Dinamička web aplikacija za organizaciju rada knjižnice

Smilović, Ivan

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:028299

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-07-13**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Stručni studij

DINAMIČKA WEB APLIKACIJA ZA ORGANIZACIJU RADA KNJIŽNICE

Završni rad

Ivan Smilović

Osijek, 2020

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

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

Osijek, 27.09.2020.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Ime i prezime studenta:	Ivan Smilović
Studij, smjer:	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
Mat. br. studenta, godina	AI4577, 21.10.2019.
OIB studenta:	40886004446
Mentor:	Marina Peko
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Robert Šojo
Član Povjerenstva 1:	Marina Peko
Član Povjerenstva 2:	dr.sc. Ivana Hartmann-Tolić
Naslov završnog rada:	Dinamička web aplikacija za organizaciju rada knjižnice
Znanstvena grana rada:	**Procesno računarstvo (zn. polje računarstvo)**
Zadatak završnog rada	Izraditi dinamičku web aplikaciju koja će služiti kao pomoć za rad knjižnice. Kreirati najmanje dva korisnička profila koja se razlikuju po ovlastima. Administratorski profil treba imati mogućnost dodavanja, izmjene i brisanja korisničkih profila. Korisničkom profilu omogućiti bilježenje i pregled trenutno aktivnih korisnika te evidenciju plaćanja članarina, posudbe knjiga u knjižnici prema prezimenu autora ili imenu knjige. Tehnologije za izradu zadatka: HTML, CSS, JavaScript, PHP, MySQL.
Prijedlog ocjene pismenog dijela ispita (završnog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	27.09.2020.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

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

IZJAVA O ORIGINALNOSTI RADA

Osijek, 14.09.2020.

Ime i prezime studenta:	Ivan Smilović
Studij:	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
Mat. br. studenta, godina upisa:	AI4577, 21.10.2019.
Turnitin podudaranje [%]:	17

Ovom izjavom izjavljujem da je rad pod nazivom**: Dinamička web aplikacija za organizaciju rada knjižnice**

izrađen pod vodstvom mentora Marina Peko

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1

1.1. ZADATAK ZAVRŠNOG RADA 1

2. KORIŠTENE TEHNOLOGIJE ZA IZRADU *WEB* APLIKACIJE 2

2.1. *HTML* 2

2.2. *CSS* 3

2.3. *PHP* 4

2.4. *JAVASCRIPT* 5

2.5. *JQUERY* 7

2.6. *BOOTSTRAP* 8

3. IMPLEMENTACIJA *WEB* STRANICE 9

3.1. BAZA PODATAKA 9

3.2. REGISTRACIJA KORISNIKA 11

3.3. PRIJAVA KORISNIKA 14

4. PREGLED *WEB* STRANICE 16

4.1. NASLOVNA STRANICA 16

4.2. POPIS KNJIGA 16

4.3. POSUDBE 17

4.4. ČLANARINE 18

4.5. KORISNICI 19

5. ZAKLJUČAK 19

LITERATURA 20

SAŽETAK 21

ABSTRACT 24

ŽIVOTOPIS 23

PRILOZI 26

1. UVOD

U završnom radu zadatak je napraviti dinamičku *web* aplikaciju za organizaciju rada knjižnice kako bi korisnicima knjižnice omogućili jednostavnu posudbu knjiga putem *web* aplikacije. Pri stvaranju dinamičke *web* aplikacije korištene su razne tehnologije koje su objašnjene u poglavlju „Korištene tehnologije za izradu dinamičke *web* aplikacije“ dok su izrada i opis funkcionalnosti aplikacije objašnjeni u poglavlju „Implementacija *web* stranice“.

1.1. Zadatak završnog rada

Zadatak završnog rada je izrada dinamičke *web* aplikacije koja će služiti kao pomoć za rad knjižnice. Djelatnik knjižnice je njen glavni administrator, a članovi knjižnice su registrirani korisnici. Članovima knjižnice biti će omogućena posudba knjige, a djelatniku knjižnice biti će dan uvid u sve posudbe knjiga registriranih članova. Dinamička *web* aplikacija nudit će i dodatne mogućnosti poput evidencije u plaćanje članarine, te mogućnost dodavanja, brisanja i izmjene raznih korisničkih profila i knjiga u knjižnici.

2. KORIŠTENE TEHNOLOGIJE ZA IZRADU *WEB* APLIKACIJE

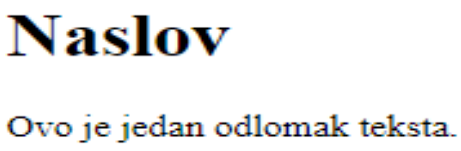
Korištene tehnologije za izradu dinamičke *web* aplikacije su opisni jezici (*HTML, CSS*), razvojni okvir *Bootstrap*, programski jezici (*JavaScript* odnosno njena biblioteka *jQuery, PHP*) te program za upravljanje bazom podataka *MySQL*. Za njihovo pisanje nije potrebno razvojno okruženje, ali kako bi se ono olakšalo i radi jasnijeg pregleda kôda koristi se *Visual Studio Code* razvijen od strane *Microsofta*.

2.1. HTML

HTML (engl. *Hypertext Markup Language*) predstavlja opisni jezik koji se koristi pri izradi *web* stranice. Pojam *Hypertext Markup* predstavlja jezik za označavanje i mogućnost povezivanja dokumenata koji se izvode koristeći znakove. *HTML* predstavlja osnovu pri izradi *web* stranice tj. njen glavni kostur, dok se oblikovanje *web* stranice obavlja preko drugih opisnih jezika. Osnovna zadaća *HTML*-a je dana uputa *web* pregledniku kako da prikaže hipertekst dokument koji treba izgledati u svakom *web* pregledniku jednako. Zbog jednostavnosti i lakoće učenja popularan je i prihvaćen u svijetu, najviše iz razloga što je od samih početaka dostupan svima potpuno besplatno. *HTML* ne spada u programske jezike poput: *C, C++, Java* jer njegovim korištenjem nije moguće riješiti zadatke koji se izvršavaju pomoću programskih jezika. *HTML* dokument predstavlja tekstualnu datoteku koja ima ekstenziju .html te kada se *HTML* dokument ispiše u *web* preglednik, *HTML* dokument postaje objekt koji je prikazan kao njegov korijenski čvor. Svaki *HTML* dokument se sastoji od osnovnih *HTML* elemenata kao što su **<html>** kojim se označava početak *HTML* dokumenta. Unutar elementa koji označava početak nalaze se element **<head>** te **<body>**. Element <**head**> predstavlja zaglavlje *HTML* dokumenta u kojem se najčešće definiraje značajke poput naslova stranice. U samom <**body**> elementu stvara se sadržaj *HTML* dokumenta. Slika 2.1. prikazuje ključne elemenata HTML dokumenta, dok slika 2.2. prikazuje izgled jednostavnog primjera *HTML* kôda u *web* pregledniku.

```html
<!DOCTYPE html>
<html>
<head>
<title>Jednostavan HTML primjer</title>
</head>
<body>
<h1> Naslov </h1>
<p>Ovo je jedan odlomak teksta.</p>
</body>
</html>
```

Slika 2.1. *Jednostavan primjer HTML kôd.*



Slika 2.2. *Prikaz jednostavnog primjera HTML kôda u web pregledniku.*

2.2. CSS

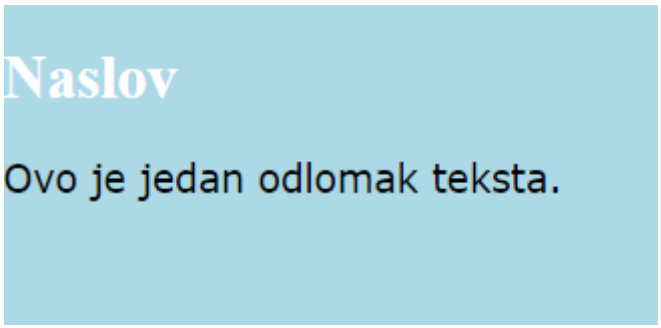
Za dizajn *web* aplikacije zadužen je stilski jezik *CSS* (engl. *cascading style sheets*) koji služi za uređivanje *HTML* dokumenta. Uz *HTML* i *JavaScript*, *CSS* predstavlja glavnu tehnologiju za razvoj web stranica koja ima ekstenziju .css. Dizajn *web* aplikacije poželjno je odvojiti u zaseban *CSS* dokument u odnosu na *HTML* dokumente zbog lakše preglednosti i održavanja. Sintaksa *CSS*-a je oslonjena na samo dva elementa, a to je selektor i deklaracijski blok. Selektor označava element na koje se stilsko pravilo odnosi tj. nad kojim će se izvršiti deklaracijski blok, dok deklaracijski blok označava kako opisani sadržaj *CSS*-om izgleda. Slika 2.3. prikazuje jednostavan primjer *CSS* kôda, dok na slici 2.4. je prikazan izgled jednostavnog primjera *CSS* kôda u *web* pregledniku.

```css
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: left;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```

Slika 2.3. *Jednostavan primjer CSS kôda.*



Slika 2.4. *Prikaz jednostavnog primjera CSS kôda u web pregledniku.*

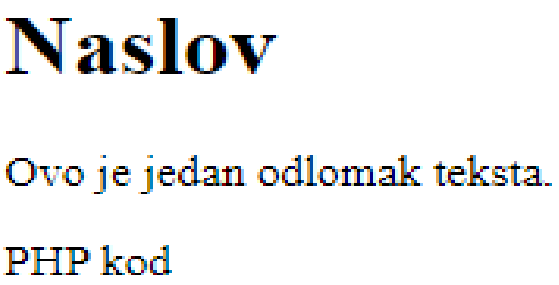
2.3. PHP

Kako bi dinamičke *web* stranice i aplikacije imale svoju funkcionalnost zaslužan je skriptni programski jezik *PHP* (engl. *Hypertext preprocessor). PHP* kôd se uključuje u *HTML* kôd te pri obradi *web* poslužitelj prepoznaje *PHP* oznake te izvršava kôd koji se nalazi samo unutar njih. Ostali dio kôda *web* poslužitelj će zanemariti te definirati kao običan *HTML* koji se samo prosljeđuje. *PHP* je *open-source* što znači da svako tko želi može upotrijebiti izvorne PHP kôdove. Prilikom korištenja *PHP* jezika nije određeno samo generiranje *HTML*-a nego je moguće i generirati slike, Flash datoteke, *PDF* itd. Sintaksa *PHP* programskog jezika je orijentirana po *C* sintaksi te se samim time izvršava na poslužiteljskoj strani. Velika prednost *PHP* programskog jezika je u tome što podržava sve popularne baze podataka kao što je primjer u dinamičkoj *web* aplikaciji *MySQL* baza podataka, ali i drugih poput: *dBase, Oracle* itd. *PHP* kôd nije vidljiv klijentu te kôd možemo zapisati u tekstnim uređivačima kao što je *Notepad* gdje

datoteku spremamo koristeći ekstenziju .php. Promatrajući razne jezike za programiranje *web* aplikacije možemo reći da je *PHP* jezik jedan od najzastupljenijih jezika. Slobodan je i njegovo korištenje ne zahtjeva nikakve troškove. Slika 2.5. prikazuje jednostavan primjer *PHP* kôda koji započinje sa „<?php“ što označava početak *PHP* kôda, dok završetak kôda se označava sa „?>“. Prikaz jednostavnog primjera *PHP* kôda u *web* pregledniku vidljiv je na slici 2.6.

```
<!DOCTYPE html>
<html>
<head>
  <title> Primjer </title>
</head>
<body>
<h1>Naslov</h1>
<p>Ovo je jedan odlomak teksta.</p>
<?php
  echo "PHP kod ";?>
</body>
</html>
```

Slika 2.5. *Jednostavan primjer PHP kôda.*



Slika 2.6. *Primjer jednostavnog PHP kôda u web pregledniku.*

2.4. JavaScript

Kako bi se povezao *HTML* element te kako bi *web* stranica postala dinamičnija potrebno je napraviti programski jezik koji će svojim karakteristikama biti sličan *Javi* radi lakšeg korištenja. Tvrtka *NetScape Communication* napravila je skriptni programski jezik kojeg su radi sličnosti između programskog jezika *Jave* nazvali *JavaScript*. Jezik se pokreće u korisnikovom *web* pregledniku na korisnikovoj strani te nije objektno orijentiran kao *Java*. Najčešće se koristi kako bi dodao interaktivnost *web* stranicama poput animacija i dijaprojekcija itd. Trenutno uz *HTML*

opisani jezik te *CSS* stilski jezik, *JavaScript* je ključan dio svake dinamičke *web* stranice te ima ekstenziju .js. Početak i kraj *Java* skripte označava se koristeći ključnu riječ <**script**>. *JavaScript* kôd se može postaviti na više mjesta npr. u tijelu stranice kada će skripta biti izvršena odmah pri učitavanju stranice. Nadalje, *JavaScript* kôd se može postaviti u zaglavlju stranice ako želimo da se skripta izvrši kad se pozove ili ukoliko se izvrši neki događaj. Slika 2.7. prikazuje jednostavan primjer *JavaScripta* funkcije dok 2.8. i 2.9. predstavljaju izgled funkcije koja prilikom klika na *button* ispisuje tekst.

```
<!DOCTYPE html>
<html>
<body>

<h1>Onclick </h1>

<p>Klikni na button</p>

<button onclick="myFunction()">Klikni</button>

<p id="demo"></p>

<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Uspješno si kliknio";
}
</script>

</body>
</html>
```

Slika 2.7. *Jednostavan primjer JavaScript funkcije koja na klik ispisuje tekst.*

Slika 2.8. *Prikaz izgleda stranice prije klika na button.*

Slika 2.9. *Prikaz izgleda stranice poslije klika na button.*

2.5. jQuery

JQuery predstavlja skupinu naredbi *JavaScript* koje su pojednostavljene, tako da ga ne možemo opisati kao programski jezik. Stoga *jQuery* predstavlja jednu od najpoznatijih biblioteka *JavaScript* programa koja je u kratkom roku zbog svoje jednostavnosti postala izrazito popularna. Korištenjem *JavaScript*-a možemo kreirati animacije i dijaprojekcije na jednostavniji način te upravljati skriptama koje se nalaze u *HTML* dokumentima. U završnom radu korišten je *jQuery* jer olakšava korištenje *JavaScript*-a, iz razloga što sa jednom linijom kôda možemo pozvati funkciju za koju bi koristeći *JavaScript* metodu morali uložiti više vremena, truda i rada. Slika 2.10. prikazuje način dohvaćanja datoteke za normalan rad JS unutar aplikacije.

```
<script src="https://code.jquery.com/jquery-3.2.1.min.js"
integrity="sha256-hwg4gsxgFZhOsEEamdOYGBf13FyQuiTwlAQgxVSNgt4=" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>
```

Slika 2.10. *Dohvaćanje datoteka preko CDN-a za normalan rad JS unutar aplikacije.*

2.6. Bootstrap

Bootstrap predstavlja skup alata za razvoj *web* aplikacija koji su otvorenog kôda napravljeni u *CSS*-u i *JavaScript*-u. Alat *Bootstrap* omogućuje respozivnost *web* stranice tako da *web* stranice imaju isti izgled na različitim internet preglednicima kao i na različitim mobilnim uređajima. Dakle, možemo reći da *Bootstrap* predstavlja jednu zbirku *JavaScript* funkcija te *CSS* klasa jer sadrži *JavaScript* te *CSS* predloške sa kojima se mogu kreirati tipke, navigacija te ostale komponente. Slika 2.11. prikazuje dohvaćanja *bootstrap css* datoteke preko *CDN*-a.

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
<link rel="stylesheet" href="assets/css/style.css">
```

Slika 2.11. *Dohvaćanje bootstrap css datoteke preko CDN-a za olakšano stiliziranje i respozivnost web aplikacije*.

3. IMPLEMENTACIJA WEB STRANICE

U poglavlju Implementacija *web* stranice prikazan je detaljniji uvid u strukturu same *web* aplikacije. Opisuje na koji je način osmišljena baza podataka te glavne funkcije za služenje samom aplikacijom.

3.1. Baza podataka

Kako bi se implementirala *web* stranica potrebno je kreirati bazu podataka u koju se spremaju podatci kao što je registracija korisnika, knjige koje se nalaze u knjižnici itd. Za rad se koristi *MySQL* baza koja dolazi u sklopu *XAMPP* programa s namjenom za rad u lokalnoj mreži. Slika 3.1. prikazuje povezivanje na bazu podataka „knjiznica“. Prikaz svih tablica prikazan je na slici 3.2. dok slika 3.3. prikazuje samu vezu između njih.

```
$connect = new PDO("mysql:host=localhost;dbname=knjiznica","root","");
$connect->setAttribute( PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION );
$connect->exec("set names utf8;");
```

Slika 3.1. *Konekcija na bazu podataka.*

Tablice koje su korištene u izradi ove dinamičke *web* aplikacije su:

- *Book*
- *Borrow*
- *Membership*
- *User*
- *Zanr*

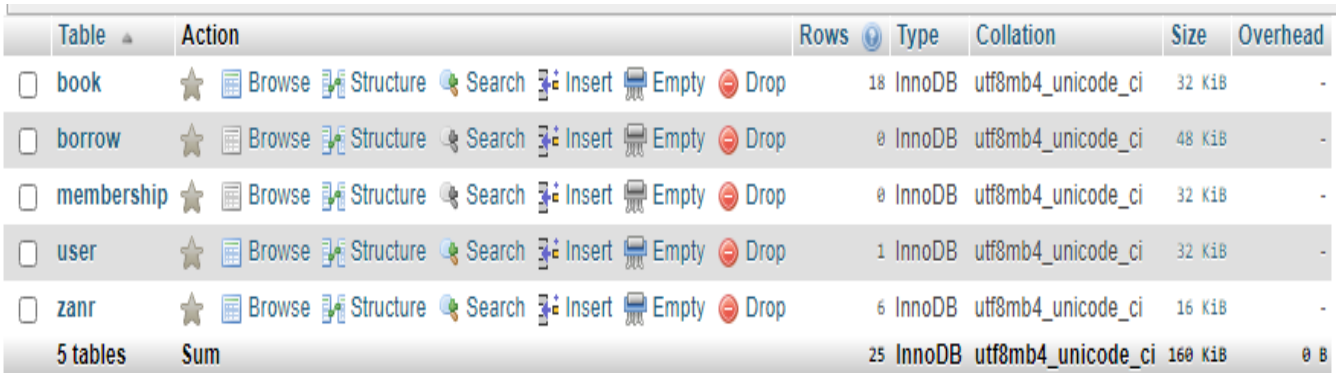


	Table	Action	Rows	Type	Collation	Size	Overhead
☐	book	Browse Structure Search Insert Empty Drop	18	InnoDB	utf8mb4_unicode_ci	32 KiB	-
☐	borrow	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48 KiB	-
☐	membership	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32 KiB	-
☐	user	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	32 KiB	-
☐	zanr	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	16 KiB	-
	5 tables	Sum	25	InnoDB	utf8mb4_unicode_ci	160 KiB	0 B

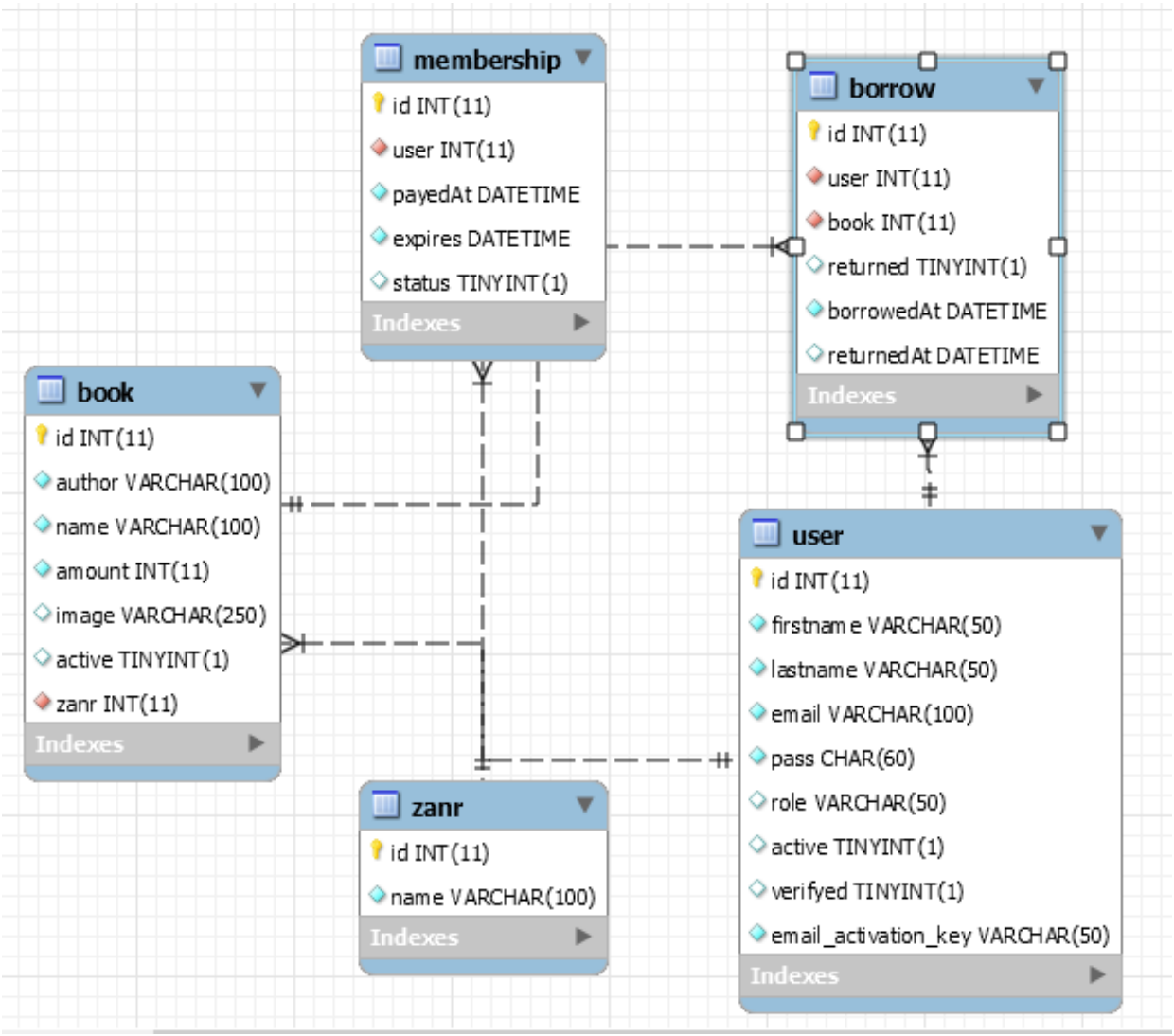
Slika 12.2. *Prikaz svih tablica.*

U tablici 3.1. prikazan je popis svih tablica te njihovih podataka koje sadrži pojedina tablica.

Tablica 3.1. *Prikaz svih tablica i njenih podataka.*

Naziv tablice	Podaci
BOOK	Id autor name amount pass active location
BORROW	Id user book returned borrowerdAt returnerdAt

MEMBERSHIP	Id user payedAt expires status
USER	Id firstname lastname email pass role active verifyed email_activation_key
ZANR	Dramski romani Fantastika Kriminalistički romani Putopis Znanstvena fantastika Humoristični romani



Slika 3.3. *Veza između tablica.*

3.2. Registracija korisnika

Kako bi se korisnik mogao započeti služiti aplikacijom potrebno je prvo izvršiti registraciju. Registraciju korisnik izvršava tako da ispuni potrebne podatke koji su navedeni u formi. Slika 3.4. prikazuje forumu za registraciju, dok je izgled forme prikazan na slici 3.5.

```
<form id="login-form" class="form" action="<?php echo $_SERVER["PHP_SELF"] ?>" method="post">
    <h3 class="text-center text-info">Registracija</h3>
    <div class="form-group">
        <label for="firstname" class="text-info">Ime:</label><br>
        <input type="text" name="firstname" id="firstname" class="form-control">
    </div>
    <div class="form-group">
        <label for="lastname" class="text-info">Prezime:</label><br>
        <input type="text" name="lastname" id="lastname" class="form-control">
    </div>
    <div class="form-group">
        <label for="email" class="text-info">Email:</label><br>
        <input type="email" name="email" id="email" class="form-control">
    </div>
    <div class="form-group">
        <label for="password" class="text-info">Lozinka:</label><br>
        <input type="password" name="password" id="password" class="form-control">
    </div>
    <div class="form-group">
        <input type="submit" name="register" class="btn btn-info btn-md" value="Registriraj se">
    </div>
</form>
```

Slika 3.4. *Forma za registraciju korisnika.*

Slika 3.5. *Izgled forme za registraciju korisnika.*

Nakon što korisnik ispuni potrebne podatke, skripta provjerava da li je forma uspješno poslana te da li su upisani podaci važeći kako ne bi postojao ranije kreirani korisnik sa istom *e-mail* adresom. Kada je korisnik prihvaćen podaci se spremaju u bazu podataka, lozinka se kriptira koristeći *bycrypt hashing function*, a korisnik potom dobiva informaciju da je uspješno registriran. Novi registrirani korisnik automatski dobiva ulogu običnog korisnika, a putem *e-mail*

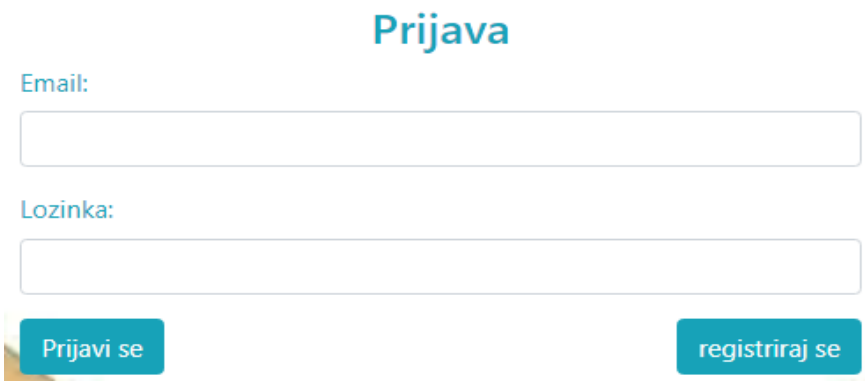
dostavljenu obavijest za aktivaciju korisničkog računa. Nakon aktivacije, korisniku je omogućena posudba knjiga.

3.3. Prijava korisnika

Svaki novi korisnik koji je registriran, kako bi započeo s posudbom knjiga, mora se prijaviti. Podaci potrebni za prijavu su korisnikov *e-mail* i lozinka. Ako uneseni *e-mail* postoji u bazi podataka te se lozinka podudara s unesenom lozinkom, korisnik će biti uspješno prijavljen. Ukoliko podaci ne odgovaraju, korisnik će dobiti odgovarajuću obavijest. Slika 3.6. prikazuje formu za prijavu, dok je sam izgled forme prikazan na slici 3.7.

```
<form id="login-form" class="form" action="authorizathion.php" method="post">
    <h3 class="text-center text-info">Prijava</h3>
    <div class="form-group">
        <label for="username" class="text-info">Email:</label><br>
        <input type="email" name="email" id="email" class="form-control">
    </div>
    <div class="form-group">
        <label for="password" class="text-info">Lozinka:</label><br>
        <input type="password" name="password" id="password" class="form-control">
    </div>
    <div class="form-group">
        <input type="submit" name="login" class="btn btn-info btn-md" value="Prijavi se">
        <a href="<?php echo $path ?>register.php" class="float-right btn btn-info">registriraj se</a>
    </div>
</form>
```

Slika 3.6. *Forma za prijavu korisnika.*



Slika 3.7. *Izgled forme za prijavu korisnika.*

U aplikaciji razlikujemo tri vrste korisnika po njihovim pravima, a to su admin, korisnik te gost. Admin, odnosno djelatnik knjižnice ima mogućnost dodavanja, uređivanja i uklanjanja

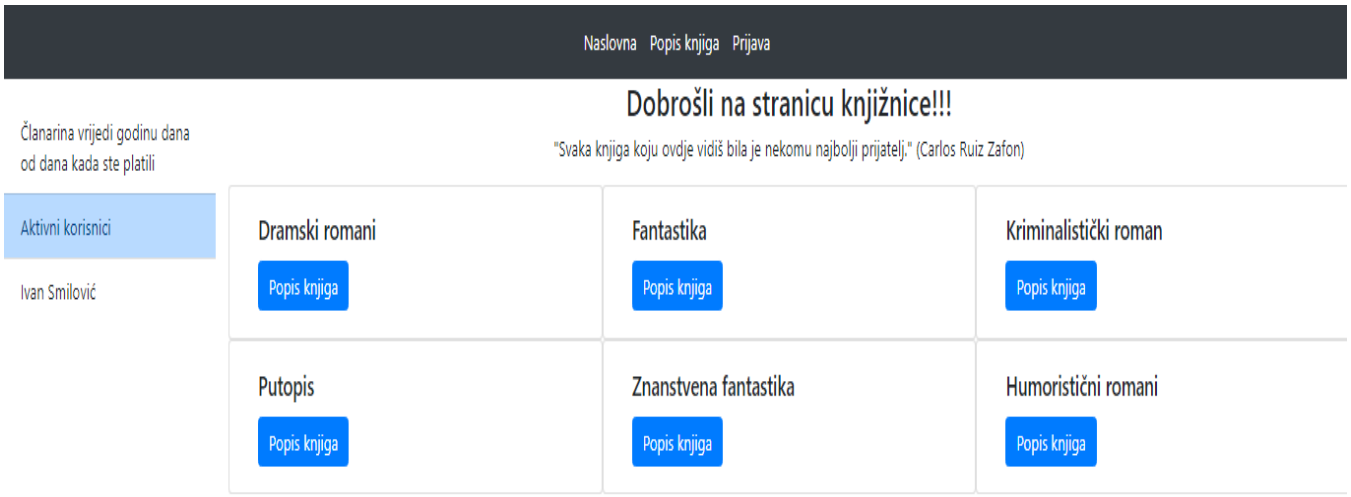
knjiga i registriranih korisnika. Ima mogućnost uvida u sve posudbe te može izvršiti posudbu knjige umjesto korisnika. Admin ima uvid u sve članarine, izvršava uplatu članarine umjesto korisnika. Korisnik nakon uplate godišnje članarine može posuditi maksimalno pet knjiga, dok gost kao korisnik ima uvid samo u pregled knjiga koje knjižnica posjeduje.

4. PREGLED WEB STRANICE

U ovom poglavlju prikazan je pregled cijele *web* aplikacije i izgled dodatnih djelatnikovih mogućnosti kojima korisnik ne može pristupiti. Za dizajn aplikacije zaslužan je ranije spomenuti stilski jezik *CSS*.

4.1. Naslovna stranica

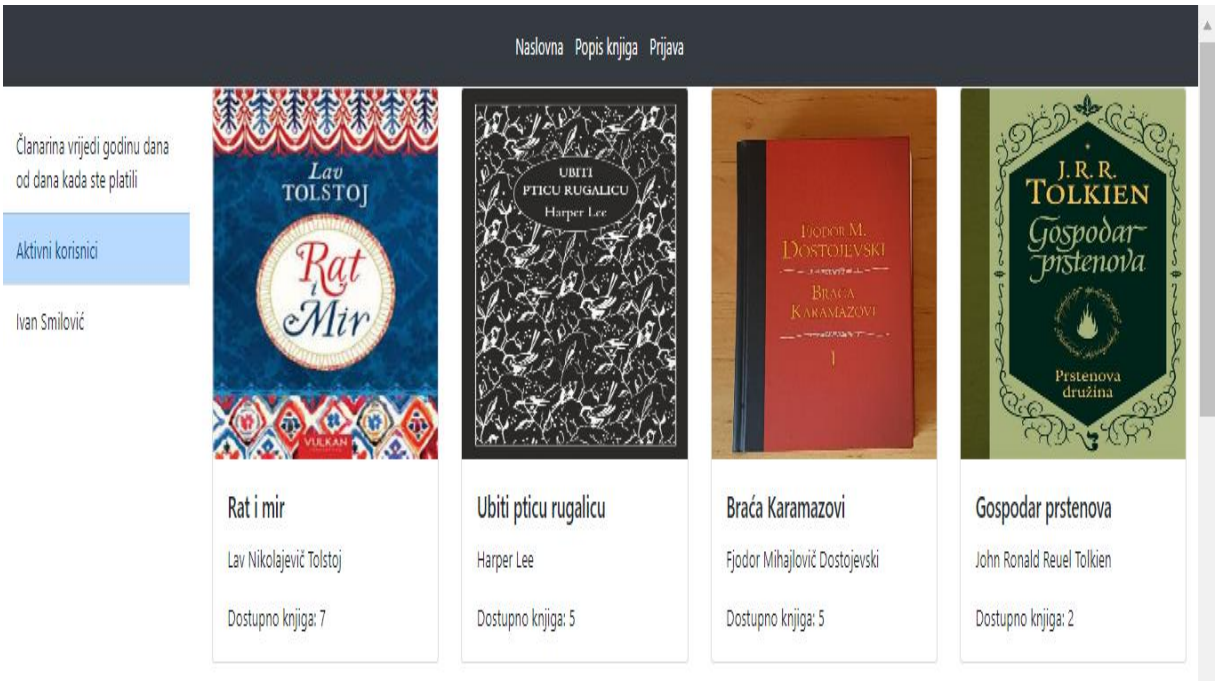
Na naslovnoj stranici nalazi se navigacijska traka koja sadrži opcije za navigaciju po *web* stranici. Na njoj se nalazi i bočna traka koja prikazuje trenutno prijavljene korisnike i informacije o knjižnici. Slika 4.1. prikazuje izgled naslovne stranice korisnika koji pristupaju stranici.



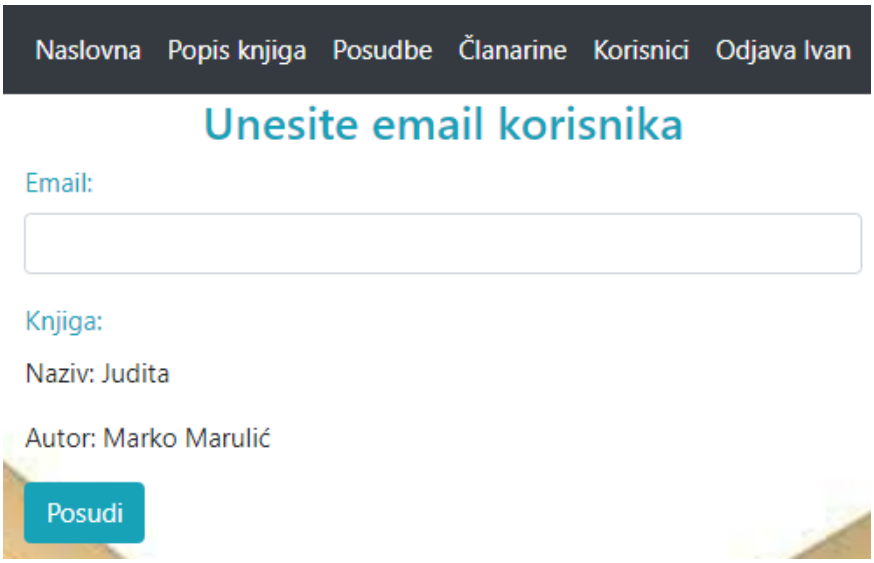
Slika 4.1. *Naslovna stranica.*

4.2. Popis knjiga

Odabirom linka „Popis knjiga“ prikazuju se sve dostupne knjige u knjižnici, a za svaku pojedinačnu knjigu prikazana je njena dostupnost. Izgled stranice razlikuje se ovisno o vrsti korisnika. Gost ima uvid u sve knjige, ali nema mogućnost posuđivanja istih. Prijavljeni korisnik uz uvid svih knjiga ima mogućnost posuđivanja knjiga, dok admin ima dodatnu mogućnost dodavanja, uređivanja i uklanjanja knjiga. Na slici 4.2. prikazan je primjer izgleda „Popis knjiga“, dok na slici 4.3. izgled forme za posuđivanje knjiga za određenog korisnika.



Slika 4.2. *Primjer izgleda "Popis knjiga" za korisnika.*



Slika 4.3. *Izgled forme za posuđivanje knjiga za određenog korisnika.*

4.3. Posudbe

Posudbe prikazuju popis posuđenih knjiga ovisno o vrsti korisnika. Gost nema uvid u posudbe. Prijavljeni korisnik ima uvid samo u svoje posudbe, dok djelatnik ima uvid u posudbe svih korisnika. Nakon što korisnik vrati knjigu, djelatnik mora potvrditi da je knjiga vraćena te se

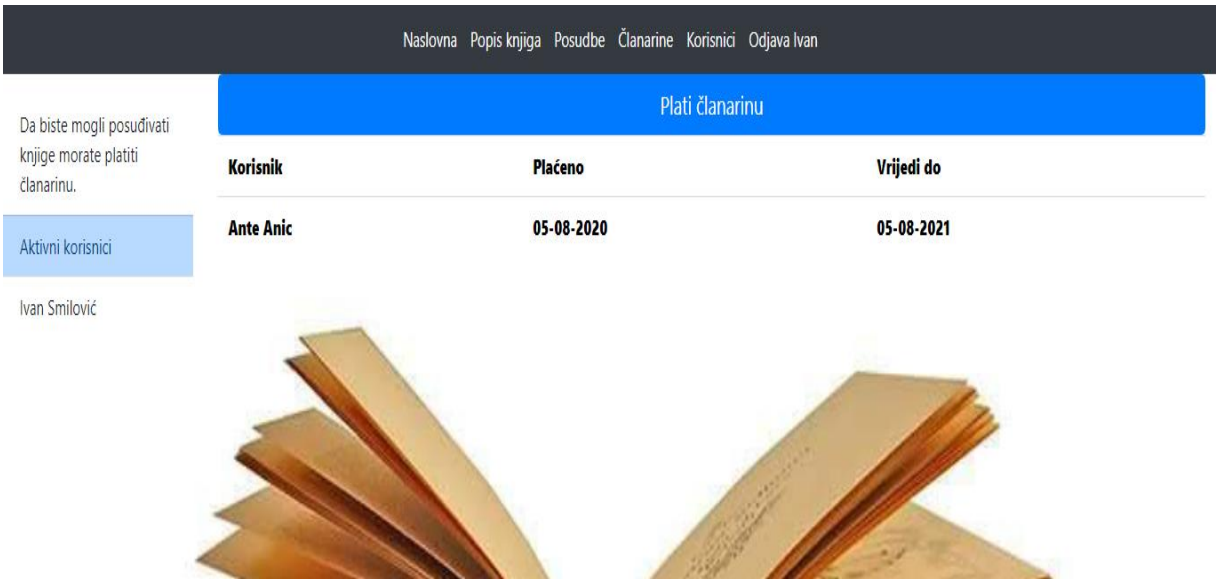
samim time količina dostupnosti određenih knjiga povećava. Izgled „Posudbe“ prikazan je na slici 4.4.

Slika 4.4. *Izgled „Posudbe“ za prijavljenog djelatnika.*

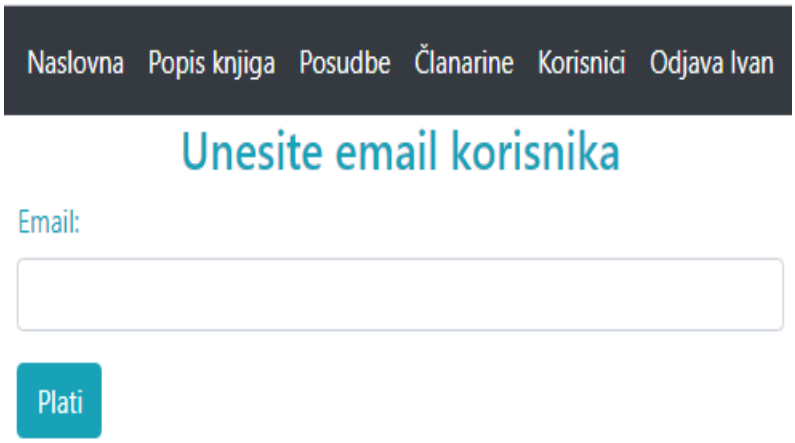
Nakon što korisnik posudi knjigu on ima rok od 14 dana za vraćanje knjige. 2 dana prije isteka roka korisnik dobiva automatski *e-mail* sa upozorenjem da mu za 2 dana ističe rok za vraćanje knjige. Prilikom vraćanja knjige, skripta provjerava da li je ista vraćena u roku te ukoliko nije, računa se zakasnina koju korisnik mora platiti.

4.4. Članarine

Odabirom linka „Članarine“ prikazuje se popis svih važećih odnosno ne važećih članarina. Korisnik ima mogućnost izvršavanja uplate svoje članarine, dok admin ima mogućnost izvršavanja uplate članarine za bilo kojeg korisnika. Slika 4.5. prikazuje primjer izgleda „Članarine“, dok slika 4.6. izgled forme kojim djelatnik obavlja uplatu članarine za određenog korisnika.



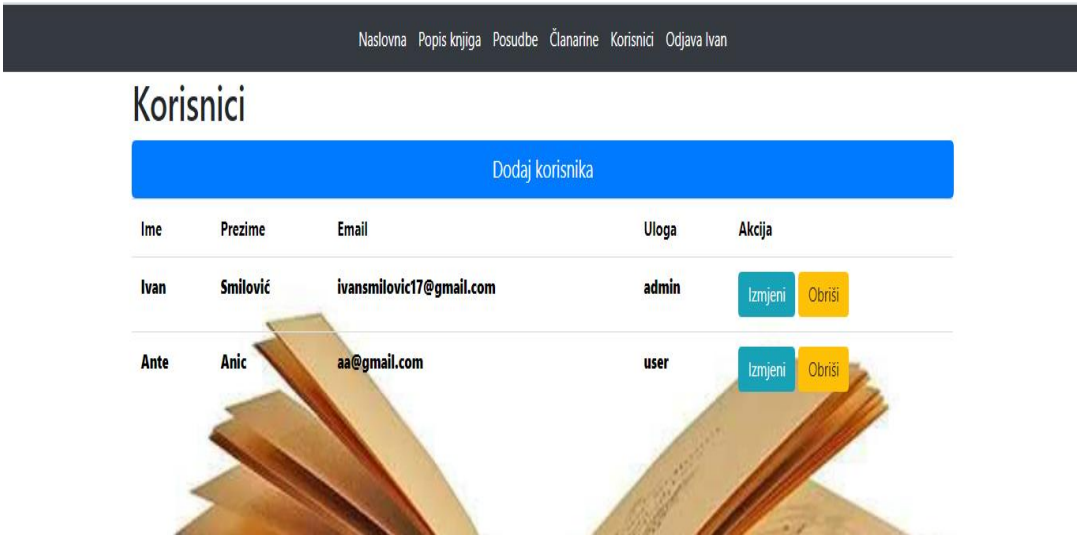
Slika 4.5. *Izgled "Članarine" za prijavljenog djelatnika.*



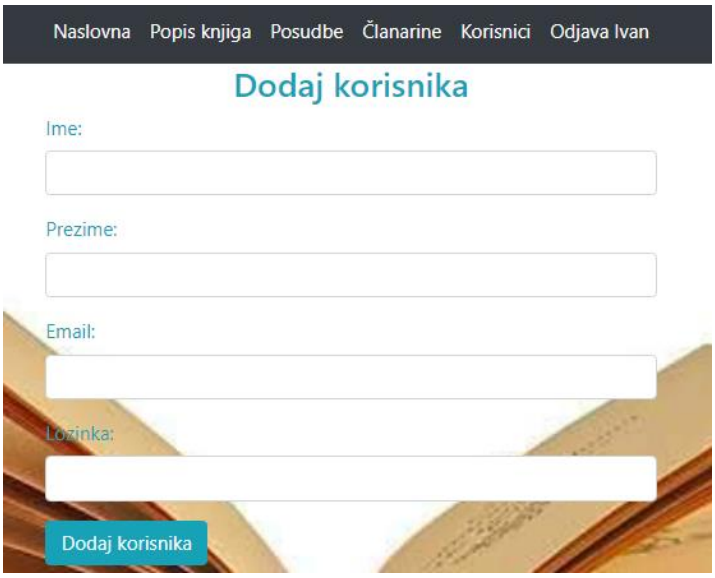
Slika 4.6. *Izgled forme za uplatu članarine za određenog korisnika.*

4.5. Korisnici

Samo djelatnik knjižnice ima uvid u pregled svih korisnika i mogućnost dodavanja, uređivanja i brisanja korisnika. Izgled linka „Korisnici“ prikazan je na slici 4.7. Djelatnik prilikom dodavanja novog korisnika mora ispuniti ponuđena polja te se skriptom provjeriti da li su dobiveni podatci korisnika važeći kao što je navedeno prilikom njegove registracije. Slika 4.8. prikazuje izgled forme kojim djelatnik dodaje novog korisnika.



Slika 4.7. *Izgled "Korisnici" za prijavljenog djelatnika.*



Slika 4.8. *Izgled forme za dodavanje novog korisnika.*

5. ZAKLJUČAK

Zadatak završnog rada bio je kreirati dinamičku *web* aplikaciju koja će služiti kao pomoć radu knjižnice. Dinamička *web* aplikacija za organizaciju rada knjižnice olakšava cjelokupan rad knjižnice kako djelatniku tako i korisnicima. Aplikacija korisniku olakšava praćenje dostupnih knjiga, evidenciju posuđenih te praćenje članarina i zakasnina posuđenih knjiga. U radu su pobliže opisane tehnologije koje se koriste za izradu te uređen izgled *web* stranice kako bi korisnik mogao lakše koristiti kreiranu *web* aplikaciju. Uz opisane tehnologije u završnome radu prikazana je i sama implementacija aplikacije. U konačnici izrađenog završnog rada zaključujemo da *web* aplikacija omogućava lakšu komunikaciju između korisnika i same knjižnice te ujedno olakšava posao djelatnika knjižnice.

LITERATURA

[1] w3schools.com, HTML Turorial, 1999-2020, dostupno na: https://www.w3schools.com/html/ [10.02.2019.]

[2] w3schools.com, JavaScript Turorial, 1999-2020, dostupno na: https://www.w3schools.com/js/ [10.02.2019.]

[3] w3schools.com, CSS Turorial, 1999-2020, dostupno na: https://www.w3schools.com/css/ [10.02.2019.]

[4] w3schools.cm, PHP Tutorial, 1999-2020, dostupno na: https://www.w3schools.com/php/DEFAULT.asp [10.02.2019.]

[5] MySQL.com, MySQL Workbench, 2020, dostupno na: https://www.mysql.com/products/workbench/ [10.02.2019.]

[6] Wikipedia.org, Bootstrap (front-end framework, 2011, dostupno na: https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework) [20.02.2019.]

SAŽETAK

Naslov: Dinamička *web* aplikacija za organizaciju rada knjižnice

Zadatak završnog rada je napraviti dinamičku *web* aplikaciju za organizaciju rada knjižnice. Djelatnik knjižnice ima mogućnost dodavanja, uklanjanja i uređivanja knjiga i korisnika, uvid u sve posudbe kao i mogućnost izvršavanja posudbi umjesto korisnika te uvid u sve članarine i mogućnost izvršavanja uplate članarine umjesto korisnika. Registrirani korisnici koji uplate godišnju članarinu imaju mogućnost posudbe maksimalno pet knjiga, dok gost ima samo uvid u pregled knjiga koje knjižnica posjeduje. Pri izdradi ove dinamičke *web* aplikacije korišten je *HTML, PHP, CSS, MySQL, JavaScript, jQuery, Bootstrap*, dok za lakše pisanje i pregled korišten je *Visual Studio Code*.

Ključne riječi: *Bootstrap*, djelatnik, *HTML*, knjižnica, korisnik, *web* aplikacija

ABSTRACT

Title: Dynamic web application for organizing the work of libraries

The task of this final work is to create a dynamic web application for organizing the work of the library. The library employee has the option of adding, removing and editing books and users, insight into all loans as well as the ability to make loans instead of users and insight into all membership fees and the ability to make membership fees instead of users. Registered users who pay the annual membershio fee have the option of borrowing a maximum of five books, while the guest only has an insight into the overview of books that the library has. HTML, PHP, CSS, MySQL, JavaScript, jQuery, Bootstrap were use din the development of this dynamic web application, while Visual Studio Code was used for easier writing and viewing.

Keywords: Bootstrap, employee, HTML, libary, user, web application

ŽIVOTOPIS

Ivan Smilović rođen 24.9.1997. godine u Vinkovcima. Završio je osnovnu školu Zrinskih Nuštar u Ceriću 2012. godine nakon čega iste godine upisuje Tehničku školu Ruđera Boškovića u Vinkovcima, smjer Tehničar za mehatroniku koju je završio 2016. godine. 2016. godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija. Stručnu praksu je odradio u firmi HGSPOT informatika u Vinkovcima gdje je stekao iskustvo u održavanju i sastavljanju računala. U sklopu kolegija Multimedijska tehnika napravio je osobnu web stranicu. Koristi se engleskim jezikom te ima vrlo dobro znanje na računalu (korisnik *Microsoft Office* paketa) te poznaje programske jezike kao što su *C, Java* i *Python*. Iskustvo u izradi web stranica s primjenom *HTML*-a za prikaz i *CSS*-a za stiliziranje elemenata, JavaScript za dinamičko upravljanje i *PHP*/*MySQL* za upravljanje bazom podataka. Poznaje *Microsoft .NET* platforme i rad u *Microsoft Visual Studiu* te ima iskustvo rada u grafičkim programima *PhotoShop*, *Inkscape*.

PRILOZI

1. Projektna mapa s izvornim kôdom nalazi se na optičkom disku koji je priložen uz printanu verziju rada zajedno s word i pdf dokumentom.
2. Kôd je stavljen i može se pronaći na GitHub-u na linku: https://github.com/smilovicivan/knjiznica