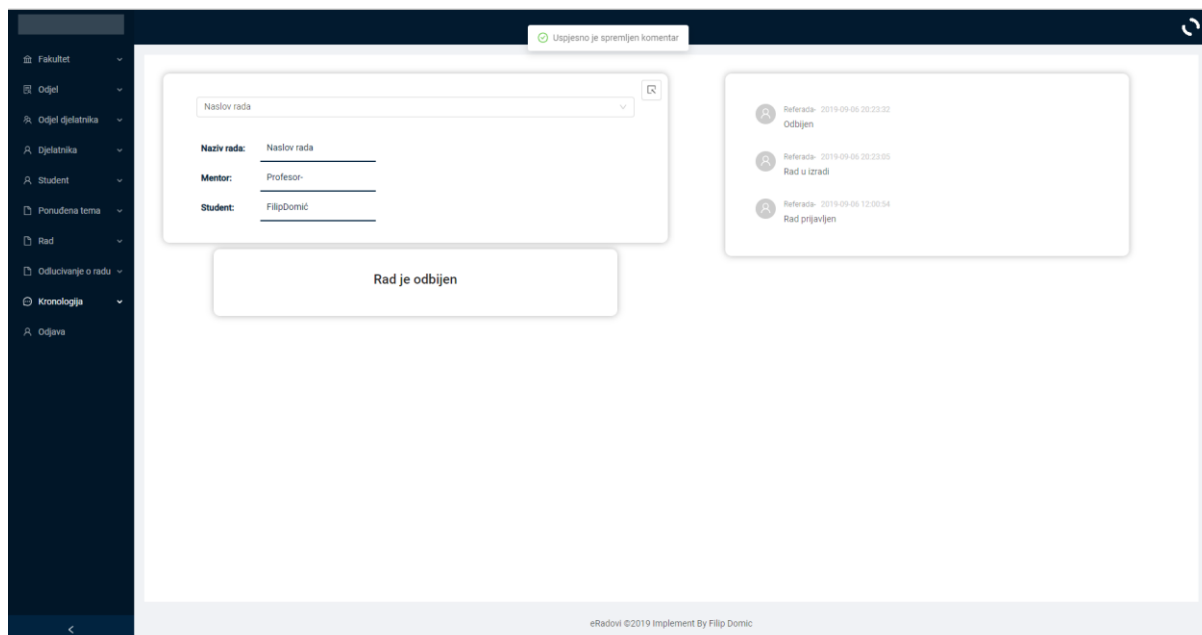
Slika 34. Odluke o radu

Postoje dvije mogućnosti odabira prihvaćen rad i odbijen. Prvo će biti prikazano što se dešava kada je rad prihvaćen. Kada je rad prihvaćen spreman je za daljnju izradu rada. Izgled navedenog ekrana prikazuje slika 35.



Slika 35. Rad prihvaćen

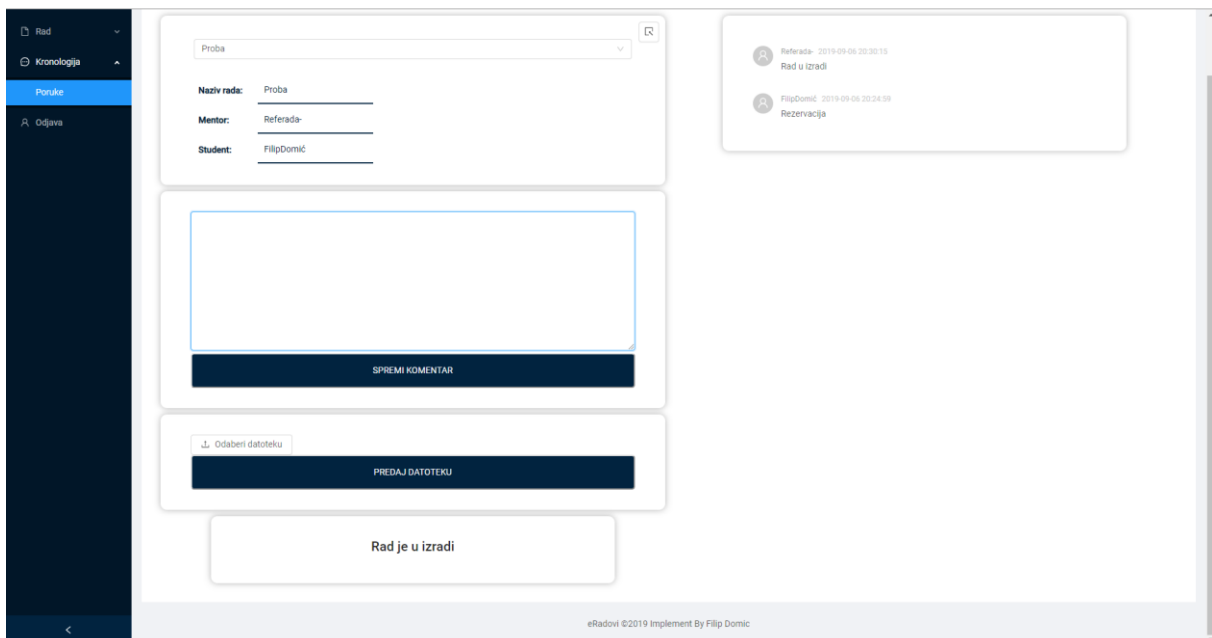
U slučaju da je rad odbijen student više ne može pisati komentar niti predati rad. I tako rad je poništen. Izgled navedenog ekrana prikazuje slika 36.



Slika 36. Rad odbijen

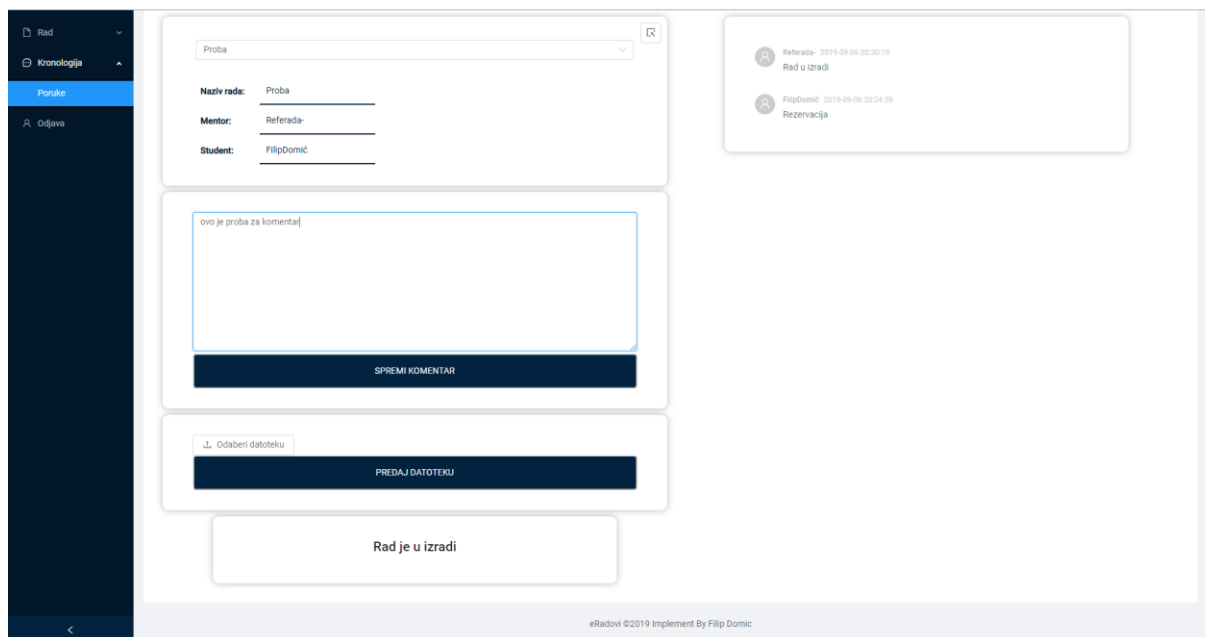
## 8.10. Upravljanje kronologijom

Naj bitnija mogućnost u ovoj aplikaciji je komunikacija između profesora i studenta. Kronologija je stranica koja omogućuje brz i jednostavan način komuniciranja s profesorom na samoj stranici ako smo ulogirani kao profesor možemo pri vrhu odabrati o kojem radu je riječ i stranica će nam otvoriti s desne strane poruke s drugom stranom odnosno sa studentom i sami profesor ima mogućnost preuzimanja pisanu verziju rada i ostaviti komentar studentu, i obratno tako može student poslati rad i ostaviti komentar. Izgled navedenog ekrana prikazuje slika 37.

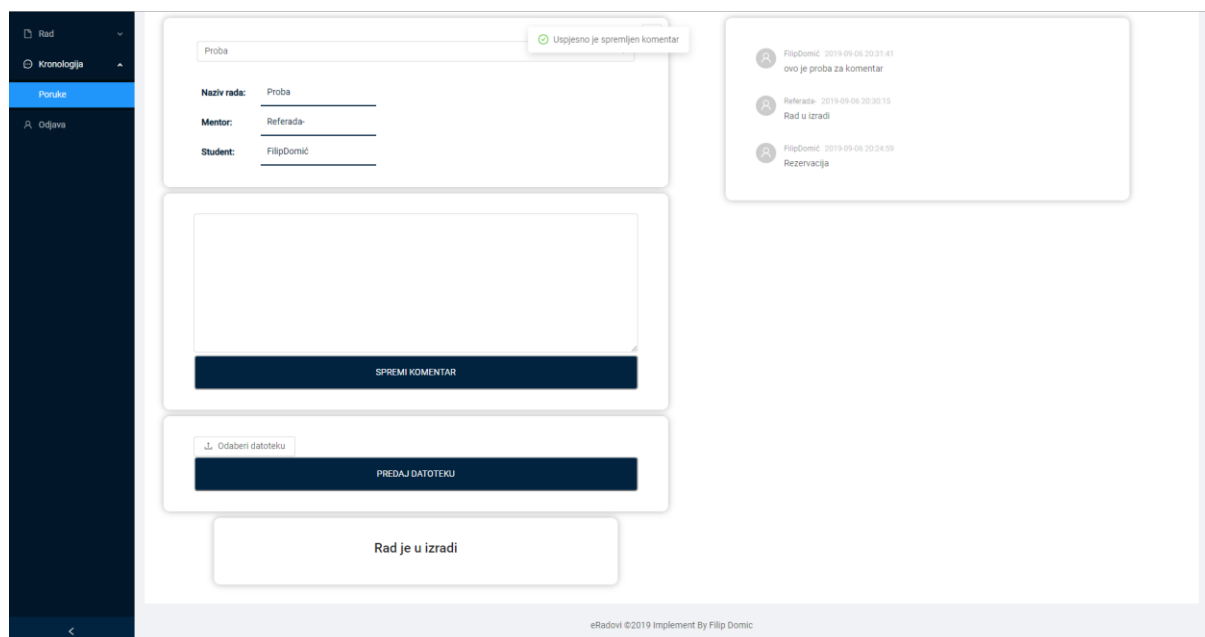


Slika 37. Kronologija

U daljnjem prikazivanju mogućnosti prikazat ćemo na koji način dodajemo komentar. Izgled navedenog ekrana prikazuje slika 38.

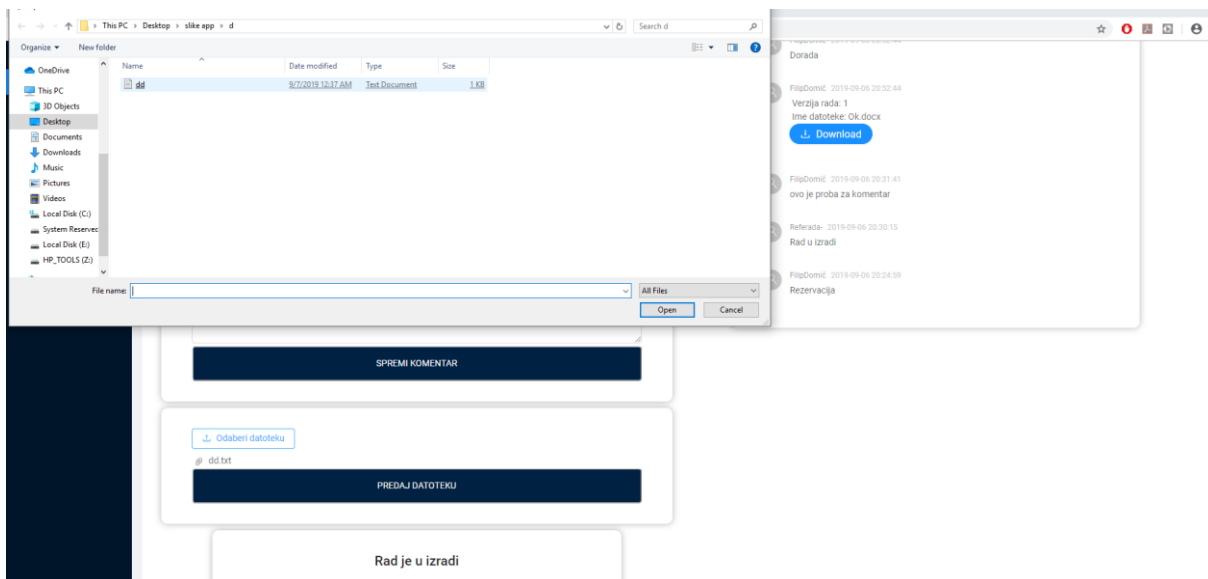


Imamo mogućnost kreiranja komentara upišemo u okvir komentar koji želimo ostaviti drugoj strani i ona se automatski ažurira s lijeve strane i druga strana može vidjeti naš komentar. Ako je komentar uspješno spremljen ispisat će se u tablici i ispisuje se obavijest da je komentar dobro spremljen. Izgled navedenog ekrana prikazuje slika 38.



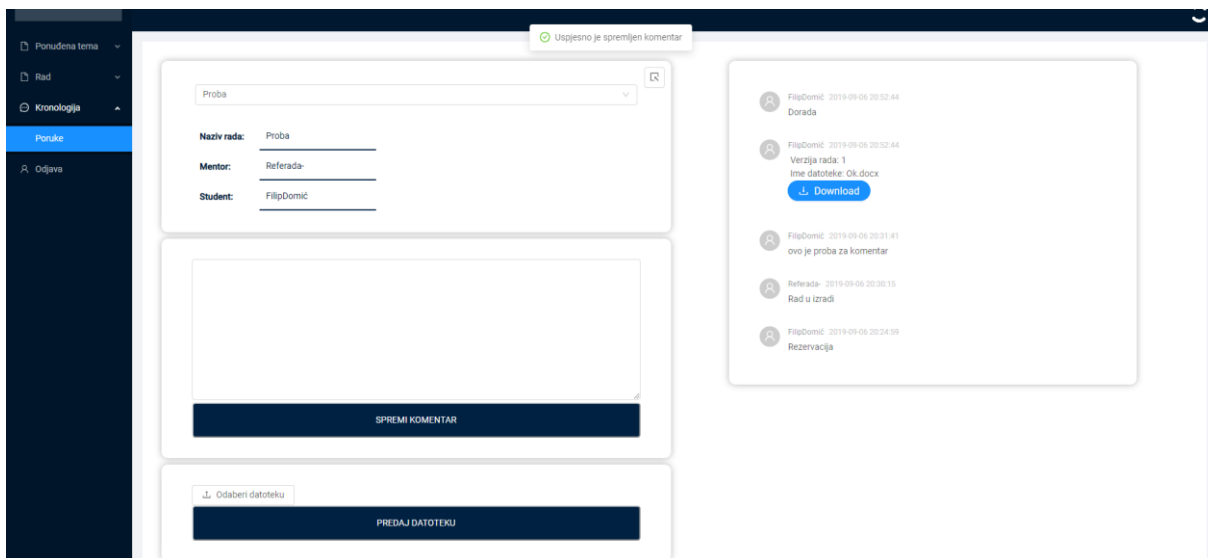
Slika 38. Komentar prihvaćen

Te također imamo mogućnost učitavanja datoteke. Izgled navedenog ekrana prikazuje slika 39



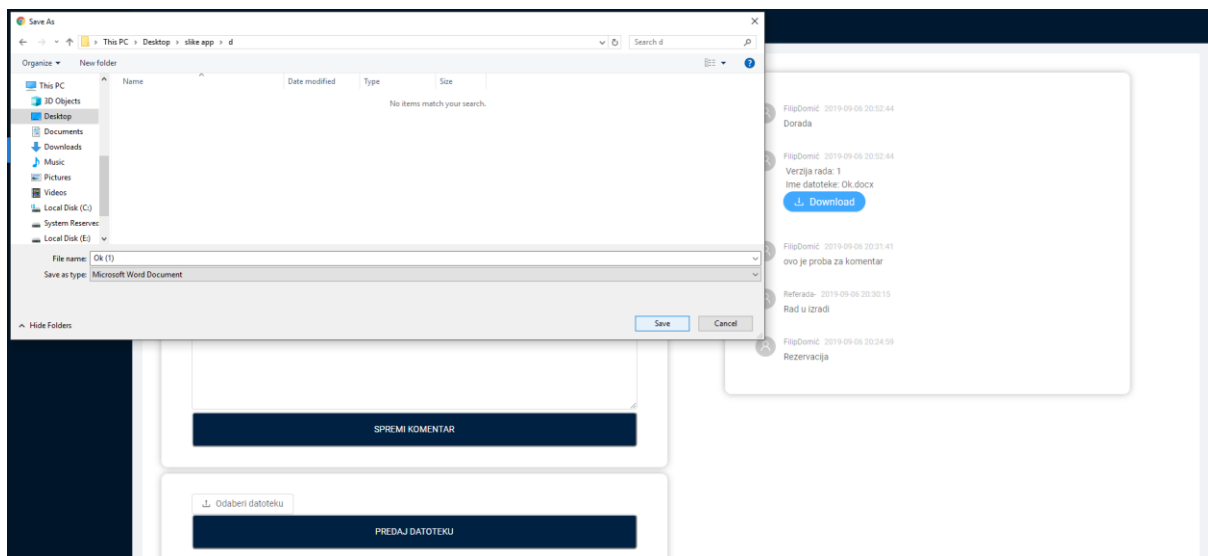
Slika 39. Učitavanje datoteke

Ako je učitana datoteka uspješno spremljen ispisiat će se u tablici i ispisuje se obavijest da je datoteka uspješno spremljena dobro spremljen. Izgled navedenog ekrana prikazuje slika 40.



Slika 40. Uspješno spremljena datoteka

Imamo mogućnost kada smo spremili datoteku da također se ista ta datoteka skinе na svoje računalo. U tablici odemo na gumb za skidanje stisnemo ga i otvori se mogućnost spremanja datoteke. prikazuje slika 41.




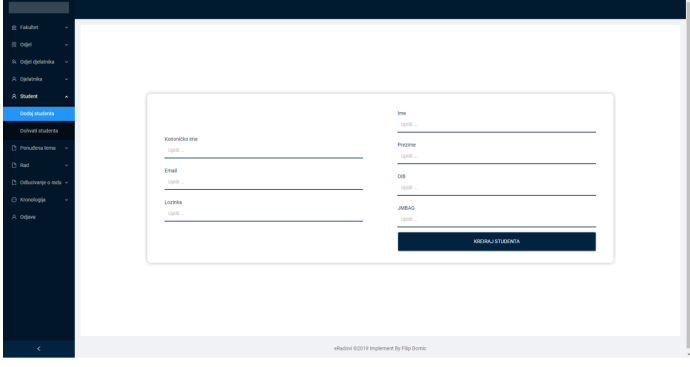
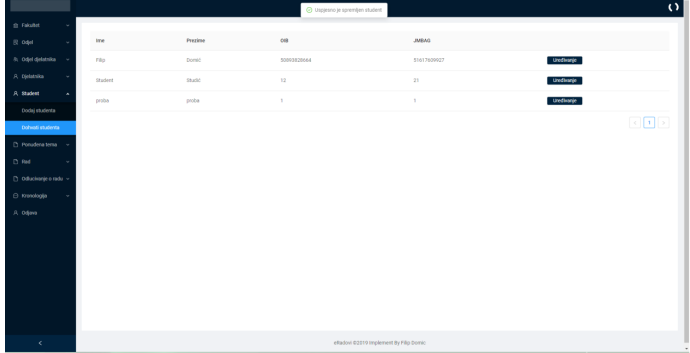
**Slika 41. Spremanje datoteke na računalo**



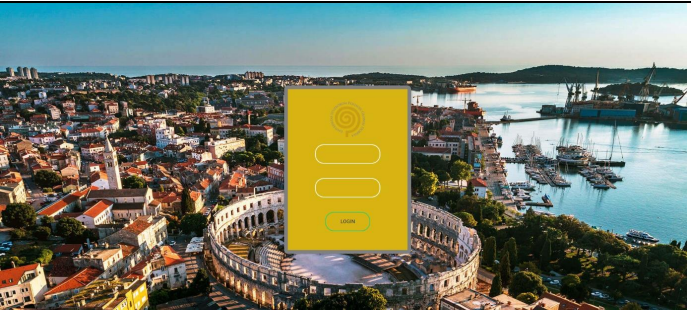
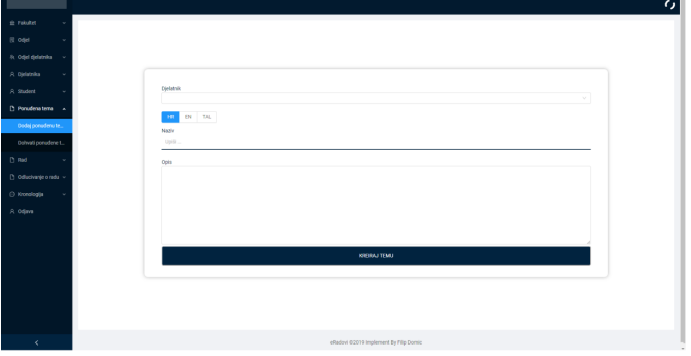
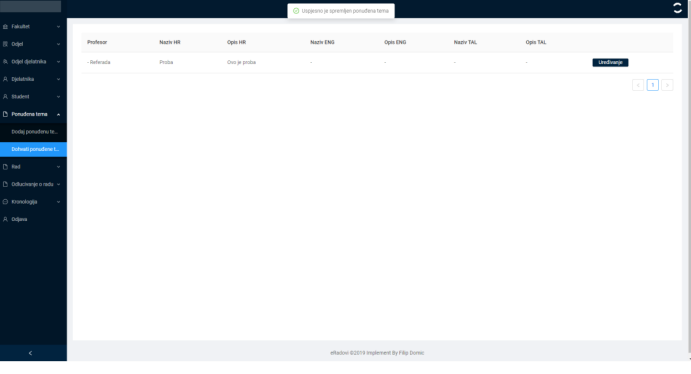
## 9. Hodogram kroz aplikaciju

U ovom poglavlju bit će prikazani neki od važnih hodograma kroz aplikaciju. Brojke koje možete vidjeti u uglatim zagrada predstavljaju poglavlja u radu gdje se može dobiti više informacija

### 9.1. Kreiranje studenta (Referada)


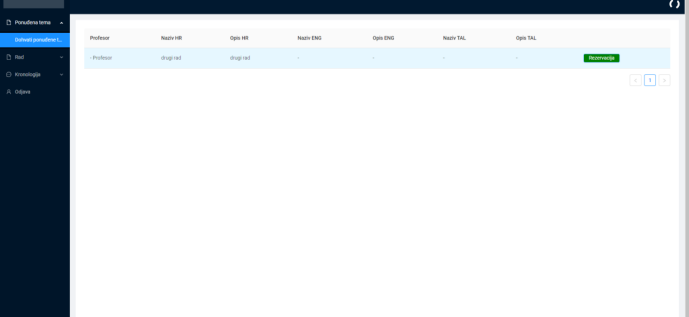
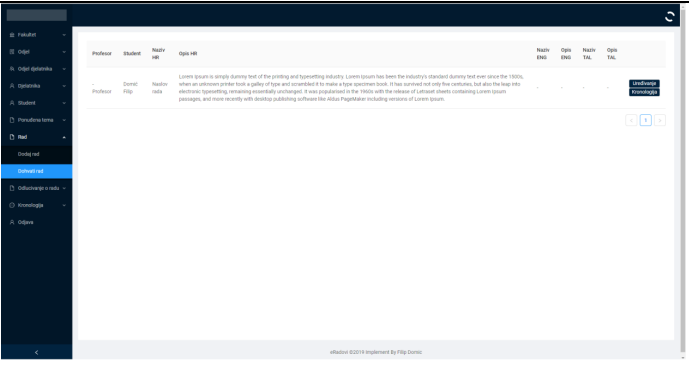
<p><b>Ekran za prijavu</b></p> <p>Pomoću ekrana za prijavu korisnik se prijavljuje u sustav. Za dodavanje studenta potrebno je prijaviti se s rolom referada [7.].</p> <p>[8.1.]</p>																					
<p><b>Ekran za dodavanje studenta</b></p> <p>Pomoću navigacijskog izbornika s lijeve strane korisnik se pozicionira na ekran za dodavanje studenta. Po ispunjavanju formulara koristi gumb za kreiranje zapisa.</p> <p>[8.5.]</p>																					
<p><b>Ekran za pregled studenata</b></p> <p>Nakon uspješnog spremanja korisnik je preusmjeren na ekran s pregledom svih zapisa.</p> <p>[8.5.]</p>	 <table border="1"><thead><tr><th>Ime</th><th>Prezime</th><th>OSB</th><th>JMBAG</th><th></th></tr></thead><tbody><tr><td>Filip</td><td>Štunc</td><td>558532054</td><td>5141762027</td><td><a href="#">Kreiraj</a></td></tr><tr><td>Student</td><td>Stuđić</td><td>12</td><td>21</td><td><a href="#">Kreiraj</a></td></tr><tr><td>prof.</td><td>prof.</td><td>1</td><td>1</td><td><a href="#">Kreiraj</a></td></tr></tbody></table>	Ime	Prezime	OSB	JMBAG		Filip	Štunc	558532054	5141762027	<a href="#">Kreiraj</a>	Student	Stuđić	12	21	<a href="#">Kreiraj</a>	prof.	prof.	1	1	<a href="#">Kreiraj</a>
Ime	Prezime	OSB	JMBAG																		
Filip	Štunc	558532054	5141762027	<a href="#">Kreiraj</a>																	
Student	Stuđić	12	21	<a href="#">Kreiraj</a>																	
prof.	prof.	1	1	<a href="#">Kreiraj</a>																	

## 9.2. Kreiranje ponuđenih teme (Profesor)

<p><b>Ekran za prijavu</b></p> <p>Pomoću ekrana za prijavu korisnik se prijavljuje u sustav. Za dodavanje tema potrebno je prijaviti se s rolom profesora [7.]. [8.1.]</p>	
<p><b>Ekran za kreiranje teme</b></p> <p>Pomoću navigacijskog izbornika s lijeve strane korisnik se pozicionira na ekran za dodavanje teme. Po ispunjavanju formulara koristi gumb za kreiranje zapisa. [8.6.]</p>	
<p><b>Ekran za pregled studenata</b></p> <p>Nakon uspješnog spremanja korisnik je preusmjeren na ekran s pregledom svih zapisa. [8.6.]</p>	

## 9.3. Prijava rada (Student)

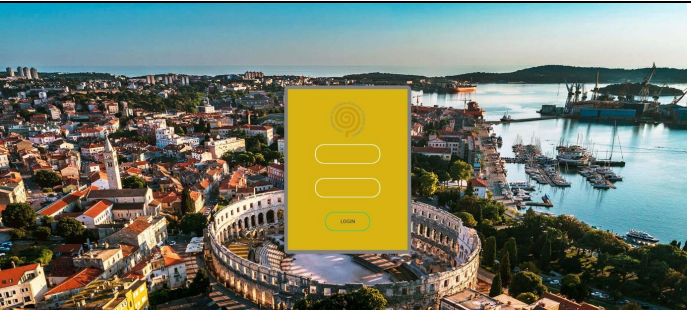

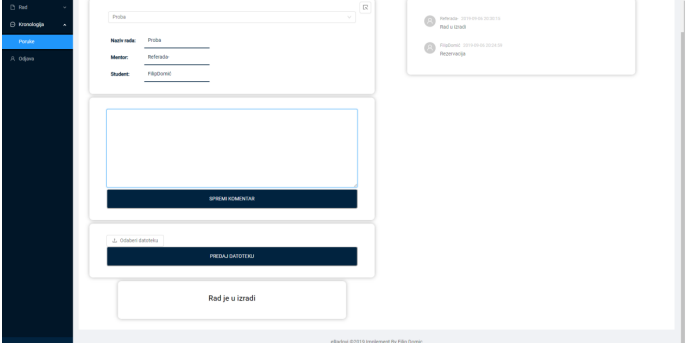
### 9.3.1. Ponuđena tema

<p>Ekran za prijavu</p> <p>Pomoću ekrana za prijavu korisnik se prijavljuje u sustav. Za prijavu rada potrebno je prijaviti se s rolom studenta [7.].</p> <p>[8.1.]</p>	
<p>Pregled i izbor teme</p> <p>Pomoću navigacijskog izbornika s lijeve strane korisnik se pozicionira na ekran za pregled teme. Izabire i rezervira svoju temu tako da klikne na gumb REZERVACIJA.</p> <p>[8.6.]</p>	
<p>Ekran za pregled radova</p> <p>Nakon uspješnog spremanja korisnik je preusmjeren na ekran s pregledom svih zapisa.</p> <p>[8.7.]</p>	

### 9.3.2. Unošenje vlastite teme

Hodogram je isti kao kada profesor kreira predloženu temu, samo u ovom slučaju prijavljen je korisnik s rolom student i u izborniku se izabere opcija Rad [8.7.].

## 9.4. Odobravanje rada (Profesor)

<p><b>Ekran za prijavu</b></p> <p>Pomoću ekrana za prijavu korisnik se prijavljuje u sustav. Za upravljanje statusom rada potrebno je prijaviti se s rolom profesora [7.].</p> <p>[8.1.]</p>	
<p><b>Potvrda ili odbijanjem teme</b></p> <p>Pomoću navigacijskog izbornika s lijeve strane korisnik se pozicionira na ekran poruke. Izabire rad iz padajućeg izbornika. Pomoću gumbova na dnu odlučuje o radu.</p> <p>[8.9.]</p>	
<p><b>Kronologija rada</b></p> <p>Ovisno o odabranoj opciji radu će biti pridijeljen status. Ako je odbijen pisat će da je odbijen, a ako je prihvaćen pisat će da je u izradi.</p> <p>[8.10.]</p>	

## 9.5. Komentiranje rada (Student)

<p>Ekran za prijavu</p> <p>Pomoću ekrana za prijavu korisnik se prijavljuje u sustav. Za upravljanje redom potrebno je prijaviti se s rolom studenta [7].</p> <p>[8.1.]</p>	
<p>Otvaranje rada</p> <p>Pomoću navigacijskog izbornika s lijeve strane korisnik se pozicionira na ekran poruke. Izabire rad iz padajućeg izbornika.</p> <p>[8.10.]</p>	
<p>Učitavanje dokumenta</p> <p>Odabire se opcija učitaj dokument gdje se otvara skočni prozor za izbor dokumenta.</p> <p>[8.10.]</p>	
<p>Pregled statusa</p> <p>Po učitavanju rad će biti prikazan na desnoj strani ekrana zajedno s ostalom korespondencijom između studenta i profesora.</p> <p>[8.10.]</p>	

## 10. Programski isječci

### 10.1. Autentifikacija

Klasa AutetifikaciskiGuardService služi nam za jednostavne provjere da li je web token (JWT token)postojan i koji stupanj pristupa korisnik posjeduje. Prva metoda canActivateChild u prvom djelu imamo dohvat tokena iz lokalnog spremnika (web token je token koji naše korisničke podatke pretvara u kriptirani objekt koji je vremenski ograničen). Spremamo ga u varijablu token i provjeravamo da li je on još pravovaljan, ako nije vratit će nas ponovo na stranicu prijave. Kada je token pravovaljan i metoda je vratila true prelazimo u sljedeću metodu gdje će se odrediti koji stupanj ovlasti imamo. Na početku metode provjeravamo dali smo se spojili s profesorom, studentom ili referadom. Kada smo odredili s kojom smo se osobom prijavili u sustav prema tim ovlastima osoba dodjeljujemo određene prozore. Ako smo se spojili s referadom možemo pristupiti prozoru fakulteti, te svim pod stranicama koje su vezane uz fakultete. Ako smo pristupili kao profesor nemamo pravo pristupiti prozoru fakulteti, kako je vidljivo u kodu ako želimo „divljački“ upisati u url e-radovi/fakulteti neće nam uspjeti. Ulazimo u if i provjeravamo za referadu da li je prijavljena, ako je prijavljena vraća true, a za sve ostale vraća false i tako nam radi ograničenje da ne možemo pristupiti prozoru fakulteti, te da ne možemo samo preko urla pokušati ući u taj prozor. Izgled navedenog koda prikazan na slici 42.

```
1 import { Injectable } from "@angular/core";
2 import {
3   canActivateChild,
4   ActivatedRouteSnapshot,
5   RouterStateSnapshot
6 } from "@angular/router";
7
8 @Injectable()
9 export class AutentifikacijaGuardService implements CanActivateChild {
10   constructor() {}
11
12   canActivateChild(route: ActivatedRouteSnapshot, state: RouterStateSnapshot ): boolean {
13     let token = localStorage.getItem("AUTH_TOKEN");
14     if(token === null) window.location.href = '/';
15
16     let isOk = token !== null && AutentifikacijaGuardService.provjera(state.url);
17
18     if(isOk == false) {
19       window.location.href = '/';
20     }
21
22     return isOk;
23   }
24
25   public static provjera(tip: string): boolean {
26     // return true;
27     let isProfesor = localStorage.getItem("AUTH_isProfesor");
28     let isReferada = localStorage.getItem("AUTH_isReferada");
29     let isStudent = localStorage.getItem("AUTH_isStudent");
30
31     if(tip === 'fakultet' || tip.indexOf('/e-radovi/fakultet') !== -1 || tip.indexOf('/e-radovi/dohvati-fakultet') !== -1) {
32       if(isReferada == "true") return true;
33       return false;
34     }
35
36     if(tip === 'odjeli' || tip.indexOf('/e-radovi/odjel') !== -1 || tip.indexOf('/e-radovi/dohvati-odje') !== -1) {
37       if(isReferada == "true") return true;
38       return false;
39     }
40   }
41 }
```

Slika 42. Programski isječak - Autentifikacija

## 10.2. Dodaj Fakultet

Klasa Fakultet služi nam za spremanje novog fakulteta ili uređivanje nekog fakulteta. Na početku imamo definirane neke od varijabla koje će se koristiti u daljnjem kodu. Na početku imamo konstruktor koji inicijalizira sami prozor. Nakon konstruktora imamo metodu ngOnInit odnosno metoda koja će se pokrenuti kada se taj prozor otvori. Ovu metodu mi smo koristili za uređivanje nekog od fakulteta. Odnosno ako smo odlučili iz tablice fakulteta promijeniti naziv poslat će se njegov id koji će se otvoriti u prozoru dodaj fakultet. Prva linija koda u metodi služi za spremanje tog poslanog id u našu praznu varijablu (kada se nalazimo u prozoru dohvati fakultete i kada smo odlučili stisnuti uređivanje id tog fakulteta poslat će se pomoću urla, i na ovaj način će se izvući id iz urla). Kada varijabla nije prazna krećemo dalje u if i na dohvat fakulteta. Kada smo dohvatili fakultet u daljnjem if-u provjeravamo dali je varijabla fakultet prazna, odnosno da li smo uspjeli pronaći traženi fakultet, ako je prazna preusmjerava nas nazad na dohvati fakultete jer je očito došlo do neke greške. Ako je sve uredi popunjava se atribut na stranici, u ovom slučaju naziv. Izgled navedenog koda prikazan na slici 43. Opisani scenarij pokreće se samo u slučajevima kada se editira postojeći zapis. Kod dodavanja novog ne šalje se id i ne rade se provjere nad njime.

```
7
8 @Component({
9   selector: "app-fakultet",
10  templateUrl: "../fakultet.component.html",
11  styleUrls: ["../fakultet.component.scss"]
12 })
13 export class FakultetComponent implements OnInit {
14
15   public id;
16
17   public myForm: FormGroup;
18
19   public fakulteti: Array<any> = [];
20
21   constructor(
22     private fb: FormBuilder,
23     private apiService: ApiService,
24
25     private message: NzMessageService,
26     private route: ActivatedRoute,
27     private router: Router
28   ) {
29     this.init();
30   }
31
32
33   ngOnInit(): void {
34     this.id = this.route.snapshot.paramMap.get("id");
35
36     if (this.id) {
37
38       this.apiService.dohvatiFakultete(this.id).subscribe(response => {
39         let fakultet = response.data;
40
41         if (!fakultet) {
42           this.router.navigate(['/e-radovi/dohvati-fakultet']);
43           return;
44         }
45
46         this.myForm.patchValue({
47           naziv: fakultet.naziv
48         });
49       });
50     }
51   }
```

Slika 43. Programski isječak - Dodaj fakultet



U drugom djelu koda za dodaj fakultet imamo metodu `init` u kojoj pozivamo druge dvije metode `dohvati fakultet` i `kreiraj formu`. Prva metoda `dohvati podatke` kao što samo ime kaže radimo dohvat svih podataka za fakultete radi poslije provjere. Druga metoda koja kreira praznu formu radi upisa nekog novog fakulteta, u tu formu će se upisivati neki od naziva fakulteta tako da će korisnik preko prozora upisati naziv fakulteta i on će se spremiti u formu. Kada se neki do naziva upisali krećemo dalje u metodu `kreiraj fakultet`. Prvo će se povući rezultat iz forme odnosno naziv će se spremiti u varijablu. Poslije će se u `if-u` provjeravati dali slučajno u tablici postoji isti naziv fakulteta, ako postoji odbiti će se unos fakulteta. Ako je unos dobar i zadovoljava sve uvjete spremit će se u tablicu fakulteta. I zadnja metoda je metoda za poruke, dali je fakultet spremljen ili nije. Izgled navedenog koda prikazan na slici 44.

```
52
53 private init(): void {
54     this.kreirajFormu();
55     this.dohvatiPodatke();
56 }
57
58 private dohvatiPodatke(): void {
59     this.apiService.dohvatiFakultete().subscribe(
60         response => this.fakulteti = response.data,
61
62         error => console.log(error)
63     );
64 }
65
66 private kreirajFormu(): void {
67     this.myForm = this.fb.group({
68         naziv: ""
69     });
70 }
71
72 // ===== HTML METODE =====
73
74 public kreirajFakultet(): void {
75     let rezultat = this.fakulteti.find(fakultet => fakultet.naziv === this.myForm.value.naziv);
76
77     if (rezultat && (!this.id || this.id !== rezultat.id)) {
78         this.createMessage("error", `Fakultet s upisanim nazivom već postoji`);
79     } else {
80
81         this.apiService
82             .kreirajFakultet(this.myForm.value.naziv, this.id)
83             .subscribe(
84                 response => this.createMessage("success", `Uspješno je spremljen fakultet`),
85                 error => console.log(error, this.createMessage("error", `Spremanje nije uspjelo`))
86             );
87     }
88 }
89
90 public createMessage(type: string, poruka: string): void {
91     if (type === "success") {
92         this.message.create(type, poruka);
93         this.router.navigate(['/e-radovi/dohvati-fakultet']);
94     } else {
95         this.message.create(type, poruka);
96     }
97 }
98
99 }
```

Slika 44. Programski isječak - Dodaj fakultet 2 dio



### 10.3. Kronologija

Ovaj dio koda je glavna funkcionalnost aplikacije. Prikazana nam je metoda dohvati komentare na samome početku. Dohvatit će se rad o kojem treba ispisati komentare i kronologija rada(u kronologiji rada za sam rad će se dobiti sve informacije o radu i svi komentari koje je neko ostavio na rad). Kada smo dohvatili podatke prvo će se za rad napraviti neke provjere. Provjeru smo nazvali odbijen ili zatvoren. Napravit će se provjera na način za rad prema njegovom id otići će se i provjerit će se u kojem je stadiju status rada. Ako je status rad u stadiju odbijen to znači da profesor nije odobrio rad i nije moguće da pišemo komentar ili predamo rad. Ako nije status odbijen ili zatvoren rad će se ispisati i sve funkcionalnosti će se moći koristiti. Kada se ta provjera obavila i svi uvjeti su zadovoljeni popunjavamo formu s nazivom rada, mentorom te imenom i prezimenom studenta. Kada se popunila forma ispisujemo kronologiju i počinjemo sa sljedećom metodom polling. Izgled navedenog koda prikazan na slici 45.

```
77 private dohvatiKomentare(): void {
78     forkJoin(
79         this.apiService.dohvatiRad(this.selectedId),
80         this.apiService.Kronologija(this.selectedId)
81     ).subscribe((response) => {
82         this.rad = response[0].data;
83
84         let odbijen_ili_zatvoren = this.rad.status.find(x => x.statusi_rada_id == 3 || x.statusi_rada_id == 7);
85         if(odbijen_ili_zatvoren) {
86             this.rad.status = odbijen_ili_zatvoren;
87         } else {
88             this.rad.status = this.rad.status[this.rad.status.length -1];
89         }
90
91         this.myForm.patchValue({
92             naziv_hr: this.rad.naziv_hr,
93             profesor: this.getPunoIme(-1, this.rad.djelatnik_id),
94             student: this.getPunoIme(this.rad.student_id, -1)
95         });
96
97         this.kronologija = response[1].data;
98
99         this.startPolling();
100     });
101 }
```

Slika 45. Programski isječak - Dohvat Kronologije

Imamo metodu `polling`. Ova metoda nam služi kao što samo ime kaže za guranje podataka odnosno automatsko osvježavanje stranice. Metoda je napravljena da svakih 5 sekundi napravi osvježavanje stranice kao što je vidljivo u kodu. Prvo se pokrene štoperica kada dođe do neke izmjene štoperica se zaustavlja te se postavlja na nulu i radi se dohvat svih podataka i tako svakih 5 sekundi. Izgled navedenog koda prikazan na slici 46.

```
102 // ===== POLLING =====
103
104
105 private pollingTimerInstance = null;
106
107 private startPolling(forceFetch?: boolean): void {
108     this.pollingTimerInstance = setTimeout(() => {
109         if(!this.pollingTimerInstance) return;
110         this.stopPolling();
111         this.dohvatiKomentare();
112     }, forceFetch ? 0 : 5 * 1000);
113 }
114
115
116 private stopPolling(): void {
117     clearTimeout(this.pollingTimerInstance);
118     this.pollingTimerInstance = null;
119 }
120
```

Slika 46. Programski isječak - Polling

## **11. Prijedlozi za poboljšanja**

Prijedlozi za buduće dorade moglo bi biti u samome dizajnu da se aplikacija prilagodi mobilnim uređajima. Također bi bilo korisno uvesti gumb za brisanje kako bi referada mogla čistiti odbijene radove. Te bi se moglo poraditi na boljem sustavu obavijesti, gdje se status samoga rada promjeni da na način da dođe obavijest na e mail da je profesor ili student promjenio status svoga rada.

## 12. Zaključak

Konverzija tradicionalnog načina obavljanja nekih stvari uvijek je težak korak, ali nije nemoguć. Taj postupak je isto tako popraćen raznim pogodnostima nakon što je izvršen i iz tog razloga je nužan korak ka naprijed. Ustaljeni sustavi i rutine nisu primamljivi te im nikako ne šteti uvoz inovacija jedna od kojih je ova aplikacija. Niz mogućnosti koje pruža ima jednu bitnu karakteristiku, u ovom brzom napretku današnjice u slučaju da se smisli novi, bolji i brži način obavljanja procedure pisanja radova uvijek se može nadograditi novim ažuriranjem i ne zahtijeva komplicirane postupke izmjene. Drugo, kratko rečeno ovo razmišljanje čini se logičnim i potrebnim s obzirom na korištenje tehnologije koja osim puke zabave može poslužiti kao vrlo koristan alat. Kroz ovaj završni rad cilj je bio dosadni standardni način prijave rada na papiru zamijeniti web aplikacijom koja će biti dostupna svim studentima i lako uporabljiva. Kroz samu aplikaciju uspješno je napravljena komunikacija dvije strane profesora i studenta, i tako olakšala prijave završnih i diplomskih radova, te slanje tih istih radova na prepravak profesoru, tako je uspješno napravljeno čuvanje svih završnih i diplomskih radova na jednom mjestu. Važno je naglasiti da aplikacijom je vrlo jednostavno rukovati, nije potrebno neko veliko informatičko predznanje. Web aplikaciju radi lakšeg razvoja sustava programiranje je razdvojeno na dva sloja, a to su klijentski sloj i serverski sloj. Klijentski razvoj smo pisali u vrlo danas poznatim web programima HTML, CSS i JavaScript. Time smo napravili dobar temelj za razvojni okvir Angular, te smo uz to koristili biblioteku NG-Zorro. Serverski razvoj napravljen je u programskom jeziku PHP s podrškom za MySQL bazu podataka. Prilikom serverskog programiranja također je korišten razvojni okvir PHP Laravel. Kada sve navedeno zbrojimo dobivamo modernu i vrlo jednostavnu web aplikaciju za korištenje.

## 13. Popis literature

### Internet izvori:

1. Laravel : <https://laravel.com/docs/6.0> , [Pristupljeno 3. ožujak.2019]
2. Angular , <https://angular.io/guide/http>, [Pristupljeno 1. travnja.2019]
3. NgZorro , <https://ng.ant.design>, [Pristupljeno 20. travnja.2019]
4. GitHub, <https://github.com/>, [Pristupljeno 3. ožujak.2019]
5. W3schools: <https://www.w3schools.com/>, [Pristupljeno 3. ožujak.2019]
6. JWT Token : <https://jwt.io/introduction/> [Pristupljeno 3. rujan..2019]
7. Microsoft(2019), *Sigurnosne uloge i privilegije*: <https://docs.microsoft.com/hr-hr/dynamics365/customer-engagement/admin/security-roles-privileges>  
[Pristupljeno 7. kolovoz..2019]

### Znanstveni članci i publikacije:

1. Vučković, A. (2016.). *Faze upravljanja projektom na primjeru projekta rekonstrukcije he zakućac*. Split. <https://repozitorij.efst.unist.hr/islandora/object/efst%3A935/datastream/PDF/view> [Pristupljeno 3. rujan..2019]
2. Barać, A. (2014.). *Razvoj dinamičkih web aplikacija*. Rijeka. <https://www.pfri.uniri.hr/knjiznica/NG-dipl.EITP/219-2014.pdf> [Pristupljeno 3. rujan..2019]
3. Maltar, J.(2017). *Angular*. Osijek. <https://zir.nsk.hr/islandora/object/mathos:142/preview> [Pristupljeno 3. rujan..2019]
4. Prelec, A.(2015.) *Izrada responzivne internetske stranice upotrebom html5 i css3 tehnologija*. Zagreb. [https://eprints.grf.unizg.hr/2293/1/Z611\\_Prelec\\_Andrija.pdf](https://eprints.grf.unizg.hr/2293/1/Z611_Prelec_Andrija.pdf) [Pristupljeno 3. rujan..2019]
5. Valjak, S. (2015). *Mysql sustav za upravljanje bazama podataka otvorenog koda*. Gospić. <https://zir.nsk.hr/islandora/object/velegs%3A487/datastream/PDF/view>  
[Pristupljeno 3. ožujak.2019]

6. Kemić, Ž. (2013), *Primjer baze podataka u sustavu mysql* .Varaždin.  
<https://docs.microsoft.com/hr-hr/dynamics365/customer-engagement/admin/security-roles-privileges> [Pristupljeno 3. ožujak.2019]
7. Dumbovic, V. (2017.) *Razvoj aplikacije u laravel okruzenju*. Zagreb.  
<https://repozitorij.pmf.unizg.hr/islandora/object/pmf%3A631/datastream/PDF/view> [Pristupljeno 3. rujan.2019]

## Popis slika, tablica, grafova

Slika 1. Use-case Student .....	2
Slika 2. Use-case Profesor .....	3
Slika 3. Excel tablica podjela projekta na zadatke .....	4
Slika 4. Stranica za prijavu .....	10
Slika 5. Dohvat fakulteta .....	11
Slika 6. Dodaj fakultet .....	11
Slika 7. Dodavanje novog fakulteta .....	12
Slika 8. Potvrda spremanja fakulteta .....	12
Slika 9. Dodaj postojeći fakultet .....	13
Slika 10. Tablica odjela .....	14
Slika 11. Dodaj odjel .....	14
Slika 12. Dodavanje novog odjela .....	15
Slika 13. Odjel uspješno spremljen .....	15
Slika 14. Tablica djelatnika .....	16
Slika 15. Dodaj djelatnika .....	16
Slika 16. Dodavanje novog djelatnika .....	17
Slika 17. Djelatnik uspješno spremljen .....	17
Slika 18. Greška krivi unos podataka .....	18
Slika 19. Tablica studenata .....	19
Slika 20. Dodaj studenta .....	19
Slika 21. Dodavanje novog studenta .....	20
Slika 22 Student uspješno spremljen .....	20
Slika 23. Tablica ponuđene teme .....	21
Slika 24. Dodaj ponuđenu temu .....	21
Slika 25. Dodavanje ponuđene teme .....	22
Slika 26. Ponuđena tema prihvaćena .....	22
Slika 27. Tablica radova .....	23
Slika 28. Dodaj rad .....	24
Slika 29 Dodavanje rada .....	24
Slika 30. Rad prihvaćen .....	25
Slika 31. Tablica ponuđene teme .....	26
Slika 32. Rezervacija .....	26
Slika 33. Rezervirana tema ne prikazuje se više u tablici .....	27
Slika 34. Odluke o radu .....	28
Slika 35. Rad prihvaćen .....	28
Slika 36. Rad odbijen .....	29
Slika 37. Kronologija .....	30
Slika 38. Komentar prihvaćen .....	31
Slika 39. Učitavanje datoteke .....	32
Slika 40. Uspješno spremljena datoteka .....	32
Slika 41. Spremanje datoteke na računalo .....	33
Slika 42. Programski isječak - Autentifikacija .....	39
Slika 43. Programski isječak - Dodaj fakultet .....	40
Slika 44. Programski isječak - Dodaj fakultet 2 dio .....	41
Slika 45. Programski isječak - Dohvat Kronologije .....	42
Slika 46. Programski isječak - Polling .....	43

## **Sažetak**

Ideja je krenula od profesora koji je smatrao da bi informatički fakultet sve svoje aplikacije i stranice trebale biti napravljene od strane studenata. Te je isti taj profesor predložio temu za web aplikaciju za završne i diplomske radove. Upravo je to tema ovoga završnog rada. U početku kreiranja rada temeljno je napravljeno planiranje projekta. Poslije dobrog planiranja projekt smo podijelili na dva djela klijentski sloj i serverski sloj. Kada smo ta dva djela napisali odredili smo ovlasti u aplikaciji. I na kraju smo dodali dizajn. Glavni cilj je bio zamijeniti klasični način na papiru, te taj formular s papira prenijeti u web aplikaciju.

Ključne riječi: aplikacija, web, završni radovi, diplomski radovi, prijave radova



## **Abstract**

The idea came from a professor who felt that the Faculty of Informatics should have all its applications and sites made by students. And the same professor suggested a theme for a web application for graduate and graduate work. This is precisely the theme of this final paper. Project planning was thoroughly done at the beginning of the creation of the work. After good planning, we divided the project into two parts: client layer and server layer. When we wrote these two works, we determined the powers of the application. And finally, we added the design. The main goal was to replace the classic paper method and transfer that form from paper to a web application.

Keywords: application, web, final paper, graduate work, application submission