Web aplikacija društvena mreža

Kolarević, Matej

Undergraduate thesis / Završni rad

2020

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:124290

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-21**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

Stručni studij informatike

WEB APLIKACIJA, DRUŠTVENA MREŽA

Završni rad

Matej Kolarević

Osijek, 2020.

SADRŽAJ

1. UVOD....2
1.1 Zadatak završnog rada....2
2. PROGRAMSKI JEZICI....3
2.1 HTML....3
2.2 CSS....4
2.2.1 Bootstrap....6
2.3 JavaScript....7
2.4 PHP....8
2.5 SQL....10
3. WEB APLIKACIJA IZ POSLUŽITELJSKE PERSPEKTIVE....13
3.1 XAMPP....13
4. WEB APLIKACIJA IZ KLIJENTSKE PERSPEKTIVE....16
5. ZAKLJUČAK....24
LITERATURA....25
SAŽETAK....26
ABSTRACT....27
Social network web application....27
PRILOZI....28

1. UVOD

S obzirom da je u današnje vrijeme sve više potrebno održati komunikaciju s obitelji, poslodavcima i prijateljima, društvene mreže su jako bitne. U ovome radu izraditi će se jednostavna društvena mreža. Aplikacija će dati korisnicima mogućnost objave javnih komentara, te komentara svojima kontaktima. Omogućiti će ocjenjivanje komentara pozitivnom ili negativnom ocjenom. Također će se istražiti koji se problemi mogu dogoditi pri izradi i održavanju društvene mreže.

Problematika će se opisati kroz 3 poglavlja. U drugom polgavlju će se opisati programski jezici korišteni za izradu rada, kao što su HTML, CSS, JavaScript i PHP. U trećem poglavlju će se opisati potrebni alati koje poslužitelj (server) mora imati radi posluživanja stranice, dok će se u četvrtome poglavlju opisati izgled i ponašanje gotove aplikacije iz perspektive krajnjeg korisnika (klijenta).

1.1 Zadatak završnog rada

U radu je potrebno napraviti web aplikaciju - društvenu mrežu koja će omogućiti korisnicima međusobnu komunikaciju, praćenje te objavu i ocjenjivanje komentara. Potrebno je koristiti HTML, CSS i JavaScript za prikaz stranice korisnicima i PHP, SQL i Apache da poslužitelj ima mogućnost pohrane, provjere, obrade i slanja podataka.

2. PROGRAMSKI JEZICI

S obzirom da se radi o web aplikaciji, koriste se svi osnovni jezici korišteni za izgradnju web stranica. Na klijentskoj strani koristit će se HTML, CSS i JavaScript koji su podržani u svim modernim web preglednicima, dok će se s poslužiteljske strane koristiti PHP i SQL.

2.1 HTML

HyperText Markup Language (Jezik za označavanje hiperteksta) je osnovni jezik korišten za prikaz web stranica. HTML zapravo nije programski jezik već jezik koji omogućava dosljednu tj. standardiziranu prezentaciju web stranica klijentima.

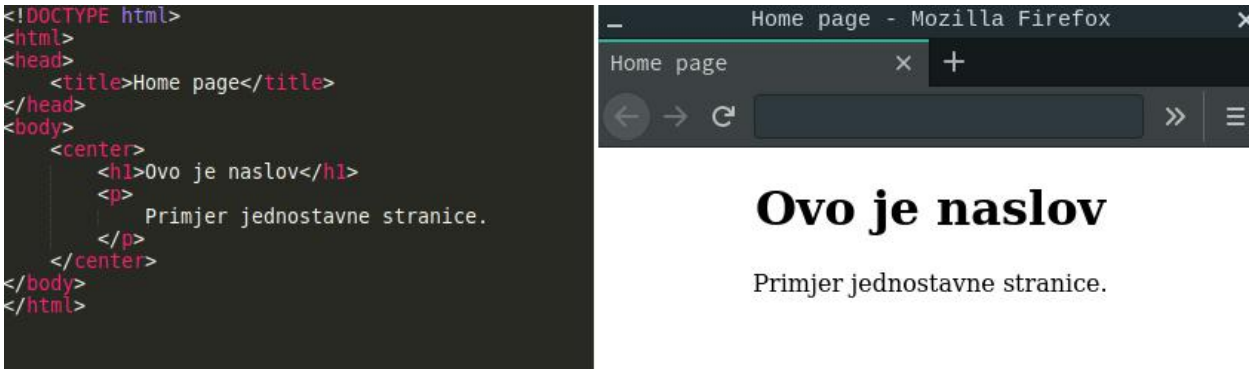
Ideja za standardizirani jezik kojim će biti omogućeno lakše prikazivanje web stranica nastala je 1989. u instituciji CERN.[1] Službena verzija objavljena je tek 1993. zajedno s prvim web preglednikom koji je podržavao procesiranje HTML formata.

Glavni cilj HTML jezika je standardizacija prikaza web stranica, tako da bez obzira kojim se web preglednikom stranica prikazuje ona će biti prikazana identično.

HTML jezik omogućava podjelu web stranice na oznake i blokove. Postoji veći broj različitih oznaka u HTML jeziku i svaka od njih ima svoju funkciju te se koristi u različite svrhe. Na primjer, oznaka govori web pregledniku da slijedeći tekst bude podebljan. Većinu oznaka je potrebno zatvoriti što se radi ponavljanjem iste oznake samo s kosom crtom, npr. . Sav tekst omeđen oznakama se naziva HTML blok. Također, postoje posebne oznake kao što je <br> koje se ne moraju zatvoriti.

Svaki HTML blok, tj. HTML element, može sadržavati atribute. HTML atributi daju dodatni opis HTML elemenata. Iako je moguće kreirati proizvoljne atribute, većinom se koriste standardizirani HTML atributi kao što su „title”, „class”, „id” i ostali. Jedan od mogućih atributa je „style” koji sadrži CSS kod kojim se omogućuje stiliziranje HTML bloka.

1 https://www.w3.org/History/1989/proposal.html



Slika 2.1. Lijevo je primjer HTML koda, desno je prikaz tog HTML dokumenta

Iako neki HTML blokovi imaju različite stilove tj. formatiraju tekst na drugačije načine, HTML sam po sebi nema mogućnost stiliziranja stranice, kao što je prikazano na slici 2.1. Zbog toga se razloga HTML proširuje s CSS jezikom.

2.2 CSS

Cascading Style Sheets je stilski jezik koji omogućuje detaljno stiliziranje web stranica.[2] CSS je nastao 1996. iz potrebe boljeg stiliziranja HTML dokumenata.[3]

Iako se CSS može zapisivati direktno unutar HTML dokumenta (Slika 2.2), to se obično izbjegava jer povećava kompleksnost razvoja stranice i uvodi dupliciranje elemenata. Tako da se u današnje vrijeme sve više teži separaciji CSS koda u zaseban dokument koji će određena stranica uključiti prilikom učitavanja (Slika 2.3).

2 https://www.w3.org/standards/webdesign/htmlcss#whatcss
3 https://www.w3.org/TR/1999/REC-CSS1-19990111

```html
<!DOCTYPE html>
<html>
<head>
    <title>Home page</title>
</head>
<body style="background-color: lightgray">
    <center>
        <h1 style="color: red">Ovo je naslov</h1>
        <p style="color: orange">
            Primjer jednostavne stranice.
        </p>
    </center>
</body>
</html>
```

Slika 2.2 primjer pisanja CSS stilova unutar HTML dokumenta

```html
<!DOCTYPE html>
<html>
<head>
    <title>Home page</title>
    <link rel="stylesheet" type="text/css" href="css_file.css">
</head>
<body>
    <center>
        <h1>Ovo je naslov</h1>
        <p class="example-text">
            Primjer jednostavne stranice.
        </p>
    </center>
</body>
</html>
```

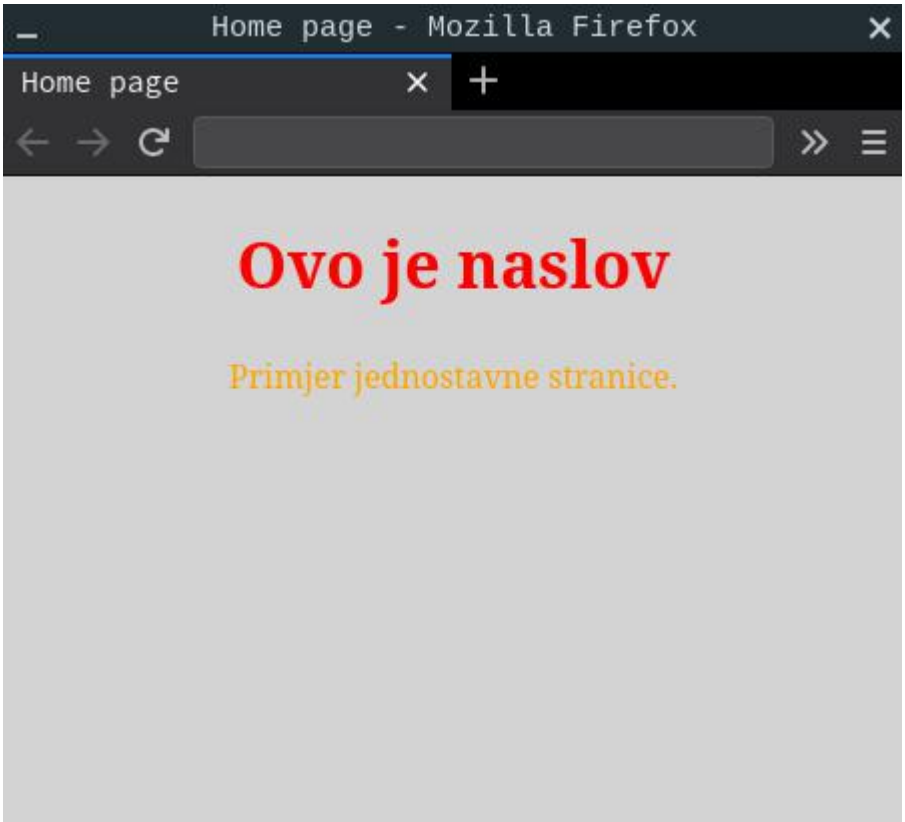
```css
body{
    background-color: lightgray;
}

h1{
    color: red;
}

.example-text{
    color: orange;
}
```

Slika 2.3 lijevo je HTML dokument s uključenim CSS dokumentom, desno je posebni CSS dokument

U oba slučaja, krajnja web stranica će izgledati jednako (Slika 2.4). Razlika je u tome što je web stranicu lakše održavati ukoliko je razdijeljena na manje elemente te se iz toga razloga svo stiliziranje odvaja u zasebnu CSS datoteku, ili više CSS datoteka.



Slika 2.4 primjer stilizirane web stranice

Jedna CSS struktura unutar CSS dokumenta, koja se odnosi na neki HTML element, naziva se skup pravila (eng. CSS ruleset). Taj skup se sastoji od selektora, deklaracije, svojstva i vrijednosti svojstva[4]. Selektor označava na koji će HTML element, obično određen identifikacijskom riječi ili klasom, CSS skup djelovati. Svojstvo određuje koje svojstvo će CSS kod stilizirati, a vrijednost svojstva određuje kako će se stilizirati. Jedan par svojstva i njegove vrijednosti zove se deklaracija. Na primjeru slike 2.3, „h1" je selektor, „color" je svojstvo, „red" je vrijednost svojstva a „color: red;" je deklaracija.

2.2.1 Bootstrap

Sa sve većim rastom kompleksnošću web stranica raste i količina stiliziranja koju web dizajner mora napraviti. Kako programeri ne bi trošili previše vremena ponavljajući isti kod na svakoj stranici na kojoj rade, počeli su se razvijati CSS okviri koji imaju već napravljen CSS kod koji se vrlo lako može uključiti u neku web stranicu. Jedan od takvih okvira je Bootstrap CSS. Cilj Bootstrap biblioteke je dozvoliti brzi i lagani *responsive design* (hrv. prilagođavajući dizajn), tj.

4 https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics

omogućiti da se web stranica korektno prikaže na bilo kojoj širini ekrana.[5] Na taj način se osigurava da korisnici manjih ekrana kao što su korisnici laptopa i mobitela imaju jednako kvalitetno iskustvo koristeći web stranicu kao i korisnici stolnih računala. To je jako bitno u današnje vrijeme jer se većina internetskog prometa odvija preko pametnih telefona.[6]

2.3 JavaScript

JavaScript je skriptni programski jezik pomoću kojega se web stranicama daje raznolike funkcionalnosti. Sami CSS i HTML ne daju web stranici nikakvu funkcionalnost, nego samo izgled. Iz potrebe da se izrade web stranice koje nisu samo statične nego mogu izvršavati nekakve napredne funkcije, Brendan Eich je na zahtjev organizacije Netscape[7] stvorio jezik Livescript koji je kasnije nazvan JavaScript. Brendan Eich je kasnije potvrdio da je jezik nazvao JavaScript jer je u to vrijeme jezik Java bio jako popularan, te je nazivom JavaScript htio dobiti interes ostalih programera.[8]

JavaScript se izvršava na korisničkom uređaju, unutar korisnikovog web preglednika. JavaScript je uz CSS, jezik koji web stranice čini prikladnim krajnjem koristiku jer ima sposobnost vraćanja povratnih informacija i ažuriranja izgleda stranice, što web stranice čini prikladnijima za korištenje. Uz pomoć JavaScript jezika stranice dobivaju sposobnost interakcije s korisnikom i izvršavanja naprednih skripti, ažuriranja sučelja i slično. Samim time web stranice se mogu ponašati kao funkcionalni programi tj. aplikacije. U današnje vrijeme se upravo zbog toga razvijaju takozvane progresivne web aplikacije[9] koje omogućuju integraciju web stranice s operacijskim sustavom korisničkog uređaja.

---

5 https://getbootstrap.com/docs/4.5/getting-started/introduction/
6 https://gs.statcounter.com/os-market-share/all/
7 http://speakingjs.com/es5/ch04.html
8 https://www.youtube.com/watch?v=XOmhtfTrRxc&t=2m5s
9 https://web.dev/what-are-pwas/

```
<script type="text/javascript">
    function validate(){
        if (document.getElementById("my_name").value == ""){
            alert("Name field empty!");
        }
        if (document.getElementById("email").value.length < 2){
            alert("Email too short");
        } else if (document.getElementById("email").value.indexOf("@") < 0){
            alert("Missing @");
        } else if (document.getElementById("email").value.split("@")[1].length < 2){
            alert("Domain error");
        } else if (!document.getElementById("email").value.split("@")[1].includes(".")){
            alert("Missing .");
        } else {
            if (document.getElementById("email").value.split("@")[1].split(".")[1].length < 2){
                alert("Incorrect tld");
            } else {
                confirm("Your name is " + document.getElementById("my_name").value.split(" ")[0] +
                ", your last name is " + document.getElementById("my_name").value.split(" ")[1] +
                ", your email is " + document.getElementById("email").value);
            }
        }
    }
</script>
```

Slika 2.5 primjer JavaScript koda za provjeru valjanosti podataka

Na slici 2.5 se nalazi primjer JavaScript koda koji korisniku vraća povratnu informaciju o valjanosti unešenih podataka. JavaScript funkcije se stavljaju unutar <script> blokova te se mogu pozivati automatski ili prilikom interakcije s nekim HTML elementima.

Kod JavaScript jezika važno je paziti što se točno daje korisniku na izvršavanje. S obzirom da se na korisnikov uređaj šalje otvoreni JavaScript kod, korisnik taj kod može izmijeniti ili blokirati. U slučaju da je maliciozni napadač preuzeo korisnički uređaj, ili u slučaju da je sam korisnik maliciozan, on ima mogućnost izvršavanja izmijenjenog koda te slanja netočnih rezultata poslužiteljskom serveru ili samom klijentu. Zbog toga se bilo što što bi moglo uzrokovati sigurnosne probleme korisnika treba izvršavati na poslužiteljskoj strani.[10]

2.4 PHP

PHP je općenamjenski skriptni jezik koji se izvršava na strani poslužitelja i omogućuje poslužitelju vraćanje dinamičnih odgovora korisniku te obavljanje ostalih funkcionalnosti potrebnih za web stranicu, a koje se ne bi trebale obavljati na strani korisnika. Rasmus Lerdorf je 1994. stvorio PHP kao alat za održavanje svoje osobne web stranice tako da je originalno PHP bio skraćenica od Personal Home Page.[11] Nakon što je Lerdorf objavio koncept za PHP,

10 https://www.serviceobjects.com/blog/why-you-should-never-put-sensitive-data-in-your-javascript/

11 https://www.php.net/manual/en/history.php.php

zajednice web programera počele su sve više koristiti PHP na svojim projektima i dodavati nove funkcionalnosti.[11]

PHP kod se zapisuje unutar PHP datoteka, tj. unutar HTML datoteka. PHP blokovi započinju oznakom „<?php” te završavaju oznakom „?>”. Sve između tih dviju oznaka se smatra PHP kodom. Sintaksa PHP koda je slična C jeziku.

```
<!DOCTYPE html>
<html>
<head>
    <title>PHP Example</title>

</head>

<?php
    if((int)date("h") < 9 || (int)date("h") > 19)
        echo "<body style='background-color: black; color: white;'>";
    else
        echo "<body style='background-color: white; color: black;'>";
?>

</body>
</html>
```

Slika 2.6 primjer PHP koda

Na slici 2.6 prikazana je PHP skripta koja na web stranicu primjenjuje CSS stil ovisno o vremenu na poslužitelju. Prije 9 sati ujutro i poslije 19 sati će stranica poprimiti crnu pozadinu s bijelim slovima.

PHP ima mogućnost dodavanja HTML, CSS ili JavaScript elemenata na web stranicu koristeći naredbu „echo”. Osim toga, PHP se može koristiti za interakciju s bazom podataka, izvršavanje logičkih i matematičkih skripti te se može koristiti i van web okruženja.[12]

12 https://github.com/jolicode/php-ar-drone

2.5 SQL

Structured Query Language je jezik specifično stvoren za upravljanje bazama podataka te podacima unutar baza. SQL baze podataka mogu sadržavati raznolike tipove podataka. Na slici 2.7 je prikazan primjer kreiranja tablice koja sadržava korisničke podatke.

```
CREATE TABLE 'users'(
'id' int(8) NOT NULL,
'email' varchar(50) NOT NULL,
'password' varchar(50) NOT NULL,
'name' varchar(40) DEFAULT '',
'info' varchar(500) DEFAULT '',
'is_admin' int(1) NOT NULL DEFAULT 0
)
```

Slika 2.7 primjer kreiranja tablice korisničkih računa

S obzirom da klijenti nemaju, niti bi trebali imati, direktan pristup bazi podataka poslužitelja, interakcija između baze podataka i krajnjeg korisnika ovisi o skriptama sa strane poslužitelja. Za pristup SQL bazi podataka može se koristiti PHP.

```
<?php
  define('DB_SERVER', 'localhost');
  define('DB_USERNAME', 'root');
  define('DB_PASSWORD', '');
  define('DB_NAME', 'echo');

  $db = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);
  session_start();

  $user_check = $_SESSION['login_user'];

  $ses_sql = mysqli_query($db,"select email, isadmin from users where
    email = '$user_check' ");

  $row = mysqli_fetch_array($ses_sql,MYSQLI_ASSOC);

  $login_session = $row['email'];
  $adminaccount = $row['isadmin'];
?>
```

Slika 2.8 primjer povezivanja na bazu koristeći PHP sjednicu

Na slici 2.8 je prikaz PHP datoteke koja održava sjednicu (eng. *session*) između klijenta i poslužitelja, te inicijalizira pristup bazi podataka, tj. korisnički uređaj se veže s postojećim

računom na poslužitelju. Sjednica se mora uključiti u svakom PHP dokumentu web aplikacije jer će inače korisnik izgubiti indirektnu povezanost na bazu podataka, tj. izgubiti će se veza između korisnika i njegovog računa. Kada se osigura sjednica, poslužitelj korisniku može dozvoliti razmjenu informacija. Na primjer, slanje poruke na poslužitelja ili promjena informacija vlastitog računa. Sjednica se može i održati koristeći kolačiće (eng. Cookies) koji omogućuju spremanje nekih informacija na klijentovom web pregledniku, no krajnju provjeru tih informacija mora osigurati poslužitelj.

```php
<?php
if($_SERVER["REQUEST_METHOD"] == "POST"){
    $myoldpass = mysqli_real_escape_string(
        $db, $_POST['password-old']);
    $mypassword = mysqli_real_escape_string(
        $db, $_POST['password']);
    $confirmpass = mysqli_real_escape_string(
        $db, $_POST['password-confirm']);

    $sql = "SELECT * FROM users
    WHERE email = '$login_session'
    AND password = '".sha1($myoldpass)."'";
    $result = mysqli_query($db, $sql);

    if(mysqli_num_rows($result) == 0){
        echo "Incorrect old password or email.";
    } else if(strlen($mypassword) < 10) {
        echo "Password too short";
    } else if($mypassword != $confirmpass) {
        echo "Passwords don't match.";
    } else {
        $sql = "UPDATE users
            SET password = '".sha1($mypassword)."'
            WHERE email = '$login_session'";
        mysqli_query($db, $sql);
        echo "Success";
        header("refresh:1 url=login.php");
    }
}
?>
```

Slika 2.9 primjer korištenja PHP za promjenu korisničke zaporke

Na slici 2.9 se može vidjeti PHP skripta koja korisniku omogućuje promjenu zaporke računa i način na koji poslužitelj provjerava ispravnost primljenih podataka te usporedbu poslanih podataka s podacima spremljenim u poslužiteljevoj bazi podataka. Poslužitelj mora biti oprezan pri povezivanju klijenata na bazu podataka jer je moguće da maliciozni klijent pokuša pristupiti

podacima kojima ne bi trebao.[13] Na primjer, ako se pristupa SQL bazi podataka preko PHP skripte, potrebno je koristiti funkciju *mysqli_real_escape_string()* jer ona osigurava da klijent neće poslati svoj SQL zahtjev[14] te tako sprječava ubacivanje SQL naredbi sa strane korisnika.

13 https://www.w3schools.com/sql/sql_injection.asp
14 https://www.php.net/manual/en/mysqli.real-escape-string.php

3. WEB APLIKACIJA IZ POSLUŽITELJSKE PERSPEKTIVE

Poslužiteljeva zadaća je odgovarati na klijentske zahtjeve. Kada klijent obavi HTTP upit za dohvaćanje nekog resursa, poslužitelj mu mora poslati taj resurs ukoliko je javan i postoji, ili vratiti odgovarajuću grešku. U ovome projektu je za sve to zadužen Apache HTTP Server i PHP jezik. U ovome radu koristit će se aplikacijski paket XAMPP koji sadrži Apache HTTP Server, PHP i MariaDB koji su potrebni, no mogu se koristiti i isti ručno konfigurirani individualni paketi ili njihove alternative ukoliko su kompatibilne s korištenim kodom.

3.1 XAMPP

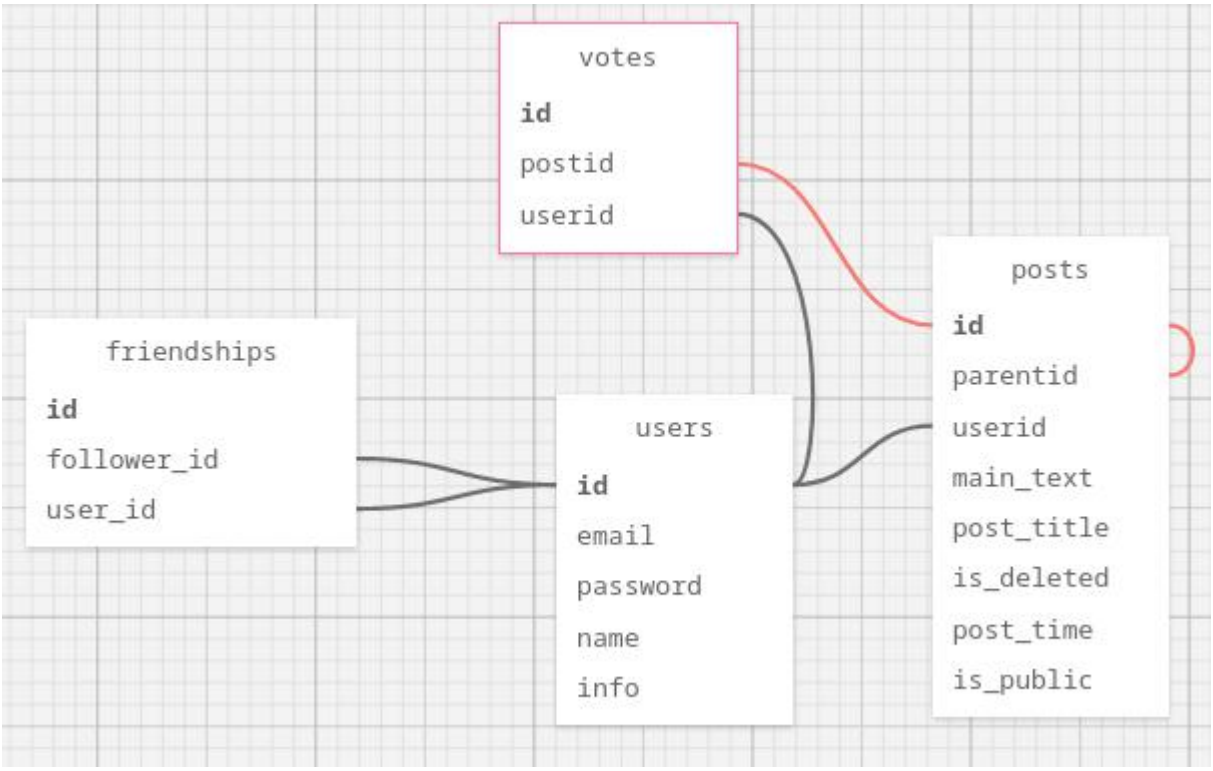
Apache je web server otvorenog koda. Njegova zadaća je dati poslužiteljskom računalu funkcionalnost odgovora na HTTP zahtjeve korisnika tj. osigurava dostavljanje web stranica i njihovog sadržaja krajnjem korisniku. Iako počinje gubiti na popularnosti zbog HTTP servera boljih performansi[15], Apache je još uvijek najpopularniji HTTP server[16].

Kao što je ranije spomenuto, PHP jezik se koristi kako bi poslužitelj dinamički slao podatke krajnjem korisniku. S obzirom da je riječ o društvenoj mreži, glavna zadaća koju PHP obavlja jest ispisivanje javnih komentara i mogućnost posjećivanja korisničkih profila te registracija i prijava u vlastiti profil. Ukoliko je korisnik prijavljen u svoj račun, poslužitelj mora omogućiti dodavanje prijatelja, ispis određenih privatnih komentara, ocjenjivanje komentara, postavljanje komentara te brisanje i uređivanje vlastitih komentara.

Svi se podaci, kao što su informacije o računima i porukama, spremaju u bazu podataka. Unutar XAMPP paketa koristi se MariaDB sustav za upravljanjem SQL bazom podataka. Ovaj projekt se sastoji od 4 tablice (Slika 3.1).

15 https://www.nginx.com/resources/wiki/start/topics/examples/likeapache-htaccess/

16 https://news.netcraft.com/archives/2020/04/08/april-2020-web-server-survey.html#graph04_202004_00



Slika 3.1 Shema baze podataka

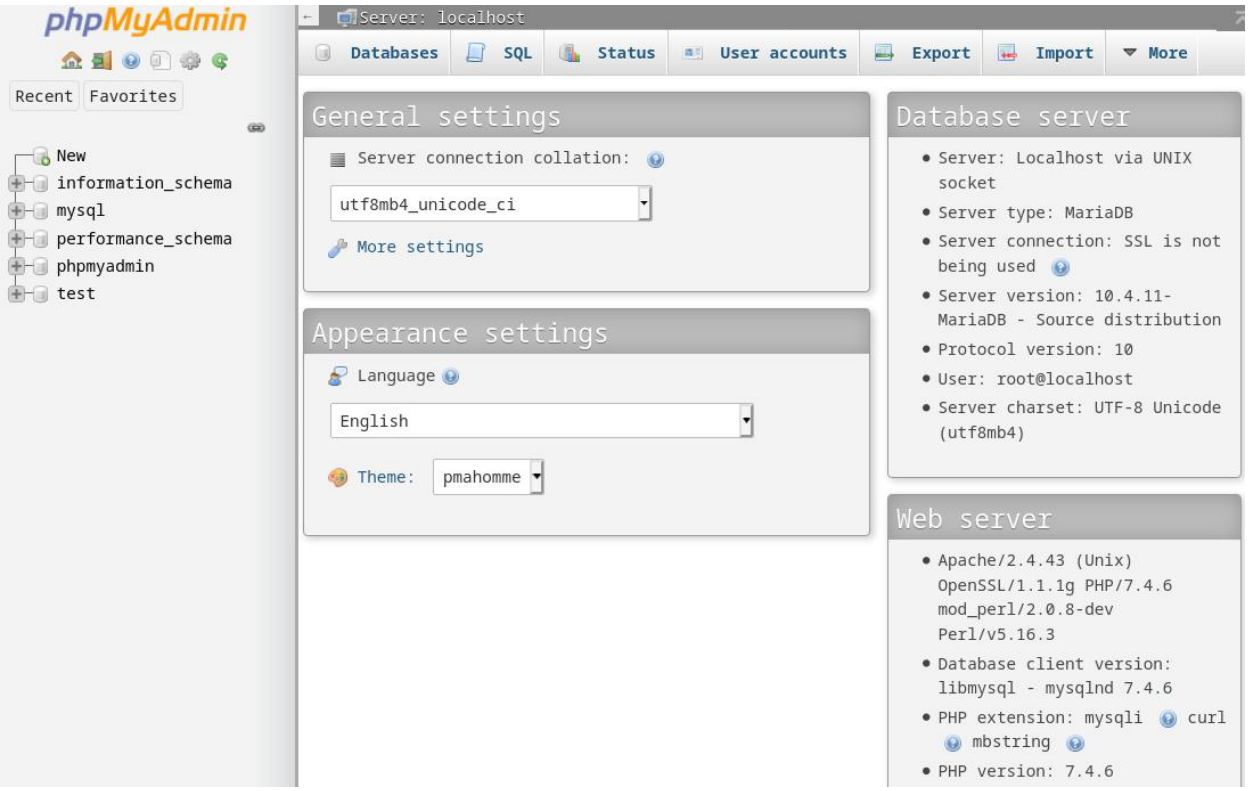
Na slici 3.1 prikazane su tablice korištene u ovome projektu. Tablica „users” sadrži podatke o korisnicima. Za pohranu korisničke zaporke koristi se SHA-1 funkcija umjesto pohrane same zaporke u obliku otvorenog teksta. Na ovaj način se osigurava da ukoliko neovlašteni korisnik dobije pristup bazi podataka, on neće imati pristup korisničkim zaporkama nego samo njihovim hash vrijednostima. Hash funkcije su jednosmjerne što znači da nije moguće dobiti originalni tekst ukoliko je poznata hash vrijednost.

Tablica „posts” sadrži podatke o objavama. Ona se povezuje s ključem iz tablice „users” kako bi se korisnici povezali sa svojim objavama.

Tablica „votes” sadrži vezu između korisnika i objava. U ovoj tablici se drže informacije o ocjenama objava. Koristeći PHP onemogućava se korisnicima glasanje za svoje komentare te se onemogućava korisnicima višestruko glasanje.

Tablica „friendships” sadrži informacije o tome koje profile određeni korisnici prate tj. imaju među prijateljima. Koristeći PHP onemogućava se korisnicima da prate sami sebe. Ova se tablica koristi za ispis privatnih poruka. Samo korisnici koje određeni korisnik prati mogu vidjeti privatne poruke tog korisnika.

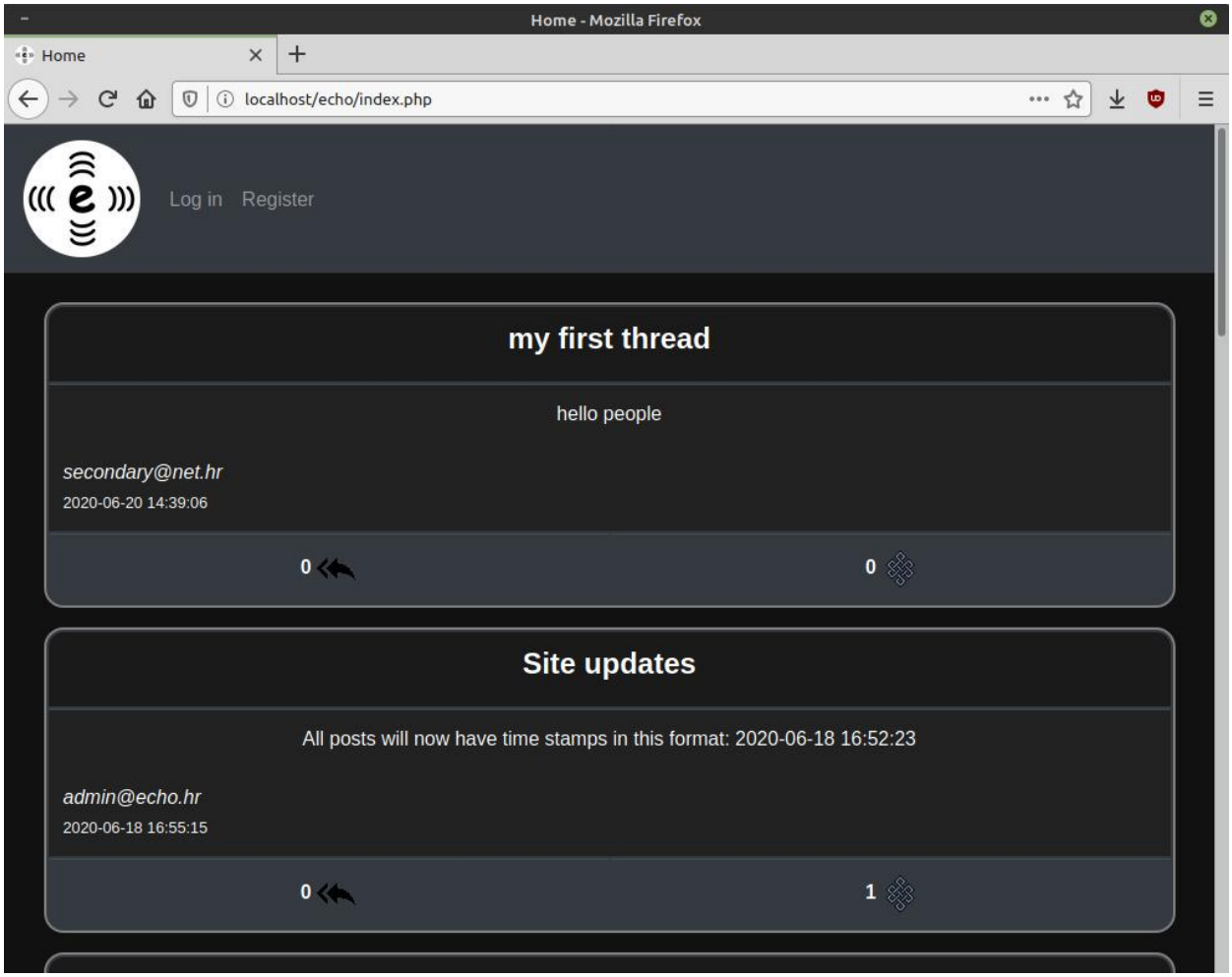
Na slici 3.2 je prikazano grafičko web sučelje programskog paketa XAMPP koje omogućava administratoru direktan pristup bazama podataka poslužitelja. Koristeći ovo sučelje, administrator ima mogućnost ručnog dodavanja vrijednosti u tablice, te mijenjanja sadržaja tablica. Također, ima mogućnost dodavanja tablica i samih baza podataka. S obzirom da su zaporke zaštićene SHA-1 funkcijom, administrator korisničke zaporke ne može vidjeti.



Slika 3.2 phpMyAdmin sučelje

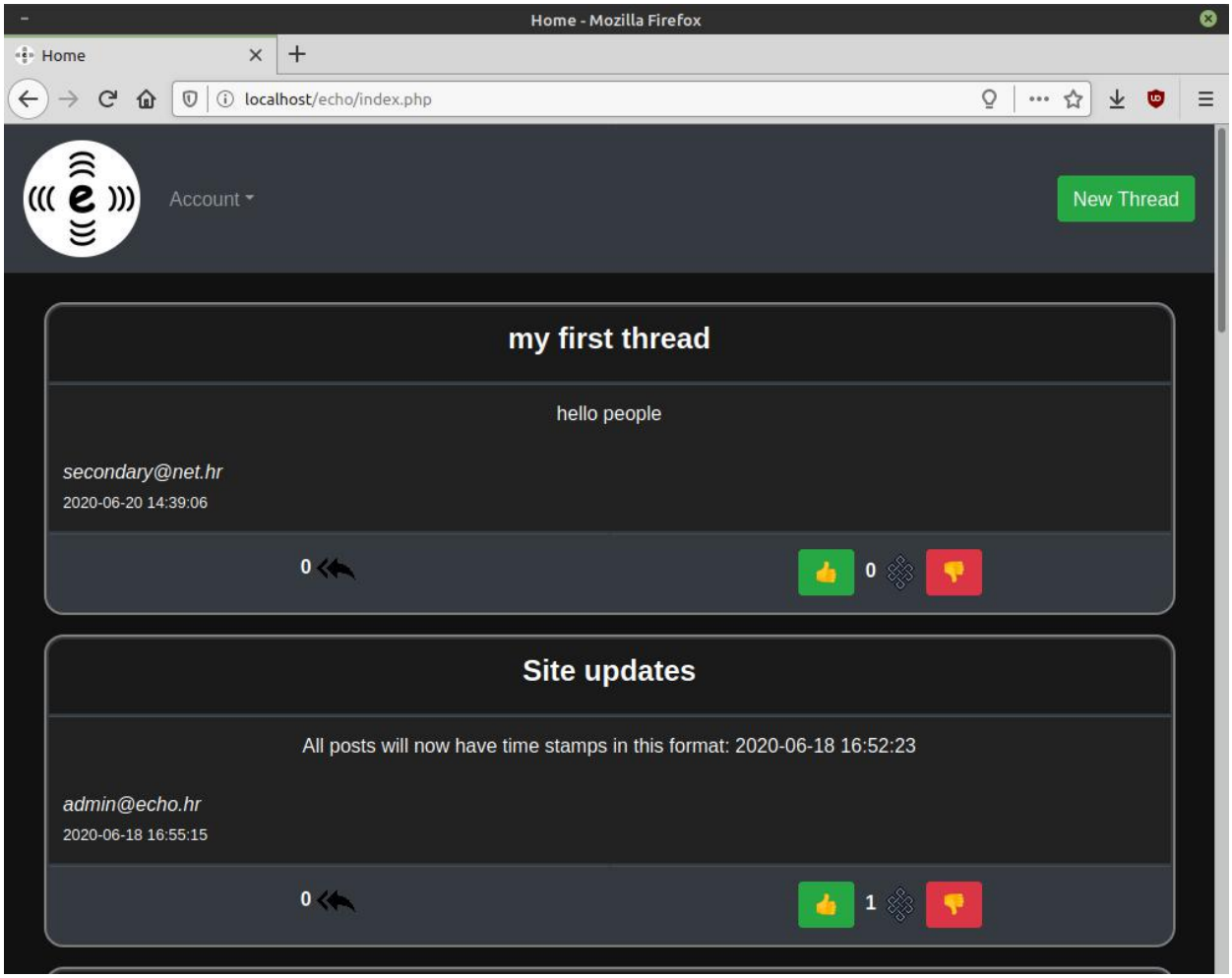
4. WEB APLIKACIJA IZ KLIJENTSKE PERSPEKTIVE

Klijenti se na web stranicu povezuju upisivanjem domene ili IP adrese stranice unutar web preglednika. Glavna stranica ovoga projekta je stranica s javnim obavijestima. Klijenti koji nisu registrirani samo mogu pregledavati javne obavijesti i stvoriti novi račun (Slika 4.1).



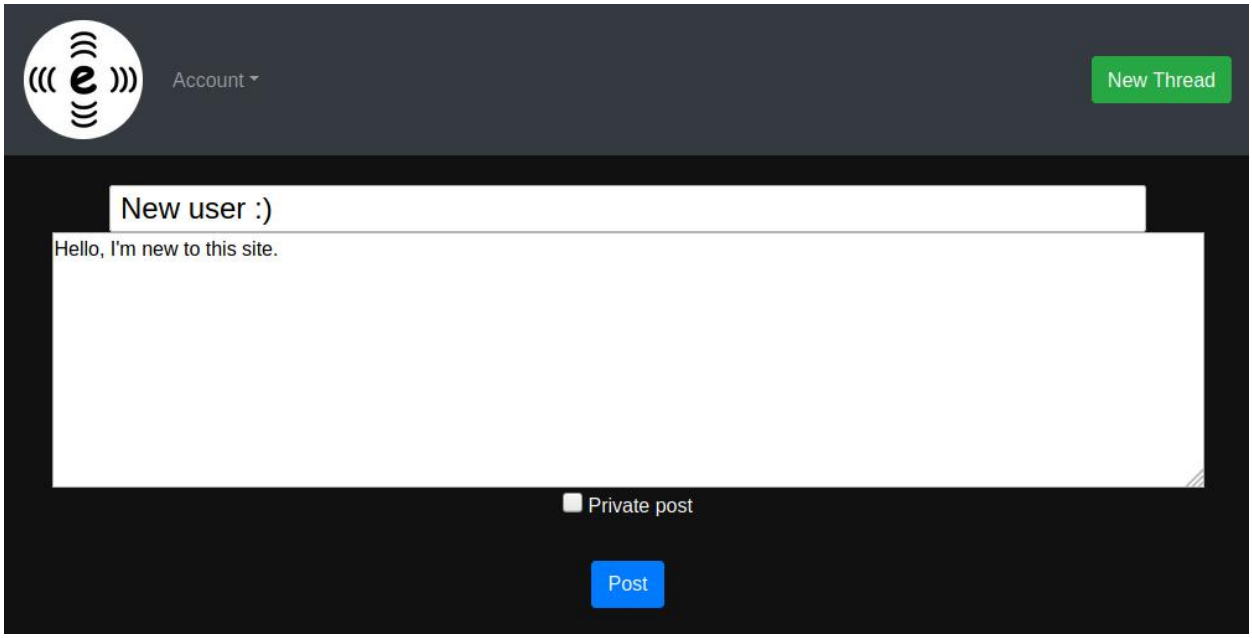
Slika 4.1 Početna stranica za neprijavljenog korisnika

Registrirani korisnici imaju opciju ocjenjivanja pojedinih poruka, praćenja drugih registriranih korisnika i najbitnije od svega dodavanje novih komentara tj. poruka na stranicu (Slika 4.2).



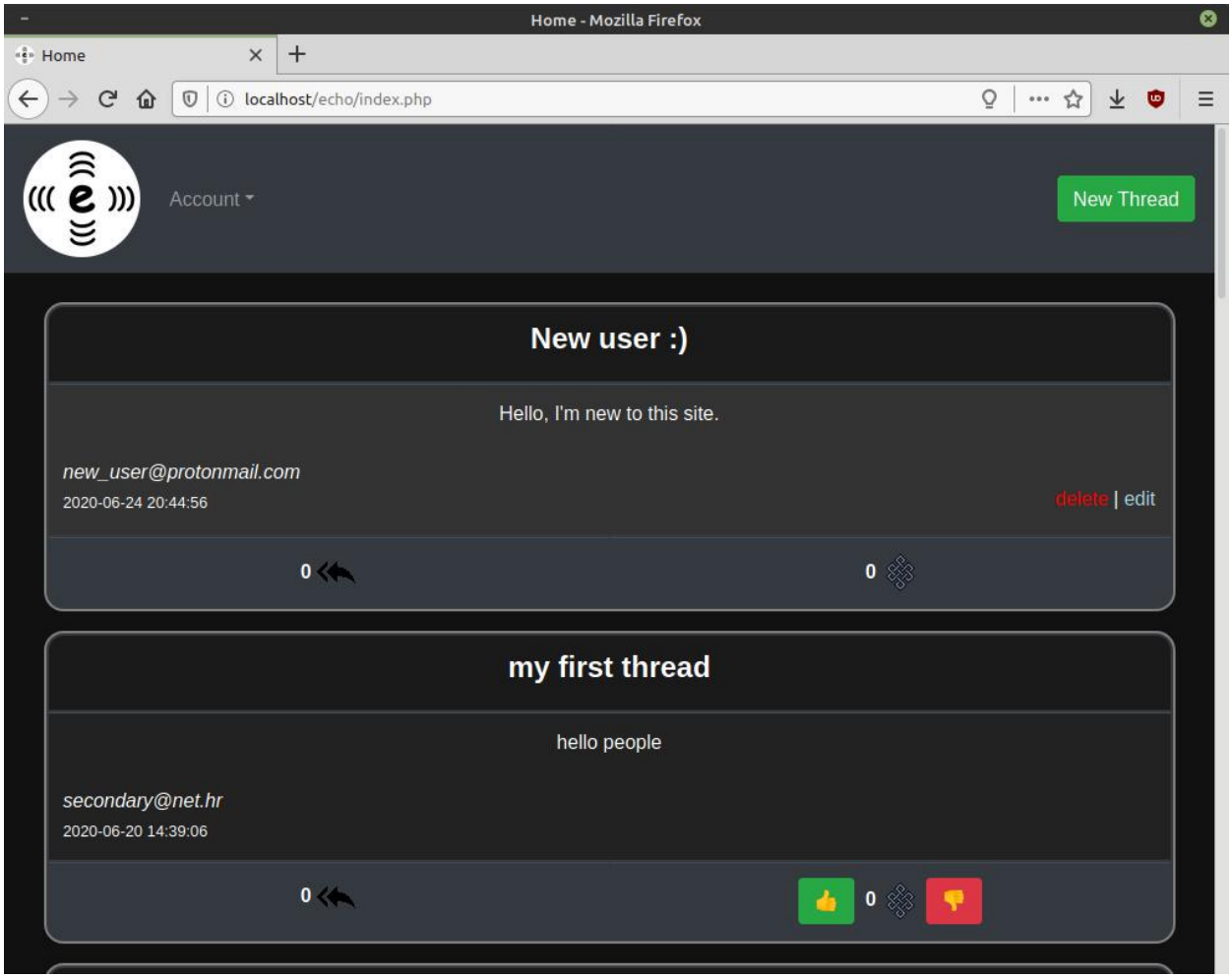
Slika 4.2 Početna stranica za prijavljenog korisnika

Pritiskom tipke „New Thread” klijentu se, koristeći PHP, šalje HTML dokument unutar kojeg može upisati naslov poruke, glavni tekst te odabrati hoće li poruka biti javna ili će se samo poslati kontaktima koje klijent prati što je prikazano na slici 4.3.



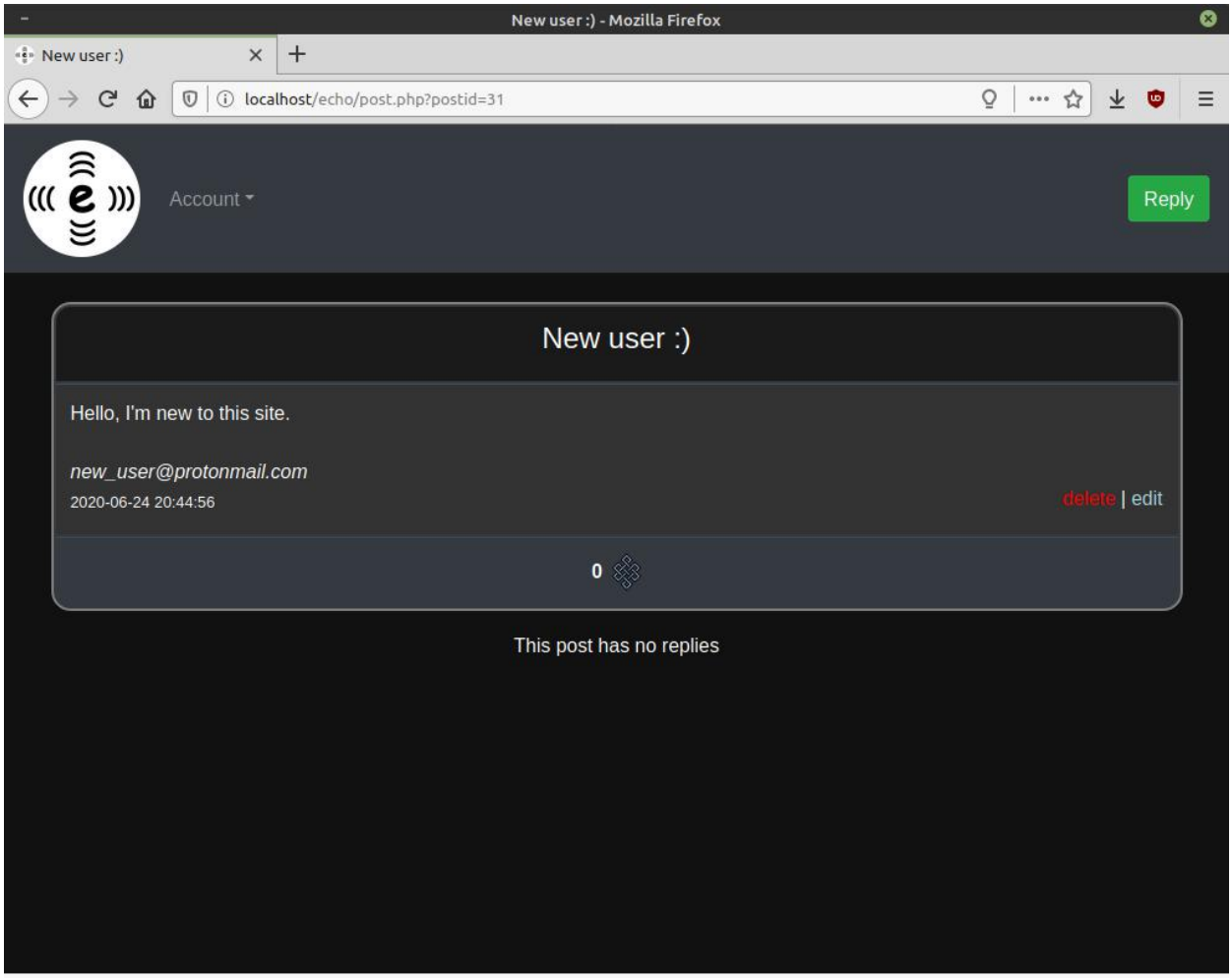
Slika 4.3 Kreiranje nove obavijesti

Nakon što klijent pošalje poruku ona se dodaje u bazu podataka. Svi korisnici koji imaju pristup toj poruci ju mogu vidjeti te registrirani korisnici mogu odgovarati na nju ili ju ocijeniti (*like*/*dislike*). Stranica pruža povratnu informaciju o broju odgovora na pojedinu glavnu poruku tj. *thread* (hrv. nit). Pritiskom na indikator odgovora korisnik može vidjeti sve odgovore na odabranu poruku (Slika 4.4). Na ovaj način se dozvoljava ugniježđena komunikacija između korisnika.



Slika 4.4 Prikaz nove poruke

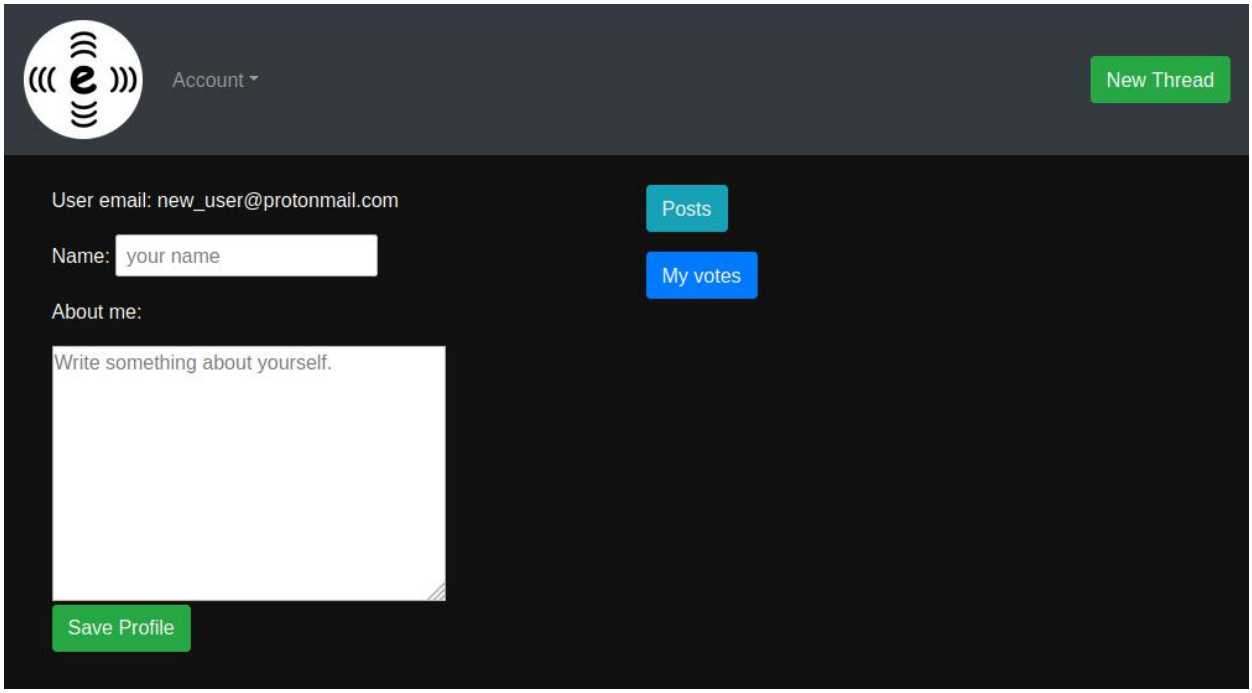
Na slici 4.4 se vidi kako su korisničke vlastite poruke označene drukčijom bojom radi bolje preglednosti. Prijavljeni korisnik može uređivati vlastite komentare pritiskom na tipku „edit” ili obrisati svoje komentare pritiskom na tipku „delete”. Pritiskom na naslov poruke ili na brojač odgovora korisnik je usmjeren na odabranu objavu, te može vidjeti komentare na tu objavu (Slika 4.5).



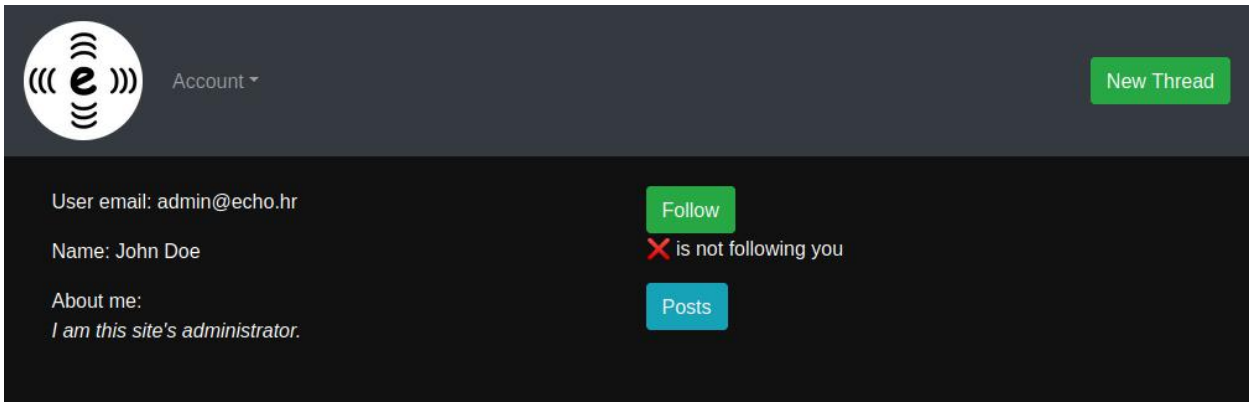
Slika 4.5 Prikaz odabrane poruke

Na poruci je također vidljivo korisničko ime korisnika koji ju je poslao, te svaki klijent može pritiskom na korisničko ime otići na profil drugog korisnika. Na korisničkom profilu se mogu vidjeti sve javne informacije korisnika, te se mogu prikazati sve javne objave toga korisnika.

Prijavljeni korisnik može dodavati informacije o sebi na vlastitom profilu (Slika 4.6), dok na profilima ostalih korisnika može pratiti te korisnike i vidjeti dali ga ti korisnici prate (Slika 4.7).

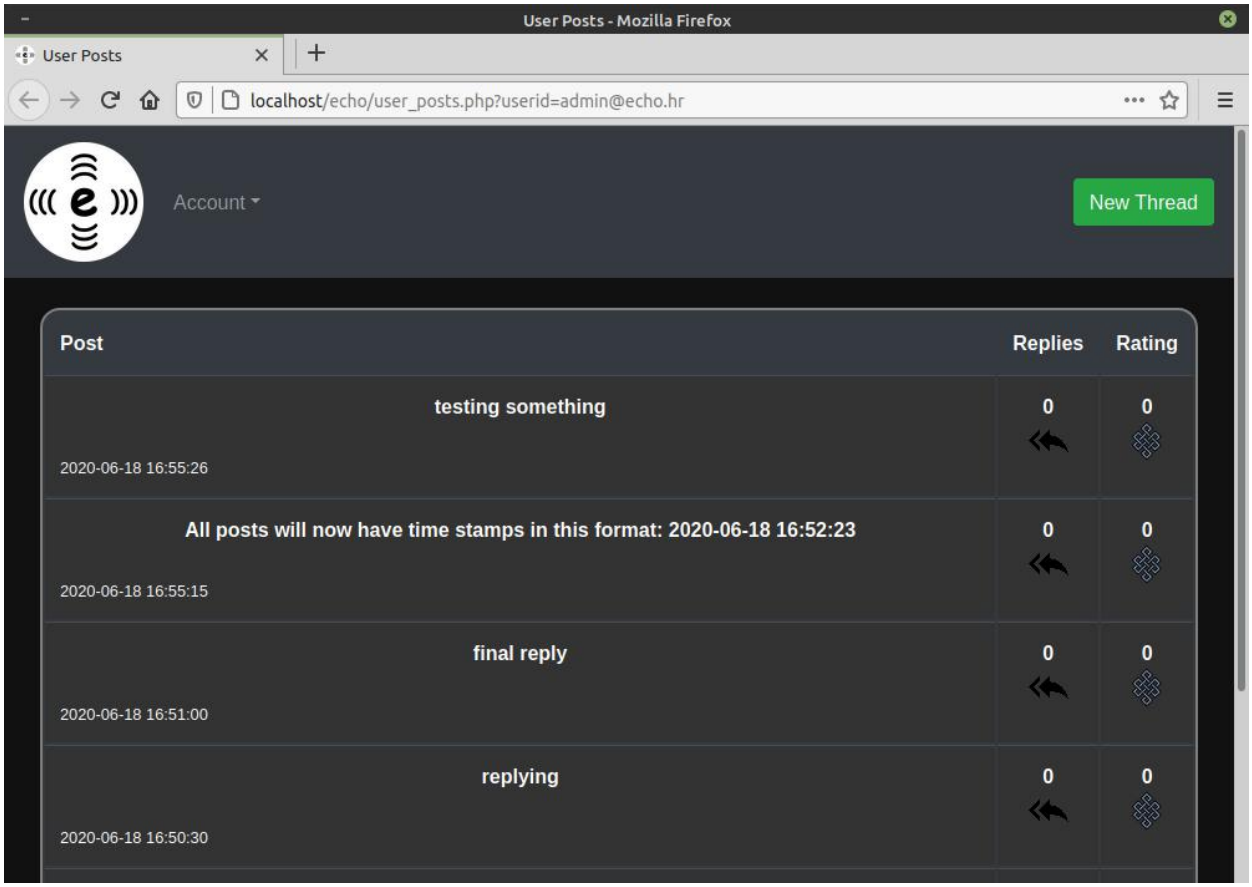


Slika 4.6 Prikaz vlastitog profila

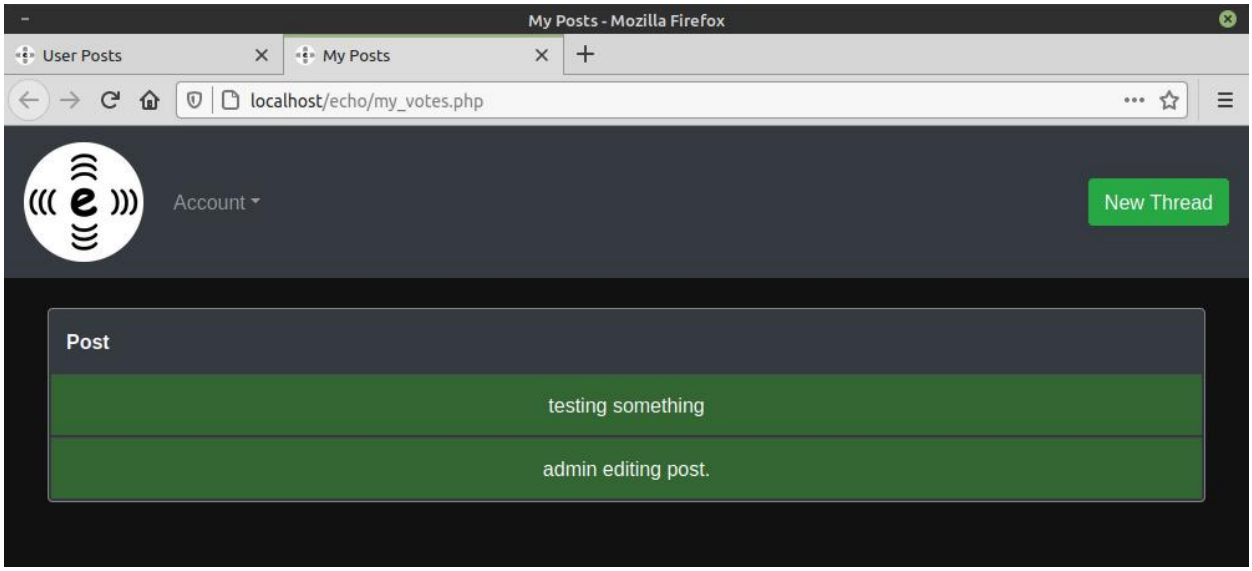


Slika 4.7 Prikaz tuđeg profila

Dok se posjećuje tuđi profil moguće je, pritiskom na tipku „Posts”, vidjeti sve prethodne komentare toga korisnika kao što je prikazano na slici 4.8. Svim vlastitim komentarima se pristupa na isti način, ali može se koristiti i padajući izbornik. Pritiskom na tipku „My Votes” korisnik dobije prikaz svih komentara na koje je glasao tj. koje označio pozitivnom ili negativnom ocjenom kao što je prikazano na slici 4.9. Komentari su označeni zelenom bojom ako im je dana pozitivna ocjena, a crvenom bojom ako im je dana negativna ocjena.

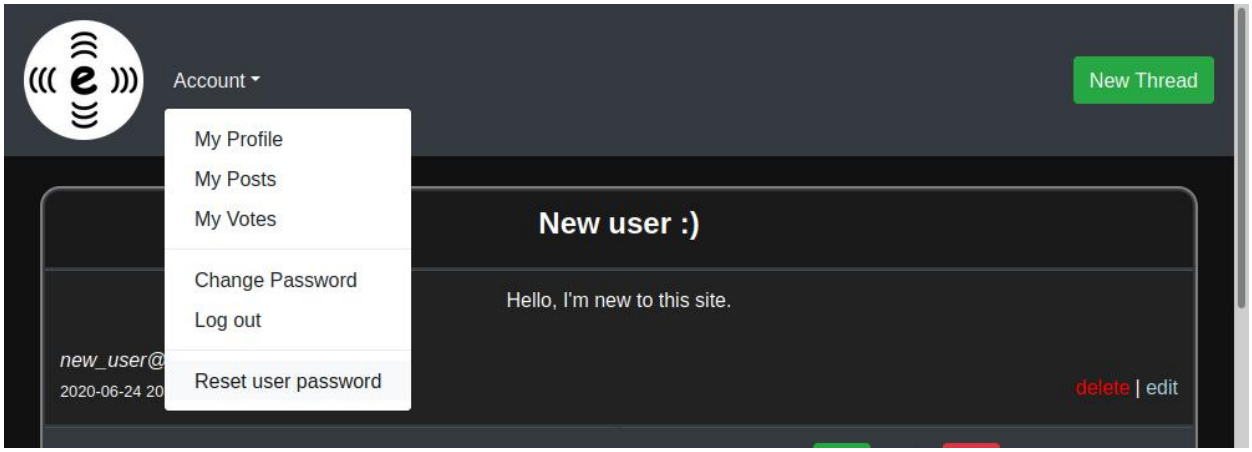


Slika 4.8 Prikaz svih vlastitih komentara



Slika 4.9 Prikaz komentara koje je korisnik ocjenio

Osim običnog korisnika postoji i uloga moderatora. Moderatori, tj. administratori mogu mijenjati ili brisati komentare ostalih korisnika. Moderatori mogu i mijenjati zaporku ostalih korisnika koristeći padajući izbornik kao što je prikazano na slici 4.10. Uloga moderatora postoji kako bi se pomoglo korisnicima u slučaju da zaborave zaporku i kako bi se brisali neželjeni komentari koji mogu naštetiti ostalim korisnicima.



Slika 4.10 Prikaz padajučeg izbornika

5. ZAKLJUČAK

Cilj ovoga projekta je napraviti jednostavnu društvenu mrežu te osposobiti web aplikaciju kojom korisnici mogu njoj pristupiti.

Naravno, već postoje različite društvene mreže kao što su Facebook, Twitter, Reddit, Mastadon, Gab, Minds, Diaspora, Manyverse, itd. Također, postoje mreže koje ciljaju određenu skupinu ljudi te kao takve mogu specijalizirati svoje usluge ciljanoj grupi ljudi (GitHub, Tinder, Youtube, Instagram).

S obzirom da nije moguće zadovoljiti potrebe svih korisnika web usluga, cilj ovoga projekta nije bio stvoriti revolucionarnu društvenu mrežu nego samo educirati zainteresirane kako stvoriti svoju mrežu te dati uvid na moguće probleme kreiranja društvene mreže ili bilo kakve web aplikacije.

Trenutna verzija web aplikacije se može dalje unaprijediti, a sav kod je postavljen na GitLab. Poveznica na projekt je u poglavlju PRILOZI.

LITERATURA

[1]. https://www.w3.org/History/1989/proposal.html

[2]. https://www.w3.org/standards/webdesign/htmlcss#whatcss

[3]. https://www.w3.org/TR/1999/REC-CSS1-19990111

[4]. https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics

[5]. https://getbootstrap.com/docs/4.5/getting-started/introduction/

[6]. https://gs.statcounter.com/os-market-share/all/

[7]. http://speakingjs.com/es5/ch04.html

[8]. https://www.youtube.com/watch?v=XOmhtfTrRxc&t=2m5s

[9]. https://web.dev/what-are-pwas/

[10]. https://www.serviceobjects.com/blog/why-you-should-never-put-sensitive-data-in-your-javascript/

[11]. https://www.php.net/manual/en/history.php.php

[12]. https://github.com/jolicode/php-ar-drone

[13]. https://www.w3schools.com/sql/sql_injection.asp

[14]. https://www.php.net/manual/en/mysqli.real-escape-string.php

[15]. https://www.nginx.com/resources/wiki/start/topics/examples/likeapache-htaccess/

[16]. https://news.netcraft.com/archives/2020/04/08/april-2020-web-server-survey.html#graph04_202004_00

SAŽETAK

U ovome je projektu napravljena web aplikacija za društvenu mrežu. Svrha ove aplikacije je kreirati alternativnu društvenu mrežu gdje korisnici mogu komunicirati na jednostavan način koristeći stolna računala ili pametne mobitele. Korisnička se komunikacija svodi na objavljivanje javnih ili privatnih obavijesti te omogućavanje ostalim korisnicima da reagiraju na obavijesti. Ovu web aplikaciju je vrlo jednostavno implementirati s poslužiteljske strane. Za kreiranje projekta korišten je GNU/Linux operacijski sustav te alat XAMPP koji pruža sve potrebne programe i biblioteke za izvođenje usluge (Apache HTTP Server, MariaDB bazu podataka, PHP i Perl prevoditelj). Za demonstraciju aplikacije korišten je web preglednik Firefox.

Ključne riječi: društvena mreža, komunikacija, poruke, web aplikacija

ABSTRACT

Social network web application

Developing a web application for a social network is explained in this paper. The purpose of this application is being an alternative social network where the users have a simple way to communicate using their desktop computers or smartphones. User communication consists of publishing public or private posts as well as allowing other users to react to the posts. This web application is easy to implement on the server. The project was created using the GNU/Linux operating system and the XAMPP tool which contains all software and libraries necessary for creating and maintaining the service (Apache HTTP Server, MariaDB, PHP and Perl interpreters). Firefox web browser was used to demonstrate the application.
Keywords: communication, messages, social network, web application

PRILOZI

Svi prilozi se nalaze na CD-u unutar direktorija PRILOZI.

Sav programski kod dostupan je i na GitLab repozitoriju: **https://gitlab.com/mkolarevic/echo**