Web sustav za administraciju projektnih prijedloga i prezentaciju rezultata projekata

Karašin Štifter, Alan

Undergraduate thesis / Završni rad

2023

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://urn.nsk.hr/urn:nbn:hr:190:654465

Rights / Prava: Attribution 4.0 International/Imenovanje 4.0 međunarodna

Download date / Datum preuzimanja: **2024-05-09**

Repository / Repozitorij:

Repository of the University of Rijeka, Faculty of Engineering

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Sveučilišni prijediplomski studij računarstva

Završni rad

Web sustav za administraciju projektnih prijedloga i prezentaciju rezultata projekata

Rijeka, lipanj 2023.

Alan Karašin Štifter

0069087964

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Sveučilišni prijediplomski studij računarstva

Završni rad

Web sustav za administraciju projektnih prijedloga i prezentaciju rezultata projekata

Mentor: izv. prof. dr. sc. Sandi Ljubić

Komentor: prof. dr. sc. Saša Vlahinić

Rijeka, lipanj 2023.

Alan Karašin Štifter

0069087964

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 10. ožujka 2023.

Zavod: **Zavod za računarstvo**
Predmet: **Baze podataka**
Grana: **2.09.01 arhitektura računalnih sustava**

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Alan Karašin Štifter (0069087964)**
Studij: Sveučilišni prijediplomski studij računarstva

Zadatak: **Web sustav za administraciju projektnih prijedloga i prezentaciju rezultata projekata / A Web System for Administration of Project Proposals and Presentation of Project Results**

Opis zadatka:

Modelirati relacijsku bazu podataka za sustav administracije projektnih prijedloga i prezentaciju rezultata projekata. Razviti web aplikaciju koja registriranim korisnicima različitih uloga omogućava: (i) prijavu novih projektnih prijedloga putem definiranih obrazaca, (ii) uređivanje i objavljivanje rezultata aktivnih projekata, te (iii) administraciju projektnih prijedloga te aktivnih i arhiviranih projekata. Na poslužiteljskoj strani moguće je integrirati proizvoljan sustav za upravljanje bazama podataka, dok je na klijentskoj strani potrebno usmjeriti pažnju na dizajn responzivnog korisničkog sučelja.

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

Zadatak uručen pristupniku: 20. ožujka 2023.

Mentor:

Izv. prof. dr. sc. Sandi Ljubić

Prof. dr. sc. Saša Vlahinić (komentor)

Predsjednik povjerenstva za završni ispit:

Prof. dr. sc. Miroslav Joler

Izjava

o samostalnoj izradi **Završnog rada**

Izjavljujem pod punom materijalnom i moralnom odgovornošću da sam ovaj rad izradio/la samostalno te da u njemu nema kopiranih ili prepisanih dijelova teksta tuđih radova, a da nisu propisano označeni kao citati s navedenim izvorom iz kojeg su preneseni.

U Rijeci, 11.7.2023.

Alan Karažin Štifter

(vlastoručni potpis)

ZAHVALA

Veliku zahvalnost, u prvom redu, želim iskazati svom mentoru, izv. prof. dr. sc. Sandi Ljubiću, što mi je, uz veliko strpljenje i trud, svojim znanjem i stručnim savjetima, pomogao u izradi ovog završnog rada.

Posebnu zahvalnost, dugujem svojoj obitelji za svu podršku koju su mi pružili tijekom života.

I na kraju, najveću zaslugu za ono što sam postigao pripisujem svojim roditeljima koji su uvijek bili uz mene kada mi je to trebalo i bez čije potpore ovo što sam dosada postigao ne bi bilo moguće.

SADRŽAJ

1. UVOD 6

2. POSTOJEĆI PROCES PRIJAVE NA NATJEČAJE I PLAN DIGITALIZACIJE ZASNOVAN NA KORIŠTENJU BAZE PODATAKA 8

2.1. Opći postupak dizajniranja baza podataka 9

3. IZRADA WEB SUSTAVA ZA ADMINISTRACIJU PROJEKTNIH PRIJEDLOGA I PREZENTACIJU REZULTATA PROJEKATA 13

3.1. Tehnološki stog za izradu programskog rješenja 13

3.2 Modeliranje baze podataka 15

3.3. Implementacijski detalji od posebnog značaja 18

3.4. Korisničke uloge u sustavu 19

4. OPIS SLUČAJEVA KORIŠTENJA 20

4.1. Registracija i prijava u sustav 20

4.2. Hodogram projektnog prijedloga 22

4.3. Administracija evaluatora 28

4.4. Administracija natječaja i njihovih prijava 29

4.5. Korištenje na zaslonima različitih veličina 32

4.6. Pregled rezultata natječaja 33

5. ZAKLJUČAK 35

LITERATURA 37

POPIS OZNAKA I KRATICA 39

SAŽETAK 40

DODACI 41

Dodatak A Primjer obrasca za ručnu prijavu 41

1. UVOD

Sustavi moderne komunikacije i interakcija u društvu su u eksponencijalnoj progresiji. Ako promatramo razvoj informatičkih sustava i web aplikacija od početka ovog milenija možemo zaključiti da se radi o jednom rapidnom evolucijskom procesu u kojem smo istovremeno svjedoci i akteri. Stoga nas ne smije čuditi da je u današnjem svijetu, složenost sustava i broj korisnika koji ih koriste u kontinuiranom porastu [1, 2].

Posljedično je potreba za automatizacijom, kako sustava, tako i različitih procesa koji su u uzročno -posljedičnoj vezi, također u porastu. Digitalizacija predstavlja sistematizirani pristup koji omogućava korisnicima skraćivanje procesnih koraka sa u konačnici boljim pregledom svake procesne faze pa tako i pregledom nad cijelim procesom [3].

U tu se svrhu koriste web sustavi. Web sustav CMS (engl. Content management system) omogućuje upravljanje sadržajem. Njegova svrha je izrada rješenja s ciljem klasifikacije i organiziranja sadržaja te povezivanja i uređivanja sadržaja. Iako se pojam može koristiti za manualne procese upravljanja sadržajem, danas ga prvenstveno koristimo za različita programska rješenja koja omogućuju napredno upravljanje velikim brojem informacija [4].

U današnje vrijeme se CMS sustavi koriste za usklađivanje podataka iz više izvora, provedbu suradnih projekata, te organiziranje radnih procesa u poslovnim okruženjima i slično.

Temeljna primjena CMS-a danas je u dinamičkom stvaranju web-stranica nove generacije, a ovaj završni rad prikazuje njegovu primjenu na primjeru izrade aplikacije za administriranje projektnih prijedloga i prezentaciju rezultata projekata.

Pri tome se vodilo računa o preduvjetima koje kvalitetni Web CMS sustav mora ispunjavati da bi ga, korisnici koji su često bez posebnih tehničkih i programerskih vještina, mogli s lakoćom koristiti te upravljati sadržajima stranica [5]. Taj cilj je ostvaren kroz jedinstvenu podatkovnu strukturu koja omogućuje preglednu organizaciju svih sadržaja i podataka.

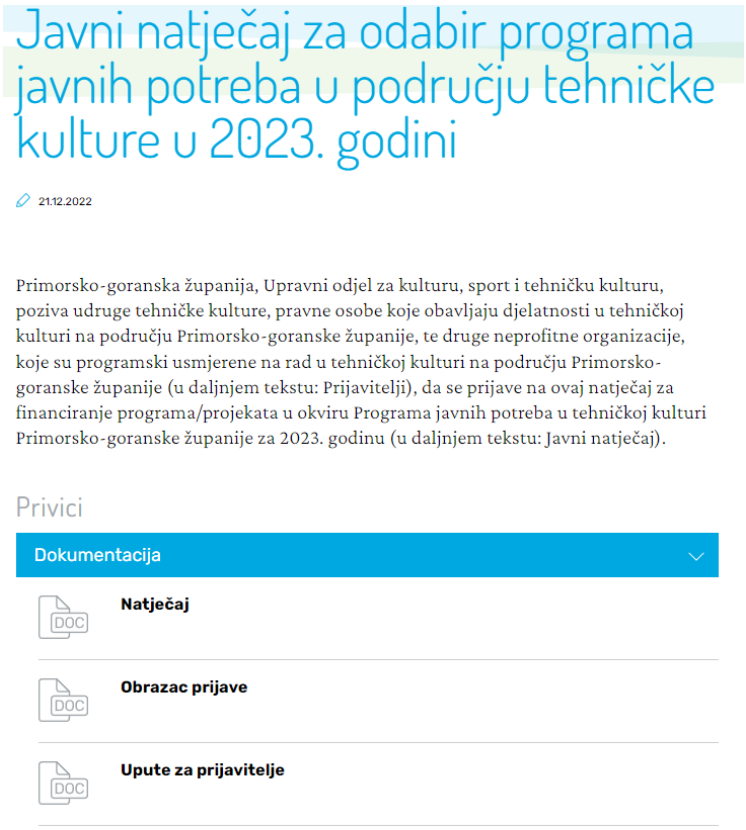
Prilikom izrade aplikacije se nastojalo da ista uključi neke od karakteristika koje se navode u nastavku, a koje se u literaturi opisuju poželjnima kod izrade kvalitetne aplikacije. To su, između ostalog, mogućnosti pretraživanja predanih projektnih prijava, sustav različitih prava pristupa zasnovan na ulogama (u ovom konkretnom slučaju prijavitelj projekta ili aplikant te evaluator ili ocjenjivač), sustav tijeka zadataka, upravljanje korisnicima i korištenje prilagođeno krajnjem korisniku.

Prilikom izrade aplikacije uključene su i neke od tzv. naprednih funkcija poput ugrađenih prozora za uređivanje teksta prilikom ispunjavanja obrazaca, te standardne sigurnosne postavke kod pristupa aplikaciji kroz registraciju, korisničko ime i zaporku, po uzoru na dobre primjere iz prakse [6].

Kroz poglavlja se prikazao postupak implementacije rješenja koji je uključio i organiziranje relacijske baze podataka sa standardnim koracima, od konceptualnog modeliranja podataka pa sve do normalizacije i apliciranja u odabrani sustav za upravljanje bazama podataka. Taj postupak uključivao je, između ostaloga, i utvrđivanje opisnih atributa i njihove međusobne povezanosti, te u konačnici logičko modeliranje podataka (kojim se utvrdila „logika" rasporeda podataka i relacija kroz tablični prikaz) [5-7].

2. POSTOJEĆI PROCES PRIJAVE NA NATJEČAJE I PLAN DIGITALIZACIJE ZASNOVAN NA KORIŠTENJU BAZE PODATAKA

Postojeći proces prijave na natječaje nije dostupan u digitalnom obliku. Radi se o kompleksnom procesu koji na naručiteljskoj strani uključuje otvaranje natječaja, pristup obrascima natječajne dokumentacije, zaprimanje predanih natječajnih dokumentacija, te nakon toga proces evaluacije kojeg provode evaluatori. Po završetku natječajnog ciklusa arhivska dokumentacija više nije javno dostupna. Na korisničkoj strani proces uključuje predaju prikupljene dokumentacije za natječaj u fizičkom obliku. Postojeća naslovna stranica natječaja prikazana je slikom 2.1 dok se u prilogu A može naći primjer originalnog obrasca natječaja.



Slika 2.1. Izgled postojeće stranice natječaja

Da bi se navedeni procesi unaprijedili i modernizirali, sukladno suvremenim tokovima, planira se digitalizacija kojom bi se do sada manualno izvođeni procesi prilikom prikupljanja, ispunjavanja i predaje natječajne dokumentacije transformirali u digitalne. Pri tome se plan digitalizacije bazira na korištenju baze podataka koja je u tu svrhu formirana [8].

Pozadina ovog pristupa se zasniva na postojanju izražene tendencije za uvođenjem modernih programskih rješenja koja se razvijaju u obliku web aplikacija [9]. Razlog tome ogleda se u dostupnosti web aplikacija izravno kroz web preglednik, pri čemu su jedini ograničavajući čimbenici brzina internet veze, sklopovske sposobnosti uređaja i mogućnosti web preglednika [10].

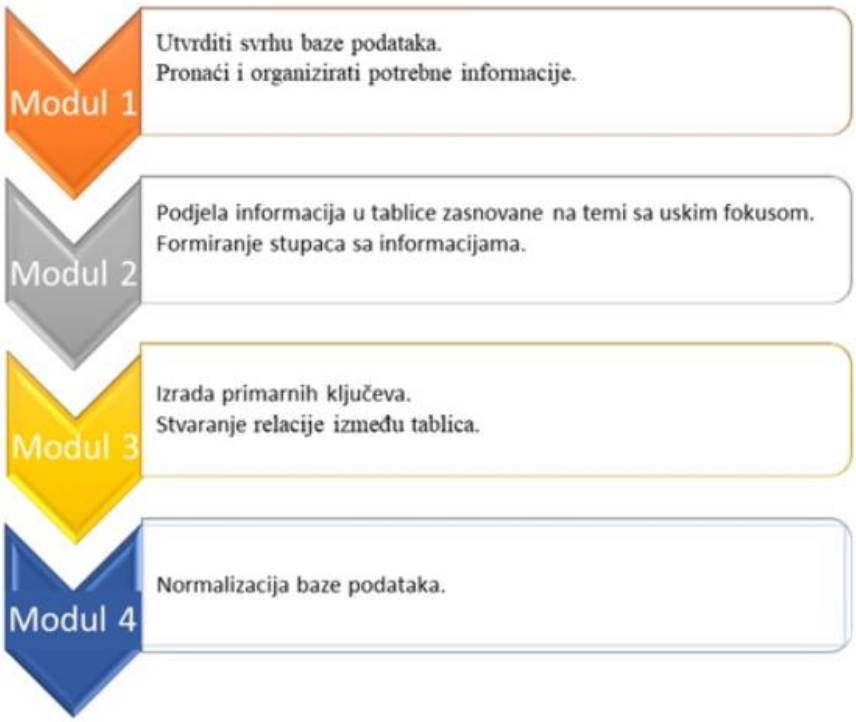
Za potrebe završnog rada je izrađena web aplikacija za administraciju projektnih prijedloga, koja omogućava registriranim korisnicima prijave na otvorene natječaje, prijave novih projektnih prijedloga putem definiranih obrazaca, te uređivanje i objavljivanje rezultata aktivnih projekata. Na naručiteljskoj strani omogućava administraciju aktivnih i pregled arhiviranih projektnih natječaja te projekata.

Na poslužiteljskoj strani sustava integriran je proizvoljni sustav za upravljanje bazama podataka, dok je na strani klijenta fokus bio više usmjeren na dizajn responzivnog korisničkog sučelja. Kako bi se to postiglo prvo se analizira domena i izdvajaju podaci koji se koriste u izradi web aplikacije. U sljedećem se koraku, započinje dizajn modela podataka koje će sadržavati aplikacija, a koji sadrži podatke svih poslovnih subjekata uključenih u poslovni proces [5, 11-13]. No, osim poslovnih subjekata (u našem konkretnom slučaju su to administrator i evaluatori na poslužiteljskoj strani te prijavitelji na korisničkoj strani), koji su predstavljali uporišne elemente, postoji i model podataka koji su predefinirani kao i međusobne veze koje se ostvaruju među pojedinim elementima. U tu se svrhu formira model baze podataka [5].

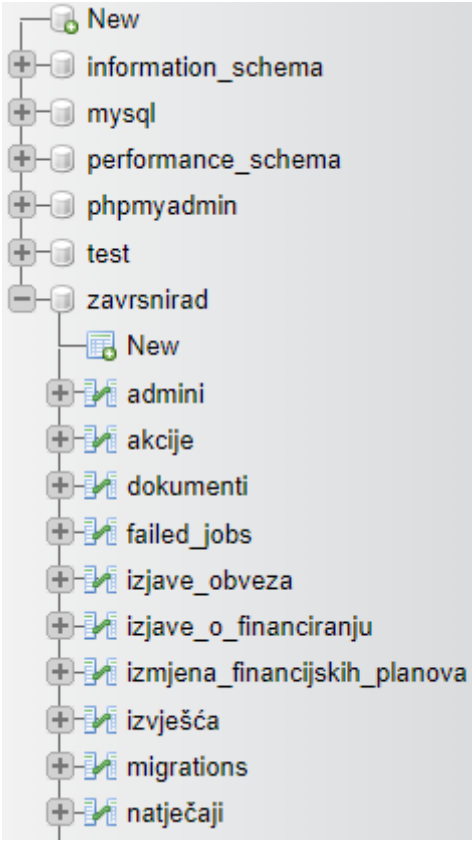
2.1. Opći postupak dizajniranja baza podataka

Dizajniranje baze podataka je vremenski zahtjevan proces koji iziskuje esencijalno razumijevanje zahtjeva kojima se mora udovoljiti kako bi se odmah napravila u potpunosti funkcionalna baza podataka koja se po potrebi može prilagoditi promjenama ili dodatnim zahtjevima [14]. Pravilno dizajnirana baza podataka omogućuje pristup ažurnim i točnim informacijama.

Nekoliko je koraka koje možemo slijediti prilikom dizajniranja baze podataka kako je prikazano na slici 2.2 dok je na slici 2.3 primjer pregleda stabla baze podataka aplikacije nakon tih koraka.



Slika 2.2. Shematski prikaz koraka dizajna baze podataka



Slika 2.3. Dio stabla baze podataka web aplikacije

Prilikom izrade završnog rada korištena je relacijska baza podataka u kojoj su podaci vezani relacijama i strukturirani tako da se osigura:

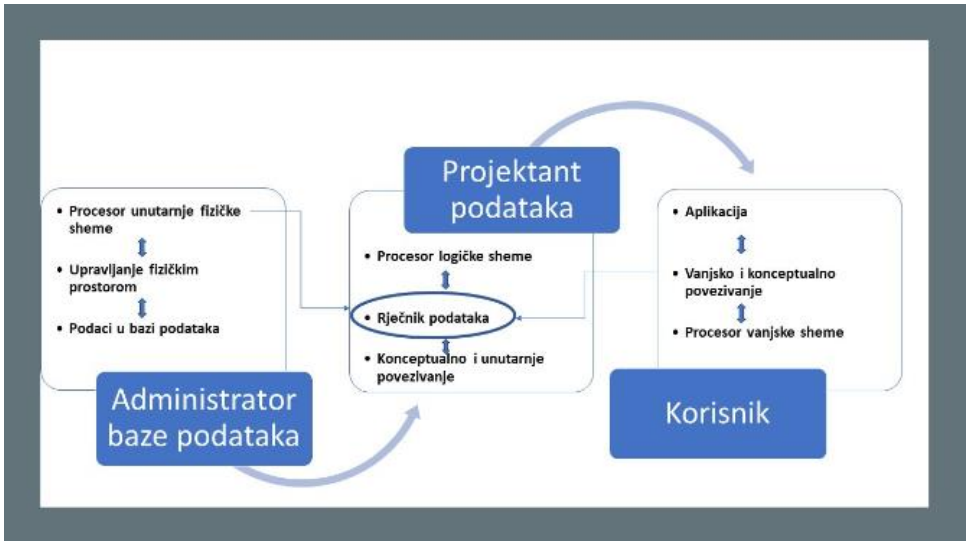
- ažurnost pohranjenih podataka,
- izgradnja sigurnosti i nadzora pristupa podatcima,
- mogućnost dnevnog sigurnosnog arhiviranja,
- mogućnost kontrole i administriranja s jednog mjesta,
- mogućnost postavljanja više upita s više kriterija u svrhu izrade analiza i sinteza podataka,
- najmanja zalihost (redundancija),
- postojanost pohranjenih podataka,
- točnost pohranjenih podataka,
- trajno očuvanje integriteta pohranjenih podataka [5],

Pritom se kod višekorisničke upotrebe odnosno mrežnog korištenja relacijske podatkovne baze postiže najveći učinak te se u najvećoj mjeri mogu iskoristiti njezine mogućnosti [5].

Nakon dizajniranja baze potrebno je istu optimizirati što se provodi postupkom normalizacije. Normalizaciju možemo ukratko objasniti kao preoblikovanje postojeće strukture objekata određeno u domenskoj povezanosti njegovih atributa odnosno zavisnosti među njima. Navedeni se postupak provodi kako bi baza postigla minimalnu moguću redundanciju i time veću učinkovitost [5].

Zavisnosti među atributima se mogu pojednostavljeno objasniti primjerom atributa grada i poštanskog broja. Poznato je da svaki grad ima jedinstveni poštanski broj i ta relacija je zavisna jer će se u bazi za određeni poštanski broj u retku uvijek pojaviti isti grad.

Naposljetku, potrebno je poznavati i pravilnosti upravljanja bazom podataka kako je sažeto ilustrirano na slici 2.4.



Slika 2.4. Shematizirani prikaz upravljanja podacima

Funkcije upravljanja podacima počinju od projektanta podataka koji definira logičku shemu. Administrator baze podataka, konzultirajući logičku shemu definira unutarnju shemu. Nju interpretira procesor unutarnje fizičke sheme i prenosi ju u rječnik podataka. Procesor logičke sheme putem rječnika podataka vrši konceptualno i unutarnje povezivanje da bi zatim projektant aplikacije, konzultirajući logičku shemu, definirao vanjske sheme. Procesor vanjske sheme vrši vanjsko i konceptualno povezivanje.

Jezik za definiranje baze podataka stoga se može koristiti za opis logičke sheme, te za definiranje vanjske i unutarnje sheme. Funkcije za manipuliranje podacima pokreće programer preko aplikacijskog programa ili korisnik koji putem upita postavlja zahtjev prema bazi podataka.

Zahtjev je vezan uz vanjsku shemu koja se linearno povezuje s logičkom shemom, unutarnjom shemom putem rječnika podataka. Zahtjev se konačno prosljeđuje na sustav za upravljanje fizičkim prostorom koji pristupa podacima na vanjskoj memoriji računala [15].

3. IZRADA WEB SUSTAVA ZA ADMINISTRACIJU PROJEKTNIH PRIJEDLOGA I PREZENTACIJU REZULTATA PROJEKATA

Prilikom izrade web sustava za administraciju projektnih prijedloga i prezentaciju rezultata projekata bilo je potrebno odabrati tehnologije koje će se koristiti što je predočeno u nastavku teksta.

3.1. Tehnološki stog za izradu programskog rješenja

Različite tehnologije su korištene u ovisnosti da li se radilo na klijentskoj strani ili poslužiteljskoj strani.

Strana klijenta predstavlja prednji dio web stranice odgovoran za njezin cjelokupni izgled i dojam. Uključuje elemente kao što su izgled, shema boja, tipografija i grafika. Pri programiranju se koriste web jezici poput HTML (HyperText Markup Language), CSS (Cascading Style Sheets) i JavaScript za izradu korisničkog sučelja i korisničkog doživljaja web stranice.

U svrhu izrade klijentske strane prilikom izrade završnog rada kombinirano se koristio React sa Viteom. Pojednostavljeno rečeno, Vite predstavlja alat koji ima za cilj pružiti brže i jednostavnije razvojno iskustvo za moderne web projekte.

Sastoji se od dva glavna dijela: razvojnog poslužitelja koji pruža bogata poboljšanja značajki u odnosu na izvorne ES module (ECMAScript – European Computer Manufacturer´s Association), na primjer, iznimno brz Hot Module Replacement (HMR) i naredbu za izgradnju koja se povezuje kao primjerice s Rollupom, unaprijed konfiguriranim za visoko optimiziran izlaz [16].

Vite je također neovisan i dolazi s korisnim zadanim postavkama odmah po otvaranju, ali je također proširiv putem svog API (Application Programming Interface) priključka i JavaScript API-ja. Vite se može koristiti za razvoj brzih aplikacija s Reactom ili drugim popularnim okvirima za izradu klijentske strane. Značaj dobro dizajniranog sučelja se ogleda u tome da ono može poboljšati korisničko iskustvo, olakšavajući korisnicima navigaciju i pronalaženje onoga što traže. Također, mogu se poboljšati performanse web stranice smanjenjem vremena učitavanja i povećanjem brzine stranice.

Za CSS se koristio Tailwind CSS. To je visoko promjenjiv radni okvir niske razine koji omogućuje jednostavno oblikovanje elemenata korištenjem jednonamjenskih klasa koje dolaze sa okvirom bez ikakvih nametnutih specifikacija oblikovanja. Tailwind sadrži sve mogućnosti

koje nam trebaju za projektiranje responzivnog sučelja. Radi skeniranjem HTML i JavaScript dokumenata i iz njih generiranjem statičnog CSS dokumenta. Prednosti ovog radnog okvira su da je oblikovanje elemenata brzo i jednostavno. Nije potrebno imenovati sve elemente i generirati njihove klase. Time se izbjegava promjena svih elemenata modifikacijom globalnih klasa. To ne znači da se one ne mogu koristiti te je još uvijek, ako je to potrebno, moguće definirati vlastite klase [17].

Poslužiteljska strana web aplikacije uključuje poslužitelja, bazu podataka i logiku aplikacije. U tu se svrhu koriste programski jezici kao što su PHP (Hypertext Preprocessor), Python i Ruby za izradu poslužiteljske web stranice, te je pri tom fokus na stvaranju logike i funkcionalnosti web stranice. Za izradu rada je korišten PHP. Kodiranje na strani poslužitelja koji komunicira s bazom podataka i obrađuje korisničke zahtjeve i stvaranje API-ja omogućuju različitim dijelovima web stranice da komuniciraju međusobno i s drugim aplikacijama.

Izrada poslužiteljskog dijela aplikacije se bazirala na korištenju Laravela. Laravel je popularan PHP okvir za izradu web aplikacija. Slijedi MVC (Model-View-Controller) arhitekturalni obrazac koji je poznat je po elegantnoj sintaksi. Laravel posjeduje brojne ugrađene značajke koje olakšavaju web razvoj, kao što je sustav usmjeravanja, spremnik usluge za ubacivanje ovisnosti, mehanizam provjere autentičnosti i ORM (Object-relational mapping) pod nazivom Eloquent za rukovanje vezama baze podataka i upitima [18]. Laravel se može koristiti za razvoj dijela web aplikacije na poslužiteljskoj strani, obavljanje zadataka kao što su upravljanje podacima, obrada zahtjeva i pružanje krajnjih točaka API-ja za korištenje na korisničkoj strani.

Naposljetku je za bazu podataka odabran MariaDB iz razloga što je MariaDB popularan sustav za upravljanje relacijskim bazama podataka otvorenog koda te je osim toga uključen u XAMPP (X-operating system, Apache MariaDB, PHA, Perl) paket. On je grana (eng. fork) jednako popularne MySQL baze podataka i generalno je kompatibilan s istom. Navodi se kao dobar izbor za mnoge web aplikacije jer je brz, pouzdan i jednostavan za korištenje. Laravel izravno podržava MariaDB, tako da se moglo jednostavno konfigurirati aplikaciju uz korištenje MariaDB kao baze podataka. Simulacija poslužitelja je provedena pomoću XAMPP-a koji je besplatni, višeplatformski paket otvorenog koda rješenja web poslužitelja koji uključuje Apache HTTP poslužitelj, već spomenutu bazu podatka i tumače za skripte napisane u programskim jezicima PHP i Perl. To je popularan

alat za simulaciju okruženja poslužitelja na lokalnom računalu, koji je omogućio razvoj i testiranje web aplikacije.

XAMPP je odabran u ovom slučaju jer je kompatibilan za postavljanje lokalnog razvojnog okruženja za Laravel aplikaciju. Nakon instaliranja i pokretanja XAMPP-a odrađeno je konfiguriranje Laravel aplikacije za korištenje pripadajuće baze podataka.

Za vizualizaciju ER (Entity Relationship) dijagrama baze korišten je draw.io. Radi se o besplatnoj aplikaciji otvorenog koda za Windows koja omogućava stvaranje dijagrama izvan mreže ili na mreži. Prednost aplikacije je da se dijagrami mogu sačuvati ili uvesti u druge aplikacije ili programe, ovisno o njihovoj vrsti. Također postoji mogućnost timskog rada kako bi se postigli učinkovitiji radni tokovi. Stoga se može primijetiti da se radi o sigurnom prvom izboru pri izradi dijagrama [19]. Navedena aplikacija odgovara profesionalnom alatu za izradu dijagrama.

3.2 Modeliranje baze podataka

Početni korak kojim je započeo postupak stvaranja baze podataka je uključivao upoznavanje sa obrascima koji su sastavni dio natječajne dokumentacije.

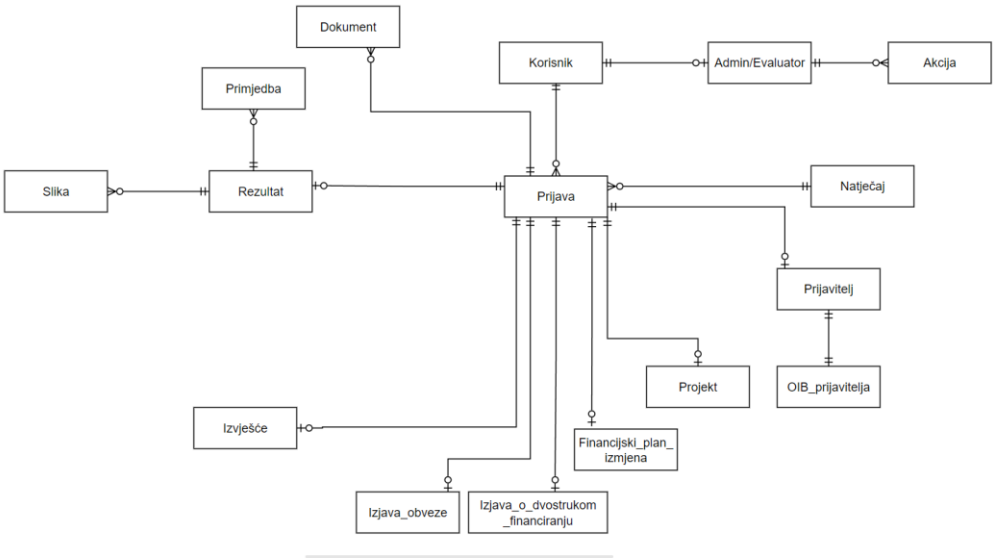
Iz korištenih navedenih obrazaca su se izveli atributi koji su predstavljali unose u bazu podataka. Kako baza podataka u pravilu mora biti atomarna, što u osnovi podrazumijeva relaciju jedan atribut – jedna vrijednost bilo je potrebno raščlaniti pojedine vrijednosti iz obrazaca [5].

Nadalje, prilikom definiranja elementarnih postavki dizajna baze nije dovoljno odrediti vrijednosti i atribute iz obrazaca, već se mora sagledati i općenito koje vrijednosti su potrebne za tražene funkcionalnosti stranica i korisnički doživljaj [5]. Stoga korisnici imaju standardiziran naziv, email i zaporku, dok također postoje evaluatori sa dodijeljenim ovlastima (svaka ovlast je zasebni atribut). Nadalje, isto tako u bazi se nalazi tablica dokumenata koja definira atribute imena kao i put dohvata dokumenta.

U sljedećem koraku je bilo potrebno odrediti zavisnosti među atributima. Većina atributa iz obrazaca je bila nezavisna. Neki od atributa iz tablice sa podacima o prijavitelju su međusobno zavisni. Primjerice, OIB i RNO (koji također predstavlja jedinstvenu brojčanu vrijednost), te godina i cilj osnivanja. Isti OIB uvijek se povezuje sa istim RNO-om, godinom osnivanja i ciljem, pa su stoga nakon postupka normalizacije, podaci grupirani u drugoj tablici u kojoj OIB postaje strani ključ kako je ilustrirano na slici 3.1. Potrebno je navesti da neki atributi mogu biti samo naizgled zavisni [5]. Primjerice ime, broj mobitela, adresa elektroničke pošte, sjedište te bankovni

račun tvrtke prijavitelja jer se oni mogu tijekom vremena promijeniti. Tvrtka se može preseliti, promijeniti banku poslovanja ili preimenovati tako da su navedeni podaci promjenjivi. Isto vrijedi i za voditelja. Potrebno je voditi računa i o slučajevima male vjerojatnosti jer se može dogoditi da postoje voditelji s jednakim imenom i prezimenom, no tada očekivano neće biti isti broj mobilnog telefona, elektroničke adrese i/ili faksa. Pored toga za mobilni telefon, elektroničku adresu i faks, ne možemo niti reći da su zavisni međusobno jer to može biti npr. elektronička adresa te tvrtke za tu poziciju, a može se raditi i o elektroničkoj adresi te osobe. Elektronička adresa, faks i telefon, ako se dodjeljuju sukladno poziciji u tvrtci, se također mogu promijeniti ako se tvrtka preimenuje ili preseli.

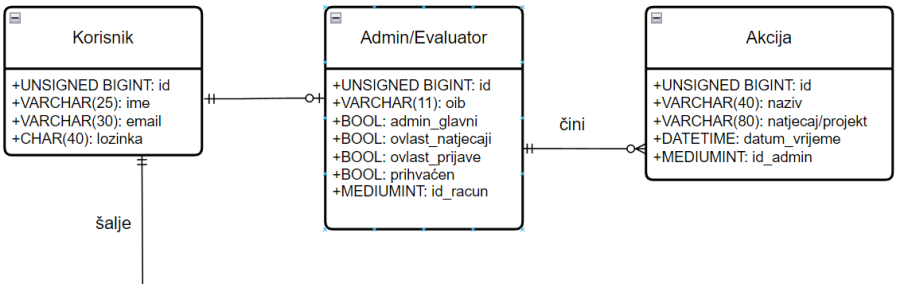
Nakon toga je slijedio korak normalizacije. Kako se u ovom slučaju na korisničkoj strani nalaze poduzetnici koji se prijavljuju na natječaj, procjena je da broj korisnika neće biti vrlo velik. Stoga je posljedično normaliziranje do 3. normalne forme bilo zadovoljavajuće. U konačnici, baza podataka završnog rada oblikovana je kako je prikazano na slici 3.1.



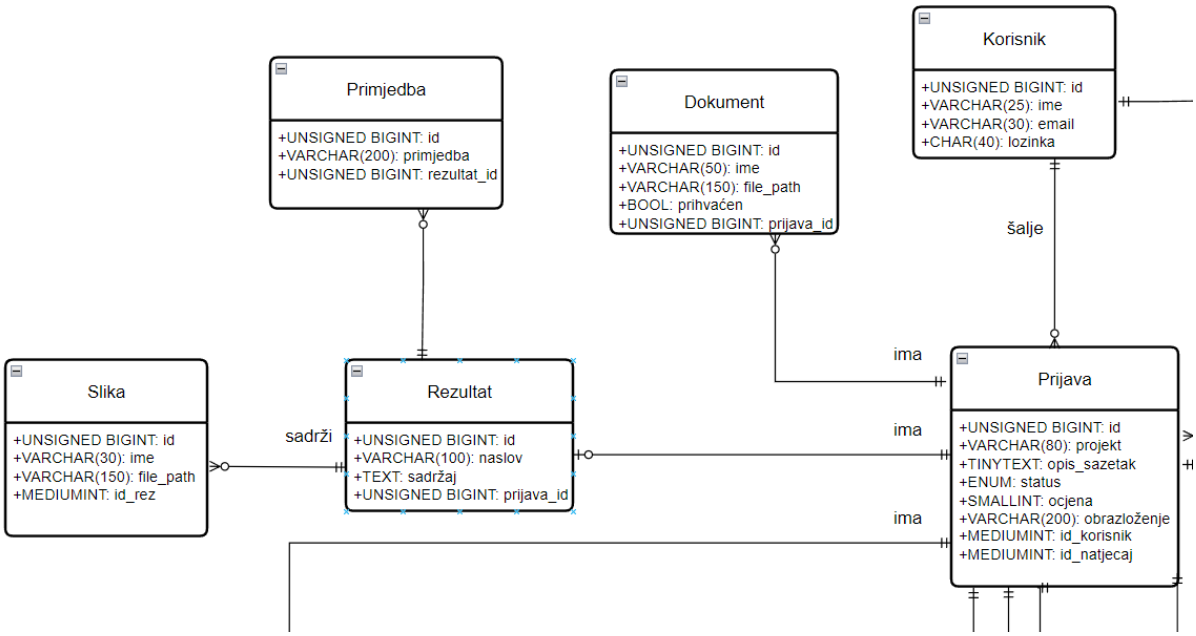
Slika 3.1 Prikaz jednostavnog ER dijagrama baze podataka

U nastavku je prikaz raščlanjene cjelokupne baze podataka kako bi se mogli uočiti detalji modeliranja i dizajna. Slika 3.2 prikazuje detaljnije prvi dio baze podataka u kojem je prikazan odnos korisnika i administratora/evaluatora te njihovih akcija. Moguće je vidjeti kako korisnik može biti administrator ili evaluator te oni mogu činiti određene akcije koje će se spremati u bazi podataka. Slikom 3.3 prikazan je odnos prijave koju korisnik podnosi na natječaj i dokumentacije koju za prijavu predaje na portal te rezultata koji će kasnije biti izrađeni. Vidljivo je da rezultat još

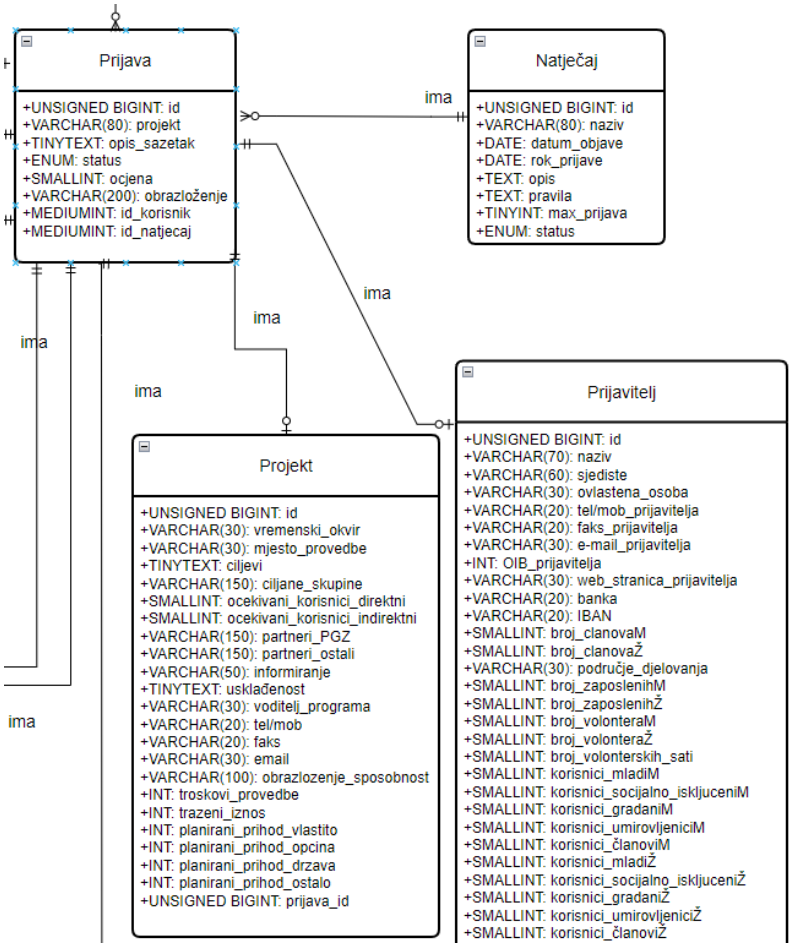
može sadržavati slike ili mogu postojati primjedbe na njega. Konačno je slikom 3.4 prikazan odnos između natječaja i prijave i obrazaca prijave. Ovdje je vezom prikazano kako prijava može ili ne mora imati odgovarajuće obrasce. Razlog tome je da je sustav dizajniran da će se obrazac izraditi tek kada ga korisnik krene popunjavati. Time se izbjegava scenarij gdje korisnik napravi prijavu, ali iz nekog razloga ne popuni obrasce, te oni potpuno prazni stoje u bazi podataka.



Slika 3.2. Odnos korisnika i administratora/evaluatora



Slika 3.3. Prikaz veza između prijava i priloženih dokumenata te konačnog rezultata



Slika 3.4. Veze između natječaja i prijave i njezinih obrazac

3.3. Implementacijski detalji od posebnog značaja

Većina funkcija razvijene web aplikacije Natječaji i prijave su standardne značajke sličnih aplikacija za administraciju i podnošenje prijava čija implementacija ne zahtijeva poseban prikaz. Ovaj rad svoju složenost primarno dobiva iz samog broja tih značajki. Međutim, još uvijek postoji jedna značajka koja je, ako ne jedinstvena, onda poprilično rijetka za ovakav tip aplikacije, a to je izrada rezultata projekta na strani korisnika. Specifično će se prikazati slučaj implementacije uređivača teksta.

Kao uređivač teksta na strani korisnika korišten je Quill. To je besplatni WYSIWYG (What you see is what you get) uređivač teksta otvorenog koda te već dolazi sa nizom opcija za konfiguraciju. Kako bi se koristio potrebno ga je prvo uvesti u pogled i konfigurirati na odgovarajući način.

Nakon konfiguracije je dovoljno pozvati funkciju u pogledu da bi uređivač radio. Drugi element ove funkcionalnosti je prikazivanje teksta iz uređivača u odvojenoj stranici. Implementacija

navedene funkcionalnosti je bila otežana zbog dva konkretna problema. Prvi je taj što Tailwind nema već definiran CSS stil za HTML oznake lista, tako da one nisu bile interpretirane na ispravan način. Drugi je da Tailwind ima svoje oznake za određene CSS klase i nije prepoznavao imena koja je Quill koristio. Ti problemi su riješeni tako da je instaliran službeni „typography“priključak [20, 21]. Time se riješio problem nepravilnog prikazivanja lista. Pored toga su se za imena ostalih klasa definirale vlastite klase, kako bi ih Tailwind prepoznao.

3.4. Korisničke uloge u sustavu

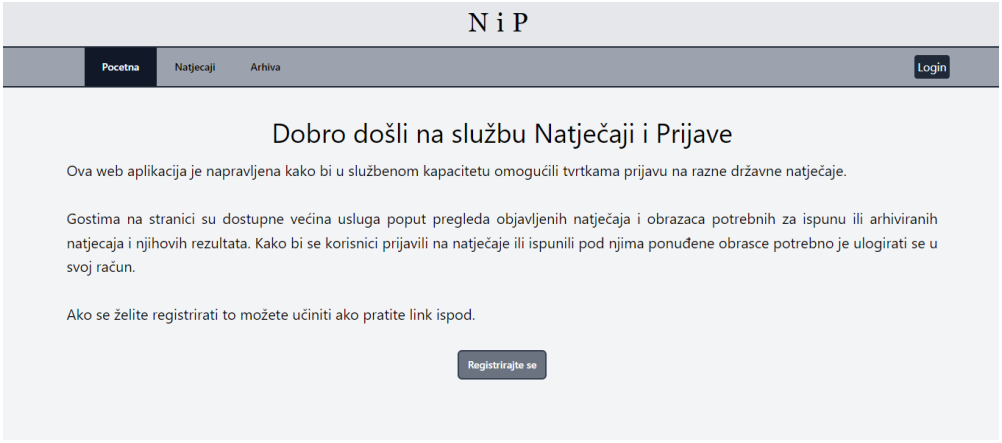
Prilikom dizajniranja sustava formirane su četiri uloge u koje se korisnici web aplikacije mogu svrstati. Prva uloga je korisnik na stranici koji nije prijavljen. Njega se smatra gostom na stranici te su mu dostupne osnovne mogućnosti gledanja objavljenih natječaja i njihovih osnovnih informacija poput opisa i pravila, kao i pregled stranica rezultata prihvaćenih i izvršenih projektnih prijedloga dostupan u arhivi natječaja. Druga uloga je standardni registrirani i prijavljeni korisnik koji, pored mogućnosti gosta, ima još i mogućnost prijave na natječaje te ispunjavanje prateće dokumentacije i funkcija vezanih uz taj proces. Detaljan hodogram prijave projekta na natječaj je prikazan u kasnijem poglavlju. Treća uloga je evaluator. Njegova je zadaća administracija prijavljenih projekata i natječaja ovisno o njemu dodijeljenim ovlastima. To obuhvaća stvaranje natječaja i promjenu njihovog statusa te pregled prijava projektnih prijedloga i promjenu njihovog statusa. Posljedna uloga je sam administrator. On ima puni pristup i ovlasti svim alatima administracije unutar aplikacije. Uloga mu je administracija samih evaluatora, ali još uvijek ima mogućnosti administracije prijava i natječaja kao i evaluator, ako je to potrebno. Za svoju primarnu ulogu dostupni su mu alati kontrole prihvaćanja prijava za evaluatore i kontrole njihovih ovlasti. Pored toga ima mogućnost uvida u njihove aktivnosti unutar aplikacije.

4. OPIS SLUČAJEVA KORIŠTENJA

U ovom poglavlju opisuju se karakteristični slučajevi korištenja koji su implementirani u oglednoj aplikaciji.

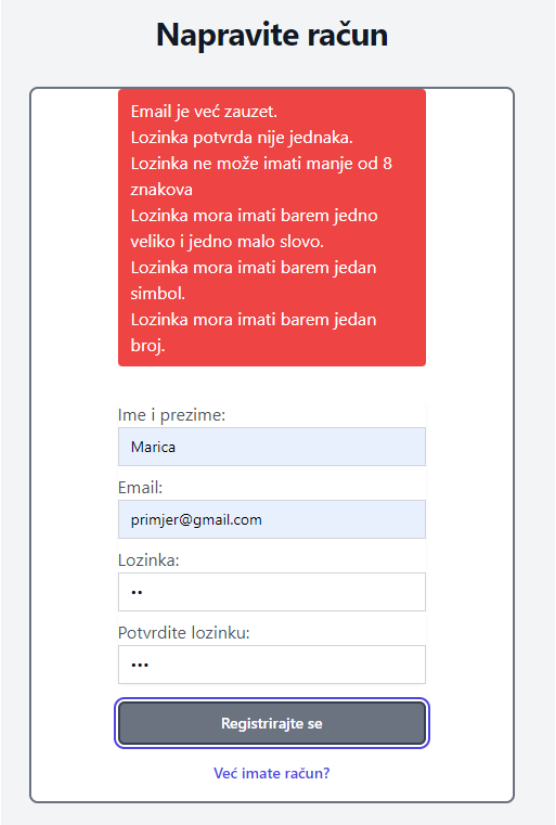
4.1. Registracija i prijava u sustav

Kroz narednih nekoliko slika se prikazuje proces registracije korisnika preko početnog prozora koji je vidljiv na slici 4.1 i obrasca registracije vidljivog na slici 4.2. Kada gost pristupi početnoj stranici aplikacije ima izbor posjetiti stranicu natječaja ili arhive. Pored toga, prikazane su mu poveznice na stranice za prijavu i registraciju u sustav.



Slika 4.1. Početna stranica s uputama za registraciju

Registracija korisnika se odvija kroz uobičajeno ispunjavanje osnovnih informacija te se slanjem zahtjeva odvija validacija koja provjerava da unesena elektronička adresa nije već zauzeta. Osim toga, provjerava se da li zaporka odgovara zadanim pravilima, to jest da nije kraća od 8 znakova, da sadrži i mala i velika slova, brojeve i posebne znakove i da je identična polju potvrde.



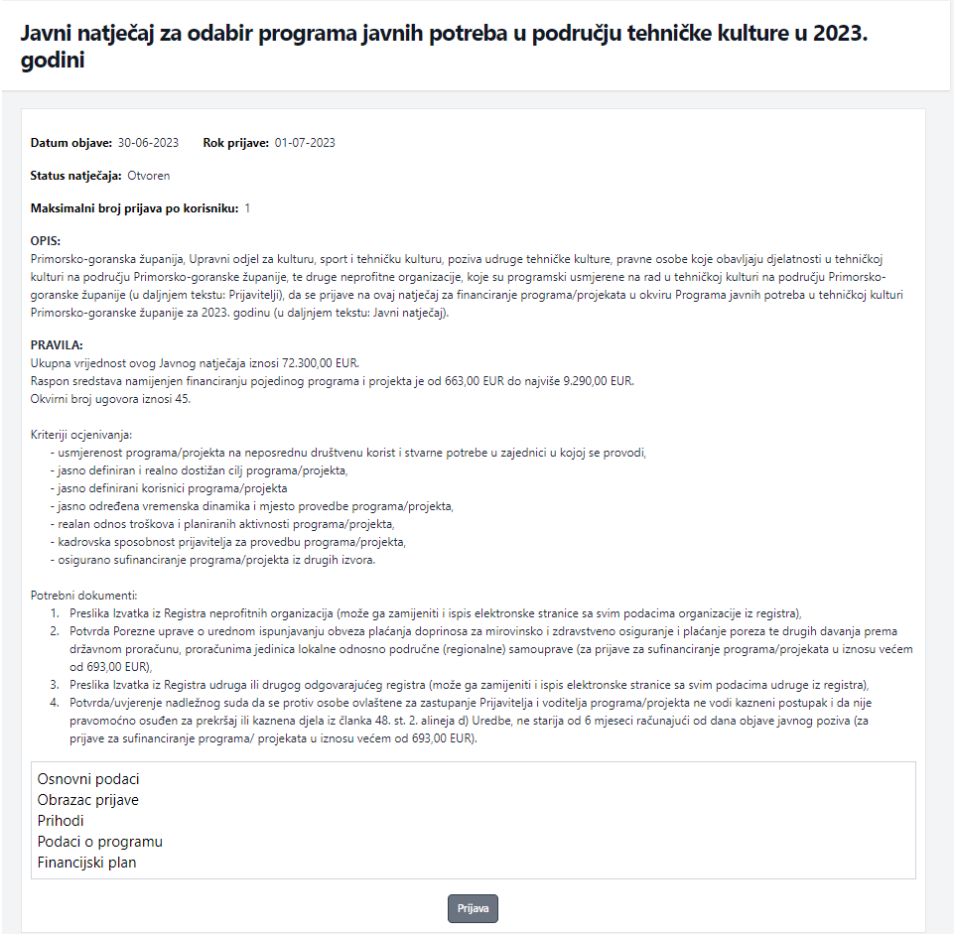
Slika 4.2. Registracija korisnika s primjerom validacijskih poruka

Registracija evaluatora je nešto složenija i kako bi joj se pristupilo mora se upisati URL (Uniform Resource Locator) adresa. To je sigurnosna mjera kako bi se smanjila vjerojatnost pokušaja registracije za to nepredviđene osobe, stoga poveznica na nju nije dostupna u aplikaciji. Sam obrazac je sličan korisničkoj registraciji sa razlikom da je potrebno još upisati i OIB koji, kao i adresa elektroničke pošte, mora biti jedinstven. Nakon što osoba pošalje svoju prijavu za evaluatora, o čijem je uspješnom slanju obavještena porukom kao i kada dođe do greške u unosu podataka, ne može se prijaviti dok njezinu registraciju ne prihvati administrator. Nakon uspješne registracije i prijave, korisniku se šalje elektroničkom poštom potvrda o uspješnosti vidljiva na slici 4.3 te ga se usmjerava na listu natječaja gdje može započeti proces prijave svojeg projektnog prijedloga.

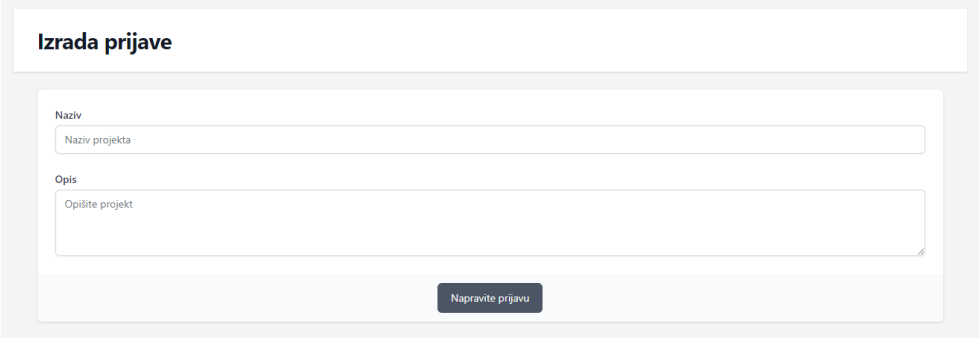
Slika 4.3. Potvrda registracije

4.2. Hodogram projektnog prijedloga

Proces prijave projekta započinje sa samim natječajem prikazanim na slici 4.4 gdje korisnik može vidjeti detalje poput statusa natječaja, njegovog opisa, pravila te dozvoljeni broj aktivnih prijava od strane jednog korisnika. Pri kraju stranice su izlistane poveznice na obrasce koje korisnik mora ispuniti. Ako se stisne na njih, unaprijed je moguće vidjeti tražene informacije. Klikom na gumb prijave započinje izrada prijave opisana slikom 4.5.



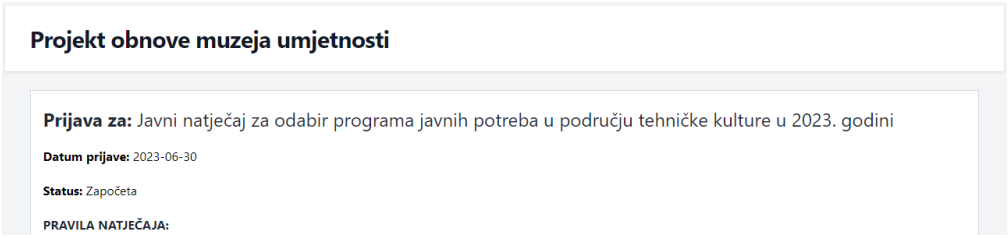
Slika 4.4. Stranica natječaja s osnovnim informacijama o natječaju



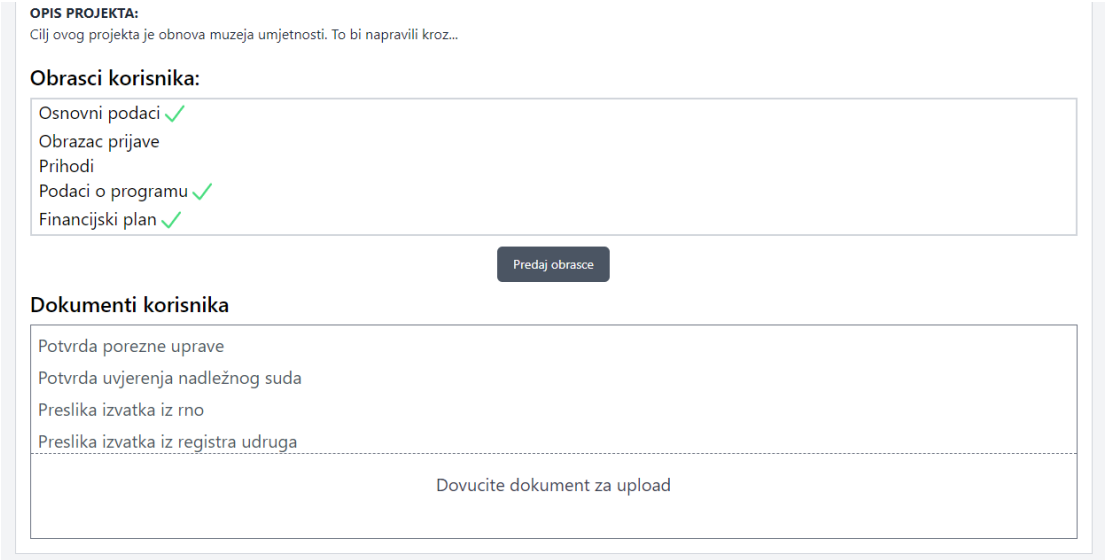
Slika 4.5. Početak izrade projektnog prijedloga

Nakon što je unesen željeni naziv projekta i njegov opis, korisnika se preusmjerava na stranicu prijave koja je prikazana slikama 4.6 i 4.7. U prozoru prijave korisnik vidi obrasce koje treba ispuniti. Unutar pravila natječaja je definirano koje dokumente je potrebno učitati u web aplikaciju i to je korisniku omogućeno tako da dokumente koje želi predati dovuče u za to označen prostor. Moguće je spremiti više dokumenata odjednom, a format dokumenata koji se prima je ograničen

na pdf. Ukoliko korisnik pokuša predati dokument drugog formata, ili ako pokuša predati obrasce prije nego su u potpunosti ispunjeni, te u slučaju da nije predao sve potrebne dokumente, javlja se obavijest greške sa odgovarajućom porukom. Status dovršene ispunjenosti pojedinih obrazaca označava se zelenom kvačicom pored njihovih imena.

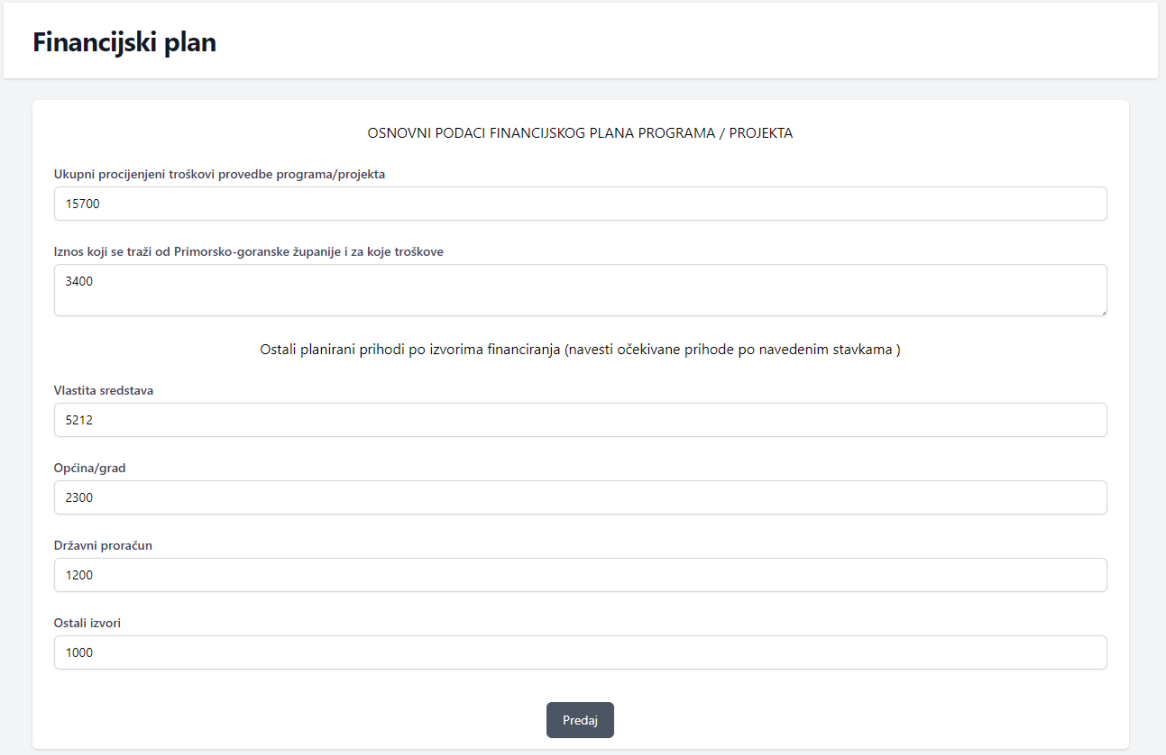


Slika 4.6. Prvi dio prozora korisnikove prijave

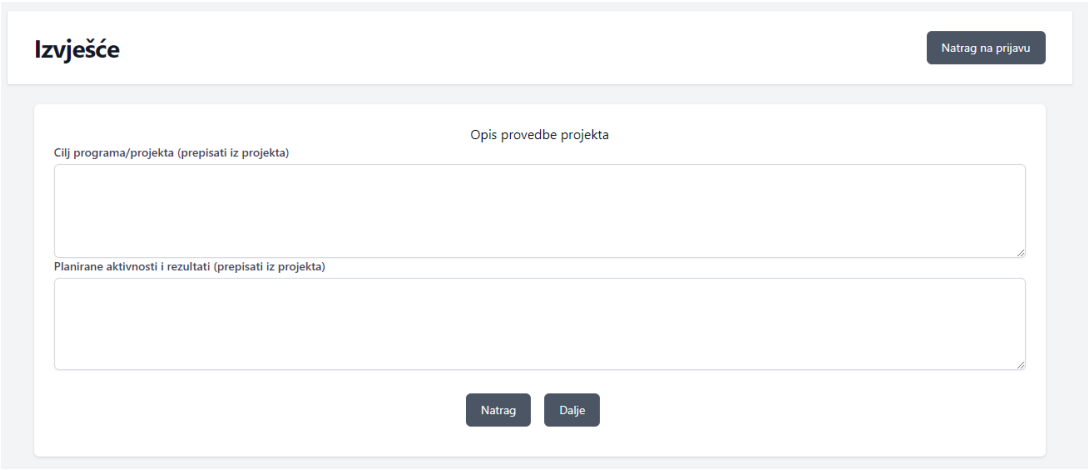


Slika 4.7. Drugi dio prozora korisnikove prijave

Obrasci koje je potrebno ispuniti su originalno bili dijelovi nekolicine većih obrazaca koji su, radi bolje preglednosti, podijeljeni. Primjer jednog obrasca prikazan je na slici 4.8. U situacijama kada je i nakon podjele obrazac preopsežan i nepregledan za ispunjavanje, podijeljen je JavaScriptom u više odjeljaka što se može vidjeti na slici 4.9. Iako su pitanja dio istog obrasca aktivacijom dostupnih gumba (dalje-nazad) mijenja se njihov prikaz. Naravno, ukoliko korisnik nije iz stranice svoje prijave predao obrasce, može mijenjati upisane podatke sve dok ih ne preda, nakon čega više nema tu mogućnost.



Slika 4.8. Primjer obrasca s manjim brojem unosnih polja

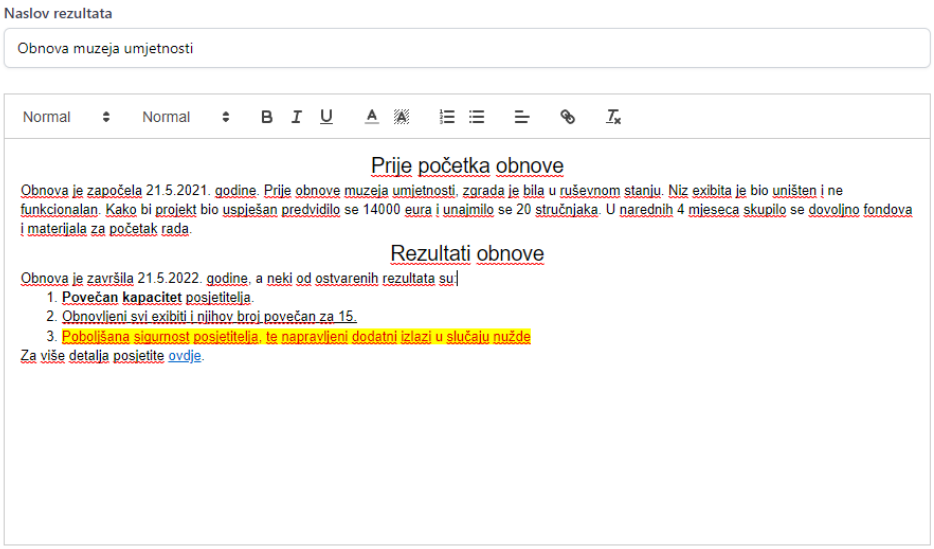


Slika 4.9. Primjer obrasca kod kojeg se prikaz pitanja kontrolira gumbima

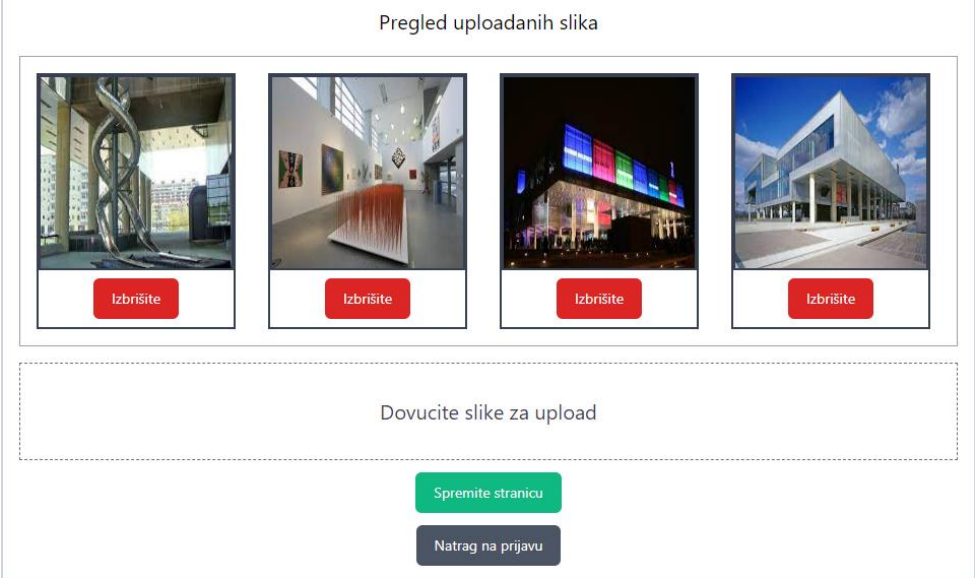
Po predaji projektne prijave, potrebno je čekati evaluaciju. Ukoliko je prijava odbijena, na stranici prijave korisniku se navodi razlog odbijanja. Ako je prijava prihvaćena, na stranici se pojavljuje informacija o odobrenom iznosu. Nadalje, postaju vidljivi novi prozori koji služe za prikaz obrazaca i dokumenata potrebnih nakon završetka projekta. Pored toga korisniku se omogućava

izrada ogledne tj. prezentacijske stranice s rezultatima projekta (prikazano slikama 4.10, 4.11 i 4.12). Ako je ispunio ostatak obrazaca i dovršio predaju dokumentacije, ostvaruje mogućnost predaje rezultata na procjenu evaluatoru.

Prilikom izrade rezultata, korisnik na lijevoj polovici prozora ima mogućnost izrade vlastite stranice preko unosa naslova same stranice i sadržaja. Sadržaj se unosi u uređivaču teksta koji ima standardne sposobnosti mijenjanja veličine naslova, veličine fonta, podebljanih, ukošenih i podcrtanih slova, brojevne i točkaste liste te mogućnosti biranja boje teksta, pozadine teksta, centriranje sadržaja i stvaranje poveznica. Nakon toga korisnik može postaviti željene slike koje se prikazuju na desnoj polovici u obliku interaktivne galerije. Po spremanju rezultata, dobiva obavijest o uspješnosti.



Slika 4.10. Izrada prezentacijske stranice projekta - uređivanje teksta



Slika 4.11. Izrada prezentacijske stranice projekta - uređivanje galerije slika

Slika 4.12. Prikaz uređene prezentacijske stranice projekta

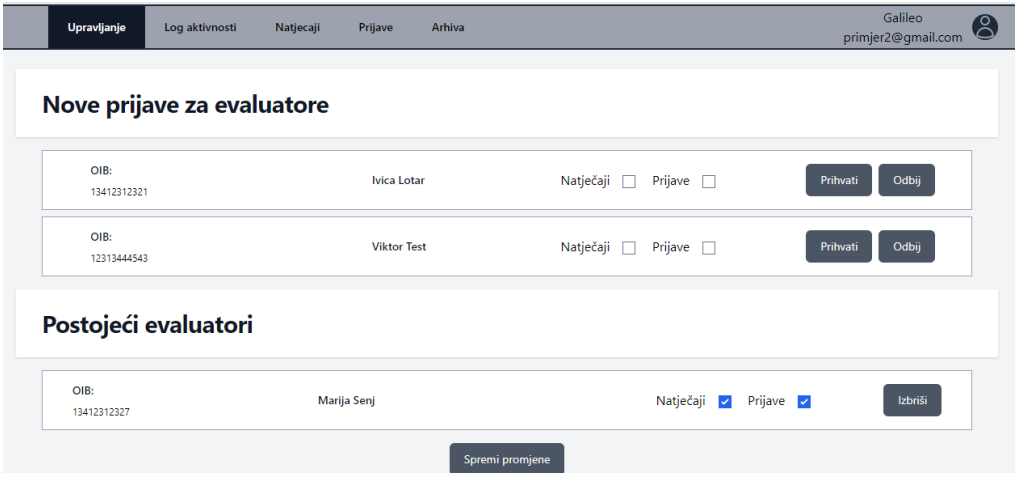
Nakon predaje rezultata, potrebno je još jedanput čekati da ih evaluator potvrdi i prihvati. Ako su potrebne izmjene, korisnik će na stranici prijave vidjeti primjedbe prema kojima treba promijeniti prikaz svojih rezultata prije ponovne predaje. Prihvaćanjem završnih rezultata u obujmu

prezentacijske stranice je završen proces izvođenja projekta te su, nakon što je natječaj arhiviran, sve stranice rezultata projekata javno dostupne pod arhiviranim natječajima.

4.3. Administracija evaluatora

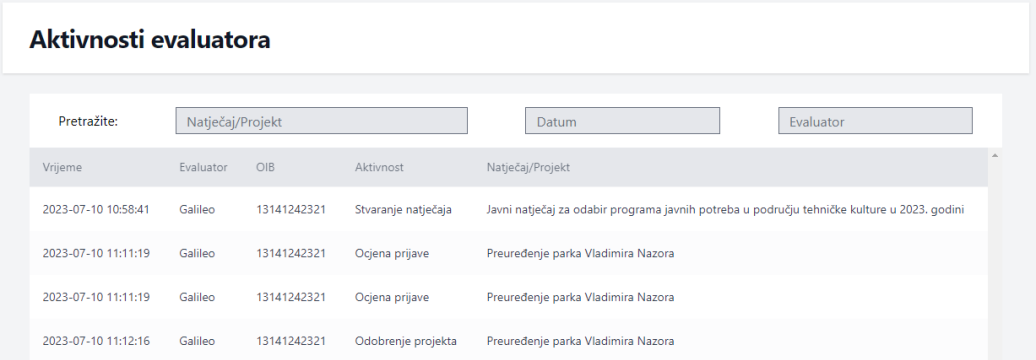
Kada osoba pošalje prijavu za evaluatora, administrator je može vidjeti prikazanu u listi novih prijava pod opcijom *Upravljanje* što je prikazano na slici 4.13. Administrator prema OIB-u i imenu evaluatora može provjeriti o kojoj je osobi riječ i je li joj potrebno dati ovlasti. OIB je administratoru potreban kako bi razlikovao evaluatore istih imena i prezimena. Koncepcijski je zamišljeno da su administrator i evaluator dio iste kompanije ili su u poslovnom odnosu gdje administrator ima suglasnost za pristup osobnim podacima evaluatora.

Nakon što je potvrđen identitet evaluatora, moguće je dodijeliti mu ovlasti vezane uz aktivne natječaje ili prijave projektnih prijedloga. Prihvaćanjem prijave, kartica evaluatora se premješta u listu već postojećih evaluatora. Prijave evaluatora se odobravaju zajedno sa pripadajućim ovlastima, dok se ovlasti već prihvaćenih evaluatora mogu promijeniti zajedno označavanjem funkcije spremanja promjene.



Slika 4.13. Administracija evaluatora

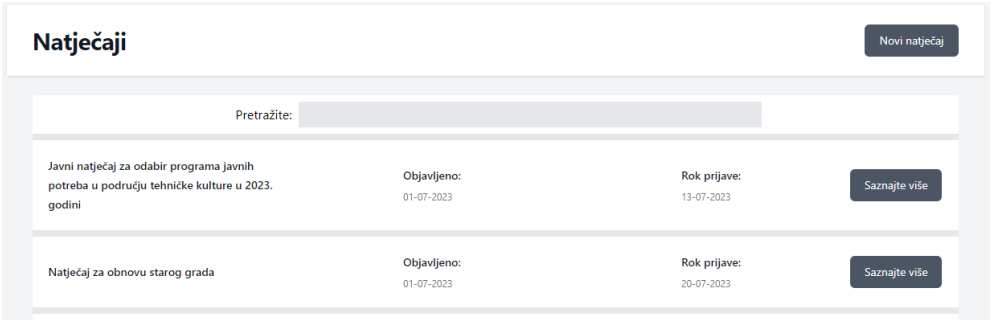
Osim upravljanja samim ovlastima, administratoru je dostupan alat za praćenje aktivnosti evaluatora u obliku dnevnika aktivnosti vidljivog na slici 4.14. U alatu je moguće pretraživati aktivnosti prema natječaju ili projektu, datumu kada je aktivnost izvršena te prema samom evaluatoru čiju se aktivnost želi provjeriti.



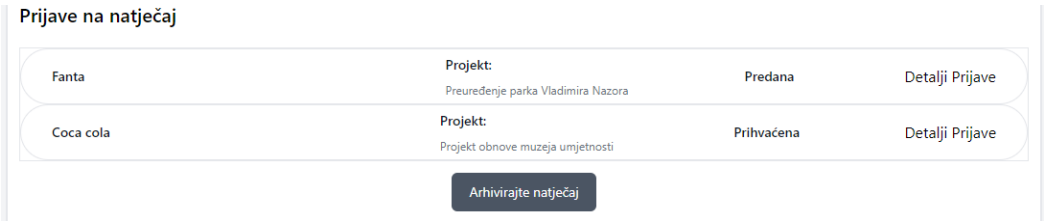
Slika 4.14. Dnevnik aktivnosti evaluatora

4.4. Administracija natječaja i njihovih prijava

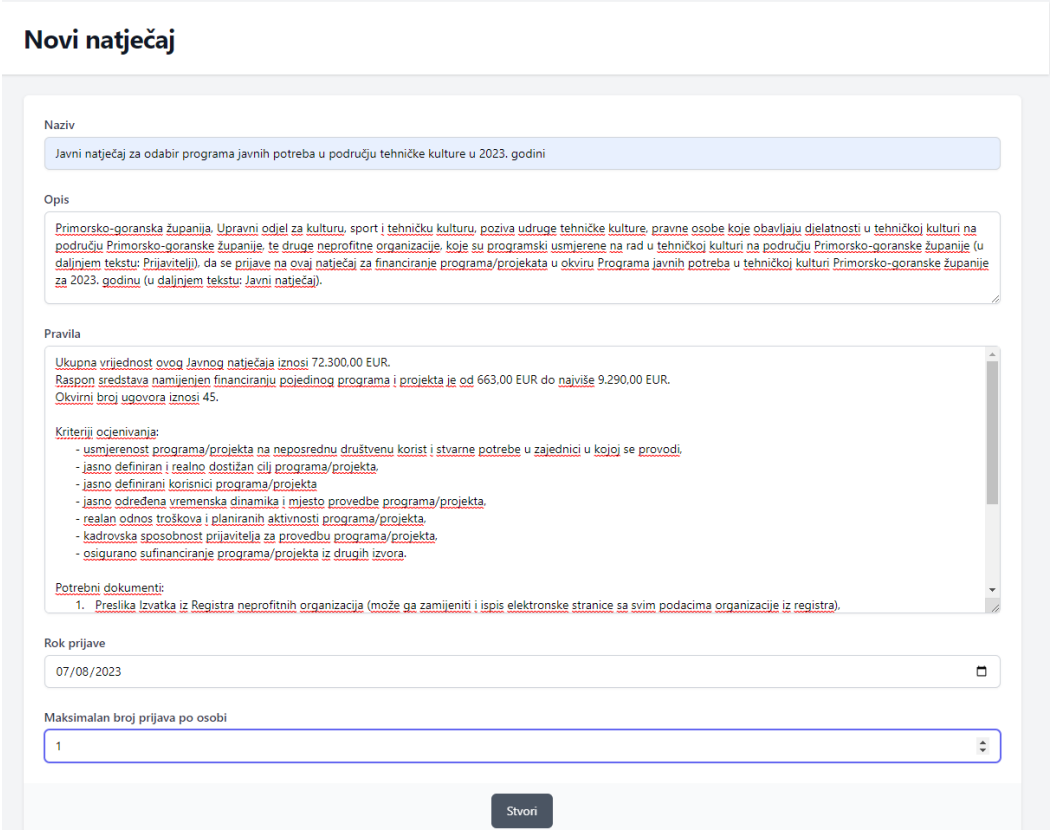
Administracija natječaja je jednostavan proces koji primarno uključuje njihovu izradu i funkcije izmjene njihovog statusa. Natječaji se mogu pretraživati u za to predviđenoj listi kako je prikazano na slici 4.15. Nakon pronalaska željenog natječaja i odlaskom na njegov prikaz, osim informacija vidljivih drugim ulogama, prikazuje se lista predanih prijava na natječaj kako je vidljivo na slici 4.16. Ispod nje se nalazi opcija za zatvaranje ili arhiviranje natječaja ovisno o trenutnom stanju istog. Glavna funkcija administracije natječaja je samo stvaranje istog prikazano na slici 4.17.



Slika 4.15. Lista aktivnih natječaja

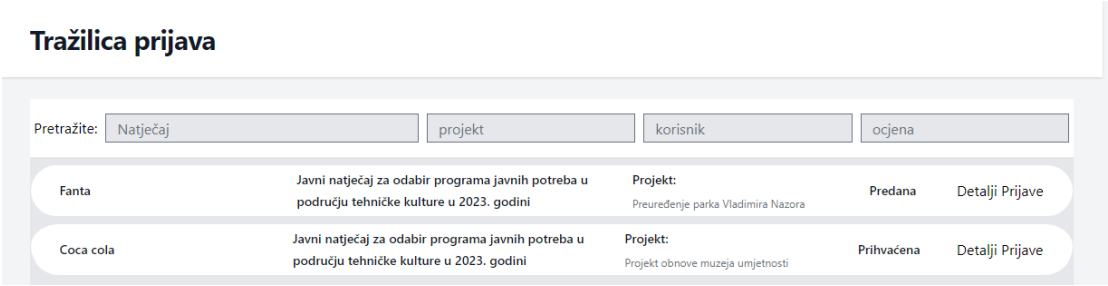


Slika 4.16. Pregled aktivnih prijava za dotični natječaj



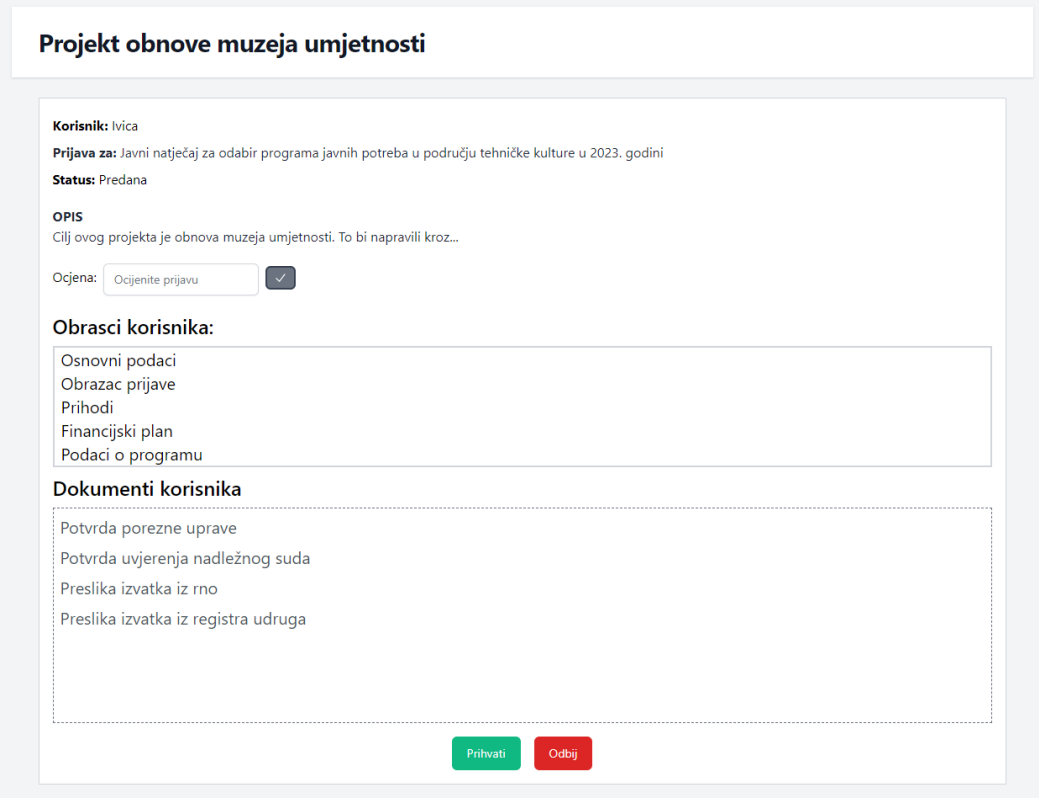
Slika 4.17. Definiranje osnovnih informacija za novi natječaj

S obzirom da je proces prijave projekata složeniji, te je sam broj prijava veći od broja natječaja, i složenost administracije je veća. Svaka predana prijava na pojedini natječaj je izlistana na pripadajućoj stranici. Ako se traži specifična prijava ili skupina prijava, dostupna je tražilica prijava prikazana slikom 4.18. Tražilicom je postignuta mogućnost pretraživanja po natječaju, korisniku prijave, projektu i ocjeni prijave.

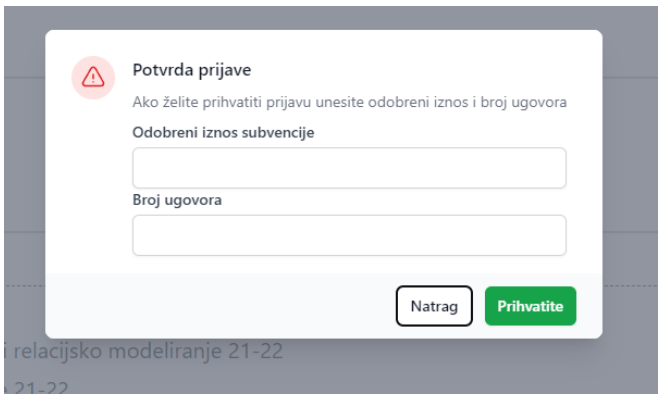


Slika 4.18. Tražilica prijava

Pogled same prijave koju vidi evaluator je prikazan slikom 4.19. Kada je status prijave "predana" evaluator može ocijeniti prijavu. Ako nedostaju neki od osnovnih uvjeta, prijavu je moguće odmah odbiti prije same evaluacije. Prilikom odbijanja je potrebno obrazložiti razlog odbijanja. Prihvaćanje prijave je moguće tek nakon njezinog ocjenjivanja, u protivnom se dobije obavijest upozorenja. Prihvaćanje prijave podrazumijeva i upis dodijeljenog iznosa financiranja projektu i broj ugovora projekta, a što je prikazano slikom 4.20.



Slika 4.19. Evaluatorski pogled na natječajnu prijavu

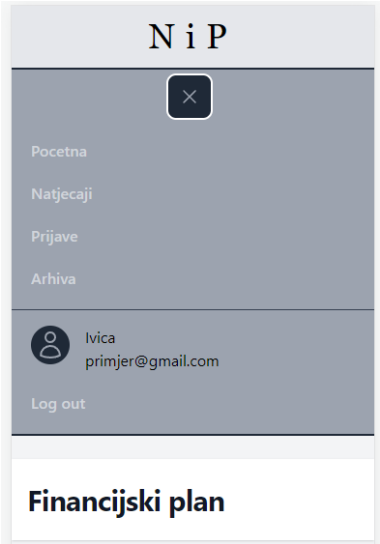


Slika 4.20. Skočni prozor u procesu prihvaćanja prijave

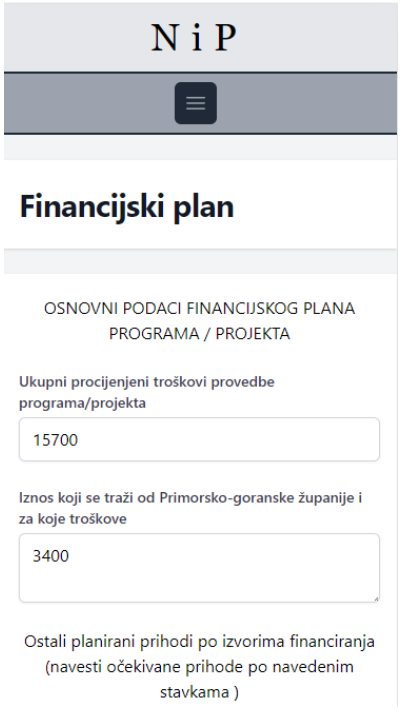
Do drugog koraka u administraciji prijave dolazimo kada je korisnik dovršio projekt i predao svoje rezultate. Tada se, kao i kod korisnika, dodatno prikažu sekcije obrazaca i dokumenata nakon prihvaćanja prijave. Pored toga evaluator ima ulogu pogledati predanu prezentacijsku stranicu rezultata za što se pojavljuje potrebna poveznica. Ukoliko se prezentacijska stranica s rezultatima odbija, unosi se primjedba koju bi korisnik trebao uvažiti prije nego opet pošalje stranicu na uvid. Prihvaćanjem stranice je gotov proces administracije prijave i cijelog projekta.

4.5. Korištenje na zaslonima različitih veličina

Jedan od naputaka završnog rada je je uključivao da web aplikacija i korisničko sučelje odgovaraju tzv. responzivnom dizajnu kako bi korisnici mogli pristupiti i koristiti aplikaciju i preko različitih tipova uređaja (mobilnih telefona, pločnih računala i standardnih stolnih računala). Primjer takvog dizajna je prikazan slikama 4.21 i 4.22.



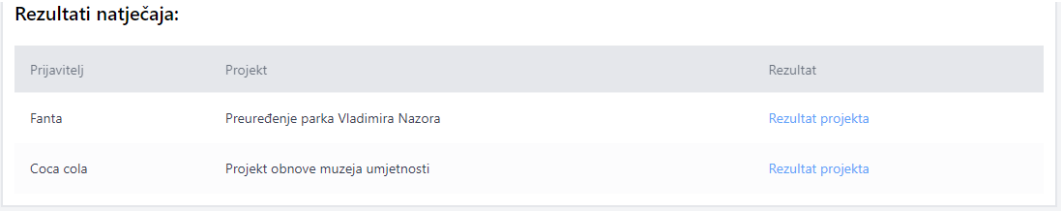
Slika 4.21. Prilagodljivi izbornik na zaslonu pametnog telefona



Slika 4.22. Primjer obrasca renderiranog na zaslonu mobilnog telefona

4.6. Pregled rezultata natječaja

Ako je natječaj arhiviran, na dnu njegove stranice se, ispod već viđenih informacija, prikazuje lista rezultata tog natječaja kako je prikazano na slici 4.23. Pristup arhiviranim natječajima nije ograničen i dostupan je svim ulogama unutar web aplikacije.



Slika 4.23. Tablica rezultata

Ako se prati poveznica rezultata, korisnik može vidjeti prezentacijsku stranicu s rezultatima dotičnog projekta poput one na slici 4.24.

Slika 4.24. Stranica rezultata

5. ZAKLJUČAK

U radu je prezentirana i opisana web aplikacija za administraciju projektnih prijedloga i prezentaciju rezultata projekata.

U aplikaciji su implementirani sljedeći slučajevi korištenja:

- prijavu novih projektnih prijedloga putem definiranih obrazaca,
- uređivanje i objavljivanje rezultata aktivnih projekata,
- administriranje aktivnih projektnih prijedloga ,
- administriranje natječaja,
- evaluaciju i pretraživanje prijava.

Razvijeno je pregledno korisničko sučelje, praćeno logičkim slijedom funkcionalnosti i mogućnostima koje ga čine responzivnim, poboljšavajući korisnikov doživljaj uz istovremeno omogućavanje korištenja aplikacije na različitim vrstama uređaja.

U usporedbi s klasičnim načinom predavanja projektnih prijava i administriranja natječaja, primjećuje se da primjena web aplikacije ostvaruje nekoliko prednosti, od čega se ističe poboljšavana preglednost i dinamika samog procesa prijava i administriranja natječaja.

Dinamika prijave koja se u cijelosti odvija unutar aplikacije nudi neusporedivo bolji korisnički doživljaj, dok mogućnost pristupa aplikaciji s različitih tipova uređaja dokida ograničenja mogućnosti predaje prijave.

Prijavi se može opetovano pristupiti i moguća su ažuriranja do predaje konačne verzije.

Aplikacija je koncipirana na način da je moguće proširivanje postavki i funkcionalnosti odnosno nema ograničenja za daljnji razvoj.

Zahvaljujući strukturiranoj bazi podataka u kojoj se čuvaju sve važne informacije opisanog procesa, moguće su usporedne analize različitih natječaja, što je osobito značajno za naručiteljsku stranu odnosno administratore i evaluatore. Naime, moguće je dobiti uvid u učinak kako pojedinog evaluatora tako i usporedne rezultate različitih natječaja.

Dio projektne dokumentacije, iako arhivirane, može se dohvatiti s ciljem uvida u najznačajnije rezultate projekata financiranih na nekom natječaju, što nije moguće kod postojećeg oblika administracije natječaja.

Konačno, mogućnost prezentacije rezultata projekata nakon završenog natječajnog ciklusa, koja ostaje dostupna korisnicima u pregledniku arhive, značajan je iskorak prema diseminaciji rezultata projektnih natječaja, transparentnosti natječaja i podizanju kvalitete cjelokupnog procesa administriranja projektnih prijedloga.

LITERATURA

1. Siegenfeld, A.F.; Bar-Yam, Y. An Introduction to Complex Systems Science and its Applications, arXiv:1912.05088v2, 2020.
2. Ladyman, J.; Lambert, J., Weisner, K.B. What is a Complex System? Eur. J. Philos. Sci. 2013, 3, 33–67
3. Gupta, M. 2. 2. B. M. S. (2020, March 24). *What is Digitization, Digitalization, and Digital Transformation?* ARC Advisory Group. https://www.arcweb.com/blog/what-digitization-digitalization-digital-transformation#:~:text=Digitization%20refers%20to%20creating%20a%20digital pristupljeno 21.06.2023
4. Srivastav, Manoj & Nath, Asoke. (2016). WEB CONTENT MANAGEMENT SYSTEM. International Journal of Innovative Research in Advanced Engineering (IJIRAE). 3. 51-56. 10.6084/M9.FIGSHARE.3504368.V1.
5. Varga M. Baze podataka, Konceptualno, logičko i fizičko modeliranje podataka, MANUALIA UNIVERSITATIS STUDIORUM ZAGRABIENSIS, Klasa: 032- 01/15-01/63, Urbroj: 380-061/252-16-6, od 9. veljače 2016.
6. Rumbaugh, J., Blaha, M., Premerlani, i sur. (1991): Object-Oriented Modeling and Design. Prentice-Hall.
7. Sommerville, I. (2015): Software Engineering, 10th ed. Pearson.
8. PGZ. (n.d.). https://www.pgz.hr/projekt_program/javni-natjecaj-za-odabir-programa-javnih-potreba-u-podrucju-tehnicke-kulture-u-2023-godini/ pristupljeno 15.05.2023.
9. Royce, Winston (1970), "Managing the Development of Large Software Systems"
10. Business Systems Planning (1981): Information Systems Planning Guide. IBM Corp.
11. https://zih.hr/konzalting/strategija-i-poslovni-procesi/upravljanje-poslovnim-procesima/ pristupljeno 22.06.2023.
12. Date, C. J. (1990): An Introduction to Data Base Systems, vol. 1.Addison-Wesley.
13. Martin, J. (1977): Computer Data-base Organization, 2nd ed. Prentice-Hall.
14. Kramberger T., Kraljević S. i Kovačević R. Baze podataka Tehničko veleučilište u Zagrebu ISBN: 2018 978-953-7048-70-9.
15. ANSI/SPARC (1978): American National Standard Institute/ Standards Planning and Requirements Committee, Study Group on Data Management Systems. Framework Report on Database Management Systems. Information Systems, vol. 3, no. 3.
16. Vite. (n.d.). Next Generation Frontend Tooling. https://vitejs.dev/ pristupljeno 17.06.2023.
17. Tailwind CSS - Rapidly build modern websites without ever leaving your HTML. (n.d.). Tailwind CSS. https://tailwindcss.com, pristupljeno 02.07.2023

18. Laravel - The PHP Framework For Web Artisans. (n.d.). https://laravel.com/docs/10.x pristupljeno 02.07.2023
19. *draw.io*. (n.d.). https://www.drawio.com/ pristupljeno 23.06.2023.
20. *Your powerful rich text editor*. (n.d.). Quill. https://quilljs.com/ pristupljeno 23.06.2023.
21. *@tailwindcss/typography - Tailwind CSS*. (n.d.). Tailwind CSS. https://tailwindcss.com/docs/typography-plugin pristupljeno 23.06.2023.

POPIS OZNAKA I KRATICA

API - Application Programming Interfaces

CMS - Content management system

CSS - Cascading Style Sheets

ER - Entity Relationship

ES module- ECMAScript – European Computer Manufacturer´s Association

HMR - Hot Module Replacement

HTML- HyperText Markup Language

MVC - Model-View-Controller

OIB – Osobni identifikacijski broj

ORM - Object relational mapping

PHP - Hypertext Preprocessor

PWA - progresivnih web aplikacija

RNO – Registar neprofitnih organizacija

SQL - Structured Query Language

URL - Uniform Resource Locator

WYSIWYG - What you see is what you get

SAŽETAK

U ovome radu razvijen je i predstavljen web sustav za administraciju projektnih prijedloga i prezentaciju rezultata projekata. Modelirana je i implementirana odgovarajuća relacijska baza podataka te poslužiteljski i klijentski dio sustava s četiri korisničke uloge. Odgovarajuća web aplikacija registriranim korisnicima, u ovisnosti o njihovoj ulozi, omogućava prijavu novih projektnih prijedloga putem definiranih obrazaca, uređivanje i objavljivanje rezultata aktivnih projekata, administraciju aktivnih natječaja i projektnih prijedloga, vrednovanje projektnih prijava, te pregled prezentacijskih stranica arhiviranih projekata. Na klijentskoj strani sustava posebna pažnja usmjerena je na dizajn responzivnog korisničkog sučelja.

Ključne riječi — web aplikacija, administracija natječaja, projektna prijava, CMS, Laravel

ABSTRACT

In this thesis, a web system for managing project proposals and presenting project results was developed and presented. A corresponding relational database was modeled and implemented, as well as the server and client part of the system with four user roles. The associated web application allows registered users, depending on their role, to submit new project proposals through defined forms, edit and publish the results of active projects, manage active tenders and project proposals, evaluate project proposals, and view presentation pages of archived projects. On the client side of the system, special attention is paid to designing a responsive user interface.

Keywords — web application, tender management, project application, CMS, Laravel

DODACI

Dodatak A Primjer obrasca za ručnu prijavu

b. OSNOVNI PODACI O PROGRAMU / PROJEKTU			
Naziv programa/projekta			
Vremenski okvir provedbe			
Mjesto provedbe			
Ciljevi provedbe			
Ciljane skupine			
Očekivani br. korisnika	Direktni		
	Indirektni		
Opisni sažetak programa/projekta			
Partneri na projektu	Iz PGŽ-a		
	Ostali		
Način informiranja javnosti o provedbi i rezultatima			
Usklađenost programa/projekta sa Strategijom razvoja PGŽ (2016.-2020/2021.) i Strategijom razvoja tehničke kulture PGŽ (2017.-2020/2021.)			
Podaci o voditelju programa/projekta	Ime i prezime:		
	Tel/mob:	E-mail adresa:	
	Faks:		
	Obrazloženje stručne osposobljenosti za provedbu programa/projekta:		

c. OSNOVNI PODACI FINANCIJSKOG PLANA PROGRAMA / PROJEKTA	
Ukupni procijenjeni troškovi provedbe programa/projekta	
Iznos koji se traži od Primorsko-goranske županije i za koje troškove	
Ostali planirani prihodi po izvorima financiranja (navesti očekivane prihode po navedenim stavkama)	• _____ (vlastita sredstava) • _____ (općina/grad) • _____ (državni proračun) • _____ (ostali izvori)