

SUSTAV ZA ONLINE UPRAVLJANJE REZERVACIJAMA DVORANA NA VISOKIM UČILIŠTIMA: PREZENTACIJSKI I POSLOVNI SLOJ

Lazo, Mia

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Dubrovnik / Sveučilište u Dubrovniku**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:155:902461>

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

Download date / Datum preuzimanja: **2025-01-24**



SVEUČILIŠTE U DUBROVNIKU
UNIVERSITY OF DUBROVNIK

Repository / Repozitorij:

[Repository of the University of Dubrovnik](#)



zir.nsk.hr



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U DUBROVNIKU
ODJEL ZA ELEKTROTEHNIKU I RAČUNARSTVO

MIA LAZO

**SUSTAV ZA ONLINE UPRAVLJANJE REZERVACIJAMA
DVORANA NA VISOKIM UČILIŠTIMA: PREZENTACIJSKI I
POSLOVNI SLOJ**

DIPLOMSKI RAD

Dubrovnik, rujan 2024.

SVEUČILIŠTE U DUBROVNIKU
ODJEL ZA ELEKTROTEHNIKU I RAČUNARSTVO

**SUSTAV ZA ONLINE UPRAVLJANJE REZERVACIJAMA
DVORANA NA VISOKIM UČILIŠTIMA: PREZENTACIJSKI I
POSLOVNI SLOJ**

DIPLOMSKI RAD

Studij: Primijenjeno/poslovno računarstvo

Kolegij: Ekspertni sustavi

Mentor: doc.dr.sc. Ivona Zakarija

Komentor: mag. ing. comp. Toni Besjedica

Student: Mia Lazo

Dubrovnik, rujan 2024.

SAŽETAK

Diplomski rad istražuje razvoj i implementaciju naprednog sustava za online upravljanje rezervacijama dvorana na visokim učilištima, s posebnim naglaskom na prezentacijski i poslovni sloj. Digitalna transformacija obrazovnog sektora značajno je promijenila upravljanje resursima, zamjenjujući tradicionalne metode ručnog vođenja i osobnih kontakata suvremenim tehnologijama koje pružaju veću učinkovitost, transparentnost i pouzdanost. Sustav za online rezervacije optimizira korištenje prostora, olakšava koordinaciju i planiranje te nudi jednostavan pristup informacijama o dostupnosti dvorana i terminima predavanja.

Detaljna analiza korisničkih zahtjeva i funkcionalnosti prikazuje cjelokupan proces razvoja sustava, od planiranja i specifikacije projektnih zahtjeva, preko front-end razvoja i korištenja alata poput Tableau, do dizajna baze podataka. Prikazani su korisnički dijagrami aktivnosti i slučajevi korištenja, koji ilustriraju interakcije između korisnika i sustava. Implementacija obuhvaća dizajn korisničkog sučelja i kodiranje, naglašavajući važnost intuitivnog i pristupačnog administratorskog sučelja za učinkovito upravljanje sadržajem.

Rad pruža sveobuhvatan uvid u modernizaciju obrazovnih institucija kroz primjenu naprednih tehnologija, demonstrirajući kako digitalna rješenja mogu unaprijediti organizacijske procese i povećati efikasnost upravljanja resursima.

Ključne riječi: Online upravljanje rezervacijama, Razvoj korisničkog sučelja, Tableau, Visoka učilišta, Prezentacijski i poslovni sloj, Efikasnost, Planiranje i koordinacija, Korisnički dijagrami aktivnosti, Slučajevi korištenja, Intuitivno korisničko sučelje, Modernizacija obrazovnih institucija.

ABSTRACT

This thesis explores the development and implementation of an advanced system for online management of hall reservations at higher education institutions, with a particular emphasis on the presentation and business layers. The digital transformation of the educational sector has significantly changed resource management, replacing traditional methods of manual tracking and personal contacts with modern technologies that provide greater efficiency, transparency, and reliability. The online reservation system optimizes space utilization, facilitates coordination and planning, and offers easy access to information on hall availability and lecture schedules.

A detailed analysis of user requirements and functionalities showcases the entire process of system development, from planning and specification of project requirements, through front-end development and the use of tools like Tableau, to database design. User activity diagrams and use case diagrams are presented, illustrating the interactions between users and the system. The implementation includes the design of the user interface and coding, emphasizing the importance of an intuitive and accessible administrative interface for efficient content management.

The thesis provides a comprehensive insight into the modernization of educational institutions through the application of advanced technologies, demonstrating how digital solutions can improve organizational processes and increase the efficiency of resource management.

Keywords: Online reservation management, Front-end development, Tableau, Higher education institutions, Presentation and business layers, Efficiency, Planning and coordination, User activity diagrams, Use cases, Intuitive user interface, Modernization of educational institutions.

SADRŽAJ

1. UVOD.....	1
2. PLANIRANJE IZRADE WEB APLIKACIJE I SPECIFIKACIJA PROJEKTNIH ZAHTJEVA.....	3
2.1 Korisnici web aplikacije.....	3
2.2 Zahtjevi za izradu web stranice	4
2.2.1 Funkcionalni zahtjevi.....	4
2.2.2 Nefunkcionalni zahtjevi	6
2.3 Model funkcija (Dijagram dekompozicije funkcija)	8
2.4 Model procesa	10
2.4.1 Dijagram konteksta (0.razina)	10
2.4.2 Pregledni dijagram glavnih procesa (1.razina)	11
2.4.3 Detaljni dijagram za odabrani proces (2.razina)	12
3. KORIŠTENE TEHNOLOGIJE	13
3.1 HTML.....	13
3.2 CSS	16
3.2.1. Vanilla CSS	19
3.3 JavaScript.....	20
3.3.1 Vanilla JavaScript.....	22
3.4 WampServer.....	23
3.5 Tableau	24
4. BAZA PODATAKA	27
4.1 Baza podataka web aplikacije.....	27
4.2 Korisnički dijagram aktivnosti	28
4.3 Dijagram slučajeva korištenja	30
5. IMPLEMENTACIJA WEB SJEDIŠTA I SUSTAVA ZA UPRAVLJANJE SADRŽAJEM	31

5.1 Naslovna stranica	31
5.2 Izvešće zauzetosti dvorana	37
5.3 Login forma	39
5.4 Sustav za upravljanje sadržajem – Puni pristup	42
5.5 Sustav za upravljanje sadržajem – Srednji pristup	45
6. ZAKLJUČAK.....	50
7. LITERATURA	51
8. PRILOZI	52
8.1 Popis tablica	52
8.2 Popis slika.....	52

1. UVOD

Digitalizacija obrazovnog sektora u proteklih nekoliko godina donijela je potpuno novu dimenziju upravljanja resursima unutar visokoškolskih institucija. Tradicionalni pristupi vođenja rasporeda i rezervacija dvorana, koji su se oslanjali na ručno vođenje i osobne kontakte, sve više ustupaju mjesto modernim tehnologijama koje omogućuju efikasnije i transparentnije upravljanje. Uvođenje sustava za online upravljanje rezervacijama dvorana predstavlja ključan korak za unapređenje organizacije i optimizacije korištenja prostora unutar visokih učilišta.

U današnje vrijeme, gotovo je nezamislivo pronaći obrazovnu instituciju bez digitalnog sustava za upravljanje resursima. Takav sustav postaje neophodan alat za osiguranje kontinuiranog i nesmetanog provođenja obrazovnog procesa, dostupan u bilo koje vrijeme i s bilo kojeg mjesta uz uvjet internetske povezanosti. Omogućava jednostavan pristup informacijama o dostupnosti dvorana, terminima predavanja, te kontaktima ključnih osoba, što znatno olakšava koordinaciju i planiranje. Fleksibilnost i učinkovitost sustava za online upravljanje rezervacijama privlače korisnike zbog njihove sposobnosti da brzo i efikasno obave potrebne zadatke.

Kvalitetno izrađen sustav za online upravljanje rezervacijama dvorana, koji uključuje sve relevantne podatke o dvoranama, profesorima, kolegijima i rasporedima, ulijeva povjerenje korisnicima i pokazuje ozbiljnost institucije. U vrijeme kada većina ljudi koristi internetske pretraživače za pronalaženje informacija, obrazovne ustanove prepoznaju važnost angažiranja stručnjaka za izradu i održavanje ovakvih sustava. Osim glavnog korisničkog sučelja, čest je zahtjev i za implementacijom administratorskog sučelja koje omogućuje jednostavno upravljanje sadržajem bez potrebe za stalnim angažiranjem IT stručnjaka. Sučelje treba biti intuitivno i pristupačno, omogućujući administrativnom osoblju lako provođenje izmjena bez tehničkih znanja.

Ovaj diplomski rad sastoji se od nekoliko poglavlja koja detaljno opisuju tehnologije korištene za izradu sustava za online upravljanje rezervacijama dvorana na visokim učilištima, kao i proces planiranja i implementacije projekta. Uvodno poglavlje postavlja kontekst i objašnjava važnost digitalizacije u obrazovnom sektoru. Fokus drugog poglavlja je na planiranju izrade web aplikacije i specifikaciji projektnih zahtjeva, uključujući model funkcija i model procesa. Treće poglavlje pruža sažet pregled korištenih tehnologija, s

posebnim naglaskom na razvoj front-end dijela aplikacije i integraciju funkcionalnosti unutar alata Tableau. U četvrtom poglavlju opisana je arhitektura baze podataka. Prikazana je shema baze podataka koja je centralna za funkcionalnost sustava za online upravljanje rezervacijama dvorana. Također, uključeni su korisnički dijagram aktivnosti i dijagram slučajeva korištenja, koji su ključni za razumijevanje interakcija korisnika s aplikacijom. Peto poglavlje prikazuje stvarnu implementaciju web aplikacije i sustava za upravljanje sadržajem, uključujući dizajn korisničkog sučelja i kodove za izgled. Kroz ova poglavlja, rad pruža sveobuhvatan uvid u razvoj i primjenu sustava za online upravljanje rezervacijama dvorana na visokim učilištima.

2. PLANIRANJE IZRADE WEB APLIKACIJE I SPECIFIKACIJA PROJEKTNIH ZAHTJEVA

Rad je strukturiran u par poglavlja, a kroz sljedeće poglavlje analizirat će se korisnički zahtjevi kao temelj za daljnji proces razvoja web sjedišta. Zahtjevi su jasan putokaz za izradu, osiguravajući precizan sadržaj i smanjujući mogućnost pogreške. Posebna pozornost obratit će se na definiranje zahtjeva vezanih za sustav upravljanja sadržaja kao osnovom za rad web aplikacije, pružajući informacije za efikasnije administriranje podacima, što smanjuje ulogu programera u održavanju sustava. Osim toga, koristit će se model funkcija, tj. dijagram dekompozicije funkcija kako bi se jasnije razumjela hijerarhijska struktura funkcionalnosti web aplikacije. Model procesa bit će implementiran u tri razine kako bi se pojasnile performanse aplikacije, od ključnih do detaljnijih operacijskih procesa. Sažeto, model daje sveobuhvatan pregled, sloj po sloj, što osigurava efikasnu implementaciju i upravljanje. Ovaj pristup osigurava sveobuhvatnu analizu i planiranje ključnih aspekata razvoja web aplikacije, ističući važnost preciznih specifikacijskih zahtjeva i optimalnog dizajna funkcionalnosti i procesa.

2.1 Korisnici web aplikacije

Korisnici proizvoda su zaposlenici visokog učilišta, poput tajnica odjela, profesora, osoblja recepcije te studenti kao jedini korisnici koji nemaju mogućnost upravljanja web aplikacijom.

Implementirano je tri pristupa korištenja web aplikacije:

- Svi korisnici imaju mogućnost pregleda rasporeda te filtracije po datumu.
- Tajnice imaju potpuni pristup podacima sustava, prijavljuju se u sustav za upravljanje sadržajem koristeći svoje registrirane e-mail adrese i lozinke, koji omogućuje unos, izmjene i brisanje cjelokupnog sadržaja, uključujući upravljanje dvoranama, profesorima, kolegijima, recepcijom te rasporedom.
- Profesori i osoblje recepcije imaju jednaki pristup koji im omogućuje unos vlastitih termina, upravljanje prethodno dodanim terminima, kao i provjeru dostupnosti dvorana. Također, prijavljuju se u sustav za upravljanje sadržajem koristeći svoje registrirane e-mail adrese i lozinke.

Opcija upravljanja profilom dostupna je svim korisnicima koji koriste sustav za upravljanje sadržajem, a sadrži brzi povratak na naslovnu stranicu, promjenu lozinke za prijavu te mogućnost odjave iz sustava prilikom završetka korištenja.

2.2 Zahtjevi za izradu web stranice

Zahtjevi za izradu su prvotni zadatak prilikom izrade web aplikacije. Ističu bitnost uključivanja samih korisnika sustava u izradu. Poznavanje cjelokupnog procesa vrlo je bitno prije početka izrade sustava, kako bi se koristile jasne smjernice tijekom izrade. Precizni zahtjevi vrlo su bitni kako bi se smanjila mogućnost nerazumijevanja i pogrešne izrade određenih procesa. Jasnim razumijevanjem željenih funkcija sustava korisnicima se pojednostavljuje uporaba web aplikacije i povećava se intuitivnost korištenja. Za jednostavnije korištenje vrlo je bitan jasan i uredan izgled samog sučelja za korisnike. Intuitivno dizajnirano sučelje, prilagođen izgled i boje sučelja olakšavaju i usmjeravaju korisnike do ispunjavanja ciljnog zadatka.

2.2.1 Funkcionalni zahtjevi

U prvoj tablici (Tablica 1.) prikazani su funkcionalni zahtjevi web aplikacije. Tablica se sastoji od tri stupca koja sadržavaju ID zahtjeva, naziv te opis. Korištena je skraćenica „CMS“ (engl. *Content management system*) koja označava sustav za upravljanje sadržajem web aplikacije, te „Login“ što odnosi na prijavu korisnika u sustav.

Tablica 1. Funkcionalni zahtjevi web aplikacije

ID ZAHTJEVA	NAZIV	OPIS
01	Naslovna stranica - Prikaz rasporeda	Prikaz rasporeda po godinama studija pruža jasno i estetski ugodno sučelje za jednostavan uvid u akademske rasporede
02	Naslovna stranica - Promjena boje web aplikacije ovisno o odjelu	Automatsko mijenjanje boje sučelja prema boji koja predstavlja svaki odjel
03	Naslovna stranica - Izvješće zauzetosti dvorana	Prikazuje vizualni pregled zauzetosti dvorana kroz grafički prikaz u Tableau-u
04	Naslovna stranica -	Pružanje jednostavnog alata za odabir i pregled

	Filtriranje po datumu	sadržaja s pomoću intuitivnog kalendara
05	CMS - Prikaz upravljanja profilom	Na jasnom i uočljivom mjestu implementirati tipku za upravljanje profilom sa zadanim mogućnostima
06	CMS - Funkcionalnost za kreiranje novih rezervacija putem korisničkog sučelja	Omogućiti korisnicima jednostavan i učinkovit način za unos kroz jasno dizajnirano sučelje koje uključuje formu za unos.
07	CMS - Upozorenja o nemogućnosti unosa termina	Informirati korisnike o problemima pri pokušaju rezervacije, poput zauzetosti dvorane, putem jasnih i vidljivih obavijesti
08	CMS - Implementirati funkcionalnosti za otkazivanje ili izmjenu postojećih rezervacija	Omogućava korisnicima da lako upravljaju svojim rezervacijama putem sučelja koje podržava promjene ili brisanje termina
09	CMS - Pregled rezervacija (trenutne i nadolazeće)	Jasan uvid u svoj planirani raspored na jednom mjestu
10	CMS - Prikaz obavijesti	Pravovremeno obavješćavanje o unosu/brisanju/ažuriranju termina
11	CMS - Prikaz upozorenja	Jasno prikazati greške i upozorenja prilikom unosa ili ažuriranja podataka ukoliko do njih dođe
12	CMS - Sučelje za upravljanje korisničkim profilima i pravima pristupa	Omogućuje prilagodbu sučelja te postavljanje različitih razina pristupa na temelju uloga
13	CMS - Implementacija forme za dodavanja novih podataka ovisno o razini pristupa	Unos novih podataka, mogućnosti ovise o razini pristupa korisnika
14	Login - Implementacija forme za prijavu i odjavu korisnika	Jednostavan unos e-mail adrese i lozinke za prijavu, te jednostavnu odjavu kroz jasno dizajnirane i interaktivne UI elemente s

		odgovarajućim vizualnim povratnim informacijama
15	Login - Vidljivost lozinke	Prilikom prijave korisnika u sustav prvotno sakriti lozinku, no omogućiti opciju prikaza lozinke

2.2.2 Nefunkcionalni zahtjevi

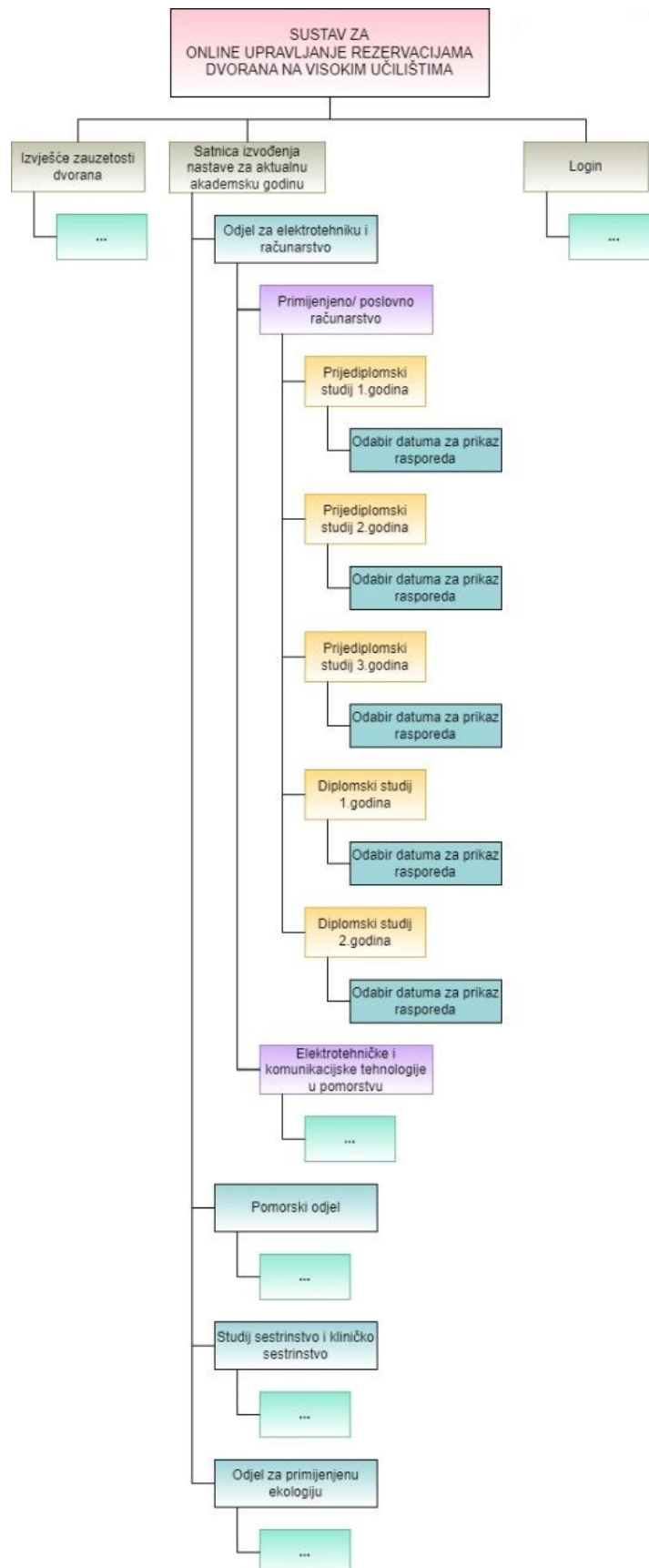
Druga tablica (Tablica 2.) prikazuje nefunkcionalne zahtjeve web aplikacije. Tablica se sastoji od tri stupca koja sadržavaju ID zahtjeva, naziv te opis.

Tablica 2. Nefunkcionalni zahtjevi web aplikacije

ID ZAHTJEVA	NAZIV	OPIS
01	Prilagodljivost	Optimizirati sučelje za različite veličine ekrana i uređaje, što osigurava učinkovito korisničko iskustvo na mobilnim telefonima, tabletima i računalima
02	Osigurati podršku za sve moderne web preglednike (Chrome, Firefox, Safari, Edge)	Osigurati konzistentno prikazivanje i funkcionalnost sučelja na svim platformama, osiguravajući da svi korisnici imaju isto kvalitetno iskustvo
03	Kompatibilnost s barem dvije prethodne verzije popularnih preglednika	Osigurati da sučelje ispravno funkcionira i prikazuje se pravilno na starijim verzijama preglednika, čime omogućuje širem krugu korisnika pristup bez problema
04	Intuitivno korisničko sučelje koje minimizira broj klikova potrebnih za izvršenje osnovnih zadataka	Implementirati brzu i efikasnu interakcija s kojom se poboljšava korisničko iskustvo i produktivnost
05	Učinkovita organizacija informacija i vizualna	Omogućiti jednostavnu navigaciju, poboljšavajući pristup ključnim

	hijerarhija	informacijama i funkcionalnostima unutar sučelja
06	Intuitivna interakcija putem miša	Omogućiti lako i prirodno korištenje sučelja kroz klikove i pomicanje, čime se poboljšava korisničko iskustvo
07	Intuitivno i korisnički prihvatljivo sučelje za sve korisničke uloge	Implementirati lako i učinkovito pristupanje Funkcionalnostima

2.3 Model funkcija (Dijagram dekompozicije funkcija)



Slika 1. Dijagram dekompozicije funkcija

Pomoću gore priložene slike (*Slika 1.*), koja prikazuje dekompozicijski dijagram funkcija, možemo vidjeti dio procesa koji se izvodi na naslovnoj stranici u „*Sustavu za online upravljanje rezervacijama na visokim učilištima*“.

Opis procesa dijagrama:

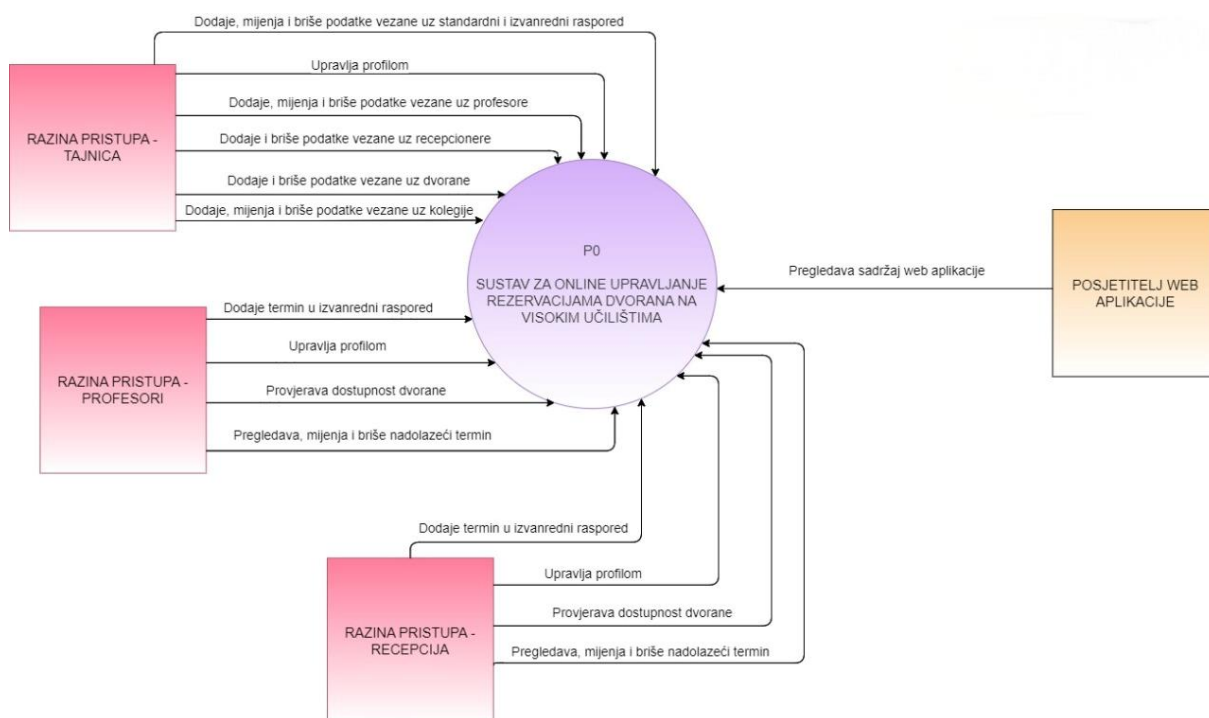
1. **Izvješće zauzetosti dvorana** - Svaki posjetitelj web aplikacije ima pristup izvješću o zauzetosti dvorana koje pruža grafički pregled trenutne i buduće zauzetosti, daljnje funkcije nisu prikazane jer dijagram prikazuje samo određeni segment funkcija sustava.
2. **Login** – Korisničko sučelje za prijavu u aplikaciju, služi korisnicima koji imaju razinu pristupa koja zahtjeva prijavu u sustav. Daljnje funkcije nisu prikazane jer dijagram prikazuje samo određeni segment funkcija sustava.
3. **Satnica izvođenja nastave za aktualnu akademsku godinu** - Svim korisnicima web aplikacije pružena je mogućnost provjere rasporeda po odjelima i smjerovima. Dijeli se na četiri odjela, koja su korištena za demo verziju aplikacije. Na primjeru jednog od njih prikazane su funkcije koje su identične i za ostale odjele.
 - 3.1. **Studij sestrinstvo i kliničko sestrinstvo** – Prikazana funkcija služi za odabir smjera studija sestrinstva i kliničkog sestrinstva, daljnje funkcije nisu prikazane jer dijagram prikazuje samo određeni segment funkcija sustava.
 - 3.2. **Pomorski odjel** - Prikazana funkcija služi za odabir smjera na pomorskom odjelu, daljnje funkcije nisu prikazane jer dijagram prikazuje samo određeni segment funkcija sustava.
 - 3.3. **Odjel za primijenjenu ekologiju** - Prikazana funkcija služi za odabir smjera na odjelu za primijenjenu ekologiju, daljnje funkcije nisu prikazane jer dijagram prikazuje samo određeni segment funkcija sustava.
 - 3.4. **Odjel za elektrotehniku i računarstvo** – Prikazana funkcija služi za odabir smjera na odjelu za elektrotehniku i računarstvo. Dijeli se na smjerove: „*Primijenjeno/poslovno računarstvo*“ i „*Elektrotehničke i komunikacijske tehnologije u pomorstvu*“.
 - 3.4.1. **Elektrotehničke i komunikacijske tehnologije u pomorstvu** – Prikazana funkcija služi za odabir godine određenog studija. Daljnje funkcije nisu prikazane jer dijagram prikazuje samo određeni segment funkcija sustava.

3.4.2 Primijenjeno/poslovno računarstvo – Prikazana funkcija služi za odabir godine određenog studija. Sastoji se od prijediplomskog studija u trajanju od tri godine i diplomskog studija u trajanju od dvije godine. Nakon odabira godine implementirana je mogućnost filtriranja rasporeda po datumu.

- **Godine studija** – Funkcija nudi mogućnost filtriranja rasporeda po datumu.
 - **Odabir datuma za prikaz rasporeda** – Funkcija koja služi odabiru datuma nakon čega se prikaže konačan raspored za odabrani datum.

2.4 Model procesa

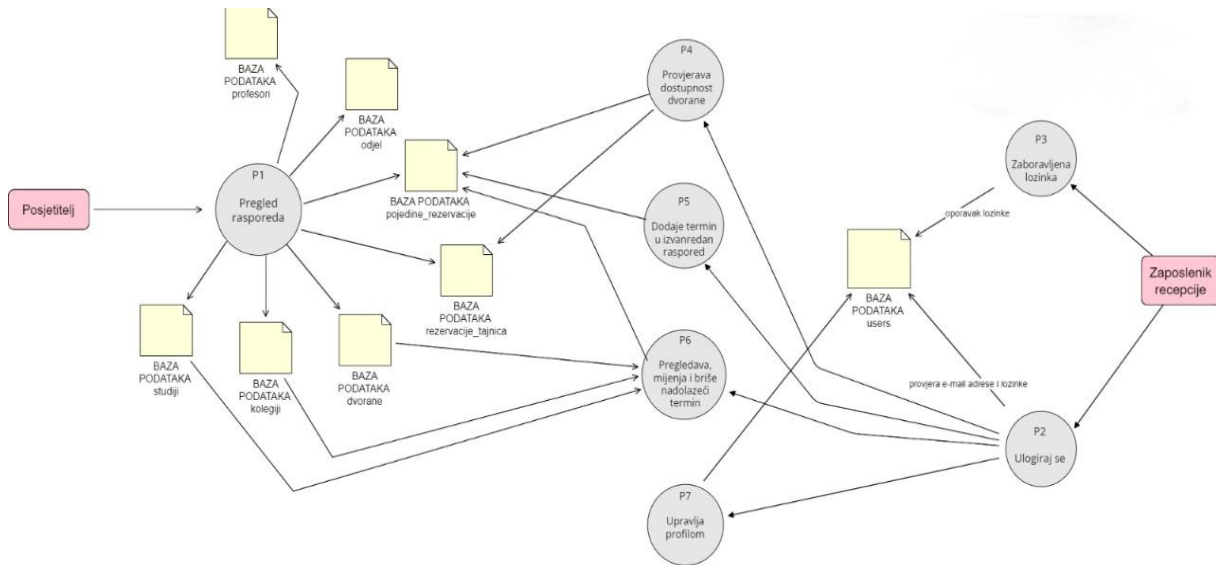
2.4.1 Dijagram konteksta (0.razina)



Slika 2. Dijagram konteksta (0. razina)

Prikazani dijagram konteksta (*Slika 2.*) pruža uvid u upite prema sustavu. Implementirane su tri razine pristupa, te dijagram jasno pokazuje moguće upite prema sustavu svakog od te tri razine. Među korisnike sustava, uz navedene razine koje su tajnice, profesori i zaposlenici recepcije, zbrajaju se i posjetitelji web aplikacije koji također imaju mogućnost poslati upit sustavu.

2.4.2 Pregledni dijagram glavnih procesa (1.razina)



Slika 3. Dijagram glavnih procesa (1. razina)

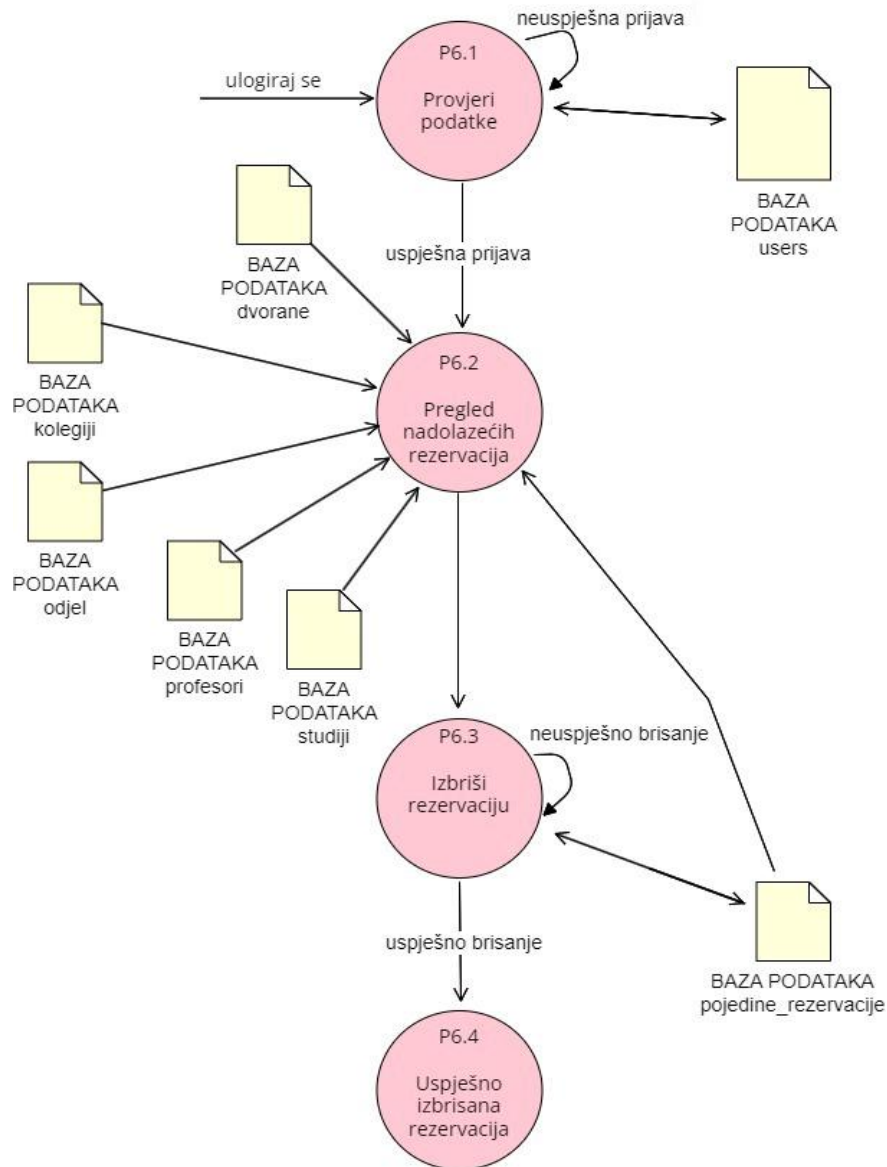
Navedeni dijagram glavnih procesa (*Slika 3.*) obuhvaća samo dio ključnih procesa sustava. Odabrani procesi su dostupni posjetiteljima web aplikacije i zaposlenicima visokih učilišta s razinom pristupa - recepcija.

U dijagramu glavnih procesa nalazi se:

- sedam procesa
- dva vanjska entiteta (posjetitelj i zaposlenik recepcije)
- osam spremišta (baza podataka „profesori“, baza podataka „odjel“, baza podataka „studiji“, baza podataka „kolegiji“, baza podataka „dvorane“, baza podataka „pojedine_rezervacije“, baza podataka „rezervacije_tajnica“, baza podataka „users“)

Dijagram uključuje glavne procese vezane uz posjetitelje web aplikacije i zaposlenike recepcije visokih učilišta, neki od njih su „Zaboravljena lozinka“, „Upravljanje profilom“, „Pregledavanje, mijenjanje i brisanje nadolazećih termina“, itd...

2.4.3 Detaljni dijagram za odabrani proces (2.razina)



Slika 4. Dijagram za odabrani proces (2. razina)

U prikazanom dijagramu (Slika 4.), detaljno je razrađen proces brisanja prethodno unesene rezervacije zaposlenika recepcije na visokom učilištu. Prvotno je potrebna prijava u sustav, ako je neuspješna potrebno je ponoviti prijavu. Nakon uspješne prijave prikazuje se sučelje odabira opcija, gdje korisnik bira prikaz postojećih rezervacija. Nakon uspješnog pokušaja brisanja rezervacije, korisnik će dobiti potvrdu o tome u obliku obavijesti, dok je prilikom neuspješnog brisanja potrebno ponoviti radnju. Dijagram sadrži četiri procesa i sedam spremišta.

3. KORIŠTENE TEHNOLOGIJE

U ovom poglavlju detaljno će se analizirati i objasniti tehnologije korištene za izradu web sjedišta, pružajući sveobuhvatan pregled njihovih funkcionalnosti i doprinosa razvoju aplikacije. HTML je korišten za strukturiranje sadržaja, CSS za stilizaciju i vizualno oblikovanje, dok je JavaScript omogućio dinamičke funkcionalnosti i interaktivnost. Implementacija je bazirana na WAMP serveru, koji uključuje Apache, MySQL i PHP, što je omogućilo lokalni razvoj i testiranje. Poseban naglasak je na front-end razvoju, dok je za vizualizaciju podataka i analitičke prikaze korišten alat Tableau, koji je unaprijedio razumijevanje informacija unutar sustava.

3.1 HTML

HyperText Markup Language ili skraćeno HTML omogućuje korisnicima da kreiraju i strukturiraju dijelove, odlomke i veze s pomoću elemenata, oznaka i atributa. Međutim, vrijedno je napomenuti da se HTML ne smatra programskim jezikom jer ne može stvoriti dinamičku funkcionalnost.

HTML je stvorio Tim Berners-Lee krajem 1991. godine, ali nije službeno objavljen te godine, nego je objavljen 1995. godine kao HTML 2.0. Verzija HTML 4.01 objavljena je krajem 1999. godine, te je bila glavna verzija HTML-a. HTML se vrlo brzo razvija i evoluirao je ažuriranjem različitih verzija. Puno ranije nego što se uvedu revidirani standardi i specifikacije, svaka verzija je omogućavala korisniku da kreira web stranice na puno lakši i ljepši način i učini web stranice vrlo učinkovitim.

HTML 1.0 objavljen je 1993. s namjerom dijeljenja informacija koje mogu biti čitljive i dostupne putem web preglednika. Ali nije puno programera bilo uključeno u izradu web stranica. Dakle, jezik također nije rastao. Zatim dolazi HTML 2.0, objavljen 1995., koji sadrži sve značajke HTML-a 1.0 zajedno s nekoliko dodatnih značajki, koje su ostale kao standardni označni jezik za dizajn i izradu web-mjesta do siječnja 1997. i usavršavale različite temeljne značajke HTML-a. Zatim dolazi HTML 3.0, gdje je Dave Raggett predstavio novi rad ili nacrt o HTML-u. Uključuje poboljšane nove značajke HTML-a, dajući moćnije karakteristike za webmastere u dizajniranju web stranica. Ali ove poboljšane značajke novog HTML-a usporile su preglednik u primjeni daljnjih poboljšanja. Zatim dolazi HTML 4.01, koji se naširoko koristi i bio je uspješna verzija HTML-a prije HTML-a 5.0, koji je trenutno

objavljen i koristi se širom svijeta. HTML 5 se može reći za proširenu verziju HTML-a 4.01, koja je objavljena 2012. godine [1].

HTML se koristi u brojnim područjima:

- **Web development** - Programeri koriste HTML kod za dizajniranje načina na koji preglednik prikazuje elemente web stranice, kao što su tekst, hiperveze i medijske datoteke.
- **Internet navigation** - Korisnici se mogu lako kretati i umetati veze između povezanih stranica i web-mjesta jer se HTML uvelike koristi za ugradnju hiperveza.
- **Web documentation** - HTML omogućuje organiziranje i formatiranje dokumenata, slično Microsoft Wordu [2].

Prosječna web stranica uključuje nekoliko različitih HTML stranica. Na primjer, početna stranica, stranica o informacijama i stranica za kontakt sve bi imale zasebne HTML datoteke. HTML dokumenti su datoteke koje završavaju nastavkom .html ili .htm. Web preglednik čita HTML datoteku i prikazuje njezin sadržaj tako da ga korisnici interneta mogu vidjeti. Sve HTML stranice imaju niz HTML elemenata koji se sastoje od skupa oznaka i atributa. HTML elementi su građevni blokovi web stranice. Oznaka govori web pregledniku gdje element počinje i gdje završava, dok atribut opisuje karakteristike elementa. Svaki HTML dokument mora započeti deklaracijom `<!DOCTYPE` kako bi web preglednik obavijestio o vrsti dokumenta (*Slika 5.*). Uz HTML5, doctype HTML javna deklaracija će biti:

The image shows the HTML doctype declaration `<!DOCTYPE html>` in a monospaced font. The characters are colored: the opening angle bracket is blue, the exclamation mark is red, the word 'DOCTYPE' is green, and 'html' is blue. The closing angle bracket is also blue. The text is centered on a black background.

Slika 5. Početna deklaracija

Tri osnovna dijela su:

- **Početna oznaka** – koristi se za navođenje gdje element počinje djelovati. Oznaka je omeđena kutnim zagradama za otvaranje i zatvaranje, npr. upotrijebite početnu oznaku `<p>` za stvaranje odlomka.
- **Sadržaj** – ovo je izlaz koji vide drugi korisnici.

- **Završna oznaka** - isto kao i početna oznaka, ali s kosom crtom ispred naziva elementa, npr. </p> za završetak odlomka.

Kombinacija ova tri dijela stvorit će HTML element, što prikazuje „Slika 6“.

```
<p> Ovako se dodaje odlomak u HTML-u. </p>
```

Slika 6. Paragraf u HTML kodu

Neki od osnovnih HTML elemenata su:

- **<DOCTYPE! html>** - deklaracija vrste dokumenta (tehnički nije oznaka). Deklarira dokument kao HTML dokument. Ne razlikuje mala i velika slova.
- **<html>** - osnovni HTML element. Svi ostali elementi sadržani su u njemu. Može se reći da je ovaj element roditelj a svi elementi u njemu njegova djeca.
- **<head>** - predstavlja zaglavlje dokumenta koje sadržava oznake poput <title>, <link>, itd...
- **<title>** - naslov stranice, on se pojavljuje na naslovnoj traci web preglednika kada se otvori navedeno web sjedište.
- **<body>** - tijelo dokumenta. Sadrži oznake poput <h1>, <p>, itd...
- **<h1>** - **<h6>** - oznaka „h“ označava naslov dok broj pored njega razinu naslova. Postoji šest razina naslova, slijedno od najvećeg prema sporednim.

Spoj osnovnih elemenata HTML-a čini strukturu koja je prikazana na „Slika 7.“.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

Slika 7. HTML struktura

Potrebno je spomenuti i neke od atributa oznake:

- **Alt** - alternativni tekst koji će biti prikazan umjesto slike.
- **Src** - određuje mjesto (web adresu) gdje se nalazi sadržaj koji treba biti prikazan na web stranici.
- **Href** - određuje lokaciju (web adresu) na koju je link povezan.
- **Style** - koristi se za dodavanje CSS stila za određeni element (veličina, boja, pozicija itd...)
- **Lang** - definira jezik elemenata
- **Id** – određuje jedinstvenu identifikaciju (id) za element [2]

3.2 CSS

CSS (*Cascading Style Sheets*) je stilski jezik koji opisuje kako se HTML elementi prikazuju na ekranu. CSS upravlja izgledom web stranice. Koristeći CSS, možete kontrolirati boju teksta, stil fontova, razmak između paragrafa, veličinu i raspored stupaca, koje se pozadinske slike ili boje koriste, dizajn izgleda, prilagođavanje u prikazu za različite uređaje i veličine zaslona kao i niz drugih učinaka. CSS sadrži puno pozitivnih stvari te znatno unaprjeđuje web stranicu [3].

CSS ima ključnu ulogu u modernom web dizajnu zbog nekoliko razloga:

- **Odvajanje stila i sadržaja:** CSS omogućava odvajanje vizualnih elemenata od strukture sadržaja, što pojednostavljuje održavanje i ažuriranje web stranica.
- **Ponovna upotreba kodova:** Stilovi definirani u CSS-u mogu se koristiti na više stranica ili elemenata, čime se smanjuje potreba za dupliciranjem koda.
- **Konzistentnost dizajna:** Korištenje CSS-a omogućuje dosljedan izgled web stranica, što poboljšava korisničko iskustvo i percepciju brenda.
- **Prilagodljivi dizajn:** CSS omogućava stvaranje prilagodljivih web stranica koje se prilagođavaju različitim veličinama ekrana, što je ključno u eri mobilnog interneta.

CSS kod se sastoji od CSS-pravila. Deklaracija i selektor dio su svakog CSS pravila. Ukoliko se CSS piše u istom dokumentu u kojem i HTML stranica, dio CSS smješta se u „*head*“ odjeljak te unutar elementa `<style>`, no također se može pisati i u odvojenoj datoteci s ekstenzijom `.css`.

Na „Slika 8.“ je prikazan dio CSS koda koji nam govori da je sav tekst koji je unutar <h1> elementa crvene boje sa sivom pozadinom.

```
▼ h1{  
  color: ■ red;  
  background-color: ■ grey;  
}
```

Slika 8. Osnovna sintaksa CSS-a

Ukoliko CSS pišemo u istoj datoteci s HTML-om postoji više načina pridruživanja određenog CSS-a željenom dijelu HTML atributa. Stoga postoje tri načina implementiranja CSS-a u HTML: vanjski, unutarnji i ugrađeni. Vanjski način je odvojena datoteka ekstenzijom .css koju moramo referencirati unutar naše HTML datoteke. Ovaj način je dosta praktičan jer je ponovno iskoristiv te je dosta pregledan za pisanje te odrađivanje popravaka po potrebi. Na „Slika 9.“ je prikazan način uključivanja CSS datoteke naziva „style“ u HTML dokument.

```
<link rel="stylesheet" href="style.css">
```

Slika 9. Referenciranje CSS datoteke

Unutarnji način implementiranja podrazumijeva pisanje CSS koda unutar samog HTML dokumenta. Zahtijeva dodavanje oznake <style> u odjeljak <head> unutar kojeg se piše kod. Ovaj način također ima tri načina dodavanja CSS koda određenim HTML atributima. Prvi način je da kod pišemo za skupine atributa kao što su body, h2, itd..., te je taj način prikazan na „Slika 10.“.


```

<style>
body{
  background-color: ■ red;
  text-align: center;
}
h2{
  color: ■ purple;
}
</style>

```

Slika 10. Primjer unutarnjeg CSS-a

Također možemo CSS kod namijeniti određenom HTML atributu, ovaj način je prikladan kada želimo jednom atributu pridružiti točno određeni CSS kod. To možemo napraviti primjenom ID selektora. Definira se oznakom ljestvice (#) te se atributu pridružuju pomoću ID-a dok je ostala sintaksa ista. Primjer takvog CSS-a je prikazan na „Slika 11.“.

```

<style>
#selectorid{
  background-color: ■ beige;
  text-decoration: solid;
  font-size: small;
}
</style>
</head>
<body>
  <p id="selectorid"> Odlično! </p>
</body>

```

Slika 11. ID selektor

Također je moguće dodijeliti CSS atributu pomoću „class“ selektora. Definira se oznakom točka (.) te se atributu pridružuje pomoću „class“. Kako bi se to napravilo potrebno je na mjesto znaka ljestvice (#) staviti znak točku (.).

Posljednji primjer dodjeljivanja CSS-a je ugrađeni. Unutar početne oznake HTML atributa, nakon oznake samog atributa dodamo „*style*“, znak jednako (=) te unutar navodnih znakova pišemo CSS istom sintaksom. Primjer je prikazan na „*Slika 12.*“ [4].

```
<p style="background-color: blueviolet; text-decoration: solid; font-size: small;"> Odlično! </p>
```

Slika 12. Ugrađeni CSS

3.2.1. Vanilla CSS

Vanilla CSS označava korištenje „čistog“ CSS-a bez dodatnih pripremnih alata, okvira ili biblioteka. Ova metoda stiliziranja web stranica često se preferira zbog svoje jednostavnosti i direktnog pristupa gdje se koriste samo standardne značajke CSS-a bez dodatne sintakse koju nude pripremnici alati poput SASS-a ili LESS-a. Vanilla CSS nudi jednostavan i direktan način za kontrolu stila web stranica, omogućujući developerima da precizno definiraju izgled i dojam korisničkog sučelja.

Vanilla CSS omogućuje programerima da se fokusiraju na temeljne aspekte CSS-a, što pomaže u stjecanju dubokog razumijevanja jezika. Osim toga, Vanilla CSS eliminira potrebu za dodatnim alatima, smanjujući kompleksnost razvoja i olakšavajući održavanje koda. Budući da se radi o standardnom jeziku, kod napisan u Vanilla CSS-u obično je kompatibilan s različitim preglednicima bez potrebe za dodatnim prilagodbama.

Prednosti Vanilla CSS-a:

- **Jednostavnost i brzina:** Vanilla CSS se može lako učiti i primjenjivati, što ga čini idealnim za početnike i male projekte.
- **Manja složenost:** Bez dodatnih alata ili pripremnih alata, razvojni proces je često brži i manje složen.
- **Kompatibilnost:** Budući da se radi o standardiziranom jeziku, Vanilla CSS je kompatibilan s gotovo svim modernim preglednicima.

Unatoč svojim ograničenjima, Vanilla CSS ostaje važan alat u skupu alata web developera. Njegova osnovna struktura služi kao temelj za naprednije stilizacijske tehnike i alate. U današnje vrijeme, s naglaskom na responzivni dizajn, Vanilla CSS igra ključnu ulogu u stvaranju web stranica koje se prilagođavaju različitim veličinama ekrana i uređajima. CSS Grid i Flexbox, dvije moćne značajke CSS-a, omogućuju kreiranje kompleksnih „*layouta*“

bez potrebe za dodatnim alatima, čime dodatno jačaju važnost čistog CSS-a. Relevantan je u kontekstu modernih trendova kao što su „*dark mode*“ dizajni i pristupačnost, gdje programeri trebaju imati kontrolu nad detaljima svakog elementa. Kako tehnologija napreduje, očekuje se da će se CSS nastaviti razvijati s novim značajkama koje će olakšati stilizaciju i interaktivnost, ali osnove koje nudi Vanilla CSS ostat će ključne za razumijevanje i pravilnu primjenu tih noviteta [5].

3.3 JavaScript

JavaScript ili skraćeno JS - skriptni programski jezik koji se koristi i na strani korisnika i na strani poslužitelja. HTML i CSS web sjedištu daju strukturu i izgled dok ih JS čini interaktivnima. Većina internetskih preglednika sadržava ugrađene mehanizme koji mogu generirati JS, te se JS naredbe mogu unositi izravno u HTML dokument, a internetski preglednici će ih ispravno interpretirati [6].

JavaScript je kreirao Brendan Eich u Netscape Communications 1995. Netscape i Eich dizajnirali su JavaScript kao skriptni jezik za korištenje s vodećim web preglednikom tvrtke, Netscape Navigator. U početku poznat kao „*LiveScript*“, Netscape je promijenio ime u JavaScript kako bi ga mogli pozicionirati kao suputnika za jezik Java, proizvod njihovog partnera, Sun Microsystems. Međutim, osim nekih površnih sintaksnih sličnosti, JavaScript ni na koji način nije povezan s programskim jezikom Java. Nakon njegovog izdavanja, sve više i više preglednika počelo je dodavati podršku za JavaScript. Ipak, veći dio svoje povijesti JavaScript se nije smatrao ozbiljnim programskim jezikom. Njegova najranija izdanja patila su od značajnih problema s performansama i sigurnosti, ali programeri nisu imali alternativu. Ako su htjeli pokrenuti programe u pregledniku, morali su koristiti JavaScript. Godine 2008. stvaranje Googleovog otvorenog koda Chrome V8, JavaScript mehanizam visokih performansi, omogućilo je ključnu prekretnicu za JavaScript. Naknadno širenje brzih JavaScript mehanizma omogućilo je programerima izradu sofisticiranih aplikacija temeljenih na pregledniku s performansama koje su se natjecale s aplikacijama za stolna računala i mobilne uređaje. Ubrzo nakon toga, Ryan Dahl je objavio *open-source, cross-platform* okruženje pod nazivom Node.js. Pružao je način za pokretanje JavaScript koda izvan preglednika. To je oslobodilo JavaScript iz okvira preglednika i dovelo izravno do trenutačne popularnosti JavaScripta [7].

JS dodaje interaktivnost web sjedištima poput prikaza ili skrivanja informacija klikom na tipku, promjena boje tipke kada na nju dođemo mišem, prikaz animacija, itd... Također JavaScript se koristi u izradi web i mobilnih aplikacija. Programeri mogu koristiti različite JavaScript okvire (eng. *Framework*) za razvoj i izgradnju web i mobilnih aplikacija. JavaScript okviri su zbirke biblioteka (eng. *Libraries*) JavaScript koda koje programerima pružaju unaprijed napisani kod za korištenje programske zadatke, okvir za izgradnju web-mjesta ili web-aplikacija. JS se još koristi za razvoj web servera i razvoj poslužiteljskih aplikacija (eng. *Server applications*), te za razvoj igara [8].

JavaScript možemo koristiti na više načina koje ćemo predstaviti. HTML oznake imaju attribute za događaje koji izvršavaju određen kod kada se pokrenu, npr. na primjeru tipke ćemo pokazati prvi način korištenja JS-a. Primjer je prikazan na „Slika 13.“.

```
<button onclick="alert('Upravo ste pritisnuli tipku')">Klikni me!</button>
```

Slika 13. Ugrađeni JavaScript

Sljedeći način za korištenje JavaScript-a je unutarnji JS. Unutar `<script>` oznaka pišemo JS kod. Oznake možemo smjestiti u bilo koji dio HTML dokumenta. Primjer je na „Slika 14.“.

```
<button onclick="MyFunction()">Klikni me!</button>

<script>
  function MyFunction(){
    alert("Unutarnji JavaScript")
  }
</script>
```

Slika 14. Unutarnji JavaScript

Posljednji način je vanjski JS. Kod JavaScripta pišemo u odvojenoj datoteci koju potom uključimo u HTML datoteku. Atribut „*src*“ oznake `script` omogućuje nam uključiti vanjsku datoteku. Vanjska JS datoteka ima `.js` ekstenziju. Uključivanje je prikazano na „Slika 15.“ [9].

```
<script src="/script.js"></script>
```

Slika 15. Uključivanje vanjske JavaScript datoteke

3.3.1 Vanilla JavaScript

Vanilla JavaScript odnosi se na korištenje „čistog“ JavaScripta bez dodatnih biblioteka ili okvira. Pojam „*vanilla*“ označava osnovnu, nepromijenjenu verziju jezika, a koristi se kako bi se naglasila upotreba standardnog JavaScripta u razvoju web aplikacija. U kontekstu modernog web razvoja, Vanilla JavaScript predstavlja osnovnu funkcionalnost jezika, koja omogućuje manipulaciju DOM-om, obrada događaja, asinkrono programiranje i druge ključne zadatke bez oslanjanja na dodatne alate ili knjižnice.

Vanilla JavaScript predstavlja izvrstan izbor u nekoliko specifičnih scenarija. Prvo, za male i srednje velike projekte s manje kompleksnošću, gdje nije potrebna dodatna funkcionalnost ili složena arhitektura, Vanilla JavaScript je brz i jednostavan izbor. Njegova osnovna priroda omogućava brzu implementaciju i jednostavno održavanje, bez potrebe za učenjem i integracijom dodatnih knjižnica ili okvira. Drugo, Vanilla JavaScript je idealan za učenje i eksperimentiranje. Ako ste novi u programiranju ili želite bolje razumjeti temeljne koncepte JavaScripta, rad s Vanilla JavaScriptom pruža solidnu osnovu za razumijevanje jezika i njegovih osnovnih funkcionalnosti. Konačno, za projekte koji zahtijevaju maksimalnu optimizaciju performansi, smanjenje ovisnosti o vanjskim knjižnicama može značajno poboljšati brzinu i učinkovitost. Korištenje „čistog“ JavaScripta omogućava programerima da minimiziraju učitavanje dodatnih resursa i osiguraju optimalne performanse svojih aplikacija.

Prednosti Vanilla JS:

1. Performanse:

- Brže učitavanje: Bez potrebe za učitavanjem vanjskih biblioteka, stranice se brže učitavaju, što poboljšava korisničko iskustvo.
- Manje resursa: Korištenje samo JavaScripta smanjuje veličinu datoteka i potrebu za dodatnim resursima.

2. Kontrola i fleksibilnost:

- Potpuna kontrola: Programeri imaju potpunu kontrolu nad kodom, omogućujući im da optimiziraju performanse i prilagode funkcionalnost specifičnim potrebama projekta.

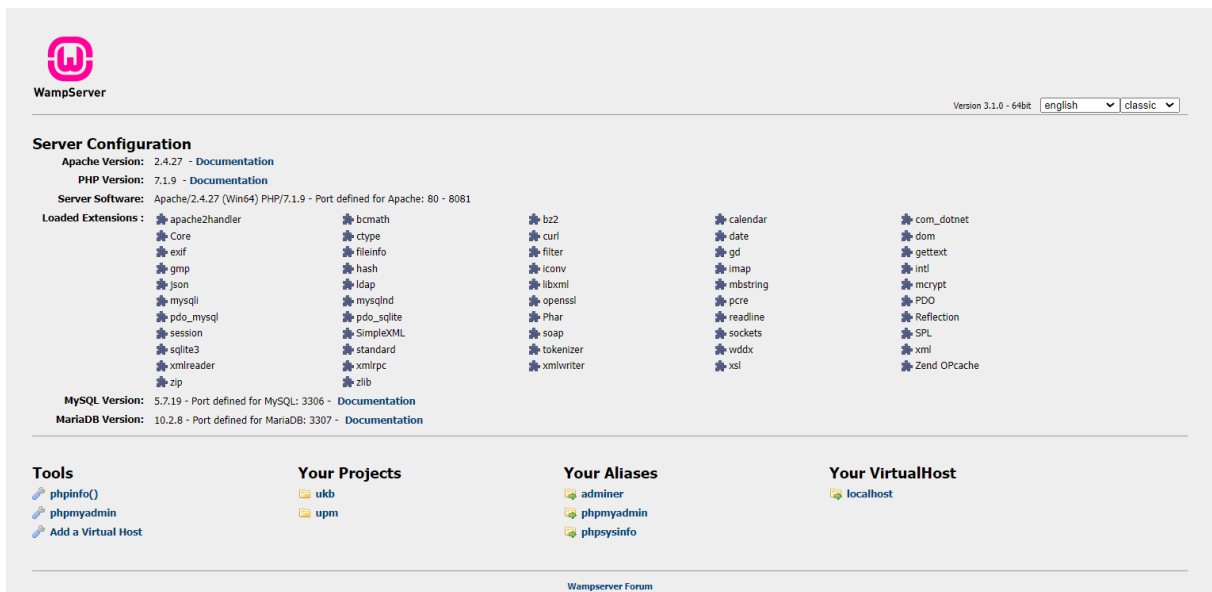
- Jednostavna integracija: Korištenje čistog JavaScripta omogućuje jednostavnu integraciju s HTML-om i CSS-om bez potrebe za dodatnim slojevima apstrakcije.
3. Učenje i razumijevanje:
- Osnovna pozadina: Rad s Vanilla JavaScriptom pomaže programerima da bolje razumiju temelje jezika, što je korisno za učenje i rješavanje složenijih problema u budućnosti.
 - Manje ovisnosti: Razumijevanje osnovnih principa JavaScripta smanjuje ovisnost o vanjskim alatima i omogućuje rješavanje problema s osnovnim znanjem jezika [10].

3.4 WampServer

WAMP je akronim koji označava kombinaciju softverskih komponenti: Windows, Apache, MySQL/MariaDB i PHP. Nakon instalacije WAMP-a na operativni sustav (Windows) automatski se instaliraju Apache, MySQL i PHP. Iako se ove komponente mogu instalirati odvojeno, obično dolaze u paketu. Potrebno je znati da WAMP potječe od LAMP gdje „L“ označava Linux. Jedina razlika između ova dva akronima je u tome što se WAMP koristi za Windows, dok se LAMP koristi za operativne sustave temeljene na Linuxu. Na (Slika 16.) prikazan je izgled ekrana za pristup WampServeru-u.

Pregled akronima WAMP:

- W – Windows: Ova komponenta označava da je WAMP namijenjen radu na Windows operativnom sustavu, ali postoji i MAMP koji je baziran na Apple Mac proizvode i njihov iOS operativni sustav.
- A – Apache: Apache je poslužiteljski softver koji upravlja pružanjem web stranica. Kada se zatraži određena stranica, Apache obrađuje zahtjev putem HTTP-a i prikazuje željenu stranicu.
- M – MySQL: MySQL je sustav za upravljanje bazama podataka. Pohranjuje sve relevantne informacije kao što su sadržaj stranica, korisnički profili i drugi podaci.
- P – PHP: PHP je programski jezik koji se koristi za razvoj dinamičkih web stranica i aplikacija. Povezuje sve komponente WAMP-a, radeći u suradnji s Apacheom i komunicirajući s MySQL-om [11].



Slika 16. Izgled ekrana za pristup WampServer-u

WAMP je višenamjenski alat za web razvoj i testiranje, posebno popularan među programerima u Windows okruženju. Koristi se za postavljanje lokalnog razvojnog okruženja, omogućujući pisanje, testiranje i otklanjanje pogrešaka web aplikacija na osobnom računalu. Omogućuje razvoj dinamičkih web stranica koristeći Apache, MySQL i PHP, te podržava izradu prototipova. Prednosti WAMP-a uključuju jednostavnu instalaciju, besplatan je i otvorenog koda, te pruža integrirano razvojno okruženje. No, WAMP također ima ograničenja, kao što su specifičnost na Windows platformu i razlike u odnosu na proizvodna okruženja koja često koriste Linux. Pojedine komponente mogu imati problema s kompatibilnošću prilikom ažuriranja, a sigurnosne postavke nisu uvijek optimizirane za javne poslužitelje. Usprkos ovim nedostacima, WAMP ostaje izuzetno koristan alat za razvoj i učenje web tehnologija [12].

3.5 Tableau

Tableau je platforma za vizualnu analitiku koja mijenja način na koji koristimo podatke za rješavanje problema – omogućuje ljudima i organizacijama da u potpunosti iskoriste svoje podatke. Osnovan je 2003. godine kao rezultat ambicioznog istraživačkog projekta u okviru računalnih znanosti na Stanfordskom sveučilištu, čiji je cilj bio unaprijediti protok analize podataka i učiniti ih dostupnijima kroz inovativne metode vizualizacije. Suosnivači Chris Stolte, Pat Hanrahan i Christian Chabot razvili su i patentirali temeljnu tehnologiju Tableau-a poznatu kao VizQL, koja omogućava vizualno izražavanje podataka

prevođenjem intuitivnih radnji poput povlačenja i ispuštanja u sofisticirane upite za podatke putem lako razumljivog korisničkog sučelja. Od svog osnutka, Tableau je kontinuirano ulagao u istraživanje i razvoj s neusporedivom brzinom, stvarajući rješenja koja omogućuju korisnicima da efikasnije dolaze do odgovora i otkrivaju neočekivane uvide iz složenih skupa podataka. Ova posvećenost napretku osigurava da Tableau ostane lider u domeni analitičkih alata, pružajući napredne mogućnosti za analizu podataka i poticanje novih otkrića.

Tableau integrira napredne tehnike kao što su strojno učenje, statistika, prirodni jezik i pametna priprema podataka, s ciljem poticanja ljudske kreativnosti u analizi podataka. Platforma ne samo da pruža sveobuhvatan i integriran analitički okvir, već nudi i dokazano učinkovite resurse za omogućavanje implementacije i skaliranja kulture vođene podacima, koja doprinosi otpornosti i stvaranju vrijednosti kroz snažne rezultate. Danas, organizacije diljem svijeta od neprofitnih organizacija do globalnih poduzeća, u svim industrijama i odjelima osnažuju svoje timove koristeći Tableau kako bi potaknuli promjene temeljem podataka [13].

Napredne funkcionalnosti koje uključuje:

1. **Vizualizacija podataka:** Tableau omogućuje korisnicima stvaranje složenih i interaktivnih vizualizacija podataka, olakšavajući razumijevanje složenih informacija kroz grafički prikaz.
2. **Integracija i povezivanje:** Platforma se može integrirati s različitim izvorima podataka, uključujući baze podataka, excel tablice, cloud usluge i mnoge druge, omogućujući korisnicima da analiziraju podatke iz različitih izvora unutar jedinstvenog okvira.
3. **Interaktivne Dashboarde:** Korisnici mogu kreirati dinamične *dashboarde* koji omogućuju interaktivno istraživanje podataka i brzo donošenje odluka na temelju stvarnih uvida.
4. **Analitika i prediktivni uvidi:** Tableau koristi napredne tehnike poput strojnog učenja i statistike za pružanje prediktivnih uvida i unapređenje analitičkih sposobnosti korisnika.
5. **Suradnja i dijeljenje:** Platforma omogućuje jednostavno dijeljenje uvida i suradnju među timovima, potičući kulturu vođenu podacima unutar organizacija.

Tableau se izdvaja kao izuzetno moćna, sigurna i fleksibilna analitička platforma, koja nudi niz značajnih prednosti. Njegova jednostavnost korištenja je ključna karakteristika; intuitivno sučelje omogućuje korisnicima svih razina iskustva da brzo usvoje i učinkovito primjenjuju alate za analizu podataka. Platforma se može brzo implementirati, omogućujući korisnicima da započnu s analizom podataka gotovo odmah nakon instalacije. Fleksibilnost Tableau-a omogućava visok stupanj prilagodbe, što korisnicima omogućuje prilagodbu alata prema specifičnim potrebama svojih projekata i poslovnih zahtjeva. Osim toga, Tableau podržava povezivanje s različitim izvorima podataka, čime se povećava svestranost u analizi i izvještavanju. Platforma također doprinosi razvoju kulture vođene podacima unutar organizacija, omogućujući učinkovito dijeljenje uvida i poticanje razumijevanja podataka među zaposlenicima. Ove prednosti zajedno omogućuju organizacijama da poboljšaju analitičke kapacitete, unaprijede donošenje odluka i osnaže svoje timove za rad s podacima [14].

4. BAZA PODATAKA

Baze podataka predstavljaju temeljni element suvremenih informacijskih sustava, omogućujući pohranu, organizaciju i upravljanje velikim količinama podataka na strukturiran način. Baza podataka može se definirati kao skup povezanih podataka koji su organizirani prema određenom modelu, omogućujući jednostavan pristup, upravljanje i ažuriranje tih podataka. U pravilu, ove podatke upravlja sustav za upravljanje bazom podataka (DBMS), koji osigurava integritet, sigurnost i konzistentnost podataka, kao i podršku za različite oblike pretraživanja i analiza.

Strukturirani upitni jezik (SQL) najčešće je korišten jezik za interakciju s relacijskim bazama podataka. Kroz SQL moguće je definirati, manipulirati i kontrolirati podatke, a njegova moć leži u mogućnosti složenih upita i kombinaciji podataka iz više tablica. Relacijski model baza podataka, koji organizira podatke u tablice povezane međusobnim odnosima, pokazao se kao izuzetno učinkovit i fleksibilan za različite primjene, od jednostavnih do vrlo složenih sustava.

Primjena baza podataka u svakodnevnom životu obuhvaća širok spektar aktivnosti, često neprimjetnih korisniku, ali ključnih za funkcioniranje modernih društava. Primjerice, financijske institucije koriste složene baze podataka za praćenje milijardi transakcija, upravljanje računima korisnika i analiziranje tržišnih trendova u stvarnom vremenu.

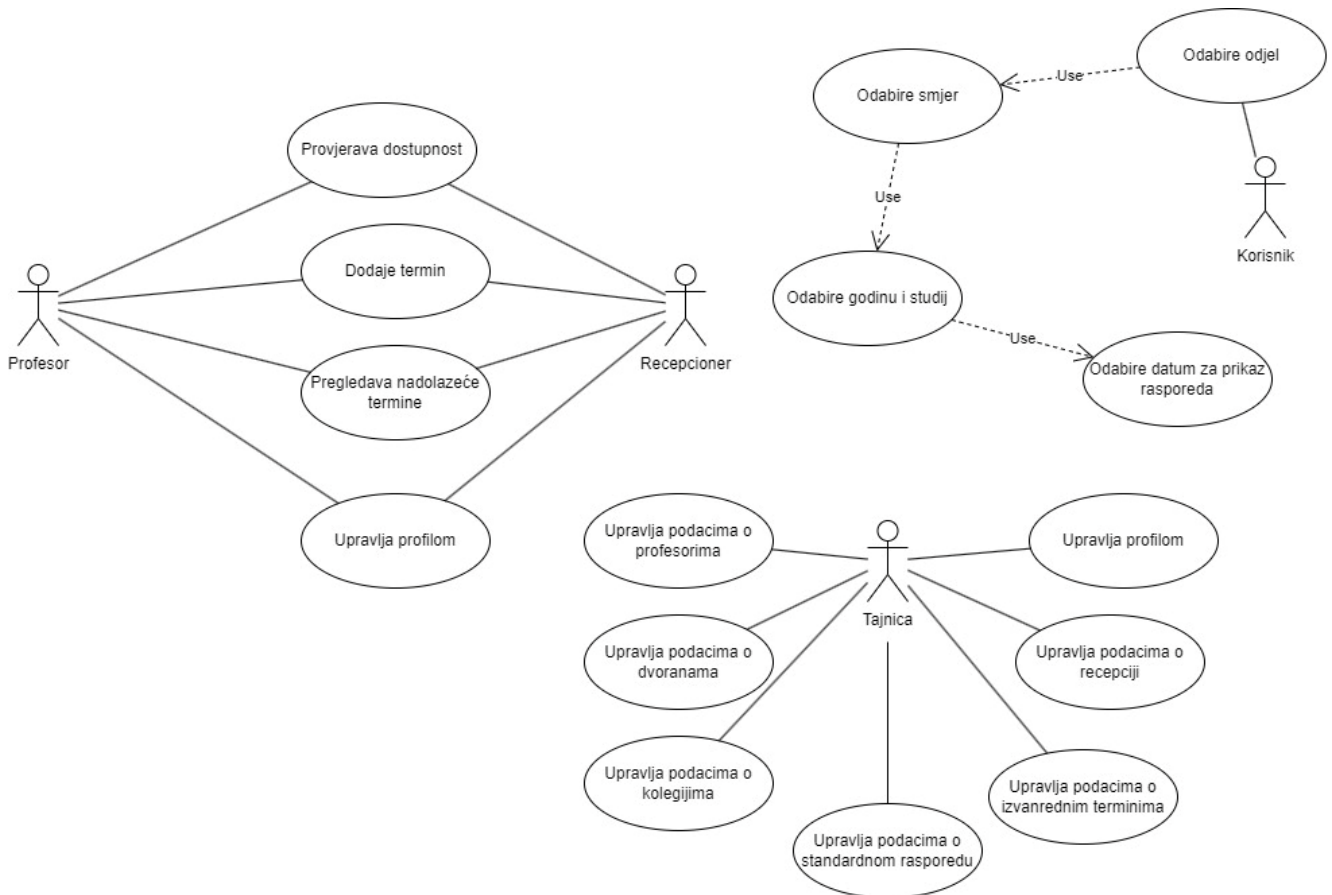
U svijetu koji postaje sve više digitaliziran, uloga baza podataka postaje sve značajnija. Veliki podaci (*Big Data*), umjetna inteligencija (*AI*), internetske tehnologije i mobilne aplikacije sve više ovise o sposobnosti sustava da brzo i učinkovito pristupe i obrađuju ogromne količine podataka. To otvara nova pitanja i izazove vezane uz skalabilnost, sigurnost i privatnost podataka, što čini istraživanje u području baza podataka ne samo tehnički izazovnim, već i izuzetno relevantnim za budući razvoj tehnologije i društva [15].

4.1 Baza podataka web aplikacije

Baza podataka se sastoji od devet tablica: „*users*“, „*odjel*“, „*kolegiji*“, „*pojedinne_rezervacije*“, „*rezervacije_tajnica*“, „*recepcija*“, „*studiji*“, „*dvorane*“ i „*profesori*“. Dijagram baze podataka je prikazan ispod (*Slika 17.*). S obzirom na to da se ovaj rad primarno fokusira na front-end aspekte razvoja sustava, detaljan tehnički opis same

4.3 Dijagram slučajeva korištenja

Dijagram slučajeva korištenja (Slika 19.) ilustrira interakcije između korisnika i sistema, prikazujući sve ključne funkcionalnosti koje sistem pruža. Ovaj dijagram pomaže u identifikaciji i analizi različitih korisničkih zahtjeva i scenarija upotrebe.



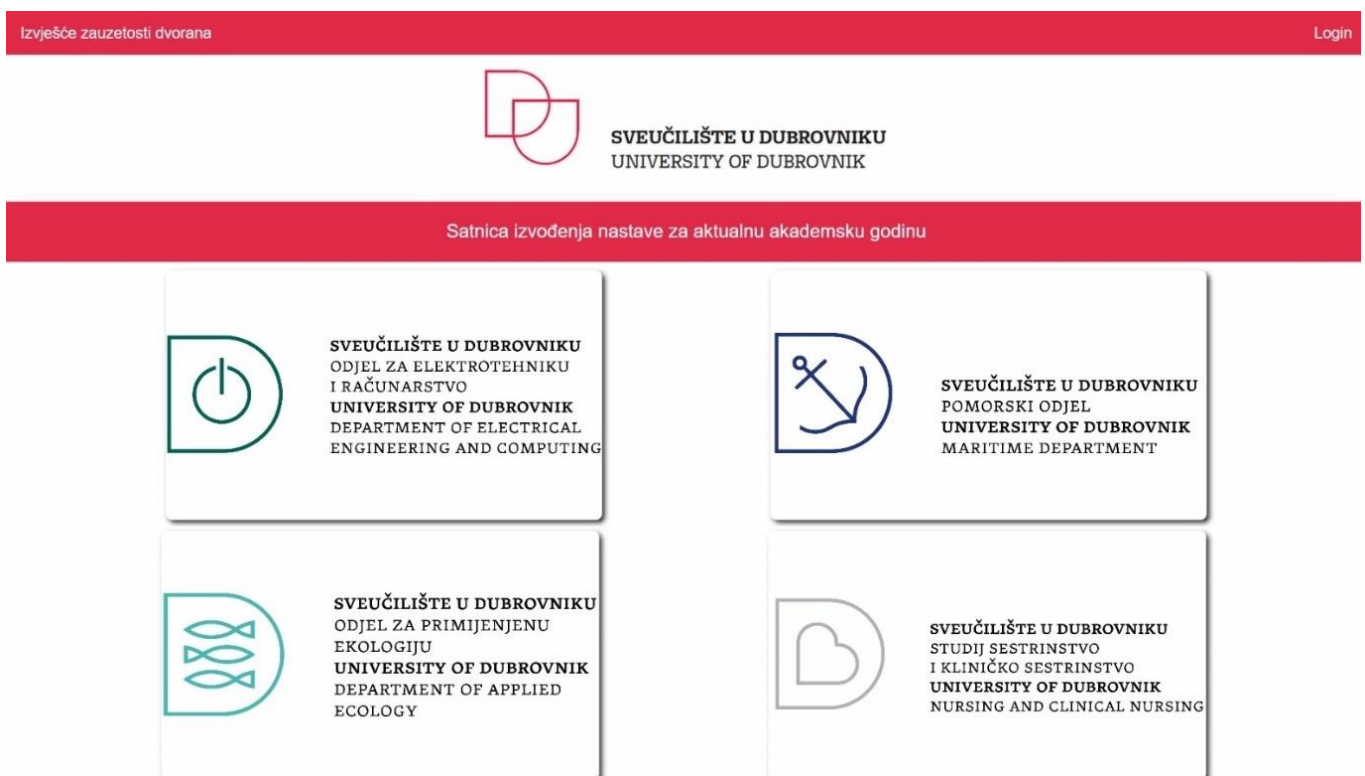
Slika 19. Dijagram slučajeva korištenja

5. IMPLEMENTACIJA WEB SJEDIŠTA I SUSTAVA ZA UPRAVLJANJE SADRŽAJEM

Za razvoj prezentacijskom sloja web sjedišta korišteni su HTML, CSS i JavaScript. Svi elementi su dizajnirani i implementirani od početka, bez template-a, koristeći „vanilla“ CSS i JavaScript, čime je omogućena potpuna kontrola nad izgledom i funkcionalnošću front-end aplikacije.

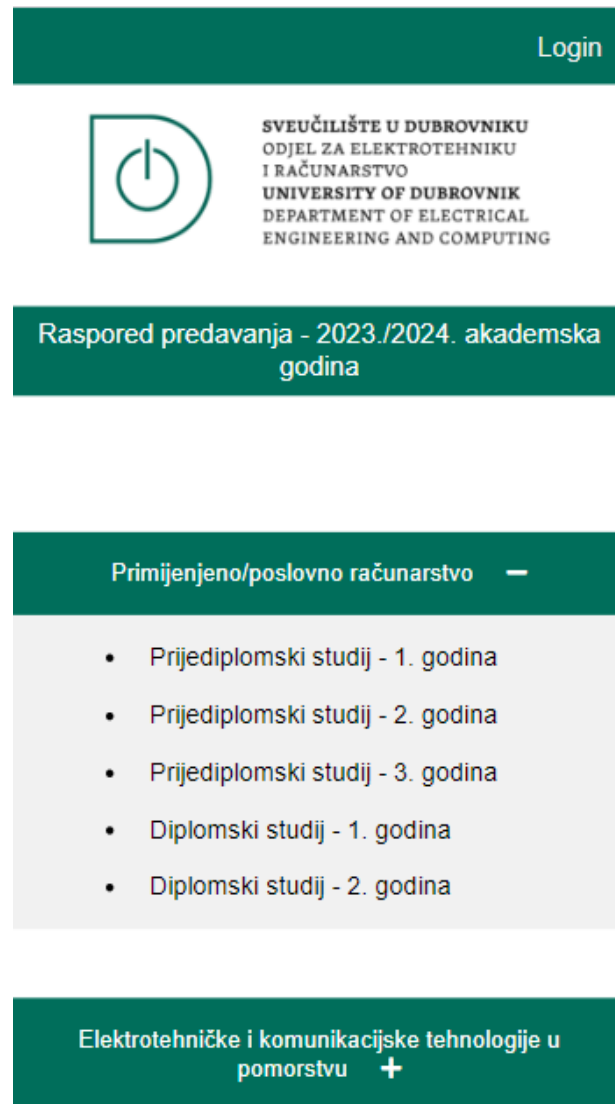
5.1 Naslovna stranica

Aplikacija obuhvaća naslovnu stranicu i niz podstranica. U nastavku će biti obrađen dio aplikacije koji je dostupan svim korisnicima bez potrebe za prijavom. Naslovna stranica omogućuje izbor između četiri odjela na visokom učilištu: Odjel za elektrotehniku i računarstvo, Pomorski odjel, Odjel za primijenjenu ekologiju i Studij sestrinstvo i kliničko sestrinstvo. Također prikazuje izvješće o zauzetosti dvorana te sadrži formu za prijavu u sustav, koja je dostupna samo zaposlenicima visokog učilišta. Aplikacija je razvijena koristeći Sveučilište u Dubrovniku kao primjer, što je prikazano na „Slika 20.“.



Slika 20. Naslovna stranica

Nakon odabira pojedinog odjela, prikazuje se podstranica koja nudi odabir smjera, studija te godine. Aplikacija je napravljena u prilagodljivom dizajnu, te „Slika 21.“ prikazuje mobilnu verziju aplikacije.



Slika 21. Odabir studija i godine u mobilnoj verziji aplikacije

Prilikom odabira smjera, otvara se izbornik za izbor studija i godine, također mijenja se ikona iz „+“ u „-“ što korisnicima omogućuje interakciju s aplikacijom. JavaScript kod za prikaz izbornika i promjenu ikona prikazuje „Slika 22.“

```
<script>
var coll = document.getElementsByClassName("collapsible");
var i;

for (i = 0; i < coll.length; i++) {
  coll[i].addEventListener("click", function() {
    this.classList.toggle("active");
    var content = this.nextElementSibling;
    if (content.style.display === "block") {
      content.style.display = "none";
    } else {
      content.style.display = "block";
    }
  });
}

// Dohvaćanje svih gumba s klasom "collapsible"
var colls = document.querySelectorAll(".collapsible");

// Iteriranje kroz sve gumbе i dodavanje event listenera
colls.forEach(function(coll) {
  coll.addEventListener("click", function() {
    // Dodavanje ili uklanjanje klase "active" prilikom klika
    this.classList.toggle("active");

    // Dohvaćanje ikone za trenutni gumb
    var icon = this.querySelector(".icon i");

    // Toggle između ikona plus i minus
    icon.classList.toggle("fa-plus");
    icon.classList.toggle("fa-minus");
  });
});
</script>
```

Slika 22. JavaScript kod za prikaz izbornika

Nakon odabira studija i godine prikazuje se raspored predavanja s mogućnošću odabira datuma, što prikazuje „Slika 23.“. Naslov i semestar su automatski generirani kroz aplikaciju.

Login

SVEUČILIŠTE U DUBROVNIKU
ODJEL ZA ELEKTROTEHNIKU
I RAČUNARSTVO
UNIVERSITY OF DUBROVNIK
DEPARTMENT OF ELECTRICAL
ENGINEERING AND COMPUTING

Diplomski studij - Primijenjeno/poslovno računarstvo - 1. godina
Zimski semestar

13. 11. 2024.

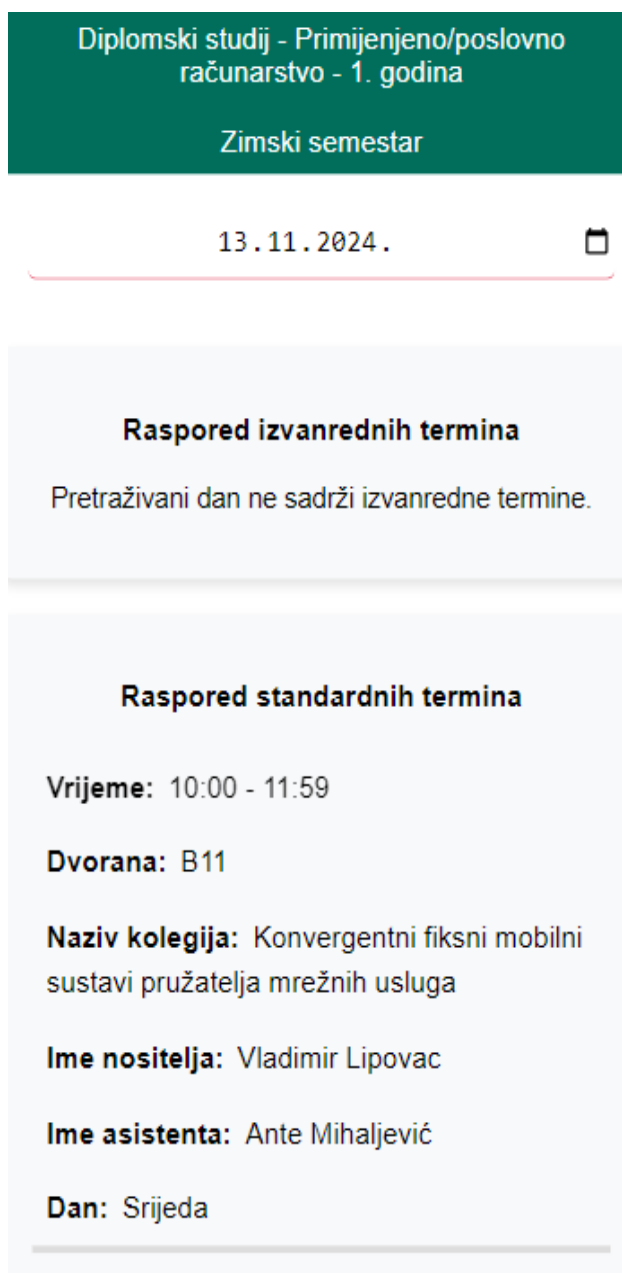
Raspored izvanrednih termina
Pretraživani dan ne sadrži izvanredne termine.

Raspored standardnih termina

Vrijeme	Dvorana	Naziv kolegija	Ime nositelja	Ime asistenta	Dan
10:00 - 11:59	B11	Konvergentni fiksni mobilni sustavi pružatelja mrežnih usluga	Vladimir Lipovac	Ante Mihaljević	Srijeda
12:00 - 13:59	B11	Upravljanje intelektualnim vlasništvom	Nebojša Stojčić	Goran Vučur	Srijeda
14:00 - 17:59	E02	Ekspertni sustavi	Ivona Zakarija	Toni Besjedica	Srijeda

Slika 23. Prikaz rasporeda

Aplikacija je potpuno prilagođena prikazu u mobilnoj verziji. Dio takve je podstranice prikazan je na „Slika 24.“



Slika 24. Prikaz rasporeda u mobilnoj verziji

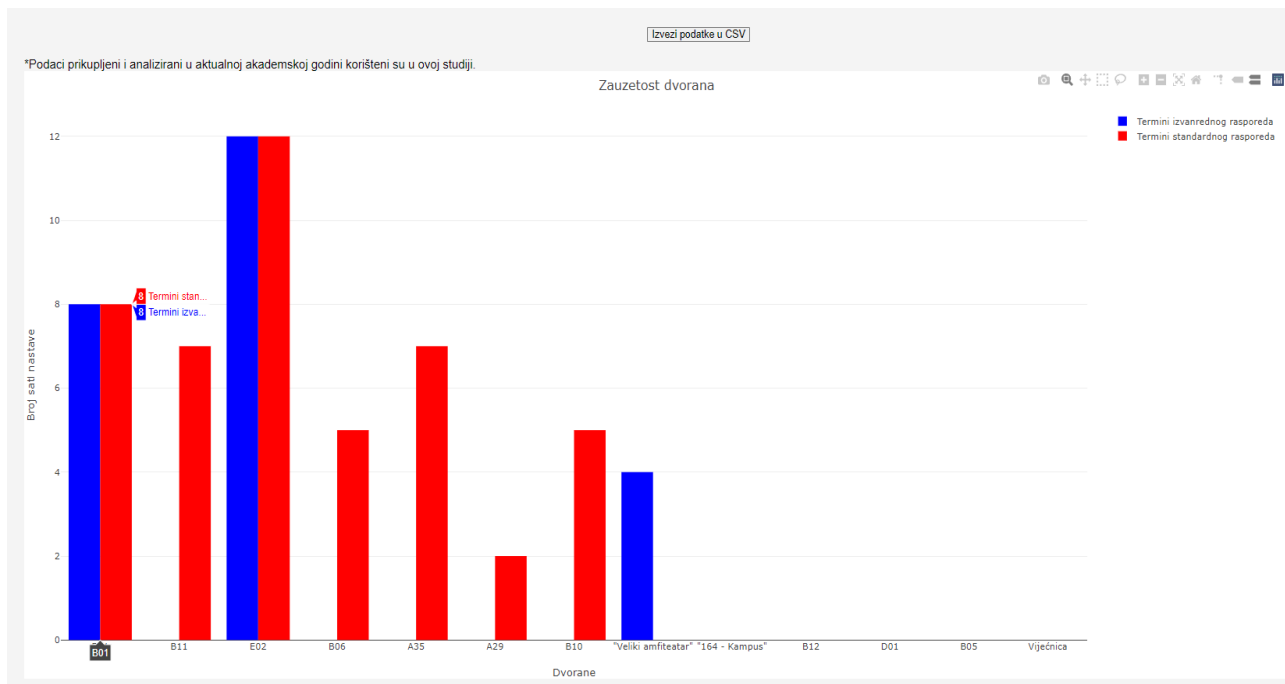
Dio CSS koda za responzivnost prikaza rasporeda predavanja u mobilnoj verziji prikazan je na „Slika 25.“

```
@media only screen and (max-width: 480px) {  
  /* _____ Ispis rasporeda _____ */  
  /* _____ */  
  
  .naslovRaspored{  
    font-size: 2vh;  
  }  
  .tableProfesori thead {  
    display: none;  
  }  
}  
  
.tableProfesori tbody, .tableProfesori tr, .tableProfesori th, .tableProfesori td {  
  display: block;  
}  
  
.tableProfesori tr {  
  margin-bottom: 2vh; /* Dodajte razmak između svakog "stupca" */  
  border-bottom: 0.5vh solid #ddd; /* Dodajte granicu između "stupaca" */  
}  
  
.tableProfesori th:first-child, .tableProfesori td:first-child {  
  position: sticky;  
  left: 0;  
  z-index: 1;  
  width: 100%; /* Prilagodite širinu prvog fiksnog stupca prema potrebi */  
}  
  
.tableProfesori th, .tableProfesori td {  
  font-size: 2vh;  
  padding: 1vh; /* Dodajte razmak unutar ćelija */  
  text-align: left;  
  border-bottom: none;  
  white-space: normal;  
  line-height: 1.5;  
}  
  
.tableProfesori td::before {  
  content: attr(data-label);  
  font-weight: bold;  
  margin-right: 1vh; /* Dodajte razmak između "oznake" i teksta */  
}  
}
```

Slika 25. Dio CSS koda za responzivnost

5.2 Izvješće zauzetosti dvorana

Izvješće o zauzetosti dvorana je još jedna od opcija na naslovnoj stranici. Prikazuje zauzetost u omjeru broja termina među dvoranama na visokom učilištu. Graf je interaktivan s mnoštvom opcija u izborniku. Također je omogućena opcija izvoza podataka, odabira tipke na vrhu podstranice, u CSV formatu. Graf s pripadajućim opcijama prikazan je na „Slika 26.“



Slika 26. Izvješće zauzetosti dvorana

JavaScript je zadužen za ispis grafa u željenom formatu. Implementirani kod je prikazan na „Slika 27.“

```
<script>
function drawGraph(x, y1, y2) {
  const trace1 = {
    x: x,
    y: y1,
    type: 'bar',
    //pojedine rezervacije
    name: 'Termini izvanrednog rasporeda',
    marker: { color: 'blue' }
  };

  const trace2 = {
    x: x,
    y: y2,
    type: 'bar',
    //rezervacije tajnica
    name: 'Termini standardnog rasporeda',
    marker: { color: 'red' }
  };

  const data = [trace1, trace2];

  const layout = {
    title: 'Zauzetost dvorana',
    xaxis: { title: 'Dvorane' },
    yaxis: { title: 'Broj sati nastave' },
    barmode: 'group', // Grupiranje grafova
    margin: { t: 50, r: 50, b: 50, l: 50 }
  };

  Plotly.newPlot('graph', data, layout);
}

fetch('TableauReport.csv')
  .then(response => response.text())
  .then(csv => {
    const rows = csv.split('\n').slice(1); // Izbaci prvi redak (header)
    const x = [];
    const y1 = []; // Pojedine rezervacije
    const y2 = []; // Rezervacije tajnica/ostali termini

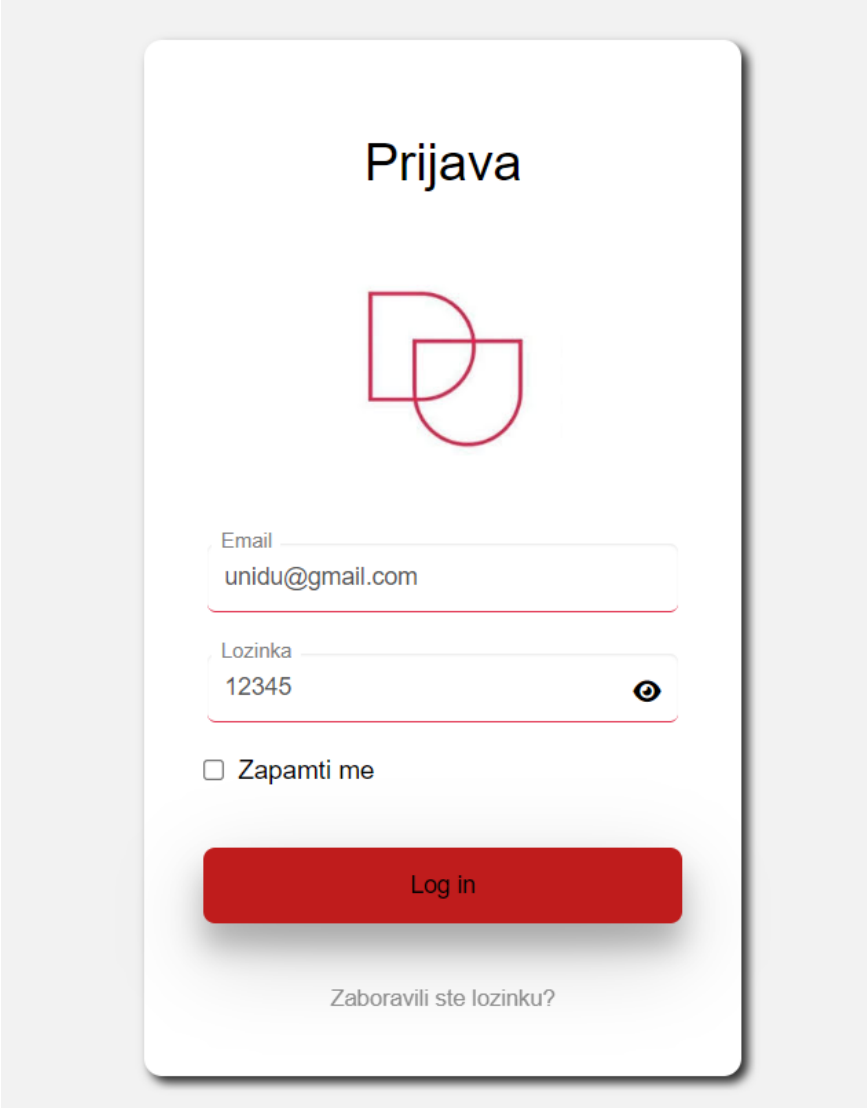
    rows.forEach(row => {
      const cols = row.split(';');
      if (cols.length >= 4) {
        x.push(cols[0].trim()); // Dodajemo ime dvorane na x-os
        y1.push(parseInt(cols[2].trim(), 10)); // Dodajemo broj pojedinih rezervacija
        y2.push(parseInt(cols[3].trim(), 10)); // Dodajemo broj rezervacija tajnica/ostalih termina
      }
    });

    drawGraph(x, y1, y2);
  })
  .catch(error => {
    document.getElementById('graph').innerHTML = '<p>Error loading graph data</p>';
    console.error('Error fetching or parsing CSV:', error);
  });
}</script>
```

Slika 27. JavaScript kod za prikaz grafa

5.3 Login forma

Forma za prijavu služi zaposlenicima visokih učilišta za autentifikaciju u sustav za upravljanje sadržajem web aplikacije. Nakon uspješne prijave, korisnici će, ovisno o razini pristupa, biti preusmjereni na odgovarajuću stranicu za upravljanje sadržajem. U slučaju neuspješne prijave, korisnik će dobiti obavijest s odgovarajućom porukom o grešci. Ispunjena forma za autentifikaciju korisnika u sustav prikazana je na „Slika 28.“ Implementirana je opcija za prikaz lozinke, što je i prikazano na slici, kako bi se korisnicima omogućila jednostavnija autentifikacija. Također, lozinku je moguće i sakriti kako bi se sačuvala privatnost i povećala sigurnost. Animacija prelaska mišem preko „Log in“ gumba sadržana je u fotografiji.

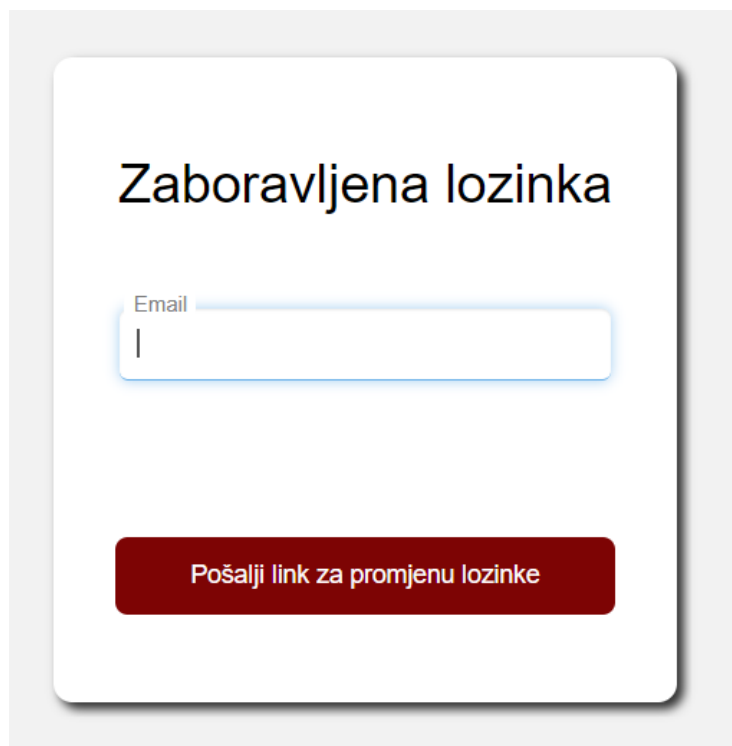


The image shows a login form with the following elements:

- Title:** "Prijava" (Login) in a large, bold, black font.
- Logo:** A red graphic consisting of two overlapping rounded rectangular shapes.
- Email Field:** A white input field with a red border, containing the text "unidu@gmail.com". The label "Email" is positioned above the field.
- Password Field:** A white input field with a red border, containing the text "12345". The label "Lozinka" is positioned above the field. To the right of the field is a red eye icon for toggling password visibility.
- Remember Me:** A checkbox labeled "Zapamti me" (Remember me).
- Login Button:** A prominent red button with the text "Log in" in white.
- Forgot Password:** A link labeled "Zaboravili ste lozinku?" (Forgot your password?) located below the login button.

Slika 28. Forma za prijavu

Opcija „Zapamti me“ također je sastavni dio forme za prijavu. Ukoliko je odabrana te je prijava uspješna pamti podatke prijave određen vremenski period. Integrirana je i opcija „Zaboravili ste lozinku?“ koja služi za oporavak lozinke ukoliko ju je korisnik zaboravio. Nakon odabira navedene opcije, pod uvjetom da se unesena e-mail adresa nalazi u bazi korisnika, tj. da je korisnik već dio sustava, prikazuje se forma za unos e-mail adrese na koju se šalje link za oporavak lozinke. „Slika 29.“ prikazuje formu za unos e-mail adrese. Potom je potrebno, preko zadanog linka, upisati novu lozinku koja se potom koristi za prijavu u sustav. Sustav korisnicima dinamički pruža obavijesti o uspjehu ili greškama tijekom cijelog procesa.



Slika 29. Forma za unos e-mail adrese

Radi ljepšeg vizualnog prikaza i jednostavnije interakcije korisnika sa sustavom, implementirane su CSS animacije koje poboljšavaju korisničko iskustvo, čineći sučelje intuitivnijim i dinamičnijim. Animacija aktivnog stanja korištena je u formi za unos e-mail adrese na „Slika 29.“. Dio CSS koda zadužen za navedenu animaciju nalazi se na „Slika 30.“.

```
/*Email i password*/
.form-group{
  position: relative;
  margin: 20px 0;
  padding: 0.3vh;
}

/*Email i password - Labela sa nazivom*/
.form-group label{
  position: absolute;
  top: 50%;
  left: 5px;
  transform: translateY(-50%);
  font-size: 16px;
  color: #grey;
  padding: 0 5px;
  pointer-events: none;
  transition: top 200ms ease-in,
  left 200ms ease-in,
  font-size 200ms ease-in;
  font-weight: normal;
  cursor: text;
}

/*Email i password input za unos*/
.form-group input{
  width: 100%;
  height: 40px;
  font-size: 14px;
  padding: 0 10px;
  border-bottom: 1.2px solid #e02a48;
  outline: none;
  border-radius: 5px;
  border-left: none;
  border-right: none;
  border-top: none;
}

/*Animacija za Email i password*/
.form-group input:active + label,
.form-group input:focus + label,
.form-group input:valid + label{
  top: 0;
  font-size: 12px;
  background: #fff;
}
```

Slika 30. Dio CSS koda za animaciju unosa e-mail adrese

5.4 Sustav za upravljanje sadržajem – Puni pristup

Nakon uspješne prijave u aplikaciju, tajnice visokih učilišta dobivaju puni pristup koji im omogućuje administriranje zadataka te upravljanje rasporedom predavanja. Sustav za upravljanje sadržajem zamišljen je kao interaktivni dio aplikacije s naglaskom na intuitivne opcije. Tajnica može upravljati podacima vezanim za profesore, dvorane, kolegije, standardne rasporede, izvanredne termine te zaposlenike recepcije.

Administratori dodaju tajnice u sustav, pri čemu svaka tajnica može upravljati podacima samo za svoj odjel ili smjer. One imaju ovlasti dodavati, brisati i uređivati podatke o profesorima, čime mogu dodijeliti ili oduzeti profesorima i zaposlenicima recepcije pristup aplikaciji. Tajnica također može upravljati podacima o dvoranama i kolegijima te vršiti izmjene u vezi s rezervacijama standardnog rasporeda i izvanrednih termina, ovisno o predviđenom rasporedu predavanja, ispita i kolokvija.

Sustav uključuje opciju upravljanja profilom, gdje korisnik može pristupiti funkcijama kao što su povratak na naslovnu stranicu, promjena lozinke i odjava iz aplikacije. Na „Slika 31.“ prikazana je upravljačka ploča sa svim dostupnim funkcijama.



Slika 31. Sustav za upravljanje sadržajem – Puni pristup

Sljedeći dio sustava za upravljanje sadržajem detaljno opisuje administraciju podataka o kolegijima i rezervacijama izvanrednih termina. Klikom na opciju za upravljanje podacima o kolegijima, otvara se zasebna stranica s popisom svih kolegija koje tajnica može administrirati (*Slika 32.*). U ovom primjeru korišteni su podaci s Odjela za elektrotehniku i računarstvo Sveučilišta u Dubrovniku. Tajnice mogu dodavati nove kolegije, brisati postojeće ili mijenjati njihove podatke. Dodatno, sustav uključuje pretraživanje kako bi se lakše upravljalo podacima te opciju brzog povratka na glavnu upravljačku ploču. Podaci su abecedno poredani radi lakšeg pregleda i navigacije kroz popis.

Kolegiji

Dodaj kolegij

Natrag

Naziv kolegija	Ime nositelja	Ime asistenta		
Arhitektura mreža nove generacije	Adriana Lipovac Vrhovac	Adriana Lipovac Vrhovac		
Brodске električne mreže	Marija Mirošević	Marija Mirošević		
Ekspertni sustavi	Ivona Zakarija	Toni Besjedica		
Elektromagnetska kompatibilnost	Nikša Burum	Nikša Burum		
Engleski jezik	Martina Hrnić	Martina Hrnić		
Ergonomija računalne opreme	Željka Car	Ana Kešelj Dilberović		
Informacijski sustavi u turizmu	Mario Miličević	Periša Zakarija		
Konvergentni fiksni mobilni sustavi pružatelja mrežnih usluga	Vladimir Lipovac	Ante Mihaljević		
Matematika IV	Martin Lazar	Mara Vlašić		
Metodologija znanstvenoistraživačkog rada	Adriana Lipovac Vrhovac	Anamaria Bjelopera		

Slika 32. Upravljanje podacima o kolegijima

Korištene su intuitivne boje ikona i gumbova za određenje radnje. Za gumb „Dodaj kolegij“ i ikonu za uređivanje podataka o pojedinom kolegiju korištena je nijansa zelene boje, dok je za ikonu brisanja podataka o pojedinom kolegiju i gumb „Natrag“ korištena nijansa crvene boje.

U slučaju ažuriranja podataka o prethodno unesenom kolegiju potrebno je odabrati ikonu za uređivanje. Nakon toga prikazuje se forma za promjenu podataka odabranog kolegija, koja je već ispunjena podacima od kojih je moguće promijeniti samo jedan podatak ili njih sve. Prikazat će se poruka o uspješnosti promjene, odnosno pogrešci tijekom procesa. „Slika 33.“ prikazuje mobilnu verziju forme za ažuriranje prethodno unesenog kolegija.

Uredi kolegij

Natrag

Naziv kolegija:

Nositelj kolegija:

Asistent kolegija:

Uredi kolegij

Slika 33. Forma za ažuriranje kolegija

Kada je potrebno upravljati rezervacijama izvanrednih termina, na upravljačkoj ploči treba odabrati opciju „*Izvanredni termini*“. Nakon toga otvara se podstranica sustava za upravljanje sadržajem koja je namijenjena upravljanju rezervacijama izvanrednih termina. Na toj stranici prikazan je popis svih termina odjela koji pripadaju tajnici, sortirani prema datumu i vremenu, a termini se mogu brisati ili ažurirati. Također su dostupne opcije za unos nove rezervacije, promjenu prikaza semestra, pretraživanje rezervacija po svim dostupnim podacima te povratak na prethodnu stranicu. Na „*Slika 34.*“ prikazan je dio sustava koji služi za upravljanje rezervacijama izvanrednih termina.

Izvanredni termini

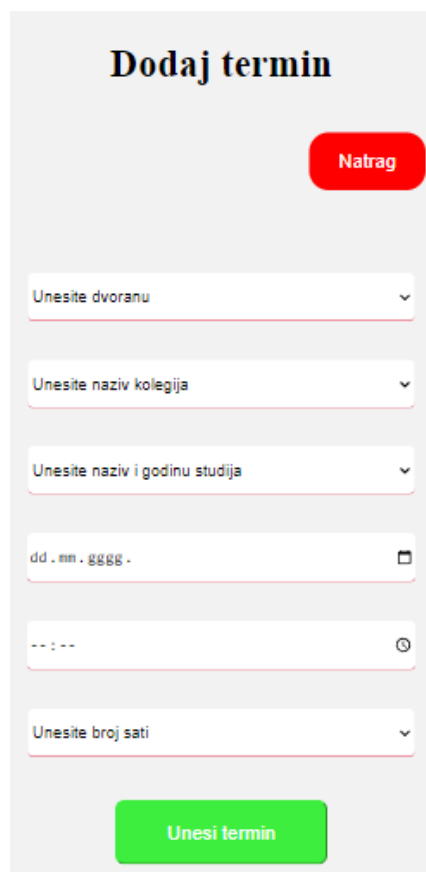
Natrag

Dodaj termin

Datum	Vrijeme	Dvorana	Naziv kolegija	Naziv studija		
Petak, 4. Listopad 2024	10:00 - 13:59	B01	Ergonomija računalne opreme	Diplomski studij - Primijenjeno/poslovno računarstvo - 1. godina		
Subota, 5. Listopad 2024	10:00 - 13:59	B01	Ergonomija računalne opreme	Diplomski studij - Primijenjeno/poslovno računarstvo - 1. godina		
Petak, 11. Listopad 2024	08:00 - 13:59	E02	Projektiranje informacijskih sustava	Diplomski studij - Primijenjeno/poslovno računarstvo - 1. godina		
Subota, 12. Listopad 2024	08:00 - 13:59	E02	Projektiranje informacijskih sustava	Diplomski studij - Primijenjeno/poslovno računarstvo - 1. godina		
Petak, 25. Listopad 2024	08:00 - 13:59	Veliki amfiteatar	Projektiranje informacijskih sustava	Diplomski studij - Primijenjeno/poslovno računarstvo - 1. godina		

Slika 34. Upravljanje podacima o rezervacijama izvanrednih termina

Ukoliko je potrebno unijeti podatke o novoj rezervaciji, nakon odabira opcije „Dodaj termin“, prikazuje se forma za unos podataka. Potrebno je unijeti dvoranu, naziv kolegija, naziv i godinu studija, datum, početak održavanja termina te broj sati trajanja termina. Nije moguće obaviti unos ukoliko je jedan od navedenih podataka prazan. Sustav će u tom slučaju obavijestiti korisnika, kao i u ostalim slučajevima nepravilnog unosa ili pak potvrditi uspješan unos za to predviđenom obavijesti. Omogućena je i opcija povratka na prethodnu stranicu. „Slika 35.“ prikazuje mobilnu verziju forme za unos podataka o novom terminu.



The image shows a mobile application form titled "Dodaj termin". At the top right, there is a red button labeled "Natrag". Below the title, there are several input fields: a dropdown menu for "Unesite dvoranu", a dropdown menu for "Unesite naziv kolegija", a dropdown menu for "Unesite naziv i godinu studija", a date input field with the placeholder "dd . mm . gggg .", a time input field with the placeholder "--- : ---", and a dropdown menu for "Unesite broj sati". At the bottom of the form, there is a green button labeled "Unesi termin".

Slika 35. Mobilna verzija forme za unos novog termina

5.5 Sustav za upravljanje sadržajem – Srednji pristup

Nakon što se uspješno prijave u aplikaciju, profesori i zaposlenici recepcije visokih učilišta dobivaju srednju razinu pristupa. Sustav za upravljanje sadržajem osmišljen je kao interaktivni segment aplikacije, s naglaskom na jednostavne i intuitivne opcije. Ovim korisnicima omogućeno je provjeravanje dostupnosti termina, dodavanje novih termina te pregled nadolazećih obaveza.

Također, postoji funkcionalnost za upravljanje korisničkim profilom, koja uključuje opcije povratka na početnu stranicu, promjene lozinke i odjave iz sustava. Na „Slika 36.“ prikazana je mobilna verzija upravljačke ploče sa svim dostupnim opcijama.



Slika 36. Mobilna verzija upravljačke ploče sustava za upravljanje sadržajem sa srednjim pristupom

U nastavku će se detaljno opisati opcije „Provjeri dostupnost“ te „Pregled nadolazećih termina“, dok opcija „Dodaj termin“ prikazuje formu za unos novog termina. Zadaća opcije „Pregled nadolazećih termina“ je upravljanje vlastitim terminima ukoliko za tim bude potrebe. Opcija "Provjeri dostupnost" omogućuje korisniku da pregleda zauzetost željene dvorane za odabrani datum, kako bi prije unosa termina provjerio koji su termini slobodni te unio rezervaciju u odgovarajuće vrijeme. Nakon odabira provjere dostupnosti prikazuje se forma u koju je potrebno unijeti naziv dvorane i datum te tako provjeriti dostupnost. Integrirana je opcija za povratak na upravljačku ploču. „Slika 37.“ prikazuje mobilnu verziju dijela sustava za upravljanje sadržajem namijenjen provjeri dostupnosti termina.

Dostupnost termina

[Natrag](#)

Unesite dvoranu

dd . mm . gggg .

[Provjeri dostupnost](#)

Raspored za Računalni lab E02 za datum
11.10.2024.

Nema zauzetih termina stalnog rasporeda

Zauzeti termini izvanrednih rezervacija

Datum	Dvorana	Vrijeme
Petak, 11. Listopad 2024	E02	08:00 - 13:59

Slika 37. Mobilna verzija za provjeru dostupnosti termina

U svrhu lakšeg upravljanja rezervacijama izvanrednih termina, ispis rezervacija treba biti pregledan i lako čitljiv. Na „Slika 38.“ prikazan je dio CSS koda zadužen za ispis rezervacija.

```

/*-----NOVI CSS DOSTUPNE TERMINE-----*/
/*Na middlu na profesorima - dostupost termina naslovi*/
.naslovDostupnihTermina{
  font-size: 20px;
  font-weight: bold;
  text-align: center;
}

/*Na middlu na profesorima - dostupost termina - div za ispis rezultata*/
.rezultat {
  display: none;
  margin: 0;
  padding: 4vh;
  border-radius: 0;
  background-color: #f8f9fa;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
  width: 75%;
}

/*Na middlu na profesorima - dostupost termina ispis naslova*/
.rezultat p {
  margin: 0;
}

/*Na middlu na profesorima - dostupost termina tablica*/
.rezultat table {
  width: 100%;
  border-collapse: collapse;
  margin-top: 15px;
}

/*Na middlu na profesorima - dostupost termina redci tablice*/
.rezultat th, .rezultat td {
  padding: 10px;
  text-align: center;
  border-bottom: 1px solid #ddd;
  width: calc(100% / 3); /* Postavljanje fiksne širine na 1/3 širine tablice za svaku kolonu */
}

.rezultat th {
  background-color: #f0f0f0;
}

.rezultat tr:nth-child(even) {
  background-color: #fafafa;
}

.rezultat tr:hover {
  background-color: #e9ecef;
}

```

Slika 38. Dio CSS koda za ispis rezervacija

Nakon odabira opcije „Pregled nadolazećih termina“ prikazuje se sučelje s popisom svih unesenih rezervacija korisnika koji je prijavljen u sustav. Omogućeno je uređivanje i brisanje pojedine rezervacije, kao i promjena semestra. Također je integrirana opcija za povratak na upravljačku ploču. „Slika 39.“ prikazuje mobilnu verziju sučelja za upravljanje nadolazećih rezervacija izvanrednih termina.

Pregled nadolazećih termina

Natrag

Zimski semestar


Datum: Petak, 25. Listopad 2024


Vrijeme: 08:00 - 13:59

Dvorana: Veliki amfiteatar

Smjer: Diplomski studij - Primijenjeno/poslovno računarstvo - 1. godina

Naziv kolegija: Projektiranje informacijskih sustava

Izbriši 

Uredi 

Slika 39. Mobilna verzija sučelja za pregled nadolazećih termina

6. ZAKLJUČAK

U ovom diplomskom radu predstavljen je sustav za online upravljanje rezervacijama dvorana na visokim učilištima, s posebnim naglaskom na prezentacijski i poslovni sloj. Kroz detaljno planiranje, specifikaciju zahtjeva te primjenu suvremenih tehnologija, cilj je bio razviti cjelovito rješenje koje će optimizirati upravljanje prostornim resursima obrazovnih institucija.

Prvi korak u realizaciji projekta uključivao je temeljitu analizu potreba krajnjih korisnika, od nastavnog osoblja do studenata, što je rezultiralo jasno definiranim funkcionalnim i nefunkcionalnim zahtjevima sustava. Modeliranje procesa, uključujući dijagrame dekompozicije funkcija i dijagrame procesa, omogućilo je precizno definiranje radnih tijekova unutar sustava, osiguravajući njegovu funkcionalnost i skalabilnost.

U tehničkom dijelu implementacije, korištenje standardnih web tehnologija kao što su HTML, CSS i JavaScript, uz podršku WampServer okruženja, omogućilo je izradu interaktivnog i prilagodljivog korisničkog sučelja. Dodatno, izgradnja sustava za upravljanje sadržajem s različitim razinama pristupa korisnicima (puni i srednji pristup) omogućila je višenamjensku upotrebu platforme, olakšavajući administraciju i povećavajući transparentnost u organizaciji akademskih resursa. Integracija alata poput Tableau-a za izvještavanje dodatno je unaprijedila sustav omogućujući detaljnu analizu zauzetosti dvorana te precizno praćenje korištenja kapaciteta.

Razvijeni sustav značajno pojednostavljuje administrativne procese, smanjuje mogućnost ljudske pogreške i doprinosi boljoj organizaciji akademskih aktivnosti. Njegova modularna arhitektura omogućuje daljnju nadogradnju i prilagodbu specifičnim potrebama visokih učilišta.

Zaključno, izgradnja ovog sustava ne samo da rješava trenutačne izazove u upravljanju rezervacijama dvorana, već postavlja čvrste temelje za buduća tehnološka poboljšanja u obrazovnim institucijama. Uvođenjem naprednih funkcionalnosti i daljnjom integracijom s već postojećim akademskim sustavima, ovaj projekt može postati ključni element digitalne transformacije visokih učilišta, pridonoseći njihovoj učinkovitosti, fleksibilnosti i modernizaciji u skladu s potrebama današnjeg obrazovnog sustava.

7. LITERATURA

- [1] »W3Schools,« HTML History, [Mrežno]. Available: <https://www.w3schools.in/html/history#:~:text=HTML%20was%20created%20by%20Sir,evolved%2>. [Pokušaj pristupa 15. 07. 2024.].
- [2] »Hostinger tutorials,« What Is HTML? Hypertext Markup Language Basics Explained, 25. 08. 2023.. [Mrežno]. Available: https://www.hostinger.com/tutorials/what-is-html#Most_Used_HTML_Tags_and_HTML_Elements. [Pokušaj pristupa 15. 07. 2024.].
- [3] »Tutorialspoint,« CSS - Introduction, [Mrežno]. Available: https://www.tutorialspoint.com/css/what_is_css.htm#. [Pokušaj pristupa 18. 07. 2024.].
- [4] »W3Schools,« CSS Introduction, [Mrežno]. Available: https://www.w3schools.com/css/css_intro.asp. [Pokušaj pristupa 18. 07. 2024.].
- [5] »What is CSS - javatpoint,« [Mrežno]. Available: <https://www.javatpoint.com/what-is-css>. [Pokušaj pristupa 18. 07. 2024.].
- [6] D. Crockford, »JavaScript: The Good Parts,« [Mrežno]. [Pokušaj pristupa 26. 07. 2024.].
- [7] »LaunchSchool,« A Brief JavaScript History, [Mrežno]. Available: <https://launchschool.com/books/javascript/read/introduction>. [Pokušaj pristupa 25. 07. 2024.].
- [8] D. Flanagan, »JavaScript: The Definitive Guide,« [Mrežno]. [Pokušaj pristupa 26. 07. 2024.].
- [9] »FreeCodeCamp,« What is JavaScript? A Definition of the JS Programming Language, [Mrežno]. Available: <https://www.freecodecamp.org/news/what-is-javascript-definition-of-js/>. [Pokušaj pristupa 25. 07. 2024.].
- [10] E. T. F. Elisabeth Robson, »Head First JavaScript Programming,« [Mrežno]. [Pokušaj pristupa 26. 07. 2024.].
- [11] »phoenixNAP - What Is WAMP?,« [Mrežno]. Available: <https://phoenixnap.com/glossary/what-is-wamp>. [Pokušaj pristupa 01. 08. 2024.].
- [12] E. Y., »Hostinger tutorials - What Is WAMP – a Friendly Guide for Beginners,« [Mrežno]. Available: <https://www.hostinger.com/tutorials/what-is-wamp>. [Pokušaj pristupa 01. 08. 2024.].
- [13] »Tableau - What is Tableau?,« [Mrežno]. Available: <https://www.tableau.com/why-tableau/what-is-tableau>. [Pokušaj pristupa 06. 08. 2024.].
- [14] »Tableau,« [Mrežno]. Available: <https://www.tableau.com/>. [Pokušaj pristupa 06. 08. 2024.].
- [15] »GeeksForGeeks - What is Database?,« [Mrežno]. Available: <https://www.geeksforgeeks.org/what-is-database/>. [Pokušaj pristupa 18. 08. 2024.].

8. PRILOZI

8.1 Popis tablica

Tablica 1. Funkcionalni zahtjevi web aplikacije	4
Tablica 2. Nefunkcionalni zahtjevi web aplikacije	6

8.2 Popis slika

Slika 1. Dijagram dekompozicije funkcija	8
Slika 2. Dijagram konteksta (0. razina)	10
Slika 3. Dijagram glavnih procesa (1. razina).....	11
Slika 4. Dijagram za odabrani proces (2. razina).....	12
Slika 5. Početna deklaracija	14
Slika 6. Paragraf u HTML kodu	15
Slika 7. HTML struktura	15
Slika 8. Osnovna sintaksa CSS-a.....	17
Slika 9. Referenciranje CSS datoteke	17
Slika 10. Primjer unutarnjeg CSS-a.....	18
Slika 11. ID selektor	18
Slika 12. Ugrađeni CSS	19
Slika 13. Ugrađeni JavaScript.....	21
Slika 14. Unutarnji JavaScript	21
Slika 15. Uključivanje vanjske JavaScript datoteke	22
Slika 16. Izgled ekrana za pristup WampServer-u	24
Slika 17. Dijagram baze podataka	28
Slika 18. Korisnički dijagram aktivnosti	29
Slika 19. Dijagram slučajeva korištenja	30
Slika 20. Naslovna stranica.....	31
Slika 21. Odabir studija i godine u mobilnoj verziji aplikacije	32
Slika 22. JavaScript kod za prikaz izbornika	33
Slika 23. Prikaz rasporeda	34
Slika 24. Prikaz rasporeda u mobilnoj verziji.....	35
Slika 25. Dio CSS koda za responzivnost.....	36
Slika 26. Izvješće zauzetosti dvorana	37
Slika 27. JavaScript kod za prikaz grafa.....	38
Slika 28. Forma za prijavu	39
Slika 29. Forma za unos e-mail adrese	40
Slika 30. Dio CSS koda za animaciju unosa e-mail adrese	41
Slika 31. Sustav za upravljanje sadržajem – Puni pristup	42
Slika 32. Upravljanje podacima o kolegijima.....	43
Slika 33. Forma za ažuriranje kolegija	44
Slika 34. Upravljanje podacima o rezervacijama izvanrednih termina	44
Slika 35. Mobilna verzija forme za unos novog termina.....	45
Slika 36. Mobilna verzija upravljačke ploče sustava za upravljanje sadržajem sa srednjim pristupom.....	46

Slika 37. Mobilna verzija za provjeru dostupnosti termina	47
Slika 38. Dio CSS koda za ispis rezervacija	48
Slika 39. Mobilna verzija sučelja za pregled nadolazećih termina	49

IZJAVA

Izjavljujem pod punom moralnom odgovornošću da sam diplomski rad izradila samostalno, temeljem znanja stečenog u dosadašnjem obrazovanju, posebice onom na studijima Odjela za elektrotehniku i računarstvo, služeći se navedenim izvorima podataka i uz stručno vodstvo mentorice doc.dr.sc. Ivone Zakarije, te komentora mag. ing. comp. Tonija Besjedice kojima se još jednom srdačno zahvaljujem. Također izjavljujem da se tijekom izrade ovog diplomskog rada nisam koristila nedopuštenim sredstvima, iznosila lažne tvrdnje i podatke niti neovlašteno i nepropisno koristila tuđim intelektualnim vlasništvom. Svjesna sam da su takve radnje teška povreda akademske čestitosti, te da su, sukladno s vrijedećim propisima i pravilnicima Sveučilišta u Dubrovniku, podložne sankcijama.

Mia Lazo