Izrada društvene mreže

Huljak, Marko

Undergraduate thesis / Završni rad

2017

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:200:940688

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

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

Sveučilišni studij računarstva

IZRADA DRUŠTVENE MREŽE

Završni rad

Marko Huljak

Osijek, 2017.

SADRŽAJ

1. UVOD 2
1.1 Zadatak završnog rada 2
2. FRONT-END TEHNOLOGIJE 3
2.1 HTML 3
2.2 CSS 5
3. BACK-END TEHNOLOGIJE 6
3.1 PHP 6
3.2 MySQL 7
3.2.1 PhpMyAdmin 8
4. IZRADA DRUŠTVENE MREŽE 9
4.1 Baza podataka 10
4.2 Korisnička sučelja 11
4.2.1 Poruke 16
4.2.2 Korisničke objave 17
4.2.3 Korisnički profil 19
4.3 Opis koda 20
5. ZAKLJUČAK 30
LITERATURA 31
SAŽETAK 32
ABSTRACT 33
ŽIVOTOPIS 34
PRILOZI 35

1. UVOD

U ovom završnom radu usredotočiti ćemo se na izradu sustava za dijeljenje sadržaja sa drugim korisnicima. Temeljna ideja je napraviti društvenu mrežu pristojnog dizajna i jednostavnu za korištenje, sa svim osnovnim funkcijama i mogućnostima. Vidljivi dio mreže opisan i dizajniran je front-end tehnologijama, točnije, markup programskim jezikom HTML i stilskim jezikom CSS. Za programski, nevidljivi dio korišten je PHP programski jezik, dok je za izradu baze podataka korišten MySQL i sve to uz pomoć XAMPP programskog paketa. U poglavlju *Društvene mreže*, prije svega, opisana je definicija, uporaba, te uloga društvenih mreža, zatim se spominje njihov utjecaj na društvo i poslovanje, te pozitivne i negativne strane istih. Pojašnjeni su i pojmovi privatnost i sigurnost korištenja društvenih mreža, te su ukratko opisane najpopularnije društvene mreže današnjice. U nastavku je uz jednostavne primjere opisan skup tehnologija korištenih u razvoju mreže, te je zatim detaljno opisana izrada društvene mreže.

1.1 Zadatak završnog rada

Izraditi društvenu mrežu u kojoj korisnik sa svojim prijateljima može dijeliti objave, slike i sadržaje. Sadržaj na profilu korisnika može biti javan ili podijeljen samo sa pojedincima i grupama. Korisnici mogu komunicirati putem chat-a.

2. FRONT-END TEHNOLOGIJE

Front-end odnosno prednji, vidljivi je dio web stranice s kojim korisnici izravno stupaju u interakciju. Front-end je sve ono što je vidljivo prilikom surfanja Internetom, od fontova i boja do padajućih izbornika i klizača. Stručnije rečeno, to je mješavina HTML, CSS i Javascript kôda u prozoru web pretraživača u vizualnom obliku. Za raliku od back-end tehnologije koja se izvodi na serveru (PHP, MySQL), front-end tehnologija se izvodi na korisnikovom računalu, tzv. client-side. Pokretanje bilo koje internet stranice zahtjeva minimalne hardverske zahtjeve jer se radi o malom zauzeću radne memorije i snage CPU-a, osim ako se radi o Flash animacijama. No, Flash animacije se više ne koriste zbog velikog interesa za Javascript-om ili točnije jQuery-jem, koji za razliku od Flasha radi na svakom uređaju i kojemu je potrebno manje vrijeme učitavanja sadržaja i zahtjeva manje hardverske zahtjeve.

2.1 HTML

HTML (engl. HyperText Markup Language) je najznačajniji jezik svakog internet preglednika koji služi za oblikovanje i uređenje sadržaja internet stranica. Činjenica da se njime ne može izvršiti nikakva zadaća niti logika, govori nam da HTML zapravo nije programski jezik. Odnosno, to je prezentacijski jezik za označavanje kojim se određuje sadržaj, struktura i funkcija HTML dokumenta. Kao i bilo koji drugi dokument, HTML dokument ima određenu strukturu. HTML oznaka (engl. tag) je oznaka kojom se definira početak i kraj sadržaja svakog HTML elementa. HTML oznake se na neki način ponašaju poput spremnika, kontejnera (engl. container) koji govore koja se vrsta informacije nalazi u istom. Svaka HTML oznaka se sastoji od identifikatora i od dvije šiljaste zagrade (znakovi < i >). Identifikator koji se nalazi unutar navedenih zagrada nam govori o kojem se točno HTML elementu radi. Početni HTML element unutar zagrada sadrži identifikator, dok završni element pored identifikatora, sadrži i kosu crtu, odnosno oznaku koja pokazuje da je riječ o završnom, a ne početnom elementu. Unutar početne oznake nekih HTML elemenata mogu se nalaziti dodatna svojstva, atributi koji većim brojem informacija definiraju sadržaj elementa. Atributi se sastoje od dva dijela, imena i vrijednosti koji su odvojeni znakom jednakosti. Svaki HTML dokument bi trebao započeti sa DOCTYPE deklaracijom verzije koja opisuje koja se inačica HTML-a jezika koristi u dokumentu. DOCTYPE deklaracija nije HTML

element, već instrukcija pregledniku koja se nalazi na samom početku svakog HTML dokumenta. Sav sadržaj svake web stranice se nalazi unutar elementa koji počinje oznakom <html>, a završava </html>. Obavezno ga je navesti unutar svakog HTML dokumenta kako bi web preglednik znao prepoznati da je riječ o HTML dokumentu. Unutar <html> elementa, prvo što se navodi je element koji sadrži informacije o stranici. Govorimo o <head> elementu koji se zapravo ne prikazuje kao dio sadržaja stranice, već predstavlja zaglavlje HTML dokumenta. Naslov web stranice se nalazi u elementu <title> koji se smješta unutar zaglavlja. Kao što je već navedeno, naslov stranice se ne prikazuje kao dio sadržaja, već se prikazuje na naslovnoj traci preglednika. Nakon <head> elementa navodi se element <body> koji prikazuje sav sadržaj koji je prikazan u web pregledniku. HTML podržava i pisanje komentara koji se otvaraju sa oznakom <!--, a zatvaraju sa -->. HTML dokument pohranjuje se lokalno na računalu kao datoteka s ekstenzijom .html [1]. U nastavku možemo vidjeti najjednostavniji primjer HTML dokumenta.

```
<!DOCTYPE html>
<html>
<head>
<title>Naziv stranice</title>
</head>
<body>
<p>Dobrodošli na stranicu</p>
</body>
</html>
```

U izradi društvene mreže je korišten HTML5 radi semantičkih elemenata zbog lakšeg pisanja i razumijevanja koda.

2.2 CSS

CSS (engl. Cascading Style Sheets), odnosno kaskadni stilovi, jednostavan su mehanizam za dodavanje stilova. Riječ cascading odnosi se na kaskadnu primjenu CSS pravila koja mogu biti primijenjena na sve elemente, samo na neke elemente ili samo na točno određeni element. Oblikovanje izgleda web stranice moguće je bilo ostvariti i prije pojave CSS-a, pomoću HTML-a, odnosno unutar <style> elementa. No, tada se stvara problem miješanja strukture i sadržaja što narušava preglednost HTML koda čija je jedina svrha opisati, prezentirati web stranicu. Još jedna mana toga je ta da bi se HTML kod morao ponavljati iznova na svakom elementu i na svakoj stranici, što otežava posao programera. CSS nastoji ukloniti upravo te probleme, odvajanjem stilova od prezentacijskog koda u "Style Sheets", eksterne datoteke sa .css extenzijom. Kao što je već navedeno, style sheet u CSS-u sastoji se od nekoliko pravila. Svako pravilo sastoji se od selektora i deklaracijskog bloka. Selektor određuje element na koji se stilsko pravilo odnosi dok deklaracija određuje kako izgleda sadržaj opisan CSS-om. Sintaksa za stilska pravila uvijek slijedi sljedeći uzorak: selektor {deklaracija;}. Deklaracija se dijeli na dvije stavke: Properties (aspekti kako da računalo prikaže tekst i grafiku) i Values (podaci koji određuju kao želimo da tekst i slike izgledaju na našoj stranici). Properties se od vrijednosti (value) u deklaraciji odvaja dvotočkom, a svaka deklaracija završava s točkom-zarez: selector {property: value;} [2]. Za primjer ćemo uzeti p element (paragraf) kojim ćemo pomoću CSS-a definirati boju i veličinu fonta.

HTML dokument

```
<p class= “pozdrav“ > Hello World </p>
```

CSS dokument

```
.pozdrav { color:red; font-size:32px; }
```

U navedenom CSS kodu smo koristili selektor .pozdrav, te mu dodali crvenu boju i veličinu fonta 32px. Kao selektor smo mogli koristiti i p element, te bismo dobili isti rezultat, s razlikom da bi to utjecalo na sve p elemente.

CSS dokument

```
P { color:red; font-size:32px; }
```

3. BACK-END TEHNOLOGIJE

Back-end predstavlja razvoj svega onoga što se nalazi u pozadini web stranice, web aplikacije i odvija se bez uvida krajnjeg korisnika, posjetioca stranice. Odnosno to je nevidljivi dio web stranice koji kontrolira tok podataka između aplikacija i preglednika. U ovoj tehnologiji, pristup web stranici imaju samo administratori i služi im za upravljanje web serverom, korisnicima i sadržajem nekog web sjedišta. Pristup u back-end nekog web servisa je najčešće osiguran korisničkim imenom i lozinkom. Kao što je već rečeno, back-end je usmjeren samo prema serverskoj strani (engl. server-side), što znači da se svi poslovi odvijaju na serveru a podaci se šalju u preglednik, koji prikazuje te podatke. Uvjet da se na osobnom računalu razvijaju web stranice sa back-end internet tehnologijama je instalirati programsko sučelje XAMPP koje simulira internet server, točnije Apache HTTP server. U izradi društvene mreže, od back-end tehnologije, koristiti ćemo PHP programski jezik i MySQL baze podataka.

3.1 PHP

PHP svoj naziv temelji na rekurzivnoj definiciji, *Hypertext preprocessor*. Skriptni je jezik koji izvršava skripte na serverskoj strani (engl. server-side scripting). Uzimajući PHP kod kao ulaz, kreira logiku dinamičkih web stranica kao izlaz. PHP ima razne načine korištenja. Moguće ga je koristiti kao podršku web aplikacija, kao konzolnu aplikaciju za pisanje skripti sa komandne linije, te ga je moguće koristiti za pisanje aplikacija u PHP-u koristeći OpenGL biblioteke za trodimenzionalnu vizualizaciju i sl. Zbog svoje sintakse koju je naslijedio od programskog jezika C i zbog svoje jednostavnosti, stekao je veliku popularnost. Ima mogućnost za čisto proceduralno programiranje, ali isto tako omogućava i korištenje klasa i drugih koncepata objektno orijentiranog programiranja. Budući da je besplatan, neovisan o operacijskim sustavima, te ima podršku za korištenje širokom paletom baza podataka, PHP se smatra prvim izborom kompanija za kreiranje vlastitih mrežnih sustava. Izuzetno je jednostavno nadograditi *web* aplikaciju korištenjem specifičnih ekstenzija poput *mysqli*, apstraktnog PDO (*PHP Data Objects*) sloja ili spajanjem na bilo koju bazu koja podržava ODBC (*Open Database Connection*) standard [3].

PHP se izvršava u dokumentu unutar svojih oznaka, a primjeri ispisa teksta su prikazani u nastavku.

```
<?php
echo 'Pozdrav svima';
?>
i
<?php
printr ('Pozdrav svima');
?>
```

Rezultat ovih primjera je isti, s time da smo koristili različite funkcije za prikaz teksta.

3.2 MySQL

MySQL je sustav za upravljanje relacijskim bazama podataka, odnosno client/server sustav koji se sastoji od multi-threaded SQL servera, nekoliko dodatnih utility programa koji služe za olakšavanje administracije BP. Također sadrži i dodatne programe koji su potrebni za rad servera. Srce sistema naravno čini MySQL server, on je ustvari "menadžer" sustava BP. Kroz njega prolaze sve instrukcije korisnika koje vrše upravljanje nad bazom podataka. MySQL nam dozvoljava kreiranje većeg broja baza podataka, te upravljanje podacima iz više baza podataka unutar jednog projekta. Na serveru je svakom korisničkom računu moguće dodijeliti određena prava pristupa na cijeli server ili pojedine baze. Tako imamo prava pristupa postojećim bazama, prava za stvaranje novih baza, te pravo uređivanja postojećih baza, odnosno pravo izmjene i unosa podataka. Relacijska baza podataka je baza podataka koja podatke radije sprema u odvojene tablice nego sve u jedno. Ovo daje brzinu i fleksibilnost bazama podataka. Tablice su povezane definiranim relacijama što omogućuje kombiniranje podataka iz nekoliko tablica u slučaju da postoji zahtjev za to. MySQL kao sustav može raditi na mnogo različitih operacijskih sustava, a najčešće se koristi na Apache web-serverima u kombinaciji sa PHP-om. Radi se o software-u kojem se na sličan način pristupanja web, HTTP serverima pristupa preko mreže, sa jedinom razlikom da se prilikom pristupa MySQL-u zahtjeva unošenje korisničkog imena i lozinke. Prilikom instalacije MySQL-a, stvara superadministrator (engl. root) kojem su dozvoljena sva administracijska prava. Konfiguracijska datoteka MySQL-a je /etc/mysql/my.cnf. U njoj se mogu podesiti osnovni parametri MySQL sustava. MySQL je također stekao veliku popularnost upravo radi toga što podržava sve operacijske

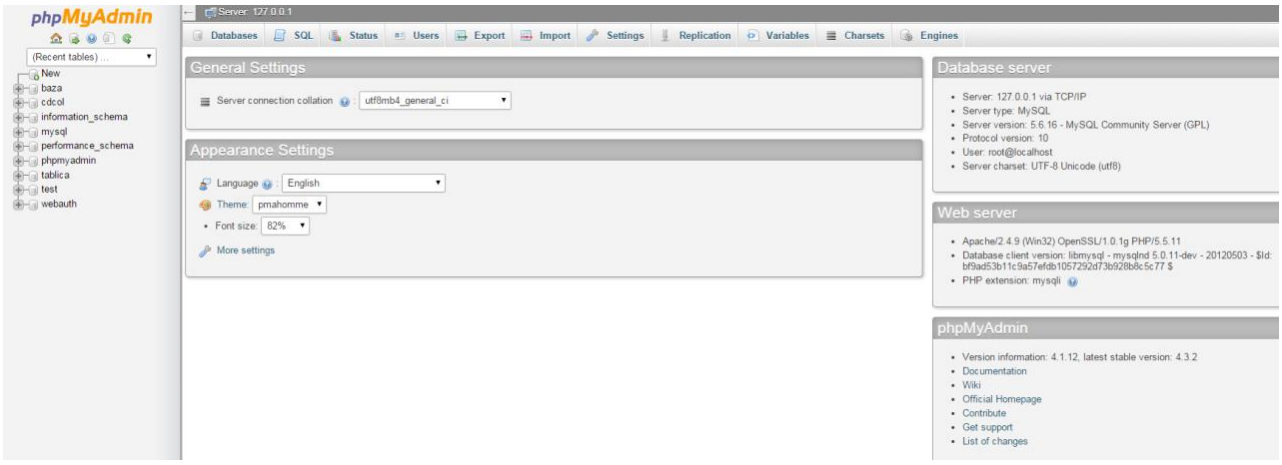
sustave, i zbog toga što se izdaje pod GPL licencom, što znači da je besplatan za kućnu upotrebu [4]. Primjer izrade baze podataka se nalazi u nastavku.

```
CREATE DATABASE baza_podataka;
CREATE TABLE korisnici(
id_korisnika INT(5),
korisnicko_ime VARCHAR(20),
lozinka VARCHAR(20)
);
```

Navedene naredbe predstavljaju bazu podataka pod nazivom "baza_podataka" koja sadrži tablicu "korisnici". Tablica "korisnici" sadrži 3 kolone: "id_korisnika", "korisnicko_ime" i "lozinka".

3.2.1 PhpMyAdmin

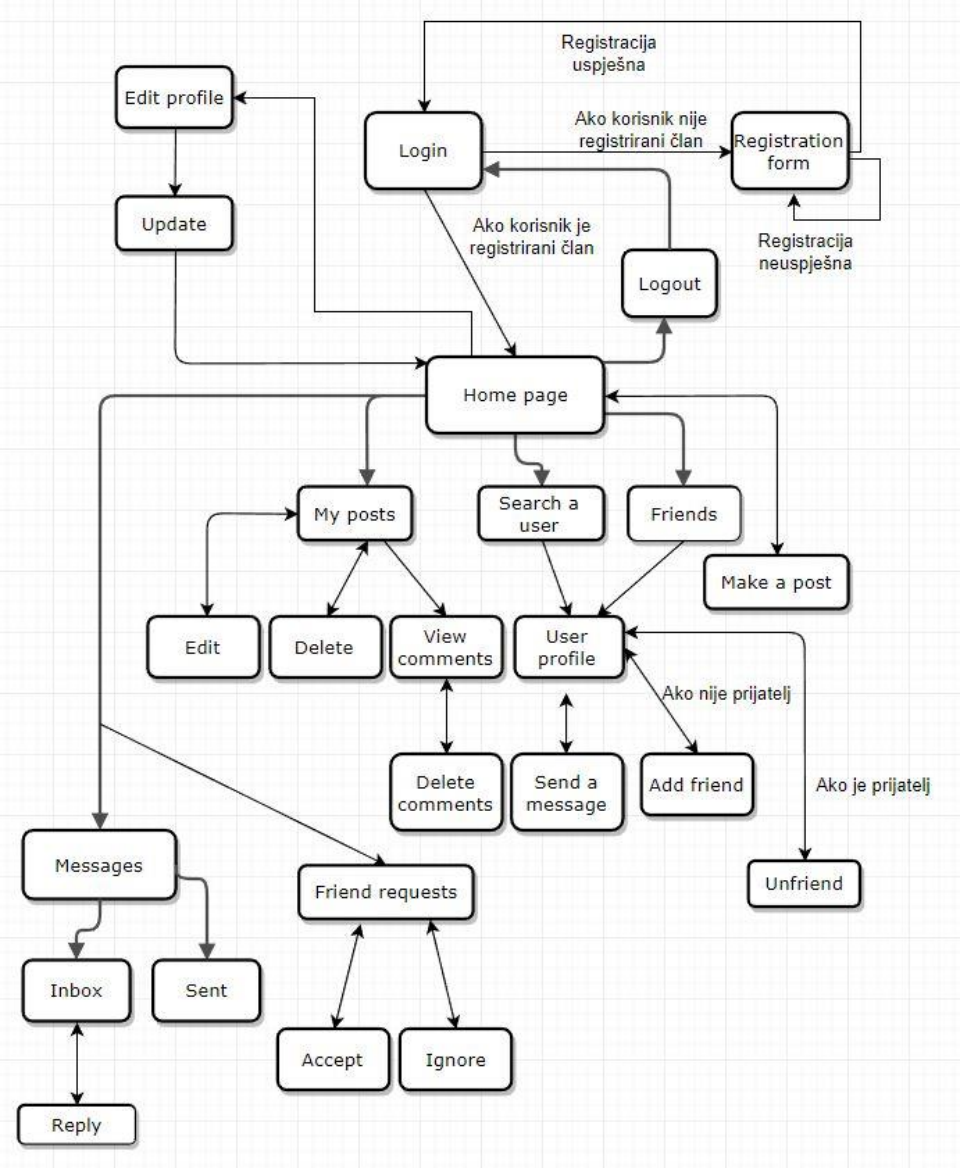
PhpMyAdmin je besplatno, open-source grafičko korisničko sučelje napisano u PHP programskom jeziku. Koristi se za upravljanje podacima i administraciju unutar MySQL baze podataka izravno u Internet pregledniku. Znači, omogućeno je jednostavno, bez poznavanja kôda, upravljanje bazama podataka, tablicama, poljima, indeksima, korisnicima, dozvolama pristupa samim bazama, izvršavati vlastite upite na bazu i još mnogo drugih stvari. Phpmyadmin se pokreće na serverskoj strani, što znači da dolazi unutar programskog paketa XAMPP.



Sl.3.1. Izgled Phpmyadmin aplikacije

4. IZRADA DRUŠTVENE MREŽE

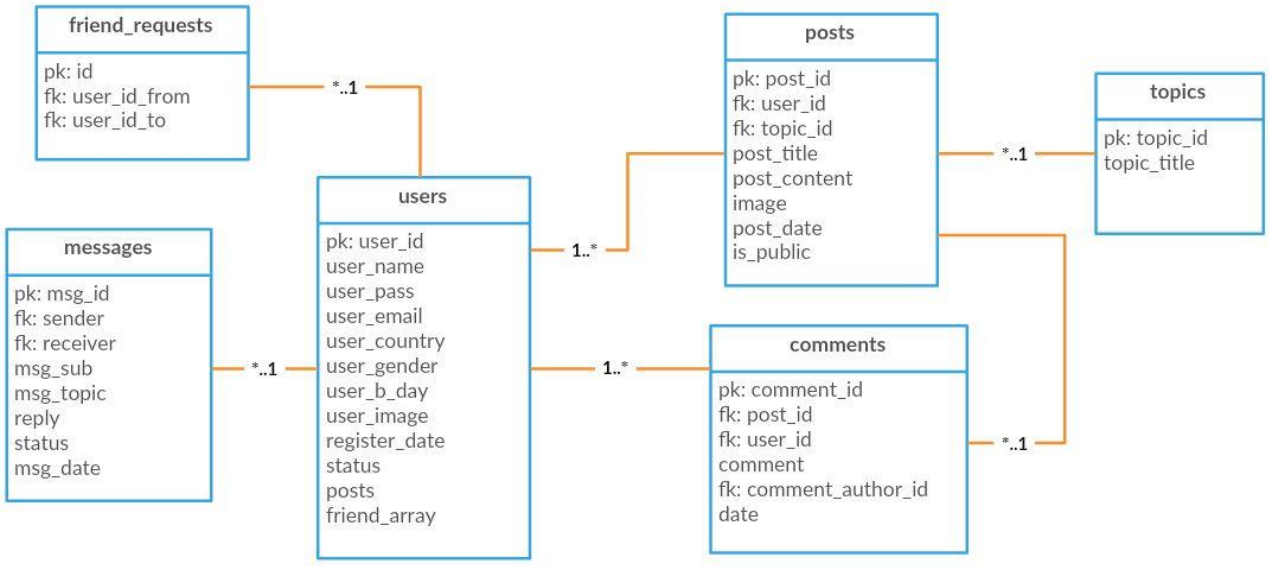
Prije samoga početka izrade društvene mreže, najbitniji parametar je imati dobar plan za izradu, odnosno dijagram toka. Dijagram započinje sa zahtjevom da se korisnik registrira, unese svoje podatke, te da se sa istom e-mail adresom i lozinkom prijavi. Nakon uspješne autentifikacije, pokazuje se naslovnica (engl. home page) na kojoj se prikazuje objavljeni sadržaj, te sa koje se može pristupiti raznim mogućnostima kao što su uređivanje profila, čitanje i slanje poruka, dodavanje kao i uklanjanje prijatelja, itd.



Sl.4.1. Dijagram toka društvene mreže

4.1 Baza podataka

Sljedeći korak je izrada baze podataka. Baza podataka se sastoji od 6 tablica koje imaju funkcijsku ulogu u izradi mreže. Tako, u tablici „users“ se nalazi popis svih korisnika koji imaju mogućnost upravljanja sadržaja na stranici, u tablici „posts“ se nalaze svi objavljeni sadržaji podijeljeni od strane korisnika, a mogu biti javni ili privatni, ovisno o želji korisnika. U tablici „comments“ se nalaze komentari svih objava koji su stranim ključevima povezani sa glavnim ključevima prethodno spomenutim tablicama. Zatim, imamo tablicu „topics“ koja sadrži teme kojima pripadaju određene objave. Tablica „friend_requests“ pomoću stranog ključa koji je ujedno primarni ključ tablice „users“ prikazuje zahtjeve za prijateljstvo između korisnika u kojima se točno vidi koji korisnik kojem šalje zahtjev. Također imamo i tablicu „messages“ u kojoj se spremaju poruke na isti način kao i u tablici „friend_requests“ pomoću stranih ključeva tablice korisnika, dakle zna se koji korisnik kojem korisniku šalje poruku i to pod točno određeno vrijeme.

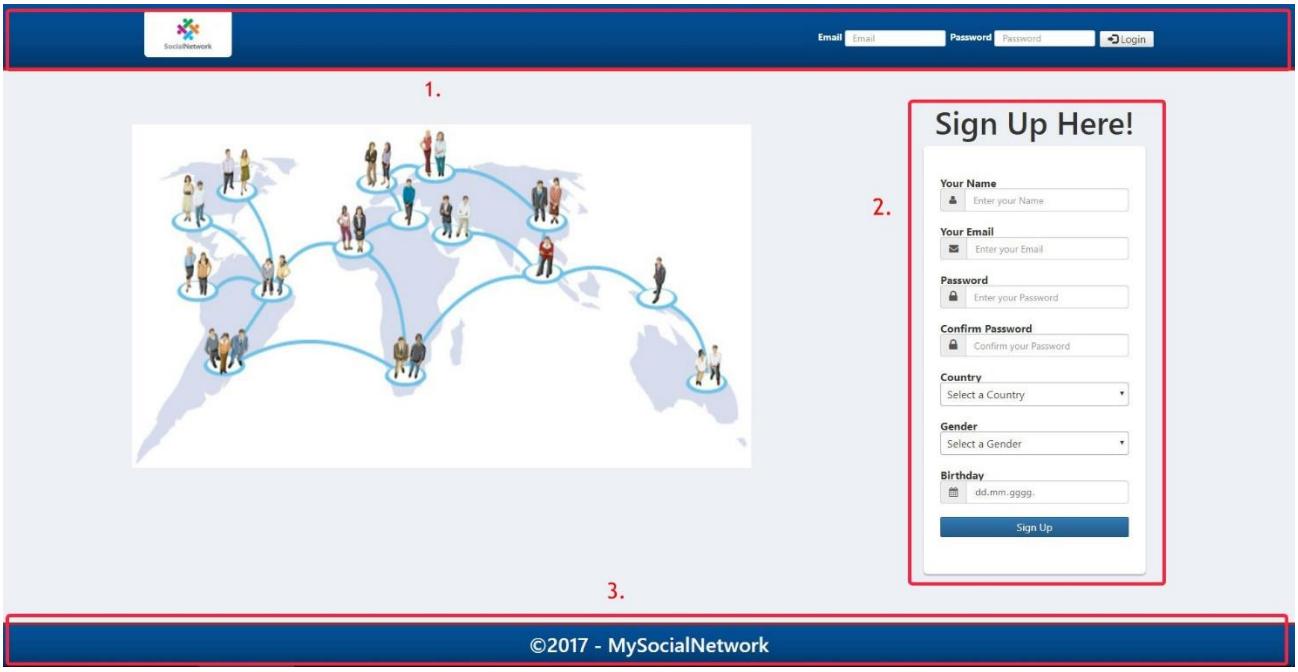


Sl.4.2 UML dijagram društvene mreže

Nakon izrade baze podataka u programskom paketu XAMPP pomoću sučelja za upravljanje podacima, točnije PhpMyAdmin, imamo uvid koji nam podaci trebaju za izradu mreže [5].

4.2 Korisnička sučelja

Nakon baze podataka, prvo je potrebno izraditi, tj. dizajnirati pojedino korisničko sučelje. Za izradu sučelja koristit ćemo HTML i CSS te front-end okvir (engl. framework) „Bootstrap“ [6]. Izgled početne stranice se vidi na sl.4.3.



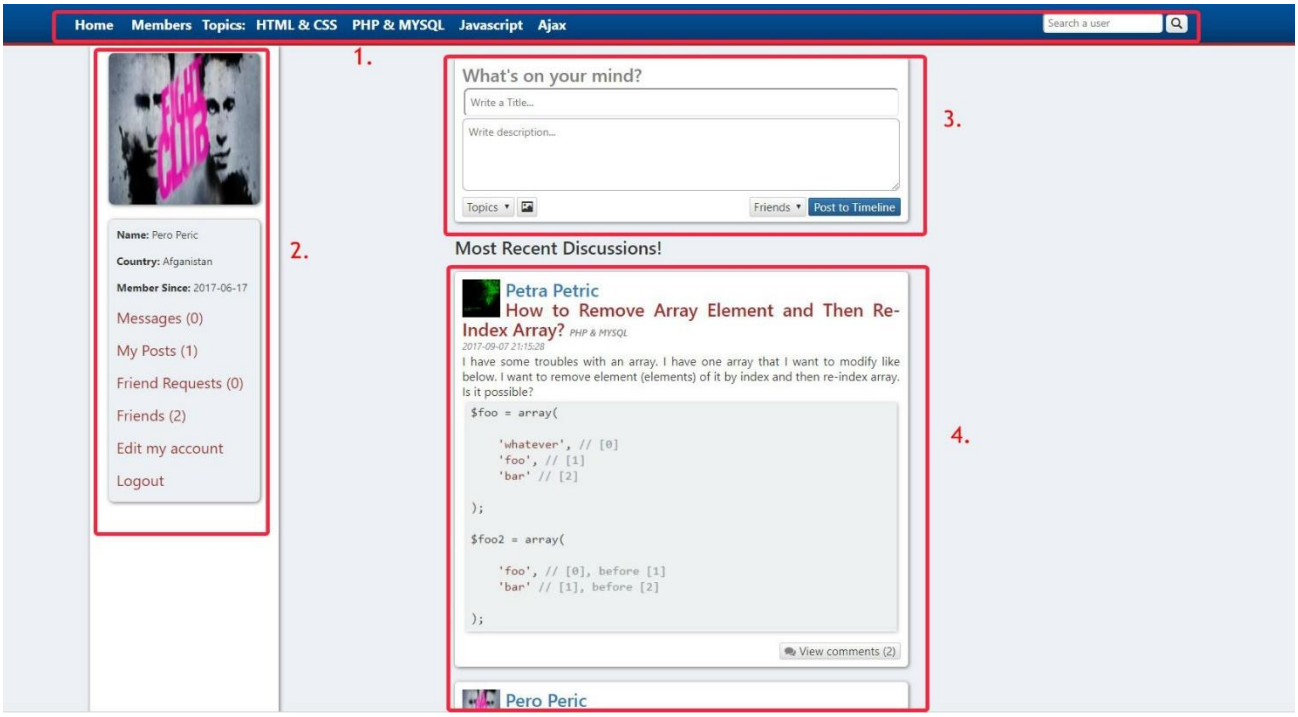
Sl.4.3 Izgled početne stranice

Na stranici su označeni dijelovi stranice sa rednim brojevima. U nastavku se nalazi opis pojedinog bloka:

1. Zaglavlje stranice (engl. header),
2. Registracijska forma,
3. Podnožje stranice (engl. footer)

Zaglavlje stranice sadrži „Login“ formu u koju korisnik, ako je već registriran, unosi svoju e-mail adresu i odgovarajuću lozinku te nakon uspješne prijave ima pristup društvenoj mreži. Registracijska forma nalazi se u glavnom sadržajnom bloku (engl. container). Ona omogućuje registriranje korisnika tako što korisnik unosi svoje podatke kao što su ime, e-mail adresa, lozinka, spol, itd. „Footer“, odnosno donji dio stranice sadrži obično dodatne i manje važne sadržaje i/ili

linkove, npr. informacije o privatnosti, autorskim pravima, odricanju, itd. Nakon uspješne autentifikacije, korisniku se prikazuje profil stranica. Izgled naslovnice je prikazan na slici 4.4.

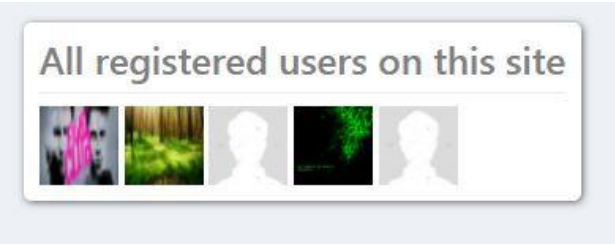


Sl.4.4 Izgled profil stranice

Na stranici su označeni dijelovi stranice sa rednim brojevima. U nastavku se nalazi opis pojedinog bloka:

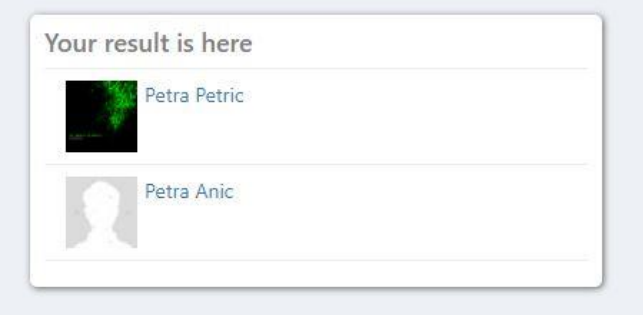
1. Navigacijska traka (engl. navigation bar)
2. Bočna traka (engl. side bar),
3. Forma za objavljivanje sadržaja
4. Blok za prikazivanje objavljenog sadržaja

Navigacijska traka sadrži popis tema koju korisnik ako odabere, prikazat će mu se samo objave koje pripadaju odabranoj temi. Zatim sadrži „Home“ link koji prilikom klika, vraća korisnika na profil stranicu na kojoj su prikazane objave svih tema, te „Members“ link koji prikazuje sve registrirane korisnike kao što je prikazano na sl.4.5.



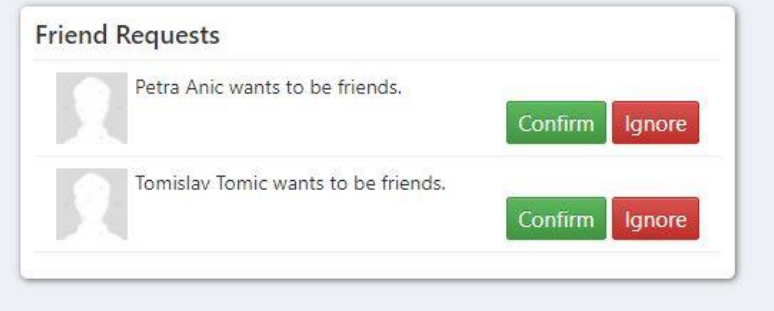
Sl.4.5 Prikaz svih registriranih korisnika

Pored linkova, navigacijska traka sadrži formu za pretraživanje korisnika. Nakon što korisnik unese ime korisnika pojavit će mu se popis svih korisnika koji sadrže uneseno ime. Za primjer je u formu uneseno ime Petra, te je dobiven rezultat kao što je prikazano na sl.4.6.

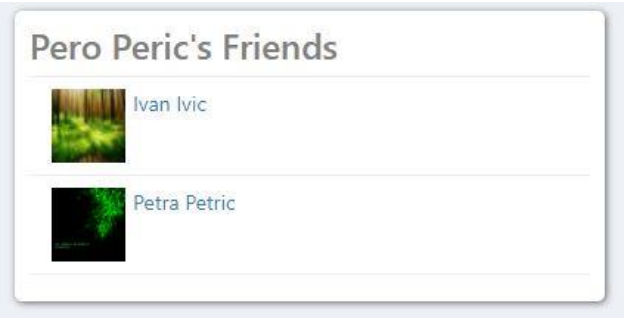


Sl.4.6 Prikaz traženih korisnika

Bočna traka prikazuje informacije prijavljenog, „ulogiranog“ korisnika, njegovu profilnu sliku, te opcije za prikazivanje poruka, vlastitih objava, zahtjeva za prijateljstvo (sl.4.7) i svih prijatelja (sl.4.8).



Sl.4.7 Prikaz zahtjeva za prijateljstvo



Sl.4.8 Prikaz svih prijatelja

Tu je i opcija za uređivanje profila pomoću koje korisnik može promijeniti svoje podatke kao i profilnu sliku (sl.4.9). Prilikom promjene lozinke, od korisnika se traži da prvo unese staru lozinku, pa zatim novu koja mora biti različita od stare lozinke. Nakon toga ponovno upisuje novu lozinku kako bi potvrdio da nije zabunom unio krivu lozinku u odnosu na željenu. Da bi se spremile promjene, korisnik mora kliknuti na gumb „Update“.



Sl.4.9 Prikaz forme za uređivanje profila

Zatim, tu je i link koji odjavljuje korisnika i vraća nazad na početnu stranicu. U glavnom, sadržajnom bloku, nalazi se forma za objavljivanje sadržaja, a ispod nje blok za prikazivanje objavljenog sadržaja. Prilikom objavljivanja sadržaja, od korisnika se traži da unese naslov objave, opis objave, da odabere temu pod kojom će se željeni sadržaj prikazati, te da odabere da li će sadržaj biti javan ili prikazan samo prijateljima. Korisnik ima i mogućnost objavljivanja slike koja dolazi u opis objave. U bloku za prikazivanje objava, prikazane su sve javne objave i objave prijatelja. Svaka objava prikazuje ime korisnika koji ju objavljuje, njegovu profilnu sliku, naslov objave, tema pod kojom se prikazuje, vrijeme objavljivanja, i naravno sadržaj objave. Ispod sadržaja, nalazi se gumb za prikazivanje komentara kao i komentiranje objava kao što je prikazano na sl.4.10.

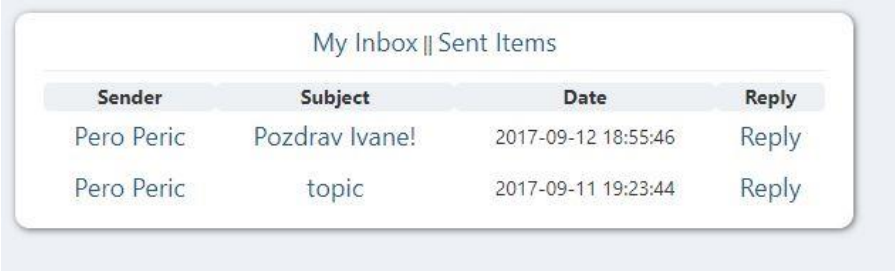


Sl.4.10 Prikaz objave i komentara

Korisnicima je omogućeno i brisanje vlastitih komentara na tuđim objavama i svih komentara na vlastitim objavama. Na gore prikazanoj slici, vidi se da je trenutno prijavljeni korisnik Petra Anic, te da je prikazan gumb za brisanje samo vlastitog komentara, pošto se nalazi na tuđoj objavi.

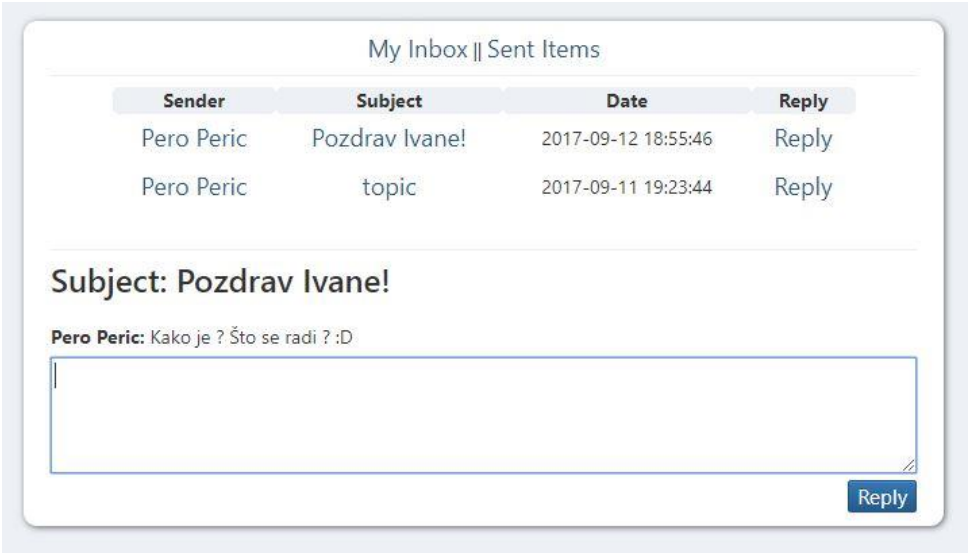
4.2.1 Poruke

Ukoliko korisnik odabere opciju „My messages“, prikazat će se blok sa pristiglim porukama kao što je prikazano na slici 4.11.



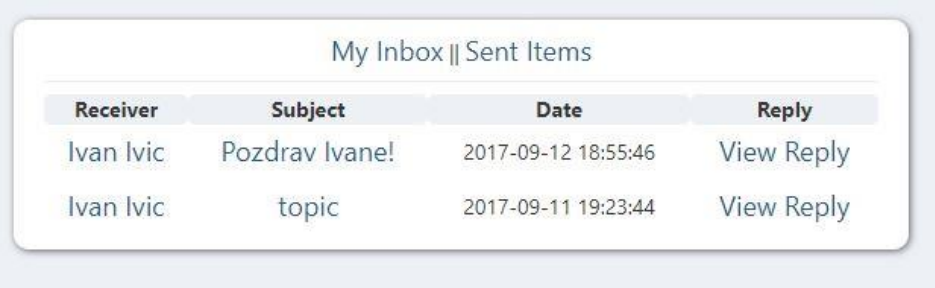
Sl.4.11 Prikaz pristiglih poruka

Korisniku je poznato od koga dobiva poruke, naslov poruka, te točan datum i vrijeme dospijeća poruka. Klikom na naslov poruke ili na „Reply“ link, proširuje se blok sa naslovom poruke, prikazanim sadržajem poruke, te formom za odgovaranje na poruku (sl.4.12).



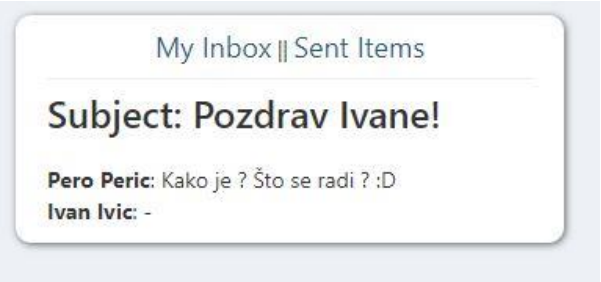
Sl.4.12 Prikaz poruka sa formom za odgovaranje

Klikom na link „Sent Items“ otvara se novi blok koji sadrži sve poslane poruke u istom formatu kao što je prikazano u bloku dospjelih poruka sa jedinom razlikom u prvom stupcu tablice, odnosno umjesto pošiljatelja prikazan je primatelj.



Sl.4.13 Prikaz poslanih poruka

Sadržaj poslane poruke se prikazuje na isti način kao i kod primljenih poruka, klikom na naslov poruke ili link „View Reply“. Klikom na link „My Inbox“ se vraća na primljene poruke.



Sl.4.14 Prikaz sadržaja poslane poruke

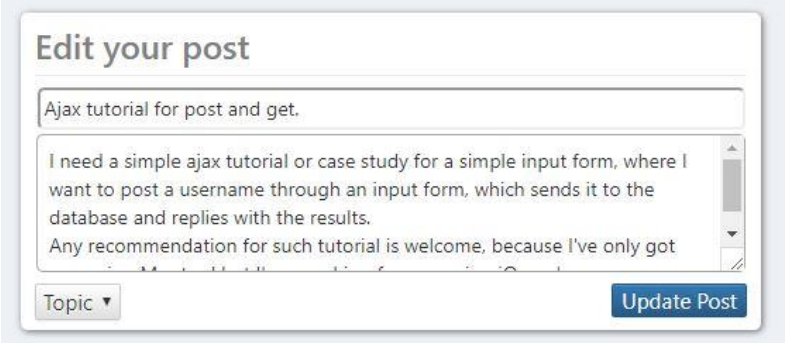
4.2.2 Korisničke objave

Kako bi korisnik vidio sve svoje objave, potrebno je kliknuti na link „My Posts“ koji se nalazi na bočnoj traci. Nakon toga, sve dohvaćene vlastite objave iz baze podataka, pored opcija za prikaz svih komentara i komentiranja (gumb View comments) sada imaju još i mogućnosti brisanja (gumb Delete) i uređivanja (gumb Edit) što je prikazano na sl.4.15.



Sl.4.15 Prikaz svih vlastitih objava

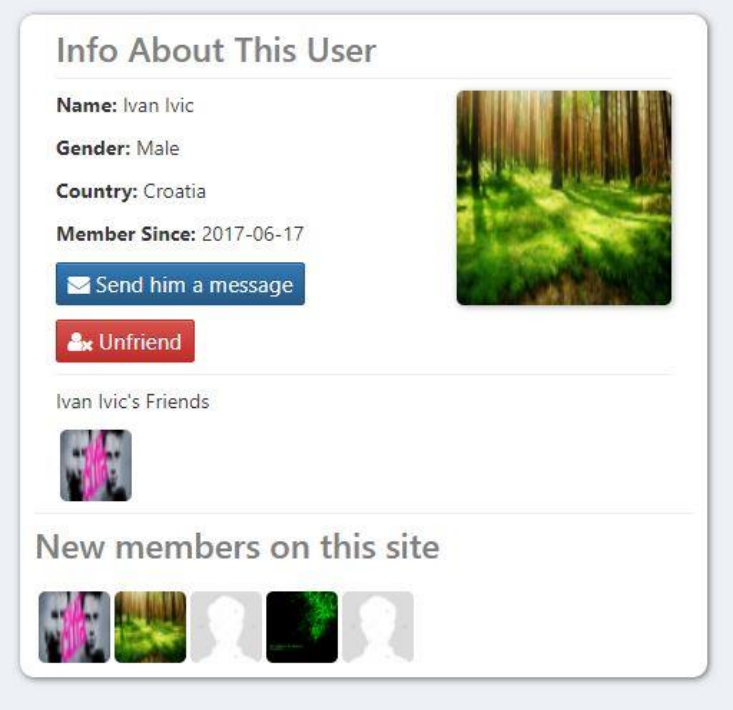
Prilikom brisanja objave, briše se kompletno iz baze podataka, naravno, zajedno sa svim komentarima koje sadrži. Ako se želi izmijeniti objavljeni sadržaj, naslov, čak i tema pod kojom se prikazuje, klikom na gumb „Edit“ otvara se novi blok u kojem je to moguće ostvariti. Klikom na „Update Post“ u bazu podataka se pod istim indeksom sprema izmijenjena objava.



Sl.4.16 Prikaz opcije uređivanja objave

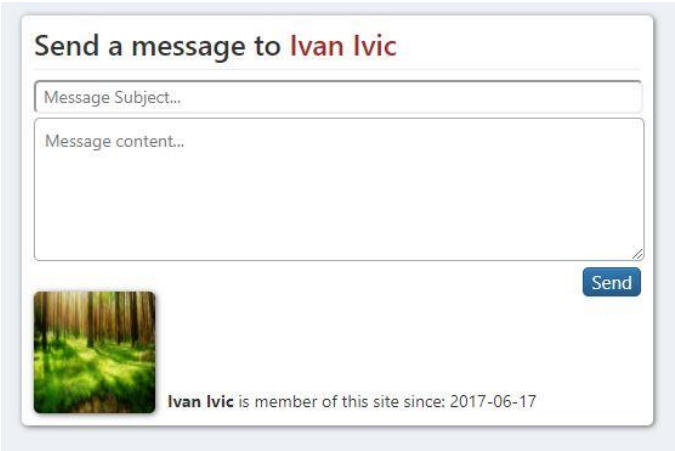
4.2.3 Korisnički profil

Blok korisničkog profila se prikazuje kada korisnik klikne na profilnu sliku ili ime drugog korisnika. Profil sadrži opće informacije o korisniku (ime, spol, država prebivanja, datum učlanjenja na društvenu mrežu), profilnu sliku u većem formatu, te gumb za slanje nove poruke i gumb za prijateljstvo. Ukoliko je korisnik prijatelj, u našem slučaju jest, bit će prikazan gumb „Unfriend“ koji uklanja korisnika kao prijatelja. U protivnom bio bi gumb „Add friend“ koji služi za dodavanje korisnika kao prijatelja. U slučaju da korisnici međusobno nisu prijatelji, a korisnik čiji je profil, je poslao zahtjev za prijateljstvo „ulogiranom“ korisniku, bili bi prikazani gumbovi kao što smo vidjeli u bloku za zahtjeve prijateljstva, odnosno gumbovi „Accept“ i „Ignore“. Profil sadrži i popis svih korisnikovih prijatelja, kao i nove članove društvene mreže.



Sl.5.17 Prikaz korisničkog profila

Za pristupanje formi za slanje nove poruke, potrebno je kliknuti na gumb „Send him a message“. Tada se forma prikazuje u novom bloku kao što to vidimo na sl.4.18.



Sl.4.18 Prikaz forme za slanje poruke

Forma je slična formi za odgovaranje na poruke s razlikom da ovdje, osim sadržaja poruke unosimo i naslov poruke. Nakon uspješno poslane poruke, korisnik se preusmjerava nazad na korisnički profil.

4.3 Opis koda

Kao što je to već ranije spomenuto, za izradu društvene mreže koristiti ćemo HTML i CSS za front-end, dok za back-end koristimo PHP i MySQL programske jezike. Izrada počinje sa kreiranjem mape proizvoljnog naziva unutar korijenske, „root“ mape, naziva „htdocs“ koja se nalazi unutar instalacijske mape XAMPP programskog okruženja. U kreiranu mapu spremamo sve datoteke potrebne za rad projekta. Da bismo povezali naše datoteke sa kreiranom bazom podataka, potrebno je unutar .php datoteke napraviti konekciju sa bazom kao što je prikazano u sljedećem kodu.

```
<?php
$con = mysqli_connect("localhost","root","","social_network") or die ("Connection wasn’t
established");
?>
```

Vidimo da se varijabli „con“ dodjeljuje rezultat funkcije „mysqli_connect“ koja prima četiri parametara, a to su server, korisnik, lozinka i baza podataka s kojom se korisnik povezuje. Ukoliko se ne ostvari konekcija, pojaviti će se poruka “Connection wasn’t established”.

Budući da ima mnogo linija koda, u nastavku će biti prikazani samo bitniji dijelovi koda.

Kod za registraciju korisnika:

```
if(isset($_POST['sign_up'])){
	// spremanje unesenih podataka u varijable
	$name = mysqli_real_escape_string($con, $_POST['u_name']); // ime
	$pass = mysqli_real_escape_string($con, $_POST['u_pass']); // lozinka
	$c_pass = mysqli_real_escape_string($con, $_POST['c_pass']); // potvrdna lozinka
	$email = mysqli_real_escape_string($con, $_POST['u_email']); //email adresa
	$country = mysqli_real_escape_string($con, $_POST['u_country']); //država
	$gender = mysqli_real_escape_string($con, $_POST['u_gender']); // spol
	$b_day = mysqli_real_escape_string($con, $_POST['u_birthday']); // datum rođenja
	$status = "unverified"; // da li je korisnički račun ovjeren
	$posts = "No"; // zadano ‘ne’ za objave, prikazuje da li korisnik ima objavu
	$get_email = "select * from users where user_email='$email'";
	$run_email = mysqli_query($con, $get_email);
	$check = mysqli_num_rows($run_email);
	// provjeravanje postojeće email adrese
	if($check==1){
		echo "<script>alert('Email is already registered, please try another 
one!')</script>";
		exit();
	}
	// postavljanje uvjeta dužine lozinke
	if(strlen($pass) && strlen($c_pass) < 8){
	echo "<script>alert('Password should contain minimum 8 characters!')</script>";
	exit();
	}
	if($pass == $c_pass){ // provjeravanje identičnosti lozinke i potvrdne lozinke
		// unošenje podataka u bazu
		$insert = "insert into users (user_name, user_pass, user_email, user_country, 
user_gender, user_b_day, user_image, register_date, status, posts) values ('$name', '$pass', 
'$email', '$country', '$gender', '$b_day', 'default.jpg', NOW(), '$status', '$posts')";

		$run_insert = mysqli_query($con, $insert);
		// provjera uspješnosti ažuriranja baze
		if($run_insert){
			$_SESSION['user_name']= $name;
			echo "<script>alert('Registration Successful!')</script>";
			echo "<script>window.open('home.php', '_self')</script>";
		}
	}
	else{ // ako je neispravno, izbaci upozorenje
		echo "<h4>Password does not match the confirm password!</h4>";
	}
}
```

Kod se izvršava tek kada korisnik klikne na gumb „sign up“. Prvo se provjerava da li je e-mail adresa zauzeta, odnosno, da li postoji ista u tablici „users“. Ako postoji, izbaciti će skriptni prozor upozorenja. U protivnom kod nastavlja dalje sa izvršavanjem. Od korisnika se očekuje da unese lozinku koja minimalno sadrži osam znakova. Lozinka se zatim mora unijeti još jednom radi

provjere što se provjerava uvjetom „if“. Kada su svi uvjeti ispunjeni i sve forme popunjene, uneseni podaci se upitom spremaju u bazu podataka i time stvaraju novi redak u tablici korisnika.

Kod za prijavu korisnika:

```
if(isset($_POST['login'])){
	// spremanje unesene email adrese i lozinke u varijable
	$email = mysqli_real_escape_string($con, $_POST['email']);
	$pass = mysqli_real_escape_string($con, $_POST['pass']);
	// uspoređivanje unesenih podataka sa onima u bazi
	$get_user = "select * from users where user_email='$email' AND user_pass='$pass'";
	$run_user = mysqli_query($con, $get_user);
	$check = mysqli_num_rows($run_user);
	// provjera ispravnosti email adrese i lozinke
	if($check==1){
		$_SESSION['user_email']=$email;
		$user_id_query = "SELECT user_id FROM users WHERE user_email='$email' AND
user_pass='$pass'";
		$run_topics = mysqli_query($con, $user_id_query);
		while($row = mysqli_fetch_array($run_topics)){
			$user_id = $row['user_id'];
		}
		$_SESSION['user_id'] = $user_id;
		echo "<script>window.open('home.php', '_self')</script>";
	}
	// ako je neispravno, izbaci upozorenje
	else{
		echo "<script>alert('Password or email is not correct!')</script>";
	}
}
```

Gumbom „Login“ se prvo uspoređuje par podataka (email i lozinka) „login“ forme sa svim parovima takvih podataka u tablici „users“. Ako postoji isti, stvara se nova sesija preko e-mail adrese pomoću koje se dohvaća korisnikov indeks iz tablice. Nakon toga, stvara se sesija preko indeksa što znači da je ostvaren pristup društvenoj mreži.

Kod funkcije za objavljivanje sadržaja:

```
function insertPost(){
	if(isset($_POST['sub'])){
	global $con; // globalna varijabla konekcije baze podataka
	global $user_id; // indeks korisnika
	$title = addslashes($_POST['title']); // naslov objave
	$content = addslashes($_POST['content']); // sadržaj objave
	$topic = $_POST['topic']; // tema pod kojom se objava pokazuje
	$is_public = $_POST['visibly']; // vidljivost objave; samo prijatelji ili javna
	// dohvaćanje slike
	$post_image = $_FILES['post_image']['name']; // slika
	$image_tmp = $_FILES['post_image']['tmp_name']; // privremena slika
```

```
        // spremanje slike u mapu post_images
        move_uploaded_file($image_tmp, "user/post_images/$post_image");
        // postavljanje uvjeta dužine naslova objave
        if(strlen($title)>35){
        echo "<script>alert('Title can contain maximum 35 characters!')</script>";
        echo "<script>window.open('home.php','_self')</script>";
        exit();
        }
        // ako je ispunjen uvjet, unesi podatke u bazu
        else {
        $insert = "insert into posts(user_id, topic_id, post_title, post_content, image,
post_date, is_public) values('$user_id','$topic','$title','$content','$post_image', NOW(),
$is_public)";
        $run = mysqli_query($con, $insert);
        // nakon prve objave korisnika, stavlja se aktivan status objava, tj. posts=yes
        if($run){
                $update = "update users set posts='yes' where user_id = '$user_id'";
                $run_update = mysqli_query($con, $update);
                echo "<script>window.open('home.php','_self')</script>";
                }
        }
        }
}
```

Funkcija započinje uvjetom „if“ koji od korisnika zahtjeva da klikne na gumb „Post to Timeline“. Ukoliko klikne nakon što unese željeni sadržaj u formu, postavlja se uvjet da naslov ne smije sadržavati više od 35 znakova. Forma ima mogućnost objavljivanja slike, što zahtjeva da se u formu doda atribut „enctype='multipart/form-data'“. Slika koju korisnik objavljuje se za razliku od ostalih podataka, koji su spremljeni samo u bazi podataka, sprema i u mapu „post_images“. Po zadanom, novim korisnicima u tablici piše da nemaju niti jednu objavu. Nakon prve objave stvara se upit koji ažurira tablicu postavljanjem statusa objavljivanja aktivnim (posts = 'yes').

Kod funkcije za dohvaćanje objavljenog sadržaja iz baze podataka:

```
function get_posts(){
        global $con;
        $sMyFriends = ''; // deklariranje string varijable prijatelja
        $my_friends_res = mysqli_fetch_assoc(mysqli_query($con, "SELECT friend_array FROM
users WHERE user_id = ".$_SESSION['user_id'])); // dohvaćanje prijatelja, ako ih ima
        $sMyFriends = $my_friends_res['friend_array'];
        $sFriendsFilter = ' '; // filter prijatelja, produžetak upita u bazu
        if($sMyFriends != '') // ako imamo prijatelje, filteru se dodaju indeksi prijatelja
        {
                $sFriendsFilter = "OR USER_ID IN (".$sMyFriends.")";
        }
        $user_id = $_SESSION['user_id'];
        // dohvaćanje vlastitih, javnih objava, te objava prijatelja
        $get_posts = "select * from posts WHERE is_public = 1 ".$sFriendsFilter." or
user_id='$user_id' ORDER by 1 DESC";
```

```
        $run_posts = mysqli_query($con, $get_posts);
        // petlja koja sprema dohvaćene podatke objava u varijable
        while($row_posts = mysqli_fetch_array($run_posts)){
                $post_id = $row_posts['post_id'];
                $user_id = $row_posts['user_id'];
                $post_title = $row_posts['post_title'];
                $content = $row_posts['post_content'];
                $post_date = $row_posts['post_date'];
                $post_image = $row_posts['image'];
                $topic_id = $row_posts['topic_id'];
                // dohvaćanje teme pod kojom se objava prikazuje
                $topic ="select * from topics where topic_id='$topic_id'";
                $run_topic = mysqli_query($con, $topic);
                $row_topic = mysqli_fetch_array($run_topic);
                $topic_title = $row_topic['topic_title'];
                //dohvaćanje korisnika koji je objavio sadržaj
                $user = "select * from users where user_id='$user_id' AND posts='yes'";
                $run_user = mysqli_query($con, $user);
                $row_user = mysqli_fetch_array($run_user);
                $user_name = $row_user['user_name'];
                $user_image = $row_user['user_image'];
                // dohvaćanje komentara koji pripadaju objavi
                $get_com = "select * from comments where post_id='$post_id'";
                $run_com = mysqli_query($con, $get_com);
                $count_com = mysqli_num_rows($run_com); // brojač komentara
                // prikazivanje sveg sadržaja odjednom
                echo "<div id='posts'>
                <p><img id='profile_pic' src='user/user_images/$user_image'/></p>
                <h3><a href='user_profile.php?u_id=$user_id'>$user_name</a></h3>
                <h3>$post_title <i style='font-size:12px;' color:black;
value='$topic_id'>$topic_title</i></h3>
                <i style='font-size:12px;'>$post_date</i><br>
                <p id='post_content'>$content</p>";
                if($post_image != ""){ echo "<a href='user/post_images/".$post_image."'
target='_blank'><img id='content_pic' style='width:100%'
src='user/post_images/".$post_image."'></a>";}
                if($count_com==0){ // ako nema komentara prikaži gumb bez brojke komentara
                        echo "<a class='btn btn-default' href='single.php?post_id=$post_id'
style='float:right; padding:1px 5px; display:block;'><i class='fa fa-comments' aria-
hidden='true'></i> View comments</a></div><br>";}
                else{ // ako ima komentara, pored gumba prikaži koliko ih ima
                        echo "<a class='btn btn-default' href='single.php?post_id=$post_id'
style='float:right; padding:1px 5px; display:block;'><i class='fa fa-comments' aria-
hidden='true'></i> View comments ($count_com)</a></div><br>";
                }
        }
}
```

Funkcija prvo provjerava da li korisnik ima prijatelje. Ako ima, varijabli „sFriendsFilter“ se dodjeljuje produžetak upita u bazu koji dohvaća objave prijatelja. Zatim se upitom dohvaćaju sve vlastite, javne objave i objave prijatelja. Podaci dohvaćenih objava se uz pomoć „while“ petlje spremaju u varijable koje se koriste da bi objava bila prikazana.

Kod funkcije za dohvaćanje rezultata tražilice:

```
function GetResults(){
	global $con;
	if(isset($_GET['user_query'])){
		$search_term = $_GET['user_query']; // spremanje traženog stringa u varijablu
	}
	// traženje korisnika koji sadrži traženi string
	$get_user = "select * from users where user_name LIKE '%$search_term%' ORDER by 1
DESC";
	$run_user = mysqli_query($con, $get_user);
	$count_result = mysqli_num_rows($run_user); // brojač rezultata
	// ako nema rezultata, izbaci upozorenje
	if($count_result == 0){
		echo "<h3>Sorry, we don't have results for that user!</h3>";
		exit();
	}
	// petlja koja sprema dohvaćene podatke korisnika u varijable
	while($row_user = mysqli_fetch_array($run_user)){
		$user_id = $row_user['user_id'];
		$user_name = $row_user['user_name'];
		$user_image = $row_user['user_image'];
		// prikazivanje nađenih korisnika
		echo "<div class='container-fluid'>
		<p><img id='imgg' src='user/user_images/$user_image' width='50'
height='50'/></p>
		<a href='user_profile.php?u_id=$user_id'>$user_name</a></div><hr>";
	}
}
```

Funkcija „GetResults“ prvo sprema traženi string u novu varijablu koja se zatim uspoređuje sa svim korisničkim imenima u bazi. Ako ne postoji takav string, pojaviti će se „heading“ sa tekstom da nema nađenih rezultata. Ukoliko postoji, pomoću „while“ petlje se osim imena, dohvaća profilna slika i indeks koji služi u adresi linka da nas prilikom klika na ime nađenih korisnika odvede na njihov profil.

Kod funkcije za slanje zahtjeva za prijateljstvo:

```
function AddFriend($user_id)
{
	global $con;
	// unošenje zahtjeva u bazu sa vrijednostima korisničkih indeksa
	$friend_request_query = "INSERT INTO friend_requests (user_id_from, user_id_to)
VALUES (".$_SESSION['user_id']."), ".$user_id.")";
	$res = mysqli_query($con, $friend_request_query);
	echo "<script>window.open('user_profile.php?u_id=$user_id','_self')</script>";
}
```

Za razliku od prethodnih funkcija, ova funkcija prima jedan parametar, odnosno indeks korisnika koji prima zahtjev. Tom indeksu pristupamo pomoću varijable „$_GET“ varijable (dodijeljena je

varijabli $user_id izvan funkcije), dok indeksu korisnika koji šalje zahtjev pristupamo pomoću sesije. Funkcija se sastoji od samo jednog upita u bazu koji u tablicu unosi prethodno navedene indekse.

Kod funkcije za prihvaćanje zahtjeva za prijateljstvo:

```
function AcceptFriend($user_id)
{
	global $con;
	$aMyFriends = array(); // polje vlastitih prijatelja
	$aHisFriends = array(); // polje prijatelja korisnika od kojeg dobivamo zahtjev
	$sMyFriends = ""; // string vlastitih prijatelja
	$sHisFriends = ""; // string prijatelja korisnika od kojeg dobivamo zahtjev

	$delete_request = "DELETE FROM friend_requests WHERE user_id_from = ".$user_id." AND
user_id_to = ".$_SESSION['user_id']; // brisanje zahtjeva iz baze
	$res = mysqli_query($con, $delete_request);
	$my_friends_res = mysqli_fetch_assoc(mysqli_query($con, "SELECT friend_array FROM
users WHERE user_id = ".$_SESSION['user_id'])); // dohvaćanje vlastitih prijatelja

	if($my_friends_res['friend_array'] != '')
	{	// ako imamo prijatelje, zanemari znak zarez poslije indeksa u polju prijatelja
		$aMyFriends = explode(",", $my_friends_res['friend_array']);
	}

	$his_friends_res = mysqli_fetch_assoc(mysqli_query($con, "SELECT friend_array FROM
users WHERE user_id = ".$user_id)); // dohvaćanje prijatelja od korisnika koji šalje zahtjev
	if($his_friends_res['friend_array'] != '')
	{	// ako imamo prijatelje, zanemari znak zarez poslije indeksa u polju prijatelja
		$aHisFriends = explode(",", $his_friends_res['friend_array']);
	}
	array_push($aMyFriends, $user_id); // unošenje indeksa korisnika u polje prijatelja
	array_push($aHisFriends, $_SESSION['user_id']); // unošenje vlastitog indeksa u polje
prijatelja korisnika kojeg prihvaćamo kao prijatelja

	$sMyFriends = implode(",", $aMyFriends); // pretvaranje polje nizova u niz
	$sHisFriends = implode(",", $aHisFriends); // pretvaranje polje nizova u niz
	// ažuriranje tablice korisnika koji prima zahtjev
	$add_to_my_query = "UPDATE users SET friend_array = '".$sMyFriends."' WHERE user_id =
".$_SESSION['user_id'];
	$add_to_my_res = mysqli_query($con, $add_to_my_query);
	// ažuriranje tablice korisnika koji šalje zahtjev
	$add_to_his_query = "UPDATE users SET friend_array = '".$sHisFriends."' WHERE user_id
= ".$user_id;
	$add_to_his_res = mysqli_query($con, $add_to_his_query);

	echo "<script>window.open('user_profile.php?u_id=$user_id','_self')</script>";
}
```

Funkcija najprije briše zahtjev iz tablice zahtjeva za prijateljstvo tako što ukloni indekse korisnika koji šalje i korisnika koji prima zahtjev. Zatim se dohvaća niz vlastitih prijatelja i niz prijatelja korisnika koji šalje zahtjev. Budući da su indeksi u nizu odvojeni zarezom, bitno ga je zanemariti.

To se radi pomoću funkcije „explode“ koja zapravo od niza pravi polje nizova. Na primjer, od niza [1, 2, 3, 4], dobiti ćemo rezultat [1], [2], [3], [4]. Kada smo dobili polje nizova, odnosno polje korisničkih indeksa, funkcijom „array_push“ se na kraj vlastitog polja stavlja indeks korisnika kojeg prihvaćamo kao prijatelja i obratno. Nakon toga se polje nizova mora vratiti u početno stanje, odnosno u jedan niz, za što je zaslužna funkcija „implode“. Na kraju funkcije se ažuriraju baze korisnika.

Kod funkcije za odbijanje zahtjeva za prijateljstvo:

```
function IgnoreRequest($user_id){
	global $con;
	// brisanje zahtjeva iz baze
	$delete_request = "DELETE FROM friend_requests WHERE user_id_from = ".$user_id." AND
user_id_to = ".$_SESSION['user_id'];
	$res = mysqli_query($con, $delete_request);
	echo "Request Ignored!";
	echo "<script>window.open('user_profile.php?u_id=$user_id','_self')</script>";
}
```

Funkcija za odbijanje zahtjeva briše zahtjev iz tablice potpuno istim upitom kao ranije spomenuta funkcija „AcceptFriend“.

Kod funkcije za uklanjanje prijatelja:

```
function RemoveFriend($user_id)
{
	global $con;
	$aMyFriends = array();
	$aHisFriends = array();
	$sMyFriends = "";
	$sHisFriends = "";

	$my_friends_res = mysqli_fetch_assoc(mysqli_query($con, "SELECT friend_array FROM
users WHERE user_id = ".$_SESSION['user_id']));
	$aMyFriends = explode(",", $my_friends_res['friend_array']);

	$his_friends_res = mysqli_fetch_assoc(mysqli_query($con, "SELECT friend_array FROM
users WHERE user_id = ".$user_id));
	$aHisFriends = explode(",", $his_friends_res['friend_array']);

	// traženje indeksa korisnika kojeg uklanjamo iz vlastitog polja prijatelja
	$his_index = array_search($user_id, $aMyFriends);
	// traženje vlastitog indeksa u polju prijatelja od korisnika kojeg uklanjamo
	$my_index = array_search($_SESSION['user_id'], $aHisFriends);
	// brisanje nađenih indexa
	unset($aMyFriends[$his_index]);
	unset($aHisFriends[$my_index]);
```

```
        $sMyFriends = implode(",", $aMyFriends);
        $sHisFriends = implode(",", $aHisFriends);
        // ažuriranje niza prijatelja oba korisnika
        $my_new_list_query = "UPDATE users SET friend_array = '".$sMyFriends."' WHERE user_id
= ".$_SESSION['user_id'];
        $my_new_list_res = mysqli_query($con, $my_new_list_query);
        $his_new_list_query = "UPDATE users SET friend_array = '".$sHisFriends."' WHERE
user_id = ".$user_id;
        $his_new_list_res = mysqli_query($con, $his_new_list_query);
        echo "<script>window.open('user_profile.php?u_id=$user_id','_self')</script>";
}
```

Funkcija pretvara niz prijatelja u polje nizova na isti način kao funkcija za prihvaćanje prijatelja, odnosno koristi funkciju „explode“. Nakon toga, pomoću funkcije „array_search“ pretražuje indekse korisnika koje zatim uklanja iz polja pomoću funkcije „unset“. Nakon što ih ukloni, vraća polje nizova u jedan niz funkcijom „implode“, te na kraju ažurira bazu podataka.

Kod za prikaz komentara i komentiranje:

```
<?php
$get_id = $_GET['post_id']; // dohvaćanje indeksa objave
// dohvaćanje komentara koje objava sadrži
$get_com = "select * from comments where post_id='$get_id' ORDER by 1 ASC";
$run_com = mysqli_query($con, $get_com);
// dohvaćanje korisnika koji je autor objave
$user = "select * from users where user_id='$user_id' AND posts='yes'";
$run_user = mysqli_query($con, $user);
$row_user = mysqli_fetch_array($run_user);
// spremanje podataka korisnika u varijable
$user_name = $row_user['user_name'];
$user_image = $row_user['user_image'];
// dohvaćanje objave
$get_posts = "select * from posts where post_id='$get_id'";
$run_posts = mysqli_query($con, $get_posts);
$row_posts = mysqli_fetch_array($run_posts);
// spremanje podataka objave u varijable
$post_id = $row_posts['post_id'];
$user_id = $row_posts['user_id'];
// petlja koja sprema podatke komentara u varijable
while($row = mysqli_fetch_array($run_com)){
        $com = $row['comment'];
        $com_author_id = $row['comment_author_id'];
        $date = $row['date'];
        $comment_id = $row['comment_id'];
        // dohvaćanje autora komentara
        $get_author_name = "select user_name from users where user_id='$com_author_id'";
        $run_get_author_name = mysqli_query($con, $get_author_name);
        $row_get_author_name = mysqli_fetch_array($run_get_author_name);
        $author_name = $row_get_author_name['user_name'];
        echo " // prikaz komentara
        <div id='comments'><a href='user_profile.php?u_id=$com_author_id'>
<h3>$author_name</h3></a><i><span>Said</i> on $date</span><p>$com</p>";
```

```
        $get_u_id = $_SESSION['user_id'];
        $query = "SELECT user_id FROM posts WHERE user_id='$get_u_id'";
        $run_query = mysqli_query($con, $query);
        // omogućeno brisanje vlastitih komentara na tuđim objavama i svih komentara na
vlastitim objavama
        if ($com_author_id==$get_u_id || $user_id==$get_u_id){
                echo "<a href='functions/delete_post.php?comment_id=$comment_id'><button
name='com_del' type='button' style='float:right; padding: 2px 6px;' class='btn btn-
danger'>Delete</button></a>";
        }
}
if(isset($_POST['reply'])){ // prilikom klika na gumb ‘reply’, komentar se unosi u bazu
        $comment = $_POST['comment']; // spremanje unesenog sadržaja komentara u varijablu
        $insert = "insert into comments (post_id, user_id, comment, comment_author_id, date)
values ('$post_id', '$user_id', '$comment','$user_com_id', NOW())";
        $run = mysqli_query($con, $insert);
        echo "<script>window.open('single.php?post_id=$post_id', '_self')</script>";
}
?>
```

Kod prvo dohvaća indeks objave na kojoj se korisnik nalazi, zatim upitom baze dohvaća sve komentare koje ista sadrži, te ih sortira uzlazno, od najstarijeg do najnovijeg. Nakon toga, dohvaća sve potrebne podatke da se prikaže objava i svi komentari koje sadrži. U kodu se postavlja uvjet koji dozvoljava brisanje vlastitih komentara na tuđim objavama i svih komentara na vlastitim objavama. Na kraju koda se definira akcija prilikom klika na gumb za objavljivanje komentara.

Kod za slanje poruke:

```
<?php
if(isset($_POST['message'])){
        $msg_title = $_POST['msg_title']; // spremanje unesenog naslova poruke u varijablu
        $msg = $_POST['msg']; // spremanje unesenog sadržaja poruke u varijablu
        // unošenje podataka poruke u bazu
        $insert = "insert into messages
(sender,receiver,msg_sub,msg_topic,reply,status,msg_date) values ('$user_id', '$u_id',
'$msg_title', '$msg', 'no_reply', 'unread', NOW())";
        $run_insert = mysqli_query($con, $insert);
        if($run_insert){
        echo "<center><h3>Message was sent to ". $user_name . " successfully</h3></center>";
        }
        else echo "<center><h3>Message was not sent...</h3></center>";
}
?>
```

Gumbom za slanje poruke se spremaju uneseni podaci (naslov i sadržaj poruke) u nove varijable koje se zatim unose u bazu upitom „run_insert“. Ukoliko je upit uspješno obavljen, prikazati će se „heading“ sa tekstom da je poruka poslana. U protivnom, bit će to „heading“ sa tekstom da nije poslana.

5. ZAKLJUČAK

Društvena mreža nije novi pojam, međutim, u današnje vrijeme pojavljuje se u novom, virtualnom, online obliku kojeg nam pretraživači, web preglednici omogućavaju prikazati. Smatra se najjednostavnijim načinom za međusobnu komunikaciju, što joj je glavna namjena. Ona dakle, socijalno povezuje korisnike što često uključuje grupiranje specifičnih pojedinaca ili organizacija. Prijatelji koje pojedinac može steći koristeći online društvene mreže samo su jedna od pogodnosti koje nam takve mreže omogućavaju. Još jedna od beneficija je i povezivanje i komuniciranje s osobama koje žive diljem svijeta. Zadatak ovog rada je bio napraviti upravo takvu mrežu, sa njenim najosnovnijim funkcijama što uključuje dopisivanje, dijeljenje i komentiranje sadržaja koji može biti filtriran od javnosti, odnosno podijeljen samo za prijatelje. Rad uključuje i grupiranje objavljenog sadržaja u teme koje korisnici odabiru, te tražilicu korisnika, što dodatno pomaže u pronalaženju prijatelja. Rad je napisan u najpoznatijim tehnologijama za izradu vanjskog „front-end“ i unutarnjeg „back-end“ dijela. U prvom dijelu rada ukratko je opisana svaka od tehnologija korištenih u radu, dok je u drugom dijelu pokazana primjena bitnijih tehnologija, PHP i MySQL programskih jezika.

LITERATURA

[1] Autorski tim SRCA, (2016.): Uvod u HTML

[2] Vlatka Paunović, Siniša Tomić, (2006.): PHP Priručnik uz seminar

[3] php.com.hr, (2008.): http://php.com.hr/66 (24.6.2017.)

[4] aubx: https://www.aubx.com/faks/ergonomija/mysql.html (24.6.2017.)

[5] Dalibor D. Dvorski, (2007.): Installing, configuring, and developing with XAMPP: (24.8.2017.)

[6] Bootstrap: http://getbootstrap.com/ (26.8. 2017)

SAŽETAK

Cilj ovog rada je bio isprogramirati i dizajnirati sustav za dijeljenje sadržaja sa drugim korisnicima, odnosno društvenu mrežu. S obzirom da je to kompleksna web aplikacija, potrebno ju je napraviti jednostavnu za korištenje, sa svim osnovnim funkcijama i mogućnostima. U radu je prikazan i opisan izgled i princip rada svake od funkcionalnosti. Mreža je dizajnirana pomoću front-end tehnologija, HTML i CSS, dok je za izradu centralne logike korišten PHP. Izmjena podataka između korisničkog sučelja i jezgre omogućena je upitima u bazu podataka pomoću MySQL tehnologije.

Ključne riječi: Društvena Mreža, HTML, CSS, Php, MySQL, XAMPP

ABSTRACT

DEVELOPING A SOCIAL NETWORK

The aim of this paper was to program and design a content sharing system with other users, that is social network. Since this is a complex web application, it is necessary to make it easy to use, with all the basic functions and capabilities. This paper presents and describes the layout and operation principle of each of the functionalities. The network is designed using front-end technology, HTML and CSS, while PHP is used to generate central logic. Data exchange between the user interface and the core is enabled by querying the database using MySQL technology.

Keywords: Social Network, HTML, CSS, Php, MySQL, XAMPP

ŽIVOTOPIS

Marko Huljak rođen je u Virovitici, Republika Hrvatska, 10. studenog 1995. godine. Završio je Osnovnu školu Vladimira Nazora u Virovitici, nakon čega 2010.g. upisuje Elektrotehničku školu u Virovitici koju završava 2014. godine. Iste godine upisuje preddiplomski sveučilišni studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

Vlastoručni potpis:

_____

Marko Huljak

PRILOZI

```
<?php
session_start();
if(!isset($_SESSION['user_email'])){ header("location: index.php");}
include("includes/connection.php");
include("functions/functionss.php");
include("user_timeline.php");
?>

<div id="content_timeline" style="margin-right: 550px;">
<form action="home.php?id=<?php echo $user_id;?>" method="post" id="f" enctype="multipart/form-
data"><h3 style="margin-bottom: 5px;">What's on your mind?</h3><center>
<input type="text" name="title" placeholder="Write a Title..." size="88" required="required"
style="border-radius: 7px; padding: 7px; margin-bottom: 2px;"/><br>
<textarea style="border-radius: 7px; padding: 7px;" cols="91" rows="4" name="content"
placeholder="Write description..." required></textarea></center>
<select name="topic" id="topic" class="btn btn-default" title="Select a topic" required>
       <option value="" id="opcija">Topics</option>
       <?php getTopics();?>
</select>
<button  name="sub" id="post_to_timeline" type="submit" class="btn btn-primary">Post to
Timeline</button>
<select name="visibly" id="visibly" class="btn btn-default" title="Visibility" required>
       <option value="0">Friends</option>
       <option value="1">Public</option>
</select>

<input style="float: left; margin-top: 3px;" id="myFileInput" name="post_image" type="file">
<button style="float: left; padding: 3px 5px;" title="Add photos to your post" type="button"
class="btn btn-default" onclick="$('#myFileInput').trigger('click');"><i class="glyphicon
glyphicon-picture"></i></button>
</form>
<?php
       insertPost();
?>
       <br><h3>Most Recent Discussions!</h3><br>
       <div id="posts_container" style="display:flex; flex-direction:column;">
       <?php get_posts();?>
       </div>
</div>
</div>
</div>
       <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
       <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.11.0/umd/popper.min.js"
integrity="sha384-b/U6ypiBEHpOf/4+1nzFpr53nxSS+GLCkfwBdFNTxtclqqenISfwAzpKaMNFNmj4"
crossorigin="anonymous"></script>
       <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/js/bootstrap.min.js"
integrity="sha384-h0AbiXch4ZDo7tp9hKZ4TsHbi047NrKGLO3SEJAg45jXxnGIfYzk4Si90RDIqNm1"
crossorigin="anonymous"></script>
</body>
</html>
```

P.4.1. Kod datoteke home.php