

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Preddiplomski studij računarstva

Završni rad

Korisničko sučelje za online kalkulator

Rijeka, rujan 2017.

Matija Jakovac
0069064833

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Preddiplomski studij računarstva

Završni rad

Korisničko sučelje za online kalkulatora

Mentor: izv.prof.dr.sc. Miroslav Joler

Rijeka, rujan 2017.

Matija Jakovac
0069064833

Umjesto ove stranice umetnuti zadatak
za završni ili diplomski rad

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, rujan 2017.

Ime Prezime

Zahvala

Zahvaljujem mentoru izv.prof.dr.sc. Miroslavu Joleru na podršci tijekom pisanja ovoga rada i korisnim raspravama i savjetima.

Sadržaj

Popis slika	viii
1 Uvod	1
1.1 Cilj rada	1
1.2 Sadržaj dokumenta	1
2 Aplikacija	2
2.1 Alati i programski jezici	2
2.2 Specifikacije zahtjeva	4
2.2.1 Zaglavlje	4
2.2.2 Body	7
2.2.3 Footer	10
2.2.4 Pozadina aplikacije	12
3 Baza podataka	18
3.1 Konekcija	18
3.2 Aplikacije	19
3.2.1 Izrada aplikacije	19
3.2.2 Prikaz aplikacija	20
3.3 Datoteke	20

Sadržaj

3.4	Logiranje	20
4	Primjer korištenja aplikacije	22
4.1	Baza podataka	22
4.2	<i>Upload</i> direktorij	23
4.3	Prijava	24
4.4	Kreiranje aplikacija	24
4.4.1	Lokalna aplikacija	24
4.4.2	Aplikacija sa drugog servera	26
5	Zaključak	29
	Bibliografija	30
	Sažetak	31

Popis slika

2.1	Izgled <i>search</i> gumba s mogućnosti proširenja.	5
2.2	Izgled izbornika aplikacije.	7
2.3	Izgled tijela aplikacije.	8
2.4	Modalno otvaranje i mogućnost prijavljenog korisnika stvaranja direktorija.	9
2.5	Izgled gumba za otvaranje logiranja korisnika.	10
2.6	Modalni izgled logiranja korisnika u sustav.	11
2.7	Upload datoteke.	12
2.8	<i>Textbox</i> u kojeg zalijepimo željeni <i>URL</i>	14
2.9	Izgled pozadine aplikacije i datoteka.	15
4.1	Izgled baze podataka nakon početnog sređivanja.	23
4.2	Izgled <i>upload</i> direktorija sa stazom na kojoj se nalazi.	24
4.3	Odabir <i>.zip</i> datoteke.	25
4.4	Izgled <i>test</i> tablice nakon izvršenog <i>upload-a</i>	26
4.5	Izgled <i>upload-ane</i> aplikacije.	26
4.6	Izgled datoteke aplikacije s drugog servera.	27
4.7	Izgled aplikacije s drugog servera.	28

Poglavlje 1

Uvod

1.1 Cilj rada

Zadatak je bio napraviti web aplikaciju koja će omogućiti da se već postojeće aplikacije mogu pokretati iz istog web sučelja. Odnosno, aplikacije koje su napravljene zasebno i imaju svaka svoju različitu svrhu, otvaraju se s istog mjesta i jednako su učinkovite.

Također, korisnik s nekim administratorskim pravima može se logirati i zatim ima mogućnost ubacivanja i brisanja aplikacija pomoću tog web sučelja bez da ulazi u kod.

1.2 Sadržaj dokumenta

U ovom radu najprije ćemo opisati kako radi naša aplikacija. Detaljno ćemo opisati svaki dio koda i kako smo pojedinačno riješili koji dio problema.

Zatim ćemo opisati bazu podataka, koja je njena uloga te kako je pomogla pri rješavanju ove aplikacije. Na kraju ćemo dati primjer ubacivanja aplikacija, s vanjskih ili lokalnih servera.

Svaki dio aplikacije koji opisujemo ćemo popratiti sa slikama zbog lakšeg objašnjenja. Također u ovom radu koristit ćemo puno riječ aplikacija. Ta riječ će se koristiti u dva smisla, ova aplikacija, te aplikacije koje će ona sadržavati. Trebat će iz konteksta shvatiti o kojoj aplikaciji je riječ.

Poglavlje 2

Aplikacija

U ovom poglavlju opisat ćemo ideju aplikacije, na koji način radi, koje su njene specifikacije te koje smo komponente koristili.

Kao što smo već spomenuli, naša aplikacija na neki način ima ulogu kišobrana. Njen cilj je napraviti web sučelje koje će već neke gotove aplikacije koje djeluju zasebno, staviti na isto mjesto te će se s tog mjesta moći normalno koristiti.

U početku izrade morao sam odabrati neke alate odnosno softvere pomoću kojih bi uspješno odradio taj zadatak. Pomoć u odabiru bilo mi je već neko iskustvo iz kolegija *"Razvoj Web Aplikacija"* koje sam stekao tijekom tog kolegija.

Također na početku valja spomenuti i da u *"Xampp"* paketu postoji direktorij *"htdocs"* koji je isti kao i *"www"* direktorij kod *"Wamp"* paketa, te je u njemu potrebno napraviti direktorij *upload* koji je kasnije potreban našoj aplikaciji.

2.1 Alati i programski jezici

Alati korišteni za izradu aplikacije su sljedeći:

- *Notepad++* koji ima ulogu uređivanja programskog koda. Ima mogućnost spremanja puno tipova datoteka (*.php*, *.css*, *.html* i druge.) Napredniji je od običnog uređivača teksta upravo i tog razloga što prepoznaje različite vrste datoteka.[1]
- *Xampp* je paket koji služi kako bi mogli lokalno raditi aplikaciju. U sebi sadrži

Poglavlje 2. Aplikacija

Apache server preko kojeg možemo vidjeti aplikaciju u svom web pregledniku i lokalnu bazu podataka na koju se spajamo i u koju pohranjujemo podatke. Vrlo je praktičan i koristan u nedostatku boljih servera.[2]

Pri izradi aplikacije, prema potrebi su se koristili različiti programski jezici:

- *Hyper Text Markup Language* je baza naše aplikacije. *HTML* nam daje statički izgled stranice i koristimo ga u svim dijelovima aplikacije. Jednostavno je neophodan i prisutan je u cijelom programskom kodu. Potreban je za povezivanje ostalih programskih jezika s aplikacijom.[3]
- *Cascade Style Sheet* nije pravi programski jezik, ali je također jedan od bitnijih. Koristimo ga za uređivanje *HTML-a*. Za razliku od *HTML-a* nije neophodan međutim aplikacija je puno pristranija i boljeg izgleda uz njega.[3] *CSS* je tijekom godina evoluirao i danas možemo koristiti primjerice *CSS3* koji nam pruža usluge kao što su animacije u ovom slučaju. U nadopuni s *HTML5* jezikom pruža zanimljive tehnologije, banere, slike, boje, zaglavlja i drugo.
- *PHP* je programski jezik koji je najčešće korišten u ovom projektu. Osnovni je jezik za povezivanje s bazom podataka (za ubacivanje/izbacivanje podataka), kao i za logiranje administratora u sustav. Zapravo daje dinamiku aplikaciji, odnosno daje mogućnost korisniku da sam mijenja stranicu ovisno o autorizaciji.[3]
- *Javascript* je programski jezik dosta sličan *PHP-u*. U počecima razvoja aplikacije, mislio sam da će se uspjeti napraviti bez korištenja *Javascript-a*, međutim kako se aplikacija širila, shvatio sam da ga neću moći izbjeći. *Javascript* koristimo također za dinamičko prikazivanje stranice kao što je u ovom slučaju modalno prikazivanje. Primjerice, korisniku se klikom na izbornik modalno otvori izbornik. Isto tako i tijekom prijave. Bez *Javascript-a* to nebi bilo ostvarivo.
- *jQuery* je platforma odnosno knjižnica ili dodatak *Javascript-u* dizajnirana da pojednostavi klijentsku stranu aplikacije. Dobar je kada trebamo neke događaje (tipkom miša ili tipkovnice), kada je potrebno pretraživanje u realnom vremenu, za animacije i druge stvari. Ovdje se koristi za pretraživanje aplikacija.

- *MySQL* koji nam je tu da bi nam pomogao izvlačiti podatke iz baze podataka.

2.2 Specifikacije zahtjeva

U početku pa i kasnije, od velike pomoći bio je *bootstrap* pomoću kojeg aplikacija nije zahtijevala dodatno uređenje već je imala s interneta gotove klase za korištenje.[4]

U ovom dijelu prikazat ćemo svaku stavku naše aplikacije te objasniti njezinu svrhu. Kao i većina web aplikacija, i ova se sastoji od 3 dijela, zaglavlja, tijela i podnožja.

2.2.1 Zaglavlje

Ulaskom u aplikaciju, prvo što se vidi na vrhu stranice je, naravno, zaglavlje. Na njoj se nalazi sljedeće:

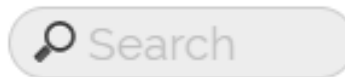
- Naslov aplikacije nalazi se na početku zaglavlja, on je izmišljen i smješten na to mjesto tek toliko da popuni prazninu. Naziv aplikacije je *AppJoiner*. Sam prijevod s engleskog jezika govori nam o kakvoj aplikaciji se radi. Pridruživanje aplikacija.
- *Search* je smješten u sredinu zaglavlja. Kao što smo već rekli, svrha aplikacije je omogućiti već postojećim aplikacijama da se koriste s naše aplikacije. Ako je tih aplikacija jako puno, potrebno nam je nešto što će nam pomoći lakše pronaći aplikaciju koju želimo. U tom na pomaže *Search*.

Pomaže nam tako da mi upisujemo ime aplikacije koje želimo koristiti, a on nam svakim upisom slova ispisuje sve aplikacije koje sadrže upisana slova. Iako strašno podsjeća na *Ajax*, to nije to. Naime, *Ajax* je puno sporiji način *live* pretraživanja. Ako se u bazi podataka nalazi jako puno podataka, on će nam odgovor davati sve sporije i sporije što je više podataka. Stoga, pomoću *jquery-a*, nazovimo to pretraživanjem pomoću indeksa, koristimo naš *Search*. Ovakav način pretraživanja ne ulazi u bazu podataka već pomoću neke oznake traži ono što se nalazi na stranici i onda to prikazuje. U našem slučaju to je *span* oznaka koja daje ime našim aplikacijama iz baze podataka. Dio koda možete vidjeti ovdje:

Poglavlje 2. Aplikacija

```
$(document).ready( function() {  
    jQuery.expr[':'].contains = function(a, i, m) {  
        return jQuery(a).text().toUpperCase()  
            .indexOf(m[3].toUpperCase()) >= 0;  
    };  
    $(' .search-input').keyup(function() {  
        var keyword = $(this).val().toLowerCase(),  
            cont = $(' .row');  
        if(keyword.length > 0) {  
            $(' .col-md-4').addClass('skriveno');  
            cont.find( 'span:contains("' + keyword + '")' ).  
                parents(' .col-md-4').removeClass('skriveno');  
        } else {  
            $(' .col-md-4').removeClass('skriveno');  
        }  
    });  
});
```

Izgled našeg *Search-a* je sličan kao što je svima nama i poznat. Inicijalno je prazan te se proširi ako ga označimo i pripremimo za rad. Njegov izgled možete vidjeti na slici 2.1 .



Slika 2.1 Izgled *search* gumba s mogućnosti proširenja.

- *Menu* je standardni izbornik u kojem se nalaze razne opcije koje nas mogu odvesti na određeni dio stranice ili odraditi za nas nešto što želimo. Prva stvar koju prilikom pritiska na gumb s tri crte u našem desnom kutu

Poglavlje 2. Aplikacija

zaglavlja dobivamo, jest modalno otvaranje našeg izbornika. Modalni izgled je takav da se u fokus stavi samo određen dio (nama je to izbornik), a ostali dio je u sjeni, zatamnjen kao na slici 2.2 .

Svaka stavka izbornika ponaša se kao gumb iako ne liči na njega. Naime to je tekst na koji se može kliknuti i koji se podcrta prilikom prolaska mišom preko određene riječi. Prva stvar koju možemo napraviti u našem izborniku je zatvoriti ga. Klikom na ovu opciju jednostavno se zatvori naš izbornik u modalnom stanju i opet smo na zadnjem mjestu na kojem smo bili prije otvaranja. Valja još spomenuti da se izbornik može otvoriti s bilo kojeg dijela aplikacije, baš iz razloga što se nalazi u zaglavlju koje je nepomično. Međutim, gumb (tako ćemo zvati sve specifikacije u ovom radu koje nam daju neku poveznicu ili nešto naprave) zatvori nije jedini način zatvaranja izbornika. Također se pritiskom miša bilo gdje sa strane izbornika, on zatvara.

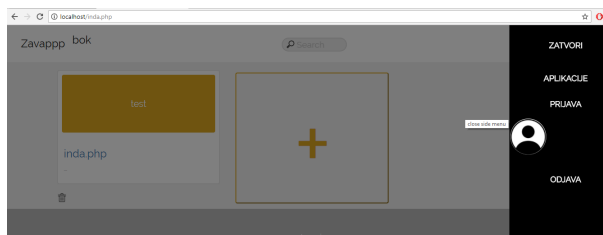
Sljedeći gumb vodi nas na aplikacije. Iako se ulaskom na našu stranicu već nalazimo među aplikacijama, također možemo biti na drugom dijelu web stranice odnosno u podnožju. Tada nam ova opcija služi kako bi nas podigla na gornji dio stranice i početak gdje naše aplikacije počinju.

Nakon nje slijedi nam administratorski dio. Klikom na admin oznaku u našem izborniku, izbornik će nas odvest u administratorski dio odnosno u podnožje koje ćemo pobliže objasniti u podsekciji podnožje.

Posljednji dio izbornika je malo teži za objasniti s obzirom na nismo još pojasnili prijavu korisnika u sustav. Međutim, ako korisnik nije prijavljen u sustav, tada ovaj posljednji dio uopće ne postoji. Ako je korisnik (kada govorimo korisnik mislimo na administratora) prijavljen, tada on ima mogućnost kliknuti na odjavu koja ga kao što sam naziv govori, odjavljuje iz sustava. Također se iznad oznake odjave pojavljuje i ikona korisnika koje je *hard* kodirana.

Svaki dio zaglavlja, i ne samo zaglavlja, cijele aplikacije, koristi *CSS* uređivač za bolji izgled aplikacije. Već u zaglavlju koristili smo nekoliko spomenutih programskih jezika. *Javascript* se koristio kod pozivanja i korištenja izbornika, a *jQuery* kod pretraživanja aplikacija pomoću *Search-a*. Također, zaglavlje ostaje na vrhu stranice fiksno bez obzira gdje se mi nalazimo.

Poglavlje 2. Aplikacija



Slika 2.2 Izgled izbornika aplikacije.

2.2.2 Body

Drugi veći dio aplikacije je njeno tijelo. Ono se nalazi ispod zaglavlja i za razliku od zaglavlja, nije fiksno. Ovisno o tome je li administrator prijavljen u sustav ili ne, možemo vidjeti sljedeće komponente.

- Aplikacije koje su unesene u bazu podataka, možemo vidjeti u ovom dijelu aplikacije. One su napravljene u obliku gumba. Prelaskom miša preko određene aplikacije stvaraju se strelice koje nam pokazuju da klikom miša na mjesto gdje je pokazivač trenutno, otvaramo tu aplikaciju. Napravljene su tako da u jednom redu u tijelu mogu stati samo tri aplikacije. Međutim broj redova nije fiksno. U naše tijelo možemo postaviti neograničen broj aplikacija. Za sam gumb koji je prikazan kao aplikacija, korišten je *CSS*, a za postavljanje njih na stranicu korišten je *bootstrap* koji ima već neke gotove klase pomoću kojih dobivamo bolji izgled aplikacije.

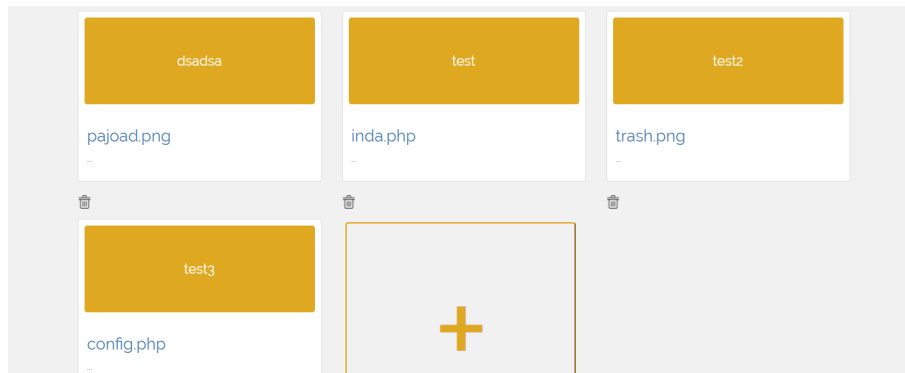
Izgled možete vidjeti na slici 2.3

Aplikacije koje su unesene za korištenje u ovu aplikaciju sastoje se od, možemo reći, na neki način direktorija i glavne datoteke koja pokreće tu aplikaciju kako bi uštedjeli jedan klik mišom. Ulazak u pozadinu aplikacije dopušten je svim korisnicima, kao i otvaranje iste, međutim brisanje aplikacije dopuštena je samo ulogiranom administratoru.

Brisanje aplikacije također je jedna od opcija kojom možemo upravljati aplikacijom, ali samo administrator ima tu opciju iz razloga kojih nije potrebno previše objašnjavati. Što bi bilo kada bi svi imali mogućnost brisanja?

Što se tiče programskog koda, brisanje cijele aplikacije bio je malo teži zadatak

Poglavlje 2. Aplikacija



Slika 2.3 Izgled tijela aplikacije.

zato što neke funkcije *PHP-a* nemaju mogućnost brisanja direktorija ako taj isti nije prazan. Stoga je najprije potrebno izbrisati sve datoteke iz direktorija, zatim njega izbrisati, te zatim urediti bazu podataka. Programski kod izgleda ovako:

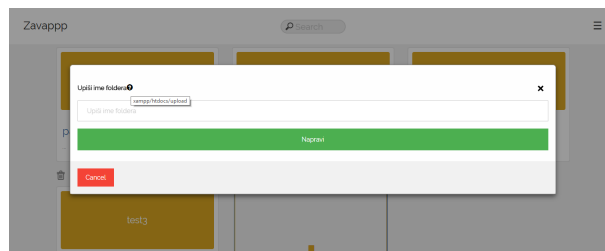
```
if(isset($_GET['hap'])){
    $tabl = $_GET['hap'];
    $dir = "upload/".$tabl;
    function rmdir($dir) {
        if (is_dir($dir)) {
            $objects = scandir($dir);
            foreach ($objects as $object) {
                if ($object != "." && $object != "..") {
                    if (is_dir($dir."/".$object))
                        rmdir($dir."/".$object);
                    else
                        unlink($dir."/".$object);
                }
            }
            rmdir($dir);
        }
    }
}
```


Poglavlje 2. Aplikacija

```
    rmdir($dir);

    $hps = $db->prepare("drop table $tabl");
    if($hps->execute()){
    }else{
    }
    header("location:inda.php");
}
```

- Kao što se može primijetiti na slici 2.3, postoji još jedan dio tijela koji nismo spomenuli. Još jedna administratorska opcija. To je prikazani veliki + koji nam dodaje aplikaciju. Ako smo ulogirani administrator, imamo mogućnost otvaranja ove opcije, koja nam daje mogućnost naziva direktorija po našoj želji (slika 2.4). Međutim, taj naziv ne smije sadržavati razmak zato što je kreacija foldera u bazi podataka napravljena tako da se pod tim nazivom napravi nova tablica u koju ćemo ubacivati datoteke. Direktorij pod tim pogrešnim nazivom će biti napravljen, ali tablica u bazi podataka neće. Zato je potrebno koristiti nazive bez razmaka. Na slici 2.4 ako mišem prođemo preko upitnika, vidjet ćemo točno lokaciju gdje će naš novi direktorij biti napravljen.



Slika 2.4 Modalno otvaranje i mogućnost prijavljenog korisnika stvaranja direktorija.

Programski kod za izradu direktorija i tablice u bazi podataka možete vidjeti ovdje:

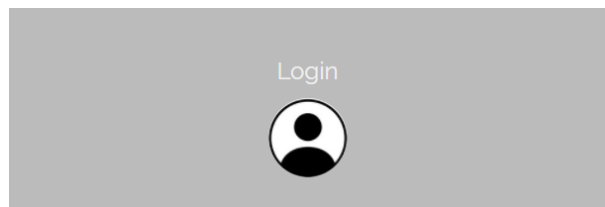
```
if(isset($_POST["fsub"])) {
    $conn = mysqli_connect("localhost","root","","apps");
```

Poglavlje 2. Aplikacija

```
$ime = $_POST['fname'];
mkdir("upload/".$ime);
$sql = "CREATE TABLE ".$_POST['fname']." (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,flag VARCHAR(30) NOT NULL)";
if ($conn->query($sql) === TRUE) {
    echo "Tablica uspješno kreirana";
} else {
    echo "Greška: " . $conn->error;
}
header("location:inda.php");
}
```

2.2.3 Footer

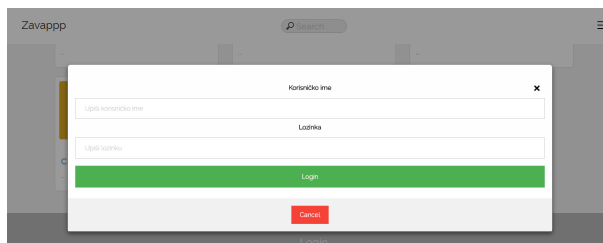
U ovom dijelu postoji samo jedna opcija koju korisnik može odabrati, a to je logiranje korisnika u sustav(prikazano na slici 2.5) Pritiskom na gumb opet nam se otvara



Slika 2.5 Izgled gumba za otvaranje logiranja korisnika.

modalni dio aplikacije koji smo ranije spominjali, a možemo vidjeti na slici 2.6. U njemu korisnik ima mogućnost upisa korisničkog imena i lozinke. Ako je korisnik unio točno korisničko ime i lozinku, ulazi u sustav i ima mogućnosti koje smo ranije navodili, a u protivnom mu se prikazuje poruka da je pogrešna prijava. Što se tiče samog logiranja, ono je prilično nekonzistentno. Napravljeno je tako da je *hard* kodiranu u bazu podataka. Razlog zbog kojeg je tako napravljeno je taj da ukoliko bi primjerice postojala registracija korisnika, svatko bi se mogao registrirati i imati iste

Poglavlje 2. Aplikacija



Slika 2.6 Modalni izgled logiranja korisnika u sustav.

ovlasti kao i ovaj jedan jedini. Smatram da treba postojati samo jedan administrator s ovlastima ubacivanja i brisanja aplikacija ili datoteka, a ostali korisnici bez logiranja imaju mogućnost samo otvaranja datoteka.

Sama kreacija jednog jedinog korisnika bila je takva da se otvorila posebna *.php* datoteka te se u nju upisalo korisničko ime i lozinka koji su se ubacili u bazu podataka. Kod lozinke valja još spomenuti da se koristila *md5* funkcija *PHP-a* kako bi u bazi podataka lozinka bila enkriptirana te bi se teško pročitala. Korisnik ima samo jednu mogućnost logiranja u sustav, a to je ako upiše:

- Korisničko ime: admin
- Lozinka: admin01

Programski kod za logiranje korisnika u sustav je sljedeći:

```
if(isset($_POST['sub']))
{
    $uname = mysqli_real_escape_string($db1,$_POST['uname']);
    $pass = mysqli_real_escape_string($db1,$_POST['pass']);
    $pass = md5($pass);

    $check = mysqli_query($db1,"SELECT * FROM 'user'
    WHERE 'uname' = '$uname' AND 'pass' = '$pass'");
    if(mysqli_num_rows($check) >= 1){
        $_SESSION['uname'] = $uname;
        $_SESSION['password'] = $pass;
```

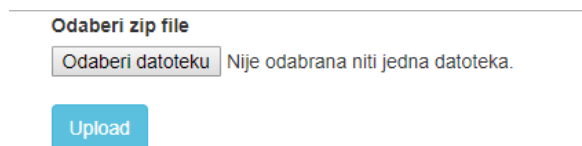
```
    header("location:inda.php");  
    exit();  
}else{  
}  
}
```

2.2.4 Pozadina aplikacije

U podsekciji *body* spomenuli smo dio koji ima na neki način ulogu direktorija. Klikom na taj dio aplikacije, one koju želimo koristiti, otvara nam se nova *.php* datoteka koja nas vodi u pozadinu aplikacije.

Pod pozadinom odabrane aplikacije smatra se sučelje koje nam omogućuje pregled svih datoteka, otvaranje istih, njihovo uređivanje te najbitnije stvari, *upload* datoteke.

- Otvaranjem pozadine aplikacije otvara nam se novi tab s nazivom odabrane aplikacije. U njemu ako još nema datoteka, potrebno je to napraviti. Na slici 2.7 možemo vidjeti kako izgleda odabir datoteke za upload. *Upload* ima dva



Slika 2.7 Upload datoteke.

dijela, jedan je van aplikacije drugi radi aplikacija. Van aplikacije, korisnik je dužan napraviti *.zip* datoteku u kojoj se moraju nalaziti sve datoteke koje želi *upload-ati*. Nakon što je to napravio u aplikaciji odabire tu datoteku i radi *upload*. U *PHP* kodu napravljeno je to da se *.zip* datoteka sama raspakira u već ranije odabrani direktorij aplikacije. Također datoteke odlaze u bazu podataka u odabranu tablicu pod istim nazivom direktorija. Kod za *upload* je sljedeći:

```
if(isset($_POST["btn_zip"]))
```

Poglavlje 2. Aplikacija

```
{
    $output = '';
    if($_FILES['zip_file']['name'] != '') {
        $file_name = $_FILES['zip_file']['name'];
        $array = explode(".", $file_name);
        $name = $array[0];
        $ext = $array[1];
        if($ext == 'zip') {
            $path = 'upload/';
            $location = $path . $file_name;
            if(move_uploaded_file
                ($_FILES['zip_file']['tmp_name'],$location)){
                $zip = new ZipArchive;
                if($zip->open($location)) {
                    $zip->extractTo($path.$_SESSION['key']);
                    $zip->close();
                }
                unlink($location);
            }
        }
    }

    $dir = "upload/".$_SESSION['key'];
    function rdir($dir) {
        include "config.php";
        if (is_dir($dir)) {
            $objects = scandir($dir);
            foreach ($objects as $object) {
                if ($object != "." && $object != "..") {
                    if (is_dir($dir."/".$object)){
                        rdir($dir."/".$object);
                    }
                    else{

```

```
$dir2=$dir."/". $object;
    $add = $db->prepare
        ("insert into ".$_SESSION['key']."
         values('','?', '0', '$dir2')");
    $add->bindParam(1,$object);
    $add->execute();
}
}
}
}
}
rmdir($dir);
header("location:proba2.php");
}
```

- Postoji još jedan način da se napravi *upload* aplikacije koji do sad nismo spomenuli. Ako postoji neka aplikacija na nekom drugom serveru, a nije na lokalnom računalu, postoji mogućnost da se i ona koristi.

Na slici 2.8 vidimo jedan *textbox* u koji jednostavno samo zalijepimo *URL* aplikacije koja se nalazi na nekom drugom serveru. Aplikacija će imati samo

Zalijepi URL

Slika 2.8 *Textbox* u kojeg zalijepimo željeni *URL*.

jednu datoteku koja zapravo neće ni biti prava datoteka. U bazi podataka će se stvoriti samo jedan redak koji će imati naziv *URL-a*. Taj naziv će nam biti jednostavno link na željenu stranicu s koje želimo otvarati aplikaciju. Kod je sljedeći:

```
if(isset($_POST['sub_url'])) {
```

Poglavlje 2. Aplikacija

```
$conn = mysqli_connect("localhost","root","","apps");
$url=$_POST['text_url'];
$time = $_SESSION['key'];
$safe_url = mysqli_real_escape_string($conn, $url);
$stmt = $db->prepare
("insert into $time values('', '$safe_url', '1', '$safe_url')");
$stmt ->execute();
}
```

- Kao što možete vidjeti na slici 2.9 sustav izgleda datoteka sličan je izgledu aplikacija na slici 2.3. Razlikuju se osim u funkciji, i u jednoj dodatnoj stavci, a to je odabiru glavne datoteke. Klikom na datoteku, ona se otvara u novom



Slika 2.9 Izgled pozadine aplikacije i datoteka.

skočnom prozoru i ima funkciju koji ima i inače.

U *.zip* datoteku potrebno je, i vrlo bitno, staviti sve datoteke koje aplikacija koristi. Mogu se koristiti poddirektoriji, ali bitno je da se onda tako koriste i u programskom kodu.

- Kao i aplikacije, datoteke također imaju mogućnost brisanja. Problem je riješen ovim programskim kodom koji je prilično sličan i brisanju aplikacija. Ne koristi se brisanje svih datoteka već samo pojedine:

```
if(isset($_GET['hapus'])){
    $tmp = $_GET['hapus'];
    unlink($tmp);
    $hps = $db->prepare
```

Poglavlje 2. Aplikacija

```
("delete from ".$_SESSION['key']." where path='$tmp'");
if($hps->execute()){
}else{
}
header("location:proba2.php");
}
```

Klikom na gumb za brisanje datoteke i potvrdom brisanja nakon toga, briše se samo datoteka ispod koje smo kliknuli taj koš za smeće. Ona se briše iz direktorija i iz baze podataka, i više nema nikakvog utjecaja na aplikaciju.

- Zadnja stvar koju ovdje vidimo je jedan *checkbox* koji nam služi za odabir glavne datoteke. Nakon *upload-a* ispod svake datoteke nam se pruža mogućnost odabira datoteke za koju želimo da bude glavna. Odabir je moguć sve dok jedna datoteka nije odabrana i dok je ona još uvijek u bazi podataka.

Nakon odabira datoteke, još uvijek imamo mogućnost mijenjanja glavne datoteke. Ako izbrišemo datoteku onda nju više ne možemo staviti za glavnu datoteku dok opet ne napravimo njen *upload*.

Glavna datoteka nam služi tome da nam uštedi jedan klik, odnosno da ne moramo ulaziti u pozadinu aplikacije i tražiti glavnu datoteku te je tamo otvarati, već da možemo otvoriti aplikaciju s početne stranice kao na slici 2.3.

Ako govorimo o aplikacijama koje otvaramo s drugih servera, tada je stvar malo drugačija. S obzirom na to da u bazi podataka imamo samo jednu stavku, ona će automatski postati i glavna datoteka. Programski kod je prilično jednostavan:

```
if(isset($_POST['check'])){
    $hol=$_POST['izvr'];
    $ime = $_SESSION['key'];
    $stmt = $db->prepare("update $ime set flag='0'");
    $stmt ->execute();
    $stm = $db->prepare
    ("update $ime set flag='1' where firstname='$hol'");
    $stm ->execute();
}
```


Poglavlje 2. Aplikacija

```
header("location:proba2.php");  
}
```

Pozadina aplikacije je najbitniji dio aplikacije, bez nje ništa ne bi bilo ispravno. Također treba još reći da su za brisanje, *upload* i odabir glavne datoteke, potrebna administratorska prava.

Poglavlje 3

Baza podataka

Prije svega treba spomenuti da je baza podataka najpotrebnija prilikom prijave korisnika u sustav. Iako se koristi i kod izrada aplikacija i njenih datoteka, nije toliko potreba kao kod prijave.

Ono što je potrebno korisniku aplikacije da napravi prije početka korištenja je napraviti bazu podataka pod nazivom *apps*. Jedino tako će bez greške moći koristiti aplikaciju.

3.1 Konekcija

Nakon što smo napravili bazu podataka ručno, možemo početi pisati programski kod. Ako želimo pristupiti bazi podataka potrebna nam je konekcija s njom.

U *PHP-u* postoji nekoliko načina pristupa bazi podataka, ali za svaku su potrebno sljedeće stvari:

- server: server koji mi koristimo je *localhost*
- username: username za pristup našoj bazi je *root*
- lozinka: našoj bazi podataka nije potrebna lozinka
- ime baze podataka koje smo već spomenuli, *apps*.

Naredbe izgledaju otprilike ovako:

```
$db = new PDO('mysql:host=localhost;dbname=apps','root','');  
$conn = mysqli_connect("localhost","root","","apps");
```

i slično. U nekim trenucima su se javljale greške kod pokušaja povezivanja s bazom pa su se onda koristile i različite naredbe.

3.2 Aplikacije

Aplikacije koje želimo koristiti u našoj aplikaciji moraju biti prvo napravljene u početnom dijelu. Već smo spomenuli kako se to radi, a možemo vidjeti i na slici 2.4. Također aplikacije koje želimo koristiti moramo prikazati na početnoj stranici, a u tome se isto služimo bazom podataka.

3.2.1 Izrada aplikacije

Izrada aplikacije je prilično jednostavna. Korisnik upisuje ime aplikacije koju želi napraviti te se pod tim imenom kreira nova tablica u bazi podataka. Već smo rekli da to ime ne smije sadržavati razmak jer tablica tada neće biti kreirana. Pomoću *sql* naredbe:

```
$sql = "CREATE TABLE ".$_POST['fname']." (  
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
firstname VARCHAR(100) NOT NULL,flag VARCHAR(30) NOT NULL,  
path VARCHAR(100) NOT NULL  
)";
```

stvaramo tablicu sa četiri stupca:

- *id*
- *firstname*
- *flag*
- *path*.

3.2.2 Prikaz aplikacija

Svaka aplikacija prikazana na slici 2.3 pod svojim imenom, koristi podatke iz baze podataka. Sam naziv aplikacije koji piše koristi se iz imena tablice iz baze podataka. Isto tako i točan broj tablica koji su u bazi podataka koristi se za prikaz aplikacija. Jedna tablica se razlikuje, a to je tablica *user* koja nam služi isključivo za prijavu korisnika, međutim nje niti nema u prikazu.

3.3 Datoteke

Datoteke imaju vrlo sličan izgled kao i aplikacije. Prikaz se koristi na isti način, samo što se ne prikazuju sve tablice nego samo određena tablica sa svojim datotekama.

Nakon što se izvrši *upload* datoteka, one se u tablicu upisuju kao tekst. Svaka datoteka ima svoj identifikacijski broj po kojoj je se može prepoznati.

Najbitnija stvar ovdje je ta što se treći stupac(*flag*) koristi za odabir glavne datoteke. Naime, kako smo ranije objasnili, nakon što se odabere glavna datoteka u bazi podataka se promijeni njen stupac. Inicijalno, zastavica na svim datotekama je nula. Kada se odabere glavna datoteka u bazi podataka njena zastavica se promijeni na jedan. Ako se ponovno odabere glavna datoteka, prethodna postaje nula, a nova jedan. Taj slučaj ne vrijedi ako je riječ o aplikacijama korištenim s vanjskih servera. One automatski postaju glavne i jedine datoteke.

Kod prikaza aplikacija tada nam je vrlo jednostavno znati koja je glavna datoteka zato što samo jedna može imati broj jedan, sve druge su nula.

3.4 Logiranje

Prijava korisnika je najbitniji dio baze podataka. Bez prijave, korisnik bi samo mogao koristiti aplikacije, ali ne bi mogao upravljati njima.

Tablica za prijavu korisnika zove se *user* i ima tri stupca koja moraju biti ručno napravljena u bazi podataka:

- *id* koji nam nije toliko ni potreban iz razloga što dopuštamo samo jednom

Poglavlje 3. Baza podataka

korisniku da bude u bazi podataka.

- *uname* koji daje ime korisniku, u ovom slučaju to je *admin*
- *pass* koji daje lozinku korisniku a nama je to *admin01*

Ako se ne koristi ova baza podataka nego se kreće s vlastitom, potrebno je napraviti novu *.php* datoteku u kojoj će se *hard* kodirati korisnik. Stoga preporuka je da se koristi ova baza podataka.

Poglavlje 4

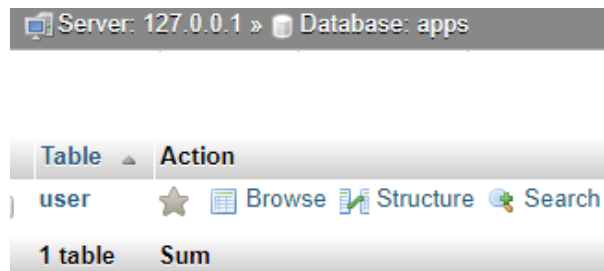
Primjer korištenja aplikacije

U ovom dijelu na dva primjera pokazat ćemo kako naša aplikacija zapravo radi. Koji su najbitniji koraci te što je potrebno napraviti prije početka korištenja aplikacije. Primjeri će biti za lokalni *upload* aplikacija te za korištenje aplikacija s drugih servera.

4.1 Baza podataka

Prvi korak za korištenje aplikacije je osposobljavanje baze podataka. Potrebno je stvoriti novu bazu podataka pod nazivom *apps*. U slučaju našeg lokalnog servera, pod *localhost/phpmyadmin* napravili smo bazu podataka imena *apps*.

Sljedeći korak je napraviti *import* datoteke *apps.sql*. *Import* je nužan za napraviti zato što tablica *user*, kao što je objašnjeno u prethodnom poglavlju, sadrži enkriptirane podatke. Bilo bi potrebno programski rješavati problem ako se ne bi radio *Import*. Nakon što smo napravili *Import*, kao što je prikazano na slici 4.1, dobivamo početnu bazu podataka s jedinom tablicom *user* koja sadrži samo jednog korisnika (administratora).



Slika 4.1 Izgled baze podataka nakon početnog sređivanja.

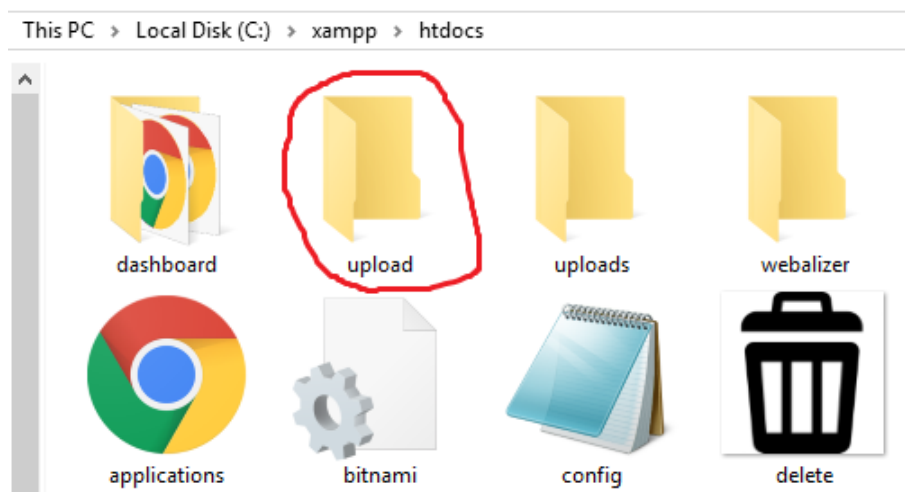
4.2 Upload direktorij

Sljedeća bitna stvar u prvim koracima je stvaranje *upload* direktorija. On nam služi kako bi u njega mogli kreirati nove poddirektorije koji nose naziv aplikacija, a sadrže sve datoteke aplikacije koje želimo koristiti.

Kako se *Xampp* paket koristio se za izradu naše aplikacije, a aplikacija radi samo lokalno, potrebno je unutar njegovog paketa stvoriti *upload* direktorij. Početna datoteka nalazi se na stazi *xampp/htdocs/*, a na istom mjestu moramo napraviti i *upload* direktorij. Njegova staza sada će biti *xampp/htdocs/upload/*. Na slici 4.2 možete vidjeti napravljeni direktorij pod spomenutom stazom.

Usporedbe radi, ako se ne koristi *Xampp*, već se koristi *Wamp*, staza *xampp/htdocs/upload/* bit će jednaka *wamp/www/upload/*.

S ovim dijelom završavamo pripremanje aplikacije i možemo je početi koristiti.



Slika 4.2 Izgled *upload* direktorija sa stazom na kojoj se nalazi.

4.3 Prijava

Otvaranjem aplikacije vidimo praznu stranicu s velikim gumbom za dodavanje novih aplikacija, ali nemamo ovlasti za to. Potrebna nam je prijava. Kao što smo objasnili u poglavlju dva, a vidimo i na slici 2.6 otvorimo modalni *container* i vršimo prijavu. Pod korisničko ime upisuje se jedino korisničko ime koje se nalazi u bazi podataka, a to je *admin*. Ispod toga *admin* mora upisati svoju lozinku koja glasi *admin01*. Ovo je ključan korak ako želimo koristiti našu aplikaciju. Ako korisnik nije prijavljen u sustav, ima mogućnost samo korištenja aplikacija koje su već ubačene.

4.4 Kreiranje aplikacija

Kreiranje aplikacije sastoji se od izrade direktorija, tablice i *upload-a .zip* datoteke.

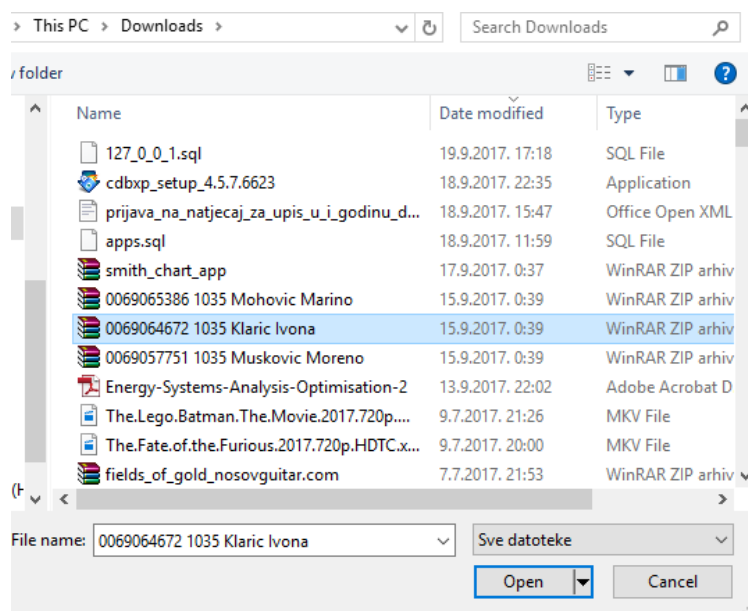
4.4.1 Lokalna aplikacija

Na slici 2.4 vidimo *textbox* u koji upisujemo ime koje želimo dati aplikaciji. Za ovaj prvi primjer napraviti ćemo aplikaciju pod nazivom *test*. Unutar *upload* direktorija,

Poglavlje 4. Primjer korištenja aplikacije

napravio se novi poddirektorij pod nazivom *test*. Također, u bazi podataka kreirana je nova tablica pod nazivom *test*.

Unutar *testa*, u pozadini aplikacije, napraviti ćemo *upload .zip* datoteke kao što je prikazano na slici 4.3. *Upload* je prvo kopirao datoteku i prebacio je u *test* direktorij.



Slika 4.3 Odabir *.zip* datoteke.

Tamo ju je raspakirao i prikazao u pozadini aplikacije sve datoteke koje su se unutra nalazile (slika 2.9), a sada se nalaze u *test* direktoriju. Zatim je svaku datoteku pod svojim nazivom upisao u bazu podataka, zajedno sa stazom koja vodi do datoteke (slika 4.4). U ovom trenutku već možemo otvarati datoteke koje su nam prikazane i koristiti ih. Ali još jedan dio je ostao koji nam koristi da bi mogli otvoriti aplikaciju s početne stranice. To je odabir glavne datoteke.

Odaberemo glavnu datoteku (za ovaj primjer odabrat ćemo *index.html* datoteku), te se možemo vratiti na početnu stranicu. S nje imamo mogućnost, pritiskom na *hyperlink* otvoriti datoteku, koja će nam otvoriti aplikaciju koju vidimo na slici 4.5.

Poglavlje 4. Primjer korištenja aplikacije

id	firstname	flag	path
1	bez_dist_fb.html	0	upload/test/zavrsni/bez_dist_fb.html
2	bez_dist_fk.html	0	upload/test/zavrsni/bez_dist_fk.html
3	bez_dist_ka.html	0	upload/test/zavrsni/bez_dist_ka.html
4	bez_dist_kp.html	0	upload/test/zavrsni/bez_dist_kp.html
5	bez_distorzije.html	0	upload/test/zavrsni/bez_distorzije.html
6	bez_fb.html	0	upload/test/zavrsni/bez_fb.html
7	bez_fk.html	0	upload/test/zavrsni/bez_fk.html
8	bez_gubitaka.html	0	upload/test/zavrsni/bez_gubitaka.html
9	bez_ka.html	0	upload/test/zavrsni/bez_ka.html
10	bez_ki.html	0	upload/test/zavrsni/bez_ki.html
11	bez_kp.html	0	upload/test/zavrsni/bez_kp.html
12	bez_kr.html	0	upload/test/zavrsni/bez_kr.html
13	bez_kt.html	0	upload/test/zavrsni/bez_kt.html
14	bez_vd.html	0	upload/test/zavrsni/bez_vd.html

Slika 4.4 Izgled *test* tablice nakon izvršenog *upload-a*.

The screenshot shows a web application interface for calculating transmission line parameters. It features a top navigation bar with tabs: 'Prijenosna linija', 'Prijenosna linija bez gubitaka', 'Prijenosna linija s malim gubicima', 'Prijenosna linija bez distorzije', and 'Posebni slučajevi'. Below this, there are sub-tabs: 'Koaksijalna linija', 'Sustav 2 bliska vodiča', and 'Sustav paralelnih ploča'. The main area contains a grid of input fields for various parameters: 'a' (mm), 'b' (mm), 'R_s' (Ω), 'f' (MHz), 'p_i', 'ε', 'tan δ', 'L' (Hm), 'C' (Fm), 'R' (Ωm), 'G' (Sm), 'Hm', 'Fm', 'Qm', and 'Sm'. At the bottom, there are buttons for 'Izračunaj' and 'Poništi'.

Slika 4.5 Izgled *upload-ane* aplikacije.

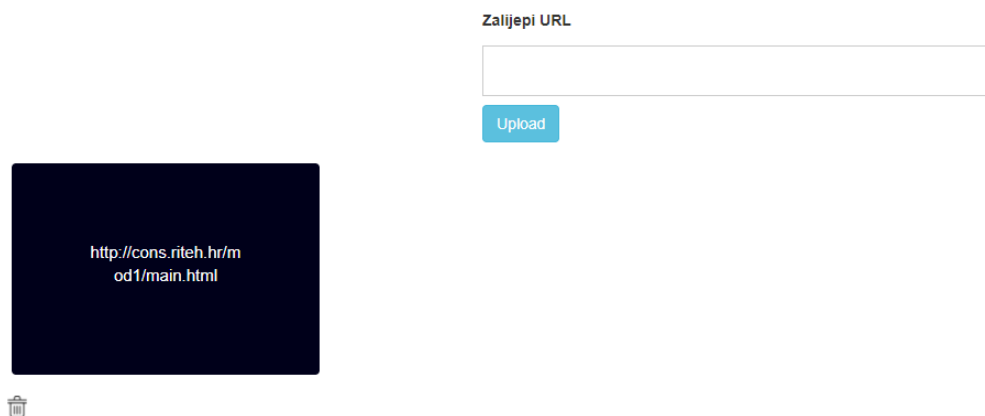
4.4.2 Aplikacija sa drugog servera

Ovaj primjer se ne razlikuje puno od onog prethodnog, međutim ima nekoliko razlika. Najveća je razlika u tome, što se kod ovakvih aplikacija ne koriste datoteke. Na slici 2.8 pokazano je mjesto na koje se sa stranog servera upisuje *URL* koji vodi do te aplikacije. Kao u prvom primjeru, kreira se poddirektorij i tablica pod imenom koji smo zadali. U ovom primjeru aplikaciju ćemo nazvati *test2*.

Nakon izrade direktorija i tablice, otvaramo pozadinu aplikacije i kopiramo *URL* u

Poglavlje 4. Primjer korištenja aplikacije

za to namijenjeno područje. Stvara se podatak u bazi podataka koji ima naziv svog *URL-a* prikazanog na stranici kao na slici 4.6.



Slika 4.6 Izgled datoteke aplikacije s drugog servera.

Još jedna razlika je u tome što se u ovom primjeru ne koristi biranje glavne datoteke, već je veza na stranicu glavna datoteka. Tako da čim se napravi *upload*, na početnoj stranici se prikazuje veza koja nas vodi na aplikaciju s drugog servera koju iz ovog primjera možete vidjeti na slici 4.7.

Modulation WebApp

Mobilne Komunikacije

Konfigurator

Modulacija: Tip modulacije ▼

Binarna informacija:

≤ 10 bitova

Logika kodiranja:

☒ Pozitivna ☐ Negativna

Amplituda: Amplituda ▼

Slika 4.7 Izgled aplikacije s drugog servera.

Poglavlje 5

Zaključak

U ovom radu opisuje se postupak razvijanja rješenja koji je rezultirao aplikacijom *AppJoiner*. U radu su detaljno opisani načini na koji se rješavao pojedini problem.

Cilj je bio pokazati koliko je korisno imati više aplikacija na istom mjestu. Pogotovo kada govorimo o aplikacijama koje se pokreću s lokalnog računala odnosno aplikacije nisu dostupne javnosti.

Također je u radu slikovito prikazan svaki dio aplikacije. Opisani su zasebni dijelovi, koja je njihova uloga te kako su svi zajedno na kraju stvorili cjelinu. Opisano je koja je bila uloga baze podataka, te je dan primjer korištenja aplikacije.

Bibliografija

- [1] Notepad++, s Interneta <https://notepad-plus-plus.org/>, srpanj 2017.
- [2] Xampp, s Interneta <https://www.apachefriends.org/index.html>, srpanj 2017.
- [3] PHP,Html,CSS, s Interneta <https://www.w3schools.com/>, srpanj 2017.
- [4] Bootstrap, s Interneta <http://getbootstrap.com/>, srpanj 2017.

Sažetak

U ovom projektu izrađena je web aplikacija pod nazivom *AppJoiner*. Aplikacija ima ulogu kišobrana, odnosno ima mogućnost pokretanja ostalih, zasebnih, web aplikacija. Administrator ima ovlasti ubacivanja i brisanja aplikacija, a svi ostali mogu samo izvršavati aplikacije. Također svi imaju mogućnost pretržavanja aplikacija i pregled njenih datoteka. Mogu se koristiti i aplikacije s ostalih servera koje su dostupne *online*.

Ključne riječi — web aplikacije

Abstract

In this project web application *AppJoiner* was created. The application has an umbrella role, ie it has the ability to launch other, separate, web applications. Administrator has the authority to add and delete apps, and everyone else can only run apps. All of them also have the ability to search applications and view their files. Applications from other servers can also be used.

Keywords — web applications